

# **ParkZone - Parking Management System**

## **Project Partners:**

1. Sejal Warke
2. Vaishnavi Bhosale
3. Pratham Shah

# INDEX

<b>Sr.No.</b>	<b>Title</b>	<b>Page No.</b>
1.	Introduction	
2.	Problem Definition & Scope	
2.1	Problem Definition	
2.2	Goals & Objectives	
2.3	Major Constraints & Outcomes	
3	Software Requirements Specification	
3.1	Proposed System	
3.2	Scope	
4	System Modules	
5	Performance Requirements	
5.1	H/W Requirements & S/W Requirements	
6	UML Diagrams	
6.1	DFD	
6.2	ERD	
6.3	Use Case Diagram	
6.4	Class Diagram	
6.5	Sequence Diagram	

6.6	Activity Diagram	
6.7	Deployment Diagram	
6.8	System Architecture	
7	Test Cases	
8	Results	
8.1	Login Page	
8.2	Registration Page	
8.3	Admin Dashboard	
8.4	User Dashboard	
8.5	Available Parking Spaces	
8.6	Parking Slot Confirmation Mail	
9	References	

# 1. INTRODUCTION

The ParkZone is a web-based application designed to streamline and automate the process of managing vehicle parking within different areas. It provides a structured and efficient solution for both administrators and users by handling reservations, tracking vehicle entries and exits, allocating parking spots, and calculating parking fees. With features such as real-time availability updates, automated session tracking, email notifications, and detailed reporting, the system enhances operational efficiency and improves the user experience.

## **Motivation:**

The primary motivations behind the development of ParkZone include:

- The increasing number of vehicles in urban areas leading to parking challenges.
- The need for a technology-driven solution that minimizes human intervention.
- Enhancing user convenience by providing an advanced reservation system.
- Implementing an eco-friendly approach by optimizing parking lot usage.

## 2. Problem Definition And Scope

### 2.1 Problem Definition

In modern cities, the rapid increase in the number of vehicles has led to significant challenges in managing parking spaces effectively. Manual parking systems are inefficient, time-consuming, and prone to human error. They often result in issues such as overcrowded parking lots, difficulty in locating available spots, revenue leakage, and customer dissatisfaction.

### 2.2 Goals and Objectives

The main objectives of ParkZone are as follows:

- **User-Friendly Interface:** Develop an intuitive and responsive front-end application
- **Real-Time Availability Updates:** Provide dynamic updates on parking spot availability
- **Email & Notification Alerts:** Enhance user engagement through instant notifications
- **Secure Authentication:** Implement JWT-based authentication for improved security.
- **Efficient Parking Reservation:** Allow users to book spots in advance with minimal effort.

### 2.3 Major Constraints & Outcomes

- Requires consistent synchronization between spot availability and UI
- Must use React.js for frontend, Spring boot, Dotnet for backend, and MySQL for the database.
- Parking spot availability must be updated in real-time to prevent double-booking.

- Admins can manage locations, parking spots, and monitor user/reservation data

## **3. Software Requirement Specification**

### **3.1 Proposed System**

The proposed system is a web based application which is designed to automate the parking spaces. By integrating dotnet the mail confirmation of booked slot, start time and end-time of the parking slot is sent to the user. This project represents an important step through online pre booking of parking. This is mainly designed for the problem that is arised due to the increasing number of vehicles in modern cities, and due to this the lack of parking places is observed.

### **3.2 Scope**

The scope of this project encompasses the development of a comprehensive parking management system designed to:

- Manage parking space allocation efficiently by tracking availability and occupancy in real-time.
- Allow users to reserve, book, and release parking slots through a user-friendly interface.
- Monitor and enforce parking rules such as booking duration limits, payment status, and slot release timing.
- Provide real-time updates and notifications to users regarding their bookings and slot availability.

- Integrate multiple user roles, including users, administrators, and security personnel, to streamline operations.



## **4. System Modules**

### **User Authentication Module**

Handles user and admin registration, login, and session management Includes password hashing, validation, and optional password reset.

### **Location & Spot Management Module**

Allows admins to add, update, and delete parking locations and spots Users can view locations and check spot availability.

### **Reservation Module**

Lets users select a time slot and reserve a spot Prevents double-booking Can include timers or auto-expiry features

## 5. H/W Requirements & S/W Requirements

### 5.1 H/W Requirements

- Minimum 4 GB RAM
- Processor: i5 or equivalent

### 5.2 S/W Requirements

- Frontend: React.js.

**Dynamic Dashboard** with real-time parking status updates.

**Interactive** slot selection.

**User Authentication** using JWT.

- Backend: Springboot, Dotnet for API development.

**User Management:** Registration, authentication, and role-based access.

**Parking Slot Management:** Real-time updates, additions, and deletions.

**Booking System:** Processing reservations and cancellations.

**Email Notification Service:** Sending automated booking confirmations.

**Session Management:** Secure handling of active user sessions.

- Database: MySQL for data storage and management.

