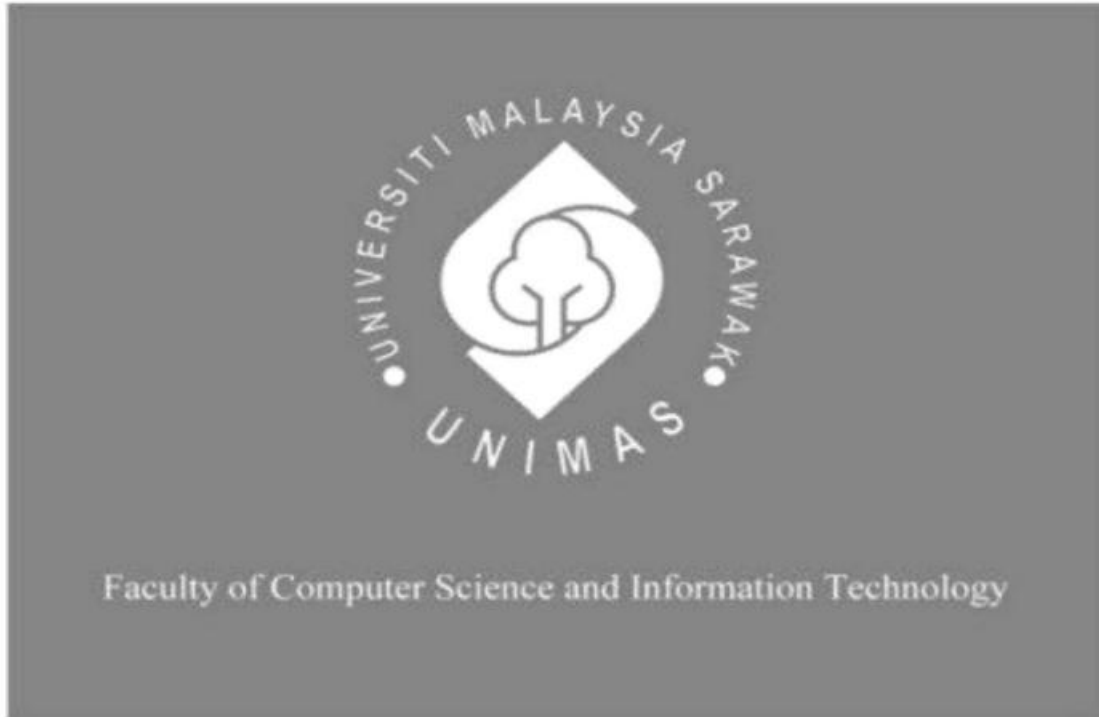




**CENTRALISED UNIVERSITY INFORMATION APPLICATION FOR MALAYSIAN
PUBLIC UNIVERSITIES (UNIFO+)**

CHIN TECK YUNG

Bachelor's degree in Computer Science with Honours (Network Computing)

2025

**CENTRALISED UNIVERSITY INFORMATION APPLICATION FOR MALAYSIAN
PUBLIC UNIVERSITIES (UNIFO+)**

CHIN TECK YUNG

This project is submitted in partial fulfilment of the requirements for the degree of Bachelor's
Degree in Computer Science with Honours (Network Computing)

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

**APLIKASI MAKLUMAT UNIVERSITI BERPUSAT UNTUK UNIVERSITI AWAM
MALAYSIA (UNIFO+)**

CHIN TECK YUNG

Projek ini dikemukakan sebagai sebahagian daripada keperluan untuk ijazah Sarjana Muda
Sains Komputer dengan Kepujian (Pengkomputeran Rangkaian).

Fakulti Sains Komputer dan Teknologi Maklumat
UNIVERSITI MALAYSIA SARAWAK

DECLARATION

I hereby declare that the project is my original work. I have not copied from any other student's work or from any other sources except due to reference or acknowledgement is not made explicitly in the text, nor has any part had been written for me by another person.



(Chin Teck Yung, 79065)

17th January 2025

Faculty of Computer Science and

Information Technology

Universiti Malaysia Sarawak

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to everyone who has contributed to the successful completion of my final year project. First and foremost, I wish to express my sincere appreciation to Madam Noor Hazlini binti Borhan for her unwavering support, motivation, and invaluable guidance throughout this journey. Her encouragement and expertise have been instrumental in navigating the challenges of this project. I am also deeply thankful to Dr. Wang Hui Hui, my examiner, and Professor Dr. Wang Yin Chai, the coordinator of the Final Year Project I and II courses. Their thorough coordination and valuable support have significantly enhanced my understanding and progress during this endeavor. Lastly, I would like to acknowledge the efforts and kindness of my lecturers, whose guidance and teaching have played a significant role in shaping my academic journey. My gratitude also extends to everyone who has supported me in every way, whether through financial assistance, words of encouragement, prayers, or simple acts of kindness. Your contributions have made this achievement possible, and I am truly thankful to everyone.

Table of Contents

DECLARATION.....	i
ACKNOWLEDGEMENT.....	ii
TABLE OF CONTENTS.....	iii
LIST OF FIGURES	v
LIST OF TABLES.....	vii
LIST OF ABBREVIATIONS	ix
ABTRACT	x
ABSTRAK	xi
CHAPTER 1: INTRODUCTION	Error! Bookmark not defined.
1.1 Introduction.....	Error! Bookmark not defined.
1.2 Problem Statement	1
1.3 Objectives	2
1.4 Scope of the Project.....	Error! Bookmark not defined.
1.5 Methodology	Error! Bookmark not defined.
1.5.1 Requirement Specification and Analysis	4
1.5.2 System Design	Error! Bookmark not defined.
1.5.3 System Implementation.....	Error! Bookmark not defined.
1.5.4 Testing and Quality Assurance	5
1.6 Significance of the Project.....	Error! Bookmark not defined.
1.7 Outline of the Project Report	Error! Bookmark not defined.
1.8 Summary.....	7
CHAPTER 2: LITERATURE REVIEW	8
2.1 Android and Features	8
2.2 iOS and Features	Error! Bookmark not defined.

2.3 Android and iOS Mobile Applications	Error! Bookmark not defined.
2.4 Project Background	Error! Bookmark not defined.
2.5 Existing applications or Web Applications Related to the Project .	Error! Bookmark not defined.
2.5.1 National University of Singapore Application (uNivUS) .	Error! Bookmark not defined.
2.5.2 United States Universities Application (Lost On Campus).....	17
2.5.3 Sunway University Application (MyCampus)	19
2.5.4 UniAPP	20
2.6 Summary.....	Error! Bookmark not defined.
CHAPTER 3: REQUIREMENTS ANALYSIS AND SYSTEM DESIGN... Error! Bookmark not defined.	
3.1 Introduction.....	Error! Bookmark not defined.
3.2 Agile Scrum Model	Error! Bookmark not defined.
3.3 Requirements Specifications	28
3.3.1 Survey	28
3.3.2 Questionnaires	28
3.3.3 Functional Requirements.....	Error! Bookmark not defined.
3.3.4 Non-Functional Requirements.....	Error! Bookmark not defined.
3.3.5 Software Development Requirements.....	Error! Bookmark not defined.
3.4 Requirements Analysis.....	Error! Bookmark not defined.
3.4.1 Use Case Diagram	Error! Bookmark not defined.
3.4.2 Product Backlog.....	Error! Bookmark not defined.
3.4.3 Sprint Backlog	52
3.4.4 Summary on Requirement Analysis	55
3.5 Design.....	56
3.5.1 Class Diagram	56

3.5.2 Software Architecture	58
3.5.3 Detailed Software Architecture	60
3.5.4 Sequence Diagram	61
3.5.5 User Interface Design	77
3.5.6 Layout Design	78
3.6 Summary.....	Error! Bookmark not defined.
CHAPTER 4 : IMPLEMENTATION AND TESTING.....	Error! Bookmark not defined.
4.1 Introduction.....	Error! Bookmark not defined.
4.2 Development Tools	Error! Bookmark not defined.
4.2.1 Programming Languages.....	Error! Bookmark not defined.
4.2.2 Visily.....	Error! Bookmark not defined.
4.2.3 Visual Studio Code	Error! Bookmark not defined.
4.2.4 Expo Go	Error! Bookmark not defined.
4.2.5 XAMPP	Error! Bookmark not defined.
4.3 Application Implementation.....	Error! Bookmark not defined.
4.3.1 Register, Login and Reset Password Screens.....	Error! Bookmark not defined.
4.3.2 Universities Activity Page.....	Error! Bookmark not defined.
4.3.3 University Details Page.....	Error! Bookmark not defined.
4.3.4 About University Page	121
4.3.5 University Facilities Page.....	Error! Bookmark not defined.
4.3.6 University Faculties Page	Error! Bookmark not defined.
4.3.7 University Course Details Page.....	Error! Bookmark not defined.
4.3.8 Dashboard Page	Error! Bookmark not defined.
4.3.9 Nearby Accommodation Page	Error! Bookmark not defined.
4.3.10 Property Details Page	Error! Bookmark not defined.

4.3.11 Add Accommodation Page	Error! Bookmark not defined.
4.3.12 Grocery Stores Page	Error! Bookmark not defined.
4.3.13 Add Grocery Store Page	Error! Bookmark not defined.
4.3.14 Nearby Eateries Page	Error! Bookmark not defined.
4.3.15 Add Eateries Page	Error! Bookmark not defined.
4.3.16 Student Activities Page	Error! Bookmark not defined.
4.3.17 Add Activity Places Page	Error! Bookmark not defined.
4.3.18 Nearby Transportation Page	Error! Bookmark not defined.
4.3.19 Add Transportation Page	Error! Bookmark not defined.
4.3.20 Open Street Map Page	Error! Bookmark not defined.
4.3.21 Event Calendar Page	Error! Bookmark not defined.
4.3.22 Events Page	Error! Bookmark not defined.
4.3.23 Personal Event Details Page	Error! Bookmark not defined.
4.3.24 Add Personal Event Page	Error! Bookmark not defined.
4.3.25 Settings Page	Error! Bookmark not defined.
4.4 Database on Server	Error! Bookmark not defined.
4.5 Summary	169
CHAPTER 5: TESTING	Error! Bookmark not defined.
5.0 Testing	Error! Bookmark not defined.
5.1 Unit Testing	Error! Bookmark not defined.
5.2 Integration Testing	Error! Bookmark not defined.
5.3 User Acceptance Testing	Error! Bookmark not defined.
5.4 Summary	180
CHAPTER 6: CONCLUSION & FUTURE WORKS	181
6.1 Introduction	181

6.2 Project Achievements	181
6.3 Project Limitations	182
6.3.1 Technical Performance Limitations	182
6.3.2 Interface Design Limitations	182
6.3.3 Language and Input Limitations	183
6.4 Future Enhancements	184
6.5 Conclusion	185
7.0 References	186
8.0 Appendix	189
8.1 Appendix A: Survey Form.....	189
8.2 Appendix B: Project Schedule	195
8.3 Appendix C: Testing.....	196
8.4 Appendix D: Usability Test Plan.....	216
8.5 Appendix E: Usability Test Questionnaire.....	217

LIST OF FIGURES

Figure 1.1: Agile Scrum Model (VectorStock, n.d.)	4
Figure 2.1: uNivUS Interface (uNivUS, n.d.)	17
Figure 2.2: Lost On Campus Interface (Lost On Campus, n.d.)	19
Figure 2.3: MyCampus Sunway Interface (MyCampus Sunway, n.d.)	21
Figure 2.4: UniAPP	22
Figure 3.1: Agile Scrum Model (VectorStock, n.d.)	29
Figure 3.2: Which of the 12 features do students think are helpful?	31
Figure 3.3: Use Case Diagram	38
Figure 3.4: Class Diagram for Unifo+	85
Figure 3.5: 3-Tier Architecture	86
Figure 3.6: Detailed Architecture	88
Figure 3.7: Sequence Diagram (Log In)	89
Figure 3.8: Sequence Diagram (Personal Event Calendar)	90
Figure 3.9: Sequence Diagram (Universities' Event Calendar)	91
Figure 3.10: Sequence Diagram (Public University List Screen)	92
Figure 3.11: Sequence Diagram (Public University List Screen - Facilities, About University, Map, Website)	93
Figure 3.12: Sequence Diagram (Public University List Screen - Courses)	94
Figure 3.13: Sequence Diagram (Dashboard Screen)	95
Figure 3.14: Sequence Diagram (Dashboard Screen - Accommodation)	96
Figure 3.15: Sequence Diagram (Dashboard Screen - Transportation)	97
Figure 3.16: Sequence Diagram (Dashboard Screen - Eateries)	98

Figure 3.17: Sequence Diagram (Dashboard Screen - Student Activity Places)	99
Figure 3.18: Sequence Diagram (Dashboard Screen - Grocery Stores)	100
Figure 3.19: Sequence Diagram (Add Option for All Dashboard Services and Event Calendar)	101
Figure 3.20: Sequence Diagram (Edit Option for All Dashboard Services and Event Calendar)	102
Figure 3.21: Sequence Diagram (Delete Option for All Dashboard Services and Event Calendar)	103
Figure 3.22: Sequence Diagram (Settings - Change Theme, Edit Profile, Logout).....	104
Figure 3.23: Navigation Flowchart	105
Figure 3.24. Sign Up Activity	107
Figure 3.25. Log In Activity	108
Figure 3.26. Forgot Password Activity.....	109
Figure 3.27. Universities Activity	110
Figure 3.28. University Details Activity.....	111
Figure 3.29. About University Activity.....	112
Figure 3.30. University Facilities Activity	113
Figure 3.31. University Faculties Activity.....	114
Figure 3.32. University Courses Activity	115
Figure 3.33. University Course Details Activity.....	116
Figure 3.34. University Dashboard Activity.....	117
Figure 3.35. Nearby Accommodation Types Activity	118
Figure 3.36. Nearby Accommodation Activity.....	119
Figure 3.37. Nearby Accommodation (Room) Activity.....	120

Figure 3.38. Add Accommodation Activity	121
Figure 3.39. Add Accommodation Activity	122
Figure 3.40. Nearby Grocery Stores Activity	123
Figure 3.41: Add Grocery Store Activity	124
Figure 3.42. Nearby Eateries Activity	125
Figure 3.43. Nearby Eateries Activity	126
Figure 3.44. Eatery Details Activity.....	127
Figure 3.45. Student Activity Places Activity.....	128
Figure 3.46. Student Activity Places Activity.....	129
Figure 3.47. Transportation Activity	130
Figure 3.48. Add Transportation Activity	131
Figure 3.49. Settings Activity	132
Figure 3.50. Event Calendar Activity.....	133
Figure 3.51. University Event Activity	134
Figure 3.52. Personal Event Activity	135
Figure 3.53. Personal Event Activity	136
Figure 4.1: Interface of Visily.....	141
Figure 4.2: Interface of Visual Studio Code	142
Figure 4.3: Expo Go Application Home Page	143
Figure 4.4: XAMPP Control Panel.....	144
Figure 4.5: Unifo+ Landing Page and Create User Account Page	146
Figure 4.6: Login Page and Reset Password Page	147
Figure 4.7: Universities Page	148

Figure 4.8: University Details Page	149
Figure 4.9: About University Page for Users.....	150
Figure 4.10: Edit About University Page for Administrators	151
Figure 4.11: University Facilities Page for Users	152
Figure 4.12: University Facilities Page for Administrators	153
Figure 4.13: University Faculties Page for Users	154
Figure 4.14: University Faculties Page for Administrators	155
Figure 4.15: University Course Details Page for Users.....	156
Figure 4.16: Edit Course Details Page for Administrators	157
Figure 4.17: University Dashboard Page.....	158
Figure 4.18: Nearby Accommodation Page with Filter Drop-down.....	159
Figure 4.19: Property Details Page for Users	160
Figure 4.20: Property Details Page for Administrators	161
Figure 4.21: Add Accommodation Page	162
Figure 4.22: Nearby Grocery Stores Page for Users	163
Figure 4.23: Nearby Grocery Stores Page for Administrators.....	164
Figure 4.24: Add Grocery Store Page	165
Figure 4.25: Nearby Eateries Page for Users.....	166
Figure 4.26: Nearby Eateries Page for Administrators.....	167
Figure 4.27: Add Eateries Page.....	168
Figure 4.28: Nearby Activity Places Page for Users.....	169
Figure 4.29: Nearby Activity Places Page for Administrators	170
Figure 4.30: Add Activity Places Page.....	171

Figure 4.31: Nearby Transportation Page for Users.....	172
Figure 4.32: Nearby Transportation Page for Administrators	173
Figure 4.33: Add Transportation Page.....	174
Figure 4.34: Open Street Map Page	175
Figure 4.35: Event Calendar Page.....	176
Figure 4.36: University Event Section in Event Page	177
Figure 4.37: Personal Event Section in Event Page	178
Figure 4.38: Personal Event Details Page.....	179
Figure 4.39: Add Personal Event Page.....	181
Figure 4.40: Settings Page for Users	182
Figure 4.41: Settings Page for Administrators.....	183
Figure 4.42: Users table.....	184
Figure 4.43: Admins table	185
Figure 4.44: About_university table.....	185
Figure 4.45: Facilities table.....	186
Figure 4.46: Faculties table.....	187
Figure 4.47: Courses table	187
Figure 4.48: Coursedetails table.....	188
Figure 4.49: Public_universities table	188
Figure 4.50: Campuses table.....	189
Figure 4.51: Propertytypes table	189
Figure 4.52: Propertyconfigurations table	190
Figure 4.53: Properties table	190

Figure 4.54: Cuisines table	191
Figure 4.55: Eateries table	191
Figure 4.56: Groceries_stores table.....	192
Figure 4.57: Activity_categories table.....	192
Figure 4.58: Activities table.....	193
Figure 4.59: Transport_types table.....	193
Figure 4.60: Transportation table	194
Figure 4.61: University_events table.....	194
Figure 5.1: Unit Testing Result	197
Figure 5.2: Integration Testing Result.....	198
Figure 5.3: Question 1 (Did the user discover any problems(s) ?)	199
Figure 5.4: Question 2 (Are the functional buttons found easily?).....	200
Figure 5.5: Question 3 (Does the user complete the task without guidance?).....	201

LIST OF FIGURES

Figure 1.1: Agile Scrum Model (VectorStock, n.d.)	4
Figure 2.1: uNivUS Interface (uNivUS, n.d.)	16
Figure 2.2: Lost On Campus Interface (Lost On Campus, n.d.)	17
Figure 2.3: MyCampus Sunway Interface (MyCampus Sunway, n.d.)	19
Figure 2.4: UniAPP	20
Figure 3.1: Agile Scrum Model (VectorStock, n.d.)	27
Figure 3.2: Which of the 12 features do students think are helpful?	29
Figure 3.3: Use Case Diagram	36
Figure 3.4: Class Diagram for Unifo+	57
Figure 3.5: 3-Tier Architecture	58
Figure 3.6: Detailed Architecture	60
Figure 3.7: Sequence Diagram (Log In)	61
Figure 3.8: Sequence Diagram (Personal Event Calendar)	62
Figure 3.9: Sequence Diagram (Universities' Event Calendar)	63
Figure 3.10: Sequence Diagram (Public University List Screen)	64
Figure 3.11: Sequence Diagram (Facilities, About University, Map, Website)	65
Figure 3.12: Sequence Diagram (Public University List Screen - Courses)	66
Figure 3.13: Sequence Diagram (Dashboard Screen)	67
Figure 3.14: Sequence Diagram (Dashboard Screen - Accommodation)	68
Figure 3.15: Sequence Diagram (Dashboard Screen - Transportation)	69
Figure 3.16: Sequence Diagram (Dashboard Screen - Eateries)	70
Figure 3.17: Sequence Diagram (Dashboard Screen - Student Activity Places)	71

Figure 3.18: Sequence Diagram (Dashboard Screen - Grocery Stores).....	72
Figure 3.19: Sequence Diagram (Add Option for All Dashboard Services and Event Calendar)	73
Figure 3.20: Sequence Diagram (Edit Option for All Dashboard Services and Event Calendar)	74
Figure 3.21: Sequence Diagram (Delete Option for All Dashboard Services and Event Calendar)	75
Figure 3.22: Sequence Diagram (Settings - Change Theme, Edit Profile, Logout).....	76
Figure 3.23: Navigation Flowchart	77
Figure 3.24. Sign Up Activity	79
Figure 3.25. Log In Activity	80
Figure 3.26. Forgot Password Activity.....	81
Figure 3.27. Universities Activity	82
Figure 3.28. University Details Activity.....	83
Figure 3.29. About University Activity	84
Figure 3.30. University Facilities Activity	85
Figure 3.31. University Faculties Activity.....	86
Figure 3.32. University Courses Activity	87
Figure 3.33. University Course Details Activity.....	88
Figure 3.34. University Dashboard Activity	89
Figure 3.35. Nearby Accommodation Types Activity	90
Figure 3.36. Nearby Accommodation Activity.....	91
Figure 3.37. Nearby Accommodation (Room) Activity.....	92
Figure 3.38. Add Accommodation Activity	93

Figure 3.39. Property Details Activity.....	94
Figure 3.40. Nearby Grocery Stores Activity	95
Figure 3.41: Add Grocery Store Activity	96
Figure 3.42. Nearby Eateries Activity	97
Figure 3.43. Add Eateries Activity.....	98
Figure 3.44. Eatery Details Activity.....	99
Figure 3.45. Student Activity Places Activity.....	100
Figure 3.46. Add Activity Places Activity.....	101
Figure 3.47. Transportation Activity	102
Figure 3.48. Add Transportation Activity	103
Figure 3.49. Settings Activity	104
Figure 3.50. Event Calendar Activity.....	105
Figure 3.51. University Event Activity	106
Figure 3.52. Personal Event Activity	107
Figure 3.53. Personal Event Activity	108
Figure 4.1: Interface of Visily	112
Figure 4.2: Interface of Visual Studio Code	113
Figure 4.3: Expo Go Application Home Page	114
Figure 4.4: XAMPP Control Panel.....	115
Figure 4.5: Unifo+ Landing Page and Create User Account Page	117
Figure 4.6: Login Page and Reset Password Page	118
Figure 4.7: Universities Page	119
Figure 4.8: University Details Page	120

Figure 4.9: About University Page for Users.....	121
Figure 4.10: Edit About University Page for Administrators	122
Figure 4.11: University Facilities Page for Users	123
Figure 4.12: University Facilities Page for Administrators	124
Figure 4.13: University Faculties Page for Users	125
Figure 4.14: University Faculties Page for Administrators	126
Figure 4.15: University Course Details Page for Users.....	127
Figure 4.16: Edit Course Details Page for Administrators	128
Figure 4.17: University Dashboard Page.....	129
Figure 4.18: Nearby Accommodation Page with Filter Drop-down	130
Figure 4.19: Property Details Page for Users	131
Figure 4.20: Property Details Page for Administrators	133
Figure 4.21: Add Accommodation Page	134
Figure 4.22: Nearby Grocery Stores Page for Users	135
Figure 4.23: Nearby Grocery Stores Page for Administrators.....	136
Figure 4.24: Add Grocery Store Page	137
Figure 4.25: Nearby Eateries Page for Users.....	138
Figure 4.26: Nearby Eateries Page for Administrators.....	139
Figure 4.27: Add Eateries Page.....	140
Figure 4.28: Nearby Activity Places Page for Users.....	141
Figure 4.29: Nearby Activity Places Page for Administrators	143
Figure 4.30: Add Activity Places Page.....	143
Figure 4.31: Nearby Transportation Page for Users.....	144

Figure 4.32: Nearby Transportation Page for Administrators	146
Figure 4.33: Add Transportation Page.....	147
Figure 4.34: Open Street Map Page	148
Figure 4.35: Event Calendar Page.....	149
Figure 4.36: University Event Section in Event Page	150
Figure 4.37: Personal Event Section in Event Page	151
Figure 4.38: Personal Event Details Page.....	153
Figure 4.39: Add Personal Event Page.....	154
Figure 4.40: Settings Page for Users	155
Figure 4.41: Settings Page for Administrators.....	156
Figure 4.42: Users table	157
Figure 4.43: Admins table	158
Figure 4.44: About_university table.....	158
Figure 4.45: Facilities table.....	Error! Bookmark not defined.
Figure 4.46: Faculties table.....	160
Figure 4.47: Courses table	160
Figure 4.48: Coursedetails table.....	161
Figure 4.49: Public_universities table	161
Figure 4.50: Campuses table	162
Figure 4.51: Propertytypes table	162
Figure 4.52: Propertyconfigurations table	163
Figure 4.53: Properties table	163
Figure 4.54: Cuisines table	164

Figure 4.55: Eateries table	165
Figure 4.56: Groceries_stores table.....	166
Figure 4.57: Activity_categories table.....	166
Figure 4.58: Activities table.....	166
Figure 4.59: Transport_types table.....	167
Figure 4.60: Transportation table.....	168
Figure 4.61: University_events table.....	168
Figure 5.1: Unit Testing Result.....	171
Figure 5.2: Integration Testing Result.....	172
Figure 5.3: Question 1 (Did the user discover any problems(s) ?)	174
Figure 5.4: Question 2 (Are the functional buttons found easily?).....	175
Figure 5.5: Question 3 (Does the user complete the task without guidance?).....	176

LIST OF TABLES

Table 2.1: Comparison between Proposed Application and Existing Applications.....	22
Table 3.1: Development software.....	33
Table 3.2: Use Case Specification for Register.....	38
Table 3.3: Use Case Specification for Login and Logout.....	38
Table 3.4: Use Case Specification for Select University (Public University List Screen)	39
Table 3.5: Use Case Specification for View Facilities.....	40
Table 3.6: Use Case Specification for View Selected Course Details.....	41
Table 3.7: Use Case Specification for View Event Calendar.....	42
Table 3.8: Use Case Specification for View Settings.....	43
Table 3.9: Use Case Specification for View Dashboard.....	44
Table 3.10: Use Case Specification for View Transportation List.....	46
Table 3.11: Use Case Specification for View Eateries List.....	46
Table 3.12: Use Case Specification for View Student Activity Places List.....	47
Table 3.13: Use Case Specification for View Grocery Stores List.....	48
Table 3.14: Use Case Specification for View Accommodation List.....	49
Table 3.15: Use Case Specification for Add Listing.....	50
Table 3.16: Use Case Specification for Edit or Delete Personal Events.....	50
Table 3.41: Sprint backlog.....	51
Table 5.1: Selected Answers from Respondents for Open-Ended Questions.....	177
Table 6.1: Project Achievements.....	181

ABTRACT

Navigating university life poses significant challenges for students in Malaysia's public universities, especially newcomers unfamiliar with campus environments and nearby amenities. Existing solutions often lack centralized support tailored to their academic and personal needs. This paper introduces Unifo+, a centralized mobile application designed specifically for students in Malaysia's public universities. Unifo+ integrates features such as interactive campus maps, detailed course directories, and real-time updates on nearby amenities. By leveraging open-source map Application Programming Interfaces (API) and real-time data, the application helps students navigate their surroundings, estimate travel times, and make informed decisions about their academic and social activities. In usability testing, Unifo+ demonstrated improved decision-making for route selection, faster access to university resources, and higher student satisfaction compared to conventional resources like printed maps or university websites. By addressing common challenges faced by university students, Unifo+ provides a seamless, inclusive, and user-friendly solution to improve both academic and personal aspects of university life.

ABSTRAK

Menavigasi kehidupan universiti merupakan cabaran besar bagi pelajar di universiti awam Malaysia, terutamanya pelajar baharu yang kurang arif dengan persekitaran kampus dan kemudahan berdekatan. Penyelesaian sedia ada sering kali tidak menyediakan sokongan berpusat yang disesuaikan dengan keperluan akademik dan peribadi pelajar. Kertas ini memperkenalkan Unifo+, sebuah aplikasi mudah alih berpusat yang direka khusus untuk pelajar di universiti awam Malaysia. Unifo+ mengintegrasikan ciri-ciri seperti peta kampus interaktif, direktori kursus yang lengkap, serta kemas kini masa nyata berkaitan kemudahan berdekatan. Dengan memanfaatkan API peta sumber terbuka dan data masa nyata, aplikasi ini membantu pelajar menavigasi kawasan sekeliling, menganggarkan masa perjalanan, serta membuat keputusan yang lebih tepat berkaitan aktiviti akademik dan sosial mereka. Dalam ujian kebolegunaan, Unifo+ menunjukkan peningkatan dalam proses membuat keputusan untuk pemilihan laluan, akses yang lebih pantas kepada sumber universiti, serta tahap kepuasan pelajar yang lebih tinggi berbanding sumber tradisional seperti peta bercetak atau laman web universiti. Dengan menangani cabaran umum yang dihadapi oleh pelajar universiti, Unifo+ menyediakan penyelesaian yang lancar, inklusif, dan mesra pengguna untuk mempertingkatkan aspek akademik dan peribadi kehidupan kampus.

CHAPTER 1: INTRODUCTION

1.1 Introduction

In Malaysia, the 20 public universities play a crucial role in shaping the nation's future workforce, catering to a diverse student population from various backgrounds. These institutions aim to provide quality education and prepare students for the workforce, but navigating the university environment can be daunting, especially for newly-admitted students. Students often face challenges familiarizing themselves with their university surroundings, accessing essential information, and managing their day-to-day activities, which can hinder their overall experience.

A key difficulty lies in locating essential services like accommodation, eateries, and academic resources, especially during the early stages of university life. Students may also struggle to access academic information such as course details, academic calendars, and campus events, which are often spread across various university websites and platforms. This scattered information leads to confusion, anxiety, and delays in important decision-making, affecting both academic performance and social integration.

Additionally, issues like slow real-time data updates, complex server management, and occasional downtime across multiple university platforms further complicate the experience. These challenges highlight the need for a centralized solution to streamline access to university resources and improve students' ability to navigate their academic and social environments.

1.2 Problem Statement

New students accepted into public universities often face significant challenges in familiarizing themselves with their new environment, including navigating university facilities,

local amenities, and essential services. Information is frequently decentralized and inconsistently distributed, making it difficult for students to efficiently access the resources they need. As De & Van (2023) point out, the difficulty in estimating travel times to campus locations, especially for students with early lectures, adds additional stress. These logistical issues, combined with the demands of adapting to a new academic environment, can lead to feelings of isolation, which may increase the risk of student dropout.

The transition from high school to university is a critical period that requires students to adjust to greater independence and responsibility. According to the British Association for Counselling and Psychotherapy (BACP), many first-year university students struggle with time management, resilience, and independence, which can contribute to increased emotional stress when transitioning to campus life. New students typically struggle to adjust to university obligations due to a lack of familiar support networks.

1.3 Objectives

- To develop an application that provides up-to-date information for all public universities in Malaysia, including official links to university websites and other resources for further exploration.
- To integrate real-time location services with open-source map APIs, enabling students to measure travel times to campus locations and surrounding areas.
- To evaluate the usability of the proposed application (Unifo+).

1.4 Scope of the Project

Unifo+ is developed to serve students enrolled in the 20 public universities across Malaysia. Whether they are first-year students adjusting to a new environment or senior students seeking specific information, Unifo+ aims to offer a personalized and inclusive experience tailored to the needs of each student. By focusing on these public universities, the application provides customized content, addressing the unique challenges students face across different institutions, thus enhancing their overall university experience.

The project includes several key features designed to make student life more manageable. Unifo+ will offer a comprehensive course directory, complete with detailed descriptions and prerequisites for each subject, allowing students to make informed decisions about their academic paths. The application will also feature real- time location- based services, helping students find nearby amenities such as restaurants, accommodation, and transportation options. Interactive campus maps will be integrated to enable effortless navigation, while quick access to official university websites will ensure students have up- to-date academic and administrative information. Additionally, real-time travel estimates for campus locations and local amenities will help students manage their time more efficiently. Together, these features will make Unifo+ a powerful tool for students throughout their university journey, enhancing both their academic and social experiences.

1.5 Methodology

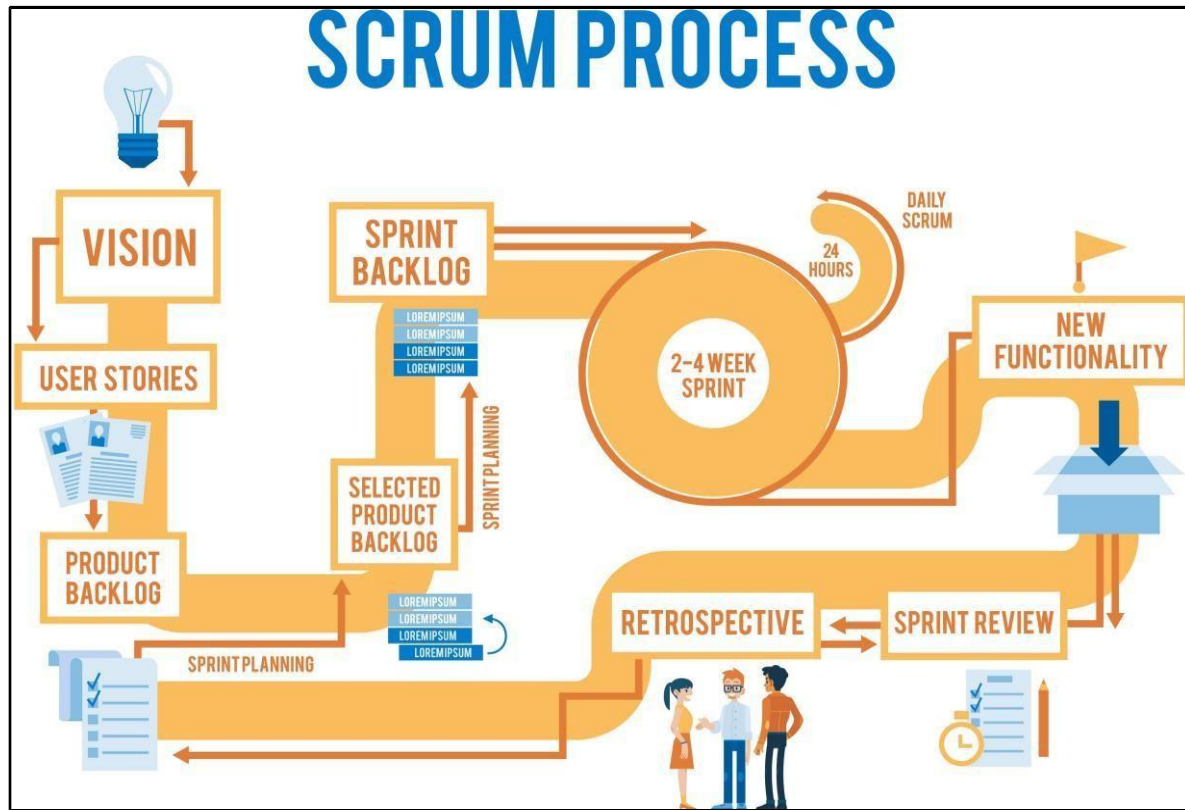


Figure 1.1: Agile Scrum Model (VectorStock, n.d.)

The development of Unifo+ follows the Agile Scrum Model as shown in Figure 1.1. The model is divided into four phases: Requirement Specification and Analysis, System Design, System Implementation, and Testing & Quality Assurance.

1.5.1 Requirement Specification and Analysis

During this phase, feedback from users, primarily students from UNIMAS, is collected through surveys and focus groups to identify key student challenges. User stories are created based on this feedback, which helps prioritize essential features such as distributed networking, data synchronization, and real-time location services.

1.5.2 System Design

In the design phase, the architecture of the system is carefully crafted to ensure scalability and reliability. A load-balancing mechanism will be implemented to distribute server requests based on geographical proximity, reducing latency and improving user experience. Real-time data synchronization across universities ensures that all information is up-to-date, while the user interface (UI) will be designed with a focus on simplicity and adaptability, supporting both light and dark mode settings.

1.5.3 System Implementation

System implementation follows the Agile methodology, with three sprints planned, each spanning four weeks. The front-end of Unifo+ will be developed using React Native with Expo Go, while the back-end will be developed using Node.js and Firebase for real-time data management. The application will focus on modular development, ensuring smooth integration of features like real-time location services using map APIs, dynamic campus maps, and live travel updates.

1.5.4 Testing and Quality Assurance

Testing will be conducted continuously throughout the project to ensure the stability, functionality, and performance of the application. Stress testing of the distributed network architecture and the accuracy of real-time data synchronization will receive special attention to ensure reliability under various usage conditions.

1.6 Significance of the Project

The Unifo+ project holds significant value for students and universities alike.

- Address the lack of centralized access to important information and resources by University students.
- Provides real-time location-based services, which helps students locate nearby amenities, which can enhance their overall well-being.
- Supports Universities acting as a direct link between students and university resources, helping institutions efficiently disseminate critical information.

1.7 Outline of the Project Report

The Centralized University Information Application for Malaysian Public Universities final report is organized into five chapters, covering topics from project initiation to completion:

Chapter 1: Introduction

This chapter provides a comprehensive overview of the Unifo+ project, outlining its background, problem statement, objectives, scope, methodology, significance, and a summary. It lays the groundwork for understanding the need for a centralized university information application.

Chapter 2: Literature Review

This chapter reviews existing research on mobile applications in higher education, with a focus on student adaptation and engagement. It compares similar applications to highlight key innovations and challenges in developing Unifo+.

Chapter 3: Requirement Analysis and Design

This chapter details the functional and non-functional requirements of the app, with user stories capturing the specific needs of Malaysian public university students. It also includes key design elements such as use case diagrams, entity-relationship models, and wire-frames.

Chapter 4: Implementation and Testing

This chapter describes the development of Unifo+ using Agile SCRUM methodology, detailing the tools and technologies implemented for both front-end and back-end development. This chapter also includes the testing strategies employed to ensure quality, including unit testing, integration testing, and user acceptance testing.

Chapter 5: Conclusion and Future Work

This chapter summarizes the key outcomes of the project, evaluating the app's success in meeting its objectives. It also outlines potential future enhancements, such as expanding the app to include more institutions and additional features, ensuring its continued relevance and growth.

1.8 Summary

Unifo+ has the ability to change the way students interact with their institutions by offering vital assistance and cultivating a sense of community. By tackling the frequent issues that new students confront, this application will help Malaysian public institutions create a more informed, connected, and engaged student body.

CHAPTER 2: LITERATURE REVIEW

This chapter reviews four current mobile applications that have similarities to the planned Unifo+ application. The applications reviewed are uNivUS, Lost Campus, MyCampus Sunway, and UniAPP. The chapter opens with a look at the background of these applications, emphasizing their academic goals and features. A full study of each of these applications is then provided, focusing on their functionality, user experience, and relevance to students' requirements. The applications' strengths and drawbacks are analyzed to acquire a better understanding of their usefulness. Following that, a comparison is made between the existing solutions and the proposed Unifo+ application, noting major similarities and areas for improvement. Finally, the chapter discusses the hardware and software requirements for developing the Unifo+ application, laying the technological groundwork for the project's development.

2.1 Android and Features

Android OS is versatile, powerful, open-source platform designed by Google (here on mobile devices, i.e., smartphones, tablets, smartwatches and wearable devices). Its origins lie in the Linux kernel, providing a strong, malleable, and extensible at base from which a broad spectrum of functions and applications are possible. Its open-source nature allows manufacturers and engineers to customize the platform in a comprehensive manner, encouraging innovation and inclusiveness for a wide variety of user requirements. The versatility of Android is not limited to traditional platforms, but it reaches forming of the Internet of Things devices and cars, hence, it becomes one of the most flexible OS platforms (BasuMallick, 2024).

The Android ecosystem is driven by Android Studio, an Integrated Development Environment (IDE) which allows the development in Kotlin and Java. This IDE offers a set of debugging, emulation, and performance analysis tools which can be used by developers to build scalable applications that can be used on a wide variety of devices. Developers are helped by the Android's support of Java programming using its Software Development Kit (SDK) which presents an emulator in order to reproduce the main device features, like navigation, network reading and multimedia features, to be tested and debugged. Such tools enable the developers to cater different audiences with different hardware configurations (Elprocus, 2019).

As functionalities of Android are multitasking, optimized graphics and integration with Google such as Gmail and Google Maps, respectively. Material Design, a design language developed by Google, ensures visually appealing, intuitive, and responsive user interfaces. Applications are mainly provided to users through the Google Play Store, covering millions of applications across a multitude of categories ranging from messaging to learning to amusement. Developers can also make applications available through third party stores or through package files (APK) thereby increasing accessibility and innovation in application distribution (Roberts, 2023).

However, these issues are accompanied by a constraint of, how do I, an open-source software and on the layered design of Android platform, which starts from the Linux kernel and goes down to user's applications are robust and adaptable enough. Over the years, Android has incorporated advanced technologies such as artificial intelligence, augmented reality, and biometric security, solidifying its position as a transformative force in the tech industry. This flexibility and scalability are what make Android a suitable platform for creative projects, such as Unifo+, that uses the operating system to provide support to both students and instructors with different backgrounds and expectations.

2.2 iOS and Features

The iPhone Operating System (iOS), the mobile operating system developed by Apple Inc., is exclusively designed for Apple devices such as iPhones, iPads, and iPods. Known for multi-touch interface, strong security, and tight integration within the Apple ecosystem, iOS is now widely used as a means for interaction within the smart home. Operating system was formally renamed as iOS in 2010 and it has been the second most widely used mobile OS all around the world (Kislay, 2024).

iOS applications are developed using Apple's Xcode, an advanced Integrated Development Environment (IDE) that supports programming languages like Swift and Objective-C. Xcode offers a comprehensive suite of tools, including a simulator, debugging utilities, and performance optimization features, enabling developers to create high-quality applications tailored specifically for Apple devices. In addition, iOS integrates the latest hardware components, such as an accelerometer, gyroscopes and GPS, to extend application capabilities and experience (Kislay, 2024).

One of the iconic features of iOS is its security focus and frequent updates. Preloaded with essential Apple applications like Safari and Apple Music, iOS ensures that user data is protected through a tightly controlled ecosystem. Major updates are released annually, keeping the platform current and secure. In addition, all iOS software is distributed and checked on iTunes only, i.e., exclusively in the Apple application Store. This closed distribution model enforces strict quality and security standards, ensuring applications are free from malware and adhere to Apple's stringent guidelines, thereby offering users a secure and reliable ecosystem (IAEME Publication, 2020).

The refined, curated iOS experience emphasizes elegance in design, minimalism, and data privacy. The interfaces remain uniform across devices, and include smooth animations, easy to use navigation, and accessibility elements, such as Voice-over and Dynamic Type. This normalization improves the user experience, as iOS applications are now relatively easy to use. In addition, the brand value and exclusivity of Apple have created brand loyal audience, even though Apple devices can be purchased for a higher price (IAEME Publication, 2020).

In contrast to its main rival, Android, iOS is a proprietary closed ecosystem (i.e., it is strongly coupled to the hardware and software by Apple). This guarantees a uniform user experience and a more optimized and secure application. But it at the same time constrains customization and device ability, by only allowing Apple-endorsed application installation pathways. Although, still, these are limitations, the simpler ecosystem is what makes iOS the most suitable platform for applications such as Unifo+ which focus on the performance, the security, and an easygoing user experience.

The set of characteristics of iOS are multitasking, i-Cloud, in-application purchases and service such as Game Center and all of them target both developers and final users. With its consistent focus on security, accessibility, and advanced functionality, iOS continues to stand out as a robust platform for delivering premium applications.

2.3 Android and iOS Mobile Applications

Mobile applications have changed the way people use technology, and how they learn in education. IOS and Android are the two prominent mobile OS that are driving the mobile phones and tablets. Although both are platforms that are designed to provide similar features and functions,

there are fundamental philosophical, technical, and developmental distinctions which make them different (Suite, 2023).

The main difference between iOS and Android is their design approach. iOS, created by Apple, runs within a closed or controlled system that includes hardware, software, and the application store, to provide a continuous and highly usable experience. This method leans to be lightweight and clear, so the interface has a beautiful, intuitive experience. By contrast, Android, designed at Google, is open-source and very extensible. The open ecosystem makes it possible for developers to develop flexible and low-cost applications that can be used on many different devices, and thus Android is best for educational applications e.g. Google Classroom. Customization also allows Android users to personalize their devices to a high degree, and but that is to be at the expense of increased fragmentation as there is a wide variety of devices with differences in screen size, hardware capabilities, and software settings (Agency, 2023).

As a matter of security iOS has been considered the more secure platform because of its rigorous application policy and closed design which minimizes opportunities for more malware and provides better quality, more reliable applications. Although Android has previously been considered insecure, the last few years have seen substantial improvements both in addressing security vulnerabilities on the platform as well as in safety of the platform as a whole (Suite, 2023).

In mobile application development, both are different development spaces. IOS application development is based on Xcode, and the programming languages are Swift and Objective-C or similar. Swift is more widely used because of its performance and ease of use, whereas the latter is still a valid option for working with legacy applications. In contrast, Android (with Android Studio) employ a collection of highly effective tools such as the

Gradle Build System, as well as are able to work with languages such as Java and Kotlin. Specifically, Kotlin provides a number of its own benefits over Java, such as increased stability and decrease in the occurrence of program crashes, which is the reason why Kotlin is the preferred language for Android programming (Agency, 2023).

Development on iOS is enhanced by a consistent platform in which device heterogeneity is reduced. This uniformity guarantees that all developers are able to easily and quickly develop applications for a variety of forms of iOS devices. By contrast, Android developers must contend with a more diffuse ecosystem, and thus they are faced with more complex development and testing process as different devices may have different screen size, hardware specifications, and OS versions (Suite, 2023).

From a user perspective, iOS applications are distributed exclusively through the Apple application Store, which enforces strict approval standards to maintain high application quality and security. Android applications, though, can be downloaded through various outlets such as the Google Play Store (as well as other third-party platforms), which provide more options but typically have poor and low quality application quality (Bigabid, 2023). Due to the flexibility of Google Play Store and the cheapness of the Android devices, educational applications have become accessible in a broader way particularly in the developing countries, where the low-cost applications play an important role to drive the educational inclusion.

Both platforms have their own advantages, and through supporting Android and iOS, Unifo+ has the ability to generate a continuous and universal experience. One provides general availability and cost effectiveness, the other provides premium performance and a highly polished user

interface. Using the advantages of each of these platform, Unifo+ can cater to a broad range of students and thus can optimize its reach in the education field.

2.4 Project Background

University mobile applications are specially designed software programs that enable students, faculty, and staff to access various services and resources directly from their smartphones or tablets. These applications serve as a one-stop platform for everything related to university life, offering features such as course information, campus navigation, and communication tools (“Benefits of University Mobile applications”, n.d.). With the advancement of technology, its impact in daily life is also increasing significantly, especially in the field of education. In today’s fast-track world, students are supposed to be well- informed about every happening around academics and co- curricular activities through easily accessible digital channels. Notifications, announcements, and events on time require an efficient medium to convey effectively to one and all. This felt need has compelled educational institutions to develop mobile applications targeting connection among students and staff.

For example, uNivUS, developed at the National University of Singapore, acts as an integrated digital portal for the students. It provides real-time information about campus navigation and access to a wide range of university services and information on various academic and social events. On the other hand, Lost On Campus has been implemented for a variety of US universities with the purpose of easing campus navigation and updating about university events and resources. MyCampus Sunway presents Sunway University with real- time notifications, facility access, and the staff directory all within the palm of a student's hands.

Also, UniAPP is an innovative platform that can enrich the experience of students' engagement and communication within universities. Designed for the management of academic information and campus resources, course management is allowed, finding accommodations, and Artificial Intelligence (AI) support. Integrating all these features, UniAPP seeks to make the student community more connected and knowledgeable so they have at their fingertips anything that is of essence.

Most of these applications, including uNivUS, Lost On Campus, MyCampus Sunway, and UniAPP, are readily available on popular platforms like the Google PlayStore and Apple application Store. This availability ensures easy access, secure downloads, and regular updates, which contribute to their efficiency in serving students and faculty.

Moreover, despite these advantages, many institutions still face the problem of ensuring notifications reach students in an effective way. The emergence of gaps in communication and inconsistencies in notification deliveries reduces the effectiveness of the use of the platforms to achieve their intended purpose in a unified, real-time, and transparent flow of information.

These applications are purposed to save time wasted searching for information, eliminate any form of complexity, and leave one platform able to have smooth access to the resources offered by the university. Effective and accurate current, real-time information enables the fuller participation of students and staff in academic and extra-curricular life

2.5 Existing applications or Web Applications Related to the Project

There are lot of education and communication applications designed with different functionalities. In this section, four different educational applications that are accessible through

both Android Google Play and Apple application Store will be reviewed where each one of them has their own benefits and drawbacks.

2.5.1 National University of Singapore Application (uNivUS)

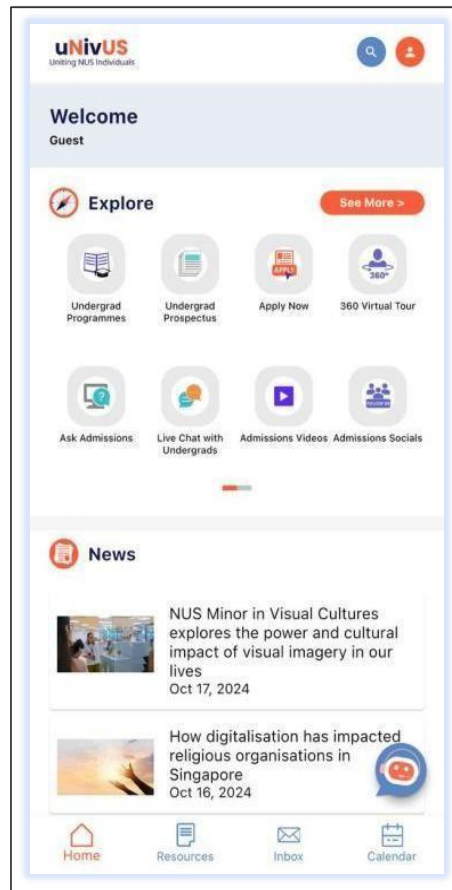


Figure 2.1: uNivUS Interface (uNivUS, n.d.)

Figure 2.1 shows uNivUS, a NUS-designed mobile application. It has served as an integrated digital touch point for students, staff, and visitors. Most of the university- wide services have been put on a single platform for easy access by its users. Besides basic functions, it provides class timetables, event calendars, and real-time announcements to notify students on campus. Among the greatest features of uNivUS is perhaps that it really integrated with the safety and

precautionary measures taken on campus. For instance, users get updated COVID-19 guidelines with functions of self- check-in, which were quite vital during the pandemic.

The application also provides many other administrative services, such as student registration, sending feedback, and viewing academic records. Its user interface is designed to be user-friendly, with personalized content and services, depending on whether one is a student, staff, or visitor. Because of this, the application has become one of the cornerstones of the NUS digital experience, easing campus life by putting the critical services into one user-friendly application.

2.5.2 United States Universities Application (Lost On Campus)

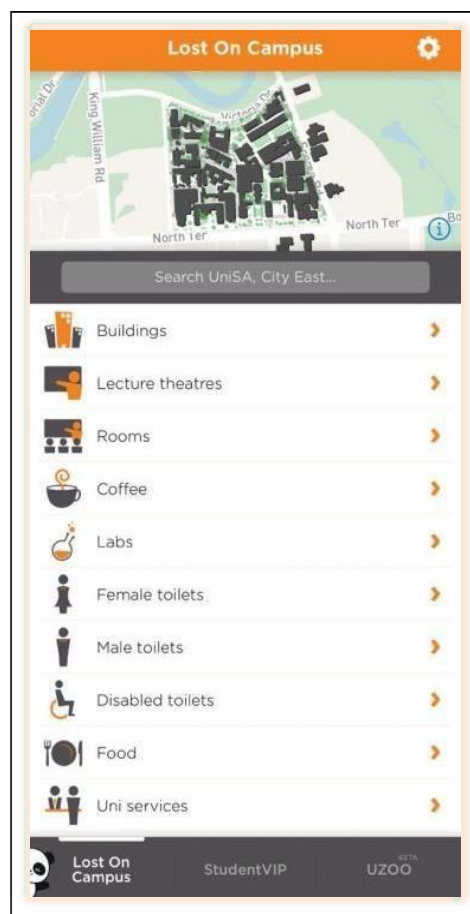


Figure 2.2: Lost On Campus Interface (Lost On Campus, n.d.)

