POLITEKNIK SULTAN SALAHUDDIN ABDUL AZIZ SHAH

SMART PARKING SYSTEM

RAAHUL SURESH

A project report submitted in partial fulfilment of the requirements for the award of the Diploma of Electronic Engineering(MEDICAL)

ELECTRICAL ENGINEERING DEPARTMENT

SESI 2 2022/2023

ACKNOWLEDGEMENT

I would like to thank my supervisor, PUAN NAAGAJOOTHI A/P ADIN NARAINA for his excellent advice, guidance and motivation. To the members, colleagues and other people that contributed to my project of the Politeknik Shah Alam (Psa), I owe sincere and earnest thankfulness for the unforgettable and unique experience of my life. I am also very thankful to my family for their supports, especially throughout this period of my life and kept me going all the way through my study.

ABSTRACT

In recent times the concept of smart cities have gained great popularity. Thanks to the evolution of Internet of things the idea of smart city now seems to be achievable. Consistent efforts are being made in the field of IoT in order to maximize the productivity and reliability of urban infrastructure. Problems such as, traffic congestion, limited car parking facilities and road safety are being addressed by IoT. In this paper, we present an IoT based cloud integrated smart parking system. The proposed Smart Parking system consists of an on-site deployment of an IoT module that is used to monitor and signalize the state of availability of each single parking space. A mobile application is also provided that allows an end user to check the availability of parking space and book a parking slot accordingly.

TABLE OF CONTENT

APPENDICES

CHAPTER 1

1.1 INTRODUCTION

It is very common to have huge crowds in shopping malls during peak hour (lunch time, dinner time, weekend) and sales periods. Discounts offered by merchants in shopping mall can attract a lot of customers to come. Most of the customer travels to the shopping mall with their own vehicles and this indirectly cause the problem of finding empty car parking space for the car drivers. Parking space is necessary for those who love to go for shopping and it is a must for shopping complex to provide a comfortable, secure and quality parking space for their customers. Nowadays, parking space need to be constructed in multi-level since the number of vehicles is increasing rapidly. As a consequence, drivers may face difficulties when they are required to seek in the lengthy multi-level parking space. It is very troublesome for a driver to seek an unknown available empty car parking space in a large car park.

Nowadays, there are several car park guidance systems using different technologies to resolve the parking problem in car park. In Vehicle Navigation and Information System Conference 1995, Akihito used the vehicle sensors at each parking lot to track the exact locations of the empty parking spaces. Wolff et al used magnetic field sensor and wired based concept to test on their simulation car park model. In the year of 2008, Sunway Pyramid and The Mines shopping complexes also installed with this vehicle sensor and indicators above each parking space. It will show green light when no occupied and red light when occupied to assist the driver to find parking space. This technology can provide comprehensive knowledge of the car park occupancy is thus available and assist driver in searching empty car parking space. Thus, it dramatically decrease the searching time need to find an empty car parking space. One shortcoming of vehicle sensor system is driver may not have luck to park their car near to the entrance. They may lead by the indicator to park their car far away and this does not benefit the driver anymore. Another possible situation is when there is only one available space; there will be more than one car drive toward it. Competing on an available car parking space is a waste of time and petrol. Furthermore, a lot of sensors have to be used inside a car park. This will be costly for a huge parking area and it also added up to the sensor maintenance cost.

Some parking systems of foreign countries are providing online reservation service to customer which they can reserve a car parking space for a period of time like Munich Airport. Anyhow, problem arise when they park over the limit of reserved time, their reserved parking space has been parked by others and etc. Some of the system provided the service that assigned parking space for driver via number printed on ticket, but the parking space may be far away from driver. Sometimes it is inconvenient as the parking space may be taken over by those who are illegal parking or looking for available parking space too. One of the major problems faced in online reservation systems is wasting resources. Some drivers may reserve the parking spaces but didn’t come to park or miss the parking period. So, the system will hold the reserved parking spaces until the drivers come or didn’t come. This will waste the parking spaces because the system cannot utilize the available parking spaces all the time in the car park. Besides, most drivers do not have the necessary computer knowledge to make reservation via the web. Therefore the system is aiming at computer literate younger generation.

With the advancement of wireless technologies, wireless mobile-based methods have been employed in car parking system. A mobile app based car parking system will be more efficient and effective method .This system provides users with information about the vacant car park areas and payment for the parking tickets. However, this system does not only allows user to check the availability of car parking spaces but also provide any exact information about car parking spaces location such as parking lot ID.

1.2 PROBLEM STATEMENT

The existing car parking system in Malaysia usually required the car drivers to search an empty parking space in the car park without providing detail direction toward the available parking space. As the result, drivers may waste a lot of time and unnecessary energy while they turn around in the car park without direction and may cause car traffic congestion in parking space. This paper investigates the problems of car parking system in Malaysia and finally proposed a Wireless Mobile-based Car Parking System using low cost mobile app service. The implementation of mobile app service into the car parking system enable drivers to receive information regarding the availability of car parking spaces. In this system, the drivers can use this app to request for new assignment of car parking spaces if they fail to get the previous assigned destination. This paper demonstrates the design and implementation of Wireless Mobile-based Car Parking System using mobile app services by Breadth First Search algorithm in finding the nearest parking space for drivers. The stimulation results reveal the intelligence of this system can efficiently allocate and utilize the car parking spaces inside the car park.

1.3 OBJECTIVE

The objectives of the project are:

- To design a system that allows the users to easily search for their vechicles in shopping complex without any stress.
- To save time finding parking spots
- To create a systematic parking environment
- To enhance the security with simplifying parking system
- To allow users to be aware their parking status.

1.4 SCOPE OF PROJECT

Wireless Mobile-based Car parking System (WMCPS) has been developed using mobile app service approach. This particular system will search for an available parking space using shortest path algorithm (BFS algorithm) and print the specific parking lot ID on the ticket that the driver taken before they enter the car park. Besides, the system allows the driver to request for a new assigned parking lot via the app if the previous assigned parking lot is occupied. The system has been tested on a stimulation model with different conditions such as slow arrival rate, medium arrival rate and fast arrival rate. By simulation results, it shows that Wireless Mobile-based Car parking System can allocate the best available parking space to the driver in term of providing the shortest path distance. Furthermore, the result also shows that the performance of the system is less affected by the arrival rate of car that enters the car park. It is because when the car park is nearly full, drivers have to reach for parking spaces in the remote location of the car park. Feature of the mobile app service is able to resolve the problem of failure parking that happened in car park. The WMCPS will be extended by considering the integration of GPS technology to provide guidance to the drivers to achieve a more reliability system in future.

1.5 PROJECT SIGNIFICATION

While making use of a car parking system, drivers don’t have to spend time searching around for an available spot instead they can directly move to an available space which is either shown on the board, indicated by the sensor or shown in their mobile depending on the type of parking system being implemented. Driving around searching for parking can be dangerous because drivers do not have a full concentration on the road because their focus is on seeing an available spot. Therefore, reaching a parking slot definitely makes it easy for drivers and also removes tension and frustration which increases safety around the car parking. The parking system also monitors the driver's’ vehicles which also increases safety.

CHAPTER 2

2.0 LITERATURE REVIEW

2.1 HISTORY OF PARKING SYSTEM

Over the years, car parking systems and the accompanying technologies have increased and diversified. Car parking systems have been around almost since the time cars were invented. In any area where there is a significant amount of traffic, there are car parking systems. Car Parking systems were developed in the early 20th century in response to the need for storage space for vehicles.

In the 1920s, forerunners of automated parking systems appeared in U.S. cities like Los Angeles, Chicago, New York City and Cincinnati. Some of these multi-storey structures are still standing, and have been adapted for new uses. One of the Kent Automatic Garages in New York (now known as the Sofia Apartments) is an Art Deco landmark that was converted into offices and luxury condominiums in 1983. A system that is now found all over Japan — the "ferris-wheel," or paternoster system — was created by the Westinghouse Corporation in 1923 and subsequently built in 1932 on Chicago's Monroe Street. The Nash Motor Company created the first glass-enclosed version of this system for the Chicago Century of Progress Exhibition in 1933, and it was the precursor to a more recent version, the Smart Car Towers in Europe.

I) SMART PARKING SYSTEM

It seems that the word ''smartness" or ''smart'' holds different meanings according to the requirements of people and their location in the stream of time. Over the years, the number of parking-system-related technologies has increased. Car parking systems have been around almost since the time cars were invented. In any era where there has been a significant amount of traffic, there have been car-parking systems (Melsen 2013). Car parking systems were first developed in the early 20th century in response to the need for storage space for vehicles. (Melsen 2013) Indeed, the using of e-smart parking systems dated back to the 1920s, as automated parking systems appeared in U.S. cities such as Los Angeles, Chicago, New York, and Cincinnati. In addition, one of the Kent automatic parking garages in New York is an art deco landmark that was converted into luxury condominiums in 1983. A system that is prevalent all over Japan Thirty Seventh International Conference on Information Systems, Dublin 2016 3 is the "Ferris-wheel," or "paternoster system", which — was created by the Westinghouse Corporation in 1923 and built in 1932 on Chicago's Monroe Street. In the past two decades, the concept of intelligence in terms of smart parking systems became more popular in the most vibrant cities, especially in malls and shopping centres. (Melsen 2013) In the mid-80s, the systems used for parking relied mainly on the traditional method of pushing a button in the device next to the checkpoint to get a parking ticket and on exiting, the driver must pay before inserting their ticket in order for the barrier to rise. This was the method used to determine how many cars came in and out the system each day, and it was used to count the number of vacant spaces available. It began by utilizing different methods such as sensors or barriers to be able to know the status of parking lots. All these methods developed dramatically further until recently the term ‘smart city vision’ emerged.

2.2 LED

As is evident from its name, LED (Light Emitting Diode) is basically a small light emitting device that comes under “active” semiconductor electronic components. It’s quite comparable to the normal general purpose diode, with the only big difference being its capability to emit light in different colors. The two terminals (anode and cathode) of a LED when connected to a voltage source in the correct polarity, may produce lights of different colors, as per the semiconductor substance used inside it. A light-emitting diode is a two-lead semiconductor light source. It is a p–n junction diode that emits light when activated. When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor.

2.3 Servo Motor

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback.

2.4 WIFI Module

The Arduino Uno WiFi is an Arduino Uno with an integrated WiFi module. The board is based on the ATmega328P with an ESP8266 WiFi Module integrated. The ESP8266 WiFi Module is a self contained SoC with integrated TCP/IP protocol stack that can give access to your WiFi network (or the device can act as an access point). One useful feature of Uno WiFi is support for OTA (over-the-air) programming, either for transfer of Arduino sketches or WiFi firmware.The Arduino Uno WiFi is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards and running both online and offline. For more information on how to get started with the Arduino Software visit the Getting Started page.

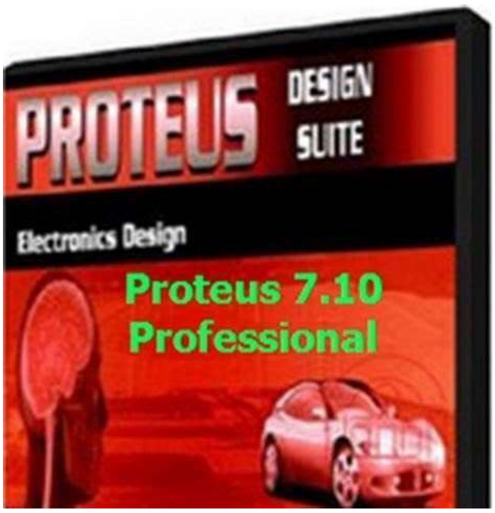
2.5 LCD Display

This **function** turns on any text or cursors that have been printed to the **LCD screen**. The **function lcd**. noDisplay() turns off any text or cursors printed to the **LCD**, without clearing it from the **LCD's** memory.This library allows an Arduino board to control LiquidCrystal displays (LCDs) based on the Hitachi HD44780 (or a compatible) chipset, which is found on most text-based LCDs. The library works with in either 4- or 8-bit mode (i.e. using 4 or 8 data lines in addition to the rs, enable, and, optionally, the rw control lines).

2.6 Ultrasonic sensors

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity.

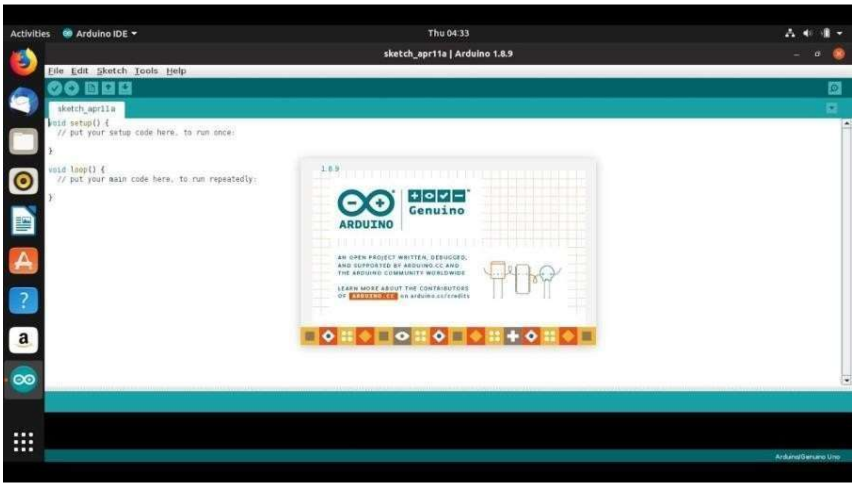
2.7 Software (Proteus 7):



Proteus (PROcessor for TExt Easy to USe) is a fully functional, procedural programming language created in 1998 by Simone Zanella. Proteus incorporates many functions derived from several other languages: C, BASIC, Assembly, Clipper/dBase; it is especially versatile in dealing with strings, having hundreds of dedicated functions; this makes it one of the richest languages for text manipulation.

Proteus owes its name to a Greek god of the sea (Proteus), who took care of Neptune's crowd and gave responses; he was renowned for being able to transform himself, assuming different shapes. Transforming data from one form to another is the main usage of this language.

2.8 Arduino IDE Software



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

AN OPEN PROJECT WRITTEN, DEBUGGED, AND SUPPORTED BY ARDUINO.CC AND THE ARDUINO COMMUNITY WORLDWIDE

LEARN MORE ABOUT THE CONTRIBUTORS OF ARDUINO.CC on arduino.cc/credits

Arduino UNO The Arduino Integrated Development Environment (IDE) is a crossplatform application (for Windows, mac OS, Linux) that is written in functions from C and C++. It is used to write and upload programs to Arduino compatible boards, but also with the help of 3rd party cores., other vendor development boards. The source code for the IDE is released under the GNU General Public License version 2. The Arduino IDE employs the program avrdude to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board firmware. By default, avrdude is used asthe uploading tool to flash the user code onto official Arduino board.

2.9 PREVIOUS RESEARCH

ElakyaR,Juhi Seth, Pola Ashritha, R Namith

It consists of three sections: first section is the parking area which includes Arduino devices along with IR sensor. The user interacts with the parking area with the help of these devices. The user cannot enter the parking area without the help of RFID card. The second section contains the cloudbased web services which acts a mediator between the user and parking area. The cloud is updated depending upon the availability of the parking area. The admin administers the cloud services and it can also be viewed by the user for checking the availability. The third section is the user side. The user gets notification on the basis of the availability via SMS through GSM module.

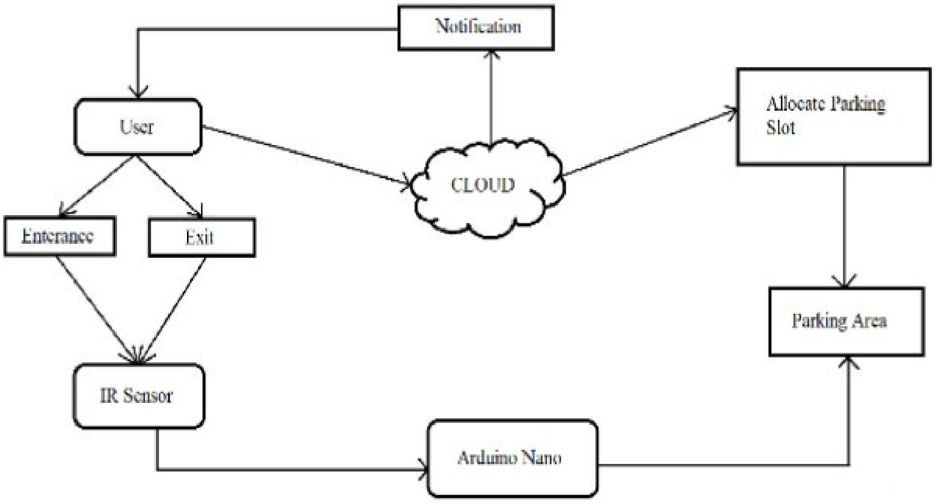


Fig. 1. System Architecture B. Hardware The three main hardware components used are GMS module, RFID card, IR sensors. A user is allowed inside a parking space only if the user has a RFID card. RFID card contains the information of the registered user. As the car enters the parking slot, reader module scans the registered user's RFID tag. The data is sent to the ardunio for checking the availability of the car parking and simultaneously, the user is notified through SMS about the status of the parking area. The GSM module sends the message according to the availability. IR sensor sends the signals according to the presence of the vehicle.

