Faculty of Computer Science and Information Technology

UNIMAS BUS TRACKING SYSTEM: ENHANCING CAMPUS TRANSPORTATION EFFICIENCY

Muhammad Ilham bin Shamsul Azhar

Bachelor of Computer Science with Honours

(Software Engineering)

2025

TABLE OF CONTENTS

Contents

CHAPTER 1 INTRODUCTION

1.1 Introduction

Efficient campus transportation is an essential and crucial component of university life, ensuring that students and staff can commute seamlessly between various locations and at a specific time as well so that they could comply to their schedule. Numerous universities and cities around Malaysia have successfully implemented similar systems, such as the Universiti Kebangsaan Malaysia bus tracking system, which uses GPS technology and an interactive map to offer accurate and timely updates. Additionally, the city of Kuala Lumpur also has integrated real-time bus tracking within its public transit system, improving user satisfaction and operational efficiency. At Universiti Malaysia Sarawak (UNIMAS), buses form the backbone of the internal transport system, catering to hundreds of users daily from various faculty and colleges. However, the current system lacks modern solutions to counter common transportation challenges.

The absence of real-time tracking leads to significant inefficiencies, including long waiting times, missed buses, and scheduling confusion. These issues disrupt daily routines, reduce productivity as a student, and negatively affect user satisfaction.

To address these challenges, this project proposes the development of a Bus Tracking System integrated into the UNIMASNow app which is UNIMAS official application. By utilizing GPS technology, the system will provide users with live updates, estimated time of arrivals (ETAs), and notifications. In the context of this project, the implementation of a real-time bus tracking system at UNIMAS is vital for enhancing the daily experience of students, staff, and visitors. By integrating technologies such as GPS and Internet of Things (IoT) devices, the project aims to create a seamless and efficient transportation system. GPS integration will ensure that buses are tracked in real-time, while IoT technologies will enable the system to provide dynamic updates based on traffic conditions and bus arrival times. This technological advancement will contribute to a smarter, more connected campus and improve the overall student experience.

1.2 Problem Statement

UNIMAS’s existing transportation system faces several pressing issues:

1. **Uncertainty in Bus Schedules**: Commuters or students are unaware of the exact arrival times of buses. Even though usually a bus schedule is given but most of the time the

arrival of buses is only on time or as said in schedule at the first stop only. This always leads to missed bus or reduce productivity.

2. **Lack of Real-Time Updates**: Everyday busses are always related to delays, sudden route changes, or cancellations are not communicated promptly. This is usually because there is very minimal way to track or communicate.
3. **Inefficient Resource Management**: The university lacks data-driven insights to optimize bus schedules and routes. Without data on bus usage, optimizing routes and schedules remains difficult.

These issues result in reduced user satisfaction and inefficiencies in transportation operations. A real-time tracking solution is critical to addressing these shortcomings and ensuring a reliable and efficient transport system at university.

1.3 Scope

The proposed project is scoped to the following areas:

1. **System Development**: Designing and implementing a GPS-enabled tracking system for campus buses.
2. **Integration**: Seamless incorporation of the tracking feature into the UNIMASNow app.
3. **Key Features**:
    - Real-time tracking of bus locations.
    - Notifications for Estimated Time Arrival (ETAs) and delays.
    - Visualization of routes and stops.
4. **Testing and Deployment**: Conducting extensive testing to ensure system reliability and deploying it for campus-wide use.

The system will focus on internal campus buses, catering primarily to students that lives in Timur and Kolej Rafflesia. The focus would be on 3 colleges which is Kolej Kenanga, Kolej Rafflesia and Kolej Seroja that use campus bus daily to commute.

1.4 Aims and Objectives

Aim

To improve the efficiency and reliability of UNIMAS's campus transportation system through real-time tracking and notification services on the UNIMAS official application

Objectives

- **Develop** a GPS-based system for tracking campus bus locations in real time.
- **Integrate** the tracking system into the existing UNIMASNow app to ensure accessibility.
- **Provide** real-time ETAs and notifications for route changes or delays.
- **Enable** data-driven decision-making by analyzing bus usage patterns.

1.5 Methodology

This project employs an iterative Agile development approach, comprising the following phases:

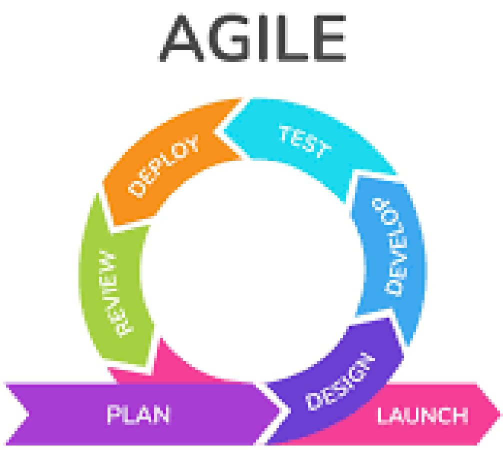

Figure 1.1

Phase 1: Requirement Gathering

- Conduct surveys and interviews with students and staff to identify transportation pain points or the specific challenges and difficulties that people face when using transportation at UNIMAS

Phase 2: System Design

Hardware

- Utilize the driver's smartphone, equipped with a GPS sensor, to send real-time location data to the server.

Software

- **Backend**
    - Develop a server to receive GPS data from the driver's smartphone.
    - Process the location data to calculate Estimated Time of Arrival (ETA) for each bus stop.
    - Store and manage route information, bus stop locations, and real-time tracking data in a centralized database.
    - Implement APIs for communication between the server and the UNIMASNow app.

- **Frontend**
    - Design and integrate a user-friendly interface within the UNIMASNow app.
    - Display real-time bus locations on an interactive map.
    - Provide ETAs and other relevant bus information (e.g., bus route, schedule).
    - Include a notification system to inform users about delays or route changes.

Phase 3: Development

- Implement GPS tracking features.
- Build the backend for data processing and notifications.
- Design the mobile app interface for seamless user interaction.

Phase 4: Testing

- Conduct functional testing to ensure GPS accuracy and reliability.

- Perform usability testing with a focus group to refine the app interface.

Phase 5: Deployment and Maintenance

- Launch the system on a trial basis and collect user feedback.
- Address issues and enhance the system based on feedback.

1.6 Significance of the Project

The project provides multiple benefits to UNIMAS, including:

- **Enhanced User Experience**: Real-time updates minimize waiting times and improve students' satisfaction. This will surely increase productivity among students as less time wasted on waiting or missing bus.
- **Operational Efficiency**: Data insights enable the university to optimize routes and schedules. This will always lead to student and university satisfaction.
- **Environmental Impact**: Promoting bus usage reduces carbon emissions and supports a greener campus.
- **Smart Campus Vision**: Aligns with UNIMAS's commitment to leveraging technology for improved infrastructure.

1.7 Project Schedule

The following timeline outlines key project phases mainly for FYP 1:

Phases	Suggested durations	Timelines
Brief Proposal Submission	2 weeks	15 October – 20 October 2024
Full Proposal Submission	3 weeks	21 October – 14 November 2024
Chapter 1 Submission	1 weeks	15 November –21 November 2024
Chapter 2 Submission	3 weeks	22 November – 13 December 2024
Chapter 3 Submission	3 weeks	14 December – 5 January 2025
Final Report Submission	1 weeks	6 January – 12 January 2025

1.8 Expected Outcomes

By the end of this project, the following deliverables will be achieved:

1. A fully functional bus tracking system integrated into the UNIMASNow app.
2. Real-time tracking and notification features for campus buses.
3. Analytical data reports on transportation patterns.
4. Improved commuting experience for students and staff.

Summary

The UNIMAS Bus Tracking System is a transformative initiative aimed at addressing critical transportation challenges on campus. By leveraging GPS technology and user-centric design, the project will enhance the commuting experience, improve operational efficiency, and support UNIMAS's smart campus aspirations. This proposal lays the groundwork for a reliable, innovative, and environmentally friendly transportation solution for the university community

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

This chapter explores existing transportation systems, emphasizing and going through their related functionalities, technologies, and relevance to the proposed UNIMAS Bus Tracking System. Reliable and efficient transportation systems are fundamental to modern infrastructure, particularly in environments that cater to large populations, which in this project refer to the UNIMAS campus students, staffs and visitors. As most of the systems address key challenges like uncertain schedules, lack of real-time updates, and inefficient route planning. The proposed UNIMAS Bus Tracking System seeks to solve these issues by leveraging necessary technologies such as GPS tracking, data analytics, and user compliance designs.

This chapter reviews three transportation systems that have achieved notable success in addressing similar challenges:

1. **MyRapid PULSE Bus System**, an app-based service offering real-time updates and route planning for public transportation in the Klang Valley.
2. **Grab**, a super app in Southeast Asia providing ride-hailing, food delivery, and payment solutions, renowned for its advanced tracking and dynamic optimization.

3. **Rapid Penang Intelligent Transport System (ITS)**, which integrates GPS-enabled tracking with mobile apps and physical infrastructure to enhance public bus services in Penang.

There are 3 main things to achieve in this literature review:

1. To analyze the key functionalities and technologies employed by these systems.
2. To identify their strengths and limitations, and how these systems impact users.
3. To extract insights that can inform the design and implementation of the UNIMAS Bus Tracking System.

By examining these systems, the review provides a great insight for developing a system similar to the needs of Universiti Malaysia Sarawak (UNIMAS). With hundreds of daily commuters relying on campus buses, an efficient, reliable, and user-friendly tracking system is vital for improving transportation operations and user satisfaction.

The analysis of the selected systems emphasizes features such as GPS-based real-time tracking, user-centered design, and intelligent route optimization. Additionally, it highlights potential challenges, such as traffic-related inaccuracies and scalability limitations. By drawing on these examples, the chapter demonstrates how technological advancements can be used efficiently to enhance campus transportation and contribute to UNIMAS's vision of a smarter, more connected campus.2.2 Review of Similar Existing Systems

2.2.1 MyRapid PULSE Bus System

Figure 2.1

Overview:

The MyRapid PULSE app, developed by Prasarana Malaysia Berhad, serves as an important part of the Klang Valley public transportation system. It connects users with real-time bus information, helping them navigate throughout the Rapid KL network that includes buses and rail transit systems. The system was introduced to modernize public transportation and improve commuter satisfaction by addressing issues like uncertain schedules and inefficient route planning.

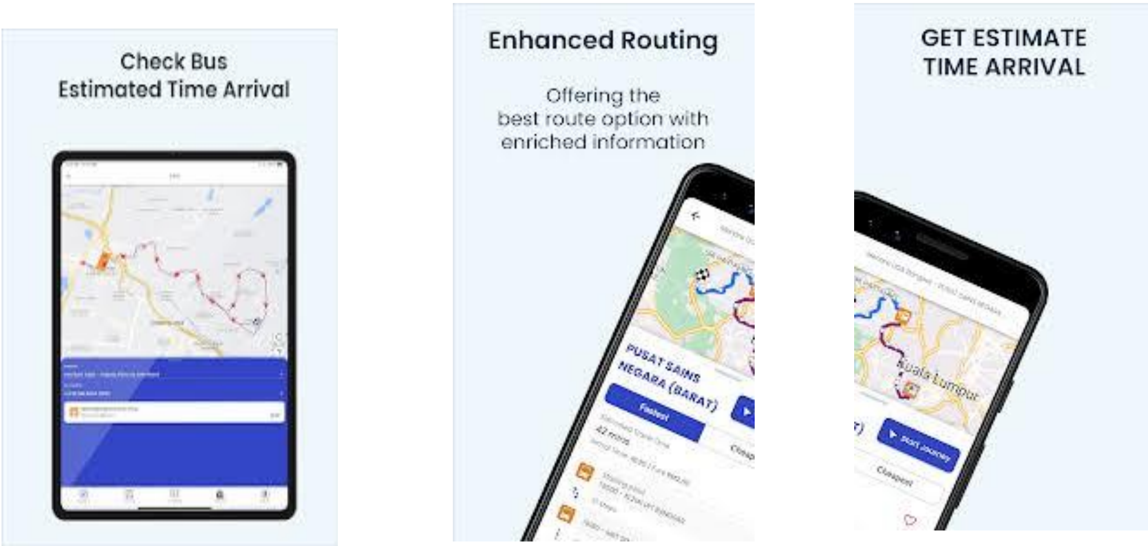

Figure 2.2

Features:

- **GPS-Based Tracking**: Live bus location updates enable users to view the exact position of buses in real time, reducing guesswork and waiting times at bus stops.
- **Notifications on Delays or Disruptions**: Alerts about delays or route changes ensure users stay informed and can adjust their plans accordingly.
- **Route Planning Tools**: Integration with Rapid KL's rail network allows users to plan multi-modal journeys efficiently, making it easier to connect between buses and trains.
- **User-Friendly App Interface**: The app’s intuitive design ensures accessibility for a wide range of users, from daily commuters to first-time visitors.