Figure 2.2 illustrates the Lost On Campus mobile application. It is a mobile application developed for university campuses nationwide and focuses primarily on navigation and event management. Quite a helpful tool for the individuals who enter the university for the first time, or just guest visitors, because big campuses can be overwhelming for them. By employing Global Positioning System (GPS) and map integration, this application provides real-time directions to specified on-campus locations such as lecture halls, libraries, dining areas, and administration offices. It also involves event listings that can allow students to know what is happening, when club meetings take place, and even when academic workshops will be conducted.

Probably the most salient features that Lost On Campus offers are notification personalization, depending on the users' preferences and academic calendar. It keeps students updated with the latest information in events or classes that interest them. Also, it allows for the reporting of lost and found items on campus, bringing some kind of community involvement to the platform. The application offers an easy way to provide features related to accessibility on campus and allow for more significant participation in university events.

2.5.3 Sunway University Application (MyCampus)

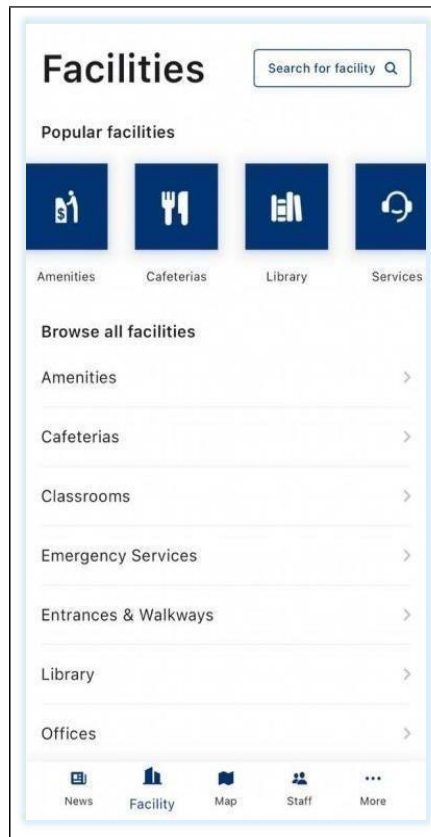


Figure 2.3: MyCampus Sunway Interface (MyCampus Sunway, n.d.)

Figure 2.3 shows the MyCampus Sunway mobile application. MyCampus Sunway is the official mobile application built for Sunway University in Malaysia, offering an unparalleled experience for students through real-time access to information and university services. The application will link various academic and administrative resources into one platform-class schedules, grades, and faculty contact details. This application will also serve campus news and event updates, besides regularly updating users about what's happening in the Sunway University community.

In addition to access to academic resources, MyCampus Sunway includes a list of Campus Navigation features, where animated maps are used to help students find facilities and buildings

with ease on campus. The application gives a staff directory through which students can get in touch with faculty members and administrative staff without much hustle. Its strong points include the ability to send push notifications for vital announcements like exam schedules or university closure. The MyCampus Sunway allows students from Sunway University to access a number of services at their fingertips, further enhancing their campus experiences.

2.5.4 UniAPP

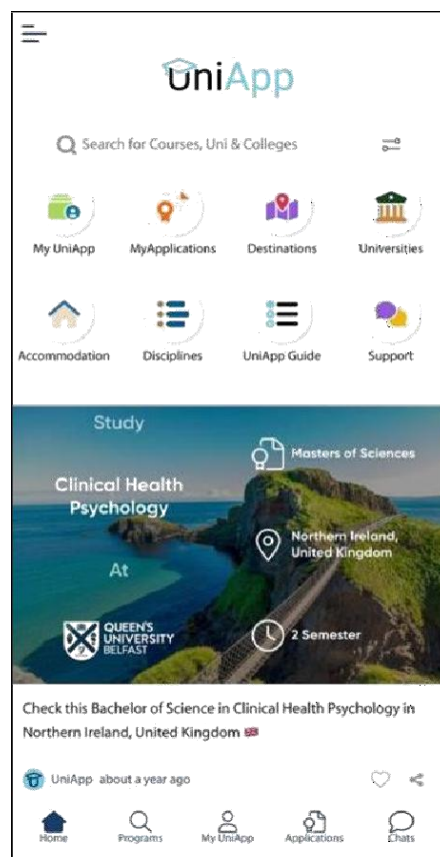


Figure 2.4: UniAPP

Figure 2.4 shows UniAPP. It is an all-inclusive mobile application designed to deliver enhanced student experiences for a number of universities and colleges in the UK, USA, Australia,

Canada, and many other nations. This multi-faceted platform will give students instant access to an enormous amount of university information and services, hence making it one of the most vital utilities for students to sail through their academic voyages comfortably. UniAPP allows the students to apply directly to courses through the application, which makes the enrollment easier. The course selection and registration feature makes it easier for students to handle their academic plan with much convenience.

Moreover, UniAPP has a specialized accommodation search feature whereby students can look for housing opportunities in the vicinity that will best suit them. This will be very helpful for freshmen who have relocated to new cities as it will give them an idea of what kind of accommodation is available, how close it is to campus, and other local amenities.

It also has an AI chat feature in UniAPP, through which students may ask any questions, from academic ones to those about campus services, and get instant answers. This interactivity makes it even easier for users, as the experience of fast access to what they need is enhanced.

UniAPP integrates a wide range of services and support in one place, hence highly enriching the campus experience for students by facilitating involvement and ease in this vibrant learning environment.

2.5.5 Comparison between Proposed Application and the Existing Applications

Table 2.1: Comparison between Proposed Application and Existing Applications

Application Name Features		National University of Singapore Application (uNivUS)	United States Universities Application (Lost On Campus)	Sunway University Application (MyCampus)	UniAPP	The proposed system (Unifo+)
Platform	Android	Yes	Yes	Yes	Yes	Yes
	IOS	Yes	Yes	Yes	Yes	Yes
Official University Links		Yes	Yes	Yes	No	Yes
Real-Time Location Services		No	Yes	No	No	Yes
Interactive Campus Maps		No	Yes	Yes	No	Yes
Access to Nearby Amenities		Yes	Yes	No	Yes	Yes
Authentication		Yes	Yes	Yes	Yes	Yes
Notifications		Yes	Yes	Yes	Yes	Yes
Campus Facility Information		Yes	No	Yes	No	Yes
Local Eateries/Accommodation		No	Yes	No	Yes	Yes
Comprehensive Course Directory		No	Yes	Yes	Yes	Yes
Real-Time Data Updates Across Universities		No	No	No	Yes	Yes

The analysis in Table 2.1 highlights the features and strengths of the proposed Unifo+ system compared to existing applications like uNivUS from the National University of Singapore, Lost on Campus from United States Universities, MyCampus from Sunway University, and Uni-application. While these applications serve their respective purposes within specific institutions or regions, Unifo+ is uniquely designed to cater to the diverse and dynamic needs of students from

all 20 public universities in Malaysia, establishing itself as a comprehensive, all-in-one solution for university-related information and services.

Like most existing applications, Unifo+ is available on both Android and iOS platforms, ensuring accessibility for a broad user base. However, unlike some applications that may prioritize functionality on a single platform or lack seamless cross-platform updates, Unifo+ is engineered for consistency and responsiveness across devices, offering a polished user experience regardless of the operating system.

One of the most significant advantages of Unifo+ is its real-time data update capability, enabling students to receive instant updates from their universities. This includes academic announcements, event notifications, and changes to class schedules, ensuring users always have the latest information at their fingertips. This feature sets Unifo+ apart from applications like MyCampus or uNivUS, which may lack this level of dynamic data integration or limit updates to static intervals.

Another key strength of Unifo+ is its all-inclusive course directory, which spans all 20 public universities in Malaysia. Unlike Lost on Campus or uNivUS, which often restrict their course directories to a single institution or require manual navigation through cumbersome interfaces, Unifo+ consolidates all relevant information into a unified and user-friendly format. This feature is invaluable for students comparing programs, selecting electives, or exploring academic opportunities.

In terms of navigation, Unifo+ incorporates dynamic campus maps with real-time location services, making navigation across university campuses intuitive and efficient. While similar features exist in applications like uNivUS, Unifo+ enhances this experience by including

information on nearby amenities such as eateries, grocery stores, accommodation options, and other facilities. These features, often absent or underdeveloped in competing applications, ensure that students have a complete ecosystem of information, not just about the campus but its surroundings as well, aiding both local and international students in adapting to their university environment.

The application's notification system is another standout feature. It is designed to keep users informed about essential updates such as academic deadlines, university events, or emergency announcements. Unlike applications like MyCampus, which may lack this feature, Unifo+ combines notifications with real-time data updates to create a proactive and engaging user experience. Additionally, its robust authentication mechanisms ensure that sensitive student information is protected. While applications like Lost on Campus do not include authentication features, and others like MyCampus may offer only basic security, Unifo+ leverages advanced encryption and secure user login to maintain data integrity and user trust.

Where most competing applications are limited in scope, focusing on one university or a specific set of services, Unifo+ aggregates information from all 20 public universities in Malaysia. This includes academic data, campus facilities, and local amenities, providing a centralized hub of resources that no other application currently offers. By doing so, Unifo+ eliminates the need for students to rely on multiple applications, streamlining their access to essential information.

Unlike other applications that focus solely on university-related services, Unifo+ extends its functionality to include details about nearby eateries, grocery stores, accommodations, and entertainment venues, providing students with a holistic experience. This is further enhanced by the application's intuitive design, which combines features like real-time location tracking and visually appealing campus maps to create a seamless and engaging user experience. By leveraging

real-time data updates, Unifo+ ensures that the information provided to students is always accurate and up-to-date, a feature unmatched by many existing platforms.

In summary, Unifo+ distinguishes itself as a one-stop solution for Malaysian public university students by combining the strengths of existing applications and introducing features that address their limitations. Its focus on real-time updates, holistic navigation tools, secure user authentication, and integration of local amenities positions it as an indispensable tool for students. By bridging the gap between academic and non-academic information, Unifo+ empowers users to thrive in their university experience, making it not just a functional application but a transformative one.

2.6 Summary

In brief, Unifo+ is developed based on the need for one fully real-time university- specific and international student engagement application to fill the gaps of existing ones, targeting Malaysian public university students. This application will integrate various necessary facilities involved such as course directories, interactive campus maps, and real- time travel estimates. While there may be a few challenges to Unifo+, this is an important step towards bringing improvement to the students' experiences and even extends to presenting an integrated solution for academic and social engagements in all 20 public universities in Malaysia.

CHAPTER 3: REQUIREMENTS ANALYSIS AND SYSTEM DESIGN

3.1 Introduction

This chapter provides a concise, yet comprehensive overview of the methodology employed in the development of the Unifo+ application. It is divided into three main sections: the system development methodology, the system analysis phase, and the system design phase. The chosen Agile SCRUM methodology, which emphasizes iterative and user-centered development, is briefly summarized. In the system analysis phase, existing solutions and relevant literature were reviewed to identify and refine the requirements for Unifo+. Additionally, this section outlines the hardware and software requirements, along with the development tools utilized throughout the project. The system design phase features several essential diagrams that illustrate the architecture and functionality of Unifo+. These include the Use Case Diagram, Flowchart, Sequence Diagrams, and Class Diagram. Furthermore, the Layout Design of the application is presented, showcasing the user interface and the navigation flow.

3.2 Agile Scrum Model

The Agile Scrum Model is a widely adopted framework within the Agile methodology, designed to enhance flexibility and collaboration in software development. Unlike linear models, Agile Scrum operates on an iterative and incremental approach, emphasizing teamwork, adaptability, and continuous delivery of functional software (Schwaber and Sutherland, 2020).

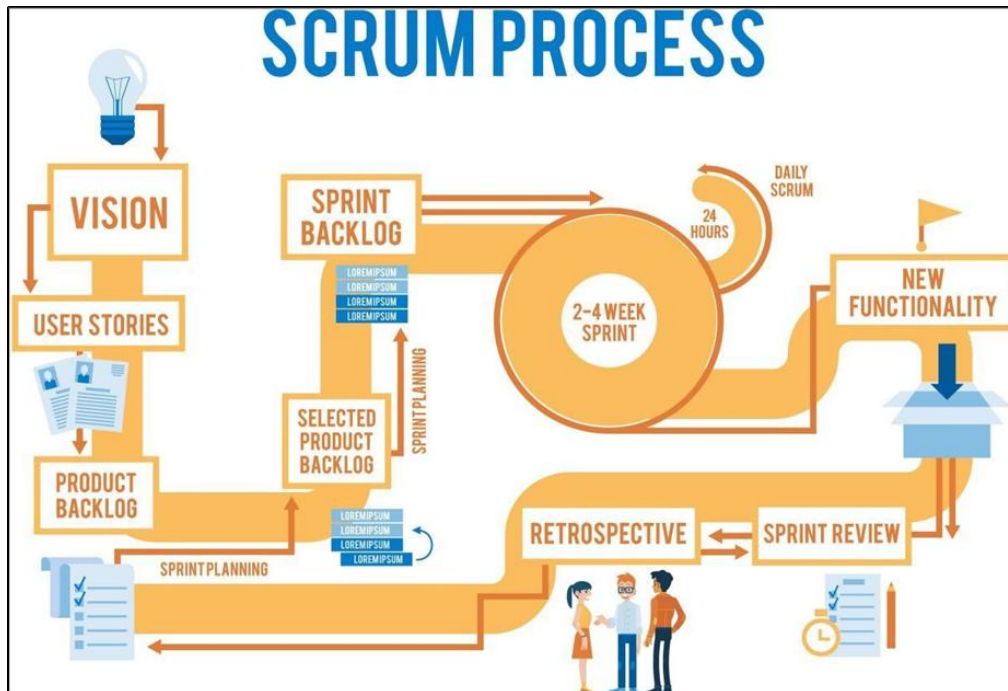


Figure 3.1: Agile Scrum Model (VectorStock, n.d.)

Figure 3.1 illustrates the Agile Scrum Model. This model consists of several key phases that are executed iteratively in time-boxed cycles known as sprints. The process begins with the Product Backlog, where all features, requirements, and enhancements are identified and prioritized. The next phase is the Sprint Planning, where the team selects tasks from the backlog to complete within the sprint. Following planning, the Sprint Execution phase involves the development, testing, and integration of features. Throughout the sprint, Daily Scrum Meetings are conducted to ensure team alignment and address any obstacles. At the end of the sprint, the team conducts a Sprint Review to showcase completed work to stakeholders and gather feedback. The Sprint Retrospective follows, where the team reflects on their performance and identifies areas for improvement. This cyclical process continues until the product is fully developed, with frequent delivery of functional increments, enabling ongoing adaptability and stakeholder collaboration throughout the development life cycle.

3.3 Requirements Specifications

In this section, requirements specification and elicitation are addressed. It consists of a description of the requirements that were extracted from the survey. Those most promising features discussed in the literature review are also extracted. This sub-section contains the description of functional, non-functional and technical requirements in three sub-sections. In addition, a software process applicable for the realization of this application is also presented in a separate sub-section.

3.3.1 Survey

To obtain the user requirement, two methods were performed. The first was the review of literature in the previous chapter and the second was the questionnaire. A comprehensive literature review was carried out from the survey of a variety of already available similar applications. Questionnaire was created and customized for the definition of requirements on the behalf of the public and the solution of this questionnaire is used to reveal the requirements. Both methods have been proved to be quite helpful for requirements elicitation and the requirements collected are classified and presented up to functional, nonfunctional, and technical.

3.3.2 Questionnaires

To capture users' experiences with university information applications, a questionnaire was developed and distributed to students from various public universities in Malaysia, gathering responses from a total of 30 students. The questions are targeted to extract the opinions and requirements from the students who are currently studying in a Malaysian public university. The questionnaire is attached in the Appendix A.

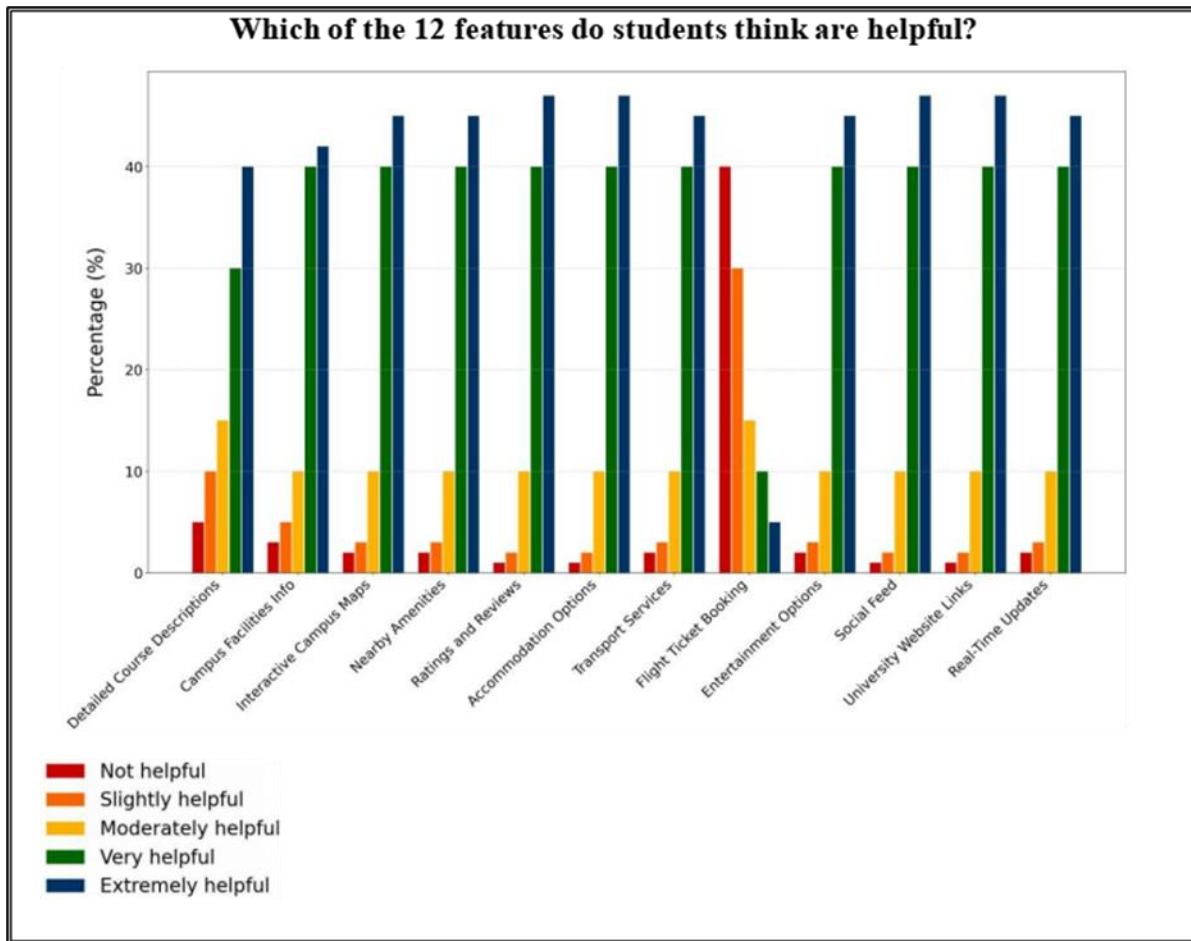


Figure 3.2: Which of the 12 features do students think are helpful?

Figure 3.2 highlights that students prioritize features offering essential university and campus-related information. Key features such as Detailed Course Descriptions, Campus Facilities Information, Interactive Campus Maps, Nearby Amenities, Accommodation Options, Transport Services, Social Feed, University Website Links, and Real-Time Updates received overwhelming approval, with 40-47% of students rating them as Extremely Useful and 30-40% as Very Useful. These features directly address students' academic, logistical, and organizational needs, underscoring their critical role in enhancing the university experience. Conversely, Flight Ticket Booking was the least valued feature, with 40% of students rating it as Not Helpful and only 5% as Extremely Useful. This indicates that travel arrangements are not a priority for university-related

applications. Other features, such as Ratings and Reviews and Entertainment Options, garnered moderate interest, reflecting a secondary level of importance. However, Nearby Amenities stood out as a useful non-academic feature, supporting students' campus lifestyles.

Among all features, Interactive Campus Maps emerged as the most impactful, offering invaluable assistance in navigating university grounds and accessing essential facilities. Its high rating as Extremely Useful solidifies its role as an indispensable tool, particularly for first-year students adapting to new environments.

In summary, the findings emphasize the need to focus on features that support students' academic and campus life while deprioritizing less relevant options like Flight Ticket Booking. By leveraging these insights, the application can be designed to deliver a truly student-centered experience.

3.3.3 Functional Requirements

This section outlines the functional requirements necessary for the system to meet its intended objectives effectively. These requirements focus on the specific functions the system must perform to fulfill user needs.

3.3.3.1 User Requirements Specification

The User Requirements Specification details the expectations and functionalities that the system must provide to its end-users. These include:

1. The user is able to Register
2. The user is able to Login and Logout
3. The user is able to Select University (Public University List Screen)
4. The user is able to View Facilities
5. The user is able to View Selected Course Details
6. The user is able to View Event Calendar
7. The user is able to View Settings
8. The user is able to View Dashboard
9. The user is able to View Transportation List
10. The user is able to View Eateries List
11. The user is able to View Student Activity Places List
12. The user is able to View Grocery Stores List
13. The user is able to View Accommodation List
14. The user is able to Add Listings
15. The user is able to Edit or Delete Added Options

3.3.3.2 Application Restrictions and Constraints

This section is specifying all the restrictions and the constraints imposed on the application. All of these rules should be consistent and maintained by the application throughout all of its operations.

- a. The application is not exposing user's personal information to public
- b. The application is not allowing users to edit or delete other users' added options in the services in the dashboard.

3.3.4 Non-Functional Requirements

This part of the report specifies the non-functional requirements that the end product should possess which include performance and usability characteristics. All of the performance characteristics are stated because the application is constantly interact with a database and the University information should be brought in a fast manner.

- a. The application presents the University information e.g. Courses and Facilities quickly after a user selects these information to view.
- b. The application presents the real time travel time estimates to each destination e.g. grocery stores and eateries based on the user's internet connection.

3.3.5 Software Development Requirements

This section specifies the software tools required to implement the system. The software requirements are shown in the table 3.1:

Table 3.1: Development software

Software/Hardware	Requirement
Operating System	Microsoft Windows 11
Hardware	Compatible with modern Android/iOS smartphones
Software	1. React Native with Expo Go for cross-platform development 2. Node.js for backend 3. MySQL and phpMyAdmin for data management 4. Google Maps API for location services
Programming Languages	1. JavaScript for both front-end and back-end (Node.js) 2. MySQL for database management
Graphic Design Tool	Visily for creating UI elements and icons

3.3.5.1 Operating System

For the development environment, Microsoft Windows 11 has been chosen for its improved performance, stability, and modern features. Windows 11 has been a stable efficient development environment providing an effortless experience with better multitasking resource allocation and with the support for newest development tools technologies.

3.3.5.2 Hardware

Unifo+ is built to be run in the majority of current smartphones that utilize either Android or iOS operating systems, thus providing wide platform and access privileges, etc. The application will be built for performance and effectively manage real time data synchronization, location, and the other features without exhausting device resources, such as memory and battery. This guarantee

that the application provides a consistent, seamless user experience on a variety of devices at the same time while remaining responsive and dependable.

3.3.5.3 Software

The software architecture of Unifo+ will be based on React Native for front-end development which will allow the development of cross-platform applications for Android and iOS using the same codebase. To streamline the development and testing process, Expo Go will be used alongside React Native, simplifying the integration of native APIs such as GPS, push notifications, and location services. Node.js will be responsible for server-side operations for scalability and high performance. Data will be collected and stored in a MySQL database, which will be controlled through phpMyAdmin, and offer databases of structured and secure data management. Integration with the Google Maps API will provide location-based services that will calculate travel times, and locate points of interest nearby to significantly improve user experience.

3.3.5.4 Programming Languages

Unifo+ will be most extensively implemented using JavaScript on both front and back-end, and will use JavaScript front-end framework React Native for cross-platform mobile, and server-side development using Node.js. For data management, MySQL will be implemented, and phpMyAdmin will be used for database administration, offering secure, scalable and efficient storage of data. Also, JavaScript will be used to provide integration with Google Maps API in order to offer location-based features, providing an unified, continuous experience throughout the application.

3.3.5.5 Graphic Design Tool

For graphic design, Visily is applied for designing user interface and icons. Visily is a user-friendly design tool that allows for rapid creation of wireframes and prototypes. It is deliberately selected for its ease of use both in the interface and in team interaction, allowing the team to easily create, tune, and visualise the user interface. Icons and visual elements designed in Visily are then brought into the Eclipse drawable package for incorporation in the graphical interface of the application, giving a professional look and feel and assuring an overall smooth and fluent experience.

3.4 Requirements Analysis

After all the requirements were gathered, the analysis phase in Agile-SCRUM is performed in this section and shown using the Unified Modeling Language. Product backlog is a part of analysis that describes what is built and translates requirements into a user story. After all user stories have been defined with their own description, acceptance criteria, priority, and story, it is assigned to the sprint backlog.

3.4.1 Use Case Diagram



Figure 3.3: Use Case Diagram

Figure 3.3 illustrates the major features included inside the Unifo+ application. These requirements are used in the product backlog as a user story.

3.4.2 Product Backlog

In this project, an Android application called Unifo+ is developed to cater to students across the 20 public universities in Malaysia. Unifo+ aims to provide a customized and inclusive experience for students, whether they are freshmen adjusting to university life or seniors looking for specific information. In order to develop this application, the Agile-Scrum methodology is applied by replacing the classic use case with user stories. User stories define features or functions of the application, while acceptance criteria describe the thresholds of the application to make sure the functionality is working as it is intended to. In terms of prioritization, the users stories are classified into high, medium, and low priority according to the importance. High-priority tasks refer to the most important tasks, which need to be started and completed first, and so-called low-priority tasks are less crucial and can be performed afterward. Story points are employed to estimate the effort of each user story. The story points are rough estimated at the start of the process, with point numbers reflecting the relative difficulty of the tasks. At the end of Sprint 1, a story points review is performed for each story points. For example, if a task with 2 story points completes faster than the task with 4 story points then the story points is estimated correctly, while if a task with 2 story points is completed slower than the task with 4 story points, the time period will be added for the task with 2 story points in the next sprint.

Use case specification:

A use case specification provides textual detail for a use case. From Table 3.2 to Table 3.40, it describes the use case specification for the proposed system.

Table 3.2: Use Case Specification for Register

Use case	Register
Short description	Students or users must register an account in order to log in and access the features of Unifo+, including viewing university data and other services.
Actor	Students/Users
Pre - condition	1. The user has not registered an account in the system. 2. The user has access to a stable internet connection.
Post - condition	1. Successful: The user successfully creates an account and is redirected to the login screen. 2. Unsuccessful: The system displays an error message and prompts the user to correct the information.
Main flow	1. The user launches the Unifo+ application. 2. The user navigates to the registration screen. 3. The user enters required registration details, including name, email, password, and confirmation of the password. 4. The system validates the entered information. 5. If the information is valid, the system creates the user account and displays a confirmation message. 6. The user is redirected to the login screen.
Alternate Flow(s)	(A1): The user enters incomplete or invalid details (e.g., mismatched passwords, missing fields). The system displays an error message: "Invalid or incomplete information. Please check your input and try again." Post-condition: The system remains on the registration screen, awaiting correct input. (A2): The user tries to register with an email that is already in use. The system displays an error message: "This email is already registered. Please use another email or log in with this account." Post-condition: The system remains on the registration screen.
Exception flow(s)	(E1): The system fails to connect to the server due to network issues. The system displays an error message: <i>"Unable to connect. Please check your internet connection and try again."</i> Post-condition: The system remains on the registration screen, waiting for reconnection.
Priority	High
Story Point	1

Table 3.3: Use Case Specification for Login and Logout

Use case	Login and Logout
Short description	Users must login in order to access their account and view university data.

Actor	Students/Users
Pre - condition	The user has already registered an account.
Post - condition	1. Successful: The user is logged in and redirected to the home screen or dashboard. 2. Unsuccessful: The system displays an error message and prompts for correction.
Main flow	1. The user launches the Unifo+ application. 2. The user navigates to the login screen. 3. The user enters their email and password. 4. The system verifies the credentials. 5. If credentials are valid, the system logs the user in and redirects to the home screen.
Alternate Flow(s)	(A1) The user enters incorrect or incomplete credentials. The system displays an error message: Invalid email or password. Please try again. Post-condition: The system remains on the login screen, awaiting correct input. (A2) The user clicks Forgot Password. The system navigates to a reset password screen. The user enters their registered email to reset their password. Post-condition: The user resets the password and logs in successfully.
Exception flow(s)	(E1) The system fails to connect to the server due to network issues. Post-condition: The system displays an error message: Unable to connect. Please check your internet connection and try again.
Priority	High
Story Point	2

Table 3.4: Use Case Specification for Select University (Public University List Screen)

Use case	Select University (Public University List Screen)
Short description	Students can select a university from a list of 20 public universities in Malaysia to access specific information and details related to that university, such as events, courses, and facilities.
Actor	Student/User
Pre - condition	1. The student is logged into the Unifo+ application. 2. The student is on the "University List" page or any page that requires selecting a university.
Post - condition	1. Successful: The system displays the selected university's details, including events, faculties, courses, location, and other relevant information. 2. Unsuccessful: The system displays an error message if the university details cannot be fetched due to a server or database issue.
Main flow	1. The student logs into the Unifo+ application. 2. The student navigates to the "University List" page or a relevant

	page where university selection is required. 3. The system displays the list of available public universities (e.g., Universiti Malaya, Universiti Kebangsaan Malaysia, etc.). 4. The student selects a university from the list. 5. The system retrieves and displays details for the selected university, including: • University map • University website • Course list • Events calendar • Facilities • Nearby amenities (e.g., eateries, accommodation) 6. The student can then choose to view more details about specific university aspects (e.g., courses, facilities, events).
Alternate Flow(s)	(A1): The student selects a university from the dropdown or search feature rather than browsing the full list. The system updates the displayed university details based on the selected option. (A2): The student decides not to proceed with selecting a university and navigates back to the previous page. Post-condition: The student returns to the previous screen without selecting a university.
Exception flow(s)	(E1): The system fails to fetch the selected university's details due to network or server issues. The system displays an error message: "Unable to retrieve university details. Please check your internet connection or try again later." (E2): The selected university has no data available. The system displays a message: "No details available for the selected university at this time."
Priority	High
Story Point	3

Table 3.5: Use Case Specification for View Facilities

Use case	View Facilities
Short description	A student accesses detailed information about university facilities by navigating through their selected university.
Actor	Student/User
Pre - condition	1. The user has logged into the application and opened the dashboard screen. 2. Students have selected the University they are interested or enrolled in from the drop-down list inside the dashboard screen.

Post - condition	The facilities images for the selected university are displayed.
Main flow	1. The student logs into the Unifo+ application. 2. The student selects a public university from the Home Screen or university selection page. 3. The student clicks on the Facilities button within the selected university's page. 4. The application displays a categorized list of facilities available at the university, such as libraries, labs, sports centers, and residential halls.
Alternate Flow(s)	(A1) If the student is not logged in, the system prompts the user to log in before accessing the facilities page. Post-condition: The student logs in and accesses the facilities page.
Exception flow(s)	(E1) If the facilities details fail to load, the system displays an error message prompting the student to retry. Post-condition: The facilities details are successfully loaded upon retry. (E2) If no facilities information is available for the university, the system displays a message: No facilities information available for this university. Post-condition: The student can navigate back to select another university.
Priority	High
Story Point	4

Table 3.6: Use Case Specification for View Selected Course Details

Use case	View Selected Course Details
Short description	Students can view detailed information about a specific course after selecting it from the list of available courses in a faculty. This includes information such as course description, credit hours, duration, prerequisites, and other relevant course details.
Actor	Student/User
Pre - condition	1. The student is logged into the Unifo+ application. 2. The student has selected a university, faculty, and course from the available lists. 3. The selected course has detailed information stored in the database.
Post - condition	1. Successful: The system displays detailed information about the selected course, including course description, credit hours, duration, prerequisites, and other relevant details. 2. Unsuccessful: The system displays an error message if the course details cannot be fetched due to a server or database issue.
Main flow	1. The student logs into the Unifo+ application. 2. The student navigates to the "University Details" page and selects a

	university. 3. The system fetches and displays the list of faculties for the selected university. 4. The student selects a faculty to view more details. 5. The system fetches and displays the list of courses offered under that faculty. 6. The student selects a course from the list to view more details. 7. The system fetches and displays the detailed information for the selected course, including: • Course description • Credit hours • Duration • Prerequisites (if applicable) • Course prospectus or syllabus (if available) • Other related course details (e.g., instructor name, schedule, etc.) 8. The student may choose to bookmark or save the course information for future reference.
Alternate flow(s)	(A1): The student decides not to view the course details and navigates away from the page. Post-condition: The student returns to the previous page without viewing the course details. (A2): The student uses the "Search" functionality to look for a course by name or other criteria. The system updates the list of displayed courses based on the search term and displays the relevant course details after the selection.
Exception flow(s)	(E1): The system fails to fetch the course details due to network or server issues. The system displays an error message: <i>"Unable to retrieve course details. Please check your internet connection or try again later."</i> (E2): The selected course does not have any details listed or is missing required information. The system displays a message: <i>"Course details are unavailable at this time. Please try again later."</i>
Priority	High
Story Point	5

Table 3.7: Use Case Specification for View Event Calendar

Use case	View Event Calendar
-----------------	---------------------

Short description	A student views and manages the event calendar for their selected university, including personal event additions and alerts.
Actor	Student/User
Pre - condition	1. The user has logged into the Unifo+ application and opened the home screen. 2. The student has selected the university they want to view events for from the drop-down list.
Post - condition	The event calendar displays upcoming university-specific events along with any personal events added by the student.
Main flow	1. The student logs into the Unifo+ application. 2. The student selects a university from the dropdown list in the home screen. 3. The system displays the event calendar with upcoming events for the selected university. 4. The student clicks on the "Add Personal Event" button to add a new event. 5. The system stores the personal event details and updates the calendar. 6. The system sends event reminders when the event date approaches. The student can edit or delete their personal events.
Alternate Flow(s)	(A1) If no university is selected, the system prompts the student to choose a university from the dropdown list. Post-condition: The student selects a university, and the event calendar is displayed for the selected university.
Exception flow(s)	(E1) If there is a network failure or server issue, the system displays an error message and asks the student to retry. Post-condition: The student can try again after a successful connection.
Priority	High
Story Point	6

Table 3.8: Use Case Specification for View Settings

Use case	View Settings
Short description	Students or users can access the settings screen to customize their app preferences, such as switching between dark and light mode, updating their personal profile, or logging out.
Actor	Students/Users
Pre - condition	1. The user is logged into the Unifo+ application. 2. The user has a stable internet connection (for updating profile information).

Post - condition	1. Successful: The user successfully views and interacts with the settings options. 2. Unsuccessful: The system displays an error message if an action fails (e.g., profile update failure due to connectivity issues).
Main flow	1. The user launches the Unifo+ application. 2. The user navigates to the settings screen via the tab bar. 3. The system displays available settings options: • Toggle between light and dark mode. • View and update personal profile information. • Log out of the application. 4. The user interacts with the desired settings option: • If the user changes the theme, the app updates the display mode instantly. • If the user updates their profile, the system saves the changes to the database. • If the user logs out, the system redirects to the login screen.
Alternate Flow(s)	(A1): The user attempts to update their profile but provides invalid or incomplete information. The system displays an error message: "Invalid details. Please check your input and try again." Post-condition: The system remains on the settings screen, awaiting correction. (A2): The user cancels the profile update before saving changes. The system discards any unsaved changes. Post-condition: The settings screen remains unchanged.
Exception flow(s)	(E1): The system fails to save the updated profile due to network issues. The system displays an error message: "Unable to save changes. Please check your internet connection and try again." Post-condition: The system retains the previous profile information.
Priority	Low
Story Point	7

Table 3.9: Use Case Specification for View Dashboard

Use case	View Dashboard
Short description	Students or users can access the dashboard screen to view and explore various services, such as nearby accommodations, eateries, grocery stores, student activity places, and transportation options based on the selected university.
Actor	Students/Users

Pre - condition	1. The user is logged into the Unifo+ application. 2. The user has a stable internet connection to fetch the latest information from the database.
Post - condition	1. Successful: • The user successfully views and interacts with the dashboard options. • The system displays the relevant information based on the selected university. 2. Unsuccessful: The system displays an error message if data retrieval fails due to connectivity issues.
Main flow	1. The user launches the Unifo+ application. 2. The user navigates to the dashboard screen via the tab bar. 3. The system displays a dropdown list for university selection. 4. The user selects a university from the dropdown list. 5. The system displays available services based on the selected university: • Nearby accommodation (Filter by property type, number of rooms, and toilets). • Nearby eateries (Filter by cuisine type). • Nearby grocery stores. • Nearby student activity places (Attractions, shopping malls, outdoor activities). • Nearby transportation options (Bus, taxi, rapid transit, car rental). 6. The user interacts with a service option to view more details. 7. The system displays the relevant details, including estimated travel time from the user's current location.
Alternate Flow(s)	(A1): The user selects a university but does not choose any specific service. The system remains on the dashboard, displaying the available service options. (A2): The user attempts to filter accommodation options but does not provide all required inputs (e.g., property type or number of rooms). The system prompts the user to complete the selection before displaying results. (A3): The user switches between universities using the dropdown list. The system updates the displayed services accordingly.
Exception flow(s)	(E1): The system fails to fetch data due to a network issue. The system displays an error message: "Unable to load data. Please check your internet connection and try again." Post-condition: The dashboard remains empty or displays previously loaded data if available.

	(E2): The selected university has no data for a particular service. The system displays a message: "No available data for this category at the selected university."
Priority	High
Story Point	8

Table 3.10: Use Case Specification for View Transportation List

Use case	View Transportation List
Short description	A student accesses information about transportation options available near the university, such as buses, taxis, or car rentals.
Actor	Student/User
Pre - condition	1. The user has logged into the application and opened the dashboard screen. 2. Students have selected the University they are interested or enrolled in from the drop-down list inside the dashboard screen.
Post - condition	The system displays the available transportation options near the student's university or selected location.
Main flow	1. The student logs into the Unifo+ application. 2. The student selects the Transportation button from the dashboard screen. 3. The system displays available transportation options near the university, including public transportation, car rentals, and taxis. 4. The student can filter the options based on transportation type or availability.
Alternate Flow(s)	(A1): If the student is not logged in, the system prompts the student to log in before accessing transportation options. Post-condition: The student logs in and can access transportation information.
Exception flow(s)	(E1): If transportation options fail to load, the system displays an error message prompting the student to retry. Post-condition: The system successfully reloads the available transportation options.
Priority	High
Story Point	9

Table 3.11: Use Case Specification for View Eateries List

Use case	View Eateries List
Short description	A student accesses information about nearby eateries, restaurants, and food vendors.
Actor	Student/User

Pre - condition	1. The user has logged into the application and opened the dashboard screen. 2. Students have selected the University they are interested or enrolled in from the drop-down list inside the dashboard screen.
Post - condition	The system displays the nearby eateries, restaurants, and food vendors based on the student's location or selected area.
Main flow	1. The student logs into the Unifo+ application. 2. The student selects the Nearby Eateries button from the dashboard screen. 3. The system displays a list of eateries and food vendors based on the University's location. 4. The student can filter or browse through the available eateries based on eatery types (asian, western, italian) to find options. 5. Students can also add new eateries into the list for the other students to refer if they have any suggestions.
Alternate flow(s)	(A1): If the student is not logged in, the system prompts the student to log in before accessing the nearby eateries list. Post-condition: The student logs in and can proceed to view nearby eateries.
Exception flow(s)	(E1): If the nearby eateries fail to load, the system displays an error message prompting the student to retry. Post-condition: The system reloads the available eateries after the retry.
Priority	High
Story Point	10

Table 3.12: Use Case Specification for View Student Activity Places List

Use case	View Student Activity Places List
Short description	A student accesses information about student activities and events available at their university.
Actor	Student/User
Pre - condition	1. The user has logged into the application and opened the dashboard screen. 2. Students have selected the University they are interested or enrolled in from the drop-down list inside the dashboard screen.
Post - condition	The system displays a list of student activity places, including shopping malls, nearby attractions and outdoor activities that are available near the selected university.
Main flow	1. The student logs into the Unifo+ application. 2. The student selects the Student Activities button from the dashboard screen. 3. The system displays a list of student activity places available near

	the university. 4. The student can browse for places of interest to enjoy and relax.
Alternate Flow(s)	(A1): If the student has not logged in, the system prompts the student to log in before accessing the student activities information. Post-condition: The student logs in and can access the student activities list.
Exception flow(s)	(E1): If student activities fail to load, the system displays an error message and prompts the student to retry. Post-condition: The system reloads the student activities list after retrying.
Priority	High
Story Point	11

Table 3.13: Use Case Specification for View Grocery Stores List

Use case	View Groceries Stores List
Short description	A student accesses information about nearby grocery stores for shopping and essentials.
Actor	Student/User
Pre - condition	1. The user has logged into the application and opened the dashboard screen. 2. Students have selected the University they are interested or enrolled in from the drop-down list inside the dashboard screen.
Post - condition	The system displays the nearby grocery stores based on the student's location or selected area.
Main flow	1. The student logs into the Unifo+ application. 2. The student selects the Nearby Groceries Stores button from the dashboard screen. 3. The system displays a list of nearby grocery stores, including supermarkets and essential vendors. 4. The student browses the stores and can filter based on location or store type.
Alternate flow(s)	(A1): If the student is not logged in, the system prompts the student to log in before accessing the nearby groceries store list. Post-condition: The student logs in, can access nearby grocery store information through the dashboard screen.
Exception flow(s)	(E1): If grocery store data fails to load due to network issues, the system prompts the student to retry. Post-condition: The system reloads the grocery store data after the retry.
Priority	High
Story Point	12

3.14: Use Case Specification for View Accommodation List

Use case	View Accommodations List
Short description	A student accesses information about nearby accommodation options, such as apartments, houses, and rooms based on their preferences.
Actor	Student/Users
Pre - condition	1. The user has logged into the application and opened the dashboard screen. 2. Students have selected the University they are interested or enrolled in from the drop-down list inside the dashboard screen.
Post - condition	The system displays the properties based on the user's selection criteria (type of property, number of rooms, and specific room preferences).
Main flow	1. The student logs into the Unifo+ application. 2. The student selects the Nearby Accommodations button from the dashboard screen. 3. The system prompts the student to select the type of property they are looking for (apartments, houses, commercial units, or rooms only). 4. The student selects the desired property type. 5. The system prompts the student to specify the number of rooms and toilets they require for the property. 6. The student selects the number of rooms and toilets needed. 7. For room selection, the student chooses from room options (master bedroom, middle room, balcony room, or single room). 8. The system displays the available properties based on the student's selections.
Alternate Flow(s)	(A1): If the student has not logged in, the system will prompt the user to log in before proceeding to the dashboard and selecting accommodations. Post-condition: The student logs in and can proceed to access nearby accommodations.
Exception flow(s)	(E1): If no properties match the student's criteria, the system displays a message: No properties available based on your search criteria. Post-condition: The student is presented with the option to modify their search criteria or try again. (E2): If the system fails to load accommodation information due to network or server issues, the system prompts the student to retry. Post-condition: The student retries the search after resolving the issue.
Priority	High
Story Point	13

Table 3.15: Use Case Specification for Add Listings

Use case	Add Listings (Transportation, Eateries, Student Activity Places, Grocery Stores, Accommodation)
Short description	A student, senior, or alumni adds local listings such as transportation, eateries, activity places, grocery stores, or accommodation options for others to refer to.
Actor	Student/User
Pre - condition	1. The user has logged into the Unifo+ application. 2. The user has accessed the dashboard screen. 3. The user has selected a university from the dropdown list. 4. The user has navigated to the relevant listing section (e.g., transportation, eateries).
Post - condition	The new listing is successfully stored in the database and displayed for others to view and reference.
Main flow	1. The user logs into the Unifo+ application. 2. The user navigates to the desired listing section (e.g., Transportation, Eateries, etc.). 3. The user selects the “Add [Listing Type]” option. 4. The system displays a form for entering relevant details. 5. The user fills in the required information and submits the form. 6. The system validates the input and stores the listing in the database.
Alternate flow(s)	(A1): If mandatory fields are left blank, the system highlights the missing fields and prompts the user to complete them. (A2): If invalid data is entered (e.g., wrong contact format, non-numeric room count), the system prompts the user to correct the information. Post-condition: The user completes or corrects the form and resubmits successfully.
Exception flow(s)	(E1): If a network failure or server error occurs, the system displays an error message (e.g., “Unable to save details. Please try again later.”). Post-condition: The user retries the submission when the issue is resolved.
Priority	Medium
Story Point	14

Table 3.16: Use Case Specification for Edit or Delete Added Options

Use case	Edit/Delete User-Added Options for Accommodation, Transportation, Grocery Stores, Eateries, and Student Activity Places
Short description	Users (students, seniors, alumni) can edit or delete the options they have added for accommodations, transportation, grocery stores, eateries, and student activity places.

Actor	Student/User
Pre - condition	1. The user has logged into the Unifo+ application. 2. The user navigates to the respective screen (e.g., Accommodation, Transportation, Grocery Stores, Eateries, or Student Activity Places). 3. The user has previously added options in the selected category.
Post - condition	1. If editing: The updated option is saved in the database and displayed in the respective list. 2. If deleting: The option is removed from the database and no longer displayed in the respective list.
Main flow	1. The user logs into the Unifo+ application. 2. The user navigates to the specific screen where they want to manage their added options (e.g., Accommodation, Transportation). 3. The system displays all user-added options, clearly labeled as editable. 4. The user selects an option they want to edit or delete. I.) If editing: a. The system opens an edit form populated with the current details. b. The user modifies the details (e.g., property type, cuisine type). c. The user confirms the changes by clicking the "Save" button. d. The system updates the option in the database. II.) If deleting: a. The user clicks the "Delete" button. b. The system prompts the user to confirm the deletion. c. The user confirms deletion. d. The system removes the option from the database.
Alternate Flow(s)	(A1): If the user tries to edit/delete an option added by another user: • The system displays a message: You can only edit or delete your own added options. • The user is redirected back to the list of options

3.4.3 Sprint Backlog

Sprint backlog is a set of stories identified during the scrum sprint. For this project, there are 3 sprints, each spanning a period of 4 weeks. W stands for the week in the sprint, i.e., meaning one weekly period for the sprint, like W1 stands for Week 1. In each story, tasks are assigned and allocated based on the schedule. Interface is being refined on Week 1 in sprint 1, because every single interface action is furnishing one clue for one of the detail functions that could be overlooked

during the design in chapter 3, as well as for the database. Interface and database for the main story of the work are finished in sprint 1 and sprint 2, and in sprint 3, a lot of time and attention is devoted to development. Testing is carried out after each code is developed. At the end of sprint 3, all components are fused and go on to be reviewed and tested. Project schedule is attached as in Appendix B.

Table 3.17: Sprint backlog

Story	Priority	Task	Sprint 1				Sprint 2				Sprint 3			
			W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
1	High	Refine the interface	✓											
		Refine the database	✓											
		Develop the code		✓										
		Testing			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	High	Refine the interface	✓											
		Refine the database	✓											
		Develop the code		✓										
		Testing			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	High	Refine the interface		✓										
		Refine the database		✓										
		Develop the code			✓									
		Testing				✓	✓	✓	✓	✓	✓	✓	✓	✓
4	High	Refine the interface		✓										
		Refine the database		✓										
		Develop the code			✓									
		Testing				✓	✓	✓	✓	✓	✓	✓	✓	✓
5	High	Refine the interface			✓									

6	High	Refine the database			✓									
		Develop the code				✓								
		Testing					✓	✓	✓	✓	✓	✓	✓	✓
8	High	Refine the interface			✓									
		Refine the database			✓									
		Develop the code				✓								
		Testing					✓	✓	✓	✓	✓	✓	✓	✓
9	High	Refine the interface				✓								
		Refine the database				✓								
		Develop the code					✓							
		Testing						✓	✓	✓	✓	✓	✓	✓
10	High	Refine the interface					✓							
		Refine the database					✓							
		Develop the code						✓						
		Testing							✓	✓	✓	✓	✓	✓
11	High	Refine the interface					✓							
		Refine the database					✓							
		Develop the code						✓						
		Testing							✓	✓	✓	✓	✓	✓
12	High	Refine the interface						✓						
		Refine the database						✓						

		Develop the code							✓					
		Testing								✓	✓	✓	✓	✓
13	High	Refine the interface						✓						
		Refine the database						✓						
		Develop the code						✓						
		Testing								✓	✓	✓	✓	✓
14	Medium	Refine the interface						✓						
		Refine the database						✓						
		Develop the code							✓					
		Testing									✓	✓	✓	✓
15	Medium	Refine the interface							✓					
		Refine the database							✓					
		Develop the code								✓				
		Testing										✓	✓	✓
7	Low	Refine the interface										✓		
		Refine the database										✓		
		Develop the code											✓	
		Testing												✓
7	Low	Refine the interface											✓	
		Refine the database											✓	
		Develop the code												✓
		Testing												✓

3.4.4 Summary on Requirement Analysis

In this chapter, analysis in the first phase of the Agile-SCRUM methodology is performed. The product backlog contains all requirements derived from user stories, and each of the user stories is given priority based on the importance and difficulty requested. Sprint backlog is scheduled by inserting tasks and sprint time period. And in this section the aim was realized of the requirement analyses and proceeded to the next stage in the methodology, namely, the design stage.

3.5 Design

In the second stage of Agile-SCRUM design; the relational database design is being described and illustrated using entity relationship diagram. Furthermore, the software architecture of the system is presented. The full class diagram is shown in order to expose the architecture inside of the application. At the end of this design stage, a minimalistic low- fidelity prototype is shown to discuss the global interface and its behavior will serve as a reference for the development stage.

3.5.1 Class Diagram

For the permanent and reliability of data, Unifo+ uses a class diagram. User information such as university information, course schedule, and accommodation information, etc., are stored on the server. As the application is limited to online access, a continuous internet connection is always needed for the user to access, modify or to work with the data. When a user registers, their information is saved on the server, and the user can access and actively use the data as long as the system has an internet connection. All actions, e.g., viewing university-relevant information, looking for new nearby restaurants and nearby grocery stores, and looking for nearby accommodation are achieved via live communication with the server. Whenever a user performs an operation or an addition, those changes are reflected in real time on the server, thus allowing for up-to-date data. As the application itself relies entirely on server-side storage, users cannot access any work without an uninterrupted internet connection, which provides the safety of all information on securely stored and synchronized cloud servers.

