RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

CS19442 SOFTWARE ENGINEERING CONCEPTS LAB

Laboratory Record

Name :………………………. …………………………...

Year / Branch / Section : ……….. ………………………………

University Register No. : …………….. ……………………………………

College Roll No. : ……………………..…………………………………

Semester : ……………………………………..…………………………………

Academic Year : ………………………………………………………

RAJALAKSHMI ENGINEERING COLLEGE RAJALAKSHMI

NAGAR, THANDALAM – 602 105BONAFIDE

CERTIFICATE

Name:____

Academic Year:____________Semester:_____Branch:___________

Register No: □

Certified that this is the bonafide record of work done by the above student in the CS19442-Software Engineering Concepts Laboratory during the year 2023- 2024

Signature of Faculty-in-charge

---

Submitted for the Practical Examination held on ___________________

Internal Examiner External Examiner

SOFTWARE ENGINEERING & CONCEPTS

OVERVIEW OF THE PROJECT

What is the Problem it is Trying to Resolve:

Cost Inefficiencies in Parking:

Traditional parking systems are often inefficient, leading to increased operational costs for parking facility owners and additional expenses for users. Users spend valuable time and fuel searching for parking spaces, which increases their costs and leads to revenue loss for parking facility owners due to underutilized spaces.

Poor Customer Experience in Urban Parking:

Urban parking is frustrating due to the lack of visibility into available parking spaces, resulting in time-consuming searches and uncertainty for drivers.Drivers face challenges in locating convenient and affordable parking spaces, leading to a negative customer experience

Productivity Loss Due to Parking Challenges:

Inefficient parking systems contribute to traffic congestion, increased vehicle emissions, and decreased urban productivity.Time spent searching for parking spaces and the resulting traffic congestion impact overall urban productivity.

Underutilization of Parking Spaces:

Parking spaces may go underutilized due to a lack of visibility, leading to missed revenue opportunities for parking facility owners.Without a centralized platform providing real-time information, parking spaces may remain vacant even during peak demand periods.

User Data:

User registration and authentication details (personal information, email, and social media accounts).Location data to provide accurate parking space information and navigation assistance.

Parking Data:

Real-time availability of parking spaces using sensors, parking management systems, and user input.Details on parking spaces including location, price, type (street parking, garage, etc.), and reservation status.

Transaction Data:

Payment information and transaction history for processing payments within the app.Reservation details to manage and confirm bookings for parking spaces.

Analytics Data:

Data on user behavior, such as frequently searched locations, peak times for parking demand, and common preferences.Utilization rates of parking spaces to optimize space management and identify underutilized areas.

How it Would Help Users When We Implement the System

For Users:

Enhanced Convenience: Users can easily find and reserve parking spaces in advance, reducing the time and effort spent searching for parking.

Cost Savings: By providing real-time information on available spaces, users can avoid driving around looking for parking, saving fuel and reducing expenses.

Improved Experience: The app provides a user-friendly interface with real-time availability updates, easy reservation options, and navigation assistance, improving the overall parking experience.

For Parking Facility Owners:

Increased Revenue: By maximizing the utilization of parking spaces, facility owners can generate optimal revenue and reduce vacancies.

Operational Efficiency: The app helps in managing parking spaces more efficiently, reducing operational costs and enhancing productivity.

Data-Driven Insights: Facility owners can gain insights into parking trends and user behavior, helping them make informed decisions on pricing, space allocation, and improvements.

For Urban Areas:

Reduced Traffic Congestion: Efficient parking space utilization can help in reducing traffic congestion caused by drivers searching for parking.

Lower Emissions: By minimizing the time spent driving around looking for parking, the app contributes to lower vehicle emissions, promoting a greener environment.

Enhanced Productivity: Reducing parking-related challenges can lead to improved overall urban productivity, benefiting businesses and residents alike.

BUSINESS ARCHITECTURE DIAGRAM

Manual Process:

Drivers: Typically, drivers manually search for parking spaces by driving around their desired destination. They rely on visual cues and previous knowledge of the area. This often leads to frustration, wasted time, and additional fuel costs.

Parking Facility Owners: Facility owners manually manage parking spaces, monitoring availability through on-site staff or basic counting methods. This approach is prone to errors and inefficiencies, leading to underutilization of spaces and revenue loss.

Automatic Process:

Drivers: Some advanced parking facilities have digital signage indicating available spaces, but these are limited to specific locations and often lack real-time updates.

Parking Facility Owners: Some use automated systems like sensors and cameras to track space availability. However, these systems are often isolated, not integrated with user-facing applications, limiting their utility.

Current Process: Manually search for parking, often leading to time-consuming and frustrating experiences.

Future Process with App: Use the app to find available parking spaces in real-time, make reservations, navigate to the space, and pay within the app, leading to a seamless and efficient experience.

Parking Facility Owners:

Current Process: Manage parking manually or with isolated automated systems, leading to inefficiencies and underutilization.

Future Process with App: Integrate their parking facilities with the app to provide real-time availability, manage reservations, and optimize space utilization, leading to increased revenue and operational efficiency.

Urban Planners and City Officials:

Current Process: Address parking and traffic issues through broad measures without real-time data, leading to suboptimal solutions.

Future Process with App: Use data from the app to understand parking patterns, optimize urban planning, and implement policies that reduce congestion and emissions, improving overall city productivity and environment.

Business Problems

For Drivers:

Time Wastage: Significant time spent searching for parking, especially in busy urban areas.

Fuel Costs: Increased fuel consumption due to extended driving while searching for parking.

Stress and Frustration: Poor parking experiences leading to dissatisfaction and stress.

For Parking Facility Owners:

Underutilization of Spaces: Inability to effectively utilize all parking spaces, leading to lost revenue.

Operational Inefficiencies: High operational costs due to manual management and lack of integrated systems.

Revenue Loss: Missed opportunities to monetize available parking spaces due to lack of real-time data and reservations.

For Urban Areas:

Traffic Congestion: Increased congestion due to drivers searching for parking, affecting overall traffic flow.

Environmental Impact: Higher vehicle emissions resulting from prolonged searching for parking spaces.

Reduced Productivity: Lower overall urban productivity due to parking-related delays and inefficiencies.

Proposed Solution: Parking Space Finder App

How It Addresses the Business Problems:

For Drivers:

Time Efficiency: Real-time information on parking availability reduces the time spent searching for spaces.

Cost Savings: Reduced fuel consumption due to more efficient parking searches.

Improved Experience: Enhanced user experience through easy reservations, payment integration, and navigation assistance.

For Parking Facility Owners:

Optimal Utilization: Real-time tracking and reservations maximize space utilization.

Operational Efficiency: Automated systems reduce the need for manual monitoring and management.

Increased Revenue: Better utilization and reservation management increase revenue opportunities.

For Urban Areas:

Reduced Congestion: Efficient parking searches reduce the time vehicles spend on the road, decreasing congestion.

Environmental Benefits: Lower emissions due to reduced driving while searching for parking.

Enhanced Productivity: Improved traffic flow and parking efficiency contribute to higher urban productivity.

REQUIREMENTS AS USER STORIES FOR BUSINESS ARCHITECTURE DIAGRAM

USER STORIES

User Registration:

As a new user,

I want to register for an account,

So that I can access the features of the parking space finder app.

View Available Parking Spaces:

As a registered user,

I want to view available parking spaces in a specific area,

So that I can choose a parking spot conveniently.

Filter Parking Spaces

As a user,

I want to filter available parking spaces by criteria such as location, price, and type,

So that I can find the most suitable parking option for my needs.

Real-Time Availability Updates

As a user,

I want to receive real-time updates on parking space availability,

So that I can make informed decisions about where to par

Reserve a Parking Space

As a user,

I want to reserve a parking space in advance,

So that I can be assured of a parking spot upon arrival

Payment for Parking Space

As a user,

I want to pay for my parking space through the app,

So that I can complete my transaction seamlessly.

Navigation Assistance to Parking Space

As a user,

I want to get navigation assistance to my reserved parking space,

So that I can reach the location without difficulty.

Cancellation of Reservation

As a user,

I want to cancel my parking space reservation if needed,

So that I can avoid unnecessary charges.

View Parking History

As a user,

I want to view my parking history,

So that I can keep track of my past parking transactions and receipts.

Leave Reviews and Ratings

As a user,

I want to leave reviews and ratings for parking spaces I have used,

So that I can help other users make informed decisions.

Receive Notifications and Alerts

As a user,

I want to receive notifications and alerts about my reservation status and nearby available parking spaces,

So that I can stay updated and adjust my plans accordingly.

Estimates Using Poker Planning Methodology

User Registration: 3 points

View Available Parking Spaces: 5 points

Filter Parking Spaces: 3 points

Real-Time Availability Updates: 8 points

Reserve a Parking Space: 5 points

Payment for Parking Space: 5 points

Navigation Assistance to Parking Space: 8 points

Cancellation of Reservation: 3 points

View Parking History: 3 points

Leave Reviews and Ratings: 5 points

Receive Notifications and Alerts: 8 points

NON FUNCTIONAL REQUIREMENTS:

