Revolutionizing UTAR Bus Transit System: Development of Mobile Application with NFC payment and Real-time Tracking

By

Chong Xin Yuet

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONOURS) INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

FEBRUARY 2026

COPYRIGHT STATEMENT

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Systems (Honours) Information Systems Engineering at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

Firstly, I would like to thank sincerely to my project supervisor, Ts Dr Phan Koo Yuen, whose guidance, support, and professionalism have been crucial to the progress of this work. His knowledge and assistance inspired my project's creative and technical viewpoints.

Besides, I would like to thank my dearest family and friends for always staying by my side. Your belief and love have made this achievement possible.

ABSTRACT

The paper aims to modernize the existing campus transportation infrastructure by integrating advanced technologies into a cohesive, user-centric solution. This project focuses on the development of a fully functional mobile application that utilizes efficient payment methods, real-time tracking, and personalized transit scheduling to enhance the user experience for students, administrators, and bus drivers. The core feature of the system is NFC-based payment integration for quick and secure transactions. Also, real-time GPS tracking for accurate bus location, and dynamic seat availability updates are included. OCR technology is employed to automate bus scheduling based on students' class timetables, ensuring timely arrivals and reducing the manual planning burden on users.

By adopting these technologies, the university is able to provide a modern, efficient, and user-friendly solution, significantly enhancing the campus shuttle experience. This project not only addresses the current challenges of the bus transit system but also positions UTAR at the forefront of campus transportation innovation, setting a new standard for efficiency, reliability, and user satisfaction.

Area of Study: Mobile application development
Keywords: NFC, GPS, OCR, Smart Transportation System, Real-time Tracking

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

UTAR	Universiti Tunku Abdul Rahman
GPS	Global Positioning System
QR	Quick Response
NFC	Near Field Communication

Chapter 1

Introduction

In university settings, where students depend on scheduled transit services for their daily commutes, effective and reliable transportation systems are essential to guaranteeing people's seamless mobility. By offering a sustainable and efficient transportation system, it reduces the concerns related to commuting to school, including parking and traffic problems.

With the rising demand for accessibility and efficiency, campus transit systems have changed dramatically in these years. To enhance the user experience, today's transit systems have adopted technologies like GPS tracking, real-time scheduling, and digital payment methods. Despite being one of the best universities in Malaysia, Universiti Tunku Abdul Rahman (UTAR) still needs students to purchase paper tickets or monthly paper pass. Even though Touch 'n Go (TNG) QR code scanning offers a digital payment option, this approach is still inefficient and falls short of the standards of existing systems.

Therefore, the goal of this project is to build the exclusive application that is specialized for UTAR community in order to provide a smooth and practical campus transportation experience. Their primary objective is to minimize the gap between the needs of a tech-savvy generation and conventional transit systems. The application incorporates innovative technologies such as live monitoring, Near Field Communication (NFC)-based payment and other advanced automation capabilities. In addition to enhancing the everyday commute for the university community, this application has the potential to establish a new benchmark for campus transportation systems and integrate it with the digital transformation objectives of smart campuses.

1.1 Problem Statement and Motivation

Efficient and dependable transportation are critical elements in a university context, especially for those students who rely on campus bus services for everyday mobility. However, the present UTAR bus fare collection method is still relying on paper tickets or TNG QR code scanning. Although UTAR provides an alternate digital payment mechanism, users may find it difficult to follow their riding history because TNG is a third-party software. Besides, even though UTAR recently collaborated with a third-party application to oversee its bus service, depending on outside platforms for long-term campus accessibility might not be the most practical or sustainable course of action as well.

Additionally, statistical studies have demonstrated a strong positive correlation between timeliness and user satisfaction, suggesting that punctuality plays a crucial role in deciding how satisfied passengers are with public transit [1]. Yet, one of the main issues with the unpredictability of bus services has been punctuality. Many students are left without seats or sometimes unable to board buses during peak hours since demand frequently exceeds capacity. Students have also reported waiting for extended periods of time without obtaining correct information about bus availability, and many have experienced frequent delays or cancellations. Despite the fact that UTAR keeps a webpage with schedule updates, students seldom visit it because it is not immediately applicable to their daily plans. Students cannot make informed commuting decisions without access to real-time updates, resulting in lost time and reduced faith in the bus service.

The inefficiency of existing system is further highlighted by the lack of automatic scheduling tools. Students currently have to arrange manually their class timetables with the bus schedules, which is time-consuming and prone to mistakes. When busses fail to match up with class scheduling, students risk missing classes or arriving late. A crucial gap is shown by this lack of integration between academic and transportation schedules, in which the campus transit system cannot effectively serve students' time management needs in the absence of intelligent planning and automation.

The motivation for this project was to use innovation to close these gaps. In order to improve efficiency and user experience, modern public transportation systems are progressively implementing cutting-edge technologies like contactless payments, monitoring with GPS, and predictive scheduling. As one of Malaysia's top universities, UTAR should not leave behind in offering its students an innovative and technologically advanced method of transportation. To revolutionize UTAR Bus Transit Application, this project incorporated NFC-based payments to reduce ticketing and QR scanning

delays, and real-time bus tracking to keep students up to date on arrivals, capacity, and possible delays. Also, students will be able to match bus schedules with their classes through this automated scheduling tools, contributing to reduced wasted time and increased punctuality.

Aside from convenience, this project was motivated by the consistent with wider environmental concerns and smart campus goals. By encouraging students to take public transportation rather to drive their own cars or bikes, a more effective bus system helps reduce traffic on campus and carbon emissions. This sets a good example for other institutions and directly promotes UTAR's sustainability objectives. In the end, the UTAR Bus Transit Application redefines the campus transportation experience by blending technology, sustainability, and user-focused design in addition to addressing the crucial issues of inefficiency and inconvenience.

1.2 Project Objectives

1.2.1 To provide an effective fare payment system that enables students to pay bus fares with ease, decreasing down on paper ticket and QR payment boarding delays.

1. Employ NFC technology as the main payment option for rapid, contactless transactions.
2. Integrate a balance transfer feature to provide flexibility and convenience.
3. Develop a report card lost feature to prevent unauthorized use.

1.2.2 To enhance UTAR's bus transportation system's dependability and transparency by implementing real-time tracking and instant resource upgrade services.

1. Build an automated notification mechanism to inform students of bus service changes.
2. Employ GPS technology to provide real-time bus locations tracking and seat availability updates.
3. Develop the system as role-based access to ensure that each stakeholder benefits from system reliability, reducing information gaps and improving responsibility.

1.2.3 To improve bus scheduling efficiency through automated alignment with students' timetables and dynamic resource management.

1. Adopt OCR technology to retrieve timetable information.
2. Perform matching between timetable and bus schedules to recommend suitable bus trips for students.
3. Trigger notifications to students 15 minutes earlier when each trip starts to ensure timeliness.

1.3 Project Scope and Direction

By the completion of this project, a fully functional mobile application will be developed to renovate the UTAR campus bus transit system. This project aims to provide a smooth transit experience for its primary stakeholders, who include administrators, bus drivers, and students. Advances in technology provide students and bus drivers with a dependable and easily available means of transportation, while administrators may manage bus service operation more effectively. To improve the dependability and simplicity of campus transportation, the application will incorporate a range of key functions, such as NFC-based fare payment, real-time bus tracking, and customized transit scheduling. By providing a consistent, user-friendly interface, the project seeks to reduce waiting times, payment delays, and improve student punctuality.

1.4 Contributions

The key enhancement in this project is the substitution of the current payment method of UTAR with the NFC technology. Students can use this function to pay bus fare efficiently, avoiding wait in long lines and to pay in QR code scanning or paper tickets. In the meanwhile, picking up individual paper tickets is no longer necessary for bus drivers. The use of GPS technology facilitates real-time updates for all parties involved. The use of OCR technology to extract information from photos, which has never been included into a campus bus transit system, is another crucial addition of this project. Students can prevent missing buses by uploading their timetables to the system, which sets their bus schedules and sends upcoming bus reminders.

In addition to modernizing the existing campus transportation infrastructure, this application sets a new benchmark for campus transit options. It solves the issues of inefficiency, crowding, and scheduling conflicts by utilizing state-of-the-art technologies and a user-centered design, making it a crucial tool for enhancing academic life and campus mobility. Because of its creativity and adaptability, it is a forward-thinking option for UTAR community which ensure that it will remain relevant and helpful as campus needs change,

Chapter 2
Literature Review

2.1 Review on existing technology

Traditional approaches that rely on cash or paper tickets usually cause delays, frustration, and higher operational costs, particularly during busy times. Modern public transportation systems are gradually introducing contactless and cashless payment methods to boost efficiency, reduce boarding times, and enhance user experience. In this context, researchers have suggested and taken into practice digital solutions, such Quick Response (QR) codes. Massive amounts of data recorded in a two-dimensional square matrix code can be quickly decoded as a result of QR code technology [2]. A study suggested using the QR code in a public transit payment system, where the fare is immediately deducted as long as the wallet is linked to their bank account [3]. A payment receipt will then be provided to users via SMS after the transaction and trip information were recorded in a database. Moreover, a QR-enabled smart transit system is suggested to make ticketing, entry, and exit simple [4]. Passengers must scan their QR boarding card both when boarding and when departing in order to ensure accurate tracking.

In addition to QR code technology, figure 2.1.1 illustrates how the global market for NFC payments has been growing over time and is expected to continue growing until 2032 [5]. The reason of this wide adoption in all around the world is because NFC is a wireless technology that can enable fast contactless payment and data transfer between two devices over a relatively small distance. An NFC card-based shuttle payment system was proposed in a study, and web-based portals and a mobile application were being created to help users manage their financial operations [6]. Dual-payment methods were suggested as an extension of this idea to improve the bus system's flexibility and convenience. An integrated payment platform using NFC as the primary payment mechanism and QR code as a backup was presented in this paper as well [5].

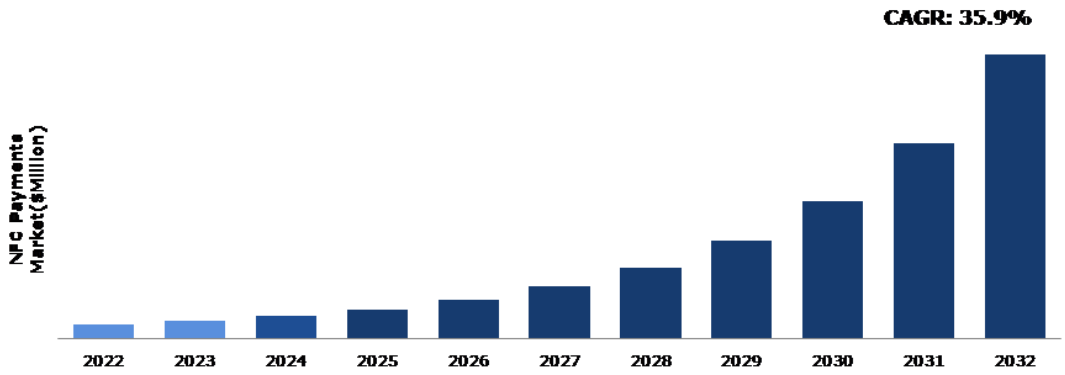


Figure 2.1.1: Global NFC Payments Market Analysis

Source: [5] "NFC Payments Market size, share, growth," Research Dive, Jan-2024. [Online]. Available: https://www.researchdive.com/9173/nfc-payments-market.

Furthermore, in modern transit systems, real-time vehicle tracking has emerged as a crucial component that tackles the enduring issues of timetable unpredictability and passenger discontent. Transportation operators can provide accurate, up-to-date information regarding vehicle locations, arrival times, and delays by using GPS technology. This improves passenger confidence in the transit system, reduces perceived wait times, and facilitates trip planning. In order to lower the cost of hardware installations and give users accurate location information, a study suggested creating a role-based user access campus bus tracking program embedded with GPS technology [7]. Administrators can keep track of system activity and maintain bus schedules with this application to ensure operational efficacy. While drivers may send messages, view assigned routes, and continuously update their location, students can use essential features like real-time bus tracker, instant notifications, and direct communication with drivers for updates on delays or route modifications.

Besides, a study published in [8] used GPS technology to build a college transportation system. The main feature of this system is that it created an Internet of Things gadget for recording seat capacity using a Raspberry Pi and webcam. Students can use this technology to get the most recent bus conditions, allowing them to plan ahead and prevent needless wait times. In the meantime, a different IoT-enabled public transportation system that combines GPS tracking, seat reservations, route suggestions, emergency warnings, and driver-side validation through sensor integration was set forth as well [9].

Another study also presented an intelligent school bus tracking system that uses GPS and QR code authentication to record the boarding of passengers [10]. In order to provide a real-time presence record for security purposes, students must scan a QR code upon boarding and alighting. The highlight of this

proposed system is that it analyses traffic conditions using machine learning techniques like Random Forest and SARIMA.

2.2 Limitations of proposed system

According to the examined articles, the bus transit system has made great strides and provided insightful information. Nevertheless, the suggested alternatives still have certain drawbacks. First, research on the efficacy of QR code payments in bus transit systems revealed that, in comparison to traditional methods, this method significantly increases the time it takes for people to board buses [11]. This is due to the fact that QR payments heavily depend on internet access. Poor connectivity might further delay the scanning and validation or even prevent payment from being made. Thus, it can frustrating the transit experience more. In order to offer inclusive and flexible payment experiences for a variety of user groups, very few works suggest hybrid models that include several choices, such as NFC, QR, and in-app wallets.

Furthermore, as noted in [8], IoT frequently strongly depends on additional hardware infrastructure, even if it has greatly aided in the creation of contemporary and intelligent bus ticketing systems. Although it is theoretically feasible, but this approach is less useful for smaller organizations, such as university campuses. In addition to adding to the cost, a reliance on hardware increases the risk of technical malfunctions and downtime, which lowers system reliability overall. However, using the built-in GPS and NFC capabilities of contemporary mobile devices is still an undiscovered way to reduce reliance on hardware. While smartphones have become such an important part of everyday life, a mobile-first approach might minimize implementation costs, speed deployment, and ensure more accessibility. Using mobile technology also makes it easier to modernize systems, connect to other digital campus services, and enhance user experience via unified apps.

2.3 Review of existing system

2.3.1 Justnaik mobile application

Justnaik is an application that has similar concept of this project. Payments, real-time bus tracking, passenger-driver conversation, and journey planning are all combined into one smartphone application. The option of QR-based cashless fare payment incorporated into the system is one of its major advantages. Users must first reload their wallet before using the app to make a payment; the fare will then be immediately charged from it. This makes it possible for people to quickly and conveniently finish transactions, reducing the need to handle cash. This application also has a feedback feature that facilitates effective communication between management and users. Additionally, real-time tracking for every trip is supported by the system.

However, the app displays a timeline view on the information page, where each stop changes as the bus moves along its route, rather than displaying the bus moving in real time on the map. Users could become confused about whether the trip has started as a result. Plus, the system lacks a seat update option, which could interfere with students' plans and result in longer wait times, particularly during busy times.

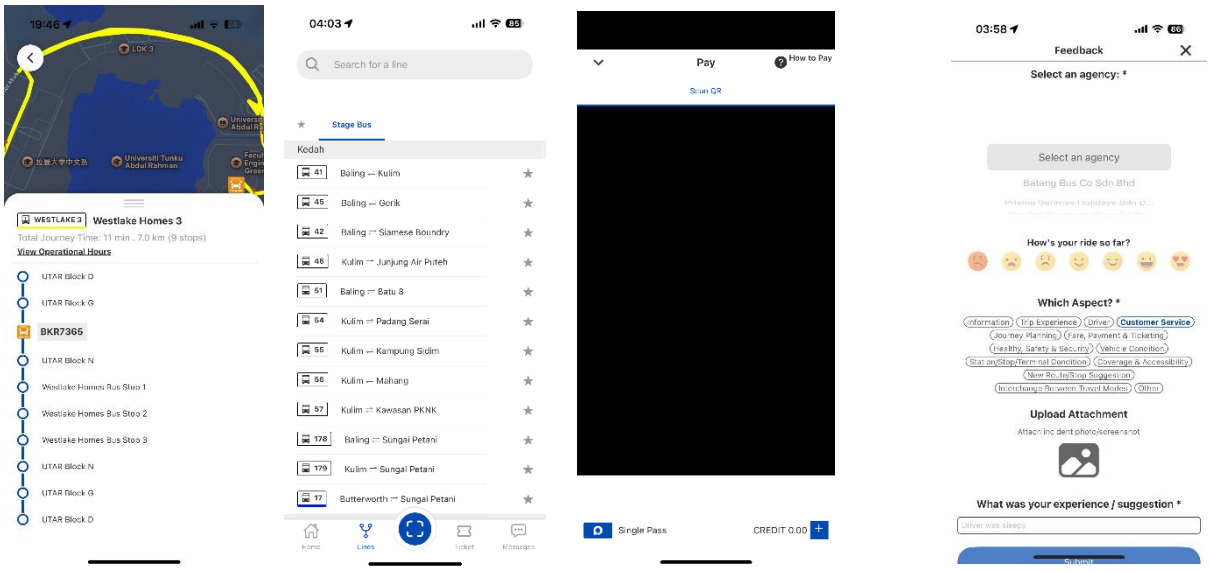


Figure 2.3.1 Justnaik mobile application screenshots

3.3.2 Intercampus Shuttle mobile application

The Intercampus Shuttle app is used by Arizona State University (ASU) to run their bus service and give students live transport tracking. Students may easily check schedules and keep track of bus locations with this application's easy-to-use layout, which has three main functions on the bottom navigation bar. On the home page, each route is indicated in a different color for user convenience. On some routes, users can also see the details of each stop and request alerts before the bus arrives.

Although the application provides real-time bus capacities information, users may find it confusing as it only displays the data as a percentage of bus occupants. Additionally, there is a recurring issue with the schedule page where the message "No schedule available" displays on every route, even when trips are active. For instance, as seen in the Figure 2.3.2 below, the Phoenix route is operational, but no schedules are shown. This issue arises on all accessible routes. User convenience may also be impacted by the system's lack of a fare paying capability.

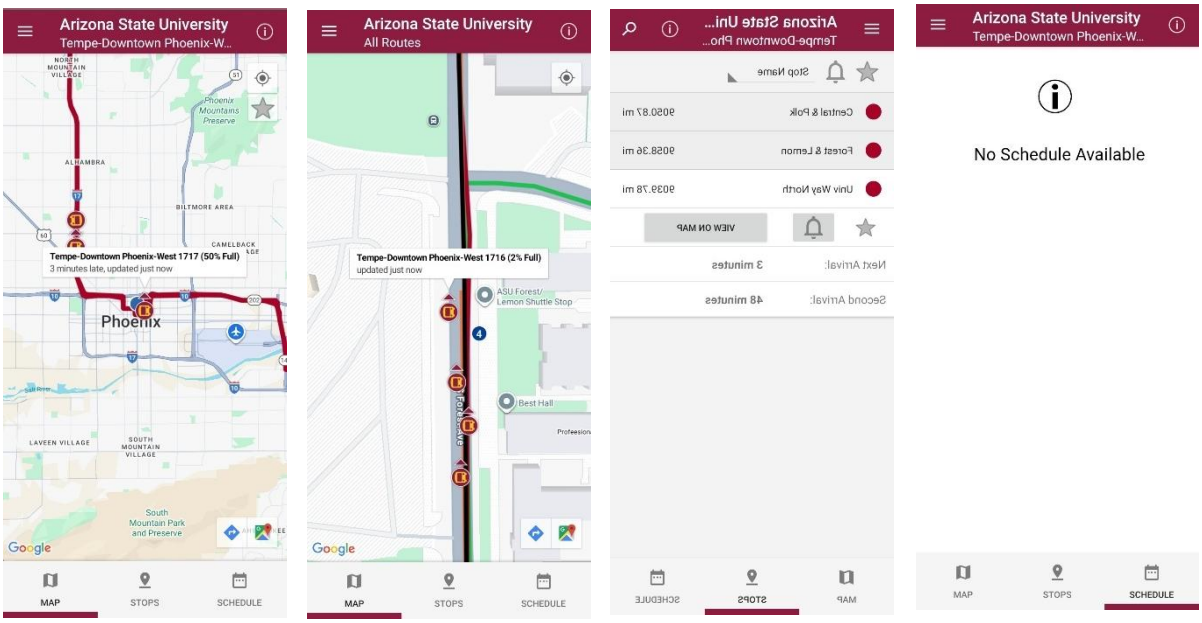


Figure 2.3.2 Intercampus Shuttle application screenshots

2.3.3 PSU Smart Campus web system

The PSU Smart Campus system is used by Songkhla University to offer their students a simple interface for accessing campus bus routes and stop locations. The straightforward and easy-to-use interface, which enables quick and easy service access is the key strength of this system. Before the bus's locations can be tracked, users must first search for their desired stops. If there is an active trip at the chosen stop, bus routes will then be shown in red for easy viewing. The system is simple enough with only two pages, while one page is used for live tracking and another is static schedule page for fast overview.

Despite that, this system's simplicity could also work against it. When there is no trip in progress, the notification "No bus is coming" may not be enough information for users to make plans. Furthermore, it is only available as an online application, which restricts flexibility and accessibility compared to smartphone applications that students can use while on the go. A web-based approach requires users to launch a browser, navigate to the portal, and then refresh the page for updates. When compared to other advanced bus transportation applications, its effectiveness is further constrained by the lack of features like intelligent scheduling, payment methods, and notifications. This complicates the user experience, particularly in situations where prompt information is needed, like bus arrival alerts.