Strengths:

1. **Reduces Commuter Uncertainty**: Real-time updates on bus locations mitigate the frustration caused by unpredictable schedules. Users no longer have to waits for a long and uncertain time at bus stops, improving their overall experience.
2. **Enhanced Trip Planning**: By offering route suggestions and timing information, users can plan their journeys more effectively, which is especially beneficial for individuals balancing tight schedules, such as students and working professionals.
3. **Increased Public Transit Adoption**: The convenience provided by the app encourages more people to rely on public transportation, contributing to reduced traffic congestion and environmental benefits.

Limitations:

1. **Coverage Restrictions**: The system is currently limited to the Klang Valley area, excluding smaller towns and rural regions. This limitation creates a gap in accessibility for users outside metropolitan areas.

    - ***Implication for UNIMAS***: The challenge of limited coverage reflects the importance of designing a scalable system for UNIMAS, ensuring inclusivity for all possible campus bus routes.

2. **Impact of Traffic Conditions**: Arrival estimates can become inaccurate due to unforeseen traffic congestion, leading to dissatisfaction.

    - ***Implication for UNIMAS:*** Integrating features that account for real-time traffic data can improve estimate accuracy, ensuring a smoother experience for campus users.

Relevance to UNIMAS:

The MyRapid PULSE app is one of the system that shows how GPS tracking and real-time notifications can transform a transportation system. Applying these strengths to the UNIMAS Bus Tracking System can address current issues such as uncertain bus schedules and lack of real-time updates. For instance:

- **Improved Efficiency**: Students and staff can access accurate Estimated Time of Arrivals (ETAs), helping them plan their commutes effectively and reduce time wasted waiting at bus stops.
- **User-Centered Design**: A simple and yet informful interface, like the one in PULSE, can make the system accessible to all users, regardless of their technical proficiency.
- **Real-Time Notifications**: Alerts for delays or route changes are especially useful during peak hours or in case of disruptions on campus.

How Strengths and Limitations Shape the UNIMAS Project:
By learning from the strengths of MyRapid PULSE, the UNIMAS Bus Tracking System can:

- Incorporate GPS-enabled tracking for precise bus locations.
- Develop a notification system for delays and schedule changes.
- Prioritize a user-friendly app design to handle a diverse university population.

2.2.2 Grab (Transport and Food Delivery)

Figure 2.3

Overview:
Grab, launched in Malaysia in 2012, has evolved into a comprehensive super app serving millions of users across Southeast Asia. Initially focused on ride-hailing, Grab now provides food delivery, digital payments, and parcel delivery services, making it a vital part of urban mobility and logistics. By leveraging real-time tracking and AI-driven optimization, Grab connects users with drivers, restaurants, or delivery agents seamlessly, ensuring efficiency and convenience.

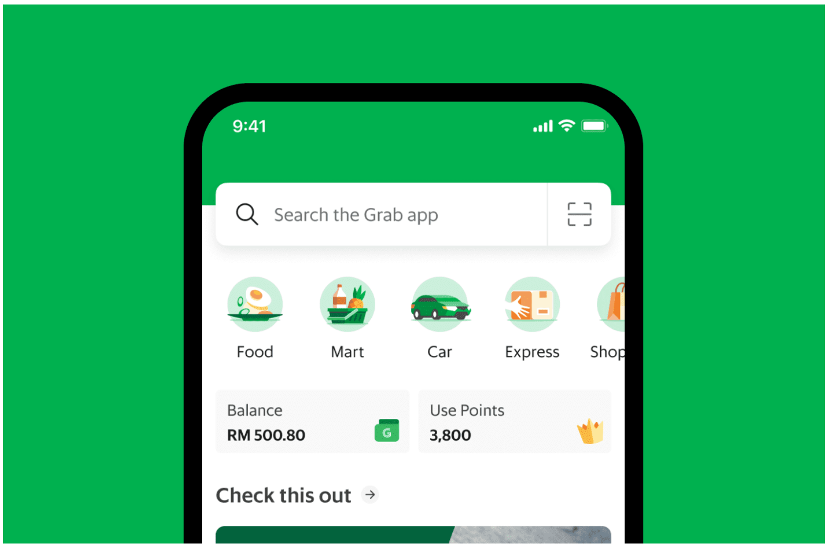

Figure 2.4

Features:

- **Real-Time Tracking**: Users can monitor vehicle locations and estimated times of arrival (ETA) in real time, enhancing transparency and trust.
- **Dynamic Pricing**: Pricing adapts based on demand and availability, balancing supply and demand dynamically.
- **Digital Payments**: Integrated GrabPay offers seamless transactions, reducing dependency on cash payments and enhancing user convenience.
- **Customer Feedback and Ratings**: Drivers and services are rated by users, creating accountability and maintaining high service standards.

Strengths:

1. **Scalability with Great Route Optimization**: Grab employs advanced algorithms to optimize routes for efficiency, reducing fuel consumption and travel time.

    - ***Impact on Users***: Shorter wait times and faster trips enhance user satisfaction.
    - ***Relevance to UNIMAS***: Route optimization can be adapted to create efficient bus schedules, ensuring minimal delays and smoother transportation for students and staff.

2. **Integrated Services**: By combining transportation, delivery, and payments, Grab creates a one-stop solution for users.

    - ***Impact on Users***: Convenience is maximized, as users can access multiple services on a single platform.
    - ***Relevance to UNIMAS:*** Integrating the bus tracking system with existing UNIMAS services (e.g., UNIMASNow) can streamline the user experience.

Limitations:

1. **Surge Pricing**: High demand during peak hours leads to increased fares, making services less affordable.

    - ***Impact on Users***: Can discourage usage during critical times.
    - ***Relevance to UNIMAS***: The UNIMAS system are already avoid ths issue as all of UNIMAS daily busses are free.

2. **Service Availability**: Driver proximity directly impacts response times, potentially causing delays.

    - ***Impact on Users***: Reliability desreases in areas with fewer available drivers.
    - ***Relevance to UNIMAS***: Ensuring sufficient bus availability and proper scheduling can address this limitation within a campus environment.

Relevance to UNIMAS:

Grab's use of real-time tracking, dynamic route optimization, and seamless payment integration demonstrates the potential for an efficient and user-friendly transport system. By adapting these strengths, the UNIMAS Bus Tracking System can ensure timely and optimized routes while offering an intuitive interface. Addressing Grab's limitations, such as affordability and availability, will be crucialto meeting the potential needs of a university campus such as UNIMAS

2.2.3 Rapid Penang Intelligent Transport System (ITS)

Figure 2.5

Overview:

Rapid Penang ITS was developed to modernize and improve the public bus services in Penang. By incorporating GPS tracking, digital displays, and mobile app features, ITS enhances the overall commuter experience. This initiative reflects Penang's commitment to using technology for better urban mobility while addressing challenges like schedule inconsistencies and limited route coverage.

Features:

- **Real-Time GPS Tracking**: Enables passengers to track buses live and access ETAs.
- **Digital Display Boards**: Bus stops feature screens showing arrival times, helping users plan their journeys better.
- **Mobile App Integration**: Provides routes, schedules, and real-time updates, empowering users with information at their fingertips.

Strengths:

1. **Enhanced User Confidence**: Accurate real-time updates reduce uncertainty about bus arrivals.

    - ***Impact on Users***: Passengers experience less frustration and wasted time at bus stops.
    - ***Relevance to UNIMAS***: Integrating real-time GPS tracking into the UNIMAS system can improve user trust and satisfaction.

2. **Integration of Physical and Digital Platforms**: Combining app-based tracking with digital boards ensures inclusivity for users who may not have access to smartphones.

    - ***Impact on Users***: Accessible for diverse demographics, including those unfamiliar with mobile apps.
    - ***Relevance to UNIMAS***: Installing digital displays at key locations on campus can complement app features, ensuring inclusivity.

Limitations:

1. **Traffic Congestion Impacts Accuracy**: Real-time updates may become less reliable during peak traffic periods.

    - ***Impact on Users***: Leads to occasional frustration and reduced confidence in the system.
    - ***Relevance to UNIMAS***: Incorporating predictive algorithms to account for traffic conditions can mitigate this issue and improve arrival time accuracy.

2. **Limited Geographic Coverage**: Services are concentrated in urban areas, leaving rural and less populated regions underserved.

    - ***Impact on Users***: Inaccessibility in remote areas reduces system effectiveness and reliability
    - ***Relevance to UNIMAS***: Ensuring all campus routes and all colleges are included in the system will prevent similar coverage issues.

Relevance to UNIMAS:
Rapid Penang ITS provides valuable insights into balancing digital and physical infrastructure for public transport systems. For the UNIMAS Bus Tracking System, adopting GPS tracking and real-time displays can greatly enhance campus mobility. By addressing challenges such as traffic-based inaccuracies and limited coverage, the system can offer a seamless and inclusive transportation experience tailored to the campus community.

2.3 Summary of Existing Systems

The analysis of the three existing systems reveals their unique strengths and limitations, offering valuable lessons for the development of the UNIMAS Bus Tracking System:

1. **MyRapid PULSE** emphasizes the importance of **GPS tracking** and **intuitive interfaces**, which enhance user experience by providing real-time updates and reducing commuter uncertainty. However, its **limited regional coverage** highlights the need for a scalable system that can adapt to the broader needs of a diverse population, such as the student and staff at UNIMAS.
2. **Grab** demonstrates **real-time optimization and scalability**, leveraging advanced algorithms for dynamic route planning and efficient operations. Its success in integrating multiple services showcases the potential of a user-friendly platform that centralizes essential functions. Nevertheless, **cost-related challenges**, such as surge pricing, and dependence on **driver availibility**, reveal the importance of affordability and consistent availability, which will be critical for UNIMAS.
3. **Rapid Penang ITS** shows **comprehensive integration** of both physical and digital platforms, combining real-time updates with digital displays for improved inclusivity. However, its **traffic-related inaccuracies** and **geographic limitations** underline the importance of accounting for dynamic traffic patterns and ensuring all campus routes are adequately covered.

By examining these systems, the UNIMAS Bus Tracking System can adopt features such as real-time GPS tracking, user-centered design interfaces, and intelligent route optimization while addressing challenges like scalability and accuracy to meet specific campus needs effectively.

2.3.1 Comparative Analysis

Features	MyRapid PULSE	Grab	Rapid Penang ITS	Current UNIMAS System	Proposed UNIMAS System
Real-Time GPS Tracking	Yes	Yes	Yes	No	Yes
Interactive Map	Yes	Yes	Yes	No	Yes
Push Notifications	Yes	Yes	No	No	Yes
Route Optimization	No	Yes	No	No	Yes
Digital Display Integration	No	No	Yes	No	Yes
Data Analytics for Optimization	No	Yes	No	No	Yes
Integration with Existing Apps	No	Yes	No	No	Yes
Traffic Prediction	No	Yes	No	No	Yes

2.4 Features of the Proposed System

The UNIMAS Bus Tracking System builds upon the lessons learned from existing systems, integrating tailored features to enhance the campus transportation experience:

1. **Real-Time Tracking**:
   The system will employ **GPS-enabled tracking** to provide live updates on bus locations. This will allow students and staff to plan their journeys more efficiently, reducing wait times and improving satisfaction. For example, commuters traveling between Kolej Rafflesia and faculty buildings can track buses in real time, ensuring they arrive on time for classes or appointments.
2. **Interactive Map**:
   A user-friendly **interactive map** will display routes, stops, and estimated arrival times (ETAs). Users can access this feature via the UNIMASNow app, enabling them to visualize the bus network and plan multi-stop trips conveniently.
3. **Notifications**:
   Push notifications will alert users to delays, cancellations, or route changes. This

feature is especially beneficial during peak hours or unforeseen events, ensuring users remain informed and can adjust their plans accordingly.

4. **Data Analytics**:
   The system will include an analytics module to collect and analyze data on bus usage, peak hours, and route efficiency. This data will empower the university's transportation management to make **data-driven decisions**, such as optimizing schedules or reallocating buses during high-demand periods.
5. **Integration with UNIMASNow App**:
   Seamless integration with the existing **UNIMASNow app** ensures accessibility and familiarity for users. By embedding the tracking system within an already popular platform, adoption rates are expected to increase, reducing the need for separate app development and maintenance.
6. **Inclusion of All Routes**:
   Unlike systems with limited geographic coverage, the UNIMAS Bus Tracking System will cater to all campus routes, importantly those who stay at Kolej Kenanga, Kolej Rafflesia, Kolej Seroja, and Timur Gate. This ensures no commuter is excluded from the benefits of real-time tracking.
7. **Traffic Prediction Algorithms**:
   To address traffic-related inaccuracies, the system will utilize **predictive algorithms** that account for traffic patterns, road conditions, and historical data, ensuring more reliable ETAs and route suggestions.

2.5 Summary

This chapter have reviewed three transportation systems MyRapid PULSE, Grab, and Rapid Penang ITS. All of the information gathered are to extract valuable insights for the development of the UNIMAS Bus Tracking System. Each system highlighted critical features, such as GPS tracking, user-centered design, and real-time updates, while also revealing challenges related to scalability and traffic-related inaccuracies

Building upon these lessons, the proposed system aims to enhance campus transportation efficiency and user satisfaction by integrating:

- **Real-time GPS tracking** for accurate bus locations.
- **Interactive maps and notifications** for user convenience and up-to-date information.

