THE UNIVERSITY OF NAIROBI DEPARTMENT OF COMPUTING AND INFORMATICS

VEHICLE PARKING ASSISTANT AND MANAGEMENT USING IOT, MOBILE AND COMPUTER VISION

BY

OGLE ABDIRAHMAN ABDULLAHI

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE OF THE UNIVERSITY OF NAIROBI.

AUGUST, 2023

DECLARATION

I declare that this project report and accompanying project implementation, submitted to the Department of Computing & Informatics,Faculty of Science and Technology, University of Nairobi, for the award of a Degree of Bachelor of Science in Computer Science, is my original work and has to the best of my knowledge, not been submitted to any other institution of higher learning for any award.

Signed: _____ Date: 15 AUGUST, 2023

Ogle Abdirahman Abdullahi

P15/136896/2019

This project report has been submitted in partial fulfillment of the requirements of the Degree of Bachelor of Science in Computer Science of The University of Nairobi with my approval as the University Supervisor.

Signed:_____ Date: _____

Professor. Elisha Toyne O Opiyo

Department of Computing and Informatics,

University of Nairobi

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to God for the blessings, strength, and guidance throughout the development of this car parking project. Without His unwavering support, this endeavor would not have been possible.I would like to extend my heartfelt appreciation to my parents for their constant encouragement, love, and belief in my abilities. Their unwavering support has been a constant source of motivation for me to pursue this project and strive for excellence.I am immensely grateful to my project supervisor, Prof. Elisha Toyne O. Opiyo for his invaluable guidance, expertise, and mentorship throughout this journey. Together with Dr. Samuel Ruhiu and Mr. Eric Ayienga, their insightful feedback, constructive criticism, and dedication have played a pivotal role in shaping the direction and success of this project. I am truly indebted to them for their patience, encouragement, and wisdom.I would also like to acknowledge my fellow students, particularly Salehudeen Saleh and Hassan Sharmake, who have been a source of support and positive criticism throughout this project. Their valuable insights, discussions, and collaboration have contributed significantly to the improvement and refinement of the ideas presented in this work. Their presence and contributions have made this journey more fulfilling and enjoyable.Additionally, I would like to express my gratitude to all those individuals who have directly or indirectly contributed to the completion of this project. Their assistance, encouragement, and feedback have been immensely valuable in shaping the outcome.Lastly, I would like to acknowledge The University of Nairobi for providing the necessary resources, infrastructure, and opportunities that have enabled me to undertake this project. Their commitment to fostering a conducive learning environment has played a vital role in my personal and academic growth.Once again, I extend my heartfelt thanks to all those mentioned above and to anyone else who has contributed to this project in any way. Your support, guidance, and encouragement have been instrumental in the successful completion of this car parking project.

God Bless You All

DEDICATION

I dedicate this project to God Almighty, who has given me the strength, wisdom, and knowledge to complete this work. I also dedicate this project to my parents, who have been a constant source of love, support, and encouragement throughout my life. To my family, who has always been there for me, thank you for your unwavering support and belief in me.

ABSTRACT

This project aims to develop a car parking system that utilizes Internet of Things (IoT) technology and an Arduino microcontroller to provide a convenient and efficient parking experience for users. The system is designed to be used with a mobile application that allows users to check the availability of parking spaces, reserve a spot, and pay for parking. The IoT system employs a range of sensors and devices, such as ultrasonic sensors, RFID readers, and LED lights, to monitor and manage the parking spaces. The Arduino microcontroller acts as the central processing unit of the system, receiving and processing data from the sensors and devices, and controlling the operation of the parking system. The solution implemented is a highly reliable and efficient car parking system that minimizes the time and effort required for users to find a parking spot and pay for parking. This report provides a detailed description of the system design and implementation, including hardware and software components, and a discussion of the system's performance and potential for future development.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	-	Application Programming Interface
IOT	-	Internet of Things
LCD	-	Liquid Crystal Display
Cnt	-	Count
SPCS	-	Spaces
NODEMCU	-	Node Microcontroller Unit
ESP8266	-	Espressif Systems' ESP8266
IDE	-	Integrated Development Environment
JS	-	JavaScript
LoRaWAN	-	Long Range Wide Area Network
PC	-	Personal Computer
USB	-	Universal Serial Bus
MYSQL	-	My Structured Query Language
I2C	-	Inter-Integrated Circuit
RAM	-	Random Access Memory
GHz	-	Gigahertz

Table Of Contents

CHAPTER ONE: INTRODUCTION

1.0 Background

The problem of finding a parking spot in urban areas has been a long-standing issue, with limited space and a growing population leading to an increase in demand for parking. This issue is compounded by the lack of real-time monitoring and control in traditional parking systems, which often leads to mismanagement and confusion.

In recent years, there has been a growing interest in using IoT technology to develop smart parking systems that can automate and optimize the parking process. These systems can use sensors, cameras, and other devices to monitor parking spaces in real-time, providing accurate information on available spots and streamlining the parking process for users. The data collected by these sensors can be analyzed to provide insights into parking patterns and demand, enabling parking lot owners and operators to optimize their operations and provide better services to their customers.

Arduino microcontrollers have become a popular choice for developing IoT-based systems due to their small size, low cost, and high processing power. The Arduino platform provides a flexible and customizable environment for building a variety of IoT applications, including parking systems.

The objective of this project is to develop a smart car parking system that utilizes IoT technology and an Arduino microcontroller to provide a convenient and efficient parking experience for users. By integrating the parking system with a mobile application, users can easily find and reserve a parking spot, pay for parking, and receive notifications about parking availability. The system's real-time monitoring and control features enable effective management of parking spaces, reducing congestion and improving the overall parking experience for users.

Overall, this project aims to address the challenges of traditional parking systems by providing a smart parking solution that is convenient, efficient, and reliable. The project also contributes to the growing body of research in the field of IoT and smart cities, demonstrating the potential of these technologies to transform urban environments and improve the quality of life for residents

1.1 Problem Definition

In urban areas, finding a parking spot is a major challenge for drivers, leading to traffic congestion, frustration, and wastage of time and fuel. The traditional parking systems are often inefficient, time-consuming, and prone to errors, causing further problems such as illegal parking and mismanagement of parking spaces. Additionally, the growing population and the increase in the number of cars on the roads have made the existing parking facilities insufficient to meet the needs of the users, leading to a shortage of parking spaces and overcrowding.

Moreover, the manual parking systems lack real-time monitoring and control, making it difficult to manage the parking spaces effectively. This results in confusion, mismanagement, and delays for users. The traditional parking systems do not provide any way for users to reserve a parking spot, making it challenging for them to plan their parking in advance. Payment for parking is also inconvenient, requiring physical tickets or cash, leading to long queues and delays.

Another issue is the lack of accessibility for disabled users, who often struggle to find a suitable parking spot that meets their needs. The current parking systems do not provide adequate facilities for disabled users, leading to discrimination and exclusion.

Car parking is also a significant problem during sport and cultural events, as well as in the proximity of administrative buildings or banks. The lack of knowledge of the current number of parking spaces can lead drivers to fully occupied parking lots, which consequently leads to having to move to another location, along with searching for a spot on another parking lot. This results in a waste of time, fuel, and resources, and it causes the deterioration of the traffic situation, increasing the levels of pollution and affecting the environment negatively. Furthermore, the local authorities often have to react with regulations, causing inconvenience to motorists.

The current outdoor car park management is dependent on human personnel keeping track of the available parking lots, which is not efficient, reliable, or cost-effective. This approach results in a lot of manual work and can lead to errors in monitoring and controlling the parking spaces. Moreover, the traditional parking systems do not provide any way for users to reserve a parking spot, making it

challenging for them to plan their parking in advance. Payment for parking is also inconvenient, requiring physical tickets or cash, leading to long queues and delays.

To address these challenges, this project aims to develop an IoT-based car parking system that uses an Arduino microcontroller and a mobile application to provide a convenient and efficient parking experience for users. The system will use a range of sensors and devices, such as infrared sensors, LCD displays and servomotors, to monitor and manage the parking spaces. The mobile application will enable users to check the availability of parking spaces, reserve a spot, and pay for parking seamlessly. The system's real-time monitoring and control features will enable effective management of parking spaces, reducing congestion and improving the overall parking experience for users.

By addressing the challenges of traditional parking systems, this project aims to provide a smart parking solution that is convenient, efficient, and reliable. The system will also provide facilities for disabled users, ensuring that they have equal access to parking spaces. Overall, this project has the potential to transform the parking experience for users and contribute to the development of smarter, more accessible, and sustainable cities.

1.2 Objectives

The objective of this counterfeit book recognition system project is to combat the piracy in the publishing industry;

1.2.1 Research Objectives

The following objectives were used to guide the research conducted:

i. To investigate the structure of parking areas in Nairobi and other major cities of the World.

ii. To evaluate the application of IoT systems in management of parking.

iii. To investigate the existence of other mobile parking applications that currently exist.

1.2.2 System Objectives

The system development objectives include;

i. To identify and define the requirements of the car parking system to ensure that the system will meet the needs of the end-users.

ii. This objective involves designing the car parking system based on the requirements identified in the requirement analysis.

iii. To develop a mobile application that will enable users to check the availability of parking spaces, reserve a spot, and pay for parking seamlessly.

iv. To design and develop an IoT-based smart parking system using Arduino technology and a range of sensors, that will enable real-time monitoring and control of parking spaces.

v. To integrate the mobile application and the IoT-based smart parking system to ensure seamless communication and coordination between the two systems.

1.3 Scope

The scope of the car parking system project includes the design, development, and implementation of an IoT-based smart parking system, integrated with a mobile application. The system will automate parking management, enabling users to check the availability of parking spaces. The system will also provide real-time information on parking availability, reducing the time and effort required to find a parking space, and improving the overall parking experience for users.

The project will involve the use of microcontrollers, sensors, and communication protocols to develop a scalable and reliable smart parking system. The mobile application will be developed for both Android and iOS platforms, providing a user-friendly interface for accessing parking services. The system will be designed to operate in outdoor environments and will be tested to ensure that it is robust and reliable under various conditions.

The project will also involve conducting a feasibility study to determine the economic viability of the system and its potential impact on the environment. The project will be limited to the development of a prototype system, and further development and commercialization will be considered based on the outcomes of the feasibility studies.

1.4 Project Justification

The car parking system project is motivated by the current challenges of managing parking spaces in urban areas. With the increase in the number of vehicles on the roads, the shortage of parking spaces has resulted in congestion, frustration, and a waste of time and fuel for drivers. Traditional parking management systems are prone to errors and inefficiencies, relying on human personnel to track the availability of parking lots.

To address these challenges, the proposed IoT-based smart parking system, integrated with a mobile application, aims to provide an efficient and user-friendly way for drivers to access parking services. The system will utilize sensors and communication protocols to provide accurate data on parking availability, enabling drivers to locate available parking spaces quickly and easily.

The system will also be scalable and reliable, meeting the needs of a growing number of users. In addition, the proposed car parking system project aligns with the trend towards the use of smart technologies to improve urban mobility and sustainability. The system will reduce the negative impact of parking on the environment by reducing the number of cars circling for a parking spot, thereby reducing emissions and congestion.

In conclusion, the car parking system project is driven by the need for a more efficient and sustainable parking management system that meets the needs of drivers and reduces the negative impact of parking on the environment. The proposed IoT-based smart parking system, integrated with a mobile application, has the potential to provide a viable solution to the growing problem of parking shortages in urban areas.

1.5 Assumptions

As with any project, certain assumptions are made in the car parking project. Some of these assumptions include:

i. Availability of power: It is assumed that there will be a reliable power supply to power the IoT-based car parking system and the mobile application.

ii. Adequate network connectivity: It is assumed that there will be adequate network connectivity in the car parking area to facilitate communication between the parking sensors, the IoT system, and the mobile application.

iii. User acceptance: It is assumed that users will readily adopt the proposed car parking system and that they will be willing to use the mobile application to locate available parking spaces.

iv. Parking lot design: It is assumed that the parking lot design is suitable for the installation of the IoT-based car parking system and that there will be no physical obstructions to the installation and functioning of the parking sensors.

v. Sensor accuracy: It is assumed that the parking sensors used in the IoT-based car parking system will be accurate in detecting the presence or absence of parked vehicles.

These assumptions are made based on available information and are subject to change based on actual project conditions. The project team will need to continuously monitor and assess these assumptions to ensure that they remain valid throughout the project.

CHAPTER TWO: LITERATURE REVIEW

Car parking is a crucial aspect of modern transportation, and its proper management has become increasingly important in many cities across the world. The growing number of vehicles on the road has resulted in a scarcity of parking spaces, particularly in urban areas. As a result, motorists often spend significant amounts of time searching for available parking spaces, leading to traffic congestion, fuel wastage, and increased carbon emissions. To address this challenge, there is a growing need to develop smart parking systems that leverage Internet of Things (IoT) technology and mobile applications to provide efficient and reliable parking management.

The aim of this literature review is to investigate the current state of car parking management in major cities worldwide and evaluate the potential of IoT-based parking systems and mobile applications in addressing the challenges associated with car parking.Through this literature review, we aim to identify the gaps in the current literature and propose recommendations for the design and implementation of an effective car parking system that leverages IoT technology and mobile applications. By doing so, we hope to contribute to the development of sustainable and efficient transportation systems in major cities worldwide.

2.1 To investigate the structure of parking areas in Nairobi and other major cities of the world.

Car parking is a significant issue in urban areas worldwide, with the increasing number of vehicles leading to a shortage of parking spaces. The lack of adequate parking spaces often results in parking-related issues, such as congestion, inconvenience, and delays. The situation is particularly dire in major cities worldwide, where traffic congestion and pollution are rampant. In Nairobi, for instance, the issue of inadequate parking spaces has become a significant challenge, resulting in a myriad of problems such as double parking, illegal parking, and other parking-related offenses. As a result, it is crucial to investigate the current structure of parking areas in major cities worldwide, including Nairobi, to identify the challenges associated with car parking management.

Certainly! Nairobi is the capital city of Kenya, located in the eastern part of Africa. As with many other urban areas, Nairobi has experienced rapid growth in

population and vehicles over the years, leading to a significant demand for parking spaces. In the city center and other densely populated areas, parking can be a major challenge.

The parking areas in Nairobi are mainly divided into two categories: off-street parking and on-street parking. Off-street parking refers to parking spaces within designated parking lots and garages, while on-street parking refers to parking spaces located along the sides of roads.

Off-street parking provides a more organized and secure parking option for car owners. In Nairobi, the off-street parking structures are mostly located in shopping malls, office buildings, and public parking lots. However, the main challenge with off-street parking is the limited space available, which often results in congestion and high demand. This situation leads to issues such as double parking and long queues, which cause traffic congestion and can be a safety hazard. Additionally, the cost of using off-street parking in Nairobi is relatively high, which can be a deterrent for some car owners. Off-street parking in Nairobi is primarily provided by commercial entities such as shopping malls, office complexes, and hotels. Some of the largest parking lots in Nairobi include the Two Rivers Mall, the Sarit Centre, and the Village Market. These parking lots are equipped with parking meters or payment systems to manage parking fees.

On-street parking, on the other hand, is more widely available and accessible to car owners in Nairobi. However, the challenges with on-street parking are numerous. One major issue is the lack of designated parking areas in public spaces, which leads to haphazard parking and obstructing traffic flow. The absence of clear regulations and enforcement of parking rules also exacerbates the problem. As a result, parking violations such as illegal parking, parking in no-parking zones, and blocking of sidewalks are prevalent in Nairobi. These violations create chaos and worsen the already poor traffic situation in the city. On-street parking, on the other hand, is mostly provided by the city government. In Nairobi, on-street parking is mainly available along major roads and in designated parking bays. Parking fees for on-street parking are also collected through payment systems such as parking meters.

Despite the availability of both on-street and off-street parking, Nairobi still faces a shortage of parking spaces, especially in the city center and other high-traffic areas. This has resulted in illegal parking and congestion, leading to traffic gridlocks and delays.

One of the major challenges associated with car parking management is the lack of adequate parking infrastructure. The majority of parking areas in major cities are either insufficient or outdated, resulting in congestion and inconvenience for motorists. In Nairobi, the situation is particularly dire, with many parking areas being unsuitable and ill-equipped to handle the growing number of vehicles. Additionally, the high cost of parking in major cities often discourages car owners from using parking facilities, resulting in illegal parking and other parking-related offenses.

Another significant challenge associated with car parking management is the lack of parking regulations and enforcement. The absence of clear parking regulations often results in chaos and confusion, with motorists frequently double-parking and parking in unauthorized areas. In Nairobi, for instance, the lack of clear parking regulations has resulted in a significant increase in illegal parking and other parking-related offenses, leading to traffic congestion and delays.

Furthermore, the lack of proper maintenance and management of parking facilities is another significant challenge associated with car parking management. Many parking areas in major cities are poorly maintained, resulting in inadequate lighting, insufficient signage, and inadequate security. This situation often discourages motorists from using parking facilities, leading to an increase in illegal parking and other parking-related offenses.

To address the challenges associated with car parking management, there is a need to develop effective parking management strategies that leverage IoT technology and mobile applications. Such strategies could include the deployment of smart parking systems that use real-time data to monitor parking spaces and provide motorists with real-time information on available parking spaces. Additionally, the development of effective parking regulations and enforcement policies could help alleviate parking-related issues in major cities worldwide.

In conclusion, the first research objective of the literature review, which is to investigate the current structure of parking areas in major cities worldwide, including Nairobi, and identify the challenges associated with car parking management, has revealed the numerous challenges that exist in car parking management. The lack of adequate parking infrastructure, parking regulations, and enforcement, and proper maintenance and management of parking facilities are some of the significant challenges that must be addressed. To overcome these challenges, there is a need to develop effective parking management strategies that leverage IoT technology and mobile applications.

2.1.1 A comparison of Nairobi with other major cities in the world

In comparing Nairobi with other major cities that have better parking systems, we can consider two examples: Tokyo, Japan and Barcelona, Spain.

2.1.1.1 Tokyo, Japan

Tokyo is known for having one of the most advanced and efficient parking systems in the world. In Tokyo, the local government has implemented various initiatives to manage the demand for parking spaces and reduce traffic congestion. One such initiative is the use of intelligent parking systems (IPS) that rely on sensors and cameras to monitor the availability of parking spaces in real-time. The information is then relayed to drivers through electronic displays and mobile applications, allowing them to quickly locate and reserve available parking spaces.

The city has invested in developing large off-street parking facilities, with over 300,000 off-street parking spaces available for public use. In addition, the city has an advanced parking guidance system that provides real-time information on the availability of parking spaces to drivers. The system is linked to a mobile application that allows drivers to locate available parking spaces before arriving at their destination. This reduces the time spent searching for parking and helps to alleviate congestion on the city's streets.

In addition to IPS, Tokyo has also implemented a comprehensive parking fee system that takes into account factors such as location, time of day, and duration of

parking. The system is designed to encourage the use of public transportation and reduce the number of cars on the road during peak hours.

Furthermore, Tokyo has established partnerships with private parking providers to increase the number of available parking spaces. These partnerships have led to the development of multi-story parking garages and underground parking lots that utilize vertical parking systems, thereby maximizing the use of limited space.

Overall, Tokyo's parking system is an excellent example of how a combination of technology, efficient pricing mechanisms, and collaboration between government and private stakeholders can result in a well-managed and sustainable parking system.

2.1.1.2 Barcelona, Spain

Barcelona's parking system is another example of an advanced and efficient system. The city has adopted a policy of reducing the number of cars on the road and promoting sustainable mobility. The parking management in Barcelona is based on two main principles: regulated parking and controlled access to the city center.

The city has implemented a comprehensive on-street parking management system, with over 2,500 sensors installed in parking spaces across the city. These sensors provide real-time information on parking space availability, which is then relayed to drivers via mobile applications and electronic signs. The city has also implemented a dynamic pricing system that adjusts the cost of parking based on demand, encouraging drivers to park in less busy areas of the city. This has led to a reduction in congestion and improved traffic flow.

One of the key features of Barcelona's parking system is the use of a dynamic pricing model based on the demand for parking. The pricing model uses sensors to monitor the occupancy of parking spaces in real-time, and adjusts the price of parking accordingly. This encourages drivers to park in less busy areas or use alternative modes of transportation.