The **MySQL database** stores all application-related data, including:

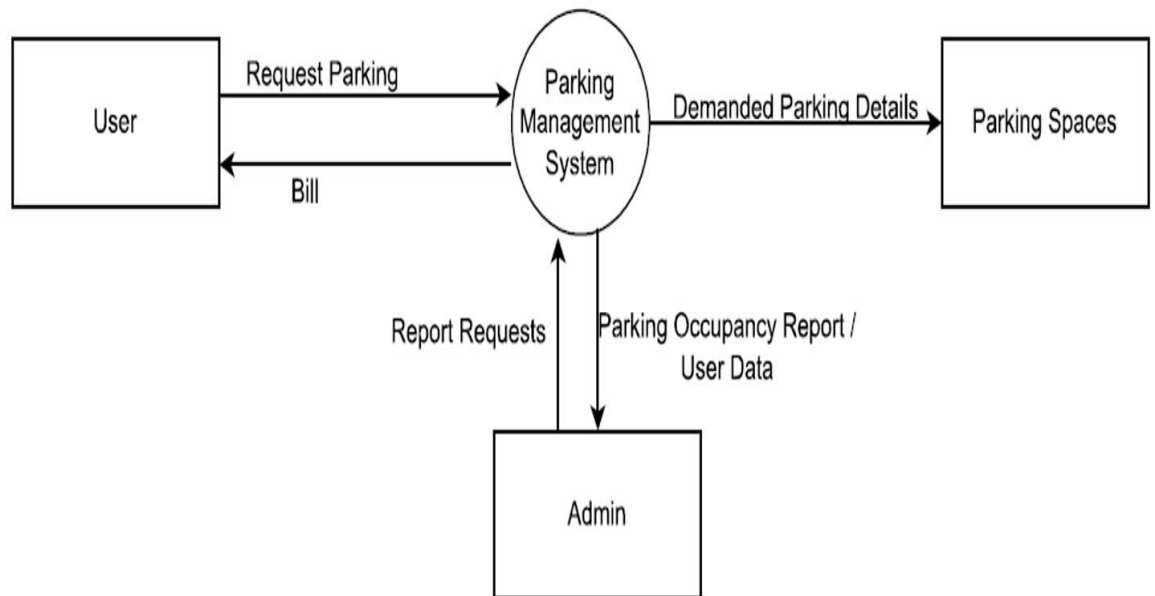
**User Information:** Profile, contact details, and authentication tokens.