- **Data analytics** to optimize schedules and routes.
- **Seamless integration with the UNIMASNow app**, leveraging an existing platform to ensure accessibility.

The UNIMAS Bus Tracking System will address current transportation challenges faced by the university, such as long wait times, scheduling inefficiencies, and lack of real-time updates. By prioritizing inclusivity, reliability, and ease of use, the system aims to align with UNIMAS's smart campus aspirations, creating a seamless and efficient transportation experience for all users.

CHAPTER 3 METHODOLOGY

3.1 Introduction

The next part would be Methodology. The methodology chapter is a crucial component of the project as it outlines the processes, techniques, and approaches need to be use to develop the proposed UNIMAS Bus Tracking System. This chapter provides a clear roadmap of the development process, system planning, design, development, and testing. By employing a structured methodology, the project will make sure that all requirements are adequately addressed and the potential system aligns with the objectives that have been set in the project proposal earlier.

The UNIMAS Bus Tracking System will be developed using an iterative software development methodology, which emphasizes continuous feedback, incremental development, and flexibility in accommodating changes. The primary goal of this system is to enhance the daily commuting experience of UNIMAS students and staff by providing real-time bus tracking, estimated time of arrivals (ETAs), and route notifications through the existing UNIMASNow app.

This chapter also elaborates on the system design elements requested by the project supervisor, including the context diagram, use case diagram, data flow diagrams (DFDs), and entity-relationship diagram (ERD). These design elements serve as the foundation for the system's architecture and database structure. The chosen methodology ensures that the project is completed efficiently, meeting both technical and user requirements while adhering to the project's timeline.

3.2 System Development Methodology

3.2.1 Methodology Overview

The system development methodology selected for the UNIMAS Bus Tracking System is the **Agile methodology**. Agile is a flexible and iterative approach that focuses on delivering small, incremental improvements through continuous collaboration with stakeholders. Given the evolving requirements of the project and the need to adapt to user feedback, Agile is a suitable choice to ensure that the system remains aligned with user expectations and technological advancements.

The Agile methodology is divided into multiple phases, each focusing on a specific aspect of the project lifecycle. The iterative nature of Agile allows for continuous testing and refinement of the system, reducing the risk of major issues during deployment. The core phases of the methodology include **requirement gathering**, **system design**, **development**, **testing**, and **deployment**.

3.2.2 Phases of the Agile Development Methodology

Phase 1: Requirement Gathering

In this project, the requirement gathering phase was conducted using secondary data. Instead of primary data collection through direct surveys or interviews, existing data is collected from a survey carried out by Majlis Perwakilan Pelajar (MPP) regarding "Hasil Tinjauan Pengangkutan Bas Shuttle UNIMAS" was analyzed. This approach provided valuable insights into the current challenges faced by UNIMAS bus users and their expectations for an improved system.

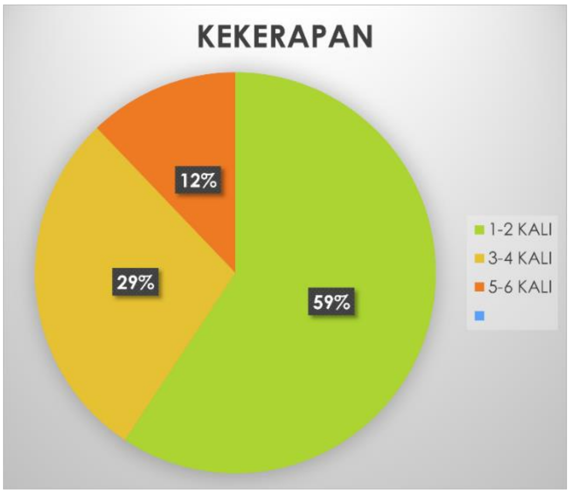

Figure 3.1

An analysis of the data revealed key patterns in bus usage. Out of 140 respondents, the majority of users (83) utilized the buses 1-2 times per week, while 40 users relied on them 3-4 times per week, and 17 users required the service 5-6 times weekly. Proposed schedule preferences highlighted peak weekday usage at 7:00 AM (18 users), with additional preferred times including 6:50 AM, 7:30 AM, 10:30 AM, 1:00 PM, 4:30 PM, 6:00 PM, and 8:00 PM. However, weekend services were noted as largely unavailable, pointing to a significant gap in service offerings.

