IS 2900 – Project on IT Applications

Interim Report

Cloud-Based Bus pass system

Team Dominators

Index No	Name
205119P	Wijethilaka W.G.D.C.D.
185067A	Rosni H.F.
205063L	Madhushika D.S.
205091U	Sadhoon M.I.A.
205016X	De Silva R.M.M.

Supervised by:

Dr. S.C. Premaratne

Mrs. M.B. Mufitha

Client:

Virtusa Pvt.Ltd.

Dr. Danister De Silva Mw,

Colombo 09.

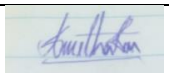
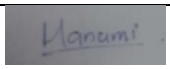
Faculty of Information Technology

University of Moratuwa

2022

Declaration

We declare that this report is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Names of Students	Signatures of Students
Wijethilaka W.G.D.C.D.	
Rosni H.F.	
Madhushika D.S.	
Sadhoon M.I.A.	
De Silva R.M.M.	

Date:

Supervised by:

Names of Supervisors Signatures of Supervisors

Dr. S.C. Premaratne

Date:

Mrs. M.B. Mufitha

Date:

Abstract

Cloud-based Bus pass system is a web application that allows users to reserve bus tickets via an online platform. The system is planned to implement the centralization of one public transportation center including several long bus routes. Passengers can log into the system online and reserve the ticket and the administrator can control all the bus schedules online. As it takes a huge amount of labor and expense when all the processes are done manually, this smart system reduces the expense and gives efficient and effective service to the passengers.

In this system, we have initialized 5 actors as Admin, Timekeeper, Passenger, Bus Owner, and Conductor who interact with each other. Admin controls the whole system providing services to every other role. Bus owners can register to the system and register their buses into the system. They obtain n income from buses at the end of the journey. Then the timekeeper schedules timetables for every bus and inspects journeys including emergency management. Passengers can purchase tickets and payments to get the smart bus pass including a QR code. Then the conductor scans the QR code to identify the passengers and updates the system. Registrations, Bus pass generation, and money transactions will happen by the Backend server.

We analyze clients' requirements and design the system UI and implement the backend using technologies like Nodejs, React, MySQL, Strip Sandbox API…etc. We have divided the scope among team members into 5 modules according to the 5 actors of the system and they are connected to each other via several gateways.

1. Registration/Login Module
2. Authentication and authorization
3. Bus pass generation
4. Money transaction
5. Sending notification
6. Timetable management
7. Inquiry handling

Table of Contents

List of Figures/Table

Chapter 1 Introduction

1.1 Introduction

The cloud-based Bus Pass Management System is a real-time project that will benefit passengers who are dissatisfied with the existing manual bus pass system. It allows passengers to travel more easily by scanning the ticket QR code with their smart devices. Passengers can purchase bus tickets beforehand for a week at any time online 24 hours a day. Seven days a week, which eliminates the problem of forgotten or stolen bus tickets. And it provides the facility of ticket reselling and time keeping processes in a smart and efficient way to the administration.

1.2 Problem in Brief

Nowadays, due to the scarcity of fuel, there is a huge tendency of using public transportation systems rather than personal vehicles. Traveling by bus is the easiest and most affordable way of all public transportation methods. But there are so many difficulties of picking a bus like unawareness of Timetables of certain days, the inability to pick a seat, unawareness of the bus fares because of the rising prices…etc. But we do not have even a manual method of booking a bus seat before the journey. There is a requirement of having a smart method to book a seat on a bus and enjoy the journey. On the administration side as well, it requires a large staff to manage the timekeeping, handling of emergencies, and admin processes that cost higher.

1.3 Aim and Objectives

1.3.1. Aim

Providing safe, dependable, affordable, cost-effective, and efficient service to passengers, easy registration, and a better profit to the bus owner, and easy administration and managing space for the administration.

1.3.2. Objectives

- Allowing users to book their seats conveniently and facilitating 24/7 services to passengers helping to avoid fraud during transactions and giving safety.
- Providing an effective payment method with safeguard infrastructure to users.
- Providing Inquiries sections for passengers for users to put forward their problems.
- Facilitating the location tracking feature
- Easing the ticketing process and seat distribution to the conductors and notifying the user about seat availability.
- Reducing the paperwork (manual bookkeeping) and reducing the staff, which reduces the expenses.
- Managing schedule effectively and avoiding bus delays and managing emergencies
- Ability to manage many buses concurrently easily keeping a database and reselling tickets

1.4 Proposed Solution

- **Registration/login module** for 5 primary actors as Admin, Timekeeper, Conductor, Bus Owners, and Passenger with different levels of permissions. Users can first enter their Gmail address and password for the authentication **and authorization module** to ensure the security of the system by verifying the user. After verification, the users can enter their details into the system and register to the system under the supervision of the admin. Bus owners have to register to the system successfully and after that, they can register their buses to the system for the purpose of earning profits through journeys. Passengers can also verify their account first as mentioned above and then they can register to the system by creating accounts. Conductors are given a username and password after registration through the administrator, and they can log in to the system. Since the timekeeper and the admin are pre-initialized, they log into the system and provide services to other users.
- Develop an **online bus ticket booking, canceling, and renewal system** to check the availability of the buses according to the route, be aware of the price, and purchase the ticket for a seat for the particular journey. Passengers can book seats at his/her required time and route easily and effectively and with a **fare calculation module** for

calculating bus fares in a standardized manner for particular distances. They get a QR code with the smart bus pass **generation module.** It easily verifies the passenger to the conductor and passengers can use the **e-wallet system** for effective money transactions to purchase the bus pass.

- **SMS notifications module** to inform the emergency cases, starting and reaching time and confirmation, etc. passengers. When the bus journey starts and ends an SMS notification is sent to the passenger for awareness and as well as when a delay occurs while the journey starts. And when a breakdown of the bus occurs during the journey the conductor can inform to timekeeper attaching the location. Then the timekeeper can check the closest depo by the database and inform them to allocate another bus as soon as possible.
- We are planning to develop a **time management module** to handle the time schedules of buses, assigning buses to a timetable, timekeeping, and assigning buses for emergency cases. These things will be done by the timekeeper. He schedules timetables for every route and allocates buses. in emergency cases of bus breakdown or heavy traffic, the timekeeper takes proper action to reallocate buses. Reselling tickets also happened under the supervision of the timekeeper. When a passenger cancels the ticket the bus fare is refunded 75% of the total and the ticket is resold to another customer.
- Develop an **online inquiry system** to get feedback and provide solutions by the administration. Passengers can get feedback about buses, conductors, and everything else. So, considering those comments the admin can take proper actions for the sustainability of the system. Admin is able to delete accounts in inappropriate situations or special occasions like duplicated accounts, fake accounts…etc. Admin has the authority to provide services and manipulate all other accounts.

Table 1.4.1 *Actors and the descriptions*

Role	**Activity**	**Input**	**Output**
Admin	• Login to the system • View activity records	• Username, Password • Updating new data	• System Management • Report supervision

	• Inquiry supervision • Update and delete accounts • provide services to other roles • User registration handling		• Interacting with users in problematic situations
Timekeeper	• Login to the system • Schedule timetables • Allocate and reallocate buses • Reselling tickets • Updating about emergencies • Update bus fares and route details • Marking starting time and ending time of journeys.	• Username, Password • Updating new data	• Contacting depos • Reallocating buses • Informing emergencies to the users
Bus Conductor	• Registering Login to the system • Scan QR and verify passenger • Informing emergencies and delays and Sharing location.	• Username, Password • QR code	• Sharing location • Informing emergencies
Passenger	• Registering Login to the system • Searching for Buses and seats • Booking seats and purchasing tickets	• Username, Password • Personal details to register • Bank account details	• Transferring money • Feedback and ratings
Bus Owner	• Registering Login to the system	• Username, Password	• Transferring money

	• Registering buses to the system • Check income reports	• Personal details and bus details • Bank account details	• Feedback and ratings

1.5 Structure of the Report

Chapter 1: Introduction to the system, Aim, Objectives, Problem, and the Solution
Chapter 2: Comparing the similar existing systems with our system
Chapter 3: Modules of our system and the Technologies we are using to implement them
Chapter 4: Analyzing user requirements using UML diagrams
Chapter 5: Implementations we have done so far in the system
References: Resources we used to study
Appendix A: Individual contribution of each member for the project
Appendix B: Time plan for the rest of work
Appendix C: Draft UIs
Appendix D: SRS Document

1.6 Summary

The first chapter of the report provides a brief introduction including an Introduction to the system, the Problem based on the proposal of the project, the Aim, and Objectives of the project, Proposed solution. Chapter 2 focuses on the Literature review including the similar projects to our project comparing the similarities and dissimilarities. Chapter 3 discusses our approach to fulfilling the user requirements including the technologies we use for the implementation. Chapter 4 simply describes how the user requirements are analyzed and the system is designed using Diagrams in the SRS. In Chapter 5 we have discusses the implementation up to now including flowcharts, algorithms, pseudo codes…etc. After that, we discussed the resources we referred for the implementation and the individual contribution.

Chapter 2 Literature Review

2.1 Introduction

In this Chapter, we are mainly concerned with discussing the different approaches to addressing the same problem for which we have chosen to propose a solution. When we consider the Sri Lankan bus system, we can apply IT knowledge to it. It will also help to reduce the paperwork (manual bookkeeping) and reduce the staff, which reduces the expenses. Not only that we can also Allow users to book their seats conveniently and facilitate 24/7 services to passengers helping to avoid fraud during transactions and giving safety, provide an effective payment method with safeguard infrastructure to users, provide Inquiries sections for passengers for users to put forward their problems, Facilitating the location tracking feature, Easing the ticketing process and seat distribution to the conductors and notifying the user about seat availability. As well as we can Managing schedules effectively and avoid bus delays and manage emergencies, ability to manage many buses concurrently easily keep a database and resell tickets. (The cloud-based Bus Pass Management System) is a real-time project that will benefit passengers as well as the owners of the buses, who are dissatisfied with the existing manual bus pass system. It allows passengers to travel more easily by scanning the ticket QR code with their smart devices. Passengers can purchase bus tickets beforehand for a week at any time online 24 hours a day. Seven days a week, which eliminates the problem of forgotten or stolen bus tickets.

2.2 Similar Products

1.RedBus
2.AbhiBus

2.2.1. RedBus

India's largest company for bus ticket booking, redBus, offers an easy-to-use online platform, thousands of bus operators to choose from, and plenty of offers on bus ticket

booking to make road journeys super convenient for travelers. A leading platform for booking bus tickets, redBus has driven the country's bus booking journey over the past **16+ years** through thousands of bus operators and routes. Striving to reach new heights when it comes to online bus reservations in India, redBus has become the perfect tool to use to have a smooth bus ticket booking experience.

2.2.2. AbhiBus

AbhiBus is India’s fastest-growing online ticket booking platform offering bus ticket booking, train ticket booking, and bus rental services. AbhiBus is the official ticketing partner of several State Road Transport Corporation (SRTC) operators, and you can also book tickets for your favorite private travels at discounted ticket prices. Travelers can also book IRCTC train tickets without any hassle-free services on AbhiBus.

AbhiBus has an inventory of over 2,500+ bus service partners operating scheduled buses in India. Travelers can easily view their bus schedules, compare ticket prices, and book bus tickets online through AbhiBus. All these buses currently cover more than 100,000 bus routes with reliable transportation in India.

2.3 Comparison between other systems

Table 2.3.1 *Features comparison with existing systems*

Features	**RedBus**	**2.3.1.** AbhiBus	**Our System**
QR code (verify the passenger Payment)	No	No	Yes
Time Keeping	No	No	Yes
Cancel & Refund	Yes	Yes	Yes
Share Location	Yes	Yes	Yes
Providing Discount	Yes	No	Yes
Manage emergency cases & provide solutions	No	No	Yes
Inquiry Sections	No	No	Yes

2.4 Summary

The online bus pass system is a common application used around the world. According to the above research, there are similar features in each system. Also, some features in our system are not included in the above two software such as Inquiry Sections, Timekeeping, managing emergency cases & provide solutions. The next chapter will give details about technologies that we adapt to solve the problem

Chapter 3

3.1 Introduction

In this chapter, we will discuss the way we designed our system and what are the technologies that we are going to use to implement the project. We are going to use 3 tier architecture for our system. A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage, and user interface are developed and maintained as independent modules on separate platforms. It is a software design pattern and a well-established software architecture. 03 tiers will be the presentation tier which is the user interface, the application tier where data is processed, and the data tier where the data associated with the application is stored and managed. End-users of the system are interacting with the presentation tier. This is usually the website where the user adds a purchase order, adds a sales order, and generates reports. In our system, the presentation tier is going to develop using ReactJS. In the application tier, information collected in the presentation tier is processed sometimes against other information in the data tier using business logic, a specific set of business rules. The application tier is developed using NodeJS. In the data tier, we are going to use MySQL Server as the database

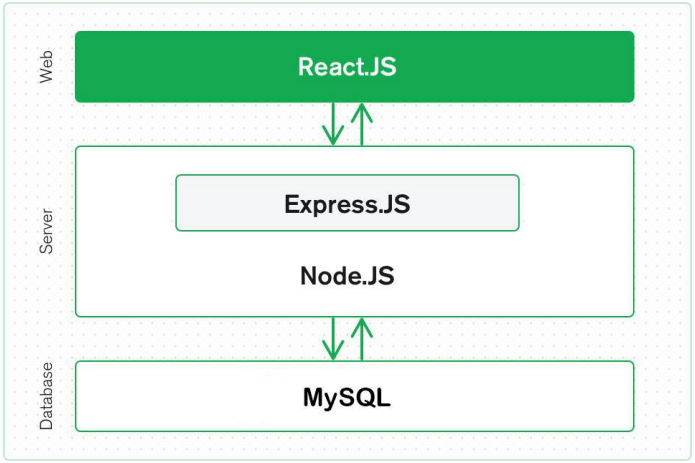


Figure 3.1 *System architecture*

3.2. Technologies.

This chapter focuses on the technologies that are used to develop the Cloud-Based Bus Pass System. Major components of the system consist of and, the technologies used under those components are as follows.

1. Frontend Technology-
    - ReactJS
2. Backend Technology
    - NodeJS
    - Express
3. Database Technology -
    - MySQL
4. Version Controlling System –
    - GitHub
5. Payment Gateway Integration-
    - Strip sandbox API
6. Email Notification-
    - SMTP Server
7. SMS Notification-
    - Twilio's APIs
8. Location Tracking-
    - The Google Maps API
9. Cloud Provider-
    - AWS Hosting

3.2.1. Justification of the Technologies adapted

3.2.1. 1. Frontend Technologies -ReactJS

An open-source JavaScript frontend library called React is used to create user interfaces or UI components. Facebook created and maintained it. It functions as a single-page web application. We use ReactJS as frontend technology in our project because it is simple to create responsive web pages, simple to maintain the code, allows components to be reused depending on the situation, is lightweight compared to other frontend frameworks, and is more versatile than AngularJS. Any JavaScript developer who wants to learn more about developing systems may learn the fundamentals of ReactJS. In order to provide the design of an online application in an ideal manner, Material-UI was used as the web UI component and CSS styling.

3.2.1. 2. Backend Technologies -NodeJS and Express

- A server-side platform based on the V8 engine is called NodeJS. (The JavaScript Engine in Google Chrome), an open-source, multi-platform runtime environment It is employed to create scalable, quick, and light applications. Since we are going to develop a data-driven application, the most appropriate backend technology is NodeJS. Due to its event-driven, non-blocking I/O paradigm, NodeJS is effective and quick when building the backend and is compatible with ReactJS. Users of the Cloud-Based Bus Pass system can use it quickly as a result. Building web applications using NodeJS is simple for inexperienced developers like us. It only uses one free codebase.

- Express is an open-source web application framework for NodeJS. Express is used to design and build web applications easily and quickly. Since it requires only JavaScript, it is easy for us to build the application without any effort.

. **3.2.1.3. Database Technologies -MySQL**

MySQL is an open-source relational database management system based on Structured Query Language. It is set up for great performance and dependable data protection. We selected MySQL as the database server since the Cloud-Based Bus pass system has a sophisticated database system and relationships that need to be considered. It will be tough to run the system without relations. MySQL provides great security measures to guarantee complete data safety. It has strong procedures that allow unauthorized users to be banned at the client-machine level and ensure that only authorized users have access to the database server. As the Cloud-Based Bus Pass system saves essential company data and information, it is crucial to provide secure connections.

3.2.1.4. Version Controlling System -GitHub

We use GitHub version controlling system to collaboratively work on code. It allows for seamless collaboration without compromising the integrity of the original project. of the original project.

3.2.1.5. Payment Gateway Integration-Strip sandbox API

Stripe integrations handle complicated processes. The API uses a single object to track each process. You create the object at the start of the process, and after every step we can check its `status` to see what needs to happen next. For instance, while completing a payment, a customer might try several payment methods. If one payment method fails, a status of required payment method to prompt the customer for another.

3.2.1.6. Email Notification-SMTP Server

An SMTP Protocol or Server Mail Transfer Protocol is a set of rules for Digital Communication. Electronic mail transmission and the SMTP servers are the applications whose main function is to send, receive or Switch between outgoing mail senders and receivers.

3.2.1.7. SMS Notification-Twilio's APIs

Twilio's APIs (Application Programming Interfaces) power its platform for communications. Behind these APIs is a software layer connecting and optimizing communications networks

around the world to allow your users to call and message anyone, globally. Twilio has a whole host of APIs, from SMS to Voice to Wireless.

3.2.1.8. Location Tracking-The Google Maps API

The Google Maps API is one of those clever bits of Google technology that helps you take the power of Google Maps and put it directly on your own site. It lets you add relevant content that is useful to your visitors and customize the look and feel of the map to fit with the style of your site.

3.2.1.9. Cloud Provider-AWS Hosting

Web hosting is a service that stores your website or web application and makes it easily accessible across different devices such as desktop, mobile, and tablets. We select AWS because Amazon Web Services offers wide range of website hosting options with low-cost ways to deliver their websites

3.3. Software Process Model

A software process model is a collection of organized steps used to create software systems. To achieve a goal, it specifies who is doing what, when, and how. An abstraction of the process being described is a software process model. Another way to describe it is as a condensed version of a software process. We adopted the Agile Scrum model as the software process model for our system.

● We are keeping the client updated about our project every two weeks.

● Sometimes, the client's requirements are changing from time to time. So, we can be aware and flexible with the client's requirements and ideas.

● Doing the development and operation at the same time reduces the operational costs as well.

● During our weekly meetings, we identify project issues and get project updates. And also we discussed those issues with our academic supervisors and mentor. Because it is important to provide regular feedback to our academic supervisors and mentor.

3.4. Project Management Plan

We follow a project management plan to approach a successful product. It is a formal approved document that defines how the project is executed, monitored, and controlled. We used Trello

software to manage our project in one place and we are working according to our project management plan. Using Trello we assign tasks among the group members, set due dates, and share documents. We create our task board with the columns such as To Do, In Progress, and Done.

Table 3.4.1 *Network Diagram*

Task ID	Description	Dependency	Duration (Week)
A	Identify the requirements	-	3
B	Gathering & analysis requirements	A	2
C	Allocate task	B	2
D	Create SRS document	C	2
E	Design ER diagram	D	1
F	Design static UML diagrams	E	1
G	Design dynamic UML diagram	E	1
H	Design UI and wireframes	F, G	2
I	Preparing Interim report	H	1
J	Backend development phase 1	I	2
K	Backend development phase 2	J	1
L	Create databases	H	2
M	Frontend development	K	1
N	Implement security functions	M, L	1
O	Implement business logic	N	3
P	Create web interfaces 1	F	2
Q	Create web interfaces 1	G	2
R	Connect frontend with backend	O, P	1
S	Test	R	2
T	Initialize deployment environment	S	2
U	Deploy the beta release	T	1
V	Test beta release	U	2
W	Debug errors	V	1
X	Deploy final product	W	11

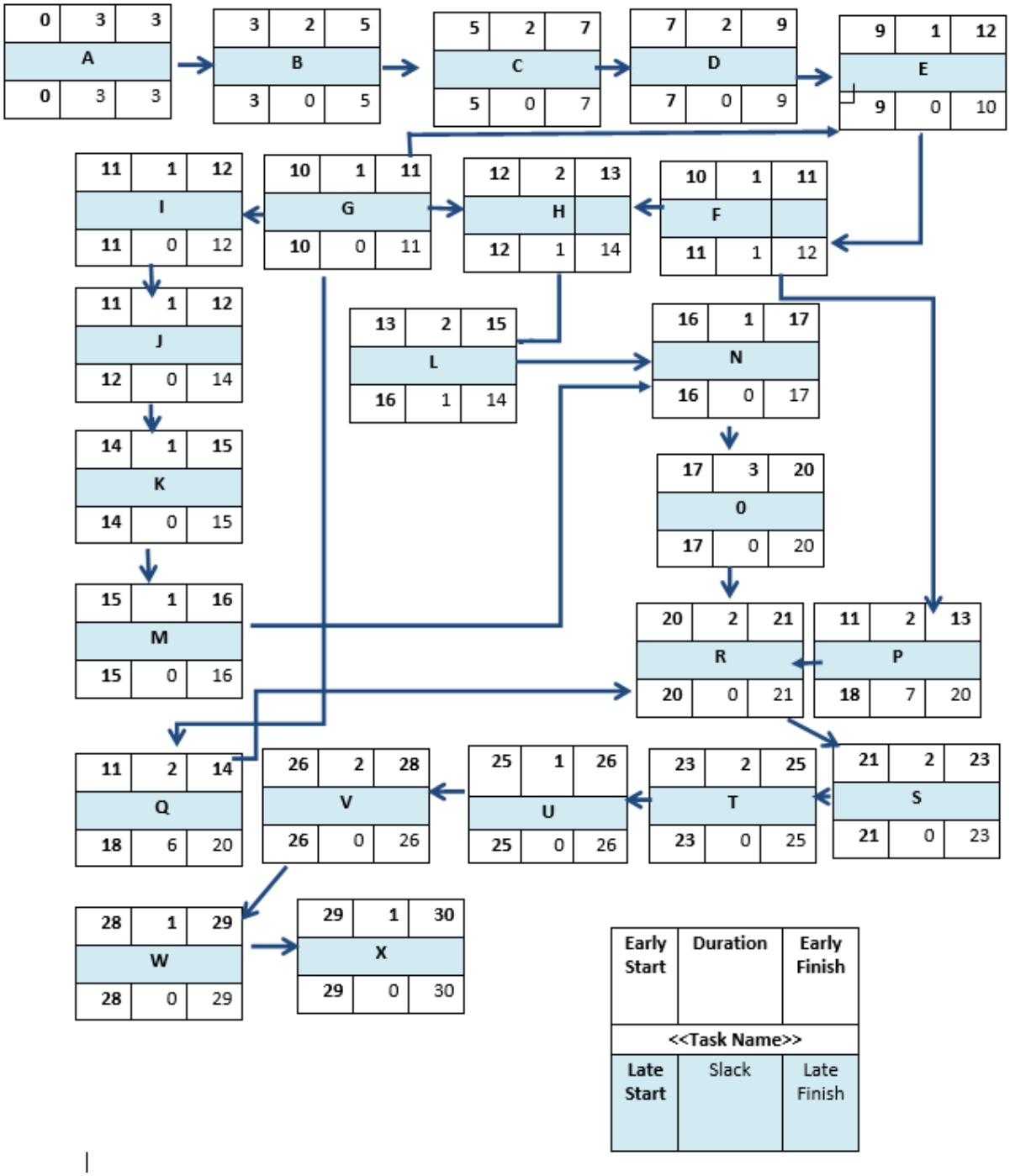
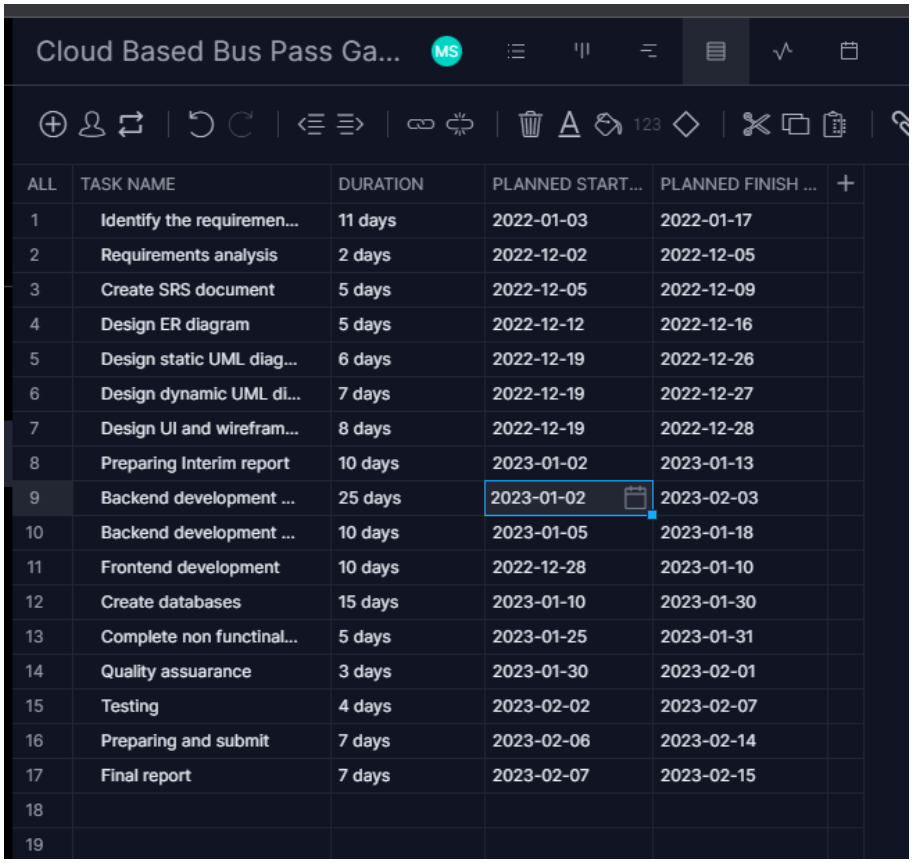


