AUTOMATIC PET FEEDER USING INTERNET OF THINGS (IoT) TECHNOLOGY

NOR ALIA MAISARAH BINTI HAMZAN

BACHELOR OF COMPUTER SCIENCE
(COMPUTER SYSTEM & NETWORKING)
WITH HONOURS

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : NOR ALIA MAISARAH BINTI HAMZAN

Date of Birth

Title : AUTOMATIC PET FEEDER USING INTERNET OF THINGS (IOT) TECHNOLOGY

Academic Session : 2022 / 2023

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: 09/02/2023

DR. NOR SYAHIDATUL NADIAH
Name of Supervisor
Date: 09/02/2023

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
Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

____
(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/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the bachelor of Computer Science (Computer System & Networking) with Honours

____
(Supervisor’s Signature)

Full Name : DR. NOR SYAHIDATUL NADIAH BINTI ISMAIL

Position : SENIOR LECTURER

Date : 09/02/2023

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 : NOR ALIA MAISARAH BINTI HAMZAN

ID Number : CA19048

Date :09/02/2023

AUTOMATIC PET FEEDER USING
INTERNET OF THINGS (IoT)
TECHNOLOGY

NOR ALIA MAISARAH BINTI HAMZAN

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

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2023

ACKNOWLEDGEMENTS

In the name of the most kind and merciful Allah. First and foremost, I want to express my gratitude to Almighty Allah for providing me with the potential to pursue this degree and fully complete it.

Second, I want to specifically appreciation goes to my supervisor, TS Dr Nor Syahidatul Nadiah Binti Ismail, for her advice and encouragement in supporting me to finish my project. Every idea generation is guided in the right path by supervision and suggestions. It also gives me the opportunity to enhance my experience and understanding in preparation for my final year project.

I would also like to give special thanks to my sister, Nor Aimiesya Maisarah, for always supporting and accompanying me while finishing the project. Next, I want to thank myentire, cherished family for their unwavering support and understanding as I performed my project and wrote my thesis. Your supplication for me has kept me going so far.

Not to be forgotten are my close friends, who constantly provide words of inspiration and strive just as hard to pass our final year projects. Being surrounded by kind people like you all makes me feel very grateful.

Finally, I would like to express my gratitude to Dr. Danakorn, who coordinated the final year project and helped me by providing advice and motivating me to finish the project. Furthermore, the supervision in the writing process allows me to comprehend the correct format in writing the thesis.

ABSTRAK

Pada masa kini, orang zaman ini telah membangunkan gaya hidup memelihara haiwan peliharaan. Ini disebabkan oleh fakta bahawa memelihara haiwan peliharaan boleh menghilangkan ketegangan dan kebosanan di rumah, terutamanya bagi individu yang bersendirian di rumah. Namun, manusia pada masa kini disibukkan dengan kesibukan, pekerjaan dan selalu melancong pada waktu bercuti. Akibatnya, pemilik bimbang tentang haiwan peliharaan di rumah. Dalam projek ini, “Automatic Pet Feeder” menggunakan teknologi Internet of Things (IoT) dengan aplikasi mudah alih dibangunkan untuk membantu pemilik haiwan peliharaan. Kaedah “Rapid Application Development” (RAD) telah ditentukan sebagai teknik terbaik yang sesuai untuk projek yang dicadangkan ini. Berbanding dengan model pembangunan perisian lain, model RAD adalah metodologi yang paling sesuai untuk merancang, menstruktur dan membangunkan sistem ini. Ini kerana model RAD boleh mempercepatkan pembangunan sistem “Automatic Pet Feeder”. “Automatic Pet Feeder” akan disambungkan dengan sambungan WIFI di rumah. Dengan menggunakan sistem ini, pemilik haiwan peliharaan boleh memantau haiwan peliharaan mereka menggunakan aplikasi mudah alih. Melalui aplikasi mudah alih, pemilik boleh memantau keadaan haiwan peliharaan mereka dalam masa sebenar dengan menggunakan kamera yang dipasang pada peranti perkakasan, boleh menjadualkan masa pemakanan berdasarkan berat yang lebih baik dan sesuai. Terdapat juga bunyi yang dilekatkan untuk menarik haiwan peliharaan mendekati peranti. Projek ini boleh membantu ramai pemilik haiwan peliharaan untuk menjaga haiwan kesayangan mereka semasa mereka berada jauh dari rumah. Selain itu, haiwan peliharaan mereka akan mendapat makanan yang mencukupi dan tidak berlebihan yang akan memudaratkan kesihatan haiwan peliharaan mereka. Sistem ini juga boleh digunakan untuk memberi makan kepada haiwan peliharaan lain seperti kucing, anjing, dan arnab, serta ikan di dalam akuarium. Akibatnya, matlamat utama projek ini adalah untuk memastikan haiwan kesayangan yang comel menerima makanan mereka tepat pada masanya, membolehkan pemilik haiwan kesayangan menumpukan perhatian pada kerja mereka tanpa berasa bimbang tentang keperluan nutrien haiwan kesayangan mereka.

ABSTRACT

Nowadays, people of this era have developed a lifestyle of keeping pets. This is due to the fact that keeping pets may relieve tension and boredom at home, especially for individuals who are alone at home. However, people nowadays are preoccupied with their busy schedules, employment and always travel on vacations. As a result, the owner worries about the pet at home. In this project, the automatic pet feeder using Internet of Things (IoT) technology with mobile application is developed to help the pet owner. The rapid application development (RAD) method was determined to be the best appropriate technique for this suggested project. In comparison to other software development models, the RAD model is the most appropriate methodology for planning, structuring, and developing this system. This is because the RAD model can speed up the development of an automatic pet feeder system. The automatic pet feeder will have connected with the WIFI connection at home. By using this system, the owner of the pet can monitor their pet using the mobile application. Through the mobile application, the owner can check on their pet's situation in the actual time by using camera that attach in the hardware device, can schedule the feeding time based on preferable weight. There is also a sound that attached to attract the pet come near to the devices. This project can help a lots of pet owner to taking care of their pet while they are away from home. Other than that, their pets will get enough food and not excess which will be harmful to their pets' health. This system also can be used to feed others pets such as cats, dogs, and rabbits, as well as fish inside the aquarium. As a result, the primary goal of this project is to ensure that the adorable pet receives their food on time, allowing the pet owner to focus on their work without feeling worried about their pet’s nutrient requirements.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IoT	Internet of Things
RFID	Radio-Frequency Identification
MQTT	MQ Telemetry Transport
SDLC	Software Development Life Cycle
RAD	Rapid Application Development
ERD	Entity Relationship Diagram
IDE	Integrated Development Environment

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Nowadays, people of this era have developed a lifestyle of keeping pets. This is due to the fact that keeping pets may relieve tension and boredom at home, especially for individuals who are alone at home. In order to determine if pet owners differ from non-pet owners in terms of wellbeing, personality type, and attachment style, 217 persons (79 percent women, average age 31, average yearly family income $77,000) participated in this study. Pet owners were consistently joyful, wealthier, and more stable than non-owners despite a number of other disparities between the groups. [1]. People nowadays, on the other hand, are preoccupied with their employment and travel on vacations. As a result, these pet owners usually forget to feed their pets or feed their pets too late. As a result, their pets are neglected.

Automatic pet feeding system, which employs IoT technology, is a system that functions automatically at the appropriate time established by the pet owner. The Internet of Things (IoT) has developed in prominence over the last couple of years. Through specific target and implement additional, connected, intelligent homes, these IoT products enhance and standardize daily lives. [2]. Therefore, pet owners may use this machine to supply food for their pets without having to be around all the time.

A modern automated pet feeder can play the role of manual pet feeding. Some pet feeders have internet connectivity, allowing remote monitoring. The demand for automated pet feeding systems will rise as a result of the writer's suggestions for some added functions that might be included to turn the pet feeder into a Smart Pet Feeder while maintaining the price of production within an affordable range for the user. The Smart Pet Care System and the Automatic Pet Feeder are two studies on automatic pet feeders that have been found [3]. Obesity affects 56% of dogs and 60% of cats, according to a veterinary doctor's survey

[4]. Obesity in pets is caused by a lack of exercise and excessive eating. When a pet owner pours an improper amount of food into the bowl at frequent intervals, it can affect the health of the pet [5]. As a result, a controlled food supply is critical for pet health.

The goal of creating this automatic pet feeder is to ensure that pets can eat on a schedule without going hungry or overeating food, both of which can be dangerous to health. With this concept, pet owners may work, travel, or be away from home without worrying about their pets. They only need to set the time and amount of food that should be supplied to their pets and be able to solve their matters comfortably.

1.2 PROBLEM STATEMENT

In today's world, being proactive in order to make a living is the typical. Some people must work overtime and remain late at work in order to complete their jobs. Too busy with everyday matters attributable to a lack of time management causes some pet owners to forget about things that have no bearing on their life, such as to manage their pets, since they are too fatigued from their daily activities. In consequence of this, people regularly neglect to feed their pets or feed them at inappropriate times.

Aside from the hustle and bustle of work, certain jobs need us to travel to other locations or sometimes when we go on a vacation away from home will cause problems to pet owners. As a result of this circumstance, we must let our pets alone at home without monitoring. However, some pet owners leave their pets in pet hotels to ensure that their pets' requirements are fulfilled. Unfortunately, sending their pets to a pet hotel is an expensive proposition. On the other hand, some people approach their neighbors for help in monitoring their pets at home. In the meantime, this behavior is inappropriate, as not everyone can provide sufficient care for our pets.

Due to the obvious issues mentioned, our pet's feeding schedule will fluctuate. On the other hand, pet food scheduling, is critical since it influences the pets' health. When, pet's food schedule changes frequently, pet’s mood will also change. They will become frustrated or depressed because they do not get enough food, with all this their health will also be affected. As pet owners who love their pets, the health and mental state of pets is very much emphasized. Asa result, providing pets adequate food at the right moment might help them stay in great condition.

Therefore, the automatic pet feeder can alleviate this problem because it is very simple to operate and ideal for all types of pets. Both the pet owner and the pet benefit from this system. Combining IoT and mechanical controls via electrical systems makes it simple to improve the performance of these equipment as efficiently as possible.

1.3 OBJECTIVE

Based on the problem statements, the objectives of the project are:

i) To study the existing Internet of Things (IoT) applications in pet care system which is automatic pet feeder.

ii) To develop an automatic pet feeder system by using Internet of Things (IoT) Technology.

iii) To test the efficiency and functionality of the Internet of Things (IoT) application in pet care system which is automatic pet feeder.

1.4 SCOPE

The project's scope is:

User Scope:

i) The pet owner

System Scope:

i) The automatic feeder will load the feeding tray with the right quantities of food.

ii) The automatic feeder will be monitoring the pets by using security camera.

iii) The database will store the information on the feeding schedule, food’s weight, and list of users.

iv) The mobile application it consists of interface that will display the activity that the user can do.

Development Scope:

i) By using technology of IoT and Visual Studio code to develop a coding byusing HTML, CSS, and TypeScript language.

ii) IONIC Framework (software used to develop mobile application).

iii) Arduino IDE (software used to program code of IoT components).

iv) The weight sensors will detect the food’s weight, it will send the signal to the website.

1.5 SIGNIFICANCE OF PROJECT

i) User / Pet Owner

This technology benefits pet owners since it reduces the need for pet owners to feed their pets, particularly when the owner has business outside the house or is going on a short trip. Because the project includes a camera that allows owners to monitor their pets while they are away, owners no longer need to be worried about their pets' well-being while they are away. Furthermore, the concept could save pet owners money by eliminating the need to send their pets to pet hotels for monitoring while they are away.

ii) Pets

The benefit of this idea for pets is that they may eat at precisely the proper moment.They also do not have to wait for the owner to feed them, which is very useful when their owner is not at home or when their owner forgets to feed them. Furthermore, since their owners can provide appropriate food without overspending, this idea can help prevent pets from becoming overweight.

1.6 REPORT ORGANIZATION

There are three chapters inside this report. The first chapter covers the project's overall view, including the Introduction, Problem Statements, Project Objective, Scope, and Report Organization.

Chapter 2 briefly explain about the literature review on existing systems of Internet of Things (IoT) applications in pet care system which is pet automatic feeder.

Chapter 3 explains about the methodology, which will specifics the approaches and processes to be used in this project.

The stages of development, including the testing and implementation phases, are covered in chapter 4.

Lastly, chapter 5 will conclude with a discussion of the project and its implications for the future.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

A research study of a few references on a pet feeding system will be presented in depth in this chapter, which is a literature review. This chapter is divided into five main sections that cover the concepts of Internet of Things (IoT), as well as a review, comparison, and summary of existing research on IoT-enabled smart pet feeders. This chapter will be focusing on providing an overview of numerous pets feeding technology systems and how to utilize it.

2.2 INTERNET OF THINGS (IoT)

The Internet of Things (IoT), also referred to as the internet of Everything and then the Smart Manufacturing, is an innovative technical concept that imagines a global network of interconnected devices and equipment [2]. Many businesses are interested in the Internet of Things since it is often regarded as one of the greatest significant areas of potential advancement [6]**.** The internet of things could be used by people to maintain intelligent living and give them complete self - determination. The Internet of Things is essential to business and offers smart devices to automate homes. With data on anything from machine performance to supply chain and logistical operations, the Internet of Things gives organizations a real-time insight of how their systems operate.

IoT devices can be wirelessly connected to accomplish their goals. Following that, the data can be shared all over devices via connectivity that utilize industry-standard communications protocols [3]. The smart connected devices, also known as "things," come in a variety of sizes, from small wearables to massive machines, and all include sensory devices. [7]. However, in order for the Internet of Things concept to become a reality, the virtualization technology will have to develop beyond standard mobile computing scenarios including smart phones and portables, to include linking regular things and embedded intelligence throughout our surroundings [8].

2.3 REVIEW OF EXISTING SYSTEM

2.3.1 NuriPet: A Smart Pet Feeding Machine for SNS

The condition of the pets at home is absolutely important to hectic hardworking pet owners since they have left to leave their pets alone at home without anyone to monitor them. As a result, NuriPet: A Smart Pet Feeding Machine for SNS was designed as an equipment and a software for online networking service communication involving pet owners, including their pets [9]. The essential conception of the NuriPet machine and software is depicted in Figure 1.

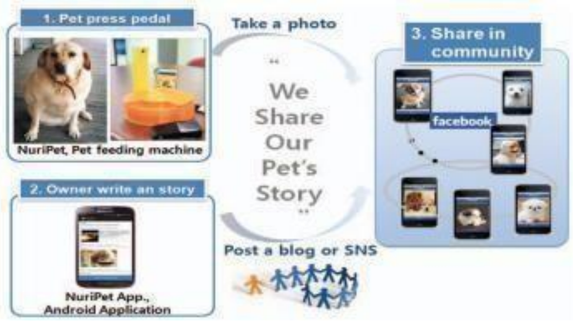


Figure 1 Concept Diagram

The Raspberry Pi is used to handle a camera, servomotor, pedal, and connection to the internet on the NuriPet machine. Designers used SketchUp to build the pet feeder, which designers then printed by using a 3d printing machine. Pets may be using the feeding device to open a tray for food and snap a photo to upload to a social media platform. The pedal quickly transmits an information to the control board. The control board, on the other hand, only handles one signal. To avoid misuse, the other signals are turned off for a few minutes. Users can essentially control it whenever and wherever they need. Despite this, designers created two situations for users. User's controls are available via the NuriPet app, while pet controls are available via the pedal.

Finally, the control board is linked to Facebook through the internet. Designers also created an Android app that allows users to manage the equipment from afar. Pet owners may use the application to control the equipment from anywhere at any time. Pet owners may also go through their pets' images and create stories to share with their friends.

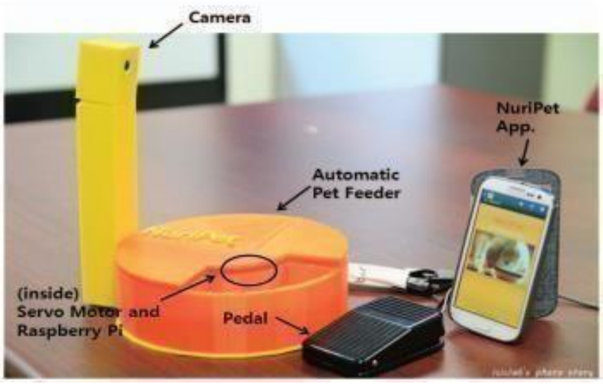
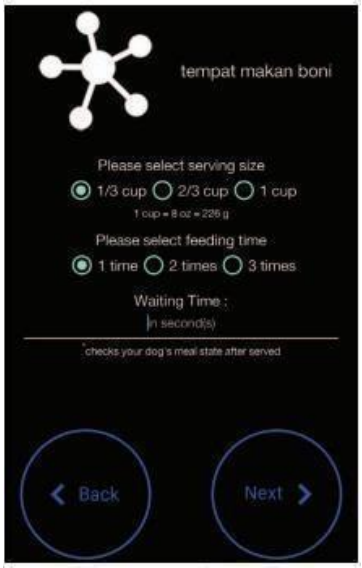


Figure 2 A Smart Pet Feeding Machine

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

One of the issues with dog management is ensuring that the dog is fed on regular basis. Because of busy jobs, owners frequently neglect to feed their dogs. The Smart Dog Feeder is the resolution to these difficulties [10]. Consistent feeding can be provided by this technique without interfering with the owner's activities. Owners might not have to worried about left their dogs alone at home, and they can keep track of their meals on their smartphones. The Figure 3 below shows the Fed Page for the mobile application.



tempat makan boni

Please select serving size

● 1/3 cup ○ 2/3 cup ○ 1 cup

1 cup = 8 oz = 226 g

Please select feeding time

● 1 time ○ 2 times ○ 3 times

Waiting Time :

in second(s)

*checks your dog's meal state after served

Figure 3 Fed Page

The entire system was examined till the primary standard was reached throughout the time trial, such as the genuine clock alarm, RFID reader, motor controller, and load cell monitoring. The outcomes showed that the servo motor and real-time clock could perform as anticipated. The authentication method is based on RFID technology. Only a dog with an RFID tag on its neck can make the Smart Dog Feeder operate however there was a reading distance restriction. Although the description stated that the highest range was 3 cm, the RFID reader used in this study can only really recognize a greater range of 2.5 cm without error.

Automatic feeding may be accomplished by using Wi-Fi connection to link the Android mobile application to the servers and the Smart Dog Feeder, and then exchanging messages using the MQTT protocol [10]. In order to ensure that interaction among the Mobile application, host, and Smart Dog Feeder cannot be viewed by outside parties, secure connections are provided via the TLS/SSL protocol.

A strain gage assessment test is performed throughout the feeding process to guarantee that the information may be used. The test demonstrated the limitations of load cell measurements. However, a comparison test may reveal the difference among the load cell and a reliable precise scale, enabling the load cell's data to be adjusted to an almost exact result. The amount of pet food delivered throughout the feeding process typically exceeds the number of consumers due to a mechanical restriction on the amount of animal feeds the propeller blade may discharge every turn (20 gram). Additionally, there's a possibility that food will get stuck between the blades, rendering the feeding process ineffective. Furthermore, the servo may occasionally come to a halt when the blade is partly opened. Figure 4 below show the Smart Dog Feeder Machine for this project.

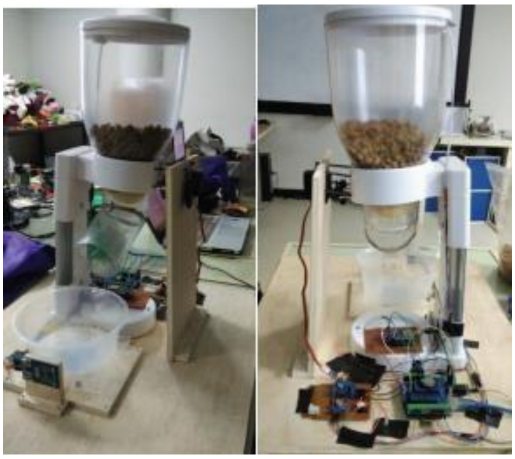


Figure 4 Smart Dog Feeder

2.3.3 Pet Food Auto Feeder by Using Arduino

Humans have a special bond with their pets. It relieves stress from a stressful life while also setting a dangerous precedent for thieves. Pets are now considered members of the family. As a result, the pet's health and wellbeing must be taken into consideration, as well as a nutritious diet. Different amounts of food and nutrients will be required depending on the pet's size [11].

To combat obesity and starvation, a concept for an automatic pet feeder was created, bringing ease to pet owners. Pet food dispensing systems can be divided into two categories: straight movement and rotatable to linear motion. Animal kibble is released outside by the spinning motion of the mechanism, which is positioned linearly beneath the tank. According to studies, the benefits of employing rotational to liner motion include preventing pet food from being stuck in the hopper.

The finalized conceptual framework and in-depth illustrations of every element are produced using Solid Works. Using a finite element model, the pet feeder's operation is investigated. The mechanism is subjected to a 40N force, and total displacement analysis, maximum bending deformation, minimum principal stress, maximum principal stress, and maximum shear stress are all performed. To ensure that the feeding system is operating without any disruptions combining the animal food technique and the prototype's allocated Arduino software, the mechanism is built into the prototype and put through trial evaluation. Figure 5 depicts the pet feeder Arduino system.

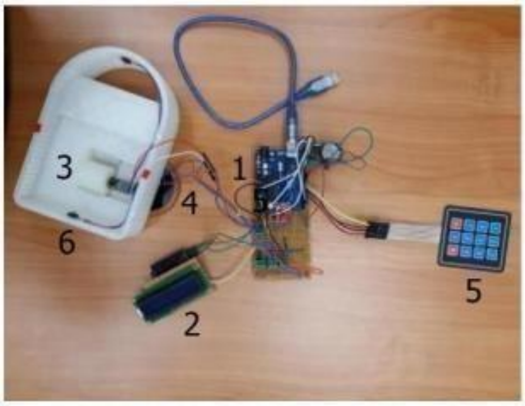


Figure 5 Automated Pet Feeder using Arduino

The parts consist of a real-time clock (1), a liquid crystal display (2), a dc motor (3), a buzzer (4), a 4x3 keypad (5), and a limit switch (6). The microcontroller is an Arduino, which has the ability to change the number of pieces and feeding time. The developed invention is an automatic pet feeder that may be modified depending on three key factors. By hitting "*," the user can change the true time so that it no longer depends on the time they first use it. Second, by pressing "1" and "2," respectively, the user can enter the first and second feeding timings. By pressing "3," the user can finally change the feed size up to 9 periods. The measurement cup-equipped rectangular rack moved one period in both directions. The rectangular rack is noted for moving forward from its initial position, which contained a measuring cup filled with pet food. The rectangular rack is noted for moving backward from its initial position, which was the measuring cup loaded with pet food. Figure 6 depicts the Pet Food Auto Feeder design.

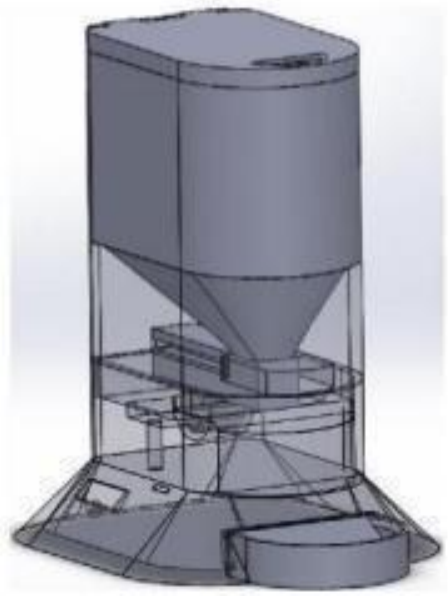


Figure 6 Pet Food Auto Feeder

2.4 COMPARISON OF EXISTING SYSTEM

This section's goal is to demonstrate three existing automatic pet feeders that include internet of things (IoT) into their systems. The NuriPet: A Smart Pet Feeding Machine for SNS, the Smart Dog Feeder Design Using Wireless Communication, MQTT and Android Client, and the Pet food auto feeder using Arduino are just a few of the existing systems that have been found. This is displayed in Table 1, which demonstrates the outcomes of the similar evaluation for the preferred system.

Table 1 Comparison of Existing System

System	NuriPet: A Smart Pet Feeding Machine for SNS	Smart Dog Feeder Design Using Wireless Communication	Pet food auto feeder using Arduino	Automated Pet Feeder using IoT (Proposed work)
Hardware	• Respberry Pi • Camera • Pedal • 360 degrees turn Servomotor	• Arduino UNO microcontroller • NodeMCU ESP8266 Wifi Model • RC522 RFID reader and tag • Servo Motor DF15RSMG • Real time clock DS3231 • Magnetic switch LM2596	• Real Time Clock • Liquid Crystal Display • DC Motor • Buzzer • 4x3 Keypad • Limit switch • Arduino UNO microcontroller	• NodeMCU ESP32Board • WT588D Voice Sound Audio Player Module • ESP32-Cam • Torque Metal Gear DC Motor • HX711 Load Cell Straight Bar • ISD1820 Sound Recording Module

Sensor	- Proximity Sensor	- Load Cell Sensor - Weight Sensor	- No Sensor used	- Weight Sensor
Features	- Pets may use the pedal to open the tray. - Pets also can use the pedal to snap a picture. - Pet’s owner can control the tray by using the mobile application.	- RFID technology is used in this machine. As a result, the Smart Dog Feeder can only be operated by a dog wearing an RFID tag around its neck. - To make the mobile application works, user must register on Sign Up Page. - User can change Profile and Password at Account Setting Menu. - At the Device Menu, the user can select the portion size, the number of times they want to feed their dog,	- By pressing '*,' the user can change the actual time depends on time since they first use it. - By pressing '1' and '2', the user can set either the first or second feeding times, respectively. - The user may modify the stream size up to 9 periods by pressing '3'. The rectangular rack with the measurement cup went forward and backward by one period.	- A portion size for the pet's food can be specified by the user. - To feed their pets, the user can set a timer. - User can monitor their pets on the mobile application by using the security camera which has been implemented on the machine. - The user's voice can be heard by their pets through speakers installed on

		and the feeding status.		the machine.
Mobile application development	Yes	Yes	No	Yes
Internet Dependency	Yes	Yes	No	Yes
Usage	All Pet	Only Dog	All Pet	All Pet
Advantages	• Have a mobile application that allows pet owners to regulate the feeding functionality of their pets virtually, through the NuriPet app for Android. • The pet feeding machine's structural stability is adequate, and it is	• The Arduino is permitted time and a feeding schedule is configured using a real-time clock. • Have a mobile application that allows pet owners to regulate the feeding functionality of their pets virtually.	• The Arduino is permitted time and a feeding schedule is configured using a real-time clock. • The pet feeding machine's structural stability is adequate, and it is unlikely to	• The Arduino is permitted time and a feeding schedule is configured using a real-time clock. • Have a mobile application that allows pet owners to regulate the feeding

	unlikely to tumble or flip.	• Developers use bigger dispenser to keep the dog food. The container for dog food also quite big.	tumble or flip. • Developers use bigger dispenser to keep the pet food.	functionality of their pets virtually. • The pet feeding machine's structural stability is adequate, and it is unlikely to tumble or flip.
Disadvantages	• Required internet connection. • The application can used by the android user only. • A pet feeding machine's architecture is fragile, making it readily damaged by pets. Container for pet food is too small and inconvenient. It can be use by one pet only. It also does not have a dispenser to keep pet food.	• Required internet connection. • The application can used by the android user only. • A pet feeding machine's architecture is fragile, making it readilydamaged bypets. • Using RFID technology will be as problematic as having a reading distance limitation, aside from the fact that this pet feeder	• No mobile application will make it impossible for users to keep track of their pets and feed their pets from afar. • Container for pet food is too small and inconvenient. It can be use byone pet only.	• Required internet connection. • It can be use by one pet only at one time. • The application can used by the android user only.

		will only operate if the dog wears an RFID chain around their neck.		

