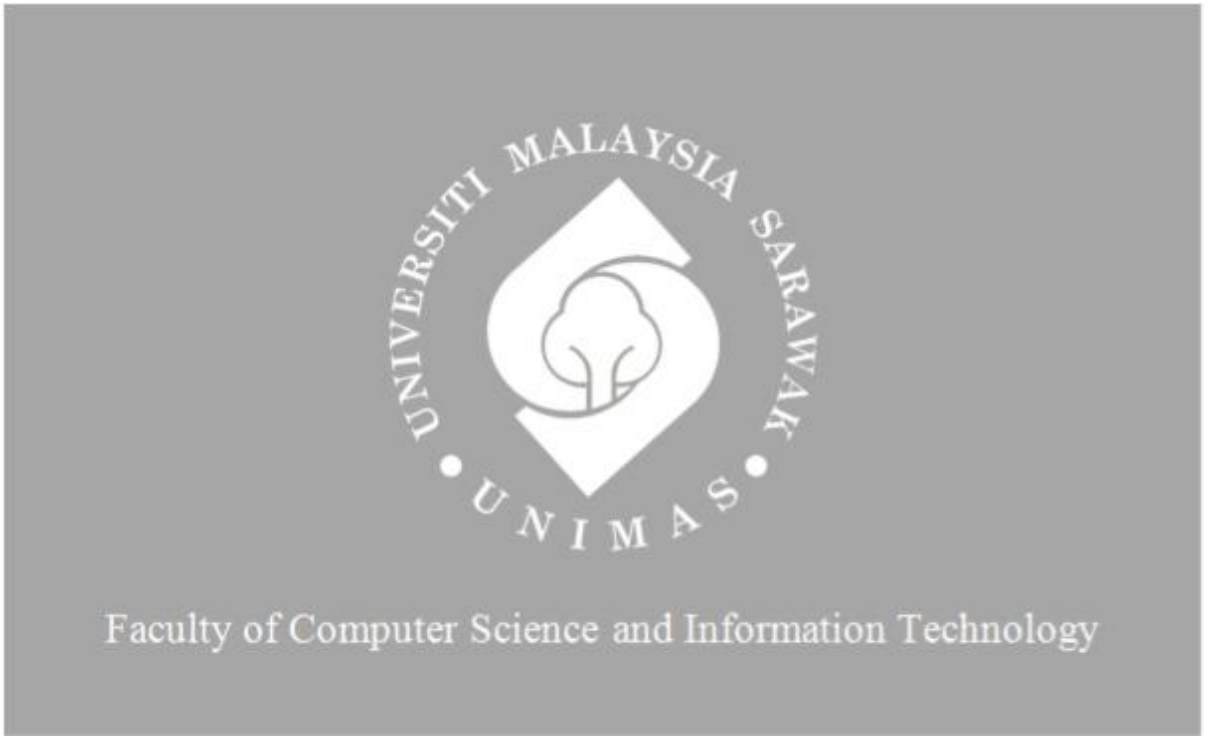


UniEXP: A GAMIFIED UNIVERSITY EVENT MANAGEMENT SYSTEM

Ling Sie Jie
(79893)

Bachelor of Software Engineering with Honours

2025

UNIEXP: A GAMIFIED UNIVERSITY EVENT MANAGEMENT SYSTEM

LING SIE JIE

This project is submitted in partial fulfilment of the

requirements for the degree of

Bachelor of Software Engineering with Honours

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2025

UNIEXP: A GAMIFIED UNIVERSITY EVENT MANAGEMENT SYSTEM

LING SIE JIE

Projek ini merupakan salah satu keperluan untuk

Ijazah Sarjana Muda Kejuruteraan Perisian dengan Kepujian

Fakulti Sains Komputer dan Teknologi Maklumat

UNIVERSITI MALAYSIA SARAWAK

2025

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE UNIEXP: A GAMIFIED UNIVERSITY EVENT MANAGEMENT SYSTEM

ACADEMIC SESSION: 2024/2025

LING SIE JIE
(CAPITAL LETTERS)

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only.
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database.
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (√)

☐	CONFIDENTIAL	(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)
☐	RESTRICTED	(Contains restricted information as dictated by the body or organization where the research was conducted)
☑	UNRESTRICTED	

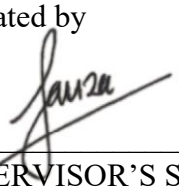
(AUTHOR'S SIGNATURE)

Permanent Address

NO. 73, LORONG 4A1, NEW AIRPORT ROAD, BORNEO GARDEN, 93250 KUCHING, SARAWAK

Date: 7th July 2025

Validated by



(SUPERVISOR'S SIGNATURE)

Ts. Nurfauza Binti Jali
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak

Date: 7/7/2025

Note * Thesis refers to PhD, Master, and Bachelor Degree
** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

DECLARATION

I hereby declare that the project entitled "UniEXP: A Gamified University Event Management System" is my original work. It has not been copied from the work of any other student or source, except where due reference or acknowledgement is explicitly made in the text. No part of this project has been written for me by another person.

(LING SIE JIE, 79893)

Faculty of Computer Science and Information Technology

Software Engineering

Universiti Malaysia Sarawak.

ACKNOWLEDGEMENT

This project would not have been possible without the guidance, support, and resources provided by various individuals. I extend my deepest gratitude to Universiti Malaysia Sarawak (UNIMAS) for providing the necessary resources and a conducive environment for completing this study. My sincere thanks to my supervisor, Ts. Nurfauza bt Jali, for her invaluable guidance, constructive feedback, and constant encouragement throughout this project. I am also grateful to Associate Professor Dr. Sze San Nah, the examiner, for offering constructive critiques and valuable insights that significantly enhanced the quality of my work. Prof. Wang Yin Chai, the FYP coordinator, played a pivotal role in coordinating and overseeing the final year project. I am grateful for the support and collaboration of Miss Diana Tracy ak Delim and students who participated in the user requirements gathering and testing phases, providing critical insights that shaped the system's development. Lastly, I express my deep appreciation to my family and friends for their unwavering support and encouragement, which motivated me during challenging times.

ABSTRACT

This project presents UniEXP, a gamified university event management system developed to overcome low student engagement and fragmented event coordination at Universiti Malaysia Sarawak (UNIMAS). Existing reliance on tools like WhatsApp and Google Forms resulted in inefficient registration and participation processes. UniEXP provides a centralized platform consisting of a mobile app for students and a web-based system for event organisers, integrating gamification features such as points, leaderboards, badges, virtual currency, and event-specific quests to motivate active involvement. Testing with 30 students revealed that all respondents agreed the gamification features significantly enhanced their event experience, with rewards, virtual currency, and achievement badges fostering repeated participation and overall satisfaction. Additionally, all students expressed willingness to continue using UniEXP for future events. On the organiser side, evaluation by a university staff member confirmed that the system effectively streamlined event creation, quest setup, attendance tracking, and performance analytics. Features like participant statistics and the student leaderboard provided valuable insights for assessing engagement and planning future initiatives. The unified platform eliminated the need for multiple disjointed tools, improving efficiency and clarity in event management. Developed using Feature-Driven Development (FDD) within the Agile framework and built with React Native, React, and Firebase, UniEXP successfully enhances both the student event experience and the administrative process, contributing to a more interactive and efficiently managed university event ecosystem.

ABSTRAK

Projek ini membentangkan UniEXP, sebuah sistem pengurusan acara universiti berasaskan gamifikasi yang dibangunkan bagi mengatasi isu penglibatan pelajar yang rendah dan penyelarasan acara yang tidak berpusat di Universiti Malaysia Sarawak (UNIMAS). Sebelum ini, penganjur acara bergantung kepada alat komunikasi tidak formal seperti WhatsApp dan Google Forms, yang menyebabkan proses pendaftaran dan kehadiran menjadi tidak efisien. UniEXP menawarkan platform berpusat yang merangkumi aplikasi mudah alih untuk pelajar dan sistem berasaskan web untuk penganjur acara, dengan integrasi elemen gamifikasi seperti mata, papan pendahulu, lencana pencapaian, mata wang maya, dan misi acara bagi mendorong penyertaan aktif. Hasil ujian bersama 30 orang pelajar menunjukkan bahawa kesemua responden bersetuju bahawa elemen gamifikasi telah meningkatkan pengalaman mereka terhadap acara universiti, di mana ganjaran dan ciri interaktif mendorong penglibatan berulang serta kepuasan keseluruhan. Semua pelajar turut menyatakan kesediaan untuk terus menggunakan UniEXP bagi acara-acara akan datang. Bagi pihak penganjur, penilaian oleh wakil staf universiti mengesahkan bahawa sistem ini berjaya memudahkan proses penciptaan acara, pengurusan misi, penjejakan kehadiran, serta analisis prestasi. Ciri statistik dan papan pendahulu pelajar terbukti berguna dalam menilai tahap penglibatan pelajar serta merancang acara yang lebih berkesan pada masa akan datang. Platform yang bersatu ini juga menghapuskan keperluan menggunakan pelbagai alat berasingan, sekaligus meningkatkan kecekapan pengurusan. UniEXP telah dibangunkan menggunakan pendekatan Feature-Driven Development (FDD) dalam kerangka Agile serta dibina dengan React Native, React, dan Firebase. Secara keseluruhan, UniEXP berjaya menjadikan pengurusan acara universiti lebih menarik dan efisien, serta mewujudkan budaya kampus yang lebih interaktif dan terurus.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Full Term
AI	Artificial Intelligence
API	Application Programming Interface
APK	Android Package Kit
BaaS	Backend-as-a-Service
CLI	Command Line Interface
DOM	Document Object Model
FCSIT	Faculty of Computer Science and Information Technology
FDD	Feature-Driven Development
GPS	Global Positioning System
ID	Identification
IDE	Integrated Development Environment
OS	Operating System
PKP	Pusat Khidmat Pelajar or Centre for Student Services
Q&A	Question and Answer
QR	Quick Response
SDK	Software Development Kit
SPA	Single Page Application
UAT	User Acceptance Testing
UI	User Interface
UID	Unique Identification
UNIMAS	Universiti Malaysia Sarawak
WOM	Word-of-Mouth
WTP	Willingness-to-Participate

Chapter 1. INTRODUCTION

1.1. Introduction

Events within every university play a critical role in contributing to both students' academic and social development (Lubbers & Joyce, 2014). Universiti Malaysia Sarawak (UNIMAS) has hosted such events throughout the year, from academic conferences to cultural festivals, all for the purpose of fostering a sense of community, encourage networking, and promoting the knowledge exchange beyond the classroom setting (Karamchandani et al., 2024; Madan & Lohani, 2024). However, managing an event efficiently would become a challenging task as it requires organisers to handle large numbers of participants and coordinate multiple activities. Therefore, an organised event management system is much needed for providing students and organisers with a seamless shared platform to plan, register, and engage in those events.

Effective event management in universities is referred as a pivotal to the success of events in academic settings. Research (Trowler, 2010; Bello, 2024) highlighted that an effective event management system not only lessen the administrative burden but also improves the coordination and communication between organisers and participants, leading to be increased event success rates and better student satisfaction. On the other hand, events risk becoming chaotic when there is without a structured management process. This usually led to poor participation rates, inefficient communication between parties, and logistical issues.

In today’s digital blooming age, the demand for such systems has become essential as universities rely heavily on them to streamline processes such as managing event registration, attendance tracking, and real-time engagement monitoring. However, many traditional systems, even those that have gone digital, often lack the interactive and motivating elements that encourage student involvement, resulting in reducing the overall impact of the events.

This is where gamification steps in. More recently, there has been a surge of interest in gamification as an effective solution to increase user engagement in non-entertainment contexts, such as education and event management (Zainuddin et al., 2020; Kalogiannakis et al., 2021). Gamification involves incorporating game-like elements, mechanics, and experiences in non-gaming contexts to enhance user motivation and making the experience more enjoyable and rewardable (Deterding et al., 2011). Therefore, the integration of gamification into university event management can transform how students interact with and perceive events, encouraging more active involvement and improving event impacts (Sisson & Whalen, 2021).

The "UniEXP" project, a mobile application for UNIMAS undergraduate students and a web-based platform for event organisers, offers a centralized and user-friendly solution for efficiently managing university event participation. It features tools for event creation, registration, and attendance tracking, while incorporating gamification elements like points, badges, and leaderboards to encourage student actively engaging in the events and enhance the overall event experience.

1.2. Problem Statement

UNIMAS currently lacks a centralized platform to manage and promote university events effectively. In contrast, event organisers rely heavily on informal channels like WhatsApp and Facebook, which are not intended to manage these events. This approach causes fragmented communication for the undergraduate students to possible miss important event announcements or receive them late, making it difficult for them to stay informed about upcoming events (Yu et al., 2020). Therefore, event organisers struggle to attract student and face inefficiencies in managing registrations that are often time-consuming and error prone. Moreover, there is no formal system in place to track student participation at events, making it

challenging to assess student involvement, quantify event impact, or reward active engagement through various recognition methods. Additionally, those existing traditional event management methods often fail to capture undergraduate students' interest to attend events. Their lack of interactive and engaging features resulted in many undergraduate students attend passively without engaging with the activities meaningfully, which diminishes the overall effectiveness and impact of such events. To address these challenges, a gamified university event management system is proposed. This is proven by the research carried out by Zainuddin (2020) and Kalogiannakis et al. (2021) showing that incorporating gamification elements such as points, badges, and leaderboards can significantly boost user engagement, yet this is underutilised by universities.

1.3. Scope

Table 1.1. *Project Scope Classification*

Aspect	Details
System Purpose	• Enhance student engagement. • Streamline event management at UNIMAS
Gamification Features	• Faculty leaderboard • Virtual currency & rewards • Achievement badges • Event-specific quests
Technologies Used	• **Frontend**: React Native (mobile app), React (web system) • **Backend**: Firebase (database, authentication, and cloud function)
Problems Addressed	• Low student engagement and participation • Inefficient event registration and overall management processes
Target Users	• **Primary**: UNIMAS undergraduate students (mobile app users) • **Secondary**: Faculty-based event organisers (web-based users)
Environment Limitations	• Primarily for undergraduates • Limited functionality for pre-U students, postgraduates, staff • GPS-based attendance may raise privacy concerns

1.4. Aims and Objectives

The main aim of UniEXP is to leverage the gamification elements into an ordinary event management system to enhance undergraduate student engagement and event participation rates. To achieve this, the following objectives have been established:

1. To analyse existing event management systems that incorporate gamification elements.
2. To design and implement the UniEXP system, a digital platform for undergraduate students and administrators.
3. To test and evaluate the event management system for usability and operational efficiency.

1.5. Brief Methodology

The software development approach of choice is the agile methodology. First, the agile methodology is an incremental and iterative approach to software development that welcomes ongoing client input and is flexible enough to adapt to often-evolving demands (Esang et al., 2024). When the requirements change often, it helps with bug fixes, expenses, and workloads (Abrahamson et al., 2002). According to Sharma et al. (2012), the agile methodology incrementally divides larger projects into smaller, more manageable modules that are independent of one another. All the finished modules are then combined into a system.

Figure 1.1. *Feature Driven Development (FDD) Processes (Feature Driven Development (FDD), 2023)*

Among the agile software development frameworks, the feature-driven development (FDD) is chosen to develop this system. FDD is an agile methodology that focuses on building software through small, functional features. It follows an iterative approach where features are developed and delivered incrementally, ensuring continuous progress. Prioritization is based on business value and impact, allowing the most critical functionalities to be addressed first. This structured yet flexible approach enhances efficiency, collaboration, and alignment with business goals. The rationale of this choice is that it focused quality aspects throughout the process by frequently delivering features (Abrahamsson et al., 2002). Furthermore, FDD is more suitable for developing critical systems (Palmer & Felsing, 2002). Based on Figure 1.1, FDD consists of five-step development process to follow.

1.5.1. Develop an Overall Model

The FDD framework starts with developing an overall model known as domain object modelling for the system. Besides, an activity diagram is drawn to represent the workflow or sequence of actions within the system, illustrating how tasks and processes flow from start to end (Elivera & Palaoag, 2020). From the diagrams created, the necessary modules of the proposed system are concluded.

1.5.2. Build a Features List

From the first phase, a features list from the system module is concluded to understand better the project scope for features list development (Elivera & Palaoag, 2020).

1.5.3. Plan by Feature

In this phase, the features in the list are analysed and evaluated by categorized them into several priority: high, medium, low. Moreover, a Gantt chart to assign the schedule and track for the activities was depicted (Elivera & Palaoag, 2020).

1.5.4. Design by Feature & Build by Feature (Iterative Phases)

These two phases refer to the construction phase which are iterative for each feature set (Elivera & Palaoag, 2020) based on the priority. In Design by Feature phase, a use case diagram is depicted to show how different parts of the system will interact with users or other systems and a list of sequence diagrams for the features in the use case diagram were designed (Elivera & Palaoag, 2020) to show the step-by-step interactions between objects or components in a system. Furthermore, a class diagram of the system is designed to provide a blueprint for system implementation by modelling the static structure of a system by representing its classes, their attributes, methods, and relationships (What is FDD in Agile?, 2021). Moreover, wireframe of the system is designed in this phase (Elivera & Palaoag, 2020). Lastly, the Build by Feature phase requires to develop the system and implement a series of system testing. The system development iteration continues with another feature set.

1.6. Significance of Project

The significance of developing "UniEXP" lies in several key dimensions:

1. **Enhanced Undergraduate Student Engagement:** "UniEXP" leverages the gamification elements such as leaderboards, quests, virtual currency, levelling system, and achievements to increase student motivation and participation in university events.
2. **More organised and streamlined event management**: The system equips both event organisers and undergraduate students with the functionality of event organisation, registration, and attendance tracking, making the event management more efficient.
3. **Increased Awareness and Participation in Events**: The system able to raise awareness among the undergraduate students to participate in events in a more

interactive and fun way through gamification which acted as a motivator to compete and recognise their involvement.

4. **Potential for Broader University Applications**: UniEXP's scalable design offers potential for broader applications across other university functions although it focuses on events. For instance, it could inspire similar systems for managing other student services, supporting learning communities, or facilitating faculty-student interactions, showcasing how gamification can enhance diverse aspects of university life.

1.7. Expected Outcomes

The expected outcomes of the "UniEXP" project include a highly user-friendly mobile application designed to offer undergraduate students a seamless experience in browsing, registering for, and actively engaging in university events. This approach is intended to make events more accessible and appealing, as those students can easily explore upcoming activities, track their involvement, and receive automated notifications directly through the app. By integrating gamification elements, such as points, badges, rewards, and recognition systems, the application aims to boost participation by creating incentives for undergraduate students to attend more events and interact within the campus community. This added layer of motivation is anticipated to foster a more vibrant and active campus culture, as undergraduate students are encouraged to explore diverse activities, socialize, and engage more deeply with university life.

On the administrative side, event organisers are expected to experience significant improvements in operational efficiency through a web-based dashboard designed specifically for event management tasks. This dashboard will streamline processes like event setup including quests or challenges for each event, tracking attendance, and managing registrations, thereby reducing manual effort and minimizing the likelihood of errors. By offering tools for real-time coordination, the system is also likely to enable quicker, more effective

communication and organization, making event planning less time-consuming and enhancing the overall quality and success of campus events. This dual benefit approach is designed to enrich the experience for both undergraduate students and organisers, building a more integrated, interactive, and efficiently managed university event ecosystem.

1.8. Summary

This chapter introduces "UniEXP," a gamified university event management system developed for Universiti Malaysia Sarawak (UNIMAS) to address issues arising from the current use of fragmented, informal communication channels that result in low student engagement, missed announcements, and inefficient event processes. Supported by Lubbers & Joyce's (2014) assertion that university events are crucial for students' academic and social development, the project offers a mobile app for undergraduate students that incorporates gamification elements such as points, leaderboards, and badges to enhance engagement.

Chapter 2. LITERATURE REVIEW

2.1. Introduction

This chapter starts off with an overview of project objectives. Then, this chapter illustrates the literature reviews of three systems that are similar to the proposed system. The main functionalities, types of gamification elements integrated into the system, and advantages and disadvantages of all these systems are examined and analysed to conclude a comparison among those systems with the proposed one. Besides that, this chapter discusses the tools and technologies to be used in developing the proposed system.

2.2. Literature Study

2.2.1. Gamification

Gamification involves incorporating game-like elements, mechanics, and experiences in a service to make the experience more engaging and increase client retention (Huotari & Hamari, 2012). It is aimed to encourage people to change their habits and develop their skills or to foster innovation for achieving goals (Welbers et al., 2019; Alsawaier, 2018). The concept of gamification started from a study by a professor at MIT Sloan School of Business, Thomas W. Malone titled “What Makes Things Fun to Learn, A Study of Intrinsically Motivating Computer Games” in 1980 which discussed how video games assisted children to learn (Chitroda, 2015). He then published the studies titled “Toward a Theory of Intrinsically Motivating Instruction” and “Heuristics for Designing Enjoyable User Interfaces” in the following year which focused on the learning outcomes of computer games (Chitroda, 2015). Until then, the impact of computer games used in engaging people begun to get recognised in the years running up to the millennium’s end.

The term “gamification” was initially used by Nick Pelling in 2002 to more motivational explain his work as a consultant (Chitroda, 2015). However, the real integration

of gamification elements into a platform, Bunchball, was introduced by Rajat Paharia in 2005 (Chitroda, 2015). The gamification of work began with this aim to make it more enjoyable (Dale, 2014). Since then, gamification has quickly taken off and has become a successful strategy because of its favourable behavioural shifting effect and growing interest in the game business among people of all demographics (Chitroda, 2015). However, gamified interactions need to be both interactive and challenging to be successful (Berger et al., 2018).

Gamification has a lot of promise, and with the correct combination of game elements, it can be a big value to a company (Markopoulos et al., 2015). According to Hamari et al. (2014) and Mazarakis (2021), it is suggested that gamification such as points, badges, and leaderboards can positively influence user engagement and motivation. However, Hamari et al. (2014) stated that these benefits are highly context-dependent and influenced by factors such as user type and the nature of the activity being gamified.

Based on the study conducted by Banyte and Gadeikiene (2015), user motivation and its relationship to video game engagement were extensively examined. Their research framework categories user motivation into three distinct types: intrinsic (internal drives), extrinsic (external rewards), and experiential (experience-based factors). In examining user engagement, they adopted a multifaceted approach that takes into account cognitive processes, emotional responses, and behavioural patterns that manifest at both the individual and gameplay levels. The researchers concentrated on four distinctive aspects of gaming engagement: the depth of immersion, the sense of presence within the game world, the state of flow during gameplay, and the level of absorption in gaming activities. Their analysis proved that experience-seeking motivation is the most potent driver, particularly in enhancing players' sense of presence, while significantly affecting their immersion and flow states. Internal motivation demonstrated a particular influence on immersion levels, whereas surprisingly, external motivators linked to game mechanics showed minimal impact on overall engagement.

The study also revealed that absorption levels remained constant regardless of motivational factors. As a result, Banyte and Gadeikiene's study (2015) holds beneficial discoveries to the stakeholders from game development teams to educators exploring gamification strategies for enhancing learning process.

Yee's (2016) study identified six distinct clusters of motivations for playing games: Action, which involves excitement and destruction; Social, centred on collaboration and competition; Mastery, driven by strategy and the desire to overcome challenges; Achievement, focused on gaining power and completing goals; Creativity, associated with design, exploration, and discovery; and Immersion, which reflects engagement with storylines and fantasy elements.

GAMEFULQUEST, an instrument introduced by Högberg et al. (2019) is a reliable tool to capture the positive emotions and motivations inspired by games and their influence on user behaviour. As an outcome, the gameful experience is described through seven dimensions: accomplishment (goal achievement and progress), challenge (engaging tasks requiring effort), competition (rivalry with others), guided (direction or assistance during the experience), immersion (deep involvement in the activity), playfulness (fun and enjoyment), and social experience (interaction and connection with others).

Until today, gamification strategies can be used in several fields and in any stage of the consumer behaviour process (Guittard et al., 2015). Many firms resort to gamification as a marketing practice by implementing it on their websites or e-commerce applications to encourage consumers to buy their products/services (Högberg et al., 2019). Tourism companies, like Lufthansa, have created online games that enable participants to view and experience some destinations (Kiryakova, et al., 2014). In hospitality, "advergames" are also used to create awareness and publicity of hotels and destinations worldwide, creating the desire

on tourists' minds, incentivising them to travel (Kiryakova, et al., 2014). In education, gamification has been applied to Moodle ,one of the most well-known platforms for teachers or lecturers to share notes and documents of the classes, including achievement tools, user´s pictures or avatars, reflection of levels and display of quiz results (Andreev, 2023). In general, Prasad (2021) stated some simple rules to create a gamification integrated application such as achievements, leaderboards, motivation, challenge, rewards, points, and badges.

2.2.2. Gamification and Event Management

The events industry has changed its approach to interacting and capturing attendees' attention, converting inactive spectators into active participants (Sisson & Whalen, 2021).

Through a gamified mobile application designed to engage new students, Fitz-Walter et al. (2012) examines the use of gamification in university orientation sessions. The project focuses on developing, implementing, and assessing an application that incorporates gaming features like leaderboards, awards, and challenges based on QR codes to encourage students to tour the campus, interact with classmates, and discover university services. Survey and use data results show different degrees of participation, with QR code challenges being especially successful at encouraging campus exploration. But issues including poor setup coordination, the lack of attractiveness of some game features (such text entry problems), and the requirement for continuous assistance during the event were noted. The results emphasise the value of iterative design modifications to maximise user experience while highlighting the potential of gamification to increase engagement in event-based contexts. This study offers insightful information about how gamified tactics may be used to increase university event participation.

According to Thibault et al. (2021), "Etsijä's Call" was introduced to refer within the broader work on gamification to be employed in events. The authors believe that experience symbolises a push to rediscover the core principles of gamification as a tool for cultural

enrichment and empowerment during the events. They emphasised that the gamification can be both a useful tool and a lens to examine societal interactions, as it should be implemented in an ethical, inclusive, and well-contextualised manner.

The study by Sisson and Whalen (2021) can be acted as a guideline for university events in general as it explored the impact of gamification on positive behavioural outcomes in hospitality events settings. By employing self-determination theory and belongingness as theoretical frameworks, the researchers linked pre-event game communication to attendees' perceptions of value, willingness to participate (WTP), word-of-mouth (WOM) intentions, and emotional commitment. As a result, it highlights that gamification positively enhances attendees' emotional connection to events, increases their willingness to engage, and fosters positive WOM communication. Notably, the perceived value of gamification—encompassing functional, social, and emotional dimensions—plays a pivotal role in shaping these outcomes. This underscores the importance of integrating game elements to enhance event experiences and encourage participation.

O'Connell et al. (2020) investigate how emergency medicine residents might continue their medical education during the COVID-19 pandemic by using gamification in virtual events. Inspired by the television program "So You Think You Can Dance," the research details a planned gamified event where participants used Zoom to participate in interactive, competitive games that reviewed fundamental obstetrics and gynaecology concepts. Team-based events were held in breakout rooms during the event, which ended with a final face-off round. Most attendees preferred this dynamic format over conventional lecture-style events, according to post-event questionnaires, which showed high levels of participant involvement. Technological limitations and uneven facilitation were recognised as obstacles. However, this study emphasises how gamified events may improve learning, enjoyment, and engagement—especially in virtual settings.

2.3. Review on Similar Existing Systems

2.3.1. Goosechase

Figure 2.1. *Goosechase Logo (Goosechase - Apps on Google Play, n.d.)*

Figure 2.1 illustrates the logo of Goosechase, a mobile application that was originated by GooseChase Adventures in 2011 during a hackathon in Waterloo, Canada (*About Goosechase*, n.d.). Goosechase is a mobile application designed to facilitate dynamic scavenger hunts, allowing users to engage in mission-based challenges during the events through their smartphones (*About Goosechase*, n.d.). It is available on both major mobile operating systems: Android, where it holds a rating of 4 out of 5 based on over 2200 reviews on the Google Play Store, and iOS, where it has a rating of 4.8 out of 5 from more than 12000 reviews on the Apple App Store. Over the years, it has become popular in various sectors, including corporate training and education, because of its adaptability and simplicity of use. By leveraging mobile technology, Goosechase is a pertinent case study for examining technology-driven engagement in diverse contexts to bridge the gap between conventional and contemporary event participation approaches.

Figure 2.2. *Goosechase's Search for Experiences & Join via Unique Codes*

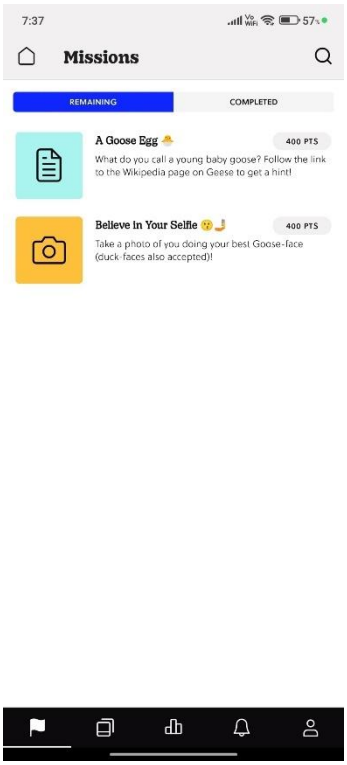


Figure 2.3. *Goosechase's Missions List Displaying Active Challenges*

Figure 2.4. *Goosechase's Completed Missions List*

Figure 2.5. *Goosechase's Event Leaderboard*

Figure 2.6. *Goosechase's Notifications List*

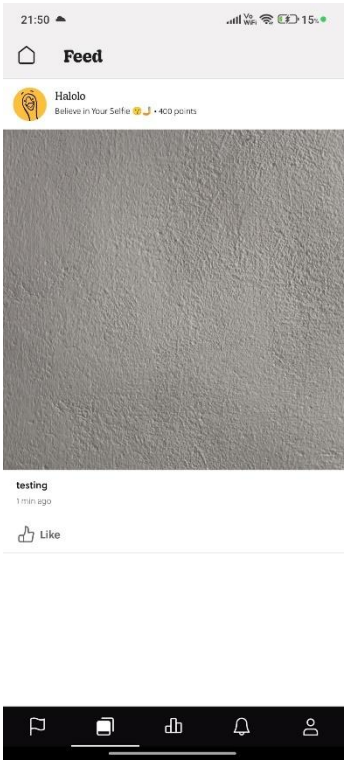


Figure 2.7. *Goosechase's Post Feed*

Goosechase provides a scavenger hunt platform for both admins and users. Goosechase provides users a smooth treasure hunt event experience on mobile devices. Using a special code or search functionality, users may join events alone or in groups as shown in Figure 2.2. Referencing Figure 2.3, the application provides a mission list with detailed instructions and point values for each task. Upon submission approval, participants can complete missions by submitting photo, video, text, or GPS-based evidence, with instant feedback from admins. Furthermore, Figure 2.4 showcases the application's ability to review the completed missions for that specific event. Real-time updates, including leaderboard standings as shown in Figure 2.5, keep users engaged and competitive throughout the game. Through leaderboard gamified mechanism, participants can earn points and badges for completing missions and compete to improve their rankings on the leaderboard. This creates a friendly environment for the users to compete with others. Figure 2.6 depicted that Goosechase provides the feature of notifications

which alert users to important updates such as approved submissions or admin announcements. Furthermore, Figure 2.7 illustrates Goosechase offered the feature that allows liking on submissions, while users can also share their achievements or highlights on social media to enhance the sense of community and engagement.

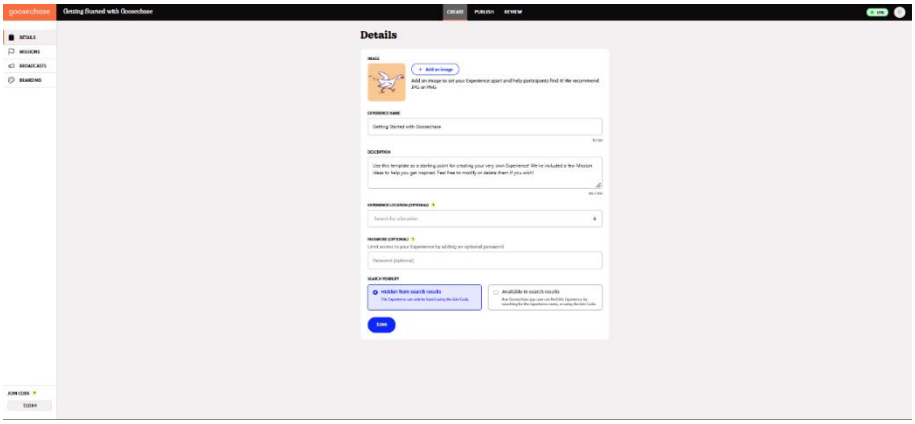


Figure 2.8. *Goosechase's Experiences/Events Management*

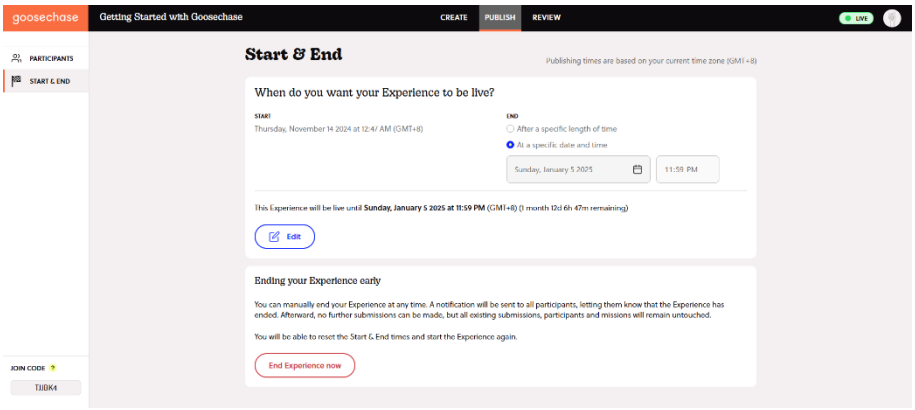


Figure 2.9. *Goosechase's Event Start & End Time Setup*

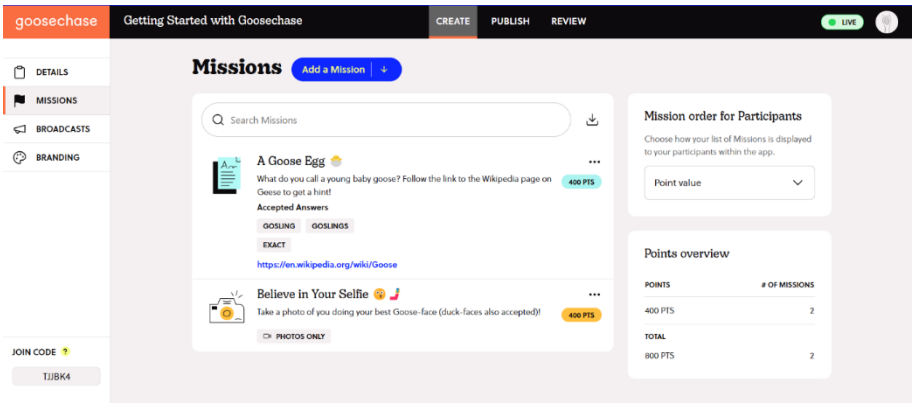


Figure 2.10. *Goosechase's Missions Setup*

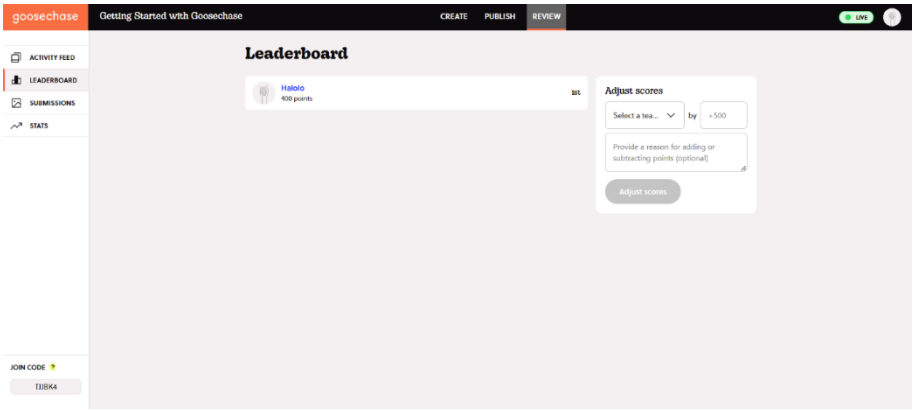


Figure 2.11. *Goosechase's Leaderboard (Admin)*

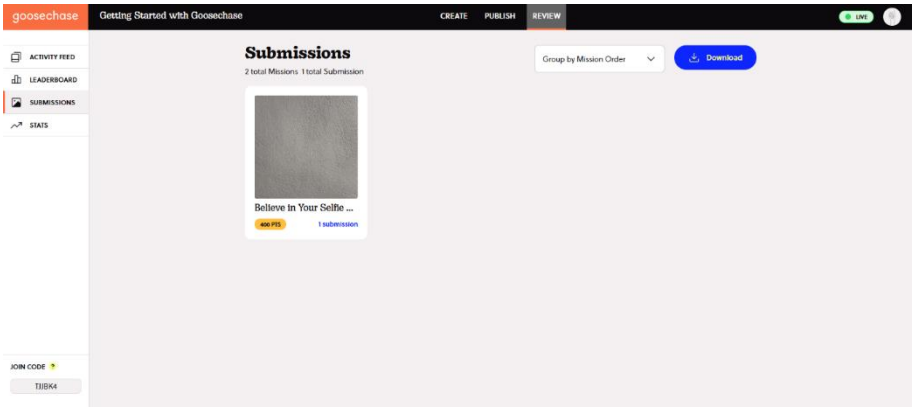


Figure 2.12. *Goosechase's Submissions Section (Admin)*

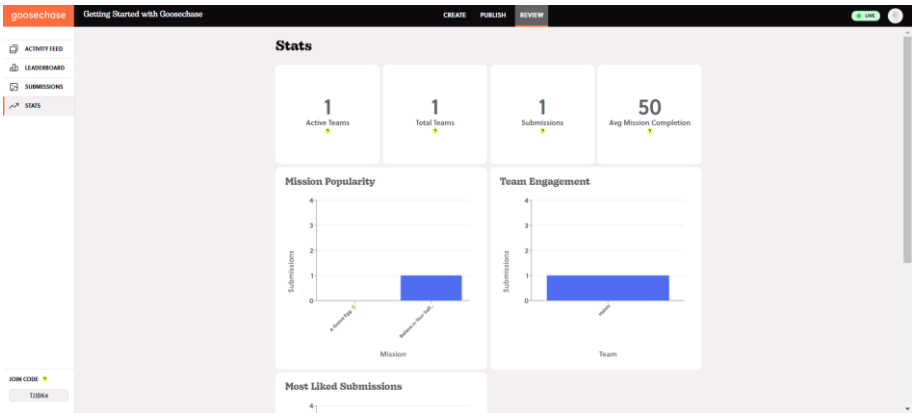


Figure 2.13. *Goosechase's Event Reporting Feature*

On the other hand, admins only can access the platform through website for creating, managing and monitoring scavenger hunt events. From Figure 2.8, admins can set up games by customising titles, descriptions, images, themes, and visibility, and they can choose between public or private modes with a unique join code. They can opt to set up location for physical

events and passwords for further accessibility security. The event's start and end times can also be defined and edited through the "Publish" navigation tab on the administrative site as shown in Figure 2.9. The game's missions, which might be photo, video, text, or GPS-based, can be greatly altered. Figure 2.10 illustrates that each mission has a point value, and administrators can create their own objectives or use pre-made templates for convenience. Along with customised instructions, these missions can be assigned to teams or people. Organising participants or groups and keeping an eye on their activity in real time make participant management simple. Figure 2.11 depicts a live leaderboard enabling administrators to monitor progress, evaluate contributions (text, images, or videos), and respond promptly by accepting or rejecting submissions (see Figure 2.12). During the events, points can also be dynamically changed. Administrators may assess completion rates, monitor engagement, and get performance data using analytics and reporting tools when the event is over. Participants can get notifications immediately, guaranteeing prompt announcements, reminders, and event status updates. Lastly, Figure 2.13 depicts Goosechase offers event reporting features that allow admins to review statistics such as active and total teams, number of submissions, average mission completion rate, mission popularity, team engagement levels, most liked submissions, and leaderboard which was beneficial for the event success evaluation and future event planning.

Goosechase generally offers users significant benefits through its gamification elements such as leaderboards and quests. These features make events livelier and more enjoyable, promoting teamwork and eliminating passive participant involvement. The platform's pre-built templates and customisable missions based on theme or goals of the events simplify the setup process for event planners. Additionally, real-time monitoring allows organisers to track progress and engagement, while the platform's data collection capabilities provide valuable insights and content for post-event analysis or promotion.

However, there are drawbacks to Goosechase, especially when used for formal or extremely planned events. Its emphasis on gamification might not be suitable for academic or professional contexts where features like ticketing and thorough attendance monitoring are necessary. Although it is easy to use for routine searches, there could be a higher learning curve for making intricate, highly personalised missions. Furthermore, the collection of participant data, including images and videos, may give rise to privacy issues if not handled properly.

2.3.2. Whova

Figure 2.14. *Whova Logo (Press & Media - Whova, n.d.)*

Figure 2.14 illustrates the logo of Whova, a comprehensive event management mobile application that was established in 2013 by Dr. Yuanyuan Zhou (*What is Whova?*, n.d.). Whova is an event management solution that aims to revolutionise events, attract and engages attendees effectively, and help event organisers save time when managing event logistics (*What is Whova?*, n.d.). It is available on both major mobile operating systems: Android, where it holds a rating of 4.8 out of 5 based on over 26500 reviews on the Google Play Store, and iOS, where it has a rating of 4.8 out of 5 from more than 194600 reviews on the Apple App Store. Whova's platform consists of an award-winning event application, easy online registration, powerful event marketing, and time-saving event management tools.

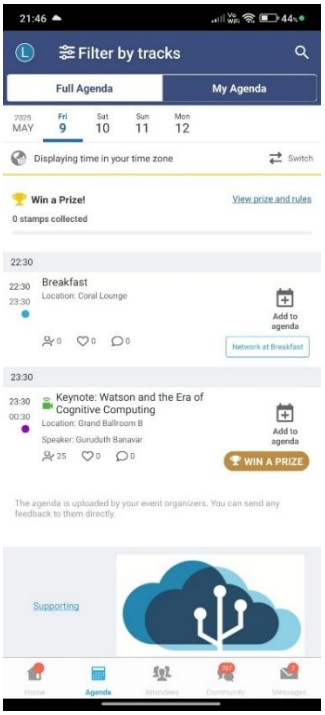


Figure 2.15. *Whova's Agenda Calendar*

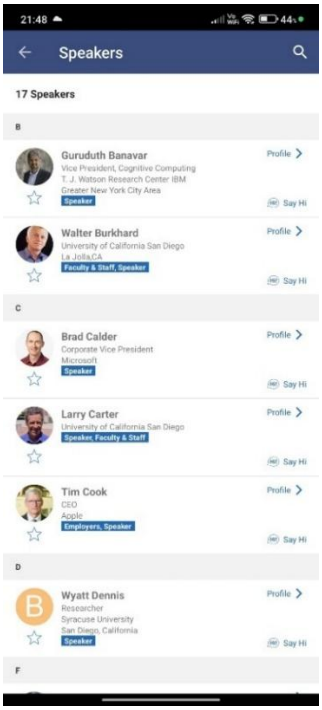


Figure 2.16. *Whova's Speakers Page*

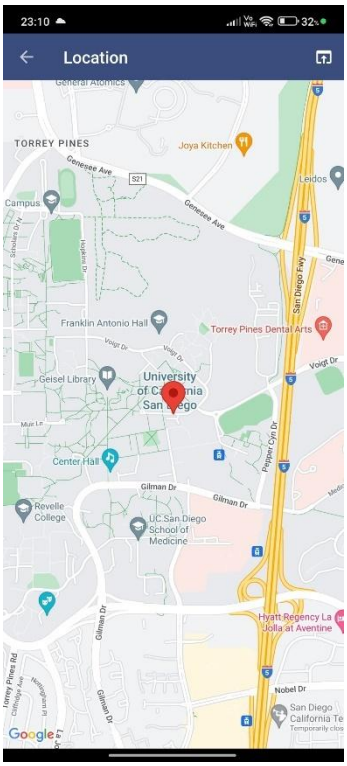


Figure 2.17. *Whova's Location Map*

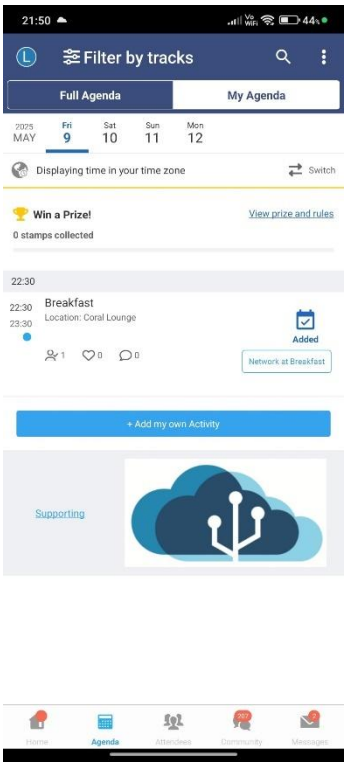


Figure 2.18. *Whova's Personalised Agenda*

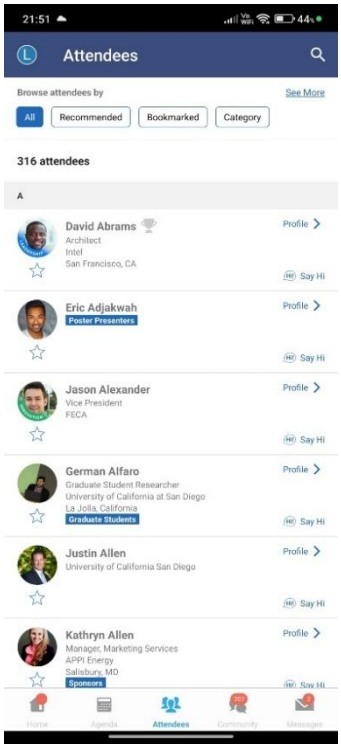


Figure 2.19. *Whova's Attendees Page*

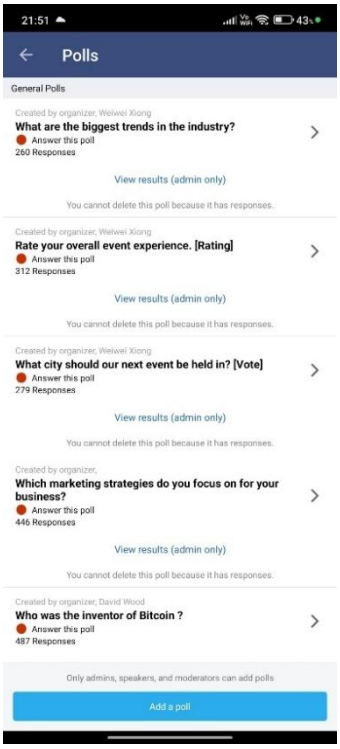


Figure 2.20. *Whova's Polls*

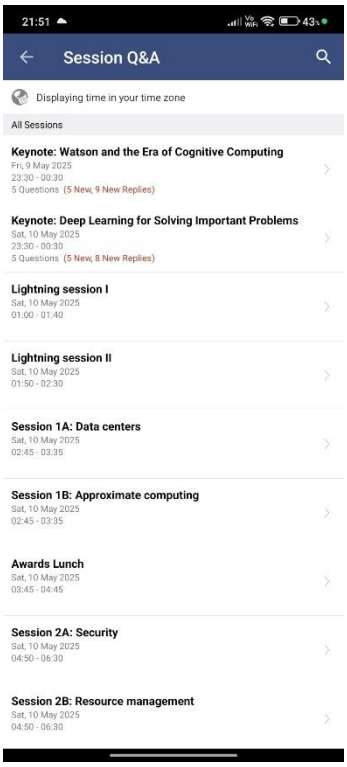


Figure 2.21. *Whova's Q&A Session*

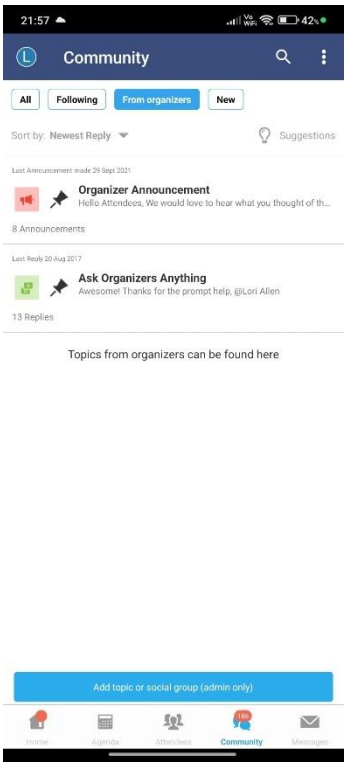


Figure 2.22. *Whova's Community Section*

The Whova application gives users a smooth and all-inclusive platform to improve their event experiences. It acts as a central repository for information on events such as agendas (see

Figure 2.15), speaker biographies (see Figure 2.16), and venue maps (see Figure 2.17). Furthermore, Figure 2.18 illustrates that Whova enables users to create customised schedules by choosing sessions or activities that pique their interest, assisting them in time management and preventing them from missing significant occasions. Based on Figure 2.19, Whova also enables users to send messages and view attendee profiles for creating deep connections before, during, and after the event. Whova further increases user engagement by enabling users to actively participate in sessions rather than only see them through live interaction tools like polls (see Figure 2.20) and Q&A sessions (see Figure 2.21). Based on Figure 2.22, real-time announcements and chatbots in Whova facilitate communication between participants and organisers to prevent misunderstandings in case of schedule or location changes.

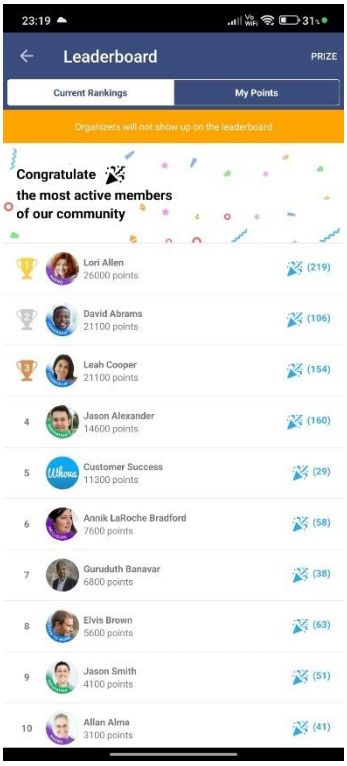


Figure 2.23. *Whova's Leaderboard*

Figure 2.24. *Whova's Photo Contest*

Figure 2.25. *Whova's Icebreaking Contest*

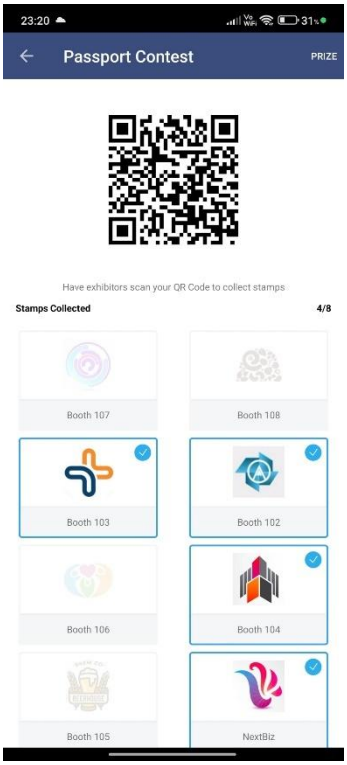


Figure 2.26. *Whova's Passport Contest*

Gamification features incorporated into the application to improve participant participation and make the event more engaging and fun. Figure 2.23 shows the leaderboard is used in Whova to encourages healthy competition among participants. As participants post event images and compete for the most likes, the photo contest promotes creativity and interaction as shown in Figure 2.24. Furthermore, Figure 2.25 depicts the icebreaker game mechanism integrated into Whova application to encourage interaction by awarding players whose answers to entertaining or thought-provoking questions get the most comments. Finally, the passport contest in the application as shown in Figure 2.26 gamifies booth visits by scanning their QR codes at all booths.

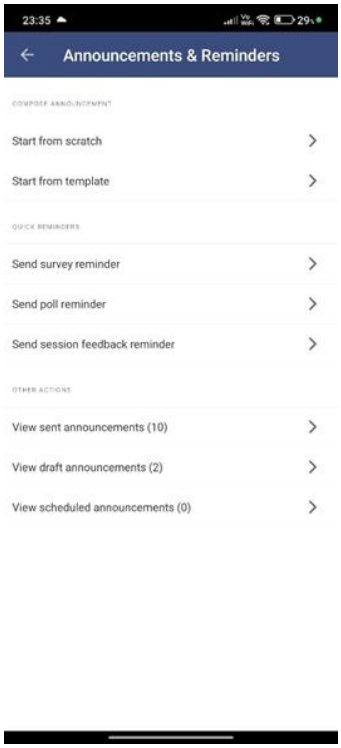


Figure 2.27. *Whova's Admin Announcements & Reminders Section*

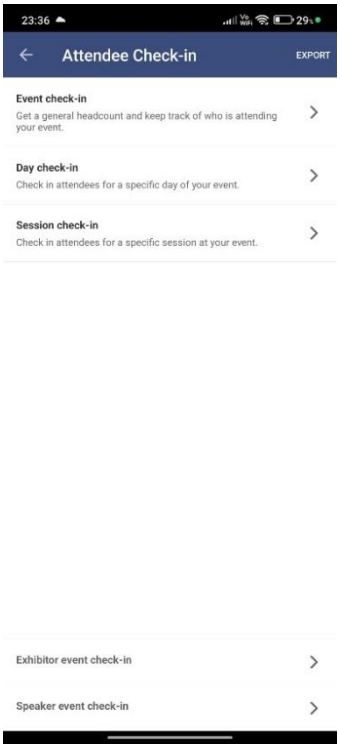


Figure 2.28. *Whova's Attendance Tracking & Manual Attendance Module*

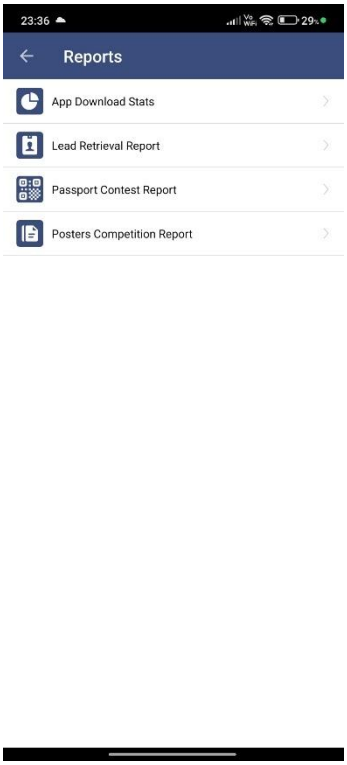


Figure 2.29. *Whova's Admin Reports Module*

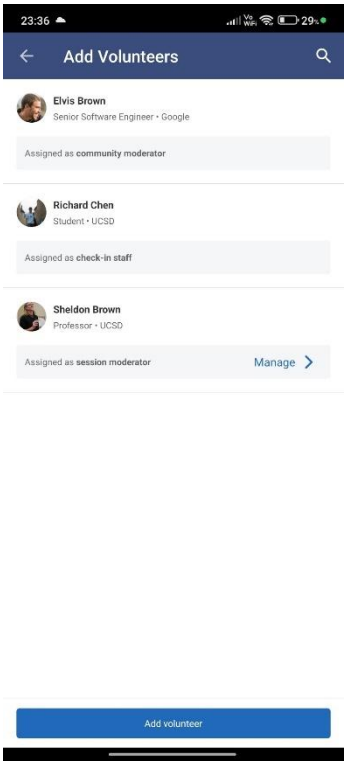


Figure 2.30. *Whova's Volunteering Module*

On the other hand, admins can compose and send timely updates to attendees as illustrated in Figure 2.27. They can craft announcements either from scratch or by using templates for convenience. Additionally, they can send quick reminders for various purposes, and manage announcements effectively by viewing sent, drafted, or scheduled messages. Furthermore, Whova simplifies the attendee, exhibitor, and speaker check-in process, as shown in Figure 2.28. Figure 2.29 depicts Whova providing the report analysis feature to reduce the administrative burden by providing detailed insights into multiple aspects of the event. To further support event execution, the application includes a volunteering module that allows admins to assign volunteers for specific events, as shown in Figure 2.30.

In general, Whova simplifies event planning while offering attendees easy access to schedules, announcements, and interactive features such as polls and Q&A. The application's networking features, such as attendee profiles and private messaging, enhance engagement, especially in professional settings. Additionally, Whova's analytics and gamification features, like leaderboards, add value by boosting participation and providing insights into attendee behaviour. However, Whova's main drawback is its user interface, which can be confusing and unintuitive. While helpful, the application's extensive features often overwhelm users, making navigation difficult, particularly for first-time attendees.

2.3.3. Eventzee

Figure 2.31. *Eventzee Logo (Eventzee Scavenger Hunt - Apps on Google Play, n.d.)*

Figure 2.31 illustrates the logo of Eventzee, a scavenger hunts mobile application for event settings that was established in 2014 by Freeze Tag, Inc. (Freeze Tag, Inc., 2022). Eventzee is an application that enables organisers to design and manage interactive scavenger hunts and challenges for various audiences. It is commonly used in settings such as corporate events, tourism campaigns, educational activities, and recreational events. It is available on both major mobile operating systems: Android, where it holds a rating of 4.6 out of 5 based on over 200 reviews on the Google Play Store, and iOS, where it has a rating of 4.8 out of 5 from more than 300 reviews on the Apple App Store. Eventzee seeks to leverage gamified elements to transform traditional events into engaging and memorable experiences.

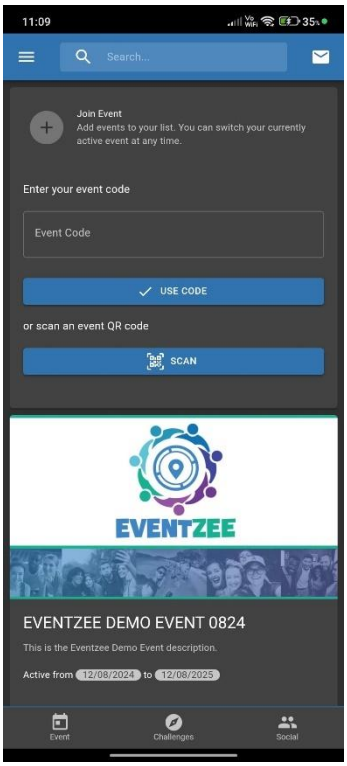


Figure 2.32. *Eventzee's Event Joining Methods & Event Details*

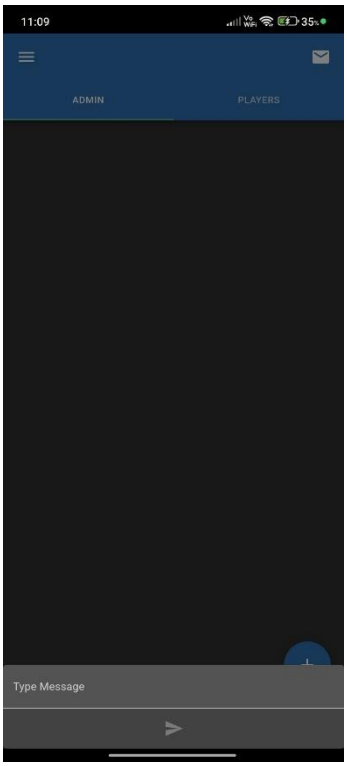


Figure 2.33. *Eventzee's Networking Feature*

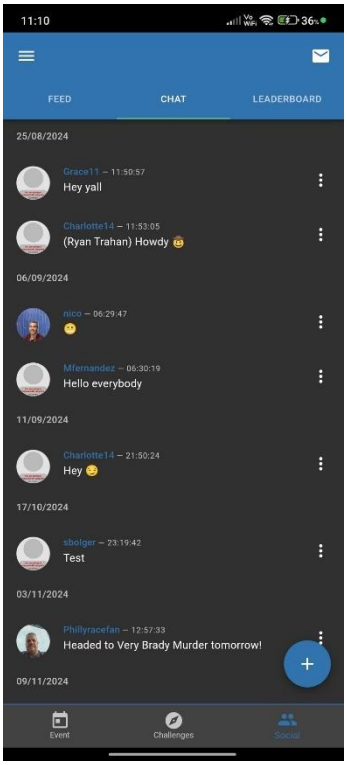


Figure 2.34. *Eventzee's Community Board*

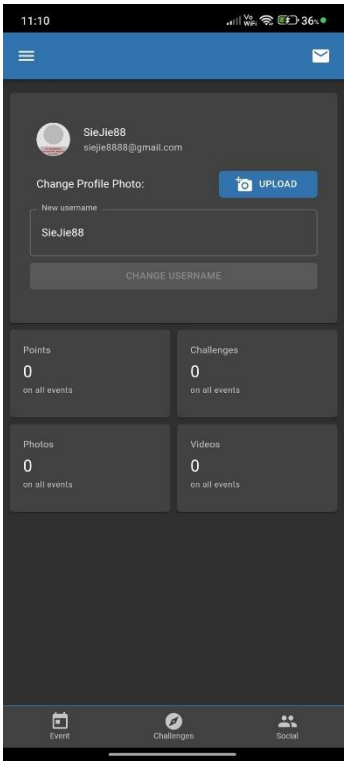


Figure 2.35. *Eventzee's User Profile*

Based on Figure 2.32, Eventzee allows users to easily join events by scanning an event-specific QR code or inputting a unique event code. Users are given clear information about those events, such as their description, start date, and finish date. As shown in Figure 2.33, Eventzee's networking function further promotes connectedness by facilitating communication between users and event organisers or individual participants. Additionally, users may interact and work together on a shared community board, as seen in Figure 2.34. Eventzee users may effortlessly update their accounts, including changing their username and profile picture and getting a detailed activity report that includes points won, activities accomplished, and media taken during events, as shown in Figure 2.35.

