



FPT UNIVERSITY

SHARED TAXI BOOKING SYSTEM FOR FPT UNIVERSITY HCM STUDENTS

Software Requirement Specification

Project Code: 19

Document Code: STBS – v 1.0

Class Code: SE1857

Group Code: Group 7

Ho Chi Minh, October 2024

RECORD OF CHANGE

*A - Added M - Modified D - Deleted

Effective Date	Changed Items	A* M, D	Change Description	New Version
21/10/2024	Initial	A	Add project overview	1.0
24/10/2024		M	Modify functional requirement	1.0

SIGNATURE PAGE

ORIGINATOR: Dam Manh Dung
Dao Trong Tien
Huynh Nhat Truong
Pham Thi Thanh Ngan

REVIEWERS: <Name>____ <Date>____
<Position>

<Name, if it's needed>____ <Date>____
<Position>

APPROVAL: <Name>____ <Date>____
<Position>

TABLE OF CONTENTS

1 INTRODUCTION	5
1.1 Purpose	5
1.2 Scope	5
1.3 Definitions, Acronyms, and Abbreviations	5
1.4 References	6
1.5 Overview	6
2 OVERALL DESCRIPTION	8
2.1 Product Perspective	8
2.2 User classes and characteristics	9
2.3 Design and implementation constraints	10
3 FUNCTIONAL REQUIREMENTS	11
3.1 Swimlane Diagrams	11
3.2 Use Case Diagrams	12
3.2.1. Use Case: Create trip	13
3.2.2. Use Case: Deposit wallet	15
3.3 State Diagrams	18
3.4 Logical Data Model	19
3.5 Wire Flows	22
4 NON-FUNCTIONAL REQUIREMENTS	23
4.1 Usability	23
4.2 Reliability	23
4.3 Performance	23
4.4 Supportability	23
4.5 Licensing Requirements	24
5 SUPPORTING INFORMATION	25

1 INTRODUCTION

1.1 Purpose

This Software Requirements Specification (SRS) document for the Shared taxi booking system describes all functional and non-functional requirements of the system, which provides transportation services for FPT University students in Ho Chi Minh City. This document is designed for the project team members responsible for implementing and verifying the system's proper functionality. Unless otherwise noted, all requirements specified here are committed for release 1.0.

1.2 Scope

This document covers a detailed description of the features and requirements of the Shared taxi booking system. The system allows students to create or join shared trips, manage their bookings, make payments via electronic wallets, and receive support. It also facilitates user account management, student support during the booking process, handling complaints, and monitoring trip activities for both staff and administrators.

1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirements Specification.

Green Taxi: A platform providing electric taxi booking services for FPT University students in Ho Chi Minh City

Student: Users who can book trips and pay via electronic wallets

Staff: Personnel who assist students during the service process and handle complaints

Admin: Administrators responsible for managing the system and coordinating activities related to accounts and trips

1.4 References

Karl E. Wiegers, 2013, Software Requirements, Third Edition, Microsoft Press

1.5 Overview

1. Introduction

This section lays the foundation for the SRS document, outlining its purpose, scope, and the definitions, acronyms, and abbreviations used throughout. It also includes references for further reading and an overview to give readers a snapshot of the document's contents.

2. Overall Description

Provides a high-level overview of the system, including its general capabilities, user interactions, constraints, and assumptions. This section sets the context for the detailed requirements that follow.

3. Functional Requirements

Describes and defines the necessary functions for the user and outlines the processes required for the system to operate according to its objectives, effectively supporting the development, testing, and deployment process.

4. Non functional Requirements

Specifies the criteria not directly related to the functionality of the system but critical for its effectiveness and user satisfaction. This includes usability, reliability, performance, supportability. Each sub-section outlines specific requirements that ensure the system's quality and operational integrity.