2.5 SUMMARY OF REVIEW OF EXISTING SYSTEM

There are benefits and limitations in each of the three existing systems, which are NuriPet: A Smart Pet Feeding Machine for SNS, Smart Dog Feeder Design Using Wireless Communication, MQTT and Android Client, and Pet food auto feeder byusing Arduino.

NuriPet: A Smart Pet Feeding Machine for SNS is an automated pet feeder with IoT and a mobile application [9]. The benefits of this system include a mobile application that enables pet owners to remotely control the feeding functions of their pets using the NuriPet app for Android. Furthermore, the pet feeding machine's structural stability is sufficient, and it is unlikely to fall or flip. This is to ensure that the machine is not readily destroyed if it is agitated by pets.

Next, Smart Dog Distributor Design MQTT and Android Client implement real clocks according to the set time and meal pattern while using wireless communication [10]. A real-time clock's main function is to accurately display the time and date for a variety of applications. This ensures that the time specified on the mobile application and the time on the feeding machine are similar. This project also includes a mobile application for the user's convenience. In addition, developers keep dog food in bigger dispensers. Dog food containers are also fairly huge. If the pet is left for an extended period of time, this will guarantee that the food kept in the container is sufficient.

Furthermore, real clocks are implemented in the Pet food auto feeder using Arduino exactly to the given time and meal pattern [11]. Additionally, the physical structure of the pet feeding machine is stable and strong, and it is unlikely to tumble or flip. The design's stability and robustness are essential so that the food in the dispenser does not spill or disperse. Additionally, to ensure ample food, developers store dog food in bigger dispensers.

However, every benefit must have a negative counterpart. As a result, each existing system contains drawbacks in addition to benefits. For example, the mobile application for NuriPet: A Smart Pet Feeding Machine for SNS is required an internet connection and applicable for android user only [9]. As a result, if the user does not have access to the internet or does not have an Android phone, the mobile application will not function. In addition, this machine has a fragile pet feeding machine architecture that is easily broken by pets. As a result, if interfered with by pets, such as being bitten or clawed, this feeding machine will not withstand long and will be damaged quickly. Furthermore, the pet food container is excessively tiny and inconvenient. It can only be used by one pet at a time. It also lacks a dispenser for storing pet food. As a result, the amount of food that can be placed in the machine is limited, and it will quickly get winded due to the lack of food storage.

Smart Dog Feeder Design Using Wireless Communication, MQTT, and Android Client is the next existing system [10]. This product also required an internet connection and applicable for android user only. As a result, if the user does not have access to the internet or does not have an Android phone, the mobile application will not function. This product's another drawback is that it has a delicate pet feeding machine architecture that is easily destroyed by pets. Aside from the fact that this pet feeder will only work if the dog wearing an RFID chain around their neck, this idea uses RFID technology. However, implementing RFID technology will be as difficult as having a reading distance constraint. This product will not operate if the dog does not have an RFID chip or if the RFID chip is gone. Users will find it challenging to guarantee that their dogs always wear the RFID chip because of this.

Finally, because this project does not employ mobile applications, there are certain drawbacks to using Pet food auto feeder using Arduino [11]. As a result of this, users are unable to manage the feeding machine remotely, such as from work or while on vacation. In actuality, the user must manually programme the machine's instructions. Because the purpose of this project is to make it easier for consumers to monitor and feed their animals remotely when they are left alone at home, things like these will be drawbacks to the project. Additionally, pet food tray is excessively small and impractical. It can only be utilized by one pet at a time. Therefore, the amount of food that may be placed at one time is limited.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

The methods used for the project, as well as the entire project process, will be covered in this chapter. The system development methodology is a framework that helps the developer system, plan, and develop the system in a systematic manner. A few of the available methods for implementation of the project include the Waterfall Model, Agile Model, and Rapid Application Development (RAD) Model.

3.2 SYSTEM DEVELOPMENT METHODOLOGY

The Software Development Life Cycle (SDLC) is a process for creating elevated technology as quickly and affordably as needed. With the help of the SDLC, a business may quickly create high-quality software that has been carefully examined and is completely functional. The SDLC method was predicated on the processes of software development, that are described in specifies below. Requirement analysis, planning, software design, software development, testing, and deployment are all steps in the software development process. The waterfall methodology, Agile model, and Rapid Application Development (RAD) model are all widely known Software development models. This section will go through the three SDLC models in further depth.

3.2.1 Waterfall Development Method

The Waterfall Methodology is the best and earliest development method. This methodology is frequently employed in government initiatives as well as by a number of large corporations [12]. A project's linear flow from start to finish is emphasized by the waterfall method of task planning. This method, in which engineers plan ahead, specify in great detail, and execute consistently, is commonly used. One thing to keep in mind is that this model is set up in such a way that users are unable to progress to the next stage of development until the preceding one has been completed. This programming approach was given the name "waterfall" because it progresses downward, similar to how flowing water falls down a hill. [13].

When the comprehensive framework, data formats, and configurability design concepts are finished, the developer team begins working on the programmed. Connectivity and validating cannot begin till the full software has been set up. This indicates that certain unit tests are run throughout the project lifecycle; the code is not tested before the Testing phase. The application then communicates all of its evaluations and is turned on in production, allowing people to utilize it for the first time. Changes are highly slow and expensive with the Waterfall technique because it might take months or even years to complete, even if the end result doesn't satisfy consumer requirements. Defects are frequently overlooked and uncorrected.

Find out the demands and expectations of the firm first. This strategy makes it simpler to satisfy the expectations of the organization because the analytical team first analyses the business objectives and expectations. [14]. This procedure establishes the project's beginning and conclusion points. Because all of the steps are clearly defined, this approach guarantees that mistakes and misunderstandings are detected early in each level. Because resultant phases are dependent on the previous phase, this strategy allows for project timeline management. Other techniques require more resources to implement than the waterfall approach. [14].

On the other side, the waterfall methodology is blamed for consuming a lot of funds and resources. In the case in point, it requires multiple papers to be approved, modification implementation is expensive, cycles require a lot of time and rework, and problems are often postponed to subsequent stages. While several reasons for the waterfall paradigm's shortcomings have been identified, little research has directly addressed them [15]. The primary cause of failure in numerous studies has been identified as the handling of a vast scope, or demands that are not handled effectively.

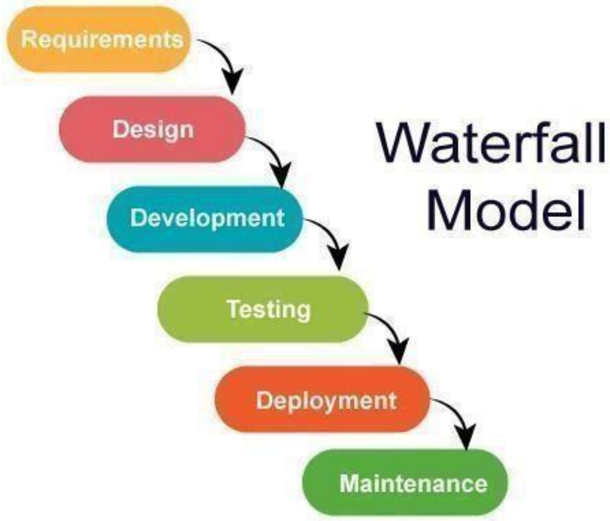


Figure 7 Waterfall Model Diagram

3.2.2 Agile Development Methodology

Fast movement is referred regarded as being agile. The basis of this software development lifecycle is the progressive and continuous approach to software development [13]. By utilizing user feedback to improve software, it continuously corresponds on concepts. Agile approach allows for the division of the development lifecycle into smaller parts known as "increments" or "iterations," as instead of one enormous procedure design as in regular SDLC. [16]. It demands continuing client interaction as well as ongoing improvement at every level. Once the job starts, teams go through a cycle of planning, doing, and evaluating. Both project clients and team members need to be engaged. The effects of each of these incremental increases on the regular growth stages.

Program is implemented in iterations using the agile software development process. At any stage of development, any improvements that update the programmed are accepted from the client, and those changes are applied [17]. Additionally, it helps in the quick discovery of errors, that are immediately rectified. However, because needs are constantly changing, it takes a lot of time and wastes resources [17]. Because the augmented element is rendered useless if customers are unhappy with the incomplete software developed in a particular iteration and change their needs.

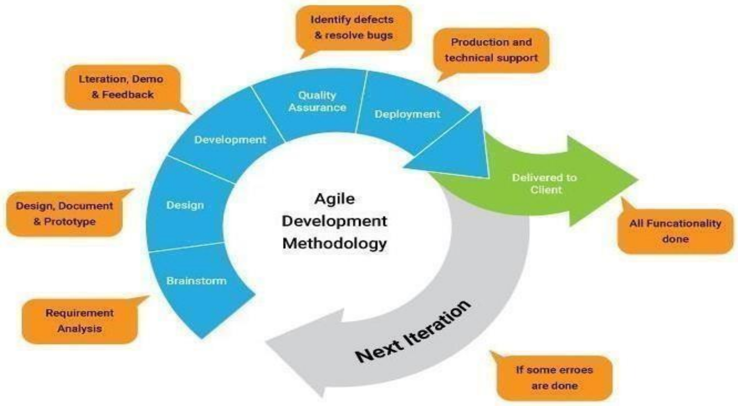


Figure 8 Agile Model Diagram

3.2.3 Rapid Application Development Methodology

The rapid application development (RAD) model has been a low-risk application advancement model based on agile methodologies that emphasize rapid prototypes and iterations [18]. The rigidity of other conventional software development approaches, which make it challenging to continue improving once the initial development is complete, inspired the Rapid Application Development (RAD) approach, a software design process. The RAD technique is created to be adaptive to changes and to accept new inputs, such as extensions and functions, at every stage of the development process. In his book, which is appropriately titled Rapid Application Development, James Martin developed the term Rapid Application Development (RAD).

For limited deadlines, rapid application development is excellent. A RAD framework may be the ideal solution if you need to produce anything rapidly. Rapid application development software is the best solution if you may not have sufficient opportunity to analyses and build your demands. Rapid application development is a flexible method that allows for quick changes. Rapid application development enables businesses to test their products in the actual world faster than traditional development methods. Software developers can display development in addition to RAD's emphasis on sprints and iterative work. Against conventional systems such as a waterfall, where organizations had to waiting until the last stage to see the outcomes, this technique allows organization to see the products right away. Rapid application development focuses on user feedback to guarantee that concepts are constantly updated to match customer requirements.

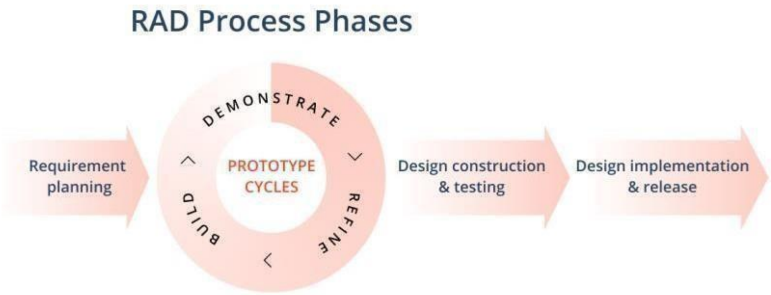


Figure 9 RAD Model Diagram

Table 2 Comparison SDLC

Methodology	Organizational Structure	Techniques	Advantages	Limitations	Usage
Waterfall	Linear	Only after the established set of objectives for the prior phase have been achieved and authorized can the subsequent phase start.	• Clearly accessible and explainable. • Early detection of errors.	• Need a lot of money and efforts • Difficult to maintain • Not suitable for complicated projects	• Requirements are clear and fixed that may not change. • The project is simple, and the cast is minimal.
Agile	Iterative	In agile development, tasks are divided within time boxes (concise time periods) that offer particular aspects for a release.	• Reduced risk • Early detection of errors • Software is regularly delivered	• Poor resource planning • Limited documentation	• To develop a new feature • Short, flexible timeline. • Budget flexibility.
Rapid Application Development (RAD)	Iterative	Four phases make comprise a quick application development cycle. quick development through multiple iterations and ongoing feedback	• Can be change at anytime • Reviews are quick • With fewer workers, user can be more productive.	• More complex • User requirements are required during the production cycle.	• If the budget allows for the usage of automatic code generators. • It is possible to modify the systems.

3.3 RAPID APPLICATION DEVELOPMENT MODEL

Supported by the research article, the RAD method was determined to be the best appropriate technique for this suggested project. In comparison to other software development models, the rapid application development (RAD) model is the most appropriate methodology for planning, structuring, and developing this system. This is because the RAD model can speed up the development of an automatic pet feeder system. Modification and improvement of the automatic pet feeder are possible during the development phase of the RAD model. This will improve the quality of the automatic pet feeder system because responses and evaluating were used to build it. The five steps of the RAD model are evaluation and fast planning of the project, development phase, testing phase, and implementation phase. Split up the development phase into three parts: build phase, demonstrate phase, and refine phase. These three stages will function as a cycle, allowing developers to modify or keep improving the automatic pet feeder based on responses. As a result, during this development phase, the developer is prepared to assure the framework, design, software, and software-hardware link.

3.3.1 Requirement Planning Phase

During the requirement planning phase, processes such as identifying the system's purposes and gathering the needs required to meet those performance targets would be performed. For this current proposal, targets have been specified. It's the first phase before implementation, and requirement analysis ensures a better grasp of the system. The goal of this project is to create an Internet of Things (IoT) application for a pet care system, namely a pet automated feeder. The system's purpose is to provide for the feeding of predefined amounts of food or medicine to pets at predetermined periods when the owner is away or otherwise occupied. Pet owners will be the target users since they will be able to implement this system for monitoring and feed their pets while they are not at home or forget to feed them.

Understanding the requirements of the users, who are pet owners, is required in this phase, for example, monitoring their pets' condition when they leave them alone at home with no one to watch them, and pet feeding because some pet owners do not feed their pets on time according to their busy schedules. Furthermore, they do not have the opportunity to feed their pet sensibly when they must travel and therefore must leave the food in the pet bowl for their pet. This circumstance causes the food to deteriorate and has the potential to harm their pet's health. The finding is made by looking at existing pet feeding systems for a better understanding.

3.3.1.1 Functional Requirements

Table 3 Functional Requirement

<table>
<tr><th>Module</th><th colspan="2">Functional Requirements</th></tr>
<tr><td>Login and registration</td><td colspan="2">• The user should be able to register on the application for those who are not registered yet.
• The user should be able to fill out the registration forms and save the details in the database.
• The registered user should be able to login by entering the correct username and password.</td></tr>
<tr><td>Forgot Password</td><td colspan="2">• The user should be able to enter their email to get the link from the system.
• The system should be able to send the link to user's email for user enter the new password.</td></tr>
<tr><td>User Manual</td><td colspan="2">• The system should allow the user to click on the "User Manual" button to view the user manual of the Automatic Pet Feeder Machine.</td></tr>
<tr><td>Manage Pet Feeder</td><td>Feeding Schedule</td><td>• The system should allow the user to click on the "Manage Pet Feeder" button to manage food's weight, feeding schedule and monitor pet.
• The user should be able to set food schedule by entering the time of the food schedule.
• The system should allow the user to click on the "Update" button after user enters the latest time.</td></tr>
</table>

	Food's Weight	- The user should be able to set food weight by entering the weight in grams. - The system should allow the user to click on the "Update" button after user add the latest food weight.
	Monitor Pet	- The system should allow the user to view their pet's condition using the application. - The system should connect to the esp32-cam to display the video of the pet in actual time.

3.3.1.2 Non-functional Requirements

Table 4 Non-functional Requirements

Non-functional Requirements	**Description**
Performance	- Login validation should be done less than 10 seconds. - The pet feeder machine should operate perfectly the dispense the pet's food and to monitor the pets.
Usability	- The system should be a user- friendly system so that the users are easy to use it. - It is quite simple to set up the automatic pet feeding equipment.

Maintainability	• The system should be able to maintain, if there is a software update. • Pet owners can easily maintain pet feeder machines, such as cleaning it, filling food, and so on.

3.3.1.3 Constraints and Limitation

- Users must have access to the internet in order to utilize the mobile application to keep track of their pets in actual time.
- Only authorized user can use the system.
- Only one animal may be fed at a time by the automated pet feeder machine.
- Only Android users will be able to install the mobile application.

3.3.2 Prototype Cycle Phase

A literature review will be conducted, so a study of other current relevant systems that can guide in the prototype cycle phase of this proposed system and in the generation of ideas. It's also a means to enhance the system's functionality. NuriPet: A Smart Pet Feeding Machine for SNS, Smart Dog Feeder Design Using Wireless Communication, MQTT and Android Client, and Pet food auto feeder using Arduino are among the current systems. For system enhancement, an evaluation of current systems with the proposed system is also done.

After gathering the system 's specifications, the prototype cycle phase, which involves developing system models, will continue. System design and interface design are among the tasks that will be completed. In system design, approaches such as flow charts, use case diagrams, and context diagrams are used to show how the system interacts with the end user. The flow chart depicts the procedure through which the automatic pet feeding works. The user's engagement with the system is shown in the use case diagram. The system's users are people who own pets. Users, generally referred to as performers, can perform duties including timing their pet feedings, modifying the size of their pet's meals, and more. Data flow from the Arduino microcontroller to the system, as well as data flow from the system to the user or the other way around, are all intended to be represented in the context diagram. After that, a model of the user interfaces will be created, with the storyboard method being used to show how the pet feeder system's interface should be navigated. During this step, the best building method will be chosen.

3.3.3 Design Construction and Testing Phase

The system development will be prioritized during the design, construction, and testing phases. On the proposed system, stages such as development and testing would be implemented. Writing code for an automated pet feeder system is a method of putting the proposed system into action. The Arduino IDE will be used to build the code for how the micro-controller works, while IONIC and Firebase will be used for developing the automated pet feeder interface and deploying the database. C and Java are two of the programming languages that are employed. In addition, any system updates or improvements will be implemented throughout this time. Development process, coding, and system testing are among the activities that will be completed.

3.3.4 Design Implementation and Release Phase

This is the design, implementation, and release phase, in which the system’s production is complete and it is ready to proceed. The automatic pet feeder system will be thoroughly tested to ensure that it operates and functions properly. Its purpose is to guarantee that the system satisfies the standards as well as the automated pet feeder's objectives and scope. During this phase, the testing will be done using a sample of pet’s food to see if it meets the system's goals and specifications. Steps such correcting or minimum debugging will be carried out in the phase, for example, system code change or improvement, in order to reduce the inefficiency of the proposed system. The system is able to deploy after the testing phase is completed; however, in this project, analysis will be conducted on the system prototype.

3.4 PROPOSE DESIGN

3.4.1 Architecture Diagram

The microcontroller used in this proposed system is a NodeMCU ESP 32. Since it is an automatic pet feeder system, hence it consists of the sensor such as the weight sensor. The ESP 32-Cam, sound module, and motor servo are the other hardware components. The components were linked to the microcontroller NodeMCU ESP 32. Using a microcontroller with Wi-Fi capabilities, data may be sent from the device to the cloud and then to other devices, like a phone. Figure 10 depicts the proposed system's architectural layout.

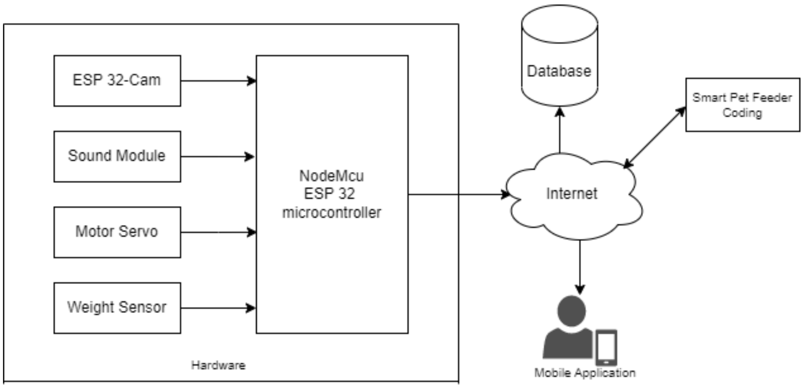


Figure 10 Architecture diagram

3.4.2 Overall System Design

Figure 11 show the overall system design for this project. The user will use the mobile application to set the feeding schedule and to set the food's weight. The Pet feeder machine will connect to the mobile application via WIFI. When the time to feed their pets is correct, the machine will dispense the food with correct weight.

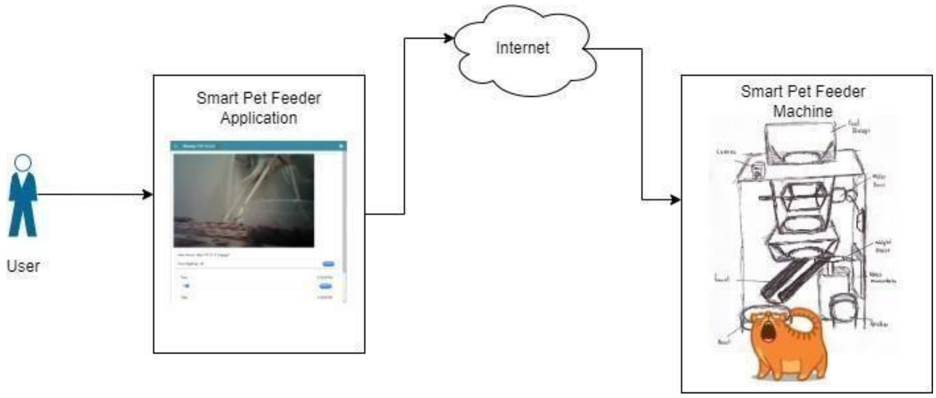


Figure 11 Overall System Architecture

Figure 12 show the sketch of the Smart pet feeder machine.

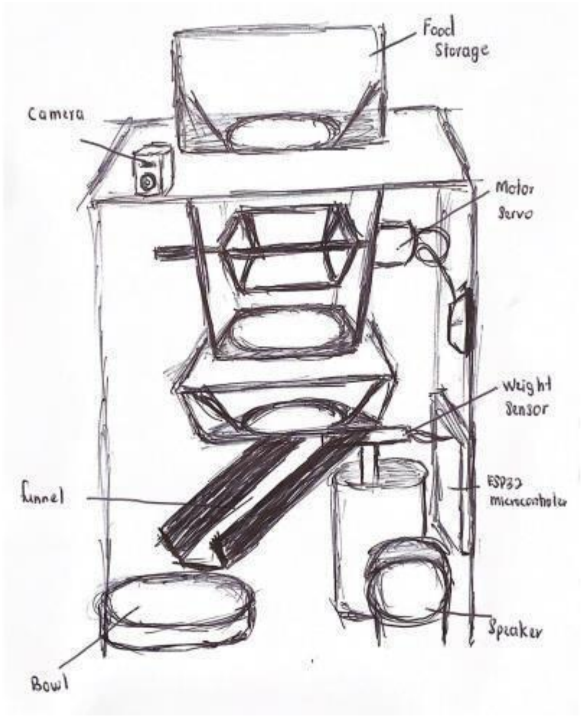


Figure 12 Sketch of Smart Pet Feeder Machine

3.5 FLOWCHART

3.5.1 Hardware Flowchart

Figure 13 shows the hardware flowchart for this system. The hardware consists of an Arduino microcontroller, and sensors. The hardware is required to operate the automatic pet feeding in accordance with the user's command through the mobile application. The sensors will transmit the message to an analogue module, which will convert it to a digital signal. The digital signal will then be transferred to the Arduino board for processing data, which will result in the user's readable instruction. To permit data transfer, the Arduino's power must be powered on, and the power source must be supplied, as the sensors require electricity to work.

