American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Seat And Fare Allocation Software for Public Transport

A Software Requirement Engineering Project Submitted

By

Semester: Spring_22_23		**Section:**	**Group Number:**	
SN	Student Name	Student ID	Contribution (CO1+CO2)	Individual Marks
18	DEBNATH, PRITOM	20-42414-1		

The project will be Evaluated for the following Course Outcomes

Evaluation Criteria	Total Marks (50)	
Introduction, Format, Submission, Defense	[10 Marks]	
System Overall Description & Functional Requirements	[10 Marks]	
System Quality Attributes and Project Requirements	[10 Marks]	
UML and E-R Diagram with Data Dictionary	[10 Marks]	
UI/UX Prototyping	[10 Marks]	

Software Requirements Specification

for

<Seat And Fare Allocation Software for Public Transport>

Version 2.0 approved

Prepared by< DEBNATH PRITOM >

< American International University Bangladesh>

<30.04.2023>

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Group 01	10/04/2023	Redundancy in intended audience and reading suggestion, Unrelated information in operating environment, incomplete identification of user classes, missing non-functional requirements.	1.0
Group 01	28/04/2023	Updated Class diagram and use case diagram	2.0

1. Introduction

1.1 Purpose

The purpose of this software is to facilitate the allocation of seats and fares for various modes of transportation, such as buses, taxis. This software aims to improve the efficiency and accuracy of the transportation industry by automating the seat and fare allocation process. The software helps to eliminate human errors and bias, while also reducing the time required to allocate seats and fares. The scope of this software includes the allocation of seats and fares for various modes of transportation, as well as the management of passenger data, payment processing, and tracking of real-time data. This software can be used by transportation companies, government agencies, and other organizations involved in the transportation industry.

The software's objectives include improving the transportation industry's efficiency, reducing costs, and enhancing the overall passenger experience. The software also aims to provide a user-friendly interface that is easy to navigate and understand, while also ensuring the security of passenger data. The software is aligned with corporate goals and business strategies by providing transportation companies with a competitive advantage, improving customer satisfaction, and increasing revenue.

1.2 Document Conventions

The manuscript is written in a simple manner, yet it follows a certain pattern to designate unique identities. Content titles with bold words, italic words indicate user classes. Normal text format is used to describe the hardware and software requirements.

This document assists in providing relevant information to stakeholders, providing an effective medium for interaction, and providing high-level information on the project's ideas in context. The report also serves as a platform for stakeholders to have a better understanding of the project. During the development of this system, this document is intended to provide interested parties with user-friendly and accessible system information.

Table 1: Abbreviations used in the SRS documents.

Abbreviations	Full form
SRS	Software requirements specification
AWS	Amazon Web Services
API	Application programming interface

1.3 Intended Audience and Reading Suggestions

The target audience for this document is Project Managers, Users, Developers, and Requirements Engineers. It is highly recommended that you read the paper to understand pharmaceuticals e-commerce before reading this document.

1.3.1 Developers:

Developers are the primary audience for technical documentation such as software design documents, code comments, and API documentation. They need to understand the technical details of the system, such as how seat allocation algorithms are implemented, how payment processing is handled, and how the software integrates with other systems.

1.3.2 Project managers:

Project managers are responsible for overseeing the development process and ensuring that it is on track and on budget. They need to understand the project scope, timelines, and resource requirements, as well as the risks and issues that may arise during the project. They may also be interested in reports and analytics on passenger occupancy and revenue generated by the software.

1.3.3 Customer support staff:

Customer support staff may need to understand how the software works in order to assist passengers who are having issues with the system. They may need access to training materials and knowledge base articles.

1.3.4 Compliance officers:

Compliance officers are responsible for ensuring that the software meets regulatory requirements, such as data privacy laws and payment processing regulations. They may need to review documentation related to security and compliance.

1.3.5 IT administrators:

IT administrators are responsible for installing, configuring, and maintaining the software. They may need access to technical documentation such as installation guides and system administration manuals.

1.3.6 Legal counsel:

Legal counsel may need to review the software requirements and documentation to ensure that there are no legal risks associated with the software. They may also be involved in negotiating contracts with vendors or customers.

1.3.7 Marketing staff:

Marketing staff may need to understand the key features and benefits of the software, as well as the target audience and value proposition. They may be interested in understanding how the software can help public transport operators differentiate themselves from competitors and attract more passengers.

1.3.8 Users:

Users are the people who will be using the software, such as bus or train passengers. They need to understand how to purchase tickets, select seats, and pay fares using the software. The documentation should be written in a user-friendly and accessible language.

1.3.9 Testers:

Testers are responsible for ensuring that the software meets the requirements and is free of defects. They need to understand the testing requirements, test cases, and test results. They may also be involved in testing the user interface and user experience of the software.

1.3.10 Documentation writers:

Documentation writers are responsible for creating user manuals, help files, and other documentation to assist users in using the software effectively. They need to understand the user perspective and be able to communicate technical information in a clear and concise manner.

1.4 References

1.4.1 The Software Engineering textbook by way of Ivan Marsic.

1.4.2 User interface style guide:

This document provides guidelines for the design and layout of the software's user interface. It includes information about typography, color schemes, and iconography. Following “The design of everyday things” by Don Norman last revision was 2013; the UI is developed.

2. Overall Description

The "Seat and Fare Allocating Software for Public Transport" project aims to create a software system that enables passengers to purchase tickets, select seats, and pay fares for public transportation services. The system will also enable transportation operators to manage their fleets, track passenger data, and analyze usage patterns. The software will be designed to be user-friendly and intuitive, with a focus on ensuring that passengers can easily access and use the system to purchase and manage their tickets.

The software will be developed using modern software development practices and will be built to be scalable and extensible. The system will be designed to integrate with existing transportation infrastructure and payment systems, enabling seamless connectivity with other systems. The software will also be designed to be secure and compliant with industry and regulatory standards, including those related to payment processing and data privacy.

The successful implementation of the "Seat and Fare Allocating Software for Public Transport" project will provide significant benefits to both transport operators and passengers. Transportation operators will benefit from improved fleet management and data analytics capabilities, while passengers will benefit from a more streamlined and convenient ticketing and payment system. Ultimately, the software system will improve the overall efficiency and effectiveness of public transportation services, enabling more people to easily access and use public transportation.

2.1 Product Perspective

Business Requirements:

2.1.1 Ticketing and Payment

The software should enable passengers to purchase tickets online and make payments using a variety of payment methods, including credit/debit cards, mobile payments, and digital wallets. The software should also allow passengers to select their preferred payment method and should provide a secure and reliable payment processing system.

2.1.2 Seat Selection

The software should allow passengers to view and select available seats on their chosen transportation service. The system should also provide real-time updates on seat availability and allow passengers to switch seats if desired.

