
**AQUINAS COLLEGE OF HIGHER STUDIES**

**SRI LANKA**

# WEB BASED BUS TICKET RESERVATION SYSTEM FOR SERENDIB COMPANY

By

K. Lahiru Kasun Sandeepana Silva

BIT/2Y19/05

Faculty of Science and Technology

Aquinas College of Higher Studies

May 2023

Dissertation submitted to the Faculty of Science and Technology, Aquinas College of Hiher Studies, Sri Lanka for the partial fulfilment of the requirements of the Degree of Bachelor of Information Technology.

# Declaration

I certify that I am an undergraduate and that the research proposal submitted is an original piece of work prepared under the guidance of my supervisor. I confirm that the submitted proposal contains my own work and to the best of my knowledge and belief contains no material previously published or written by another person, except that which is acknowledged.

K. Lahiru Kasun Sandeepana Silva ……..……

Name of Student Signature of Student

……..……

Date

**Superviced by**

Mr. Chathura Jayamuni ……..……

Name of Supervisor Signature of Supervisor

……..……

Date

**Certified by**

Dr. D. S. U. Peiris ……..……

Dean Signature

……..……

Date

# Dedication

I dedicate this dissertation to my beloved wife, Samurdhi, who has been a constant source of support and inspiration while I was studying for my Bachelor of Information Technology degree. Without her love, patience, and encouragement, the completion of this research would not have been possible. This work is also dedicated to my dear parents, Lesly and Chithra, for their unconditional love and for the lessons they taught me in achieving my dreams. Their fullest and truest attention during these past years has undoubtedly been a great source of motivation throughout my life.

# Acknowledgment

I am delighted to acknowledge the valuable contributions of several individuals who helped me successfully complete my BIT final-year dissertation.

Firstly, I would like to express my deepest gratitude to my project supervisor, Mr. Chathura Jayamuni, for providing me with the necessary resources and facilities to carry out this project. His encouragement and cooperation were greatly appreciated.

I would like to express our heartfelt thanks to Dr. D.S.U. Peiris, who is the Senior Lecturer-Department of ICT and Dean-Faculty of Science and Technology, Aquinas College of Higher Studies. He spent his valuable time supporting and guiding me in completing this final project report and providing us with the necessary resources and support to carry out this project. We are grateful for your guidance and mentorship throughout the project, which helped us overcome challenges and deliver the project on time.

And I would like to thank Mr. Kosala Perera and Mr. Roshan Ranatunga, lecturers from the Faculty of Science and Information Technology at Aquinas College of Higher Studies, for their help throughout the project.

I would like to thank the faculty of science and information technology at Aquinas College of Higher Studies staff, including Mr. Nirosh Wanigasekara, the administrative officer, for helping me complete my degree program on time.

Also, I would like to thank Rev. Fr. Prasanka Rodrigo, Priest-in-Charge Faculty of Science and Information Technology, and Aquinas College of Higher Studies for assisting me to grow as an independent individual in society and motivating me to complete this final year project on time.

Finally, I am grateful to my loving wife, family, and friends for their unwavering support and understanding during the course of this project. Finally, I acknowledge the contributions of all participants, who generously gave their time and expertise to participate in the study. Their willingness to share their experiences and insights was crucial to the success of this project.

# Abstract

The web-based bus ticket reservation system is an online platform designed to simplify the process of reserving and purchasing bus tickets through the internet. The system is accessible through a user-friendly website that allows passengers to search for available bus routes, select their preferred travel dates and times, and reserve their seats with ease. Traditionally, bus ticket purchase has been done over the counter in bus terminals; however, today it has evolved with the rapid expansion of e-commerce.

This system lets the member make a reservation for the bus ticket via the Internet, and the administrators can do some back-end work such as add, edit, delete, and view the information. After they make a reservation, customers can make payment online so that they do not waste time going to the counter to make payment.

This project provides a review of the software program "Web Based Bus Ticket Reservation System" as it is to be used in a bus transportation system, including seat reservation, cancellation of reservations, and various types of route inquiries used in securing transportation.

Previously, they used the manual (traditional) system as well as an internal booking system. However, the manual system has many issues, as does the internal booking system. They thought they should develop a new bus reservation system with more features. Structured Systems Analysis and Design Methodology (SSADM) was adopted. Furthermore, it uses the free and open-source CSS framework for its front-end UI elements and also includes Bootstrap vanilla CSS, JQuery with JS and PHP. The back end is built using MySQL.

*Keywords:* web-based bus ticket reservation system, online bus ticket booking system, internet, e-commerce, structured systems analysis and design methodology, open source software, user interface, bootstrap, cascading style sheet, java script, hypertext preprocessor

# Table of content

# List of Figures

# List of Tables

# List of Abbreviations

API - Application Programming Interface

CSS – Cascading Style Sheets

CTB – Ceylon Transport Board

DBMS – Database Management Systems

ER Diagram – Entity Relationship Diagram

GIF – Graphics Interchange Format

GUI - Graphical user interface

HGHL – Hands Global Holdings (Private) Limited

HTML – Hypertext Markup Language

ID – Identification

IOC- Image of the Passenger

IT – Information Technology

JS - JavaScript

NTC – National Transport Commission

OS – Operating System

PHP - Hypertext Preprocessor

SLTB – Sri Lanka Transport Board

SMS - Short Message/Messaging Service

SQL – Structured Query Language

SSADM - Structured Systems Analysis and Design Methodology

SSL – Secure Sockets Layer

UI – User Interface

UML- Unified Modeling Languages

WBBTRS – Web Based Bus Ticket Reservation System

XML – Extensible Markup Language

# Chapter 1 - Introduction

## 1.1 Introduction

“Serendib” is a travel company. They provide travel solutions and visa consultations in Sri Lanka. At present, they possess an internal bus ticket booking system as a manual method for bookings within their organization. Manual system has lot of drawbacks such as: time-consuming, limited accessibility, limited payment options, limited information, limited passenger support, and passenger privacy issues. In addition, they have internal bus booking system. Although that system operates effectively, there is a growing demand and trend in moving to web based bus ticket reservation systems. Users’ (passengers) self-interaction in ticket reservation has been changing in a positive way with the introduction of these new systems. With the evolution of these, the traditional bus ticket reservation space can be changed. Therefore, the company is planning to refurbish and widen their services by introducing a novel web-based bus ticket reservation system to reach out to more local and foreign passengers. This system is used for long-distance intercity buses in the private sector.

This system includes a front end, a back end, and a database. Admins, other employees, bus conductors, and ticket inspectors can be accessed from the back end. Passengers can be accessed from the front end. Special features of this system include such things as: users can view their travel history, view timetables, select the desired route, and check seat availability. Depending on the cancellation policy, users can cancel their ticket. The system allows users to send alerts and notifications about their booking status, changes in schedules, or other relevant information.

## 1.2 Aim & objectives

### 1.2.1 Aim

The aim of this project is to design and develop web based bus ticket reservation system for Serendib travel company with the help of JQuery,PHP, HTML, CSS, JQuery, JS and MYSQL.

### 1.2.2 Objectives

The objectives of this project are
To provide a web-based bus ticket booking function to passengers and to purchase bus tickets without standing in line.

- To enable passengers to check the availability and types of buses online and to check the departure time and arrival time for each bus.
- To update and cancel payments, routes, and vehicle records under admin user privilege.
- To be the ability of passengers to cancel their reservation.
- To buy tickets 24/7 over the internet.
- To reduce the number of frauds that occur in manual systems.
- To broaden the use of the technology (information technology).
- To reduce the manual work.

## 1.3 Intended audience or “beneficiaries” of the work done

The web-based bus ticket reservation system (WBBTRS) has a front end and a back end. The intended audience here includes both internal and external system users. Internally, as backend users, we have the system users (developer, administrators, other employees, bus owner, etc.). And externally, as front-end users, we have passengers. As beneficiaries, the internal users can get the solutions and facilities through the backend of the system, and in the same way, the external users can also get the solutions and facilities through the frontend system.

## 1.4 Scope of the project

The web-based bus ticket reservation system will allow users to book bus tickets online from anywhere with an internet connection. The system will provide a user-friendly interface where users can select their desired departure and arrival locations, choose their preferred travel dates and times, and select their seats. Users will also be able to view the bus schedules, routes, and ticket prices.

The system will be designed to cater to both individual and group bookings, and it will allow users to make payments online through various payment gateways. The system will also generate e-tickets and send confirmation emails to users, providing them with all the necessary details about their journey.

The system will be designed to manage the booking process efficiently, including seat availability, bus schedules, and ticket prices. It will also provide administrators with the ability to manage user accounts, view bookings, and generate reports. The system will be designed to ensure the security of user information and transactions, with appropriate security measures implemented to prevent unauthorized access or data breaches.

The web-based bus ticket reservation system will be designed to streamline the ticket booking process and improve passenger satisfaction while providing administrators with efficient management tools to enhance the overall user experience.

### 1.4.1 Admin panel with Bus and Route management

On the other hand, an admin has full control over the system. An admin has the right to manage the proper flow of the system. The passengers can simply sign up and use the system. The admin can oversee the pieces of information about registered passengers. With it, the very first step of the management side falls upon the arrangement of bus records. The admin has to provide a few details, such as the name of the bus, its capacity, and its category. In addition, the admin has the right to manage bus routes by providing various details. Indeed, these details include the names of address points. Lastly, now the admin can proceed with arranging schedules for various times.

### 1.4.2 Schedule management

It is important to have a schedule arrangement at the right time for the passengers in order to maintain their reservations. The admin must arrange a schedule for the future dates because the system only allows reservations for future dates. Therefore, the system allows adding a range of schedules and a one-time schedule for managing bus schedules. In terms of a one-time schedule, the user has to select the bus, route, category, date, and time. Whereas, the system allows another date range field under a range of schedules. In fact, a one-time schedule works only for the specific date mentioned, whereas the other one stays active until its last-mentioned date. An admin can make changes to these schedules anytime and remove them from the records too.

### 1.4.3 Reports, earnings, and feedback management

In general, the system prepares a report highlighting the details of a particular reserved route and bus. Of course, this whole report comes in the format of a PDF. It shows details such as the passenger's name, seat number, and class code, along with the date and time. Similarly, the admin can view all the earning history along with the total collection and seat capacity. Meanwhile, this section represents the total number of ticket sales per route and bus with their respective collection figures. Furthermore, the passenger can send feedback, and all the feedback is listed under the admin dashboard, where he or she can take action. Besides, the administrators can search for details using the passenger's ticket number and have an overview of total earnings with other fields too.

### 1.4.4 Ticket booking and its management

Once the passenger has selected the seats, the system allows them to book and pay for the tickets online using a secure payment gateway. If you want to cancel the tickets, the system allows you to cancel their reservations and issue a refund based on the cancellation policy. Admin can edit, modify, search, and add records.

### 1.4.5 Email notifications

The system provides email notification facilities to passengers regarding their bookings, cancellations, and any updates or changes to their travel schedule.

### 1.4.6 Feedback management

The system provides the facility of forwarding messages to its passengers. And the admin manages all the feedbacks.

## 1.5 Assumptions and dependencies

A web-based bus ticket reservation system has several assumptions and dependencies that need to be considered. One of the primary assumptions is that users have access to the internet and a device with a web browser to access the system. Additionally, the system assumes that there is a database that stores bus schedules, seat availability, and user information and that online payments can be processed securely and reliably. The system also assumes that users can select their desired bus route, date, time, and seat preferences and that it can handle multiple concurrent users and transactions. Confirmation emails or SMS are also assumed to be sent to users upon successful ticket reservation, and the system can generate reports on ticket sales, bus occupancy rates, and revenue.

On the other hand, the system's dependencies include reliable internet connectivity to ensure users can access the system and complete transactions. A secure payment gateway is also necessary to process online payments and protect user data. A reliable and scalable database system is also critical to storing and retrieving data efficiently. The system may depend on external APIs or services to retrieve information such as bus schedules or to send confirmation emails or SMS to users.

Additionally, third-party tools or libraries may be required to implement features such as user authentication, ticket booking, or payment processing. Understanding these assumptions and dependencies is crucial to developing a functional and effective web-based bus ticket reservation system.

## 1.6 Broad summary of important outcomes

The implementation of a web-based bus reservation system can have numerous important outcomes that benefit both bus owners and passengers. The system's ability to provide a convenient and efficient platform for ticket reservations can significantly enhance the user experience, leading to increased passenger satisfaction and loyalty. Additionally, the system's online payment processing capabilities can improve revenue generation for bus owners, as they eliminate the need for passengers to physically visit a ticket counter to make payments.

The system's real-time tracking capabilities provide passengers with a sense of security and reassurance, as they can track the status of their reservation in real-time. Moreover, the system's ability to generate reports on ticket sales, bus occupancy rates, and revenue can provide valuable insights that can be used to improve operations and decision-making. These insights can help bus owners identify trends and patterns in their business and make data-driven decisions that improve their overall efficiency and profitability.

Overall, the implementation of a web-based bus reservation system can have a transformative impact on bus operations and the passenger experience. By leveraging technology to provide a more convenient, efficient, and secure platform for ticket reservations, bus operators can drive revenue growth and enhance passenger satisfaction, leading to long-term success and profitability.

## 1.7 Outline of the report

**Chapter 1** describes the introduction and background of the project. It sets the stage for the reader to understand the aim, objectives, and scope of the project.

**Chapter 2** describes some of the other approaches to solving similar types of problems under the heading "Review of Others." It will also compare the approach of this project with others. This project includes eight similar systems. It contains five local domains and three foreign domains.

**Chapter 3** describes the analysis and design stages of the proposed system by using requirement specifications and UML diagrams such as case diagrams, ER diagrams, activity diagrams, sequence diagrams, and class diagrams. Further, this section discusses all technology adaptations and user interfaces.

**Chapter 4** describes the implementation part of the system. Here, the focus is on integrating the backend and dashboard with the database. Flowcharts are used to illustrate it.

**Chapter 5** describes the testing and evaluation section of the proposed system. In this section, some testing methods and test cases will be discussed. And here will be demonstrated some test results by comparing them with bar graphs and other graphs.

**Chapter 6** is the final chapter. This chapter is the discussion phase of the proposed system. In this section, the conclusion, the future plan of the project, the challenges that I faced, the justification of the choice of tools, and my personal reflection will be discussed. Readers can also get a comprehensive understanding of the progress of the project here.

**Appendix:** After the chapter section, we can include an appendix to add some supplementary materials, such as forms, documents, the source code of the system backend and frontend, or an extension of the main chapter.

## 1.8 Summary

This chapter's sections include an introductory section, aim and objectives, the intended audience or beneficiaries of the work done, project scope, assumptions, and dependencies, a broad summary of important outcomes, and a report outline.

The next chapter discusses "Review of the Literature." It compares the drawbacks and advantages of similar systems. It includes eight similar systems, including five local domains and another three foreign domains. Local domains are BusSeat.lk, BusTicket.lk, sltb.eseat.lk, Superline.lk, and Dailybus.lk. and foreign domains are redBus, CheckMyBus, and ZingBus.

# Chapter 2 - Literature Review

## 2.1 Introduction

Bus transportation has been an important mode of transportation for many years, and with the advent of technology, it has become easier for passengers to make bus ticket reservation online. In this chapter will be discussed the existing system as well as similar web-based bus ticket reservation system. Also, will be discussed the drawbacks of the existing system. In addition will be talked about who the stakeholders are in our system here. Here we talk about why we are creating a new system.

And here will be discussed eight similar bus ticket reservation systems. They are classified as local and foreign. Here will be explained five local domains and three foreign domains in detail. Here will be delved deeper into the unique features and drawbacks of these systems finally will be analyzed all the facts in a summary table.

## 2.2 Problem statement

Currently, tickets are issued to passengers through counters or ticket machines. These are all traditional methods. The traditional methods of booking bus tickets are often time-consuming, inconvenient, and lacks a centralized platform for users to compare prices and select the best option for their travel needs. Many individuals still struggle to book bus tickets due to limited access to the bus station or travel agent, or long queues and wait times. The traditional bus ticket booking process also raises security concerns, as users are required to provide personal and financial information at the bus station or travel agent.

## 2.3 Stakeholders within problem area

### 2.3.1 Internal stakeholders

The internal stakeholders of a web-based bus ticket reservation system are individuals or groups within the bus company who have a vested interest in the successful development, implementation, and operation of the system. Some of the key internal stakeholders of a web-based bus ticket reservation system include:

#### 2.3.1.1 *Management*

The management team is responsible for making strategic decisions about the development and operation of the web-based bus ticket reservation system. They will play a key role in ensuring the system meets the needs of the company and delivers desired outcomes.

#### 2.3.1.2 *IT department*

The IT department is responsible for the technical design, development, and maintenance of the web-based bus ticket reservation system. They will ensure the system is secure, reliable, and meets the needs of the company and its users.

#### 2.3.1.3 *Operations team*

The operations team is responsible for the day-to-day management of the web-based bus ticket reservation system. They will play a key role in ensuring the system is used effectively and efficiently by all stakeholders.

#### 2.3.1.4 *Marketing team*

The marketing team is responsible for promoting the web-based bus ticket reservation system to potential users and building awareness about its benefits and features.

#### 2.3.1.5 *Customer service team*

The customer service team is responsible for providing support and assistance to users of the web-based bus ticket reservation system. They will play a key role in ensuring the system is user-friendly and meets the needs of users.

#### 2.3.1.6 *Finance team*

The finance team is responsible for managing the financial aspects of the web-based bus ticket reservation system, including make daily reporting, budgeting, revenue forecasting, and cost control.

Each of these internal stakeholders will play a unique and important role in the success of the web-based bus ticket reservation system. It is important that they work together to ensure the system delivers the desired outcomes and meets the needs of the company and its users.

### 2.3.2 External dtakeholders

The external stakeholders of a web-based bus ticket reservation system can include:

#### 2.3.2.1 *Passengers*

The primary users of the system who will use it to search for and book bus trips, manage their itineraries, and make payments.

#### 2.3.2.2 *Bus owner*

The people who own the buses they will use the system to manage bookings and passenger information related to them.

#### 2.3.2.3 *Ticket checker*

Bus ticket checkers will also use the system to manage bookings and passenger information related to the bus.

#### 2.3.2.4 *Payment providers*

Companies that provide payment-processing services, such as credit card companies, payment gateways, and online wallets.

#### 2.3.2.5 *Regulatory agencies*

Government organizations that regulate the transportation industry and may set standards for online reservation systems.
It is important for the development team to consider the needs and requirements of these external stakeholders when designing and building the web-based bus ticket reservation system.

## 2.4 Motivation and significance of the project

Buses are the most widely used form of public transportation in Sri Lanka. Buses are a common sight on the country's roads, and they operate on both short and long-distance routes. While buses are a convenient and affordable way to travel in Sri Lanka, they can get crowded, especially during peak traffic times. Nevertheless, they are an essential part of the country's transport infrastructure and remain a popular mode of public transport for many Sri Lankans. However, the current methods of reserving tickets for buses are very inefficient. Long queues are time-consuming and tiring, especially for long-distance buses. Booking tickets over the phone was a difficult task, as the phone line was often down or busy. Those reasons motivated the development of a new, effective web-based bus reservation system.
This system includes a front end, a back end, and a database. Admins, other employees, bus conductors, and ticket inspectors can be accessed from the back end. Passengers can be accessed from the front end. Special features of this system include such things as: users to view their travel history, view timetables, select the desired route, and check seat availability. Depending on the cancellation policy, users can cancel their ticket if required. The system allows users to send alerts and notifications about their booking status, changes in schedules, or other relevant information. Provision is also made for necessary passenger support and inquiries during the booking process.

The passenger is also allowed to provide feedback regarding the system. Through data and information such as ticket sales, passenger numbers, and route popularity, operations are enabled to continue optimally.

## 2.5 Critical evolution of existing systems

### 2.5.1 Local domain systems

2.5.1.1 *BusSeat.lk*

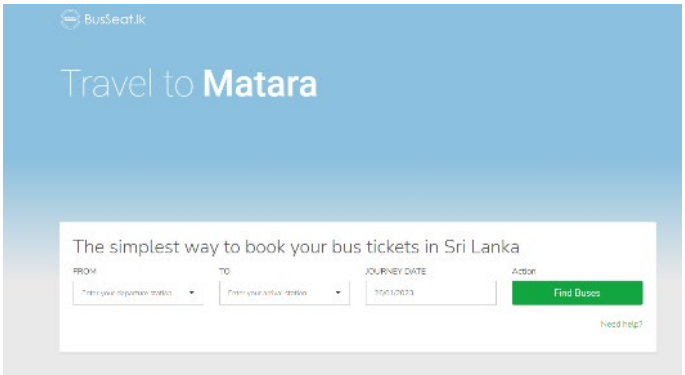



*Figure 2-1: BusSeat.lk*

Public transport, specifically travel by bus, is a large and growing business in Sri Lanka and other countries; hence, innovatory mechanisms for bus ticket booking systems are imperative to adequately provide a user-friendly interface for the passenger to reserve a seat on a journey and an effective compilation and maintenance of individual records for bus companies and regulatory bodies. For that, Techkitez (Pvt) Ltd developed "BusSeat.lk," an innovative online bus ticket booking platform.

An efficient transport system is the key to the development of a country. Although the transport system in Sri Lanka has seen many improvements over time, the bus ticket booking platform still lacks a significant technological advantage. Pens and papers are unable to cope with the massive booking requests from passengers all over the island. Since our inception, we have seen that the general public and tourists alike are not completely aware of bus timetables, how to find a bus, and where to find it. After using our platform, around 100,000 people from all parts of the country have given us very positive feedback that the online bus ticket booking platform has improved their journey with efficiency and convenience, two pertinent and fundamental factors necessary to improve Sri Lanka's transport system.

BusSeat.lk now allows users to book trips to around 50 cities in Sri Lanka. There are 115 bus-booking agents registered with BusSeat.lk.

Their job is to collect payments from people who have reserved their seats via the website but are unable to make an online payment for the trip at the time of reservation. BusSeat.lk is also assisting the tourism industry by providing a very user-friendly platform for tourists to review and book buses to travel to their preferred destination, based on their itinerary.

Busseat.lk currently provides bus routes to almost all tourist destinations on the island. Hence, we are proud to say that we have played a role in assisting a lot of tourists to find buses and pre-book them. Tourists pre-book the bus seats from their own country before visiting Sri Lanka. BusSeat.lk offers women the option of booking a "ladies' seat" on long journeys. This is a measure to ensure passenger safety. This was done with the idea of preventing violence against women on long journeys in mind.

As a drawback of BusSeat.lk, it is a clear condition that the seat next to a women's seat should remain empty when there is no passenger opting for it. It is a loss of income for bus company owners. No alternative has been offered in this regard (Anon., n.d.).

### 2.5.1.2 *BusTicket.lk*

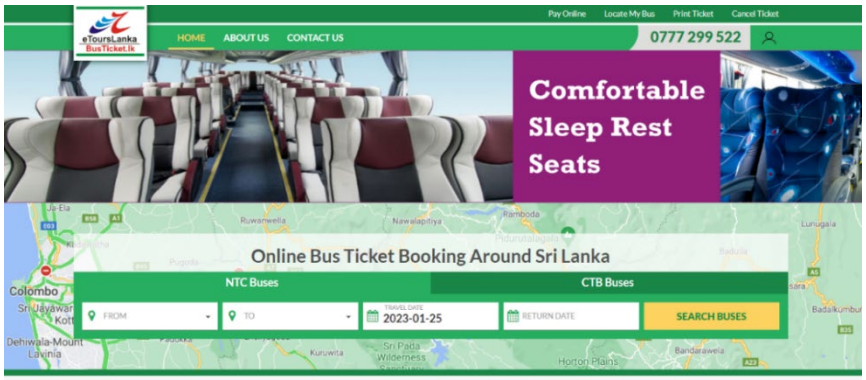



*Figure 2-2: BusTicket.lk*

eToursLanka (Pvt) Ltd., which has been the pioneer of online bus ticket booking services in Sri Lanka since 2010, In this segment, the company has launched a web site, www.BusTicket.LK, with the collaboration of NTC and CTB-registered buses in Sri Lanka. Long-distance travelers can use this service to book any bus seat for anywhere in Sri Lanka and receive the bus ticket details on their mobile phone; the operation is done online.

Presently, most of the passengers are unable to make their itineraries because of the unscheduled times of departure from the boarding point by the private bus operators.

We have a system with all private bus operators to obtain the details of the bus's operating route and the time of departure. so that the online booker is well informed of the exact bus number, route, time of departure, and fare of the particular route.

All routes and destinations will be announced on our booking platform, and we will monitor passenger boarding points to ensure the operation's success.

The online booking services' vision is to provide an easy and convenient way for passengers to book their tickets by using our system, paying easily with Visa, Mastercard, eZcash, mCash, Reload, Sampath Bank, or Commercial Bank Deposits or Transfers, and receiving the bus details and mTicket instantly to their mobile or email.

BusTicket.lk has plenty of advantages. but that doesn't mean that there aren't any minor disadvantages. Some of them are forgetting to include an online inquiry management facility for passengers, excluding passenger feedback, not displaying a bus fare table, and so on (Anon., n.d.).

### 2.5.1.3 *sltb.eseat.lk*

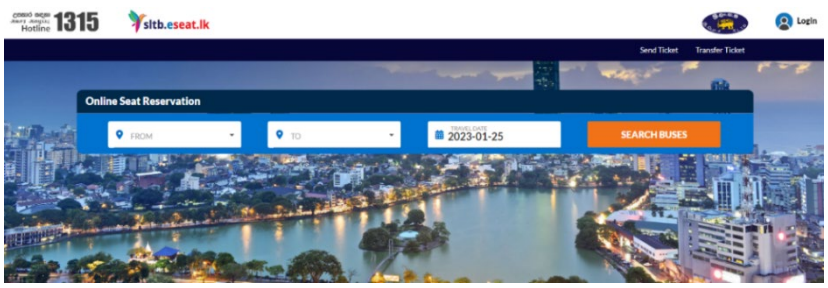



*Figure 2-3: sltb.eseat.lk*

sltb.eseats.lk is a collaboration between Sri Lanka Transport Board (SLTB) and Hands Global Holdings (Private) Limited (HGHL). SLTB is the state-owned transportation service in Sri Lanka and is the largest and most reliable service provider in the country. It has a large fleet of buses covering the entire country, from highway buses to buses operating in the remotest part of the country.

They provide online bus ticket booking, reservations through our 24-hour hotline, 1315, bus tracking, and fleet management services. Once you reserve a seat, you can change or modify it, subject to the seat cancellation policy. Furthermore, you will be promptly notified if a bus that you have booked has been changed or if schedules change. Everyone booking through sltb.eseat.lk and using our mobile app can track the bus online, and passengers can reserve their seats one month ahead of the journey.

Some of the drawbacks of the system are as follows: On cancellation of your tickets, the entire booking will be canceled, and all tickets booked under the canceled booking will be canceled. not be cancelled as a partial.

Furthermore, Sltb.eseat.lk is not responsible for bus quality, journey quality, passenger safety, or bus schedules (Anon., n.d.).

### 2.5.1.4 *Superline.lk*

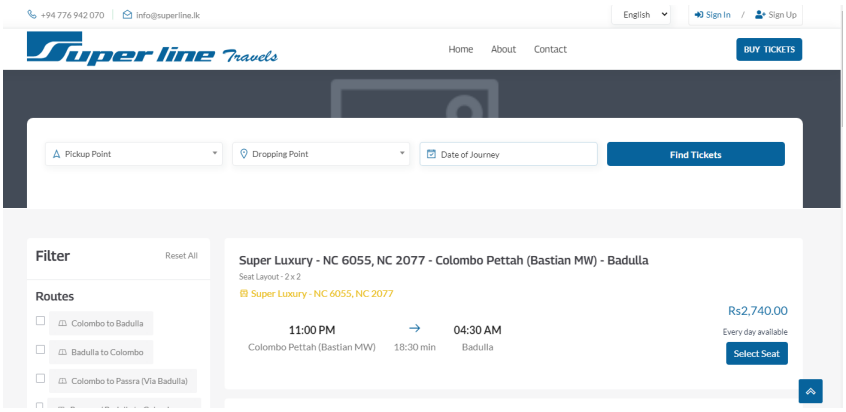



*Figure 2-4: Superline.lk*

Superline.lk belongs to Superline Line Travels. Superline Travels Ltd. is a transport solutions provider in Sri Lanka, specializing in super-luxury coaches. Their fleet of coaches operates daily to and from Colombo, Batticola, Akkaraipaththu, and Trincomalee. Long-distance travelers could use this service to book any bus seat in Sri Lanka from one location. Superline Bus Service is located in a busy area, and they are covering at least 630 places around it on Srilanka-Places.com. Presently, most of the passengers are unable to make their itineraries because of the unscheduled times of departure from the boarding point by the private bus operators. We have made a system to obtain the details of the bus's operating route and the time of departure. so that the online booker is well informed of the exact bus number, route, time of departure, and fare of the particular route. All routes and destinations will be announced on our booking platform, and we will monitor passenger boarding points to ensure the operation's success. Bus booking covers all booking requirements, including all online bus seat reservations, and ensures you have a safe and comfortable journey. Sri Lanka offers a unique online bus booking platform for all travel within Sri Lanka to avoid hassles and inconveniences, reach your destination in comfort, and make seat reservations.

Superline has both advantages and disadvantages. Some disadvantages are: It does not include any policies like the privacy policy, terms and conditions, or ticket policy.

How can the public have confidence in the system without including them in it? And they have forgotten to include passenger feedback and an online inquiry management facility for passengers (Anon., n.d.).

### 2.5.1.5 *Dailybus.lk*

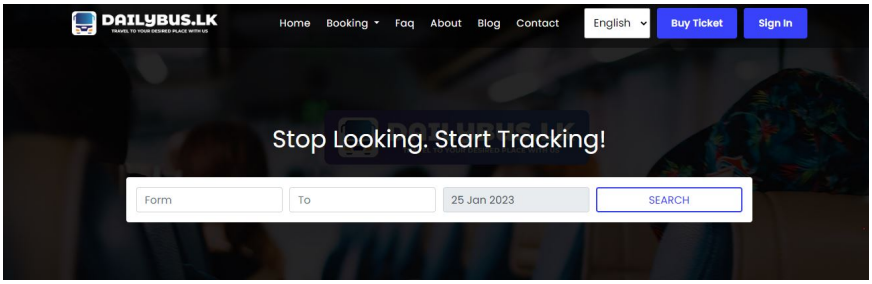



*Figure 2-5: Dailybus.lk*

"Dailybus.lk" is a system developed by Redcode. This is an innovative online bus ticket booking platform that envisions a hassle-free and improved public transport system in Sri Lanka. Since our inception in 2018, we have been able to provide a large number of services through public transport. They have more than 1,000 passengers and have been in this industry for over 12 years.

Bus travel in particular is a growing business in Sri Lanka and elsewhere. Hence, innovative mechanisms are essential for bus ticket booking systems to adequately provide a user-friendly interface for the passenger to book a seat on a journey and to effectively compile and maintain individual records for bus companies and regulatory bodies. Our online bus booking portal can be considered an ideal solution for this.