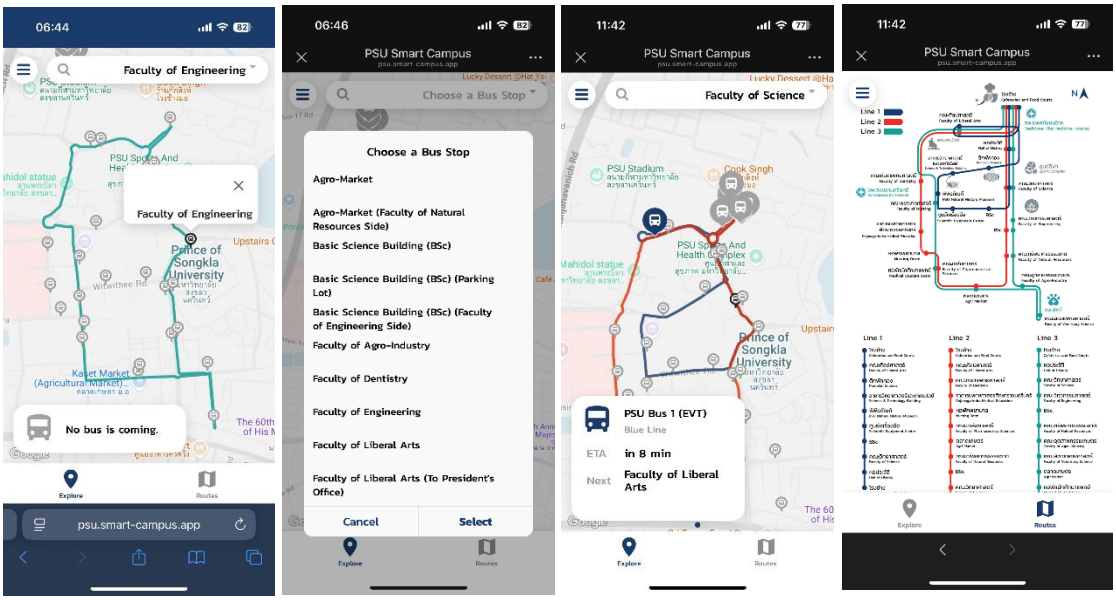


Figure 2.3.3: PSU Smart Campus system screenshots

2.3.4 Comparison of existing system

System/ Function	Justnaik	Intercampus Shuttle	PSU Smart Campus
System type	Mobile application	Mobile application	Web-based system
Live tracking	Yes. Trip updated in journey timeline.	Yes. Map displays a clear location.	Yes. Map displays a clear location.
Schedule update	Yes. Ccomprehensive and dynamic display.	Embedded but inaccurate. Always show empty result when trips are active.	No. Stastic route map only.
Payment feature	Yes	No	No
Feedback	Yes. Built-in communication channel.	Yes. Integrated with third-party source	No
Notification	Yes	Yes	No
System design	Fully functional with accessibility features	Basic design with major features	Extreme simple for fast tracking only

Table 2.3.4.1: Comparison of existing system

2.4 Summary

The analysis of existing campus bus transit applications shows that while these systems have practical features like live monitoring, route illustration, and quick notifications, they also have critical shortcomings that reduce their overall effectiveness. The PSU Smart Bus system, for example, has an easy-to-use interface, but its utility is limited by its web-only platform and lack of advanced capabilities like personalized planning and notifications. In a comparable manner, the Intercampus Shuttle program has integrated tracking and scheduling features, but issues including misleading schedule displays and a lack of payment method integration compromise their dependability. On the other hand, the Justnaik

app, which serves as an inspiration for contemporary shuttle bus systems, demonstrates the potential of a complete solution by combining booking, feedback, QR-based payments, and real-time tracking. All things considered, current systems offer decent work of monitoring and general functions, but they remain below of offering a comprehensive, user-focused framework that includes dependable scheduling, flexible reach between devices, and advanced payment options. These shortcomings highlight the significance of a more complete and inclusive system, such as this designed UTAR Bus Transit application, which provides live trip tracking, smart scheduling with conflict detection, quick and effective NFC-based payment, and also integrated communication channel tailored to a campus environment.

Chapter 3

Proposed Method/Approach

This project implemented the iterative software engineering approach, Agile development process, to ensure that the mobile application is practical, scalable, and compliant with user needs. Firstly, the project objectives were established and all demands were precisely identified in a thorough document during the requirement analysis phase of the work process. Important modules that were identified as incremental deliverables included authentication, schedule management, wallet and NFC payment, OCR timetable processing, lost-and-found module, and feedback management.

Next, high-priority tasks were chosen during the planning phase of each iteration based on their dependency and impact. In order to guarantee that each sprint could obtain measurable results, scope was restricted in every cycle. After that, design and implementation were carried out iteratively. Instead of isolated code completion, each module was built as an end-to-end part so that early verification of actual user flows can be done in advance.

Lastly, functional testing was conducted to evaluate the performance of the implementation. The results of these assessments were reflected back into backlog refinement, allowing bus and issues to be handled quickly in every cycles. It also helps to decreased integration risk and increased system stability.

Once the system was completed, the project moved on to integration and validation phase. Cross-module consistency, error management, and final evaluation against project goals were the main focuses of this phase. With this Agile framework, it allows for controlled delivery while keeping flexibility for refinement, resulting in a workable system that met the set functional goals within project limitations.

3.1 System Architecture Diagram

The bus transit application was designed in a three-layer architecture to maintain consistency and clarity in roles. The top is presentation layer, which it handles the interface design and user interaction. The user interfaces were built in Flutter framework, allowing cross-platform development. It contains different clients, which are the student, driving, and administrative interfaces. Role-specific interactions are the main focus for each client. For example, while all users share some interfaces like authentication screen, but only driver role can access to the trip starting screen.

Next, the core component, application layer which allows for communications and activities happened. It was developed using Express and Node.js to make Socket.IO and REST APIs available, providing standard client-interaction and real-time data exchange. For instance, when students scan their NFC pass, this layer will check for the pass validity, deduct fare and send information to linked clients. Additionally, this layer links to other infrastructure when necessary, including email services for outbound notifications, map services for real-time bus location tracking and Stripe for wallet top-up payment. By serving as the coordination hub, this layer guarantees that rules are applied uniformly throughout the user processes.

The last layer is data layer, which all operational data are stored and maintained by MongoDB. It ensures structural integrity by defining schemas, validating fields, indexing, and establishing entity relationships. Users, schedules, trips, cards, wallets, wallet, timetable and so on are examples of core entities of this system. While the Application Layer defines business rules, all final state changes are stored to the Data Layer, allowing for transparency, auditability, and recovery.

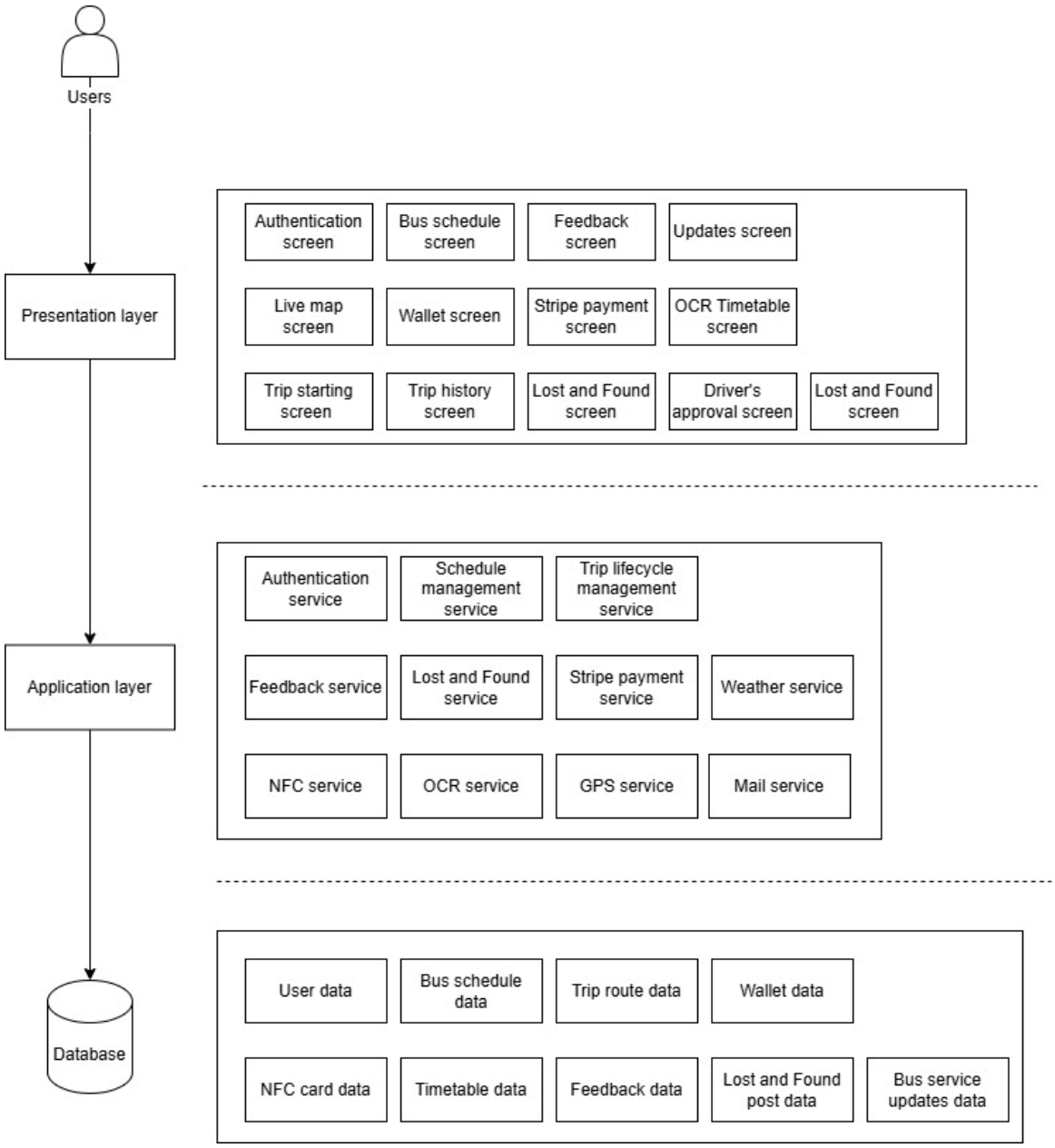


Figure 3.1.1 System Architecture Diagram

3.2 Use Case Diagram

Students

In this application, students can sign in to their existing account or register a new one and reset their password if forgotten, ensuring that only authorized users can access personalized features. Upon login, students have the access to bus schedules and live tracking of bus locations through map integration. Then, students can view the real-time availability of seats on each bus in the map. The information is updated dynamically based on passenger activity, helping students decide whether to board a particular bus. Students can also upload their timetable, and the system will extract the information automatically using OCR technology. Thus, the application can match students' class and bus departure time to suggest the best riding trip for students. These help students to plan their journeys efficiently and reduces waiting time.

Besides, students can manage their digital wallet for all bus transit activities, including top up their wallet through online payment integration, apply for an NFC-enabled pass, report a lost card, and transfer balance to other users. These features support seamless and cashless transactions for bus fare payments. With this application, students are able to browse and claim for lost and found items. It facilitates communication between users and administrators to recover lost belongings. To further enhance the travel experience, students can submit feedback such as complaints, suggestions, or compliments. This helps improve the overall quality of the bus service. Also, students can view the latest announcements and updates related to bus services, such as schedule changes, delays, or important notices.

Figure 3.2.1 Student Use Case Diagram

Driver

In this application, drivers with secret code can register account and reset their passwords to securely access the system. It ensures that only verified drivers can perform operational tasks. Next, drivers can view their assigned schedules and trip details, including routes, departual and arrival time. It helps drivers stay informed about their responsibilities. They can start a trip to send periodic location updates to the backend with GPS service enabled, allowing students to track the bus in real time. Drivers can also activate the NFC payment feature, enabling students to tap their NFC pass to pay for bus fares, which supports fast and contactless transactions. Then, drivers can end trip and view records of completed trips. They can provide feedback or suggestions regarding bus operations as well to improve system efficiency and operational workflows.

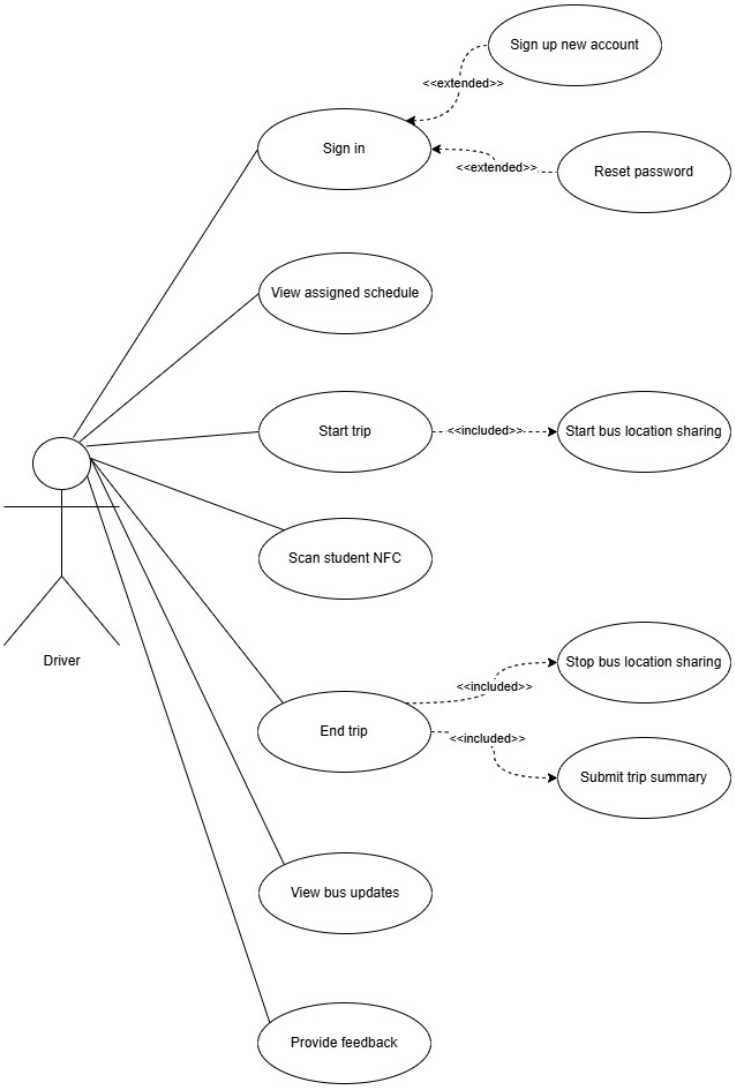


Figure 3.2.2 Driver Use Case Diagram

Admin

In this application, admin can log into the system using pre-created accounts from backend, where the registration is restricted to maintain system security and control over administrative access. In this application, admin can verify driver registrations and manage the list of approved drivers. It ensures that only authorized drivers are allowed to operate within the system. They can manage the bus schedule by creating, updating or deleting the trips, as well as assigning trips to drivers. It is essential for organizing and managing transportation services. Besides, admin can view completed trip records and analyze operational data for monitoring and future decision-making. Apart from schedule management, admins can manage bus service updates or announcements as well. It ensures that important information is communicated effectively to users. They can also view, respond to, and resolve user feedback. It helps maintain service quality and address user concerns efficiently. Admin also has access to manage lost and found items, including creating posts and verifying claims made by students. It ensures proper handling and recovery of lost items.

Figure 3.2.3 Admin Use Case Diagram

3.3 Activity Diagram

Schedule management activity

This diagram illustrates admin schedule operations including crate schedules with trip details, editing or deleting schedule, as well as assigning drivers with conflict checking mechanism.

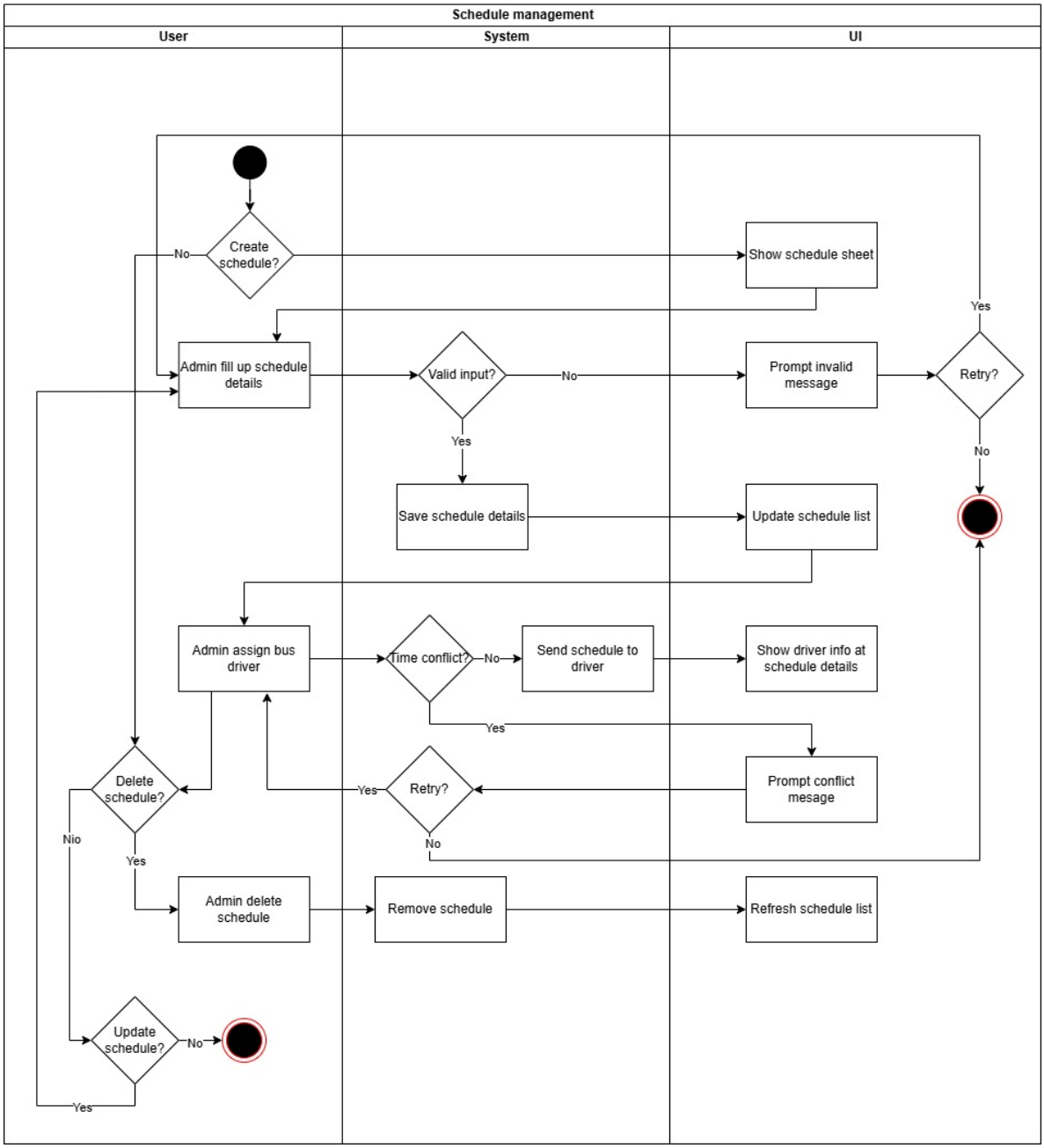


Figure 3.3.1 Schedule Management Activity Diagram

Trip lifecycle management activity

This figure shows the process of initiating the bus trip and sharing real-time location. To enable location sharing, users have to enable the GPS function in their devices, so that the Google Maps can extract the precise location for real-time bus tracking.

