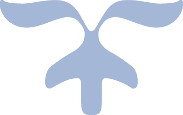


3ala el Taree2

***Mobile App***



**Supervised By : Dr. Manal Shabaan Teaching Assistant : Eng. Rabee Ayman**

**Team Members :**

|  |  |
| --- | --- |
| **Student Name** | **ID** |
| **Yasmin Hamdy Abd-alhamid** | **2102142** |
| **Omnia Mamdoh Mohamed** | **2101433** |
| **Arwa Yasser Abd Altawab** | **2100519** |
| **Kareem Saleh Mohamed** | **2102009** |
| **Ahmed Essam Mohamed** | **2101836** |
| **Ahmed Mostafa Mohamed** | **2100391** |
| **Abdulrahman Mohamed Rashad** | **2101956** |

**ABSTRACT**

Public transportation plays a crucial role in connecting people across cities, but traditional bus booking systems often suffer from inefficiencies, long queues, and a lack of real-time availability. To address these challenges, **3al Taree2** is developed as a **smart bus reservation system** that offers a seamless digital booking experience. The project focuses on optimizing seat reservations, providing real-time scheduling, and enhancing user convenience through an interactive and user-friendly mobile application.

In conventional bus ticketing systems, passengers must visit terminals in person to check for available seats or rely on outdated manual systems, which can lead to **inaccurate seat allocations, overbookings, and inefficiencies**. **3al Taree2** eliminates these issues by allowing users to search for available buses, select their preferred seats, and confirm bookings instantly through a digital interface. The platform also enables passengers to receive electronic tickets, track their booking history, and manage their reservations with ease.

From the operational perspective, **3al Taree2** enhances the efficiency of bus operators by providing a **centralized system** for managing routes, monitoring seat availability, and streamlining passenger data. This not only improves resource allocation but also helps transport companies reduce revenue losses caused by unoccupied seats and scheduling conflicts.

The system is built using **Flutter** for the frontend, ensuring a **cross-platform and responsive user experience**, allowing both Android and iOS users to access the application seamlessly. The **backend is powered by ASP.NET Core Web API**, providing a **scalable and secure** environment for handling transactions, processing bookings, and authenticating users. **SQL Server** is used as the **database management system**, ensuring the safe storage and retrieval of passenger information, bookings, and route details. The platform is hosted on **Somee.com**, offering a reliable infrastructure for smooth operation and accessibility.

One of the core innovations of **3ala el Taree2** is its **seat selection feature**, where users can view an interactive seating map and choose their preferred spot before confirming their booking. The system also incorporates a **driver registration module**, allowing verified drivers to register and manage their schedules.

With the growing demand for digitized transportation solutions, **3ala el Taree2** aims to **bridge the gap between traditional and modern bus booking systems** by offering a **convenient, efficient, and transparent platform**. By automating bus reservations, reducing wait times, and providing real-time updates, **3al Taree2** sets a new standard for intercity travel, ultimately **enhancing the overall experience for passengers and operators alike**.

This project represents a **significant step toward digital transformation** in the transportation sector, leveraging cutting-edge technologies to improve service delivery, passenger satisfaction, and operational efficiency.

**ACKNOWLEDGEMENT**

We take this opportunity to express our **deepest gratitude and sincere**

**appreciation** to our esteemed **Supervisor, Dr. [Manal shabaan]**, for his **invaluable guidance, continuous support, and unwavering encouragement** throughout the development of this project. His insightful feedback, constructive criticism, and expertise have played a crucial role in shaping our work and ensuring its successful completion.

We would also like to extend our **heartfelt thanks** to the **Faculty of Information Technology** and its **Board Members** for their continuous support, valuable resources, and for providing us with a learning environment that fostered innovation and creativity. Their dedication to academic excellence has been instrumental in our journey.

Furthermore, we express our **profound appreciation** to our **team members**, whose **hard work, dedication, and collaborative efforts** have been the driving force behind this project. Each contribution, big or small, has been essential in bringing our vision to life.

A special thank you goes to our **families and friends** for their unwavering support, patience, and encouragement throughout this journey. Their belief in us has been a source of motivation and inspiration.

Lastly, we extend our gratitude to the **Almighty**, who has blessed us with the strength, perseverance, and opportunity to embark on this remarkable endeavor.

This project marks the successful completion of the first phase of our graduation journey, and we look forward to continuing our efforts toward its full realization.

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| **Content** |  |
| **Abstract** |  |
| **Acknowledgment** |  |
| **Chapter 1: Introduction** | **8** |
| **1.1 Introduction** | **9** |
| **1.2 Overview** | **10** |
| **1.3 Motivation** | **11** |
| **1.4 Objective of the system** | **12** |
| **1.5 Project Scope** | **13** |
| **1.6 Feasibility Study** | **15** |
| **1.7 Software process model** | **18** |
| **Chapter 2: Related Work** | **19** |
| **2.1 Introduction** | **20** |
| **2.2 Core features** | **21** |
| **2.3 Technology Stack** | **21** |
| **2.4 Objectives** | **22** |
| **Chapter 3: System Model** | **23** |
| **3.1 Introduction** | **24** |
| **3.2 Requirement Analysis** | **24** |
| **3.3 Gathering Requirements** | **24** |
| **3.4 System Architecture** | **38** |
| **3.5 Existing Similar Systems** | **41** |
| **Chapter 4:** **Proposed system** | **43** |
| **4.1 Introduction** | **44** |
| **4.2 Use-Case** | **45** |
| **4.3 Class Diagram** | **48** |
| **4.4 ERD Diagram** | **50** |
| **4.5 Schema Diagram** |  |
| **4.6 Sequence Diagram** | **53** |
| **4.7 Data Flow Diagram** |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Chapter 5: Chapter Implementation** | | **5.1Technologies, tools, and programming languages used.** | | **5.2 Key components/modules of the system.** | | **5.3 Challenges faced and how they were resolved.** | | **Chapter 6: testing & evaluation** | | **6.1Testing strategies (unit testing, integration testing, user testing).** | | **6.2Performance metrics (accuracy, speed, scalability).** | | **6.3 Comparison with existing solutions.** | | **Chapter 7****: Results & Discussion** | | **7.1 Introduction** | | **7.2 Summary of findings.** | | **7.3Interpretation of results (Did the project meet its objectives?).** | | **7.4 Limitations of the proposed solution.** | | **Chapter 8:** **Conclusion & Future Work** | | **8.1 Summary of contributions** | | **8.2 Possible improvements or extensions for future work** | | **Reference** | | **59** |
| **60** |
| **60** |
| **91** |
| **92** |
| **93** |
| **93** |
| **96** |
| **98** |
| **99** |
| **99** |
| **99** |
| **99** |
| **100** |
| **101** |
| **101** |
| **102** |
|  |
|  |
|  |

# CHAPTER 1 INTRODUCTION

## Introduction

## Public transportation plays a vital role in facilitating mobility for individuals and communities, yet many travelers face challenges in securing bus reservations efficiently. The conventional methods of purchasing bus tickets, which often require long queues and manual booking processes, can lead to inefficiencies, inconvenience, and mismanagement of resources. With the increasing demand for smart and automated transportation solutions, there is a pressing need for a system that streamlines the reservation process while optimizing seat utilization and scheduling.

## Recognizing these challenges, 3al Taree2 is developed as an innovative bus reservation system designed to enhance the efficiency of intercity travel. The system aims to automate the process of searching, booking, and managing bus tickets, providing passengers with a seamless experience while enabling transport operators to manage their fleets more effectively. A well-structured bus reservation system should allow passengers to book tickets, choose seating arrangements, and receive real-time updates about bus availability.

A **bus reservation system** consists of several key components that ensure the smooth operation of the platform:

* **B (Set of Bus Routes)** – Defines the network of available routes.
* **S (Set of Schedules)** – Specifies departure times, arrival times, and frequency.
* **P (Set of Passengers)** – Represents users booking tickets.
* **C (Set of Constraints)** – Includes seat availability, scheduling conflicts, and operational capacity.

The process of managing bus reservations involves optimizing seating capacity while ensuring that **ticketing, seat selection, and scheduling constraints** are properly maintained. Transport companies often face **difficulties in handling last-minute bookings, cancellations, and overbookings**, which can lead to inefficiencies in bus operations.

By utilizing **modern technology and automation**, **3al Taree2** introduces a **systematic approach** to bus reservations. The platform leverages a **real-time database, automated seat selection, and an intuitive user interface** to provide a **user-friendly, secure, and efficient** ticketing experience. Built with **Flutter for the frontend**, **ASP.NET Core Web API for the backend**, and **SQL Server for data management**, this project ensures that passengers and operators benefit from a **robust, scalable, and streamlined solution**.

In this chapter, we will discuss the development and implementation of **3al Taree2**, highlighting its **functionalities, technical architecture, and problem-solving approach** in revolutionizing the bus reservation system.

## Overview

**Managing bus reservations manually can be a complex and time-consuming task, often leading to inefficiencies, scheduling conflicts, and overbookings. The traditional approach relies on manual ticketing systems, which are prone to errors and lack real-time availability updates. Additionally, managing cancellations, rebookings, and seat availability manually creates logistical challenges for transportation companies. To address these issues, an automated bus reservation system is necessary to improve efficiency, accuracy, and user experience.**

**3al Taree2** is a **web and mobile-based bus reservation system** designed to streamline the booking process and enhance transportation management. This system provides passengers with **real-time bus schedules, seat availability, and instant booking confirmations**, reducing the time and effort required for ticket reservations. The system also allows bus operators to **manage routes, schedules, and seat allocations efficiently**, ensuring optimal resource utilization.

The proposed system will be accessible by various stakeholders, including passengers, bus operators, and administrators, with specific roles and permissions:

1. **Passengers** can search for available buses, select routes, book seats, and receive digital tickets instantly.
2. **Bus Operators** can add new routes, set schedules, and monitor seat availability in real-time.
3. **System Administrators** have full control over **managing routes, updating schedules, handling user accounts, and ensuring smooth operations**.
4. **Drivers** can view their assigned schedules and update trip statuses in real-time.

### ****Key Constraints Considered in 3al Taree2:****

1. **Bus routes and schedules** are managed only by the **Admin** to ensure accuracy.
2. **Seat bookings are subject to real-time availability** to prevent overbooking.
3. **Each bus has a predefined capacity**, and once fully booked, additional reservations are not allowed.
4. **Drivers' schedules** are assigned based on predefined work hours and route allocations.
5. **Passengers can modify or cancel bookings within a set timeframe**, ensuring flexibility while preventing last-minute disruptions.

The **3al Taree2 system** provides a **user-friendly, automated, and scalable** solution that ensures a **hassle-free travel booking experience**. By digitizing the process, the system **improves operational efficiency**, enhances customer satisfaction, and optimizes bus utilization, making transportation more accessible and reliable.

## Motivation

The transportation industry plays a crucial role in daily commuting and long-distance travel, yet **traditional bus reservation systems remain inefficient, time-consuming, and prone to errors**. Many passengers still face **difficulties in securing seats, dealing with overbookings, and accessing real-time schedule updates**, leading to frustration and inconvenience. Similarly, bus operators struggle with **manual seat allocation, route management, and handling last-minute changes**, which results in operational inefficiencies and revenue loss.

Recognizing these challenges, we were motivated to develop **3al Taree2**, an **automated bus reservation system** that leverages modern technology to **simplify and optimize** the ticket booking process. Our motivation stems from the following key factors:

1. **Eliminating Inefficiencies in Manual Booking**
   * Traditional booking systems require passengers to visit terminals in person, leading to **long queues, delays, and scheduling conflicts**.
   * Manual tracking of seat availability results in **errors, double bookings, and operational inefficiencies**.
2. **Enhancing Passenger Convenience**
   * A mobile and web-based reservation system allows users to **book seats, check schedules, and receive digital tickets instantly**.
   * Providing **real-time updates on seat availability** reduces uncertainty and enhances the overall travel experience.
3. **Optimizing Resource Utilization for Bus Operators**
   * Bus companies can **manage schedules, routes, and seat allocations in an organized manner**.
   * A centralized system **prevents last-minute seat shortages or wasted capacity**, maximizing revenue potential.
4. **Leveraging Technology for a Smarter Transportation System**
   * By incorporating **Flutter for the frontend and ASP.NET Core Web API for the backend**, we ensure a **scalable and efficient system** that operates smoothly across different platforms.
   * **SQL Server ensures secure and reliable data management**, preventing discrepancies in reservations.
5. **Addressing the Growing Need for Smart Mobility Solutions**
   * With **increased reliance on digital platforms**, there is a demand for **mobile-friendly, automated, and intelligent transport systems**.
   * A **real-time bus reservation platform** aligns with the shift toward **smart cities and digital transformation** in transportation.