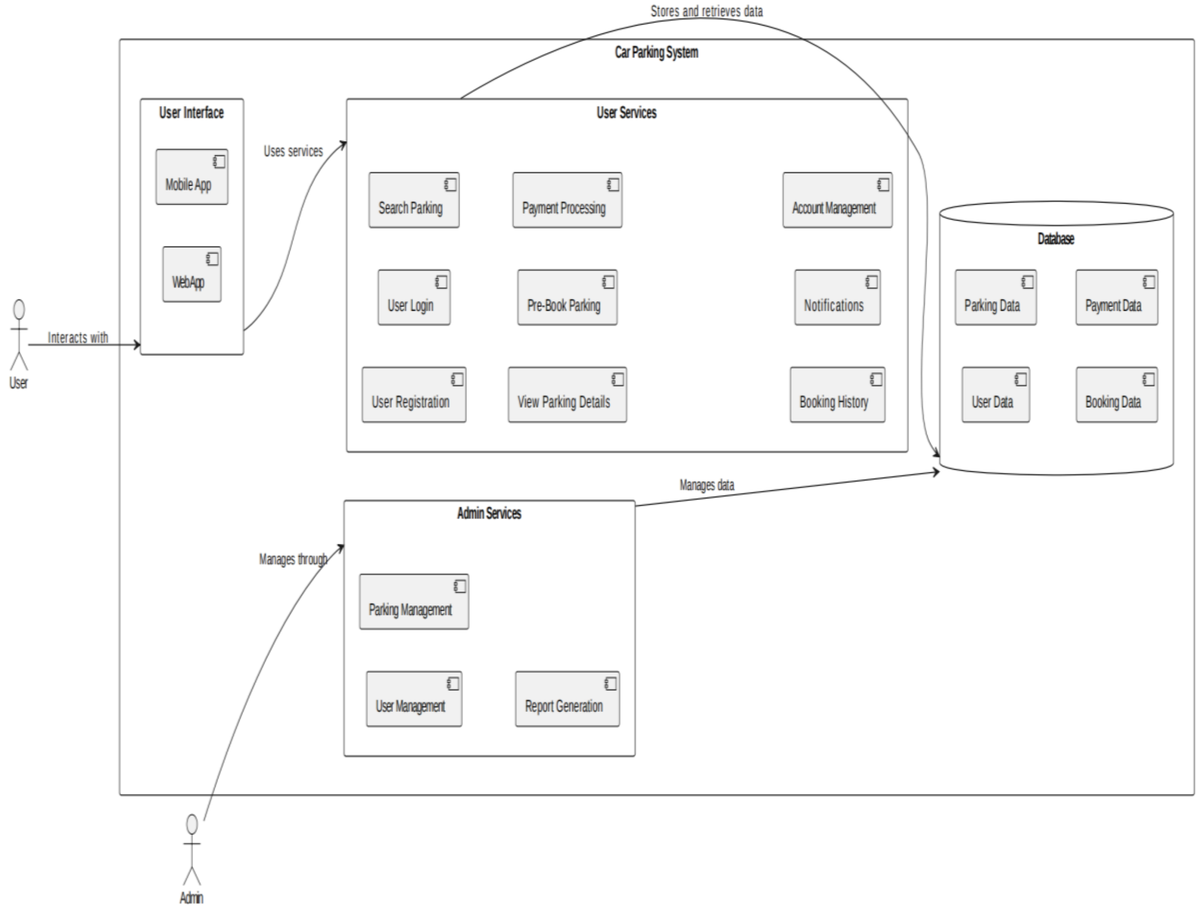
Performance: The system should provide real-time updates on parking space availability within 2 seconds of the query

Security : The system must ensure that all user data, including payment information, is encrypted and securely stored, adhering to industry-standard security practices.

Scalability: The system should be able to handle up to 10,000 simultaneous users without performance degradation.

These user stories, estimates, and non-functional requirements provide a comprehensive outline for developing the parking space finder app, ensuring it addresses key functionalities, user needs, and system performance expectations.

BUSINESS ARCHITECTURE DIAGRAM:



1. Modules and Their Interactions

Modules:

User Interface (UMicroservicesandles user interactions.

Forwards user requests to the appropriate backend services.

API Gateway:

Acts as an entry point for all clients.

Routes requests to the appropriate microservices.

Handles rate limiting, authentication, and authorization.

Authentication Service:

Manages user authentication.

Issues JWT tokens upon successful login.

User Service:

Manages user data and profiles.

Interacts with the database to store and retrieve user information.

Product Service:

Manages product catalog.

Handles CRUD operations for products.

Order Service:

Manages orders.

Handles order creation, updates, and cancellations.

Payment Service:

Processes payments.

Interacts with third-party payment gateways.

Notification Service:

Sends notifications (email, SMS) to users.

Logging Service:

Centralized logging for all services.

Stores logs for auditing and monitoring.

Error Handling Module:

Centralized error handling mechanism.

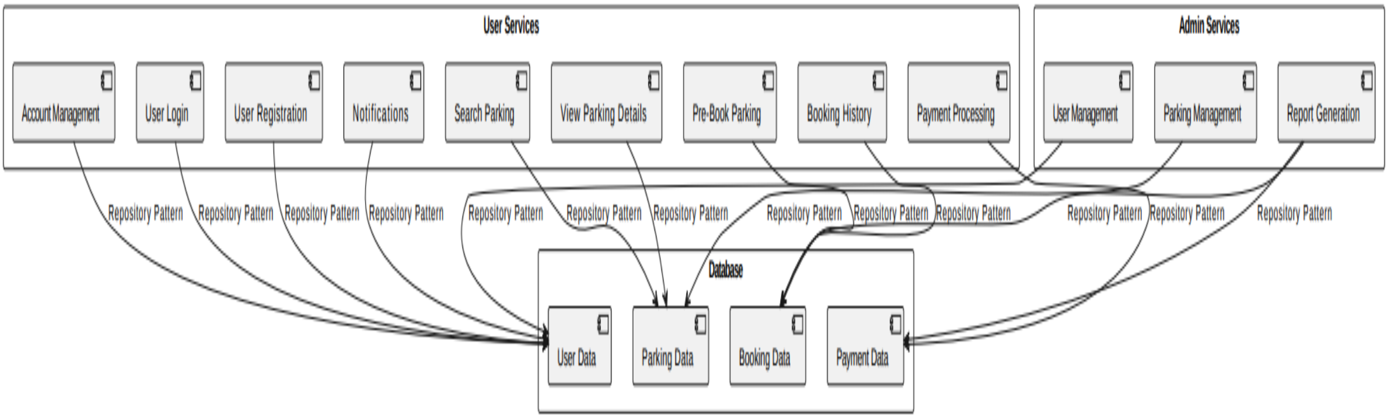
Provides consistent error responses.

Database:

Centralized data storage.

Different databases for different services if needed (SQL/NoSQL).

ARCHITECTURAL PATTERNS



Pattern Used:

Microservices Architecture

Scalability: Each service can be scaled independently.

Maintainability: Easier to maintain and update small, focused services.

Flexibility: Technology stack can vary between services.

Isolation: Fault isolation improves the reliability of the system.

Design Principles

Single Responsibility Principle (SRP):

Each microservice has a single responsibility, making the system easier to understand and manage.

Open/Closed Principle (OCP):

Services are open for extension but closed for modification, enabling easier upgrades and feature additions.

Interface Segregation Principle (ISP):

Services expose specific interfaces rather than a general-purpose interface, improving the clarity and usability of the API.

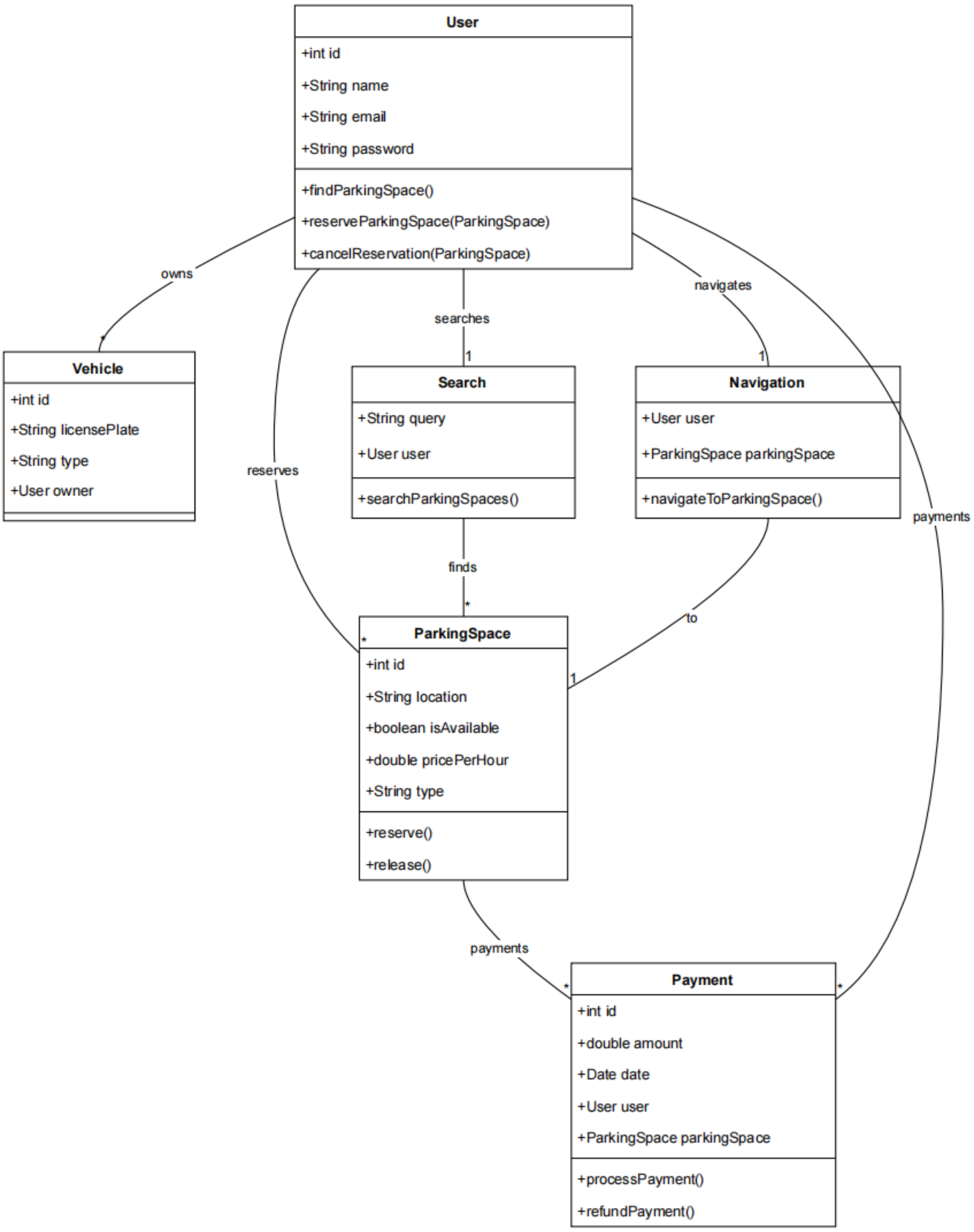
Dependency Inversion Principle (DIP):

Services depend on abstractions rather than concrete implementations, enhancing flexibility and testability.