Some of the drawbacks of the system are not providing user friendly views to passengers. and not including the bus fare table and graphical overview of the seats. On the other hand, developers had missed some of the special features of the system (Anon., n.d.).

### 2.5.2 Forign domain systems

#### 2.5.2.1 *redBus*


*Figure 2-6: redBus*

redBus is India's largest online bus ticketing platform that has transformed bus travel in the country by bringing ease and convenience to millions of Indians who travel using buses. Founded in 2006, redBus is part of India's leading online travel company, MakeMyTrip Limited (NASDAQ: MMYT).

By providing the widest choice, superior customer service, the lowest prices, and unmatched benefits, redBus has served over 18 million passengers. redBus has a global presence with operations in Indonesia, Singapore, Malaysia, Colombia, and Peru, apart from India.

redBus is India's leading bus ticketing company that offers incredible offers to its passengers. If you choose redBus, you will find undeniable benefits. As a result, redBus has millions of satisfied and loyal passengers around the world. redBus has a number of outstanding features. Like: With Safety+, they have brought a series of measures like sanitary buses, mandatory face masks to ensure safe travel for their passengers, and they always try to provide maximum service to their passengers at the lowest price with offers. Thus, you get innovative and unique benefits. Giving has attracted a highly stratified target audience to redBus. redBus definitely makes it more convenient and comfortable for its passengers by providing multiple travel booking offers, such as: thousands of routes across multiple travel options, efficient customer service, low fares for tickets; pilgrimage tours; mobile ticketing apps; bus hirings; discounts, deals, and festival offers; etc. redBus has had enormous success since its launch in 2006. There are numerous bus services available 24 hours a day that cater to all of the traveler's needs. Its business model is quite simple and strong, but most importantly, redBus is wildly famous and welcomed among the common people.

Also, redBus has several disadvantages. redBus has a passenger query management problem. Because they have to handle thousands of passenger inquiries per day. So, some of them may be missed. As a result, some passengers give negative feedback. I think they will overcome that problem soon (Anon., n.d.).

### 2.5.2.2 *CheckMyBus*


*Figure 2-7: CheckMyBus*

CheckMyBus is a search engine for intercity bus trips all over the world. Founded in 2013, CheckMyBus began as a small company with a big vision: to revolutionize the intercity bus market around the world. Since its beginning, CheckMyBus has grown exponentially with 21 domains, featuring bus content in over 80 countries around the world. Moving forward, they plan to continue growing and expanding our network to help users travel, no matter where they are or want to go. Their mission is to become the world's leading bus search engine and change the way people travel.

Its search engine is available in a dozen languages and allows passengers to compare many bus providers to find the most affordable transfer. The price comparison platform also provides travel tips and other useful information. You can use it to compare and reserve a bus ticket using any web browser or the CheckMyBus phone app. CheckMyBus is available on both Google Play and the App Store for free of charge. The process is the same as on the website, and the app has all the helpful features. Search through the list and use filters to find what you're looking for effortlessly. The app allows you to find the best deal in any location with just a few clicks. CheckMyBus is a reliable search engine that helps you plan your journey and reduce your expenses! It's a price comparison tool that allows you to find (but not book) the most affordable bus ticket.

When you select your desired destination, CheckMyBus will redirect you to the official bus company's booking page. So, it doesn't help you book, but it helps you get the most affordable transfer to your destination!

This site seems to provide a service for downloading books. This may be a legitimate service, but we have found several sites run by scammers in the past. Scammers seem to offer digital books at a discount or for free.

Passengers also complain that it does not respond quickly to their complaints. I think they are trying to solve those issues and provide better service to the passenger (Anon., n.d.).

#### 2.5.2.3 *Zingbus*

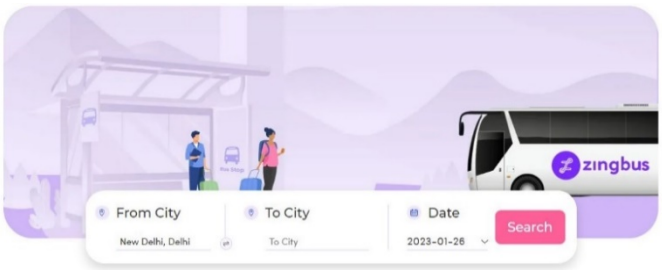



*Figure 2-8: Zingbus*

Zingbus is an emerging brand in the intercity bus travel industry that aims to provide an affordable and dignified travel experience.

India's fastest-growing intercity mobility startup, which aspires to provide safe, reliable, and affordable travel to its users across India, is set to raise the bar of the intercity travel experience in India. Zingbus has grown into a prominent family, connecting 350+ cities, serving almost 1 million passengers, and flaunting a fleet of 250+ buses. We successfully launched four zingbus lounges in Ahmedabad, Delhi, Lucknow, and Manali, with Manali being India's largest lounge ever.

Zingbus has launched the first-of-its-kind "Pay at Bus" facility for its passengers. The new service allows Zingbus passengers to book their travel tickets online through the app or web and pay the amount of their ticket while boarding the bus. Zingbus has deployed the "Pay at Bus" facility on every route and service, starting with Delhi, Uttar Pradesh, and the Punjab belt. The new service facility enables those passengers who are not comfortable with pre-payments or digital payments to book and travel with Zingbus with ease. The dynamics of the "Pay at Bus" facility are equivalent to those of the "Cash on Delivery" facility prompted by the e-commerce industry.

However, with the integration of "Pay at Bus,"Zingbus has focused on offering a more flexible payment method for passengers' travel needs. Moreover, after integrating this feature, Zingbus has seen over 15% growth in app bookings. Zingbus believes in revolutionizing the travel industry with progressive technology-based features.

Despite being an advanced technology-based bus service, Zingbus is not without its flaws. Although Zingbus is a quality service, it sometimes receives passenger complaints that it falls far short of their own standards in terms of comfort, aesthetics, and cleanliness.

Their team also accepted it. Also, their team is now coordinating to resolve this and make sure no other passenger has the same experience (Anon., n.d.).

## 2.6 Summary table

*Table 2-1: Summary table*

| **Requirements** | **Local Domain** | | | | | **Foreign Domain** | | | **Manual System** | **Proposed System** |
|---|---|---|---|---|---|---|---|---|---|---|
| | **Bus Seat.lk** | **Bus Ticket.lk** | **sltb.eseat.lk** | **Super line Travels** | **DAILYBUS.LK** | **redBus** | **Check My Bus** | **Zingbus** | | |
| Admin Sign in | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ |
| Passenger Registration/Login | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ |
| Bus owner details | ✓ | ✓ | ✓ | X | X | ✓ | X | ✓ | X | ✓ |
| Bus ticket checker details | X | X | ✓ | X | X | X | X | X | X | ✓ |
| Inventory Management | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Bus route Management | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Bus time Scheduling | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Seat details | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Bus fare details | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Ticket printing | X | ✓ | ✓ | X | ✓ | ✓ | ✓ | ✓ | X | ✓ |
| Push notifications | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Confirmation (mail) | X | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ |
| Ticket cancellation | X | ✓ | X | X | ✓ | ✓ | ✓ | ✓ | X | ✓ |
| Concurrent transaction | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ |
| Secure online payment gateway | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | X | ✓ |
| Income & other reports generate | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Feedback management | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

## 2.7 Summary

In this literature review, I examined a total of eight bus reservation systems. It includes five domestic domain systems and three foreign domain systems. Local domains are BusSeat.lk, BusTicket.lk, sltb.eseat.lk, Superline.lk, and Dailbus.lk. Foreign domains are redbus, checkmybus, and zingbus.

That summarizing table describes what the include and exclude functions are in those similar systems. For example, when it comes to fully and partially canceling, some systems allow both, while others only allow fully canceling. BusTicet.lk and Dialybus.lk, as well as redBus, CheckMyBus, and zingBus, include that feature; others exclude it. The right and wrong marks are displayed here. The right mark indicates which features are included, and the wrong mark indicates which features are excluded from that system. Here is a new column for manual systems. Because some existing systems may be manual processes, some tasks may be entered manually. The proposed system is our developing system. It includes all the functions.

In the next chapter will discuss the analysis and design stages of the proposed system by using UML diagrams such as case diagrams, ER diagrams, activity diagrams, sequence diagrams, and class diagrams. Further, this section discusses all technology adaptations and user interfaces.

# Chapter 3 - System Design

## 3.1 Introduction

This section describes all the analysis and design stages of the proposed system by using all UML diagrams, which are Use case diagrams, activity diagrams, sequence diagrams, class diagrams, and ER diagrams are all included here. The ER diagram, which displays the logical view of the database designs, is used to create the relational schema for the system. And this includes all software, hardware, and database technology. As well as here, it consists of the user interfaces, through which the backend and frontend interfaces give information. Also, it includes a high-level diagram of the adapted systems and technologies.

## 3.2 High level diagram

### 3.2.1 A three-tier client/server architecture

A 3-tier application architecture is a modular client-server architecture that consists of a presentation tier, an application tier and a data tier. The data tier stores information, the application tier handles logic and the presentation tier is a graphical user interface (GUI) that communicates with the other two tiers. The three tiers are logical, not physical, and may or may not run on the same physical server (Anon., n.d.).

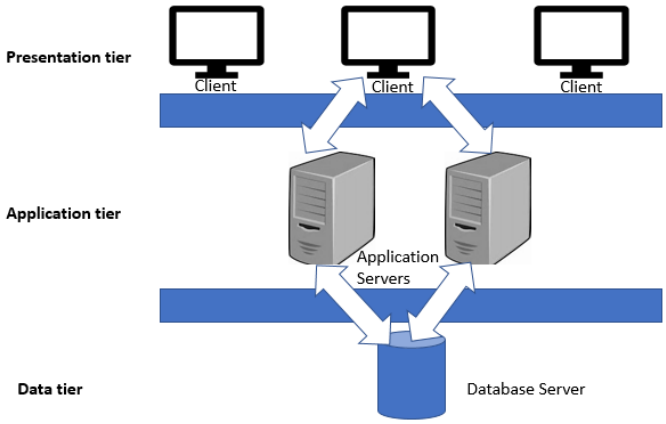



*Figure 3-1: Three-tier client/server Architecture*

## 3.3 Technology adapted

### 3.3.1 Software

3.3.1.1 *Web development technologies*

The whole system was programmed in Visual Studio Code (VS Code). And its use as web development technologies such as HTML, Bootstrap, CSS, JavaScript, and PHP, JQuery are used to create the user interface and functionality of the website.

3.3.1.2 *Database technologies*

This system's database is created by MySQL and XAMPP web server software.

3.3.1.3 *Software requirements*

- Operating System : Windows 10 64-bit Operating System
- Internet connection

### 3.3.2 Hardware

3.3.2.1 *Server*

A server is the heart of any web-based system. It's a computer that stores all the data and processes the requests from users. The server hardware should be reliable and have enough processing power, memory, and storage to handle the traffic and requests from users.

3.3.2.2 *Network infrastructure*

A high-speed internet connection is necessary to ensure that the web-based system runs smoothly. You may need to invest in networking hardware such as routers, switches, and firewalls to ensure that the network is secure and reliable.

3.3.2.3 *Hardware requirements*

- Processor : Intel Core i3 processor, third generation, running at 2.60 GHz
- RAM : 512 MB
- Hard Disk : 5 GB
- Graphic Card : 1GB

## 3.4 Requirement specification

### 3.4.1 Introduction

Gathering requirements for a web-based bus ticket reservation system generally involves identifying the needs and expectations of users and stakeholders, defining the scope of the system, and specifying the required features and functions. Here are some methods I used for gathering the requirements for this web-based bus ticket booking system.

3.4.1.1 *Interviews*

Conducting one-on-one interviews with stakeholders, users, and subject matter experts to gather their insights and requirements.

3.4.1.2 *Surveys and questionnaires*

Distributing surveys and questionnaires to gather data from a large number of stakeholders, users, and passengers.

3.4.1.3 *Observation*

Observing users and stakeholders in their natural environment to understand their behavior, tasks, and workflows.

#### 3.4.1.4 *Document analysis*

Reviewing existing documents, reports, and data to gather insights and requirements for the project.

#### 3.4.1.5 *Benchmarking*

Researching and analyzing similar systems and projects to identify best practices and gather ideas and requirements.

#### 3.4.1.6 *Focus groups*

Facilitating group discussions with stakeholders and users to gather their feedback and opinions on the project

*Table 3-1: Requirement specification- Admin login*

| | |
|---|---|
| **Function** | Admin login |
| **Description** | Validate id and password to verify a valid admin |
| **Inputs** | Id and password |
| **Source** | adminsignin.php |
| **Outputs** | Directed to the admin dashboard |
| **Destination** | admin.php |
| **Requires** | Admin enter id and password |
| **Pre-condition** | The admin id and password are in the database |
| **Post-condition** | Admin can update, edit, delete all system details |
| **Side effects** | None |

*Table 3-2: Requirement specification- Passenger sign-up*

| | |
|---|---|
| **Function** | Passenger sign-up |
| **Description** | Checking if an existing user (passenger) and if not, the user can create a new account |
| **Inputs** | Full name, contact no, email, Ioc, address, password, confirm password |
| **Source** | individual_reg.php |
| **Outputs** | Directed to the passenger login page |
| **Destination** | signin.php |
| **Requires** | Passenger should enter Full name, contact no, email, Ioc, address, password, confirm password into the system |
| **Pre-condition** | The passenger has to submit a form and connect it to a database |
| **Post-condition** | Passengers can login to the system |
| **Side effects** | Passenger details are stored in the database, and passengers can display the login page (sign-in page) |

*Table 3-3: Requirement specification- Passenger login*

| **Function** | Pasenger login |
|---|---|
| **Description** | Validate the ID and password to verify a valid user (passenger) |
| **Inputs** | Id and password |
| **Source** | signin.php |
| **Outputs** | Directed to the passenger dashboard |
| **Destination** | individual.php |
| **Requires** | Passenger enter id and Password |
| **Pre-condition** | Passenger id and password in the database |
| **Post-condition** | Passengers can login to the system |
| **Side effects** | Passenger details are saved in the system, and passengers can access the login page |

*Table 3-4:Requirement specification- Bus management*

| **Function** | Bus management |
|---|---|
| **Description** | Enter bus details into the system |
| **Inputs** | Bus plate no, bus category, seat capacity, bus owner name, bus ticket checker name, bus driver name |
| **Source** | Bus.php |
| **Outputs** | Admin can edit and delete the bus details |
| **Destination** | Bus.php |
| **Requires** | Admin should enter plate no, category, seat capacity, bus owner name, ticket checker name and driver name into the system |
| **Pre-condition** | Admin should login to the system and enter details into the database, and the database should be properly working |
| **Post-condition** | Passengers can get information on bus details |
| **Side effects** | Bus details are saved in the system, and passengers can get some bus details through the bus schedule |

*Table 3-5: Requirement specification- Route management*

| **Function** | Route management |
|---|---|
| **Description** | Enter bus route details into the system |
| **Inputs** | Route no, start point, end point, distance |
| **Source** | route.php |
| **Outputs** | Admin can edit and delete the route details |
| **Destination** | route.php |
| **Requires** | Admin should enter route no, start point, end point, distance into the system |
| **Pre-condition** | Admin should login to the system, admin should enter details into the system, and the database is properly working |
| **Post-condition** | Passengers can get information on route details |
| **Side effects** | Route details are saved in the system, and passengers can get route information through the bus schedule |

*Table 3-6: Requirement specification- Schedule management*

| | |
|---|---|
| **Function** | Schedule management |
| **Description** | Enter bus schedule details into the system |
| **Inputs** | Schedule date, schedule time, route no, start point, end point, distance, estimated arrival time, departure time, ticket fee |
| **Source** | dynamic_schedule.php |
| **Outputs** | Admin can edit and delete the schedule details |
| **Destination** | dynamic_schedule.php |
| **Requires** | Admin should enter the schedule date, schedule time, route number, start point, end point, distance, estimated arrival time, departure time, and ticket fee into the system. |
| **Pre-condition** | Admin should login to the system, admin should enter details into the system, and the database is properly working |
| **Post-condition** | Passengers can get information on schedule details |
| **Side effects** | Schedule details are saved in the system, and passengers can get schedule information |

*Table 3-7: Requirement specification- Booking ticket*

| | |
|---|---|
| **Function** | Booking ticket |
| **Description** | Passengers book a ticket and enter booking details into the system |
| **Inputs** | Route details, date & time, seat details, payment details |
| **Source** | reg.php |
| **Outputs** | Directed to the payment page |
| **Destination** | pay.php |
| **Requires** | Passenger should enter route details, date & time, seat details, payment details into the system |
| **Pre-condition** | Passenger should login to the system and database is properly working |
| **Post-condition** | Passenger can successfully book a ticket |
| **Side effects** | Passenger booking details are saved in the database and admin can edit and delete the booking details |

*Table 3-8: Requirement specification- Feedback management*

| | |
|---|---|
| **Function** | Feedback management |
| **Description** | Passengers send messages to admin, and admin responds to them |
| **Inputs** | Passengers send messages |
| **Source** | feedback.php |
| **Outputs** | Admin quickly responded to the passenger’s message |
| **Destination** | feedbck.php |
| **Requires** | Passengers have inquiries |
| **Pre-condition** | Passengers and admin should login to the system |
| **Post-condition** | Passengers send messages or inquiries, and admin quickly replies to those messages |
| **Side effects** | Passenger’s feedback is saved in the database, and admins can view it |

*Table 3-9: Requirement specification- SMS alert*

| | |
|---|---|
| **Function** | Sms alert |
| **Description** | The system automatically sends sms alerts to the passengers at the time of the customer's booking, cancellation, or other important update |
| **Inputs** | Passengers book tickets |
| **Source** | pay.php |
| **Outputs** | System sends sms alert to passengers |
| **Destination** | notify.php |
| **Requires** | Passengers book or cancel tickets while buses are delayed or cancelled, and an automatic SMS generation system is integrated into the system |
| **Pre-condition** | Passengers login to the system and make bookings and payments for tickets |
| **Post-condition** | Passengers can receive an SMS for their booking or cancellation of tickets, as well as for any bus cancellations and other daily updates |
| **Side effects** | Passengers can easily get the booking details, increasing their confidence in the system |

## 3.5 UML diagrams

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques.

It is based on diagrammatic representations of software components. As the old proverb says, "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

UML was created as a result of the chaos revolving around software development and documentation. In the 1990s, there were several different ways to represent and document software systems. The need arose for a more unified way to visually represent those systems, and as a result, in 1994-1996, the UML was developed by three software engineers working at Rational Software. It was later adopted as the standard in 1997 and has remained the standard ever since, receiving only a few updates (Anon., n.d.).

### 3.5.1 Use case diagram

In UML, use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally. Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout the development process (Anon., n.d.).

#### 3.5.1.1 *High level diagram*

This diagram shows the overall view of the proposed system. It includes step-by-step passenger registration, administrator and passenger login, bus information entry by the administrator in the system, route details entry in the system by the administrator, schedule details entry in the system by the administrator, passenger schedule search and ticket booking, confirmation messages, mail, and other services. Also included.

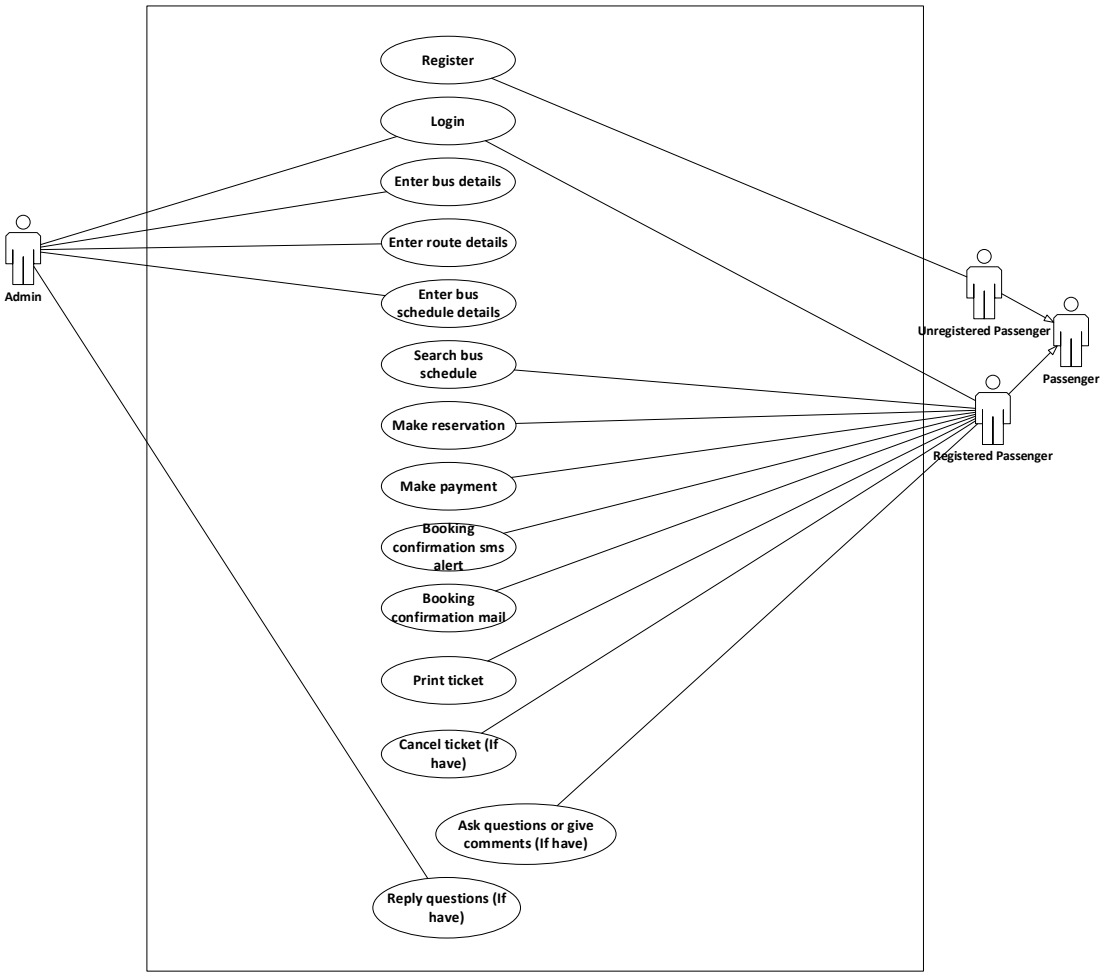



*Figure 3-2: High level diagram*

### 3.5.1.2 *Admin login*

This diagram shows the admin login function. If the administrator enters the correct email ID and password, they can access the system successfully. Otherwise, they cannot access the system and may see the error message.

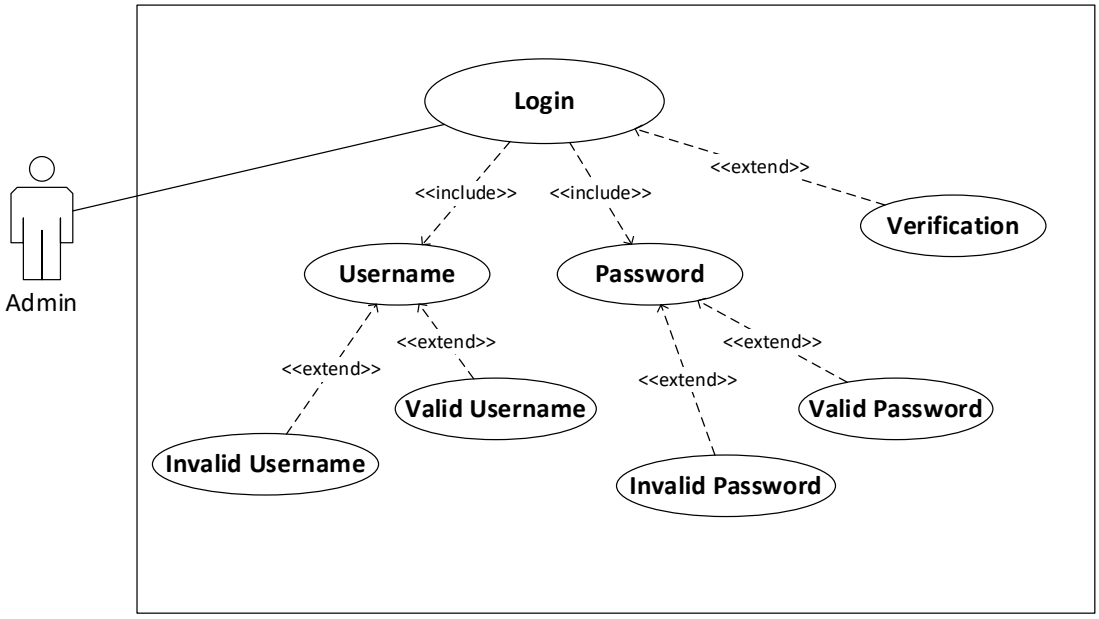



*Figure 3-3: Use case diagram- Admin login*

*Table 3-10: Use case scenario- Admin login*

| **Use Case ID:** | 001 | | |
|---|---|---|---|
| **Use Case Name:** | Admin login | | |
| **Created By:** | Lahiru | Last updated by: | Lahiru |
| **Date Created:** | 2023/03/03 | Date last updated: | 2023/03/03 |

| **Actor:** | Admin |
|---|---|
| **Preconditions:** | Admin should access website<br>Admin should possess a device to access the website |
| **Post conditions:** | Admin can login into the system |
| **Main Success Scenario:** | **Admin**<br>1. Enter username<br>2. Enter password<br>3. Authentication (verification)<br>4. Valid user name & password (if successful)<br>5. Invalid user name & password (incorrect user name or password)<br>6. Login |
| **Extensions:** | 2a. Password invalid<br>2a1. Prompt an error message<br>2a2. Re-enter password |

3.5.1.3 *Passenger Sign-up & Sign-in*

This diagram shows the passenger sign-up and sign-in functions. Firstly, check whether the passenger is already registered. If passengers are already registered, they can visit the login page. Otherwise, the passenger should create a new account. After filling out and submitting the details to the system, they can visit the login page. And if the correct email ID and password are entered, they can access the system successfully. Otherwise, they can't access the system and indicate the error message.

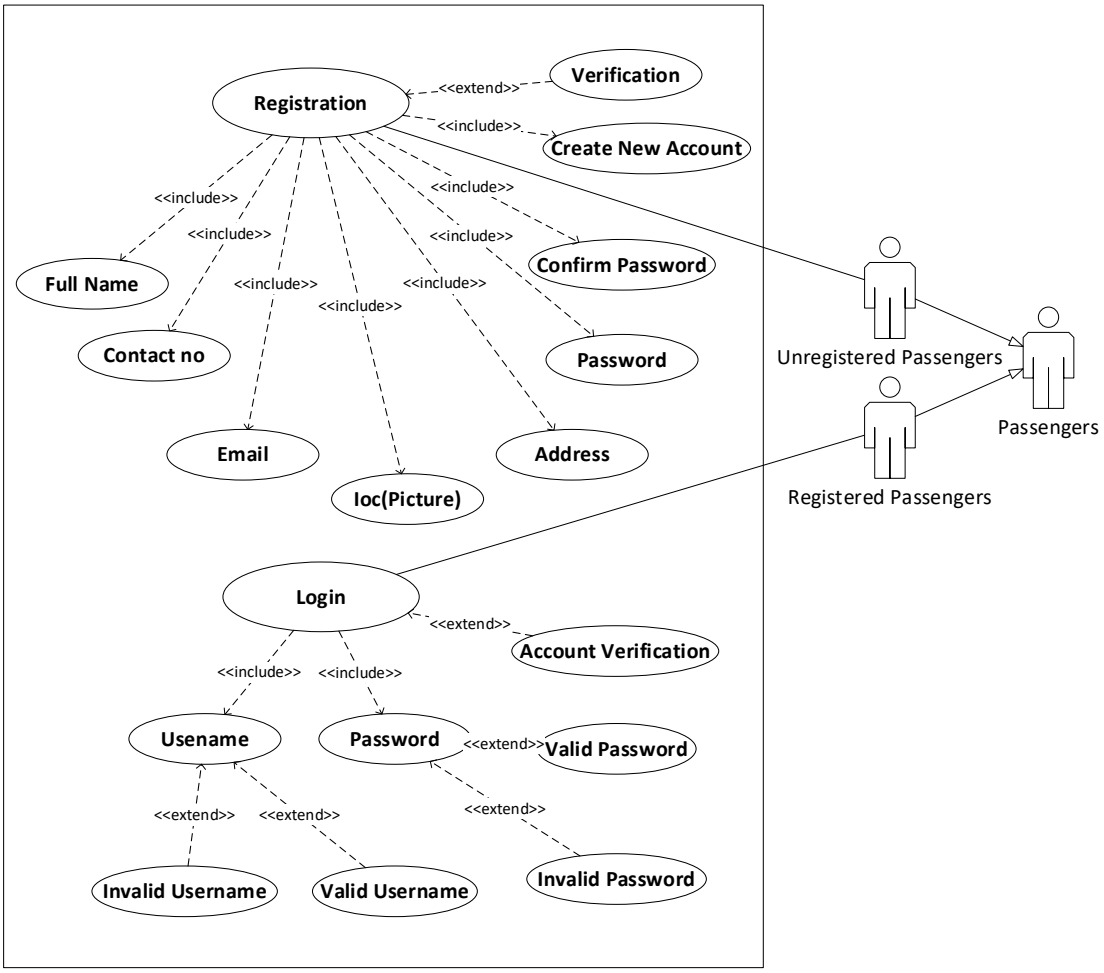



*Figure 3-4: Passenger sign-up & sign-in*

*Table 3-11: Passenger sign-up & sign-in*

| **Use Case ID:** | 002 | | |
|---|---|---|---|
| **Use Case Name:** | Passenger sign-up<br>Passenger sign-in | | |
| **Created By:** | Lahiru | Last updated by: | Lahiru |
| **Date Created:** | 2023/03/03 | Date last updated: | 2023/03/03 |

| **Actor:** | Passengers |
|---|---|
| **Preconditions:** | Passenger should access website<br>Passenger should possess a device to access the website |
| **Post conditions:** | Passenger can create a new account<br>Passenger can login into the system |
| **Main Success Scenario:** | **Passenger**<br>1. Enter user details<br>2. Click "create account"<br>3. Authentication (details)<br>4. New account confirmation<br>5. Enter username<br>6. Enter password<br>7. Authentication (verification)<br>8. Valid user name & password (if successful)<br>9. Invalid user name & password (incorrect user name or password)<br>10. Login |
| **Extensions:** | 6a. Password invalid<br>6a1. Prompt an error message<br>6a2. Re-enter password |

3.5.1.4 *Bus management*

This diagram shows bus management functions. This includes all of the bus details. Administrators can enter those details into the system. It includes such things as plate number, driver name, ticket checker name, seat capacity, etc. Passengers can view some of the details.