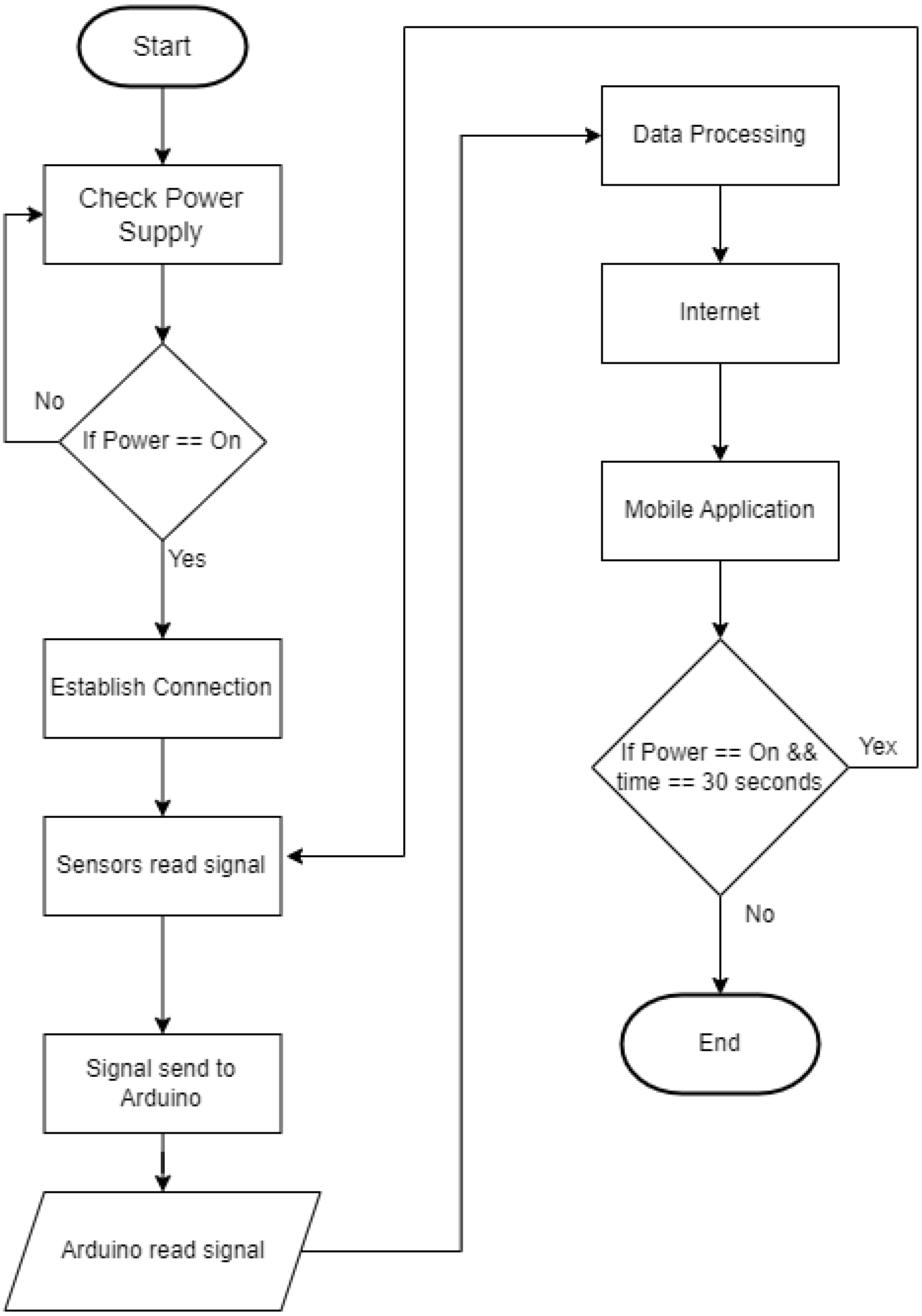


Figure 13 Hardware Flowchart

3.5.2 Software Flowchart

3.5.2.1 User Login Flowchart

Before accessing the mobile application, users must first log in. Only the user is authorized can use the mobile application after inputting a valid password and ID. Verification will be done using the login details. If the user's ID and password meet those in the database, the user is permitted access to the mobile application. If the user's identification is validated as that of a user, the user is permitted access to the user interface. Figure 14 show the login flowchart.

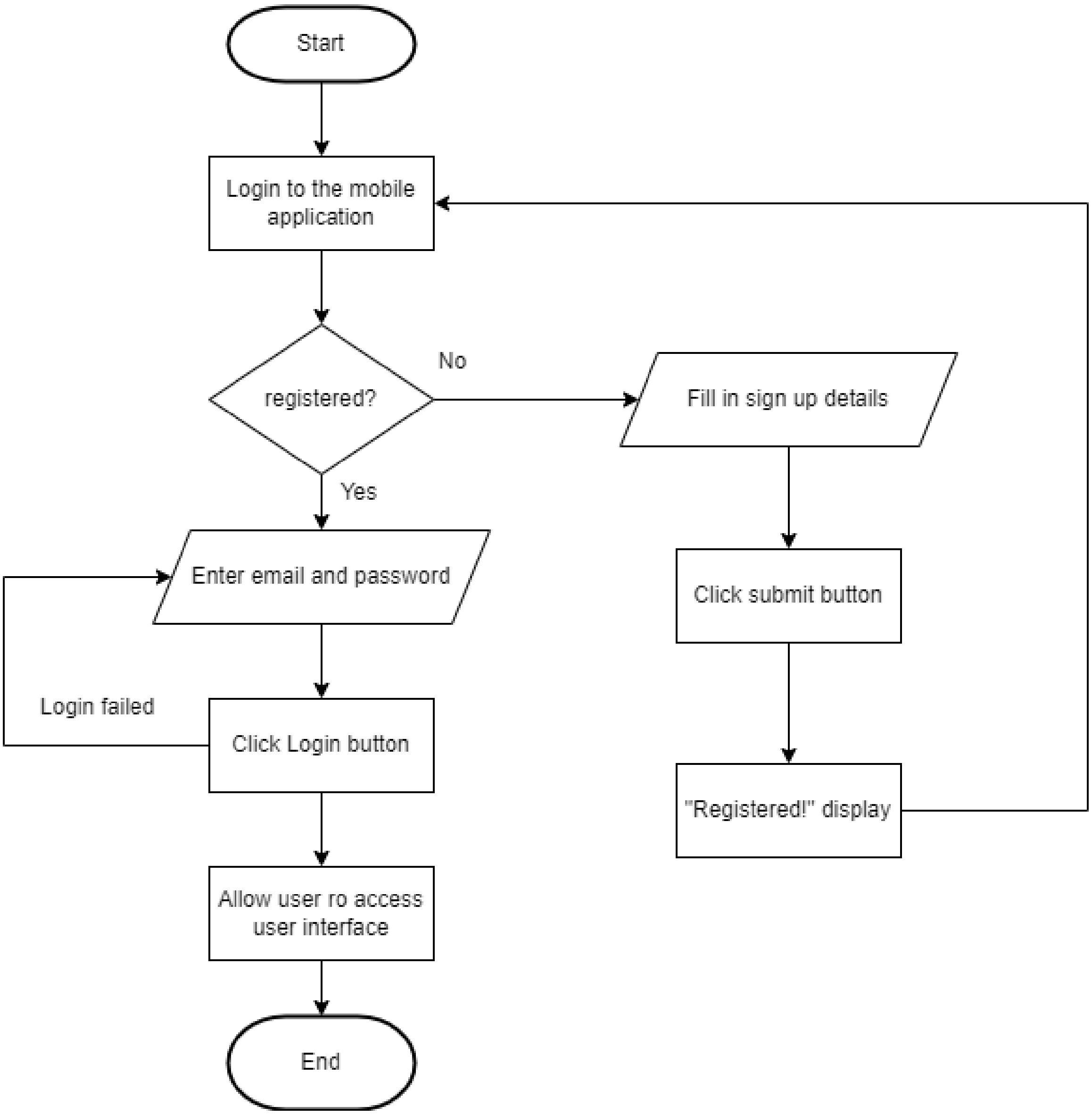


Figure 14 Login flowchart

3.5.2.2 Main Activity Flowchart

After entering into login, the mobile app, the user will be directly going to the user interface, at this page user can choose to click on manage pet feeder button or user manual button. If the user clicks on manage pet user button, user will go directly to manage pet user page where user can monitor their pet, manage feeding schedule and set the weight for the pet's food. However, if the user chooses the click on user manual button, the user will go directly to the user manual page where the user can get more information regarding the smart pet feeder machine. Figure 15 below show the main activity flowchart.

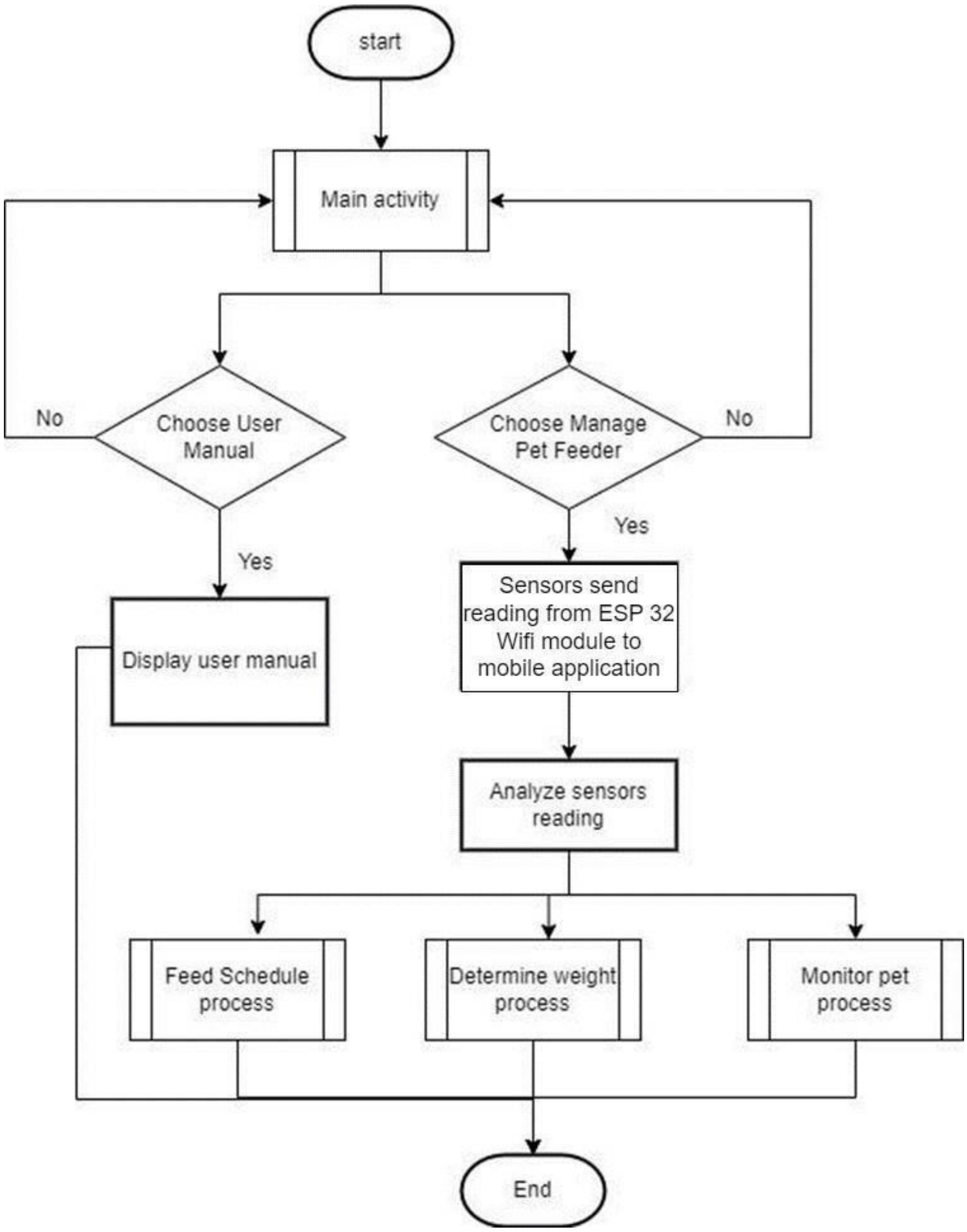


Figure 15 Main Activity Flowchart

3.5.2.3 Feed Schedule Flowchart

If the user selects the activity of providing commands to the machine, the user must also select the commands to be given. If the user wants to schedule a pet feeding time, he or she can do so by selecting the feed schedule button. Users may arrange the machine to deliver food to their pets at any moment. The data will be transmitted to the database when the time has been set. The app will also display the time period that the user has chosen to dispense the food. When it is time to dispense the food, the machine will ensure that it is dispensed in accordance with the weight that has been set. Figure 16 show the feed schedule Flowchart.

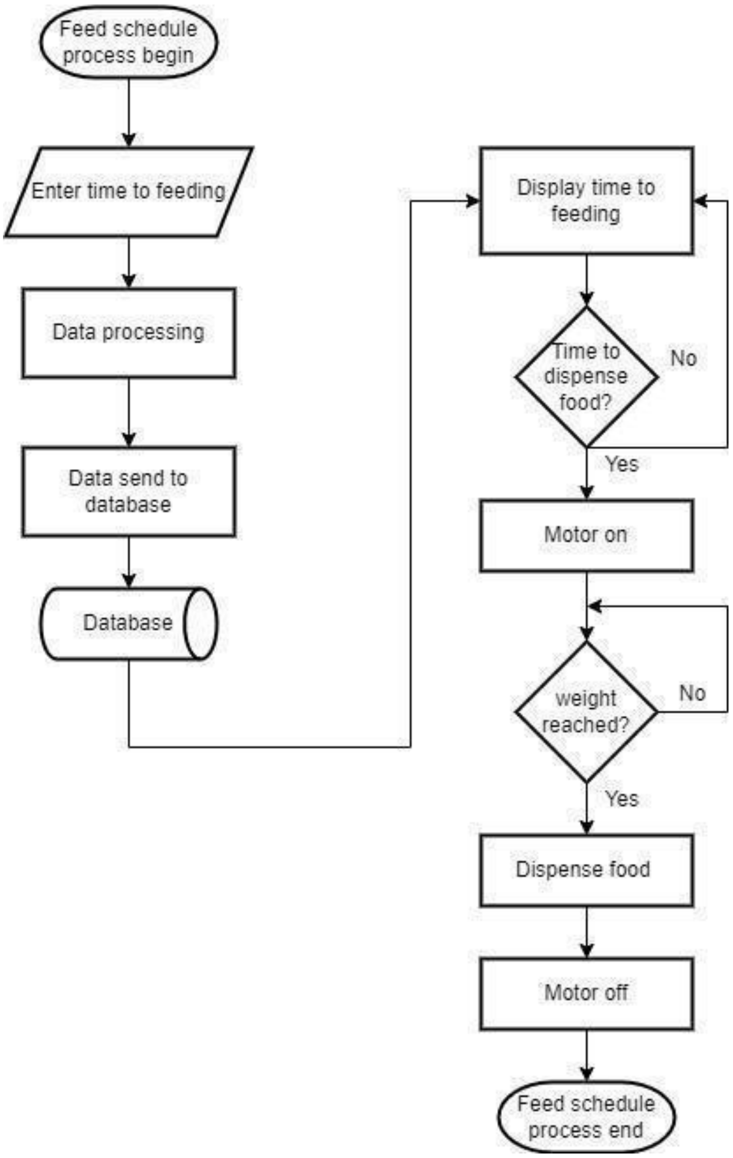


Figure 16 Feed Schedule Flowchart

3.5.2.4 Determine Weight Flowchart

The user can choose to set the weight of the food by pressing the button to determine the weight of the food. Following that, customers may specify the weight of the food they wish to provide to their pet. The weight sensor will then measure the weight of the meal duringfeeding to fit what the user has specified. The food will be served after it has matched. Figure17 show the flowchart of determine weight.

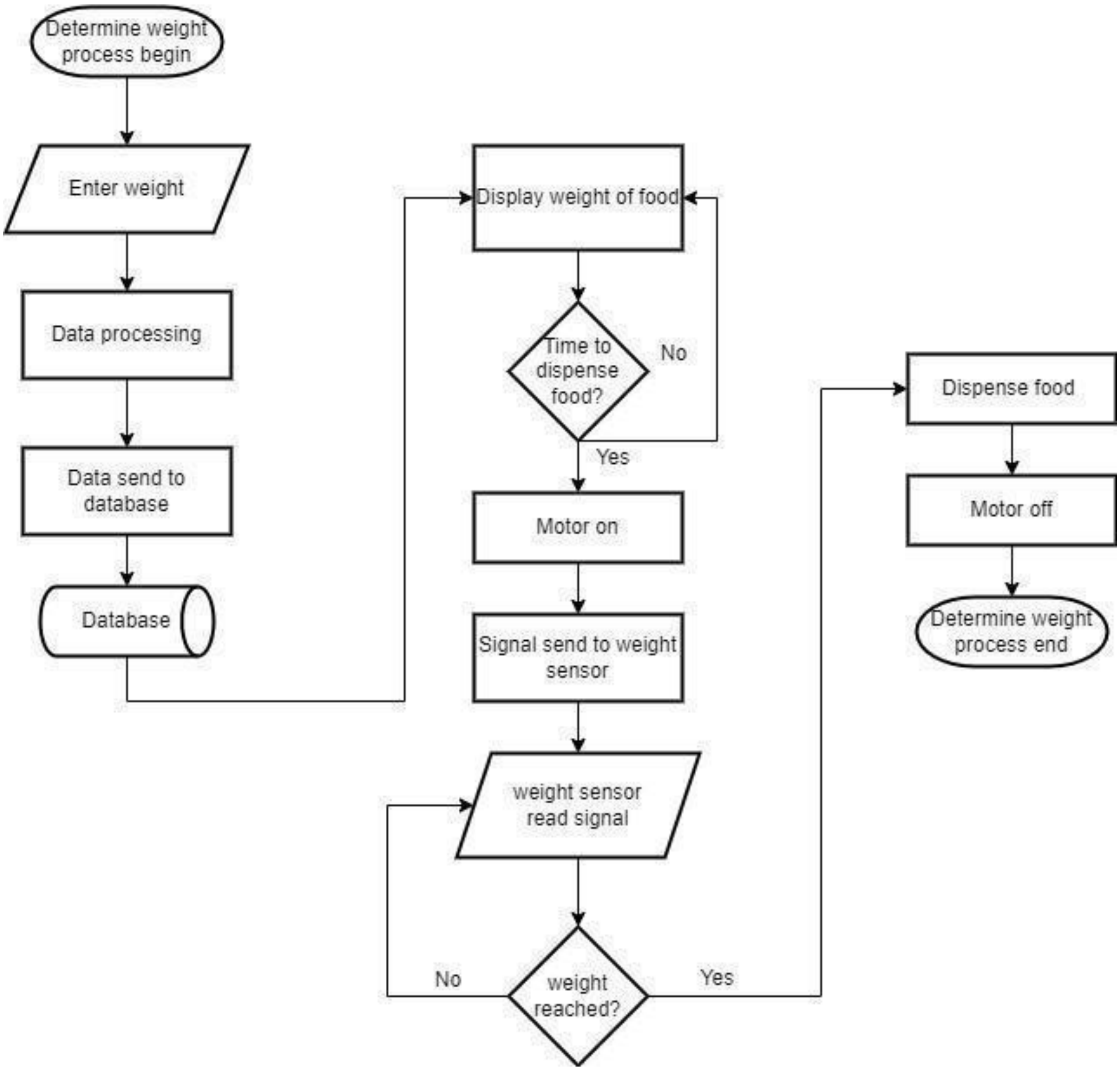


Figure 17 Determine weight flowchart

3.5.2.5 Monitor Pet Flowchart

Users may monitor the situation of their pets by using the monitor pets' button. The ESP32cam will then receive a signal, and the camera will display their pet's current situation in real time. As a result, users can view the condition of their pets at home via the camera. Figure 18 show the flowchart for Monitor pet.

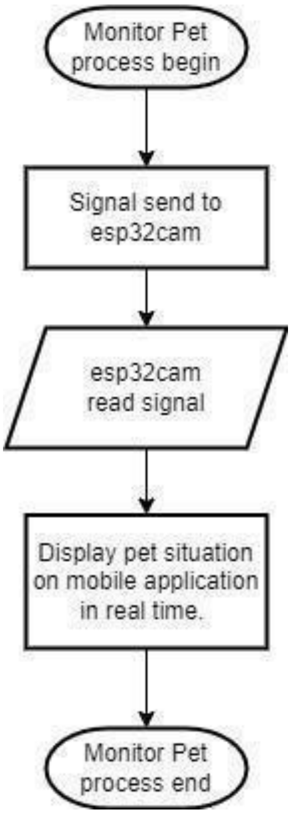

Monitor Pet process begin
Signal send to esp32cam
esp32cam read signal
Display pet situation on mobile application in real time.
Monitor Pet process end

Figure 18 Monitor Pet Flowchart

3.6 Context Diagram

The context diagram defines the system's context and limitations, as well as the relationship between the system and external entities by displaying the flow of information between the system and the external entities. The context diagram of the proposed system is shown in Figure 19. Pet's owner, and controller are examples of external entities. Feeding schedule time, adjust the food weight, and monitor their pets. Finally, the controller may read and update data from a database created by the pet owner, such as feeding schedule, food weight and monitoring details.

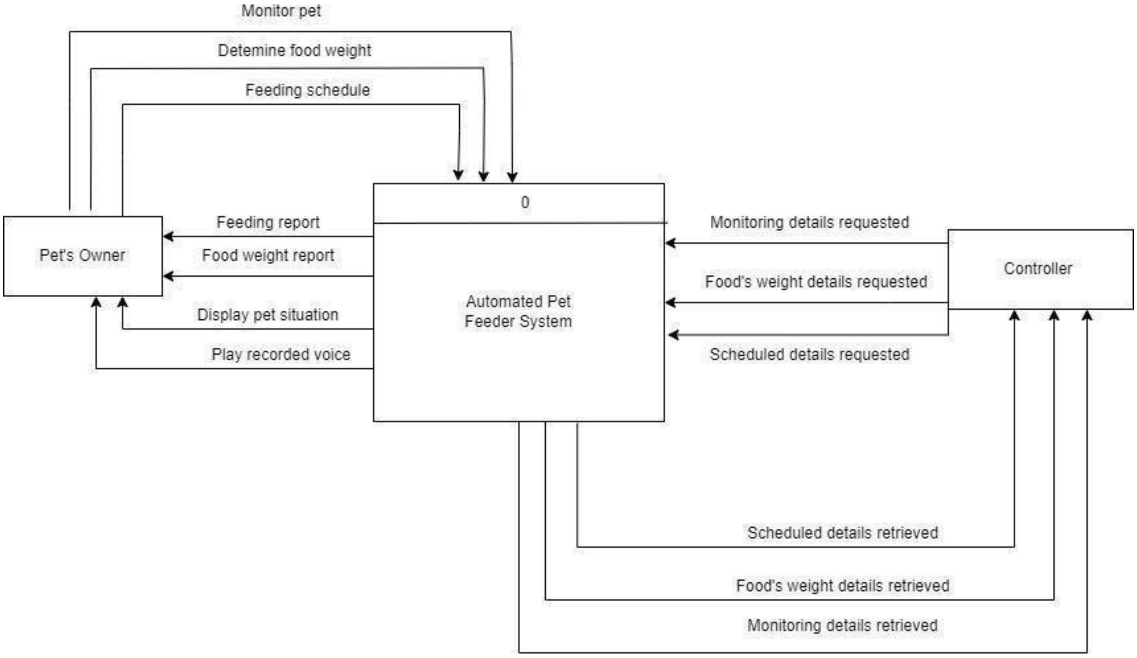

Monitor pet
Detemine food weight
Feeding schedule
0
Pet's Owner
Feeding report
Food weight report
Display pet situation
Play recorded voice
Automated Pet Feeder System
Monitoring details requested
Food's weight details requested
Scheduled details requested
Controller
Scheduled details retrieved
Food's weight details retrieved
Monitoring details retrieved

Figure 19 Context Diagram

3.7 Use Case Diagram

The use case diagram is a visual depiction of the system's behaviors. Figure 20 depicts the interactions between the system, the actors, and the use case. The use case refers to the actions that the actors might be doing. To gain access to the system, all users must login. The pet's owner may Login or register, reset password, rearrange feeding times, determine the weight of their food and monitor their pet's situation. The action for the ESP32 microcontrollers is to retrieve the data from the database such as schedule details and food’s weight details. Figure 20 show the use case diagram for this project. Lastly, the function of the developer is to manage software and also the error occur in this system.

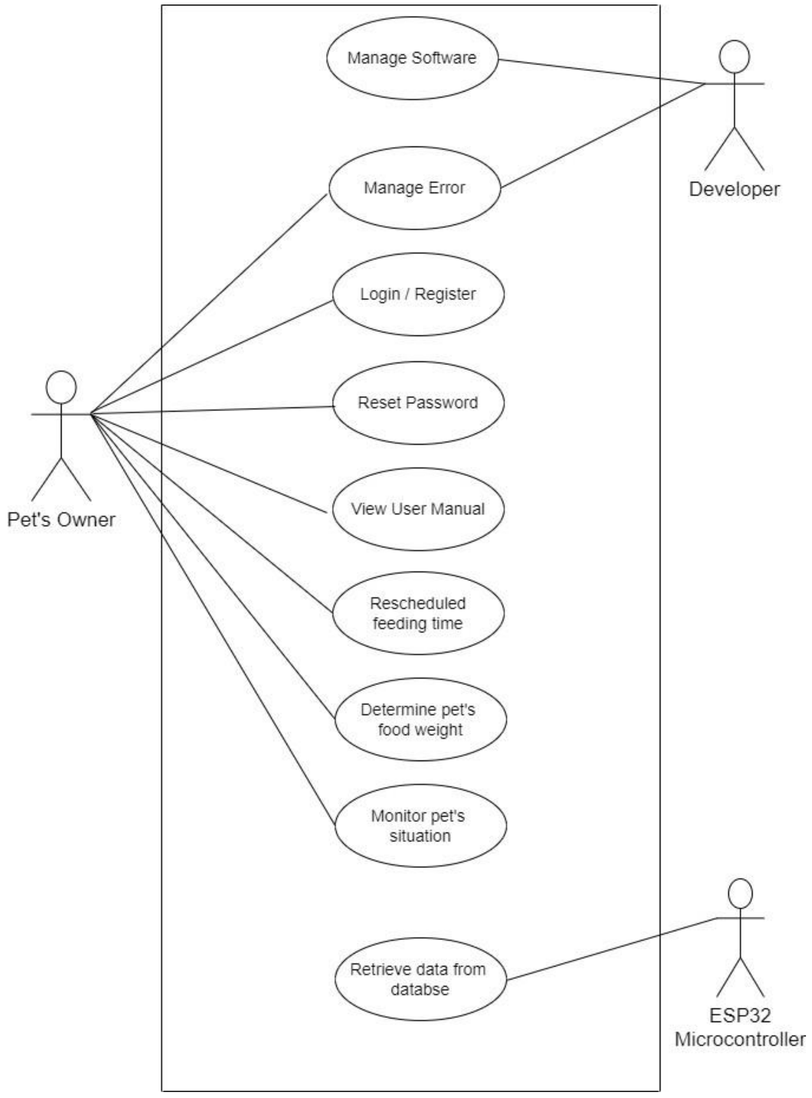


Figure 20 Use Case Diagram

3.7.1 Use Case Description

Table 5 Use Case Description - Register

Use Case ID	Register
Brief Description	This use case is about the user need to create an account before start using the application.
Actor	Pet’s Owner
Pre-Conditions	• The user needs to go into the sign up page. • User must fill in the details to create an account, such as email, password, first name, last name, and password. After that, user need to press “Register” button.
Basic Flow	1. The use case starts when the website displays the sign up page. 2. The user needs to enter their details such as username, password, full name, email, and phone number. 3. The password needs to be 6 minimum characters. [**C1- Character of password does not follow the rules**] **[R1 – Character of password]**. 4. After filling all the information, customers need to press the “Register” button. 5. Then, it will go to the login page. 6. The use case ends.
Alternative Flow	None
Exception Flow	None
Post-Conditions	• The registration process is successful. • The user already have an account and they can start use the application.
Rules	**[R1 – Character of password]** User must insert the password that follow the rules which is 6 minimum characters.

