2023

Smart Parking System

[Rakenny]

Team Members:

- Mario Magdy Momtaz
- Ziad Ahmed Mohamed
- Martin Hany Zakaria
- Hozaifa Magdy Ali
- Ahmed Ashraf Ahmed

Project Supervisor:

DR. Fatma Harby

Future Academy - Higher Future Institute for Specialized Technological Studies
School of Computer Science

Project report submitted to the Future Academy
for the degree of Bachelor of Computer Science

May 2023

[Rakenny]

	**Mario Magdy Momtaz**
	**Ziad Ahmed Mohamed**
	**Martin Hany zakaria**
	**Hozaifa Magdy Ali**
	**Ahmed Ashraf Ahmed**

رؤيــة المعهــد

أن يكــون معهــد المســتقبل العــالي للدراســات التكنولوجيــة المتخصصــة مؤسســة علميــة رائــدة ومتميــزة فــي أوســاط التعلــيم الجــامعي الخــاص وعنصــراً فــاعلاً وداعمــاً لخدمة المجتمع وتنمية البيئة .

رسالـة المعهــد

إعـداد كـوادر علميـة متخصصـة ومتميـزة فـي مجـالات العلـوم الإداريـة والمعلوماتيـة وعلـوم الحاسـب وقـادرة علـي المنافسـة فـي سـوق العمـل مـن خـلال أعضـاء هيئـة تـدريس حاصـلين علـي درجـات علميـة مـن جامعـات مرموقـة ، وتـوفير بيئـة تعليميـة متطـورة ومتميـزة ، وتنميـة مهـارات الطـلاب العلميـة والعمليـة بمـا يجعلهـم قـادرين علي الابتكار والتعلم الذاتي والعمل الجماعي .

Acknowledgement

Praise be to Allah this project was successfully completed.

We should like to express our gratitude for everyone who helped us during the graduation project starting with endless thanks for our supervisor DR. Fatma who didn't keep any effort in encouraging us to do a great job, providing our group with valuable information and advices to be better each time. We have been extremely lucky to have a supervisor who cared so much about our work, Thanks for continuous support and kind communication which had a great effect regarding to feel interesting about what we are working on.

We should also like to thank all the members of staff at Higher Future Institute for Specialized Technological Studies who helped us.

Abstract

This is a project report on mobile application " **Rakenny** " based on smart parking system", during the developing of this project we explored new ideas and functionalities.

This project is the output of our planning, schedule, programming skills and the hard work, and this report reflects our steps taken at various levels of programming skills, planning and schedule.

Efficient and smart way to automate the management of the parking system that allocates an efficient parking space using internet of things technology. The IoT provides a wireless access to the system and the user can keep a track of the availability of the parking area. The aim of this paper is to resolve this issue. The user usually wastes his time and efforts in search of the availability of the free space in a specified parking area.

Table of Contents

Table of Figures

Chapter 1

Introduction and Background

Main points

- A brief introduction of what Smart parking estate is.
- The main problems of the real estate parking and how to develop new solution.
- Our solution to parking problems and how they worked.

1.1 Introduction

Nowadays every park, malls, hospitals, multi-stored buildings have parking spaces. But due to the rapid growth in population, the usage of vehicles are also increasing. Thereby a better solution is needed to park the vehicles because the existing systems are time consuming and also less secure. Finally, users have to spend a lot of time for payment processing and there are chances for people to leave the parking space without doing the payment. The Internet of Things or IoT is a system of interrelated devices connected to internet to transfer and receive data from one another. IoT has a great influence in our lifestyles from the way we react to the way we behave. Smart home is the best example of IoT technology.

1.2 Problem definition

By using ultrasonic sensors be able to keep a record of the number of cars parked inside of a parking garage. Consequently, once a car enters a parking garage followed by a parking space, a ping ultrasonic sensor will then be able to determine if a car is parked in the space or not.

1.3 What is the importance of this problem?

will lead to traffic jam and lost a lot of time. According to an IoT analysis report, the market expenditure of smart parking product and service will increase at a compound annual growth rate of 14% by 2023 and surpass 3.8 billion dollars. The demand for parking is ever-growing, and with cars becoming more fuel-efficient, the number of vehicles will continue to increase. As a result, there are fewer available parking spaces.

1.4 What are the current solutions?

- Increase On-Street (Curb) Parking.
- Subsidize Off-street Parking
- Redesign Existing Parking Facilities These systems help drivers to find an available spot faster and more easily than traditional methods like circling around or waiting for someone to leave. By leveraging this technology, drivers can avoid wasting time looking for a space.

1.5 How will our solution solve the problem? What is new?

The smart parking system also allows drivers to use the space remotely via their smartphones, which is convenient if drivers prefer not to walk to find a space or avoid potential traffic congestion Our solution will help to know available places in the parking around me, and to know if it possible for me to park my car there based on my car's dimensions and the dimensions of the available place in the parking, it will solve traffic jam problem, solve losing time problem, and parking problem all of that based on IOT.

Figure 1.1: Smart parking example

Project Description

Rakkeny mobile application. It is an application that facilitates the parking process for drivers. Through this application, they can know the available parking spaces remotely. So far, the mobile app has been completed, and the parking lot mock-up is complete

1.6 Scope:

The system provides details of the vacant parking slots in the vicinity and reduces the traffic issues due to illegal parking in the vicinity. It is designed with an objective to meet the requirements of controlled parking that offers effortless parking tactics to the authorities.

Deliverables

IoT-based smart parking system transmits available and occupied parking spaces via a mobile application. Each parking space has an IoT gadget, which includes sensors and microcontrollers. The user gets real-time updates on the availability of all parking spaces and, therefore, an option to choose the best one.

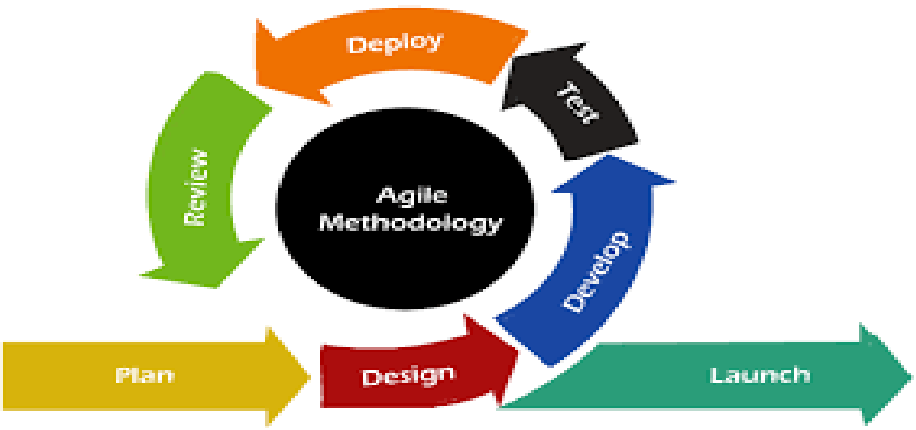


Figure 1.2: software development life cycle

1.7 Summary

In this chapter we provided an overview of how the system should work, how our system is important for the environment and for the real world, and how we solve current problems.

Chapter 2

Analysis and Design

Main points

- What the user and system should do?
- How are the stack holders of this project?
- How the system is designed in system design?
- What is the technologies used to develop that project?

2.1 Introduction

In this chapter we will discuss the software engineering of parking system application to show you how parking system application is done and what it actually provides and how our services are provided to the users.

2.2 User and System Requirements

User requirements are the needs and expectations of the people who will be using the parking system application. The user requirements should be gathered through a variety of methods such as surveys, focus groups, and interviews. Some common user requirements for a parking system application include:

1. Easy and intuitive user interface
2. Real-time availability of parking spaces
3. Clear and timely communication of parking rules and regulations

System Requirements:

System requirements are the technical specifications and constraints that must be considered when developing the parking system application. These requirements should be defined by the development team and should take into account the needs of the users. Some common system requirements for a parking system application include:

1. Scalability and flexibility to accommodate a large number of users and parking spaces
2. Security features to protect user data and prevent fraud
3. Integration with third-party services such as geolocation services

4. Reliable and efficient data storage and retrieval
5. Compatibility with various operating systems and web browsers
6. Robust testing and error handling mechanisms to ensure system stability and reliability.
7. By considering both the user and system requirements, the parking system application can be designed and developed to meet the needs of the users while also ensuring that the technical constraints are met.

2.2.1 Functional requirements

1. Admin must be able to Define new parking areas, specify a range of parking lots, the parking cost per minute/hour, and other details.
2. Admin must be able to update data of existing parking areas.
3. Admin must be able to view the information of all registered parking areas.
4. The parking operator must be able to Send the vehicle plate number and reservation password (Session ID) to a central server for verification once users check in.
5. User must be able to Register for the service and enter personal and vehicle details.
6. User must be able to find a parking area from the list of areas, registered by parking admins.

2.2.2 Non – functional requirements

1. The whole system is expected to be delivered in four months of time with weekly evaluation by the project guide.
2. The system is designed for user friendly environment and ease of use.
3. The system should provide a reliable environment to both operator and admin.
4. All orders should be reaching at the admin without any errors.

2.3 Stack holders

Stakeholders are individuals or groups who have a vested interest in the success of the parking system application. Some common stakeholders for a parking system application include:

Parking lot operators and managers: These stakeholders are responsible for the day-to-day operations of the parking lot and need a system that can help them manage parking spaces, issue permits, monitor occupancy, and enforce parking regulations.

Property managers: Property managers may be responsible for managing multiple properties that have their own parking lots or garages. They need a parking system application that can provide centralized management and reporting across all properties.

Parking enforcement officials: These stakeholders need a system that can help them monitor parking violations and issue citations.

Users: Users of the parking system application include drivers who need to find parking.

Developers: Developers are responsible for designing, developing, testing, and deploying the parking system application.

Third-party service providers: The parking system application may need to integrate with third-party service providers such as payment processors, geolocation services, and mapping services.

By identifying and understanding the needs and interests of each stakeholder, the parking system application can be designed and developed to meet the needs of all parties involved. This can help ensure the success and adoption of the application by all stakeholders.

2.4.1 Block Diagram

A typical block diagram for a smart parking IoT system might include the following components:

- Parking sensors: These are typically installed in parking spaces to detect the presence or absence of a vehicle. They can be based on a variety of technologies, such as ultrasonic, infrared, or magnetic sensors.
- Gateway: The sensors are connected to a gateway device, which collects data from the sensors and sends it to the cloud server. The gateway may also provide local processing and intelligence to reduce the amount of data that needs to be sent to the cloud.
- Cloud server: The cloud server receives data from the gateway and performs data processing and analysis. It may use machine learning algorithms and other techniques to predict parking availability, optimize parking utilization, and provide real-time information to drivers.
- Mobile app/web portal: A mobile app or web portal provides drivers with information about parking availability, location, and pricing. It may also allow drivers to reserve a parking space in advance.
- Parking management system: The parking management system provides administrative tools to manage parking operations, including monitoring occupancy, generating reports, and setting pricing policies.
- Overall, the smart parking IoT system aims to improve parking efficiency, reduce traffic congestion, and enhance the overall user experience for drivers.