Figure 3.4.1 *Network Diagram*



Cloud Based Bus Pass Ga... MS

ALL	TASK NAME	DURATION	PLANNED START...	PLANNED FINISH ...
1	Identify the requiremen...	11 days	2022-01-03	2022-01-17
2	Requirements analysis	2 days	2022-12-02	2022-12-05
3	Create SRS document	5 days	2022-12-05	2022-12-09
4	Design ER diagram	5 days	2022-12-12	2022-12-16
5	Design static UML diag...	6 days	2022-12-19	2022-12-26
6	Design dynamic UML di...	7 days	2022-12-19	2022-12-27
7	Design UI and wirefram...	8 days	2022-12-19	2022-12-28
8	Preparing Interim report	10 days	2023-01-02	2023-01-13
9	Backend development ...	25 days	2023-01-02	2023-02-03
10	Backend development ...	10 days	2023-01-05	2023-01-18
11	Frontend development	10 days	2022-12-28	2023-01-10
12	Create databases	15 days	2023-01-10	2023-01-30
13	Complete non functinal...	5 days	2023-01-25	2023-01-31
14	Quality assuarance	3 days	2023-01-30	2023-02-01
15	Testing	4 days	2023-02-02	2023-02-07
16	Preparing and submit	7 days	2023-02-06	2023-02-14
17	Final report	7 days	2023-02-07	2023-02-15
18				
19				

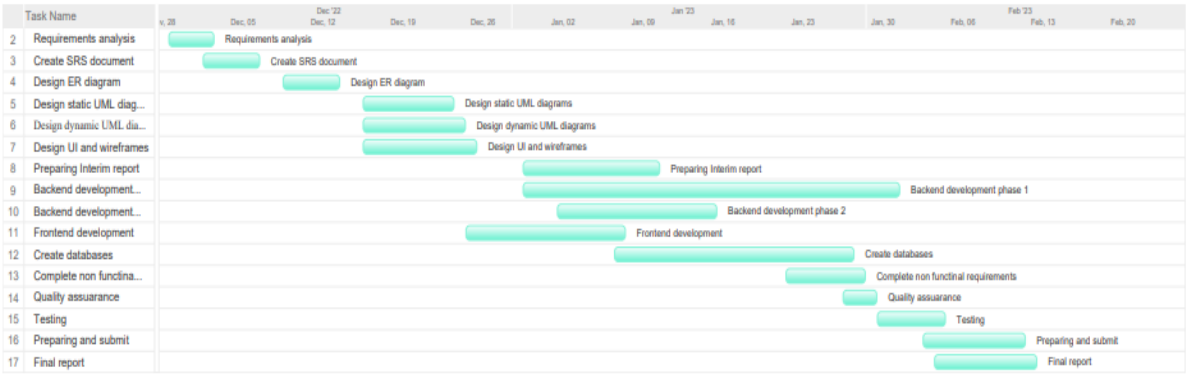


Figure 3.4.2 *Gantt Chart*

Chapter 4 Analysis and Design

4.1 Introduction

After determining the issue, we gathered our client's requirements and began formulating a solution. We created some UML diagrams to help us get a better understanding of the system. The structural and behavioral diagrams were both created by us.

4.2 Analysis

We determined the system's functional and non-functional needs before creating it, as our client mentioned. The SRS document lists the functional and non-functional requirements.

4.3 Design

We will concentrate on the diagrams that we created to represent the functional and non-functional needs in this chapter. We created the diagrams in Microsoft Visio. We have used the following UML diagrams