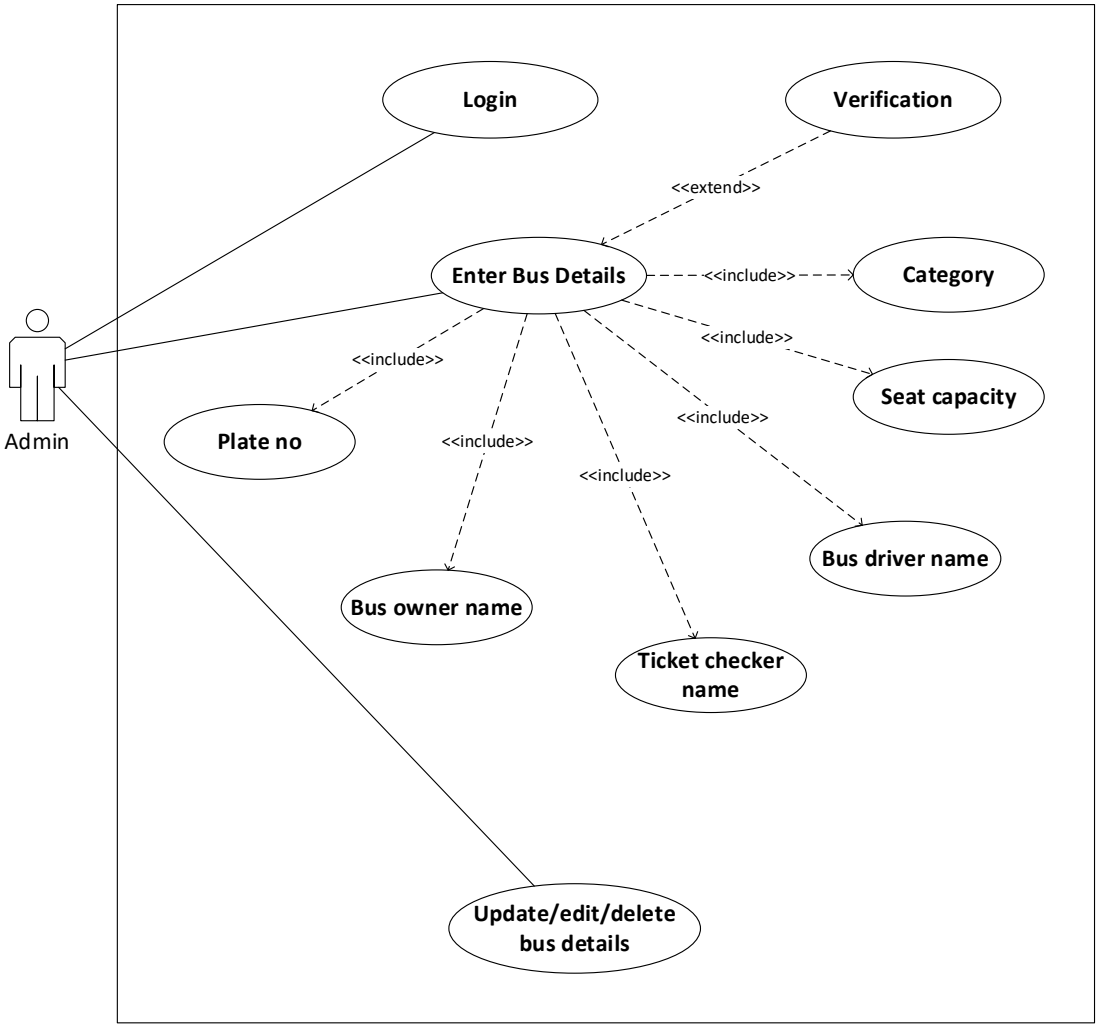



*Figure 3-5: Use case diagram- Bus management*

*Table 3-12: Use case scenario- Bus management*

| **Use Case ID:** | 003 | | |
|---|---|---|---|
| **Use Case Name:** | Bus management | | |
| **Created By:** | Lahiru | Last updated by: | Lahiru |
| **Date Created:** | 2023/03/04 | Date last updated: | 2023/03/04 |

| **Actor:** | Admin |
|---|---|
| **Preconditions:** | Admin already had registered<br>Admin already had login into the system |
| **Post conditions:** | Maintain the bus details |
| **Main Success Scenario:** | **Admin**<br>1. Admin login into the system<br>2. Enter bus plate no.<br>3. Enter bus owner name<br>4. Enter bus driver name<br>5. Bus ticker checker name<br>6. Enter seat capacity<br>7. Enter bus category<br>8. Click enter<br>9. Authentication (details)<br>10. Details store database<br>11. Update, edit, or delete bus records |
| **Extensions:** | 1a. Invalid user<br>1a1. Prompt an error message<br>1a2. Re-enter correct user id<br>1a3. Re-enter correct password |

3.5.1.5 *Route management*

This diagram shows the route management function. This includes all of the route details. Administrators can enter those details into the system. It includes such things as route number, starting point, ending point, and distance. Passengers can view these details.

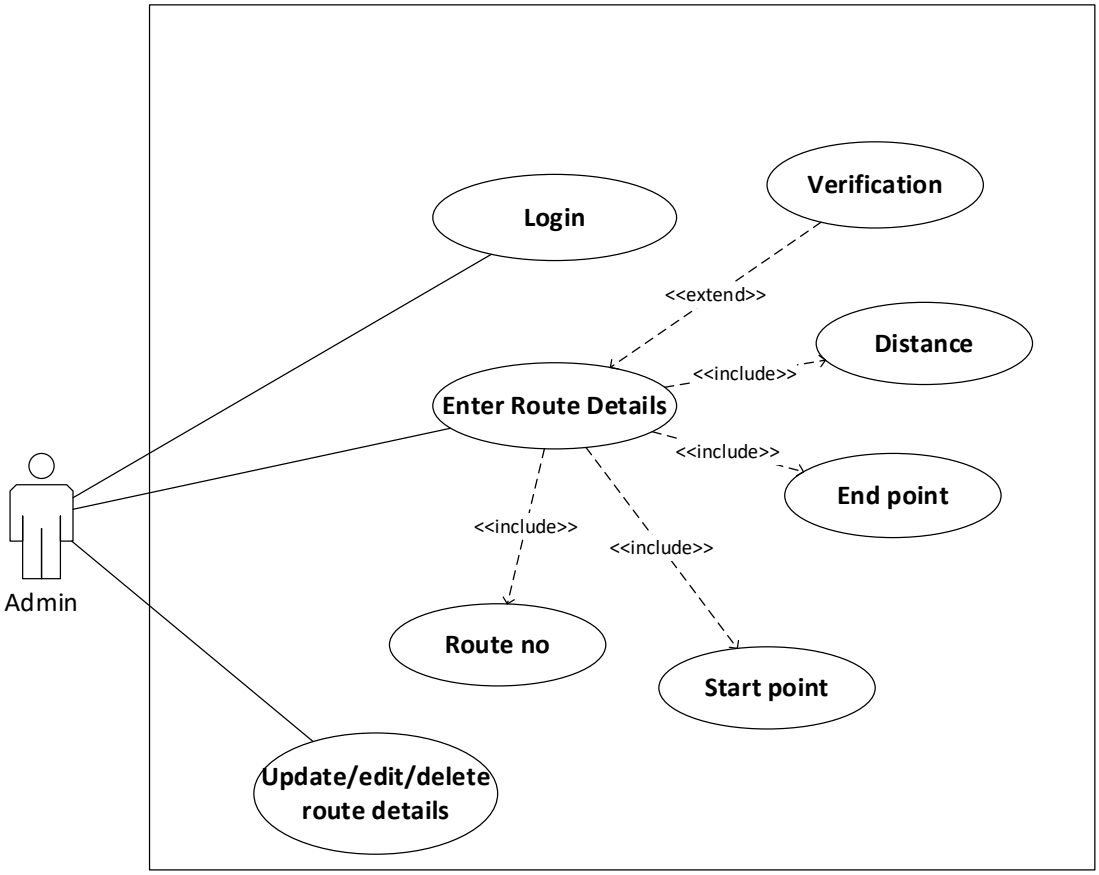



*Figure 3-6: Use case diagram- Route management*

*Table 3-13: Use case scenario- Route management*

| **Use Case ID:** | 004 | | |
|---|---|---|---|
| **Use Case Name:** | Route management | | |
| **Created By:** | Lahiru | Last updated by: | Lahiru |
| **Date Created:** | 2023/03/04 | Date last updated: | 2023/03/04 |

| **Actor:** | Admin |
|---|---|
| **Preconditions:** | Admin already had registered |
| **Post conditions:** | Maintain route details |
| **Main Success Scenario:** | 1. Admin login into the system<br>2. Enter route no<br>3. Enter start point<br>4. Enter end point<br>5. Enter distance<br>6. Click enter<br>7. Authentication (details)<br>8. Details store database<br>9. Update, edit, or delete route records |
| **Extensions:** | 1a. Invalid user<br>1a1. Prompt an error message<br>1a2. Re-enter correct user id<br>1a3. Re-enter correct password<br>8a. Details are saved<br>8a1. Prompt display as a message |

3.5.1.6 *Schedule management*

This diagram shows the schedule management function. This includes all of the schedule details. Administrators can enter those details into the system. It includes such things as route number, date, time, distance, fee, etc. Passengers can view these details.

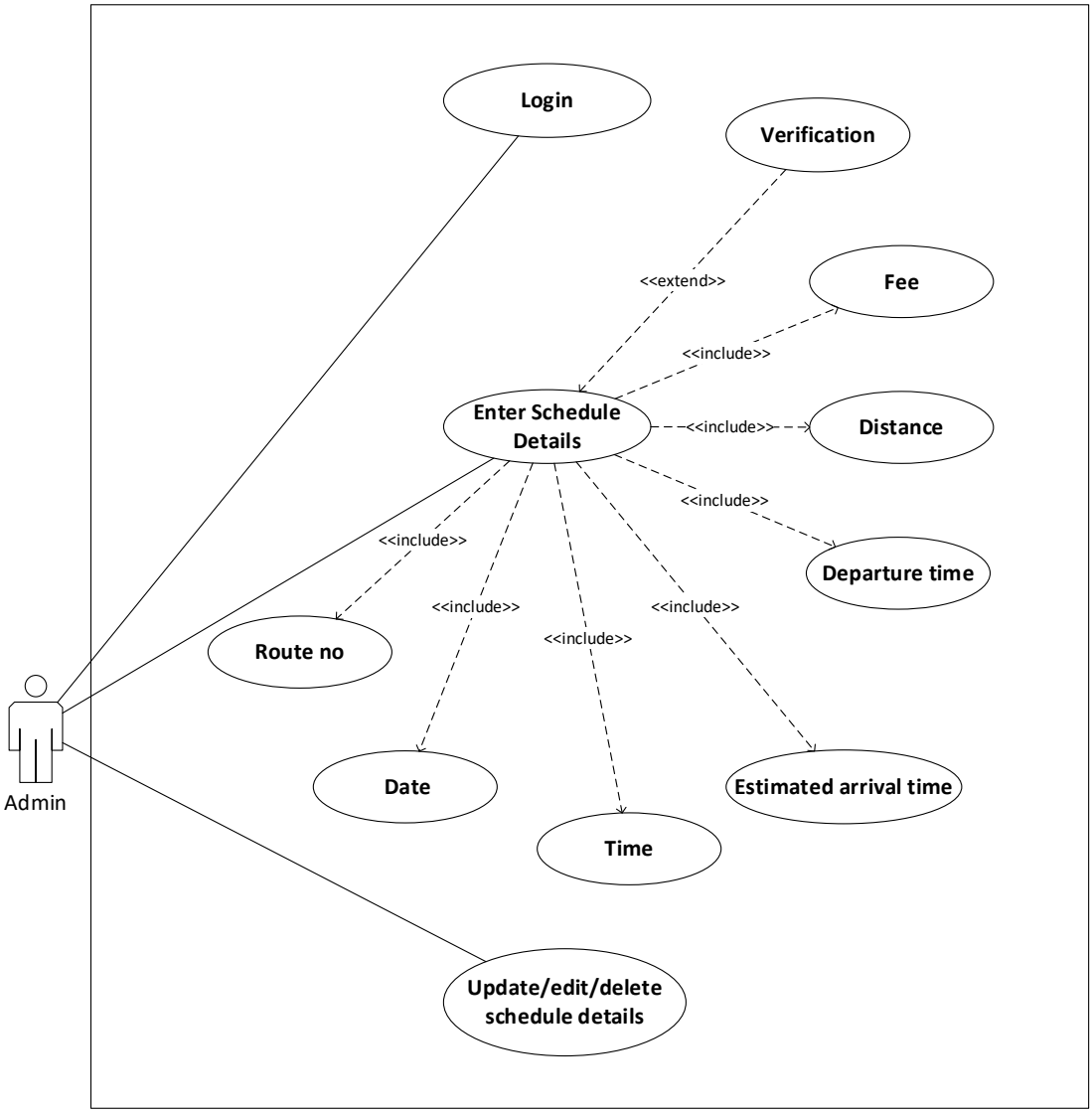



*Figure 3-7: Use case diagram- Schedule management*

*Table 3-14: Use case scenario- Schedule management*

| **Use Case ID:** | 005 | | |
|---|---|---|---|
| **Use Case Name:** | Schedule management | | |
| **Created By:** | Lahiru | Last Updated By: | Lahiru |
| **Date Created:** | 2023/03/04 | Date Last Updated: | 2023/03/04 |

| **Actor:** | Admin |
|---|---|
| **Preconditions:** | Admin already had registered |
| **Post conditions:** | Admin can update and edit the day-to-day bus schedule |
| **Main Success Scenario:** | **Admin**<br>1. Admin login into the System<br>2. Enter route no<br>3. Enter date<br>4. Enter time<br>5. Enter estimated arrival time<br>6. Enter departure time<br>7. Enter distance<br>8. Enter fee<br>9. Click enter<br>10. Authentication (Details)<br>11. Details store Database<br>12. Update, edit, or delete schedule records |
| **Extensions:** | 1a. Invalid user<br>1a1. Prompt an error message<br>1a2. Re-enter correct user id<br>1a3. Re-enter correct password<br>12a. Details are saved<br>12a1. Prompt display as a message |

3.5.1.7 *Booking ticket*

This diagram shows the ticket booking function. This includes all steps of booking tickets sequentially. Here are the tasks for passengers and administrators: Both passengers and administrators should be involved in ticket booking. Passengers have to enter the route details, date and time, and seat details into the system and pay for the reserved seats. Additionally, booked tickets can be canceled here. Administrators update and edit passenger records.

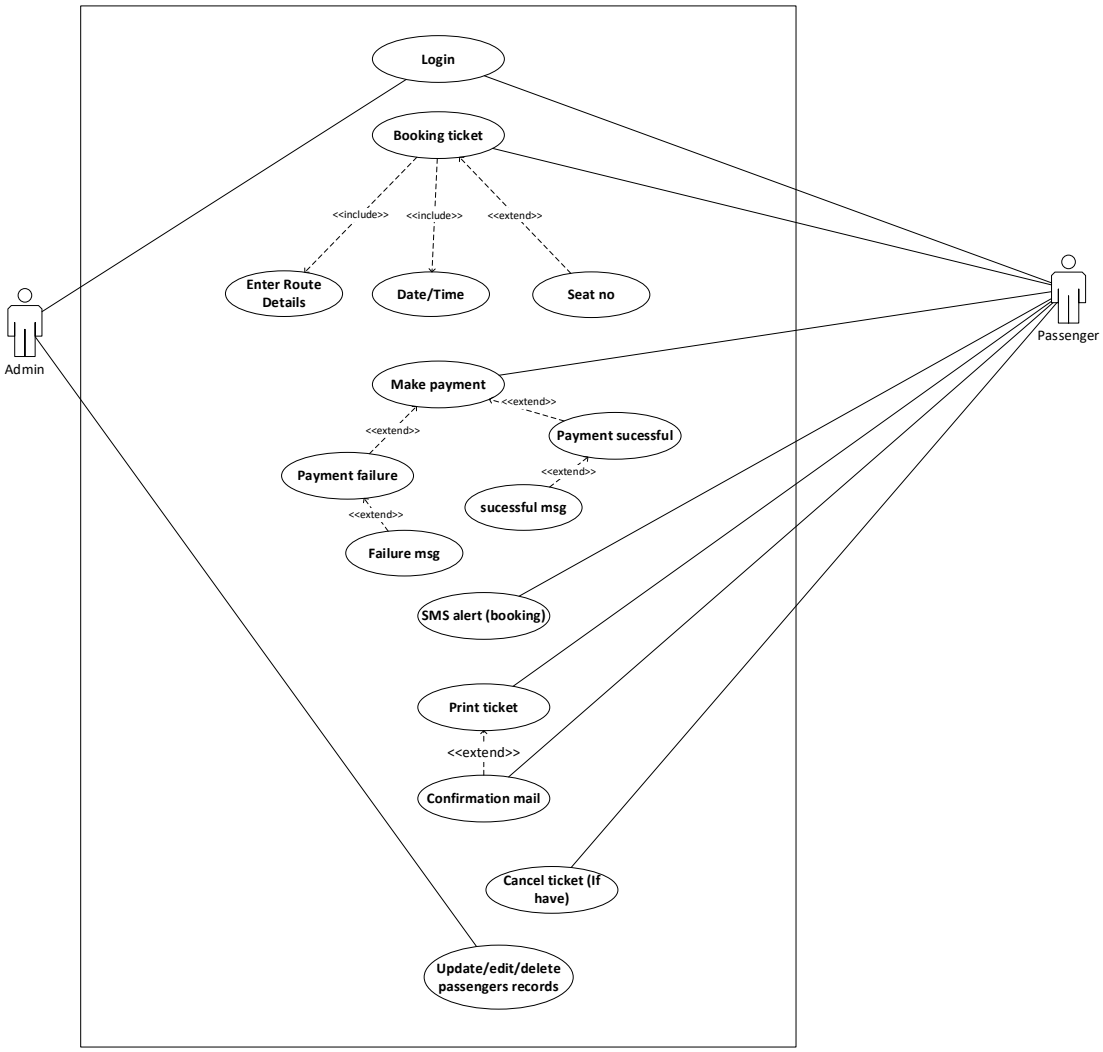



*Figure 3-8: Use case diagram- Booking ticket*

*Table 3-15: Use case scenario- Booking ticket*

| **Use Case ID:** | 006 | | |
|---|---|---|---|
| **Use Case Name:** | Passenger booking ticket | | |
| **Created By:** | Lahiru | Last Updated By: | Lahiru |
| **Date Created:** | 2023/03/04 | Date Last Updated: | 2023/03/04 |

| **Actor:** | Passengers, admin |
|---|---|
| **Preconditions:** | The passenger should access the website<br>The passenger should possess a device to access the website<br>The passenger had already registered<br>Seats are available<br>Admin already had a registered |
| **Post conditions:** | Passengers booking seats (reserving tickets)<br>Admin maintains booking details<br>Admin maintains ticket details |
| **Main Success Scenario:** | **Passenger**<br>1. Passenger login into the System<br>2. Enter route Details<br>3. Enter date/time<br>4. Enter seats<br>5. Click booking<br>6. Make payment<br>7. Booking confirmation msg (SMS alert)<br>8. Cancel ticket (if have)<br>9. Details save database<br>10. Confirmation mail<br>**Admin**<br>11. Update, edit, or delete passenger records<br>12. Generate Reports<br>13. Booking Management<br>14. Ticket Management |
| **Extensions:** | 6a. Your card is invalid<br>6a1. Prompt an error message<br>6a2. Please insert correct card details<br>6b. Your card is expired<br>6b1. Prompt an error message<br>6b2. Please select other payment method |

### 3.5.1.8 *Feedback management*

This diagram shows the feedback function of this system. Any system has an inquiry and feedback function for communication between passengers and administrators. This system also provides the same function. Passengers can send messages to the administrator. Admins can reply to passenger messages immediately. Finally, the data is saved in the database.

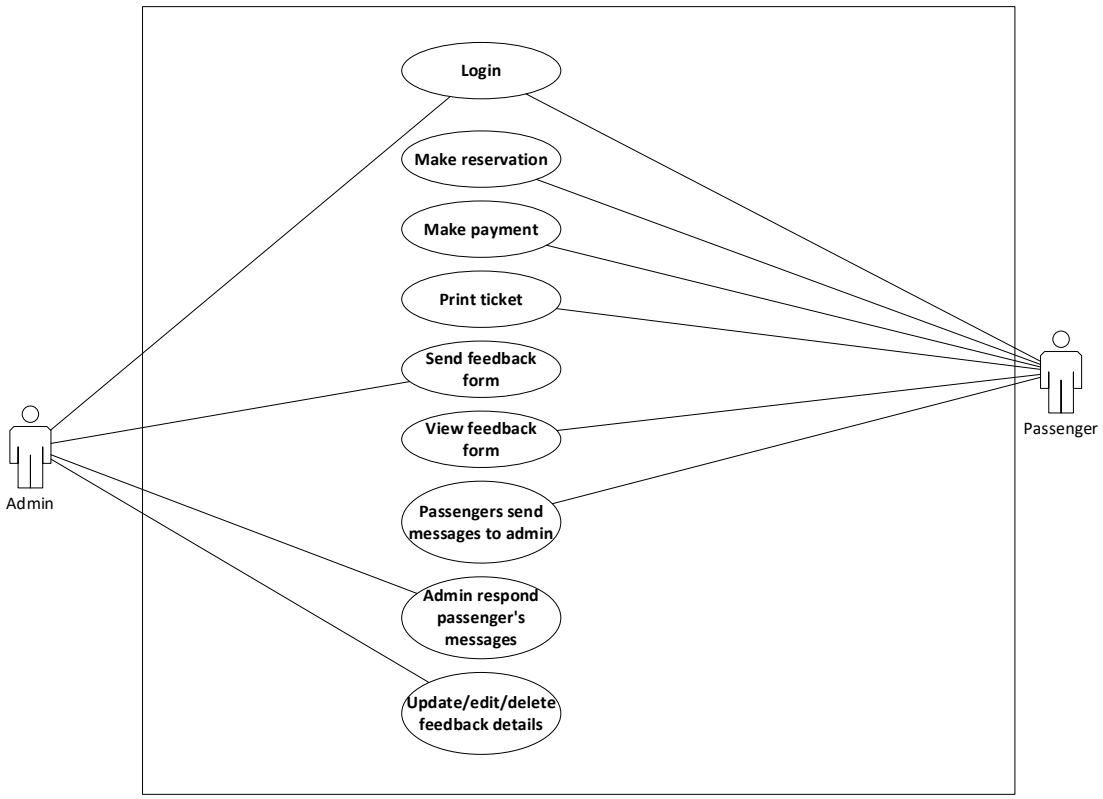



*Figure 3-9: Use case diagram- Feedback management*

*Table 3-16: Use case scenario- Feedback management*

| | | | |
|---|---|---|---|
| **Use Case ID:** | 007 | | |
| **Use Case Name:** | Feedback management | | |
| **Created By:** | Lahiru | Last Updated By: | Lahiru |
| **Date Created:** | 2023/03/04 | Date Last Updated: | 2023/03/04 |

| | |
|---|---|
| **Actor:** | Admin, passenger |
| **Preconditions:** | The passenger should access the website<br>The passenger should possess a device to access the website<br>Passenger already had registered<br>The Passenger already had a reserved ticket<br>The Passenger already had a received booking confirmation mail or message<br>Admin already had a registered |
| **Post conditions:** | The passengers send messages to admin, and admin quickly responds to them |
| **Main Success Scenario:** | **Passenger**<br>1. Passengers login into the system<br>2. Passengers can send messages or inquiries to admin<br>**Admin**<br>3. Admin login to the system<br>4. Admin responds to passengers' messages<br>5. Update, edit, or delete passenger records |
| **Extensions:** | 2a. We have received your message. We will respond it soon<br>2a1. Prompt display as a message |

### 3.5.2 Sequence diagram

To understand what a sequence diagram is, it's important to know the role of the Unified Modeling Language, better known as UML. UML is a modeling toolkit that guides the creation and notation of many types of diagrams, including behavior diagrams, interaction diagrams, and structure diagrams. A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects work together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios (Anon., n.d.).

3.5.2.1 *Sequence diagram of User login*

This diagram shows the user login function. It includes both administrators and passengers. Here, users need to enter usernames and passwords and click the login button on the login page. Then the system will verify (authenticate) the user's information.

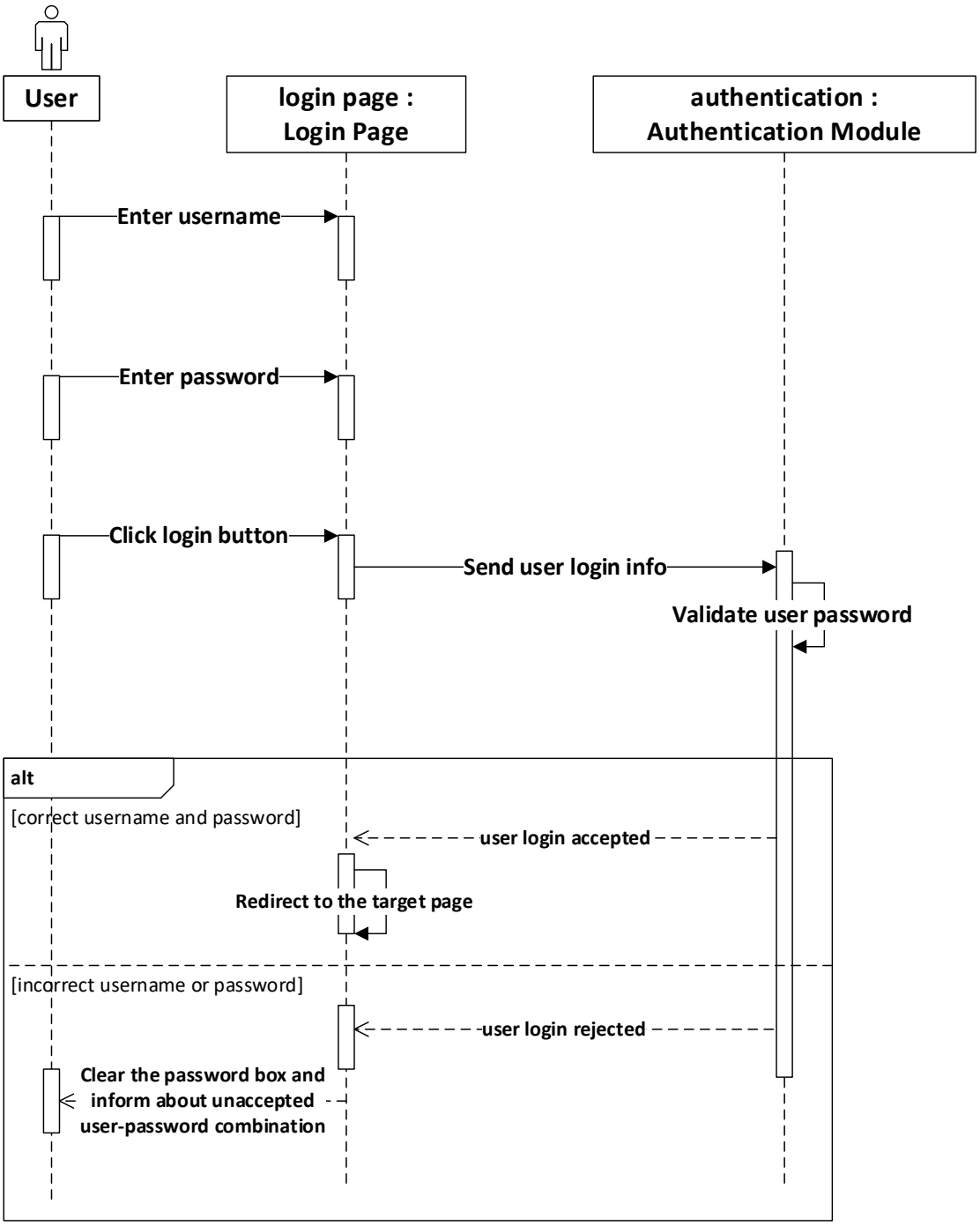



*Figure 3-10: Sequence diagram- User login*

### 3.5.2.2 *Sequence diagram of Admin functions*

This diagram shows all the admin functions like system user management, customer (passenger) management, bus management, route management, timetable management, etc. Administrators can always add or edit records, save or update records, list records, or delete records.

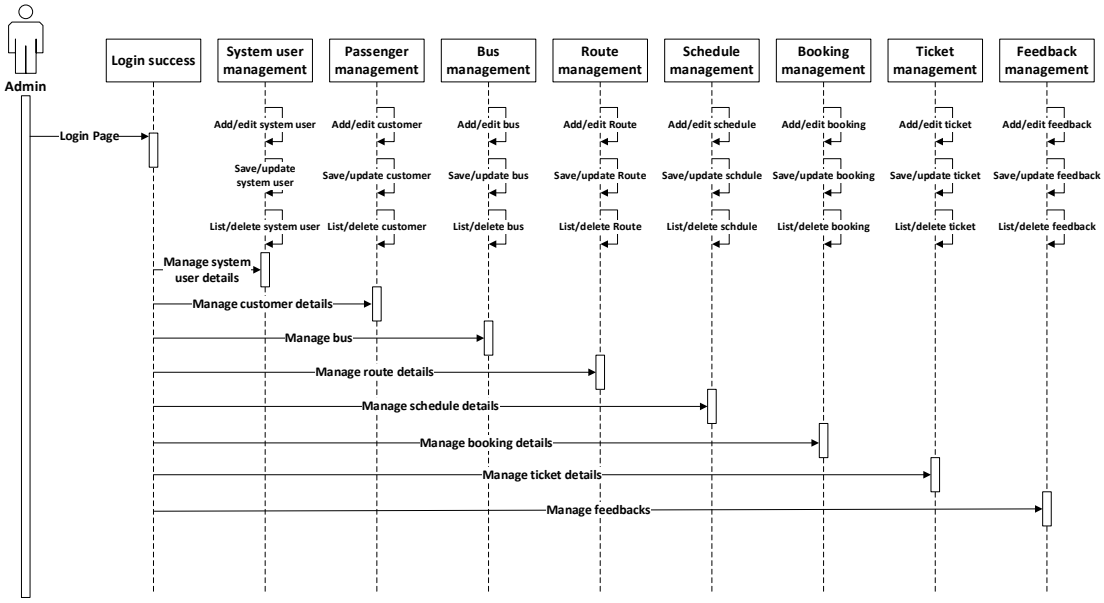



*Figure 3-11: Sequence diagram- Admin functions*

### 3.5.2.3 *Sequence diagram of Passenger functions*

This diagram shows the passenger tasks step by step. Passengers must enter route information, dates and times, and seats. Then, the database will check the seats of the buses on the selected route. In respective buses, passengers can book tickets if there are enough seats available for booking. Finally, passengers can see the ticket details.

