

PET FEEDER WITH NODEMCU USING
MOBILE APPLICATION

MUHAMMAD SODIQ AMIN BIN MOHAMD
ISA

BACHELOR OF COMPUTER SCIENCE
(COMPUTER SYSTEMS AND NETWORKING)

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : MUHAMMAD SODQ AMIN BIN MOHAMD ISA

Date of Birth : 29 APRIL 1994

Title : PET FEEDER WITH NODEMCU USING MOBILE
APPLICATION

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)

940429-05-5371
Date: 12 DEC 2018

DR. WAN ISNI SOFIAH
Date: 12 DEC 2018

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.

Author's Name	MUHAMMAD SODIQ AMIN BIN MOHAMD ISA
Thesis Title	PET FEEDER WITH NODEMCU USING MOBILE APPLICATION

Reasons	(i)
	(ii)
	(iii)

Thank you.

Yours faithfully,

(Supervisor's Signature)

Date: 12 DEC 2018

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.



SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project, and, in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Computer Systems & Networking).

(Supervisor's Signature)

Full Name : DR WAN ISNI SOFIAH BINTI WAN DIN

Position : LECTURER

Date : 12 DECEMBER 2018



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 : MUHAMMAD SODIQ AMIN BIN MOHAMD ISA

ID Number : CA15052

Date : 12 DECEMBER 2018

PET FEEDER WITH NODEMCU USING MOBILE APPLICATION

MUHAMMAD SODIQ AMIN BIN MOHAMD ISA

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

Faculty of Computer Systems & Software Engineering

UNIVERSITI MALAYSIA PAHANG

DECEMBER 2018

ACKNOWLEDGEMENTS

“In the Name of ALLAH, Most Gracious, Most Merciful and big Gratitude to Prophet Muhammad SAW”

Alhamdulillah, praises and thanks to Allah because of His Almighty and His utmost blessings, I was able to finish this project within the time duration. Firstly, my special thanks to my supervisor, Dr Wan Isnii Sofiah binti Wan Din for the ideas, patience, motivated advices, constructive criticism, spending times, morale support and guiding me throughout every process in completion of this project.

Special appreciation also goes to my beloved parents and whole family especially my mother, Rosmawati binti Abdul Rashid for her understanding and always prays for my success to reach this path of study.

Last but not least, I would like to give my gratitude to all my lecturers and my entire classmate friends that involve in discussion, sharing idea and knowledge, brainstorming together and also help and support me to complete my every task on this project.

May Allah S.W.T bless all of you. Thank You.

ABSTRAK

Pada masa kini, haiwan peliharaan dianggap sebagai teman hidup. Walau bagaimanapun, mengurus haiwan peliharaan kadangkala adalah tugas yang berat untuk pemiliknya terutama apabila tiada seorang pun di rumah. Kebanyakan pemilik haiwan kesayangan tidak boleh tinggal di rumah setiap kali untuk memberi makan haiwan mereka dengan kerap, namun haiwan kesayangan mereka perlu diberi makan sepanjang hari. Oleh itu, dengan pembangunan Pemberi Makanan Haiwan menggunakan NodeMCU dan Aplikasi Mudah Alih akan mengatasi masalah ini. Sistem ini akan membantu dan membenarkan pemilik dari jauh memberi makan haiwan mereka pada masa tertentu tanpa perlu hadir di lokasi pemakanan. Bagi membangunkan system ini, kaedah yang digunakan ialah Air Terjun Iteratif. Kaedah ini dibahagikan kepada empat fasa bermula dari Analisis Keperluan, Reka Bentuk, Pelaksanaan dan Pengujian. Metodologi Air Terjun Iteratif akan menghasilkan perisian yang berfungsi dengan cepat dan awal dalam perisian kitaran hayat Seterusnya, haiwan kesayangan dapat menikmati makanan mereka tanpa kehadiran pemilik sementara pemilik dapat menikmati kerja dan perjalanan tanpa perlu kluatir tentang memberi makan haiwan peliharaan mereka. Projek ini berjaya di bangunkan yang membolehkan pemilik memberi makanan haiwan mereka jauh dari rumah dan haiwan peliharaan dapat menikmati makanan pada waktu yang tepat.

ABSTRACT

Nowadays, pets are considered as a life companion. However, managing pets sometimes is such a nuisance task for the owner especially when no one is at home. Most pet owners cannot stay at home every time to feed their pet regularly, yet their pets need to be fed throughout the day. Therefore, with a development of Pet Feeder with NodeMCU using Mobile Application will overcome this problem, this system will assist and allow owner to remotely feed their pets at a specific time without having to present at the feeding location. Iterative Waterfall method has been used to develop this system. This method is divided into four phases starting from Requirement Analysis, Design, Implementation and Testing. Iterative waterfall methodology will produce software that works quickly and early in the life cycle software. Next, pets can have their food without the present of owners while owners can enjoy works and travelling without having worries about feeding their pets. This project has been successfully developed which enable owners to feed their pets remotely away from home and pets gets food at the right time.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x
CHAPTER 1 INTRODUCTION	1
1.1 INTRODUCTION	1
1.2 PROBLEM STATEMENT	2
1.3 OBJECTIVE	2
1.4 SCOPE	3
1.5 REPORT ORGANIZATION	3
CHAPTER 2 LITERATURE REVIEW	5
2.1 INTRODUCTION	5
2.2 EXISTING SYSTEMS	5
2.2.1 Smart Dog Feeder Design Using Wireless Communication, MQTT and Android Client	5
2.2.2 GEMPET PF-103 SmartFeeder	7
2.2.3 Automatic Fish Feeder System Using PIC Microcontroller	8

2.3	COMPARING THE EXISTING SYSTEM	9
2.4	CONCLUSION	10
CHAPTER 3 METHODOLOGY		11
3.1	INTRODUCTION	11
3.2	PROJECT METHODOLOGY FRAMEWORK	11
3.3	OVERVIEW OF METHODOLOGY	13
3.4	REQUIREMENT GATHERING AND ANALYSIS	14
	3.4.1 Feasibility Study	14
	3.4.2 Hardware and Software Comparison	15
3.5	DESIGN	18
	3.5.1 Context Diagram	19
	3.5.2 Data Flow Diagram	20
	3.5.3 Hardware and Software Specification	21
	3.5.4 General Process Flow Diagram	26
3.6	IMPLEMENTATION	27
	3.6.1 Project Timeline	28
3.7	PROTOTYPE	28
3.8	TESTING	28
3.9	SUMMARY	29
CHAPTER 4 RESULTS AND DISCUSSION		30
4.1	INTRODUCTION	30
4.2	IMPLEMENTATION REQUIREMENTS	30
	4.2.1 Implementation of Database	31
	4.2.2 Implementation of Hardware	32

4.3	PROJECT TESTING	36
	4.3.1 Development of Mobile Application	36
4.4	CONCLUSION	40
4.5	SUMMARY	40
	CHAPTER 5 CONCLUSION AND RECOMMENDATIONS	41
5.1	INTRODUCTION	41
5.2	LIMITATION OF THE PROJECT	41
5.3	RECOMMENDATION	41
5.4	CONCLUSION	42
	REFERENCES	43
	APPENDIX A Gantt Chart	45
	APPENDIX B Coding	46
	APPENDIX C Prototype Pet Feeder	54

LIST OF TABLES

Table 2.1	Comparisons between Existing Systems	9
Table 3.1	Project Methodology Description	13
Table 3.2	Comparison of Servo Motor	15
Table 3.3	Comparison of Power Supply	16
Table 3.4	Comparison of Device.	16
Table 3.5	Comparison of Sensor.	17
Table 3.6	Comparison of Mobile Platform.	17
Table 3.7	Comparison of Code Implementation.	18
Table 3.8	Hardware Requirement Description	21
Table 3.9	Software had been used	25

LIST OF FIGURES

Figure 2.1	Diagram of Smart Dog Feeder.	6
Figure 2.2	GEMPET SmartFeeder.	7
Figure 2.3	Mechanical Design of the system.	8
Figure 3.1	Iterative Waterfall Methodology.	12
Figure 3.2	Context Diagram for Pet Feeder	19
Figure 3.3	Data Flow Diagram (DFD) for Pet Feeder	20
Figure 3.4	NodeMCU ESP8266 Wi-Fi Controller Board ESP-12.	22
Figure 3.5	NodeMCU GPIO pin.	23
Figure 3.6	Servo Motor MG995.	23
Figure 3.7	Servo Motor rotation calculation.	24
Figure 3.8	Battery 6V Holder.	24
Figure 3.9	Ultrasonic Sensor HC-SR04.	25
Figure 3.10	Pet Feeder data flow diagram.	27
Figure 3.11	Project timeline.	28
Figure 4.1	Firebase Get Started Web Page.	31
Figure 4.2	Firebase Authentication Data.	32
Figure 4.3	Pet Feeder connected circuit.	33
Figure 4.4	Arduino IDE Windows Application	34
Figure 4.5	Void setup program code.	35
Figure 4.6	Void food_qty_state() functions.	35
Figure 4.7	Homepage.java code in Android Studio.	36
Figure 4.8	First page for Mobile Application.	37
Figure 4.9	Homepage for Mobile Application.	38
Figure 4.10	Add Pet info page.	39
Figure 4.11	Feeding timer for pet.	39

LIST OF ABBREVIATIONS

PWM	Pulse Width Modulation
GPIO	General-Purpose Input Output
RM	Ringgit Malaysia
DC	Duty Cycle
MQTT	Message Queuing Telemetry Transport
RFID	Radio-frequency Identification
JSON	JavaScript Object Notation
SSL/TLS	Secure Sockets Layer / Transport Layer Security
IOS	iPhone OS
CPU	Central Processing Unit
PIC	Peripheral Interface Controller
V	Voltage
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
QFN	Quad-Flat No-Leads
OS	Operating System
RAM	Random-Access Memory
TB	Terabyte
HDD	Hard Disk Drive
IDE	Integrated Development Environment
DFD	Data Flow Diagram

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Cats are the most popular pets (Hodgson & Barton, 2015) among other animal where people having them as a companion in daily life. There are many types of cats in this world, such as British Short Hair, American Short Hair, Bengal cat, Persian cat and many more. Number of cats that being pet are increase from day by day especially in Malaysia.

Many commitments are requested to keep pets while there are families do not always have the right knowledge to manage them (Russo, Vergnano, Bergero, & Prola, 2017). Part of it are concern, keeping them company, and the important one is feeding pets in the right way and on the time. However, taking care of pet's diet can be hard and time consuming because not everyone is a pet expert. One of the top health concerns of pets are overeating and obesity. Just as in humans, excessive fat in pets increases the risk of often-preventable health conditions (Increase et al., 2015). Especially at younger age, they are usually satisfied with whatever much is given to them.

Pet Feeder with NodeMCU using Mobile Application is the system that can help the pet's owner to feed their pet either using mobile application or manual feeding. This system also has function that can feed pet, schedule feed, save pets name and check the quantity of food left on the container.

1.2 PROBLEM STATEMENT

In this globalization era, people are tending to spend most of their plenty time outside their house. A companion like a pet are seem needed in our daily life as a stress takes a toll on our life. Such an irony, a pet can company their owner and fulfil their need, but the owners themselves are not able to prepare their pets' priority such as food and water. Pet owners had been facing with the problem due to how they can feed their pet properly when they are not at home.

Pet owners are being warned that having a pet means they have contributed extra commitment to give the best support and care to them. Based on industry revenue, there are increasing demand in Asian markets for quality pet food (Balzer, 2015). Unfortunately, there are certain pet owners did not have much energy or time to fulfil their pets' food because owners might not always be at home regularly. Their personal plan is like travelling and working, will cause them dilemma. The dilemma deal by the pet owners can tend to unforeseen animal abuse due to lack of feeding. Other than that, pet owners may find themselves making troubling their families and friends to babysit their pet while they are away, which manage be unsuited after some haunt requests.