2.1.3 Fare Calculation

The software should calculate fares based on a variety of factors, including distance traveled, route taken, and any additional services provided. The software should also provide passengers with a clear and transparent breakdown of the fare calculation, including any additional fees or charges.

2.1.5 Data Analytics

The software should enable transportation operators to analyze passenger data, including usage patterns, travel preferences, and payment history. The system should also provide operators with actionable insights into passenger behavior, enabling them to optimize their services and improve overall efficiency.

2.1.6 Security and Compliance

The software should be designed to be secure and compliant with industry and regulatory standards, including those related to payment processing and data privacy. The system should also include robust security features, including encryption, authentication, and access controls, to protect against unauthorized access and data breaches.

2.1.7 User Experience

The software should be designed to be user-friendly and intuitive, with a focus on ensuring that passengers can easily access and use the system to purchase and manage their tickets. The system should also include features such as a mobile app and responsive web design to ensure that passengers can access the system from a variety of devices and platforms.

2.1.8 Integration

The software should be designed to integrate with existing transportation infrastructure and payment systems, enabling seamless connectivity with other systems. The system should also be designed to be extensible, enabling new features and capabilities to be added as needed.

Context and Origin:

The "Seat and Fare Allocating Software for Public Transport" is a new, self-contained software product that is being developed to provide a streamlined and efficient ticketing and payment system for public transportation. The product is not a follow-on member of an existing product family, nor is it a replacement for any existing system.

The software will be a component of a larger system that includes various transportation services, such as buses, trains, and subways. The software will integrate with existing transportation infrastructure and payment systems, enabling seamless connectivity with other systems.

2.2 Product Functions

- Allow new users to register with the software and provide their personal details such as name, email address, phone number, and home address.
- Validate user credentials during registration to ensure that they have a valid email address and phone number.
- Automatically track the position coordinates of the passenger and take input about the destination and preferred vehicle type.
- Trace nearby transports available based on passenger location and vehicle availability.
- Allow passengers to view the real-time GPS locations of their rides or transports they are waiting for.
- Deduct a fee from passenger’s mobile banking account or SIM balance when a transport is confirmed, with rates varying based on the vehicle type.
- Allow passengers to delay payment with added increment.
- Notify local law and order authorities in case of any emergency situation while in a journey.

2.3 User Classes and Characteristics

The project is designed to fulfil the needs of various user classes. The following are the identified user classes along with their characteristics:

- *Admin:* The admin has full access to the system and is responsible for managing the allocation of seats and fares. They have the technical expertise to use all the features of the software and have high-level security or privilege levels.
- *Customer Support*: Customer support agents have access to the system and are responsible for handling customer queries, issues and complaints. They have the technical expertise to use the features of the software related to customer support.
- *Passengers:* • Passengers are the end-users of the software who use the system to book tickets and manage their travel bookings. They may have varying levels of technical expertise and educational levels.
- *Management:* The management team has access to the system and is responsible for analyzing the data generated by the software to make business decisions. They may have varying levels of technical expertise and educational levels.
- *Drivers:* These are the operators of the public transport vehicles who use the software to manage their routes, schedules, and passenger information. They may have varying levels of technical expertise and may require training on how to use the software effectively. They expect the software to be intuitive, responsive, and user-friendly, with clear and concise instructions.
- *Maintenance Personnel:* These are the technicians and support staff who use the software to perform maintenance and repairs on the public transport vehicles and infrastructure. They require access to detailed technical documentation, schematics, and repair manuals, as well as the ability to track maintenance schedules and performance metrics. They expect the software to be reliable, accurate, and up-to-date, with real-time notifications and alerts for critical issues.
- *Tourists*: These are visitors to the area who use the public transport system to explore the city and surrounding areas. They may be less familiar with the local transit system and may require more assistance and guidance from the software. They expect the software to provide detailed information about routes, schedules, fares, and popular destinations, as well as user-friendly navigation and search features.
- *Special Needs Passengers*: These are passengers who require special assistance or accommodations when using public transportation, such as immersive reading, or video tutorial. They require the software to be designed with their needs in mind, with clear and concise instructions, and accessible features that are easy to use.
- *Security Personnel*: These are law enforcement or security personnel who use the software to monitor and track suspicious activity or incidents on public transport vehicles or at stations. They require access to real-time data and video feeds, as well as the ability

to communicate with other security personnel or emergency services. They expect the software to be secure, reliable, and capable of handling large volumes of data in real-time.

The admin and customer support are the most important user classes as they have high-level access and are responsible for managing the allocation of seats and fares and handling customer queries, issues, and complaints respectively.

2.4 Operating Environment

The transport allocating software is intended to operate in a web-based environment, accessible through a modern web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge. The following are the minimum recommended specifications for the hardware and software components:

2.4.1 Hardware

CPU: Intel Core i3 or equivalent
RAM: 4 GB or higher
Internet connection: Broadband connection or higher

2.4.2 Software

Operating system: Windows 10 or higher, macOS, or Linux, Android
Web browser: Google Chrome, Mozilla Firefox or Microsoft Edge

The transport allocating software will be developed using modern web technologies such as HTML5, CSS3, and JavaScript. It will communicate with a back-end server that is responsible for storing and processing data, which can be deployed on a variety of platforms including Windows, Linux, or cloud-based servers such as Amazon Web Services (AWS) or Microsoft Azure.

The software must peacefully coexist with other web-based applications that may be running on the same hardware and using the same web browser. It should not interfere with any other software components on the user's system, and should not cause any conflicts with the operating system or any other software applications.

2.5 Design and Implementation Constraints

There are several items and issues that will limit the options available to the developers of the "Seat and Fare Allocating Software for Public Transport" These include:

2.5.1 Regulatory policies

The software must comply with all applicable regulatory policies related to public transportation and payment systems.

2.5.2 Hardware Limitations

The software must be designed to run on a range of hardware configurations, including both desktop and mobile devices. The software must also be optimized for performance, taking into account timing and memory requirements.

2.5.3 Interfaces to other applications
The software must interface with existing transportation infrastructure and payment systems, including ticketing software and fleet management systems.

2.5.4 Specific technologies, tools, and databases
The software must be developed using specific technologies, tools, and databases that have been approved by the project team, including programming languages, development frameworks, and database management systems.

2.5.5 Language requirements
The software must support multiple languages, including English and any other languages that are required by regulatory policies or customer needs.

2.5.6 Communications protocols
The software must comply with specific communications protocols, including secure data transmission and encryption protocols.

2.5.7 Security considerations
The software must be designed with security in mind, including access control, data encryption, and other measures to protect sensitive information.

