ANDROID-BASED CAR PARKING MONITORING SYSTEM

(ACPMS)

MOHD IZZAT SYAHMI BIN SAIFUL AZMAN

CA15010

BACHELOR OF COMPUTER SCIENCE

(COMPUTER SYSTEM & NETWORKING)

WITH HONOUR

UNIVERSITI MALAYSIA PAHANG

JANUARY 2019

ANDROID-BASED CAR PARKING MONITORING SYSTEM (ACPMS)

MOHD IZZAT SYAHMI BIN SAIFUL AZMAN
CA15010

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Computer System & Networking)

Faculty of Computer System & Software Engineering

UNIVERSITI MALAYSIA PAHANG

JANUARY 2019

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	: MOHD IZZAT SYAHMI BIN SAIFUL AZMAN
Date of Birth	: 09 DEC 1993
Title	: ANDROID-BASED CAR PARKING MONITORING SYSTEM
Academic Session	: 2018/2019

I declare that this thesis is classified as:

□	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
□	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

_______________________ (Student's Signature)

_______________________ (Supervisor's Signature)

____
New IC/Passport Number
Date:

____
Name of Supervisor
Date:

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang,
Lebuhraya Tun Razak,
26300, Gambang, Kuantan.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Thank you.

Yours faithfully,

Author's Name
Thesis Title

Reasons (i)

(ii)

(iii)

____
(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

SUPERVISOR’S DECLARATION

I Dr. Nor Bakiah Binti Abd Warif hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Computer System & Networking)

____

(Supervisor’s Signature)

Full Name : NOR BAKIAH BINTI ABD WARIF

Position : Lecturer

Date : 12 DECEMBER 2018

UMP Universiti Malaysia PAHANG
Engineering • Technology • Creativity

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student's Signature)

Full Name : MOHD IZZAT SYAHMI BIN SAIFUL AZMAN

ID Number : CA15010

Date : 12 DECEMBER 2018

ACKNOWLEDGEMENTS

All praises to Allah S.W.T because of the strength given and his bless as I am able to complete this project. I would like to express my sincere appreciation to the individuals who have been involved in helping me throughout completing of the project.

My appreciation and thanks goes to my supervisor, Dr. Nor Bakiah Binti Abd Warif for all the guidance, assistance and positive comment in completing the proposal. My sincere thanks also goes to the lecturer of my faculty who willing to help me directly or indirectly to finish this research paper. I would also like to express my appreciation to my parent, En. Saiful Azman Bin Dollah and Pn. Jamilah Binti Senafi, also my siblings for all the moral support, motivation and inspiration they have been given all this while.

Last but not least my appreciation goes to individuals that were contributed throughout completing of this project paper.

ABSTRAK

Peningkatan bilangan kenderaan di jalan raya hari ini boleh menyebabkan lalu lintas menjadi sesak kerana kenderaan adalah keperluan asas yang digunakan oleh semua orang. Memandangkan jumlah pertumbuhan penggunaan kenderaan, keperluan tempat meletak kereta juga akan menjadi terhad dan akan menjadi masalah bagi pengguna mencari ruang letak kereta terutama di pusat membeli-belah. Teknologi sebagai Internet of Thing (IoT) adalah sistem peranti pengkomputeran yang saling berkaitan dengan keupayaan untuk memindahkan data melalui rangkaian tanpa memerlukan interaksi manusia-ke-manusia atau manusia-ke-komputer yang digunakan dalam hampir setiap jenis bidang dalam masyarakat hari ini. Dalam konteks karya ini, aplikasi mudah alih untuk pengguna dibangunkan untuk memudahkan dalam mencari tempat parkir yang khusus. Sistem Android-based Car Parking Monitoring System mampu menyediakan pengguna untuk mengkaji semula ruang letak kereta yang ada dan mencari tempat letak kereta terdekat dari lokasi pengguna semasa dengan sensor yang dilengkapi untuk mengesan kereta masuk dan keluar dari ruang letak kereta di pusat membeli-belah. Oleh itu, dengan skrin paparan yang mudah dan mesra pengguna yang dibentangkan, pengguna boleh memahami dan menggunakan aplikasi mudah alih ini untuk mendapatkan ruang letak kereta. Akhirnya, kajian ini menerokai penggunaan teknologi Internet of Thing dalam persekitaran sebenar untuk tujuan pengesanan gerakan dan perkhidmatan lokasi untuk memberitahu pengguna yang telah digambarkan dalam aplikasi mudah alih.

ABSTRACT

The increasing number of vehicles on the road today can cause of traffic jammed since vehicle is a basic necessity used by all people. As the growth number of vehicles usage, the needs of car parking lot also will be limited and will be a problem for user to find a parking space especially in shopping mall. Technologies as Internet of Thing (IoT) is a system of interrelated computing devices with ability to transfer data over a network without requiring human-to-human or human-to-computer interaction being are utilized in almost every kind of fields in today society. In the context of this work, a mobile application for user is developed to facilitate in finding a specific parking space. Android-based Car Parking Monitoring System is capable to provide a user to review the available parking space and find the nearest parking from current user location with sensor equipped to detect incoming and outgoing cars from parking space in the mall. Therefore, with the simple and user-friendly interface presented, user can easily understand and use this mobile application to explore the parking space. Finally, this study explored the used of Internet of Thing technology in real environment for the purpose of movement detection and location service to notify the user that has been visualized in mobile application.

TABLE OF CONTENT

DECLARATION

TITLE PAGE

ACKNOWLEDGEMENTS **iv**

ABSTRAK **v**

ABSTRACT **vi**

TABLE OF CONTENT **vii**

LIST OF TABLES **x**

LIST OF FIGURES **xi**

CHAPTER 1 INTRODUCTION **1**

1.1 Introduction 1

1.2 Problem Statement 2

1.3 Objective 2

1.4 Scope 3

1.5 Thesis Organization 3

CHAPTER 2 LITERATURE REVIEW **4**

2.1 Introduction 4

2.2 Overview of the Hardware for ACPMS 5

2.2.1 Hardware 5

2.3 Review of Related Systems 7

2.3.1 Park Smart 7

2.3.2 Automation of Real Time Car Parking System 9

2.3.3 Low Cost Wireless Parking 11

2.4 Comparison between Related System 13

CHAPTER 3 METHODOLOGY 14

3.1 Introduction 14

3.2 Rapid Application Development (RAD) 15

3.3 Preliminary Design 16

3.3.1 System Description 17

3.3.2 Printed System Description 18

3.3.3 Context Diagram 19

3.3.4 Data Flow Diagram (DFD) Level 0 20

3.3.5 Use Case Diagram 21

3.3.6 Flowchart Diagram 22

3.4 Hardware & Software 25

3.4.1 Hardware Requirement 25

3.4.2 Software Requirement 26

3.5 Gantt Chart 27

3.6 Conclusion 29

CHAPTER 4 RESULTS AND DISCUSSION 30

4.1 Introduction 30

4.2 Implementation Requirement 31

4.2.1 Hardware Implementation 31

4.2.2 000webhost PhpMyAdmin 36

4.2.3 ACPMS Mobile Application Development 38

4.3 Testing 41

4.3.1 Testing Report 42

4.4 Conclusion 46

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS OF FUTURE WORKS 47

5.1 Introduction 47

5.2 Research Constraint 48

5.2.1 Development Constraint 48

5.2.2 System Constraint 48

5.3 Future Work 49

REFERENCES 50

LIST OF TABLES

Table 2.1 Comparison between related system. 13
Table 3.1 Hardware Requirement 25
Table 3.2 Software Requirement 26

LIST OF FIGURES

Figure 2.1 Cytron ESP8266 WiFi Shield 5

Figure 2.2 Ultrasonic Sensor 5

Figure 2.3 Arduino UNO 6

Figure 2.4 Example of how Park Smart system. 7

Figure 2.5 Illustration of hardware and software in the system. 8

Figure 2.6 LED display in Wireless parking system. 11

Figure 2.7 Sensor that has been implement in this system. 11

Figure 3.1 Phases in the James Martin approach to RAD 16

Figure 3.2 Flow of the system. 17

Figure 3.3 Arduino UNO connect with ultrasonic sensor 18

Figure 3.4 Arduino UNO connect with Wi-Fi shield. 18

Figure 3.5 Context Diagram for ACPMS. 19

Figure 3.6 Data Flow Diagram (DFD) Level 0 for. 20

Figure 3.7 Use case diagram for ACPMS. 21

Figure 3.8 Flowchart for ACPMS. 22

Figure 3.9 Flowchart for ACPMS. 23

Figure 3.10 Gantt chart for whole project 27

Figure 3.11 Gantt chart for requirement and planning phase during PSM 1 27

Figure 3.12 Gantt chart for design phase during PSM 1 28

Figure 3.13 Gantt chart for PSM 2 28

Figure 4.1 ACPMS kits 32

Figure 4.2 Arduino IDE Windows Application 33

Figure 4.3 Functions for void setup program 34

Figure 4.4 Loop functions 35

Figure 4.5 000webhost PhpMyAdmin page 36

Figure 4.6 000webhost for write_data.php 37

Figure 4.7 000webhost for data.php 37

Figure 4.8 Parking view page for ACPMS system development 38

Figure 4.9 Pop-up message when “Nearest Parking” button is clicked 39

Figure 4.10 On Off GPS page on the user android 39

Figure 4.11 Colour guide page for ACPMS system development 40

Figure 4.12 Testing Process 41

CHAPTER 1

INTRODUCTION

1.1 Introduction

In this sophisticated era, the sharp rises in vehicle usage by the local in Malaysia are very high. This is because of the increasing of population especially in the cities because today the vehicle is a necessity (Parking, Event, & Network, 2018). This problem also led to difficulties for car parking hence the number of parking requirements are not in line with the increasing in the number of vehicles, especially cars.

Internet of Thing (IoT) is a platform for human to make things easy for their daily life using the ability to transfer data over a network. Study has been made that IoT is one of the most rapidly growth technology that used network physical device that embedded with any electronics, software, sensors and network connectivity which enables these objects to connect and exchange data (Friess, 2018).

In order to make it easier for users to find a parking space, the concept of Android-based Car Parking Monitoring System (ACPMS) has been research to be implement. This system is based on real time concept which is users just need to access the application via smartphone then view the number of available parking space in the specific mall. ACPMS needs a different type of equipment such as Arduino Uno, ultra-sonic sensor, assorted wired and Mobile apps.

1.2 Problem Statement

The difficulty in finding car parking space is particularly significant in shopping malls where it is difficult for user to figure out whether parking is available or not. They need to find parking space from one level to another, especially on weekends and public holidays which is malls usually packed with people. As the result, user 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 problem can also increase stress and frustrated then it can lead to bad mood for them.

Current parking system only provides information for available parking space at the main entrance parking gate. User still have the difficulty to find an available parking due to the in and out parking is update from time to time. So, the available parking that user see at the main entrance are not sure will be available or not when the user arrives at their destination.

The other current parking system is using GSM that will give information when the users send SMS to the system to get a new assigned parking space. This will consume extra bill charge each time user use the system and consume more time for SMS to process sending and receiving the message compare to current technology.

1.3 Objective

The main objective of the design is to build application that can show available mall parking space in real-time for the user.

Specific Objective:

i. To study the current limitation of parking availability system.
ii. To design a prototype device for user to monitor parking space that available via mobile phone.
iii. To evaluate the prototype of the proposed system.

1.4 Scope

The following is the scope of this research:

I. The system for indoor and outdoor parking space.
II. Only can be use when connected to any internet.
III. Android only.

1.5 Thesis Organization

This report for Arduino-Based Car Parking Monitoring System consists of five chapters. Chapter 1 contain introduction of the project, problem statement, objective and scope of the project. Chapter 2 will discuss about literature review that will compare and describe about the existing system. Chapter 3 contain methodology and flow of the system. Chapter 4 will describe on implementation, testing and result. The final chapter which is chapter 5 will conclude the entire project of Arduino-based car parking monitoring system future works about the project.

CHAPTER 2

LITERATURE REVIEW

This chapter will further discussion on the related past studies and information that make significant contributions in this area of study ACPMS or closely related system.

2.1 Introduction

ACPMS is an Arduino based system that will ease the human efforts in find parking space from a level to another level in mall. Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs light on a sensor, or a Twitter message and turn it into an output, activating a motor, turning on an LED, publishing something online. In this project, Arduino UNO will be used. Android-based parking systems are improved and gain it efficiency by using Cytron ESP8266 Wi-Fi Shield and Ultrasonic Sensor. Cytron ESP8266 Wi-Fi Shield allows an Arduino Board to connect to the Internet. It is based on the Wiznet W5500 Ethernet chip. The shield come pre-fabricate with ESP8266-12 WROOM-02 module which offers Wi-Fi connection to Arduino board.

2.2 Overview of the Hardware for ACPMS

In this section will be discussed on what hardware will be used in this project. Below are the list of the hardware.

2.2.1 Hardware

a) Cytron ESP8266 WiFi Shield

The Arduino Wi-Fi Shield is the component that will allow this system to connect to the internet. This Wi-Fi shield is choose because it compatible with Arduino Uno (ArduinoDevelopers, 2017).

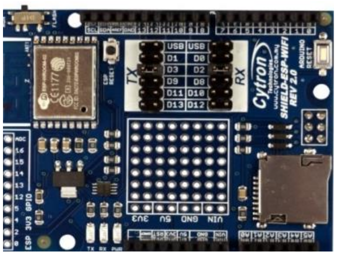


Figure 2.1 Cytron ESP8266 WiFi Shield

b) Ultrasonic Sensor

An Ultrasonic sensor is used to measure the distance of an object by using sound waves. It can measure the distance from 2cm to 400 cm (Page, 2018). This sensor is chosen to measure the distance between floor and the car.