1. Usecase Diagram
2. Activity Diagram
3. Class Diagram
4. Sequence Diagram
5. ER Diagram

4.3.1. Usecase Diagram

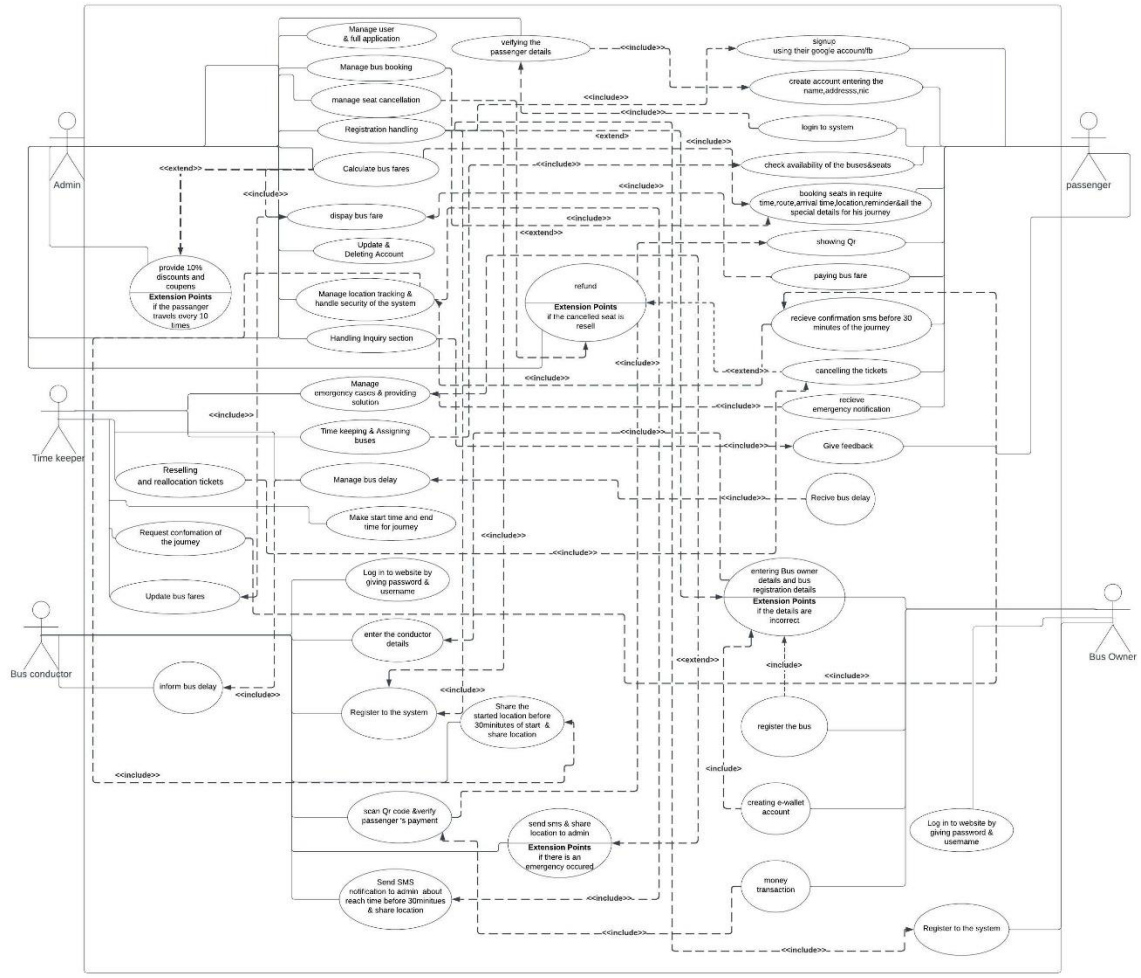


Figure 4.3.1 *Usecase Diagram*

4.3.2. Activity Diagram

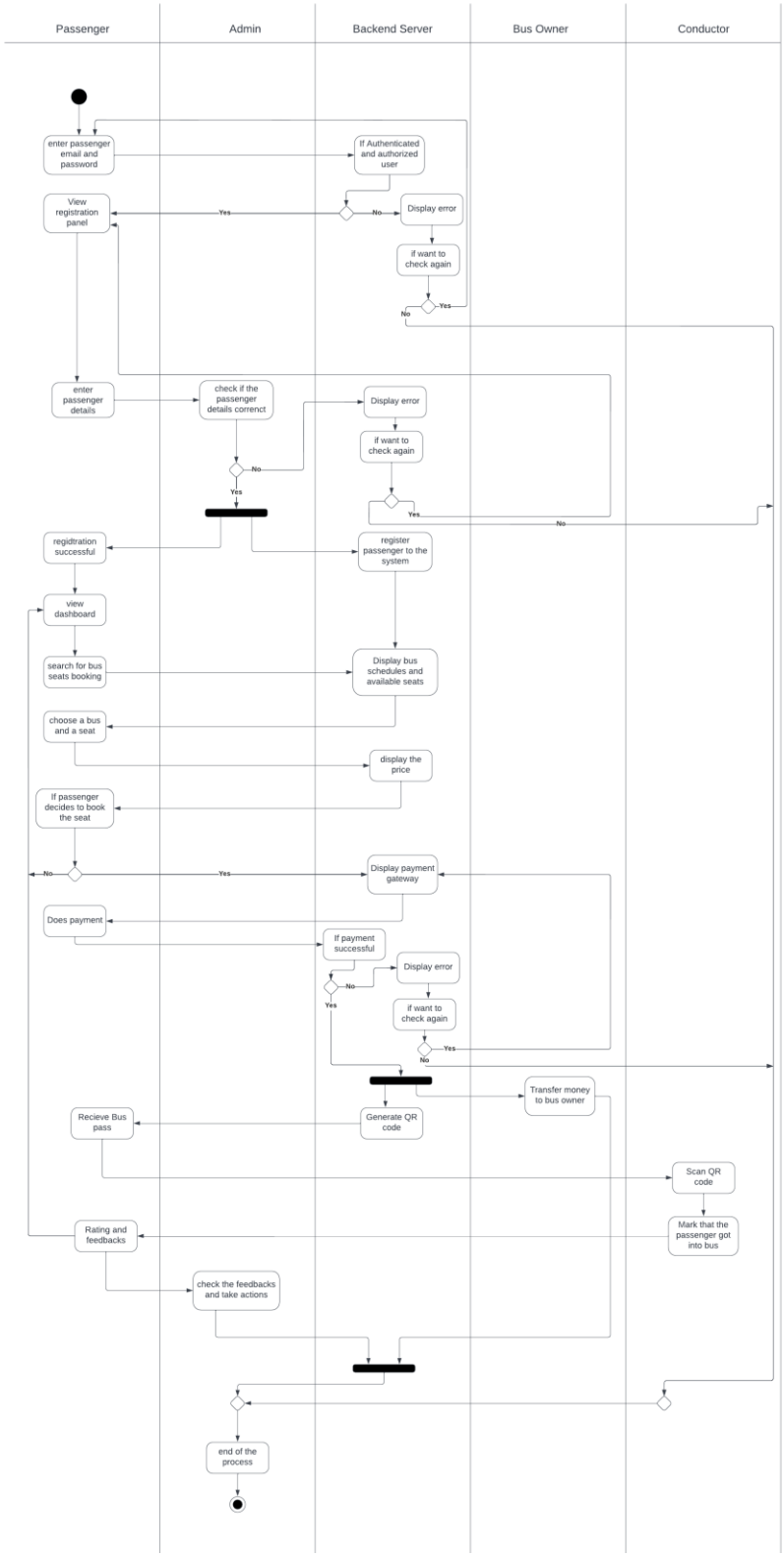


Figure 4.3.2 *Activity Diagram*

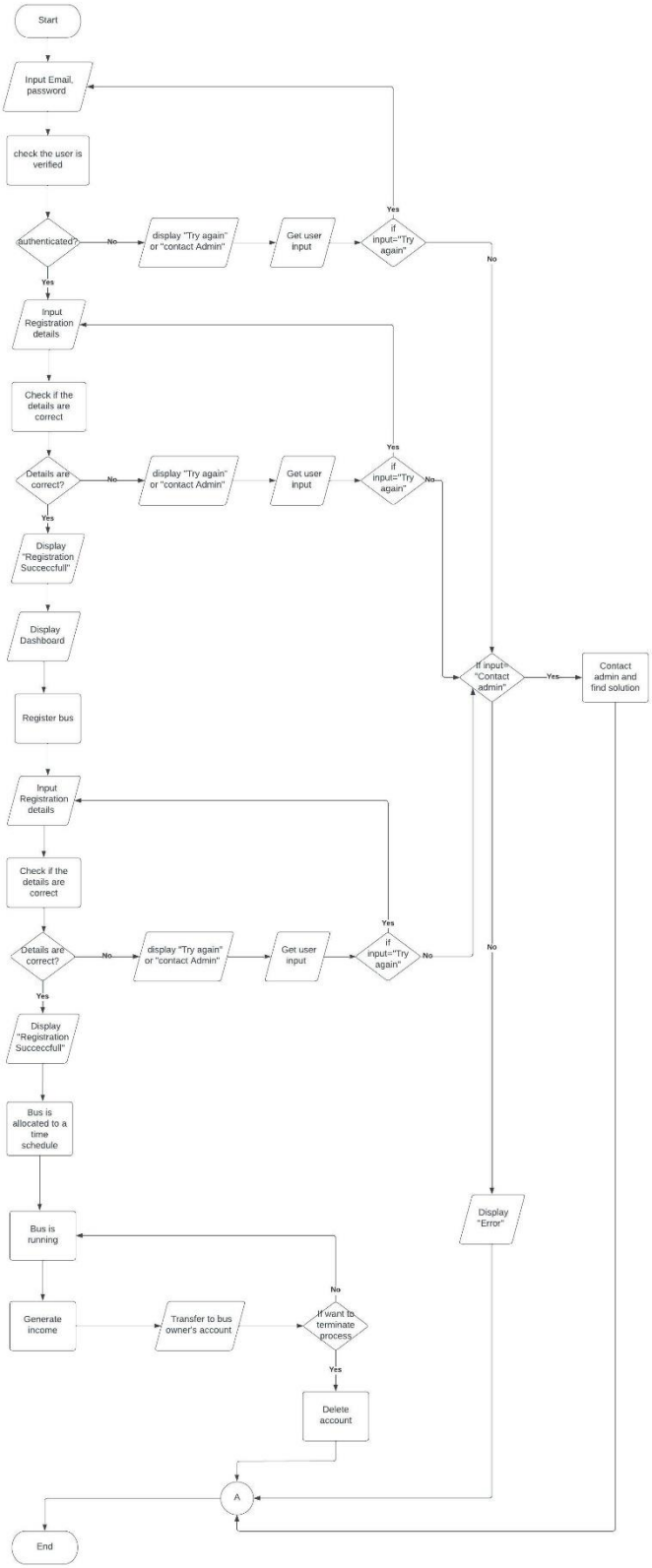


Figure 4.3.2 *Activity Diagram*

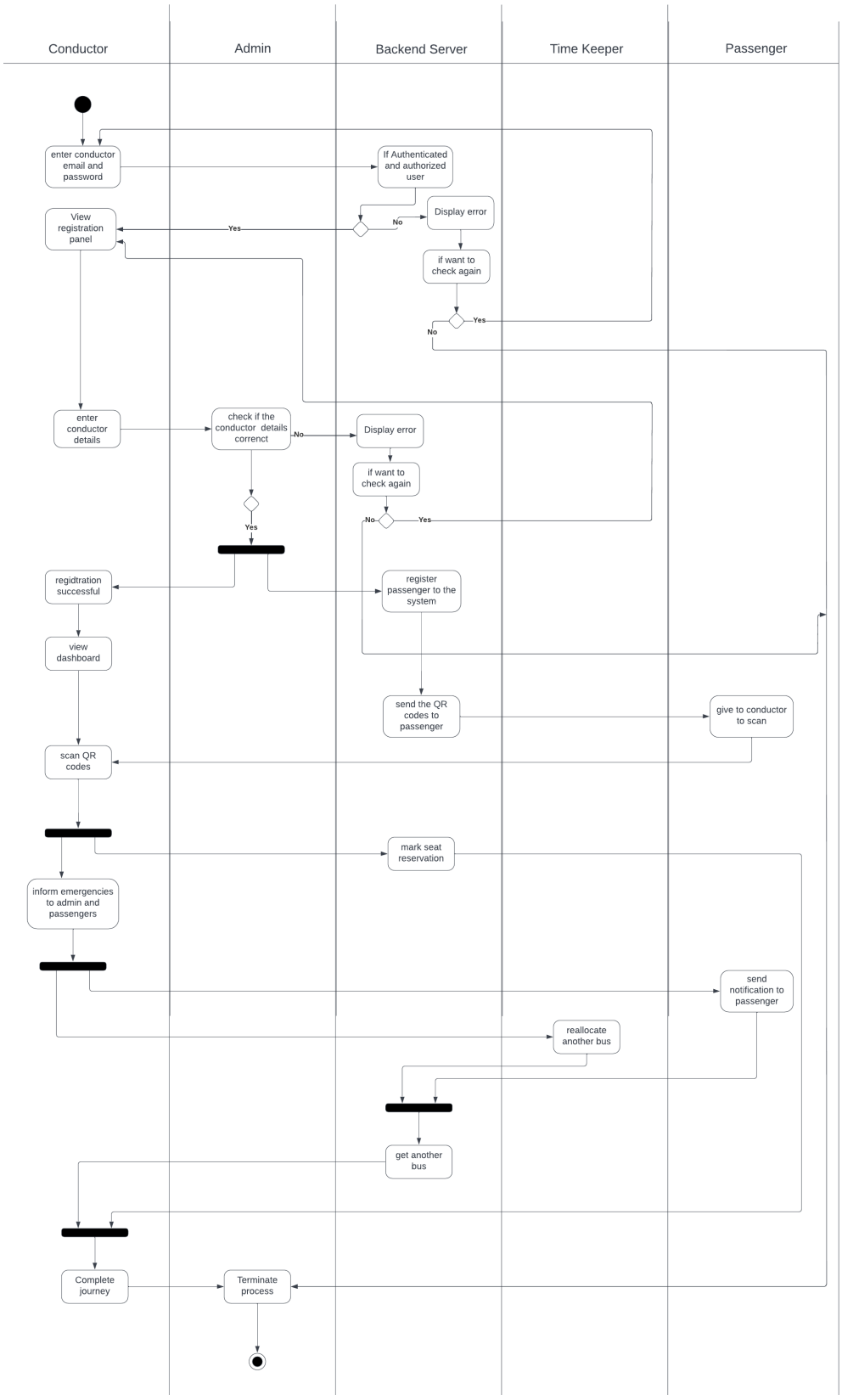


Figure 4.3.2 *Activity Diagram*

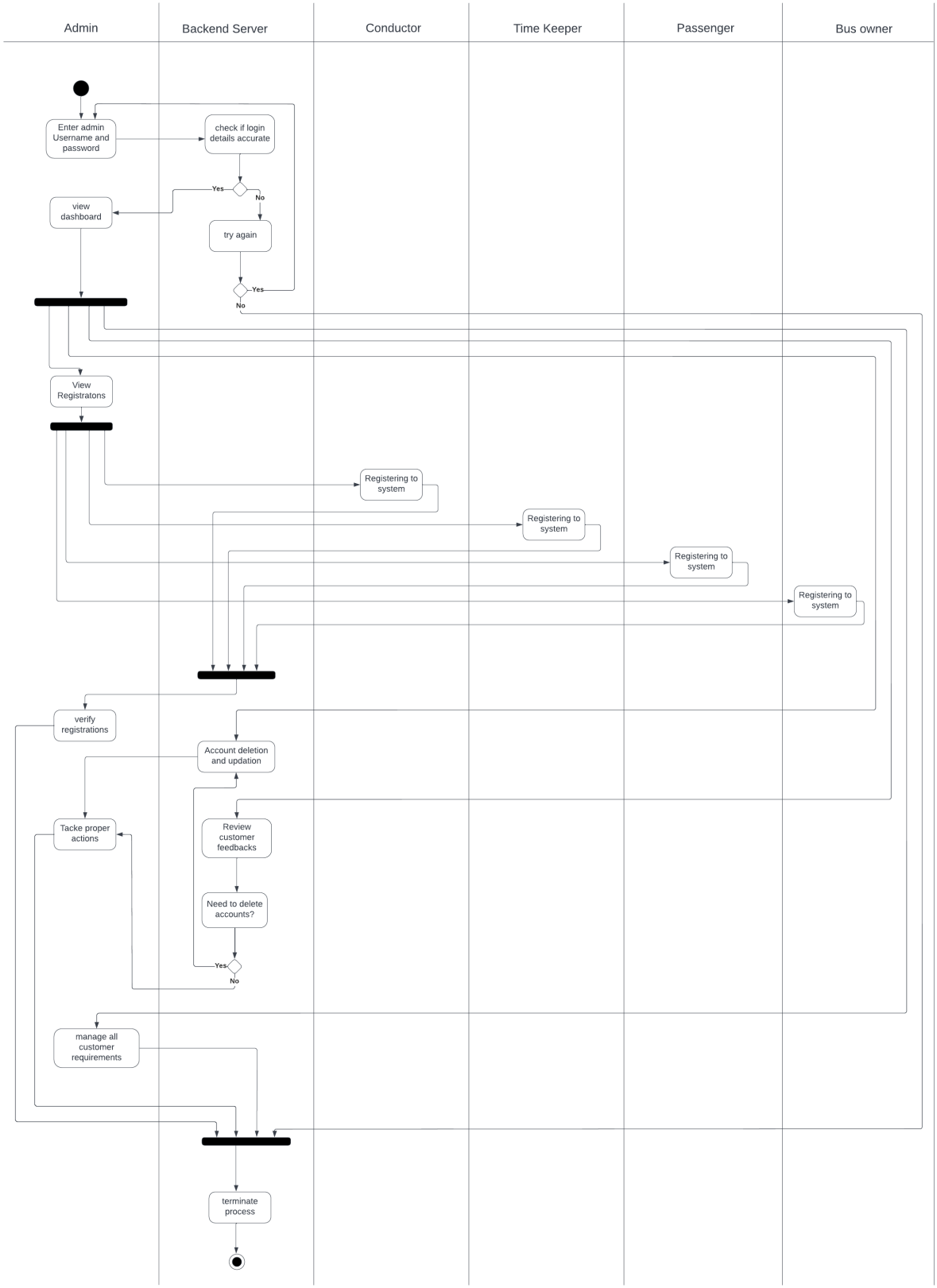


Figure 4.3.2 *Activity Diagram*

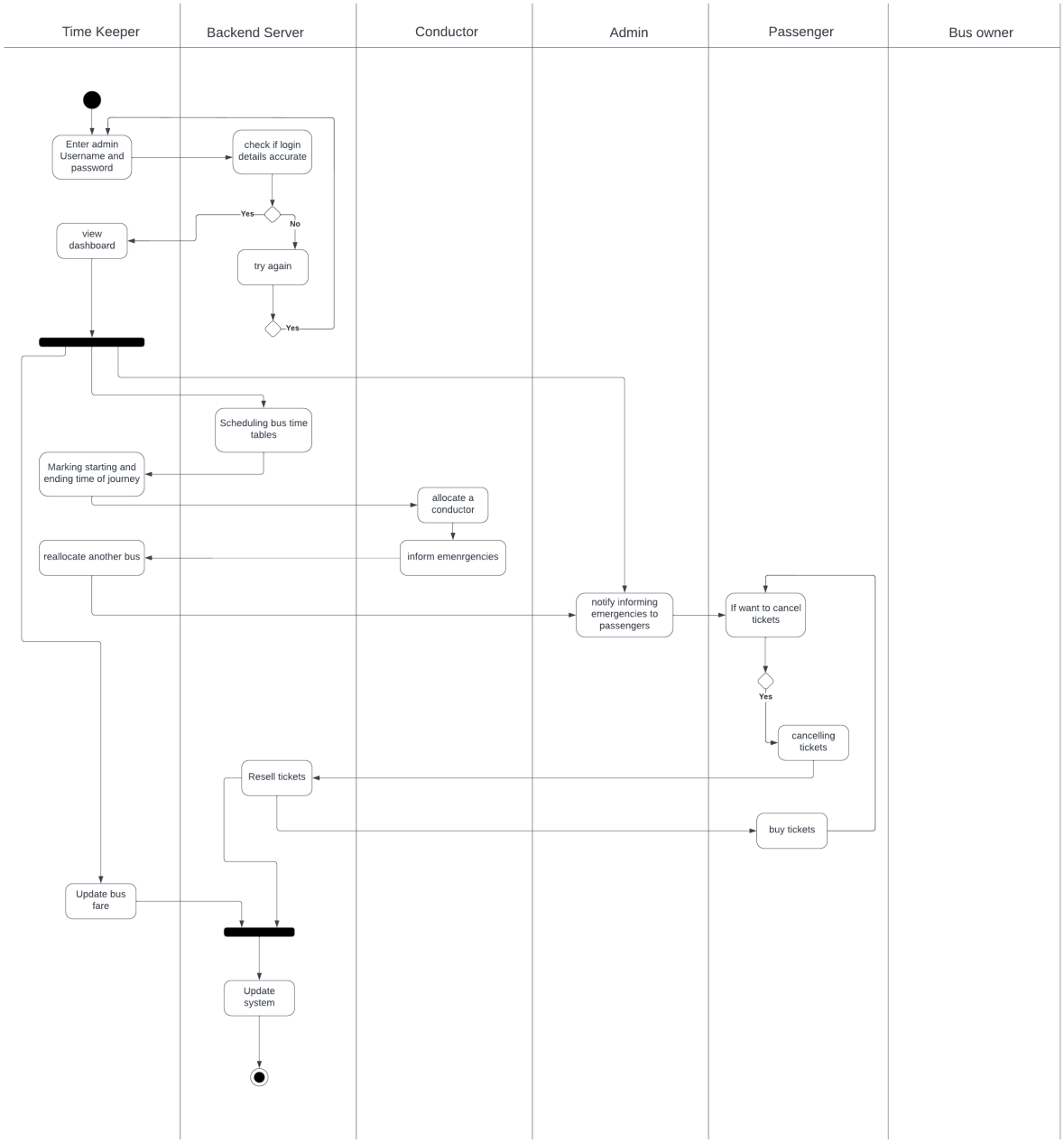


Figure 4.3.2 *Activity Diagram*

4.3.3. Class Diagram

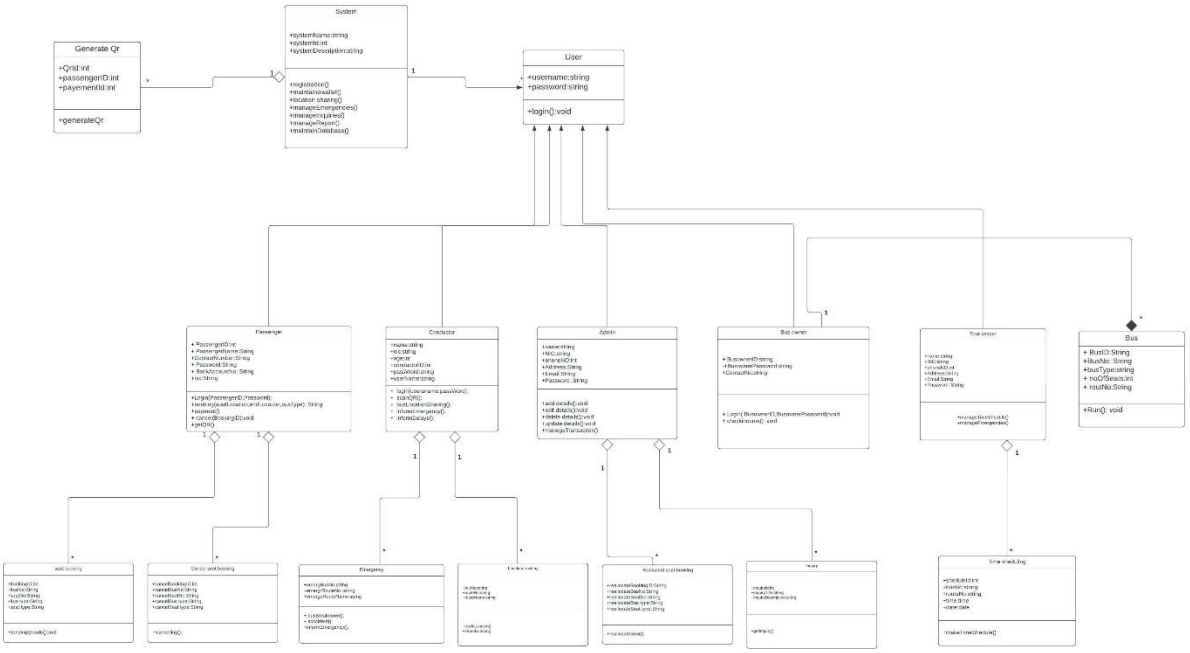


Figure 4.3.3 *Class Diagram*

4.3.4. ER Diagram

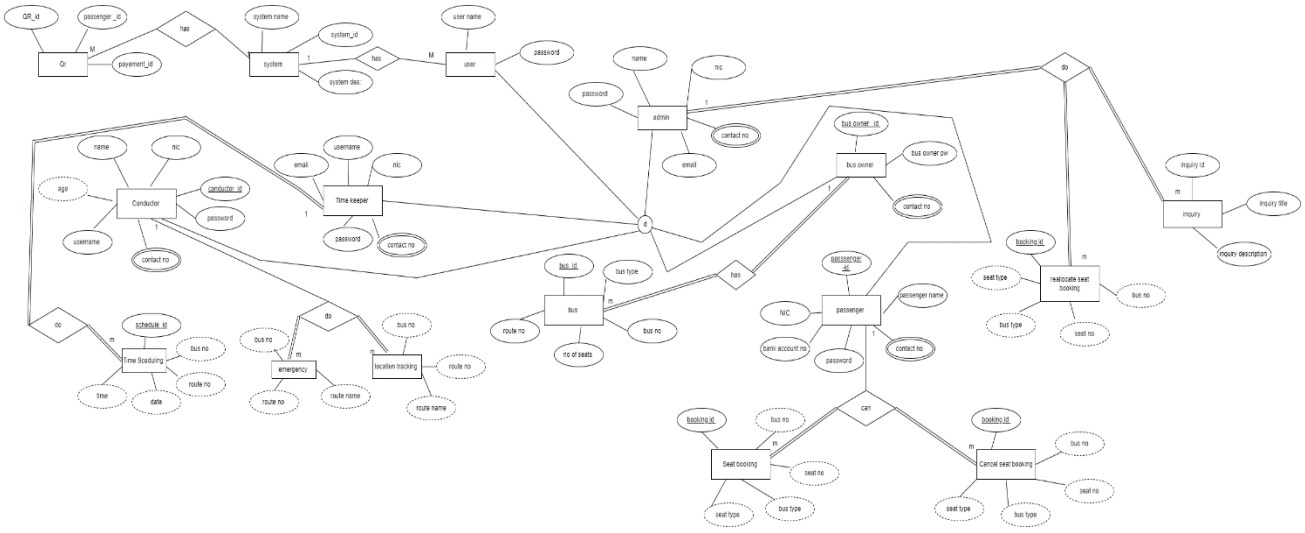


Figure 4.3.4 *ER Diagram*

4.3.5. Sequence Diagram

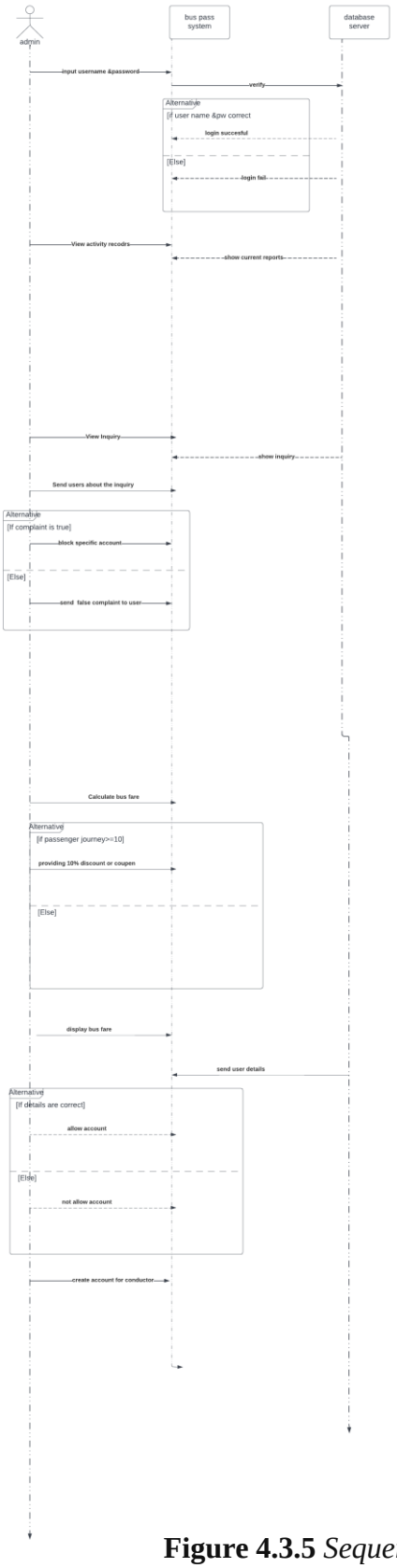


Figure 4.3.5 *Sequence Diagram*

Chapter 5 Implementation

5.1 Introduction

In this phase the status of the implementation of system will be discussed. We have designed UML diagrams for the easiness of delivering the key factors of the system and with the support of them we have completed the UI designing phase. And using those designs we are expecting to develop the front end. We have used Figma to design our user interfaces.
Meanwhile considering the requirements we have started implementing the backend using MySQL and have written basic queries for the databases of each module with the reference DBMS module and YouTube tutorials.