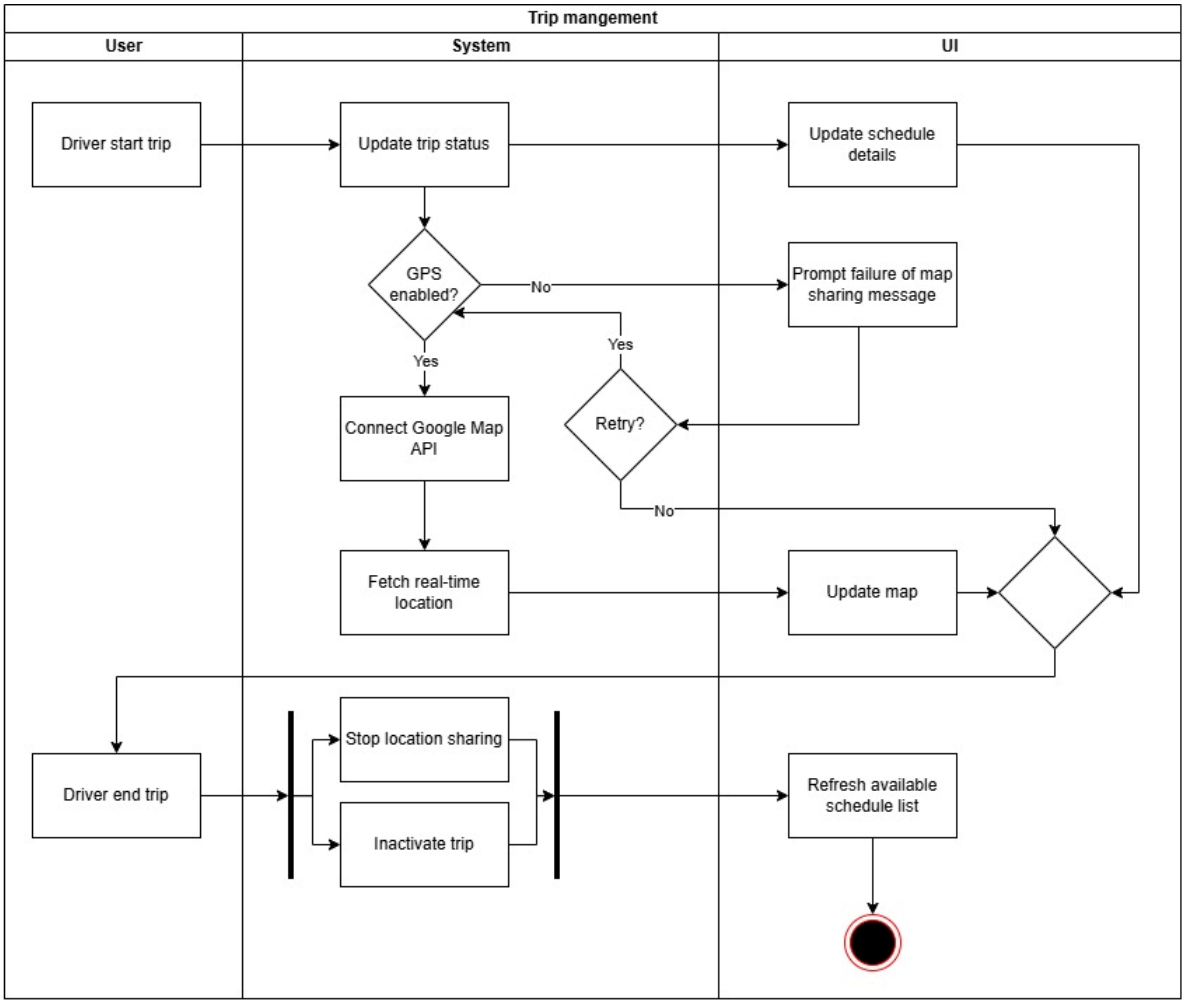


Figure 3.3.2 Trip Lifecycle Management Activity Diagram

NFC payment activity

This illustration demonstrates how students scan an NFC card to initiate the Tap-to-Pay process. To start the process, driver has to enable the NFC feature. The fare payment is only successful when NFC function is enabled and the pass is valid. Then, the system resets to wait for the next tap, sends a payment request to the backend, and shows success or failure message.

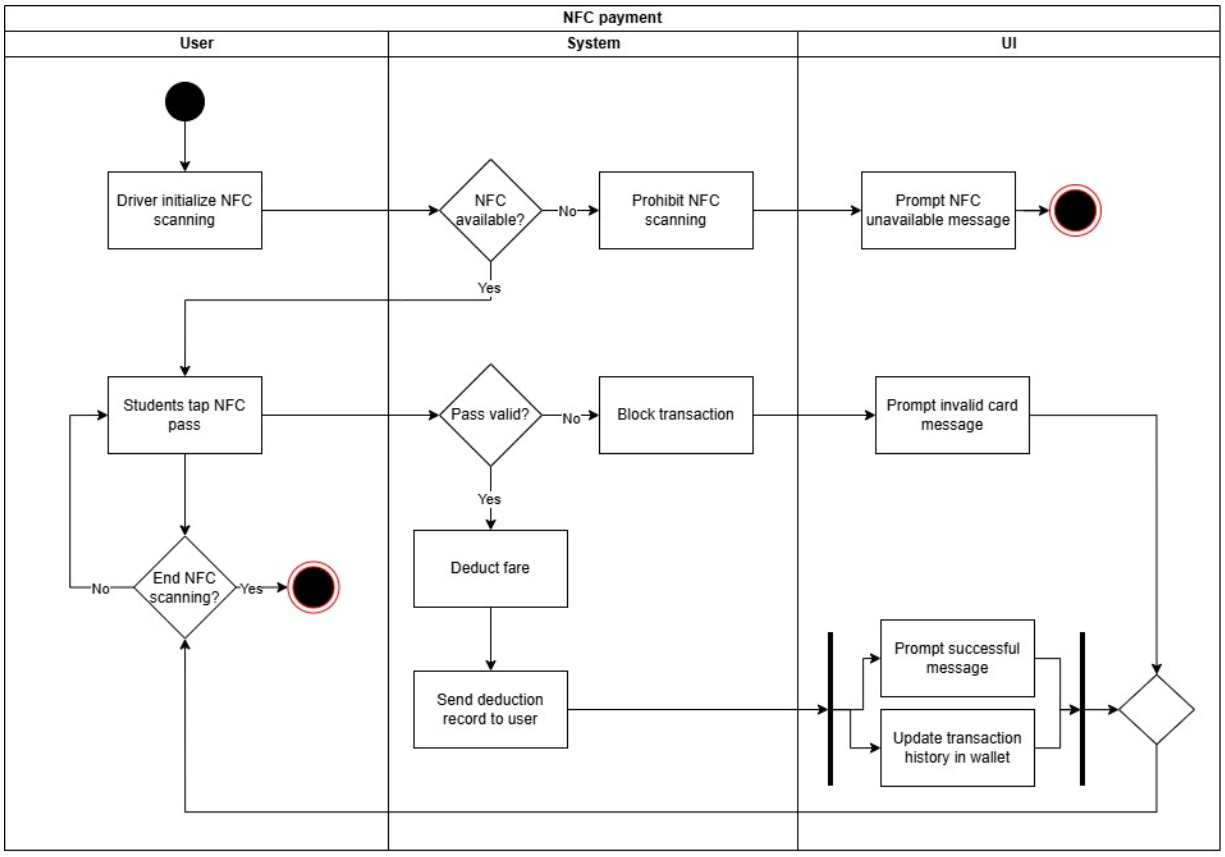


Figure 3.3.3 NFC Payment Activity Diagram

Top up wallet activity

This diagram illustrates the wallet reload process where after users enters an amount, the system creates a Stripe payment intent, display payment sheet and confirms payment with the backend. Then, the wallet balance and history are refreshed to updates the latest details.

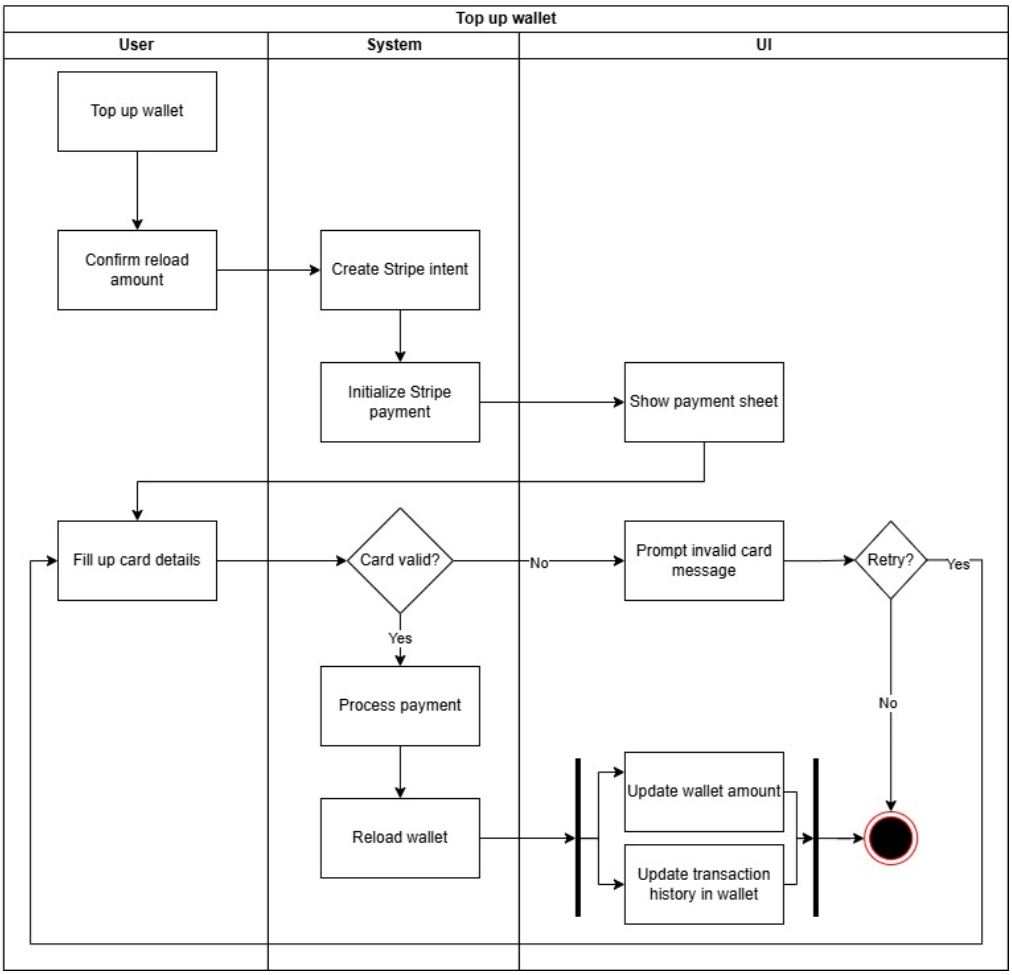


Figure 3.3.4 Top Up Wallet Activity Diagram

Transfer balance activity

This diagram shows a peer-to-peer wallet transfer in which users enter the recipient's email address and intended transfer amount. Then, the application validates the inputs, calls the transfer API, and updates the balance as well as transaction history upon success.

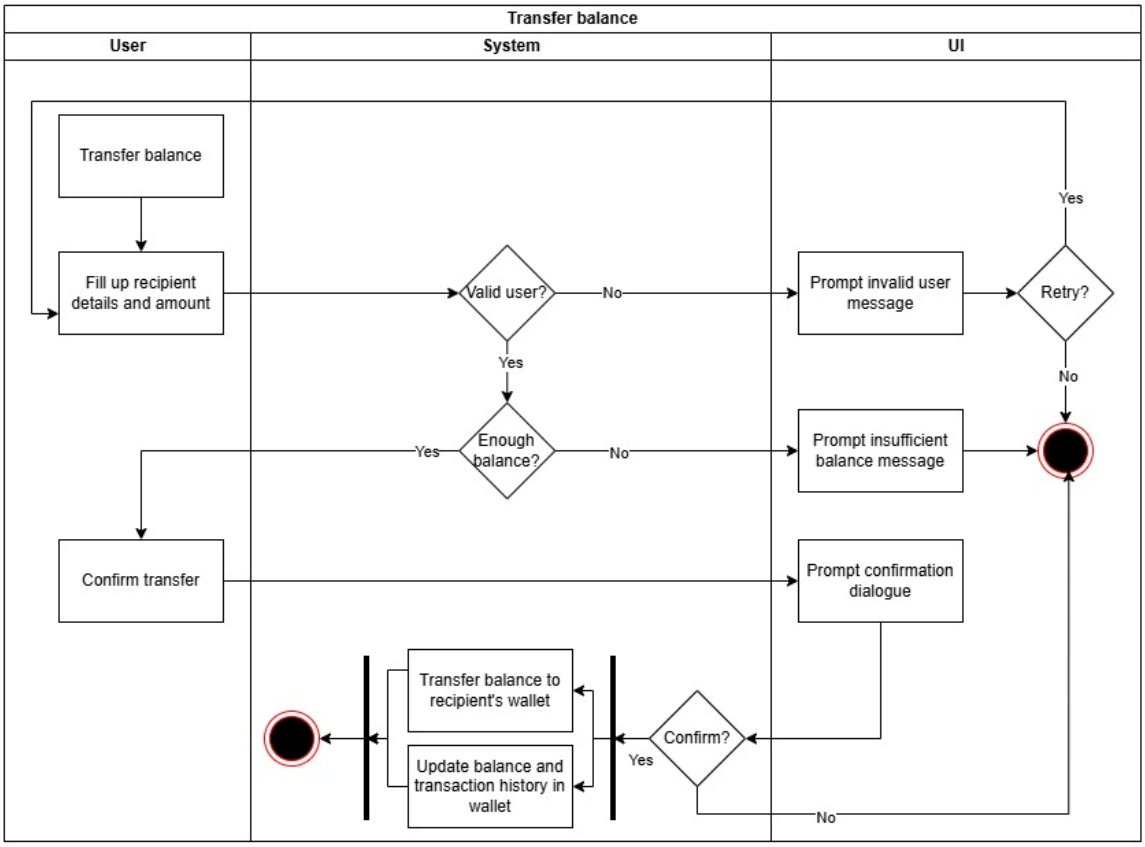


Figure 3.3.5 Transfer Balance Activity Diagram

OCR timetable activity

This figure shows how the application extract a timetable information from an image via OCR technology. Firstly, users have to upload their timetable image clearly, and the system will validate the image. Then, OCR parses class entries and saves to the backend for users to examine the information. This feature is used for smart scheduling and send upcoming bus reminder to users.

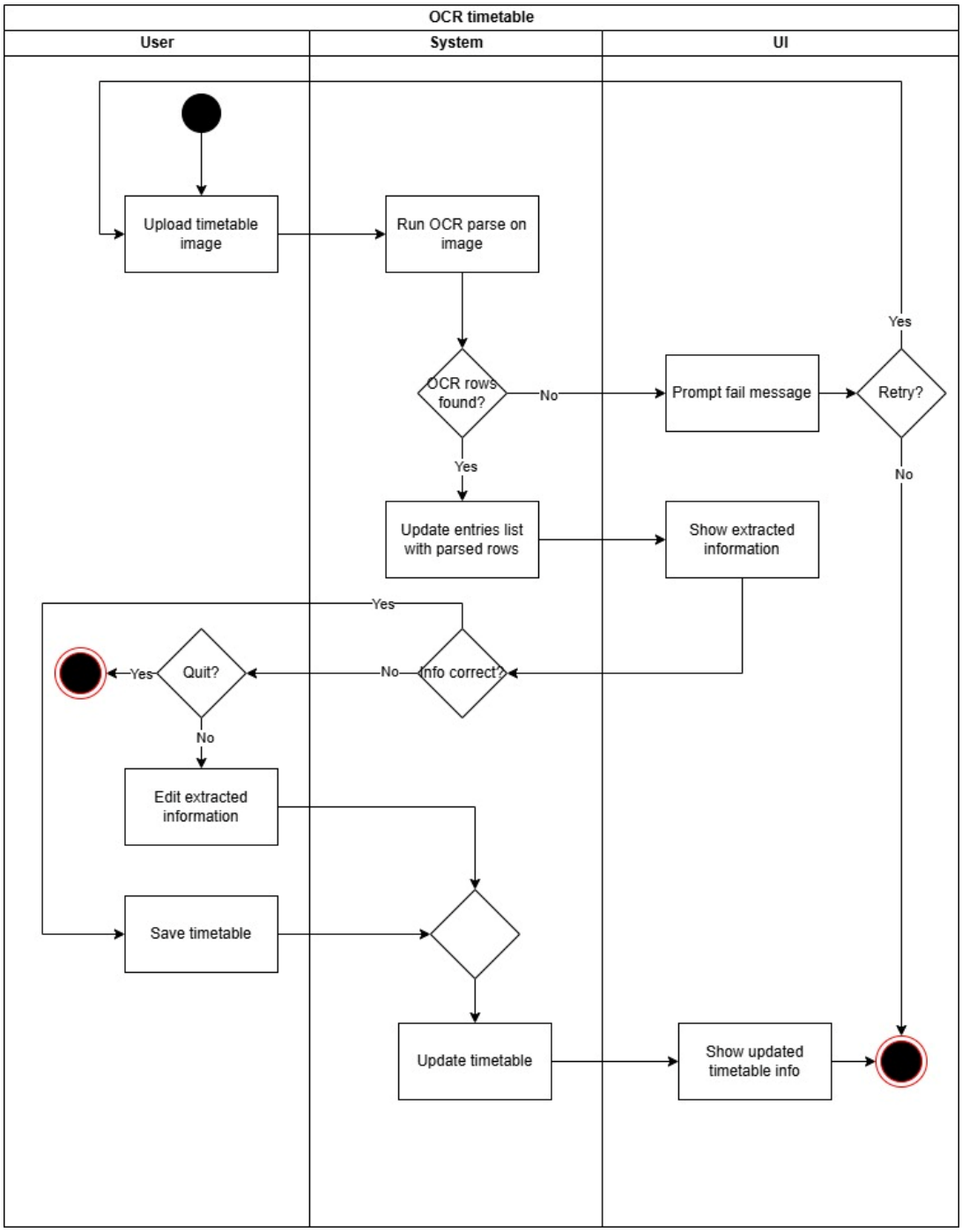


Figure 3.3.6 OCR Timetable Activity Diagram

Lost and found activity

This diagram demonstartes the entire lost-and-found workflow, where admin can create lost-item posts, while students can review and claim their lost items. Then, admin can approve or reject claims, and the approved students confirm collection to close the case.

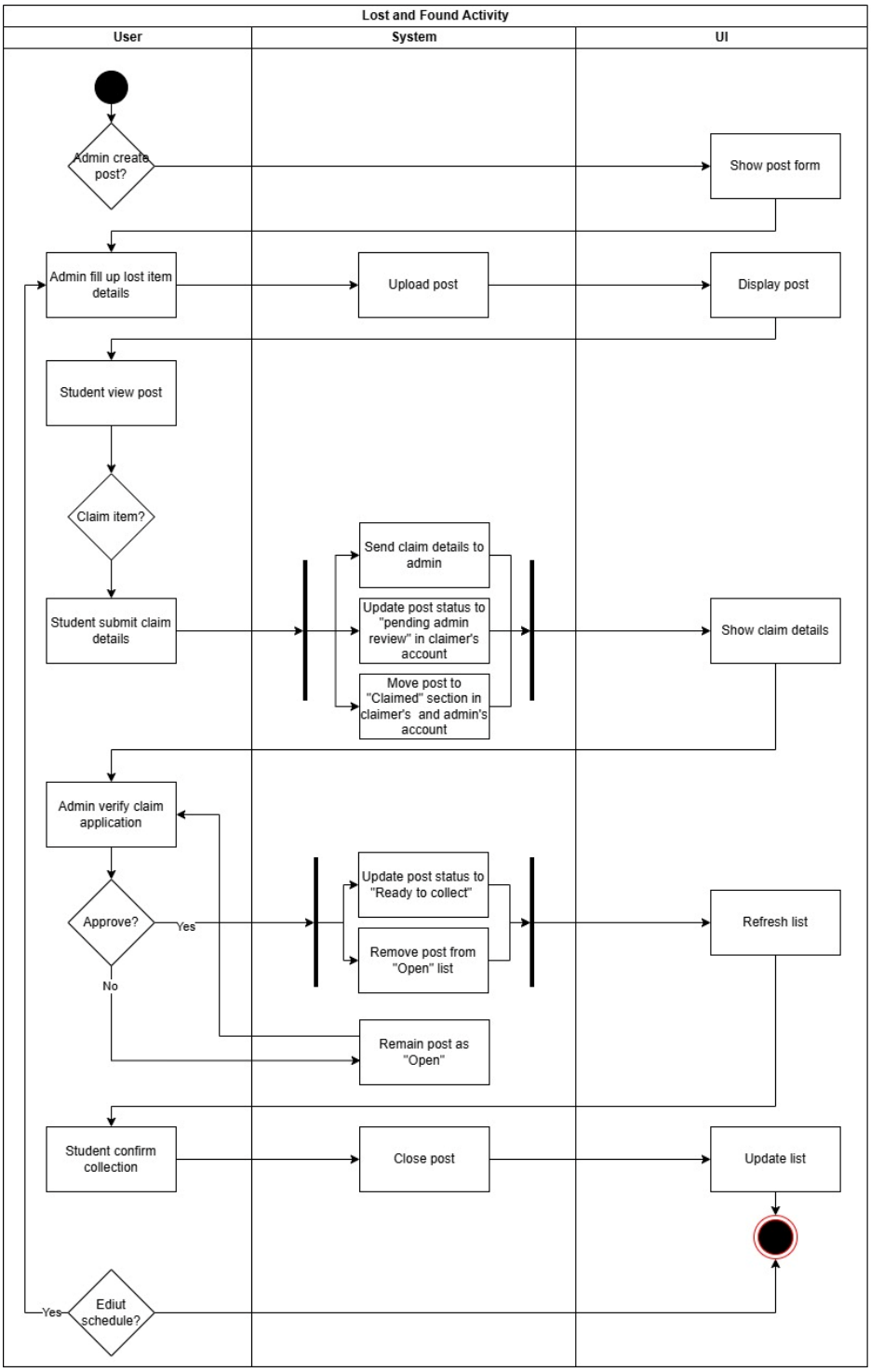


Figure 3.3.7 Lost and Found Activity Diagram

Feedback activity

This figure shows the feedback lifecycle of this application, in which both students and drivers can send feedback to imrpove the bus service. Theb, the system saves feedback as open status, and closes it until admin resolves the ticket with optional notes. Users may also track the status and responses in their account.