## Objective of the system

The **3al Taree2** bus reservation system is designed to **enhance the efficiency and accessibility** of intercity travel by leveraging automation and modern technology. The system aims to optimize the booking process for passengers while improving operational efficiency for transport providers. Below are the **key objectives** that guide the development and implementation of this system:

### Time Optimization:

* + - * implementing a real-time booking system that allows users to instantly search, select, and confirm reservations without unnecessary delays.
      * Ensuring efficient scheduling of bus routes to minimize idle time, improve turnaround efficiency, and reduce gaps in seat availability.

### Automation:

* + - * Automating the entire booking process, from seat selection to payment confirmation, reducing the need for manual intervention.
      * Enhancing efficiency by automatically updating seat availability, sending digital tickets, and processing cancellations or modifications without human errors.

### Conflict Resolution:

* + - * Preventing double bookings and over-allocations through real-time seat tracking and a dynamic database system.
      * Ensuring schedule synchronization between passengers and bus operators, reducing cases of miscommunication and last-minute schedule conflicts.

### Flexibility:

### Allowing administrators to manage bus schedules, adjust seat availability, and modify trip details as needed.

### Providing passengers with options to modify or cancel bookings, ensuring a seamless experience while maintaining operational efficiency.

### Enabling transport companies to scale operations by easily adding new routes, buses, and services to the system

## Project Scope

The **3al Taree2 Bus Reservation System** is a **comprehensive digital solution** designed to automate and enhance the **bus ticket booking process** for passengers and operators. The system provides a **user-friendly, efficient, and scalable** platform for managing reservations, schedules, and seat availability while ensuring optimal **resource utilization**.

#### ****1. Functionalities of the System****

✔ **User Role-Based Access Control:**

* Passengers: **Search, book, and manage** their bus reservations.
* Bus Operators: **Manage routes, schedules, and monitor seat availability.**
* Administrators: **Oversee the system, manage users, and handle operational settings.**

✔ **Bus and Seat Management:**

* Add, update, or remove **bus routes, schedules, and ticket availability**.
* Implement **real-time seat tracking** to prevent overbookings.

✔ **Automated Booking & Scheduling:**

* A **fully functional booking engine** that ensures seamless **seat allocation** based on availability.
* Automated handling of **cancellations and rebookings**.

✔ **Real-Time Updates & Notifications:**

* Instant booking confirmation and **e-ticket generation**.
* **Automated alerts** for trip status, schedule changes, and cancellations.

✔ **Management Information & Reporting System:**

* Generate **reports based on routes, schedules, booking trends, and passenger data**.
* Track and analyze bus occupancy rates and **revenue insights**.

✔ **Flexible & Scalable System:**

* Support for **multiple bus operators** and expansion to **new routes and locations**.
* User-friendly **mobile and web-based interface** with potential **integration with third-party payment systems**.

#### ****2. Technology Stack****

* **Frontend:** Flutter (for mobile-friendly experience).
* **Backend:** ASP.NET Core Web API (for secure and scalable data handling).
* **Database:** SQL Server (for structured storage and efficient queries).
* **Hosting:** Somee.com (for accessibility and online operations).

#### 3. Project Goals

* **Enhance Efficiency:** Minimize **manual errors, booking delays, and scheduling conflicts**.
* **Optimize Resource Utilization:** Improve **bus occupancy rates** and reduce revenue losses.
* **Increase User Convenience:** Provide a **seamless, intuitive, and automated booking experience**.
* **Ensure Scalability:** Allow for **future expansion** in routes, services, and features.

**With a strong focus on automation, user experience, and operational efficiency, 3al Taree2 aims to redefine bus travel by modernizing the reservation process and improving accessibility for passengers and transport providers alike.**

## Feasibility Study of Timetable Management System

Before initiating the **3al Taree2 Bus Reservation System**, a **comprehensive feasibility study** was conducted to assess its **practicality, cost-effectiveness, and technical viability**. This study evaluates whether the proposed system can successfully address the challenges of **manual bus reservations**, improve the **efficiency of seat allocation**, and optimize the **public transportation experience** for both passengers and operators. The feasibility analysis covers the following key areas:

### ****1. Technical Feasibility****

The **technical feasibility** assesses whether the **available technology** and infrastructure can support the system’s development and functionality.

✔ **Technology Stack:**

* **Frontend:** Flutter (for a cross-platform, mobile-friendly interface).
* **Backend:** ASP.NET Core Web API (for secure, scalable, and efficient server-side operations).
* **Database:** SQL Server (for structured storage, fast queries, and data integrity).
* **Hosting:** Somee.com (for online accessibility and real-time availability).

✔ **System Capabilities:**

* The system is designed to **automate seat booking, route selection, and real-time schedule updates**.
* Integration with **payment gateways, notifications, and ticketing systems** is feasible.
* The platform can **scale efficiently**, supporting multiple routes, operators, and dynamic scheduling.

✔ **Conclusion:**  
The technology stack chosen is **widely used, reliable, and supports scalability**, making the system **technically feasible**.

### ****2. Economic Feasibility****

The **economic feasibility** examines whether the **project’s benefits justify its costs** and whether it will provide a **return on investment (ROI)**.

✔ **Cost Considerations:**

* **Development Costs:** Includes software development, database setup, and UI/UX design.
* **Hosting & Maintenance Costs:** Regular hosting, security updates, and performance monitoring.
* **Operational Costs:** Customer support, server maintenance, and bug fixes.

✔ **Expected Benefits:**

* **Increased Revenue for Bus Operators**: More efficient seat utilization and reduced manual effort.
* **Reduced Administrative Costs**: Automating the reservation process minimizes the need for manual intervention.
* **Improved Passenger Experience**: Easier and faster ticket booking, leading to increased user adoption.

✔ **Conclusion:**  
The system is **economically feasible** as the **initial investment is justified by long-term cost savings, revenue optimization, and passenger convenience**.

### ****3. Operational Feasibility****

Operational feasibility examines whether the system can be **effectively implemented and used by its intended audience**.

✔ **Usability & Accessibility:**

* The system offers an **intuitive UI/UX for passengers, bus operators, and administrators**.
* Mobile-friendly design ensures **accessibility from anywhere**.

✔ **Impact on Users:**

* **Passengers:** Convenient online booking, instant ticket generation, and real-time seat availability.
* **Bus Operators:** Automated scheduling, optimized seat allocation, and real-time data insights.
* **Administrators:** Easy monitoring of bookings, revenue tracking, and improved customer support.

✔ **Conclusion:**  
The system is **operationally feasible**, as it aligns with **user needs, streamlines bus operations, and enhances customer satisfaction**.

### ****4. Legal & Regulatory Feasibility****

The legal feasibility ensures that the system complies with **transportation, data privacy, and financial transaction laws**.

✔ **Compliance with Regulations:**

* The system adheres to **passenger data protection laws** to ensure secure transactions and data privacy.
* Compliance with **transport licensing regulations** ensures bus operators meet industry standards.
* Secure **payment processing** meets financial security protocols.

✔ **Conclusion:**  
The system is **legally feasible**, provided that necessary **data protection and transaction security measures** are in place.

### ****5. Schedule Feasibility****

Schedule feasibility evaluates whether the project can be **completed within a reasonable timeframe**.

✔ **Development Timeline:**

* **Phase 1:** System requirements gathering & UI/UX design (1-2 months).
* **Phase 2:** Backend and frontend development (3-4 months).
* **Phase 3:** Testing, debugging, and deployment (2 months).
* **Phase 4:** System launch and post-launch support (1-2 months).

✔ **Conclusion:**  
The project is **schedule-feasible**, as it can be developed and deployed **within a structured timeline** while allowing room for **enhancements and future scalability**.

## Software process model

The Waterfall Model stands as the inaugural process model in software development. It follows a sequential structure where each phase must reach completion before the subsequent phase commences, avoiding any overlap. This model divides the entire software development process into distinct phases, with the output of one phase serving as the input for the next one in a sequential manner. The progression through phases—Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation, and Maintenance— resembles a continuous flow akin to a waterfall. This linear and sequential progression characterizes the Waterfall Model, earning it the designation of a Linear-Sequential Life Cycle Model.

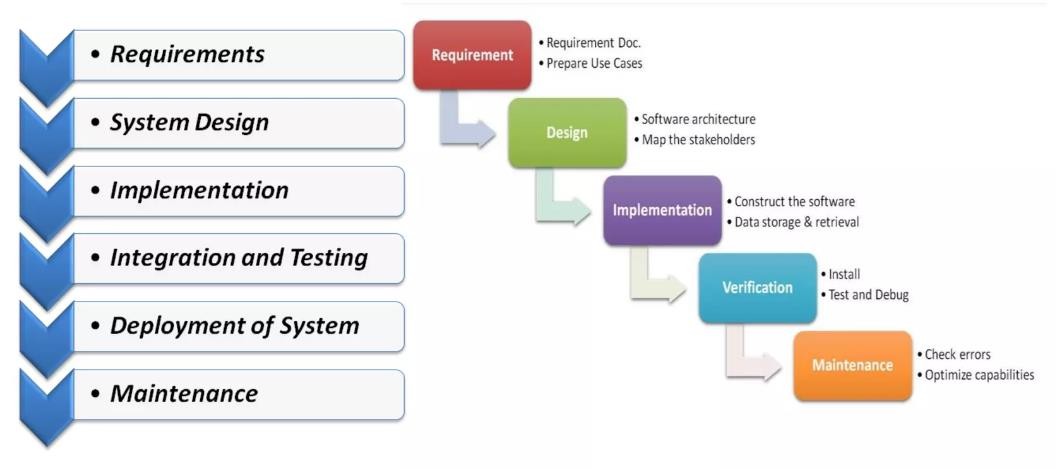


Figure 1.1 (Linear-Sequential Life Cycle Model)

# CHAPTER 2

# Related work

2.1 **Introduction**

In today's fast-paced world, transportation services play a crucial role in ensuring convenience, efficiency, and accessibility. With increasing urbanization and the growing demand for reliable transport solutions, there is a need for an innovative ride-hailing service that seamlessly connects riders with trusted drivers.

Inspired by the success of platforms like Uber and Lyft, our application is designed to provide a safe, efficient, and user-friendly transportation experience. Whether it's a daily commute, an airport transfer, or a ride across the city, our app offers a hassle-free solution tailored to modern users' needs.

With real-time tracking, multiple payment options, ride-sharing features, and a robust rating system, we aim to enhance the user experience while promoting affordability and environmental sustainability. By leveraging the latest mobile and cloud technologies, our platform ensures smooth and reliable operations for both riders and drivers.

This document outlines our app’s target audience, core features, technology stack, and key objectives, demonstrating how we intend to revolutionize urban mobility.

Target Audience:

The app is designed to serve different types of users who require reliable and efficient transportation. These include:

**1. Urban Dwellers**:

People living in cities often need quick and affordable transport options.

They may not own a car or prefer ride-hailing due to traffic and parking challenges.

The app provides them with a flexible alternative to traditional taxis and public transport.

**2. Tourists**:

Visitors in a new city often struggle to find safe and reliable transport.

The app ensures they can book rides easily, even if they are unfamiliar with the area.

Real-time tracking and multiple payment options improve their experience.

**3. Busy Professionals**:

Working individuals often have tight schedules and need dependable transport to meetings, airports, or offices.

The app saves them time by offering quick ride-booking and efficient routes.

By targeting these groups, the app ensures a broad user base and meets the transportation needs of various individuals.

2.2 **Core Features**

To provide a smooth and efficient service, the app includes several key features:

1**. User-Friendly Interface**:

The app is designed to be intuitive and easy to navigate for both riders and drivers.

Simple menus, clear buttons, and a straightforward booking process enhance usability.

2**. Real-Time Tracking**:

Users can track their ride’s location, estimated arrival time, and route.

This feature improves safety (users can share their ride details with family/friends) and reliability (users know exactly when their driver will arrive).

3. **Multiple Payment Options**:

The app supports various payment methods, including:

Credit/Debit Cards

Digital Wallets (e.g., Apple Pay, Google Pay)

Cash (if applicable)

This flexibility ensures that users can pay in a way that suits them best.

4. **Ride-Sharing Option**:

Allows multiple users to share a ride, reducing costs.

Encourages eco-friendly travel by lowering carbon emissions.

Beneficial for users looking for affordable transport.

5. **Rating and Feedback System**

Riders and drivers can rate each other based on their experience.

Feedback helps improve service quality by identifying issues and maintaining high standards.

Drivers with better ratings may receive more ride requests, encouraging professional behavior.

2.3 **Technology Stack**

To ensure a smooth and scalable experience, the app is built using modern technologies:

1. **Mobile Development**:

The app will be available on both iOS and Android.

Developers will use frameworks like Flutter (for a unified experience) or native development (Swift for iOS, Kotlin for Android).

2**. Backend Development**:

The backend will use cloud-based services (such as AWS, Google Cloud, or Firebase) for scalability.

A secure and efficient database (like PostgreSQL or MongoDB) will store ride information, user data, and payment details.

