B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

OOMD Mini Project Report

SMART PARKING SYSTEM

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering in
Computer Science and Engineering

Submitted by:

Mithun.M(1BM23CS192)
Mahantesh(1BM23CS175)
Mahesh(1BM23CS176)
Mallikarjun(1BM23CS179)

Department of Computer Science and Engineering B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019

2025-26

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We,**Mithun.M(1BM23CS192),Mahantesh(1BM23CS175),Mahesh Lamani (1BM23CS176),Mallikarjun(1BM23CS179),** students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD Mini Project entitled "**SMART PARKING SYSTEM**" has been carried out in Department of CSE, B.M.S. College of Engineering, Bangalore during the academic semester August 2025- December 2025. I also declare that to the best of our knowledge and belief, the OOMD mini Project report is not from part of any other report by any other students.

Signature of the Candidate

Mithun.M(1BM23CS192)
Mahantesh(1BM23CS175)
Mahesh(1BM23CS176)
Mallikarjun(1BM23CS179)

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the OOMD Mini Project titled "**SMART PARKING SYSTEM"** has been carried out by Mithun.M(1BM23CS192),Mahantesh(1BM23CS175),Mahesh Lamani (1BM23CS176),Mallikarjun(1BM23CS179) during the academic year 2025-2026.

Signature of the Faculty in Charge (Your Guide Name)

Table of Contents

Chapter 1: Problem Statement

Urban areas across India are experiencing rapid growth in vehicle ownership, resulting in severe parking challenges across cities, campuses, commercial zones, and residential complexes. Drivers often waste significant time searching for parking spaces, which leads to traffic congestion, increased fuel consumption, and rising frustration. Traditional parking management in many places remains manual, poorly organized, and lacks transparency, causing inefficiency for both users and administrators.

A major gap in the current system is the absence of a unified, technology-driven platform for intelligent parking management. Users rarely have access to real-time information about available parking slots, pricing, or navigation assistance. Parking operators struggle with issues such as improper space utilization, revenue leakage, lack of monitoring, and inefficient enforcement. Without digital tracking or automated systems, parking zones often become overcrowded and mismanaged, worsening traffic conditions and user dissatisfaction.

To address these issues, there is a need for a comprehensive smart parking system that can monitor, optimize, and guide users to available parking spaces. By incorporating technologies such as IoT-based sensors, mobile applications, RFID/ANPR-based access control, digital payment systems, and centralized analytics dashboards, the platform can significantly reduce search time and improve operational efficiency. Such a system would enhance user convenience, reduce congestion, promote sustainable traffic flow, and ultimately create a smarter, more efficient parking ecosystem across urban environments.

Chapter 2: Software Requirement Specification

Disaster Preparedness and Response Education System for Schools and Colleges

1. Introduction

1.1 Purpose of this Document:

The purpose of this document is to define the requirements and specifications for the development of the Smart Parking Management System (SPMS). The document outlines the project's objectives, scope, functionalities, constraints, and deliverables. It aims to provide a unified understanding for developers, administrators, parking authorities, and other stakeholders involved in the system.

1.2 Scope of this Document:

This SRS describes the overall workflow and objectives of SPMS. It covers system modules such as real-time parking space detection, user mobile app interface, centralized database logging, online booking and payment, slot allocation algorithms, monitoring dashboards, and communication with parking authorities. It also includes estimates for project duration, development cost, and system constraints.

1.3 Overview:

The Smart Parking Management System is a digital platform designed for urban areas, commercial zones, malls, and educational campuses to improve parking efficiency, reduce congestion, and enhance user convenience. The system enables users to locate available parking spaces, reserve slots, and make digital payments, while administrators can monitor occupancy, manage revenue, and enforce rules using automated tools. It also provides analytics dashboards, alerts, navigation assistance, and centralized reporting for better decision-making and space utilization.

2. General Description

The **Smart Parking Management System (SPMS)** will serve drivers, parking operators, campus or municipal administrators, and enforcement authorities. The system provides:

❖ Real-time parking slot detection with precise location and availability details
❖ Automated vehicle entry/exit logging through IoT sensors, RFID tags, or ANPR cameras
❖ A user-friendly mobile/web interface for slot booking, navigation, and payment
❖ Digital payment options for parking fees, including UPI, wallets, and card payments
❖ Periodic reporting with occupancy analytics, revenue insights, and performance tracking
❖ A secure centralized database for storing parking logs, reservation details, transactions, and user records
❖ Role-based access levels for drivers, operators, admins, and enforcement authorities

The system will be browser-based with optional mobile app compatibility to ensure accessibility for users with varying technical expertise.

3. Functional Requirements

3.1 Parking Slot Detection & Monitoring Module

❖ Detect available parking slots using IoT sensors, RFID scanners, or ANPR-based monitoring.
❖ Capture real-time slot status, vehicle presence, timestamp, and location.
❖ Send instant availability updates to the user interface.
❖ Generate a unique slot transaction ID for each parking session.

3.2 Slot Booking & Verification Module

❖ Enable users to view and reserve available parking slots in real time.
❖ Provide verification tools (QR code scan, RFID card validation, ANPR vehicle match).
❖ Mark reservations as **"Confirmed"**, **"Expired"**, or **"Already Occupied"**.
❖ Automatically release the slot if a user does not arrive within the allowed time.

3.3 Payment & Billing Management Module

❖ Deliver structured payment options such as UPI, digital wallets, and card payments.
❖ Provide invoices, transaction history, and automated billing after parking session completion.
❖ Track user payment activity and generate revenue and billing reports.

3.4 Parking Operator / Admin Interface

❖ Receive real-time reservation updates and user arrival notifications.
❖ Provide tools to classify slot issues (misuse, sensor error, unauthorized vehicle).
❖ Allow operators to override slot status, manually verify vehicles, and update occupancy.
❖ Enable two-way updates between operator dashboard and the user system.

3.5 Slot & Facility Management Module

❖ Allow scheduling of maintenance or temporary slot closure.
❖ Track slot performance, uptime, and sensor health.
❖ Generate reports on slot usage, occupancy trends, and peak hours.