Table 6 Use Case Description - Login

Use Case ID	Login
Brief Description	This use case is about the user logging in their username and password before they want to proceed with another step.
Actor	Pet's Owner
Pre-Conditions	• The user must have at least one account that they have signed up for before doing any login process. • The user needs to sign up and fill their details first.
Basic Flow	1. The use case starts when the application displays the Login page. 2. The user enters their email address and password [**E2 - Forgot password**] **3.** The user will click the Login button to proceed. [**E1 - unable to login**] 4. Then, the user will click the Login button to proceed. 5. The use case ends.
Alternative Flow	None
Exception Flow	**E1 – Unable to login** When the user is unable to Login the website, then it will appear the message. **E2 – Forgot Password** When the user forgets their password, a message will appear and there is the 'Reset Password' button for the customer to reset their password.
Post-Conditions	• The current account of the customer is now logged in. • The login process is successful.
Rules	None
Constraints	None

Table 7 Use Case Description - Reschedule Feeding Time

Use Case ID	Reschedule feeding time
Brief Description	This use case is about the user can reschedule the feeding time by insert the wanted time of feeding.
Actor	Pet's Owner
Pre-Conditions	• User already inside the automatic pet feeder application system. • The user has selected the "Manage Pet Feeder" button.
Basic Flow	**1.** The use case starts when the application displays the Manage Pet Feeder page after user click on the "Manage Pet Feeder" button onthe homepage. **2.** Then the Manage Pet Feeder page will display monitor pet, food's weight, and list of feeding schedule. **3.** The user needs to enter a new feeding time by scrolling the time and clicking on "Update" button. **4.** The use case ends.
Alternative Flow	None
Exception Flow	None
Post-Conditions	• Adding or updating the schedule list process is successful. • The new schedule list is updated in the database.
Rules	None
Constraints	None

Table 8 Use Case Description - Determine pet's food weight

Use Case ID	Determine pet’s food weight
Brief Description	This use case is about the user can set the food’s weight by insert the wanted weight of pet’s food in grams.
Actor	Pet’s Owner
Pre-Conditions	• User already inside the automatic pet feeder application system. • The user has selected the “Manage Pet Feeder” button.
Basic Flow	**1.** The use case starts when the application displays the Determine food’s weight page after user click on the “Manage Pet Feeder” button on the homepage. **2.** Then the Manage Pet Feeder page will display monitor pet, food’s weight, and list of feeding schedule. **3.** The user will insert the food’s weight in grams and then click on the “Update” button. **4.** The use case ends.
Alternative Flow	None
Exception Flow	None
Post-Conditions	• Adding or updating the food’s weight process is successful. • The new food’s weight list is updated in the database.
Rules	None
Constraints	None

Table 9 Use Case Description - Monitor pet situation

Use Case ID	Monitor pet's situation
Brief Description	This use case is about the user can monitor their pet's situation.
Actor	Pet's Owner
Pre-Conditions	• User already inside the automatic pet feeder application system. • The user has selected the "Manage Pet Feeder" button.
Basic Flow	**1.** The use case starts when the application displays the Monitor pet page after user click on the "Manage Pet Feeder" button on the homepage. **2.** Then the Manage Pet Feeder page will display monitor pet, food's weight, and list of feeding schedule. **3.** The system will display their pet's condition. **4.** The use case ends.
Alternative Flow	None
Exception Flow	None
Post-Conditions	• The pet's situation successfully displayed.
Rules	None
Constraints	None

3.8 Activity Diagram

Figure 21 show the activity diagram.

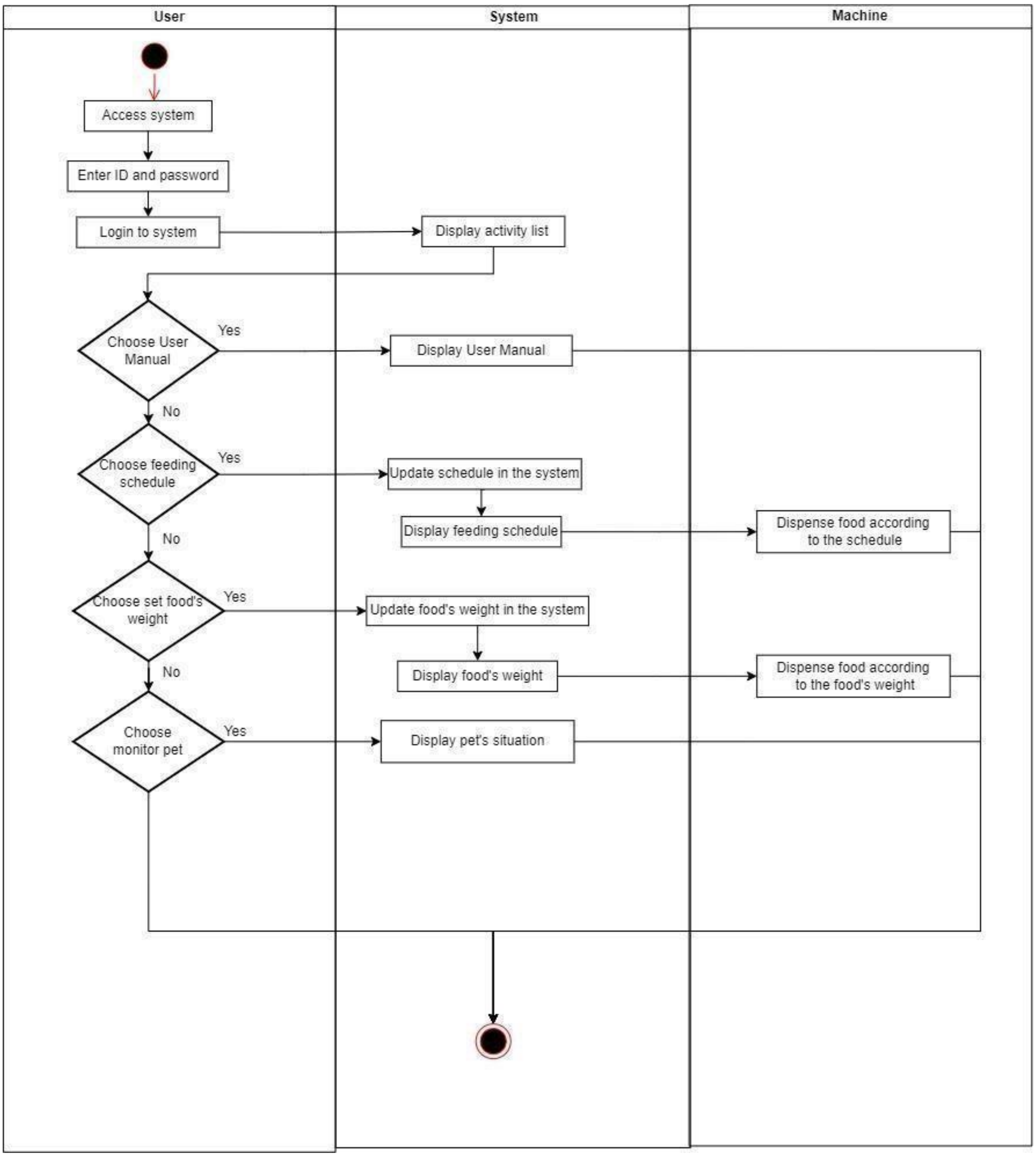


Figure 21 Activity Diagram

3.9 Database

A database is a collection of organized data that allows for easy management, access, and update. The database plays a significant role in data storage in this proposed system. It saves user information, scheduled time information and food weight information.

3.9.1 Data Dictionary

Pet’s Owner

Table 10 Data Dictionary for Pet's Owner

Attribute	Type	Length	Key	Description
userID	varchar	10	Primary (PK)	Unique ID for pet’s owner
email	varchar	50		Pet owner email
password	varchar	10		Pet owner password
firstname	varchar	100		Pet owner first name
lastname	varchar	100		Pet owner last name

Feeding Schedule

Table 11 Data Dictionary for Feeding Schedule

Attribute	Type	Length	Key	Description
scheduleID	varchar	10	Primary (PK)	Unique ID for schedule time
userID	varchar	10	Foreign (FK)	Unique ID for pet’s owner
time	time			Pet feeding time

Food’s Weight

Table 12 Data Dictionary for Food Weight

Attribute	Type	Length	Key	Description
weightID	varchar	10	Primary (PK)	Unique ID for food’s weight
userID	varchar	10	Foreign (FK)	Unique ID for pet’s owner
foodweight	float			Food weight

Controller (NodeMCU)

Table 13 Data Dictionary for Microcontroller

Attribute	Type	Length	Key	Description
deviceID	varchar	10	Primary (PK)	Unique ID for microcontroller
userID	varchar	10	Foreign (FK)	Unique ID for pet’s owner

3.9.2 Entity Relationship Diagram (ERD)

Entity Relationship Diagrams (ERD) are used to simplify a relationship and make it a lot easier to comprehend. This ERD below depicts the relationship and properties of five entities in the database. The entities are the pet owner, the feeding schedule, the food weight, and the controller (NodeMCU). Each thing has its own number of attributes. Figure 22 show the entity relationship diagram.

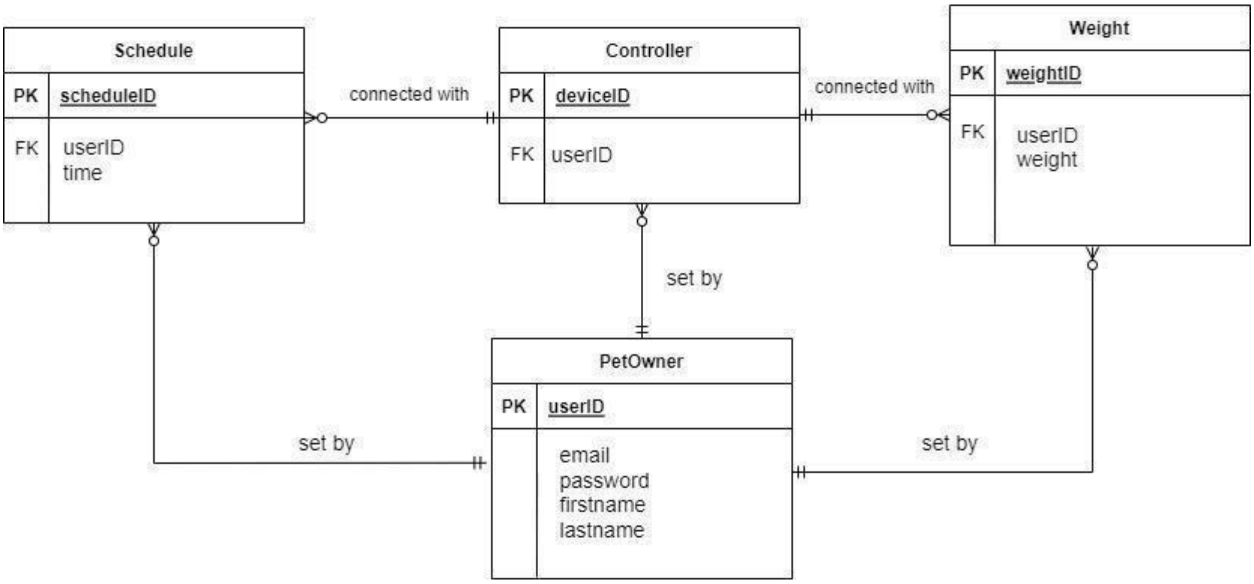
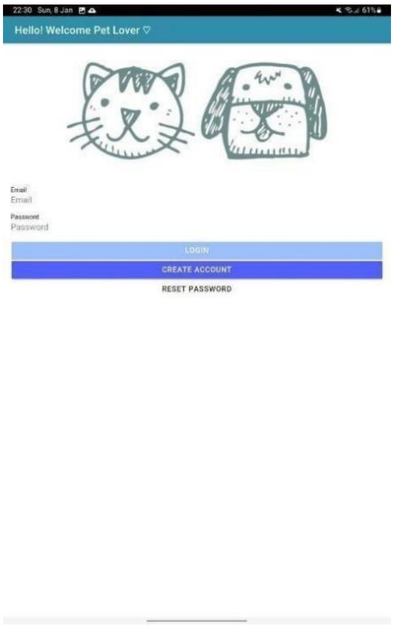


Figure 22 Entity Relationship Diagram

3.10 Proposed Interface Design

3.10.1 Login and Sign Up



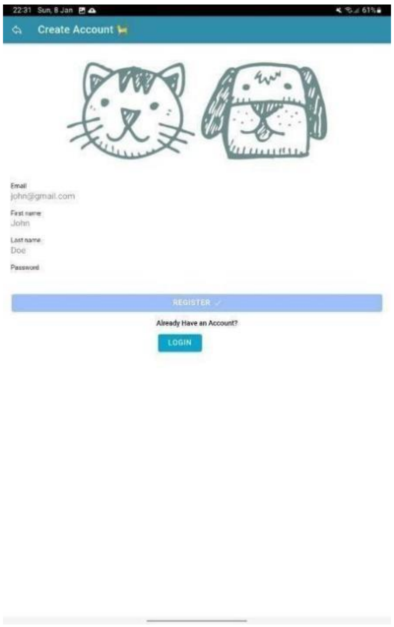


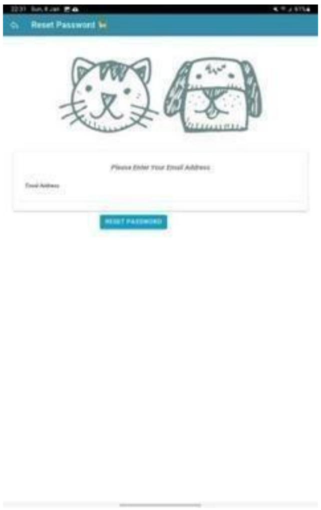
Figure 23 Login Interface

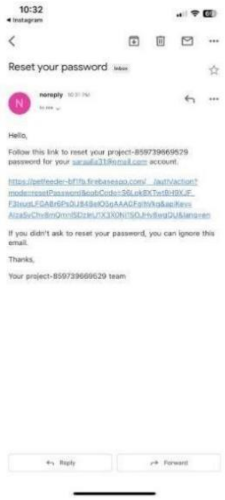
Figure 24 Sign Up Interface

Figure 23 The Login Interface for the User displays an interface that allows the user to log into the Automatic Pet Feeder System. To access the system, users must provide their Username and Password. If the user does not yet have an account, they may sign up by clicking the Register button, and if they forget their password, they can click the Reset Password button.

Figure 24 show the user sign-up screen allows the user to register for the automatic pet feeder system. Before logging onto the system, users must first create their own account. Users must provide their Email Address, Password, First Name and Last Name.

3.10.2 Reset Password Interface





Reset your password

Hello,

Follow this link to reset your project-859739669529 password for your saraalia31@gmail.com account.

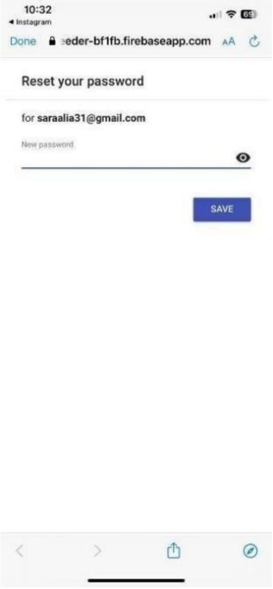
https://petfeeder-bf1fb.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=[illegible]&apiKey=[illegible]&lang=en

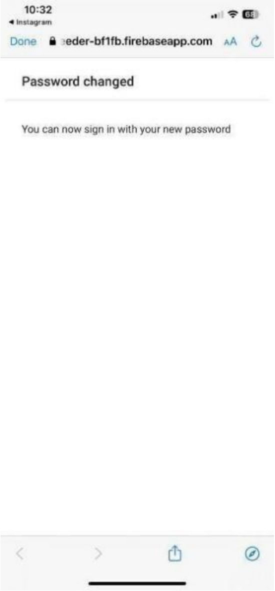
If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your project-859739669529 team

Figure 25 Reset Password Interface **Figure 26 Received Email to reset password**





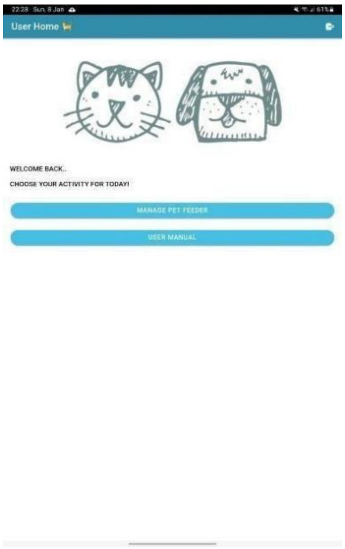
Password changed

You can now sign in with your new password

Figure 27 Enter new password **Figure 28 Successfully reset password**

Reset Password interface allow the user to enter the email address if the user forgot their password. After user click on the reset password button, the user will receive the email that contain link to enter new password. If user click on the link given, the pop-up message will appear to enter new password. After click on save, the user can use the application by enter new password. Figure 25 until figure 28 show how the process to reset the password.

3.10.3 User Home Interface



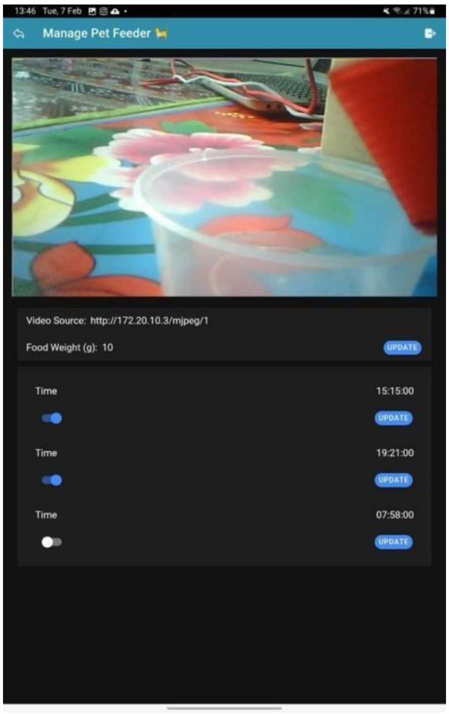
WELCOME BACK..

CHOOSE YOUR ACTIVITY FOR TODAY!

Figure 29 User Home Interface

Figure 29 show the main page for the user after login to the system. Users may perform any activity by simply pressing any of the accessible buttons to get to the activity page. However, if the user decides to leave this system, the user can do so by clicking the Log Out button at the top right of this page.

3.10.4 Manage Pet Feeder Interface



Video Source: http://172.20.10.3/mjpeg/1

Food Weight (g): 10

Time 15:15:00

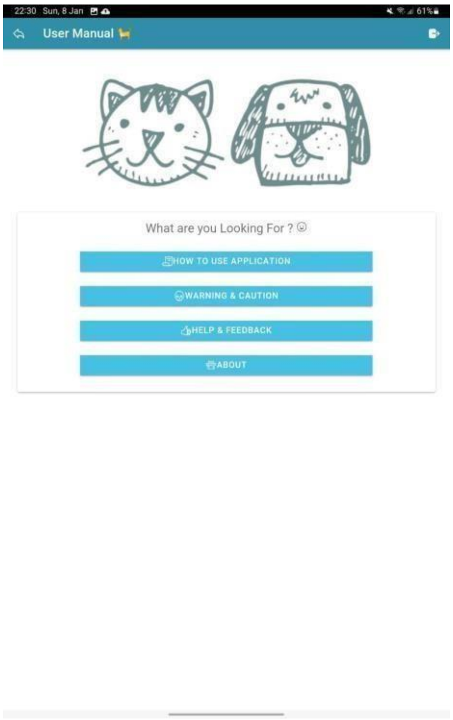
Time 19:21:00

Time 07:58:00

Figure 30 Manage Pet Feeder

Figure 30 shows the manage pet feeder interface that allow user to monitor their pet, manage food’s weight and manage feeding schedule. The user can enter the food’s weight in gram and then click on update button to save the data in the database. In addition, user also can select time to feed their pet, user can scroll the time to choose which time their want feed their pet, next click on update therefore the automatic pet feeder machine will dispense the food according to the time and weight that has been entered by user. However, if the user decides to leave this system, the user can do so by clicking the Log Out button at the top right of this page.

3.10.5 User Manual Interface

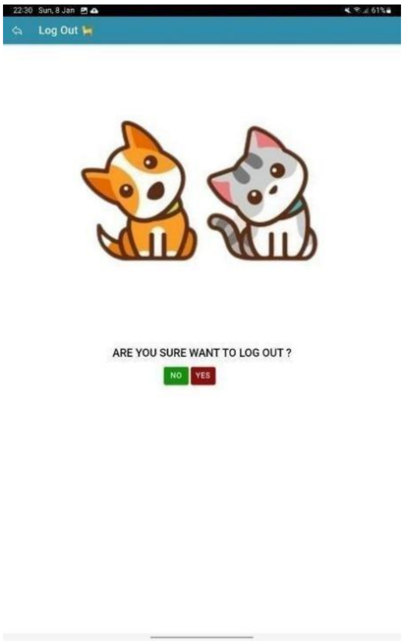


What are you Looking For ?

Figure 31 User Manual Interface

Figure 31 show the user manual interface. This page will explain on how to use the Automatic Pet Feeder machine to make user ease to use the machine. However, if the user decides to leave this system, the user can do so by clicking the Log Out button at the top right of this page.

3.10.6 Log Out Interface



ARE YOU SURE WANT TO LOG OUT ?

Figure 32 Log Out Interface

Figure 32 show the log out interface. User will be taken to this page after clicking the logout button. The user will be prompted on this page to confirm their want to log out. If a user wishes to log out, they can click the Yes button. If they don't, they can click the No button, which will take them to the User Home screen.

3.11 Potential Use of Proposed Solution

Although the Automatic Pet Feeder system is a new application, it has the potential to become one of the most popular pet feeder applications with the right amount of work and dedication. Despite the fact that the system is still in development, it will be beneficial to pet owners who are always busy with their daily lives. Furthermore, as the system gains more profit and users, it can be expanded across the state or even nationally. The system is also intended to assist pet owners in caring for their pets, not only in terms of feeding times, but also in terms of their mental and physical well-being.

3.12 HARDWARE AND SOFTWARE

3.12.1 Hardware Support

3.12.1.1 NodeMCU ESP32 Microcontroller

The ESP32 is the name of the microcontroller developed by Reactive Systems. This provides Wi-Fi (and, in some versions, dual-mode Bluetooth) access for embedded devices. The producer commonly applies to components and development boards that use this chip as "ESP32," despite the fact that it is essentially just a chip. A CPU core, faster Wi-Fi, Bluetooth 4.0 (BLE), touch sensibility ports, Hall Effect sensors, and a temperature sensor are just a few of the extra features that are included with the ESP32. More GPIO pins are available on the ESP32. Figure 33 show the NodeMCU ESP32.

Figure 33 NodeMCU ESP32

3.12.1.2 WT588D Voice Sound Audio Player Module

Figure 34 show the WT588D which is a solid voice chip microcontroller that can be modified. This programmed is simple to use, supports online download, and even while the chip is turned on, data may be downloaded to the relevant SPI flash and then reset the circuit, updating the control mode. This chip contains mp3 player functions, such as play, pause, previous, next, volume up, volume down, in the Mp3 control mode. The trigger mode is adjustable in the key control mode, with on/off 15 different trigger modes and a maximum of10 key trigger output.

Figure 34 Voice Sound Audio Module

3.12.1.3 ISD1820 Sound Recording Module

The figure 35 show the ISD1820 Sound Recording Module records up to 10 seconds of audio and then play it back using MCU or manual control. You may record one message of up to 10 seconds in length using the built-in microphone. With a resistor modification, this may be increased to 20 seconds. The message is non- volatile and may be rerecorded up to 100,000 times. An 8-ohm speaker can be controlled by a tiny built-in audio amplifier. The module's primary characteristics are illustrated below. Record, PlayE, and PlayL functions can be accessed by pressing a button or entering a logic HIGH on the header pin with the same designation.

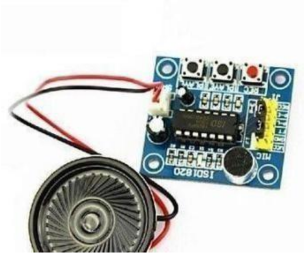


Figure 35 Sound Recording Module

3.12.1.4 ESP32 -Cam WIFI

Figure 36 show the ESP32-CAM which is a 2740mm development board module. It may be combined with an ESP32 module and a camera to form a camera system. The ESP32-CAM is a compact, low-power camera module based on the ESP32. It includes an OV2640 camera and an inbuilt TF card slot. The ESP32-CAM may be utilized in a variety of sophisticated IoT applications, including wireless video monitoring, Wi-Fi picture upload, QR identification, and so on.

Figure 36 ESP32-Cam WIFI

3.12.1.5 HX711 Load Cell Straight Bar

In weighing scales and industrial control applications, the HX711 is a precision 24-bit analogue-to-digital converter (ADC) that is intended to connect directly with a bridge sensor. Its intended use is to amplify cell signals and send them towards another microcontroller. The HX711 electronic scale module's basic principle of operation is to use a conversion circuit to convert observed differences in impedance numbers change into electrical output. The module is linked to the computer system using TTL 232. Load Cell straight Bar is shown is figure 37.

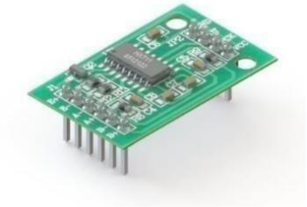


Figure 37 Load Cell Straight Bar

3.12.1.6 Straight Bar Weight Sensor

An equipment that detects force and load is a weight sensor. One type of transducer is a weight sensor, sometimes known as a weight transducer. It transforms a mechanical force—like a load, weight, tension, compression, or pressure—into another physical variable, in this case, an electrical output signal that can be calibrated, converted, and monitored. The majority of weigh sensors used integrated test equipment to measure weight and force while detecting weight. Figure 38 show the weight sensor that been used in this project.