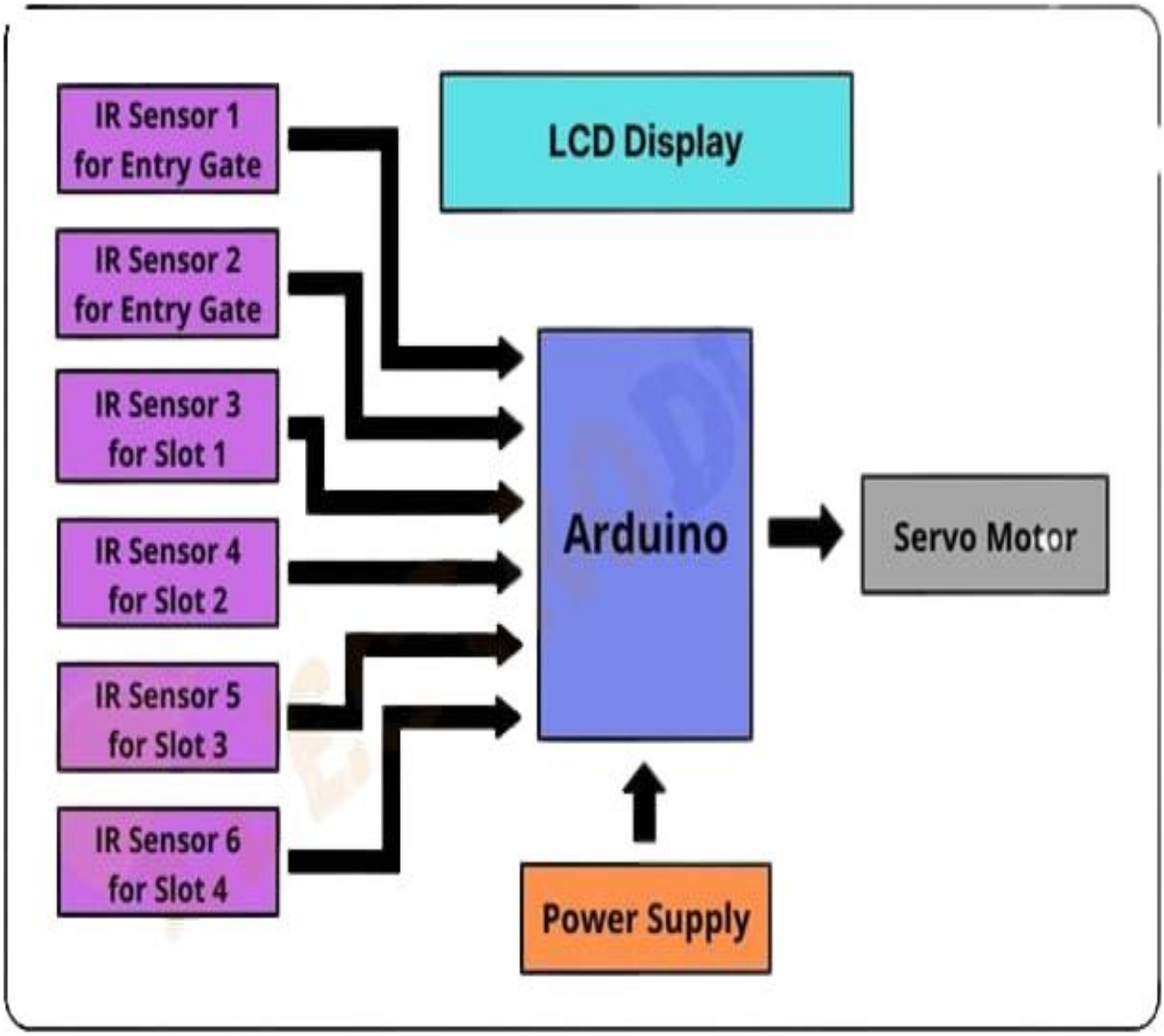


Figure 2.1: Block Diagram

2.4.2 Use Cases

Smart parking IoT systems can offer additional benefits and use cases beyond traditional smart parking systems. Here are some examples:

- Real-time occupancy monitoring: Smart parking IoT systems can provide real-time information on parking occupancy, which can be used to optimize parking utilization and reduce congestion. The data can be used to dynamically adjust parking rates, direct drivers to available parking spaces, and even predict future demand.
- Data analytics and predictive modeling: By collecting and analyzing data from parking sensors and other sources, smart parking IoT systems can provide insights into parking demand and usage patterns. This information can be used to optimize parking operations, pricing, and resource allocation.
- Integration with other IoT devices: Smart parking IoT systems can be integrated with other IoT devices, such as traffic sensors, weather sensors, and air quality sensors, to provide a more complete picture of urban mobility. This data can be used to optimize traffic flow, reduce emissions, and improve overall quality of life for residents.
- Autonomous vehicles: As autonomous vehicles become more common, smart parking IoT systems can provide a seamless parking experience for these vehicles. Parking sensors can communicate with autonomous vehicles to guide them to available spaces, and payment can be automatically deducted from the vehicle's account.
- Energy efficiency: Smart parking IoT systems can be designed to be energy-efficient, using low-power sensors and gateway devices, and optimizing data transmission to minimize energy consumption. This can help reduce the carbon footprint of parking facilities and contribute to overall sustainability goals.
- Overall, smart parking IoT systems offer a range of benefits and use cases, from real-time occupancy monitoring to integration with other IoT devices.

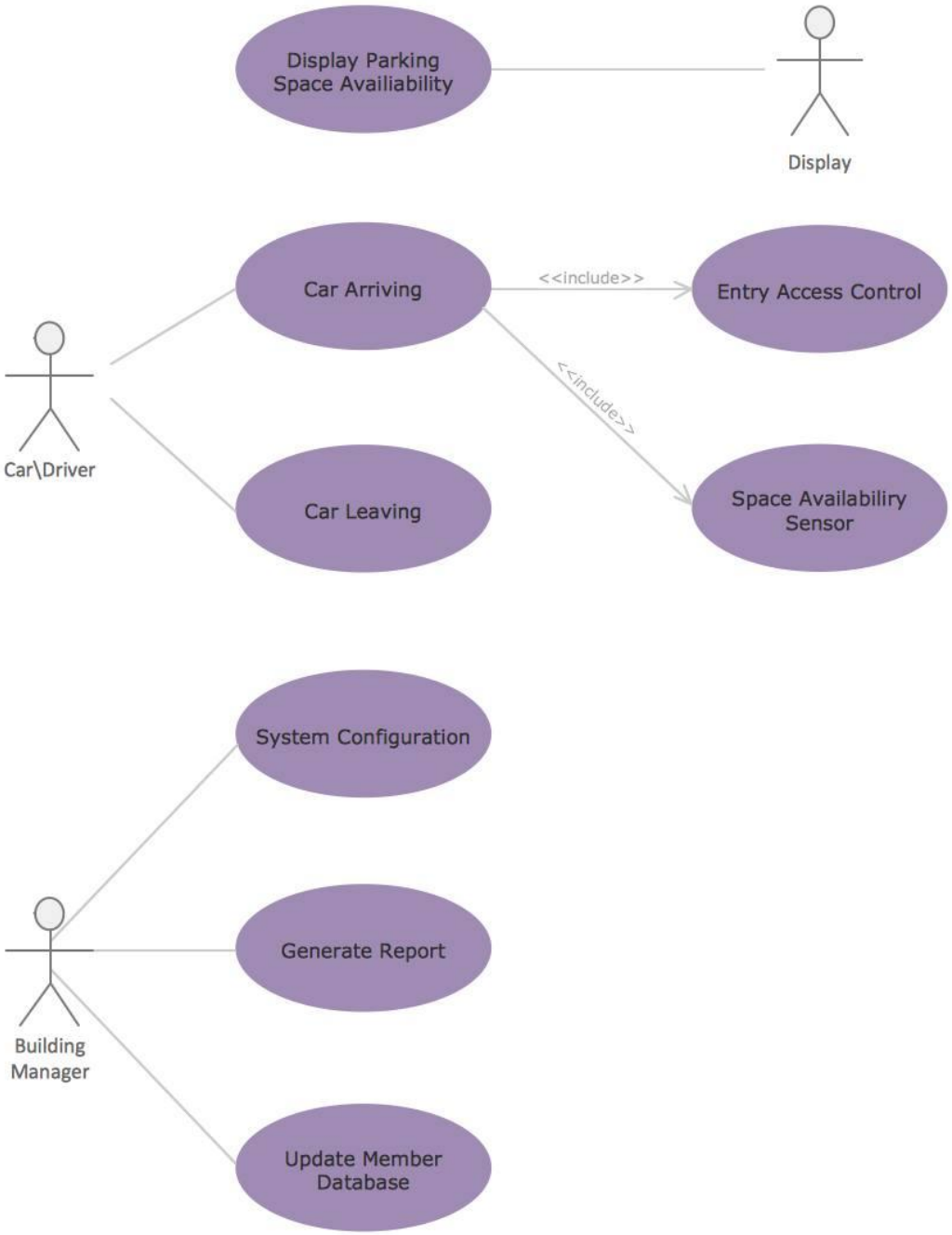


Figure 2.2: Use Cases

2.4.3 Class Diagram

The Parking Lot class represents a physical parking lot, with attributes such as id, name, location, capacity, and available Spaces.

- The Parking Sensor class represents a sensor installed in a parking lot, with attributes such as id, status, and location.
- The Gateway class represents a device that connects multiple sensors and transmits data to the cloud server. It has attributes such as id, location, and sensors.
- The Parking Operator class represents the operator of a parking lot, with attributes such as id, name, email, and phone.
- The Payment System class represents a payment gateway used by the parking system, with attributes such as id, name, description, and type.
- The Mobile App class represents a mobile application used by drivers to access information about parking availability and make reservations. It has attributes such as id, name, version, OS, and parking Lots.