Separation of Concerns:

Clear separation between different modules (UI, API Gateway, Microservices, Database).

CLASS DIAGRAM:

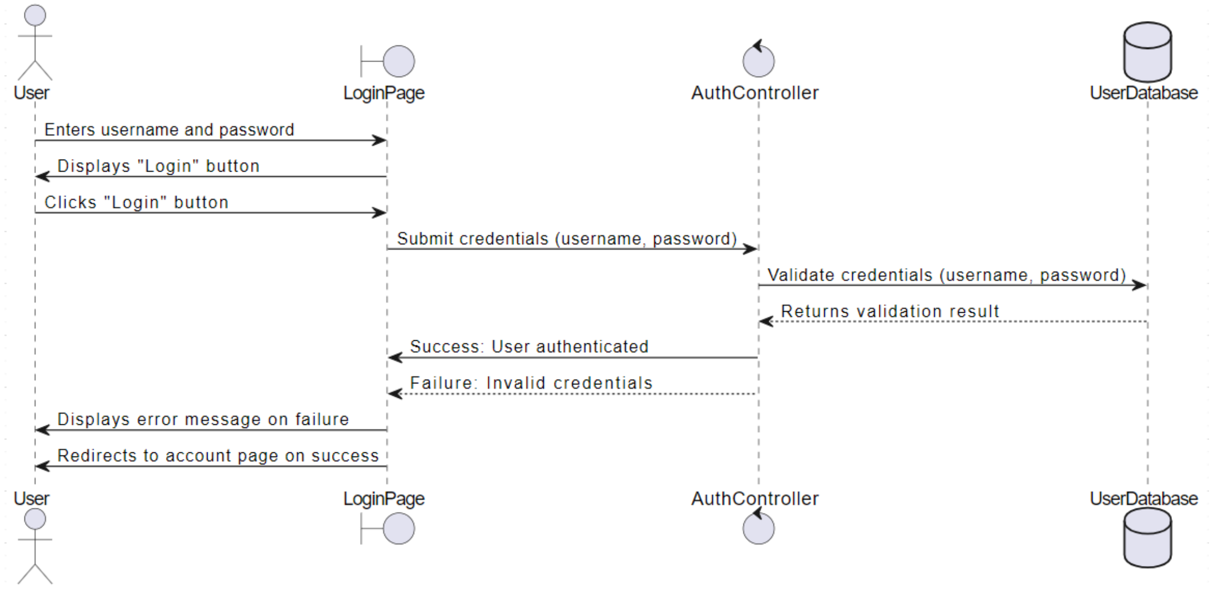


SEQUENCE DIAGRAMS FOR THE RESPECTIVE USER STORIES:

1. LOGIN PAGE :

As a user, I want to log in with my username and password so that I can access my account.

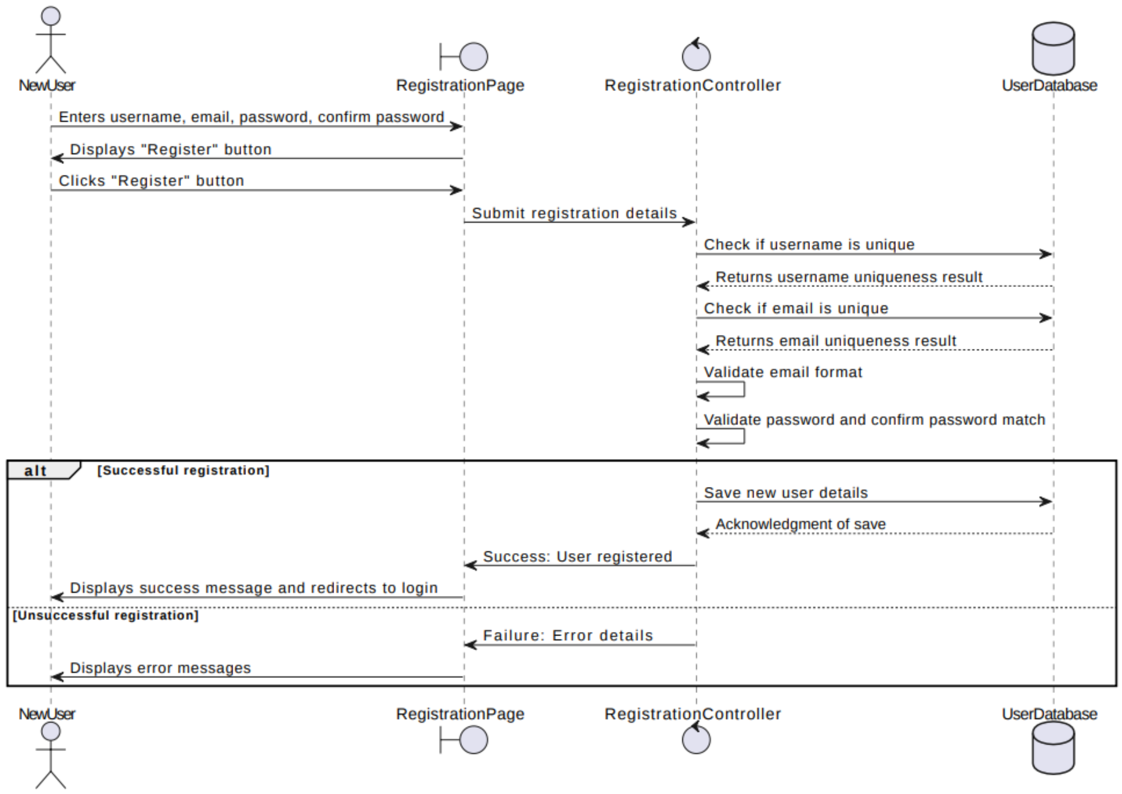
Sequence Diagram:


User
LoginPage
AuthController
UserDatabase
Enters username and password
Displays "Login" button
Clicks "Login" button
Submit credentials (username, password)
Validate credentials (username, password)
Returns validation result
Success: User authenticated
Failure: Invalid credentials
Displays error message on failure
Redirects to account page on success
User
LoginPage
AuthController
UserDatabase

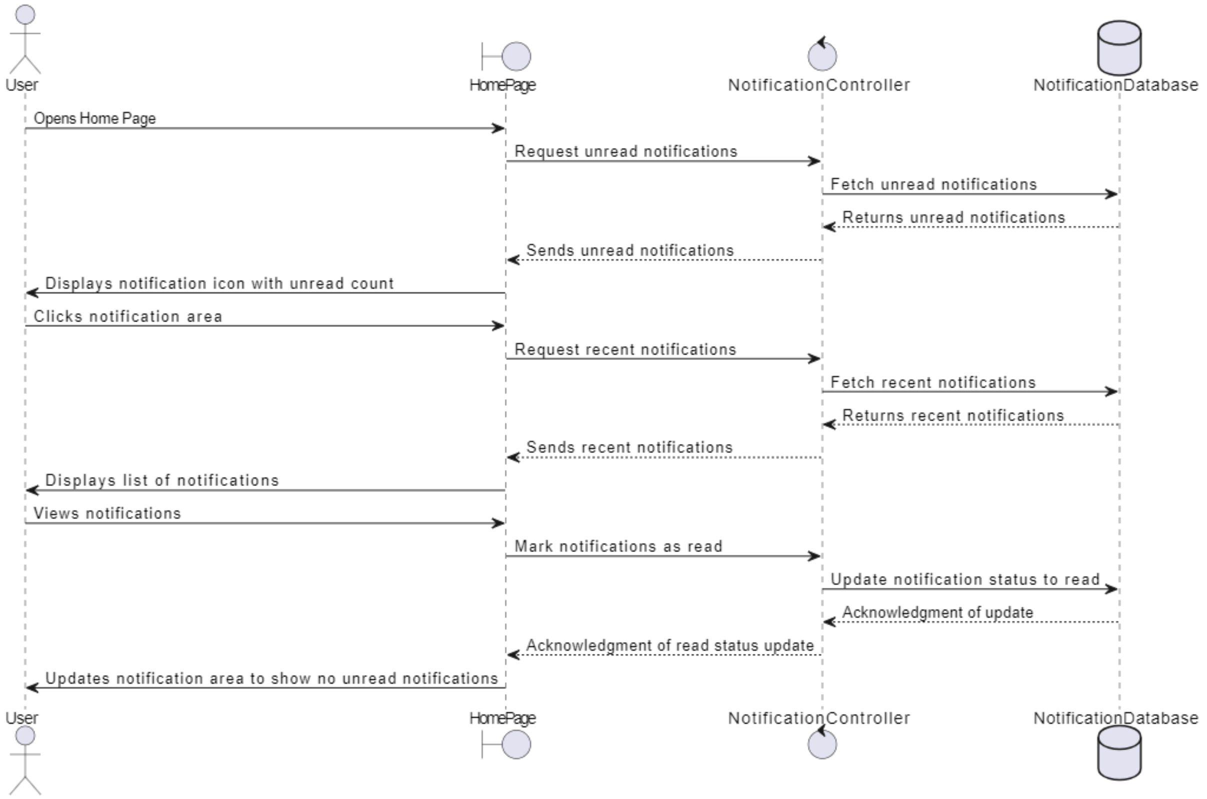
2. REGISTER PAGE :

User story 1: As a new user, I want to create an account so that I can access the app's features

Sequence Diagram:


NewUser
RegistrationPage
RegistrationController
UserDatabase
Enters username, email, password, confirm password
Displays "Register" button
Clicks "Register" button
Submit registration details
Check if username is unique
Returns username uniqueness result
Check if email is unique
Returns email uniqueness result
Validate email format
Validate password and confirm password match
alt
[Successful registration]
Save new user details
Acknowledgment of save
Success: User registered
Displays success message and redirects to login
[Unsuccessful registration]
Failure: Error details
Displays error messages
NewUser
RegistrationPage
RegistrationController
UserDatabase