3.6 Administrative Control & Reporting

❖ Manage users, roles, and access permissions.
❖ Maintain parking zones, pricing rules, and system configurations.
❖ Generate multi-level reports on reservations, payments, slot usage, and violations.

4. Interface Requirements

4.1 User Interface

❖ Intuitive and responsive UI designed for drivers, operators, and administrators.
❖ Accessible through web browsers, Android/iOS mobile apps, and desktop systems.
❖ Dashboard-based navigation with real-time slot availability, booking options, and role-specific panels.

4.2 Integration Interfaces

❖ Integration with IoT sensors, RFID systems, and ANPR cameras for automated slot detection.
❖ Integration with digital payment gateways (UPI, wallets, card processors) for seamless billing.
❖ Secure REST API for future integration with municipal parking networks or smart-city platforms.

5. Performance Requirements

5.1 Response Time

❖ The system should display real-time parking slot availability updates within **2 seconds** of sensor detection.

5.2 Scalability

❖ Capable of supporting **5000+ concurrent users** checking or booking slots across multiple parking zones.

5.3 Data Integrity

❖ Ensure accuracy and consistency of reservation records, payment logs, sensor data, and parking session histories across all modules

6. Design Constraints

6.1 Hardware Limitations

❖ The system should operate efficiently on standard devices such as parking kiosks, operator PCs, tablets, and basic Android/iOS smartphones.

6.2 Software Dependencies

❖ Use a relational database (MySQL / PostgreSQL) for structured parking data storage.
❖ Backend frameworks may include Java Spring Boot / Node.js / Django (per development requirement).
❖ Must support UML-based documentation and modular microservice-friendly architecture

7. Non-Functional Attributes

7.1 Security

❖ Implement authentication, authorization, and encryption to protect user and payment data.
❖ Role-based access control for Driver / Operator / Admin roles.

7.2 Reliability

❖ Ensure **99% uptime** for uninterrupted parking operations.
❖ Fault-tolerant design to prevent downtime due to device or sensor failure.

7.3 Scalability

❖ The system should accommodate additional parking zones, sensors, and user groups without major redesign.

7.4 Portability

❖ Support cross-platform accessibility (Windows, Linux, Android, iOS).

7.5 Usability

❖ Provide a simple, user-friendly UI with intuitive navigation for booking and slot viewing.
❖ Operator dashboards should clearly display slot status, alerts, and real-time occupancy.

7.6 Reusability

❖ Modules such as booking, payment, and slot management should be reusable for future parking lots or smart-city expansions.

7.7 Compatibility

❖ Compatible with major web browsers (Chrome, Firefox, Microsoft Edge, Safari).

7.8 Data Integrity

- ❖ Ensure accurate logging, retrieval, and synchronization of parking session data.
- ❖ Implement validation rules to prevent incorrect bookings, duplicate entries, or payment mismatch.

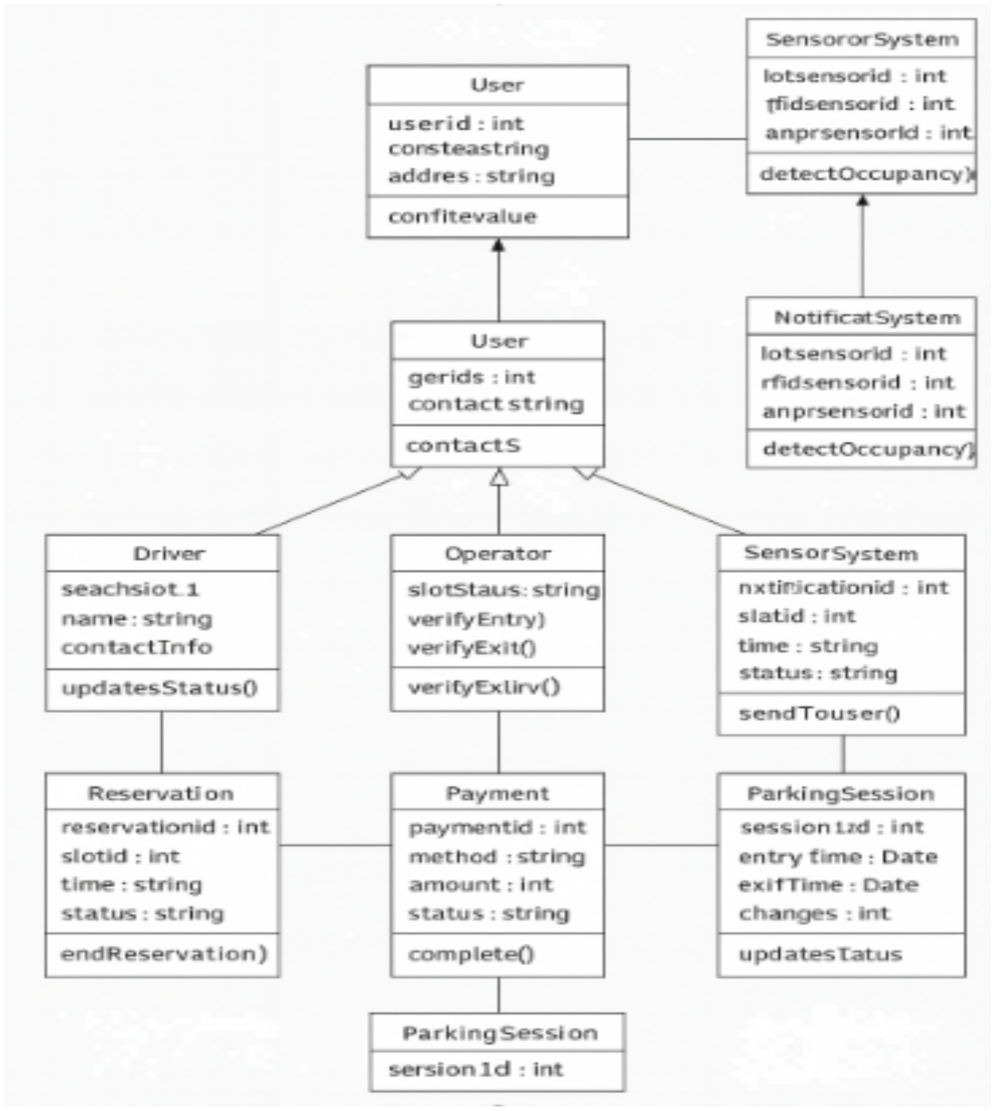
8. Preliminary Schedule and Budget

The development of the Smart Parking Management System is estimated to take 8 months, including:

- ❖ **Requirements analysis**
- ❖ **System design & architecture**
- ❖ **Backend, frontend, and IoT integration development**
- ❖ **Testing (functional, sensor accuracy, reliability, security, performance)**
- ❖ **Deployment across pilot parking zones**

Chapter 3: Class Modeling

1. Class Diagram



1.1. User (Base Class)

Relevance:
The User class represents the foundational identity for anyone interacting with the Smart Parking System. It contains shared attributes such as userID, name, phone number, and login credentials. It also includes common functions like login/logout.
This class allows the system to manage multiple user roles (driver, operator, admin) in a unified and scalable manner.

1.2. Driver (Derived from User)

Relevance:
Represents individuals who search for parking slots, reserve spaces, and make payments. Drivers interact with slot availability, booking, and history-related features.
This class ensures that parking services and payment operations are correctly mapped to the end user.

1.3. Operator (Derived from User)

Relevance:
Operators manage slot-related activities such as updating slot status, monitoring occupancy, verifying vehicles, and checking for misuse.
This class captures the supervisory or ground-level management role within the parking facility.

1.4. Admin (Derived from User)

Relevance:
Admins oversee high-level system functions such as managing users, configuring parking slots, setting pricing rules, and generating analytical reports.
This class ensures the system remains organized, secure, and fully operational.

1.5. ParkingSlot

Relevance:
Represents an individual parking slot in the system. It stores slot ID, location, type (car/bike), availability, and

sensor data.
This class is essential for tracking real-time slot status and enabling accurate reservations.

1.6. Reservation

Relevance:
Represents a booking made by a driver for a specific slot at a specific time. It stores booking ID, driver ID, slot ID, reservation time, and status.
This class allows the system to manage reservation workflows—from slot selection to payment confirmation.

1.7. Payment

Relevance:
Handles the processing and verification of payments made for parking reservations. It stores transaction details, payment methods, and statuses.
This class supports secure financial operations and ensures traceability and billing accuracy.

1.8. ParkingSession

Relevance:
Tracks the user's active parking duration after the vehicle enters the slot. It stores entry time, exit time, duration, and calculated charges.
This class links real-time parking usage with billing and reporting functions.

1.9. SensorSystem

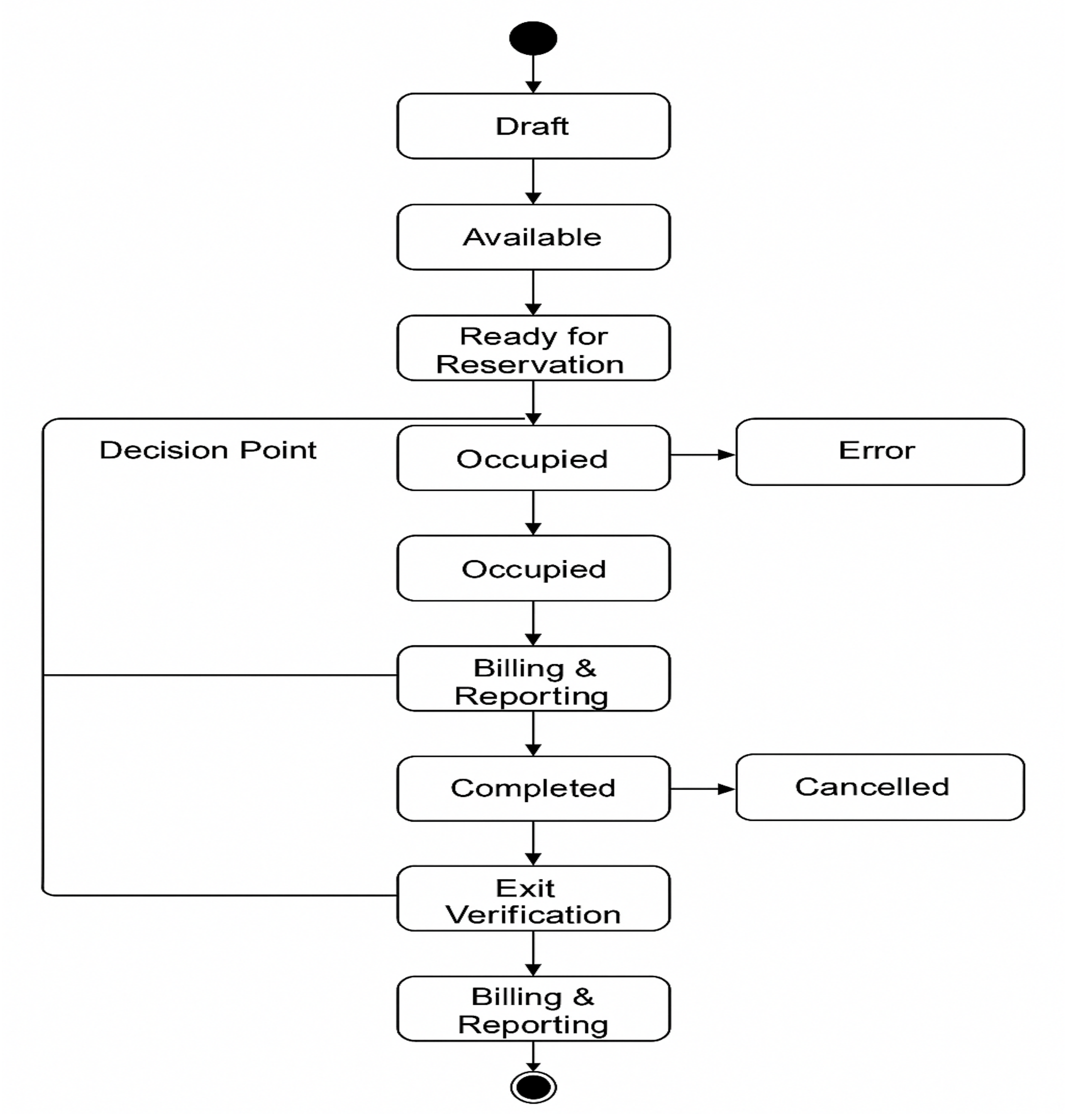
Relevance:
Represents the IoT-based sensors or ANPR/RFID systems that detect slot occupancy, vehicle entry, and exit.
This class ensures that the Smart Parking System maintains accurate, automated status updates.

