

On-Time Bus



Developed By

Muhammad Aown Ali.

3971-FBAS/BSCS4/F18

&

Muhammad Usman Karamat

4076-FBAS/BSCS4/F18

BS Computer Science

Supervised by

Sir Majid Bashir

Lecturer

Department of Computer Science & Software Engineering Faculty
of Computing

International Islamic University, Islamabad

2022

Department of Computer Science and Software Engineering
International Islamic University, Islamabad.

Final Approval

Dated: **22-August-2022**

It is certified that we have read the project report submitted by **Muhammad Aown Ali , 3971 FBAS/BSCS4/F18** & **Muhammad Usman Karamat , 4076-FBAS/BSCS4/F18** and we judge that this project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the BS Degree in Computer Science/Software Engineering.

Committee:

External Examiner

Mr. Ahsan Tanveer
Lecturer
Department of Computer Science & Software Engineering
International Islamic University, Islamabad.

Internal Examiner

Sayed Qaiser Ali Shah
Lecturer
Department of Computer Science & Software Engineering
International Islamic University, Islamabad.

Supervisor

Mr. Majid Bashir
Lecturer
Department of Computer Science & Software Engineering
International Islamic University, Islamabad.

Dissertation

A dissertation submitted to the Department of Computer Science & Software Engineering, International Islamic University, Islamabad, as partial fulfillment of the requirements for the award of the degree of Bachelors in Software Engineering/Computer Science.

Dedication

We dedicate this project to our beloved parents, without their backing and consolidation it would be impossible to achieve this, moreover, we would also like to all teachers especially our supervisor for helping and all those who prayed for our success.

Declaration

I hereby declare that this Software, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that I have developed this Software entirely based on my efforts made under the sincere guidance of my teachers and supervisor.

No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Muhammad Aown Ali.

3971-FBAS/BSCS4/F18

&

Muhammad Usman Karamat.

4076-FBAS/BSCS4/F18

Acknowledgment

All praise to Almighty Allah, who gave me the understanding, courage, and patience to complete this project.

Thanks to my parents and all well-wishers, who helped me during my most difficult times, and it is due to their untiring efforts that I am in this position today.

I express my gratitude to my kind Teacher **Mr. Majid Bashir** for providing me the opportunity to learn and enhance my knowledge. He had been ready to help and guide me throughout the project.

Project in Brief

| | |
|------------------------|---|
| Project Title: | On-Time Bus |
| Objective: | This project is chosen to ease the life of travelers daily. |
| Undertaken By: | Muhammad Aown Ali (3971-FBAS-BSCS-F18) Muhammad Usman Karamat (4076-FBAS-BSCS-F18) |
| Supervised By: | Mr. Majid Bashir Lecturer |
| Date Started: | October 2021 |
| Date Completed: | August 2022 |
| Tools Used: | <ul style="list-style-type: none">• Visual Studio Code• React Native CLI• React• No SQL• JavaScript• Node Mcu• Expo- CLI• Figma• Draw.IO• RFID• Arduino |
| System Used: | Intel Core i5 6 th Generation |

Abstract

Across the world, automation of most manual systems has already taken place. Especially in the world of automobiles and transportation, automation and technology have become a significant part of the industry without which the industry cannot imagine a future. However, in Pakistan, we can observe a lack of innovation and lack of technological advancements which can prove to be highly beneficial for the public as well as businesses. To solve such a problem, we have come up with a brilliant “Soft-Solution” which would ease the lives of millions daily.

Table of Contents

Contents

CHAPTER 1 INTRODUCTION.....1

1.1. Project Motivation..... 2

1.2. Scope2

1.3. Existing Systems.....2

1.4. Objectives..... 2

1.4.1 Keeping Travelers Informed.....2

1.4.2 Keeping Service Operators in touch with their fleet.....3

1.4.3 Pay as You Go.....3

1.4.4 Management Services.....3

CHAPTER 2 EXISTING SYSTEM.....4

2.1. Existing System..... 5

2.2 Problem Statement.....5

2.3 Proposed System.....5

2.4 Existing System.....6

2.4.1SVWL.....6

2.4.2 Careem.....6

CHAPPTER 3 SYSTEM ANALYSIS.....7

3.1. Users8

3.1.1. Passengers.....8

3.1.2. Bus-Operators.....8

3.2. Requirements Analysis.....8

3.2.1. Objectives8

3.2.2. Constraints8

3.2.3. Dependencies9

3.3. Platform Requirements.....9

3.3.1. Supportive Operating System.....9

- 3.4. Software requirements.....9
 - 3.4.1. Front-End Software Requirements.....9
 - 3.4.2. Back-End Software Requirements.....9
- 3.5 Hardware Requirements9
 - 3.5.1 Hardware Requirements for Implementation.....9
 - 3.5.2 Hardware Requirements for Deployment.....9
- 3.6. Other Requirements.....9
 - 3.6.1 Functional Requirements.....9
 - 3.6.2 Non-Functional Requirements.....10
 - 3.6.3 Modules.....11
- 3.7. Use Case Diagrams.....12
- 3.8 Fully Dressed Use Cases.....13
 - 3.8.1. Registration.....13
 - 3.8.2. Login.....13
 - 3.8.3. Touch On/Off.....14
 - 3.8.4. Vehicle Tracking.....15
 - 3.8.5. View Wallet.....15
 - 3.8.6. View Schedule.....16
 - 3.8.7. Manage Route.....16
 - 3.8.8. Manage Buses/Fleet17
 - 3.8.9. Manage Bus Stop.....17
 - 3.8.10. Manage Bus Schedule.....18
 - 3.8.11. Manage Users.....19
- 3.9 System Sequence Diagrams.20
 - 3.9.1. Login.....20
 - 3.9.2. Registration.....21
 - 3.9.3. Touch On/Off.....22
 - 3.9.4. Vehicle Tracking.....22
 - 3.9.5. View Schedule23

- 3.9.6. View Wallet23
- 3.9.7. Manage Bus.....24
- 3.9.8. Manage Route25
- 3.9.9. Manage Schedule.....26
- 3.9.10. Manage Bus Stop.....27
- 3.9.11. Manage Users.....28
- 3.10 Domain Model.....29
- CHAPTER 4 SYSTEM DESIGN.....30**
- 4.1. Sequence Diagram.....31
 - 4.1.1 User.....31
 - 4.1.2 Bus Operator/Company/Admin32
- 4.2 Class Diagram33
- 4.3 Activity Diagram.....34
- 4.4 Entity Relationship Diagram.....35
- CHAPTER 5 IMPLEMENTATION.....36**
- 5.1 System Implementation.....37
- 5.2 Functional features.....37
- 5.3 Tools and technologies.....37
 - 5.3.1 React.....37
 - 5.3.2 React-Native.....37
 - 5.3.3 Database.....37
 - 5.3.4 Arduino.....38
 - 5.3.5 Microsoft Visual Studio Code.....38
- CHAPTER 6 TESTING.....39**
- 6.1 Testing.....40
 - 6.1.1 Component of testing.....40
 - 6.1.2 Integration of testing.....40
 - 6.1.3 User Testing.....40
 - 6.1.4 Test case scenarios.....40

| | |
|-------------------------------------|-----------|
| 6.1.5 Goal achieved..... | 42 |
| CHAPTER 7 CONCLUSION..... | 43 |
| 7.1 Conclusion and future work..... | 44 |
| 7.1.1Conclusion..... | 44 |
| 7.1.2Limitation..... | 44 |
| . | |

List of Figures

Figure 3.1 Use Case Diagram.....11

Figure 3.2 Login.....19

Figure 3.3 Signup.....20

Figure 3.4 Touch On/Off21

Figure 3.5 Track Vehicle.....21

Figure 3.6 View Schedule.....22

Figure 3.7 View Wallet.....22

Figure 3.8 Manage Bus.....23

Figure 3.9 Manage Route.....24

Figure 3.10 Manage Schedule.....25

Figure 3.11 Manage Bus Stop.....26

Figure 3.12 Manage User Profile.....27

Figure 3.13 Domain Model28

Figure 4.1 Sequence Diagram of User.....30

Figure 4.2 Sequence Diagram Bus/Company/Admin.....31

Figure 4.3 Class Diagram....32

Figure 4.4 Activity Diagram.....33

Figure 4.5 Entity Relationship Diagram.....34

List of Tables

| | |
|------------------------------------|----|
| Table 3.1 Signup..... | 12 |
| Table 3.2 Login..... | 12 |
| Table 3.3 Touch On/Off | 13 |
| Table 3.4 Vehicle Tracking..... | 14 |
| Table 3.5 View Wallet..... | 15 |
| Table 3.6 View Schedule..... | 15 |
| Table 3.7 Manage Routes..... | 16 |
| Table 3.8 Manage Buses/Fleet | 16 |
| Table 3.9 ManageBusStop..... | 17 |
| Table 3.10 Bus Schedule..... | 17 |
| Table 3.11 Manage User..... | 18 |
| Table 6.1 Test Case..... | 41 |

CHAPTER 1
INTRODUCTION

1. Introduction

Across the world, automation of most manual systems has already taken place. Especially in the world of automobiles and transportation, automation and technology have become a significant part of the industry without which the industry cannot imagine a future. However, in Pakistan, we can observe a lack of innovation and lack of technological advancements which can prove to be highly beneficial for the public as well as businesses. To solve such a problem, we have come up with a brilliant “Soft-Solution” which would ease the lives of millions daily.

1.1 Project Motivation

The main purpose of this project is to bring ease into the lives of travelers who use the outdated transport system daily. Our aim of ours is to bring innovation into this field which is already underway in countries other than ours. Thus, there is a need to launch this project in this region to get this country up to speed in this field. This project will be able to revolutionize the way public transport operates. The clients of this app will be able to add value to their businesses.

1.2 Scope

The application of this system will be in the transportation industry. As there is a need to bring forth automation and innovation in the field of people moving. Hence, there is a need to launch this project in this region to get this country up to speed in this field. This system does exactly that. On-Time Bus is a smart bus scheduling system that allows users to view, track and route their journey on given bus service. This system also acts as a reminder system for the drivers to allow them to be punctual and track their progress on a given route via using GPS. This system would also enable the business owners/managers to manage their running buses/ routes/ drivers. They could also track the buses running on a particular route.

1.3 Existing Systems

Certain transportation apps are currently in business in this region, some of which include: SWVL, Careem, Uber, etc. Even though these systems are similar in terms of the area of an industry being implemented, these systems are missing some of the aspects which are being provided in the On-Time Bus service. Our service is being provided to the public to check the bus Schedule and track the progress of the bus on its respective route. This platform provides the customers and the bus operators with a completely stand-alone and all-in-all system including payment, journey management, tracking, etc. but can also be implemented as an extra feature and a convenience of the bus operator’s business which would increase the worth of the company as a business in the eyes of the customers.

1.4 Objectives

The On-Time-Bus system acts as an intermediary between the end-user (passenger) and the service provider (Business owner/Manager) and fills the gaps in communication using the latest technology, which is not necessarily human-to-human communication. With the development of this system, we aim to take out the “guesswork” and ambiguity involved in the automobile industry, for example guessing how long it would take for a bus to arrive at a particular stop.

1.4.1 Keeping Travelers informed:

The main objective of this system is to eliminate the feeling of isolation when waiting for the bus. No more asking other people, no more fidgeting, no more anxiousness. All of that goes away with the tap of a button. The users will be able to view their current location and find the nearest bus stop. The users can look for the schedule of the next arriving bus. Additionally, the system will also keep the users informed about the whereabouts of the bus in real-time.

1.4.2 Keeping Service Operators in touch with their Fleet:

The bus operators go through a lot of hassle while chasing their fleet. They must constantly keep calling the drivers to let them know their location. With GPS-enabled buses. All they need to do is to log on to their systems and the buses will appear right in front of their eyes. This reduces botheration on both sides, the bus drivers as well as the service operators.

1.4.3 Pay as you go:

The travelers will be able to travel without the worry of prior bookings, taking time out to enquiring the schedules and being ready before they need to travel. This system will allow them to “Touch-On and Touch-Off” as they are ready to travel, and they are present at the designated bus stops. All they will need to do is to scan a QR code present at the bus stop, and the system will register that as the start of their journey. While disembarking from the bus, they need to Scan the QR code again and the system will register that as the end of that journey.

With the pre-coded algorithm, the system will calculate the fare of the journey and automatically deduct that amount from the users’ virtual wallets. This will allow fairness for both; service providers and the customers, as they will only be charged for the distance that they have traveled.

1.4.4 Management Services:

The On-Time Bus will allow the bus operators to manage their fleet, assign fleet to routes, Manage routes and manage the regions that their service extends.

CHAPTER 2
EXISTING SYSTEM

2.1 Existing Systems:

Certain transportation apps are currently in business in this region, some of which include: SWVL, Careem, Uber, etc. Even though these systems are similar in terms of the area of an industry being implemented, these systems are missing some of the aspects which are being provided in the On-Time Bus service. Our service is being provided to the general public to check the bus Schedule and track the progress of the bus on its respective route. This platform provides the customers and the bus operators with a completely stand-alone and all-in-all system including payment, journey management, tracking, etc. but can also be implemented as an extra feature and a convenience of the bus operator's business which would increase the worth of the company as a business in the eyes of the customers.

2.2 Problem Statement:

The automation of most of the manual systems in most industries has already taken place. Especially in the world of automobiles and transportation, automation and technology have become a significant part of the industry without which the industry cannot imagine a future. However, in Pakistan, we can observe a lack of innovation and lack of technological advancements which can prove to be highly beneficial for the general public as well as businesses. To solve such a problem, we have come up with a brilliant "Soft-Solution" which would ease the lives of millions daily.

2.3 Proposed System:

We have come up with a solution to the above-defined problems which are faced on the daily basis by travelers all across the country and countless different bus operating companies. The solution is a system which we have decided to call the "On-Time Bus" system. The basic way to describe this system is by saying that system is a connecting platform for many travelers who do not have any means of receiving the necessary information from the bus operator such as the schedule. What this system does is provide them with the necessary information. Therefore, reducing the ambiguity and waiting time the users spend while waiting for the bus. The system will allow them to only set out whenever the bus arrives, and not necessarily hours before the bus will arrive. The system will also allow the users to pay as they go. The travelers will be able to travel without the worry of prior bookings, taking time out to enquiring the schedules and being ready before they need to travel. This system will allow them to "Touch-On and Touch-Off" as they are ready to travel, and they are present at the designated bus stops. All they will need to do is to scan a QR code present at the bus stop, and the system will register that as the start of their journey. While disembarking from the bus, they need to Scan the QR code again and the system will register that as the end of that particular journey. With the pre-coded algorithm, the system will calculate the fare of the journey and automatically deduct that amount from the users' virtual wallets. On the other hand, the system will also allow the bus service operators to be in touch with their fleet. The On-Time Bus will also let the bus operators manage their fleet, assign fleet to routes, manage routes and manage the regions that their service extends.

2.4 Existing Systems:

Certain transportation apps are currently in business in this region, some of which include: SWVL, Careem, Uber, etc. Even though these systems are similar in terms of the area of an industry being implemented, these systems are missing some of the aspects which are being provided in the On-Time Bus service. Our service is being provided to the general public to check the bus Schedule and track the progress of the bus on its respective route. This platform provides the customers and the bus operators with a completely stand-alone and all-in-all system including payment, journey management, tracking, etc. but can also be implemented as an extra feature and a convenience of the bus operator's business which would increase the worth of the company as a business in the eyes of the customers.

2.4.1 SWVL: This company describes itself as: "A tech-driven, affordable, and convenient transportation service, committed to enabling cities, people, and businesses to move everywhere." SWVL is a solely owned company that implements its system and is heavily and entirely dependent on the operation of its system. Our System on the contrary can be implemented by any bus operator/company and provides certain features which add to their already existing manual setup with minimal hindrance to their operation while implementation. Also, this system will add to the value of the business by getting automated payments at the time of customers embarking and disembarking.

2.4.2 Careem: Careem is a private cab booking platform that connects traveling individuals with drivers. This system has very little to no similarity with our system. This platform is a private booking app meanwhile our system is a public transport system.

CHAPTER 3
SYSTEM ANALYSIS

3.1 Users:

Users of this system are described in more detail below:

3.1.1 Passengers:

- **Bus-Scheduling:** The system will allow the users to view the bus schedule to its route and bus stop. The user will also be displayed the nearest bus stops which are approachable from the user's geographical location.
- **Bus Tracking:** The user can look up a particular bus stop and view how long it will take for the bus to arrive.
- **Bus-Stop Search:** The user can also search for a particular bus stop using keywords and predictive suggestions.
- **Touch-On/Off system:** The system will also contain a touch-on and off system which will allow "spontaneous customers" to pay for the bus service without any prior booking. The system will have a QR code scanner and a QR code associated with each bus stop.

3.1.2 Bus-Operators:

- **Fleet Management:** This system would allow the business owners/managers to manage their running buses. They could also track the buses running on a particular route.
- **Route Management:** This system would also enable the business owners/managers to manage the routes to which their services extend.
- **User Management:** This system would let the business owners/managers manage their registered clients. They can overview the client's activities in case any anomalies in the system need to be corrected. And also modify their user profiles.
- **Bus stop Management:** This system would also enable the business owners/managers to manage their bus stops.

3.2 Requirement Analysis:

Requirement analysis analyzes, defines, and specifies the requirements needed to build a good application may include the application's functionality, constraints, and goals understandable and acceptable to the client, developer, and user. It also defines the input, output, and dependency of the application.