5.2 Flowcharts

5.2.1. Flowchart: Timekeeper

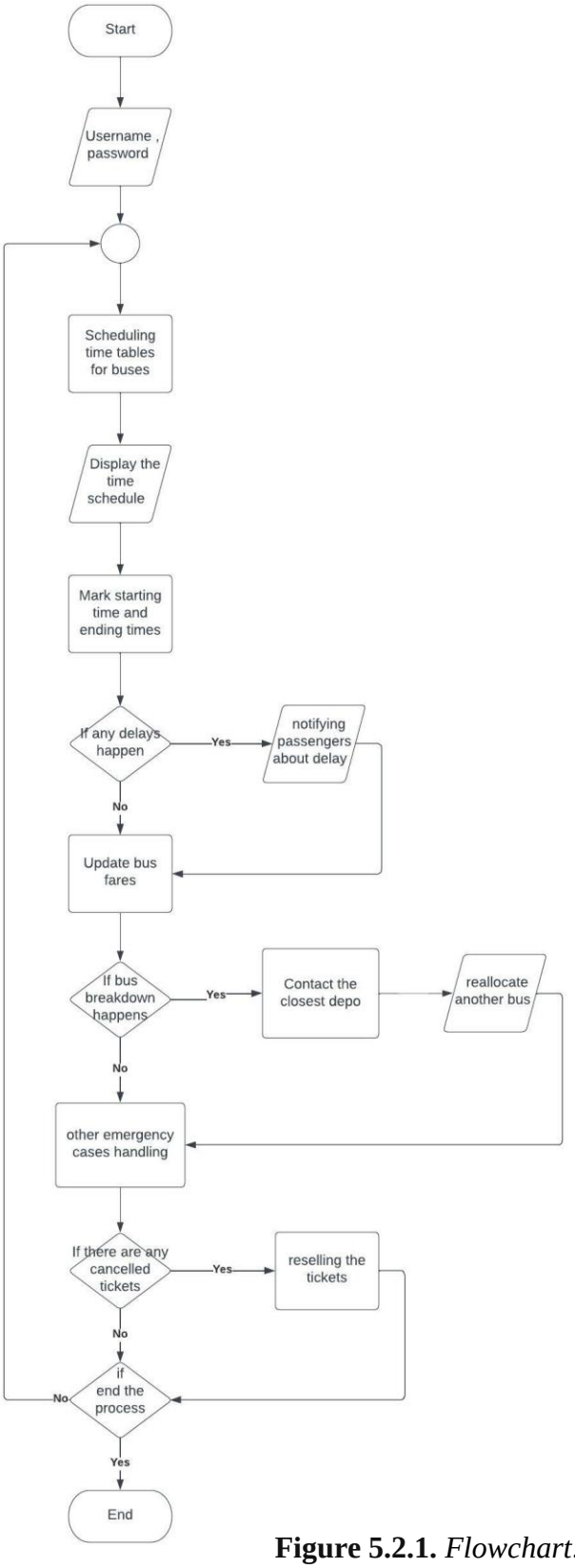


Figure 5.2.1. *Flowchart: Timekeeper*

5.2.2. Flowchart: Bus Owner

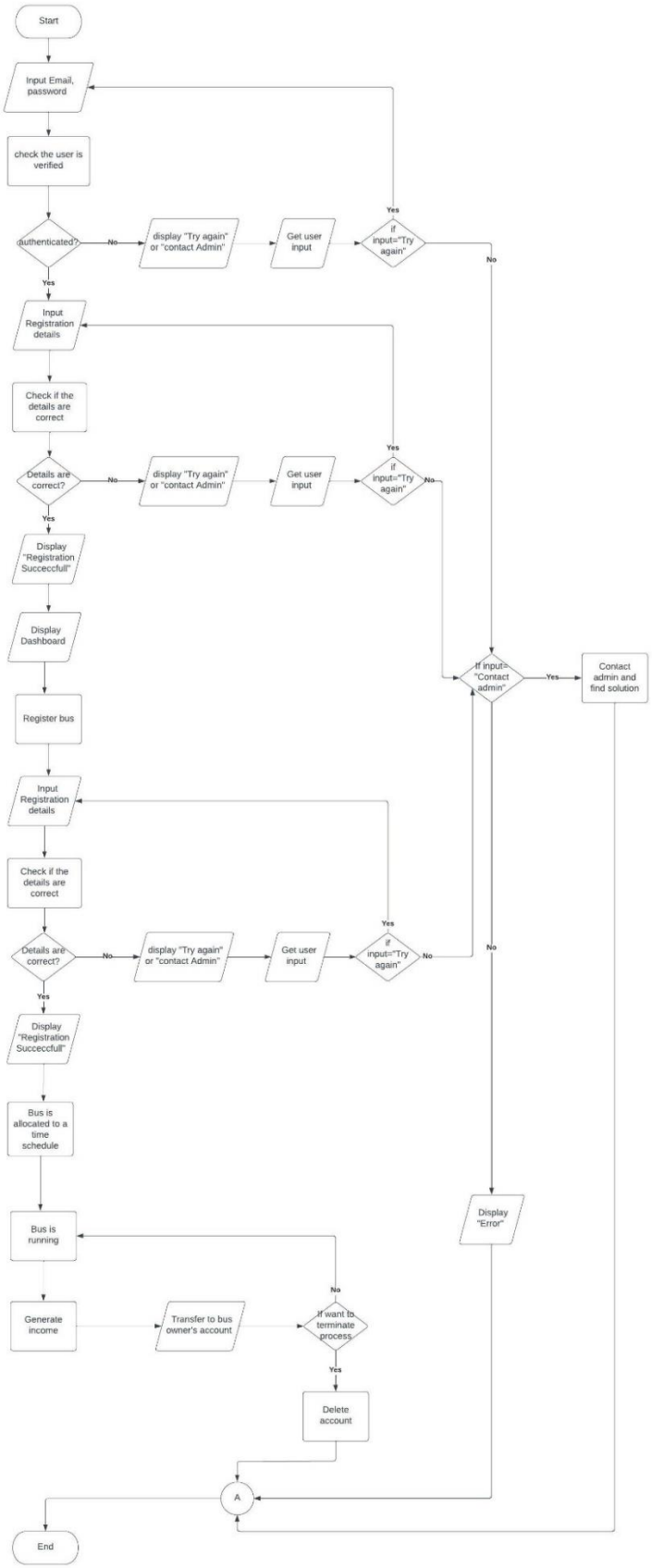


Figure 5.2.2. *Flowchart: Bus Owner*

5.2.3. Flowchart: Admin

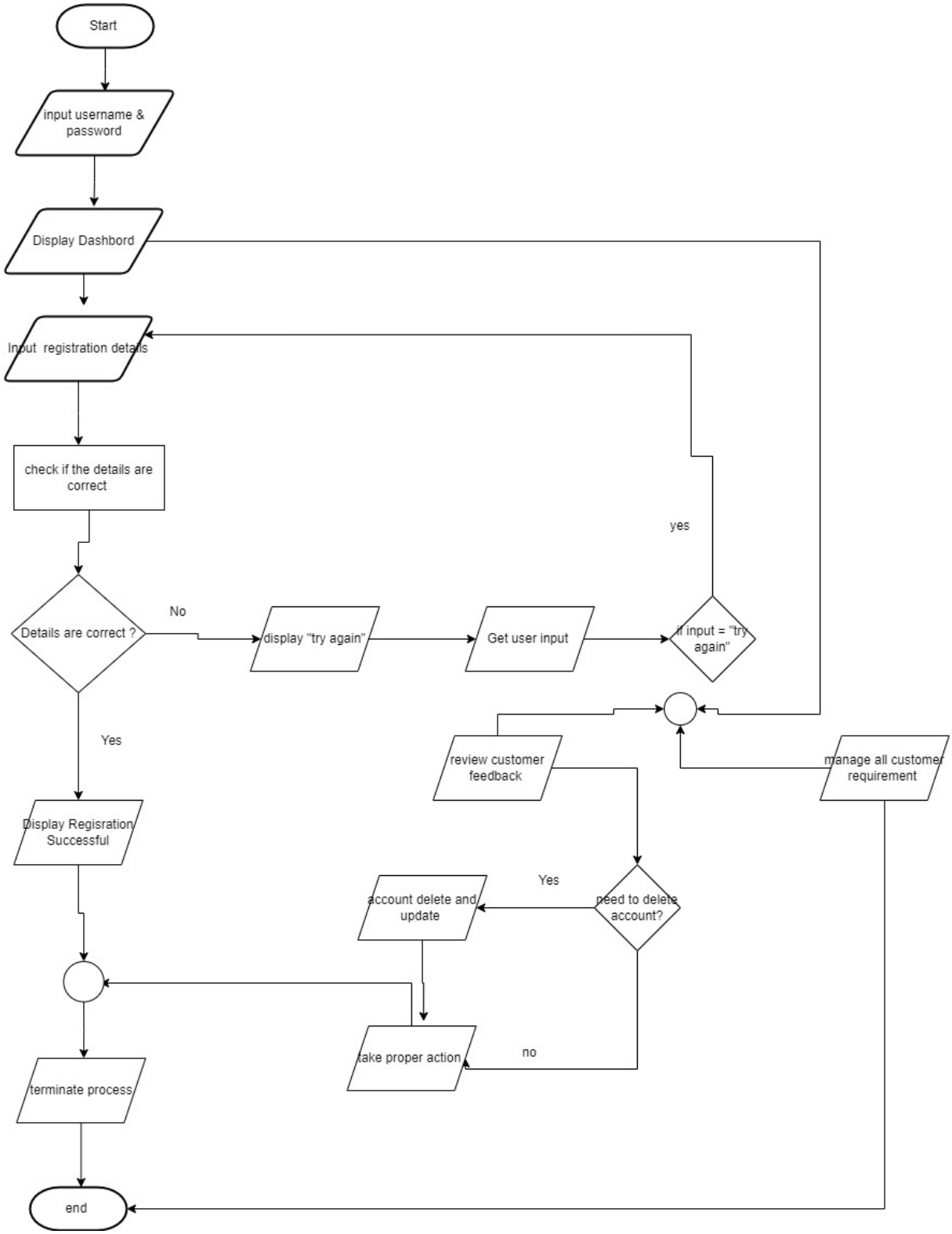


Figure 5.2.3. *Flowchart: Admin*

5.2.4. Flowchart: Passenger

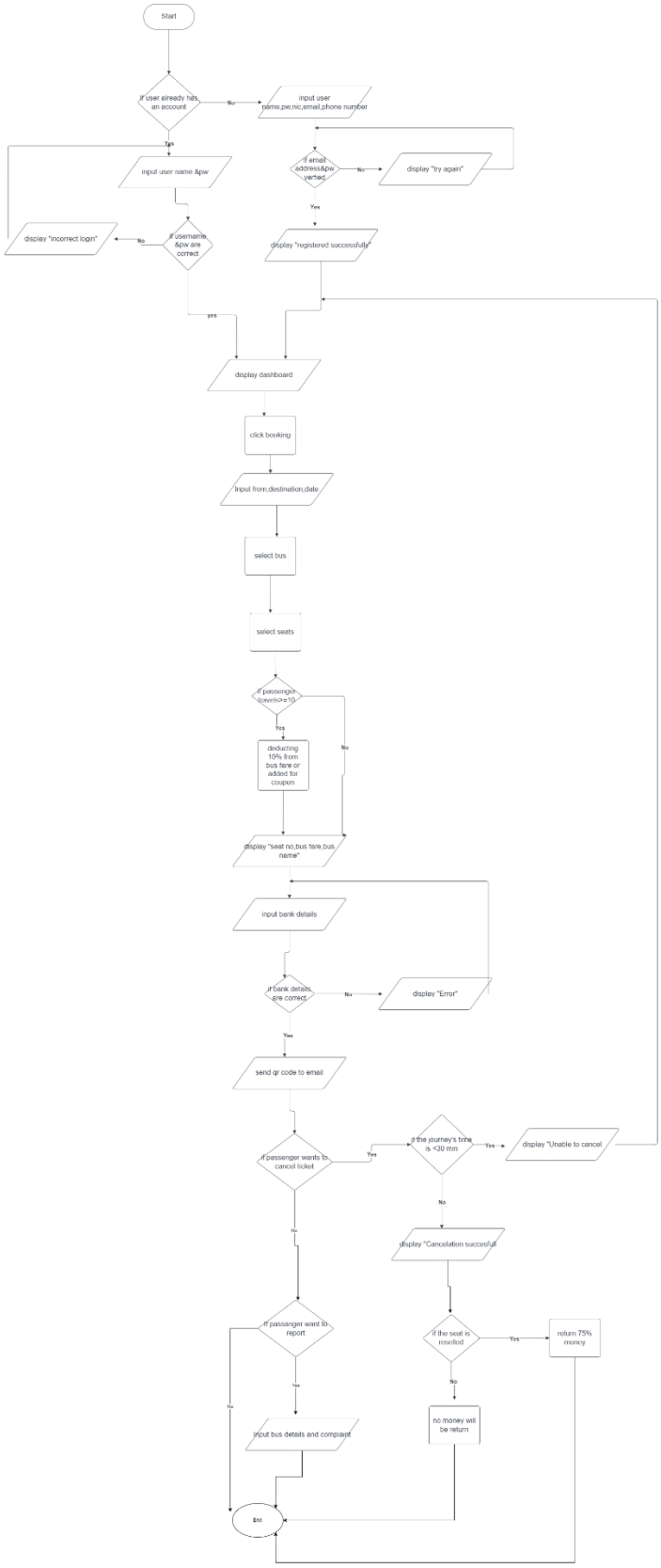


Figure 5.2.4. *Flowchart: Passenger*

5.2.5. Flowchart: Conductor

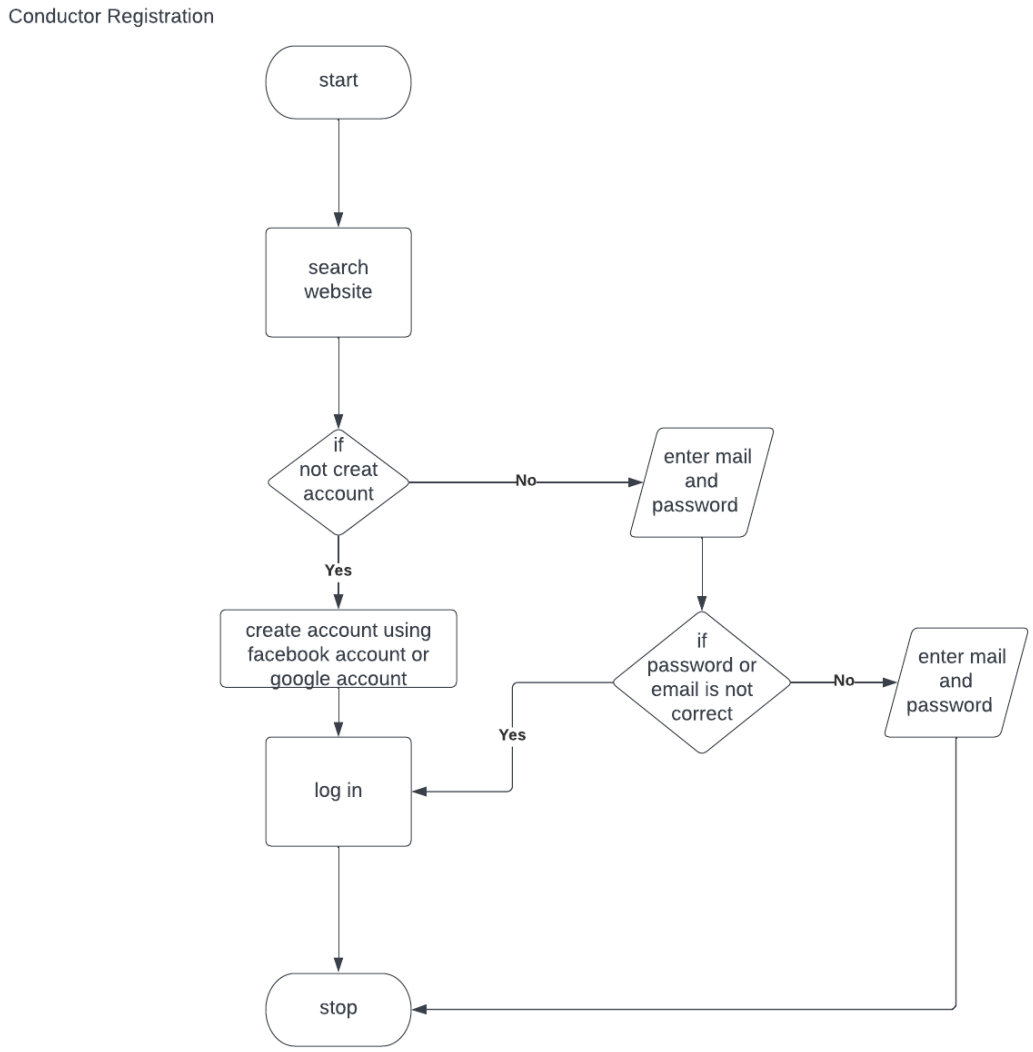
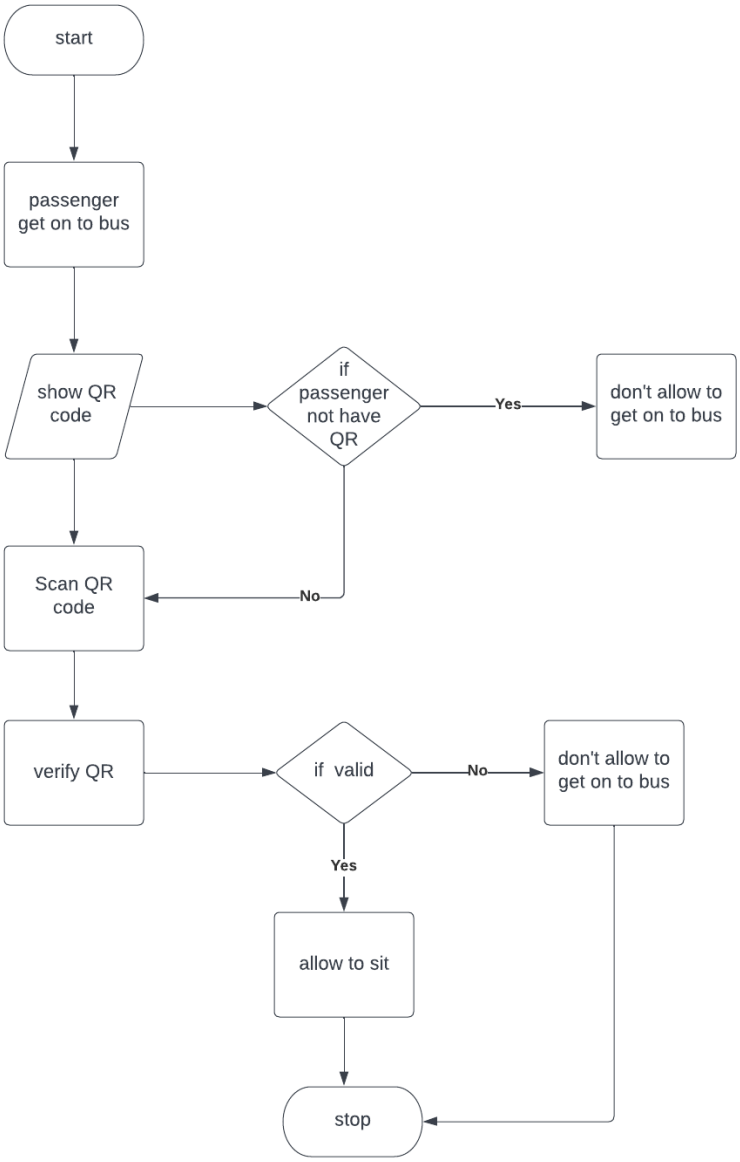
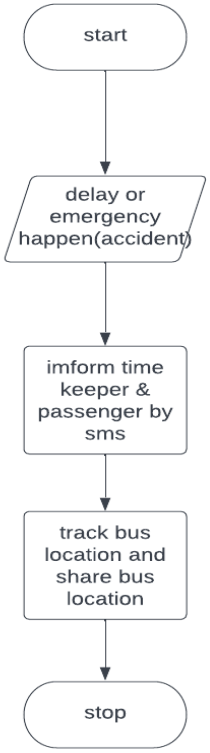


Figure 5.2.5.1 *Flowchart: Conductor*

QR code Scanning



Inform emergency or delay



5.3 Pseudocodes

5.3.1. Pseudocode: Timekeeper

```
BEGIN
    WHILE timekeeper wants to continue the process
        Input Username, password
        Scheduling timetables for buses
        Display the time schedule
        Mark starting time and end times
    IF any delays happen
            notifying passengers about the delay
        END IF
        Update bus fares
        IF a bus breakdown happens
            Contact the closest depo
            Contact the closest depo
        END IF
        other emergency cases handling
        IF there are any canceled tickets
            reselling the tickets
        END IF
    END
END
```

5.3.2. Pseudocode: Bus Owner

```
BEGIN
    Input Email, Password
    Check the validity of Email, Password
    IF Email, Password is verified
        Authenticate and authorize user
        Input Registration details
        Check the validity of Registration details
        IF Registration details are correct
```

```
        Registration Successful
        View Dashboard
            Input Bus Registration details
            IF Bus Registration details are correct
                Bus Registration Successful
                Check the validity of Bus Registration details
                Bus is allocated to a time schedule
                WHILE bus owner wants to continue
                    Bus is running
                    Generate income
                    Transfer to the bus owner's account
                END
            ELSE IF
                WHILE
                    bus owner wants to try again
                    Check the validity of Bus Registration details
                END
            ELSE IF
                Contact Admin
            ELSE
                Display Error
            END IF
ELSE IF
        WHILE
            bus owner wants to try again
            Check the validity of Bus Registration details
        END
ELSE IF
        Contact Admin
ELSE
        Display Error
END IF
```

```
        ELSE IF
                WHILE
                        bus owner wants to try again
                        Check the validity of Bus Registration details
                END
        ELSE IF
                Contact Admin
        ELSE
                Display Error
        END IF
END
```

5.3.3. Pseudocode: Admin

```
BEGIN
        Input user name , Password
        Check the validity of username , Password
        IF username , Password is verified
                Authenticate and authorize user
                  View Dashboard
                Input Registration details
                Check the validity of Registration details
                IF Registration details are verified
                        Registration Successful
                 Check  Review customer feedbacks
                 IF need to delete account
                 Account Deletion and updation
                Take proper Action
                Terminate Process
                END
                Manage all customer requirement
                Terminate Process
                END
```

5.3.4. Pseudocode: Passenger

```
Begin
    If user already has account
        Input username&password
        if username & password are correct
            Display dashboard
        Else
            Display" Incorrect login"
       End If
    Else
       Input username,password,nic,phone number and email address
       If email address and phone number verified
           Display "registration successful"
           Display dashboard
       Else
           Display "try again"
       End If
   Click booking
   Input from,destination &date
   Select Bus
   Select seats
    If passenger travels>=10
      Busfare=busfare-(10/100)*bus fare OR add to coupen
        Display bus fare
     Else
       Display busfare
    EndIf
  Input bank details
     If bank details are correct
         Send qr code
     Else
          Dispay "Error"
```

```
   End If
If passenger wants to cancel the ticket
      If the journey's time <30
          Display "unable to cancel the ticket"
      Else
          Display "Cancellation successful"
          If the seat is reselled
                Return 75% money
          Else
                No money will be return
          EndIf
      EndIf
  EndIf
   If passenger wants to report
    Input bus details &complaint
     EndIf

End
```

5.3.5. Pseudocode: Conductor

- For registration

```
Begin
   Search website
   input username, password
        if not have account
        create account using facebook or google account
                if registration not successful
                inform admin
                else
                register to system
        else
```