In addition, Barcelona has implemented a number of measures to promote sustainable transportation, such as the introduction of a bike-sharing program and the construction of more bike lanes. The city has also established an integrated

public transport system that includes buses, metro, and tram lines, making it easier for people to commute without a car.

Furthermore, Barcelona has implemented smart parking technology, including the use of sensors and mobile applications, to help drivers find available parking spaces quickly and easily. The system also allows drivers to pay for parking remotely, reducing the time and effort required to find a parking spot.

Overall, Barcelona's parking system is an excellent example of how a combination of dynamic pricing, sustainable transportation, and smart parking technology can result in a well-managed and sustainable parking system.

2.1.1.3 Nairobi from Tokyo and Barcelona’s perspectives

In comparison to these cities, Nairobi faces significant challenges in managing its parking system. The city has limited off-street parking facilities, which has led to congestion on its streets. On-street parking is also poorly managed, with insufficient enforcement and a lack of real-time information on parking space availability. These challenges have led to significant traffic congestion, wasted time and fuel for drivers, and a negative impact on the city's environment.

Therefore, by studying and adopting best practices from cities like Tokyo and Barcelona, Nairobi can improve its parking system and reap the benefits of reduced congestion, improved traffic flow, and increased revenue from parking fees.

2.2 To evaluate the use of IoT technology in parking management and analyze the potential benefits of implementing such systems.

In recent years, the application of IoT systems in parking management has been gaining popularity due to the increasing demand for efficient parking systems. IoT systems in parking management use various technologies to monitor and manage parking spaces, which include sensors, mobile applications, and cloud-based platforms.

One of the main benefits of IoT systems in parking management is the ability to reduce traffic congestion caused by drivers looking for parking spaces. The sensors installed in parking spaces provide real-time information on the availability of parking spaces, which is communicated to drivers via mobile applications or other platforms. This reduces the amount of time drivers spend searching for parking spaces and, in turn, reduces traffic congestion.

The following are some examples of IoT devices and technologies used in car parking systems:

i. Ultrasonic sensors: These sensors use high frequency sound waves to detect the presence of a vehicle in a parking space. They are commonly used in car parking systems as they are easy to install and provide accurate results.

ii. License plate recognition: This technology uses cameras to capture images of a vehicle's license plate, which are then processed to determine if the vehicle is authorized to park in a specific area. This is useful for managing restricted parking spaces, such as those reserved for residents or disabled persons.

iii. Smart parking meters: These meters use wireless communication to connect with a central server, allowing parking operators to monitor parking activity and adjust prices based on demand. They can also provide real-time information to drivers about available parking spaces in the area.

iv. Parking guidance systems: These systems use sensors and signage to guide drivers to available parking spaces in a parking garage or lot. They can also provide real-time information on the location and availability of parking spaces.

v. Automated parking systems: These systems use robotics and sensors to park and retrieve vehicles in a parking garage. They are particularly useful in areas where space is limited, as they can park vehicles closer together than a human driver

could.

These are just a few examples of the many IoT devices and technologies used in car parking systems today. Each system will have unique requirements and may use a combination of different devices to meet its needs.

Additionally, IoT systems can provide efficient parking management by collecting data on parking usage patterns, which can be used to optimize the use of parking spaces. For instance, if data collected over time shows that a particular parking space is rarely used, the parking management system can reallocate it for another use, such as bike parking or public amenities.

However, the implementation of IoT systems in parking management faces some challenges. One of the challenges is the high cost of installation and maintenance of the technology. The sensors used to monitor parking spaces can be expensive to install, and the technology used to transmit data from the sensors to the management system can be costly as well.

Another challenge is the need for reliable communication infrastructure. The sensors used to monitor parking spaces need to be able to communicate with the management system in real-time, which requires reliable communication infrastructure. In areas with poor communication infrastructure, the effectiveness of IoT systems in parking management can be limited.

In conclusion, the application of IoT systems in parking management has the potential to revolutionize the way parking is managed in urban areas. However, the high cost of installation and maintenance, as well as the need for reliable communication infrastructure, presents challenges that need to be addressed for the full benefits of IoT systems in parking management to be realized.

2.3 To investigate other mobile parking applications that currently exist

The objective aims to provide insights into the current state of mobile parking applications and identify potential gaps in the market that can be addressed through the development of a new application.

Mobile parking applications are becoming increasingly popular, especially in urban areas where finding a parking spot can be a challenge. These applications offer drivers the convenience of finding and reserving parking spaces in advance, as well as making payments electronically. As a result, there are numerous mobile parking applications currently available in the market.

One of the most popular mobile parking applications is Parkmobile, which is available in multiple cities across the United States and Europe. Parkmobile allows users to search for available parking spaces and reserve them in advance, as well as pay for parking using their mobile devices. Another popular mobile parking application is PayByPhone, which is available in over 200 cities worldwide. PayByPhone allows users to search for parking spaces, pay for parking, and even extend their parking time remotely using their mobile devices.

Despite the existence of numerous mobile parking applications, there are still potential gaps in the market that can be addressed through the development of a new application. For example, some applications may not be available in certain areas, or they may not offer the specific features that drivers need. By identifying these gaps, a new mobile parking application can be developed to address these specific needs.

2.4 Review of current systems

2.4.1 Nairobi County Parking App

The Nairobi County Parking App is a mobile application designed for the citizens of Nairobi to make it easier to pay for parking fees and avoid the long queues that were common with manual parking fee payments. This app is a result of the partnership between the Nairobi City County Government and JamboPay, a payment gateway provider.

The Nairobi County Parking App has several features that make it easy to use. First and foremost, it allows users to create a parking account, which can be used to pay for parking fees electronically. The app also allows users to view their parking history, which can help them keep track of their parking expenses. Additionally, the app has a feature that sends notifications to users when their parking time is about to expire, allowing them to renew their parking remotely.

Additionally, the app has made it easier for the Nairobi County Government to collect parking revenue, as it allows for easier tracking of payments and reduces the number of cases of revenue leakage.

Advantages

i. Convenience: The app offers users a convenient way to pay for their parking fees from anywhere and at any time using their mobile phones.

ii. Time-saving: The app saves users the time and effort of having to physically go to the parking ticket machines to purchase parking tickets.

iii. Cashless payments: The app allows users to make cashless payments for their parking fees, which eliminates the need for carrying cash or loose change.

iv. Flexibility: The app provides users with the flexibility to extend their parking sessions remotely, and also to top-up their parking fees in case they exceed their initial payment.

Disadvantages

i. Limited functionality: The app only allows users to pay for parking fees, and does not provide other parking-related information such as the availability of parking spaces or the location of parking lots.

ii. Network connectivity issues: The app is heavily reliant on network connectivity, and in case of network downtime or slow network speeds, users may experience difficulties in making their payments.

iii. User experience: Some users have reported difficulties in navigating the app and making payments, which may lead to frustration and discourage the use of the app.

iv. Limited coverage: The app is currently only available for parking payments in Nairobi County, and users cannot use it to pay for parking fees in other counties or cities.

2.4.2 Easy Parking Kenya

Easy Parking Kenya is a mobile application designed to make parking easier and more convenient for drivers in Nairobi. The app provides a platform for users to find parking spaces, reserve and pay for them in advance.

One of the advantages of Easy Parking Kenya is its user-friendly interface. The app is easy to navigate, making it simple for users to find available parking spaces in real-time. Additionally, the app provides users with detailed information about parking fees, location and distance from their destination.

Another advantage of Easy Parking Kenya is the option to pre-book parking spaces. This feature enables users to reserve parking slots in advance, which saves time and reduces the risk of arriving at a fully occupied parking lot. The app also provides a convenient payment option, allowing users to pay for parking via mobile money, debit or credit card.

However, Easy Parking Kenya has some disadvantages. One of the main drawbacks is the limited coverage area. The app only covers a few areas in Nairobi, which means that users in other areas cannot benefit from its services. Additionally, the app does not provide real-time updates on the availability of parking spaces, which can lead to inconvenience and frustration for users who find the parking lot full upon arrival.

Furthermore, Easy Parking Kenya does not provide directions to the parking lot. This can be a significant challenge for new users who are not familiar with the area. As a result, users may have to rely on external navigation apps or ask for directions, which can be time-consuming and frustrating.

In conclusion, while Easy Parking Kenya has several advantages, such as a user-friendly interface and the ability to pre-book parking spaces, its limited coverage area, lack of real-time updates and direction to parking lots are significant disadvantages. These limitations highlight the need for a more comprehensive and integrated car parking mobile application in Nairobi.

2.4.3 Pango Parking

Pango Parking is a mobile application that provides drivers with the ability to pay for parking fees in a simple and convenient way. It allows users to register their car details and then select the parking zone where they want to park. Users can then pay

for the parking session through the app, which deducts the amount from their preloaded account or bills their credit card.

One advantage of Pango Parking is that it eliminates the need for drivers to carry cash or visit parking meters, which can save time and reduce the risk of theft. It also allows users to extend their parking session remotely, so they do not have to rush back to their car to top up the meter.

However, one disadvantage of Pango Parking is that it requires users to have access to a smartphone and internet connection to use the service, which may not be accessible to everyone. Additionally, the app has faced criticism for its high transaction fees, which can add up for frequent parkers.

Overall, Pango Parking is a convenient solution for those who want to pay for parking using their mobile devices. However, its high transaction fees and reliance on smartphones and internet connectivity may limit its accessibility to some users.

2.4.4 Easy ParkByPhone

Easy ParkByPhone is a mobile application that offers a convenient way for drivers to find and pay for parking spots. It allows users to search for parking spots near their location and view the availability of parking spaces in real time. The app also provides users with a map that shows the location of available parking spots, making it easier for them to find the spot.

Once a parking spot is selected, the user can easily pay for the parking session through the app, without the need for physical cash. The app supports multiple payment methods, including credit and debit cards, mobile money, and other digital payment platforms. The app also provides users with the option to extend their parking session remotely, as long as there are available parking spaces.

Easy ParkByPhone also offers a feature that enables users to set parking reminders, which sends notifications when the parking session is about to expire. This helps users avoid parking fines and penalties.

Advantages:

i. Easy to Use: The Easy ParkByPhone app is designed to be user-friendly, making it easy for drivers to navigate and use.
ii. Saves Time: With the app, drivers can pay for parking without having to physically visit a parking meter or a parking attendant, saving them time.
iii. Convenience: The app offers a convenient way for drivers to pay for parking as they do not have to carry cash or coins.
iv. Multiple Payment Options: Easy ParkByPhone offers multiple payment options, including M-PESA, Airtel Money, and credit cards, making it easy for drivers to pay for parking using their preferred method.
v. Parking History: The app provides drivers with a parking history, enabling them to keep track of their parking payments and expenses.

Disadvantages:

i. Limited Coverage: The Easy ParkByPhone app is not available in all parking areas, limiting its usefulness to only certain areas.

ii. Network Coverage: The app relies on network coverage to process payments, and this can be a challenge in areas with poor network coverage.

iii. Technical Issues: The app may experience technical issues from time to time, which can affect its performance.

iv. Additional Charges: The app charges a convenience fee for using its services, which can add up over time and increase the cost of parking.

v. Limited Features: Easy ParkByPhone only offers payment services and does not provide information on available parking spaces or directions to the parking areas.

2.5 Proposed System

The proposed car parking system will be a comprehensive solution to the challenges faced by car owners and parking managers in Nairobi. The system will incorporate the latest advancements in technology, including the Internet of Things (IoT) and mobile applications, to provide a seamless parking experience to users.

The system will comprise two main components: a mobile application and an IoT-based car parking system. The mobile application will be user-friendly and provide real-time information about available parking spaces and directions to the parking lot. Users will be able to reserve parking spaces in advance, pay for parking fees through the app, and receive alerts when their parking time is about to expire.

The IoT-based car parking system will include smart sensors installed in the parking spaces to detect the presence of vehicles. The sensors will be connected to a central server that will collect real-time data on parking space availability, parking duration, and fees paid. The system will also include a dashboard for parking managers to monitor and update the parking lot's occupancy and adjust prices .

Advantages

i. Improved parking efficiency: With the implementation of IoT devices and a mobile application, the proposed system will provide real-time information on parking spaces available in different parking areas. This will help reduce the time taken to find parking spaces and increase the number of cars that can be parked within a specific parking area.

ii. Reduced traffic congestion: Finding a parking space in Nairobi is often a challenge, and it often results in traffic congestion. With the proposed system, drivers can easily locate available parking spaces and reserve them in advance. This will help reduce traffic congestion as drivers will spend less time looking for parking spaces.

iii. Improved parking management: The proposed system will provide real-time information on parking spaces available in different parking areas. This will enable parking managers to monitor and manage parking spaces more effectively, leading to better parking space allocation and utilization.

iv. Enhanced security: The proposed system will provide secure payment options, eliminating the need for cash transactions. It will also provide real-time monitoring

of the parking areas, increasing security and reducing the risk of theft or vandalism.

v. Cost-effective: The proposed system will reduce the cost of parking management, as it will require less manpower and resources. It will also reduce the cost of fuel and time spent looking for parking spaces, leading to cost savings for drivers.

vii. Convenient: The proposed system will provide drivers with the convenience of finding and reserving parking spaces using a mobile application. It will also provide drivers with the convenience of making payments electronically, eliminating the need for cash transactions.

viii. Environmentally friendly: The proposed system will help reduce carbon emissions by reducing the time spent looking for parking spaces and reducing traffic congestion.

Overall, the proposed car parking system will provide significant advantages to drivers, parking managers, and the environment, making it a worthwhile investment for Nairobi County.Despite the benefits, the proposed car parking system may face some challenges during implementation. For instance, the cost of installing the IoT-based system and maintaining it may be high, requiring a significant investment. Additionally, there may be resistance from traditional parking operators who are accustomed to manual parking fee collection. However, the benefits of the proposed system outweigh the challenges, making it a viable solution to the challenges facing the car parking industry in Nairobi.

CHAPTER THREE: METHODOLOGY

3.1 System Development Methodology

The Agile methodology is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer satisfaction. It is based on four core values: individuals and interactions, working software, customer collaboration, and responding to change. These values are supported by a set of 12 principles that guide the Agile development process.

The Agile methodology promotes cross-functional teams working closely together to deliver high-quality software products in a timely and efficient manner. It involves breaking down the project into smaller, manageable parts called sprints. Each sprint focuses on delivering a specific set of features or functionality, with a clear definition of what success looks like at the end of each sprint.

Continuous planning is a key feature of Agile methodology, with regular meetings to review progress and make necessary adjustments to the project plan. Daily stand-up meetings are also a common practice in Agile, where team members gather to discuss what they accomplished the previous day, what they plan to do that day, and any roadblocks they are facing.

The Agile methodology promotes collaboration and communication among team members, as well as with stakeholders, including customers and end-users. This ensures that everyone involved in the project has a clear understanding of the project goals and requirements, and that the software being developed meets the needs and expectations of the end-users.

Regular testing and quality assurance are also key features of Agile methodology. This helps to identify and address any issues early in the development process, reducing the risk of delays and cost overruns. Regular customer feedback is also encouraged in Agile, allowing for adjustments to be made based on real-world user experiences.

In summary, the Agile methodology is an effective approach to software development that promotes flexibility, collaboration, and customer satisfaction. It enables teams to deliver high-quality software products in a timely and efficient manner, while also allowing for adjustments to be made throughout the development process based on feedback from stakeholders.

One of the key advantages of the Agile methodology is its flexibility, which allows

for changes to be made quickly and efficiently during the development process. This can be particularly beneficial for individual projects where there may be changes in requirements or scope.

Another advantage of the Agile methodology is its emphasis on collaboration and communication. This methodology encourages frequent meetings and open communication channels between the development team and stakeholders, including end-users and customers. This can help ensure that the system being developed meets the needs and expectations of those who will be using it.

Furthermore, the Agile methodology promotes iterative development, which means that a working product can be delivered incrementally, with new features and improvements being added over time. This can be particularly advantageous for individual projects, as it allows for the system to be developed and tested in smaller pieces, reducing the risk of delays or issues arising later in the development process.

Finally, the Agile methodology promotes continuous testing and quality assurance, which can help ensure that the system being developed is of high quality and meets the needs of the end-users. This can be particularly important for a car parking system, as it needs to be reliable and efficient to provide an optimal user experience.

3.1.1 How agile methodology would work for Car Parking System

i. Define the project scope: The first step in applying the Agile methodology to a car parking project is to define the project scope. This involves identifying the key features and functionality required for the car parking system, as well as any specific constraints or requirements.

ii. Create the product backlog: The product backlog is a prioritized list of features and requirements that need to be developed for the car parking system. This list should be continually updated throughout the development process based on feedback from stakeholders.

iii. Plan the first sprint: The next step is to plan the first sprint, which is a short development cycle typically lasting between one and four weeks. During this sprint, the development team will work on a subset of the features identified in the product backlog.

iv. Conduct daily stand-up meetings: Daily stand-up meetings are brief meetings held each day to review progress, discuss any issues or roadblocks, and plan the

next steps. These meetings should involve all members of the development team and should be kept short and focused.

v. Develop the car parking system: During each sprint, the development team will work on developing the car parking system. This involves writing code, testing, and integrating the features into the system.

vi. Conduct sprint reviews: At the end of each sprint, the development team will conduct a sprint review to demonstrate the features that were developed during the sprint and gather feedback from stakeholders. This feedback should be used to update the product backlog and plan the next sprint.

vii. Conduct sprint retrospectives: After each sprint, the development team will conduct a sprint retrospective to reflect on what went well during the sprint, what didn't go well, and what could be improved in the next sprint. This feedback should be used to improve the development process and increase efficiency and effectiveness.

viii. Repeat the process: The development process should be repeated for each sprint until all features in the product backlog have been developed and the car parking system is ready for release.

3.2 System Requirements

3.2.1 Hardware Requirements

For the system to be implemented, there must be;

i. A personal computer or laptop running on Windows operating System with at least 4GB RAM, 2 GHZ, 128 GB Hard disk.

ii. Android Mobile phone: internet enabled feature phone.

iii. Arduino Uno

iv. Infrared Sensors

v. 16 x 2 LCD with I2C

vi. Servo-motor

vii. Breadboard

viii. Jumper Wires

ix. USB cable

x. Wifi Module

3.2.2 Software Requirements

The following software resources will be required in the development of the system;

i. Visual studio code editor or Android Studio

ii. Node Package Manager v9.3.1

iii. Windows 8 or higher

iv. React native

v. Google Docs and Google Slides used for project documentation and presentation respectively.

vi. NodeJS and Express.

vii. MongoDB and MySQL.

viii. Expo Go.

ix. Google Cloud Services.

x. PayStack.

3.3 Research Methodology

3.3.1 Research Design:

The research will utilize a mixed-methods approach, combining qualitative and quantitative data collection techniques. This will involve the use of document analysis and informal interviews to collect data from car owners, parking attendants, and other stakeholders.

3.3.2 Data Collection Methods:

a) Document Analysis: The study will involve analyzing various reports, articles, and surveys on car parking in Nairobi. The documents to be analyzed will include those from KARA, ITDP, and Nairobi City County, among others. This will provide an overview of the current state of car parking in Nairobi, the challenges faced, and previous attempts to address the issue.

b) Informal Interviews: In-depth interviews will be conducted with parking

attendants, and other drivers to collect qualitative data on their experiences, opinions, and perspectives on car parking in Nairobi. The interviews will be conducted face-to-face and would not be recorded.

c) Observations: The study will also involve on-site observations of selected parking areas in Nairobi. The observations will provide insight into the current state of parking infrastructure, the availability of parking spaces, and the behavior of drivers and parking attendants.

3.3.3 Ethical Considerations:

The study will adhere to ethical guidelines, including obtaining informed consent from participants, maintaining confidentiality and anonymity, and ensuring data security.

3.3.4 Limitations:

The study may be limited by factors such as sampling bias, self-reporting bias, and social desirability bias. Efforts will be made to minimize these limitations by using appropriate data collection techniques and analytical methods.