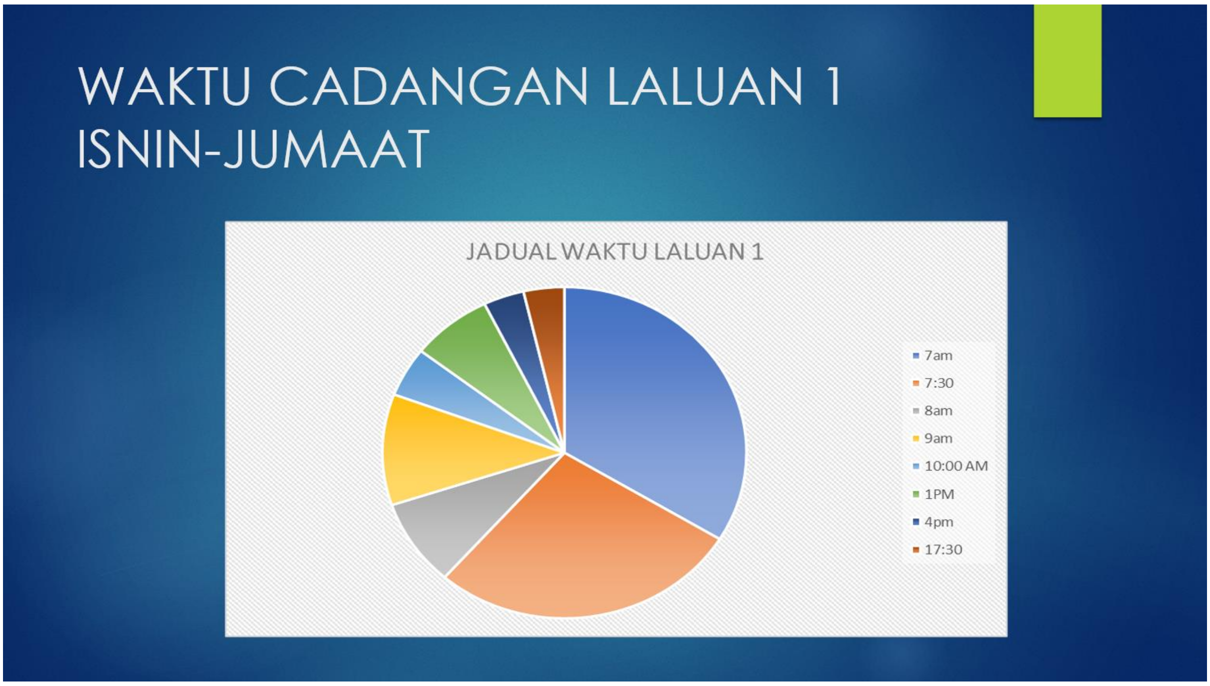

Figure 3.2

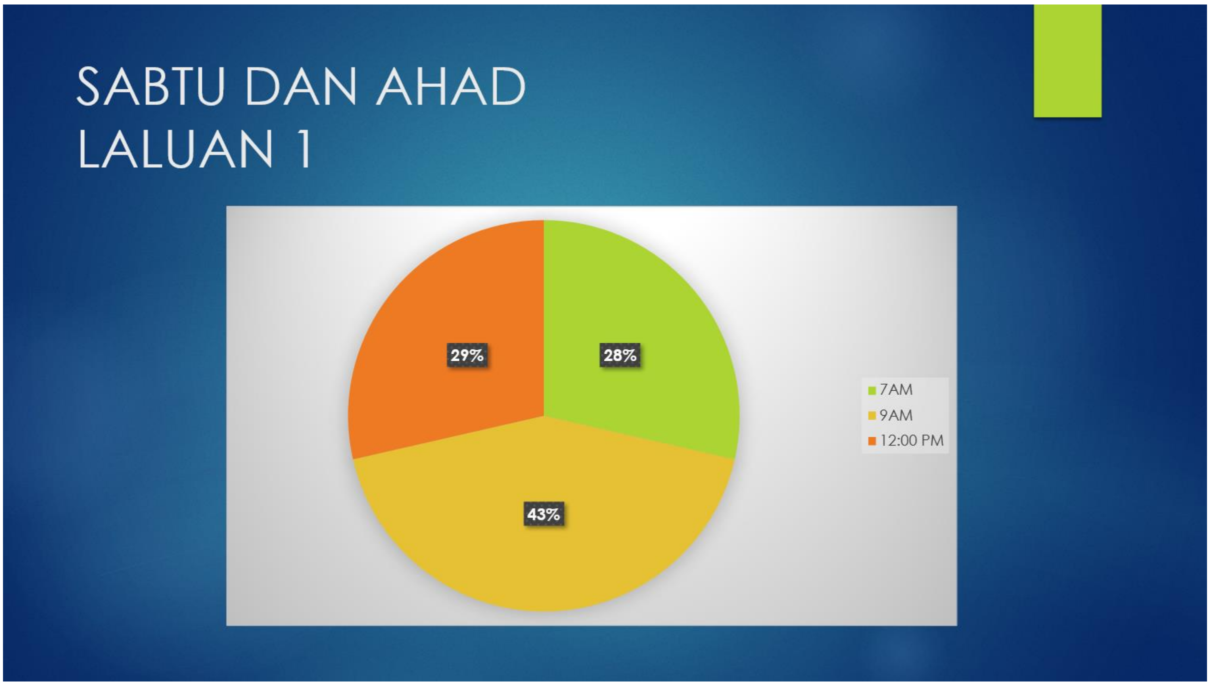

Figure 3.3

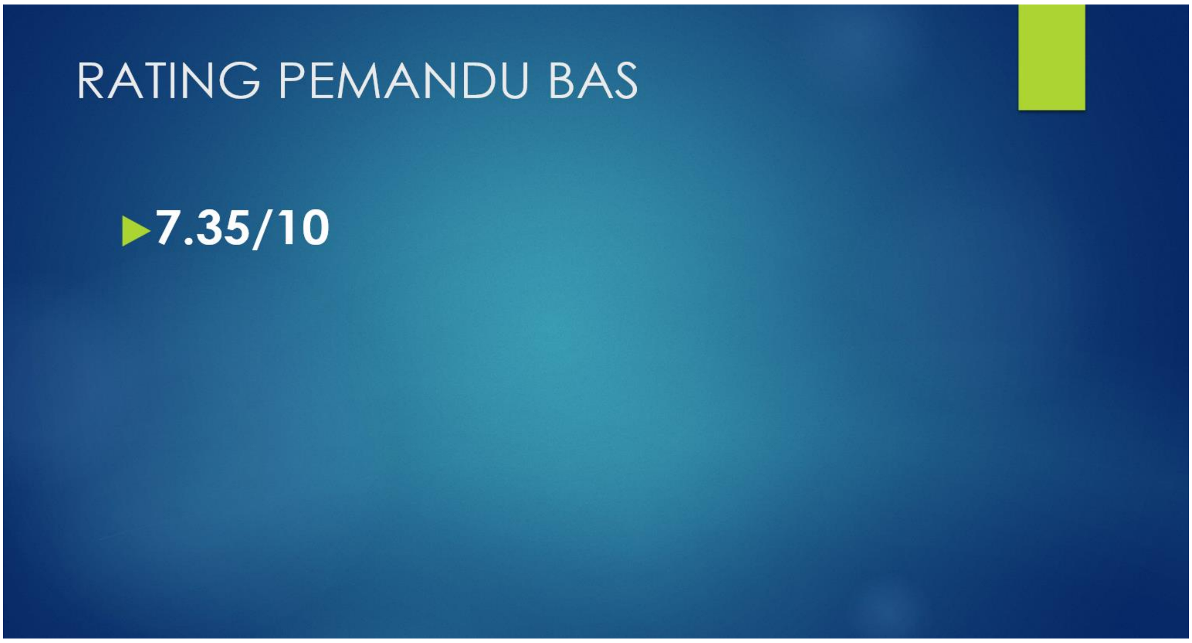

Figure 3.4

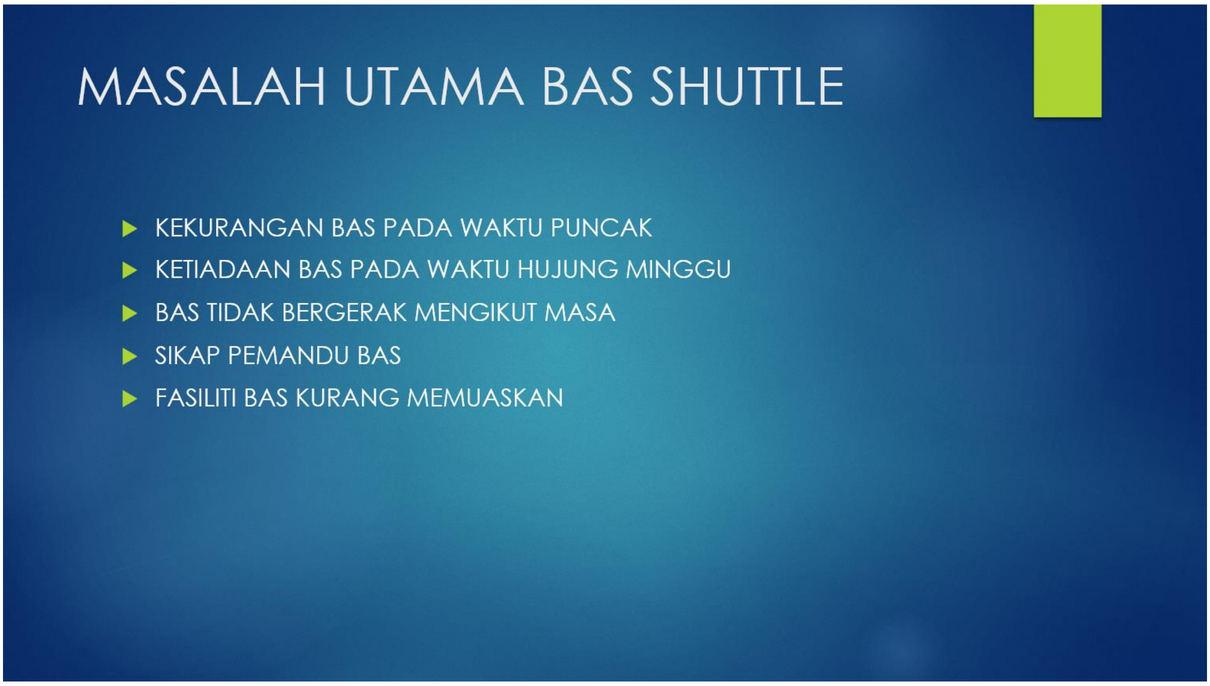

Figure 3.5

Key issues identified from the data included insufficient buses during peak hours, a complete lack of weekend services, frequent schedule violations, and substandard bus facilities alongside drivers behaviours concerns. Additionally, user ratings for bus drivers averaged 7.35 out of 10, reflecting room for improvement in service quality. Respondents suggested several enhancements to address these challenges: increasing bus frequency to 30-minute intervals, adding more buses during peak hours, establishing a direct communication channel with the bus department, and developing a mobile application akin to the University of Malaya's (UM) bus tracking app.

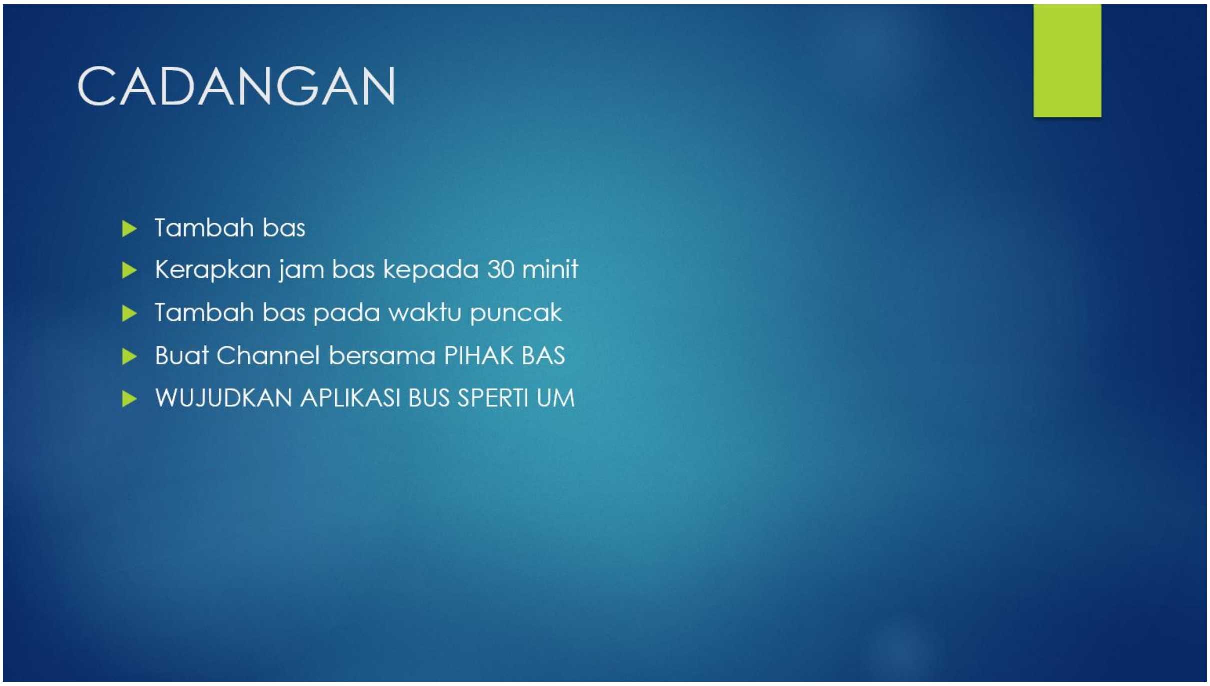

Figure 3.6

Based on these insights, critical user needs were defined to guide the project's development. These included the implementation of reliable real-time bus tracking to address schedule adherence or violations issues, push notifications for delays and schedule updates, improved service frequency during peak hours, and the introduction of better weekend bus services. Furthermore, users expressed a need for a communication platform to interact directly with the transportation department and emphasized the importance of enhanced bus facilities and better training for drivers.

To further refine the project's scope, a benchmarking exercise was conducted, comparing the current system and proposed improvements with successful models, such as UM's bus application. This comparison identified transferable features, including route maps, live updates, and user feedback tools, which can be integrated into the proposed system as well. Additionally, typical use case scenarios were synthesized to illustrate practical user interactions. For instance, a student checking the app at 6:45 AM would see a notification about the next bus arriving at 7:00 AM, along with an alert when the bus is five minutes away. Another scenario involved a staff member using the app to locate a newly introduced 10:00 AM weekend service. Lastly, the app's communication channel would enable users to report delays, with the transportation department providing real-time responses and estimated arrival times.

The requirement gathering phase resulted in the development of key deliverables, including a **User Requirements Document** that outlined the essential features: real-time tracking and notifications, improved service frequency, weekend availability, enhanced facilities, and a direct communication platform. These findings form the foundation for designing a user-centered and efficient transportation system for UNIMAS.

All of this data can be summarized as such :

Key findings include:

Usage Patterns:

- Most users utilize the bus 1-2 times per week (83 users), with smaller groups relying more frequently (3-6 times weekly).

Schedule Preferences:

- Peak usage occurs at 7:00 AM on weekdays, with additional demands at various times.
- Weekend services are nearly absent, representing a significant gap.

Identified Issues:

- Insufficient buses during peak hours.
- Lack of weekend service.
- Poor schedule adherence.
- Not optimized driver behavior.

User Ratings and Feedback:

- Drivers averaged a rating of **7.35/10**.
- Suggestions include increasing bus frequency, introducing weekend services, and creating a communication channel with the transport department.

Benchmarking:

- Successful features from other systems, such as real-time tracking, route maps, and user feedback tools, were identified for integration.

Phase 2: System Design

The design phase is critical for defining the structure and components of the UNIMAS Bus Tracking System. This phase establishes the architectural blueprint, which includes developing diagrams to visualize system components, their workflows, and interactions. These visual representations provide clarity for the development team and ensure that all aspects of the system are accounted for during implementation.

Deliverables:

- **Context Diagram:** Shows the high-level interaction between the system and external entities (e.g., users, buses, and the transportation department).
- **Use Case Diagram:** Illustrates the primary functionalities of the system and the roles of different users.
- **Data Flow Diagrams (DFD):**
    - **Level 0:** A high-level view of the system's data flow.
    - **Level 1:** A detailed breakdown of processes and their interactions.
- **Entity-Relationship Diagram (ERD):** Maps the database structure, detailing relationships between entities such as buses, users, and schedules.

Activities:

1. **Design the Overall System Architecture:**
    - Define key components, such as the GPS tracking module, notification system, and user interface.
    - Establish communication protocols between the backend, database, and frontend.
2. **Develop Visual Models to Represent System Workflows:**
    - Create diagrams to ensure stakeholders have a clear understanding of system functionalities and workflows.
    - These diagrams will serve as guides for development and testing.
3. **Identify Database Requirements and Relationships:**

- Define entities, attributes, and relationships for efficient data storage and retrieval.
- Ensure scalability to accommodate future enhancements.

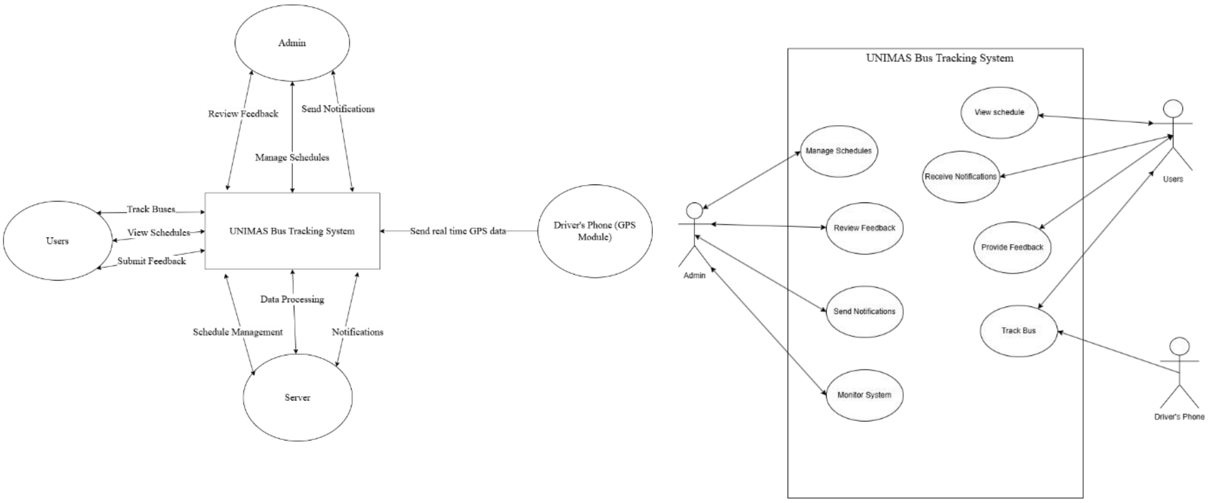

Figure 3.7

Phase 3: Development

The development phase involves building the core functionalities of the system based on the design specifications. The process follows an iterative approach, where specific modules are developed and tested incrementally to ensure a seamless integration of features.

Deliverables:

- A functional **prototype** of the UNIMAS Bus Tracking System.
- A **backend server** capable of processing real-time GPS data.
- A **frontend user interface** integrated into the existing UNIMASNow app, providing users with interactive maps, real-time updates, and notifications.

Activities:

1. **Develop Backend Services to Process GPS Data:**

    - Create APIs to handle GPS data from bus devices.
    - Implement algorithms for estimating arrival times based on location and traffic conditions.

2. **Integrate APIs for Real-Time Location Tracking:**

    - Use APIs (e.g., Google Maps API) to display bus locations and routes on an interactive map.
    - Ensure low latency and high reliability for real-time updates.

3. **Build the User Interface for the UNIMASNow App:**

    - Design an intuitive interface with features like bus tracking, schedule viewing, and push notifications.
    - Incorporate accessibility options to ensure ease of use for all users.

Phase 4: Testing

Testing is integral to ensuring the functionality, reliability, and user-friendliness of the system. This phase begins early in development and continues iteratively to identify and resolve issues promptly.

Deliverables:

- Comprehensive **test cases** and **test plans** to validate system performance.
- Detailed **bug reports** and **resolution logs** for tracking issues and fixes.

Activities:

1. **Conduct Functional, Usability, and Performance Tests:**

    - **Functional Testing:** Verify that features like GPS tracking, notifications, and schedule updates work as intended.
    - **Usability Testing:** Ensure the interface is user-friendly and meets accessibility standards.
    - **Performance Testing:** Test system behavior under various conditions, including high traffic and network latency.

2. **Gather Feedback from a Focus Group of Users:**

- Engage students and staff to test the system and provide insights into potential improvements.
- Use feedback to refine features and workflows.

3. **Resolve Identified Issues and Optimize System Performance:**

   - Address bugs and usability challenges to enhance system reliability and efficiency.

Phase 5: Deployment and Maintenance

The final phase involves launching the system and ensuring its continued operation through maintenance and updates. Feedback collected during this phase is essential for ongoing improvements and the addition of new features.