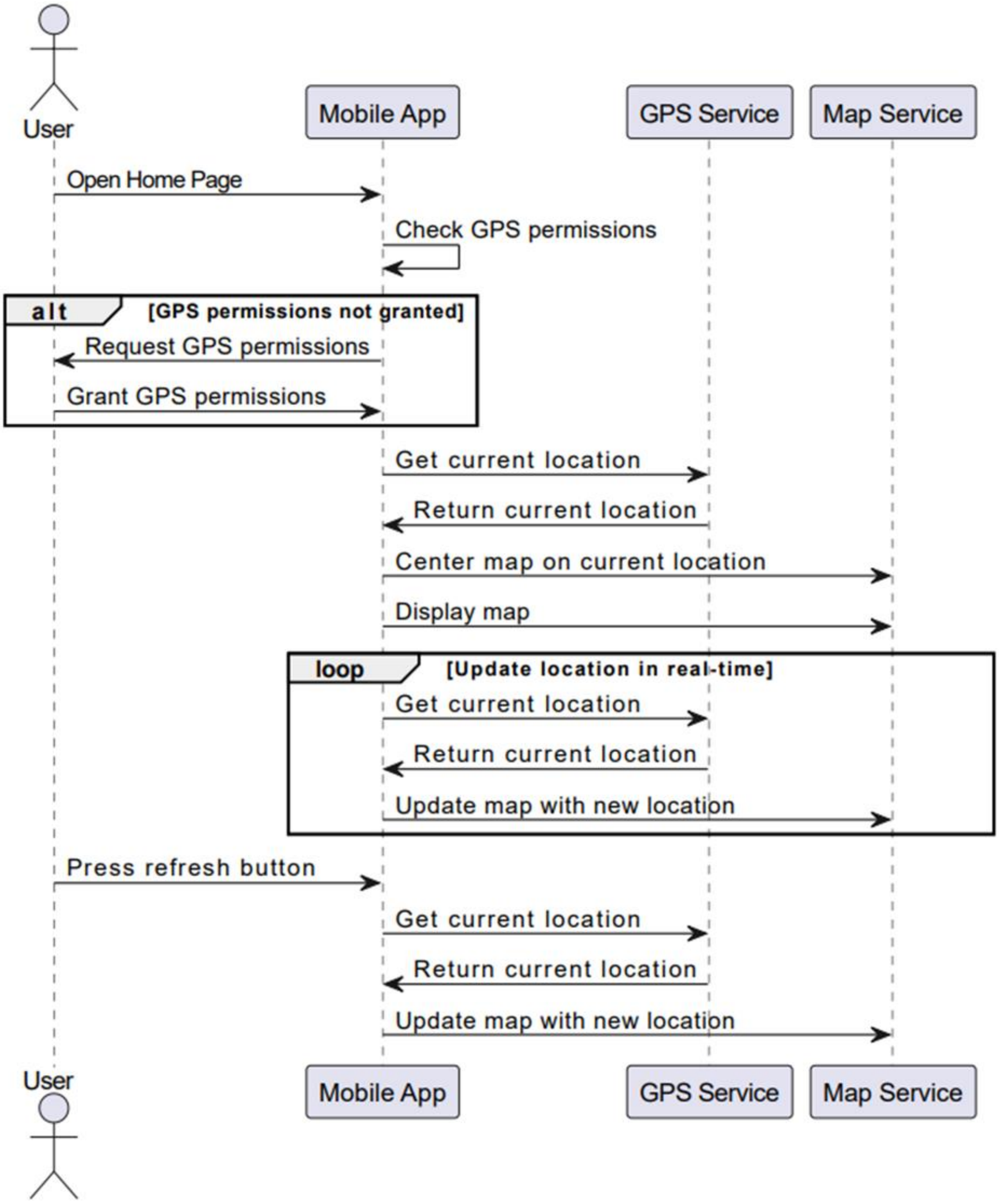
ii. USER STORY 2 : As a new user, I want to receive a confirmation email after registering so that I can verify my account.


User
HomePage
NotificationController
NotificationDatabase
Opens Home Page
Request unread notifications
Fetch unread notifications
Returns unread notifications
Sends unread notifications
Displays notification icon with unread count
Clicks notification area
Request recent notifications
Fetch recent notifications
Returns recent notifications
Sends recent notifications
Displays list of notifications
Views notifications
Mark notifications as read
Update notification status to read
Acknowledgment of update
Acknowledgment of read status update
Updates notification area to show no unread notifications
User
HomePage
NotificationController
NotificationDatabase

3. HOME PAGE :

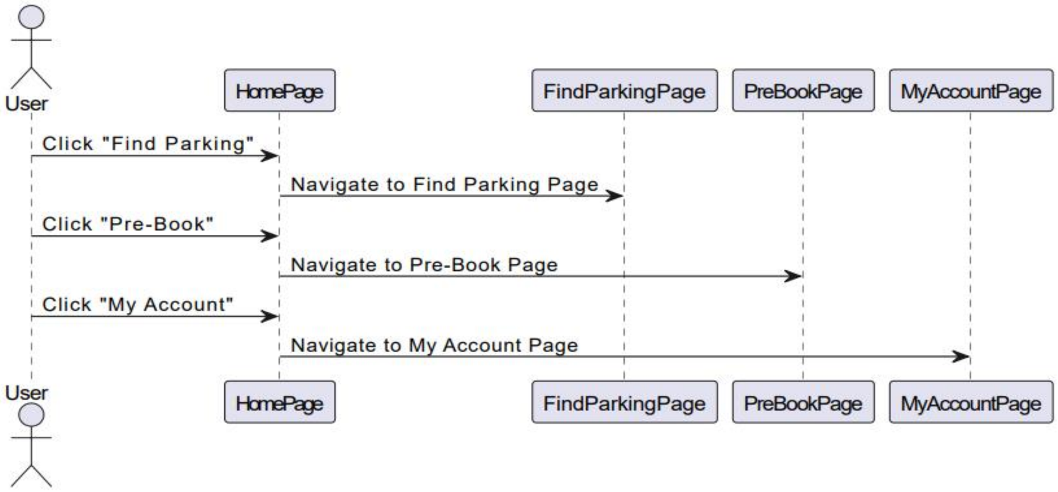
User Story 1 : As a user, I want to see my current location on the home page map so that I can find nearby parking spots.

Sequence Diagram:



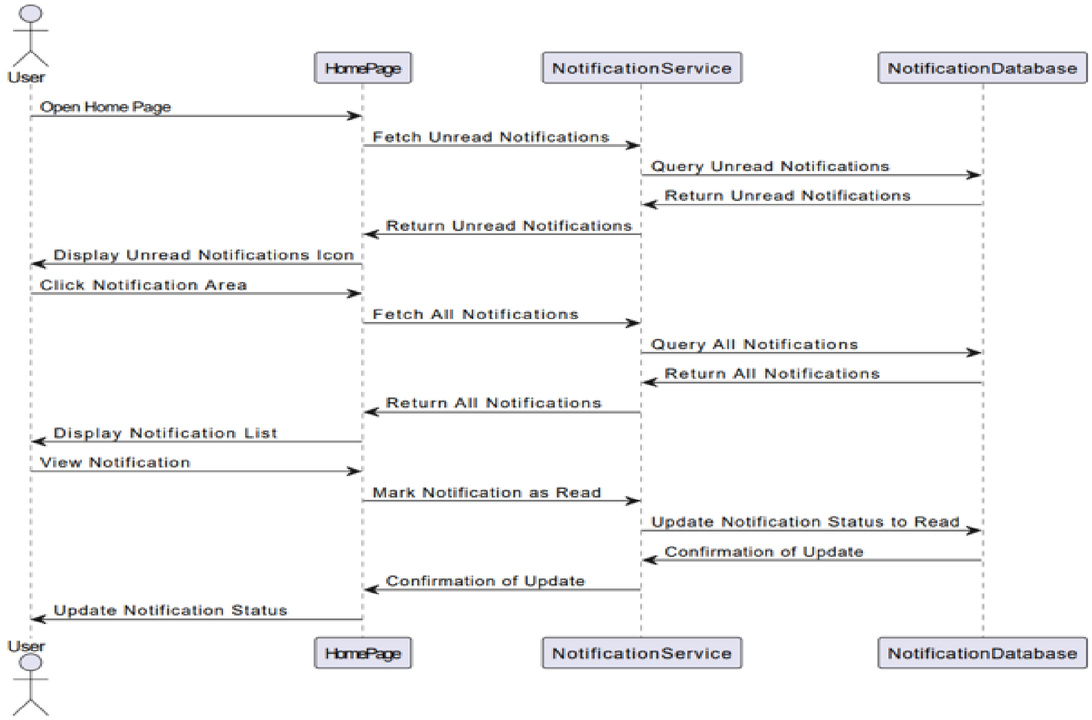
User Story 2 : As a user, I want quick access buttons to find parking, pre-book, and view my account so that I can navigate the app easily.

Sequence Diagram :



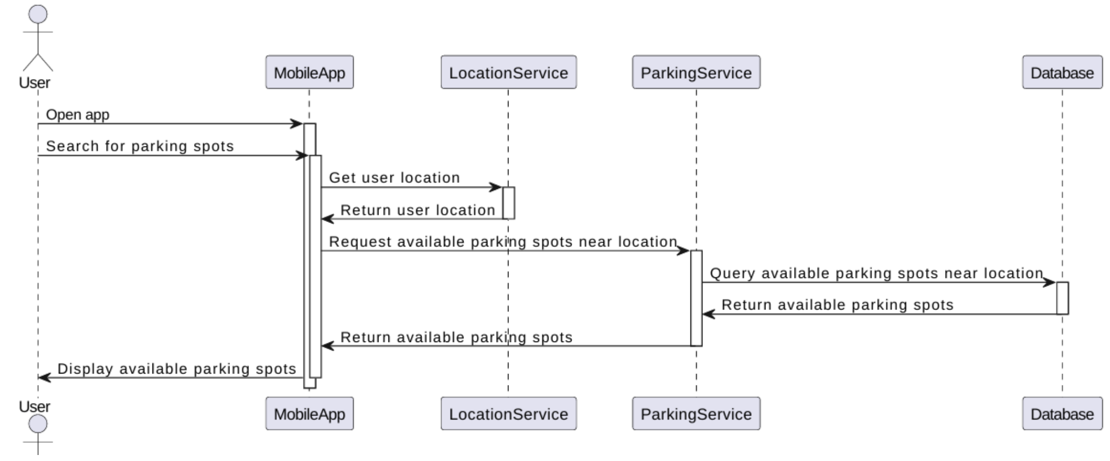
User Story 3 : As a user, I want to see notifications related to my bookings and payments on the home page so that I stay informed.