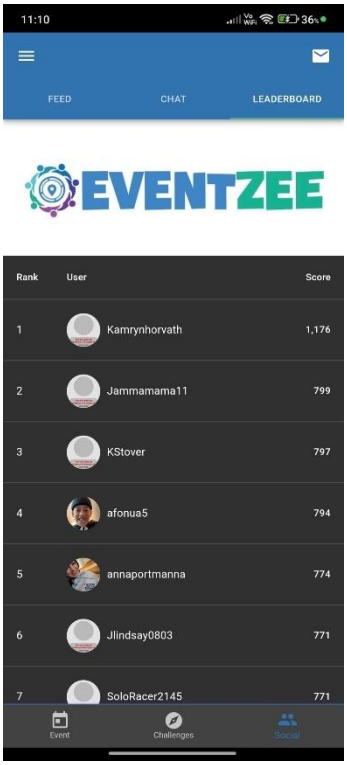


Figure 2.36. *Eventzee's Leaderboard*

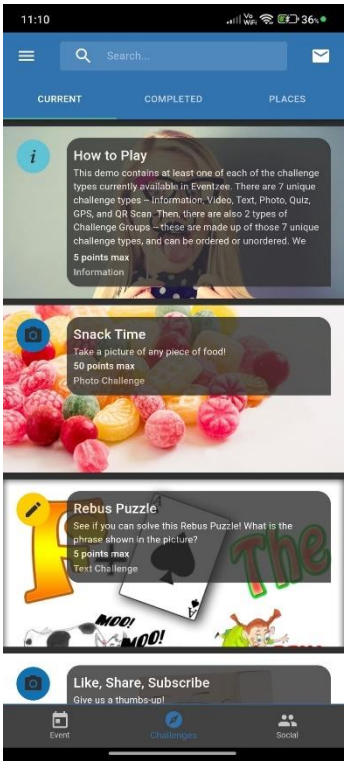


Figure 2.37. *Eventzee's Event Challenges*

Figure 2.38. *Eventzee's Achievement Badges Recognition*

In the context of gamification, Eventzee features an event-specific leaderboard, as illustrated in Figure 2.36, where participants can track their rankings in real-time based on points earned during the event. Moreover, as shown in Figure 2.37, Eventzee includes gamified challenges such as photo, video, quiz, QR code, GPS, text input, info, and group challenges to keep participants engaged and motivated. Based on Figure 2.38, Eventzee recognises participants' achievements through badges, a key gamification element. These gamified features combine to create a comprehensive and interactive platform that transforms event participation into an engaging and rewarding experience.

This platform is particularly valued for its capacity to design customised, activity-focused experiences that encourage participant involvement and competitiveness. Eventzee is well-known for being user-friendly and adaptable, making it appropriate for informal and formal settings. Applications include educational challenges for students, citywide scavenger hunts for tourists, and corporate team-building activities. These varied use examples show how flexible and successful the platform is at encouraging active engagement. However, the absence of a direct event search function is a significant drawback of the Eventzee mobile application. Based on Figure 2.31, users can only join events by providing a unique event code or scanning a QR code. This dependence limits prospects for wider participation by preventing users from participating in other events they might find interesting. The platform's ability to encourage impromptu involvement and link users to a greater variety of events is limited in the absence of a centralised event directory or search feature.

2.3.4. Table for Comparison between Systems

Table 2.1. *Comparison between Existing Systems*

	Goosechase	Whova	Eventzee	UniEXP
Platform Focus	Scavenger hunt event platform for team-based activities and missions.	Event management with attendee engagement and networking features.	Gamified scavenger hunt and event engagement.	Gamified university event management system.
Gamification Features	Customisable missions, live scoreboards, and team/individual competition.	Leaderboards, trivia games, photo and video contests, icebreakers, passport contest, and community-based challenges.	Event-specific challenges, leaderboards, and achievement badges recognition	Event-specific quests, leaderboard, virtual currency for redeeming merchandise and achievement badges.
Registered events calendar or agenda	No	Yes	No	Yes
Attendance Check-in Tools	No	Yes (QR code based)	No	Yes (QR code based and GPS verification)
Event Analytic Module	Yes	Yes	No	Yes
Networking Features	No	Yes	Yes	Yes, but in the format of quest
Volunteering Module	No	Yes	No	Yes

2.4. Development Tools and Technologies

2.4.1. Programming Languages and Frameworks

This section provides an overview of the programming languages utilised to develop the proposed application.

2.4.1.1. React Native

React Native is a popular open-source framework developed by Meta Platforms (formerly Facebook, Inc.) in 2015. It allows developers to build mobile applications for both Android and iOS platforms using JavaScript and React (Vaghela, 2023; Bakr, 2024). Essentially, React Native is React adapted for mobile development, using XML-esque markup (JSX) for designing user interfaces. React Native employs an abstraction layer known as the "bridge," which enables it to invoke the native rendering APIs on iOS and Android (Eisenman, 2016).

The framework simplifies application development by allowing developers to maintain a single codebase, offering a cheaper and faster way to create cross-platform apps without compromising on quality (Vaghela, 2023). Developers working with React Native can choose between two main tools: the React Native Command Line Interface (CLI) and Expo CLI. For this project, Expo CLI is used because it provides a fully managed development environment, reducing the complexity of setting up native development tools and dependencies. Moreover, in most cases, Expo eliminates the need to install Android Studio or Xcode during development, further streamlining the process.

React Native has been chosen as the frontend framework for this project because its cross-platform nature ensures accessibility to a diverse student base using different operating systems (OS). Its strong community support and extensive library ecosystem are essential for the project's development.

2.4.1.2. React

React, also known as ReactJS, is an open-source JavaScript library developed by Facebook to simplify the process of building user interfaces (Herbert, 2022), particularly for single-page applications (SPAs). First released in 2013, it revolutionised front-end development with its declarative programming model and component-based architecture, enabling developers to build modular and reusable UI elements (Camus, 2022). By leveraging a virtual document object model (DOM), React efficiently calculates the minimal updates required for the real DOM, enhancing rendering performance and speeding up updates (Sebastian, 2024; *W3Schools*, n.d.). One of its standout features is JSX, a syntactic extension that combines HTML-like code with JavaScript, streamlining development and improving readability (Camus, 2022).

In this project, React will be used to develop the administrative (event organiser) site due to its ability to create reusable components, ensuring design consistency and accelerating development. Its virtual DOM further optimises performance, making React particularly well-suited for dynamic admin dashboards that handle frequent updates and large amounts of data efficiently.

2.4.2. Software

This section provides an overview of the software utilised for developing the proposed application.

2.4.2.1. Expo Go

Expo Go is a sandbox environment that makes it easier to develop and test React Native applications that use Expo CLI (Moedano, 2024). It was available for both iOS and Android to eliminate the need for emulators and complete builds by allowing developers to preview and

interact with their projects in real-time on physical devices. The project is loaded and reflected the changes in real time-by scanning a QR code or clicking on a link when the project is running through Expo CLI (Moedano, 2024). The software equipped with this “hot reloading” feature avoided the delays of rebuilding and reinstalling projects (*Development and Production Modes*, n.d.). For this project, the mobile application will be developed and tested using Expo Go because it is simpler to test and play without writing native code as it comes preloaded with a variety of libraries and application programming interfaces (APIs). It serves as a bridge between the React Native development environment and physical devices, providing an easy and efficient way to debug applications.

2.4.2.2. Firebase

Firebase is a platform brought by Google that offers a wide range of services designed to help build and manage mobile and web applications more effectively. In fact, Firebase originated from Envolve, a startup founded by James Tamplin and Andrew Lee in 2011. Envolve initially provided an API for integrating online chat functionality into websites. However, developers began using it to sync application data, not just chat messages. This led Tamplin and Lee to pivot, founding Firebase in September 2011 as a separate company publicly launched in April 2012 (*Firebase Introduction – Javapoint*, n.d.). Until today, it provides backend solutions for real-time databases, authentication, cloud storage, push notifications, and more. Firebase Authentication streamlines user management and sign-in procedures across many platforms (Google, Facebook, Twitter, etc.) and Firebase's core features, such Firebase Realtime Database and Cloud Firestore, enable developers to sync data in real time between users effortlessly.

2.4.2.3. Visual Studio Code

Visual Studio Code, often referred to as VS Code, is a free, powerful, and lightweight source code editor developed by Microsoft, designed to serve developers using a variety of programming languages and frameworks. It is compatible with Windows, macOS, and Linux, therefore developers can run it on their desktop or through the website (Heller, 2022). It has recently been more well-liked in the development community due to the flexibility, cross-platform compatibility, and enriched features that enhance the project development lifecycle. VS Code offers a user-friendly interface paired with tools such as syntax highlighting, error detection, and IntelliSense, which provides smart code completion (*Why Visual Studio Code?*, 2021). The editor also allows a wide range of extensions, making it ideal for a wide range of project requirements. The integrated terminal within the editor simplifies project execution, allowing developers to run commands directly without switching to a separate command prompt. This feature makes the editor particularly well-suited for working on React and React Native projects.

2.5. Summary

This chapter examined the key components behind the development of the UniEXP system, focusing on the objective of using gamification to boost engagement in university events. It reviewed related literature to highlight gamification's motivational potential and compared existing platforms like Goosechase, Whova, and Eventzee to identify useful features and challenges. The chapter also evaluated the suitability of tools like Expo, Firebase, and React Native in achieving the project’s goals.

Chapter 3. METHODOLOGY

3.1. Introduction

This chapter includes the design of the suggested system, requirements analysis, and methodology. Agile Feature Driven Development (FDD) is the methodology employed; this will be covered in more detail in the following section. Diagrams of the activity, system modules, use case, use case requirements, class, and sequence will also be added based on the conversation. A system wireframe is presented to show how the proposed system would seem in order to give a clear visual representation.

3.2. Requirements Collection and Analysis

As seen in Figure 3.1, the requirements collecting and analysis phase is a critical prelude to the implementation of the Feature-Driven Development (FDD) approach inside the Agile framework in this project. Gathering requirements from the intended users is the goal of this phase. The requirements of the intended users have been gathered for the proposed system using two distinct methods: questionnaires and interviews.

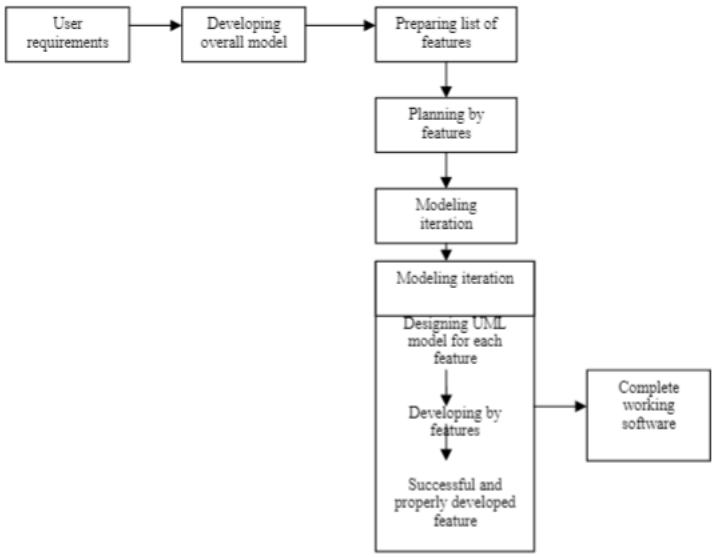


Figure 3.1. *Method of Developing Agile Processes using FDD Graph (Sharma et al., 2012)*

3.2.1. Questionnaire

During this phase, questionnaires as shown in Appendix B were distributed to a group of target users to gather their requirements and opinions about the proposed system. A total of 100 respondents across different years of study and faculties at UNIMAS had participated in the survey. The questionnaire was structured into three sections: Section A focused on demographic information, Section B explored feedback and experiences related to UNIMAS event management, and Section C collected opinions on potential features and gamification elements. The collected responses were analysed and summarized below.

3.2.1.1. Section A: Demographic Information

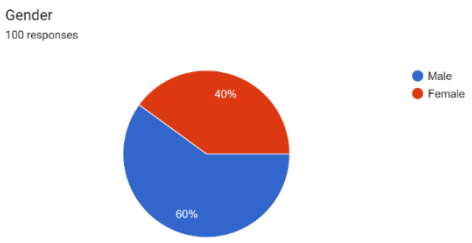


Figure 3.2. *Gender Distribution of Respondents*

In this section, the basic demographic information of respondents is collected to better understand them. Based on Figure 3.2, it depicts the gender distribution of respondents, showing that 60% (60 respondents) are male, while 40% (40 respondents) are female.

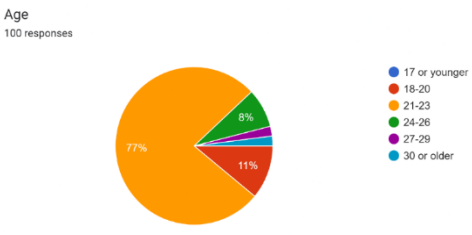


Figure 3.3. *Age Range Distribution of Respondents*

Based on Figure 3.3, most respondents fall within the 21-23 age range (77%) followed by 18-20 age range (11%) and 24-26 age range (8%).

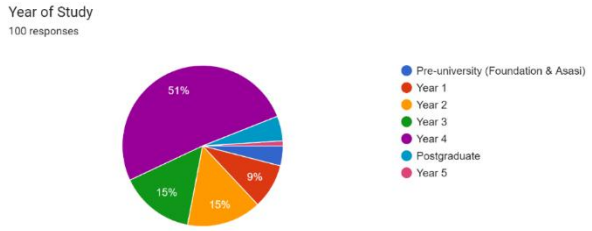


Figure 3.4. *Year of Study Distribution of Respondents*

Figure 3.4 illustrates the distribution of respondents across various years of study, with 4% in Pre-university (Foundation and Asasi), 9% in Year 1, 15% in Year 2, 15% in Year 3, 51% in Year 4, 1% in Year 5, and 5% in Postgraduate. From the analysis, year 1 until year 4 UNIMAS students contributed the most to the questionnaire, which aligns with the primary target users (undergraduate students) of this project.

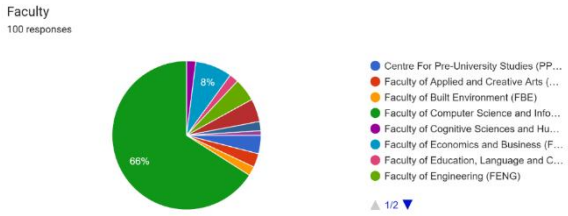


Figure 3.5. *Faculty Distribution of Respondents*

As seen in Figure 3.5, the majority of respondents are from Faculty Computer Science and Information Technology (FCSIT), comprising 66% of the respondents, suggesting a strong technical background of this group in providing input on features and functionalities.

3.2.1.2. Section B: UNIMAS Event Management, Feedback & Experience

This section investigates into the current methods used in event management process at UNIMAS. The feedback of the current event management process and the overall experience are quantified using Likert Scale format.

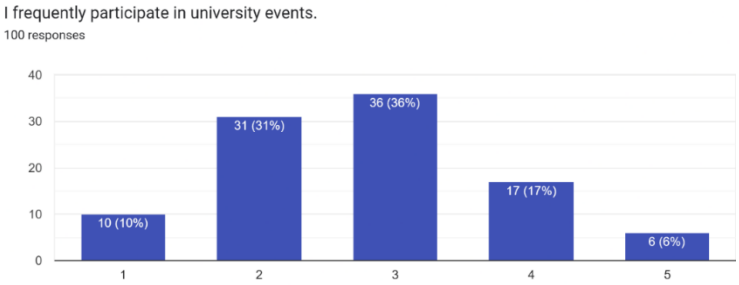


Figure 3.6. *Analysis of Respondents' Frequency in Participating in University Events*

According to the data presented in Figure 3.6, the analysis highlights the frequency with which survey participants engage in university events. The information shows that the largest group of respondents (36%) reported a moderate level of participation, while 31% indicated they seldom take part in such events. Additionally, 10% of respondents selected the "very rare" level of participation. On the higher end of the spectrum, 17% of respondents stated they participated frequently, while 6% reported very frequent participation in university events. Overall, the data suggests that most university students participate in events at a moderate or lower frequency, with only a minority engaging frequently or very frequently.

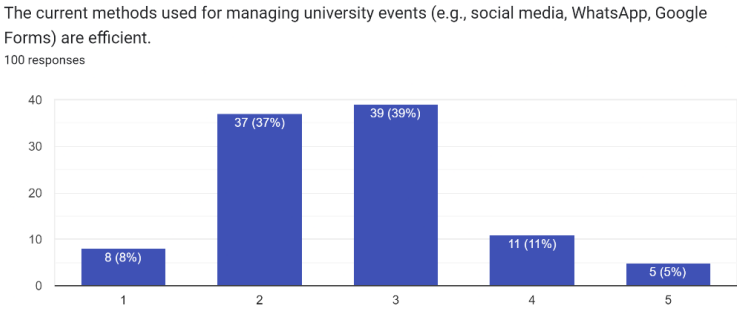


Figure 3.7. *Analysis of Effectiveness on Current University Event Management Methods*

Based on Figure 3.7, the analysis reveals the respondents' perception of the efficiency of current methods used for managing university events, such as social media, WhatsApp, and Google Forms. The majority of respondents (39%) rated the efficiency as neutral (3), indicating mixed opinions about the effectiveness of these methods. This is closely followed by 37% of respondents who chose to disagree (denoted as 2) with the efficiency of current methods used. Additionally, 8% of respondents rated the statement as strongly disagree (1), while only a smaller percentage rated the methods positively, with 11% choosing a rating of 4 (agree) and 5% selecting the highest rating of 5 (strongly agree). This data suggests that the current methods for managing university events may not be meeting user expectations and might benefit from improvements or alternative approaches to enhance their efficiency and user satisfaction.

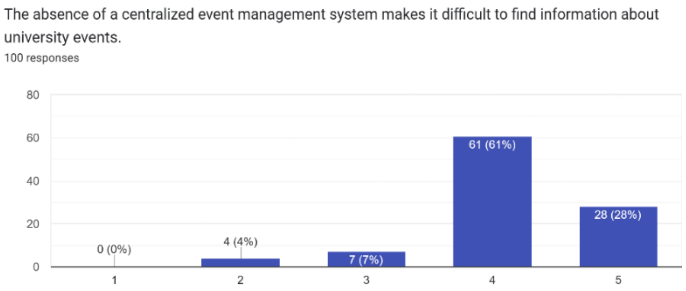


Figure 3.8. *Respondents' Opinions on the Difficulty of Finding Information About University Events Without a Centralised Management System*

The analysis emphasizes respondents' perceptions of the challenge of locating information regarding university activities because of the lack of a centralized event management system, as shown in Figure 3.8. The sentiment that the absence of a centralized system is a challenge is further reinforced by the fact that 28% of respondents fully agree (denoted as 5) and the majority (61%) strongly agree (denoted as 4). However, a small percentage of respondents disagreed, with 7% choosing neutral (denoted as 3), 4% opting to disagree (denoted as 2), and 0% strongly disagreeing (denoted as 1). According to this research, the majority of respondents believe that one major obstacle to obtaining event-related information is the lack of a centralized event management system.

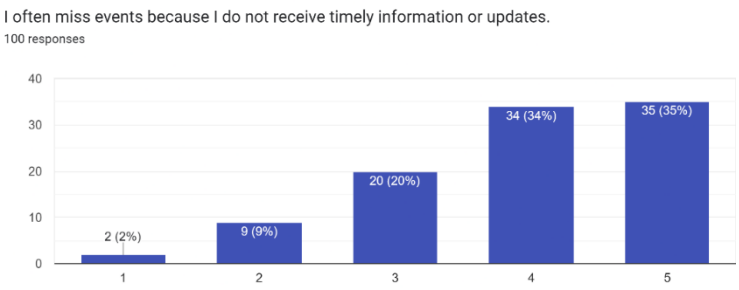


Figure 3.9. *Analysis of Respondents' Agreement with the Statement on Missing Events Due to Lack of Timely Information or Updates*

The study shows that respondents agreed with the statement that they frequently miss events because they don't receive timely information or updates (see Figure 3.9). 34% of respondents agree (denoted as 4) and 35% strongly agree (denoted as 5) with the statement, indicating that a sizable percentage of students have difficulty getting timely updates about events. Furthermore, just 9% and 2% of respondents disagreed (denoted as 2) or strongly disagreed (denoted as 1), respectively, while 20% of respondents chose a neutral opinion (denoted as 3). In order to guarantee more attendance and awareness, this data shows that delayed communication is a frequent problem, underscoring the need for better notified methods for reminders.

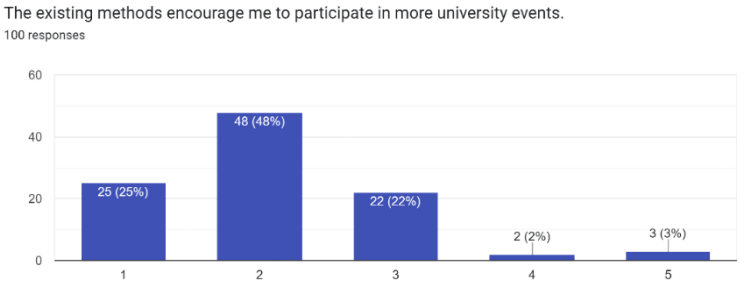


Figure 3.10. *Respondents' Level of Agreement on Whether Existing Methods Encourage Participation in University Events*

The figure above illustrates the survey results, which show that the majority of respondents do not strongly agree that the current approaches motivate them to attend more university events. Significantly, 25% of respondents (25 in all) gave this statement a level 1

agreement rating, meaning they strongly disagree with it. Furthermore, 48% of respondents (48) gave it a level 2 rating, indicating that the majority of them disagree. 22% (22 respondents) gave their agreement a level 3 rating, which is on the moderate side and denotes a neutral position. This is in contrast to the 3% (3 respondents) who evaluated it at level 5, indicating strong agreement, and the 2% (2 respondents) who rated it at level 4, indicating agreement. The majority of participants, according to the data, believe that the current approaches are insufficient to encourage them to participate more actively in university events.

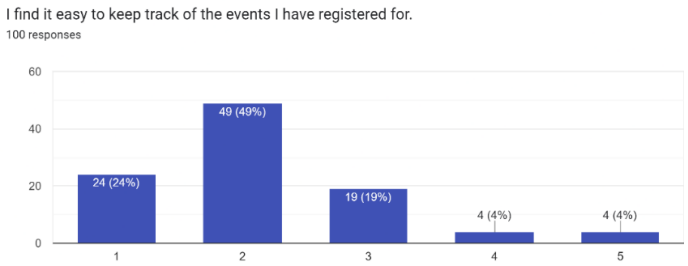


Figure 3.11. *Respondents' Level of Agreement on the Ease of Tracking Registered Events*

According to this data, survey results show that different people find it easier or harder to keep track of occurrences that have been registered. Twenty-four percent (24 respondents) strongly disagreed that it is easy to keep track of events, giving their experience a level 1 rating. Furthermore, 49 respondents, or 49% of the sample, chose level 2, indicating that they largely disagree with the statement. On the other hand, 19% (19 respondents) gave their experience a level 3 rating, indicating that they had no opinion about how easy it was to track occurrences. However, just 4% (4 respondents) chose level 4 to rate their experience, agreeing that it is easy, and 4% (4 respondents) chose level 5 to strongly agree with the assertion. These findings show that only a small percentage of participants found it easy to maintain track of their recorded events, with a large chunk finding it difficult.

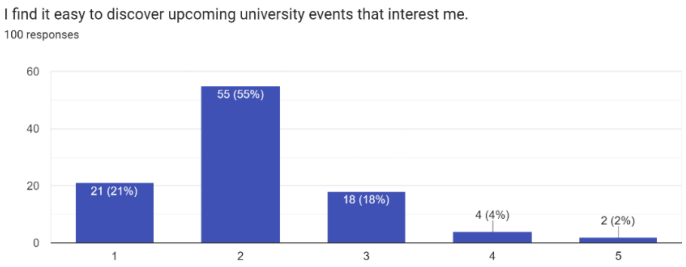


Figure 3.12. *Analysis of Students' Ease in Discovering Upcoming Events of Interest*

Based on Figure 3.12, the survey responses reveal varying levels of agreement regarding the ease of discovering upcoming university events of interest. A total of 21% of the respondents strongly disagree with the statement, giving it a rating of 1. Furthermore, 55% of the respondents rated it at level 2, indicating that they mostly disagree. On the neutral side, 18% of the respondents rated their agreement at level 3, suggesting they neither agree nor disagree. On the higher end of the scale, 4% of the respondents rated their experience at level 4, agreeing that it is easy to discover events, and 2% of the respondents strongly agreed with the statement, giving it a rating of 5. The results indicate that the majority of participants find it challenging to discover upcoming events that interest them. This suggests an opportunity to enhance event promotion strategies to improve accessibility and awareness.

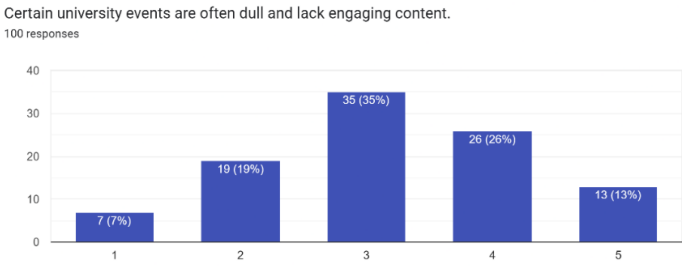


Figure 3.13. *Analysis on Perception of University Event Engagement*

Based on this figure, the analysis shows that most of the respondents (35%) provided a neutral response (denoted as 3), suggesting they neither agree nor disagree with the statement that certain university events are dull and lack engaging content. However, there are 26% and 13% who rated their agreement at levels 4 and 5, respectively, showing that a portion of

respondents find university events lacking in engagement. Meanwhile, only 19% and 7% of the respondents disagree with the statement at scales 2 and 1, respectively.

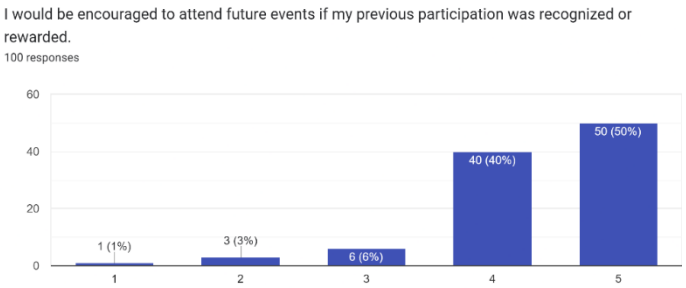


Figure 3.14. *Analysis of Student Perceptions on Attendance Motivation through Recognition and Rewards*

According to the analysis, the majority of respondents (50%) agreed with the statement, as seen in this figure; 40% agreed with the statement at scale 4, indicating a strong desire for prizes or recognition as a motivator, while the remaining 40% agreed with the statement at scale 5. On the other hand, just a tiny percentage of respondents expressed less interest in this strategy, with only 1% choosing strongly to disagree (denoted as 1), 3% choosing to disagree (denoted as 2), and 6% choosing neutral (denoted as 3).

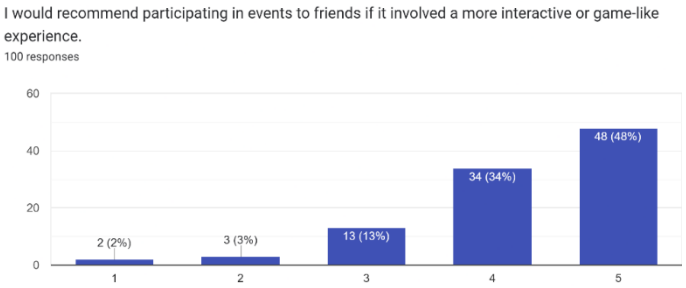


Figure 3.15. *Analysis of Respondents' Opinions on Relationship Between Gamified Event Experiences and Friends Recommendations*

As seen from Figure 3.15, the analysis shows a strong positive response toward recommending university events if they involved a more interactive or game-like experience. Most respondents (48%) strongly agree (denoted as scale 5) with the statement, while 34% agree (denoted as scale 4). This indicates that over 80% of respondents would be encouraging

their friends to participate if events were designed to be more engaging and interactive. A smaller portion of respondents (13%) were neutral (denoted as scale 3) with the statement, while only 3% and 2% expressed disagreement (denoted as scale 2 and scale 1, respectively). In general, the idea of incorporating gamification elements in university events resonates well with most respondents and could significantly boost engagement and participation.

3.2.1.3. Section C: Opinions on Features & Gamification Elements

This section seeks respondents' insights regarding the fundamental features of an event management system and the appropriate gamification elements to be integrated into UniEXP system that could enhance their event experience.

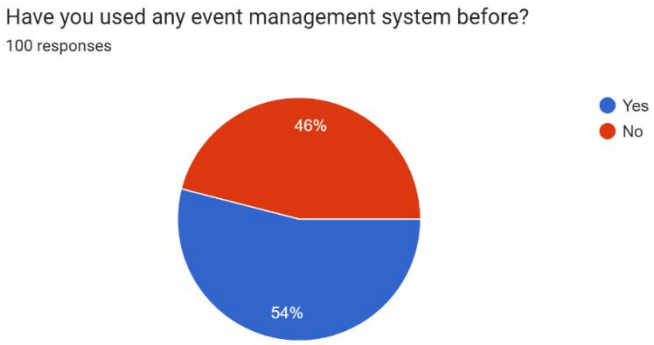


Figure 3.16. *Respondents' Prior Experience with Event Management Systems*

Based on Figure 3.16, the analysis reveals that 54% of the respondents have prior experience using an event management system, while 46% have not. These findings suggest that the UniEXP system should balance ease of use for new users with advanced features to meet the expectations of experienced users, ensuring inclusivity and broad user adoption.

Figure 3.17. *Analysis of Expected Basic Features in a Student Event Management System*

91% of respondents stated that "Event Discovery and Browsing" is a necessary feature for an event management system for a student site, as indicated by the survey findings, which are displayed in Figure 3.17. Similarly, "Event Registration and RSVP" (87%) and "Profile Management & Preferences" (84%) are prioritized, reflecting the need for easy-to-use ways to locate events, manage participation, and customize user experiences. As demonstrated by features like "QR Code-Based Attendance System" (75%) and "Push Notifications for Upcoming Events" (76%), efficient event communication and attendance tracking are crucial. However, 85% of the support went to "User registration and authentication," highlighting its vital role in ensuring system security and access. However, "Attendance Verification (GPS-based)" and "Past Events List" were less popular, with 35% and 14% support, respectively, suggesting that consumers may not regard these services as crucial.

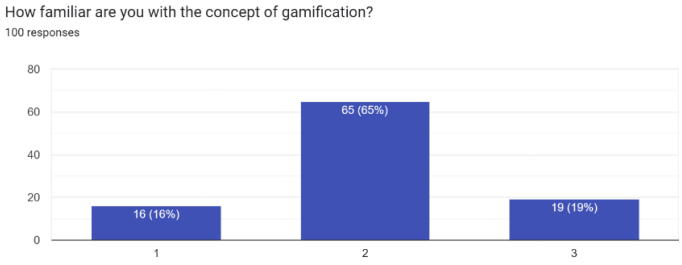


Figure 3.18. *Respondents' Familiarity with the Concept of Gamification*

Figure 3.18 shows the study of respondents' knowledge of the gamification idea. Scale 2 was chosen by the majority of respondents (65%), suggesting a modest level of familiarity

or limited comprehension of gamification. 19% of respondents, on the other hand, selected scale 3, indicating that they are familiar with the idea. On scale 1, however, 16% of respondents chose to show total unfamiliarity.

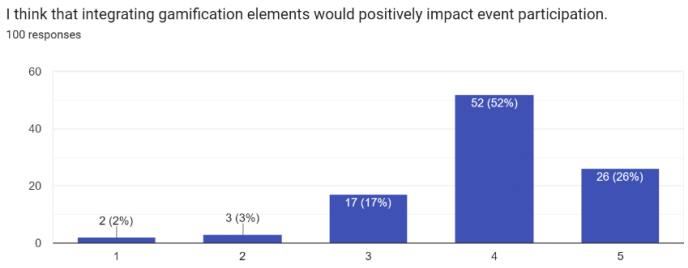


Figure 3.19. *Analysis of Responses to the Statement "Integrating Gamification Elements Would Positively Impact Event Participation"*

According to Figure 3.19, 52% of participants chose "Agree" (scale 4) in response to the statement about the benefits of gamification for increased event participation, while 26% even went one step further and chose "Strongly Agree" (scale 5). A negligible portion of the respondents (17%) claim to fall into the neutral category, suggesting that some may not have had enough exposure to or firsthand experience with gamification to form firm opinions about it. Just 3% and 2% of respondents chose "Disagree" (scale 2) and "Strongly Disagree" (scale 1), respectively, indicating that very few chose dissent.

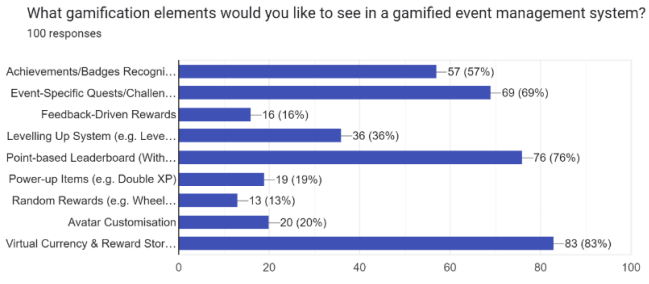


Figure 3.20. *Preferences for Gamification Elements in a Gamified Event Management System*

According to Figure 3.20, the results indicated the preference among respondents for gamification elements would be expected in a gamified event management system. 83% of

respondents favoured the inclusion of a "Virtual Currency & Reward Store" feature, highlighting their desire to trade coupons or university-issued goods and associating participation with concrete advantages. Then, 76% of respondents said they would be interested in a "Point-based Leaderboard," suggesting that users are competitive and enjoy monitoring their performance during or between faculties. Additionally, 69% gave "Event-Specific Quests/Challenges," suggesting a predilection for objective, captivating activities that align with certain events. Finally, 57% of respondents preferred "Achievement/Badges Recognition," indicating a need to give credit for individual efforts and a sense of accomplishment. These findings demonstrate how important it is to integrate competition, rewards, and personal recognition to boost user engagement and motivation.

3.2.2. Interview

An interview session was conducted with Miss Diana Tracy anak Delim, who is the assistant registrar in the administrative unit of the Centre for Student Services, via email, along with the interview consent form as shown in Appendix C. The interview aimed to understand the current practices, challenges, and expectations regarding university event management at UNIMAS. These insights assisted in shaping the system's requirements and feature set, and the detailed responses from the session are documented below.

Table 3.1. *Questions and Responses from the Session*

Bil	Questions	Response
1.	Could you share your current role and post?	She is the Senior Assistant Registrar at the Student Services Centre (PKP), handling administrative duties and coordinating events when needed.
2.	How long have you been serving in this position?	She serves in this position for 11 years.
3.	What are your primary responsibilities as an Assistant Registrar in Administration Unit?	She is helping her director in charge of the things related to the finance for organising events or upgrading the infrastructure.

4.	Could you describe the current process for managing university events, from planning to registration and follow-up? What challenges do you face?	Events are promoted via Facebook, WhatsApp, TikTok, and UNIMAS Now. Registration uses QR codes and Google Forms, but poor promotion often leads to low turnout.
5.	In your experience, why do certain events have low student participation rates?	She explained that last-minute planned events, especially those involving external factors, impacted the rates of student participation.
6.	Have you implemented any strategies in the past to improve student participation? Were they successful?	She made a concerted effort to encourage students to participate in events. In fact, she collaborated with each college head to identify students and ensure successful event outcomes.
7.	How do you think staff and students might respond to using a digital platform for event management, as opposed to the current traditional methods?	She believes that using a digital platform for event management is a good and transformational approach compared to the current traditional methods.
8.	How do you understand the term gamification?	She stated that she was uncertain with the concept of gamification. Then, I proceed with explaining briefly the concept of it.
9.	What do you think about using gamification (e.g., quests, challenges, rewards) to increase student interest in university events?	She thinks using gamification, such as quests, challenges, and rewards, is a good way to encourage people to attend university events and collect virtual currency to redeem. However, she also acknowledges that there are drawbacks for certain types of events.
10.	Imagine you are organizing a university-wide event. What kind of gamification elements would you create to ensure students stay engaged from the start to the end of the event?	She would use a points-based leaderboard and reward system (e.g., diamonds for merchandise) to maintain engagement.
11.	What kind of pre-defined tasks or challenges (e.g., top early birds, event secret phrases) would you like to associate with events to encourage student engagement?	She suggested incorporating tasks such as Q&A tasks, feedback-driven task, and picture submissions for competitions to encourage student engagement in events.
12.	How would you like rewards for students to be structured—redeemable tangible items (e.g., university merchandise) using virtual currency, intangible rewards (e.g., badges or	She chose both, specifically redeemable university merchandise via virtual currency (e.g., diamonds) and achievement badges that students may see at their profile.

	certificates), or a combination of both?	
13.	Should students be able to compete individually (e.g., within the same faculty) or in teams (e.g., compete between faculties)? How do you think this might impact participation?	She preferred students to compete individually within their respective faculties because it allows them to focus on showcasing their personal strengths. This could potentially increase participation by encouraging more students to feel confident in their individual abilities.
14.	How would you define a success or effectiveness of gamified event management system?	Success is based on student feedback through Likert-scale surveys about experience and gamification features.
15.	What concerns do you have about implemented gamification in the event management process (e.g., fairness, complexity)?	She worries students might focus less on event goals or not engage regularly.
16.	Other than that, what specific features would you find most useful for students in a system like UniEXP to help register and engage in university events?	She would find a face recognition attendance system, a scheduling calendar, and a swiper section for promoting upcoming events to be the most useful features for students in UniEXP system.
17.	Do you think structuring admin accounts to be faculty-based is an effective approach?	She thinks structuring admin accounts to be faculty-based is an effective approach.
18.	Besides faculty-based admin accounts, do you think there's a need for a general or centralized admin account for non-faculty-specific event management?	For simplicity, she stated there is no need for a general admin account for event management.
19.	Would you prefer the system to allow cross-faculty admins for collaborative events?	She stated there is no needed for cross-faculty admins for collaborative events for the sake of simplicity.
20.	What features (e.g., event data analysis – number of attendees, feedback summaries) would you expect from an administrative site to make system more efficient?	She would expect features such as a feedback form designed to evaluate whether students achieved the event objectives and areas for improvement, which could use a Likert scale format. Additionally, she would anticipate tools to track the number of attendees or participants.

3.3. Feature-Driven Development (FDD)

3.3.1. Develop an Overall Model

A modules diagram is depicted in Figure 3.21 to represent domain object modelling for the system.

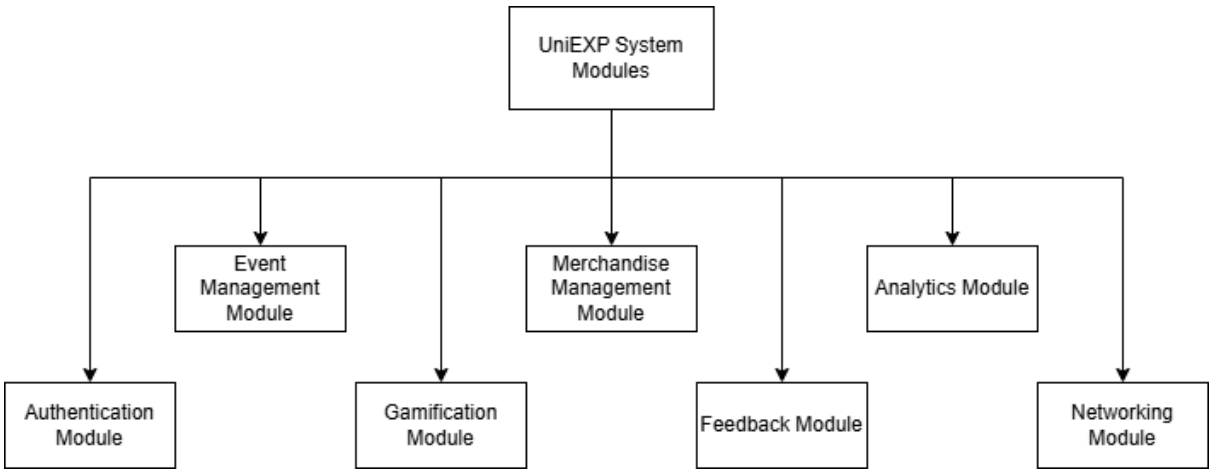


Figure 3.21. *UniEXP's System Modules Diagram*

Additionally, several activity diagrams are created to depict the system's workflow or sequence of operations, showing how tasks and processes move through the FDD approach from beginning to conclusion (Elivera & Palaoag, 2020).

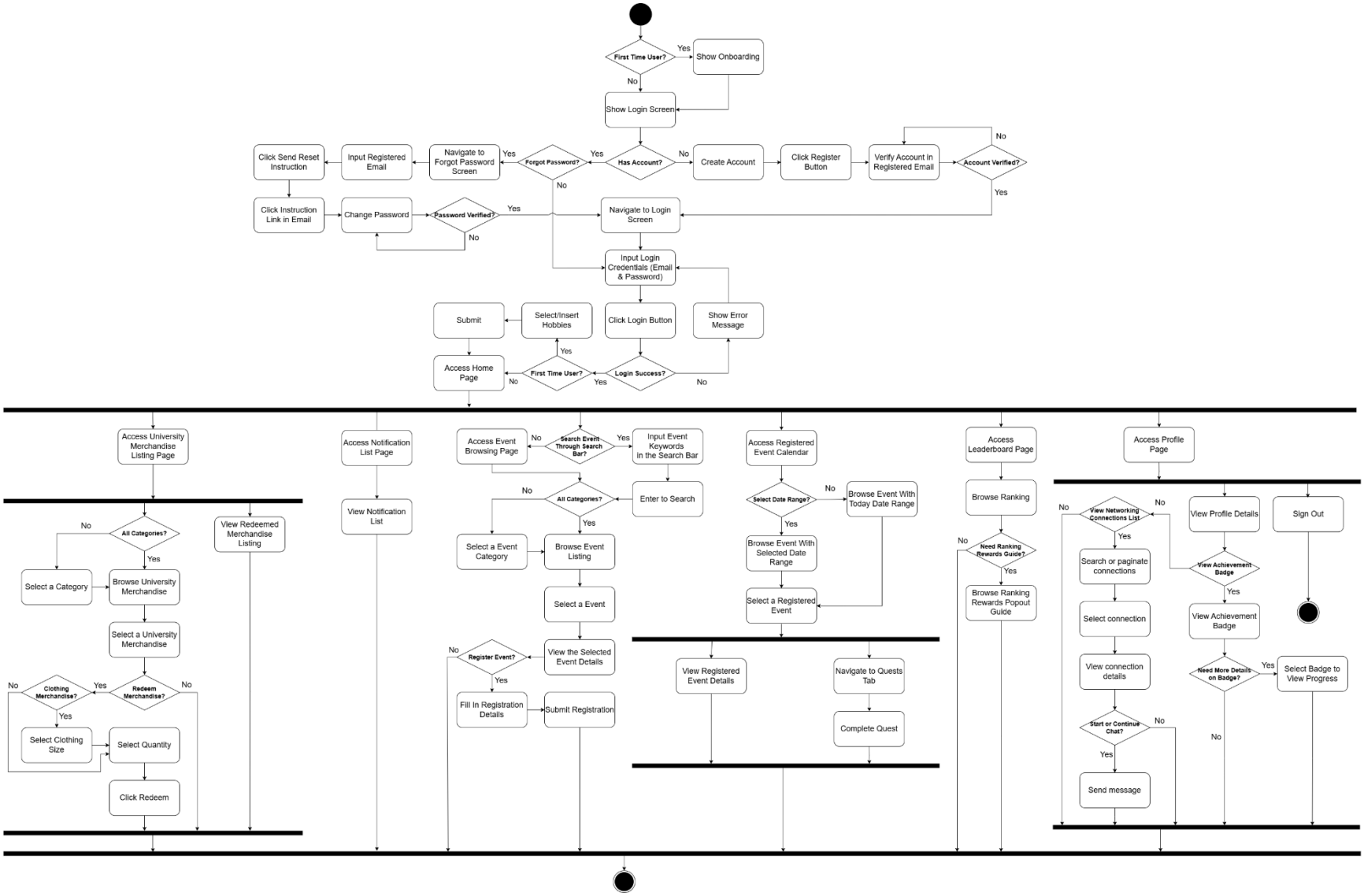


Figure 3.22. *Student – Activity Diagram*

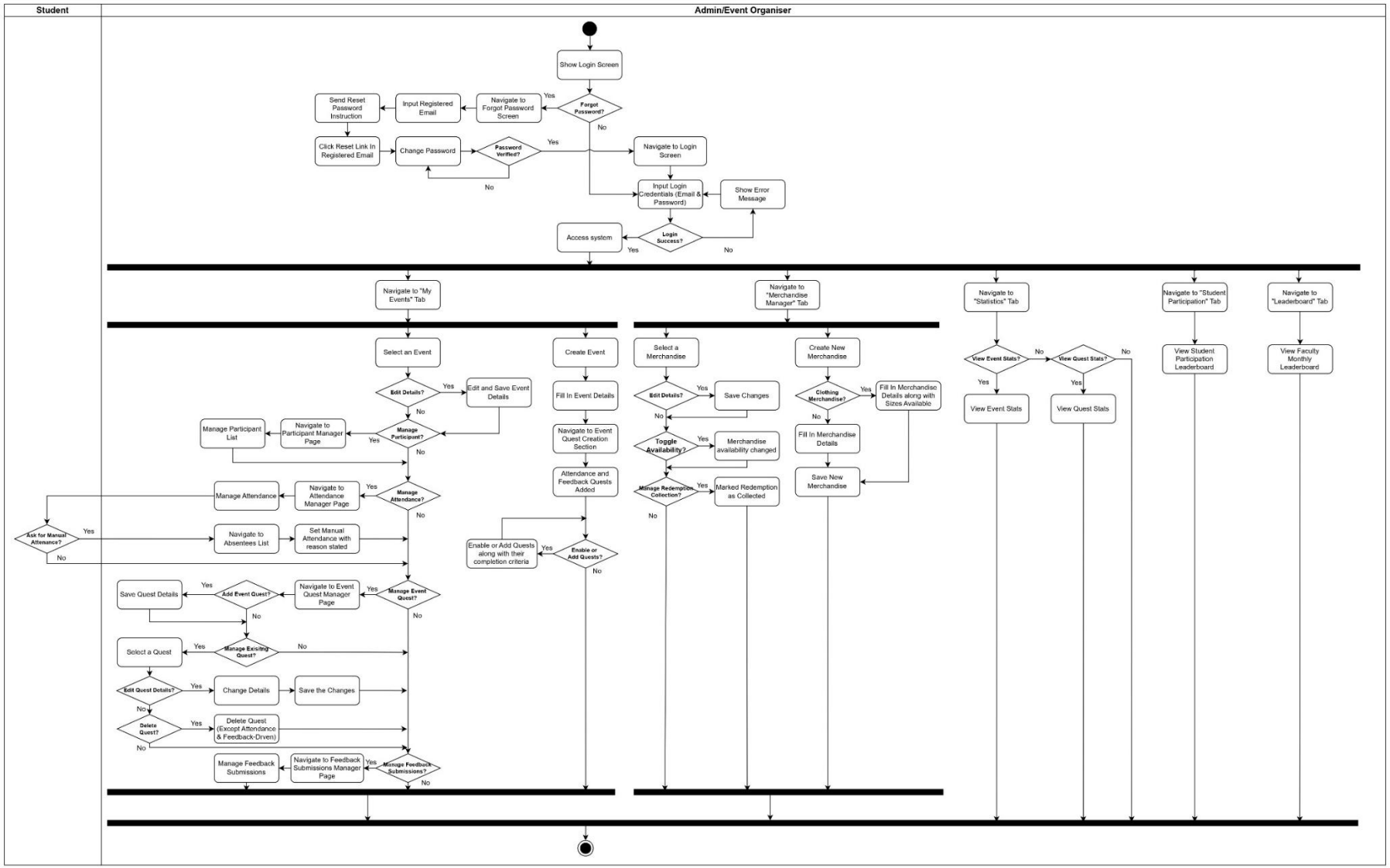


Figure 3.23. *Admin-Oriented – Activity Diagram*

3.3.2. Build a Features List

From the first phase of the FDD methodology, a features list from the system module is concluded to understand better the project scope for features list development (Elivera & Palaoag, 2020).

Table 3.2. *Features List for UniEXP System*

System Module	Bil.	Features for Each Module
Authentication Module	1.	Both students and admins/event organisers can log in.
	2.	Both students and admins/event organisers can change their password.
	3.	Students can sign up.
Event Management Module	4.	Students can browse and register for events.
	5.	Students can manage registered events through the agenda.
	6.	Admins/event organisers can add new events.
	7.	Admins/event organisers can manage existing events.
	8.	Admins/event organisers can manage the list of participants.
	9.	Admins/event organisers can track and validate event attendance.
Gamification Module	10.	Students can see their ranking in the faculty leaderboard.
	11.	Students can complete quests tied to events.
	12.	Students can view and earn the achievement badge.
	13.	Admins/event organisers can add event-specific quests.
	14.	Admins/event organisers can manage quests.
	15.	Admins/event organisers can view current month leaderboard.
Merchandise Management Module	16.	Students can redeem merchandise using virtual currency (diamonds).
	17.	Admins/event organisers can add new merchandise.
	18.	Admins/event organisers can manage merchandise.
Feedback Module	19.	Admins/event organisers can manage feedback submitted by students.

Analytics Module	20.	Admins/event organisers can view overall student participation leaderboard.
	21.	Admins/event organisers can view event and quest statistics.
Networking Module	22.	Students can manage networking connections and messaging them.
Notification Module	23.	Students can view notifications list.

3.3.3. Plan by Feature

In this phase, the features in the list are analysed and evaluated by categorized them into several priority: high, medium, low as shown Table 3.3.

Table 3.3. *Categorisation of Features by Priority*

Bil.	Priority	Feature
1.	High	• Both students and admins/event organisers can log in. • Both students and admins/event organisers can change their password. • Students can sign up. • Students can browse and register for events. • Students can see their ranking in the faculty leaderboard. • Students can manage registered events through the agenda. • Students can complete quests tied to events.
2.	Medium	• Students can redeem merchandise using virtual currency (diamonds). • Students can view the earned achievement badge. • Admins/event organisers can add new events. • Admins/event organisers can manage existing events. • Admins/event organisers can design event-specific quests. • Admins/event organisers can manage quests.
3.	Low	• Admins/event organisers can manage the list of participants. • Admins/event organisers can track and validate event attendance. • Admins/event organisers can manage feedback submitted by students. • Admins/event organisers can add new merchandise.

		• Admins/event organisers can manage merchandise. • Admins/event organisers can view event and quest related statistics. • Admins/event organisers can view current month leaderboard. • Admins/event organisers can view student participation leaderboard. • Students can manage networking connections and messaging them. • Students can view notifications list.

Moreover, a Gantt chart to assign the schedule and track for the activities was depicted (Elivera & Palaoag, 2020).

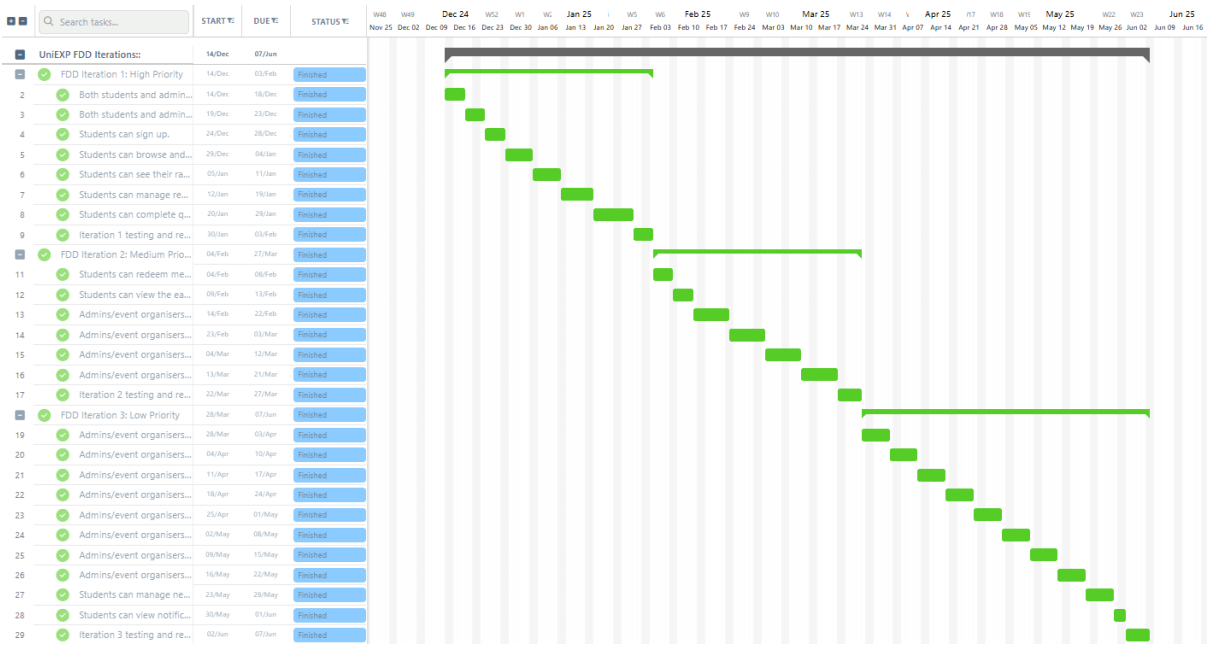


Figure 3.24. *Gantt Chart – FDD Features Development Iteration Planning*

3.3.4. Design by Feature & Build by Feature (Iterative Phases)

These two phases refer to the construction phase which are iterative for each feature set (Elivera & Palaoag, 2020) based on the priority. In Design by Feature phase, a use case diagram is depicted to show how different parts of the system will interact with users or other systems and a list of sequence diagrams for the features in the use case diagram were designed (Elivera & Palaoag, 2020) to show the step-by-step interactions between objects or components in a system. Furthermore, a class diagram of the system is designed to provide a blueprint for system implementation by modelling the static structure of a system by representing its classes,

their attributes, methods, and relationships (What is FDD in Agile?, 2021). Moreover, wireframe of the system is designed in this phase (Elivera & Palaoag, 2020). Lastly, the Build by Feature phase requires to develop the system and implement unit and integration testing. The system development iteration continues with another feature set.

3.3.4.1. Use Case Diagram

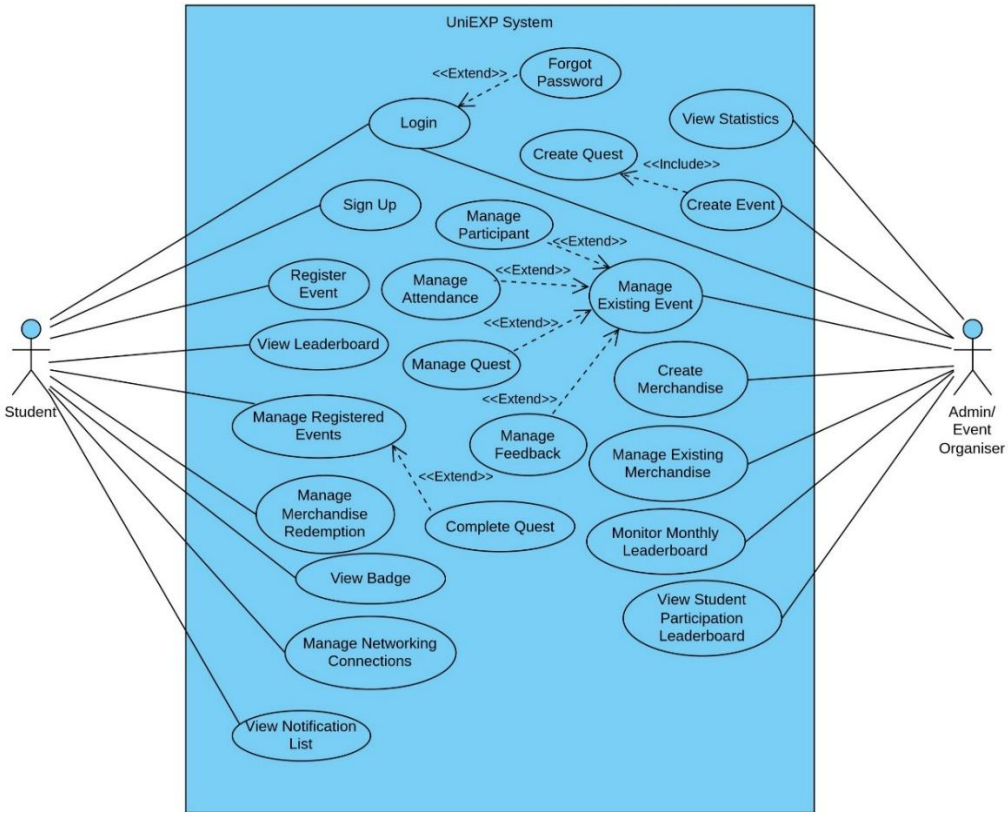


Figure 3.25. *Use Case Diagram for UniEXP System*

3.3.4.2. Use Case Specification

The completed use case specifications for UniEXP system according to the use case diagram above are shown in Appendix D.

3.3.4.3. Sequence Diagrams

A list of sequence diagrams selected based on the use cases are shown in Appendix E.