5. Supporting information

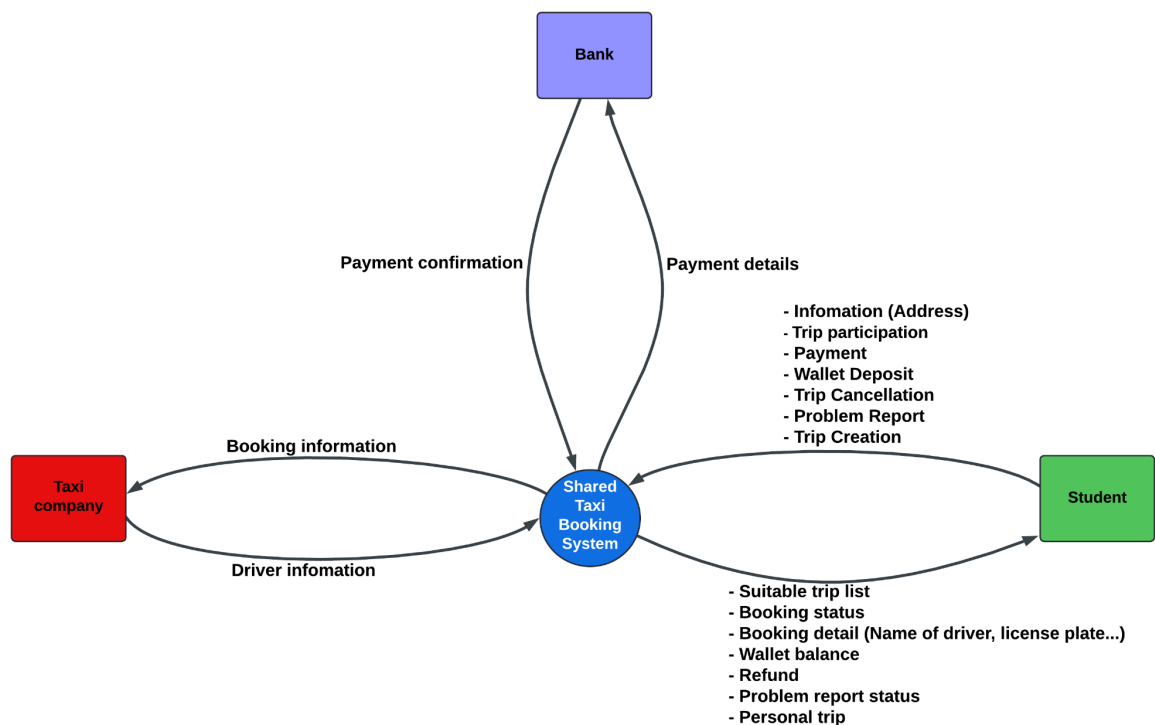
Additional information that supports the understanding, development, and implementation of the system. This could include data models, additional

requirements, guidelines for developers, and any supplementary documents or references needed for a comprehensive understanding of the system.

2 OVERALL DESCRIPTION

2.1 Product Perspective

A large number of FPT University HCM students live at Vinhomes, and commuting daily to school by personal motorbike poses several challenges, such as high fuel costs and the time-consuming task of finding parking spaces. To address these issues, the Shared Taxi Booking System has been developed, offering a convenient, cost-effective transportation solution for students. Through this system, students can easily book electric taxis to travel between Vinhomes and FPT University HCM or the Student Cultural House, making their daily commute more efficient and sustainable.



Context diagram of the system

2.2 *User classes and characteristics*

Admin

- **Characteristics:**
 - Highest administrative rights in the system and monitors the platform to ensure good operation
 - Resolve problems that occur with users, staffs, and systems.
- **Interactions:**
 - Manage account, disable account, and authorization
 - Manage User's Wallet, transaction
 - Solve Issues from Users and Staff
 - Generate Reports

Staff

- **Characteristics:**
 - Support users with issues related to booking a car
 - Processing refund requests for users
 - Monitor and report to admin about system errors that occur
- **Interactions:**
 - Handle trips refund
 - Live chat, phone and mail
 - Send system error notification to admin