Deliverables:

- A fully **deployed UNIMAS Bus Tracking System** integrated into the UNIMASNow app.
- A **user manual and training materials** to guide users in utilizing the new system.
- A **maintenance plan** outlining periodic checks and updates.

Activities:

1. **Deploy the System to the UNIMASNow App:**

   - Ensure a smooth deployment with minimal downtime.
   - Conduct final testing in the production environment to confirm system stability.

2. **Provide User Training and Support:**

   - Offer workshops or tutorials to familiarize users with the app's features.
   - Establish a support channel for troubleshooting and feedback collection.

3. **Collect Feedback for Future Improvements:**

   - Monitor system performance and user satisfaction.
   - Use collected data to plan feature updates and address emerging needs.

3.2.3 Advantages of Agile Methodology for This Project

The Agile methodology offers several benefits that align perfectly with the goals and requirements of the UNIMAS Bus Tracking System. These advantages ensure that the project remains user-focused, adaptable, and efficient throughout its lifecycle:

Flexibility

Agile's adaptability is one of its greatest strengths. In this project, the ability to respond to changing requirements, stakeholder feedback, or unforeseen challenges is crucial. For instance, if new user preferences emerge during the testing phase, Agile allows the development team to incorporate these changes without disrupting the entire project. This ensures that the final system is well-suited to meet the evolving needs of UNIMAS students and staff, such as additional features like enhanced notifications or alternative schedule management tools.

Continuous Improvement

The iterative nature of Agile ensures that the system is continuously improved with each development cycle. Feedback from each iteration helps refine features, address bugs, and optimize performance. For example, an initial version of the system may focus on basic bus tracking, while subsequent iterations can add push notifications, route optimization, or user feedback integration. This approach minimizes the risk of significant issues at the deployment stage and ensures a polished final product.

User-Centered Approach

Agile prioritizes stakeholder collaboration, making it highly effective for addressing the real-world problems faced by the UNIMAS community. Through iterative feedback loops, the development team can involve students, staff, and the transportation department in shaping the system. This ensures that the system aligns with user expectations, such as providing real-time tracking, improving schedule adherence, and offering a user-friendly interface. Agile's emphasis on user involvement ensures that the system delivers tangible benefits to its intended audience.

3.2.4 Tools and Technologies Used

The development of the UNIMAS Bus Tracking System relies on a combination of robust tools and technologies to ensure scalability, performance, and ease of use. The following tools and frameworks were selected to support various aspects of system design, development, and management:

Backend Development

- **Python and Flask:** Python offers simplicity and extensive libraries, while Flask, a lightweight web framework, enables rapid development of RESTful APIs and backend services.
- **Node.js:** As an alternative, Node.js provides high-performance, asynchronous capabilities for handling real-time data processing and API requests.

Frontend Development

- **React Native:** This framework allows the development of cross-platform mobile applications with a single codebase, ensuring compatibility with both iOS and Android.
- **Flutter:** Flutter offers a highly customizable UI toolkit for creating visually appealing and responsive mobile applications, suitable for seamless integration into the UNIMASNow app.

Database

- **MySQL:** A reliable relational database management system (RDBMS) that handles structured data efficiently. Ideal for storing information such as bus schedules, user profiles, and tracking logs.
- **Firebase:** Provides real-time database capabilities, enabling live updates for features like bus location tracking and notifications.

APIs

- **Google Maps API:** Used for interactive maps and real-time location tracking. This API provides accurate geolocation and routing features, enhancing user experience by visualizing bus movements on the map.

Project Management

- **Jira:** A comprehensive project management tool for tracking progress, managing tasks, and facilitating team collaboration.
- **Trello:** A simpler alternative, Trello allows the team to organize tasks visually using boards, lists, and cards for efficient workflow management.

These tools and technologies collectively support the successful execution of the Agile methodology, ensuring that the system is scalable, user-friendly, and capable of meeting the specific needs of the UNIMAS community.

3.3 Requirement Analysis

3.3.1 Functional Requirements

Real-Time Bus Tracking

The UNIMAS Bus Tracking System is designed to provide a seamless and efficient transportation experience for users. A key feature of the system is **Real-Time Bus Tracking**, which allows users to view the current location of buses on an interactive map. The system updates the bus location dynamically using data from GPS modules installed on the buses, ensuring that users receive accurate real-time information. With this feature, users can visually track bus movements along predefined routes.

Estimated Time of Arrival (ETA)

Another essential feature is the **Estimated Time of Arrival (ETA)**, which provides users with accurate arrival times for each stop. These ETAs are calculated based on real-time traffic conditions, historical data, and the bus's current speed, allowing users to plan their journeys effectively. This feature will help users to plan their journeys well and accurately.

Notifications

In addition, the system incorporates **Push Notifications** to keep users informed about any delays, route changes, or cancellations. Users can also set personalized reminders for upcoming buses or schedule updates, which will be sent as notifications.

Schedule Viewing

The app also includes a **Schedule Viewing** feature, offering users a comprehensive view of bus schedules. The schedules display stops and timings, with filters available to narrow down results based on specific routes, days, or peak/off-peak hours.

User Feedback Module

To ensure continuous improvement in service quality, the system features a **User Feedback Module**, which allows users to rate bus drivers and provide feedback. This feedback is stored and analyzed to identify areas for improvement.

Route Optimization

To improve overall commuting efficiency, the system offers **Route Optimization**. This feature suggests alternate routes based on real-time traffic data and user preferences, helping to reduce waiting times and enhance the commuting experience.

3.3.2 Non-Functional Requirements

Besides the functional requirements, the system must meet several non-functional requirements that ensure its performance, security, and reliability.

Performance

For **Performance**, the system is expected to process and display location updates within 1-2 seconds by using GPS technology. It must also support up to 500 concurrent users without noticeable performance degradation.

Security

In terms of **Security**, the system must ensure that user credentials and personal data are encrypted during transmission and storage. Compliance with the Personal Data Protection Act (PDPA) is required to guarantee data confidentiality.

Scalability

The system must also be designed for **Scalability**, allowing for future expansions such as adding new routes or colleges or faculty. Load balancing mechanisms will be implemented to manage increased traffic and ensure smooth operation during high usage periods.

User Friendly

As for **User-Friendliness**, the interface should be intuitive, with clear navigation menus and interactive elements. Accessibility options like text resizing and color contrast adjustments should also be provided to cater to different user preferences. Additionally, the app must be compatible with both Android and iOS devices as well.

Reliability

Finally, **Reliability** is critical, with the system guaranteeing 99% uptime through redundant server setups and backup mechanisms. Failover protocols will also be in place to handle unexpected outages or failures.

3.3.3 Hardware and Software Requirements

To implement the UNIMAS Bus Tracking System, both hardware and software requirements are needed.

Hardware Requirements

Hardware Requirements include the use of driver’s smartphones as the primary hardware for tracking. The drivers' phones will utilize GPS functionality to collect and transmit location data at regular intervals, enabling real-time tracking. This approach is chosen to ensure faster and more efficient tracking without relying on additional GPS devices installed on buses. The system will also require a high-performance server to handle backend operations such as data processing, storage, and API responses. The server must be equipped with load balancing and redundant configurations to ensure reliability. Finally, users will access the system through smartphones with the UNIMASNow app installed, which should be compatible with Android version 8.0 (Oreo) or higher, and iOS version 12 or higher.

Software Requirements

For the Software Requirements, the backend development will utilize **Python with Flask** or **Node.js**, which are both lightweight frameworks suitable for building fast, RESTful APIs. For frontend development, **React Native** or **Flutter** will be used to ensure a consistent experience across both Android and iOS platforms. The system’s database will rely on **MySQL** for structured data such as bus schedules, routes, and user profiles, while **Firebase** will be used to provide real-time updates and synchronize location data. **Google Maps API** will be employed for accurate geolocation services and interactive mapping features, enabling users to view bus routes and track their real-time movements.**Project Tools** such as **Visual Studio Code (VS Code)** will be used for backend and frontend development, respectively. **Trello** or **Jira** will help manage tasks, track progress, and promote collaboration within the development team.

3.4 System Design

The system design phase involves creating a comprehensive blueprint for the UNIMAS Bus Tracking System. This includes visual models and diagrams that represent the system's architecture, workflows, and database structure. These design elements ensure that the development process is guided by a clear and structured plan.

3.4.1 Context Diagram

The Context Diagram provides a high-level overview of the system, illustrating how different entities interact with each other. It visualizes external actors and their relationships with the system. The primary components in the context diagram for the UNIMAS Bus Tracking System are:

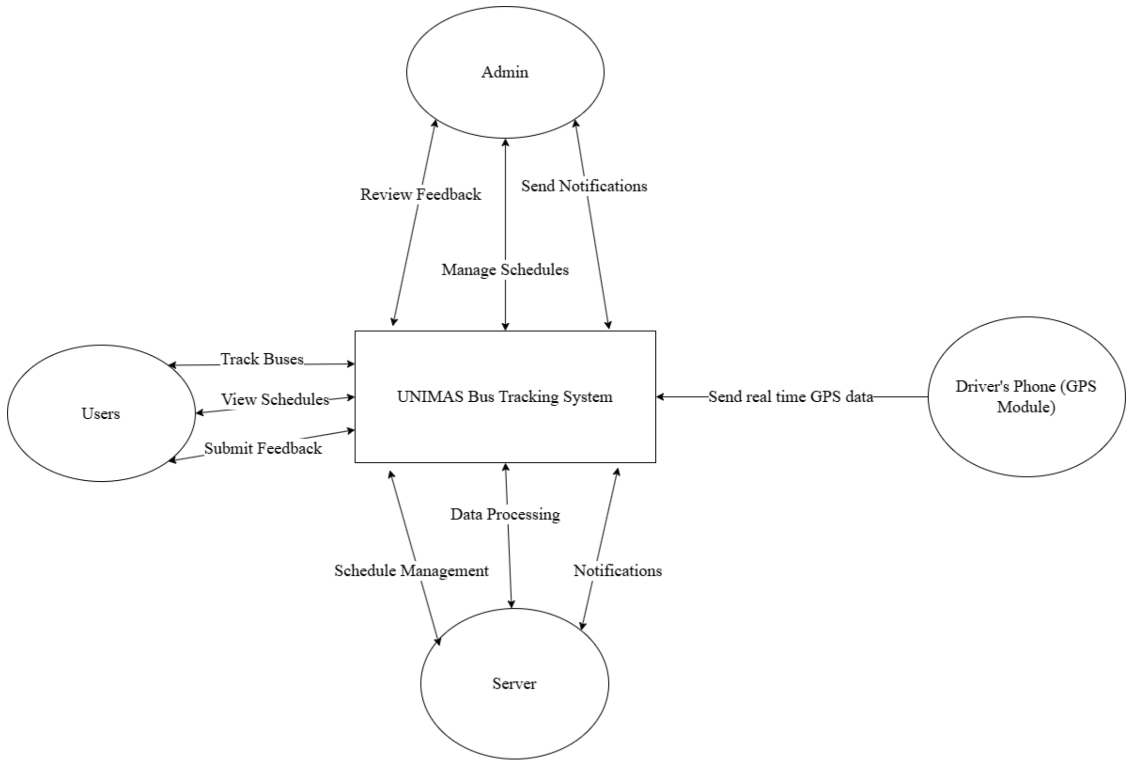

Figure 3.8

- **Users**: The primary users of the system access the UNIMASNow app to track buses, view schedules, and receive notifications. They interact with the system through mobile devices running the app.
- **Driver's Phone (GPS Module)**: The driver's smartphone acts as the GPS module, collecting real-time location data and transmitting it to the server. This eliminates the need for standalone GPS devices while ensuring fast and accurate tracking.
- **Admin**: The admin oversees system operations, managing schedules, routes, and notifications. The admin also reviews user feedback, monitors system performance, and ensures data accuracy.
- **Server**: The server handles backend operations, such as processing location data, storing user and schedule information, managing communications between components, and sending notifications to users.

Interaction Description

1. **Users**: Access the system through the UNIMASNow app to track buses and view schedules. The system sends real-time notifications or updates based on server processing.
2. **Driver's Phone**: Acts as a GPS device, sending live location data to the system for processing and updates.
3. **Admin**: Manages the overall operation by updating schedules, sending critical notifications, and analyzing user feedback to enhance service quality.
4. **Server**: Centralizes data processing, ensuring communication between the app, driver's GPS, and admin dashboard.

3.4.2 Use Case Diagram and Description

The Use Case Diagram identifies the main interactions between the system's actors and its functionalities. It visually represents how users, drivers, and admins interact with the system.