3.2.1 Objectives:

The On-Time-Bus system acts as an intermediary between the end-user (passenger) and the service provider (Business owner/Manager) and fills the gaps in communication using the latest technology, which is not necessarily human-to-human communication. With the development of

this system, we aim to take out the “guesswork” and ambiguity involved in the automobile industry, for example, guessing how long it would take for a bus to arrive at a particular stop.

3.2.2 Constraints

- React-Native-CLI or Expo-CLI should be installed in the system along with the required libraries need to run and achieve proper functionality of the application
- A good internet connection (WIFI or Mobile Data)
- Android/ IOS mobile
- An application needs to be installed on the Client's and Tradesmen's end also.

3.2.3 Dependencies:

- This system consists of 2 aspects, one of which is a mobile app that will be used by travelers/ customers. The application will be cross-platform and will be able to run on Android and IOS platforms.
- The customers will need to install the client app on their smartphones.
- The bus operator admins will need a web browser to run the management app.

3.3 Platform requirements:

3.3.1 Supportive operating system

The supported Operating system for users includes:

- Windows
- Linux

Windows and Linus are two operating systems that will support the comparative application. A window is a meta-family of graphical operating systems developed consisting of several families of the operating system. This project is developed on windows but compatible with Linux too.

3.4 Software requirements:

3.4.1 Front-end software requirements:

- In this project we use React-Native-CLI offers even more features that enhance your productivity when building cross-platform application, some of the features are as follow
- Cross-Platform: In React Native you write cross-platform native apps using JavaScript in a single codebase for both Android and IOS.
- It saves the development time that is required to build a mobile application for multiple platforms.
- React Native offers a much more simplified UI and reduces the loading time as well.

3.4.2 Backend software requirements:

- Node JS

- Django
- Firebase

3.5 Hardware requirements

3.5.1 Hardware requirements for implementation:

- A device with support to access the current location
- The device with support to access the gallery

3.5.2 Hardware requirements for deployment

- RAM: 4GB minimum or more
- Processor: Intel core i5 5th Gen

3.6 Other requirements:

3.6.1 Functional requirements:

- **Registration:** Clients can register and if not registered already then sign up.
- **Login:** Registered clients can Log in to the system.
- **Display Schedule:** The schedule of the needed bus stop, route, and bus will be displayed.
- **Live Tracking:** The bus can be tracked live for the users to display them concerning the specified bus route.
- **Touch-On/Touch-Off:** The system will also contain a touch-on and off system which will allow “spontaneous customers” to pay for the bus service without any prior booking. The system will have a QR code scanner and a QR code associated with each bus stop.
- **TraWallet:** The system has a feature namely “TraWallet” which is a virtual wallet in the app. Money can be replenished by the users by visiting any near bus office.
- **Generate Bills:** With the pre-coded algorithm, the system will calculate the fare of the journey and automatically deduct that amount from the users’ virtual wallets.
- **Payment Deduction:** With the pre-coded algorithm, the system will calculate the fare of the journey and automatically deduct that amount from the users’ virtual wallets.
- **Manage Profile:** This system would let the business owners/managers manage their registered clients. They can overview the client’s activities in case any anomalies in the system need to be corrected. And also modify their user profiles.
- **Fleet Management:** This system would allow the business owners/managers to manage their running buses. They could also track the buses running on a particular route.
- **Route Management:** This system would also enable the business owners/managers to manage the routes on which their services extend.

- **Bus stop Management:** This system would also enable the business owners/managers to manage their bus stops.

3.6.2 Non-functional requirements

Those requirements are not necessary for the functionality of the system, but if implemented, can improve the quality of the whole system.

- **Performance requirements:** The system is required to work properly under the constraints and requirements mentioned.
- **Availability:** Once the application is installed with the required packages, it is available for use 24/7.
- **Maintainability:** The application is easy and economical to maintain.
- **Portability:** The system can run on any cross-platform phones
- **Security:** The application is secure to use as the user needs to log in to enter the application.
- **Reliability:** This system is user-friendly. Its UI requires less effort to operate.
- **Understandability:** The system is easily understandable as we provide a smooth interface for every user.
- **Usability:** As this is a cross-platform app so it can only be accessible if a user has an active internet connection.

3.6.3 Modules:

- **Passengers:**

1. Registration
2. Login
3. Display Schedule
4. Live Tracking
5. Touch-On/Touch-Off
6. TraWallet
7. Generate Bills
8. Payment Deduction
9. Manage Profile

- **Bus-Operators:**

1. Manage Profile
2. Fleet Management
3. Route Management
4. Bus Stop Management

3.7 Use Case Diagram:

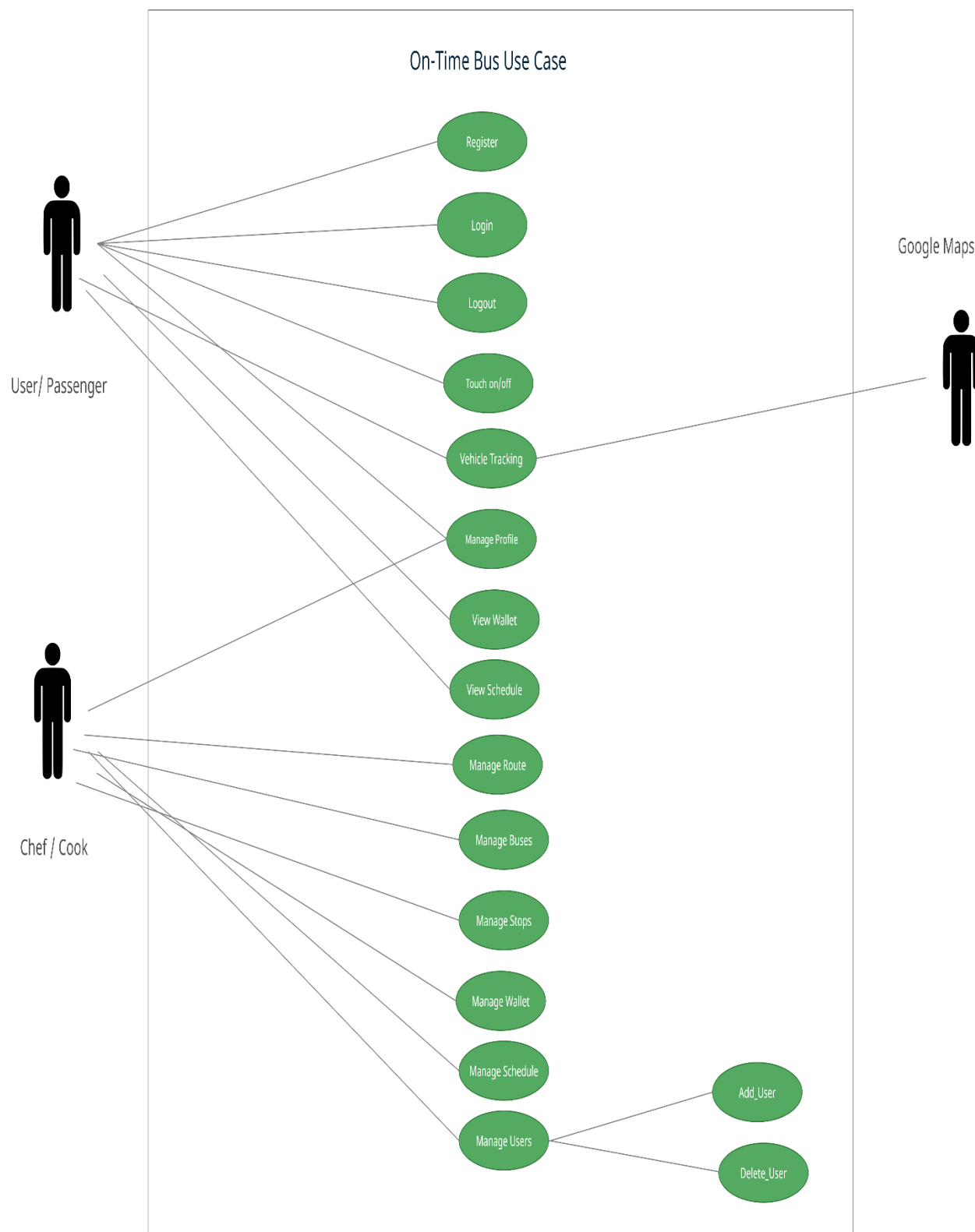


Figure 3.1 Use Case Diagram

3.8 Fully Dressed Use Cases:

3.8.1 Registration:

| | |
|---|--|
| Scope: | On-Time Bus Mobile App (Client) |
| Primary Actor: | Passenger |
| Preconditions: | The user device must be powered on and must be connected to an active internet connection. |
| Success Guarantee / Post Condition: | Signup/Registration successful |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| The user signs up for the application by supplying his username and password credentials. | The app validates & displays the login page of the mobile app. |
| Alternate Scenario: | |
| In case a user is already signed up with the provided username and password. | Prompt the user to supply another email, or password combination or try logging in with the existing id. |
| Frequency: | Once in the lifetime of a user account. |
| Non-Functional Requirements: | Information hiding, response speed. |

Table 3.1 Signup

3.8.2 Login

| | |
|---|--|
| Scope: | On-Time Bus Mobile App (Client) |
| Primary Actor: | Passenger |
| Preconditions: | The user device must be connected to an active internet connection. |
| Success Guarantee / Post Condition: | Prompt successful login + navigate the user to the app home screen. |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| The user logs into an application by supplying his username and password credentials. | Prompt successful login + navigate the user to the app home screen. |
| Alternate Scenario: | |
| The user provides an incorrect combination to his account | Prompt the user to provide an incorrect username/password combination, or ask to reset the password. |
| Frequency: | As many times as the user tries to use the app after an inactive session. |
| Non-Functional Requirements: | Provide the user with security for the login credentials. |

Table 3.2 Login

3.8.3 Touch On/Off

| | |
|--|--|
| Scope: | On-Time Bus Mobile App (Client) |
| Primary Actor: | Passenger |
| Preconditions: | The user device must be connected to an active internet connection. The user must be present at the bus stop. Users must allow the app to use the device Camera to Scan the QR code. The user must have credits in the account wallet;" TraWallet." |
| Success Guarantee / Post Condition: | The user gets counted in on a bus journey. |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| The user scans the QR code provided at each bus stop before embarking on a journey. A user scans the QR code provided at the destination bus stop again after disembarking from the bus towards the end of the journey. | The system automatically creates a journey for the user in the database with the scanned QR code where it will acquire the required information about the origin and destination of the journey and the user will automatically be charged for the respective journey via the TraWallet. |
| Alternate Scenario: | |
| The user unsuccessfully scans the QR code. | Prompt the user to scan the QR code again. |
| The user tries to embark on a journey without an active session. | Prompt the user to log in to the service. |
| Frequency: | As many times as the user tries to embark on a journey. |
| Non-Functional Requirements: | Provide the user with instructions to operate the system for the first time. |

Table 3.3 Touch On/Off

3.8.4 Vehicle Tracking

| | |
|---|--|
| Scope: | On-Time Bus Mobile App (Client) and Bus Operator Terminal |
| Primary Actor: | Passenger, Admin |
| Preconditions: | The user device must be connected to an active internet connection and an enabled GPS. The user must be present at the bus stop, or in the vicinity of the bus stop. |
| Success Guarantee / Post Condition: | The user gets the necessary information about the bus arrival/departure. |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| The user pans and zooms on the provided map UI to search for the buses, and the user taps on a displayed stop Icon. | The user will be provided with the necessary information about the arrival and departure of the bus, at the required bus stop. The user will also be able to spot the bus on the displayed UI. |
| Alternate Scenario: | |
| No currently running services. | Prompt the user to select another bus stop. |
| Frequency: | High. |
| Non-Functional Requirements: | Provide the user with instructions to operate the system for the first time. |

Table 3.4 Vehicle Tracking

3.8.5 View Wallet

| | |
|--|---|
| Scope: | On-Time Bus Mobile App (Client) and Bus Operator Terminal |
| Primary Actor: | Passenger, Admin (On behalf of the user) |
| Preconditions: | The user device must be connected to an active internet connection. The user must be logged in. |
| Success Guarantee / Post Condition: | The user gets the necessary information about his wallet. |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| User taps on the provided icon for the wallet. | The system navigates to the user’s Wallet Screen. |
| Alternate Scenario: | |
| Not Logged in | Prompt the user to log in. |
| Frequency: | High. |
| Non-Functional Requirements: | Provide the user with instructions to operate the system for the first time. |

Table 3.5 View Wallet

3.8.6 View Schedule

| | |
|--|--|
| Scope: | On-Time Bus Mobile App (Client) |
| Primary Actor: | Passenger |
| Preconditions: | The user device must be connected to an active internet connection. |
| Success Guarantee / Post Condition: | The user gets the necessary information about the required bus schedule. |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| The user searches for a bus stop. The user selects a bus stop from the provided Map UI. | The system navigates to the user’s required bus stop information. |
| Alternate Scenario: | |
| The user enters an invalid keyword for search. | Prompt the user to search for a valid bus stop name. |
| Frequency: | High. |
| Non-Functional Requirements: | Provide the user with instructions to operate the system for the first time. |

Table 3.6 View Schedule

3.8.7 Manage Route

| | |
|--|---|
| Scope: | Bus Operator Terminal |
| Primary Actor: | Admin |
| Preconditions: | Admin must be logged in. |
| Success Guarantee / Post Condition: | Admin can view, update, delete or add a new Route to the system. |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| 1. Admin selects to add a new route. 2. Admin selects to update an existing route. 3. Admin selects to delete an existing route. | 1. The system navigates the admin to the new route entry form. 2. The system navigates to the list of existing routes and then prompts the admin to edit the required changes to the existing route. 3. The system navigates to the list of existing routes and then prompts the admin to delete the unwanted existing route. |
| Alternate Scenario: | |
| Not Logged in | Prompt the admin to log in. |
| Frequency: | High. |
| Non-Functional Requirements: | Provide the user with instructions to operate the system for the first time. |

Table 3.7 Manage Routes

3.8.8 Manage Buses/Fleet

| | |
|---|--|
| Scope: | Bus Operator Terminal |
| Primary Actor: | Admin |
| Preconditions: | Admin must be logged in. |
| Success Guarantee / Post Condition: | Admin can view, update, delete or add a new Bus/Vehicle to the system. |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| <div>1. Admin selects to add a new bus.</div> <div>4. Admin selects to update an existing bus.</div> <div>5. Admin selects to delete an existing bus.</div> | <div>1. The system navigates the admin to the new bus entry form.</div> <div>2. The system navigates to the list of existing buses and then prompts the admin to edit the required changes to the existing bus.</div> <div>3. The system navigates to the list of existing buses and then prompts the admin to delete the unwanted existing bus.</div> |
| Alternate Scenario: | |
| Not Logged in | Prompt the admin to log in. |
| Frequency: | High. |
| Non-Functional Requirements: | Provide the user with instructions to operate the system for the first time. |

Table 3.8 Manage Buses/Fleet

3.8.9 Manage Bus Stop

| | |
|--|---|
| Scope: | Bus Operator Terminal |
| Primary Actor: | Admin |
| Preconditions: | Admin must be logged in. |
| Success Guarantee / Post Condition: | Admin can view, update, delete or add a new Bus Stop to the system. |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| <div>1. Admin selects to add a new bus stop.</div> <div>6. Admin selects to update an existing bus stop.</div> <div>7. Admin selects to delete an existing bus stop.</div> | <div>1. The system navigates the admin to the new bus entry form.</div> <div>2. The system navigates to the list of the existing bus stop and then prompts the admin to edit the required changes to the existing bus stop.</div> <div>3. The system navigates to the list of existing bus stops and then prompts the admin to delete the unwanted existing bus stop.</div> |
| Alternate Scenario: | |
| Not Logged in | Prompt the admin to log in. |
| Frequency: | High. |

| | |
|-------------------------------------|--|
| Non-Functional Requirements: | Provide the user with instructions to operate the system for the first time. |
|-------------------------------------|--|

Table 3.9 Manage Bus Stop

3.8.10 Manage Bus Schedule

| | |
|---|--|
| Scope: | Bus Operator Terminal |
| Primary Actor: | Admin |
| Preconditions: | Admin must be logged in. |
| Success Guarantee / Post Condition: | Admin can view, update, delete or add a new schedule to the system. |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| 1. Admin selects to add a new schedule. 8. Admin selects to update an existing schedule. 9. Admin selects to delete an existing schedule. | 1. The system navigates the admin to the new schedule entry form. 2. The system navigates to the list of existing schedules and then prompts the admin to edit the required changes to the existing schedule. 3. The system navigates to the list of existing schedules and then prompts the admin to delete the unwanted existing schedule. |
| Alternate Scenario: | |
| Not Logged in | Prompt the admin to log in. |
| Frequency: | High. |
| Non-Functional Requirements: | Provide users with instructions to operate the system for the first time. |

Table 3.10 Bus Schedule

3.8.11 Manage Users

| | |
|--|---|
| Scope: | Bus Operator Terminal |
| Primary Actor: | Admin |
| Preconditions: | Admin must be logged in. |
| Success Guarantee / Post Condition: | Admin can view, update, delete or add a new user/client to the system. |
| Main Success Scenario: | |
| User Action: | System Responsibility: |
| 1. Admin selects to add a new user/client. 10. Admin selects to update an existing user/client. 11. Admin selects to delete an existing user/client. | 1. The system navigates the admin to the new user/client entry form. 2. The system navigates to the list of existing schedules and then prompts the admin to edit the required changes to the user/client. 3. The system navigates to the list of existing schedules and then prompts the admin to delete |

| | |
|-------------------------------------|--|
| | the unwanted existing user/client. |
| Alternate Scenario: | |
| Not Logged in | Prompt the admin to log in. |
| Frequency: | High. |
| Non-Functional Requirements: | Provide the user with instructions to operate the system for the first time. |

Table 3.11 Manage User

3.9 System Sequence Diagrams:

A system sequence diagram (SSD) is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate their order, and possible inter-system events. System sequence diagrams are visual summaries of the individual use cases. System sequence diagrams illustrate how certain tasks are done between users and the system. These tasks may include repetitive, simple, or complex tasks. The purpose is to illustrate the use case in a visual format.

