SMART CAR PARKING APPLICATION

A project report submitted in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

under the guidance of

DR. DHRUBA KUMAR BHATTACHARYYA

Sr. Professor

TEZPUR UNIVERSITY

Submitted by:

Vikash Kumar- CSB21095

Hrishita Bhuyan- CSB21097

Hirangka Hazarika- CSB21098

Department of Computer Science and Engineering

Tezpur University, Tezpur (784 028), Assam

2024

TEZPUR UNIVERSITY

Certificate from Project Guide

This is to certify that the project report entitled **Smart Car Parking Application**, submitted to the Department of Computer Science and Engineering, Tezpur University, in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, is a record of bona fide work carried out by Mr. Vikash Kumar, CSB21095, Miss Hrishita Bhuyan, CSB21097 and Mr. Hirangka Hazarika, CSB21098 under my supervision and guidance.

All help received by them from various sources have been duly acknowledged.

Place: Tezpur

Date: 28/05/2024

Dr. Dhruba Kumar Bhattacharyya

Sr. Professor

Supervisor

TEZPUR UNIVERSITY

Certificate from Examiner

This is to certify that the project report entitled **Smart Car Parking Application**, submitted to the Department of Computer Science and Engineering, Tezpur University, in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, carried out Mr. Vikash Kumar, CSB21095, Miss Hrishita Bhuyan, CSB21097 and Mr. Hirangka Hazarika, CSB21098, has been examined.

Place: Tezpur

Date: 28/05/2024

Examiner

TEZPUR UNIVERSITY

Declaration

We hereby declare that the Project Report titled **Smart Car Parking Application**, is a result of our independent work and has been prepared by us during February 2024 to May 2024. All the information and content presented in this report is based on our original research, analysis, and implementation. Any references and sources of information used have been duly acknowledged and cited.

We further declare that this project report has not been submitted to any other academic institution or published in any form. The intellectual property rights of this project and the accompanying software application solely belongs to us, unless otherwise indicated.

We take full responsibility for the accuracy, completeness, and integrity of the contents presented in this report.

Place: Tezpur

Date: 28/05/2024

Vikash Kumar- CSB21095
Hrishita Bhuyan- CSB21097
Hirangka Hazarika-CSB21098

Acknowledgements

We would like to take this opportunity to convey my sincere gratitude to Dr. Dhruba K. Bhattacharyya, who served as our project supervisor and guide, for his exceptional guidance, constant support, and essential mentorship during the course of this project. This Smart Car Parking Application has benefited much from his in-depth expertise, incisive comments, and professional guidance. We are truly indebted to him for his patience, encouragement, and the countless hours he dedicated to reviewing and refining the project. His never-ending encouragement and faith in my competence have served as an inspiration to us.

Additionally, we would like to express my profound gratitude to Dr. Sarat Saharia, who is the Head of the Department of Computer Science at Tezpur University. His inspiring guidance, encouragement, and support have been instrumental in fostering an academic setting that is conducive to the success of initiatives this inventive. His advice and commitment to developing new software development talent have been priceless.

In addition, we want to thank my friends for their contributions to this project through brainstorming sessions, teamwork, and constructive criticism. Their combined efforts and differing viewpoints have enhanced the software development process and produced a solid software solution.

Finally, we want to thank our families for their unwavering support, patience, and inspiration during this journey. Our accomplishments have been propelled by their confidence in us and ongoing encouragement.

Place: Tezpur

Date: 28/05/2024

Vikash Kumar- CSB21095
Hrishita Bhuyan- CSB21097
Hirangka Hazarika-CSB21098

CONTENT

1. INTRODUCTION

In modern urban environments, finding a parking spot can be a challenging and stressful endeavor for drivers. Traditional parking systems typically require drivers to enter parking facilities to locate and book a spot, often resulting in inefficiencies, increased congestion, and frustration. Moreover, the necessity to physically pay for parking adds another layer of inconvenience, particularly in busy locations. These issues highlight the need for a more streamlined and user-friendly solution that allows drivers to book and pay for parking in advance without entering the parking area first.

To address these challenges, we propose the development of a Smart Car Parking system that leverages IoT technology and a mobile application to provide real-time data on parking availability, streamline the booking and payment process, and enhance the overall parking experience for users. This project aims to create a seamless and efficient parking solution that significantly reduces the time spent searching for parking spaces, alleviates traffic congestion, and improves user satisfaction.

The Smart Car Parking system will utilize IoT sensors to detect and display available parking spaces in real-time. Users will be able to reserve parking spaces, make online payments, and receive navigation assistance through a mobile application developed using Flutter and Dart. By integrating modern technologies, this system seeks to optimize parking space usage and provide a hassle-free experience for drivers.

This document outlines the requirements, functionalities, and specifications of the Smart Car Parking system, providing a comprehensive guide for its development. The project aims to deliver a reliable, user-friendly, and efficient parking management solution that meets the needs of urban drivers and enhances the overall efficiency of urban parking infrastructure.

2. PROBLEM STATEMENT

2.1. In urban areas,

- Finding a parking spot can be a time-consuming and stressful task for drivers.
- Traditional parking systems often require drivers to enter the parking area before they can book a slot, leading to inefficiencies and increased congestion.
- Furthermore, the need to physically pay for parking after arriving adds to the inconvenience, especially in busy locations.
- There is a clear need for a more streamlined and user-friendly solution that allows drivers to book and pay for parking in advance, without the need to physically enter the parking area first.
-

2.2. Motivation:

Current parking management systems are inefficient in handling the process of booking and payment, causing frustration among drivers and contributing to traffic congestion. Key issues include:

- Time Consumption: Drivers often spend a significant amount of time searching for available parking spots, particularly in crowded urban areas.
- Inefficient Booking Systems: Traditional parking systems typically require drivers to enter the parking facility before they can book a slot, which can lead to overcrowding and increased waiting times.
- Payment Inconvenience: The necessity to make payments after parking adds another layer of inconvenience, especially if there are long queues or malfunctioning payment machines.
- Traffic Congestion: Inefficient parking management contributes to traffic congestion, as drivers circulate the area in search of available spots.
- User Dissatisfaction: The combined effect of these issues leads to a poor user experience, causing frustration and dissatisfaction among drivers.

2.3. Proposed solution:

The smart parking management application will leverage modern technologies, including IoT sensors, to provide real-time data on parking availability and integrate with a mobile application developed using Flutter and Dart. This application will:

- Allow users to book parking slots for a specified duration before arriving at the parking facility.
- Facilitate online payments for the reserved slots, ensuring a hassle-free parking experience.
- Provide real-time updates on slot availability, ensuring users have up-to-date information.

3. FEASIBILITY REPORT

3.1.Software Requirements:

- Arduino : To write program of this project we are using Arduino IDE with Arduino language.
- FLUTTER : To make an Android app we are using the Flutter framework which is based on Dart programming language.
- MySQL : We are using MySQL to store data sent by the NodeMCU, and this data will be received through an application.

3.2.Hardware Requirements:

- Nodemcu: it is a microcontroller with inbuilt Esp8266 Wi-Fi, with the help of Nodemcu we can send and data to the server.
- IR Obstacle Sensor: when any object comes in front of this sensor LEDs, then the sensor will send 1 to the nodemcu.
- Battery: 12v Power supply is used.
- Register: preferably 250 Ohm register to safe led lights.
- LED light: We can use 3 coloured LED lights to represent occupancy. Red for occupied area, Yellow for booked but not occupied area and Green for available parking slot.

3.3. The feasibility that is being looked forward to is as follows:

1. Technical Feasibility:

- Sensors and Cameras: Use of sensors and cameras for real-time monitoring and detection of car sizes.
- IoT Technology: Integration of IoT devices for communication and data processing.
- Mobile Application: Development of a user-friendly mobile application for booking and payment.

2. Economic Feasibility:

- Cost of Implementation: Initial investment in sensors, cameras, and IoT infrastructure.
- Revenue Generation: Potential revenue from parking fees and partnerships with local businesses.

3. Operational Feasibility:

- Ease of Use: User-friendly interface for drivers to book parking spaces.
- Maintenance: Regular maintenance of sensors and cameras to ensure accurate detection.
- Training: Training for staff to manage the system effectively.

4. Legal Feasibility:

- Regulatory Compliance: Compliance with local parking regulations and privacy laws.
- Liability: Addressing liability issues in case of accidents or system failures.

4. SOFTWARE REQUIREMENT SPECIFICATIONS

Introduction: The purpose of this document is to define and describe the requirements of the Smart Car Parking system and to outline its functionality and constraints.