2.5.8 Design conventions or programming standards
The software must be developed using specific design conventions and programming standards that have been approved by the project team, ensuring consistency and maintainability of the codebase.

2.6 User Documentation

The following user documentation components will be delivered along with the software:

2.6.1 User manual: A comprehensive guide that provides step-by-step instructions for using the software, including how to book seats, purchase tickets, and manage fares.

2.6.2 Online help: An online help system that provides context-sensitive assistance to users while they are using the software.

2.6.3 Tutorials: A series of interactive tutorials that introduce users to the software's features and functionality, including how to search for available seats, choose a route, and pay for fares.

3. System Requirements

3.1 System Features

3.1 Passenger Registration

Functional Requirements (FRs)

3.1.1 New users shall register with the software using their name, email address, phone number, and home address.

3.1.2 Registration shall fail for faulty phone numbers or email addresses.

Priority Level: High
Precondition: User have valid phone number or e-mail.

Cross-references: 3.2

3.2 Software Login

Functional Requirements (FRs)

3.2.1 The software shall allow users to login with their given username and password.
3.2.2 The login credentials (username and password) will be verified with database records. If the login successful, the home page of the user account will be displayed.
3.2.3 If the username and/or password has been inserted wrong, the random verification code will be generated and sent to the user's email address by the system to retry login.

3.1.3 If the number of login attempt exceed its limit (3 times), the system shall block the user account login for one hour *[optional function]*

Priority Level: High
Precondition: User have valid user id and password
Cross-references: 3.1. 3.3

3.3 Dispatcher

Functional Requirements (FRs)

3.3.1 The software shall automatically track the passenger's position coordinates and take input about the destination and preferred vehicle type.

3.3.2 The software shall trace nearby available transports.

Priority Level: High
Precondition: User must have a GPS connection
Cross-references: 3.2.2, 3.5

3.4 Fee Provider

Functional Requirements (FRs)

3.4.1 A small fee shall be deducted from the passenger's mobile banking account or SIM balance when they confirm a transport.

3.4.2 Passengers shall delay the payment with added increments.

3.4.3 The fee rates shall vary for different transport company.

Priority Level: Medium
Precondition: User must have any online payment account
Cross-references: 3.2.2, 3.8

3.5 Real-Time Location Viewer

Functional Requirements (FRs)

3.5.1 Passengers shall view the real-time GPS locations of their current or waiting transport.

3.5.2 The transport shall be registered in the client database.

Priority Level: Medium
Precondition: User must have a GPS connection
Cross-references: 3.2.2

3.6 Security Alarm

Functional Requirements (FRs)

3.6.1 In case of an emergency, the passenger shall prompt the security alarm, which will immediately notify the local law and order authorities.

3.6.2 The security alarm shall work in unison with the real-time location viewer.

Priority Level: High
Precondition: N/A
Cross-references: 3.2.2

3.7 Seat Booking

Functional Requirements (FRs)

3.7.1 The system shall allow users to select their desired bus or train route, departure time, and seat location, and book a seat using an online interface.

3.7.2 The system shall display the availability of seats in real-time and prevent double bookings.

3.7.3 Once a seat has been booked, the system shall generate a confirmation number and send it to the user via email or text message.

Priority Level: High
Precondition: User have a registered account
Cross-references: 3.2.2

3.8 Fare Calculation

Functional Requirements (FRs)

3.8.1 The system shall calculate the fare for each journey based on the distance traveled, type of vehicle, and any discounts or promotions applied.

3.8.2 The fare calculation shall take into account any changes in route or stopovers along the way, and be based on up-to-date fare data.

3.8.3 The fare for each journey shall be displayed to the user before they confirm their booking.

Priority Level: High
Precondition: User must select the pick-up and destination place
Cross-references: 3.3

3.9 Payment Processing

Functional Requirements (FRs)

3.9.1 The system shall process payments securely and efficiently, using a variety of payment methods including credit card, debit card, and mobile payment options.

3.9.2 The system shall store payment details securely and be compliant with industry standards for payment processing.

3.9.3 The system shall generate a payment receipt for each transaction, and provide users with the option to save their payment details for future bookings.

Priority Level: High
Precondition: Payment credential must be valid
Cross-references: 3.2.2, 3.4, 3.8

3.10 Route Planning and Optimization

Functional Requirements (FRs)

3.10.1 The system shall optimize routes and schedules for each vehicle, taking into account factors such as demand, traffic congestion, and vehicle capacity.

3.10.2 The system shall generate optimized routes and schedules for each vehicle and driver, and be able to adjust them in real time based on changes in demand or traffic conditions.

3.10.3 The system shall also provide drivers with route maps and directions to ensure they follow the optimized routes.

Priority Level: Medium
Precondition: User must have a GPS connection
Cross-references: 3.5

3.2 Non-Functional/Quality Requirements

3.2.1 Availability: This projects goal shall be providing the customer with 24-hour service with little to no discrepancies.

Priority Level: High
Precondition: None
Cross-references: N/A

3.2.2 Interoperability: The output of the project shall allow it to communicate with other government apps.

Priority Level: Low
Precondition: None
Cross-references: N/A

3.2.3 Maintainability: The system shall be easy to maintain and support, with well-documented code and architecture, and clear instructions for deployment, testing, and troubleshooting. The system shall use standardized coding practices and design patterns to facilitate maintenance and updates, and be compatible with industry-standard tools and technologies.

Priority Level: Medium
Precondition: None
Cross-references: N/A

3.2.4 Performance: The system shall be able to handle a high volume of concurrent users and transactions, with response times of less than two seconds for seat booking and payment processing operations. The system shall be able to scale horizontally to accommodate additional users and traffic spikes, without any degradation in performance.

Priority Level: High
Precondition: None
Cross-references: N/A

3.2.5 Usability: The system shall be easy to use and intuitive, with a user-friendly interface that enables users to complete booking and payment processes with minimal effort. The system shall be accessible to users with disabilities, and comply with relevant accessibility guidelines and standards.

Priority Level: Medium
Precondition: None

Cross-references: N/A

3.2.6 Security: The system shall be secure and protect user data and payment information against unauthorized access, hacking, and other security threats. The system shall use encryption and other security measures to protect sensitive data in transit and at rest, and comply with relevant security standards and regulations.

Priority Level: High
Precondition: None
Cross-references: N/A

3.2.7 Reliability: The system shall be reliable and available 24/7, with a minimum of 99.9% uptime. The system shall be able to recover quickly from any system failures or data losses, and have a backup and disaster recovery plan in place.

Priority Level: Medium
Precondition: None
Cross-references: N/A

3.3 Project Requirements.

3.3.1 The project shall be developed using modern programming languages and frameworks, such as Java, Python, React, and Node.js, using API. All code must be well-documented, modular, and adhere to industry-standard coding practices and design patterns.