Student

- **Characteristics:**

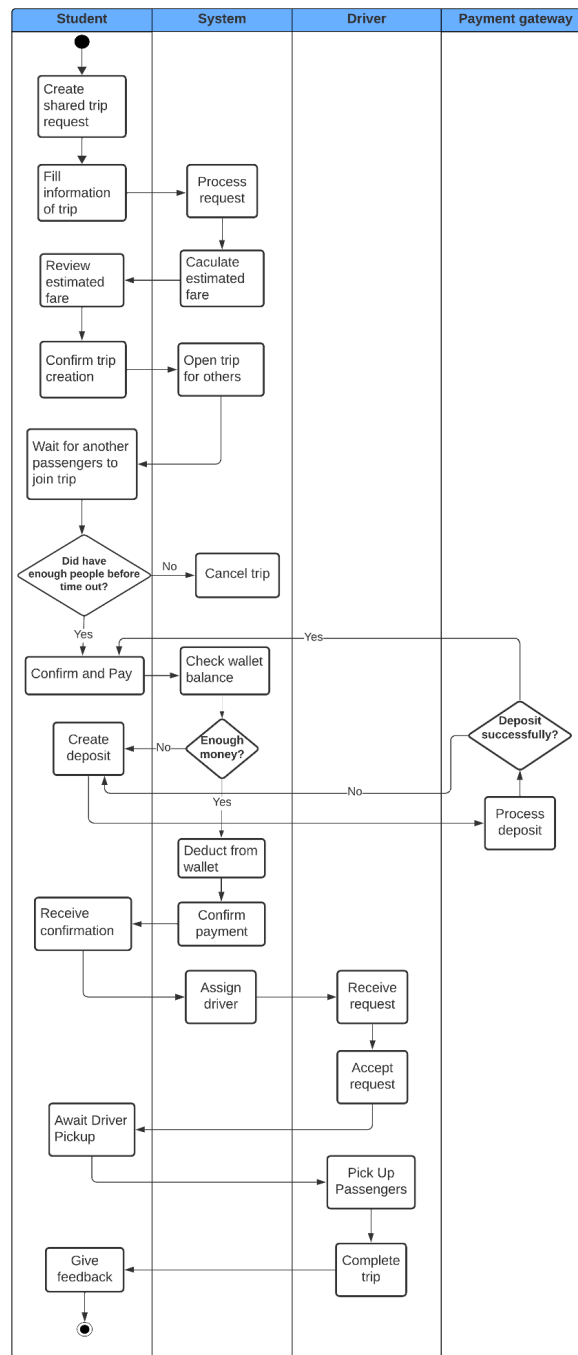
Desire to share car booking with many people nearby to save time, money and ensure safety in traveling
- **Interactions:**
 - Create trip, join available trip
 - Feedback driver and carpool
 - Ask question for Staff, report problems to Admin

2.3 Design and implementation constraints

- Use specific technologies, such as .NET 8.0, ReactJS Framework and SQL Server, and avoid incompatible platforms.
- The system should interface with external systems, such as student databases and payment services, using standardized data interchange formats like JSON.
- This system works best on Chrome and Edge browsers, other browsers may not work properly.