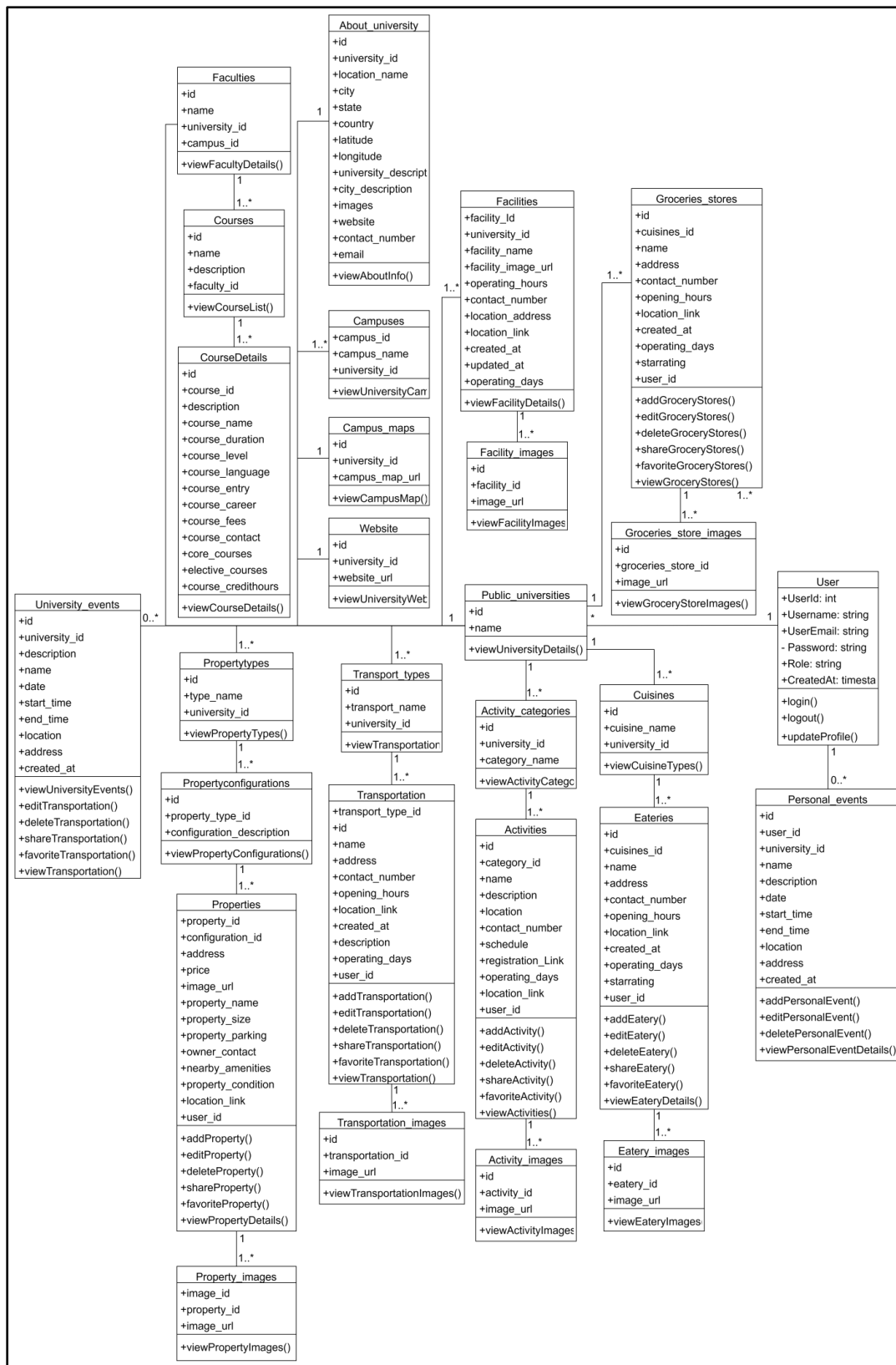


Figure 3.4: Class Diagram for Unifox+

3.5.2 Software Architecture

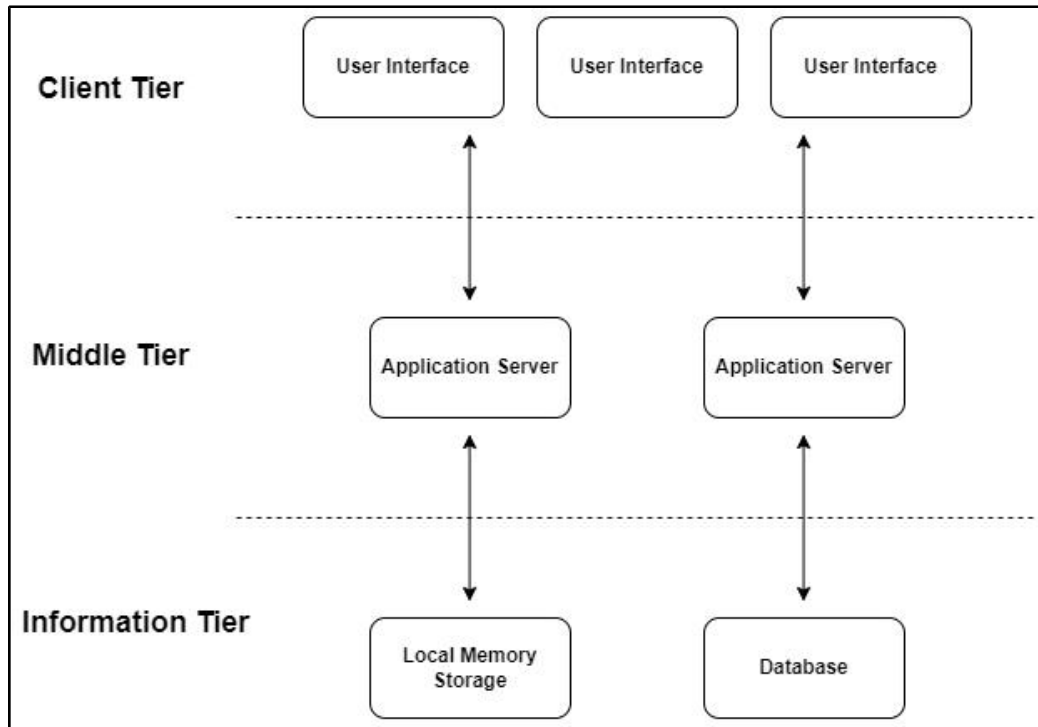


Figure 3.5: 3-Tier Architecture

Figure 3.5 shows the architecture of the 3-tier used by the Unifo+ application. The Client Tier is where users interact directly with the application's user interface, whether online or offline. This layer behaves like presentation layer and is the initial point of interaction between the user and the system. Using the intuitive interface user can get access to features related to course directories, interactive campus maps, and location-aware services. When users submit requests (eg, to deliver information about a course or locate nearby accommodations, the input is routed to the Middle Tier for processing. Upon completion of the requested process, the Client Tier displays the relevant information in a user-friendly format, ensuring that students receive the data they need in an accessible manner.

The Middle Tier works as the intermediate between the Client Tier and the Information Tier on the one hand and implements the fundamental business logic of an application on the other hand. It processes user requests by validating, transforming, and applying predefined rules to the data retrieved from the Information Tier. For example, if a user asks for semester-based courses, the Middle Tier guarantee that the data is correct, up-to- date, and well-structured before back-to-the Client Tier. This layer also includes algorithms that enforce business rules, such as ensuring only authorized users access specific university details or administrative updates. Also, it ensures data correctness, therefore improving the system reliability and consistency. The Middle Tier includes complicated exercises, masking detailed processes from the Client Tier so as to improve the user experience.

The Information Tier is in charge of securely storing and maintaining application's data. This layer provides the lifeblood of the application, which is university and campus specific course details, campus maps, location based offerings and live updates. In order to preserve confidentiality and security, the Information Tier does not directly contact the Client Tier. The whole data requests and updates flow through the Middle Tier to guarantee the security of confidential information against unauthorized access. This segregation also has allow for good data integrity and consistency, and thus allow the application to store and access large amounts of data with consistency. Basically, the 3-layered architecture of Unifo+ guarantees a scalable, secure, and performant system although it targets the varied demands of the Malaysian public university students.

3.5.3 Detailed Software Architecture

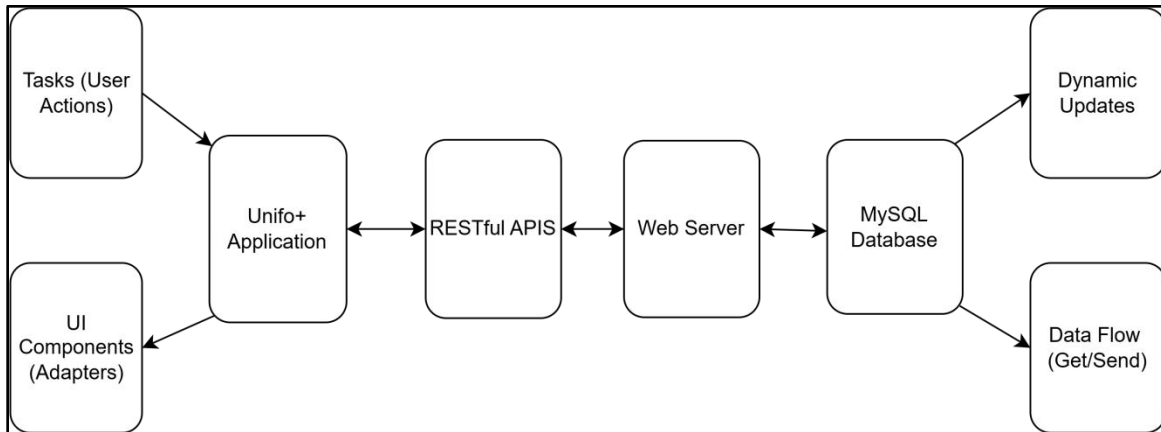


Figure 3.6: Detailed Architecture

Figure 3.6 shows the internal flow of the Unifo+ application, which is centered around online data storage, ensuring that all user and system data is managed through a centralized database. The one-way interaction is based on an application which utilizes a MySQL database hosted on a web server containing all information related to the user such as personal profiles, hotel/accommodation listings, restaurant listings, means of transport, university information and information about social engagements. This information is accessed through RESTful APIs, which allow an application to talk to the backend web server. The Restful APIs get/send data to and from the MySQL database. The data then is sent to the application's adapters, which have the task of moving the data where the right UI components to display it. Adapters are a key element in controlling the presentation of data to the application, such as for presenting university information, local conveniences, or social streams. In Unifo+, activity items are assigned to specific tasks, which are stimulated by user actions, e.g. requesting a university, or searching for nearby facilities. The user interface of the application can show information in different ways, such as list views or grid views, depending on the nature of the information being shown. This architecture can be guaranteed

a completely seamless user interface, where all data is dynamically retrieved from the web-based database to provide up to date data and a current and suitable content on the application.

3.5.4 Sequence Diagram

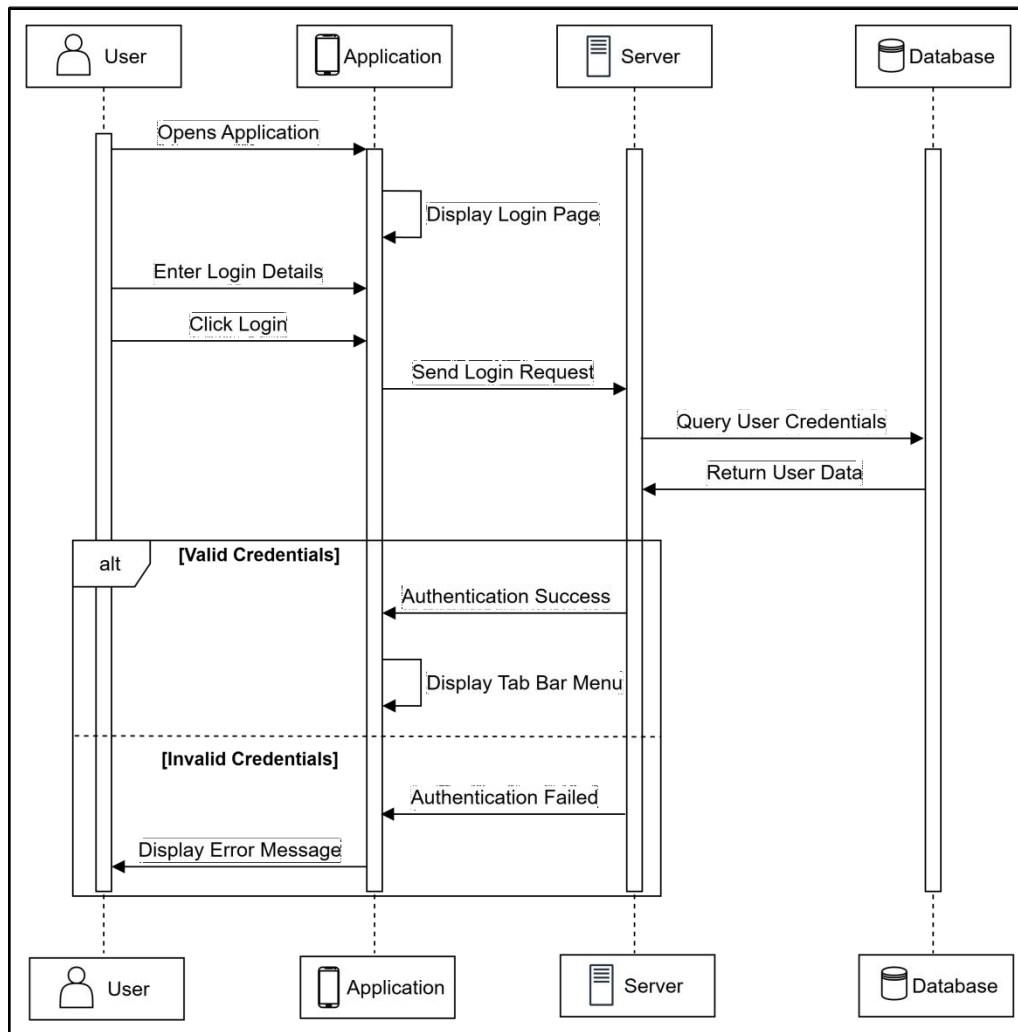


Figure 3.7: Sequence Diagram (Log In)

Figure 3.7 shows the process students follow to log in to the Unifo+ application. The system verifies the student's credentials (username and password) against the MySQL database. Upon successful authentication, the system fetches user-specific settings (e.g., theme preference, profile data) and navigates to the Home Screen with a fully personalized experience.

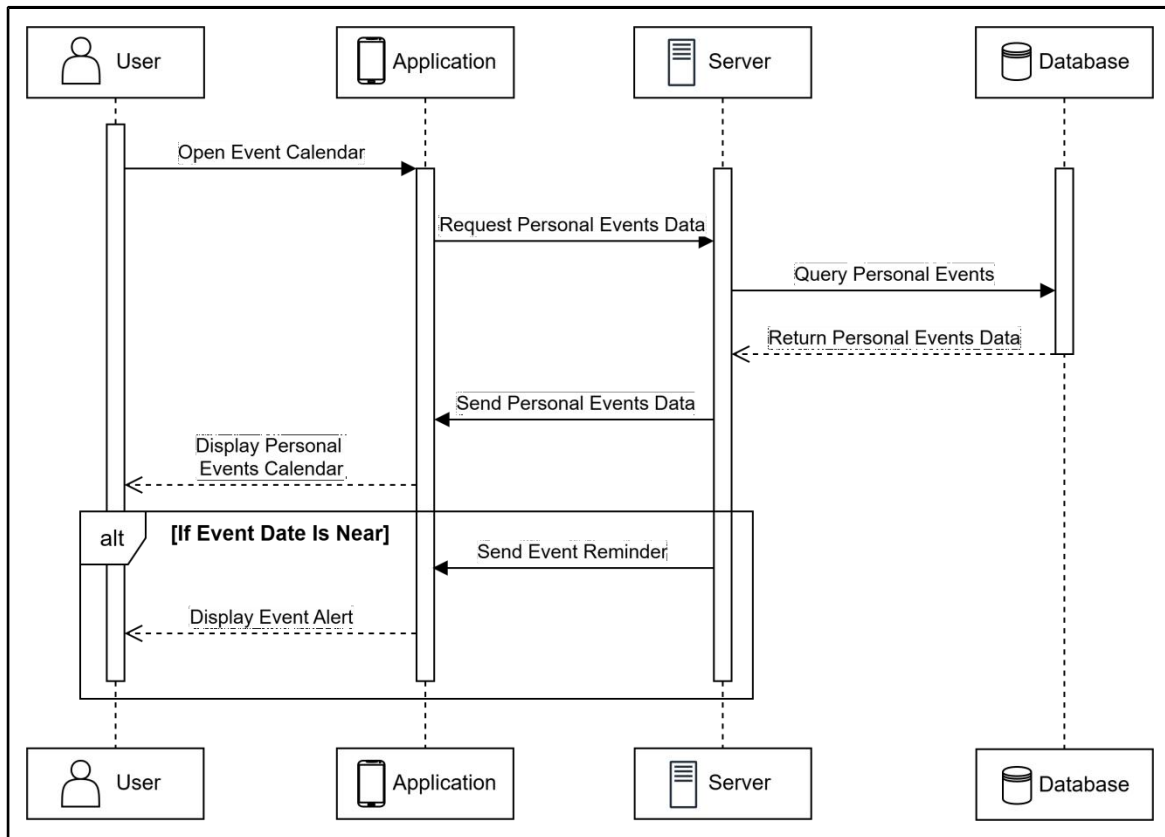


Figure 3.8: Sequence Diagram (Personal Event Calendar)

Figure 3.8 shows the sequence for managing personal events. Students add, edit, or delete personal events via the interface. The application interacts with the database to create or modify entries, ensuring all changes reflect in real-time. Alerts are also scheduled based on the event date.

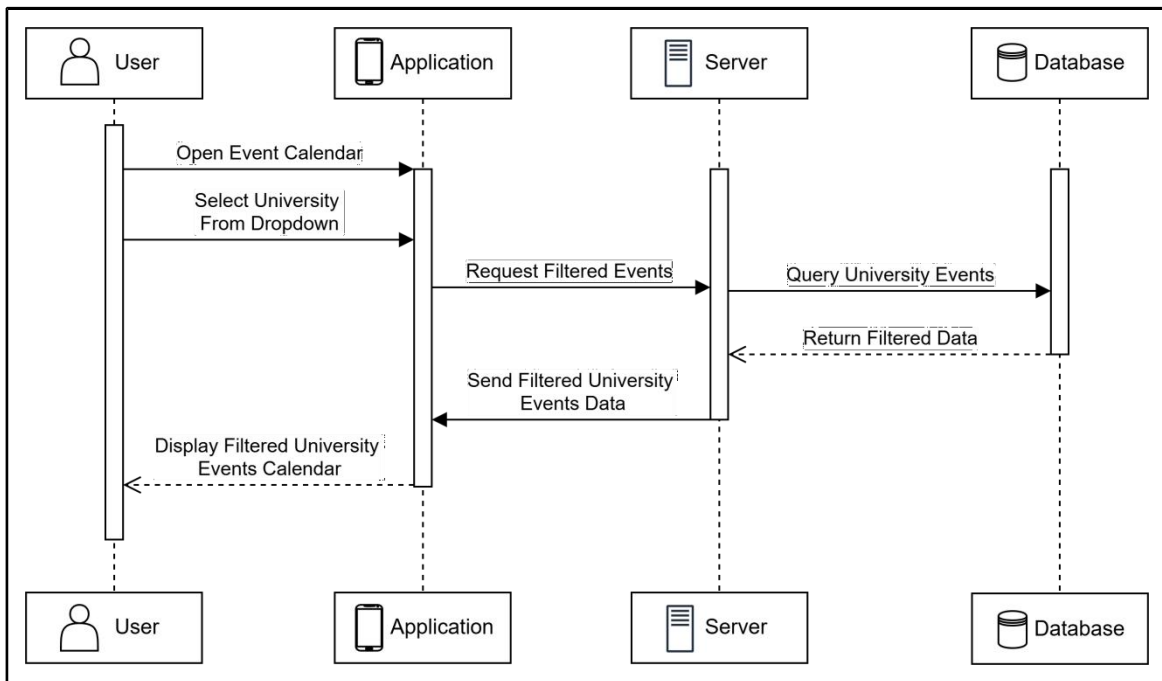


Figure 3.9: Sequence Diagram (Universities' Event Calendar)

Figure 3.9 shows how students view upcoming university-specific events. After selecting a university from the drop-down, the system queries the database for the selected university's event data and displays it. Notifications are triggered for upcoming events based on user preferences.

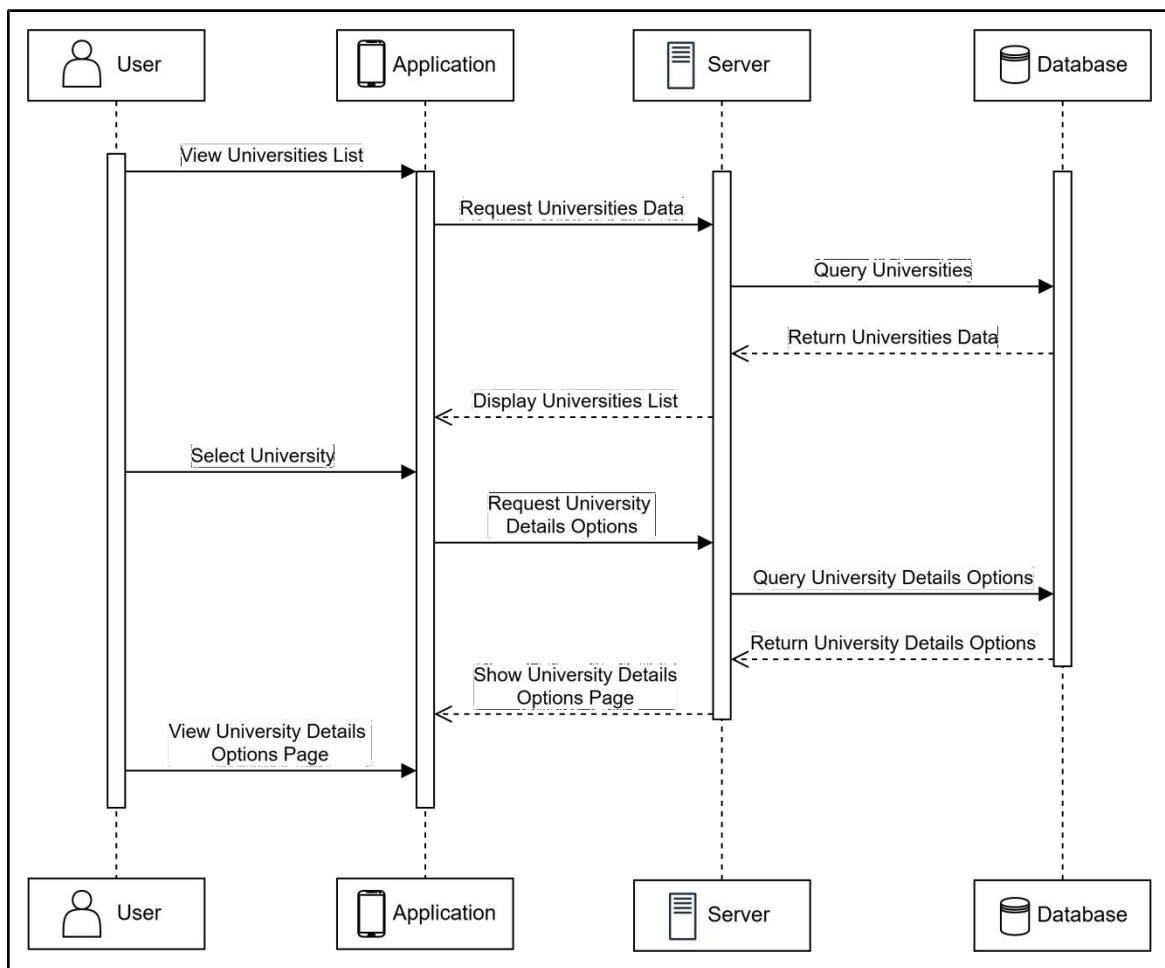


Figure 3.10: Sequence Diagram (Public University List Screen)

Figure 3.10 shows how students browse the list of 20 public universities. The application retrieves university names and basic information from the database and displays them. Selecting a university navigates the user to a detailed information page.

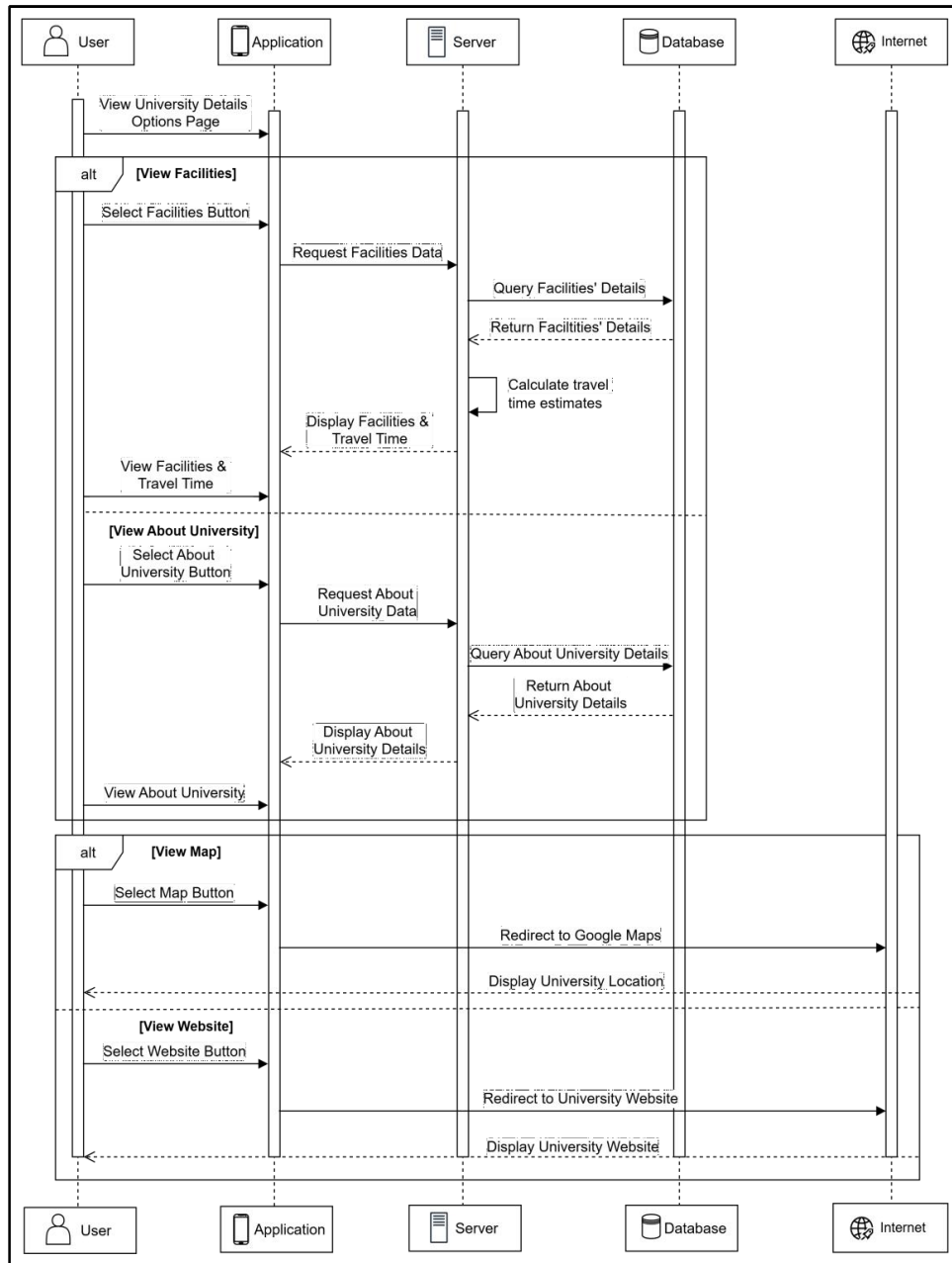


Figure 3.11: Sequence Diagram (Facilities, About University, Map, Website)

Figure 3.11 showcases the navigation process for viewing university facilities, map, or website. When a student selects one of these options, the application either displays stored facility details or redirects the user to external links (Google Maps or the university's official website) for maps and websites.

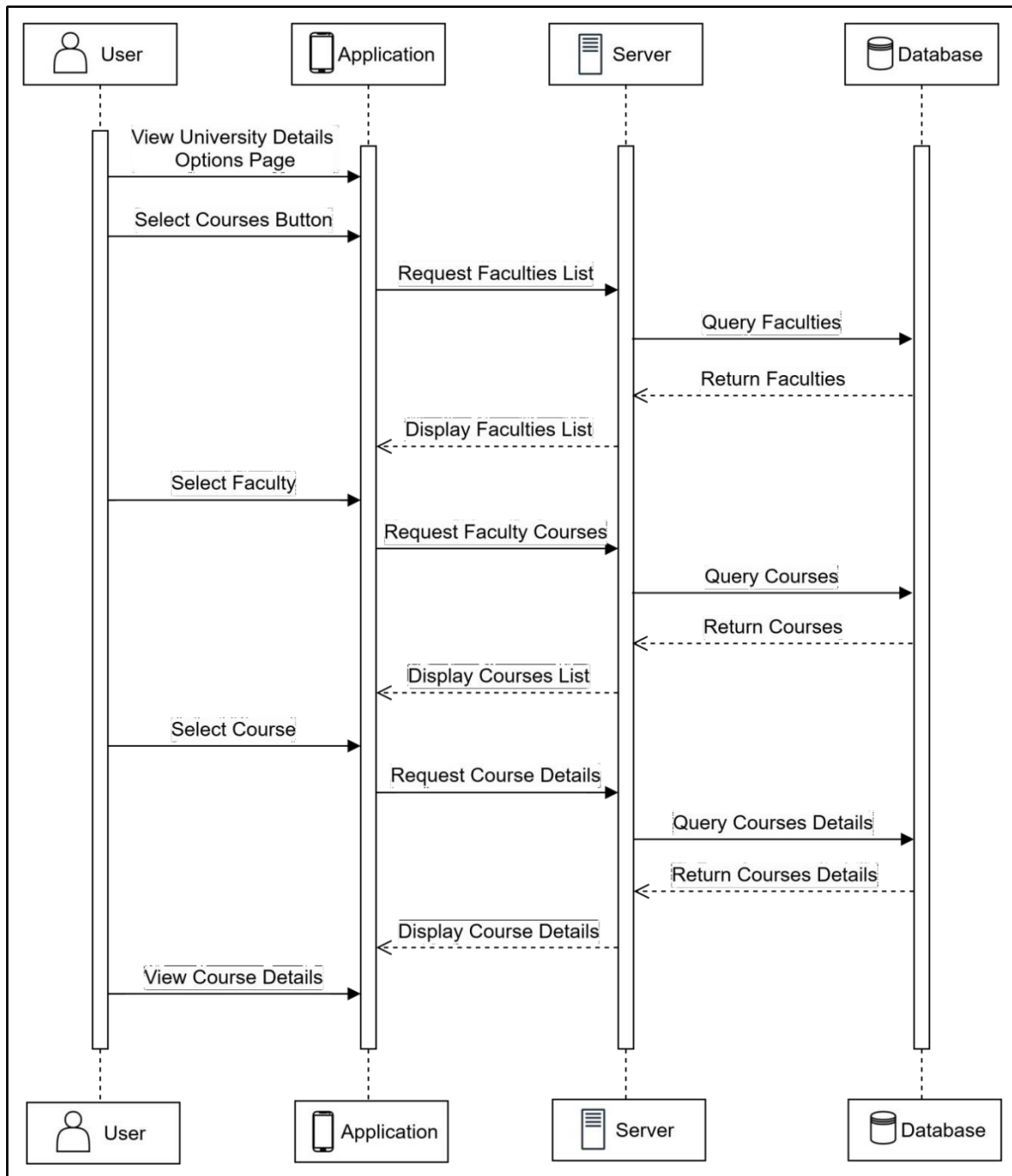


Figure 3.12: Sequence Diagram (Public University List Screen - Courses)

Figure 3.12 shows the flow for accessing course information. Students select a faculty after choosing a university. The application fetches courses for the selected faculty from the database, displaying detailed course information, including credit hours, duration, and course structure.

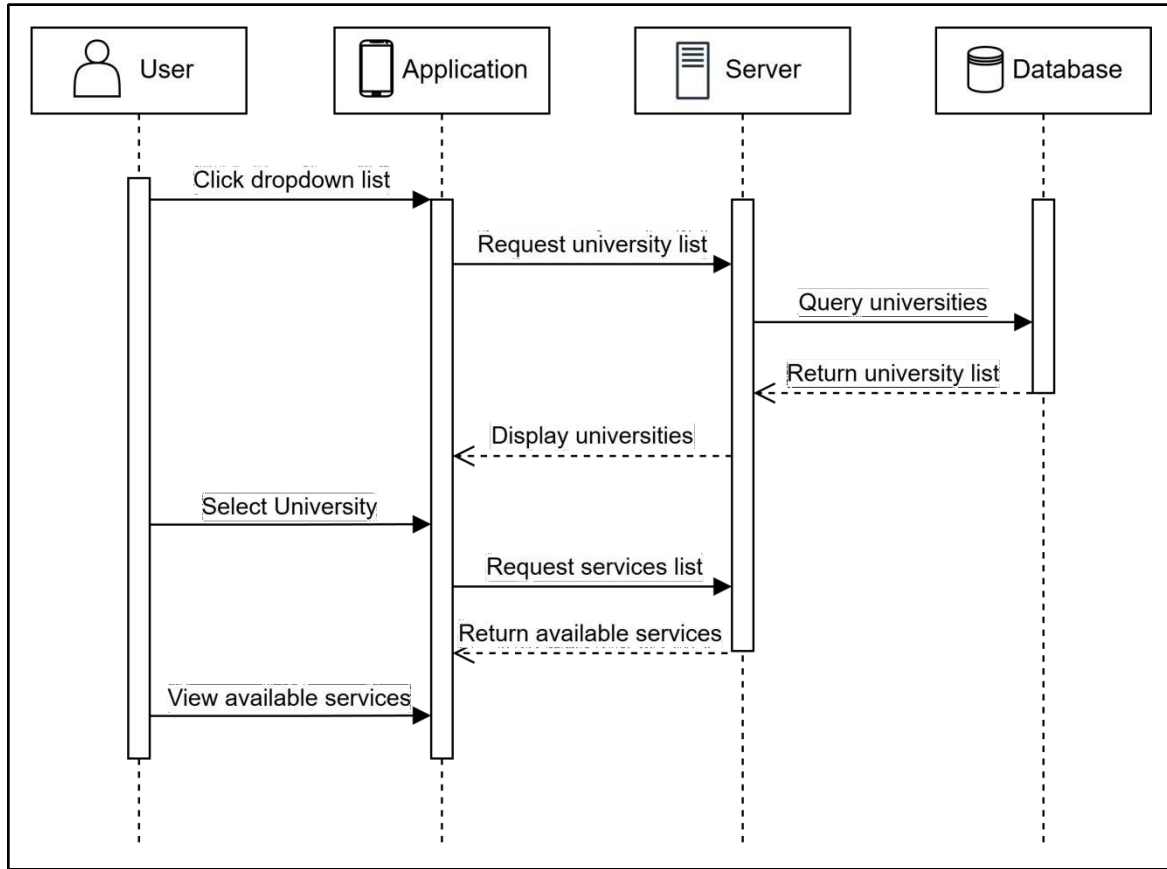


Figure 3.13: Sequence Diagram (Dashboard Screen)

Figure 3.13 shows the general interaction on the Dashboard Screen. Students use the drop-down to select a university, and the application retrieves data for accommodation, eateries, grocery stores, student activity places, and transportation based on the selection.

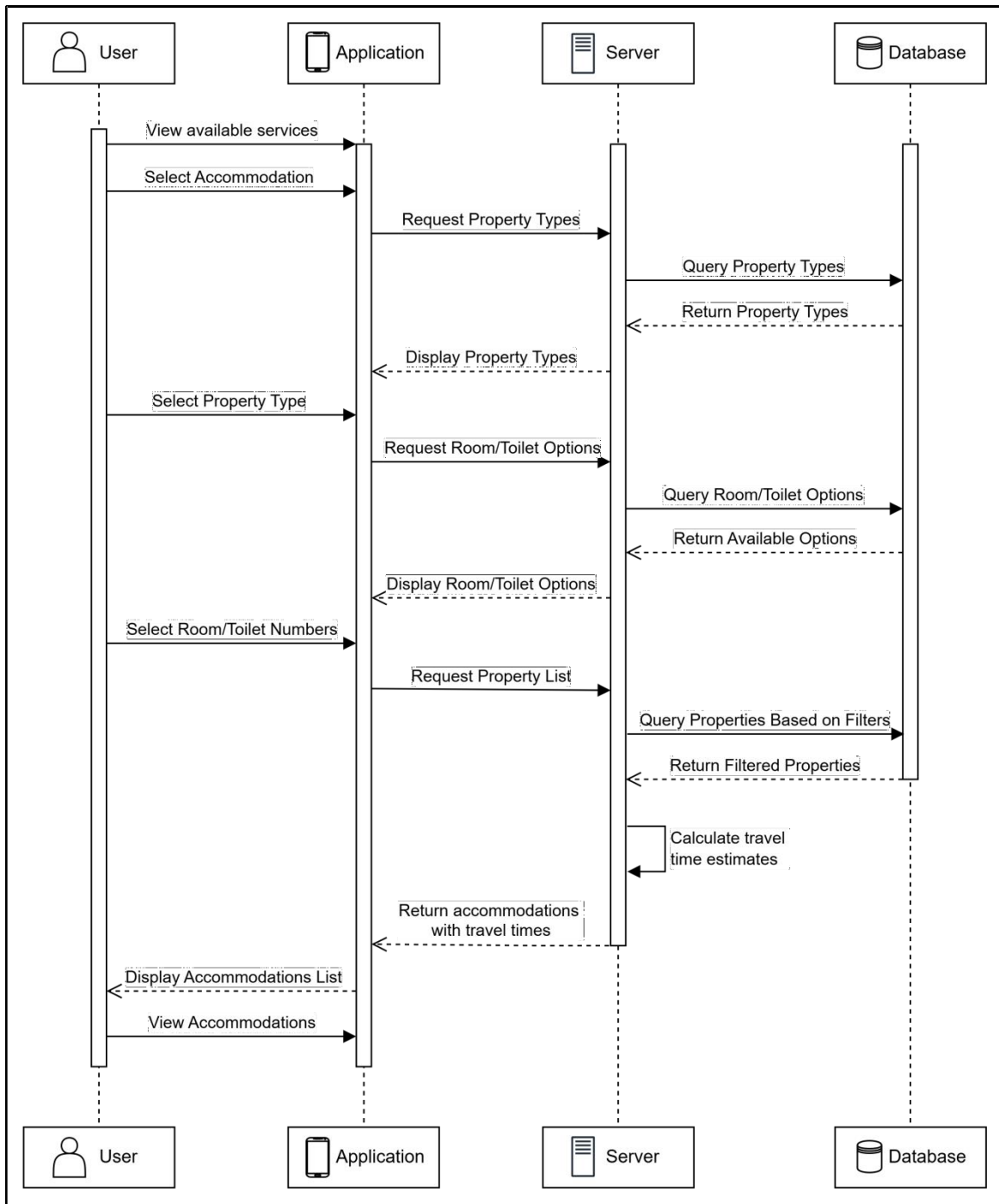


Figure 3.14: Sequence Diagram (Dashboard Screen - Accommodation)

Figure 3.14 shows the process for browsing accommodations. After specifying filters (e.g., property type, number of rooms, number of toilets), the application queries the database to fetch matching options and displays them, including real-time travel estimates.

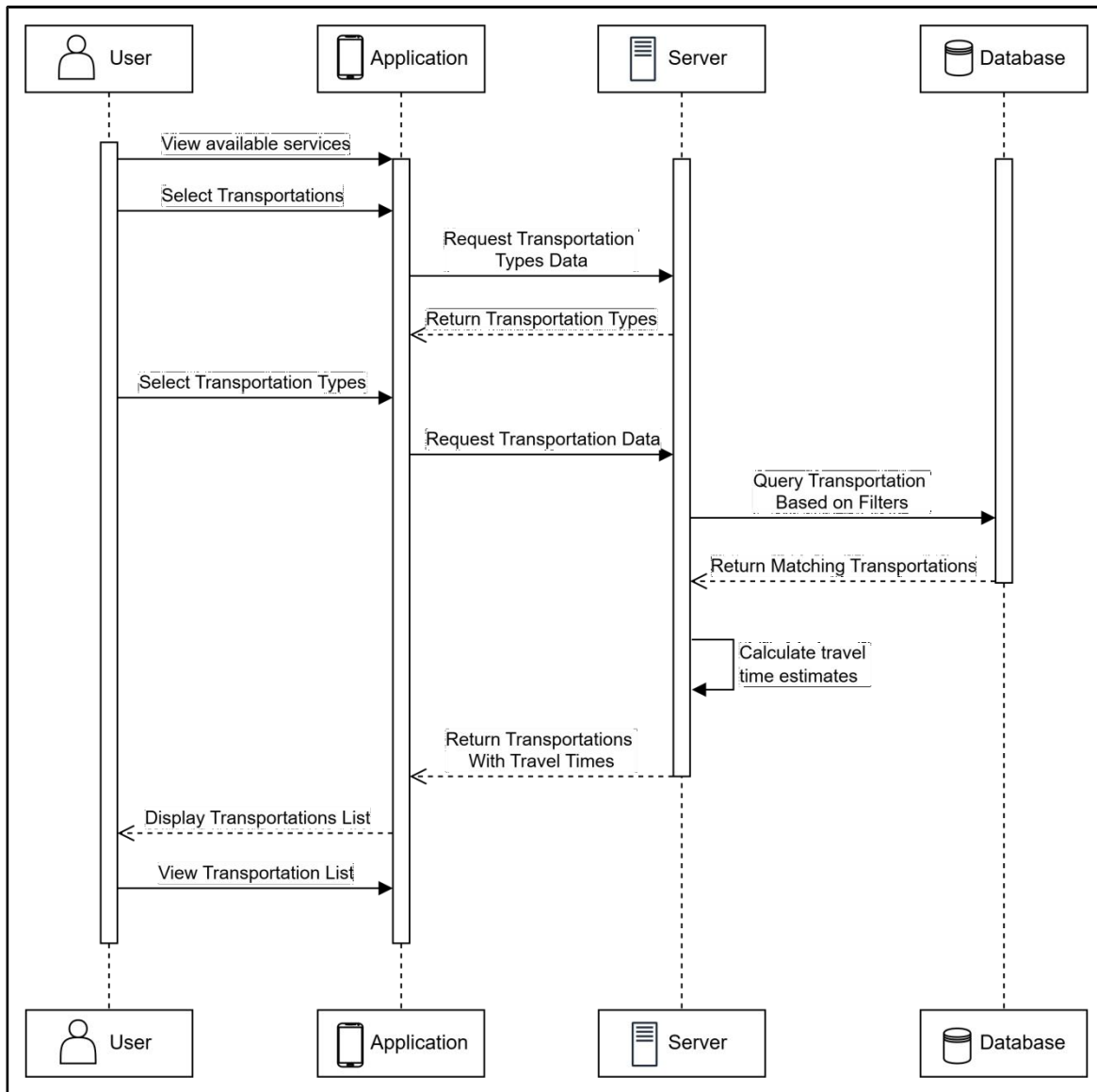


Figure 3.15: Sequence Diagram (Dashboard Screen - Transportation)

Figure 3.15 illustrates how transportation options are retrieved. Students select the type of transportation (e.g., bus, taxi, rapid transit, or car rental). The application filters data based on the user's criteria and displays relevant options with real-time travel estimates.

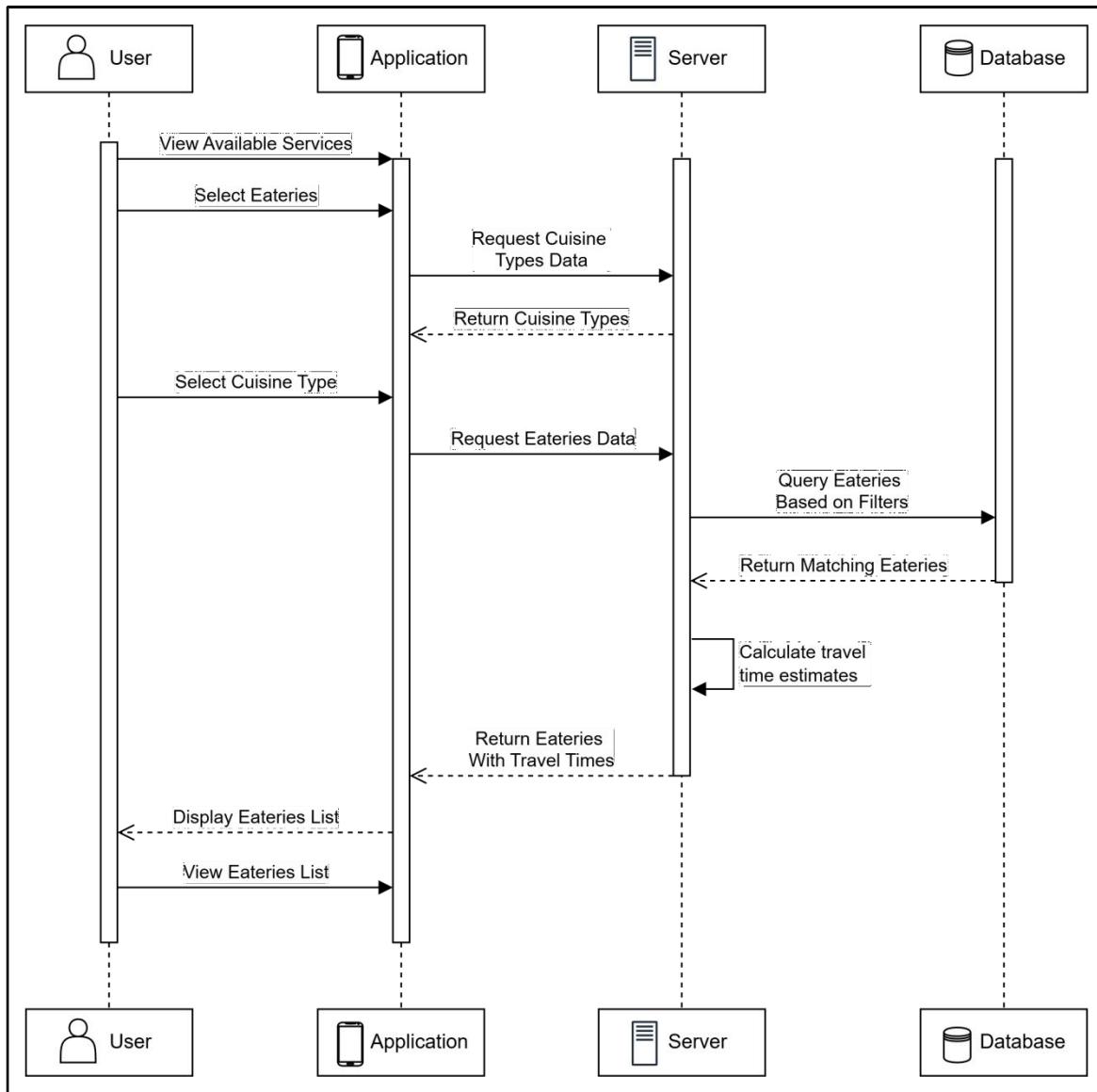


Figure 3.16: Sequence Diagram (Dashboard Screen - Eateries)

Figure 3.16 demonstrates the process for viewing eateries. Students select their preferred cuisine type (Western, Asian, Italian, Vegetarian). The application queries the database for matching eateries and displays the results, along with real-time travel time estimates.

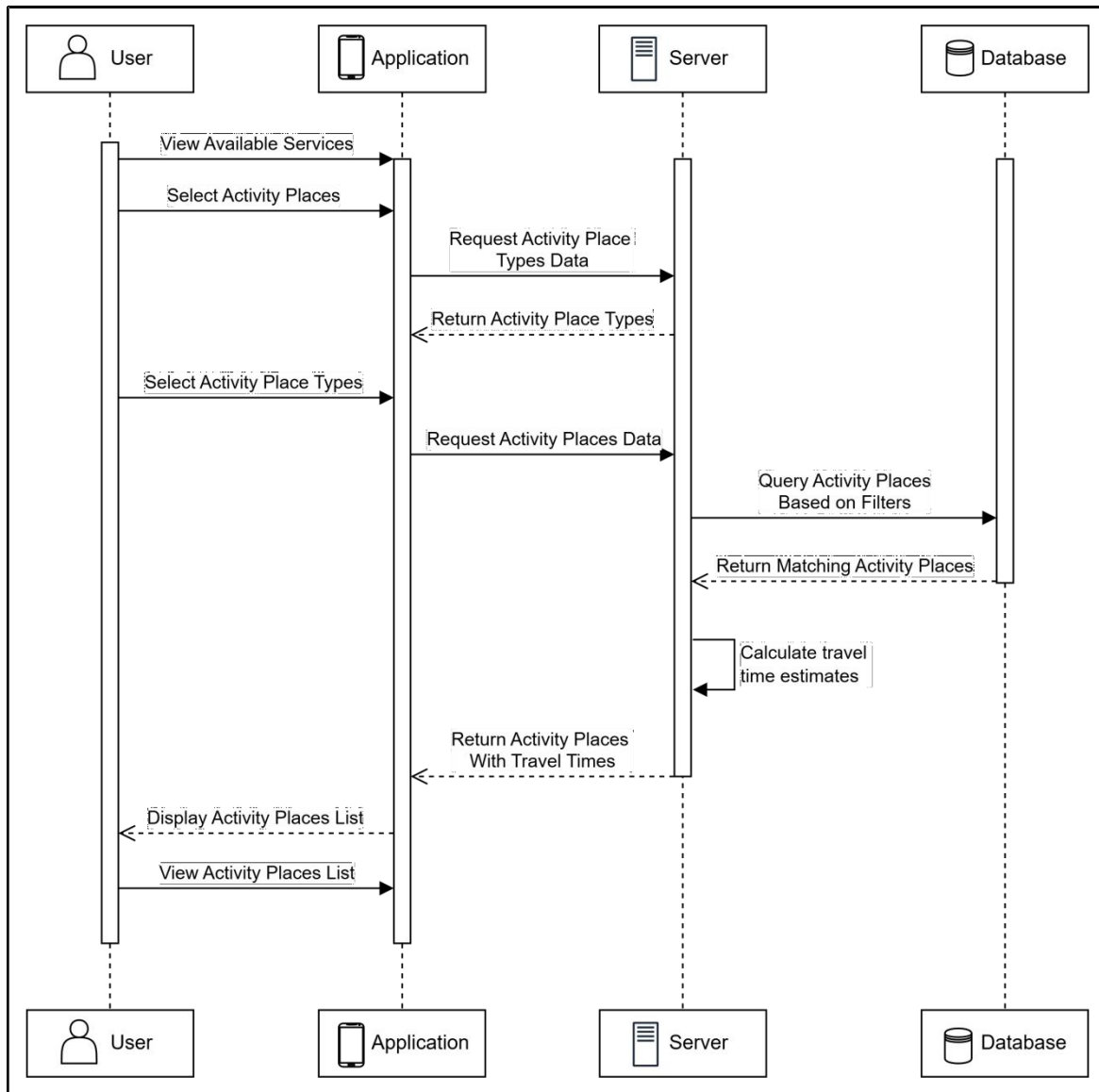


Figure 3.17: Sequence Diagram (Dashboard Screen - Student Activity Places)

Figure 3.17 illustrates how activity places are retrieved. Students select activity categories (e.g., attractions, shopping malls, outdoor activities). The application queries the database for matching options and displays results, including travel time estimates.