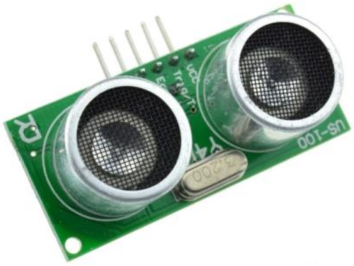


Figure 2.2 Ultrasonic Sensor

c) Arduino UNO

Arduino UNO are also known as Arduino UNO big brother that’s have the capabilities four Arduino UNO boards combined (Uno et al., 2018). Besides, the cost of this Arduino is inexpensive compare to another Arduino. Arduino Uno has simple design and it can run a program at a time.

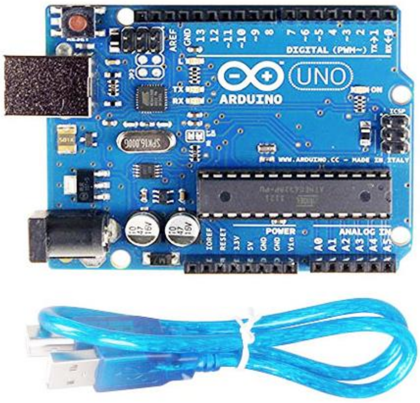


Figure 2.3 Arduino UNO

2.3 Review of Related Systems

There are three existing system that will be mention and the comparison between these three different systems will be discuss. The three of existing system are Park Smart, Park Here and Wireless Parking Module.

2.3.1 Park Smart

Park Smart is a system which aim to solve the pain of finding a free parking space in public and private areas for example cities, malls. Hence to optimize parking stalls allocation as well as to increase revenues for the companies which manage the parking system. The proposed solution exploits cutting edge technologies such as IoT, Cloud Computing and Deep Learning. The main component for this system is cameras which acquire the stream using dedicated embedded systems that will send the results to the main server system. Its use wide angle cameras to optimize the number of parking spaces monitored. The mobile app or browsers are the ending point for the people who are looking for a free spot to park their vehicle. Figure 2.4 is an overview of Park Smart system.

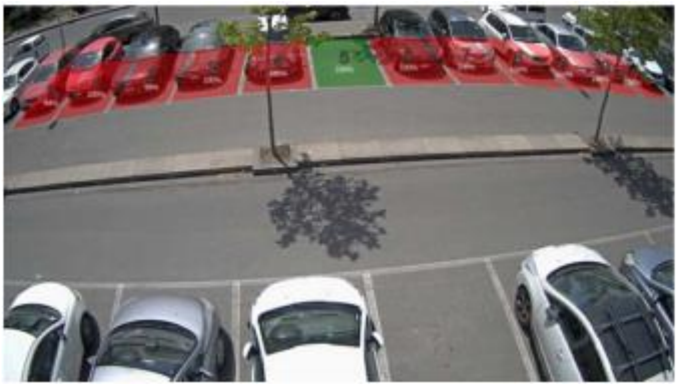


Figure 2.4 Example of how Park Smart system.

2.3.1.1 How the system works

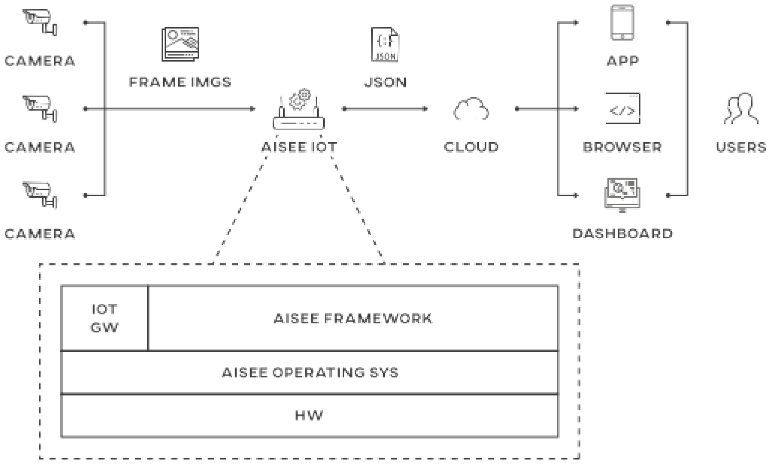


Figure 2.5 Illustration of hardware and software in the system.

Figure 2.5 shows that images and videos are captured by cameras which send them to the AISEE embedded where the computation is performed. From there the information about the parking status is send to the cloud in order to be viewed by users. Images were taken from three different parking areas and were positioned at different height.

2.3.1.2 Hardware & Software

The hardware & software that this system contains is camera, AISEE IoT, and Could.

a) Camera

This system use wide angle cameras to maximize the effective number of parking spaces monitored. Best resolution results need at least 50px for each side of parking space. It has more than 12000 images with resolution of 1280 × 720 pixels.

b) AISEE IoT

They analyse the video stream as detailed as possible to the camera. It is an embedded system capable of elevated computing power, enough to do inference using deep learning models. Once inference is done the results are sent to the cloud platform. The embedded operating system has been developed with security, privacy and resilience in mind. Then it can deploy several AISEE IoT boxes depending on the number of cameras and the dimension of the installation.

c) Cloud

This system will get all the information from the installed system through a cloud platform which is platform which is scalable by design that using a network of remote servers hosted on the Internet.

2.3.2 Automation of Real Time Car Parking System

Automation of real time car parking system (RTCPS) is the system that using mobile cloud computing (MCC) and vehicular networking (VN). It gives popularity to a novel concept of integrated communication-computing platforms (ICCP). The goals of ICCP is to develop an productive means of addressing problem such as inappropriate parking management scheme, traffic congestion in parking lots, unsecured of vehicles, and other Infrastructure-to-Vehicle (I2V) services for providing data distribution and content of services delivered to connected Vehicular Clients (VCs) (Anderson, Okafor, Nkwachukwu, & Dike, 2017).

2.3.2.1 How the system works

Interactive based sensor network system in the parking lot operates as a non-identical node. The sensors in the parking lot using Ultrasonic Distance Sensor and Passive Infrared Sensors are all connected to the Arduino Duo throughout wired connections. All sensor nodes are connected to the single Arduino Duo. Camera LinkSprite 2MP UART JPEG is also apply in the system to capture or get image of the driver’s face at both entrance and exit of parking lot and send those images for the system to process. For security, finger print scanner (FPM384) and SD card are applied to the processor through wired connection. LEDs are also deployed at each parking slot to ensure proper parking management scheme in the parking lot, while the LCD is used to display vacant parking slot. The system also contain of a GSM Module which communicate with the user via SMS after receiving a promt from the website and also get data from the Arduino Duo is used to update the website using GSM Module. The website allows the users to view status of parking lot in real-time.

2.3.2.2 Hardware & Software

The hardware & software that this system contain is Ultrasonic Sensor, Voice enabled System, GSM Module, Camera and LCD, Pi-Camera and IoT.

a) Ultrasonic Sensor

PING ultrasonic sensor gives an easy technique to measure distance. This sensor is good for applications that need to work on measurements between moving or immobile objects. In this system, this ultrasonic sensor represents as a vehicle detector.

b) Voice enabled System

Voice enabled System are used to guide the users to the selected parking slot and Camera to increase Security in Parking lot.

c) GSM Module

GSM module is used to send SMS to users to inform them about status of parking lot and payment fees.

d) Camera and LCD

Camera is to get the image of the parking Lot and the LCD used to show status of parking lot.

e) Pi-Camera and IoT

Pi-camera is used to get status of parking lot and all the information is sent to the database and internet.

2.3.3 Low Cost Wireless Parking

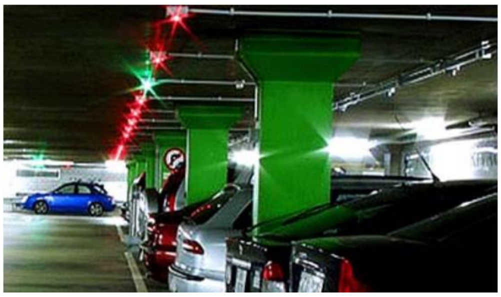


Figure 2.6 LED display in Wireless parking system.

This article propose a design and implementation for a low cost, wireless parking module, which contains of a parking sensor to detect a vehicle, and an LED display unoccupied parking space (Mutiara, Agung, & Handayani, 2018). Figure 2.6 is an overview of low cost wireless parking where the LED display an occupied of the parking space. This module uses a PING ultrasonic sensor and a NodeMCU as a microcontroller. The module connected wirelessly to the main server via a router. The module should cost less than average retail prices of similar module available on the market. A better approach for the vehicle detection scenario is also presented as a result. Figure 2.7 shows the example of PING ultrasonic sensor that has been implement on the parking space.

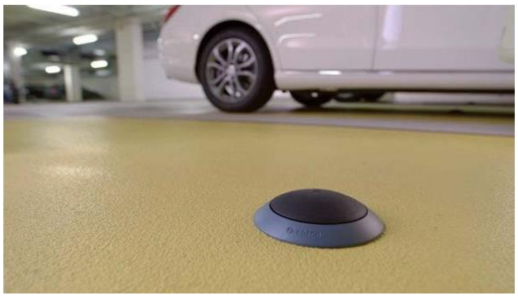


Figure 2.7 Sensor that has been implement in this system.

2.3.3.1 How the system works

The main concept of this system is most likely to ultrasonic sensor. Instead of IR, ultrasonic sensor uses sound produce in ultrasonic wavelength. Ultrasonic and infrared sensor is mainly used to detect vehicle because they are small and relatively cheap. The sensor can be arranged in assorted positions against the vehicle. The sensor can be placed separately with an LED, to notify the driver for the unoccupied of a parking space.

2.3.3.2 Hardware & Software

The hardware and software this system contain is PING ultrasonic sensor, LED notification and NodeMCU.

a) PING ultrasonic sensor

PING ultrasonic sensor allows an easy technique to measure the distance between object. This sensor is good for applications that need to perform measurements or calculation between moving or stationary objects. In this system it represent as a vehicle detector.

b) LED notification

LED notification distributes information about whether a space is free or occupied to the user. Occupied area will turn the red LED, while unoccupied space will turn the green LED.

c) NodeMCU

A NodeMCU represent as a microcontroller. This module is equipped with ESP8266 Wi-Fi System on Chip, which is used to interact with the main server. It has 128 kBytes of memory and 4 Mbytes of storage.

2.4 Comparison between Related System

The comparison between ACPMS, Park Smart System, Automation of Real Time Car Parking System and Low-Cost Wireless Parking System are shown on the table 2.1 below.

	ACPMS	Park Smart System	Automation Of Real Time Car Parking System	Low Cost Wireless Parking System
Purpose	Design and implementation for a low cost, wireless parking module.	Optimizing the path from the current driver position to a free parking lot.	This research pursue to ease urban improvement strategies and highlight the importance of communication-computing platform (ICCP) prototype tested.	Design and implementation for a low cost, wireless parking module.
Mobile Apps	Android	Android	Android	N/A
Sensor	Ultrasonic sensors	N/A	Ultrasonic sensors	Ultrasonic sensors
Type of connection	Wi-Fi	Wi-Fi	GSM	Wi-Fi
Location	Indoor	Outdoor	Indoor/outdoor	Indoor

Table 2.1 Comparison between related system.

CHAPTER 3

METHODOLOGY

3.1 Introduction

For this chapter, the flow of the system alongside with the detailed description about the software and hardware is discussed in order to design and develop ACPMS. It will cover context diagram, flow chart, Data Flow Diagram (DFD) Level 0, use case and all related things that are involved in this project. Gantt chart is also developed as a timeframes the full progress of process in the application to indicate the steps alongside with their time duration of. The particular information of implementation that has been fulfil in this project is provide in this chapter.

Software Development Life Cycle (SDLC) is one of the processes used to describe the sequence of activities and tasks that has been applied by software business to design, develop and test high standard software. It explains in specific about how to develop, maintain, replace and alter or upgrade particular software. There are many types of methodologies in SDLC to advise the operation involved such as Waterfall Model, V-Shaped Model, Evolutionary Prototyping Model, Spiral Method (SDM), Rapid Application Development (RAD), Extreme Programming and others. For this project, Rapid Application Development (RAD) methodology will be employ. This methodology will be focus on iterative development process, business modelling, data modelling, process modelling, testing & turnover and also any corresponding software programming and hardware equipment for the whole development process. All the data in Chapter 1 and Chapter 2 are gathered and analysed before actualize to Chapter 3.

3.2 Rapid Application Development (RAD)

In software development, Rapid Application Development (RAD) is a concept that has been improved from waterfall software design approach establish by a British Information Technology consultant and author named James Martin since that the result in products were too often inefficient by the time they were actually released. It is develop to take benefits of strong development software like frameworks tools, CASE tools and code generators. The crucial aim of this methodology is to develop a high speed, high quality and low cost system for ACPMS.

Rapid application development contain four (4) steps which are requirement planning, design, construction and cutover. Now, the phases involved to develop this system in this methodology will be discuss in detail.