Muftah Fraifer , **Mikael Fernström**

IDC-UNIVERSITY OF LIMERICK

LIMERICK-IRELAND

Challenges of the Proposed Smart Parking Systems

There are major challenges facing today's transportation systems and drivers on a daily basis regarding special parking systems for which smart city engineers and designers have to be prepared. Numerous recent studies have led to the conclusion that new smart parking systems are needed in almost every metropolitan city in the world especially in the next ten years to alleviate many problems, such as petrol consumption and pollution emission, and to improve time-saving and reduce frustration when looking for a parking space. Therefore, for any proposed system to be considered smart in relation to the parking process, it should have as a minimum, the following factors and specifications:

. Be able to accurately sense vehicle occupancy in real-time

. Provide guidance for users about available parking

. Simplify the parking experience and add value for parking stakeholders, such as drivers

. Enable intelligent decisions to be made using data, including real–time status applications, and historical analytic reports

. Be able to provide the user with all the necessary information about the status of any changes in the parking area that might happen in real time

These challenges must be addressed from the very beginning to ensure that the system will work efficiently. Many studies related to traditional smart parking systems in the last decade have indicated that they satisfy neither the drivers' requirements nor the parking facility's budget.

Related Work and Classification of Smart Parking Systems

This section discusses the different methodologies used for smart parking. Also in this section, all related studies are gathered into groups based on the techniques used (conference papers, articles) specifically in the academic domain. It is very clear from all the references below that categories and classifications of smart parking vary from source to source. Some rely on the technology used while others rely on data processing to get information about the parking statues. For example, in the centralized assisted parking search, the information processing will be stored on the central processor (server). The non-assisted parking search does not have a server, and no information will be provided to a user.

Prosper philip1, Sathish kumar selvaperumal2, Ravi lakshmanan3, Chandrasekharan nataraj4

1,2,3,4,Faculty, of Computing, Engineering and Technology, Asia Pacific University

The designed system is composed of five levels, these are: sensor, network, middleware, application and mechanical. The sensor and mechanical level exists in the parking lot, where they're used to detect a vehicle, and control entry and exit. Communication and transmission of information to the servers is done wirelessly, and the middleware act as a bridge between the two systems. Fig.1 depicts critical functionalities of each sub-system. The client parking system primarily handles vehicle detection, uploading of lot status to server, and configuring

the LEDs. The parking system server receives these information and processes it to determine the total available parking in the specified location. The information is also saved as historical data, of which can be retrieved using the API.

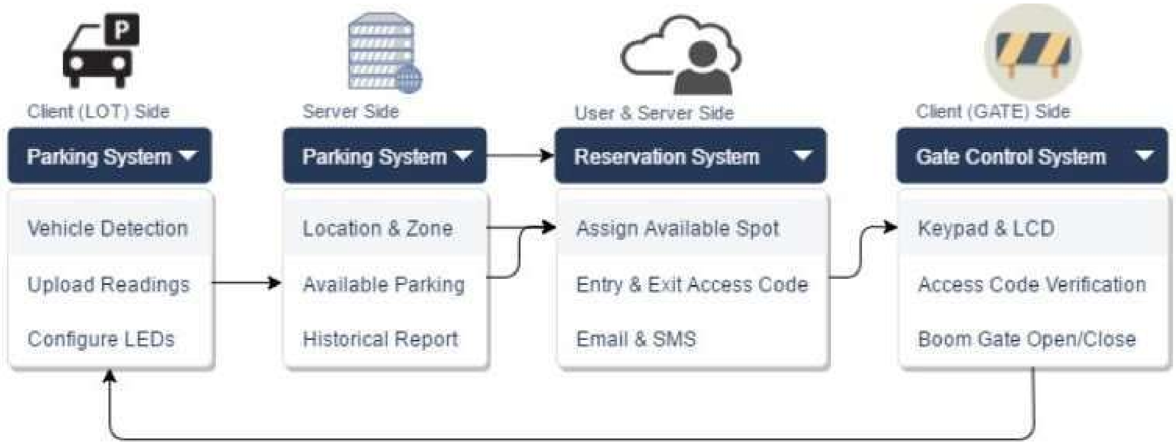


Fig.1: Block Diagram of the system implemented

The reservation system allows the user to make reservation, as it assigns an available parking spot, along with entry and exit access codes to be used at the entrance gate. Confirmation of the reservation is sent via SMS and email to the specified user mobile number and registered email.

The gate control system monitors and allows input of the access codes using a keypad. Verification and validation of the code is done by querying the database server. A result is returned back to the controller; if the status result is “1” the boom gate opens, else if “0” the gate remains closed whilst prompting a message to the user.

Constructional details

This section describes the production and communication protocol processes for the project. In order to ensure that all components and system algorithm is working, the system is designed and simulated using PROTEUS. After verification and validation of the simulation process, the system is then assembled and tested on a breadboard. When all system components and tests are finalized, the system is then implemented on a PCB and final connectivity and continuity tests are conducted. The server application developed in an Ubuntu 16.04 Server using Laravel Framework and is hosted via Cloud9. It is designed using the following use-case scenarios: user is unique, email is used for registration, user can make reservations, phone number is required to make reservation, entry and exit access codes are generated upon successful reservation, access codes are sent via SMS & email.

J. Cynthia, C. Bharathi Priya, P. A. Gopinath

In metropolitan areas, people prefers cab or car as convenient to go to shopping centers, theaters or hotels. Finding place to park vehicles in densely populated area would waste time and consumes fuel during searching for parking space. Hence there is a need for assistive technology, which would communicate the availability of parking slots to the registered user‟s. Mobile app would allow the users to register for the service and if the destination and estimated arrival time is specified, app need to find the free parking space and send the location to the user. User makes the online payment to book the parking slot. Figure 2, illustrates the architecture of smart parking system.

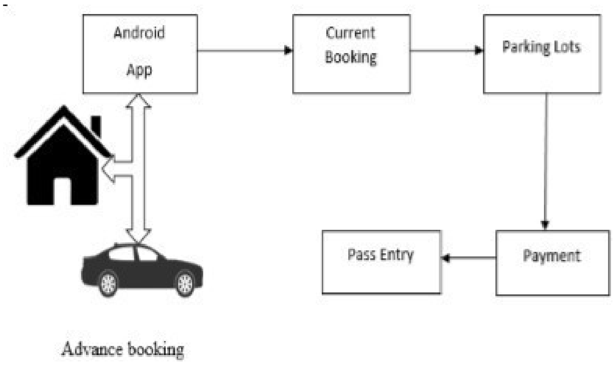


Figure 2. Architecture of Online booking for parking slot For each parking region, Infra-Red (IR) sensors are deployed and IR sensors would detect the number of parking slots, Number of free and booked slots are graphically displayed in LCD screen, WIFI module is used for communication between mobile app and sensors. Figure 3 shows a detecting of empty parking slot and communicating used Wi-Fi to Arduino.

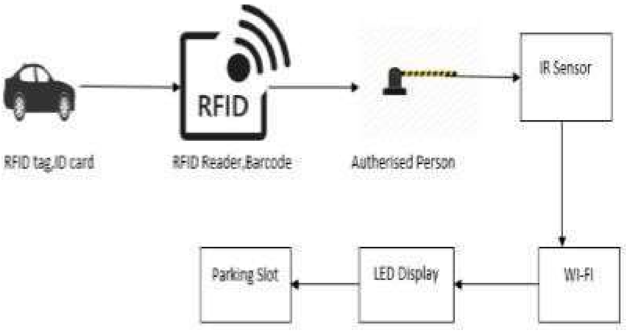


Figure 3.Architecture of Deducting Empty Parking Slot

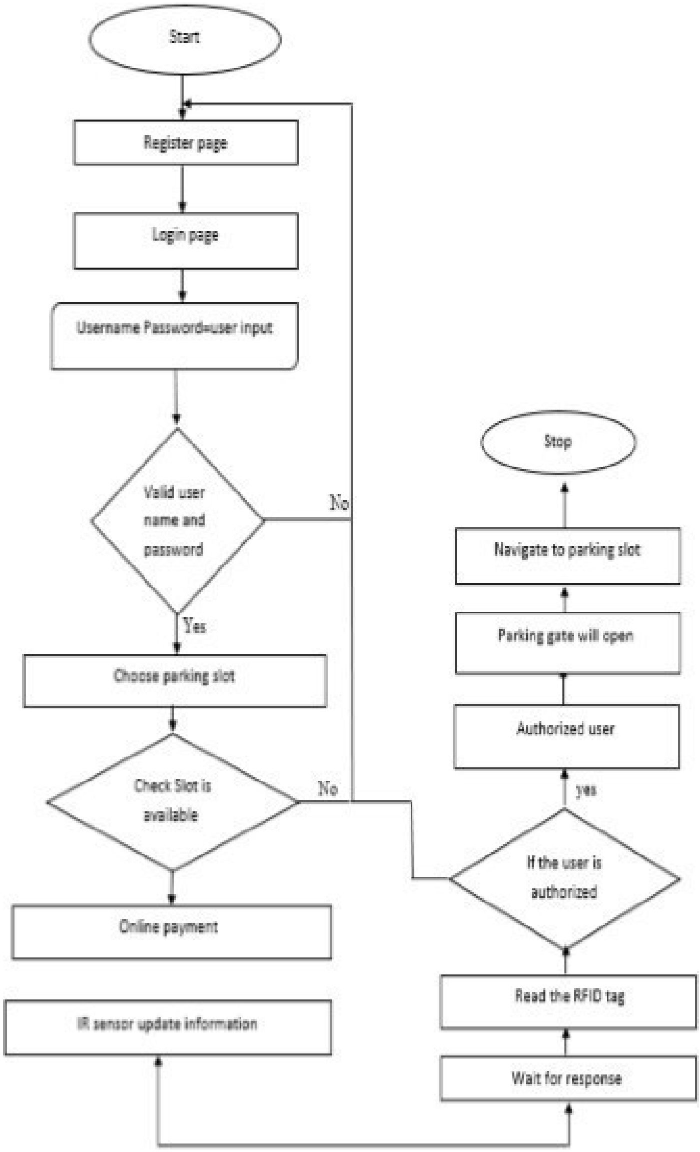
PROPOSED SYSTEM

The proposed system consists of phases. Each of the phase is explained below:

1. Development of Android app
2. Free Space Identification
3. Authenticating user vehicle
4. Classify parking slot

5. Navigating to parking Slot
6. Visualization in Server for Owner to Analyze

To enable a user to use the smart parking system, user need to register with user ID with vehicle number. User can set up the default payment option in his account settings. The android app is built for booking parking slot and payments. The application is used to find the free slot and user need to specify the estimated time of arrival and parking slot usage start and end time. The IR sensors used to identify the parking slot is free or occupied. Parking slot is empty LED shows slot number N (empty), D (occupied). After booking for free parking slot, if the vehicle enters the entrance gate, it is assumed that each car has built in RFID card and RFID reader verifies the vehicle and is authenticated. The parking slot may be allotted for small vehicle and large vehicle. Navigating to parking Slot Android application having GPS location to navigate the allotted parking area to booked user. It graphically navigates from current location to parking area location. Web page shows lane details date and time, booking time lane status, user detail and user feedback.


Start
Register page
Login page
Username Password=user input
Valid user name and password
No
Yes
Choose parking slot
Check Slot is available
No
Online payment
IR sensor update information
Stop
Navigate to parking slot
Parking gate will open
Authorized user
yes
If the user is authorized
Read the RFID tag
Wait for response

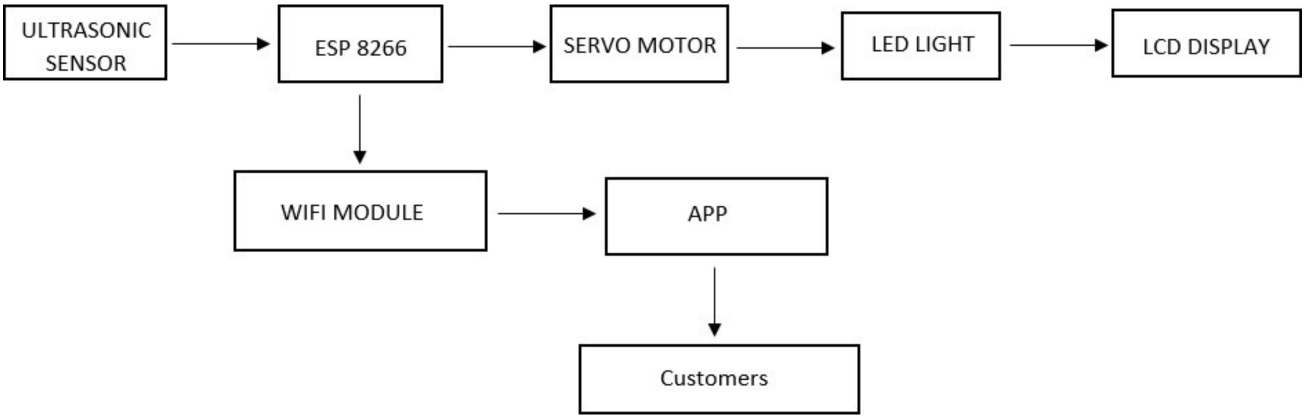
CHAPTER 3

3.1 INTRODUCTION OF METHODOLOGY

Essentially, a methodology is a collection of methods, practices, processes, techniques, procedures, and rules. In project management, methodologies are specific, strict, and usually contain a series of steps and activities for each phase of the project's life cycle. Moreover we also did some survey to succeed this project.he methodology is the research methods that are used for different procedures of data collection, calculation, and analysis in the research, along with the justification for using specific methods.It is generally the part where researchers explain their study. The methodology has greater importance in research.

3.2

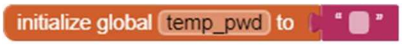
3.2.1 : (I) BLOCK DIAGRAM



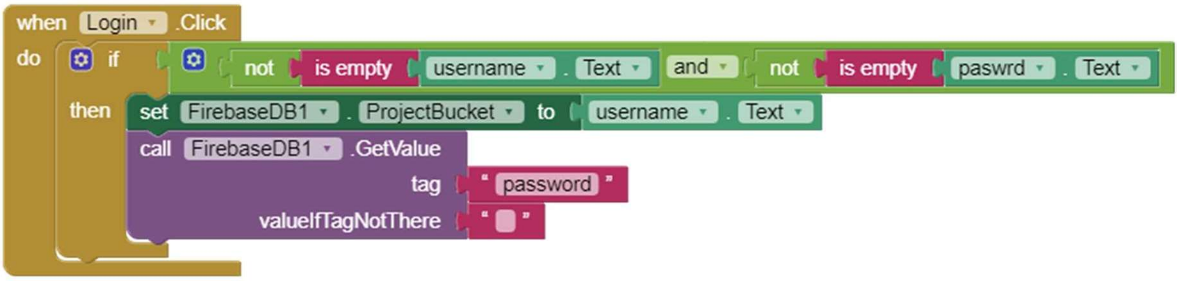
As shown is figure 3.1 above, Firstly the ultrasonic sensor will detect the presence of the car whether it is occupied or unoccupied by car and send the signal to the arduino uno.Then,the arduino uno will send the signal to the servo motor and make the automated barrier gate to open or close. Furthermore the arduino uno is also send the information to the wifi module and the wifi module will send the information to the app. Then the customers can check whether they have a parking lot in any level of the building or not. If they do have a parking lot space they can immediately book it anyone does.