The Pet Feeder with NodeMCU using Mobile Application will be able to help both owners and their pets happy. It is designed to make sure the problem that all pet owners are facing will be ended. It may provide awareness to pet owners of its advantages and feasibility in the future, with the opportunity that similar Pet Feeder with NodeMCU using Mobile Application are not common in Malaysia.

1.3 OBJECTIVE

Goals that are intended to be achieved on this project is to develop a Pet Feeder with NodeMCU using Mobile Application. The main objective are the specific actions that need to be achieved on this research which deeply focus on aspect as below:

- (i) To investigate the existing product of pet feeding.
- (ii) To design Pet Feeder with NodeMCU using Mobile Application.
- (iii) To evaluate the performance of the Pet Feeder with NodeMCU using Mobile Application.

1.4 SCOPE

Scope is a principal element to make sure this project achieves their objective. Moreover, scope of this project has been agreed upon discussion. The agreed scope of this projects is:

(i) User Scope:

For the user scope, it was limited for the cat only. It is because Pet Feeder with NodeMCU using Mobile Application must be tested on the pet, and the only pet that available to be testes are cat only.

(ii) Function available:

a. Manual feeding

Pet owner can feed their pet manually when they are in the home with only pressing the button on Pet Feeder.

b. Automatic feeding.

Pet owner can feed their pet automatically when they are not in the home using mobile application.

c. Schedule feeding.

Pet owners can feed their pet schedulable when there are in or not in the home to prevent from forget to feed them.

1.5 REPORT ORGANIZATION

This project consists of five chapters. Chapter 1 consist of introduction about this project and continue with the problem statement which identified where it leads to the development of the solution system discusses on the project. Then, it will follow with the objective of this project. Lastly is the scope and thesis organization about this project.

Chapter 2 discusses on the literature of the project. Literature review will be explained in more details about this system that includes the introduction, method and techniques or technologies which are suitable to be used into this project.

Chapter 3 is about the project methodology that will be used to develop to this research. Furthermore, this chapter will discuss the software or hardware that will be used to implement to this project. It also discussed how to analysis and design of this project.

Chapter 4 discusses on the implementation, testing and result discussion. This chapter explains on how the system been implemented according to the requirements. This chapter then discusses the test cases done for this system to ensure that the systems requirement is verified and validated so that it is free from any bug. The test results then are stated here.

Chapter 5 discusses on the conclusion of the project. This chapter includes the introduction which concludes the project that have been done, research constraints which includes the clarification on the constraints throughout the project and the future work of this prototype system.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter is about literature review which is focusing on technology consideration and related works. Literature review is about any article or papers which involve the exploration topic in an effective technique. This chapter has cover on the Pet Feeder background information. The development of a pet feeder has been described and discussed in detail to assist readers in their understanding of the project. Moreover, discussion about pet feeder has been described further on. Some related works have also been presented to the readers to share some past and recent progress made. After reviewing many articles and journal, many of researches and papers are collected as references.

2.2 EXISTING SYSTEMS

Existing systems refers to another project developed by others. It can be any commercialized products that related to this project regardless of its technology. It is used to make comparison between the existing product and the product that will develop later. The comparison is being viewed by the advantage and disadvantage of the product.

2.2.1 Smart Dog Feeder Design Using Wireless Communication, MQTT and Android Client

Smart Dog Feeder provides general feeding process without troublesome the owner. Owners can control the process of feeding using Android smartphones remotely. In Figure 2.1, Smart Dog Feeder can authenticate pet using RFID technology, monitor the time of feeding and portion. Android smartphone is the platforms where the owner can set the feeding time, portion, waiting time and stock that require least requirement of Android's latest operating systems, along with the other application installed. MQTT

protocol is used by this feeder that has info about data stock, timetable, time, name of the owner from server.

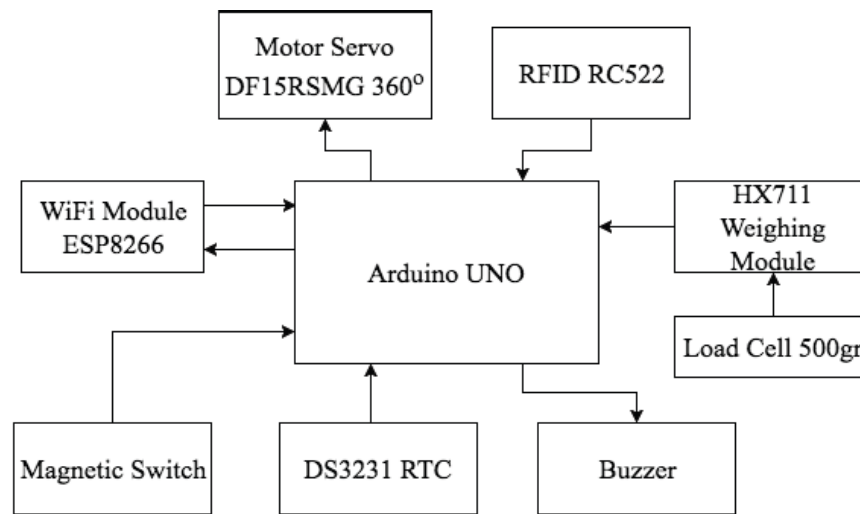


Figure 2.1 Diagram of Smart Dog Feeder.
Source: Vania, Karyono & Nugroho (2017)

The data will be directed in the form of JSON and controlled by this feeder, server and Android. This feeder interconnects with server through ESP8266 Wi-Fi module circuit boards, NodeMCU, along with MQTT protocol over SSL/TLS. This feeder also will record the timetable and trigger an alert that can intercept when feeding time is due. Validation method is done by the RFID tag who done the checking that attached on dog's collar.

Strengths of Smart Dog Feeder are;

- 1) Allow owner to remotely feed the dog without having to present at the feeding location using Android smartphone.
- 2) The system has the function of RFID to authenticate dog, set feeding time and portion which ease owner in feeding process.

However, this Smart Dog Feeder system only supports Android smartphone lower version. Therefore, it is clearly defined that this product does not compatible with the Android newest version. Thus, owner who does not have latest Android version smartphone cannot use of this Smart Dog Feeder system.

2.2.2 GEMPET PF-103 SmartFeeder

Refer to Figure 2.2, GEMPET SmartFeeder allows user to program and control the automatic feeder anytime from anywhere with user's smartphone. First, users must download the free SmartFeeder App which is compatible with Apple IOS, Androids and tablets. GEMPET SmartFeeder works even if it is disconnected from Wi-Fi, it will automatically change the setting to "Offline Feeding Model" to serve the food according to the programmed schedule.



Figure 2.2 GEMPET SmartFeeder.
Source: Jempet (2018)

The unit dispenses at least of 30 grams and up to 100 grams. The application allows user to monitor the quantity of food dispensed each time. The feeder features a real-time camera, so user can lookout and communicate with their pet anytime from everywhere. The 1-megapixel camera archives your pet's movement at any time. The feeder is appropriate for users who currently using smartphone with iOS 7.0 and above or Android 4.0 and above. It is recommended for owner who is feeding a single cat which is not too obsessed with food and only operates with sized rounded kibble which is the disadvantage of this feeder.

However, this feeder does not work well with big shaped kibble indicate that this feeder is only applicable for small size of pet. This is due to chute may get blocked if the owner feed a big meal, it because of kibble's size is too big.

2.2.3 Automatic Fish Feeder System Using PIC Microcontroller

This device developed combines mechanical and electrical system in controlling fish feeding activity as shown in Figure 2.3. The pellets controlled by DC motor which located under the pellet storage. A control system was then attached to this device allowing the fish to be fed at the right cycle time as required or predefined by user or entrepreneur. Timer was employed in this device to control the motor rotation attached to sphere former, which dispense the pellets into the water. The pellets dispensed into the marking area of the pond based solely on the rotation speed of the motor itself. The controller came with a keypad giving user more option in determining the suitable speed for the motor depends on their cattle.

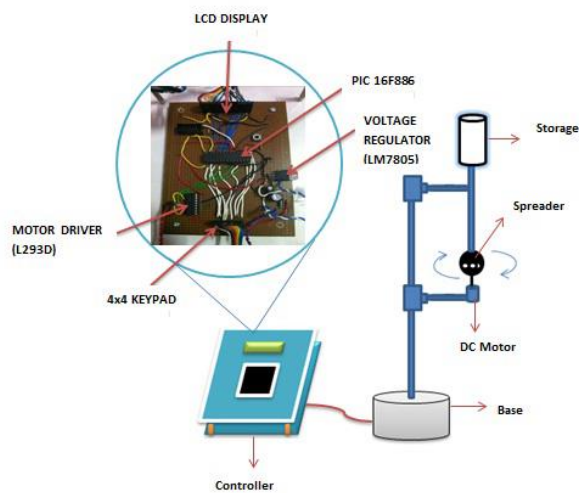


Figure 2.3 Mechanical Design of the system.

Source: Noor, Hussian, Saaid, Ali, & Zolkapli (2012)

The design of this system comprised four main part namely main controller, pellet storage, stand and spreader. The controller of this system, a 4x4 Keypad functional as input device which provided the user abilities to set timer and motor speed to spread the pellet into the water. Apart from that, LCG display played a significant role in illustrating the data entered by the user before DC motor start to operate. PIC 16F886 controller was employed as main controller output of DC motor. L293D motor driver which work with PWM (Pulse Width Modulation) technique are to control the speed rotation of DC motor. All appropriate components used in hardware design stage were assembled to set up the feeding device. This system was simple in construction and operation, also relatively inexpensive.

The advantages of this device are it is effective because easy to feed the fish at the right cycle time. These devices also reduce the owner to hire more workers and reduce the time needed.

2.3 COMPARING THE EXISTING SYSTEM

The comparisons between Smart Dog Feeder Design Using Wireless Communication, GEMPET PF-103 SmartFeeder and Automatic Fish Feeder System Using PIC Microcontroller are shown in Table 2.1.

Table 2.1 Comparisons between Existing Systems

No.	Project Title	Similarities	Differences
1	Smart Dog Feeder Design Using Wireless Communication	Allows owner to remotely feed the pet without having to present at the feeding location	Existing project: This system only supports Android smartphone lower version. Proposed project: Supports lower and newest Android operating system on smartphone.
2	GEMPET PF-103 SmartFeeder	Allows user to control the automatic feeder from anywhere by using a smartphone application	Existing project: Does work well with big shaped kibble, only applicable for small size of pet. Proposed project: Compatible to all type of pet
3	Automatic Fish Feeder System Using PIC Microcontroller	Allow user to set timer to automatically feed the fish	Existing product: Does not have mobile application to control the timer. Proposed project: Build the mobile application to easier owner set timer.

2.4 CONCLUSION

In conclusion, this chapter shows the existing technology that is related to Pet Feeder development. Based on the research conducted, vital problem faced by the owner such as owner cannot feed their pet remotely. The advantage of this research is many webpages provide adequate functions as many owners need. Based on the analysis on related works, this pet feeder has been widely developed but improvements can be done by using different method, hardware and software. Study on existing hardware and software indicates that it is suitable for daily use and user friendly.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter cover the methodology used to achieve the objective of proposed project which is to develop a Pet Feeder with NodeMCU using Mobile Application. Iterative waterfall model was chosen as this project methodology approach which consist of requirement analysis, design, implementation and testing phase. Apart from discussing about the method used, this chapter also discuss about how the research is going to be conducted hence the flow and the progress of the project.

