

AN-NAJAH NATIONAL UNIVERSITY



FACULTY OF ENGINEERING AND INFORMATION
TECHNOLOGY

Computer Engineering Department

Hardware Graduation Project

Smart PetHouse

Group members

Hassan Saed Abu Thiab

Mohammad Belal Mousa

Supervisors

Dr.Luai Malhis & Dr.Emad Natsheh

Presented in partial fulfillment of the requirements for Bachelor degree in
computer engineering.

August, 2022

Disclaimer

This report has been written by students Hassan Abu Thiab and Mohammad Mousa from the computer engineering department at An-Najah National University. It might contain linguistic or informational mistakes. An-Najah National University is not responsible for its content. It also has no responsibility for any misuse of it for anything else than what has been written for.

Acknowledgement

“ First and foremost “ الحمد والشكر لله ” , we would like to thank our supervisors who guided us in doing this project. They provided us with invaluable advice and helped us. Their motivation and help contributed tremendously to the successful completion of the project.

Besides, we would like to thank our friends for their support and for helping us by giving advice and providing the equipment which we needed.

Also we would like to thank our families for their support. Without that support we couldn't have succeeded in completing this project.

At last but not least, we would like to thank everyone who helped and motivated us to work on this project. ”

Contents

Acknowledgement	3
Abstract	6
1 Introduction	1
1.1 Problem Statement	1
1.2 Project Objective	1
1.3 Project Scope	2
1.4 Project Importance	2
1.5 Report Organization	2
2 Constraints, Standards and Earlier Coursework	3
2.1 Constraints	3
2.1.1 Limited information	3
2.1.2 Limited Time	3
2.1.3 Limited Income	3
2.2 Standards	3
2.2.1 Database Back-end	3
2.2.2 React-Native	4
2.2.3 Arduino	4
2.3 Earlier coursework	4
3 Literate Review	5
4 Methodology	6
4.1 Hardware Component	6
4.2 ESP32 Pin Mapping	7
4.3 System Features and Implementation	8
4.3.1 Indoor lighting System	8
4.3.2 Water refilling system	9
4.3.3 Cooling System	10
4.3.4 Gate Control System	11
4.3.5 Animal tracking System	12
4.3.6 Pet feeding System	13
4.3.7 Camera streaming System	14
5 Results & Discussion	16
5.1 Project outcomes	16

6 Conclusion	17
6.1 Things we learnt and future work	17
6.2 Future work	17
References	18

Figures

Figure 4.3.1.1: Indoor lighting system	8
Figure 4.3.1.2.: Light Control by the application	8
Figure 4.3.2.1: Water refilling system	9
Figure 4.3.2.2.: Water Control by the application	9
Figure 4.3.3.1: Cooling system	10
Figure 4.3.2.2: Fan Control by the application	10
Figure 4.3.4.1: Gate Control system	11
Figure 4.3.4.2: Gate and Pet Sensor Control by the application	11
Figure 4.3.5.1: Animal tracking system	12
Figure 4.3.5.2: Track Pet by the application	12
Figure 4.3.6.1: Pet Feeding system	13
Figure 4.3.6.2: Feeding Pet Control and scheduling by the application	14
Figure 4.3.7.1: Camera Streaming system	14
Figure 4.3.6.2: Camera Control/Streaming by the application	15

Abstract

A lot of pet owners struggle with taking care of their pet, especially if the person has work to do and has none to look after them while they are away. And even if the person is around taking care of such tasks can be quite a toll, our project aims to provide these people with the ability to do so through modern methods and ones that will make the process much easier and encourage more people to adopt a pet and give them a safe home.

Pethouse is a system that connects with factors within your house in order to automate specified tasks by the user, it will be remotely controlled by the user through a mobile application that connects with the database which the system is also connected to.

A pet owner house does contain multiple systems that the project can handle, such as heat, air conditioning, providing food automatically, and the ability to provide security for the pets through detection in case the pet leaves the house by sending the user a notification through the mobile application.

All these monitoring and control abilities are handled through a hardware kit, multiple hardware pieces such as sensors as well and a software mobile application will be built using the cross-platform React Native UI framework to work in both android and IOS, and connected to MySQL database instances.