(II) MIT APP INVENTOR

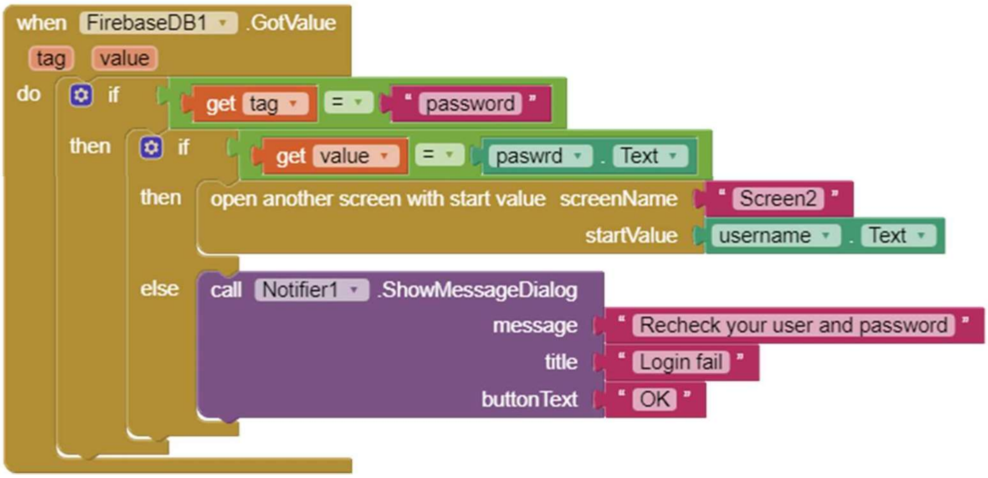
SCREEN 1



```
initialize global temp_pwd to " "
```

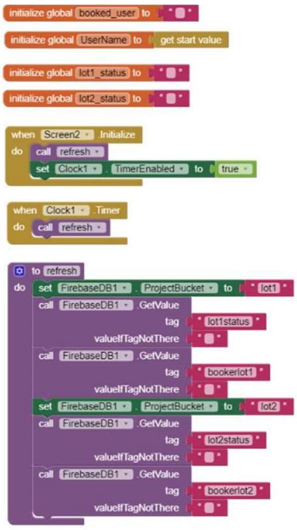


```
when Login .Click
do  if  not is empty username . Text  and  not is empty paswrd . Text
    then  set FirebaseDB1 . ProjectBucket to username . Text
          call FirebaseDB1 .GetValue
                              tag  " password "
               valueIfTagNotThere  " "
```

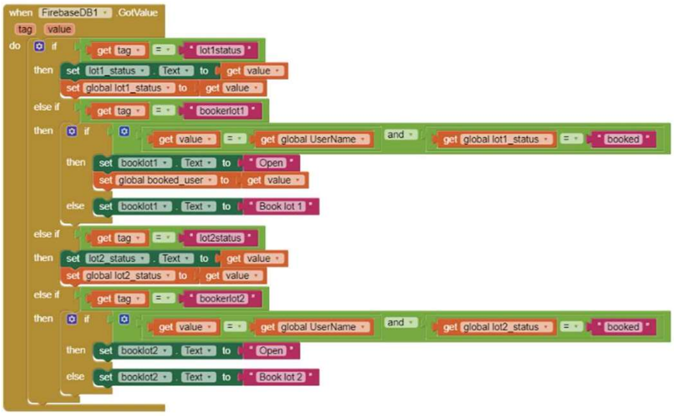


```
when FirebaseDB1 .GotValue
 tag  value
do  if  get tag = " password "
    then  if  get value = paswrd . Text
          then  open another screen with start value  screenName " Screen2 "
                                                       startValue username . Text
          else  call Notifier1 .ShowMessageDialog
                                      message " Recheck your user and password "
                                        title " Login fail "
                                   buttonText " OK "
```

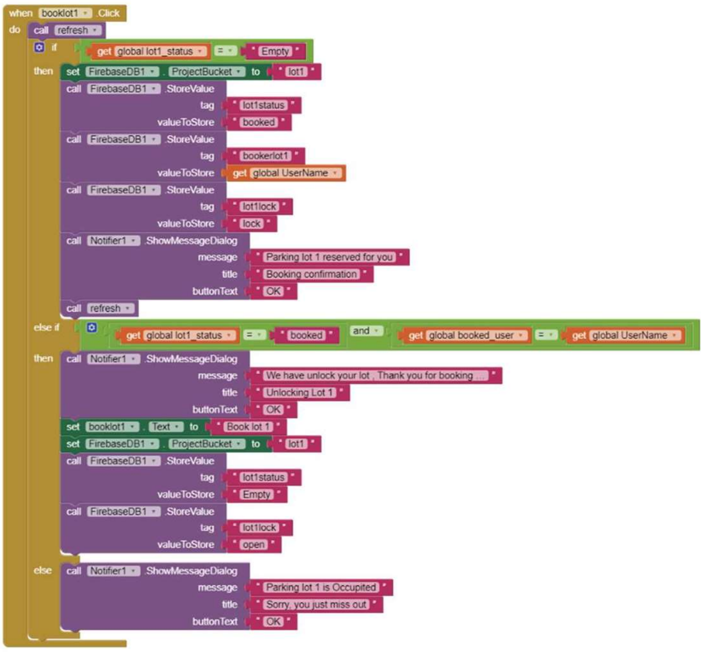
SCREEN 2



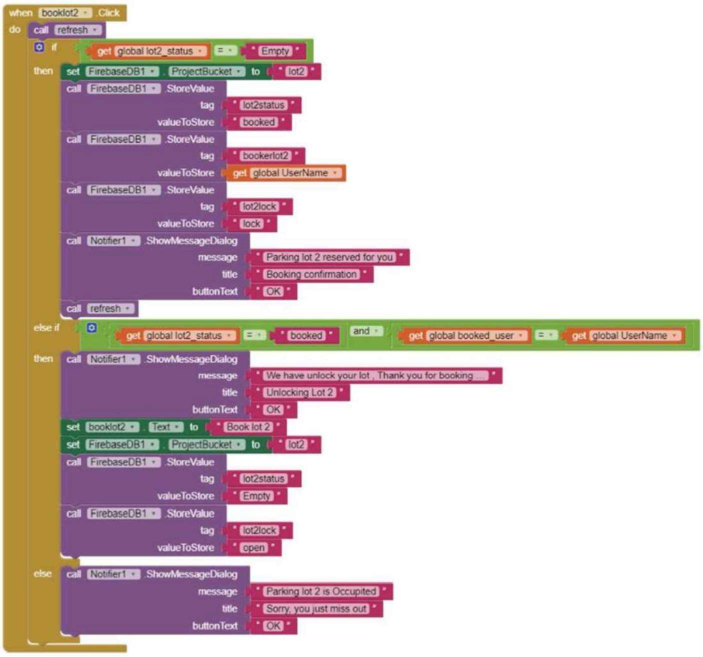
```
initialize global booked_user to " "
initialize global UserName to get start value
initialize global lot1_status to " "
initialize global lot2_status to " "

when Screen2 .Initialize
do  call refresh
    set Clock1 . TimerEnabled to true

when Clock1 .Timer
do  call refresh

to refresh
do  set FirebaseDB1 . ProjectBucket to "lot1"
    call FirebaseDB1 .GetValue
                     tag "lot1status"
       valueIfTagNotThere " "
    call FirebaseDB1 .GetValue
                     tag "bookerlot1"
       valueIfTagNotThere " "
    set FirebaseDB1 . ProjectBucket to "lot2"
    call FirebaseDB1 .GetValue
                     tag "lot2status"
       valueIfTagNotThere " "
    call FirebaseDB1 .GetValue
                     tag "bookerlot2"
       valueIfTagNotThere " "
```



```
when FirebaseDB1 .GotValue
 tag  value
do  if    get tag = "lot1status"
    then  set lot1_status . Text to get value
          set global lot1_status to get value
    else if  get tag = "bookerlot1"
    then  if    get value = get global UserName and get global lot1_status = "booked"
          then  set booklot1 . Text to "Open"
                set global booked_user to get value
          else  set booklot1 . Text to "Book lot 1"
    else if  get tag = "lot2status"
    then  set lot2_status . Text to get value
          set global lot2_status to get value
    else if  get tag = "bookerlot2"
    then  if    get value = get global UserName and get global lot2_status = "booked"
          then  set booklot2 . Text to "Open"
          else  set booklot2 . Text to "Book lot 2"
```

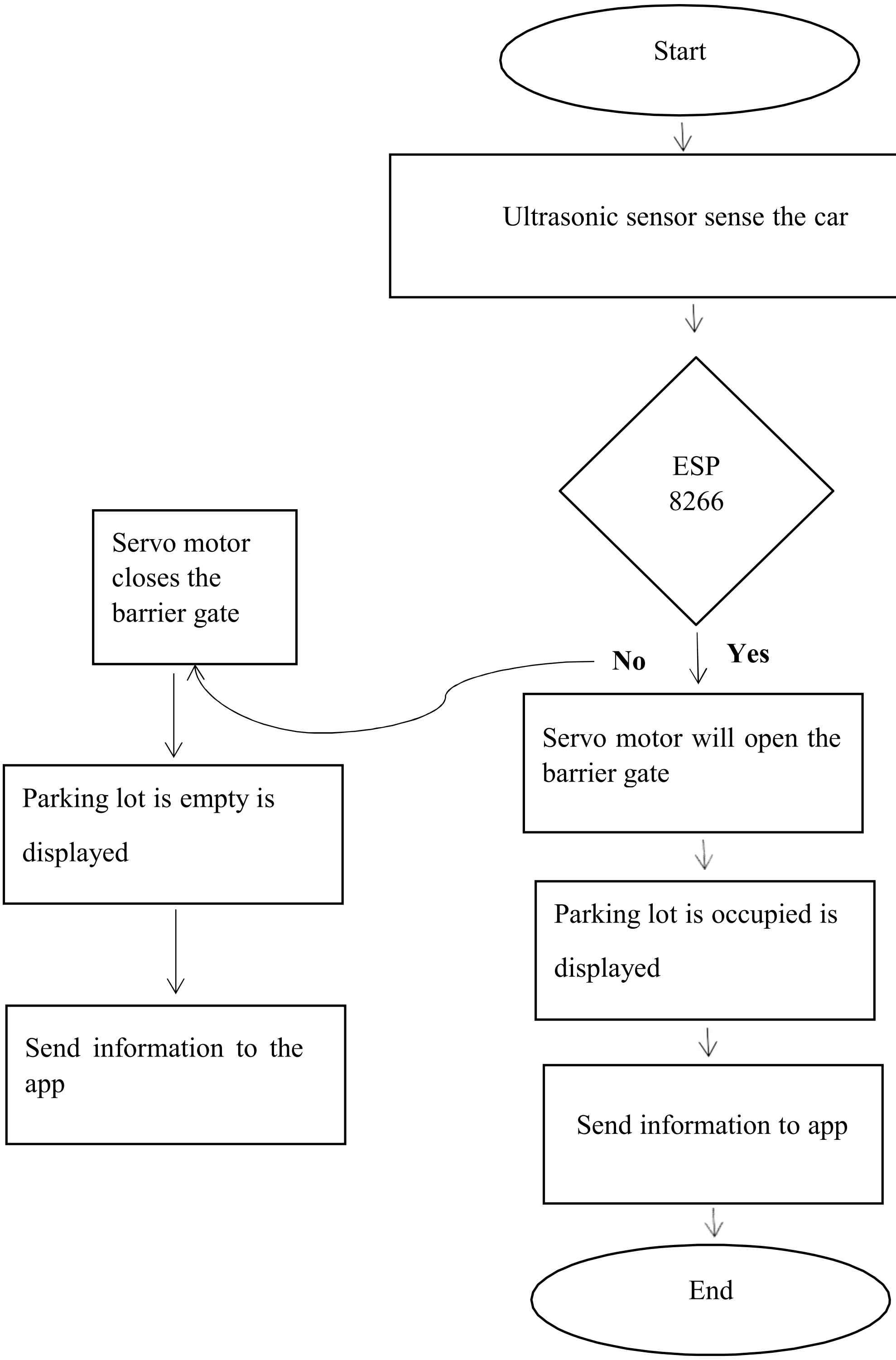


```
when booklot1 .Click
do  call refresh
    if    get global lot1_status = "Empty"
    then  set FirebaseDB1 . ProjectBucket to "lot1"
          call FirebaseDB1 .StoreValue
                          tag "lot1status"
                 valueToStore "booked"
          call FirebaseDB1 .StoreValue
                          tag "bookerlot1"
                 valueToStore get global UserName
          call FirebaseDB1 .StoreValue
                          tag "lot1lock"
                 valueToStore "lock"
          call Notifier1 .ShowMessageDialog
                          message "Parking lot 1 reserved for you"
                            title "Booking confirmation"
                       buttonText "OK"
          call refresh
    else if  get global lot1_status = "booked" and get global booked_user = get global UserName
    then  call Notifier1 .ShowMessageDialog
                          message "We have unlock your lot , Thank you for booking ..."
                            title "Unlocking Lot 1"
                       buttonText "OK"
          set booklot1 . Text to "Book lot 1"
          set FirebaseDB1 . ProjectBucket to "lot1"
          call FirebaseDB1 .StoreValue
                          tag "lot1status"
                 valueToStore "Empty"
          call FirebaseDB1 .StoreValue
                          tag "lot1lock"
                 valueToStore "open"
    else  call Notifier1 .ShowMessageDialog
                          message "Parking lot 1 is Occupited"
                            title "Sorry, you just miss out"
                       buttonText "OK"
```



```
when booklot2 .Click
do  call refresh
    if    get global lot2_status = "Empty"
    then  set FirebaseDB1 . ProjectBucket to "lot2"
          call FirebaseDB1 .StoreValue
                          tag "lot2status"
                 valueToStore "booked"
          call FirebaseDB1 .StoreValue
                          tag "bookerlot2"
                 valueToStore get global UserName
          call FirebaseDB1 .StoreValue
                          tag "lot2lock"
                 valueToStore "lock"
          call Notifier1 .ShowMessageDialog
                          message "Parking lot 2 reserved for you"
                            title "Booking confirmation"
                       buttonText "OK"
          call refresh
    else if  get global lot2_status = "booked" and get global booked_user = get global UserName
    then  call Notifier1 .ShowMessageDialog
                          message "We have unlock your lot , Thank you for booking ..."
                            title "Unlocking Lot 2"
                       buttonText "OK"
          set booklot2 . Text to "Book lot 2"
          set FirebaseDB1 . ProjectBucket to "lot2"
          call FirebaseDB1 .StoreValue
                          tag "lot2status"
                 valueToStore "Empty"
          call FirebaseDB1 .StoreValue
                          tag "lot2lock"
                 valueToStore "open"
    else  call Notifier1 .ShowMessageDialog
                          message "Parking lot 2 is Occupited"
                            title "Sorry, you just miss out"
                       buttonText "OK"
```

3.2.2 Flowchart of the project


Start
Ultrasonic sensor sense the car
ESP 8266
No
Yes
Servo motor closes the barrier gate
Parking lot is empty is displayed
Send information to the app
Servo motor will open the barrier gate
Parking lot is occupied is displayed
Send information to app
End

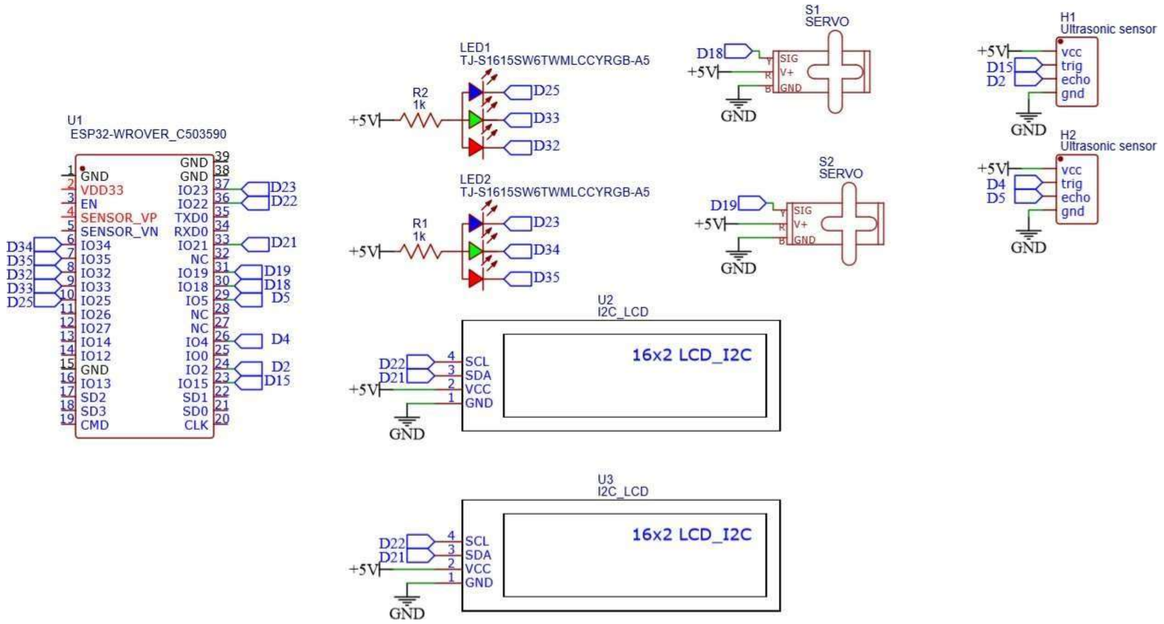
3.2.3 : Project Description

The project is divided into two main parts which will be done simultaneously.Before we created this smart parking system everyone have a major problem when festival times.This is because when festival time there will be lot of people will come to shopping complex.If lot of people come to the shopping complex there will be lot of cars.Sometimes people will took so many hours trying to search for a parking lot.Even sometimes people will give up searching for a parking lot and return to their house.Some even park their vehicles far away from the shopping complex area and will walk to shopping complex.This all happens because the parking system in most of the shopping complex is not well planned and developed.