3.2 PROJECT METHODOLOGY FRAMEWORK

The project methodology framework used for this project is Iterative Waterfall Model. It is divided into four phases starting from Requirement Analysis, Design, Implementation and Testing. Winston W. Royce first described the Systems Development Lifecycle based on the Waterfall Model in 1970 (Kramer, 2018). This model provided practical guidelines for developing software product and became popular. Its name is derived from structural specification. Every phase comes after a phase is completed and tasks can be divided according to phases. The output of one phase becomes input of next phase but have the option to revisit phases in the next cycle (Trivedi & Sharma, 2013).

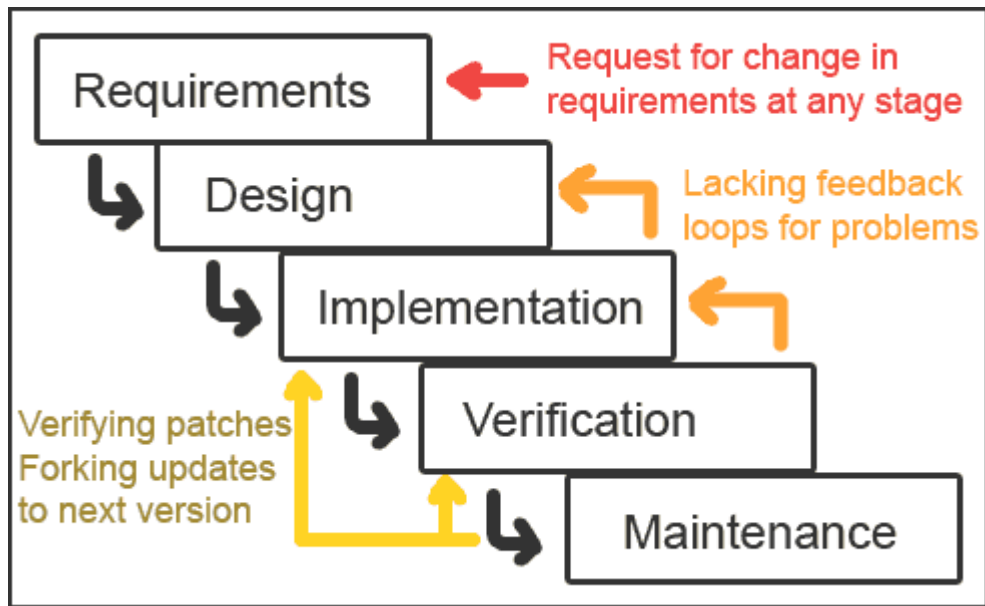


Figure 3.1 Iterative Waterfall Methodology.
 Source: Tony. (2018, April 14)

Iterative waterfall methodology was chosen of this project methodology process. This methodology preferred in this project is because it is appropriate and suitable since it allows a turn back to previous phase when it detects error or needs some troubleshooting. By referring to Figure 3.1, it consists of repeating the four phases which are Requirement Analysis, Design, Implementation and Testing.

The advantage of using this iterative waterfall model is its flexibility since it does not require much cost to change scope and requirements. Moreover, each of the iteration is easy managed milestone. Other than that, by using this iterative waterfall methodology as the methodology method, when preparing or developing the project, it is easy in terms of testing, troubleshoot during a smaller phase. For the risk, it is easy to handle. Lastly, iterative waterfall methodology will produce software that works quickly and early in the life cycle software (Kumar, 2015).

3.3 OVERVIEW OF METHODOLOGY

Table 3.1 shows the description for each phase of Iterative Waterfall Model methodology and its deliverables.

Table 3.1 Project Methodology Description

Phase	Tasks	Deliverables
Requirement Gathering and Analysis	<ul style="list-style-type: none"> Gathering essential information and analyse it regarding on the proposed project. Do the feasibility test to make sure that the requirement is testable. 	<ul style="list-style-type: none"> Project background Problem statements Objectives Scope of project Techniques to be used in the project List of appropriate software and hardware to be used
Design	<ul style="list-style-type: none"> Capture appropriate software and hardware to be used in this project. The flow of the project including the project flow chart and use-case diagram. 	<ul style="list-style-type: none"> Flow chart Use-case diagram
Implementation	<ul style="list-style-type: none"> Implement the program with code with the selected software and 	<ul style="list-style-type: none"> Designing tasks up to this point that integrate the code and

	hardware.	implement into the project.
Testing	<ul style="list-style-type: none"> • Testing the project to identify if there are errors or issues that need to be fix 	<ul style="list-style-type: none"> • The project is finish and ready to be used by the user.

3.4 REQUIREMENT GATHERING AND ANALYSIS

This is the process of identifying the necessary and related components for the proposed project. This phase is about the understanding of the whole project and gathering the information as much as possible that might support on developing this project such as articles and journals. The objective is to collect all related work and information and find the suitable hardware, software thus making the system more efficient. This phase can determine the best method, tools and techniques to use in developing this project. This phase also can be a guideline while developing this project so that there will be no duplication of work happen. Various data and researches have been collected from articles, book and online journals to help in developing the proposed project including:

1. Collecting information regarding the components when developing the project, along with the software and hardware to be used.
2. Select and search the best development method for the proposed project.

3.4.1 Feasibility Study

Feasibility study is conducted to study and whether the proposed project can be developed or not. The proposed project is specifically to ease the owner in terms of feeding their pet. Owner often cannot remotely feed their pet when they are not at home thus it causes many problems to the owner and the pet itself. This project is implemented to allow owner to remotely feed their pet by using a mobile application from Android platform.

3.4.2 Hardware and Software Comparison

This section will compare the related hardware and software available in the market in terms of its model and features. Before starting to develop this project, the right hardware and software must first be selected as there are many options available. Owner's requirements and budgets must also be considered when selecting the appropriate hardware and software.

a) Hardware

i. Servo Motor

Table 3.2 show the comparison between the Servo Motor available.

Table 3.2 Comparison of Servo Motor

Hardware	Features	Decision
Servo Motor SG90	Speed: 4.8v, 0.12sec/60 degree Weight: 9 grams Cost: RM 20	Servo motor is used in this project to release food from the food container. Therefore, it requires a power and strong and full rotation to release the food. Thus, Servo Motor MG995 is selected.
Servo Motor MG99	Speed: 4.8v. 0.23sec/360 degree Weight: 37 grams Cost: RM 22.50	

ii. Power Supply

Table 3.3 shows the comparison between power supplies available.

Table 3.3 Comparison of Power Supply

Hardware	Features	Decision
Power Supply 12V DC adapter	Power: 12V Requires plug. Cost: RM 19	Because of this project is indoor, low power supply which battery is suitable because its low cost.
Battery 6V Holder	Power: 1.5V x 4 Requires battery slot. Cost: RM 10	

iii. Device

Table 3.4 show the comparison between the device available.

Table 3.4 Comparison of Device.

Hardware	Features	Decision
SIM900 Board for Arduino	Work on GSM network GPRS multi-slot class 10/8 Cost: RM 30	Because of this project is outdoor and indoor, NodeMCU ESP8266 Wi-Fi Controller Board ESP-12 is suitable because it requires Wi-Fi to connect with it.
NodeMCU ESP8266 Wi-Fi Controller Board ESP-12	32-pin QFN package On-chip low-dropout Cost: RM 20	

iv. Sensor

Table 3.5 show the comparison between the sensor available.

Table 3.5 Comparison of Sensor.

Hardware	Features	Decision
Ultrasonic Sensor HC-SR04	Input Voltage: 5V. Current Draw: 20mA (Max) Digital Output: 5V. Cost: RM 17	Because of this project is requires low cost, Ultrasonic Sensor HC-
Grove Ultrasonic Ranger	Input Voltage: 5V. Current Draw: 30mA (Max) Digital Output: 5V. Cost: RM 18	SR04 is suitable.

b) Software

i. Mobile Platform

Table 3.6 show the comparison between the mobile platform available.

Table 3.6 Comparison of Mobile Platform.

Software	Features	Decision
Android Platform	Have open source. Example: Android Studio. Easy to implement.	Because used Windows OS version to do this project, rather than use
iOS Platform	Must buy the platform first. Example: XCode. Difficult to implement.	iOS platform, it was clearly Android platform that suitable to build this project.

ii. Code Implementation

Table 3.7 show the comparison between the code implementation available.

Table 3.7 Comparison of Code Implementation.

Code Implementation	Features	Decision
Arduino IDE	Multi-Platform Application. Have user GUI if run in Windows. Easy to implement.	Because using Windows OS version to do this project, it
Python	Multi-Platform Application. Must run in cmd if run in Windows. Difficult to implement.	was clearly Arduino IDE platform that suitable to build this project.

3.5 DESIGN

In this phase, the requirement of specifications from the requirement analysis phase is studied thus the project design is prepared. There are two main parts that act as the integrity of the project which are data recording and data analysis. Data recording consists contain of function to record all the data and information whereas data analysis is the process where data has been invoked from the storage and restored directly from the manual. The project design is important in order to specify the suitable hardware and software needed and help in defining the whole project architecture. The project design includes context diagram and data flow diagram. This system design plays a significant role as a guideline to get better understanding about the whole project.

3.5.1 Context Diagram

A context diagram is a diagram that defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it. This diagram is a high-level view of a system. Figure 3.2 shows context diagram for this project:

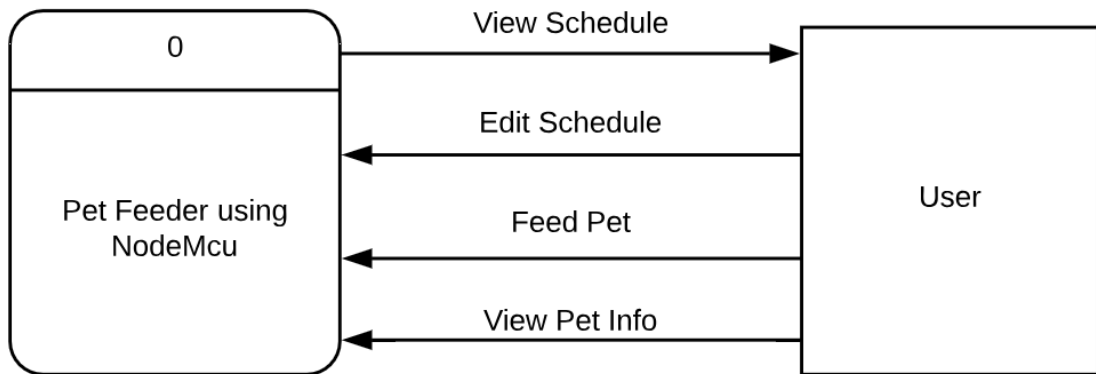


Figure 3.2 Context Diagram for Pet Feeder

3.5.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). Figure 3.3 shows DFD for this project:

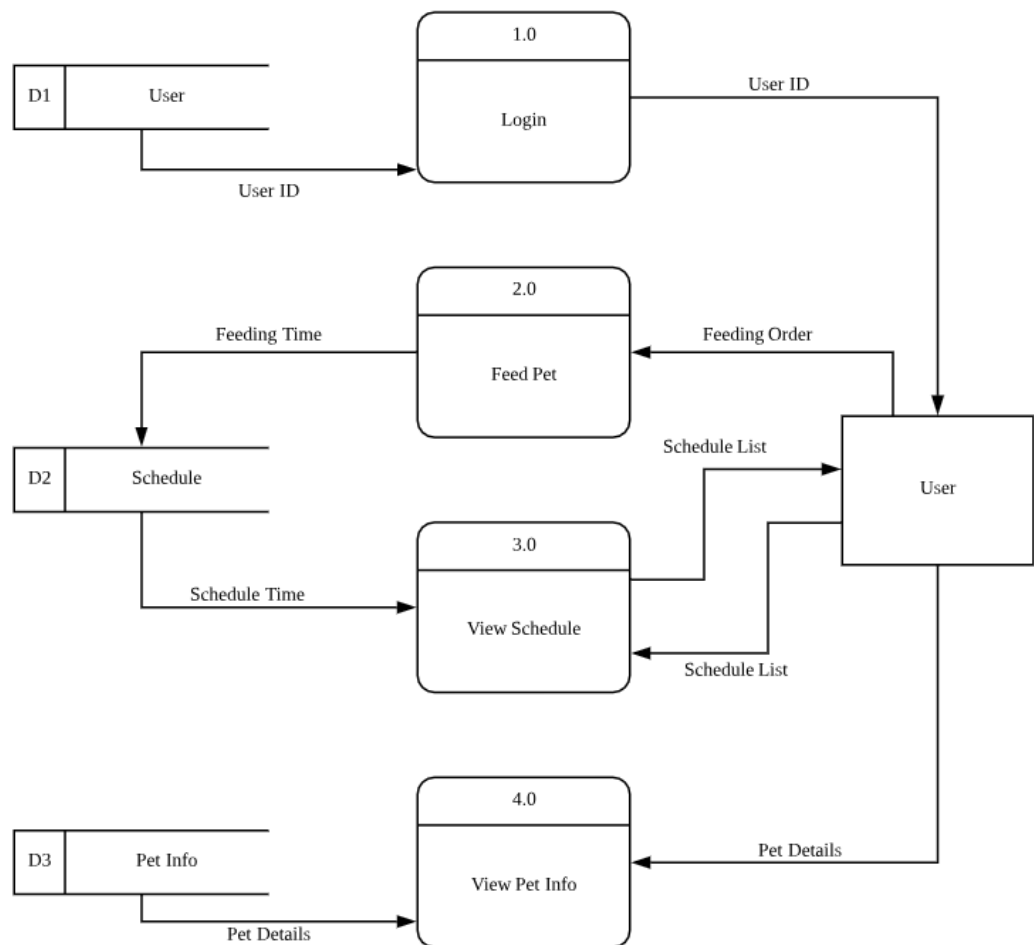


Figure 3.3 Data Flow Diagram (DFD) for Pet Feeder

3.5.3 Hardware and Software Specification

This section explains briefly the hardware and software selected in 3.3.1.2 with its price and specification. The most important requirements for this project are hardware and software. Hardware can be a memory, processor while the output devices can be a hard disk that needs to store a program or a monitor that display on screen. But hardware cannot work on its own, without software attached to it. Hence, in order to develop this project, both hardware and software are required.

Hardware requirement is important to carry out tools and developing application effectively and successfully. This project is currently using the latest hardware to make sure it is efficient. Table 3.8 shows the lists of hardware used in terms of price and its specification.

Table 3.8 Hardware Requirement Description

Hardware	Price (RM)	Function/Specification
Laptop	RM 2400 * 1 unit	<ol style="list-style-type: none"> 1. Intel Core i5 7th Gen 2. 12GB RAM 3. 1TB of HDD 4. Windows OS
Servo Motor	RM 22.50 * 2 unit	<ol style="list-style-type: none"> 1. Model: MG995 <ul style="list-style-type: none"> • Used to release the food from container and open lid.
Device	RM 20 * 2 unit	<ol style="list-style-type: none"> 1. Model: NodeMCU <ul style="list-style-type: none"> • Use to communicate with the mobile device to function the servo motor.
Sensor	RM 17 * 2 unit	<ol style="list-style-type: none"> 1. Model: HC-SR04 <ul style="list-style-type: none"> • Use to measure distance between

food left in the container, thus user can know when to refill the container.

A. Hardware Requirement

a) NodeMCU ESP8266 Wi-Fi Controller Board ESP-12

In this project, NodeMCU ESP8266 Wi-Fi Controller Board ESP-12 is used as the main system that acts as a microcontroller, it is shown on Figure 3.4. It used to allow other sensors to communicate with each other by its General-Purpose Input Output (GPIO) as shown in Figure 3.5 using jumper wires and breadboard.

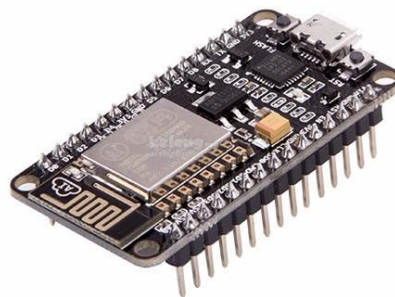


Figure 3.4 NodeMCU ESP8266 Wi-Fi Controller Board ESP-12.
Source: IoT LoLin NodeMCU ESP8266 (2018, April 17)

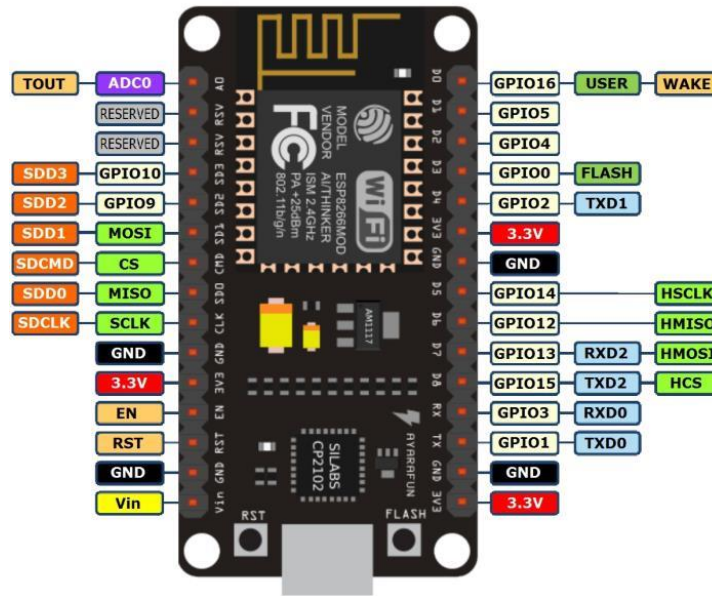


Figure 3.5 NodeMCU GPIO pin.
Source: GPIO pins & Network Access (2018, April 17)

b) Servo Motor

A servo acts like a motor that can rotate up to 180 degrees clockwise or anti-clockwise depending on what type of servo motor used. It has 3 pins which are Ground, Power and Signal coloured brown, red, and orange respectively. Is shown in Figure 3.6.



Figure 3.6 Servo Motor MG995.
Source: Componentes101 (2018, April 17)

There are four servos that will be used. Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM) through the

control wire. The functions of a servo motor in this project are to release food from the container and to open or close the lid for the pet to eat. To control the rotation, user must first calculate its duty cycle (DC). Refer to Figure 3.7 shows how calculation is made.

$$dc = \frac{\text{length}}{\text{period}}$$

$$dc = \frac{0.5}{20} \times 100 = 2.5\%$$

$$dc = \frac{1.5}{20} \times 100 = 7.5\%$$

$$dc = \frac{2.5}{20} \times 100 = 12.5\%$$

Figure 3.7 Servo Motor rotation calculation.
Source: RPI Labs (2018, April 17)

c) Power Supply

Power supply is used specifically for NodeMCU ESP8266 Wi-Fi Controller Board ESP-12. It is used to make sure NodeMCU can be function because NodeMCU need power supply to be used as shown in Figure 3.8.



Figure 3.8 Battery 6V Holder.
Source: 2018 Holder 6v Battery (2018, April 17)

d) Distance Sensor

Distance sensor is used to specific measure the remaining pet food in the container as shown in Figure 3.9. Therefore, user can know whether to refill their pet's food or not. Sensor will send detected distance to user's mobile application.



Figure 3.9 Ultrasonic Sensor HC-SR04.

B. Software Requirement

The software that have been used in developing this project is essential to make sure the Pet Feeder is functioning well along with the input and output correctly programmed. Table 3.9 shows the software that has been used.

Table 3.9 Software had been used

Software	Function / Specification
Android Studio	Android Studio is Android's official IDE. It is purpose built for Android to accelerate your development and help you build the highest-quality apps for every Android device. It offers tools custom-tailored for Android developers, including

rich code editing, debugging, testing, and profiling tools

C++ Programming Language C++ is a general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation.

3.5.4 General Process Flow Diagram

The flow chart diagram is the simplest way to show the flow about Pet Feeder using NodeMCU project along with its components and functions. Figure 3.10 shows Pet Feeder data flow diagram. Firstly, the Mobile Application will send request to NodeMCU. After that, it will check whether the NodeMCU is ready or not which mean NodeMCU have successfully connected to Wi-Fi or not. If NodeMCU does not ready, then it will wait and keep send request to NodeMCU until successful. The NodeMCU will go to next process which is feed pet. There are two type of feeding pet through Mobile Application. It will check either it is schedule feed or not. If there are schedule feed, then NodeMCU will checking the schedule of the feeding pet. If it is valid schedule, server motor will serve food for pet. If the schedule is not valid, the servo motor does not respond. Besides, if it is not schedule feed which is automatic feeding, servo motor will serve pet food when user push button on Mobile Application:-

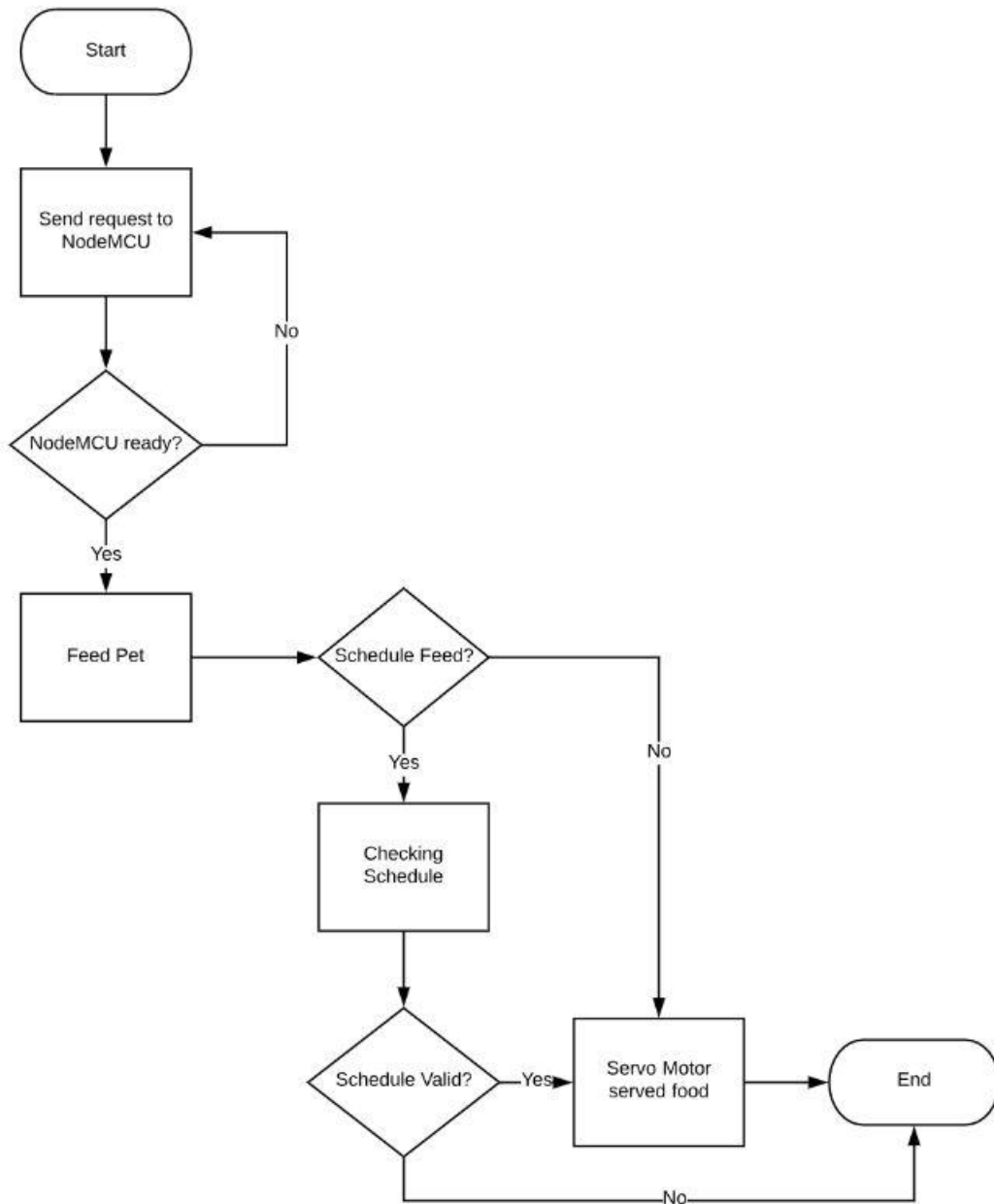


Figure 3.10 Pet Feeder data flow diagram.