3.3.5 Expected Outcome:

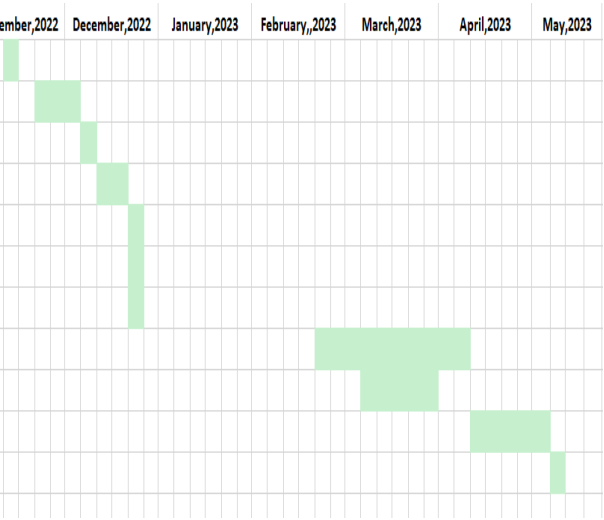
The study is expected to generate insights into the current state of car parking in Nairobi, including the challenges faced by car owners and parking attendants, the effectiveness of parking policies and regulations, and potential solutions to improve the parking situation. The findings will be presented in a comprehensive report that can be used by policymakers, parking operators, and other stakeholders to inform decision-making and planning.

3.4 Project Schedule

The project planning and implementation shall start from November 2019 to April 2020.

TABLE 3.1 PROJECT SCHEDULE

Task Name	Duration(days)	Planned Start Date	Planned End Date
Prepare project proposal(define problem and scope)	5	10/11/2022	15/11/2022
Research on all the topics listed in the research objectives	15	20/11/2022	5/12/2022
Feasibility studies and cost benefit analysis	5	5/12/2022	10/12/2022
Review and selecting technology	10	10/12/2022	20/12/2022
System Architecture Design	5	21/12/2022	25/12/2022
System Algorithm Flow Design	5	21/12/2022	25/12/2022
User Interface Design	5	21/12/2022	25/12/2022
Implement Mobile Application(frontend and backend)	45	25/2/2023	10/4/2023
Implementing the IoT section of the project	25	5/3/2023	30/3/2023
Integration of Mobile with IoT	25	10/4/2023	5/5/2023
System Testing	5	6/5/2023	10/5/2023



3.5 Resources

TABLE 3.2 RESOURCES

Item	**Cost (Ksh)**
Zuku Wifi for 7 months	10500/=
Development Machine(HP G4 Desktop) - Intel i7, 8gb RAM, 1TB Hard Disk + 256GB SSD	90,000/=
Arduino Uno(4)	8500/=
Infrared Sensors(28)	5600/=
Servo-motors(4)	1000/=
Jumper Wires	1200/=
Wifi Module	4000/=
LCD display with I2C(4)	3200
Google Maps Api	300USD
Total	**124000/= + 300USD**

CHAPTER FOUR: SYSTEM ANALYSIS

4.1 Introduction

System analysis is a critical process that involves examining complex systems to understand how they function, identify areas for improvement, and develop solutions to enhance their performance. It is a systematic approach to problem-solving that involves breaking down a system into its component parts and analyzing how they interact.

The goal of system analysis is to identify the strengths and weaknesses of a system and develop strategies to improve its overall effectiveness. This process involves examining the various components of a system, including hardware, software, and people, and analyzing how they work together.

System analysis is essential in many industries, including information technology, manufacturing, and healthcare. It is used to identify areas for improvement in processes and procedures, optimize resource allocation, and enhance the user experience.

4.2 Modeling Tools and Techniques

Modeling involves graphical methods and non-technical language that represent the system at different stages of development. Various tools are used to describe business processes, requirements and user interaction with the system. Different models like data flow diagrams and Unified modeling Language diagrams can be used for modeling.

4.2.1 UML Diagrams

It is a diagram based on the Unified Modeling Language with the intention of visualizing and documenting software systems. UML provides various graphical tools such as use case diagrams and sequence diagrams.

4.2.1.1 Use Case Diagrams

Use case diagrams are used to visually represent the interaction between users and the system.

1) Normal User(Driver)

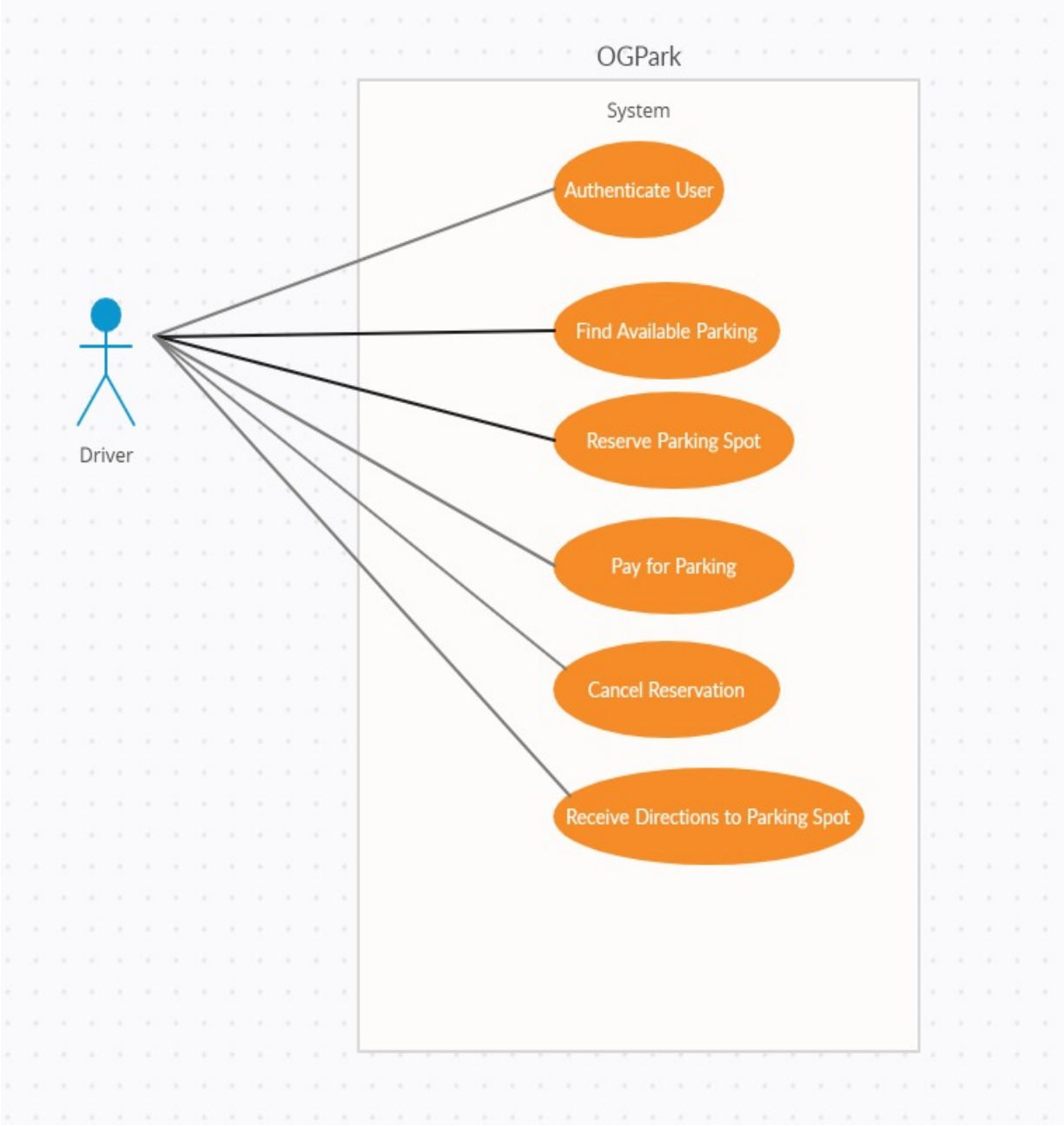


Figure 4.1 Normal user driver

i. Authenticate User:

This use case involves verifying the identity of the driver using the mobile application before they can access any features of the car parking system. The user can authenticate themselves using their email and password, or through their Google account. Once the user is authenticated, they can access the other use cases.

ii. Find Available Parking:

This use case enables drivers to find available parking spots within the car parking system. The mobile application retrieves data from the sensors installed in the parking lot, which shows the number of available parking spots. The mobile application then displays the information to the driver, who can select a parking spot and navigate to it.

iii. Reserve Parking Spot:

This use case allows drivers to reserve a parking spot in advance through the mobile application. The user selects a parking lot and the date and time they wish to reserve a spot for. They can then pay for the reservation through the mobile application. The car parking system will reserve the selected parking spot for the driver, and the driver can then park in the reserved spot when they arrive.

iv. Pay for Parking:

This use case enables drivers to pay for parking through the mobile application. The mobile application will display the parking rates and the driver can select the duration they wish to park for. The driver can then pay for the parking using a credit card or another payment method. The car parking system will verify the payment and allow the driver to park in the designated parking spot.

v. Cancel Reservation:

This use case allows drivers to cancel a parking reservation that they have previously made through the mobile application. The driver selects the reservation they wish to cancel, and the car parking system will cancel the reservation and release the parking spot for other drivers to use.

vi. Receive Directions to Parking Spot:

This use case enables the driver to receive directions to their selected parking spot

through the mobile application. The mobile application will use GPS data to guide the driver to the parking lot and then to the selected parking spot.

2) Business(Parking Provider)

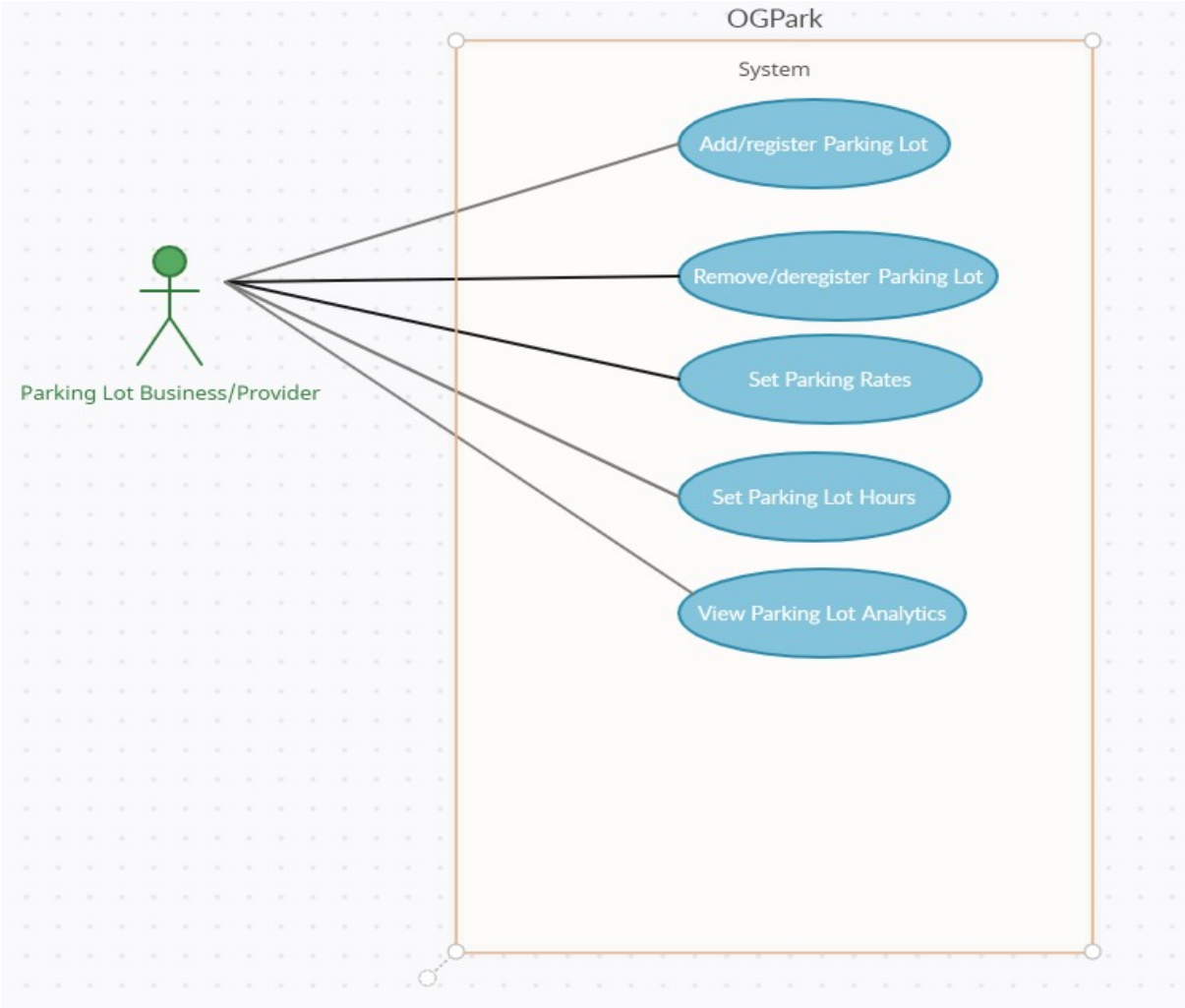


Figure 4.2 Business(Parking Provider)

i. Add Parking Lot:

This use case allows the parking lot owner to add a new parking lot to the car parking system. The owner enters the location and other details of the new parking lot, and the car parking system will create a new entry for the lot in the system.

ii. Remove Parking Lot:

This use case allows the parking lot owner to remove a parking lot from the car parking system. The owner selects the parking lot they wish to remove, and the car parking system will remove all data related to that parking lot from the system.

iii. Set Parking Rates:

This use case enables the parking lot owner to set the parking rates for each parking lot in the system. The owner can set different rates for different time periods, such as hourly rates or daily rates.

iv. Set Parking Lot Hours:

This use case allows the parking lot owner to set the hours of operation for each parking lot in the system. The owner can set the opening and closing times for the parking lot, as well as any special hours or closures.

v. View Parking Lot Analytics:

This use case enables the parking lot owner to view analytics and reports on the performance of each parking lot in the system. The owner can view data on the number of parking spots used, revenue generated, and other metrics to help them make informed decisions about managing the parking lots.

3) Parking Attendant

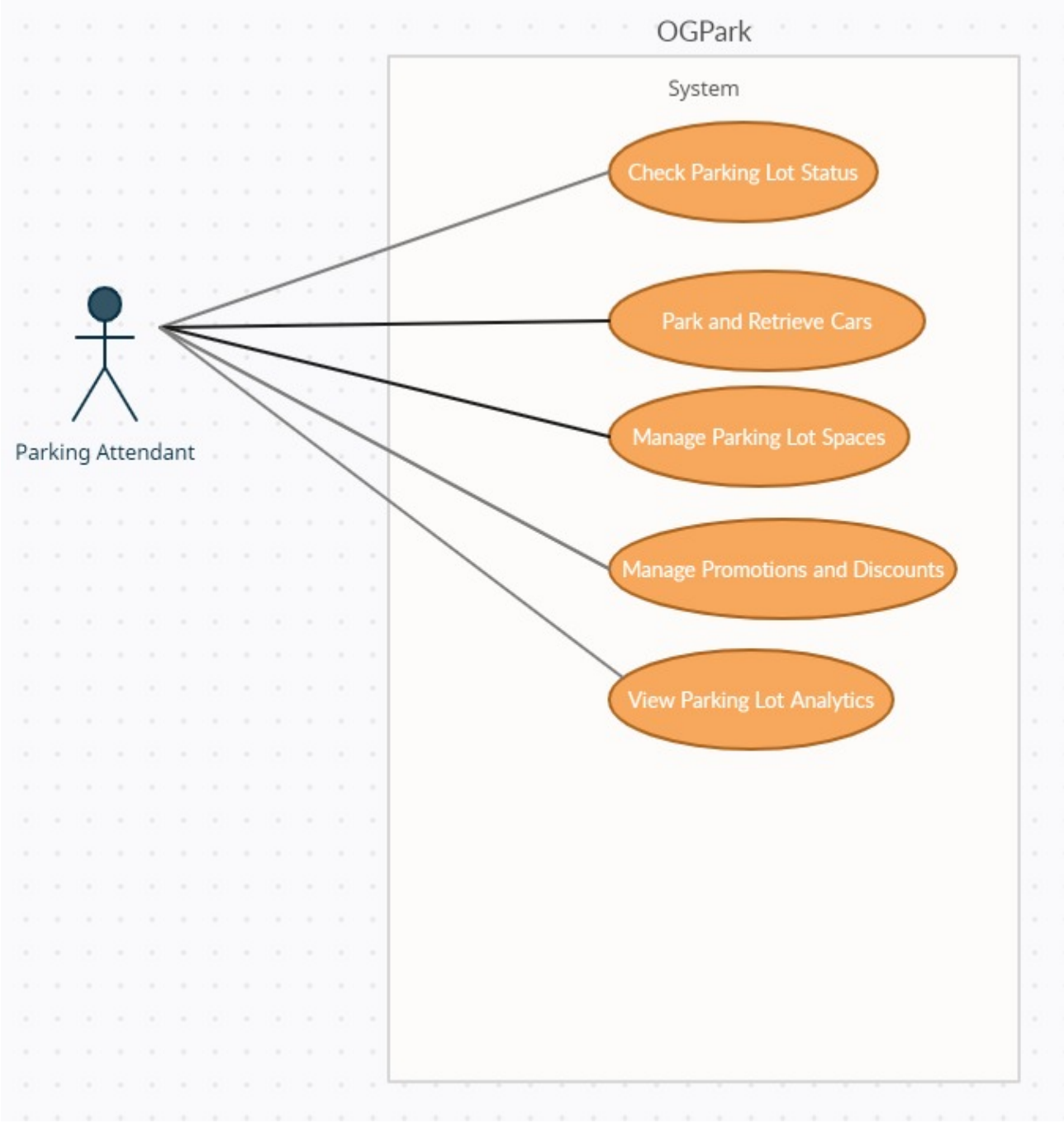


Figure 4.3 Parking Attendant

i. Check Parking Lot Status:

This use case enables the parking attendant to check the status of the parking lot using the mobile application. The parking attendant can view data from the sensors installed in the parking lot, such as the number of available parking spaces and the location of each space. This information helps the parking attendant guide drivers to available parking spots.

ii. Park and Retrieve Cars:

This use case enables the parking attendant to park and retrieve cars for customers in the parking lot. The attendant can use the sensors installed in the parking lot to guide them to an available parking spot. When a driver arrives at the parking lot, the attendant can park the car in an available spot and use the mobile application to mark the parking spot as occupied. When the driver returns to retrieve their car, the attendant can use the mobile application to locate the car and retrieve it for the driver.

iii. Manage Parking Lot Spaces:

This use case allows the parking attendant to manage the parking lot spaces in the parking lot. The attendant can use the mobile application to mark parking spots as occupied or available, and to update the status of parking spots as needed. This helps ensure that drivers can easily find available parking spots and that the parking lot is organized and efficient.

iv. Manage Promotions and Discounts:

This use case enables the parking attendant to manage promotions and discounts offered by the parking lot. The attendant can use the mobile application to view and update the promotions and discounts offered, and to apply them to customer transactions as needed. This helps the parking lot attract and retain customers and helps to increase revenue.