3.3.2 The project team shall consist of at least 6 developers, 2 testers, 1 project manager, and 1 documentation specialist. Additional resources may be added as needed with the approval of the project manager.

3.3.3 The project shall follow an agile development methodology, with sprints of 2 weeks and regular stand-up meetings, retrospectives, and demos.

3.3.4 The project budget shall not exceed $2 million, including all development, testing, documentation, and deployment costs.

3.4.5 The project shall undergo rigorous testing at multiple stages of development, including unit testing, integration testing, system testing, and user acceptance testing. All bugs and issues identified during testing must be resolved before the software is released to production.

3.4.6 The project shall be deployed on a cloud-based infrastructure, such as Amazon Web Services (AWS), to ensure scalability, high availability, and security. The deployment process shall be automated and repeatable, using tools such as Ansible or Docker.

3.4.7 The project shall be delivered with comprehensive documentation, including user manuals, installation guides, release notes, and API documentation. The documentation must be easy to understand and follow, and be available in multiple formats, such as PDF and online help.

4. Design and Interface Requirements

4.1 UML Diagrams

4.1.1 Use case diagram:

In this diagram, we have six actors: Admin, Customer Support, Customers, Drivers, Tourists, and Special Needs Passengers. The admin actor has the ability to manage fares, routes, and drivers in the system. The Customer Support actor has the ability to handle customer inquiries and complaints. The Customers actor can book tickets, view bookings, cancel bookings, check fare rates, and view transport status. The Drivers actor has the ability to manage routes, schedules, and view passenger information. The Tourists actor can book tickets, cancel bookings, check fare rates, view popular destinations, and view routes. The Special Needs Passengers actor has access to accessible features, immersive reading, video tutorials, and clear instructions.

All of these actors interact with the Seat and fare allocating software to perform their respective tasks.

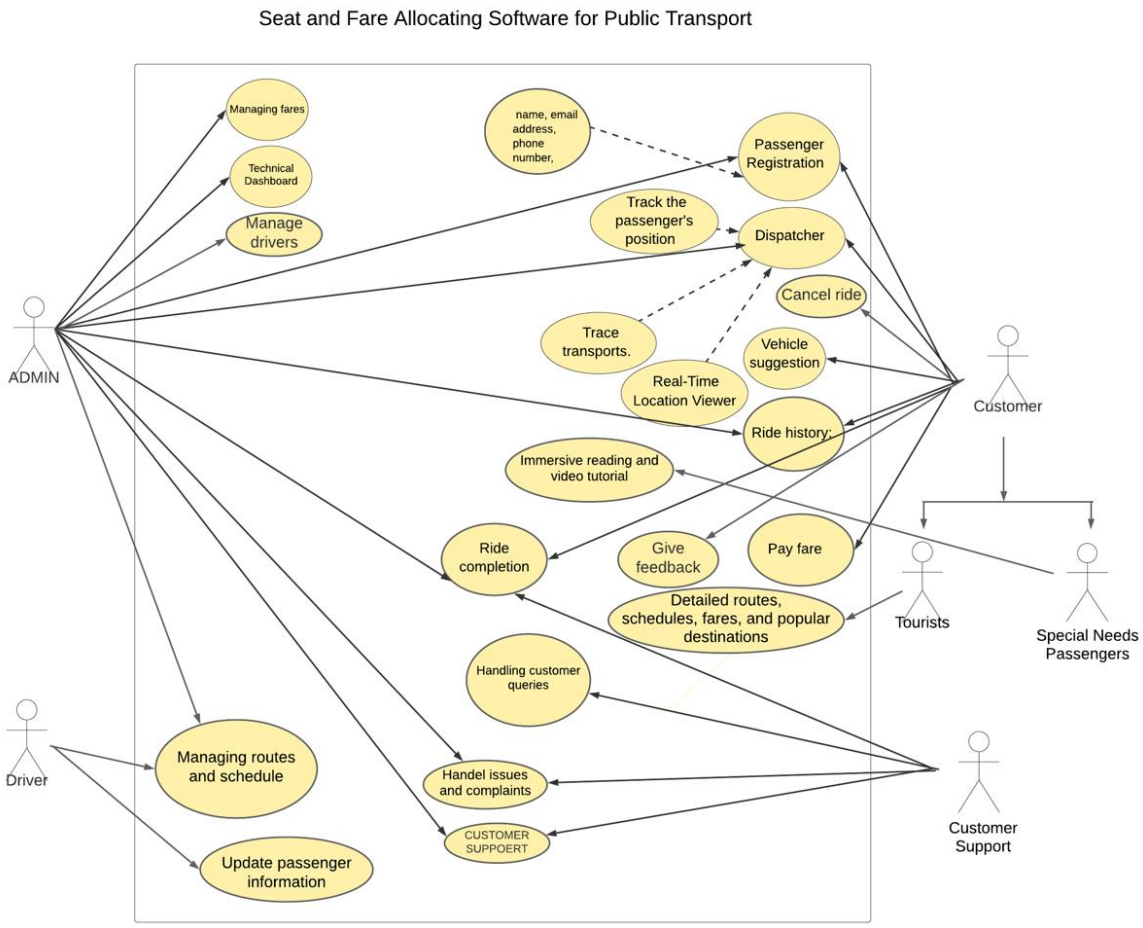

Figure-1: Use-case diagram

4.1.2 Class diagram

In this diagram, we have five main classes: Passenger, Driver, Administrator, Customer Support and Vehicle.
The Passenger class represents the passengers who use the system to book trips and manage their travel bookings. They have attributes such as name, email, phone, and address, and methods such as bookTrip() and viewTrip().

The Driver class represents the operators of the public transport vehicles who use the system to manage their schedules, routes, and passenger information. They have attributes such as name, id, vehicle, and schedule, and methods such as updateSchedule() and viewSchedule().

The admin class represents the users who have full access to the system and are responsible for managing the allocation of seats and fares. They have attributes such as name, id, role, and password, and methods such as addDriver(), removeDriver(), and viewReport().

The Vehicle class represents the vehicles that are part of the public transport system. They have attributes such as id, type, seats, and driver, and methods such as addVehicle(), removeVehicle(), and viewVehicles().

The Customer support class represents the users who handles the feedback from the passenger. They have attributes such as name, id, role, and password, and methods such as viewFeedback(), handelFeedback() and delivarReport().

Overall, this class diagram provides a high-level view of the main classes and their relationships in the public transportation management system.