3.3.4.4. Class Diagram

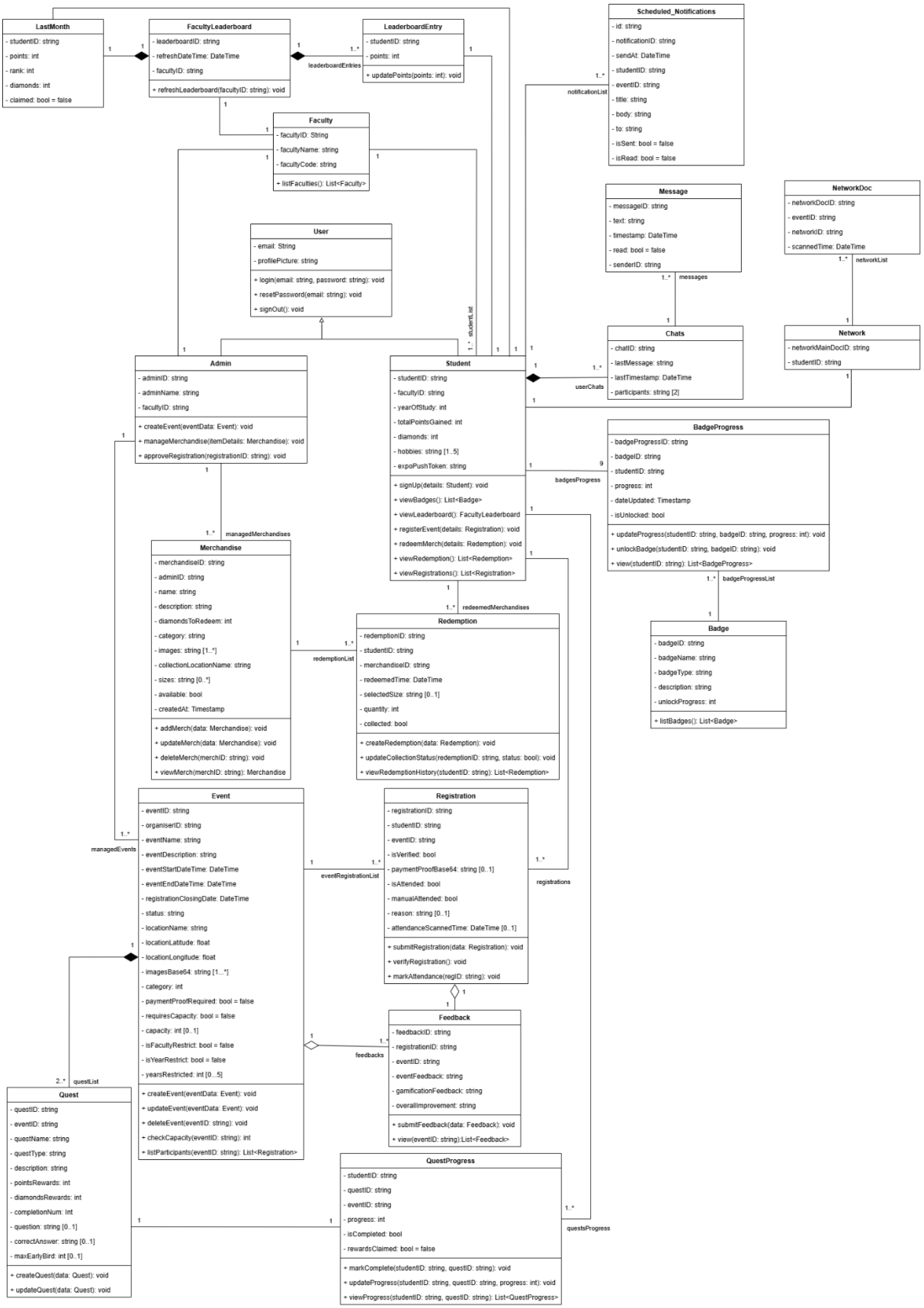


Figure 3.26. *UniEXP's Class Diagram*

3.3.4.5. Wireframes

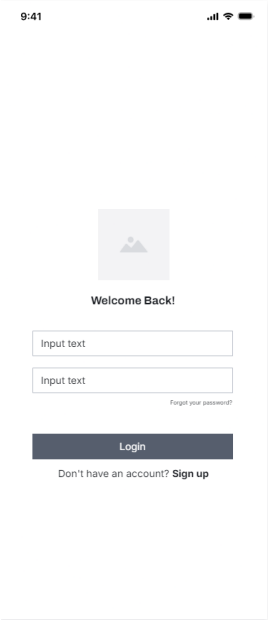


Figure 3.27. *Student - Login Wireframe*

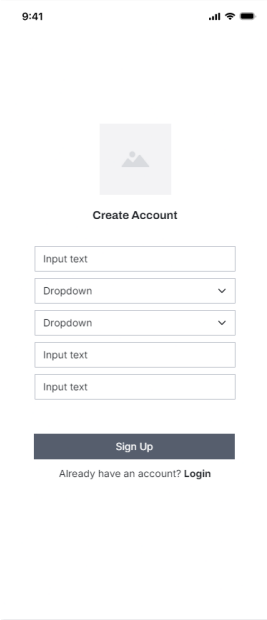


Figure 3.28. *Student - Sign Up Wireframe*

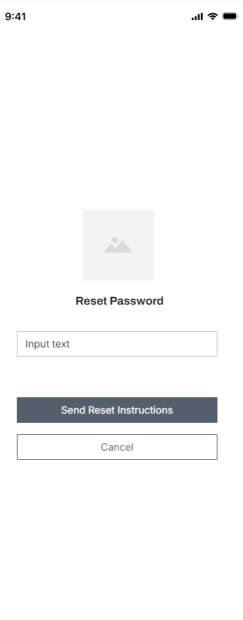


Figure 3.29. *Student - Forgot Password Wireframe*

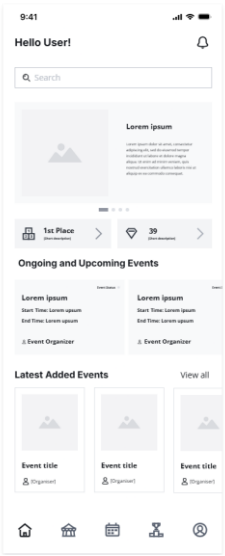


Figure 3.30. *Student - Home Page Wireframe*

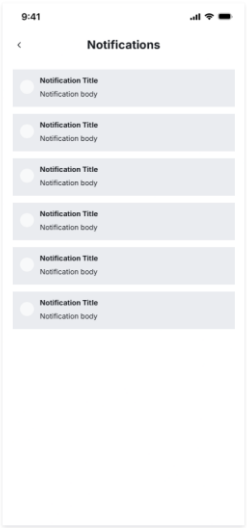


Figure 3.31. *Student - Notification List Page Wireframe*

From above, Figure 3.27 shows student's logic page wireframe while Figure 3.28 shows student's sign-up page wireframe. Besides, Figure 3.29 shows student's forgot password wireframe while Figure 3.30 shows student's home page wireframe. Besides, Figure 3.31 shows student's notifications list.

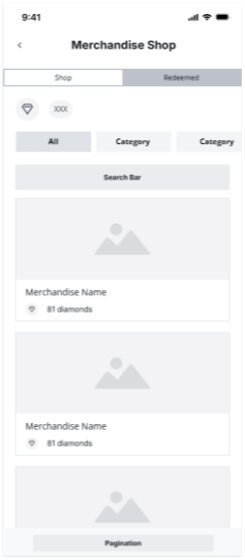


Figure 3.32. *Student – Merchandise Listing Wireframe*

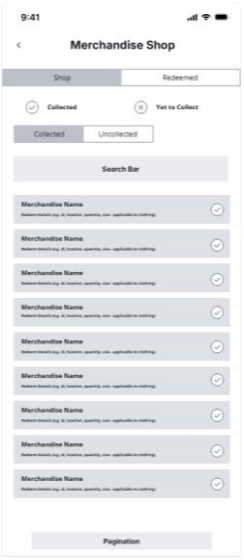


Figure 3.33. *Student – Redeemed Merchandise Listing Wireframe*

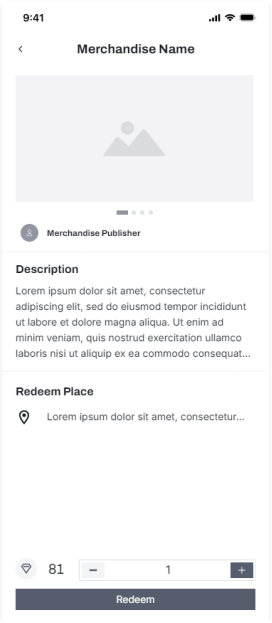


Figure 3.34. *Student - Merchandise Details Wireframe*

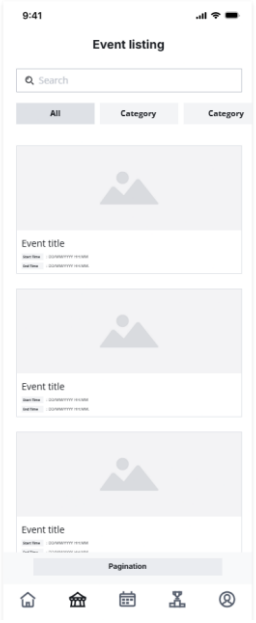


Figure 3.35. *Student - Event Listing Wireframe*

Figure 3.32 shows the student's merchandise listing page wireframe while Figure 3.33 shows the student's redeemed merchandise listing page wireframe. Figure 3.34 shows the merchandise details page wireframe. Figure 3.35 shows the student's event listing page wireframe.

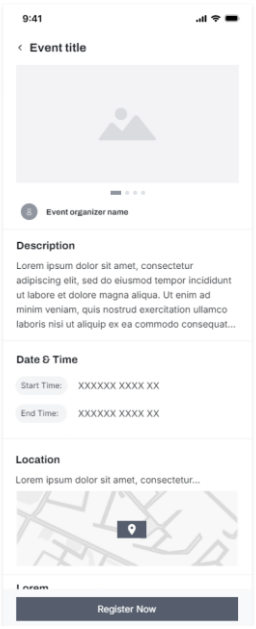


Figure 3.36. *Student - Event Details Wireframe*

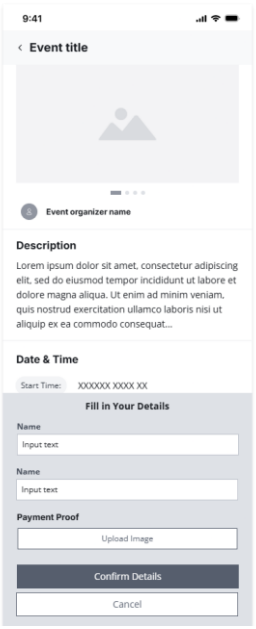


Figure 3.37. *Student - Event Registration Form Wireframe*

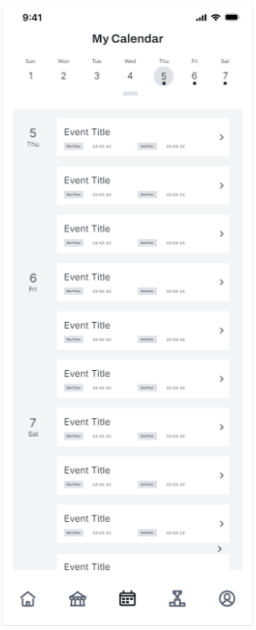


Figure 3.38. *Student - Registered Events Calendar Wireframe*

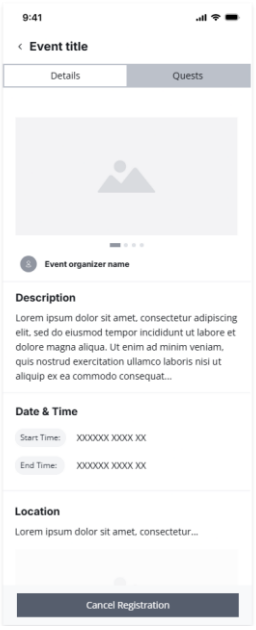


Figure 3.39. *Student - Registered Event Details Wireframe*

Figure 3.36 shows the event details page wireframe and Figure 3.37 shows the event registration form pop up modal wireframe. Besides, Figure 3.38 shows the student's registered events calendar wireframe while Figure 3.39 shows the registered events details wireframe with registration cancellation option.

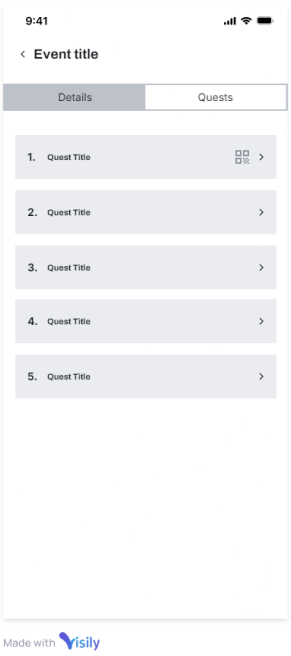


Figure 3.40. *Student - Event Quest List Wireframe*

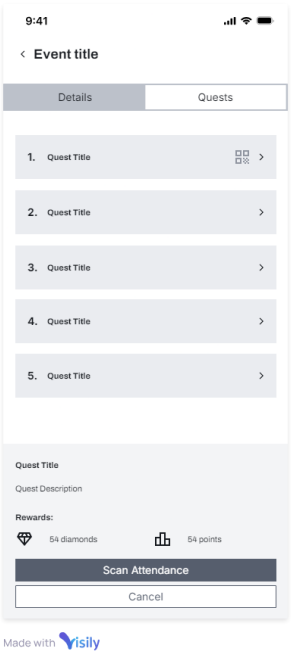


Figure 3.41. *Student - Attendance-Based Quest Wireframe*

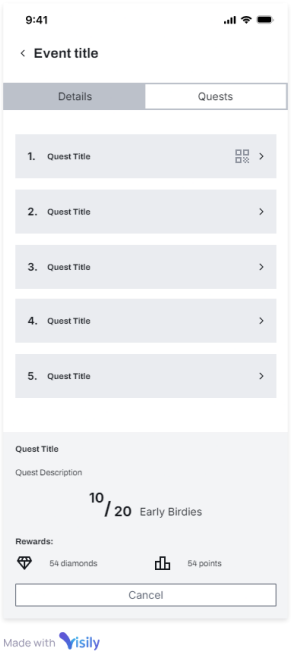


Figure 3.42. *Student - Early Bird Attendees' Quest Wireframe*

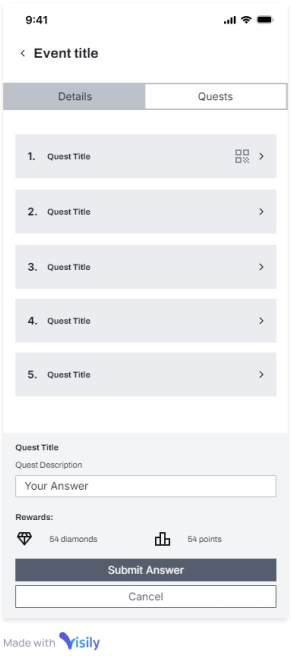


Figure 3.43. *Student - Q&A Quest Wireframe*

Figure 3.40 shows the event quest list page wireframe and Figure 3.41 shows the attendance-based quest pop up modal wireframe. Figures 3.42 and 3.43 show the early bird attendees' quest and the question and answer (Q&A) pop up modal wireframes, respectively.

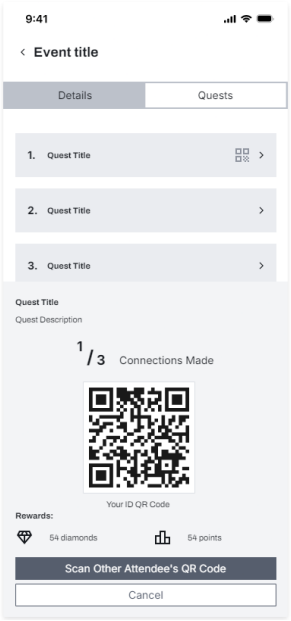


Figure 3.44. *Student – Networking Quest Wireframe*

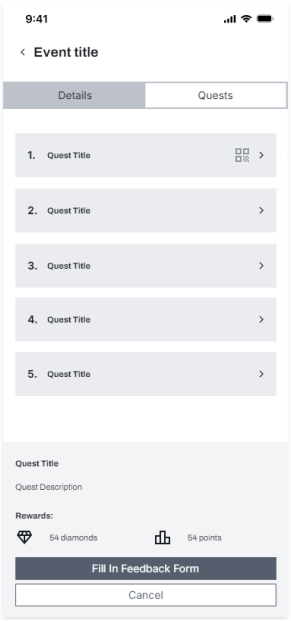


Figure 3.45. *Student - Feedback-Driven Quest Wireframe*

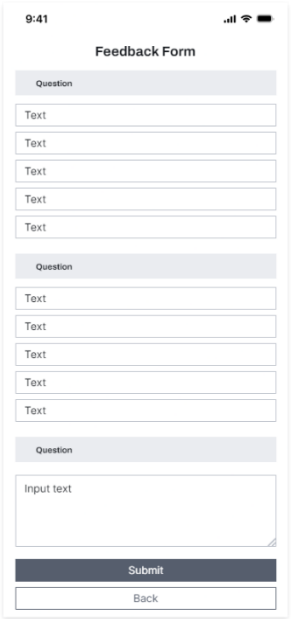


Figure 3.46. *Student - Feedback Form Wireframe*

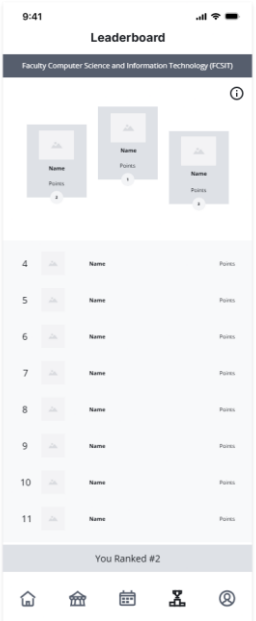


Figure 3.47. *Student - Leaderboard Wireframe*

Besides, Figures 3.44 and 3.45 show the networking quest and the feedback-driven quest pop up modal wireframes, respectively. Figure 3.46 shows the feedback form wireframe for feedback driven quest and Figure 3.47 shows the student's faculty leaderboard wireframe.

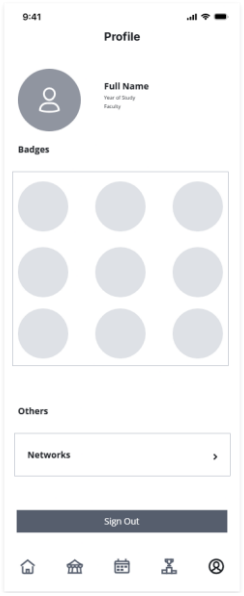


Figure 3.48. *Student - Profile Wireframe*

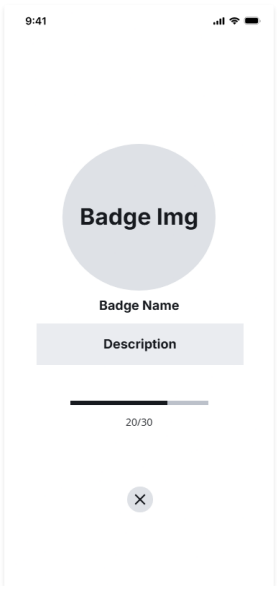


Figure 3.49. *Student - Badge Wireframe*

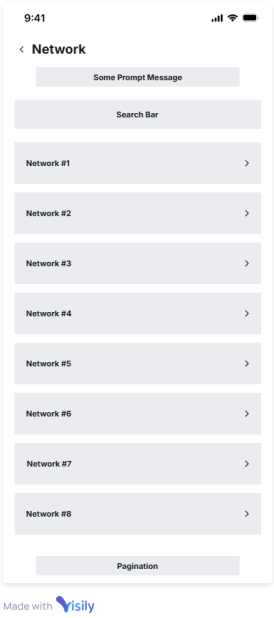


Figure 3.50. *Student - Network List Wireframe*

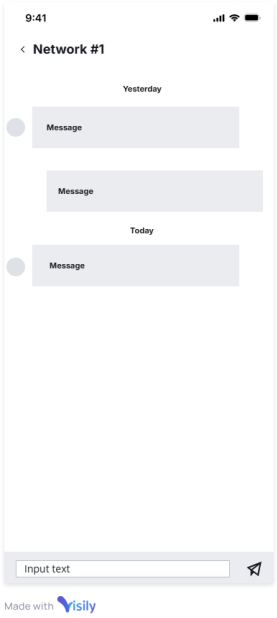


Figure 3.51. *Student - Messaging Wireframe*

Lastly, Figure 3.48 shows the student's profile page wireframe. By clicking one of the badges in the profile screen, the badge details will be displayed as shown in Figure 3.49. Moreover, students can navigate to the networks list, as seen in Figure 3.50, as they made through the networking quest by clicking the "Network" card in profile screen. The students can navigate to messaging screen (Figure 3.51) by selecting the network they want to chat with.

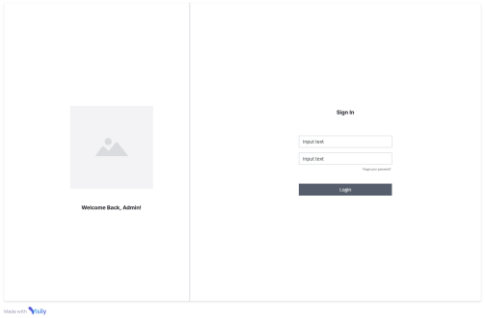


Figure 3.52. *Admin - Login Wireframe*

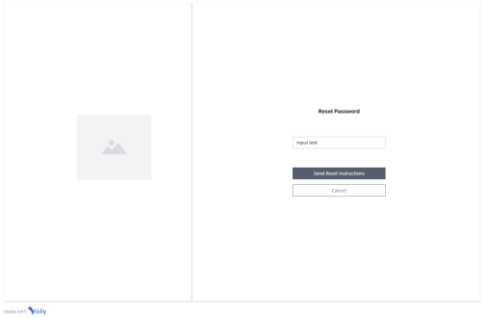


Figure 3.53. *Admin - Forgot Password Wireframe*

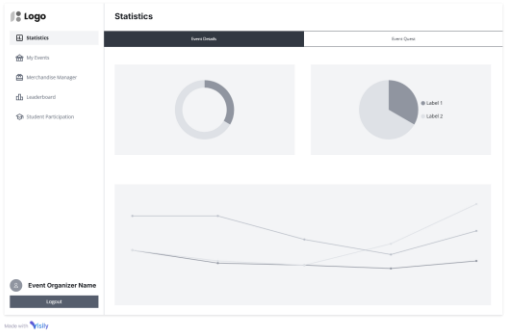


Figure 3.54. *Admin - Statistics (Event) Wireframe*

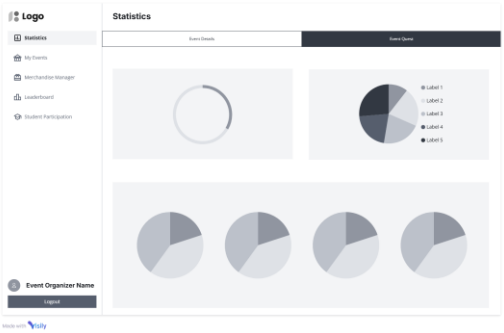


Figure 3.55. *Admin - Statistics (Quest) Wireframe*

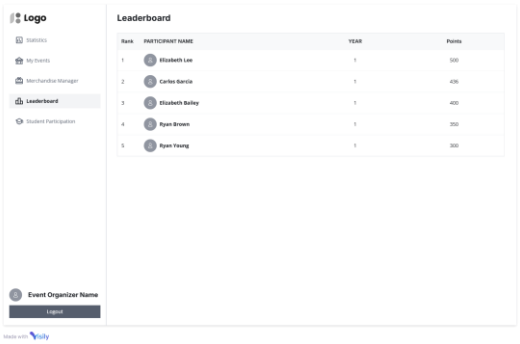


Figure 3.56. *Admin - Faculty Monthly Leaderboard Wireframe*

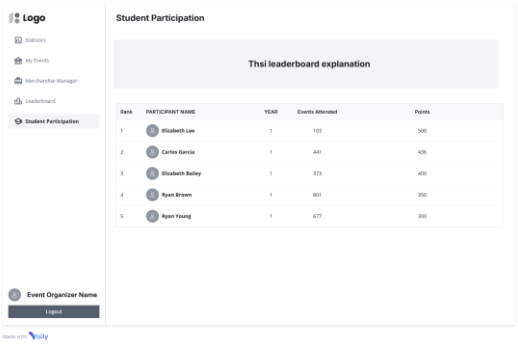


Figure 3.57. *Admin - Student Participation Leaderboard Wireframe*

On the other hand, Figure 3.52 shows the admin's login page wireframe while Figure 3.53 shows the admin's forgot password page wireframe. After logged in, the system will display the event and quest-related statistics as shown in Figure 3.54 and Figure 3.55. Through the side navigation, admin can navigate to faculty leaderboard which refreshes monthly as shown in Figure 3.56 and can navigate to student participation leaderboard as shown in Figure 3.57.

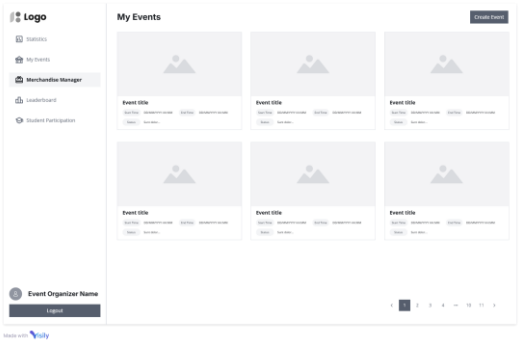


Figure 3.58. *Admin - Event Listing Wireframe*

Figure 3.59. *Admin - Event Creation Wireframe*

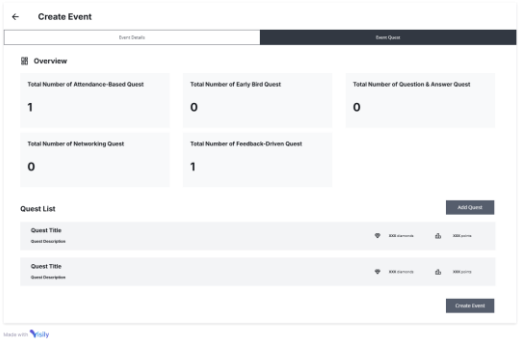


Figure 3.60. *Admin - Event Quest Creation Wireframe*

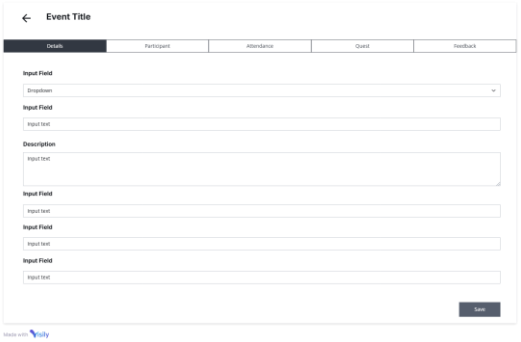


Figure 3.61. *Admin - Event Details Manager Wireframe*

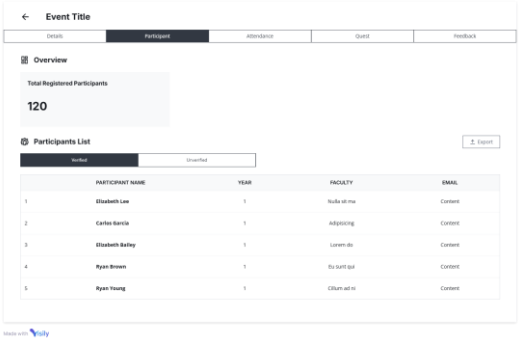


Figure 3.62. *Admin - Event Participant (Verified) Manager Wireframe*

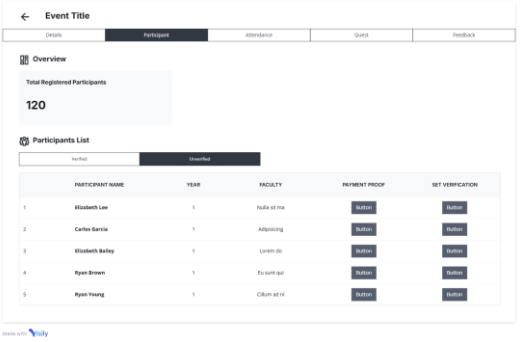


Figure 3.63. *Admin - Event Participant (Unverified) Manager Wireframe*

Besides, Figure 3.58 and 3.59 shows the admin's event listing wireframe and event creation wireframe, respectively. Moreover, Figure 3.60 shows admin's event quest creation wireframe while Figure 3.61 shows the admin's event details manager wireframe. Figures 3.62 and 3.63 shows the admin's event participant manager wireframe for both verified and unverified participants, respectively.

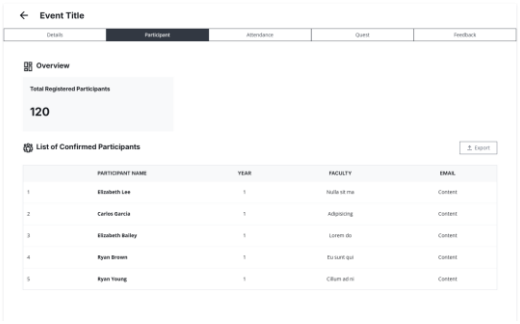


Figure 3.64. *Admin - Event Participant (No Payment Proof Required) Manager Wireframe*

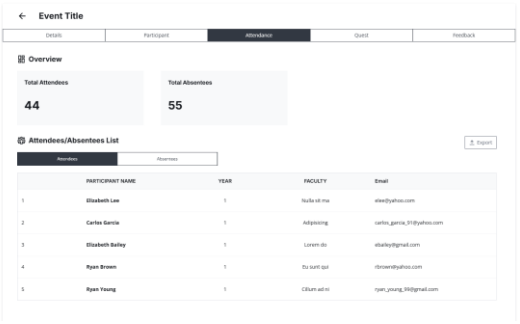


Figure 3.65. *Admin - Event Attendance (Attendees) Manager Wireframe*

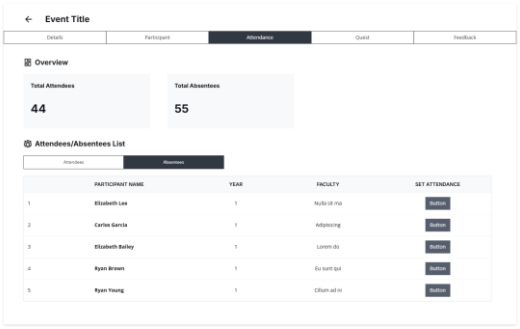


Figure 3.66. *Admin - Event Attendance (Absentees) Manager Wireframe*

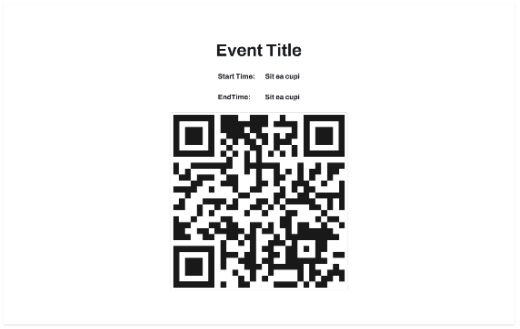


Figure 3.67. *Admin - Event Attendance QR Code Wireframe*

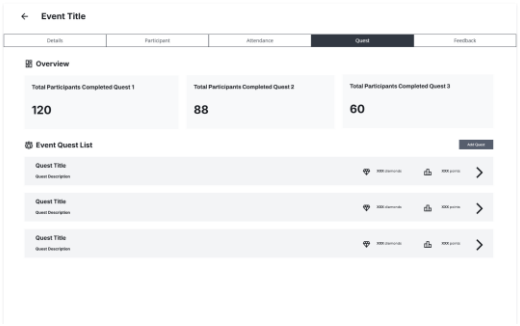


Figure 3.68. *Admin - Event Quest Manager Wireframe*

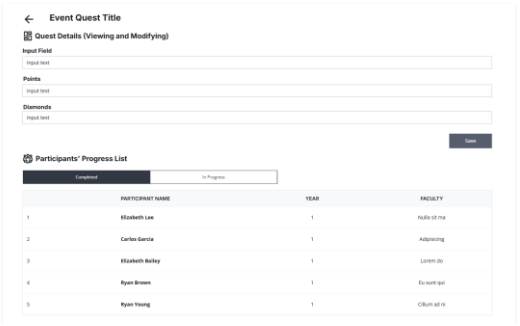


Figure 3.69. *Admin - Event Quest (Completed) Manager Wireframe*

Figure 3.64 shows the admin's event participant manager wireframe for those events do not require payment proof. Besides, Figure 3.65 and 3.66 shows the admin's event attendance wireframe for both attendees and absentees, respectively. As seen from the Figure 3.67, it shows the event attendance QR code wireframe. Figure 3.68 shows the admin's event

quest manager wireframe while Figure 3.69 shows the admin's event quest manager wireframe for completed.

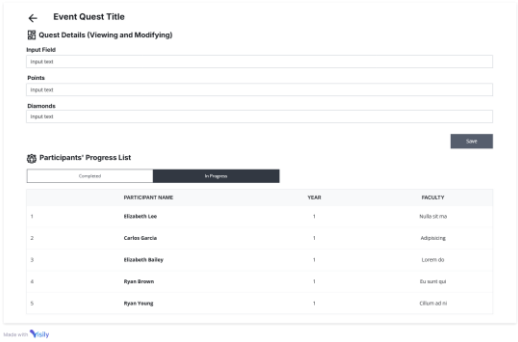


Figure 3.70. *Admin - Event Quest (In Progress) Manager Wireframe*

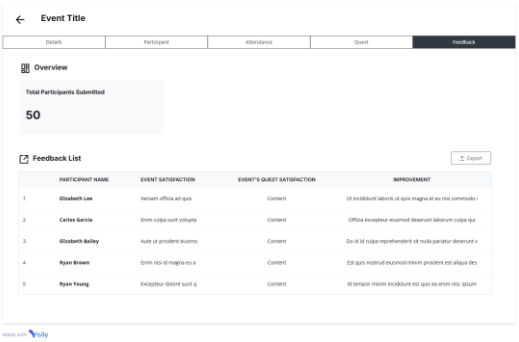


Figure 3.71. *Admin - Feedback Submission Manager Wireframe*

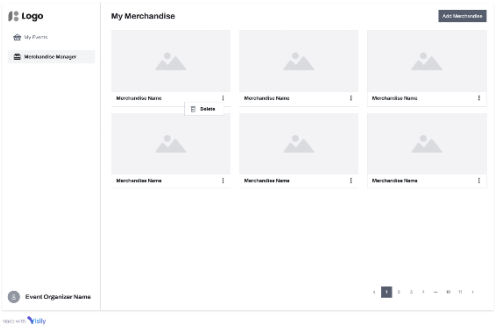


Figure 3.72. *Admin - Merchandise Listing Wireframe*

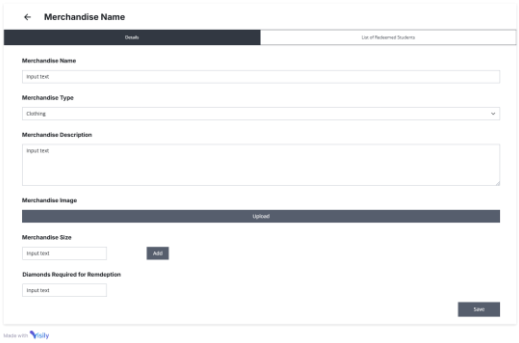


Figure 3.73. *Admin - Merchandise Details Manager Wireframe*

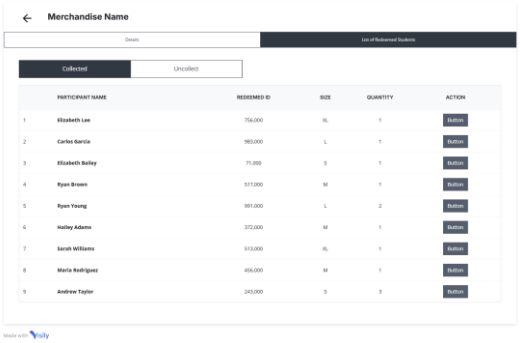


Figure 3.74. *Admin - Merchandise Redemption Student List*

Figure 3.75. *Admin - Merchandise Creation Wireframe*

Besides, Figure 3.70 shows the admin's event quest manager wireframe for in progress. As seen from the Figure 3.71, it shows the admin's feedback submission manager wireframe. Meanwhile, Figure 3.72 shows the admin's merchandise listing wireframe where those

merchandises managed by them. Figure 3.73 shows the admin's merchandise details manager wireframe while Figure 3.74 shows the admin's merchandise redemption student list wireframe. Lastly, Figure 3.75 shows the admin's merchandise creation wireframe.

3.4. Summary

This chapter outlines the Agile methodology framework to develop the UniEXP system: Feature-Driven Development (FDD). The requirements collecting and analysis phase, which used interview and questionnaires to get user insights and system expectations, is covered in full at the beginning of the chapter. The main conclusions drawn from the 100 participants at UNIMAS emphasized the difficulties with the current event management procedure as well as the possible advantages of including gamification components. The five steps of the FDD technique were as follows: (1) develop an overall model; (2) build a features list; (3) plan by feature; (4) design by feature; and (5) build by feature. System modules diagram, activity diagrams, use case diagrams, sequence diagrams, system wireframes, and class diagrams were among the visual aids used in each phase to support system functionality and design.

Chapter 4. IMPLEMENTATION

4.1. Introduction

This chapter describes the implementation of the proposed system, UniEXP, for student and faculty-based event administrators. It details the configuration of components, role-based access, workflows, and feature modules used to build the system according to the design specifications. The intended functionality of each system component and its role in driving the overall goals is described.

4.2. Installation and Configuration of System's Components

During the proposed system installation and configuration, the development environments and system components were configured to function together seamlessly.

4.2.1. Visual Studio Code (VS Code)

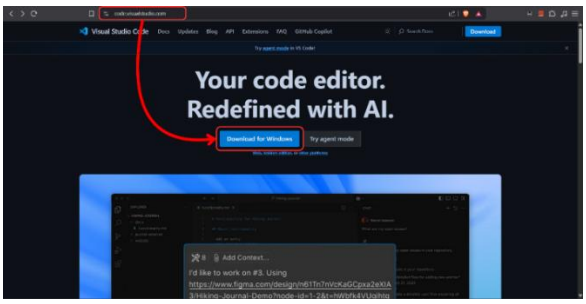


Figure 4.1. *Visual Studio Code Installation Website*

In this project, both web and mobile system development were conducted using VS Code as the primary integrated development environment (IDE). Figure 4.1 shows that VS Code was downloaded from the official website (https://code.visualstudio.com/) and installed on the development machine. Installation was straightforward; it followed their setup wizard, and the setup took no further configuration. VS Code was used to open project folders, manage files, and run terminal commands. Besides, its built-in Git integration was also used to handle version control, including committing changes and synchronising with the project's GitHub repository.

4.2.2. Node.js and npm

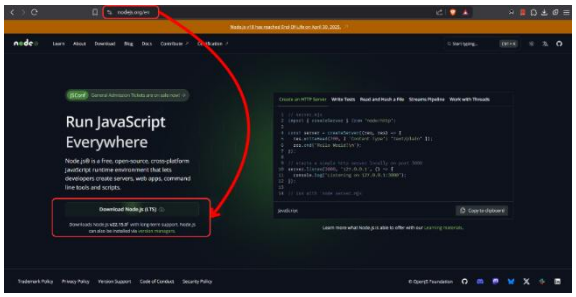


Figure 4.2. *Node.js and npm Installation Website*

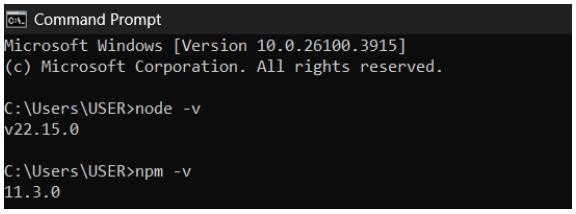


Figure 4.3. *Node.js and npm Version Checking*

Node.js and npm were utilised in the development of the web and mobile systems, which executed JavaScript code outside of the browser and managed the installation of the various libraries and dependencies needed, respectively. Node.js was installed from the official website (https://nodejs.org/) with version 22.15.0, and npm (Node Package Manager) was included automatically as shown in Figure 4.2. Next, the versions were checked using the command line (refer to Figure 4.3) *node -v* and *npm -v* so the project runs with compatible executable versions and does not throw errors.

4.2.3. Firebase Configuration

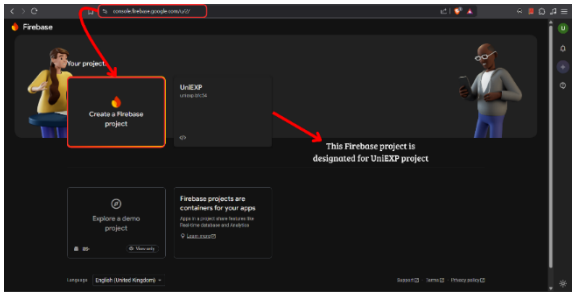


Figure 4.4. *Firebase Project Creation Instruction*

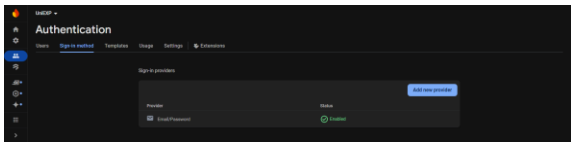


Figure 4.5. *Firebase Authentication Setup*

Firebase was used as the backend service, providing both the database (Firestore) and authentication (Firebase Auth) functionalities. Figure 4.5 shows a Firebase project was created via the Firebase Console (https://console.firebase.google.com/), and the following services were configured:

- **Authentication:** Email and password sign-in method was enabled (see figure 4.5) to allow users to register and log in.
- **Firestore Database:** Collections were created for storing users, events, attendance records, and leaderboard data.
- **Project settings:** Firebase configuration keys were generated from the Firebase Console for integration into both the web and mobile applications.

Besides, the Firebase JavaScript SDK was required to be installed in both system's environments using the command *npm install firebase*.

4.2.4. React Web System

The admin web system was developed using React, set up with Vite for faster build times and hot module replacement. The following steps were carried out:

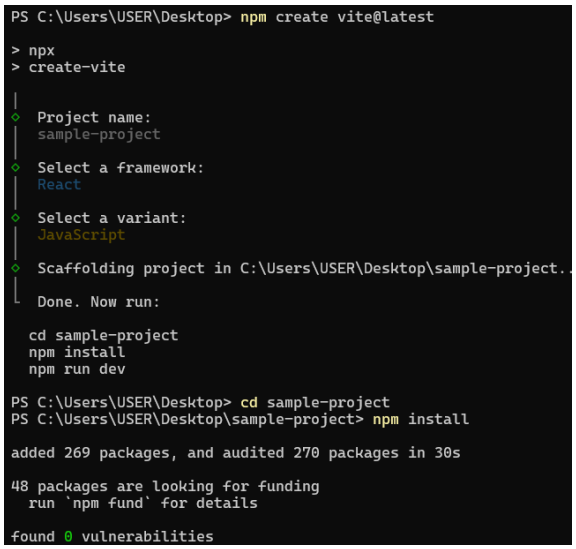


Figure 4.6. *React.js (Vite) Project Creation Setup*

Figure 4.7. *Libraries to be Installed in React Project*

- **Project setup:** The project was initialised using the command *npm create vite@latest* via the command prompt (refer to Figure 4.6).
- **Library installation:** The libraries shown in Figure 4.7 were installed for project run.

- **Configuration:** The Vite development server was started with the command *npm run dev* to serve the application locally.

4.2.5. React Native and Expo Mobile App

The student mobile application was developed using React Native with the Expo framework, which simplified the build and testing process:

- **Environment setup:** Expo CLI was installed globally using *npm install -g expo-cli*.

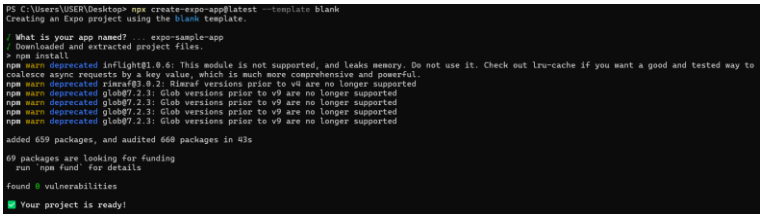


Figure 4.8. *React Native (Expo) Project Setup*

- **Project initialisation:** The mobile project was created using *npx create-expo-app@latest --template blank* (refer to Figure 4.8).

Figure 4.9. *Libraries to be Installed in React Native Expo Project*

- **Library installation:** The libraries shown in Figure 4.9 were installed for project run.

Figure 4.10. *Expo Go App on Mobile Device*

- **Testing:** Expo Go was installed on physical devices to enable live testing as seen in Figure 4.11. The app was started using *npx expo start*, which generated a QR code to be scanned by Expo Go for real-time preview on mobile devices.

4.3. Introduction to Role-Based Access

The proposed system, UniEXP, is designed to serve multiple types of user groups, specifically UNIMAS students and faculty-based event administrators. Each user type has distinct roles, responsibilities, and system privileges that allow them to perform specific tasks through their designated system portals. Role-based access control ensures that each user type can only access the features relevant to their responsibilities, improving both security and usability. The table below provides a clear comparison of the roles and functions available to each user type within the UniEXP system.

Table 4.1. *Roles for Both Student and Faculty-based Event Administrator in the Proposed System*

Student	Faculty-based Event Administrator
• Sign in to the student portal	• Sign in to the admin portal
• Reset password via email	• Reset password via email
• Sign up for an account	• Add new events
• Browse and register for events	• Manage existing events
• View leaderboard rankings	• Add event-specific quests
• Manage registered events through agenda	• Manage existing event quests
• Complete event quests	• Manage list of participants
• Redeem merchandise	• Track and validate event attendance

• Manage achievement badges	• Manage feedback submissions
• Message with networks made	• Add and manage merchandise
• View notification list	• Access event and quest statistics
	• Monitor faculty monthly leaderboard
	• View student participation leaderboard

4.4. Workflow for System's Application

This section details how the UniEXP mobile and web system operates through designed workflows to provide efficient user interaction.

4.4.1. Student Mobile Application Workflow

The system began when first-time users encountered an onboarding screen that displayed the application's features after launching it. The application directs users with existing accounts to the login interface upon launch. To complete email verification, users without an account must fill out the registration form. Users must complete verification before accessing the system with their email address and password. If users forget their password, they can reset it by requesting a password recovery link via email. Users gain entry to the home page after completing successful authentication. Then, users can browse and register for events, view registered events through the agenda, check leaderboard rankings, complete event quests, redeem merchandise using diamonds, manage achievement badges, interact with the connections they have made, and view notification list.

4.4.2. Event Organiser/Administrator Web System Workflow

Event administrators log into the web system using their registered credentials to access it, and the system offers a password reset feature. The dashboard becomes available after event administrators enter their login credentials to access a range of management functions,

including event information display and detail updates and participant list control and attendance tracking and absentee handling alongside event-based quest monitoring and feedback review for improved data collection. Through the system, users can manage merchandise details add items, modify item availability, and track item redemptions for precise documentation.

4.5. Modules for System's Application

This section outlines the key system modules developed for both the UniEXP mobile application (for students) and the web application (for admin or event organisers). These modules were designed to address the core functional requirements of each user group while maintaining a focus on usability, maintainability, and alignment with the project goals.

4.5.1. Student Modules

4.5.1.1. Authentication

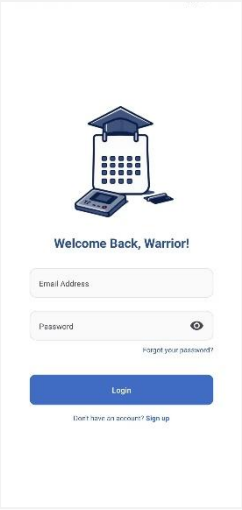


Figure 4.11. *Student's Sign in Screen*

From Figure 4.11, the sign-in screen serves as the system's entry point, allowing users to log in using their email and password. When a user attempts to sign in, the system performs validations, as shown in Appendix Figure F.1, which first verifies that both fields are filled. If

either is empty, an appropriate error message is displayed. Upon submitting valid credentials, the system attempts to authenticate the user via Firebase Authentication. If authentication is successful, the system performs several checks.

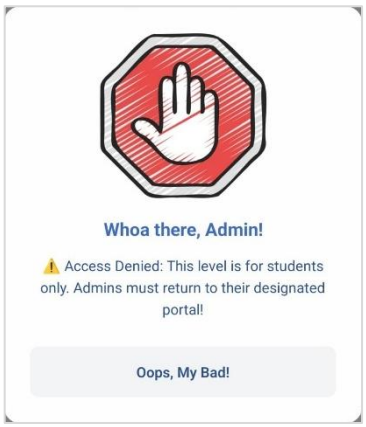


Figure.4.12. *Admin Access Restricted Modal*

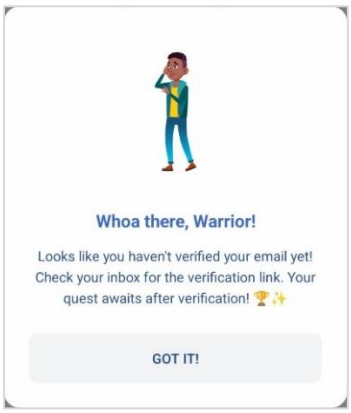


Figure 4.13. *Email Unverified Prompt Message*

First, it determines whether the account belongs to an admin by querying the "admin" collection using the user's unique ID. Suppose the user is identified as an admin. In this case, a dedicated modal is displayed, as shown in Figure 4.12, informing the user that they are not eligible to sign in through the student interface. If the user is not an admin, the system then checks whether their email address has been verified. If the email is unverified, a modal is triggered, prompting the user to verify their email before proceeding, as shown in Figure 4.13.

If the user passes both checks, the system retrieves the user document from the "user" collection using their UID, as shown in Appendix Figure F.2. It then stores the student ID and faculty ID into local storage to be used across the app. To ensure the most up-to-date session state, the system signs the user out and then signs them in again with the same credentials.

In the event of authentication errors, such as a non-existent account, incorrect password, invalid email format, or too many failed login attempts, the system displays appropriate error messages to guide the user through corrective actions.

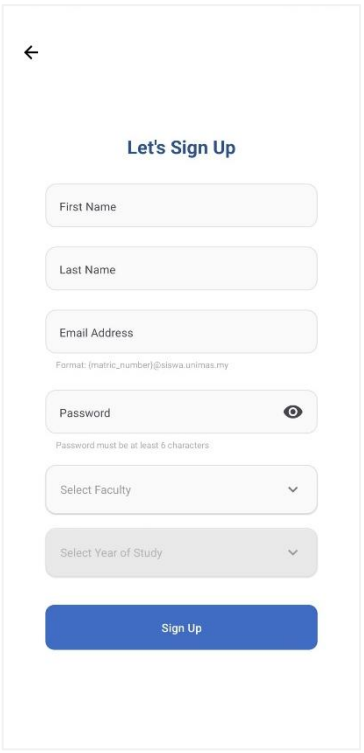


Figure 4.14. *Student's Sign-Up Screen*

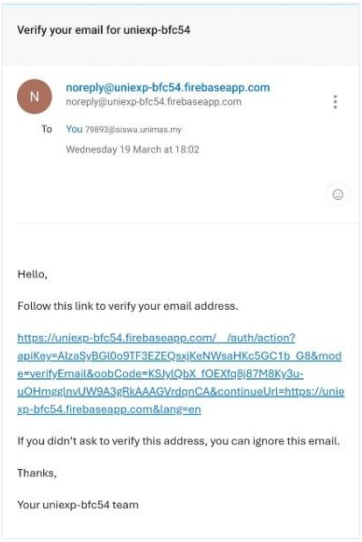


Figure 4.15. *Account Verification Email*

Furthermore, if the student does not have an account, they can navigate to the sign-up screen, as shown in Figure 4.14, by using the “Sign Up” button on the sign-in screen. From this figure, students are asked to fill in their first name, last name, university email address (only UNIMAS siswa accounts are accepted), faculty, and current year of study. After filling these details, it will undergo validation as shown in the code snippet from Appendix Figure F.3.

After successful submission, a pop-up message appears to the student that a verification email has been sent to the registered email address. To complete the process, the student must view their inbox and click on the verification link, as indicated in Figure 4.15. After verification, the user’s email verified status in Firebase Authentication is updated to true, and system access is granted to the user.

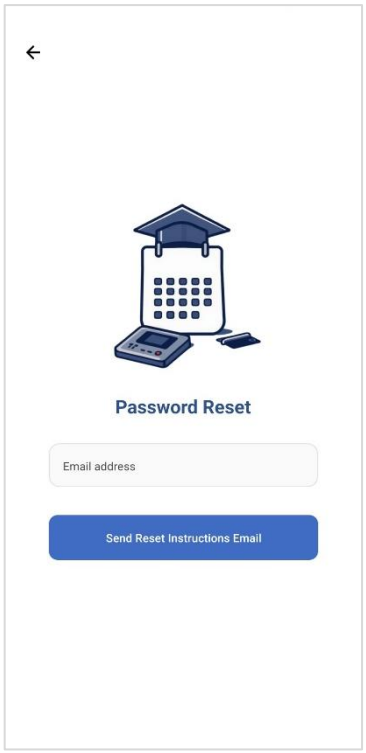


Figure 4.16. *Password Reset Screen*

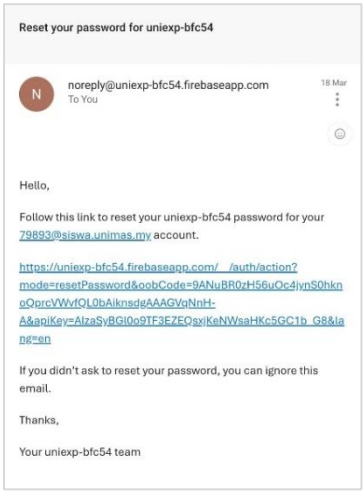


Figure 4.17. *Password Reset Email*

Additionally, users can reset their password through the *Forgot Password* prompt in Figure 4.11 if they forget it. Students are required to submit their registered email address in the password reset form as seen in Figure 4.16. If the system recognises the email address, an instruction email, as shown in Figure 4.17, will be sent to the inbox to guide the student through the password reset process, following the code logic outlined in Appendix Figure F.4.

4.5.1.2. Event Management and Notification

Figure 4.18. *Student's Site Event Listing*

Figure 4.19. *Event Details Screen*

The event management module allows students to browse and register for university events. Figure 4.18 shows the event listing page, which includes category filters, a search function, and pagination for lists with more than 5 events. Each event card displays the event poster thumbnail, the managing faculty, start date, event name, start and end time, and event location. Students can navigate to a selected event to view detailed information, as seen in Figure 4.19.

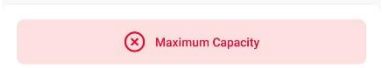


Figure 4.20. *Event Maximum Capacity Footer*

Figure 4.20 shows an event card when the event has reached full capacity, where the footer displays “Maximum Capacity” instead of the “Register Now” button, indicating that the event is currently full but may open if a participant cancels their registration.

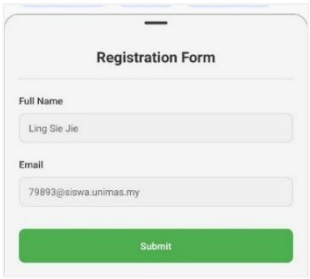


Figure 4.21. *Event Registration Form with Prefilled Details*

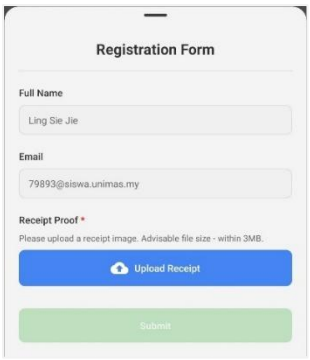


Figure 4.22. *Event Registration Form with Payment Proof Upload Requirement*

Students can register for events by clicking the "Register Now" button, as shown in Figure 4.19. Upon clicking, a registration form with prefilled fields (full name and email) is displayed as illustrated in Figure 4.21, allowing students to review their basic details before submission. Figure 4.22 illustrates an event that requires payment proof, where students must upload an image file (only image file formats are allowed) as payment verification for the administrator before completing the registration process.

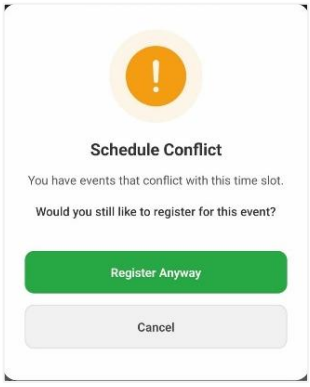


Figure 4.23. *Confirmation Window for Clashing Event Registration*

Figure 4.24. *Successful Registration Confirmation Window*

When a student attempts to register for an event that clashes with other registered events, a confirmation window will appear, prompting the student to confirm the registration despite the conflict, as shown in Figure 4.23. Upon successful registration, a confirmation window is displayed to inform the student of the successful registration, as shown in Figure 4.24.

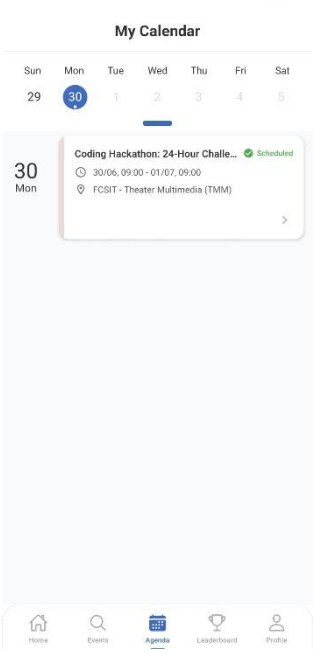


Figure 4.25. *Agenda Tab Screen*

Figure 4.26. *Registered Event Details Screen*

Lastly, students can manage their registered events through the *agenda* tab, as shown in Figure 4.25. In this tab, students can choose a date from the calendar to see what events were planned for that day. Each event card displays the category (indicated by the left border colour), event name, start and end date and time, status, and registration-related details such as attendance and verification status. Figure 4.26 represents the registered event details which can be accessed by going from the event card in agenda calendar. This screen provides a comprehensive overview of the registered event and includes the option to cancel the registration, available only if the current time is more than one hour before the event starts.

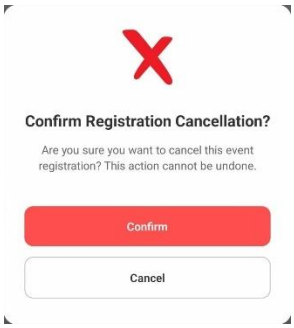


Figure 4.27. *Registration Cancellation Confirmation Modal*

If the student decides to cancel the registration, they can tap the “Cancel Registration” button, and a modal will pop out to prompt the student to confirm they want to cancel the registration, as shown in Figure 4.27. When they tap “Cancel” on the modal, the system will delete the registration entry, send a notification to the student, and update the student's quest progress, as illustrated using code snippet logic in Appendix Figures F.5 through F.7.

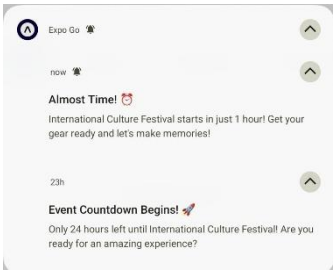


Figure 4.28. *Event Starting Soon Notifications*

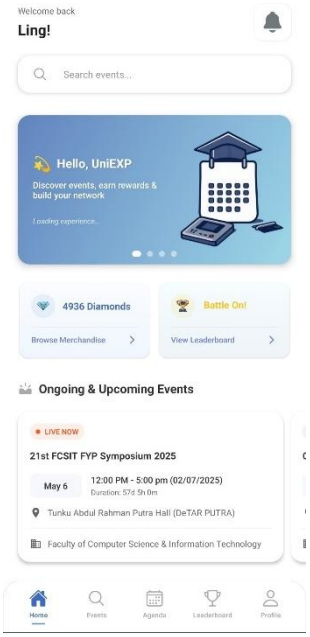


Figure 4.29. *Student - Home Page with Notification Icon*

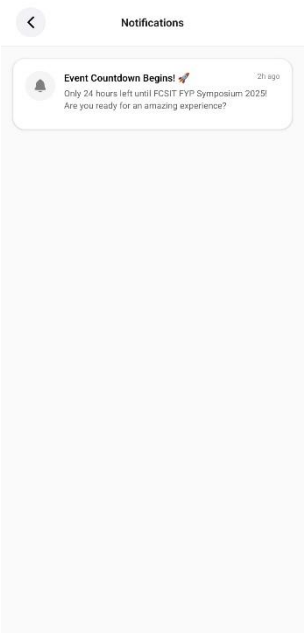


Figure 4.30. *Student - Notification List Page*

Based on Figure 4.28, students will receive two notifications as reminders for upcoming events. The first notification is sent one day before the event is scheduled to begin, serving as an early reminder. The second notification is delivered one hour before the event starts, providing a timely alert to ensure students are prepared and do not miss the event. These notifications are managed by the cron jobs through the Firebase Cloud Functions which run every 5 minutes. All the sent notifications will be visible in the notification list as shown in Figure 4.30. The navigation to the notification list is through the bell icon at home page as shown in Figure 4.29.