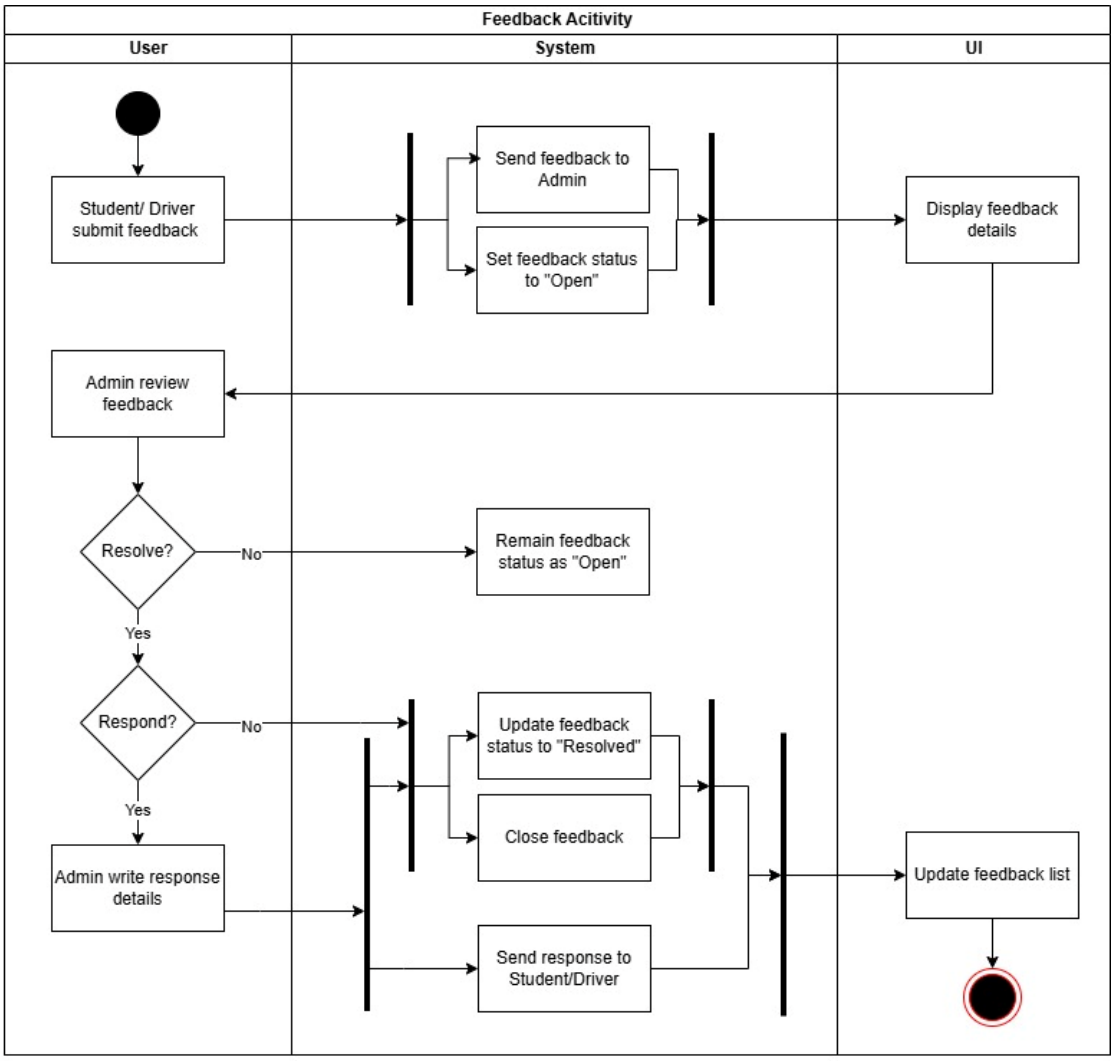


Figure 3.3.8 Feedback Activity Diagram

Chapter 4

System Design

4.1 System Requirement

4.1.1 Functional requirements

1. Student
    i. Authentication module: Students can sign up new accounts, log in and change passwords
    ii. Wallet module: Students can perform financial activities like reloading wallets, link NFC card, reporting lost card and transferring money
    iii. Bus tracking module: Students can view live maps to track the bus location and remaining seats count
    iv. OCR timetable module: Students can upload timetable and retrieve the information automatically
    v. Bus reminder module: Students can receive updates when bus schedules are matched with their timetable
    vi. Lost & Found module: Students can browse and claim for lost items
    vii. Feedback module: Students can submit feedback to and receive reply from management
    viii. Service updates module: Students can receive the latest announcements from management

2. Drivers
    I. Authentication module: Drivers can sign up new accounts, log in and change password
    II. Schedule list: Drivers can view assigned schedules and their details
    III. Trip management: Driver can start assigned trip and enable GPS to send constant location updates
    IV. Trip history: Driver can view the list of ended trips and details
    V. Feedback module: Drivers can submit feedback to and receive reply from management
    VI. NFC scanning: Drivers can start NFC features for students to pay transit fare
    VII. Service updates module: Drivers can receive the latest announcements from management
3. Administrator

I. Authentication module: Admin can only log in, prohibit admin account creation
II. Driver management module: Admin can verify and approve driver account registration
III. Schedules management module: Admin can manage the bus schedules and assign drivers to each of them
IV. Trip history: Admin can track the ended trips and details
V. Updates management module: Admin can create different types of announcements and decide the access level
VI. Feedback module: Admin can receive and response to user feedback
VII. Lost & Found management module: Admin can create posts and verify student's claim

4.1.2 Non-functional requirements

1. System responsiveness
    i. Location updates must be reflected in the app within 5 seconds of change.
    ii. Seat count updates should synchronize immediately after boarding or alighting events

2. Security
    i. All user data must be securely stored with standard encryption protocol

3. Availability
    i. The system should maintain at least 99% uptime to ensure continuous service during peak period

4.2 User Flowchart

Student

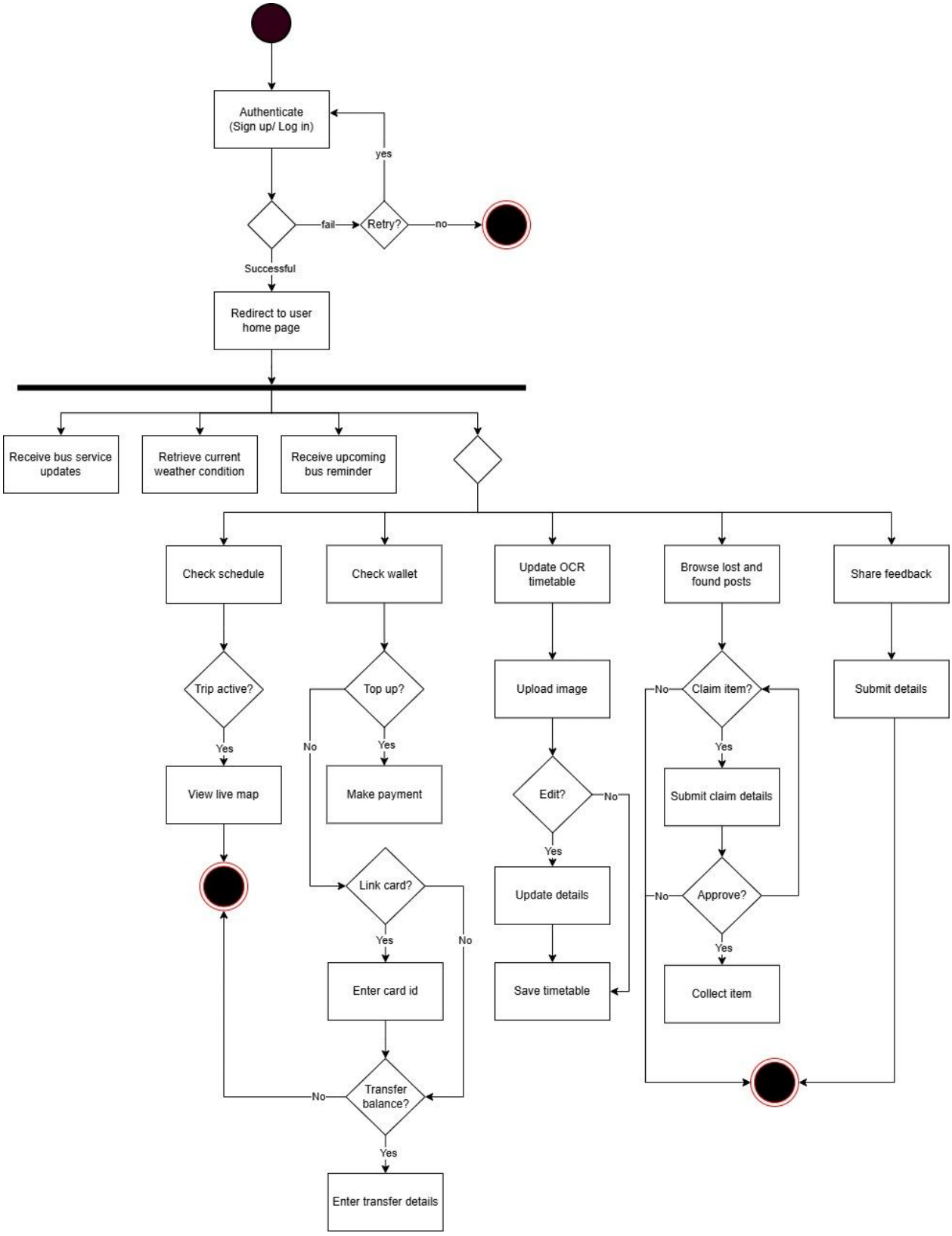


Figure 4.2.1 Student Flowchart

Driver

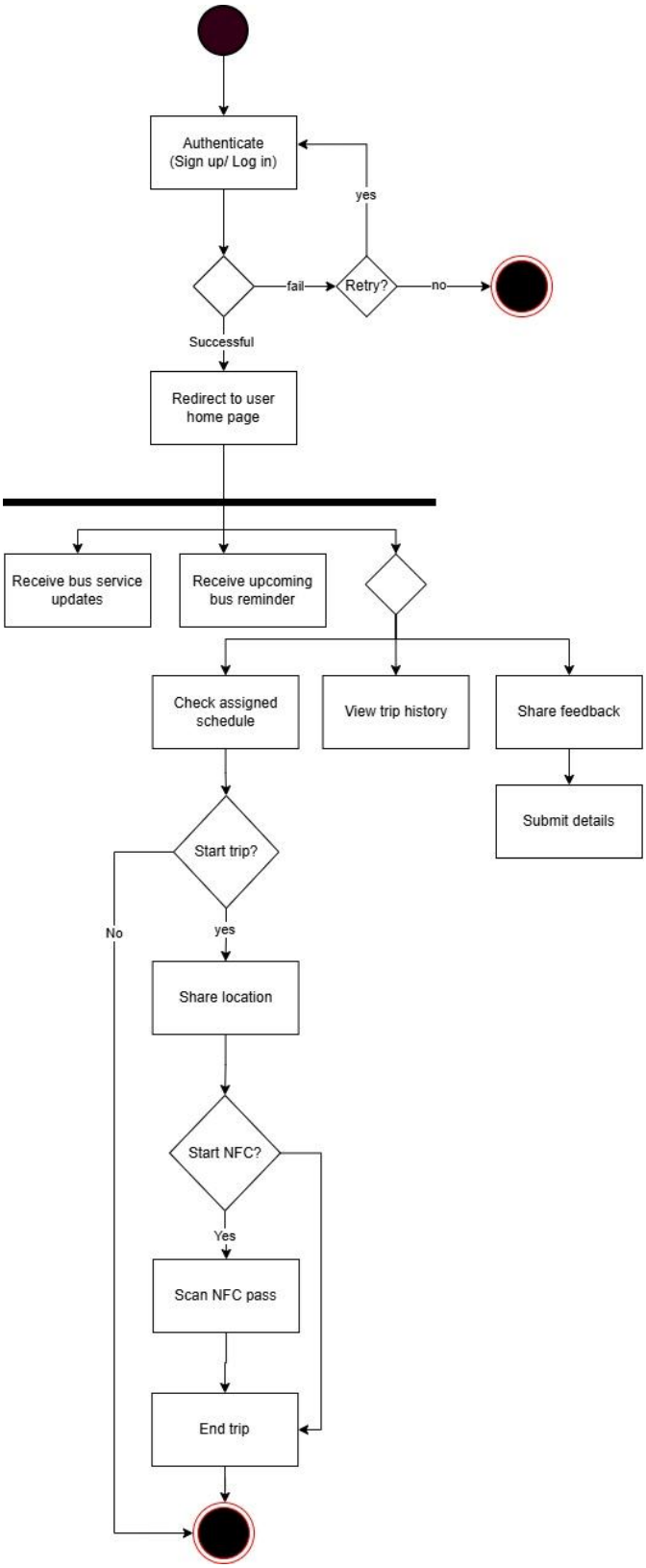


Figure 4.2.2 Driver Flowchart

Administrator

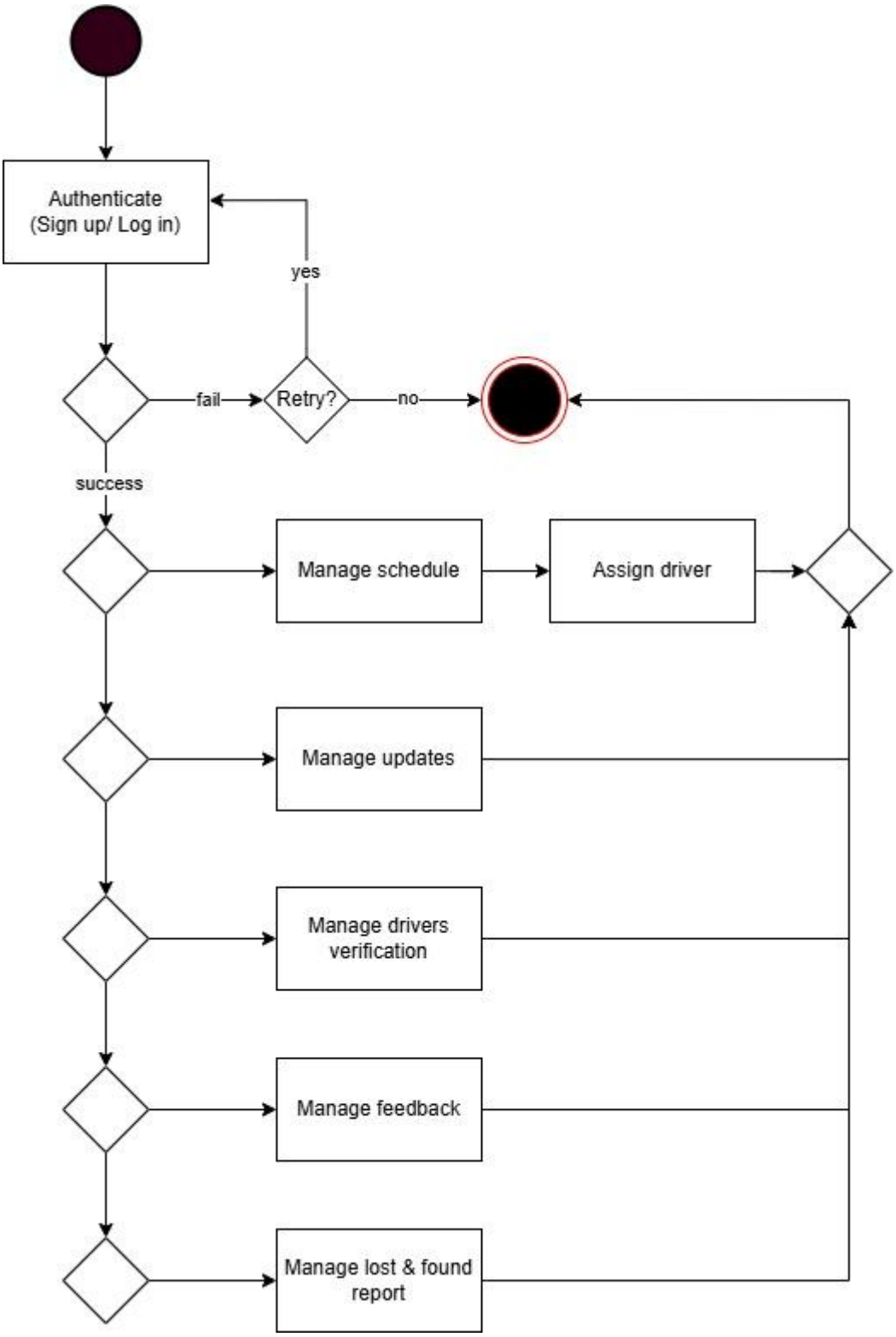


Figure 4.2.3 Admin Flowchart

4.3 System Block Diagram

The block diagram illustrates the high-level design of this project. The mobile application serves as the central link between external services and internal modules. The application's features consist of user identification, schedule administration, location tracking, wallet, NFC payment, lost and found, feedback, and OCR timetable processing. These modules offer user-facing features and manage the primary business operations of the system. Third-party services including MongoDB for data storage, Mailgun for email notification, Google Maps for real-time location sharing, Stripe for payment gateway, and Weather API for weather-related data are all integrated with this application from external sources. The two-way relationship indicates that the mobile app sends queries to these services and receives responses, allowing for real-time, data-driven activities across the platform.

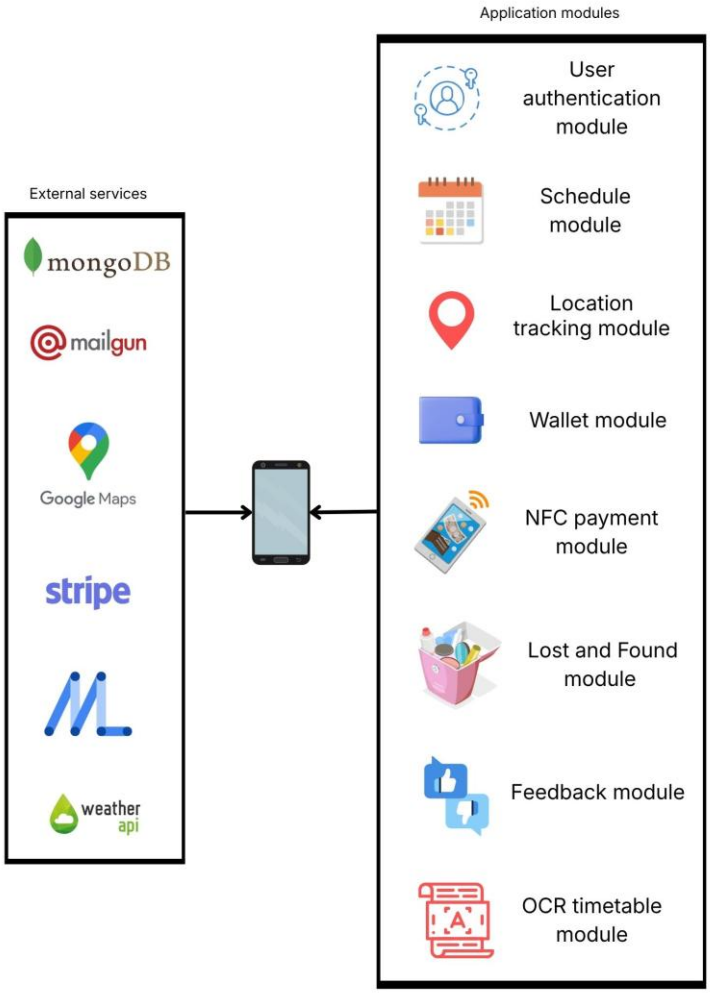


Figure 4.3.1 System Block Diagram

4.4 Context Diagram

Figure illustrates the system context of this bus transit application. The central component is the integrated application platform that manages campus bus operations and student services. The service operation is maintained by administrator, who manages schedules, publish bus service updates, verify driver accounts and handle passengers' lost items. Besides, students utilize the system to examine operational information and ticketing-related functions, such as checking available schedules, uploading timetable for smart schedule matching reminder, making payments, claiming lost items, and providing feedback. In exchange, the system supports passenger awareness by providing real-time trip status and bus location updates. In order to facilitate live monitoring and service coordination, drivers can check assigned bus trips or relevant operational information from the system, and continuously update bus location to system.