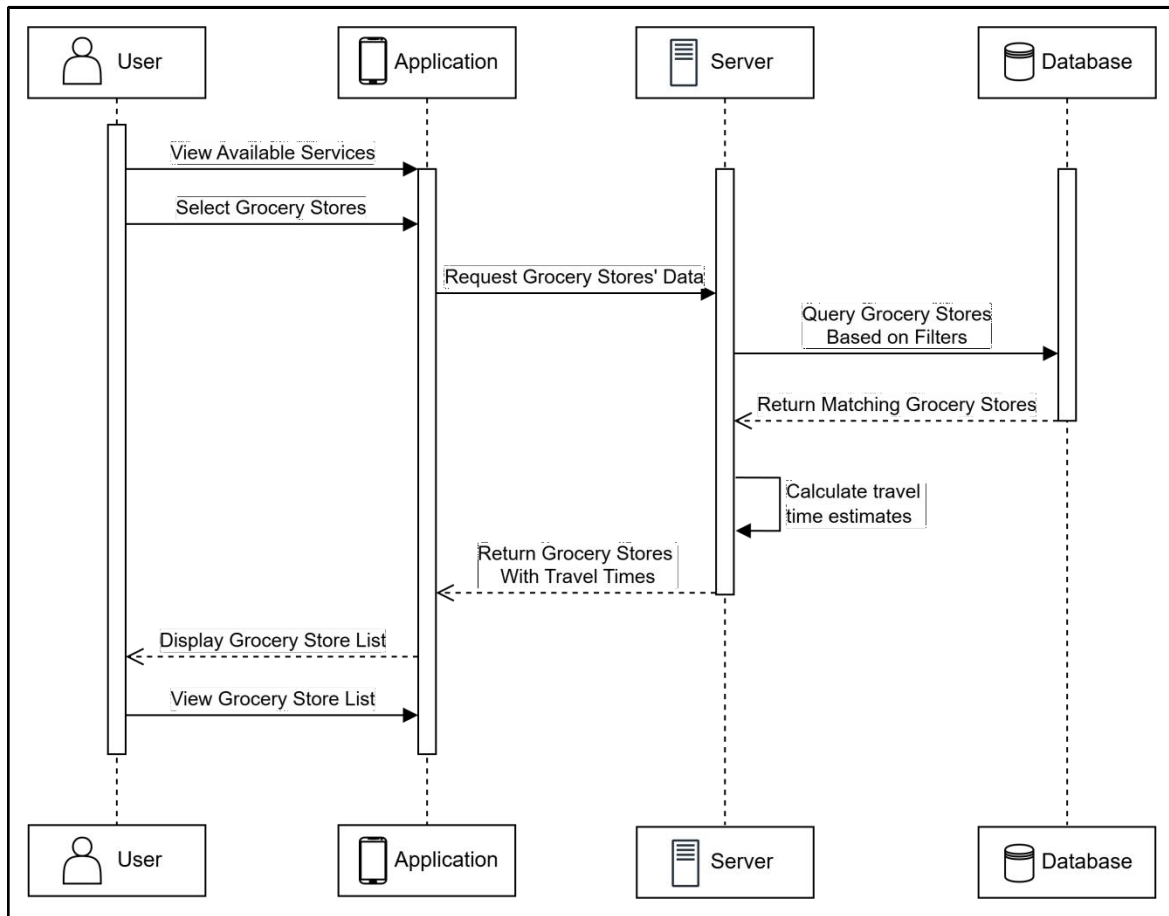


Figure 3.18: Sequence Diagram (Dashboard Screen - Grocery Stores)

Figure 3.18 shows how nearby grocery stores are displayed. Students select a university, and the application retrieves a list of grocery stores based on the selected location. Real-time travel estimates are shown for each grocery store.

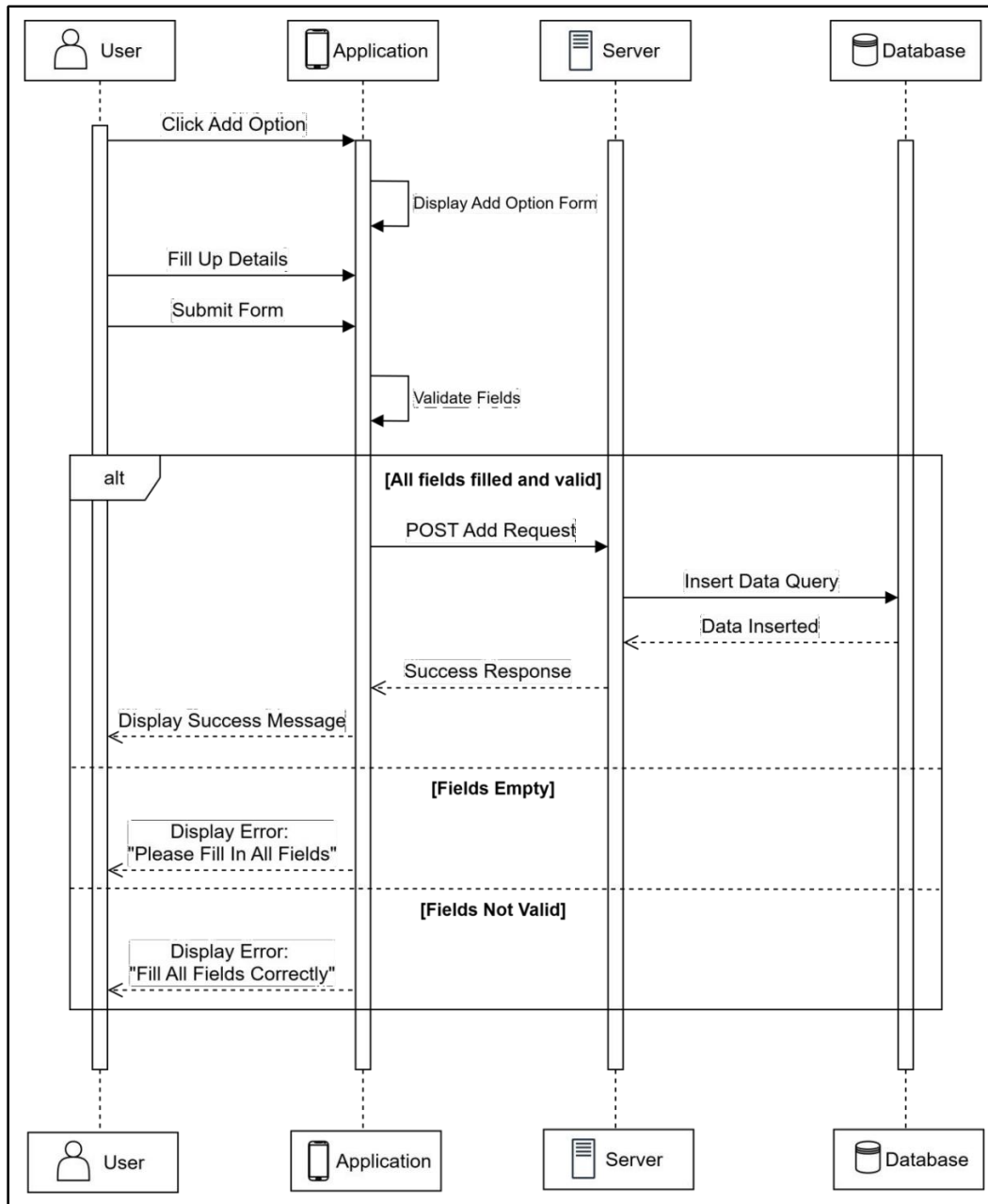


Figure 3.19: Sequence Diagram (Add Option for All Dashboard Services and Event Calendar)

Figure 3.19 depicts how users add options for Dashboard services or personal events. After entering details, the application validates the input and stores the new entry in the database, making it visible to other users (for Dashboard services) or only the current user (for personal events).

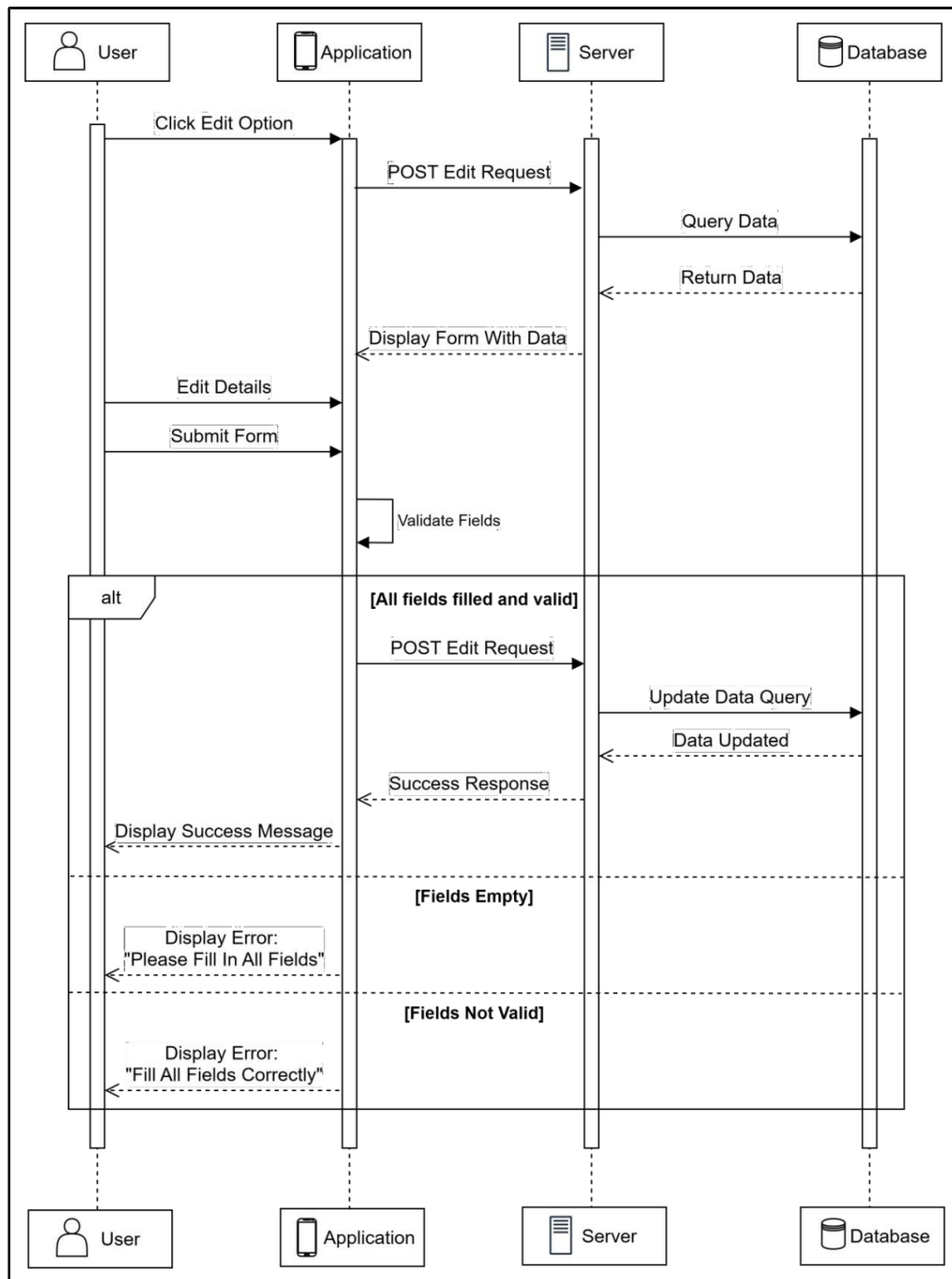


Figure 3.20: Sequence Diagram (Edit Option for All Dashboard Services and Event Calendar)

Figure 3.20 shows the edit functionality. Users can only modify entries they have added. The application fetches the data for the selected entry, allows edits, and updates the database with the new details.

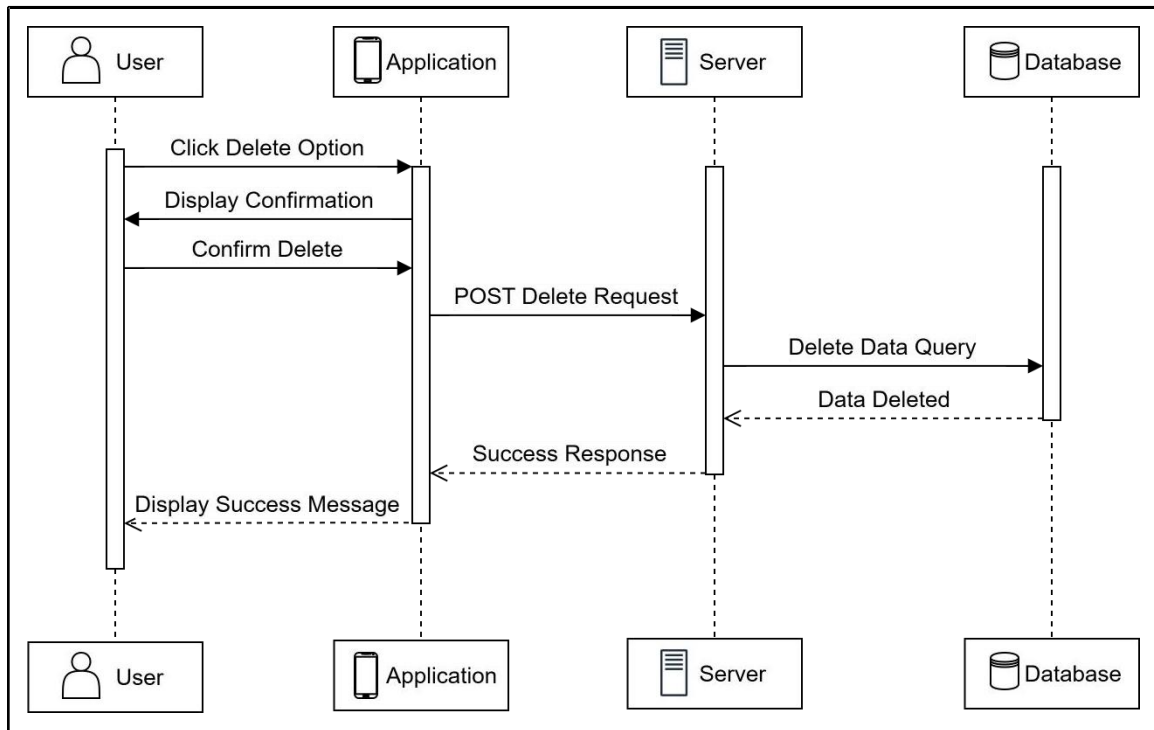


Figure 3.21: Sequence Diagram (Delete Option for All Dashboard Services and Event Calendar)

Figure 3.21 illustrates the delete process. Users select an entry they created, confirm the deletion, and the application removes the entry from the database. Only user-created options can be deleted.

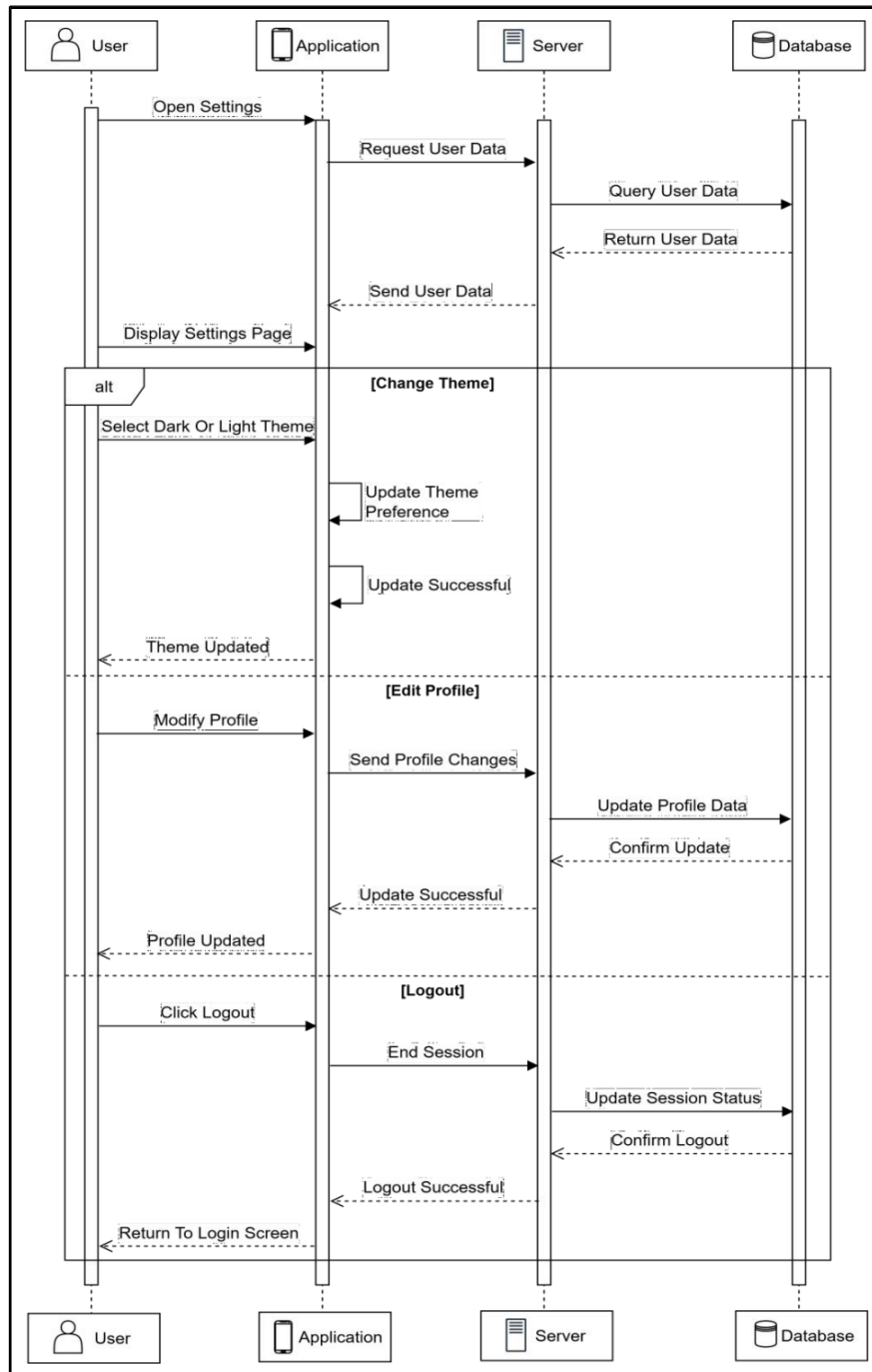


Figure 3.22: Sequence Diagram (Settings - Change Theme, Edit Profile, Logout)

Figure 3.22 showcases user interactions in the Settings Screen. Users can toggle between dark and light themes, edit their profile information, or log out. Changes are updated in the database for themes and profile edits, and the session is terminated upon logout.

3.5.5 User Interface Design

This section is discussing about the user interface design. Navigation design and layout design are discussed in different sections.

3.5.5.1 Navigation Design

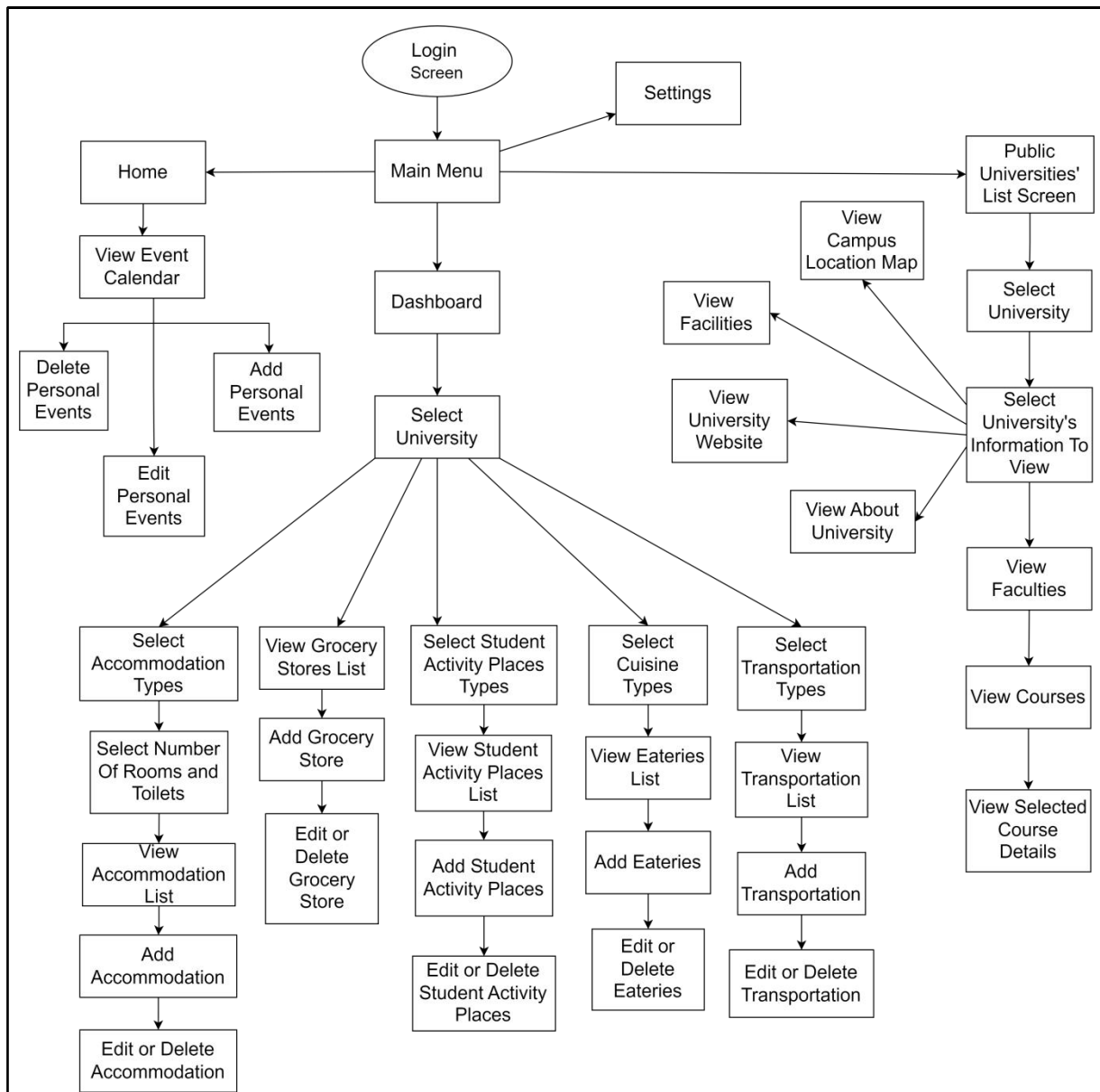


Figure 3.23: Navigation Flowchart

Figure 3.23 illustrates the navigation diagram of the Unifo+ application, designed to provide a seamless experience for students. The application starts with the Login Screen, leading users to the Main Menu, which features a bottom tab navigation system: Public University List Page, Home Page, Dashboard Page, and Settings Page. The University Details Page offers comprehensive information about Malaysia's 20 public universities. Users can access the University Campus Location Map, University Website, Facilities, Courses, and Location. Selecting a course leads to specific details such as course description, credit hours, and duration. External links like Google Maps and university websites are also integrated. Meanwhile, the Home Page will have an event calendar, that displays upcoming activities, events, and competitions which varies from different Public Universities to encourage student engagement. Students can also add, edit or delete their own personal events such as celebrations or other activities. The Dashboard Page offers essential services such as Nearby Accommodation, Eateries, Grocery Stores, Student Activity Places, and Transportation. Users can filter options (e.g., accommodation preferences, cuisine types) for personalized results, enhancing convenience. Additionally, the application allows community contributions, enabling students, seniors, and alumni to add information like new accommodations, eateries, or transport options, where they can also edit or delete the options they have added. The Settings Page provides customization features, including Dark or Light Mode, Profile Management, and a secure Logout option.

3.5.6 Layout Design

The layout design is the final design before clearance for the second phase, system design phase. This design contributes to fulfil the specifications outlined in the system requirements document. This approach can help developers physically manage output, input, and security issues.

The design generally defines how individuals interact with a computer system and includes all hardware, software, screen, menus, functions, and output, which influence two-way communications between the user and computer.

3.5.6.1 Sign Up Page

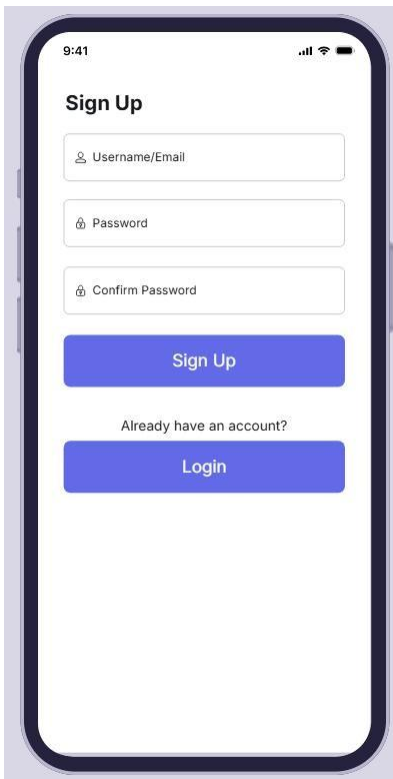


Figure 3.24: Sign Up Page

Figure 3.24 illustrates the “Sign Up” page, where new users can register for an account to access the system. The screen contains input fields for essential information, such as username, email, password, and confirm password. A Sign-Up button is provided to complete the registration process. Users can also find a link to the login screen at the bottom if they already have an account. This screen ensures secure and streamlined user onboarding for the application.

3.5.6.2 Log In Page

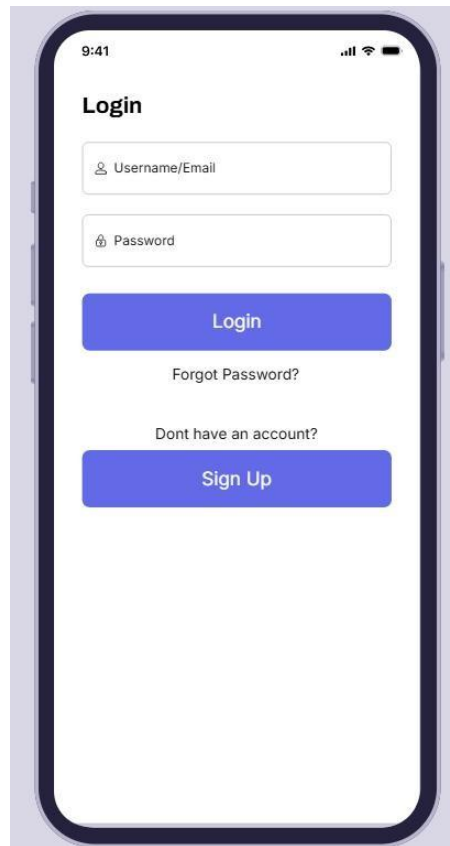


Figure 3.25: Log In Page

Figure 3.25 shows the “Login” page, where users are required to log in to access the system's functionalities. The page contains input fields for the user's email and password, along with a Login button. A Forgot Password link is available to assist users in resetting their credentials if needed. This page ensures that only authenticated users can access the application features. For new users, they go to the “Sign Up” button to register for an account.

3.5.6.3 Forgot Password Page

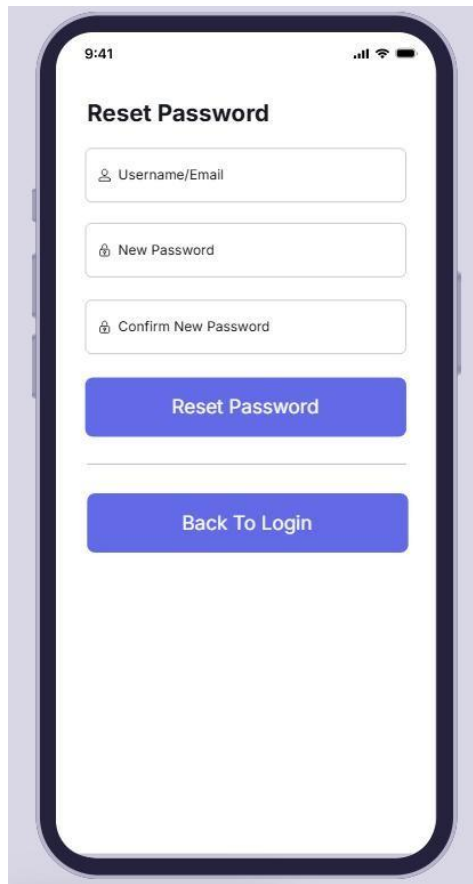


Figure 3.26: Forgot Password Page

Figure 3.26 illustrates the “Forgot Password” page, where users can reset their account password if they have forgotten it. The page contains a single input field for users to enter their registered email address. Once submitted by clicking the “Reset Password” button, the system sends a password reset link to the provided email. A confirmation message is displayed upon successful submission. Additionally, the page includes a link to navigate back to the “Login” page for convenience. This feature ensures users can regain access to their accounts securely and efficiently.

3.5.6.4 Public Universities Page

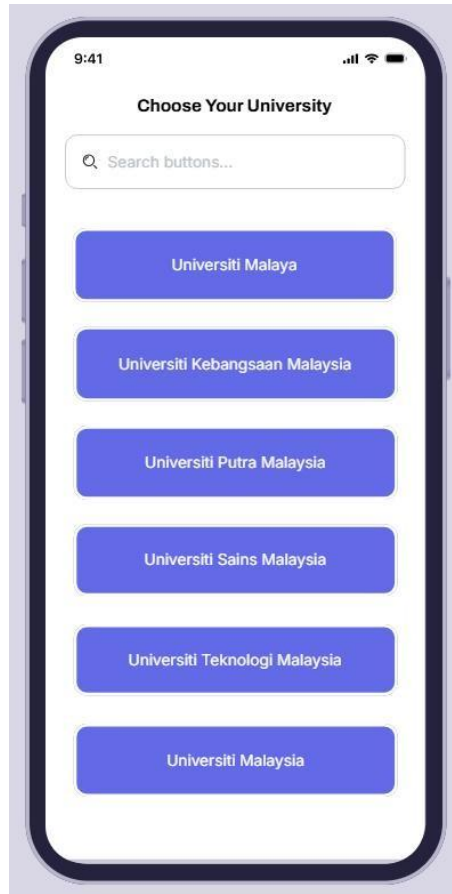


Figure 3.27: Public Universities Page

Figure 3.27 displays the “Public Universities” page, where users can browse through a comprehensive list of Malaysian public universities. Each university is presented in an organized card-style layout with its name, logo, and a brief description. Users can click on any university card to navigate to the university's details page. This page also includes a search bar at the top, enabling users to quickly find a specific university. A navigation bar is available at the bottom of the page for seamless access to other features of the application, such as the dashboard or settings. This page serves as the gateway to exploring detailed information about each university.

3.5.6.5 University Details Page

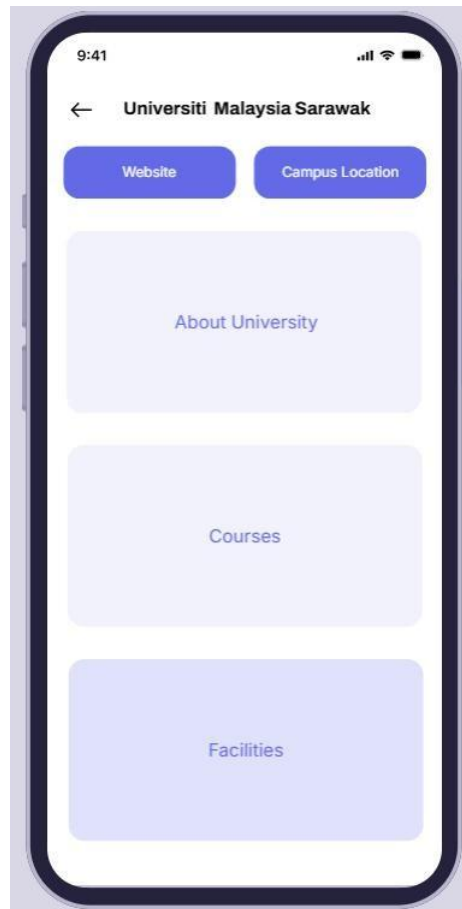


Figure 3.28: University Details Page

Figure 3.28 displays the “University Details” page, which offers an overview of a selected university. At the top, users can quickly access the university's official website and campus location through dedicated buttons. Below, options for About University, Courses, and Facilities provide detailed insights. Users can explore the university’s details, such as descriptions and images, view faculties and course details, or learn about university amenities like libraries and sports complexes. This page consolidates essential information, ensuring a seamless and intuitive experience for accessing university-specific details.

3.5.6.6 About University Page

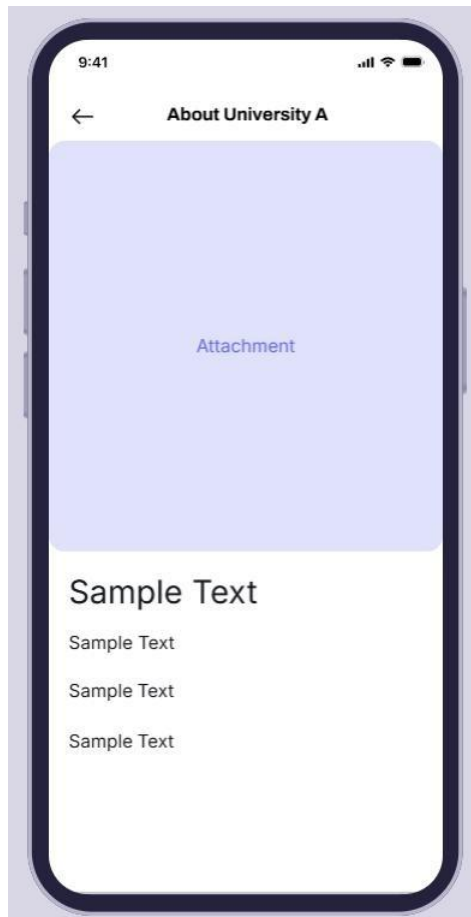


Figure 3.29: About University Page

Figure 3.29 illustrates the “About University” page, accessed from the “University Detail” page. On this page, users can view detailed information about a specific university. After selecting a university from the list, such as “Universiti Malaya”, users can learn more about the university’s reputation, ranking, city in proximity, and academic offerings. This page provides general information, including the university's establishment year, notable achievements, and any relevant updates or highlights about the institution. The “About University” page helps students understand the university's background, enabling them to make more informed decisions about their academic journey.

3.5.6.7 University Facilities Page



Figure 3.30: University Facilities Page

Figure 3.30 shows the “University Facilities” page, accessible from the “Public Universities” page, after the user selects a university to view. This page provides an interactive map of the selected university. Users can navigate through the map and access detailed information about the facilities available at the selected university, such as libraries, sports complexes, research centers, and other campus amenities. Users can click on any pinpoint inside the map, which represents each facility, allowing students to quickly find specific locations. Users can explore each facility through descriptions, images, or additional resources, enabling them to better understand the university.

3.5.6.8 University Faculties Page



Figure 3.31: University Faculties Page

Figure 3.31 displays the “University Faculties” page, accessible after selecting a university from the “University Details” page. This page provides a list of all faculties within the chosen university. Users can select a faculty to proceed and explore its associated courses.

3.5.6.9 University Courses Page

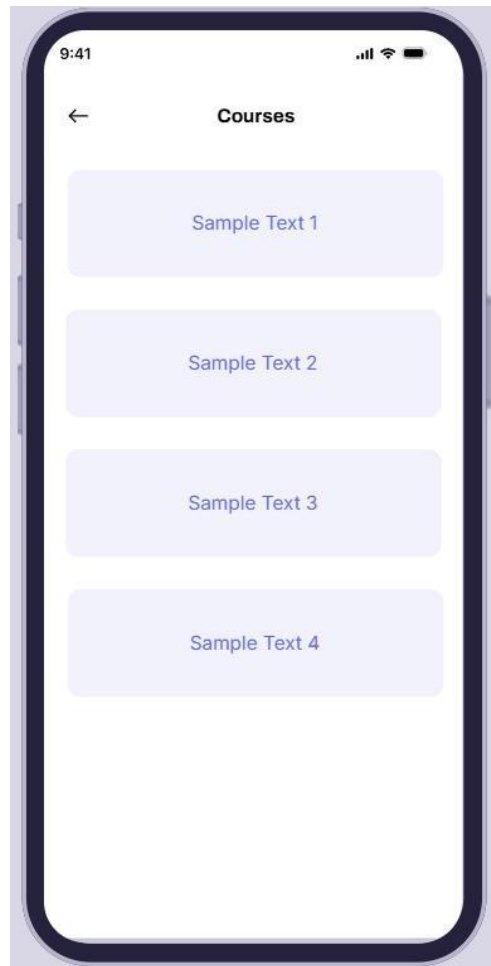


Figure 3.32: University Courses Page

Figure 3.32 showcases the “University Courses” page, displayed after a user selects a faculty from the “Faculties” page. The page presents a detailed list of courses offered by the chosen faculty, allowing users to browse and choose a specific course to view more information.

3.5.6.10 University Course Details Page

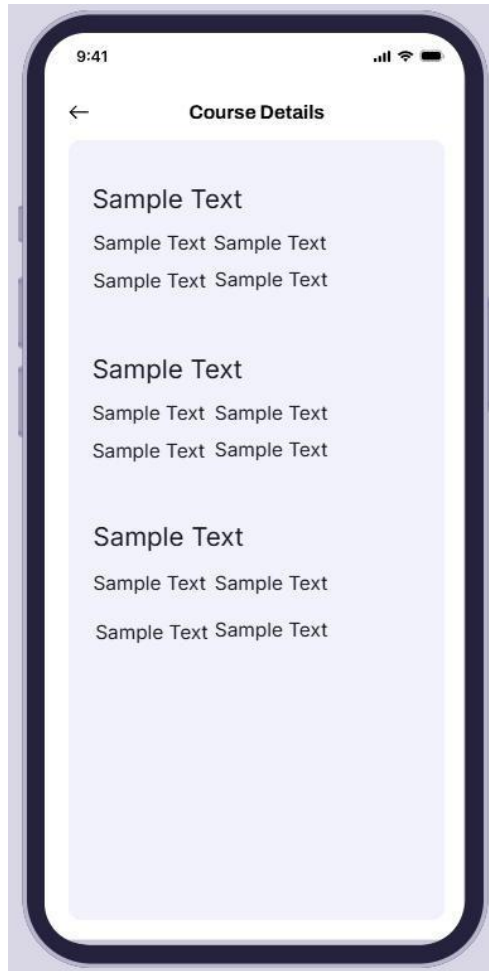


Figure 3.33: University Course Details Page

Figure 3.33 illustrates the “University Course Details” page, accessible after selecting a course from the “University Courses” page. This page provides comprehensive information about the chosen course, including its description, prerequisites, credit hours, duration, and any associated fees, ensuring users can make informed decisions about their academic options.

3.5.6.11 Dashboard Page

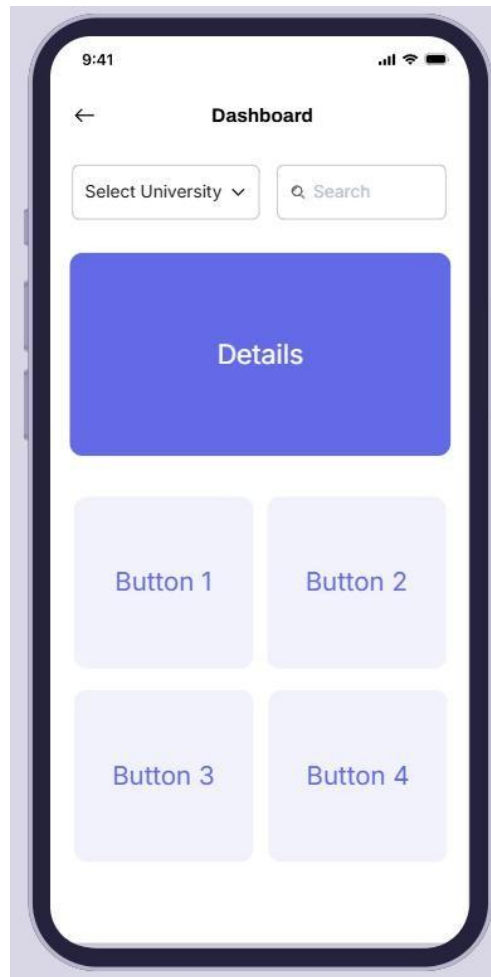


Figure 3.34: Dashboard Page

Figure 3.34 illustrates the “Dashboard” page, the first page users see after successfully logging into the system. On this page, users can access multiple features, including university information, nearby accommodations, eateries, groceries, transportation, student activities, and the campus map. Each feature is represented by a button, making navigation intuitive and user-friendly. The header bar at the top allows seamless access to additional settings or navigation options.

3.5.6.12 Accommodation Types Page

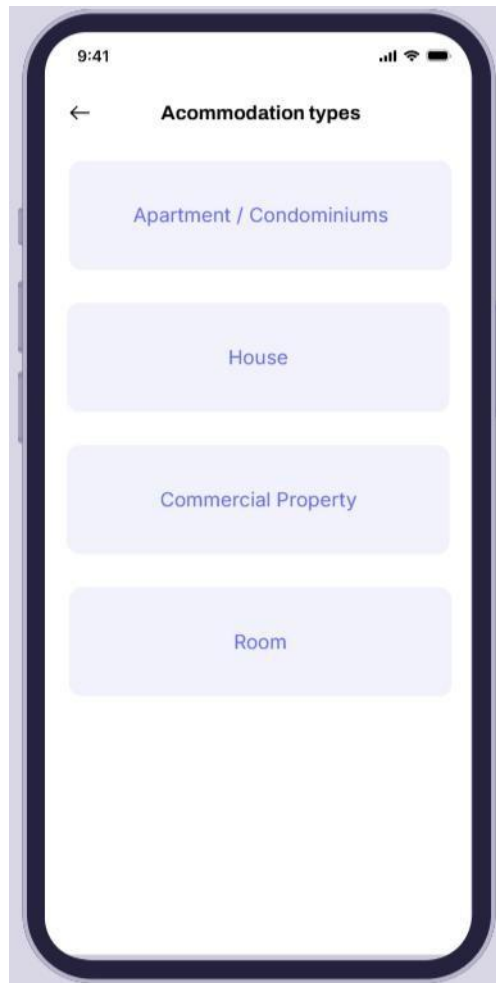


Figure 3.35: Nearby Accommodation Types Page

Figure 3.35 illustrates the “Accommodation Types” page, accessed from the “Dashboard” page. On this page, users can select the type of accommodation they are looking for, including apartments, houses, commercial spaces, and rooms. Each type is represented by a button, making navigation intuitive and user-friendly.

3.5.6.13 Accommodation Page

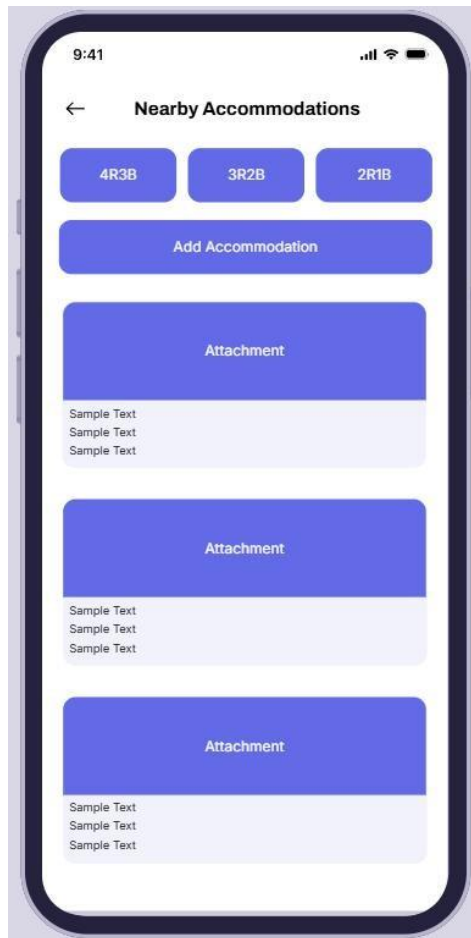


Figure 3.36: Accommodation Page

Figure 3.36 illustrates the “Accommodation” page, accessed from the “Accommodation Types” page. On this page, users can select their preferred number of rooms and toilets for the property, such as 4 rooms, 3 toilets or 3 rooms, 2 toilets. The page then displays a list of accommodations that match the selected criteria, making it easy for users to find properties that suit their needs. Additionally, users have the option to add their own accommodation suggestions, allowing for a more comprehensive and personalized search experience.

3.5.6.14 Accommodation (Room) Page

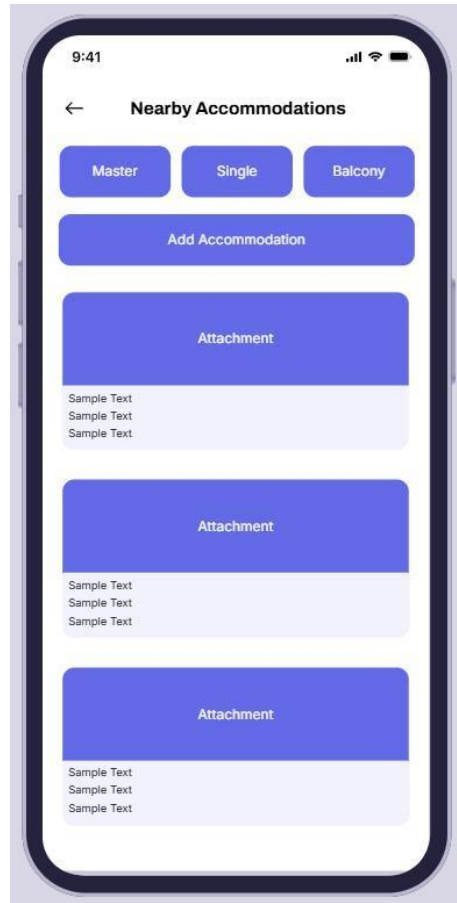
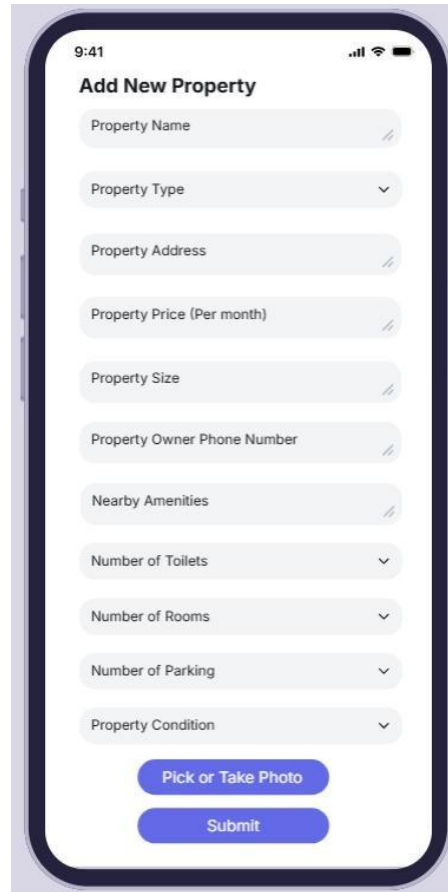


Figure 3.37: Nearby Accommodation (Room) Page

Figure 3.37 illustrates the “Accommodation (Room)” page, accessed from the “Accommodation Types” page. On this page, users can select the type of room they are looking for, such as master, balcony, or single rooms. The page displays a list of properties that match the selected room type, helping users find accommodations that fit their preferences. Users also have the option to add new room listings for other or future students to refer to, enhancing the search experience and community contribution.

3.5.6.15 Add Accommodation Page



9:41

Add New Property

Property Name

Property Type

Property Address

Property Price (Per month)

Property Size

Property Owner Phone Number

Nearby Amenities

Number of Toilets

Number of Rooms

Number of Parking

Property Condition

Pick or Take Photo

Submit

Figure 3.38: Add Accommodation Page

Figure 3.38 illustrates the “Add Accommodation” page, accessed from the “Accommodation” page. On this page, users can add new accommodation listings for future students to explore. The screen allows users to input the property’s name, address, size, and select property types such as apartments, houses, or commercial properties. Additionally, users can specify the number of rooms and toilets in the property, and choose the preferred room type (e.g., Master, Medium, or Single). Users can also input essential details like property condition, price, nearby amenities, number of parking and contact information. This page enables students, seniors, and alumni to contribute valuable accommodation options to the platform, enhancing the search

experience for others. Users can also modify or delete their added options if necessary, ensuring that their added options remain accurate.

3.5.6.16 Property Details Page

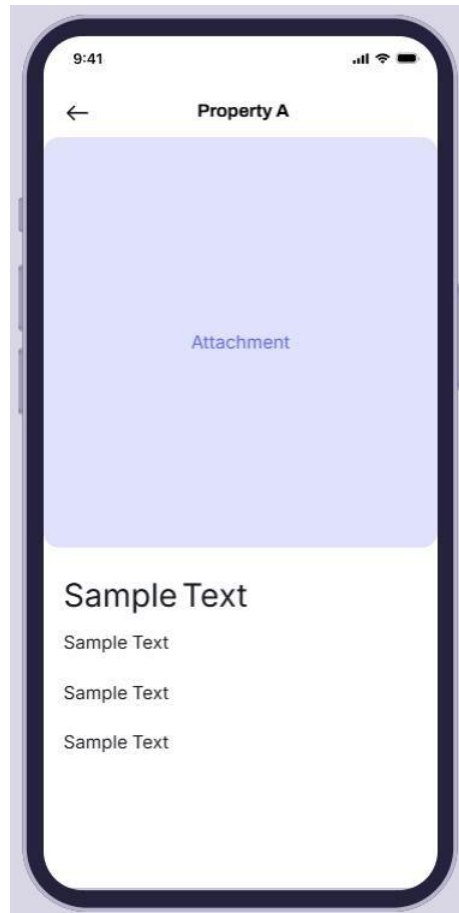


Figure 3.39: Property Details Page

Figure 3.39 illustrates the “Property Details” page, accessed by clicking on a specific property from the “Accommodation” page. On this page, users can view an in-depth description of the selected property, helping them better understand the available accommodation options. This page provides more detailed information about the property, including the number of rooms, bathrooms, available amenities, and rental prices. Users can also view clearer photos of the property, location details, and a real-time travel estimate from the current location to the property.