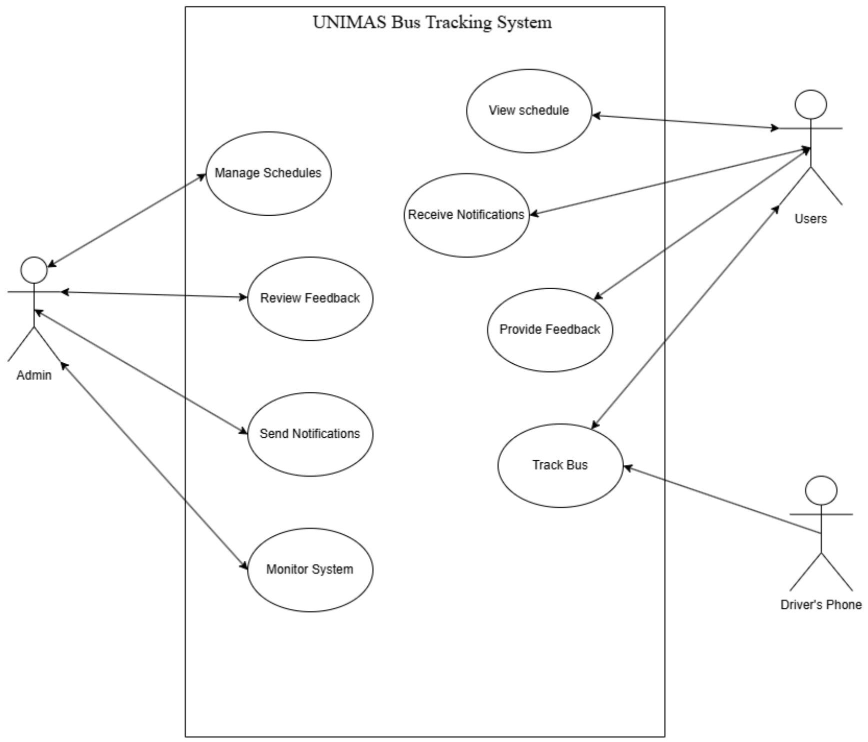

Figure 3.9

Primary Use Cases:

- **Track Bus**: Users view the real-time location of buses along predefined routes. The system fetches location data from the driver's phone and updates the app.
- **Receive Notifications**: Users receive alerts about delays, cancellations, or route changes. Notifications are triggered by the server or initiated by the admin.
- **View Schedule**: Users access comprehensive bus schedules, including stops, timings, and route information.
- **Provide Feedback**: Users submit feedback on bus services and drivers. This data is stored and analyzed by the admin to improve service quality.
- **Manage Schedules**: Admins create, update, and manage bus schedules and routes to ensure accurate system information.
- **Send Notifications**: Admins can manually send critical notifications to users, such as updates on service disruptions.
- **Review Feedback**: Admins analyze feedback provided by users and take actions to address service issues.
- **Monitor System**: Admins track system performance, review logs, and ensure data consistency.

Actors:

- **Users**: Engage with the app for tracking, schedules, notifications, and feedback.
- **Driver's Phone**: Sends GPS data to the server for real-time tracking.
- **Admin**: Manages system operations, including schedules, notifications, and feedback.
- **Server**: Processes data, manages interactions, and stores information.

Interaction Description

1. **Users**: Use the system through the app to:

    - **Track Bus**: View the real-time location of buses.
    - **View Schedule**: Access bus schedules, including routes and stops.
    - **Receive Notifications**: Get alerts about delays, cancellations, or updates.
    - **Provide Feedback**: Submit feedback regarding bus services or drivers.

2. **Driver's Phone**: Sends GPS data to the system, enabling real-time bus tracking.
3. **Admin**: Oversees the system's operations, including:

    - **Manage Schedules**: Update bus routes and schedules.
    - **Send Notifications**: Notify users about critical updates or disruptions.
    - **Review Feedback**: Analyze user feedback to improve services.
    - **Monitor System**: Ensure the system operates efficiently and resolve issues.

3.4.3 Data Flow Diagram (DFD)

The DFD illustrates how data flows through the system, showing data sources, processes, and storage locations.

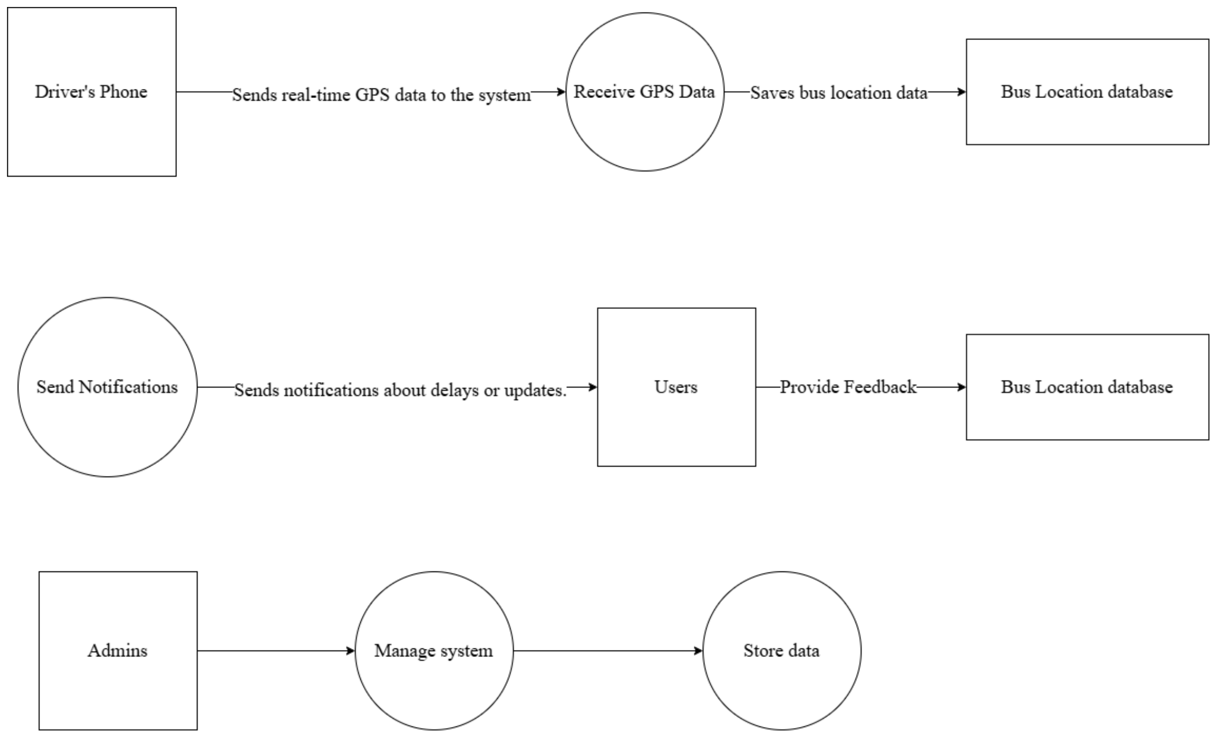

Figure 3.10

Level 0 DFD (High-level) :

- **Processes**:
  - Receive GPS Data: Driver's phone sends real-time location data to the server.
  - Send Notifications: The server generates notifications or relays admin-triggered alerts to users.
  - Store Data: The server stores user details, schedules, and feedback in the database.
  - Manage System: Admin manages schedules, routes, and notifications through the server.

Data Stores:

- **User Database**: Stores user information, preferences, and feedback.
- **Bus Location Database**: Stores real-time bus location and route data.
- **Schedule Database**: Stores detailed bus schedules, stops, and timings.

External Entities:

- **Users**: Interact with the system for tracking, notifications, schedules, and feedback.
- **Admins**: Update schedules, review feedback, and send manual notifications.
- **Driver's Phone**: Sends GPS data to the server.

Level 1 DFD (Detailed view):

- **Processes**:
    - Track Bus: Fetches and displays real-time bus location data to users.
    - Generate Notifications: Sends notifications to users based on admin inputs or bus statuses.
    - User Interactions: Users request schedules, track buses, or provide feedback.
    - Admin Actions: Admins manage schedules, view feedback, and send alerts.
    - Store Data: Saves schedule, user, and feedback data in relevant databases.

Data Flows:

- GPS Data: Flows from the driver's phone to the server.
- Notification Data: Sent from the server or admin to the users.
- Feedback Data: Flows from users to the server for admin review.

3.4.4 Entity-Relationship Diagram (ERD)

The ERD defines the database structure and relationships between entities in the system.

Entities:

- **User**: Stores user details such as name, contact info, and preferences.
- **Bus**: Stores bus details like bus number, driver name, and route ID.
- **Schedule**: Stores schedule data, including stops, timings, and routes.
- **Feedback**: Stores user feedback regarding bus services and drivers.
- **Admin**: Stores admin credentials and logs actions for system management.

Relationships:

- **User - Feedback**: A user can submit multiple feedback entries.
- **Bus - Schedule**: Each bus is associated with a specific schedule.

- **Admin - Schedule**: Admins can modify or manage multiple schedules.
- **Admin - Notifications**: Admins can send system-wide notifications to users.
- **User - Schedule**: Users can view schedules based on selected routes.

3.5 Tools and Technologies

The development of the UNIMAS Bus Tracking System requires the integration of various tools, technologies, and frameworks to achieve efficiency, scalability, and user-friendliness. This section describes the programming languages, development environments, APIs, and libraries selected to support the system's functionalities. These technologies were chosen based on their compatibility with Agile methodology and their ability to handle real-time operations, ensuring a robust and effective system.

Programming Languages and Frameworks

Figure 3.11

The backend development relies on **Python** and **Flask**, providing simplicity and extensive libraries for creating scalable RESTful APIs. Python's versatility makes it a suitable choice for handling real-time data processing and integrating with third-party APIs. Flask, being a lightweight framework, facilitates rapid development of server-side functionalities.

Figure 3.12

An alternative technology considered was **Node.js**, which offers asynchronous processing capabilities. Node.js is particularly effective for handling multiple API requests simultaneously, which is vital for live GPS tracking data.

Figure 3.13

For frontend development, the system uses **React Native**. This framework enables the creation of a cross-platform mobile application that is compatible with both iOS and Android devices. React Native's single codebase approach ensures consistency across platforms, reducing development time and resources. Additionally, **Flutter**, with its customizable UI components, is used to design an intuitive user interface, seamlessly integrating features like bus tracking and push notifications.

Database Management

Figure 3.14

The project employs **MySQL**, a relational database management system, for structured data storage. MySQL is used to store essential information such as bus schedules, user profiles, and system logs. To complement this, **Firebase Realtime Database** is utilized to handle live updates and notifications. Firebase ensures that users receive real-time data, enhancing the functionality of features such as bus location tracking and ETA notifications.

APIs and Libraries

Figure 3.15

The **Google Maps API** is a core component of the system, providing interactive map functionalities and real-time geolocation tracking. This API enhances user experience by allowing users to visualize bus movements and routes on an interactive map. Other geolocation libraries are integrated to ensure precise tracking and efficient routing.

Development and Project Management Tools

Figure 3.16

The development team utilizes **Visual Studio Code**, an integrated development environment (IDE), for coding and debugging.

Figure 3.17

For project management, **Jira** supports Agile workflows by enabling task tracking and collaboration. Alternatively, **Trello** is used for its visual task organization features, allowing the team to manage sprints efficiently.

3.6 Project Schedule and Milestones

The UNIMAS Bus Tracking System's development follows a structured timeline, divided into specific milestones to ensure steady progress. The Gantt chart below provides a visual representation of the project's phases and their corresponding timelines. The project's Agile approach allows for iterative development, accommodating flexibility and continuous improvement throughout the process.

Gantt Chart

Phase	Timeline	Description
Requirement Gathering	March 2025	Analysis of user needs and creation of a requirements document.
System Design	March - April 2025	Development of diagrams (context, use case, DFDs, ERD) and workflow designs.
Development	April - May 2025	Iterative development of backend APIs, frontend UI, and database integration.
Testing	May - June 2025	Conduct functional, usability, and performance testing.
Deployment and Maintenance	June - July 2025	System rollout, user training, and feedback collection.

Project Phase Descriptions

Requirement Gathering (March 2025): For this phase, the project team already use a secondary data to collects user requirements, So by analyzing survey data and benchmarking existing systems, such as the Rapid Pulse, Grab and Rapid Penang we already could gather enough information to start. This phase results in a detailed requirements document outlining essential features like real-time tracking and notifications.

System Design (March - April 2025): The design phase involves creating system architecture diagrams, including context diagrams, use case diagrams, data flow diagrams (DFD), and entity-relationship diagrams (ERD). These visualizations provide a blueprint for development, ensuring all workflows and components are clearly defined.

Development (April - May 2025): Development begins with the backend, where APIs are created to process GPS data and estimate bus arrival times. Frontend development integrates these features into the UNIMASNow app, using React Native for cross-platform compatibility and Flutter for an engaging user interface. The database is configured to manage schedules, user profiles, and real-time updates.

Testing (May - June 2025): Testing ensures the system's reliability, functionality, and usability. Functional tests verify core features like live tracking, while performance tests evaluate the system's behavior under different network conditions. Usability testing gathers feedback from students and staff to refine the user experience.

Deployment and Maintenance (June - July 2025): The final phase includes launching the system, integrating it into the UNIMASNow app, and conducting user training sessions. A feedback loop is established to monitor performance and plan future updates.