3.6 IMPLEMENTATION

This phase is where the development started. In this phase, the proposed project has been developed to meet the project objectives. From the proposed project design that was prepared, the developer started to code and develops the project based on the design a guideline. With the complete tasks in previous phase, the implementation phase can be done smoothly.

3.6.1 Project Timeline

This project timeline consists of task name, start date, finish date and duration of a project. It is an important milestone the developer must follow. Figure 3.11 shows the project timeline.

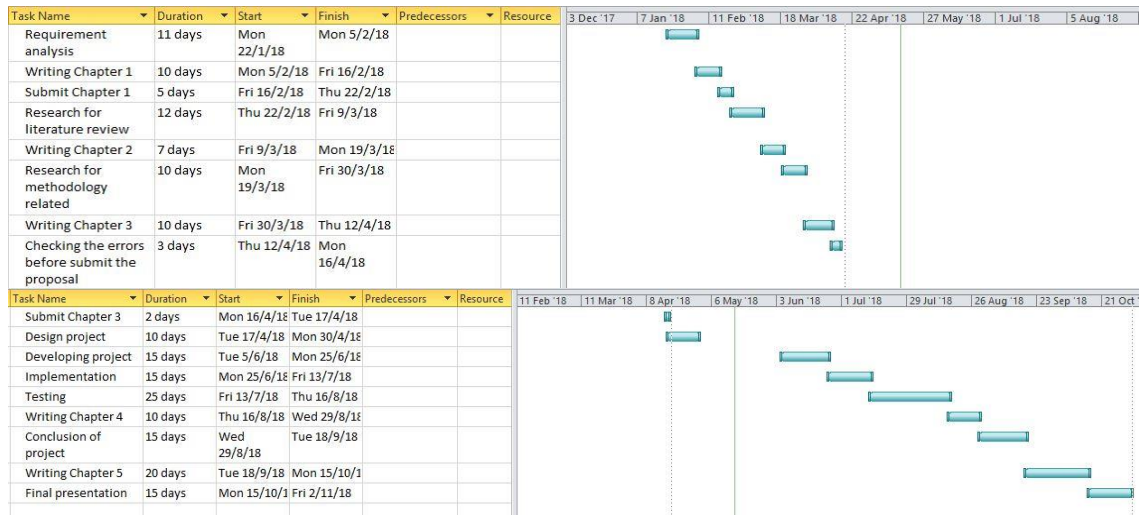


Figure 3.11 Project timeline.

3.7 PROTOTYPE

A prototype is an early sample, model, or release of a product built to test a concept or process or to act as a thing to be replicated or learned from. It is a term used in a variety of contexts, including semantics, design, electronics, and software programming. A prototype is generally used to evaluate a new design to enhance precision by system analysts and users. For Pet Feeder with NODEMCU using Mobile Application, the prototype has been made to see the result of this project.

(See Appendix C)

3.8 TESTING

Testing phase is the last phase in the methodology. Testing is needed after the project has been developed for the purpose to test all the function to ensure it works perfectly as planned. Moreover, while performing all the testing activities including functional and non-functional, to make sure the system meets the requirements and

objectives. This phase is important to verify if there is any error or bugs in the system. If there are any, the process must reverse back to the system design phase to fix the error. The testing phase is done to all components involved in the system. If it is error-free, the project is ready for users. Here is a test that is carried out to test the functionality of the project:

- i. Testing the functionality of Pet Feeder with NodeMCU.
- ii. Testing the functionality of NodeMCU including its components.
- iii. Testing the ultrasonic sensor whether is it working properly or not.
- iv. Testing the functionality of mobile application that will control the servo motor.
- v. Testing the functionality of servo motor.
- vi. For the testing target, it is tested to two trained pets.

3.9 SUMMARY

This chapter has explained about the research methodology selected in this project which is Iterative Waterfall Model which involves Requirement and Analysis, Design, Implementation and Testing phases. Each phase has its own function and duty and acts as a guideline by the developer in order to develop this project.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION

In this chapter, it discusses the functionality of the whole system that has been developed. The result and findings of the system are gained after all the functionality testing is conducted. This chapter analyses the results obtained from the testing phase of the pet feeder system along with the Mobile Application and Arduino; which include Servo Motor MG995, Battery 6V Holder, NodeMCU ESP8266 Wi-Fi Controller Board ESP-12, Ultrasonic Sensor HC-SR04, LM7805 Voltage Regulator, and Resistor that been used. This discussion also includes the results and finding that had been documented during the development and testing phase.

Moreover, it also will discuss about NodeMCU ESP8266 Wi-Fi Controller Board ESP-12 where it will send the sensor data by using Wi-Fi. This will make sure the NodeMCU ESP8266 Wi-Fi Controller Board ESP-12 will stay connected with the Firebase server. Then, Mobile Application that been developed will attire all the database from Firebase server and display all its data in Mobile Application. Therefore, user can get in touch to know the amount left in the container for feeding the pet and can directly remote or scheduling to feeding their pet.

4.2 IMPLEMENTATION REQUIREMENTS

In this phase, there are several stages that need to be complete before this project will be successful. It is including to setup the database to store the information of user where there will be sign up and login using this database. It will acknowledge the existing user or else user must sign up to use Mobile Application. In the nutshell, if user forgot their password, there will be interaction between Mobile Application and Firebase database for user to get their account back. Furthermore, developing this

project requires proper software and hardware installation where configuration also must be correct to ensure project can run without any problem.

First step is to setup the Firebase server by creating database. After that, setup hardware part for Servo Motor MG995, Battery 6V Holder, NodeMCU ESP8266 Wi-Fi Controller Board ESP-12, Ultrasonic Sensor HC-SR04, LM7805 Voltage Regulator, and Resistor before the programming take place for the first time. The pin should be connected properly to Analogue output, digital analogue, ground and voltage for every component to ensure this hardware part can be run smoothly. Finally, is developing Mobile Application for Pet Feeder with NodeMCU using Mobile Application by using the Android Studio.

4.2.1 Implementation of Database

For the beginning to start the development of Mobile Application, database is needed as important to store all the user data. For this project, Firebase database had been chosen as web server. Firebase is well known free webhosting because it can easily connect with the Android Studio.

Firebase must be registered in order to use its database. Figure 4.1 show the Firebase Get Started web page. If the button “Get Started” clicked, it will redirect to Google email for its user to sign up. As for system database, Firebase database is used as the database server of the system in order to store the data from the system.

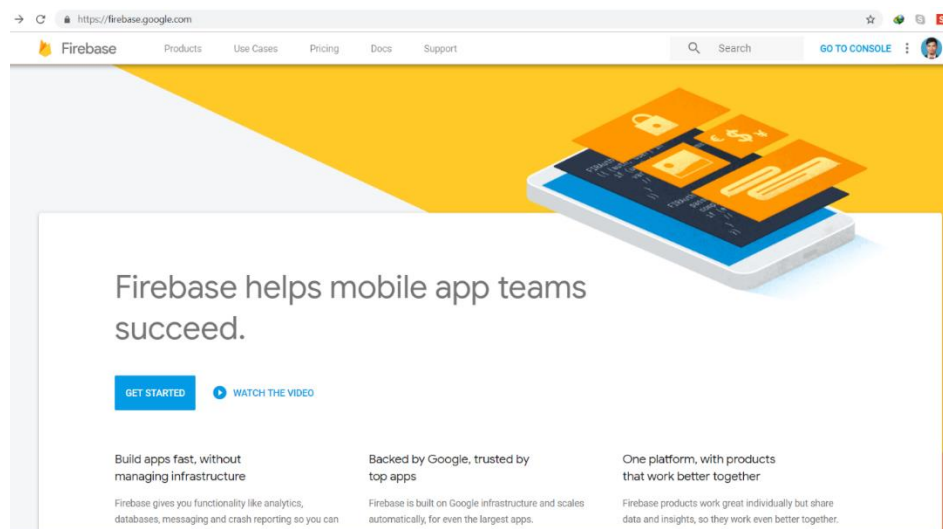
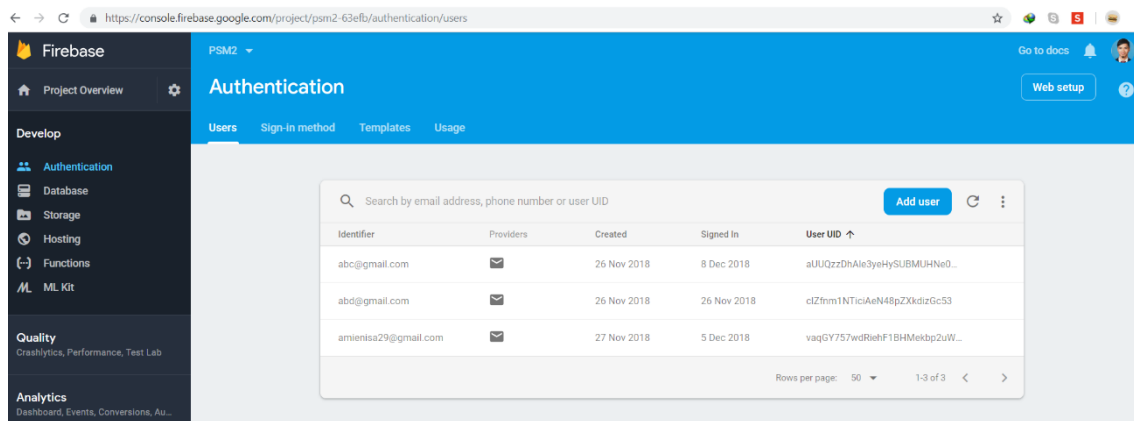


Figure 4.1 Firebase Get Started Web Page.

Figure 4.2 shows the Firebase Authentication Data which include the username and password of the user who had been sign up. This authentication data was used to enable user to use the Mobile Application or not. If the username and password does not match, user cannot go through to use the contain of the Mobile Application. Therefore, user must remember what username and password that they used for this project to be authenticated.



The screenshot shows the Firebase Authentication console interface. At the top, there is a search bar with the text "Search by email address, phone number or user UID" and an "Add user" button. Below the search bar is a table with the following columns: Identifier, Providers, Created, Signed In, and User UID. The table contains three rows of user data.