3. **Geolocation Services**:

GPS technology will power the app’s real-time tracking and route optimization.

Integration with Google Maps or Mapbox ensures accurate navigation.

By leveraging these technologies, the app will offer a smooth and reliable experience for both riders and drivers.

2.**4 Objectives**

The main goal of the app is to provide a safe, reliable, and efficient transportation service. The focus is on:

**1. Safety**:

Ensuring drivers undergo background checks.

Allowing users to share ride details for security.

Implementing a panic button for emergencies.

**2. Reliability**:

Matching riders with nearby drivers quickly.

Offering real-time tracking and accurate estimated arrival times.

Using AI to optimize routes and reduce delays.

**3. Efficiency**:

Providing quick and hassle-free ride-booking.

Offering multiple vehicle types (economy, premium, shared rides).

Continuously improving the app based on user feedback.

By focusing on these objectives, the app aims to increase customer satisfaction and build a loyal user base.

**Conclusion**

This app is designed to compete with leading ride-hailing services by offering a user-friendly, technology-driven, and customer-focused experience. By ensuring safety, reliability, and efficiency, it aims to become a preferred transportation choice for urban dwellers, tourists, and professionals.

# CHAPTER 3

# System Model

## 3.1Introduction

The **background chapter** provides essential knowledge to understand the rest of the document. This section offers an overview of the **current challenges in bus reservation systems**, the **need for an automated booking platform**, and the **technologies implemented** to address these issues. By analyzing the limitations of traditional bus ticketing methods, this chapter highlights the necessity of a **digital transformation** to enhance passenger convenience and operational efficiency for bus operators.

## 3.2Requirement Analysis

**Requirement analysis** is a crucial step in the software development process, shaping the design, development, testing, and implementation phases. A well-structured requirement analysis ensures that the project aligns with **user expectations, business goals, and technical feasibility**. By identifying key functionalities, potential risks, and user needs, this phase helps mitigate challenges such as **operational inefficiencies, system bottlenecks, and scope creep**, ensuring the successful execution of the project.

**Functional requirement**

* Online Payment System
* Real-Time Location Tracking.
* Trip Management for Drivers
* Notifications and Alerts
* Rating and Feedback System
* User Registration and Management
* Trip Booking

**Non functional requirement**

* Performance
* Security
* Scalability
* Usability
* Availability
* Maintainability
* Confidentiality

## 3.3Gathering Requirements

## The success of any bus reservation system depends on the proper identification and detailing of its core functionalities and system requirements. To avoid ambiguities and ensure accuracy in development, a combination of requirement gathering techniques was employed, including:

* **Observation of existing systems** (manual booking, traditional reservation methods).
* **Stakeholder interviews** to gather insights from passengers, bus operators, and administrators.
* **Surveys and questionnaires** to understand user needs and common travel concerns.
* **Prototyping and iterative feedback loops** to refine system requirements progressively.

The stakeholders of the **3al Taree2 Bus Reservation System** include:

a) **Admin** – Responsible for overall system **management, bus route oversight, and ensuring smooth platform operations**. The admin can **add or remove bus routes, manage user access, and oversee reservations** to maintain service integrity.

b) **Bus Operators** – Responsible for managing **bus schedules, monitoring seat availability, and handling ticket bookings**. Operators can adjust trip timings and seat allocations as needed.

c) **Passengers** – The end-users who utilize the platform to **search for buses, book seats, make payments, and receive electronic tickets** for their trips.

d) **Drivers** – Assigned to specific routes, drivers receive **real-time schedule updates** and can report trip statuses through the system, ensuring smooth coordination between passengers and operators.

By clearly defining the **roles, responsibilities, and expectations** of all stakeholders, the **requirement gathering process** ensures that **3al Taree2** effectively meets the needs of passengers and transportation providers while maintaining a **scalable and efficient** reservation system.

## Flow of current system:

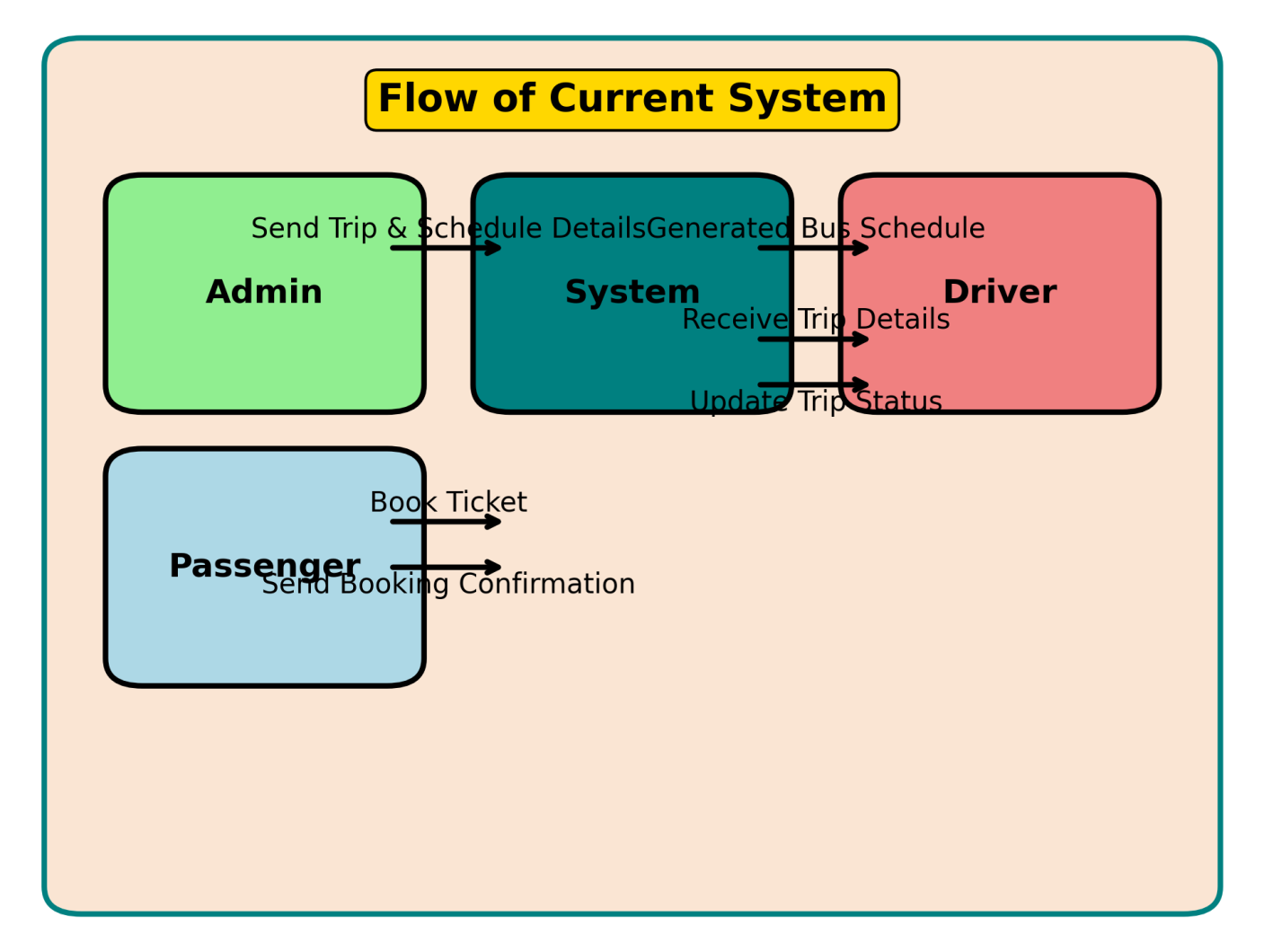
****

Figure 2.1 (flow of system)

## 3.1.1Requirement Analysis - Project Goals

**Modules of the System**

The system is divided into multiple functional modules, each handling a specific task to ensure an efficient, user-friendly experience. These modules help in managing bus reservations, optimizing schedules, and providing seamless user interactions.

As per the requirements, the 3al Taree2 Bus Reservation System primarily consists of six core modules: Login Module, Home Module, Routes & Schedule Module, Booking Module, Payment Module, and User Profile Module, all of which are accessible through the navigation bar, with the Logout option available separately.

## Login Module:

The Login Module allows users to securely access their accounts using their email and password. Only registered users can log in, ensuring data protection and security.

* **Authentication & Authorization:** Prevents unauthorized users from accessing the system.
* **Role-Based Access Control:** Different levels of access for **Passengers, Drivers, and Admins**.
* **Secure Login Mechanism:** Ensures encryption for passwords to **protect user data**

**Home Module**

### The **Home Module** serves as the **central control panel** for the system. It features:

* **A navigation bar** that links to different system modules.
* **A workspace** where users can view their **dashboard, active bookings, and trip history**.
* **System-wide control**, allowing admins and operators to **manage trip schedules, bookings, and payments**.

### ****Booking Module****

The **Booking Module** enables passengers to **search for available buses, select their seats, and confirm reservations**.

* **Real-time Seat Availability:** Ensures passengers **see available seats before booking**.
* **Interactive Seat Selection:** Allows users to **choose their preferred seats**.
* **Booking Confirmation:** Generates a **digital ticket** upon successful reservation.

### ****Payment Module****

The **Payment Module** facilitates **secure online transactions** for ticket purchases.

* **Multiple Payment Options:** Supports **credit/debit cards, mobile wallets, and cash on delivery**.
* **Transaction Security:** Ensures **encrypted payment processing**.
* **Instant Booking Confirmation:** Sends an **e-ticket** to the passenger **after payment approval**

### ****User Profile Module****

* **The User Profile Module allows passengers and drivers to manage their personal details and track their booking history.**
* **Passenger Features: View active bookings, past trip history, and personal details.**
* **Driver Features: View assigned trips, update trip status, and check earnings.**
* **Admin Controls: Manage user accounts, update settings, and handle customer inquiries.**

### ****Routes & Stops Module****

This module is responsible for **managing all bus routes and stop points**. Each route includes **multiple stops, travel durations, and assigned buses**, ensuring **efficient trip planning**.

* **Route Management:** Allows admins to **add, edit, or remove** routes based on demand.
* **Stop Point Management:** Each route contains **predefined stops** where passengers can board or exit.
* **Capacity Management:** Each bus has a **specific seat capacity**, preventing overbooking.

### ****Trip Scheduling Module****

This module handles **trip scheduling, departure times, and route optimization** to ensure **efficient travel management**.

* **Bus Allocation:** Assigns buses to specific **routes based on demand and availability**.
* **Timetable Management:** Ensures **buses operate on a structured schedule**, minimizing delays.
* **Real-time Updates:** Allows **operators to modify schedules** if needed and informs passengers instantly.

### ****Seat Reservation & Booking Module****

The **Seat Reservation Module** ensures a **smooth and optimized booking experience** for passengers while maximizing bus efficiency.

* **Real-time Seat Selection:** Passengers can **view available seats and choose their preferred spot**.
* **Automated Seat Allocation:** Prevents **double bookings** and ensures **optimal seat distribution**.
* **Booking Confirmation:** Generates **e-tickets instantly** upon successful booking.

## Functional Requirements

## Functional requirements in the context of software development refer to detailed specifications that define the specific functionalities, features, and behaviors that a system must exhibit. These requirements serve as a blueprint for the development team, guiding the design, implementation, and testing processes. They define the desired outcomes, inputs, processes, and outputs of the system, ensuring that it meets the functional expectations of its users and stakeholders.

The following major components and their functions were identified as **core business processes** in the **3al Taree2 Bus Reservation System**.

## User Login

* + The system has **three main user roles**: **Admin, Passenger, and Driver**.  
    ▪ Only **registered users** (Admin, Driver, or Passenger) can log in using valid credentials.  
    ▪ If the **username or password is incorrect**, an error message will be displayed, and users must **reset their password** if needed.  
    ▪ The system **validates username and password securely** to prevent unauthorized access.  
    ▪ A **successful login** directs users to their respective home pages based on their role.

## System Home page

▪ The **home page** provides a **login form and help section** for new users.  
▪ The system **identifies the user role (Admin, Passenger, or Driver)** and redirects them to their respective dashboard.

## Admin home page

* + The Admin Dashboard allows for the management of all system operations.  
    ▪ Admin can register and manage users (Passengers & Drivers).  
    ▪ Admin can add, edit, and remove bus routes, schedules, and stops.  
    ▪ Admin can monitor active bookings, revenue, and system reports.  
    ▪ Admin has the ability to grant or revoke privileges for other system users.

## Passenger Dashboard

* + Passengers can **search for bus routes, select travel dates, and book tickets**.  
    ▪ The system allows passengers to **view available seats and choose their preferred spot**.  
    ▪ A **digital ticket** is generated upon successful booking.  
    ▪ Passengers can **track their trip status** and receive real-time updates.  
    ▪ They can **cancel or reschedule trips** based on system policies.
* **Driver Dashboard**
  + Drivers can **view their assigned trips and schedules**.  
    ▪ The system allows drivers to **update trip status in real-time** (e.g., "On the way," "Arrived").  
    ▪ Drivers receive notifications about **new trip assignments** and changes.  
    ▪ They can **report trip issues** directly to the admin.