4.5.1.3. Gamification and Networking

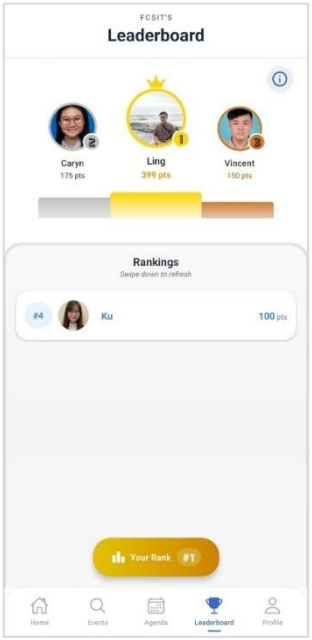


Figure 4.31. *Faculty-Based Leaderboard*

Figure 4.32. *Leaderboard Information Guide*

The gamification module begins with a faculty-based leaderboard as shown in Figure 4.31. Each faculty has its own leaderboard, allowing students to compete by participating in events and completing quests. The leaderboard is refreshed monthly; when a new month starts, the previous month's rankings are determined through the function shown in Appendix Figure F.8, which would be displayed on a modal that will be seen on this screen to tell the students their ranking and award them virtual currency (diamonds) based on their ranking. There is an information guide icon at the top right of the leaderboard screen for students seeking clarification (Figure 4.30), which details the rewards for ranking and how to earn points.

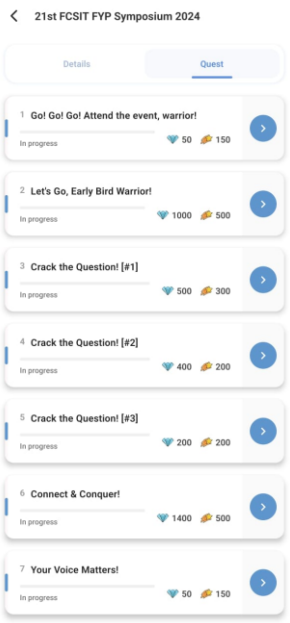


Figure 4.33. *Event Quests List*

The gamification module also introduces quests to enhance student engagement within events. As shown in Figure 4.31, the quest list is displayed within the selected event. The quests are primarily divided into five major categories: Attendance-based, Early Bird Attendance, Q&A, Networking-based, and Feedback-driven. These quests promote active involvement and reward students for participating in a variety of event-related activities.

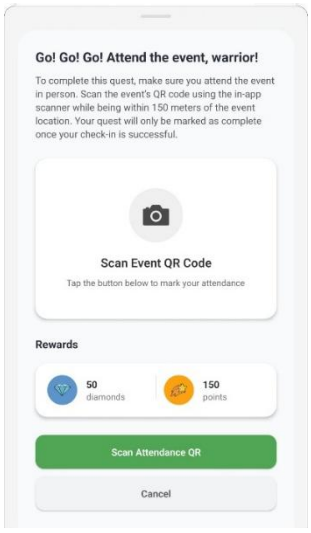


Figure 4.34. *Attendance Quest (Default Interface)*

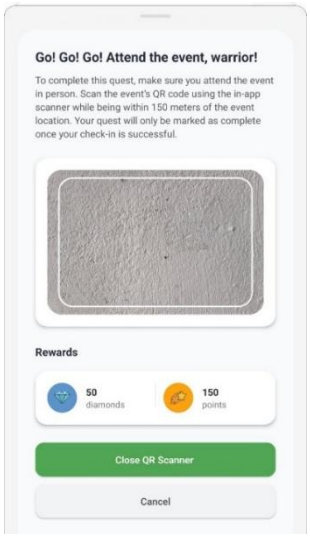


Figure 4.35. *Attendance Quest (QR Scanning Mode)*

One of the key quest types in the gamification module is the Attendance-Based Quest, which transforms traditional attendance check-ins into interactive quests. When the attendance-based quest card is opened, a bottom sheet appears (Figure 4.34) displaying the quest name, description, and rewards section. To participate, the student must scan the event's attendance QR code shown by the admin. Upon clicking the Scan Attendance QR button, the bottom sheet interface switches into camera scanning mode, as shown in Figure 4.35. For first-time access, the system will request permission for location tracking and camera access to proceed with the quest.

Figure 4.36. *Too Far Distance Error*

Figure 4.37. *Expired QR Code Error*

Figure 4.38. *Invalid QR Code Error*

To make the quest fair and to discourage cheating behaviour, several measures that are aimed at preventing misuse have been incorporated into the attendance-based system based on the logic illustrated as code snippet in Appendix Figure F.9. As it is shown in Figure 4.36, when a student tries to scan the QR code from outside the 150-metre radius, which is measured from the event's geolocation of latitude and longitude, the system will provide a warning message that the user is not near the event venue. Consequently, only people in the set venue can finish the quest. Figure 4.37 illustrates another safeguard for attendee completion, utilising a QR code expiration. The validity of each code lasts for three seconds, and it is automatically renewed. If the scanned QR code is outdated, the user will be notified of an error and will be prompted to use a new one. This action prevents users from completing the quest because they must upload a screenshot or picture of the code from the person who is actually at the event. Finally, Figure 4.36 covers cases in which an invalid event QR code is scanned. When a student

attempts to scan a code from a different event, the system reports a mismatch and displays an "Invalid QR Code" message, instructing the user to scan a code for the current event.

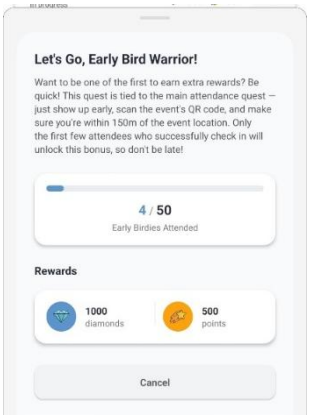


Figure 4.39. *Early Bird Attendance Quest*

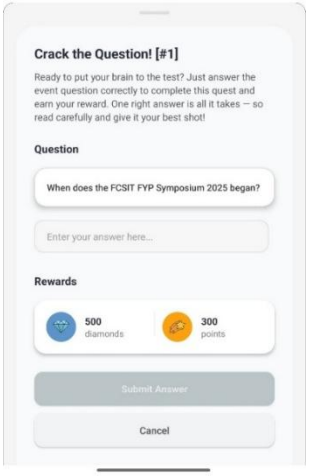


Figure 4.40. *Q&A Quest*

Another way to promote active participation is through the Early Bird Attendance Quest, as illustrated in Figure 4.39. This quest is connected with the attendance-based quest mechanism depicted in Figure 4.34, and it identifies students who are punctual for the event. This goal enables attendees to arrive at the venue on time, ensuring a smooth kick-off. After scanning the QR code and getting verified by the system's location confirmation, the system determines whether the student is one of the early bird attendees (the first 50) in real time where the logic is handled by the code snippet illustrated in Appendix Figure F.10. In such a case, the quest is deemed to be completed, and rewards are given to the student. Nevertheless, if the student arrives after the early bird limit, the quest is deemed to be unsuccessful. The quest card is marked as failed with a red left border on the quest card.

In addition, Figure 4.40 illustrates that the Q&A Quest enables participants to engage with questions, provided by guest speakers or event organisers, to test their knowledge base related to the event content. Users are presented with a question and a box in which they can enter their answer. After the submission, the system verifies the answer via the logic presented in Appendix Figure F.11. If the answer is wrong, the system suggests an error under the input

field. When the answer is correct, the quest is marked as completed with a completion modal, and the student can proceed to receive rewards.

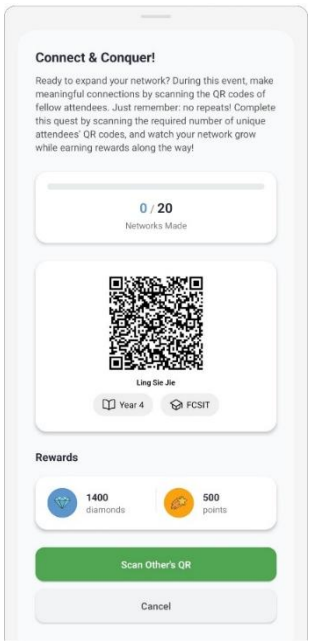


Figure 4.41. *Networking Quest (QR Display Mode)*

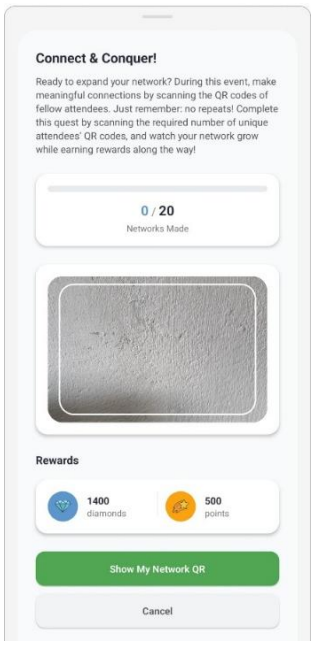


Figure 4.42. *Networking Quest (QR Scanning Mode)*

The networking-based quest encourages students to build meaningful connections with fellow attendees during events. It is aimed at creating broader campus interaction by encouraging the students to make contact with new people. As illustrated in Figure 4.41, the quest sheet displays relevant details including the quest name, description, completion criteria, an identity QR code, reward information, and a toggle to switch between QR display and scanning modes. In QR display mode (Figure 4.41), each student presents a personalised QR code containing basic attendee information. Other participants can scan this code to progress in their own quest. For instance, if the quest requires making 20 new connections, the student must scan 20 unique, event-registered attendees' QR codes to complete it.

Figure 4.42 displays the QR scanning mode, which activates the camera interface for scanning other students' QR codes. When a student scans a QR code, the system processes the data according to the logic illustrated in Appendix Figure F.12. To prevent misuse, several

cheating prevention measures are implemented. These include displaying an error if a student scans their own QR code, if the scanned user is not registered for the event, if the QR code was already used in a previous event, or if the code is unrelated or invalid. Upon a successful scan, the system updates the student's quest progress and increments their achievement badge progress as seen in Appendix Figure F.13 and Figure F.14. After reaching the required number of connections, the user is entitled to claim the rewards for completing the quest.

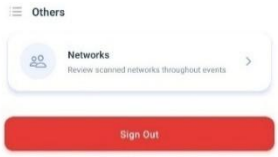


Figure 4.43. *Network Navigation in Profile Screen*

The connections made during this quest also enable users to chat with one another. To access their network, users can navigate to the "Others" section on the profile page, as illustrated in Figure 4.43.

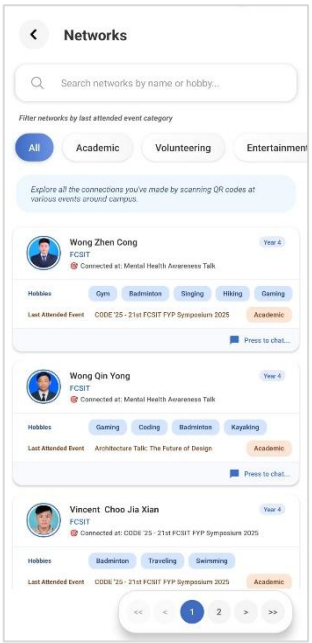


Figure 4.44. *Network List*

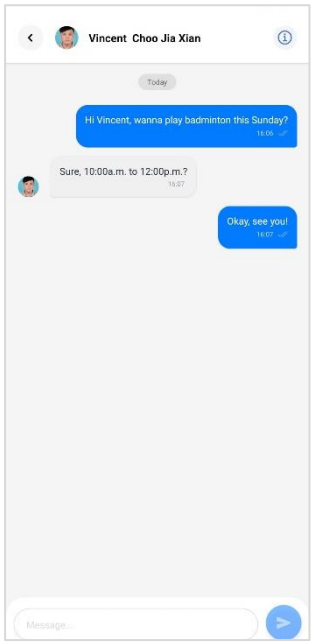


Figure 4.45. *Chat Function Interface*

This section (Figure 4.44) displays the list of connections that students have established through participation in the networking quest associated with an event. Students can either

search for a specific connection by name or browse through the entire list using the built-in pagination feature for easier navigation, especially when the list grows over time.

To improve the relevance and user experience of the network list, two enhancements were introduced: hobby-based filtering and the last attended event indicator. The hobby-based filtering feature allows students to filter their connections based on shared interests, which are selected during the first-time sign-in process. Each user may choose up to five hobbies from a predefined list, enabling them to identify and connect with peers who share similar extracurricular interests. This encourages more meaningful and interest-driven networking among students.

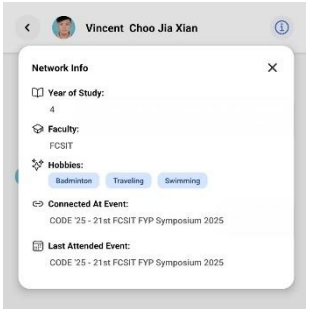


Figure 4.46. *Network Information Modal*

In addition, the network list displays a tag indicating the last event attended by each connection. This feature provides contextual information that helps students recall where they met their peers and also allows them to filter or group connections based on shared event experiences. These enhancements are designed to create a more personalized and purposeful networking environment within the platform. To initiate a conversation, students simply tap on any connection in the list. This action will open the messaging interface, where users can engage in real-time chat, as demonstrated in Figure 4.45. Students may click the information guide icon on the top right as shown in Figure 4.45 to see the network information as shown in Figure 4.46.

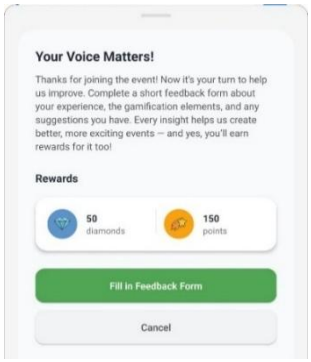


Figure 4.47. *Feedback-Driven Quest*

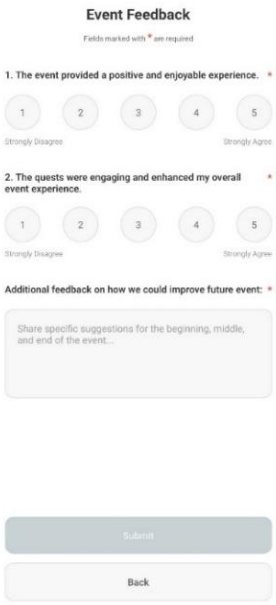


Figure 4.48. *Event Feedback Form*

The Feedback-Driven Quest reinvents traditional event feedback into an exciting quest format, that rewards students for their reflections after the event. Figure 4.47 reveals that the quest sheet where students can access details and launch the quest by clicking on the “Fill in Feedback Form” button. Students are directed to the feedback form section, as indicated in Figure 4.48, after clicking this link. As shown in Figure 4.48, the feedback form consists of three main sections. Event Satisfaction, User Experience of the Quest and Additional Feedback. These parts help gather thoughtful feedback from attendees, which enables organisers to improve future events. The submit button for the form is disabled until all required fields are completed by students. When the form is successfully submitted, an animation appears on the screen to indicate that the student’s rewards have been granted.

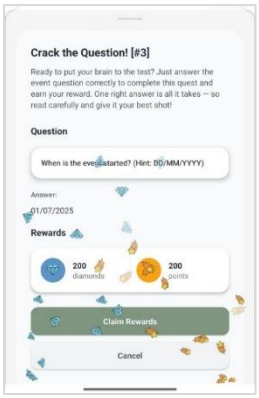


Figure 4.49. *Rewards Claiming Animation*

To enhance user feedback and satisfaction, a captivating rewards claiming animation is triggered when users complete quests. As illustrated in Figure 4.49, after the quest is completed, a “Claim Rewards” button appears, giving the user the opportunity to claim their rewards. When the button is pressed, an interactive and visually attractive animation appears, with rewards being credited to the user's account and this month's leaderboard entry. During the entire animation process, the button is disabled, preventing the user from submitting multiple times. After the animation is complete, the button is hidden, and the quest card is flagged with “Rewards Claimed,” indicating that the user has claimed their rewards.

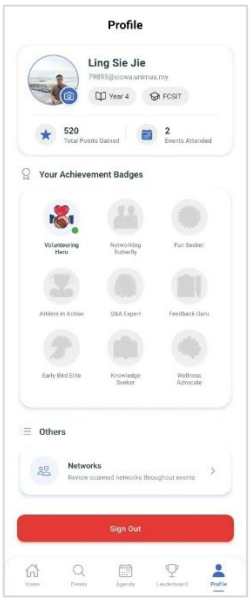


Figure 4.50. *Badges List in Profile Page*

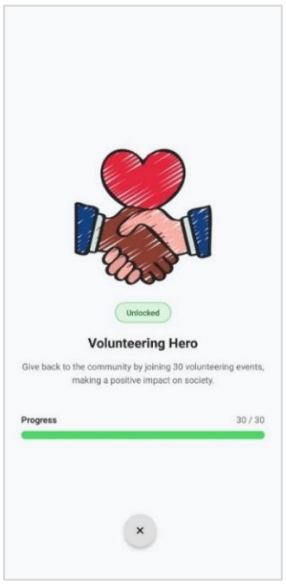


Figure 4.51. *Badge Details Page*

The achievement badge feature further motivates student participation by rewarding milestones. As shown in Figure 4.50, the profile page displays a list of achievement badges, with a total of nine badges currently introduced. Locked badges are tinted in grey, indicating that the user has not yet met the completion criteria. Unlocked badges are displayed in full colour and marked with a green dot. Figure 4.51 shows the badge details page, which can be accessed by clicking on any badge icon. Every badge has their own unlocked criteria. For example, the **Volunteering Hero** badge requires the student to participate in 30 volunteering-type events through attendance-based quests (verified by location) to unlock the badge.

4.5.1.4. Merchandise Management

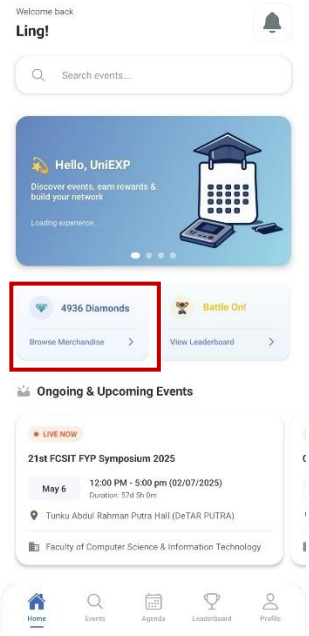


Figure 4.52. *Merchandise Management Navigation from Home Screen*

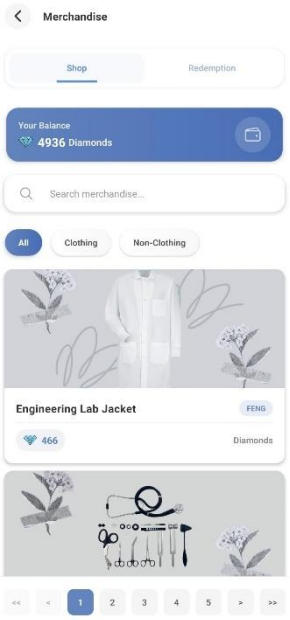


Figure 4.53. *Merchandise Listing*

Lastly, the merchandise management module allows students to browse and redeem items using their virtual currency gained through the gamification elements in UniEXP. Figure 4.52 shows the navigation to the merchandise listing from the home screen, where the student can also view their total virtual currency balance (highlighted with a red coloured frame). Figure

4.53 displays the merchandise listing with category filters (All, Clothing, Non-Clothing), search function, and pagination, enabling students to easily browse, search, and select items based on their preferred category.

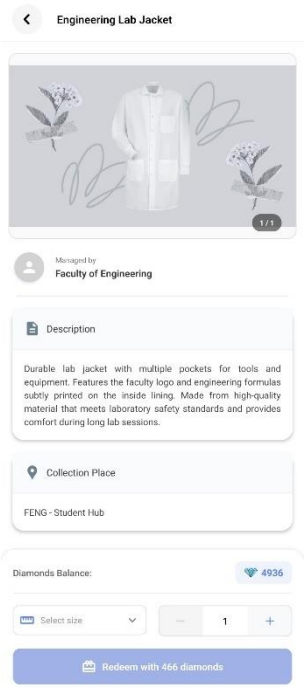


Figure 4.54. *Merchandise Item Details (Clothing) with Size and Quantity Selection*

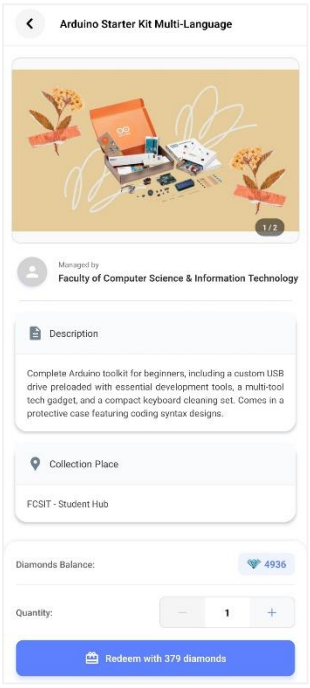


Figure 4.55. *Merchandise Item Details (Non-Clothing) with Quantity Selection*

The redemption process allows students to view and redeem merchandise items using their virtual currency. Figures 4.54 and 4.55 show the merchandise item details, including the item image catalogue, the faculty in charge, description, and collection place for students to review. The first figure displays a clothing item, which includes a size selection and quantity selection. The redemption button, showing the total virtual currency to be spent, remains disabled until a size is selected. The second figure displays a non-clothing item, which only requires quantity selection for redemption.

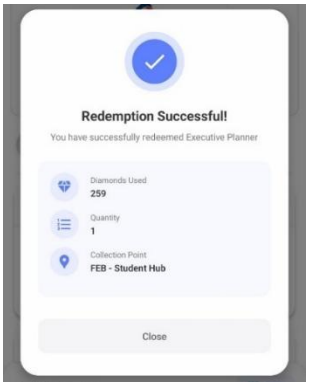


Figure 4.56. *Successful Redemption Confirmation Pop-up*

Upon successful redemption, a pop-up modal, as shown in the figure above, appears to indicate that the redemption was successful, displaying brief details of the redeemed item.

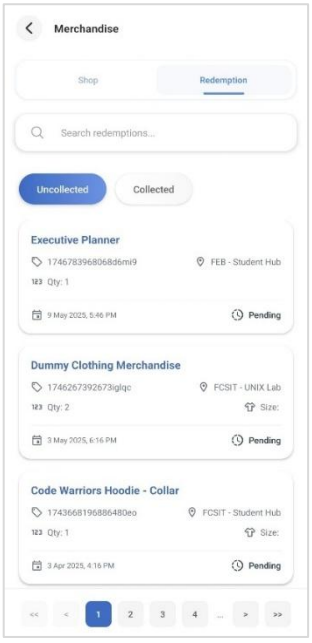


Figure 4.57. *Redemption Item List (Uncollected)*

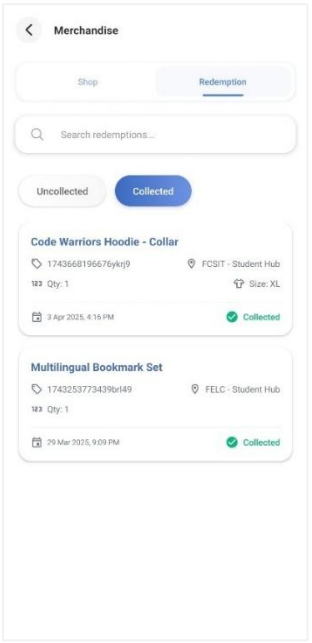


Figure 4.58. *Redemption Item List (Collected)*

In addition, there is also a *Redemption* tab in this module that allows users to access their redemption list to view the details of previously redeemed items. As shown in Figure 4.57, items with a pending collection status are displayed in *Uncollected* tab, while items marked as collected (refer to Figure 4.58) in *Collected* tab, indicated with a green tick icon.

4.5.2. Event Administrator Modules

This section describes the core modules implemented in the UniEXP admin web system. Each module was designed to support the essential tasks of admin or faculty-based event organiser users, ensuring smooth management of events, participants, and system data. Relevant system screenshots are included to illustrate the module interfaces and functionalities.

4.5.2.1. Authentication

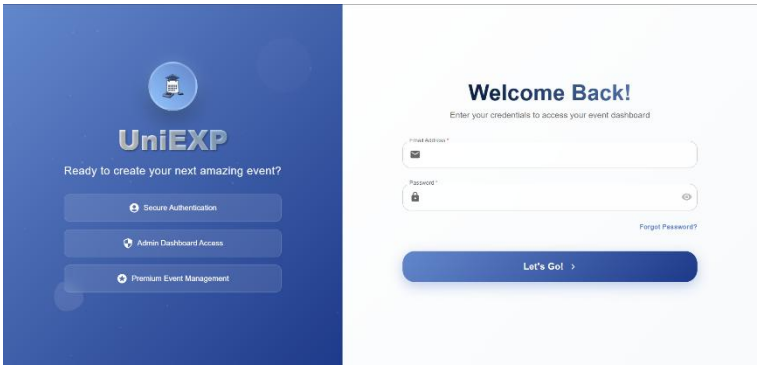


Figure 4.59. *Authentication (Sign In) Module*

The system started with the login process, as shown in Figure 4.59. This page provides fields for entering the email address and password where each faculty has a unique pair of login credentials.

When a user submits their credentials, the system performs an important validation step, as illustrated in Appendix Figure F.15. The system first validates that all fields are not left empty. If all fields are validated, the system queries the admin collection in the Firestore database to check if the provided email belongs to an authorised admin. If no match is found, the system identifies the user as unauthorised (likely a public user or student) and triggers an error message using a snackbar notification. If the email is valid, the admin's data is retrieved and stored in local storage and the system proceeds to authenticate the user using *signInWithEmailAndPassword*.

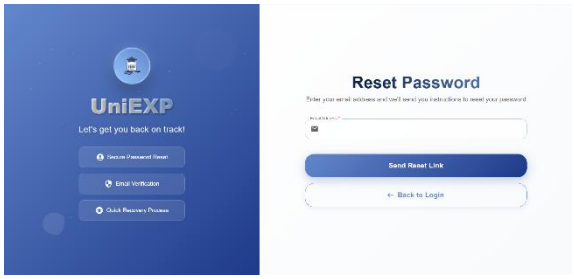


Figure 4.60. *Authentication (Password Reset) Module*

Figure 4.61. *Password Reset Email*

Besides, the system also provides an option to reset the password on the sign-in page (refer to Figure 4.60) if the admin forgets it. On this page, the user is prompted to enter their email address associated with the admin account.

Once submitted, the system sends a reset link to the provided email address, as illustrated in Figure 4.61, allowing the user to securely set a new password.

4.5.2.2. Event and Quest Statistics

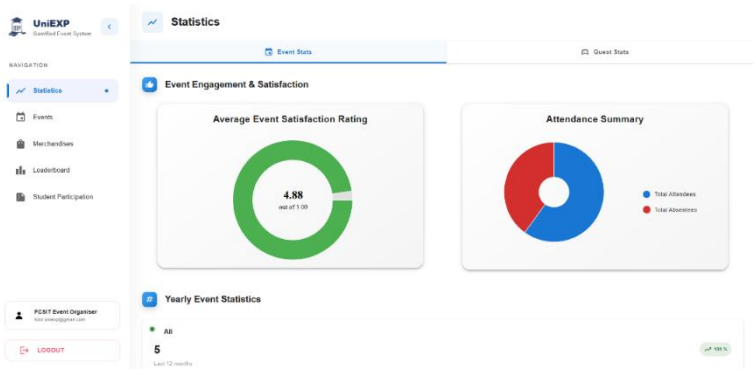


Figure 4.62. *Event-Related Statistics – Part I*

This module offers admins a visual and data-driven overview of the events they have organised within the UniEXP platform. This module presents two key sets of information through intuitive donut charts: event satisfaction and attendance breakdown. The average satisfaction rating is calculated based on Likert scale responses submitted by participants via the feedback-driven quest. The results are visualised using a donut chart to clearly reflect the distribution of ratings, providing a quick assessment of participants' satisfaction levels. In parallel, an

attendance summary donut chart displays the proportion of students who attended versus those who were absent across events.

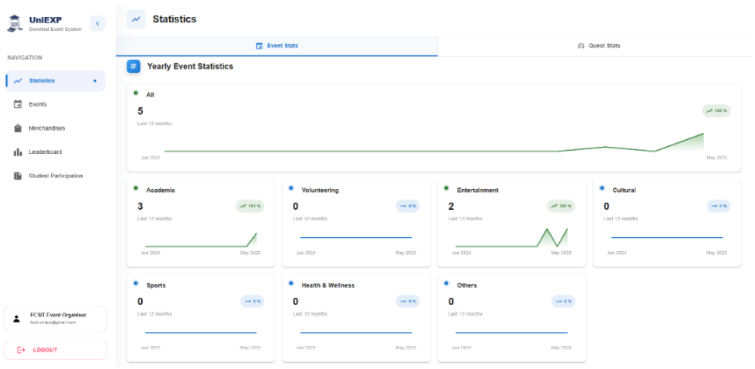


Figure 4.63. *Event-Related Statistics - Part II*

Beneath the satisfaction and attendance charts, the module also features a Yearly Event Statistics section (see Figure 4.63), which provides a comprehensive summary of the events organised by the admin over the past 12 months. Events are automatically categorised by type, allowing admins to monitor the diversity and focus of their event planning. Each event type is displayed within its own card, accompanied by a line graph that visualises the monthly trend of events organised under that category.

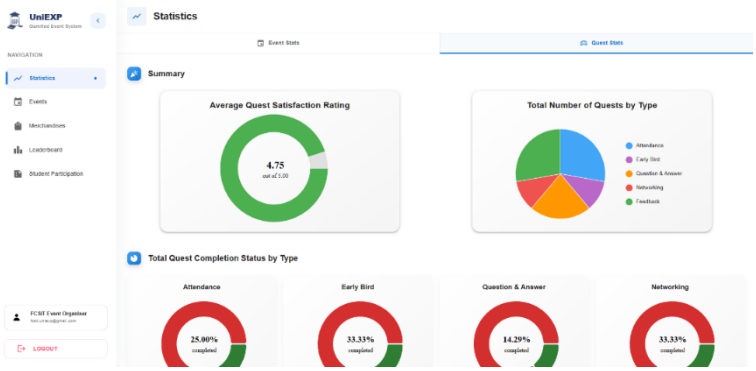


Figure 4.64. *Quest-Related Statistics*

As shown in Figure 4.64, this module provides admins with insightful visual summaries of participant engagement and feedback for quests within their managed events. It features a donut chart displaying the average quest satisfaction rating, derived from participant responses on a Likert scale collected through feedback-driven quests. Additionally, a pie chart illustrates

the distribution of quests created, categorised by their types. This module also presents completion statistics for each quest type, indicating the percentage of participants who have completed or not completed the respective quests.

4.5.2.3. Event Management, Feedback Management, and Gamification

This module enables administrators or event organisers to efficiently manage university events within the system. It plays a crucial role in ensuring that event information remains current and accurate, allowing students to browse, register for, and participate in events. Administrators can create new events, update event details, manage participant lists, and monitor event progress to ensure smooth execution.

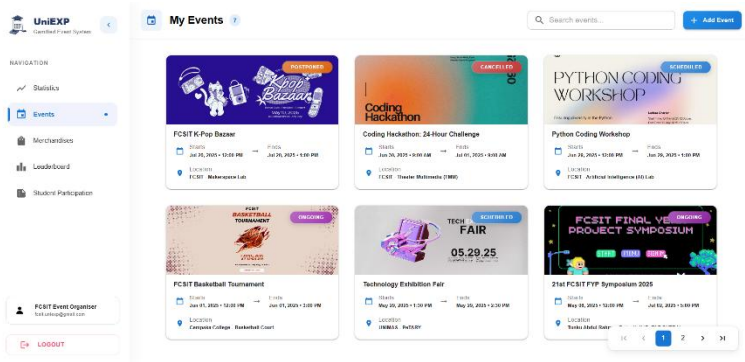


Figure 4.65. *Event Listing Page*

The main interface of this module, as shown in the figure above, displays the event listing cards which presents a summary of the events managed by this admin. Each entry typically includes details such as event name, start date time, end date time, poster thumbnail, venue, and status. The system can display a maximum of six event cards per page and is therefore equipped with a pagination function to allow the admin to view additional events.

At the top-right of the page, there is an “Add Event” button allow the admin to easily add new event entry along with the quests to the list by filling the event creation form shown in Appendix Figure F.16. The event creation process consists of two main steps: Details Creation and Quests Creation. In the first step, the system requires users to input essential

information for the new event, including up to four event-related posters, the event name, description, category, start date and time, end date and time, registration closing time, location, geolocation pinpoint, and registration requirements.

From the event creation form, the geolocation pinpoint is a critical field to note because the attendance-based quest relies on the latitude and longitude of the pinned location. Students must be within a 150-meter radius of this location to fulfil the attendance requirement. Additionally, the event end date and time must be set at least one hour after the start date and time, while the registration closing date and time must be scheduled at least one hour before the event starts.

For certain events, specific requirements may apply, such as participant capacity limits or eligibility restrictions. The system allows administrators to enable a limited capacity setting, which requires them to specify the maximum number of participants that can be accommodated. Additionally, administrators can restrict event access to students from a particular faculty and, if needed, further narrow eligibility by academic year. When faculty restriction is enabled, only students from the selected faculty can register. If year restriction is also enabled, only students from the specified year levels within that faculty are allowed. Moreover, for paid events, administrators can require proof of payment, prompting students to upload supporting documents during the registration process. Once all required fields are completed, the "*Next*" button located at the bottom right of the page becomes active, allowing the administrator to proceed to the second step by activating the Quests Creation tab. This behaviour is controlled by the attached code snippet provided in Appendix Figure F.17.

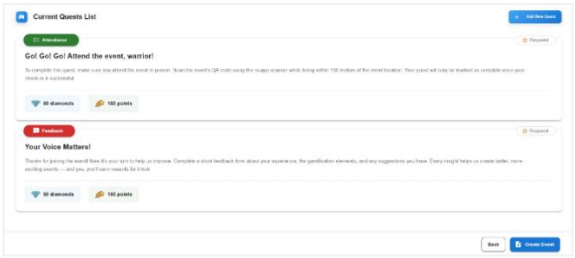


Figure 4.66. *Default Quests in the List*

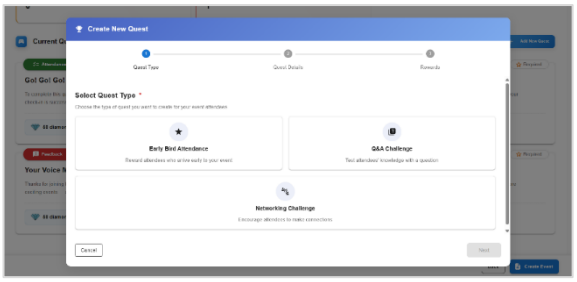


Figure 4.67. *Quest Type Selection*

In the Quests Creation tab, the system allows admins to manage event-specific quests after completing the event details setup. By default, two quests (an attendance-based quest and a feedback-driven quest) are automatically added to the active quest list and cannot be removed or edited. If the admin wishes to add additional quests, they can do so by clicking the *Add New Quest* button, as shown in Figure 4.66 above, which opens a quest creation dialog form for entering the relevant quest details, as illustrated in Figure 4.67.

After the quest creation form is loaded, the first step is a selection of the quest type by the administrator from three options provided: Q&A, Networking, or Early Bird Attendance. It should also be noted that Early Bird Attendance and Networking quest types can only be selected once per event to ensure variety and avoid duplication in the active quest list.

Figure 4.68. *Early Bird Attendance Quest Creation Field*

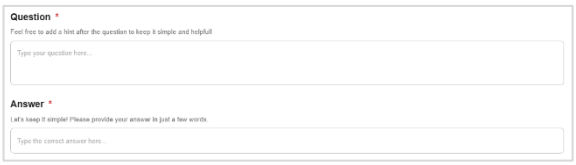


Figure 4.69. *Question & Answer Quest Creation Fields*

Figure 4.70. *Networking-Based Quest Creation Field*

After choosing the quest type and moving forward, the system will ask the administrator to input the quest-specific details. For instance, the Early Bird Attendance quest requires defining the number of attendees that would be eligible for getting the reward based on being

among the first in completing the attendance-based quest. In the Q&A quest, one has to enter the question combined (as an optional feature) with a hint and the correct answer, and in the Networking quest, one has to specify the number of connections that the attendee must make at the event to complete that quest.

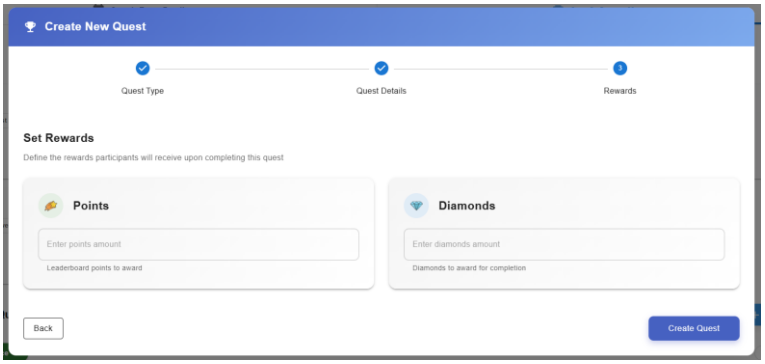


Figure 4.71. *Quest Creation - Rewards Section*

Once finished with these details, the admin will then move to the rewards section (refer Figure 4.71), where required to specify the number of points (leaderboard points) and virtual currency (diamonds) should be awarded to the students once they completed the quest. Lastly, by clicking “Create Quest” button, the quest will be appended to the active event quests list.

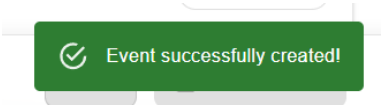


Figure 4.72. *Event Creation Success Snackbar*

When the administrator has verified the quests to be included in the event, they can press Create Event button (as shown in Figure 4.66). After a successful submission, snackbar notification would appear to notify the administrator that the event has been successfully created.

To view or make changes to the event details, the admin must select an event from the list shown in Figure 4.65, the admin can view or edit its details through a prefilled form. Based on Appendix Figure F.18, the form displays fields such as the event poster catalogue with an

upload button, name, description, category, start date and time, end date and time, registration closing date, location, geolocation pinpoint, and specific event requirements. At the bottom-right of the form, there is a "*Save Changes*" button, which is initially disabled and only becomes enabled when the admin makes changes to any field to reflect them in the Firebase database.

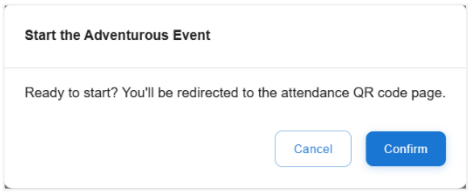


Figure 4.73. *Event Starting Confirmation Dialog*

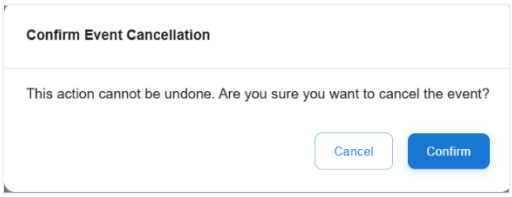


Figure 4.74. *Event Cancel Confirmation Dialog*

Figure 4.75. *Event Attendance QR Code Page*

In addition, this module has a functionality that enables you to start the event using the Start Event button on the top right of the page (refer Appendix Figure F.18). When clicking on this button, a confirmation dialog, as shown in Figure 4.73, is displayed to make sure the administrator really wants to start the event. If the event is confirmed, the event status is marked as 'Ongoing' and an event attendance QR code is displayed on the interface, as shown in Figure 4.75. The QR code displayed here is properly encrypted with a secret key, as it is contained in the codebase. The automatic refresh mechanism programmed in the QR code refreshes every 3 seconds to reduce the chance of attendees misusing or cheating.

Apart from that, this module also features a Cancel Event button, located next to the Start Event button, which can be used to cancel an event. When this button is clicked,

confirmation dialogue appears, as shown in Figure 4.74, to confirm that the administrator wishes to cancel the event. If the event is confirmed, the event status will be changed to "Cancelled" and the event will be hidden from student application event list.

Furthermore, appendix Figure F.19 shows the Quest Manager module which allows administrators or event organisers to monitor and manage event-specific quests for the selected event effectively. Through the Quest tab of the event, it shows the quest manager of the selected event. Administrators can see an overview of participant progress and the status of all quests related to the event. The first section of the Quest tab provides a summary of the total number of participants who have successfully completed each quest while the second section contains the list of existing active quests. Quests in the list are visually differentiated by colour, the colour coding representing different types of quests.

To add a new quest to the existing list of event quests, the administrator can click the "Add New Quest" button, which opens a dialogue form (refer to Appendix Figure F.20) prompting the administrator to select the quest type. It is important to note that if an Early Bird Attendance or Networking quest has already been added, these options will no longer be available for selection to ensure quest variety. After selecting the desired quest type, the administrator must fill in the necessary quest details and then submit the quest creation form to add the new quest to the active quest list.

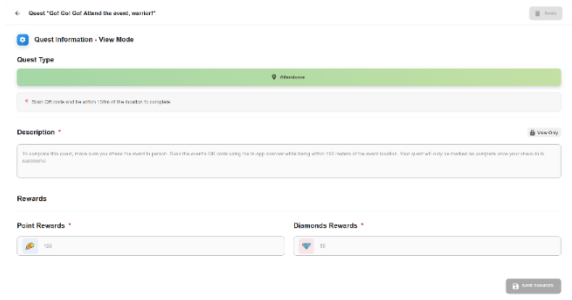


Figure 4.76. *Attendance-Based Quest Manager*

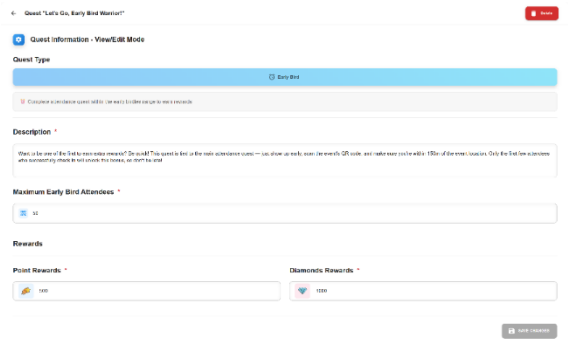


Figure 4.77. *Early Bird Attendance Quest*

To maintain the integrity of core event tracking, this module locks down the specifics of the two default quests, Attendance (refer to Figure 4.76 as an example) and Feedback. Conversely, the details for the other three quest types: Q&A, early bird attendance (refer to Figure 4.77 as an example), and networking, can be customised to accommodate the flexibility of the event's engagement strategies. Once the administrator edits any of the editable fields, the Save Changes button automatically becomes active, allowing the updated quest information to be submitted and updated in the Firebase database.

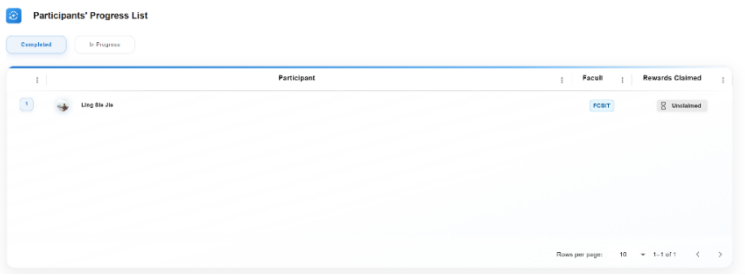


Figure 4.78. *Quest Progress Monitoring List*

As shown in Figure 4.78, a progress monitoring table is assigned to each quest in this module, allowing administrators to track progress. This table provides a breakdown of participants who have completed and those who have not, allowing the admin to see real-time engagement throughout the event.

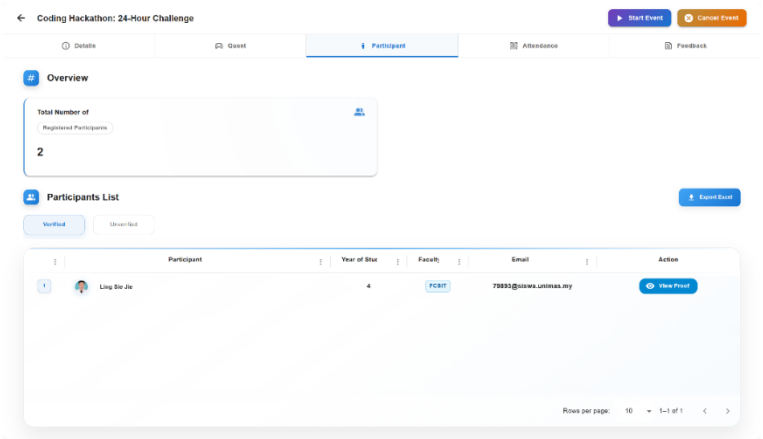
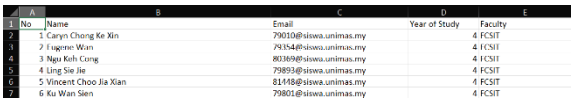


Figure 4.79. *Event Participants (Verified) Manager*

In the participant module, the system allows the administrator to track all participants registered for the selected event. The administrator can view the total number of registered participants first, followed by a table of participants, as shown in Figure 4.79. The presence of Verified and Unverified tabs depends on whether payment proof is required for the event. For events requiring payment proof, the Verified tab displays participants whose payments have been confirmed. Moreover, as seen in the Figure 4.80, the admin can export the list of participants to an Excel file for further analysis.



No	Name	Email	Year of Study	Faculty
1	Caryn Chong Ke Xin	79010@siswa.unimas.my	4	FCSIT
2	Eugene Wan	79354@siswa.unimas.my	4	FCSIT
3	Ngu Keh Cong	80369@siswa.unimas.my	4	FCSIT
4	Ling Sie Jie	79893@siswa.unimas.my	4	FCSIT
5	Vincent Choo Jia Xian	81418@siswa.unimas.my	4	FCSIT
6	Ku Wan Sien	79801@siswa.unimas.my	4	FCSIT

Figure 4.80. *Participants List Excel File*

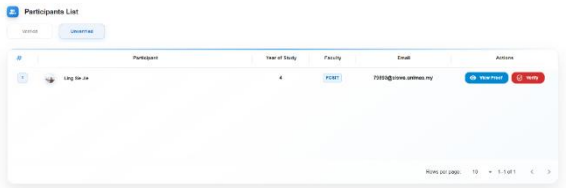


Figure 4.81. *Event Participants (Unverified) Manager*

Moreover, the admin can navigate to the Unverified tab to review the list of participants awaiting verification as shown in Figure 4.81. By clicking the View Proof action button, the admin can examine each participant's submitted payment proof and can update their status by using the Verify button.

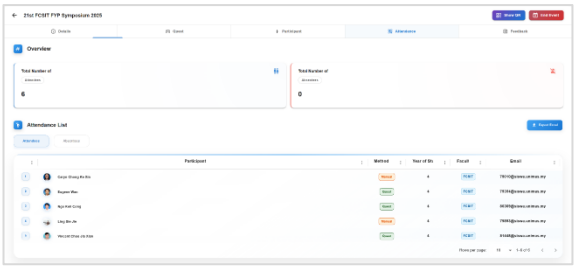


Figure 4.82. *Event Attendance (Attendees) Manager*

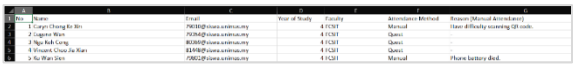


Figure 4.83. *Attendees List Excel File*

Within the Event Management module, there is an attendance management section, as shown in Figure 4.82 that helps administrators track the event's participation. The system displays a summary of the total number of attendants and absentees at the top of the page, providing a quick idea of how the attendance is performing. Under this summary, the administrator can check the attendance status of each individual participant on a detailed attendee list and also see the method they used to attend the event. This system provides an 'Export to Excel' option to download attendance details for further analysis, as shown in Figure 4.83.

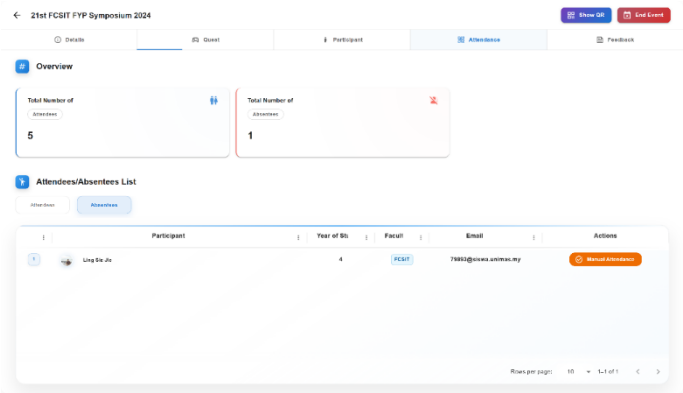


Figure 4.84. *Event Attendance (Absentees) Manager*

Additionally, the system provides an Absentees tab (refer Figure 4.84), enabling administrators to view the list of absent participants. When necessary, administrators can manually mark attendance for students who faced issues with QR code scanning by clicking the action button next to the respective participant and entering the reason for the manual

override. This feature ensures that students can still fulfil attendance-based quest requirements despite encountering technical difficulties as handled by the logic in appendix Figure F.21.

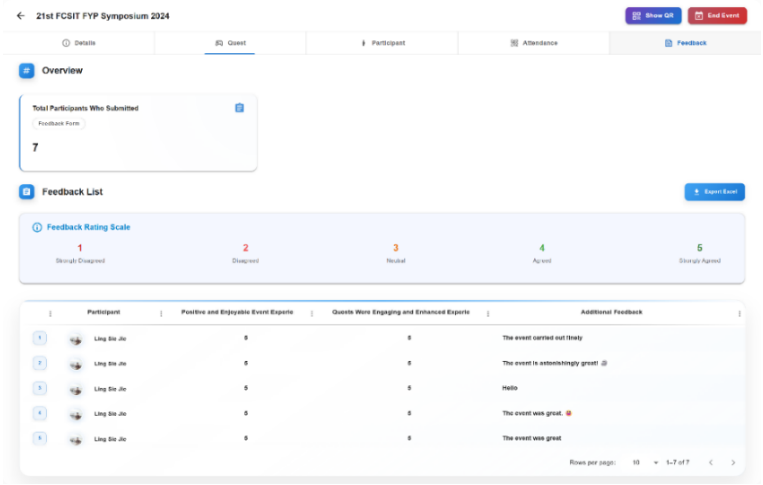
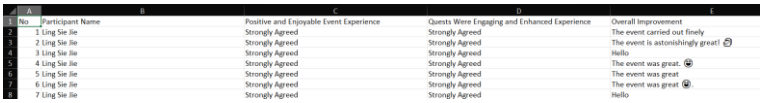


Figure 4.85. *Feedback Management Module*

Based on Figure 4.85, the event management also provides feedback management section which allows admins to manage and review participant feedback in an efficient way for each event. Through the navigation to the Feedback tab of the selected event, the admin can track the engagement of the participants and collect useful insights that may help to make future events better.

In this section, administrators can track the overall number of participants that have filled out the feedback form, which can be associated with the completion of a quest based on feedback. This provides a summary of engagement in events, as well as participation rates in quests.

The feedback section also provides a summary view of participant responses, including key areas such as event satisfaction, and satisfaction with gamification, as well as any other comments or suggestions that can be used to inform the organisation of future events.



No	Participant Name	Positive and Enjoyable Event Experience	Quests Were Engaging and Enhanced Experience	Overall Improvement
1	Ling Sie Jie	Strongly Agreed	Strongly Agreed	The event carried out finely
2	Ling Sie Jie	Strongly Agreed	Strongly Agreed	The event is astonishingly great!
3	Ling Sie Jie	Strongly Agreed	Strongly Agreed	Hello
4	Ling Sie Jie	Strongly Agreed	Strongly Agreed	The event was great.
5	Ling Sie Jie	Strongly Agreed	Strongly Agreed	The event was great
6	Ling Sie Jie	Strongly Agreed	Strongly Agreed	The event was great
7	Ling Sie Jie	Strongly Agreed	Strongly Agreed	Hello

Figure 4.86. *Feedback List Excel File*

For further analysis or recordkeeping, the interface has an Export Excel button, which enables the administrator to export the full feedback list into an Excel file, as shown in the figure above.

4.5.2.4. Merchandise Management

This module enables administrators or event organisers to efficiently manage the university's redeemable merchandise in the system. This module is essential for maintaining up-to-date merchandise offerings, which can be redeemed by students using their (diamonds).

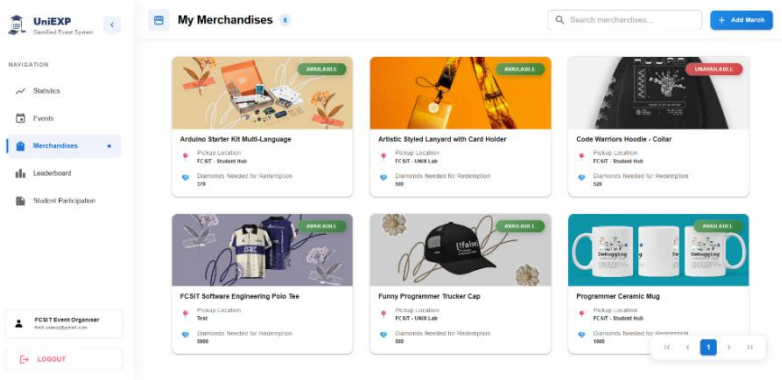


Figure 4.87. *Merchandise Listing Page*

The main interface, as shown in Figure 4.87, displays the merchandise listing cards which presents a summary of the merchandise managed by this admin. Each entry typically includes details such as merchandise name, thumbnail, pickup location, price in virtual currency, and current availability status.

At the top of the page, there is an "Add Merch" button to allow the admin to easily add new merchandise entry to the list by filling the merchandise addition form shown in Appendix Figure F.22. The system requires users to input the necessary details for a new merchandise item, including maximum 4 images, name, description, price in virtual currency, category, and the pickup location for student collection. If the selected category is "Clothing," an additional "Sizes" field appears, allowing administrators to specify the available sizes for students to

choose from when redeeming the item. The upload button remains disabled until all required fields are completed.

According to the code snippet shown in Appendix Figure F.23, the system checks again to make sure none of the required fields are empty after the admin submits those details. If validation is successful, the inserted data will be inserted in the database. Additionally, it also adds size information if the admin sets the item category to 'clothing'.

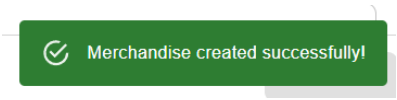


Figure 4.88. *Snackbar Upon Merchandise Creation*

Upon successful submission, a snackbar notification will appear to confirm that the merchandise item has been created, as shown in the figure above.

Through the merchandise item listing page, the admin can select the item they want to manage. By selecting an item from the list, the admin can view or edit its details through a prefilled form as shown in Appendix Figure F.24. The form displays fields such as the merchandise image catalogue with an upload button, name, description, price in virtual currency, category-related details, and pickup location.

```
const hasChanges = useMemo(() => {
    if (!originalData || !formData) return false;

    return Object.keys(formData).some(key => {
        if (Array.isArray(formData[key])) {
            if (formData[key].length !== originalData[key]?.length) return true;
            return !formData[key].every((item, index) => item === originalData[key][index]);
        }

        if (key === "diamondsToRedeem") {
            return Number(formData[key]) !== Number(originalData[key]);
        }

        return formData[key] !== originalData[key];
    });
}, [originalData, formData]);
```

Figure 4.89. *Function for Enabling Save Changes button*

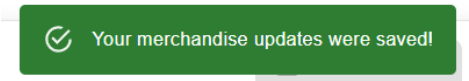


Figure 4.90. *Merchandise Updates Success Snackbar*

At the bottom-right of the form, there is a "*Save Changes*" button, which is initially disabled and only becomes enabled when the admin makes changes to the selected item, as handled by the method shown in the code snippet above.

If any changes are made by the admin to the merchandise details, the submit button will be enabled, and a snackbar will appear to indicate that the updates to the selected merchandise have been saved, as shown in the figure above.

The admin can decide whether a merchandise item is currently available by toggling its availability status. As shown in Appendix Figure F.24, this can be done using the button located at the top right of the screen. The button label dynamically changes based on the item's status, displaying "Mark as Available" when the item is unavailable, and "Mark as Unavailable" when it is currently available. When marked as unavailable, the item will be hidden from students and cannot be redeemed.

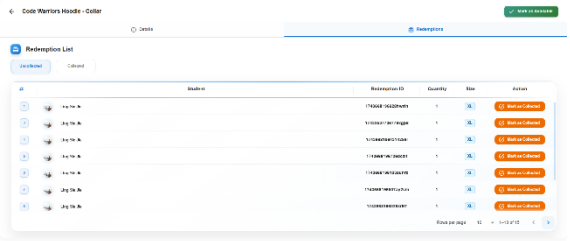


Figure 4.91. *Merchandise Redemption List (Uncollected)*

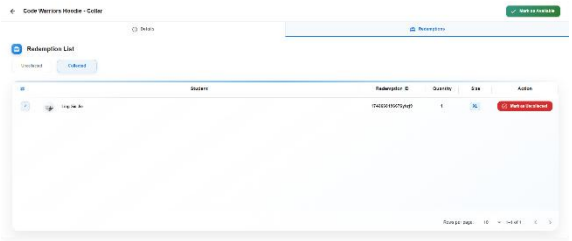


Figure 4.92. *Merchandise Redemption List (Collected)*

In addition, the administrator is responsible for handling merchandise item redemptions. If a student comes to pick up an item, then the administrator can go into the redemption tab, choose the merchandise item, and compare the ID to the system records. Once the verification is successful, if the administrator clicks the action button, as shown in Figure 4.90, associated with the corresponding redemption entry, the redemption is marked as collected.

Based on the Figure 4.92, the system also allows reverting this action in the case where the administrator has mistakenly marked a redemption as collected. If the administrator selects the "Mark as Uncollected" button next to the redemption entry, they can mark the redemption

as uncollected. This feature is flexible and ensures the ability to keep accurate records of merchandise collections.

4.5.2.5. Leaderboard Module

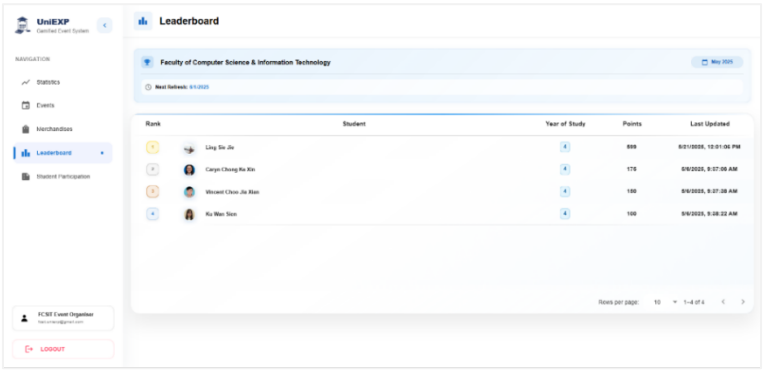


Figure 4.93. *Admin's Leaderboard Module*

The Leaderboard module, illustrated in Figure 4.93, enables faculty-specific administrators to monitor the real-time performance of students within their respective faculties. For example, an FCSIT admin can only view the leaderboard of FCSIT students. This module displays the current month's ranking based on points earned, along with the timestamp of the latest updates for each student entry. It provides an overview of student engagement and competition, supporting gamification objectives. Leveraging Firebase's real-time listener, the leaderboard reflects ranking changes instantly as new data is recorded, ensuring that admins always have access to the most up-to-date performance metrics.