log in using username and password

end if

end if

End

- For scan QR

Begin

Passenger gets on the bus

Input QR code

If QR passenger is not have QR

Passenger can't enter to the bus

Else

Passenger can enter to the bus

Scan QR

If QR is not valid

Passenger can't enter to the bus

Else

Passenger can seat

End if

End if

End

- For emergency causes

Begin

Emergency (a breakdown) happen

Inform timekeeper and passenger by SMS

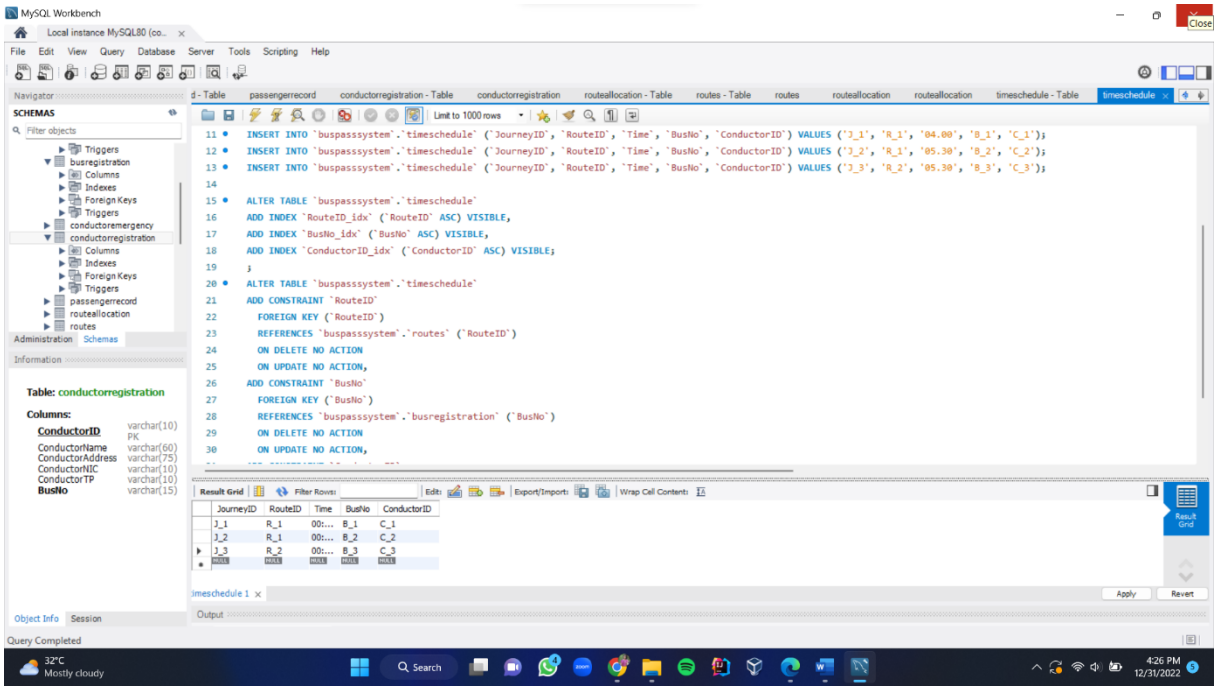
Track the bus location

Share the bus location

End

5.4 Database Implementation

5.4.1. Timekeeper



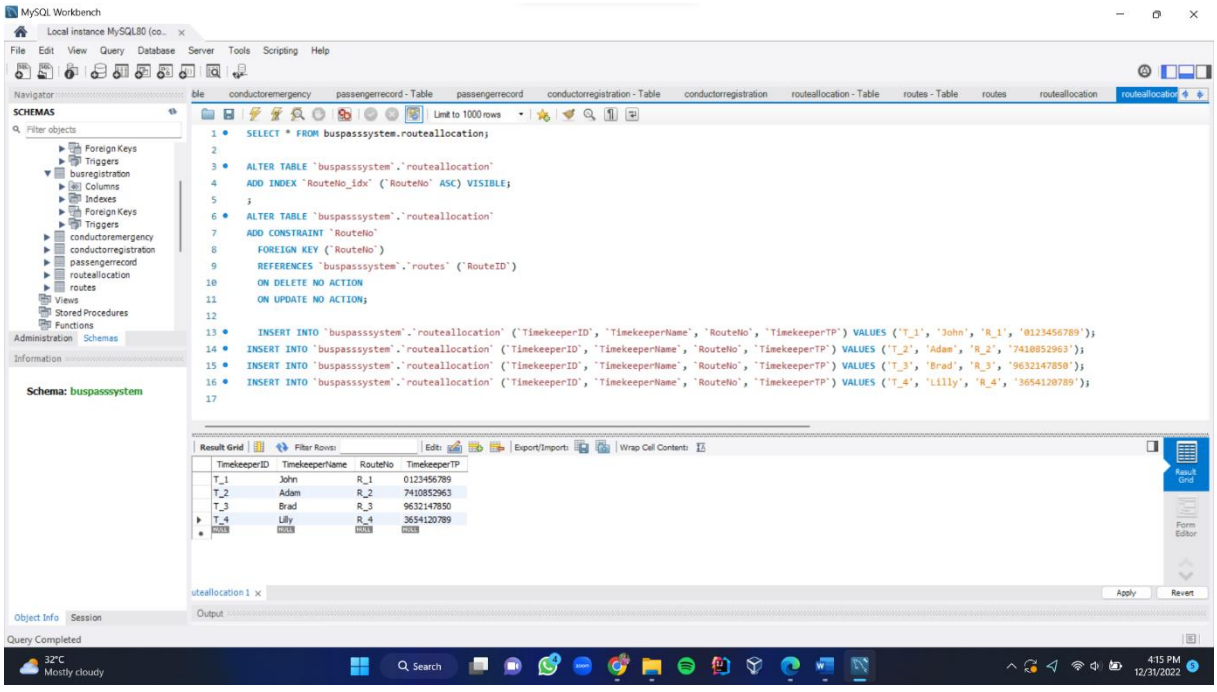


Figure 5.4.1. *SQL Queries: Timekeeper*

5.4.2. Bus Owner

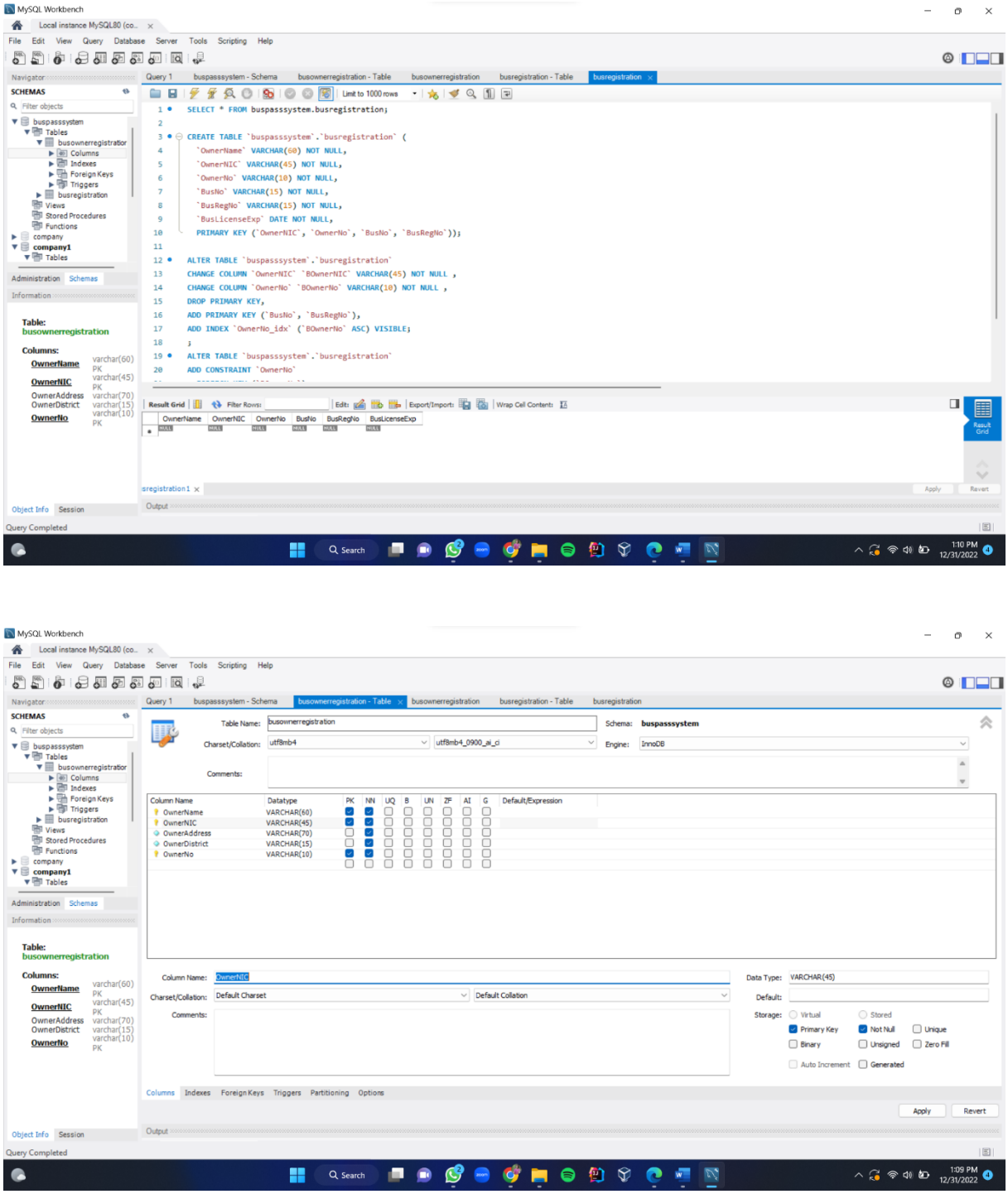


Figure 5.4.2. *SQL Queries: Bus Owner*

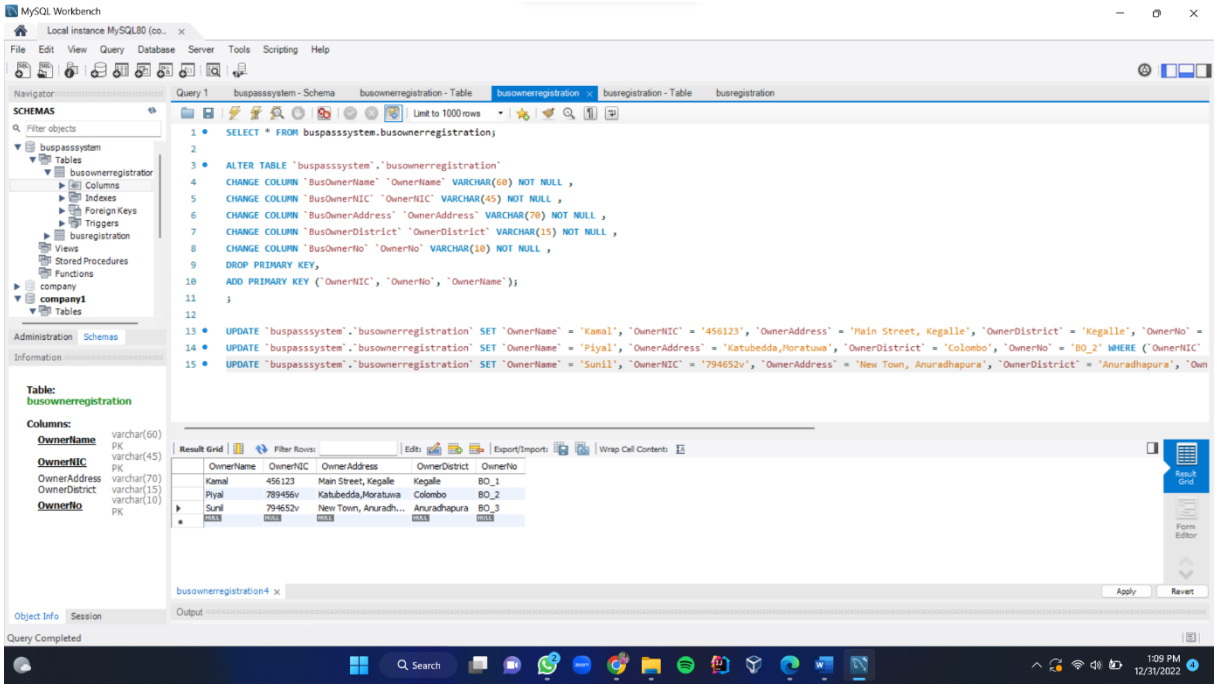


Figure 5.4.2. *SQL Queries: Bus Owner*

5.4.3. Passenger

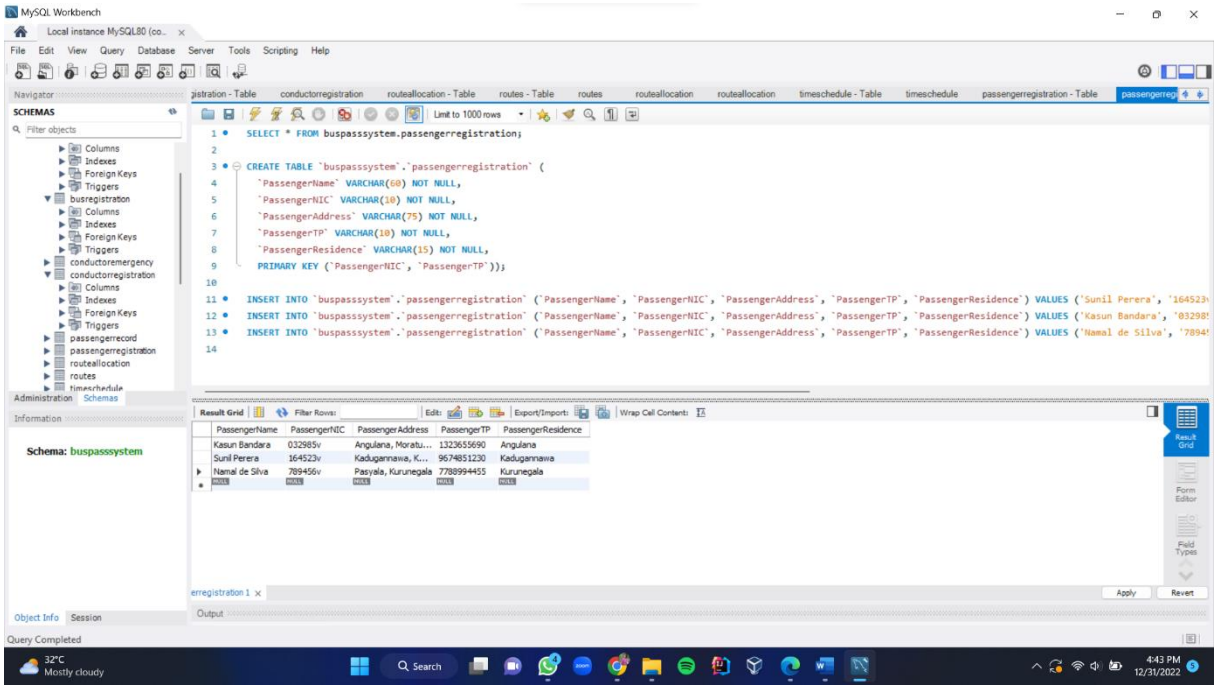
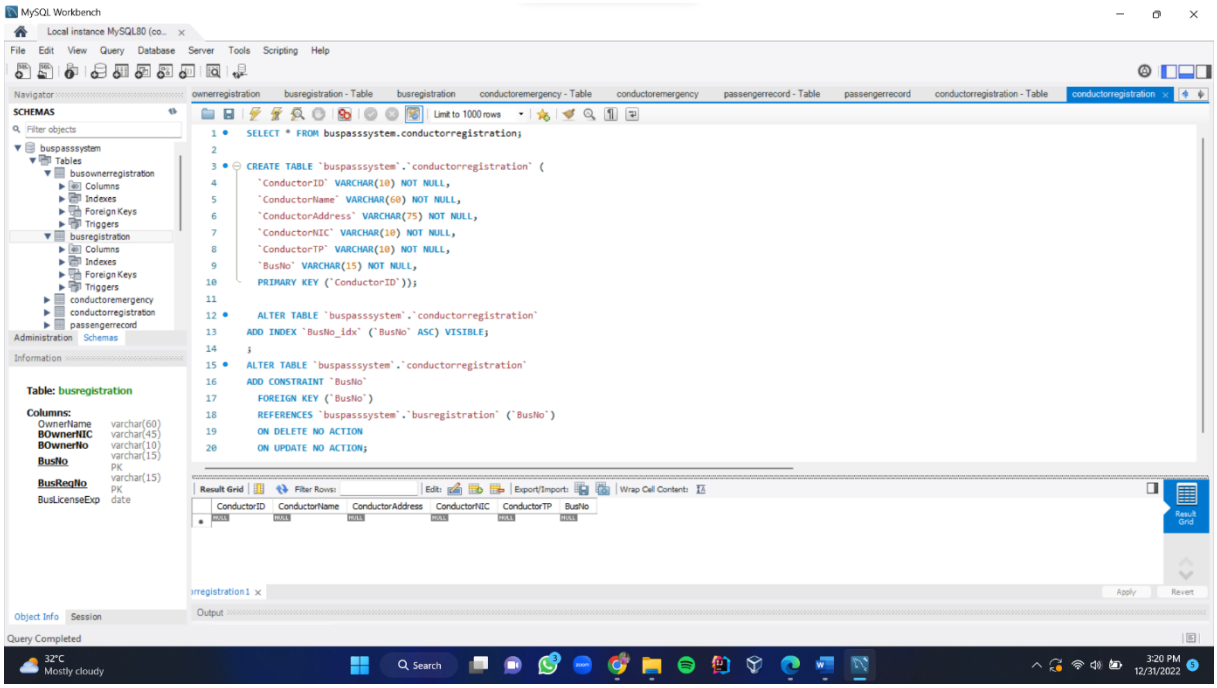


Figure 5.4.3. *SQL Queries: Passenger*

5.4.4. Admin



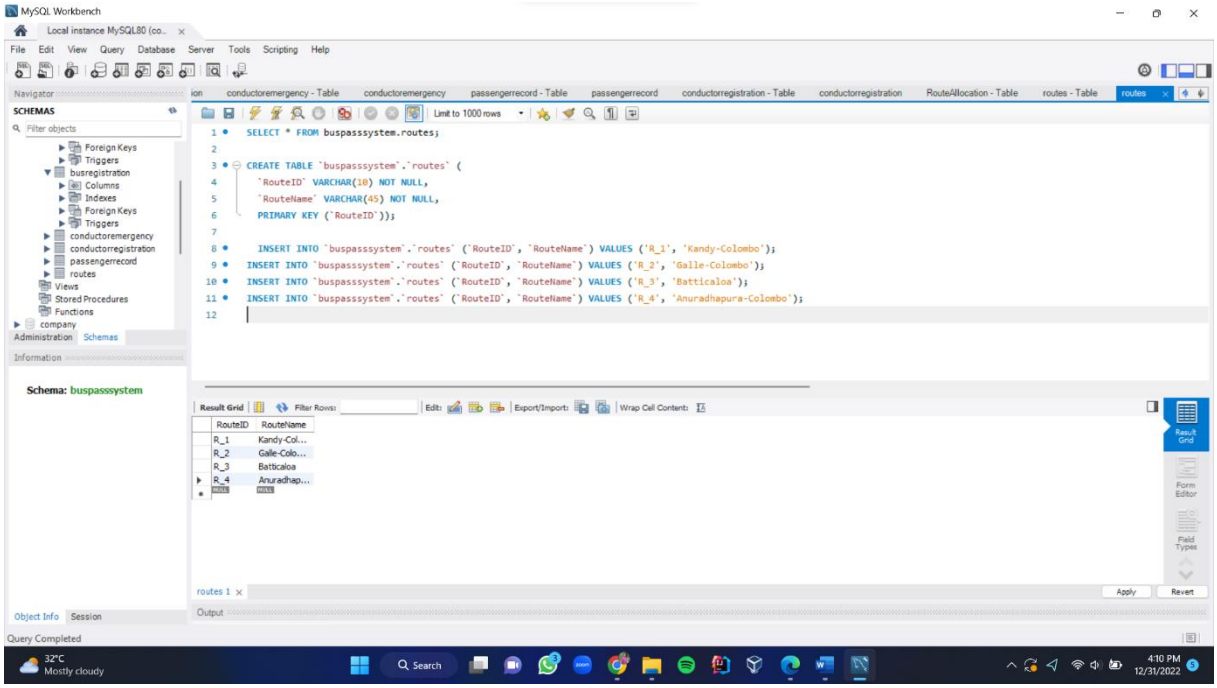


Figure 5.4.4. *SQL Queries: Admin*

5.4.5. Conductor

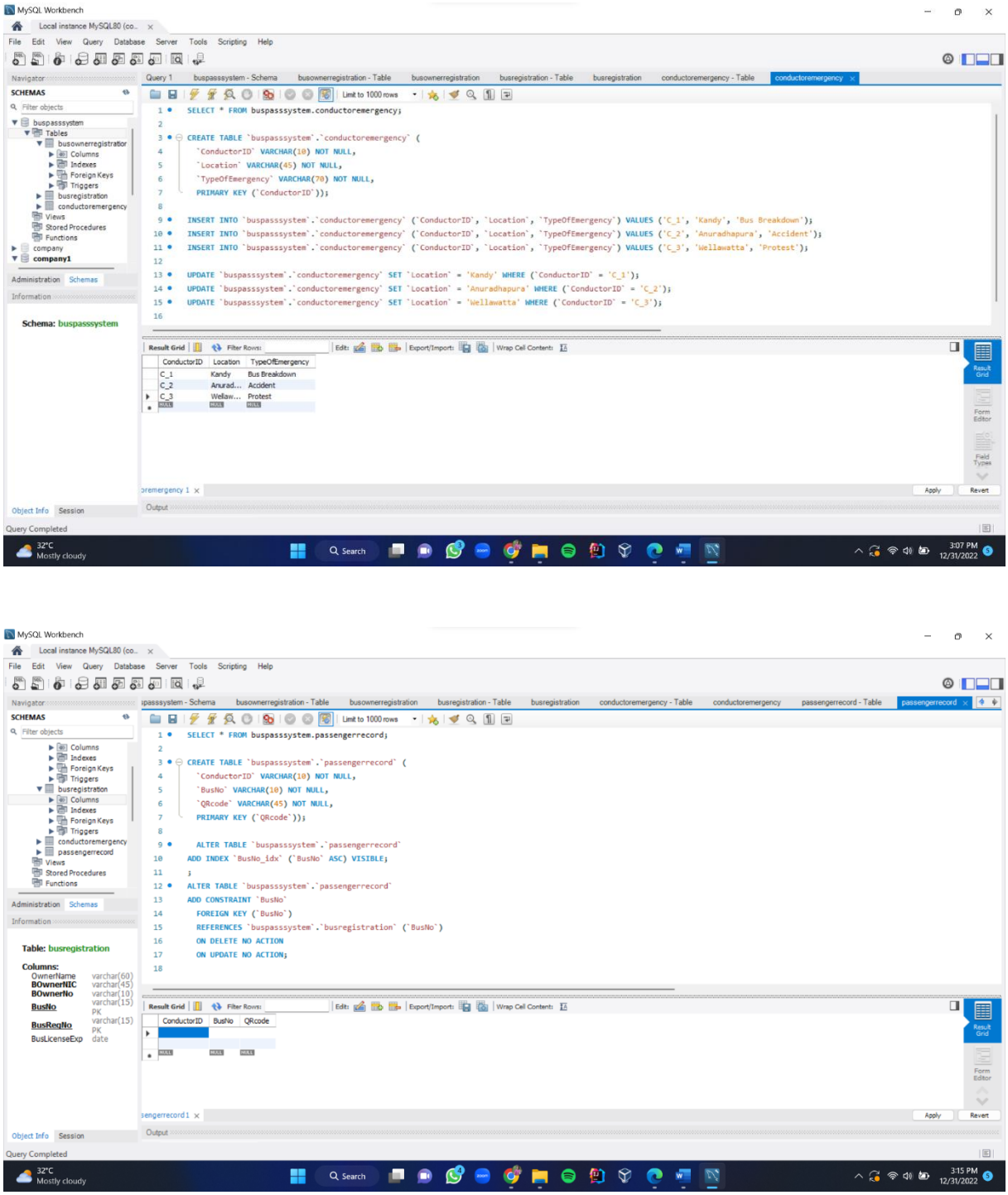


Figure 5.4.5. *SQL Queries: Conductor*

References

[1] https://www.mongodb.com/mern-stack

[2] https://www.w3schools.com/

[3] https://www.youtube.com/@Codevolution

[4]https://www.figma.com/file/mpVdkaDiS5E6IRM6Ii21zR/Untitled?nodeid=15%3A5&t=ndMild1QurIhVY1d-1

[5] https://youtu.be/Dorf8i6lCuk

[6]https://www.youtube.com/watch?v=zb3Qk8SG5Ms&list=PL4cUxeGkcC9jsz4LDYc6kv3ymONOKxwBU

[7] https://stripe.com/docs/keys

[8] https://www.twilio.com/docs/usage/api

Appendix A Individual Contribution to the Project

Name: Wijethilaka W.G.D.C.D.

Index No: 205119P

I am responsible for designing and implementing the sub system related to the role of "Bus owner". I expect to develop both front end and back end related to my module. I'm using HTML, CSS, Bootstrap, JavaScript and React for the frontend development. PHP, Node.js, SQL are used for the backend developing. Since money has to be transferred to the Bus owners account, I use Strip Sandbox API as the payment gateway. I have referred tutorials and related websites for gaining knowledge under instructions of our mentor as well. Currently I have designed the User Interfaces related to my modules and started referring resources to start implementations and I have implemented basic SQL queries for the backend using MySQL.

For the interim stage, I have drawn activity diagrams, wrote the chapter of the report and chapter 3(System Features) from software requirement specification, and contributed to finalizing the documents.

My contribution to the interim report:

- Drawing Activity diagrams
- Chapter 1 and Chapter 5 in the Interim report
- Chapter 3 and Chapter 6 in SRS
- Finalizing the Report

Name: Madhushika D.S.

Index no: 205063L

I am responsible for designing and implementing the role of “Admin”. I expect to develop both front end and back end related to my module. I’m using HTML, CSS, Bootstrap, JavaScript and React for frontend development. PHP, Node.js, and SQL are used for the backend development. I have referred to tutorials and related websites for gaining knowledge under the instructions of our mentor as well. Currently, I have designed the Admin Interfaces related to my modules and started referring resources to start implementations.

My contribution to the interim report:

- Drawing ER Diagram
- Drawing Class Diagram
- Chapter 2 in the Interim report
- Chapter 1 and Chapter 5 in SRS

Name: De Silva R.M.M.

Index no: 205016X

I am responsible for designing and implementing the role of "Conductor" and fuctions and activities that related to conductor role. I expect to develop both front end and back end related to my module. I'm using HTML, CSS, Bootstrap, JavaScript and React for the frontend development. PHP, Node.js, SQL are used for the backend developing. I have referred tutorials and related websites for gaining knowledge under instructions of our mentor as well. Currently I have designed the Conductor Interfaces related to my modules and started referring resources to start implementations.

My contribution to the interim report:

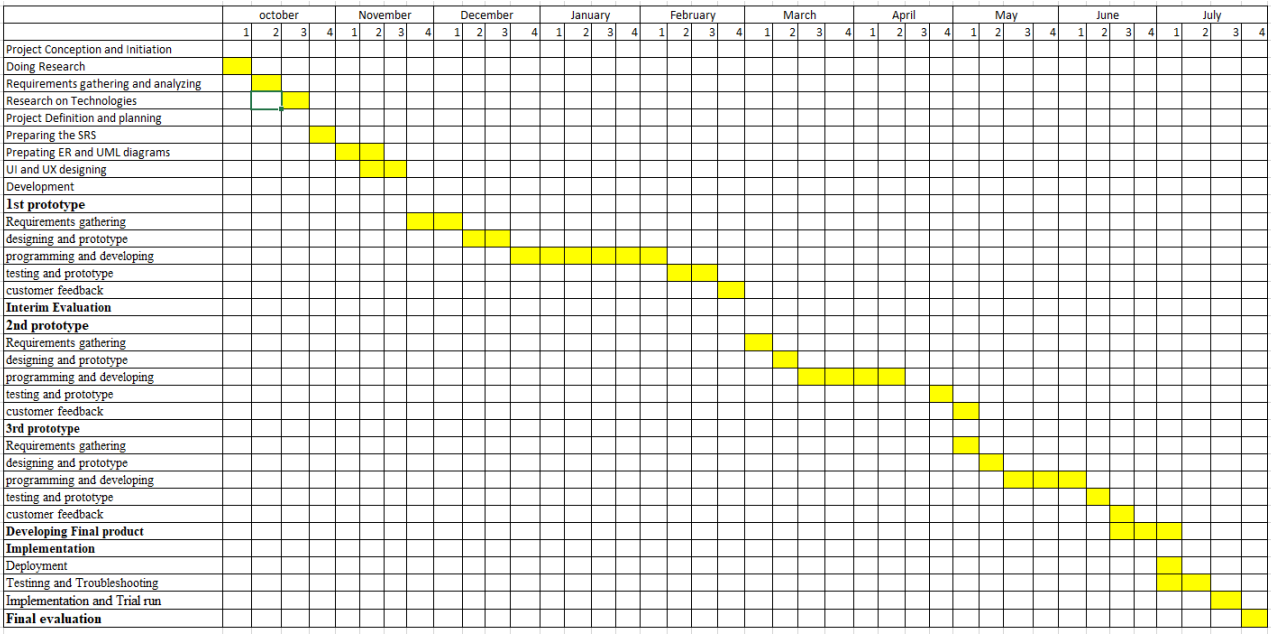
- Drawing Use case Diagram
- Making project management planning using gantt chart and network diagram.
- Chapter 3 in the Interim report
- Chapter 2 and Chapter 6 in SRS

Name: Rosni H.F

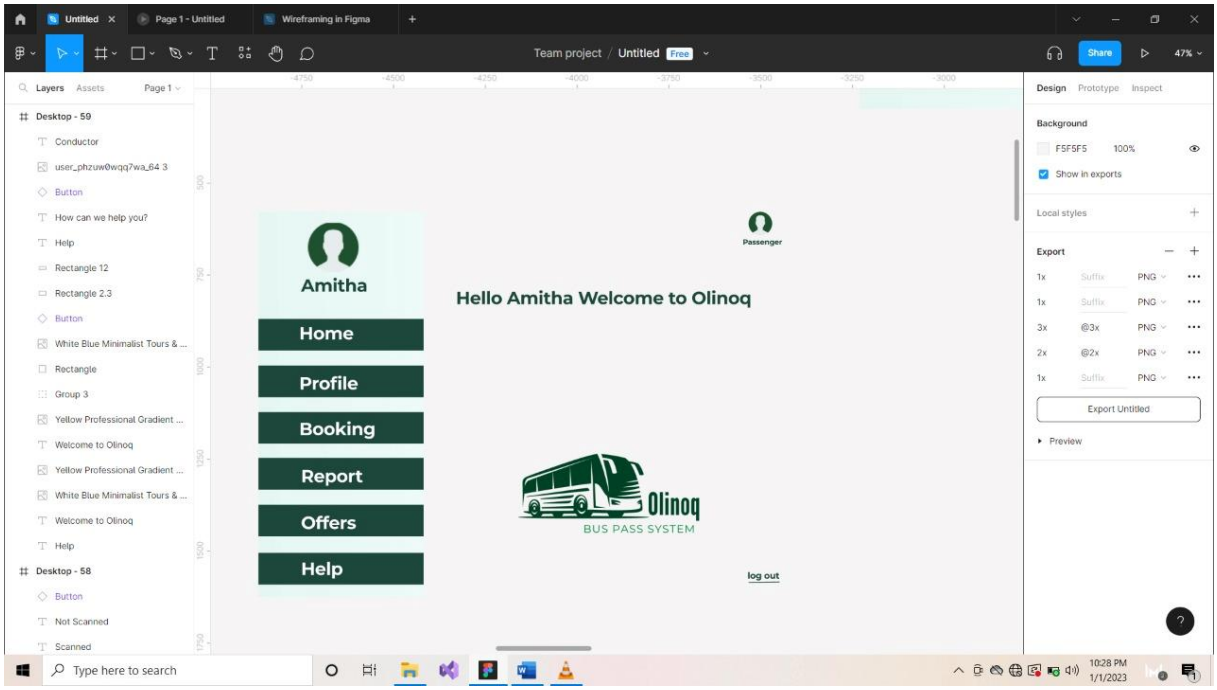
Index no: 185067A

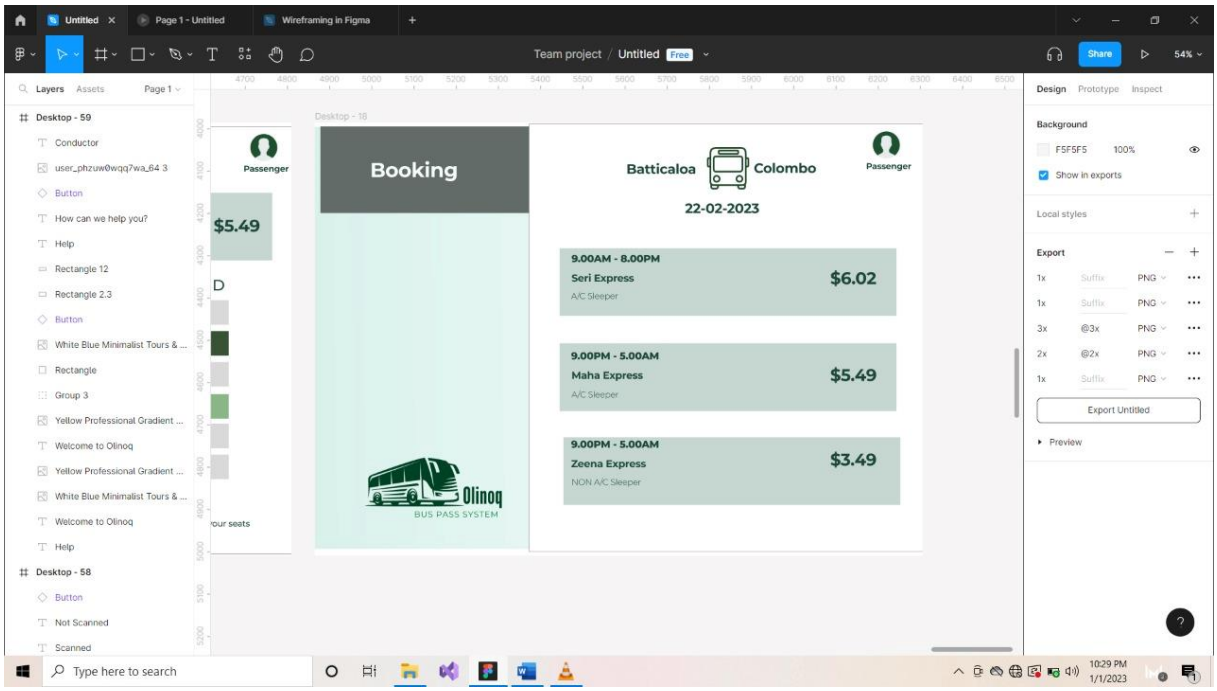
I am responsible for designing and implementing the role of “Timekeeper”. I expect to develop both front end and back end related to my module. I’m using HTML, CSS, Bootstrap, JavaScript and React for frontend development. PHP, Node.js, and SQL are used for the backend development. I have referred to tutorials and related websites for gaining knowledge under the instructions of our mentor as well.