## Non-Functional Requirements

"Non-functional requirements may be more critical than the functional requirements. If these are not met, the system is useless" (Sommerville, 2004).

Non-functional requirements define the overall quality, security, and performance expectations of the 3al Taree2 Bus Reservation System. These aspects ensure that the system operates efficiently, securely, and reliably for passengers, drivers, and administrators.

### ****➤ Performance:****

**Description:** The system should **respond quickly** to user interactions and transactions.  
**Criteria:**  
✔ Fast **seat selection and booking process**.  
✔ Real-time **bus schedule updates**.  
✔ Efficient **database queries** for route, booking, and payment information.

### ****➤ Scalability:****

**Description:** The system should be able to **handle increasing numbers of users, bookings, and routes** as demand grows.  
**Criteria:**  
✔ Support for an **expanding fleet of buses** and new routes.  
✔ Ability to **handle thousands of concurrent users** without performance degradation.

### ****➤ Reliability:****

**Description:** The system should be **highly available and operate without failures**.  
**Criteria:**  
✔ Ensure **99.9% uptime** for uninterrupted bookings.  
✔ Minimal **downtime for maintenance and updates**.

### ****➤ Usability:****

**Description:** The system should be **intuitive, user-friendly, and easy to navigate**.  
**Criteria:**  
✔ A **simple, clean, and interactive interface** for passengers and drivers.  
✔ User **help guides and support** for troubleshooting.

### ****➤ Security:****

**Description:** Ensure **data confidentiality, integrity, and protection** from cyber threats.  
**Criteria:**  
✔ **Secure user authentication** (e.g., encrypted passwords, OTP verification).  
✔ **Role-based access control** for Admin, Passenger, and Driver.  
✔ **Secure payment processing** to protect financial transactions.

### ****➤ Maintainability:****

**Description:** The system should be **easy to maintain, update, and improve**.  
**Criteria:**  
✔ **Well-documented and modular** code for easier debugging and future enhancements.  
✔ **Admin panel for easy system configuration** (routes, schedules, and pricing updates).

### ****➤ Compatibility:****

**Description:** The system should work **across various devices and web browsers**.  
**Criteria:**  
✔ **Responsive design** for mobile, tablet, and desktop use.  
✔ **Cross-browser compatibility** (Chrome, Firefox, Edge, Safari, etc.).

### ****➤ Data Backup and Recovery:****

**Description:** The system should have **a backup strategy** to prevent data loss.  
**Criteria:**  
✔ **Automated daily backups** to a secure storage location.  
✔ **Quick data recovery** in case of accidental deletions or system failures.

### ****➤ Auditability:****

**Description:** The system should **log all user activities** for tracking and security audits.  
**Criteria:**  
✔ **User logs for bookings, payments, and trip updates**.  
✔ **Admin dashboard for generating audit reports** when needed.

## 2. User Requirements

The **end-user requirements** are minimal. Passengers, drivers, and admins only need:  
✔ Basic **computer and mobile usage skills**.  
✔ Knowledge of **web and mobile browsing techniques**.  
✔ A **valid internet connection** for accessing the system online.

## System Requirements

The **3al Taree2 Bus Reservation System** follows a **3-Tier Architecture**, ensuring **structured data processing, security, and scalability**.

✔ **Backend Layer (Database Server):** Stores all **user data, bus schedules, seat availability, and bookings**. Uses **SQL Server** for structured data storage.

✔ **Application Layer (Web Server):** Processes all **requests and system logic**, developed using **ASP.NET Core Web API**.

✔ **Frontend Layer (User Interface):** A **Flutter-based web and mobile interface** providing an **interactive experience for passengers, drivers, and administrators**.

## The Backend (Database Server)

The database server is a crucial component of the 3al Taree2 Bus Reservation System, responsible for storing and managing all user data, bus schedules, booking records, and payment transactions.

* **Recommended Server Specifications:**  
  ✔ **2 CPU cores (recommended)** for efficient query processing.  
  ✔ **32GB or higher RAM** to handle large data operations.  
  ✔ **1TB or higher storage space**, depending on the expected system growth.  
  ✔ **SQL Server** is used as the database management system for **high performance and data integrity**.

## The Application Server (Web Server)

The **application server** handles all **business logic, request processing, and system functionalities**, ensuring smooth operation and high performance.

* **Recommended Server Specifications:**  
  ✔ **4 CPU cores (recommended)** for handling multiple user requests.  
  ✔ **32GB or higher RAM** for optimized application execution.  
  ✔ **1TB or higher storage capacity** to support logs, updates, and dynamic data.  
  ✔ **1 Mbps or higher network speed** for uninterrupted access to cloud-based services.  
  ✔ **ASP.NET Core Web API** is used for server-side processing, ensuring security and efficiency.

## Client

The **client-side requirements** are minimal, as the **3al Taree2 Bus Reservation System** is a **web and mobile-based platform** accessible from any internet-enabled device.

* **Requirements:**  
  ✔ Any **standard web browser** (Chrome, Firefox, Safari, Edge).  
  ✔ **Stable internet connection (1 Mbps or higher)** for smooth transactions.  
  ✔ Compatible with **desktops, tablets, and smartphones** for user convenience.

## Related Technologies

Since **3al Taree2** needs to be **accessible to passengers, drivers, and admins** across multiple locations, a **web-based system** is the most suitable solution. The system is designed to work seamlessly in a **unified online environment**, ensuring **cross-platform compatibility and high availability**.

## Web Application

## 3al Taree2 is developed as a web and mobile application to provide users with a fast, efficient, and convenient way to book and manage bus trips.

## ✔ Server-Side Technologies:

* **ASP.NET Core Web API** is used to **process requests, manage data, and handle authentication**.
* **SQL Server** stores all system data **securely and efficiently**.

✔ **Client-Side Technologies:**

* **Flutter** is used to develop the **cross-platform front-end** for both web and mobile users.
* The system supports **responsive design**, ensuring accessibility across all devices.

By leveraging these technologies, **3al Taree2** ensures **high performance, security, and a seamless user experience**, making bus reservations **easier and more accessible for passengers and transport operators**. 🚍✨

### ****Tools Used in 3al Taree2****

To ensure a **smooth development process**, the **3al Taree2 Bus Reservation System** is built using a combination of **design, development, and database management tools**.

1. **Design:** Figma – Used for designing the **user interface (UI) and user experience (UX)** of the system.
2. **Coding:** Visual Studio Code – Used for **frontend development**.
3. **Backend Development:** Visual Studio (for ASP.NET Core Web API).
4. **Database Management:** SQL Server – Handles **data storage, queries, and transactions**.
5. **Diagrams & Documentation:** Draw.io – Used for creating **system architecture diagrams and flowcharts**.

### ****Techniques Used****

**To ensure an optimized and efficient system, the following technologies and frameworks are implemented:**

✔ **Front-End Development:**

* **Flutter** – A cross-platform framework for creating **responsive and interactive user interfaces**.
* **HTML, CSS, SCSS, JavaScript** – Used for designing the **web version** of the platform.

✔ **Back-End Development:**

* **ASP.NET Core Web API** – Used for handling **system logic, authentication, and database interactions**.

✔ **Database Management:**

* **SQL Server** – Ensures **secure and structured data storage** for user details, booking records, and bus schedules.

## Database Access

The **3al Taree2 Bus Reservation System** is built on a **real-time database** that supports **storing, updating, deleting, and retrieving data efficiently**. The system is designed to **manage large-scale transportation data** while ensuring **high availability and security**.

The database stores the following **core system data**:  
✔ **User Data:** Admins, Passengers, and Drivers with their credentials and booking history.  
✔ **Bus Routes & Schedules:** Information on **routes, bus stops, departure times, and trip durations**.  
✔ **Booking & Seat Allocation:** Tracks **ticket purchases, seat assignments, and cancellations**.  
✔ **Payment Records:** Stores **transaction details for completed and pending payments**.  
✔ **Trip Updates & Logs:** Maintains logs of **trip statuses, driver updates, and system notifications**.

### ****Database Technology****

The **3al Taree2 system** is powered by **SQL Server**, a **high-performance relational database** designed for:

* **Secure and Scalable Transactions:** Ensuring **safe storage of user and booking data**.
* **ACID Compliance:** Supports **commit, rollback, crash recovery, and row-level locking**.
* **Optimized Query Performance:** Ensures **fast data retrieval** for real-time booking and scheduling.

The **database system** is designed to **store data for long-term analysis**, allowing the system to **track trends, optimize routes, and enhance service offerings**. By leveraging **SQL Server**, **3al Taree2** ensures **data integrity, reliability, and high-speed processing**, making it a **robust and scalable solution** for bus reservations. 🚍✨

## System Architecture

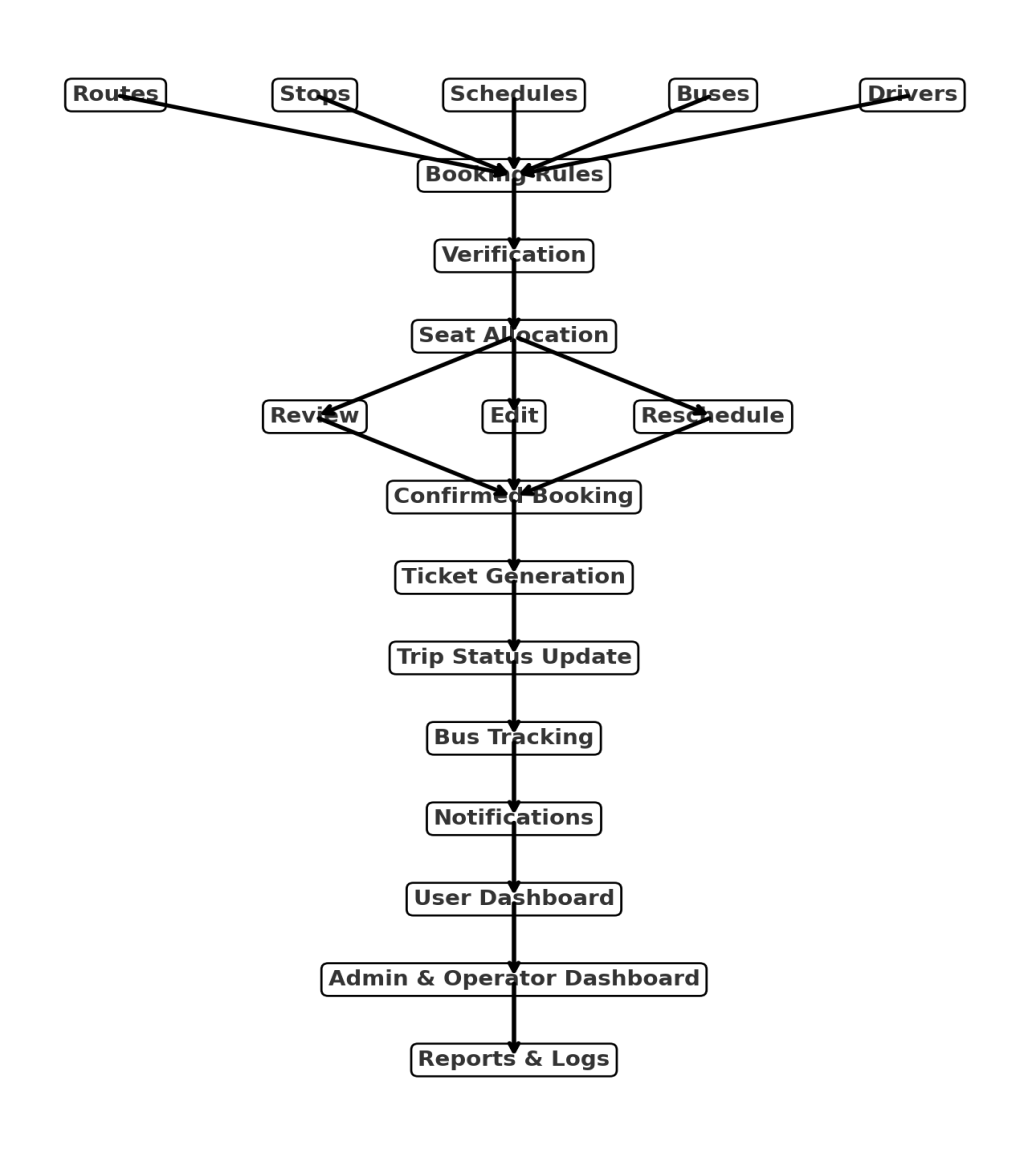
The architecture of a system reflects how the system is used and how it interacts with other systems and the outside world. It describes the interconnection of all the system’s components and the data link between them. The architecture of a system reflects the way it is thought about in terms of its structure, functions, and relationships.