4) System Administrators

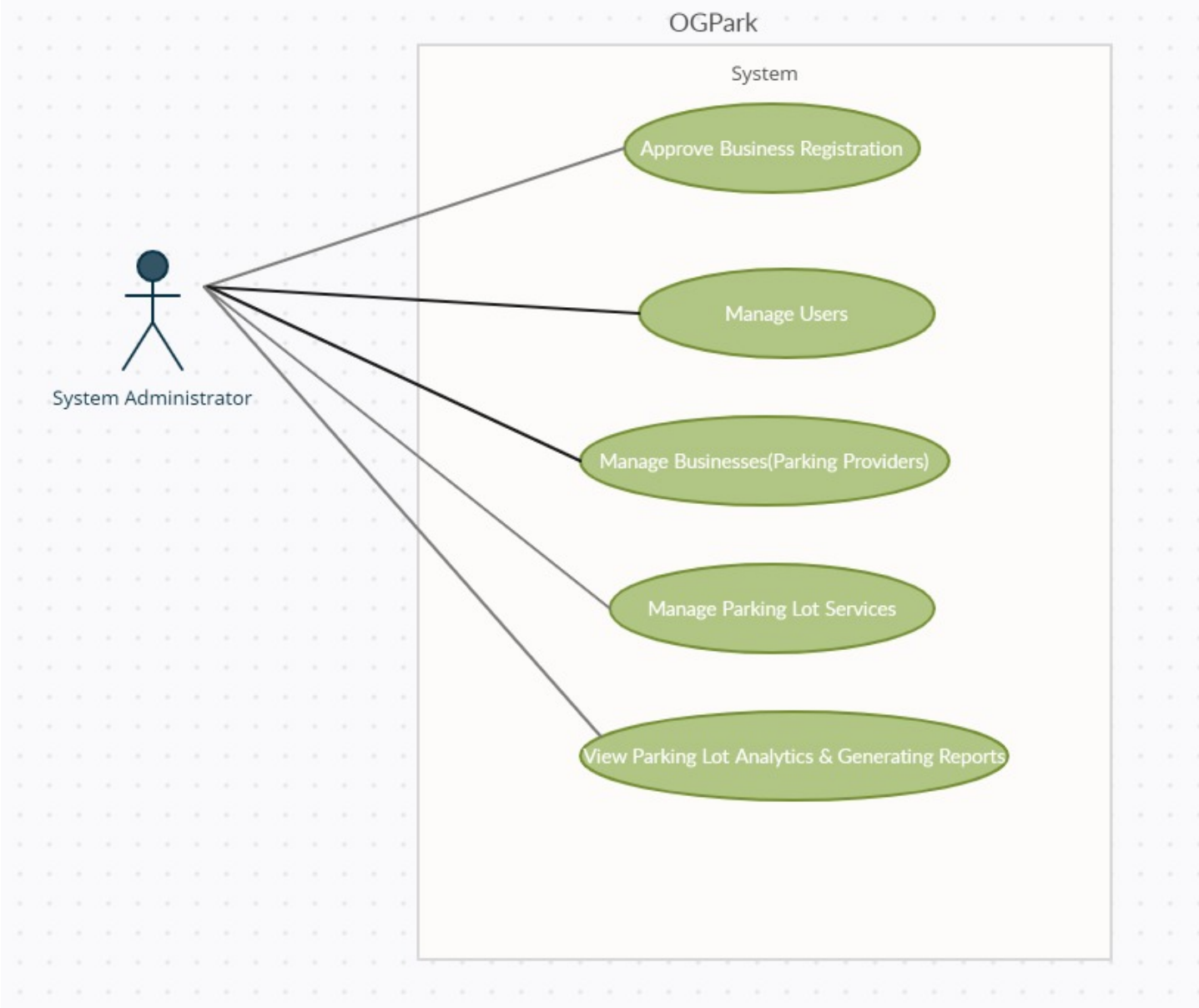


Figure 4.4 System Administrators

i. Approve Business Registration:

The system administrator receives a request from a new parking lot to join the system. The system administrator verifies that the parking lot meets the requirements for joining the system and approves or rejects the registration request accordingly.

ii. Manage User Accounts:

The system administrator can manage user accounts, including adding new users, deleting accounts, and resetting passwords.

iii. Manage Parking Providers:

This use case allows system administrators to manage businesses that provide parking services through the car parking system. The system administrator can add, modify, or delete businesses from the system

iv. Manage Parking Lot Services:

The system administrator can manage parking lot services, including updating the list of services offered, managing service providers, and ensuring that services are being offered according to established guidelines.

v. Generate Reports:

This use case enables the system administrator to generate reports on various aspects of the car parking system, such as parking lot usage, revenue, and customer satisfaction. The administrator can use these reports to make data-driven decisions and improve the overall performance of the car parking system.

4.3 Feasibility Study

Feasibility study measures the practicality of the proposed system into adoption. It assesses whether the book counterfeit recognition system is a viable project that can be used by the publishing industry. The following are the factors that determine the project feasibility;

4.3.1 Operational Feasibility

Operational feasibility is an important aspect to consider for the successful implementation and adoption of the proposed car parking system. The system is designed to provide a seamless and efficient parking experience for users, and therefore, its operational feasibility is critical.

The operational feasibility of the car parking system can be evaluated based on various factors. Firstly, the system should be able to handle a large volume of users and parking transactions without any glitches or delays. This can be achieved by ensuring that the system has adequate processing power, storage capacity, and network bandwidth to handle user requests and transactions.

Secondly, the system should be easy to use and user-friendly, both for drivers and parking lot operators. The user interface should be intuitive, and the system should provide clear instructions and guidance to users. Parking lot operators should also be able to manage the system easily and efficiently, without the need for extensive training or technical knowledge.

Thirdly, the system should be reliable and available round-the-clock, with minimal downtime or maintenance issues. This can be achieved by implementing robust backup and disaster recovery systems, as well as conducting regular maintenance and updates.

Lastly, the system should be cost-effective and provide value for money for both users and parking lot operators. The cost of implementing and operating the system should be reasonable and justifiable, and the system should generate sufficient revenue to cover its costs and provide a return on investment.

In summary, the operational feasibility of the proposed car parking system will depend on its ability to handle a large volume of users and parking transactions, its ease of use and user-friendliness, its reliability and availability, and its cost-effectiveness. If these factors are taken into consideration and addressed appropriately, the system should be operationally feasible and successful in providing a seamless and efficient parking experience for users.

4.3.2 Technical Feasibility

The technical feasibility of a project assesses whether the proposed system can be developed with the available technology and expertise. For the car parking project, there are several technical aspects that need to be considered to ensure that the system can be developed and implemented successfully.

Firstly, the project requires the use of sensors and cameras to detect the availability of parking spaces and monitor the entry and exit of vehicles. These sensors and cameras need to be integrated with the backend software that manages the parking system. The system also needs to be able to communicate with the payment gateway to process payments for parking. Therefore, the project will require expertise in hardware development, software development, and network communication.

Secondly, the project needs to ensure that the system is scalable and can handle a large volume of users and parking spaces. The system should be able to process requests from multiple users simultaneously and update parking space availability in real-time. Therefore, the project will require expertise in designing and developing high-performance systems that can handle heavy loads.

Thirdly, the project needs to ensure that the system is secure and can protect sensitive user data. The system should be able to prevent unauthorized access to the parking management system, payment gateway, and user data. Therefore, the project will require expertise in cybersecurity and encryption.

Fourthly, the project needs to ensure that the system is compatible with various types of vehicles and parking areas. The sensors and cameras used in the system need to be able to detect different types of vehicles, including cars, trucks, and motorcycles, and provide accurate information on parking availability. The system also needs to be able to work with different types of parking areas, including indoor and outdoor parking lots, and be able to adapt to different parking layouts. Therefore, the project will require expertise in system integration and compatibility testing.

Finally, the project needs to ensure that the system is reliable and can provide

continuous service to users. The system should be able to detect and resolve errors and system failures quickly to minimize downtime. Therefore, the project will require expertise in system monitoring, maintenance, and support.

In conclusion, the technical feasibility of the car parking project will require expertise in hardware development, software development, network communication, system integration, cybersecurity, encryption, compatibility testing, system monitoring, maintenance, and support. The project needs to ensure that the system is scalable, secure, compatible, reliable, and can handle a large volume of users and parking spaces.

4.3.3 Schedule Feasibility

Schedule feasibility is an important aspect of any project, including the development of a car parking system. It refers to the project's ability to be completed within the allocated time frame, taking into account any constraints that may affect the project timeline. In order to determine the schedule feasibility of the car parking project, various factors need to be considered.

Firstly, it is important to identify the project timeline, including the start and end dates, and any milestones or deliverables that need to be met along the way. This timeline should take into account any external factors that may impact the project, such as seasonal changes, weather conditions, and holidays.

Once the project timeline has been established, it is important to assess the availability of resources, including equipment, and materials. Next, it is important to break down the project into smaller, more manageable tasks or work packages. Each task should be assigned to specific team members, and a timeline should be established for each task. This helps to ensure that the project stays on track and that each task is completed within the allocated time frame.

Regular monitoring and reporting of project progress is also essential to ensure schedule feasibility. This involves tracking the completion of each task against the established timeline, and identifying any potential delays or issues. This information can then be used to adjust the project timeline as necessary, to ensure that the project stays on track.

It is also important to consider contingency plans in case of unforeseen events that may impact the project timeline. This may include having backup personnel, equipment or materials, or having a plan in place to handle unexpected delays.

In summary, schedule feasibility is a critical aspect of the car parking project. By carefully assessing the project timeline, resource availability, and breaking down the project into manageable tasks, the project team can ensure that the project stays on track and is completed within the allocated time frame. Regular monitoring and reporting of progress, and contingency planning, are also essential to ensure successful completion of the project.

4.3.4 Economic Feasibility

Economic feasibility is an important aspect to consider when developing any project, including a car parking system. Economic feasibility analysis involves evaluating whether the project is financially viable and sustainable in the long-term. This includes examining the project's costs and benefits, as well as conducting a cost-benefit analysis to determine whether the benefits of the project outweigh its costs.

One of the main costs associated with developing a car parking system is the initial investment in equipment and technology. This includes the cost of sensors, cameras, software, and other hardware necessary for the system to function. Additionally, there may be ongoing costs associated with maintenance, upgrades, and repairs to the system. It is important to consider these costs and ensure that the project's budget can accommodate them.

On the other hand, there are potential benefits associated with the implementation of a car parking system. One of the most significant benefits is the potential increase in revenue generated by the parking lot. A well-designed and efficient car parking system can attract more customers, resulting in higher revenues for the parking lot owner. Additionally, a car parking system can help reduce operating costs, such as labor costs associated with managing the parking lot.

To determine whether the benefits of the car parking system outweigh its costs, a cost-benefit analysis should be conducted. This involves comparing the total costs of the project to its expected benefits over time. Benefits may include increased

revenue, cost savings, and improved customer satisfaction. If the benefits outweigh the costs, then the project is considered economically feasible.

Other factors to consider when evaluating the economic feasibility of a car parking system include the expected return on investment (ROI) and the payback period. ROI is a measure of the profitability of the project, while the payback period is the amount of time it takes for the project to recoup its initial costs. A shorter payback period is generally more desirable, as it means the project will begin generating a positive cash flow sooner.

It is important to note that economic feasibility analysis is not a one-time process. The economic feasibility of the project should be regularly reevaluated throughout the project's life cycle to ensure that it remains financially viable and sustainable in the long-term. This may involve making adjustments to the project's budget, scope, or schedule to optimize its benefits and minimize its costs.

In summary, the economic feasibility of a car parking system involves evaluating the project's costs and benefits, conducting a cost-benefit analysis, and considering factors such as ROI and payback period. By carefully considering these factors, project stakeholders can ensure that the project is financially viable and sustainable in the long-term.

4.4 Requirements Elicitation

Requirements elicitation is the process of identifying, analyzing, and documenting the needs and requirements of a system or software application. This involves gathering input from stakeholders, users, and other sources to define the features, functions, and performance criteria that the system must meet in order to be successful.

Requirements elicitation typically involves a range of techniques, including interviews, surveys, focus groups, observation, and data analysis. The goal of the process is to gather as much information as possible about the problem or opportunity that the system is meant to address, and to translate this information into specific, actionable requirements.

Effective requirements elicitation is critical for the success of any software or

system development project, as it helps to ensure that the final product meets the needs of its users and stakeholders, and delivers the expected value and functionality. It also helps to minimize the risk of project failure or scope creep, by providing a clear and detailed roadmap for development, testing, and deployment.

4.4.1 Methods

Various methods were used to gather requirements for the car parking project, which included investigative research, study of literature such as reports, articles, and documents on car parking systems, as well as informal interviews with drivers(mostly online taxi drivers) and users in Nairobi. These methods were used to gain insights into the current state of car parking in Nairobi, identify the main challenges faced by car owners and users, and explore potential solutions to improve the situation. Additionally, online surveys and questionnaires conducted by organizations like KARA(Kenya Alliance of Resident Associations), ITDP(Institute for Transportation and Development Policy) and Nairobi City County helped to gather data from a larger pool of respondents and obtain more quantitative insights into the attitudes and opinions of car owners and users in Nairobi.

4.4.2 Research Findings

The findings derived from various documents, interviews and investigative documentaries include;

1. Access to parking spaces at residences:

According to the survey reports, many residents in Nairobi indicated that parking spaces were not readily available or were inadequate. It is likely that a significant proportion of respondents did not have access to designated parking spaces at their residence, and some may have expressed concerns about the security of their vehicles.

2. Access to parking spaces in public places:

The survey reports found that accessing public parking spaces in Nairobi was challenging for many respondents, with many expressing frustration at the difficulty of locating and accessing available parking spaces.

3. Regularity and Duration in which people park in Nairobi city center:

The KARA Parking Survey report found that many residents in Nairobi park their vehicles in the city center, and that parking duration varies significantly depending on the purpose of the visit. The survey revealed that a significant proportion of respondents park their vehicles in the city center on a daily basis, with some indicating that they visit the city center multiple times a day.

4. Current Parking fee structure in Nairobi City Center:

According to the survey reports, many respondents viewed the current parking fee structure in Nairobi as unreasonable, with some expressing a desire for more affordable and flexible pricing options.

5. Challenges faced when accessing parking in Nairobi City Center:

The survey reports identified several main challenges faced by residents when accessing or using parking facilities in Nairobi, including the lack of available parking spaces, the high cost of parking, and compliance issues such as illegal parking or violations of parking regulations.

6. People's awareness of the existing parking policies and regulations in Nairobi:

The KARA Parking Survey report found that a significant proportion of respondents were not aware of the existing parking policies and regulations in Nairobi. Only 41.7% of respondents reported being aware of the policies and regulations, indicating a need for increased awareness and education efforts around parking rules and regulations in the city.

7. Thoughts on potential solutions to improve parking in Nairobi:

According to the survey report, many respondents expressed interest in innovative solutions to improve parking in Nairobi, such as the use of technology to better manage parking demand or the development of shared parking facilities. There was also support for the introduction of new parking policies and regulations to improve compliance and ensure fair access to parking.

4.5 System Requirements

4.5.1 Functional Requirements

They define what the system should accomplish. They include;

i. User registration and authentication: The system should allow users to register and authenticate themselves to access the parking lot.

ii. Vehicle detection and tracking: The system should have sensors or cameras to detect and track the vehicles entering and exiting the parking lot.

iii. Space availability detection: The system should be able to detect the availability of parking spaces in real-time.

4.5.2 Non-functional Requirements

Non-functional requirements are the quality attributes or characteristics that the car parking system should possess in order to meet the needs of its users. Here are my non-functional requirements for a car parking system:

1. Reliability: The system should be reliable, meaning it should be available and operational 24/7 with minimum downtime.
2. Performance: The system should perform efficiently, meaning it should be able to process parking transactions quickly and accurately, and handle a high volume of vehicles during peak times.
3. Security: The system should be secure, meaning it should have measures in place to prevent unauthorized access and protect the personal and financial information of its users.
4. Usability: The system should be user-friendly, meaning it should be easy to use and navigate for both the parking lot users and the administrators.
5. Accessibility: The system should be accessible, meaning it should be usable by people with different abilities, including those with visual or hearing impairments.
6. Scalability: The system should be scalable, meaning it should be able to expand or contract as the parking lot grows or shrinks.
7. Maintainability: The system should be maintainable, meaning it should be easy to repair, update, and upgrade as needed, without causing any disruption to the parking lot operations.
8. Compatibility: The system should be compatible with different hardware and

software components, and integrate seamlessly with other systems, such as payment gateways, traffic management systems, and security systems.

4.5.3 Constraints

In system development projects, constraints refer to limitations or restrictions that impact the design and implementation of the system. Constraints can take many different forms, such as technical limitations, budgetary limitations, time constraints, resource constraints, or legal and regulatory constraints.

Constraints play an important role in shaping the overall design and functionality of the system, as they help to define what is feasible and what is not. For example, if the project has a tight budget, this may constrain the amount of resources and features that can be included in the final system. Similarly, if there are legal or regulatory constraints, these may dictate specific requirements that must be met by the system, such as data privacy or security regulations.

Understanding and managing constraints is an important part of system development, as it can impact the success and viability of the project. By carefully analyzing the constraints and finding creative solutions to work within them, project teams can develop systems that are both effective and feasible within the given constraints.

Some of the constraints include:

i. Physical constraints: The size and layout of the parking lot may limit the number and placement of IoT sensors and other hardware components, as well as the number and size of parking spaces.

ii. Technology constraints: The car parking system must be designed to work with specific types of hardware, such as the Arduino microcontroller, and may require specific programming languages or development frameworks.

iii. Budget constraints: The cost of hardware components, software development, and ongoing maintenance must be carefully managed to ensure that the project remains financially viable.

iv. Time constraints: The car parking system must be developed and deployed within a specific timeframe, and may require ongoing maintenance and updates to ensure its continued functionality and effectiveness.

v. Regulatory constraints: The car parking system must comply with local, state, and

federal regulations, such as those related to safety, security, and data privacy.

vi. User constraints: The car parking system must be designed with the needs and expectations of users in mind, and must be user-friendly, accessible, and easy to use.

vii. Resource constraints: The car parking system may require specific resources, such as skilled developers or technical support staff, that may be limited or difficult to obtain.

viii. Environmental constraints: The car parking system may need to be designed with environmental factors in mind, such as energy efficiency or sustainability.

CHAPTER FIVE: SYSTEM DESIGN

5.1 Introduction

This chapter describes the physical designs that will meet the specifications described in the system requirements. The tasks will include user interface design, data design and system architecture.

5.2 System Architecture

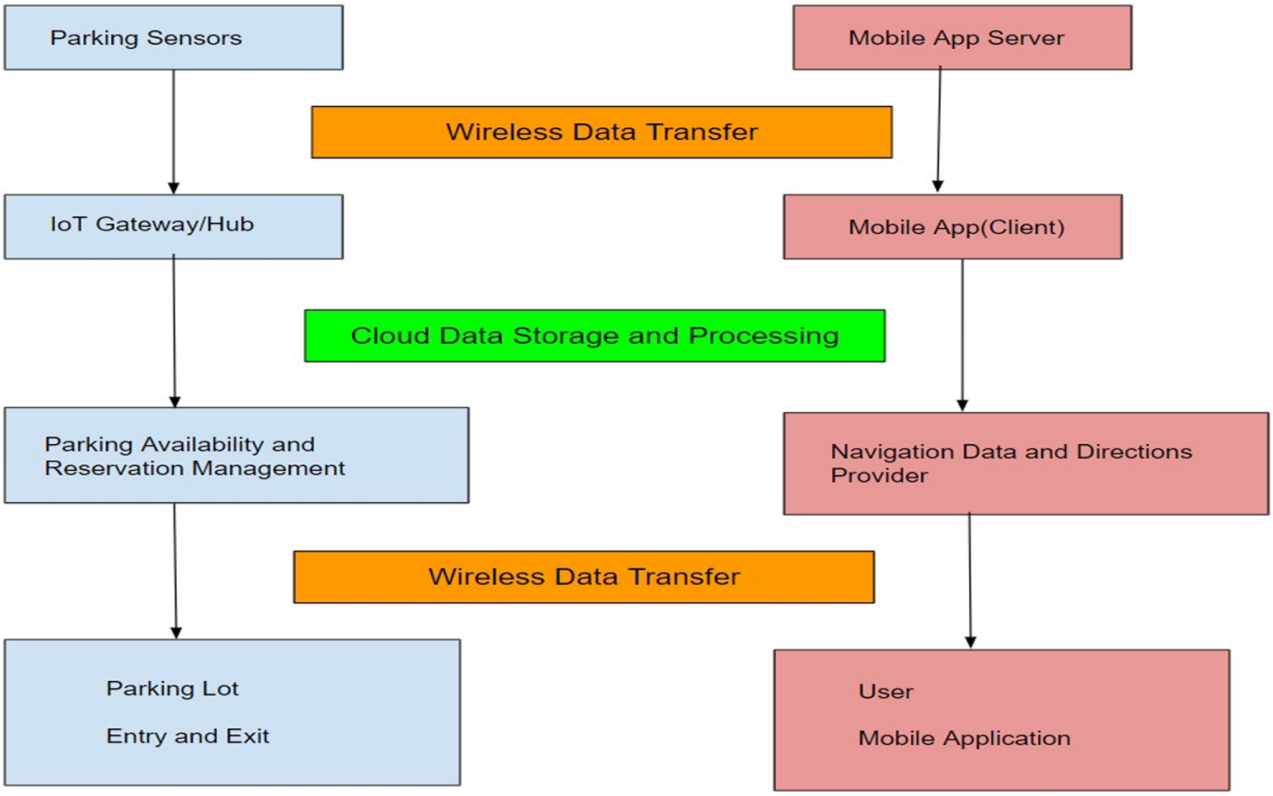


Figure 5.1 System Architecture

i. Parking Sensors: IoT devices, such as ultrasonic sensors, are installed in each parking spot to detect whether it is occupied or vacant. The sensors detect the presence of a vehicle and transmit this information wirelessly to an IoT gateway.

ii. IoT Gateway: The IoT gateway receives the data from the parking sensors and forwards it to the cloud. The gateway can use various communication protocols such as Wi-Fi, Bluetooth, or cellular networks to transmit the data. The system

would have an Arduino with several IR and ultrasonic sensors, we can use an IoT gateway that is compatible with Arduino boards. One popular option is the Arduino IoT Cloud, which is a cloud-based platform designed specifically for Arduino boards.