Phase 1: Requirement Planning Phase

The requirement and planning to make ACPMS will be gathered and combine together to reduce the duration of this system progress. The main component, Arduino Uno board will be described in detail including its task function, discover the system scope and discover the data subject areas based on its functionality of what this system will support. The outcome or deliverables from this phase are modelled. It include of system scopes and research of the existing systems.

Phase 2: Design Phase

This phase will gather all requirement and decision that has been made in requirement planning phase. This phase will be repeated as often as necessary when the project evolves. The requirement that has been collected will be specified into hardware, software, system configuration and system architecture to complete ACPMS.

Phase 3: Development Phase

Development or construction phase of ACPMS will began after the basic user and system design has been develop. This phase will be include the application coding, testing and integration. When user design phase are changing, construction phase also will be repeat as new requirement are required to meet the objective of this project.

Phase 4: Cutover Phase

The cutover phase is the final phase in the Software Development Life Cycle (SDLC) implementation phase. Every ACPMS component will be test to make sure the product and system will meet the requirement of the system. This includes fixing any fault or error that found when testing the entire component. Figure 3.1 is the phases by James Martin to approach RAD.

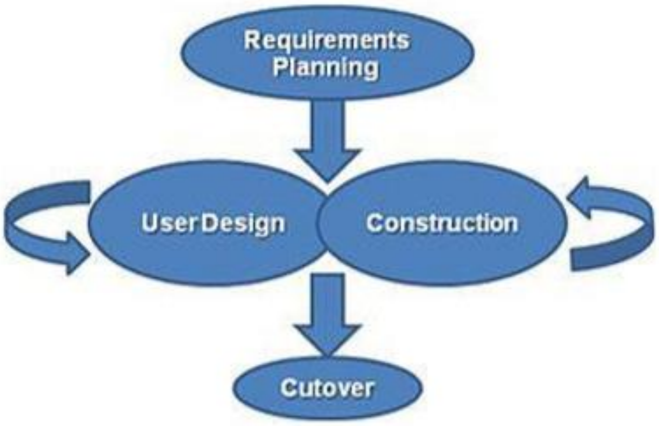


Figure 3.1 Phases in the James Martin approach to RAD

3.3 Preliminary Design

This section will discuss the proposal of the proposed system along with the necessary flow of the system alongside software and hardware arrangements that will be suitable to complete this project.

3.3.1 System Description

The target of the system is to prevent waste of time and unnecessary energy when find parking space in parking mall for the user. The main device that will be used for this system is Arduino Uno. Figure below shows the flow of the system.

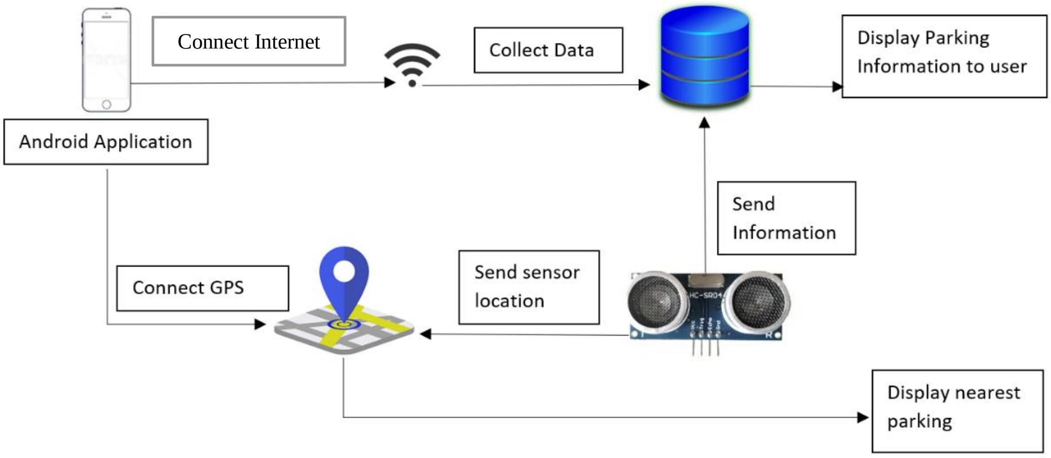


Figure 3.2 Flow of the system.

Based on the figure 3.2, user need connect to internet connection in order to access the system. The system provides three function for user which is to view parking space detail, get nearest parking and view colour parking guide. For view parking space it only required for user to open the application on their smartphone after they connect to the internet. For the nearest parking, pop-up message and parking information view will turn blue for a second after user push the available button. Colour parking guide is to give user information about what the colour indicates for. While for ultrasonic sensor, when it detects present of car parking or car leaving the parking it will send the information to the system database.

3.3.2 Printed System Description

The printed system descriptions for connection between Arduino, ultra-sonic sensor and Wi-Fi-shield are shown on the figure 3.3 and figure 3.4 below.

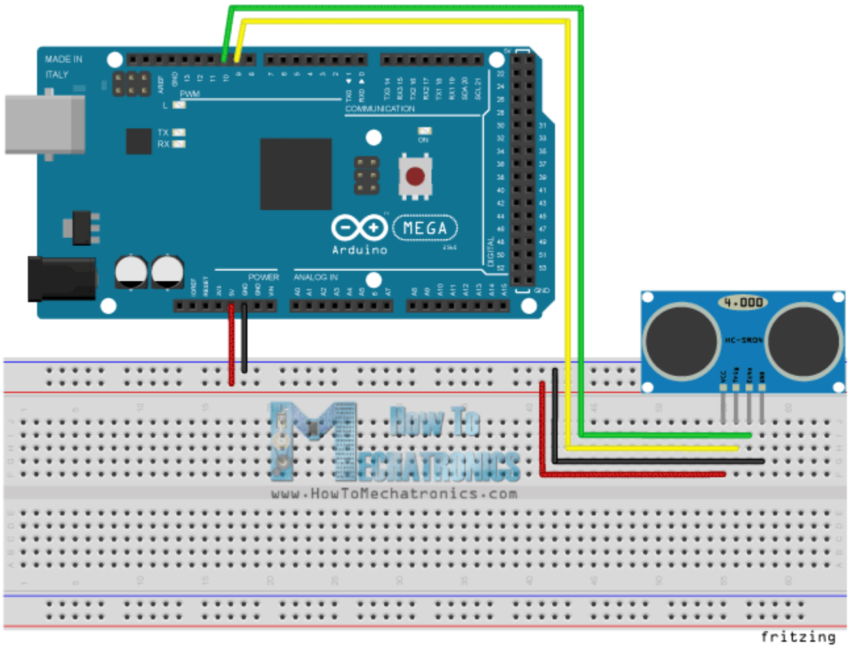


Figure 3.3 Arduino UNO connect with ultrasonic sensor

.

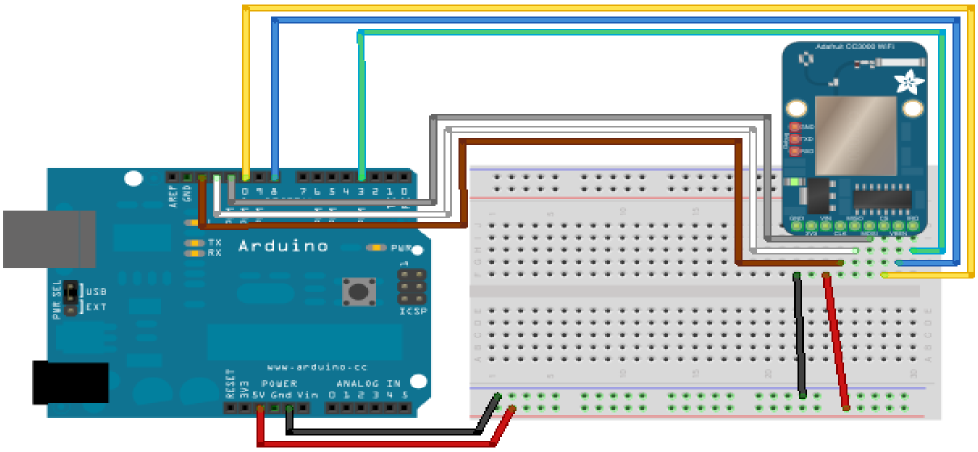


Figure 3.4 Arduino UNO connect with Wi-Fi shield.

3.3.3 Context Diagram

Context diagram explains the boundary between parts of the system and display the entities that interact with it. For this situation, Android-Based Parking System is developed to prevent waste a lot of time and unnecessary energy when find parking space. It is capable to show either parking space is occupied or not. User also able to get the nearest parking available for them.

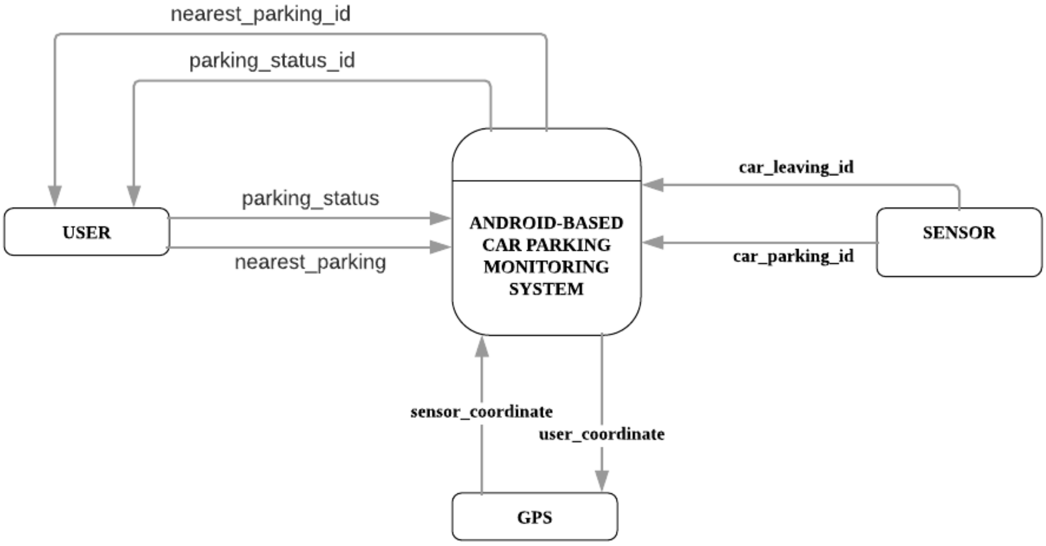


Figure 3.5 Context Diagram for ACPMS.

According to figure 3.5, ACPMS is used by user using their mobile application. User can monitor the available parking space through the system in real which the system will update every second. User can be notified the nearest parking available in the parking lot that is nearest to their location by clicking the available button. After user click the button, GPS will get their location and calculate the nearest parking to them. While ultrasonic sensor will always update the information about the car going in and out in the parking space and record all the information in the database.

3.3.4 Data Flow Diagram (DFD) Level 0

Data flow diagram level 0 will discuss more detail about context diagram on how the system interacts with each other. Here are the details in DFD level 0.

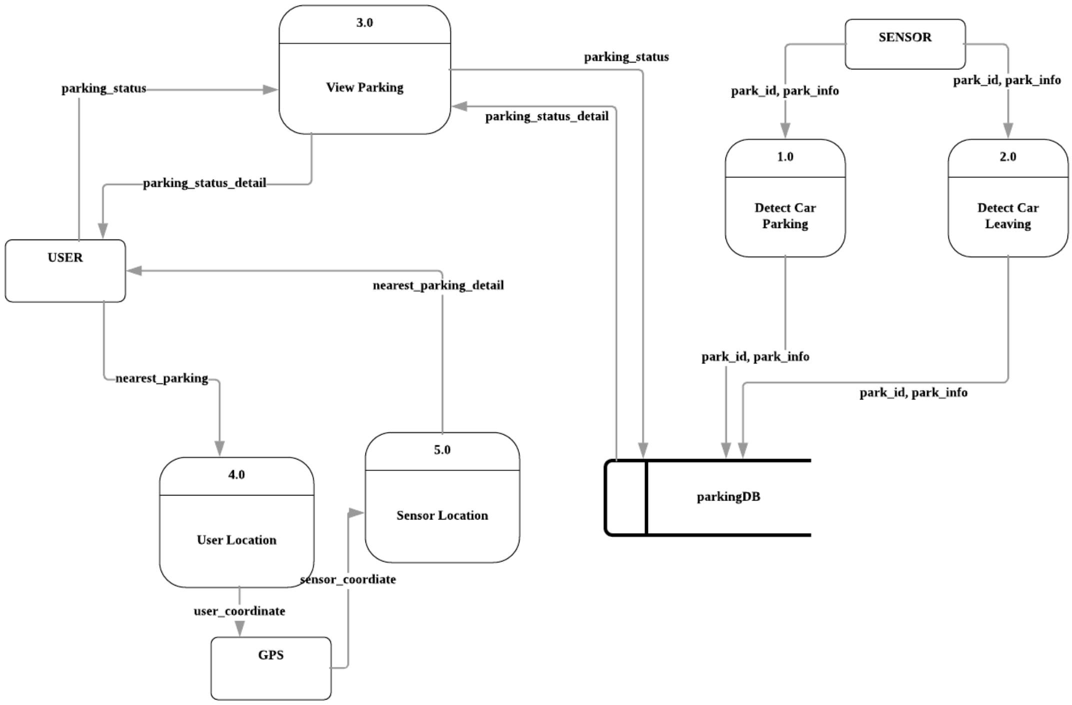


Figure 3.6 Data Flow Diagram (DFD) Level 0 for.