Sequence Diagram:

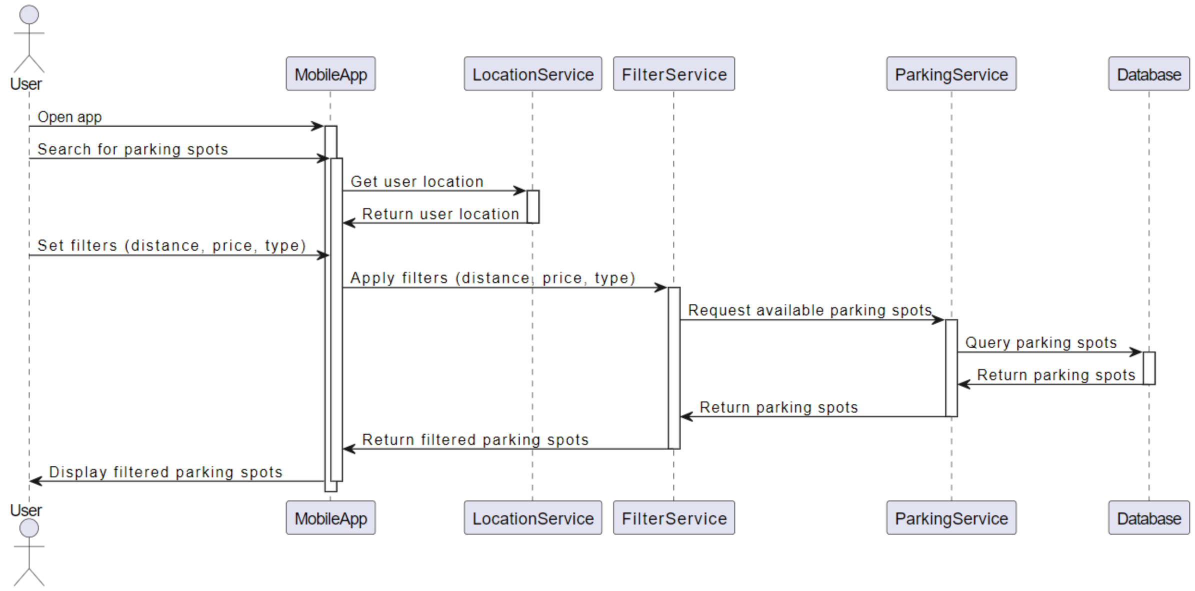


4. FIND PARKING PAGE :

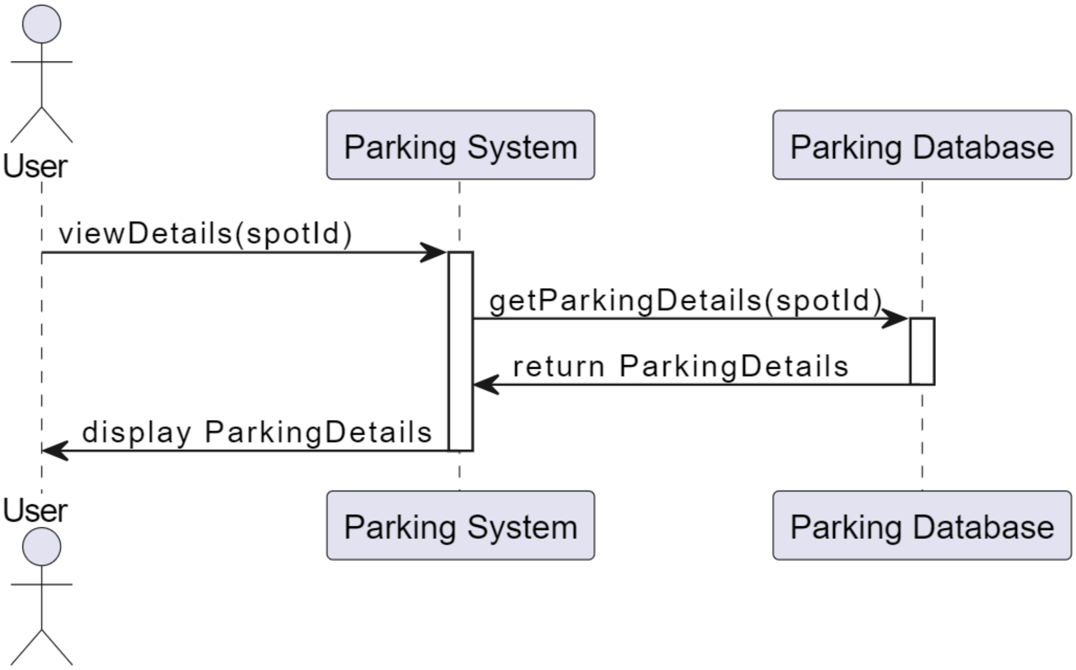
User Story 1: As a user, I want to search for available parking spots near my location so that I can find a place to park easily


User
MobileApp
LocationService
ParkingService
Database
Open app
Search for parking spots
Get user location
Return user location
Request available parking spots near location
Query available parking spots near location
Return available parking spots
Return available parking spots
Display available parking spots
User
MobileApp
LocationService
ParkingService
Database

User Story 2:As a user, I want to filter parking spots by distance, price, and type so that I can find the most suitable spot.



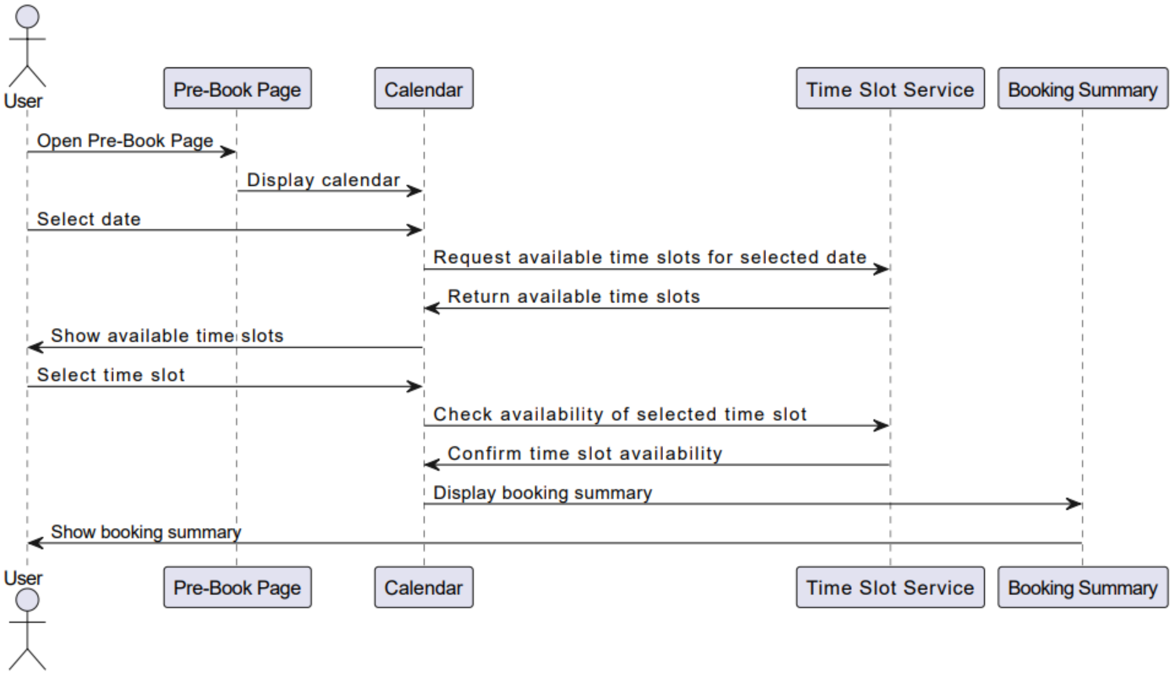
User Story 3 :As a user, I want to view detailed information about a parking spot so that I can make an informed decision.



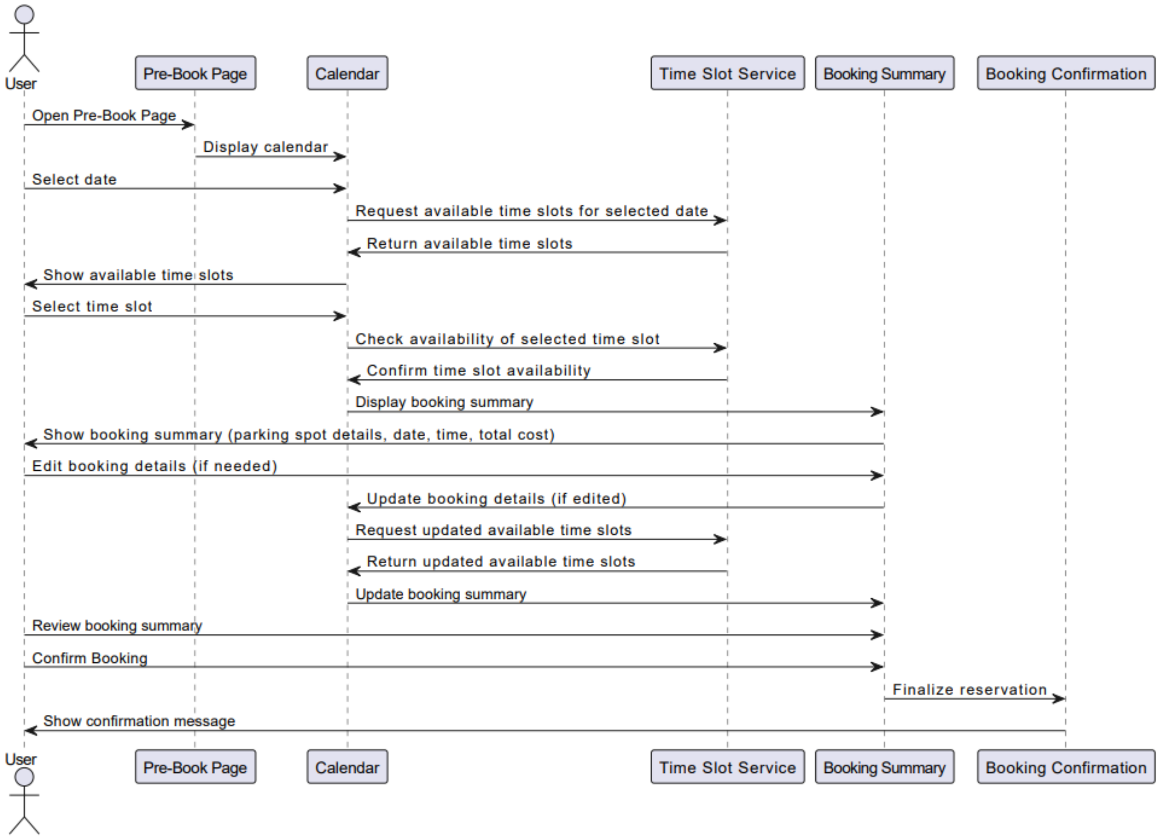
5. PRE-BOOK PAGE:

User Story 1: As a user, I want to select a date and time to pre-book a parking spot so that I can ensure availability.