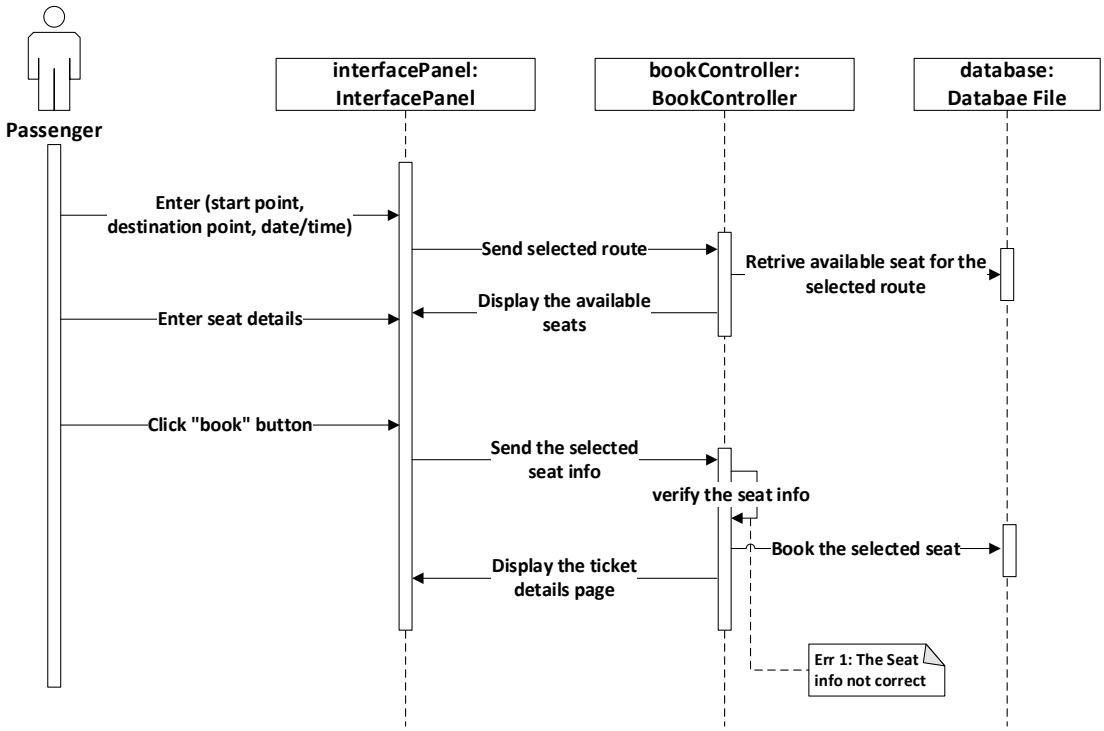



*Figure 3-12: Sequence diagram- Passenger functions*

### 3.5.3 Activity diagram

An activity diagram is another important diagram in UML to describe the dynamic aspects of the system. An activity diagram is basically a flowchart to represent the flow from one activity to another. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all types of flow control by using different elements such as fork, join, etc (Anon., n.d.).

#### 3.5.3.1 *Activity diagram of Admin functions*

This diagram shows the system administrator's functions. First, the administrator must enter the login ID and password on the login page. The system then verifies the information. If the correct login ID and password are entered, the administrator can access the system and perform functions such as user management, passenger management, bus management, route management, schedule management, etc. Otherwise, the system cannot be accessed and returns to the login page.

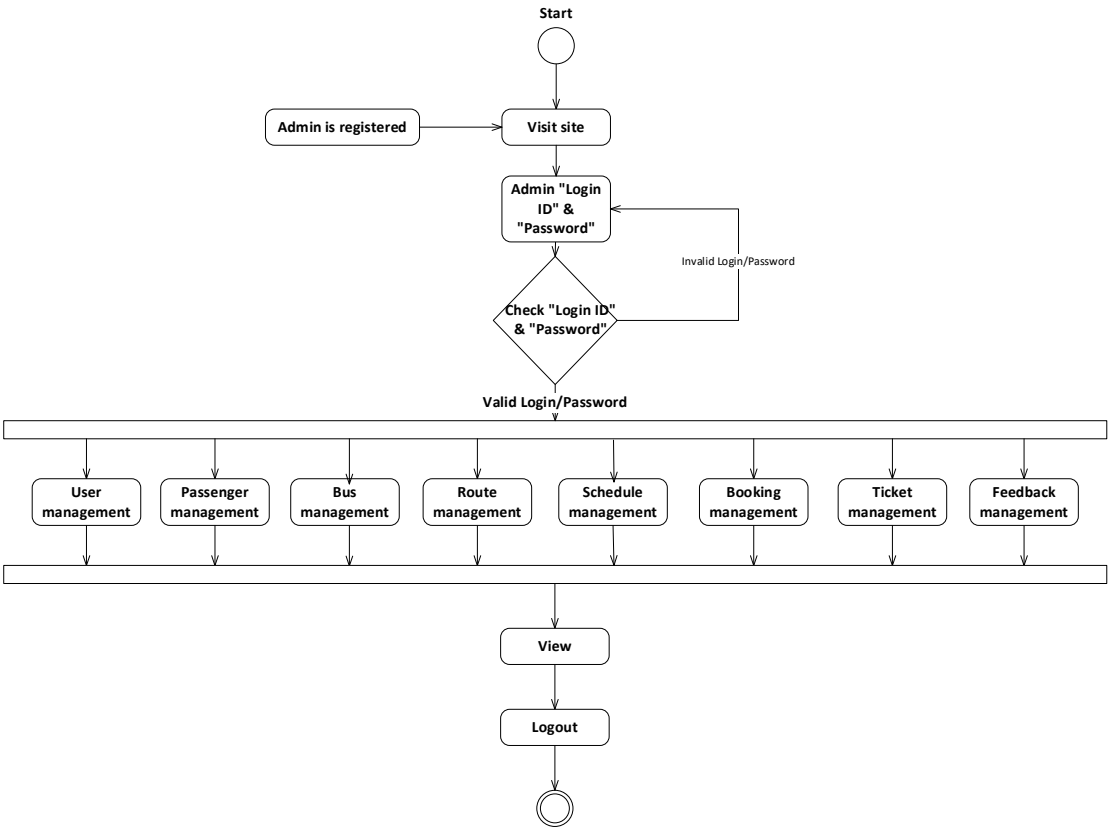



*Figure 3-13: Activity diagram- Admin functions*

### 3.5.3.2 *Activity Diagram of Passenger functions*

This diagram shows passenger functions. If the passenger does not have an account, they must first register with the system. After logging in to the system, passengers can view timetables, book tickets, pay, and view tickets.

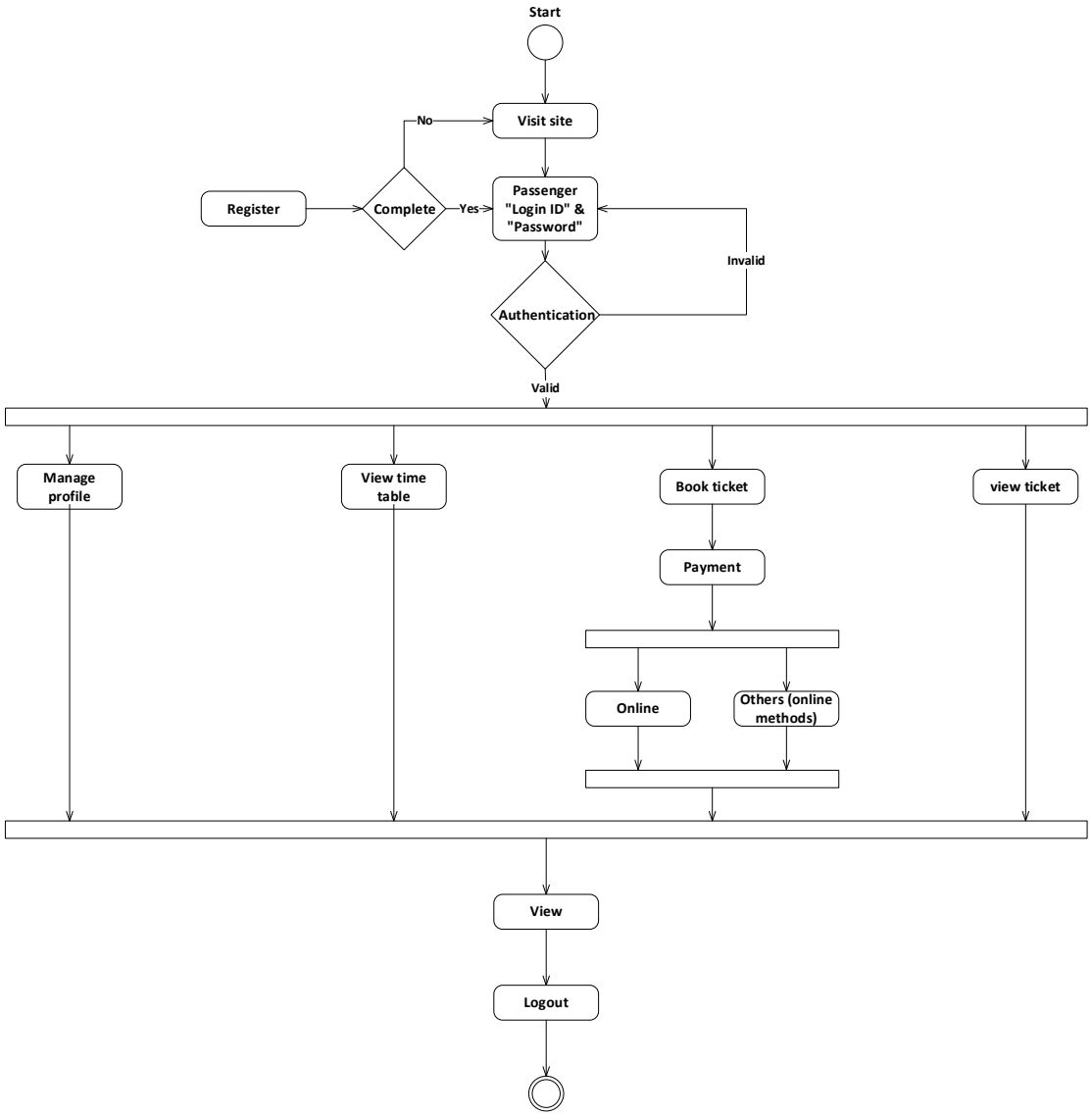



*Figure 3-14: Activity diagram- Passenger functions*

### 3.5.4 ER diagram

An entity relationship (ER) diagram is a type of flowchart that illustrates how "entities" such as people, objects, or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education, and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals, and connecting lines to depict the interconnectedness of entities, relationships, and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs (Anon., n.d.).

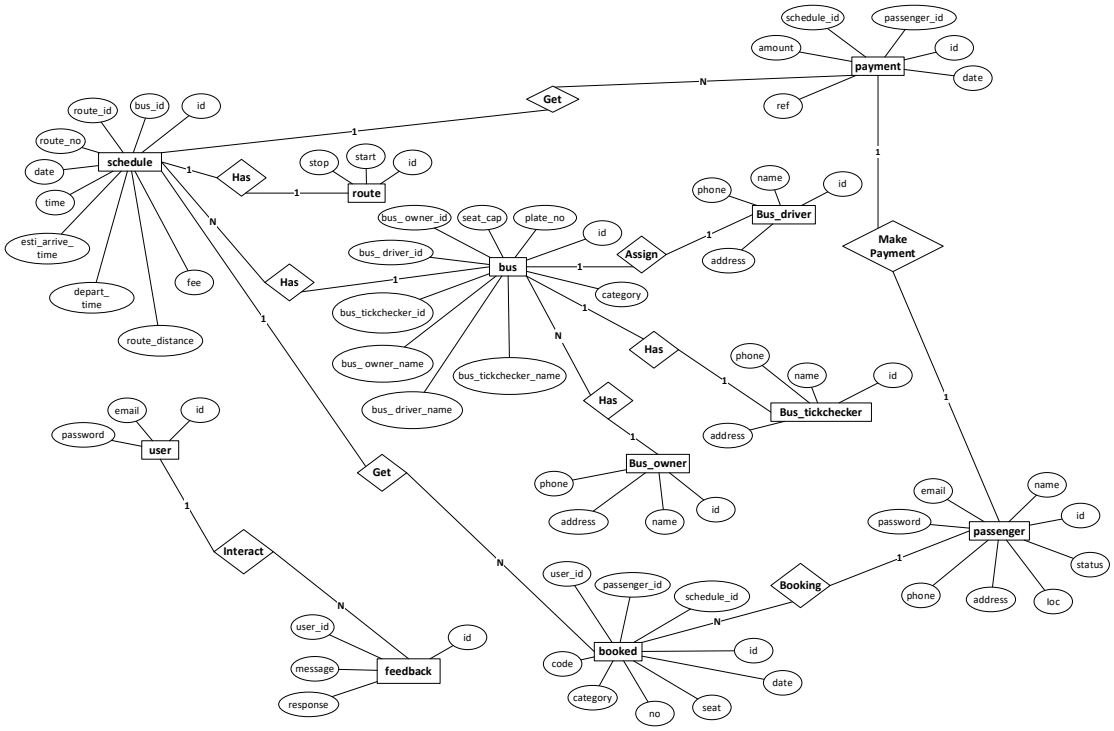



*Figure 3-15: ER diagram*

### 3.5.5 Class diagram

A class diagram is a static diagram. It represents the static view of an application. A class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code for the software application. A class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams that can be mapped directly to object-oriented languages. A class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram (Anon., n.d.).

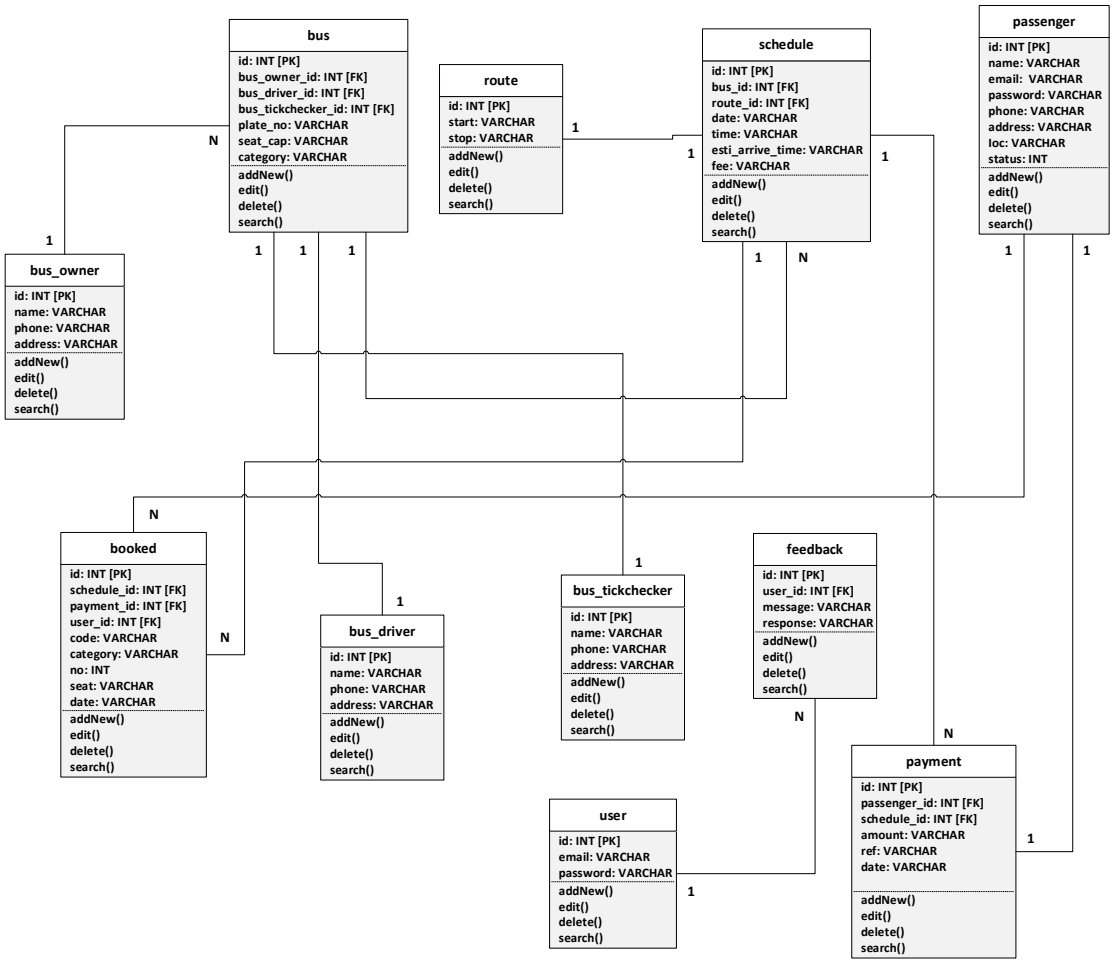



*Figure 3-16: Class diagram*

### 3.5.6 Relational database schema

A relational database program is a standard tool for storing and analyzing data. Examples include Microsoft SQL Server, PostgreSQL, MySQL, and various products from companies such as Oracle and IBM. A relational schema outlines the database relationships and structure in a relational database program. It can be displayed graphically or written in the Structured Query Language (SQL) used to build tables in a relational database (Anon., n.d.).

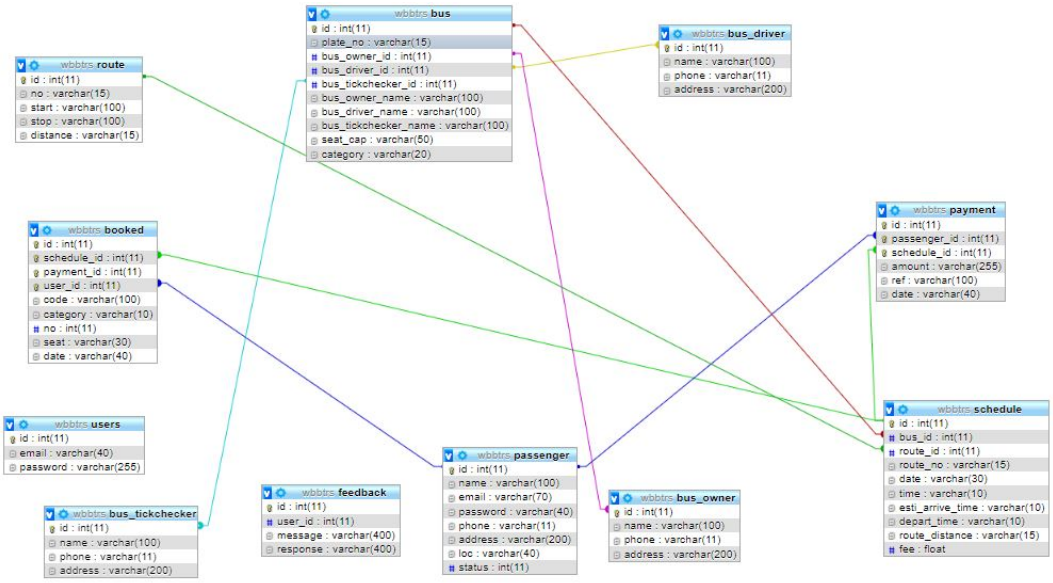



*Figure 3-17: Relational database schema*

### 3.5.7 Data tables

A data table is a document comprising columns, rows, and cells that contain specific values. They store information that people can retrieve later and update as needed. The data table title, column headers, and row headers can help a user understand the information in the table more clearly. The document also processes the data through various operations, such as ordering, arranging, filtering, and searching. Most data tables provide insight into sets of information. For example, someone running a children's summer camp might use a data table to track the names, arrival times, registration status, and contact information of the attending families. This allows them to sort the most important data from their records and keep it separate from irrelevant details. Storing all this information in one place can make it easier to access and can also help those analyzing the data develop meaningful conclusions about a table's contents (Anon., n.d.).

*Table 3-17: booked table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| schedule_id | int(11) | No | | |
| payment_id | int(11) | No | | |
| user_id | int(11) | No | | |
| code | varchar(100) | No | | |
| category | varchar(10) | No | second | |
| no | int(11) | No | 1 | |
| seat | varchar(30) | No | | |
| date | varchar(40) | No | | |

*Table 3-18: booked indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 0 | A | No | |
| schedule_id | BTREE | Yes | No | schedule_id | 0 | A | No | |
| | | | | user_id | 0 | A | No | |
| | | | | payment_id | 0 | A | No | |
| schedule_id_2 | BTREE | No | No | schedule_id | 0 | A | No | |
| user_id | BTREE | No | No | user_id | 0 | A | No | |

*Table 3-19: bus table*

| Column | Type | Null | Comments |
|---|---|---|---|
| id *(Primary)* | int(11) | No | |
| plate_no | varchar(15) | No | |
| bus_owner_id | int(11) | No | |
| bus_driver_id | int(11) | No | |
| bus_tickchecker_id | int(11) | No | |
| bus_owner_name | varchar(100) | No | |
| bus_driver_name | varchar(100) | No | |
| bus_tickchecker_name | varchar(100) | No | |
| seat_cap | varchar(50) | No | |
| category | varchar(20) | No | |

*Table 3-20: bus indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 2 | A | No | |
| bus_owner_id | BTREE | No | No | bus_owner_id | 2 | A | No | |
| bus_driver_id | BTREE | No | No | bus_driver_id | 2 | A | No | |
| bus_tickchecker_id | BTREE | No | No | bus_tickchecker_id | 2 | A | No | |

*Table 3-21: bus_driver table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| name | varchar(100) | No | | |
| phone | varchar(11) | No | | |
| address | varchar(200) | No | | |

*Table 3-22: bus_driver indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 0 | A | No | |

*Table 3-23: bus_owner table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| name | varchar(100) | No | | |
| phone | varchar(11) | No | | |
| address | varchar(200) | No | | |

*Table 3-24: bus_owner indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 0 | A | No | |

*Table 3-25: bus_tickchecker table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| name | varchar(100) | No | | |
| phone | varchar(11) | No | | |
| address | varchar(200) | No | | |

*Table 3-26: bus_tickchecker indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 0 | A | No | |

*Table 3-27: feedback table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| user_id | int(11) | No | | |
| message | varchar(400) | No | | |
| response | varchar(400) | Yes | NULL | |

*Table 3-28:feedback indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 0 | A | No | |
| user_id | BTREE | No | No | user_id | 0 | A | No | |

*Table 3-29: passenger table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| name | varchar(100) | No | | |
| email | varchar(70) | No | | |
| password | varchar(40) | No | | |
| phone | varchar(11) | No | | |
| address | varchar(200) | No | | |
| loc | varchar(40) | No | | |
| status | int(11) | No | 1 | |

*Table 3-30: passenger indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 4 | A | No | |

*Table 3-31: payment table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| passenger_id | int(11) | No | | |
| schedule_id | int(11) | No | | |
| amount | varchar(255) | No | | |
| ref | varchar(100) | No | | |
| date | varchar(40) | No | | |

*Table 3-32: payment indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 0 | A | No | |
| passenger_id | BTREE | Yes | No | passenger_id | 0 | A | No | |
| | | | | schedule_id | 0 | A | No | |
| passenger_id_2 | BTREE | No | No | passenger_id | 0 | A | No | |
| schedule_id | BTREE | No | No | schedule_id | 0 | A | No | |

*Table 3-33: route table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| no | varchar(15) | No | | |
| start | varchar(100) | No | | |
| stop | varchar(100) | No | | |
| distance | int(11) | No | | |

*Table 3-34: route indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 4 | A | No | |

*Table 3-35: schedule table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| bus_id | int(11) | Yes | NULL | |
| route_id | int(11) | Yes | NULL | |
| route_no | varchar(15) | Yes | NULL | |
| date | varchar(30) | Yes | NULL | |
| time | varchar(10) | Yes | NULL | |
| esti_arrive_time | varchar(10) | Yes | NULL | |
| depart_time | varchar(10) | Yes | NULL | |
| route_distance | varchar(15) | Yes | NULL | |
| fee | float | Yes | NULL | |

*Table 3-36: schedule indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 8 | A | No | |
| route_id | BTREE | No | No | route_id | 8 | A | Yes | |
| bus_id | BTREE | No | No | bus_id | 4 | A | Yes | |

*Table 3-37: users table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| email | varchar(40) | No | | |
| password | varchar(255) | No | | |

*Table 3-38: users indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 0 | A | No | |

## 3.6 User interfaces

The user interface is the front-end application view with which the user interacts in order to use the software. The user can manipulate and control the software as well as the hardware by means of the user interface. Today, user interfaces are found at almost every place where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships, etc. The user interface is part of the software and is designed in such a way that it is expected to provide the user with insight into the software. UI provides the fundamental platform for human-computer interaction.

UI can be graphical, text-based, or audio-video-based, depending upon the underlying hardware and software combination. UI can be hardware, software, or a combination of both (Anon., n.d.).

### 3.6.1 Frontend interfaces

These are front-end GUIs. Passengers and administrators can login to the system through the front end.

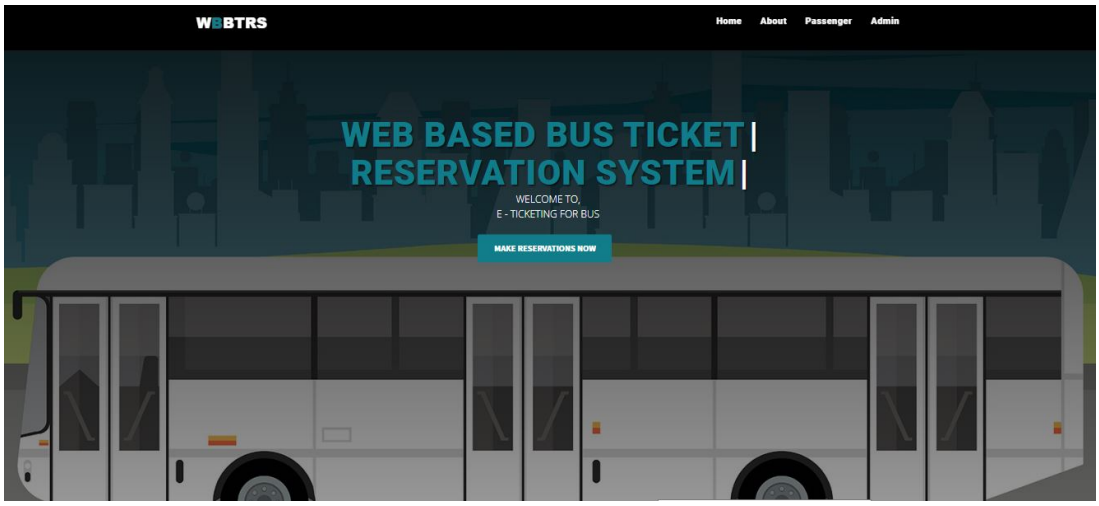



*Figure 3-18: Frond-end 1*

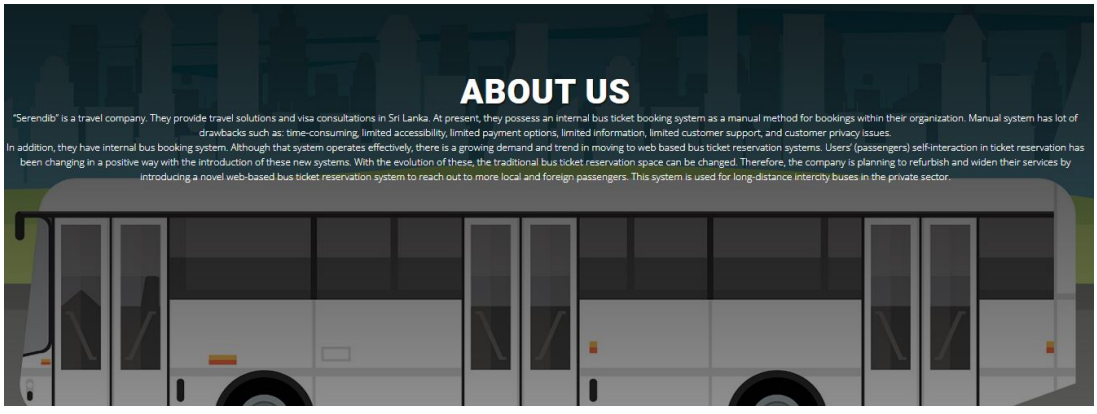



*Figure 3-19: Frond end 2*

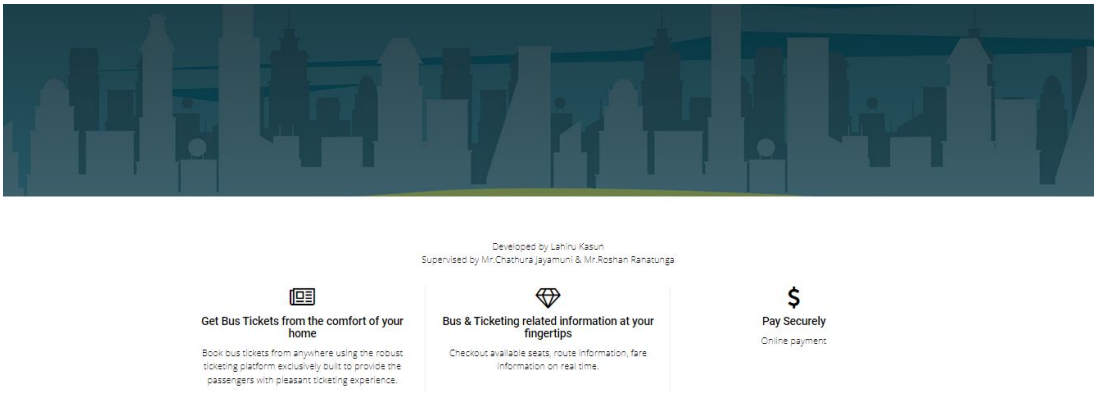



*Figure 3-20: Frond end 3*

### 3.6.1.1 *Admin sign-in*

If the administrators enter the correct email and password, they can successfully log into the system.

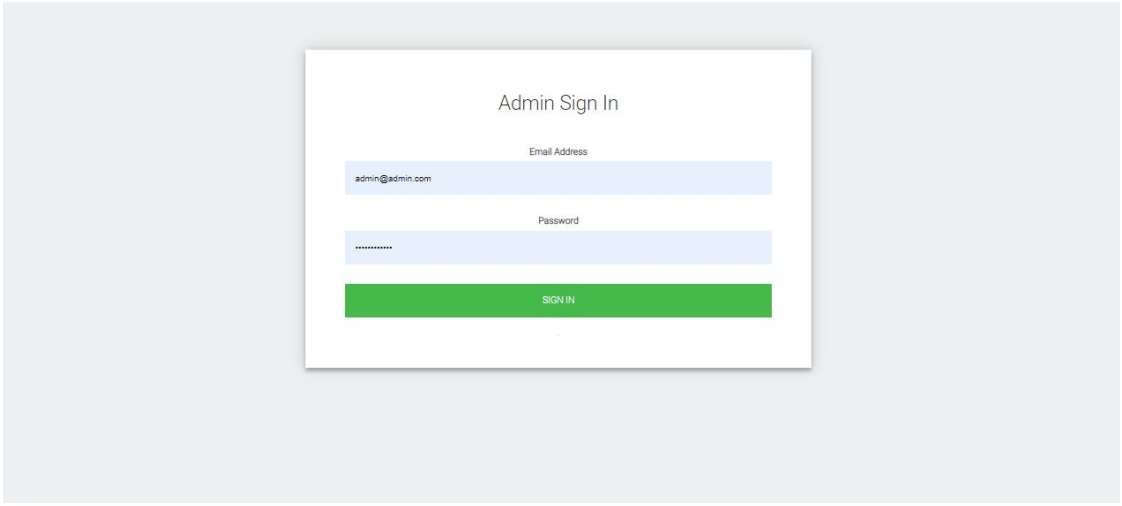



*Figure 3-21: Admin sign- in*

### 3.6.1.2 *Passenger sign-up*

If passengers do not already have accounts, a new one must be created.