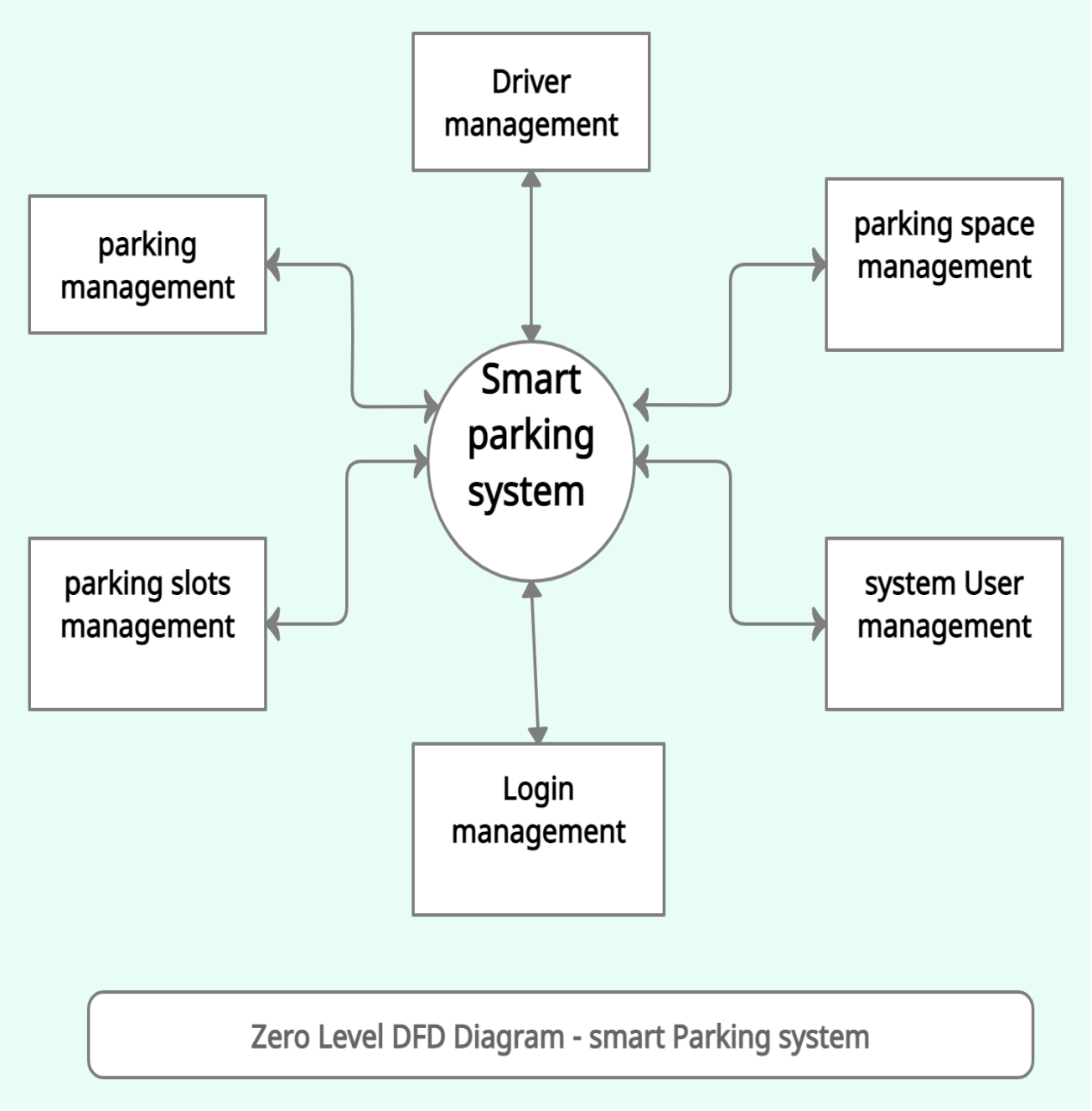


Figure 2.3: Class Diagram

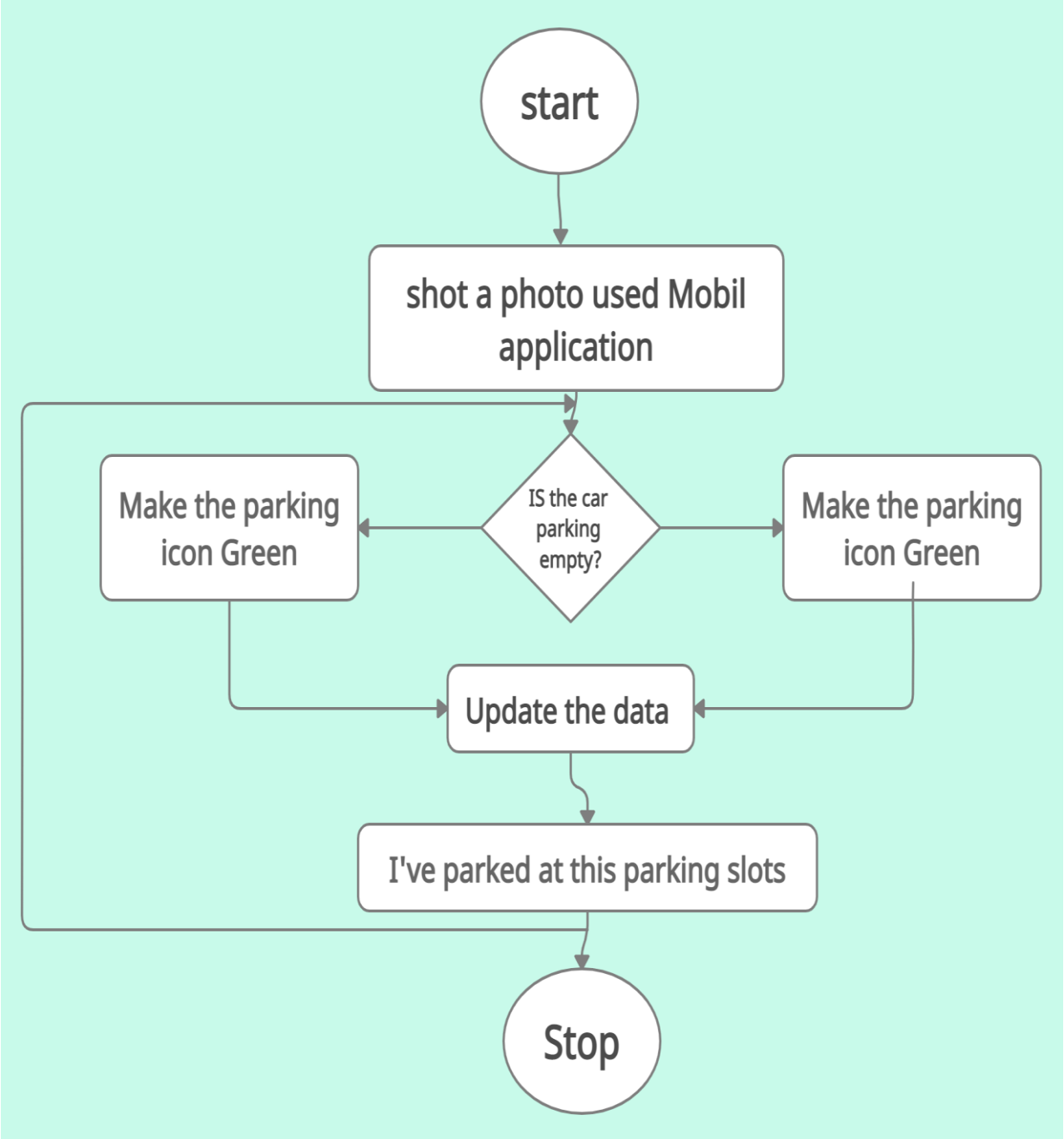


Figure 2.4: Flowchart Diagram

2.4.4 Design Patterns

There are several design patterns that can be used for a smart parking IoT system. Here are some examples:

- Observer Pattern: The Observer pattern can be used to notify the cloud server and mobile app of changes in parking availability. In this pattern, the parking sensors act as the subject, and the cloud server and mobile app act as the observers. When a sensor detects a change in parking availability, it notifies the cloud server, which in turn notifies the mobile app. This pattern can help ensure that users have up-to-date information on parking availability.
- Singleton Pattern: The Singleton pattern can be used to ensure that there is only one instance of certain classes in the system. For example, the payment gateway and parking operator classes could be implemented as singletons to ensure that there is only one instance of each in the system. This can help simplify the code and reduce the risk of errors caused by multiple instances of the same class.
- Strategy Pattern: The Strategy pattern can be used to implement different pricing strategies for parking. For example, the parking operator could define different pricing strategies based on factors such as time of day, day of the week, or parking lot location. Each pricing strategy could be implemented as a separate strategy class, which can be selected at runtime based on the current conditions.
- Decorator Pattern: The Decorator pattern can be used to add additional functionality to parking sensors. For example, a sensor could be decorated with a Bluetooth module that allows it to communicate with nearby smartphones. This can help improve the accuracy of the sensor data and provide additional features such as auto-check-in and check-out.
- Facade Pattern: The Facade pattern can be used to simplify the interface between different components of the system. For example, a facade class could be implemented to provide a simplified interface between the mobile app and the payment gateway. This can help reduce the complexity of the system and improve maintainability.
- Overall, design patterns can help improve the structure and maintainability of a smart parking IoT system, and ensure that it is flexible and adaptable to changing requirements.

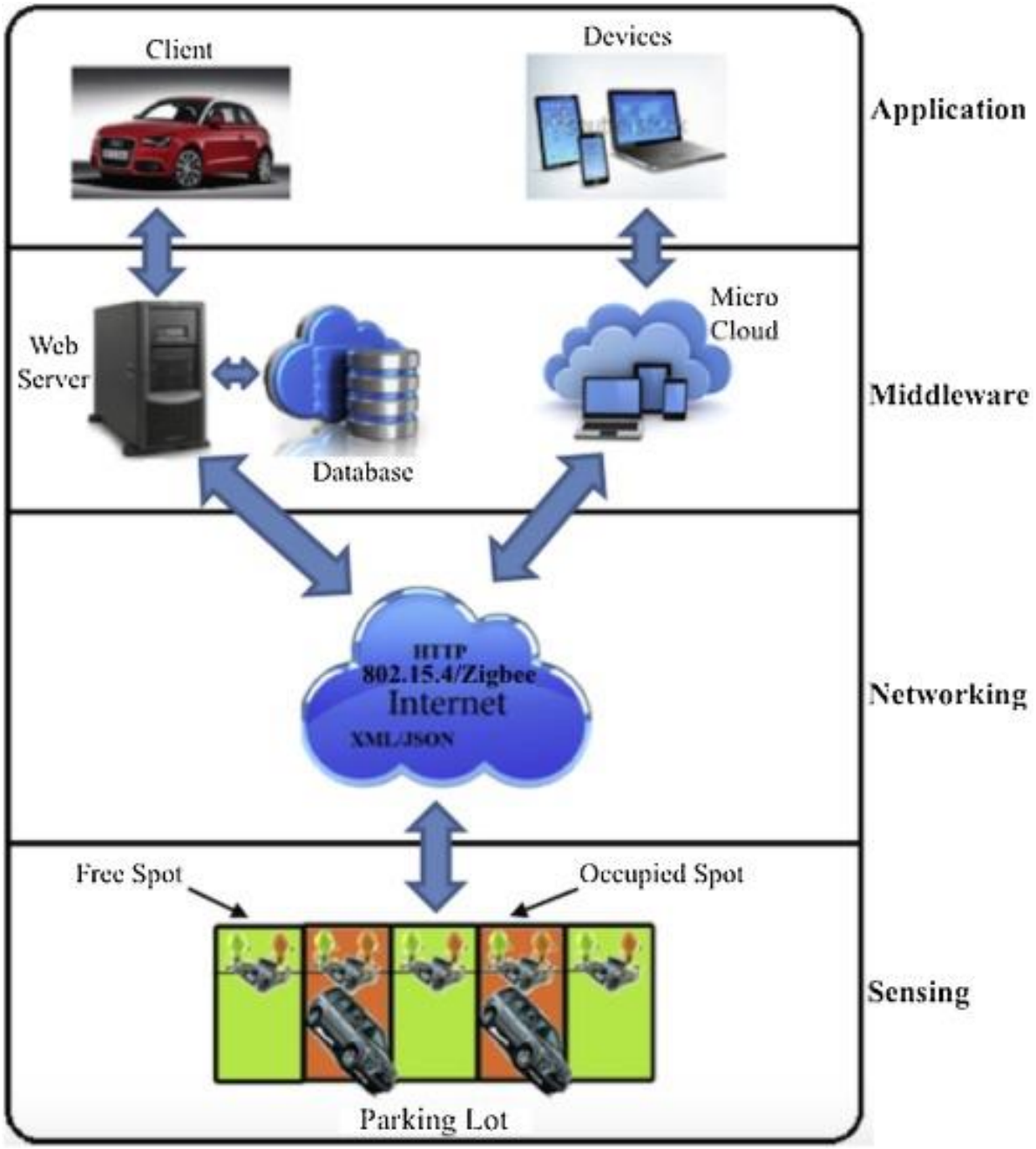


Figure 2.5: Design Patterns

2.4.5 Database Design

A database design for a smart parking IoT system would typically involve several tables to store data related to parking lots, sensors, parking transactions, payment information, and other relevant data. Here is an example of a possible