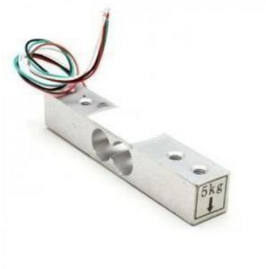


Figure 38 Weight Sensor

Table 14 Hardware requirements

Hardware	Specification	Purpose
Laptop	• Brand: Dell • Processor: Intel Core i7 • Windows 10 Pro • RAM: 8GB • System Type: 64-bit operating system, x64-based processor	A machine required to create the system, it works as the medium for code development, software installation, and system designing.
NodeMCU	• ESP32 • 2x19pin extension headers • Wi-Fi protocol: IEEE 802.11 b/g/n Bluetooth: Bluetooth 4.2	It functions as a microcontroller between the sensors and the internet.
Sound Audio Player Module	• WT588D • Memory: 8M • 16pin, 28pin COB package	Using a 'voice chip,' user may add personal sound samples in the form of Audio data to the project.
Voice Recording	• ISD1820	A small Voice Recorder and Playback module that can do the multi-segment recording.
Camera	• ESP32 • Wi-Fi protocol: IEEE 802.11 b/g/n • Bluetooth: Bluetooth 4.2	It is function for wireless video monitoring

Torque metal gear dc motor	• 6V • 266 RPM	
Load Cell Straight Bar	• HX711	Amplifying signals from cells and reporting them to another microcontroller.
Weight Sensor		Weight sensors are a device used to measure force and load.
Breadboard		It is a construction base for electronic circuit prototype.

3.12.2 Software Support

3.12.2.1 Arduino IDE

Figure 39 show the Arduino Software (IDE), also referred to as the Arduino Integrated Development Environment (IDE), has a programming environment, organizations utilize, a text terminal, a toolbar with buttons for typical operational tasks, and a variety of menus. It communicates with the Arduino hardware and transfers code to it. The system was created using open-source Java software. Linux, Mac OS X, and Windows are among the operating systems that the Arduino IDE is supported with. The C or C++ programming languages are supported by the Arduino IDE with a specific framework.

Figure 39 Arduino Logo

3.12.2.2 Visual Studio Code

In a complete unit, Visual Studio Code (VS Code) that has been shown in figure 40, combines the convenience of use of an operating system with the functionalities that developers are using for their basic modify process. It has a complete and accurate code completion system, modest checking, a solid connection design, and least engagement with existing tools. It is accessible and open source.

Figure 40 Visual Studio Code Logo

3.12.2.3 IONIC Framework

Ionic is a thorough open-source SDK for developing hybrid mobile apps that was created in 2013 by Max Lynch, Ben Sperry, and Adam Bradley of Drifty Co. The initial version was created using AngularJS and Apache Cordova and released in 2013. However, the most recent version was completely rewritten as a collection of Web Components, enabling the user to choose any UI framework, including Angular, React, or Vue.js. Ionic feature's extensible base themes, typography, interaction paradigms, and mobile components. Android 4.4 and above is compatible with Ionic. iOS 10 and later are compatible with Ionic. Ionic 2 supports the Universal Windows Platform for the development of Windows 10 apps. The Ionic Framework, which is based on Angular.js, supports BlackBerry 10 applications. Figure 41 show the ionic framework logo.

Figure 41 Ionic Framework logo

3.12.2.4 Firebase

A framework for developing apps called Firebase, supported by Google, enables developers to produce iOS, Android, and Web apps. Firebase provides tools for tracking statistics, analyzing reports, fixing app issues, and running marketing and product experiments. Firebase Authentication facilitates the development of safe authentication systems for programmers while enhancing user sign-in and onboarding. Data may be stored and updated in real time between users due to the cloud-hosted NoSQL Firebase Realtime Database. The data is available even when an app is offline because it is linked in instantaneously across all clients. Figure 42 show the firebase logo.

Figure 42 Firebase logo

3.12.2.5 Other Software

Table below shows the list of other software used in this project.

Table 15 Other Software

Software	Purpose
Microsoft 365 Word	To write the report of the project.
Draw.io	To draw diagrams that relevant to this project.
Mockflow	To design the system interfaces.

3.13 IMPLEMENTATION

The project starts with requirements analysis. The previous automatic pet feeder will be analyzed and compared. The analysis helps identify the shortcomings of the existing system and improve the qualities of the new system. Environmental impacts and information are explored in order to have a better overview, which contributes to system design.

The following step is system design. System design methodologies include context diagrams, use case diagrams, flow charts, and activity diagrams. The technique implemented can demonstrate the system's prototype and simpler design.

After the design process, the implementation process is proceeded. During system implementation, programming languages such as HTML, CSS, and JavaScript will be employed. Ionic and Firebase will be utilized to construct mobile applications and databases.

The important functions of this automatic pet feeder system include maintaining profiles, organizing feed schedules, controlling food weight, monitoring pet, and recording voice. The system provides varied amounts of authority to different users. For example, user does not have admin-owned authority. As a result, the authority varies according on the type of user, such as pet owner or administrator. The administrator is then given permission to manage user accounts.

Finally, data from the machine and sensors such as the weight sensor will be sent to the NodeMCU controller and recorded in the database. The mobile application is linked to the database. As a result, the user will access the data via the mobile platform. Furthermore, the internet features allow the user to access the system from anywhere and at any time to feed and monitor their pet.

3.14 GANTT CHART

The Gantt Chart depicts the project's progress along the way. The project will follow the suggested methodology, which includes the planning phases for requirements, designing phases for system interfaces, developing phases, and cutover phases. The Gantt Chart is available in APPENDIX A.

CHAPTER 4

RESULTS AND DISCUSSION

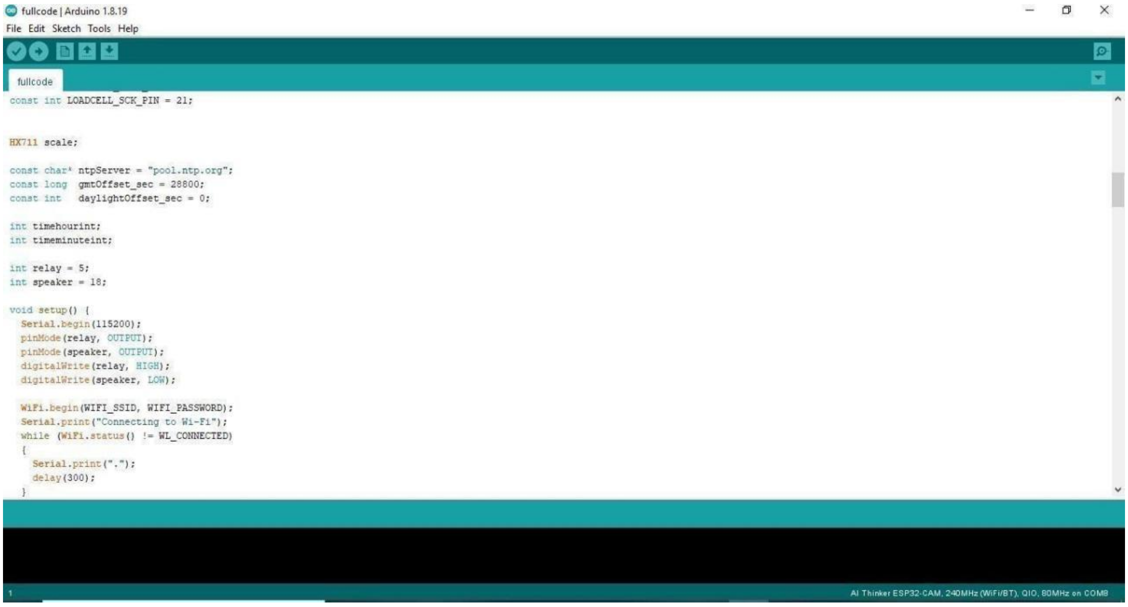
4.1 INTRODUCTION

This chapter focuses on the overall system implementation process, testing, and results that were attained during the project system implementation process. The RAD technique that was covered in the previous chapter was used to implement the system and test it. This chapter will provide a thorough overview of the system implementation process, the results that were gained, and the testing methodology to make sure the project's goals were met. In order to build up the hardware and software for the project system, programming languages like HTML, Typescript, and Arduino were used. This chapter will also outline the project's limitations. The debate and findings from this project's system help plan and steer the development of a future system.

4.1.1 Development Environment

Android Studio and the Arduino IDE are the programmed used to construct this system. The Arduino IDE is a piece of software that enables the authoring of code, uploading of code, and serial monitoring of sensor data. The Arduino programming language was used to design the build code to connect with the ESP32 microcontroller while the sensors were connected, and the serial monitor was employed to track the sensor data. The mobile application to control pet feeders and enable user login was developed in the meantime using the Android Studio.

The ESP32 microcontroller board with WIFI was used in the Automatic Pet Feeder, and the Arduino IDE was used to develop the build code. An output area, a toolbar, a text editor, upload and compilation options, and simple hardware and software make up the free and open-source Arduino electronics platform. The ESP32 microcontroller and the attached sensors are communicated with by the build many codes written in the Arduino IDE. It enables the ESP32 microcontroller, which defines the hardware functions, to receive the written code for uploading. The figure 43 show the example of Arduino IDE interface.


fullcode | Arduino 1.8.19
File Edit Sketch Tools Help
fullcode
AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz on COM8

```
const int LOADCELL_SCK_PIN = 21;


HX711 scale;

const char* ntpServer = "pool.ntp.org";
const long  gmtOffset_sec = 28800;
const int   daylightOffset_sec = 0;

int timehourint;
int timeminuteint;

int relay = 5;
int speaker = 18;

void setup() {
  Serial.begin(115200);
  pinMode(relay, OUTPUT);
  pinMode(speaker, OUTPUT);
  digitalWrite(relay, HIGH);
  digitalWrite(speaker, LOW);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
```

Figure 43 Arduino IDE Interface

The system developers also made extensive use of the Android Studio. It was utilized to create a mobile application for this system that synchronized user input with sensor data from the ESP32 microcontroller. To enable the user to take the appropriate action, the data will be presented to the user in an easily understandable manner. HTML, CSS, and Typescript will be the programming languages used to create this mobile application; they were the best programming languages to write the scripts in.

4.2 AUTOMATIC PET FEEDER APPLICATION SYSTEM

4.2.1 Register Interface

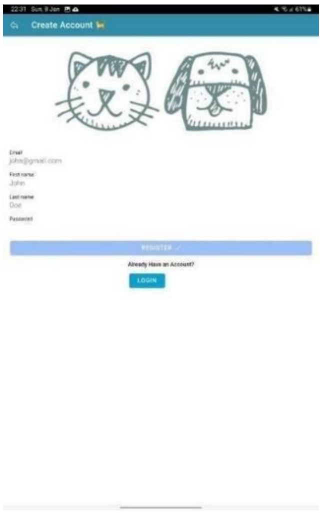


Figure 44 Register Interface

Figure 44 show the user sign-up screen allows the user to register for the automatic pet feeder system. Before logging onto the system, users must first create their own account. Users must provide their Email Address, Password, First Name and Last Name.

4.2.2 Login Interface

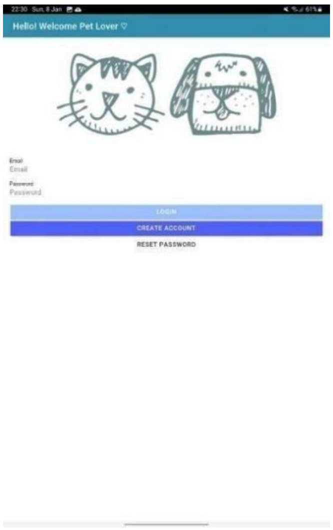
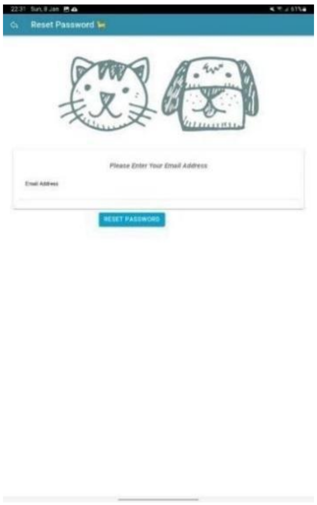
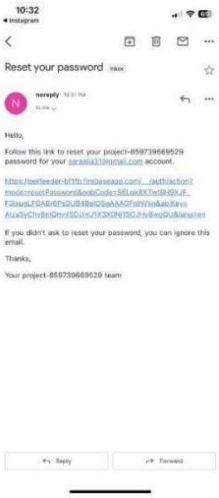


Figure 45 Login Interface

Figure 45 The Login Interface for the User displays an interface that allows the user to log into the Automatic Pet Feeder System. To access the system, users must provide their Username and Password. If the user does not yet have an account, they may sign up by clicking the Register button, and if they forget their password, they can click the Reset Password button.

4.2.3 Reset Password Interface





Reset your password

noreply 10:31 PM

Hello,

Follow this link to reset your project-859739669529 password for your saraalia31@gmail.com account.

https://petfeeder-bf1fb.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=[illegible]&apiKey=[illegible]&lang=en

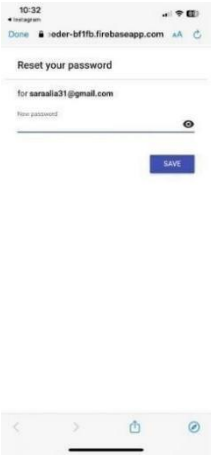
If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your project-859739669529 team

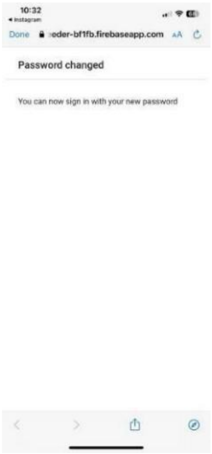
Figure 46 Reset password interface

Figure 47 Receive email to reset password



Reset your password

for saraalia31@gmail.com



Password changed

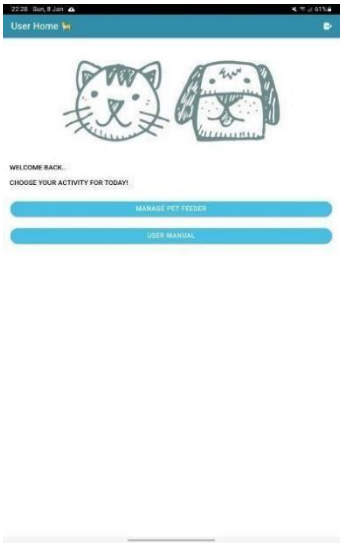
You can now sign in with your new password

Figure 48 Enter new password

Figure 49 Successfully reset password

Figure 46 until 49 show the process to reset the password. Reset Password interface allow the user to enter the email address if the user forgot their password. After user click on the reset password button, the user will receive the email that contain link to enter new password. If user click on the link given, the pop-up message will appear to enter new password. After click on save, the user can use the application by enter new password.

4.2.4 User Home Interface



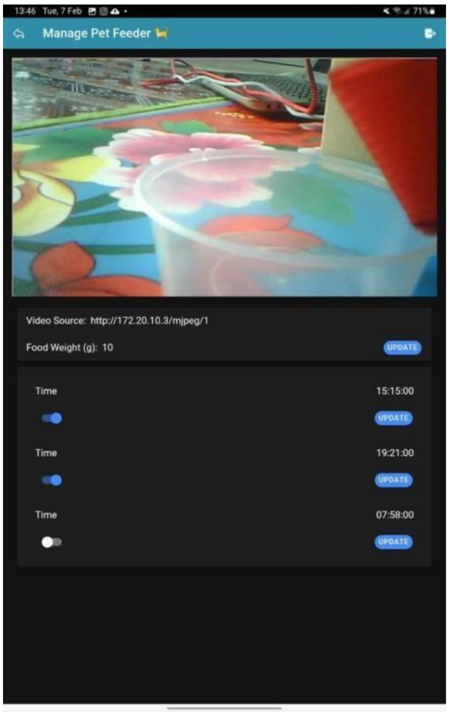
WELCOME BACK..

CHOOSE YOUR ACTIVITY FOR TODAY!

Figure 50 User home interface

Figure 50 show the main page for the user after login to the system. Users may perform any activity by simply pressing any of the accessible buttons to get to the activity page. However, if the user decides to leave this system, the user can do so by clicking the Log Out button at the top right of this page.

4.2.5 Manage Pet Feeder Interface



Video Source: http://172.20.10.3/mjpeg/1

Food Weight (g): 10

Time 15:15:00

Time 19:21:00

Time 07:58:00

Figure 51 Manage Pet Feeder Interface

Figure 51 shows the manage pet feeder interface that allow user to monitor their pet, manage food’s weight and manage feeding schedule. The user can enter the food’s weight in gram and then click on update button to save the data in the database. In addition, user also can select time to feed their pet, user can scroll the time to choose which time their want feed their pet, next click on update therefore the automatic pet feeder machine will dispense the food according to the time and weight that has been entered by user. However, if the user decides to leave this system, the user can do so by clicking the Log Out button at the top right of this page.

4.2.6 User Manual Interface

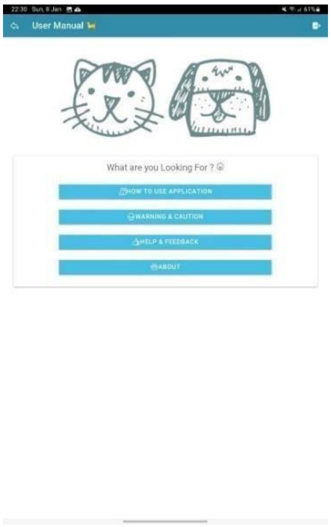


Figure 52 User Manual Interface

Figure 52 show the user manual interface. This page will explain on how to use the Automatic Pet Feeder machine to make user ease to use the machine. However, if the user decides to leave this system, the user can do so by clicking the Log Out button at thetop right of this page.

4.2.7 Log Out Interface

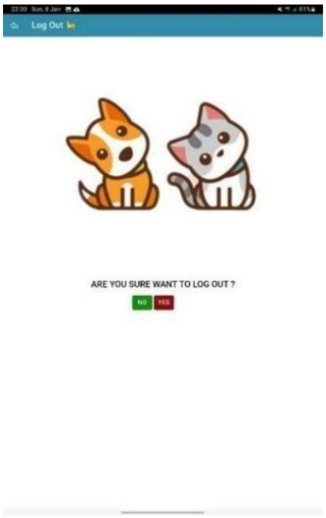


Figure 53 Log Out Interface

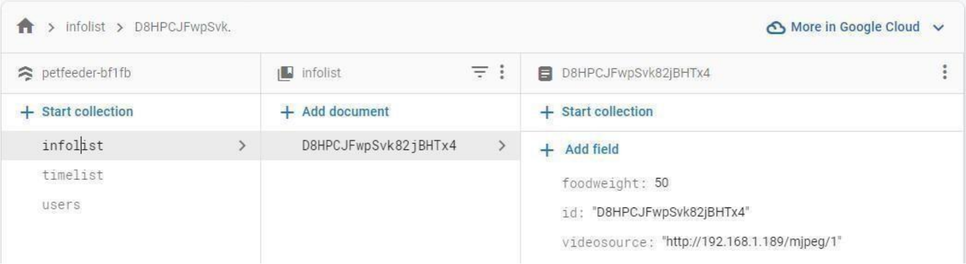
Figure 53 show the log out interface. User will be taken to this page after clicking the logout button. The user will be prompted on this page to confirm their want to log out. If a user wishes to log out, they can click the Yes button. If they don't, they can click the No button, which will take them to the User Home screen.

4.3 IMPLEMENTATION

4.3.1 Database Architecture

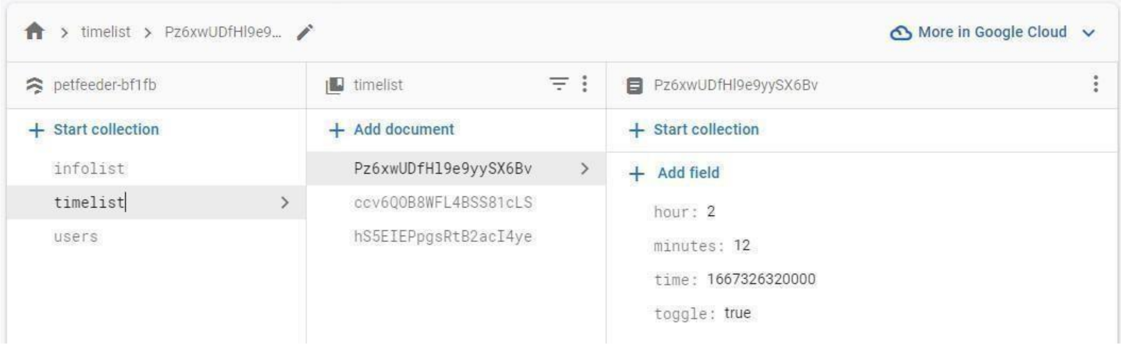
Database that has been used for this Smart Pet Feeder system is Firebase. The tables that been used consists of 3 tables which is infolist, timelist and users. Table infolist is for the weight that has been updated by the user, while table timelist is for user to update the schedule time for Smart Pet Feeder and lastly table users contain the user list.

Infolist table is as figure 54 that contain food weight, id and the video source which is the link for monitor the pets by using the camera esp-32 cam. Next, for the timelist table in figure 55 that contain hour, minute, time, and toggle. The toggle function is to show that the schedule is on or not. Lastly, for the users table in figure 56 is contain email,first name and last name of the user that already register for the system.



foodweight: 50
id: "D8HPCJFwpSvk82jBHTx4"
videosource: "http://192.168.1.189/mjpeg/1"

Figure 54 Table Infolist



infolist
timelist
users

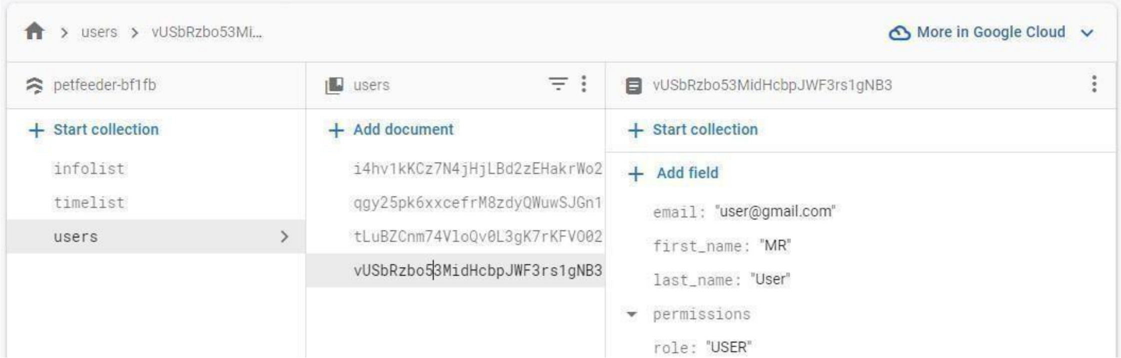
timelist

Pz6xwUDfH19e9yySX6Bv
ccv6QOB8WFL4BSS81cLS
hS5EIEPpgsRtB2acI4ye

Pz6xwUDfHl9e9yySX6Bv

hour: 2
minutes: 12
time: 1667326320000
toggle: true

Figure 55 Table timelist



infolist
timelist
users

users

i4hv1kKCz7N4jHjLBd2zEHakrWo2
qgy25pk6xxcefrM8zdyQWuwSJGn1
tLuBZCnm74V1oQv0L3gK7rKFV002
vUSbRzbo53MidHcbpJWF3rs1gNB3

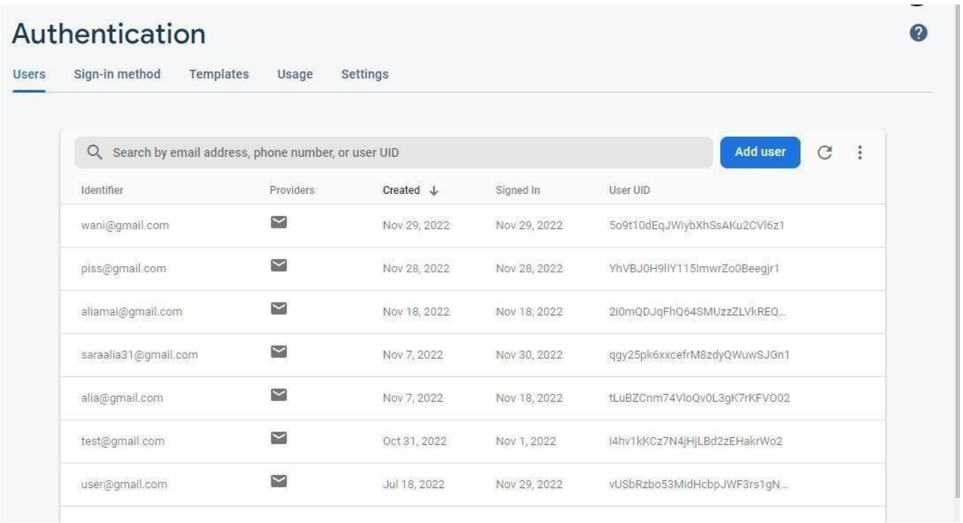
vUSbRzbo53MidHcbpJWF3rs1gNB3

email: "user@gmail.com"
first_name: "MR"
last_name: "User"
permissions
role: "USER"

Figure 56 Table users

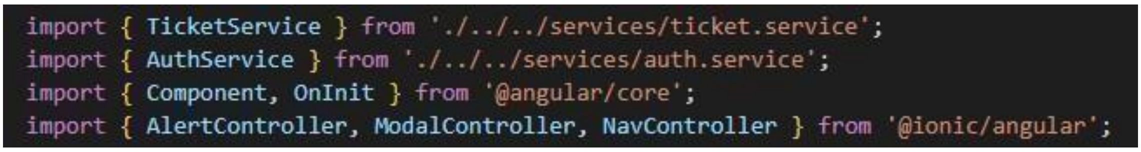
4.3.1.1 Authentication

During registration, this database was used to store the email and password in encrypted form. When a user is formed, it also generates a user id and displays the most recent timestamp of the user's application sign-in. Figure 57 show the authentication in the database. Figure 58 and figure 59 shows the authentication database code in android studio and database connection code in android studio.