3.5.6.17 Grocery Stores Page

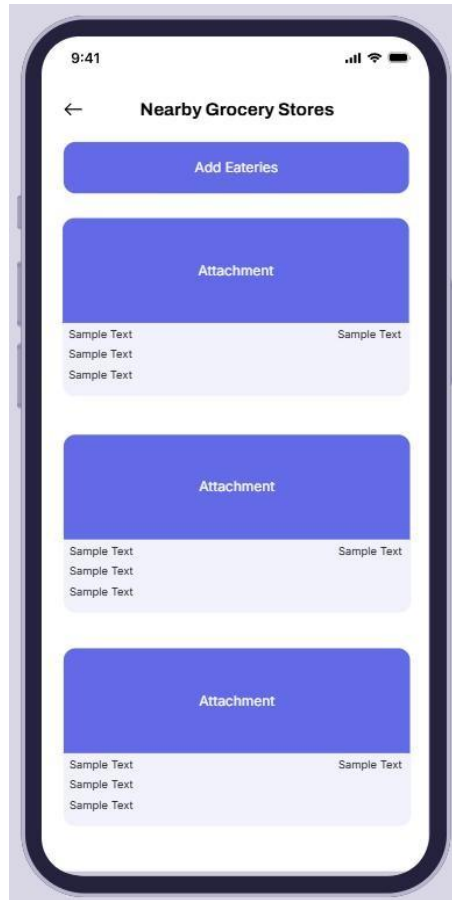
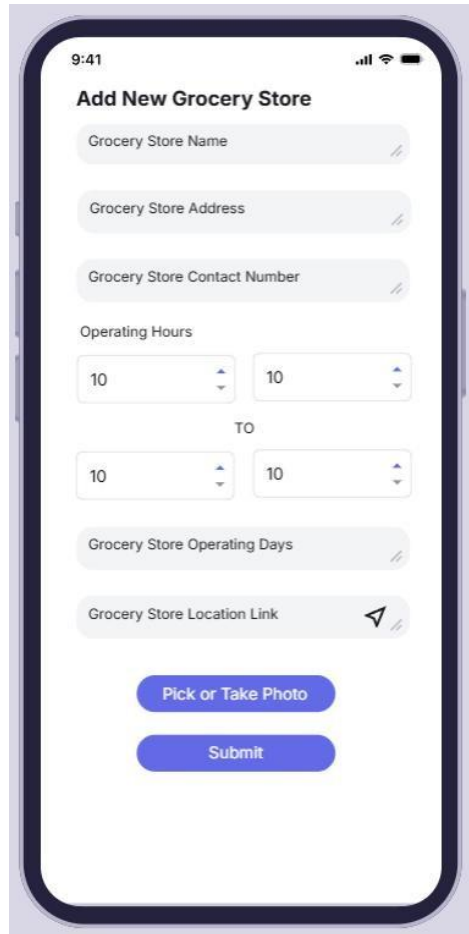


Figure 3.40: Grocery Stores Page

Figure 3.40 illustrates the “Grocery Stores” page, accessed from the “Dashboard” page. This page enables users to locate grocery stores near their university. Users can view store details such as address, contact information, and the types of goods offered. The layout is designed to make finding essential supplies convenient and efficient.

3.5.6.18 Add Grocery Store Page



The image shows a mobile application screen titled "Add New Grocery Store". The screen contains the following elements:

- At the top, the status bar shows the time "9:41" and signal/battery icons.
- The title "Add New Grocery Store" is centered at the top of the form.
- Below the title are five input fields, each with a double-slash icon on the right:
 - "Grocery Store Name"
 - "Grocery Store Address"
 - "Grocery Store Contact Number"
 - "Operating Hours": This field consists of two time pickers. The first picker shows "10" and the second shows "10", with a "TO" label between them.
 - "Grocery Store Operating Days"
- Below the "Operating Hours" field is a "Grocery Store Location Link" field, which includes a location pin icon on the right.
- At the bottom of the form are two blue buttons:
 - "Pick or Take Photo"
 - "Submit"

Figure 3.41: Add Grocery Store Page

Figure 3.41 illustrates the “Add Grocery Store” page, accessed from the “Grocery Stores” screen. This page allows users to add new grocery stores in proximity to their selected university. Users can input the name of the grocery store, its address, opening hours and days, location link, and contact details. The added grocery stores will be available for future students to browse, helping them locate convenient stores for daily essentials. This feature enhances the platform by allowing users to contribute valuable local information that benefits the entire student community. Users can also modify or delete their added options if necessary, ensuring that their added options remain accurate.

3.5.6.19 Eateries Page

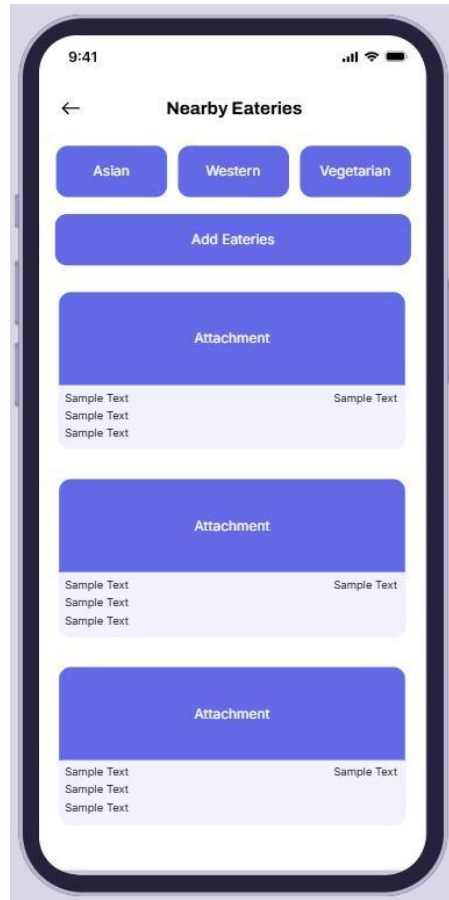


Figure 3.42: Eateries Page

Figure 3.42 shows the “Eateries” page, accessible through the “Dashboard” page. This page allows users to explore eateries around their selected university by choosing their preferred cuisine type, such as Asian, Western, or Vegetarian. After selecting a cuisine, the page displays a curated list of nearby eateries based on the user's cuisine selection. Additionally, the page offers an option for users to contribute by adding new eateries they recommend. This feature enables students to provide suggestions, enriching the database for future users seeking dining options near their campus.

3.5.6.20 Add Eateries Page

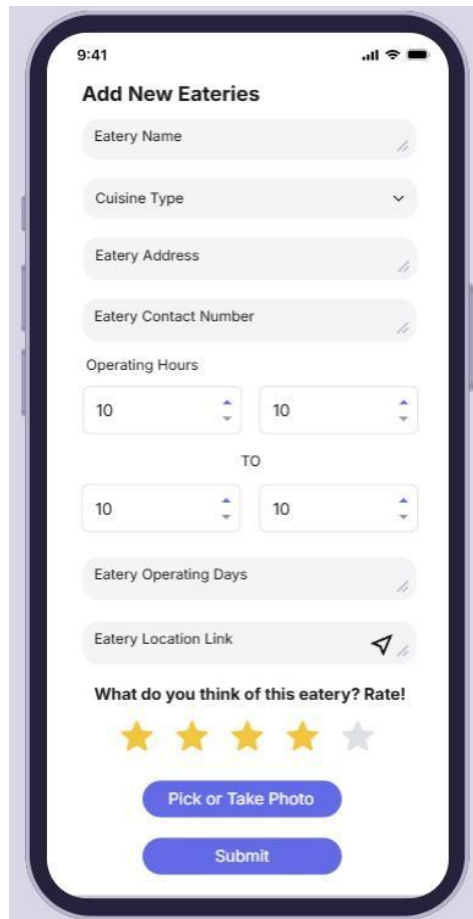
The image shows a mobile application interface for adding new eateries. At the top, the status bar shows the time as 9:41 and signal strength. The title 'Add New Eateries' is centered at the top of the form. Below the title are several input fields: 'Eatery Name' with a text entry field and a clear icon; 'Cuisine Type' with a dropdown menu; 'Eatery Address' with a text entry field and a clear icon; 'Eatery Contact Number' with a text entry field and a clear icon; 'Operating Hours' with two time selection fields (both set to 10) and a 'TO' separator; 'Eatery Operating Days' with a text entry field and a clear icon; and 'Eatery Location Link' with a text entry field and a location pin icon. Below these fields is a rating section titled 'What do you think of this eatery? Rate!' with five stars (the first four are yellow, the fifth is grey). At the bottom are two blue buttons: 'Pick or Take Photo' and 'Submit'.

Figure 3.43: Add Eateries Page

Figure 3.43 illustrates the “Add Eateries” page, accessed from the “Eateries” page. On this page, users can add new eateries or restaurants near their selected university. The page offers filters for different cuisine types such as Western, Asian, Italian, and Vegetarian. Users can provide details such as the eatery’s name, location, menu options, hours of operation, images, and contact information. They can also rate the current eatery they want to add to the list. Once added, these eateries will appear in the list of options for other students to view, helping them find dining spots that match their preferences and needs. Users can also modify or delete their added options if necessary, ensuring that their added options remain accurate.

3.5.6.21 Eatery Details Page

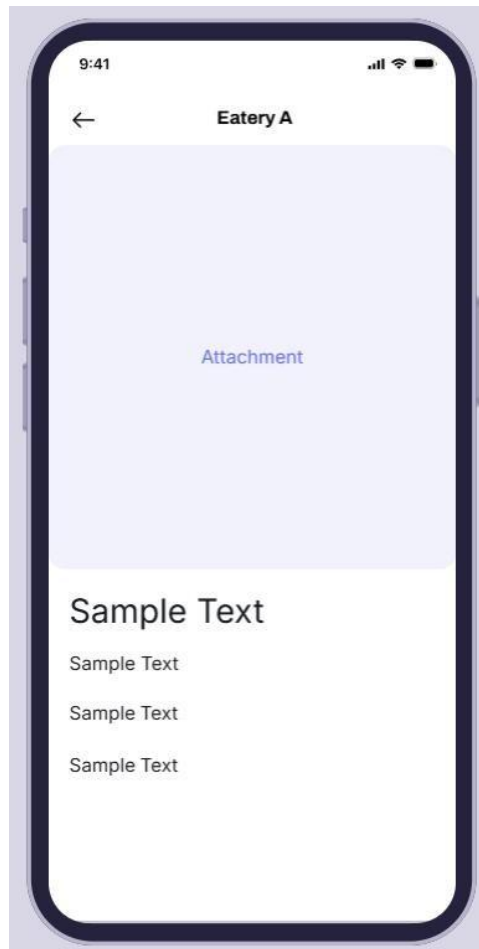


Figure 3.44: Eatery Details Page

Figure 3.44 illustrates the “Eatery Details” page, accessed by clicking on a specific eatery from the Eateries list. On this screen, users can learn more about the selected eatery, including its food categories and specific food offerings within each category. For example, if a user selects a Western restaurant, they can see detailed information about the dishes served, such as burgers, steaks, and pastas. The screen also includes the foods’ images, price range, and ratings. By providing a breakdown of food categories and menu options, the “Eatery Details” page helps students decide where to dine based on their preferences and dietary needs.

3.5.6.22 Student Activities Page

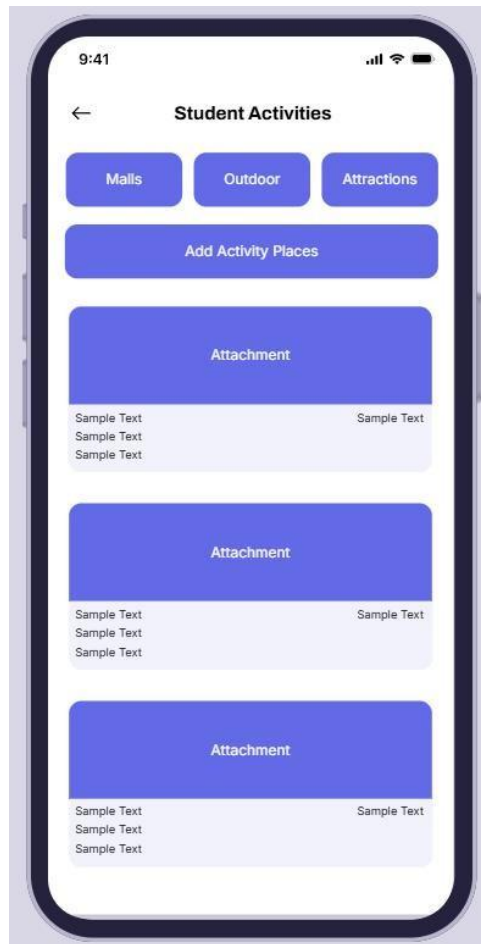


Figure 3.45: Student Activity Places Page

Figure 3.45 shows the “Student Activity Places” page, which is accessed through the “Dashboard” page. This page enables users to explore various activity locations near their selected university. Users can choose between categories such as malls, attractions, and outdoor activities. Upon selecting a category, the system displays a list of recommended activity places based on the chosen filter. Additionally, users have the option to add new activity places, allowing them to suggest interesting destinations for future students and users to explore, thereby enhancing the platform's utility for discovering local leisure opportunities.

3.5.6.23 Add Student Activities Page

The image shows a mobile application interface for adding a new activity place. The screen is titled "Add New Activity Place". It contains the following fields and controls:

- Place Name:** A text input field.
- Place Category:** A dropdown menu.
- Place Address:** A text input field.
- Place Contact Number:** A text input field.
- Operating Hours:** Two time selection fields (hour and minute) for start and end times, separated by a "TO" label.
- Place Operating Days:** A text input field.
- Place Location Link:** A text input field with a location pin icon.
- Description:** A text input field.
- Pick or Take Photo:** A blue button.
- Submit:** A blue button.

Figure 3.46: Add Student Activity Places Page

Figure 3.46 illustrates the “Add Student Activity Places” page, accessed from the “Student Activity Places” page. This page allows users to contribute new activity places for students, such as attractions, shopping malls, or outdoor activity spots. Users can enter details like the activity’s name, category (e.g., attraction, shopping mall, or outdoor activity), address, location link, operating hours and days, description, and contact information. By adding these locations, students, seniors, and alumni help build a valuable database of student-friendly places that can be referenced by future university students looking for leisure or social activities. Users can also modify or delete their added options if necessary, ensuring that their added options remain accurate.

3.5.6.24 Student Transportation Page

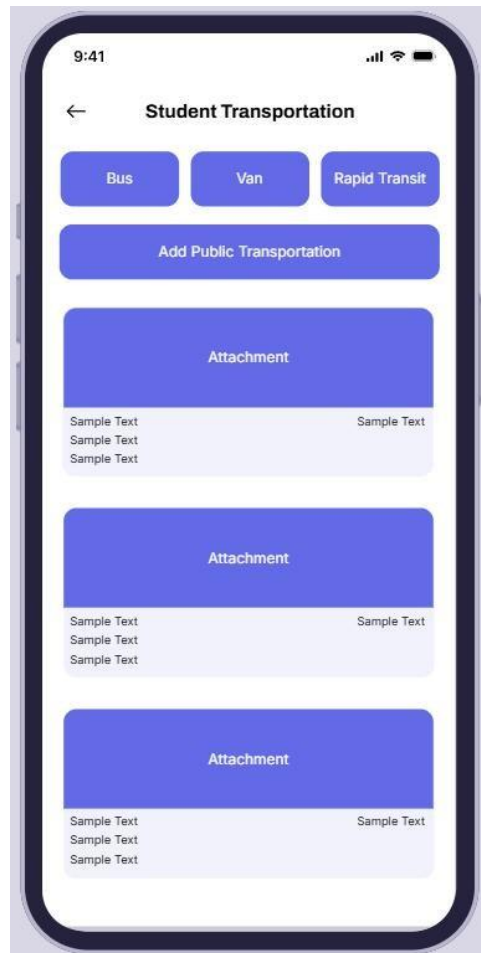


Figure 3.47: Transportation Page

Figure 3.47 shows the “Transportation” page, accessible from the “Dashboard” page. This page enables users to explore various transportation options available near their selected university. Users can filter transportation modes such as buses, vans, or rapid transit to find the most convenient options. Once a mode is selected, the system displays a list of available transportation services and their details. Additionally, users have the option to contribute by adding new transportation services to the list, allowing future students and users to benefit from updated and expanded options.

3.5.6.25 Add Transportation Page

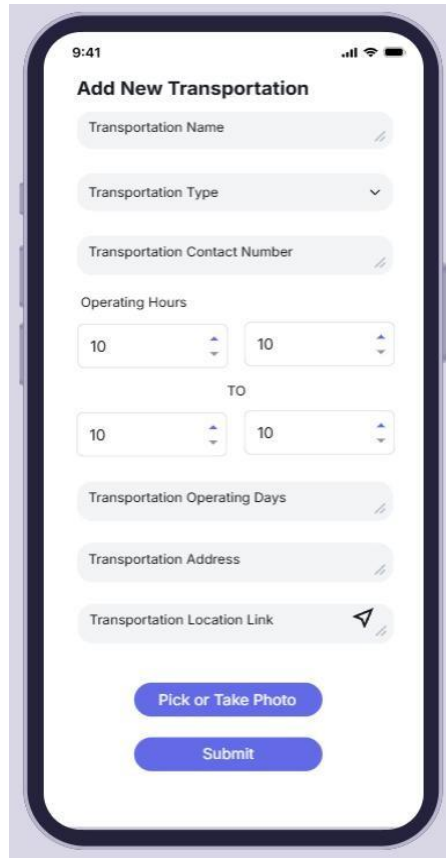
The image shows a mobile application interface for adding new transportation options. The screen is titled "Add New Transportation" and features several input fields: "Transportation Name", "Transportation Type" (a dropdown menu), "Transportation Contact Number", "Operating Hours" (two time pickers separated by "TO"), "Transportation Operating Days", "Transportation Address", and "Transportation Location Link" (with a location pin icon). At the bottom, there are two buttons: "Pick or Take Photo" and "Submit". The status bar at the top shows the time as 9:41 and signal strength indicators.

Figure 3.48: Add Transportation Page

Figure 3.48 illustrates the “Add Transportation” page, accessed from the “Transportation” page. On this page, users can add new transportation options available near their selected university. The page offers filters for selecting the transportation type, including bus, taxi, rapid transit, or car rental. Users can enter transportation details such as name, types, operating hours and days, address, location link, and contact information. The added options can then be viewed by future students who are looking for transportation solutions near their campus. This feature promotes sharing and helps future students find relevant transportation options efficiently. Users can also modify or delete their added options if necessary, ensuring that their added options remain accurate.

3.5.6.26 Settings Page

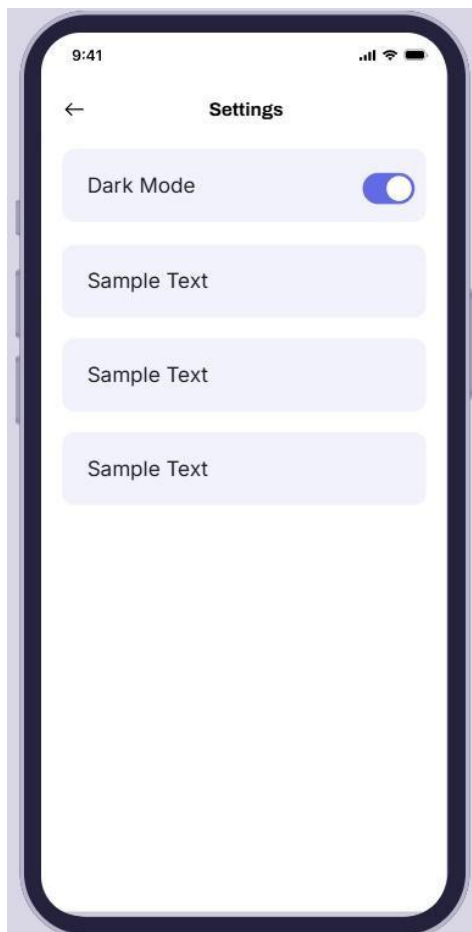


Figure 3.49: Settings Page

Figure 3.49 shows the “Settings” page, which provides various customization options to enhance the user experience. This page includes a toggle switch for enabling or disabling dark mode, allowing users to choose their preferred theme for better accessibility and comfort. Along with the dark mode feature, the “Settings” page offers additional options such as profile settings, and links to the privacy policy, terms of service, and about us pages. These features ensure that users can personalize their experience and access essential application-related information seamlessly.

3.5.6.27 Event Calendar Page

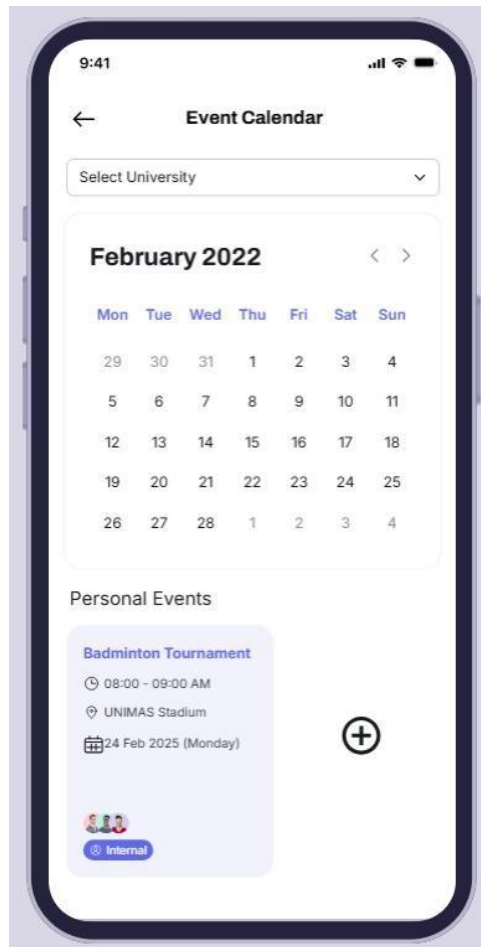


Figure 3.50: Event Calendar Page

Figure 3.50 illustrates the “Event Calendar” page, accessed from the “Home” page. On this page, users can select their desired university from a drop-down list to view the university’s upcoming events displayed on the calendar. Events are marked with a dot under specific dates, and users can click on these dates to view more details about each event. Alongside the university events, users can also see their personal events added to the calendar. These personal events are shown alongside the official university events, allowing users to manage both academic and social

calendars in one place. The page promotes easy tracking and planning for students, ensuring they stay updated on both institutional activities and personal commitments.

3.5.6.28 University Event Page



Figure 3.51: University Event Page

Figure 3.51 illustrates the “University Event” page, accessed from the “Event Calendar” page. On this page, users can click on a specific date marked with a dot to view detailed information about the selected university event. The page provides information such as the event's title, description, location, time, and any additional details like registration links or event organizers. This page helps students stay informed about important university events, activities, and announcements, making it easier for them to plan and participate.

3.5.6.29 Personal Event Page



Figure 3.52: Personal Event Page

Figure 3.52 illustrates the “Personal Event” page, accessed from the “Event Calendar” page. On this page, users can view the details of any personal events they have added to the calendar. This could include events like university prom nights, personal celebrations, or other social gatherings. The page provides users with essential event details such as the event title, description, date, time, and location. Users can also modify or delete their personal events directly from this screen if necessary, ensuring that their calendar remains up-to-date and accurate.

3.5.6.30 Add Personal Event Page

The image shows a mobile application interface for adding a new personal event. The screen is titled "Add New Personal Event". It contains several input fields: "Event Name", "Event Category" (a dropdown menu), "Event Address", "Event Contact Number", "Event Time" (which includes two time pickers separated by "TO"), "Event Date", "Event Location Link" (with a location pin icon), and "Event Description". At the bottom of the form is a blue "Submit" button. The interface is displayed on a smartphone screen with a status bar at the top showing the time as 9:41 and signal/battery icons.

Figure 3.53: Add Personal Event Page

Figure 3.53 illustrates the “Add Personal Event” page, accessed from the “Event Calendar” page . On this page, users can add new personal events to their calendar for easy tracking and organization. The page allows users to input essential event details, such as the event name, category, address, contact information, description, date, time, and location. Users can also set reminders or alerts for these events, ensuring they are notified as the event date approaches. The added personal events will be displayed alongside university events in the calendar, providing a

comprehensive overview of all important activities. This page helps students manage their personal and academic lives more efficiently, ensuring they don't miss any important commitments.

3.6 Summary

Chapter 3 elaborates on the systematic approach taken to develop Unifo+, ensuring that every aspect of the project is aligned with the user's needs and expectations. The iterative prototype cycle, supported by the Agile SCRUM methodology, ensures continuous improvement and stakeholder engagement. Through robust system design, detailed architecture, and effective use case analysis, Unifo+ is set to revolutionize the way students navigate their university environments, providing them with a centralized and user-friendly solution to enhance their academic and social experiences.

CHAPTER 4 : IMPLEMENTATION AND TESTING

4.1 Introduction

This section outlines the implementation, testing, and quality assurance processes of the application. As described in the design phase, the implementation followed three iteration cycles or sprints, with each sprint lasting approximately four weeks. The implementation discussion focuses on the developed features, including classes and databases. Meanwhile, the testing discussion covers various testing methods such as unit testing, integration testing, and user acceptance testing.

4.2 Development Tools

The development of Coding Adventure utilizes a range of modern software tools and services to ensure a smooth and efficient creation process across both Android and iOS platforms. The mobile application is built using React Native with Expo Go, which enables cross-platform mobile development with a unified codebase. Node.js is employed on the backend to handle server-side logic, while MySQL and phpMyAdmin are used for database management tasks such as storing user data and progress.

To incorporate real-world location features into the application, the Open Street Maps (OSM) API is integrated. This allows the application to access location data when necessary, enhancing certain gameplay or user interface experiences. For UI/UX design, Visily is used to design graphical interface elements, icons, and layout prototypes, ensuring an intuitive and visually appealing user interface.

4.2.1 Programming Languages

Several programming languages are used to build and maintain different components of Coding Adventure. JavaScript plays a central role, being used on both the front-end and back-end of the application. On the front-end, JavaScript is used in conjunction with React Native to manage user interactions, render UI components, and handle application logic. On the back-end, JavaScript is used with Node.js to process requests, manage APIs, and interact with the database.

For data storage and retrieval, MySQL is used as the primary database language. It allows structured storage of user accounts, game progress, episode completion status, and other dynamic data. phpMyAdmin supports this by providing a user-friendly interface to manage and interact with the MySQL database. Additionally, Open Street Maps (OSM) API is implemented using JavaScript, enabling location-based services within the application. For design purposes, Visily helps convert design concepts into UI mockups and icons that are later implemented in code.

4.2.2 Visily

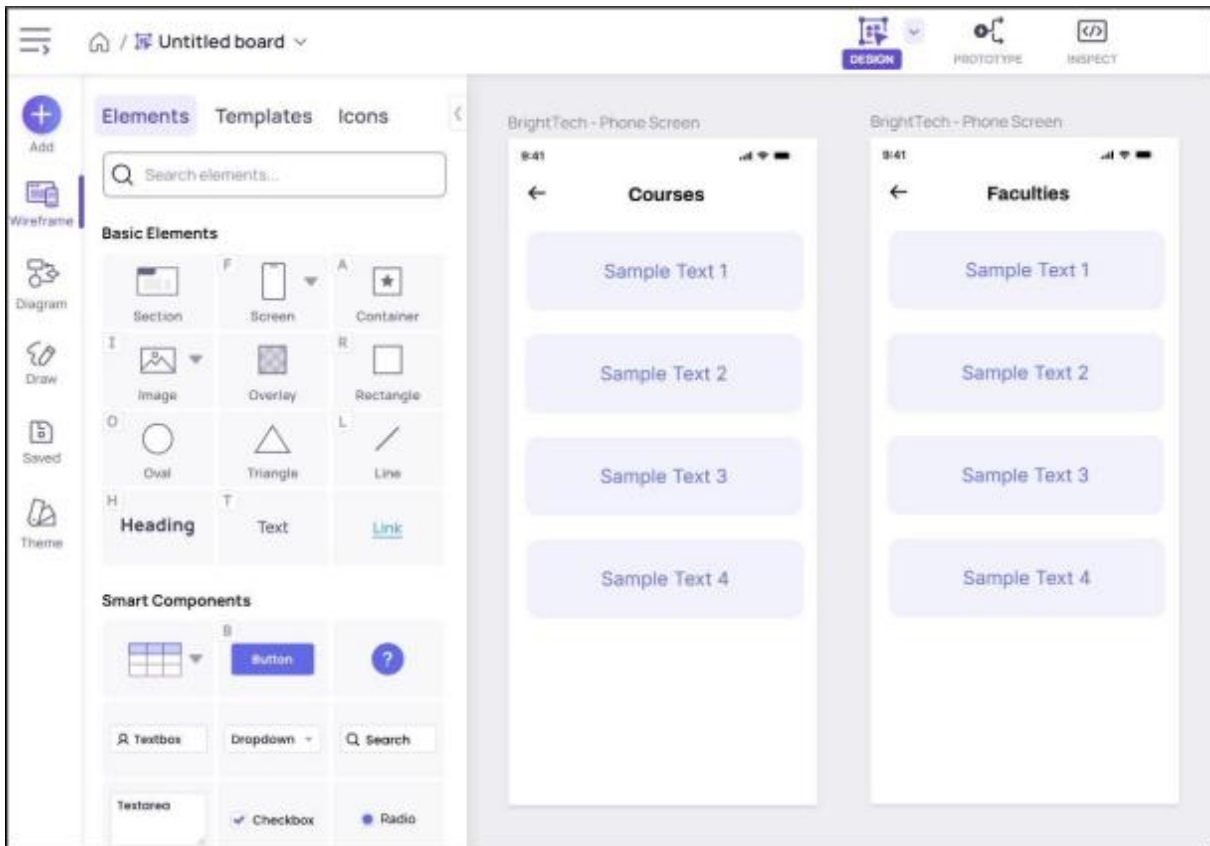


Figure 4.1: Interface of Visily

Figure 4.1 shows the interface of Visily used in the Unifo+ project. Visily is a user- friendly design tool used to create high-fidelity UI prototypes for the Unifo+ application. It enables quick and efficient design of user interfaces without requiring advanced graphic design skills. With Visily, the development team can visually plan and prototype key screens such as the home page, dashboard, university information pages, and navigation flows.

The tool offers pre-designed components like buttons, input fields, icons, and layout templates, which can be easily customized to match the branding and functionality of Unifo+. This allows for consistent visual presentation across all screens while enabling flexibility in design

adjustments. The prototypes created in Visily help bridge the communication between the design and development phases, providing a clear reference for implementing UI elements in the actual application.

4.2.3 Visual Studio Code

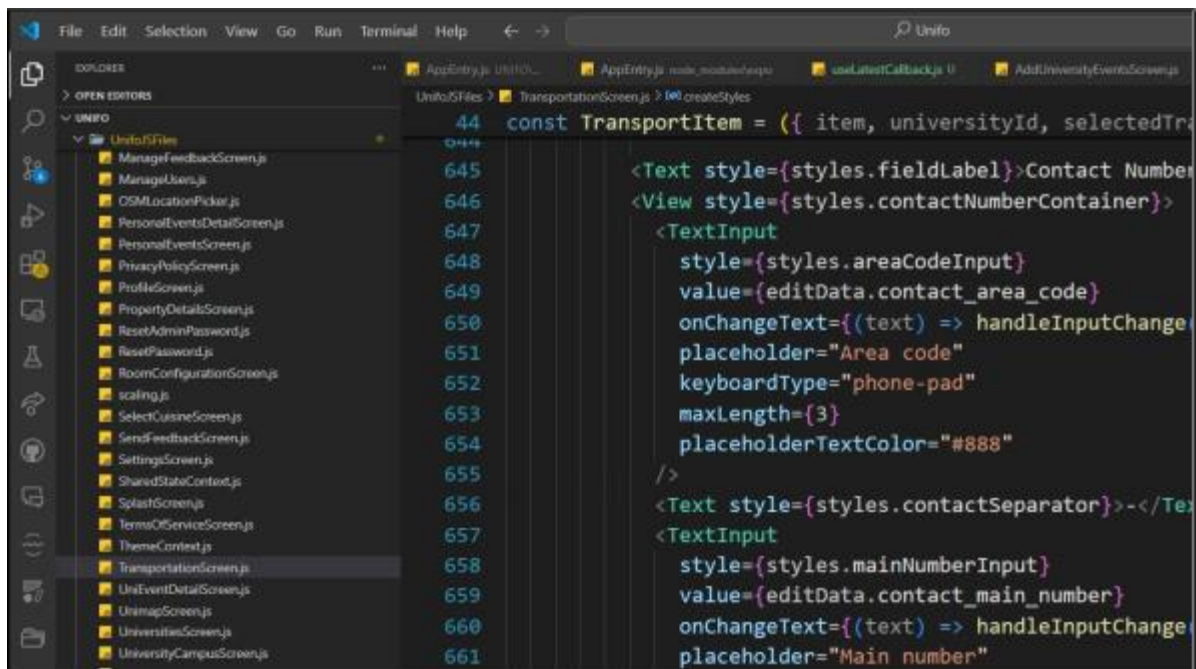


Figure 4.2: Interface of Visual Studio Code

Figure 4.2 shows the interface of Visual Studio Code used in the Unifo+ project. Visual Studio Code is a versatile and free code editor that supports a wide range of programming languages, including JavaScript, Node.js, and MySQL, which are used in the development of Unifo+. It serves as the primary development environment for writing, organizing, and debugging the application's code. The editor's ability to open entire project folders in its explorer panel allows developers to efficiently manage and access files without manually navigating through directories. Its support for extensions, syntax highlighting, and intelligent code suggestions further enhances

productivity during the development process. Figure 4.2 shows the interface of Visual Studio Code used in the Unifo+ project.

4.2.4 Expo Go

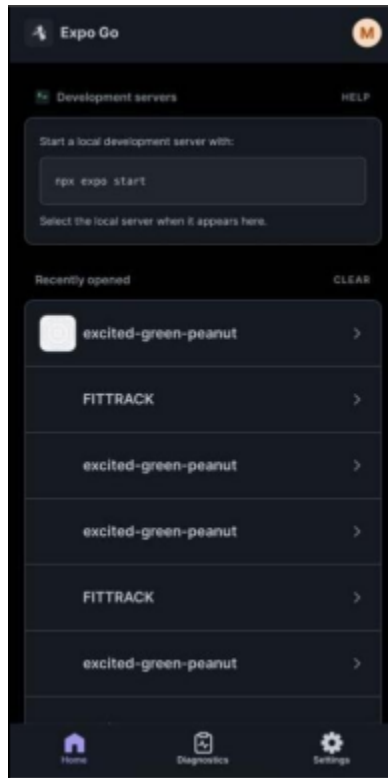


Figure 4.3: Expo Go Application Home Page

Figure 4.3 shows the home page of the Expo Go Application used in the Unifo+ project. Expo Go is a mobile application used to preview and test React Native projects in real time during development. For the Unifo+ project, Expo Go plays a crucial role by allowing developers to instantly view changes made to the code on physical Android or iOS devices without the need for full application compilation or installation. This significantly accelerates the development cycle by providing immediate visual feedback and interaction testing. Using a QR code generated by the

Expo development server, developers can launch the Unifo+ application on their mobile devices through Expo Go and interact with it as if it were a fully installed application. This makes it easier to identify UI issues, verify navigation flow, and test feature functionality across different platforms. Expo Go also simplifies debugging and supports live reloading, making it a valuable tool in maintaining development efficiency throughout the project. Figure 4.3 shows the home page of the Expo Go Application used in the Unifo+ project.

4.2.5 XAMPP

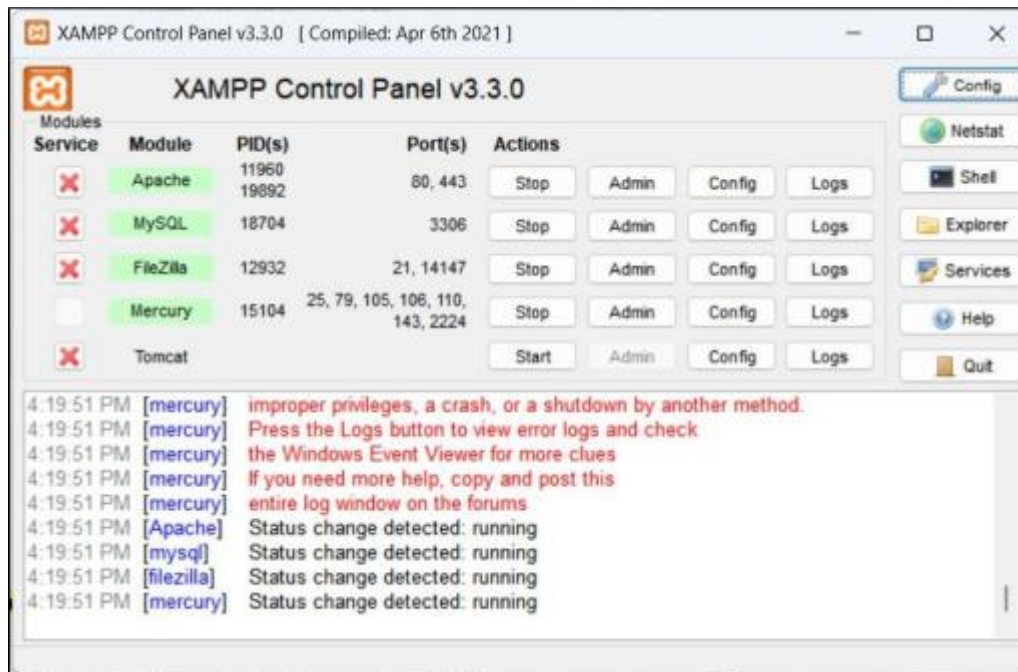


Figure 4.4: XAMPP Control Panel

Figure 4.4 illustrates the XAMPP control panel with Apache and MySQL services running. XAMPP is a widely-used, open-source web server solution that combines essential components such as Apache and MySQL to create a local development environment. It is commonly utilized by developers for testing and deploying web applications locally before moving them to a production server. XAMPP simplifies the setup process by packaging web server services into a

single installation, allowing smooth and efficient development without the need for an active internet connection. By running both Apache and MySQL locally, developers can simulate real server environments and manage databases directly from their machines.

4.1 Application Implementation

This section presents a comprehensive overview of the Unifo+ application implementation, supported by relevant application screenshots that illustrate key functionalities. The implementation is organized into distinct subsections covering both user and administrative interfaces to ensure clarity and comprehensive understanding. The implementation encompasses core user account features, including secure registration, login, and password reset processes that provide reliable access control. The main user interface includes navigation structures, header and menu systems, profile customization options, and interactive features for university exploration, campus facility guides, accommodation listings, and local service directories. Dynamic content modules integrate event calendars, student activity tracking, and user-generated content management for housing, dining, and transportation listings.

The administrative component includes secure administrator authentication, comprehensive user account management, content moderation tools, and approval workflows for listings and events. Dashboard analytics provide platform usage monitoring and performance tracking, while backend controls enable university data updates, notification management, and application configuration oversight. Together, these implementation components demonstrate Unifo+'s user-centered design integrated with robust administrative capabilities, ensuring efficient platform operation, content accuracy, and comprehensive security measures.

4.1.1 Register, Login and Reset Password Screens

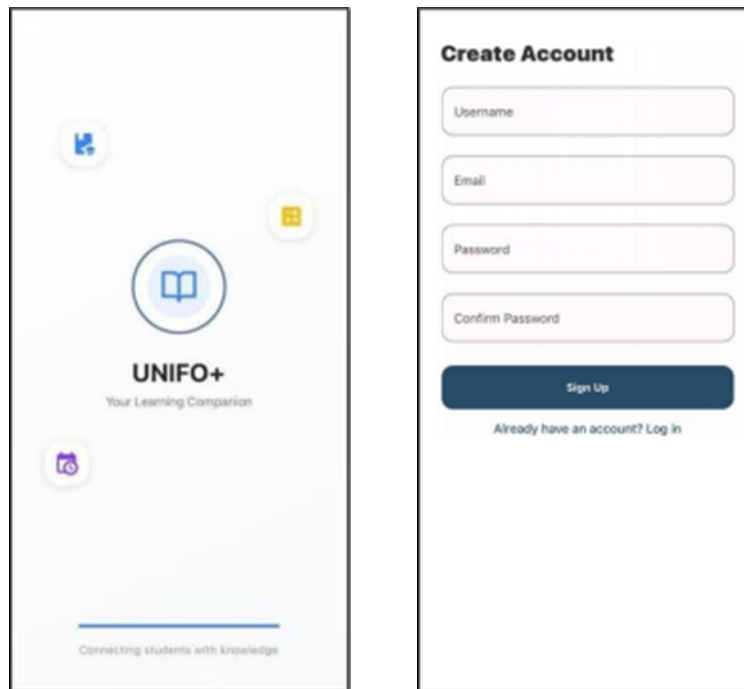


Figure 4.5: Unifo+ Landing Page and Create User Account Page

Figure 4.5 shows the “Landing” page and “Create User Account” page. Upon installing and opening Unifo+ on an iOS device, users will be displayed with a dynamic application landing page. Then, users are required to create a new account by entering a username, email address, and password.

The image displays two mobile application screens side-by-side. The left screen is titled 'Login' and features two input fields labeled 'Email' and 'Password'. Below these fields is a dark blue 'Login' button. Underneath the button are three links: 'Forgot Password?', 'Don't have an account? Sign up', and 'Switch to Admin Login'. The right screen is titled 'Reset Password' and contains three input fields labeled 'Email or Username', 'New Password', and 'Confirm Password'. Below these fields is a dark blue 'Reset Password' button, followed by a 'Back to Login' link.

Figure 4.6: Login Page and Reset Password Page

Figure 4.6 shows the “Login” page and “Reset Password” page. Users can log in using their registered email and password, reset their password via email if they have forgotten it, or change their passwords. If a user forgets their password, they can initiate the "Forgot Password" process, where they are prompted to enter their registered email address. A password reset link or temporary password is then sent to their email, allowing them to create a new password and regain access to their account securely. Administrators access the same dynamic landing page but are presented with a secure login interface. Administrator accounts are pre-configured by the application's super administrator and are distinct from regular user accounts, with elevated permissions for managing the application. Administrators can also use the "Forgot Password" process with password reset links sent to their registered email.

4.1.1 Universities Activity Page



Figure 4.7: Universities Page

Figure 4.7 shows the “Universities” page with the default collapsed view and expanded layout for one of the universities with multiple branches across Malaysia. The “Universities” page is where users can explore a comprehensive collection of Malaysian public universities. At the top of the page, a search bar allows users to quickly search for universities by name, making it easy to find a specific institution. Each university is displayed in a clean, card-style layout, showcasing the university’s name. Some universities, such as Universiti Teknologi MARA (UiTM), include a drop-down icon on their card, indicating that the university has multiple branch campuses across Malaysia. By tapping the drop-down icon, users can expand the card to reveal a list of its branches, allowing them to explore the specific locations and access further details.

Additionally, a bottom navigation bar is available for smooth transitions between different sections of the application, which are the Dashboard, Settings, Home and the Universities section

itself. This “Universities” page acts as a central hub for discovering and accessing detailed information about each public university and its branches, supporting students in making informed choices about their academic journey. Administrators see the exact same layout and functionality as users, with identical search capabilities and expandable branch lists.

4.1.1 University Details Page

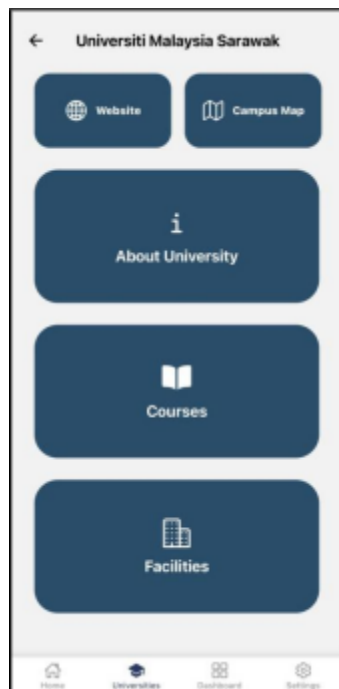


Figure 4.8: University Details Page

Figure 4.8 shows the “University Details” page after selecting a university. The “University Details” page provides a comprehensive overview of the selected university. At the top of the page, users can access quick-action buttons to visit the university's official website or view its campus location via integrated Open Street Map. Users are also presented with navigable buttons such as “About University”, “Courses”, and “Facilities”. These buttons allow users to explore detailed university information, including its background, available faculties and courses, and campus infrastructure. This page centralises all relevant information, ensuring a seamless and informative

browsing experience for the user. Both users and administrators see identical layouts for the “University Details” page.

4.1.1 About University Page



Figure 4.9: About University Page for Users

Figure 4.9 illustrates the “About University” page for users. The “About University” page provides users with a comprehensive overview of the selected university. Accessed from the “University Details” page, this page showcases a variety of key details to give prospective students a clear and informative profile of the university. Displayed on this page are the university images, a detailed university description as well as a city description, introducing the city where the university is located, for example, if the university is in Kuching, this section offers insights into the city's environment, culture, and lifestyle.

Other essential information includes the location name, city, state, country, as well as the university's contact number and email address. This page is designed to help students better understand not only the academic institution but also the surrounding area, supporting more informed and confident decisions in their educational journey.

Kota Samarahan is a calm and green town near Kuching. It's known as an education hub with affordable living. Students can

City
Kuching

State
Sarawak

Country
Malaysia

Contact Information

082581000

corporate@unimas.my

✓ Save Changes × Cancel

Home Universities Dashboard Settings

Figure 4.10: Edit About University Page for Administrators

Figure 4.10 shows the “Edit About University” page for administrators. Administrators initially see the same content as users but have access to an "Edit" button beneath all displayed content. When clicked, the static content transitions into editable input fields, allowing administrators to update university details or images inline.

4.1.1 University Facilities Page



Figure 4.11: University Facilities Page for Users

Figure 4.11 shows the “University Facilities” page for users. The “University Facilities” page features an interactive campus map occupying the top half of the page, displaying all pinpointed icons that represent key facilities of the selected university. By tapping on any pinpoint, users can view the facility name, the distance from their current location, the estimated travel time by driving or walking, and the expected time of arrival. Tapping the same pinpoint a second time will display the route to the facility, providing a clear navigation path directly from the user’s current location.

The bottom half of the page displays a scrollable list of facilities, each containing images of the facility, its name, address, contact number, operating hours, and operating days. Each listing also includes a "View Location" button. By clicking the "View Location" button twice, it will initiate the same route guidance feature, showing both the travel time and estimated arrival

to the selected facility. A “WhatsApp” icon is also located at the top right of each listing, allowing users to easily share the facility information with their friends who may need the location.



Figure 4.12: University Facilities Page for Administrators

Figure 4.12 shows the “University Facilities” page for administrators. Administrators see the same interactive map and facility listings as the users but with additional administrative tools. Each facility listing includes “Edit” and “Delete” buttons for inline editing of details such as names, contact info, operating hours, and images. Administrators also have access to an "Add New Facility" button for inputting new facilities complete with location data, which automatically appear on the map via pinpoint icons.

4.1.1 University Faculties Page

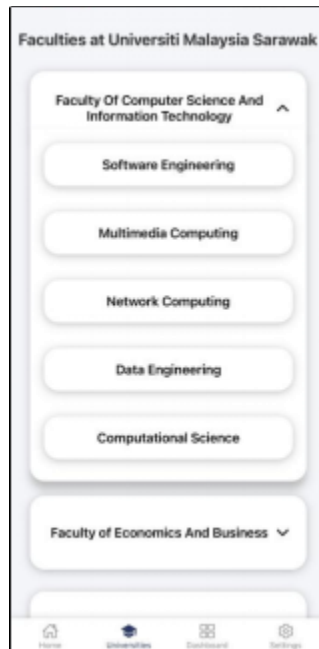


Figure 4.13: University Faculties Page for Users

Figure 4.13 shows the “University Faculties” page for users. After clicking the “Courses” button from the “University Details” page, users are taken to the “University Faculties” page, which presents a scroll-able, card-based layout of all academic faculties associated with the selected university. Each card represents a specific faculty. When a user selects a faculty, the page seamlessly transitions to display a list of academic programs offered under that faculty. Each course is presented with its title and a brief overview, and users can tap on any course to view more detailed information.

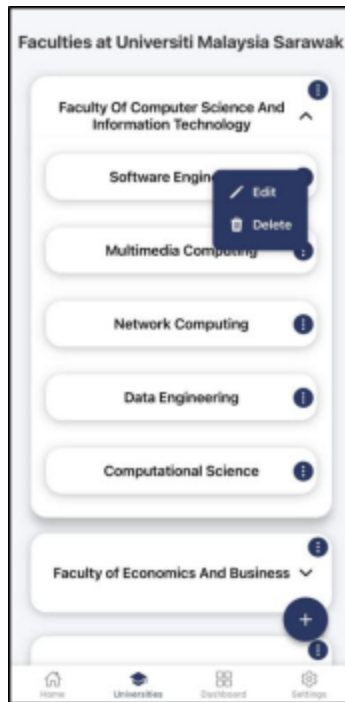


Figure 4.14: University Faculties Page for Administrators

Figure 4.14 illustrates the “University Faculties” page for administrators. Administrators see the same layout as the users but with additional management options via a three-dot menu on each faculty card and course listing. For faculties, this menu allows editing details, deleting faculties (only after removing all courses), or adding new courses. Each course also includes the same menu for editing or deleting. Administrators can add new faculties using a floating add button at the bottom right.

4.1.1 University Course Details Page

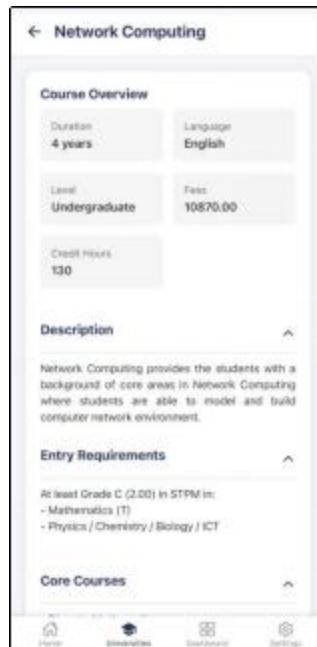


Figure 4.15: University Course Details Page for Users

Figure 4.15 shows the “University Course Details” page for users. The “University Course Details” page provides users with comprehensive information about a specific course to support well-informed educational planning. The details displayed include the course name, duration of study, language of instruction, and course level. Users can also view the total fees, total credit hours required, and potential career paths related to the course.

In addition, the page features a detailed course description and clearly outlines the entry requirements needed for enrollment. It also lists the core courses that are mandatory, along with available elective courses that students can choose from to tailor their learning experience. For further inquiries, the page includes contact information such as the university office’s contact number and official email address, making it easier for prospective students to reach out and obtain additional details directly from the institution.

Edit Course Details

Course Overview

Name	Duration
Network Comput...	4 years
Language	Level
English	Undergraduate
Fees	Credit Hours
10870.00	

Description

Network Computing provides the students with a background of core areas in Network Computing where students are able to model and build computer network environment.

Entry Requirements

At least Grade C (2.00) in STPM in:

- Mathematics (T)
- Physics / Chemistry / Biology / ICT

Core Courses

Save Changes

• System Analysis and Design

• Introduction to Programming

Home Universities Dashboard Settings

Figure 4.16: Edit Course Details Page for Administrators

Figure 4.16 shows the “Edit Course Details” page for administrators. Administrators see the same “University Course Details” page with identical content and layout as the users but with additional “Edit Details” and “Delete Details” buttons at the bottom. The “Edit Details” button navigates to the “Edit Course Details” page where administrators can modify any course information, requiring a "Save Changes" confirmation. The “Delete Details” button allows removal of all course information if necessary.

4.1.1 Dashboard Page



Figure 4.17: University Dashboard Page

Figure 4.17 shows the “University Dashboard” page. The “University Dashboard” page serves as the main interface after a successful user login, providing streamlined access to the application's core functionalities. At the top of the page, a university selection drop- down is available, allowing users to choose a specific public university. Based on the selected university, the content within the dashboard is dynamically updated to reflect relevant data. For instance, if Universiti Malaya is selected, the displayed features such as “Nearby Accommodation”, “Nearby Places To Eat” and “Nearby Grocery Stores”, will be tailored to Universiti Malaya specifically.