Sequence Diagram :

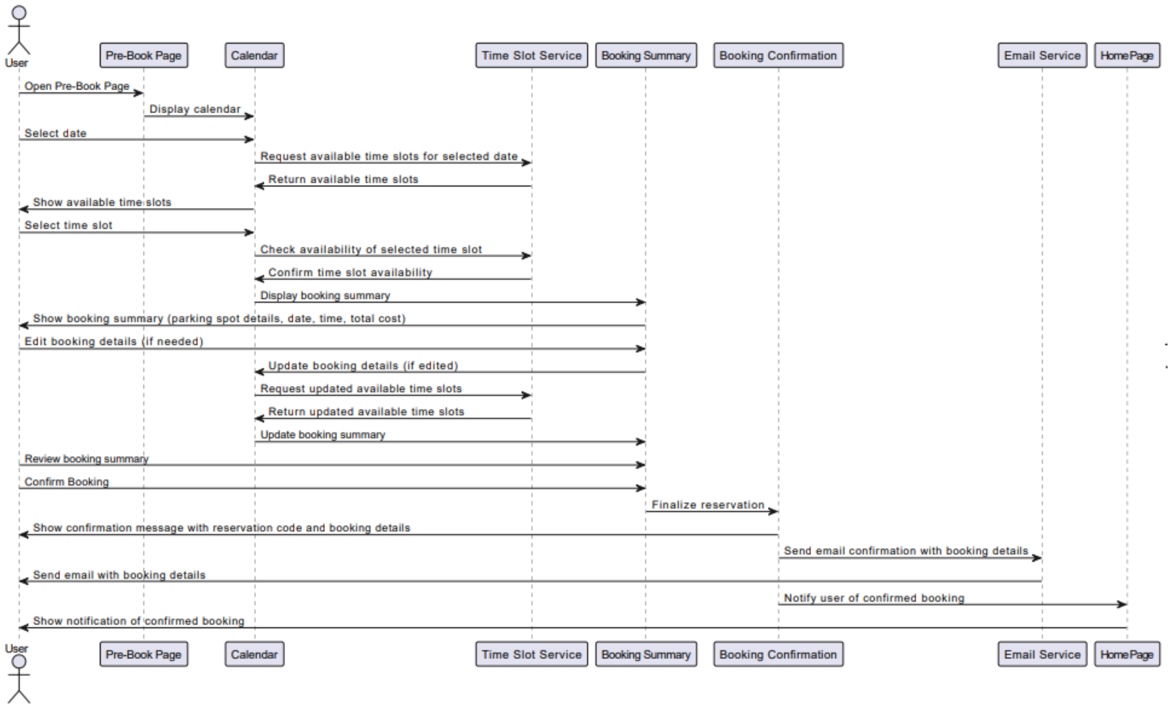


User Story 2: As a user, I want to review the details of my booking before confirming it so that I can ensure everything is correct.


User
Pre-Book Page
Calendar
Time Slot Service
Booking Summary
Booking Confirmation
Open Pre-Book Page
Display calendar
Select date
Request available time slots for selected date
Return available time slots
Show available time slots
Select time slot
Check availability of selected time slot
Confirm time slot availability
Display booking summary
Show booking summary (parking spot details, date, time, total cost)
Edit booking details (if needed)
Update booking details (if edited)
Request updated available time slots
Return updated available time slots
Update booking summary
Review booking summary
Confirm Booking
Finalize reservation
Show confirmation message
User
Pre-Book Page
Calendar
Time Slot Service
Booking Summary
Booking Confirmation

User Story 3: As a user, I want to receive a confirmation of my pre-booking so that I have proof of my reservation.

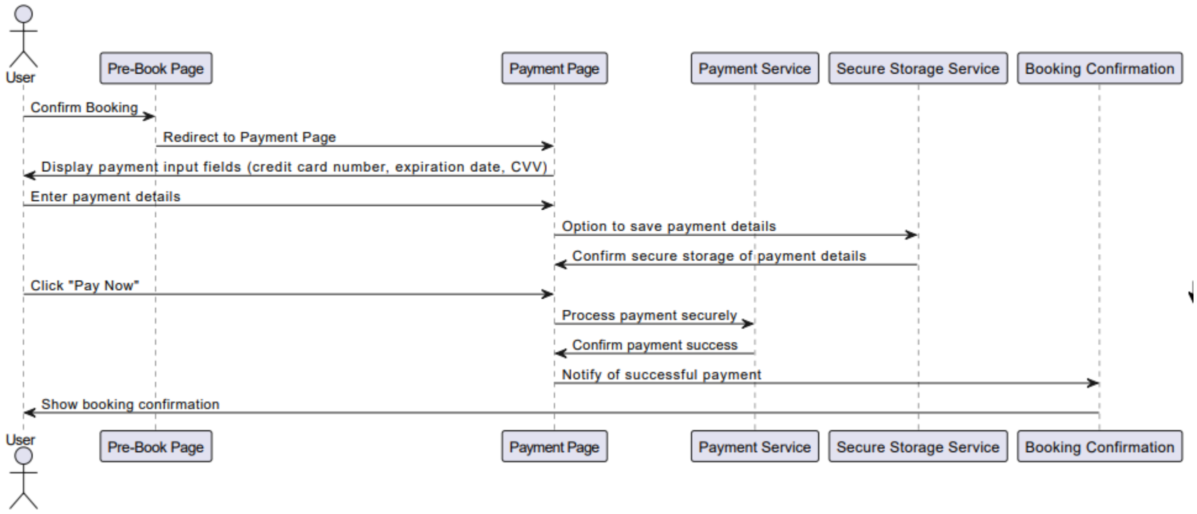
Sequence Diagram:


User
Pre-Book Page
Calendar
Time Slot Service
Booking Summary
Booking Confirmation
Email Service
HomePage
Open Pre-Book Page
Display calendar
Select date
Request available time slots for selected date
Return available time slots
Show available time slots
Select time slot
Check availability of selected time slot
Confirm time slot availability
Display booking summary
Show booking summary (parking spot details, date, time, total cost)
Edit booking details (if needed)
Update booking details (if edited)
Request updated available time slots
Return updated available time slots
Update booking summary
Review booking summary
Confirm Booking
Finalize reservation
Show confirmation message with reservation code and booking details
Send email confirmation with booking details
Send email with booking details
Notify user of confirmed booking
Show notification of confirmed booking
User
Pre-Book Page
Calendar
Time Slot Service
Booking Summary
Booking Confirmation
Email Service
HomePage

PAYMENT PAGE:

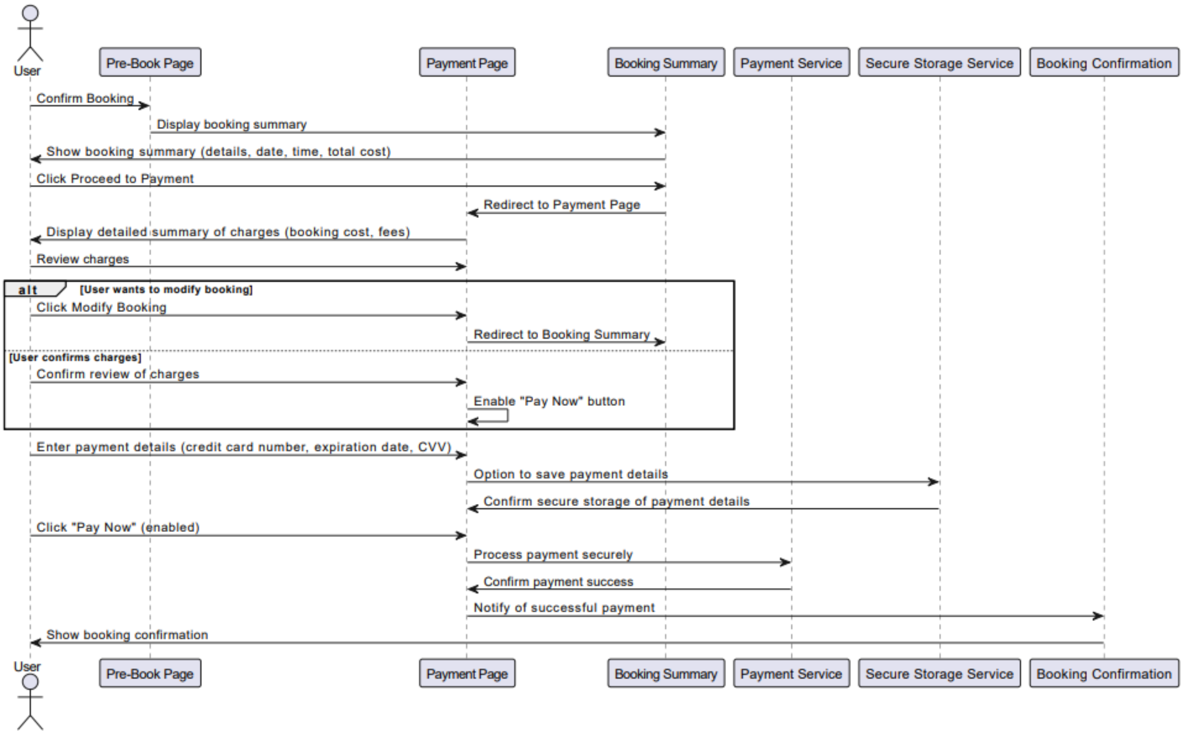
User Story 1: As a user, I want to securely enter my payment details so that I can pay for my pre-booking.