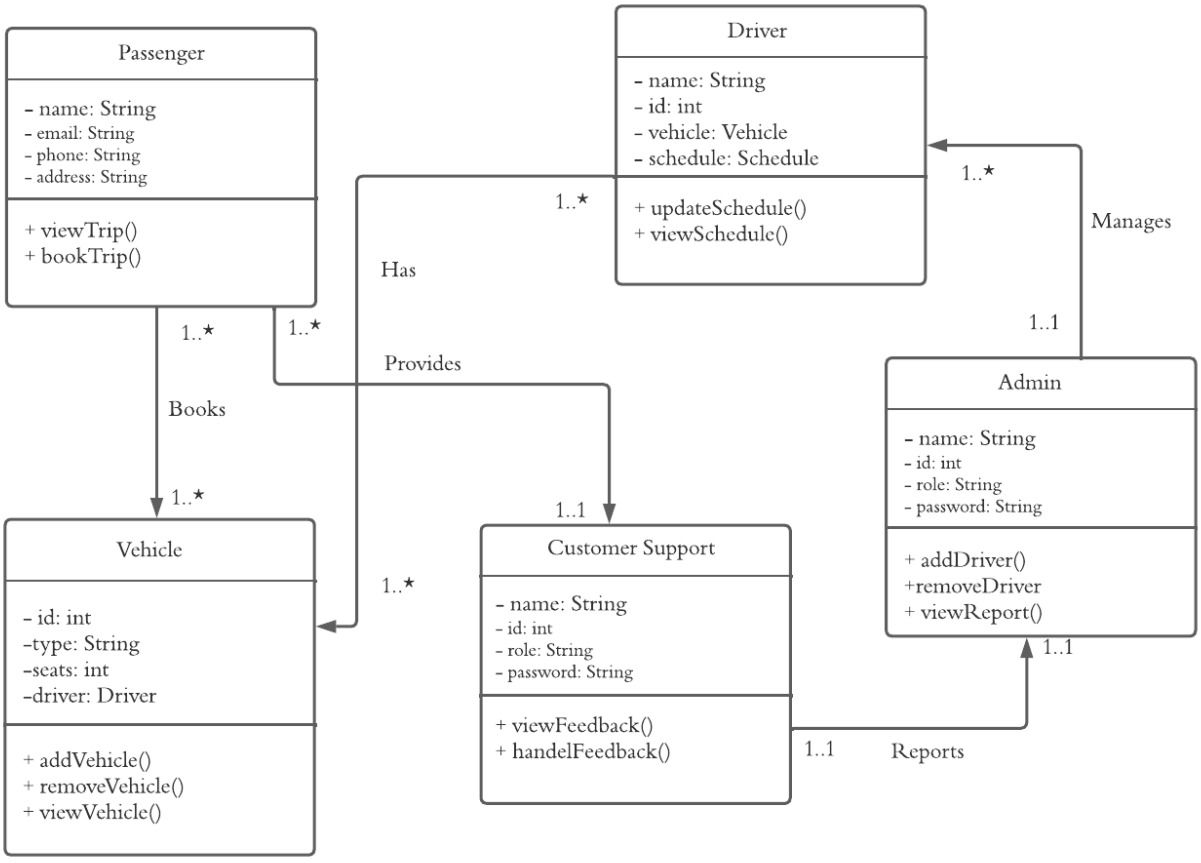

Figure-2: Class Diagram

4.1.3 Activity diagram

In this activity diagram, admin manages the fares, customer login to the system, book tickets, sends feedback, customer support handles the feedback.

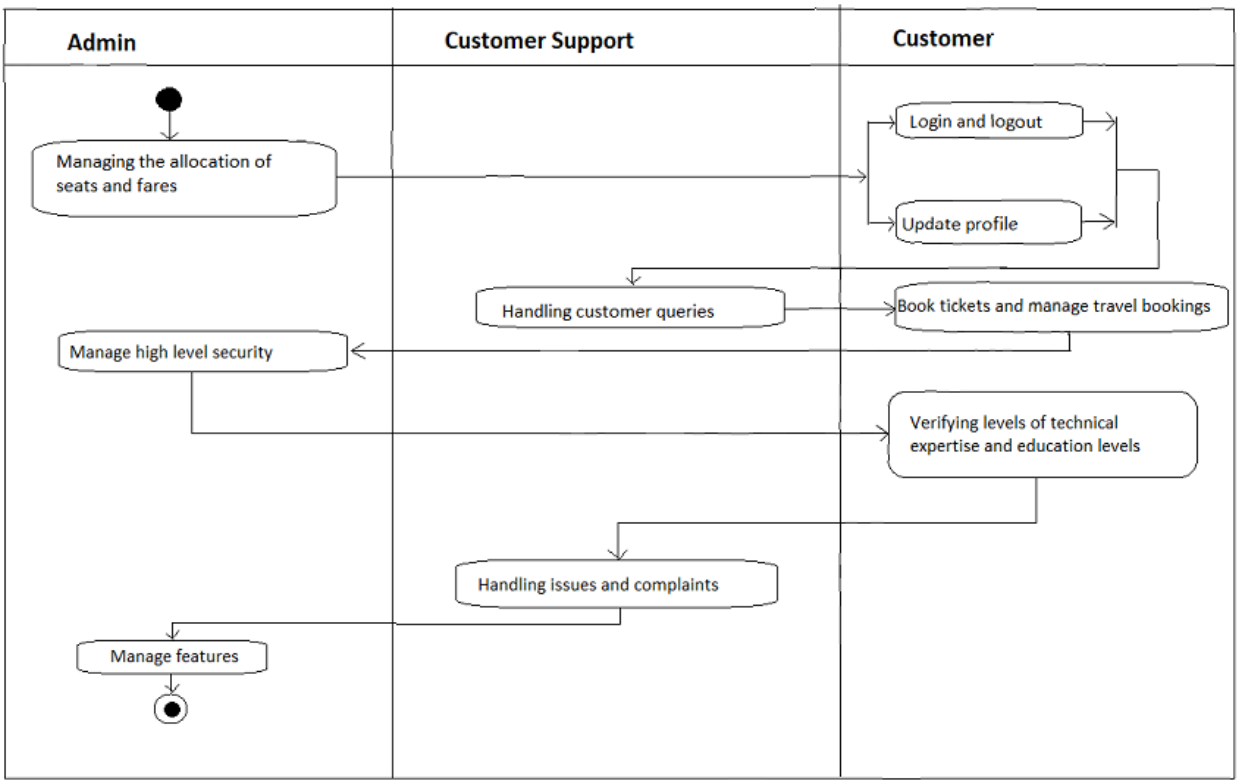

Figure 3: Activity diagram

4.1.4 ER diagram:

This diagram gives the glamps on entity relation of this software. Firstly, here is an Admin which is an entity. It has some attributes like id, username and password. Admin manages transport entity, which is consist of destination, arrival time, seat number, available seat, total seat, departure location and departure time attributes. Transports get reservations whit attributes of ticket number, ticket price, final price. These reservations are booked from passengers. Passengers has name, passengerID, contact, address, gender, age attributes. Passenger give feedbacks which is consist of name, fid, email, message. These feedbacks are handled by the customer support and viewed by admin.