Figure 2.2 (System Architecture)

## Client-Server Architecture

Users access the web application installed on the Application server through a web browser. Initially, the web application connects to the University Internet. Subsequently, the connection proceeds to a security layer responsible for user validation. Upon successful validation, users gain access to the Application server and MySQL database server.

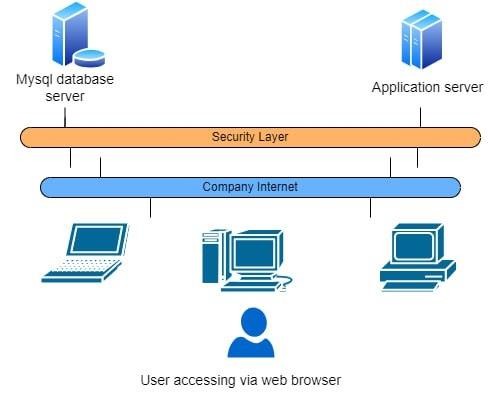
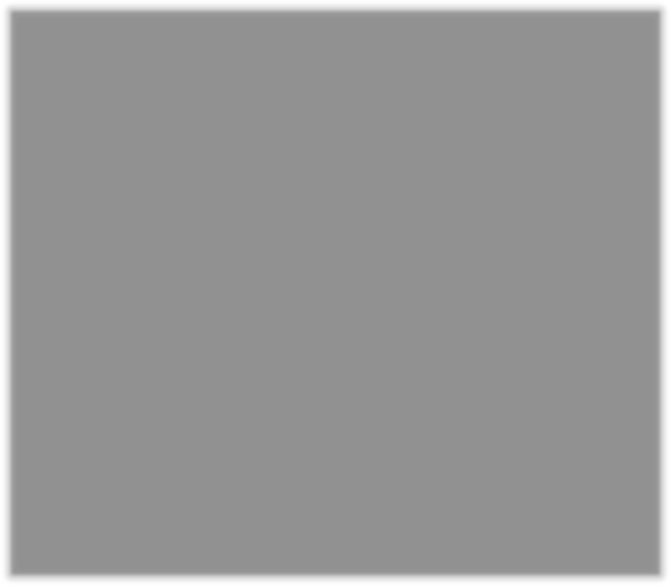
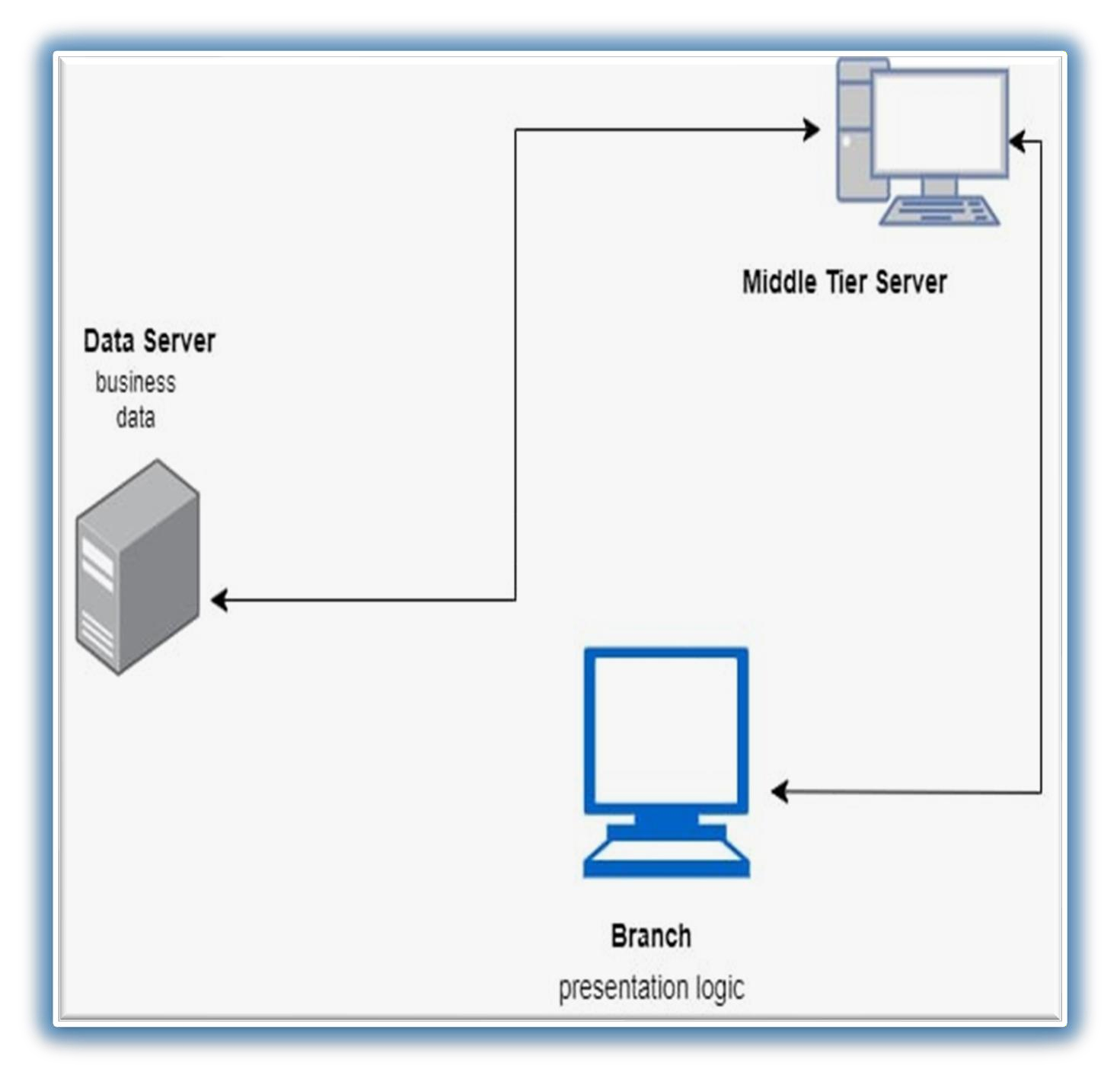


Figure 2.3(User accessing via web browser)

## Tier Architecture



## Tier architecture contains UI/ Presentation Layer (Branch), Middle tier layer and Data Access Layer.

Figure 2.4 (Tier Architecture)

## Existing Similar Systems

A significant amount of effort and time was spent researching existing bus reservation systems to analyze their benefits, drawbacks, and functionalities. By studying these systems, we identified key features, challenges, and areas for improvement, which helped in designing 3al Taree2 to be more efficient, user-friendly, and scalable.

### ****Redbus – Online Bus Ticket Booking System****

**Redbus** is one of the most popular **bus booking platforms** that allows users to **search, compare, and book bus tickets online**. It operates across multiple regions, offering a **seamless travel booking experience** with advanced features.

#### ****Key Features:****

✔ **Automated Seat Selection:** Users can view **real-time seat availability** and choose their preferred seats.  
✔ **Secure Payment Gateway:** Supports **multiple payment options**, including wallets, debit/credit cards, and UPI.  
✔ **Live Bus Tracking:** Passengers can track their **bus in real-time** and receive estimated arrival times.  
✔ **Flexible Booking & Cancellations:** Allows **easy rescheduling** and **refund policies**.

#### ****Limitations Identified:****

✖ **Limited customization for small operators.**  
✖ **High commission fees for bus service providers.**  
✖ **No offline mode for ticket validation in case of poor network connectivity.**

### ****FlixBus – Long-Distance Bus Travel Platform****

FlixBus is an **intercity bus booking system** that provides **a digital ticketing solution for passengers** across different countries.

#### ****Key Features:****

✔ **Dynamic Pricing:** Ticket prices change based on demand and seat availability.  
✔ **Integrated Mobile App:** Users can **book, modify, and manage trips** on their mobile phones.  
✔ **Loyalty & Discount Programs:** Offers **discounts and special offers** to frequent travelers.  
✔ **Eco-Friendly Travel Initiative:** Focuses on **sustainable and carbon-neutral transportation**.

#### ****Limitations Identified:****

✖ **Limited local and regional bus networks.**  
✖ **Complex refund policies in some regions.**  
✖ **Service restrictions in certain areas with limited internet access.**

### ****How 3al Taree2 Improves on Existing Systems****

Based on the analysis of **Redbus, FlixBus, and similar bus reservation systems**, **3al Taree2** aims to provide a **more adaptable and accessible** solution with:

✔ **User-Friendly Seat Selection:** Passengers can **visually select seats** and confirm bookings instantly.  
✔ **Transparent Pricing & Flexible Bookings:** No **hidden charges** or high commissions for operators.  
✔ **Integrated Driver & Admin Dashboard:** Ensuring **smooth communication between passengers, drivers, and operators**.  
✔ **Offline Ticket Verification Option:** Allows **conductors to verify tickets even in areas with poor network coverage**.  
✔ **Localized Approach:** Unlike global competitors, **3al Taree2** is designed to **support regional and intercity transport networks**.

By **addressing the limitations of existing systems** and incorporating **innovative solutions**, **3al Taree2** provides a **more reliable, cost-effective, and efficient** **bus booking experience** for passengers and transport operators alike

# CHAPTER 4

# Proposed system

## Introduction

The system design phase is one of the most critical and time-consuming stages in the development of a software project. This phase translates the functional and non-functional requirements gathered earlier into a structured and detailed system architecture. The objective is to develop a logical design that outlines the flow, structure, and behavior of the system.

During this phase, the focus is on:  
✔ User Interface Design: Ensuring an intuitive and user-friendly experience for passengers, drivers, and administrators.  
✔ Database Design: Structuring data efficiently to support real-time bookings, trip management, and payment records.  
✔ System Workflow & Outputs: Defining how users interact with the system and receive feedback.

The technical or implementation component of 3al Taree2 is designed with an emphasis on efficiency, security, and scalability. The system analysis phase conducted earlier provides the blueprint, which is now translated into a technical framework through software architecture, database design, and interface development.

This chapter will detail how 3al Taree2 is designed to deliver a seamless, robust, and efficient bus reservation experience for passengers, drivers, and transport administrators. 🚍✨

## Use-Case

**Use Case Diagram**

A Use Case Diagram is a visual representation that helps identify the main components and actions within the 3al Taree2 Bus Reservation System. This diagram provides an overview of system interactions, where the key elements are:

* **Actors** – Represent **users or external systems** interacting with the system.
* **Use Cases** – Define **specific actions or processes** that the system performs.

The **Use Case Diagram** focuses on the **functional aspects** of the system, capturing **business processes** such as **user authentication, ticket booking, trip scheduling, and payment processing**. These diagrams help in:

✔ **Understanding system functionality** at a high level.  
✔ **Defining requirements** for development.  
✔ **Serving as a basis** for creating test scripts to ensure **smooth system performance**.

In the **3al Taree2 Bus Reservation System**, the main **actors** include:

* **Passenger** – Searches for routes, books tickets, makes payments, and views trip details.
* **Driver** – Views assigned trips, updates trip statuses, and manages schedules.
* **Admin** – Manages routes, schedules, bookings, and user accounts.

The **Use Case Diagram** visually represents how these actors **interact with the system’s features**, ensuring a **seamless and user-friendly experience** for all stakeholders.

.