After we come up with this new developed and well planned parking system people no need to worry about parking their car because before this people does not knew where they have a empty parking lot so they need to search for a empty parking lot until they get.After this,they no need to worry about it because of this brilliant idea.So this is how our new smart parking system works.People just need to download a app in their android or apple device.Then they need to choose which shopping complex they are in.That’s it.Our system will automatically book them a empty parking lot to park their car.To avoid any vehicles from parking in the booked parking lot we designed a mini automated barrier gate.So people no need to worry about their parking lot occupied by other cars because only people who booked for the parking lot can access the mini automated barrier gate.Moreover,this system also can avoid people from wasting their time searching for a empty parking lot especially in festival times.People can be always stress free and go on with their shopping happily.

3.3 Project Hardware

3.3.1 Schematic Circuit

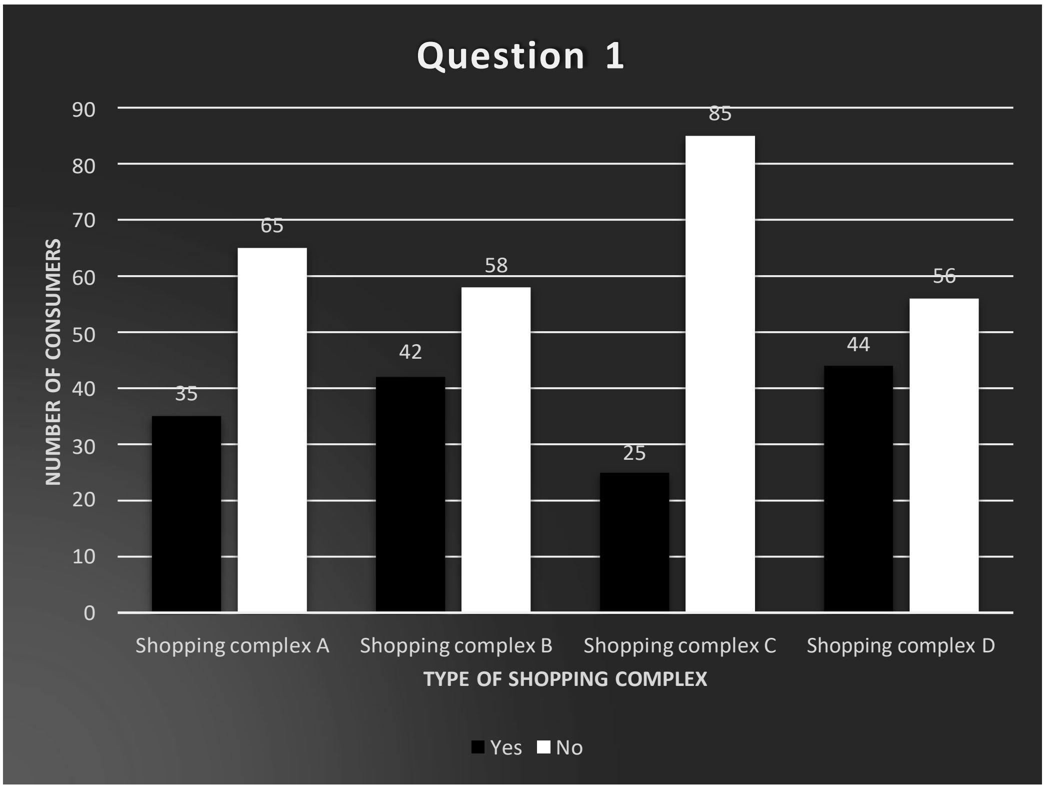


3.4 Summary

The rapid urbanization of the world has made the perception of “smart cities” gain momentum in the global agenda. The change of cities into smart cities brings along an unbelievable occasion for improving citizens’ welfare and nurturing economic progress. The vision to be a smart city has for eternity been a deliberation to all the urban cities. Since a couple of years, projects were in use and thoughts were engaged in many countries to make it into authenticity. Internet of Things stands out to be the obligatory technology implemented along with Cloud Computing. To be a smart city, Smart Parking facility is a crucial service. Preceding technologies were oppressed which proved to be either not efficient or too expensive. The sensors used to notice the vehicle are the essential mechanism. Here, we have engaged Arduino Uno which seemed to be cost efficient with easy setting up and preservation. In future we would develop application for iOS and also with virtual reality and test its workability in a real time atmosphere. We deduce that our future work would make possible parking issues and decline traffic congestion and pollution fashioned by the search for parking. The components worn for implementation of system afford efficient output at a variety of stages of execution. Interfaces recognized between various mechanisms provide an effective announcement across overall functioning of system. Thus, system implementation is well-organized and not compulsory for profitable implementation. In future, certain changes can be included as per necessities of organizations implementing system. Search of free parking slots can be enhanced using Google Search. System is extensive to multi-level, multiple parking areas by making potential changes in hardware group. SMS sent during Android Application can be made safe and sound by applying encryption algorithms. For refuge purpose, Login facility can be provided to users. Along with car number plate recognition facial recognition can also be implemented. And a GPS could be used for instantaneous tracking of available parking area near and the parking position.

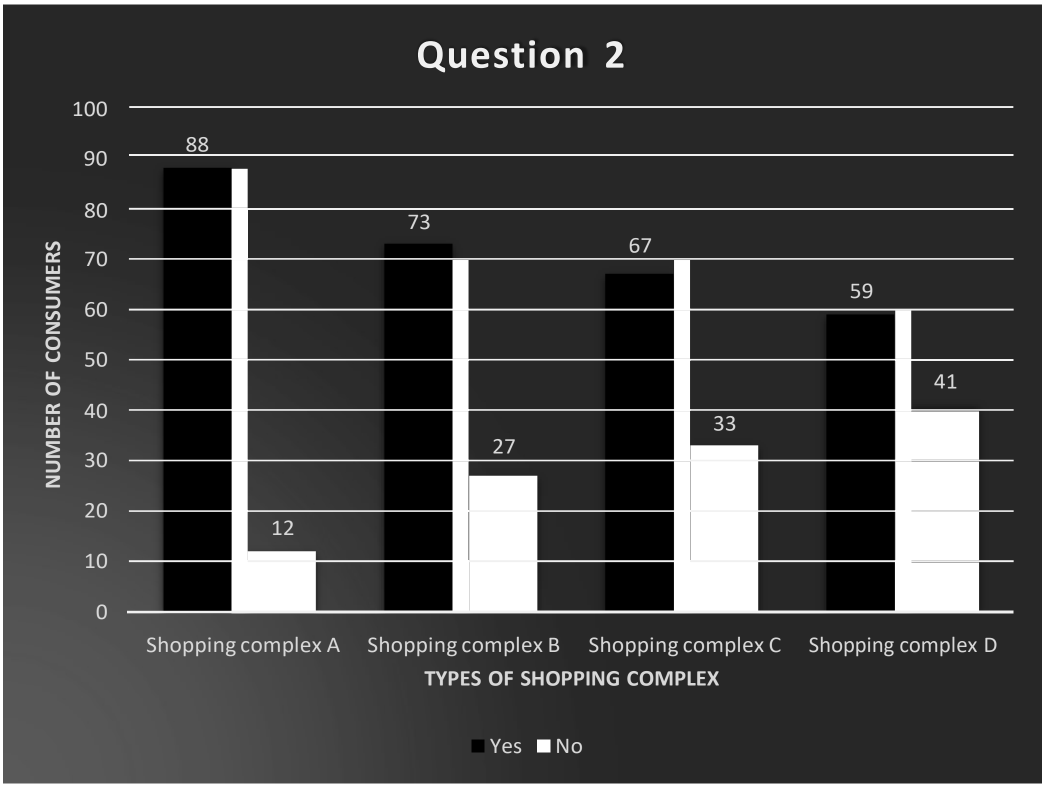
3.5 Survey

Based on the survey it is find out most of consumers said that they doesn’t experience a proper parking system in any shopping complex so far.so here is the prove.



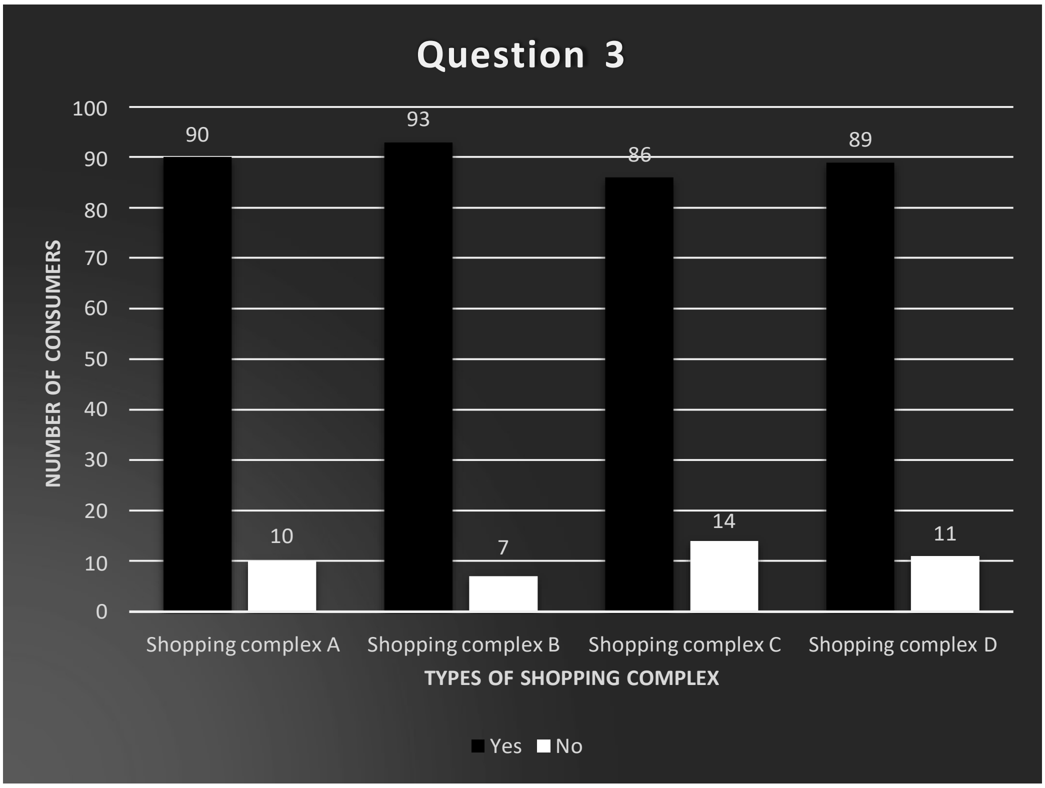
Do you experience a proper parking system when you go for various type of shopping complex?

Based on the 1st question most of the consumers say NO.



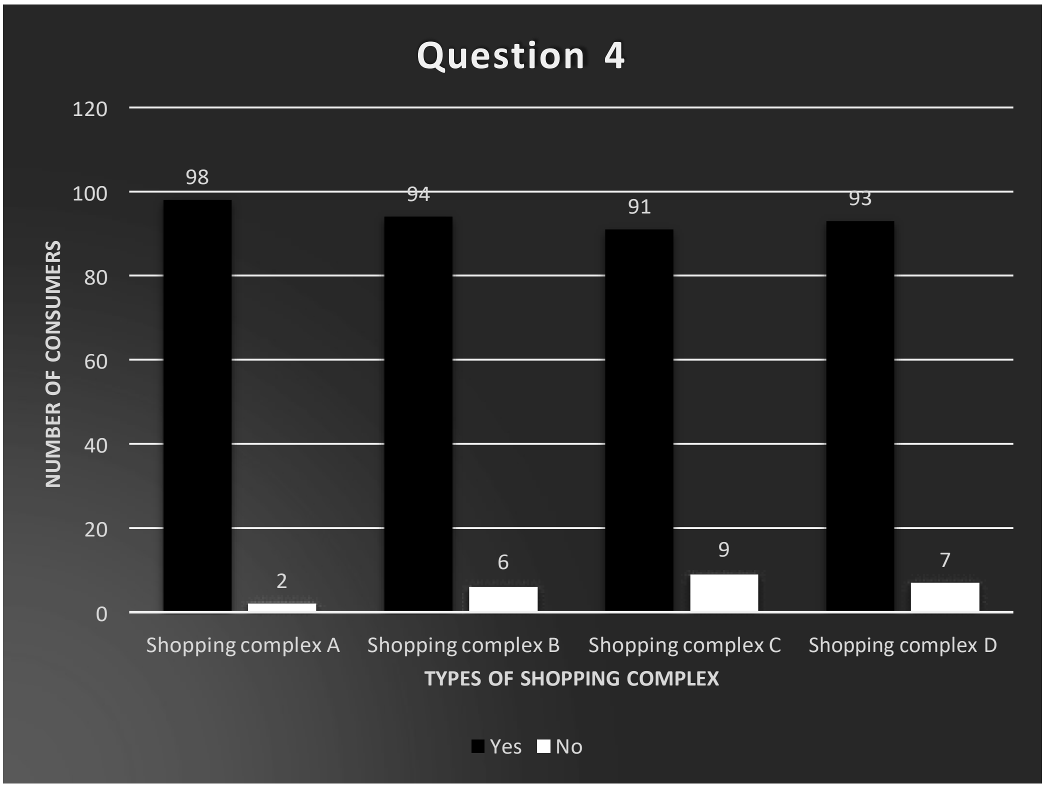
Do you take so many hours on searching for parking space at the shopping complex you go on festival times?

Based on the 2nd question most of the consumers say YES.



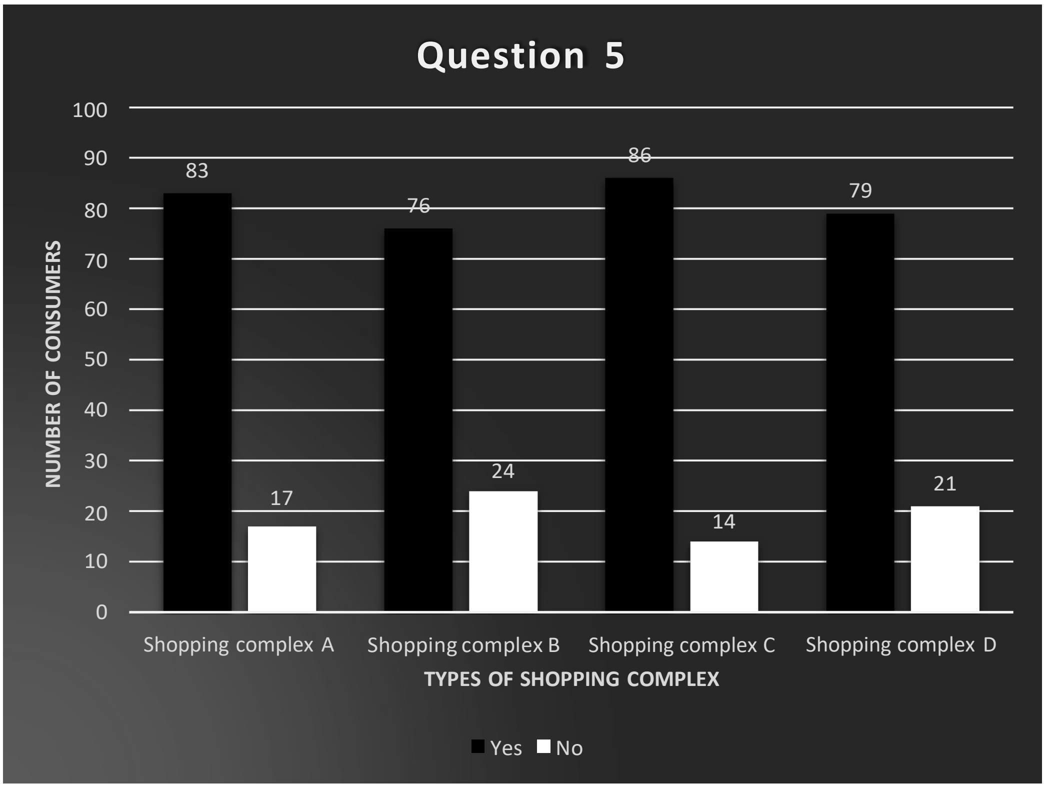
Do you think our new smart parking system will be helpful to you?