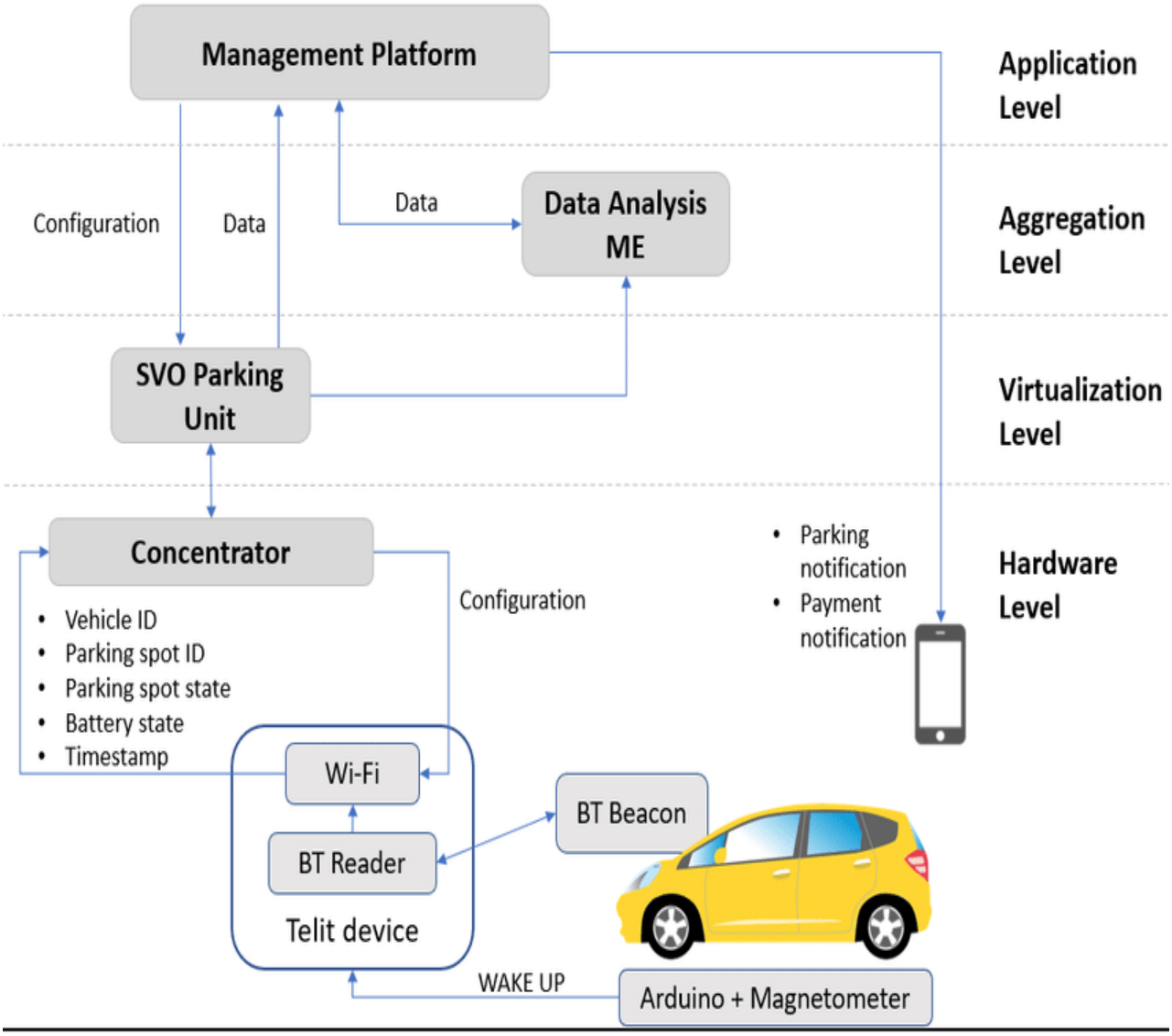


Figure 2.6: Database Design

2.4.6 Sequence diagrams:

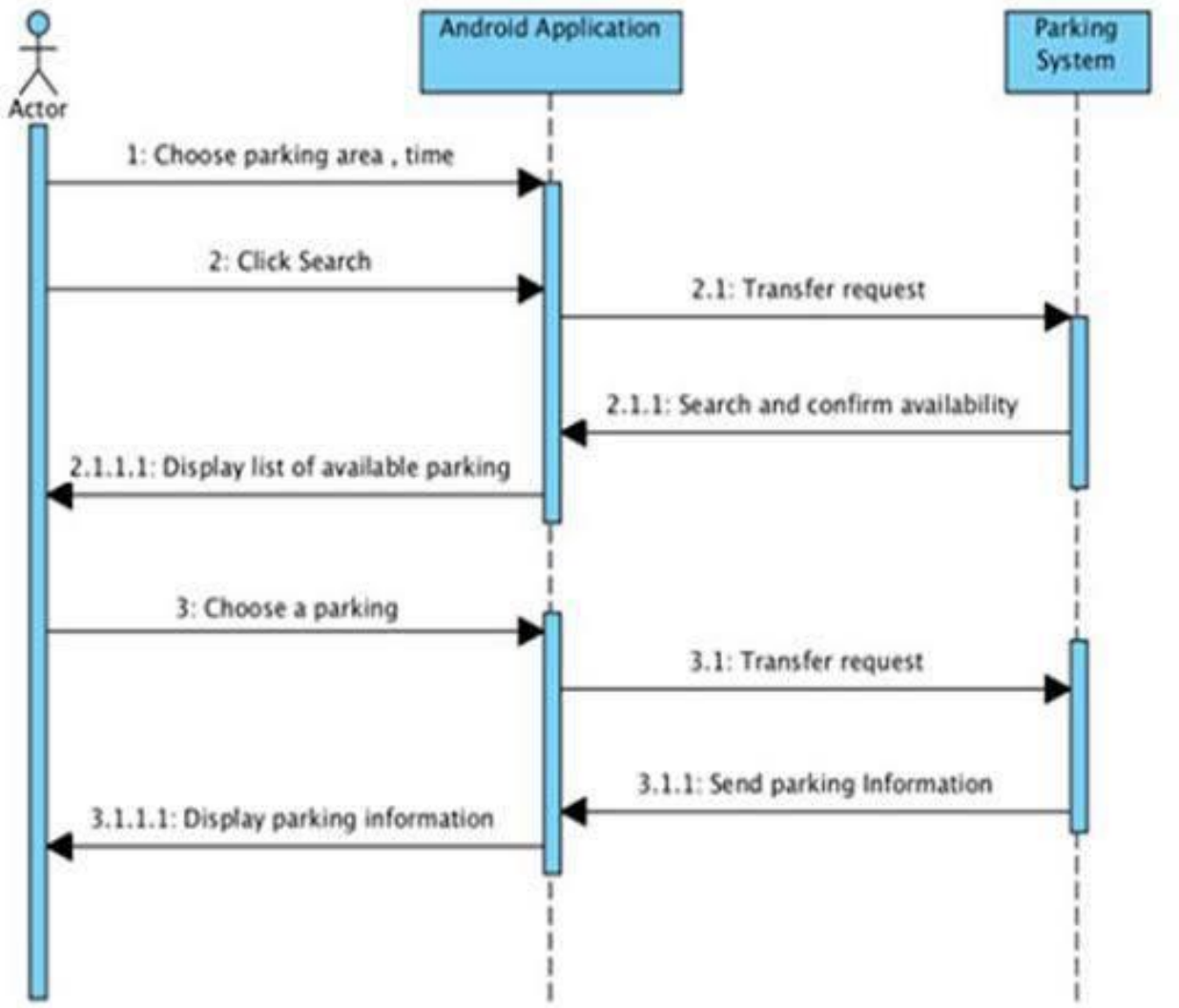


Figure 2.7: Sequence diagram

2.5 Used Technologies and tools

- ### Mobile application

➢ **Flutter**

Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, it allows you to create a native mobile application with only one codebase. This means that you can use one programming language and codebase to create two different apps (for iOS and Android).

Flutter consists of two important parts:

An SDK (Software Development Kit): A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android).

A Framework (UI Library based on widgets): A collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs. To develop with Flutter, you will use a programming language called Dart. The language was created by Google in October 2011, but it has improved a lot over these past years.

Dart focuses on front-end development, and you can use it to create mobile and web applications. If you know a bit of programming, Dart is a type object programming language. You can compare Dart's syntax to JavaScript.

- **Back end**

➢ Firebase

Firebase is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firebase side which makes it easy to use its features more efficiently. It provides services to android, iOS, web, and unity. It provides cloud storage. It uses NoSQL for the database for the storage of data.

➢ **Firebase Real time database**

The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in Realtime.

➢ **Firebase Storage**

Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google

➢ **Flutter launcher icons**

A command-line tool which simplifies the task of updating your Flutter app's launcher icon. Fully flexible, allowing you to choose what platform you wish to update the launcher icon for and if you want, the option to keep your old launcher icon in case you want to revert back sometime in the future.

2.6 Summary

Now you Know what smart parking is providing and how and how provide it now we also know all about our software engineering to build that project and we did talk about:

1.user requirements

2.system requirements

3.functional requirements

4.Non-functional requirements

5.who is stack holders of smart parking

6.what is our use case diagram

7.what is our class diagram

8.what is the design pattern

9.what is our class diagram

Chapter 3

Deliverables and Evaluation

Main points

- Introduction
- User Manual
- Evaluation
- Summary

3.1 Introduction

In this chapter we will introduce three major processes needed in order to deliver and produce a fine product. First, user manual is important in many ways. If you just installed an application or open and do not know how to use it, or how to use a specific feature of the application? Here comes the importance of user manual, it should be well-written, well-organized and well-explained. It is the first thing that users look for it just after they get stuck. Second, software testing is really required to point out the defects and errors that were made during the development phases. It is very important to ensure the quality of the product. Testing is required for an effective performance of software application or product. Third, evaluation and user experiment, it helps developers to fix encountered bugs and errors.