Scope of this Document : The scope of this Software Requirements Specification (SRS) document is to provide a comprehensive understanding of the requirements and specifications for the Smart Car Parking system using Flutter, Dart, and IoT technologies. It will guide the development process to ensure that the final product meets the desired objectives and standards.

Overview: The objective of this project is to develop a smart car parking system that leverages IoT technology to provide users with real-time information about available parking spaces. The system aims to optimize parking space usage, reduce the time spent searching for parking, and enhance the overall user experience through a user-friendly mobile application developed using Flutter and Dart.

4.1.General Description

Product Functions

The Smart Car Parking system aims to provide a seamless and efficient parking experience for users. Key functions include:

• Real-time detection and display of available parking spaces using IoT sensors.

• Reservation of parking spaces through the mobile application.

• Navigation to available or reserved parking spaces.

• Payment processing for parking fees.

• Monitoring and managing parking space occupancy.

Similar System Information

The Smart Car Parking system is developed using Flutter for cross-platform compatibility, similar to other modern mobile applications. Its distinguishing feature is the integration with IoT sensors for real-time parking space detection and management.

User Characteristics

Users of the Smart Car Parking system include drivers seeking to find and reserve parking spaces. They are typically familiar with using mobile applications and basic navigation systems. No specific technical expertise is required, but users should have a basic understanding of mobile app interfaces.

User Problem Statement

Current parking systems are often inefficient, leading to prolonged search times for parking spaces and frustration among drivers. The lack of real-time information about parking availability results in wasted time and increased traffic congestion.

User Objectives

Users seek a mobile application that provides real-time information about available parking spaces, allows for easy reservation and payment, and offers navigation assistance to the

General Constraints

Constraints for the Smart Car Parking system include:

• The need for an intuitive and user-friendly interface.

• Compatibility with both Android and iOS platforms.

• Integration with IoT sensors for real-time data collection.

• Development using Flutter and Dart to ensure cross-platform functionality and ease of maintenance.

4.2.Functional Requirements

Data Storage

• The system shall store user data, parking space availability, reservations, and transaction records in a central database.

• Data shall be synchronized with the cloud to ensure accessibility and reliability.

• Criticality: Very high, as accurate and reliable data storage is essential for the system's functionality.

• Mitigation strategies for limited network availability include implementing local caching and offline storage capabilities.

Data Accessibility

• Users shall be able to access real-time information about available parking spaces through the mobile application.

• The system shall provide features to query and browse parking availability, reservations, and transaction history.

• Criticality: Very high, as data accessibility is crucial for user satisfaction and system reliability.

• The system must ensure seamless data synchronization between the mobile application and the central database.

Data Manipulation

• The application shall provide forms and interfaces for users to make and manage reservations, payments, and profile settings.

• Users should be able to easily navigate, reserve, and pay for parking spaces through intuitive interfaces.

• Criticality: Very high, as user interaction with the system depends on efficient data manipulation.

• Usability issues can be mitigated by providing clear instructions, intuitive design, and user training resources.

4.3. **Interface Requirements**

GUI

The user interface for the Smart Car Parking application is designed using Flutter, ensuring a visually appealing and interactive experience across both Android and iOS platforms. Flutter provides a wide range of components and libraries for building responsive and user-friendly interfaces.

CLI

There is no command-line interface for the Smart Car Parking application.

API

The application shall provide APIs for integrating with IoT sensors and payment gateways, facilitating real-time data collection and transaction processing.

Diagnostics or ROM

The mobile application includes a help section and troubleshooting guides to assist users with common issues.

Hardware Interfaces

The system primarily interacts with IoT sensors deployed in parking spaces to detect occupancy status. It also utilizes the hardware components of the user's mobile device, such as GPS for navigation and network interfaces for data synchronization.

Communications Interfaces

The application relies on standard network protocols (e.g., HTTP, HTTPS) for communication with external servers, including cloud databases and payment gateways. It also uses MQTT or similar protocols for real-time communication with IoT sensors.

Software Interfaces

The application integrates with third-party services for functionalities such as payment processing, map services for navigation, and cloud storage for data synchronization.

4.4. Other Non-Functional Requirements

Security

User authentication and data transfer will be secured using HTTPS and other industry-standard encryption methods. User data will be securely stored in the backend database with access controls in place.

Binary Compatibility

The application should be compatible with both Android and iOS platforms.

Reliability

The application should be stable and reliable, with minimal crashes or downtime.

Maintainability

Code should be well-structured and documented to facilitate ease of maintenance and future updates.

Portability

The application should be easily deployable on different devices and operating systems.

Extensibility

The architecture should allow for easy integration of new features and enhancements.

Reusability

Components and modules should be designed for reuse in other parts of the application.

Application Affinity/Compatibility

The application should be compatible with other popular applications and services, particularly navigation and payment services.

Resource Utilization

The application should optimize resource usage to ensure efficient performance and responsiveness.

Serviceability

The application should be easy to troubleshoot and maintain, with built-in diagnostic tools and clear documentation.

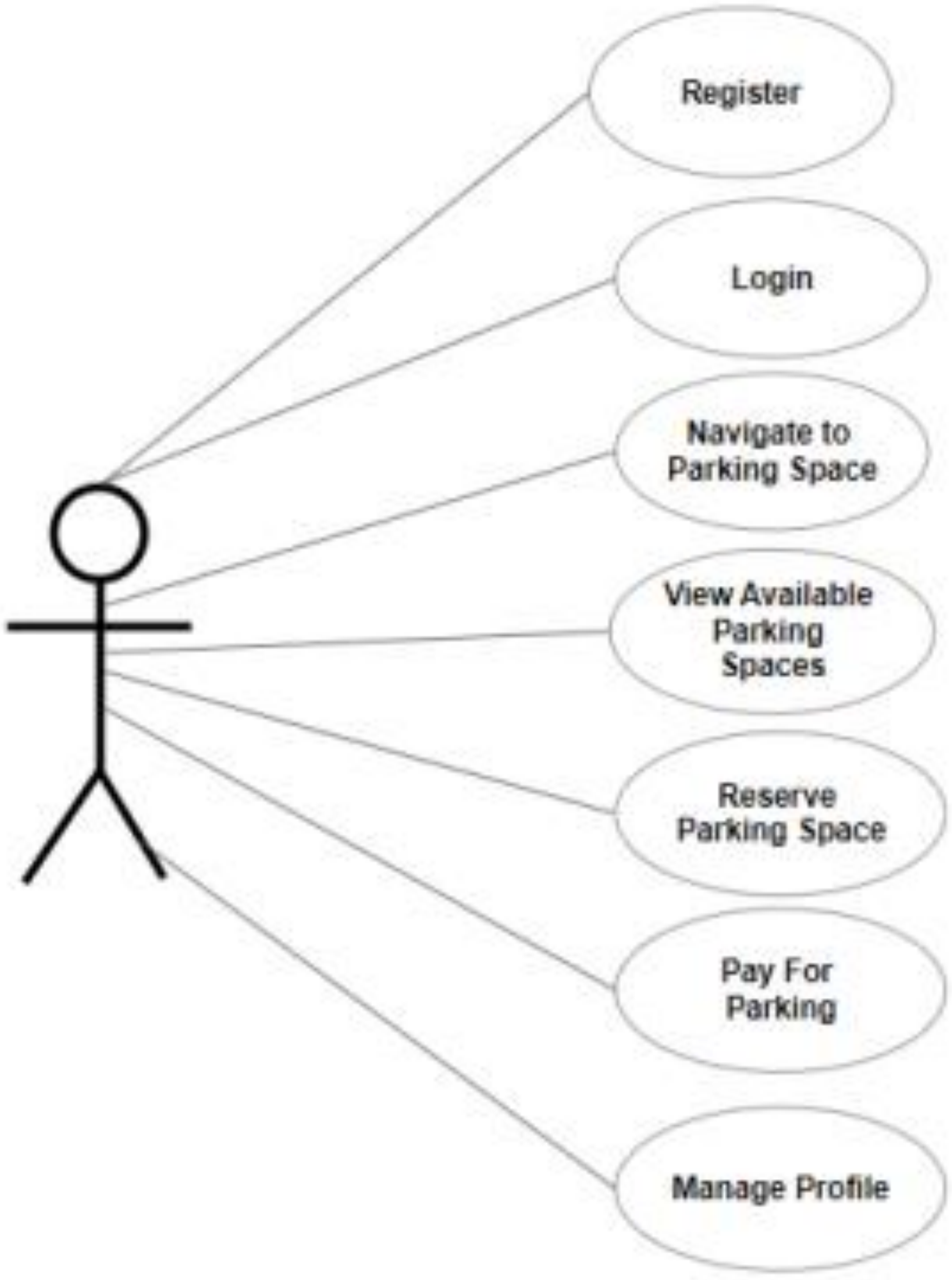
4.5.Operational Scenarios

Users can register and log in to the application using their email or social media accounts. They can then view available parking spaces, make reservations, navigate to reserved spaces, and manage their profile settings and transaction history.