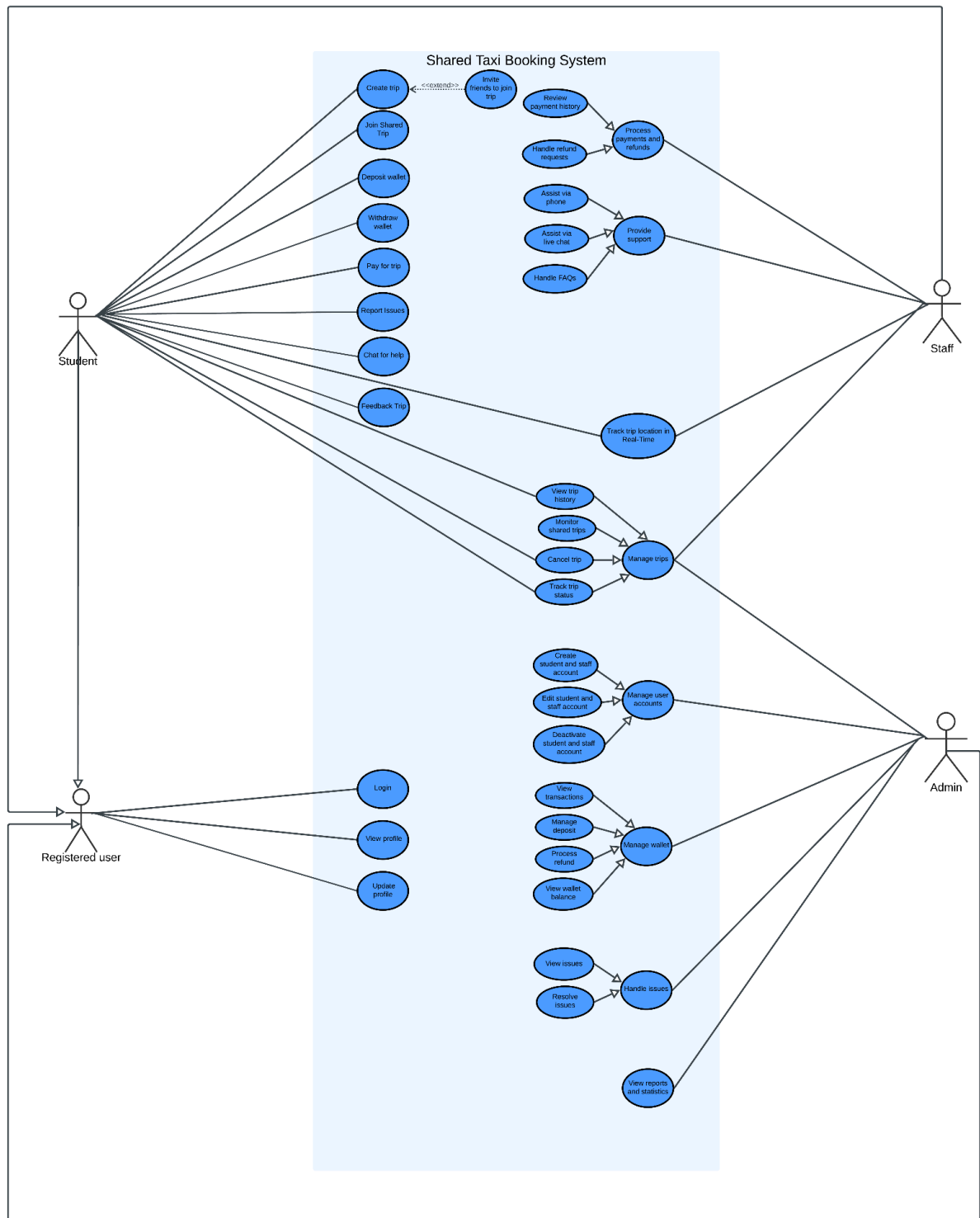
3 FUNCTIONAL REQUIREMENTS

3.1 Swimlane Diagrams



Swimlane diagram

3.2 Use Case Diagrams



Use case diagram

3.2.1. Use Case: Create trip

USE CASE-1 SPECIFICATION			
Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Create trip		
Author	Dam Manh Dung		
Date	23/10/2024	Priority	High
<p>Actor:</p> <p>Student</p> <p>Summary:</p> <p>This feature allows students to create a new trip.</p> <p>Goal:</p> <p>Allows students to create new trips with the required information, from which the system will save and display the trip in the list of trips.</p> <p>Triggers</p> <p>Students click on the "Create trip" button on the application interface.</p> <p>Preconditions:</p> <p>Students have logged into the system.</p> <p>Post Conditions:</p> <p>Success: Redirect to the waiting page.</p> <p>Fail: The system shows an error message.</p> <p>Main Success Scenario:</p>			

1. In the home page, student clicks on the “create trip” button, the system redirects to create a new trip page.
2. Student chooses where the starting location, ending location, start time, the number of people.
3. Student clicks the submit button. The system displays a form to confirm information.
4. Students click on the “create trip” button, the system redirects to the waiting page.

Alternative Scenario:

1. In the home page student clicks on the “create trip” button, the system redirects to create a new trip page.
2. Students choose the starting location, number of people, and do not choose ending location, start time. The system will display a message asking to enter the ending location and start time.
3. Students enter the ending location and start time, The system displays a form to confirm information.
4. Actor clicks on the “create trip” button, the system redirects to the waiting page.

Exceptions:

1. The start time is before the current time, the system will display an error message and ask students to re-enter.
2. Number of people exceeds the allowed number, the system will display an error message and ask students to re-enter.

Relationships:

Extends: Users can invite other members to join the trip after it has been successfully created.

Business Rules:

1. Students can only create new trips if they are logged in to the system.
2. All required fields must be filled in completely and accurately.

3.2.2. Use Case: Deposit wallet

USE CASE-2 SPECIFICATION			
Use-case No.	UC002	Use-case Version	1.0
Use-case Name	Deposit wallet		
Author	Dam Manh Dung		
Date	23/10/2024	Priority	High

Actor:

Student

Summary:

This feature allows the students to deposit money into the wallet to pay for the trip.

Goal:

Allows students to deposit money into the system wallets for use in transactions within the system.

Triggers:

Students click on the "deposit" button on the home page.