**Parking Slot Data:** Slot availability, reservations, and pricing.

**Reservation History:** Reservation details for users and administrators.

## 6. UML Diagrams

### 6.1 DFD Diagram



**fig 6.1 DFD Level 0**

## 6.2 ER Diagram

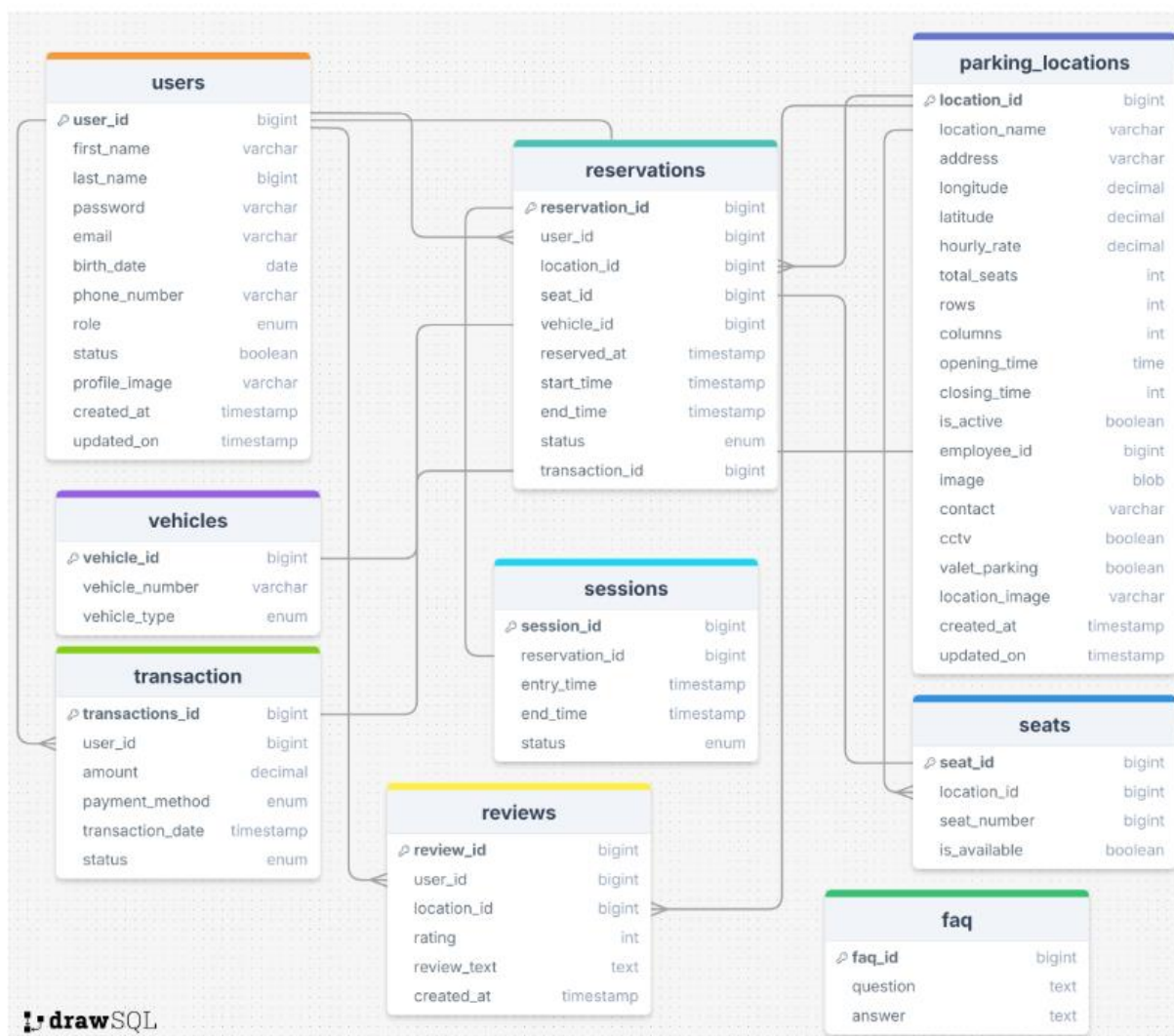


fig 6.2 ER Diagram

## 6.3 Use Case Diagram

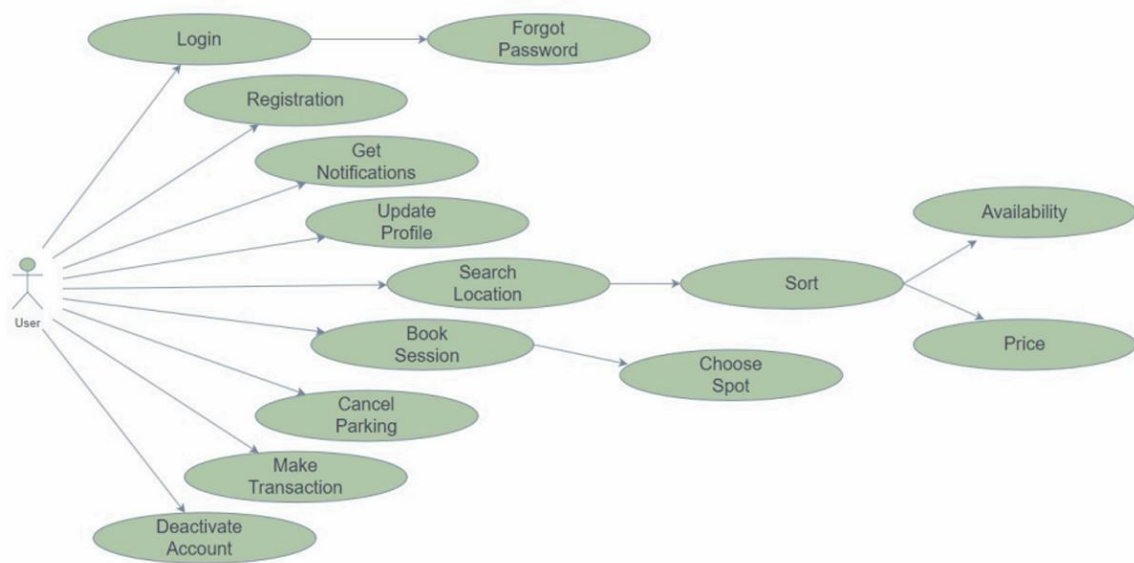
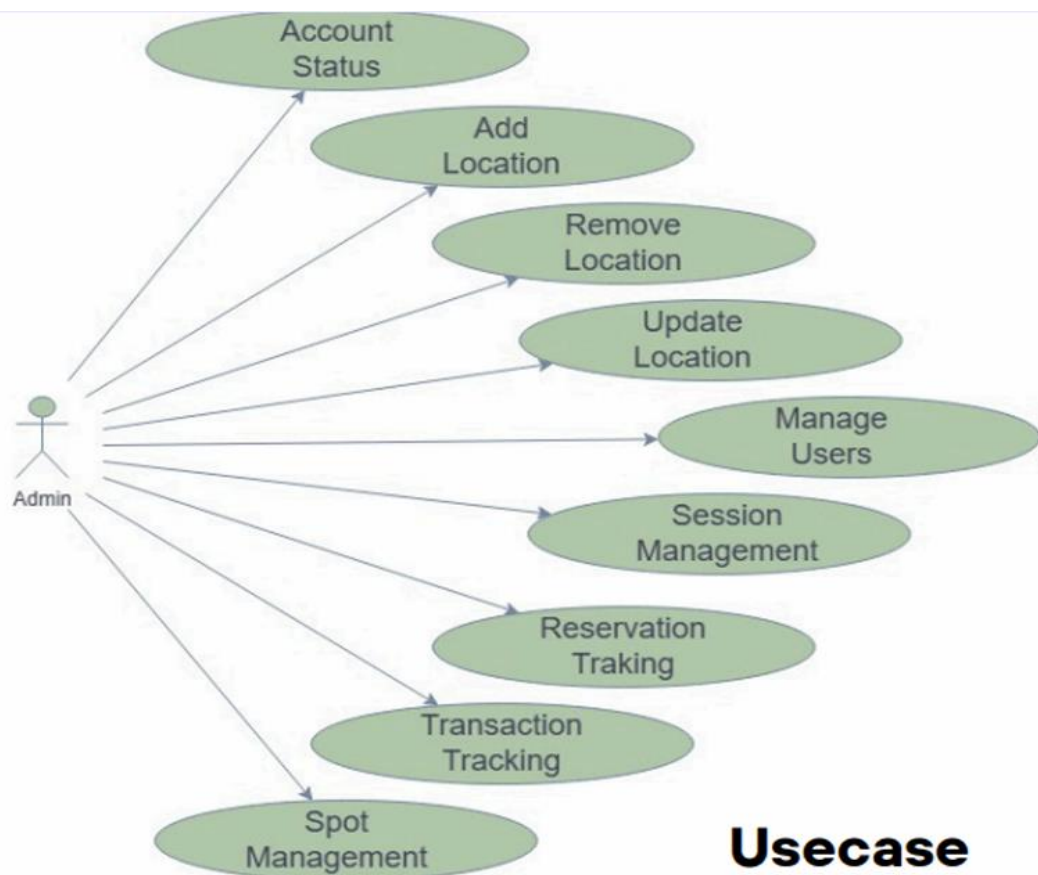