Each feature is represented through intuitive buttons arranged in a grid layout. This dynamic structure ensures users receive university-specific content, enhancing the relevance and usefulness of the application. Both users and administrators see identical “University Dashboard” pages serving as the main interface after login.

4.1.1 Nearby Accommodation Page

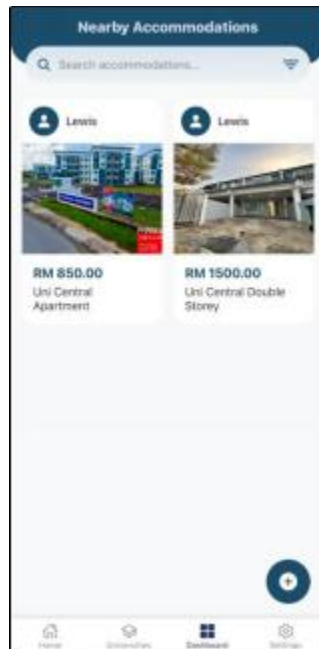


Figure 4.18: Nearby Accommodation Page with Filter Drop-down

Figure 4.18 shows the “Nearby Accommodation” page with filter drop-down. The “Nearby Accommodation” page functions as a centralised interface for users to explore various accommodation listings near their selected university. It incorporates a dynamic search and filter application, allowing users to efficiently locate accommodations that meet their specific needs. At the top of the page, users will find a search bar that enables them to search for accommodations by name. Located on the right side within the same bar as the search feature is a filter button, which when tapped, opens a panel of filtering options.

Users can select the type of accommodation they’re interested in. They can also sort the results by pricing order, choosing to view the cheapest or most expensive listings first. Additional filters include defining a price range, such as RM 500 to RM 1000, and specifying room and bathroom configurations, like "4 rooms, 3 bathrooms" for whole units. The filtering application

also includes an option to reset all applied filters. This allows users to quickly return to the full list of accommodations without manually deselecting each filter. The accommodation results update dynamically based on the selected filters or search query, displaying relevant listings complete with important details such as pricing, name and preview images.

Additionally, users can easily add new accommodation listings by tapping on the floating “Add” button located at the bottom right corner of the page. This feature encourages students, or local residents to share accommodations they know of, helping expand the options available to others. Administrators see the same layout, features, and filters as users, except without the "Add" button. This ensures consistent user experience while maintaining content control through administrative channels.

4.1.1 Property Details Page

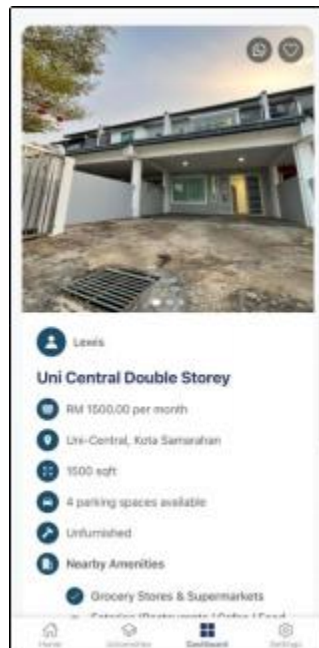


Figure 4.19: Property Details Page for Users

Figure 4.19 shows the “Property Details” page for users. The “Property Details” page is accessed by selecting a property from the Accommodation list and provides a comprehensive overview of the selected listing. The page displays key details including the property image, property name, property address, rental price per month, property size, and number of available parking spaces. Additional information such as the condition of the unit, whether it is fully furnished, partially furnished, or unfurnished, is clearly indicated, along with a list of nearby amenities. The owner's contact number is also provided for direct communication regarding inquiries.

A “View Location” button is available for users to check the exact location of the accommodation. Upon clicking the button, users can see the property’s position on the map in relation to their current location as well as the route guidance, along with estimated travel times both by car and on foot. Each property’s details page also features a “Love” icon and a “WhatsApp” icon positioned side by side at the top right corner, allowing users to mark the listing as a favorite for easy access later or share the accommodation details with friends who may need them respectively.

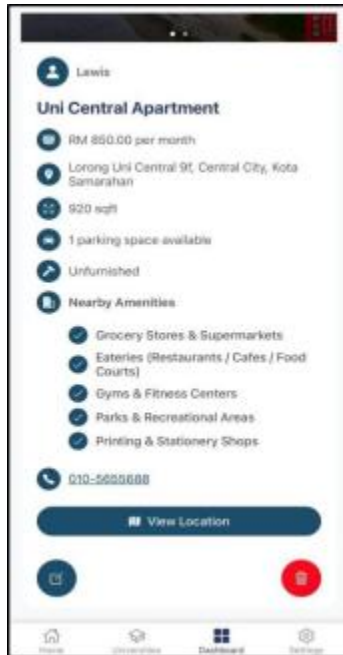


Figure 4.20: Property Details Page for Administrators

Figure 4.20 shows the “Property Details” page for administrators. Administrators see the same layout and details as the users but with additional "Edit" and "Delete" buttons for content management through inline editing. Unlike users, administrators cannot access “Add Accommodation”, "Love" (favorite), or WhatsApp sharing features, focusing their role on moderation and content integrity.

4.1.1 Add Accommodation Page

The figure displays two mobile application screens for adding a new property. The left screen, titled 'Add New Property', shows the 'Basic Information' section with input fields for 'Property Address', 'Price (RM)' (e.g., 1500), 'Size (sqft)' (e.g., 800), 'Parking Spaces' (Number of parking spaces), and 'Contact Number' (e.g., (012) 12345678). Below these is the 'Condition & Amenities' section and a 'Location' dropdown. A 'Submit Property' button is at the bottom. The right screen shows the 'Property Details' section, specifically the 'Condition & Amenities' part. It includes a 'Property Condition' section with three checkboxes: 'Fully furnished', 'Partially furnished', and 'Unfurnished'. Below this is a 'Nearby Amenities' section with a list of checkboxes for various facilities: 'Grocery Stores & Supermarkets', 'Eateries (Restaurants / Cafes / Food Courts)', 'Public Transportation (Bus / MRT / LRT / Ride-Sharing)', 'Healthcare Facilities (Clinics / Pharmacies / Hospitals)', 'Gyms & Fitness Centers', 'Parks & Recreational Areas', 'Printing & Stationery Shops', 'Entertainment (Cinemas/Gaming Cafes)', and 'Shopping Malls'. At the bottom of this section is an 'Other Amenities' input field with the placeholder text 'Add any other nearby amenities...'. Both screens have a bottom navigation bar with icons for Home, University, Dashboard, and Profile.

Figure 4.21: Add Accommodation Page

Figure 4.21 shows the “Add Accommodation” page. The “Add Accommodation” page allows users to contribute new property listings to assist future students by providing detailed and accurate information. Users can upload up to a maximum of 9 property images to give a clear visual representation of the accommodation. Input fields include the property name, property address, rental price per month, and property size in square feet. Users can specify the number of available parking spaces, which is particularly useful for local students who drive to campus. The condition of the property, whether fully furnished, partially furnished, or unfurnished, can be selected using check-boxes, with only one option allowed to ensure clarity. Nearby amenities can be selected by ticking multiple check-boxes from a provided list, and there is also an “Other Amenities” input field available for users to enter any additional features not included in the predefined list.

The form also includes a field for the owner's contact number to allow direct communication with the property owner. For location details, a location picker is available where users can either search for a specific destination or pinpoint the exact location by clicking on the map. To enhance ease of use, the map supports multiple view types, including satellite maps, enabling users to more easily identify and select the correct location if they are already familiar with the area visually.

4.1.1 Grocery Stores Page



Figure 4.22: Nearby Grocery Stores Page for Users

Figure 4.22 shows the “Nearby Grocery Stores” page for users. The “Nearby Grocery Stores” page provides users with an easy way to browse nearby grocery stores around their selected university. This page is especially useful for students who are new to the area and need quick access to daily necessities. A search bar placed at the top of the page allows users to quickly locate specific stores by name or keyword. The listing displays essential information for each grocery store,

including store images, the store name, address, opening hours, and operating days. Each listing also includes a "View Location" button, which, when clicked, shows the grocery store's location from the user's current position along with the distance and estimated travel time by both driving and walking. To enhance ease of use, the map supports multiple view types, including satellite maps, enabling users to more easily identify the location of the grocery store.

The page also includes a floating "Add" button at the bottom right corner, allowing users to contribute new grocery store listings. Each grocery store listing also features a "Love" icon and a "WhatsApp" icon positioned side by side at the top right corner, allowing users to mark the listing as a favorite for easy access later or share the grocery store details with others respectively.

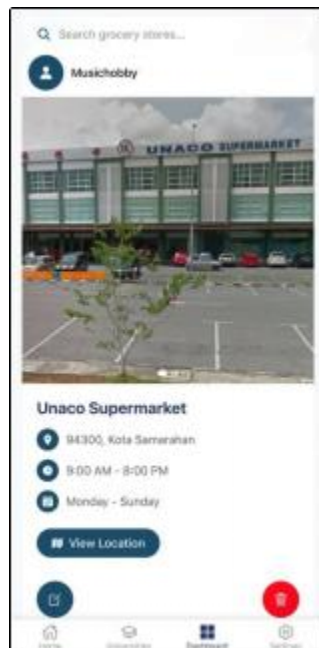


Figure 4.23: Nearby Grocery Stores Page for Administrators

Figure 4.23 shows the "Nearby Grocery Stores" page for Administrators. Administrators view the same layout and features as the users but with additional "Edit" and "Delete" buttons

within each listing for content management through inline editing. Social interaction features (Add, Love, WhatsApp) are disabled for administrators.

4.1.1 Add Grocery Store Page

Figure 4.24: Add Grocery Store Page

Figure 4.24 shows the “Add Grocery Store” page. The “Add Grocery Store” page enables users to contribute new store entries for the benefit of future students by submitting detailed and relevant information. Input fields include grocery store images (with a maximum of 9 images), grocery store name, store address, operating days (e.g., Mon– Fri), and operating hours. The operating hours are selected using a time picker, which restricts minute selection to either 00 or 30 minutes to maintain standardization and avoid confusion.

Additionally, a location picker is provided, allowing users to either search for a destination, such as specific grocery stores or areas if they know the location, or pinpoint the exact location manually by tapping on the map. To assist users who are already familiar with a location visually,

the map also supports several map types, including satellite maps, making the location selection process more intuitive.

4.1.1 Nearby Eateries Page



Figure 4.25: Nearby Eateries Page for Users

Figure 4.25 shows the “Nearby Eateries” page for users. The “Nearby Eateries” page is designed to help users explore nearby food outlets, cafes, and restaurants around their selected university. At the top of the page, a search bar allows users to quickly search for specific eateries by name or keyword. Located conveniently at the right end of the search bar is a filter button that opens up a range of sorting and filtering options, such as by food type, price range, popularity, or distance. These filters help users find eateries that best match their preferences, whether they’re looking for affordable meals, specific cuisines, or popular student hangouts. Each listing displays key details such as the eatery’s images, name, address, contact number, opening hours, operating days, and star rating. To enhance navigation, a "View Location" button is provided for each listing.

By clicking this button, users can see the location of the eatery from their current location along with the distance and estimated travel time by car and walking. To further enhance usability, the integrated map supports multiple view types, including satellite maps. This allows users who are already visually familiar with the area to more easily identify and select the correct location.

In addition, the page includes a floating "Add" button at the bottom right corner, allowing users to contribute new eatery listings. This feature encourages students and locals to share food spots they recommend, building a richer and more diverse collection of dining options. Each eatery listing also features a "Love" icon and a "WhatsApp" icon positioned side by side at the top right corner, allowing users to mark the listing as a favorite for easy access later or share the eatery details with others respectively.



Figure 4.26: Nearby Eateries Page for Administrators

Figure 4.26 shows the "Nearby Eateries" page for administrators. Administrators see identical layout and data as the users but with added "Edit" and "Delete" buttons for content

moderation through inline editing. Administrators cannot access “Add Eatery”, “Love”, or WhatsApp sharing features.

4.1.1 Add Eateries Page

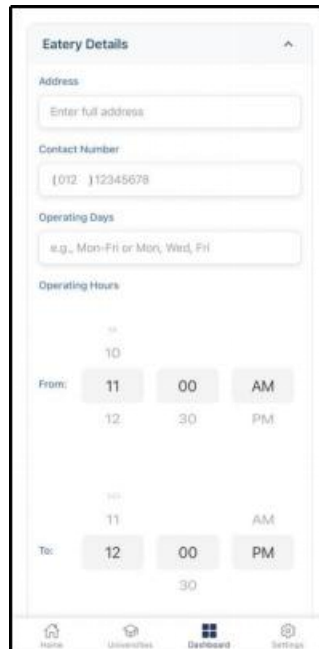
The image is a screenshot of a mobile application interface titled "Eatery Details". It contains several input fields: "Address" with a placeholder "Enter full address", "Contact Number" with a placeholder "(012) 12345678", and "Operating Days" with a placeholder "e.g., Mon-Fri or Mon, Wed, Fri". Below these is the "Operating Hours" section, which includes a "From:" time picker set to 11:00 AM and a "To:" time picker set to 12:00 PM. The time pickers allow selection of hours (10, 11, 12) and minutes (00, :30) in either AM or PM. At the bottom of the screen is a navigation bar with four icons: Home, University, Dashboard, and Settings.

Figure 4.27: Add Eateries Page

Figure 4.27 shows the “Add Eateries” page. Users can use this page to contribute new eatery listings for others to explore around their selected university. The form includes a cuisine type selector (e.g., Asian, Western, Vegetarian, Italian) to categorize the eatery. Users can upload up to 9 eatery images and provide essential details such as the eatery’s name, address, contact number, operating days (e.g., Mon–Fri), and operating hours. The time picker is designed for clarity, allowing users to select times in 30-minute intervals only, either at :00 or :30, reducing ambiguity.

A star rating feature is also included, where users can rank the eatery by clicking on the star index, for instance, tapping the fourth star fills up to four stars, indicating a 4-star rating. To accurately pinpoint the eatery’s location, users can use the integrated location picker, either by

typing a destination (e.g., a known apartment or area) or by directly clicking on the map. Multiple map views, including satellite maps, are supported to help users more easily identify the location if they are already visually familiar with the area.

4.1.1 Student Activity Places Page

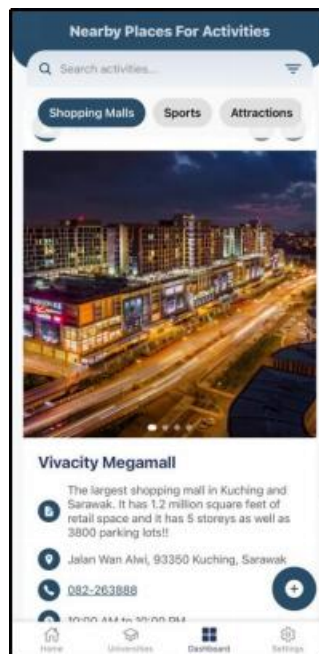


Figure 4.28: Student Activity Places Page for Users

Figure 4.28 illustrates the “Student Activity Places” page for users. The “Student Activity Places” page is a lively and engaging section of the application designed to help users discover various student-related places for entertainment and activities such as local attractions, shopping, sports, and volunteering opportunities. Positioned at the top of the page is a search bar that allows users to quickly find specific events or activity places by name or keyword. Adjacent to the search bar is a filter button that enables users to refine their search results based on criteria such as activity type, date, popularity, or relevance, making it easier to find activities that match their interests or fit their schedules.

Each listing on the page displays important details including the activity place's images, name, description, address, contact number, opening hours, and operating days. For added convenience, a “View Location” button is available with every listing. By tapping this button, users can see the activity place’s location relative to their current position, along with the distance, estimated travel time by car and on foot, and expected time of arrival.

To enhance the navigation experience, the built-in map supports multiple view types, including satellite maps. This allows users who are visually familiar with certain areas to identify and select the correct location more easily. Additionally, a floating “Add” button located at the bottom right corner of the page empowers users to contribute their own events or activity places, helping to cultivate a vibrant and interactive student life within the application. Each activity place listing also features a “Love” icon and a “WhatsApp” icon positioned side by side at the top right corner, allowing users to mark the listing as a favorite for easy access later or share the activity place details with others respectively.



Figure 4.29: Nearby Activity Places Page for Administrators

Figure 4.29 illustrates the “Nearby Activity Places” page for administrators. Administrators see identical layout and data as the users but with added edit and delete buttons for content moderation through inline editing. Administrators cannot access “Add Eatery”, “Love”, or WhatsApp sharing features.

4.1.1 Add Activity Places Page

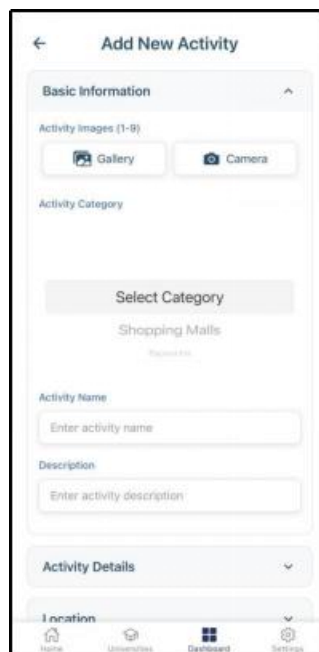


Figure 4.30: Add Activity Places Page

Figure 4.30 illustrates the “Add Activity Places” page. This page allows students to contribute new activity locations that can benefit others in their university community. Users can select an activity category such as shopping malls, sports, or attractions to classify the submission. They can upload up to 9 images showcasing the activity place and provide essential details including the activity place name, address, contact number, operating days (e.g., Mon–Fri), and

operating hours. The time picker simplifies input by only allowing selection of times in 30-minute intervals (e.g., 9:00, 9:30), ensuring clarity and consistency.

To accurately mark the location, a built-in location picker is provided. Users can either search for a destination (like a specific apartment or area) or directly tap on the map to pin the activity spot. The map supports multiple view types, including satellite maps, to make it easier for users who are visually familiar with the area to confirm the correct location.

4.1.1 Nearby Transportation Page



Figure 4.31: Nearby Transportation Page for Users

Figure 4.31 shows the “Nearby Transportation” page for users. The “Nearby Transportation” page is designed to help users explore a variety of transportation options available near their selected university. These options may include car rentals, bus services, or public transportation. At the top of the page is a search bar that allows users to look up specific transportation services

or keywords. Located to the right within the search bar is a filter button, which enables users to sort listings based on transportation type.

Each listing provides detailed information including transportation service images, name, address, contact number, operating hours, and operating days. To further enhance convenience, every listing includes a “View Location” button. When tapped, this button displays the service’s location in relation to the user’s current position, along with the distance, estimated travel time by driving or walking, and time of arrival, helping users choose the most efficient travel option.

The page also supports multiple map view types, including satellite maps, allowing users who are visually familiar with the area to pinpoint exact locations more easily. Additionally, a floating “Add” button at the bottom right corner encourages users to contribute new transportation listings or carpool posts, fostering a more connected and resourceful student community. Each transportation listing also features a “Love” icon and a “WhatsApp” icon positioned side by side at the top right corner, allowing users to mark the listing as a favorite for easy access later or share the transportation details with others.



Figure 4.32: Nearby Transportation Page for Administrators

Figure 4.32 illustrates the “Nearby Transportation” page for administrators. Administrators see identical layout as the users with additional edit and delete buttons for content management through inline editing. Administrators cannot access “Add Eatery”, “Love”, or WhatsApp sharing features.

4.1.1 Add Transportation Page

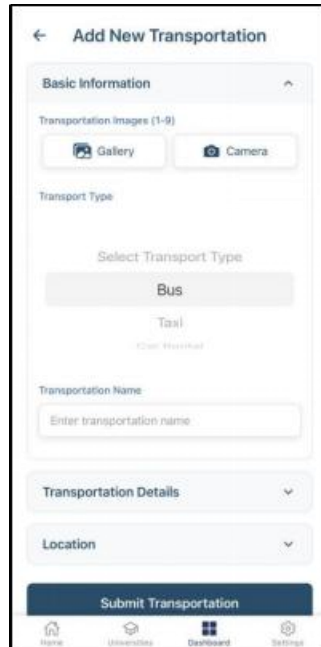
The screenshot shows a mobile application interface for adding new transportation. At the top, there is a back arrow and the title "Add New Transportation". Below this is a section titled "Basic Information" with a chevron icon. Under "Basic Information", there is a label "Transportation Images (1-9)" followed by two buttons: "Gallery" and "Camera". Below these is a label "Transport Type" and a section titled "Select Transport Type" containing three buttons: "Bus", "Taxi", and "Rapid Transit". Below the transport type section is a label "Transportation Name" and a text input field with the placeholder "Enter transportation name". Below the name field is a section titled "Transportation Details" with a chevron icon. Under "Transportation Details", there is a label "Location" and a dropdown menu. At the bottom of the form is a large blue button labeled "Submit Transportation". At the very bottom of the screen is a navigation bar with four icons: Home, Universities, Dashboard, and Settings.

Figure 4.33: Add Transportation Page

Figure 4.33 shows the “Add Transportation” page. The “Add Transportation” page enables users to contribute new transportation service listings for their selected university. Users begin by selecting the transport type, such as bus, taxi, or rapid transit, to help categorize and organize the listings. They can upload up to 9 transportation images and are required to fill in key details including the transportation name, address, and contact number.

For operating hours, users select from a time picker designed to minimize confusion, only allowing minute intervals of 00 or 30 (e.g., 9:00, 9:30). This ensures consistency across all entries. To accurately mark the location, a location picker is provided where users can either search for a specific destination (such as known apartments or areas) or manually pin the location on the map. To further assist with accuracy, the map supports multiple viewing types, including satellite maps,

making it easier for users to select the correct location, especially if they're already familiar with the area visually.

4.3.20 Open Street Maps Page



Figure 4.34: Open Street Map Page

Figure 4.34 displays the “Open Street Map” page in the Unifo+ application. Users can access this page by clicking on the “View Location” button on any of the listings. This page plays a crucial role in helping users navigate campus grounds and discover nearby amenities using an interactive map powered by “OpenStreetMap”. In Figure 4.34 above, the grocery store location is used as an example to demonstrate how the map functions in the page. The “Open Street Map” page also provides accurate location data, estimated travel times, and route guidance for walking or driving. The same “Open Street Maps” page is used for all nearby amenities, including accommodations, eateries, transport options, and activity places, only the displayed locations differ

based on the selected category and it's listings. Designed to support students who are unfamiliar with their new environment, this page makes it easier to find daily necessities without the need for multiple applications or manual searches.

4.3.21 Event Calendar Page



Figure 4.35: Event Calendar Page

Figure 4.35 shows the “Event Calendar” page. The “Event Calendar” page is designed to help users manage both university and personal events in one interface. At the top of the page, a university selection drop-down, allows users to choose the university they are currently enrolled in or interested in viewing. Upon selecting a university, the calendar will dynamically update to display only that university’s official events, ensuring event listings are relevant and free from conflicts. For instance, if there is a UNIMAS event on April 20, it will only be shown when UNIMAS is selected in the drop-down.

Just below the drop-down is the calendar itself, which remains consistently visible regardless of the selected university. Dots will appear on specific dates to indicate the presence of events. These dots change based on the university selected, allowing users to instantly identify which days have scheduled university activities.

Beneath the calendar is the personal events section where users can manage and view their own custom events. This layered view ensures that students can easily plan their schedules and be well organized within a single page. Administrators see only the university drop-down and calendar with official events. They do not have access to the personal events section, which is exclusive to normal users.

4.3.22 Events Page

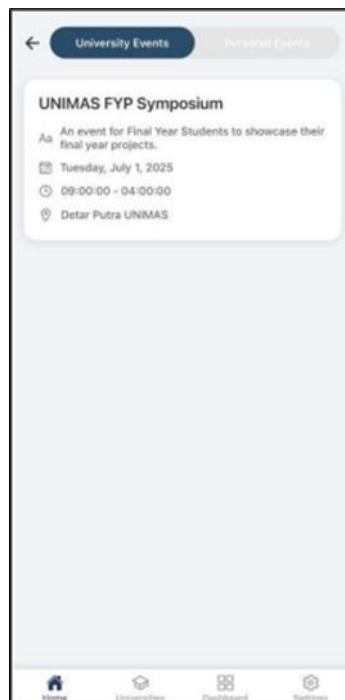


Figure 4.36: University Event Section in Event Page

Figure 4.36 shows the “University Event” section in the “Event” page. The “Events” page is accessed by tapping on a date marked with a dot in the Event Calendar, indicating that university event(s) are scheduled for that day. Upon entering this page, users will find a toggle switch at the top, allowing them to seamlessly switch between viewing “University Events” and “Personal Events”. When the toggle is set to “University Events”, the page displays a series of event cards tailored to the selected university. Each card provides key event details in a clean and structured layout, including the event name, a concise description, the full date (e.g., Monday, March 31, 2025), the duration (e.g., 10:15 AM – 06:30 PM), and the event address. This enables students to stay informed about a diverse range of campus activities. The interface is not only functional but also fosters student engagement by making it easier for them to discover and keep track of campus-related opportunities.

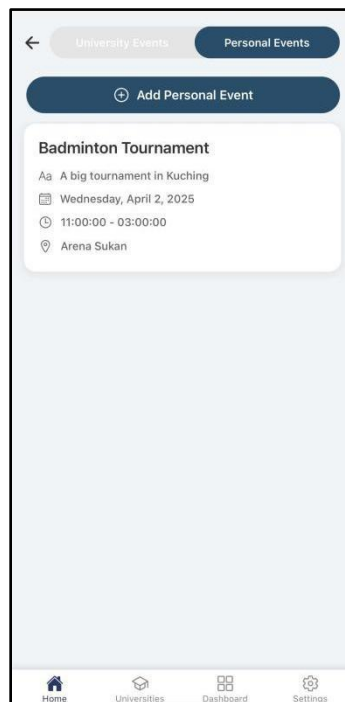


Figure 4.37: Personal Event Section in Event Page

Figure 4.37 shows the “Personal Event” section in the “Event” page. When the toggle is switched to “Personal Events”, users are shown events that they have created, independent of any university selection. These personal events are also displayed in a card format consistent with the university events, maintaining a cohesive user experience. The layout of each card includes the event name, a short description, the event date, time range, and location. Additionally, a prominent “Add Personal Event” button is situated below the toggle switch, empowering users to create and manage their own schedule conveniently. Once a personal event is created, the event will be displayed both on this page and in the Event Calendar page, listed below the calendar interface. Notably, if no university is selected in the calendar interface, only personal events will be shown, further emphasizing the personalized nature of this section.

For administrators, the interface slightly differs to cater to their specific responsibilities. Administrators do not have access to the Personal Events section and thus do not see the toggle switch. Instead, their view is dedicated solely to University Events. In this version of the page, an “Add University Event” button is made available, enabling administrators to input and manage university-wide events for each selected public university.

4.3.23 Personal Event Details Page

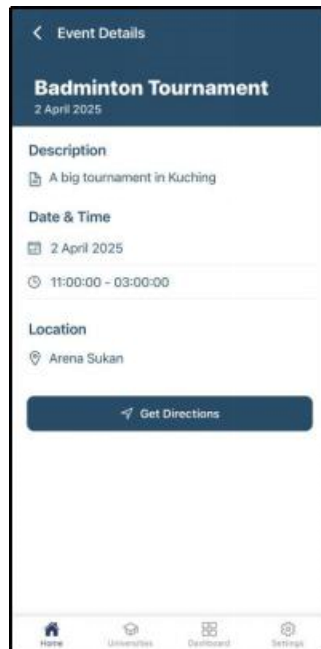


Figure 4.38: Personal Event Details Page

Figure 4.38 shows the “Personal Event Details” page. Similar to the University Event card, by tapping on a Personal Event card, users will be navigated to the “Personal Event Details” page, where they can view the event information more clearly and in greater detail. This page mirrors the layout of the “University Event Details” page, providing a comprehensive overview of the personal event, including its title, date, time, duration, location, and description. A key feature on this page is the “View Event Location” button. When tapped, it opens an interactive map that displays the event’s location relative to the user’s current position. The map also provides real-time distance and estimated travel time by car and on foot, which is especially helpful for events located off-campus.

To further enhance user experience, the map includes multiple view types, such as satellite view, making it easier for users to recognise the destination visually, particularly if they are already

familiar with the area. This navigation tool ensures users have all the necessary information to plan and reach their personal events efficiently and confidently. Users also have the ability to edit or delete their personal events directly from this page, ensuring easy event management and helping them maintain an accurate, customized schedule. The intuitive interface is designed to support students in organizing their academic and personal lives all in one place.

4.3.24 Add Personal Event Page

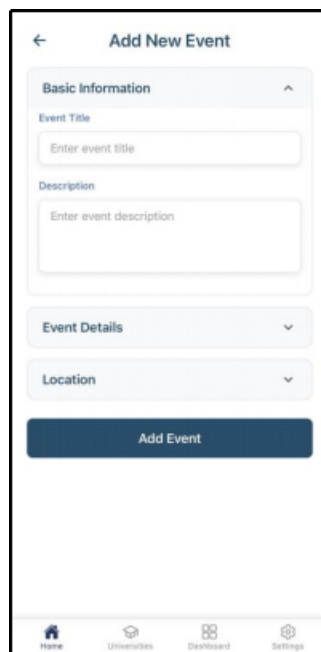
The image is a screenshot of a mobile application interface for adding a new event. At the top, there is a header bar with a back arrow on the left and the title 'Add New Event' in the center. Below the header, the form is organized into sections. The first section is 'Basic Information', which contains two input fields: 'Event Title' and 'Description'. The second section is 'Event Details', which contains a 'Location' input field. At the bottom of the form is a large, dark blue button labeled 'Add Event'. The bottom of the screen features a navigation bar with four icons: a home icon, a calendar icon, a location pin icon, and a settings icon.

Figure 4.39: Add Personal Event Page

Figure 4.39 shows the “Add Personal Event” page. The “Add Personal Event” page enables users to create and add custom personal events to their calendar. The form includes input fields for the event title, event description, event date, event address, start time, and end time, both selected using a time picker. Additionally, the page features a location picker, allowing users to either search for a specific destination or pinpoint the location directly on an interactive map. The map supports multiple view modes, including satellite view, making it easier for users to visually identify their

intended location. This feature allows students to organize their personal schedules, such as social gatherings, club meetings, or task deadlines, alongside university events within the same calendar interface. Once submitted, the personal events are seamlessly integrated into the calendar, providing a unified and convenient overview of all upcoming responsibilities.

4.3.25 Settings Page

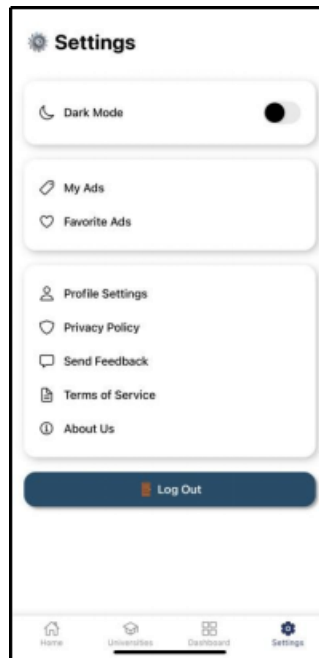


Figure 4.40: Settings Page for Users

Figure 4.40 shows the “Settings” page for users. The “Settings” page offers several user-centered customization and account management options to enhance the overall application experience. At the top, a Dark Mode toggle switch allows users to switch between light and dark themes according to their visual preferences and comfort. This page also includes quick access to key features such as “My Ads” and “Favorite Ads”, enabling users to manage their posted listings and saved content with ease. In the “Profile Settings” section, users can update their personal information, including username, email address, and password for account security and

personalisation. Additional options on this page include “Send Feedback”, “Terms of Service”, and “ About Us” pages, allowing users to provide input and learn more about the application's policies and background. At the bottom of the page, a prominent “Log Out” button is available for users who wish to securely sign out of their account. Overall, the “Settings” page ensures users have full control over their experience and account within the application.

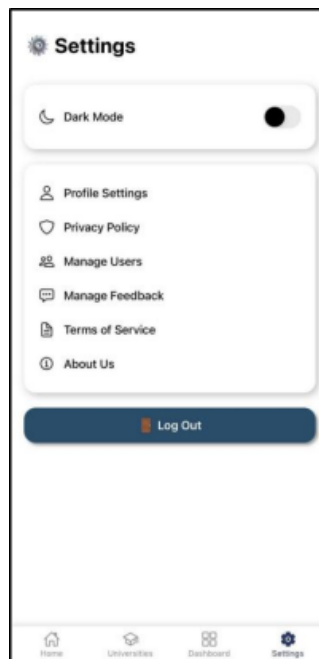


Figure 4.41: Settings Page for Administrators

Figure 4.41 shows the “Settings” page for administrators. For administrators, the layout excludes the “My Ads” and “Favorite Ads” pages, even though they can add listings elsewhere. Instead, administrators have access to manage user feedback, they can review, resolve, and delete feedback once the issues are addressed, ensuring smooth communication and application quality control. The administrators also a dedicated “Manage Users” page, which displays a list of registered users and provides the ability to review and delete accounts that repeatedly violate community guidelines.

4.1 Database on Server

The Unifo+ server utilizes a PHP MySQL database to store and manage all application data, ensuring seamless functionality across various screens. Each database table is carefully structured to support specific features, enabling efficient data retrieval, synchronization, and user interaction. Below is a detailed breakdown of each table and its role in the application's operation.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 UserID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 Username	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
☐	3 UserEmail	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	4 Password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐	5 CreatedAt	timestamp			No	current_timestamp()			Change Drop More
☐	6 Role	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Figure 4.42: Users table

Figure 4.42 shows the Users table. The Users table serves as the foundation for user authentication and profile management within Unifo+. It stores essential details such as UserID, Username, UserEmail, Password, CreatedAt, and Role. When a new user registers, their credentials are securely inserted into this table, and the server validates their login attempts by cross-referencing the provided email and password. The Role column distinguishes between regular users and administrators, ensuring appropriate access levels. This table is crucial for maintaining secure login sessions and managing user-specific data throughout the application.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	AdminID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	AdminName	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	AdminEmail	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	Password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	Role	varchar(20)	utf8mb4_general_ci		Yes	Admin			Change Drop More
6	CreatedAt	timestamp			No	current_timestamp()			Change Drop More

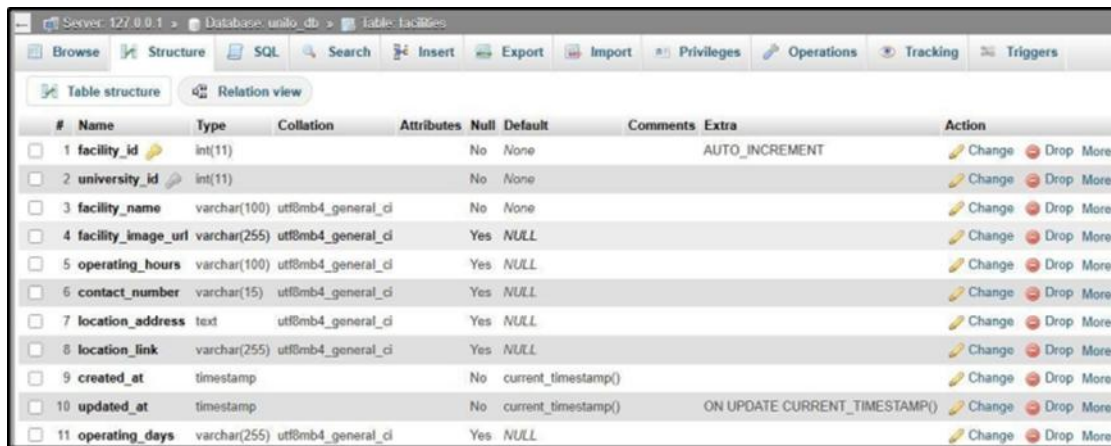
Figure 4.43: Admins table

Figure 4.43 shows the Admins table. The Admins table is dedicated to storing administrative account information, including AdminID, AdminName, AdminEmail, Password, Role, and CreatedAt. This table ensures that only authorized personnel can access backend management functions, such as content moderation and application configurations. By separating administrator accounts from regular user accounts, the application maintains a secure and organized administrative framework.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	university_id	int(11)			Yes	NULL			Change Drop More
3	location_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	city	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
5	state	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
6	country	varchar(100)	utf8mb4_general_ci		Yes	Malaysia			Change Drop More
7	latitude	decimal(10,8)			Yes	NULL			Change Drop More
8	longitude	decimal(11,8)			Yes	NULL			Change Drop More
9	university_description	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
10	city_description	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
11	images	longtext	utf8mb4_bin		Yes	NULL			Change Drop More
12	website	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
13	contact_number	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More
14	email	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Figure 4.44: About_university table

Figure 4.44 shows the About_university table. This table provides comprehensive details about universities, including id, university_id, location_name, city, state, country, university_description, city_description, images, contact_number, and email. It acts as the primary data source for the About University page, where users explore university profiles, rankings, and campus descriptions. The images column stores visual references, enhancing the user experience by displaying campus photos and key landmarks.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1 facility_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2 university_id	int(11)			No	None			Change Drop More
☐	3 facility_name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐	4 facility_image_url	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	5 operating_hours	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	6 contact_number	varchar(15)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	7 location_address	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	8 location_link	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	9 created_at	timestamp			No	current_timestamp()			Change Drop More
☐	10 updated_at	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More
☐	11 operating_days	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Figure 4.45: Facilities table

Figure 4.45 shows the Facilities table. The Facilities table holds detailed information about campus amenities, including facility_id, university_id, facility_name, facility_image_url, operating_hours, contact_number, location_address, location_link, created_at, updated_at, and operating_days. This data powers the University Facilities page, where users can view facility descriptions, operating schedules, and contact details. The facility_image_url column references images stored externally to optimize database performance.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	university_id	int(11)			Yes	NULL			Change Drop More
4	campus_id	int(11)			Yes	NULL			Change Drop More

Figure 4.46: Faculties table

Figure 4.46 shows the Faculties table. The Faculties table (id, name, university_id, campus_id) lists all the faculties within the public university selected by the user. This table supports the University Faculties page.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	description	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	faculty_id	int(11)			Yes	NULL			Change Drop More

Figure 4.47: Courses table

Figure 4.47 shows the Courses table. The Courses table (id, name, description, faculty_id) contains the courses within a specific faculty of the public university the user selected. This table supports the University Courses page, enabling students to explore available degrees courses and specialization.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	description	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
2	course_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
3	course_duration	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	course_level	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
5	course_language	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
6	course_entry	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
7	course_career	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
8	course_fees	decimal(10,2)			Yes	NULL			Change Drop More
9	course_contact	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
10	core_courses	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
11	elective_courses	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
12	course_credithours	int(11)			Yes	NULL			Change Drop More
13	course_id	int(11)			No	None			Change Drop More
14	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More

Figure 4.48: Coursedetails table

Figure 4.48 shows the Coursedetails table. This table expands on course-specific information, including description, course_name, course_duration, course_level, course_language, course_entry, course_career, course_fees, course_contact, core_courses, elective_courses, and course_credithours. It provides in-depth academic details in the Course Details page, helping students make informed enrollment decisions.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More

Figure 4.49: Public_universities table

Figure 4.49 shows the Public_universities table. The Public_universities table (id, name) lists all the malaysian public universities in Malaysia. This table supports the University Details screens, ensuring users can explore campus-specific information.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	campus_id	int(11)			No	None			Change Drop More
2	campus_name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	university_id	int(11)			Yes	NULL			Change Drop More

Figure 4.50: Campuses table

Figure 4.50 shows the Campuses table. The Campuses table (campus_id, campus_name, university_id) stores branch campus details. Similar with the Public_universities table, this Campuses table also supports the University Details screens.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	type_name	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	university_id	int(11)			Yes	NULL			Change Drop More

Figure 4.51: Propertytypes table

Figure 4.51 shows the Propertytypes table. This table categorises rental properties by type (id, type_name, university_id). They enable advanced filtering in the “Nearby Accommodation” page, allowing users to search for specific housing setups such as apartments, shared rooms, or studio units.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	configuration_description	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	property_type_id	int(11)			Yes	NULL			Change Drop More

Figure 4.52: Propertyconfigurations table

Figure 4.52 shows the Propertyconfigurations table. This table categorises rental properties by configuration (id, configuration_description, property_type_id). They enable advanced filtering in the “Nearby Accommodation” page, allowing users to search for specific housing setups such as apartments, shared rooms, or studio units.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	property_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	configuration_id	int(11)			Yes	NULL			Change Drop More
3	address	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	price	decimal(10,2)			Yes	NULL			Change Drop More
5	image_url	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
6	property_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
7	property_size	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
8	property_parking	int(11)			Yes	NULL			Change Drop More
9	owner_contact	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
10	nearby_amenities	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
11	property_condition	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
12	location_link	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
13	user_id	int(11)			Yes	NULL			Change Drop More

Figure 4.53: Properties table

Figure 4.53 shows the Properties table. The Properties table stores rental listings with details like property_id, address, price, property_name, and property_size. This table powers the “Nearby Accommodation” page, allowing users to search for specific housing setups such as apartments, shared rooms, or studio units.

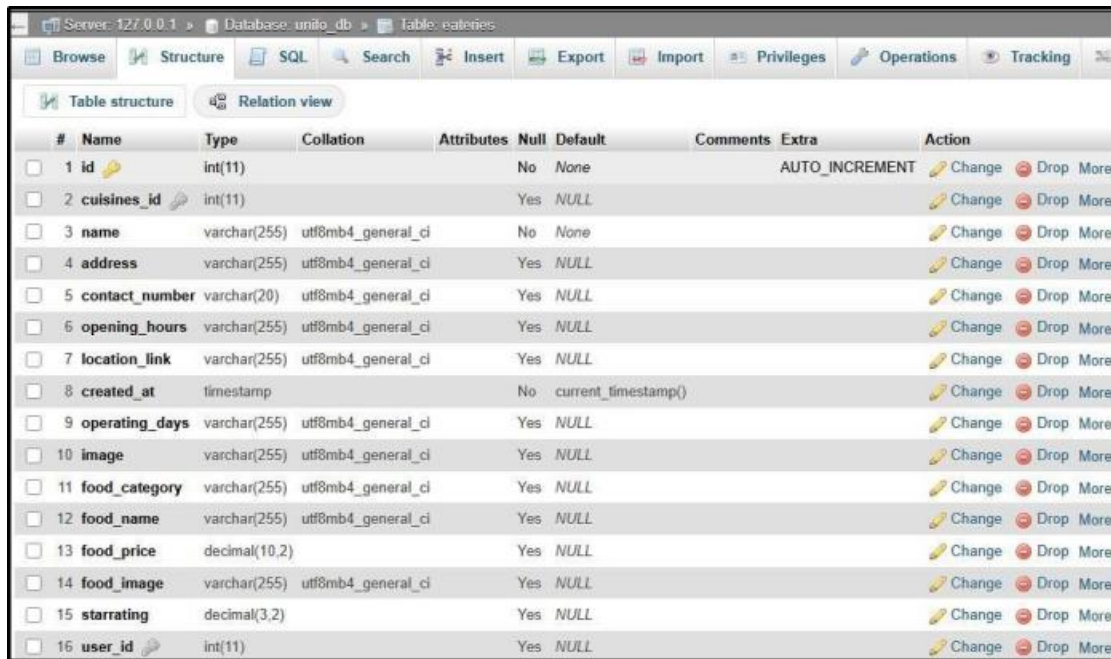
Server: 127.0.0.1 > Database: unifo_db > Table: cuisines

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)

[Table structure](#)
[Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐ 2	cuisine_name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
☐ 3	university_id	int(11)			Yes	NULL			Change Drop More

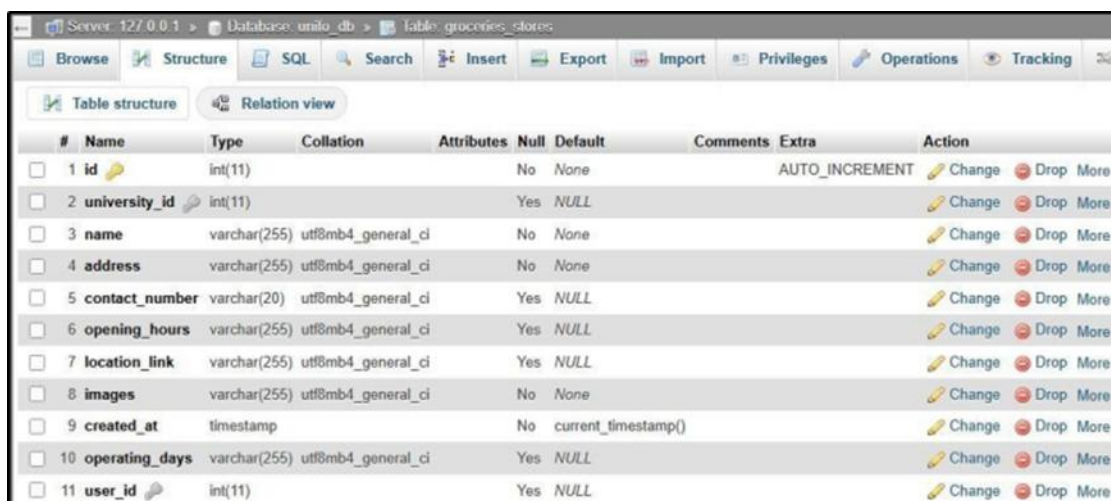
Figure 4.54: Cuisines table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	id	int(11)			No	None		AUTO_INCREMENT	
☐ 2	cuisines_id	int(11)			Yes	NULL			
☐ 3	name	varchar(255)	utf8mb4_general_ci		No	None			
☐ 4	address	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 5	contact_number	varchar(20)	utf8mb4_general_ci		Yes	NULL			
☐ 6	opening_hours	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 7	location_link	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 8	created_at	timestamp			No	current_timestamp()			
☐ 9	operating_days	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 10	image	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 11	food_category	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 12	food_name	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 13	food_price	decimal(10,2)			Yes	NULL			
☐ 14	food_image	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 15	starrating	decimal(3,2)			Yes	NULL			
☐ 16	user_id	int(11)			Yes	NULL			

Figure 4.55: Eateries table

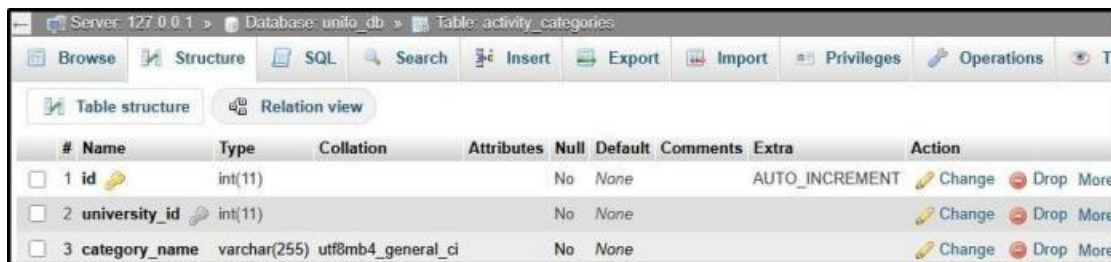
Figure 4.54 shows the Cuisine table and Figure 4.55 shows the Eateries table. The Cuisines table (id, cuisine_name, university_id) categorizes food types, while the Eateries table (id, cuisines_id, name, address, contact_number, opening_hours, location_link, created_at, operating_days, starrating, user_id) lists restaurants and cafes. These tables power the “Nearby Eateries” page, where users discover dining options near their campus.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	id	int(11)			No	None		AUTO_INCREMENT	
☐ 2	university_id	int(11)			Yes	NULL			
☐ 3	name	varchar(255)	utf8mb4_general_ci		No	None			
☐ 4	address	varchar(255)	utf8mb4_general_ci		No	None			
☐ 5	contact_number	varchar(20)	utf8mb4_general_ci		Yes	NULL			
☐ 6	opening_hours	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 7	location_link	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 8	images	varchar(255)	utf8mb4_general_ci		No	None			
☐ 9	created_at	timestamp			No	current_timestamp()			
☐ 10	operating_days	varchar(255)	utf8mb4_general_ci		Yes	NULL			
☐ 11	user_id	int(11)			Yes	NULL			

Figure 4.56: Groceries_stores table

Figure 4.56 shows the Groceries_stores table. The Groceries_stores table (id, university_id, name, address, contact_number, opening_hours, location_link, created_at, operating_days, user_id) lists nearby supermarkets. These tables support the “Nearby Grocery Stores” page, helping students locate essential shopping destinations.

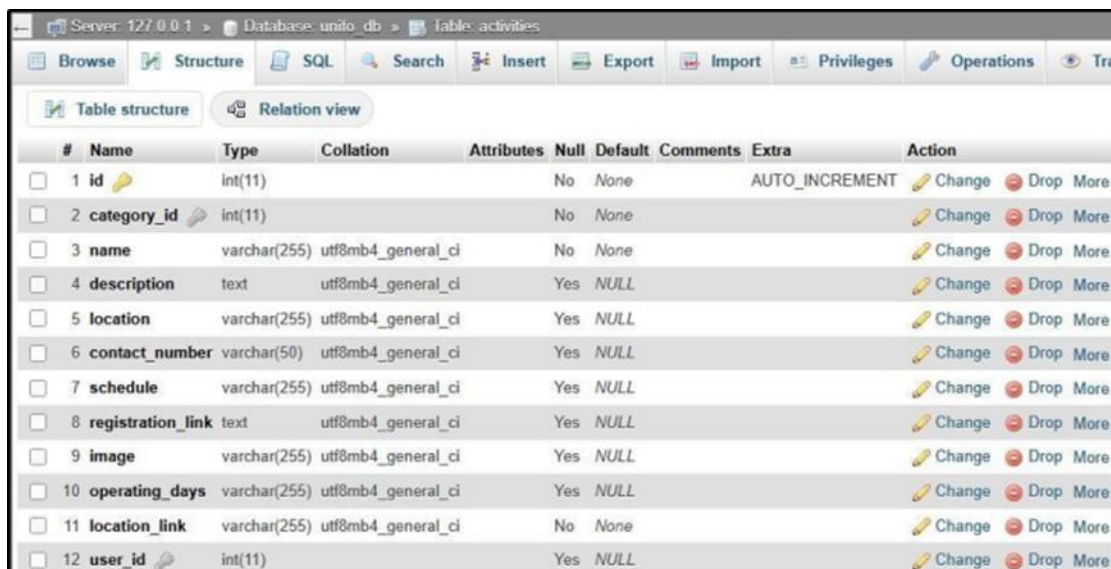


The screenshot shows the 'Table structure' view for the 'activity_categories' table in a database management tool. The table has three columns: 'id' (int(11), AUTO_INCREMENT), 'university_id' (int(11)), and 'category_name' (varchar(255), utf8mb4_general_ci). Each column has 'Change', 'Drop', and 'More' action buttons.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	university_id	int(11)			No	None			Change Drop More
3	category_name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

Figure 4.57: Activity_categories table

Figure 4.57 shows the Activity_categories table. The Activity_categories table (id, university_id, category_name) classifies recreational activities. This table drives the “Nearby Activity Places” page, where users explore campus events and social opportunities.