According to figure 3.6, it shows that flow of the system more detail in ACPMS. The system will retrieve information from the sensor whether car parking or leaving from the parking lot by their assigned ID. When it detects a car parking or leaving, it will send the information to database. This information is updated from time to time as long the system is not turned off. While for user, they can view all the parking information from their smartphone and the information is updated every second. User can get the nearest parking to them by clicking the button on the interface and the system will access user location using GPS and calculate distance between the user and the nearest parking that available for them.

3.3.5 Use Case Diagram

Use case diagram will show the interaction between the user and main component for Android-Based Parking System. The use case is shown in figure below.

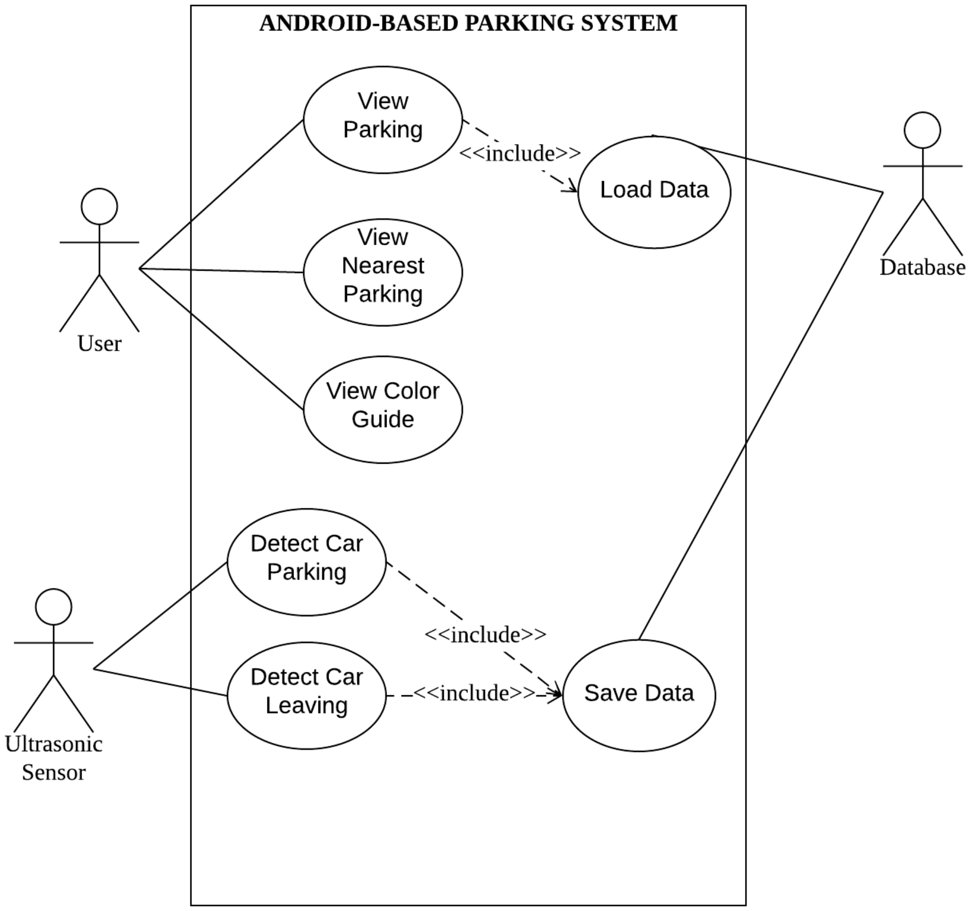


Figure 3.7 Use case diagram for ACPMS.

According to figure 3.7, ACPMS contain three external entities which is user and sensor and database. User will interact with three use case which is view parking and view nearest parking and view colour guide. For view parking, it will load data from database to get the parking information. While for sensor it is interact with two other use case which is detect car parking and detect car leaving and those information are send to database. Database entity act as middle man which connect between Arduino and phone to send and receive data from each other.

3.3.6 Flowchart Diagram

Flowchart will discuss in detailed about the flow of the ACPMS from user and the hardware (ultra-sonic sensor) from beginning until end. The flows are shown in figure below.

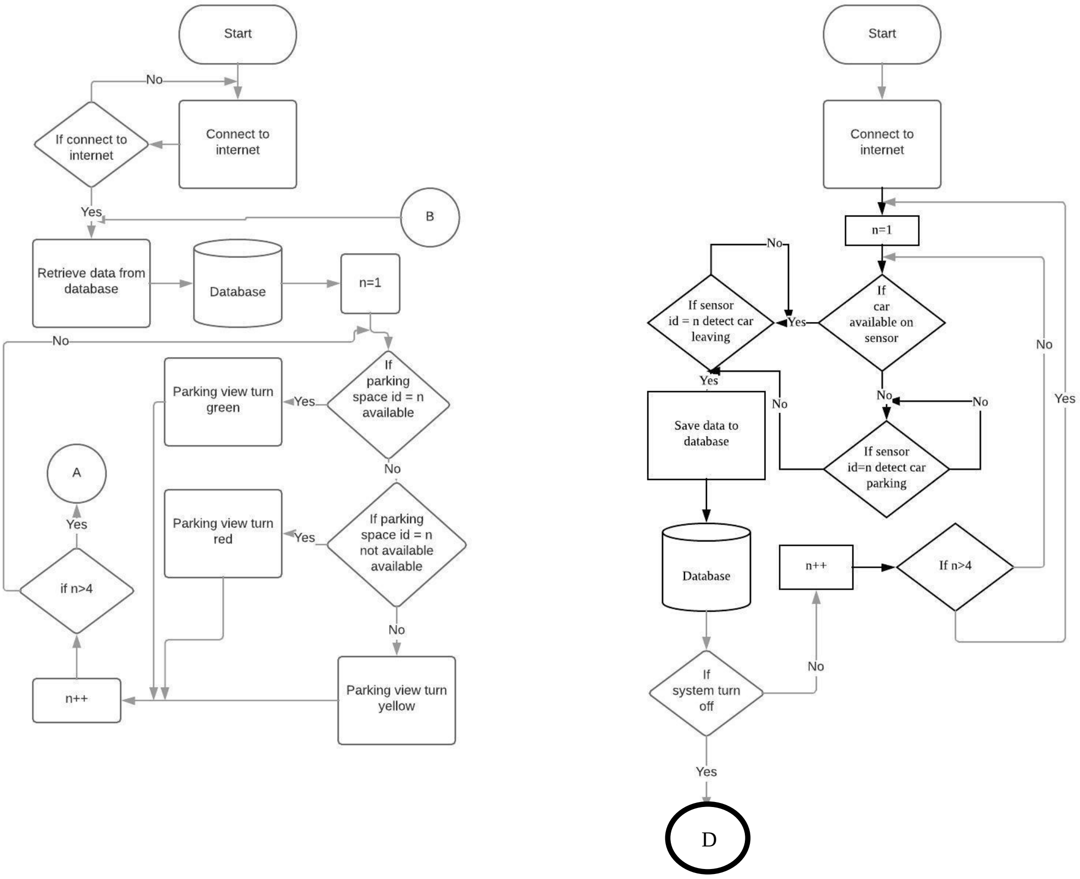


Figure 3.8 Flowchart for ACPMS.

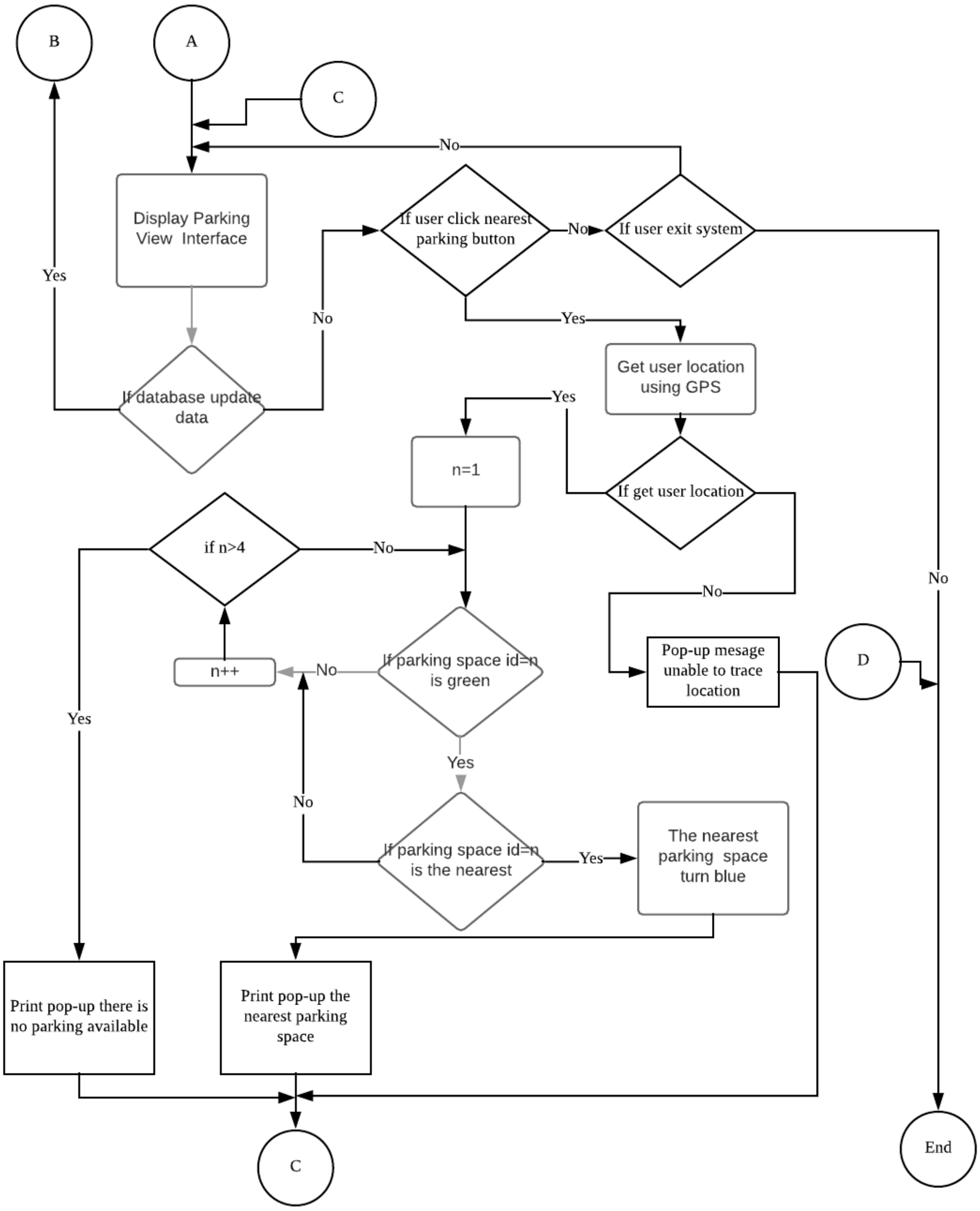

B
A
C
No
Display Parking View Interface
If user click nearest parking button
No
If user exit system
Yes
No
Yes
Get user location using GPS
If database update data
Yes
n=1
If get user location
if n>4
No
No
No
If parking space id=n is green
Pop-up mesage unable to trace location
D
n++
No
Yes
Yes
No
If parking space id=n is the nearest
Yes
The nearest parking space turn blue
Print pop-up there is no parking available
Print pop-up the nearest parking space
C
End

Figure 3.9 Flowchart for ACPMS.

According to figure 3.8.1 and figure 3.8.2, ACPMS will start if user connect to internet. For each second, the system will loop to get data from database for all four sensors that available. If sensor detect parking available, the system will view green colour in the parking space according to their id and if there is no parking available the colour will turn red. While if there is no sensor in the parking lot, parking space in the system will turn to yellow colour. User also can get the nearest parking available if the user clicked the specific button on the same interface. If user click the button, the system will ask for user to turn GPS on. Then the system will get their location using the GPS service and calculate the shortest distance between user and the nearest parking. If the parking is available and nearest to the user, then the parking space in the system interface will turn blue and give pop-up message which parking is the nearest to them. While for Ultrasonic-sensor, Cytron ESP8266 Wi-Fi Shield will connect to the internet then it will loop to detect whether there is parking available or not for each parking that have Ultrasonic-sensor installed on the parking. The data will be updated from time to time to the database.

3.4 Hardware & Software

This section will discuss about the hardware and software needed to complete the ACPMS.

3.4.1 Hardware Requirement

Table 3.1 below shows the hardware items that are being used throughout the phases to complete the project.

Hardware	Specification	Purpose	Quantity
Laptop	ASUS A42J, Intel® Core™ i5-430M (2.26 GHz, Cache 3 MB), 500 GB Serial ATA 5400 RPM	To develop the system, prepare the documents and proposals.	1
Printer	HP Deskjet 2480	To print documents.	1
Microcontroller	Arduino Uno	As microcontroller for the whole PIC.	1
Sensor	Ultrasonic sensors	To detect the present of vehicle.	1
Wi-Fi Shield	Cytron ESP8266 Wi-Fi Shield	To connect Arduino board to the network.	1
Handphone	OPPO F7, 64 GB, 4 GB RAM, Android version 8.1.0	To run android apps that monitoring the System.	1

Table 3.1 Hardware Requirement

3.4.2 Software Requirement

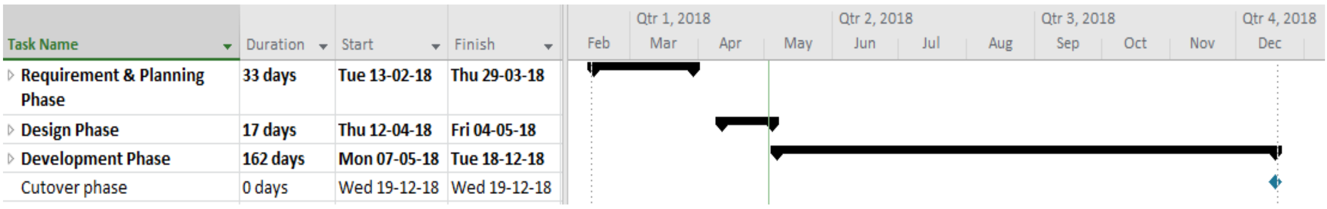
Table 3.2 below shows the software items that are being used throughout the phases to complete the project.