3.7 Summary

Chapter 3 outlines the methodologies, tools, and timelines that guide the development of the UNIMAS Bus Tracking System. By employing Agile methodology, the project ensures flexibility and user-centric development, allowing iterative improvements based on feedback. The integration of technologies like Python, Flask, React Native, and Firebase ensures scalability and real-time responsiveness, while tools like Jira and Trello facilitate efficient project management.

The structured project schedule, divided into requirement gathering, design, development, testing, and deployment, ensures that all objectives are met within the specified timeline. The Gantt chart highlights the sequential and iterative nature of the process, ensuring steady progress and adherence to deadlines.

This chapter establishes the foundation for creating a robust, user-friendly bus tracking system that addresses the transportation needs of the UNIMAS community effectively. The outlined tools, methodologies, and timelines provide a comprehensive roadmap for achieving the project's objectives while maintaining adaptability and efficiency.

CHAPTER 4: SYSTEM IMPLEMENTATION

4.1 Introduction

This chapter provides a comprehensive explanation of the implementation process for the UNIMAS Bus Tracking System. This project, developed as an independent mobile application, aims to deliver a real-time bus tracking experience to the campus community. The system uses

Firebase and Flutter technologies to enable drivers to broadcast their live GPS location and allows students to track buses via an intuitive interface. The following sections detail the system configuration, tools used, and the development stages leading to the realization of the system’s core functionalities. Initially, the project was proposed to be integrated into the existing UNIMASNow app to maximize accessibility. However, during the development phase, it became clear that direct integration was not feasible due to limited access to the UNIMASNow source code and administrative restrictions. As a result, the system was implemented as an independent mobile application that replicates the intended user experience while allowing full development control, rapid testing, and feature customization. The architecture remains compatible with future integration into UNIMASNow, should access be granted.

4.2 System Configuration

The system is developed using a combination of mobile and cloud technologies to ensure real-time location tracking and data accessibility. The choice of tools and platforms is guided by the project's requirement for rapid development, cross-platform compatibility, and real-time communication.

On the hardware side, the system is built to operate efficiently on modern smartphones. The driver's device, typically an Android phone equipped with GPS capabilities, is responsible for continuously sending location updates to the backend. The student’s device, on the other hand, fetches this data and displays it using Google Maps API integrated within the app. These devices are chosen due to their accessibility and widespread availability among users on campus.

The software stack is centered around Flutter, a UI toolkit by Google that allows for fast development of applications for both Android and iOS from a single codebase. Development is carried out in Visual Studio Code. Firebase serves as the backend solution, handling real-time database operations, cloud functions, and optional authentication services. Firebase Realtime Database or Firestore is used to store and update the driver’s GPS coordinates in real-time, ensuring that students receive the most up to date bus location.

4.3 Implementation of the Proposed System

The development of the UNIMAS Bus Tracking System follows a modular and iterative approach, aligning with Agile principles to allow continuous improvements and testing during the development cycle. This section explains in detail the implementation strategies used to build and deploy the independent mobile application using Flutter and Firebase as the core technologies.

The first step in the implementation process was setting up the backend services using Firebase. A Firebase project was initialized and configured to serve as the central hub for data management, particularly for storing and retrieving the real-time location of buses. Firebase Realtime Database was chosen over Firestore for its faster syncing capabilities and lower latency, which are crucial for live tracking functionalities. The structure of the database was designed to store GPS coordinates (latitude and longitude), timestamps, and identifiers for each bus instance. These values are continuously updated by the driver's device and read by the student's app in near real-time.

On the driver's side, a simplified mobile interface was created to allow the driver to start and stop location sharing with a single interaction. Upon activation, the driver's app requests permission to access the device's GPS module and begins transmitting the location at a regular interval typically every 5 to 10 seconds. This interval can be adjusted to balance accuracy and power consumption. The latitude and longitude values are extracted from the device's location service and sent to Firebase using HTTP requests or SDK-integrated methods provided by the Firebase Realtime Database package for Flutter.

On the student's side, the application is designed with usability and clarity in mind. The main interface includes a map powered by the Google Maps Flutter plugin. This map listens to changes in the database and dynamically updates the bus marker to reflect the current location of the tracked bus. To provide additional context to the user, the app also displays estimated time of arrival (ETA) based on the distance between the bus and predefined stop locations. The calculation is currently performed using a simplified formula based on static average speed but can later be improved by integrating Google's Directions API or real-time traffic analysis for higher accuracy.

In terms of architecture, the application adopts a clear separation between logic and UI by using a state management approach such as Provider. This ensures that UI components are reactive and update automatically when location data is modified. The backend logic is encapsulated in service classes, responsible for fetching, parsing, and pushing data between Firebase and the user interface.

To enhance system robustness, error handling mechanisms were implemented to manage potential failures such as lost GPS signals, denied permissions, or data transmission issues. The app is also designed to support offline scenarios to a certain degree, where the last known location remains visible until the connection is restored.

For development and testing, the system was deployed to emulators and physical Android devices. This allowed for real-time testing in campus-like environments to observe how location changes were reflected on the student app. Debugging tools and Firebase's live dashboard were instrumental in monitoring the data flow and ensuring consistent behavior across devices.

Although the system is currently deployed as a standalone application, its architecture is designed to support future integrations or extensions. For instance, a notification feature can be added using Firebase Cloud Messaging (FCM) to alert users when buses are approaching specific stops or experiencing delays. Similarly, user authentication can be enabled to differentiate between student, driver, and admin access roles.

In summary, the implementation of the UNIMAS Bus Tracking System demonstrates how modern mobile development frameworks and cloud services can be used to build a responsive, real-time, and user-friendly transportation tool. The current version successfully meets the primary objective of tracking buses on campus and can serve as a foundation for further enhancement and scaling.

4.4 Key Features Implemented

The UNIMAS Bus Tracking System includes several core features that contribute to a more efficient, transparent, and user-friendly transportation experience on campus. Each feature was carefully designed and implemented based on the identified problems and feedback collected during the requirement analysis phase. This section elaborates on the major functionalities that have been successfully developed in the current iteration of the application.

One of the primary features of the system is real-time bus tracking, which allows students to monitor the location of campus buses through a live map interface. This feature is made possible through continuous data streaming from the driver's mobile device to Firebase, where GPS coordinates are updated every few seconds. The student's application listens to these updates and reflects them immediately on the map interface, showing a moving bus marker. This eliminates uncertainty and guessing around bus arrival times, providing a much-needed improvement over the traditional paper-based schedule or waiting in blind anticipation.

Another important component is the Estimated Time of Arrival (ETA) feature. Although currently implemented in a basic form, this functionality estimates the bus arrival time based on the distance between the bus's current location and the student's selected bus stop. The calculation uses a fixed average speed value to estimate travel time. While the current version does not yet incorporate real-time traffic data, this foundation enables students to get a general sense of when their bus will arrive, improving planning and reducing stress associated with long waiting times.

The driver interface is intentionally kept simple to avoid distraction and ensure safety. Upon logging in or opening the app, the driver is presented with a single button to start location sharing. Once activated, the app continues to send location data to Firebase in the background. This ensures that the driver can focus solely on driving while the system handles the data transmission automatically. The simplicity of the driver-side interface was also designed to accommodate varying levels of tech literacy among drivers.

On the student side, the application offers a clean and intuitive user interface. The home screen presents an interactive map with the current bus location, along with options to select a bus route or refresh the map if needed. The UI is optimized for responsiveness, ensuring smooth user experience even on devices with limited processing power. The map interaction and design elements are created using Flutter's widget system, ensuring cross-platform compatibility and a consistent look and feel on both Android and iOS devices.

A notification system has also been planned, with initial groundwork laid for implementation using Firebase Cloud Messaging. This system is intended to provide proactive alerts to students regarding bus delays, route changes, or critical system announcements. Although this feature is not yet fully active, its infrastructure is in place and will be enhanced in future iterations to make the system even more responsive to real-world transportation challenges.

Finally, the system's backend logic and scalability play a crucial role in its long-term effectiveness. The database structure supports the addition of multiple buses and different routes without requiring major code changes. This design decision ensures that the system can grow alongside the university's transportation needs, accommodating more bus lines or user roles in the future.

Overall, these features collectively address the core problems identified in the early stages of the project, particularly in terms of real-time visibility, better time management, and a more informed commuting experience for students and staff alike. As the system continues to develop, more features will be added to further improve its usefulness, reliability, and alignment with UNIMAS's vision of a smart and connected campus.

4.5 Screenshots of the system

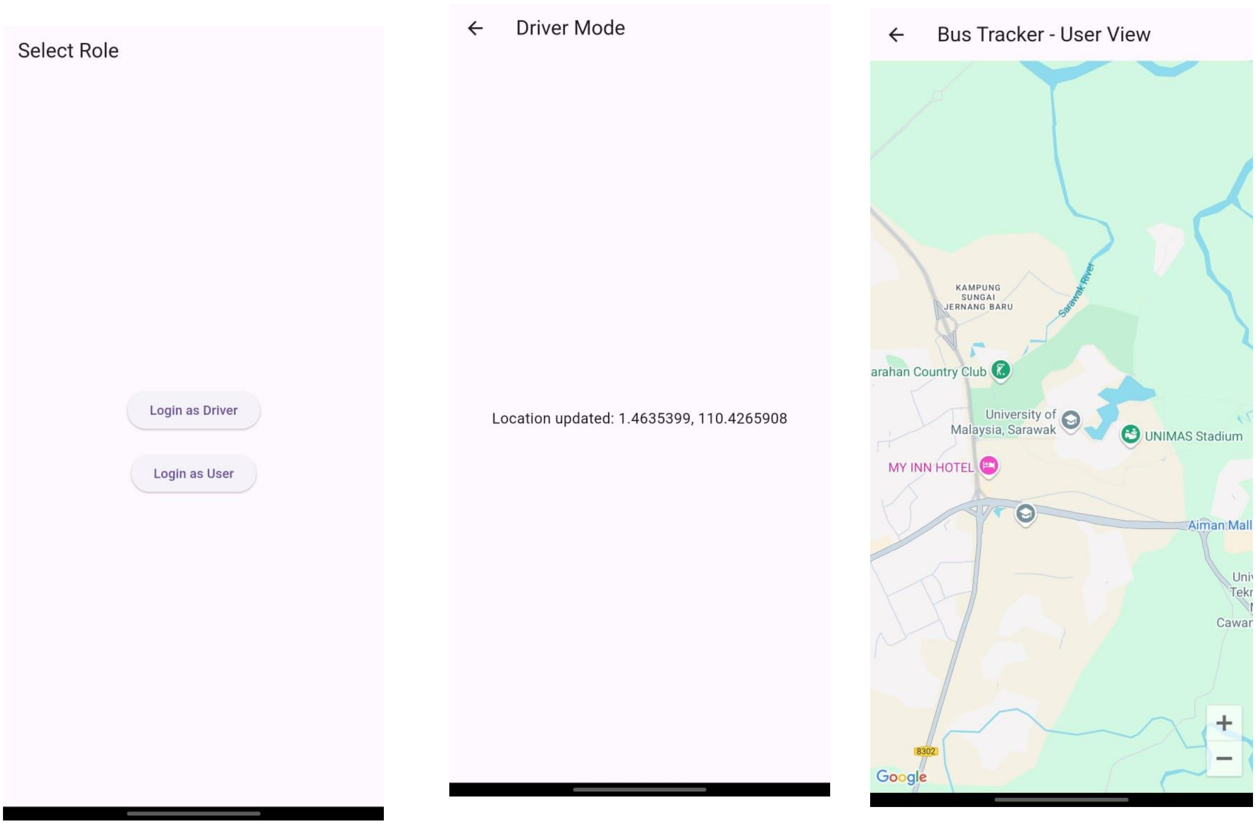

Figure 4.1

This is the first design of the application UI, the design aims for a useful and understandable UX instead of complicated design. Although that have been put in mind, there is an update of the UI where it meets the user requirement and more user related so that it appear reliable to the user. The following update is as follows:

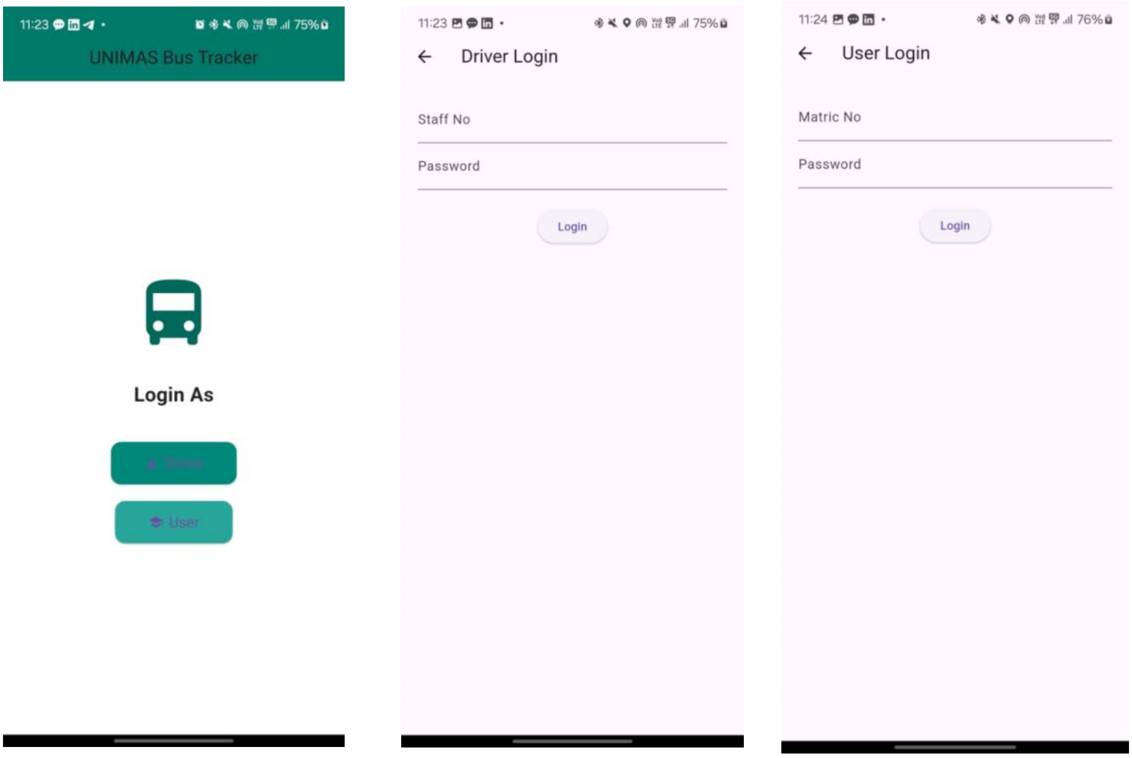

Figure 4.2

For the second time of design it have been confirm to add a login page so that it easier to track and limit only for student and staff as that is the group of people who actually use bus as a transportation in UNIMAS.