Sequence Diagram:

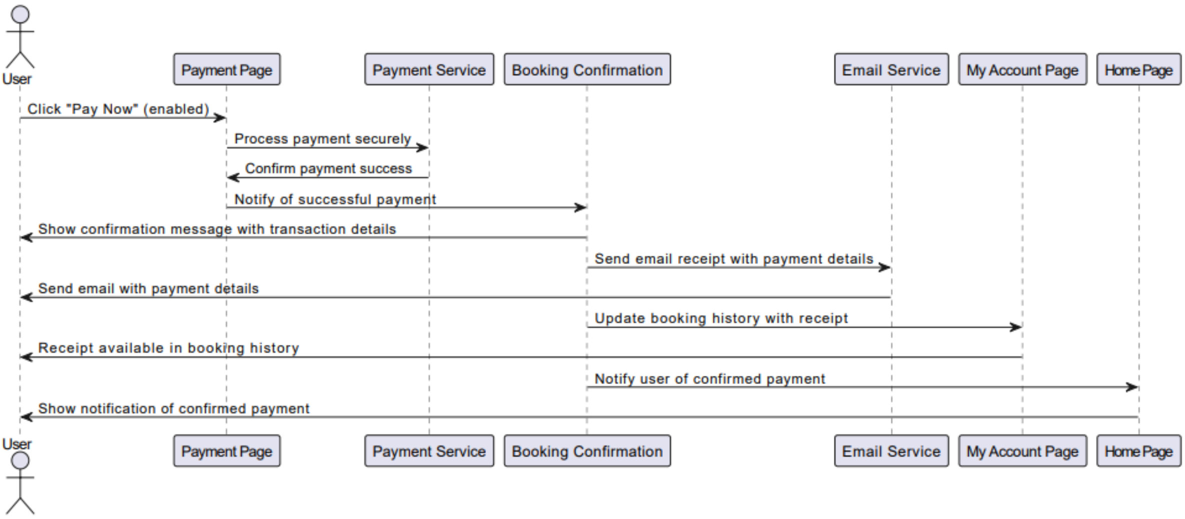


User Story 2: As a user, I want to see a summary of my charges before completing the payment so that I know the total amount.

Sequence Diagram :


User
Pre-Book Page
Payment Page
Booking Summary
Payment Service
Secure Storage Service
Booking Confirmation
Confirm Booking
Display booking summary
Show booking summary (details, date, time, total cost)
Click Proceed to Payment
Redirect to Payment Page
Display detailed summary of charges (booking cost, fees)
Review charges
alt
[User wants to modify booking]
Click Modify Booking
Redirect to Booking Summary
[User confirms charges]
Confirm review of charges
Enable "Pay Now" button
Enter payment details (credit card number, expiration date, CVV)
Option to save payment details
Confirm secure storage of payment details
Click "Pay Now" (enabled)
Process payment securely
Confirm payment success
Notify of successful payment
Show booking confirmation

User Story 3: As a user, I want to receive a receipt after making a payment so that I have proof of the transaction.


User
Payment Page
Payment Service
Booking Confirmation
Email Service
My Account Page
Home Page
Click "Pay Now" (enabled)
Process payment securely
Confirm payment success
Notify of successful payment
Show confirmation message with transaction details
Send email receipt with payment details
Send email with payment details
Update booking history with receipt
Receipt available in booking history
Notify user of confirmed payment
Show notification of confirmed payment

Sequence Diagram:

USER STORIES FOR CAR PARKING SYSTEM

1. Login Page

User Story:As a user, I want to log in with my username and password so that I can access my account.

Acceptance Criteria:

- The login page must have input fields for the username and password.
- There must be a "Login" button that submits the entered credentials.
- The app must validate the credentials and log the user in if they are correct.
- The app must display an error message if the credentials are incorrect.

2. Register Page

User Story 1:As a new user, I want to create an account so that I can access the app's features.

Acceptance Criteria:

- Input fields for username, email, password, and confirm password.
- Unique username validation.
- Correct format and uniqueness validation for email.
- Password and confirm password matching validation.
- Appropriate error messages for invalid inputs.

User Story 2: As a new user, I want to receive a confirmation email after registering so that I can verify my account.

Acceptance Criteria:

- Display confirmation message after successful registration.
- Send confirmation email with a verification link.

- Inactive account until verification link is clicked.
- Notification if email is already registered, suggesting login instead.

3. Home Page

User Story 1: As a user, I want to see my current location on the home page map so that I can find nearby parking spots.

Acceptance Criteria

- The home page must display a map centered on the user's current location.
- The app must request and obtain GPS permissions if not already granted.
- The user's current location must be updated in real-time on the map.
- The map must have a refresh button to manually update the location.

User Story 2:As a user, I want quick access buttons to find parking, pre-book, and view my account so that I can navigate the app easily.

Acceptance Criteria

- The home page must have clearly labeled buttons for "Find Parking," "Pre-Book," and "My Account."
- Clicking "Find Parking" must navigate to the Find Parking Page.
- Clicking "Pre-Book" must navigate to the Pre-Book Page.
- Clicking "My Account" must navigate to the My Account Page.

User Story 3: As a user, I want to see notifications related to my bookings and payments on the home page so that I stay informed.

Acceptance Criteria

- The home page must display a notification area or icon indicating unread notifications.
- Clicking the notification area must show a list of recent notifications.

- Each notification must include relevant details (e.g., booking confirmation, payment receipt).
- The app must mark notifications as read when viewed.

4. Find Parking Page

User Story 1:As a user, I want to search for available parking spots near my location so that I can find a place to park easily.

Acceptance Criteria

- The Find Parking Page must display a map with markers indicating available parking spots.
- The app must show a list of available parking spots with details like distance, price, and type.
- The user must be able to input search criteria (e.g., location, distance) to filter results.
- The search results must update in real-time based on the search criteria.

User Story 2: As a user, I want to filter parking spots by distance, price, and type so that I can find the most suitable spot.

Acceptance Criteria

- The Find Parking Page must have filter options for distance, price, and type (street, garage, lot).
- The user must be able to select and apply multiple filters.
- The app must update the displayed parking spots based on the selected filters.
- The app must display a clear message if no spots match the filter criteria.

User Story 3: As a user, I want to view detailed information about a parking spot so that I can make an informed decision.

Acceptance Criteria

- Clicking on a parking spot marker or list item must navigate to the Parking Spot Details Page.
- The Parking Spot Details Page must display comprehensive information about the spot (availability, price, restrictions, etc.).
- There must be options to view the spot on the map or get navigation directions.

- The page must have a button to proceed with booking the spot.

6. Pre-Book Page

User Story 1**:** As a user, I want to select a date and time to pre-book a parking spot so that I can ensure availability.

Acceptance Criteria

- The Pre-Book Page must have a calendar to select the desired date.
- The page must show available time slots for the selected date.
- The user must be able to select a time slot and see a summary of the booking.
- The app must prevent selection of already booked time slots.

User Story 2: As a user, I want to review the details of my booking before confirming it so that I can ensure everything is correct.

Acceptance Criteria

- After selecting a date and time, the Pre-Book Page must display a booking summary.
- The summary must include the parking spot details, date, time, and total cost.
- There must be an option to edit the booking details before confirming.
- The app must have a "Confirm Booking" button to finalize the reservation.

User Story 3: As a user, I want to receive a confirmation of my pre-booking so that I have proof of my reservation.

Acceptance Criteria

- After confirming the booking, the app must display a booking confirmation page.
- The confirmation page must include a reservation code and booking details.
- The app must send an email confirmation to the user with the booking details.

- The user must see a notification on the Home Page confirming the booking.**6. Payment Page**

User Story 1: As a user, I want to securely enter my payment details so that I can pay for my pre-booking.

Acceptance Criteria

- The Payment Page must have input fields for payment details (e.g., credit card number, expiration date, CVV).
- The app must securely handle and store the payment information.
- There must be an option to save payment details for future use.
- The "Pay Now" button must process the payment securely.

User Story 2: As a user, I want to see a summary of my charges before completing the payment so that I know the total amount.

Acceptance Criteria

- The Payment Page must display a detailed summary of the charges, including the booking cost and any applicable fees.
- The summary must be clear and easy to understand.
- The user must have the option to go back and review or modify the booking details.
- The "Pay Now" button must be disabled until the user reviews the charges.