The Arduino IoT Cloud provides a simple and secure way to connect your Arduino board to the cloud and manage your IoT projects. It supports a range of communication protocols, including Wi-Fi, Ethernet, and cellular, and includes a built-in IoT gateway that can communicate with your Arduino board and transmit data to the cloud.

Using the Arduino IoT Cloud, you can easily configure your sensors and create a dashboard to view real-time data from your sensors. You can also set up rules and triggers to automate actions based on sensor data, such as sending notifications when a parking spot becomes available or triggering an alarm when a car is parked in a reserved spot.

In summary, the Arduino IoT Cloud is a good option for your car parking project, since it is specifically designed for Arduino boards and includes a built-in IoT gateway that can communicate with your sensors and transmit data to the cloud. However, it's important to evaluate different options and choose the one that best fits your specific requirements and budget.

iii. Cloud Data Storage and Processing: The cloud-based platform receives the data from the IoT gateway and stores it in a database. The data can then be processed to determine parking spot availability in real-time. This processing can involve complex algorithms to predict future parking demand or optimize the use of parking spots.

iv. Parking Availability and Reservation Management: The parking availability and reservation management software uses the data from the cloud to manage parking spot availability and reservation requests. Users can check the availability of parking spots in real-time through a mobile application, which sends requests to the cloud to obtain up-to-date information.

v. Navigation Data and Directions Provider: A navigation data and directions provider, such as Google Maps or Apple Maps, provides map data and directions to the user. The navigation data can include information on the parking lot's location, available parking spots, and the best route to take to get there.

vi. Mobile App Server: The mobile app server manages the communication between the mobile app and the cloud data storage and processing. The server receives requests from the mobile app for parking availability information, and sends this information back to the app.

vii. Mobile App (Client): The mobile application allows users to view parking spot availability, reserve parking spots, and receive directions to the parking lot. Users can reserve a parking spot through the app, and receive a confirmation of their reservation.

This system provides a seamless experience for users who want to find a parking spot. The system uses IoT devices to detect parking spot availability in real-time, and a cloud-based platform to process and manage parking spot reservations. Users can access this information through a mobile application and receive directions to the parking lot. The system provides a convenient and efficient way to manage parking, saving time and reducing frustration for drivers.

5.3 System Flowchart

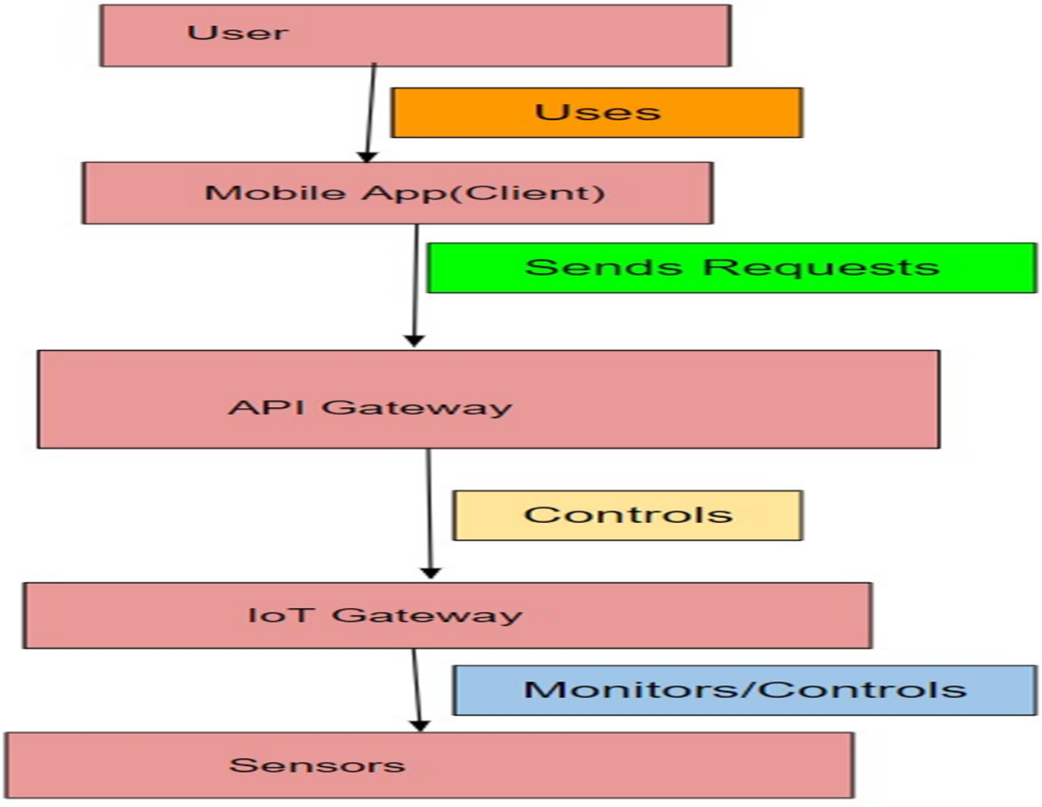


Figure 5.2 System flowchart

The system consists of four actors: User, Mobile App, API Gateway, and IoT Gateway.

The User is the person who uses the Mobile App to find and reserve a parking spot.

The Mobile App is responsible for presenting parking spot availability to the user, allowing the user to reserve a parking spot, and directing the user to the reserved parking spot.

The API Gateway is responsible for controlling the communication between the Mobile App and the IoT Gateway. It receives requests from the Mobile App and forwards them to the IoT Gateway for processing. It also receives data from the IoT Gateway and sends it back to the Mobile App for display.

The IoT Gateway is responsible for controlling and monitoring the sensors that are placed in the parking spots. The sensors are used to detect whether a parking spot is occupied or vacant. The IoT Gateway sends the sensor data to the cloud for storage and analysis.

5.4 Hardware System Flowchart

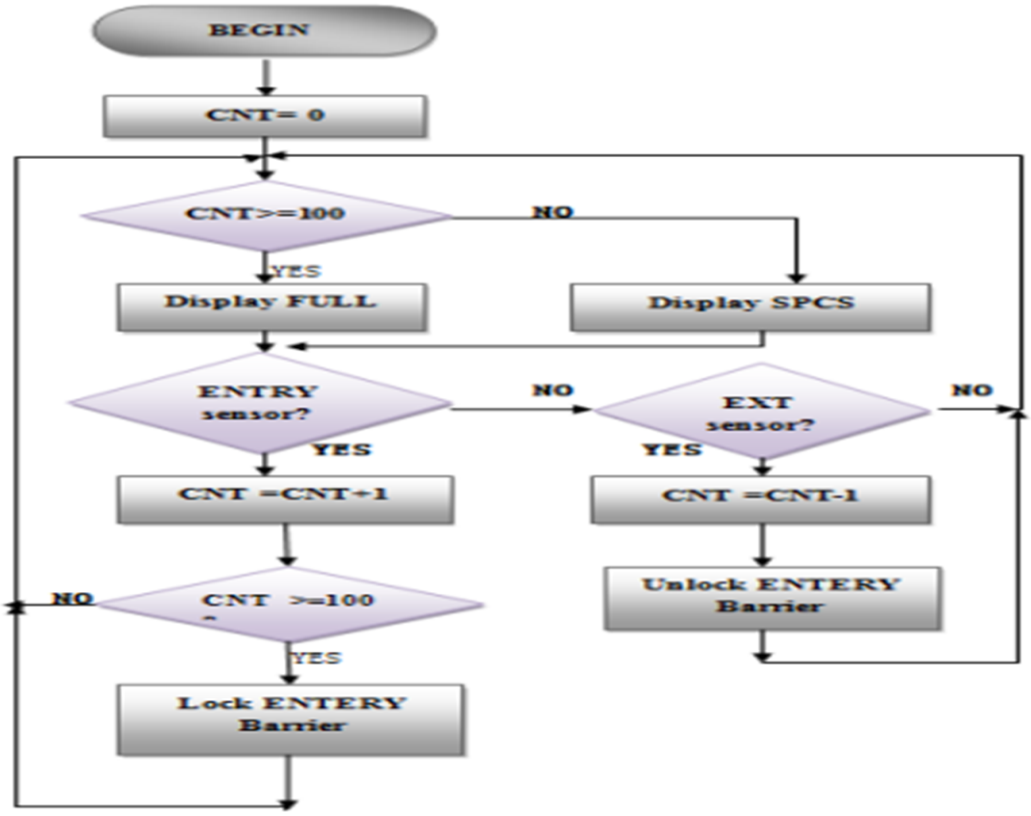


Figure 5.3 Hardware System Flowchart

The project "Car Parking System" demonstrates the concept of an automatic car parking system. This system automatically detects the entry and exit of cars through the gate and monitors the occupancy of parking slots, displaying the number of cars in the parking lot.

A microcontroller is deployed to sense the car movements and determine whether there is available parking space, controlling the opening and closing of the gate accordingly. The system consists of three sets of sensors: one installed at the entry gate, another at the exit gate, and additional sensors for each parking slot. When a car approaches the entry gate, the microcontroller receives signals from the entry sensors and checks if there is space for the car to park. Simultaneously, it displays the current number of cars in the parking lot on an LCD screen and opens the gate if there is available space. When a car leaves the parking area, the microcontroller updates the displayed count on the LCD and closes the gate.

The entry and exit of cars are sensed using infrared transmitters and receivers. An infrared transmitter is mounted on one side of the door, while the receiver is placed directly across from it. When a car arrives, it blocks the infrared beam, causing a change in the receiver's output. This change is detected by the microcontroller, which increments the count and opens the gate if there is an empty parking spot. The process for car exit is similar to the entry process.

At the start of the program, an event counter variable, Cnt, is set to zero. The program checks the value of Cnt: if Cnt is greater than or equal to 100, it assumes that the parking lot is full and displays the message "FULL." If Cnt is less than 100, it assumes that there are available spaces and displays the message "SPCS." The program then monitors the ENTRY switch, incrementing Cnt by one when a car enters the parking lot and decrementing Cnt by one when a car leaves.

5.5 Hardware Circuit Diagram

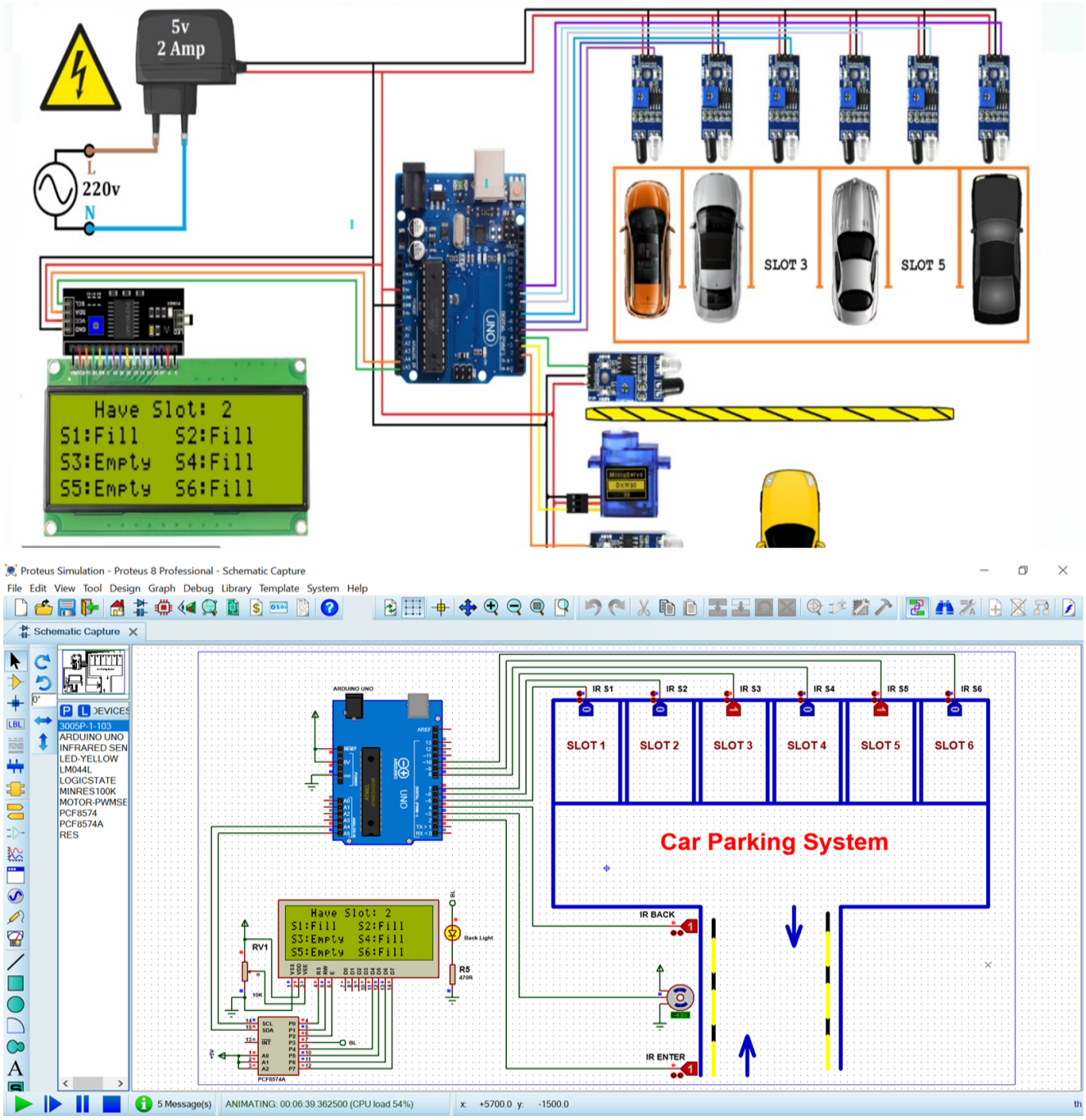


Figure 5.4 Hardware Circuit Diagram

5.6 Entity Relationship Diagram

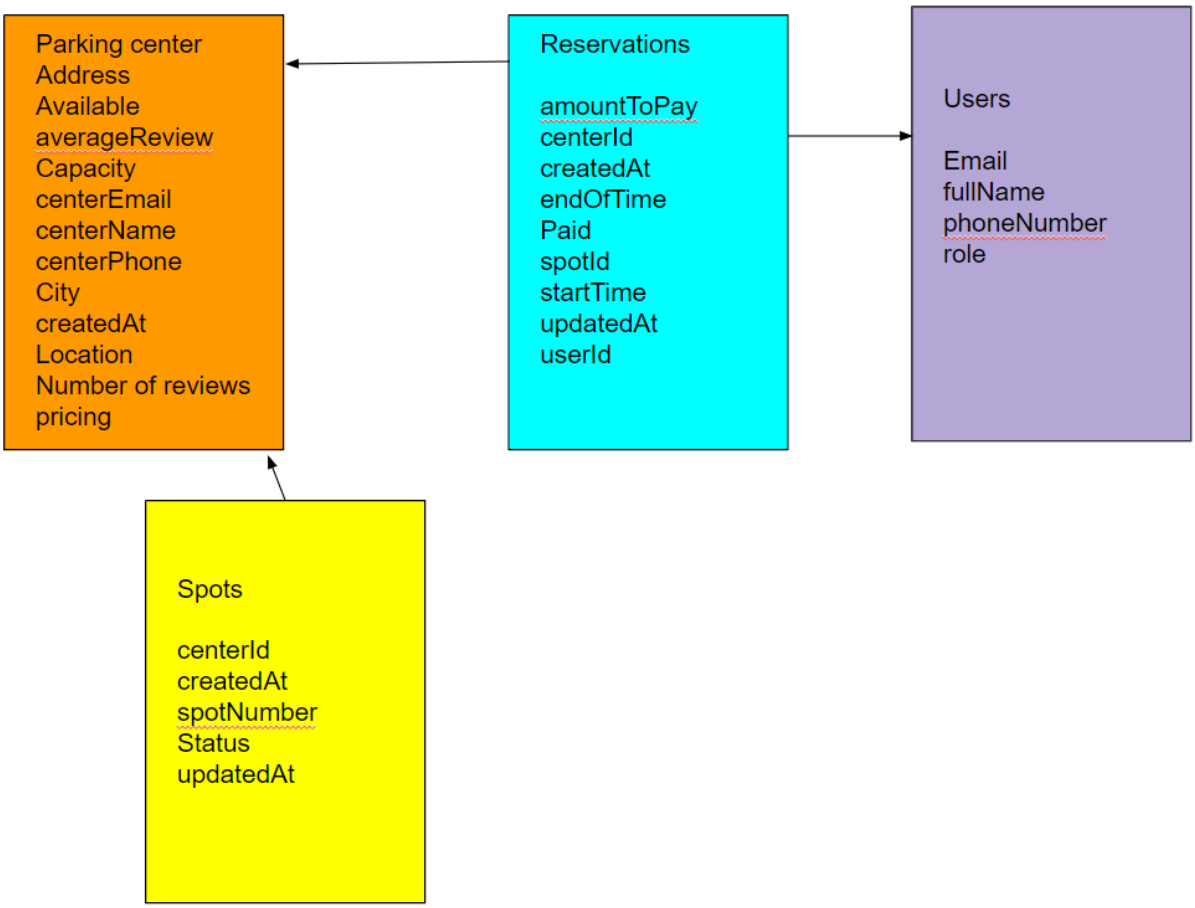


Figure 5.5 Entity Relationship Diagram

i. **Parking Area:**

The Parking Area entity represents a location where cars can be parked. It has the following attributes:

a. parking_area_id: This is a unique identifier for each parking area. It is the primary key for the Parking Area entity.

b. name: This attribute represents the name of the parking area, which can be used to identify it.

c. address: This attribute represents the physical address of the parking area.

d. latitude: This attribute represents the latitude of the parking area's location.

e. longitude: This attribute represents the longitude of the parking area's location.

f. total_spots: This attribute represents the total number of parking spots that are available in the parking area.

ii. **Parking Spot:**

The Parking Spot entity represents an individual parking spot within a Parking Area. It has the following attributes:

a. parking_spot_id: This is a unique identifier for each parking spot. It is the primary key for the Parking Spot entity.

b. status: This attribute represents the current status of the parking spot (e.g., vacant, occupied).

c. is_reserved: This attribute represents whether the parking spot is reserved or not (e.g., true, false).

d. reserved_for: This attribute represents the user who has reserved the parking spot.

iii. Business Owner:

The Business Owner entity represents a registered business that owns one or more Parking Areas. It has the following attributes:

a. business_id: This is a unique identifier for each business owner. It is the primary key for the Business Owner entity.

b. business_name: This attribute represents the name of the business that owns the parking area(s).

c. business_address: This attribute represents the physical address of the business owner.

d. business_contact_no: This attribute represents the contact number of the business owner.

The ERD also shows how the entities are related to each other. Specifically, the Parking Area entity has a one-to-many relationship with the Parking Spot entity (i.e., a single parking area can have multiple parking spots). Additionally, the Business Owner entity has a one-to-many relationship with the Parking Area entity (i.e., a single business owner can own multiple parking areas).

iv. User:

The User entity represents a user of the car parking application. It has the following attributes:

a. user_id: This is a unique identifier for each user. It is the primary key for the User entity.

b. name: This attribute represents the name of the user.

c. email: This attribute represents the email address of the user.

d. phone_number: This attribute represents the phone number of the user.