3.2 User Manual

In Mobile application

Splash screen

Figure 3.1: Splash screen

Home screen

Figure 3.2: Home screen

Car screen

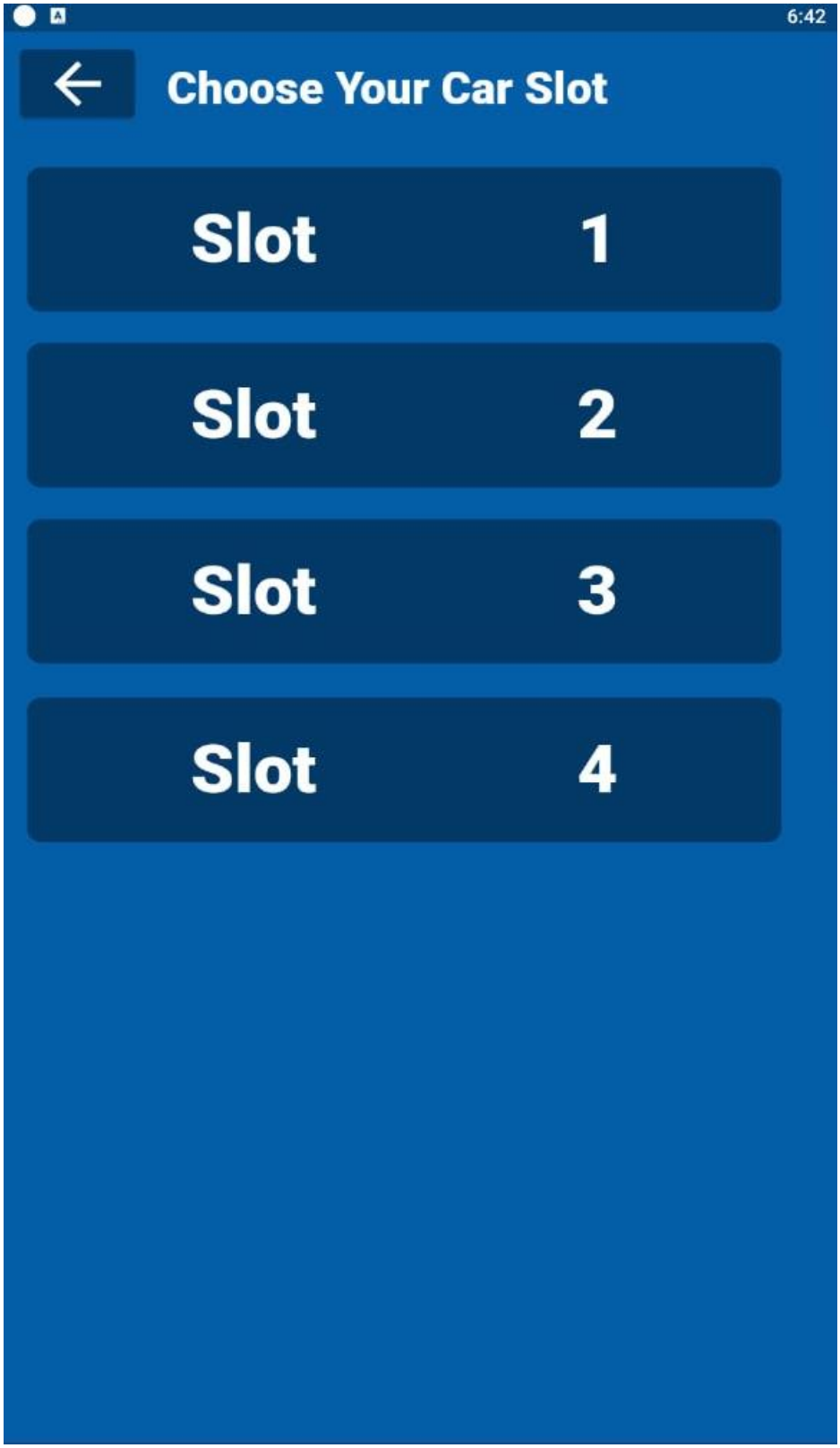


Figure 3.3: Car screen

Scooter screen

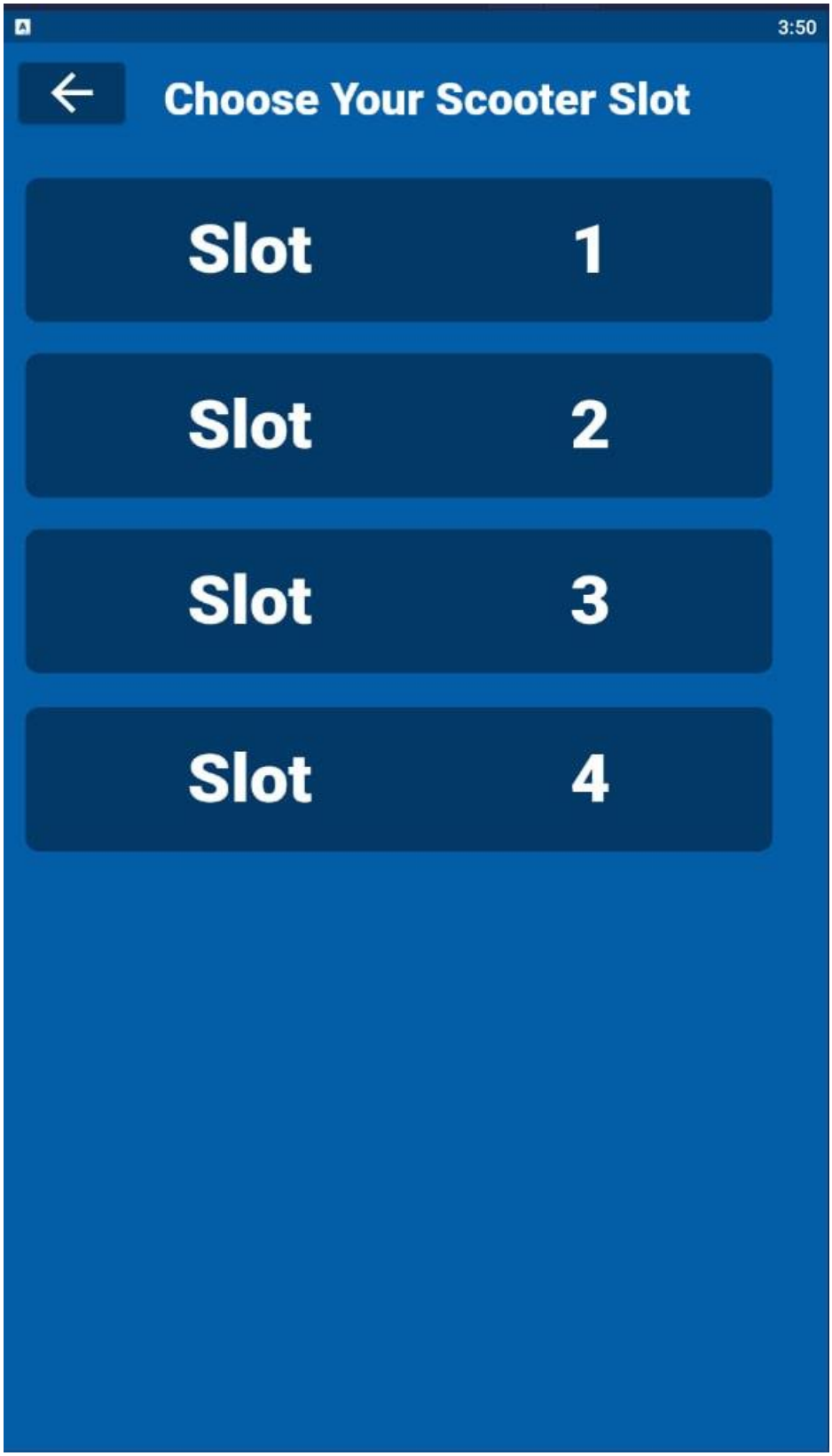


Figure 3.4: Scooter screen

Bicycle screen

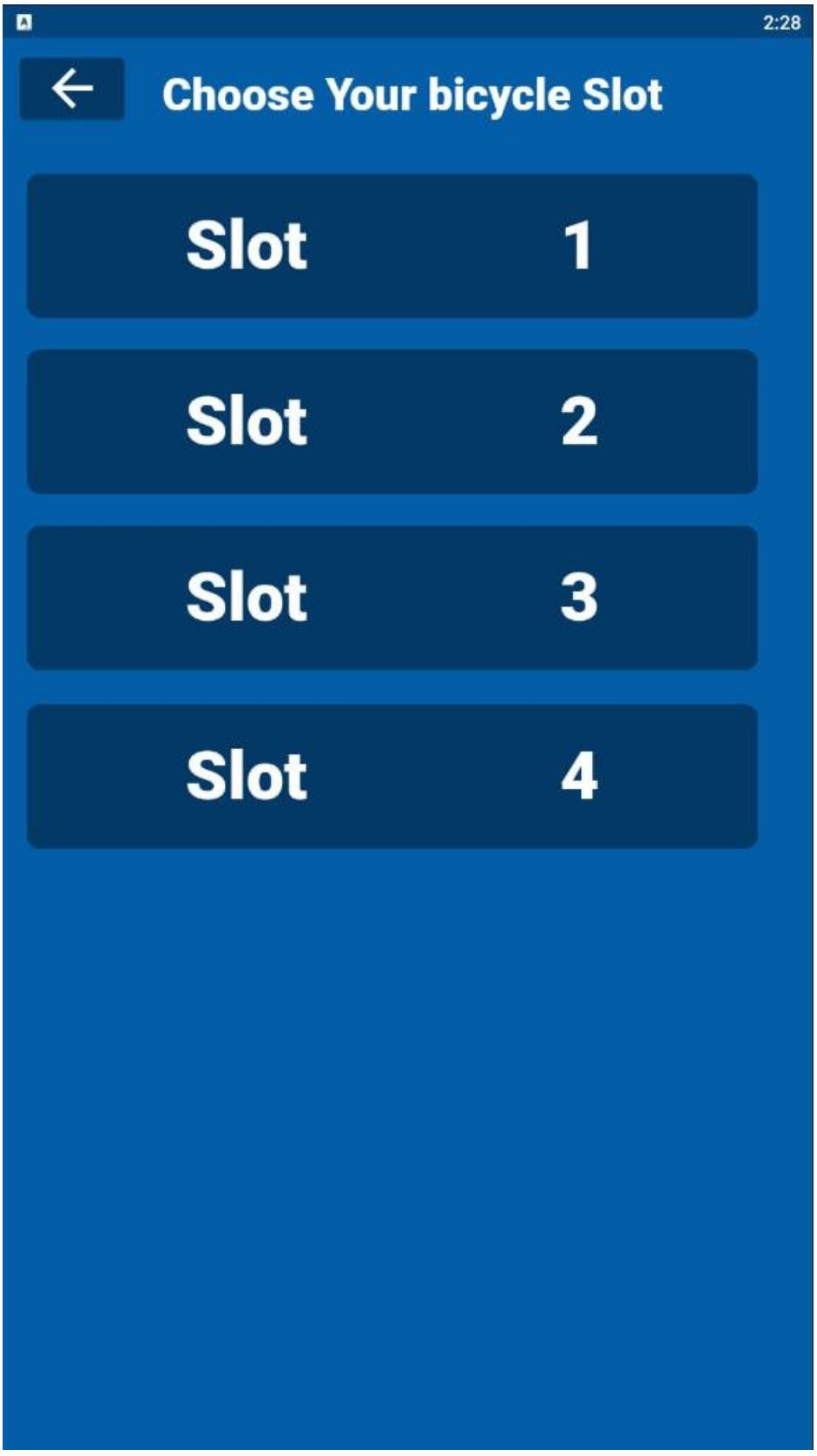


Figure 3.5: Bicycle screen

Available message

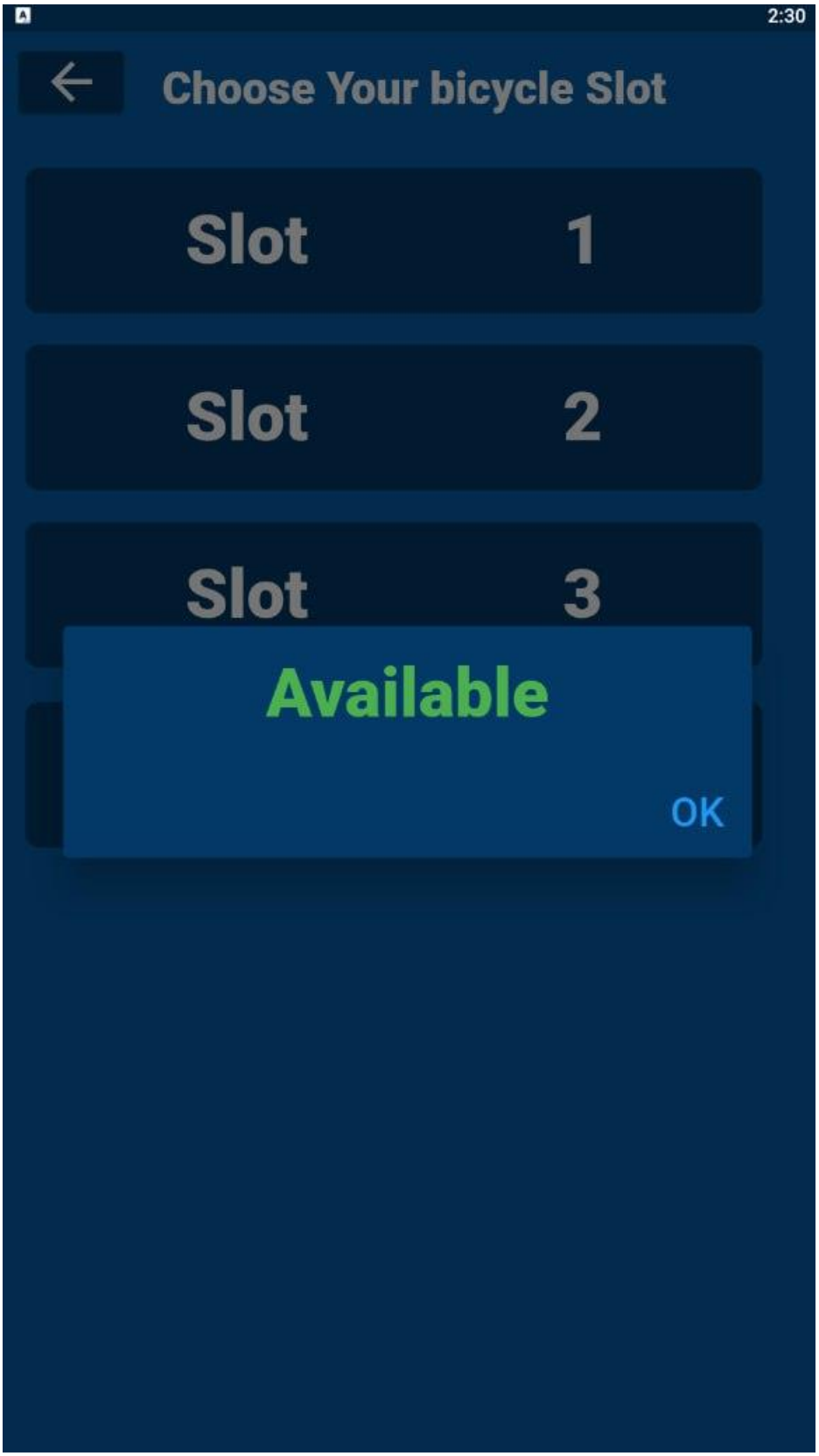


Figure 3.6: Available message

Not available message

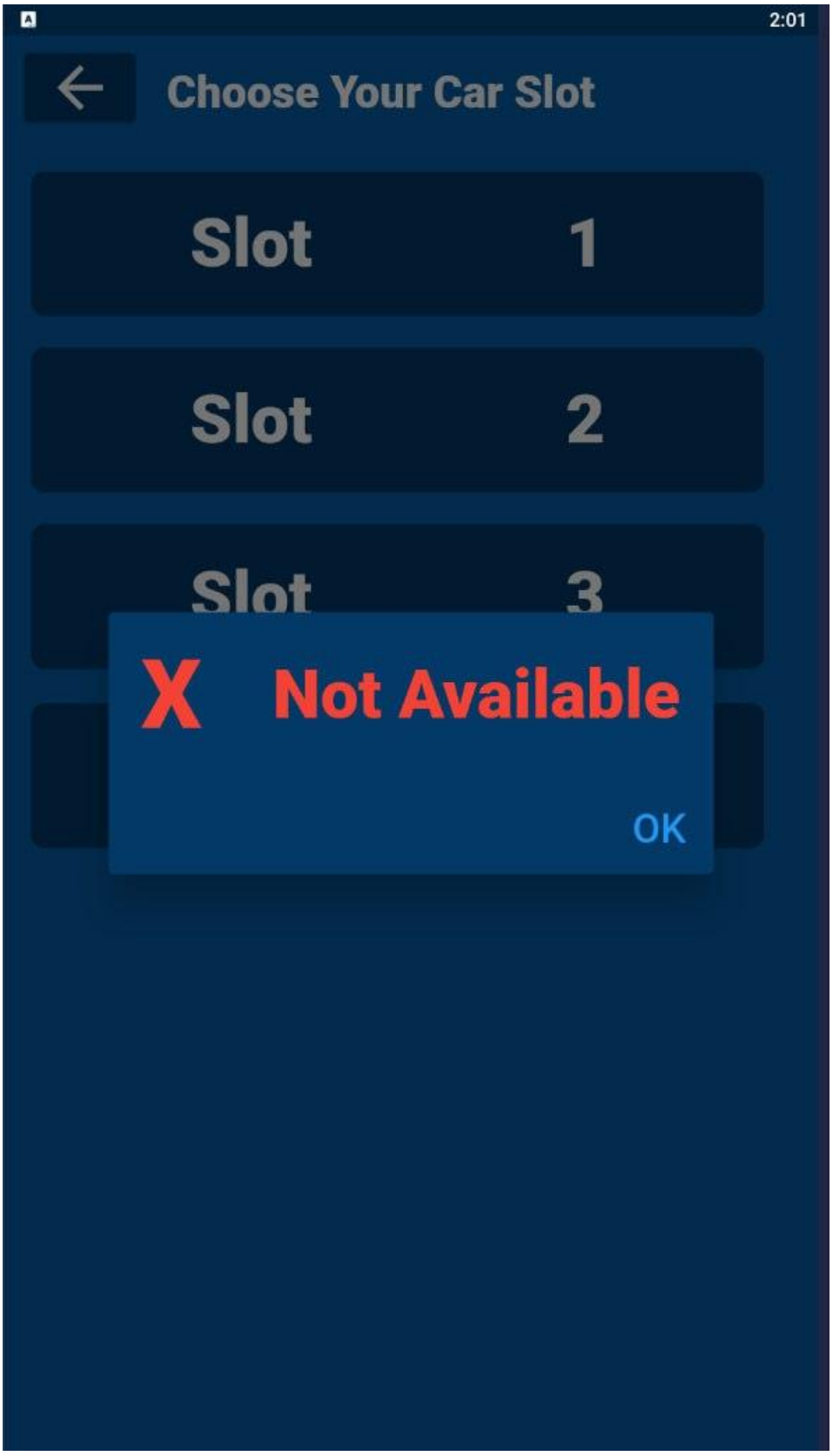


Figure 3.7: Not available message

3.3 Evaluation (User experiment)

Our system is developed under experiment to find Weaknesses and strengths and User opinions interest us.

After evolution, users found the system easy with no renting experience for make it easy for any user.

3.4 Summary

To build a successful system developers need to have their users involved and interacting with them in order to help those finding bugs, improving system functionalities and giving feedback that will help developers to maintain the software. Testing phase is important, it is the phase when you find your software's weakness, flaws and security issues. To maintain a successful community, users' guides and manual should be well written and easy to follow.

Chapter 4

Discussion and Conclusion

Main points

- Main Finding
- Why is this project important?
- Practical Implementations
- Future Recommendation
- Conclusion Summary

4.1 Introduction

With the remarkable, effective and extremely rapid development of technology and its expansion and reach to most of the Earth's population, transforming all paper and traditional services into technological methods has become a necessary and more effective method. Which fits him as soon as he enters a website and a mobile application, provided that it is safe and effective in this is one of the best services in technology at the present time. The purpose of this project is to reduce congestion and not find parking spaces and parking spaces. In our project, we tried as much as possible to solve this matter through the phone application that helps you find a place to park remotely.

4.2 Main Findings

1. A scale model of the car park was made
2. An application has been created that facilitates the process of parking
3. A wireless connection was made remotely via Wi-Fi
4. A miniature part was added using sensors and Arduino

4.3 Why is this project important?

Application **Rakkeny** helps the customer or people in general with the presence of a starting point to park their cars from each other. This makes it easier for them by opening the application and entering the existing parking lot and seeing the empty places and booking them and going to them without wasting a lot of time in searching for an empty place to park

4.4 Practical Implementations

Arduino Uno

Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.
The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

Figure 4.1: arduino uno

ESP8266

The ESP8266 is a low-cost Wi-Fi microchip, with built-in TCP/IP networking software, and microcontroller capability, produced by Espressif Systems.