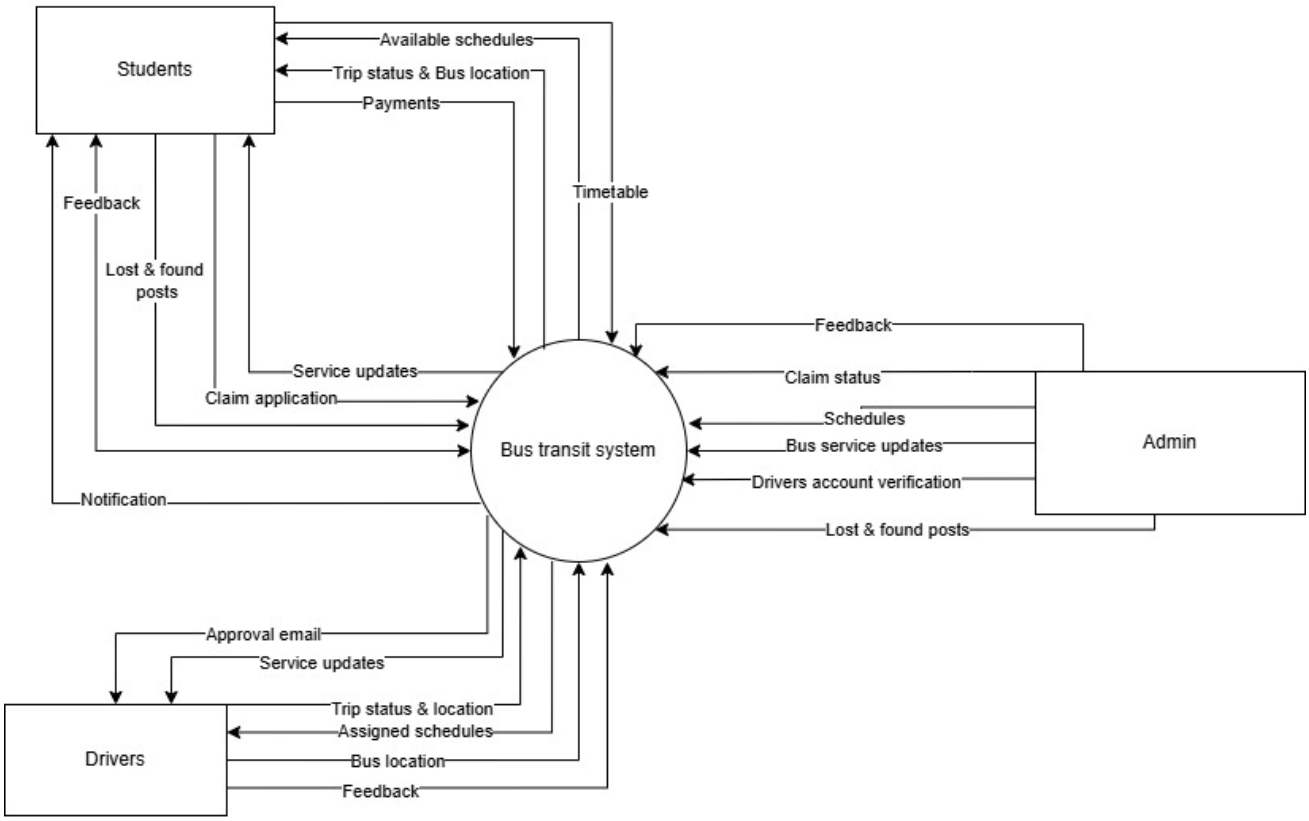


Figure 4.4.1 Context Diagram

4.5 Database Design

In this system, user is the primary entity, which includes user credentials, role, verification and others. A bus run's route, departure and arrival times, embedded stop objects with names and locations, bus identity, capacity, active flag, and notes are all described in the Schedule. Every driver is optionally to be assigned to different schedules, as long as there is no time conflict. The Trip is an entity that belongs to a certain schedule and is handled by only one driver.

Besides, financial related data are extended from users. Each students have their own wallet to make transactions such as balance reload, transfer and fare payment. Transactions may include optional identifiers like trip ID, Stripe payment intent and so on, while card connects a user to an actual NFC card ID. Trip_Ride maintains boarding and alighting metadata, fare charging status, and may refer to a Wallet_Transaction for fare payment process.

Apart from operations, Student_Timetable stores a student's class timetable as embedded entries within a single document per user. In-app messages are stored by Notifications for each user. Feedback records user-submitted queries with status monitoring, and Update is a representation of broadcast announcements made by admin. Meanwhile, Lost_Found _Post and Lost_Found_Claim model for lost-and-found items and student claims, including reviewer connection. Similarly, Card_Lost_Report keeps track for student’s lost card records.

The figure has illustrated how wallets, cards, and transaction rows maintain payment integrity, schedules and trips drive real-time transport logic, and additional entities facilitate communication other campus-life features.

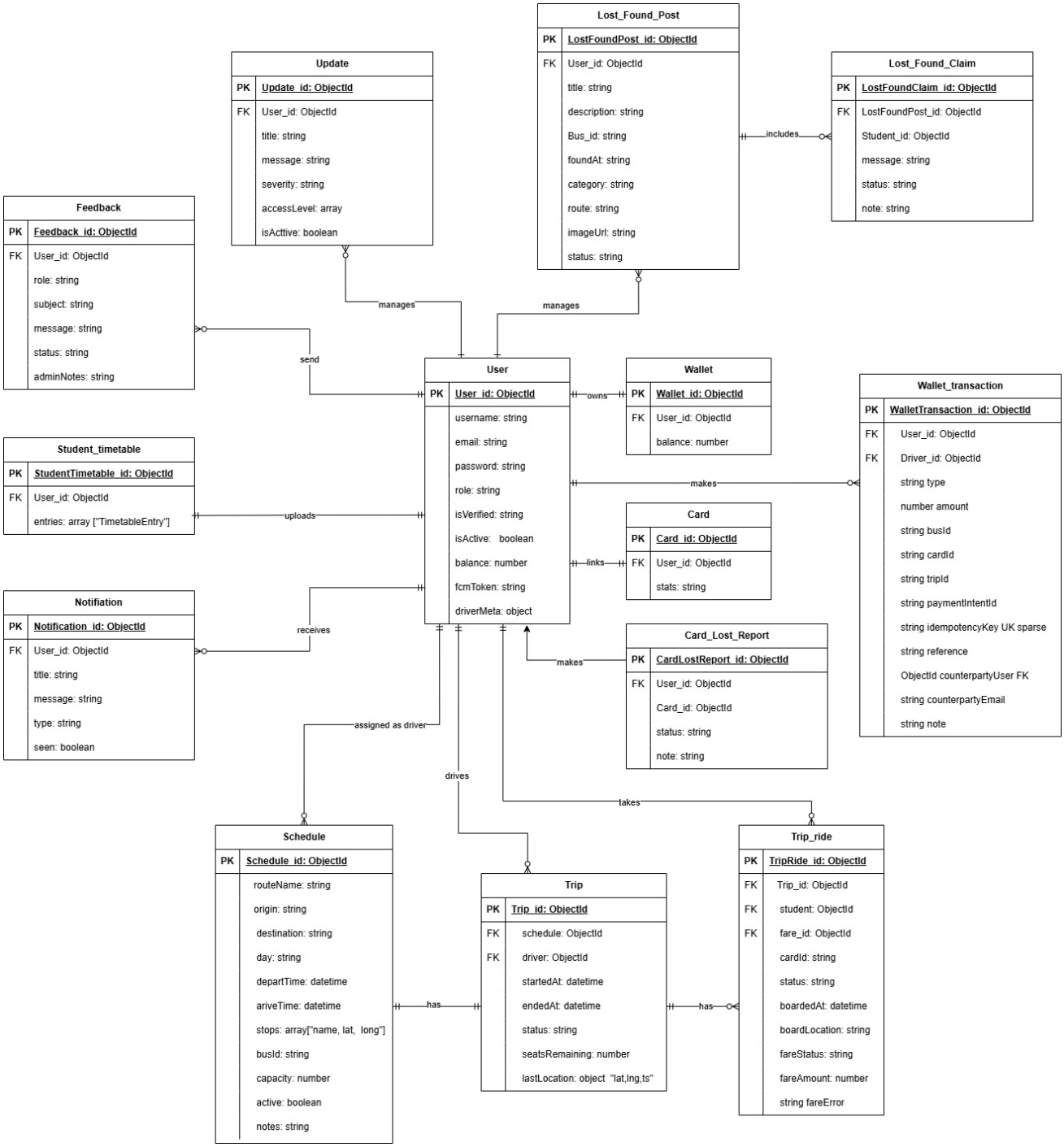


Figure 4.5.1 ER Diagram

Chapter 5

System Implementation

5.1 Required tools

Software	**Description**
Visual Studio Code	An IDE which is suitable for application development via Flutter and Node.js
Android studio	To launce the Android Emulator for system testing directly from computer
MongoDB Atlas	An online database tool to store all data in this application
Mailgun	An email service for sending and receiving updates for driver account status
Hardware	**Description**
Mobile phone	A device with NFC feature for contactless fare payments, and GPS function to update real-time bus location
Computer	A device with a minimum specification of Intel i5 (8th gen+) / Ryzen 5, 16 GB memory and 512 GB SSD
NFC tag	A NTAG213 to store pass ID for simulating the NFC fare payment process
Supporting Technology	**Description**
Node.js	For server-side development to facilitate real-time interaction among applications and database
Flutter	A frontend framework that ensures consistent performance across iOS and Android by allowing developers to create mobile applications with a single codebase
Weather API	To update current weather data for users to take their ride carefully

Google Maps API	A location service for map visualization to offer precise trip tracking
Google ML Kit	A pre-built machine learning API to extract students' timetable information

Table5.1.1 Required Tools

5.2 Setting and Configuration

MongoDB

In this project, MongoDB Atlas serves as the main cloud database with its modelling tool, Mongoose, for asynchronous setting. Firstly, create an account and a project on MongoDB, and connect it to the application by adding the MongoDB URI in .env file. Then, install dependencies and restart the server. Console logs can be created as well to verify if the database was connected successfully.

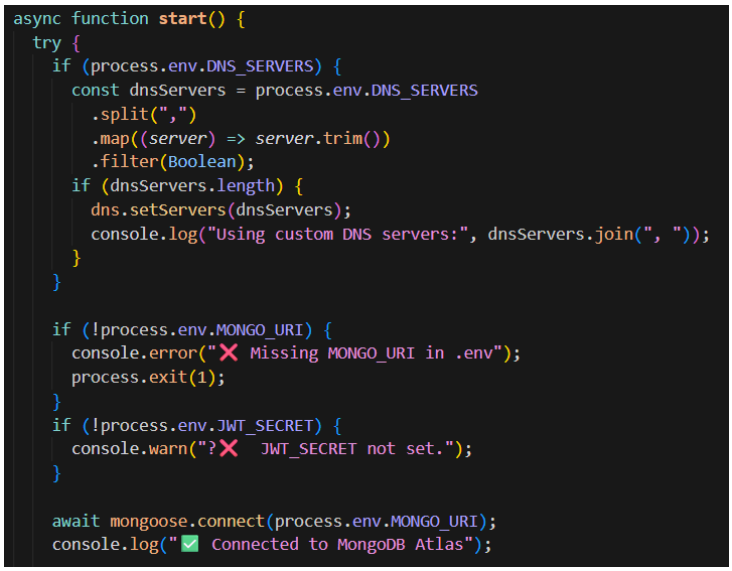


Figure 5.2.1 MongoDB Configuration

Stripe

For the wallet reload feature, Stripe was integrated as the payment gateway. Similarly, a secret key is needed for server to initialize a Stripe client, which can get from the Stripe sandbox registration. The top-up flow is based on a Payment Intent architecture. First, the backend endpoint generates a PaymentIntent with validated amount and currency constraints. Using the given secret key and Stripe's client SDK, the frontend then completes the payment. If successful, the backend confirmation endpoint will retrieve the PaymentIntent from Stripe, confirming ownership and final status. The backend only credits the internal wallet register through these verifications, ensuring secure payment process.

```
    return res.json({
      clientSecret: intent.client_secret,
      paymentIntentId: intent.id,
      publishableKey: process.env.STRIPE_PUBLISHABLE_KEY || null,
    });
  } catch (e) {
    return res.status(500).json({ message: 'Failed to create Stripe PaymentIntent', error: e.message });
  }
});
```

Figure 5.2.2 Stripe Configuration

Mailgun

In this system, Mailgun was integratedds through SMTP transport using Nodemailer, which provides a simple email delivery while keeping mail logic lightweight at the application layer. The mail client is initialized with smtp.mailgun.org, a customizable SMTP port, and credentials loaded from .env file.

```
const nodemailer = require("nodemailer");

const transporter = nodemailer.createTransport({
  host: "smtp.mailgun.org",
  port: Number(process.env.SMTP_PORT || 587),
  secure: false,
  auth: { user: process.env.SMTP_USER, pass: process.env.SMTP_PASS },
});
```

Figure 5.2.3 Mailgun Configuration

Google Maps

This application integrated Google Maps for map rendering and geocoding for stop-name fallback resolution. To use the map feature, install dependencies in pubspec.yaml file, add Google Maps Android SDK key as app metadata, and declare location permission in AndroidManifest.xml as in figures below.

```
geolocator: ^14.0.2
socket_io_client: ^2.0.3
google_maps_flutter: ^2.12.3
google_polyline_algorithm: ^3.1.0
```

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyD1UmCU2C_LFw_qpQjl7o_[redacted]" />
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Figure 5.2.4 Google Maps Configuration

5.3 System Operation

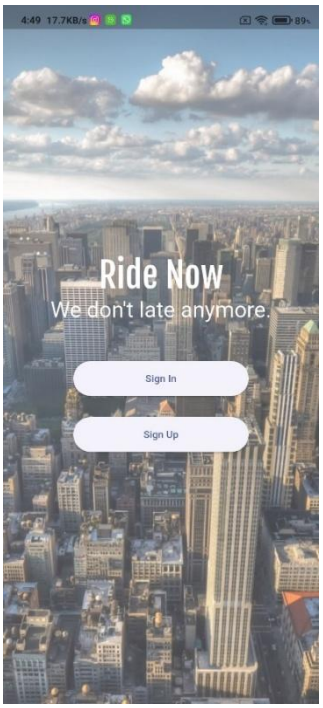


Figure 5.3.1 Welcome page

Authentication module

Users can register and log into to their account in this module. Since this application was designed specifically for UTAR community, hence it only allows for registration using UTAR email, and drivers have to get the invite code to sign up their account. To secure the management of bus service, admin account was created from backend, and no one can create for a new one.

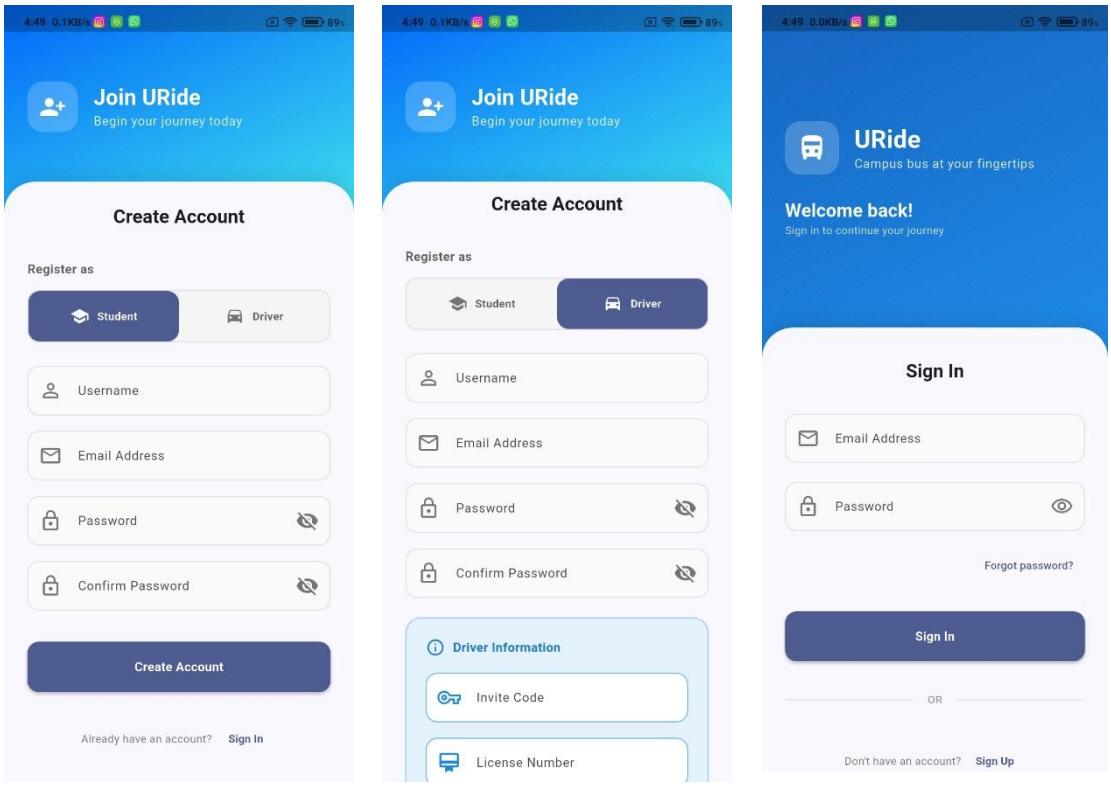


Figure 5.3.2 Authentication module

Driver verification module

To ensure a safe campus transit experience, all drivers' account applications need to be verified by admin. Once the application has been approved, the driver will be notified via email of the activation status.

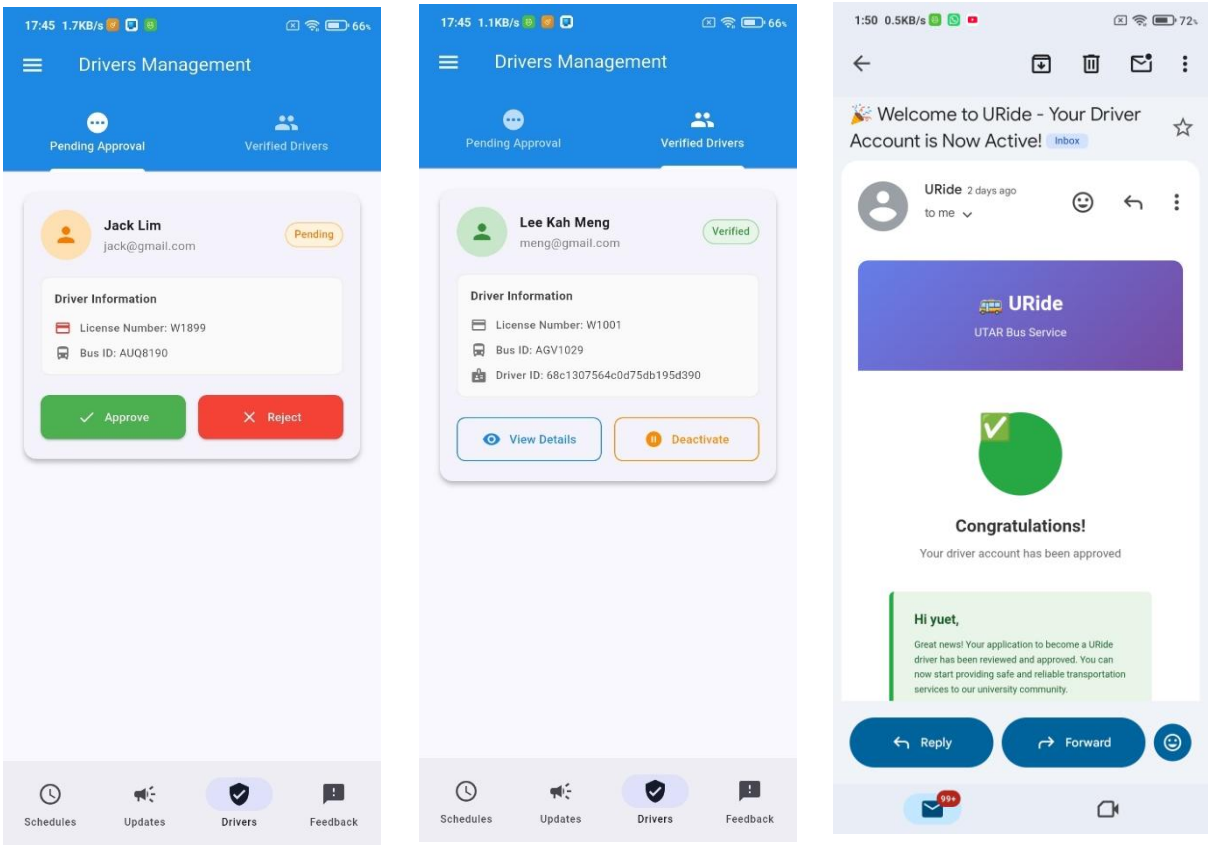


Figure 5.3.3 Driver verification module

Schedule module

In this module, admin first has to create a schedule and assign driver to it. Drivers can only check their own task, while students can view all available schedules and the details from their account. Then, drivers can start the trip in application with GPS permission enabled in order to share the real-time bus location with students.