3.9.1 Login:

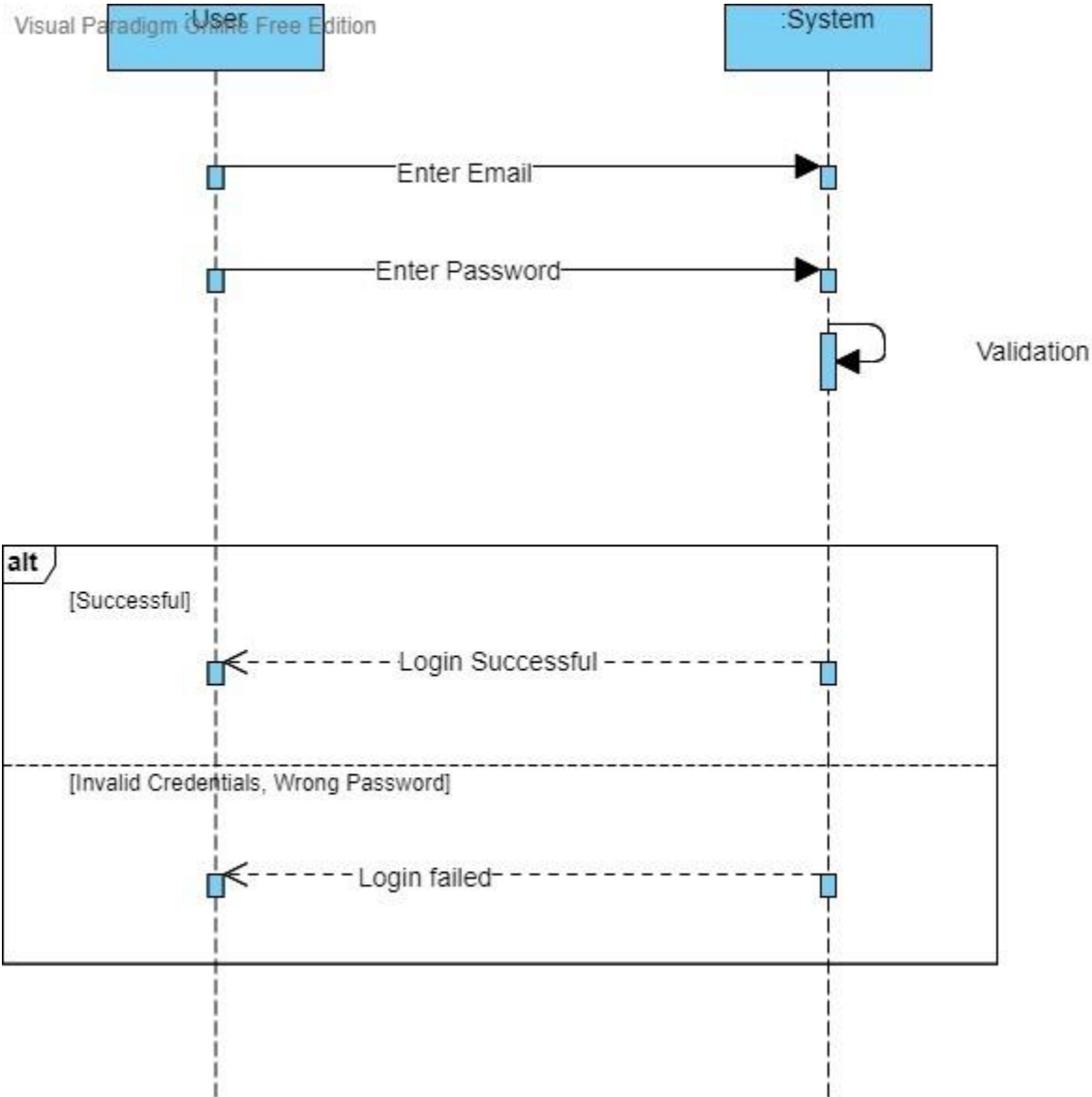


Figure 3.2 Login

3.9.2 Registration:

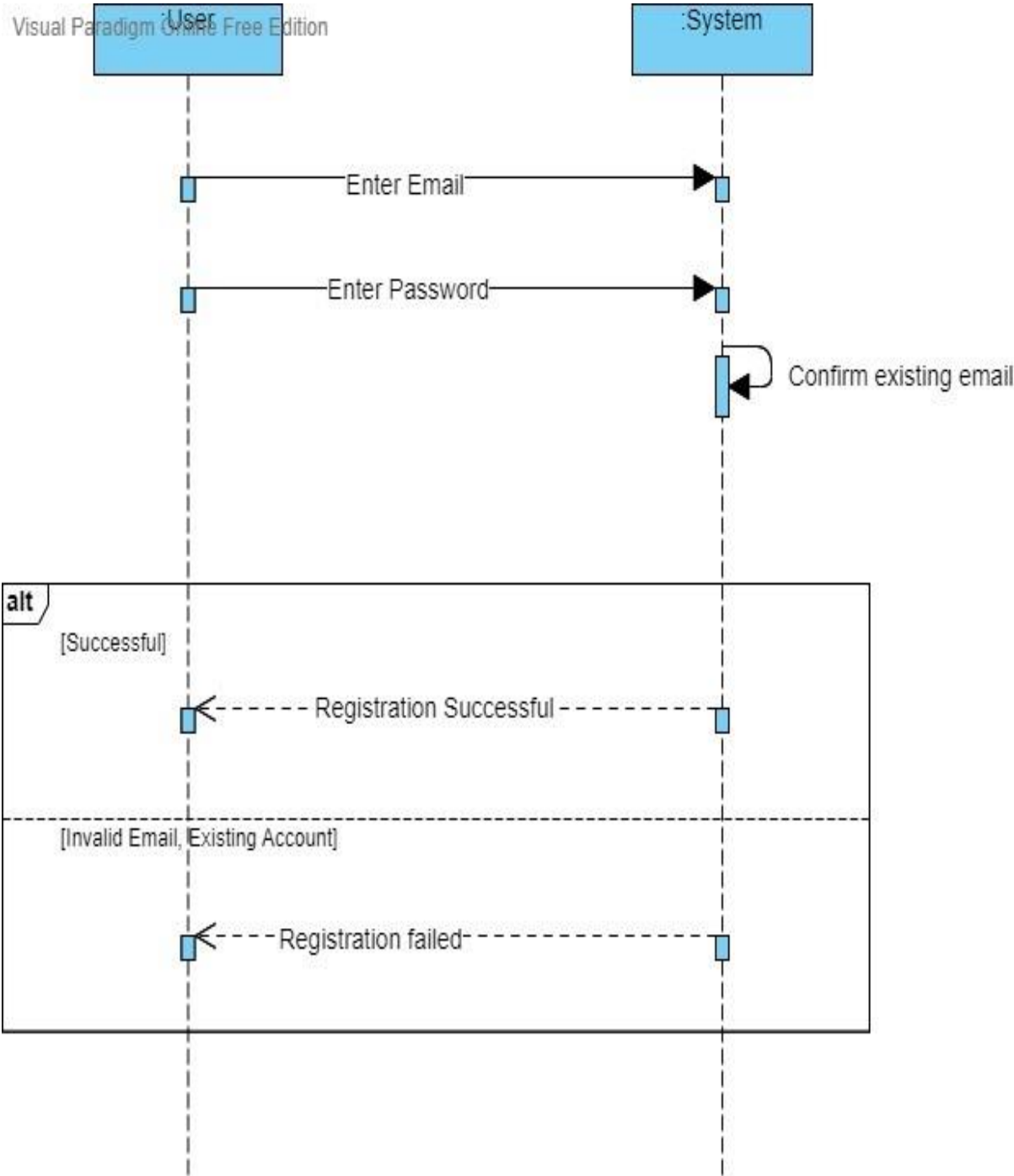


Figure 3.3 Signup

3.9.3 Touch On/Off:

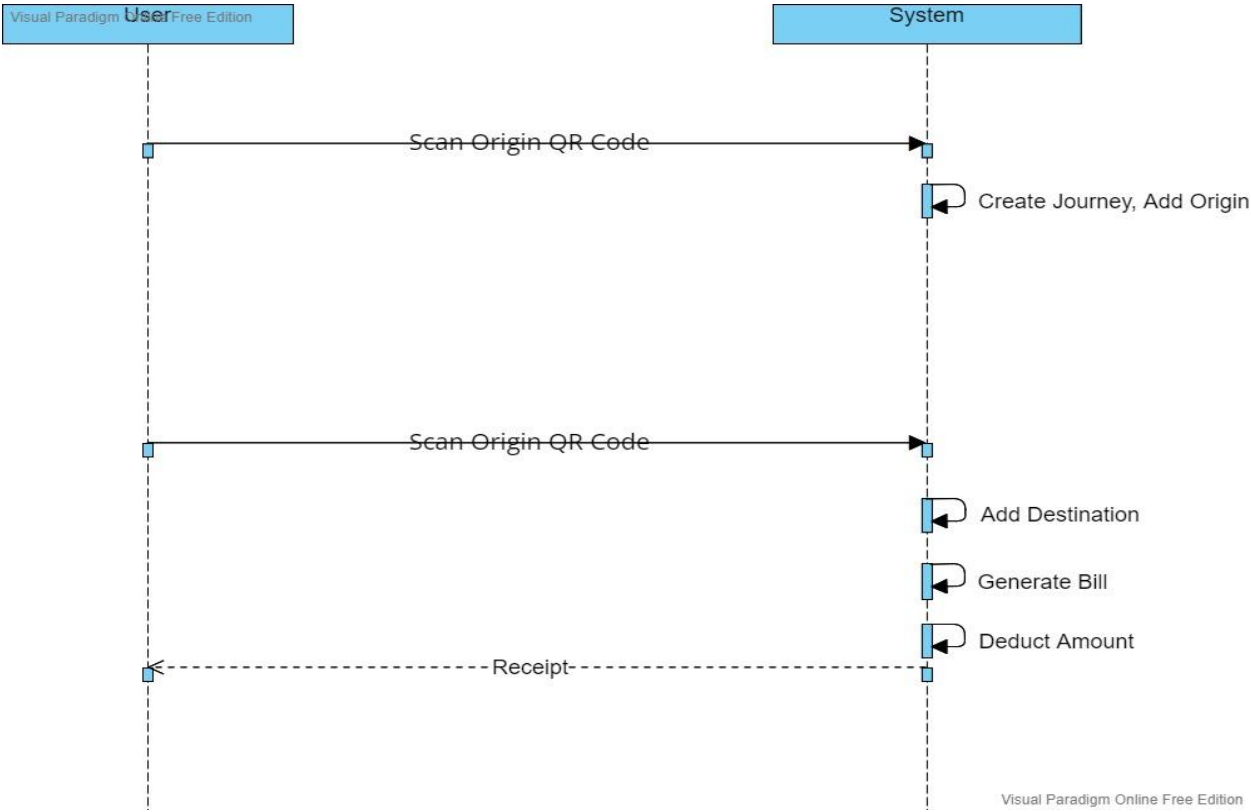


Figure 3.4 Touch On/Off

3.9.4 Track Vehicle:

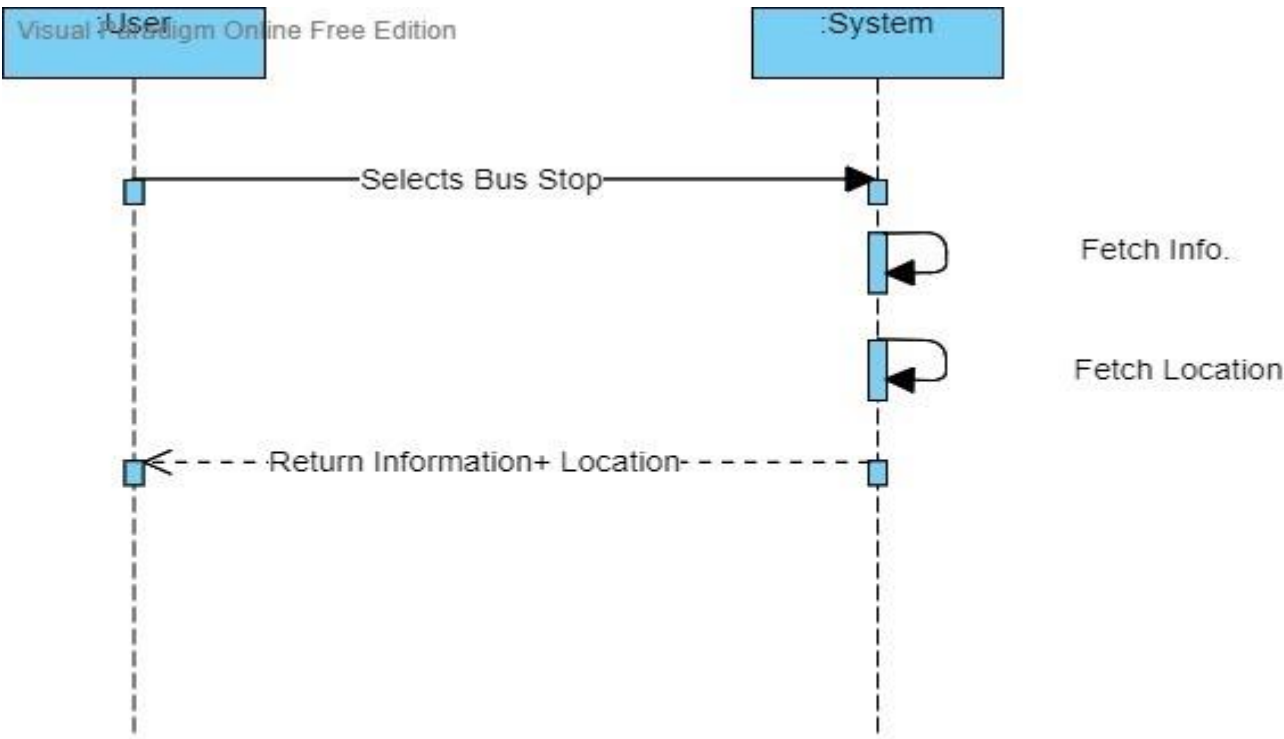


Figure 3.5 Track Vehicle

3.9.5 View Schedule:

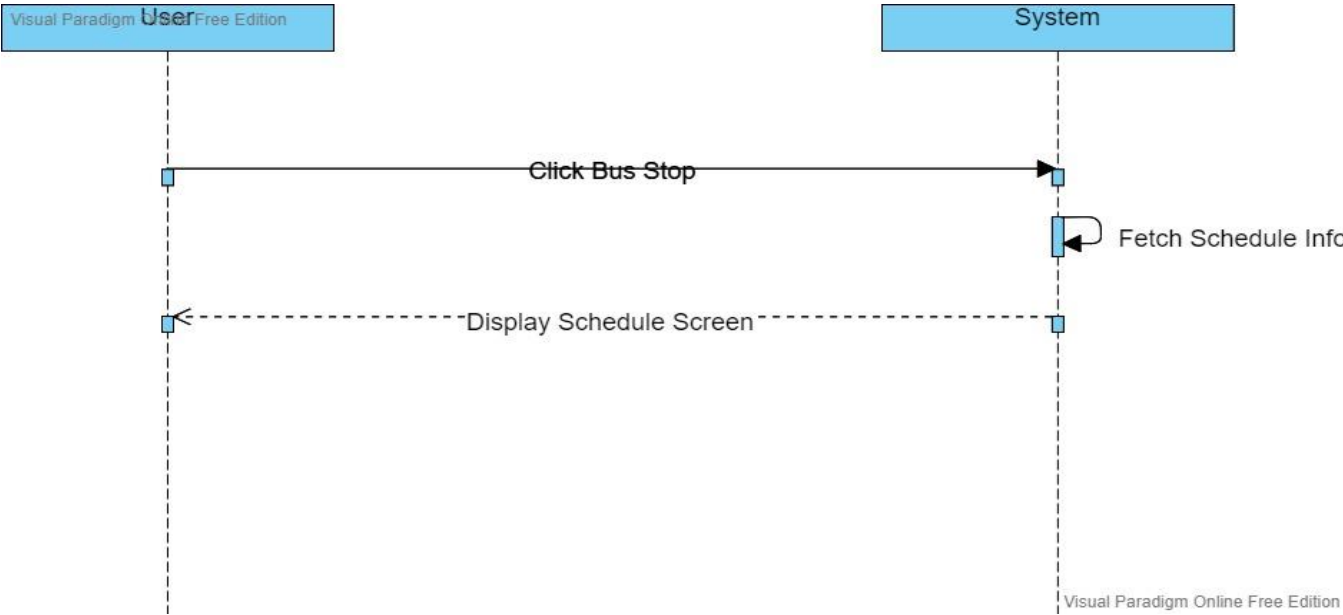


Figure 3.6 View Schedule

3.9.6 View Wallet:

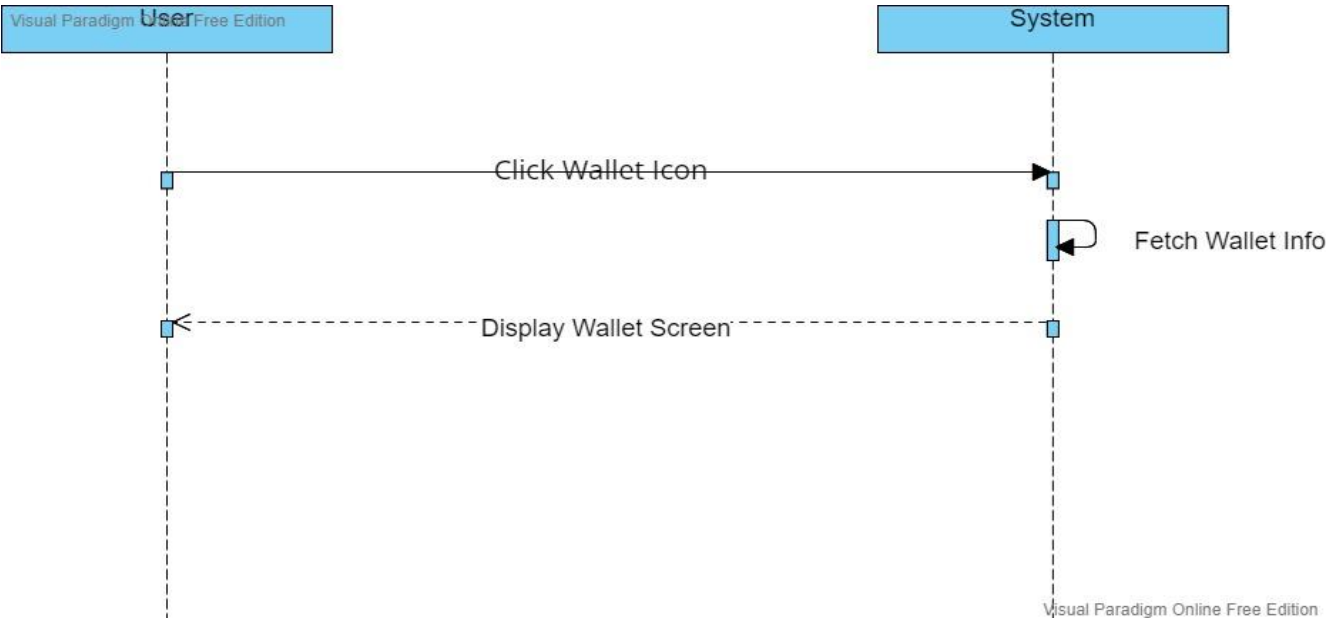


Figure 3.7 View Wallet

3.9.7 Manage Bus:

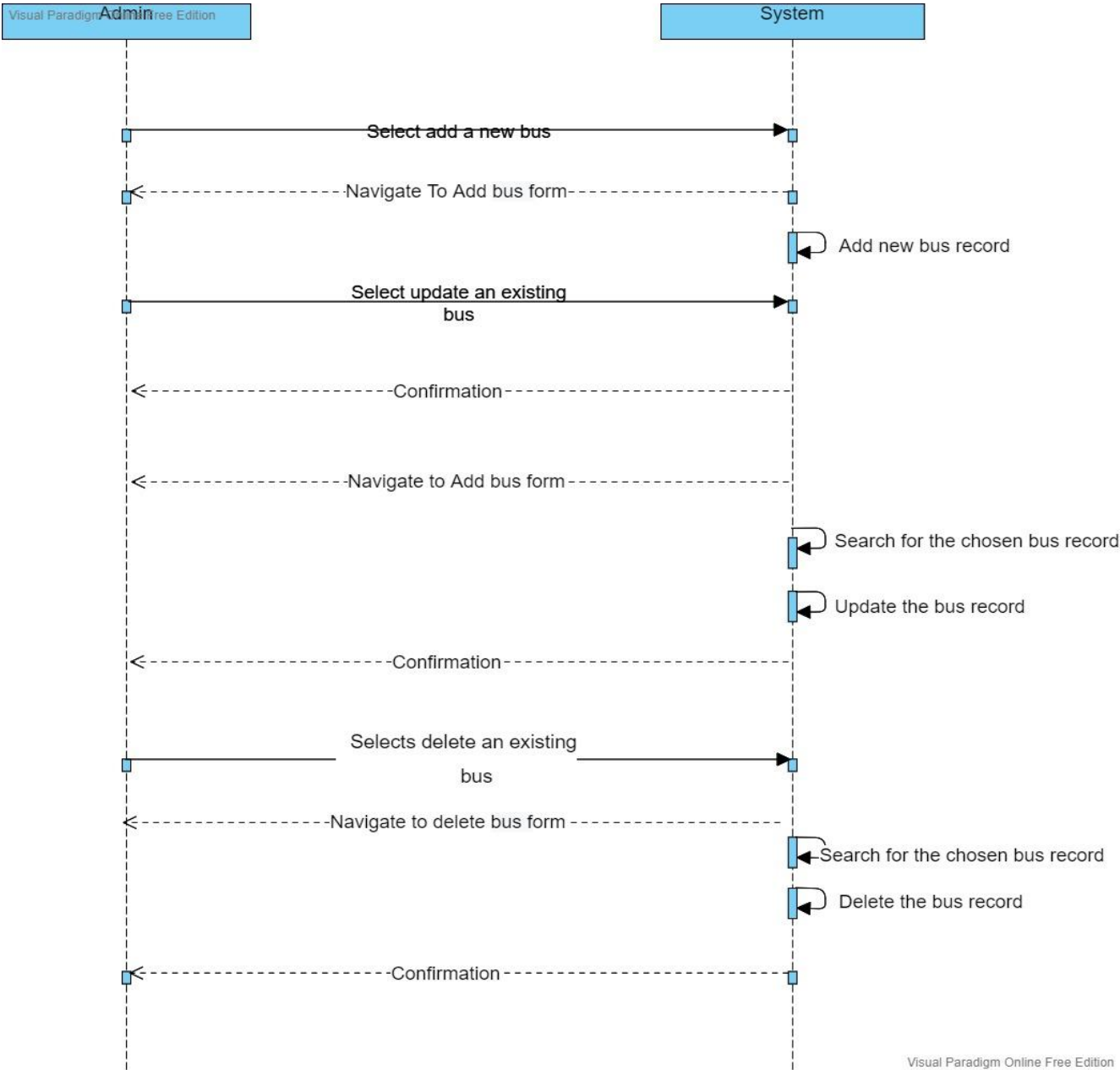


Figure 3.8 Manage Bus