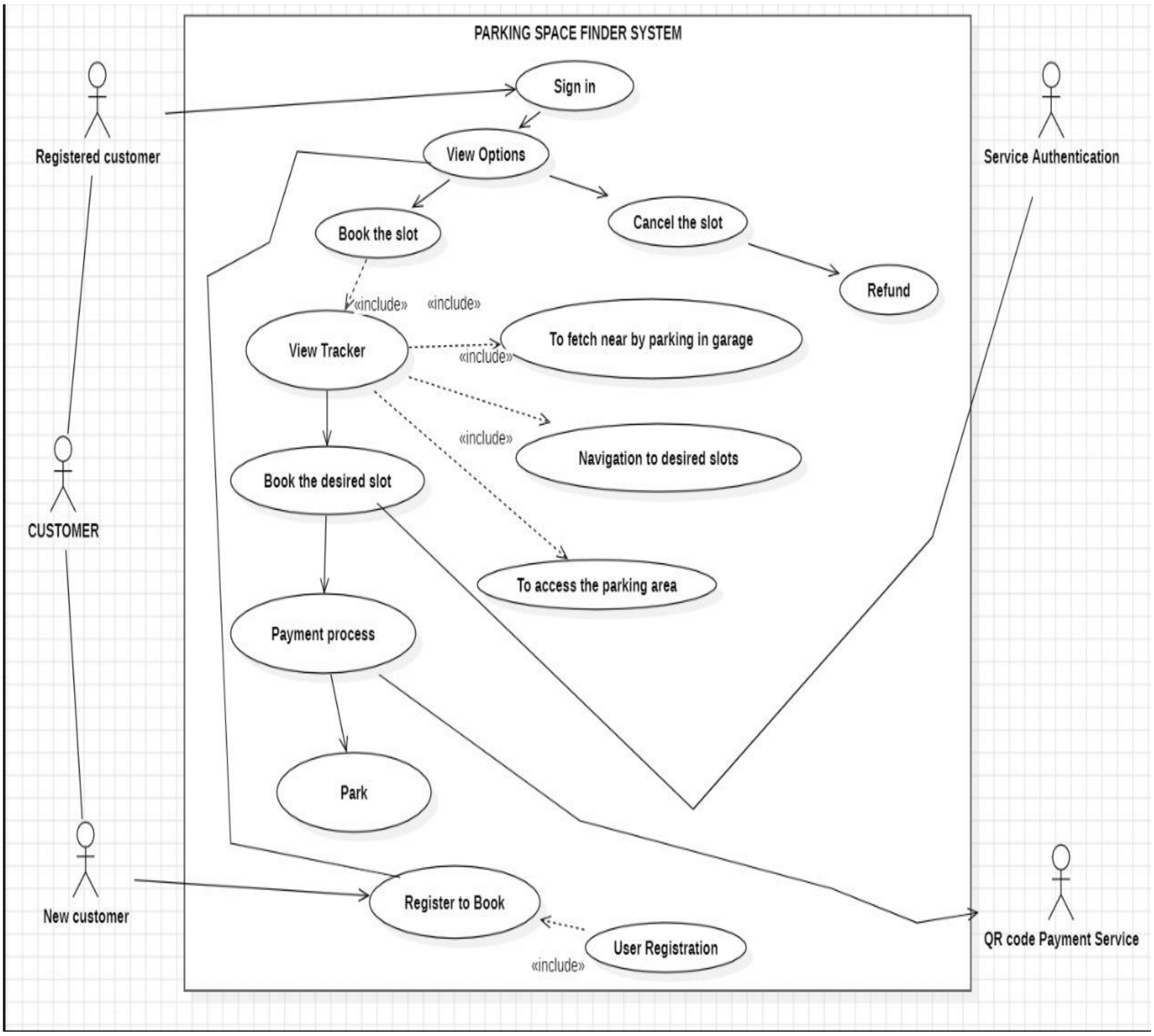
User Story 3: As a user, I want to receive a receipt after making a payment so that I have proof of the transaction.

Acceptance Criteria

- Upon successful payment, the app must display a confirmation message with transaction details.
- The app must send an email receipt to the user with the payment details.
- The receipt must be available in the user's booking history on the My Account Page.

- The user must see a notification on the Home Page confirming the payment

USE CASE DIAGRAM



TEST STRATEGY :

Document the Test Plans

Test Plan Overview:

Project Name: Parking Space Finder App

Objective: To ensure that the app meets all functional and non-functional requirements, providing a seamless and efficient user experience.

Scope: This test plan covers user registration, viewing available parking spaces, filtering parking spaces, real-time availability updates, and reserving a parking space.

Testing Types: Functional Testing, Usability Testing, Performance Testing, Security Testing.

Tools: Selenium, JUnit, Postman, JMeter, OWASP ZAP.

Environment: Test environment replicates the production environment including databases, servers, and network configurations.

Test Phases:

1. Unit Testing
2. Integration Testing
3. System Testing
4. User Acceptance Testing (UAT)

Schedule: Detailed timeline with milestones for each test phase.

Test Cases for User Stories

User Registration

Happy Path:

Preconditions: User is on the registration page.

Test Steps:

Enter valid registration details (username, password, email).

Click on "Register" button.

Verify email received and click verification link.

Expected Result: User account is created and verified successfully.

Error Scenarios:

Invalid Email:

Test Steps:

Enter invalid email format.

Click on "Register" button.

Expected Result: Display error message "Invalid email format."

Duplicate Email:

Test Steps:

Enter an email that is already registered.

Click on "Register" button.

Expected Result: Display error message "Email already registered."

View Available Parking Spaces

Happy Path:

Preconditions: User is logged in and on the main page.

Test Steps:

Enter location in the search bar.

Click on "Search" button.

Expected Result: Display a list of available parking spaces on the map.

Error Scenarios:

No Results Found:

Test Steps:

Enter a location with no available parking spaces.

Click on "Search" button.

Expected Result: Display message "No parking spaces available."

Filter Parking Spaces

Happy Path:

Preconditions: User is on the available parking spaces page.

Test Steps:

Select filter criteria (price, type).

Click on "Apply Filters" button.

Expected Result: Display filtered list of parking spaces.

Error Scenarios:

Invalid Filter Combination:

Test Steps:

Select incompatible filter criteria.

Click on "Apply Filters" button.

Expected Result: Display message "No parking spaces match your criteria."

Real-Time Availability Updates

Happy Path:

Preconditions: User is viewing a list of available parking spaces.

Test Steps:

Wait for real-time updates.

Expected Result: Availability status updates in real-time.

Error Scenarios:

Network Error:

Test Steps:

Simulate network disconnection.

Expected Result: Display message "Unable to update availability. Please check your connection."

Reserve a Parking Space

Happy Path:

Preconditions: User is logged in and has selected a parking space.

Test Steps:

Click on "Reserve" button.

Confirm reservation details.

Complete payment process.

Expected Result: Display reservation confirmation.

Error Scenarios:

Space Unavailable:

Test Steps:

Select a parking space that becomes unavailable during reservation.

Click on "Reserve" button.

Expected Result: Display message "Parking space no longer available."

GitHub Repository View

Repository Name: parking-space-finder-app

Project Structure:

src/: Contains source code

main/: Main application code

java/: Java source files

resources/: Application resources

test/: Test cases

java/: Test source files

resources/: Test resources

docs/: Documentation

config/: Configuration files

scripts/: Scripts for build, deployment, etc.

README.md: Project overview and setup instructions

pom.xml or build.gradle: Build configuration file

DevOps Architecture and Tools Used in Azure

Azure DevOps Pipeline:

Build Pipeline: Automated build process using Azure Pipelines.

Tools: Maven/Gradle, Azure Pipelines

Steps:

Checkout code

Compile code

Run unit tests

Package application

Release Pipeline: Automated release process using Azure Pipelines.

Tools: Azure Pipelines, Azure App Service, Azure SQL Database

Steps:

Deploy to staging environment

Run integration tests

Manual approval

Deploy to production environment

CI/CD Tools:

Continuous Integration: GitHub Actions or Azure Pipelines for build and test automation.

Continuous Deployment: Azure Pipelines for automated deployments to Azure services.

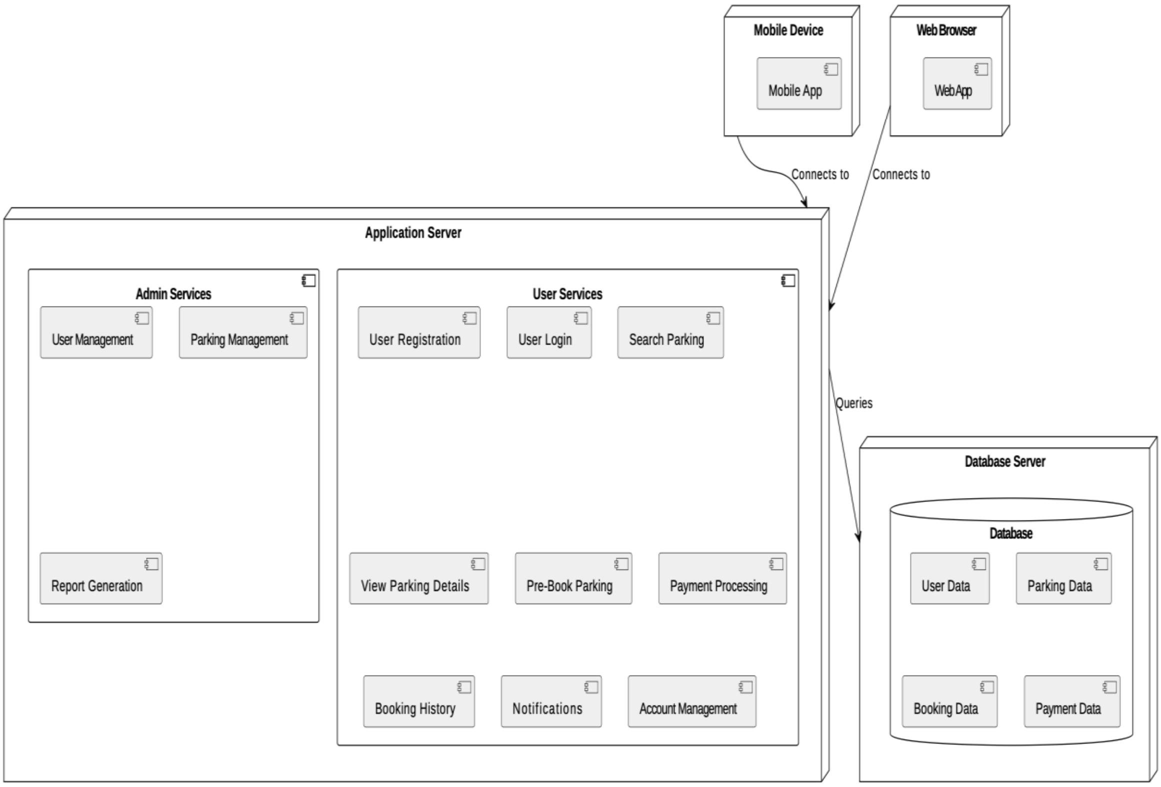
Monitoring and Logging:

Tools: Azure Monitor, Application Insights

Features: Real-time monitoring, logging, and alerting for application performance and health.

By implementing these strategies, the parking space finder app aims to ensure high-quality, reliable, and efficient service delivery, meeting user expectations and business objectives.

DEPLOYMENT ARCHITECTURE OF THE PROJECT



CONCLUSION

The car parking system project represents a comprehensive solution designed to address the challenges faced by drivers in urban environments, where finding and reserving parking spaces can be a significant source of stress and inefficiency. Through the integration of advanced technologies such as GPS, mobile applications, and automated payment systems, this project aims to streamline the entire parking experience, making it more convenient, efficient, and user-friendly.