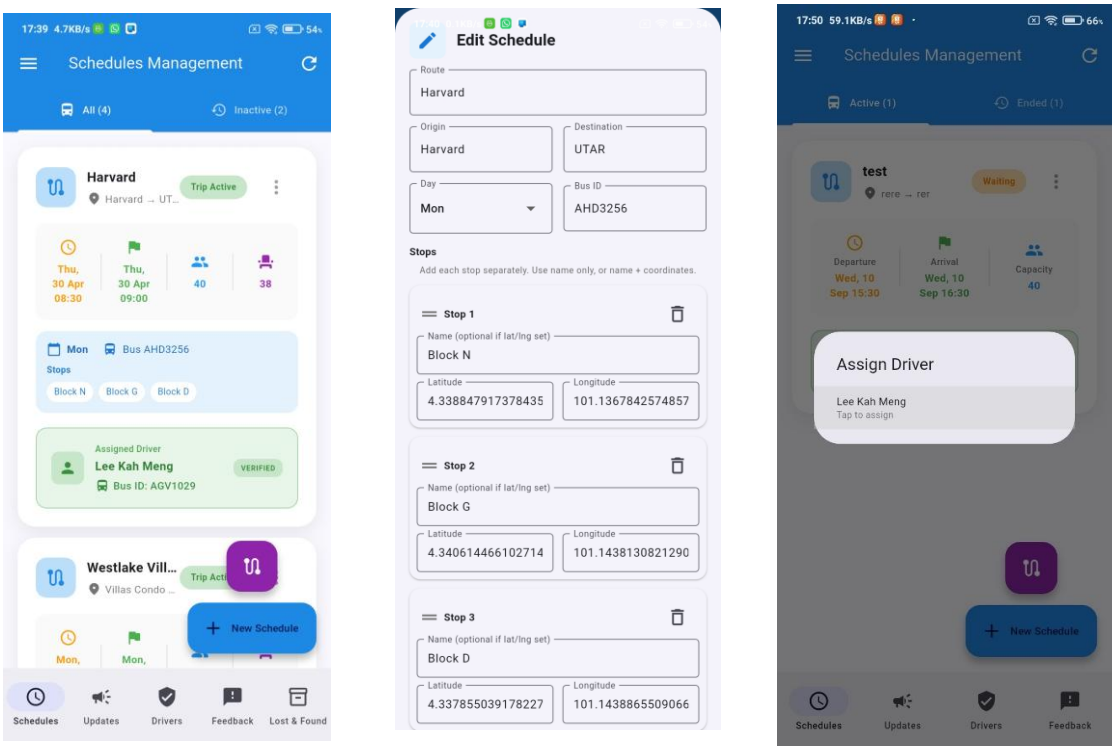


Figure 5.3.4 Schedule module (Admin)

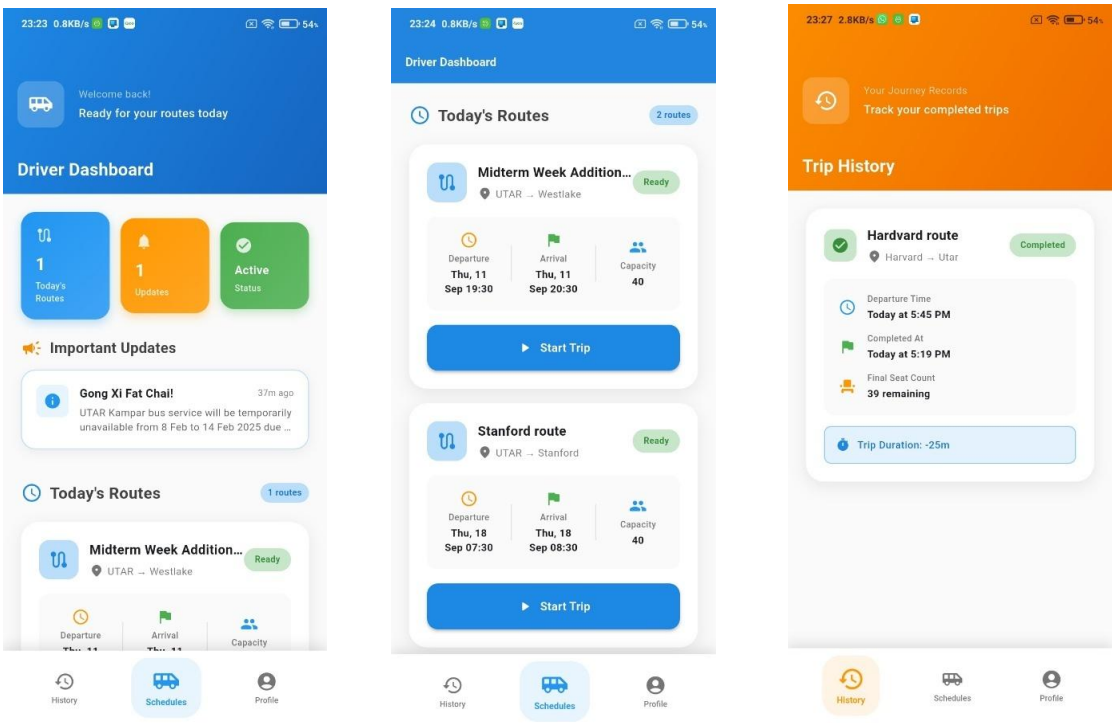


Figure 5.3.5 Schedule module (Driver)

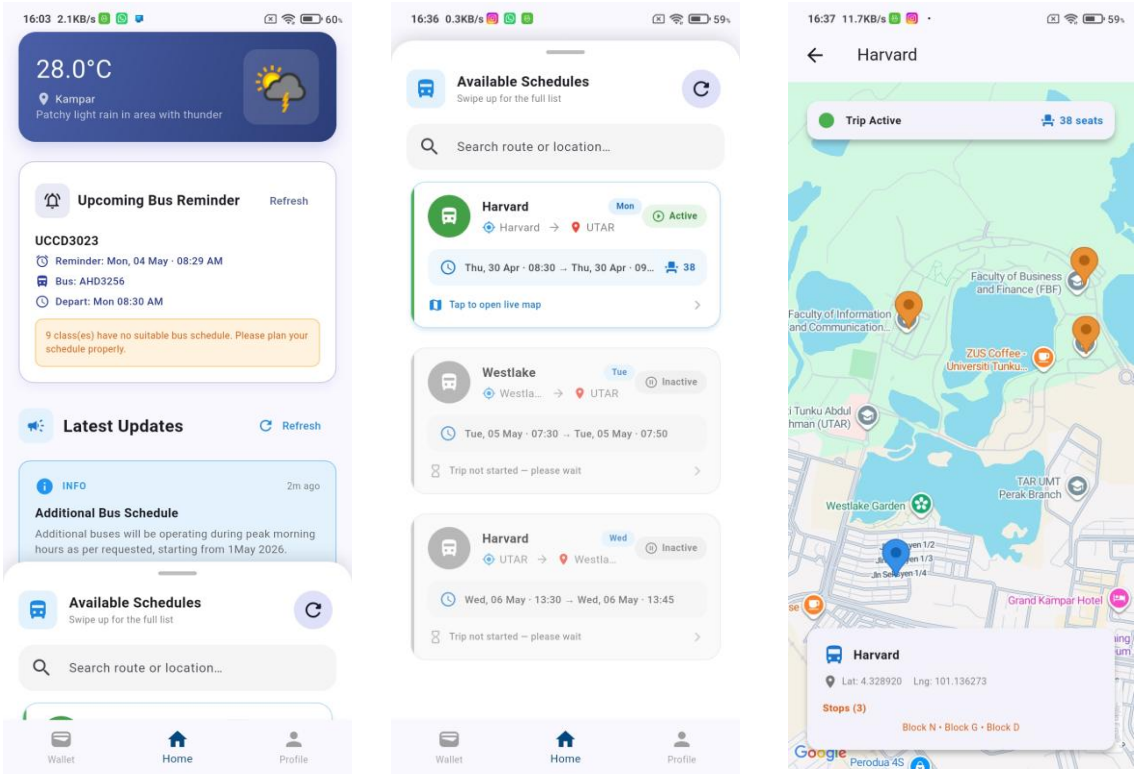


Figure 5.3.6 Schedule module (Student)

NFC payment module

Once driver has started a trip, they may start to scan the NFC pass of students. Only valid and active cards will be charged in this NFC fare payment. The seats count will also be updated upon each successful payment.

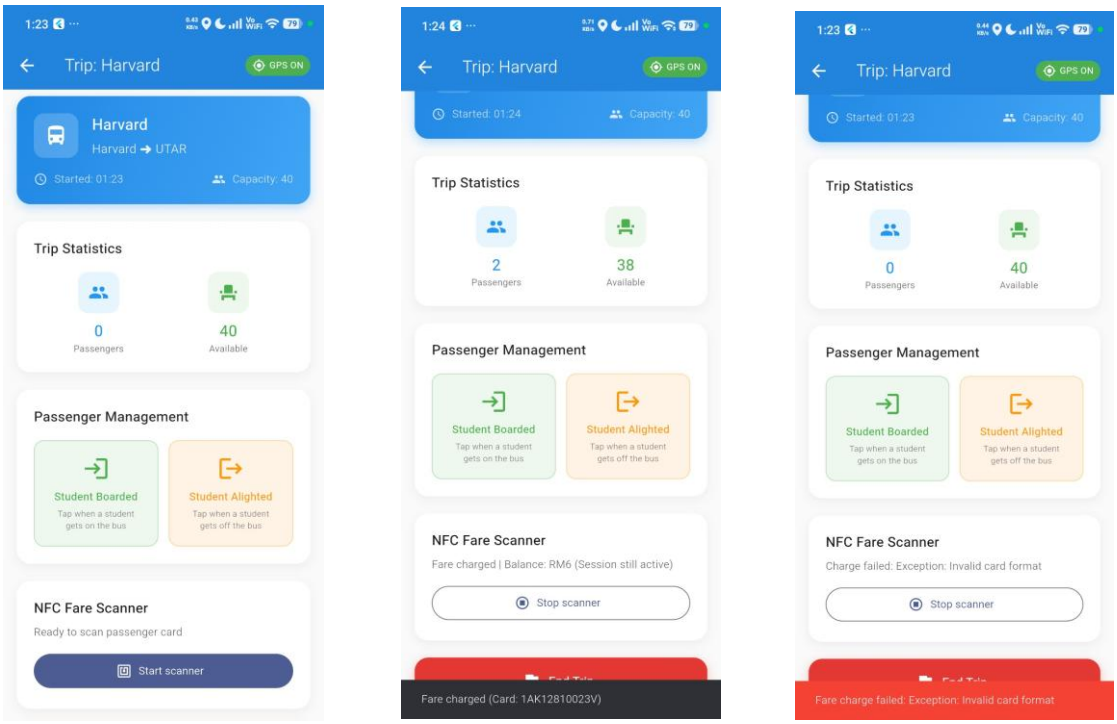


Figure 5.3.7 NFC Payment module

Wallet module

In this module, students can check their wallet balance and reload through the Stripe payment gateway. Students can link their NFC pass and report lost to block the pass as well. Besides, a transfer feature was built in case that users forget to bring their NFC pass when riding the bus, then they may transfer money directly to other students' wallet to help them out. All transaction activities can be found in the history section.

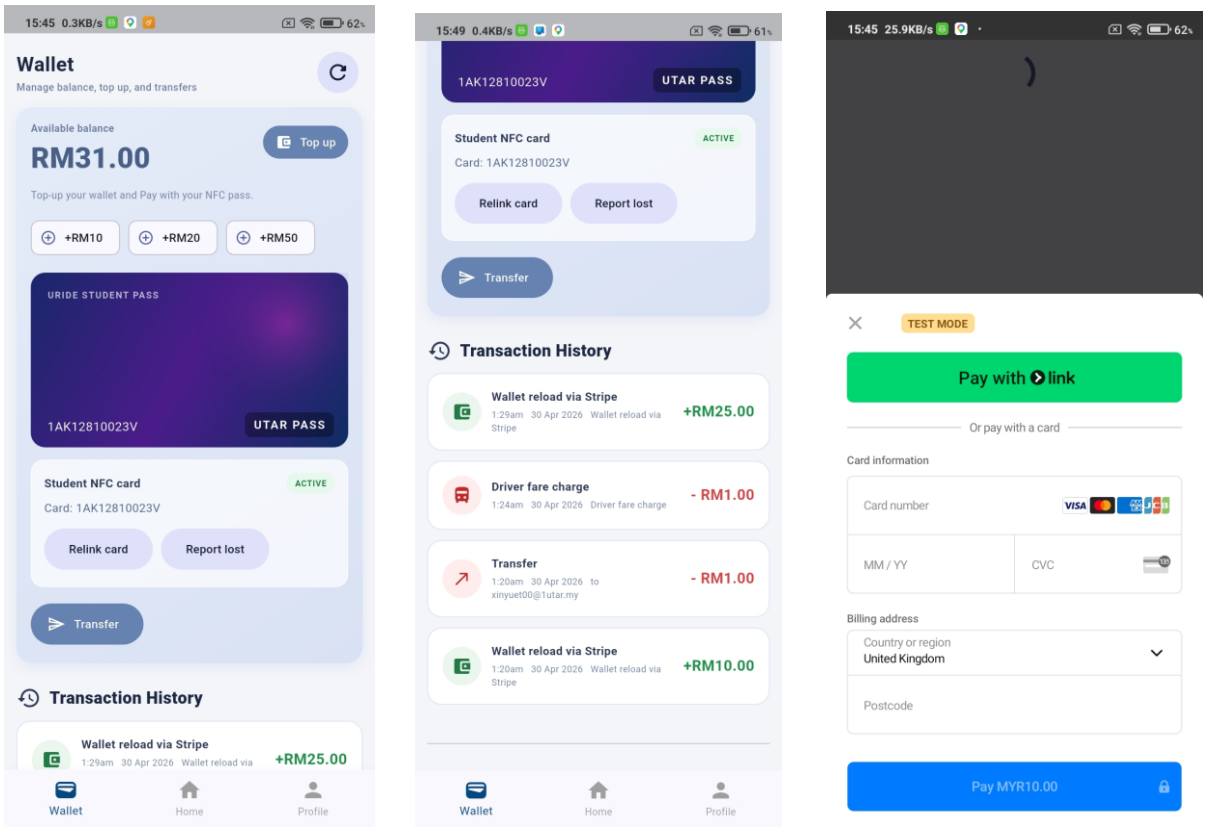


Figure 5.3.8 Wallet module

OCR timetable module

For this module, students can upload or capture their class timetable image and let the app pull out the class rows via OCR technology. They can edit or remove rows if needed and save the final timetable. Once saved, the system keeps their timetable up to date with available schedule so that they can receive upcoming bus reminders automatically.

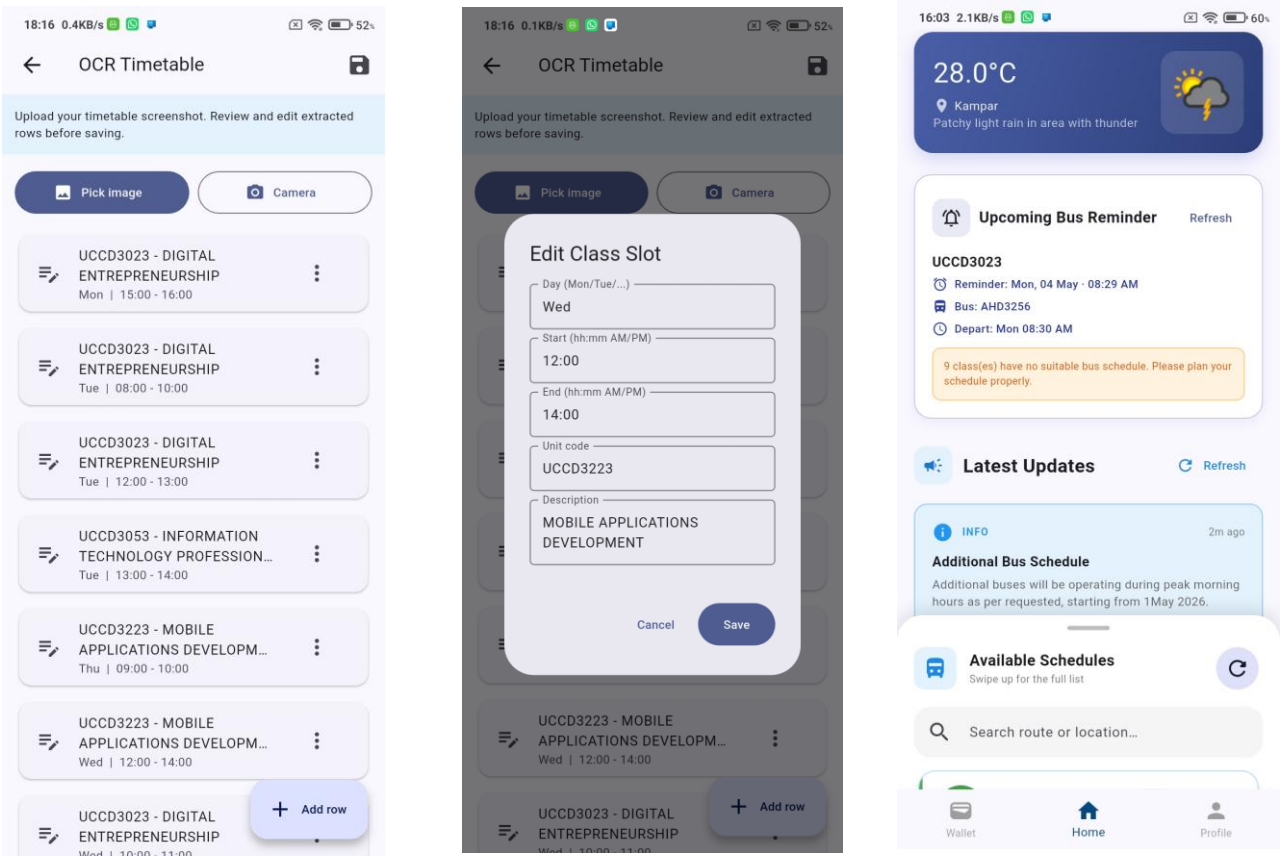


Figure 5.3.9 OCR Timetable module

Feedback module

For this module, it is available for all user roles, in which students and drivers may share their feedback or request on the service to management. Meanwhile, admin may resolve the ticket with additional reply.

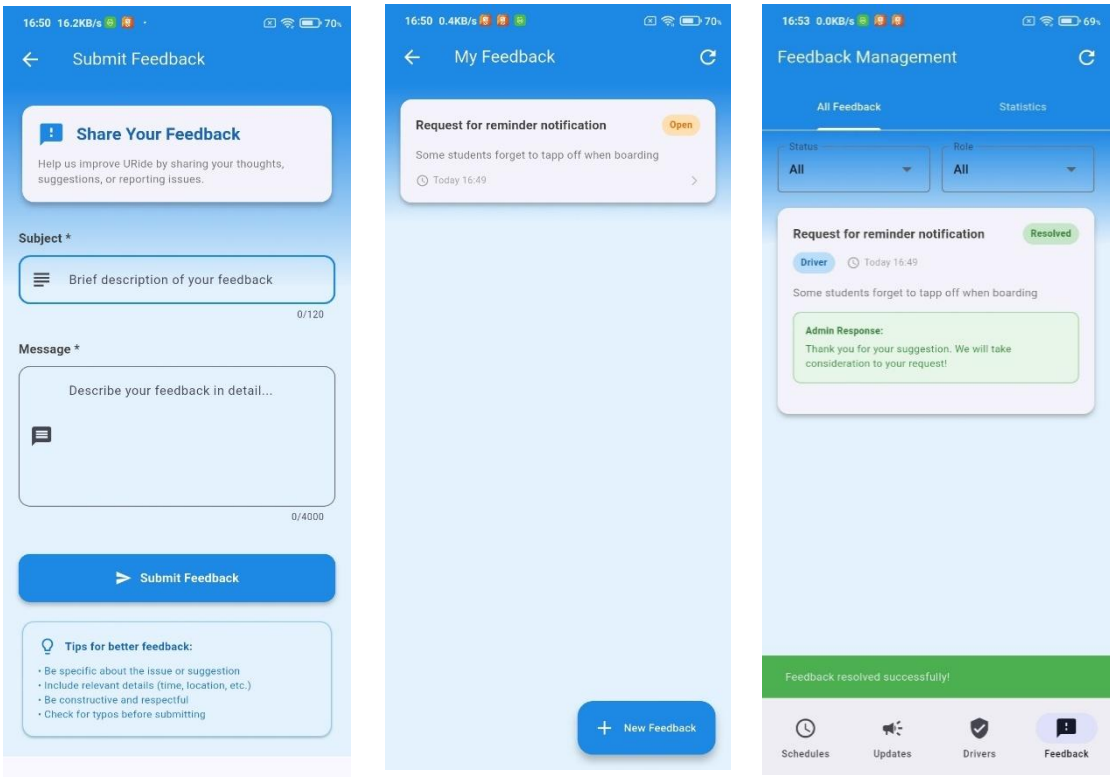


Figure 5.3.10 Feedback module (Student & Driver)

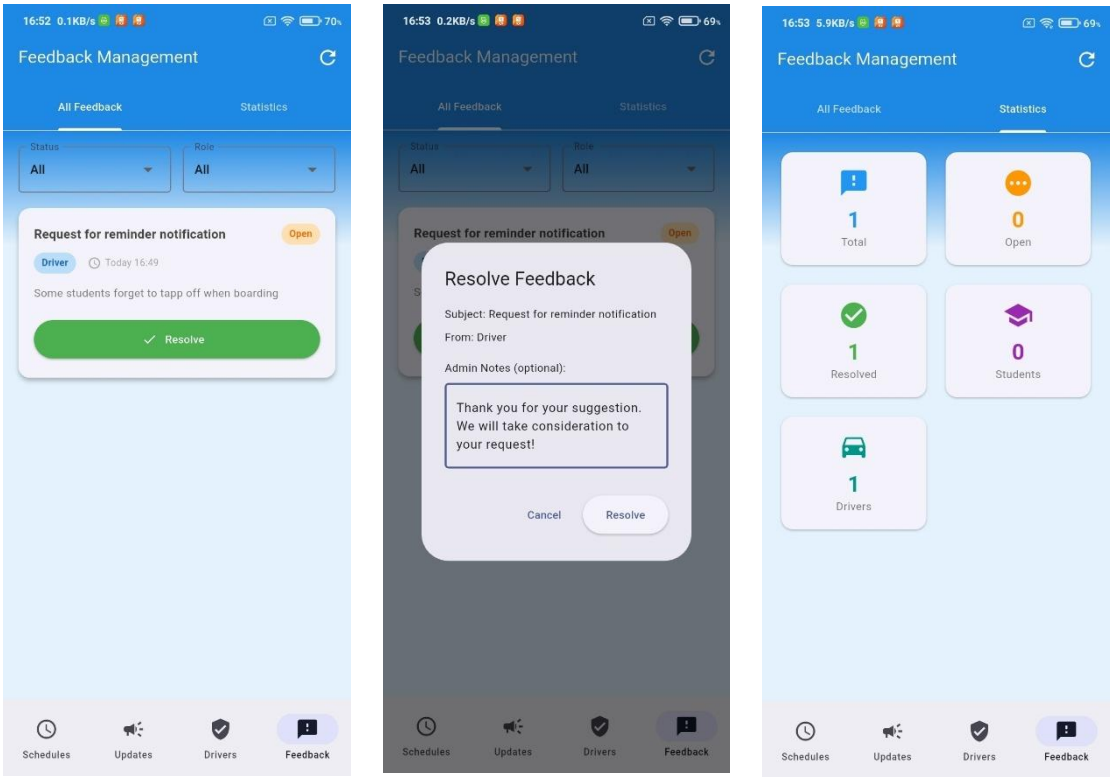


Figure 5.3.11 Feedback module (Admin)

Update module

Similarly, this module is available across all user roles, where admin can create different types of updates and decide the access level.