3.9.8 Manage Route:

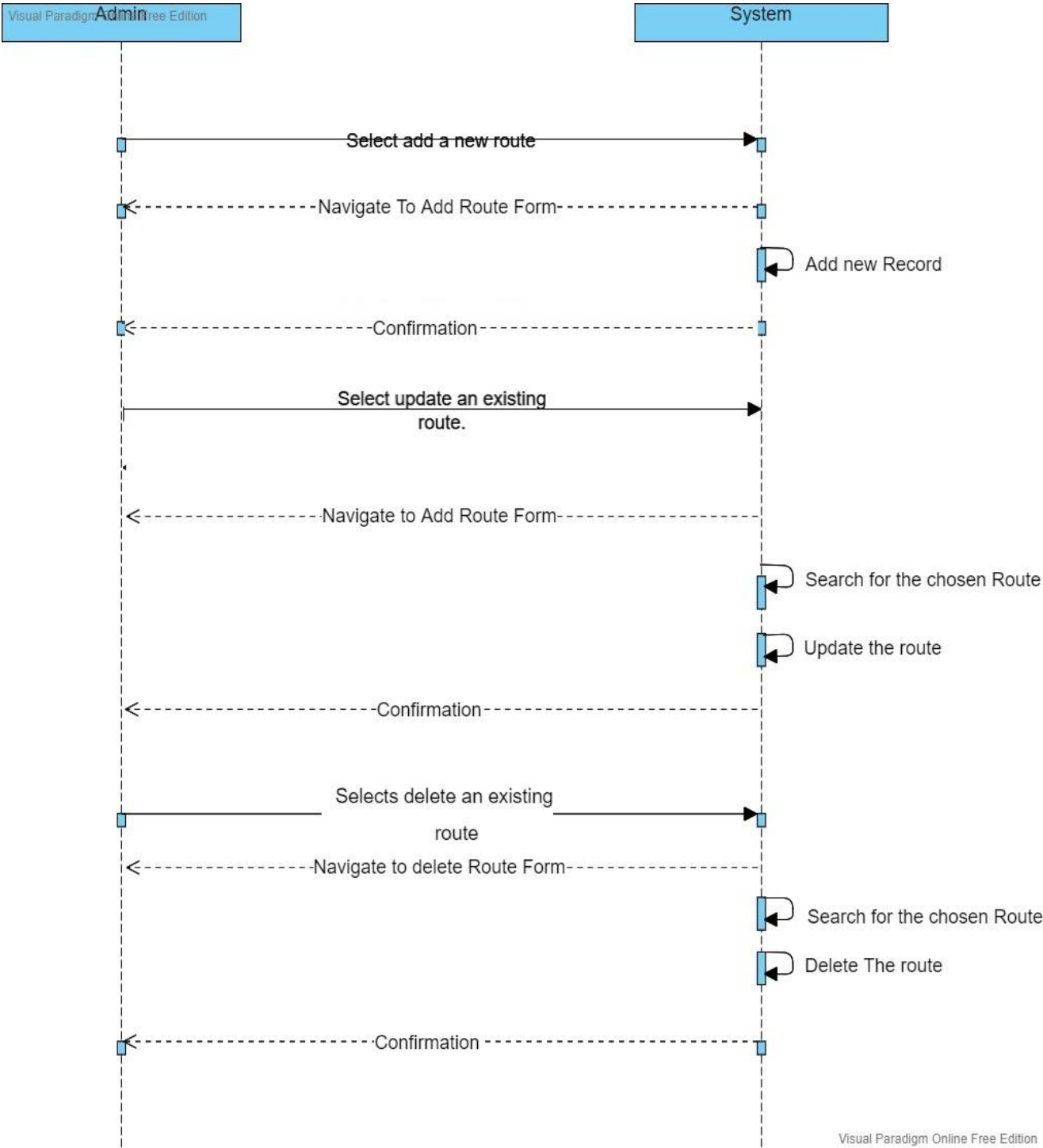


Figure 3.9 Manage Route

3.9.9 Manage Schedule:

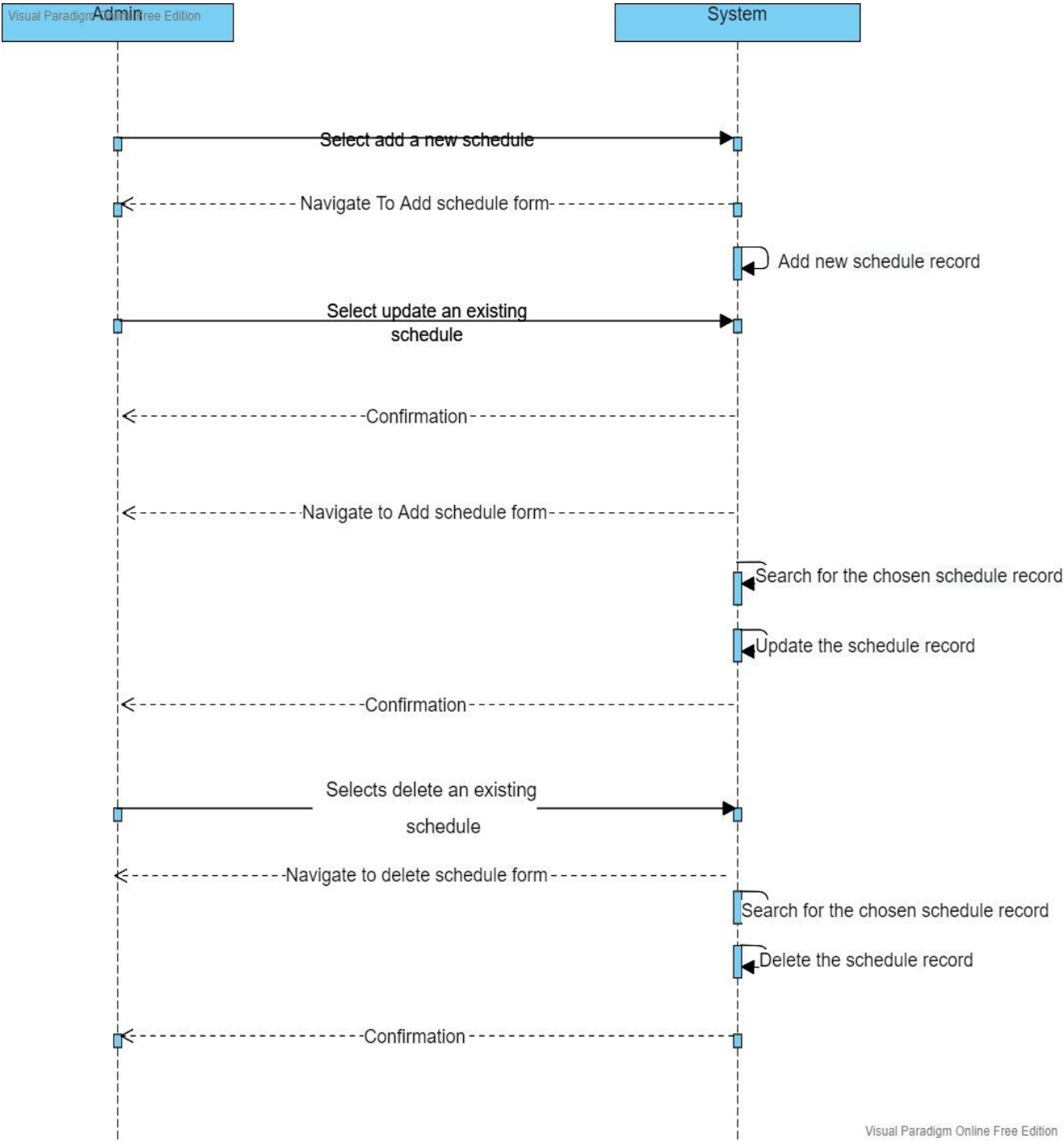


Figure 3.10 Manage Schedule

3.9.10 Manage Bus Stop:

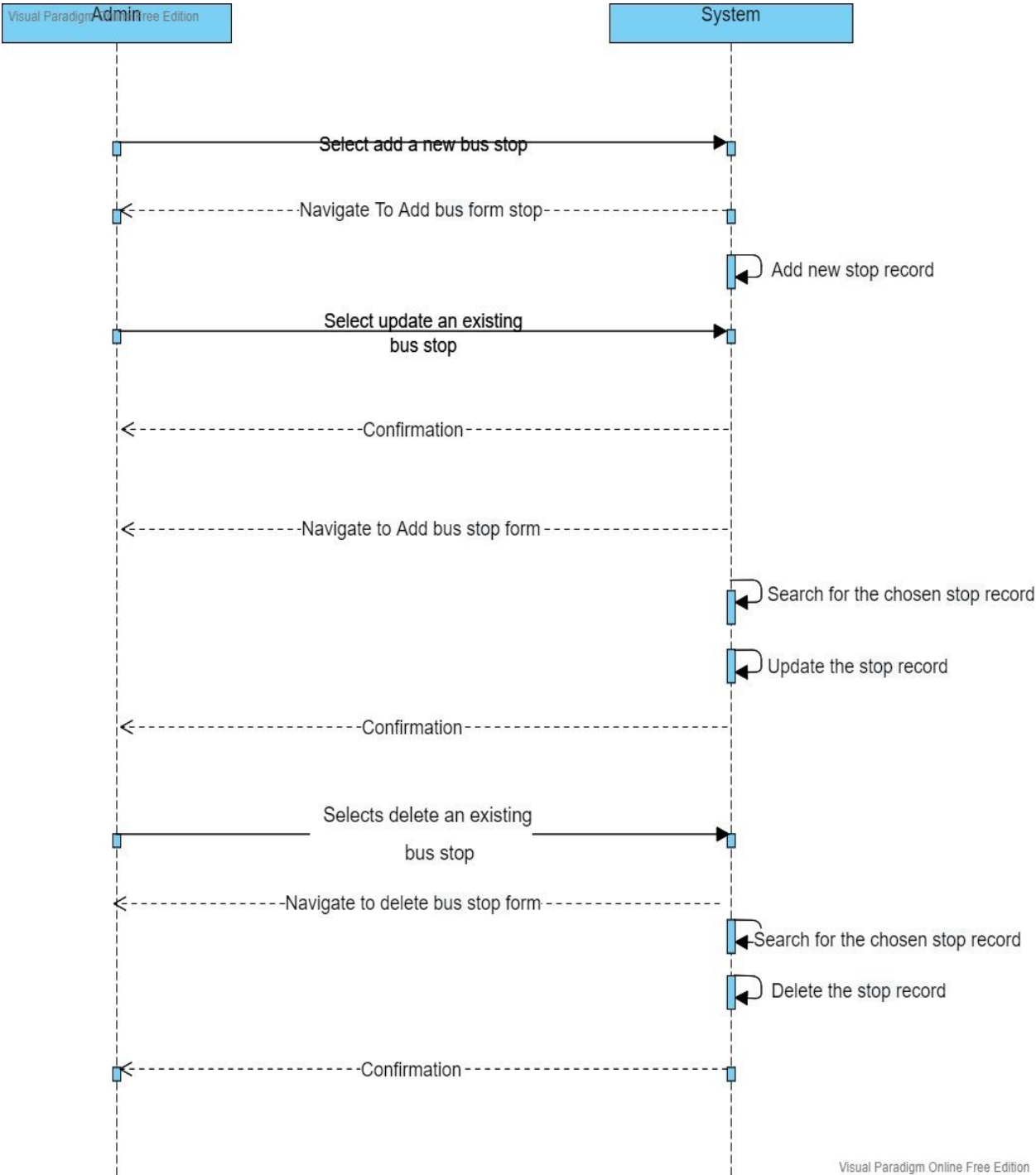


Figure 3.11 Manage Bus Stop

3.9.11 Manage User Profile:

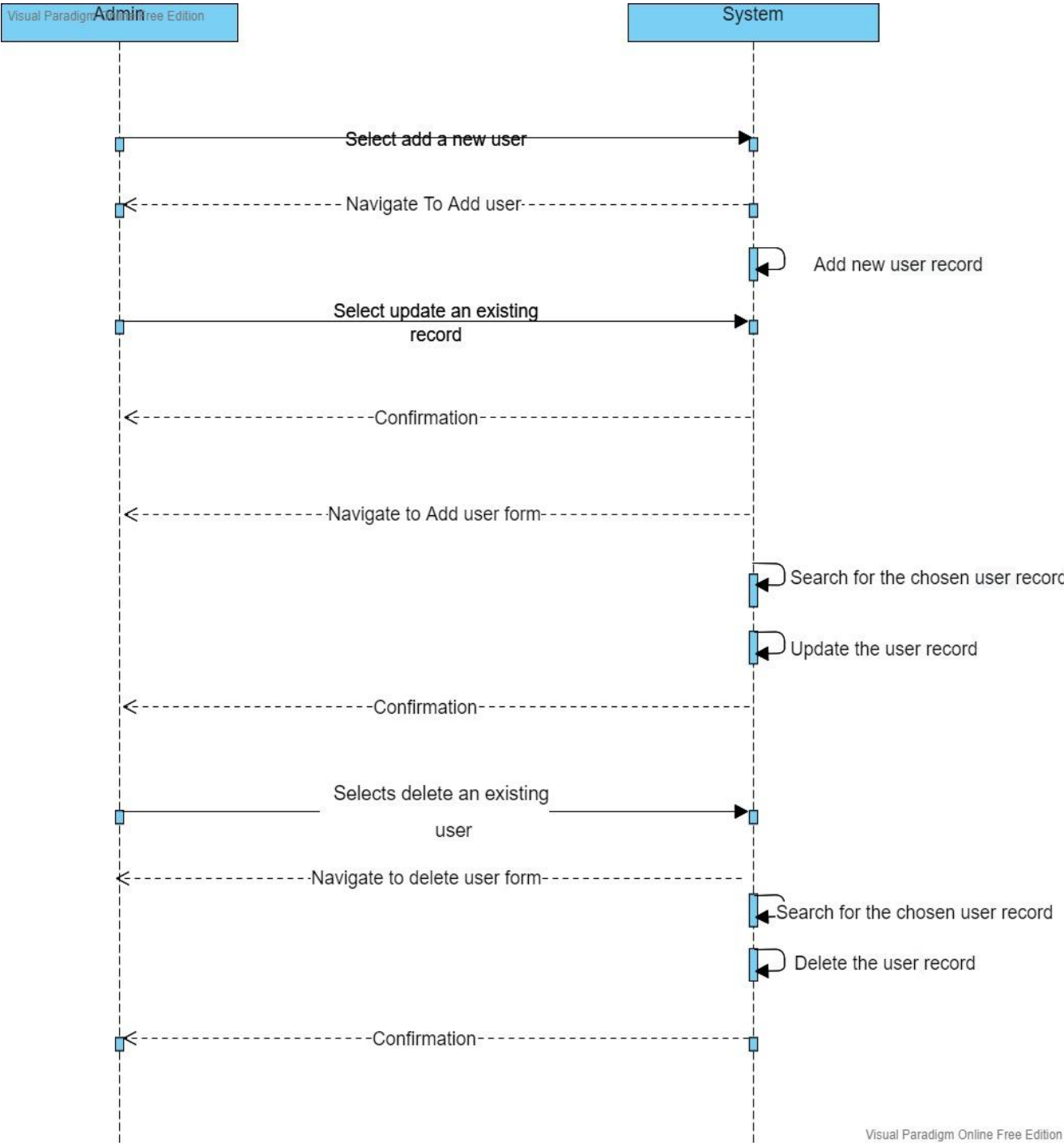


Figure 3.12 Manage User Profile

3.10 Domain Model:

Domain Model shows important concepts in the domain of interest. Depicts objects, their attributes, and associations between objects.

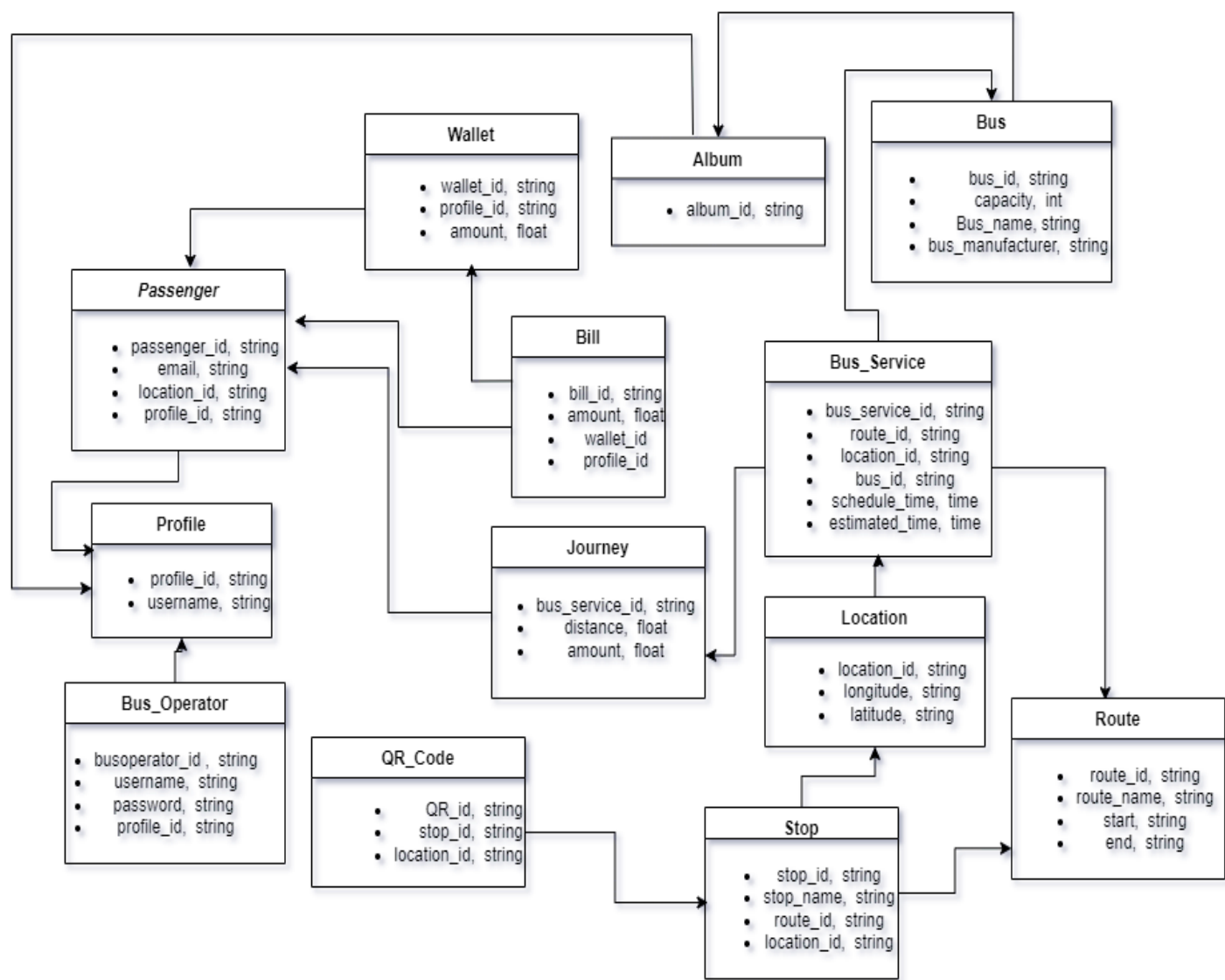