Based on the 3rd question most of the consumers say YES.



Do you think our new smart parking system can save your time when go for shopping?

Based on the 4th question most of the consumers say YES.



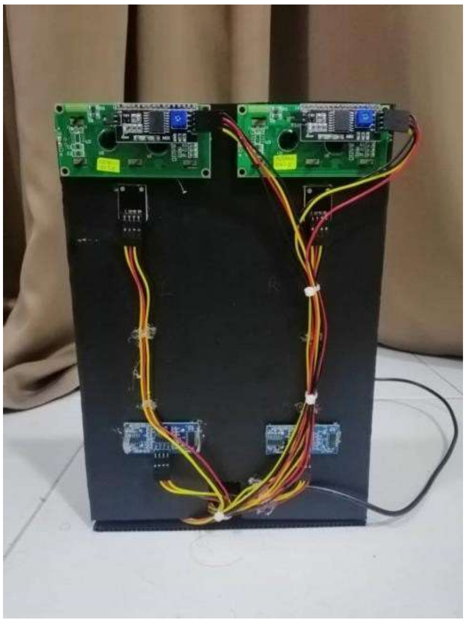
Do you think it will be easy to use a app to book for a parking lot?

Based on the 5th question most of the consumers say YES.

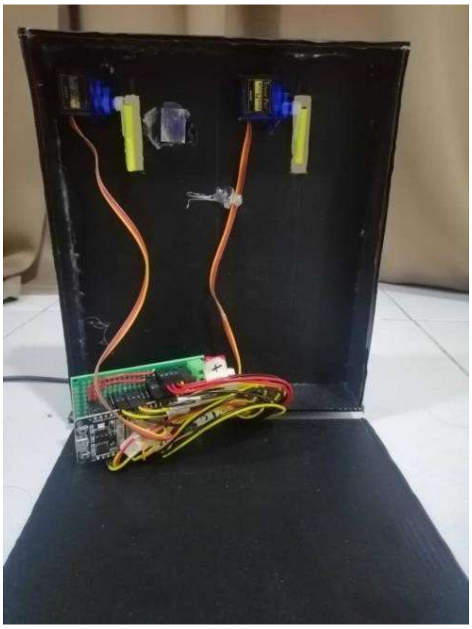
Chapter 4
RESULT AND DISCUSSION

4.1 Introduction

In this part, the chapter will be discussing about the Smart Parking System that chosen for the Final year project. In this discussion, there is few attachment with the pictures that took during the invention the project that we made. Here, the discussion for each and every method that have been used to create the it:



The back view of the smart parking system



The inner view of the smart parking system

4.2 Result and Analysis

For programming ESP8266 NodeMCU, just plug the NodeMCU to Computer with a Micro USB Cable and open Arduino IDE. The libraries are required for I2C Display and Servo Motor. The LCD will display the availability of Parking Spaces and the Servo motors will be used to open and close the Entry and Exit gates. The *Wire.h* library will be used to interface LCD in i2c protocol. The Pins for I2C in the ESP8266 NodeMCU are D1 (SCL) and D2 (SDA). The database here used will be Firebase so here we are also including the library *(FirebaseArduino.h)* for the same.

```
#include <ESP8266WiFi.h>

#include <Servo.h>

#include <LiquidCrystal_I2C.h>

#include <Wire.h>

#include <FirebaseArduino.h>
```

Then include the firebase credentials got from Google Firebase. These will include the Host name containing your project name and a secret key.

```
#define FIREBASE_HOST "https://smartparking-b8563.firebaseio.com/"

#define FIREBASE_AUTH "xoYAu8GkHHOPoQCZuCFvk10UoD1YUdpDVVcHrMuA"
```

Include the Wi-Fi Credentials such as WiFi SSID and Password.

```
#define WIFI_SSID "Hunter"

#define WIFI_PASSWORD "suriya61"
```

Initialise I2C LCD with device address (Here it is 0x27) and type of LCD. Also include the Servo Motors for entry and exit gate.

```
LiquidCrystal_I2C lcd(0x27, 16, 2);

Servo myservo;

Servo myservo1;
```

Start the I2C communication for I2C LCD.

```
Wire.begin(D2, D1);
```

Connect the Entry and Exit Servo Motor to the D5, D6 Pins of the NodeMCU.

```
  myservo.attach(D6);

  myservos.attach(D5);
```

Select the Trigger Pin of Ultrasonic sensor as Output and Echo Pin as Input. The ultrasonic sensor will be used to detect the parking spot availability. If Car has occupied the space then it will glow else it will not glow.

```
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
```

Begin connection with Firebase with Host and Secret Key as credentials.

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

Begin i2c 16x2 LCD and set cursor position at 0th row 0th column.

```
lcd.begin();
lcd.setCursor(0, 0);
```

Take the Distance from Ultrasonic sensor. This will be used to detect the vehicle presence in the particular spot. First send the 2 microsecond pulse and then read the received pulse. Then convert it to the ‘cm’.

```
  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  duration = pulseIn(ECHO, HIGH);
  distance = (duration / 2) / 29.1;
```

Digitally read the IR sensor pin as entry sensor and check if it is high. If it is high then increment entry count and print it to 16x2 LCD display and also to serial monitor.

```
  int carEntry = digitalRead(carEnter);
  if (carEntry == HIGH) {
    countYes++;
    Serial.print("Car Entered = " ); Serial.println(countYes );
    lcd.setCursor(0, 1);
    lcd.print("Car Entered");
```

Also move the servo motor angle to **open entry gate**.

```
    for (pos = 180; pos >= 45; pos -= 1) {
      myservos.write(pos);
      delay(5);
    }
    delay(2000);
    for (pos = 45; pos <= 180; pos += 1) {
      // in steps of 1 degree
      myservos.write(pos);
      delay(5);
    }
```

And send the reading to firebase by using *pushString* function of Firebase library.

```
Firebase.pushString("/Parking Status/", fireAvailable );
```

Do similar steps as above for **Exit IR sensor and Exit servo motor.**

```
  int carExit = digitalRead(carExited);
  if (carExit == HIGH) {
    countYes--;
    Serial.print("Car Exited = " ); Serial.println(countYes);
    lcd.setCursor(0, 1);
    lcd.print("Car Exited");
    for (pos1 = 140; pos1 >= 45; pos1 -= 1) {
      myservo.write(pos1);
      delay(5);
    }
    delay(2000);

    for (pos1 = 45; pos1 <= 180; pos1 += 1) {
      // in steps of 1 degree
      myservo.write(pos1);
      delay(5);
```

```
   }
   Firebase.pushString("/Parking Status/", fireAvailable );
   lcd.clear();
  }
```

Check if the car has come to the parking spot and if it has arrived then glow led giving the signal that the spot is full.

```
  if (distance < 6) {
      Serial.println("Occupied ");
   digitalWrite(led, HIGH);
  }
```

Else show that the spot is available.

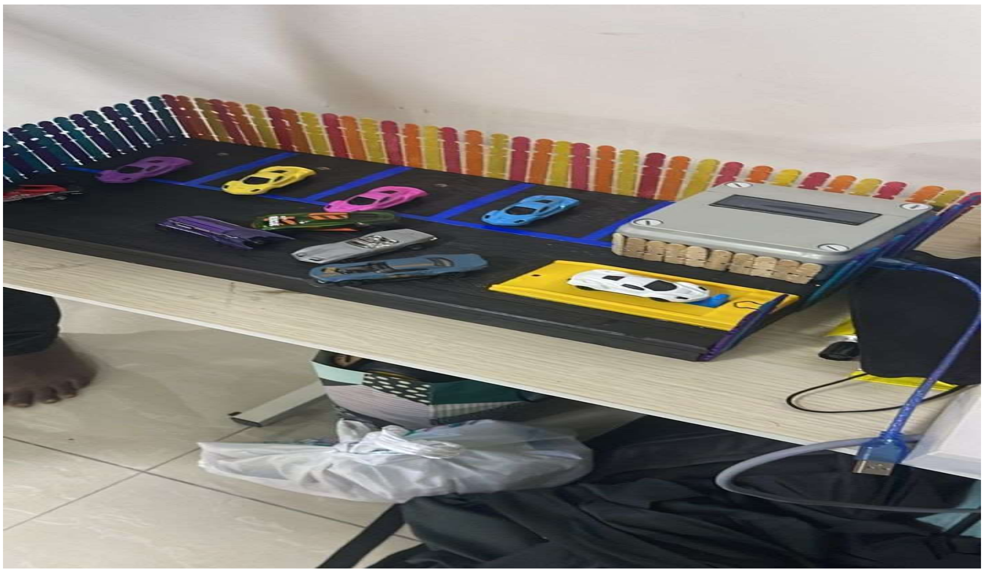
```
  if (distance > 6) {
      Serial.println("Empty ");
   digitalWrite(led, LOW);
  }
```

Calculate the total empty space inside the parking lot and save it in the string to send the data to firebase.

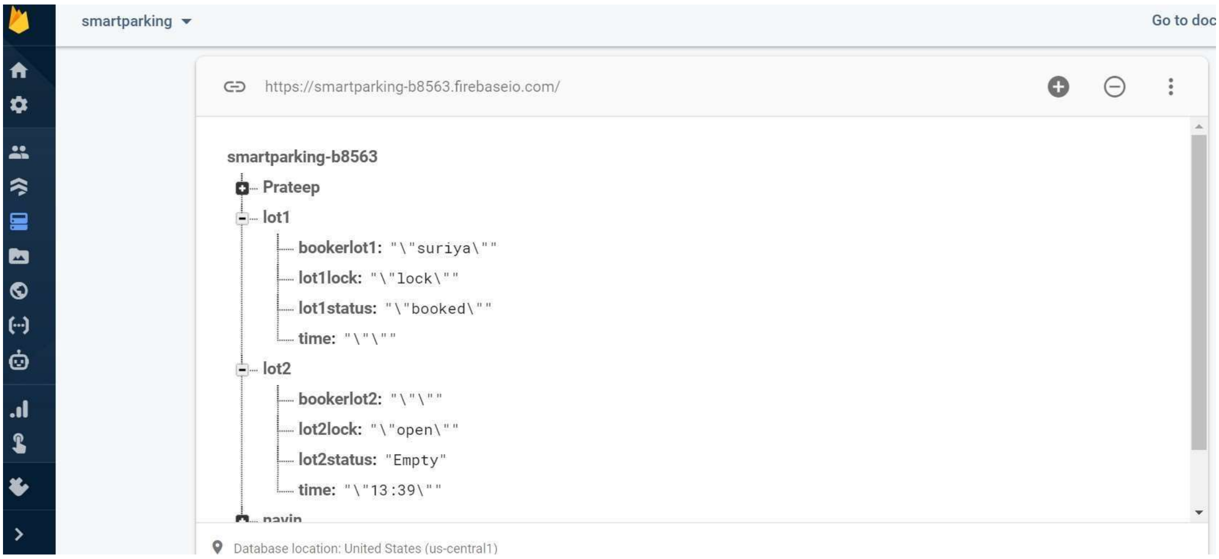
```
Empty = allSpace - countYes;
  Available = String("Available= ") + String(Empty) + String("/") + String(allSpace);
fireAvailable = String("Available=") + String(Empty) + String("/") + String(allSpace);
```

Also print the data to the i2C LCD.

```
  lcd.setCursor(0, 0);
  lcd.print(Available);
```



This is how the **availability of parking can be tracked online on Firebase** as shown in the screenshot below:



```
smartparking-b8563
  Prateep
  lot1
      bookerlot1: "\"suriya\""
      lot1lock: "\"lock\""
      lot1status: "\"booked\""
      time: "\"\""
  lot2
      bookerlot2: "\"\""
      lot2lock: "\"open\""
      lot2status: "Empty"
      time: "\"13:39\""
  navin
```

Database location: United States (us-central1)

This finishes the complete **smart parking system using the ESP8266.**

4.3 DISCUSSION

The demand of smart parking system is increasing significantly. This allows user to involve real time access of the availability of the parking space. The existing system in today’s world doesn’t contains the facilities of parking reservation and parking slot availability checker. The existing system was vision-based monitoring system which estimates the number of the parking slots available in the area by counting the number of incoming and outing cars which consumes lot of time and efforts. The next existing system was sensor-based system which uses ultrasonic sound waves for detecting the presence of vehicles and then two-tier parking came into existence which used the concept of parking cars one above another. The result of the paper is to make the parking area connected with the world as well as reduces time and can be cost effective for the user. The result of this paper is to reduce car theft. This paper reduces overall fuel energy of the vehicle which is consumed in the search of the car.

4.4 SUMMARY

Based on hardwork this report has a complex and diverse theme, so there are some things that could have been done better.The time was very limited but every part is as detailed and specific as it should be. So really would want to realize this concept, still a lot of work needs to be down. However, this document could provide a good basis for further research, design and eventual implementation. In addition it also able to create Final year project and the results is achieve the objective.

Chapter 5
Conclusion and Recommendation

5.1 Introduction

From this Chapter 5, we going to conclude about our project which is Smart Parking System. Next, the summary of this project will be tell throughout this chapter and the further recommendation of this project also will be discuss in this chapter wisely.

5.2 Conclusion

The concept of Smart Cities have always been a dream for humanity. Since the past couple of years large advancements have been made in making smart cities a reality. The growth of Internet of Things and Cloud technologies have give rise to new possibilities in terms of smart cities. Smart parking facilities and traffic management systems have always been at the core of constructing smart cities. In this project, we address the issue of parking and present a smart parking system using a mobile application. The system that we propose provides real time information regarding availability of parking slots in a parking area. Users from remote locations could book a parking slot for them by the use of our mobile application. The efforts made in this paper are indented to improve the parking facilities of a city and thereby aiming to enhance the quality of life of its people.

5.3 Future Recommendations

For future enhancements, this project also upgraded or can be introduced with a GPS.As for the future work the users can book a parking space from a remote location (such as villages). GPS, reservation facilities and license plate scanner can be included in the future. Live gateway enables automated parking meter payment, with zero human supervision.

REFERENCES

1. Supriya Shinde1, AnkitaM Patial2, pSusmedha Chavan3,Sayali Deshmukh4, and Subodh Ingleshwar5 "IOT Based Parking System Using Google", I-SMAC,2017,pp.634-636.

2. HemantChaudhary, PrateekBansal., B.Valarmathi," Advanced CAR Parking System using Arduino", ICACCSS, 2017.

3. Nastaran Reza NazarZadeh, Jennifer C. Dela,"Smart urban parking deducting system" ICSCE, 2016, pp-370-373.

4. PavanKumarJogada and VinayakWarad, "Effective Car Parking Reservation System Based on Internet of things Technologies ".BIJSESC, 2016, Vol. 6, pp.140-142.

5. Prof. Yashomati R. Dhumal1, Harshala A. Waghmare2, Aishwarya S. Tole2, Swati R. Shilimkar2,"Android Based Smart Car Parking System"-IJREEIE, Vol. 5, Issue 3, pp-1371-74,mar-2016.

6. Faiz Ibrahim Shaikh, Pratik NirnayJadhav, Saideep Pradeep Bandarakar" Smart Parking System based on embedded system and sensor Network" IJCA, vol.140.pp.45-51.Apr-2016.

7. RicardGarra, Santi Martinez, and Francesc Seb‟e" A PrivacyPreserving Pay-by-phone Parking system"IEEE-TVT, pp.1-10, Dec2016.

8. Amir O. Kotb, Yao-chunShen, and Yi Huang "Smart parking Guidance, Monitoring and Reservation: A Review," IEEE-ITSM, pp.616.Apr-2017.

9. Ching-FeiYang, You-HueiJu, Chung-Ying Hsieh "Iparking -a real-time parking space monitoring and guiding system", Elsevier, pp.301-305. Apr-2017.

10. Fei-Yue Wang, Liu-Qing Yang, Fellow, Jian Yang," Urban Intelligent Parking system based on Parallel Theory", IEEE-ICNC, 2016.