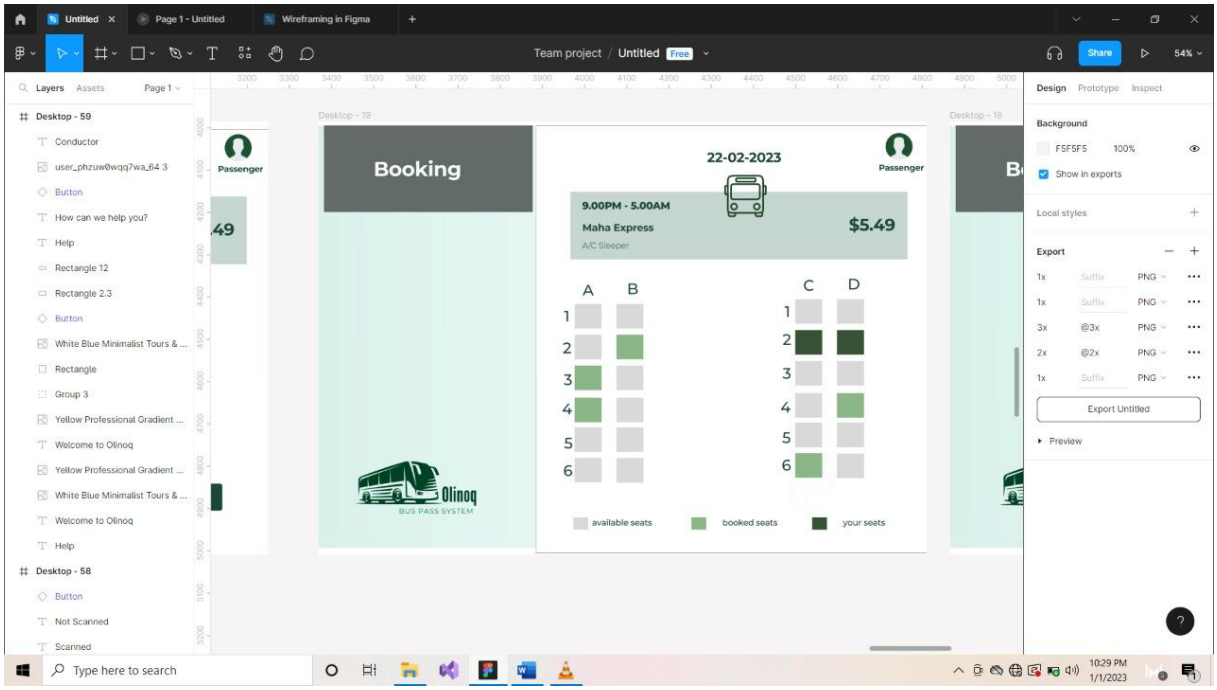
Appendix B Action Plan

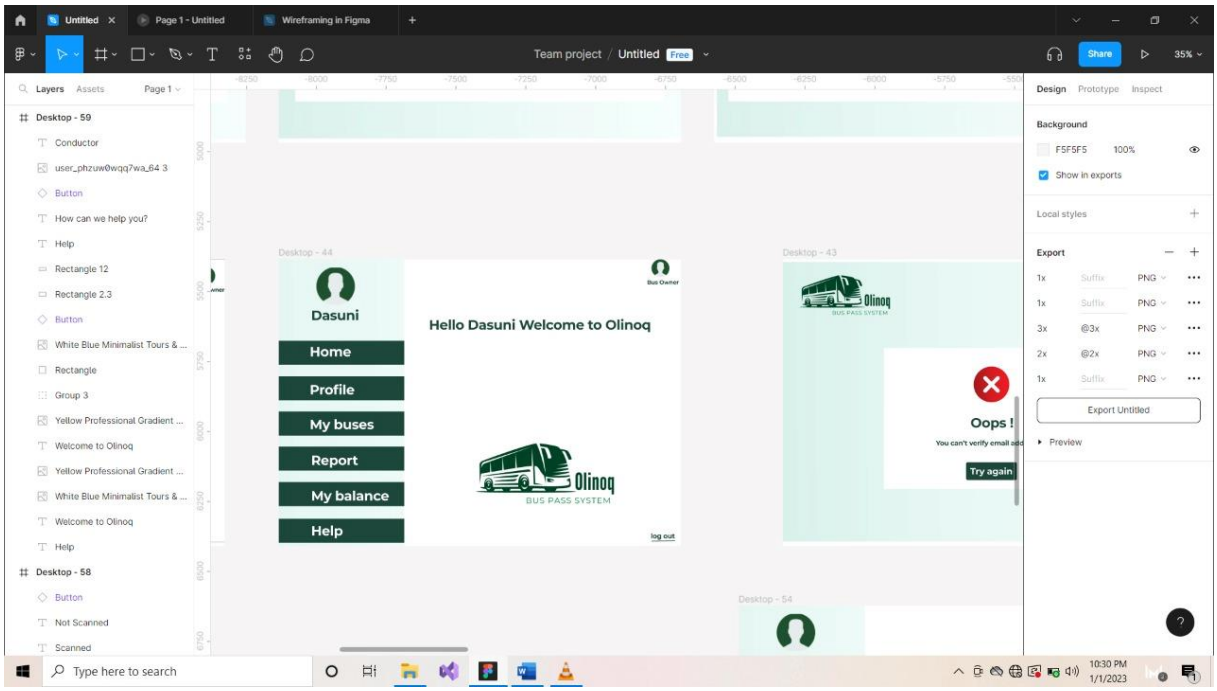


Appendix C Mocks-Up






Booking
22-02-2023
Passenger
9.00PM - 5.00AM
Maha Express
A/C Sleeper
$5.49
A B
C D
available seats
booked seats
your seats
Olinoq
BUS PASS SYSTEM


Dasuni
Home
Profile
My buses
Report
My balance
Help
Hello Dasuni Welcome to Olinoq
Olinoq
BUS PASS SYSTEM
Oops !
Try again

Appendix D Software Requirements Specification

This document contains an appendix with the software requirements specification report linked to it. Please take note that the SRS document's captions, page numbers, figure numbers, and table numbers are not related to the project interim report.

Software Requirements Specification

for

Cloud-Based Bus Pass System

Version 1.0 approved

Prepared by Team Dominators

Faculty of Information Technology/University of Moratuwa

01/01/2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of “” the real-time project that will benefit passengers as well as the owners of the buses, who are dissatisfied with the existing manual bus pass system developed by the team “Dominators”.

1.2 Document Conventions

IEEE 830-1998 standard for writing SRS document was used in writing this SRS document. 1.3 Intended audience and reading suggestion

1.3 Intended Audience and Reading Suggestions

This project is a prototype for the Online buss pass system that has been implemented under the guidance of university professors. This project is useful for people for booking their tickets. This document will be helpful to the users who are willing to use this web application. We recommend users go through the overall description of this document. Developers and testers are recommended to read the functional and non-functional requirements and the external interface requirements of this document.

1.4 Project Scope

This project aims to Provide safe, dependable, affordable, cost-effective, and efficient service to passengers, easy registration, a better profit to the bus owner, and easy administration and managing space for the administration.

1.5 References

• IEEE 830-1998 standard for writing SRS documents.

2. Overall Description

2.1 Product Perspective

The Cloud-Based Bus Pass Management System is a real-time web application that will benefit passengers as well as the owners of the buses, who are dissatisfied with the existing manual bus pass system. It allows passengers to travel more easily by scanning the ticket QR code with their smart devices. Passengers can purchase bus tickets beforehand for a week at any time online 24 hours a day. This system reduces the time wastage that happens to the passengers during a bus journey and provides ease of money transactions. It gives a live update as well about the relevant bus and the bus route to the passengers and confirms the safety of the journey too through the system.

In this system, passengers can check the availability of the buses, and seats and book seats at his/her required time and route. Also, passengers can cancel and renew their bus bookings through this system. QR code generation, money payment, and transaction, bus fare calculation, inquiry section, and special emergency and delay cause management will facilitate through this system automatically and efficiently that reducing the time and money of the users.

2.2 Product Features

The key features of this Clod-Based Bus Pass system are,

- Registration/login module for 5 primary actors as Admin, Conductor, Bus Owners, Timekeeper, and Passenger with different levels of permissions with an Authentication and Authorization module for ensuring the security of the system.
- Add, update, and cancel the accounts.
- Online bus seat booking, canceling, and renewal system to check the availability of the buses, and seats and can book seats at his/her required time and route easily and effectively.
- Fare calculation module for calculating bus fares in a standardized manner for distances.
- QR code generation module to verify the passenger's payment and use as bus pass/ticket for passengers.

- SMS and Email notifications to inform the emergency causes, started and reached time, etc.
- Emergency cause management module to inform passengers about the situation and provide a suitable solution.
- Location tracking module for emergency causes.
- E-wallet system for effective money transactions.
- Time management module to handle the time schedules of buses.
- Data-based management module to store all the information that the system is required.
- Online inquiry system to get feedback.
- Generate income reports.

2.3 User Classes and Characteristics

1. Passenger: Book, cancel, renew the seat booking, and make payments.
2. Bus Owner: - Register their bus/s in the system.
3. Conductor: - Inform emergency causes and delays to the timekeeper, scan the QR code to confirm the passenger's participation for the journey.
4. Admin: - Managing all accounts, handling inquiries, updating bus fares, managing seat booking, and canceling and monitoring the whole application.

2.4 Operating Environment

The system will operate on any web browser (except Internet Explorer) that is accessible through desktop operating systems which are Windows 7 or above, Mac OS, or Linux.
This system can be used on any machine/pc/mobile that has a web browser with a stable internet connection.

2.5 Design and Implementation Constraints

● Synchronization: - The system will only be accessible through a microcomputer with windows 7 or above, Mac OS, or Linux operating system. Mobile devices can connect to the system.
● Language Requirements: - English language will be available in the system.

- Frontend Technologies: - React
- Backend Technology: - Nodejs, Express
- Database Technology: -MY SQL
- Cloud Provider is AWS cloud hosting
- Stable internet connection for uninterrupted service.

2.6 User Documentation

- All the users of the system should be connected to the internet for the proper functioning of the system.
- It is assumed that all registered users have smartphones.

2.7 Assumptions and Dependencies

- It is assumed that admin and timekeeper already initialized,
- Only applicable for long journey routes (Ex: -Kandy to Colombo)and passengers can't get on to the bus without seat booking
- It is developed for one center.(Ex:-Kandy)

3. System Features

3.1 Creating Accounts

3.1.1 Description and Priority

It has a high priority since all the passengers, bus owners, conductors, and the other roles should be able to be identified separately.

3.1.2 Stimulus/Response Sequences

First, the users are authenticated and authorized by email to figure out they are not fake and to increase the security of the system. Once they are authenticated, they can register with the system by entering the relevant information.
In case the registration is unsuccessful, they can contact the admin to figure out what error has occurred.
The account is created after they register to the system successfully. After that, the dashboard is visible to the system according to their account type and they get the ability to function with the system to fulfill their requirements.

3.1.3 Functional Requirements

REQ-1: Enter email and password to verify the user
REQ-1: registering to the system by entering personal details
REQ-1: Admin involve in the process of account registration if any consequences occurred

3.2 Time Scheduling

3.2.1 Description and Priority

This is a very high priority for the system because the basic core of the system is booking a ticket online. A timetable of buses being visible to the passengers is what the passengers make to book a ticket from a certain bus.

3.2.2 Stimulus/Response Sequences

In this module the timekeeper schedules timetables for bus routes including all buses. After that, the system is updated with timetables and they are visible to the passengers so they can book a ticket for the bus they needed. The timekeeper is responsible to check whether the buses depart and arrive on time and marking them. He can update the timetables according to special reasons like emergencies described under another feature.

3.2.3 Functional Requirements

REQ-1: Getting bus details
REQ-2: Creating timetables according to routes
REQ-3: Update system
REQ-4: Mark starting time and ending time of a journey

3.3 Ticket Generating

3.3.1 Description and Priority

This is also a very high priority for, and it is the key process of the system. Passengers can easily obtain the bus pass efficiently.

3.3.2 Stimulus/Response Sequences

Passengers check the bus schedule and pick a ticket. Then the fare calculation happens, and the price is displayed to the passenger. If the passenger pays the price the bus pass will be generated with a QR code and sent to the passenger.
Passengers can let the conductor scan the QR code to the bus conductor to verify him and get into the bus. After scanning the QR code the relevant seat will be marked as taken in the system for the particular journey.

3.3.3 Functional Requirements

REQ-1: Pick a ticket

REQ-2: Purchasing the ticket

REQ-3: Sending the bus pass

REQ-4: scanning QR and verifying passenger

3.4 Emergency Cases Handling

3.4.1 Description and Priority

Describe how the system provides alternatives when a troublesome situation occurred while the process is happening.

3.4.2 Stimulus/Response Sequences

When there is a delay to start the journey, the conductor can inform to the admin and the admin can send a notification to the relevant passengers of the relevant journey about the delay.
When a breakdown of the bus occurs while traveling, bus conductor can inform to the timekeeper of the location where the breakdown happened. And the timekeeper can refer to the database of the deposit and contact the nearest depo manually and alternate another bus to avoid difficulties that the passengers could face.

3.4.3 Functional Requirements

REQ-1: Conductor inform the timekeeper about delays or breakdowns

REQ-2: Sharing the location

REQ-3: Allocating another bus and rescheduling timetables by the timekeeper

3.5 Bus Registration

3.5.1 Description and Priority

This is a priority feature for bus owners to enter their buses into this system.

3.5.2 Stimulus/Response Sequences

Once the bus owner has created an account, he can register his buses one by one in the system. He has to register the bus through the dashboard and has to fill in the fields that ask about the details of the bus and submit the relevant documents to verify the bus. Then once the bus starts running on the routes, the relevant income is transferred to the bus owner's account, and he can also check the balance on the dashboard.

3.5.3 Functional Requirements

REQ-1: Having a "Bus owner" account
REQ-2: Entering bus details
REQ-3: Submitting license, insurance, eco test certificate, and relevant documents.
REQ-4: registering the bus successfully
REQ-5: having a money transaction account.

3.6 Money Transactions

3.6.1 Description and Priority

Higher priority feature for passengers and bus owners, passengers must have an account to transfer money to obtain the bus pass and the bus owners collect the income via their account.

3.6.2 Stimulus/Response Sequences

Both passenger and bus owners get their accounts to transfer money at the moment their accounts are created. Once the bus pass is obtained by the passenger, the relevant amount of money is deducted from his account and debited to the bus owner's account.
If any passenger cancels a ticket before the journey, 75% of the total fare will be refunded back to the passenger's account.

3.6.3 Functional Requirements

REQ-1: Having an account to do money transitions

REQ-1: having enough balance to buy a ticket or refund an amount.

3.7 Inquiry section

3.7.1 Description and Priority

It is important for the passengers to mention their complaints and experiences about the whole service.

3.7.2 Stimulus/Response Sequences

Passengers can mention their experience and feedback in the inquiry section. Those are viewed by other passengers as well so passengers can get an idea about a certain bus, conductor, or service before booking a ticket. Admin can reply to the feedback and take proper action if something inappropriate happens. And admin can ban accounts that make unsuitable comments annoying other users.

3.7.3 Functional Requirements

REQ-1: Having a valid account
REQ-2: Experiencing the service
REQ-3: Making comments be viewed by other users
REQ-4: Close inspection by admin to take proper actions

4. External Interface Requirements-Amitha

4.1 User Interfaces

The User Interfaces are where the user interacts with the system. We have designed several UIs for the 5 roles in our system. And what they view on the dashboard is different depending on the role. We have designed the UIs and we have started implementing the frontend using HTML, CSS and Bootstrap.