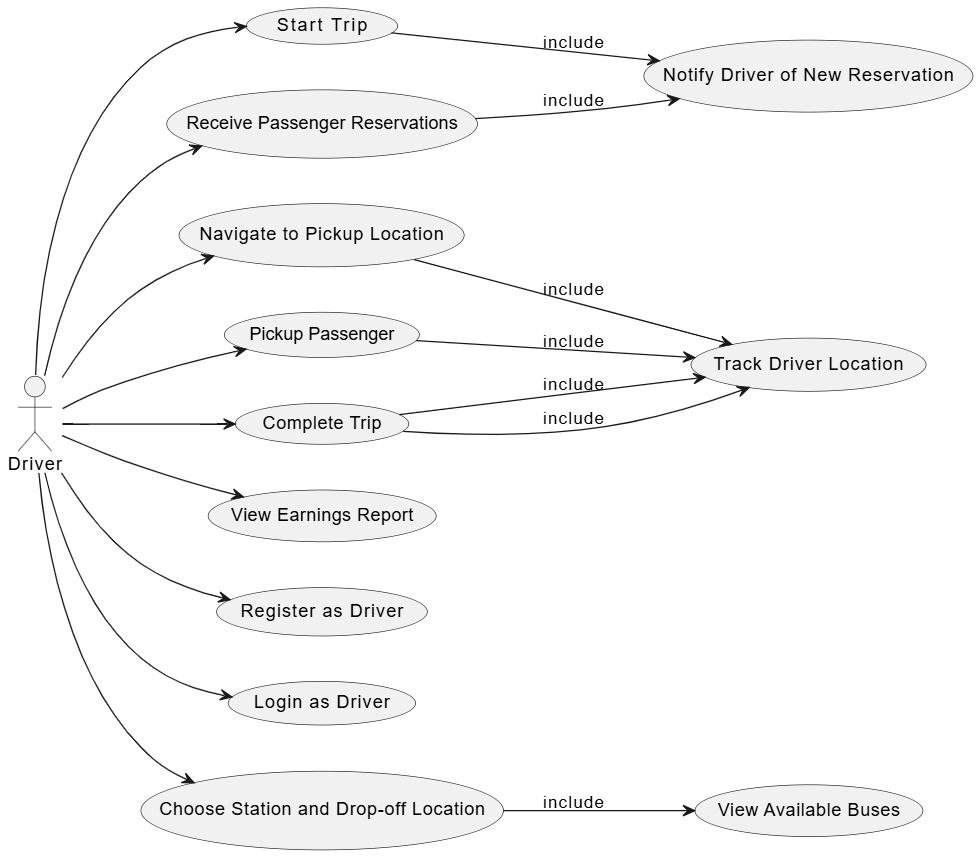
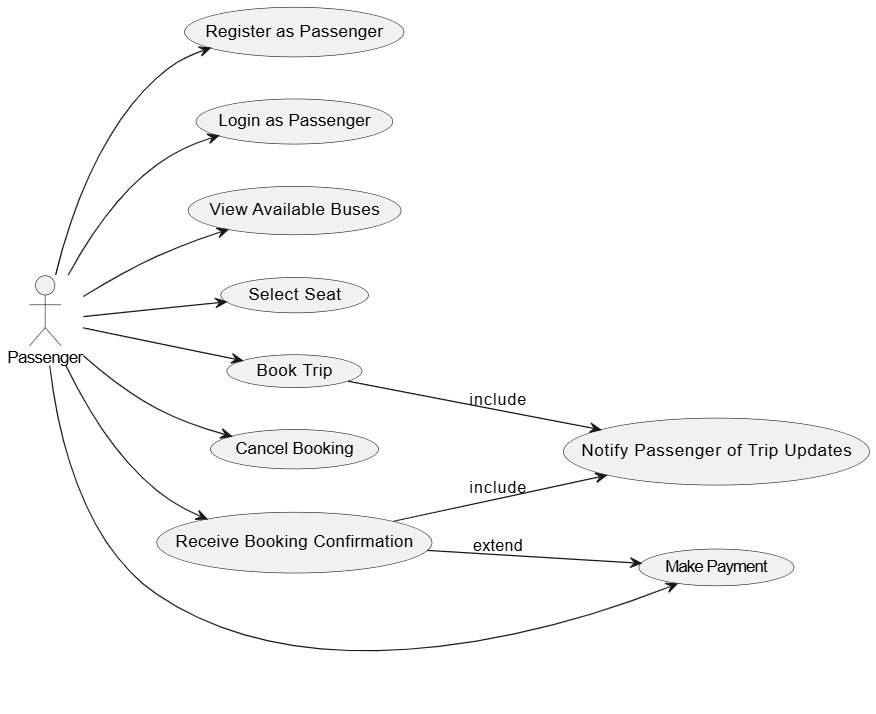
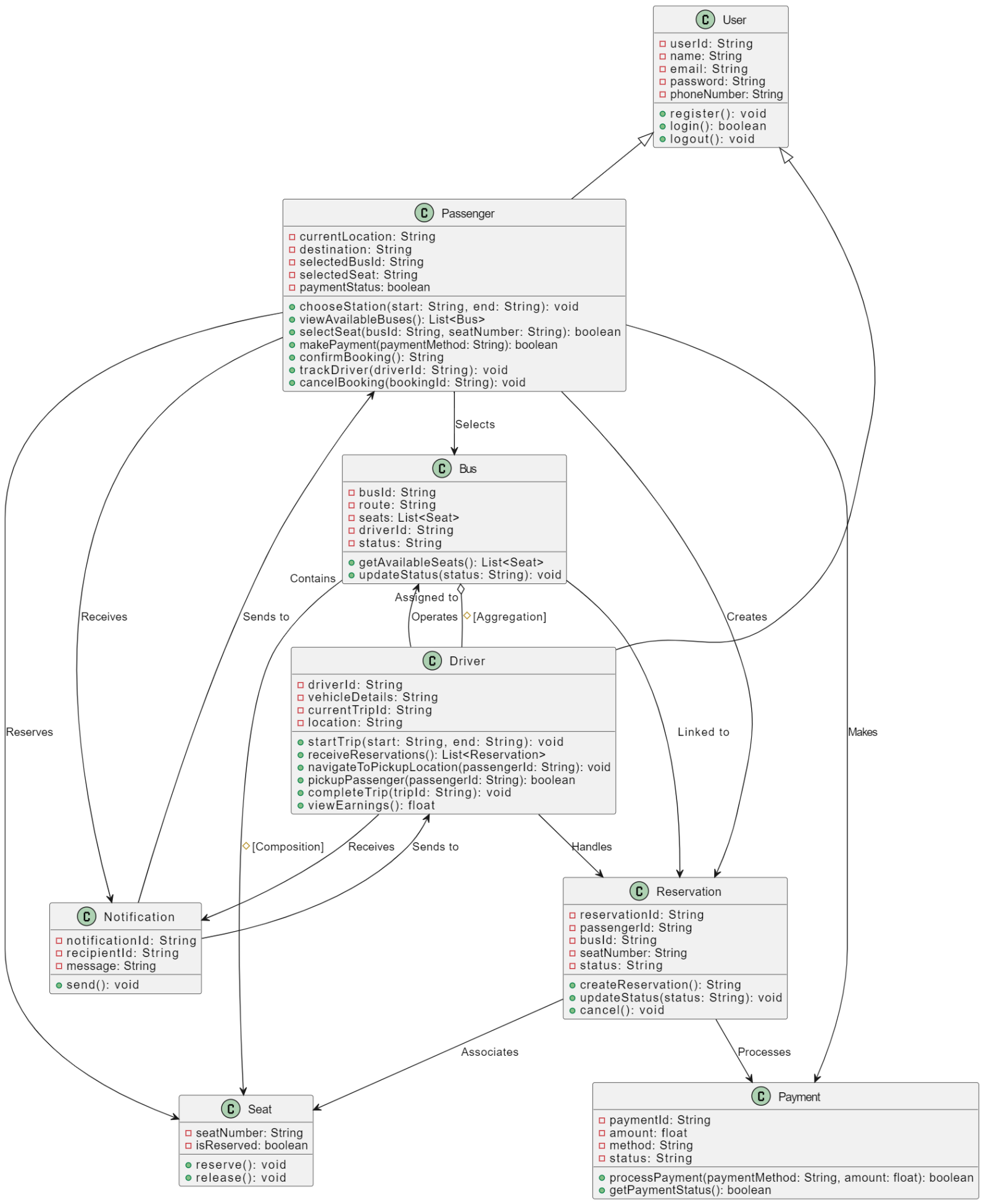


Figure 3.1(use-case diagram)



## Class Diagram

Class diagrams play a crucial role in object-oriented analysis and design. They visually represent the classes in a system, showcasing their relationships, such as inheritance, aggregation, and association, along with the activities and attributes of these classes. These diagrams serve multiple purposes, ranging from conceptual/domain modeling to detailed design modeling, offering a comprehensive view of the system's structure and functionality. To see the full diagram, [click here](https://drive.google.com/file/d/1oRo28lUinUeJEX4PupAmNpSojrUoosNk/view?usp=sharing).



## ERD Diagram

It is a visual representation used in system analysis to model and describe the relationships between different entities in a system. In the context of database design and software engineering, ERDs are a crucial tool for understanding the structure of a system and its underlying data

This diagram represents a database schema or an Entity Relationship Diagram (ERD) for a system

that manages trip reservations, passengers, payments, and drivers. Below is the explanation:

**1. Passenger**:

- Attributes: Id, Email, FullName, PhoneNumber, DateOfBirth, Password, ProfilePicture.

- The Passenger can have multiple Reservations, as indicated by the relationship between

Passenger and Reservation (one-to-many).

**2. Trip:**

- Attributes: Id, StartLocation, EndLocation, BusId, DriverId, StartTime, EndTime.

- Each trip is associated with multiple reservations and may be linked to feedback.

**3. Reservation**:

- Attributes: Id, SeatNumber, ReservedAt, TripId, PassengerId.

- Each reservation has a specific passenger and a trip associated with it. The relationship between

Reservation and Trip is one-to-many (one trip can have many reservations).

**4. Reservation-Payment**:

- This is a junction table linking Reservation and Payment.

- It contains attributes such as ReservationId, PaymentId, and Amount.

**5. Payment**:

- Attributes: Id, Amount.

- A payment is associated with a reservation through the Reservation-Payment entity. The

payment can have an amount tied to it.

**6. Feedback**:

- Attributes: Id, ReservationId, PassengerId, Comment.

- Passengers can leave feedback for trips, and each feedback is linked to a specific reservation.

**7. Driver**:

- Attributes: Id, FullName, Email, PhoneNumber, LicenseNumber.

- Drivers are associated with trips, with a one-to-many relationship from Driver to Trip (one driver

can be associated with many trips).

**8. Relationships**:

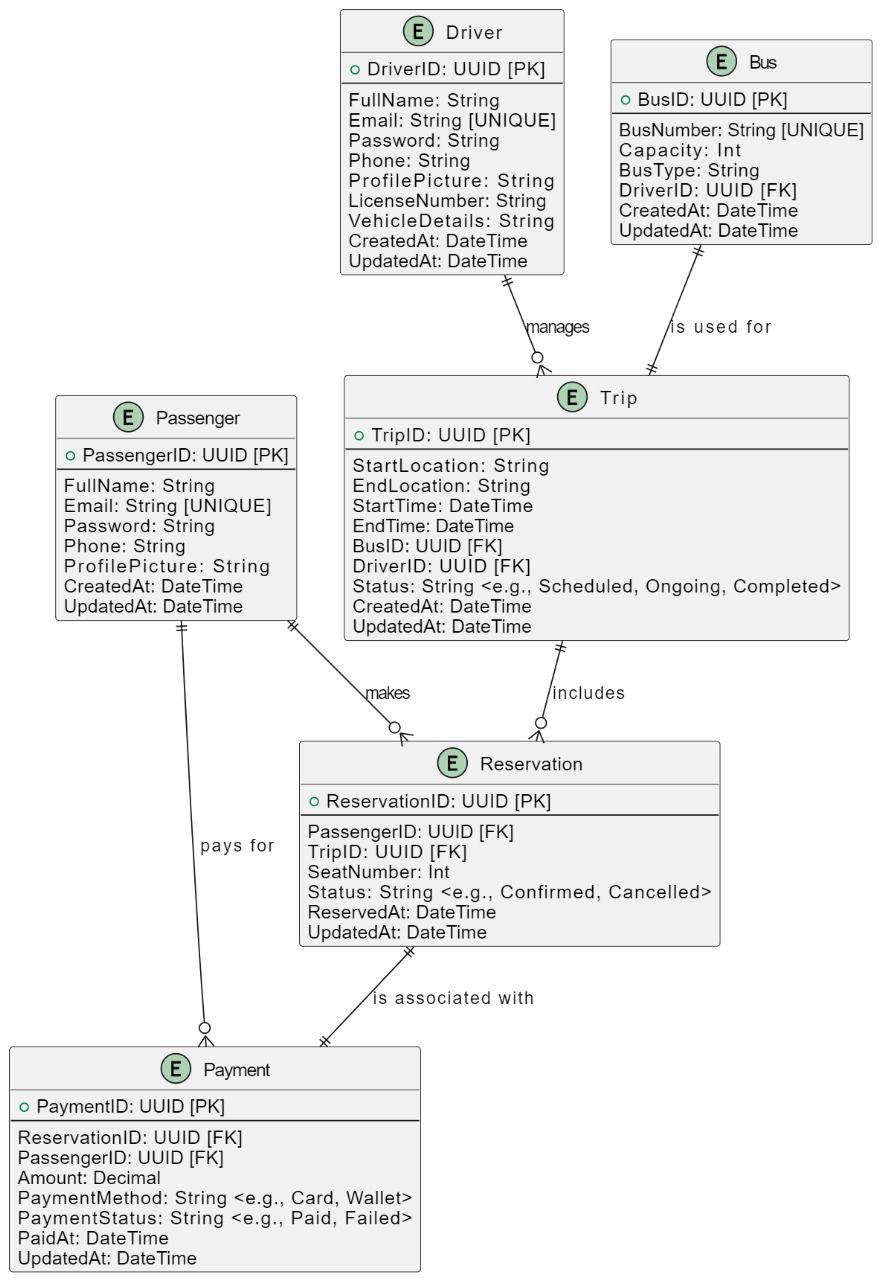
- Passenger-Reservation: One passenger can make multiple reservations.