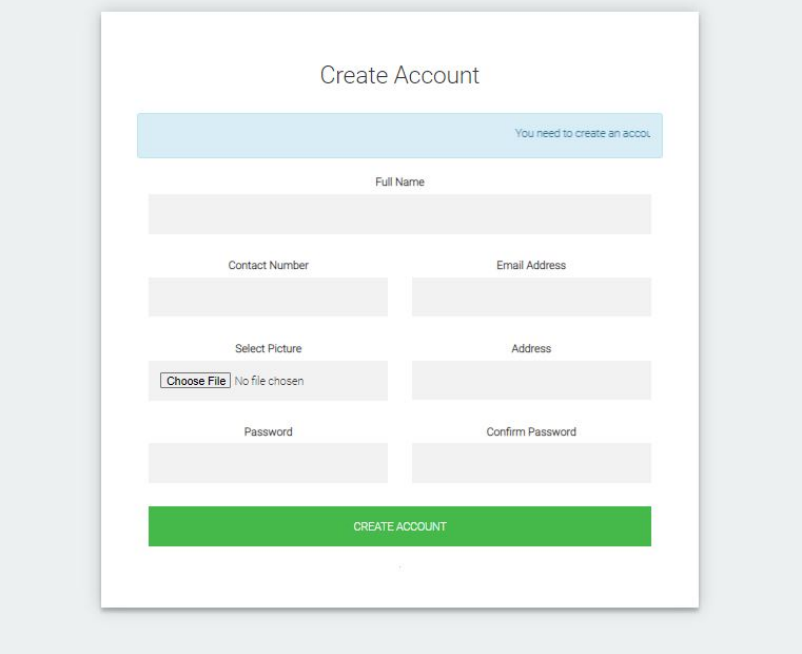



*Figure 3-22: Passenger sign-up*

### 3.6.1.3 *Passenger sign-in*

If passengers do not already have accounts, they must create one before logging into the system. If an existing user can log into the system directly after providing the correct email and password.

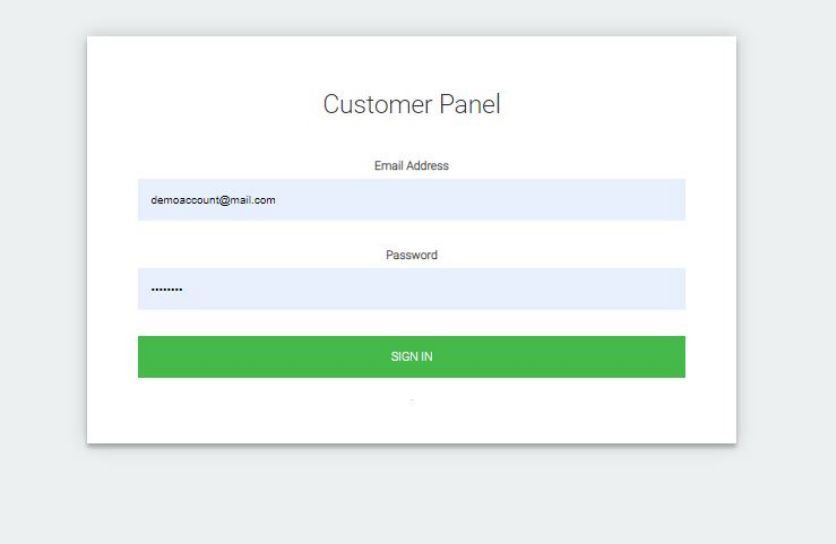



*Figure 3-23: Passenger sign-in*

### 3.6.1.4 *Passenger dashboard*

Passengers can book tickets, cancel tickets, view booking history, send feedback, and also logout through this passenger's dashboard.

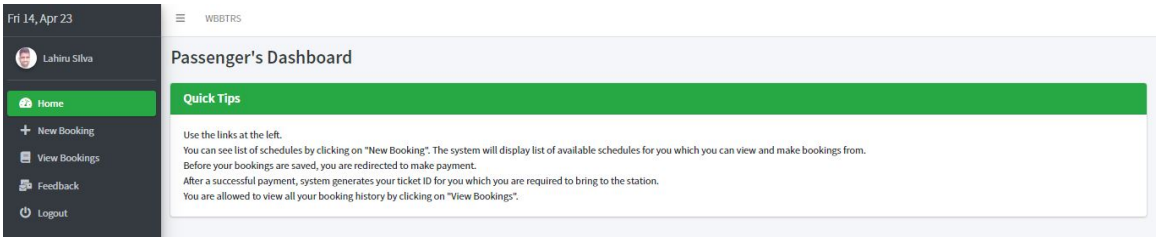



*Figure 3-24: Passenger dashboard*

### 3.6.1.5 *Passenger new booking*

Passengers can book tickets and, if they want, cancel them through this.

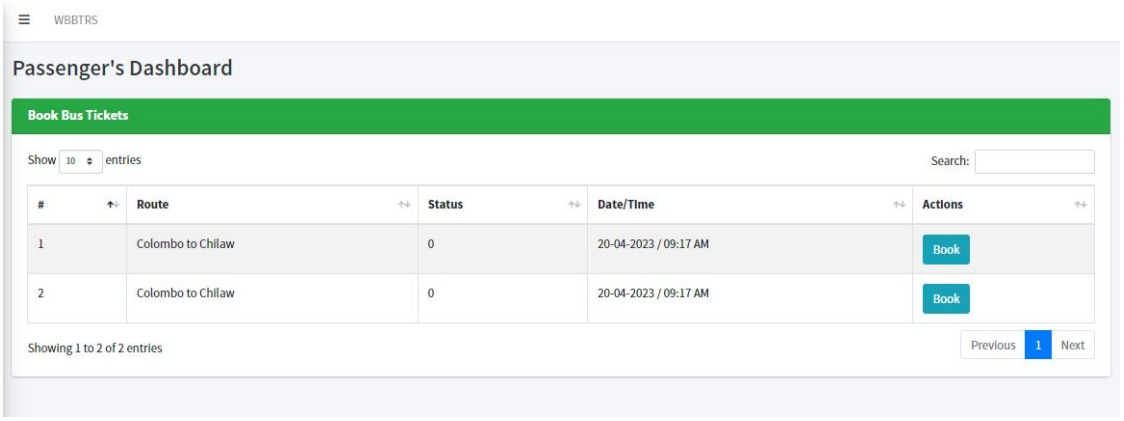



*Figure 3-25: Passenger booking*

### 3.6.1.6 *Booking preview*

The reservation preview page is part of the reservation itself. If you click on the book button, you will see ticket details like route details and fares. You can enter the number of tickets and see the corresponding total price.

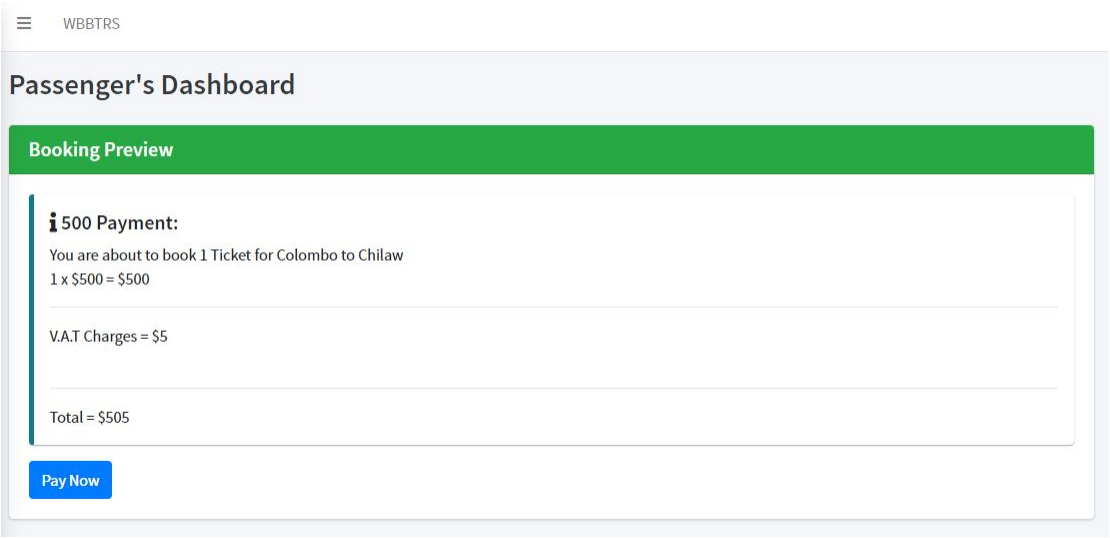



*Figure 3-26: booking preview page*

### 3.6.1.7 *Payment gateway*

After clicking the "Pay Now" button, you can be redirected to the payment gateway page. On this page, you can pay your bus fare online. For that, you can use your credit or debit card. After payment, you will receive a confirmation email as well as an SMS message.

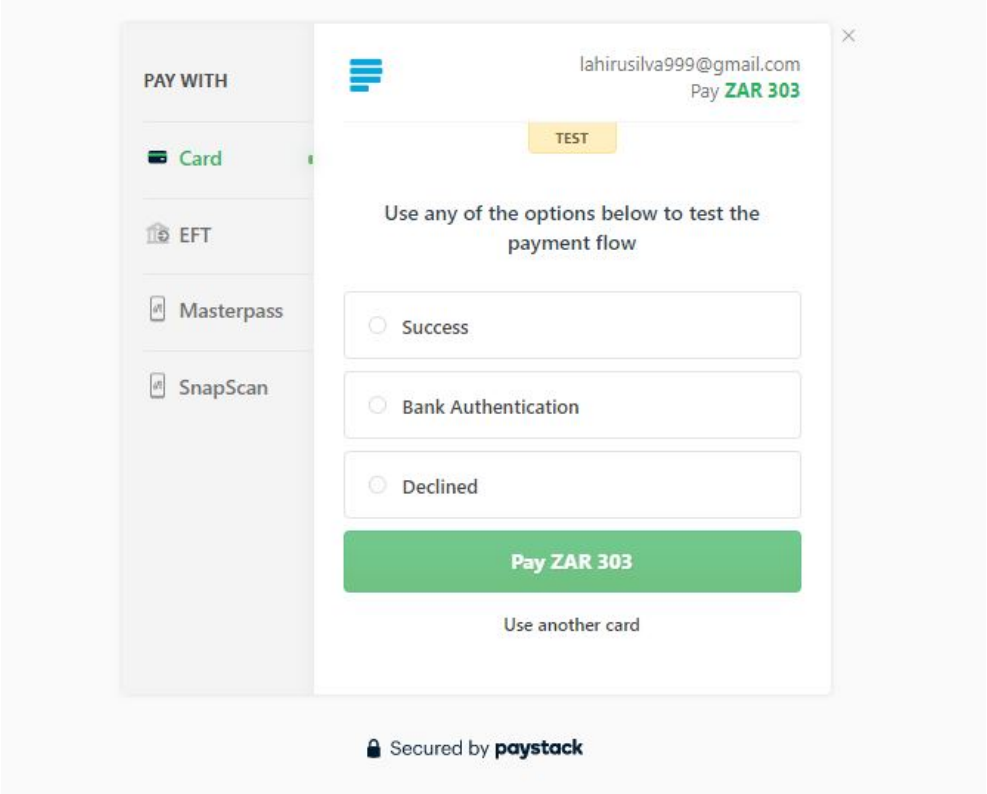



*Figure 3-27: Payment gateway page*

### 3.6.2 Backend interfaces

These are back-end GUIs. Only administrators can access and login to the backend. The administrators can manage all functions through this.

#### 3.6.2.1 *Admin dashboard*

This is the administrator dashboard, or back end. Administrators can add, edit, update, and delete all records in this dashboard, such as bus, users, route, schedule, and others.

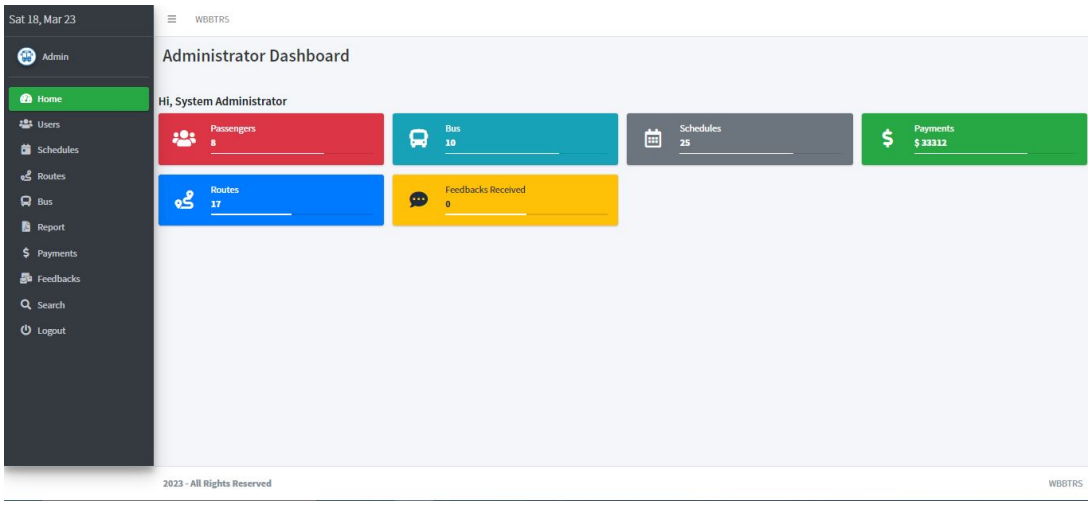



*Figure 3-28: Admin dashboard*

#### 3.6.2.2 *User management*

Administrators can disable registered users at any time through this.

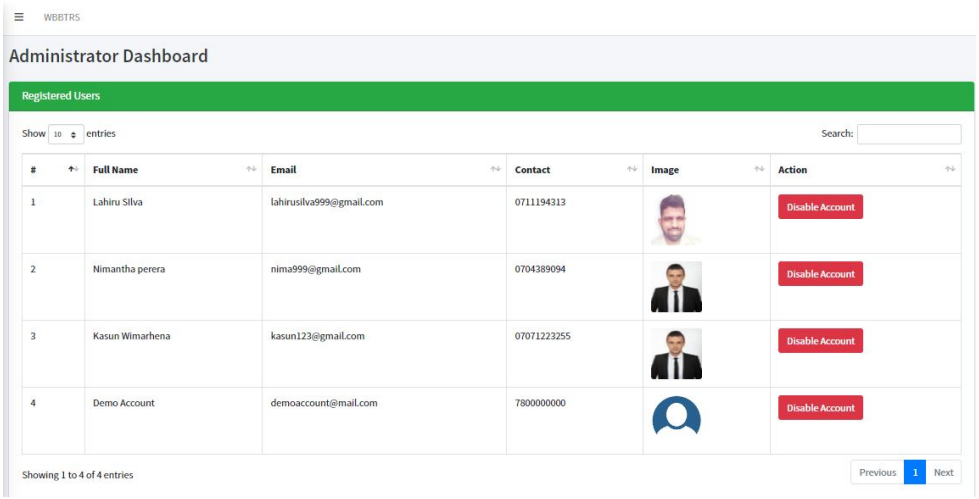



*Figure 3-29: User management*

### 3.6.2.3 *Route management*

Administrators can add, edit, update, and delete route details through this.

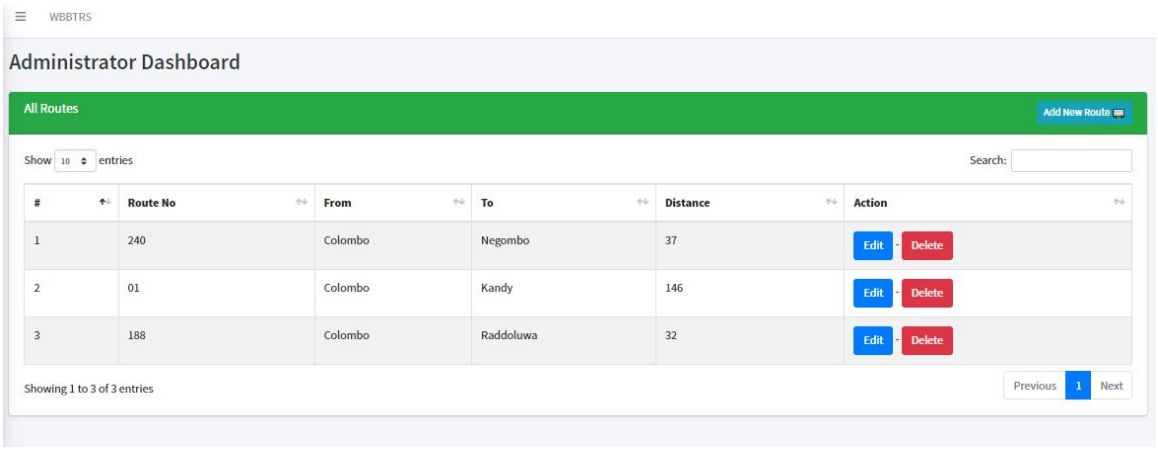



*Figure 3-30: Route management*

### 3.6.2.4 *Schedule management*

Administrators can add, edit, update, and delete schedule details through this.

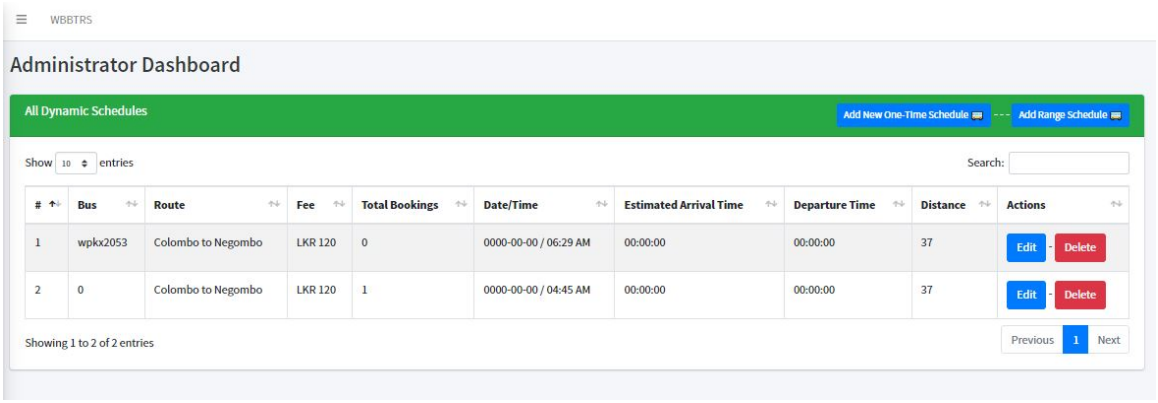



*Figure 3-31: Schedule management*

## 3.7 Summary

This section described all the analysis and design stages of the proposed system by using all UML diagrams, including use case diagrams, activity diagrams, sequence diagrams, class diagrams, and ER diagrams, which are all included here. In addition, relational schema and data tables were also used. Furthermore, this section described all the requirements, specifications, technology adaptations, and user interfaces of the proposed system.

In the next chapter, the implementation phase of the proposed system will be discussed using flow charts. With this, the codes of the proposed system will also be discussed.

# Chapter 4 - Implementation

## 4.1 Introduction

In this chapter, the implementation phase of the proposed system will be discussed. Here the focus is on integrating the backend and dashboard with the database.
This chapter includes an overview of the project and provides flowcharts. A flowchart is a graphical representation of a process that shows the steps involved and the flow of information or materials between them. In the context of implementation, flowcharts help illustrate the various steps involved in the installation, configuration, testing, training, rollout, and maintenance phases, as well as the inputs and outputs of each step. This makes it easier to understand the overall process and identify potential bottlenecks or areas for improvement. Therefore, including a flowchart in the implementation chapter of an IT project can be useful to provide a visual representation of the implementation process.

## 4.2 System architecture

The web-based bus ticket reservation system is a software application that requires both backend and frontend components to work together seamlessly. Backend and frontend are two important parts of a software application, including a web-based bus ticket system.
The frontend is the part of the application that users see and interact with, while the backend is the part behind the scenes that handles data processing and system operations. The frontend communicates with the backend to retrieve data and perform business operations. For example, when a user searches for available buses, the frontend sends a request to the backend, which retrieves the bus schedules and seat availability data from the database and sends it back to the frontend to display the results. Both the frontend and the backend are essential parts of a web-based bus ticket system, and they must work together seamlessly to provide a great user experience.

### 4.2.1 Backend application

The backend application of a web-based bus ticket booking system is responsible for managing the business logic of the system, handling user authentication and authorization, processing payments, managing bus schedules and availability, and handling all database-related operations.

The backend of the application communicates with the database using PHP scripts. PHP is a server-side scripting language that is used to create dynamic web pages. It is commonly used in web development because it is easy to learn, fast, and flexible.

### 4.2.2 Frontend application

The frontend is the part of the application that users interact with. It includes the user interface (UI), which allows users to search for buses, view schedules, select seats, and make reservations. The frontend also includes the design, layout, and appearance of the user interface. The frontend is built using web technologies such as HTML, CSS, JQuery and JavaScript.

### 4.2.3 Other technologies

The web-based bus ticket booking system has also used other technologies. To this, I added data security features for data protection. For that, I used the MD5 hash generator.

In summary, the architecture of the web-based bus ticket reservation system is designed to ensure reliable, fast, and secure communication between the back-end and front-end. The use of MySQL, PHP, HTML, CSS, JQuery and other technologies ensures that the system is user-friendly and functional.

### 4.2.4 Connecting to MySQL database with XAMPP and PHP

XAMPP is a free and open-source cross-platform web server solution stack that includes Apache HTTP Server, MariaDB database, and PHP scripting language. It also includes additional modules such as OpenSSL, phpMyAdmin, and PERL.

XAMPP is commonly used for local development and testing of websites and web applications. It can be installed on various operating systems including Windows, Linux, and macOS. The acronym XAMPP stands for Cross-Platform, Apache, MySQL, PHP, and Perl.

XAMPP is easy to install and configure, and provides a user-friendly interface for managing the server and databases. It is widely used by developers and webmasters for creating and testing websites and web applications on their local machines before deploying them to live servers.

To establish a connection using XAMPP, you would first need to download and install the software on your computer. Once installed, you can start the Apache and MySQL services from the XAMPP control panel.

To start the Apache service, click on the "Start" button next to the "Apache" module in the control panel. This will start the Apache server, which is responsible for serving web pages.

To start the MySQL service, click on the "Start" button next to the "MySQL" module in the control panel. This will start the MySQL database server, which is responsible for managing the data used by your web application. Once both services are started,

you can connect to the MySQL database using the PHP code snippet described in the previous answer. The host name in the connection details should be set to "localhost" since you are connecting to a database running on the same machine as your web server.

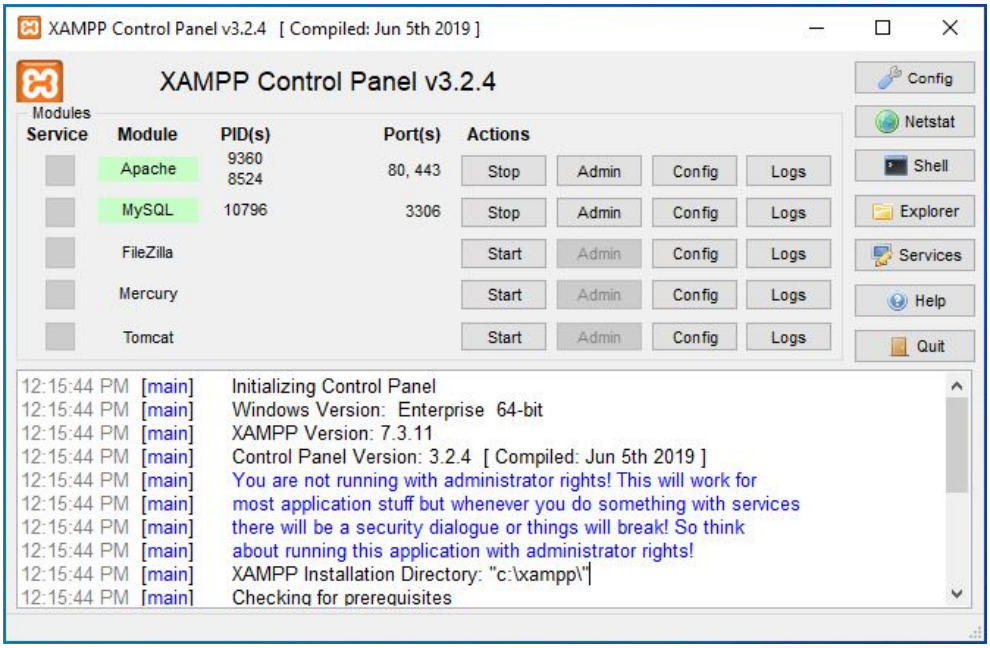



*Figure 4-1: Connecting to MySQL database with XAMPP and PHP*

## 4.3 Flow charts

A flowchart is a graphical representation of steps. It originated in computer science as a tool for representing algorithms and programming logic but has been extended to all other kinds of processes. Nowadays, flowcharts play an extremely important role in displaying information and assisting reasoning. They help us visualize complex processes or make explicit the structure of problems and tasks. A flowchart can also be used to define a process or project to be implemented.

### 4.3.1 Admin login

This is the system administrator's login. The administrator has the main role in this system. The administrators can access the system by providing the correct username and password. After verifying their identities, administrators can login to the system. Returns to the login page if it is no longer valid.

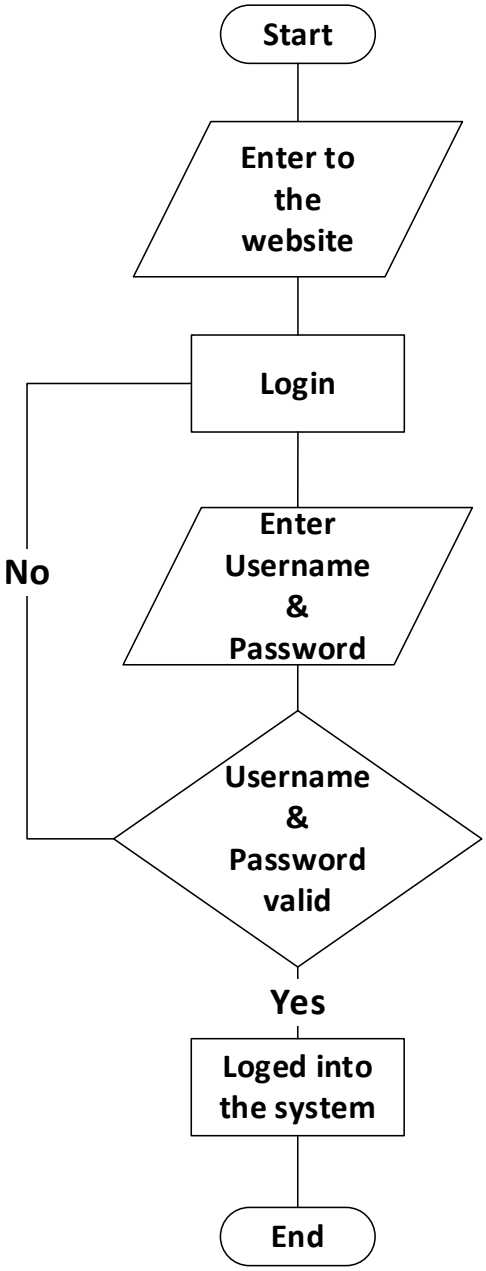



*Figure 4-2: Admin sign-in*

### 4.3.2 Passenger sign-Up

This is the passenger's registration process. If passengers do not already have accounts, they can create one. Passengers must enter and submit details in the system.

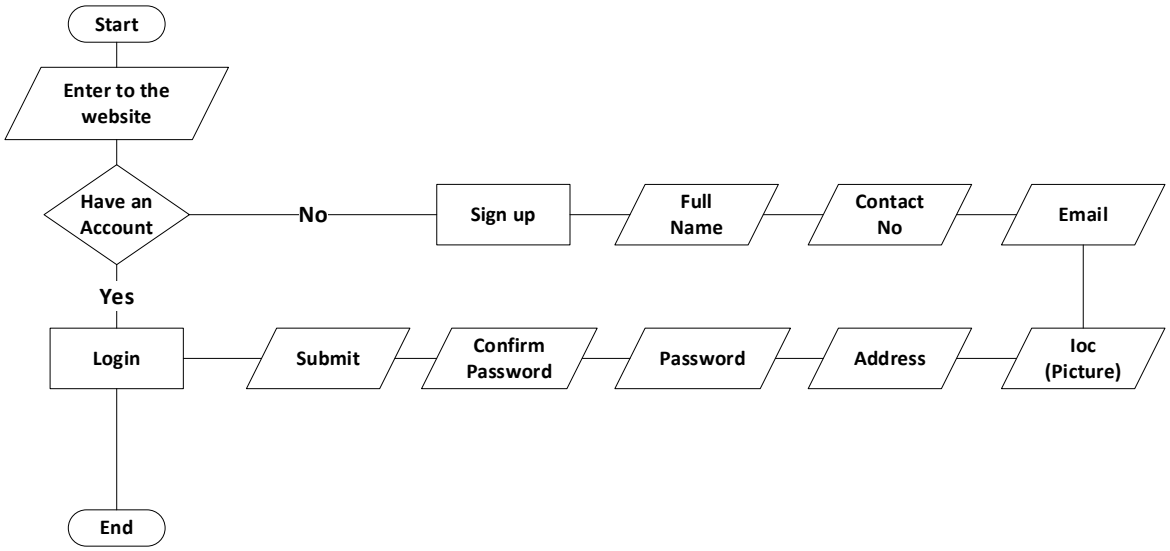



*Figure 4-3: Passenger sign up*

### 4.3.3 Passenger sign-in

This is the passenger's login function. First, it checks if the passenger is an existing user. Otherwise, create a new account, fill in the details, and submit. If an existing user can login to the system after providing the correct username and password. If the username or password is invalid, you will be redirected to the login page.