fig 6.3.1 UseCase Diagram



**Usecase**

fig 6.3.2 Usecase Diagram

## 6.4 Class Diagram

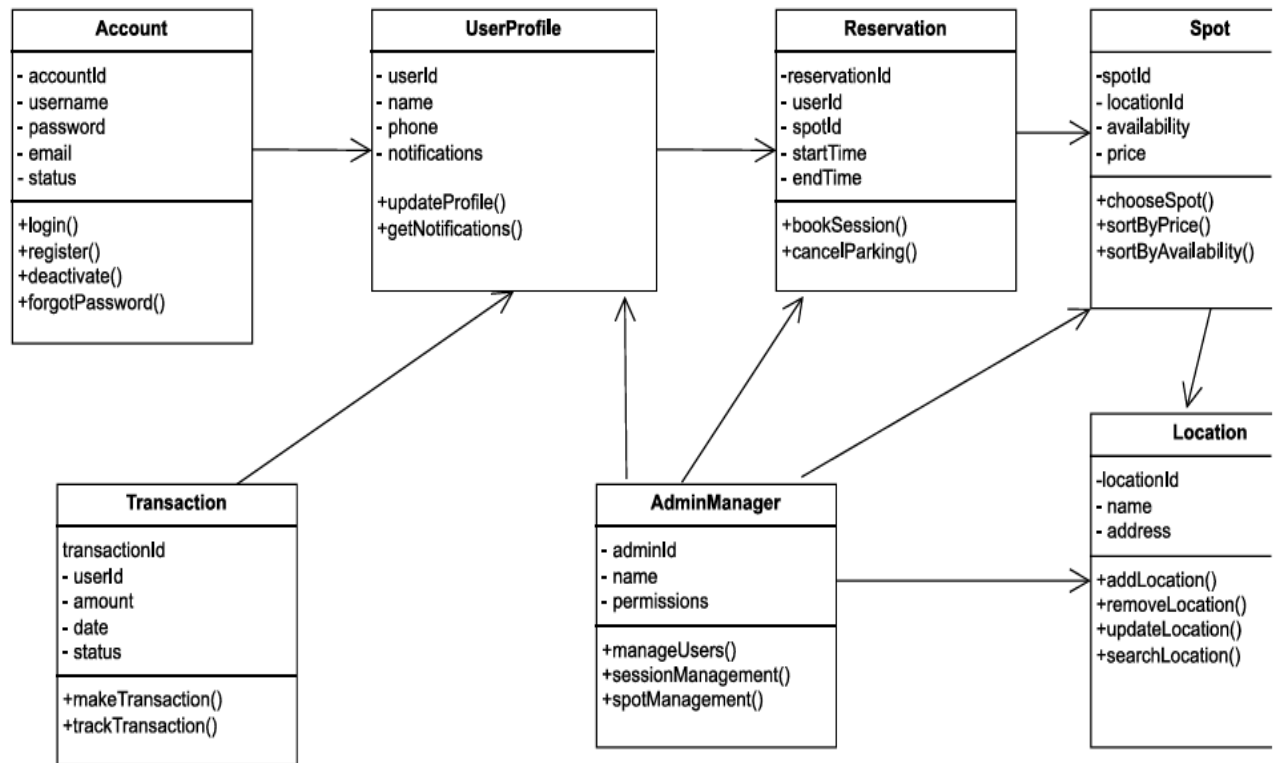


fig 6.4 Class Diagram