Identifier	Providers	Created	Signed In	User UID ↑
abc@gmail.com	📧	26 Nov 2018	8 Dec 2018	aUUQzzDhAle3yeHySUBMUHNe0...
abd@gmail.com	📧	26 Nov 2018	26 Nov 2018	clZfm1NTiciAeN48pZXkdiZGc53
amienisa29@gmail.com	📧	27 Nov 2018	5 Dec 2018	vaqGY757wdRiehF1BHMeKbp2uW...

At the bottom of the table, there is a "Rows per page" dropdown set to 50 and a "1-3 of 3" indicator with navigation arrows.

Figure 4.2 Firebase Authentication Data.

4.2.2 Implementation of Hardware

This project required several hardware to be done. It is Servo Motor MG995, Battery 6V Holder, NodeMCU ESP8266 Wi-Fi Controller Board ESP-12, Ultrasonic Sensor HC-SR04, LM7805 Voltage Regulator, and Resistor. First step is attached the NodeMCU ESP8266 Wi-Fi Controller Board ESP-12 to the Breadboard. Servo Motor MG995 was connect to the NodeMCU at the D05 pin. As the Servo Motor MG995 require 6V power to be working, it VCC pin were connect directly to the Battery 6V Holder which will give Servo Motor MG995 full 6V power to work. While the Ultrasonic Sensor HC-SR04 requires 5V to be operated, it requires LM7805 Voltage Regulator to be run smoothly. LM7805 Voltage Regulator will connect to the Battery 6V Holder where it will get 6V power, then it will regulate 5V power to whom needed. In this project, there are 2 hardware need to be regulated; which are Push Button and Ultrasonic Sensor HC-SR04. Ultrasonic Sensor HC-SR04 trigger D01 and echo D02 are attached to NodeMCU pin. Resistor is used to limit the amount of current going to Push Button to prevent from Push Button get over current.

Figure 4.3 shows the connection between all the hardware component used in the Pet Feeder with NodeMCU using Mobile Application. It is using Arduino IDE (Integrated Development Environment) compiler to program the NodeMCU ESP8266 Wi-Fi Controller Board ESP-12 and Firebase database via Wi-Fi.

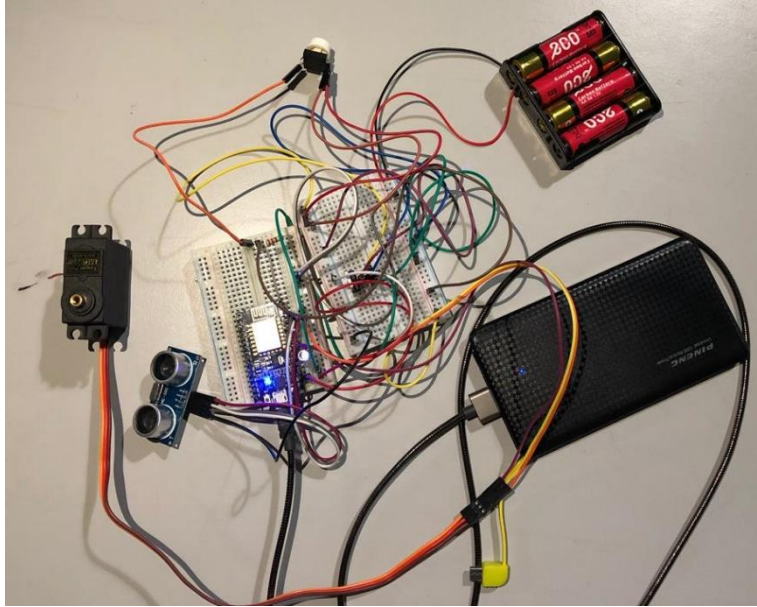


Figure 4.3 Pet Feeder connected circuit.

On the previous chapter, it was clearly stated the platform to build the program is NodeMCU where it used Arduino IDE. 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. Figure 4.4 shows Arduino IDE Windows Application.

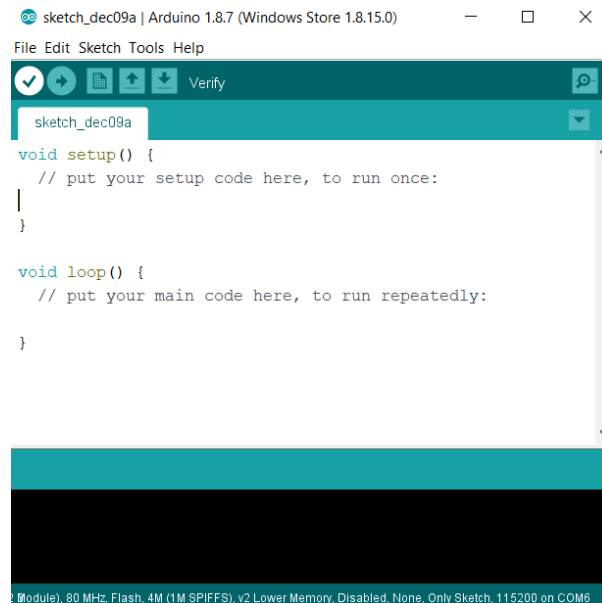


Figure 4.4 Arduino IDE Windows Application

This Arduino IDE application is based using C programming language, which is easy to understand and implement. Before starting program NodeMCU ESP8266 Wi-Fi Controller Board ESP-12, it is important to know what libraries to use in this system. In this project, libraries used is ESP8266, FirebaseArduino and NodeMCU-MQTT-ESP. 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 connect once. Then in void looping, it is for looping program such moisture sensor which is needing to be display by different value for every time it loops.

Figure 4.5 shows coding located in the void setup(). It also shows that the ssid and password for the Wi-Fi must be stated for NodeMCU to get connect with Wi-Fi. The initializePin() function will call the sensor which located on that function.

```

Verify
Pet_Feed$
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <Servo.h>
#define TRIGGER 5
#define ECHO 4

Servo servo;
//Our Wi-Fi ssid and password
char* ssid = "Amien iPhone";
char* password = "aminkacak";
String feed_1 = "0";
String feed_2 = "0";
String feed_3 = "0";
String food_qty = "0";

ESP8266WebServer server; //server variable

void setup() {
  initializePin(); //call function

  //Making Connection With network
  WiFi.begin(ssid, password);

  Serial.begin(115200);
  Serial.print("Searching Connection");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
}

```

Figure 4.5 Void setup program code.

Figure 4.6 show example on of code for the looping section which is food_qty_state() functions. This function reacts as the measurement of the quantity balance for food in the container and will send it through the Mobile Application. User can get to know the level in the container with this function by simplify checking in the Mobile Application.

```

Pet_Feed$
void food_qty_state() {
  if(food_qty == "0") {
    food_qty = "1";
    long duration, distance;
    digitalWrite(TRIGGER, LOW);
    delayMicroseconds(2);

    digitalWrite(TRIGGER, HIGH);
    delayMicroseconds(10); |

    digitalWrite(TRIGGER, LOW);
    duration = pulseIn(ECHO, HIGH);
    distance = (duration/2) / 29.1;

    Serial.print("Centimeter:");
    Serial.println(distance);
    delay(1000);
    server.send(200, "text/html", food_qty);
  } else {
    food_qty = "0";
    digitalWrite(D8, LOW);
    server.send(200, "text/html", food_qty);
  }
}

```

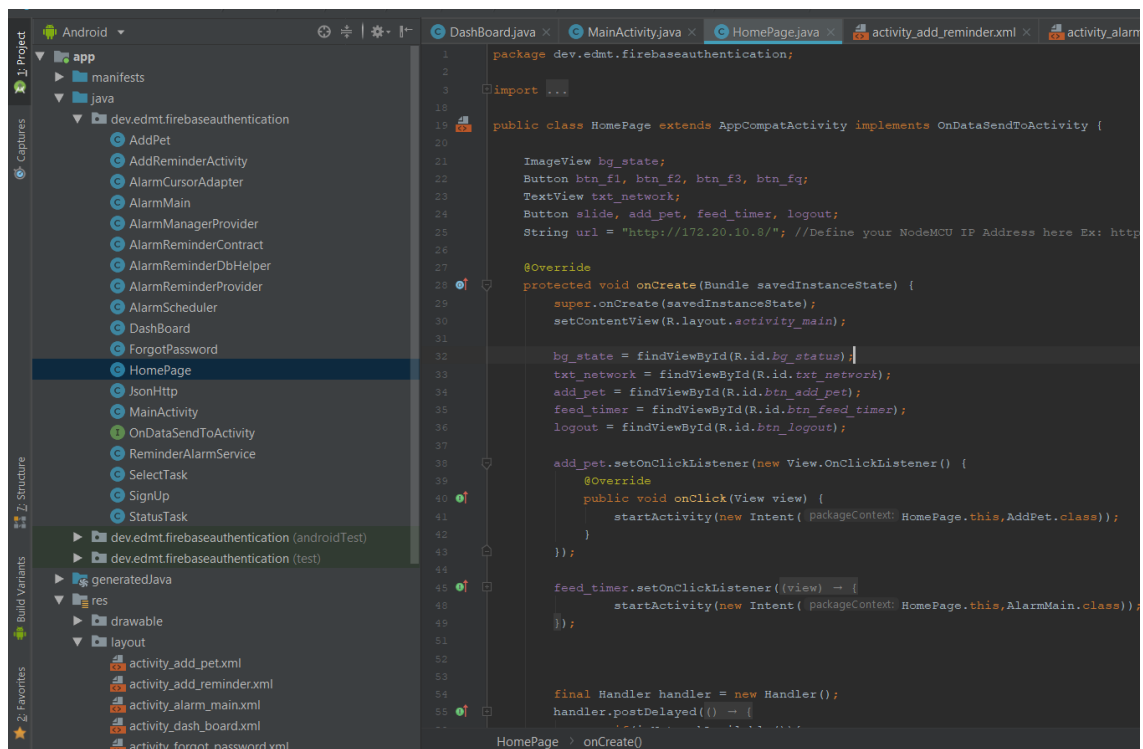
Figure 4.6 Void food_qty_state() functions.

4.3 PROJECT TESTING

This section explains the development of the Mobile Application and testing phase with the hardware and database that been implemented starting from scratch until the end of the development process. It also discusses about the coding along with its programming language and error occurs during the development process.