4.5.2.6. Student Participation Leaderboard Module

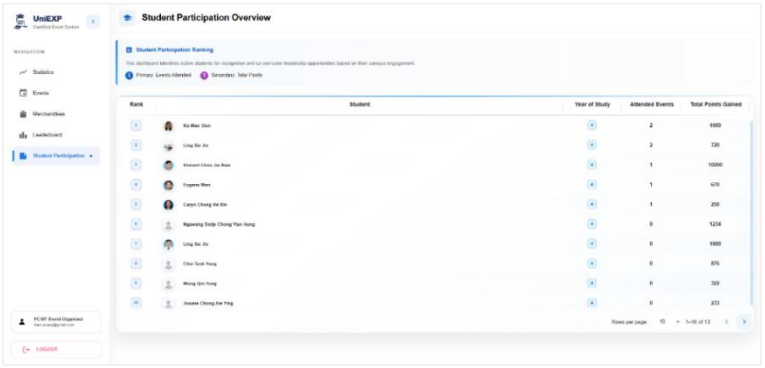


Figure 4.94. *Student Participation Summary Module*

The Student Participation Leaderboard module allows administrators to track and evaluate students' overall participation throughout their usage of the UniEXP application. This module displays a cumulative ranking based primarily on the total number of events a student has attended, with total points gained serving as a secondary ranking factor. In cases where students have attended the same number of events, the system will use their total points to determine the final ranking order. This summary serves as a valuable reference for faculty administrators in identifying and awarding the most active students with recognitions based on their long-term co-curricular involvement and contributions.

4.6. Summary

This chapter presents the comprehensive implementation of UniEXP, a gamified university event management system tailored for UNIMAS students and event administrators. It outlines the installation and configuration of development tools, Firebase backend setup, and the integration of role-based access control. The chapter details both the student mobile app and admin web system workflows, covering major modules such as authentication, event management, gamification features (quests, leaderboard, badges), merchandise redemption, and feedback collection.

Chapter 5. TESTING

5.1. Introduction

This chapter discusses on the testing done to measure the performance and reliability of UniEXP, a gamified event management system developed for UNIMAS students and faculty-based event organisers. The primary goal of testing is to ensure that the system functions as intended and fulfils the requirements and expectations defined during the planning and design phases.

There were two main categories of testing which were functional testing and non-functional testing to be conducted during testing phase. Functional testing is checking that each feature works as its specifications dictate, while non-functional testing checks usability, reliability and performance. These tests play a crucial role in identifying issues, ensuring quality, and validating that the application delivers a satisfactory user experience.

5.2. Functional Testing

Functional testing was carried out to verify that each module in the UniEXP system behaves according to the functional requirements outlined during the system design phase.

To conduct functional testing effectively, system testing was employed. System testing involves testing the complete and integrated system as a whole, rather than in isolated units. This approach was chosen to validate the interactions between various modules and confirm that the overall system meets the specified requirements.

Each feature listed in the feature list was treated as a test case, and test scenarios were designed based on the expected behaviour of the system. The tests were executed on both the mobile application (for students) and the web-based portal (for event organisers) to ensure functional integrity across platforms. The detailed functional test cases, including descriptions,

expected results, and actual outcomes, are documented in the Appendix. These can be referred to from Table G.1 to Table G.29 for full traceability of the tested features.

5.3. Non-Functional Testing

Non-functional testing was conducted to evaluate aspects of the UniEXP system that go beyond individual features, including usability, satisfaction, and overall user experience. This testing focused on validating that the system meets quality standards in terms of user expectations and operational effectiveness. To carry out non-functional testing, **User Acceptance Testing (UAT)** was performed using evaluation form designed to collect both quantitative and qualitative feedback. The goal was to determine whether the system is ready for real-world deployment from the perspective of both student users and administrators.

5.3.1. Mobile App Evaluation by Student Users

The mobile app was evaluated by **30 UNIMAS undergraduate students** who had hands-on experience with the system.

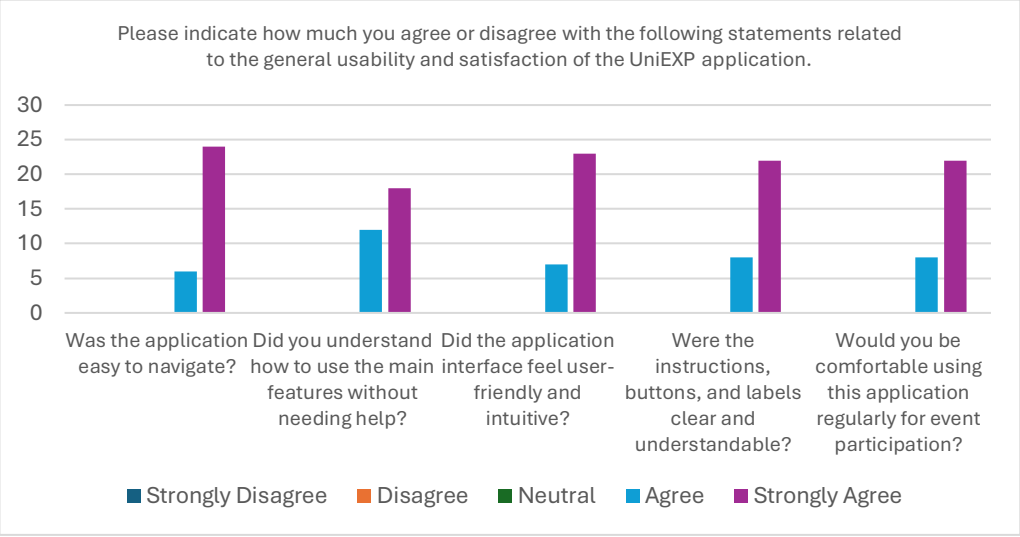


Figure 5.1. *Student responses on general usability and satisfaction of the UniEXP app*

Based on Figure 5.1, the evaluation shows that the UniEXP program has good usability and user satisfaction. Of the 30 responders, 24 (80%) strongly agreed with the statement, "Was

the application easy to navigate?" and 6 (20%) agreed. Twelve respondents (40%) agreed, and eighteen respondents (60%) strongly agreed with the statement, "Did you understand how to use the main features without needing help?" Seven (23.3%) of the respondents agreed that the application's UI is easy to use and intuitive, and 23 (76.7%) highly agreed. Likewise, 8 (26.7%) and 22 (73.3%) highly agreed that the labels, buttons, and directions were clear. Finally, 22 (73.3%) strongly agreed and 8 (26.7%) agreed that they would feel at ease utilising the app on a regular basis to participate in events.

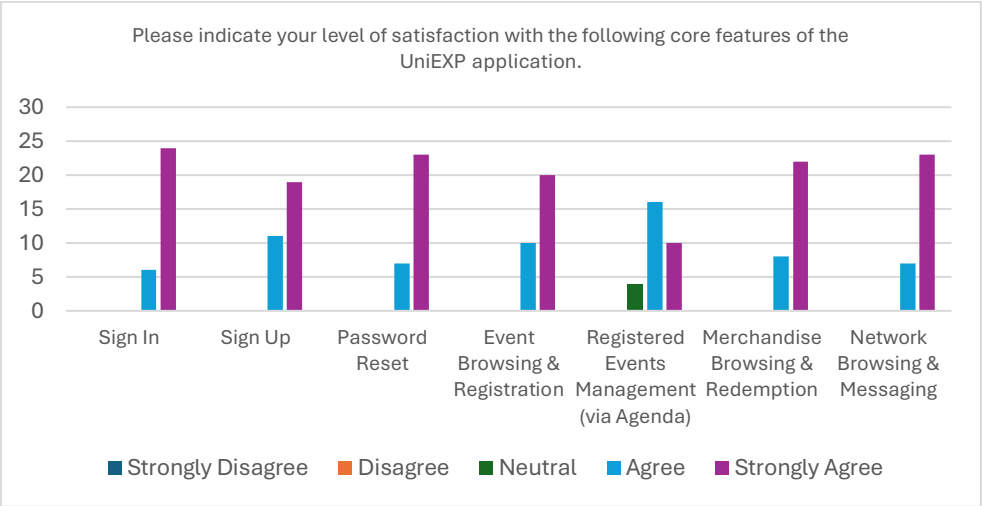


Figure 5.2. *Student satisfaction levels with core features of the UniEXP application*

Based on Figure 5.2, the evaluation indicates that the core features of the UniEXP application received a high level of user satisfaction. For the Sign In feature, 24 respondents (80%) strongly agreed, and 6 respondents (20%) agreed that the feature met their expectations. Regarding the Sign-Up process, 19 participants (63.3%) strongly agreed while 11 (36.7%) agreed. The Password Reset feature also scored highly, with 23 users (76.7%) strongly agreeing and 7 (23.3%) agreeing on its effectiveness. For the Event Browsing & Registration function, 20 (66.7%) strongly agreed and 10 (33.3%) agreed that it worked smoothly. The Registered Events Management (via Agenda) feature received slightly more varied responses, with 10 users (33.3%) strongly agreeing, 16 (53.3%) agreeing, and 4 (13.3%) remaining neutral. As for Merchandise Browsing & Redemption, 22 users (73.3%) strongly agreed and 8 (26.7%) agreed

with its usability. Lastly, the Network Browsing & Messaging feature was well received, with 23 respondents (76.7%) strongly agreeing and 7 (23.3%) agreeing that it functioned effectively and met their needs.

Figure 5.3. *Student responses on whether points or rewards motivated event participation*

Based on Figure 5.3, all 30 respondents who used UniEXP thought earning rewards was enough to make them join and participate in more events. Seeing all the positive responses, gamifying the app with points and rewards increased how engaged users were. It means users were motivated to take part in campus events due to the visible appreciation or awards they got. When users are satisfied with gamification, it highlights that the idea has worked with the application.

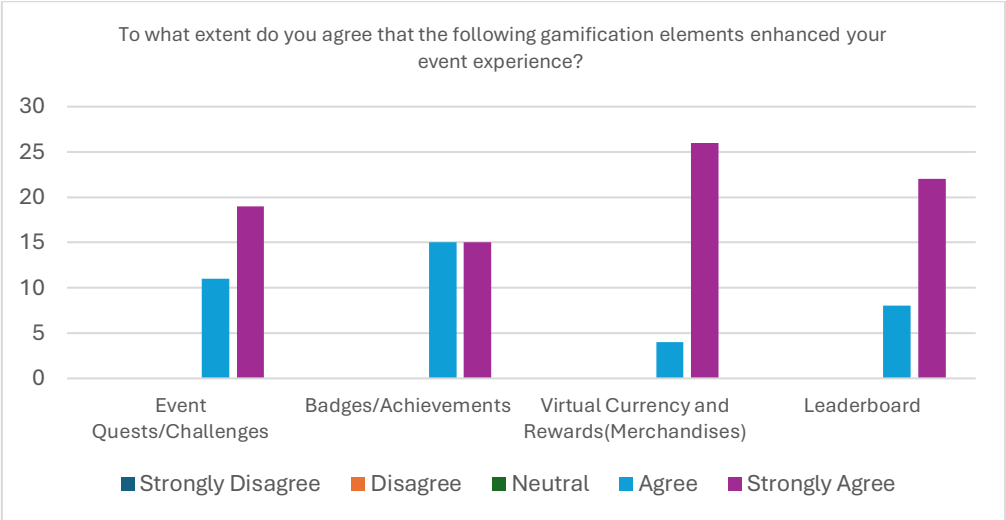


Figure 5.4. *Student agreement on the impact of gamification elements on event experience*

According to Figure 5.4, the introduced gamification in UniEXP helped respondents enjoy their event experiences more. For the Event Quests/Challenges feature, 11 participants

(36.7%) agreed and 19 (63.3%) strongly agreed that it enhanced their engagement. In terms of Badges/Achievements, 15 (50%) agreed and another 15 (50%) strongly agreed. The Virtual Currency and Rewards feature received the most positive feedback, with 26 respondents (86.7%) strongly agreeing and only 4 (13.3%) agreeing. Lastly, for the Leaderboard, 8 (26.7%) agreed while the majority, 22 (73.3%), strongly agreed. These responses show that all gamification features were well-received, with Virtual Currency and Rewards being the most impactful in enhancing the overall event experience.

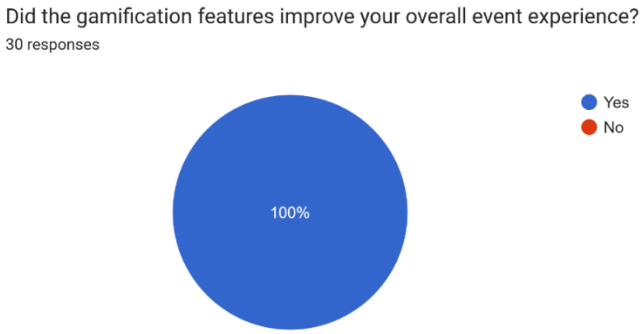


Figure 5.5. *Student responses on the impact of gamification features on overall event experience*

Based on Figure 5.5, all 30 respondents (100%) agreed that the gamification features enhance their overall event experience. A very positive reception can be seen by everyone towards the gamification elements present in the UniEXP application is shown through this unanimous response. The points system, rewards and interactive events (badges, leaderboards, things like that) were used by users to add one more motivating and fun element when participating in those events. This implies that therefore, other than making the experience more engaging, these features fostered continued use of the platform. The resulting application turns out to be more dynamic and fun for students which displays that gamification was a successful addition to the application.

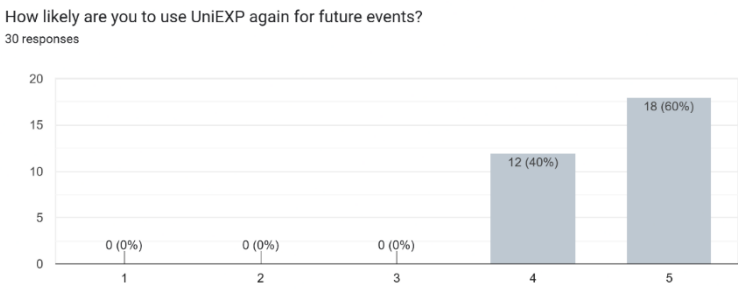


Figure 5.6. *Student likelihood of using UniEXP for future events.*

From Figure 5.6, it can conclude that all 30 respondents are positive to use the UniEXP application again in future events after using it. In particular, 12 people (40%) said they would use UniEXP again as likely and 18 (or 60%) said they were very likely to use it. Given this feedback, respondents were not only able to find the application useful for the managing of university events but also excited to keep engaged throughout the events they participated. Thus, it means that the application met their needs and expectations.

Table 5.1. *Student suggestions for additional features or improvements to the UniEXP application*

Question: Is there anything you think should be added or improved?
• Advice: Add more quest types to make the event more engaging.
• Hope there are more quest variants
• Hope there is passport stamping contest introduce in future quest type
• More quest types and advisable to include AI
• Also include system quests (not only focus on event-specific)
• More quest types in future
• Rewards-based and shareable achievement badges
• AI-powered event recommendation based on user historical data
• Should integrate AI for multiple purposes, such as AI chatbot etc.
• System-based quests (generic platform quests, not restrict to event-specific)

Overall, Table 5.1 highlights strong interest in enhancing the quest system within UniEXP. Most users suggested adding more diverse and engaging quest types, including

system-based or non-event-specific quests to keep users active beyond individual events. Several respondents also recommended incorporating AI features, such as AI-powered event recommendations and AI chatbots, to improve personalization and user support. Additionally, ideas like passport stamping contests, achievement badges, and shareable rewards were proposed to further boost user engagement and motivation. These suggestions point to a clear desire for deeper gamification and intelligent features in future updates.

5.3.2. Admin System Evaluation by Staff Representative

For the web-based admin system, Miss Diana Tracy anak Delim, the Assistant Registrar in the administrative unit of the Centre for Student Services (PKP), was invited to conduct a face-to-face evaluation. During the session, a walkthrough of UniEXP admin panel was conducted, followed by the completion of an evaluation form tailored to the administrative functionalities.

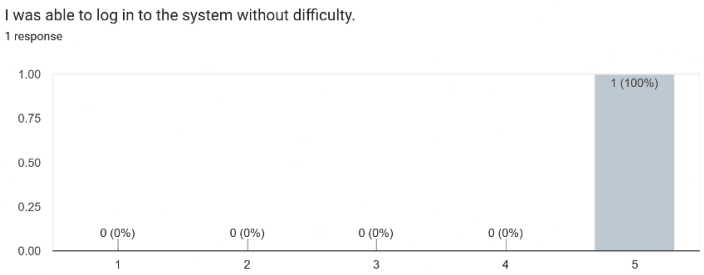


Figure 5.7. *Admin response on ease of login and password reset in the UniEXP web system.*

According to the feedback in Figure 5.7, the respondent strongly agreed that they could log in to the system easy. It implies that staff can use the UniEXP admin panel login without experiencing access problems or confusion. The system's way for authorizing users makes it easy for them to access the administrative area without encountering problems.

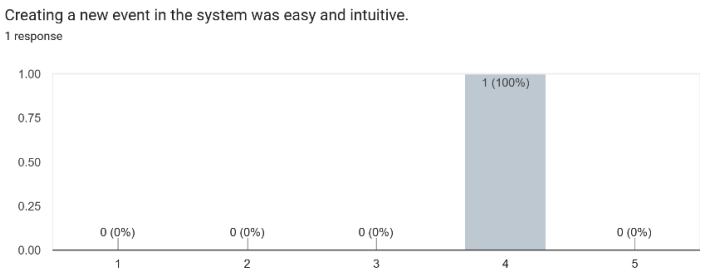


Figure 5.8. *Admin response on ease and intuitiveness of event creation in UniEXP web system*

Based on Figure 5.8, the respondent gave a rating of 4 (Agree) for "Creating a new event in the system was easy and intuitive" statement. It shows that the process of creating events in the UniEXP admin panel is organized and easy for users. It is easy for administrative users to set up events using the interface. On the other hand, more steps can be taken to strengthen the system. The respondent thought it would be better if the system could identify the event's location using the name, so users don't have to choose it themselves on a map. Also, although UniEXP mainly helps with physical events, allowing members to host online events would make the site more flexible and useful for many kinds of events.

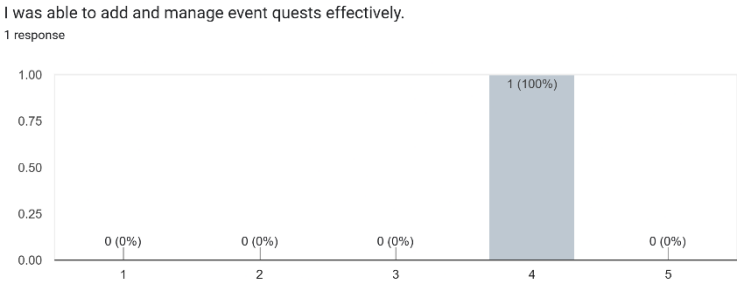


Figure 5.9. *Admin response on effectiveness of adding and managing event quests in the UniEXP web system.*

For the statement "I was able to add and manage event quests effectively," the respondent gave a rating of 4 (Agree), as shown in Figure 5.9. It demonstrates a good experience using the event quest management tool in the UniEXP admin panel. The features work well and help administration users create and monitor quests for events. Although it could be better in specific areas, the system now does a good job helping with quest activities.

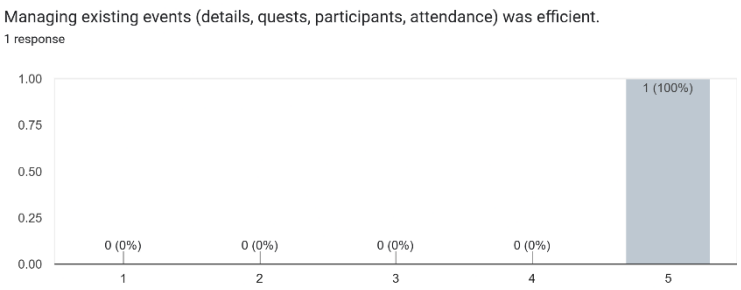


Figure 5.10. *Admin response on efficiency of managing events in UniEXP web system.*

According to Figure 5.10, the respondent strongly agreed that managing existing events (details, quests, participants and attendance) in UniEXP admin panel was efficient. It means that working with event management in the panel is highly rewarding. Everything works well and users can easily access complete event information, quests, who is coming an attendance. It shows that the system works well and is simple to use for those handling ongoing events.

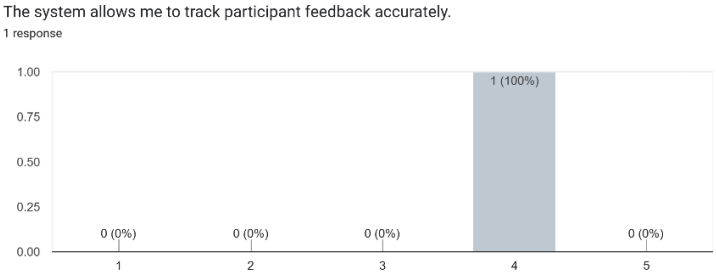


Figure 5.11. *Admin response on accuracy of tracking participant feedback*

Based on Figure 5.11, the respondent gave a rating of 4 (Agree) on the statement “The system allows me to track participant feedback accurately,” indicating that the feature is functioning well and provides reliable access to participant insights.

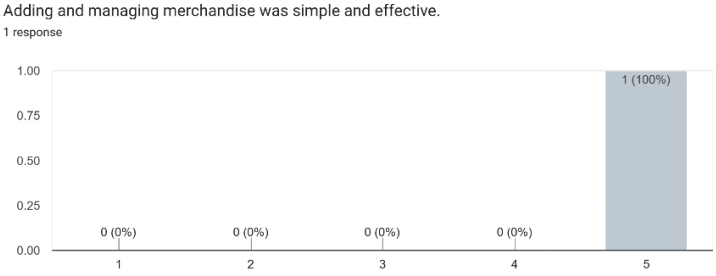


Figure 5.12. *Admin response on ease and effectiveness of managing merchandise.*

The respondent indicated that it was easy and effective to manage their merchandise by giving it a rating of 5 (Strongly Agree). It shows that managing merchandise items through the UniEXP admin panel is very satisfactory. The way merchandise is managed and the interface used appeared straightforward, efficient and easy for administrative staff to operate. This brings out that the system is well-made and delivers the features users want.

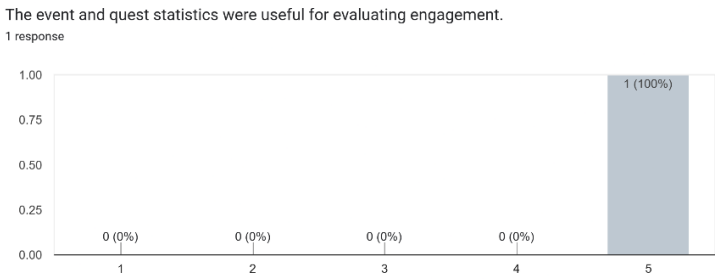


Figure 5.13. *Admin response on usefulness of event and quest statistics for evaluating engagement*

Based on Figure 5.13, the respondent strongly agreed that the event and quest statistics in UniEXP admin panel assisted them to understand how engaged the users are, showing that they were very pleased with how the admin panel helps them understand the activity of their users. All the statistics such as event and quest satisfaction, overall attendance tracking, various event types and frequency over the past year, and a breakdown of quest creation and completion data contributed greatly toward identifying the engagement trends of students. They help a lot when it comes to making data-supported decisions and preplanning the next quests and events.

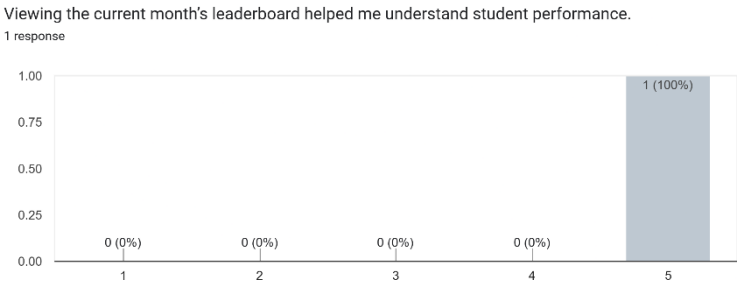


Figure 5.14. *Admin response on usefulness of viewing the current month's leaderboard for understanding student performance*

According to Figure 5.14, the respondent chose a rating of 5 (Strongly Agree), agreeing that the leaderboard in the admin panel provides a clear and effective overview of student performance. It means that the leaderboard is a helpful way for administrators to track and review how students are doing. Being able to see real-time rankings shows how involved students are and makes it easier for administrators to both encourage participation and identify the top achievers.

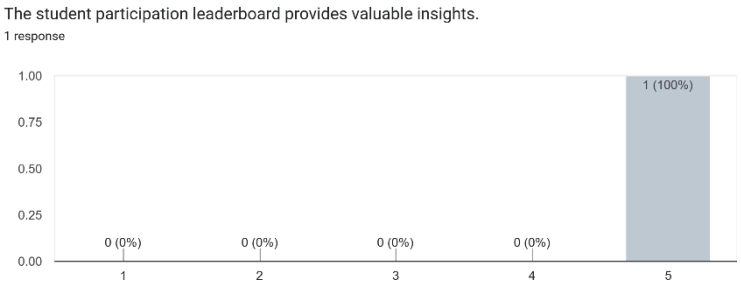


Figure 5.15. *Admin response on the value of the student participation leaderboard for gaining insights*

In response to "The student participation leaderboard provides valuable insights," the participant chose a 5 (Strongly Agree) rating, signalling that they fully support this feature. By highlighting how often students use the application to participate in events, the leaderboard ensures that the objective of increased campus activity is achieved. Quest points are used as an additional aspect that helps determine how involved someone is. By having a full picture of student activity, students are encouraged to remain engaged, and faculty members can select and honour top performers within the university.

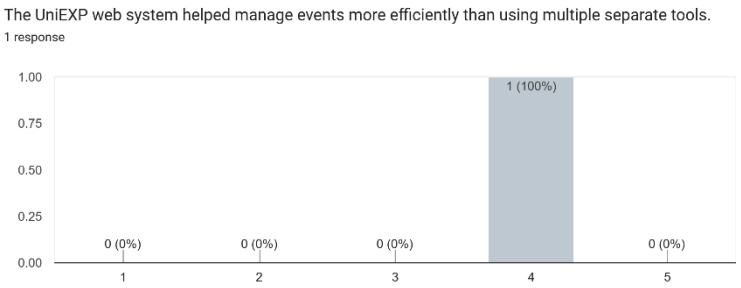


Figure 5.16. *Admin response on system efficiency compared to using multiple separate tools*

Based on Figure 5.16, the respondent marked the statement “The UniEXP web system helped manage events more efficiently than using multiple separate tools.” with a rating of 4 (Agree). This demonstrates that people have a favourable view of how the system supports in managing events. It looks like UniEXP gathers most of the necessary functions for events on the platform, making it unnecessary for those in charge to use separate tools like spreadsheets, chat applications or separate registration forms. It combines several services, allowing the management process to be smoother, neater and easier to handle.

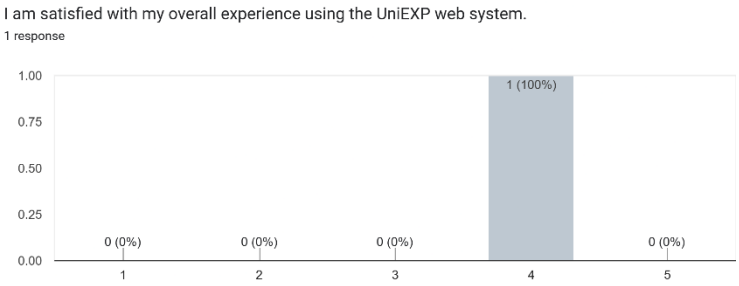


Figure 5.17. *Admin overall satisfaction with the UniEXP web system*

For the sentence “I am satisfied with my overall experience using the UniEXP web system,” the respondent rated it a 4 (Agree), as seen in Figure 5.17. It indicates that the respondent is happy with the system because its basic functions live up to their expectations and meet the needs of faculty-based administrators. Although the system is generally appreciated, the rating hints at a few places where making further improvements could improve the way users enjoy it. Regardless, the system gives a strong and reliable tool for organising university events and similar activities.

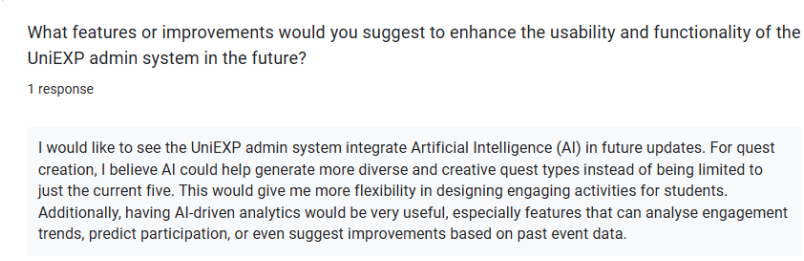


Figure 5.18. *Admin suggestions for improving usability and functionality of the UniEXP web system*

To the response of Figure 5.18, the respondent shows how AI can be used to improve the UniEXP admin system in the future. The use of AI to do so would satisfy this need for more flexibility and creativity with student engagement outside of the regular quest formats. In addition, AI based analytics are also recommended by the respondent, so that the system can provide them with more intelligent interpretation of their data, e.g. trend analysis, forecasting of attendee numbers or automatic recommendations.

5.4. Summary

This chapter outlines the rigorous testing processes applied to ensure UniEXP's functionality, reliability, and user satisfaction. Functional testing confirmed that core features across both student and admin platforms operated according to specifications, with all test cases successfully passed. Non-functional testing involved real user evaluations: 30 UNIMAS students and an administrative staff member. Results showed strong usability, positive feedback on gamification elements, and high user satisfaction. Suggestions collected indicate interest in more diverse quests and AI-powered features. Overall, testing validated that UniEXP is reliable, user-friendly, and effective in enhancing event participation through gamification.

Chapter 6. CONCLUSION AND FUTURE WORK

6.1. Introduction

This chapter concludes the development journey of UniEXP, a gamified university event management system tailored for UNIMAS. It summarizes the accomplishments of the project, reflects on its limitations, and outlines potential enhancements for future iterations. The chapter also revisits the project objectives and evaluates how well they have been achieved.

6.2. Objective Achievements

The UniEXP project was guided by three primary objectives. The following discusses the degree to which each objective was fulfilled:

Objective 1

To analyse existing event management systems that incorporate gamification elements.

- This objective was successfully achieved through a comprehensive literature review and competitive analysis of existing gamified event management platforms such as Goosechase, Whova, and Eventzee. Key gamification strategies such as point-based rewards, leaderboard systems, badge achievements, and quest-like participation were studied and selectively integrated into UniEXP. A notable enhancement in UniEXP is the introduction of a rewards-based mechanism using virtual currency (diamonds), which students can earn by completing event-related quests and use to redeem merchandise.

Objective 2

To design and implement the UniEXP system, a digital platform for undergraduate students and administrators.

- This objective was fully met. A React Native mobile application was developed for UNIMAS students to browse events, register, complete gamified quests, and track their progress on leaderboards. The app integrates gamification features such as achievement

badges, virtual currency, and event-specific quests to enhance student engagement. In parallel, a web-based dashboard was built for event organisers and faculty administrators to manage event details, assign gamified quests, generate dynamic QR codes for attendance, monitor participant engagement, and track leaderboard performance. These two platforms work in tandem to create a seamless and engaging experience for both students and administrators.

Objective 3

To test and evaluate the event management system for usability and operational efficiency.

- This objective was successfully achieved through the execution of two fundamental testing types: functional testing and non-functional testing. For functional testing, system testing was conducted for each feature listed in the feature list to ensure proper functionality on both the Expo Go environment and the exported APK file. For non-functional testing, User Acceptance Testing (UAT) was carried out with 30 UNIMAS undergraduate students using the mobile application, while Miss Diana participated in testing the admin web system. This process provided valuable feedback on usability, design clarity, and overall system performance. The evaluations confirmed that the system met its functional requirements and demonstrated operational efficiency suitable for university-wide usage.

6.3. Project Limitations

Despite the system's functional success, several limitations were identified during the development and testing phases:

1. **Fixed Quest Types**

   The current version of UniEXP only supports a limited set of five predefined, event-specific quest types. This restricts organisers from adding more dynamic or creative quest formats without manual intervention or customization.

2. **Lack of AI Integration**

   Although AI, machine learning, and deep learning technologies offer potential enhancements for personalization, quest generation, and feedback analysis, they are not yet integrated into the current system. Features such as AI-driven recommendations, automated sentiment analysis, and smart quest suggestions remain future possibilities.

3. **No Support for Team-Based Registrations**

   At present, the system only allows individual event registration. It does not support team formation or group-based participation, which limits its applicability for events that require participants to register and compete in teams (e.g., hackathons, sports tournaments, or group competitions).

4. **Quest Scope Limited to Event-Specific Activities**

   All quests in the current version are designed specifically for individual events. There is no support for platform-wide generic quests that could encourage continuous student engagement beyond a single event or across multiple events.

5. **Simplified Admin Tools for Quest and Reward Management**

   Admin functionalities for defining quests, managing rewards, and analysing participation are relatively basic. Organisers must manually define each quest and reward structure without advanced analytics or assistance.

6.4. Conclusion

UniEXP successfully bridges the gap between traditional university event management methods and modern digital engagement strategies by incorporating gamification into student event participation. The project provides a practical, scalable, and engaging solution that benefits both students and faculty. The system not only modernises event attendance tracking but also encourages student involvement through competition, rewards, and recognition.

Through methodical development using Feature-Driven Development (FDD) and multiple testing cycles, UniEXP has demonstrated strong usability, functional completeness, and reliability. The feedback received from student testers and faculty organisers reinforces the platform's relevance and value in a university setting.

6.5. Future Work

The current UniEXP platform has proven effective in digitalising and gamifying event participation among UNIMAS students. However, there are many opportunities to advance its features through the integration of modern technologies such as artificial intelligence (AI). These enhancements may significantly improve user personalisation, automation, and engagement.

1. **AI-Powered Event Recommendations**

- AI models can be used to analyse each student's event history, preferences, and faculty affiliations to provide personalised event suggestions. This will not only improve user satisfaction but also boost event participation using AI.

2. **Dynamic , AI-Generated Quests for Gamification**

- Currently, UniEXP platform supports five predefined event-specific quests, such as attendance-based, early bird attendance, question and answer (Q&A), networking, and feedback-driven which manually assigned by organisers. While effective, this static range may lead to reduced engagement over time.

  To enhance this:

  (a) AI-Enhanced Event-Specific Quests

  - AI can assist event organisers by suggesting creative and context-relevant quests based on the event category, expected audience, or engagement goals.

Examples include “Take a selfie with the guest speaker and share your key takeaway” and “Passport stamping activity”. These dynamic suggestions would enrich the gamified experience while reducing planning workload for organisers.

(b) Generic, System-Wide Quests (AI-Driven)

- To further increase engagement across the entire platform, AI-generated generic quests can be introduced. These are not tied to a single event, but encourage continuous involvement, such as “Participate in three events this month,” “Invite two friends to join an event,” and “Earn 1000 points from any faculty-related activities within a week.” These quests can be personalised based on user behaviour, and their difficulty or frequency can adapt based on the user's participation history.

This dual-layered quest system—event-specific and platform-wide—enhanced by AI, can maintain long-term user motivation while promoting a more vibrant event culture within the university.

3. Team Formation Support for Group-Based Events

- Currently, UniEXP supports individual-based event registration. However, many university events—such as innovation competitions, esports tournaments, or community service projects—require participants to register as teams. In future, UniEXP can include a team formation module, enabling students to:
  - Create or join teams during event registration.
  - Invite friends using unique team codes or from a contact list.
  - Set roles or responsibilities within the team (e.g., leader, presenter, designer).
  - Track team-based quests progress, especially in gamified competitions.

4. **AI-Powered Feedback Analysis**

- By leveraging Natural Language Processing (NLP), the system can automatically analyse qualitative feedback provided by students' post-event. This sentiment analysis can offer actionable insights to organisers, such as identifying pain points or highly appreciated elements of the event.

5. **AI Chatbot for Assistance and Engagement**

- A chatbot powered by natural language understanding (NLU) can be implemented within the app to answer FAQs, guide students in event selection, remind users about quests, and assist in registration. This improves user support and drives ongoing engagement.

References

About Goosechase. (n.d.). https://goosechase.com/about

Abrahamson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods: Review and analysis. VTT Publications, 112.

Alsawaier, R. S. (2018). The effect of gamification on motivation and engagement. International Journal of Information and Learning Technology, 35(1), 56–79. https://doi.org/10.1108/ijilt-02-2017-0009.

Andreev. (2023). Gamification Examples in Employee Training [2020]. Valamis. https://www.valamis.com/hub/gamification

Bakr, A. A. (2024). Getting Started with React Native: A Beginner's Guide. Medium. https://medium.com/@ahmed.nums345/getting-started-with-react-native-a-beginners-guide-cf63368bb887

Banyte, J., & Gadeikiene, A. (2015). The effect of consumer motivation to play games on video game-playing engagement. Procedia economics and finance, 26, 505-514.

Bello, A. H. (2024). "Eventful": Revolutionizing Event Management through Technology Integration and User-Centered Design. *Saudi Journal of Engineering and Technology*, *9*(03), 173–191. https://doi.org/10.36348/sjet.2024.v09i03.008

Berger, A., Schlager, T., Sprott, D., & Herrmann, A. (2018). Gamified interactions: whether, when, and how games facilitate self–brand connections. 21.

Camus, A. (2022). Introduction to ReactJS: A Guide for Beginners. Microverse. https://www.microverse.org/blog/introduction-to-reactjs-a-guide-for-beginners

Chauhan, A. (2024). Overview of Firebase Backend as a service platform - Anshul Chauhan - medium. *Medium*. https://medium.com/@engrshul/overview-of-firebase-aa8e05710542

Chitroda, H. (2015). A brief history on Gamification. KNOLSKAPE. https://knolskape.com/brief-history-gamification/.

Dale, S. (2014). Gamification: Making work fun, or making fun of work? Sage, 31(2), 82-90.

Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining "gamification". *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, 9-15.

Development and production modes. (n.d.). Expo Documentation. https://docs.expo.dev/workflow/development-mode/

Eisenman, B. (2016). Writing Cross-Platform Apps with React Native. InfoQ. https://www.infoq.com/articles/react-native-introduction/

Elivera, A. & Palaoag, T. D. (2020). Development of Computer Assisted Automotive Technology Training Course through Augmented Reality. *TEST Engineering & Management*, 83, 23369-23377.

Esang, M. O., Johnson, E. A., Attai, K., Inyangetoh, J. A., Dan, E. E., Okonny, K. E., Bardi, I., Bassey, A., & John, A. (n.d.). Exploring Agile Methodology in Developing a Web-Based Result Computation and Transcript System: A Case Study of Federal Polytechnic Ukana. *European Journal of Computer Science and Information Technology*, *12*(4), 1–17. https://doi.org/10.37745/ejcsit.2013/vol12n4117

Eventzee Scavenger Hunt - apps on Google Play. (n.d.). https://play.google.com/store/apps/details?id=com.freezetag.eventzee

Feature Driven Development (FDD). (2023). ProductPlan. https://www.productplan.com/glossary/feature-driven-development/

Firebase Introduction - Javatpoint. (n.d.). www.javatpoint.com. https://www.javatpoint.com/firebase-introduction

Freeze Tag, Inc. (2022). Eventzee Announces Partnership with Fundraising Platform Provider, Qgiv. GlobeNewswire News Room. https://www.globenewswire.com/en/news-release/2022/01/20/2370417/23476/en/Eventzee-Announces-Partnership-with-Fundraising-Platform-Provider-Qgiv.html

Goosechase - apps on Google Play. (n.d.). https://play.google.com/store/apps/details?id=com.goosechaseadventures.goosechase

Guittard, C., Schenk, E., & Burger-Helmchen, T. (2015). Crowdsourcing and the Evolution of a Business Ecosystem. Advances in Crowdsourcing, 49–62. https://doi.org/10.1007/978-3- 319-18341-1_4

Hamari, J, Koivisto, J & Sarsa, H. (2014). Does Gamification Work? -- A Literature Review of Empirical Studies on Gamification. In Proceed-ings of the 2014 47th Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 6–9 January 2014; pp. 3025–3034.

Heller, M. (2022, July 8). What is Visual Studio Code? Microsoft's extensible code editor. InfoWorld. https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html

Herbert, D. (2022). What is React.js? Uses, Examples, & More. What is React.js? Uses, Examples, & More. https://blog.hubspot.com/website/react-js

Högberg, J., Hamari, J., & Wästlund, E. (2019). Gameful Experience Questionnaire (GAMEFULQUEST): an instrument for measuring the perceived gamefulness of system use. User Modeling and User-Adapted Interaction, 29(3), 619–660. https://doi.org/10.1007/s11257-019-09223-w

Huotari, K., & Hamari, J. (2012). Defining gamification. Proceeding of the 16th International 45 Academic MindTrek Conference on - MindTrek '12, 17–22. https://doi.org/10.1145/2393132.2393137

Kale, S. (2023). Firebase vs Supabase: Battle of the BAAS - Sagar Kale - Medium. *Medium*. https://medium.com/@thesagarkale/firebase-vs-supabase-2c2541ae9bd

Kalogiannakis, M., Papadakis, S., Zourmpakis, A. I. (2021). *Gamifcation in science education. A systematic review of the literature.* Education Sciences, 11(1), 22.

Karamchandani, M., Purswani, J., Dsouza, L., Rakhunde, A. I. P. H., Bhoyar, P. S. (2024). Event Management App. *International Journal of Research Publication and Reviews, 5*(3), 4662-4669. https://ijrpr.com/uploads/V5ISSUE3/IJRPR23966.pdf

Keserű, K. (2024). Scheduling tasks on the Google Cloud Platform (GCP). Medium. https://medium.com/@keseruk/scheduling-tasks-on-the-google-cloud-platform-gcp-8a4e3daf0f9a

Kiryakova, G., Angelova, N. & Yordanova, L. (2014). GAMIFICATION IN EDUCATION. 9th International Balkan Education and Science Conference, Edirne, Turkey. https://www.researchgate.net/publication/320234774_GAMIFICATION_IN_EDUCATION

Lubbers, C. A. & Joyce, T. A. (2014). Promoting campus activities: Encouraging studentparticipation. Quarterly Review of Business Disciplines, 1(1), 1-14.

Madan, A. & Lohani, V. J. (2024). Leveraging Event Gamification for Event Planning and Creating a Memorable Experience. In Kulshreshtha, S. K. & Webster, C (Eds.), *New Technologies in*

Virtual and Hybrid Events (pp. 198-227). Premier Research Source. https://doi.org/10.4018/979-8-3693-2272-7.ch011

Markopoulos, A. P., Fragkou, A., Kasidiaris, P. D., & Davim, J. P. (2015). Gamification in engineering education and professional training. International Journal of Mechanical Engineering Education, 118-131.

Mazarakis, A. (2021). Gamification Reloaded: Current and Future Trends in Gamification Science. i-com, 20(3), 279-294. https://doi.org/10.1515/icom-2021-0025.

Moedano, K. (2024). Expo Go vs Development Builds: Which should you use? Expo Blog. https://expo.dev/blog/expo-go-vs-development-builds

O'Connell, A., Tomaselli, P. J., & Stobart-Gallagher, M. (2020). Effective use of virtual gamification during covid-19 to deliver the ob-gyn core curriculum in an emergency medicine resident conference. Cureus. https://doi.org/10.7759/cureus.8397

Palmer, S. R. & Felsing, J. M. (2002). A Practical Guide to Feature-Driven Development.

Prasad, K. D. V. (2021). Gamification and its applications. Journal of Business Strategy Finance and Management, 3(1–2), 04–07. https://doi.org/10.12944/jbsfm.03.01-02.02

Press & Media - Whova. (n.d.). Whova. https://whova.com/press/

Sebastian, N. (2024). Learn React – a handbook for beginners. freeCodeCamp.org. https://www.freecodecamp.org/news/react-for-beginners-handbook/

Sharma, S., Sarkar, D., & Gupta, D. (2012). Agile Processes and Methodologies: A Conceptual Study. International Journal on Computer Science and Engineering, 4(5), 892–898.

Sisson, A. D., & Whalen, E. A. (2021). Exploratory study on the perceptions of event gamification on positive behavioral outcomes. *Journal of Hospitality and Tourism Insights*, *5*(5), 822–841. https://doi.org/10.1108/jhti-04-2021-0085

Trowler, V. (2010). Student engagement literature review. *The Higher Education Academy,* 1-15. https://www.heacademy.ac.uk/sites/default/files/studentengagementliteraturereview_1.pdf

Vaghela, B. (2023). The benefits of using React Native for mobile development. IEEE Computer Society. https://www.computer.org/publications/tech-news/trends/benefits-of-react-native

W3Schools.com. (n.d.). https://www.w3schools.com/react/react_intro.asp

Welbers, K., Konijn, E. A., Burgers, C., de Vaate, A. B., Eden, A., & Brugman, B. C. (2019). Gamification as a tool for engaging student learning: A field experiment with a gamified app. E-Learning and Digital Media, 16(2), 92–109. https://doi.org/10.1177/2042753018818342

What is FDD in Agile? (2021). Planview. https://www.planview.com/resources/articles/fdd-agile/#:~:text=Plan%20by%20Feature,-Enter%3A%20Tasks.&text=Analyze%20the%20complexity%20of%20each,each%20development%20stage%20in%20mind.

Why Visual Studio code? (2021). https://code.visualstudio.com/docs/editor/whyvscode

Yee, N. (2016). The gamer motivation profile: What we learned from 250,000 gamers. In Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play (pp. 2–2).

Yu, X., Cao, J., & Huang, X. (2020). *Student Perception of Mobile Application for Campus Events: A Case Study*. Journal of Education and Technology.

Zainuddin, Z., Chu, S. K. W., Shujahat, M., Perera, C. J. (2020). *The impact of gamification on learning and instruction: A systematic review of empirical evidence.* Educational Research Review, 30, 100326

Appendices

Appendix A: Project Schedule (Gantt Chart)

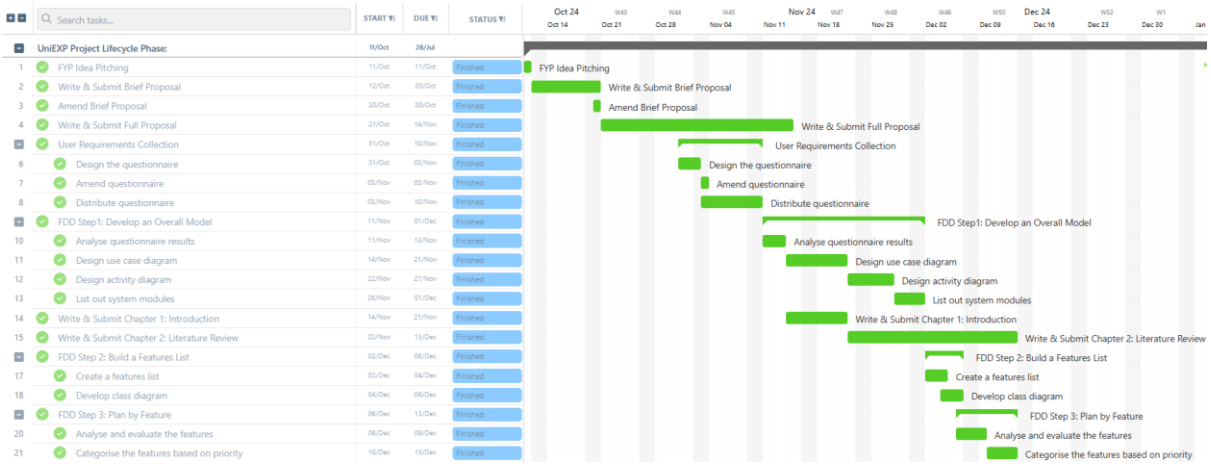


Figure A.1. *Project Schedule – Part 1*

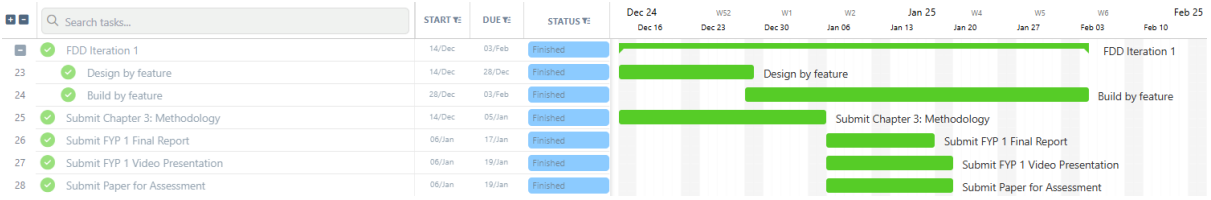


Figure A.2. *Project Schedule – Part 2*

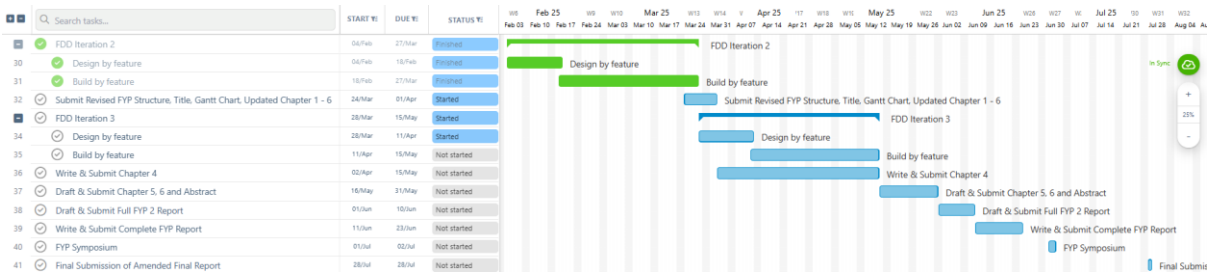
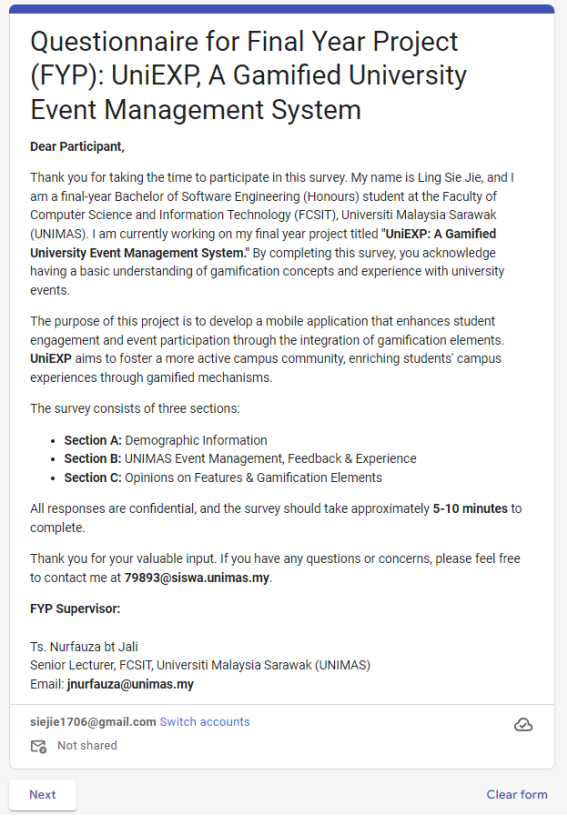


Figure A.3. *Project Schedule – Part 3*

Appendix B: User Requirements Questionnaire Form



Questionnaire for Final Year Project (FYP): UniEXP, A Gamified University Event Management System

Dear Participant,

Thank you for taking the time to participate in this survey. My name is Ling Sie Jie, and I am a final-year Bachelor of Software Engineering (Honours) student at the Faculty of Computer Science and Information Technology (FCSIT), Universiti Malaysia Sarawak (UNIMAS). I am currently working on my final year project titled "**UniEXP: A Gamified University Event Management System.**" By completing this survey, you acknowledge having a basic understanding of gamification concepts and experience with university events.

The purpose of this project is to develop a mobile application that enhances student engagement and event participation through the integration of gamification elements. **UniEXP** aims to foster a more active campus community, enriching students' campus experiences through gamified mechanisms.

The survey consists of three sections:

- **Section A:** Demographic Information
- **Section B:** UNIMAS Event Management, Feedback & Experience
- **Section C:** Opinions on Features & Gamification Elements

All responses are confidential, and the survey should take approximately **5-10 minutes** to complete.

Thank you for your valuable input. If you have any questions or concerns, please feel free to contact me at **79893@siswa.unimas.my**.

FYP Supervisor:

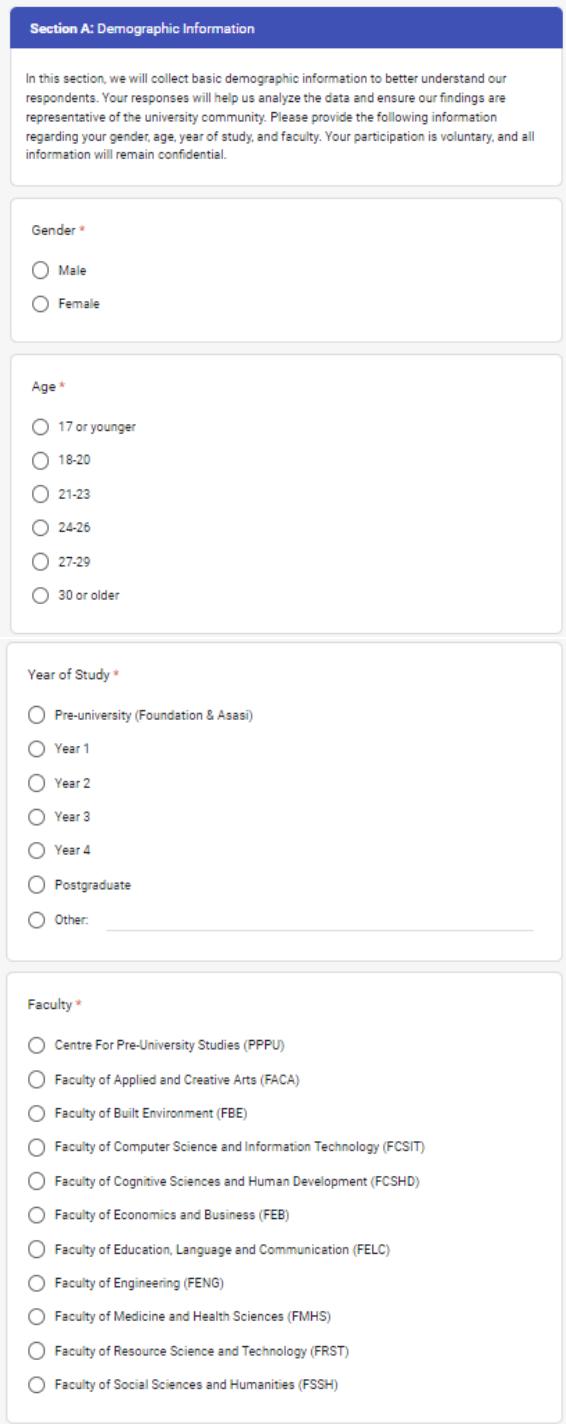
Ts. Nurfauza bt Jali
Senior Lecturer, FCSIT, Universiti Malaysia Sarawak (UNIMAS)
Email: **jnurfauza@unimas.my**

siejie1706@gmail.com Switch accounts

Not shared

Next Clear form

Figure B.1. *Questionnaire Introduction*



Section A: Demographic Information

In this section, we will collect basic demographic information to better understand our respondents. Your responses will help us analyze the data and ensure our findings are representative of the university community. Please provide the following information regarding your gender, age, year of study, and faculty. Your participation is voluntary, and all information will remain confidential.

Gender *

- Male
- Female

Age *

- 17 or younger
- 18-20
- 21-23
- 24-26
- 27-29
- 30 or older

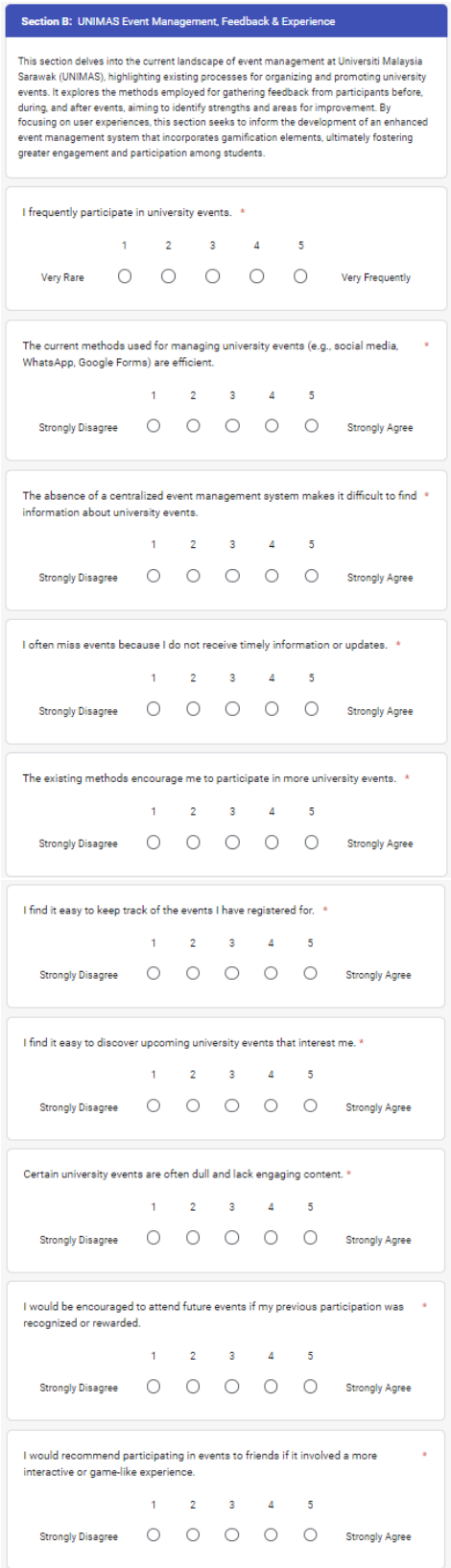
Year of Study *

- Pre-university (Foundation & Asasi)
- Year 1
- Year 2
- Year 3
- Year 4
- Postgraduate
- Other:

Faculty *

- Centre For Pre-University Studies (PPPU)
- Faculty of Applied and Creative Arts (FACA)
- Faculty of Built Environment (FBE)
- Faculty of Computer Science and Information Technology (FCSIT)
- Faculty of Cognitive Sciences and Human Development (FCSHD)
- Faculty of Economics and Business (FEB)
- Faculty of Education, Language and Communication (FELC)
- Faculty of Engineering (FENG)
- Faculty of Medicine and Health Sciences (FMHS)
- Faculty of Resource Science and Technology (FRST)
- Faculty of Social Sciences and Humanities (FSSH)

Figure B.2. *Questionnaire Section A (Demographic Information)*



Section B: UNIMAS Event Management, Feedback & Experience

This section delves into the current landscape of event management at Universiti Malaysia Sarawak (UNIMAS), highlighting existing processes for organizing and promoting university events. It explores the methods employed for gathering feedback from participants before, during, and after events, aiming to identify strengths and areas for improvement. By focusing on user experiences, this section seeks to inform the development of an enhanced event management system that incorporates gamification elements, ultimately fostering greater engagement and participation among students.