5.7 User interface design

5.7.1 Landing Page

Figure 5.6 Landing Page

5.7.2 Sign In Page

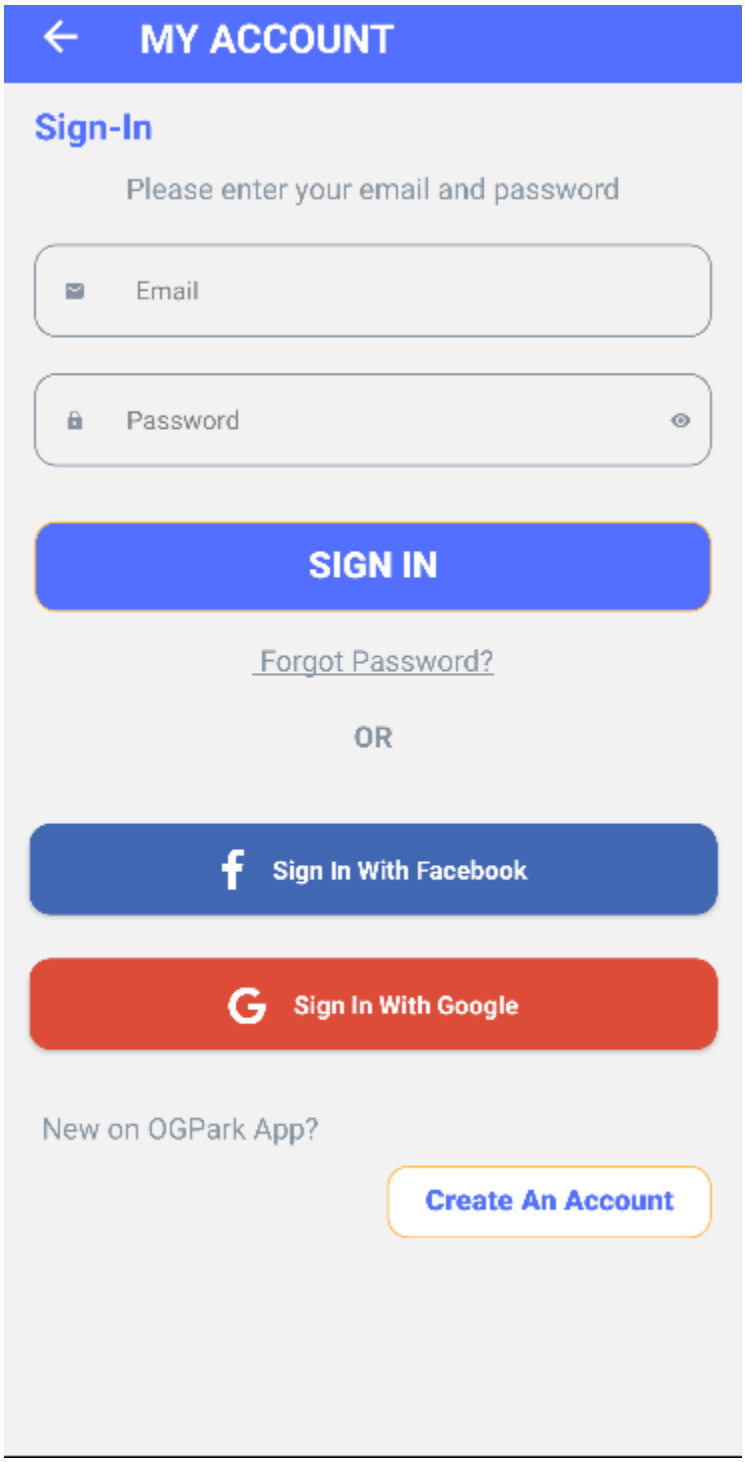


Figure 5.7 Sign in Page

5.7.3 Dashboard Page

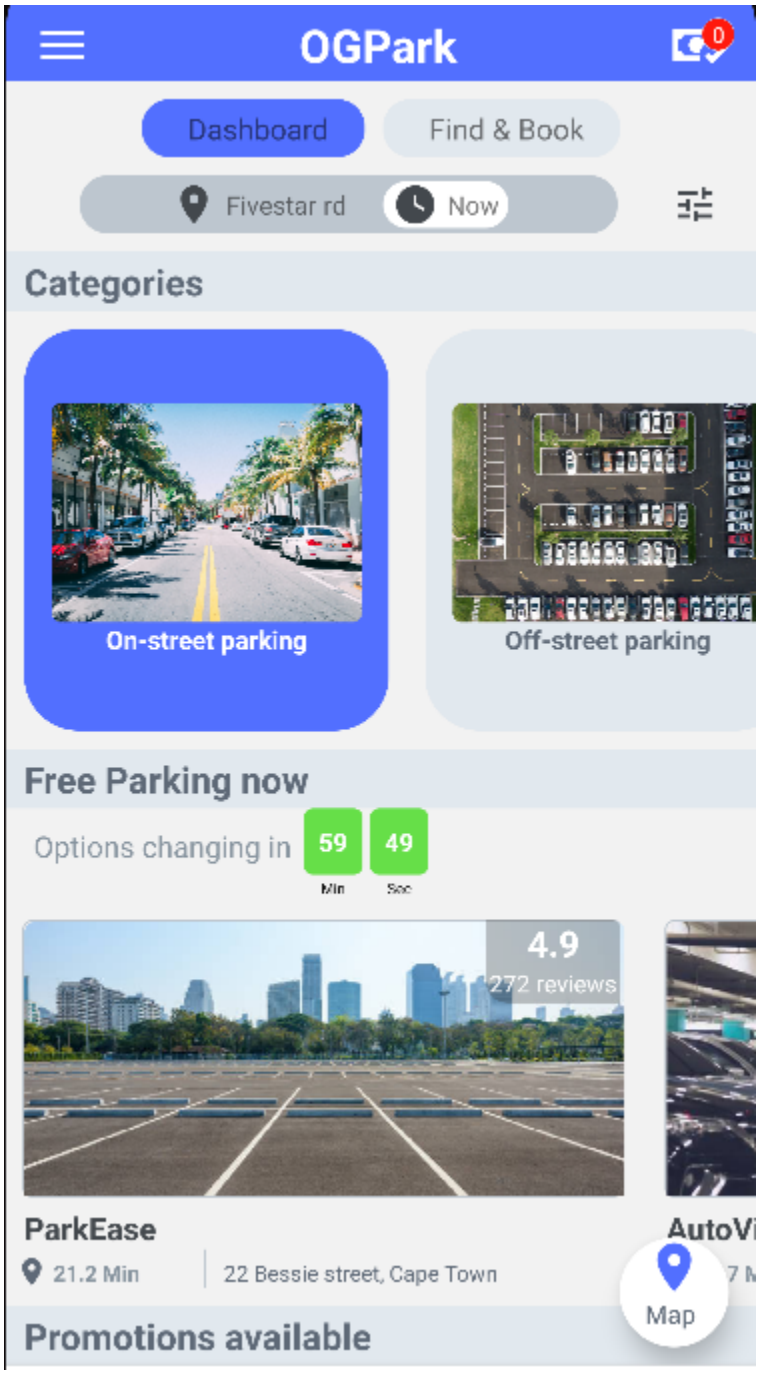


Figure 5.8 Dashboard page

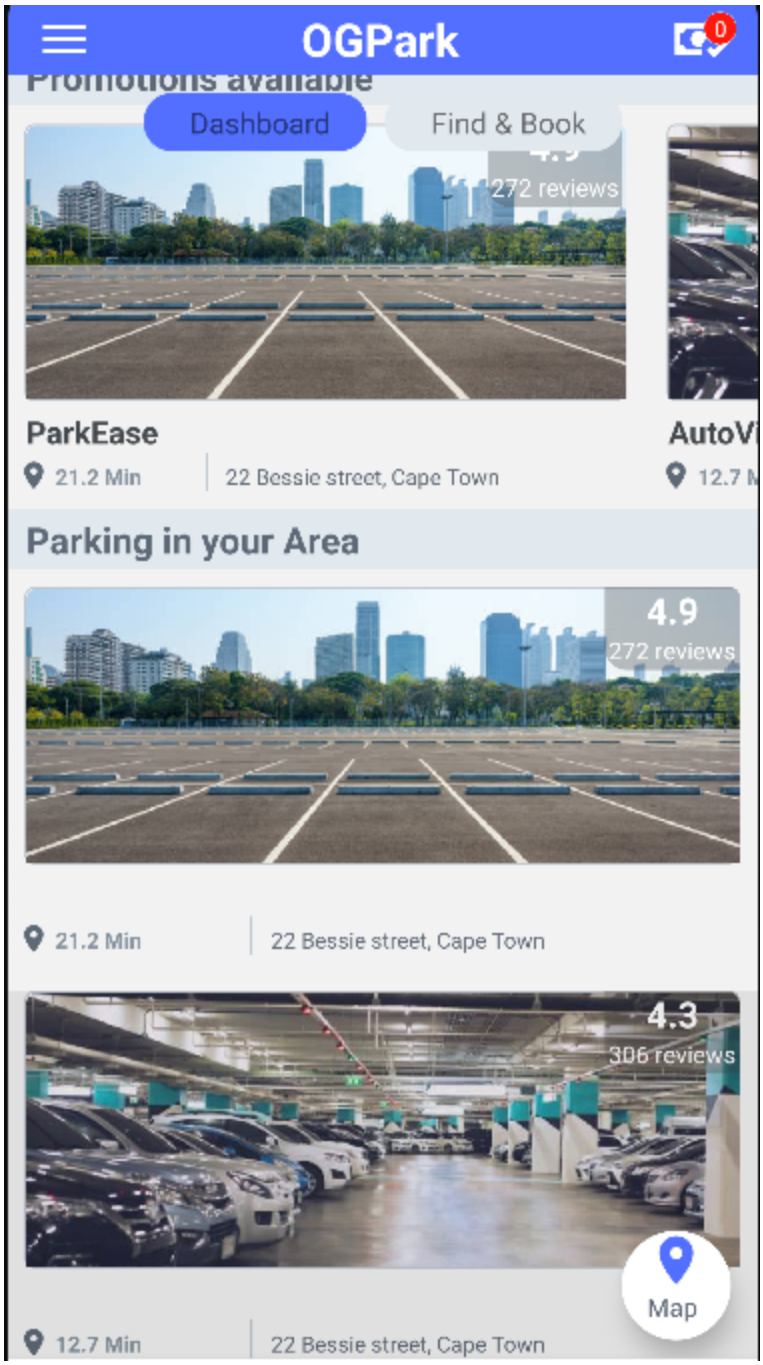


Figure 5.8 Dashboard page

5.3.4 Drawer Navigation

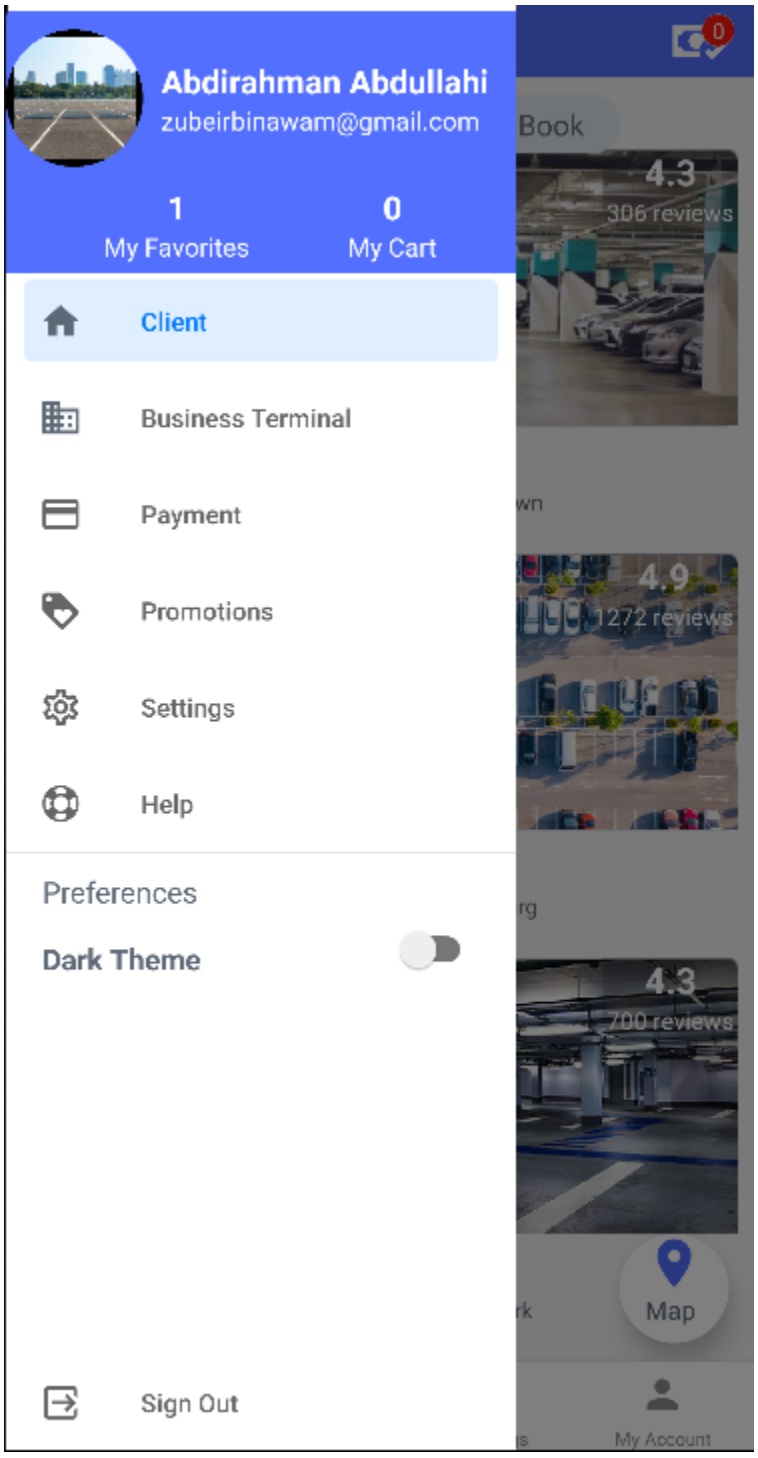


Figure 5.9 Drawer Navigation page

5.3.5 Bottom Navigation

Figure 5.10 Bottom Navigation

5.3.6 Search By Category Page

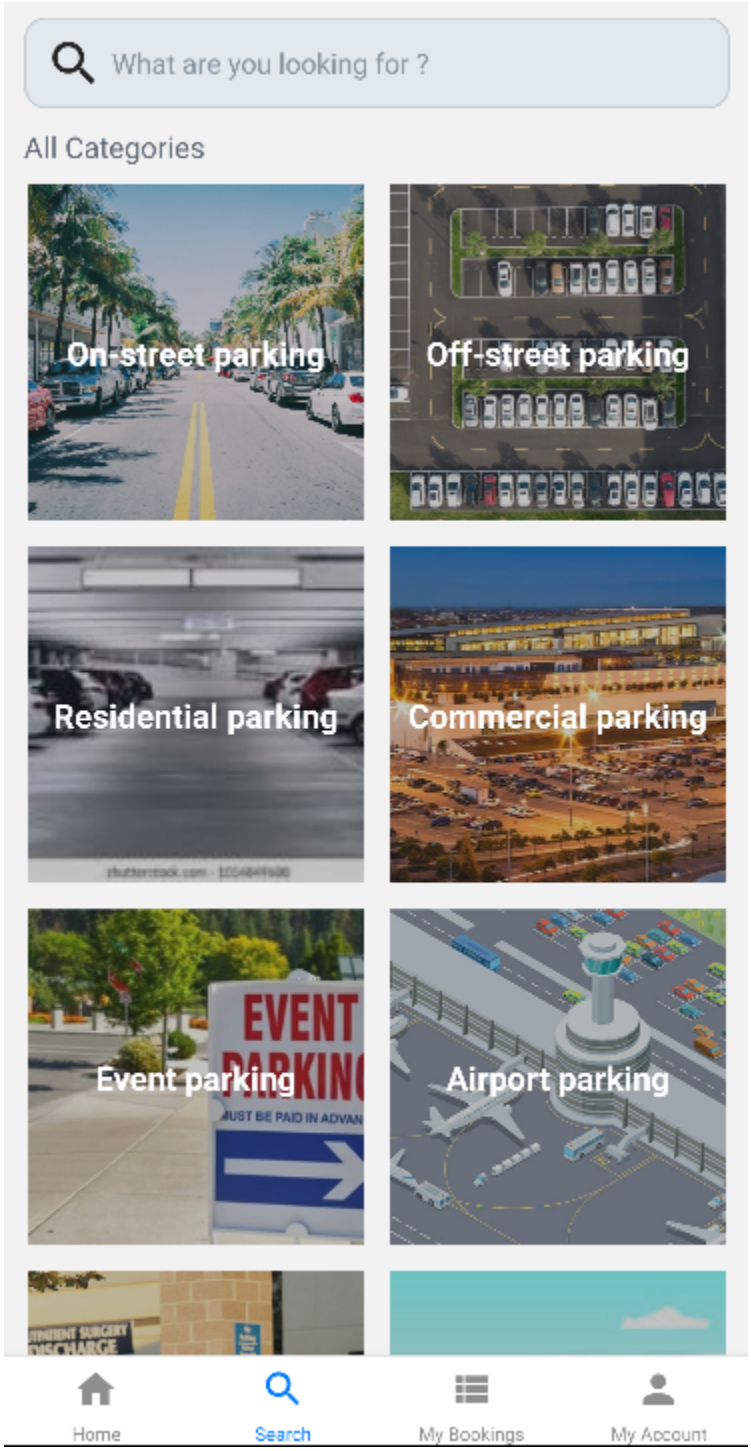


Figure 5.11 Search By Category Page

5.3.7 Search By Category Text Input Page

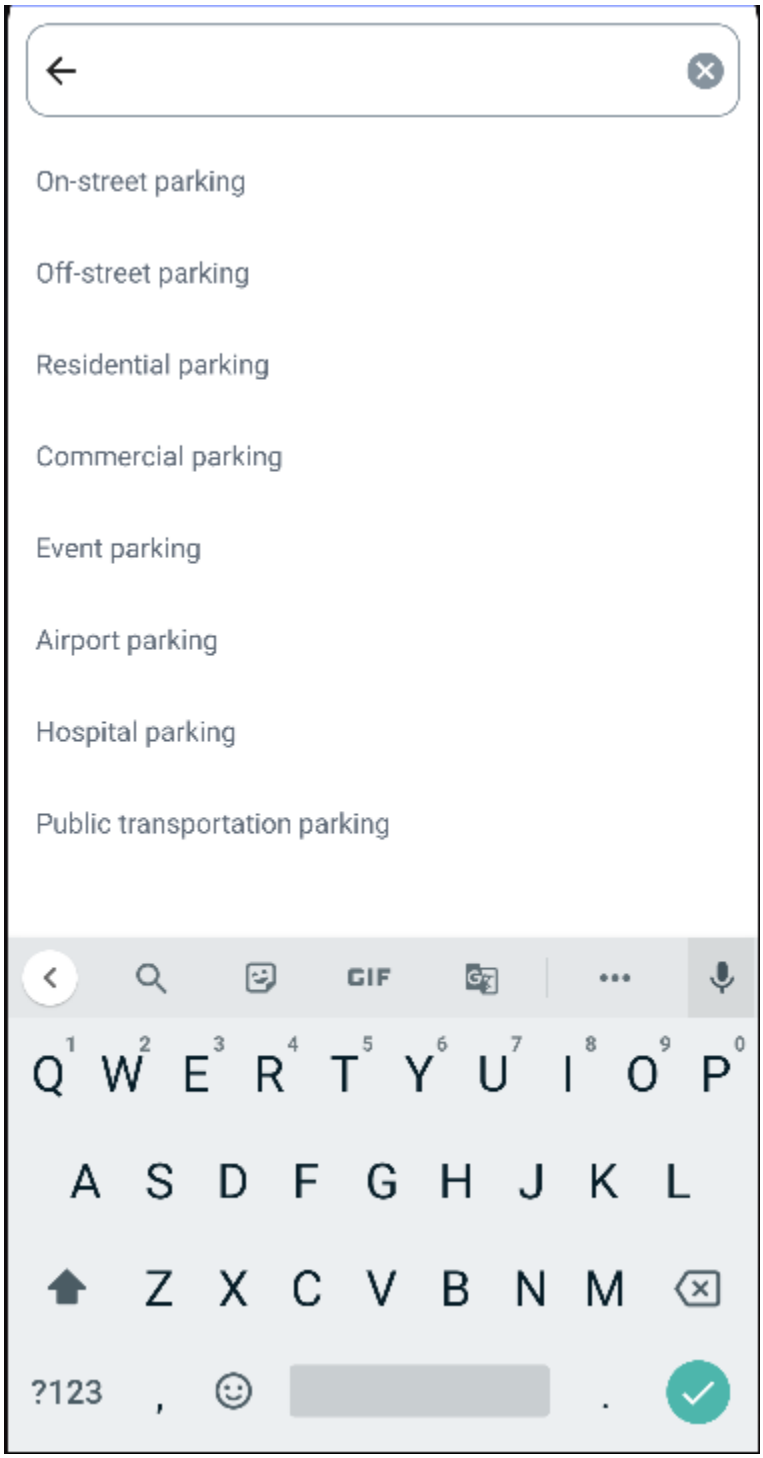


Figure 5.12 Search By Category Text Input Page

5.3.8 Category Results Page

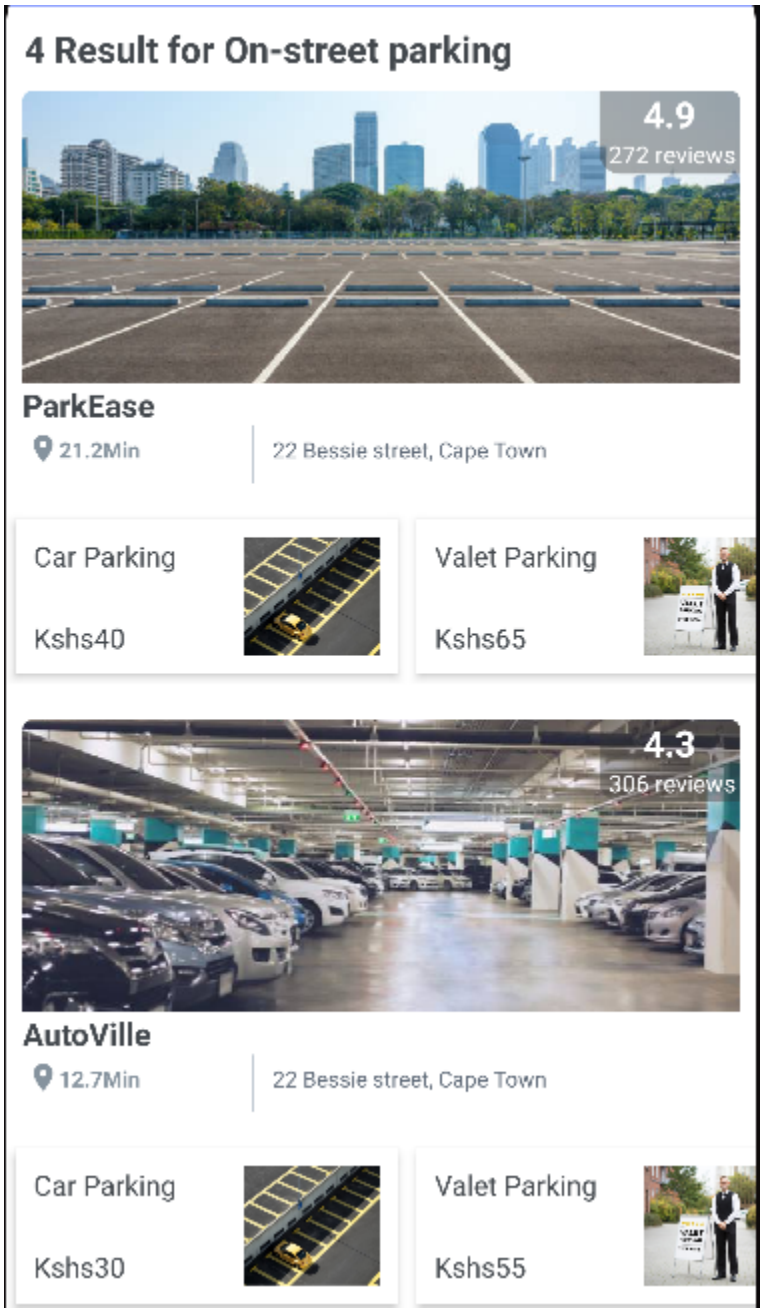


Figure 5.13 Category Results Page

5.3.9 Parking Provider's Home Page

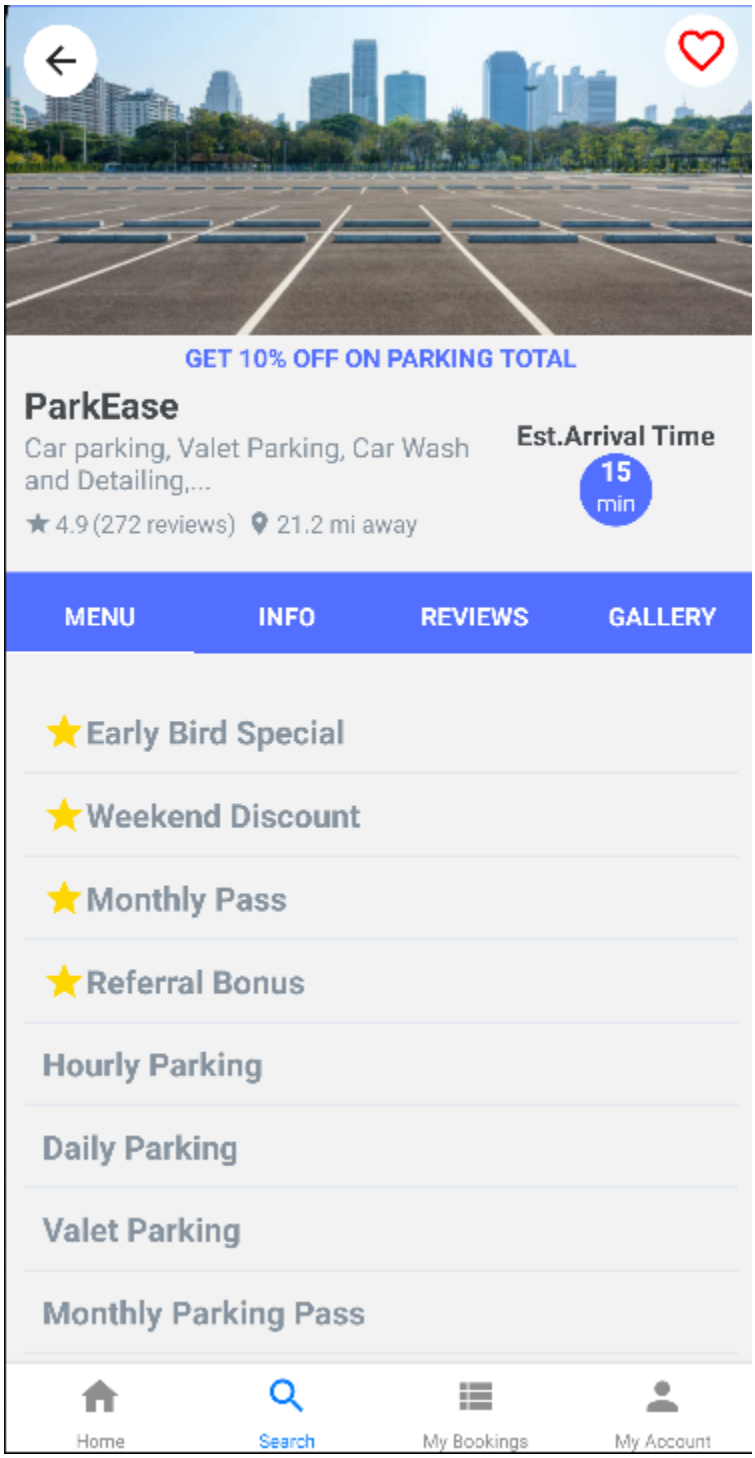


Figure 5.14 Parking Provider's Home Page

5.3.10 Find Parking Page

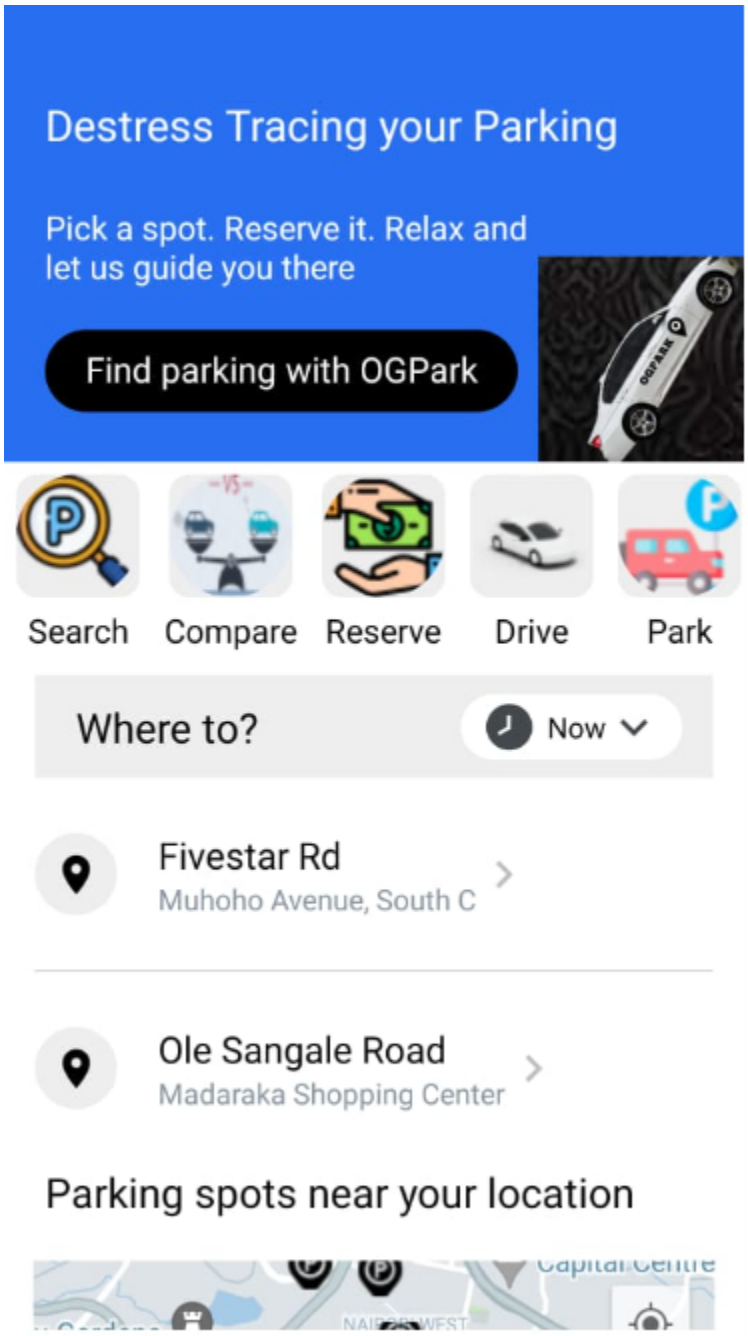
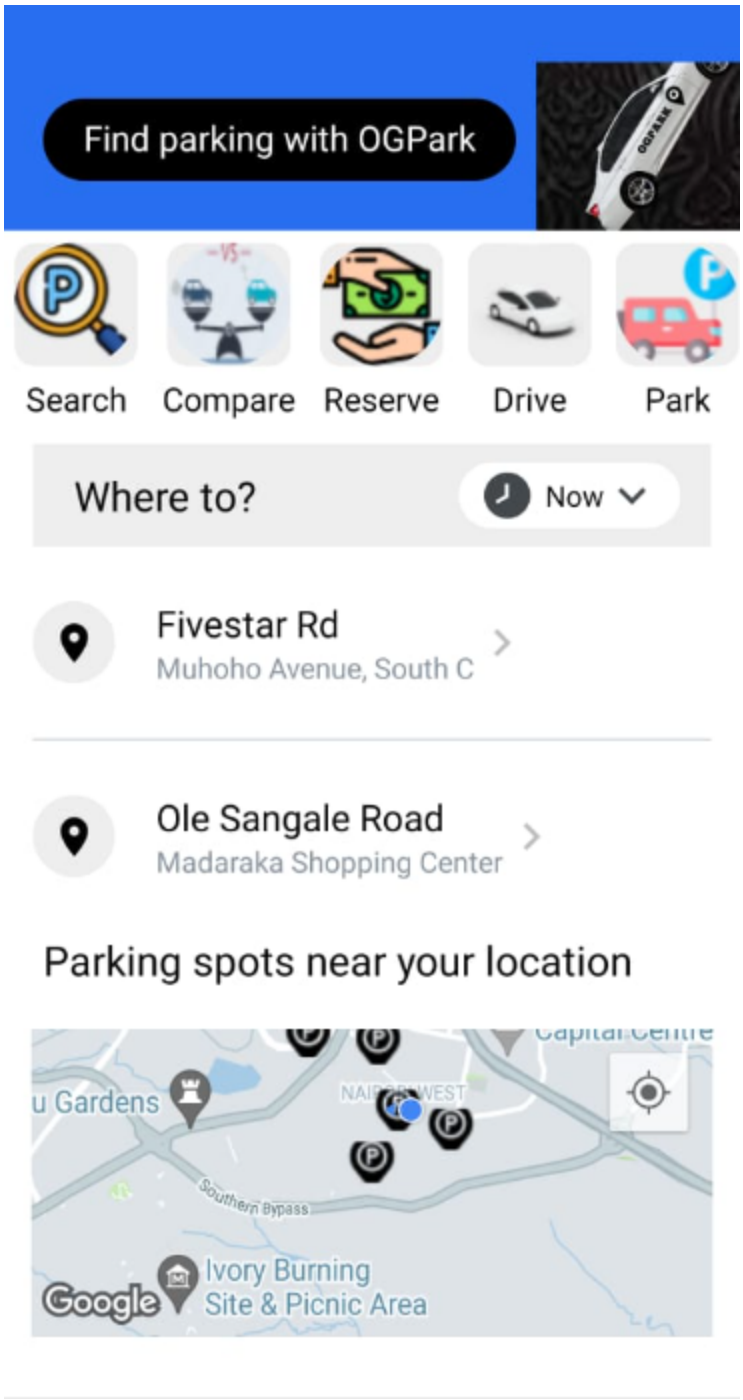


Figure 5.15 Find Parking Page


Find parking with OGPark
Search
Compare
Reserve
Drive
Park
Where to?
Now
Fivestar Rd
Muhoho Avenue, South C
Ole Sangale Road
Madaraka Shopping Center
Parking spots near your location
Southern Bypass
Ivory Burning
Site & Picnic Area
Google

5.3.11 Search For Parking By Location

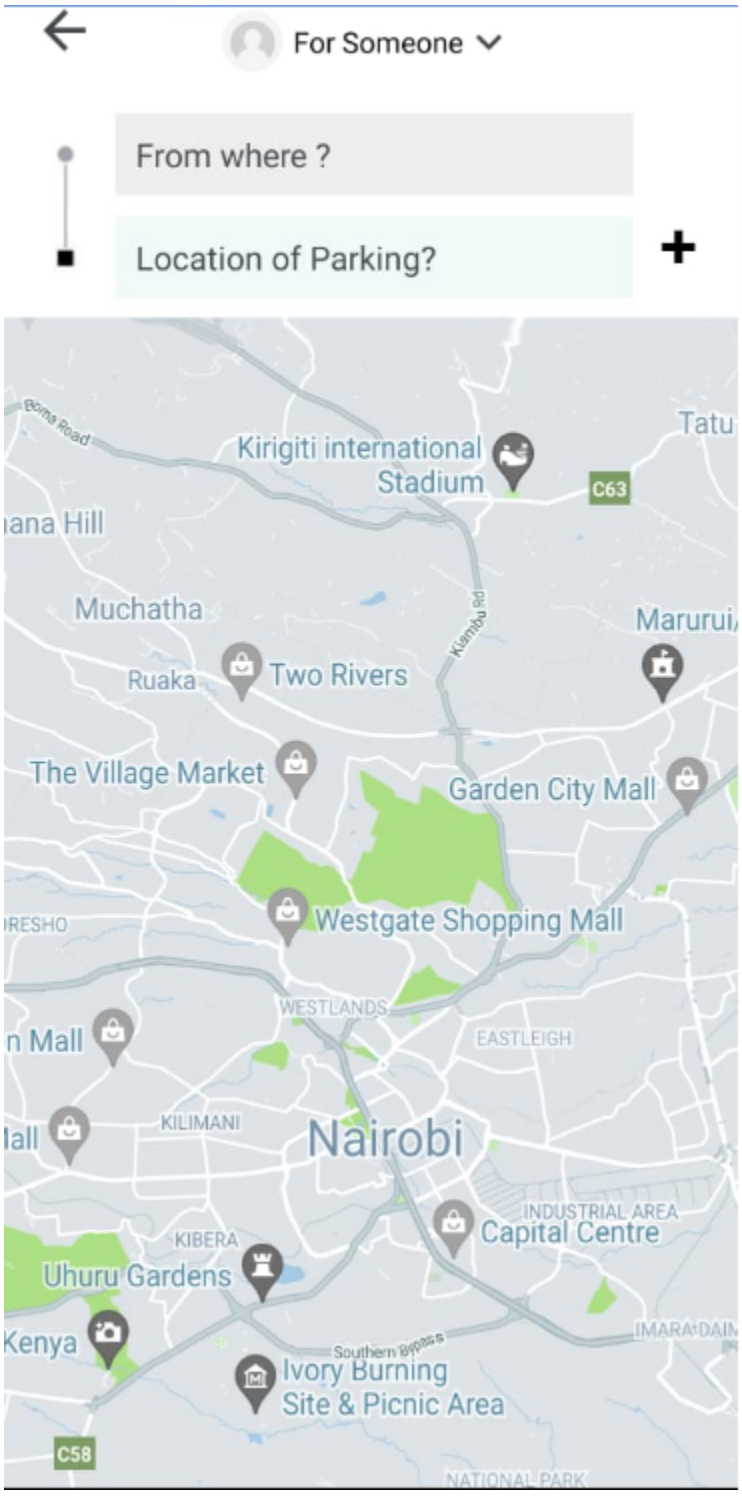


Figure 5.16 Search For Parking By Location

5.3.12 Enter Current Location Page

Figure 5.16 Enter Current Location Page

CHAPTER SIX: SYSTEM IMPLEMENTATION

The car parking project implementation involves various components, including a mobile application, IoT system using Arduino, a cloud-based database using Firestore Firebase, and server-side processing with Node.js. Here is a detailed explanation of each component:

6.1 Mobile User Interface in React Native:

For the mobile application, we developed a user interface using React Native. The application allows users to interact with the car parking system conveniently. We implemented user authentication functionality, enabling users to log in using their email and password or via Google authentication. This ensures secure access to the system and personalizes the user experience.