4.6.Preliminary Use Case Models and Sequence Diagrams

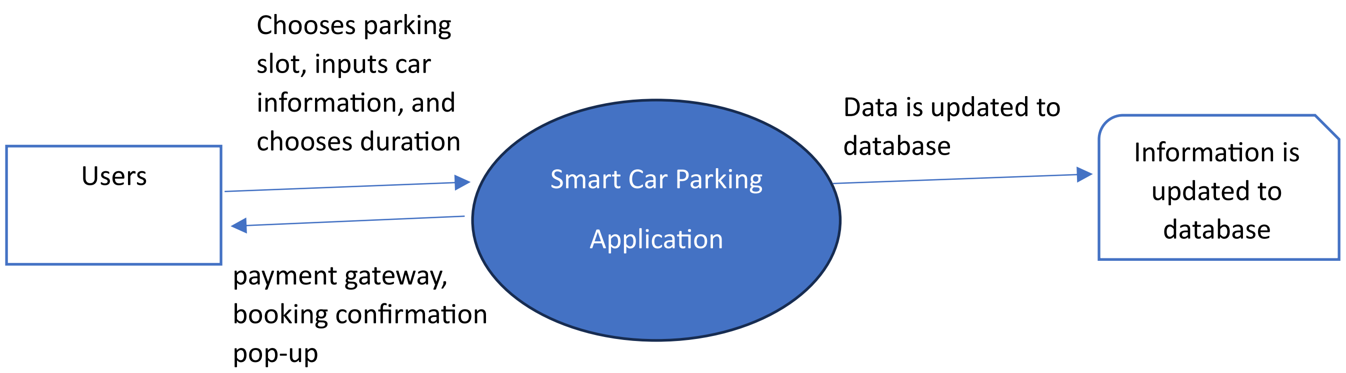
This section will present the fundamental sequence diagrams and use cases that satisfy the system's requirements. These diagrams will provide a structural view of how the requirements are fulfilled within the system.