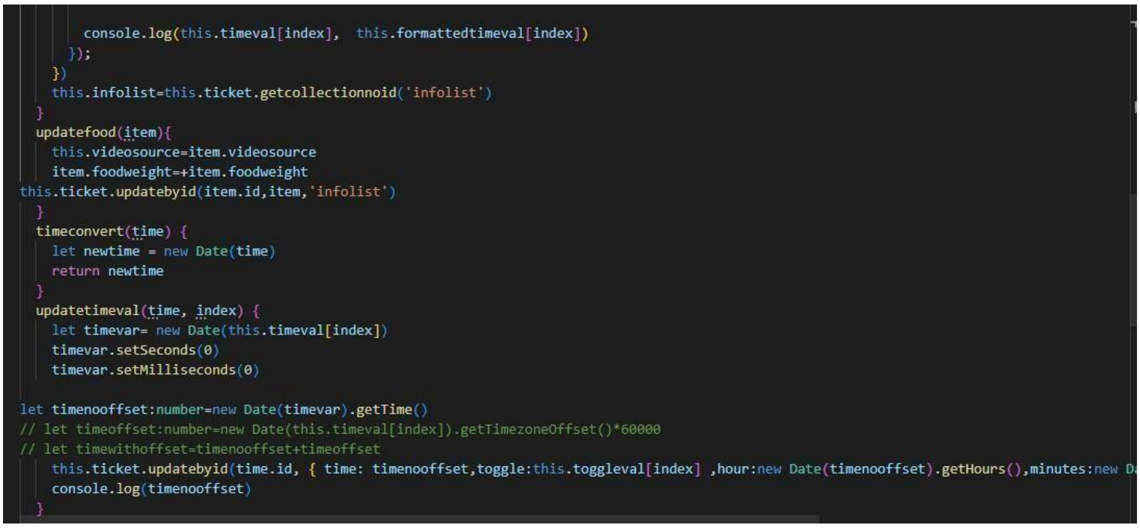
Identifier	Providers	Created ↓	Signed In	User UID
wani@gmail.com	✉	Nov 29, 2022	Nov 29, 2022	5o9t10dEqJWiybXhSsAKu2CVl6z1
piss@gmail.com	✉	Nov 28, 2022	Nov 28, 2022	YhVBJ0H9lIY115ImwrZo0Beegjr1
aliamai@gmail.com	✉	Nov 18, 2022	Nov 18, 2022	2i0mQDJqFhQ64SMUzzZLVkREQ...
saraalia31@gmail.com	✉	Nov 7, 2022	Nov 30, 2022	qgy25pk6xxcefrM8zdyQWuwSJGn1
alia@gmail.com	✉	Nov 7, 2022	Nov 18, 2022	tLuBZCnm74VloQv0L3gK7rKFVO02
test@gmail.com	✉	Oct 31, 2022	Nov 1, 2022	i4hv1kKCz7N4jHjLBd2zEHakrWo2
user@gmail.com	✉	Jul 18, 2022	Nov 29, 2022	vUSbRzbo53MidHcbpJWF3rs1gN...

Figure 57 Authentication Database



```
import { TicketService } from './../../services/ticket.service';
import { AuthService } from './../../services/auth.service';
import { Component, OnInit } from '@angular/core';
import { AlertController, ModalController, NavController } from '@ionic/angular';
```

Figure 58 Authentication Database code in Android Studio

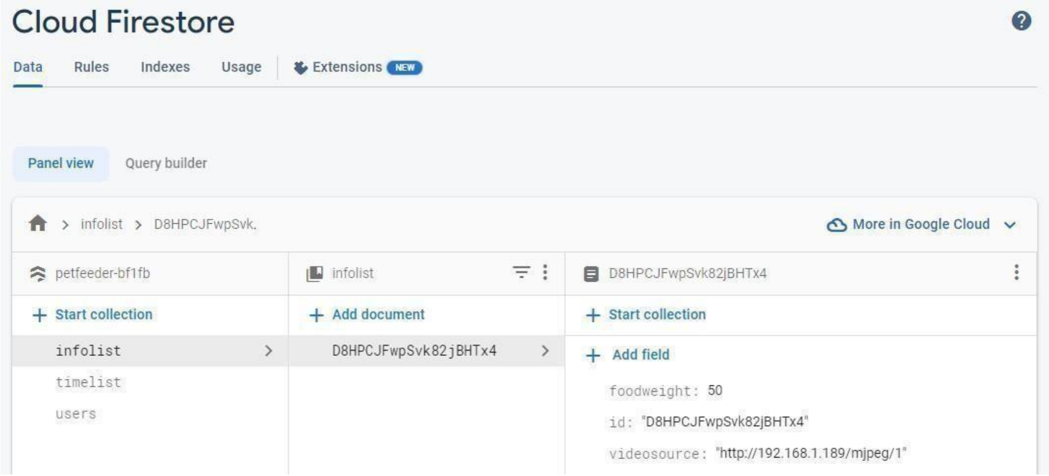


```
        console.log(this.timeval[index],  this.formattedtimeval[index])
      });
    })
    this.infolist=this.ticket.getcollectionnoid('infolist')
  }
  updatefood(item){
    this.videosource=item.videosource
    item.foodweight=+item.foodweight
this.ticket.updatebyid(item.id,item,'infolist')
  }
  timeconvert(time) {
    let newtime = new Date(time)
    return newtime
  }
  updatetimeval(time, index) {
    let timevar= new Date(this.timeval[index])
    timevar.setSeconds(0)
    timevar.setMilliseconds(0)

let timenooffset:number=new Date(timevar).getTime()
// let timeoffset:number=new Date(this.timeval[index]).getTimezoneOffset()*60000
// let timewithoffset=timenooffset+timeoffset
    this.ticket.updatebyid(time.id, { time: timenooffset,toggle:this.toggleval[index] ,hour:new Date(timenooffset).getHours(),minutes:new Da
    console.log(timenooffset)
  }
```

Figure 59 Database Connection Code in Android Studio

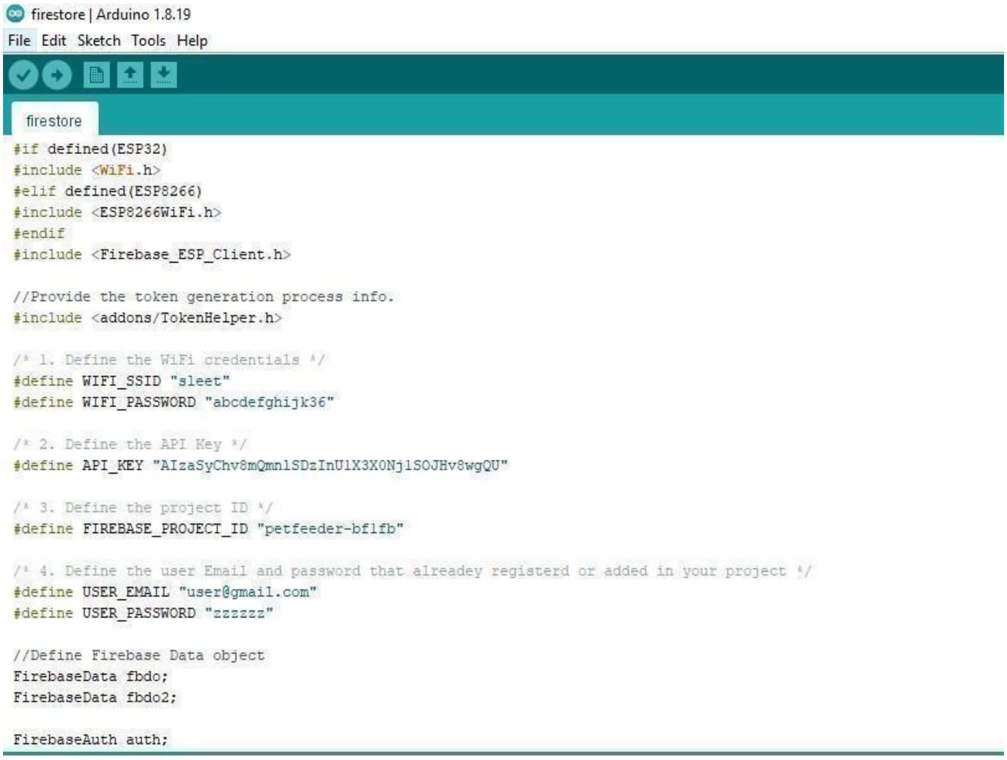
4.3.1.2 Real-time Database

The main purpose of the real-time database in this system was to enable theESP32 to push fresh reading into the database without expanding its capacity. Figure 60 show the real-time database in the firebase. Figure 61 show that database code in Arduino.



foodweight: 50
id: "D8HPCJFwpSvk82jBHTx4"
videosource: "http://192.168.1.189/mjpeg/1"

Figure 60 Real-time database

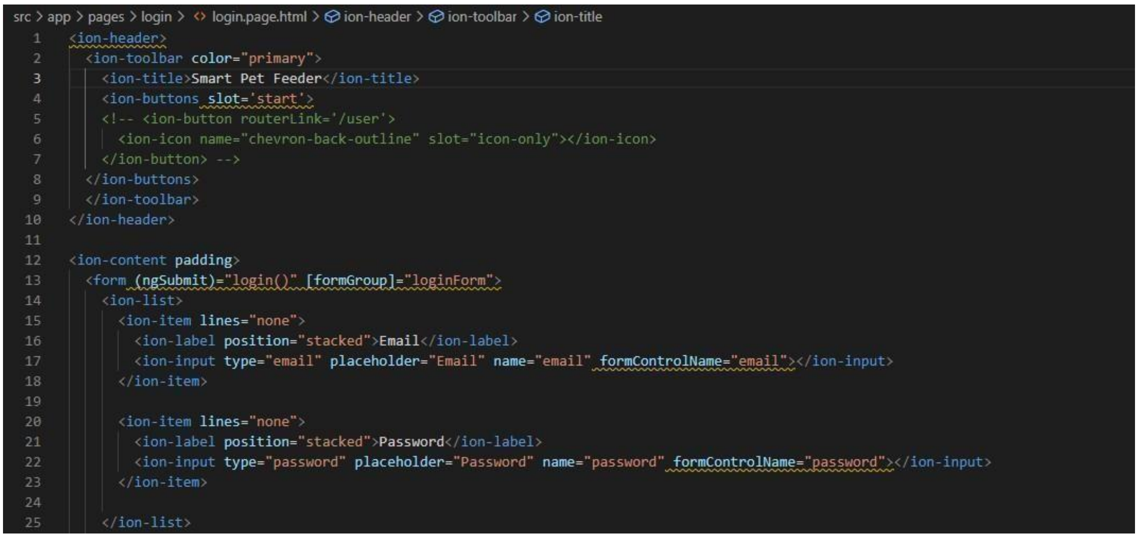


```
#if defined(ESP32)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>

//Provide the token generation process info.
#include <addons/TokenHelper.h>

/* 1. Define the WiFi credentials */
#define WIFI_SSID "sleet"
#define WIFI_PASSWORD "abcdefghijk36"

/* 2. Define the API Key */
#define API_KEY "AIzaSyChv8mQmnlSDzInU1X3X0Nj1SOJHv8wgQU"

/* 3. Define the project ID */
#define FIREBASE_PROJECT_ID "petfeeder-bf1fb"

/* 4. Define the user Email and password that alreadey registerd or added in your project */
#define USER_EMAIL "user@gmail.com"
#define USER_PASSWORD "zzzzzz"

//Define Firebase Data object
FirebaseData fbdo;
FirebaseData fbdo2;

FirebaseAuth auth;
```

Figure 61 Database code in Arduino

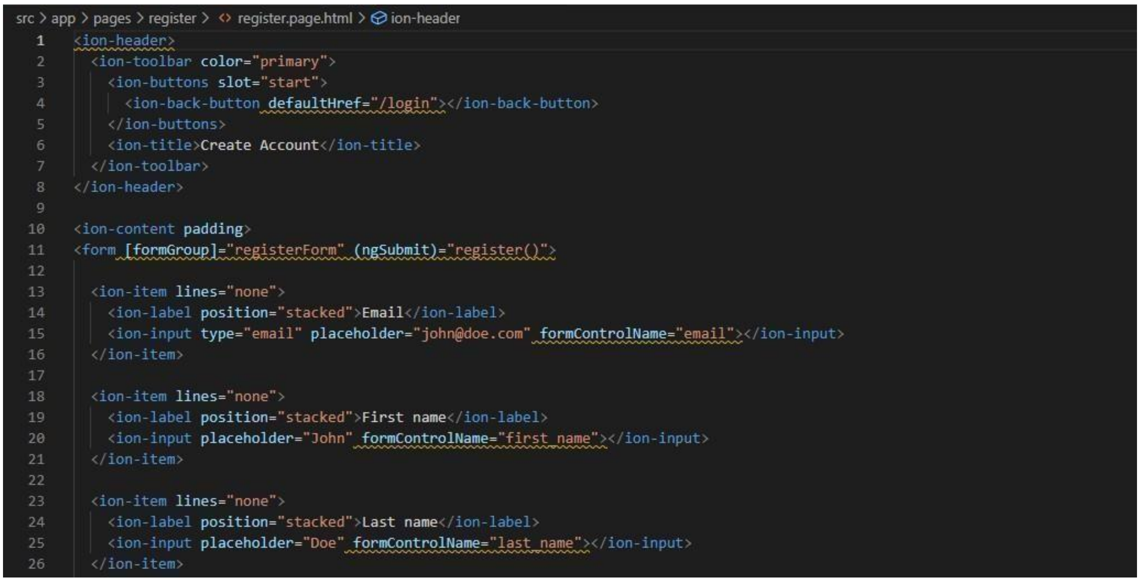
4.3.2 Automatic Pet Feeder System Source Code

Visual Studio Code was utilized as a platform to develop the code for the Automatic Pet Feeder application, which is the software that was used to construct this system. HTML and Typescript were employed as the primary languages in the development of this system. Figure 62 until figure 67 shows the source code for the mobile application.


src > app > pages > login > login.page.html > ion-header > ion-toolbar > ion-title

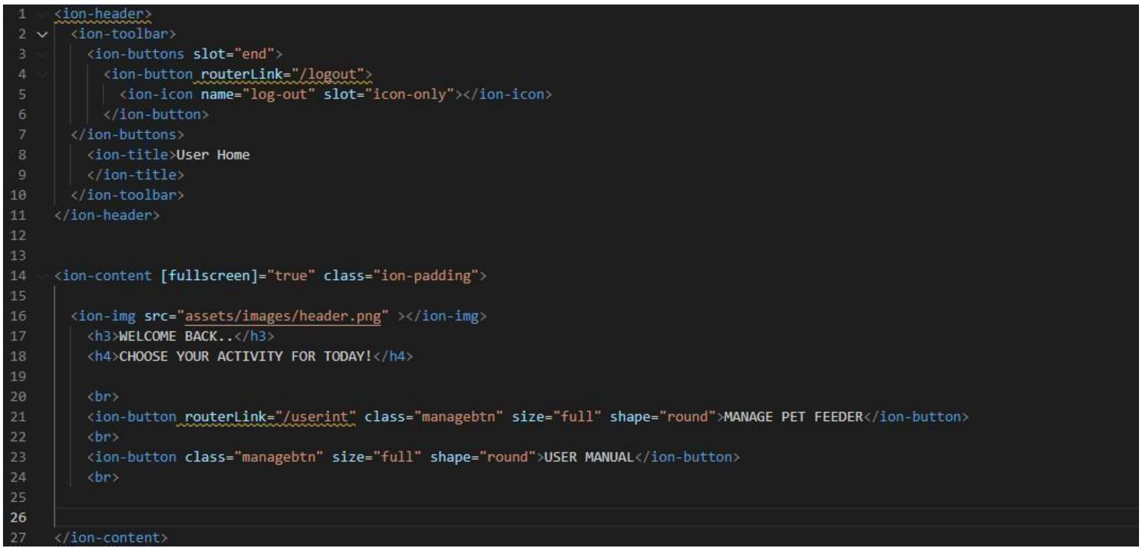
```html
<ion-header>
  <ion-toolbar color="primary">
    <ion-title>Smart Pet Feeder</ion-title>
    <ion-buttons slot='start'>
    <!-- <ion-button routerLink='/user'>
      <ion-icon name="chevron-back-outline" slot="icon-only"></ion-icon>
    </ion-button> -->
  </ion-buttons>
  </ion-toolbar>
</ion-header>

<ion-content padding>
  <form (ngSubmit)="login()" [formGroup]="loginForm">
    <ion-list>
      <ion-item lines="none">
        <ion-label position="stacked">Email</ion-label>
        <ion-input type="email" placeholder="Email" name="email" formControlName="email"></ion-input>
      </ion-item>

      <ion-item lines="none">
        <ion-label position="stacked">Password</ion-label>
        <ion-input type="password" placeholder="Password" name="password" formControlName="password"></ion-input>
      </ion-item>

    </ion-list>
```

Figure 62 Source Code for Login Page


src > app > pages > register > register.page.html > ion-header

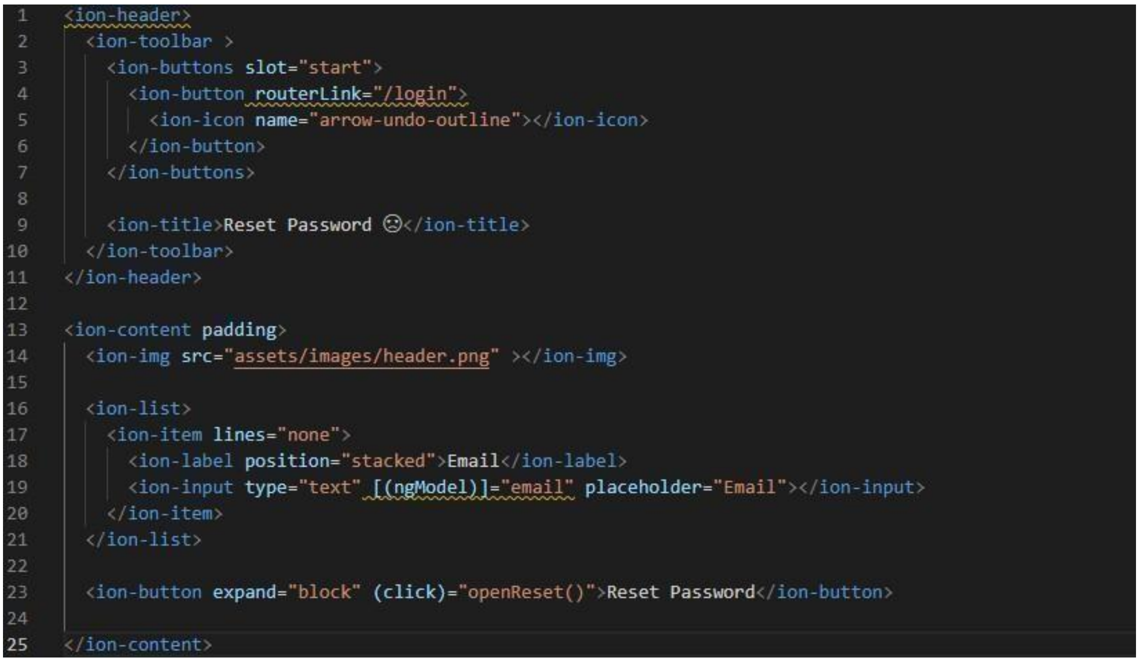
```html
<ion-header>
  <ion-toolbar color="primary">
    <ion-buttons slot="start">
      <ion-back-button defaultHref="/login"></ion-back-button>
    </ion-buttons>
    <ion-title>Create Account</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content padding>
<form [formGroup]="registerForm" (ngSubmit)="register()">

  <ion-item lines="none">
    <ion-label position="stacked">Email</ion-label>
    <ion-input type="email" placeholder="john@doe.com" formControlName="email"></ion-input>
  </ion-item>

  <ion-item lines="none">
    <ion-label position="stacked">First name</ion-label>
    <ion-input placeholder="John" formControlName="first_name"></ion-input>
  </ion-item>

  <ion-item lines="none">
    <ion-label position="stacked">Last name</ion-label>
    <ion-input placeholder="Doe" formControlName="last_name"></ion-input>
  </ion-item>
```

Figure 63 Source Code for Register Page



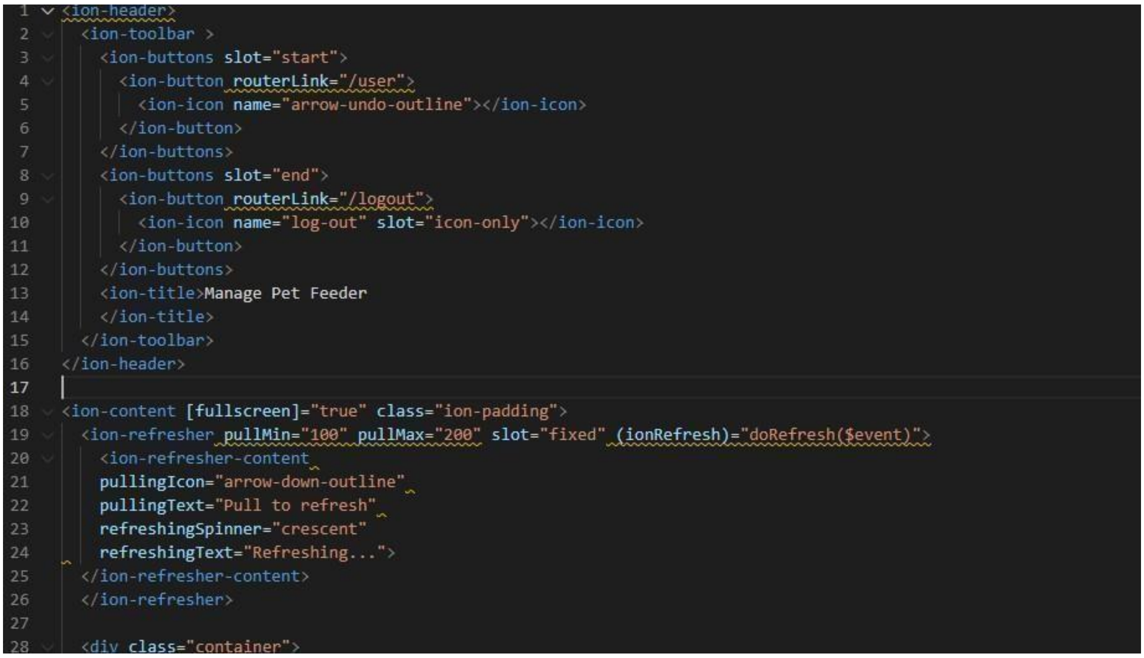
```
<ion-header>
  <ion-toolbar>
    <ion-buttons slot="end">
      <ion-button routerLink="/logout">
        <ion-icon name="log-out" slot="icon-only"></ion-icon>
      </ion-button>
  </ion-buttons>
    <ion-title>User Home
    </ion-title>
  </ion-toolbar>
</ion-header>


<ion-content [fullscreen]="true" class="ion-padding">

  <ion-img src="assets/images/header.png" ></ion-img>
    <h3>WELCOME BACK..</h3>
    <h4>CHOOSE YOUR ACTIVITY FOR TODAY!</h4>

    <br>
    <ion-button routerLink="/userint" class="managebtn" size="full" shape="round">MANAGE PET FEEDER</ion-button>
    <br>
    <ion-button class="managebtn" size="full" shape="round">USER MANUAL</ion-button>
    <br>


</ion-content>
```

Figure 64 Source Code for User Home Page



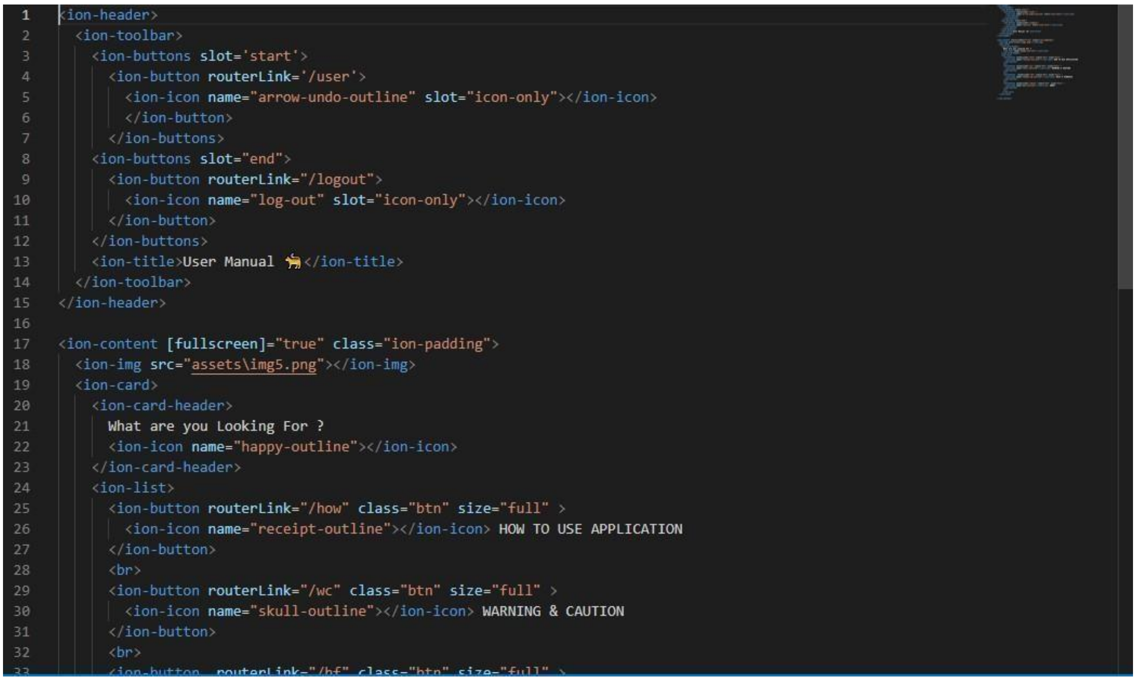
```
<ion-header>
  <ion-toolbar >
    <ion-buttons slot="start">
      <ion-button routerLink="/login">
        <ion-icon name="arrow-undo-outline"></ion-icon>
      </ion-button>
    </ion-buttons>

    <ion-title>Reset Password ☹</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content padding>
  <ion-img src="assets/images/header.png" ></ion-img>

  <ion-list>
    <ion-item lines="none">
      <ion-label position="stacked">Email</ion-label>
      <ion-input type="text" [(ngModel)]="email" placeholder="Email"></ion-input>
    </ion-item>
  </ion-list>

  <ion-button expand="block" (click)="openReset()">Reset Password</ion-button>

</ion-content>
```

Figure 65 Source Code for Reset Password



```
<ion-header>
  <ion-toolbar >
    <ion-buttons slot="start">
      <ion-button routerLink="/user">
        <ion-icon name="arrow-undo-outline"></ion-icon>
      </ion-button>
    </ion-buttons>
    <ion-buttons slot="end">
      <ion-button routerLink="/logout">
        <ion-icon name="log-out" slot="icon-only"></ion-icon>
      </ion-button>
    </ion-buttons>
    <ion-title>Manage Pet Feeder
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true" class="ion-padding">
  <ion-refresher pullMin="100" pullMax="200" slot="fixed" (ionRefresh)="doRefresh($event)">
    <ion-refresher-content
    pullingIcon="arrow-down-outline"
    pullingText="Pull to refresh"
    refreshingSpinner="crescent"
    refreshingText="Refreshing...">
  </ion-refresher-content>
  </ion-refresher>

  <div class="container">
```

Figure 66 Source Code for Manage Pet Feeder



```
<ion-header>
  <ion-toolbar>
    <ion-buttons slot='start'>
      <ion-button routerLink='/user'>
        <ion-icon name="arrow-undo-outline" slot="icon-only"></ion-icon>
        </ion-button>
      </ion-buttons>
    <ion-buttons slot="end">
      <ion-button routerLink="/logout">
        <ion-icon name="log-out" slot="icon-only"></ion-icon>
      </ion-button>
    </ion-buttons>
    <ion-title>User Manual 🐈</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true" class="ion-padding">
  <ion-img src="assets\img5.png"></ion-img>
  <ion-card>
    <ion-card-header>
      What are you Looking For ?
      <ion-icon name="happy-outline"></ion-icon>
    </ion-card-header>
    <ion-list>
      <ion-button routerLink="/how" class="btn" size="full" >
        <ion-icon name="receipt-outline"></ion-icon> HOW TO USE APPLICATION
      </ion-button>
      <br>
      <ion-button routerLink="/wc" class="btn" size="full" >
        <ion-icon name="skull-outline"></ion-icon> WARNING & CAUTION
      </ion-button>
      <br>
      <ion-button  routerLink="/hf" class="btn" size="full" >
```

Figure 67 Source Code for User Manual Page

4.3.3 Hardware Implementation

A major piece of hardware in this system is the microcontroller. It performs the role of a controller for an integrated circuit that is intended to perform a function. This system uses the Arduino ESP32 microcontroller board because it has 34 digital input/output pins and other features that make it ideal for connecting sensors. The sensor is weight sensor. The Wi-Fi module incorporated into this microcontroller is utilized to export and import sensor data into the Firebase database. Afterwards, hardware components such a breadboard, male to male jumper wires, and male to female jumper wires are used to link the sensors to the Arduino microcontroller board. First, the equipment needed to be configured. The breadboard ports were wired with male-to-male jumper wires. Then, for the sensors and components to interact with the microcontroller, they are all linked to the breadboard. Using a USB, A connection to the computer, the code was uploaded to the ESP32, and the serial monitor was utilized for debugging. Figure 68 and figure 69 show the automatic pet feeder prototype.