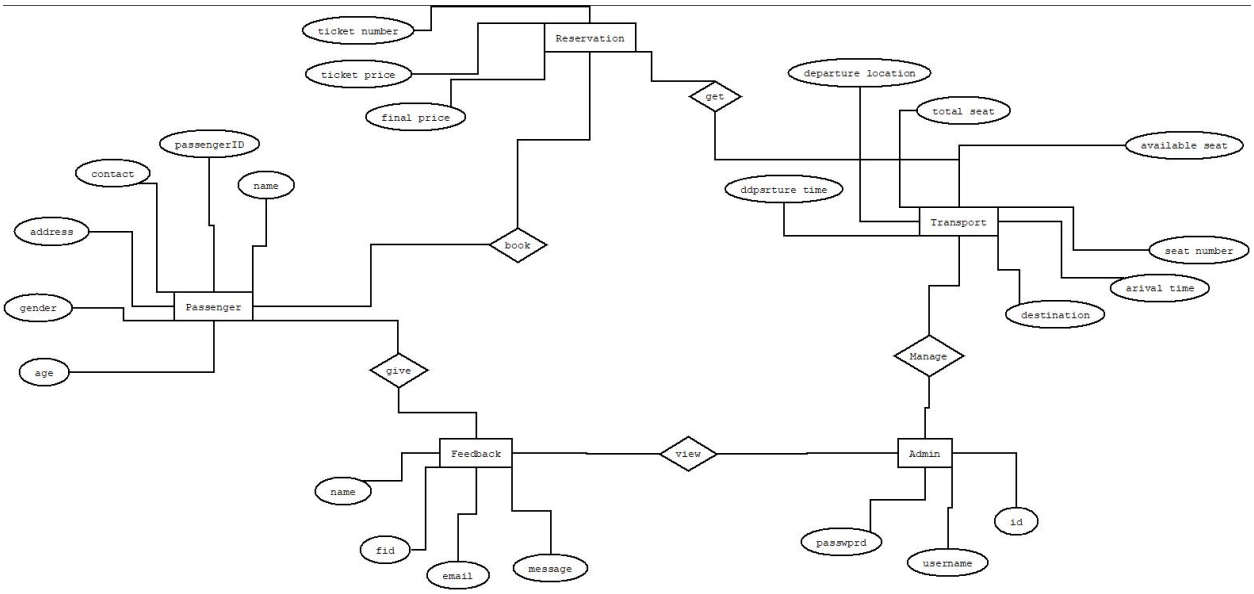

Figure 4: ER diagram

4.2 Data Dictionary

Entity	Attribute	Type/Size	Validation	Key
Passenger	P_name	String	Required	Primary
Passenger	email	String	Required	
Passenger	phone	String	Required	
Passenger	P_address	String	Required	
Vehicle	V_id	int	Required	Primary
Vehicle	type	String	Required	
Vehicle	seats	int	Required	
Vehicle	V_driver	Driver	Required	
Customer Support	C_name	String	Required	
Customer Support	C_id	int	Required	Primary
Customer Support	role	String	Required	
Customer Support	C_password	String	Valid password	
Driver	D_name	String	Required	
Driver	D_id	int	Required	Primary

Driver	vehicle	Vehicle	Required	
Driver	schedule	Schedule	Required	
Admin	A_name	String	Required	
Admin	A_id	int	Required	Primary
Admin	role	String	Required	
Admin	A_password	String	Valid password	

4.3 UI/UX Design Specification

The Figma tool was used to develop the prototypes of this project such as User Interface (UI) and User Experience (UX).