4.3.1 Development of Mobile Application

The last stage for Pet Feeder with NodeMCU using Mobile Application is development of Mobile Application. It was develop using Android Studio by using Java language. This Android Studio have their own user interface to make mobile application more beautiful and responsive. Figure 4.7 show example of Java class which is HomePage.java. In this class, it needs to declare and call the element in the layout so that the element can interact with the class. The result of this class when it be compiled, will be shown below.



```
1 package dev.edmt.firebaseauthentication;
2
3 import ...
4
16
17
18
19
20
21 ImageView bg_state;
22 Button btn_f1, btn_f2, btn_f3, btn_fq;
23 TextView txt_network;
24 Button slide, add_pet, feed_timer, logout;
25 String url = "http://172.20.10.8/"; //Define your NodeMCU IP Address here Ex: http
26
27
28 @Override
29 protected void onCreate(Bundle savedInstanceState) {
30     super.onCreate(savedInstanceState);
31     setContentView(R.layout.activity_main);
32
33     bg_state = findViewById(R.id.bg_status);
34     txt_network = findViewById(R.id.txt_network);
35     add_pet = findViewById(R.id.btn_add_pet);
36     feed_timer = findViewById(R.id.btn_feed_timer);
37     logout = findViewById(R.id.btn_logout);
38
39     add_pet.setOnClickListener(new View.OnClickListener() {
40         @Override
41         public void onClick(View view) {
42             startActivity(new Intent( packageContext: HomePage.this, AddPet.class));
43         }
44     });
45
46     feed_timer.setOnClickListener((view) - {
47         startActivity(new Intent( packageContext: HomePage.this, AlarmMain.class));
48     });
49
50
51
52
53
54 final Handler handler = new Handler();
55 handler.postDelayed(() = {
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Figure 4.7 Homepage.java code in Android Studio.

Figure 4.8 shows the first page of Mobile Application. In this page, user need to key in their username and unique password. Their username and password must be tallying to log into the system. However, if there are new user that use this Mobile Application, they must sign up to insert new data to database. Furthermore, there are button forgot password if the existing user forgot their created password. They can reset their password and get back with their account.

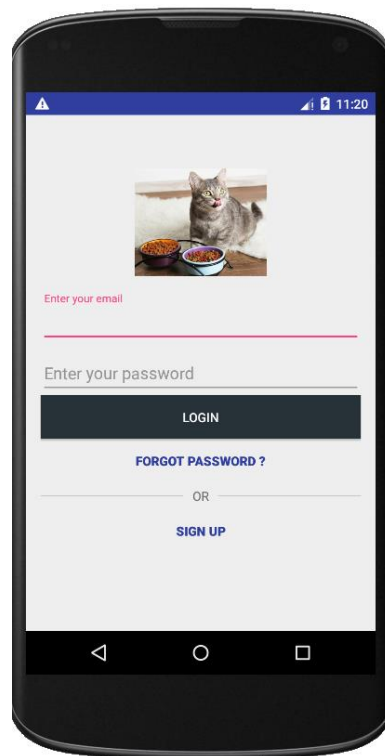


Figure 4.8 First page for Mobile Application.

Homepage is the home menu for user of Mobile Application as shown in Figure 4.9. In this page, there are button for feeding pets. User can choose up to 3 feeding values. The values are standard size for feeding pet. User can follow these guidelines (Animal Wised, 2018):

- a) 2 kg pet, between 30 and 40 grams
- b) 4 kg, between 60 and 80 grams
- c) 6 kg, between 90 and 120 grams

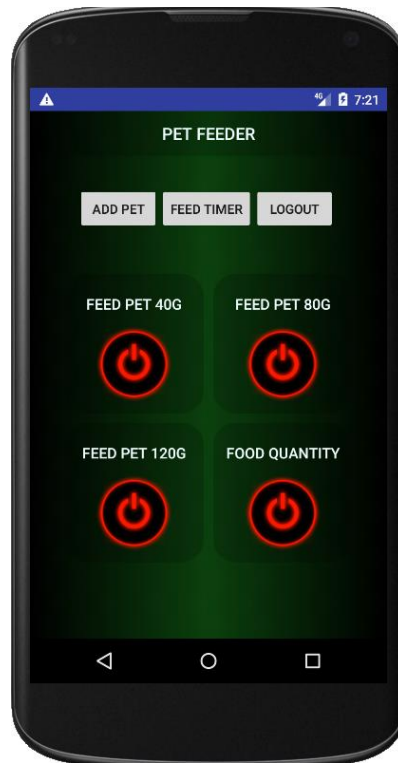


Figure 4.9 Homepage for Mobile Application.

Moreover, if user wanted to know the remaining food in the container, user can just click at the Food Quantity button. As for option, user can add pet info on the button “ADD PET”. The page for Add Pet is as Figure 4.10. This page is optional for the user. User can add their pet info if they wanted. This page is just like a dictionary for the user. User does not have to write down their pet info on the paper, just add it here and it will remain on the database. View pet button will show the information of the pet that user have added to database.

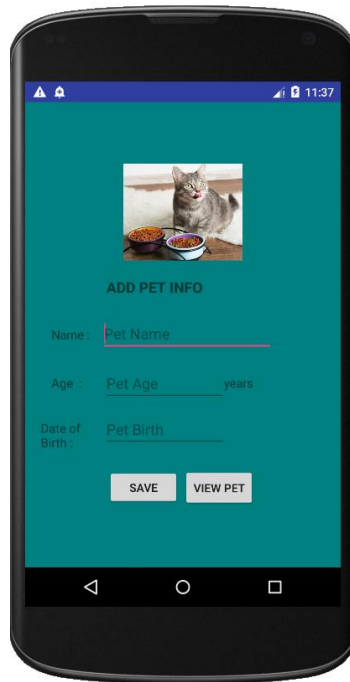


Figure 4.10 Add Pet info page.

Figure 4.11 shows feeding timer for Mobile Application during Mobile Application development. This page will control timer of Mobile Application system, which user can adjust the date, time, and repetition requested as repeat per minute, hour, weeks, or month. User also can delete or edit existing timer.

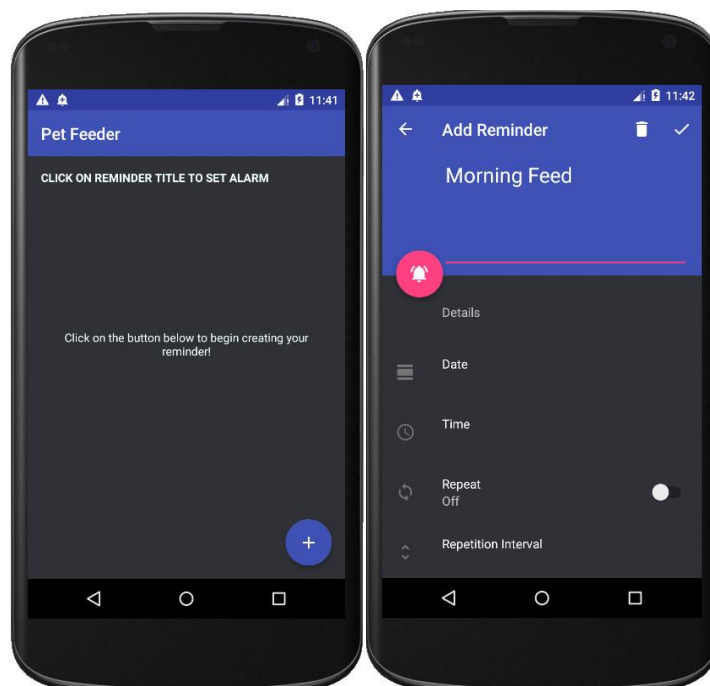


Figure 4.11 Feeding timer for pet.

4.4 CONCLUSION

After all the testing phase of related sensors has been successfully delivered, there is a crucial part of this phase which is final testing phase where it needs to be tested on real environment. It is necessary to train pets as it is part of the main requirement of this project. Therefore, owner must take his/her responsibility to train the pets to get familiar or adapt with the system. There are few challenges for owners in order to train his/her pets to get along with this system such as pet cannot recognize its food because it is in close food container for the purpose to protect the food's quality. It requires owner's intention to check the balance food in container via Mobile Application. It is mandatory for owners to understand pet's behaviour in order for them to use the system effectively.

4.5 SUMMARY

This chapter has discussed about the project development from the scratch until the end including the problem and error occurs while developing the project. Hardware and software needed to make this project succeeds also has been discussed. This chapter also explains in detail on how to operate each hardware and software involved.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 INTRODUCTION

This project has developed to grant pet owners' need by allowing them to remotely feed their pet without having to present during the feeding. From Chapter 1 to Chapter 4, all has been discussed including the methodology of the project, project architecture, experience, development and lastly testing phase. This chapter will be discussing about conclusion, limitation and recommendation for the proposed project.

5.2 LIMITATION OF THE PROJECT

The limitations of this project are:

- I. This project only allows dry food to be served to the pets. Wet food is not allowed because it cannot be rotated by servo motor.
- II. For the testing part, it is hard to get a well-trained pet or to train pet to adapt the current project for experiment purpose.

5.3 RECOMMENDATION

In the process of developing this project, objectives have been achieved according to pet owners' need. But somehow it opens another opportunity for other developers out there to improve the current project to make it better and be able to compete with another product in the market. Future enhancement of the project could be implementing in the project such as:

- I. Build a full pet application for pet that involve taking care of pets' food, health and waste
 - a) Add a temperature sensor that will sense pets' current temperature. From that sensor, it can tell whether the pet is healthy or not.
 - b) A product that allows any types of pets' food.
 - c) A product that takes care of pets' waste without any human intervention needed.

- II. Build a pet controller that can cater a large population of pets
 - a) Create a database and a system that stores pets' information especially current health status, food preferences, type and weight.

5.4 CONCLUSION

As a conclusion, this project was successfully developed and had met its aim and objectives for making the prototype of smart pet feeder. The objective of this project is to design and develop a project that allows the owner to serve food to their pets without having to present at the feeding location. Through the development of this prototype, all the objectives have been achieved. Based on Chapter 3, the process of analysis, design, development, testing have provided a lot of data and information that will lead to the success of this project knowing that this project is worth rather than investing on complex and expensive system. From this project development, the developer has gained a whole new knowledge about Arduino NodeMCU technology and Android Studio that are suitable for daily life used. In developing the project, the sensors and its circuit has opened the doors to new opportunities in the future as making an innovative project is currently in demand in the industry.

REFERENCES

1. Balzer, M. (2015). Australian Veterinary Journal, 93(11).
2. Hodgson, K., & Barton, L. (2015). Pets' Impact on Your Patients' Health: Leveraging Benefits and Mitigating Risk, 28(4), 526–534.
3. Increase, S., Data, O. R., Prevention, P. O., Shortens, O., Just, L., Conditions, M. C. O. Additionally, S. Pet Obesity Epidemic Fact Sheet Financial Costs of Pet Obesity (2015).
4. Russo, N., Vergnano, D., Bergero, D., & Prola, L. (2017). veterinary sciences Small Pilot Survey on Parents' Perception of the Relationship between Children and Pets, 1–5.
5. Jempet. (2018, March 14). JemPet SmartFeeder, Automatic Pet Feeder controlled With Your Iphone,Andriod or other smart devices. Retrieved from <https://jem.pet/products/gempet-smartfeeder-automatic-pet-feeder-controlled-with-your-iphone-andriod-or-other-smart-devices>
6. Noor, M. Z. H., Hussian, A. K., Saaid, M. F., Ali, M. S. A. M., & Zolkapli, M. (2012). The design and development of automatic fish feeder system using PIC microcontroller. Proceedings - 2012 IEEE Control and System Graduate Research Colloquium, ICSGRC 2012, (Icsgrc), 343–347.
7. Vania, Karyono, K., & Nugroho, I. H. T. (2017). Smart dog feeder design using wireless communication, MQTT and Android client. Proceeding - 2016 International Conference on Computer, Control, Informatics and Its Applications: Recent Progress in Computer, Control, and Informatics for Data Science, IC3INA 2016, 191–196.
8. Kramer, M. (2018). Lifecycle: An Analyses Based On The Waterfall Model, 9(1), 77–84.
9. Trivedi, P., & Sharma, A. (2013). A Comparative Study between Iterative Waterfall and Incremental Software Development Life Cycle Model for Optimizing the Resources Using Computer Simulation.
10. Tony. (2018, April 14). Educational flaws: Programming with the Waterfall Model. Retrieved from <http://compsci.ca/blog/educational-flaws-programming-with-the-waterfall-model/>

11. Kumar, D. (2015). Advantages and Disadvantages of Waterfall Model. Retrieved from <http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/>
12. IoT LoLin NodeMCU ESP8266 Wifi Controller Board ESP-12: Best Price in Malaysia. (n.d.). Retrieved from <https://www.lelong.com.my/iot-lolin-nodemcu-esp8266-wifi-controller-board-esp-12-arelectronics-F528319-2007-01-Sale-I.htm>
13. NODEMCU–Getting Started with Arduino Core 1. GPIO pins & Network Access. (2017, October 21). Retrieved from <https://alselectro.wordpress.com/2017/10/21/nodemcu-getting-started-with-arduino-core-1-gpio-pins-network-access>
14. Components101 (2018, April 17). Servo Motor SG-90. Retrieved from <https://components101.com/servo-motor-basics-pinout-datasheet>
15. RPI Labs (2018, April 17). How is a Servo Controlled? Retrieved from <http://rpi.science.uoit.ca/lab/servo/>
16. 2018 Holder 6v Battery. (n.d.). Retrieved from <https://www.dhgate.com/product/battery-storage-case-box-holder-4-aa-1-5v/401926616.html>
17. How Much Food Should I Feed My Cat? (n.d.). Retrieved from <https://www.animalwised.com/how-much-food-should-i-feed-my-cat-2016.html>

APPENDIX A GANTT CHART

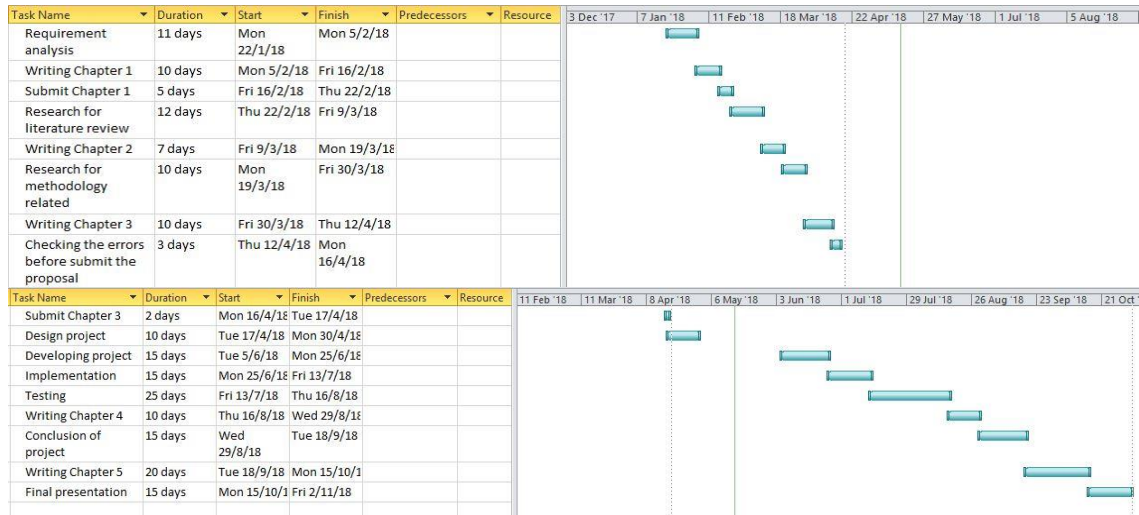


Figure 1 Gantt Chart.

APPENDIX B CODING

This coding is in MainActivity.java located on Android Studio.