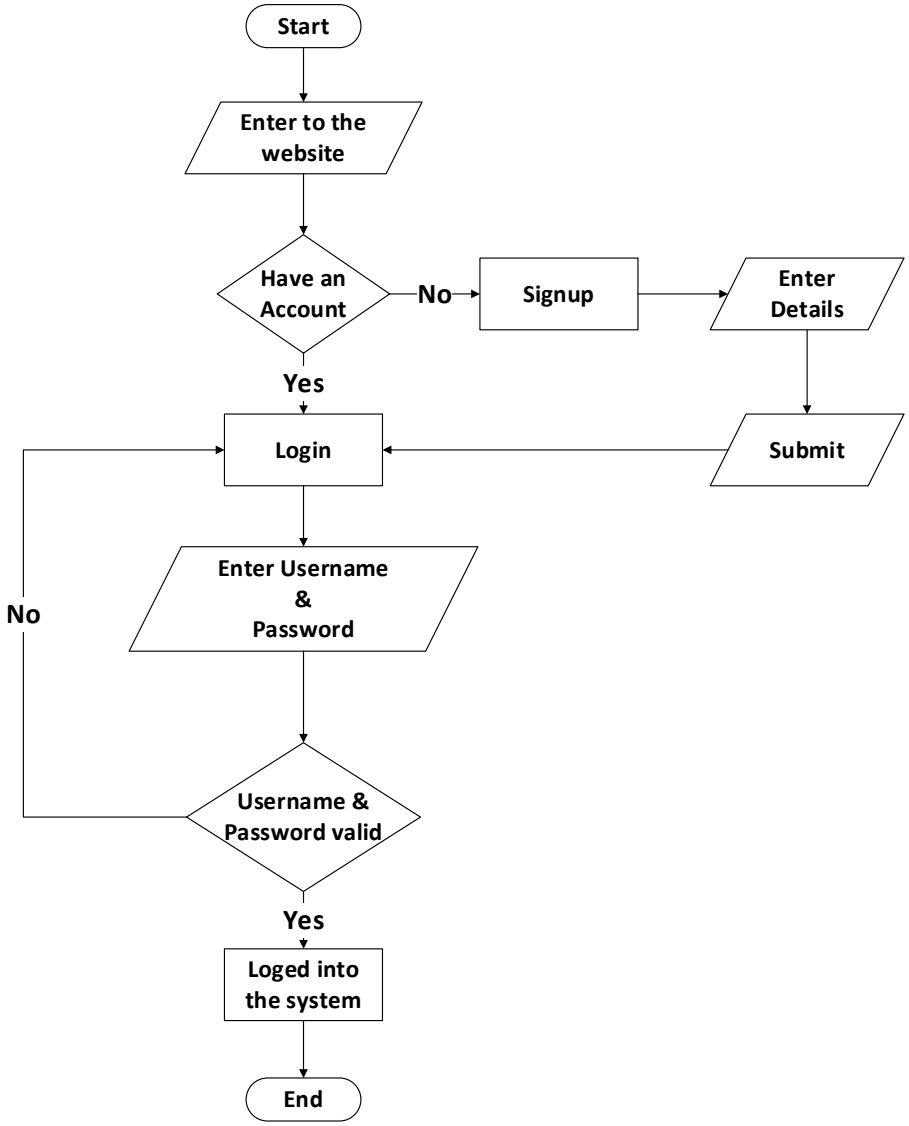



*Figure 4-4: Passenger sign-in*

### 4.3.4 Bus management

This is the bus detail input function. Administrators can manage this function. The administrator should enter all of the information requested on the form and submit it. Then all the details are saved in the database.

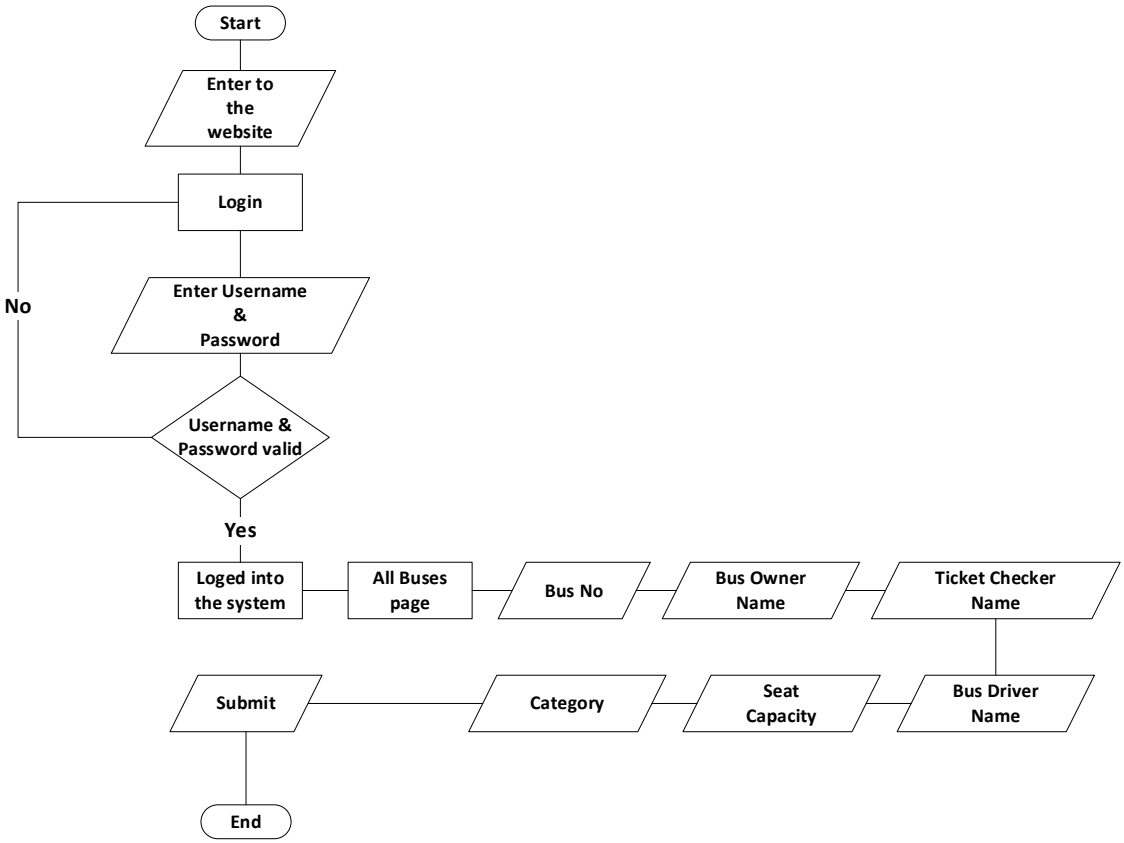



*Figure 4-5: Bus management*

### 4.3.5 Route management

This is the route detail input function. Administrators can manage this function. The administrator should enter all of the information requested on the form and submit it. Then all the details are saved in the database.

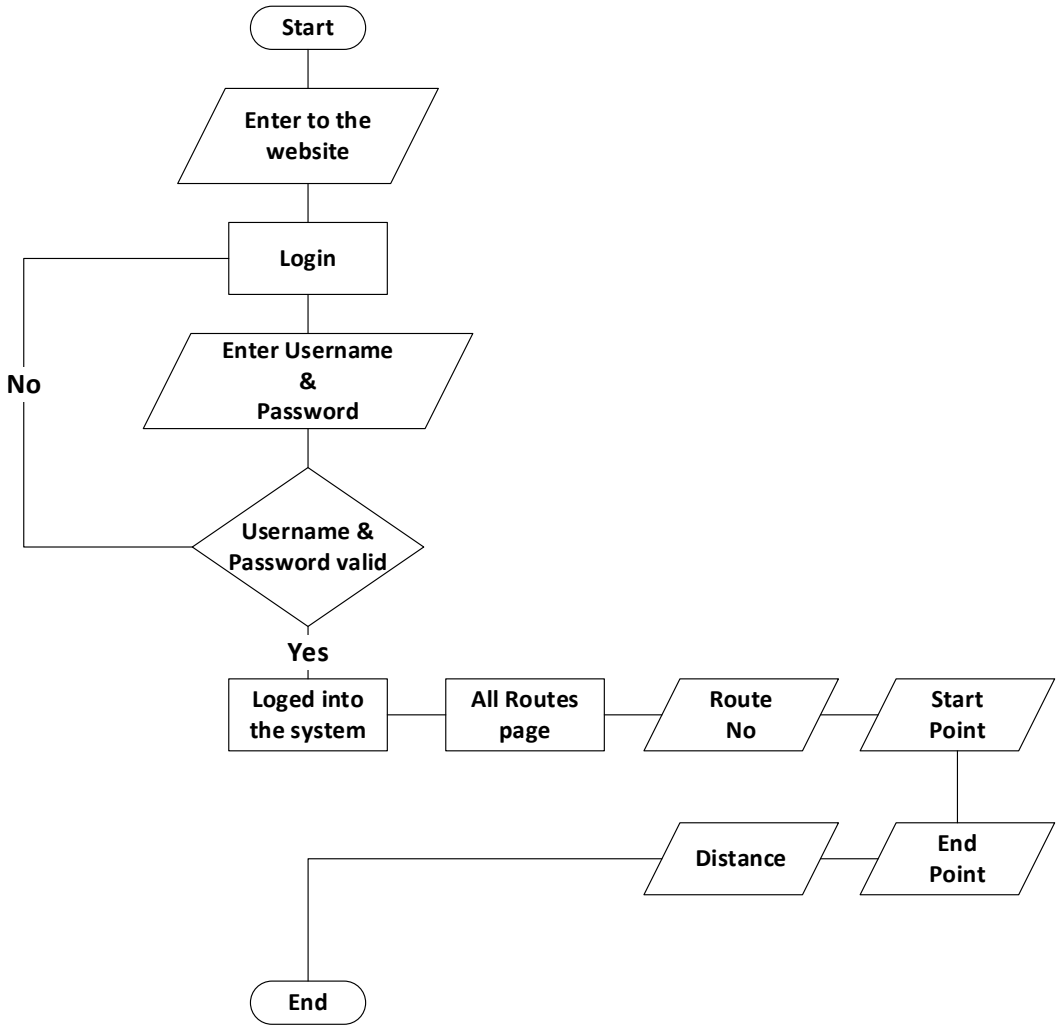



*Figure 4-6: Route management*

### 4.3.6 Schedule management

This is the schedule detail input function. Administrators can manage this function. The administrator should enter all of the information requested on the form and submit it. Then all the details are saved in the database.

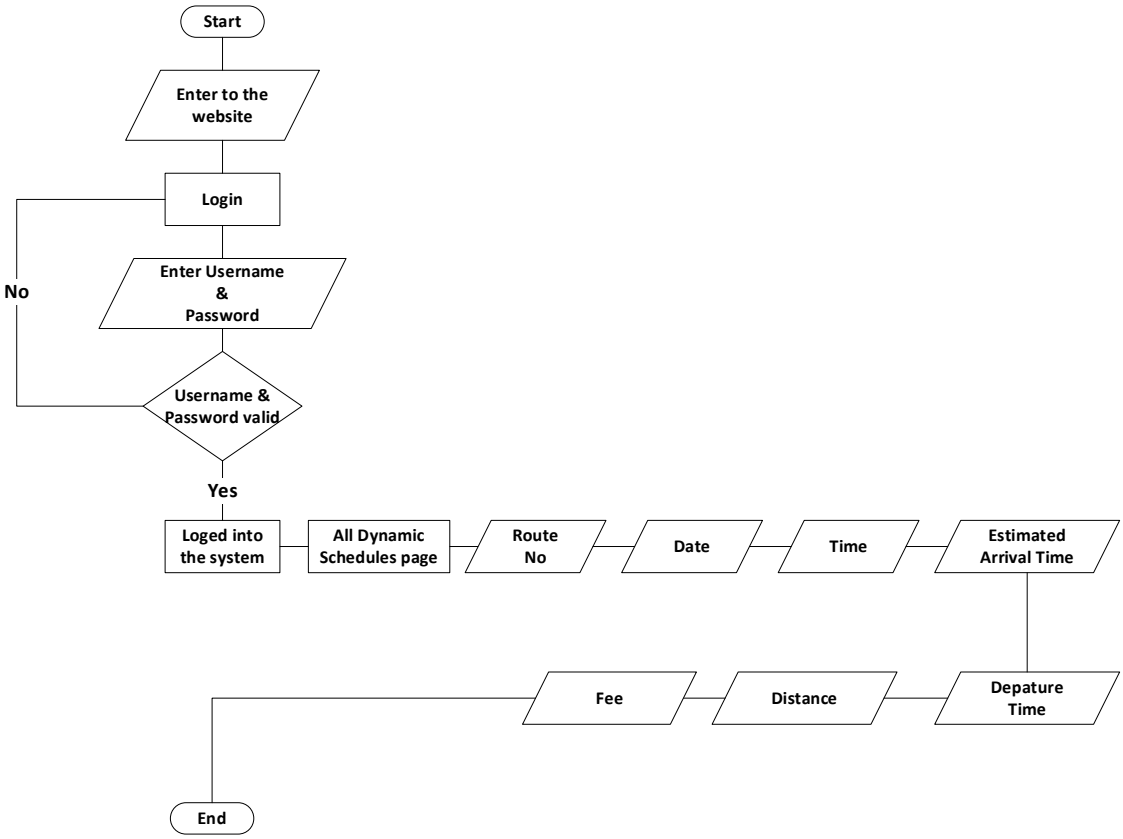



*Figure 4-7: Schedule management*

### 4.3.7 Booking ticket

This is the passenger's ticket-booking function. First, passengers must register and log into the system. After that, you have to give the correct details and pay for the ticket. After payment, the system checks whether the payment is valid or the card is valid. If the payment is valid, a text message will be sent to the passenger using the integrated SMS processing system. If not, the passenger is returned to the login page. The passenger can cancel the tickets after booking them.

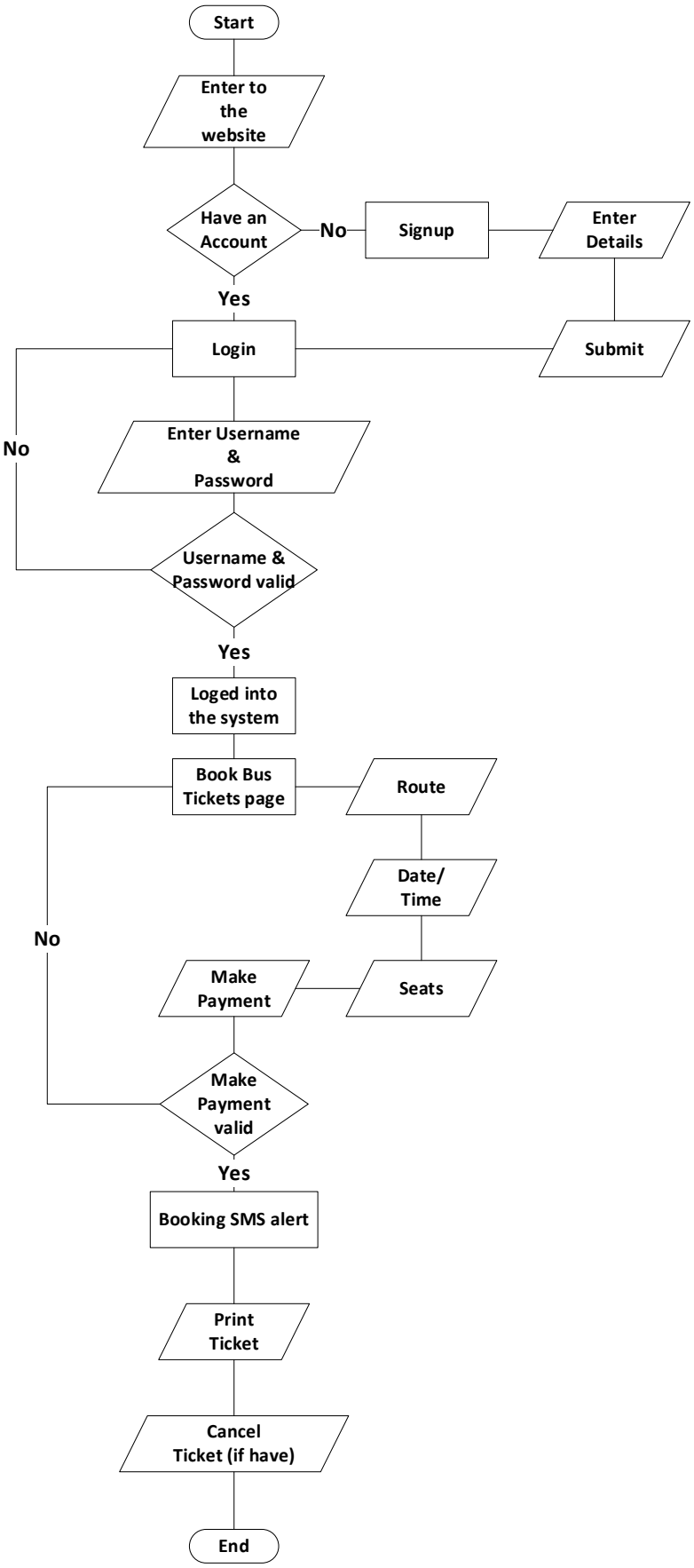



*Figure 4-8: Booking ticket*

### 4.3.8 Feedback management

This is the feedback function of the system. Passengers can send messages and inquiries to the administrator. Administrators can view it and quickly respond to it.

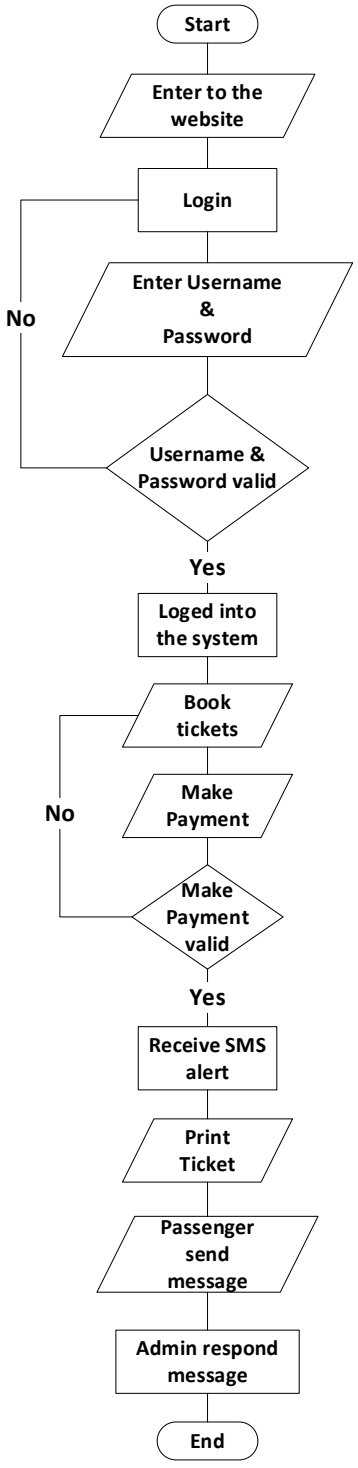



*Figure 4-9: Feedback management*

## 4.4 Database connection

```
<?php
function connect()
{
   $conn = new mysqli("localhost", "root", "", "wbbtrs");
   if (!$conn) die("Database is being upgrade!");
   return $conn;
}
$conn = connect();
if (!$conn) die("Under Construction!");
```

Here, the XAMPP server is used to host the database locally, which provides a reliable and stable way to store and retrieve data. Let's see how to connect it to the system.
The above code is a PHP script for function connect() and then tries to use it to connect to a database.
The connect() function creates a new mysqli object that connects to a MySQL database running on the same machine as the PHP script. It uses the credentials "root" and an empty string for the username and password, respectively, and specifies a database name of " wbbtrs ". If the connection is successful, the function returns the new mysqli object. If the connection fails, the function calls die() to immediately terminate the script and display an error message.
The script then calls the connect() function and assigns the returned mysqli object to the $conn variable. It checks whether $conn is truthy (i.e., not null, zero, false, or an empty string), and if it is not, it calls die() again to terminate the script and display another error message.

## 4.5 Summery

This section describes the implementation phase of the proposed system. This chapter includes an overview of the project and provides flowcharts. In the context of implementation, flowcharts help illustrate the various steps involved in the installation, configuration, testing, training, rollout, and maintenance phases, as well as the inputs and outputs of each step. Also, this section describes the backend and dashboard connected to the database.

In the next chapter describes the testing and evaluation section of the proposed system. In this section, some testing methods and test cases will be discussed. And here will be discussed some test results by comparing them with bar graphs and other graphs.

# Chapter 5 - Testing and Evaluation

## 5.1 Introduction

Testing involves operating a system or application under controlled conditions and evaluating the results. The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they should not or do not happen when they should. By preparing the pre-system testing and post-system testing, we can detect the bugs, runtime errors, and logic errors so that we can take the necessary corrective actions in time. A perfect system must be tested several times to ensure it is bug-free and achieves high quality.

## 5.2 Testing methodologies

The goal of utilizing numerous testing methodologies in your development process is to make sure your software can successfully operate in multiple environments and across different platforms. These can typically be broken down into functional and non-functional tests. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using use cases provided by the design team or business analyst. Non-functional testing methods incorporate all test types focused on the operational aspects of a piece of software. We have used several methodologies for testing this project.

### 5.2.1 Unit testing

Unit testing is to test software in terms of a unit, a module, a function, or a specific section of code. This testing occurs while the software is being developed and before completion. For unit testing, test cases are designed to verify that an individual unit implements all design decisions made in the unit's design specification. A detailed unit test specification should include both positive and negative tests.

### 5.2.2 Integration testing

After unit testing is completed, integration testing begins. The aim is to ensure that the unique components of the app still work in line with passenger needs. This task will be carried out by passengers to develop test suites with the aim of exercising the interfaces between the components. Integration testing will be complete when actual results and expected results line up, or when changes can be explained or accepted based on user input.

### 5.2.3 System testing

Upon completion of integration testing, system testing will begin. During system testing, the complete system is configured in a controlled environment to validate its accuracy and completeness in performing the functions as designed. The system test will simulate production as it will occur in the "production-like" test environment and test every function of the system that will be required in production. It is also important that validation of the system meets the functional and non-functional requirements.

### 5.2.4 Acceptance testing

Acceptance testing will give both customers and administrators the opportunity to verify the system's functionality and usability prior to its deployment. The users will test the system's interaction with the database, using network communications, or interacting with other hardware or other applications. The system is tested with data supplied by end users rather than simulated test data. Acceptance testing reveals errors and omissions in the system requirement definition because real data exercises the system in different ways from the test data. It also reveals requirements problems where the system’s facilities do not adequately meet the user’s need or the system's performance is unacceptable. The testing process continues until the system developer and client agree that the online bus ticketing web portal is an acceptable implementation of the system requirement.

## 5.3 Test case

A test case is defined as a group of conditions under which a tester determines whether a software application is working as per the customer's requirements or not. Test case design includes preconditions, case name, input conditions, and expected result. A test case is a first-level action derived from test scenarios. Following are a few test case examples of this project:

### 5.3.1 Test case of admin login

*Table 5-1: Test case of admin login*

| Test Case ID | Test Description | Prerequisite | Test Steps | Input Data | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|---|---|
| TC admin login 001 | verify login with invalid email and password | should be registered user | 1.browse the site 2. go to login page 3. enter invalid email and password | invalid email and Password | display "Access Denied." error message at the top of the page | as expected | pass |
| | verify login with valid email and password | should be registered user | 1.browe the site 2. go to login page 3. enter valid email and password | valid email and Password | direct to the admin's dashboard | as expected | pass |

### 5.3.2 Test case of passenger sign-up

*Table 5-2: Test case of passenger sign-up*

| Test Case ID | Test Description | Prerequisite | Test Steps | Input Data | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|---|---|
| TC passenger signup 002 | passenger sign-up with invalid details | should be browse the site | 1.browse the site 2. go to sign-up page 3. enter invalid details | invalid name, contact, email, picture, address and both of password and confirm password | display "could not complete your registration" error message at the top of the page | as expected | pass |
| | passenger sign-up with valid details | should be browse the site | 1.browse the site 2. go to sign-up page 3. enter valid details | valid name, contact, email, picture, address and both of password and confirm password | direct to the passenger login page | as expected | pass |

### 5.3.3 Test case of passenger login

*Table 5-3: Test case of passenger login*

| Test Case ID | Test Description | Prerequisite | Test Steps | Input Data | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|---|---|
| TC passenger login 003 | verify login with invalid email and password | should be registered user | 1.browse the site 2. go to login page 3. enter invalid email and password | invalid email and Password | display "Access Denied." error message at the top of the page | as expected | pass |
| | verify login with valid email and password | should be registered user | 1.browe the site 2. go to login page 3. enter valid email and password | valid email and Password | direct to the passenger's dashboard | as expected | pass |

### 5.3.4 Test case of entering bus details

*Table 5-4: Test case of entering bus details*

| Test Case ID | Test Description | Prerequisite | Test Steps | Input Data | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|---|---|
| TC enter bus details 004 | admin enters bus details with invalid inputs | admin should login to the system | 1.login<br>2.go to bus page<br>3.enter invalid inputs | invalid bus no, owner name, driver name, ticket checker name, seat capacity, category | display " Fill Form Properly !" error message at the top of the page | as expected | pass |
| | admin enters bus details with valid inputs | admin should login to the system | 1.login<br>2.go to bus page<br>3.enter valid inputs | valid bus no, owner name, driver name, ticket checker name, seat capacity, category | display "Bus Added!" message at the top of the page | as expected | pass |

### 5.3.5 Test case of entering route details

*Table 5-5: Test case of entering route details*

| Test Case ID | Test Description | Prerequisite | Test Steps | Input Data | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|---|---|
| TC enter route details 005 | admin enters route details with invalid inputs | admin should login to the system | 1.login 2.go to route page 3.enter invalid inputs | invalid route no, from, to, distance | display " Fill Form Properly !" error message at the top of the page | as expected | pass |
| | admin enters route details with valid inputs | admin should login to the system | 1.login 2.go to route page 3.enter valid inputs | valid route no, from, to, distance | display "Route Added!" message at the top of the page | as expected | pass |

## 5.4 Test evaluation

In order to conduct the testing, ten random passengers in the Colombo central bus stand had been invited to test the prototype. The results obtained are as below:

### 5.4.1 Features and functionalities

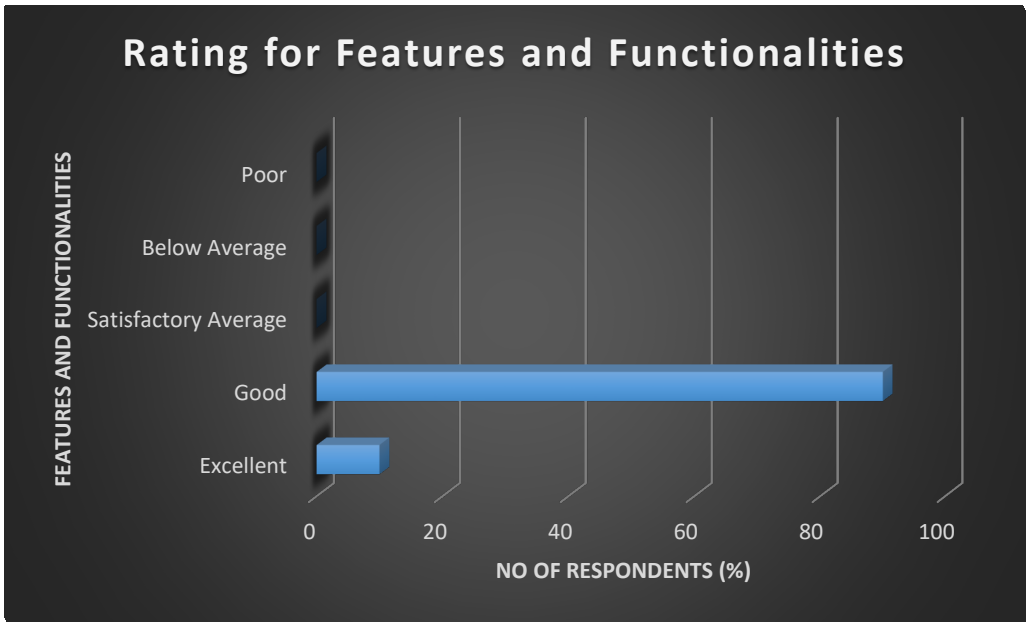



*Figure 5-1: Respondents who have rated the systems features and functionalities*

Based on Figure 5-1, most of the respondents rated the system as good, and some rated it as excellent. This shows that the respondents are satisfied with the existing features and functionalities of the online bus ticketing system.

### 5.4.2 Familiarity of usage

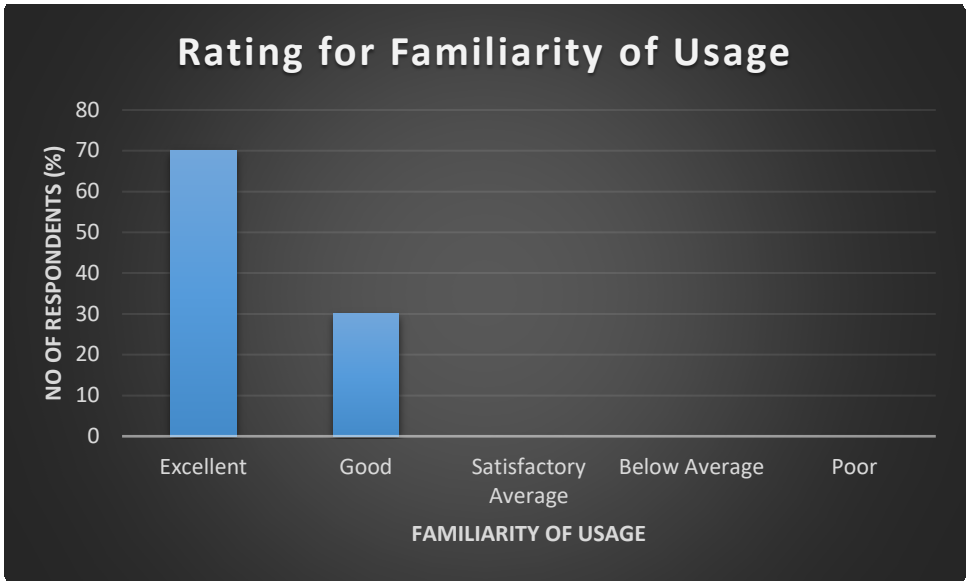



*Figure 5-2: Respondents who have rated the familiarity of the online bus ticketing system*

Based on Figure 5-2, 70% of the testers rated their familiarity with the usage of the system as excellent and the rest, 30%, as good. This shows that the system is easy to use and will not be an excuse for a customer to blame the complexity of the system for not using the system. This result again shows that the system can take over the current manual selling of bus tickets, finally meeting this project's objective.

### 5.4.3 Helpful in ticket purchasing


*Figure 5-3: Respondents who have rated helpfulness in ticket purchasing*

According to Figure 5-3, 40% of respondents rated the system as excellent in assisting them during ticket purchasing, 30% rated it good, and 30% rated it satisfactory. This is good feedback since no one has rejected the system for use, and the system can therefore be fully deployed by the bus operators. Again, this fact has proven this project's objective to be the transformation of ticket purchasing via the internet.