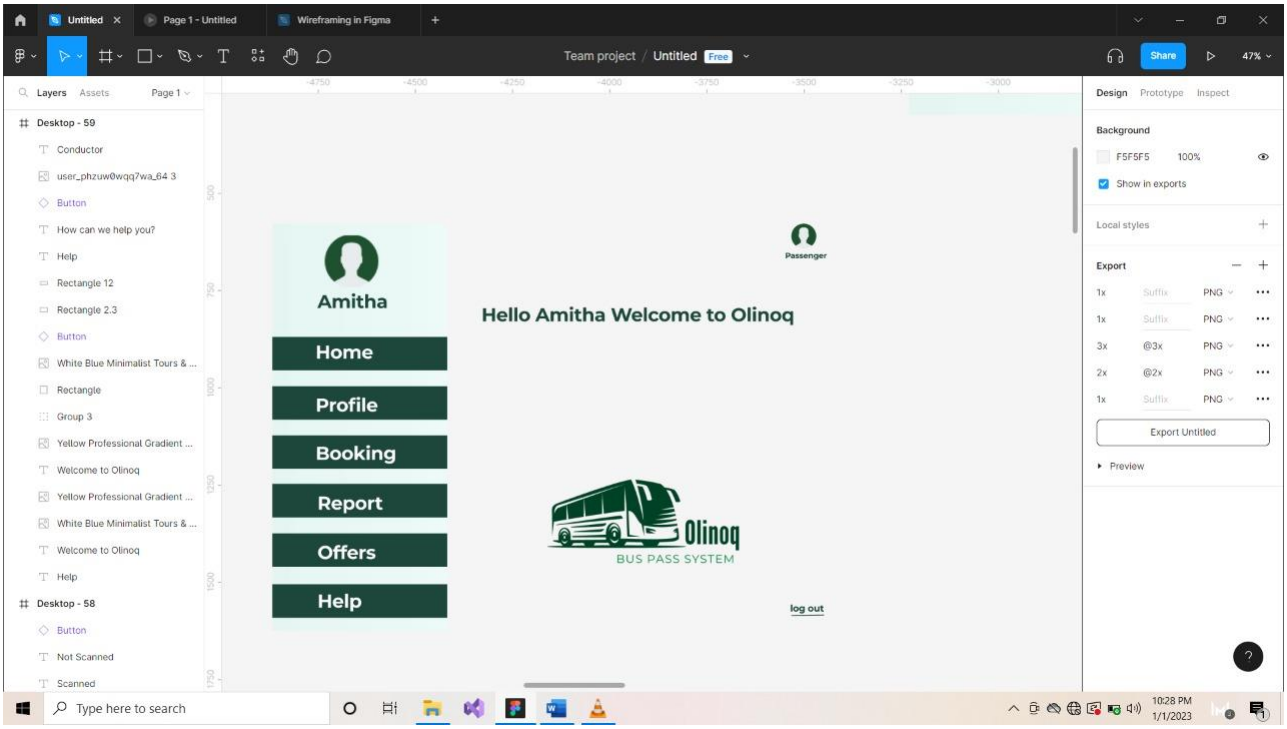


Figure 1 *Passenger dashboard view*

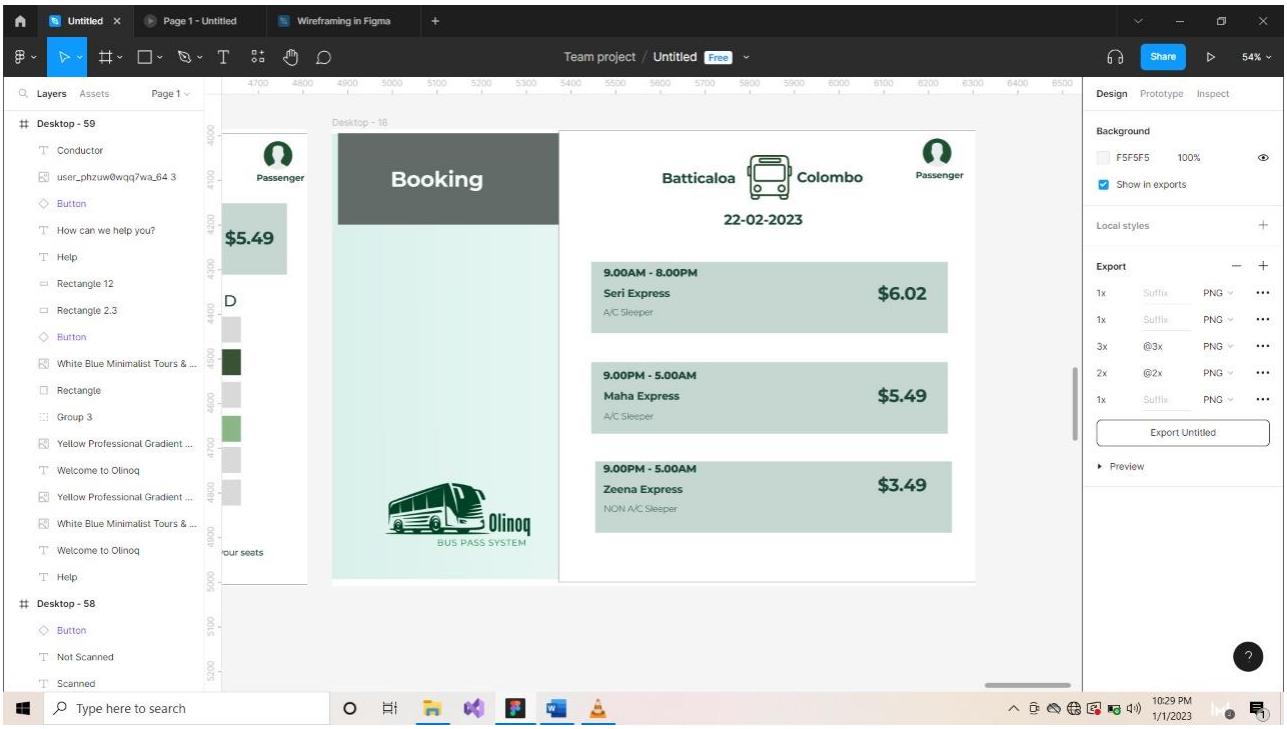


Figure 2 *Passenger Seat booking view*

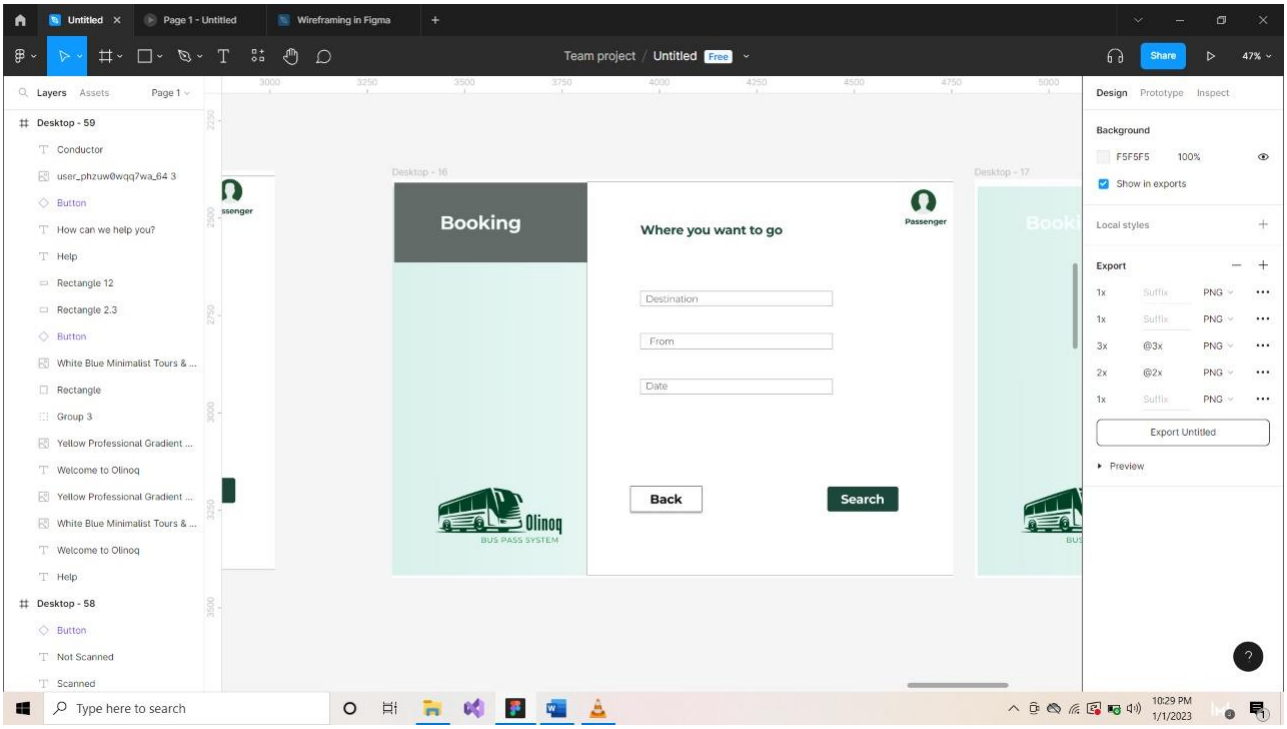


Figure 3 *Passenger Seat booking view*

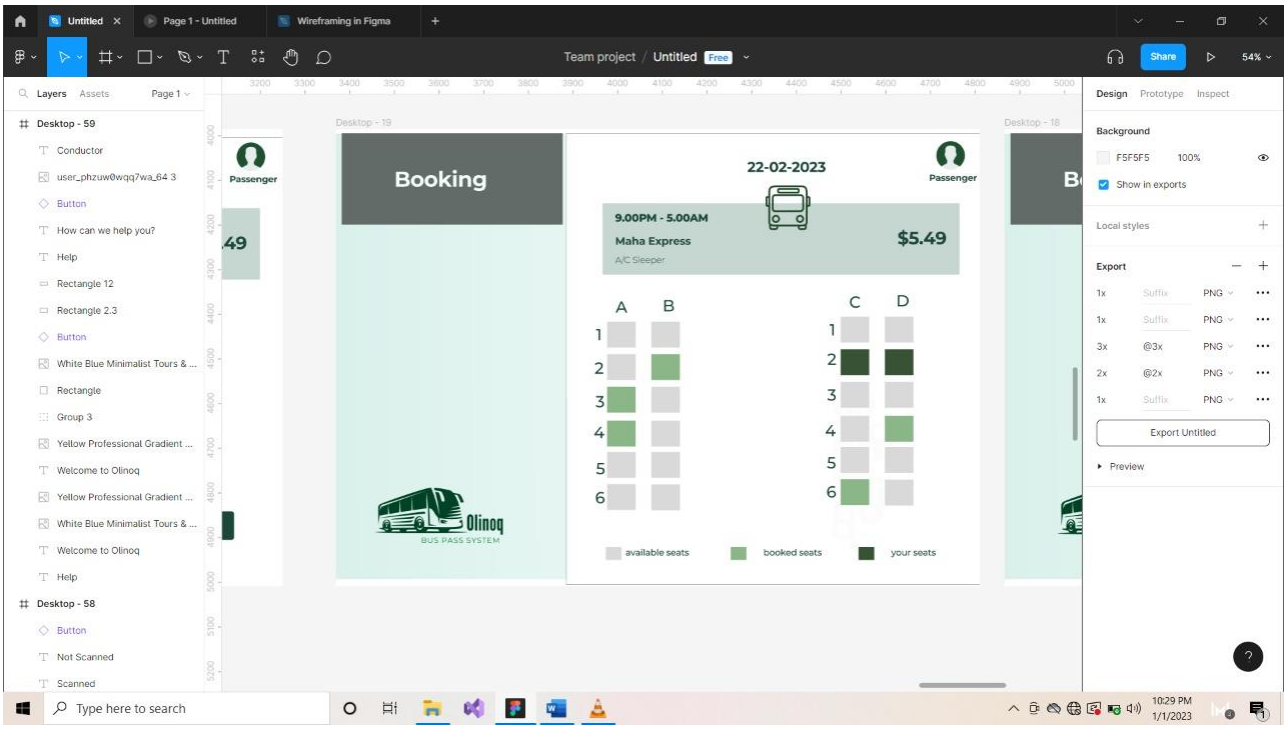


Figure 4 *Passenger Seat booking view*

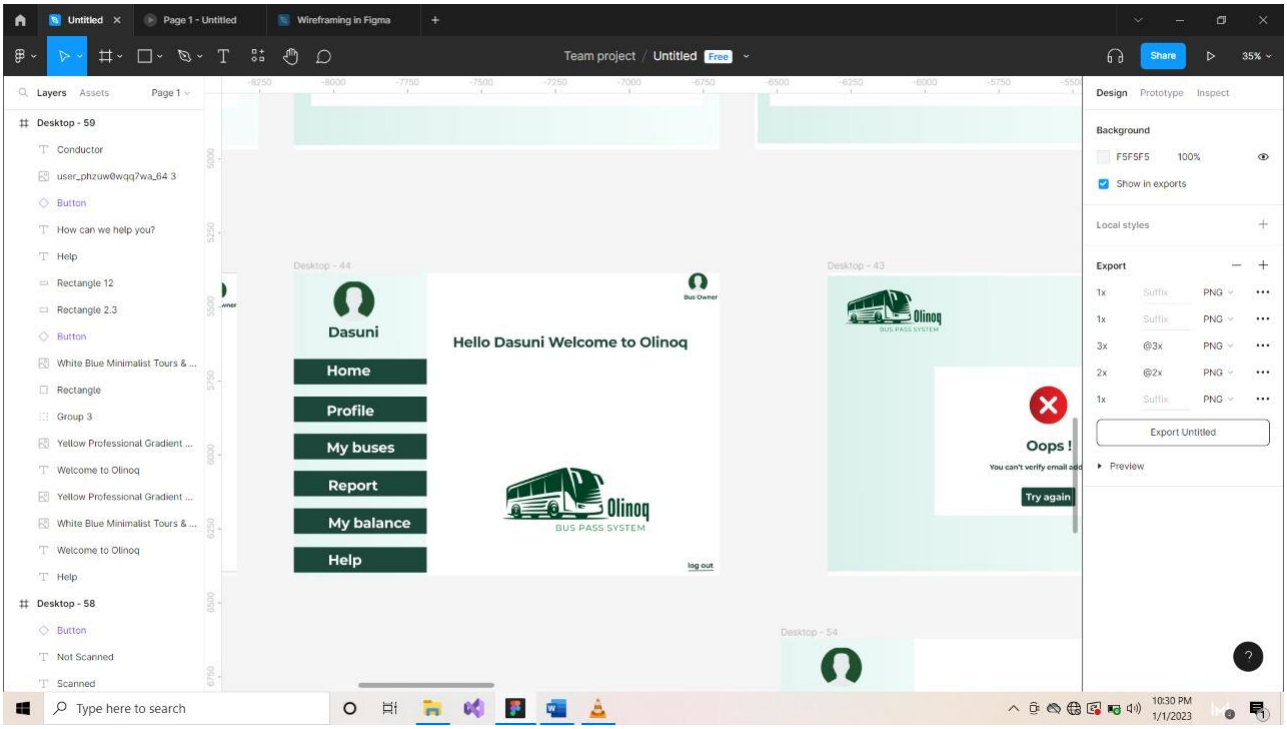


Figure 5 *Bus Owner Dashboard view*

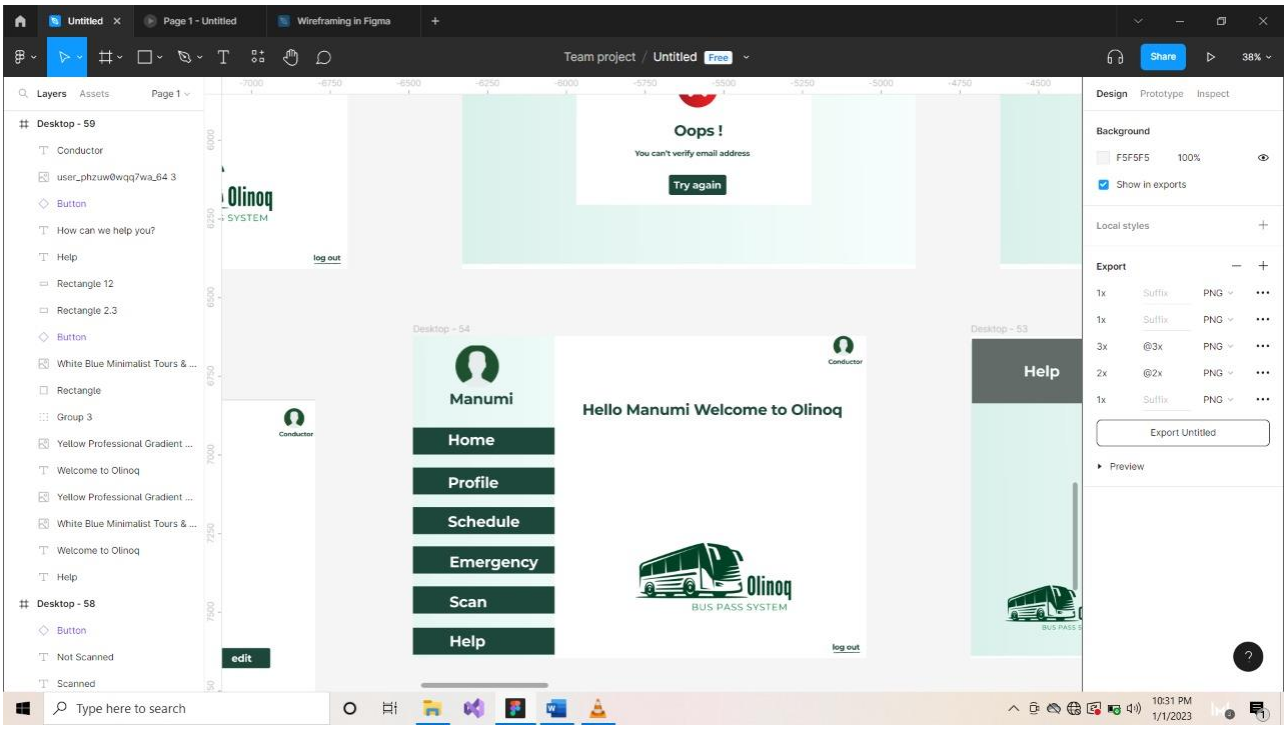


Figure 6 *Conductor Dashboard view*

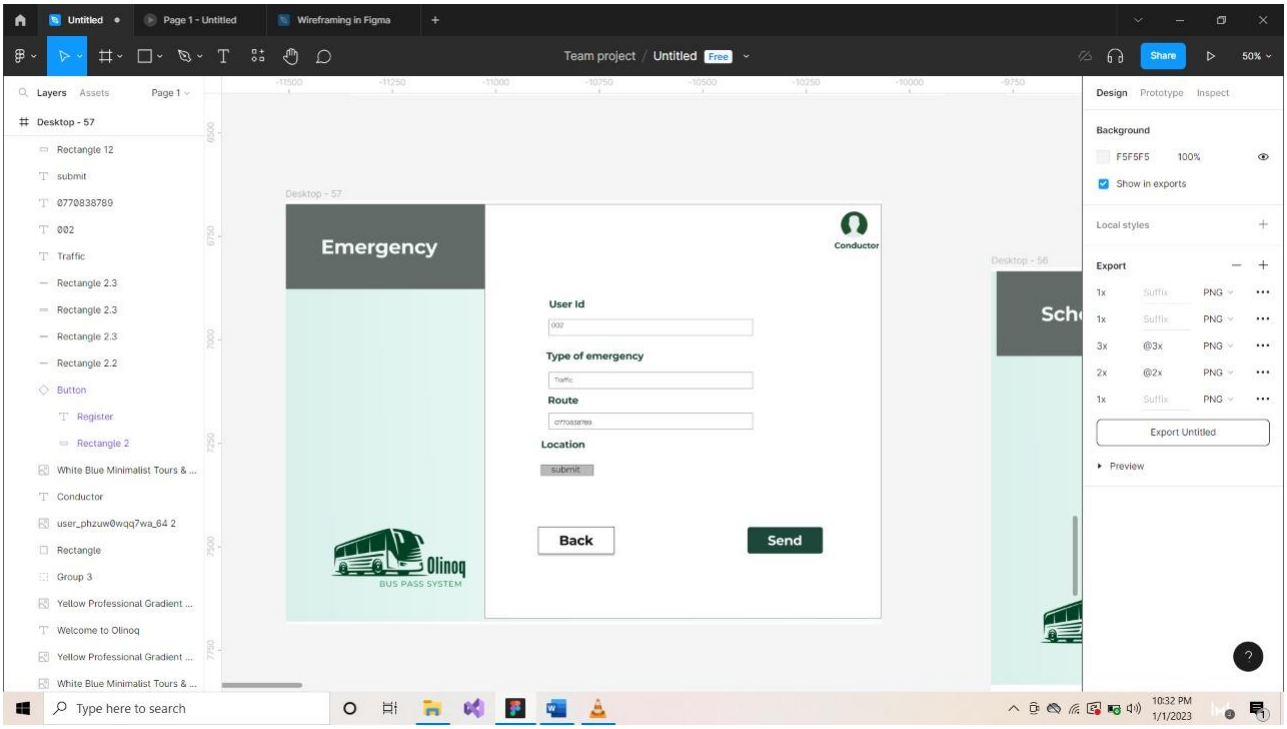


Figure 7 *Conductor Informing emergencies view*

4.2 Hardware Interfaces

In this system we expect to implement it as a web application. So, any device can work with any kind of web browser, this system can also access without any interruption

This system does need specific internet speed or extra CPU or GPU power. There normal data processing exists. So that device depends on some hardware specification for normal browsing. It will be the minimum hardware requirement for running this

4.3 Software Interfaces

The Cloud- Based Bus Pass system runs on web browsers operating in Windows 7 or above, MAC OS, and Linux operating systems with progressive enhancement to enhance the HTML used to develop the system. The system uses MySQL for the database of the web application and HTML, CSS with React (JavaScript framework) for the front end. The JavaScript frameworks such as Node.js and Express.js have been used to develop server-side communication with the software and the database. AWS is using as the cloud provider.

4.4 Communications Interfaces

Communication interface

We host the project in a cloud system

Application and the backend communication happen on HTTPS Web request

When sending and saving passwords we use each related encrypting methods

5. Other Nonfunctional Requirements

5.1 Performance Requirements

In this web application, Users can book their seats conveniently and have 24/7 services to passengers. Easing the ticketing process and seat distribution to the conductors and notifying the user about seat availability. Reducing the paperwork (manual bookkeeping) and staff, reducing expenses.

5.2 Safety Requirements

This web application should use secure communication between the client and server using HTTPS, to ensure that data transmitted between the client and server is encrypted and cannot be intercepted by third parties. This web application should also use secure storage of user credentials, using a password hashing function to store passwords rather than storing them in plain text or using reversible encryption

5.3 Security Requirements

- firewall protection
- web application firewall protection
- server security hardening
- communication channels identified for https
- point to point encryption

5.4 Software Quality Attributes

Our system is platform-independent. Whether it is on the computer, on the tablet or phone, the tool can reformat itself to any screen size. Therefore, it is a mobile-friendly tool. As well as it adapts to the current situation and it is flexible. As well as easy to maintain the System. Apart from above qualities, it has more features such as availability, correctness, interoperability, reliability, reusability, robustness, testability, and usability.

6. Other Requirements-Manumi

6.1 User Stories

Passenger

- As a passenger, I want to sign into the system, so that I can get more facilities like booking, payments etc.
- As a passenger, I want to book, cancel and renew the bus ticket booking through online, so that I can reduce my time and money.
- As a passenger, I want to know about bus fares and do payment online; so that I can have effective money transaction (can get correct balance accurately).
- As a passenger, I want to get a QR code, so that i can prove my payment and use as bus ticket.
- As a passenger, I want to know about time schedules of journeys with routes, so that I can select preferred destination with time at any time anywhere.
- As a passenger, I want to aware about emergency and solution so that I can safely enjoy the journey.
- As a passenger, I want to ask inquiries, so that I can get solutions for my problems when using this system.

Bus Owner

- As a bus owner, I want to sign into the system, so that I can register my buses.
- As a bus owner, I want to view income reports, so that I can understand how money transaction happen and improve the income.
- As a bus owner, I want to register the bus in the system, so that I can earn income properly and passengers can understand what are the facilities that my bus has
- As a bus owner, I want to know about time schedules of buses, so that I can aware about where buses are in.
- As a bus owner, I want to aware about the situation of my bus when facing an emergency, so that I can make a solution for that immediately.
- As bus owner, I want to ask inquiries, so that I can get solutions for my problems when using this system.

Conductor

- As a conductor, I want to sign into the system and view the time schedule of buses, so that I can get to know about my schedule.
- As a conductor, I want to inform the emergency and track the location of the bus, so that time keeper and passenger can aware of the emergency situation.
- As a conductor, I want to scan the QR code, so that passenger can seat and verify the passenger's payment.
- As a conductor, I want to ask inquiries, so that I can get solutions for my problems when using this system.

Appendix A: Glossary

Abbreviation	Meaning
UI	User Interface
DBMS	Database Management System
SQL	Structured Query Language
SMS	Short Message Service