Figure 3.13 Domain Model

CHAPTER 4

SYSTEM DESIGN

4. System Design

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. System design could be seen as the application of systems theory to product development.

4.1. Sequence Diagram:

4.1.1 User:

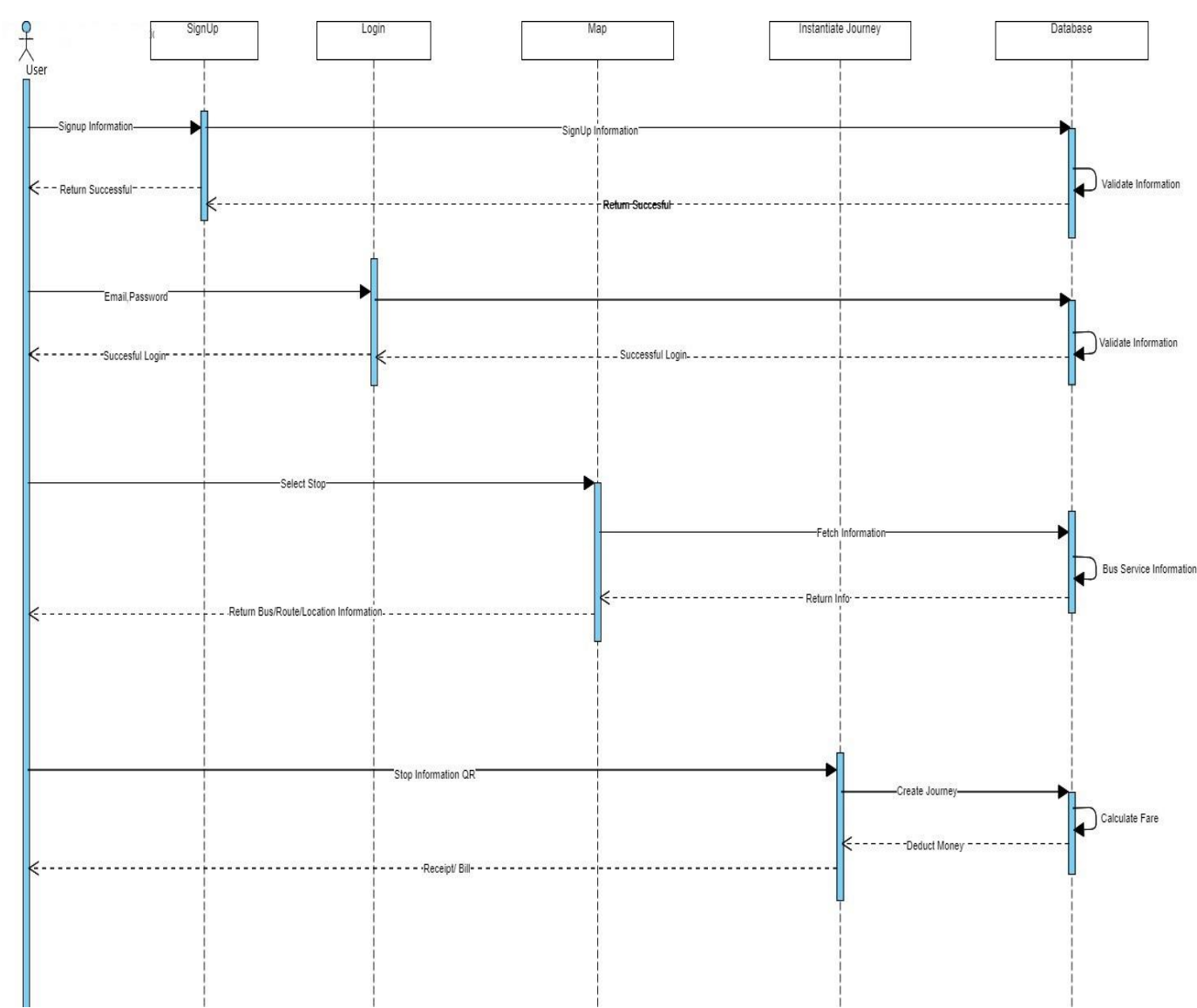


Figure 4.1 Sequence Diagram of User

4.1.2 Bus Operator/ Company/ Admin:

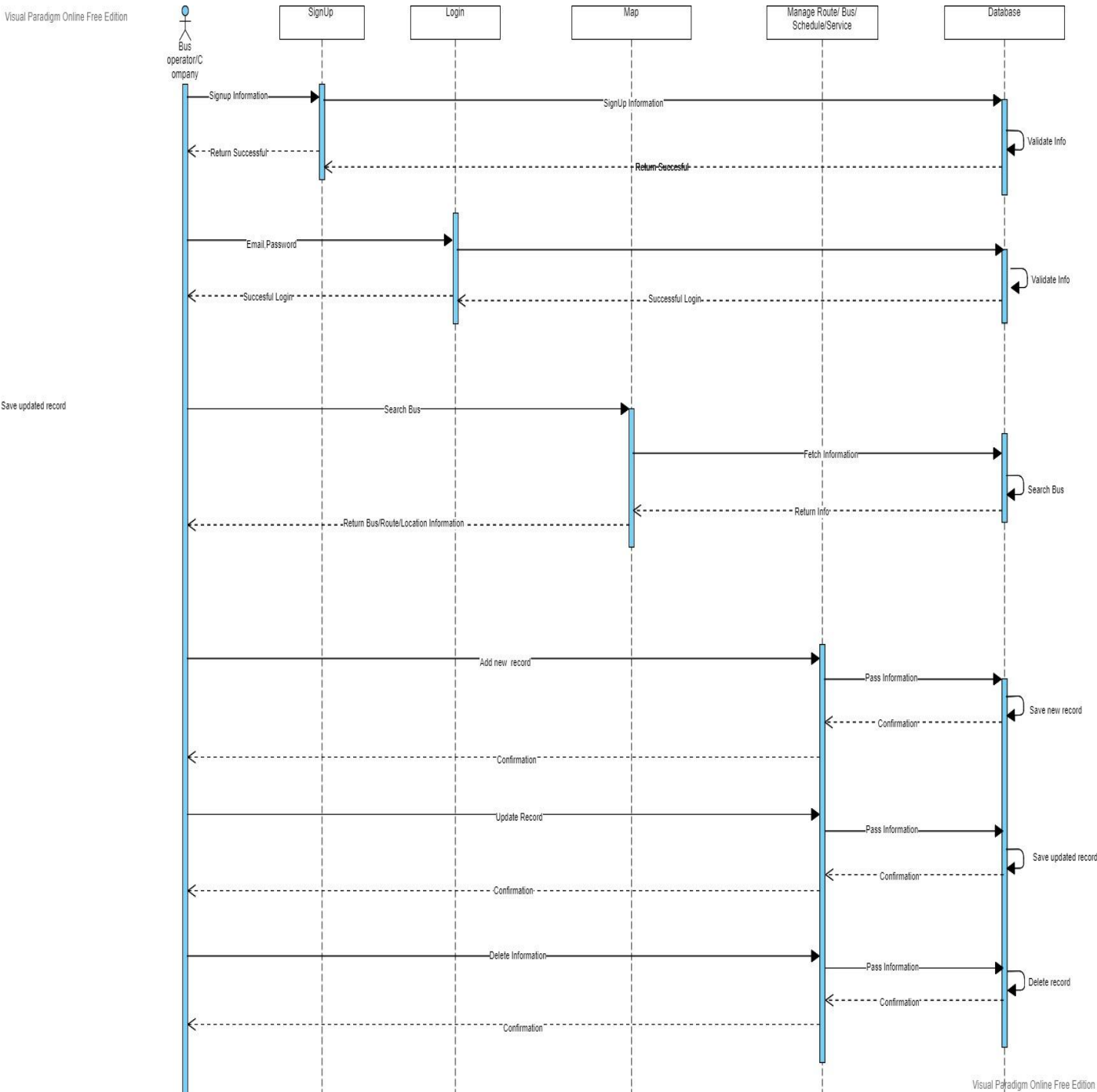


Figure 4.2 Sequence Diagram Bus/Company/Admin

4.2 Class Diagram:

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (or methods), and the relationships among objects or classes.