1.10. NotificationSystem

Relevance:
Handles messages, alerts, and communication between the system and users. This includes booking confirmations, payment updates, expiry alerts, and misuse warnings.
This class improves user awareness and enhances operational efficiency across the system.

Chapter 4: State Modeling

2. State Diagram



2.1 Explanation of Each State

2.1.1 Draft

❖ This is the initial state of a newly added parking slot or newly created reservation setup.
❖ Basic slot details (slot ID, type, location, pricing) are entered.
❖ The slot or reservation is not yet active and can be modified or deleted.

2.1.2 Available

❖ The slot has been validated and approved.
❖ Slot information is complete and ready for use.
❖ The system now displays it in the availability map for users.
❖ Admin/operator can still update slot settings at this stage.

2.1.3 Ready for Reservation

❖ The system prepares the slot for user bookings.
❖ Internal checks ensure sensors, ANPR/RFID, and status tracking are functioning.
❖ Slot enters the queue of reservable parking spaces.

2.1.4 Pending Reservation

❖ A user has selected the slot and initiated a reservation request.
❖ The system waits for the user to confirm and complete payment.
❖ Slot is temporarily blocked to prevent double-booking.

2.1.5 Reserved

❖ User has completed booking and payment.
❖ Slot is officially reserved for the user during a specific time window.
❖ Reservation countdown timer begins.

2.1.6 Occupied

❖ User vehicle has been detected at the slot (via ANPR, sensors, or RFID).
❖ The parking session is active.
❖ The system tracks duration, status, and potential violations.

2.1.7 Decision Point (Any)

❖ A branching state where the system decides between:
• Normal parking flow → **Exit Verification**
• Error or violation → **Error State**

2.1.8 Error

❖ Triggered when:
• Unauthorized vehicle detected
• Sensor/camera malfunction
• Payment failure or mismatch
• Slot occupancy inconsistency
❖ System requests corrective action or alerts operator.

2.1.9 Exit Verification

❖ System confirms vehicle exit using:
• ANPR checks
• RFID scanning
• Timestamp validation
❖ Compares expected vs actual exit behavior.

2.1.10 Billing & Reporting

❖ System calculates:
• Parking duration
• Charges
• Additional fees (if any)
❖ Generates transaction summary
❖ Operator/admin can review the session log.

2.1.11 Completed

❖ Parking session has successfully concluded.
❖ Reports and logs are stored.
❖ Slot resets to **Available** for the next user.

2.1.12 Cancelled

❖ Reservation may be cancelled by the user or admin before the slot becomes **Occupied**.
❖ Slot is released back to availability.
❖ System records the reservation as cancelled

.

2.2 Explanation of Each Event / Transition

New Slot Added

❖ An admin/operator registers a new parking slot in the system.
❖ Moves system → **Draft**.

Validate Slot Details

❖ Operator validates slot information (slot ID, location, type).
❖ If valid → moves to **Available** state.

Prepare Slot for User Booking

❖ Admin configures slot settings such as pricing, reservation rules, and visibility.
❖ System moves → **Ready for Reservation**.

User Checks Slot Availability

❖ System processes user request and returns real-time slot status:
❖ Free
❖ Occupied
❖ Reserved
❖ If slot is free → stays in Ready state.

User Initiates Booking

❖ User selects a slot and clicks "Reserve Slot."
❖ Moves → **Pending Payment**.

Payment Successful

❖ System receives confirmation from the payment gateway:
❖ UPI / Wallet
❖ Card Payment
❖ Net Banking
❖ Moves → **Slot Reserved**.

Reservation Time Reached / User Arrives

❖ Timer reaches booking start time OR user checks in at entry gate.
❖ Moves → **In Use**.

Vehicle Detected at Entry (Start Timer)

❖ ANPR camera / RFID detects the user’s vehicle entering.
❖ Parking session timer begins.
❖ Moves → **Occupied**.

Slot Misuse Detected
❖ System detects irregularities such as:
❖ Unauthorized vehicle
❖ No payment record
❖ Incorrect license plate
❖ Moves → **Error**.

System Failure Detected
❖ Triggered by:
❖ Sensor malfunction
❖ Camera failure
❖ Slot status mismatch
❖ System moves → **Error**.

User Ends Parking Session
❖ User checks out / vehicle exits the slot.
❖ Moves → **Pending Verification**.