Software	**Purpose**
Microsoft Windows 10 (x64) Operating System	A platform to run the applications required for the documentation and development.
Arduino Integrate Development Environment (IDE)	A tool to write the code for Arduino Uno and compile it in Arduino Uno.
Microsoft Office Word 2013	A word processor to prepare the documentation
Microsoft Project 2013	A tool to prepare the Gantt Chart.
Google Chrome web browser	An application to access to the internet to find related information about the project.
Android Studio	A tool to develop a mobile application.
000webhost phpMyAdmin	To create database and view the project for admin To connect Arduino Uno and Android Studio.

Table 3.2 Software Requirement

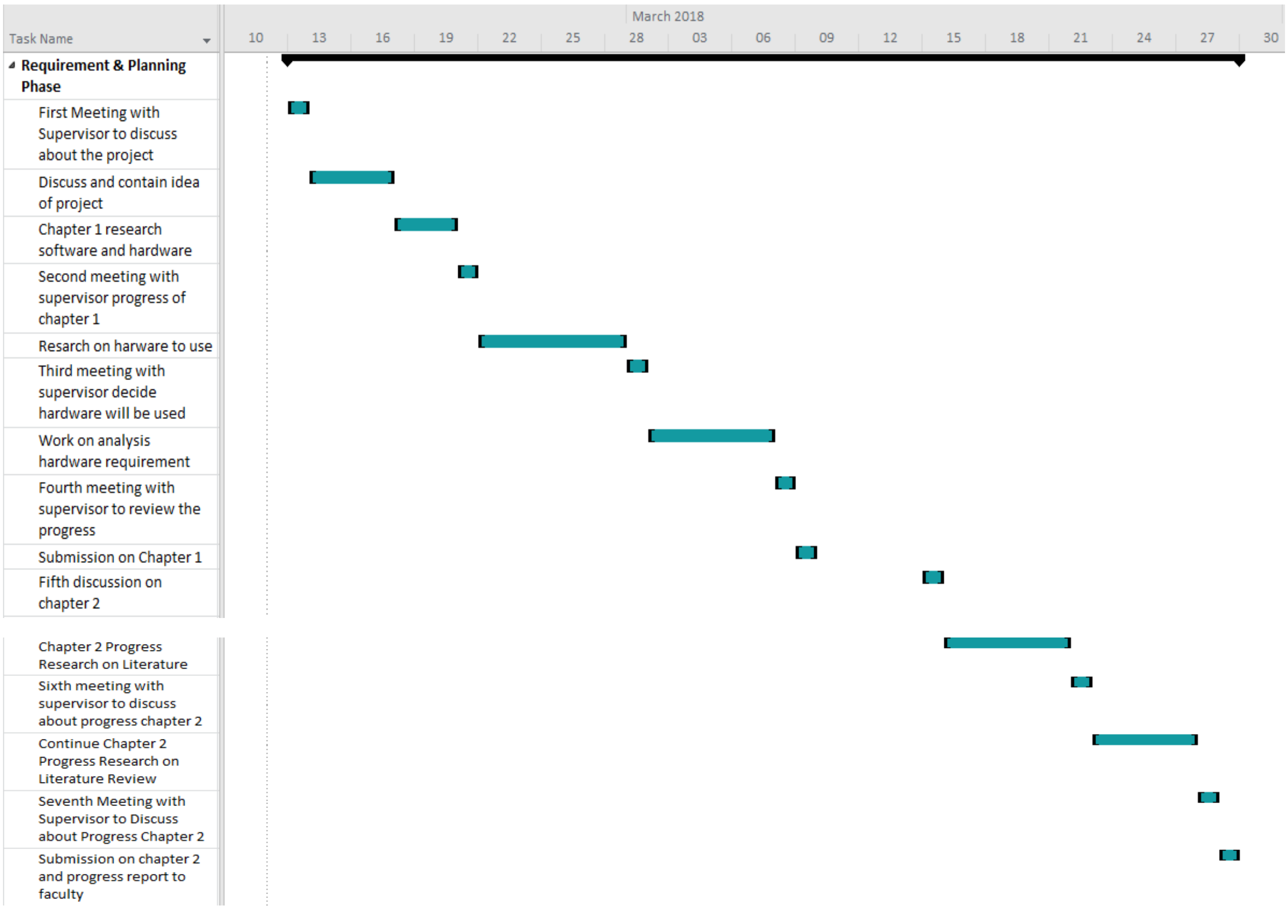
3.5 Gantt Chart

In this section, Gantt chart will be shown from day one until the project finish in chapter 1. It is shown in figure below.



Task Name	Duration	Start	Finish
Requirement & Planning Phase	33 days	Tue 13-02-18	Thu 29-03-18
Design Phase	17 days	Thu 12-04-18	Fri 04-05-18
Development Phase	162 days	Mon 07-05-18	Tue 18-12-18
Cutover phase	0 days	Wed 19-12-18	Wed 19-12-18

Figure 3.10 Gantt chart for whole project



Task Name

- Requirement & Planning Phase
  - First Meeting with Supervisor to discuss about the project
  - Discuss and contain idea of project
  - Chapter 1 research software and hardware
  - Second meeting with supervisor progress of chapter 1
  - Resarch on harware to use
  - Third meeting with supervisor decide hardware will be used
  - Work on analysis hardware requirement
  - Fourth meeting with supervisor to review the progress
  - Submission on Chapter 1
  - Fifth discussion on chapter 2
  - Chapter 2 Progress Research on Literature
  - Sixth meeting with supervisor to discuss about progress chapter 2
  - Continue Chapter 2 Progress Research on Literature Review
  - Seventh Meeting with Supervisor to Discuss about Progress Chapter 2
  - Submission on chapter 2 and progress report to faculty

Figure 3.11 Gantt chart for requirement and planning phase during PSM 1

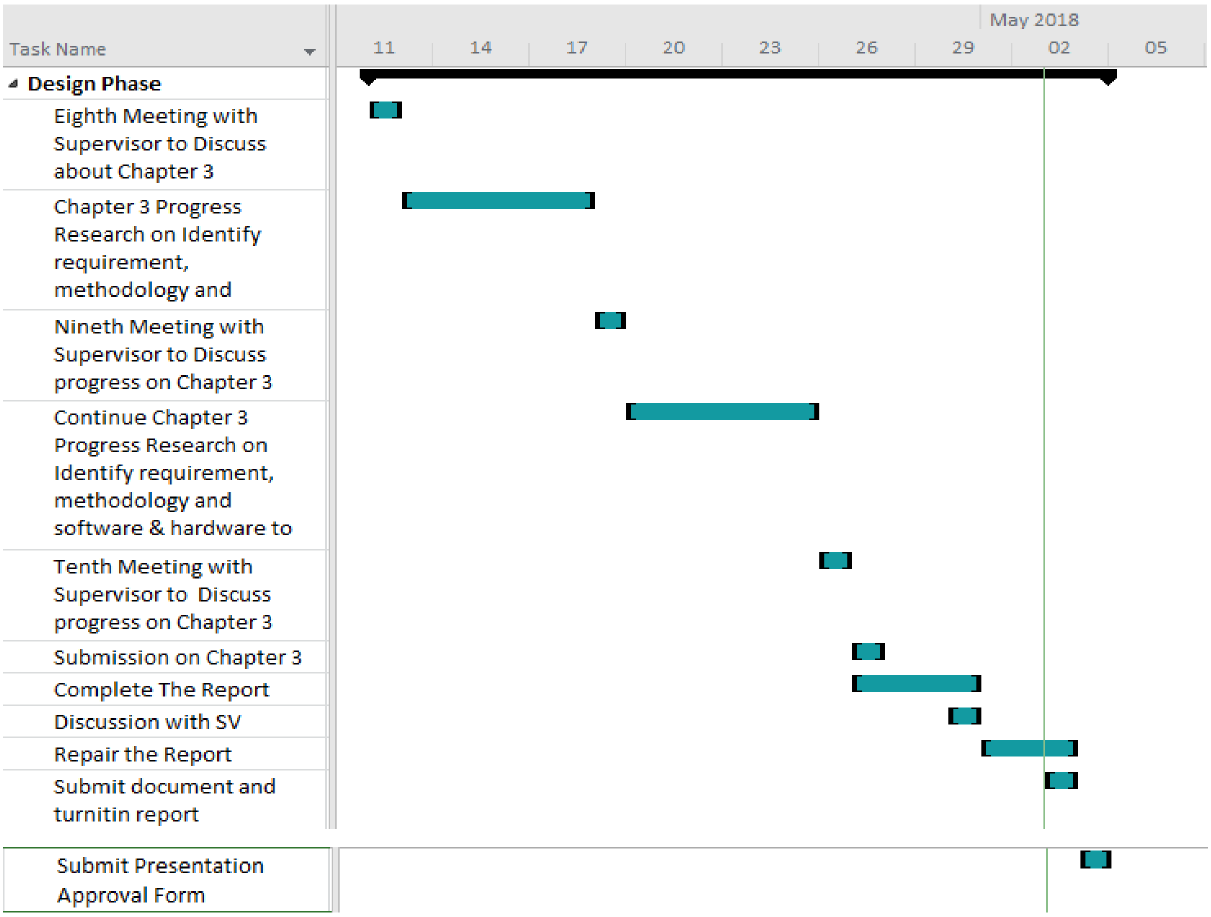


Figure 3.12 Gantt chart for design phase during PSM 1

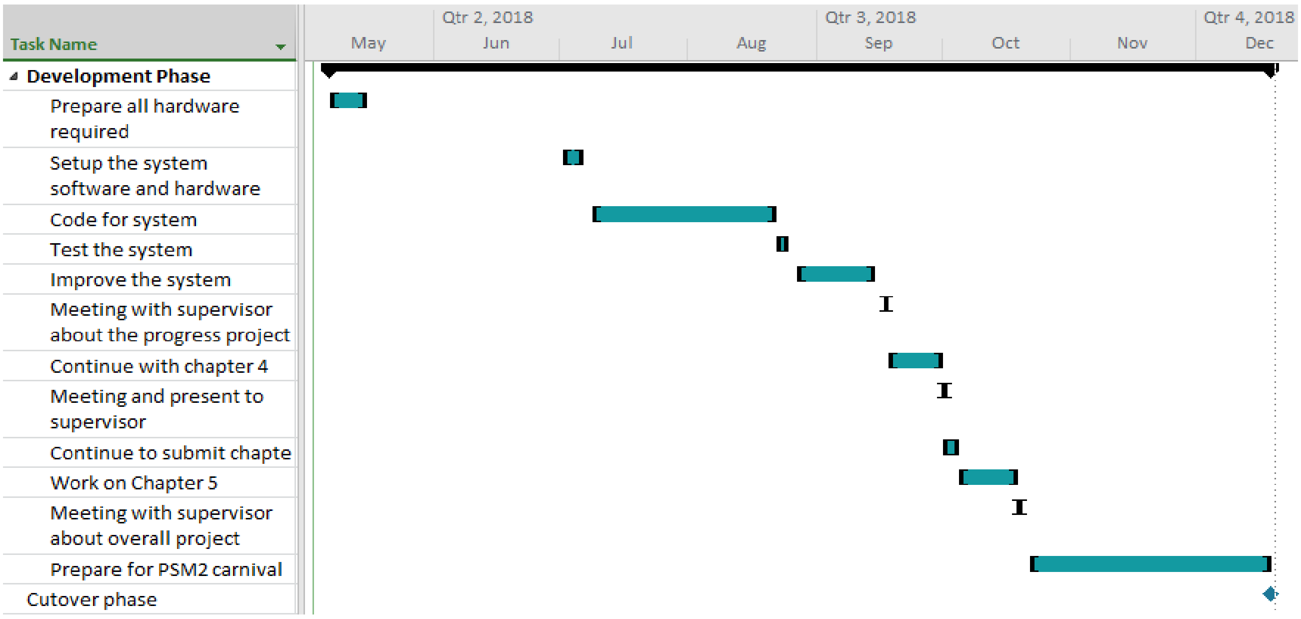


Figure 3.13 Gantt chart for PSM 2

3.6 Conclusion

Rapid Application Development (RAD) methodology was selected due to its flexibilities and adaptabilities. The system can be updated easily from time to time because there are no process forms of management. Moreover, this methodology is very suitable for small system that requires a brief developing period.

In order to develop the system, the best and reliable hardware and software was listed and chose. The process of selecting the best hardware and software are very important to make sure the requirement that was chosen best meet the system design.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, process of implementation will be discussed further in detail. It will also include the documentation of process in the project development. This chapter will cover the implementation of the project which are using Arduino Uno, Ultrasonic Sensor, Cytron ESP8266 Wi-Fi Shield and development of Android-based Car Parking Monitoring System (ACPMS).