Using React Native's capabilities, we integrated Firestore Firebase as the database for storing user data and parking information. The mobile application communicates with the database to fetch and update relevant data, such as available parking spaces, specific slot information, user reservations, and additional services like car washing, valet parking, and long-term car storage.

The application also provides features like displaying promotions, discounts, or free parking offers, which are periodically provided by the parking centers. This information is retrieved from the database and presented to the users.

6.2 Mobile Server in React Native:

To handle server-side operations within the mobile application, we developed a mobile server using React Native. The mobile server interacts with the Firestore Firebase database through API calls. It handles user authentication and authorization, ensuring that only authenticated users can access the system's functionalities.

Through the API calls, the mobile server retrieves data from the Firestore Firebase database and updates it as needed. This includes fetching information about available parking spaces, user reservations, and additional services. The mobile

server also manages the communication between the mobile application and the other system components.

6.3 IoT Code in Arduino IDE:

For the IoT system, we utilized Arduino boards and implemented code using the Arduino IDE. The code enables the Arduino boards to interface with various sensors deployed in the parking system. Specifically, we incorporated infrared sensors at the entrance, exit, and within each parking space.

The Arduino code reads the sensor data from the infrared sensors, providing information on the occupancy status of the parking spaces. This data is then transmitted to the NodeMCU, which acts as the IoT gateway.

6.4 IoT Gateway (NodeMCU) and Node.js Server:

The NodeMCU serves as the IoT gateway in our car parking project. It receives the sensor data from the Arduino boards via serial communication. The NodeMCU is responsible for establishing a network connection with the server for data transmission.

We developed code for the NodeMCU to send a post request to a Node.js server API. This post request contains the sensor data collected by the Arduino boards. The Node.js server, which is connected to the Firestore Firebase database, processes the received data and updates the database accordingly.