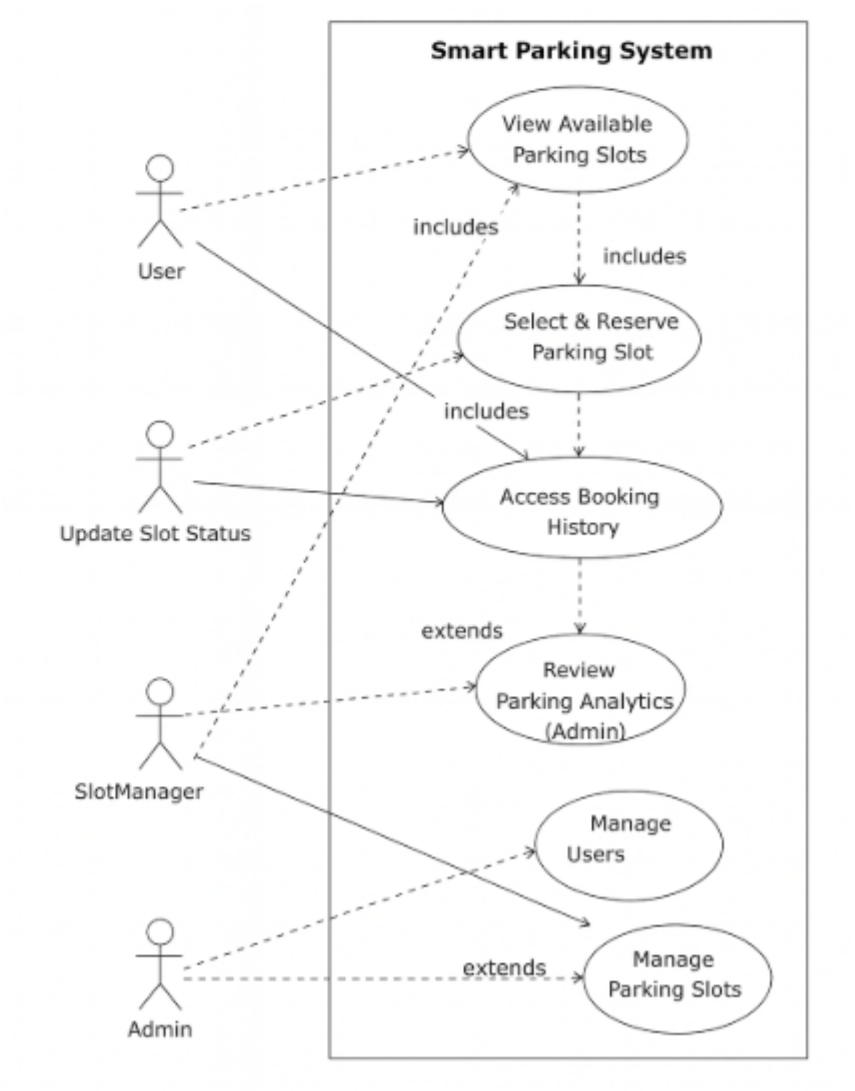
Verification Complete (Exit Confirmed)
❖ System validates exit using:
❖ ANPR scan
❖ RFID card
❖ Camera logs
❖ If validated → moves to **Generate Report**.
❖ If mismatch → stays in verification or triggers Error state.

Report Generated (System Finalize)
❖ Parking summary report generated:
❖ Total duration
❖ Amount paid
❖ Slot usage logs
❖ Moves → **Completed**.

Admin Cancels Reservation
❖ Admin may cancel reservation before check-in.
❖ Moves → **Cancelled**.

Chapter 5: Interaction Modeling

5.1 Use Case Diagram



Relevance of Each Actor

User (Driver)

Drivers are the primary users who search for, reserve, and utilize parking spaces. Their interaction with the system directly influences parking flow, space utilization, and overall system effectiveness.

Parking Operator

Parking operators manage day-to-day slot monitoring, handle verification of vehicle entry/exit, and ensure that the parking facility runs efficiently. They help maintain accurate occupancy data and handle slot-related issues.

Admin

Admins manage users, pricing rules, parking slots, reports, and system configurations to keep the platform functional and compliant. They monitor analytics and ensure that parking operations remain optimized and transparent.

Payment Gateway

The payment gateway securely handles online transactions for parking charges. It ensures smooth processing of digital payments, confirmations, and transaction history updates for users and admins.

Relevance of Each Use Case

Register / Login

Allows users and operators to securely access the system through role-based authentication.

View Available Parking Slots

Provides users with real-time information about free, occupied, and reserved parking spaces.

Select & Reserve Parking Slot

Enables users to choose a preferred slot and submit a reservation request.

Make Payment

Allows users to pay parking fees digitally using UPI, wallets, or card payment methods.

Access Booking History

Gives users access to previous reservations, payment receipts, and parking durations.

Update Slot Status

Allows parking operators to mark slots as occupied, free, under maintenance, or reserved.

Review Parking Analytics (Admin)

Shows admins insights such as occupancy rates, revenue, user activity, and peak hours.

Manage Users

Allows the admin to add, modify, or remove system users and assign access roles.