Furthermore, it also discusses about the Arduino Uno where it will receive the information from Ultrasonic Sensor and send those data to Mobile Application by using Cytron ESP8266 Wi-Fi Shield. Then, ACPMS mobile application will retrieve the data and display all the data that has been send from the Arduino Uno in real time. So, user can get latest information about the availability of car parking using ACPMS that has been develop.

4.2 Implementation Requirement

There are three stages need to be completed for this system which is the hardware installation for Arduino including the code, PHP file code for XAMPP phpMyAdmin and ACPMS mobile application development. In addition, developing mobile application for this project which is to ease the user to view the current parking available. The project requires proper software installation and hardware configuration to facilitate the system run smoothly. In order to implement the prototype of this system, all hardware must be installed correctly with the right instruction. A few mistakes may cause the malfunction to the system. All precautions must be taken before started the installation to avoid any problems in the future.

4.2.1 Hardware Implementation

The hardware required for this system are Arduino Uno, Ultrasonic Sensor, Cytron ESP8266 Wi-Fi Shield, Bread Board, Male to Male jump Wire and Male to Female Jump Wire. Firstly, steps for the hardware connection is attach Cytron ESP8266 Wi-Fi Shield to Arduino Mega by putting it to the correct pin. After that, all four Ultrasonic Sensor connect to the bread board using Male to Female Jump Wire. Then using Male to Male Jump Wire to connect from Bread Board to Arduino Uno with GND pin to GND pin and VCC pin to 5V pin. While for the Trig pin and Echo pin for Ultrasonic Sensor No 1 will connect to the Arduino Uno pin #2 and #3, Ultrasonic Sensor No 2 to pin #4 and #5, Ultrasonic Sensor No 3 to pin #6 and #7 and Ultrasonic Sensor No 4 to pin #8 and #9. Then, after all component are attached to Arduino, it needs to be program by using Arduino IDE.

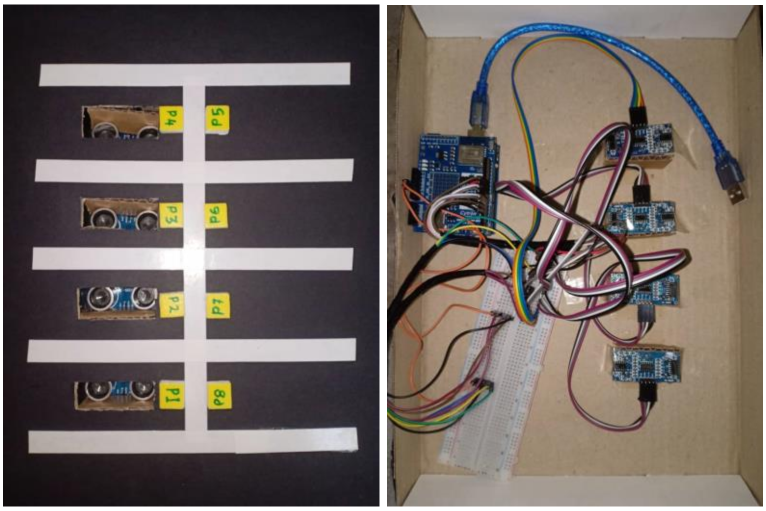
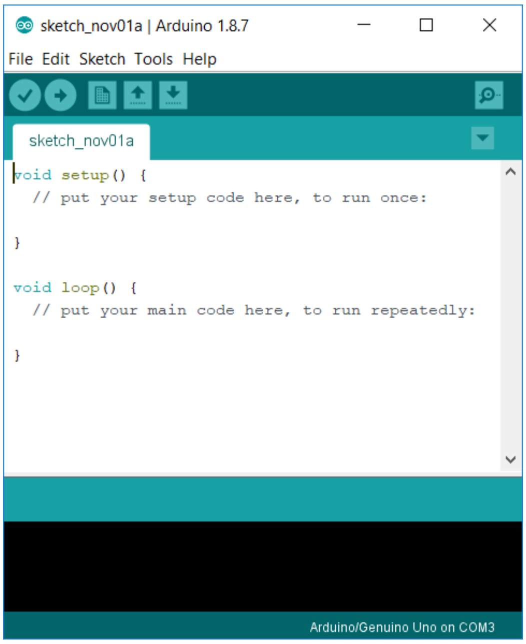


Figure 4.1 ACPMS kits

Figure 4.1 shows the connection of all the hardware components used to be ACPMS Kits. Use Arduino IDE (Integrated Development Environment) compiler to program the Arduino Uno to post and all information from sensor to internet through Cytron ESP8266 Wi-Fi Shield.

Based on the previous chapter stated, the platform to build the program is Arduino Uno where this compiler provided own IDE as shown in Figure 4.1. Basically, to get starting programming using this IDE, the main board should connect to the computer by plugging the board to the USB cable and then, connect that cable to computer. Besides, the computer also gives power to the board to make it interact with Arduino IDE.

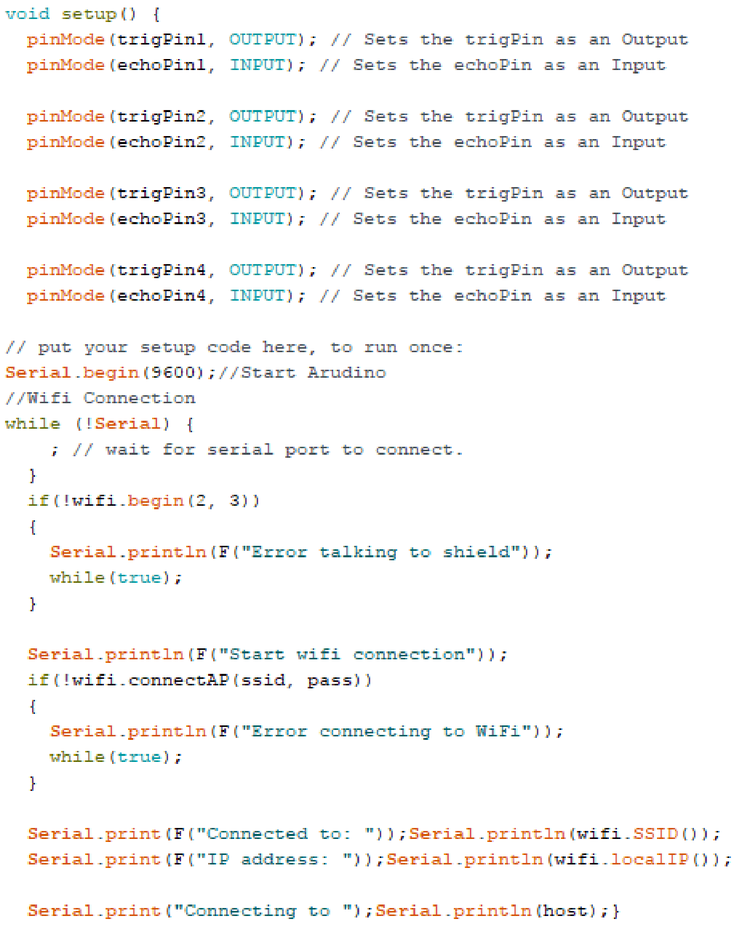


sketch_nov01a | Arduino 1.8.7
File Edit Sketch Tools Help
sketch_nov01a
Arduino/Genuino Uno on COM3

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

}

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

}
```

Figure 4.2 Arduino IDE Windows Application

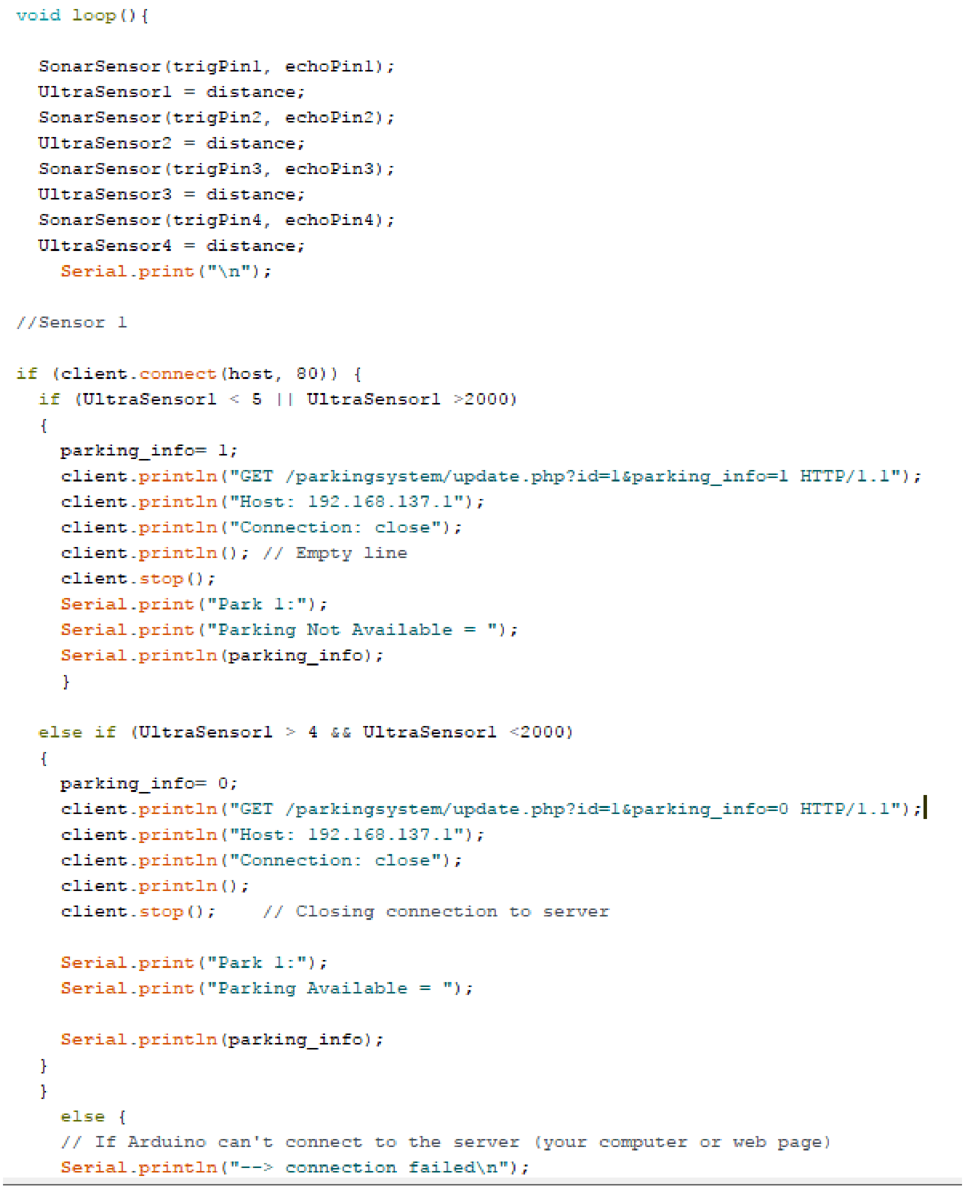
This Arduino IDE application is based using C programming language, which is easy to understand and implement (Software, 2018). Before starting program Arduino Uno, it is important to know what libraries to use in this system. For this system, libraries used is Cytron Wifi shield, Cytron Wifi Client, Cytron Wifi server and ESP8266_AT. After that, coding and programming can be started. Arduino IDE always have void setup and void looping which means in void setup all things need to be setup once should be program in this section such as setup for Wi-Fi as the component need to connected to router once. Then in void looping, it is for looping program such Ultrasonic sensor which is need to detect vehicle that park at the parking lot for every time it loops.