Preconditions:

1. Students are logged into the system.
2. Students have an e-wallet account linked or bank account linked.

Post Conditions:

Success: show message that deposit was successful, system redirect to home page.

Fail: show message deposit failed, system redirect to deposit page.

Main Success Scenario:

1. In the homepage, the student clicks on the deposit button, the system redirects to the deposit page.
2. Student enters the amount wants to deposit and chooses a payment method, the system show model confirms information.
3. Students click the "Ok" button, the system redirects to the enter OTP page.
4. Students enter the otp code and click the "Ok" button, the system displays a message "deposit successfully".
5. Students click "Ok", the system redirects to the home page.

Alternative Scenario:

1. In the homepage, the student clicks on the deposit button, the system redirects to the deposit page.
2. Student enters the amount he wants to deposit and chooses a payment method. The system displays a message that the balance is insufficient to pay.
3. Student clicks to try another method, the system redirects to the deposit page.
4. Students choose another method, and the system show model confirms information.
5. Students click the ok button, the system redirects to the enter OTP page.
6. Students enter the otp code and click the “Ok” button, the system displays a message “deposit money successfully”.
7. Students click “Ok”, the system redirects to the home page.

Exceptions:

1. E-wallet and bank account do not have enough money, the system displays a message of failed deposit.
2. E-wallet and bank account not linked,the system displays a message that you have not linked your e-wallet account.
3. Incorrect OTP, System inform “OTP is incorrect”.
4. OTP number is expired, System shows error message “Expired OTP”.

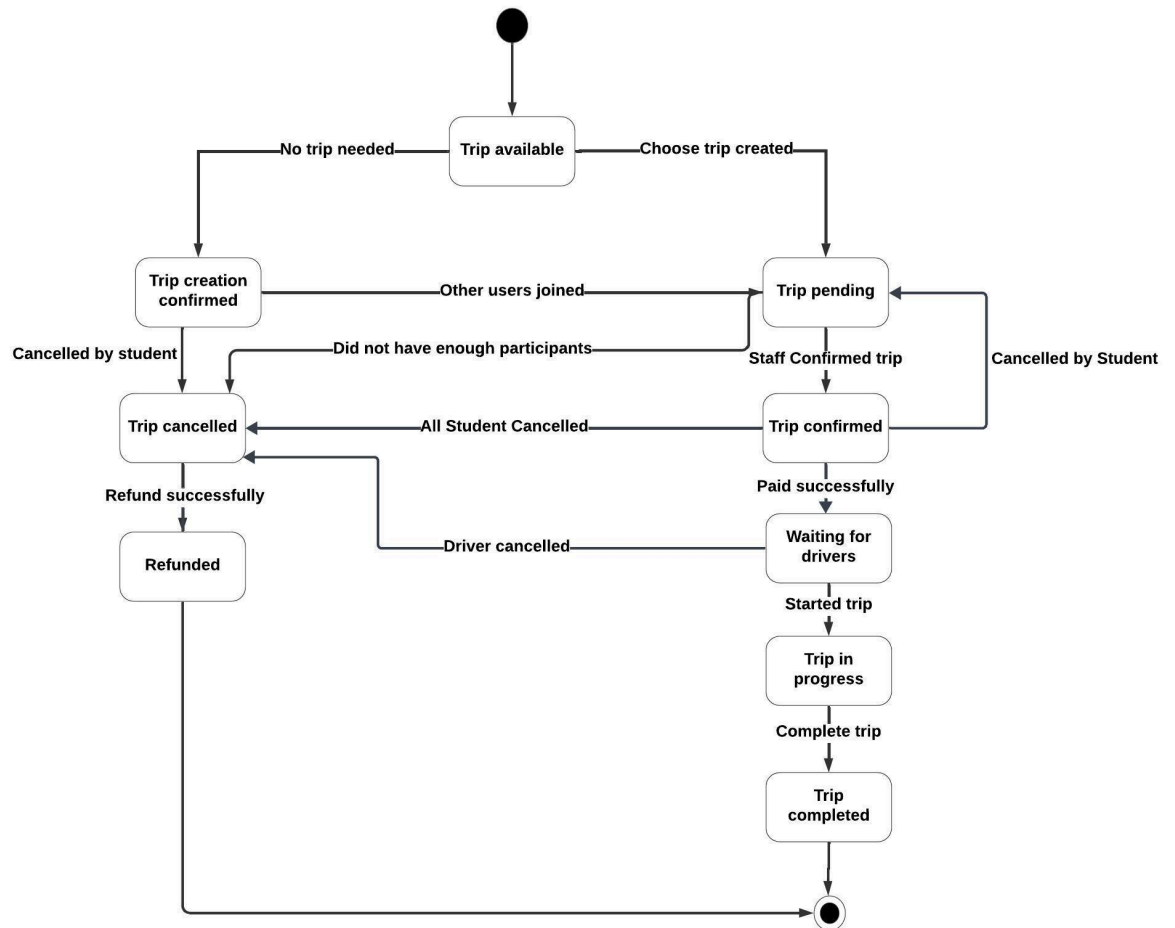
Relationships:

N/A

Business Rules:

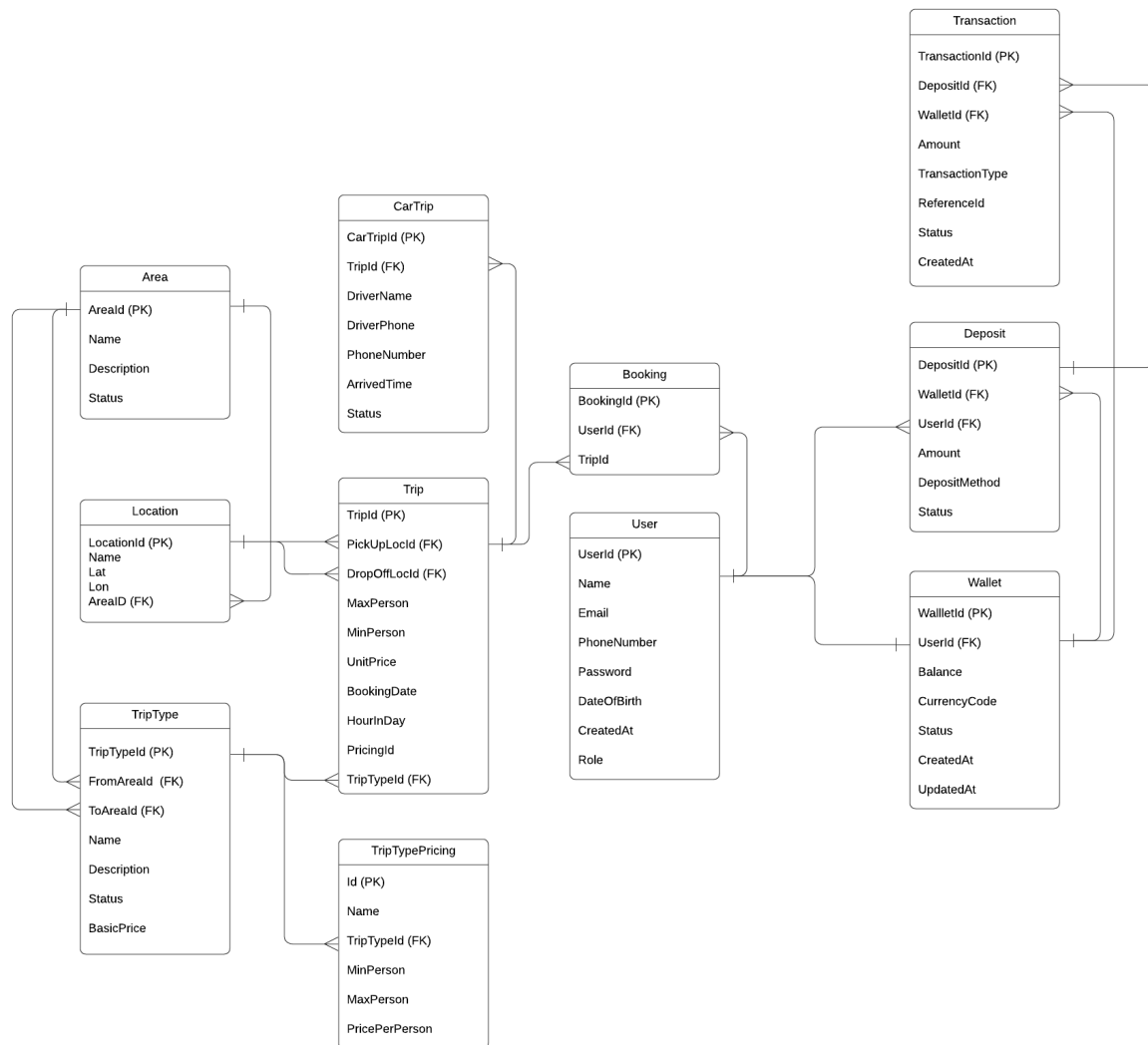
1. E-wallet and bank accounts must be linked to the taxi booking website.
2. The balance in your e-wallet or bank account must be greater than or equal to the amount you want to deposit.