The Node.js server API handles the post request, extracts the relevant sensor data, and utilizes Firestore Firebase's APIs to update the database in real-time. This ensures that the parking occupancy information is consistently up to date for the mobile application and other system components.

6.5 Cloud Processing and Storage (Firebase and Firestore Database):

Firebase serves as the cloud platform for processing and storing data in our car

parking project. Specifically, we employ Firestore as the NoSQL database for storing real-time sensor data, parking information, and user reservations.

The Firestore database provides a scalable and secure solution for storing and retrieving data. It enables real-time updates, ensuring that the parking information displayed in the mobile application remains synchronized with the data collected by the IoT system.

The integration between the Node.js server and the Firestore database enables seamless communication and data synchronization. The Node.js server utilizes Firestore's APIs to update the database based on the received sensor data, ensuring accurate and real-time parking occupancy information.

CHAPTER SEVEN: CONCLUSION

The car parking project successfully achieved its objectives of identifying and defining the requirements of the parking system, developing a mobile application, designing an IoT-based smart parking system, and integrating both components to provide a seamless user experience.

The car parking project accomplished its objectives by delivering a comprehensive solution that addressed the requirements of the end-users. The project combined a user-friendly mobile application, an efficient IoT-based smart parking system, and integration with a cloud-based database for real-time data management. The successful implementation of the project provides an effective and convenient solution for parking management, enhancing the overall parking experience for users.

7.1 Achievements

Through requirement analysis, the needs of the end-users were identified and the necessary features and functionalities formulated for the car parking system. This ensured that the system was designed to meet the specific requirements and expectations of the users.

Secondly, a mobile application was developed using React Native to enable users to conveniently check the availability of parking spaces. The application provides features such as user authentication, real-time updates on parking availability, reservation options, and additional services. This mobile application caters to the needs of users by offering a user-friendly interface and delivering relevant information in a convenient manner.

Thirdly, an IoT-based smart parking system was designed and implemented using Arduino technology. The system utilizes a range of sensors, including infrared sensors at the entrance, exit, and parking spaces, to enable real-time monitoring and control of parking spaces. The Arduino code collects sensor data and transmits it to the NodeMCU, which acts as the IoT gateway.

Fourthly, the mobile application and the IoT-based smart parking system were successfully integrated. The mobile application communicates with the Firestore Firebase database, fetching and updating data related to parking availability, user reservations, and additional services. The IoT system, facilitated by the NodeMCU, sends sensor data to a Node.js server API, which updates the Firestore database in real-time. This integration ensures seamless communication and coordination between the mobile application and the IoT system, providing users with accurate and up-to-date parking information.

7.2 Challenges and Limitations

During the development of the car parking project, the developer encountered several challenges and limitations. These obstacles required creative problem-solving and careful planning to overcome. Here are some key challenges and limitations faced:

7.2.1 Hardware Limitations:

The availability and affordability of hardware components posed a challenge. Certain devices and sensors required for the implementation of a more robust application were either not readily accessible or expensive. This limitation influenced the overall functionality and capabilities of the IoT-based smart parking system.

7.2.2 React Native Package Deprecation:

The project encountered challenges due to the deprecation of certain React Native packages. These deprecated packages caused compatibility issues and forced the developer to search for alternative solutions or migrate to newer packages. This added complexity and required additional time and effort to ensure the stability and performance of the mobile application.

7.2.3 Deprecation of Google Services:

The project relied on various Google services for functionalities such as user authentication, maps integration, and other related features. However, during the

development process, the developer faced challenges due to the deprecation or changes in these services. This necessitated updating the codebase and adopting alternative solutions to ensure the continued functionality of the application.

7.2.4 Obtaining Google Cloud Console Billing Account:

Integrating Google services, such as Firestore Firebase, required a Google Cloud Console Billing account. However, obtaining such an account posed challenges for the developer, particularly regarding the need to have a credit card. This added an extra step and potential complexity in the setup process, causing delays and hindrances in the development timeline.

7.3 Recommendation for further work

While the car parking project has achieved its primary objectives, there are several areas where further work and enhancements can be considered. These recommendations aim to improve the overall functionality, user experience, and extend the capabilities of the system. Some potential areas for further work include:

7.3.1 Reservation and Payment System:

Implementing a reservation and payment feature would enhance the convenience and efficiency of the car parking system. Users could reserve a parking spot in advance and make payments through the mobile application. Integrating a secure and user-friendly payment gateway would streamline the process and provide a seamless experience for users.

7.3.2 Enhanced User Support:

Introducing a comprehensive user support feature within the mobile application would greatly assist users in navigating the system and addressing any issues they may encounter. This could include an FAQ section, live chat support, or a dedicated support ticketing system. Providing prompt and helpful assistance to users would further enhance their satisfaction and overall experience with the application.

7.3.3 Integration with Third-Party Services:

Consider integrating the car parking system with other third-party services to offer additional functionalities and convenience to users. For example, integration with car wash services or valet parking providers could allow users to easily request and schedule these services within the mobile application. Collaborating with relevant service providers would enhance the value proposition of the car parking system.

7.3.4 Advanced Analytics and Reporting:

Incorporating advanced analytics and reporting capabilities can provide valuable insights into parking usage patterns, occupancy rates, and revenue generation. Analyzing such data can help optimize parking management, make informed decisions, and identify areas for improvement. Generating detailed reports and visualizing data within the application would enable administrators to track key performance metrics effectively.

7.3.5 Integration with IoT-based Surveillance:

Explore the integration of IoT-based surveillance systems to enhance security and monitoring within the parking areas. Implementing surveillance cameras and leveraging image recognition technology can help detect unauthorized parking, monitor parking violations, and ensure a safer environment. Real-time video feeds and alerts can be integrated into the mobile application to provide users with added peace of mind.

7.3.6 Continuous User Feedback and Iteration:

Encourage users to provide feedback and suggestions through the mobile application. Regularly collect user feedback to identify areas of improvement and prioritize future enhancements. Engaging users in the development process and implementing iterative updates based on their needs and preferences will contribute to the long-term success and adoption of the car parking system.

REFERENCES

1. Barriga, Jhonattan J., Juan Sulca, José León, Alejandro Ulloa, Diego Portero, José García, and Sang Guun Yoo. 2019. "A Smart Parking Solution Architecture Based on LoRaWAN and Kubernetes." Sensors 19 (24): 5458.

2. Sobeslav, Vladimir, and Josef Horalek. 2018. "A Smart Parking System Based on Mini PC Platform and Mobile Application for Parking Space Detection." Sensors 18 (1): 153.

3. Samsukha, Amit. 2020. "Car Parking Finder Mobile App Development: Cost and Features."

4. Bajpai, Ayush. 2019. "IoT based Smart Parking System using Mobile Application." International Journal of Engineering and Technology (IJET) 8 (3.2): 665-668.

5. Airtel Business. 2017. "IoT-based Smart Parking System – A Step Towards Building Smart City."

6. Zhang, Wei, and Yong Wang. 2019. "Smart Parking System Using RFID Technology." Sensors 19 (17): 3727.

7. Wang, Xin, Wenbo Wang, and Bin Liu. 2018. "A Survey on Smart Parking Systems." IEEE Access 6: 15641-15655.

8. Zhang, Xiao, Xiaodong Li, and Xiaoli Li. 2017. "A Smart Parking System Using Wireless Sensor Networks." IEEE Sensors Journal 17 (11): 3976-3984.

9. Zhang, Xuesong, and Xinping Wang. 2017. "A Smart Parking System Using Image Processing." IEEE Transactions on Intelligent Transportation Systems 18 (8): 2157-2166.

10. Chen, Xin, and Xiaodong Li. 2016. "A Smart Parking System Using Cloud Computing." IEEE Transactions on Intelligent Transportation Systems 17 (4): 1077-1086.

11. Institute for Transportation and Development Policy. 2018. "Nairobi Parking Survey." Nairobi, Kenya: Institute for Transportation and Development Policy.

12. Ministry of Transport, Infrastructure, Housing, Urban Development and Public Works. 2017. "Parking Demand and Supply in Nairobi." Nairobi, Kenya: Ministry of Transport, Infrastructure, Housing, Urban Development and Public Works.

13. Mwaura, M. N., and M. M. Njeru. 2016. "A Study of the Parking Situation in Nairobi." International Journal of Engineering and Technology (IJET).

APPENDIX A: CODE FOR READING SENSOR DATA

```
#include <SoftwareSerial.h>
#include <Servo.h> //includes the servo library
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
SoftwareSerial nodemcu(0, 1);
int data1 = 0;
int data2 = 1;
LiquidCrystal_I2C lcd(0x27, 20, 4);
Servo myservo;
#define ir_enter 2
#define ir_back  4
#define ir_car1 5
#define ir_car2 6
#define ir_car3 7
#define ir_car4 8
int S1=0, S2=0, S3=0, S4=0;
int flag1=0, flag2=0;
int slot = 4;
void setup() {
  nodemcu.begin(9600);
  pinMode(ir_car1, INPUT);
  pinMode(ir_car2, INPUT);
  pinMode(ir_car3, INPUT);
  pinMode(ir_car4, INPUT);
  pinMode(ir_enter, INPUT);
  pinMode(ir_back, INPUT);
```

```
  Read_Sensor();
  int total = S1+S2+S3+S4;
  slot = slot-total;
}
void loop() {
  Read_Sensor();
  slot = S1 + S2 + S3 + S4;
  String concatenatedValue = String(slot);
  nodemcu.print(concatenatedValue);
  delay(1000);
}
void Read_Sensor(){
S1=0, S2=0, S3=0, S4=0;
if(digitalRead(ir_car1) == 0){S1=1;}
if(digitalRead(ir_car2) == 0){S2=1;}
if(digitalRead(ir_car3) == 0){S3=1;}
if(digitalRead(ir_car4) == 0){S4=1;}
}
```

APPENDIX B: NODEMCU CODE TO READ DATA TRANSMITTED FROM ARDUINO

```
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>
SoftwareSerial nodemcu(D5, D6);
const char* ssid = "OPPO Reno7";
const char* password = "abdul@1383";
const char* serverIP = "192.168.56.251";  // Replace with your Node.js server IP address
const int serverPort = 5000;
String serverName = "http://192.168.56.251:5000/data";
int data;
String ch;
WiFiClient client;
unsigned long lastTime = 0;
unsigned long timerDelay = 20000;

void setup() {
  // Initialize Serial port
  Serial.begin(9600);
  nodemcu.begin(9600);

  WiFi.begin(ssid, password);
}

void loop() {
  data = nodemcu.read();
```

```
ch = String(data);
Serial.write(data);
delay(1000);
if ((millis() - lastTime) > timerDelay) {
  //Check WiFi connection status
  if(WiFi.status()== WL_CONNECTED){
    WiFiClient client;
    HTTPClient http;

    http.begin(client, serverName);

    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    int httpResponseCode = http.POST(ch);

    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    http.end();
  }
  else {
    Serial.println("WiFi Disconnected");
  }
  lastTime = millis();
 }
}
```

APPENDIX C: NODE JS SERVER CODE TO TRANSMIT DATA TO FIRESTORE

```
const express = require("express");
const app = express();
const admin = require("firebase-admin");
const credentials = require("./key.json");
admin.initializeApp({ credential: admin.credential.cert(credentials)});
const db = admin.firestore();
app.use(express.json());
app.use(express.urlencoded({extended: true}));
app.post('/data', (req, res) => {
   res.sendStatus(200);
   const key = parseInt(Object.keys(req.body)[0]);
   // Convert the key from ASCII to character
   const character = String.fromCharCode(key);
                                                const                parkingCenterRef                =
db.collection('parkingcenters').doc('NsTLzu7lFoZDQrjaQAaRAyZf47V2');
   parkingCenterRef.update({ available: character })
      .then(() => {console.log('Parking center updated successfully');})
      .catch((error) => {
         res.sendStatus(500);
      });
});
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => {
   console.log(`Server listening on port ${PORT}`);
})
```

APPENDIX D: REACT NATIVE CODE TO FIND PARKING CENTERS

```
import { ScrollView, StyleSheet, Text, View, Dimensions, TouchableOpacity, Modal } from
'react-native'
import React,{useState, useEffect} from 'react'
import {Icon} from 'react-native-elements'
import ParkingHeader from '../components/ParkingHeader'
import { menu, parkingData } from '../global/Data'
import {colors,fonts} from '../global/mstyles'
import { TabBar, TabView } from 'react-native-tab-view'
import MenuScreen from './ParkingTabs/MenuScreen'
import firebase from 'firebase';
import { db } from '../firebase';

const SCREEN_WIDTH = Dimensions.get('window').width
const initialLayout = SCREEN_WIDTH;

const ParkingHomeScreen = ({navigation,route}) => {
  const {id,parking} = route.params;
  const [centerData, setCenterData] = useState([]);
  const [centerPromo, setCenterPromo] = useState('');
  const [centerId, setCenterId] = useState('');
 useEffect(() => {
  fetchParkingCenter();
 }, []);
 const fetchParkingCenter = async () => {
  try {
                                    const      parkingCenterSnapshot      =      await
```

```
db.collection('parkingcenters').where('centerName', '==', parking).get();
    if (!parkingCenterSnapshot.empty) {
      const parkingCenter = parkingCenterSnapshot.docs[0];
      const parkingCenterId = parkingCenter.id;
      const parkingCenterData = parkingCenterSnapshot.docs[0].data();
      console.log("id: ", parkingCenterId);
      //console.log(parkingCenter);
      setCenterId(parkingCenterId);
      console.log("centerData: ", parkingCenterData);
      setCenterData(parkingCenterData);
      console.log("centerId: ", centerId);
    }
  } catch (error) {
    console.error('Error fetching parking center:', error);
  }
};
/*useEffect(() => {
  checkPromotions();
}, [centerId]);*/
const checkPromotions = async () => {
  if (centerId) {
    try {
      //console.log("centerId: ", centerId);
      const promotionsSnapshot = await db.collection('promotions')
        .where('parkingLotId', '==', centerId)
        .where('endDate', '>=', firebase.firestore.Timestamp.now())
        .orderBy('endDate')
```

```
      .limit(1)
      .get();
    if (!promotionsSnapshot.empty) {
      const promotion = promotionsSnapshot.docs[0].data();
      // Do something with the
      console.log(promotion.description);
      //setCenterPromo(promotion.description)
    }
  } catch (error) {
    console.error('Error checking promotions:', error);
  }
 }
};
 return (
   <View style={styles.container}>
     <ScrollView>
       <View>
         <ParkingHeader id={id} navigation={navigation}/>
           <View style ={styles.view1}>
             <Text style ={styles.text1}>centerPromo</Text>
           </View>
         <View style ={styles.view2}>
           <View style ={styles.view3}>
             <Text style ={styles.text2}>{centerData.centerName}</Text>
             {/*<Text style ={styles.text3}>{parkingData[id].serviceType}</Text>*/}
             <View style ={styles.view4}>
               <Icon name ="star" type ="material-community" color = {colors.grey3}
```

```
size = {15} />
                    <Text style ={styles.text4}>{centerData.averageReview}</Text>
                                <Text style ={styles.text5}>({centerData.numberOfReview}
reviews)</Text>
                            <Icon name ="map-marker" type ="material-community" color =
{colors.grey3} size = {15} />
                    <Text style ={styles.text5}>{centerData.farAway} mi away</Text>
                  </View>
                </View>
                <View style ={styles.view8}>
                  <Text style ={styles.text6}>Est.Arrival Time</Text>
                  <View style ={styles.view9}>
                    <Text style = {styles.text9}>{centerData.estimatedArrivalTime}</Text>
                    <Text style ={styles.text11}>min</Text>
                  </View>
                </View>
              </View>
            </View>
            <View style={styles.view10}>
              <TabView
                navigationState={{index,routes}}
                renderScene={UpdateRoute1}
                onIndexChange={setIndex}
                initialLayout={initialLayout}
                renderTabBar={renderTabBar}
                tabBarPosition='top'
              />
```

```
            </View>
            {/*
               index === 0 &&
                  <MenuScreen onPress={menuPressed}/>
            */}
         </ScrollView>
                                                                  <TouchableOpacity
onPress={()=>{navigation.navigate("ReservationScreen",{id:id,email:centerData.centerEm
ail,hprice:centerData.prices.hourly})}}>
            <View style={styles.view11}>
               <View style={styles.view12}>
                  <Text style={styles.text13}>Reserve</Text>
               </View>
            </View>
         </TouchableOpacity>


      </View>
   )
}


export default ParkingHomeScreen


const styles = StyleSheet.create({
   container:{
      flex:1,
      paddingTop:20
   },
```

```
view1:{
    padding:3,
    alignItems:"center",
    justifyContent:"center"
},
text1:{
    color:colors.buttons,
    fontSize:14,
    fontWeight:"bold"
},
view2:{
    flexDirection:"row",
    flex:1,
    marginBottom:5,
    marginHorizontal:10,
    justifyContent:"space-between",
},
view3:{
    flex:8,
},
text2:{
    fontSize:20,
    fontWeight:"bold",
    color:colors.grey1
},
text3:{
    fontSize:15,
```

```
    color:colors.grey3
  },
  view4:{
    flexDirection:'row',
    alignItems:"center",
    marginTop:5
  },
  text4:{
    //fontFamily :fonts.android.bold,
    fontWeight:"bold",
    fontSize:13,
    color:colors.grey3,
    marginLeft:2,
  },
  text5:{
    //fontFamily :fonts.android.bold,
    fontWeight:"bold",
    fontSize:13,
    color:colors.grey3,
    marginLeft:2,
    marginRight:5
  },
  text6:{
    //fontFamily :fonts.android.bold,
    fontWeight:"bold",
    fontSize:13,
    color:colors.grey3,
```

```
    marginLeft:0,
  },
  view5:{
    flex:3,
    alignItems:"center"
  },
  text6:{
    fontSize:15,
    fontWeight:"bold",
    color:colors.grey1
  },
  view7:{
    width:40,
    height:40,
    alignItems:"center",
    borderRadius:20,
    justifyContent:"space-around",
  },
  text7:{
    fontSize:16,
    fontWeight:"bold",
    color:colors.black,
    marginTop:5
  },
  text8:{
    fontSize:13,
    color:colors.black,
```

```
    marginBottom:5
  },
  view8:{
    flex:4,
    alignItems:"center",
    paddingVertical:20,
    paddingRight:2,
  },
  text9:{
    fontSize:15,
    fontWeight:"bold",
    color:colors.cardbackground
  },
  view9:{
    width:40,
    height:40,
    backgroundColor:colors.buttons,
    alignItems:"center",
    borderRadius:20,
    justifyContent:"space-around",
  },
  text10:{
    fontSize:16,
    fontWeight:"bold",
    color:colors.cardbackground,
    marginTop:5
  },
```

```
  text11:{
    fontSize:13,
    color:colors.cardbackground,
    marginBottom:5
  },
  view10:{
    elevation:10,
    backgroundColor:colors.pagebackground
  },
  view11:{
    backgroundColor:colors.buttons,
    height:50,
    alignContent:"center",
    marginBottom:0,
    justifyContent:"center"
  },
  view12:{
    flexDirection:"row",
    justifyContent:"space-between",
    alignItems:"center"
  },
  text12:{
    padding:10,
    fontWeight:"bold",
    fontSize:18,
    color:colors.cardbackground
  },
```

```
view13:{
    borderWidth:1,
    marginRight:10,
    borderColor:colors.cardbackground,
    borderRadius:6,
    paddingBottom:2
},
text13:{
    paddingHorizontal:3,
    fontWeight:"bold",
    fontSize:18,
    color:colors.cardbackground,
},
tab:{
    paddingTop :0,
    backgroundColor:colors.buttons,
    justifyContent:"space-between",
    alignItems:"center"
},
tabContainer:{
    alignItems:'center',
    alignContent:'center',
    justifyContent:'center',
},
tabLabel:{
    fontWeight:'bold',
    color: colors.cardbackground
```

```
  },
  tabStyle:{
    width:SCREEN_WIDTH/4,
    maxHeight:45,
  },
  view14:{
    flexDirection:"row",
    alignItems:"center",
    padding:10,
    backgroundColor:colors.buttons,
    top:0,
    left:0,
    right:0,
    paddingTop:25
  },
  text14:{
    fontWeight:"bold",
    marginLeft:40,
    color:colors.black,
    fontSize:18
  },
  view15:{
    marginTop:5,
    paddingBottom:20
  },
})
```