```
void setup() {
  pinMode(trigPin1, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin1, INPUT); // Sets the echoPin as an Input

  pinMode(trigPin2, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin2, INPUT); // Sets the echoPin as an Input

  pinMode(trigPin3, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin3, INPUT); // Sets the echoPin as an Input

  pinMode(trigPin4, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin4, INPUT); // Sets the echoPin as an Input

// put your setup code here, to run once:
Serial.begin(9600);//Start Arudino
//Wifi Connection
while (!Serial) {
    ; // wait for serial port to connect.
  }
  if(!wifi.begin(2, 3))
  {
    Serial.println(F("Error talking to shield"));
    while(true);
  }

  Serial.println(F("Start wifi connection"));
  if(!wifi.connectAP(ssid, pass))
  {
    Serial.println(F("Error connecting to WiFi"));
    while(true);
  }

  Serial.print(F("Connected to: "));Serial.println(wifi.SSID());
  Serial.print(F("IP address: "));Serial.println(wifi.localIP());

  Serial.print("Connecting to ");Serial.println(host);}
```

Figure 4.3 Functions for void setup program

Figure 4.3 shows the function used for the void setup that will connect to the specific Wi-Fi in the mall. It will capture the SSID, password, host and port where it needs in order to make connection to the Wi-Fi. If the SSID and password is correct, then the Wi-Fi will start the connection. Otherwise, connection will be interrupt. The trigger and echo pin for all sensor also were declared here.



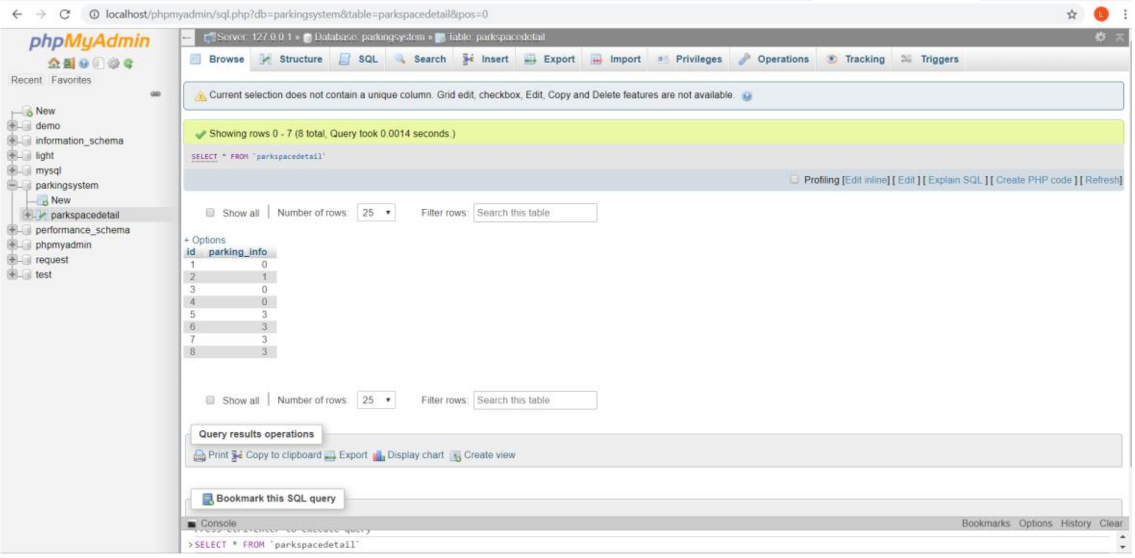
```
void loop(){

  SonarSensor(trigPin1, echoPin1);
  UltraSensor1 = distance;
  SonarSensor(trigPin2, echoPin2);
  UltraSensor2 = distance;
  SonarSensor(trigPin3, echoPin3);
  UltraSensor3 = distance;
  SonarSensor(trigPin4, echoPin4);
  UltraSensor4 = distance;
    Serial.print("\n");

//Sensor 1

if (client.connect(host, 80)) {
  if (UltraSensor1 < 5 || UltraSensor1 >2000)
  {
    parking_info= 1;
    client.println("GET /parkingsystem/update.php?id=1&parking_info=1 HTTP/1.1");
    client.println("Host: 192.168.137.1");
    client.println("Connection: close");
    client.println(); // Empty line
    client.stop();
    Serial.print("Park 1:");
    Serial.print("Parking Not Available = ");
    Serial.println(parking_info);
    }

  else if (UltraSensor1 > 4 && UltraSensor1 <2000)
  {
    parking_info= 0;
    client.println("GET /parkingsystem/update.php?id=1&parking_info=0 HTTP/1.1");
    client.println("Host: 192.168.137.1");
    client.println("Connection: close");
    client.println();
    client.stop();    // Closing connection to server

    Serial.print("Park 1:");
    Serial.print("Parking Available = ");

    Serial.println(parking_info);
  }
  }
    else {
    // If Arduino can't connect to the server (your computer or web page)
    Serial.println("--> connection failed\n");
```

Figure 4.4 Loop functions

Figure 4.4 shows one of the functions in looping section which is to capture the data of all four sensors. The sensor will read the distance between the vehicle and the Ultrasonic Sensor which is if the distance is more than 4, there will be available parking space. Otherwise, if the distance is less than 5 it means the parking space is taken. Then all the data will be stored in the database.

4.2.2 000webhost PhpMyAdmin

The second stage for ACPMS is developed the database which enable the system to store the data of car parking and car leaving. As for this project, 000webhost PhpMyAdmin is chosen as the web server. 000webhost is free webhosting provider with PHP, MYSQL and free website builder that powered by Hostiger. It is chosen for the project development because it is a free webhosting provider package and more importantly supports PHP File to send GET data to database. 000webhost includes PhpMyAdmin module that is made to be the most suitable module to become the database server for the system. The parking information are sends from the ultrasonic sensor to the database and the android application will retrieve the data and view it to user.



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 7 (8 total, Query took 0.0014 seconds.)

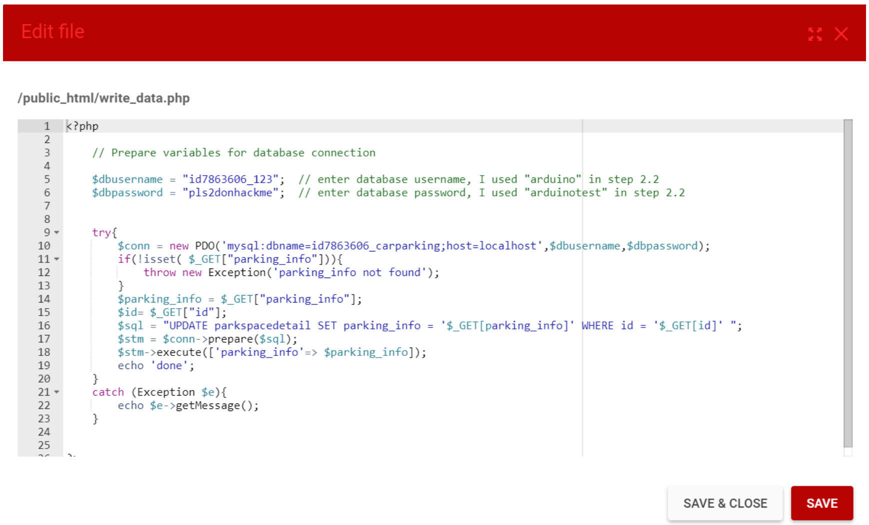
```
SELECT * FROM `parkspacedetail`
```

id	parking_info
1	0
2	1
3	0
4	0
5	3
6	3
7	3
8	3

```
>SELECT * FROM `parkspacedetail`
```

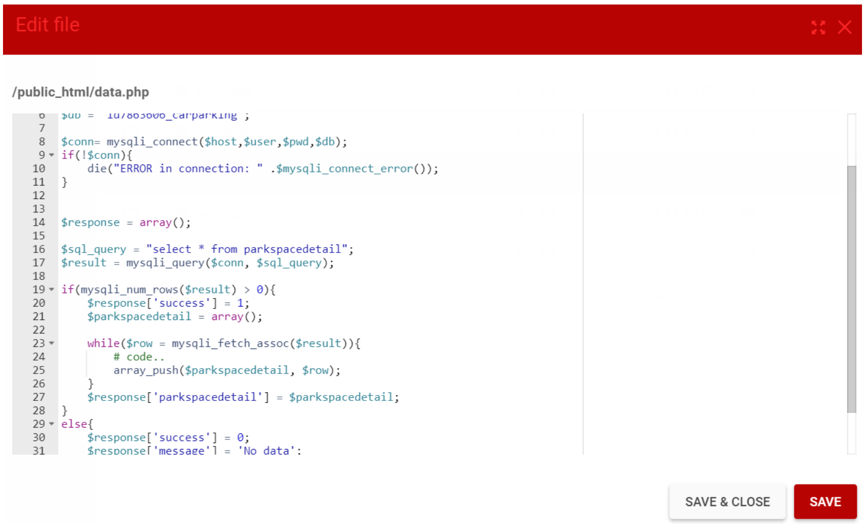
Figure 4.5 000webhost PhpMyAdmin page

Figure 4.5 shows the database for ACMPS where it contain only 1 table which is “parkspacedetail” and 2 column which is the parking id and parking info. If the parking info is 0, the android application will read the data as green. When the parking info is 1, the data will be read as red and for parking info 3, the android application will read the data as yellow.



```php
<?php

    // Prepare variables for database connection

    $dbusername = "id7863606_123";  // enter database username, I used "arduino" in step 2.2
    $dbpassword = "pls2donhackme";  // enter database password, I used "arduinotest" in step 2.2


    try{
        $conn = new PDO('mysql:dbname=id7863606_carparking;host=localhost',$dbusername,$dbpassword);
        if(!isset( $_GET["parking_info"])){
            throw new Exception('parking_info not found');
        }
        $parking_info = $_GET["parking_info"];
        $id= $_GET["id"];
        $sql = "UPDATE parkspacedetail SET parking_info = '$_GET[parking_info]' WHERE id = '$_GET[id]' ";
        $stm = $conn->prepare($sql);
        $stm->execute(['parking_info'=> $parking_info]);
        echo 'done';
    }
    catch (Exception $e){
        echo $e->getMessage();
    }


?>
```

Figure 4.6 000webhost for write_data.php



```php
$db = "id7863606_carparking";

$conn= mysqli_connect($host,$user,$pwd,$db);
if(!$conn){
    die("ERROR in connection: " .$mysqli_connect_error());
}


$response = array();

$sql_query = "select * from parkspacedetail";
$result = mysqli_query($conn, $sql_query);

if(mysqli_num_rows($result) > 0){
    $response['success'] = 1;
    $parkspacedetail = array();