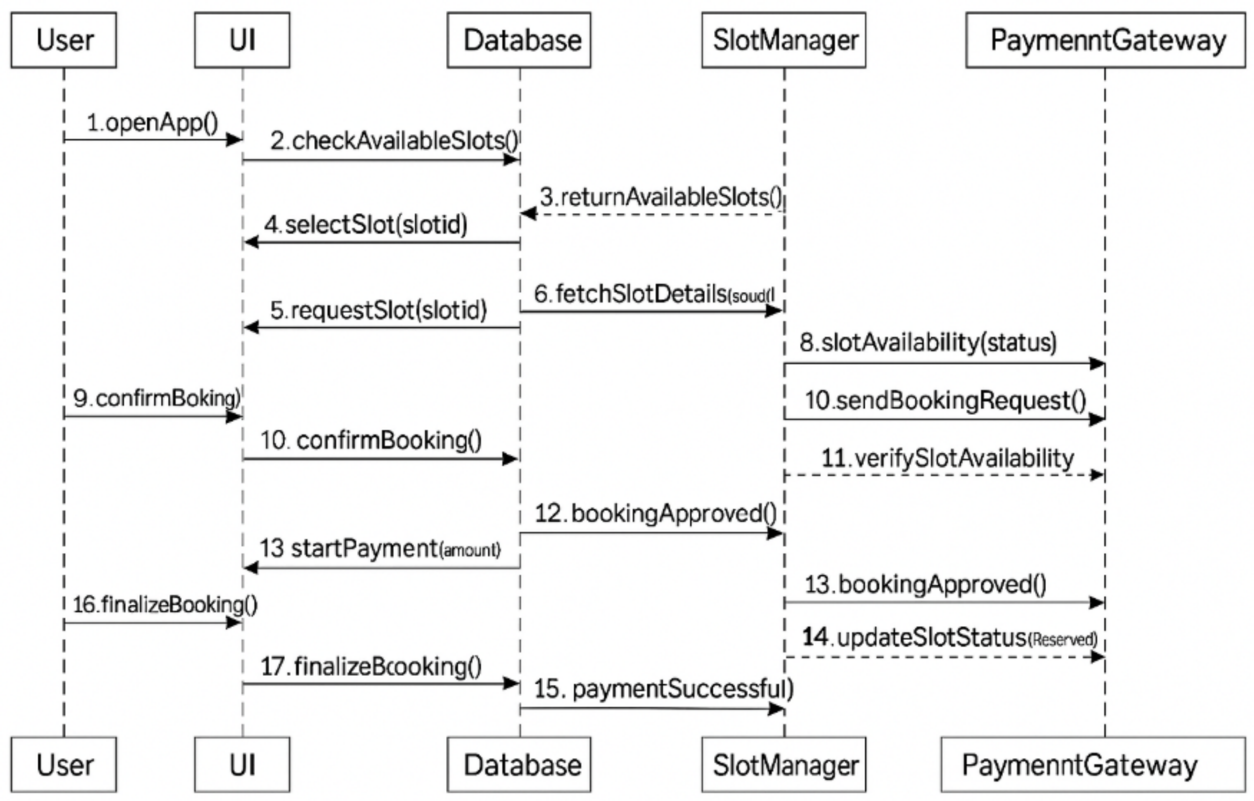
Manage Parking Slots

Enables admins to configure parking areas, slot availability, pricing, and maintenance status.

Generate Reports

Allows admins to extract daily, weekly, or monthly reports on parking usage and revenue.

5.2 Sequence Diagram

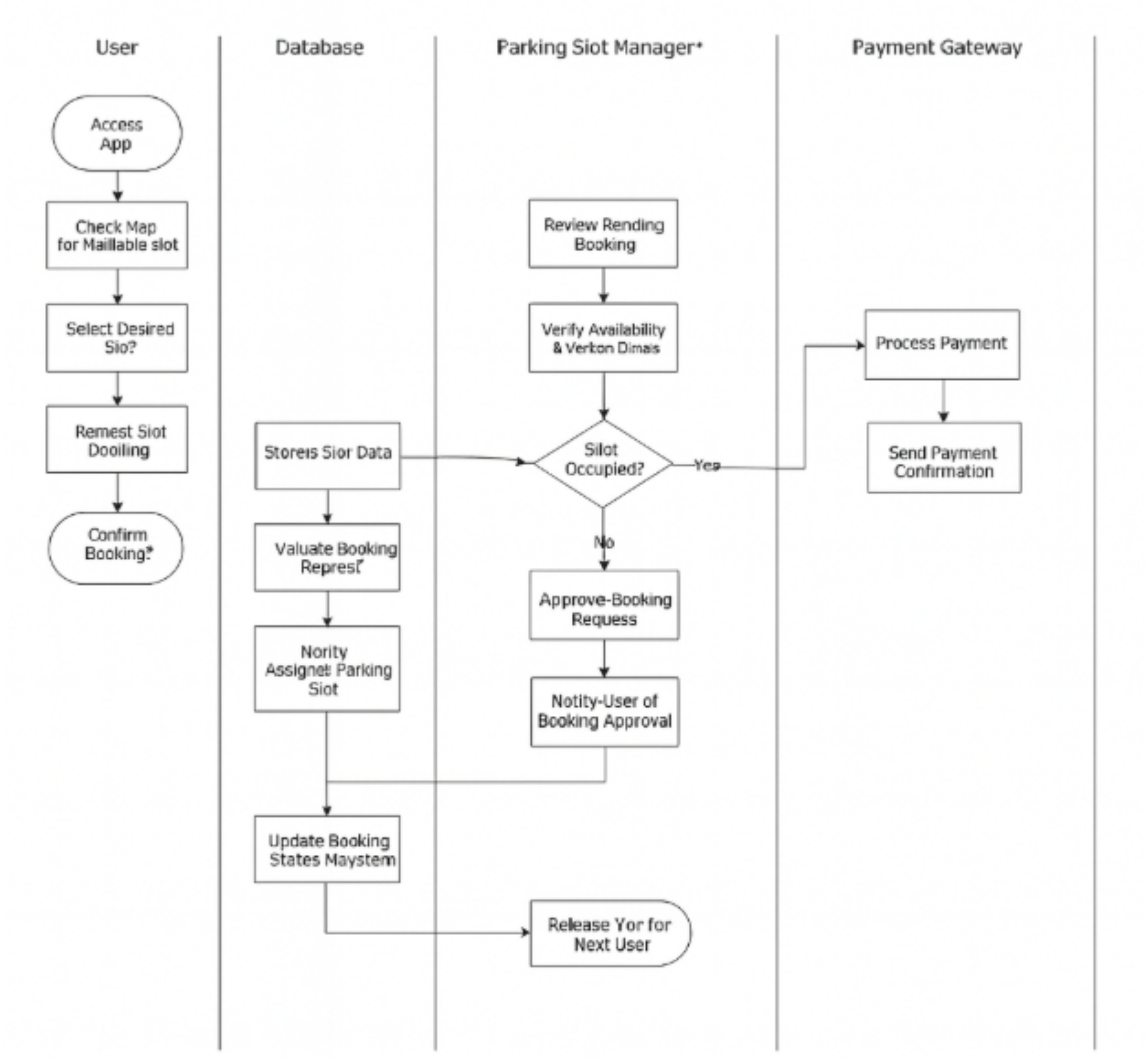


Explanation:

The sequence diagram illustrates how the Smart Parking System manages the user's parking workflow, beginning from opening the application to finalizing a parking reservation. When the user launches the app, the UI requests available parking slots from the database, which retrieves the current slot status and returns it to the interface. After reviewing the options, the user selects a specific parking slot, prompting the UI to forward this request to the SlotManager. The SlotManager then fetches slot details from the database to verify the current occupancy status and returns the availability result to the UI for user confirmation.

The second part of the diagram focuses on the booking and payment interaction. When the user confirms the booking, the UI sends the booking request to the SlotManager, which validates the slot availability once again and, if available, approves the booking. The UI then initiates the payment process by communicating with the PaymentGateway. Once the user completes the payment, the gateway sends a payment confirmation back to the system. The SlotManager then updates the slot status in the database to “Reserved,” finalizing the booking. Finally, a confirmation message is returned to the UI, notifying the user that the parking reservation has been successfully completed.