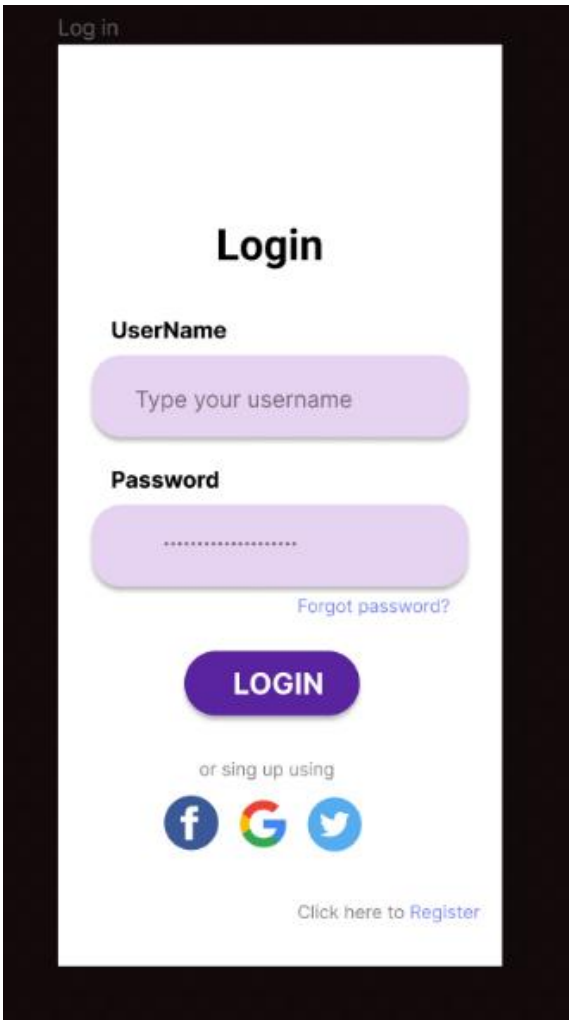

Figure 5: UI for login

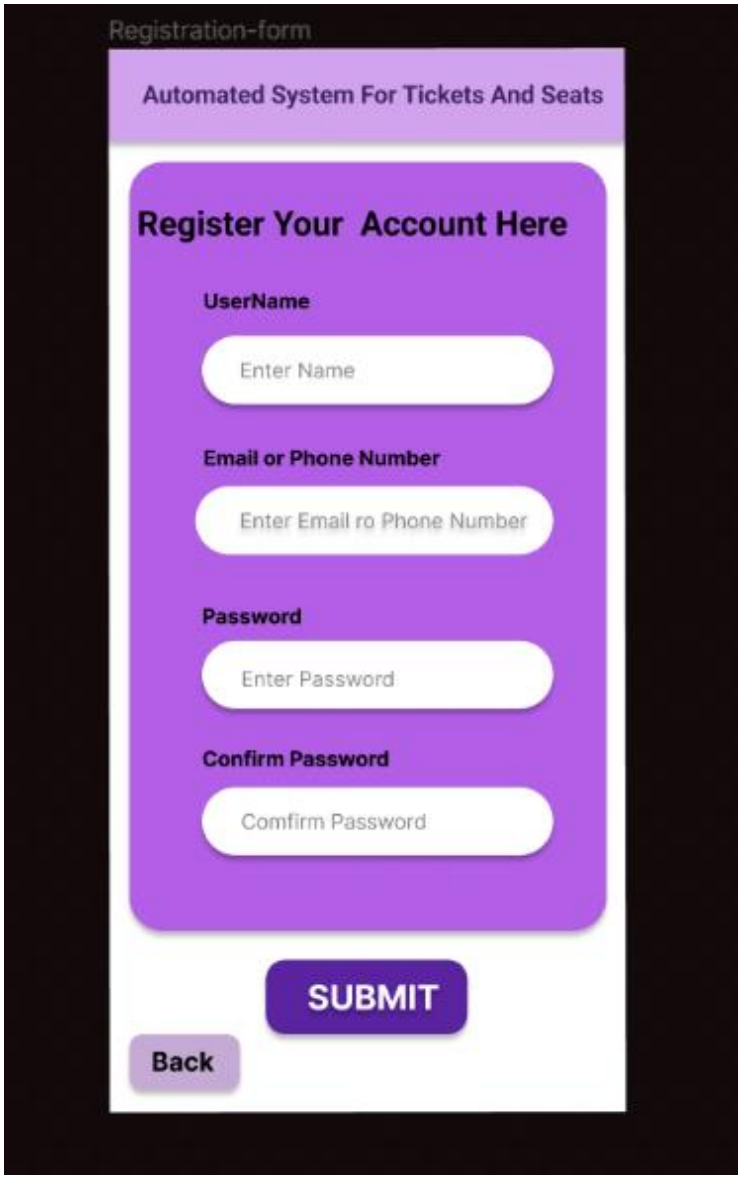

Figure 6: Passenger registration

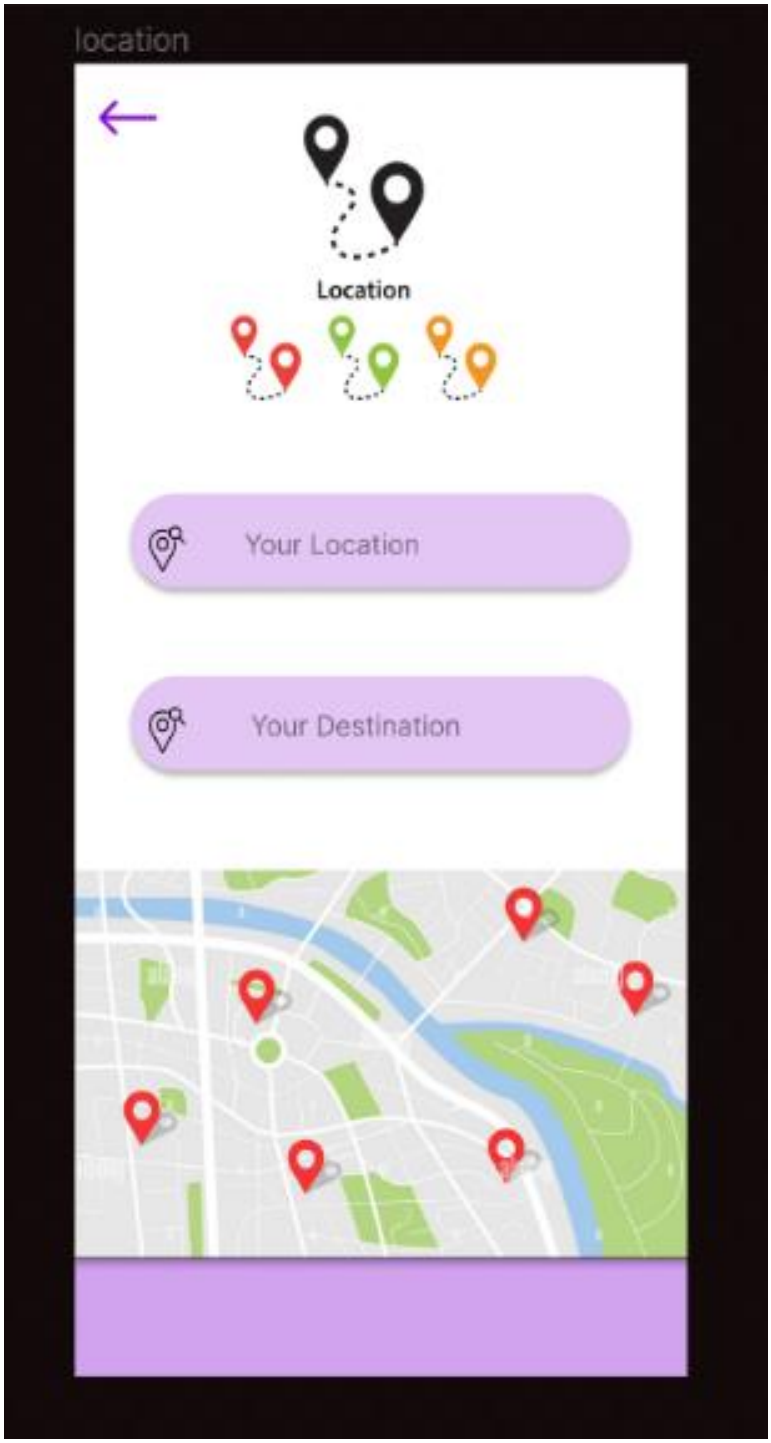

Figure 7: Finding vehicle