3.3 State Diagrams



[State diagram](#)

3.4 Logical Data Model

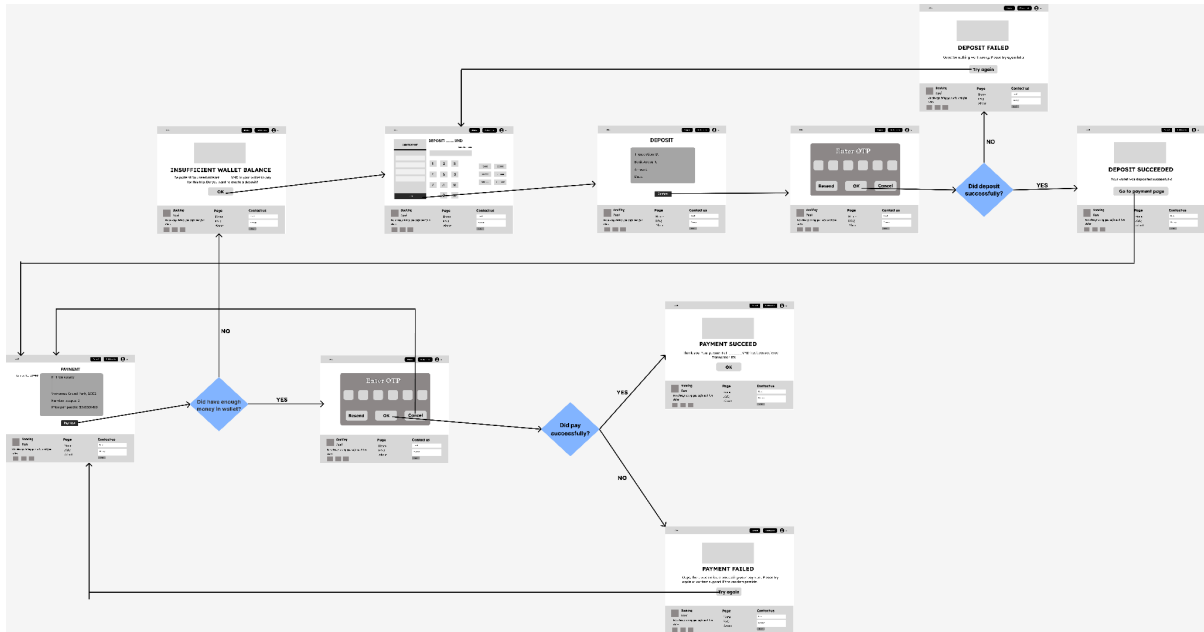


Logical data model

Entity	Relationship
User	<ul style="list-style-type: none">- Linked to Booking via UserId (FK)- Linked to Wallet via UserId (FK)- Linked to Deposit via UserId (FK).
Trip	<ul style="list-style-type: none">- Linked to Location for both pickup and dropoff locations via foreign key- Linked to TripType, Booking and CarTrip entities via TripTypeId (FK)
TripType	<ul style="list-style-type: none">- Linked to Area via FromAreaId and ToAreaId.- Connected to Trip as a type for multiple trips.- Connected to TripTypePricing for pricing details.
TripTypePricing	<ul style="list-style-type: none">- Linked to TripType via TripTypeId (FK).
CarTrip	<ul style="list-style-type: none">- Linked to Trip via TripId (FK).
Booking	<ul style="list-style-type: none">- Linked to Trip via TripId (FK).- Linked to User via UserId (FK).
Transaction	<ul style="list-style-type: none">- Linked to Deposit via DepositId (FK).- Linked to Wallet via WalletId (FK).