11. Fei-Yue Wang, Liu-Qing Yang, Fellow, Jian Yang, [2016]," Urban Intelligent Parking system based on Parallel Theory", IEEEComputing, Networking and Communications, Mobile Computing and Vehicle Communications.

12. TarekAlmahdi and chitturivenkatratnum, [2016]"Intelligent automated parking System hacking intimation Features,"IEEE-computing and engineering.

13. Huey-Der Chu, Yong-QuanYeh, Yi-Cheng Lin, Meng-hung Lai, Yi-Jie Lin, [2017]," The Study Intelligent Roadside Park Charging Systems", IEEE- International Conference on Applied System Innovation, pp.1064-67.

14. D.J.Bonde,"Automated car parking systemCommanded by Android application", IEEE Conf., 05-03, Jan 2014.

15. YangengGeng, Christos G. Cassandras," A new „Smart parking‟ system Infrastructure and Implementation ", 1278- 1287 Science Direct, Social and Science behavioural sciences, 2012.

LINKS

1. https://circuitdigest.com/microcontroller-projects/iot-based-smart-parking-system-using-nodemcu

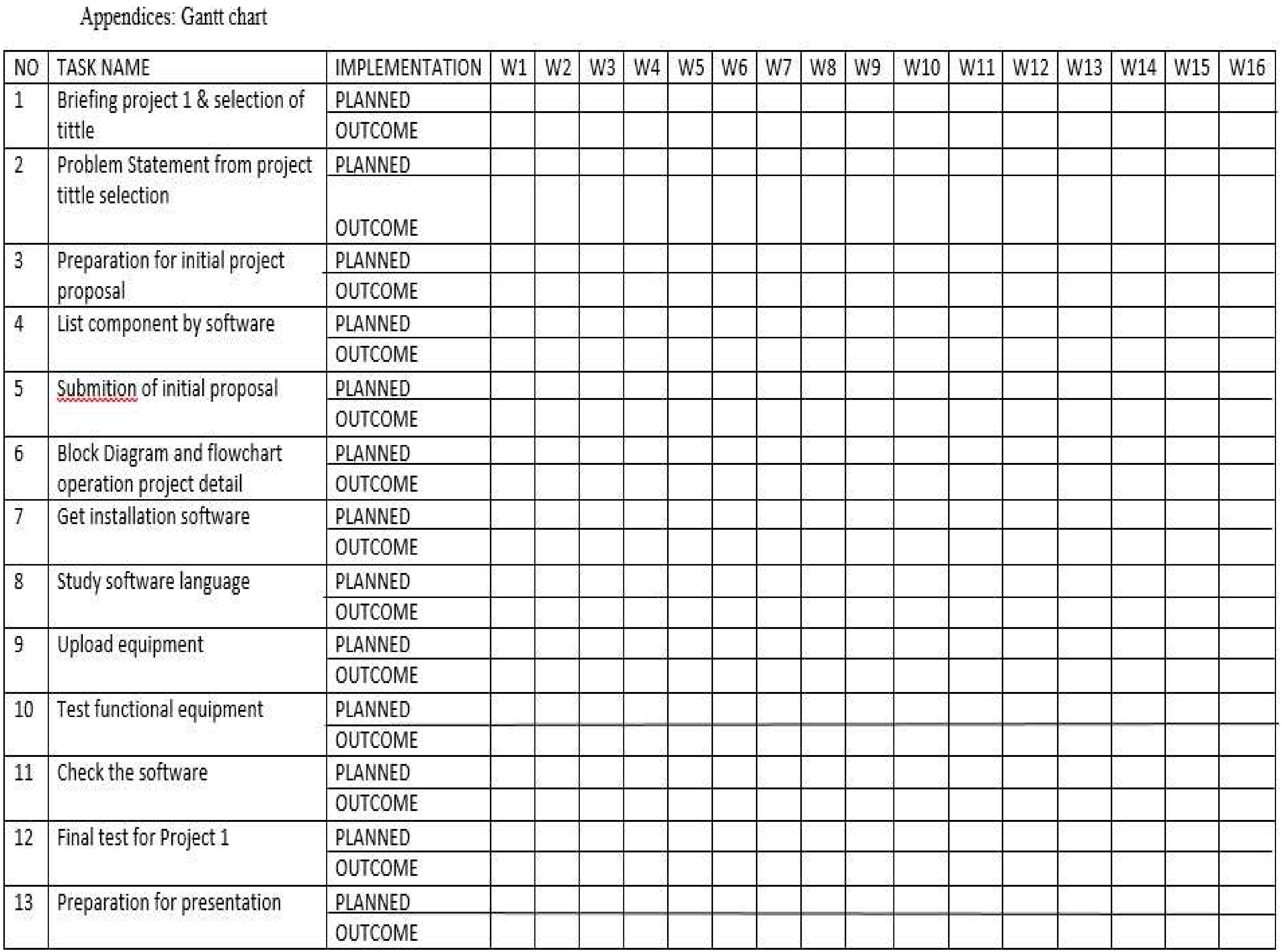
2.https://www.researchgate.net/publication/313667380_Smart_Parking_System_Student_Activity_Project

3.https://www.researchgate.net/publication/320356747_Design_and_Implementation_of_Smart_Car_Parking_System

4.https://www.slideshare.net/slmnsvn/smart-parking-system-66565699

5.https://smartnet.niua.org/sites/default/files/smart_parking_cybercinatics_pvt_ltd.pdf

Appendix 1



Appendices: Gantt chart

NO	TASK NAME	IMPLEMENTATION	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16
1	Briefing project 1 & selection of tittle	PLANNED																
		OUTCOME																
2	Problem Statement from project tittle selection	PLANNED																
		OUTCOME																
3	Preparation for initial project proposal	PLANNED																
		OUTCOME																
4	List component by software	PLANNED																
		OUTCOME																
5	Submition of initial proposal	PLANNED																
		OUTCOME																
6	Block Diagram and flowchart operation project detail	PLANNED																
		OUTCOME																
7	Get installation software	PLANNED																
		OUTCOME																
8	Study software language	PLANNED																
		OUTCOME																
9	Upload equipment	PLANNED																
		OUTCOME																
10	Test functional equipment	PLANNED																
		OUTCOME																
11	Check the software	PLANNED																
		OUTCOME																
12	Final test for Project 1	PLANNED																
		OUTCOME																
13	Preparation for presentation	PLANNED																
		OUTCOME																

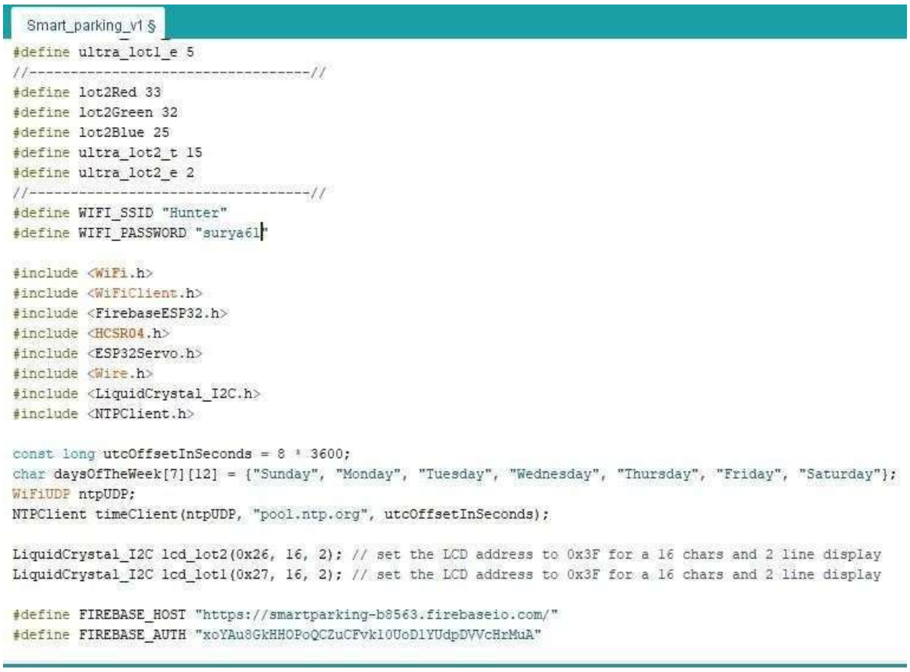
Appendix 2

Project Cost:

The cost of this project is not too expensive it’s just RM545 only that was worth it for us too but the materials and equipments that we need for this project. Which is:

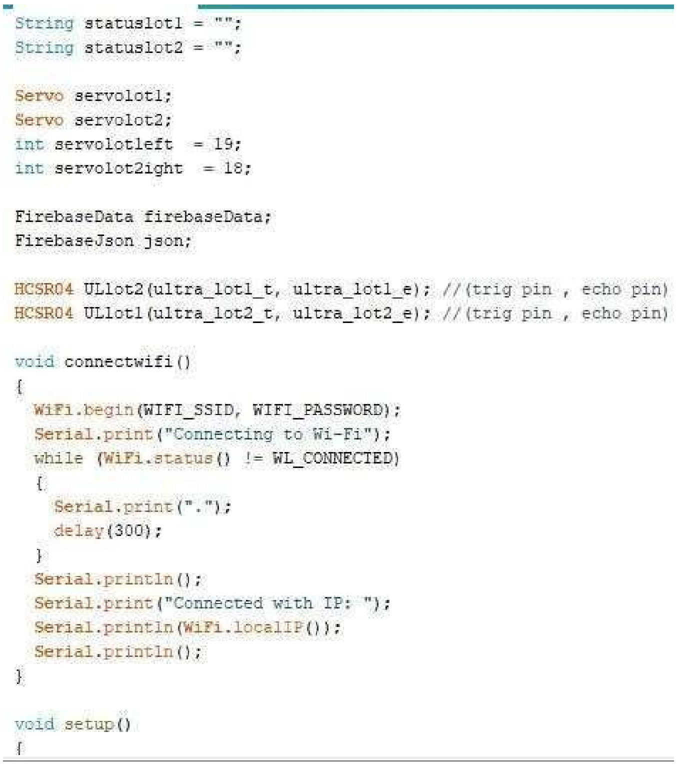
No	**Item**	**Price/ Unit (RM)**	**Quantity**	**Total (RM)**
1	ESP 8266 WIFI MODULE WITH POWER SUPPLY ADAPTER	RM30	1 Set	RM30
2	LED	RM4	2	RM8
3	Ultrasonic sensor	RM8.00	2	16.00
4	SERIAL LCD DISPLAY	RM10	2	RM20
5	SERIAL INTERFACE ADAPTER	RM3	2	RM6.00
6	SERVO MOTOR	RM12	2	RM24
7	Consultation fee	RM 200	1	RM200
8	Equipments: i)Soldering Iron ii)Hot Glue Gun iii)Screw Driver Iv)Bolt and Nuts	RM150	1 SET	RM150
Total Cost				**RM454**
Price (+20%)				**RM544.80**
Price (GST 0%)				**RM544.80**

Appendix 3

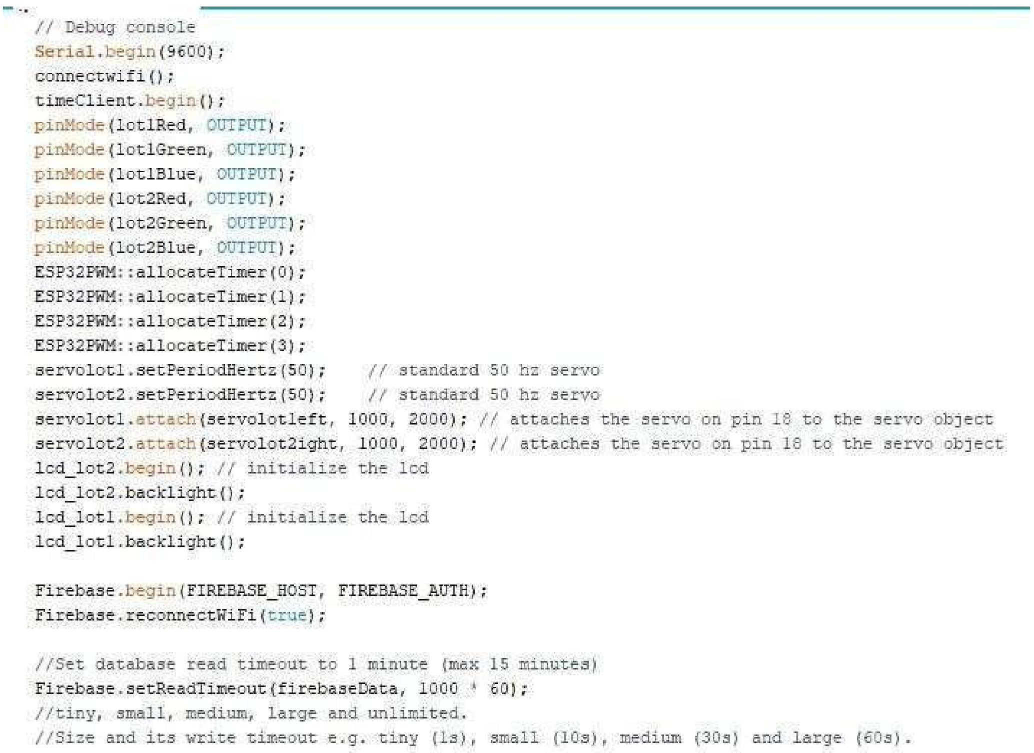


Smart_parking_v1 §

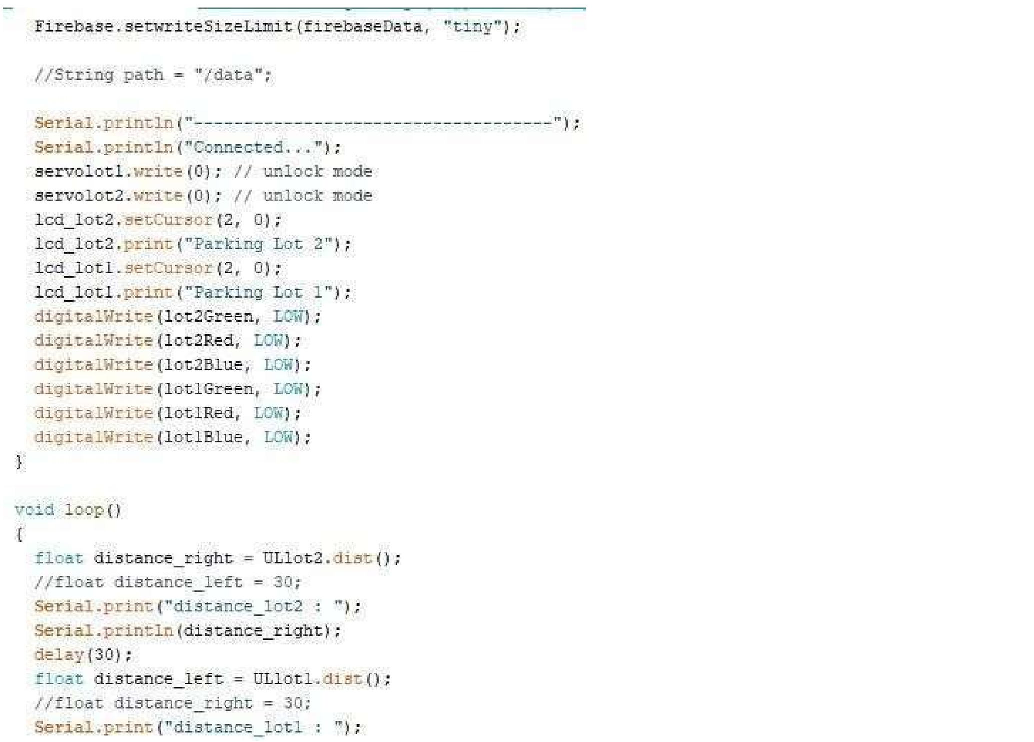
```
#define ultra_lot1_e 5
//----------------------------------//
#define lot2Red 33
#define lot2Green 32
#define lot2Blue 25
#define ultra_lot2_t 15
#define ultra_lot2_e 2
//----------------------------------//
#define WIFI_SSID "Hunter"
#define WIFI_PASSWORD "surya61"

#include <WiFi.h>
#include <WiFiClient.h>
#include <FirebaseESP32.h>
#include <HCSR04.h>
#include <ESP32Servo.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <NTPClient.h>

const long utcOffsetInSeconds = 8 * 3600;
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", utcOffsetInSeconds);

LiquidCrystal_I2C lcd_lot2(0x26, 16, 2); // set the LCD address to 0x3F for a 16 chars and 2 line display
LiquidCrystal_I2C lcd_lot1(0x27, 16, 2); // set the LCD address to 0x3F for a 16 chars and 2 line display

#define FIREBASE_HOST "https://smartparking-b8563.firebaseio.com/"
#define FIREBASE_AUTH "xoYAu8GkHHOPoQCZuCFvk10UoD1YUdpDVVcHrMuA"
```