```
package dev.edmt.firebaseauthentication;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RelativeLayout;
import android.widget.TextView;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    Button btnLogin;
    EditText input_email,input_password;
    TextView btnSignup,btnForgotPass;

    RelativeLayout activity_main;

    private FirebaseAuth auth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        //View
        btnLogin = (Button)findViewById(R.id.login_btn_login);
        input_email = (EditText)findViewById(R.id.login_email);
        input_password = (EditText)findViewById(R.id.login_password);
        btnSignup = (TextView)findViewById(R.id.login_btn_signup);
        btnForgotPass = (TextView)findViewById(R.id.login_btn_forgot_password);
        activity_main = (RelativeLayout)findViewById(R.id.activity_main);

        btnSignup.setOnClickListener(this);
        btnForgotPass.setOnClickListener(this);
        btnLogin.setOnClickListener(this);

        //Init Firebase Auth
        auth = FirebaseAuth.getInstance();
```

```

//Check already session , if ok-> DashBoard
if(auth.getCurrentUser() != null)
    startActivity(new Intent(MainActivity.this,HomePage.class));
}

@Override
public void onClick(View view) {
    if(view.getId() == R.id.login_btn_forgot_password)
    {
        startActivity(new Intent(MainActivity.this,ForgotPassword.class));
        finish();
    }
    else if(view.getId() == R.id.login_btn_signup)
    {
        startActivity(new Intent(MainActivity.this,SignUp.class));
        finish();
    }
    else if(view.getId() == R.id.login_btn_login)
    {
        loginUser(input_email.getText().toString(),input_password.getText().toString());
    }
}

private void loginUser(String email, final String password) {
    auth.signInWithEmailAndPassword(email,password)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if(!task.isSuccessful())
                {
                    if(password.length() < 6)
                    {
                        Snackbar snackBar = Snackbar.make(activity_main,"Password length must
be over 6",Snackbar.LENGTH_SHORT);
                        snackBar.show();
                    }
                }
                else{
                    startActivity(new Intent(MainActivity.this,HomePage.class));
                }
            }
        });
}
}
}

```

This coding is in AlarmMain.Java located on Android Studio.

```
package dev.edmt.firebaseauthentication;
```

```
import android.app.ProgressDialog;
import android.content.ContentUris;
import android.content.ContentValues;
```

```

import android.content.DialogInterface;
import android.content.Intent;

import android.database.Cursor;
import android.net.Uri;
import android.support.design.widget.FloatingActionButton;
import android.support.v4.app.LoaderManager;
import android.support.v4.content.CursorLoader;
import android.support.v4.content.Loader;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.Toolbar;
import android.text.InputType;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class AlarmMain extends AppCompatActivity implements
LoaderManager.LoaderCallbacks<Cursor> {

    private FloatingActionButton mAddReminderButton;
    private Toolbar mToolbar;
    AlarmCursorAdapter mCursorAdapter;
    AlarmReminderDbHelper alarmReminderDbHelper = new AlarmReminderDbHelper(this);
    ListView reminderListView;
    ProgressDialog prgDialog;
    TextView reminderText;

    private String alarmTitle = "";

    private static final int VEHICLE_LOADER = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_alarm_main);

        mToolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(mToolbar);
        mToolbar.setTitle(R.string.app_name);

        reminderListView = (ListView) findViewById(R.id.list);
        reminderText = (TextView) findViewById(R.id.reminderText);

        View emptyView = findViewById(R.id.empty_view);
        reminderListView.setEmptyView(emptyView);

```

```

mCursorAdapter = new AlarmCursorAdapter(this, null);
reminderListView.setAdapter(mCursorAdapter);

reminderListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int position, long
id) {

        Intent intent = new Intent(AlarmMain.this, AddReminderActivity.class);

        Uri currentVehicleUri =
ContentUris.withAppendedId(AlarmReminderContract.AlarmReminderEntry.CONTENT_URI,
id);

        // Set the URI on the data field of the intent
        intent.setData(currentVehicleUri);

        startActivity(intent);

    }
});

mAddReminderButton = (FloatingActionButton) findViewById(R.id.fab);

mAddReminderButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //Intent intent = new Intent(v.getContext(), AddReminderActivity.class);
        //startActivity(intent);
        addReminderTitle();
    }
});

getSupportFragmentManager().initLoader(VEHICLE_LOADER, null, this);

}

@Override
public Loader<Cursor> onCreateLoader(int i, Bundle bundle) {
    String[] projection = {
        AlarmReminderContract.AlarmReminderEntry._ID,
        AlarmReminderContract.AlarmReminderEntry.KEY_TITLE,
        AlarmReminderContract.AlarmReminderEntry.KEY_DATE,
        AlarmReminderContract.AlarmReminderEntry.KEY_TIME,
        AlarmReminderContract.AlarmReminderEntry.KEY_REPEAT,
        AlarmReminderContract.AlarmReminderEntry.KEY_REPEAT_NO,
        AlarmReminderContract.AlarmReminderEntry.KEY_REPEAT_TYPE,
        AlarmReminderContract.AlarmReminderEntry.KEY_ACTIVE
    };

};

return new CursorLoader(this, // Parent activity context

```



```

        AlarmReminderContract.AlarmReminderEntry.CONTENT_URI, // Provider content
URI to query
        projection,      // Columns to include in the resulting Cursor
        null,            // No selection clause
        null,            // No selection arguments
        null);          // Default sort order

    }

    @Override
    public void onLoadFinished(Loader<Cursor> loader, Cursor cursor) {
        mCursorAdapter.swapCursor(cursor);
        if (cursor.getCount() > 0){
            reminderText.setVisibility(View.VISIBLE);
        }else{
            reminderText.setVisibility(View.VISIBLE);
        }
    }

    @Override
    public void onLoaderReset(Loader<Cursor> loader) {
        mCursorAdapter.swapCursor(null);
    }

    public void addReminderTitle(){

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Set Reminder Title");

        final EditText input = new EditText(this);
        input.setInputType(InputType.TYPE_CLASS_TEXT);
        builder.setView(input);

        builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                if (input.getText().toString().isEmpty()){
                    return;
                }

                alarmTitle = input.getText().toString();
                ContentValues values = new ContentValues();

                values.put(AlarmReminderContract.AlarmReminderEntry.KEY_TITLE, alarmTitle);

                Uri newUri =
getContentResolver().insert(AlarmReminderContract.AlarmReminderEntry.CONTENT_URI,
values);

                restartLoader();

                if (newUri == null) {

```

```

        Toast.makeText(getApplicationContext(), "Setting Reminder Title failed",
Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), "Title set successfully",
Toast.LENGTH_SHORT).show();
        }

    }
});
builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});

builder.show();
}

public void restartLoader(){
    getSupportLoaderManager().restartLoader(VEHICLE_LOADER, null, this);
}
}
}

```

This coding is in SignUp.Java located on Android Studio.

```

package dev.edmt.firebaseauthentication;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RelativeLayout;
import android.widget.TextView;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class SignUp extends AppCompatActivity implements View.OnClickListener {

    Button btnSignup;
    TextView btnLogin,btnForgotPass;

```

```

EditText input_email,input_pass;
RelativeLayout activity_sign_up;

private FirebaseAuth auth;
Snackbar snackbar;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_sign_up);

    //View
    btnSignup = (Button)findViewById(R.id.signup_btn_register);
    btnLogin = (TextView)findViewById(R.id.signup_btn_login);
    btnForgotPass = (TextView)findViewById(R.id.signup_btn_forgot_pass);
    input_email = (EditText)findViewById(R.id.signup_email);
    input_pass = (EditText)findViewById(R.id.signup_password);
    activity_sign_up = (RelativeLayout)findViewById(R.id.activity_sign_up);

    btnSignup.setOnClickListener(this);
    btnLogin.setOnClickListener(this);
    btnForgotPass.setOnClickListener(this);

    //Init Firebase
    auth = FirebaseAuth.getInstance();
}

@Override
public void onClick(View view) {
    if(view.getId() == R.id.signup_btn_login){
        startActivity(new Intent(SignUp.this,MainActivity.class));
        finish();
    }
    else if(view.getId() == R.id.signup_btn_forgot_pass){
        startActivity(new Intent(SignUp.this,ForgotPassword.class));
        finish();
    }
    else if(view.getId() == R.id.signup_btn_register){
        signUpUser(input_email.getText().toString(),input_pass.getText().toString());
    }
}

private void signUpUser(String email, String password) {
    auth.createUserWithEmailAndPassword(email,password)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if(!task.isSuccessful())

```

```
        {
            snackbar = Snackbar.make(activity_sign_up,"Error:
"+task.getException(),Snackbar.LENGTH_SHORT);
            snackbar.show();
        }
        else{
            snackbar = Snackbar.make(activity_sign_up,"Register success!
",Snackbar.LENGTH_SHORT);
            snackbar.show();
        }
    }
});

}

}
```

APPENDIX C
PROTOTYPE PET FEEDER



Figure 1 Pet Feeder with Pet's food.



Figure 2 Pet's food is going through to its container.



Figure 3 Pet Feeder with pet's food in its container.