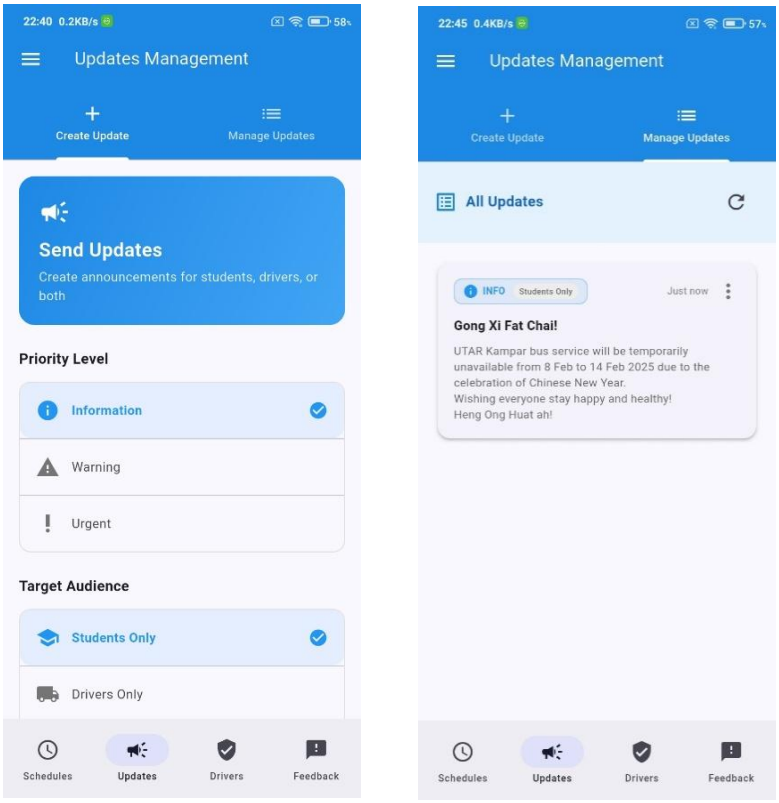


Figure 5.3.12 Update module (Admin)

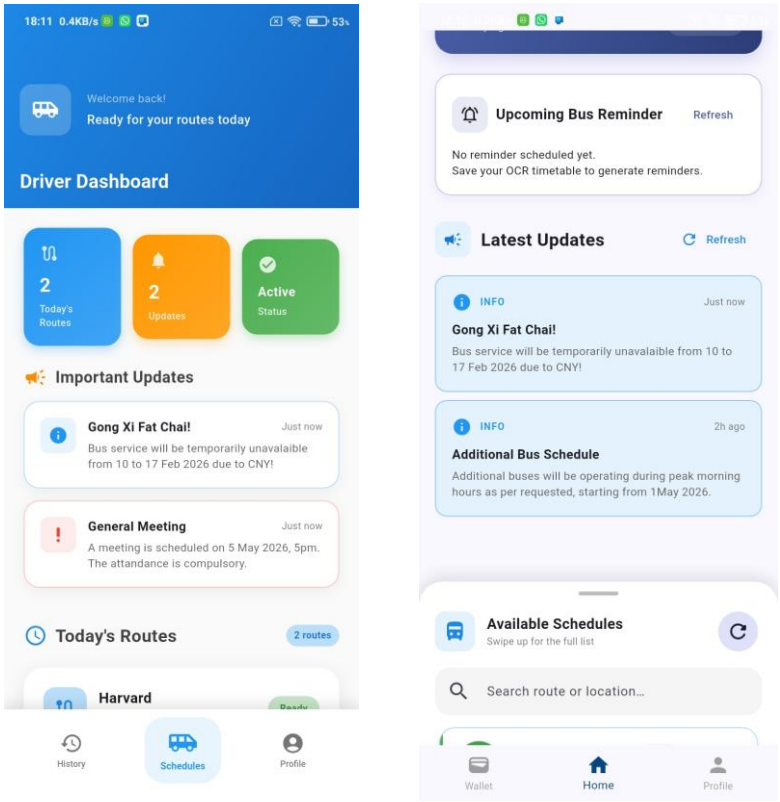


Figure 5.3.13 Update module (Driver & Student)

Lost and Found module

With this module, admin can create posts for lost items found on campus transport, while students can browse the list and submit a claim if they believe an item is theirs. After admin's approval, students can confirm the collection when they have collected it. Admin can also track for the collection record like student's email for all items.

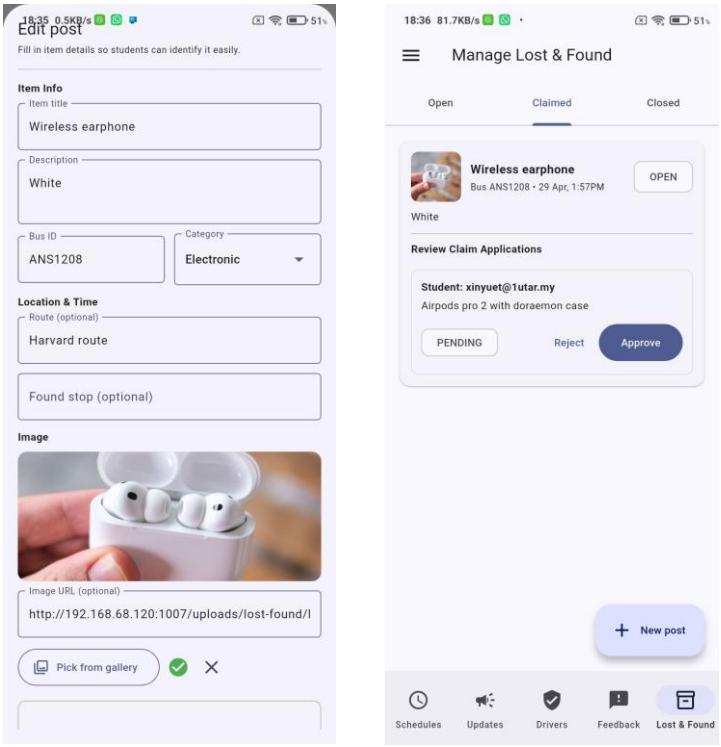


Figure 5.3.14Lost and Found module (Admin)

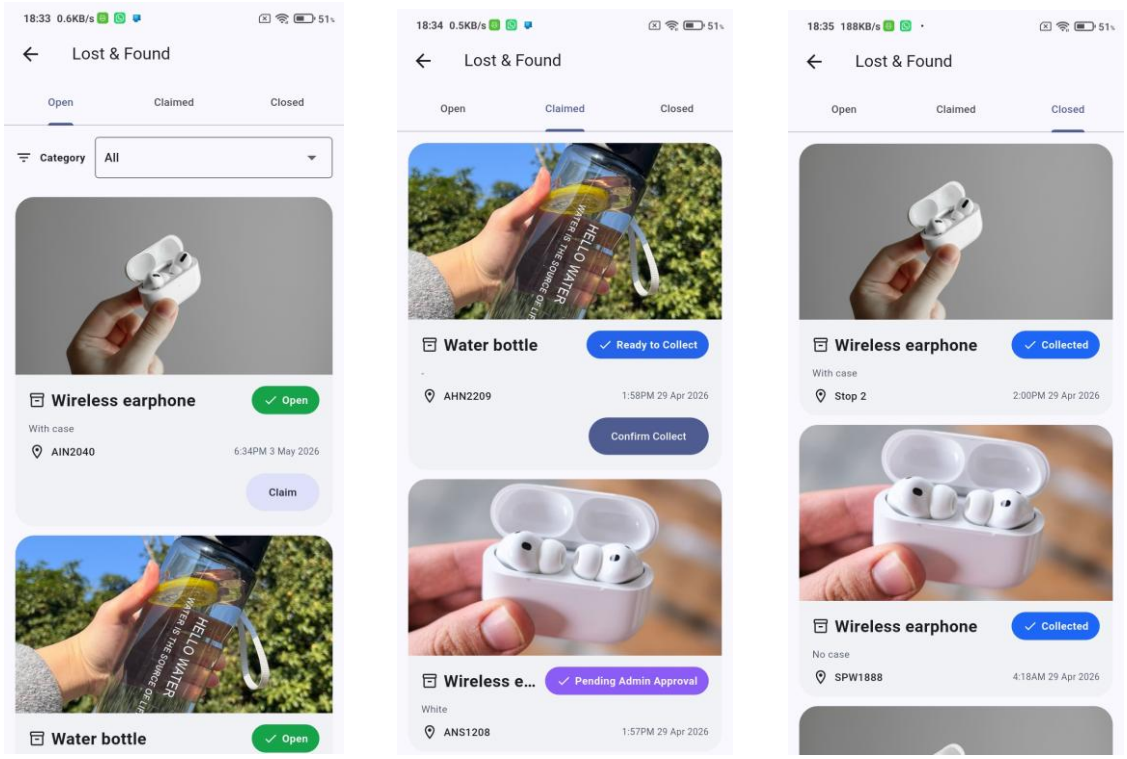


Figure 5.3.15 Lost and Found module ((Student)

5.4 Implementation Issues and Challenges

During the development of this project, the major challenge is the accuracy of OCR timetable extraction. OCR works well in controlled samples, however the quality, format, and structure of photos provided by students vary significantly. Due to variations in screenshot resolution and table layout, output that has missing fields, combined rows, wrong time ranges, and misread unit codes. This was not only a technical difficulty, but also a functional one, because timetable inaccuracies have a direct impact on downstream functionality like reminder generating and scheduling. To overcome this issue, a user review stage was developed to allow students to make corrections before saving.

Another challenge is to maintain the consistency and real-time information of bus updates. The information may be inconsistent due to unstable network connection, and consequently delivering outdated updates which can decrease user trust. For instance, map positions may update irregularly, or seat counts may lag behind actual boarding events. To solve this, the approach focused on robust update processing rather than assuming perfect connectivity. The update flows were designed with comprehensive fallback handling and periodic refresh logic.

Chapter 6 System Evaluation and Discussion

6.1 System testing

With the development and testing of individual modules, an extensive system testing was performed at the end of this project. The black box testing method was used to evaluate the functionality and performance, as well as identify any bugs in the system. In this test, total of 8 main functions was conducted, including authentication, schedule, NFC payment, wallet transaction, OCR timetable, feedback, lost and found posts and service updates. Every feature was evaluated using thorough scenario simulations, and all of them passed. This verifies that the application reaches intended functional requirements.

Authentication

Test Case	Input	Expected Output	Actual Output	Result
TC-A1	Student sign up without UTAR email	Prompt user to use UTAR email	Fields highlighted with validation warning	Pass
TC-A2	Sign up with empty fields	Prompt user to fill in required fields	Fields highlighted with validation warning	Pass
TC-A3	Sign up with valid email and password	Sign up successful, redirect to home screen	Sign up successful, redirected to home screen	Pass
TC-A4	Driver sign up with invalid invite code	Error message: "Invalid driver code"	Error message displayed correctly	Pass
TC-A5	Driver sign up with valid invite code	Display pending approval message	Message displayed correctly, user is unable to log in	Pass
TC-A6	Log in with valid email & password	Login successful, redirect to repective role's home screen	Login successful, redirected to specific role's home screen	Pass
TC-A7	Log in with invalid email or password	Error message: "Invalid credentials"	Error message displayed correctly	Pass
TC-A8	Log in with empty fields	Prompt user to fill in required fields	Fields highlighted with validation warning	Pass
TC-A9	Driver log in without approval	Error message: "Invalid credentials"	Error message displayed correctly	Pass

Schedule

Test Case	Input	Expected Output	Actual Output	Result
TC-S1	Admin creates schedule & assigns it to available driver	Schedule saved successfully, driver is notified	Schedule created and visible by driver	Pass
TC-S2	Admin assigns overlapping schedules to driver	Error: "Schedule conflict"	Error displayed, assignment prevented	Pass
TC-S3	Student views schedules	All available schedules visible	All schedules displayed correctly	Pass
TC-S4	Driver views schedules	Only assigned schedules visible	Correct driver-only view	Pass
TC-S5	Driver starts a trip	Live location start, student able to view live map	Students are able to view live map	Pass
TC-S6	Driver ends a trip	Live location sharing stop, trip ended and updated to history	Trip is not visible by students and updated to trip history in drivers and admin views	Pass

NFC Payment

Test Case	Input	Expected Output	Actual Output	Result
TC-N1	Driver starts to scan NFC card with NFC supported device	NFC scan starts and status prompts user to tap card	NFC scan initialized, status shown correctly	Pass
TC-N2	Driver starts to scan NFC card without NFC supported device	System shows NFC unavailable fallback message	"NFC unavailable" message displayed	Pass
TC-N3	Student taps valid NFC card with sufficient wallet balance	Fare deducted, payment success message shown	Payment successful message displayed	Pass
TC-N4	Student taps NFC card with insufficient balance	Payment rejected, failure message shown, no deduction	Payment failed message shown, balance unchanged	Pass
TC-N5	Network timeout during payment request	Payment fails gracefully with exception message	Timeout error displayed, no deduction applied	Pass

Wallet

Test Case	Input	Expected Output	Actual Output	Result
TC-W2	Student top-ups wallet with valid amount via Stripe	Payment sheet completes, backend confirms payment, balance increases, new transaction appears	Top-up success message shown, balance updated, history updated	Pass
TC-W3	Student cancels Stripe payment sheet during top-up	No balance change, cancellation message shown	Top-up canceled message displayed, balance unchanged	Pass
TC-W4	Student transfers valid amount to existing user email	Transfer succeeds, sender balance decreases, transfer recorded in history	Transfer success message shown, balance and history updated	Pass
TC-W5	Student transfers amount greater than available balance	Prohibit transaction and prompt insufficient balance error	Form shows "Insufficient balance", transfer not sent	Pass
TC-W6	Student links NFC card with invalid card id	Link is rejected with validation message	Error message shown: card ID format invalid	Pass
TC-W7	Student links NFC card with valid card id	Link card to student's account, card id shown at wallet	Link card successfully with updated wallet ui	Pass

OCR Timetable

Test Case	Input	Expected Output	Actual Output	Result
TC-O1	Student uploads clear timetable screenshot	OCR extracts class rows (day, time, unit code, description)	Timetable rows extracted and displayed	Pass
TC-O2	Student uploads unclear image	OCR returns no rows, warning shown	"No timetable rows detected" message displayed	Pass
TC-O3	Student edits extracted row and saves timetable	Updated row saved and reflected in timetable list	Edited row saved successfully	Pass
TC-O4	Student deletes one extracted row then saves	Row removed and timetable updated	Row deleted and changes saved successfully	Pass
TC-O5	Create bus schedule that matches timetable	Reminder shown with corresponding class and schedule details	Reminder shown correctly	Pass
TC-O6	Create bus schedule that does not match timetable	Prompt no matching schedule warning	"No suitable bus schedule" message displayed	Pass

Feedback

Test Case	Input	Expected Output	Actual Output	Result
TC-F1	Student submits feedback	Feedback is visible by admin	Feedback displayed correctly at admin dashboard	Pass
TC-F2	Admin resolves feedback without response	Close feedback directly	Feedback marked as "Resolved"	Pass
TC-F3	Admin resolves feedback with response	Close feedback and update with admin's response	Feedback marked as "Resolved" and admin's response is shown clearly	Pass

Lost and Found

Test Case	Input	Expected Output	Actual Output	Result
TC-L1	Admin creates new lost item post with complete details	Post created with status Open and visible to students	New post appears in open tab	Pass
TC-L2	Student browse Lost & Found and submits claim with message	Claim stored as pending status	Claim submitted successfully and status updated	Pass
TC-L3	Student submits empty claim message	Validation blocks submission	"Please enter your claim details" message shown	Pass
TC-L4	Admin approves pending claim	Post status changes to "Ready to Collect"	Admin approval reflected on student side, status updated	Pass
TC-L5	Admin rejects pending claim	Claim rejected and post remains Open	Post reamined as Open	Pass
TC-L6	Student confirm collection on approved item	Post status changes to Closed and Collected status	Collection confirmed and moved to closed tab	Pass

Updates

Test Case	Input	Expected Output	Actual Output	Result
TC-U1	Admin creates updates for student	Updates received by students only	Updates displayed correctly	Pass
TC-U2	Admin creates updates for driver	Updates received by driver only	Updates displayed correctly	Pass
TC-U3	Admin creates updates for students & drivers	Updates received by students & drivers	Updates displayed correctly	Pass

Figure 6.1.1 System Testing Result

6.2 User Evaluation

To further investigate the performance and user experience of this bus transit application, a survey was conducted among UTAR students, as they are the primary users of the system. There is total 11 participation in this evaluation by testing the system functionalities, such as navigation, bus tracking, wallet operations, and so on.

The first section was user background, aiming to understand the familiarity of bus service and transportation application among the users. The result indicates that although none of the respondents rely heavily on the bus service, most of them have taken the bus before and are familiar with similar apps.

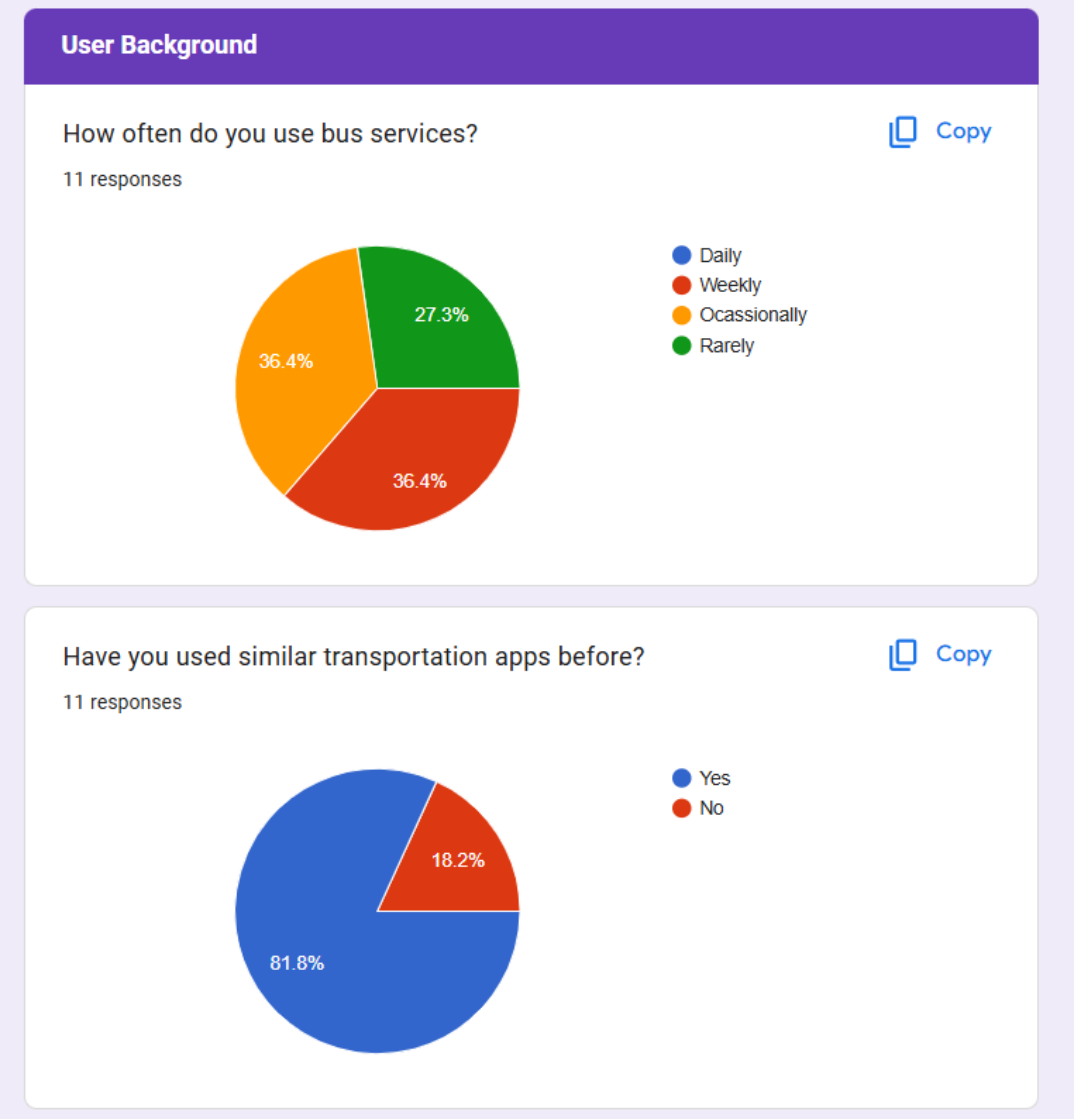


Figure 6.2.1 User Background Result

The next section is the usability evaluation, which aims to evaluate the system's user-friendliness, response time, and interaction flow. From the results, all respondents rated positively, showing that the UI is clear and understandable. While for the system responsiveness and ease of use are generally good as well, with a few respondents rated average. These pattern highlights an opportunity to further optimize the application.