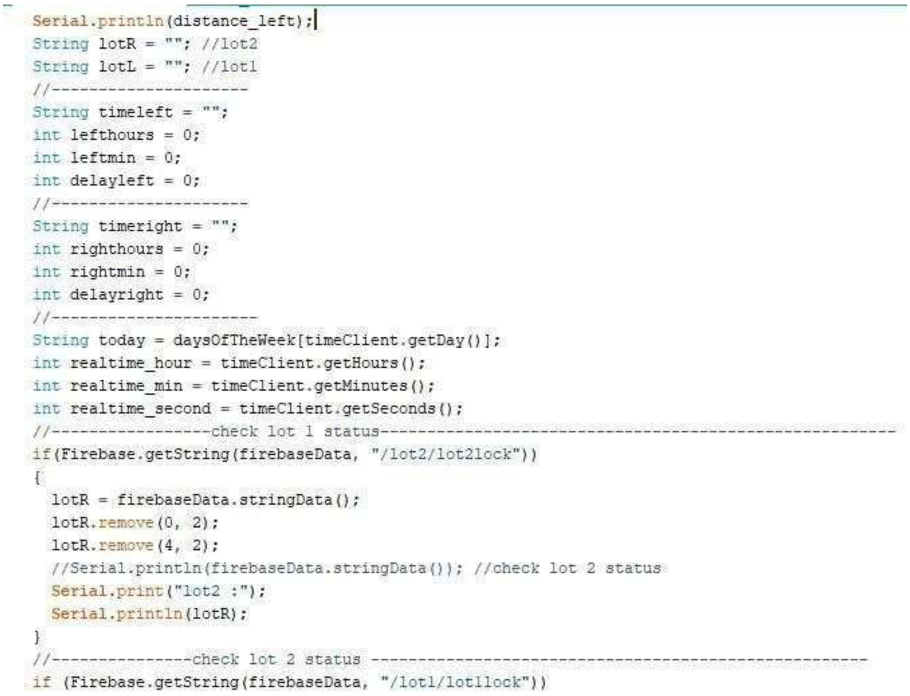
```
String statuslot1 = "";
String statuslot2 = "";

Servo servolot1;
Servo servolot2;
int servolotleft  = 19;
int servolot2ight  = 18;

FirebaseData firebaseData;
FirebaseJson json;

HCSR04 ULlot2(ultra_lot1_t, ultra_lot1_e); //(trig pin , echo pin)
HCSR04 ULlot1(ultra_lot2_t, ultra_lot2_e); //(trig pin , echo pin)

void connectwifi()
{
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();
}

void setup()
{
```



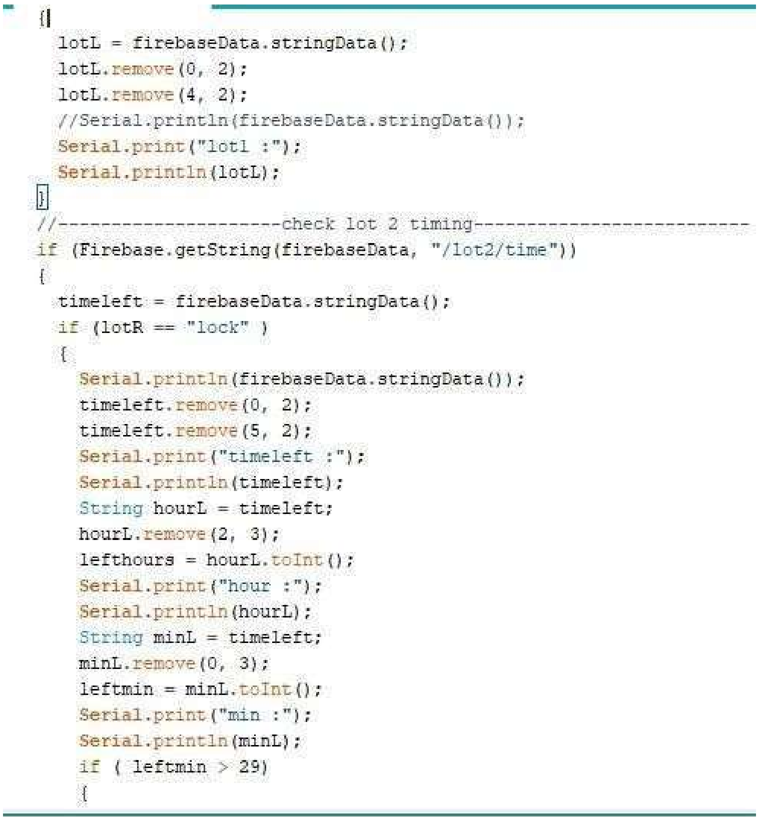
```
  // Debug console
  Serial.begin(9600);
  connectwifi();
  timeClient.begin();
  pinMode(lot1Red, OUTPUT);
  pinMode(lot1Green, OUTPUT);
  pinMode(lot1Blue, OUTPUT);
  pinMode(lot2Red, OUTPUT);
  pinMode(lot2Green, OUTPUT);
  pinMode(lot2Blue, OUTPUT);
  ESP32PWM::allocateTimer(0);
  ESP32PWM::allocateTimer(1);
  ESP32PWM::allocateTimer(2);
  ESP32PWM::allocateTimer(3);
  servolot1.setPeriodHertz(50);    // standard 50 hz servo
  servolot2.setPeriodHertz(50);    // standard 50 hz servo
  servolot1.attach(servolotleft, 1000, 2000); // attaches the servo on pin 18 to the servo object
  servolot2.attach(servolot2ight, 1000, 2000); // attaches the servo on pin 18 to the servo object
  lcd_lot2.begin(); // initialize the lcd
  lcd_lot2.backlight();
  lcd_lot1.begin(); // initialize the lcd
  lcd_lot1.backlight();

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);

  //Set database read timeout to 1 minute (max 15 minutes)
  Firebase.setReadTimeout(firebaseData, 1000 * 60);
  //tiny, small, medium, large and unlimited.
  //Size and its write timeout e.g. tiny (1s), small (10s), medium (30s) and large (60s).
```



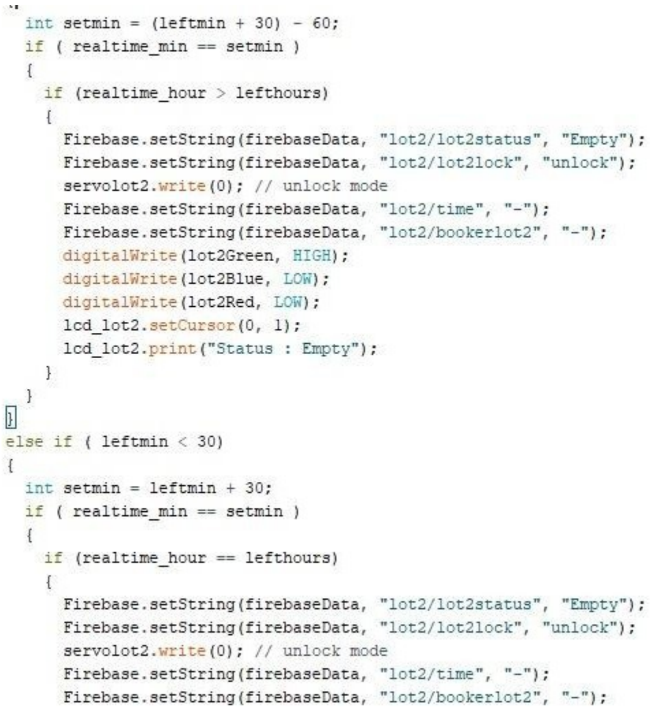
```
  Firebase.setwriteSizeLimit(firebaseData, "tiny");

  //String path = "/data";

  Serial.println("------------------------------------");
  Serial.println("Connected...");
  servolot1.write(0); // unlock mode
  servolot2.write(0); // unlock mode
  lcd_lot2.setCursor(2, 0);
  lcd_lot2.print("Parking Lot 2");
  lcd_lot1.setCursor(2, 0);
  lcd_lot1.print("Parking Lot 1");
  digitalWrite(lot2Green, LOW);
  digitalWrite(lot2Red, LOW);
  digitalWrite(lot2Blue, LOW);
  digitalWrite(lot1Green, LOW);
  digitalWrite(lot1Red, LOW);
  digitalWrite(lot1Blue, LOW);
}

void loop()
{
  float distance_right = ULlot2.dist();
  //float distance_left = 30;
  Serial.print("distance_lot2 : ");
  Serial.println(distance_right);
  delay(30);
  float distance_left = ULlot1.dist();
  //float distance_right = 30;
  Serial.print("distance_lot1 : ");
```



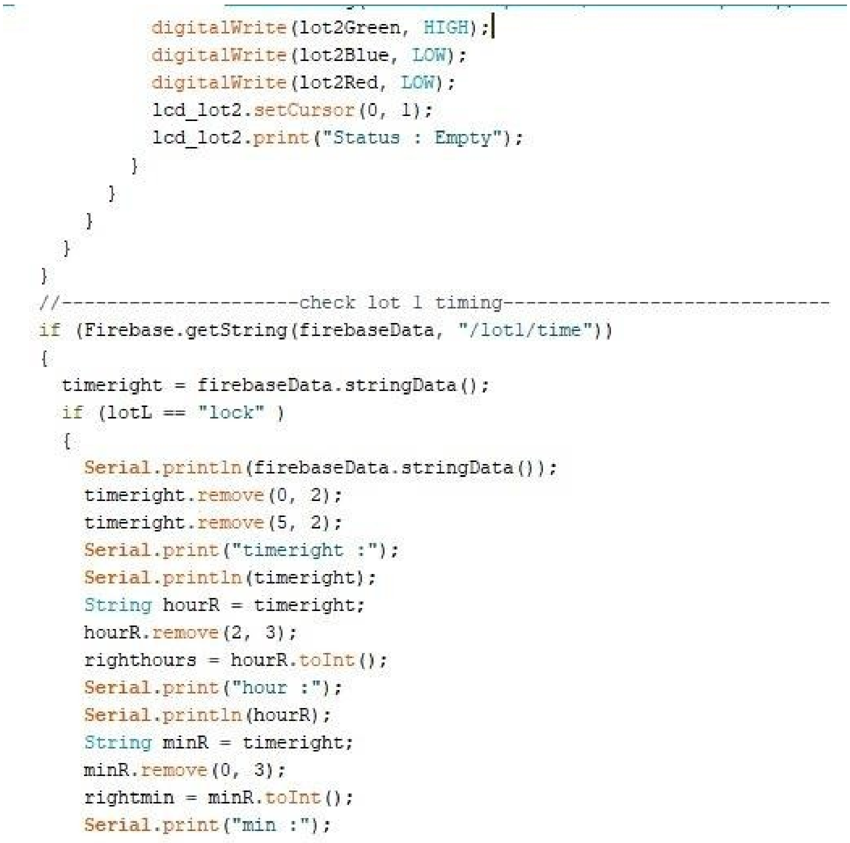
```
  Serial.println(distance_left);
  String lotR = ""; //lot2
  String lotL = ""; //lot1
  //---------------------
  String timeleft = "";
  int lefthours = 0;
  int leftmin = 0;
  int delayleft = 0;
  //---------------------
  String timeright = "";
  int righthours = 0;
  int rightmin = 0;
  int delayright = 0;
  //----------------------
  String today = daysOfTheWeek[timeClient.getDay()];
  int realtime_hour = timeClient.getHours();
  int realtime_min = timeClient.getMinutes();
  int realtime_second = timeClient.getSeconds();
  //-----------------check lot 1 status-------------------------------------------------------
  if(Firebase.getString(firebaseData, "/lot2/lot2lock"))
  {
    lotR = firebaseData.stringData();
    lotR.remove(0, 2);
    lotR.remove(4, 2);
    //Serial.println(firebaseData.stringData()); //check lot 2 status
    Serial.print("lot2 :");
    Serial.println(lotR);
  }
  //---------------check lot 2 status -----------------------------------------------------
  if (Firebase.getString(firebaseData, "/lot1/lot1lock"))
```



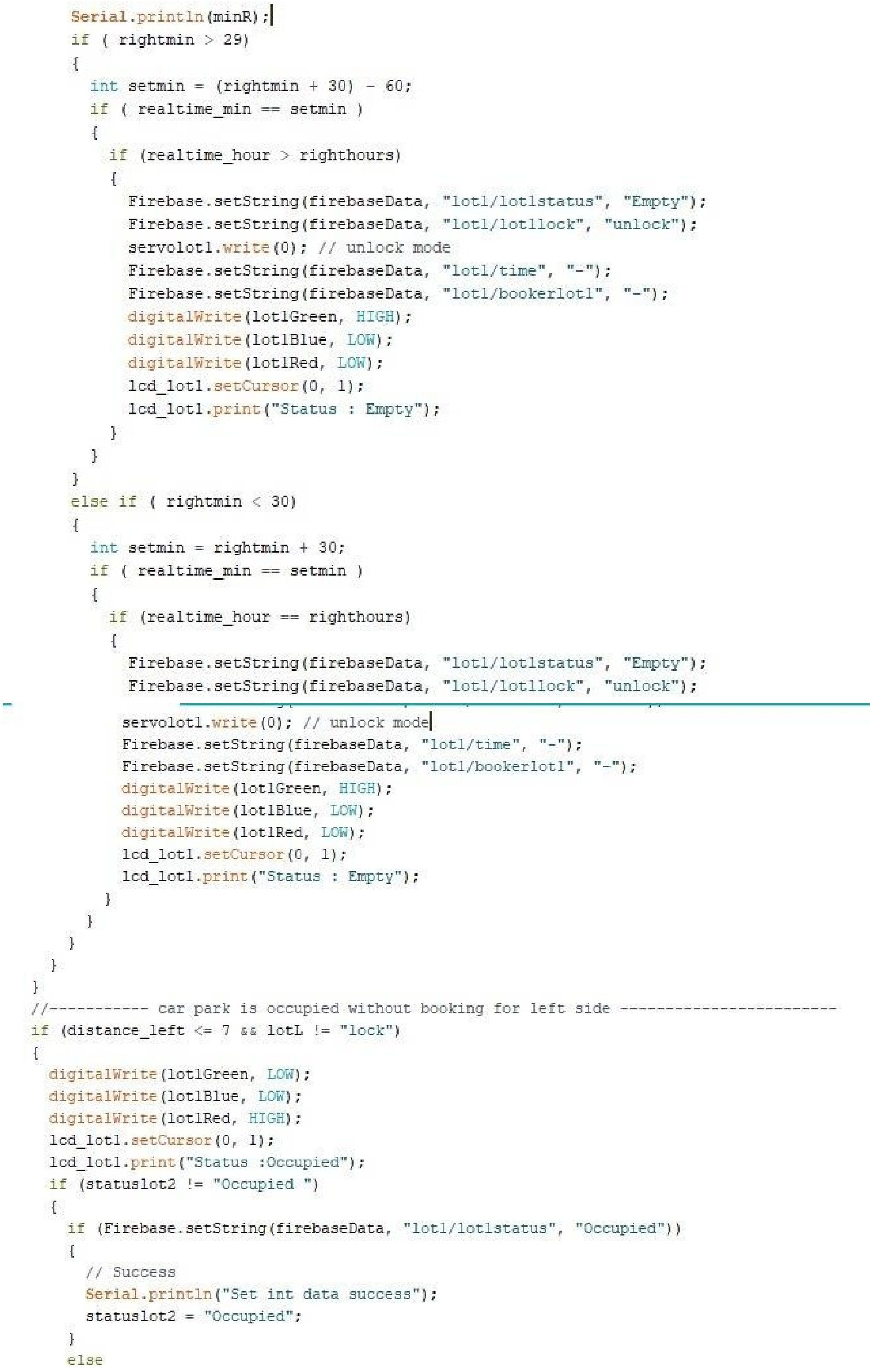
```
  {
    lotL = firebaseData.stringData();
    lotL.remove(0, 2);
    lotL.remove(4, 2);
    //Serial.println(firebaseData.stringData());
    Serial.print("lot1 :");
    Serial.println(lotL);
  }
  //---------------------check lot 2 timing--------------------------
  if (Firebase.getString(firebaseData, "/lot2/time"))
  {
    timeleft = firebaseData.stringData();
    if (lotR == "lock" )
    {
      Serial.println(firebaseData.stringData());
      timeleft.remove(0, 2);
      timeleft.remove(5, 2);
      Serial.print("timeleft :");
      Serial.println(timeleft);
      String hourL = timeleft;
      hourL.remove(2, 3);
      lefthours = hourL.toInt();
      Serial.print("hour :");
      Serial.println(hourL);
      String minL = timeleft;
      minL.remove(0, 3);
      leftmin = minL.toInt();
      Serial.print("min :");
      Serial.println(minL);
      if ( leftmin > 29)
      {
```