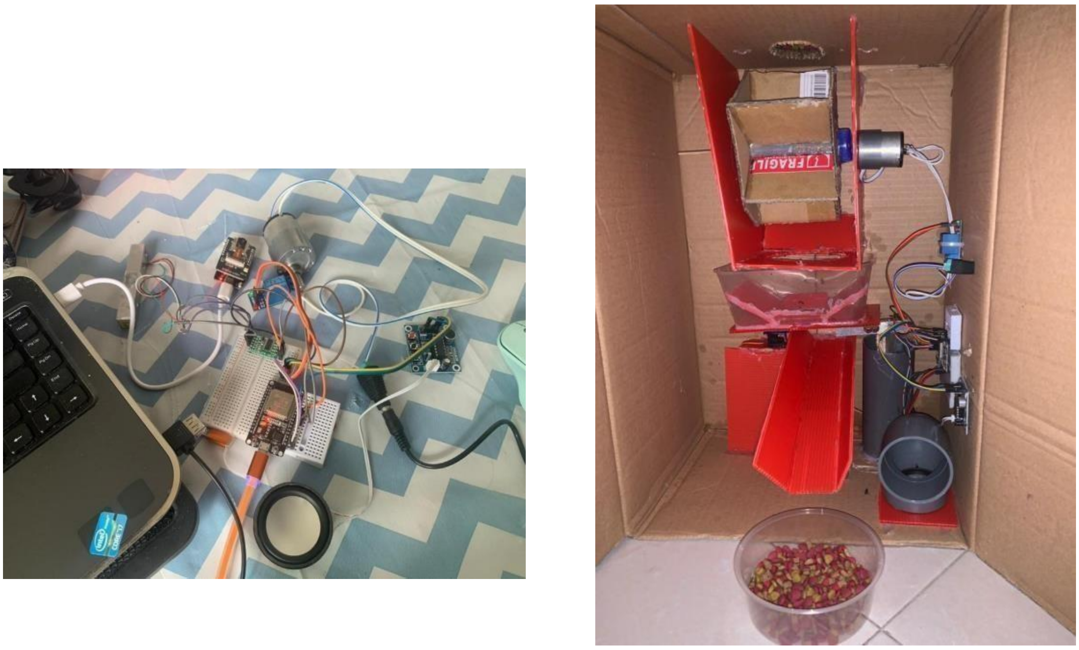


Figure 68 Automatic Pet Feeder Prototype

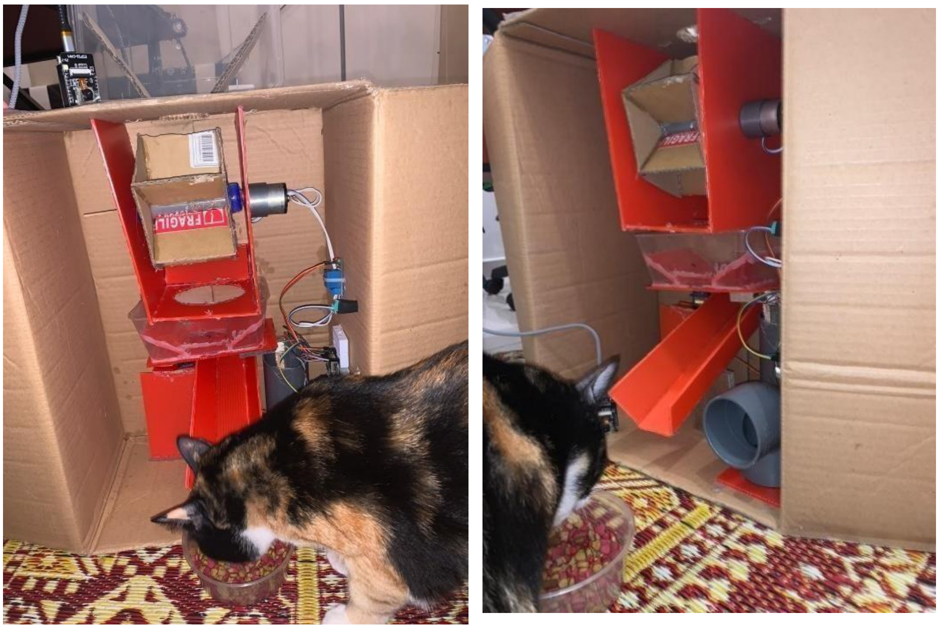
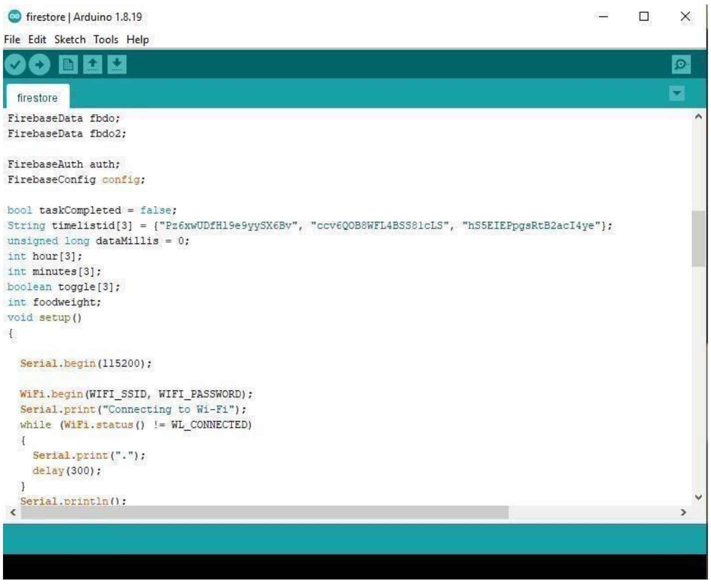


Figure 69 Automatic Pet Feeder Prototype

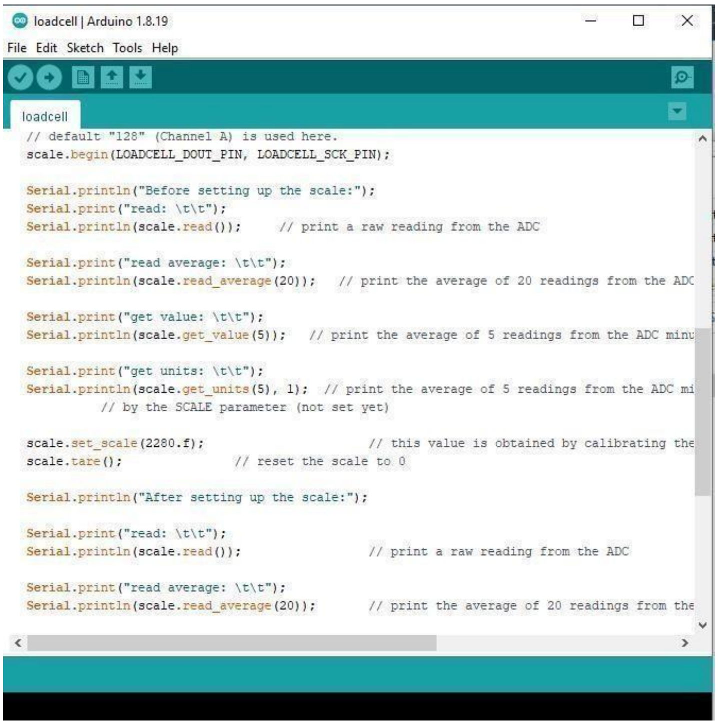
4.3.4 Hardware Source Code

Figure 70 until figure 73 show the source code for each hardware.



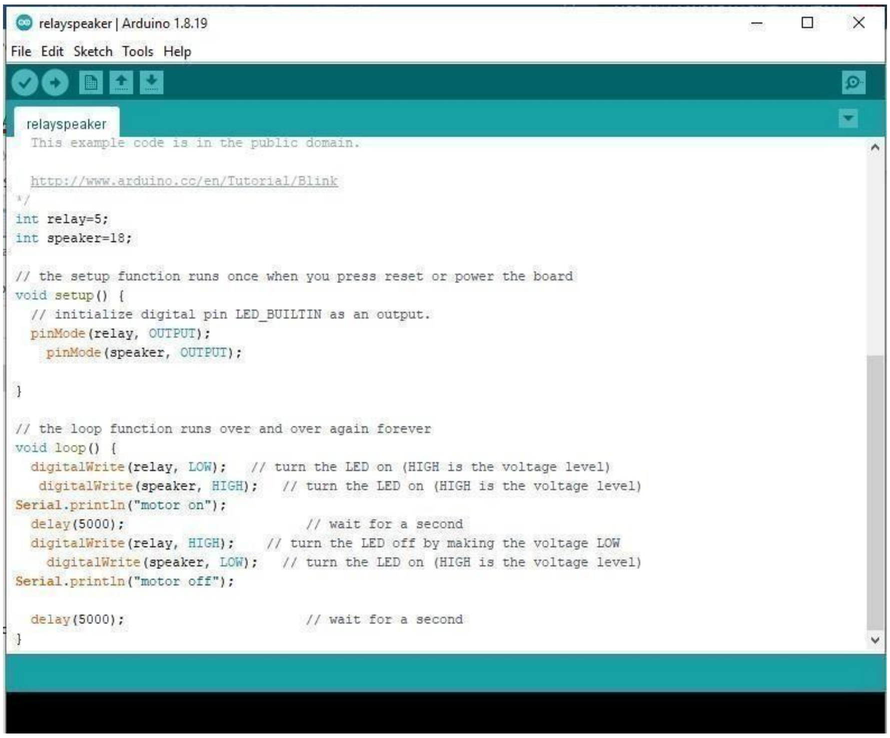
```
FirebaseData fbdo;
FirebaseData fbdo2;

FirebaseAuth auth;
FirebaseConfig config;

bool taskCompleted = false;
String timelistid[3] = {"Pz6xwUDfHl9e9yySX6Bv", "ccv6QOB8WFL4BSS8lcLS", "hS5EIEPpgsRtB2acI4ye"};
unsigned long dataMillis = 0;
int hour[3];
int minutes[3];
boolean toggle[3];
int foodweight;
void setup()
{

  Serial.begin(115200);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
```

Figure 70 Database Code



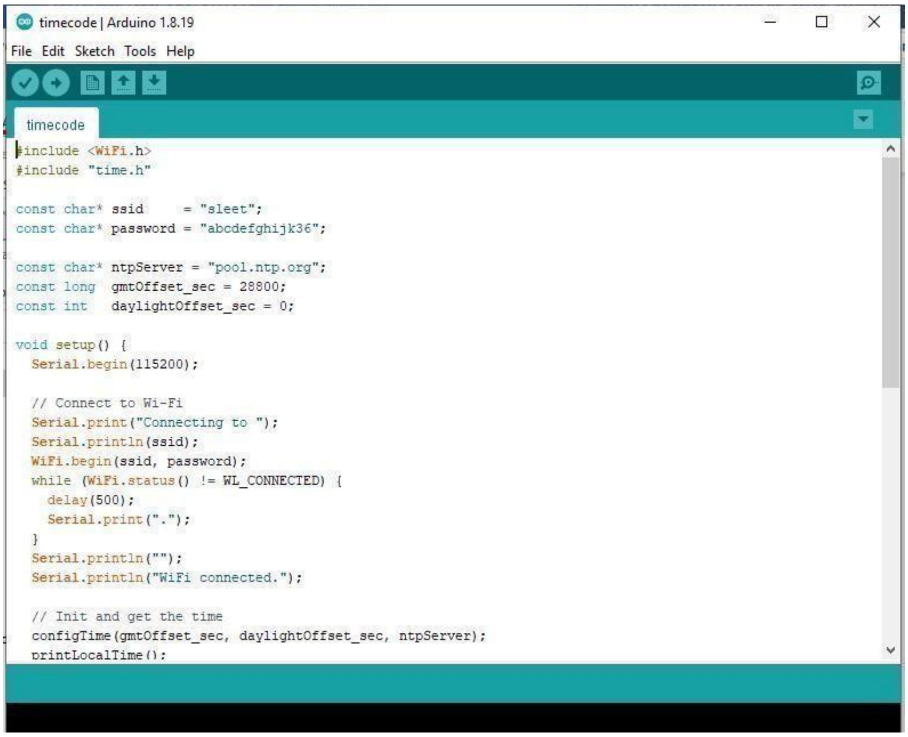
```
// default "128" (Channel A) is used here.
scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);

Serial.println("Before setting up the scale:");
Serial.print("read: \t\t");
Serial.println(scale.read());     // print a raw reading from the ADC

Serial.print("read average: \t\t");
Serial.println(scale.read_average(20));   // print the average of 20 readings from the ADC

Serial.print("get value: \t\t");
Serial.println(scale.get_value(5));   // print the average of 5 readings from the ADC minu

Serial.print("get units: \t\t");
Serial.println(scale.get_units(5), 1);  // print the average of 5 readings from the ADC mi
          // by the SCALE parameter (not set yet)

scale.set_scale(2280.f);                      // this value is obtained by calibrating the
scale.tare();               // reset the scale to 0

Serial.println("After setting up the scale:");

Serial.print("read: \t\t");
Serial.println(scale.read());                 // print a raw reading from the ADC

Serial.print("read average: \t\t");
Serial.println(scale.read_average(20));       // print the average of 20 readings from the
```

Figure 71 Manage Weight Code



```
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/
int relay=5;
int speaker=18;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(relay, OUTPUT);
    pinMode(speaker, OUTPUT);

}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(relay, LOW);   // turn the LED on (HIGH is the voltage level)
   digitalWrite(speaker, HIGH);   // turn the LED on (HIGH is the voltage level)
Serial.println("motor on");
  delay(5000);                       // wait for a second
  digitalWrite(relay, HIGH);    // turn the LED off by making the voltage LOW
    digitalWrite(speaker, LOW);   // turn the LED on (HIGH is the voltage level)
Serial.println("motor off");

  delay(5000);                       // wait for a second
}
```

Figure 72 Speaker Code



```
#include <WiFi.h>
#include "time.h"

const char* ssid     = "sleet";
const char* password = "abcdefghijk36";

const char* ntpServer = "pool.ntp.org";
const long  gmtOffset_sec = 28800;
const int   daylightOffset_sec = 0;

void setup() {
  Serial.begin(115200);

  // Connect to Wi-Fi
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected.");

  // Init and get the time
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  printLocalTime();
```

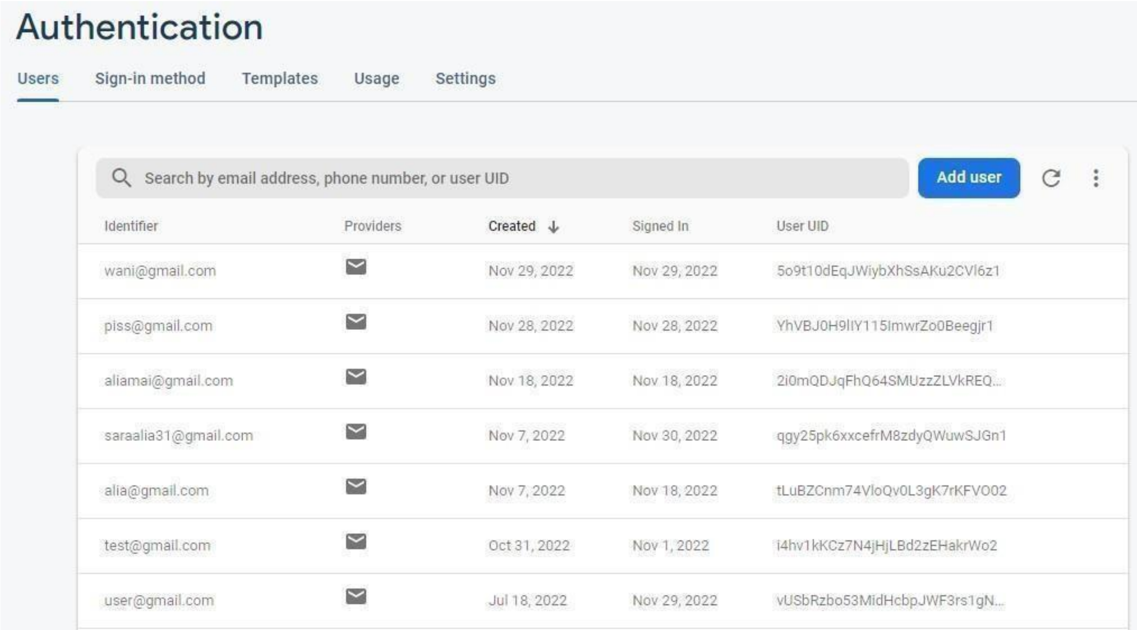
Figure 73 Manage Schedule Code

4.4 TESTING AND DISCUSSION

To verify that the authentication function, database connection, sensors, and mobile applications are on track to fulfil the goal of this system, testing and discussion include both the hardware and the software. The hardware prototype, the mobile application, and an internet connection are necessary tools and materials for the testing to be done. For comprehensive testing, the system underwent the User Acceptance Test.

4.4.1 Automatic Pet Feeder Authentication

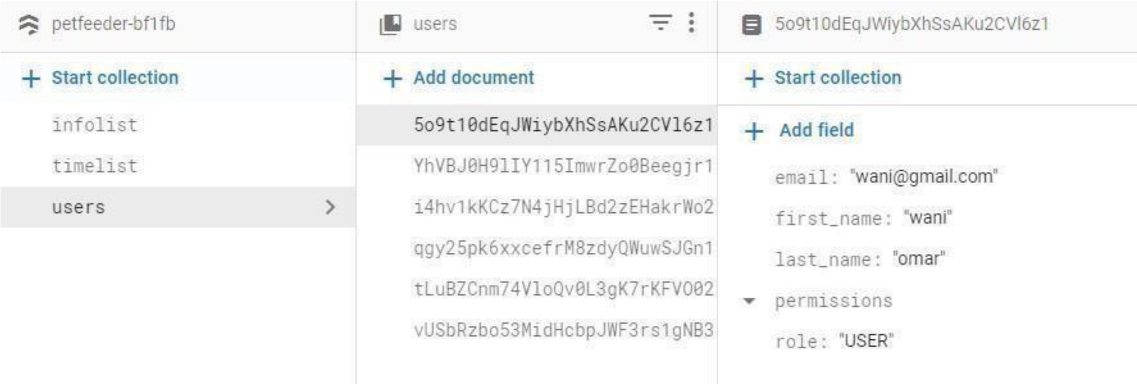
After a user successfully created an account using a mobile application, the information that was created in the authentication database. The real-time database was updated with the alternative specific data. As soon as the connection to the Firebase database was established and successful, the data could be synchronized stored in both databases. Figure 74 and figure 75 show the authentication after user has created.



Authentication

Identifier	Providers	Created ↓	Signed In	User UID
wani@gmail.com	✉	Nov 29, 2022	Nov 29, 2022	5o9t10dEqJWiybXhSsAKu2CVl6z1
piss@gmail.com	✉	Nov 28, 2022	Nov 28, 2022	YhVBJ0H9lIY115ImwrZo0Beegjr1
aliamai@gmail.com	✉	Nov 18, 2022	Nov 18, 2022	2i0mQDJqFhQ64SMUzzZLVkREQ...
saraalia31@gmail.com	✉	Nov 7, 2022	Nov 30, 2022	qgy25pk6xxcefrM8zdyQWuwSJGn1
alia@gmail.com	✉	Nov 7, 2022	Nov 18, 2022	tLuBZCnm74VloQv0L3gK7rKFVO02
test@gmail.com	✉	Oct 31, 2022	Nov 1, 2022	i4hv1kKCz7N4jHjLBd2zEHakrWo2
user@gmail.com	✉	Jul 18, 2022	Nov 29, 2022	vUSbRzbo53MidHcbpJWF3rs1gN...

Figure 74 Authentication After User created



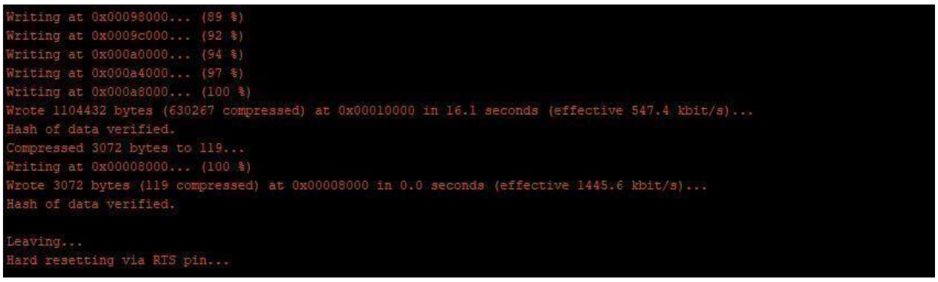
petfeeder-bf1fb
Start collection
infolist
timelist
users
users
Add document
5o9t10dEqJWiybXhSsAKu2CVl6z1
YhVBJ0H9lIY115ImwrZo0Beegjr1
i4hv1kKCz7N4jHjLBd2zEHakrWo2
qgy25pk6xxcefrM8zdyQWuwSJGn1
tLuBZCnm74VloQv0L3gK7rKFVO02
vUSbRzbo53MidHcbpJWF3rs1gNB3
5o9t10dEqJWiybXhSsAKu2CVl6z1
Start collection
Add field

email: "wani@gmail.com"
first_name: "wani"
last_name: "omar"
permissions
role: "USER"

Figure 75 Authentication after user created

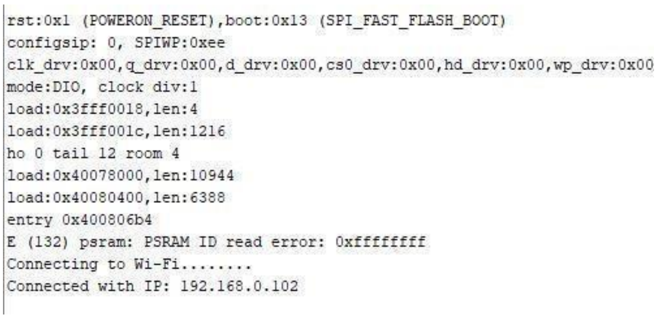
4.4.2 ESP32 Connection to Real-Time Database

The hardware is working after being uploaded via the Arduino IDE, as seen in Figure 76 The ESP32 is connected to the internet and a real-time database in Figure 77, which allows updated data about the user-inputted time and weight for dispensing pet food to be dispense. The value has been successfully read and stored in the Real-time database, as shown in Figure 78.