Use Case Model

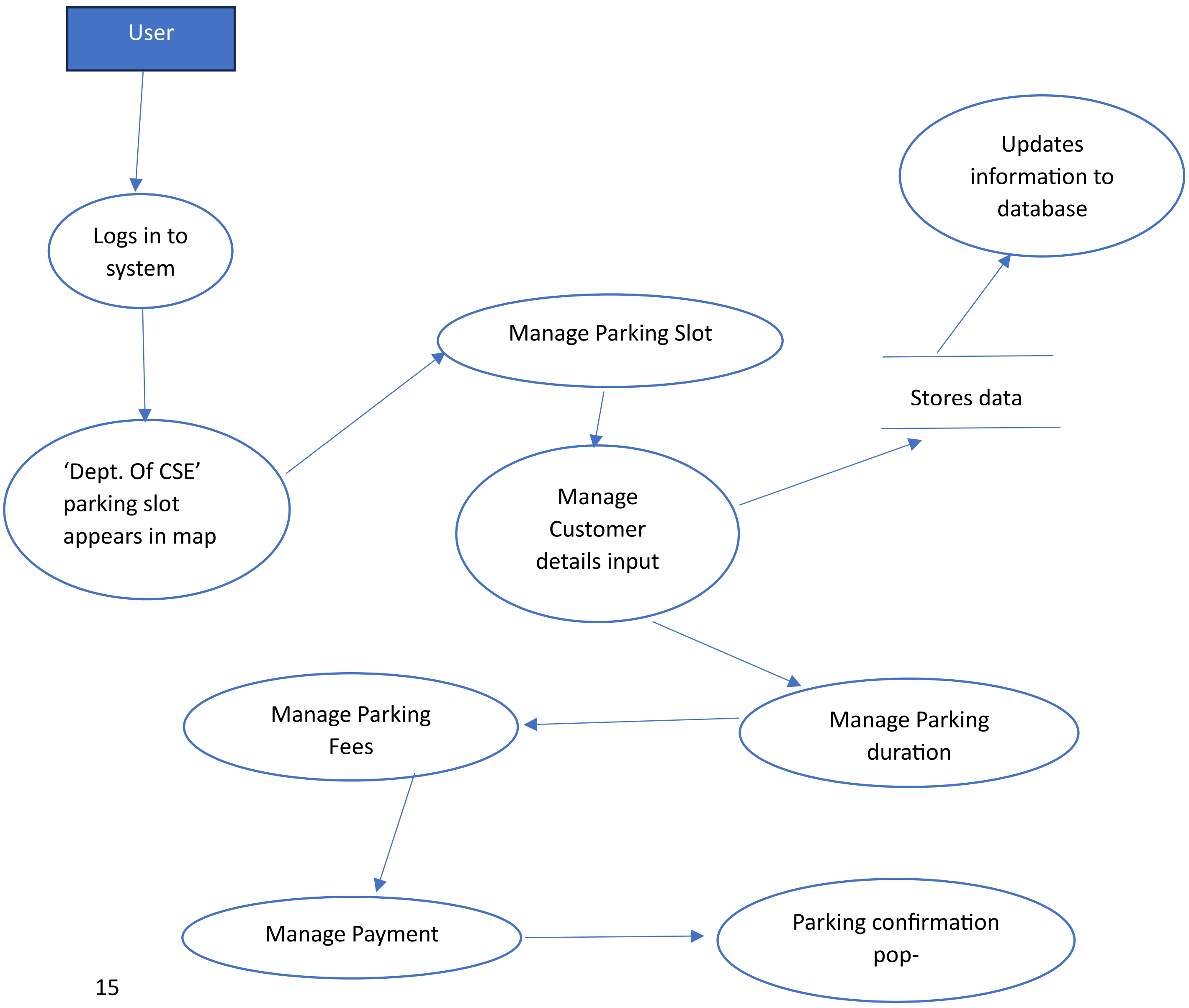


5. DESIGN DOCUMENT

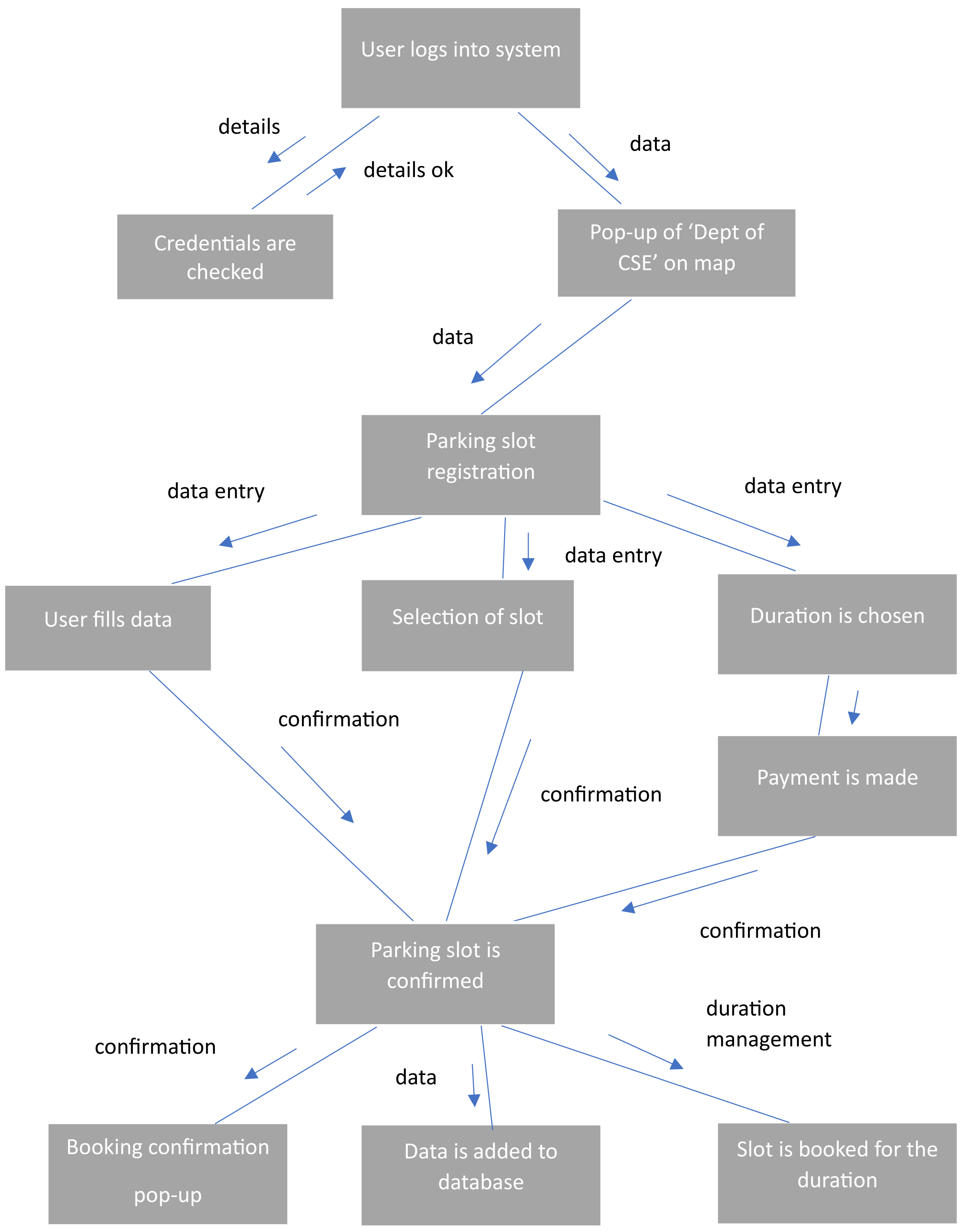
Level-0 DFD


Chooses parking slot, inputs car information, and chooses duration
Users
Smart Car Parking Application
Data is updated to database
Information is updated to database
payment gateway, booking confirmation pop-up

Level-1 DFD


User
Logs in to system
'Dept. Of CSE' parking slot appears in map
Manage Parking Slot
Updates information to database
Stores data
Manage Customer details input
Manage Parking Fees
Manage Parking duration
Manage Payment
Parking confirmation pop-

Structured Chart



Data Directory

- User Registration: Captures new user information and stores it in the User Data Store.

Inputs: User details (name, email, password, etc.)

Outputs: Registration confirmation

- User Login: Authenticates users by checking the credentials against the User Data Store.

Inputs: Username, password

Outputs: Login status (success/failure)

- Parking Slot Selection: Allows the user to select a parking slot.

Inputs: User's preferred slot

Outputs: Selected slot information

- Car Information Input: Collects car details from the user.

Inputs: Car make, model, license plate number

Outputs: Car information stored

- Duration Selection: User selects the duration for parking.

Inputs: Duration time

Outputs: Selected duration information

- Payment Processing: Handles payment for the parking slot via the payment gateway.

Inputs: Payment details

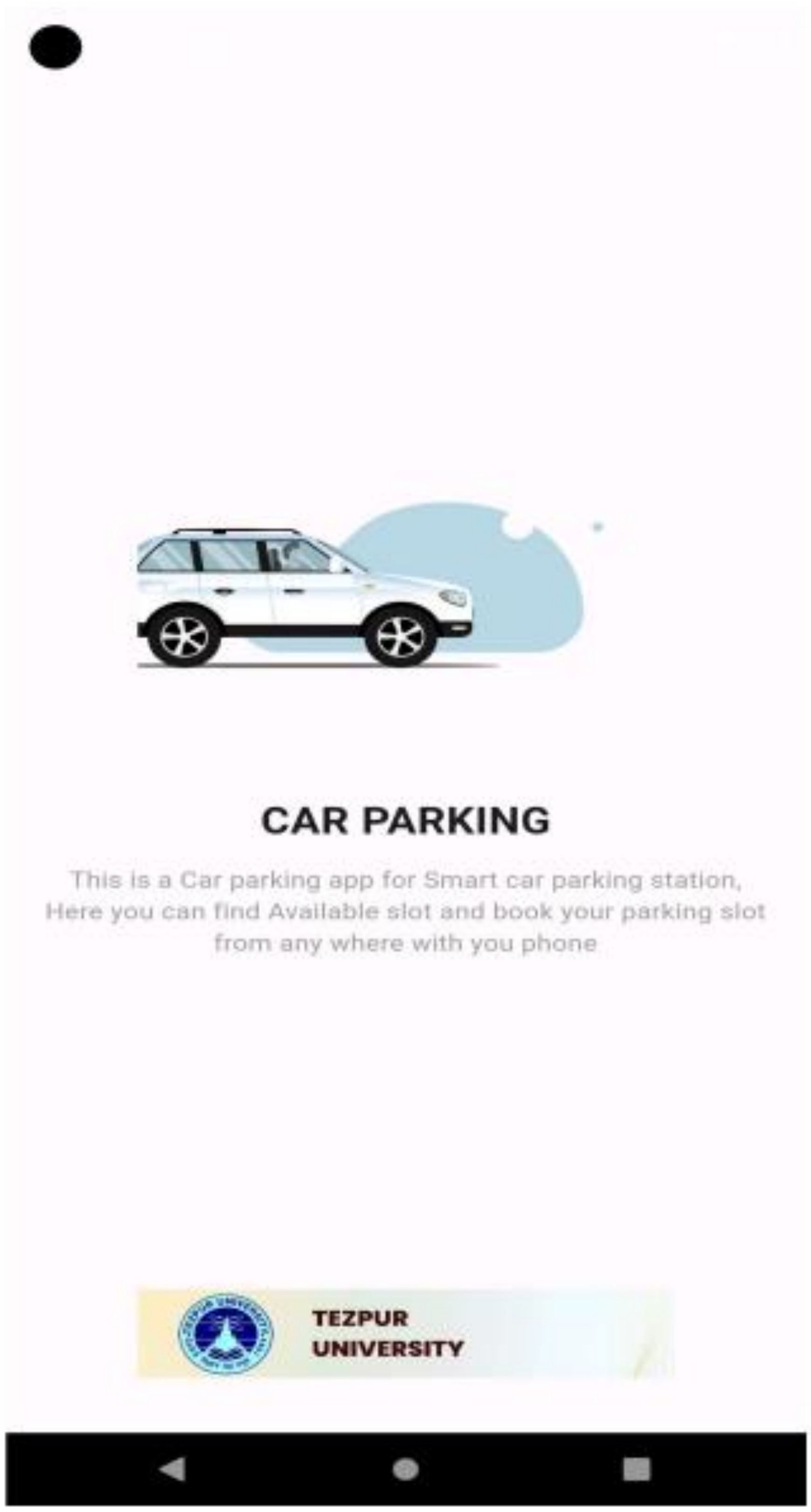
Outputs: Payment status (success/failure)

- Booking Confirmation: Confirms the booking and provides a confirmation message to the user.

Inputs: Booking details (slot, car info, duration, payment status)