- Trip-Reservation: One trip can have many reservations.

- Reservation-Payment: Each reservation may have a related payment.

- Reservation-Feedback: A passenger can provide feedback for each reservation they make.

This ERD is useful for understanding how different entities in the system interact with one another,

such as how a passenger makes reservations, how payments are processed, and how feedback is linked to t

## Sequence Diagram

During the execution of a use case or any action, this defines how objects communicate with one another via messages. They show how messages are delivered and received between objects, as well as the order in which messages are sent. It also describes how operations are carried out according to the time of operation.

* This **sequence diagram** illustrates the interaction between different entities (actors) in the system.
* **Actors:**
  + **Passenger**: Registers, books a trip, selects a bus, and makes a payment.
  + **Driver**: Starts trips, picks up passengers, or declines reservations.
  + **App**: Manages booking, payments, and notifications.
  + **Services**:
    - **Booking Service**: Handles reservation requests.
    - **Payment Service**: Processes payments.
    - **Notification Service**: Sends notifications about reservations, trip status, and confirmations.
* **Key Steps:**
  + **Registration and Login**: Both passengers and drivers register and log in.
  + **Passenger Booking a Trip**:
    - Passenger selects a route, bus, and seat.
    - Booking service confirms the reservation.
    - Payment is processed.
    - Notifications are sent to inform the driver.
  + **Driver Starts a Trip**: Trip status is updated, and notifications are sent.
  + **Trip Completion**: The system updates trip status and notifies the passenger.

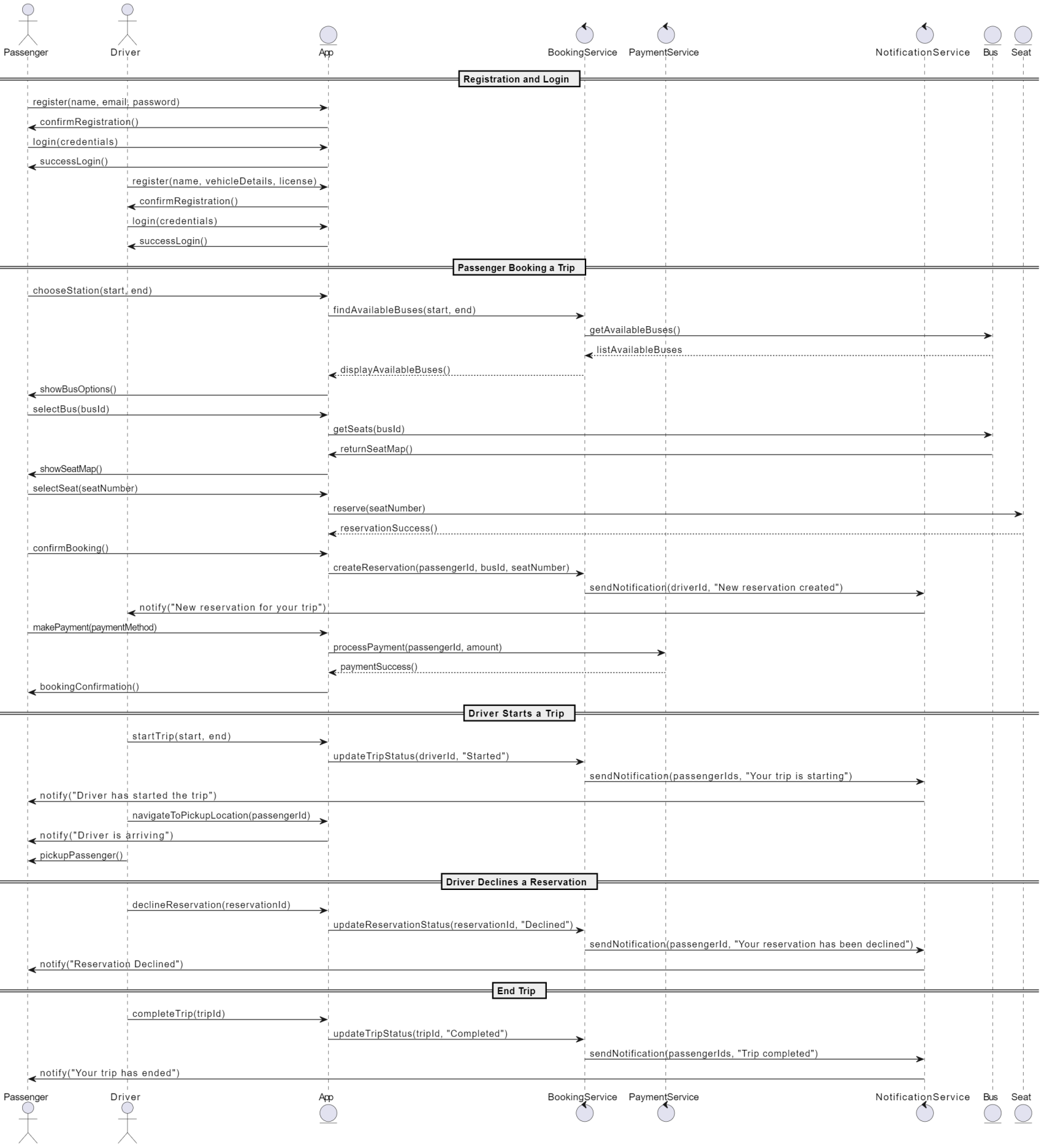


Figure 3.22 (Sequence diagram)

## Activity Diagram

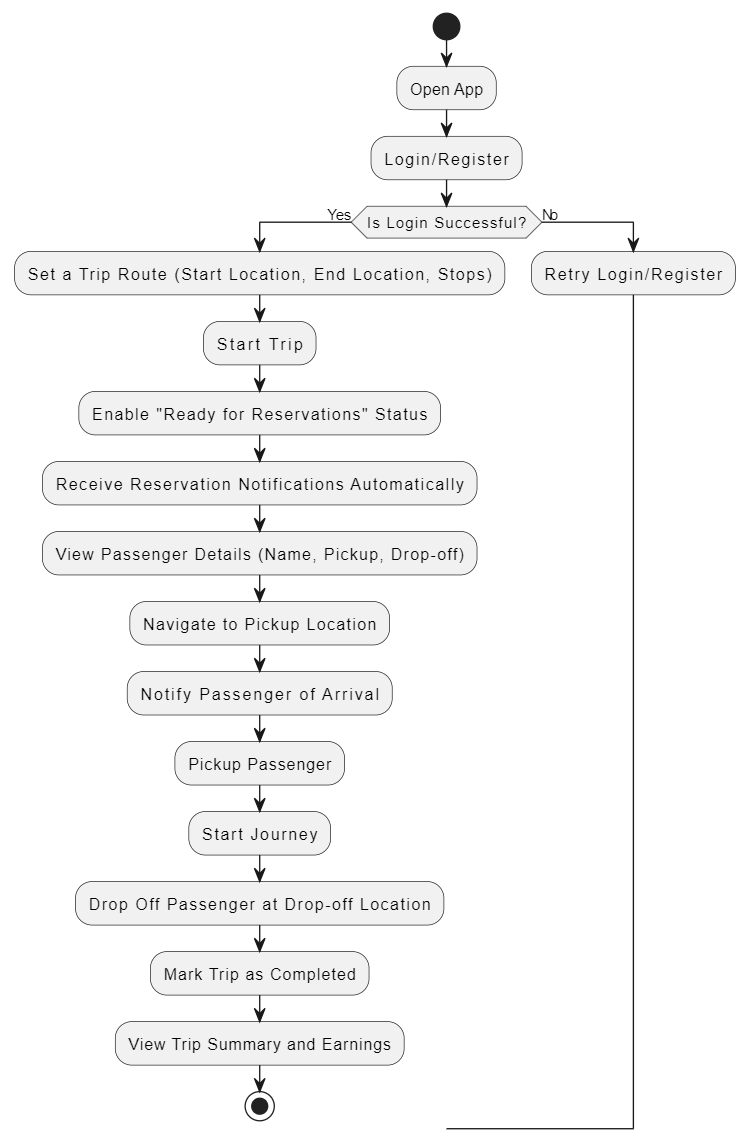
### ****Driver Activity Flow****

This **activity diagram** outlines the process for a driver using the app.

#### ****Key Steps:****

1. **Open App**
2. **Login/Register**
   * If login fails, retry login/register.
3. **Set Trip Route** (Start Location, End Location, Stops)
4. **Start Trip**
5. **Enable "Ready for Reservations" Status**
6. **Receive Reservation Notifications Automatically**
7. **View Passenger Details** (Name, Pickup, Drop-off)
8. **Navigate to Pickup Location**
9. **Notify Passenger of Arrival**
10. **Pickup Passenger**
11. **Start Journey**
12. **Drop Off Passenger at Drop-off Location**
13. **Mark Trip as Completed**
14. **View Trip Summary and Earnings**

This diagram focuses on how drivers manage trips, receive and accept ride requests, and complete their journey.



### ****Passenger Activity Flow****

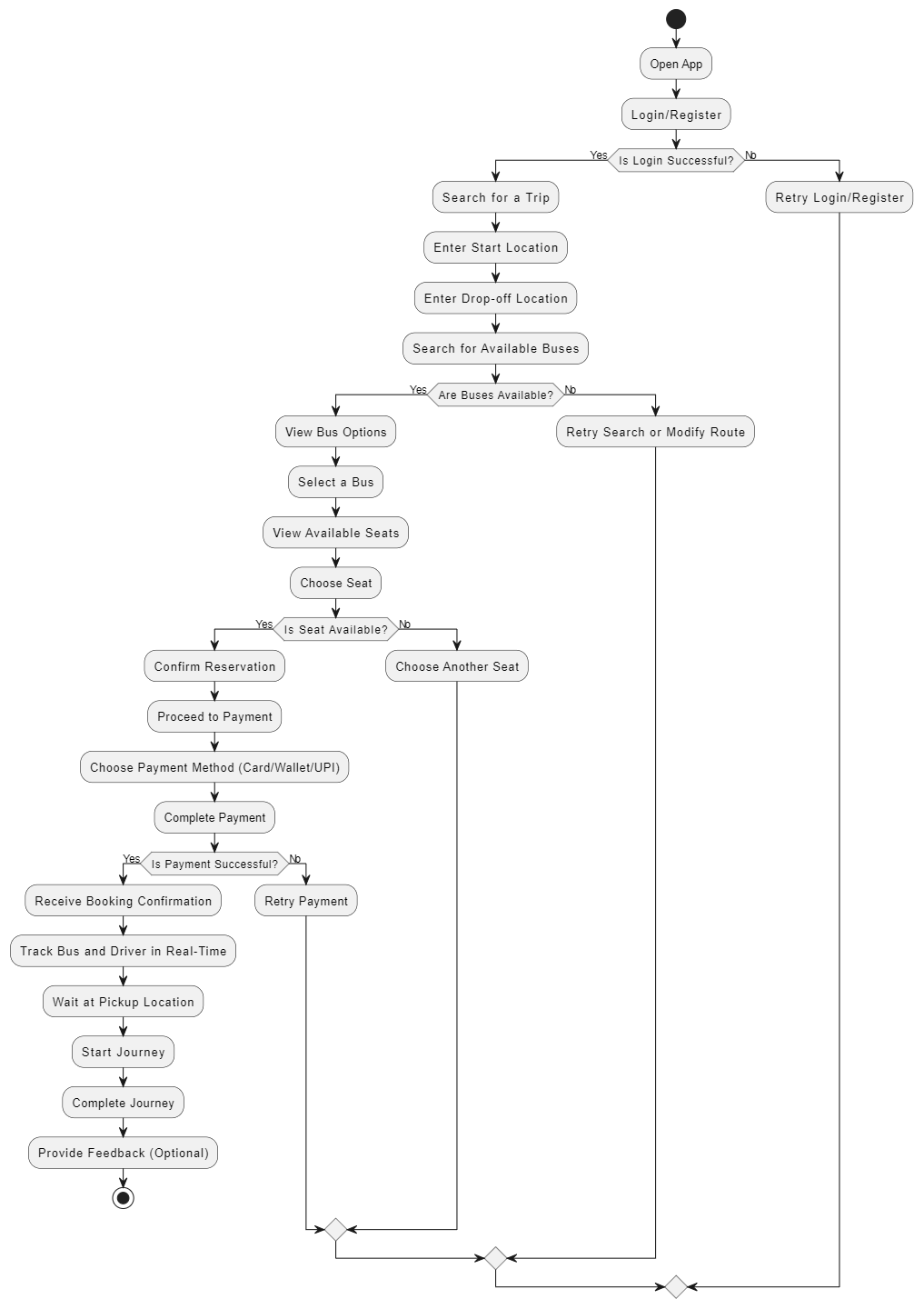
This **activity diagram** illustrates the **passenger's journey** in the app.