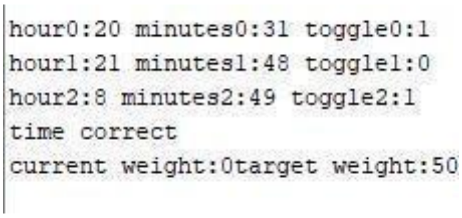
```
Writing at 0x00098000... (89 %)
Writing at 0x0009c000... (92 %)
Writing at 0x000a0000... (94 %)
Writing at 0x000a4000... (97 %)
Writing at 0x000a8000... (100 %)
Wrote 1104432 bytes (630267 compressed) at 0x00010000 in 16.1 seconds (effective 547.4 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 119...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (119 compressed) at 0x00008000 in 0.0 seconds (effective 1445.6 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Figure 76 Uploading successful in the ESP32



```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4
E (132) psram: PSRAM ID read error: 0xffffffff
Connecting to Wi-Fi........
Connected with IP: 192.168.0.102
```

Figure 77 The ESP32 successfully connected to the internet

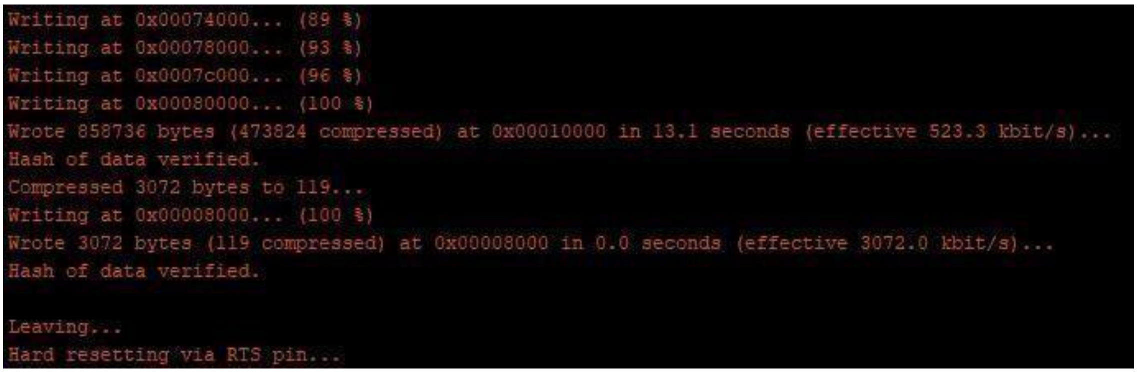


```
hour0:20 minutes0:31 toggle0:1
hour1:21 minutes1:48 toggle1:0
hour2:8 minutes2:49 toggle2:1
time correct
current weight:0target weight:50
```

Figure 78 The ESP32 time and weight reading successfully

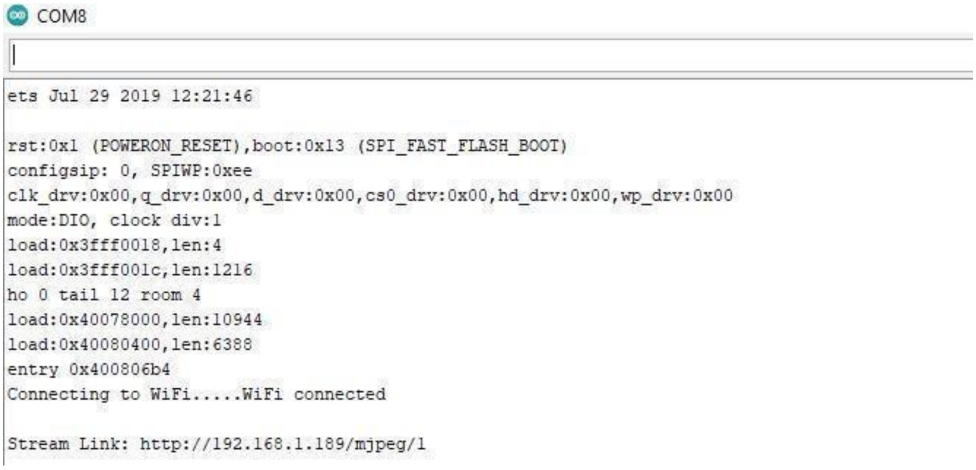
4.4.3 ESP32-CAM connection to Real-Time Database

Figure 79 illustrates that the hardware is operational following upload using the Arduino IDE. Figure 80 shows how the ESP32-CAM is linked to the internet and a live database. The system will then provide the camera's stream link.



```
Writing at 0x00074000... (89 %)
Writing at 0x00078000... (93 %)
Writing at 0x0007c000... (96 %)
Writing at 0x00080000... (100 %)
Wrote 858736 bytes (473824 compressed) at 0x00010000 in 13.1 seconds (effective 523.3 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 119...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (119 compressed) at 0x00008000 in 0.0 seconds (effective 3072.0 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Figure 79 Uploading successful in the ESP32-CAM



```
ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4
Connecting to WiFi.....WiFi connected

Stream Link: http://192.168.1.189/mjpeg/1
```

Figure 80 The ESP32-CAM successfully connected to internet and stream link for camera

4.5 CONCLUSION

The hardware for this system was assembled on a breadboard using the ESP32 at the beginning of its implementation, and after that, the creation of the mobile application began. After configuring the hardware and the mobile application with simple compose code to run both, the connection to the Firebase database was established. Jumper cables were used to link the sensors and connection to the ESP32, eliminating the need for soldiering. The mobile application was initialized with configuration and installed once the combined sensor code was uploaded to the ESP32 using the Arduino IDE. The connection between the ESP32 and mobile application was easily established using the firebase database. The machine will dispense the pet's food based on the weight that the user has input in the mobile application once the data is stored in the Firebase real-time database. The device will also feed the pet at scheduled times according to user input. Due to the machine's integration of ESP32-CAM, the user may also monitor the status on their pet in real time.

CHAPTER 5

CONCLUSION

5.1 INTRODUCTION

The auto pet feeder is the solution for people who adore having pets within the house but may not be able to feed them at regular intervals due to a hectic schedule. By entering the desired weight and release time, this prototype can be used to release food. The time and weight both can be altered and can be given many at once. The stored food will automatically pass through the pipe at the designated time and be gathered in the bowl if the time and weight are chosen through a mobile application. The hardware, ESP32 microcontroller, and mobile application may be used together as software to create an automatic pet feeder system that can be accessed via the WIFI network. This device also allows the user to monitor the condition of their pet in real time. The user has the ability to set both the feeding schedule time and the food weight for the chosen feeding time. The Internet of Things (IoT) technology-enabled Automatic Pet Feeder will help pet owners feed their animals while also being simple to customize to the needs and tastes of the pet owner. By leveraging the ESP32 WIFI module to transfer the sensor readings to the Firebase database through Internet protocol, the Automatic Pet Feeder enables efficient monitoring of the schedule of the pet's food in the container and monitoring of food weight. The use of the Automatic Pet Feeder system was suggested in this study. The weight sensor was linked to the Arduino ESP32 to implement the system, and both the ESP32 and the mobile application were written. This method used a weight sensor to determine the weight of the food that the user had entered into the mobile application. The ESP32 microcontroller received the sensor reading and converted it into readable data before sending it to the Firebase database. The user may monitor the pet feeder schedule and food weight using the mobile application, and they can also regulate the pet's food dispensing by sending messages to the database from the ESP32 microcontroller. This chapter is presented as the paper's conclusion and identifies constraints encountered during implementation as well as potential future work on the system.

5.2 CONSTRAINTS

These constraints, which include hardware, time, cost and technological ones, were some of the ones encountered during the system’s implementation process.

5.2.1 Hardware Constraints

The setup of the automatic pet feeder was a very challenging operation because of the delicate hardware, some of which was broken while handling. Due to a voltage supply imbalance that occurred when the ESP32 microcontroller was being programmed, it was shorted out and could not be used to write programs again. The unstable internet connection is the reason for the ESP32-CAM's malfunction. Because the voltage supply was less than what the weight sensor needed, it was unable to produce the desired results. Even if there were hardware limitations, all of the hardware could be replaced and the voltage supply could be fixed before the project submission deadline.

5.2.2 Time Constraints

Although the system was able to be fully completed before the deadline, there are still a variety of enhancements that could have been made. Since the mobile application was created with only the bare minimum of features, there are still other features that might be added and are covered in the future work that follows. The improvement was added to the discussion of the system's future work since the time constraint was applied in the execution of the system for enhancement.

5.2.3 Cost Constraints

This system's hardware requirements were prohibitively expensive for students, and the recurring purchase of components like microcontrollers and cables due to the need to replace existing components increased the costs associated with completing the system.

5.2.4 Technical Constraints

To build this system, new skills in mobile applications and the Internet of Things were required. Lack of expertise has made it more difficult to conduct research, learn the Arduino and Android Studio programming languages, and set up the ESP32 with sensors. There were numerous failures up until the system's implementation in code was finished, but it took a lot of research and understanding about the Automatic Pet Feeder Machine's concept and implementation process to achieve this goal.

5.3 FUTURE WORK

The implementation of the future work was crucial for enhancing the functionality and robustness of the system. In order to make this system as efficient as possible, a few functions were minimized. These functions can be added in subsequent development. The enhancement of the system will be given priority in this upcoming effort. Even though the system can automatically distribute pet food, keep track of the pet feeder's schedule, and allow users to adjust the food's weight, there are still a few enhancements that might be made in the system's future operations.

Since the mobile application was designed to be used by a single system at a time, it may be used to connect several systems to a single user so that they can all be seen simultaneously by a smart application. The user can easily organize various systems using this way. In addition, the functionality of the system could be improved by adding more users to a single system under the first user of the system where others are also able to handle the system in case the first user cannot perform maintenance on the system such as changing the weight of the food or setting the feeding time. In addition, feeding times in the system can be added in the future to make it easier for users to choose the right feeding times for their pets.

The automatic pet feeder machine can also be improved by including a sensor to detect when the amount of food in the food storage compartment is not enough. For instance, if the food in the food storage container weighs less than 50 grams, users will be notified. Users will find it simpler to avoid becoming complacent with the pet food available in stores not being adequate. This is due to a few users who neglected to add food to the storage area. So, when it's time to feed, it won't be enough to dispense and will have an impact on their pets and also the application.

REFERENCES

[1] “The truth about cats and dogs_ Pets are good for mental health of ‘everyday people’”.

[2] D.-R. Berte, “Defining the IoT,” *Proceedings of the International Conference on Business Excellence*, vol. 12, no. 1, pp. 118–128, May 2018, doi: 10.2478/picbe-2018-0013.

[3] S. Kim, “Smart pet care system using internet of things,” *International Journal of Smart Home*, vol. 10, no. 3, pp. 211–218, 2016, doi: 10.14257/ijsh.2016.10.3.21.

[4] “2017 — Association for Pet Obesity Prevention”.

[5] S. Koley, S. Srimani, D. Nandy, P. Pal, S. Biswas, and I. Sarkar, “Smart pet feeder,” in *Journal of Physics: Conference Series*, Mar. 2021, vol. 1797, no. 1. doi: 10.1088/1742-6596/1797/1/012018.

[6] I. Lee and K. Lee, “The Internet of Things (IoT): Applications, investments, and challenges for enterprises,” *Business Horizons*, vol. 58, no. 4, pp. 431–440, Jul. 2015, doi: 10.1016/j.bushor.2015.03.008.

[7] M. A. Khan and K. Salah, “IoT security: Review, blockchain solutions, and open challenges,” *Future Generation Computer Systems*, vol. 82, pp. 395–411, May 2018, doi: 10.1016/j.future.2017.11.022.

[8] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.

[9] J. Y. Jung, C. M. Ji, J. R. Sohn, H. J. Meng, and B. S. Hwang, “NuriPet: A smart pet feeding machine for SNS,” in *2016 IEEE International Conference on Consumer Electronics, ICCE 2016*, Mar. 2016, pp. 117–118. doi: 10.1109/ICCE.2016.7430544.

[10] Vania, K. Karyono, and I. H. T. Nugroho, “Smart dog feeder design using wireless communication, MQTT and Android client,” in *Proceeding - 2016 International Conference on Computer, Control, Informatics and its Applications: Recent Progress in Computer, Control, and Informatics for Data Science, IC3INA 2016*, Feb. 2017, pp. 191–196. doi: 10.1109/IC3INA.2016.7863048.

[11] M. Ibrahim, H. Zakaria, and E. E. W. Xian, “Pet food autofeeder by using Arduino,” in *IOP Conference Series: Materials Science and Engineering*, Dec. 2019, vol. 670, no. 1. doi: 10.1088/1757-899X/670/1/012069.

[12] A. Alshamrani, R. Qureshi, and A. Bahattab, “A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model Related papers A Comprehensive St udy of Commonly Pract iced Heavy and Light Weight Soft ware Met hodolo… A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model.” [Online]. Available: www.IJCSI.org

[13] M. Mccormick, “Waterfall vs. Agile Methodology.”

[14] R. D. Amlani, “Advantages and Limitations of Different SDLC Models,” 2012. [Online]. Available: www.ijcait.com

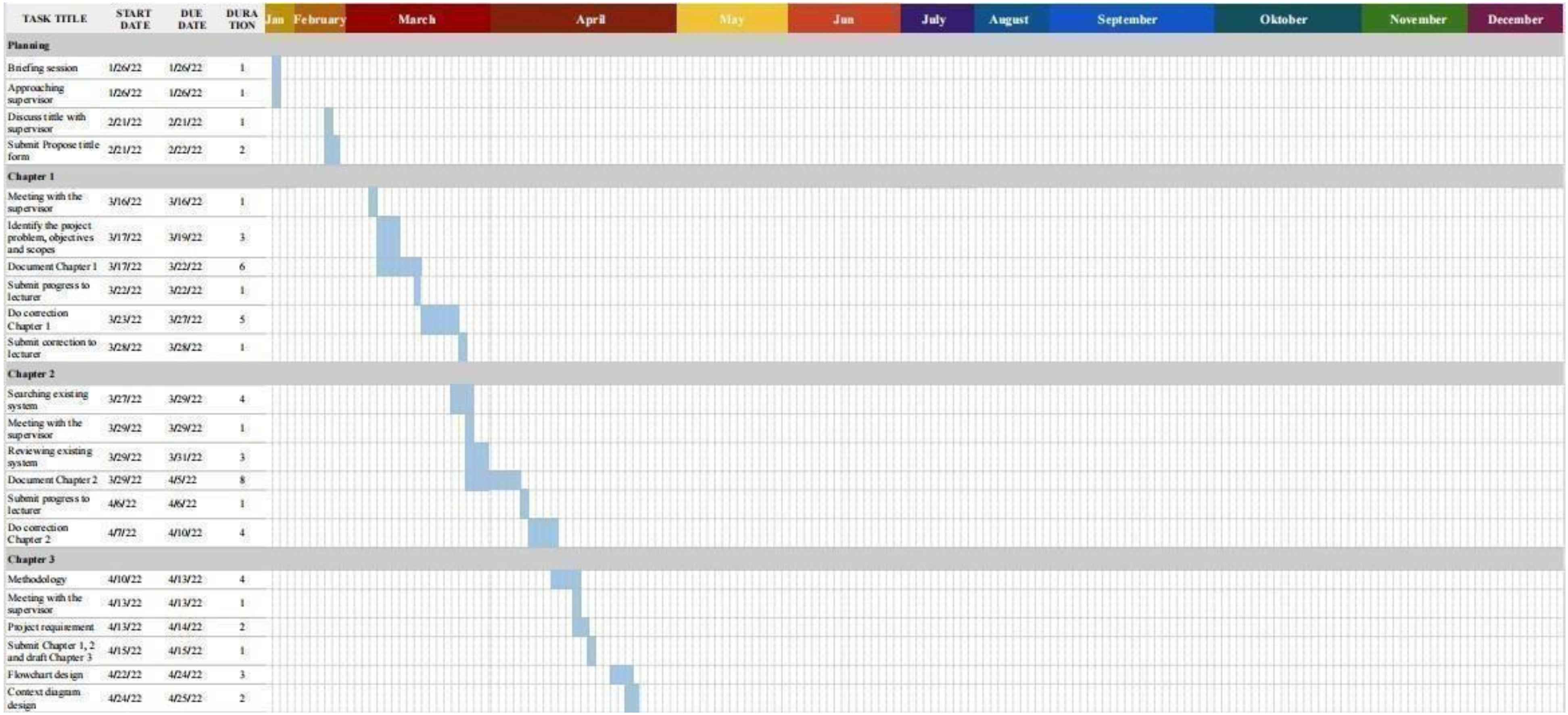
[15] K. Petersen, C. Wohlin, and D. Baca, “The Waterfall Model in Large-Scale Development.”

[16] Y. Beng Leau, W. Khong Loo, W. Yip Tham, and S. Fun Tan, “Software Development LifeCycle AGILE vs Traditional Approaches.”

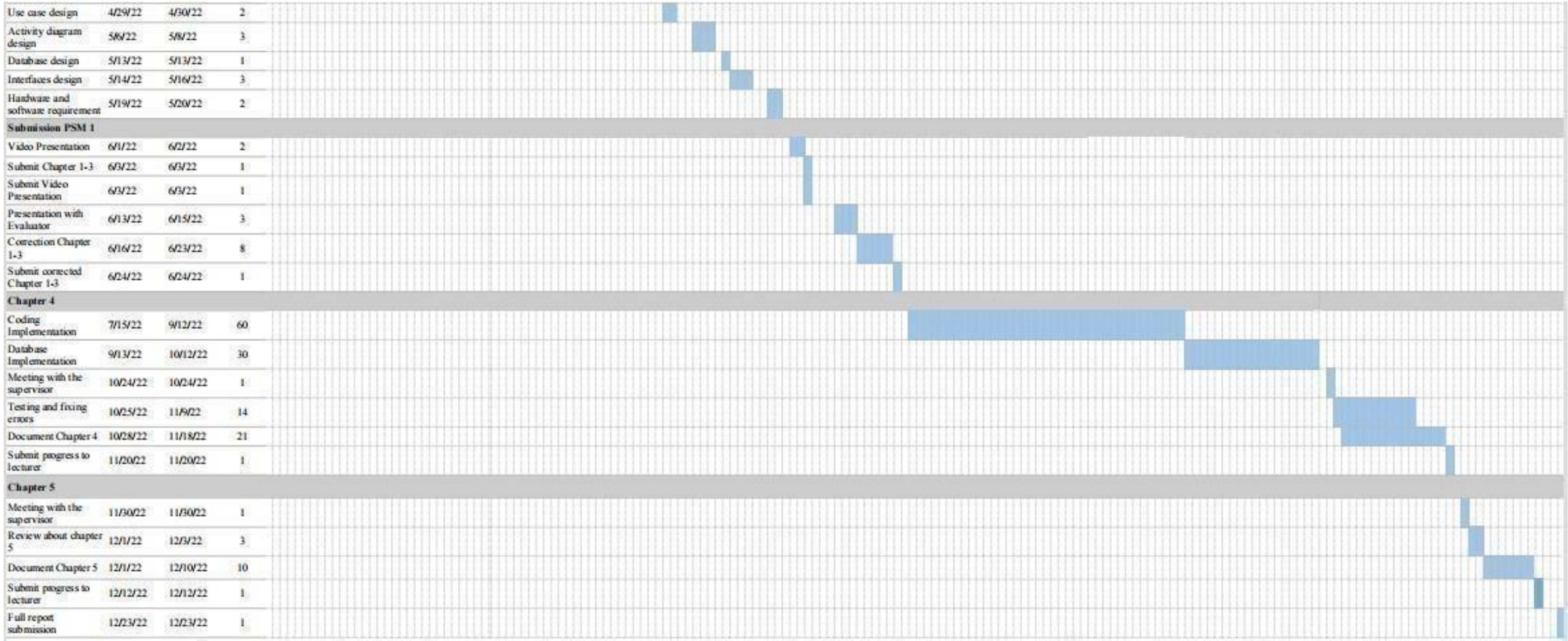
[17] S. Sharma, D. Sarkar, and D. Gupta, “Agile Processes and Methodologies: A ConceptualStudy.”

[18] A. K. Nalendra, “Rapid Application Development (RAD) model method for creating an agricultural irrigation system based on internet of things,” *IOP Conference Series: Materials Science and Engineering*, vol. 1098, no. 2, p. 022103, Mar. 2021, doi: 10.1088/1757-899x/1098/2/022103.

APPENDIX A
GANTT CHART



TASK TITLE	START DATE	DUE DATE	DURATION
Planning			
Briefing session	1/26/22	1/26/22	1
Approaching supervisor	1/26/22	1/26/22	1
Discuss tittle with supervisor	2/21/22	2/21/22	1
Submit Propose tittle form	2/21/22	2/22/22	2
Chapter 1			
Meeting with the supervisor	3/16/22	3/16/22	1
Identify the project problem, objectives and scopes	3/17/22	3/19/22	3
Document Chapter 1	3/17/22	3/22/22	6
Submit progress to lecturer	3/22/22	3/22/22	1
Do correction Chapter 1	3/23/22	3/27/22	5
Submit correction to lecturer	3/28/22	3/28/22	1
Chapter 2			
Searching existing system	3/27/22	3/29/22	4
Meeting with the supervisor	3/29/22	3/29/22	1
Reviewing existing system	3/29/22	3/31/22	3
Document Chapter 2	3/29/22	4/5/22	8
Submit progress to lecturer	4/6/22	4/6/22	1
Do correction Chapter 2	4/7/22	4/10/22	4
Chapter 3			
Methodology	4/10/22	4/13/22	4
Meeting with the supervisor	4/13/22	4/13/22	1
Project requirement	4/13/22	4/14/22	2
Submit Chapter 1, 2 and draft Chapter 3	4/15/22	4/15/22	1
Flowchart design	4/22/22	4/24/22	3
Context diagram design	4/24/22	4/25/22	2



Use case design	4/29/22	4/30/22	2
Activity diagram design	5/6/22	5/8/22	3
Database design	5/13/22	5/13/22	1
Interfaces design	5/14/22	5/16/22	3
Hardware and software requirement	5/19/22	5/20/22	2
Submission PSM 1			
Video Presentation	6/1/22	6/2/22	2
Submit Chapter 1-3	6/3/22	6/3/22	1
Submit Video Presentation	6/3/22	6/3/22	1
Presentation with Evaluator	6/13/22	6/15/22	3
Correction Chapter 1-3	6/16/22	6/23/22	8
Submit corrected Chapter 1-3	6/24/22	6/24/22	1
Chapter 4			
Coding Implementation	7/15/22	9/12/22	60
Database Implementation	9/13/22	10/12/22	30
Meeting with the supervisor	10/24/22	10/24/22	1
Testing and fixing errors	10/25/22	11/9/22	14
Document Chapter 4	10/28/22	11/18/22	21
Submit progress to lecturer	11/20/22	11/20/22	1
Chapter 5			
Meeting with the supervisor	11/30/22	11/30/22	1
Review about chapter 5	12/1/22	12/3/22	3
Document Chapter 5	12/1/22	12/10/22	10
Submit progress to lecturer	12/12/22	12/12/22	1
Full report submission	12/23/22	12/23/22	1

APPENDIX B
USER ACCEPTANCE TEST (UAT)

Testing Report

The user acceptability test for the automatic pet feeder system is covered in this section. To makesure the system is adequate and meets the criteria, testing is done on each module. The system is thoroughly tested and qualified for system implementation, according to the permission of this testing. The flaws that were found are fixed.

Test Case for Mobile Application Module

Table 16 Test Case for Login and Registration Module

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Create an account with unregistered email.	Email: syazwanieomarr@gmail.com First name: Syazwani Last name: Omar Password: waniecantik	Create account successful	Create account successful	Pass	
Create an account with registered email.	Email: saraalia31@gmail.com First name: alia maisarah Last name: hamzan Password: alia1288	Create account unsuccessful and error message will pops-up	Create account unsuccessful and error message will pops-up	Pass	

Login with correct email and password	Email: saraalia31@gmail.com Password: alia1288	Login successful and redirect to home page	Login successful and redirect to home page	Pass	
Login with incorrect email and password	Email: saraalia32@gmail.com Password: alia1104	Login unsuccessful and error message will pop-up	Login unsuccessful and error message will pop-up	Pass	
When Log Out button clicked	Null	Redirect to Log Out page	Redirect to Log Out page	Pass	
When Yes button at the Log Out Page clicked	Null	Log Out from system successfully and redirect to Login page	Log Out from system successfully and redirect to Login page	Pass	
When No button at the Log Out Page clicked	Null	Redirect to home page	Redirect to home page	Pass	

Table 17 Test Case for Reset password Module

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
Insert correct email	Email: saraalia31@gmail.com	Received email to reset password	Received email to reset password	Pass	
Insert incorrect email	Email: saraalia32@gmail.com	Error message will pops-up	Error message will pops-up	Pass	

Reset password successfully	Password: alia1234	Successfully login with new password	Successfully login with new password	Pass	

Table 18 Test Case for Manage Pet Feeder Module

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
When Manage Pet Feeder button clicked	Null	Redirect to Manage Pet Feeder page	Redirect to Manage Pet Feeder page	Pass	
Monitor pet in actual time	Null	The application will display the video in actual time	The application will display the video in actual time	Pass	
User set food weight	Food Weight (g): 40	Send the food weight to exact system in database	Successfully send the food weight reading in database	Pass	
User set feeding time	Time: 13:41:00	Send the feeding time to exact system in database	Successfully send the feeding schedule reading in database	Pass	
User update the toggle status	Toggle: On	Send the toggle status to exact system in database	Successfully send the toggle status reading in database	Pass	
Dispense Food	Food weight (g): 40 Time: 13:41:00 Toggle: On	Machine dispenses foods according to specified weight at the selected feeding time.	Successfully dispenses foods according to specified weight at the selected feeding time.	Pass	

Table 19 Test Case for User Manual Module

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
When User Manual button clicked	Null	Redirect to page User Manual Page	Redirect to page User Manual Page	Pass	
When Application Manual button clicked	Null	Redirect to Application Manual Page	Redirect to Application Manual Page	Pass	
When Warning & Caution button clicked	Null	Redirect to Warning & Caution page	Redirect to Warning & Caution page	Pass	
When Help & Feedback button clicked	Null	Redirect to Help & Feedback page	Redirect to Help & Feedback page	Pass	
When About button clicked	Null	Redirect to About page	Redirect to About page	Pass	

Hardware and Sensor Testing

Table 20 Test Case for Hardware and Software Testing

Test Case	Test Data	Expected Result	Actual Result	Pass / Fail	Comment
When Power on	Null	Proposed system is active and send readings	Proposed system is active and send readings	Pass	
Time feeding is true	Time entered by user	Activate the motor 1 and speaker will sound	Activate the motor 1 and speaker will sound	Pass	

Weight Sensor when food weight is true	Food dispenses according to food weight entered by user	Motor 1 is deactivated	Motor 1 is deactivated	Pass	
Motor 2 when motor 1 is deactivated	Null	Activate the motor 2	Activate the motor 2	Pass	

System Approval

	Name	Date
Verified by: Developer	NOR ALIA MAISARAH BINTI HAMZAN	09/02/2023
Verified by: Supervisor	TS. DR. NOR SYAHIDATUL NADIAH BINTI ISMAIL	09/02/2023

APPENDIX C
SOURCE CODE

Scan the QR code to get the source code for this project.