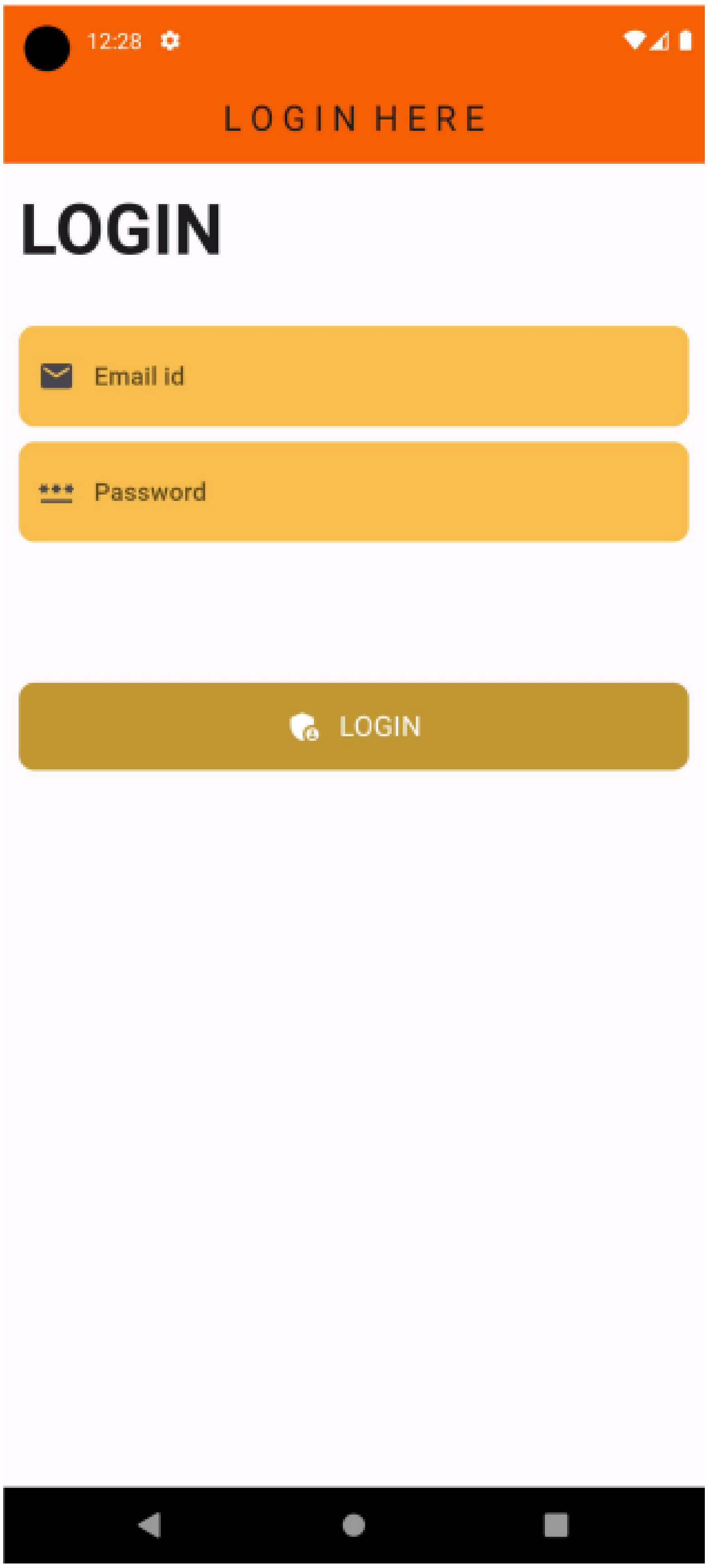
Outputs: Booking confirmation message

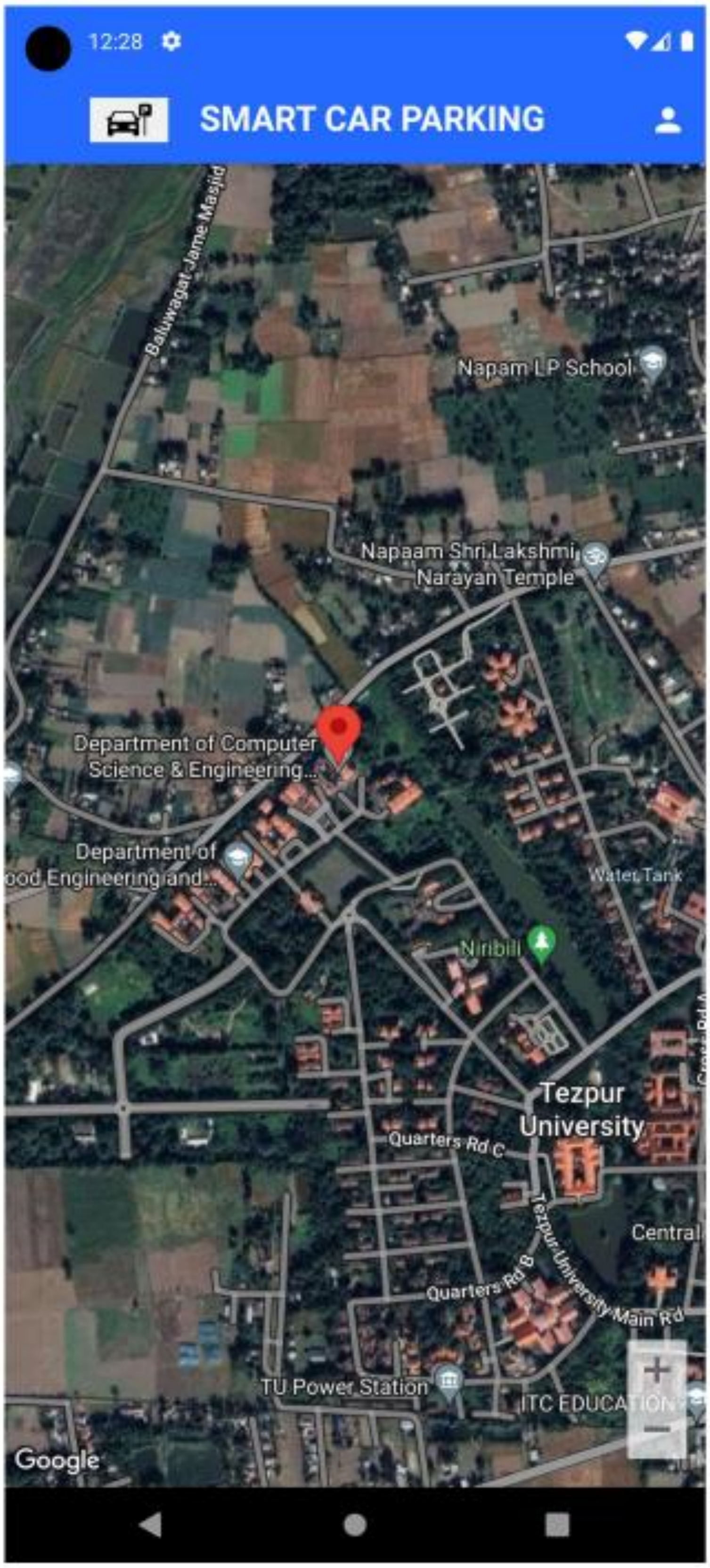
6. SYSTEM IMPLEMENTATION

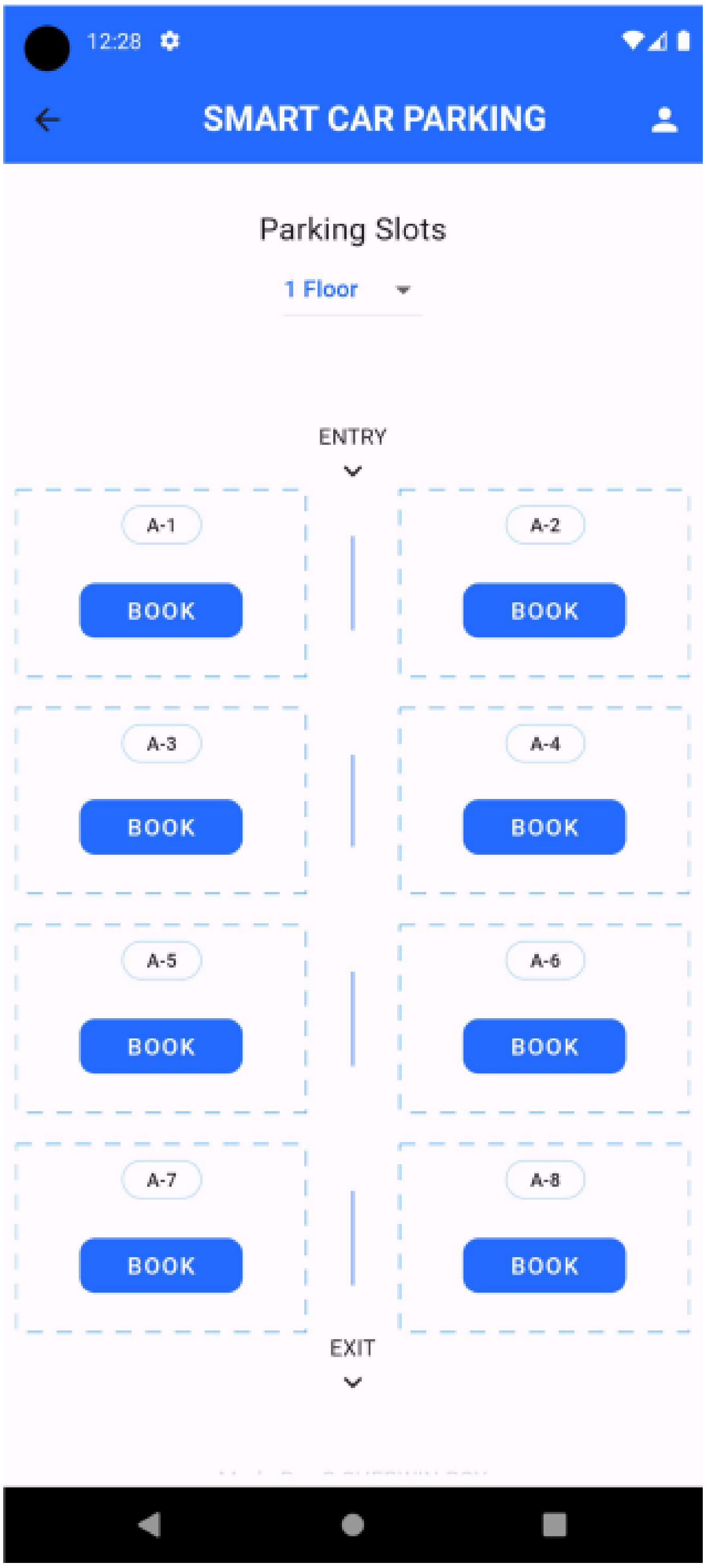


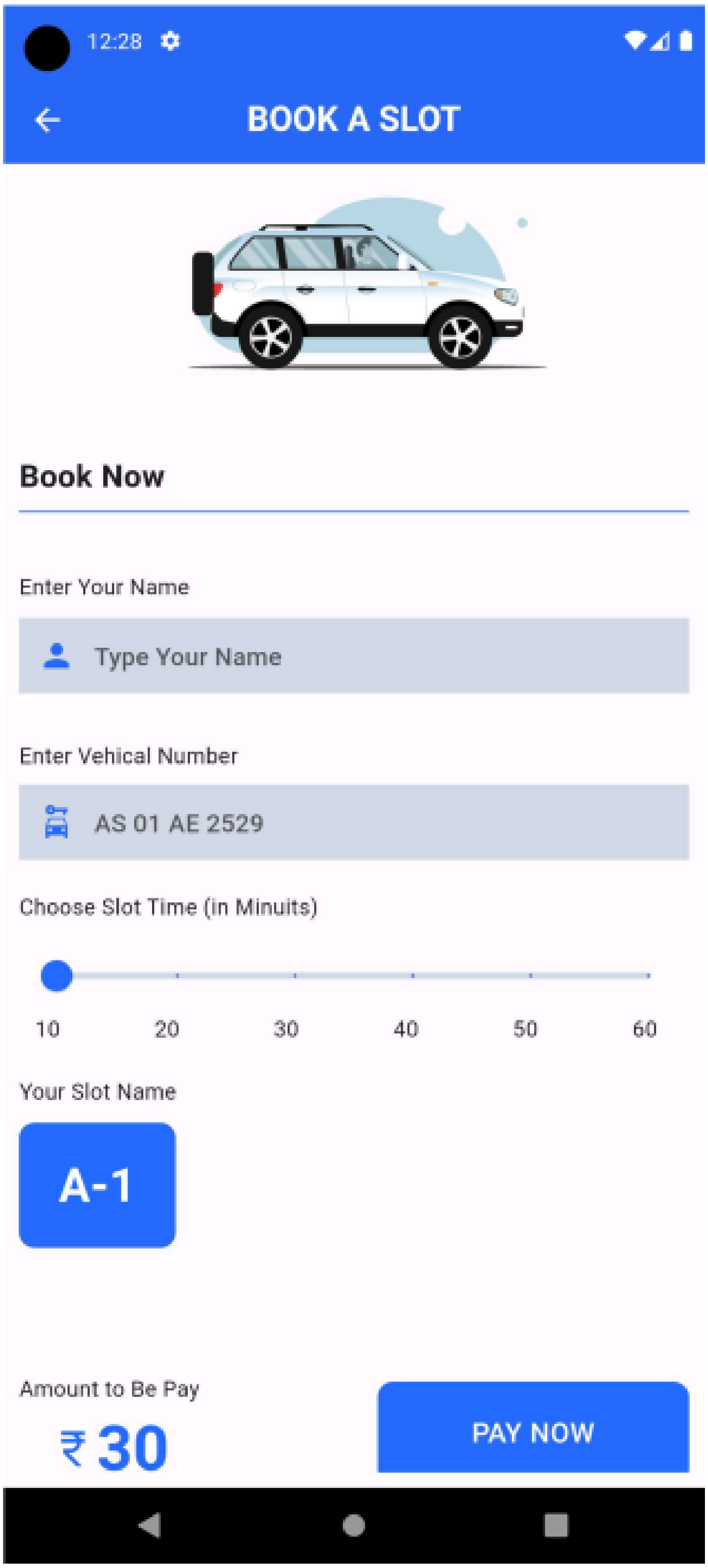
CAR PARKING

This is a Car parking app for Smart car parking station, Here you can find Available slot and book your parking slot from any where with you phone


12:28
LOGIN HERE
LOGIN
Email id
Password
LOGIN


12:28
SMART CAR PARKING
Baluwagat Jame Masjid
Napam LP School
Napaam Shri Lakshmi Narayan Temple
Department of Computer Science & Engineering...
Department of ood Engineering and...
Water Tank