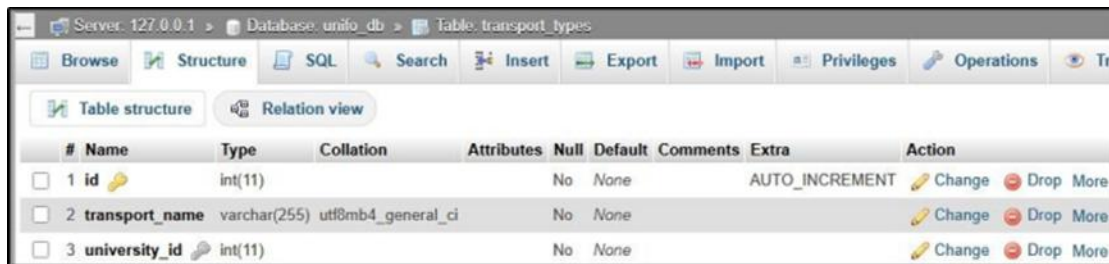


The screenshot shows the 'Table structure' view for the 'activities' table in a database management tool. The table has twelve columns: 'id' (int(11), AUTO_INCREMENT), 'category_id' (int(11)), 'name' (varchar(255), utf8mb4_general_ci), 'description' (text, utf8mb4_general_ci), 'location' (varchar(255), utf8mb4_general_ci), 'contact_number' (varchar(50), utf8mb4_general_ci), 'schedule' (varchar(255), utf8mb4_general_ci), 'registration_link' (text, utf8mb4_general_ci), 'image' (varchar(255), utf8mb4_general_ci), 'operating_days' (varchar(255), utf8mb4_general_ci), 'location_link' (varchar(255), utf8mb4_general_ci), and 'user_id' (int(11)). Each column has 'Change', 'Drop', and 'More' action buttons.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	category_id	int(11)			No	None			Change Drop More
3	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	description	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
5	location	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
6	contact_number	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More
7	schedule	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
8	registration_link	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
9	image	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
10	operating_days	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
11	location_link	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
12	user_id	int(11)			Yes	NULL			Change Drop More

Figure 4.58: Activities table

Figure 4.58 shows the Activities table. The Activities table (id, category_id, name, description, location, contact_number, schedule, registration_link, operating_days, location_link, user_id) stores event details. This table also supports the “Nearby Activity Places” page.

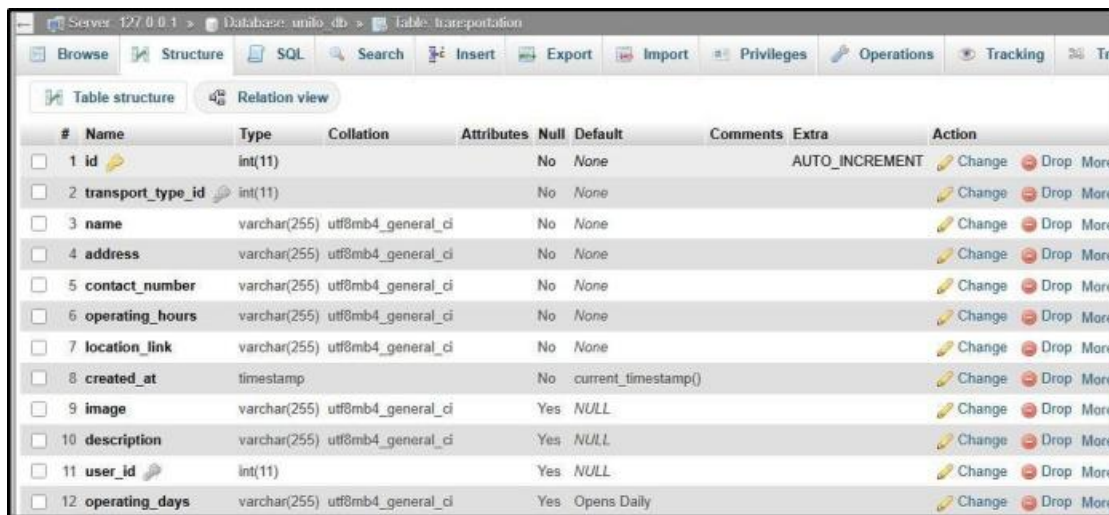


The screenshot shows the 'Table structure' view for the 'transport_types' table. The table has three columns: 'id' (int(11), primary key, AUTO_INCREMENT), 'transport_name' (varchar(255), utf8mb4_general_ci), and 'university_id' (int(11), foreign key). The interface includes tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Tracking.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	transport_name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	university_id	int(11)			No	None			Change Drop More

Figure 4.59: Transport_types table

Figure 4.59 shows the Transport_types table. The Transport_types table (id, transport_name, university_id) categorizes transportation options.



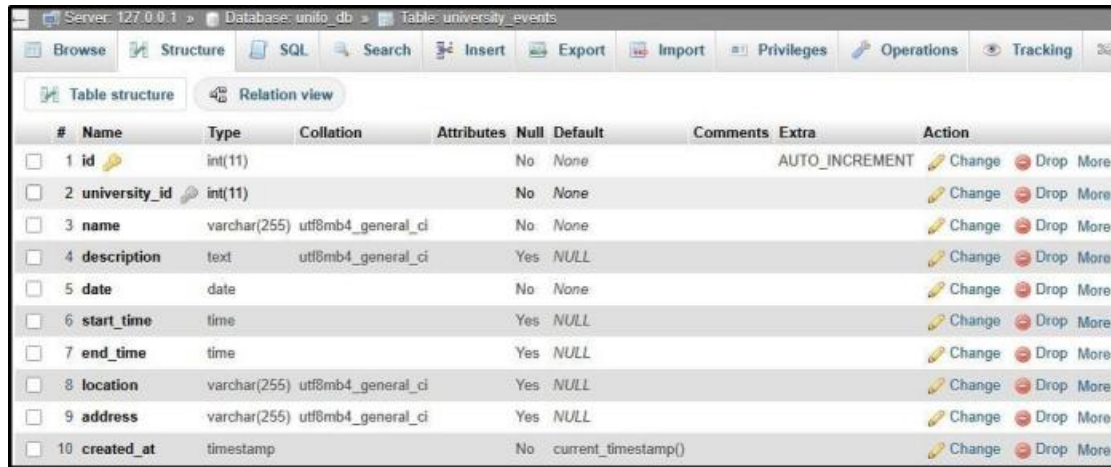
The screenshot shows the 'Table structure' view for the 'transportation' table. The table has 12 columns: 'id' (int(11), primary key, AUTO_INCREMENT), 'transport_type_id' (int(11), foreign key), 'name' (varchar(255), utf8mb4_general_ci), 'address' (varchar(255), utf8mb4_general_ci), 'contact_number' (varchar(255), utf8mb4_general_ci), 'operating_hours' (varchar(255), utf8mb4_general_ci), 'location_link' (varchar(255), utf8mb4_general_ci), 'created_at' (timestamp, current_timestamp()), 'image' (varchar(255), utf8mb4_general_ci), 'description' (varchar(255), utf8mb4_general_ci), 'user_id' (int(11), foreign key), and 'operating_days' (varchar(255), utf8mb4_general_ci). The interface includes tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Transactions.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	transport_type_id	int(11)			No	None			Change Drop More
3	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	address	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	contact_number	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
6	operating_hours	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
7	location_link	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
8	created_at	timestamp			No	current_timestamp()			Change Drop More
9	image	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
10	description	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
11	user_id	int(11)			Yes	NULL			Change Drop More
12	operating_days	varchar(255)	utf8mb4_general_ci		Yes	Opens Daily			Change Drop More

Figure 4.60: Transportation table

Figure 4.60 shows the Transportation table. The Transportation table (id, transport_type_id, name, address, contact_number, operating_hours, location_link, created_at, user_id,

operating_days) lists available services. This table supports the “Nearby Transportation” page, helping users navigate campus and city transit systems.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐ 2	university_id	int(11)			No	None			Change Drop More
☐ 3	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
☐ 4	description	text	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐ 5	date	date			No	None			Change Drop More
☐ 6	start_time	time			Yes	NULL			Change Drop More
☐ 7	end_time	time			Yes	NULL			Change Drop More
☐ 8	location	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐ 9	address	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐ 10	created_at	timestamp			No	current_timestamp()			Change Drop More

Figure 4.61: University_events table

Figure 4.61 shows the University_events table. The University_events table (id, university_id, name, description, date, start_time, end_time, location, address, created_at) stores institutional events. This table powers the “Event Calendar” page and “University Event” page, enabling students to track academic and event schedules.

4.1 Summary

In summary, this chapter has detailed the implementation of the Unifo+ application, building upon the analysis and design outlined in the previous chapter. Various libraries and tools were utilized to facilitate development, enhance user experience, and accelerate progress. For instance, modern UI frameworks helped streamline navigation and layout consistency across different university pages, while mapping APIs enabled seamless integration of location-based features such as nearby facilities and eateries.

The implementation section systematically explains key components of the application, beginning with the main dashboard, followed by university-specific information pages, course prospectus, accommodation and facilities details, and nearby services. Each feature was designed to provide students with comprehensive and easily accessible information tailored to their selected university.

Additionally, the chapter describes the database structure implemented to efficiently manage user data, university details, course information, and location services, ensuring data consistency and smooth application performance.

CHAPTER 5: TESTING

5.0 Testing

This section discusses on the testing of Unifo+. The testing methods include unit testing, integration testing, and usability testing. The application is tested using an iPhone 14 Plus running on iOS 18.1.1, accessed through the Expo Go application. The Unifo+ application itself is developed using Visual Studio Code. The iPhone 14 Plus features a 6.7- inch display with a resolution of 2778 x 1284, allowing for detailed mobile interface evaluation. For usability testing, various iPhone models with different page sizes are used to ensure responsiveness and compatibility across devices. Test documentation is provided in the Appendix C section of this report.

5.1 Unit Testing

In unit testing, individual parts of the mobile application are tested in isolation to verify their functionality. When several units are grouped within the same interface, they are referred to as components. This testing phase helps confirm that each module behaves correctly and fulfills its intended purpose based on functional requirements. The developer carries out this testing by executing a set of predefined test procedures using specific inputs. The outcomes are recorded with statuses such as ‘pass’, ‘fail’, ‘not executed’, or ‘suspended’. Additionally, any detected issues are classified by severity levels: ‘low’, ‘medium’, ‘high’, or ‘critical’. The actual results are then compared against the expected outcomes outlined in the test cases to identify any discrepancies or faults.

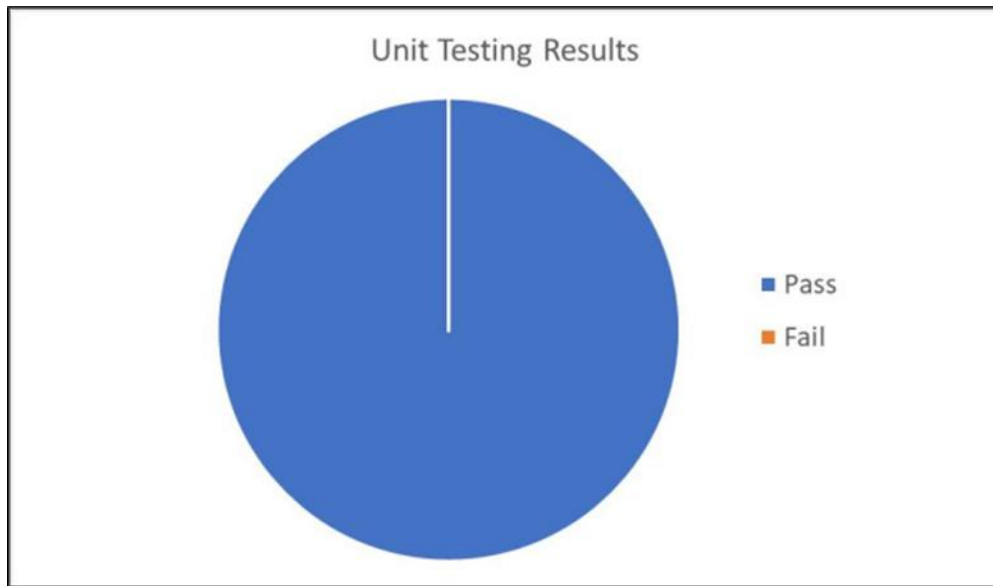


Figure 5.1: Unit Testing Result

The detailed unit test procedure on each module is attached in the report Appendix C. Total of 161 unit testing have been carried out on 29 different modules and performed by the developer. From Figure 5.1, it is depicted that 1 out of 161 tests, or approximately 0% have failed, while the rest have passed. From the percentage shown, it can be concluded that the application is functional because it is higher than 80%. Most of the error has been eliminated after testing process and achieve the score of 100%.

5.2 Integration Testing

Integration testing focuses on evaluating how different software modules interact and communicate with each other. It helps confirm the overall application's performance and reliability by ensuring that individual components function properly when combined. This testing verifies that the integrated parts work together as intended and meet the functional requirements. Typically, developers carry out integration testing after completing unit testing.

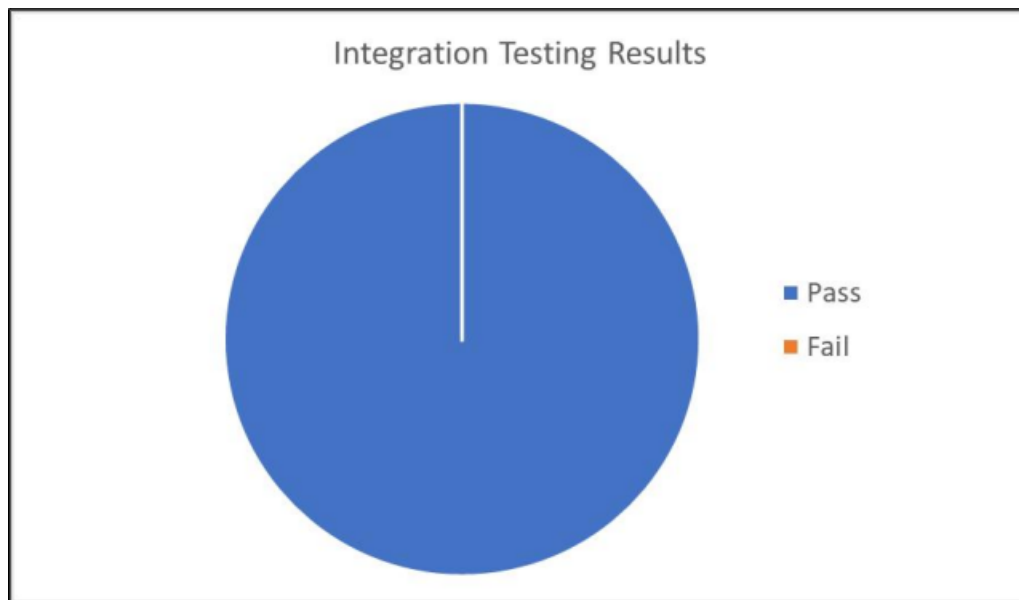


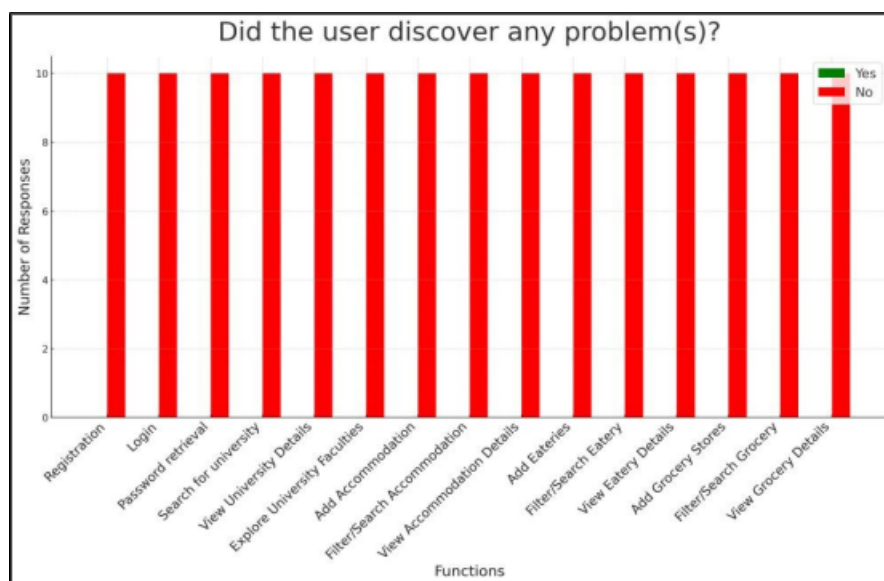
Figure 5.2: Integration Testing Result

The integration test procedures are attached as Appendix C. There are a total of 29 modules, and 206 tests are carried out by the developer. From Figure 5.2, it is depicted that the total percent that has passed the test is 100%, equivalent to 206 tests being passed. The result shows that the major components can integrate with each other to make the application function, giving it a 100% score.

5.3 User Acceptance Testing

User Acceptance Testing (UAT) was carried out to assess the usability, functionality, and overall user experience of the Unifo+ mobile application. The goal was to identify any issues that might affect user satisfaction, ease of use, or the performance of core features before final deployment. A total of 10 respondents participated in the UAT. Testing was conducted in person, as all respondents were required to be connected to the same local network as the developer. The application was run locally using Expo Go, and respondents accessed it by first downloading the Expo Go application on their mobile devices. They then scanned a QR code displayed from the developer's terminal in Visual Studio Code to launch the Unifo+ application on their phones.

During the testing session, respondents interacted with core features such as user registration, university exploration, accommodation search, event management, and settings customization. Feedback was collected through a structured questionnaire comprising both Likert-scale (quantitative) and open-ended (qualitative) questions. Details of the UAT scenarios and the questionnaire used can be found in Appendix D and Appendix E, respectively.



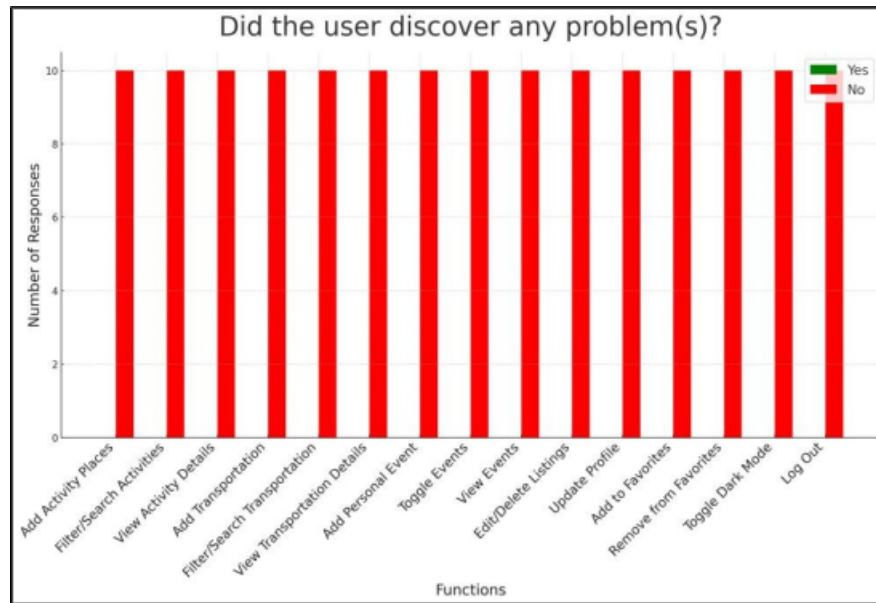
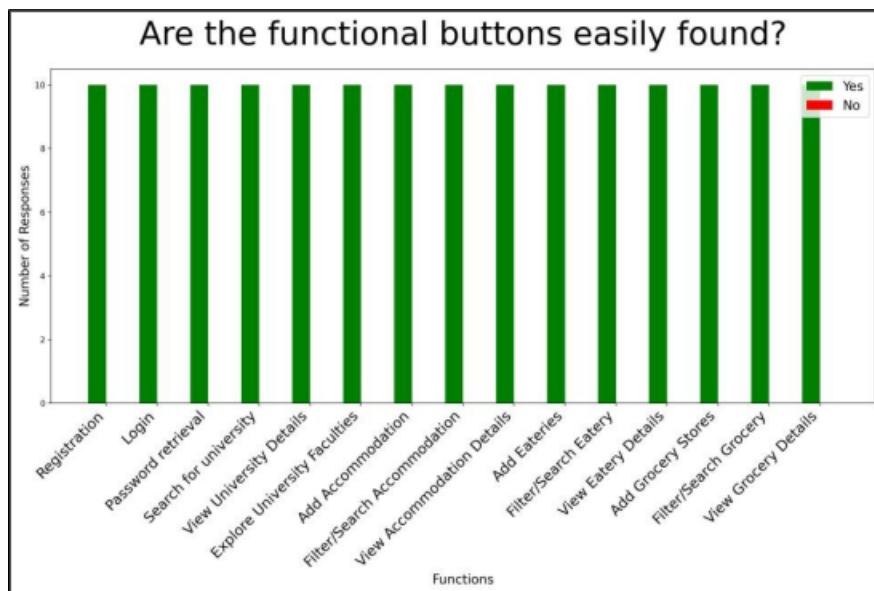


Figure 5.3: Question 1 (Did the user discover any problems?)

Figure 5.3 shows the result of for the first close-ended question. The question is mainly asking the respondents whether they found any problems in using the functions listed or not. Most of the answers are No. Therefore, the application performed well while carrying out the features.



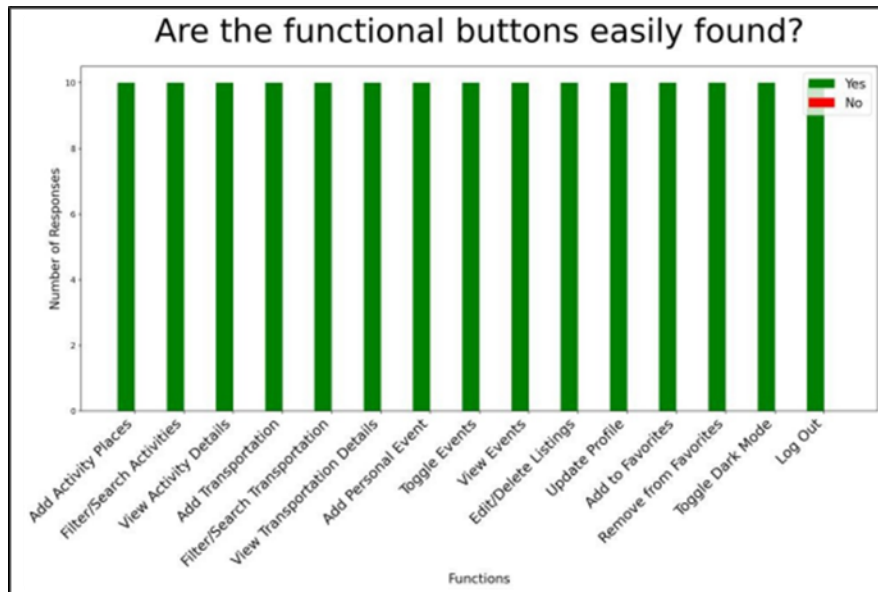


Figure 5.4: Question 2 (Are the functional buttons easily found?)

Figure 5.4 depicts the result for the second close-ended question. The question is aimed whether the respondents easily locate the functional buttons or not. Most of the answers are Yes. Therefore, the application performed well while carrying out the features. Most of the respondents easily locate the buttons to complete the scenario when performing the user acceptance testing, such as registration, project creation, grouping process, exporting the result, and so on. Therefore, the application has user-friendly interface where the respondents able to locate the functional button without any issues.

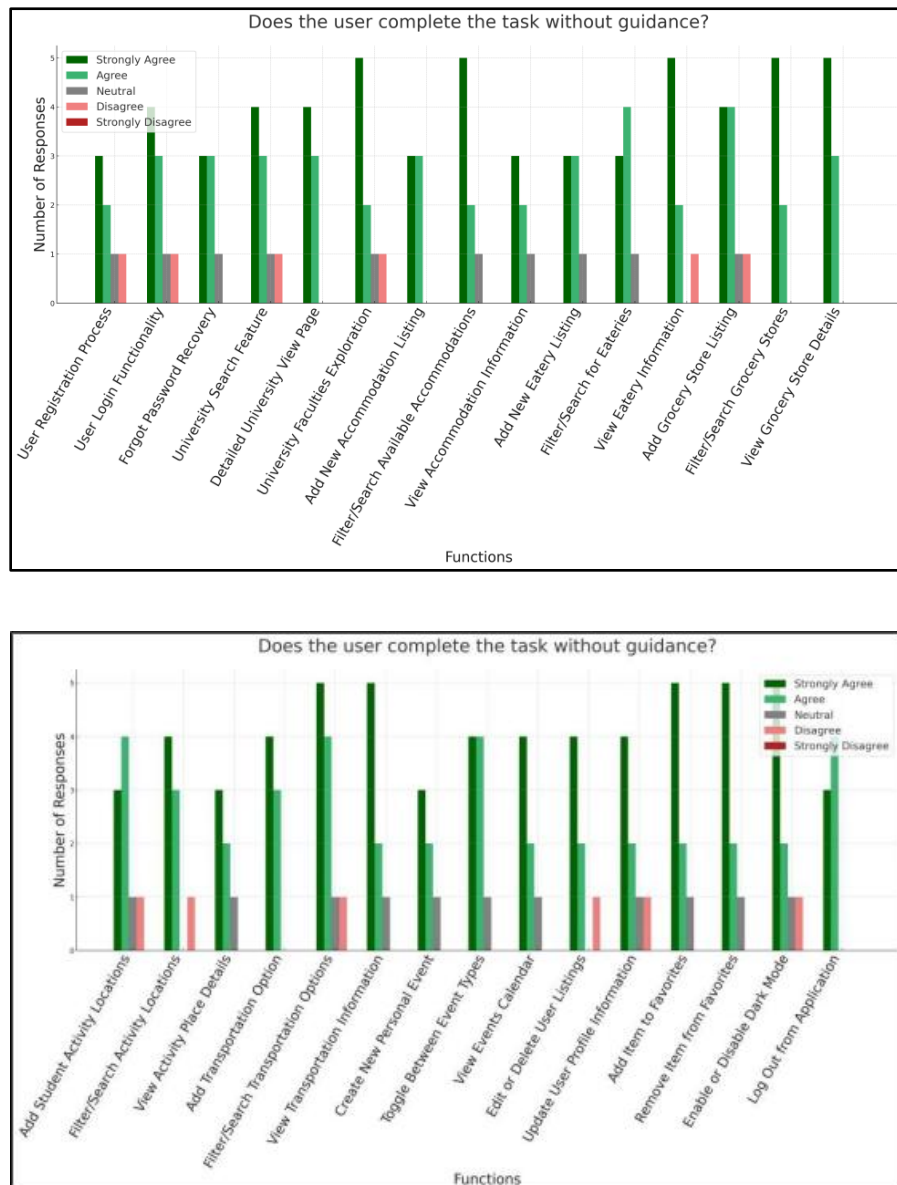


Figure 5.5: Question 3 (Does the user complete the task without guidance?)

Figure 5.5 shows the result of for the third close-ended question. Based on the results in Figure 5.5, nearly all 30 scenarios are easily performed by the respondents without any guidance because most of them have voted strongly agree, and agree one respondent has chosen strongly disagree for the grouping process.

As part of the User Acceptance Testing (UAT) for the Unifo+ application, users were asked several open-ended questions to gather feedback on the interface, key features, and overall usefulness of the application. The responses highlight both strengths and areas for improvement. Users generally appreciated the organized layout, helpful tools like the course prospectus and nearby facilities, and the application's ability to support new students. However, some mentioned the need for faster screen transitions, clearer navigation, and more personalized university pages. Suggestions for improvement included adding budgeting tools, student forums, and scholarship information. Selected responses from the UAT are summarized in Table 5.1 below.

Table 5.1: Selected Answers from Respondents for Open-Ended Question

No.	Open – ended questions and answers
1	What do you think about the user interface of Unifo+?
	The UI is quite good. The icons and buttons are organized also, so far everything good.
	The user interface looks nice, but some transitions between screens feel a bit slow, especially when switching universities.
	The navigation a bit confusing, if can provide some tutorials.
	I think the design could reflect each university's identity a little more.
	I like your app colours, but the font size make it larger?
2	What feature(s) you prefer the most and hate the most?
	I think the course prospectus is useful, can compare programmes but for car rental a bit useless I think? Can remove lah.
	The nearby facilities not bad, many times I keep cannot find tutorial rooms for my class near the FCSIT. Hmm...other than that, not much comments.
	The dashboard looks quite nice. For the part I hate is some pictures take too long to load.
	I really like the "Nearby Eateries" feature, can find more things to eat, donno where to eat after lectures. I don't like the nearby groceries stores part, why no price range one?
	I like the view location for the screens because I can see the map and route when I wanna go somewhere. However, the nearby eateries feature I don't really like, as it don't provide any reviews so I don't know whether the food there is good or not.
3	What other features do you think should be added into the Unifo+

	application to make it better?
	Pls...put scholarships and financial aid options available at each university.
	I'd like to see integration with social media or a "student story" section where students can post their university life experiences.
	I think next time you can add an expense management tool inside, as I think we really need to save money to be spent for food and other stuff.
	It would be amazing if Unifo+ had a live chat feature or discussion forum where students from different universities can interact and help each other.
4	If you have just been offered to study in one of the public universities in Malaysia, would you use Unifo+ to guide you through your university life? Please give your reason.
	Yeah definitely, so I no need to find here and there where to eat or live, so can focus on my studies.
	Yeah but not every day lah, maybe when I cannot find places to eat or I wanna go somewhere.
	Yes, because it's especially helpful for planning ahead.
	Maybe. I'd try it for the first few weeks but I'd still rely on seniors and WhatsApp groups for up-to-date tips.
	Yes, definitely. As someone new to the university environment, Unifo+ gives me a good overview of everything I need, from accommodation to course information.

Firstly, the first question explored the respondents' experience with the user interface of Unifo+. Most responses were generally positive. Many users appreciated the clean layout, organized buttons, and ease of navigation between different university pages. Some respondents highlighted the visual appeal and welcoming design, while others noted areas for improvement, such as responsiveness and the need for on boarding tutorials. These responses indicate that while Unifo+ is largely user-friendly and accessible, there is room for further enhancement, particularly in personalisation and first-time user guidance.

Secondly, respondents were asked to identify which features they liked the most and the least. The Course Prospectus, Nearby Facilities, University Dashboards, and Nearby Eateries were frequently mentioned as favorite features. These were praised for their usefulness in providing localized, relevant information to students new to the campus environment. On the other hand, some respondents found features like Car Rental, Groceries Stores Nearby, and

certain image-heavy sections (e.g., Facilities Gallery) less appealing. The reasons included limited usefulness for students, incomplete data, and slow loading times. These insights suggest areas where features could be optimized or streamlined to better suit student needs.

Thirdly, when asked about suggestions for improvement, respondents proposed valuable additions to the application. These included features such as a scholarship and financial aid section, student budgeting tools, a social interaction platform (like discussion forums or student stories), and enhanced university identity customization. These suggestions reflect users' desire for a more comprehensive and community-driven platform that goes beyond static information to support their academic and social experience.

Lastly, regarding whether users would use Unifo+ if they were newly accepted into a public university, most responses were affirmative. Many respondents stated that the application would be extremely helpful during the early stages of university life, especially for navigating the campus and accessing centralized information. A few noted they would use it primarily during orientation or the first few weeks, while continuing to rely on peer networks for real-time updates. This shows that Unifo+ is perceived as a valuable onboarding tool for new students, although its long-term use may depend on how dynamic and regularly updated the content is.

5.0 Summary

Chapter 5 demonstrates the rigorous testing approach undertaken to validate Unifo+, ensuring that every component meets the highest standards of functionality and user satisfaction. The comprehensive testing framework, encompassing unit testing (100% pass rate across 258 tests), integration testing (100% success across 333 tests), and user acceptance testing with real-world respondents, confirms the application's reliability and effectiveness. Through systematic evaluation of individual modules, seamless integration verification, and positive user feedback highlighting the application's intuitive interface and valuable features, Unifo+ is proven to be a robust and user-centric solution that successfully fulfills its mission of transforming how students navigate and experience their university environments.

CHAPTER 6: CONCLUSION & FUTURE WORKS

6.1 Introduction

This chapter will summarize the overall project achievements and discuss about the limitation and future enhancements.

6.2 Project Achievements

The achievement and completeness of the project are measured by the accomplishment in target objectives. A summary is shown in Table 6.1

Table 6.1: Project Achievements

Objectives	Achievements
To develop an application that provides up-to-date information for all public universities in Malaysia, including official links to university websites and other resources for further exploration.	Unifo+ is a completed Centralised University Information application, featuring a centralised database that offers detailed profiles for all 20 public universities in Malaysia. Each university's page includes direct links to official websites, course information, academic calendars, facilities, and other essential student resources.
To integrate real-time location services with open-source map APIs, enabling students to measure travel times to campus locations and surrounding areas.	Unifo+ has included the implementation of real-time location features using open-source mapping APIs (e.g., Open Street Map API), allowing users to view campus locations, estimate travel time, and explore nearby accommodations, eateries, grocery stores, and entertainment options.
To evaluate the usability of the proposed application (Unifo+).	Unifo+ is built and tested for its quality by conducting usability testing with student users from Universiti Malaysia Sarawak (UNIMAS). Feedback was gathered through surveys and observation, indicating that the application is effective in supporting students' academic and lifestyle planning. Iterative improvements were made based on user feedback to enhance the overall user experience.

6.1 Project Limitations

While Unifo+ delivers core functionalities as intended, several technical and design constraints emerged during development that currently limit its full potential. These limitations stem primarily from time constraints and the developer's evolving expertise in cross-platform mobile development.

6.1.1 Technical Performance Limitations

While the application works well for basic use, some technical limitations exist due to using React Native. The framework helps the app run on both Android and iOS, but some parts could be smoother. For example, when loading long lists of universities or accommodation options, there might be slight delays. The code structure could be more efficient - currently it uses simpler state management that works but isn't optimized for complex data handling. Database operations take longer on slow networks. These aren't major problems that break the app, but improvements could make it faster and more responsive.

6.1.2 Interface Design Limitations

The application's design is clean and functional, but not as polished as professional applications. Some screens with lots of information (like campus maps) could be better organised. Buttons and menus work properly but lack the smooth animations found in many modern applications. Icons and visual elements are consistent but could be more refined. Some color choices might be difficult for users with vision impairments to see clearly. These design limitations come from focusing on making the app work well first, rather than making it look perfect. The interface gets the job done effectively, even if it's not as visually impressive as some

commercial applications.

6.1.3 Language and Input Limitations

The current English-only implementation restricts the application's accessibility for many Malaysian students who primarily use Malay Language or Chinese dialects in their academic communications. While English serves as a common academic language, the lack of Malay Language support, Malaysia's national language, limits adoption among local students, particularly in public universities. The system currently struggles with non-Latin character inputs, which could affect students searching for universities or locations using Malay or Chinese terms. Additionally, date and number formats follow Western conventions, which may not align with local academic calendars or common usage patterns. While full multilingual support would require significant backend adjustments, even basic Malay Language integration would greatly improve accessibility for the primary target audience. Future versions could prioritise Malay Language support before considering additional languages like Mandarin or Tamil based on user demographics.

6.1 Future Enhancements

Despite its current limitations, Unifo+ has significant potential for growth and could become a widely used platform among Malaysian public university students. Future updates will focus on enhancing usability, expanding features, and improving overall performance to create a more seamless and engaging experience.

One of the key areas for improvement is the user interface and overall user experience. A more polished and dynamic UI with smoother transitions, refined animations, and modern design elements will make navigation more intuitive. The application could also benefit from deeper customization options, such as adjustable themes beyond just dark and light mode, allowing users to personalise their interface. Additionally, expanding language support to include Malay and Chinese would make Unifo+ more accessible to a broader audience, particularly in multilingual regions where students may prefer using the application in their native language.

Beyond visual and accessibility improvements, Unifo+ could integrate several advanced features to further assist students. Real-time messaging would enable direct communication between users for accommodation inquiries, event coordination, or ride-sharing, fostering a stronger sense of community. AI-driven recommendations could analyze user behavior to suggest relevant accommodations, eateries, and events, streamlining decision-making. Augmented Reality (AR) campus navigation would be particularly useful for new students, helping them locate buildings, facilities, and event venues with visual guidance. Furthermore, integrating with university systems through APIs could provide live updates on academic schedules, exam timetables, and official announcements, ensuring students stay informed without switching between multiple platforms.

To ensure long-term sustainability, Unifo+ could explore monetization strategies while

maintaining value for students. Introducing a premium subscription model with benefits such as an ad-free experience, exclusive discounts, and early access to new features could generate revenue without compromising core functionality. Partnering with local businesses, such as restaurants, bookstores, and transport services, would allow Unifo+ to offer student- exclusive deals, enhancing user engagement while supporting the local economy. A verified user system, where accommodations and service providers undergo a trust-based review process, would also improve reliability and safety, making the platform more dependable for students seeking housing or services. By continuously refining its design, expanding its features, and fostering strategic partnerships, Unifo+ can evolve into an indispensable tool for university students, helping them navigate academic and social life with greater ease and confidence.

6.1 Conclusion

In this project, Unifo+, a cross-platform university assistance application was successfully developed from concept to implementation. The development followed an iterative approach, with each sprint refining core functionalities such as university exploration, accommodation listings, event management, and local service directories. Despite challenges in UI design and performance optimization, the application delivers a functional and user- friendly experience for students. Testing phases, including unit testing, integration testing, and usability testing, helped identify and resolve critical issues, ensuring a stable final product. Moving forward, Unifo+ will continue evolving with feature expansions, performance enhancements, and broader accessibility improvements. The ultimate goal is to publish the application on Google Play Store and Apple App Store, for university students seeking academic and lifestyle support. With further development and user feedback, Unifo+ has the potential to become an essential tool for students navigating university life.

7.0 References

- BACP. (2023). British Association for Counselling and Psychotherapy. BACP.co.uk.
Retrieved October 23, 2024, from <https://www.bacp.co.uk/>
- BasuMallick, C. (2024, March 19). Android OS: History, features, versions, and benefits.
Spiceworks. Retrieved November 26, 2024, from
<https://www.spiceworks.com/tech/tech-general/articles/android-os/>
- De Bruyn, S., & Van Eekert, N. (2023). Understanding the academic and social integration process of students entering higher education: Lessons learned from the COVID-19 pandemic. *Social Sciences*, 12(2), 67. <https://doi.org/10.3390/socsci12020067>
- DECODE Agency. (2023, March 16). Android vs. iOS application development: What's the difference? Decode Agency. Retrieved November 29, 2024, from <https://decode.agency/article/android-vs-ios-application-development/>
- Elprocus. (2019, March 28). What is Android? Introduction of Android OS & its applications. ElProCus - Electronic Projects for Engineering Students. Retrieved December 2, 2024, from <https://www.elprocus.com/what-is-android-introduction-features-applications/>
- IAEME Publication. (2020). Development of iOS: A revolutionary transformation and the future. IAEME Publication. <https://doi.org/10.34218/ijaret.11.6.2020.040>

Kislay. (2024, November 19). iOS introduction | iOS tutorial. Intellipaat Blog. Retrieved November 23, 2024, from <https://intellipaat.com/blog/tutorial/ios-tutorial/ios-introduction/>

KMS Solutions. (2022, November 10). Mobile app development: 10 best practices in 2022 to follow. Medium. Retrieved October 11, 2024, from <https://medium.com/@KMSSolutions/mobile-app-development-10-best-practices-in-2022-to-follow-4482bbffab8d>

Konings, K. D., & Seidel, T. (2022). Student expectations when entering an innovative learning environment: Identifying longitudinal patterns across student subgroups. *Educational Studies*, 1–21. <https://doi.org/10.1080/03055698.2022.2117544>

Momentum Suite. (2023, January 14). The differences between iOS and Android: A comparative analysis. Momentum Suite. Retrieved November 26, 2024, from <https://momentumsuite.com/software-testing/the-philosophy-and-differences-between-ios-and-android-a-comparative-analysis/>

Info UPU. (2018, January 25). Senarai UA universiti awam di Malaysia. Info UPU. Retrieved November 1, 2024, from <https://www.infoupu.com/senarai-ua-universiti-awam-di-malaysia/>

Roberts, S. (2023, November 30). What is Android? A complete guide to mobile technology. The Knowledge Academy. Retrieved November 24, 2024, from <https://www.theknowledgeacademy.com/blog/what-is-android/>

Suite, Momentum. (2023, January 14). The differences between iOS and Android: A comparative analysis. Momentum Suite. Retrieved November 26, 2024, from <https://momentumsuite.com/software-testing/the-philosophy-and-differences-between-ios-and-android-a-comparative-analysis/>

Yu, A. (2020, April 30). What is iOS? Apple's popular mobile operating system, explained. Udemy Blog. Retrieved November 20, 2024, from <https://blog.udemy.com/what-is-ios-apples-popular-mobile-operating-system-explained/>

8.0 Appendix

8.1 Appendix A - Survey Form

Survey on Centralised University Information Application For Malaysian Public Universities (Unifo+)

Dear Respondent,

My name is Chin Teck Yung, and I am an undergraduate student in Network Computing at Universiti Malaysia Sarawak (UNIMAS). For my Final Year Project, I am developing Unifo+, a centralized mobile application designed to enhance the university experience for students across Malaysia's 20 public universities.

Unifo+ integrates features such as up-to-date academic information, interactive campus maps, real-time location-based services, and travel time estimates. By consolidating essential resources into one user-friendly platform, the app aims to help students transition smoothly into university life and make informed decisions about their education, social activities, and daily routines.

To ensure that Unifo+ meets the needs of users like you, I kindly request your participation in a short survey. Your feedback on desired features and functionalities will be invaluable in making the application more user-friendly, inclusive, and effective. Rest assured, all information you provide will be kept strictly confidential and used solely for the purpose of this research project.

Thank you for your time and valuable input in helping us improve Unifo+.

Sincerely,
Chin Teck Yung
Network Computing
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak (UNIMAS)

[Next](#)[Clear form](#)

Survey on Centralised University Information Application For Malaysian Public Universities (Unifo+)

* Indicates required question

Public University

Select Your Public University *

Choose

Academic Year *

Choose

Back

Next

Clear form

Survey on Centralised University Information Application For Malaysian Public Universities (Unifo+)

* Indicates required question

Core features that should be included inside the University Mobile Application.

How important is it to have detailed course descriptions and requirements for every public university to help you choose the right courses and plan your academic journey? *

- ☐ Not useful
- ☐ Slightly useful
- ☐ Moderately useful
- ☐ Very useful
- ☐ Extremely useful

How helpful would you find real-time interactive campus maps for locating buildings and facilities, especially during your first year? *

- ☐ Not useful
- ☐ Slightly useful
- ☐ Moderately useful
- ☐ Very useful
- ☐ Extremely useful

Would it be useful to have easy access to information about campus facilities (e.g., library, dining, sports) through the application? *

- ☐ Not useful
- ☐ Slightly useful
- ☐ Moderately useful
- ☐ Very useful
- ☐ Extremely useful

Back

Next

Clear form

Survey on Centralised University Information Application For Malaysian Public Universities (Unifo+)

* Indicates required question

Core features that should be included inside the University Mobile Application.

Would you find it helpful to have information on nearby transport and car rental services for easier mobility? *

- ☐ Not useful
- ☐ Slightly useful
- ☐ Moderately useful
- ☐ Very useful
- ☐ Extremely useful

Would a feature that allows students to book flight tickets through the app be beneficial for your travel needs? *

- ☐ Not useful
- ☐ Slightly useful
- ☐ Moderately useful
- ☐ Very useful
- ☐ Extremely useful

How helpful would it be to access information on entertainment options near your university to enhance your social experience? *

- ☐ Not useful
- ☐ Slightly useful
- ☐ Moderately useful
- ☐ Very useful
- ☐ Extremely useful

Back

Next

Clear form

Survey on Centralised University Information Application For Malaysian Public Universities (Unifo+)

* Indicates required question

Core features that should be included inside the University Mobile Application.

How interested are you in having a social feed or bulletin board for student events * and activities in the app?

- ☐ Not useful
- ☐ Slightly useful
- ☐ Moderately useful
- ☐ Very useful
- ☐ Extremely useful

Would it be useful if the app provided links to official university websites and other resources for exploring university services further? *

- ☐ Not helpful
- ☐ Slightly helpful
- ☐ Moderately helpful
- ☐ Very helpful
- ☐ Extremely helpful

How important is it for the app to support real-time updates for university resources and local amenities? *

- ☐ Not helpful
- ☐ Slightly helpful
- ☐ Moderately helpful
- ☐ Very helpful
- ☐ Extremely helpful

Back

Next

Clear form

Survey on Centralised University Information Application For Malaysian Public Universities (Unifo+)

Core features that should be included inside the University Mobile Application.

What other features or services would you like to see in this application that could help improve your university experience?