Wallet	<ul style="list-style-type: none">- Linked to User via UserId (FK).- Connected to Deposit and Transaction entities.
Deposit	<ul style="list-style-type: none">- Linked to Wallet via WalletId (FK).- Linked to Transaction via DepositId (FK).- Linked to User via UserId (FK).
Area	<ul style="list-style-type: none">- Related to Location (one-to-many) via AreaId (FK).
Location	<ul style="list-style-type: none">- Connected to Area via AreaId.- Related to Trip as PickupLocId and DropOffLocId (FK).

3.5 Wire Flows



Wireflow about task flow

4 NON-FUNCTIONAL REQUIREMENTS

4.1 Usability

- The system will be built using a user-centered approach following established usability principles and standards such as Material Design
- The system should provide a brief tutorial or step-by-step guidance to ensure students can successfully book or join a shared taxi within 5 minutes of their first use.
- Staff and admin should become proficient after two days of training.

4.2 Reliability

- The system must be available over 95% of the time, with maintenance outside peak hours.
- The system should have a Mean Time Between Failures of at least 1 year.
- The system should recover from failures within 4 hours.
- Payment calculations and wallet balances must be accurate within 0.5%.
- No more than 1 bug per 1000 lines of code. Critical bugs must be resolved within 2 business days.

4.3 Performance

- The system should handle ride bookings with an average response time of 1 seconds, max 5 seconds.
- The system must support up to 500 transactions per second.
- The system should accommodate up to 5,000 concurrent users.
- In high-load scenarios, essential features like booking and payments must remain operational.
- Memory usage should not exceed 80% under normal conditions.

4.4 Supportability

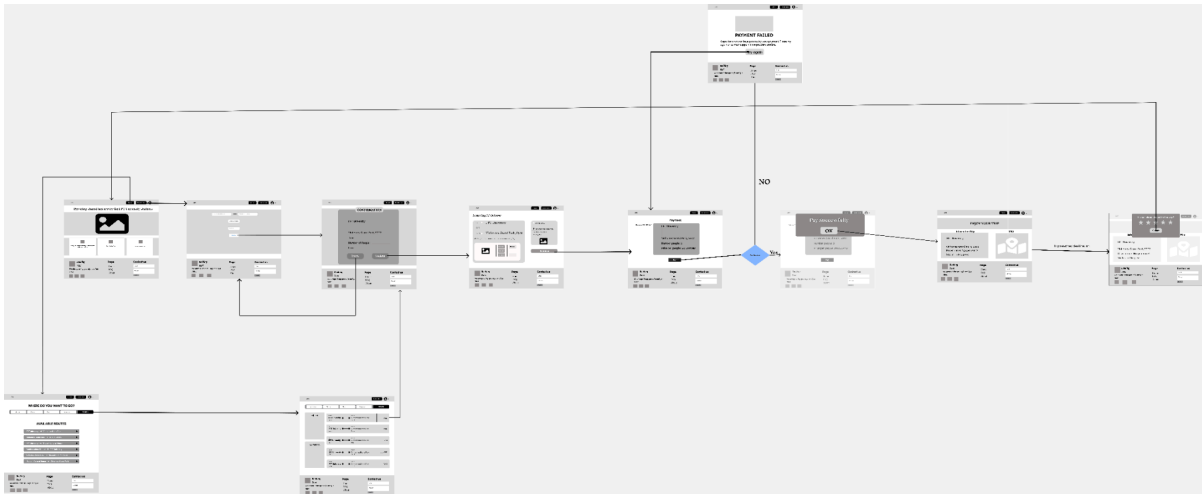
Ensure consistent code formatting, use meaningful variable names, and provide clear documentation for all code modules.

4.5 Licensing Requirements

- **User License Agreement**
 - Users must agree to the terms of service before registering an account
 - Each account can only be registered for a single user
- **Usage Restrictions**
 - Service operates with only Vinhome Grand Park, FPT University and The Student's Cultural House areas
 - Account sharing between multiple users is not permitted
 - Using the service for illegal activities is prohibited
- **Data Privacy & Security**
 - Comply with personal data protection regulations
 - Users have the right to request deletion of personal data
 - Trip data is stored in accordance with legal regulations
 - User information will not be shared with third parties without permission
- **Service Limitations**
 - Number of passengers must not exceed vehicle capacity
 - Time limit for passenger pickup
 - Regulations on trip cancellation and penalty fees

5 SUPPORTING INFORMATION

Appendix



[Wireflow about main flow](#)

Glossary

OTP: One Time Password

FK: Foreign Key

PK: Primary Key

JSON: JavaScript Object Notation.