## 6.5 Sequence Diagram

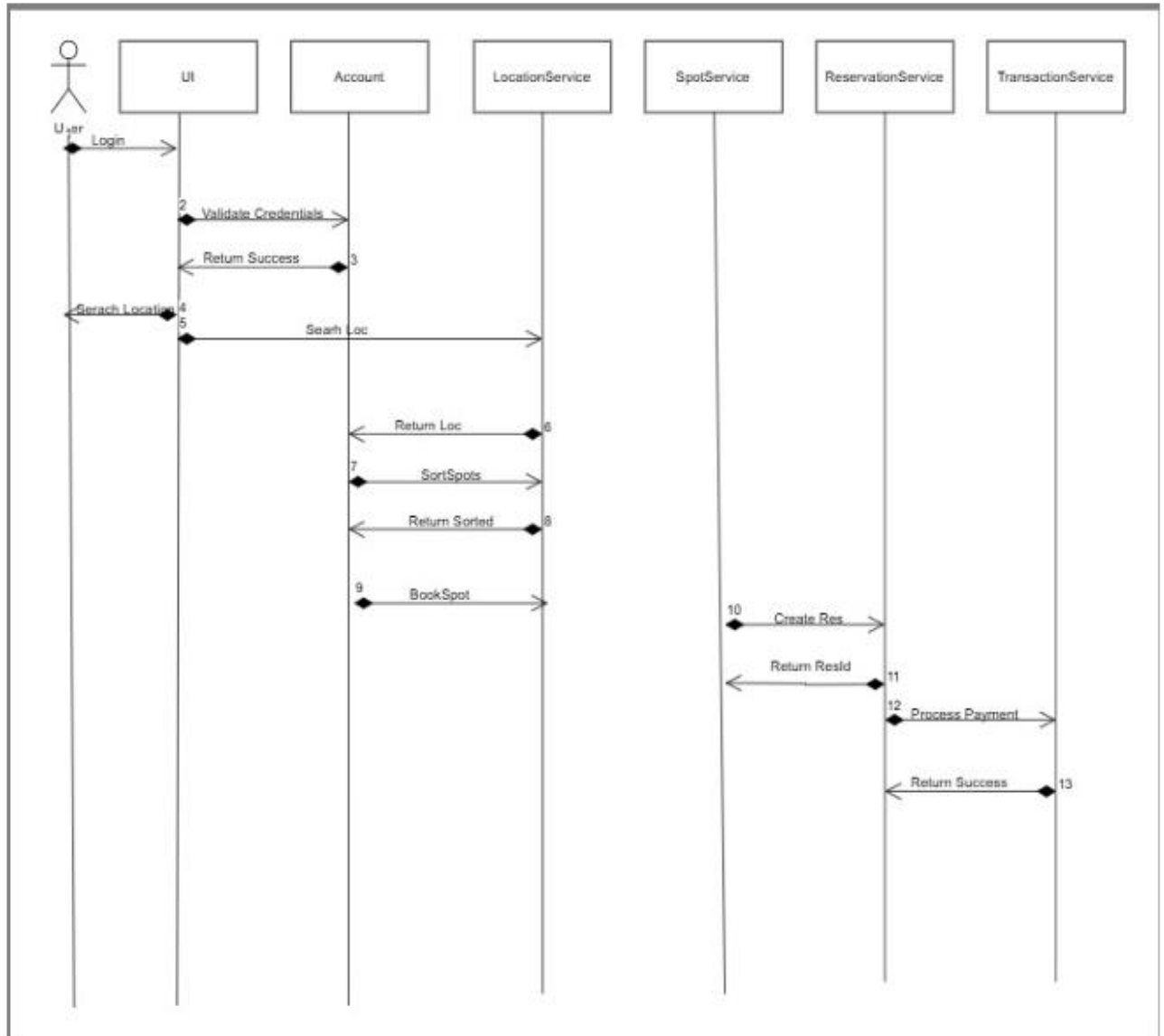


fig 6.5 Sequence Diagram



## 6.6 Activity Diagram

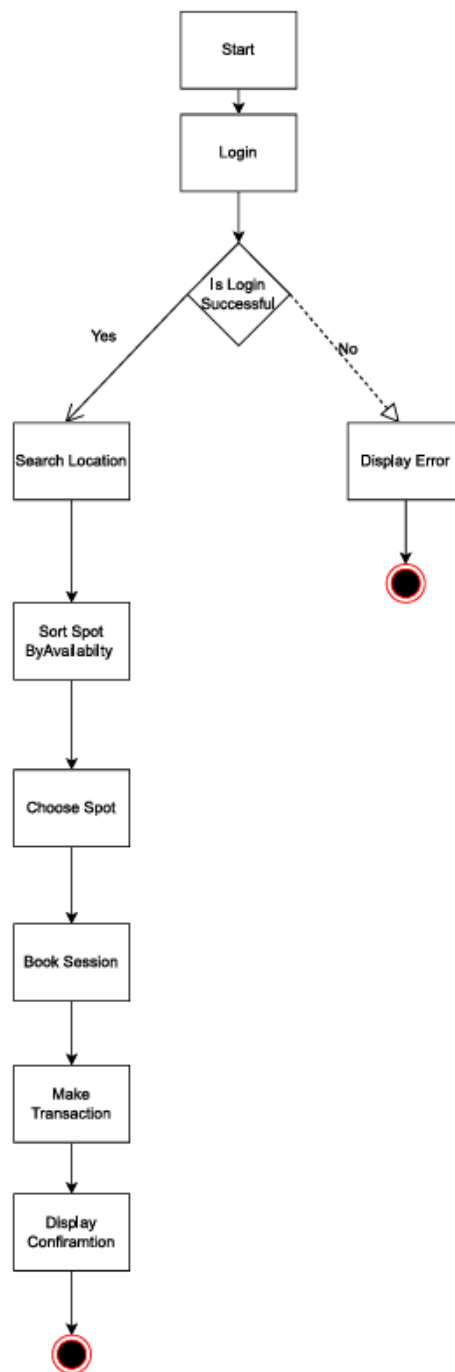
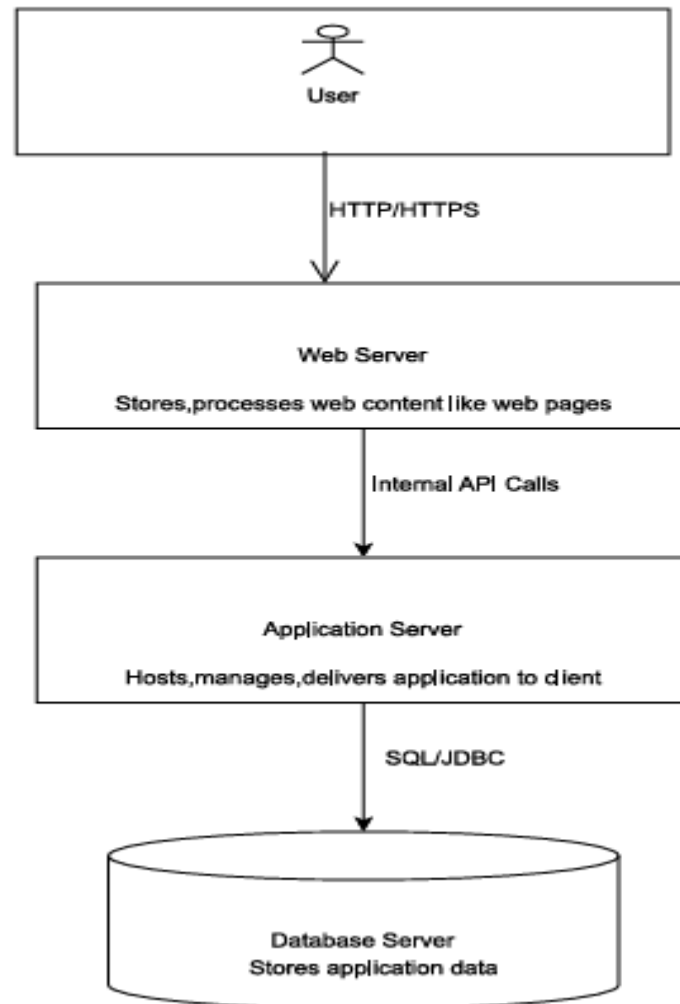