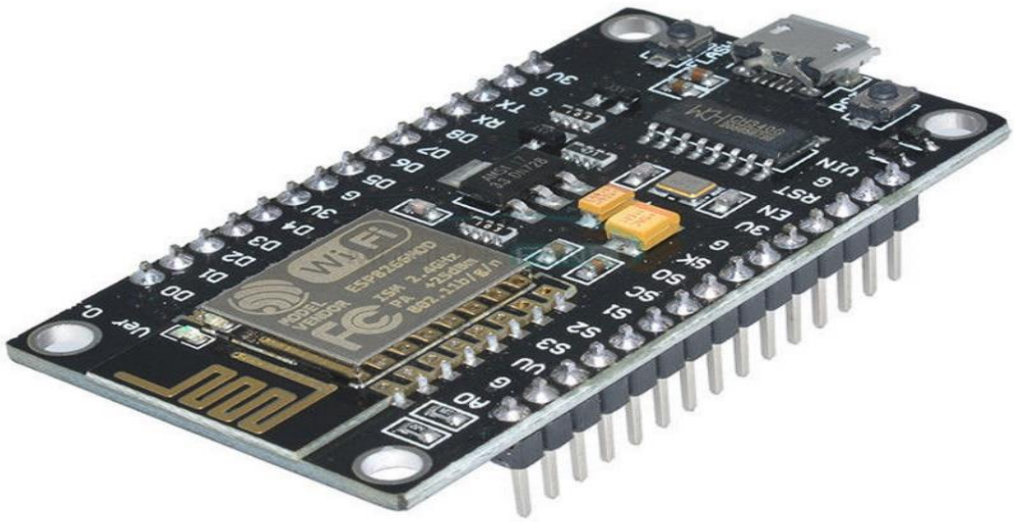


Figure 4.2: ESP8266

IR sensor

The IR sensor is one of the most common sensors in the Arduino world, due to its technology. Depending on infrared technology, we can send messages between controllers or detect objects, as well as detect fires and others.

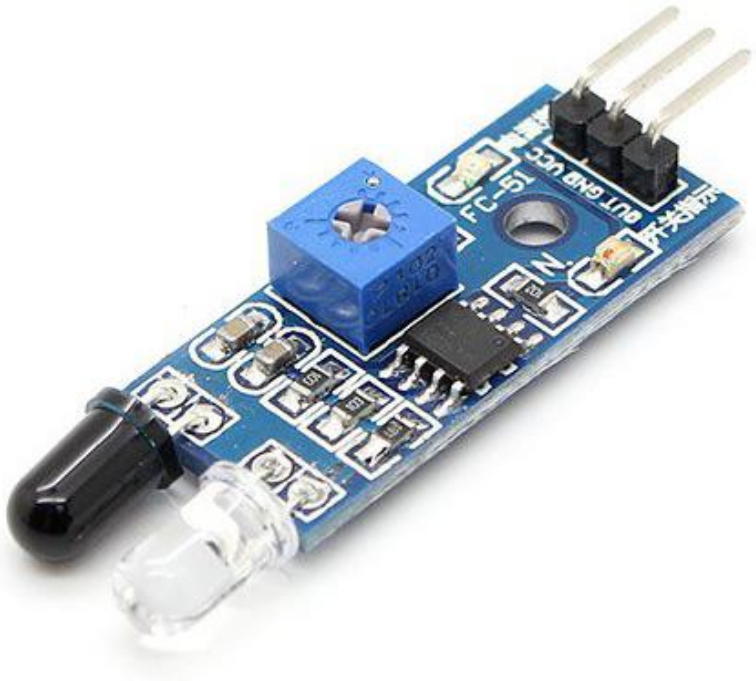


Figure 4.3: IR sensor

Servo motor

Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a three horns (arms) and hardware.

Figure 4.4: Servo

LCD displays

LCD displays like these are very popular and broadly used in many electronics projects because they are great for displaying simple information, like sensors data, while being very affordable.