I frequently participate in university events. *

	1	2	3	4	5	
Very Rare	○	○	○	○	○	Very Frequently

The current methods used for managing university events (e.g., social media, WhatsApp, Google Forms) are efficient. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The absence of a centralized event management system makes it difficult to find information about university events. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I often miss events because I do not receive timely information or updates. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The existing methods encourage me to participate in more university events. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I find it easy to keep track of the events I have registered for. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I find it easy to discover upcoming university events that interest me. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Certain university events are often dull and lack engaging content. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

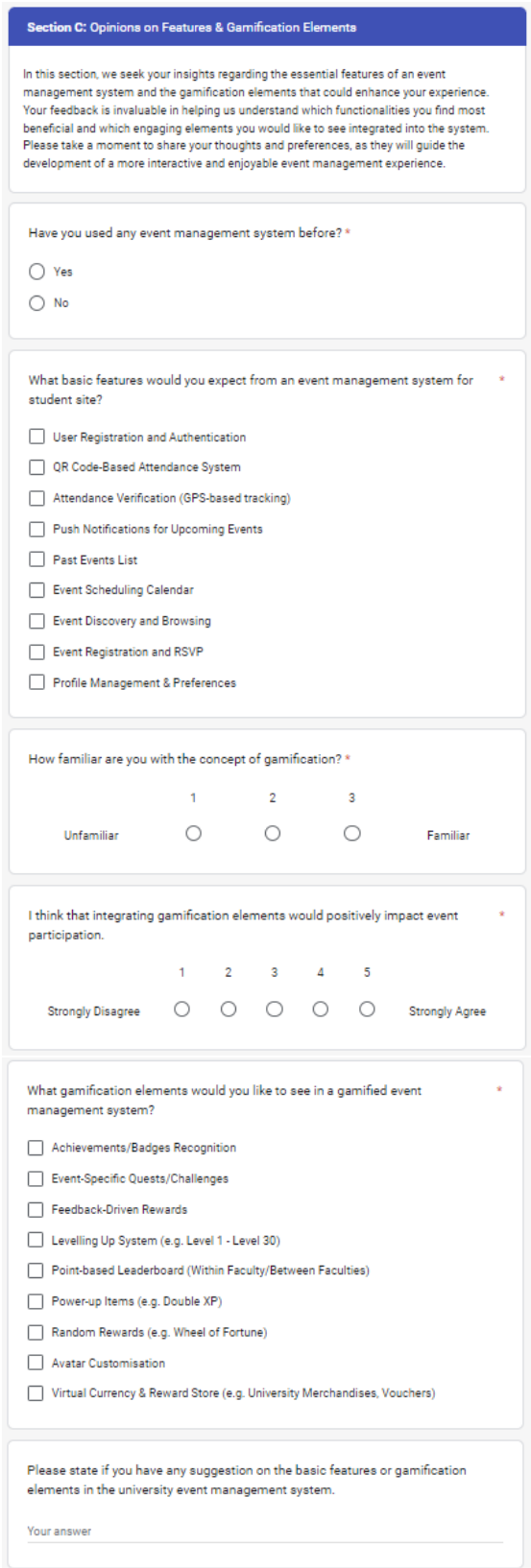
I would be encouraged to attend future events if my previous participation was recognized or rewarded. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I would recommend participating in events to friends if it involved a more interactive or game-like experience. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Figure B.3. *Questionnaire Section B (UNIMAS Event Management, Feedback & Experience)*



Section C: Opinions on Features & Gamification Elements

In this section, we seek your insights regarding the essential features of an event management system and the gamification elements that could enhance your experience. Your feedback is invaluable in helping us understand which functionalities you find most beneficial and which engaging elements you would like to see integrated into the system. Please take a moment to share your thoughts and preferences, as they will guide the development of a more interactive and enjoyable event management experience.

Have you used any event management system before? *

- ○ Yes
- ○ No

What basic features would you expect from an event management system for student site? *

- ☐ User Registration and Authentication
- ☐ QR Code-Based Attendance System
- ☐ Attendance Verification (GPS-based tracking)
- ☐ Push Notifications for Upcoming Events
- ☐ Past Events List
- ☐ Event Scheduling Calendar
- ☐ Event Discovery and Browsing
- ☐ Event Registration and RSVP
- ☐ Profile Management & Preferences

How familiar are you with the concept of gamification? *

	1	2	3	
Unfamiliar	○	○	○	Familiar

I think that integrating gamification elements would positively impact event participation. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

What gamification elements would you like to see in a gamified event management system? *

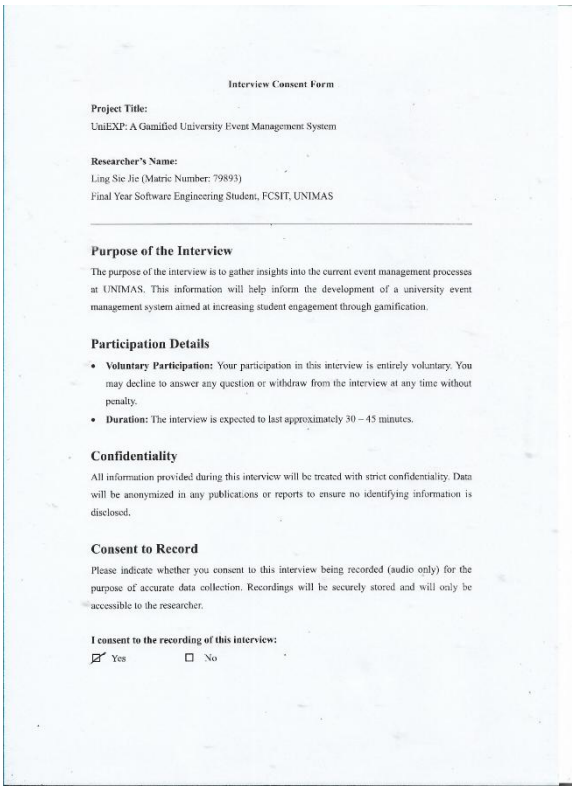
- ☐ Achievements/Badges Recognition
- ☐ Event-Specific Quests/Challenges
- ☐ Feedback-Driven Rewards
- ☐ Levelling Up System (e.g. Level 1 - Level 30)
- ☐ Point-based Leaderboard (Within Faculty/Between Faculties)
- ☐ Power-up Items (e.g. Double XP)
- ☐ Random Rewards (e.g. Wheel of Fortune)
- ☐ Avatar Customisation
- ☐ Virtual Currency & Reward Store (e.g. University Merchandises, Vouchers)

Please state if you have any suggestion on the basic features or gamification elements in the university event management system.

Your answer

Figure B.4. *Questionnaire Section C (Opinions on Features & Gamification Elements)*

Appendix C: Interview Consent Form



Interview Consent Form

Project Title:
UniEXP: A Gamified University Event Management System

Researcher's Name:
Ling Sie Jie (Matric Number: 79893)
Final Year Software Engineering Student, FCSIT, UNIMAS

Purpose of the Interview

The purpose of the interview is to gather insights into the current event management processes at UNIMAS. This information will help inform the development of a university event management system aimed at increasing student engagement through gamification.

Participation Details

- **Voluntary Participation:** Your participation in this interview is entirely voluntary. You may decline to answer any question or withdraw from the interview at any time without penalty.
- **Duration:** The interview is expected to last approximately 30 – 45 minutes.

Confidentiality

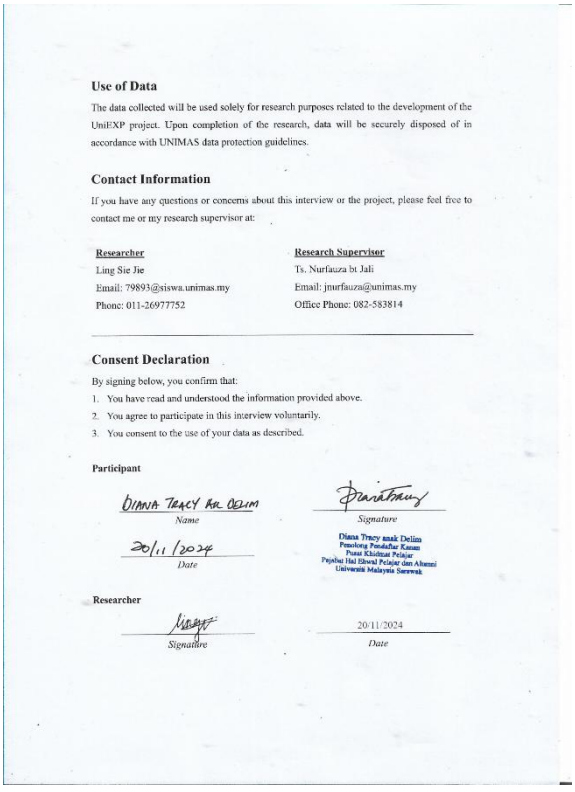
All information provided during this interview will be treated with strict confidentiality. Data will be anonymized in any publications or reports to ensure no identifying information is disclosed.

Consent to Record

Please indicate whether you consent to this interview being recorded (audio only) for the purpose of accurate data collection. Recordings will be securely stored and will only be accessible to the researcher.

I consent to the recording of this interview:
☑ Yes ☐ No

Figure C.1. *Interview Consent Form - First Page*



Use of Data

The data collected will be used solely for research purposes related to the development of the UniEXP project. Upon completion of the research, data will be securely disposed of in accordance with UNIMAS data protection guidelines.

Contact Information

If you have any questions or concerns about this interview or the project, please feel free to contact me or my research supervisor at:

Researcher	**Research Supervisor**
Ling Sie Jie	Ts. Nurfauza bt Jali
Email: 79893@siswa.unimas.my	Email: jnurfauza@unimas.my
Phone: 011-26977752	Office Phone: 082-583814

Consent Declaration

By signing below, you confirm that:

1. You have read and understood the information provided above.
2. You agree to participate in this interview voluntarily.
3. You consent to the use of your data as described.

Participant

DIANA TRACY AK DELIM
Name

Signature

20/11/2024
Date

Diana Tracy anak Delim
Pemakong Pendaftar Kanan
Pusat Khidmat Pelajar
Pejabat Hal Ehwal Pelajar dan Alumni
Universiti Malaysia Sarawak

Researcher

Signature

20/11/2024
Date

Figure C.2. *Interview Consent Form - Second Page*

Appendix D: Use Case Specification

Table D.1. *Use Case Specification for Login*

Use Case ID	UC-001
Use Case Name	Login
Actor	Student, Admin/Event Organiser
Description	This use case allows users to log into the application using their registered credentials. It also provides an option to reset their password if they forget it, which extends this use case.
Pre-Condition(s)	1. The user has already registered an account in the system.
Main Flow	1. The system displays their dedicated login screen with fields for the email and password. 2. The user enters their email and password. 3. The user clicks the "Login" button. [A1] 4. The system verifies the email and password with Firebase Auth. 5. If the credentials are valid, the user is successfully logged in. 6. The system redirects the user to their dedicated dashboard.
Alternative Flow	[A1]: Invalid credentials 1. The system finds the credentials invalid. 2. The system displays an error message: "Invalid email or password." 3. The user is prompted to retry logging in or select the "Forgot Password" option. (Refer Extend Use Case: **Forgot Password**)
Exceptional Flow	-
Post Condition	1. The user is logged into the system and can access their dedicated dashboard.

Table D.2. *Use Case Specification for Forgot Password*

Use Case ID	UC-002
Use Case Name	Forgot Password
Actor	Student, Admin/Event Organiser
Description	This use case allows users to reset their password if they cannot log in due to forgotten credentials.

Pre-Condition(s)	1. The user has a registered account in the system.
Main Flow	1. From the login screen, the user clicks the "Forgot Password" link. 2. The system prompts the user to enter their registered email address. 3. The user enters their email and clicks "Send Reset Link." 4. The system verifies the email in Firebase Auth. [A1] 5. If the email exists, the system sends a password reset link to the user's email. 6. The system displays a confirmation message: "Password reset link sent to your email." 7. The user clicks the link in their email and resets their password.
Alternative Flow	[A1]: Invalid Email Address 1. The system finds that the entered email address is not registered. 2. The system displays an error message: "Email not found. Please try again."
Exceptional Flow	-
Post Condition	The user receives a password reset link and can reset their password.

Table D.3. *Use Case Specification for Sign Up*

Use Case ID	UC-003
Use Case Name	Sign Up
Actor	Student
Description	This use case describes how a student registers for an account in the system by providing personal and university-related details, setting up login credentials, and verifying their email.
Pre-Condition(s)	1. The application is installed and accessible on the user's device.
Main Flow	1. The student navigates to the "Sign Up" screen from the login screen. 2. The system prompts the student to enter: • First name, last name, email address (must be in siswa account format), Password (minimum 6 characters), Faculty (selected from a list), Year of Study (selected from a list). 3. The student fills in all required fields and taps the "Sign Up" button. 4. The system validates all inputs [E2]:

	• Checks for empty fields, validates email format, checks password length, and ensures faculty and year are selected. 5. The system attempts to create a new account [A1]: • If the email is already registered, an error is shown ([E1]). • If account creation is successful, a verification email is sent. 6. The system displays a modal: "Verification Email Sent!". 7. The student checks their email and clicks the verification link. 8. The system marks as verified and completes account creation. 9. The student is redirected to the Sign In screen.
Alternative Flow	**[A1] Validation Error** • The system highlights invalid fields and displays appropriate error messages.
Exceptional Flow	**[E1] Email Already Exists** • The system displays: "Email is already registered." • The student is prompted to sign in or reset their password. **[E2] Validation Error** • The system highlights invalid fields and displays appropriate error messages.
Post Condition	1. A new, verified student account is created in the system. 2. The student can now sign in and access the application.

Table D.4. *Use Case Specification for Register Event*

Use Case ID	UC-004
Use Case Name	Register Event
Actor	Student
Description	This use case allows a student to browse available events, view event details, and register for an event if eligible.
Pre-Condition(s)	1. The student is logged in and navigated to the Event Listing page.
Main Flow	1. The student navigates to the event listing screen. 2. The system displays a list of available events, with options to filter by category and search by keyword. 3. The student selects an event from the listing.

	4. The system displays the event details screen, showing event info, images, location, and registration button. 5. The student clicks the "Register Now" button for the desired event. 6. The system displays a slide-in popup registration form. 7. The system auto-fills user details (full name, email). 8. If the event requires payment proof, the student uploads a payment proof image. 9. Once all required fields are filled and verified, the student clicks the "Submit" button. 10. The system verifies the provided information. [A1] 11. The system shows a registration confirmation popup message. 12. The system redirects the student to the event listing page.
Alternative Flow	**[A1] Payment Proof Not Uploaded** • The student tries to submit registration for an event requiring payment proof without uploading an image. • The system disables the submit button and prompts the student to upload payment proof.
Exceptional Flow	• If the event is full, restricted by faculty/year, or the student is already registered, the registration button is hidden, and a message is shown.
Post Condition	The student is registered for the event, and a confirmation message is displayed.

Table D.5. *Use Case Specification for View Leaderboard*

Use Case ID	UC-005
Use Case Name	View Leaderboard
Actor	Student
Description	This use case allows a student to view the leaderboard for their faculty. The leaderboard displays the top 3 students, the full ranking list, the student's own rank, and provides information about how to earn points and rewards for each ranking tier.
Pre-Condition(s)	1. The student is authenticated and logged into the application.

Main Flow	1. The student taps the "Leaderboard" icon in the bottom tab of the application. 2. The system displays the leaderboard for the student's faculty, showing: • Faculty name, top 3 students on a podium, remaining students in a ranked list, the student's own rank and points, highlighted. 3. The student can scroll through the ranking list to view all ranked students.
Alternative Flow	[A1] Student needs the information guide. 1. The student taps the information icon on the leaderboard screen. 2. The system displays a modal with: • Leaderboard information, rewards for each ranking tier, and explanation of how points are earned
Exceptional Flow	-
Post-Condition	• The leaderboard data is successfully displayed to the student. • If applicable, a modal with additional information is shown.

Table D.6. *Use Case Specification for View Registered Event Details*

Use Case ID	UC-006
Use Case Name	Manage Registered Events
Actor	Student
Description	This use case describes how a student views and manages detailed information about a registered event from the event calendar or agenda.
Pre-Condition(s)	1. The student must have at least one registered event displayed on the calendar list.
Main Flow	1. The student navigates to the registered event calendar page by tapping the “Agenda” icon at the bottom tab of the navigation bar. 2. The student taps on an event from the registered event calendar list. 3. The system navigates the student to the Registered Event Details page. 4. The page displays details such as event details, registration status, attendance, and payment proof (if event required payment). 5. The student reviews the event details.

	6. The student may navigate to the "Quest" tab **(see Extend use case: Complete Quest)**. 7. If eligible (more than 1 hour before event), the student can cancel their registration.
Alternative Flow	-
Exceptional Flow	-
Post-Condition	• The student reviews information about the selected registered event and can manage their registration if eligible.

Table D.7. *Use Case Specification for Complete Quest*

Use Case ID	UC-007
Use Case Name	Complete Quest
Actor	Student
Description	This use case describes how a student completes event-specific quests to earn rewards, including leaderboard points and virtual currency (diamonds). Quests are accessible from the event details page and include attendance-based, early bird attendee, Q&A-based, networking-based, and feedback-driven quests.
Pre-Condition(s)	1. The student must have navigated to the event details page. 2. The event is within 1 hour of its start time and is not cancelled.
Main Flow	1. The student taps the "Quest" tab at the event details page. 2. The system navigates to the event quest page. 3. The system displays a list of quests in card format (title only). [A1] 4. The student taps on a quest card. 5. A slide-in popout modal displays the quest details, including the description, completion criteria, and rewards. [A2] 6. The student completes the quest based on the type: • Attendance-Based Quest (i). The student taps the "Scan Attendance" button. (ii). The system opens the QR scanning page. (iii). The student scans the event QR code. [A3][E1] (iv). The system performs GPS verification.

	(v). The system marks the quest as completed and awards rewards if the student is within a 150m radius of the venue. [A4] (vi). The system navigates back to the quest listing page. • Early Bird Attendee Quest (i). The system awards rewards to the first targeted number of attendees who verify their attendance via QR code scanning and GPS. [A5] • Question and Answer (Q&A) Quest (i). The student answers the displayed question. (ii). The student taps "Submit Answer". (iii). The system verifies the answer. (iv). The system marks the quest as completed and awards rewards if the answer is correct. [A6] • Networking-Based Quest (i). The student taps the "Scan Attendee QR Code" button. (ii). The student scans other attendees' QR codes. (iii). The system tracks progress toward the required number of connections. (iv). The system marks the quest as completed and awards rewards if the required connections are met. [E1][A6] • Feedback Form-Driven Quest (i). The student taps "Fill in Feedback Form". (ii). The system navigates to the feedback form page. (iii). The student completes the feedback form. (iv). The student taps "Submit". (v). The system marks the quest as completed and awards rewards if all fields are completed correctly. [A7]
Alternative Flow	[A1] Quests Not Available: a) The system displays: "Quests will be available 1 hour before the event starts." [A2] Navigating Back:

	a) The student taps "Cancel" on the popout modal to return to the quest listing. [A3] Event QR Code Does Not Match: a) The system displays a scan failed message and returns to the quest listing. [A4] Student Outside 150m GPS Radius: a) The system displays a message and marks the quest as incomplete. [A5] Failed Early Bird Quest: a) The quest is marked as failed and no rewards are given if the student is not among the first N attendees. [A6] Required Connections Not Met: a) The system returns to the quest page and updates quest progress. [A7] Incomplete Feedback Form: a) The system prompts the student to fill all fields before submitting.
Exceptional Flow	**[E1] QR Code Scanning Error:** a) The system displays a scan failed message and prompts the student to try again.
Post-Condition	The student successfully completes one or more event quests and receives rewards (diamonds, points, badge progress, leaderboard updates).

Table D.8. *Use Case Specification for Manage Merchandise Redemption*

Use Case ID	UC-008
Use Case Name	Manage Merchandise Redemption
Actor	Student
Description	This use case enables students to redeem merchandise using their virtual currency (diamonds) earned through event participation. Students can browse, search, and filter merchandise, view details, select items, and complete the redemption process. Redeemed items are tracked for collection.
Pre-Condition(s)	1. The student has a sufficient balance of diamonds. 2. Merchandise inventory is available for redemption.

Main Flow	1. The student views their diamond balance on the home page. 2. The student taps the diamond balance card to navigate to the merchandise listing. 3. The system displays the merchandise list available for redemption. 4. The student can filter merchandise by category (e.g., Clothing, Non-Clothing). [A1] 5. The student can search for merchandise by keyword. 6. The student selects a merchandise item to view its details. 7. The system displays the merchandise details, including images, description, collection place, size chart (if applicable), and publisher. 8. The student selects the desired quantity and, if applicable, size, then confirms the redemption. [E1][E2] 9. The system deducts the required number of diamonds from the student's balance. 10. The system adds the redeemed item to the student's redemption list. 11. The student can view redemption history, including uncollected and collected items, with details such as redemption ID, quantity, size, and collection status.
Alternative Flow	[A1]: Filter Not Applied 1. If the student does not apply any filter, the system displays all available merchandise by default.
Exceptional Flow	**[E1] Insufficient Diamonds**: • If the student does not have enough diamonds, the system displays an error message: "Insufficient Diamonds to Redeem This Item." The student is redirected back to the merchandise details page. **[E2] Size Not Selected (for Clothing)**: • If the student attempts to redeem a clothing item without selecting a size, the redeem button is disabled and redemption is not submitted.
Post-Condition	1. The redemption is recorded and visible in the redemption list. 2. The student's diamond balance is updated. 3. The redeemed merchandise status is marked as "Yet to Collect".

Table D.9. *Use Case Specification for View Badge*

Use Case ID	UC-009
Use Case Name	View Badge
Actor	Student
Description	This use case allows the student to view the details of a specific achievement badge, including its description, unlock status, and progress, from their profile page. The student can return to the profile page after viewing badge details.
Pre-Condition(s)	1. The student must be logged into their account.
Main Flow	1. The student taps the "Profile" icon/tab in the bottom navigation bar. 2. The student clicks on one of the badges displayed on the profile page. 3. The system displays the badge details page, showing: • Badge image (dimmed if locked), badge name, description, progress bar and value, unlock status (Unlocked or locked) 4. The student reviews the badge details.
Alternative Flow	[A1] Navigating Back to Profile Page: 1. The student clicks the close button to close the badge details page. 2. The system returns to the profile page.
Exceptional Flow	-
Post-Condition	1. The student views the badge details.

Table D.10. *Use Case Specification for View Notification*

Use Case ID	UC-010
Use Case Name	View Notification List
Actor	Student
Description	This use case allows a student to view a list of notifications sent to them, including notifications with timestamps and message details.
Pre-Condition(s)	1. Student is logged into the application. 2. Student has access to the Notification List screen.

Main Flow	1. Student navigates to the Notification List screen (e.g., by tapping the notification icon on the home screen). 2. The system fetches all notifications for the logged-in student from the database. 3. The system displays a loading indicator while fetching notifications. 4. The system displays the list of notifications, sorted by newest first. 5. Unread notifications are visually distinct from read notifications. 6. Student can scroll through the list to view all notifications.
Alternative Flow	• If there are no notifications, the system displays a message such as "No notifications yet." • If the student taps the back button, the system navigates back to the previous screen.
Exceptional Flow	-
Post-Condition	1. Student has viewed their notifications.

Table D.11. *Use Case Specification for Manage Networking Connections*

Use Case ID	UC-011
Use Case Name	Manage Networking Connections
Actor	Student
Description	This use case allows a student to browse, search, filter, and paginate their networking connections—established through networking quests. The student can view detailed information for each connection, apply filters based on hobbies and the last attended event, and initiate or continue a conversation from the profile page.
Pre-Condition(s)	1. The student is authenticated and logged into their account. 2. The student has at least one networking connection (from networking quests).
Main Flow	1. The student navigates to the Profile screen. 2. The student scrolls to the “Others” section and taps the “Network” card. 3. The system displays the Network screen, showing a paginated list of networking connections, each with:

	• Name, faculty, year of study, event they met, profile picture, last attended event (with filter capability), unread message indicator (if applicable), and hobbies. 4. The student may use the **search bar** to find connections by name or hobby. 5. The student can filter the list by hobbies and paginate through the list if there are multiple pages. 6. The student taps a connection card to view details and open the Messaging screen. 7. The system opens the Messaging screen for that connection: • If a chat does not exist, it is created. • If a chat exists, previous messages are loaded. 8. The student may click the information guide icon to see network information. 9. The student sends a message. 10. The system sends the message, updates the chat history, and displays the new message in real time.
Alternative Flow	• If the student enters a search query, only matching connections are displayed. • If there are no connections, an empty state UI is shown.
Exceptional Flow	• If the student has no networking connections, the system displays an empty state message.
Post-Condition	The student can view, search, and paginate their networking connections, and send or receive messages with any connection.

Table D.12. *Use Case Specification for Create Event*

Use Case ID	UC-012
Use Case Name	Create Event
Actor	Admin/Event Organiser
Description	This use case describes the process where the admin creates a new event by navigating to the "My Events" tab, entering event details, setting up event quests, and confirming the creation. The system then

	successfully creates the event and redirects the admin to the event listing page.
Pre-Condition(s)	1. The admin is authenticated and logged into the system.
Main Flow	1. The admin navigates to the "My Events" tab in the side navigation bar. 2. The system displays a list of events managed by the admin. 3. The admin clicks the "Add Event" button. 4. The system navigates to the event creation page. 5. The system displays input fields for event details, including: • Event name (required), event description (required), event category (required), start date/time (required), end date/time (required), registration closing date/time (required), location name (required), map pinpoint (required), poster upload (at least 1, max 4 images, required), • Optional restrictions: faculty, year, capacity, payment proof [Capacity value (if enabled, required), year selection (if enabled, at least one year required)] 6. The admin fills in the form and verifies the details. 7. The system validates all required fields ([E1]): • No required field is left blank, start date/time is not in the past, end date/time is after start date/time, registration closing date/time is before event start and at least 1 hour before, at least one image is uploaded, no more than 4,, if capacity restriction is enabled, capacity is a positive number, if year restriction is enabled, at least one year is selected. 8. If validation passes, the admin clicks "Next". 9. The system navigates to the quest creation step. 10. The admin adds and configures quests **(Refer Include Use Case: Create Quest)** 11. The admin verifies quest details. 12. The admin clicks "Create Event" (or "Submit"). 13. The system saves the event, images, and quests to the database. 14. The system displays a success message/snackbar.

	15. The system navigates the admin back to the event listing page. 16. The new event appears in the event list.
Alternative Flow	[A1]: Optional Input Fields 1. The system allows form submission without optional fields (e.g., faculty restriction, payment proof requirement, etc.) being filled.
Exceptional Flow	[E1]: Missing Input for Required Input Fields 1. If the admin tries to proceed with missing required fields, the system highlights errors and disables the "Next" button.
Post-Condition	1. The event is successfully created and stored in the system. 2. The admin is navigated back to the event listing page where the newly created event is displayed.

Table D.13. *Use Case Specification for Create Quest*

Use Case ID	UC-013
Use Case Name	Create Quest
Actor	Admin/Event Organiser
Description	This use case describes how, after completing event details, the admin proceeds to the Quest Creation step where the system automatically includes Attendance and Feedback quests with fixed titles and descriptions but editable rewards. The admin can add one Early Bird quest, one Networking quest, and multiple Q&A quests, each with configurable settings. Eligible quests can be edited or removed before finalizing the event, and all quests are validated before being added.
Pre-Condition(s)	1. The admin has completed the event details form and is on the Quest Creation page.
Main Flow	1. The system displays the Quest Creation page with Attendance and Feedback quests already listed, showing default titles, descriptions, and rewards (editable). 2. The admin clicks "Add New Quest". [E2] 3. The system opens a stepper dialog for quest creation. 4. The admin selects a quest type from Early Bird Attendance Quest (only one allowed), Networking Quest (only one allowed), or

	Question & Answer (Q&A) Quest (multiple allowed). The system disables selection of Early Bird or Networking if one already exists. 5. The system prompts for quest-specific fields: • Early Bird: Max early bird attendees (required, >0) • Networking: Number of required connections (required, >0) • Q&A: Question and correct answer (both required) 6. The admin enters the required details. 7. The admin enters points and diamonds rewards (both required, >0). 8. The system validates all required fields for the selected quest type. If any required field is missing or invalid (e.g., zero/negative rewards), an error is shown, and the quest cannot be saved. 9. If validation passes, the admin saves the quest. The system adds the quest to the quest list and displays it as a card. [E1] 10. The admin can edit or remove Early Bird, Networking, or Q&A quests before event submission. Attendance and Feedback quests cannot be edited or removed. [A1] 11. The admin reviews all quests in the list. Upon event submission, all quests are saved and associated with the event.
Alternative Flow	• [A1] The admin clicks the edit or delete icon on a quest card (except Attendance/Feedback). The system allows editing or confirms deletion, updating the quest list accordingly.
Exceptional Flow	• [E1] If the admin tries to save a quest with missing or invalid fields, the system displays a validation error and prevents saving. • [E2] If the admin tries to add a second Early Bird or Networking quest, the system prevents it and shows an error.
Post-Condition	The event quests are successfully created and stored in the system.

Table D.14. *Use Case Specification for Manage Existing Event*

Use Case ID	UC-014
Use Case Name	Manage Existing Event
Actor	Admin/Event Organiser

Description	This use case describes how the admin manages existing events through the "Events" tab, with the ability to view and edit event details, manage quests, participants, attendance, and feedback, and control the event status (start, end, cancel).
Pre-Condition(s)	1. The admin has created events to manage.
Main Flow	1. The admin clicks the “Events" tab in the side navigation bar. 2. The system displays a list of events managed by the admin, each card showing event poster, title, start/end times, and status. 3. The admin selects an event by clicking on its respective card. 4. The system navigates to the event management page with tabs for Details, Quest, Participant, Attendance, and Feedback. 5. The admin views the event details in the "Details" tab. 6. The admin edits event details (name, description, category, dates, location, restrictions, images) as needed. 7. The admin uploads, replaces, or deletes event images (up to 4, cannot delete the only image). 8. The admin toggles and edits restrictions (year, faculty, capacity, payment proof) and fills in required fields if enabled. 9. The admin clicks the "Save Changes" button. 10. The system validates the input. If valid, changes are saved and a success snackbar is shown; if invalid, error messages are displayed, and changes are not saved. 11. The admin can change event status by clicking "Start Event", "End Event", or "Cancel Event" at the top of the page. The system updates the event status and UI accordingly. 12. The admin can manage participants in the "Participants" tab (Refer Extend Use Case: **Manage Participants**). 13. The admin can manage attendance in the "Attendance" tab (Refer Extend Use Case: **Manage Attendance**). 14. The admin can manage event quests in the “Quest” tab (Refer Extend Use Case: **Manage Quest**). 15. The admin can manage feedback in the "Feedback" tab (Refer Extend Use Case: **Manage Feedback**).

Alternative Flow	**[A1] No Events Managed by Admin** • The admin clicks the “My Events” tab. The system displays a message indicating no events are managed by the admin. **[A2] Input Fields Verification Failed** • The system displays error messages for invalid or missing inputs. The admin corrects the errors and submits again.
Exceptional Flow	• If the admin attempts to upload more than 4 images, the system shows "Maximum 4 images allowed". • If the admin tries to delete the only image, the delete icon is hidden. • If the admin tries to save with invalid data (e.g., blank required fields), error messages are shown, and changes are not saved.
Post-Condition	1. The selected event’s details are successfully updated and saved. 2. The admin can access and manage the event’s associated features (participants, attendance, quests, feedback) through their tabs.

Table D.15. *Use Case Specification for Manage Participant*

Use Case ID	UC-015
Use Case Name	Manage Participant
Actor	Admin/Event Organiser
Description	This use case describes how the admin manages the list of event participants through the "Participant" tab, with the ability to view all participants, verify registrations (if payment proof is required), view participant details and payment proof, and export verified participant data to Excel.
Pre-Condition(s)	1. The event has at least one registered participant.
Main Flow	1. The admin navigates to the "Participant" tab for a selected event. 2. The system displays a list of all registered participants, separated into "Verified" and "Unverified" tabs if payment proof is required. 3. The admin views participant details in the list, including name, email, faculty, year, and profile picture. 4. If payment proof is required, the admin can click "View Proof" to see the participant’s payment proof image in a modal.

	5. In the "Unverified" tab, the admin can verify a participant's registration by clicking "Verify" (after viewing payment proof if needed). The participant moves to the "Verified" tab and a success snackbar is shown. 6. In the "Verified" tab, the admin can export the list of verified participants to Excel by clicking the "Export" button. The system downloads an Excel file with participant data.
Alternative Flow	• If payment proof is not required, all participants are shown in a single list without verification actions. • If there are no participants, the system displays an appropriate empty state message.
Exceptional Flow	• If there is an error fetching participant data or updating verification status, an error snackbar is shown. • If a participant's payment proof is missing, the modal displays "No image available".
Post-Condition	1. The admin can view, verify, and export participant data as needed. 2. Participant verification status is updated in real time and reflected in the UI

Table D.16. *Use Case Specification for Manage Attendance*

Use Case ID	UC-016
Use Case Name	Manage Attendance
Actor	Admin/Event Organiser
Description	This use case enables the admin to manage event attendance. The admin can view an overview of attendance data, display a QR code for participants to scan and mark attendance, review a detailed list of attendees, and manually mark attendance for participants facing technical difficulties.
Pre-Condition(s)	1. The admin must be logged into the system. 2. At least one event must be created and ongoing. 3. Participants must be registered for the event.
Main Flow	1. The admin navigates to the "Attendance" tab for a selected event.

	2. The system displays two tabs: "Attendees" and "Absentees". 3. The admin selects the "Attendees" tab to view all attendees with details (name, email, faculty, year, profile picture, attendance method, and reason if manual). 4. The admin selects the "Absentees" tab to view all absentees with details. 5. In the "Absentees" tab, the admin can click "Manual Attendance" for a student, enter a reason in the dialog, and confirm. The student moves to the "Attendees" tab, the method is set to "Manual", the reason is saved, and the attendance quest is marked as completed for the student. 6. If the admin attempts to confirm manual attendance with an empty reason, an error message "This field cannot be empty!" is shown and attendance is not updated. 7. In the "Attendees" tab, for students with manual attendance, the method shows "Manual", and a tooltip displays the reason. 8. The admin can export the list of attendees to Excel by clicking the "Export" button in the "Attendees" tab. The system downloads an Excel file with attendee data.
Alternative Flow	• If there are no attendees or absentees, the system displays an appropriate empty state message. • If there are no attendees, the export button is hidden.
Exceptional Flow	• If the admin tries to mark manual attendance without a reason, the dialog shows a validation error and does not proceed.
Post-Condition	1. The admin can view, filter, manually verify, and export attendance records as needed. 2. Attendance status and reasons are updated in real time and reflected in the UI.

Table D.17. *Use Case Specification for Manage Quest*

Use Case ID	UC-017
Use Case Name	Manage Quest
Actor	Admin/Event Organiser

Description	The admin manages event quests by overseeing summaries, creating new quests, viewing and modifying quest details, and monitoring participant progress. This involves navigating to the event quest manager, accessing and editing quest details, and manually setting quest completion for participants facing issues.
Pre-Condition(s)	1. The admin is authenticated and logged into the system.
Main Flow	1. The admin navigates to the "Quest" tab for a selected event. 2. The system displays an overview of all quests, showing quest type, name, and the number of participants who have completed each quest. 3. The admin clicks "Add Quest" to open the quest addition form. 4. The admin selects a quest type from the dropdown: o Early Bird Attendance Quest (only if not already present) o Networking Quest (only if not already present) o Question & Answer (Q&A) Quest (multiple allowed) The system disables selection of Early Bird or Networking if one already exists. 5. The admin fills in the required fields for the selected quest type and submits the form. 6. The system validates the input. If valid, the quest is added and appears in the quest list; if invalid, validation errors are shown, and the quest is not added. 7. The admin can click on a Q&A, Early Bird, or Networking quest to view and edit its details, then save changes. Attendance and Feedback quests are view-only and cannot be edited. 8. The admin can delete a Q&A, Early Bird, or Networking quest. Attendance and Feedback quests cannot be deleted. 9. The admin can view participant progress for each quest, including "Completed" and "In Progress" tabs, showing participant details, completion status, and rewards claimed.
Alternative Flow	• If the admin attempts to add a duplicate Early Bird or Networking quest, those quest types are not selectable in the dropdown.

	• If there are no quests (other than Attendance and Feedback), the system displays an appropriate empty state message.
Exceptional Flow	-
Post-Condition	The system reflects updated quest details, including additions, modifications, or participant progress updates.

Table D.18. *Use Case Specification for Manage Feedback*

Use Case ID	UC-018
Use Case Name	Manage Feedback
Actor	Admin/Event Organiser
Description	This use case describes how the admin manages feedback submitted by event participants through the "Feedback" tab, with the ability to view all feedback entries, including participant details, Likert scale ratings, and written comments.
Pre-Condition(s)	1. The admin is authenticated and has access to the event management page. 2. The event has at least one feedback submission.
Main Flow	1. The admin navigates to the "Feedback" tab for a selected event. 2. The system displays an overview with the total number of participants who submitted feedback. 3. The admin views the feedback list, which shows each participant's name, profile picture, Likert scale ratings (event satisfaction, gamification satisfaction), and written comments (overall improvement). 4. The admin can view the "Feedback Rating Scale" section, which explains the Likert scale values and their colour codes. 5. If feedback exists, the admin can click the "Export" button to download all feedback data as an Excel file, including participant names, ratings, and comments.
Alternative Flow	• If there is no feedback, the system displays an appropriate empty state message.

	• If there is no feedback, the export button is hidden.
Exceptional Flow	• If there is an error fetching or exporting feedback data, an error snackbar is shown.
Post-Condition	1. The admin can view, filter, and export all feedback for the event. 2. Feedback data is available for further analysis or reporting.

Table D.19. *Use Case Specification for Create Merchandise*

Use Case ID	UC-019
Use Case Name	Create Merchandise
Actor	Admin/Event Organiser
Description	This use case describes how the admin creates a new merchandise item using the "Create Merchandise" page by entering required details, uploading images, and handling category-specific logic.
Pre-Condition(s)	1. The admin is logged into the system.
Main Flow	1. The admin navigates to the Merchandise Listing page and clicks the "Create Merch" button. 2. The system displays the "Create Merchandise" form with fields for name, description, diamonds required, pickup location, category, sizes (if Clothing), and image upload. 3. The admin uploads 1–4 valid images. The system compresses and previews the images. If more than 4 images are selected, an error message "Maximum 4 images allowed" is shown and only 4 images are accepted. 4. The admin fills in all required fields: name, description, diamonds needed, pickup location, and category. 5. If the category is "Clothing", the admin adds at least one size. 6. The admin can remove or replace any uploaded image before submission. 7. The "Submit" button is enabled only when all required fields are filled and validation passes. 8. The admin clicks "Submit". The system validates the form, processes the images, and submits the data to the backend.

	9. On successful creation, a success snackbar appears and the admin is redirected to the merchandise listing page, where the new item is visible.
Alternative Flow	• If required fields are missing or invalid, the form displays validation errors and the "Submit" button is disabled. • If the admin tries to upload more than 4 images, an error message is shown and only 4 images are accepted. • If the admin replaces an image, the new image is previewed and replaces the old one without errors. • If the admin switches from "Clothing" to "Non-Clothing", the sizes array is cleared automatically.
Exceptional Flow	• If image compression or upload fails, an error message "Failed to process image(s)" is shown. • If there is a backend error during submission, an error is logged, and the admin is not redirected.
Post-Condition	1. The new merchandise item is created and stored in the backend. 2. The admin is notified of success and can see the new item in the merchandise listing.

Table D.20. *Use Case Specification for Manage Existing Merchandise*

Use Case ID	UC-020
Use Case Name	Manage Existing Merchandise
Actor	Admin/Event Organiser
Description	This use case describes how the admin manages merchandise items through the merchandise management page, with the ability to view and edit item details, manage images, change categories (with size logic), and update student redemption statuses.
Pre-Condition(s)	1. The admin is authenticated and has access to the merchandise management page. 2. At least one merchandise item exists.
Main Flow	1. The admin navigates to the Merchandise Listing page and selects a merchandise item.

	2. The system displays all details for the selected merchandise (name, description, images, diamonds, category, sizes if Clothing, pickup location). 3. The admin edits any field (name, description, diamonds, category, sizes, location) as needed. 4. The admin uploads up to 4 images, replaces, or deletes images. The system prevents deleting the only image and shows errors if limits are violated. 5. If the admin changes the category from "Clothing" to "Non-Clothing", the sizes field is removed. 6. The "Save Changes" button is enabled when there are unsaved changes. The admin clicks "Save Changes" to update the merchandise. The system validates all fields, saves changes, and shows a success snackbar. 7. The admin navigates to the "Redemptions" tab to manage student redemptions. 8. In the "Uncollect" tab, the admin clicks "Mark as Collected" for a student. The student moves to the "Collected" tab and the database is updated. 9. In the "Collected" tab, the admin clicks "Mark as Uncollected" for a student. The student moves back to the "Uncollect" tab, and the database is updated.
Alternative Flow	• If required fields are missing or invalid, error messages are shown, and changes are not saved. • If the admin tries to upload more than 4 images, an error message is shown and only 4 images are accepted. • If the admin tries to delete the only image, the delete action is prevented and an error is shown.
Exceptional Flow	-
Post-Conditions	1. Merchandise details and redemption statuses are updated and reflected in the UI and database. 2. All changes are validated, and errors are shown for invalid actions.

Table D.21. *Use Case Specification for View Statistics*

Use Case ID	UC-021
Use Case Name	View Statistics
Actor	Admin/Event Organiser
Description	This use case describes how the admin views comprehensive statistics through the "Statistics" page, which includes two tabs: "Event Stats" and "Quest Stats". Each tab displays relevant metrics, averages, counts, and trends based on up-to-date data from events managed by the admin.
Pre-Condition(s)	1. The admin is authenticated and has access to the Statistics page. 2. There are events and quests managed by the admin, with feedback and participation data.
Main Flow	1. The admin navigates to the Statistics page. 2. The system displays two tabs: "Event Stats" and "Quest Stats". 3. In the "Event Stats" tab: • The system displays the average event satisfaction rating (out of 5), calculated from participant feedback. • The system shows an attendance summary, including total attendees and absentees across all events. • The system presents yearly statistics, showing the number of events by type, with monthly breakdowns and trends. 4. The admin switches to the "Quest Stats" tab: • The system displays the average quest satisfaction rating (out of 5), calculated from participant feedback. • The system shows the total number of quests by type (e.g., Early Bird, Networking, Q&A). • The system presents quest completion status by type, showing overall completion percentages for each quest type. 5. All statistics are updated in real time as new data is received.
Alternative Flow	• If there is no data for a particular metric (e.g., no feedback submitted), the system displays "No data available". • If the admin manages no events or quests, the statistics sections are empty or show an appropriate message.

Exceptional Flow	-
Post-Condition	1. The admin is able to see event or quest related statistics.

Table D.22. *Use Case Specification for Monitor Monthly Leaderboard*

Use Case ID	UC-022
Use Case Name	Monitor Monthly Leaderboard
Actor	Admin/Event Organiser
Description	This use case describes how the admin views the current month's leaderboard for their faculty, which shows a real-time ranked list of students with profile pictures, names, year of study, points, and last updated time.
Pre-Condition(s)	1. The system has student leaderboard data for the current month and faculty (or not, for empty state).
Main Flow	1. The admin navigates to the "Feedback" tab for a selected event. 2. The system displays an overview with the total number of participants who submitted feedback. 3. The admin views the feedback list, which shows each participant's name, profile picture, Likert scale ratings (event satisfaction, gamification satisfaction), and written comments (overall improvement). 4. The admin can view the "Feedback Rating Scale" section, which explains the Likert scale values and their colour codes. 5. If feedback exists, the admin can click the "Export" button to download all feedback data as an Excel file, including participant names, ratings, and comments.
Alternative Flow	• If there is no data for the current month and faculty, the system displays a message such as "No Leaderboard Data Yet." • If there are no students with points, the leaderboard table is empty, and the empty state message is shown.
Exceptional Flow	-
Post-Condition	1. The admin can view the current month's leaderboard for their faculty, including all relevant student details and refresh information.

Table D.23. *Use Case Specification for View Student Participation Leaderboard*

Use Case ID	UC-023
Use Case Name	View Student Participation Leaderboard
Actor	Admin/Event Organiser
Description	This use case describes how the admin views a leaderboard of students in their faculty, ranked by events attended and then by total points.
Pre-Condition(s)	1. The admin has access to the Student Participation Overview page. 2. There are students registered in the faculty.
Main Flow	1. The admin navigates to the Student Participation Overview page. 2. The system fetches all students in the admin's faculty and their participation data. 3. The system displays a leaderboard table with the following columns for each student: • Rank (bil), profile picture, full name, year of study, number of events attended (primary sort, descending), and total points gained (secondary sort, descending) 4. The system sorts students first by events attended (descending), then by total points (descending) for ties. 5. The admin observes the leaderboard, which updates in real time as participation data changes.
Alternative Flow	• If two or more students have the same number of events attended, the student with higher total points is ranked higher. • If there are no students with event participation, the table is empty or displays a message such as "No participation data available."
Exceptional Flow	-
Post-Condition	1. The admin can view an up-to-date, correctly ranked leaderboard of student participation for their faculty.