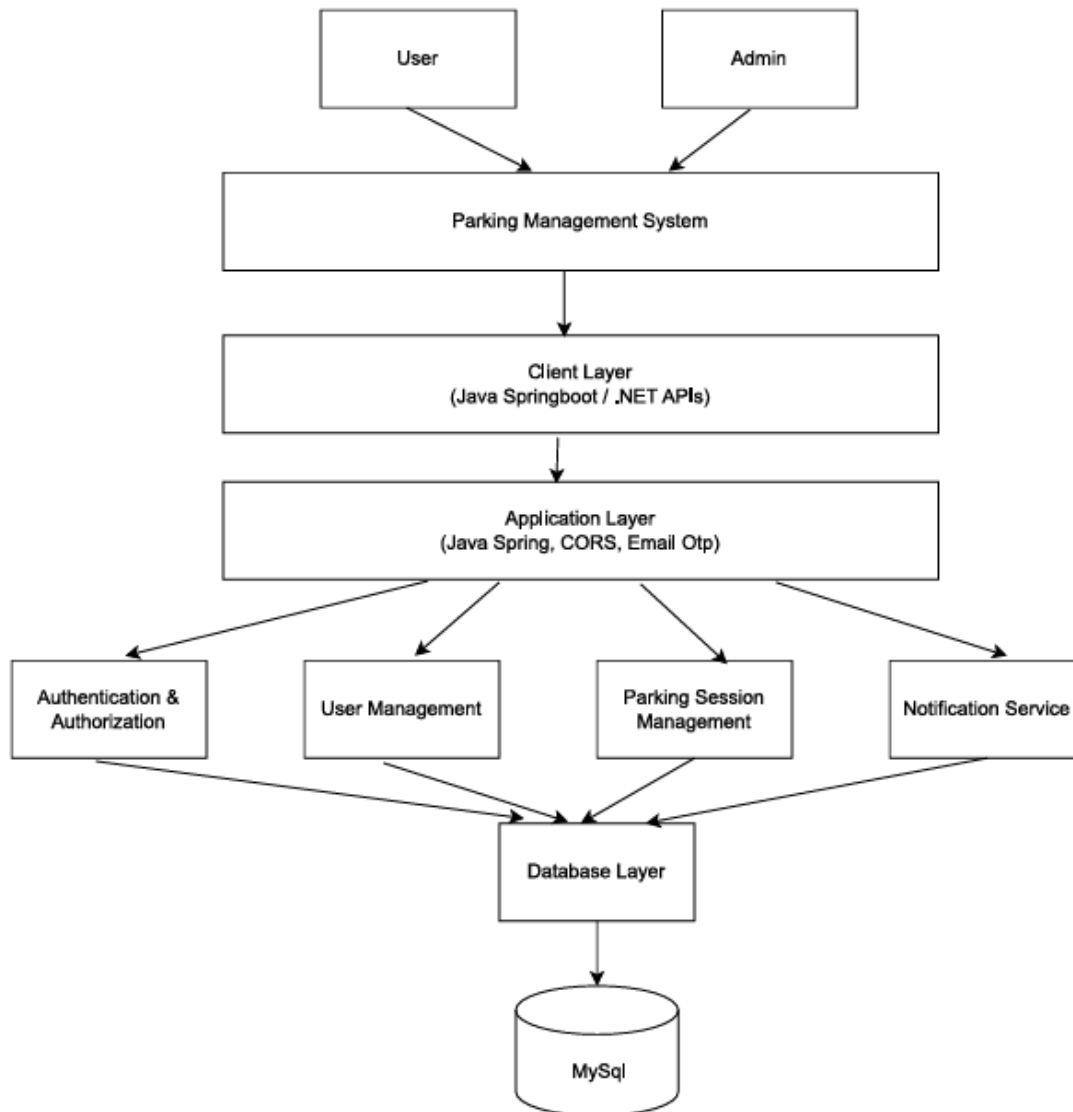
fig 6.6 Activity Diagram

## 6.7 Deployment Diagram



**fig 6.7 Deployment Diagram**

## 6.8 System Architecture



**fig 6.8 System Architecture**

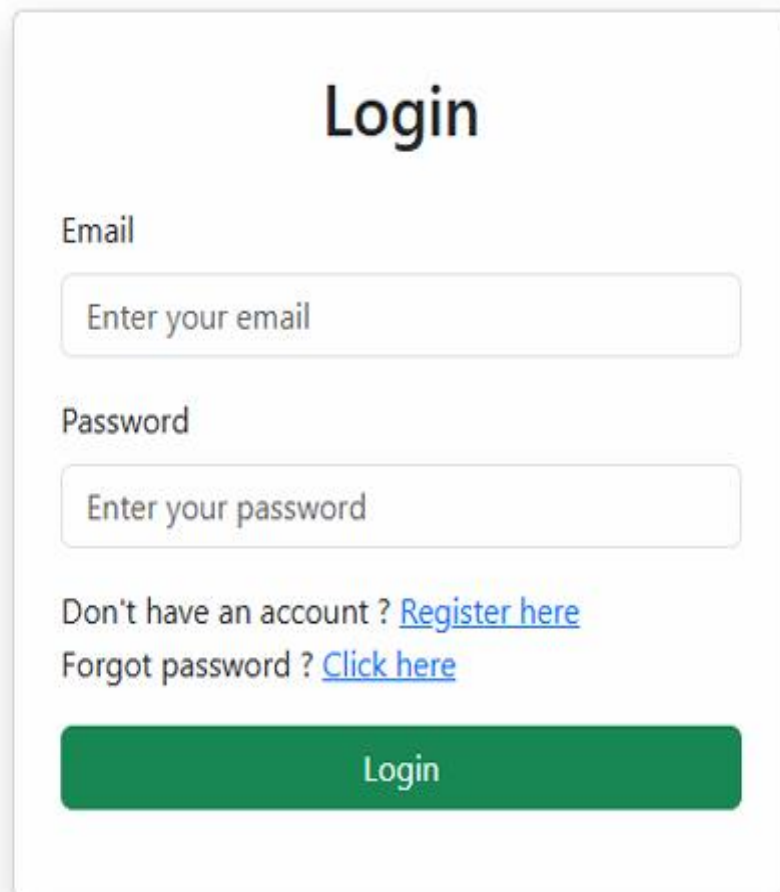
## 7. Test Cases

TestCase ID	Description	Expected Result	Status
TC01	Login with valid credentials	Login Successful	Pass
TC02	Login with invalid credentials	Login Unsuccessful	Fail
TC03	Search Parking Spaces	Display Parking Details	Pass
TC04	Book Parking	Parking Confirmed	Pass

**fig 7.1 Test Cases**

## 8. RESULT

### 8.1 Login Page



The image shows a login form with a white background and a subtle shadow. At the top, the word "Login" is centered in a large, bold, black font. Below it, there are two input fields. The first is labeled "Email" and contains the placeholder text "Enter your email". The second is labeled "Password" and contains the placeholder text "Enter your password". Below these fields, there are two lines of text: "Don't have an account ? [Register here](#)" and "Forgot password ? [Click here](#)". At the bottom, there is a green button with the text "Login" in white.

Login

Email

Enter your email

Password

Enter your password

Don't have an account ? [Register here](#)

Forgot password ? [Click here](#)