Figure 8: Vehicle suggestion and detailed information

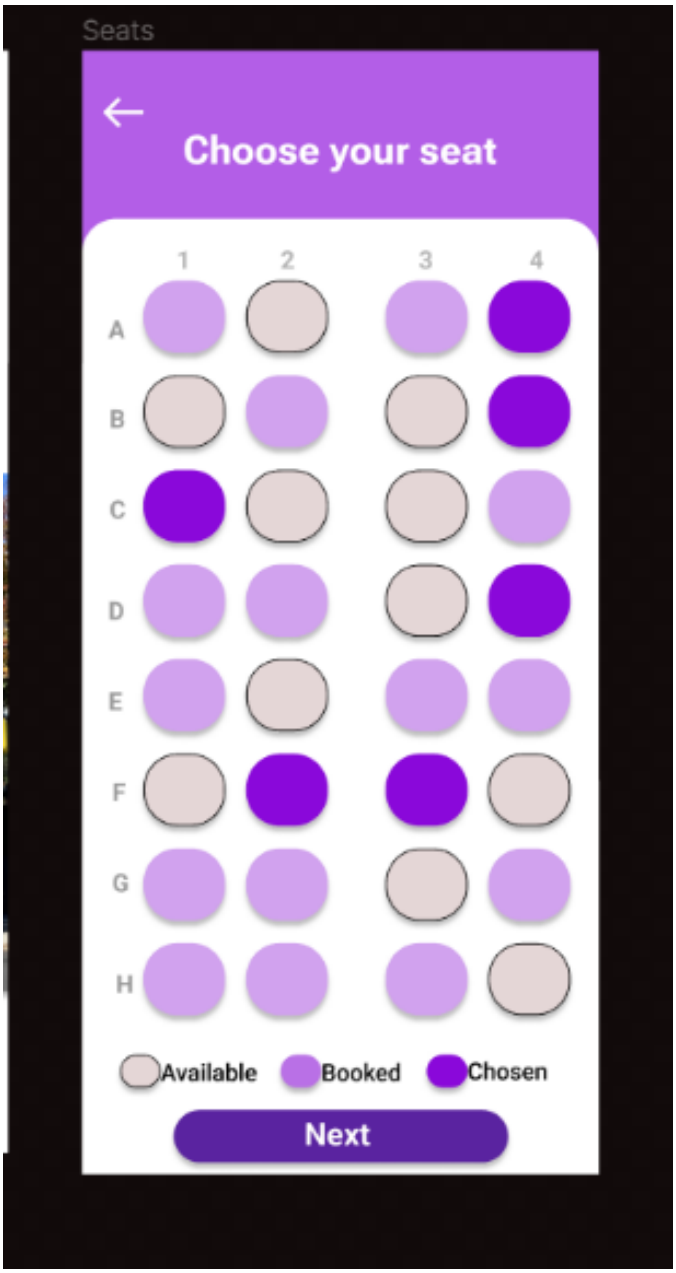

Figure 9: Choosing seats

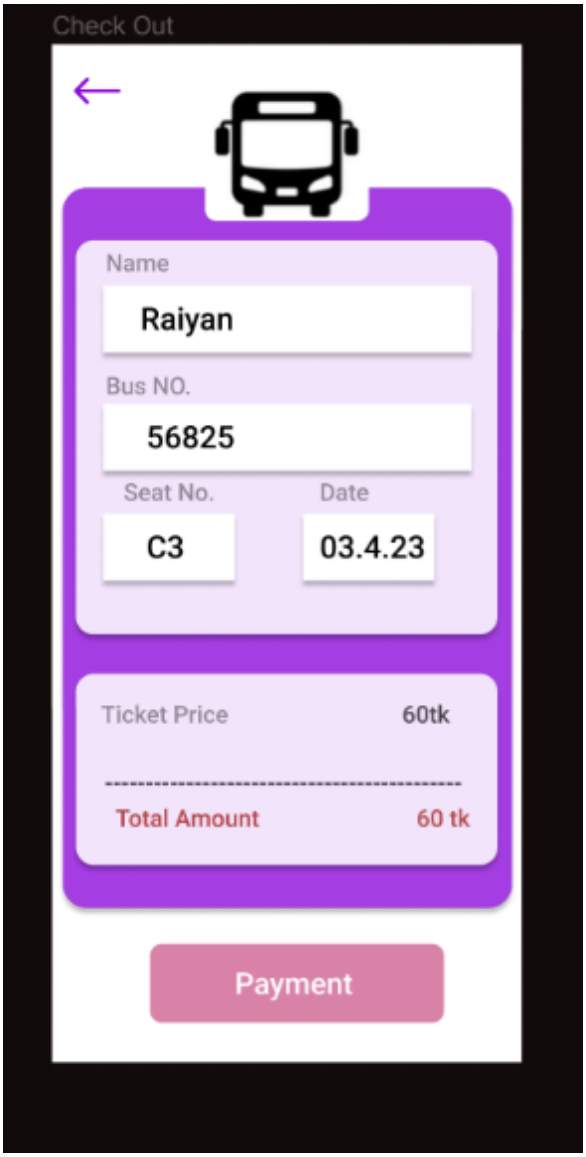

Figure 10: Providing fee

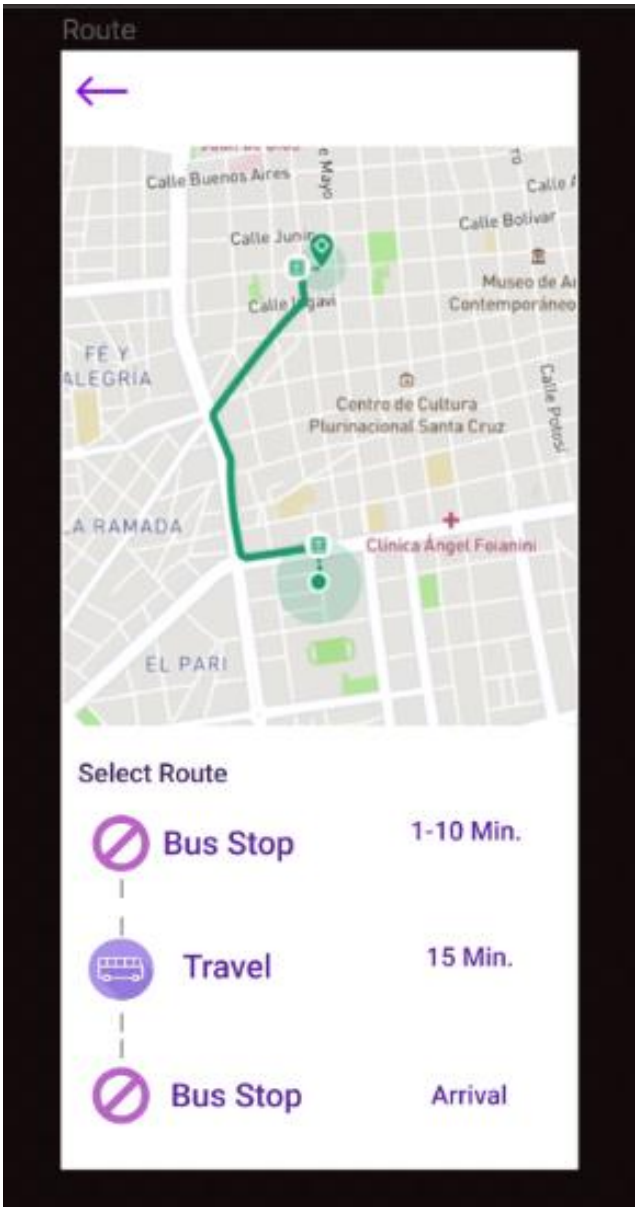

Figure 11: Ride summery