### 5.4.4 Additional features and functions

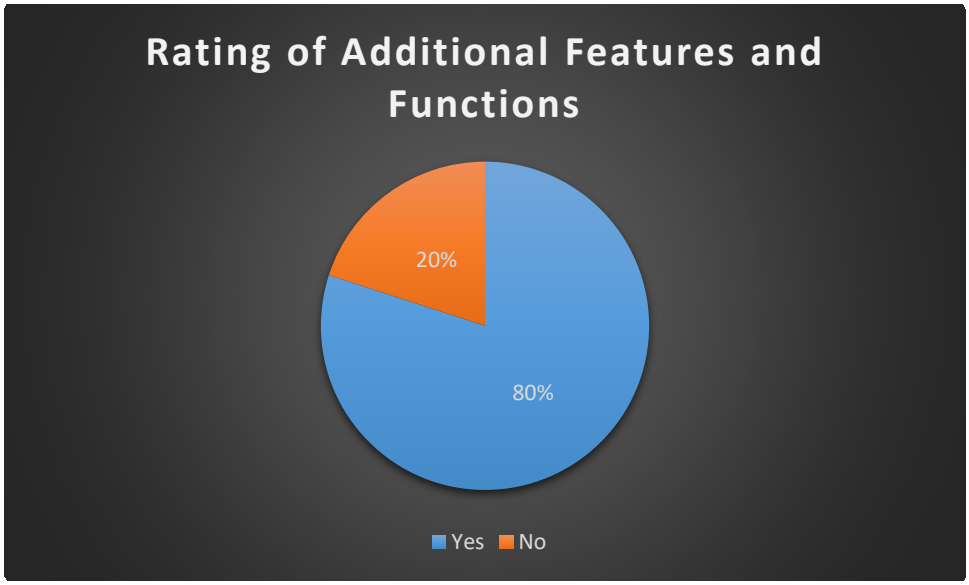



*Figure 5-4: Respondents who have requested additional features and functions in the system*

Based on Figure 5-4, 80% of the respondents agreed to include more features and functions for the online bus ticketing system. Some of the suggested features include the inclusion of a ticket reservation feature. This suggestion has been taken into consideration and will be incorporated as a future enhancement.

### 5.4.5 Attractiveness of interface design

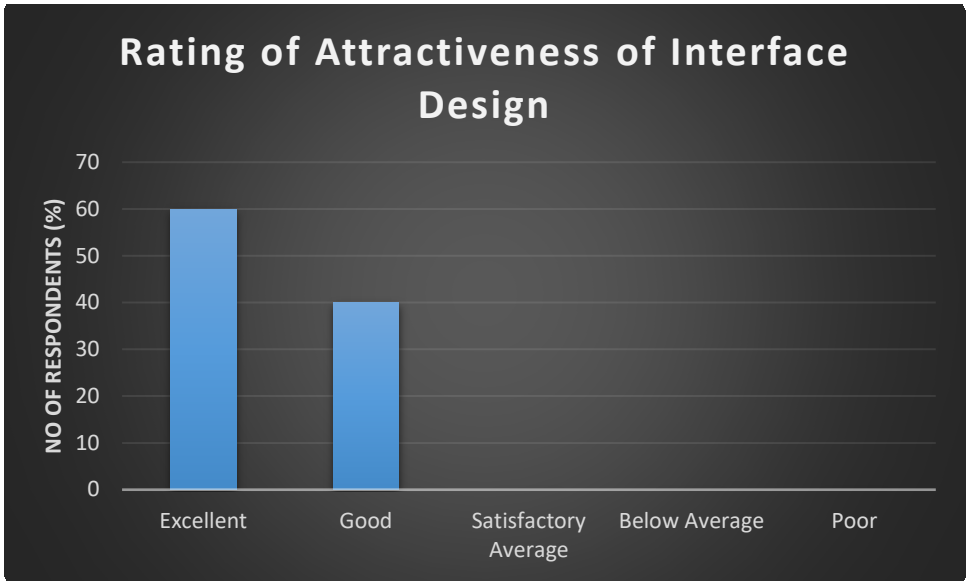



*Figure 5-5: Respondents who have rated the online bus ticketing web portal interface design*

Based on Figure 5-5, 60% of the respondents have rated the interface as excellent and 40% as good. From this fact, it is depicted that this system has successfully met one of the design guidelines, which is strive Human-Computer Interaction.

## 5.5 Summery

This section describes the testing and evaluation phases of the proposed system. This chapter includes test cases and test methods. In addition, we show some test results in bar graphs and compare them with other graphs. In this section, some feedback can be obtained from the stakeholders about the overall system and its functions. It helps in system development.

In the next chapter, we will discuss the discussion phase of the proposed system. In this section, the conclusion, the future plan of the project, the challenges that I faced, the justification of the choice of tools, and my personal reflection will be discussed. Readers can also get a comprehensive understanding of the progress of the project here.

# Chapter 6 - Conclusion and Future work

## 6.1 Introduction

This chapter will conclude the final year's research papers. This chapter will discuss future project improvements and challenges faced during the project development phase.

Additionally, the last chapter is about the technologies used in the study and their focus on activities, including justification of tool selection and personal reflection.

## 6.2 Conclusion

The Web Based Bus Ticket Booking System is a web-based application. It includes checking the availability of seats in the bus and checking the bus route, timetable, etc. It eliminates many disadvantages of ticket collection systems, such as the fact that users can buy tickets anytime before the journey. Through this system, the users can pay the fare online, and they don't need to carry exact change to buy the tickets.

This is a potential system developed in JQuery, PHP and uses HTML, CSS and JS for web view and SQL as a database.

This report covers all front-end and back-end functions. The front end functions are admin login, passenger registration, passenger login, booking a passenger ticket, passenger past history, sending passenger feedback, and others. Backend functions are: managing users, buses, routes, timetables, report generation, reservations, tickets, and feedback.

We have provided the design diagrams to get a clear idea about the system as well as this report consists of the flowcharts.

## 6.3 Future plans

### 6.3.1 Language support

The Online Bus Ticketing System web portal could be enhanced to provide more language support, such as a Sinhala language version, besides the current English language version. This will enable information to be displayed in a different language. As a result, this will broaden the usage of the system and allow it to interact with more customers.

### 6.3.2 Enhanced user interface

The user interface of the system can be enhanced to be more attractive, impressive, and interactive when this web portal is converted to a real-time system.

### 6.3.3 Increase administrators' tasks

The administrator's task can be further enhanced to include more features to ease the maintenance process. For example, analytical tools, data mining, other relevant reports, and database backup are recommended to be included in this online bus ticketing web portal to provide more analytical function to the company.

### 6.3.4 Common working community

The system can also be a "newsletter" to all employers and employees who are related to bus transportation in Sri Lanka. Various information or news can be displayed to the targeted audience, such as employment availability, bus operator news, and so on.

## 6.4 Challenges

During the development of this project, some difficulties occurred. So we managed to calm things down by looking more deeply into the problems and trying to find solutions over the Internet. Besides, I feel frustrated when I need to use some new programming logic that I have never learned before. Luckily, my friends are willing to guide me and show me the way to achieve it. Besides that, I did a lot of research over the internet while learning from my friends.

## 6.5 Justification of choice of tools

I thought of using the Laravel PHP framework instead of core PHP. Because I believe the Laravel framework is easier to use than basic PHP. But there was a problem: I didn't know much about the Laravel framework, and I didn't have much time to complete this project. Then I discussed it with my supervisor and friends, where I decided to work on basic PHP.

## 6.6 Personal reflection

From system planning, requirement analysis, system design, coding, and testing, this project was finally done in the given time. I found that time management was very important when implementing a system. If the system cannot be done in time, the consequence is system failure, which will cause a massive loss for a company. Besides, working together is better than working individually.

During developing this project, I faced a lot of problems, such as database connection problems and project programming. Hours and hours I spent trying to find the solution and praying for people to help me solve the problem.

Thanks to the internet and my friend, I am able to solve my problems, such as retrieving data from a database, viewing it in grid view, inserting images into the database, and using the date and time picker in a web form.

With the help and support from my friends, I am solving the problem and completing the project on time. I also would like to thank my family members, who are always by my side motivating and pushing me to complete my project.

And also, I would like to express my sincere gratitude to my project supervisors, Mr. Chathura and Mr. Roshan, who are very kind and considerate to me. They always give me opinions and feedback on my project. Also, I would like to thank the dean of our faculty, Mr. Saminda, for his immense support to complete this project. I would like to express my gratitude to everyone who helped in any way to complete this project.

## 6.7 Summary

In conclusion, this report has provided an overview of the frontend module for passengers in the bus booking system. The system is built using various technologies, such as JQuery, PHP, MySQL, HTML, Bootstrap, and CSS for design. The report has also discussed the functionality of the system, search functionality, ticket booking, and the backend modules responsible for managing bus booking.

# References


Anon., n.d. *busseat.lk.* [Online]
Available at: https://busseat.lk/

[Accessed 25 01 2023].
Anon., n.d. *busticket.lk.* [Online]
Available at: https://www.busticket.lk/

[Accessed 25 01 2023].
Anon., n.d. *sltb.eseat.lk.* [Online]
Available at: https://sltb.eseat.lk/

[Accessed 25 01 2023].
Anon., n.d. *Super line Travels.* [Online]
Available at: https://superline.lk/tickets

[Accessed 25 01 2023].
Anon., n.d. *Dailybus.lk.* [Online]
Available at: http://dailybus.lk/search

[Accessed 25 01 2023].
Anon., n.d. *redBus.* [Online]
Available at: https://www.redbus.in/bus-tickets

[Accessed 25 01 2023].
Anon., n.d. *CheckMyBus.* [Online]
Available at: https://www.checkmybus.com/

[Accessed 25 01 2023].
Anon., n.d. *zingbus.* [Online]
Available at: https://www.zingbus.com/

[Accessed 25 01 2023].
Anon., n.d. *TechTarget.* [Online]
Available at: https://www.techtarget.com/searchsoftwarequality/definition/3-tier-application

[Accessed 10 03 2023].
Anon., n.d. *Tallyfy.* [Online]
Available at: https://tallyfy.com/uml-diagram/#:~:text=A%20UML%20diagram%20is%20a,document%20information%20about%20the%20system.

[Accessed 08 03 2023].
Anon., n.d. *IBM.* [Online]
Available at: https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case

[Accessed 05 03 2023].
Anon., n.d. *Lucidchart.* [Online]
Available at: https://www.lucidchart.com/pages/uml-sequence-diagram#:~:text=A%20sequence%20diagram%20is%20a,to%20document%20an%20existing%20process.

[Accessed 07 03 2023].
Anon., n.d. *tutorialspoint.* [Online]
Available at: https://www.tutorialspoint.com/uml/uml_activity_diagram.htm

[Accessed 10 03 2023].
Anon., n.d. *Lucidchart.* [Online]
Available at: https://www.lucidchart.com/pages/er-diagrams

[Accessed 10 03 2023].
Anon., n.d. *tutorialspoint.* [Online]
Available at: https://www.tutorialspoint.com/uml/uml_class_diagram.htm#:~:text=Class%20diagram%20is%20basically%20a,while%20drawing%20a%20class%20diagram%20%E2%88%92

[Accessed 10 03 2023].
Anon., n.d. *Techwalla.* [Online]
Available at: https://www.techwalla.com/articles/what-is-relational-database-schema

[Accessed 10 03 2023].
Anon., n.d. *Career Guide.* [Online]
Available at: https://www.indeed.com/career-advice/career-development/what-is-a-data-table#:~:text=A%20data%20table%20is%20a,in%20the%20table%20more%20clearly.

[Accessed 10 03 2023].
Anon., n.d. *tutorialspoint.* [Online]
Available at: https://www.tutorialspoint.com/software_engineering/software_user_interface_design.htm

[Accessed 10 03 2023].

# Appendix A

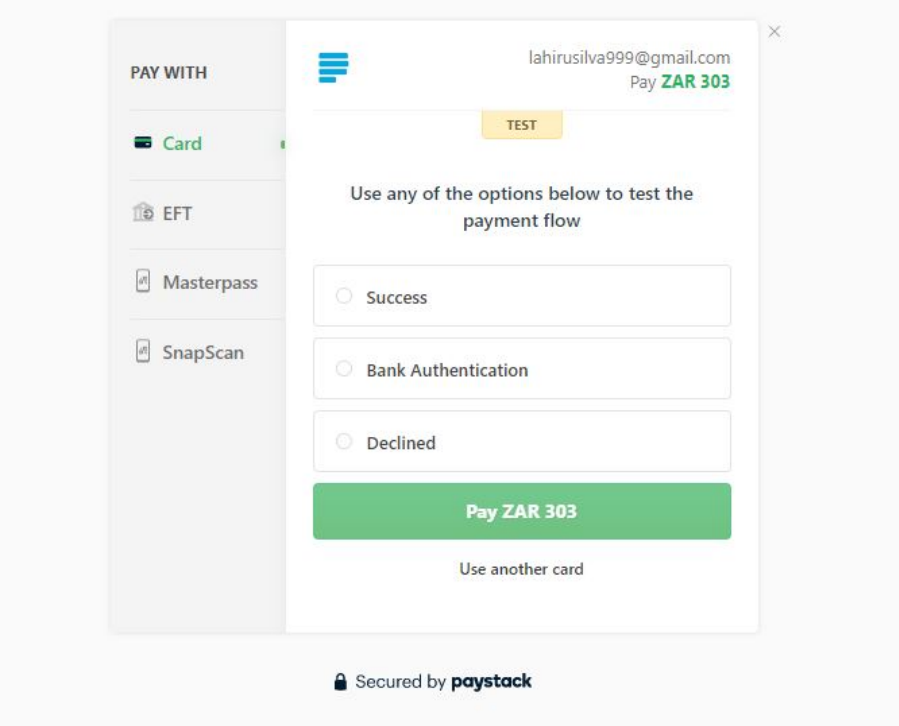
## Technology Adapted

### API (Application Programming Interface)

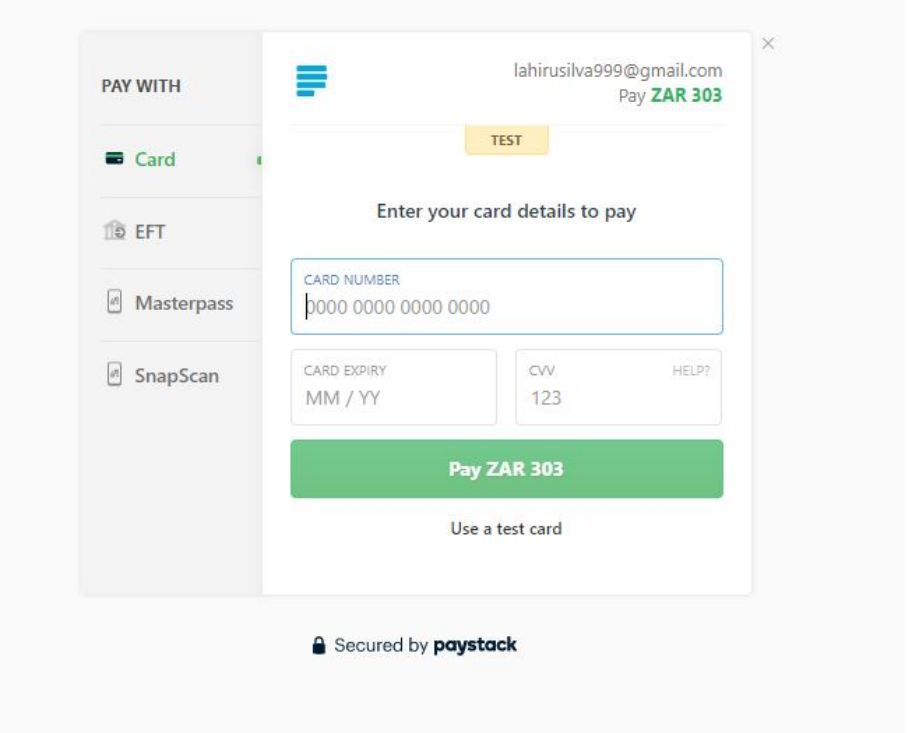
API stands for Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses.

*Paystack payment gateway API*

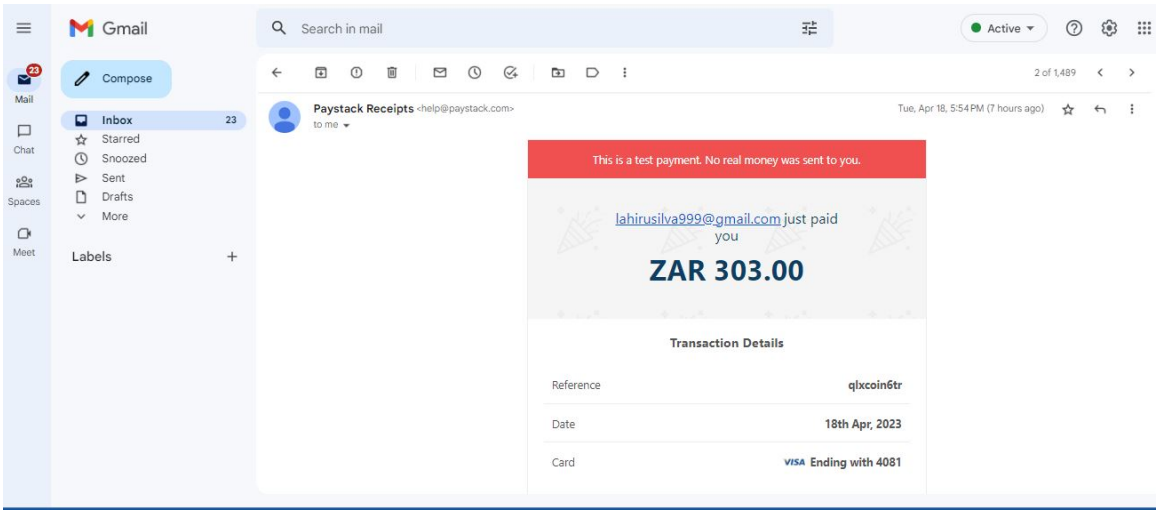
I used a payment gateway API for this system. Its name is Paystack. It's given free of charge for testing to software developers and IT professionals. And also, Paystack is a payment solution provider that facilitates online payment through its payment gateway and in-person payment through its Point-of-Sale (PoS) Terminal, which makes it easy for merchants to accept credit and debit card payments online and at physical retail ocations from users or customers.



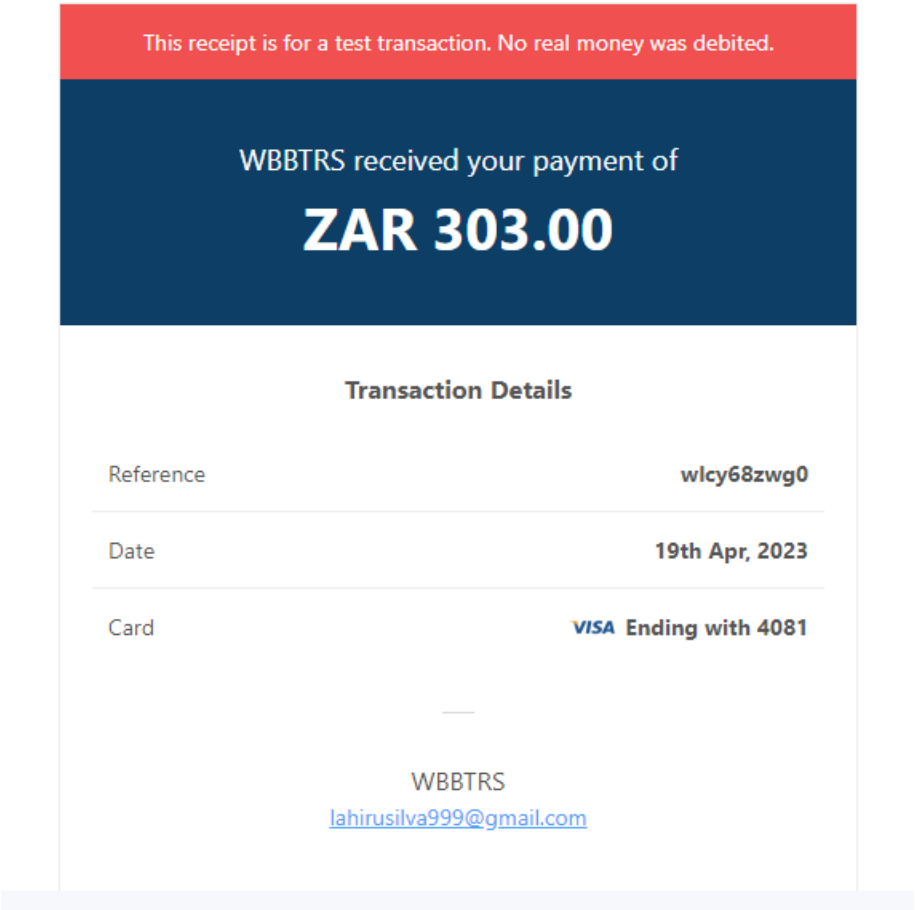

*Appendix A 1: Test card use for payment testing*




*Appendix A 2: Credit/Debit card use for payment testing*




*Appendix A 3: Mail confirmation message*

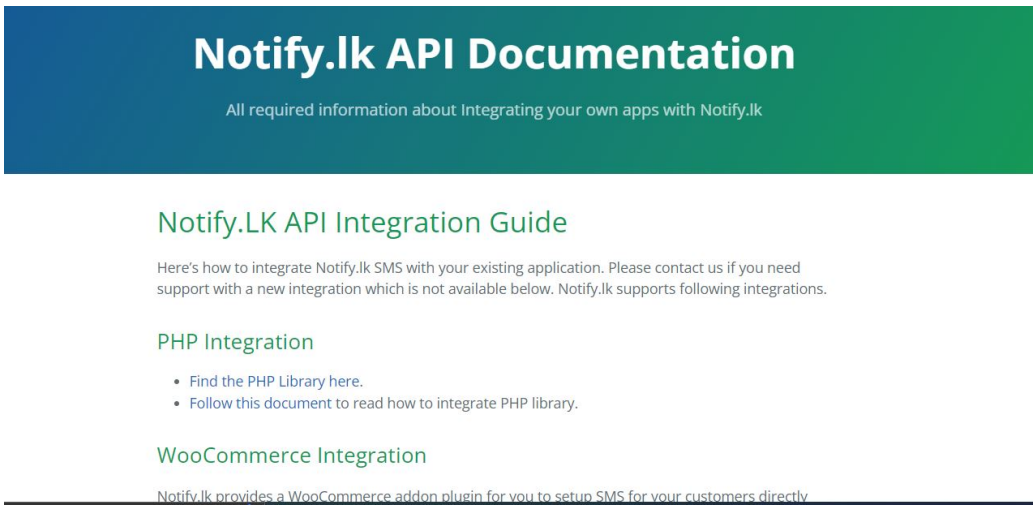



*Appendix A 4: Payment confirmation message*

*Notify SMS API*

I used the SMS API for this system. Its name is Notify. It’s given a 10-day free SMS for system testing and integration. While it's also a Sri Lankan-based company.


*Appendix A 5: Notify.lk home page*




*Appendix A 6: Notify.lk API documentation*

## Data tables

*Appendix A 7: route table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| no | varchar(15) | No | | |
| start | varchar(100) | No | | |
| stop | varchar(100) | No | | |
| distance | int(11) | No | | |

*Appendix A 8: route indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 4 | A | No | |

*Appendix A 9: schedule table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| bus_id | int(11) | Yes | NULL | |
| route_id | int(11) | Yes | NULL | |
| route_no | varchar(15) | Yes | NULL | |
| date | varchar(30) | Yes | NULL | |
| time | varchar(10) | Yes | NULL | |
| esti_arrive_time | varchar(10) | Yes | NULL | |
| depart_time | varchar(10) | Yes | NULL | |
| route_distance | varchar(15) | Yes | NULL | |
| fee | float | Yes | NULL | |

*Appendix A 10: schedule indexes*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 8 | A | No | |
| route_id | BTREE | No | No | route_id | 8 | A | Yes | |
| bus_id | BTREE | No | No | bus_id | 4 | A | Yes | |

*Appendix A 11: users table*

| Column | Type | Null | Default | Comments |
|---|---|---|---|---|
| id *(Primary)* | int(11) | No | | |
| email | varchar(40) | No | | |
| password | varchar(255) | No | | |

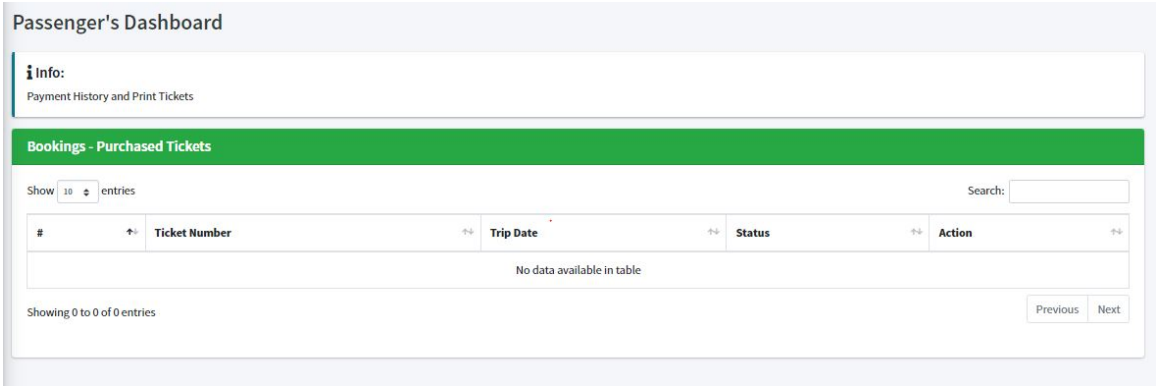
*Appendix A 12: users table*

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 0 | A | No | |

## Frontend interfaces

### Passenger view booking history

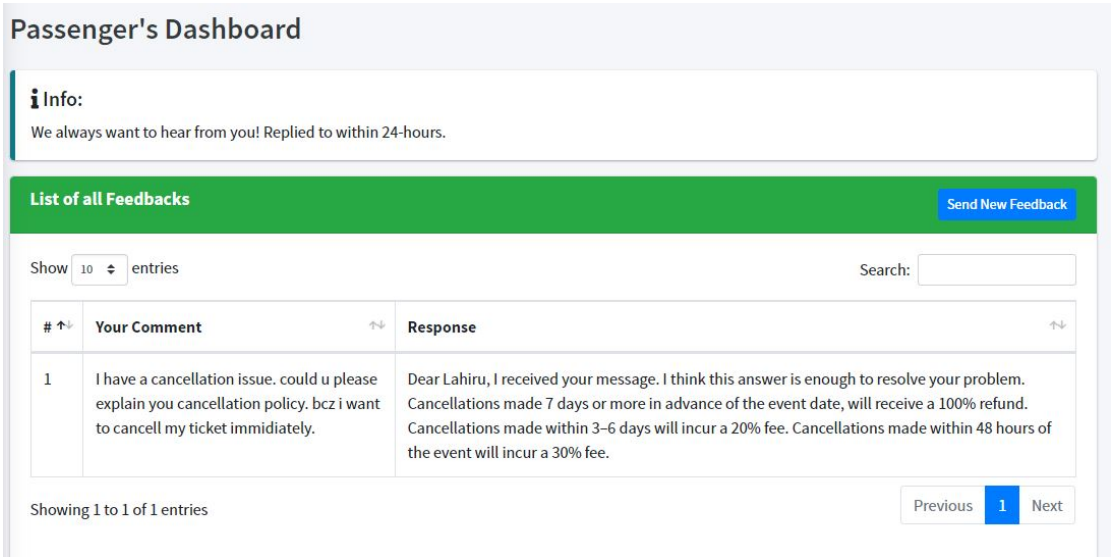
Passengers can view past booking histories through this. Apart from this, a booking ticket can also be printed through this.




*Appendix A 13: View booking history*

### Passenger feedback

Passengers can send feedback or make inquiries through this.

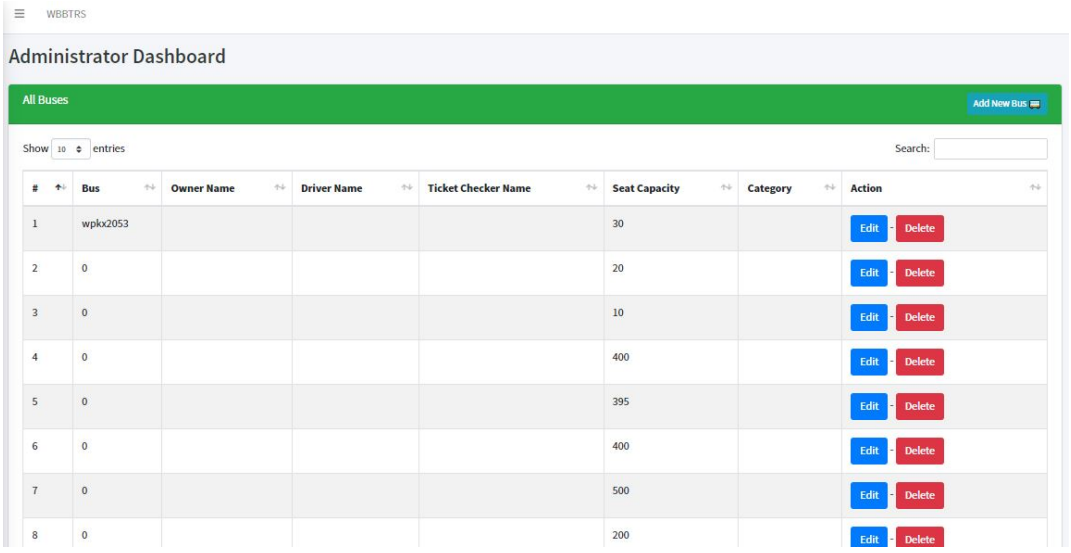



*Appendix A 14: View booking history*

## Backend interfaces

### Bus management

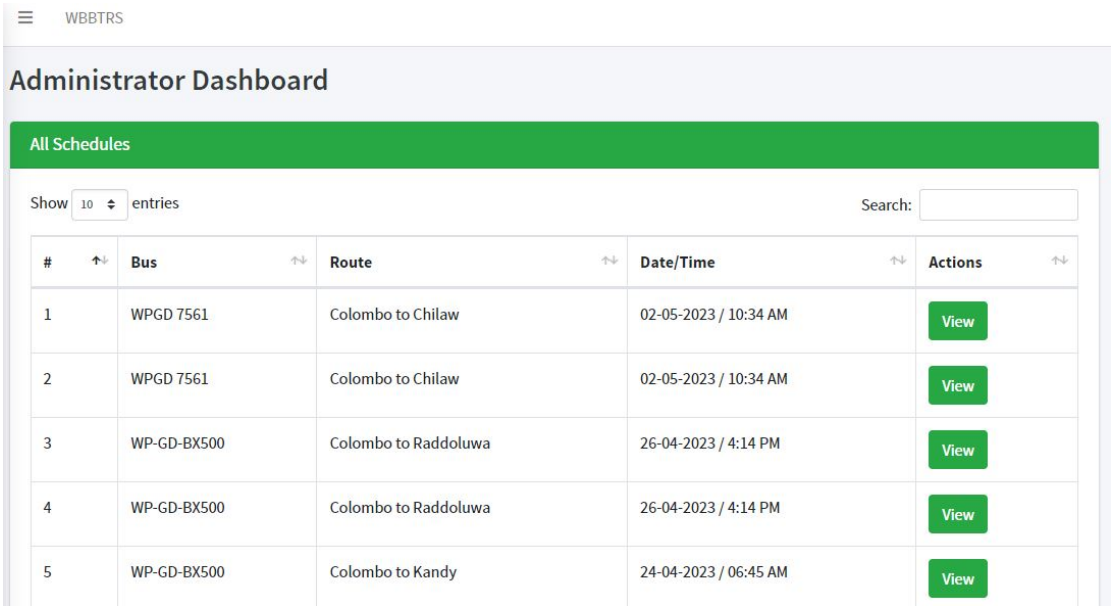
Administrators can add, edit, update, and delete bus details through this.