#### ****Key Steps:****

1. **Open App**
2. **Login/Register**
   * If login fails, retry login/register.
3. **Search for a Trip**
   * Enter **Start Location**
   * Enter **Drop-off Location**
   * Search for **Available Buses**
4. **Check Bus Availability**
   * If no buses are available, retry search or modify route.
5. **View Bus Options** → **Select a Bus**
6. **View Available Seats** → **Choose Seat**
7. **Check Seat Availability**
   * If unavailable, choose another seat.
8. **Confirm Reservation**
9. **Proceed to Payment**
   * Select Payment Method (Card/Wallet/UPI)
   * Complete Payment
10. **Check Payment Success**

* If failed, retry payment.

1. **Receive Booking Confirmation**
2. **Track Bus and Driver in Real-Time**
3. **Wait at Pickup Location**
4. **Start Journey**
5. **Complete Journey**
6. **Provide Feedback (Optional)**



# CHAPTER 5

***IMPLEMENTATION***

## 

## 

## Technologies, tools, and programming languages used:

## The development of the "2ala eltaree2" application utilized the following technologies and tools:

## Frontend: The mobile application was built using Flutter, a UI toolkit by Google for building natively compiled applications for mobile from a single codebase.

## Backend: The server-side logic and APIs were developed using ASP.NET, providing a robust and secure backend infrastructure.

## Database: Microsoft SQL Server was used to manage and store user data, trip information, bookings, and payment records.

## GPS Integration: The app includes GPS location services to allow users and drivers to select departure and arrival locations on the map.

## Payment Gateways: The system supports payments through Vodafone Cash and Visa via third-party payment APIs.

## Version Control: GitHub was used for source code management and collaboration among team members.

## Testing Tools: Postman for API testing and Flutter DevTools for debugging and performance analysis.

## . Key components/modules of the system

## The application consists of several core modules that work together to deliver a seamless experience for both passengers and drivers:

## User Registration and Authentication Module: Allows passengers and drivers to register, login, and maintain profiles. Gender is collected during registration to enable gender-based seat filtering.

## Frontend : passenger signup + login

## 

## Back end: passenger login

## 

## Front end : driver signup +login

## 

## Backend : driver login

## 

## 

## Trip Management Module:

## Frontend:

## For Drivers: Add trips, specify route, date, price, available seats, and pickup/drop-off locations using GPS

## 

## Back end; driver trip

## 

## 

## Front end :For Passengers: Browse available trips, filter based on location and date, and view details.

## 

## 

## Backend: passenger find trip

## 

## 

## Frond end : Seat Selection and Booking Module: Passengers can select specific seats based on availability. Booked seats are locked in real-time and displayed as reserved to other users.

## 

## Back end : seats + booking

## 

## 

## 

## 

## Frontend :Payment Module: Handles online payments using Vodafone Cash or Visa cards.

## 

## Backend: payment models

## 

## 

## 

## 

## Front end: Real-Time Communication Module: Passes information to the driver about total number of booked passengers, remaining seats, and trip status.

## 

## 

## Backend: show seats and booking

## 

## 

## 

## 

## Frontent: Rating & Feedback&about & profile Module: Enables drivers and passengers to rate each other after the trip, helping improve service quality

## 

## 

## 

## 

## Back end : feed back

## 

## 

## 

## 

## 5.3 Challenges faced and how they were resolved

## During the development process, the team faced several challenges, including:

## Integrating GPS functionality: Ensuring accurate location selection and updates.

## Solution: Leveraged built-in Flutter plugins for GPS and maps, tested thoroughly in different scenarios.

## Real-time seat availability and synchronization: Preventing double bookings or incorrect seat reservations.

## Solution: Implemented a robust seat-locking mechanism on the backend using SQL transactions and real-time updates through APIs.

## Secure and smooth payment integration: Ensuring seamless transaction flow with local payment services.

## Solution: Integrated Vodafone Cash and Visa via secure third-party APIs and handled exceptions for failed transactions.

## Gender-based seat filtering: Ensuring that seat selection respects gender preferences.

## Solution: Seats are filtered based on the gender of the passenger, and the backend enforces seat restrictions based on rules.

## Maintaining clear communication between drivers and passengers

## Solution: Provided a dashboard for drivers to view number of booked passengers and seat details, and implemented notifications for trip updates.

# 

# CHAPTER 6

# Testing & Evaluation

## 5.1 Introduction

Testing is done to verify the results by testing each build, including both internal and intermediate builds as well as final versions of the system to be released to external parties” (Booch, Jacobson and Rumbaugh, 2003).

The steps involved are:

* Build the test plans required in each iteration.
* Perform integration and system tests.
* Create test cases to design and implement tests. Test cases define what to test.
* how to test and create executable test components Systematically handle and record test results of each test
* so that the significant defects and be fixed.

**6.1 Testing strategie**

To ensure the reliability, usability, and correctness of the system, various testing strategies were employed throughout the development process:

**Unit Testing**: Each module and function was individually tested, particularly in the backend, to ensure that operations such as booking seats, processing payments, and user registration worked correctly. NUnit was used to perform unit testing in the ASP.NET backend.

**Integration Testing**: The interactions between frontend and backend were tested to ensure smooth data flow, especially for user login, trip creation, and seat reservation.

**User Acceptance Testing (UAT)**: A group of volunteers (students) tested the application on their mobile phones. Feedback was collected about the ease of use, design, speed, and accuracy of booking.

**GPS Functionality Testing:** Tested on different devices and locations to confirm accurate position detection and proper display of maps.

**Payment Testing:** Dummy transactions were executed in sandbox environments to test payment gateway functionality and error handling.

**6.2 Performance metrics**

The following performance metrics were considered to evaluate the efficiency and scalability of the application:

**Accuracy:** Correctness of user information storage, seat booking confirmation, and trip details. The system achieved a booking accuracy rate of over 95% during testing.

**Response Time:** The average response time for booking and trip searches was around 1.2 seconds, ensuring a smooth user experience.

**Scalability**: The system is capable of handling multiple simultaneous users. Stress tests showed that the application remained stable with over 200 concurrent user sessions.

**Error Rate:** Less than 3% during user testing due to handled exceptions in booking, login, and payment scenarios.

**User Satisfaction**: 88% of testers expressed satisfaction with the application’s usability and features.

## 5.2 Test Plan

A software project test plan is a written document that outlines the goals, scope, strategy, and emphasis of a software testing activity. Preparing a test plan is a good approach to think through the steps required to confirm the acceptance of a software product. The finished test plan will assist consumers outside of the test group in comprehending the "why" and "how" of product validation.

* Requirements Analysis: collecting and assessing customer requirements.
* Software Test Plan: defining the scope and goals, developing acceptable testing techniques, developing a software testing strategy, assigning roles and duties, defining resource needs, and establishing start and completion criteria.
* Test Environment: establishing the test infrastructure, determining the testing environment and tools, and installing and configuring the product.
* Test Metrics: a description of the areas to be measured, as well as the creation and collection of metrics
* Test execution is the static and dynamic performance of testing that is given for the use of human and automatic test cases as needed by STP and STS.
* Defect Management (Bug Tracking), documenting testing results, defect description (Problem Reports, Change Requests); defect review and testing results analysis, faults repair, and defect resolution verification
* Status reports, weekly reports, milestone reports, and a closing report are

all examples of reporting

**Test Cases and Results:**

A test case is a series of actions used to verify the correct behavior of an application's functionality or feature. A test case is a document that defines Input, Action, Event, and Expected Response in order to assess whether or not a feature of an application is operating properly

The following tables include observations and summaries of carefully selected test cases and outcomes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **User Login** |  |  |
| **Test case** | **Input** | **Expected result** | **Status** |
| Testing Connection to the Server | Server name, user name, password and database name | Display connection confirmation page | Pass |
| Login – Registered User | Username and password | Home page | Pass |

### Table 6.1 (User Login test)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Request passenger test** | | |
| **Test case** | Input | Expected result | Status |
| Testing user request with all details | user's location, destination, time, car type | Confirmation message: user requested successfully | pass |
| Testing user request with missing details | user's location only, no destination or time | Error message: "Please complete all fields" | pass |
| Testing user request cancellation | Press "Cancel Request" after booking user | request is canceled and removed from the list | Failed |

### Table 6.2 (Request passenger test)

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Driver management test | | |
| Test case | Input | Expected result | Status |
| Testing driver accepting a user request | Driver presses Accept Request | Request status updated to Confirmed | pass |
| Testing driver rejecting a user request | Driver presses "Reject Request | Request returns to available requests list | pass |
| Testing driver logout | Driver presses "Log Out" | Driver becomes unavailable for new requests | failed |

Table 6.3 (Driver management test)

## 5.8 System Testing Process

To guarantee that all parts of the system have been evaluated, test cases are created using a wide range of test data. The inputs designed to test the system are referred to as test data. These test cases are created using the system requirements created during the project's design phase (i.e., knowledge of the program's structure and implementation).

## 5.9 Testing Data

The current manual System collected the system's test data, which was saved in Excel and Word formats.

## 5.6 User Evaluation

The user evaluation is done by using a questioner. The questioner compost of 10 questions to obtain user feedback. Regarding requirement fulfillness, user friendliness, report generation, supporting to decision making

functionality, authentication, interface and response time.

**6.3 Comparison with existing solutions**

The system was compared with existing transportation and ride-sharing platforms such as Swvl, Buseet, and Uber Shuttle:

* Feature 2ala eltaree2 Swvl Buseet
* Gender-based seat filtering ✅ Yes ❌ No ❌ No
* Vodafone Cash Payment ✅ Yes ❌ No ❌ No
* Real-time seat status ✅ Yes ✅ Yes ✅ Yes
* Rating system for both sides ✅ Yes ✅ Yes ✅ Yes
* GPS-based trip planning ✅ Yes ✅ Yes ✅ Yes

Targeted for local cities ✅ Yes ❌ No ❌No

# CHAPTER 7

***Results & Discussion***

## 7 .1 Introduction

## This chapter presents the results obtained after testing and deploying the application in a real-world environment. The findings are analyzed to assess how effectively the system met its goals.

## 7.2 Summary of findings

## The app successfully handled the full lifecycle of a trip, from creation by the driver to booking and payment by passengers.

## GPS location functionality worked accurately in different regions.

## Users were able to book seats and pay using their preferred payment method.

## Drivers received real-time updates on seat reservations and passenger counts.

## The gender-based filtering feature helped improve comfort and trust among users.

## 7.3 Interpretation of results

## Based on testing and user feedback, the application met most of its design objectives:

## The user interface was found to be intuitive and user-friendly.

## Performance was stable even under moderate load.

## Booking accuracy and payment processing worked with minimal errors.

## The implemented rating system improved accountability among users.

## Overall, the project successfully addressed the primary problem of organized intercity travel through technology.

## 7.4 Limitations of the proposed solution

## While the application achieved its core goals, certain limitations still exist:

## Lack of Real-time Chat: There is no in-app chat between drivers and passengers.

## Limited Admin Panel: There is no full-featured web-based admin dashboard for managing content.

## No Trip Cancellation/Refund Feature: Currently, passengers cannot cancel bookings or request refunds.

## Limited Multi-language Support: The app supports only one language at the moment (e.g., Arabic or English).

## Limited Notification System: Push notifications are basic and can be improved for reminders and trip updates.

# CHAPTER 8

***Conclusion & Future Work***

* **8.1 Summary of contributions**
* The project contributes to the field of transportation applications by:
* Developing a user-friendly mobile application using Flutter.
* Implementing a reliable backend system using ASP.NET and SQL Server.
* Providing gender-based seat booking for cultural compatibility.
* Integrating with local payment gateways like Vodafone Cash.
* Supporting GPS-based trip planning and real-time driver-passenger communication.
* Enabling post-trip ratings and reviews to improve service quality.
* **Possible improvements or extensionsFor future versions of the application.**
* several improvements can be made:
* **In-app chat feature**: Allow real-time communication between drivers and passengers.
* **Trip cancellation and refund system**: Add a structured policy and backend logic for trip changes.
* **Admin Web Portal**: Develop a dedicated dashboard for admins to manage users, trips, reports, and performance analytics.
* **Advanced Notifications**: Integrate Firebase Cloud Messaging for rich, interactive notifications.
* **Multi-language Support**: Allow users to switch languages according to preference.
* **AI-based trip recommendations**: Suggest trips based on user history and preferences.

# References

# 1. UI figma tools and Google Developers. (n.d.). Flutter Documentation. Retrieved from https://flutter.dev

# 2.Microsoft Docs. (n.d.). ASP.NET Core Documentation. Retrieved from https://docs.microsoft.com

# 3.SQL Server Docs. (n.d.). Microsoft SQL Documentation.

# 4.Swvl Official App. (2024). Play Store Page.

# 6. Egypt. (n.d.). Vodafone Cash API.