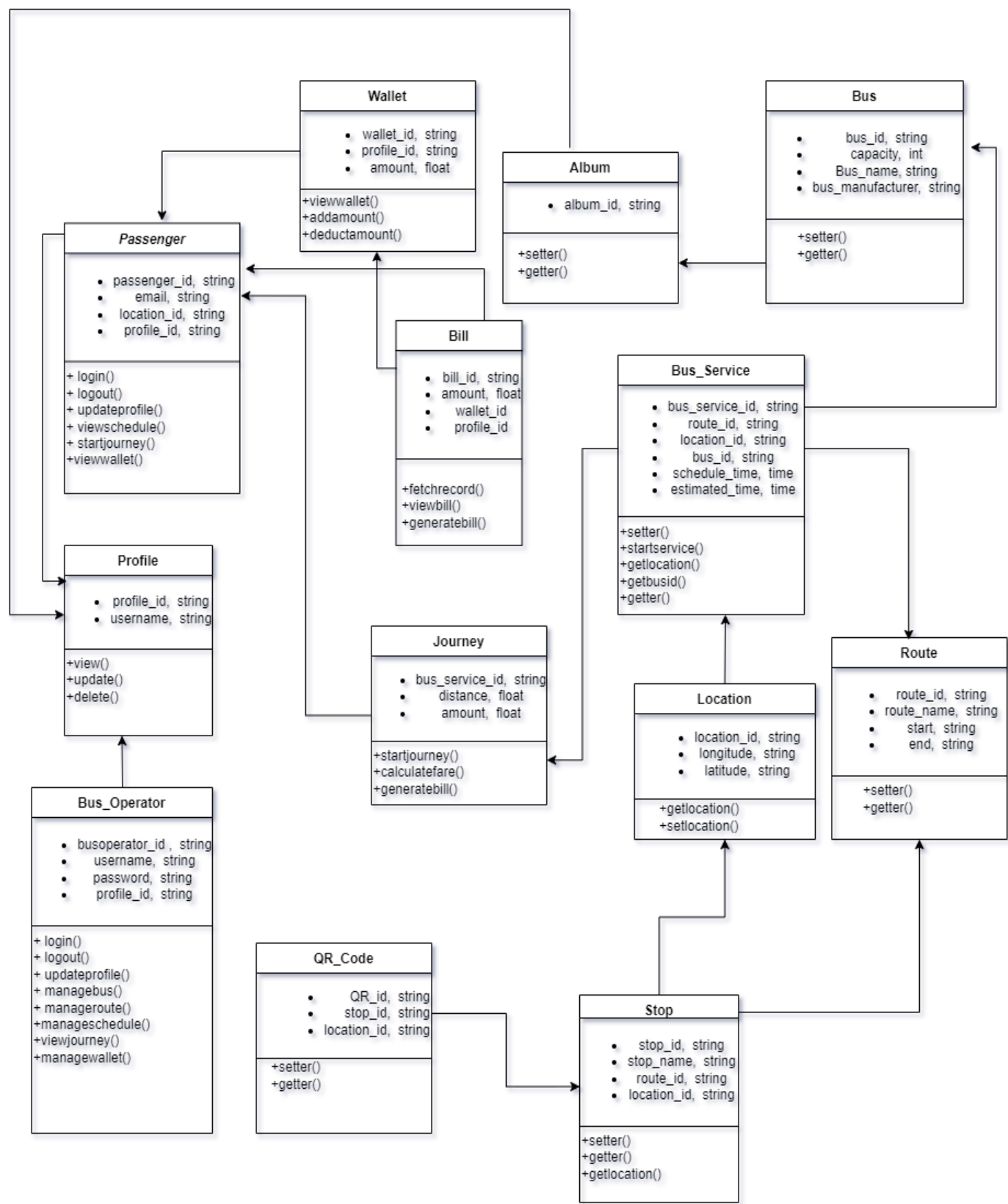


Figure 4.3 Class Diagram

4.3 Activity Diagram:

Activity diagrams are Graphical representations of workflows of stepwise activities and actions with the support for choice, iteration, and concurrency. In the Unified Modeling Language, activity diagrams

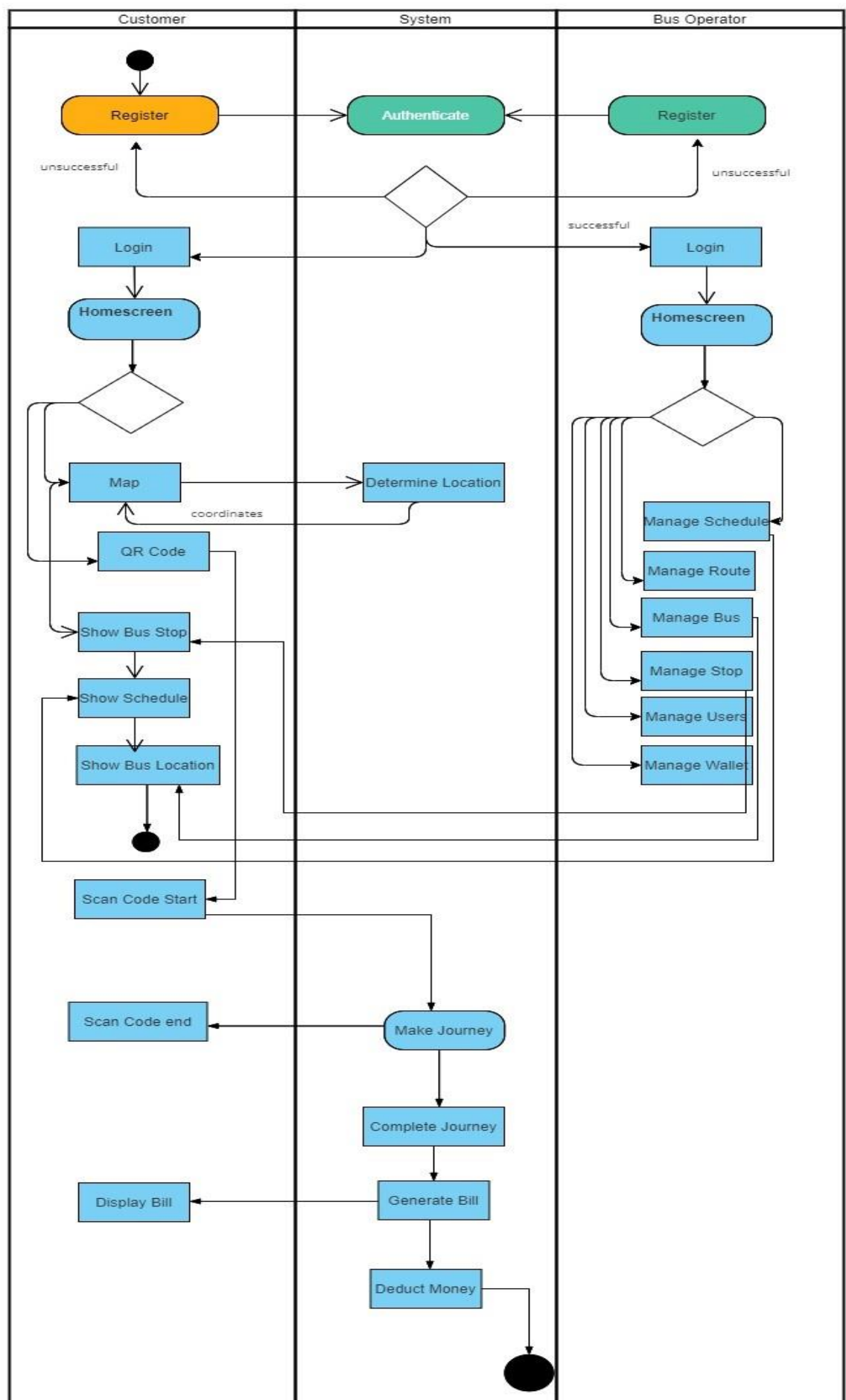


Figure 4.4 Activity Diagram

4.4 Entity Relationship Diagram:

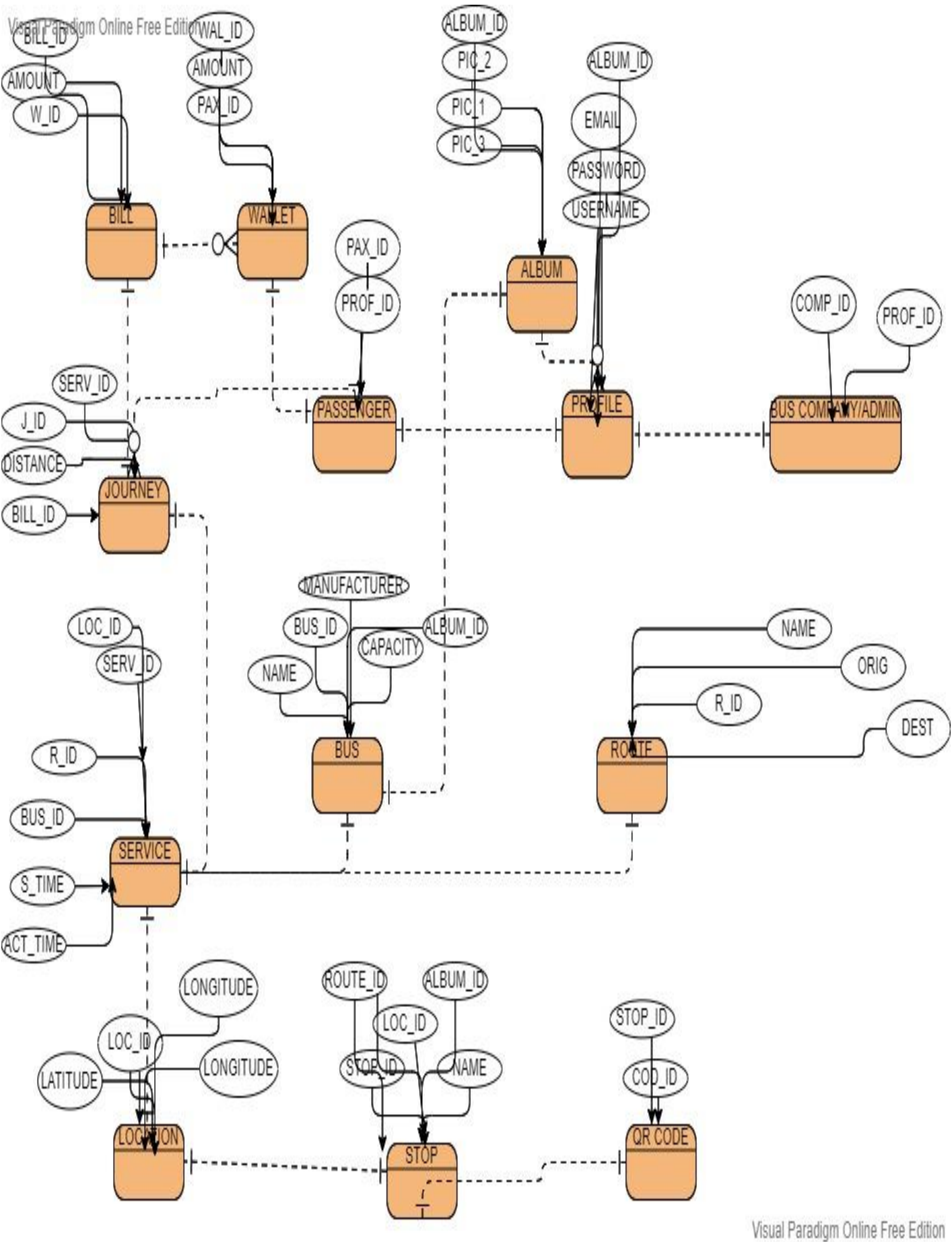


Figure 4.5 Entity Relationship Diagram

CHAPTER 5

IMPLEMENTATION

5.1 System Implementation

Implementation is just another word for coding/software programming.

5.2 Functional Features

- User-friendly interface.
- Responsiveness.
- Reliability.

5.3 Tools and Technologies

5.3.1 React

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front-end library responsible only for the view layer of the application. It is maintained by Facebook. React uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and flexible. It designs simple views for each state in your application and reacts will efficiently update and render just the right component when your data changes. The declarative view makes your code more predictable and easier to debug. A React application is made of multiple components, each responsible for rendering a small, reusable piece of HTML. Components can be nested within other components to allow complex applications to be built out of simple building blocks. In our project, we use react tool to build the admin panel dashboard.

5.3.2 React-Native

React-Native is used to create native iOS and Android applications using JavaScript. It is a framework of JavaScript. Although it is built on the same principles as React, it renders user interfaces using native components rather than web components (UI). In our project, we have built our mobile app using react-native. If you are supporting more than one platform, React Native gives a solid advantage in terms of reusing core code from your app. You may still have to provide some platform-specific code, but it will certainly be less to write and maintain.

5.3.3 Database

The database used in our app is Firebase because it is easy for the developers to integrate it with a front end in any technology and to build, manage, and extend their apps. This helps developers build their apps faster and more securely. You can use the features more efficiently because you don't have to program on Firebase. We provide services for Android, iOS, Web, and Unity. Provides cloud storage. Store the data in the database using NoSQL.

5.3.4 Arduino

An open-source electronics platform called Arduino is built on simple hardware and software. A motor can be started, an LED can be turned on, and something may be published online by using an Arduino board to receive inputs like light on a sensor, a finger on a button, or a tweet. Sending a set of instructions to the board's microcontroller will instruct your board on what to do. You achieve this by using the Arduino Software (IDE), which is based on Processing, and the Wiring-based Arduino Programming Language. In our project, we use Arduino for tracking vehicle directions and showing them to a user on the map screen of the app.

5.3.5 Microsoft Visual Studio Code

We have used Microsoft Visual Studio Code to write our project. Because Visual studio code is a free source code chief made by Microsoft for Windows, Linux, and MAC OS. It can be used in various programming languages such as Java, JavaScript, Go, and Node.js, Python, C++, Fortran. It is based on the Electron framework used to develop Node. A js web application running on the Blink layout engine.

CHAPTER 6

TESTING

6.1 Testing

System testing includes testing the performance, output, behavior, and operation of a system in such an environment that it is very close to the possible end user's environment. System testing tests the system in the condition in which it is going to be used by the end user, thus developers and testing engineers monitor the functionality and behavior. Application is tested in all simpler and worst scenarios, and it remains successful even in the worst cases.

6.1.1 Component of Testing

Functions and procedures are tested separated in this stage. If there is a bug, it can be removed easily. Since modules are combinations of functions and procedures, therefore, after testing of functions and procedures, testing of a module should be carried out.

6.1.2 Integration of Testing

A combination of modules is called a sub-system and a complete system is built out of some sub-system. Integration testing is the testing of a sub-system and then the testing of the overall system.

6.1.3 User Testing

In this stage, a user is involved in the testing process. The user provides some inputs and outputs. Known inputs are given to the system. If a system provides the same outputs that are required by a user, then the user will accept the system. This is also called acceptance testing.

6.1.4 Test Case Scenario

Test cases are generated to test the functionality of each use case and conform it to the requirements identifies earlier. Test cases are executed according to a defined sequence of tasks and the expected results are defined before execution. The actual results of execution are compared to the expected results and hence a test case is said as pass/fail or not executed. The test cases are given below. We have tested our app in every case best case, worst case. And we found no errors in our application. It is running smoothly, and all features are working perfectly fine. We have tested our apps through many scenarios and found no errors.

| Test Case ID | Test Scenario | Test Steps | Test Data | Results | Fail/Pass |
|--------------|------------------------------------|---|--|--|-----------|
| TU01 | Compatibility | Run the Application on different browsers | N/A | The frontend and backend server is working | Pass |
| TU02 | Sign up (User) | 1) Click on the Sign-up button in the top right corner 2) Fill in the Sign-up form 3) Click on the Sign-up button | First name= Usman Last name= Karamat Email= muhammed.usman77@gmail.com Password=Soccer#95 | The user is signed up successfully, please verify your email | Pass |
| TU03 | Sign in | 1) Open the mobile app 2) Fill in the sign-in data 3) Click on sign in button | Email= muhammed.usman77@gmail.com Password=Soccer#95 | Complete your Profile template is displayed | Pass |
| TU04 | Show Map | Toggle Map Screen from Bottom tab navigator | Current location | Bus stops near the current location | Pass |
| TU05 | Scan Card on Provided RFID machine | Mobile App Navigates automatically to the Journey Screen. | Mobile App Navigates automatically to the Journey Screen. | Journey details are shown. | Pass |
| TU06 | Scan Card on Provided RFID machine | Journey Ends automatically, the app navigates to the journey summary screen. | The journey summary screen shown | Journey details are shown. | Pass |
| TU07 | Show Bus Stops | Search stop name using search | Stopname: Iqra College | Related results shown | Pass |
| TU08 | Show Stop Schedule | Touch a particular bus stop | Touch a particular bus stop | Stop schedule shown | Pass |
| TU09 | Show bus location. | Scroll on the map to find the Bus | Touch on bus | Bus journey details shown | Pass |

| | | | | | |
|-------------|-----------------|---|---|------------------------|------|
| TU10 | Sign in (Admin) | Open Webapp and log in using credentials. | Username= admin Password = Admin@123 | Signed in Successfully | Pass |
| TU11 | Manage Schedule | Navigate to the schedule screen using the side navigation panel | CRUD appropriate details | Success | Pass |
| TU12 | Manage Route | Navigate to the Route screen using the side navigation panel | CRUD appropriate details | Success | Pass |
| TU13 | Manage Stop | Navigate to the Route screen using the side navigation panel | CRUD appropriate details | Success | Pass |
| TU14 | Manage Users | Navigate to the Route screen using the side navigation panel | CRUD appropriate details | Success | Pass |
| TU15 | Manage Wallet | Navigate to the Wallet screen using the side navigation panel | CRUD appropriate details | Success | Pass |

Table 6.1 Test Cases

6.1.5 Goal Achieved

The goal achieved for which this application is developed is achieved. The user after logging in enters the application and gets a variety of functionality, using which he can access multiple transportation-related issues. Users watch the buses through a map and schedule their trips accordingly from one point to another. Users can search for stops where buses come and after completion of their ride users can pay for their ride through an RFID card.

CHAPTER 7
CONCLUSION

7.1 Conclusion and future work

In this chapter, we will discuss the conclusion and future work of the project. How much work is completed? what are the limitations? what the scope is? and what should be done in the future and how many features will be added in the future? This chapter is the result of the previous five chapters.

7.1.1 Conclusion

On-Time Bus is a smart bus scheduling system that allows users to view, track and route their journey on given bus service. This system would also enable the business owners/managers to manage their running buses/ routes/ drivers. They could also track the buses running on a particular route. The system utilizes the modern multi-Platform app development tool, React-Native. It also utilizes APIs from Google Maps. The On-Time-Bus system acts as an intermediary between the end-user (passenger) and the service provider (Business owner/Manager) and fills the gaps in communication using the latest technology, which is not necessarily human-to-human communication. This system also takes 3 out of the “guesswork” and ambiguity involved in the automobile industry, for example guessing how long it would take for a bus to arrive at a particular stop and pay for its ride through RFID and for the admin panel, an admin always sees users’ data, add routes, add buses using the web.