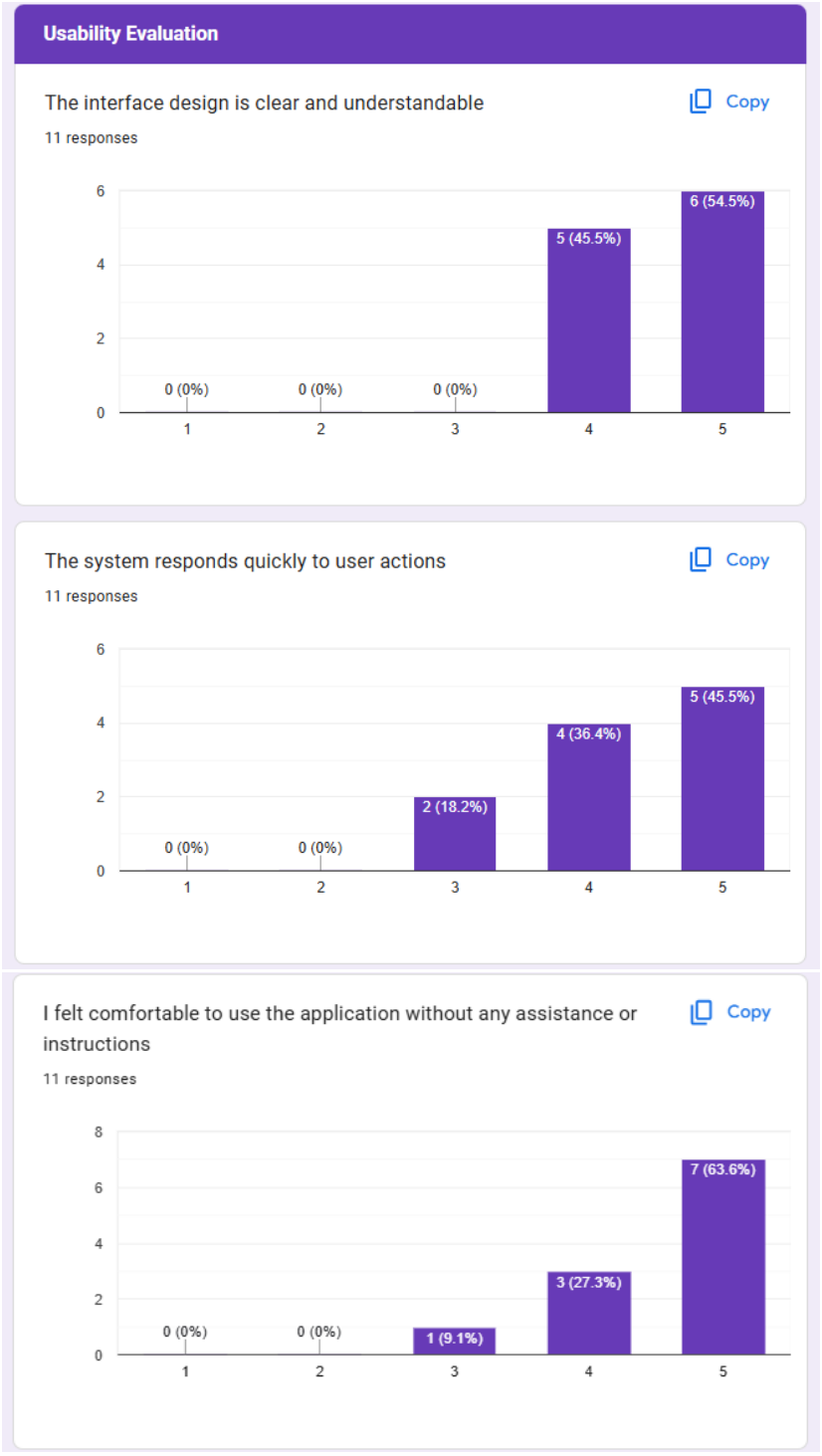


Figure 6.2.2 Usability Evaluation Result

This section evaluates the usefulness and efficacy of the core functions of the application. The assessment focuses on how well each feature facilitates user tasks and the general level of satisfaction with its operation. Based on the results, all features were rated highly. This demonstrates that the application is able to deliver value and reliability, contributing to strong user experience without critical issues.

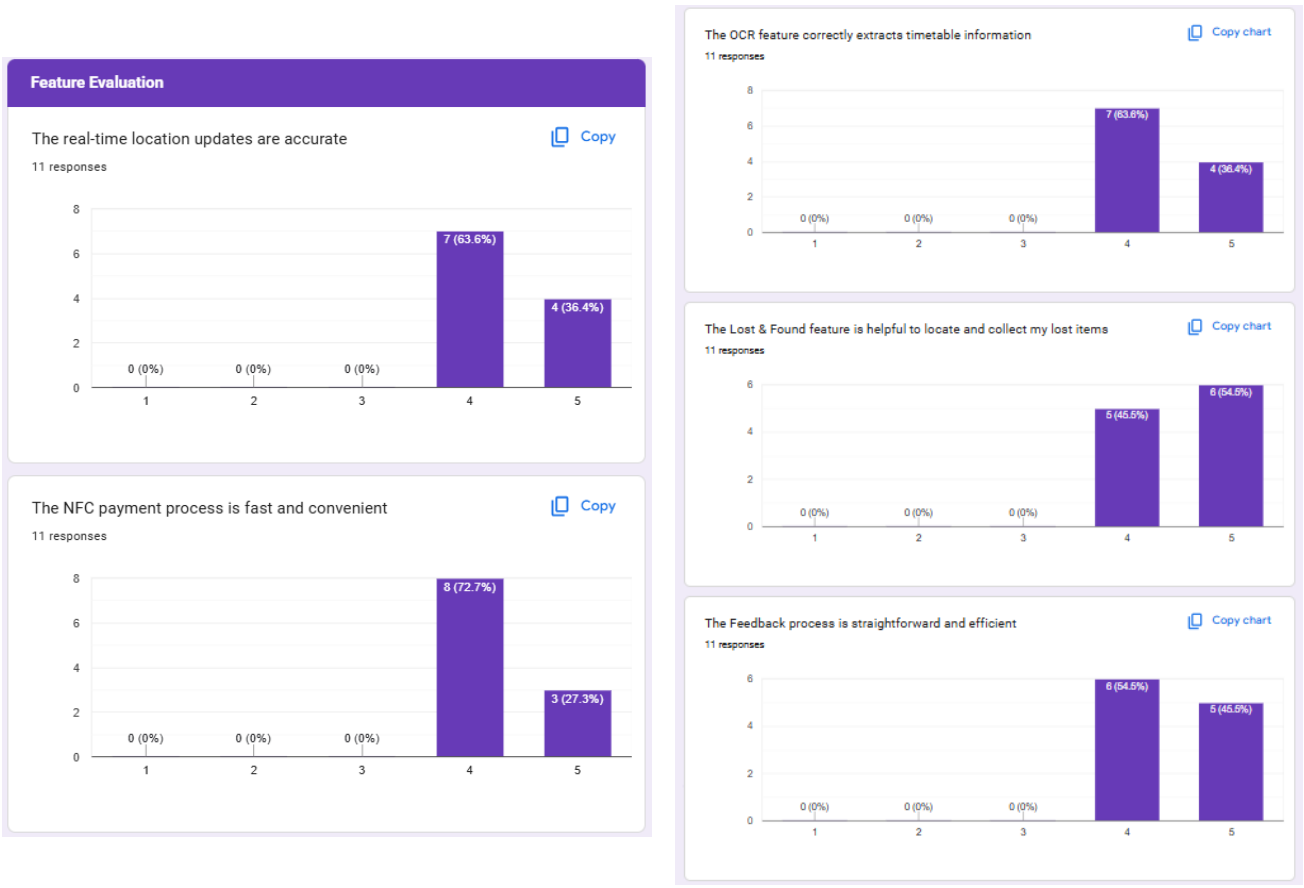


Figure 6.2.3 Features Evaluation Result

The last section aims to examine the overall satisfaction of the application, including respondents' willingness to recommend it to others and suggestions for improvement. The goal is to understand the broader value the app delivers and identify opportunities to enhance user satisfaction. Overall, all respondents were satisfied and would recommend the application with their friends. The feedback result also identifies some significant areas for improvement, including technical accuracy and notification functionality.

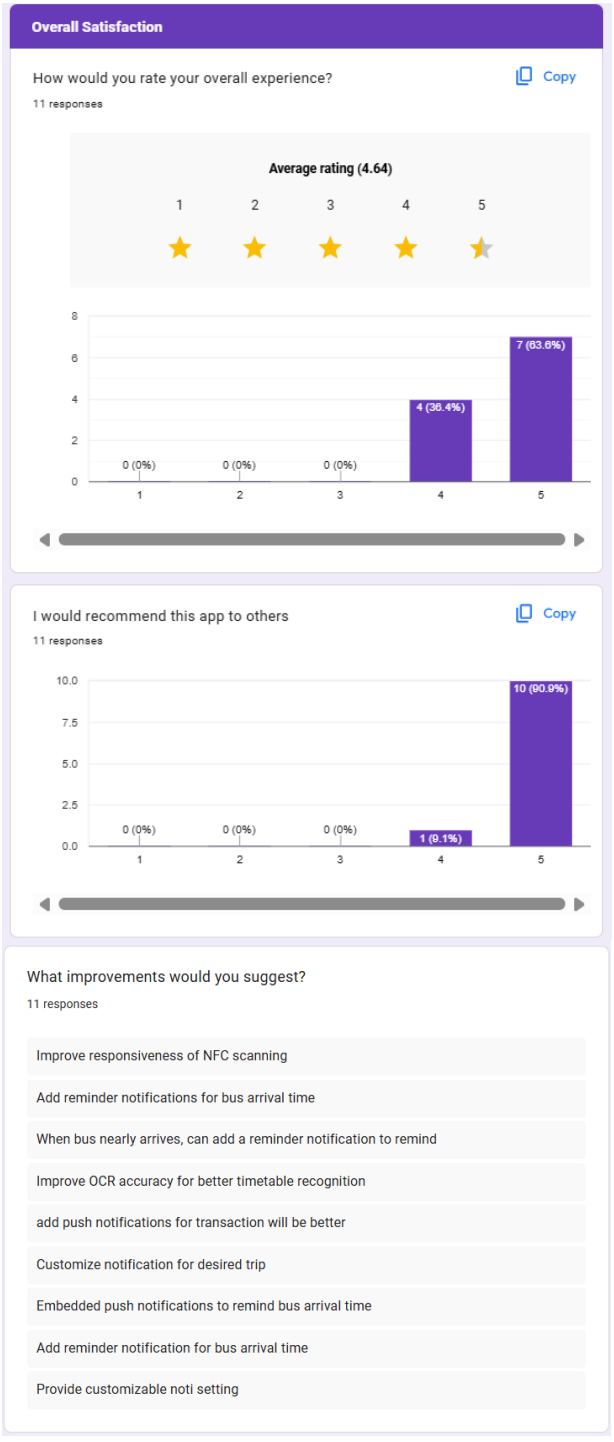


Figure 6.2.4 Overall Satisfaction Result

6.4 Objectives evaluation

The objectives of this project were develop an efficient fare payment system to decease boarding delay, integrate real-time bus tracking feature to enhance system reliability, as well as building a smart scheduling system to reduce user's effort. All in all, this project was aimed at improving the campus bus experience by offering students with a centralized mobile platform for daily transportation and fast transaction process, and also improve administrative efficacy for campus transportation service.

With the development of this application, all goals were successfully accomplished based on the implemented scopes, including wallet-based financial features, modern fare payment through NFC, smart scheduling via OCR technology and so on. Drivers can constantly update the schedule information and location, providing instant and transparent works. Meanwhile, management is able to store service information and respond to user-submitted issues using structured modules rather than informal conversation. It also supports basic management needs, such as scheduling oversight and student support channels.

Chapter 7

Conclusion and Recommendation

7.1 Conclusion

This project concludes with the successful development of this mobile campus bus application that integrates transportation operations, student convenience features, and administrative management into a single system. The project emphasizes the significance of achieving a balance between usability, dependability, and implementation scope in addition to feature delivery. This has showed that the proposed idea is practical and effective for real campus use with the implemented modules, including authentication, schedule access, live trip tracking, wallet and NFC payment support, OCR timetable processing, lost-and-found, feedback and service updates management. Functional validation also indicates that the main workflows perform properly and the test results match expected behaviour throughout the main user journeys. In brief, this project accomplished its primary goal of enhancing campus bus service accessibility and coordination between all user roles.

7.2 Recommendation

To further increase the user experience of this campus bus system, several enhancements are suggested to be developed. Instead of restricting alerts to specific features, the system should implement a single push notification layer for important activities across modules. For examples, low wallet balance alerts, fare deduction, service announcements and so on. This can help to reduce users' efforts and increase user awareness in real time. Additionally, the system could include a precise route sketching with expected arrival time for active trips. The live map would be much more helpful for planning if it included the expected time to the next stop and destination in addition to the current bus location. Basic analytics dashboards could also be integrated to track trip punctuality, payment throughput, and user engagement trends. These improvements would enhance administrative decision-making and user experience.

REFERENCES

[1] W. C. Hoo, A. Waheeda, and A. Reesha, "Factors Influencing the Passenger Satisfaction at Public Transport in Kuala Lumpur, Malaysia," Researchgate.net, July-2023. [Online]. Available: https://www.researchgate.net/publication/371968809_Factors_Influencing_the_Passenger_Satisfaction_at_Public_Transport_in_Kuala_Lumpur_Malaysia.

[2] S. Tiwari, "An introduction to QR code technology," in 2016 International Conference on Information Technology (ICIT), 2016, pp. 39–44.

[3] C. Upendra Reddy, D. L. Vara Prasad Reddy, N. Srinivasan, and J. Albert Mayan, "Bus Ticket System for Public Transport Using QR Code," Iop.org, 2019. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/590/1/012036/pdf.

[4] K. Ashok, B. M. Sanketh, M. A. Shashank, K. N. Tanush, and R. Varun, "Qr scan ride: Smart bus passenger ticketing system," May-2024. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/590/1/012036.

[5] "NFC Payments Market size, share, growth," Research Dive, Jan-2024. [Online]. Available: https://www.researchdive.com/9173/nfc-payments-market. [Accessed: 06-Sept-2025].

[6] G. Osei, R. A. Sowah, M. R. Ansah, I. A. Aboagye, N. L. Sowah, and D. Xeflide, "Self-managed NFC contactless payment system for commercial shuttles," in 2024 IEEE 9th International Conference on Adaptive Science and Technology (ICAST), 2024, vol. 9, pp. 1–5.

[7] S. Yeruva, A. Grandhi, S. D. Katta, S. S. C. Palutla, and G. D. Amruthaluri, "EduBusGuru: Enhancing college bus tracking and management with real-time mobile app," in 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2024, pp. 1–8.

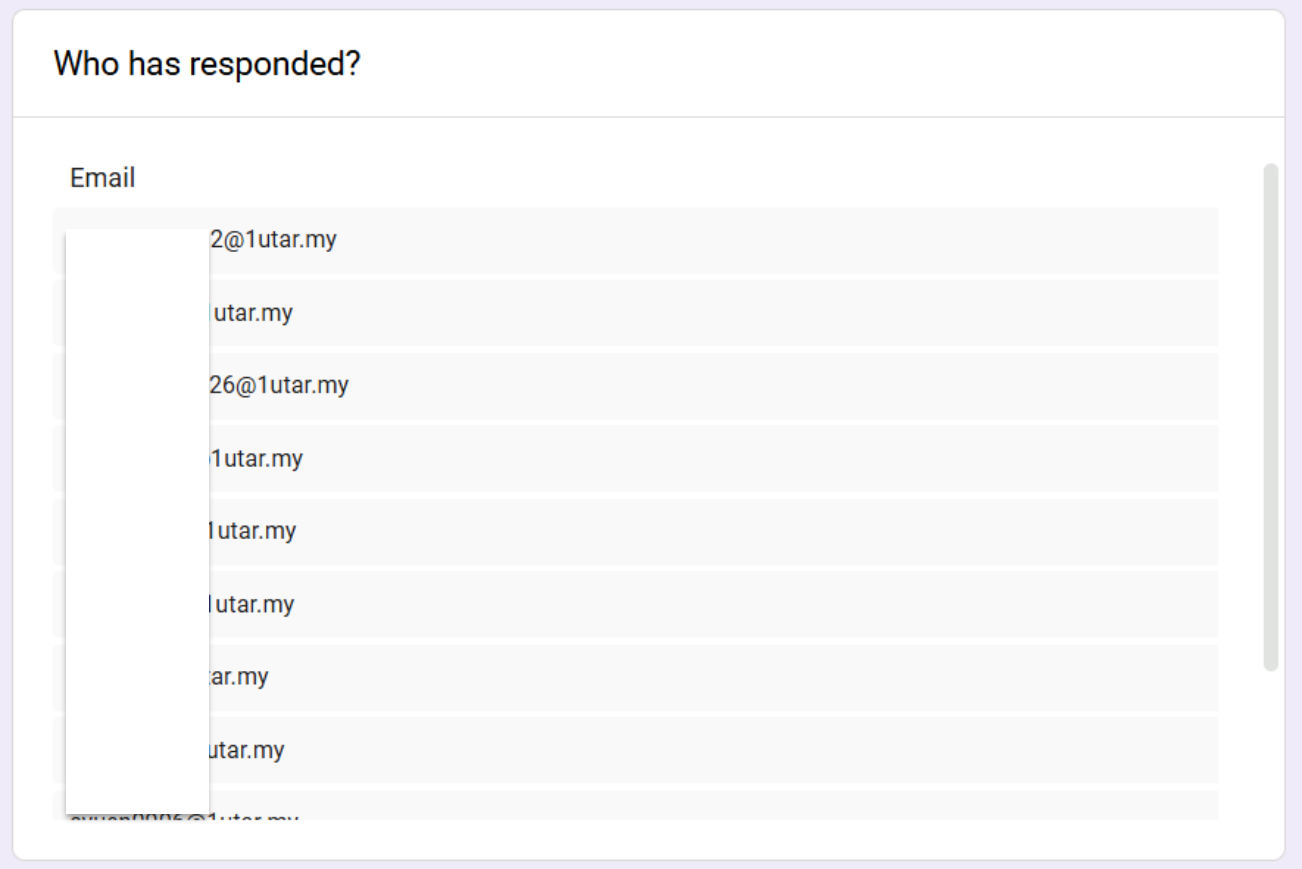
[8] S. A. Sharif, M. S. Suhaimi, N. N. Jamal, I. K. Riadz, I. F. Amran, and D. N. A. Jawawi, "Real-Time Campus University Bus Tracking Mobile Application," Researchgate.net, July-2018. [Online]. Available: https://www.researchgate.net/publication/328834009_Real-Time_Campus_University_Bus_Tracking_Mobile_Application.

[9] B. Apoorva, C. Kavitha, M. Bharath, N. N. Vaidya, and M. HarshithaG, "A review on smart bus ticketing and tracking system using IoT," International journal of engineering research and technology, 2018.

[10] A. Gadekar, A. Kandoi, G. Kaushik, and S. Dholay, "QR scan based Intelligent System for School Bus Tracking," in 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 1074–1080.

[11] G. Chen, W. Chen, S. Zhang, D. Zhang, and H. Liu, "Influence of Mobile Payment on Bus Boarding Service Time," *Researchgate.net*, July-2020. [Online]. Available: https://www.researchgate.net/publication/343134689_Influence_of_Mobile_Payment_on_Bus_Boarding_Service_Time.

Appendix



Appendix 1: Feedback survey among UTAR students

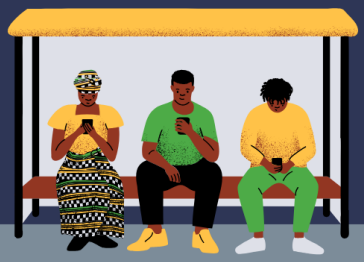
POSTER

UNIVERSITI TUNKU ABDUL RAHMAN
Faculty of Information and Communication Technology

Revolutionizing UTAR Bus Transit System: Development of Mobile Application with Real-Time Tracking and Smart Scheduling

Proposed by: Chong Xin Yuet

Supervised by: Ts Dr Phan Koo Yuen



PROBLEM STATEMENT

- ❌Ineffective payment method
- ❌Dependence on third-party app
- ❌Unreliable tracking
- ❌Lack of customization

OBJECTIVES

✅To build a scalable mobile solution for campus mobility

✅To improve the reliability and transparency of UTAR's bus transit system by incorporating real-time data tracking and immediate resource updates.

✅To develop an efficient fare payment system that allows students to pay bus fares seamlessly, reducing boarding delays of paper tickets and QR paymen

DEVELOPMENT TOOL

- Flutter
- Node.js
- VS Code
- MongoDB Atlas
- Google Maps API
- Google ML Kit
- NFC

NOVELTY

- Integrated platform with real-time tracking, communication & schedule in one app
- NFC-based payment
- Intelligent timtable matching with bus schedule using OCR technology

METHODOLOGY

Project Planning
Define project goals and perform requirement analysis

System Design
Develop app UI & system architecture

AGILE

Testing
Conduct Black-box testing and feedback survey

Project Development
Develop system in small sprints