4.6 Summary

This chapter has detailed the comprehensive process involved in implementing the UNIMAS Bus Tracking System as an independent mobile application. The implementation journey began with the configuration of system requirements, both hardware and software, leveraging tools such as Flutter for the frontend and Firebase for the backend. These choices were driven by the need for cross-platform support, real-time capabilities, and rapid development—qualities well aligned with the project's objectives.

The development was focused on delivering a practical solution to long-standing problems faced by UNIMAS students and staff, such as the lack of real-time bus tracking and unpredictable schedules. The integration of Firebase Realtime Database enabled seamless two-way communication between drivers and students, allowing the system to relay live location data with minimal latency. Through this setup, students can visualize the movement of campus buses on a dynamic map interface, significantly enhancing their commuting experience.

Core features such as real-time location sharing, estimated time of arrival (ETA), and simple driver-side location broadcasting have been successfully implemented and tested on physical devices. Although some advanced functionalities like push notifications and traffic-based ETA refinement are reserved for future updates, the current system already demonstrates a functional and valuable solution. The database structure has also been designed for scalability, ensuring that additional buses and routes can be accommodated with ease.

Overall, the implementation phase has achieved its primary goal: to build a working prototype that addresses the transportation inefficiencies on campus by utilizing modern mobile and cloud technologies. This chapter concludes the development aspect of the project and sets the foundation for the subsequent evaluation and testing phases, which will be discussed in the next chapter.

CHAPTER 5: SYSTEM TESTING

5.1 Introduction

System testing is a crucial phase in software development that ensures the application meets its intended functionalities and performance expectations. For the UNIMAS Bus Tracking System, the primary goal of testing was to validate that all modules and features operated correctly under realistic usage scenarios. As the system was developed to solve real-world issues involving campus transportation inefficiencies, the quality and reliability of its performance are vital. This chapter outlines the testing strategies, methodologies, and results obtained during the verification and validation of the system, focusing on functional, non-functional, and user experience testing.

5.2 Testing Strategy and Environment

To ensure thorough evaluation, the system was subjected to multiple layers of testing. The testing strategy involved a combination of black-box testing for functional validation, exploratory testing for usability assessment, and compatibility testing across devices. The testing environment consisted of both physical smartphones and emulators via Android Studio. Firebase’s Realtime Database dashboard was used for backend monitoring, while real-time interactions between driver and student modules were simulated in real-world campus-like conditions.

The black-box testing approach focused on assessing how the system responds to input and delivers output without delving into the internal logic. This approach was suitable for verifying whether the user interfaces, GPS location flow, and data synchronization functioned correctly from the end-user's perspective. By treating the application as a “black box,” the goal was to ensure that the system met user expectations and functional requirements.

5.3 Functional Testing

The functional testing phase aimed to verify that the main features of the system performed as intended. The driver-side app was tested for its ability to capture and push location data to Firebase at regular intervals. The student-side app was then evaluated to ensure it could successfully retrieve this data and display the location in real-time on a map interface. The location sharing was tested by moving with the device to different spots, verifying whether location changes reflected correctly on the student’s screen.

In particular, the location updates were tested in scenarios such as rapid movement, switching between mobile data and Wi-Fi, and changes in GPS signal strength. It was observed that Firebase successfully handled these updates with minimal latency. In most test cases, the student interface showed updated bus locations within two seconds of the change occurring. ETA calculations, while based on a simplified speed-distance logic, produced consistent and realistic estimations.

Login and authentication mechanisms (if enabled), app permissions, and refresh functionalities were also tested. Edge cases were evaluated such as disabling location permission, background minimization of the app, and attempting to start location sharing without GPS enabled. The application handled these cases by providing appropriate alerts and fallback behavior, ensuring robustness.

5.4 Non-Functional Testing

Non-functional testing was conducted to assess aspects such as performance, usability, reliability, and device compatibility. Performance testing showed that the application loaded within three seconds on most devices, with map rendering taking no more than one to two additional seconds after initialization. The Firebase backend was able to sustain frequent data writes without delays, ensuring smooth real-time behavior.

Usability testing involved informal sessions with several student testers who provided feedback on the interface design, readability, icon clarity, and overall experience. The majority reported that the app was intuitive and easy to navigate, especially with the clean design and visible map markers. Some feedback suggested potential improvements in providing clearer ETA indicators and distinguishing buses using route-specific labels or colors.

Reliability was evaluated through long-duration tracking tests where the driver's app ran continuously for up to 30 minutes, with the student's interface actively reading data throughout. The application maintained a consistent stream without crash or memory leaks, confirming its stability.

Compatibility testing confirmed that the application worked across a range of Android devices with different screen sizes and OS versions. While testing for iOS was limited, the Flutter framework ensures that similar behavior can be expected upon iOS deployment with proper adjustments.

5.5 Summary of Test Results

In summary, the system successfully passed all critical functional and non-functional tests, meeting its core objectives. Real-time tracking, location updates, and basic ETA display worked as expected across all test cases. Minor usability feedback was received for future enhancement. The testing phase confirmed that the app is ready for real-world use among students and drivers in the UNIMAS campus environment, though further refinements and additional modules can be incorporated post-deployment.

This successful testing phase reflects the importance of adopting reliable development tools, simplified yet efficient backend architecture (Firebase), and a design philosophy centered around user experience. The test results provide strong evidence that the system is capable of fulfilling its intended role in enhancing the campus transportation system.

CHAPTER 6: CONCLUSION

6.1 Introduction

This chapter provides a comprehensive conclusion to the development and implementation of the UNIMAS Bus Tracking System. It revisits the project's objectives, highlights the major achievements, and reflects on the journey from system planning to testing. In addition, this chapter outlines the challenges encountered during the project, the solutions applied, and the scope for future improvements. The development of this project has been an insightful process that has blended theoretical knowledge, technical skills, and practical problem-solving to address real needs within the UNIMAS community.

6.2 Project Overview

The UNIMAS Bus Tracking System was conceptualized to solve a very real issue faced by students and staff: the lack of real-time information on campus bus movement. The traditional reliance on static schedules and uncertain waiting times has long been a source of frustration. This project aimed to create a more efficient, technology-driven solution that could improve the transportation experience by enabling users to track buses in real time.

At its core, the system enables drivers to transmit their GPS location data using a smartphone, which is then sent to Firebase and retrieved by student devices. Through the student-facing interface, users can view live bus locations on a map and estimate arrival times. The solution eliminates guesswork and reduces the time wasted waiting for buses, especially during peak hours or in uncertain weather conditions.

6.3 Objectives and Methodology Revisited

Three primary objectives were established during the early stages of the project: to implement real-time bus tracking using GPS, to display bus location and ETA in a mobile app, and to build a user-friendly application interface. All three objectives were successfully achieved. The real-time aspect was accomplished using Firebase Realtime Database, GPS integration, and Flutter's reactive UI features.

The methodology chosen for the project was based on Agile and iterative development practices. This approach allowed flexibility in development, constant testing, and rapid response to design or logic flaws. The iterative process ensured that core components were implemented first, tested independently, and refined before integrating additional features.

The project began with planning and system requirement gathering, followed by the design of diagrams including use cases, ERD, and DFDs. The implementation phase then proceeded with the app scaffold, Firebase integration, and UI/UX layout design. Continuous testing throughout development ensured each module was stable before proceeding to the next.

6.4 Major Achievements

One of the most notable achievements of this project is the successful creation of an independent mobile application, capable of operating outside of any third-party system such as UNIMASNow. This independence provides flexibility and control over future updates and integrations.

The use of open-source tools like Flutter and Firebase not only made the development cost-effective but also contributed to quicker deployment. Furthermore, the app architecture is scalable, allowing multiple buses or routes to be supported with minimal modification. Basic ETA prediction and live map updates have laid a strong foundation for more advanced features to be added later.

Another significant success was the testing phase, where the system demonstrated high reliability, responsiveness, and ease of use. Positive feedback from early testers validated the design choices and confirmed that the application meets its intended purpose effectively.

6.5 Challenges and Limitations

Several challenges were encountered throughout the project. One of the initial difficulties was determining how to integrate live GPS tracking without using expensive external GPS devices. This was resolved by utilizing the driver's phone as the data source, which simplified the hardware requirements.

Another major constraints encountered was the inability to access the UNIMASNow app's internal infrastructure. Despite early plans to embed the bus tracking system within the existing platform, institutional policies and restricted permissions led to a strategic shift. To maintain development momentum and ensure timely delivery, the system was instead developed as a standalone or independent mobile app. This approach allowed full control over design, features, and testing, although it temporarily sacrifices the seamless integration that was originally envisioned. Future collaboration with Tun Abang Haji Openg Digital Centre (TAHODC) may enable proper merging of the two systems.

Another challenge involved managing state and real-time updates in Flutter. Ensuring that map markers updated in real-time without performance degradation required careful state management, especially on lower-end devices.

Limitations include the lack of integration with the university's main app, the absence of admin dashboards, and simplified ETA calculation without traffic-aware routing. These features are marked as future enhancements.

6.6 Future Recommendations

To further enhance the system, several future improvements are proposed:

- Integration with Firebase Cloud Messaging for push notifications.
- Expansion to support multiple routes and buses, with route filtering.
- Improved ETA calculation using the Google Maps Directions API and live traffic data.
- Development of an admin panel for managing routes, drivers, and system status.
- Deployment to Google Play Store and continuous updates based on user feedback.

6.7 Final Conclusion

The UNIMAS Bus Tracking System has succeeded in delivering a modern, mobile-based solution for campus bus tracking, developed entirely as a standalone application. The project has demonstrated how real-time data and cloud computing can be harnessed to improve daily life on campus. It reflects the growing need for digital transformation in public service delivery and exemplifies how students can contribute meaningfully to solving community problems. This standalone project, while differing from the initial plan of UNIMASNow integration, ultimately enabled a more agile and testable system. It remains compatible with future merger into UNIMASNow, ensuring the original objective can still be fulfilled with institutional support.

Through thoughtful planning, consistent development, and comprehensive testing, this project has laid the groundwork for a fully functional and scalable system that aligns with the smart campus vision of UNIMAS. With future enhancements, this system can serve as a long-term solution for campus transportation logistics.

References

1. Prasarana Malaysia Berhad. (n.d.). *MyRapid PULSE app*. MyRapid. https://myrapid.com.my/myrapid-pulse/
2. Grab Holdings. (n.d.). *Grab SuperApp Malaysia*. https://www.grab.com/my/
3. Firebase. (n.d.). *Realtime Database*. Firebase. https://firebase.google.com/docs/database
4. Flutter Dev. (n.d.). *Google Maps for Flutter*. Flutter Official Docs. https://docs.flutter.dev
5. Flutter. (n.d.). *Build apps for any screen*. Flutter Official Website. https://flutter.dev/
6. Google Developers. (n.d.). *Directions API overview*. Google Maps Platform. https://developers.google.com/maps/
7. Universiti Malaysia Sarawak. (n.d.). *UNIMASNow*. Google Play Store.
8. Firebase. (n.d.). *Firebase Cloud Messaging (FCM)*. https://firebase.google.com/docs/cloud-messaging