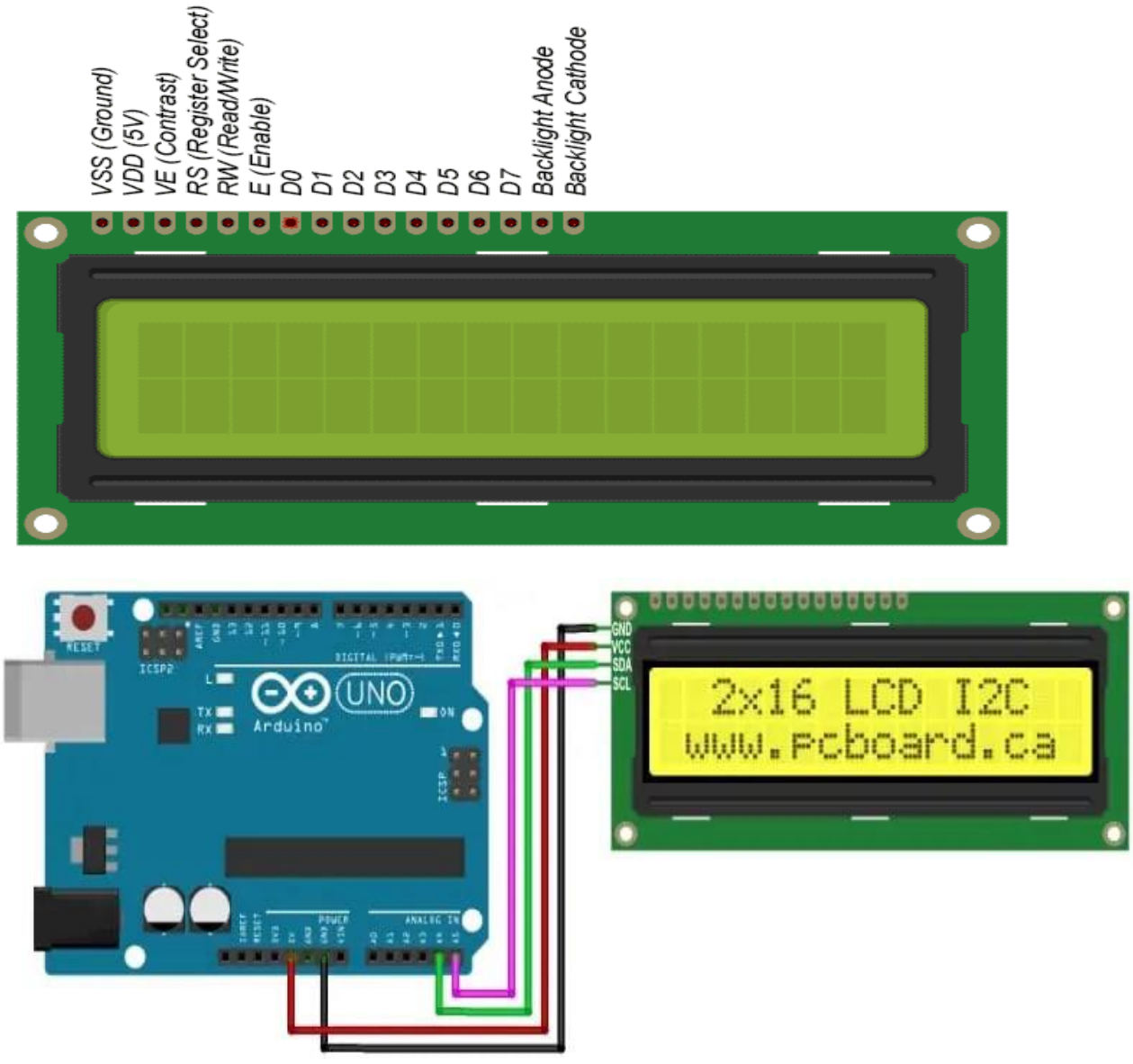


Figure 4.5: LCD display

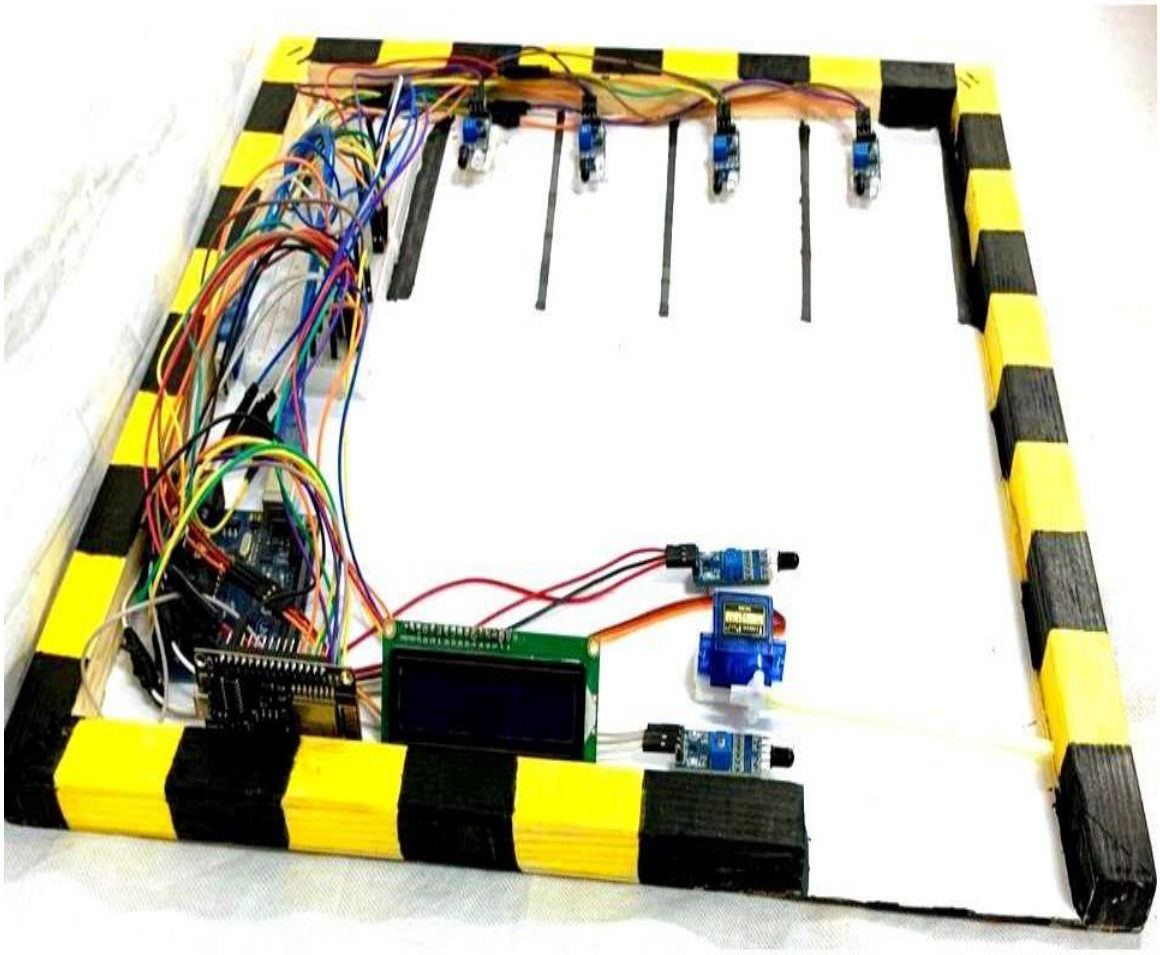


Figure 4.6: Parking model

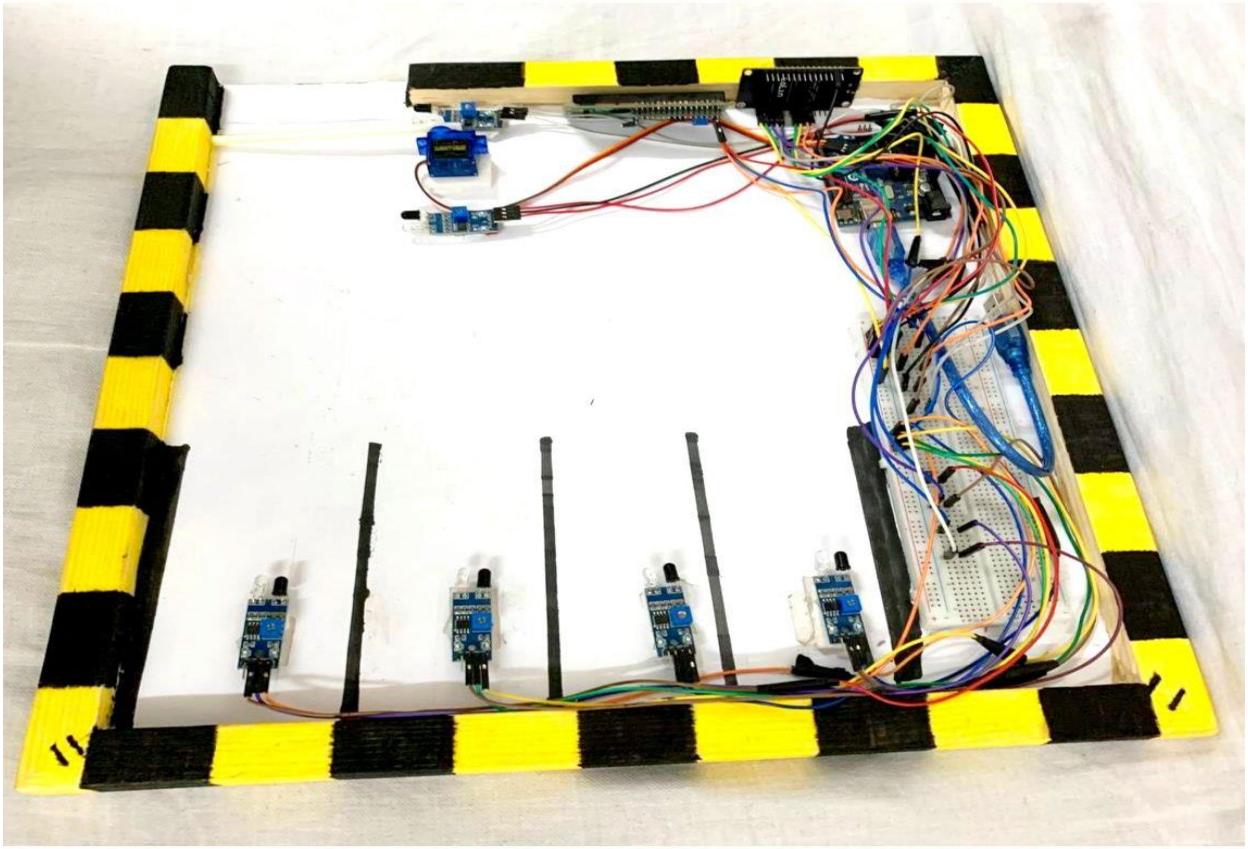


Figure 4.7: Parking model

Figure 4.8: 3D Max model

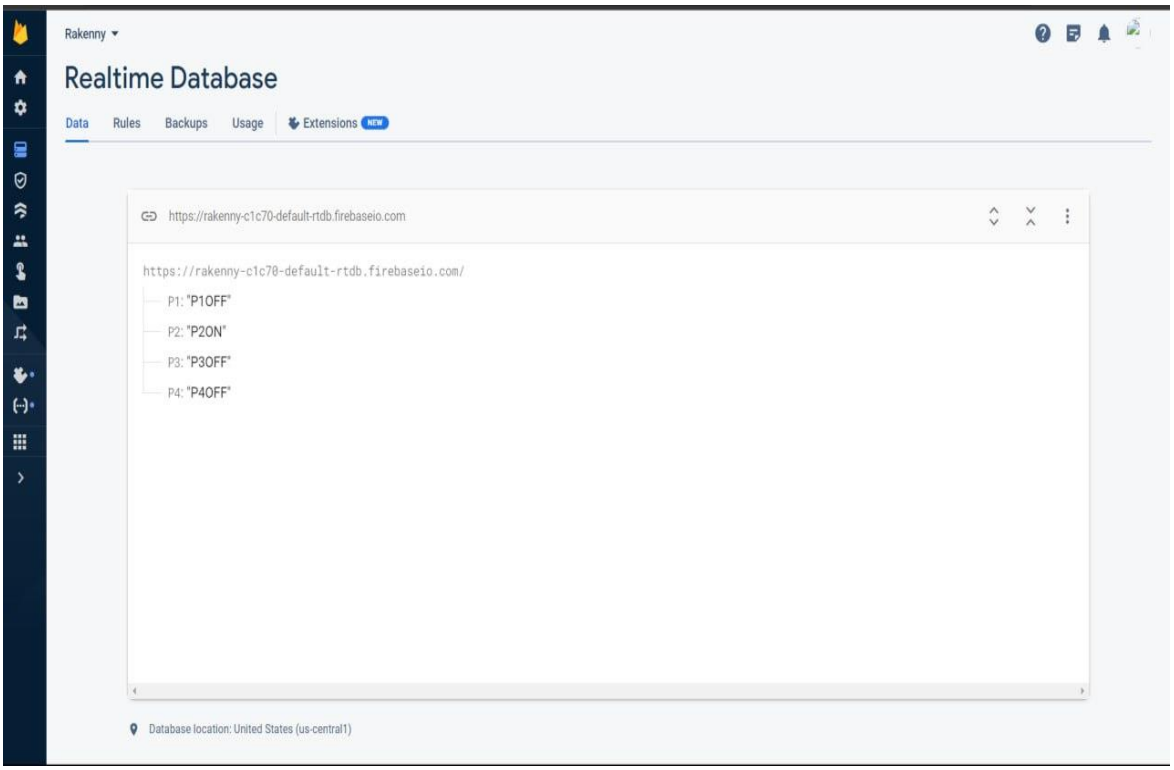
Figure 4.9: 3D Max model

Figure 4.10: 3D Max model

Figure 4.11: 3D Max model

Figure 4.11: 3D Max model

Arduino with firebase (Realtime Database)




```
18  #define FIREBASE_HOST "https://rakenny-c1c70-default-rtdb.firebaseio.com/"
19  #define FIREBASE_AUTH "SWA57tlv3z6pH6PtMyHnjyPfYeuIWt4nzl2oBM4u"
20  FirebaseData firebaseData;
```

```
66    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
```

```
103
104     int P1= digitalRead(SP1);
105     if(P1){
106       Firebase.setString(firebaseData, "/P1", "P1OFF");
107     } else {
108       Firebase.setString(firebaseData, "/P1", "P1ON");
109     }
110
111
112     int P2= digitalRead(SP2);
113     if(P2){
114       Firebase.setString(firebaseData, "/P2", "P2OFF");
115     } else {
116       Firebase.setString(firebaseData, "/P2", "P2ON");
117     }
118
119
120     int P3= digitalRead(SP3);
121     if(P3){
122       Firebase.setString(firebaseData, "/P3", "P3OFF");
123     } else {
124       Firebase.setString(firebaseData, "/P3", "P3ON");
125     }
126
127
128     int P4= digitalRead(SP4);
129     if(P4){
130       Firebase.setString(firebaseData, "/P4", "P4OFF");
131     } else {
132       Firebase.setString(firebaseData, "/P4", "P4ON");
133     }
134
```

4.5 Limitations

- ✓ While implementing the back-end firstly we implemented with Figma next turned it with flutter because we found that doesn't access firebase very well
- ✓ We had difficulty to find a way to update a record on firebase so we change the way as we can get the modified address with its reference.
- ✓ Without internet connection the user cannot access the mobile app.
- ✓ We had difficulty to pass a variable from page to page on the mobile app then we solve it by the dart and flutter

4.6 Conclusion Summary

From the market study and data analysis, we found that the system applied in the parking lot field is a huge innovation and deserves the initiative. It can be a waste of time looking for a place to park the car, but using our application it will be easier.

4.7 Future Recommendation

1. Navigation and Directions: Integrate navigation and directions within the mobile app to guide users to their reserved parking spots. This feature can utilize GPS and mapping services to provide accurate directions and minimize parking-related stress.
2. Reservation and Payment: Enable users to reserve parking spots in advance through the mobile app. Implement a secure payment system that allows users to pay for their parking electronically. Integration with popular payment gateways will enhance convenience.
3. Notifications: Send notifications to users regarding their parking reservations, upcoming reservation expiration, or if any changes occur

(e.g., parking spot unavailability). Notifications can be delivered via push notifications or SMS.

4. Parking Guidance System: Implement a parking guidance system within the mobile app that suggests available parking spots based on the user's destination. This feature can optimize parking availability and reduce traffic congestion within parking lots.
5. Integration with Smart City Infrastructure: Integrate the smart parking system with the larger smart city infrastructure. For example, integrate with traffic management systems to optimize traffic flow in congested areas and reduce parking search time.
6. Analytics and Insights: Collect data on parking utilization, peak hours, and user behavior to gain insights and optimize the parking system's efficiency. This data can be used to plan for future expansions or improvements.
7. Sustainability: Incorporate sustainability features such as electric vehicle (EV) charging stations in select parking spots, encouraging the use of EVs. Integrate renewable energy sources to power the parking system, reducing its environmental impact.

References

- https://chintglobal.com/blog/iot-based-smart-parking-system/

- https://www.mokosmart.com/smart-parking-system-using-iot/

- https://www.hindawi.com/journals/wcmc/2020/9179530/?utm_source=google&utm_medium=cpc&utm_campaign=HDW_MRKT_GBL_SUB_ADWO_PAI_DYNA_JOUR_X_PCUPS_Sage&gclid=EAIaIQobChMIn5H31bmK-wIVkO53Ch0oEQL7EAAYASAAEgK2o_D_BwE

- https://www.mokosmart.com/smart-parking-system-using-iot/

الملخص العربي

هذا تقرير مشروع عن تطبيق الجوال "Rakenny" المستند إلى نظام ذكي لمواقف السيارات. خلال تطوير هذا المشروع، استكشفنا أفكارًا ووظائف جديدة.

يعتبر هذا المشروع نتاجًا لتخطيطنا وجدولتنا ومهاراتنا في البرمجة والعمل الجاد، ويعكس هذا التقرير خطواتنا التي اتخذناها على مستويات مختلفة من المهارات في البرمجة والتخطيط والجدولة.

طريقة فعالة وذكية لتوتير إدارة نظام مواقف السيارات التي تخصص مساحة وقوف سيارة فعالة باستخدام تكنولوجيا الإنترنت للأشياء. توفر الإنترنت للأشياء وصولًا لاسلكيًا إلى النظام ويمكن للمستخدم أن يتابع توافر منطقة وقوف السيارات. هدف هذا البحث هو حل هذه المشكلة. يضيع المستخدم عادة وقته وجهوده في البحث عن توافر المساحة الفارغة في منطقة وقوف السيارات المحددة.