Niribili
Tezpur University
Quarters Rd C
Central
Quarters Rd B
Tezpur University Main Rd
TU Power Station
ITC EDUCATION
Google


12:28
SMART CAR PARKING
Parking Slots
1 Floor
ENTRY
A-1
BOOK
A-2
BOOK
A-3
BOOK
A-4
BOOK
A-5
BOOK
A-6
BOOK
A-7
BOOK
A-8
BOOK
EXIT


12:28

BOOK A SLOT

Book Now

Enter Your Name

Type Your Name

Enter Vehical Number

AS 01 AE 2529

Choose Slot Time (in Minuits)

10 20 30 40 50 60

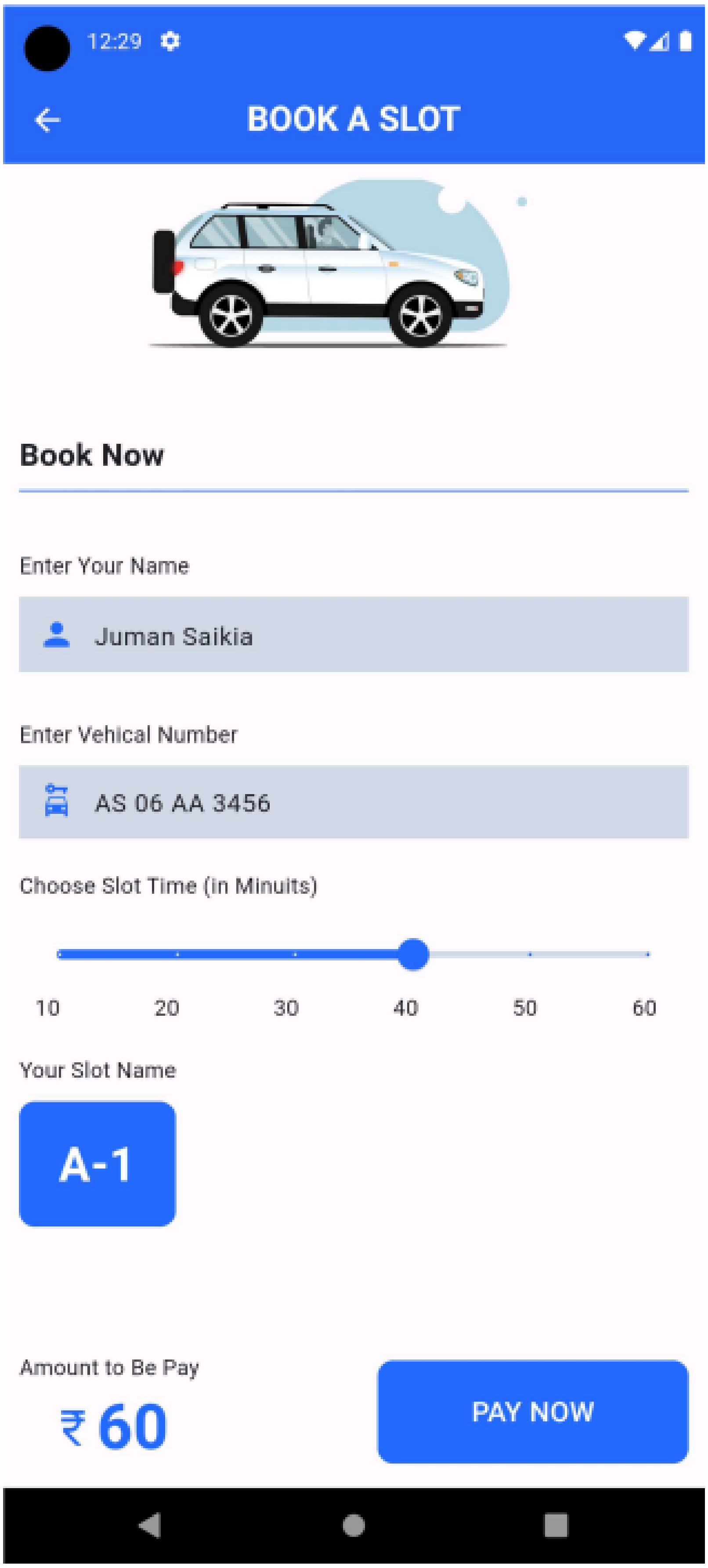
Your Slot Name

A-1

Amount to Be Pay

₹ 30

PAY NOW



12:29

BOOK A SLOT

Book Now

Enter Your Name

Juman Saikia

Enter Vehical Number

AS 06 AA 3456

Choose Slot Time (in Minuits)