5.3 Activity Diagram



Explanation:

The activity diagram describes the workflow of a Smart Parking System involving four roles: **User**, **Database**, **Parking Slot Manager**, and **Payment Gateway**. The process begins when a user opens the parking application, checks the map for available slots, and selects a preferred parking space. Once the user confirms the booking request, the system sends the details to the database.

The database immediately stores the booking information, validates the request format, and notifies the parking slot manager. The Parking Slot Manager then reviews the pending booking, verifies slot availability and vehicle details, and decides whether the slot is still free. If the slot is occupied, the user is notified and asked to choose another slot; if it is available, the booking is approved.

After the parking manager approves the booking, the system forwards the payment requirement to the Payment Gateway. The user completes the payment, after which the payment gateway sends confirmation back to the system. The booking status is updated, and the assigned slot is reserved. Once the user completes their parking session, the slot is released for the next user.

Chapter 6: UI Design with Screenshots

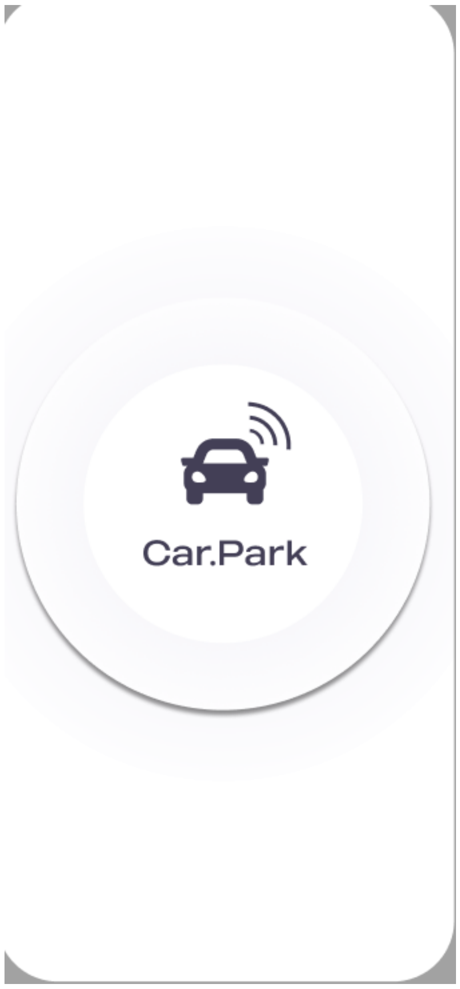


Fig 6.1 Starting page

❖ A secure login interface for a school-focused disaster preparedness system, offering rolebased access and training tools. [Fig 6.1]

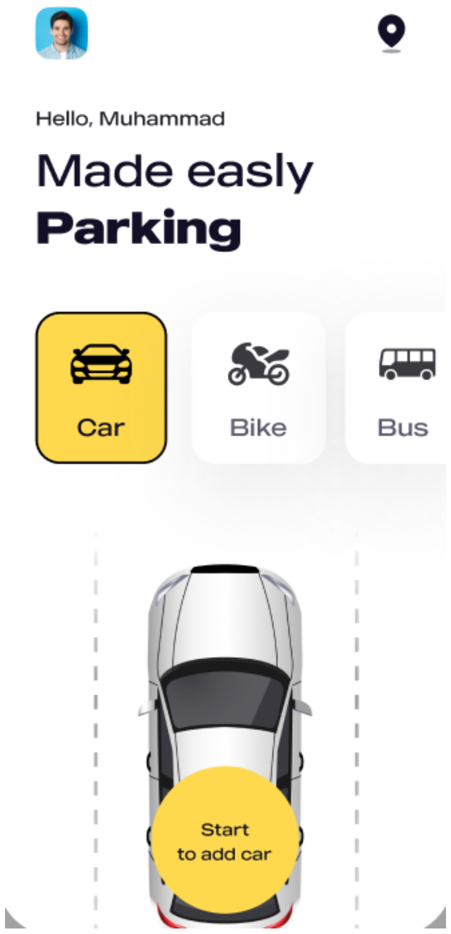
❖ 

Fig 6.2 Dashboard

❖ A sleek dashboard interface for a school smart parking system, offering quick access to type of vehicle. [Fig 6.2]

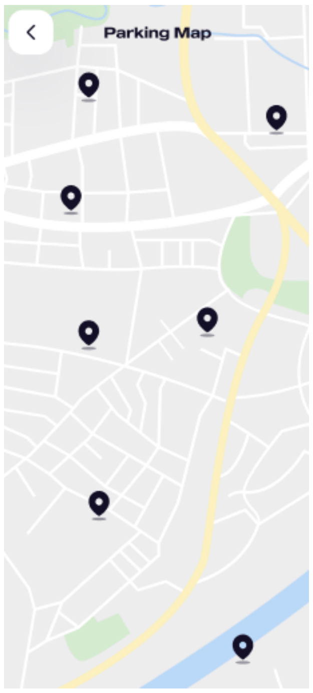


Fig 6.3 available area

❖ A clear SOP interface for looking the available area for parking. [Fig 6.3]

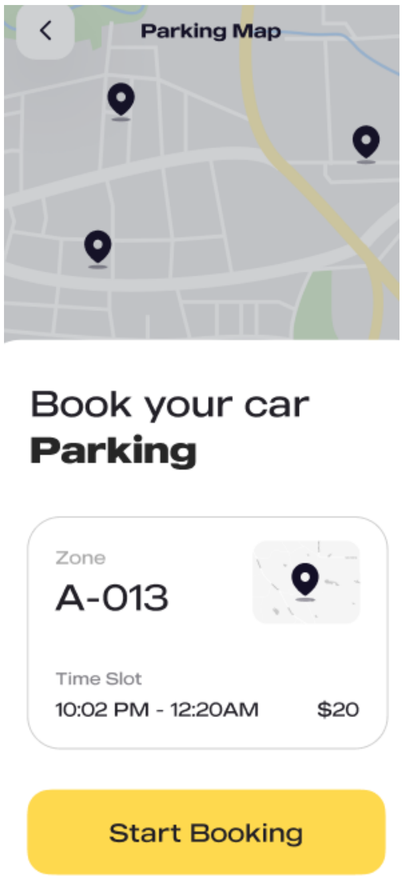


Fig 6.4 Booking page

❖ A figure showing the selected area name and details[Fig 6.4]