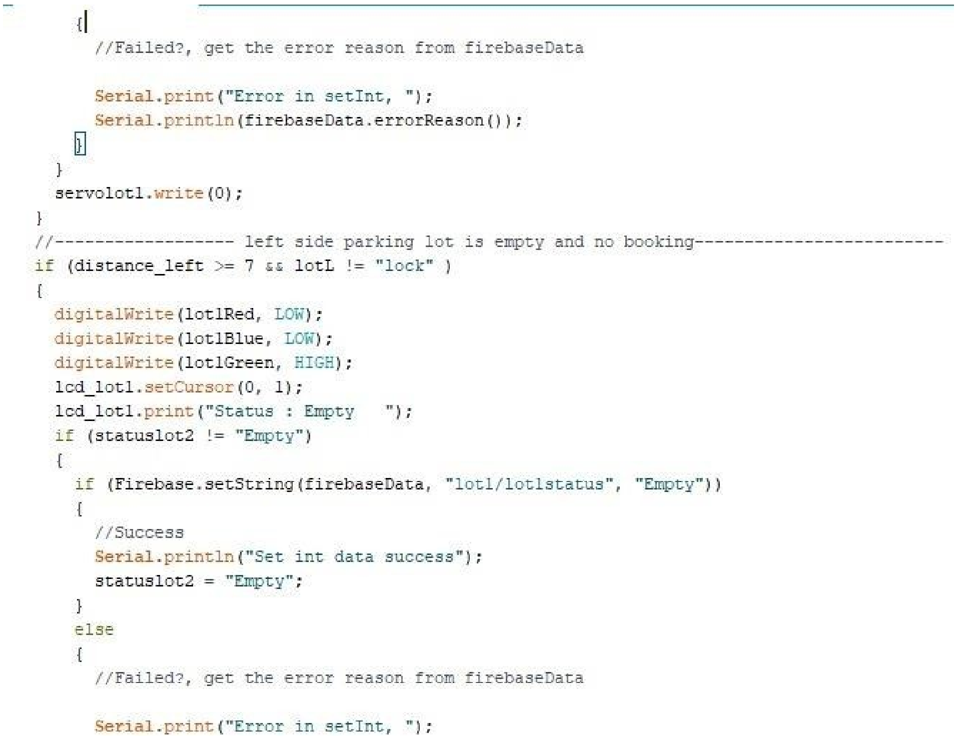
```
      int setmin = (leftmin + 30) - 60;
      if ( realtime_min == setmin )
      {
        if (realtime_hour > lefthours)
        {
          Firebase.setString(firebaseData, "lot2/lot2status", "Empty");
          Firebase.setString(firebaseData, "lot2/lot2lock", "unlock");
          servolot2.write(0); // unlock mode
          Firebase.setString(firebaseData, "lot2/time", "-");
          Firebase.setString(firebaseData, "lot2/bookerlot2", "-");
          digitalWrite(lot2Green, HIGH);
          digitalWrite(lot2Blue, LOW);
          digitalWrite(lot2Red, LOW);
          lcd_lot2.setCursor(0, 1);
          lcd_lot2.print("Status : Empty");
        }
      }
    }
    else if ( leftmin < 30)
    {
      int setmin = leftmin + 30;
      if ( realtime_min == setmin )
      {
        if (realtime_hour == lefthours)
        {
          Firebase.setString(firebaseData, "lot2/lot2status", "Empty");
          Firebase.setString(firebaseData, "lot2/lot2lock", "unlock");
          servolot2.write(0); // unlock mode
          Firebase.setString(firebaseData, "lot2/time", "-");
          Firebase.setString(firebaseData, "lot2/bookerlot2", "-");
```



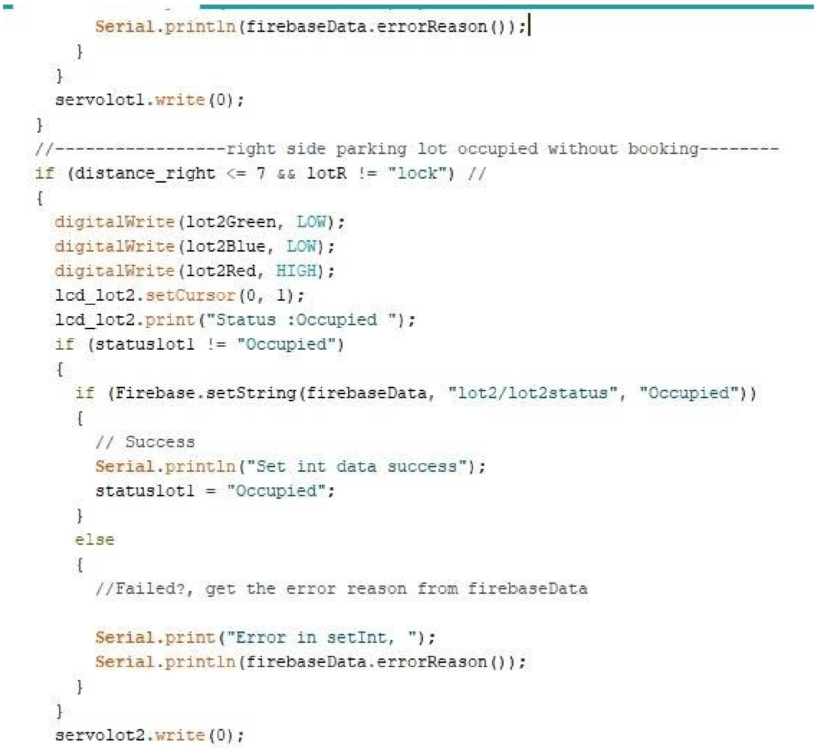
```
          digitalWrite(lot2Green, HIGH);
          digitalWrite(lot2Blue, LOW);
          digitalWrite(lot2Red, LOW);
          lcd_lot2.setCursor(0, 1);
          lcd_lot2.print("Status : Empty");
        }
      }
    }
  }
}
//---------------------check lot 1 timing----------------------------
if (Firebase.getString(firebaseData, "/lot1/time"))
{
  timeright = firebaseData.stringData();
  if (lotL == "lock" )
  {
    Serial.println(firebaseData.stringData());
    timeright.remove(0, 2);
    timeright.remove(5, 2);
    Serial.print("timeright :");
    Serial.println(timeright);
    String hourR = timeright;
    hourR.remove(2, 3);
    righthours = hourR.toInt();
    Serial.print("hour :");
    Serial.println(hourR);
    String minR = timeright;
    minR.remove(0, 3);
    rightmin = minR.toInt();
    Serial.print("min :");
```



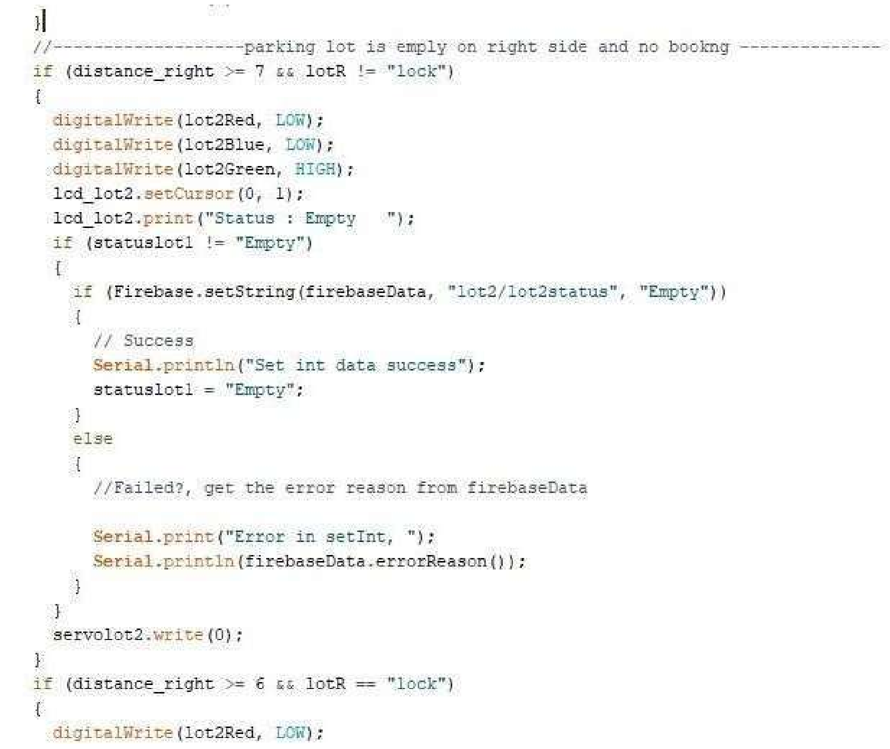
```
    Serial.println(minR);
    if ( rightmin > 29)
    {
      int setmin = (rightmin + 30) - 60;
      if ( realtime_min == setmin )
      {
        if (realtime_hour > righthours)
        {
          Firebase.setString(firebaseData, "lot1/lot1status", "Empty");
          Firebase.setString(firebaseData, "lot1/lot1lock", "unlock");
          servolot1.write(0); // unlock mode
          Firebase.setString(firebaseData, "lot1/time", "-");
          Firebase.setString(firebaseData, "lot1/bookerlot1", "-");
          digitalWrite(lot1Green, HIGH);
          digitalWrite(lot1Blue, LOW);
          digitalWrite(lot1Red, LOW);
          lcd_lot1.setCursor(0, 1);
          lcd_lot1.print("Status : Empty");
        }
      }
    }
    else if ( rightmin < 30)
    {
      int setmin = rightmin + 30;
      if ( realtime_min == setmin )
      {
        if (realtime_hour == righthours)
        {
          Firebase.setString(firebaseData, "lot1/lot1status", "Empty");
          Firebase.setString(firebaseData, "lot1/lot1lock", "unlock");
          servolot1.write(0); // unlock mode
          Firebase.setString(firebaseData, "lot1/time", "-");
          Firebase.setString(firebaseData, "lot1/bookerlot1", "-");
          digitalWrite(lot1Green, HIGH);
          digitalWrite(lot1Blue, LOW);
          digitalWrite(lot1Red, LOW);
          lcd_lot1.setCursor(0, 1);
          lcd_lot1.print("Status : Empty");
        }
      }
    }
  }
}
//----------- car park is occupied without booking for left side -----------------------
if (distance_left <= 7 && lotL != "lock")
{
  digitalWrite(lot1Green, LOW);
  digitalWrite(lot1Blue, LOW);
  digitalWrite(lot1Red, HIGH);
  lcd_lot1.setCursor(0, 1);
  lcd_lot1.print("Status :Occupied");
  if (statuslot2 != "Occupied ")
  {
    if (Firebase.setString(firebaseData, "lot1/lot1status", "Occupied"))
    {
      // Success
      Serial.println("Set int data success");
      statuslot2 = "Occupied";
    }
    else
```



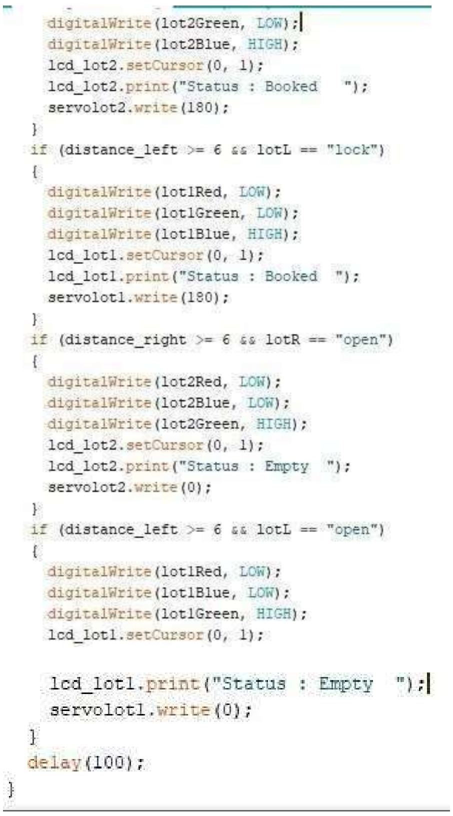
```
      {
        //Failed?, get the error reason from firebaseData

        Serial.print("Error in setInt, ");
        Serial.println(firebaseData.errorReason());
      }
    }
    servolot1.write(0);
  }
  //------------------ left side parking lot is empty and no booking------------------------
  if (distance_left >= 7 && lotL != "lock" )
  {
    digitalWrite(lot1Red, LOW);
    digitalWrite(lot1Blue, LOW);
    digitalWrite(lot1Green, HIGH);
    lcd_lot1.setCursor(0, 1);
    lcd_lot1.print("Status : Empty    ");
    if (statuslot2 != "Empty")
    {
      if (Firebase.setString(firebaseData, "lot1/lot1status", "Empty"))
      {
        //Success
        Serial.println("Set int data success");
        statuslot2 = "Empty";
      }
      else
      {
        //Failed?, get the error reason from firebaseData

        Serial.print("Error in setInt, ");
```



```
        Serial.println(firebaseData.errorReason());
      }
    }
    servolot1.write(0);
  }
  //-----------------right side parking lot occupied without booking--------
  if (distance_right <= 7 && lotR != "lock") //
  {
    digitalWrite(lot2Green, LOW);
    digitalWrite(lot2Blue, LOW);
    digitalWrite(lot2Red, HIGH);
    lcd_lot2.setCursor(0, 1);
    lcd_lot2.print("Status :Occupied ");
    if (statuslot1 != "Occupied")
    {
      if (Firebase.setString(firebaseData, "lot2/lot2status", "Occupied"))
      {
        // Success
        Serial.println("Set int data success");
        statuslot1 = "Occupied";
      }
      else
      {
        //Failed?, get the error reason from firebaseData

        Serial.print("Error in setInt, ");
        Serial.println(firebaseData.errorReason());
      }
    }
    servolot2.write(0);
```



```
  }
  //-------------------parking lot is emply on right side and no bookng --------------
  if (distance_right >= 7 && lotR != "lock")
  {
    digitalWrite(lot2Red, LOW);
    digitalWrite(lot2Blue, LOW);
    digitalWrite(lot2Green, HIGH);
    lcd_lot2.setCursor(0, 1);
    lcd_lot2.print("Status : Empty   ");
    if (statuslot1 != "Empty")
    {
      if (Firebase.setString(firebaseData, "lot2/lot2status", "Empty"))
      {
        // Success
        Serial.println("Set int data success");
        statuslot1 = "Empty";
      }
      else
      {
        //Failed?, get the error reason from firebaseData

        Serial.print("Error in setInt, ");
        Serial.println(firebaseData.errorReason());
      }
    }
    servolot2.write(0);
  }
  if (distance_right >= 6 && lotR == "lock")
  {
    digitalWrite(lot2Red, LOW);
```



```
    digitalWrite(lot2Green, LOW);
    digitalWrite(lot2Blue, HIGH);
    lcd_lot2.setCursor(0, 1);
    lcd_lot2.print("Status : Booked   ");
    servolot2.write(180);
  }
  if (distance_left >= 6 && lotL == "lock")
  {
    digitalWrite(lot1Red, LOW);
    digitalWrite(lot1Green, LOW);
    digitalWrite(lot1Blue, HIGH);
    lcd_lot1.setCursor(0, 1);
    lcd_lot1.print("Status : Booked  ");
    servolot1.write(180);
  }
  if (distance_right >= 6 && lotR == "open")
  {
    digitalWrite(lot2Red, LOW);
    digitalWrite(lot2Blue, LOW);
    digitalWrite(lot2Green, HIGH);
    lcd_lot2.setCursor(0, 1);
    lcd_lot2.print("Status : Empty  ");
    servolot2.write(0);
  }
  if (distance_left >= 6 && lotL == "open")
  {
    digitalWrite(lot1Red, LOW);
    digitalWrite(lot1Blue, LOW);
    digitalWrite(lot1Green, HIGH);
    lcd_lot1.setCursor(0, 1);

    lcd_lot1.print("Status : Empty   ");
    servolot1.write(0);
  }
  delay(100);
}
```