10 20 30 40 50 60

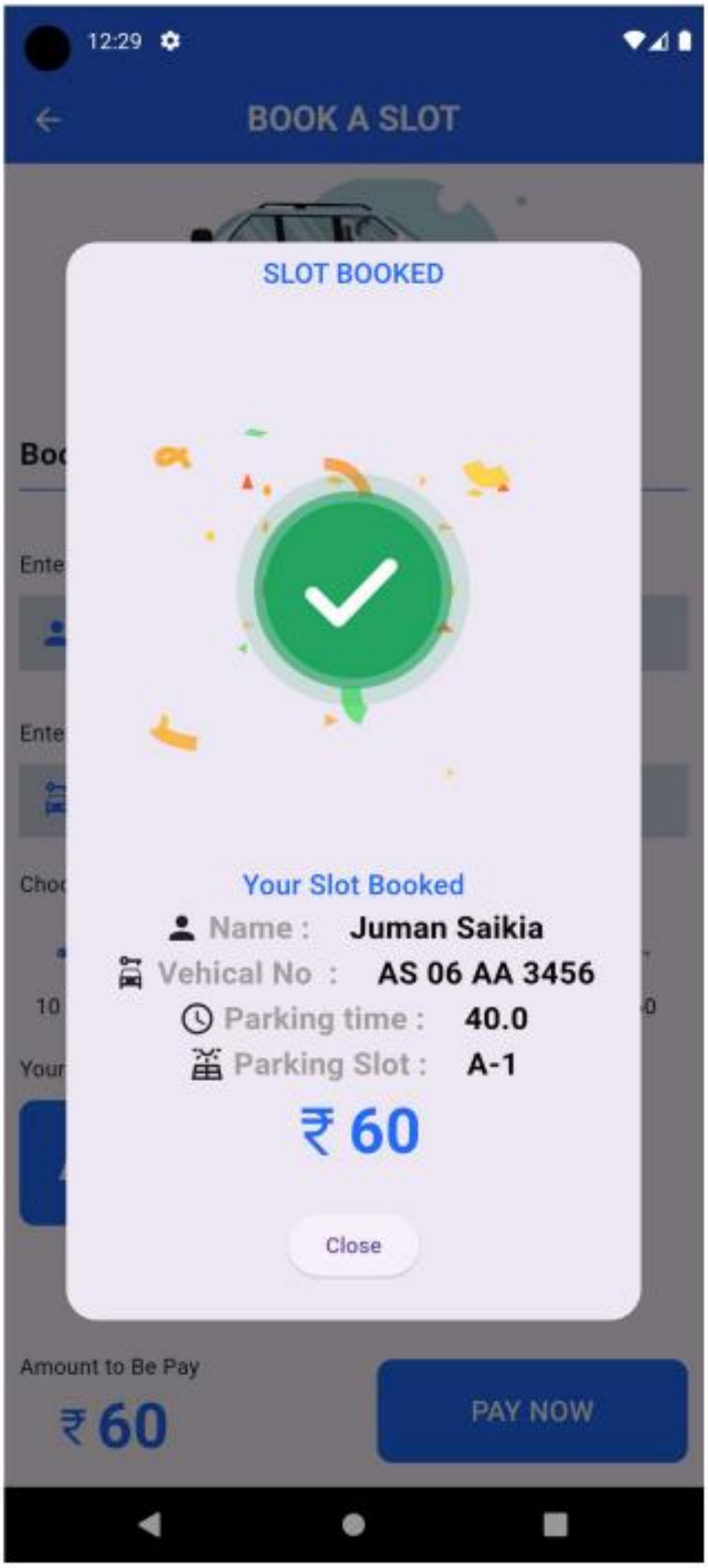
Your Slot Name

A-1

Amount to Be Pay

₹ 60

PAY NOW



Your Slot Booked

Name : Juman Saikia

Vehical No : AS 06 AA 3456

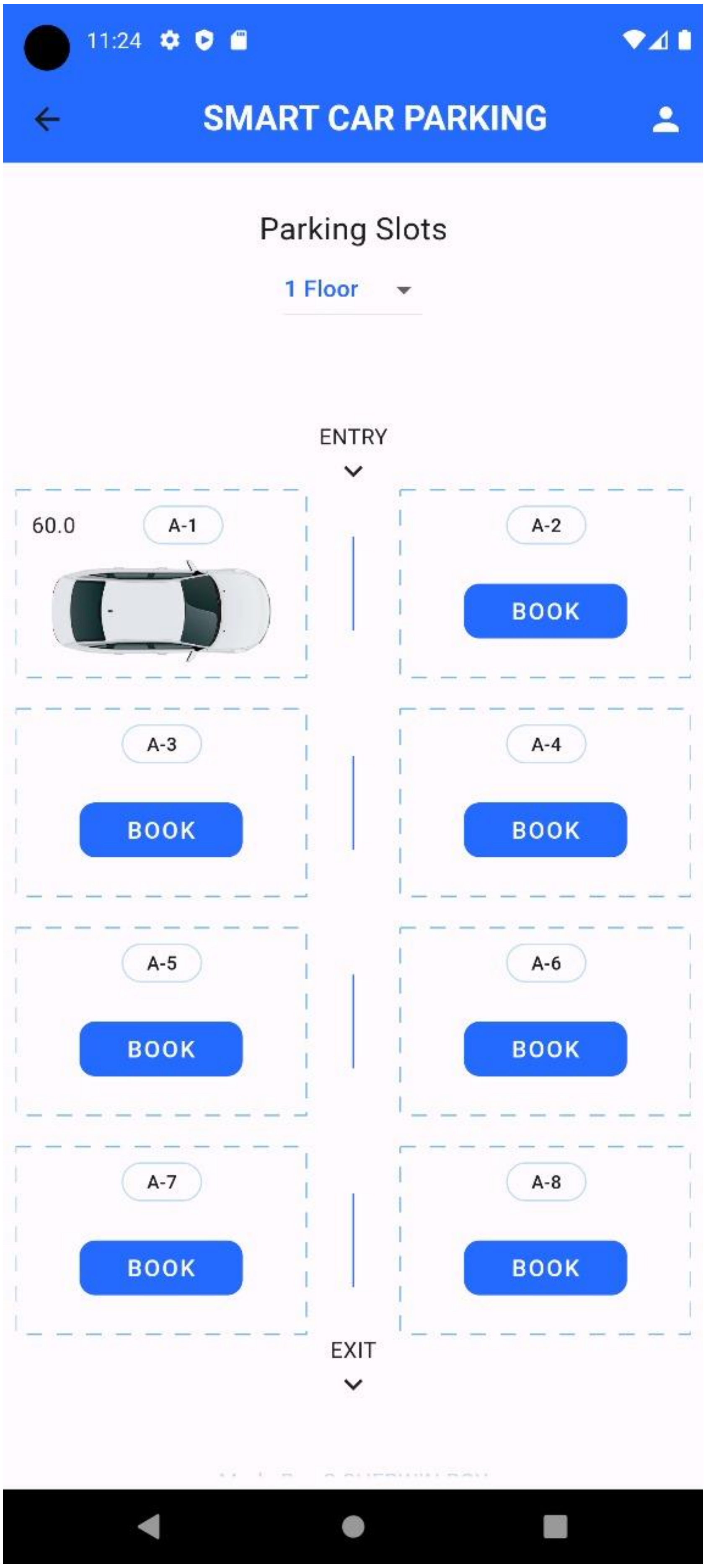
Parking time : 40.0

Parking Slot : A-1

₹ 60

Amount to Be Pay

₹ 60


11:24
SMART CAR PARKING
Parking Slots
1 Floor
ENTRY
60.0
A-1
A-2
BOOK
A-3
BOOK
A-4
BOOK
A-5
BOOK
A-6
BOOK
A-7
BOOK
A-8
BOOK
EXIT

7. SYSTEM TESTING

7.1. Functional Testing:

Functional testing is an essential part of software testing that focuses on verifying the functionality of a system. In the context of the Smart Car Parking Application for users, functional testing ensures that all the intended features and functionalities of the system are working correctly. It involves validating components like user login, data submission, slot approval process, payment fee calculation, and parking status tracking.