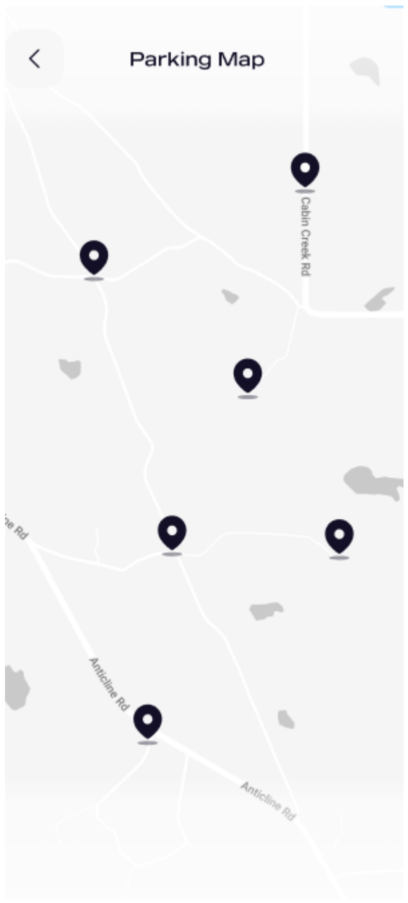


Fig 6.5 Available spots

❖ A real-time alert system for vehicles showing abailable spots for parking[Fig 6.5]

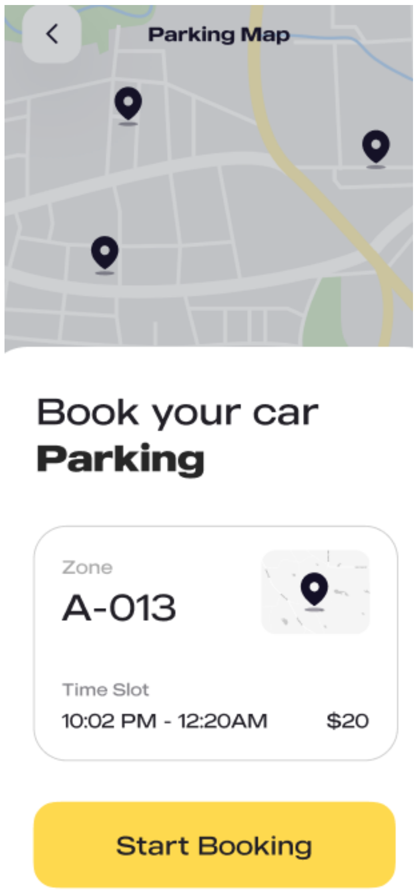


Fig 6.6 To confiem Booking

- A responsive alert interface for confirming the booking [Fig 6.6]

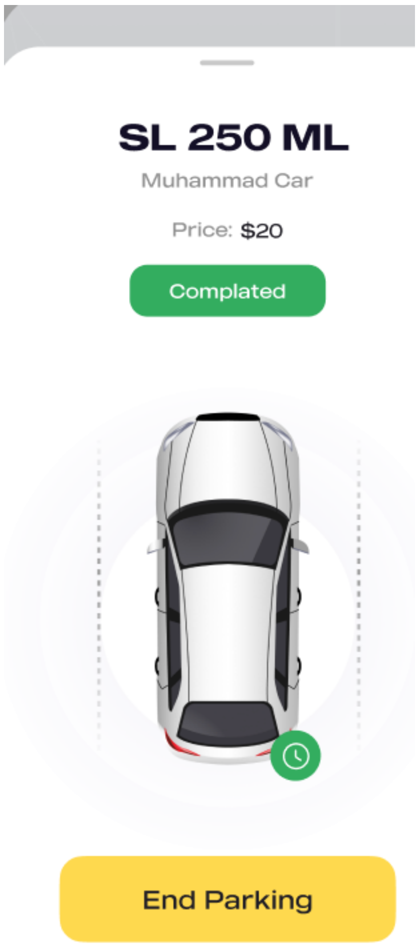
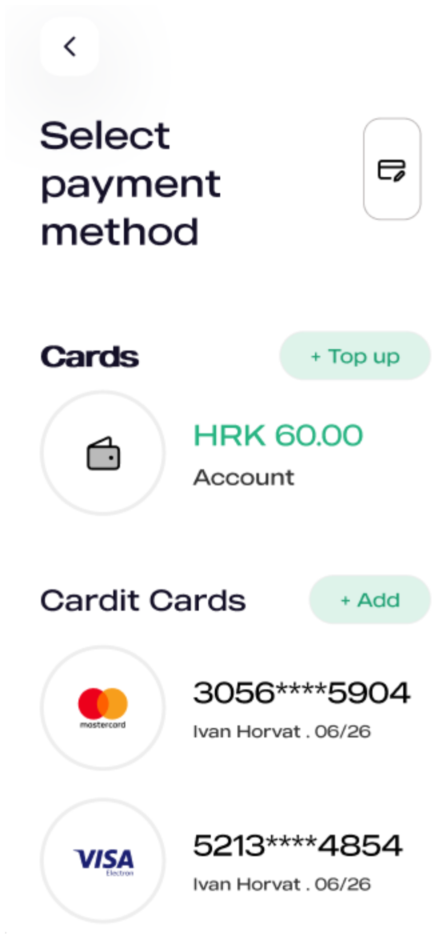


Fig 6.7 Booking Information

❖ An interactive training interface for seeing the cost. [Fig 6.7]



Select payment method

Cards

+ Top up

HRK 60.00

Account

Cardit Cards

+ Add

3056****5904

Ivan Horvat . 06/26

5213****4854

Ivan Horvat . 06/26

Fig 6.8 Paayment Gateway

❖ A interface design for making the payment[Fig 6.8]