Appendix B: Analysis Models

1. Usecase Diagram:

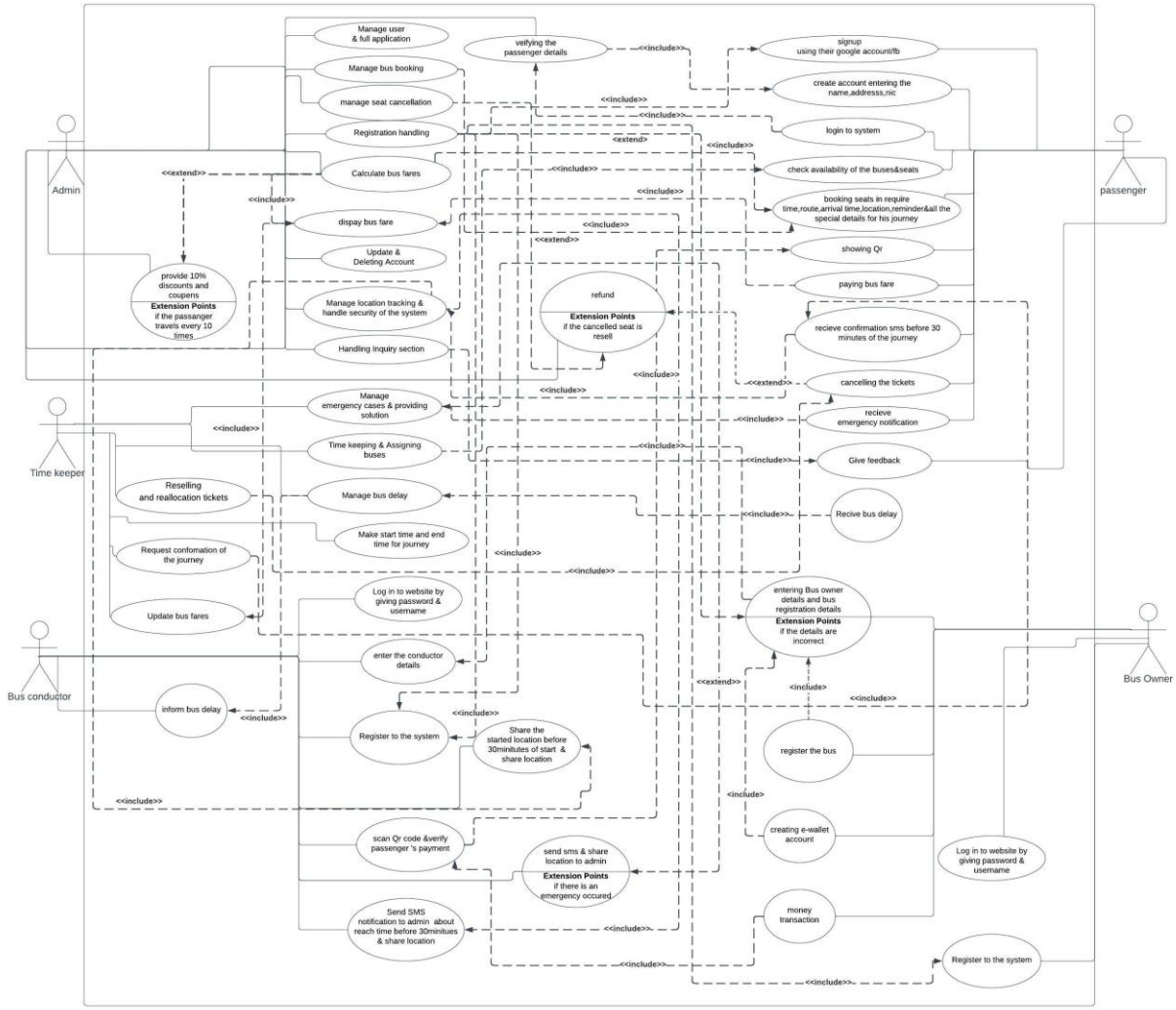


Figure 8 *Passenger dashboard view*

2. Activity Diagram:

2.1 Passenger

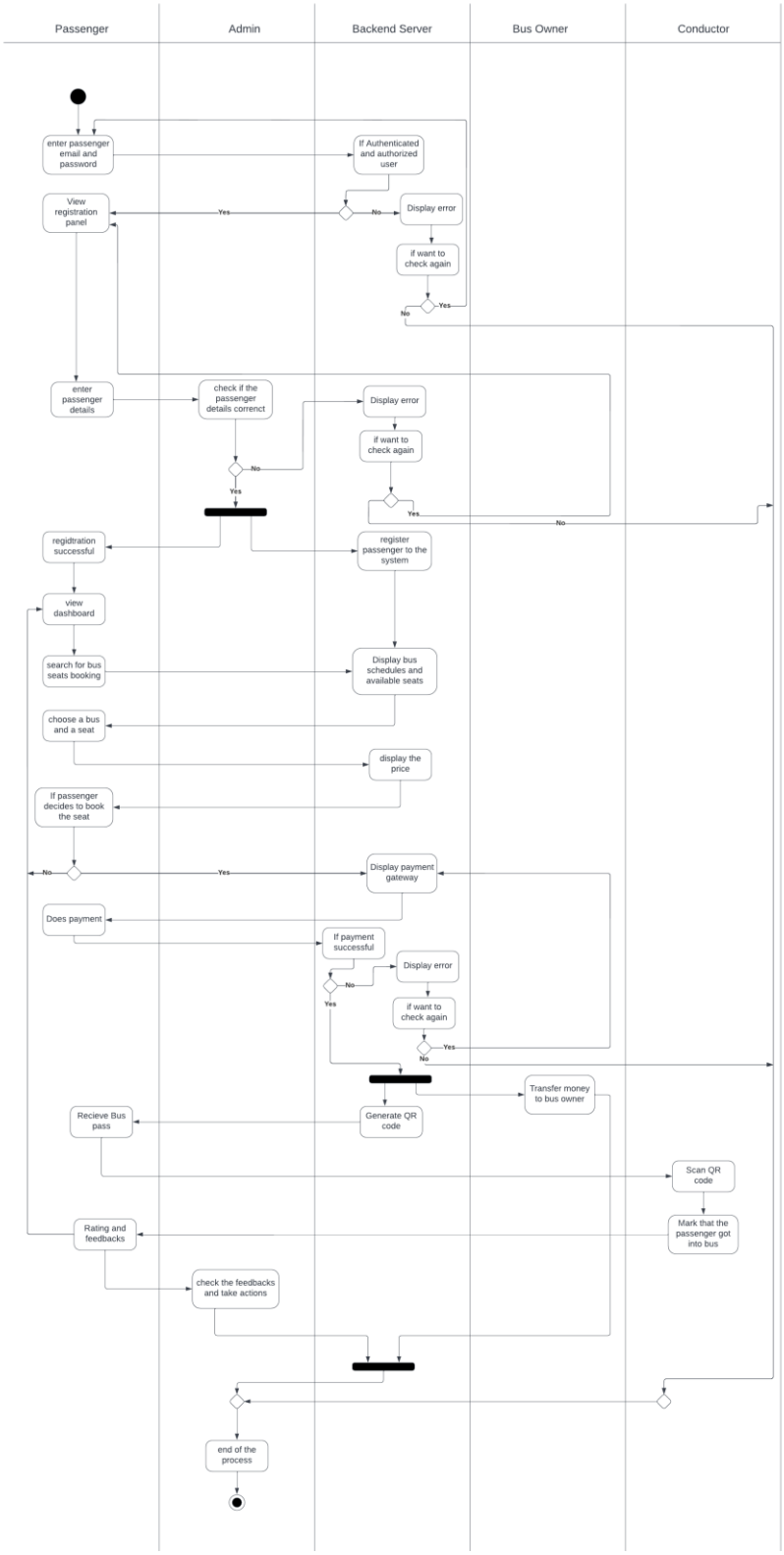


Figure 9 *Activity Diagram: Passenger*

2.2 Bus Owner

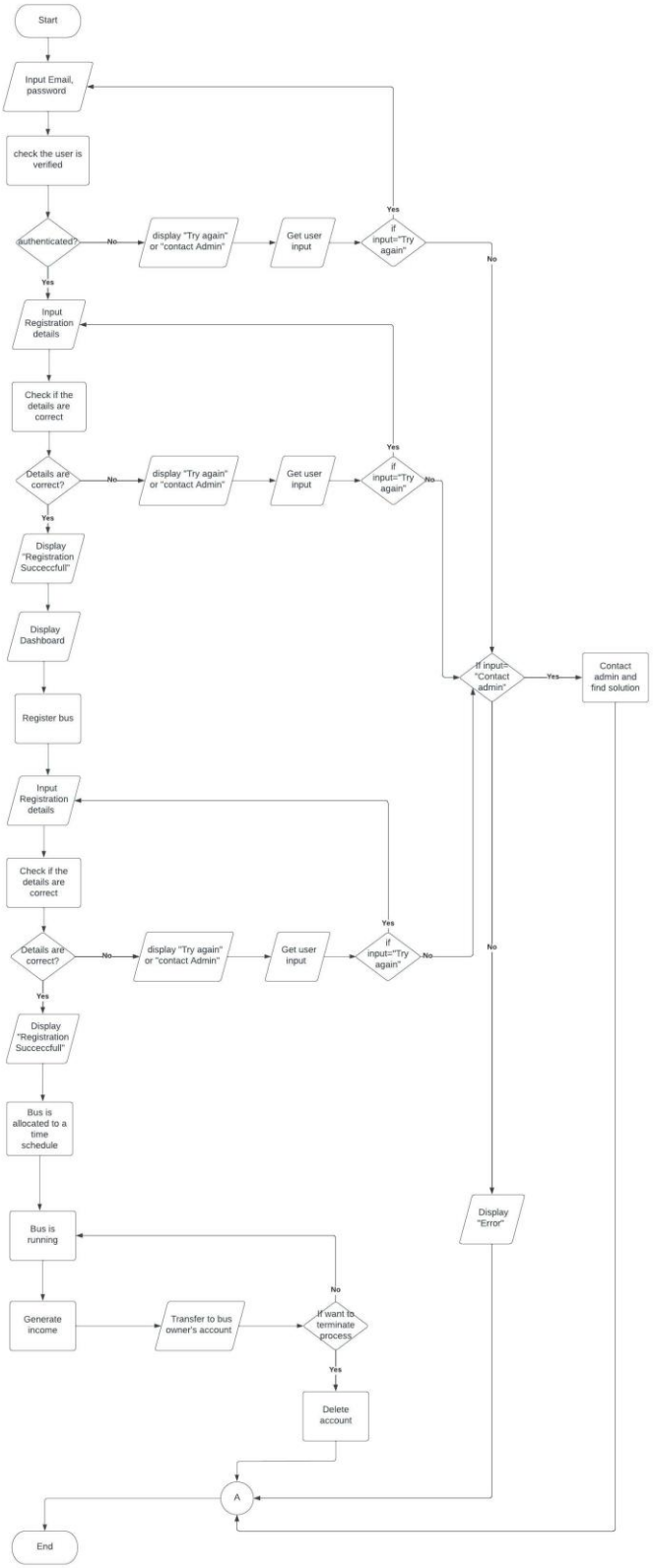


Figure 10 *Activity Diagram: Bus Owner*

2.3 Conductor

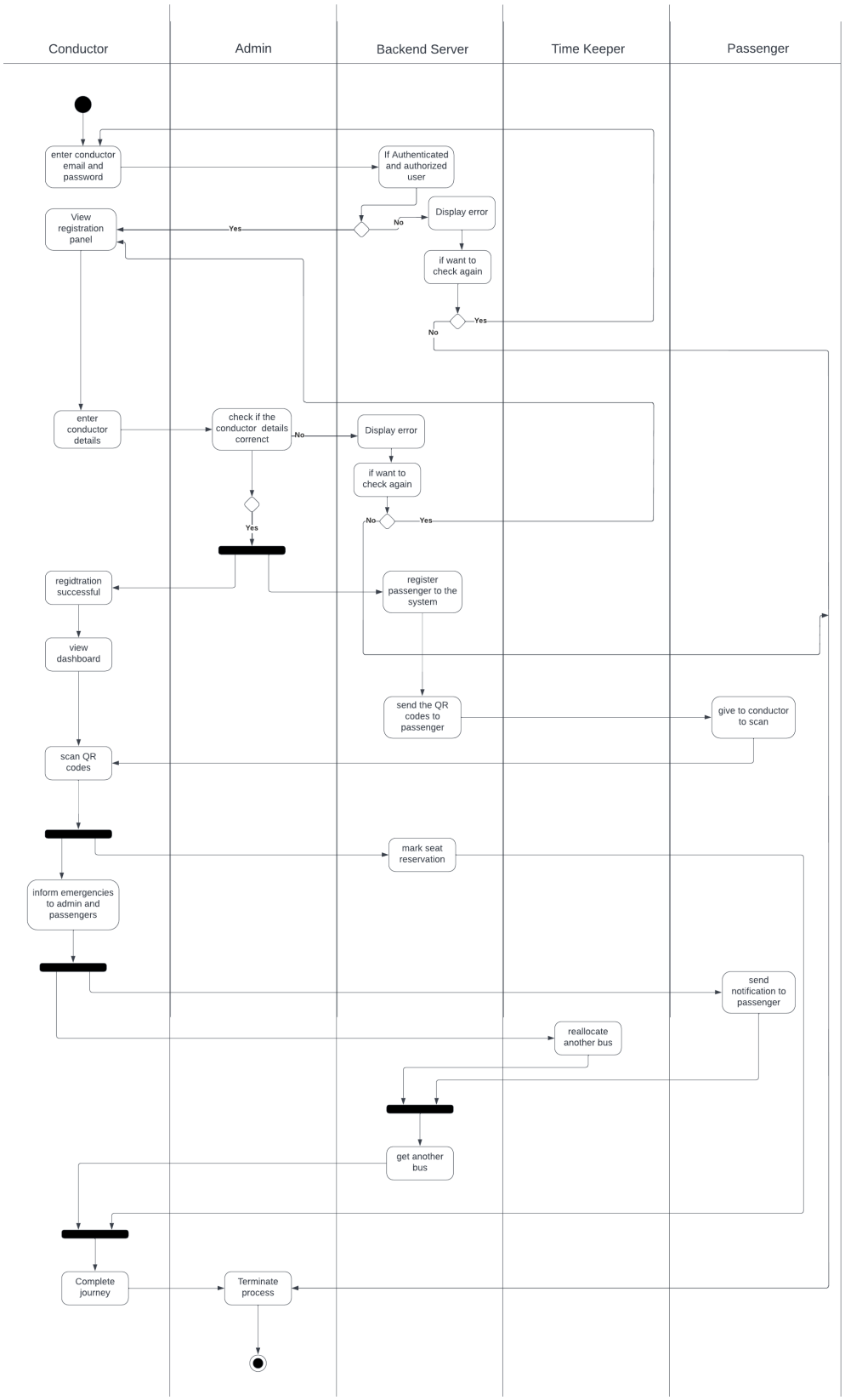


Figure 11 *Activity Diagram: Conductor*

2.4 Admin

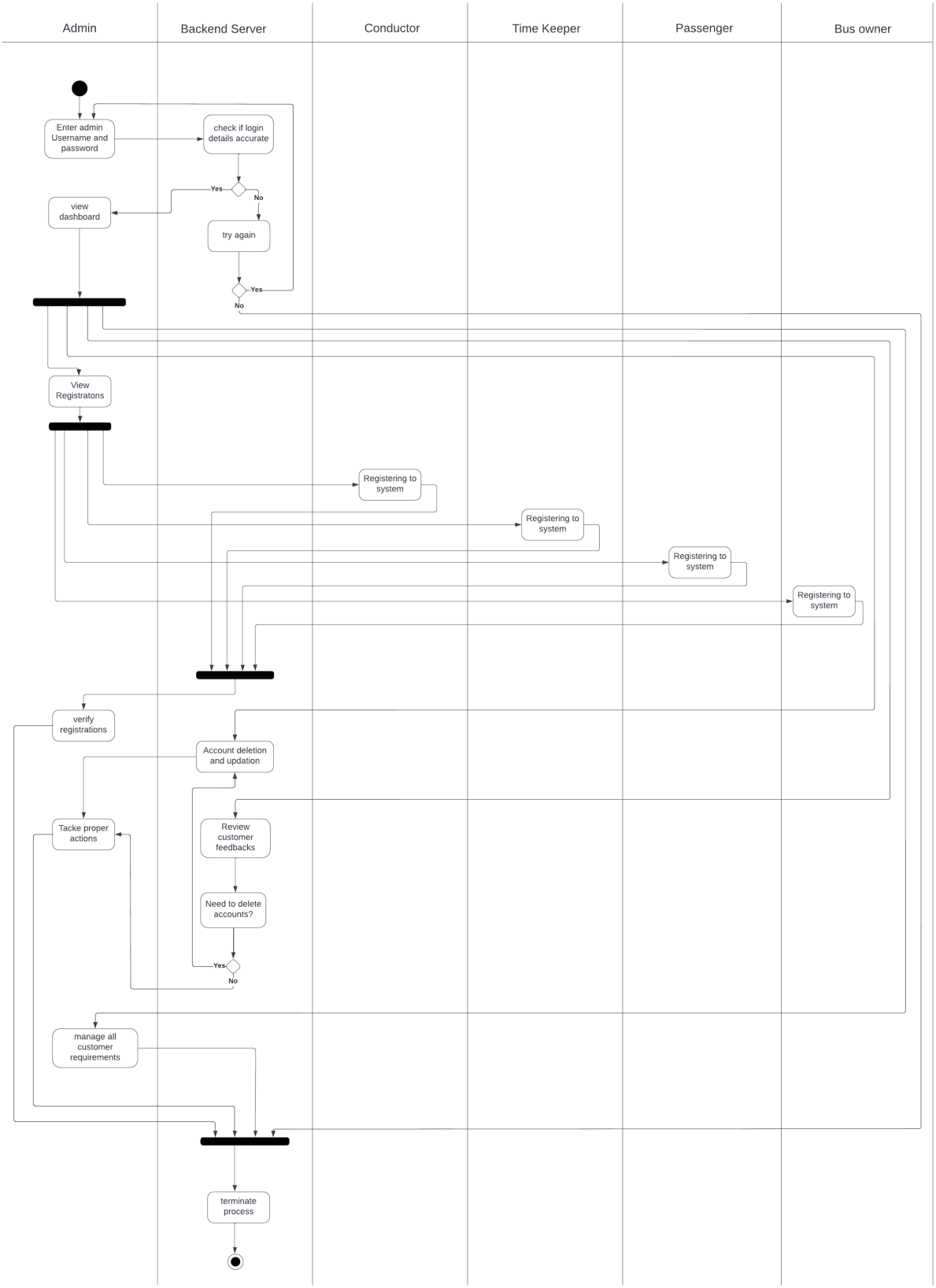


Figure 12 *Activity Diagram: Admin*

2.5 Timekeeper

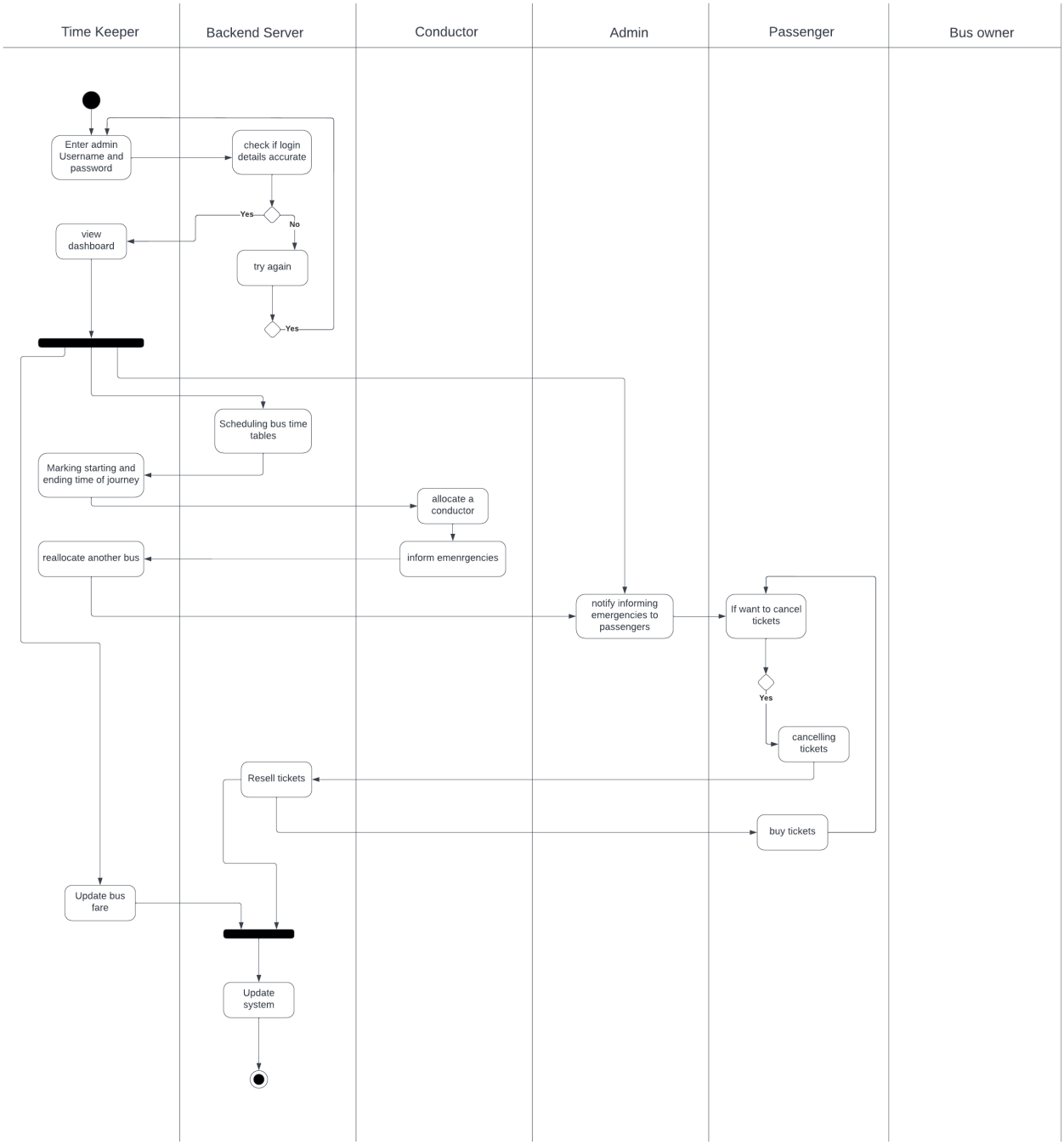


Figure 13 *Activity Diagram: Timekeeper*

3. Class Diagram

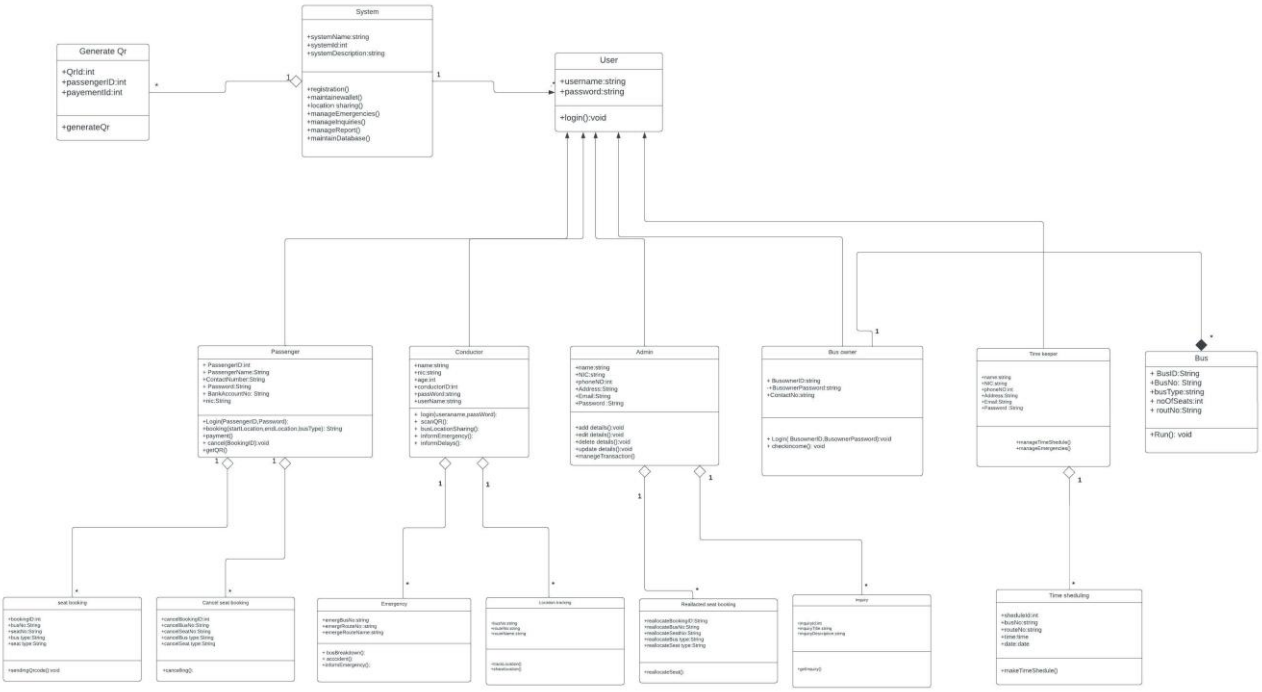


Figure 14 *Class Diagram*

4. ER Diagram

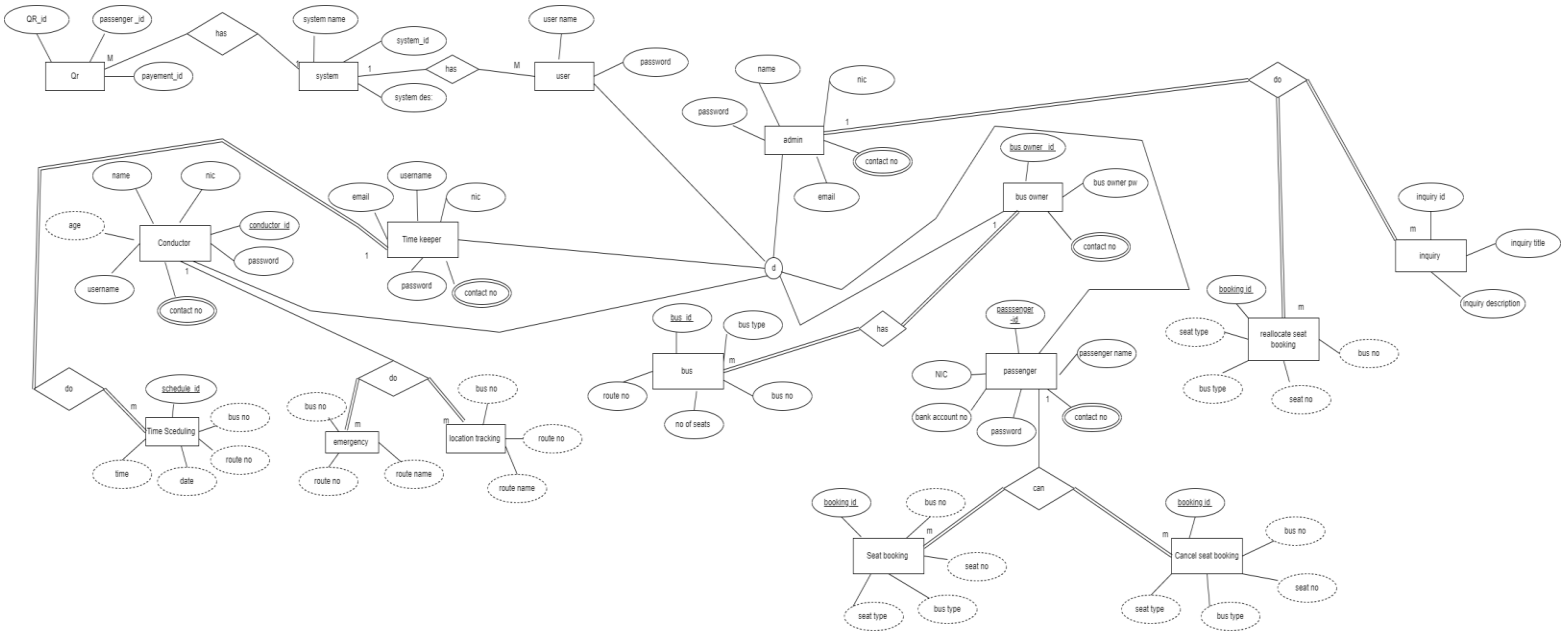


Figure 15 *ER Diagram*

5. Sequential Diagram

5.1 Admin

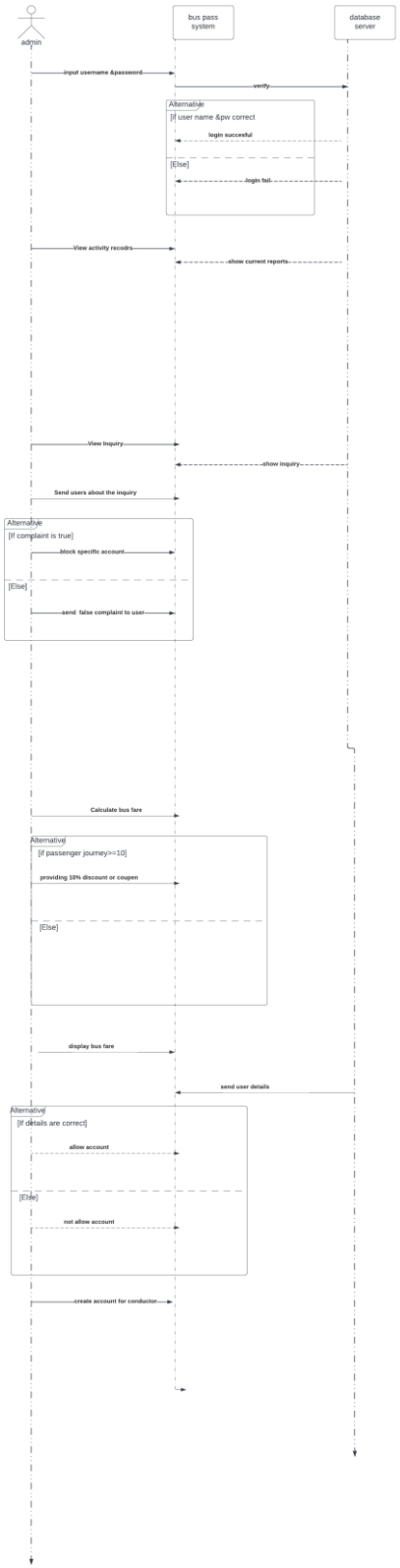


Figure 16 *Sequential Diagram: Admin*

5.2 Timekeeper

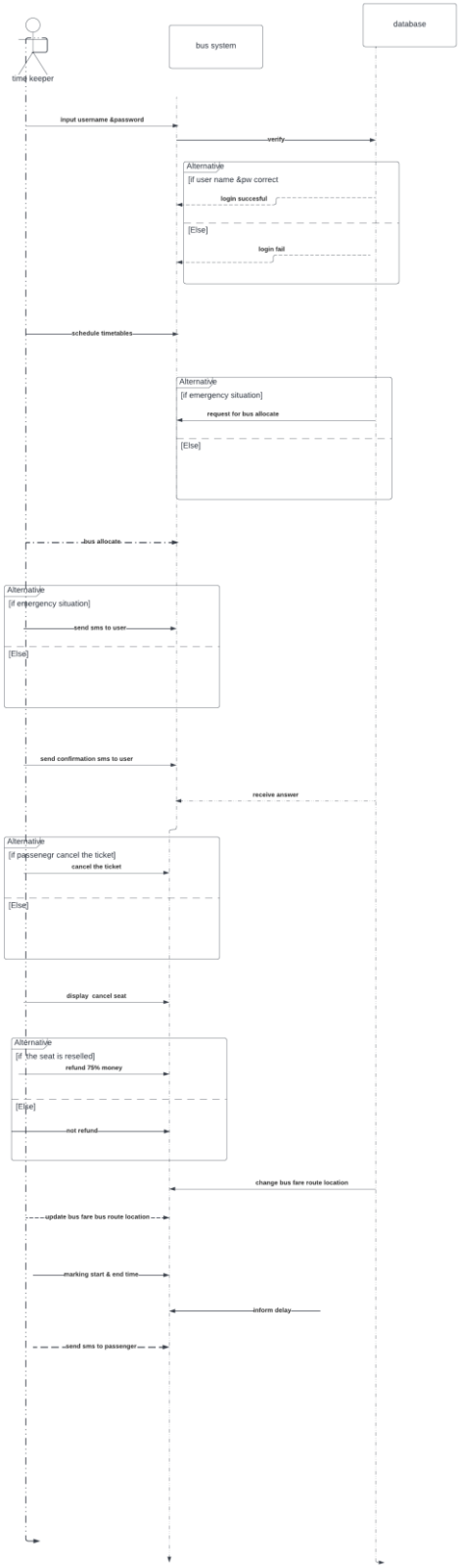


Figure 17 *Sequential Diagram: Timekeeper*