Appendix E: Sequence Diagram

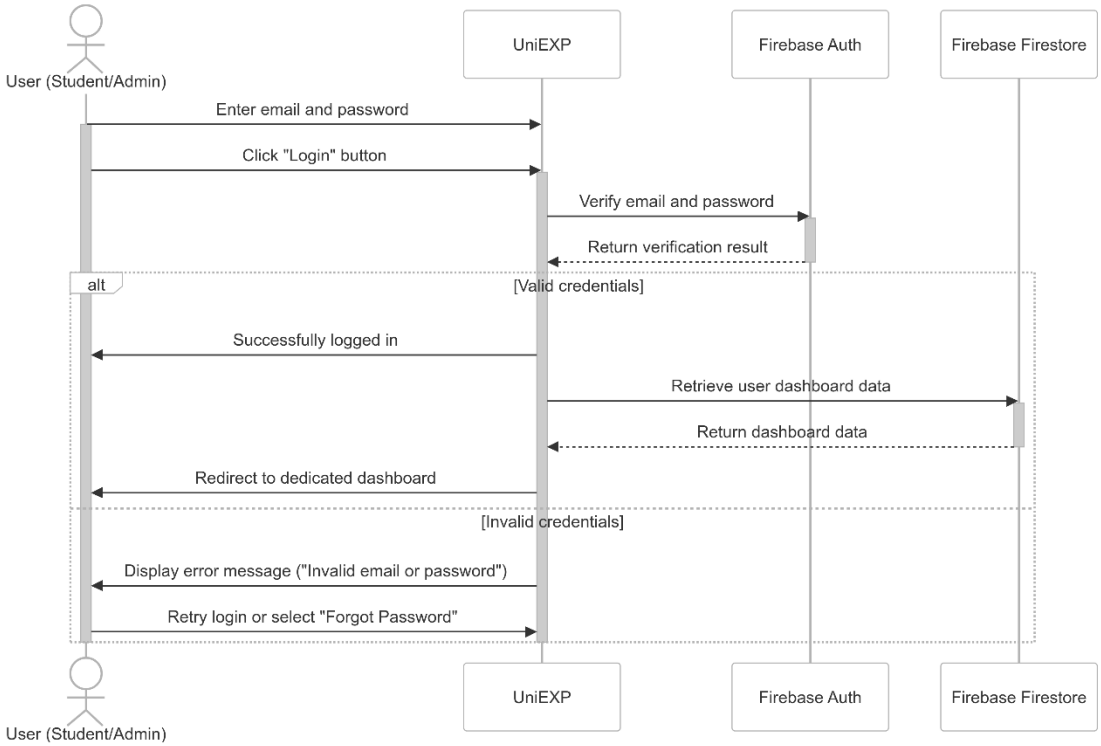


Figure E.1. *Sequence Diagram for Login*

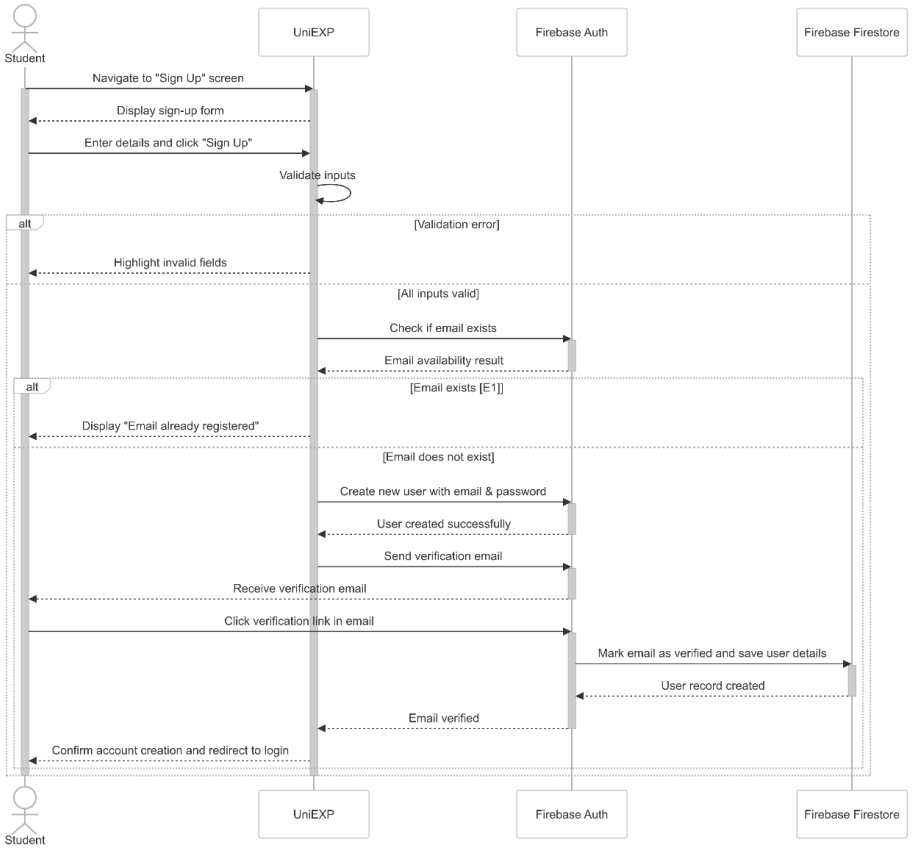


Figure E.2. *Sequence Diagram for Sign Up*

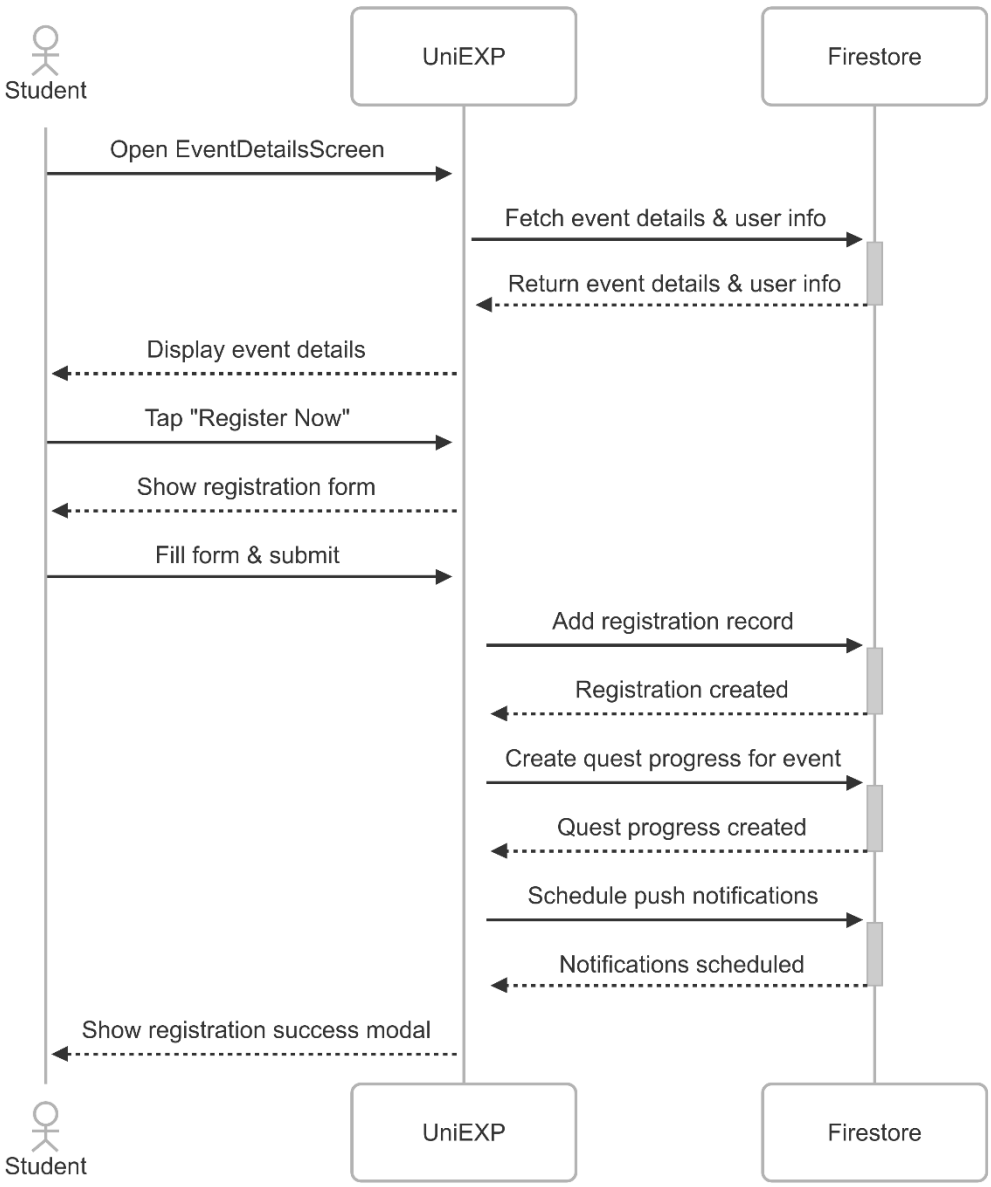


Figure E.3. *Sequence Diagram for Register Event*

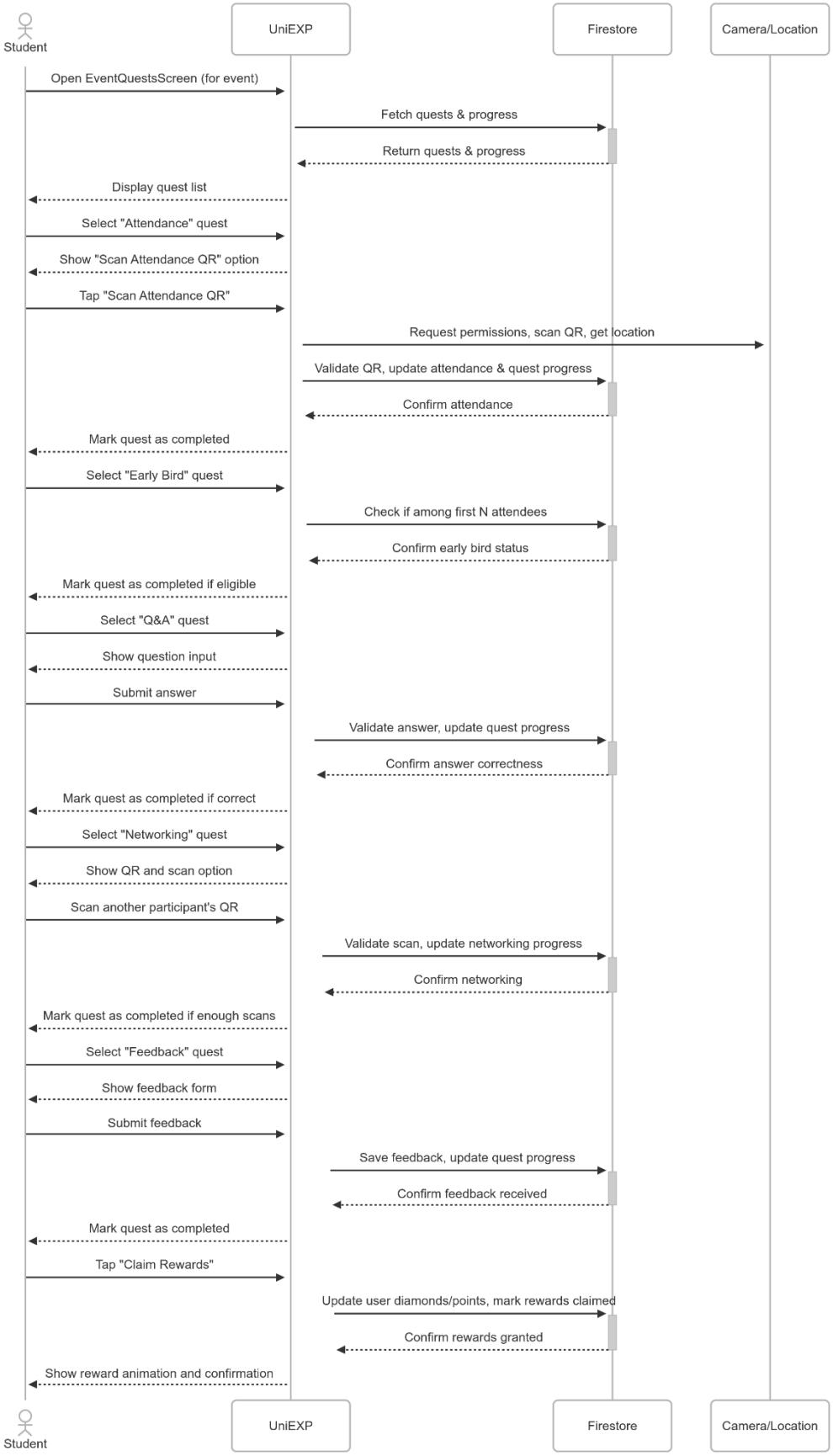


Figure E.4. *Sequence Diagram for Complete Quest*

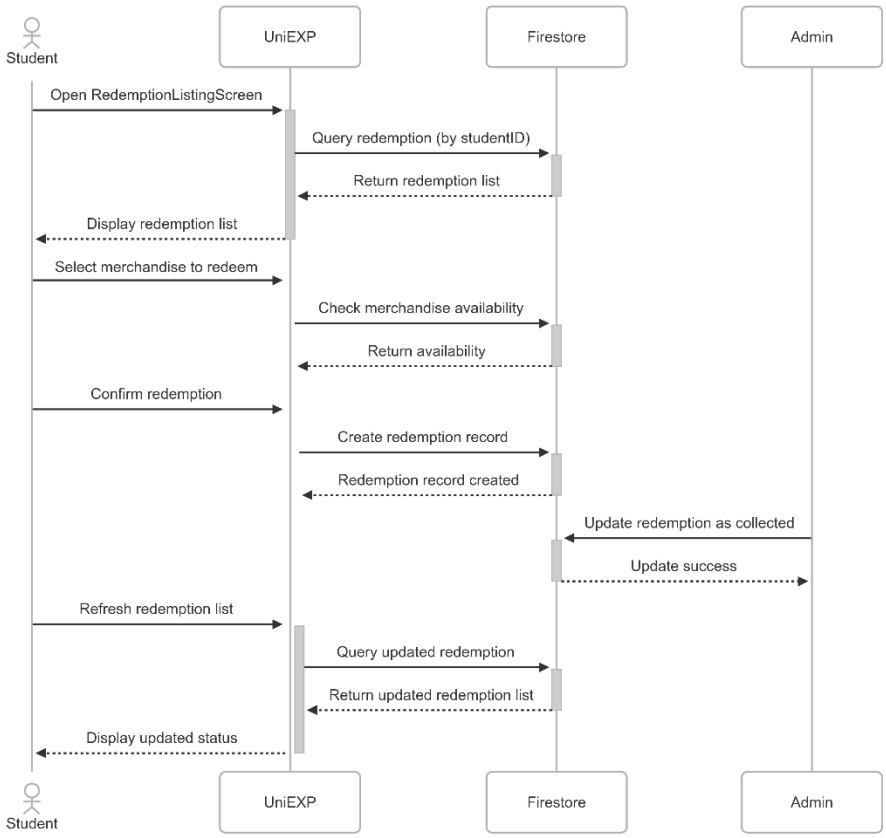


Figure E.5. *Sequence Diagram for Manage Merchandise Redemption*

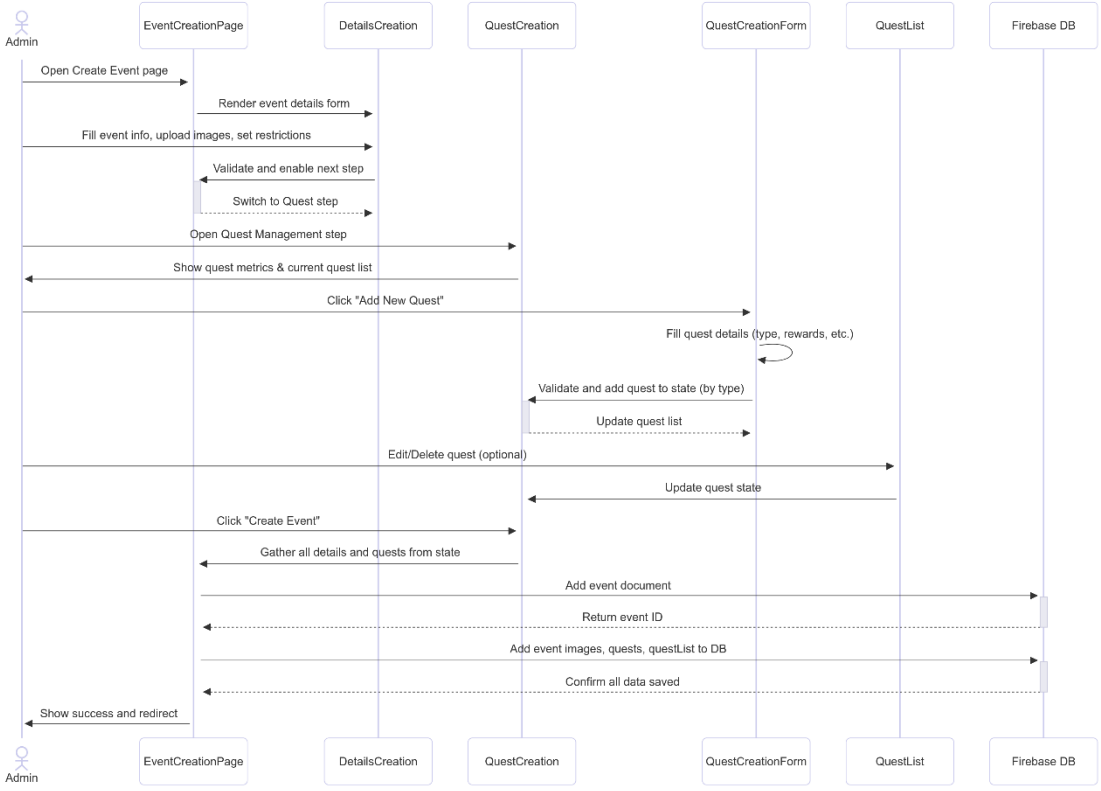


Figure E.6. *Sequence Diagram for Create Event*

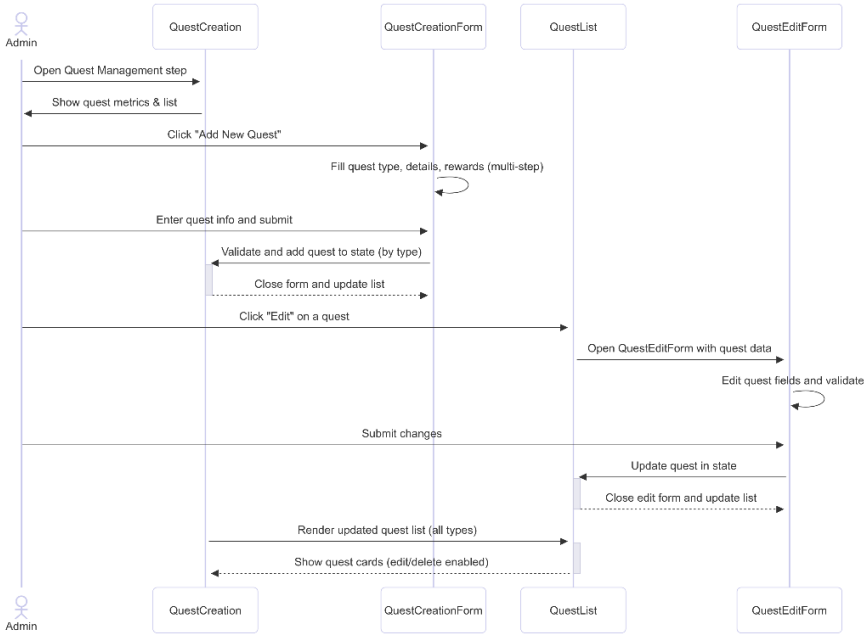


Figure E.7. *Sequence Diagram for Create Quest*

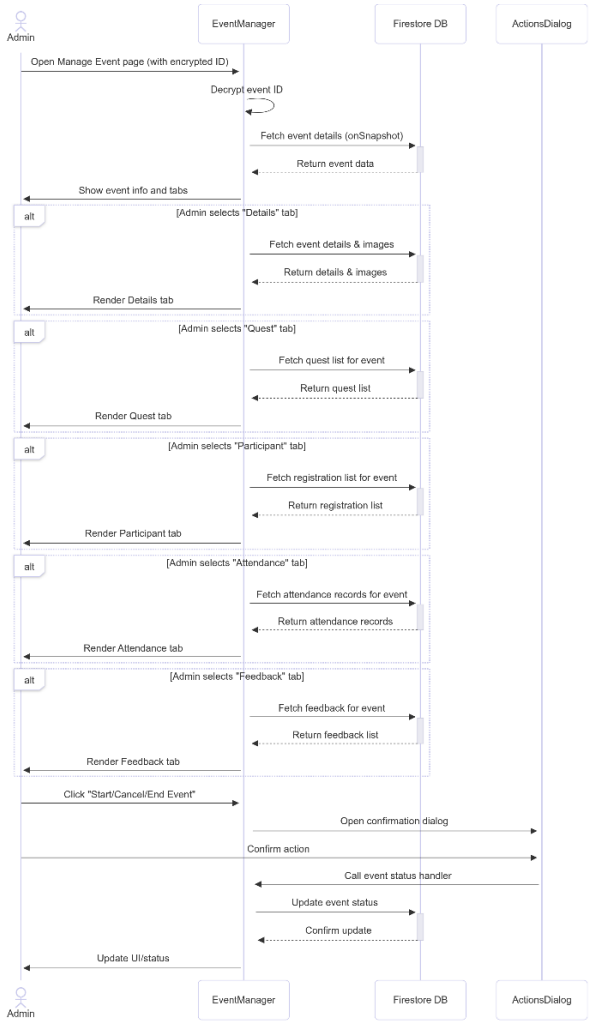


Figure E.8. *Sequence Diagram for Manage Event*

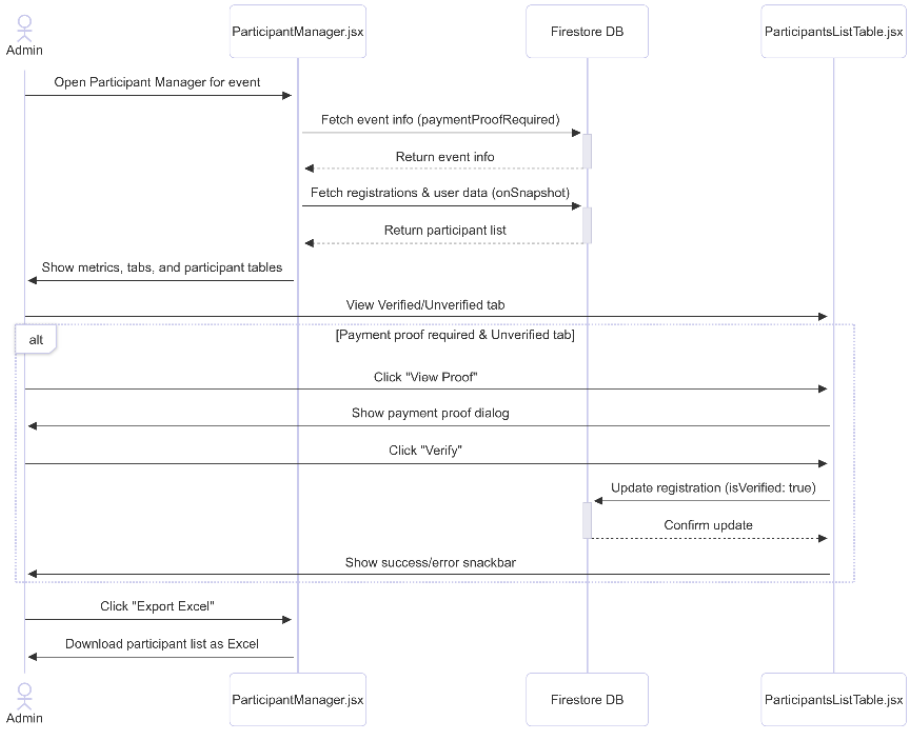


Figure E.9. *Sequence Diagram for Manage Participant*

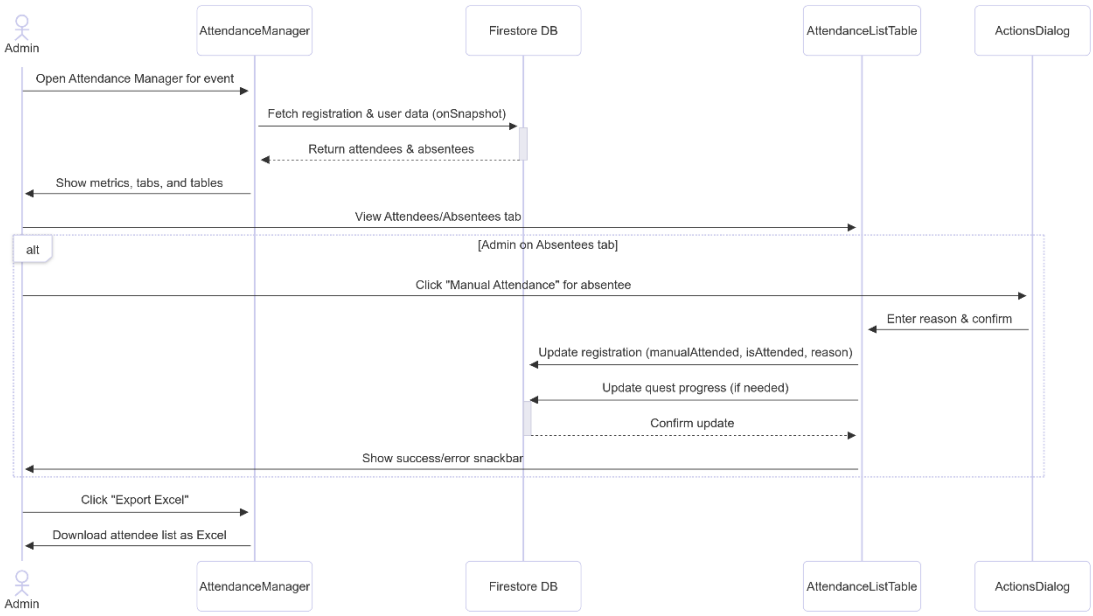


Figure E.10. *Sequence Diagram for Manage Attendance*

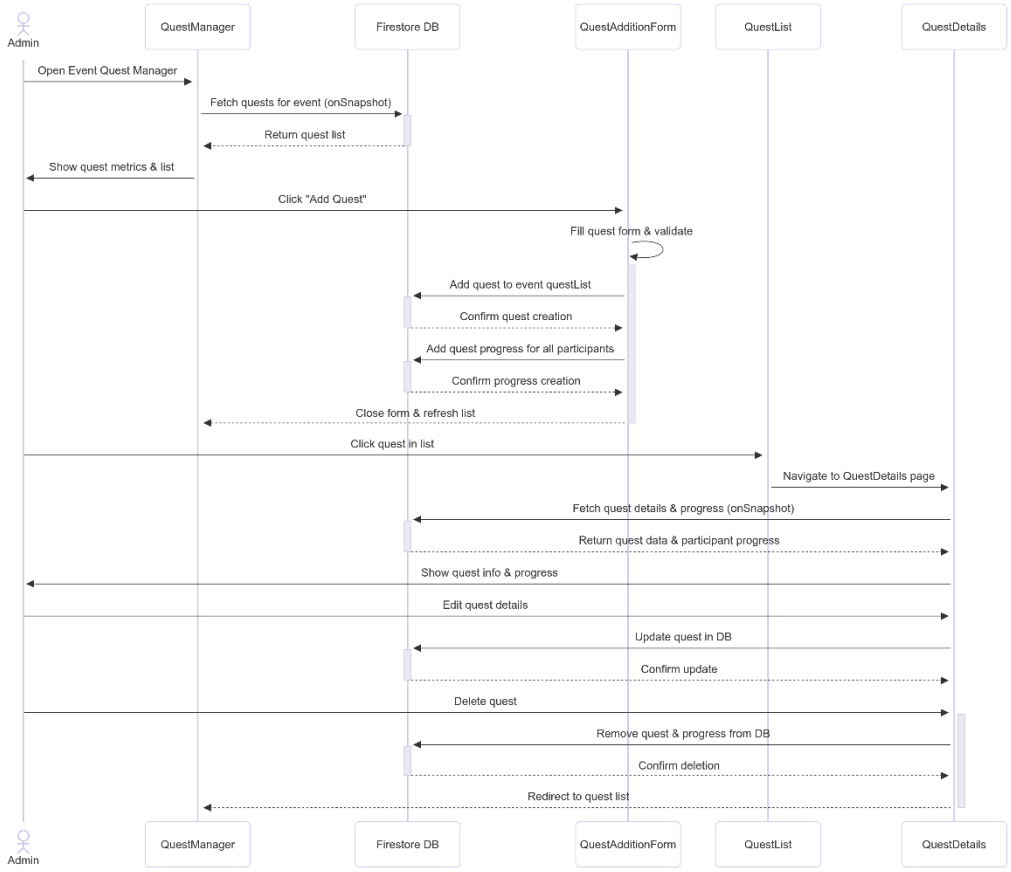


Figure E.11. *Sequence Diagram for Manage Quest*

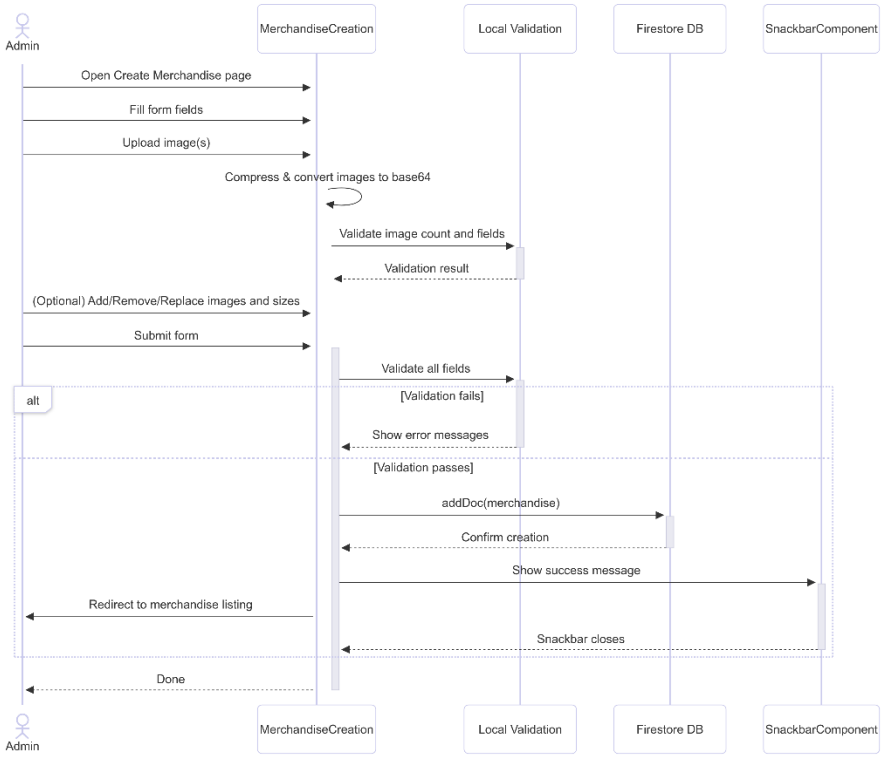


Figure E.12. *Sequence Diagram for Create Merchandise*

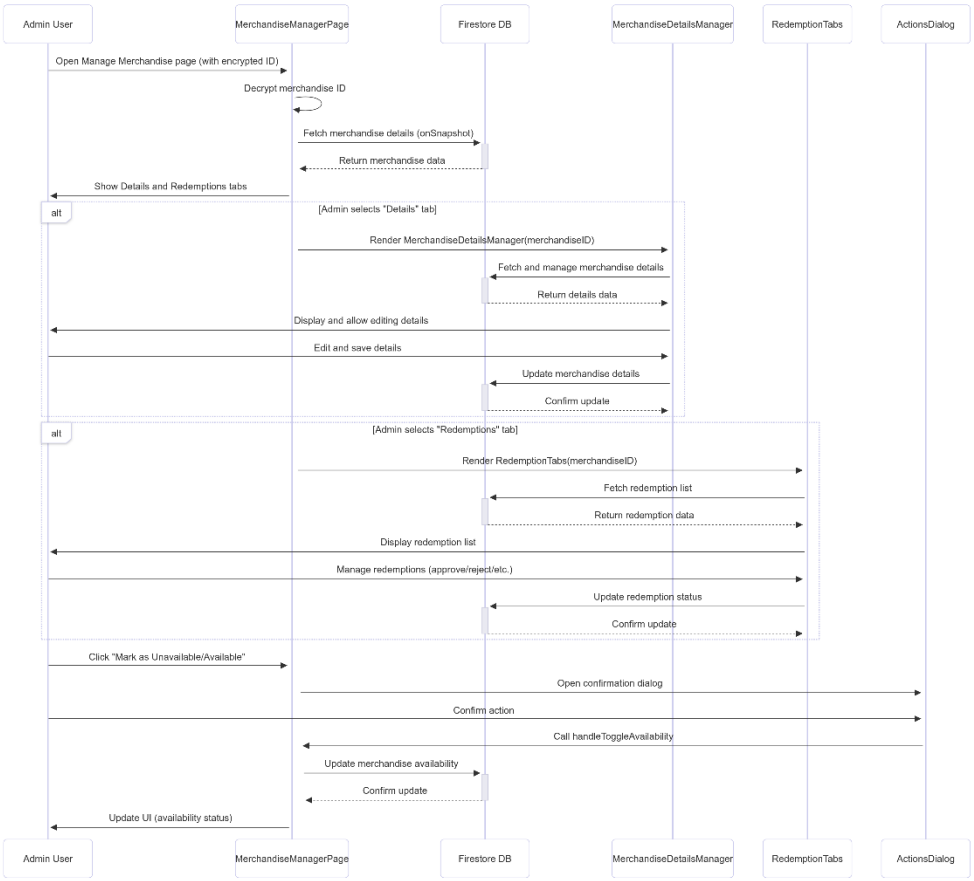


Figure E.13. *Sequence Diagram for Manage Merchandise*

Appendix F: Code Snippet and System Interfaces

```
const _onLoginPressed = async () => {
  // Reset errors
  const emailError = email.value ? '' : 'Email cannot be empty';
  const passwordError = password.value ? '' : 'Password cannot be empty';

  if (emailError || passwordError) {
    setEmail({ ...email, error: emailError });
    setPassword({ ...password, error: passwordError });
    return;
  }

  setLoading(true);

  try {
    // Attempt to sign in with Firebase
    const userCredential = await signInWithEmailAndPassword(auth, email.value, password.value);
    const user = userCredential.user;

    if (!user.uid) {
      setLoading(false);
      await auth.signOut();
      setPassword({ ...password, error: 'User not found. Please sign up first!' });
      return;
    }

    const isAdminRef = query(collection(db, "admin"), where("adminID", "==", user.uid));
    const isAdminSnapshot = await getDocs(isAdminRef);

    if (!isAdminSnapshot.empty) {
      setIsAdminModalVisible(true);
      await auth.signOut();
      setLoading(false);
      return;
    }

    if (!user.emailVerified) {
      setLoading(false);
      setIsVerificationModalVisible(true);
      await auth.signOut();
      return; // Exit the function here
    }
```

Figure F.1. *Code Snippet for Login Validation - Exception*

```
    const userDocRef = doc(db, "user", user.uid);
    const userDocSnap = await getDoc(userDocRef);

    await setItem('studentID', user.uid);
    await setItem('facultyID', userDocSnap.data().facultyID);
    await auth.signOut();
    await signInWithEmailAndPassword(auth, email.value, password.value);
  } catch (error) {
    setLoading(false);

    // Handle different Firebase auth errors
    switch (error.code) {
      case 'auth/user-not-found':
        setEmail({ ...email, error: 'No account found with this email. Please sign up first!' });
        break;
      case 'auth/wrong-password':
        setPassword({ ...password, error: 'Incorrect password. Please try again.' });
        break;
      case 'auth/invalid-email':
        setEmail({ ...email, error: 'Invalid email format.' });
        break;
      case 'auth/too-many-requests':
        setPassword({ ...password, error: 'Too many failed attempts. Please try again later.' });
        break;
      default:
        setPassword({ ...password, error: `Sign in failed: ${error.message}` });
    }
  }
};
```

Figure F.2. *Code Snippet for Login Validation – Authentication Setup*

```
const _onSignUpPressed = async () => {
  setLoading(true);

  // Clear all previous errors
  setFirstName({ ...firstName, error: '' });
  setLastName({ ...lastName, error: '' });
  setEmail({ ...email, error: '' });
  setPassword({ ...password, error: '' });
  setFacultyError('');
  setYearError('');

  if (!validateForm()) {
    setLoading(false);
    return;
  }

  try {
    const userCredential = await createUserWithEmailAndPassword(auth, email.value, password.value);
    const user = userCredential.user;

    await updateProfile(user, {
      displayName: `${firstName.value} ${lastName.value}`
    });

    let expoPushToken = null;
    if (Device.isDevice) {
      const { status: existingStatus } = await Notifications.getPermissionsAsync();
      const finalStatus = existingStatus === 'granted'
        ? existingStatus
        : (await Notifications.requestPermissionsAsync()).status;

      if (finalStatus === 'granted') {
        expoPushToken = (await getExpoPushTokenAsync({
          projectId: "ddfc0801-bc5c-4c5a-81ae-29f43c63ed75",
        })).data;
      } else {
        console.warn('Push notification permission not granted');
        setLoading(false);
        return;
      }
    }

    const userDocRef = doc(db, "user", user.uid);

    const profilePicQuery = query(collection(db, "config"), where("name", "==", "defaultProfilePic"));
    const profilePicSnapshot = await getDocs(profilePicQuery);

    const ProfilePicBase64 = profilePicSnapshot.docs[0].data().base64;

    await setDoc(userDocRef, {
      firstName: firstName.value,
      lastName: lastName.value,
      email: email.value,
      yearOfStudy: year,
      facultyID: String(faculty),
      diamonds: 0,
      totalPointsGained: 0,
      profilePicture: ProfilePicBase64,
      expoPushToken: expoPushToken || null,
    })

    const badgeProgressDocRef = await addDoc(collection(db, "badgeProgress"), { studentID: user.uid });

    const badgeSnapshot = await getDocs(collection(db, "badge"));

    const batchPromises = badgeSnapshot.docs.map(async (badgeDoc) => {
      const badgeID = badgeDoc.id;
      const badgeProgressSubDocRef = doc(db, "badgeProgress", badgeProgressDocRef.id, "userBadgeProgress", badgeID);

      await setDoc(badgeProgressSubDocRef, {
        dateUpdated: serverTimestamp(),
        isUnlocked: false,
        progress: 0,
      });
    });

    await Promise.all(batchPromises);

    await sendEmailVerification(user);
    setIsEmailSentModalVisible(true);
```

Figure F.3. *Code Snippet for Sign Up Validation*

```
const _onSendPressed = async () => {
  setIsLoading(true);
  setEmail({ ...email, error: '' });

  if (!email.value) {
    setEmail({ ...email, error: 'Email cannot be empty' });
    setIsLoading(false);
    return;
  }

  try {
    await sendPasswordResetEmail(auth, email.value);
    setIsPasswordResetModalVisible(true);
  } catch (error) {
    let errorMessage = 'Something went wrong. Please try again.';

    if (error.code === 'auth/invalid-email') {
      errorMessage = 'Invalid email address.';
    } else if (error.code === 'auth/user-not-found') {
      errorMessage = 'No account found with this email.';
    }

    setEmail({ ...email, error: errorMessage });
    setIsPasswordResetModalVisible(false);
  }
}
```

Figure F.4. *Password Recovery Function*

```
const handleCancelRegistration = useCallback(async () => {
    try {
        setIsDeleting(true);
        const registrationRef = doc(db, "registration", registrationID);

        const deleteRegistration = deleteDoc(registrationRef);
        const deleteProgress = deleteQuestProgress();
        const cancelNotifs = cancelNotifications();

        await Promise.all([
            deleteRegistration,
            deleteProgress,
            cancelNotifs
        ]);

        setCancelModalVisible(false);
        navigation.goBack();
    } catch (error) {
        setError('Failed to delete. Please try again.');
        console.error('Delete error:', error);
    } finally {
        setIsDeleting(false);
    }
}, [registrationID, navigation, deleteQuestProgress, cancelNotifications]);
```

Figure F.5. *handleCancelRegistration Function*

```
const cancelNotifications = useCallback(async () => {
    const studentID = await getItem("studentID");
    if (!studentID || !eventID) return;

    const q = query(
        collection(db, 'scheduled_notifications'),
        where('studentID', '==', studentID),
        where('eventID', '==', eventID)
    );

    const snap = await getDocs(q);

    const deletePromises = snap.docs.map(doc =>
        deleteDoc(doc.ref)
    );

    await Promise.all(deletePromises);
    console.log('Notifications have been deleted.');
}, [eventID]);
```

Figure F.6. *cancelNotifications Function*

```
const deleteQuestProgress = useCallback(async () => {
    try {
        const studentID = await getItem("studentID");
        if (!studentID || !eventID) return;

        const questProgressRef = collection(db, "questProgress");
        const questProgressQuery = query(
            questProgressRef,
            where("studentID", "==", studentID),
            where("eventID", "==", eventID)
        );

        const questProgressSnap = await getDocs(questProgressQuery);
        if (questProgressSnap.empty) return;

        const questProgressDoc = questProgressSnap.docs[0];
        const progressListRef = collection(db, "questProgress", questProgressDoc.id, "questProgressList");

        // Fetch subcollection and delete all in batch
        const progressListSnapshots = await getDocs(progressListRef);
        const batch = writeBatch(db);

        progressListSnapshots.forEach((doc) => {
            batch.delete(doc.ref);
        });

        // Commit batch and delete parent doc in parallel
        await Promise.all([
            batch.commit(),
            deleteDoc(questProgressDoc.ref)
        ]);
    } catch (error) {
        console.error("Error deleting quest progress list:", error);
    }
}, [eventID]);
```

Figure F.7. *deleteQuestProgress Function*

```
const fetchLiveLastMonthRanking = async () => {
  try {
    const [facultyID, studentID] = await Promise.all([
      getItem("facultyID"),
      getItem("studentID"),
    ]);

    const leaderboardSnapshot = await getDocs(
      query(collection(db, "leaderboard"), where("facultyID", "==", facultyID))
    );

    if (leaderboardSnapshot.empty) return;

    const leaderboardDoc = leaderboardSnapshot.docs[0];

    const lastMonthQuery = query(
      collection(db, "leaderboard", leaderboardDoc.id, "lastMonth"),
      where("studentID", "==", studentID)
    );

    unsubscribeLastMonth = onSnapshot(lastMonthQuery, (snapshot) => {
      if (!snapshot.empty) {
        const data = snapshot.docs[0].data();
        setPreviousMonthRanking(data.rank);
        setDiamondsRewards(data.diamonds);
        setRewardsModalVisible(true);
      }
    });
  } catch (error) {
    console.error("Error fetching last month's ranking:", error);
  }
};
```

Figure F.8. *fetchLiveLastMonthRanking Function*

```
const handleBarCodeScanned = async ({ type, data }) => {
    setScanned(true);

    try {
        const decryptedBytes = CryptoJS.AES.decrypt(data, secretKey);
        const decryptedText = decryptedBytes.toString(CryptoJS.enc.Utf8);

        // Check if decryption gave something non-empty
        if (!decryptedText || decryptedText.trim() === "") {
            throw new Error("Failed to decrypt QR code — possibly invalid or corrupted");
        }

        let parsedData;
        try {
            parsedData = JSON.parse(decryptedText);
        } catch (jsonError) {
            throw new Error("Decryption succeeded but produced invalid JSON");
        }

        if (!parsedData.eventID || !parsedData.timestamp) {
            throw new Error('Invalid QR Code — missing eventID or timestamp');
        }

        if (parsedData.eventID != eventID) {
            setMode('display');
            setScanned(false);
            throw new Error(`Ouh no... You scanned the wrong event attendance QR code`);
        }

        const currentTimestamp = new Date().getTime();
        const timestampDifference = currentTimestamp - parsedData.timestamp;

        if (timestampDifference <= 5000) {
            const location = await Location.getCurrentPositionAsync({});

            const distance = getDistance(
                {
                    latitude: location.coords.latitude,
                    longitude: location.coords.longitude
                },
                {
                    latitude: latitude,
                    longitude: longitude,
                }
            );
```

```
            if (distance <= 150) {
                const registrationRef = doc(db, "registration", registrationID);
                const registrationSnap = await getDoc(registrationRef);

                if (registrationSnap.exists()) {
                    await updateDoc(registrationRef, {
                        isAttended: true,
                        manualAttended: false,
                        attendanceScannedTime: serverTimestamp(),
                    })
                }

                const studentID = await getItem("studentID");

                updateQuestProgress(studentID);

                if (EVENT_TYPE_MAPPING.hasOwnProperty(categoryID)) {
                    updateBadgeProgress(studentID);
                }
            } else {
                setAttendanceFailureModalContent({
                    title: "You're Too Far Away",
                    subtitle: "You're outside the allowed scanning range. Please move closer to the location and try again.",
                    buttonText: "Understood"
                });
                setAttendanceFailureModalVisible(true);
            }
        } else {
            setAttendanceFailureModalContent({
                title: "QR Code Expired",
                subtitle: "This QR code is no longer valid. Please scan the latest one to proceed.",
                buttonText: "Okay, Got It!"
            });
            setAttendanceFailureModalVisible(true);
        }
    } catch (error) {
        setAttendanceFailureModalContent({
            title: "Invalid QR Code",
            subtitle: "The QR code you scanned is not valid. Please try again with a valid code.",
            buttonText: "Let's Try Again"
        });
        setAttendanceFailureModalVisible(true);
    } finally {
        setScanned(false);
        setMode('display');
    }
}
```

Figure F.9. *handleBarCodeScanned for Attendance-Based Quest*

Figure F.10. *Real-time fetchCurrentEarlyBirdiesNum Function*

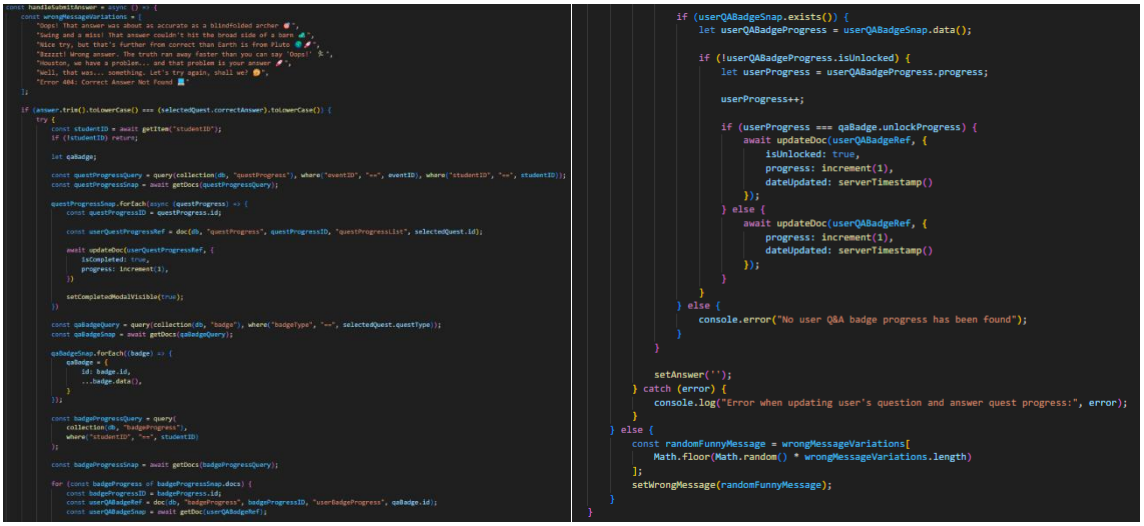


Figure F.11. *Question and Answer (Q&A) Quest Handling Logic*

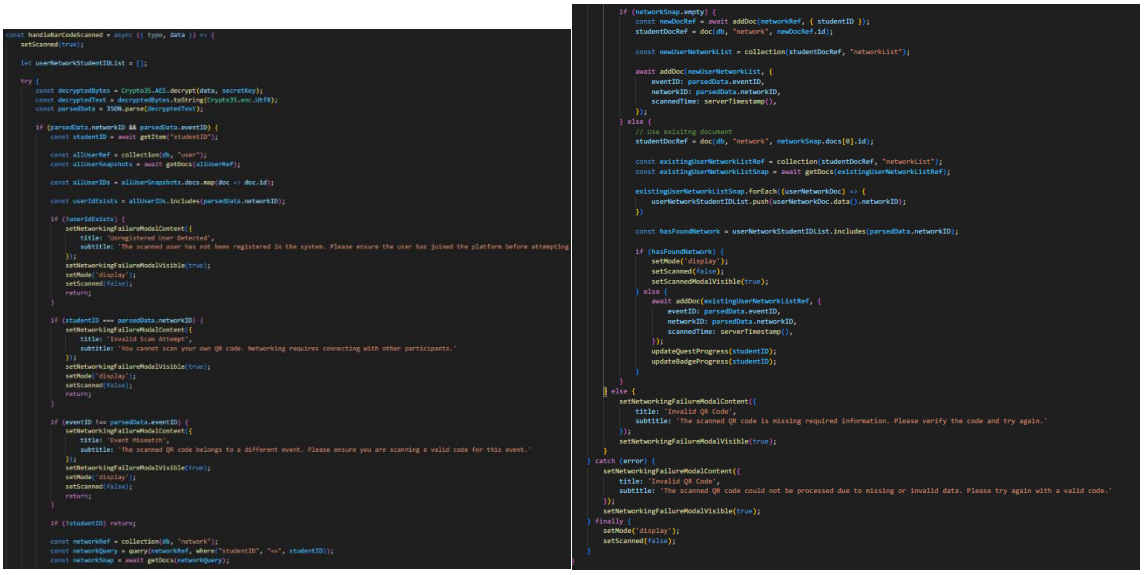


Figure F.12. *handleBarCodeScanned for Networking-Based Quest*

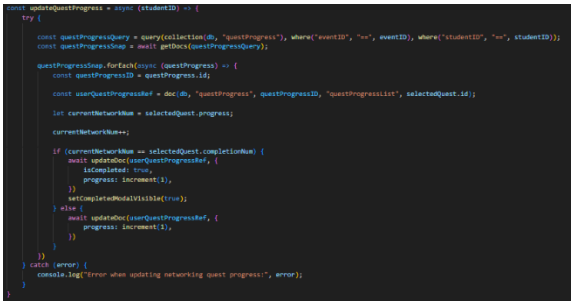


Figure F.13. *Generic Quest Progress Update Function*

Figure F.14. *Networking Badge Progress Update Function*

Figure F.15. *Admin Login Validation Function*

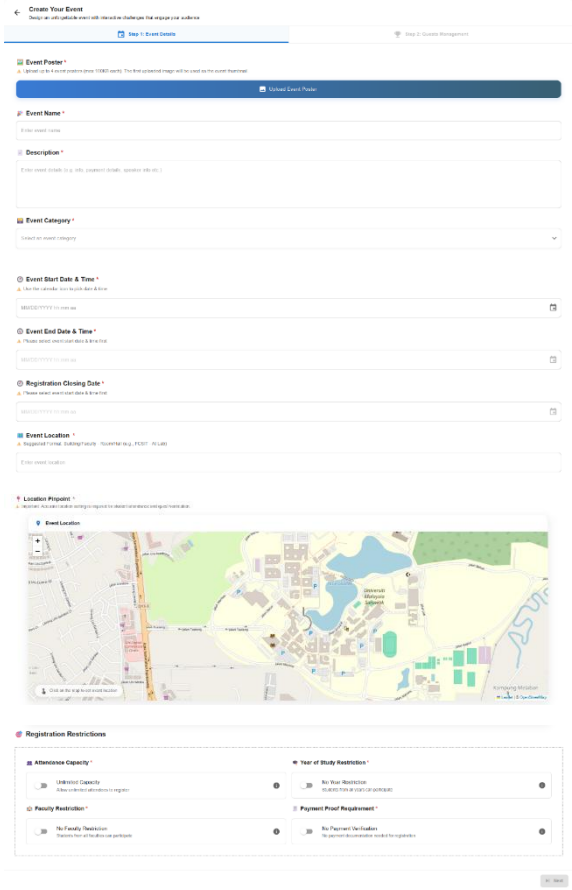


Figure F.16. *Event Details Creation Form*

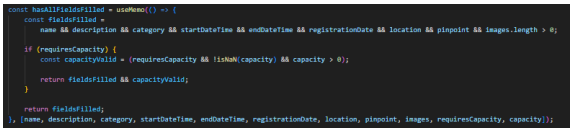


Figure F.17. *Function for Enabling the Next Button in Event Details Creation Form*

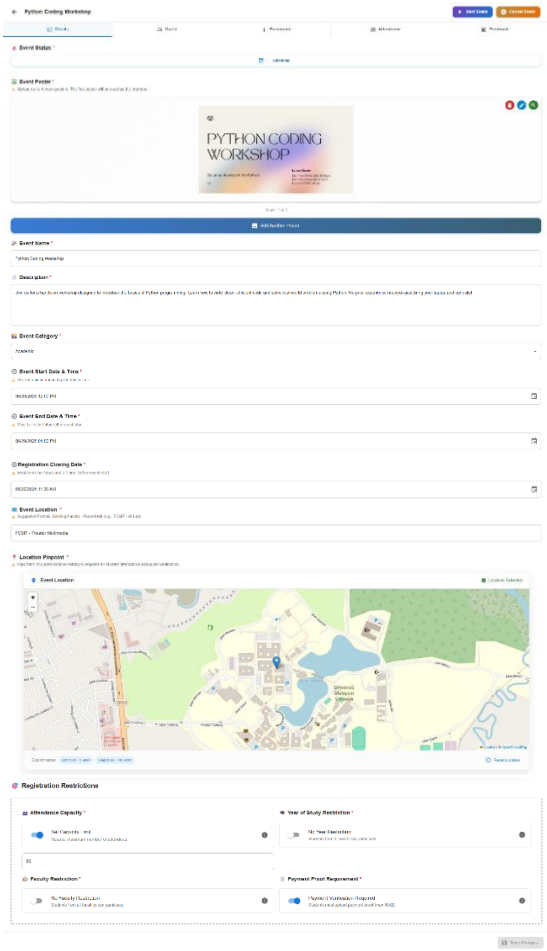


Figure F.18. *Selected Event Details Manager*

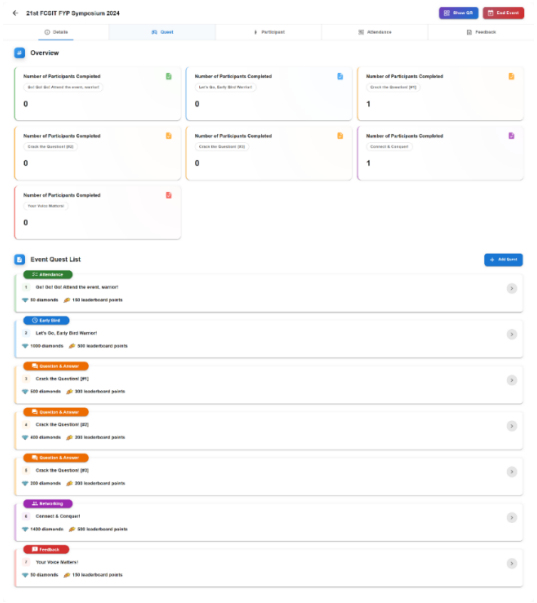


Figure F.19. *Selected Event's Quest Manager*

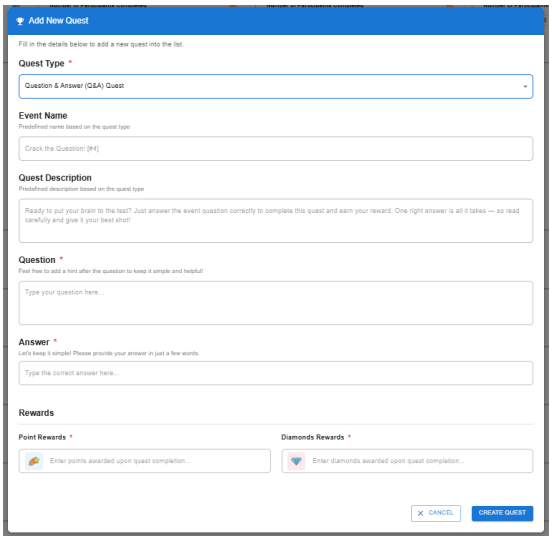


Figure F.20. *Quest Addition Form*

```
const handleVerifyAttendance = async () => {
    if (!reason) {
        console.log("No reason")
        setReasonError("This field cannot be empty!");
        return;
    }

    try {
        const registrationRef = doc(db, "registration", selectedRegistrationID);

        await updateDoc(registrationRef, {
            isAttended: true,
            manualAttended: true,
            reason: reason,
            attendanceScannedTime: serverTimestamp()
        });

        const questQuery = query(collection(db, "quest"), where("eventID", "==", eventID));
        const questSnapshot = await getDocs(questQuery);

        if (questSnapshot.empty) {
            console.error("Something went wrong when retrieving event quest.");
            return;
        }

        const questDoc = questSnapshot.docs[0];
        const questListID = questDoc.id;

        const questListRef = collection(db, "quest", questListID, "questList");
        const attendanceQuestQuery = query(questListRef, where("questType", "==", "attendance"));
        const attendanceQuestSnapshot = await getDocs(attendanceQuestQuery);

        if (attendanceQuestSnapshot.empty) {
            console.error("Something went wrong when retrieving event attendance quest.");
            return;
        }

        const attendanceQuestDoc = attendanceQuestSnapshot.docs[0];
        const attendanceQuestID = attendanceQuestDoc.id;

        const userQuestQuery = query(collection(db, "questProgress"), where("eventID", "==", eventID), where("studentID", "==", selectedStudentID));
        const userQuestSnapshot = await getDocs(userQuestQuery);

        if (userQuestSnapshot.empty) {
            console.error("Something went wrong when retrieving user quest list.");
            return;
        }

        const userQuestDoc = userQuestSnapshot.docs[0];
        const userQuestID = userQuestDoc.id;

        const userAttendanceQuestQuery = query(collection(db, "questProgress", userQuestID, "questProgressList"), where("questID", "==", attendanceQuestID));
        const userAttendanceQuestSnapshot = await getDocs(userAttendanceQuestQuery);

        if (userAttendanceQuestSnapshot.empty) {
            console.error("Something went wrong when retrieving user attendance quest progress.");
            return;
        }

        const userAttendanceQuestDoc = userAttendanceQuestSnapshot.docs[0];
        const userAttendanceQuestID = userAttendanceQuestDoc.id;
        const { isCompleted, progress, rewardsClaimed } = userAttendanceQuestDoc.data();

        const userAttendanceQuestRef = doc(db, "questProgress", userQuestID, "questProgressList", userAttendanceQuestID);

        if (!isCompleted && progress === 0 && !rewardsClaimed) {
            await updateDoc(userAttendanceQuestRef, {
                isCompleted: true,
                progress: increment(1)
            })

            setSnackbarContent({
                msg: "Student's attendance and the corresponding quest status is updated successfully.",
                type: 'success'
            })
            setSnackbarOpen(true);
        } else {
            setSnackbarContent({
                msg: "Student's attendance status is updated successfully.",
                type: 'success'
            })
            setSnackbarOpen(true);
        }
    } catch (err) {
        setSnackbarContent({
            msg: "Something went wrong when updating student's attendance and quest status.",
            type: 'error'
        })
        setSnackbarOpen(true);
    } finally {
        setReason("");
        setSelectedRegistrationID("");
        setSelectedStudentID("")
        setDialogOpen(false);
    }
}
```

Figure F.21. *Attendance-Based Quest Auto Completion Through Manual Attendance*

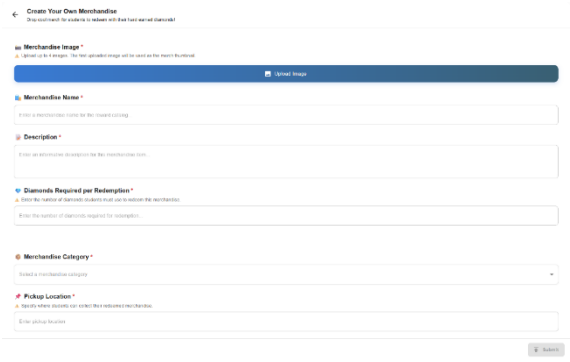


Figure F.22. *Merchandise Item Addition Form*

```
const handleSubmit = async (e) => {
    e.preventDefault();
    setSubmitted(true);

    const formErrors = {};

    if (!name) formErrors.name = 'Name is required';
    if (!description) formErrors.description = 'Description is required';
    if (!diamondsNeeded) formErrors.diamondsNeeded = 'Diamonds needed is required';
    if (!locationName) formErrors.locationName = 'Collection location is required';
    if (!category) formErrors.category = 'Category is required';
    if (category === 'Clothing' && sizes.length === 0) formErrors.sizes = 'At least one size is required';
    if (images.length === 0) formErrors.images = 'At least one image is required';

    setErrors(formErrors);

    if (Object.keys(formErrors).length === 0) {
        const adminData = await getItem("admin");
        const parsedAdminData = JSON.parse(adminData);

        const merchMergedData = {
            name,
            description,
            adminID: parsedAdminData.facultyID,
            diamondsToRedeem: Number(diamondsNeeded),
            collectionLocationName: locationName,
            category,
            images: images.map(image => image.preview),
            available: true,
        }

        if (category === "Clothing") merchMergedData.sizes = sizes;

        try {
            const merchandiseRef = collection(db, "merchandise");

            await addDoc(merchandiseRef, merchMergedData);

            setSnackbarOpen(true);
            setSnackbarContent({ msg: 'Merchandise created successfully!', type: 'success' });
            setTimeout(() => {
                navigate("/merchandise");
            }, 1500)
        } catch (error) {
            console.error("Something went wrong when creating merchandise:", error);
        }
    }
};
```

Figure F.23. *Code Snippet for Handling Merchandise Creation Submission*

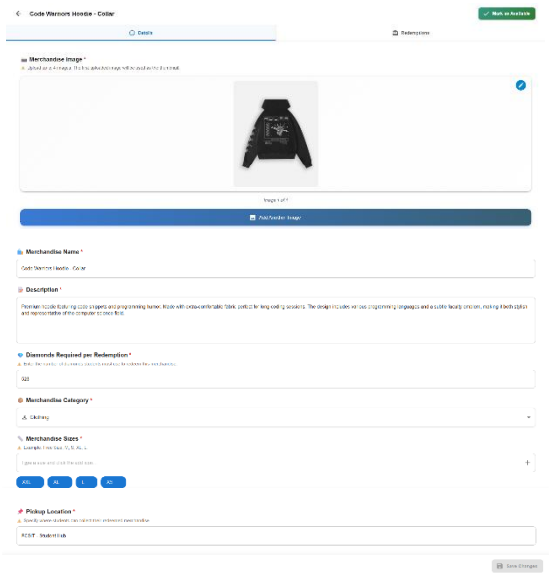


Figure F.24. *Merchandise Viewing/Editing Form*

Appendix G: Test Cases

Table G.1. *Student - Sign In Test Cases*

Feature Name:	**Student can sign in.**					
Test Case Description:	Verify that a student can successfully sign in using valid credentials and is prevented from signing in with invalid credentials, unverified email, or admin account.					
Testing Objective:	To ensure the Sign In functionality works as intended for students, including error handling for invalid input, unverified emails, and admin access restrictions.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_SI_01	Student signs in with valid credentials and verified email	1. Open the Sign In screen. 2. Enter a valid student email and password. 3. Tap “Login”	Email: 79893@siswa.unimas.my Password: Running#Man1	Student should be signed in and accessed to the app functions.	Student is signed in and accessed to the app functions.	Pass
STU_SI_02	Student attempts sign in with invalid email format	1. Enter an invalid email and any password. 2. Tap “Login”	Email: abc Password: ABCD	Error message: "Invalid email format." should be shown under the email field.	Error message: "Invalid email format." is shown under the email field.	Pass
STU_SI_03	Student attempts sign in with incorrect password	1. Enter a valid student email and incorrect password. 2. Tap "Login"	Email: 79893@siswa.unimas.my Password: RunningMan1	Error message: "Incorrect password. Please try again." should be shown under the password field.	Error message: "Incorrect password. Please try again." is shown under the password field.	Pass
STU_SI_04	Student attempts sign in with	1. Enter an unregistered email and any password. 2. Tap "Login"	Email: 99999@siswa.unimas.my Password: any	Error message: "No account found with this email. Please sign up	Error message: "No account found with this email. Please sign up	Pass

	unregistered email			first!" should be shown under the email field.	first!" is shown under the email field.	
STU_SI_05	Student attempts sign in with empty fields.	1. Leave email and/or password blank. 2. Tap "Login"	Email: <blank> Password: <blank>	Error messages: "Email cannot be empty" and/or "Password cannot be empty" should be shown under respective fields.	Error messages: "Email cannot be empty" and/or "Password cannot be empty" shown under respective fields.	Pass
STU_SI_06	Student attempts sign in with unverified email	1. Enter a valid student email (not yet verified) and password. 2. Tap "Login"	Email: 81448@siswa.unimas.my Password: Vin_UniEXP	Modal should be appeared: "Whoa there, Warrior! Looks like you haven't verified your email yet!..."	Modal appears: "Whoa there, Warrior! Looks like you haven't verified your email yet!..."	Pass
STU_SI_07	Admin attempts to sign in on student portal	1. Enter a valid admin email and password. 2. Tap "Login"	Email: fcsit_uniexp@gmail.com Password: FCSIT_UniEXP25	Modal should be appeared: "Whoa there, Admin! ⚠ Access Denied: This level is for students only..."	Modal appears: "Whoa there, Admin! ⚠ Access Denied: This level is for students only..."	Pass

Table G.2. *Student - Sign Up Test Cases*

Feature Name:	**Students can sign up.**
Test Case Description:	Verify that a student can successfully register for an account using valid details and is prevented from registering with invalid or incomplete information. Ensure that a verification email is sent and the correct modal is displayed.
Testing Objectives:	To ensure the Sign-Up functionality works as intended for students, including validation, error handling, and email verification flow.

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status
STU_SU_01	Student signs up with valid details	1. Navigate to sign up screen. 2. Enter valid first name, last name, email, password, select faculty and year. 3. Tap "Sign Up"	**First Name**: Eugene **Last Name**: Wan **Email**: 79354@siswa.unimas.my **Password**: Eugene_UniEXP **Faculty**: FCSIT – Faculty of Computer Science & Information Technology **Year**: 4	Verification email should be sent, "Verification Email Sent!" modal should be appeared, user can tap "I'm on it!" to return to Sign In.	Verification email is sent, "Verification Email Sent!" modal appears, user can tap "I'm on it!" to return to Sign In.	Pass
STU_SU_02	Student attempts sign up with missing required fields	1. Leave one or more fields blank. 2. Tap "Sign Up"	First Name: (blank) Other fields: valid	Error message: "First name is required" should be shown. No account should be created.	Error message: "First name is required" shown under the field. No account is created.	Pass
STU_SU_03	Student enters invalid email format	1. Enter an invalid email. 2. Tap "Sign Up"	Email: johndoe@gmail.com	Error message: "Email must be in format: {matric_num}@ siswa.unimas.my" shown. No account is created.	Error message: "Email must be in format: {matric_num}@ siswa.unimas.my" shown. No account is created.	Pass

STU_SU_04	Student enters password less than 6 characters	1. Enter a password with less than 6 characters. 2. Enter other fields. 3. Tap "Sign Up"	Password: 12345	Error message: "Password must be at least 6 characters" shown under the password field. No account is created.	Error message: "Password must be at least 6 characters" shown under the password field. No account is created.	Pass
STU_SU_05	Student tries to sign up with an already registered email	1. Enter an email already registered. 2. Tap "Sign Up"	Email: 79893@siswa.unimas.my	Error message from Firebase (e.g., "The email address is already in use by another account.") is shown. No new account is created.	Error message from Firebase (e.g., "The email address is already in use by another account.") is shown. No new account is created.	Pass

Table G.3. *Student - Password Reset Test Cases*

Feature Name:	**Students can change their password.**					
Test Case Description:	Verify that a student can request a password reset email using a valid email address and is prevented from requesting with invalid or unregistered emails. Ensure the correct modal and error messages are displayed.					
Testing Objective:	To ensure the Password Reset functionality works as intended, including validation, error handling, and UI feedback.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_PR_01	Student requests password reset with valid email	1. Navigate to password reset screen. 2. Enter a registered student email. 3. Tap "Send Reset Instructions Email"	Email: 79893@siswa.unimas.my	Modal appears: "Houston, We Have a Password!" and reset email is sent to inbox.	Modal appears: "Houston, We Have a Password!" and reset email is sent to inbox.	Pass

STU_PR_02	Student requests password reset with empty email	1. Leave email field blank. 2. Tap "Send Reset Instructions Email"	Email: (blank)	Error message: "Email cannot be empty" shown. No email sent.	Error message: "Email cannot be empty" shown. No email sent.	Pass
STU_PR_03	Student requests password reset with unregistered email	1. Enter an unregistered email. 2. Tap "Send Reset Instructions Email"	Email: 99999@siswa.unimas.my	Error message: "No account found with this email." shown. No email sent.	Error message: "No account found with this email." shown. No email sent.	Pass

Table G.4. *Student - View Leaderboard Test Cases*

Feature Name:	**Students can see their ranking in the faculty leaderboard.**					
Test Case Description:	Verify that a student can view the leaderboard, see the top 3 students, their own rank, and the full ranking list for their faculty. Ensure correct data, UI feedback, and modal information.					
Testing Objective:	To ensure the leaderboard displays accurate, real-time ranking data, correct UI elements, and provides additional information via modals.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_VL_01	Student views leaderboard for their faculty	1. Navigate to leaderboard screen.	Student with valid faculty and points	Leaderboard should display faculty name, top 3 students on podium, remaining students in list, student's own rank highlighted	Leaderboard displays faculty name, top 3 students on podium, remaining students in list, student's own rank highlighted	Pass

STU_VL_02	Student opens leaderboard info modal	1. On Leaderboard screen, tap info icon	-	Modal appears with leaderboard info, rewards, and points explanation	Modal appears with leaderboard info, rewards, and points explanation	Pass
STU_VL_03	Student views previous month's ranking rewards	1. Navigate to Leaderboard when it is a new month.	Student with previous month entry.	Rewards modal appears showing last month's rank and diamonds earned	Rewards modal appears showing last month's rank and diamonds earned	Pass

Table G.5. *Student - Complete Quest (Attendance-Based Quest) Test Cases*

Feature Name:	**Students can complete attendance-based quest of an event.**					
Test Case Description:	Verify that a student can complete an attendance-based quest by scanning a valid event QR code within the allowed time and location, and is prevented from completing the quest with invalid, expired, or out-of-range QR codes. Ensure correct feedback, error handling, and reward claiming.					
Testing Objective:	To ensure the complete quest flow (for attendance-based quests) works as intended, including QR scanning, validation, error handling, quest progress update, and reward claiming.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_AQ_01	Student completes attendance quest with valid QR code and within 150 metres from event location	1. Navigate to Event Quests screen. 2. Select attendance quest. 3. Tap "Scan Attendance QR" 4. Scan valid QR code at event location within allowed time.	Valid event QR code, correct location (within 150 metres from event location), within 5 seconds of QR generation	Quest is marked as completed, "Quest Completed" modal appears, student can claim rewards, diamonds and points are added to student account, leaderboard is updated, button is disabled after claim.	Quest is marked as completed, "Quest Completed" modal appears, student can claim rewards, diamonds and points are added to student account, leaderboard is updated,	Pass

		5. Tap "Claim Rewards" button			button is disabled after claim.	
STU_AQ_02	Student scans expired QR code	1. Repeat step 1 to step 3 in STU_AQ_01 2. Scan QR code generated more than 5 seconds.	Expired event QR code (e.g. QR photo taken by the camera)	"QR Code Expired" modal appears, camera turned off quest not completed	"QR Code Expired" modal appears, camera turned off quest not completed	Pass
STU_AQ_03	Student scans invalid or corrupted QR code	1. Repeat step 1 to step 3 in STU_AQ_01 2. Scan a random/invalid QR code	Invalid QR code	"Invalid QR Code" modal appears, camera turned off quest not completed	"Invalid QR Code" modal appears, camera turned off quest not completed	Pass
STU_AQ_04	Student scans QR code for wrong event	1. Repeat step 1 to step 3 in STU_AQ_01 2. Scan QR code for a different event	QR code for another event	"Invalid QR Code" or "Wrong event" modal appears, camera turned off, quest not completed	"Invalid QR Code" or "Wrong event" modal appears, camera turned off, quest not completed	Pass
STU_AQ_05	Student scans valid QR code but is out of allowed location range	1. Repeat step 1 to step 3 in STU_AQ_01 2. Scan valid QR code while more than 150m from event location	Valid event QR code, wrong location (e.g. QR live share by friends while not being at the event location)	"You're Too Far Away" modal appears, camera turned off, quest not completed	"You're Too Far Away" modal appears, camera turned off, quest not completed	Pass

Table G.6. *Student - Complete Quest (Early Bird Attendance Quest) Test Cases*

Feature Name:	**Students can complete early-bird attendance quest of an event.**
Test Case Description:	Verify that a student can complete the Early-Bird Attendance Quest by being among the first N attendees to check in (attendance-based) and can claim rewards if eligible. Ensure correct quest progress, reward claiming, and UI feedback.
Testing Objective:	

		To ensure the Early-Bird Attendance Quest flow works as intended, including real-time attendee tracking, eligibility, quest completion, reward claiming, and error handling. This quest relies on successful attendance (check-in) and is only available to the first N students.				
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_EQ_01	Student completes Early-Bird quest as eligible attendee	1. Attend event and check in (attendance-based quest) 2. Be among first N attendees. 3. Open Event Quests screen 4. Select Early-Bird quest. 5. Tap “Claim Rewards”	Student is among first N to check in	Quest is marked as completed, "Claim Rewards" button is enabled, student can claim rewards along with the rewards animation	Quest is marked as completed, "Claim Rewards" button is enabled, student can claim rewards along with the rewards animation	Pass
STU_EQ_02	Student attempts Early-Bird quest after N slots filled	1. Attend event and check in after N students have already checked in. 2. Open Event Quests screen 3. Select Early-Bird quest	Student is not among first N	Quest is marked as failed, "Claim Rewards" button is not shown	Quest is marked as failed, "Claim Rewards" button is not shown	Pass

Table G.7. *Student - Complete Quest (Question & Answer Quest) Test Cases*

Feature Name:	**Students can complete question & answer (Q&A) quest of an event.**
Test Case Description:	Verify that a student can complete a Question & Answer Quest by submitting the correct answer and is prevented from completing the quest with an incorrect or empty answer. Ensure correct quest progress, reward claiming, badge progress, and UI feedback.