Login

**fig 8.1 Login Page**

## 8.2 Registration Page

**Register**

First Name

Last Name


Email


**Send OTP**

Phone Number

Password

Confirm Password

Birth Date  

Role  

**Signup**

fig 8.2 Registration Page

## 8.3 Admin Dashboard

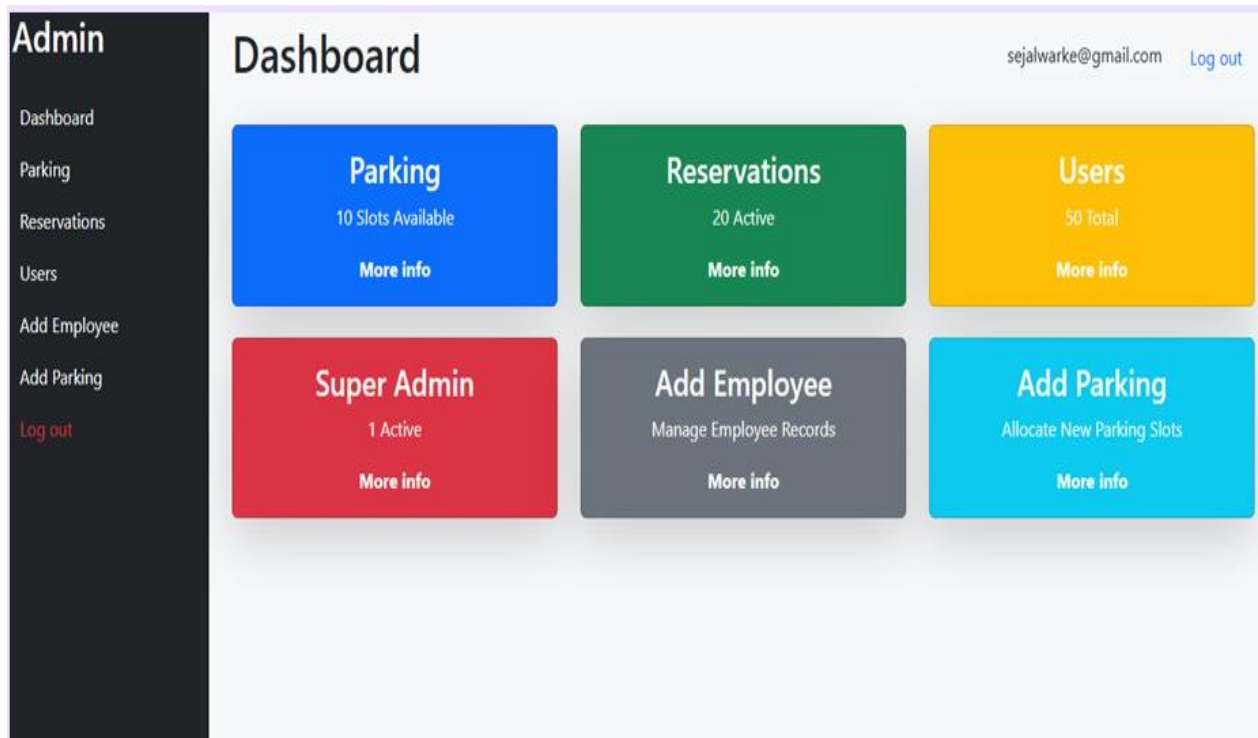


fig 8.3 Admin Dashboard

## 8.4 User Dashboard

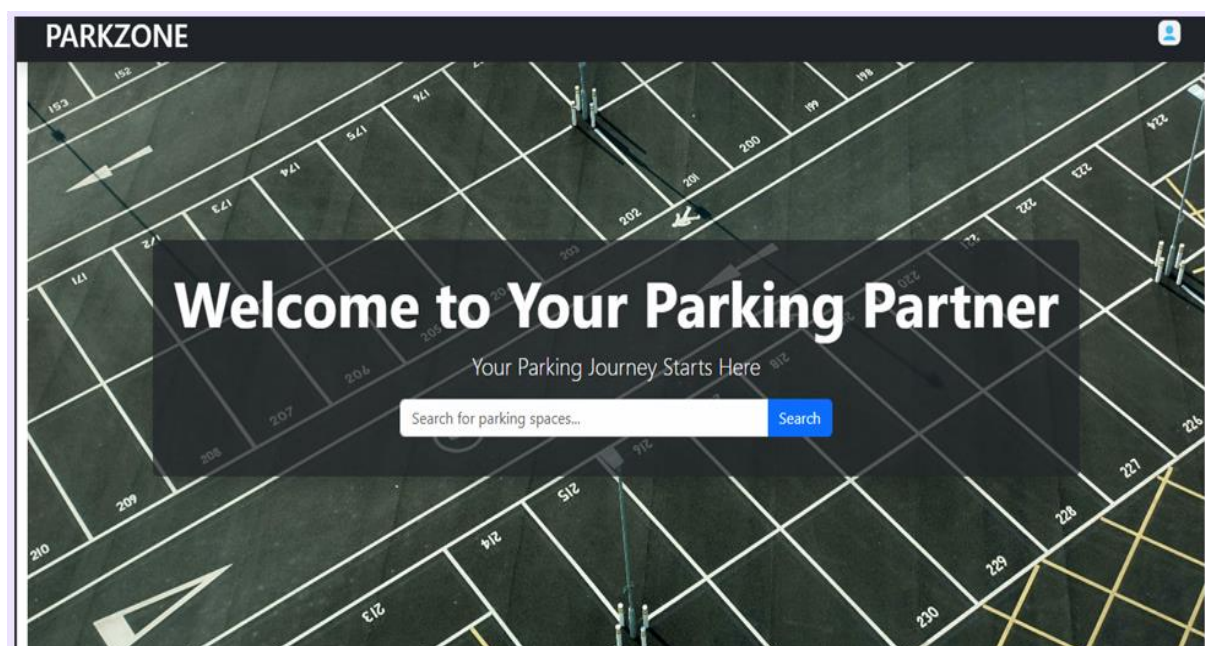


fig 8.4 User Dashboard

## 8.5 Available Parking Spaces

Search Results			
Image	Location Name	Address	Hourly Rate
	Mumbai	YCP	₹20/hr

**fig 8.5 Available Parking Spaces**

## 8.6 Parking Slot Confirmation Mail

Dear Shruti,

Thank you for choosing our parking services!

**Here are the details of your booking:**

Reservation ID:	5
Parking Location:	Mumbai
Location Address:	YCP
Slot Start Time:	2025-08-10 19:41:00
Slot End Time:	2025-08-10 20:45:00

Please keep this email handy.

We look forward to seeing you.

Sincerely,

ParkZone Pvt. Ltd.

**fig 8.6 Parking Slot Confirmation Mail**



## 9. REFERENCES

1. Ahmed, S., Khan, M. (2022). A reinforcement learning approach for optimal parking space allocation in a smart parking reservation system. *IEEE Transactions on Intelligent Vehicles*, 7(4), 678-690.
2. Gupta, R., Sharma, S. (2022). Block chain-enabled secure and trans- parent parking reservation system. *2022 IEEE International Conference on Block chain*, pp. 123-128.
3. SCHOLTZ, Bauke; TIJMS, Arjan. *The Definitive Guide to Jakarta Faces in Jakarta EE 10: Building Java-based Enterprise Web Applications*. Springer, 2022.
4. KARNIŠ, Filip. *Shongo Reservation System Backend REST API*. 2022. Bachelor's thesis. Masaryk University.