7.2 Limitations

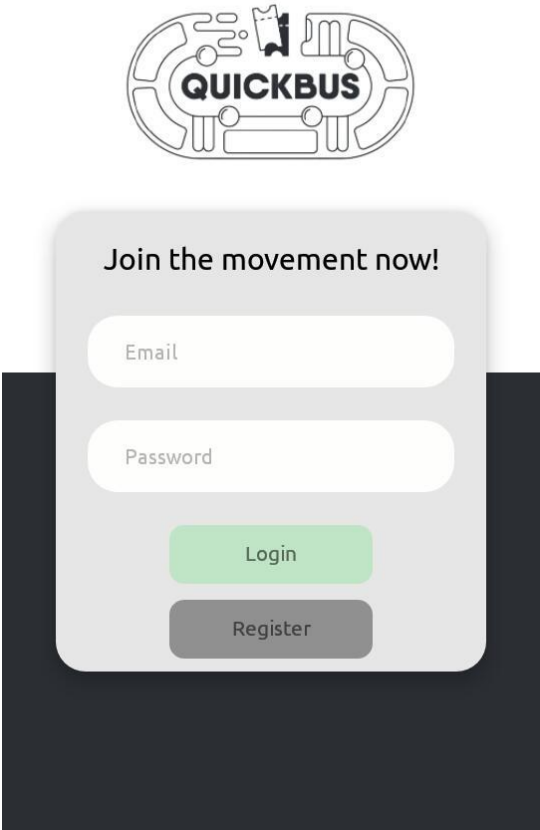
Application is designed for a user of local buses and the user always requires an internet connection to watch buses’ schedules, routes, and real-time location. For the user to ride on a bus and to pay for a ride he/she should have to register in the app. Admin should be logged in to use/access/see the data to user, schedule, buses etc.

Screenshots of Mobile App


Loading/Splash



Login



Signup



Join the movement now!

Email

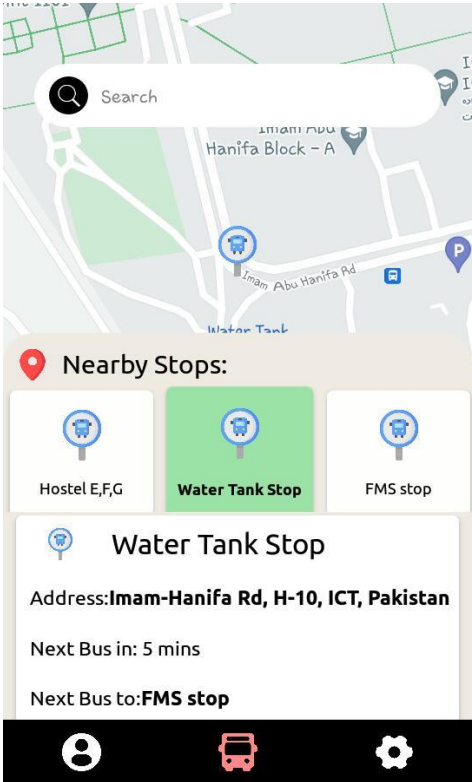
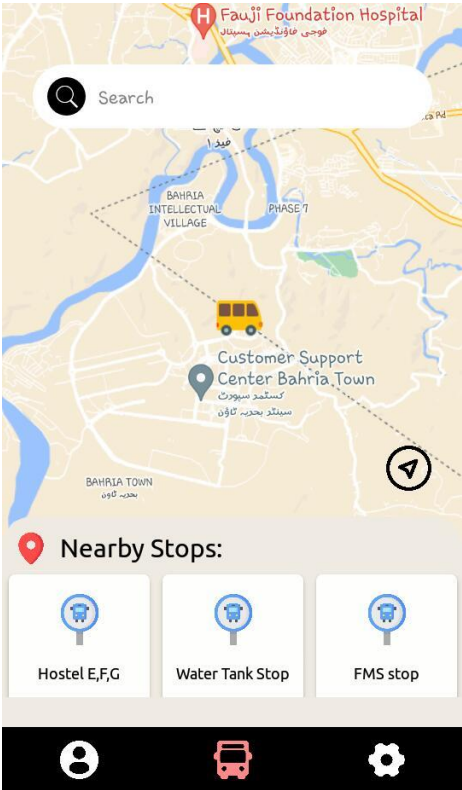
Password

Re-Enter Password

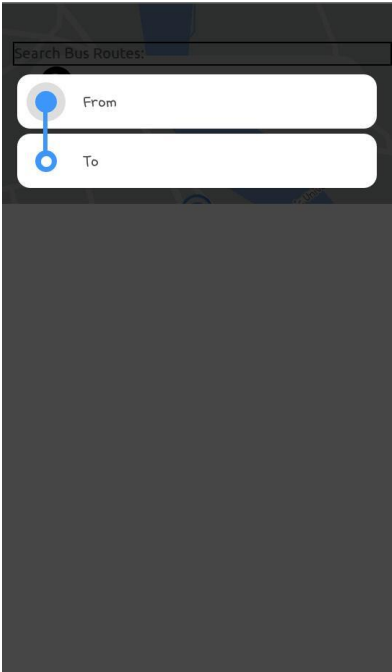
Register

Already have an account? Press here to login.

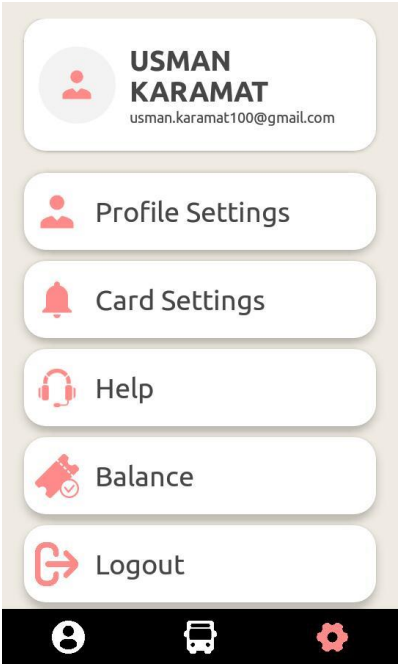
Home



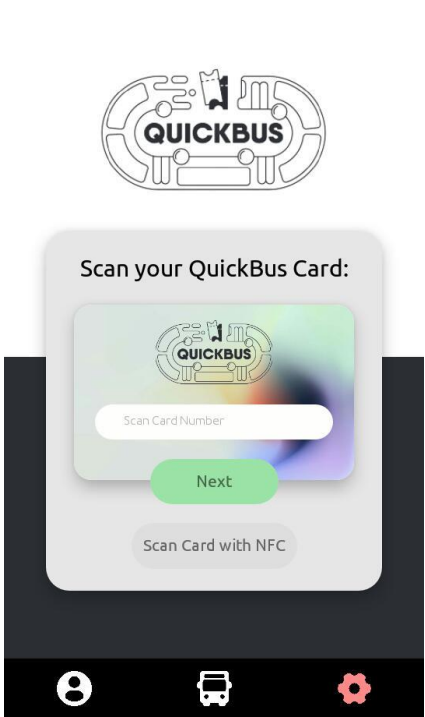
Search



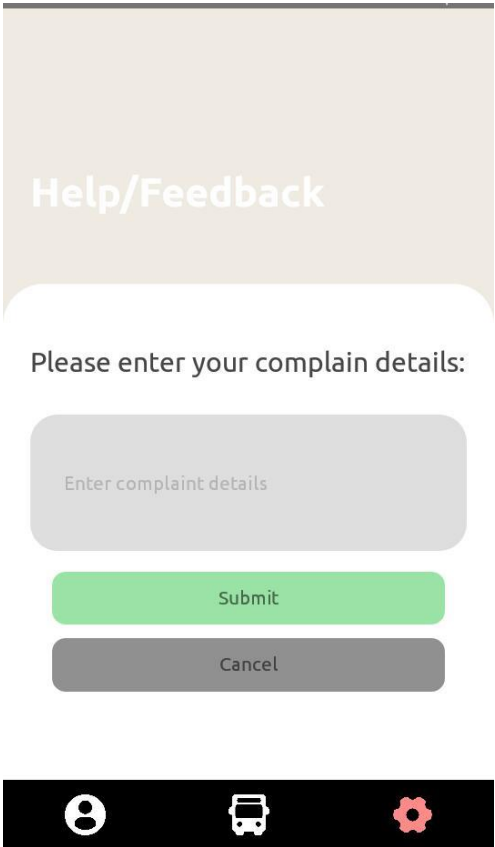
Settings



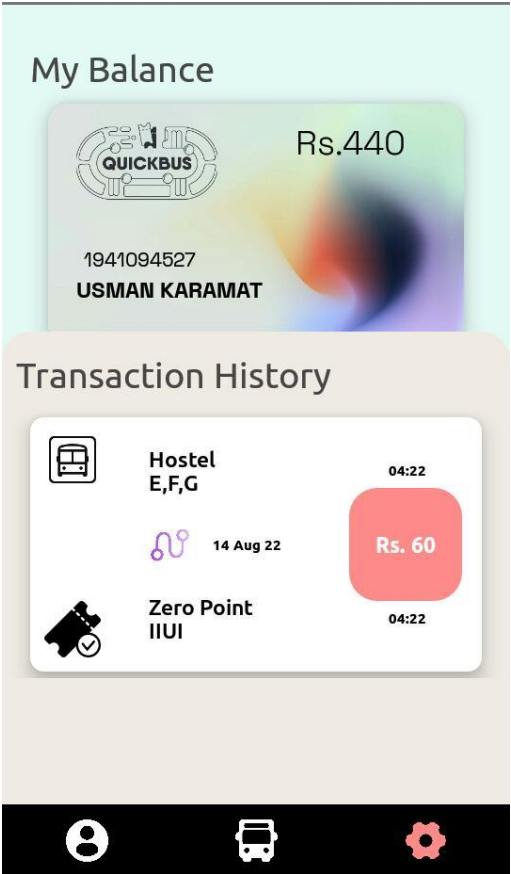
Card Setting



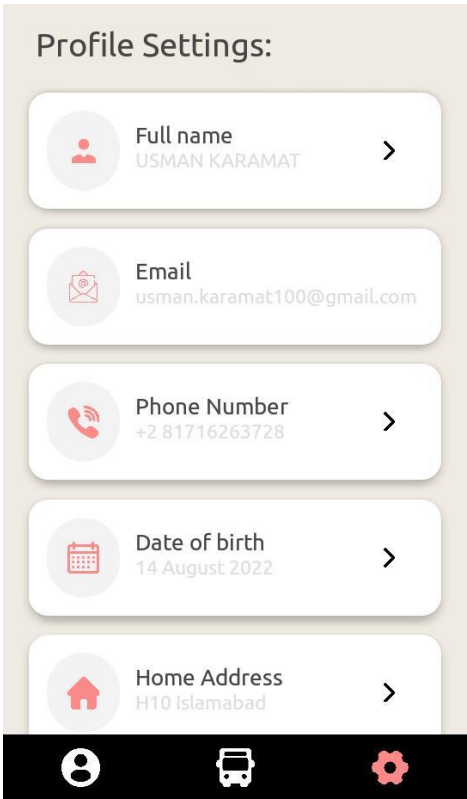
Help



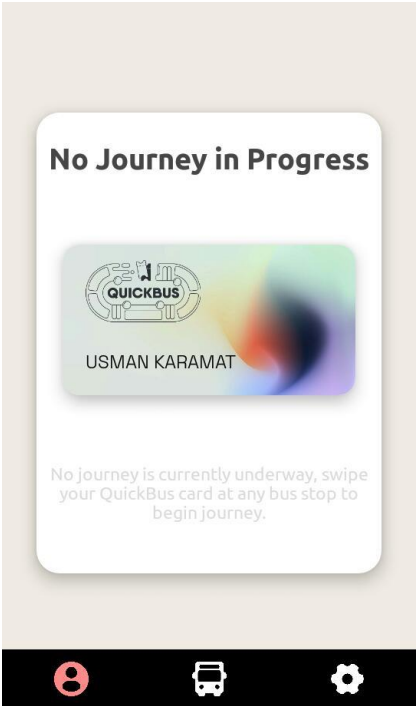
Wallet



Profile setting

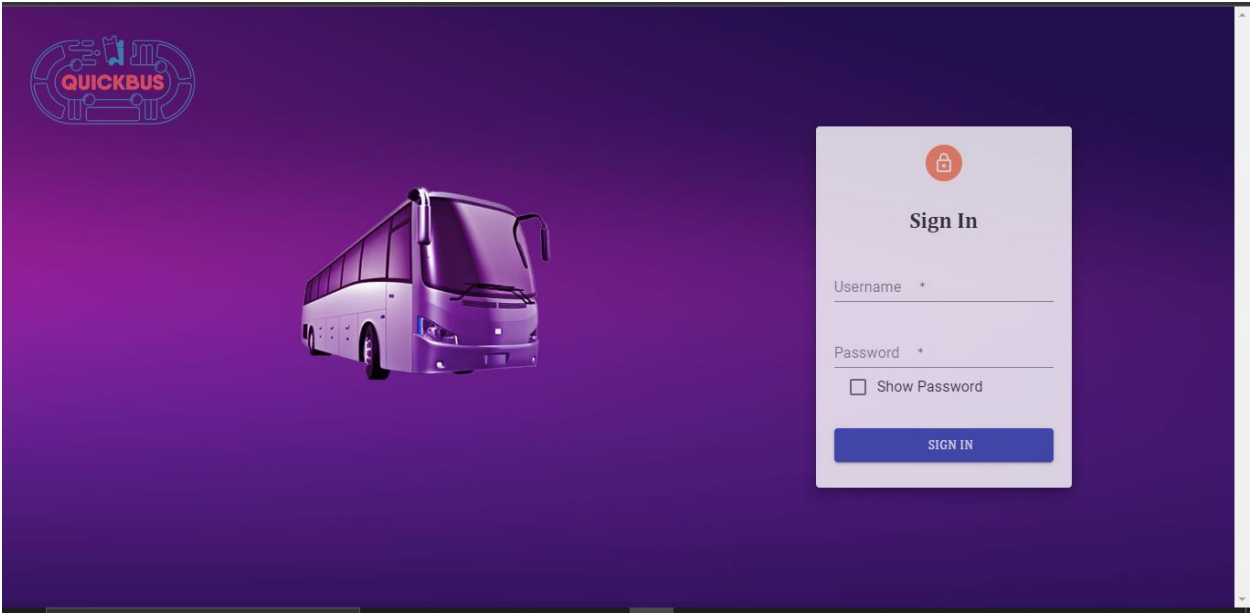


journey



Screenshots of Web (Admin Panel)
For Simple Admin

Login



Navbar

On-Time Bus

User

QUICKBUS

ON TIME BUS ADMIN

Search By CardNumber/PhoneNumber/Email/Name

| CARD NUMBER | FULL NAME | EMAIL | PHONE NUMBER | DATE OF BIRTH | AMOUNT | ACTIONS |
|--------------|--------------|-----------------------|----------------|------------------|--------|---------------------------|
| 115150239154 | Jimi Jackson | jimijackson@gmail.com | +9 23185204385 | 21 November 1999 | 1950 | <div>\$</div> <div></div> |
| 1941094527 | | rizkaramat@yahoo.com | | | 420 | <div>\$</div> <div></div> |

QUICKBUS

ON TIME BUS ADMIN

Search

| Card Number | Full name | Email | Phone Number | Date Of Birth | Amount | Actions |
|-------------|---------------|----------------------------|----------------|----------------|----------|---------------------------|
| 1941094527 | USMAN KARAMAT | usman.karamat100@gmail.com | +2 81716263728 | 14 August 2022 | 440 | <div>\$</div> <div></div> |
| 1941094527 | Muhammad U | | | | 000 1200 | <div>\$</div> <div></div> |

Add Payment

Payment

0

Save changes

Cancel

Cards

QUICKBUS

ON TIME BUS ADMIN

Cards List

Add Card (+)

| CARD ID | SHOW JOURNEY |
|----------------|--------------|
| 000000000000 | <div></div> |
| 115150239154 | <div></div> |
| 123456789 | <div></div> |
| 1941094527 | <div></div> |
| 987654321 | <div></div> |
| 99999999999999 | <div></div> |

Card Details

QUICKBUS

ON TIME BUS ADMIN

Cards List

Add Card (+)

| CARD ID | SHOW JOURNEY |
|----------------|--------------|
| 000000000000 | <div></div> |
| 115150239154 | <div></div> |
| 123456789 | <div></div> |
| 1941094527 | <div></div> |
| 987654321 | <div></div> |
| 99999999999999 | <div></div> |

Card Details

Add Card

Card Number:

Card Number Show Here

User ID:

User ID Shows Here

Save Card

Cancel

Profile

Qanat

Admin

Contact

23143245235234

qanatabbas@gmail.com

dfsa

Edit

User_Name

Qanat

Email

qanatabbas@gmail.com

Phone

23143245235234

Address

dfsa

update

User Queries

Users List

| User Name | Delete | Show |
|-----------|-------------|-------------|
| Aown Ali | <div></div> | <div></div> |

Details

ON TIME BUS ADMIN

Users List

| User Name | Delete | Show |
|-----------|--------|------|
| Aown Ali | | |

Details

Complaint ID:
R0hhYThD5DHxTObBF8SV

User Name:
Aown Ali

Date:
14 August 2022

User Complain:i have no complain

i have no complain

ON TIME BUS ADMIN

Users List

| User Name | Delete | Show |
|-----------|--------|------|
| Aown Ali | | |

Details

Complaint ID:

Date:

14 August 2022

User Complain:Fuck

Fuck

Confirmation

Are you Sure you want to delete this User

Yes

No

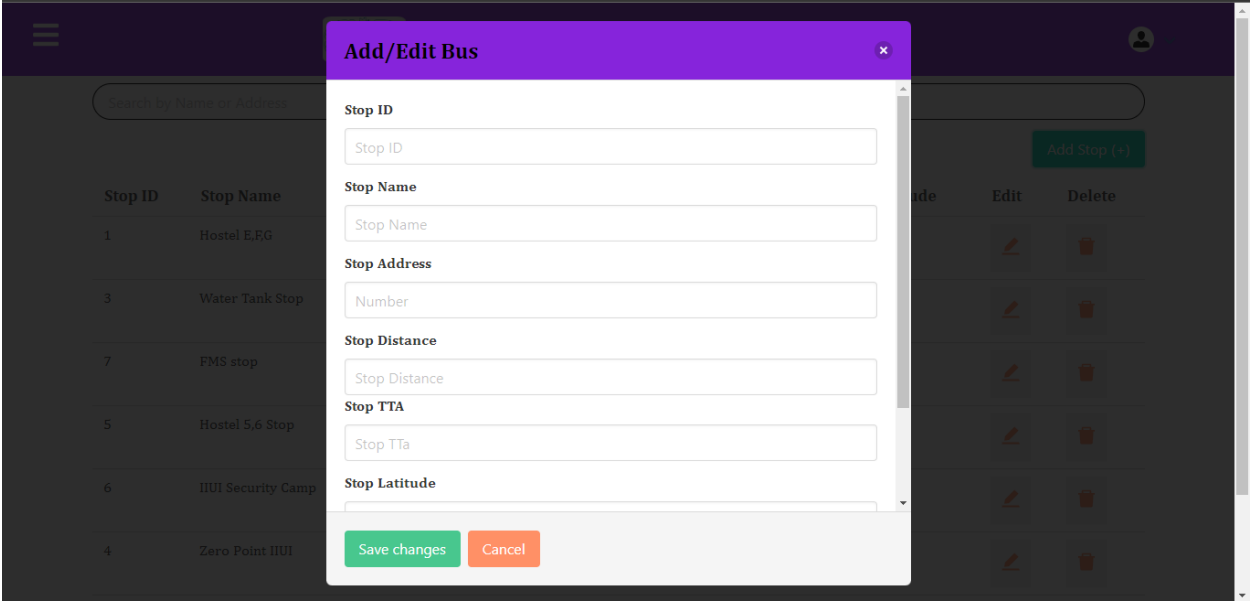
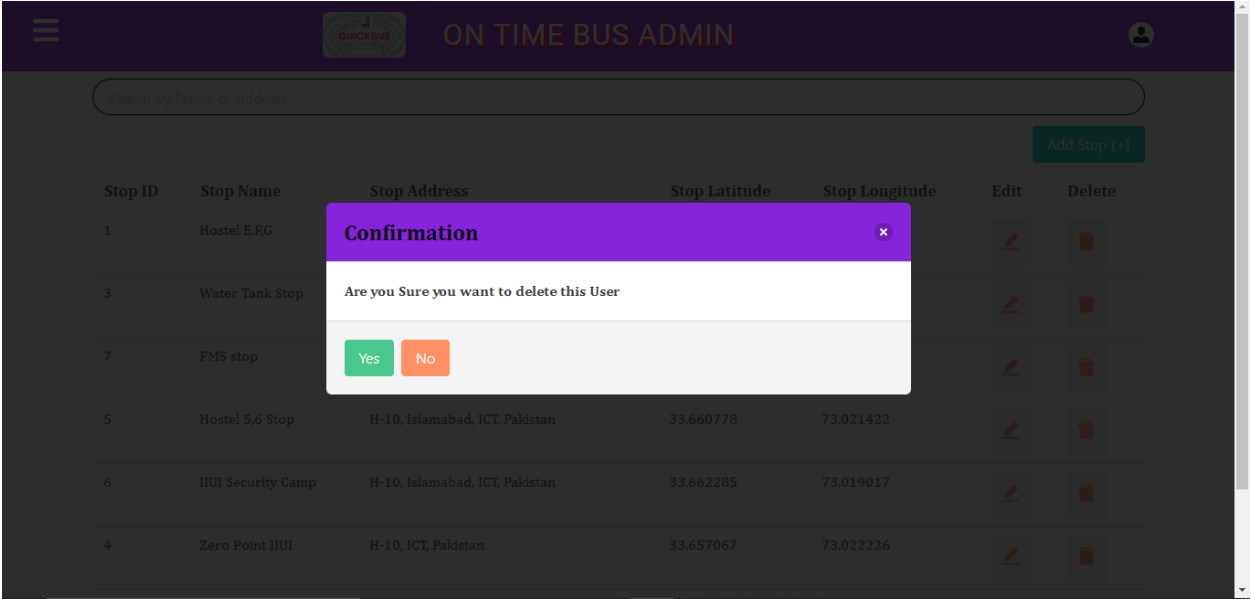
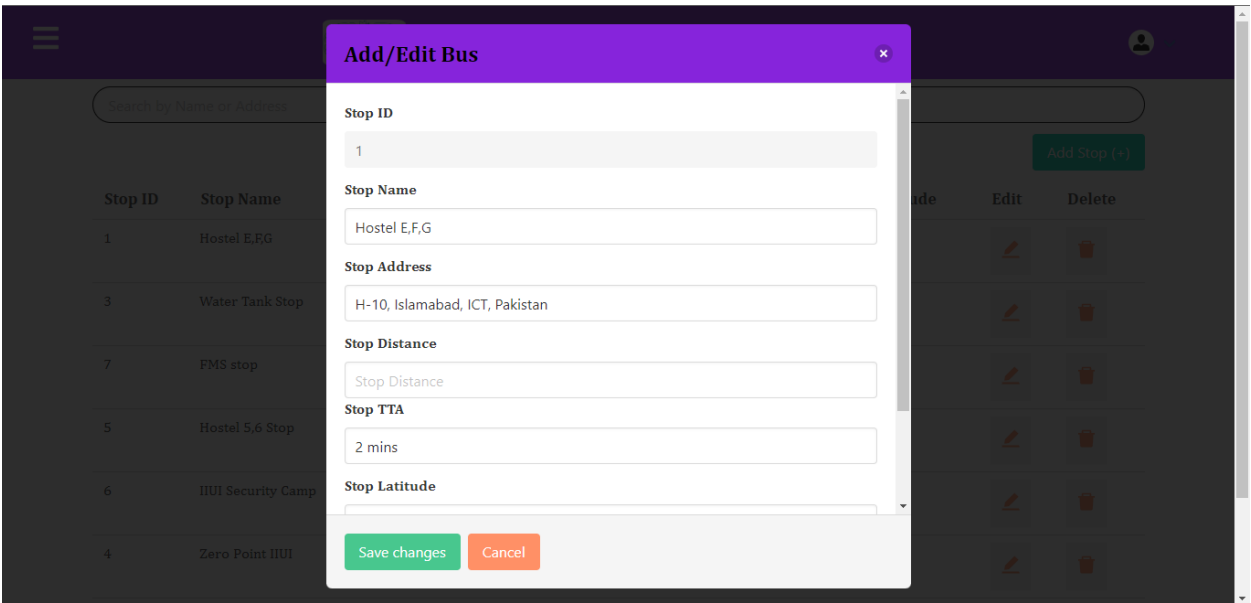
Stops

ON TIME BUS ADMIN

Search by Name or Address

Add Stop (+)

| Stop ID | Stop Name | Stop Address | Stop Latitude | Stop Longitude | Edit | Delete |
|---------|--------------------|-------------------------------------|---------------|----------------|------|--------|
| 1 | Hostel E,F,G | H-10, Islamabad, ICT, Pakistan | 33.659123 | 73.034217 | | |
| 3 | Water Tank Stop | Imam-Hanifa Rd, H-10, ICT, Pakistan | 33.655689 | 73.023094 | | |
| 7 | FMS stop | H-10, Islamabad, ICT, Pakistan | 33.66374 | 73.024782 | | |
| 5 | Hostel 5,6 Stop | H-10, Islamabad, ICT, Pakistan | 33.660778 | 73.021422 | | |
| 6 | IIUI Security Camp | H-10, Islamabad, ICT, Pakistan | 33.662285 | 73.019017 | | |
| 4 | Zero Point IIUI | H-10, ICT, Pakistan | 33.657067 | 73.022226 | | |



User Journey

QUICKBUS

ON TIME BUS ADMIN

Users List

Search By CardNumber/Name

| CARD NUMBER | USER NAME | SHOW JOURNEY |
|--------------|--------------|--------------|
| 115150239154 | Jimi Jackson | <div></div> |
| 1941094527 | | <div></div> |

Journey Details

Select User to show Journey

| JOURNEY NO | DATE | BOARDING TIME | SOURCE STOP NAME | DISEMBARKING TIME | DESTINATION NAME | FARE |
|------------|-----------|---------------|------------------|-------------------|--------------------|------|
| 0 | 21 Aug 22 | 02:35 | Water Tank Stop | | Hostel 5,6 Stop | 60 |
| 1 | 21 Aug 22 | 02:38 | Hostel E,F,G | | Hostel 5,6 Stop | 120 |
| 2 | 21 Aug 22 | 02:50 | Hostel E,F,G | | Girl's Hostel Stop | 30 |
| 3 | 21 Aug 22 | 02:51 | Hostel E,F,G | | Admin Stop | 210 |
| 4 | 21 Aug 22 | 04:26 | Hostel E,F,G | | Hostel E,F,G | 0 |
| 5 | 21 Aug 22 | 04:39 | Hostel E,F,G | | Hostel E,F,G | 0 |

For Super Admin

Navbar

X

QUICKBUS

ON TIME BUS ADMIN

Dashboard

Users

Cards

Users Journey

User Queries

Stops

Admins

Add new Admin

2

Total Users

Stops

8

Total Stops

On-Time Bus

56

Profile

Muhammad Aown Ali.

Super Admin

Contact

03229948365

aown.ali49@gmail.com

House no 729-C, Street 66, i14/4, Islamabad.

Edit

User_Name

Muhammad Aown Ali.

Email

aown.ali49@gmail.com

Phone

03229948365

Address

House no 729-C, Street 66, i14/4, Islamabad.

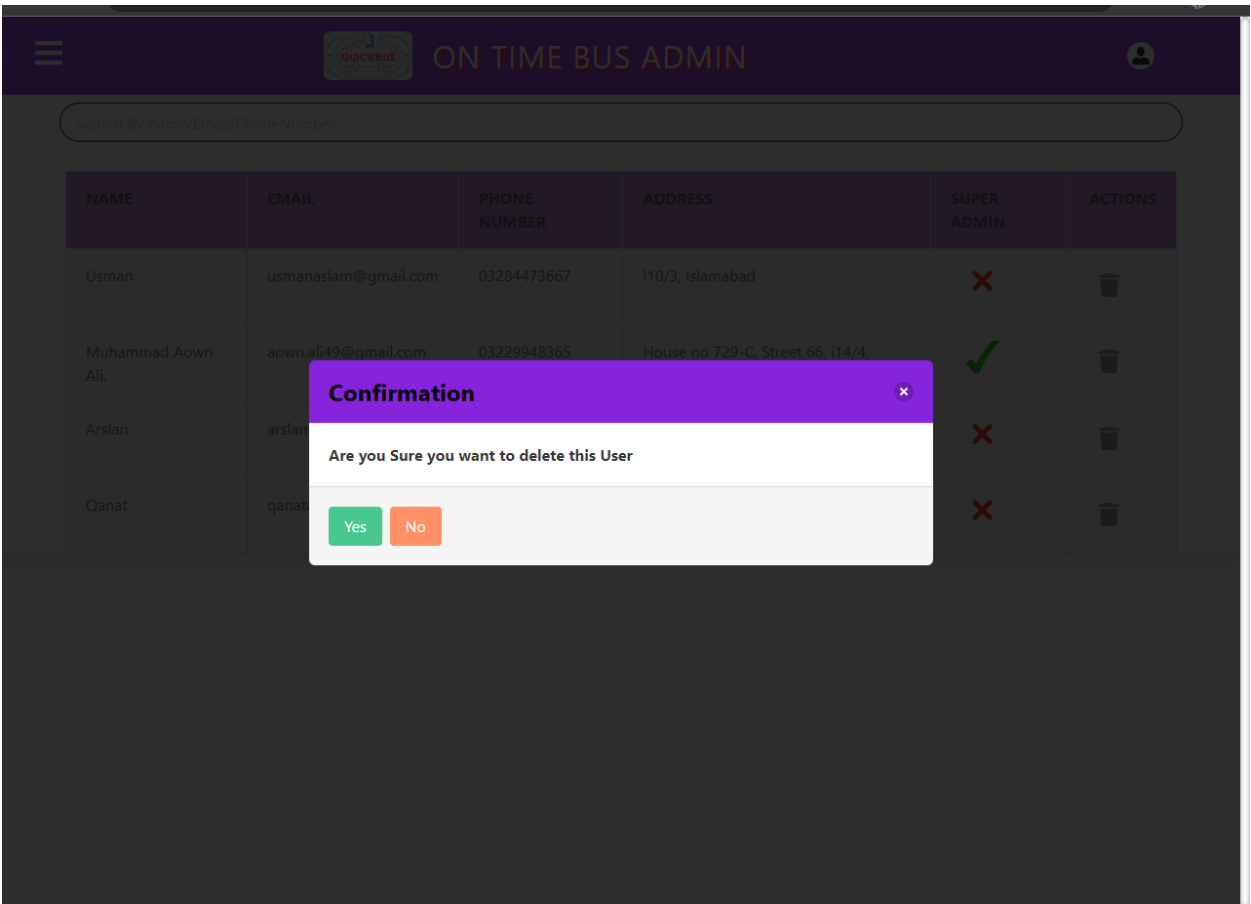
update

Admins

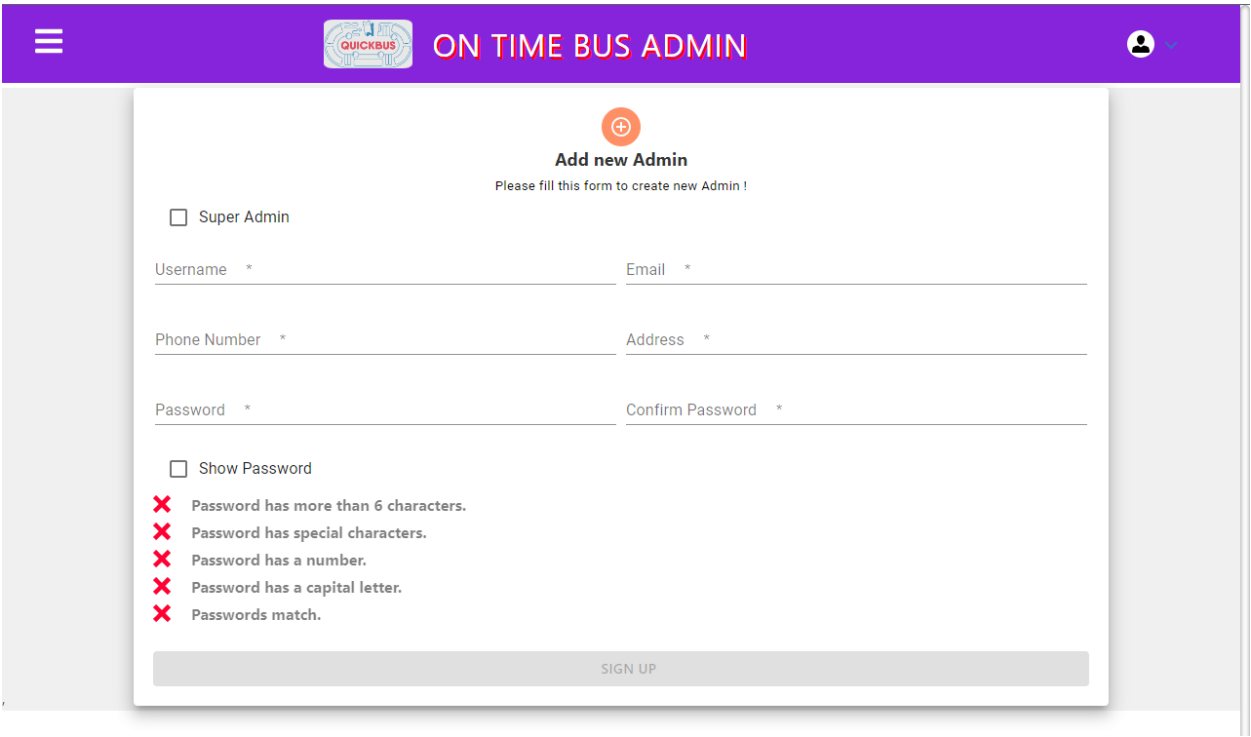
ON TIME BUS ADMIN

Search By name/Email/PhoneNumber

| NAME | EMAIL | PHONE NUMBER | ADDRESS | SUPER ADMIN | ACTIONS |
|--------------------|----------------------|----------------|--|-------------|-------------|
| Usman | usmanaslam@gmail.com | 03284473667 | i10/3, Islamabad | ✗ | <div></div> |
| Muhammad Aown Ali. | aown.ali49@gmail.com | 03229948365 | House no 729-C, Street 66, i14/4, Islamabad. | ✓ | <div></div> |
| Arslan | arslan@gmail.com | 235325 | bgjibgjeifgj | ✗ | <div></div> |
| Qanat | qanatabbas@gmail.com | 23143245235234 | dfsa | ✗ | <div></div> |



Signup



References

1. React-Native: <https://reactnative.dev/>
2. React: <https://reactjs.org/docs/getting-started.html>
3. Arduino: <https://www.arduino.cc/>
4. Firebase: <https://firebase.google.com/>
5. Stack Overflow: <https://stackoverflow.com/>
6. GitHub: <https://github.com/>