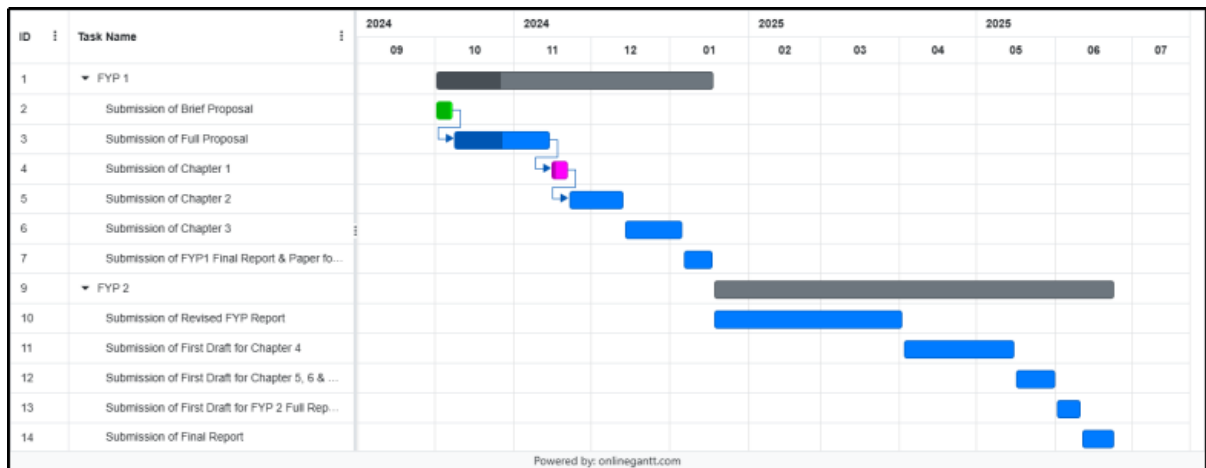
Your answer

Back

Submit

Clear form

8.2 Appendix B - Project Schedule



Appendix C - Unit Testing

Module Name: Public Universities Screen								
Test Case Description: Tests the display, search, and navigation functionalities of the public universities list, including branch expansion and filtering.								
Testing Objective: Verify that universities are correctly listed, searchable, and navigable, with proper handling of branch campuses.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-UA-01	Render and display university cards	Simulate loading the universities list screen	Mock university list with names, logo URLs, and descriptions	All universities are displayed as cards with logos and brief descriptions	Same as expected	Pass	High	Ensure quick loading and correct layout
TC-UA-02	Search universities by exact/partial name and reset	1. Enter full name (e.g., "Universiti Malaysia Sarawak") 2. Enter partial name (e.g., "UNIMAS") 3. Enter unrelated term (e.g., "Sunway University") 4. Clear input	Search queries as above	1. Exact match result 2. Partial match shows relevant university 3. No results for unrelated term 4. All universities reappear	Same as expected	Pass	High	Validates filtering logic and input handling
TC-UA-03	Branch campus expansion via chevron	1. Tap chevron icon on a university with branches (e.g., UiTM) 2. Tap anywhere else on the card	UiTM with branches in Sarawak and Penang	1. Branch list expands under card on chevron tap 2. Card remains unchanged on general tap	Same as expected	Pass	High	Confirms correct interaction handling
TC-UA-04	Branch campus list rendering	Simulate card with multiple branches	UiTM with sub-branches	Branches display clearly and correctly spaced under main university card	Same as expected	Pass	High	Check proper layout and spacing for nested items
TC-UA-05	Navigation and redirection validation	1. Tap on university card (e.g., Universiti Malaya) 2. Tap bottom navigation tabs: Dashboard, Settings, Events	Navigation events	1. University card tap navigates to detailed screen 2. Bottom tab bar switches views and highlights current tab	Same as expected	Pass	Medium	Confirms correct routing and screen transitions
TC-UA-06	UI state management and visual cues	1. Tap navigation tabs and observe icon state 2. Verify active tab highlight	Tab clicks	Correct active tab is visually highlighted	Same as expected	Pass	Low	Confirms visual cue for navigation feedback
Module Name: University Details Screen								
Test Case Description: Validates the accuracy of university information display, including external actions (website, map) and dynamic content updates.								
Testing Objective: Ensure university details load correctly, external links work, and content refreshes when switching universities.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-	Validate correct	Select university from list (e.g.,	University ID:	Header displays full	Same with	Pass	High	Ensures selected university data loads correctly

UD-01	university detail loading and header binding	UNIMAS) and load details screen	UNIMAS	university name: "Universiti Malaysia Sarawak (UNIMAS)"	the expected result			
TC-UD-02	Verify external actions: website and map buttons	1. Tap Website button 2. Tap Map button	URL: https://www.unimas.my Coordinates: UNIMAS location	1. Website opens in browser 2. Map opens with correct location	Same with the expected result	Pass	High	Confirms intent calls to external apps
TC-UD-03	Validate informational sections: About, Faculties, Facilities	Tap each info section button (About, Faculties, Facilities)	UNIMAS data for each section	Each section displays correct text, list, or icons	Same with the expected result	Pass	Medium	Confirms each section loads and binds correct content
TC-UD-04	Ensure detail screen updates when university is changed	Switch from one university (e.g., UNIMAS) to another (e.g., UPM)	University ID changes: UNIMAS → UPM	All screen content updates to reflect new university	Same with the expected result	Pass	Critical	Verifies dynamic state and content refresh
TC-UD-05	Verify back navigation from university details screen	Tap top-left back arrow on details screen	Current screen: University Details	App returns to Public Universities screen	Same with the expected result	Pass	Medium	Tests correct back-stack behavior and navigation logic
Module Name: University Faculties Screen Test Case Description: Tests the display and expansion of faculty lists and associated courses for a selected university. Testing Objective: Verify faculty cards expand/collapse correctly and course lists render accurately.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-FC-01	Fetch and display faculty list per university	Load faculties for UNIMAS	University: UNIMAS	List displays correct faculties for selected university	Same as expected	Pass	High	Ensures university-specific data loads properly
TC-FC-02	Expand/collapse faculty card and fetch courses	Tap on faculty card (once to expand, again to collapse)	Faculty ID: FCS_UNIMAS	Card toggles; course list shows/hides accordingly	Same as expected	Pass	High	Covers toggle logic and UI behavior
TC-FC-03	Display course list and fallback message	Expand faculty with and without assigned courses	Faculties: Engineering (with), Law (without)	Shows courses if available, otherwise “No courses available”	Same as expected	Pass	Medium	Handles data presence and absence correctly
TC-FC-04	Ensure single-card expansion and UI layout	Expand one faculty, scroll list, observe others	Multiple faculties (15+)	Only tapped card expands; layout scrolls smoothly	Same as expected	Pass	Medium	Includes scroll ability, spacing, and UI state
TC-FC-05	Validate chevron icon and course list style	Expand any faculty and observe icon and course styling	Faculty: Economics	Chevron icon updates; courses styled consistently	Same as expected	Pass	Low	Confirms visual feedback and style uniformity
Admin-Only Test Cases – Applicable to Admins Only (Users have no access)								
TC-	Verify admin-	Login as admin vs user	Roles: Admin, Student	Admin sees menu;	Same as	Pass	High	Role-based UI

FC-06	only menu visibility			user does not	expected			
TC-FC-07	Validate edit/delete faculty with course checks	Edit faculty & delete with/without courses	Faculty with & without courses	Edit form opens; delete blocked if courses exist	Same as expected	Pass	High	Edit/delete rules
TC-FC-08	Verify add course form opens properly	Tap Add Course in faculty menu	Faculty: FE_UNIMAS	Course form opens with faculty pre-selected	Same as expected	Pass	Medium	Form linkage
TC-FC-09	Verify course-level edit/delete for admin	Tap edit/delete on course	Course: Bachelor of Civil Engineering	Edit screen shows; delete removes course	Same as expected	Pass	High	Course actions
TC-FC-10	Validate add new faculty functionality	Tap add button, submit form	New Faculty: "Design"	Faculty added to list	Same as expected	Pass	High	Add faculty
TC-FC-11	Verify fallback for no faculties	Select university with no faculties	University: Dummy_U	"No faculties available" message shown	Same as expected	Pass	Medium	Empty state
Module Name: University Courses Details Screen Test Case Description: Validates the rendering of course details, including expandable sections, fee formatting, and navigation. Testing Objective: Ensure course information displays correctly, sections toggle properly, and financial data is formatted accurately.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-CD-01	Verify course details text fields	Positive test: Bind and render course info fields	Input Course Object: title = "Software Engineering", creditHours = 126, contact = "082-123456"	Title shows "Software Engineering", Credit Hours: 126, Contact: 082-123456	Same as expected	Pass	Medium	Covers direct data binding & formatting checks
TC-CD-02	Validate UI container structure and layout	Positive test: Render full course page	Course Data: Software Engineering	Course Info, Requirements, Description, Career Paths containers visible; Cards have visible borders, padding, shadows	Same as expected	Pass	High	Checks layout, z-index, visual styling
TC-CD-03	Verify expand/collapse behavior of sections	Positive test: Tap each expandable section twice	Trigger expand/collapse for Description, Entry Requirement, Core & Electives, Career Paths	Each section toggles independently, Chevron icon changes accordingly, no duplicate content	Same as expected	Pass	Medium	Includes rapid tap test and no duplication
TC-CD-04	Validate content rendering and dynamic	Positive & Negative test: Tap vs no tap on sections	Course description string, elective course list: [Cloud Computing,	Description text matches expected string exactly;	Same as expected	Pass	Medium	Guards against auto-expand bug and content correctness

	behavior		Data Analytics]	Electives rendered as bullet list; No auto-expansion without tap				
TC-CD-05	Confirm financial info display and formatting	Positive test: Render total fee field	Input: total fees = 14000	Displays "Total Fees: RM 14000.00" with correct Malaysian currency format	Same as expected	Pass	High	Critical for currency and fee display accuracy
Admin-Only Test Cases – Applicable to Admins Only (Users have no access)								
TC-CD-06	Check admin-only controls visibility	Login as admin vs user	Roles: Admin, User	Edit/Delete buttons visible only to admin	Same as expected	Pass	High	Access control
TC-CD-07	Validate navigation and data update on edit	Tap Edit, modify data, and save	Course: Software Engineering	Edit screen opens; changes update on details page	Same as expected	Pass	High	Edit/save flow
TC-CD-08	Validate form validation on save	Attempt save with empty required field	Title: ""	Error message shown; save blocked	Same as expected	Pass	Medium	Validation
TC-CD-09	Confirm delete functionality and confirmation	Tap Delete and confirm	Course: Software Engineering	Confirmation shown; course deleted and list updates	Same as expected	Pass	Critical	Delete flow
TC-CD-10	Verify no undo option after deletion	Delete course and check undo	Course deleted	No undo option available	Same as expected	Pass	Medium	UX finality
Module Name: Dashboard Screen Test Case Description: Tests the university selection dropdown, dynamic content updates, and navigation from feature buttons. Testing Objective: Verify dropdown functionality, content filtering by university, and correct redirection to feature screens. Remarks: Both users and admins will see the same layout.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-DA-01	Validate drop-down functionality and state changes	Load Dashboard, check default value, select university, reopen drop-down	Default → "Select University" Selected: "Universiti Malaya"	Default text shown initially; updates to selected university; list contains all universities	Same as expected	Pass	Medium	Drop-down basics
TC-DA-02	Verify dynamic content updates based on university selection	Select university, observe content changes	University: "Universiti Malaya"	Dashboard updates to show UM-specific data	Same as expected	Pass	High	Dynamic content

TC-DA-03	Show error message when no university is selected	Leave drop-down as default, try using buttons	No university selected	Error message shown: "Please select a university first"	Same as expected	Pass	High	Error if no selection
TC-DA-04	Validate grid layout of feature buttons	Load Dashboard screen and observe button layout	Features: Nearby Accommodation, Eateries, etc.	Buttons appear in proper grid format	Same as expected	Pass	Low	Button layout
TC-DA-05	Verify navigation from feature buttons	Select a university and click "Nearby Accommodation"	University: "Universiti Malaya"	User is navigated to Accommodation page	Same as expected	Pass	High	Navigation check
Module Name: Nearby Accommodation Screen Test Case Description: Validates accommodation listing, search, filtering, and user contribution flows. Testing Objective: Ensure users can search, filter, and add accommodations with proper validation and navigation. Remarks: Both users and admins will see the same layout.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-NA-01	Verify basic accommodation listing displays correctly.	Positive test: Ensure that the accommodation list loads with valid data for the selected university.	University: "Universiti of Malaya"	List of accommodations displays with images, names, and prices.	Same as expected result.	Pass	High	Verify images load dynamically as user scrolls.
TC-NA-02	Confirm search bar filters accommodations by name.	Positive test: Search for a known accommodation ("Unicentral Apartment") and ensure it appears in the list.	Search Query: "Unicentral Apartment"	Displays only the matching accommodation.	Same as expected result.	Pass	Medium	Search should handle case insensitivity and partial matches.
TC-NA-03	Validate price range filter.	Positive test: Apply price range filter (e.g., RM500 - RM1000) and verify that only matching accommodations are displayed.	Price Range: RM500 - RM1000	Displays accommodations within the specified price range.	Same as expected result.	Pass	Medium	Ensure currency formatting is consistent.
TC-NA-04	Validate accommodation type filter.	Positive test: Apply filter for "Apartments" and verify that only apartment listings are shown.	Type: "Apartments"	Displays only listings of type "Apartments."	Same as expected result.	Pass	Medium	Verify multi-select functionality.
TC-NA-05	Test sorting by price (Cheapest to most expensive).	Positive test: Apply "Cheapest first" sorting and verify the list starts with the lowest price.	Sort Option: "Cheapest first"	List starts with the cheapest accommodation, ordered by price.	Same as expected result.	Pass	Medium	Verify sorting persists after scrolling.
TC-NA-06	Verify accommodation list when price range, accommodation type, and price sorting filters applied.	Positive test: Apply multiple filters (e.g., price range RM800–RM1200, type "Apartments", sorted by "Cheapest first") and verify the displayed results match.	Price Range: RM800–RM1200, Type: "Apartments", Sort: "Cheapest First"	Displays only apartments within RM800–RM1200, sorted by price.	Same as expected result.	Pass	High	Ensure overlapping filters work correctly.
TC-	Verify filter reset	Positive test: Reset all filters and	Precondition:	All filters are	Same as	Pass	Medium	Ensure search query remains intact after reset.

NA-07	functionality.	verify the full accommodation list is restored.	Applied filters	cleared, and the full list is shown.	expected result.			
User-Only Test Cases – Applicable to Users Only (Admins have no access)								
TC-NA-08	Verify the floating "Add" button navigates to the correct screen.	Positive test: Click the floating "Add" button and verify it navigates to the Add Accommodation screen.	N/A	Redirects to the Add Accommodation screen with empty form fields.	Same as expected result.	Pass	High	Ensure correct navigation to the Add Accommodation screen.
Module Name: Add Accommodations Screen Test Case Description: Tests the accommodation submission form, including image uploads, field validation, and map integration. Testing Objective: Ensure form inputs are validated, images upload correctly, and location selection works as expected. These test cases are applied to the Add Grocery Stores Screen, Add Eateries Screen, Add Activity Places Screen and Add Transportation Screen.								
User-Only Test Cases – Applicable to Users Only (Admins have no access)								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-AA-01	Validate image upload limits	Upload 9 and 10 images	Images: 9 files, 10 files	9 images upload successful; 10 images show max error	Same as expected	Pass	High	Boundary and overflow validation
TC-AA-02	Verify picker selections and key inputs	Select Property Type, Configuration, Condition; Enter Property Name and Size	Property Type: Apartment; Configuration: 4 rooms 3 bathrooms; Property Condition: Fully Furnished; Property Name: valid & blank; Size: valid & blank	Pickers and condition selections stored correctly; Required fields validated with errors; Property size appends "sq ft"; Only one condition selectable	Same as expected	Pass	Medium	Covers dropdowns, radio logic, and required validations
TC-AA-03	Validate Rental Price and Parking Spaces inputs	Enter valid and invalid data	Rental Price: "850", "ABC", "-500"; Parking: "2", "2.5", "Two"	Valid inputs accepted and formatted; invalid inputs rejected with errors	Same as expected	Pass	High	Numeric validation, no negatives or decimals
TC-AA-04	Validate Owner Contact input format	Enter valid and invalid contact numbers	Contact: "012-3456789", "012-ABCD"	Valid Malaysian number accepted and formatted; invalid rejected	Same as expected	Pass	High	Strict phone format enforcement
TC-AA-05	Validate Amenities, Property Condition, and Free Text	Select multiple amenities, enter text for Other Amenities, select furnishing	Amenities: ["Gym", "Malls"]; Other Amenities: "Swimming Pool", 201+ characters; Property Condition: Fully Furnished	Multiple amenities stored; valid free text stored; overly long text rejected; only one condition selectable at a time	Same as expected	Pass	Medium	Tests mutual exclusivity, length limits, and multi-select
TC-AA-	Validate map and navigation	Drop pin, use search, and test back navigation	Coordinates: valid lat/long; Search:	Pin saves coordinates; search	Same as expected	Pass	High/Medium	Covers geolocation and navigation stack

06	functionality		"University of Malaya"; Back button	fetches location; back navigates to Add Accommodation screen				
TC-AA-07	Validate form submission with required fields	Submit with missing required fields; submit complete form	Property Name blank; Property Type/configuration missing; All fields valid	Missing field triggers error and blocks submission; Valid form redirects to preview	Same as expected	Pass	Critical	Final validation and E2E test coverage
Module Name: Property Details Screen Test Case Description: Validates the display of property details, favorite toggling, map interactions, and sharing functionality. Testing Objective: Ensure property information renders correctly, map views toggle, and sharing features work.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-PD-01	Render property details with or without images	Load property screen with and without images	Property: Unicentral Apartment; Property with no image data	All key fields displayed; images swipeable if available; placeholder if missing	Same as expected	Pass	High	Validates content rendering with and without image assets.
TC-PD-02	Verify user contribution	Tap on user avatar	User avatar	Navigates to user profile	Same as expected	Pass	High	Tests contributor profile access and external app sharing functionality
TC-PD-03	Validate map view modes and interaction	Tap “View Location”; change map view modes	Property with coordinates; View Types: Standard, Hybrid, Terrain, Satellite	Map opens with pin; each map view displays correctly with smooth transition	Same as expected	Pass	Medium	Confirms all map types render correctly and support smooth switching.
TC-PD-04	Test navigation from map and property screen	Press back from map; tap top-left back icon	Current screens: Map, Property Details	Navigates back to correct previous screens (Property Details / Nearby Listings)	Same as expected	Pass	Medium	Verifies back navigation logic and screen stack behavior.
User-Only Test Cases – Applicable to Users Only (Admins have no access)								
TC-PD-05	Toggle favorite button state	Tap heart icon repeatedly	Initial: Unfilled or Filled	Icon toggles between filled/unfilled; updates UI and data store	Same as expected	Pass	Medium	Ensures UI updates and state persistence for favorite functionality
TC-PD-06	Verify Whatsapp sharing flow	Tap on WhatsApp icon	Accommodation object with location and link	Opens WhatsApp with pre-filled accommodation info	Same as expected	Pass	High	Tests external app sharing functionality
Admin-Only Test Cases – Applicable to Admins Only (Users have no access)								
TC-PD-07	Admin controls visibility	Login as Admin/Regular user	Admin/User roles	Admin sees no Edit/Delete,	Same as expected	Pass	High	Role-based UI

				favorite/share; User will see the opposite				
TC-PD-08	Admin edit inline save/cancel	Edit and save or cancel	Modified data	Changes persist or revert on cancel	Same as expected	Pass	High	Edit functionality
TC-PD-09	Admin delete confirmation	Tap Delete, confirm/cancel	Any property	Deletes on confirm, stays on cancel	Same as expected	Pass	High	Delete + confirm
TC-PD-10	Amenities list display	Property with amenities	Amenities data present	Amenities shown clearly	Same as expected	Pass	Medium	Amenities display
TC-PD-11	Route guidance link	Tap Route Guidance	Valid location	External map app opens with directions	Same as expected	Pass	High	External map launch
Module Name: Nearby Eateries Screen Test Case Description: Tests eatery listing rendering, search/filter logic, and navigation to contribution forms. Testing Objective: Verify eatery cards display correctly, search/filter works, and navigation flows function. Remarks: These test cases are applied to the Nearby Grocery Stores Screen, Nearby Activity Places Screen and Nearby Transportation Screen.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-NE-01	Render eatery cards (with/without images)	Load screen with eateries with full and missing data	Eatery: Gardens Cafe; One with missing image	Name, address, rating, phone shown; images swipe or show placeholder	Same as expected	Pass	High	Covers card UI, image fallback
TC-NE-02	Search and filter eateries	Search and apply cuisine filters	Keywords: “Gardens Cafe”, “Invalid123”; Filters: “Western”, “Mexican”	Matching entries show results; unmatched show “No results” or empty view	Same as expected	Pass	Medium	Validates search and filter logic
TC-NE-03	Validate map view modes and interaction	Tap “View Location”; change map view modes	Property with coordinates; View Types: Standard, Hybrid, Terrain, Satellite	Map opens with pin; each map view displays correctly with smooth transition	Same as expected	Pass	Medium	Confirms all map types render correctly and support smooth switching.
TC-NE-04	Verify user contribution	Tap on user avatar	User avatar	Navigates to user profile	Same as expected	Pass	High	Tests contributor profile access and external app sharing functionality
User-Only Test Cases – Applicable to Users Only (Admins have no access)								
TC-NE-05	Toggle favorites	Tap heart icon repeatedly	Initial: Filled or Unfilled	Favorite icon toggles and updates state	Same as expected	Pass	Medium	Verifies favorite toggle behavior
TC-NE-06	Verify Whatsapp sharing flow	Tap on WhatsApp icon	Eatery object with location and link	Opens WhatsApp with pre-filled accommodation info	Same as expected	Pass	High	Tests external app sharing functionality
TC-NE-07	Verify the floating “Add” button navigates	Click the floating “Add” button and verify it navigates to the Add Eateries screen.	N/A	Redirects to the Add Eateries screen with empty form fields.	Same as expected result.	Pass	High	Ensure correct navigation to the Add Eateries screen.

	to the correct screen.							
Admin-Only Test Cases – Applicable to Admins Only (Users have no access)								
TC-NE-08	Admin Edit/Delete button visibility	Login as Admin/Regular user	Admin and user credentials	Admin sees Edit/Delete; user does not	Same as expected	Pass	High	Role-based UI
TC-NE-09	Admin edit inline save and validation	Edit eatery, save with valid/invalid data	Modified data	Saves if valid; errors if invalid	Same as expected	Pass	High	Edit + validation
TC-NE-10	Admin delete with confirmation	Delete eatery, confirm or cancel	Any eatery listing	Deletes on confirm, retains on cancel	Same as expected	Pass	Medium	Delete + cancel
Module Name: Event Calendar Screen Test Case Description: Tests university event filtering, calendar dot markers, and navigation to event details. Testing Objective: Verify events display correctly based on university selection and navigation to details works.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-EC-01	Ensure UI elements render correctly	Positive test: Launch the application, login, and navigate to "Home" tab.	Registered email, valid password	UI elements such as the university dropdown, calendar, and personal events section render correctly.	Same as expected	Pass	High	Verify layout consistency across devices.
TC-EC-02	Validate university drop-down displays universities	Positive test: Tap on the university drop-down.	Valid university list	Drop-down lists all available universities (e.g., UNIMAS, UM, UTM).	Same as expected	Pass	Medium	Check for smooth dropdown animation.
TC-EC-03	Ensure calendar updates based on university selection	Positive test: Select a university from the drop-down, then select other universities.	University name	Calendar dots only appear for the selected university's events (e.g., UNIMAS events).	Same as expected	Pass	High	Verify no unrelated events appear.
TC-EC-04	Validate calendar dots when no university is selected	Positive test: Observe the calendar after selecting no university.	No university selected	No event dots appear when no university is selected.	Same as expected	Pass	Medium	
TC-EC-05	Validate tapping a date navigates to event activity	Positive test: Tap on a date with a dot and one without a dot.	Date with or without event dot	Navigates to University Event Activity screen with details for dotted dates.	Same as expected	Pass	High	Ensure correct event data is passed.
Shared Test Cases – Applicable to Both Users & Admins								
TC-	Validate tapping	Positive test:	Personal event card	Navigates to	Same as	Pass	High	Ensure correct event data is passed.

EC-06	on a personal event navigates to the event activity	Tap on an event card under the Personal Events section.		Personal Event Activity screen with event details.	expected			
Module Name: University Event Screen Test Case Description: Validates the toggle between university and personal events and event card rendering. Testing Objective: Ensure events display correctly and toggling between event types works.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-UE-01	Ensure all UI elements render correctly	Positive test: Verify that all UI elements are displayed on the University Event screen.	Registered email, valid password	UI elements such as toggle switch, event cards, and details (name, description, date, duration, address) are displayed correctly.	Same as expected	Pass	High	Verify layout consistency.
TC-UE-02	Ensure event cards display correct university event details	Positive test: Check event card details for a UNIMAS seminar.	Event card for UNIMAS seminar	Displays correct event details (name, description, date, duration, address).	Same as expected	Pass	High	Verify data accuracy from backend.
TC-UE-03	Validate tapping event card navigates to University Event Details	Positive test: Tap on a University Event card for a seminar event.	Event card for UNIMAS seminar	Navigates to the University Event Details screen with full event info.	Same as expected	Pass	High	Ensure correct event ID is passed.
TC-UE-04	Validate multiple event cards for the same date	Positive test: Check for multiple event cards for the same date.	3 UNIMAS events scheduled on the same date	All event cards for the same date are displayed with correct details.	Same as expected	Pass	High	Test scrolling for many events.
TC-UE-05	Validate back button invokes proper navigation	Positive test: Tap on arrow back icon on the top left of the screen.	Current screen: University Event Screen	App navigates back to Event Calendar Screen with the same university selected.	Same as expected	Pass	Medium	
User-Only Test Cases – Applicable to Users Only (Admins have no access)								
TC-UE-06	Validate toggle switch between University and Personal Events	Positive test: Toggle to Personal Events and then back to University Events.	Toggle switch	Displays personal events when toggled; reverts to university events when toggled back.	Same as expected	Pass	High	Ensure smooth toggle animation.
Admin-Only Test Cases – Applicable to Admins Only (Users have no access)								
TC-UE-07	Validate toggle and UI state for roles	Log in as admin/user, view screen	Admin and normal user accounts	Admin sees "Add Event", no toggle; user sees toggle only	Same as expected	Pass	High	Role-based UI
TC-	Validate event	Admin adds a new event	Valid event form	Event saved and	Same as	Pass	Critical	Creation + refresh

UE-08	creation and list update		data	shown in list without reload	expected			
TC-UE-09	Validate event form validation	Submit form with missing fields	Incomplete event data	Shows validation errors, blocks save	Same as expected	Pass	Medium	Form validation
TC-UE-10	Validate text overflow and layout	Create/view event with long text	Event with long name/description	Text wraps or truncates cleanly	Same as expected	Pass	Low	UI layout resilience
TC-UE-11	Validate back navigation behavior	Navigate back from detail/activity	Event screen open	Returns with correct university selected	Same as expected	Pass	Medium	Nav state retention
Module Name: Settings Screen Test Case Description: Tests settings UI, dark mode toggle, profile updates, and navigation to sub-screens. Testing Objective: Verify settings options work correctly and navigation flows function.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-SET-01	Validate Settings screen UI and Dark Mode	Load Settings screen and toggle Dark Mode	Registered email, valid password	All UI elements shown; Dark Mode toggles correctly	Same as expected	Pass	High	Checks layout and theme toggle
TC-SET-02	Profile Settings updates	Update username, email, and password; test back navigation	Username: "TestUser", Email, Password	Data updates reflect in app; back returns to Settings	Same as expected	Pass	High	Validates account info changes
TC-SET-03	View Terms of Service and return	Tap “Terms of Service”; press back	N/A	Opens Terms page; back returns to Settings	Same as expected	Pass	Medium	Verifies static content view
TC-SET-04	View About Us and return	Tap “About Us”; press back	N/A	Opens About Us page; back returns to Settings	Same as expected	Pass	Medium	Confirms info screen access
TC-SET-05	Logout functionality	Tap “Log Out” and verify redirection	N/A	User is logged out; redirected to login screen	Same as expected	Pass	Critical	Confirms session termination
User-Only Test Cases – Applicable to Users Only (Admins have no access)								
TC-SET-06	Navigation to My Ads and Favorite Ads	Tap "My Ads" and "Favorite Ads" and test back navigation	N/A	Navigates to correct screens and back to Settings	Same as expected	Pass	High	Checks layout and theme toggle
TC-SET-07	Feedback flow and return navigation	Tap "Send Feedback", submit message, tap back	Feedback: "Unifo is a great app"	Navigates to feedback form; returns to Settings after back	Same as expected	Pass	Medium	Validates account info changes
Admin-Only Test Cases – Applicable to Admins Only (Users have no access)								
TC-SET-08	Validate Manage Users (list, delete, cancel delete)	Open Manage Users; try deleting & canceling	Admin account	Users listed; deletion works or cancels	Same as expected	Pass	Critical	Confirms info screen access
TC-	Validate	Open Manage Feedback;	Admin account	Feedback	Same as	Pass	Medium	Confirms session termination

SET-09	Manage Feedback (list, resolve, delete, cancel)	resolve/delete/cancel		updated/removed or retained	expected			
--------	---	-----------------------	--	-----------------------------	----------	--	--	--

Integration Testing

Module Name: Login and Sign Up Screen Test Case Description: Validates user registration, login, and password reset functionality, including input validation for weak passwords, empty fields, and incorrect credentials. Testing Objective: Ensure secure authentication by verifying correct handling of valid and invalid inputs during sign-up, login, and password recovery.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-LS-01	Verify that the application displays dynamic landing screen before login interface on iOS device.	1. Launch the Unifo+ application. 2. Observe the initial screen display.	N/A	Dynamic application landing screen is displayed before secure login interface.	Same as the expected result	Pass	Low	iOS-specific behavior.
TC-LS-02	Verify that the system does not allow password reset for unregistered emails or usernames. The system should return an appropriate error message "Email not found."	1. Launch the Unifo+ application. 2. Click "Forgot Password?". 3. Enter an unregistered email or username. 4. Enter a valid password. 5. Re-enter the same password in the confirmation field. 6. Click "Reset Password".	Negative test: Email: notregistereduser@gmail.com	Password reset fails with 'Email not found' message.	Same as the expected result	Pass	Medium	Validation for unregistered email in the password reset process.
TC-LS-03	Verify that the system prevents password reset when a valid registered email but a password with a wrong format are used.	1. Launch the Unifo+ application. 2. Click "Forgot Password?". 3. Enter a registered email or username. 4. Enter a password in the wrong format. 5. Re-enter the same password in the confirmation field. 6. Click "Reset Password".	Positive test: Email: registereduser@gmail.com	Password reset fails with error message about password format.	Same as the expected result	Pass	Medium	Password format should include: one uppercase letter, one number, and one special character.
TC-LS-04	Verify that the system resets the password using a valid registered email and a valid password. A reset link or temporary password is sent via email.	1. Launch the Unifo+ application. 2. Click "Forgot Password?". 3. Enter a registered email or username. 4. Enter a valid password. 5. Re-enter the same password in the confirmation field. 6. Click "Reset Password".	Positive test: Email: registereduser@gmail.com	Reset email sent successfully, user is able to set new password and log in.	Same as the expected result	Pass	Medium	Email delivery and reset link must be verified.
TC-LS-05	Confirm that users can log in with correct credentials. The system grants access and loads the user/admin interface.	1. Launch the Unifo+ application. 2. Enter a registered email. 3. Enter a valid password. 4. Re-enter the same password in the confirmation field.	Positive test: Email: registereduser@gmail.com Password:	Login successful, user directed to event calendar.	Same as the expected result	Pass	High	Validates successful authentication.

		5. Click "Login"	Student@123					
TC-LS-06	Verify that by logging in with incorrect email, the system shows error message: 'Invalid email or password.'	1. Launch the Unifo+ application. 2. Enter an unregistered email. 3. Enter a valid password. 4. Re-enter the same password in the confirmation field. 5. Click "Login".	Negative test: Email: unregistereduser@gmail.com Password: Student@123	Login fails with error message.	Same with the expected result	Pass	Medium	Ensures incorrect credentials are properly handled
TC-LP-07	Verify that the system rejects login when the password is incorrect. The system shows error message: 'Invalid email or password.'	1. Launch the Unifo+ application. 2. Enter a registered email. 3. Enter an invalid password. 4. Re-enter the same password in the confirmation field. 5. Click "Login".	Negative test: Email: student@gmail.com Password: wrongpassword@1234	Login fails with error message.	Same with the expected result	Pass	Medium	Password format should include: one uppercase letter, one number, and one special character.
TC-LP-08	Ensure the system prevents login if mandatory fields are left blank. The system should alert: 'All fields must be entered.'	1. Launch the Unifo+ application. 2. Leave only one of the fields empty OR both fields. 4. Click "Login".	Negative test: Email: student@gmail.com Password: (empty) OR Email: (empty) Password: Student@123	Login fails with alert about incomplete fields.	Same with the expected result	Pass	Medium	All registration fields are mandatory.
User-Only Test Cases – Applicable to Users Only (Admins have no access)								
TC-LS-09	Verify that the system rejects registration when the password does not meet complexity requirements.	1. Launch the Unifo+ application. 2. Click on "Don't have an account? Sign up." 3. Enter a valid username. 4. Enter a valid email address. 5. Enter a password in the wrong format. 6. Re-enter the same password in the confirmation field. 7. Click Sign Up.	Negative test: Username: student123 Email: student@gmail.com Password: testpass Confirm Password: testpass	Registration fails with error message about password format.	Same as the expected result	Pass	Medium	Password format should include: one uppercase letter, one number, and one special character.
TC-LS-10	Ensure the system prevents registration if mandatory fields are left blank. The system should alert: 'All fields must be entered.'	1. Launch the Unifo+ application. 2. Click on "Don't have an account? Sign up." 3. Leave all fields empty OR enter only one or two fields. 4. Click Sign Up.	Negative test: Username: (empty) Email: student@gmail.com Password: (empty) Confirm Password: (empty)	Registration fails with alert about incomplete fields.	Same as the expected result	Pass	Medium	All registration fields are mandatory.
Admin-Only Test Cases – Applicable to Admins Only (Users have no access)								
TC-LS-11	Verify that admin accounts are pre-configured and cannot self-register.	1. Launch the Unifo+ application. 2. Click on "Switch to Admin Login" 3. Look for "Admin Sign Up" option.	N/A	No sign-up option available for admin accounts. Only super admin can create admin accounts.	Same as the expected result	Pass	Medium	Admin account creation restriction.
Module Name: Public Universities Activity								
Test case description: This test will determine whether the public university list screen and its functionalities work correctly.								
Testing objective: To test and validate the public universities listing, search function, expandable branch lists, and navigation within the application.								
Shared Test Cases – Applicable to Both Users & Admins								

Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-UA-01	Verify universities are displayed in card layout with logos and descriptions.	1. Launch the Unifo+ application. 2. Enter a registered email. 3. Enter a valid password. 4. Re-enter the same password in the confirmation field. 5. Click "Login" 6. Navigate to "Universities" tab.	N/A	All universities are shown as cards with names, logos, and brief descriptions.	Same with the expected result	Pass	High	Ensure images load without delays.
TC-UA-02	Confirm search bar filters universities by name.	Pre-quisites: Performed TC-UA-01 1. Type "Universiti Malaysia Sarawak" in the search bar. 2. Press Enter on device keyboard or click the magnifier icon inside the search bar. 3. Repeat steps 1 and 2 for all other public universities in the list.	Positive test: Search Query: Universiti Malaysia Sarawak	Only Universiti Malaysia Sarawak appear in results.	Same with the expected result	Pass	High	Test partial queries (e.g., "UNIMAS").
TC-UA-03	Confirm search bar returns no results when an unknown university name is entered.	Pre-quisites: Performed TC-UA-01 1. Type "Sunway University" in the search bar. 2. Press Enter on device keyboard or click the magnifier icon inside the search bar.	Positive test: Search Query: Sunway University	No results should appear.	Same with the expected result	Pass	High	
TC-UA-04	Verify expanding/collapsing branch campuses for universities with chevron icons (e.g. UiTM).	Pre-quisites: Performed TC-UA-01 1. Type "Universiti Teknologi Mara (UiTM)" in the search bar. 2. Press Enter on device keyboard or click the magnifier icon inside the search bar. 3. Tap the chevron-down icon on UiTM's card.	Positive test: University: Universiti Teknologi Mara (UiTM)	Card expands to show branch campuses (e.g., UiTM Sarawak Branch, UiTM Penang Branch).	Same with the expected result	Pass	Medium	Check if branches have clickable details.
TC-UA-05	Ensure tapping a university card redirects to its details page.	Pre-quisites: Performed TC-UA-01 1. Tap on "Universiti Malaya" card. 2. Repeat step 1 for all other public universities in the list.	Positive test: University = Universiti Malaya	Redirected to University Details screen.	Same with the expected result	Pass	Critical	Verify back navigation works.
TC-UA-06	Validate bottom navigation bar functionality.	Pre-quisites: Performed TC-UA-01 1. Tap each icon "Dashboard", "Settings", and "Event Calendar".	Positive test: All navigation tabs	Each tab opens its respective screen.	Same with the expected result	Pass	High	Icons should highlight when active.
TC-UA-07	Verify that tapping the university card (excluding the chevron icon) does not expand branch campuses.	Pre-quisites: Performed TC-UA-01 1. Type "Universiti Teknologi Mara" in the search bar. 2. Press Enter on device keyboard or click the magnifier icon inside the search bar. 3. Tap any area on the "Universiti Teknologi Mara" card except the chevron-down icon.	Negative test: Action = tap card body only	Card remains collapsed.	Same with the expected result	Pass	Low	Icon action separation
Module Name: University Details Activity Test case description: This test checks if the University Details screen shows the correct information for the selected university and if all buttons and sections work properly. Testing objective: To make sure the details match the selected university and users can open the website, map, and other info sections without issues.								
Shared Test Cases – Applicable to Both Users & Admins								

Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-UD-01	Verify correct university name is shown upon selection	1. Launch the Unifo+ application. 2. Enter a registered email. 3. Enter a valid password. 4. Re-enter the same password in the confirmation field. 5. Click "Login" 6. Navigate to "Universities" tab. 7. Tap on "Universiti Malaysia Sarawak (UNIMAS)" card.	Positive test: University: UNIMAS	USM's name and logo are shown on the top of the details screen	Same with the expected result	Pass	High	
TC-UD-02	Ensure quick action button to open university website functions correctly	Pre-requisites: Performed TC-UD-01 1. Tap the "Website" button	Positive test: Website link from UNIMAS data	UNIMAS's official website opens in browser	Same with the expected result	Pass	High	Validate against university's official URL
TC-UD-03	Ensure "View on Map" button opens map correctly	Pre-requisites: Performed TC-UD-01 1. Tap the "Campus Map" button	Positive test: Location link: UNIMAS, Kota Samarahan.	Integrated map opens centered on UNIMAS's campus	Same with the expected result	Pass	High	
TC-UD-04	Confirm navigation sections (About University, Courses, Facilities) are clickable and load respective content	Pre-requisites: Performed TC-UD-01 1. Tap each quick action button: About University, Courses, Facilities	N/A	Each quick action button opens its relevant sub-screen	Same with the expected result	Pass	High	
TC-UD-05	Validate switching between universities updates the details accordingly	Pre-requisites: Performed TC-UD-01 1. Repeat TC-UD-02, TC-UD-03, TC-UD-04 for all other public universities in the "Universities" tab.	Positive test: Test all universities' detail screens.	Each university detail screen updates info correctly.	Same with the expected result	Pass	Critical	No cache overlap or incorrect data from previous screen
TC-UD-06	Verify that tapping browser "back" returns to the public universities screen	Pre-requisites: Performed TC-UD-01 1. Tap on the "arrow-back" icon on the top left of the screen.	N/A	User returns to Public Universities screen	Same with the expected result	Pass	Medium	
Module Name: Dashboard Screen Test case description: This test verifies the functionality of displaying and selecting faculties from a university. Testing objective: To ensure users can view and navigate to faculty-specific courses correctly. Remarks: Ensure that facility data is not shared across universities, and fallback messages are shown when no data is available.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-DA-01	Verify the default state of the university drop-down.	1. Launch the Unifo+ application. 2. Enter a registered email. 3. Enter a valid password. 4. Re-enter the same password in the confirmation	N/A	Drop-down displays "Select University" as the default placeholder.	Same with the expected result	Pass	Low	Ensures clear user prompting.

		field. 5. Click "Login" 6. Navigate to "Dashboard" tab.						
TC-DA-02	Verify if an alert displays, when accessing features without selecting a university.	Pre-quisites: Performed TC-DA-01 1. Keep drop-down as "Select University" without selecting a university. 2. Click on all feature buttons (e.g., Nearby Accommodation, Nearby Eateries).	Positive test: Selected Uni: None	System shows alert: "Please select a university first."	Same with the expected result	Pass	Medium	Critical for guiding user actions.
TC-DA-03	Verify that the university drop-down is complete (all public universities included).	Pre-quisites: Performed TC-DA-01 1. Click on the university drop-down. 2. Verify the list contains all supported public universities.	Positive test: Expected Unis: UM, UTM, UKM, USM, etc.	Dropdown shows all predefined universities.	Same with the expected result	Pass	Medium	List must be exhaustive.
TC-DA-04	Verify the grid layout and button functionality for each feature.	Pre-quisites: Performed TC-DA-01 1. Click on the university drop-down. 2. Select on a university. 3. Click on all feature buttons (e.g., Nearby Accommodation, Nearby Eateries).	Positive test: Selected Uni: Any universities from the list	Each button redirects to the correct subpage with relevant data.	Same with the expected result	Pass	High	All buttons must be functional.
TC-DA-05	Verify dynamic content updates based on selected university.	Pre-quisites: Performed TC-DA-01 1. Click on the university drop-down. 2. Select on "Universiti Malaya". 3. Click on all feature buttons (e.g., Nearby Accommodation, Nearby Eateries). 4. Repeat steps 1 - 3 for all other public universities in the drop-down.	Positive test: Selected Uni: Universiti Malaya	Dashboard updates dynamically to show UM-specific content (e.g., UM eateries, accommodations).	Same with the expected result	Pass	High	Content must match the selected university.
Module Name: Event Calendar Screen Test case description: This test ensures the Event Calendar screen functions correctly, including university event filtering, dynamic calendar updates, dot markers, and personal event retrieval. Testing objective: To validate the rendering, filtering, dynamic updating, and layered interaction of university and personal events. Remarks:								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-EC-01	Ensure all UI elements render correctly.	1. Launch the Unifo+ application. 2. Enter a registered email. 3. Enter a valid password. 4. Re-enter the same password in the confirmation field. 5. Click "Login" 6. Navigate to "Home" tab.	Positive Test: Valid login	University dropdown, calendar, and personal events section are displayed.	Same as expected	Pass	High	Verify layout consistency across devices.
TC-EC-02	Validate university drop-down displays available universities.	Pre-quisites: Performed TC-EC-01 1. Tap on the university drop-down.	Positive Test: Select Universiti Malaysia Sarawak (UNIMAS)	Drop-down lists all available universities (e.g., UNIMAS, UM, UTM).	Same as expected	Pass	Medium	Check for smooth dropdown animation.
TC-EC-03	Ensure calendar updates dynamically based on university selection.	Pre-quisites: Performed TC-EC-01 1. Tap on the university drop-down. 2. Select on a university. 3. Repeat steps 1 and 2 for other universities.	Positive Test: UNIMAS selected	Dots appear only for UNIMAS events (e.g., April 20 event).	Same as expected	Pass	High	Verify no unrelated events appear.

TC-EC-04	Validate calendar dots will not appear when no university is selected.	Pre-quisites: Performed TC-EC-01 1. Observe the event calendar. 2. Navigate to other months.	Positive Test: No university selected	No university event dots are displayed.	Same as expected	Pass	Medium	
TC-EC-05	Validate tapping a date with or without a dot navigates to University Event Activity.	Pre-quisites: Performed TC-EC-01 1. Tap on the university drop-down. 2. Select on a university. 3. Click on a date with a dot. 4. Repeat steps 2 and 3 for dates with no dots.	Positive Test: April 20 UNIMAS event	Navigates to University Event Activity screen with event details (for dotted dates).	Same as expected	Pass	High	Ensure correct event data is passed.
User-Only Test Cases – Applicable to Users Only (Admins have no access)								
TC-EC-06	Verify that no personal events are displayed for new users.	1. Launch the Unifo+ application. 2. Create a new user account 3. Enter the new registered email. 4. Enter the new valid password. 5. Re-enter the same password in the confirmation field. 6. Click “Login” 7. Navigate to "Home" tab. 8. Observe Personal Events section.	Positive Test: Valid login	For new users, “No events added yet” will be displayed in the Personal Events section.	Same as expected	Pass	Medium	
TC-EC-07	Validate tapping on a personal event navigates to Personal Event Activity.	Pre-quisites: Performed TC-EC-01 1. Click on an event card, under the Personal Event section.	Positive Test: Personal Event	Navigates to Personal Event Activity screen with event details.	Same as expected	Pass	High	Ensure correct event data is passed.
Module Name: University Event Screen Test case description: This test ensures that the University Event Activity correctly displays events, handles toggle functionality, and loads data accurately. Testing objective: To validate correct rendering, toggle action, event data loading, and navigation.								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-UE-01	Ensure all UI elements render correctly.	1. Launch the Unifo+ application. 2. Enter a registered email. 3. Enter a valid password. 4. Re-enter the same password in the confirmation field. 5. Click “Login” 6. Navigate to "Home" tab. 7. Tap on the university drop-down. 8. Select on University Malaysia Sarawak (UNIMAS). 9. Click on a date with a dot.	Positive Test: UNIMAS event	Toggle switch, event cards, and details (name, description, date, duration, address) are displayed.	Same as expected	Pass	High	Verify layout consistency.
TC-UE-02	Ensure event cards display correct university event details.	Pre-quisites: Performed TC-UE-01 1. Check event card details.	Positive Test: UNIMAS seminar	Displays name: "Seminar 2025", description, date: April 20, 2025, duration: 10:15 AM – 06:30 PM, address.	Same as expected	Pass	High	Verify data accuracy from backend.

TC-UE-03	Validate tapping event card navigates to University Event Details.	Pre-quisites: Performed TC-UE-01 3. Tap on an University Event card.	Positive Test: Seminar event	Navigates to University Event Details screen with full event info.	Same as expected	Pass	High	Ensure correct event ID is passed.
TC-UE-04	Ensure appropriate message displayed when there are no university events for the date.	Pre-quisites: Performed TC-UE-01 1. Launch the Unifo+ application. 2. Enter a registered email. 3. Enter a valid password. 4. Re-enter the same password in the confirmation field. 5. Click "Login" 6. Navigate to "Home" tab. 7. Tap on the university drop-down. 8. Select on University Malaysia Sarawak (UNIMAS). 9. Click on a date with no dot.	Negative Test: No events	Displays message: "No university events scheduled."	Same as expected	Pass	Medium	Verify UI remains clean.
TC-UE-05	Validate multiple event cards for same date.	Pre-quisites: Performed TC-UE-01 1. Leave one or more fields blank. 2. Click "Submit".	Positive Test: 3 UNIMAS events	All event cards are displayed with correct details.	Same as expected	Pass	High	Test scrolling for many events.
TC-UE-06	Ensure back button returns to Event Calendar.	Pre-quisites: Performed TC-UE-01 1. Tap on back button.	Positive Test: Back navigation	Returns to Event Calendar screen with same university selected.	Same as expected	Pass	Medium	Verify state retention.
User-Only Test Cases – Applicable to Users Only (Admins have no access)								
TC-UE-07	Validate toggle switch between University and Personal Events.	Pre-quisites: Performed TC-UE-01 1. Toggle to Personal Events. 2. Toggle back to University Events.	Positive Test: Switch toggle	Displays personal events when toggled; reverts to university events.	Same as expected	Pass	High	Ensure smooth toggle animation.
TC-UE-08	Ensure toggle to Personal Events with no personal events for new users.	Pre-quisites: Performed TC-UE-01 1. Toggle to Personal Events.	Negative Test: No personal events	Displays message: "No personal events found."	Same as expected	Pass	Medium	Ensure message is user-friendly.
Admin-Only Test Cases – Applicable to Admins Only (Users have no access)								
TC-UE-09	Verify the "Add University Event" button navigates to the Add University Event screen	1. Launch the Unifo+ application. 2. Enter a registered email. 3. Enter a valid password. 4. Re-enter the same password in the confirmation field. 5. Click "Login" 6. Navigate to "Home" tab. 7. Tap on the university drop-down. 8. Select on University Malaysia Sarawak (UNIMAS). 9. Click on a date with a dot. 10. Tap on the 'Add University Event' button.	N/A	- Redirects to the Add University Event screen - Form fields are empty and ready for new input.	Same as the expected result	Pass	High	

Module Name: Settings Screen Test case description: This test ensures the Event Detail screen displays comprehensive event data and enables location-based navigation via an interactive map. Testing objective: To validate the detail layout, mapping feature, data rendering, and travel estimation tools. Remarks:								
Shared Test Cases – Applicable to Both Users & Admins								
Test case ID	Test scenario	Test steps	Test data	Expected result	Actual result	Status (Pass/Fail)	Severity failure (Low/Medium/High/Critical)	Notes
TC-SET-01	Ensure all UI elements render correctly.	1. Launch the Unifo+ application. 2. Enter a registered email. 3. Enter a valid password. 4. Re-enter the same password in the confirmation field. 5. Click “Login” 6. Navigate to "Settings" tab.	Positive Test: Valid login	Dark Mode toggle, My Ads, Favorite Ads, Profile Settings, Send Feedback, Terms of Service, About Us, and Log Out button are displayed.	Same as the expected result	Pass	High	Verify layout consistency.
TC-SET-02	Validate Dark Mode toggle functionality.	Pre-requisites: Performed TC-SET-01 1. Toggle Dark Mode on/off.	Positive Test: Toggle Dark Mode	App switches between light and dark themes.	Same as the expected result	Pass	High	Verify smooth theme transition.
TC-SET-03	Ensure Profile Settings allows username update.	Pre-requisites: Performed TC-SET-01 1. Tap Profile Settings. 2. Update username to “TestUser”. 3. Click “Update Profile”	Positive Test: Username: "TestUser"	Username is updated and reflected in app.	Same as the expected result	Pass	High	Verify backend sync.
TC-SET-04	Validate Profile Settings allows email update.	Pre-requisites: Performed TC-SET-01 1. Tap Profile Settings. 2. Update email to “ test@example.com ”. 3. Click “Update Profile”	Positive Test: Email: " test@example.com "	Email is updated; confirmation email sent.	Same as the expected result	Pass	High	Verify email validation.
TC-SET-05	Ensure Profile Settings allows password change.	Pre-requisites: Performed TC-SET-01 1. Tap Profile Settings. 2. Click “Change Password” 3. Enter current password 4. Enter a new password 5. Re-enter new password 6. Click “Update Password” 7. Perform TC-SET-01 again.	Positive Test: New Password: "NewPass12345"	New password is updated and reflected in app.	Same as the expected result	Pass	High	Verify password strength check.
TC-SET-06	Ensure Terms of Service navigation.	Pre-requisites: Performed TC-SET-01 1. Tap Terms of Service.	Positive Test: Terms of Service	Navigates to Terms of Service page.	Same as the expected result	Pass	Medium	Verify content loads correctly.
TC-SET-07	Validate About Us navigation.	Pre-requisites: Performed TC-SET-01 1. Tap About Us.	Positive Test: About Us	Navigates to About Us page.	Same as the expected result	Pass	Medium	Verify content loads correctly.
TC-SET-08	Ensure Log Out button functionality.	Pre-requisites: Performed TC-SET-01 1. Tap Log out. 2. Confirm Log out.	Positive Test: Log Out	User is logged out; redirects to login screen.	Same as the expected result	Pass	Critical	Verify session termination.
User-Only Test Cases – Applicable to Users Only (Admins have no access)								
TC-SET-09	Ensure My Ads button navigation for users.	Pre-requisites: Performed TC-SET-01 1. Tap My Ads.	Positive Test: My Ads	Navigates to My Ads screen.	Same as the expected result	Pass	High	Verify correct navigation for users.

TC-SET-10	Validate Favorite Ads button navigation for users.	Pre-quisites: Performed TC-SET-01 1. Tap Favorite Ads.	Positive Test: Favorite Ads	Navigates to Favorite Ads screen.	Same as the expected result	Pass	High	Verify correct navigation for users.
TC-SET-11	Validate Send Feedback navigation.	Pre-quisites: Performed TC-SET-01 1. Tap Send Feedback. 2. Write "Unifo is a great application" in the field. 3. Click "Send Feedback"	Positive Test: Send Feedback	Navigates to feedback form screen and feedback is able to be sent.	Same as the expected result	Pass	Medium	
Admin-Only Test Cases – Applicable to Admins Only (Users have no access)								
TC-SET-12	Ensure admin-specific UI elements render correctly.	1. Launch the Unifo+ application. 2. Click on "Switch to Admin Login" 3. Enter a registered email. 4. Enter a valid password. 5. Re-enter the same password in the confirmation field. 6. Click "Login" 7. Navigate to "Settings" tab.	Positive Test: Admin login	Manage Users and Manage Feedback sections are displayed. My Ads and Favorite Ads sections are NOT displayed.	Same as the expected result	Pass	High	Verify admin-specific layout excludes user sections.
TC-SET-13	Validate Manage Users section navigation and ensure the screen displays user list correctly.	Pre-quisites: Performed TC-SET-01 1. Tap Manage Users. 2. View the user list in Manage Users screen.	Positive Test: User list display	Displays comprehensive list of registered users with relevant details (username, email, registration date, status).	Same as the expected result	Pass	High	Verify user data display accuracy.
TC-SET-14	Validate user account deletion functionality.	Pre-quisites: Performed TC-SET-01 1. Tap Manage Users. 2. Select a test user account. 3. Tap delete/remove user option. 4. Confirm deletion.	Positive Test: Delete violating user	User account is successfully deleted from the system.	Same as the expected result	Pass	Critical	Verify proper user removal and data cleanup.
TC-SET-15	Validate Manage Feedback section navigation and ensure feedback list displays correctly for admin review.	Pre-quisites: Performed TC-SET-01 1. Tap Manage Feedback. 2. View feedback list in Manage Feedback screen.	Positive Test: Feedback display	Displays list of user feedback with details (feedback text, user, date, status).	Same as the expected result	Pass	High	Verify feedback data display and organization.
TC-SET-16	Validate feedback review and resolution functionality.	Pre-quisites: Performed TC-SET-01 1. Select a feedback item. 2. Mark as "Resolved". 3. Save changes.	Positive Test: Resolve feedback	Feedback status is updated to "Resolved" and reflected in the system.	Same as the expected result	Pass	High	Verify feedback status management.
TC-SET-17	Validate feedback deletion functionality.	Pre-quisites: Performed TC-SET-01 1. Select a resolved feedback item. 2. Delete the feedback. 3. Confirm deletion.	Positive Test: Delete resolved feedback	Feedback is successfully removed from the system after resolution.	Same as the expected result	Pass	Medium	Verify proper feedback cleanup process.

Appendix D: Usability Test Plan

Scope	Unifo+ Mobile Application (iOS)
Purpose	To evaluate functionality, usability, and interface of Unifo+ for university students.
Schedule and Location	Anytime and anywhere from 12 th May 2025 until 18 th May 2025
Sessions	30 minutes for each session
Equipment	• Laptop or desktop computer any size • iOS device (iPhone/iPad) with Expo Go application installed. • Stable internet connection.
Respondent	10 students from any Malaysian public universities
Scenario	1. Registration & Login 2. Password Reset 3. University Exploration 4. Accommodation/Eateries/Grocery Stores/Activity Places/ Transportation Management 5. Event Calendar Usage 6. Settings & Profile Updates
Metrics	1. Did the user discover any problem(s)? 2. Are the functional buttons easily found? 3. Does the user completes the task without guidance? 4. Comment and suggestion
Quantitative metrics	1. Task completion rate 2. Time taken per task 3. Error frequency (critical/non-critical) 4. Subjective satisfaction (Likert scale) 5. User feedback/suggestions.

Appendix E: Usability Test Questionnaire

Scenario	Do you found any problem on completing this task?	Are the function button easily found?	I know how to do and where to do without any guidance	Comment/ Suggestion
Registration			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Login			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Password retrieval			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Search for a university			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
View University Details such as Website, Campus Map and About University			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Explore University Faculties, Courses or Facilities			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Add Accommodation			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Filter/Search Accommodation Listings			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
View			Strongly agree	

Accommodation Details and Accommodation Location			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Add Eateries			Strongly agree	
			Agree	
			Neutral	
			Disagree	
Filter/Search Eatery Listings			Strongly agree	
			Agree	
			Neutral	
			Disagree	
View Eatery Details and Eatery Location			Strongly agree	
			Agree	
			Neutral	
			Disagree	
Add Grocery Stores			Strongly agree	
			Agree	
			Neutral	
			Disagree	
Filter/Search Grocery Store Listings			Strongly agree	
			Agree	
			Neutral	
			Disagree	
View Grocery Store Details and Grocery Store Location			Strongly agree	
			Agree	
			Neutral	
			Disagree	
Add Activity Places			Strongly agree	
			Agree	
			Neutral	
			Disagree	
Filter/Search Activity Places Listings			Strongly agree	
			Agree	
			Neutral	
			Disagree	
View Activity Place Details and Activity Place Location			Strongly agree	
			Agree	
			Neutral	
			Disagree	

			Strongly disagree	
Add Transportation			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Filter/Search Transportation Listings			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
View Transportation Details and Transportation Location			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Add Personal Event			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Toggle University/ Personal Events			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
View University and Personal Events and Event Location			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Edit/Delete Added Listings through My Ads			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Update Profile Info			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Add listings to Favorite Ads			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Remove listings from Favorite			Strongly agree	
			Agree	

Ads			Neutral	
			Disagree	
			Strongly disagree	
Toggle Dark Mode			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	
Log Out			Strongly agree	
			Agree	
			Neutral	
			Disagree	
			Strongly disagree	

1. What do you think about the user interface of Unifo+?
2. What feature(s) you prefer the most and hate the most?
3. What other features do you think should be added into the Unifo+ application to make it better?
4. If you have just been offered to study in one of the public universities in Malaysia, would you use Unifo+ to guide you through your university life? Please give your reason.