Testing Objective:		To ensure the Question & Answer Quest flow works as intended, including answer validation, error handling, quest completion, reward claiming, badge progress, and UI feedback.				
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_QA_01	Student completes Early-Bird quest as eligible attendee	1. Navigate to Event Quests screen. 2. Select Q&A quest. 3. Enter correct answer. 4. Tap "Submit Answer" 5. Tap “Claim Rewards” upon completion	Correct answer for the quest	Quest is marked as completed, "Quest Completed" modal appears, answer is shown, "Claim Rewards" button is enabled, diamonds and points are added to student account, leaderboard is updated, button is disabled after claim, animation shown	Quest is marked as completed, "Quest Completed" modal appears, answer is shown, "Claim Rewards" button is enabled, diamonds and points are added to student account, leaderboard is updated, button is disabled after claim, animation shown	Pass
STU_QA_02	Student submits incorrect answer	1. Repeat step 1 to step 2 in STU_QA_01 2. Enter incorrect answer. 3. Tap "Submit Answer”	Incorrect answer	Funny error message is shown, quest not completed, can try again	Funny error message is shown, quest not completed, can try again	Pass
STU_QA_03	Student tries to submit empty answer	1. Repeat step 1 to step 2 in STU_QA_01 2. Leave answer blank	Answer: (blank)	"Submit Answer" button is disabled, no submission occurs	"Submit Answer" button is disabled, no submission occurs	Pass

Table G.8. *Student - Complete Quest (Networking Quest) Test Cases*

Feature Name:	**Students can complete networking quest of an event.**
Test Case Description:	

Testing Objective:		Verify that a student can complete a Networking Quest by scanning another participant's valid QR code for the same event, and is prevented from completing the quest with invalid, self, or mismatched QR codes. Ensure correct quest progress, reward claiming, badge progress, and UI feedback. To ensure the Networking Quest flow works as intended, including QR scanning, validation, error handling, quest progress update, reward claiming, badge progress, and UI feedback.				
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_NQ_01	Student progresses Networking quest by scanning valid QR	1. Navigate to Event Quests screen. 2. Select Networking quest. 3. Tap "Scan Other's QR" 4. Scan another participant's valid QR code for the same event	Valid and unique participant's QR code	Quest progress increases	Quest progress increases	Pass
STU_NQ_02	Student completes Networking quest by scanning required N networks	1. Tap "Scan Other's QR" 2. Scan last required valid network QR code	Valid and unique participant's QR code	"Quest Completed" modal appears, rewards are added to student account, button is disabled after claim, animation shown	"Quest Completed" modal appears, rewards are added to student account, button is disabled after claim, animation shown	Pass
STU_NQ_03	Student scans their own QR code	1. Scan own QR code	Own QR code	"Invalid Scan Attempt" modal appears, quest not completed	"Invalid Scan Attempt" modal appears, quest not completed	Pass

STU_NQ_04	Student scans QR code for different event	1. Scan QR code from another event	QR code for different event	"Event Mismatch" modal appears, quest not completed	"Event Mismatch" modal appears, quest not completed	Pass
STU_NQ_05	Student scans invalid or corrupted QR code	1. Scan random/invalid QR code	Invalid QR code	"Invalid QR Code" modal appears, quest not completed	"Invalid QR Code" modal appears, quest not completed	Pass
STU_NQ_06	Student scans QR code of unregistered user	1. Scan QR code of user not registered for the event	QR code of unregistered user	"Unregistered User Detected" modal appears, quest not completed	"Unregistered User Detected" modal appears, quest not completed	Pass
STU_NQ_07	Student scans the same participant's QR code twice	1. Repeat step 1 to step 3 in STU_NQ_01 2. Scan valid QR code of existing network twice	QR code of the existing network user ID	Modal "Connection Already Established" shown, quest no progress	Modal "Connection Already Established" shown, quest no progress	Pass

Table G.9. *Student - Complete Quest (Feedback-Driven Quest) Test Cases*

Feature Name:	**Students can complete feedback-driven quest of an event.**					
Test Case Description:	Verify that a student can complete a Feedback-Driven Quest by submitting valid feedback form and is prevented from completing the quest with incomplete or invalid feedback.					
Testing Objective:	To ensure the Feedback-Driven Quest flow works as intended, including form validation, error handling, quest completion, reward claiming, badge progress, and UI feedback.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_FQ_01	Student completes	1. Navigate to Event Quests screen.	Valid ratings (1-5) for both	Quest is marked as completed, "Quest	Quest is marked as completed, "Quest	Pass

	Feedback quest with valid feedback	2. Select Feedback quest. 3. Tap "Fill in Feedback Form" 4. Complete all required fields. 5. Tap "Submit"	questions, non-empty feedback text	Completed" modal appears, "Claim Rewards" button is enabled	Completed" modal appears, "Claim Rewards" button is enabled	
STU_FQ_01	Student tries to submit feedback form with missing required fields	1. Repeat step 1 to step 3 in STU_FQ_01 2. Leave one or more required fields blank	Missing rating or feedback text	The “Submit” button is disabled	The “Submit” button is disabled	Pass
STU_FQ_02	Student claims Feedback quest rewards and receives badge progress after completion	1. Complete Feedback quest 2. Tap “Claim Rewards”	-	Diamonds and points are added to student account and leaderboard, button is disabled after claim, animation shown, badge progress is updated, badge unlocked if threshold reached	Diamonds and points are added to student account and leaderboard, button is disabled after claim, animation shown, badge progress is updated, badge unlocked if threshold reached	Pass

Table G.10. *Student - View Badge Test Cases*

Feature Name:	**Students can view the earned achievement badge.**					
Test Case Description:	Verify that a student can view their achievement badges, see badge details (including progress, unlock status, and description), and distinguish between locked and unlocked badges.					
Testing Objective:	To ensure the badge viewing feature works as intended, including badge list display, navigation to badge details, correct progress calculation, and UI feedback for locked/unlocked badges.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**

STU_VB_01	Student views badge list on Profile screen	1. Navigate to Profile screen	-	All badges (locked and unlocked) are displayed with correct images and names	All badges (locked and unlocked) are displayed with correct images and names	Pass
STU_VB_02	Student views unlocked badge details	1. On Profile screen, tap an unlocked badge	Unlocked badge	Badge details screen opens, showing badge image, name, description, progress bar at 100%, and "Unlocked" status	Badge details screen opens, showing badge image, name, description, progress bar at 100%, and "Unlocked" status	Pass
STU_VB_03	Student views locked badge details	1. On Profile screen, tap a locked badge	Locked badge	Badge details screen opens, showing badge image (dimmed), name, description, progress bar < 100%, and "Locked" status	Badge details screen opens, showing badge image (dimmed), name, description, progress bar < 100%, and "Locked" status	Pass
STU_VB_04	Student sees badge progress update after quest completion	1. Complete a quest that contributes to badge progress. 2. Navigate to Profile screen. 3. Tap relevant badge	Badge with progress	Badge progress is updated, progress bar reflects new value, badge unlocks if threshold reached	Badge progress is updated, progress bar reflects new value, badge unlocks if threshold reached	Pass

Table G.11. *Student - Register Event Test Cases*

Feature Name:	**Students can browse and register for events.**
Test Case Description:	Verify that a student can browse available events, view event details, and register for an event if eligible. Ensure registration is prevented for full, restricted, or already registered events, and that the registration form and feedback are correct.

Testing Objective:		To ensure the event registration flow works as intended, including event listing, eligibility checks, registration form validation, and UI feedback.				
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_RE_01	Student browses event listing	1. Navigate to Event Listing screen	-	List of available events is displayed, or empty state UI is displayed if there are no events	List of available events is displayed, or empty state UI is displayed if there are no events	Pass
STU_RE_02	Student filters events by category	1. Navigate to Event Listing screen. 2. Select a category from the category filter	Category: "Sports"	Only events in the "Sports" category are displayed, or empty state UI is displayed if there are no events for this category	Only events in the "Sports" category are displayed, or empty state UI is displayed if there are no events for this category	Pass
STU_RE_03	Student searches for an event by keyword	1. Navigate to Event Listing screen. 2. Enter a keyword in the search bar	Search: "volunteer"	Events matching the keyword are displayed in the list or empty state UI is displayed if there are no matching events	Events matching the keyword are displayed in the list or empty state UI is displayed if there are no matching events	Pass
STU_RE_04	Student views event details	1. On Event Listing, tap an event card	Any event	Event Details screen opens, showing event info, images, location, and registration button	Event Details screen opens, showing event info, images, location, and registration button	Pass
STU_RE_05	Student registers for event not requiring payment proof	1. On event listing, select an event that does not requires payment. 2. Tap "Register Now"	Event without payment requirement	Registration is successful, confirmation modal appears, student is added to event	Registration is successful, confirmation modal appears, student is added to event	Pass

		3. Tap "Submit"				
STU_RE_06	Student registers for event requiring payment proof	1. On Event Listing, select an event that requires payment. 2. Tap "Register Now" 3. Upload valid payment proof image. 4. Tap "Submit"	Event with payment requirement, valid image	Registration is successful, confirmation modal appears, student is added to event	Registration is successful, confirmation modal appears, student is added to event	Pass
STU_RE_07	Student tries to register for sudden changed restricted event (faculty/year)	1. On Event Details, navigate to the footer of the screen for register	Event with restrictions	Restriction message shown, registration button is hidden	Restriction message shown, registration button is hidden	Pass
STU_RE_08	Student tries to register for full event	1. On Event Details, navigate to the footer of the screen for register	Event at full capacity	"Maximum Capacity" message shown, registration button is hidden	"Maximum Capacity" message shown, registration button is hidden	Pass
STU_RE_09	Student attempts to submit registration without uploading payment proof	1. On Event Details, tap "Register Now" for event requiring payment. 2. Do not upload receipt	Event requiring payment proof	The registration button is disabled	The registration button is disabled	Pass

Table G.12. *Student - Manage Registered Event (Agenda) Test Cases*

Feature Name: **Test Case Description:**	**Students can manage registered events through the agenda.**

Testing Objective:		Verify that a student can view, track, and manage their registered events through the agenda/calendar, including viewing event details, registration status, attendance, payment proof, and cancelling registration if eligible. To ensure the agenda/calendar feature allows students to effectively manage their registered events, including real-time updates, navigation to event details, and registration management actions.				
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_ME_01	Student views registered events in agenda	1. Navigate to Agenda screen	Student with registered events	Registered events are displayed on the calendar by date, with event cards showing name, time, location, and status	Registered events are displayed on the calendar by date, with event cards showing name, time, location, and status	Pass
STU_ME_02	Student views empty agenda and navigates to browse events	1. Navigate to Agenda screen. 2. Tap "Browse Available Events"	No registered events	"No Events Scheduled" message and empty state UI are displayed, navigates to Event Listing screen	"No Events Scheduled" message and empty state UI are displayed, navigates to Event Listing screen	Pass
STU_ME_03	Student views event details from agenda	1. On Agenda, tap an event card	Any registered event	Registered Event Details screen opens, showing event info, registration status, attendance, payment proof (if required), and location map	Registered Event Details screen opens, showing event info, registration status, attendance, payment proof (if required), and location map	Pass
STU_ME_04	Student views registration related and attendance status	1. On Registered Event Details, view status sections	Event with various statuses	Registration status ("Verified" / "Pending"), attendance ("Attended" / "Not Yet Attended"), and payment proof ("Submitted" /	Registration status ("Verified" / "Pending"), attendance ("Attended" / "Not Yet Attended"), and payment proof ("Submitted" /	Pass

				"Required") are displayed with correct badges and messages	"Required") are displayed with correct badges and messages	
STU_ME_05	Student cancels registration before event starts	1. On Registered Event Details, tap "Cancel Registration" (if >1 hour before event) 2. Confirm cancellation	Eligible event	Registration is cancelled, event is removed from agenda, confirmation modal shown	Registration is cancelled, event is removed from agenda, confirmation modal shown	Pass
STU_ME_06	Student attempts to cancel registration within 1 hour of event	1. On Registered Event Details, check for "Cancel Registration" button within 1 hour of event start	Event starting soon	"Cancel Registration" button is not visible.	"Cancel Registration" button is not visible	Pass

Table G.13. *Student - Manage Merchandise Redemption Test Cases*

Feature Name:	**Students can redeem merchandise using virtual currency (diamonds).**					
Test Case Description:	Verify that a student can browse, search, and filter available merchandise, view merchandise details, redeem merchandise using diamonds, and manage their redemptions.					
Testing Objective:	To ensure the merchandise redemption and management flow works as intended, including browsing, searching, filtering, viewing details, redemption eligibility, redemption history management, and UI feedback.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_MR_01	Student browses merchandise listing	1. Navigate to Merchandise Listing screen	Student with virtual currency (diamonds)	List of available merchandise is displayed, with categories and search	List of available merchandise is displayed, with categories and search	Pass

STU_MR_02	Student filters merchandise by category	1. On Merchandise Listing, select a category	Category: “Clothing”	Only merchandise in the selected category is displayed	Only merchandise in the selected category is displayed	Pass
STU_MR_03	Student searches for merchandise by keyword	1. On Merchandise Listing, enter a keyword in the search bar	Search: "shirt"	Merchandise matching the keyword is displayed	Merchandise matching the keyword is displayed	Pass
STU_MR_04	Student views merchandise details	1. On Merchandise Listing, tap a merchandise card	Any available merchandise	Merchandise Details screen opens, showing images, description, collection place, and redemption button	Merchandise Details screen opens, showing images, description, collection place, and redemption button	Pass
STU_MR_05	Student redeems clothing merchandise (requires size selection) with sufficient diamonds	1. On Merchandise Listing, select a clothing item. 2. Tap "Redeem" 3. Select a size and quantity to redeem. 4. Confirm redemption	Clothing item, size: "M"	Size selection dropdown is shown; after selecting size and confirming, redemption is successful, size is recorded in redemption details	Size selection dropdown is shown; after selecting size and confirming, redemption is successful, size is recorded in redemption details	Pass
STU_MR_06	Student attempts to redeem clothing merchandise without selecting size	1. On Merchandise Details for clothing, attempt to redeem the item without selecting size	Clothing item, no size selected	The redeem button is disabled, redemption not submitted	The redeem button is disabled, redemption not submitted	Pass
STU_MR_07	Student redeems non-clothing merchandise	1. On Merchandise Listing, select a non-clothing item. 2. Select quantity to redeem. 3. Tap "Redeem"	Non-clothing item	Modal appears upon successful redemption and student’s diamonds are deducted	Modal appears upon successful redemption and student’s diamonds are deducted	Pass

STU_MR_08	Student attempts to redeem merchandise with insufficient diamonds	1. On Merchandise Details, select quantity to redeem. 2. Attempt to redeem	Merchandise price > student's diamonds	Error message: "Not enough diamonds in your balance", redemption not allowed	Error message: "Not enough diamonds in your balance", redemption not allowed	Pass
STU_MR_09	Student views redemption history (uncollected / collected)	1. Navigate to Redemption Listing screen. 2. Switch between "Uncollected" and "Collected" tabs	Student with redemptions	Redemptions are displayed under correct tabs, with their respective details (name, redemption ID, timestamp, selected size etc.)	Redemptions are displayed under correct tabs, with their respective details (name, redemption ID, timestamp, selected size etc.)	Pass
STU_MR_10	Student searches redemption history	1. On Redemption Listing, enter a keyword in the search bar	Search: "shirt"	Only redemptions matching the keyword are displayed	Only redemptions matching the keyword are displayed	Pass

Table G.14. *Student - View Notifications List Test Cases*

Feature Name:	**Students can view notifications list.**					
Test Case Description:	Verify that a student can view their notifications, including loading, empty state, and error handling.					
Testing Objective:	To ensure notifications are fetched, displayed, and updated correctly, and the UI responds appropriately to different states.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_NL_01	Student views notifications with existing data	1. Navigate to navigation list screen. 2. Wait for notifications to load.	Student with notifications in database	Notifications are listed, sorted by newest first, unread are highlighted	Notifications are listed, sorted by newest first, unread are highlighted	Pass

STU_NL_02	Student views notifications when none exist	1. Navigate to notification list screen. 2. Wait for notifications to load.	Student with no notifications	"No notifications yet" message and icon are shown	"No notifications yet" message and icon are shown	Pass
STU_NL_03	Loading indicator is shown while fetching	1. Navigate to notification list screen. 2. Observe UI before notifications appear.	-	Loading spinner and "Loading notifications..." text are visible until data loads	Loading spinner and "Loading notifications..." text are visible until data loads	Pass

Table G.15. *Student - Manage Networking Connections Test Cases*

Feature Name:	**Students can manage networking connections and send messages to them.**					
Test Case Description:	Verify that a student can browse their networking connections (made via networking quests), search and paginate the list, view details of each connection, and start or continue a chat with any connection.					
Testing Objective:	To ensure the networking management feature allows students to effectively browse, search, and interact with their connections, including viewing details and messaging.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
STU_MNC_01	Student browses networking connections	1. Navigate to Profile screen. 2. Look for “Others” section. 3. Press the “Network” card. 4. View the list	Student with networking connections	List of connections is displayed, each with name, faculty, year, event, and profile picture	List of connections is displayed, each with name, faculty, year, event, and profile picture	Pass
STU_MNC_02	Student searches networking connections by name or hobby	1. On Network screen, enter a keyword in the search bar	Search: "Eugene"	Only connections matching the keyword (name or hobby) are displayed	Only connections matching the keyword (name or hobby) are displayed	Pass

STU_MNC_03	Student filters networks by last attended event category	1. On Network screen, select a last attended event category from the list.	Selected Category: "Academic"	Only connections matching the selected last attended event category are displayed.	Only connections matching the selected last attended event category are displayed.	Pass
STU_MNC_04	Student starts a new chat with a connection	1. On Network screen, tap a connection card. 2. On Messaging screen, send a message	Any connection	Messaging screen opens, chat is created if not existing, message is sent and appears in chat history	Messaging screen opens, chat is created if not existing, message is sent and appears in chat history	Pass
STU_MNC_05	Student continues chatting with an existing connection	1. On Network screen, tap a connection card with existing chat. 2. On Messaging screen, send a message	Existing chat	Previous messages are loaded; new message is sent and appears in chat history	Previous messages are loaded; new message is sent and appears in chat history	Pass

Table G.16. *Admin - Sign in Test Cases*

Feature Name:	**Admins/Event Organisers can sign in.**					
Test Case Description:	Verify that the admin can sign in using valid credentials and is prevented from signing in with invalid or incomplete credentials.					
Testing Objective:	To ensure the login form validates input, handles authentication, and provides appropriate feedback for both successful and failed login attempts.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_SI_01	Admin signs in with valid credentials	1. Navigate to Login page. 2. Enter valid admin email and password. 3. Click "Let's Go!"	Email: fcsit.uniexp@gmail.com Password: FCSIT_UniEXP25	Sign in button is disabled, loading indicator appears, admin is authenticated	Sign in button is disabled, loading indicator appears, admin is authenticated	Pass

				and redirected to dashboard	and redirected to dashboard	
AD_SI_02	Admin signs in with invalid email	1. Enter invalid email. 2. Enter any password. 3. Click "Let's Go!"	Email: johndoe@gmail.com Password: any	"Invalid admin credentials" snackbar appears, not authenticated	"Invalid admin credentials" snackbar appears, not authenticated	Pass
AD_SI_03	Admin signs in with invalid password	1. Enter valid admin email. 2. Enter wrong password. 3. Click "Let's Go!"	Email: fcsit.uniexp@gmail.com Password: FCSIT#UniEXP25	"Invalid admin credentials" snackbar appears, not authenticated	"Invalid admin credentials" snackbar appears, not authenticated	Pass
AD_SI_04	Admin submits with empty email	1. Leave email blank. 2. Enter any password. 3. Click "Let's Go!"	Email: "" Password: any	"Email is required" error shown under email field	"Email is required" error shown under email field	Pass
AD_SI_05	Admin submits with invalid email format	1. Enter invalid email format. 2. Enter any password. 3. Click "Let's Go!"	Email: johndoe#.gmail.com Password: any	"Please enter a valid email address" error shown	"Please enter a valid email address" error shown	Pass
AD_SI_06	Admin submits with empty password	1. Enter valid email. 2. Leave password blank. 3. Click "Let's Go!"	Email: fcsit.uniexp@gmail.com Password: “”	"Password is required" error shown under password field	"Password is required" error shown under password field	Pass

Table G.17. *Admin - Forgot Password Test Cases*

Feature Name:		**Admins/Event organisers can reset their password.**				
Test Case Description:		Verify that the admin receives a password reset email in their inbox and can use the link to reset their password.				
Testing Objective:		To ensure the password reset email is sent, contains a valid link, and allows the admin to set a new password.				
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**

AD_FP_01	Admin completes password reset with valid email	1. Navigate to "Forgot Password" page in login page. 2. Enter registered admin email. 3. Click "Send Reset Link" 4. Open email inbox 5. Locate password reset email. 6. Click the reset link. 7. Enter and confirm new password. 8. Submit	Email: fcsit.uniexp@gmail.com New Password: "NewPass25!"	Success message shown, reset email sent, "Check Your Email" screen appears. Link opens reset page. Password is updated. Success message shown. Admin can log in with new password	Success message shown, reset email sent, "Check Your Email" screen appears. Link opens reset page. Password is updated. Success message shown. Admin can log in with new password	Pass
AD_FP_02	Admin requests password reset with unregistered email	1. Enter email not in admin collection. 2. Click "Send Reset Link"	Email: johndoe@gmail.com	"Invalid admin credentials" snackbar appears, no email sent	"Invalid admin credentials" snackbar appears, no email sent	Pass
AD_FP_03	Admin submits with empty email	1. Leave email blank. 2. Click "Send Reset Link"	Email: (blank)	"Email is required" error shown under field	"Email is required" error shown under field	Pass
AD_FP_04	Admin submits with invalid email format	1. Enter invalid email. 2. Click "Send Reset Link"	Email: johndoe#gmail.com	"Please enter a valid email address" error shown	"Please enter a valid email address" error shown	Pass

Table G.18. *Admin - View Event and Quest Statistics Test Cases*

Feature Name:	**Admins/event organisers can view event and quest statistics.**
Test Case Description:	Verify that the admin can view comprehensive statistics for events and quests, including event satisfaction ratings, attendance summaries, yearly event counts by type, quest satisfaction ratings, total quests by type, and quest completion status by type.
Testing Objective:	To ensure the admin dashboard accurately displays all relevant event and quest statistics, with correct calculations, visualizations, and up-to-date data.

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status
AD_VS_01	Admin views event satisfaction rating	1. Navigate to Statistics page. 2. Select "Event Stats" tab, view event satisfaction rating	Events with satisfaction feedback	Average event satisfaction (out of 5) is displayed correctly	Average event satisfaction (out of 5) is displayed correctly	Pass
AD_VS_02	View overall attendance summary	1. On "Event Stats" tab, view attendance section	Events with attendees/absentees	Total attendees and absentees are displayed accurately	Total attendees and absentees are displayed accurately	Pass
AD_VS_03	View yearly event number statistics by type	1. On "Event Stats" tab, view yearly statistics section	Events of various types over 12 months which managed by admin	Yearly event counts by type are shown, with trends and monthly breakdowns	Yearly event counts by type are shown, with trends and monthly breakdowns	Pass
AD_VS_04	View quest satisfaction rating	1. Select "Quest Stats" tab, view quest satisfaction rating	Events with quest satisfaction feedback	Average quest satisfaction (out of 5) is displayed correctly	Average quest satisfaction (out of 5) is displayed correctly	Pass

AD_VS_05	View total quests by type	1. On "Quest Stats" tab, view quest type counts	Events with multiple quest types	Total number of quests for each type is displayed	Total number of quests for each type is displayed	Pass
AD_VS_06	View quest completion status by type	1. On "Quest Stats" tab, view quest completion section	Quests with various completion statuses	For each quest type, overall completion percentage are shown	For each quest type, overall completion percentage are shown	Pass

Table G.19. *Admin - Create Event Test Cases*

Feature Name:	**Admins/event organisers can add new events.**					
Test Case Description:	Verify that the admin can successfully create a new event by filling in all required and optional fields in the event details form, uploading images, setting restrictions, and adding all quest types in the Quest step.					
Testing Objective:	To ensure that the event creation process works as intended, including validation of all fields, image handling, restriction settings, map pinpointing, and integration with the quest creation process.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_CE_01 **(Notes: Cover full integration flow)**	Admin creates event with all valid fields and quests.	1. Navigate to Event Listing page. 2. Tap the "Add Event" button. 3. Upload 1-4 valid images. 4. Fill in all required fields. 5. Set valid start, end, and registration dates/times. 6. Set location and pinpoint on map.	**Name**: "FCSIT Annual Gala" **Description**: "A grand event for FCSIT with networking and fun activities. Entry fee: RM5" **Category**: "Entertainment" **Start Date/Time**: 2025-08-01 18:00 **End Date/Time**:	Event and all quests are created successfully. Success snackbar / message appears. Redirected to event listing, new event is visible in the list with all details and quests	Event and all quests are created successfully. Success snackbar / message appears. Redirected to event listing, new event is visible in the list with all details and quests	Pass

		7. Enable faculty, year, capacity, and payment proof restrictions. 8. Select multiple years. 9. Click “Next”. 10. Proceed to Quest step and add default quests. 11. Click "Submit"	2025-08-01 22:00 **Registration Deadline**: 2025-07-25 23:59 **Location**: Dewan Delima, UNIMAS **Pinpoint**: Latitude - 3.12345 Longitude - 101.67890 **Images**: 4 valid JPEG/PNG files **Year Restriction**: Enabled **Years**: Year 1, Year 2, and Year 3 **Capacity Restriction**: Enabled **Capacity**: 300 **Payment Proof**: Enabled			
AD_CE_02	Admin creates event with missing required fields	1. Leave one or more required fields blank. 2. Try to proceed	Missing name, category, etc.	The “Next” button is disabled to proceed with quest creation step.	The “Next” button is disabled to proceed with quest creation step.	Pass

AD_CE_03	Admin creates event with invalid date/time (start in past)	1. Try to set start date time in the past	Past date time	The date time before the current date time is disabled	The date time before the current date time is disabled	Pass
AD_CE_04	Admin creates event with end date before start date	1. Try to set the end date time before start	End date time < Start date time	The date time before the start date time is disabled	The date time before the start date time is disabled	Pass
AD_CE_05	Create event with registration closing after event start	1. Try to set the registration closing date time within 1 hour or after the event start	Registration closing date time > Start date time	The date time within 1 hour or after the start date time is disabled	The date time within 1 hour or after the start date time is disabled	Pass
AD_CE_06	Admin creates event with no images	1. Do not upload images. 2. Fill other required fields. 3. Try to proceed	No images, other valid fields	The “Next” button is disabled, cannot proceed with quest creation	The “Next” button is disabled, cannot proceed with quest creation	Pass
AD_CE_07	Admin creates event with more than 4 images	1. Try to upload 5 images	Any valid image	Error: "Maximum 4 images allowed"	Error: "Maximum 4 images allowed"	Pass
AD_CE_08	Admin creates event with capacity restriction enabled but no value or 0	1. Enable capacity. 2. Leave capacity blank. 3. Try to proceed	Capacity: ***(blank)*** or **0**	The “Next” button is disabled, cannot proceed with quest creation	The “Next” button is disabled, cannot proceed with quest creation	Pass

AD_CE_09	Admin creates event with year restriction enabled but no years selected	1. Enable year restriction. 2. Do not select years. 3. Try to proceed	Years: ***(blank)***	The "Next" button is disabled, cannot proceed with quest creation	The "Next" button is disabled, cannot proceed with quest creation	Pass
AD_CE_10	Admin creates event with faculty restriction enabled	1. Enable faculty restriction. 2. Proceed	Faculty restriction: true	Proceed with quest creation; only selected faculty can register	Proceed with quest creation; only selected faculty can register	Pass

Table G.20. *Admin - Manage Event Test Cases*

Feature Name:	**Admins/event organisers can manage existing events.**					
Test Case Description:	Verify that the admin can view, edit, and manage all aspects of an event, including event details, quests, participants, attendance, and feedback, as well as control event status (start, end, cancel).					
Testing Objective:	To ensure the admin can fully manage an event's related data, with correct UI feedback, validation, and backend updates.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_ME_01	Admin views event details	1. Navigate to Event Listing page. 2. Select an event. 3. View "Details" tab	"UNIMAS Gala Night" event card	All event details displayed correctly	All event details displayed correctly	Pass
AD_ME_02	Admin edits and saves event details	1. Edit fields (name, description, category, dates, location, restrictions, images) 2. Click "Save Changes"	**Event Name**: "FCSIT Gala Night" to "FCSIT Gala Night 2025"	Changes saved, success snackbar, updated data shown	Changes saved, success snackbar, updated data shown	Pass

AD_ME_03	Admin uploads, replaces, and deletes event images	1. Upload up to 4 images. 2. Replace an image. 3. Delete an image	4 valid images	Images updated, cannot exceed 4, cannot delete the first image (only 1), errors shown if violated	Images updated, cannot exceed 4, cannot delete the first image (only 1), errors shown if violated	Pass
AD_ME_04	Admin edits event restrictions	1. Toggle year / faculty / capacity / payment proof restrictions. 2. If applicable, fill in the required field. 3. Save changes	Enable / disable restrictions	Restrictions updated, only allowed users can register or user needs to upload payment proof when registering event.	Restrictions updated, only allowed users can register or user needs to upload payment proof when registering event	Pass
AD_ME_05	Admin changes event status: Start event	1. Click "Start Event" button at top of the page	Scheduled or Postponed event	Status changes to "Ongoing", QR code page opens	Status changes to "Ongoing", QR code page opens	Pass
AD_ME_06	Admin changes event status: End event	1. Click "End Event" button at top of the page	Ongoing event	Status changes to "Completed", the button list disappears	Status changes to "Completed", the button list disappears	Pass
AD_ME_07	Admin changes event status: Cancel event	1. Click "Cancel Event" button at top of the page	Scheduled or Postponed event	Status changes to "Cancelled", the button list disappears	Status changes to "Cancelled", the button list disappears	Pass
AD_ME_08 **(Notes: Refer "Manage Quests" test cases)**	View and manage quests	1. Go to "Participant" tab. 2. View participant list	-	Quests displayed and managed correctly	Quests displayed and managed correctly	Pass

AD_ME_09 **(Notes: Refer "Manage Participant" test cases)**	View and manage participant	1. Go to "Attendance" tab. 2. View attendance records	-	List displayed, can view details	List displayed, can view details	Pass
AD_ME_10 **(Notes: Refer "Manage Attendance" test cases)**	View and manage attendance	1. Go to "Attendance" tab. 2. View attendance records	-	Records displayed	Records displayed	Pass
AD_ME_11 **(Notes: Refer "Manage Feedback" test cases)**	View and manage feedback	1. Go to "Feedback" tab. 2. View feedback entries	-	Feedback displayed, can view details	Feedback displayed, can view details	Pass
AD_ME_12	Admin attempts to save with invalid data	1. Clear required fields 2. Click "Save Changes"	**Event Name**: (blank)	Error messages shown, changes not saved	Error messages shown, changes not saved	Pass
AD_ME_13	Admin attempts to upload more than 4 images	1. Upload 5 images	5 images or any image to be the fifth image	Error: "Maximum 4 images allowed"	Error: "Maximum 4 images allowed"	Pass
AD_ME_14	Admin attempts to delete the only one image	1. Only 1 image 2. Try to delete	The only image in the poster catalogue	The delete icon at the top right of the image catalogue is hidden	The delete icon at the top right of the image catalogue is hidden	Pass

Table G.21. *Admin - Manage Participant Test Cases*

Feature Name:		**Admins/event organisers can manage the list of participants.**				
Test Case Description:		Verify that the admin can view, verify, and export event participants, as well as view payment proof and participant details.				
Testing Objective:		To ensure the admin can manage all participants for an event, including verifying registrations, viewing payment proof, and exporting participant data.				
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_MP_01	View all participants	1. Navigate to "Participant" tab. 2. View participant list	Event with participants	All registered participants are listed with correct details	All registered participants are listed with correct details	Pass
AD_MP_02	View participant details	1. View on a participant row	-	Participant's name, email, faculty, year, and profile picture are shown	Participant's name, email, faculty, year, and profile picture are shown	Pass
AD_MP_03 **(Notes: Only if payment proof required)**	View payment proof for participant	1. Click "View Proof" for a participant with payment proof	Participant with payment proof	Payment proof image is displayed in modal	Payment proof image is displayed in modal	Pass
AD_MP_04 **(Notes: Only if payment proof required)**	Verify participant registration	1. Go "Unverified" tab. 2. View payment proof 3. Click "Verify" for a participant	-	Participant moves to "Verified" tab, success snackbar shown	Participant moves to "Verified" tab, success snackbar shown	Pass
AD_MP_05	Export verified participants to Excel if verified	1. Go to "Verified" tab. 2. Click "Export"	-	Excel file is downloaded with participant data	Excel file is downloaded with participant data	Pass

	participants exist					

Table G.22. *Admin - Manage Attendance Test Cases*

Feature Name:	**Admins/event organisers can track and validate event attendance.**					
Test Case Description:	Verify that the admin can view, filter, manually verify, and export event attendance, as well as see attendee/absentee details and reasons for manual attendance.					
Testing Objective:	To ensure the admin can manage all attendance records for an event, including marking manual attendance, viewing reasons, and exporting data.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_MA_01	View all attendees	1. Navigate to “Attendance” tab. 2. Select “Attendees” tab	-	All attendees are listed with correct details	All attendees are listed with correct details	Pass
AD_MA_02	View all absentees	1. Select "Absentees" tab	-	All absentees are listed with correct details	All absentees are listed with correct details	Pass
AD_MA_03	Mark manual attendance for absentee	1. Go to "Absentees" tab. 2. Click "Manual Attendance" for a student. 3. Enter a reason in opened dialog form. 4. Confirm	**Reason**: Technical issue	Student moves to 'Attendees' tab, method shows 'Manual', reason is saved, and attendance quest is marked as completed for the student.	Student moves to 'Attendees' tab, method shows 'Manual', reason is saved, and attendance quest is marked as completed for the student.	Pass
AD_MA_04	Attempt manual attendance with empty reason	1. Click "Manual Attendance" for absentee. 2. Leave reason blank. 3. Confirm	**Reason**: *(blank)*	Error: "This field cannot be empty!", attendance not updated	Error: "This field cannot be empty!", attendance not updated	Pass

AD_MA_05	View method and reason for manual attendance	1. In "Attendees" tab, find student with manual attendance	Student attended the event through manual attendance	Method shows "Manual", tooltip displays reason	Method shows "Manual", tooltip displays reason	Pass
AD_MA_06	Export attendees to Excel if attendees existed	1. In "Attendees" tab, click "Export"	-	Excel file downloaded with correct data	Excel file downloaded with correct data	Pass

Table G.23. *Admin - Create Quest Test Cases*

Feature Name:	**Admins/event organisers can add event-specific quests.**					
Test Case Description:	Verify that the admin can create all supported quest types (Early Bird, Q&A, Networking) during event creation, with proper validation, data entry, and integration into the event.					
Testing Objective:	To ensure that each quest type can be created with valid data, that validation is enforced for required fields, and that quests are correctly associated with the event.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_CQ_01	Admin adds Early Bird quest with valid data	1. Click "Add New Quest" 2. Select "Early Bird Attendance Quest" 3. Enter max early bird attendees, points, diamonds. 4. Complete all steps and save	**Max Early Bird**: 20, **Points**: 100, **Diamonds**: 30	Quest appears in quest list, no errors	Quest appears in quest list, no errors	Pass
AD_CQ_02	Admin adds Q&A quest with valid data.	1. Click "Add New Quest" 2. Select "Q&A Quest" 3. Enter question, answer, points, diamonds.	**Question**: "What is the theme?" **Answer**:	Quest appears in quest list, no errors	Quest appears in quest list, no errors	Pass

	(Notes: Can add multiple Q&A quests)	4. Complete all steps and save	"Unity" **Points**: 80 **Diamonds**: 20			
AD_CQ_03	Admin adds Networking quest with valid data	1. Click "Add New Quest" 2. Select "Networking Quest" 3. Enter required connections, points, diamonds. 4. Complete all steps and save	**Connections**: 5 **Points**: 60 **Diamonds**: 15	Quest appears in quest list, no errors	Quest appears in quest list, no errors	Pass
AD_CQ_04	Admin tries to leave any required fields blank when adding quest	1. Click "Add New Quest" 2. Select any quest type. 3. Leave required fields blank. 4. Try to save	***(Leave any required fields blank)***	Validation error shown, quest not saved	Validation error shown, quest not saved	Pass
AD_CQ_05	Admin tries to add quest with invalid rewards	1. Click "Add New Quest" 2. Enter negative or zero for points/diamonds. 3. Try to save	**Points**: 0 **Diamonds**: -5	Validation error shown, quest not saved	Validation error shown, quest not saved	Pass
AD_CQ_06	Admin removes a quest before event submission	1. Remove quest from list. 2. Submit event	Any quest in the list	Attendance-based and feedback-driven quests are not removable, other quests are removable, not included in event	Attendance-based and feedback-driven quests are not removable, other quests are removable, not included in event	Pass

AD_CQ_07	Admin edits a quest before event submission	1. Edit quest details. 2. Save changes	Change points, diamonds, description, or completion criteria (if applicable)	Attendance-based and feedback-driven quests are not editable, other quest updated in quest list	Attendance-based and feedback-driven quests are not editable, other quest updated in quest list	Pass
AD_CQ_08	Admin attempts to add duplicate Early Bird or Networking quest	1. Click “Add Quest” button. 2. Try to add existing Early Bird or Networking quest.	-	Only one allowed, cannot add duplicate	Only one allowed, cannot add duplicate	Pass

Table G.24. *Admin - Manage Quest Test Cases*

Feature Name:	**Admins/event organisers can manage quests.**					
Test Case Description:	Verify that the admin can oversee all quests for a selected event, including viewing the number and type of quests, adding new quests (with restrictions on duplicates), editing/viewing quest details (excluding attendance and feedback quests), and monitoring participant progress for each quest.					
Testing Objective:	To ensure the admin can fully manage event quests, including adding, viewing, and editing quests, and tracking participant completion, with correct UI feedback and validation.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_MQ_01	Admin views all quests for an event	1. Navigate to Quest tab for an event. 2. View overview and quest list	Event with multiple quests	All quests are listed with type, name, and correct completion count	All quests are listed with type, name, and correct completion count	Pass
AD_MQ_02	Admin adds Early Bird / Networking	1. Click "Add Quest"	**Quest Type**: “Early Bird” **Max Early Birds**:	Quest is added, appears in list	Quest is added, appears in list	Pass

	quest when not present	2. Select "Early Bird Attendance Quest" or "Networking Quest" 3. Fill required fields. 4. Click "Create Quest"	10, **Diamonds**: 100, **Points**: 100 ***// OR //*** **Quest Type**: "Networking Quest" **Connections to Make**: 10, **Diamonds**: 100, **Points**: 100			
AD_MQ_03 **(Notes: Only one of each type is allowed)**	Admin attempts to add duplicate Early Bird or Networking quest	1. Try to add duplicate Early Bird or Networking quest	-	Early bird & networking quest type is not selectable from the dropdown list	Early bird & networking quest type is not selectable from the dropdown list	Pass
AD_MQ_04	Add multiple Q&A quests	1. Click "Add Quest" 2. Select "Question and Answer (Q&A) Quest" 3. Fill required fields.	**First Q&A**: - Question: "When is the event started?"	Two Q&A quests are added and listed	Two Q&A quests are added and listed	Pass

		4. Click “Create Quest” 5. Repeat for another Q&A quest	- Answer: “7:00p.m.” **Second Q&A**: - Question: “What is the theme of the event?” - Answer: “Old school”			
AD_MQ_05 **(Notes: Attendance and Feedback quests are view-only)**	Admin edits details of a Q&A , Early Bird or Networking quest	1. Click on a Q&A , Early Bird or Networking quest. 2. Edit details. 3. Save changes	**Chosen Quest**: “Networking” **Number of Connections to Make**: from 10 to 20	Changes saved; quest updated in list	Changes saved; quest updated in list	Pass
AD_MQ_06	Admin views details of Attendance or Feedback quest	1. Click on Attendance or Feedback quest	-	Quest details shown in view-only mode; editing disabled	Quest details shown in view-only mode; editing disabled	Pass
AD_MQ_07	Admin views participant progress for a quest	1. Click on a quest. 2. View "Completed" and "In Progress" tabs	Event quest with participant data	Participant progress is displayed, including completion status, rewards claimed	Participant progress is displayed, including completion status, rewards claimed	Pass

AD_MQ_08	Admin attempts to add quest with missing / invalid data	1. Click "Add Quest" 2. Leave required fields blank or invalid. 3. Submit	Incomplete or invalid quest data field when creating quest	Validation errors shown for missing or invalid data field, quest not added	Validation errors shown for missing or invalid data field, quest not added	Pass
AD_MQ_09 **(Attendance & Feedback quests cannot be deleted)**	Admin deletes a Q&A , Early Bird or Networking quest	1. Click on a Q&A , Early Bird or Networking quest. 2. Click "Delete" 3. Confirm deletion	-	Quest is removed from list and participant progress	Quest is removed from list and participant progress	Pass

Table G.25. *Admin - Create Merchandise Test Cases*

Feature Name:	**Admins/event organisers can add new merchandise.**					
Test Case Description:	Verify that an admin can successfully create a new merchandise item using the "Create Merchandise" page, including all required fields, image uploads, and category-specific logic.					
Testing Objective:	To ensure the merchandise creation process works as intended, including form validation, image handling, category logic (Clothing/Non-Clothing), and successful data submission to the backend.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_CM_01	Admin creates merchandise with all valid fields	1. Navigate to Merchandise Listing page. 2. Tap the "Create Merch" button. 3. Upload 1-4 valid images. 4. Fill in all required fields (name, description,	**Images**: 2 valid JPEGs **Name**: "FCSIT T-Shirt" **Description**: "Official FCSIT T-shirt"	Merchandise is created, success snackbar appears, redirected to merchandise listing, new item visible	Merchandise is created, success snackbar appears, redirected to merchandise listing, new item visible	Pass

		diamonds, location, category) 5. If category is "Clothing", add at least one size. 6. Click "Submit"	**Diamonds**: 100 **Location**: "FCSIT - Student Hub" **Category**: "Clothing" **Sizes**: "M", "L"			
AD_CM_02	Admin submits form with missing required fields	1. Repeat step 1 to step 2 in AD_CM_01 2. Leave one or more required fields empty. 3. Click "Submit"	**Category**: “Clothing” **Sizes**: *(blank)*	Submit button is disabled, form displays validation errors, merchandise not created	Submit button is disabled, form displays validation errors, merchandise not created	Pass
AD_CM_03	Admin uploads more than 4 images	1. Repeat step 1 to step 2 in AD_CM_01 2. Upload 5 images	5 valid images	Error message: "Maximum 4 images allowed", only 4 images accepted	Error message: "Maximum 4 images allowed", only 4 images accepted	Pass
AD_CM_04	Admin replaces an image	1. Upload 2 images. 2. Replace the first image with another valid image	Any valid image file	Image is replaced, no errors	Image is replaced, no errors	Pass
AD_CM_05	Admin creates "non-clothing" merchandise and verifies sizes are cleared	1. Select "Clothing", add sizes. 2. Change to "non-clothing”. 3. Submit	**Category**: "Non-Clothing"	Sizes array is cleared; merchandise created without sizes	Sizes array is cleared; merchandise created without sizes	Pass

Table G.26. *Admin - Manage Merchandise Test Cases*

Feature Name:	**Admins/event organisers can manage merchandise.**					
Test Case Description:	Verify that the admin can view and edit merchandise details and manage student redemptions by toggling between uncollected and collected statuses.					
Testing Objective:	To ensure that all merchandise details can be viewed and updated, and that redemption statuses can be accurately toggled, reflecting changes in the UI and database.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_MM_01	Admin views merchandise details	1. Navigate to Merchandise Listing page. 2. Select a merchandise item (e.g., FCSIT T-Shirt merchandise item)	-	All details (name, description, images, diamonds, category, sizes, location) are displayed correctly	All details (name, description, images, diamonds, category, sizes, location) are displayed correctly	Pass
AD_MM_02	Admin edits and saves merchandise details	1. Edit fields (e.g., name, description, diamonds, category, sizes, location) 2. Click "Save Changes"	**Merch Name**: from “FCSIT T-Shirt” to "FCSIT T-Shirt 2025", **Diamonds**: 200	Save Changes button is enabled, changes are saved, success snackbar appears, updated data shown	Save Changes button is enabled, changes are saved, success snackbar appears, updated data shown	Pass
AD_MM_03	Admin uploads, replaces, and deletes merchandise images	1. Upload up to 4 images. 2. Replace an image. 3. Delete an image	4 valid images	Images are updated, cannot exceed 4, cannot delete the only one image, errors shown if violated	Images are updated, cannot exceed 4, cannot delete the only one image, errors shown if violated	Pass

AD_MM_04	Admin changes category from "Clothing" to "Non-Clothing"	1. Change category. 2. Save changes	**Category**: "Non-Clothing"	Sizes field is removed; merchandise updated	Sizes field is removed; merchandise updated	Pass
AD_MM_05	Admin manages redemption status (Uncollected → Collected)	1. Go to "Redemptions" tab. 2. In "Uncollect" tab, click "Mark as Collected" for a student	Any uncollected redemption	Status changes to "Collected", student moves to "Collected" tab, database updated	Status changes to "Collected", student moves to "Collected" tab, database updated	Pass
AD_MM_06	Admin manages redemption status (Collected → Uncollected)	1. Go to "Redemptions" tab. 2. In "Collected" tab, click "Mark as Uncollected" for a student	Any collected redemption	Status changes to "Uncollected", student moves to "Uncollect" tab, database updated	Status changes to "Uncollected", student moves to "Uncollect" tab, database updated	Pass
AD_MM_08	Admin attempts to save with invalid data	1. Clear required fields 2. Click "Save Changes"	**Name**: *(blank)*	Error messages shown, changes not saved	Error messages shown, changes not saved	Pass

Table G.27. *Admin - Manage Feedback Test Cases*

Feature Name:		**Admins/event organisers can manage feedback submitted by students.**				
Test Case Description:		Verify that the admin can view, filter, and export feedback submitted by event participants, including Likert scale ratings and written comments, and see participant details.				
Testing Objective:		To ensure the admin can manage all feedback for an event, including viewing satisfaction ratings, improvement suggestions, and exporting feedback data.				
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**

AD_MF_01	View all feedback for an event	1. Navigate to “Feedback” tab for an event. 2. View feedback list	Event with feedback	All submitted feedback is listed with participant name, profile, Likert ratings, and comments	All submitted feedback is listed with participant name, profile, Likert ratings, and comments	Pass
AD_MF_02	View Likert scale explanation	1. View "Feedback Rating Scale" section	-	Likert scale values and colour codes are displayed	Likert scale values and colour codes are displayed	Pass
AD_MF_03	Export feedback to Excel if feedback existed	1. Click "Export" button	-	Excel file is downloaded with all feedback data, including participant name, ratings, and comments	Excel file is downloaded with all feedback data, including participant name, ratings, and comments	Pass

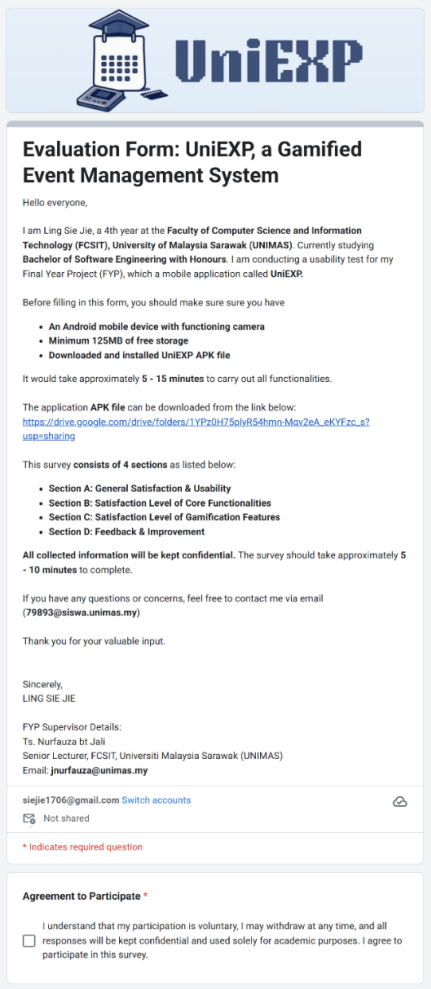
Table G.28. *Admin - View Current Month Leaderboard Test Cases*

Feature Name:	**Admins/event organisers can view current month leaderboard.**					
Test Case Description:	Verify that the admin can view the current month’s leaderboard for their faculty, including student rankings, profile info, points, year of study, and last updated time.					
Testing Objective:	To ensure the leaderboard displays accurate, up-to-date rankings for the current month, with correct student details.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_VL_01	View current month leaderboard	1. Navigate to Leaderboard page. 2. Observe the leaderboard table	Students with points for current month	Leaderboard displays ranked students, profile pictures, names, year of study, points, and last updated time	Leaderboard displays ranked students, profile pictures, names, year of study, points, and last updated time	Pass
AD_VL_02	No leaderboard data for current month	1. View leaderboard for faculty/month with no data	-	“No Leaderboard Data Yet” message shown	“No Leaderboard Data Yet” message shown	Pass

Table G.29. *Admin - View Student Participation Leaderboard Test Cases*

Feature Name:	**Admins/event organisers can view overall student participation leaderboard.**					
Test Case Description:	Verify that the admin can view a leaderboard of students ranked by the number of events attended (primary) and total points gained (secondary), including student details, year of study, and profile picture.					
Testing Objective:	To ensure the leaderboard accurately ranks students for recognition and rewards, displays all relevant student information, and handles edge cases such as ties, missing data, and no participation.					
Test Case ID	**Test Scenario**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status**
AD_VSP_01	View student participation leaderboard	1. Navigate to Student Participation page. 2. Observe the leaderboard table	Students with varying event attendance and points	Students are ranked by events attended (desc), then points gained (desc); all columns display correct info	Students are ranked by events attended (desc), then points gained (desc); all columns display correct info	Pass
AD_VSP_02	Correct ranking with ties in events attended	1. Two or more students have the same number of events attended. 2. Observe their ranking	***[Students A]*** **Events Attended**: 5 events **Points Gained**: 200 pts ***[Students B]*** **Events Attended**: 5 events **Points Gained**: 150 pts	Student with higher points is ranked higher among those with same event count	Student with higher points is ranked higher among those with same event count	Pass

Appendix H: UAT Evaluation Form



UniEXP

Evaluation Form: UniEXP, a Gamified Event Management System

Hello everyone,

I am Ling Sie Jie, a 4th year at the **Faculty of Computer Science and Information Technology (FCSIT), University of Malaysia Sarawak (UNIMAS)**. Currently studying **Bachelor of Software Engineering with Honours**. I am conducting a usability test for my Final Year Project (FYP), which a mobile application called **UniEXP**.

Before filling in this form, you should make sure sure you have

- **An Android mobile device with functioning camera**
- **Minimum 125MB of free storage**
- **Downloaded and installed UniEXP APK file**

It would take approximately **5 - 15 minutes** to carry out all functionalities.

The application **APK file** can be downloaded from the link below:
https://drive.google.com/drive/folders/1YPz0H75plyR54hmn-Mqv2eA_eKYFzc_s?usp=sharing

This survey **consists of 4 sections** as listed below:

- **Section A: General Satisfaction & Usability**
- **Section B: Satisfaction Level of Core Functionalities**
- **Section C: Satisfaction Level of Gamification Features**
- **Section D: Feedback & Improvement**

All collected information will be kept confidential. The survey should take approximately **5 - 10 minutes** to complete.

If you have any questions or concerns, feel free to contact me via email (**79893@siswa.unimas.my**)

Thank you for your valuable input.

Sincerely,
LING SIE JIE

FYP Supervisor Details:
Ts. Nurfauza bt Jali
Senior Lecturer, FCSIT, Universiti Malaysia Sarawak (UNIMAS)
Email: **jnurfauza@unimas.my**

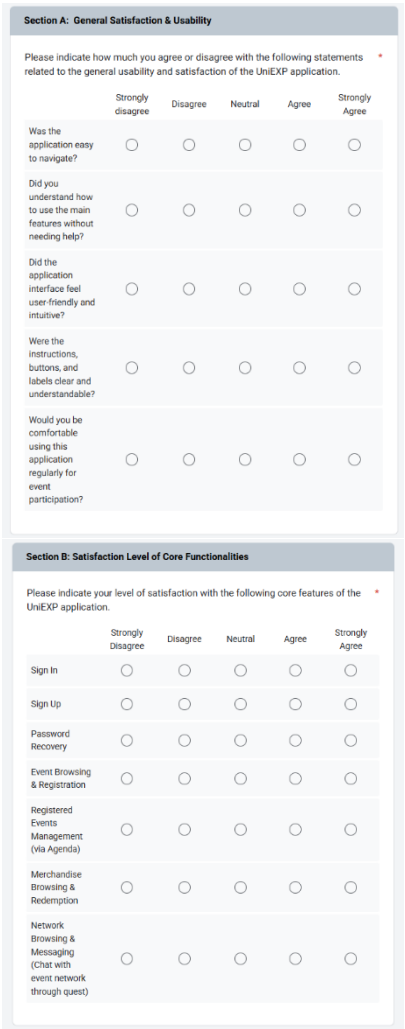
siejie1706@gmail.com Switch accounts
Not shared

* Indicates required question

Agreement to Participate *

I understand that my participation is voluntary, I may withdraw at any time, and all responses will be kept confidential and used solely for academic purposes. I agree to participate in this survey.

Figure H.1. *Student - Evaluation Form (Cover Page)*



Section A: General Satisfaction & Usability

Please indicate how much you agree or disagree with the following statements related to the general usability and satisfaction of the UniEXP application. *

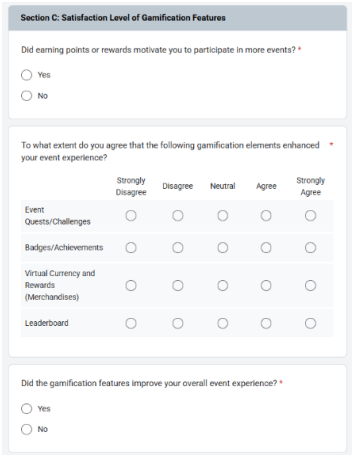
	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree
Was the application easy to navigate?	○	○	○	○	○
Did you understand how to use the main features without needing help?	○	○	○	○	○
Did the application interface feel user-friendly and intuitive?	○	○	○	○	○
Were the instructions, buttons, and labels clear and understandable?	○	○	○	○	○
Would you be comfortable using this application regularly for event participation?	○	○	○	○	○

Section B: Satisfaction Level of Core Functionalities

Please indicate your level of satisfaction with the following core features of the UniEXP application. *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Sign In	○	○	○	○	○
Sign Up	○	○	○	○	○
Password Recovery	○	○	○	○	○
Event Browsing & Registration	○	○	○	○	○
Registered Events Management (via Agenda)	○	○	○	○	○
Merchandise Browsing & Redemption	○	○	○	○	○
Network Browsing & Messaging (Chat with event network through quest)	○	○	○	○	○

Figure H.2. *Student - Evaluation Form (Section A & Section B)*



Section C: Satisfaction Level of Gamification Features

Did earning points or rewards motivate you to participate in more events? *

○ Yes
○ No

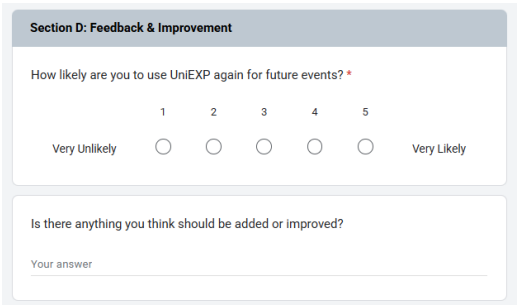
To what extent do you agree that the following gamification elements enhanced your event experience? *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Event Quests/Challenges	○	○	○	○	○
Badges/Achievements	○	○	○	○	○
Virtual Currency and Rewards (Merchandises)	○	○	○	○	○
Leaderboard	○	○	○	○	○

Did the gamification features improve your overall event experience? *

○ Yes
○ No

Figure H.3. *Student - Evaluation Form (Section C)*



Section D: Feedback & Improvement

How likely are you to use UniEXP again for future events? *

	1	2	3	4	5	
Very Unlikely	○	○	○	○	○	Very Likely

Is there anything you think should be added or improved?

Your answer

Figure H.4. *Student - Evaluation Form (Section D)*

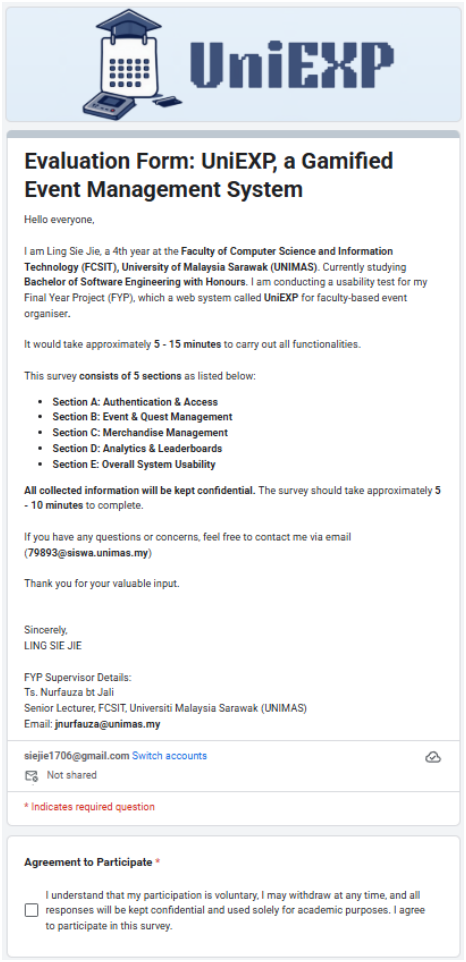


Figure H.5. *Admin - Evaluation Form (Cover Page)*

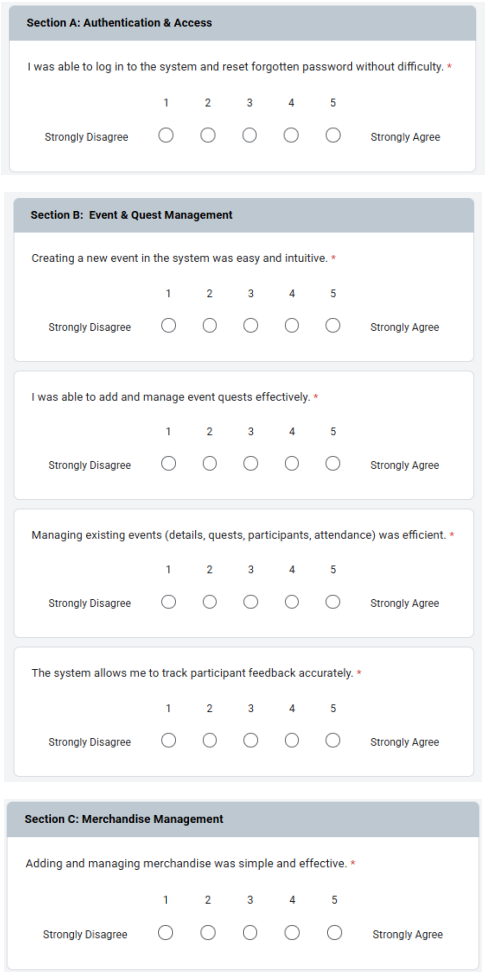


Figure H.6. *Admin - Evaluation Form (Section A, Section B & Section C)*

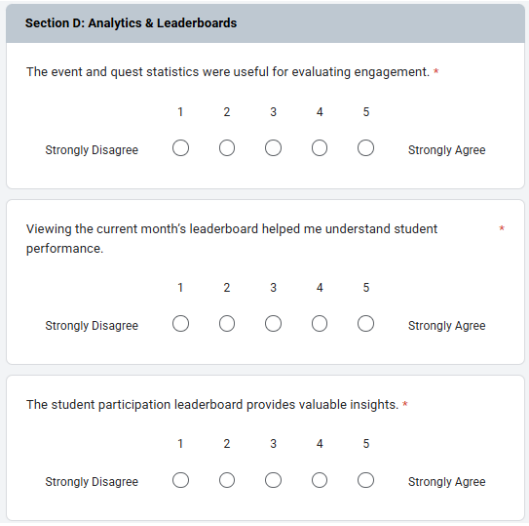


Figure H.7. *Admin - Evaluation Form (Section D)*

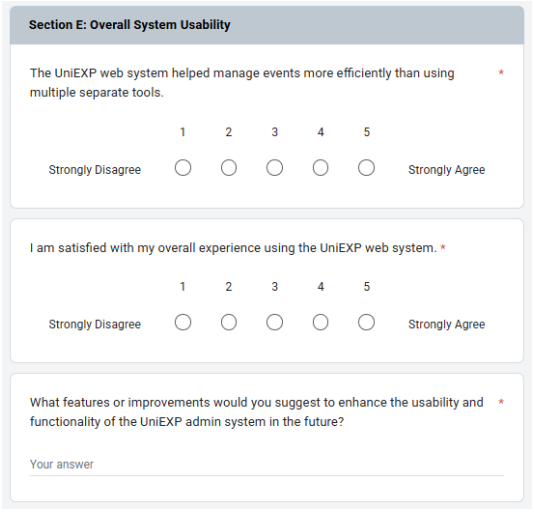


Figure H.8. *Admin - Evaluation Form (Section E)*