During functional testing, various test cases are designed to cover normal and exceptional scenarios. This includes verifying that users can log in successfully, select parking slots accurately, receive appropriate pop-ups on confirmation, and that the system correctly calculates payment. By conducting comprehensive functional testing, any issues or defects in the system's functionality can be identified and addressed early in the development process. This helps in building a robust and user- friendly system that meets the organization's requirements and provides employees with a seamless experience. Functional testing ensures that the system performs its intended tasks accurately and reliably.

7.2. Integration Testing:

Integration testing was conducted to validate the seamless integration and interaction between different components, modules, or subsystems of the system. It aimed to identify any interface or communication issues that may arise when multiple parts of the system were combined. Integration testing in the project involved testing the interaction between the frontend and the backend components, ensuring they work together correctly.

7.3.Performance Testing:

Performance testing was carried out to assess the system's performance and behavior under various load conditions. It helped identify bottlenecks, scalability issues, or resource limitations that affected the system's response time, throughput, or stability. Performance testing in the project involved simulating multiple users accessing the system simultaneously, measuring response times, monitoring resource utilization, and optimizing database queries to ensure efficient performance.

7.4.Usability Testing:

Usability testing focused on evaluating the system's user-friendliness, intuitiveness, and ease of use. It aimed to identify any usability issues, navigation difficulties, or inconsistencies in the user interface.

8. CONCLUSION

The development of a smart parking management application presents a transformative solution to the inefficiencies and frustrations associated with traditional parking systems. By leveraging modern technologies such as IoT sensors and integrating them with a mobile application developed using Flutter and Dart, this project aims to significantly enhance the parking experience for drivers in urban areas.

The proposed application will enable users to book parking slots in advance and make payments online, eliminating the need to physically enter the parking area before securing a spot. This feature addresses key issues such as time consumption, inefficient booking systems, payment inconvenience, and traffic congestion. Real-time updates on parking availability further ensure that users have the most current information, allowing them to make informed decisions quickly and efficiently.

By streamlining the booking and payment processes, the smart parking management application aims to reduce the time drivers spend searching for parking, thereby decreasing traffic congestion and improving overall user satisfaction. The successful implementation of this solution will set a new standard for parking management, offering a more efficient, convenient, and user-friendly experience for drivers.

This project not only addresses the immediate challenges faced by drivers but also contributes to a more organized and sustainable urban environment. The smart parking management application promises to enhance urban mobility, reduce stress for drivers, and optimize the use of available parking spaces, ultimately paving the way for smarter and more efficient urban infrastructure.

Future Scope:

Issues that need to be resolved in the Smart Car Parking Applications:

A, To handle conflicts when two users try to book the same parking slot at the same time in our smart car parking application, we can implement the following strategies:

History: Track the customer's booking history, frequency of use, and reliability.

- Behavior: Monitor how often the customer uses the service, cancellations, and adherence to booked time slots.
- Payment Regularity: Evaluate the customer's payment history, including timeliness and consistency.

Priority Score Calculation:

- Assign a score to each customer based on their profile, behavior, and payment history.
- Develop a scoring algorithm that weighs these factors (e.g., regular payments might have higher weight than booking frequency).

Conflict Resolution Mechanism:

- When two users attempt to book the same slot simultaneously, calculate their priority scores.

- Grant the booking to the customer with the higher priority score.
- Notify the lower-priority customer of the conflict and suggest alternative slots.

Feedback Loop:

- Allow users to provide feedback on the conflict resolution process.
- Continuously refine the scoring algorithm based on user feedback and system performance.

Customer Reward System:

- Implement a reward system to encourage positive behavior and regular payments.
- Offer incentives like discounts or priority booking for high-scoring customers.

B. To handle fines for customers who stay longer than their booked time in our smart car parking application, we can implement a systematic approach that includes notifications, tracking, and automated fine calculations. Here's a detailed plan:

1. Real-time Monitoring and Notifications

- Pre-emptive Alerts: Send notifications to the customer 15-30 minutes before their booking ends, reminding them to vacate the slot or extend their booking.
- Overstay Alerts: Immediately notify the customer if they exceed their booking time, informing them about potential fines.

2. Overstay Detection

Real-time Tracking:

- Use sensors or cameras to monitor the presence of the vehicle in the parking slot.
- Automatically detect when a vehicle stays beyond the booked duration.

3. Fine Calculation and Application

Fine Structure:

Define a fine structure based on the duration of the overstay. For example:

- First 30 minutes: Rs.X per minute
- 31-60 minutes: Rs.Y per minute (higher rate)
- Beyond 60 minutes: Rs.Z per minute (even higher rate)

4. Customer Notification and Payment Processing

Notification:

- Inform the customer about the fine through multiple channels (email, SMS, in-app notification).
- Provide details of the overstay duration and the fine amount.
- Payment Processing:
- Allow customers to pay the fine through the application.

- Integrate with payment gateways for seamless transactions.

5. Repeat Offender Handling

Escalation Mechanism:

- Track repeat offenders who consistently overstay.
- Implement escalating fines or penalties for repeat offenses.
- Consider restricting booking privileges for habitual offenders.

C. Handling a huge database with hundreds of users in a smart car parking application requires a scalable, efficient, and well-architected approach. Here are several strategies to ensure our system can manage a large number of users effectively:

- Caching:

Implement caching for frequently accessed data to reduce the load on the database.

- Archiving:

We have to regularly archive old data to reduce the size of active tables and store archived data in cheaper storage solutions like S3.

- Regular Maintenance:

We can schedule regular maintenance tasks such as vacuuming, rebuilding indexes, and checking for data integrity.

- Replication:

Set up replication to ensure high availability. We can use master-slave or master-master configurations to ensure data is available even if one server fails.

- Backups:

Implement regular backup strategies. We can store backups in multiple locations to ensure data recovery in case of a disaster.

Learnings:

This project not only addresses the immediate challenges faced by drivers but also contributes to a more organized and sustainable urban environment. The smart parking management application promises to enhance urban mobility, reduce stress for drivers, and optimize the use of available parking spaces, ultimately paving the way for smarter and more efficient urban infrastructure.

- User-Centric Design: Prioritizing user needs and preferences in the design process leads to more intuitive and user-friendly applications. Optimizing the use of parking spaces can help reduce congestion and improve the overall flow of traffic in urban areas.
- Environmental Impact: By reducing the time spent searching for parking, we can potentially decrease fuel consumption and emissions from vehicles. Improving parking efficiency can enhance the quality of life for residents by reducing noise and pollution associated with traffic congestion.

- Collaboration and Communication: Effective teamwork and communication are essential for the successful implementation of complex projects involving multiple stakeholders. Being able to adapt to changing circumstances and requirements is crucial for the success of any project.
- Problem-Solving Skills: Identifying and addressing challenges in the project development process helps build problem-solving skills that are valuable in various contexts. Planning, organizing, and executing a project from start to finish provides valuable experience in project management principles. Meeting customer needs and expectations is key to the success of any product or service.

Overall, the project has provided valuable insights into urban planning, community impact, and the importance of user-centered design in creating solutions that address real-world challenges.

9. BIBLIOGRAPHY

- Arduino: https://www.arduino.cc/reference/en/
- https://www.hackerearth.com/blog/developers/arduino-programming-for-beginners/
- Flutter : https://www.tutorialspoint.com/flutter/flutter_tutorial.pdf
- https://www.geeksforgeeks.org/flutter-tutorial/
- https://wiki.eprolabs.com/index.php?title=IR_Obstacle_Sensor#:~:text=6%20References-,Introduction,lets%20user%20adjust%20detection%20rangeFlutter
- https://www.youtube.com/watch?v=VPvVD8t02U8
- https://www.w3schools.com/MySQL/default.asp
- https://www.mysqltutorial.org/
- Nodemcu: https://robocraze.com/blogs/post/what-is-nodemcu-esp8266
- Book : NodeMCU ESP8266 Communication Methods and Protocols : Programming with Arduino IDE Kindle Edition
- by Manoj R. Thakur (Author) Format: Kindle Edition
- MySQL : Querying MySQL: Make your MySQL database analytics accessible with SQL operations, data extraction, and custom queries Paperback – 29 July 2022 by Adam Aspin (Author)
- Beginning App Development with Flutter: Create Cross-Platform Mobile Apps 1st ed. Edition by Rap Payne (Author)
- Beginning Flutter: A Hands-On Guide to App Development by Marco L. Napoli
- For IoT “The Internet of Things” by Samuel Greengard