    while($row = mysqli_fetch_assoc($result)){
        # code..
        array_push($parkspacedetail, $row);
    }
    $response['parkspacedetail'] = $parkspacedetail;
}
else{
    $response['success'] = 0;
    $response['message'] = 'No data';
```

Figure 4.7 000webhost for data.php

Figure 4.6 shows the PHP file to get data from Arduino and send to 000webhost. While figure 4.7 shows the PHP file that can be retrieve by mobile phone to view data from database. Notepad ++ is used in this development for verify and validate the correct command of database query. So, this 000webhost server is important as component interaction between hardware part which is Arduino Uno and ACPMS mobile application to make them connect via internet.

4.2.3 ACPMS Mobile Application Development

The last stage for ACPMS is developed mobile application for ACPMS. ACPMS mobile application is developed using Android Studio which is to make it beautiful and interactive mobile apps using Kotlin and C/C++ (https://developer.android.com/studio/intro/index.html, 2016). ACPMS mobile application is used to display all data from sensor for ACPMS kits. The kits will send the data of the sensor via internet and will displayed by using apps.

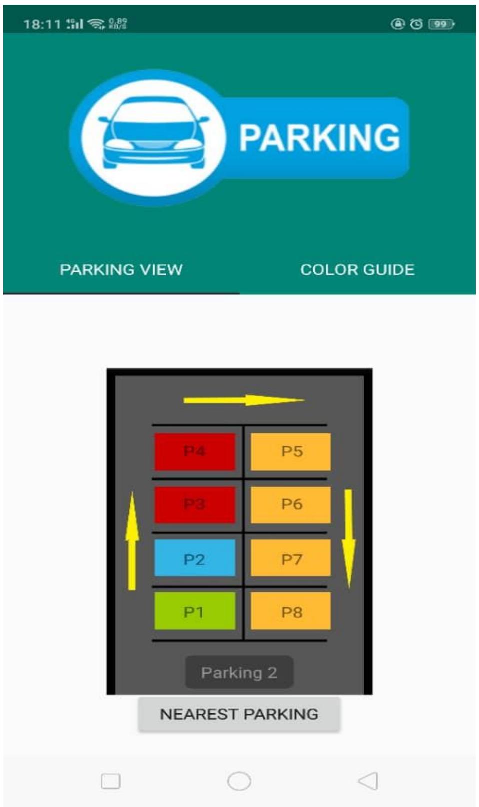
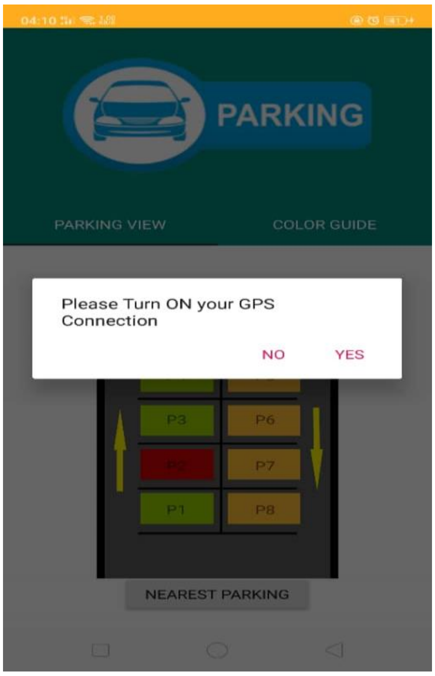


Figure 4.8 Parking view page for ACPMS system development

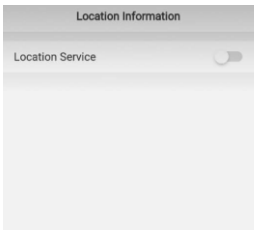
Figure 4.8 shows the parking view page of ACPMS during the development of mobile application. In this page, it will display the current parking information that the data is retrieve from 000webhost phpMyAdmin for each second. There are 4 colours that each of them indicates different meaning which is green if the parking available. Red when the parking is occupied and yellow is for the parking that does not have sensor on them. While blue will appear when user click the nearest parking available along with the pop-up message that state what parking that is nearest to them.



Please Turn ON your GPS Connection

Figure 4.9 Pop-up message when “Nearest Parking” button is clicked

Figure 4.9 shows the pop-up message if user click the “Nearest Parking” and user did not open their GPS beforehand. But if user already open the GPS the system will immediately show the nearest parking that available for them.

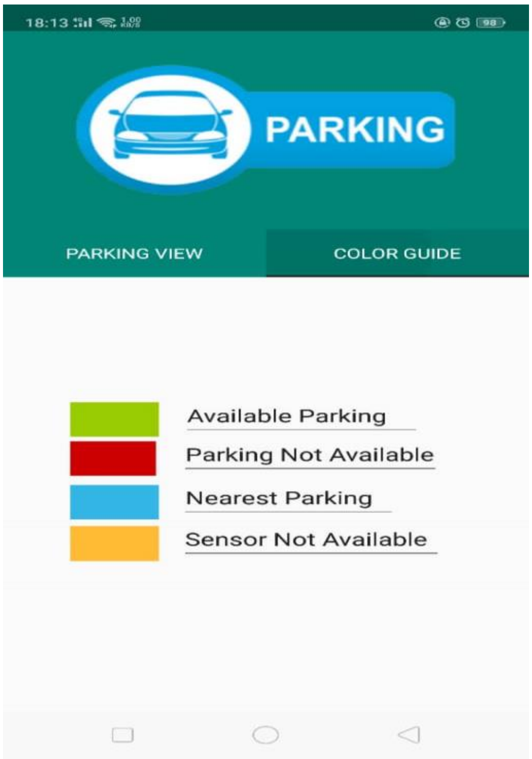


Location Information

Location Service

Figure 4.10 On Off GPS page on the user android

Figure 4.10 shows the GPS page on the user smart phone. This page will appear after user click YES on the figure 4.9 and user need to on the location service or GPS on their smart phone then continue to the view parking page.



🟩 Available Parking
🟥 Parking Not Available
🟦 Nearest Parking
🟧 Sensor Not Available

Figure 4.11 Colour guide page for ACPMS system development

Figure 4.11 is the colour guide for user to know what are the colour indicated for. This will give comfort to the user to use this system where they will know what each colour means.

4.3 Testing

ACPMS is developed using Arduino, a database and mobile application. Throughout the process, testing is one of the essential phases in the development. Testing is done after the implementation phase is completed. The main testing emphasize on the system is important test which base on the test cases on the specifications of the software component under test. Figure 4.12 shows the testing process used during development.

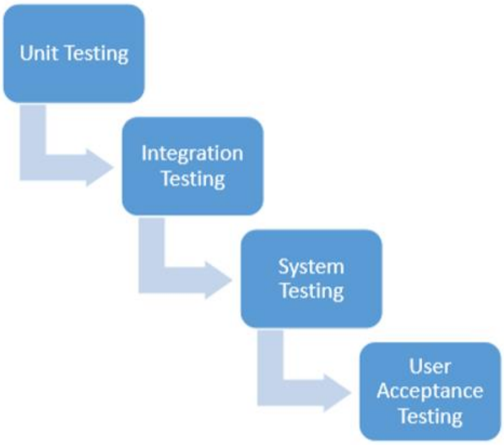
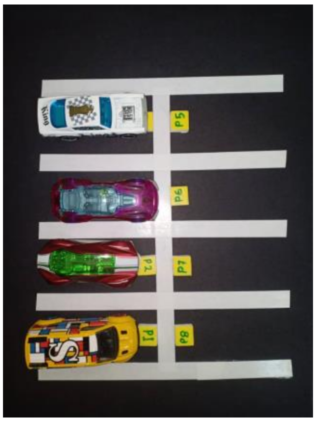
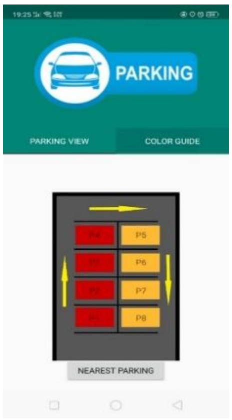


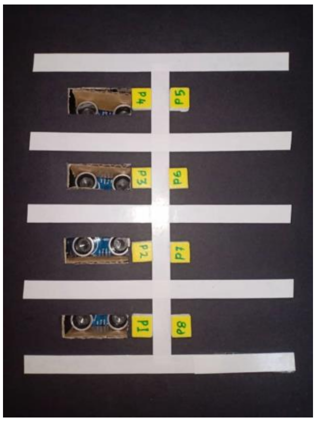
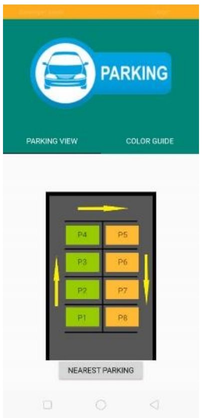
Figure 4.12 Testing Process

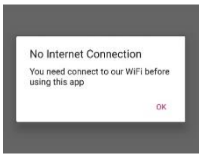
4.3.1 Testing Report

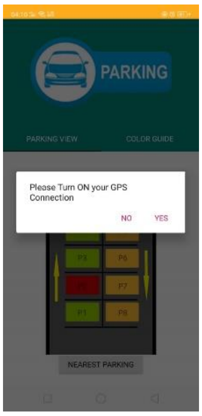
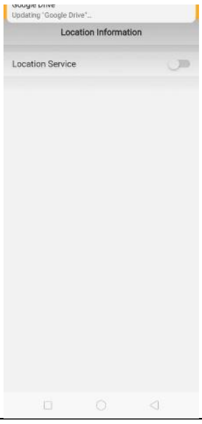
For this project, test cases of the ACPMS are documented as testing guideline and follow the standard begin with the scenario, test data, expected result, actual result and status pass or fail. The test case of ACPMS consists of seven test case as shown in table 1 until 7 and was documented as standard guideline and follow the standard begin with scenario, test case, expected result and status either success or failed.

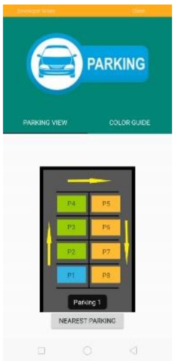
Scenario: The condition will be executed and generate result based on the condition that has been applied. The colour of parking space in the system will turn green if there is no car on the sensor and will turn red if there is a car parking on it. For mobile application, user need to connect to internet first before access to the application. Also, user need to open GPS before get the nearest parking to them. The system will calculate the nearest distance to them for available parking only. When all parking is occupied, the system will pop-up message to let the user know there is no parking available for the moment.

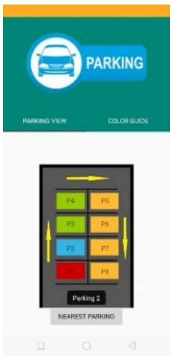
Test case 1	If parking lot occupied.
Expected result	I. Parking space in the system will turn red.   PARKING PARKING VIEW COLOR GUIDE P5 P6 P7 P8 NEAREST PARKING
Result	SUCCESS

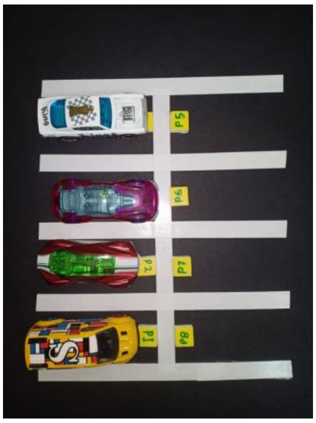
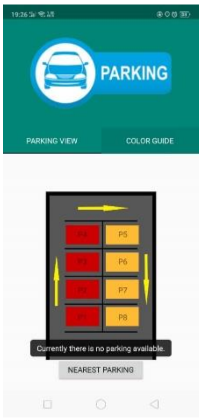
Test case 2	If parking lot is not occupied.
Expected result	I. Parking space in the system will turn green.   PARKING PARKING VIEW COLOR GUIDE P4 P5 P3 P6 P2 P7 P1 P8 NEAREST PARKING
Result	SUCCESS

Test case 3	If user open the application without connect to the internet.
Expected result	I. Pop-up message: “You need connect to Internet in order to use this application”. II. After user click OK then the system will close.  No Internet Connection You need connect to our WiFi before using this app
Result	SUCCESS

Test case 4	If user click the nearest button without open location services on their mobile phone.
Expected result	I. Pop-up message: “Please Turn ON your GPS Connection” “YES/NO”. II. If user click YES then the system will proceed to location services in their mobile phone and user need enable their “Location Service” before get their nearest location to the parking.  Please Turn ON your GPS Connection PARKING PARKING VIEW COLOR GUIDE NO YES P3 P6 P7 P1 P8 NEAREST PARKING  Google Drive Updating 'Google Drive'... Location Information Location Service
Result	SUCCESS

Test case 5	If user click the nearest button and the nearest parking is parking
Expected result	I. If parking 1 is available, then pop-up message will appear: “Parking 1”. II. Parking 1 space will turn blue for a second.  PARKING PARKING VIEW COLOR GUIDE P4 P5 P3 P6 P2 P7 P1 P8 NEAREST PARKING Parking 1
Result	SUCCESS

Test case 6	If user click the nearest button and the nearest parking is parking 1 while parking 1 is red.
Expected result	I. The system will get other nearest parking available which is the green parking. If parking 2 is available, then the pop-up message will appear: “Parking 2” II. Parking 2 space will turn blue for a second.  PARKING PARKING VIEW COLOR GUIDE P4 P5 P3 P6 P2 P7 P1 P8 NEAREST PARKING Parking 2
Result	SUCCESS

Test case 7	If user click the nearest button and the is no parking available.
Expected result	I. Pop-up message will appear: “The is no parking available”.   19:26 PARKING PARKING VIEW COLOR GUIDE P5 P6 P7 P8 Currently there is no parking available.
Result	SUCCESS

4.4 Conclusion

ACPMS is develop using 3 main implementation that is hardware implementation, database and mobile application. Arduino is used for hardware which is Arduino UNO, Ultrasonic Sensor, Cytron ESP8266 Wi-Fi Shield, Bread Board, Male to Male jump Wire and Male to Female Jump Wire. For database it uses 000webhost phpMyAdmin that act as middle man for the other two main implementation on this system. While for mobile application is using Android Studio.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS OF FUTURE WORKS

5.1 Introduction

This chapter will summarize everything of this project. The content of objective, problem statement, methodology, implementation and result will be conclude in this chapter.

The current limitation of parking availability in mall is generally occur on weekend or public holiday which many people spend their time there. So, the development of ACPMS will ease these users to find parking space through their smart phone.

Both design prototype which is the application on the smart phone and the device on the parking space has been implement to make this project successful. Users able to view the information through their smart phone that is updated from time to time for every second.

Evaluation of this prototype has been made where the error handling of the proposed system is handled smoothly. In conclusion, the implementation of ACPMS will overcome the problem statement and enhance the current problem that related to this system.

5.2 Research Constraint

5.2.1 Development Constraint

Constraints that faced along the development are:

1. TimeConstraints

   More time is needed in order to build and make the system more complete and robust.

2. Setup-Constraints

   There are some difficulties to configure the IP address and the coordinates of the sensor that installed on the parking where both of this is constantly changing before implements in the real environments.

5.2.2 System Constraint

Constraints that faced along are:

1. Platform

   This mobile application can only run on Android OS. iOS and Windows Mobile OS cannot be installed. User need to install the application to access the system.

2. Ultrasonic Sensor

   Sometimes the Ultrasonic Sensor will read wrong data for one loop periodically.

3. GPS

   Current technology does not have 1-meter accuracy from GPS to get accurate location for 1-meter distance. So sometimes the system will not get the accurate nearest parking for the distance between 5-meters.

4. WiFi-Shield

   The used Wi-Fi shield on this project has its own limitation which is its send data from Arduino to database is a bit delay. So, if the system need to send data without any delay it need more expensive quality device to make it run smoothly for the system.

5.3 Future Work

There are several enhancements that can be carried out for future improvement of ACPMS to enhances the beneficial that will follow the current environment.

I. Implement the system to the parking mall for each level of parking space that available in there.

II. Give directions to the user for them to go to the car park that is closest to them.

REFERENCES

Anderson, E. C., Okafor, K. C., Nkwachukwu, O., & Dike, D. O. (2017). *Real time car parking system: A novel taxonomy for integrated vehicular computing*. *Proceedings of the IEEE International Conference on Computing, Networking and Informatics, ICCNI 2017*. https://doi.org/10.1109/ICCNI.2017.8123788

ArduinoDevelopers. (2017). Arduino Wi-Fi Shield. Retrieved from https://www.arduino.cc/en/Main/ArduinoWiFiShield

Friess, P. (2018). Publishers Series in Communication net of Things – m Research and Innovation to ket Deployment Editors Ovidiu Vermesan Peter Friess River Publishers. Retrieved from https://www.researchgate.net/publication/278798179_Internet_of_Things_Applications_-_From_Research_and_Innovation_to_Market_Deployment

https://developer.android.com/studio/intro/index.html. (2016). Android Studio Overview. Retrieved from https://developer.android.com/studio/intro/index.html

Mutiara, G. A., Agung, A. A. G., & Handayani, R. (2018). *Low cost wireless parking module design and implementation*. *Proceeding of 2017 11th International Conference on Telecommunication Systems Services and Applications, TSSA 2017*. https://doi.org/10.1109/TSSA.2017.8272905

Page, L. (2018). Ultrasonic Range Detector Distance Sensor HC-SR04 (2cm - 400cm) 3mm Resolution. Retrieved from http://qqtrading.com.my/ultrasonic-range-detector-distance-sensor-hc-sr04

Parking, O. A., Event, N., & Network, P. (2018). Have shopping center parking problems really changed?, 2016–2018.

Software, A. (2018). What is Arduino ? Why Arduino ? How do I use Arduino ? https://doi.org/10.1111/j.1061-2971.2004.120401.x

Uno, T. A., Arduino, T., Uno, A., Specs, D., Srl, A., & Ide, A. (2018). Arduino Uno R3, (13), 1–4. Retrieved from https://www.pololu.com/product/2191