*Appendix A 15: Bus management*

## Generate report

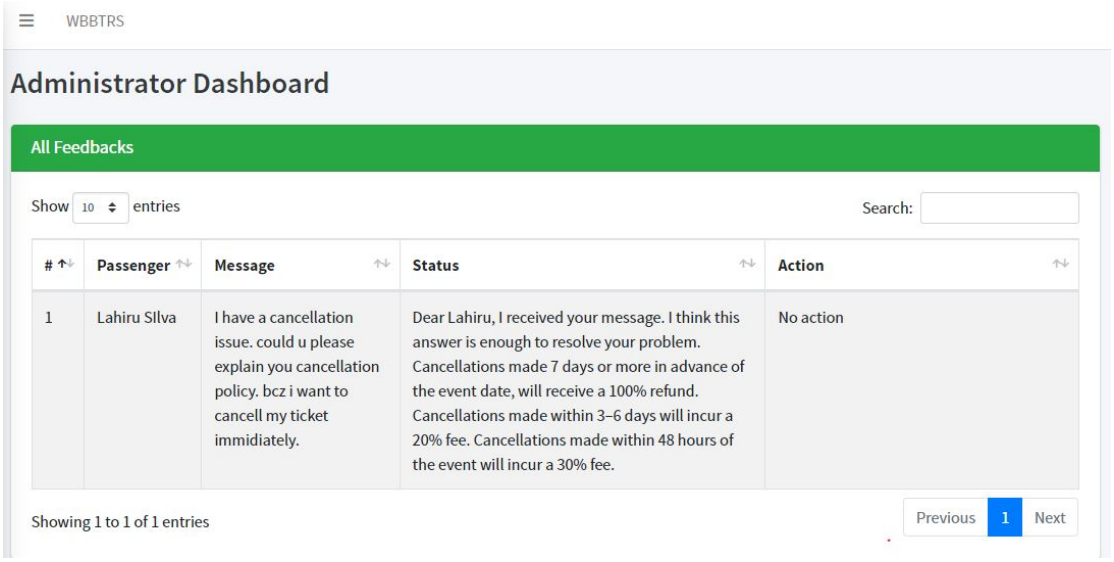
Administrators can get reports through this.




*Appendix A 16: Generate report*

## Feedback management

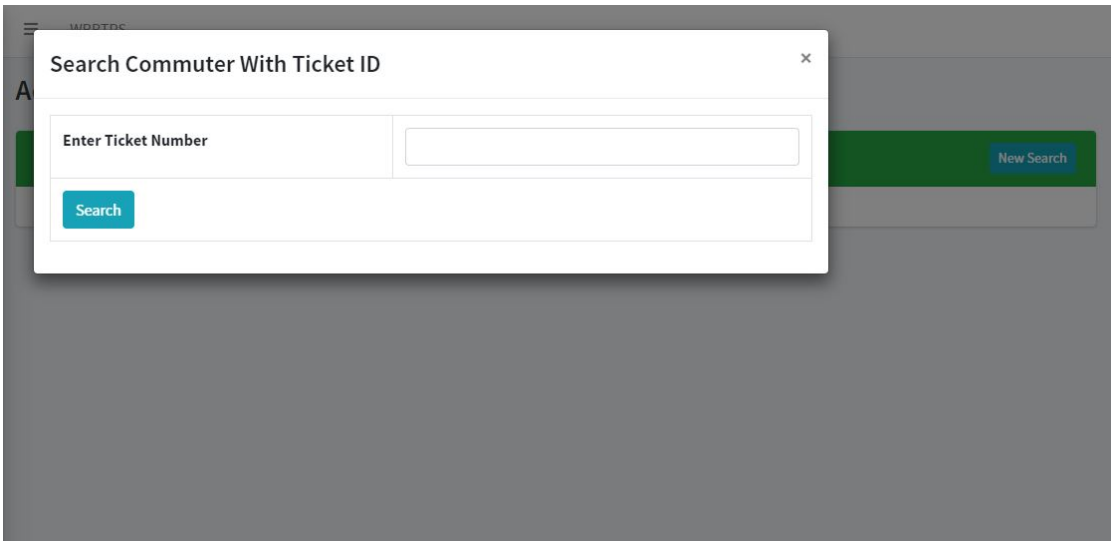
Administrators can view passenger feedback and answer any queries they may have.




*Appendix A 17: Feedback management*

**Search feature**

Administrators can search any ticket numbers through this.




*Appendix A 18: Search feature*

# Appendix B

## Test Cases

### Test case of entering schedule details

*Appendix B 1: Test case of entering schedule details*

| Test Case ID | Test Description | Prerequisite | Test Steps | Input Data | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|---|---|
| TC enter schedule details 006 | admin enters schedule details with invalid inputs | admin should login to the system | 1.login 2.go to schedule page 3.enter invalid inputs | invalid bus no, route, fee, date, time, estimated arrival time, departure time, distance | display " Fill Form Properly !" error message at the top of the page | as expected | pass |
| | admin enters schedule details with valid inputs | admin should login to the system | 1.login 2.go to schedule page 3.enter valid inputs | valid bus no, route, fee, date, time, estimated arrival time, departure time, distance | display "Schedule Added!" message at the top of the page | as expected | pass |

**Test case of booking ticket**

*Appendix B 2: Test case of booking ticket*

| Test Case ID | Test Description | Prerequisite | Test Steps | Input Data | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|---|---|
| TC booking ticket 007 | passengers booking ticket with invalid inputs | passengers should login to the system | 1.login 2.go to book bus tickets page 3.enter invalid inputs | invalid number of tickets | display "Invalid Number " error message at the top of the page | as expected | pass |
| | passengers booking ticket with valid inputs | passengers should login to the system | 1.login 2.go to book bus tickets page 3.enter valid inputs | valid number of tickets | direct to the "Booking Preview " page | as expected | pass |

**Test case of feedback**

*Appendix B 3: Test case of feedback*

| Test Case ID | Test Description | Prerequisite | Test Steps | Input Data | Expected Result | Actual Result | Status |
|---|---|---|---|---|---|---|---|
| TC feedback 008 | passengers send messages with invalid inputs | passengers should login to the system | 1.login 2.go to feedback page 3.enter invalid inputs | invalid inputs | display "Feedback could not be sent! Try again!" error message at the top of the page | as expected | pass |
| | passengers send messages with valid inputs | passengers should login to the system | 1.login 2.go to feedback page 3.enter valid inputs | valid inputs | display "Feedback sent! We will get back to you" message at the top of the page | as expected | pass |

# Appendix C

## Front-end code segment

### adminsignin.php

This includes part of the adminsignin.php code segment. Here, the system checks that the admin email ID and password entered are correct. If correct, the administrator can login to the system. Otherwise, administrators cannot login to the system.

```
<?php
$cur_page = 'signup';
include 'includes/inc-header.php';
if (isset($_POST['email'])) {
$email = $_POST['email'];
$password = $_POST['password'];
if (!isset($email, $password)) {
?>
<script>
alert("Ensure you fill the form properly.");
</script>
<?php
} else {
//Check for login
$password = md5($password);
$check = $conn->prepare("SELECT * FROM users WHERE email = ? AND
password = ?");
$check->bind_param("ss", $email, $password);
if (!$check->execute()) die("Form Filled With Error");
$res = $check->get_result();
$no_rows = $res->num_rows;
if ($no_rows ==  1) {
$row = $res->fetch_assoc();
$id = $row['id'];
session_regenerate_id(true);
$_SESSION['category'] = "super";
$_SESSION['admin'] = $id;
?>
<script>
alert("Access Granted!");
window.location = "admin.php";
</script>
```

**individual_reg.php**

This includes part of the individual_reg.php code segment. Here, passengers need to fill out and submit the form. If the entered details are valid, passengers can register in the system. If not valid, passengers can't register in the system.

```
<?php
$cur_page = 'signup';
include 'includes/inc-header.php';
include 'includes/inc-nav.php';
if (isset($_POST['name'])) {
$name = $_POST['name'];
$phone = $_POST['phone'];
$email = $_POST['email'];
$file = 'file';
$address = $_POST['address'];
$cpassword = $_POST['cpassword'];
$password = $_POST['password'];
if (!isset($name, $address, $phone, $email, $password, $cpassword) || ($password !=
$cpassword)) { ?>
<script>
alert("Ensure you fill the form properly.");
</script>
<?php
} else {
//Check if email exists
$check_email = $conn->prepare("SELECT * FROM passenger WHERE email = ?
OR phone = ?");
$check_email->bind_param("ss", $email, $phone);
$check_email->execute();
$res = $check_email->store_result();
$res = $check_email->num_rows();
if ($res) {
?>
<script>
alert("Email already exists!");
</script>
<?php
} elseif ($cpassword != $password) { ?>
<script>
alert("Password does not match.");
</script>
```

**signin.php**

This includes part of the signin.php code segment. Here, the system checks that the passenger's email ID and password are correct. If correct, the passenger can login to the system. Otherwise, passengers cannot login to the system.

```
<?php
$cur_page = 'signup';
include 'includes/inc-header.php';
include 'includes/inc-nav.php';
if (isset($_POST['email'])) {
$email = $_POST['email'];
$password = $_POST['password'];
if (!isset($email, $password)) {
?>
<script>
alert("Ensure you fill the form properly.");
</script>
<?php
} else {
//Check for login
$password = md5($password);
$check = $conn->prepare("SELECT * FROM passenger WHERE email = ? AND
password = ?");
$check->bind_param("ss", $email, $password);
if (!$check->execute()) die("Form Filled With Error");
$res = $check->get_result();
$no_rows = $res->num_rows;
if ($no_rows ==  1) {
$row = $res->fetch_assoc();
$id = $row['id'];
$status = $row['status'];
if ($status != 1) {
?>
<script>
alert("Account Deactivated!\nContact The System Administrator!");
window.location = "signin.php";
</script>
```

**reg.php**

This includes part of the reg.php code segment. Here, passengers find their desired schedule, and the system displays the available schedules under date and time.

```
<?php
$row = querySchedule('future');
if ($row->num_rows < 1) echo "<div class='alert alert-danger' role='alert'>
Sorry, There are no schedules at the moment! Please visit after some time.
</div>";
$sn = 0;
while ($fetch = $row->fetch_assoc()) {
//Check if the current date is same with Database scheduled date
$db_date = $fetch['date'];
if ($db_date == date('d-m-Y')) {
//Oh yes, so what should happen?
//Check for the time. If there is still about an hour left, proceed else, skip this data
$db_time = $fetch['time'];
$current_time = date('H:i');
if ($current_time >= $db_time) {
continue;
}
}
$id = $fetch['id']; ?><tr>
<td><?php echo ++$sn; ?></td>
<td><?php echo $fullname =  getRoutePath($fetch['route_id']);
?> </td>
<td><?php $array = getTotalBookByType($id);
echo $max_first = $array;
?></td>
<td><?php echo $fetch['date'], " / ", formatTime($fetch['time']); ?></td>
<td>
<button type="button" class="btn btn-info" data-toggle="modal"
data-target="#book<?php echo $id ?>">
Book
</button>
</td>
</tr>
```

This includes another part of the reg.php code segment. Here, it shows passengers can enter the number of tickets.

```
<div class="modal fade" id="book<?php echo $id ?>">
<div class="modal-dialog modal-lg">
<div class="modal-content">
<div class="modal-header">
<h4 class="modal-title">Book For <?php echo $fullname;
?> &#128653;</h4>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form action="<?php echo $_SERVER['PHP_SELF'] . "?loc=$id" ?>"
method="post">
<input type="hidden" class="form-control" name="id"
value="<?php echo $id ?>" required id="">
<p>Number of Tickets (If you are the only one, leave as it is) :
<input type="number" min='1' value="1"
max='<?php echo $max_first >= $max_second ? $max_first : $max_second ?>'
name="number" class="form-control" id="">
</p>
<p>
Fee: <input type="number" min='1' value="<?php echo ($fetch['fee']); ?>"
name="class" class="form-control" readonly>
</p>
<input type="submit" name="submit" class="btn btn-success"
value="Proceed">
</form>
</div>
```

**verify.php**

This includes part of the verify.php code block. Here, passengers who click on the "Pay Now" button will be redirected to the payment gateway page. It tests with the given parameters. If no error is found, the transaction can be completed successfully.

```
if (isset($_GET['reference'])) {
$email = $_SESSION['email'];
$reference = $_GET['reference'];
$uid = $_SESSION['user_id'];
$pay = curl_init();
$paid = $_SESSION['original'];
$schedule_id = $_SESSION['schedule'];
$number = $_SESSION['no'];
$price = 0;
$amount = $_SESSION['amount'] . "00";
curl_setopt_array($pay, array(
CURLOPT_URL => "https://api.paystack.co/transaction/verify/" .
rawurlencode($reference),
CURLOPT_RETURNTRANSFER => true,
CURLOPT_SSL_VERIFYPEER => 0,
CURLOPT_HTTPHEADER => [
"accept: application/json",
"authorization: Bearer $paystack",
"cache-control: no-cache"
],
));
$response = curl_exec($pay);
$err = curl_error($pay);
if ($err) {
// there was an error contacting the Paystack API
header("Location: individual.php?page=pay&error=payment");
exit();
}
if ($response) {
$result = json_decode($response, true);
if (array_key_exists('data', $result) && array_key_exists('status', $result['data']) &&
($result['data']['status'] === 'success') && ($result['data']['requested_amount'] ===
intval($amount))) {
//confirm access to payment success page
$paid = substr($paid, 0, -2);
$reference = strtoupper($reference);
$ins = $conn->query("INSERT INTO payment (passenger_id, schedule_id, amount,
ref, date) VALUES ('$user_id','$schedule_id', '$paid', '$reference', '$date')");
```

**notify.php**

This includes the notify.php code snippet. Here, it shows once passengers pay for their tickets, they will receive a booking confirmation SMS.

```
<?php
require_once('../notify-php/autoload.php');
$api_instance = new NotifyLk\Api\SmsApi();
$user_id = "24781"; // string | API User ID - Can be found in your settings page.
$api_key = "iXY6Ru3zudZe4f0sXgXK"; // string | API Key - Can be found in your settings page.
$message = "Your Payment is received.Thank you for joining with us"; // string | Text of the message. 320 chars max.
$to = "94704389094"; // string | Number to send the SMS. Better to use 9471XXXXXXX format.
$sender_id = "NotifyDEMO"; // string | This is the from name recipient will see as the sender of the SMS. Use \\\"NotifyDemo\\\
" if you have not ordered your own sender ID yet.
$contact_fname = ""; // string | Contact First Name - This will be used while saving the phone number in your Notify contacts (optional).
$contact_lname = ""; // string | Contact Last Name - This will be used while saving the phone number in your Notify contacts (optional).
$contact_email = ""; // string | Contact Email Address - This will be used while saving the phone number in your Notify contacts (optional).
$contact_address = ""; // string | Contact Physical Address - This will be used while saving the phone number in your Notify contacts (optional).
$contact_group = 0; // int | A group ID to associate the saving contact with (optional).
$type = null; // string | Message type. Provide as unicode to support unicode (optional).
```

**paid.php**

This includes part of the paid.php code segment. Here, after paying the ticket fee, passengers can view booking history and print tickets.

```
<div class="card">
<div class="card-header alert-success">
<h5 class="m-0">Bookings - Purchased Tickets</h5>
</div>
<div class="card-body">
<table class="table table-bordered" id='example1'>
<thead>
<tr>
<th>#</th>
<th>Ticket Number</th>
<th>Trip Date</th>
<th>Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<?php
$conn = connect()->query("SELECT *, booked.id as id, payment.date as pd FROM
`booked` INNER JOIN payment ON booked.payment_id = payment.id INNER JOIN
schedule ON schedule.id = booked.schedule_id  WHERE payment.passenger_id =
'$user_id' ORDER BY booked.id DESC");
$sn = 0;
while ($row = $conn->fetch_assoc()) {
$fullname = getRouteFromSchedule($row['schedule_id']);
$id = $row['id'];
$sn++;
echo "<tr>
<td>$sn</td>
<td>" . $row['code'] . "</td>
<td>" . $row['date'] . "</td>
<td>" . ((isScheduleActive($row['schedule_id']) ? '<span class="text-bold text-
success">Active' : '<span class="text-bold text-danger">Expired')) . "</span></td>
<td>
<button type='button' class='btn btn-primary' data-toggle='modal'
data-target='#view$id'> View   </button>
```

**feedback.php**

This includes part of the passenger's feedback code segment. Here, it shows passengers can send messages and feedback to the administrator. It’s saved in the database.

```
<div class="modal fade" id="add">
<div class="modal-dialog modal-lg">
<div class="modal-content" align="center">
<div class="modal-header">
<h4 class="modal-title">Send New Feedback </h4>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form action="" method="post">
<div class="row">
<div class="col-sm-12">
<div class="form-group">
Type Message : <textarea name="message" required minlength="10" id="" cols="30"
rows="10" class="form-control"></textarea>
</div>
</div>
<hr>
<input type="submit" name="sendFeedback" class="btn btn-success"
value="Send"></p>
</form>
</div>
</div>
```

## Back-end code segment

### bus.php

This includes part of the bus.php code segment. Here, it shows administrators can input bus details into the system. All details are saved in the database.

```
<div class="modal fade" id="add">
<div class="modal-dialog modal-lg">
<div class="modal-content" align="center">
<div class="modal-header">
<h4 class="modal-title">Add New Bus &#x1F68D;
</h4>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form action="" method="post">
<table class="table table-bordered">
<tr>
<th>Bus</th>
<td><input type="text" class="form-control" name="plate_no" required
minlength="3" id=""></td>
</tr>
<tr>
<th>Owner Name</th>
<td><input type="text" min='0' class="form-control" name="bus_owner_name"
required id=""></td>
</tr>
<tr>
<th>Driver Name</th>
<td><input type="text" min='0' class="form-control" name="bus_driver_name"
required id="">
</td>
</tr>
<tr>
<th>Ticket Checker Name</th>
<td><input type="text" min='0' class="form-control" name="bus_tickchecker_name"
required id="">
</td>
</tr>
<tr>
<th>Seat Capacity</th>
<td><input type="number" min='0' class="form-control" name="seat_cap" required
id="">
</td>
```

**route.php**

This includes part of the route.php code segment. Here, it shows administrators can input route details into the system. All details are saved in the database.

```
<div class="modal fade" id="add">
<div class="modal-dialog modal-lg">
<div class="modal-content" align="center">
<div class="modal-header">
<h4 class="modal-title">Add New Route &#128653;
</h4>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form action="" method="post">
<table class="table table-bordered">
<tr>
<th>Route No</th>
<td><input type="text" class="form-control" name="no" required id=""></td>
</tr>
<tr>
<th>From</th>
<td><input type="text" class="form-control" name="start" required id=""></td>
</tr>
<tr>
<th>To</th>
<td><input type="text" class="form-control" name="stop" required id="">
</td>
</tr>
<tr>
<th>Distance</th>
<td><input type="text" class="form-control" name="distance" required id="">
</td>
</tr>
<tr>
<td colspan="2">
<input class="btn btn-info" type="submit" value="Add Route" name='submit'>
</td>
</tr>
</table>
</form>
```

**dynamic_schedule.php**

This includes part of the dynamic_schedule.php code segment. Here, it shows administrators can input schedule details into the system. All details are saved in the database.

```
<div class="modal fade" id="add">
<div class="modal-dialog modal-lg">
<div class="modal-content" align="center">
<div class="modal-header">
<h4 class="modal-title">Add New Schedule &#x1F68D;
</h4>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form action="" method="post">
<div class="row">
<div class="col-sm-6">
Bus : <select class="form-control" name="bus_id" required id="">
<option value="">Select Bus</option>
<?php
$con = connect()->query("SELECT * FROM bus");
while ($row = $con->fetch_assoc()) {
echo "<option value='" . $row['id'] . "'>" . $row['plate_no'] . "</option>";
}
?>
</select>
</div>
<div class="col-sm-6">
Route : <select class="form-control" name="route_id" required id="">
<option value="">Select Route</option>
<?php
$con = connect()->query("SELECT * FROM route");
while ($row = $con->fetch_assoc()) {
echo "<option value='" . $row['id'] . "'>" . getRoutePath($row['id']) . "</option>";
}
?>
</select>
</div>
</div>
<div class="row">
<div class="col-sm-6">
Charge : <input class="form-control" type="number" name="fee" required
id="">
</div>
```

```
</div>
<div class="row">
<div class="col-sm-6">
Date : <input class="form-control" onchange="check(this.value)" type="date"
name="date"
required id="date">
</div>
<div class="col-sm-6">
Time : <input class="form-control" type="time" name="time" required id="">
</div>
</div>
<hr>
<input type="submit" name="submit" class="btn btn-success" value="Add
Schedule"></p>
</form>
<script>
function check(val) {
val = new Date(val);
var age = (Date.now() - val) / 31557600000;
var formDate = document.getElementById('date');
if (age > 0) {
alert("Past/Current Date not allowed");
formDate.value = "";
return false;
}
}
</script>
</div>
</div>
<!-- /.modal-content -->
</div>
<!-- /.modal-dialog -->
</div>
```

**feedback.php**

This includes part of the feedback.php code segment under admin. Here, it shows administrators can read and respond to all passengers' messages. All messages and responses are saved in the database.

```
<div class="modal fade" id="edit<?php echo $id ?>">
<div class="modal-dialog modal-lg">
<div class="modal-content">
<div class="modal-header">
<h4 class="modal-title">Replying to <?php echo $fullname; ?>'s
Message</h4>
<button type="button" class="close" data-dismiss="modal"
aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form action="" method="post">
<input type="hidden" class="form-control" name="id"
value="<?php echo $id ?>" required id="">
<p>Reply : <textarea class="form-control" name="reply" required
minlength="3"></textarea>
<p>
<input class="btn btn-info" type="submit" value="Reply"
name='send_reply'>
</p>
</form>
```

**report.php**

This includes part of the report.php code block. Here, it shows that administrators can generate booking reports.

```
<div class="card-body">
<div class="table-responsive">
<table style="width: 100%;" id="example1" style="align-items: stretch;"
class="table table-hover table-bordered">
<thead>
<tr>
<th>#</th>
<th>Bus</th>
<th>Route</th>
<th>Date/Time</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<?php
$row = $conn->query("SELECT * FROM schedule ORDER BY id DESC");
if ($row->num_rows < 1) echo "No Records Yet";
$sn = 0;
while ($fetch = $row->fetch_assoc()) {
$id = $fetch['id']; ?><tr>
<td><?php echo ++$sn; ?></td>
<td><?php echo getBusName($fetch['bus_id']); ?></td>
<td><?php echo getRoutePath($fetch['route_id']);
$fullname = " Schedule" ?></td>
<td><?php echo $fetch['date'], " / ", formatTime($fetch['time']); ?></td>
<td>
<a href="admin.php?page=report&id=<?php echo $id; ?>">
<button type="submit" class="btn btn-success">
View
</button>
```

# Appendix D

**Questionnaire (Passengers)**

## Introduction

The web-based bus ticketing system initiative is similar to many other web-based bus ticketing systems around the world. A web-based bus ticketing system is designed to create paperless ticketing and also ease traffic congestion in bus stations.
The questionnaire is divided into a few sections, from general questions to more specific questions, such as Section I on background, Section II on experience with internet usage, Section III on experience with purchasing on an existing web-based bus ticketing system, and Section IV on suggestions.

## Section i: Background

1. What age group are you?

☐ Below 20

☐ 20-25

☐ 26-35

☐ 36-45

☐ Above 45

2. What is the reason you are in Colombo center bus stand today?

☐ To purchase a bus ticket

☐ Waiting for a bus

☐ Waiting for a passenger

☐ Others (please specify) _____

3. Have you purchased an over the counter bus ticket before?

☐ Yes

☐ No *(Please proceed to question 5)*

4. If yes, what is the average time do you take to purchase a bus ticket (including coming to the

☐ bus station, parking, etc)?

☐ 30 minutes and below

☐ 30 minutes to 45 minutes

☐ 45 minutes to 1 hour

☐ Above 1 hour

Please rate the condition of Colombo central bus stand in

the table below. Please tick ( ✓) your answer.

*Appendix C 1: Rate of condition of CCB stand*

| | **Very Good** | **Good** | **Medium** | **Bad** | **Very Bad** |
|---|---|---|---|---|---|
| Human Congestion | | | | | |
| Air Environment | | | | | |
| Self Safety | | | | | |
| Traffic | | | | | |
| Convenience to Purchase Tickets | | | | | |
| Travel/ Bus Information Acquiring | | | | | |

## Section ii: Experience on internet usage

5. Do you have a computer that has access to the Internet?

☐ Yes, specify (home/office)

☐ No *(Thank you for answering the questionnaire)*

6. How long do you normally surf the Internet per week?

☐ < 2 hours

☐ 2-4 hours

☐ 5-6 hours

☐ hours

7. Why do you use the Internet for most of the time? (You may choose more than one answer)

☐ General information retrieval

☐ Shopping

☐ Banking

☐ Email

☐ Search for specific information

☐ Other (please specify) _____

8. Have you ever conducted an Online Payment through the Internet before?

☐ Yes

☐ No

## Section iii: Experience on purchasing an online bus ticket

9. Have you heard of Web based Bus Ticketing System before?

☐ Yes

☐ No *(Please proceed to question 17)*

10. Are you aware of Web based Bus Ticketing System being provided by bus operators in Sri Lanka?

☐ Yes

☐ No *(Please proceed to question 17)*

11. Have you used this Web based Bus Ticketing System to purchase a bus ticket before?

☐ Yes *(Please proceed to question 14)*

☐ No *(Please proceed to the next question& thank you for answering the questionnaire)*

12. Please choose the reason why you have never used an online system to purchase a bus ticket before. You may choose more than one answer.

☐ Do not trust online bus ticketing systems

☐ Do not trust online payment

☐ Lack of information

☐ Not user friendly

☐ Slow page download (Slow server)

☐ Lack of multimedia elements / No direct interactivity

☐ Not enough information

☐ Lack of security

☐ Lack of understanding of e-commerce

☐ Others (please specify) _____

Please choose the best relating to the existing Online Bus Ticketing system.

*Appendix C 2: Review of online bus ticketing system*

| Do you agree that: | **YES** | **NO** |
|---|---|---|
| Online bus ticketing is more efficient compared to buying a bus ticket over the counter | | |
| Online bus ticketing is user friendly | | |
| Online bus ticketing web sites are informative | | |
| Online bus ticketing is cost and time effective | | |
| Online bus ticketing is more reliable | | |
| Online bus ticketing payment is convenient | | |

13. Are you satisfied with the services provided in these web sites?

☐ Yes

☐ No

14. Will you recommend purchasing an online bus ticket to your friends?

☐ Yes *(Please proceed to the next question)*

No If No, why?

☐ Do not trust online bus ticketing systems

☐ Do not trust online payment

☐ Lack of information

☐ Not user friendly

☐ Slow page download (Slow server)

☐ Lack of multimedia elements / No direct interactivity

☐ Not enough information

☐ Lack of security

☐ Lack of understanding of e-commerce

☐ Others (please specify) _____

## Section iv: Suggestions

15. Do you normally use one type of bus operator only for your traveling purpose?

☐ Yes (Please proceed to the following question)

☐ No (Please proceed to question 19)

16. Why do you only use the services of one type of bus operator to your destination?

☐ There is only one bus operator to my destination

☐ Difficult to search for tickets from other bus operators

☐ Cheaper

☐ The bus operator has an Online Ticketing system

☐ Others (please specify) _____

17. If there is a portal combining all express bus operators into a single system, do you think it will be a better than the current method of purchasing a bus ticket?

☐ Yes

☐ No

18. If there is a portal combining all express bus operators into a single system, will you use this portal?

☐ Yes

☐ No

19. Which payment option would you prefer when purchasing an online bus ticket from this portal?

☐ Credit card

☐ Prepaid system

☐ E-payment (Direct Banking)

☐ Debit card

☐ E-wallet

☐ Smart card

☐ Others (please specify) _____

20. What are the features do you feel should be available in this Online Bus Ticketing system?
You may choose more than 1 answer.

☐ Attractive Web Page Design

☐ Fast Loading And Response Time

☐ User Interactivity

☐ Online Registration

☐ Online Payment

☐ Online Reservation/ Purchasing

☐ Online Cancellation

☐ Print Out Reservation

☐ Company Information

☐ User Instruction

☐ FAQ/ Help section

☐ Schedule

☐ Fare Rate

☐ Terminal Location Map

☐ Seat Map

☐ Seat Status

☐ Other Web Site Linkage_____

21. How would this ranking help you as a consumer?
You may choose more than 1 answer.

☐ To be able to choose the best service provider

☐ Awareness of bus operators sales performance

☐ Safety and security reasons

☐ Confidence in services provided

☐ To get the best deals (value for money)

☐ Others (please specify) _____

22. What criteria do you think should be included to determine the classification or rank of a bus operator? You may choose more than 1 answer.

☐ Bus operator's sales performance

☐ Popularity through public polling system

☐ Accidents and traffic offences track record

☐ In-bus service, example, meal serving, cable tv, internet access, etc.

☐ Cheaper bus ticket

☐ Others (please specify) _____

---

Thank you for answering this questionnaire.

# Interview Questions for Online Bus Ticketing System (Bus Operators)

Name:

Organization Name:

Position:

Objectives:

To understand the operations of the ticketing department in an express bus company. To analyze the acceptance, potential, and opportunities of implementing an online bus Ticketing System.

1. How does your company conduct bus ticket sales, bookings, and purchases? process? Could you please explain in more detail?
2. Will there be any reports or statistics generated daily?
3. How is the bus schedule planned?
4. Does your company have a fixed schedule for the bus, or is it changed regularly?
5. Does your company appoint agents to sell your tickets? If yes, how many of them?
6. Is your staff computer literate, and are they using computers to do their daily duties?