As far as we know, this project was not done before but something quite similar to it already exists in the form of a smart house and our project is inspired by it.

1 Introduction

1.1 Problem Statement

The problem we encounter is the lack of the modern ability which allows the person to manage their pets by making sure to take care of them through multiple means while away and keeping track of where they are on top of their house electronics:

- The limited ability to control the house remotely.
- Being unable to keep track of your pet when they leave the house.
- The constraint of not knowing if your pet left the house or is still around.
- The struggle of managing a pet while being away at work in terms of feeding them and making sure they receive the water required.
- Overall difficulty of managing a pet on top of daily activities such as work and house chores.

1.2 Project Objective

Create a hardware system that's controlled by a React-Native application which enables the User to overcome the mentioned obstacles, by providing methods to automatically control your electronics or remotely through manual means and making sure your pet is taken care of automatically as well.

- The ability to control your lighting system through the mobile application or have it be done automatically.
- The ability to control the heat system through the same means as the lighting system while also deciding a temperature for the cooling system to keep.
- Users being notified if their pet leaves the house.
- A tracking system that makes sure to display the pet location while they are outside the house.
- The ability to schedule meals for the pet while also being able to control the amount of food dropped.
- The ability to refill the water on demand or when it drops below a certain value to make sure the pets are never out of water.

1.3 Project Scope

In this Project, our target audience is house pet owners. It can extend beyond Palestine due to it being able to work within any house and any region so long as the server is able to function within them.

1.4 Project Importance

The project's importance is to help encourage more people to take in pets by making the chores that come with taking care of one less bothersome and time consuming, or enable taking them in while having a busy life and being away at work through remote control while also ensuring their safety through the tracking system.

1.5 Report Organization

The second section (Constraints & Earlier Coursework), will first of all explain the project constraints. It will then go over previously taken courses that were helpful to us in our journey of building and creating the project from the ground up.

The third section (Literature Review), will discuss related work, including similar applications that were built for the same purpose and will mention the differences between their work and this project.

The fourth chapter (Methodology), will explain the system and mobile application features and then how they were implemented. Moreover, it will show what tools and languages were used in implementing the project functionality.

The fifth section (Results & Discussion), In this chapter we will present the results that are achieved from this project and it will give a sufficient details to justify the conclusion.

The final section (Conclusions), will summarize the whole project, final outcomes and it will show what we learned in the journey of developing this project. Also, it will introduce some of the work that will be in the future in this project.

2 Constraints, Standards and Earlier Coursework

2.1 Constraints

2.1.1 Limited information

Despite being familiar with the Arduino language we had new components and pieces to work with that we had to learn how to implement into the code on top of how to connect them physically.

2.1.2 Limited Time

A time period of approximately 3 months to build a project with hardware and components we are unfamiliar with did hinder our progress as things took much longer than we would have anticipated and desired.

2.1.3 Limited Income

A lot of the components used in the hardware did cost quite a lot, and the 3D printing for a few pieces as well. This was the biggest limitation by far as the more features we wanted to implement the more we had to pay and the more we had to replace as well in case of one going faulty over time.

2.2 Standards

2.2.1 Database Back-end

Since we have insisted on using different technology, we used PHP framework - *Laravel*, it is a server-side open-source platform PHP web framework, it is a suitable language to connect the back-end because it offers scalability and parallel code execution. It introduced a new logic and approach for PHP server-side development. It is also light and fast.

2.2.2 React-Native

The reason behind us using *React Native* is because it is a highly used open-source JavaScript Library. It helps in creating impressive mobile apps that require minimal effort and coding. The main objective of *React Native* is to develop User Interfaces (UI) that improves the speed of the apps. And it enables all of that through the ability to reuse components, multiple libraries that make it easy to build and not to forget how many tools are there that support it and enhance its performance.

2.2.3 Arduino

We use Arduino in order to program the chip we are working on and this case it's an ESP32, it enables us to write a C++ code within the Arduino IDE and then upload it onto the ESP32 through the USB in order to have the ESP32 execute functionalities based on the Code we have given it while also having it connect with the Server and WiFi as well.

2.3 Earlier coursework

Thanks to courses we have taken throughout our academic years such as *VHDL*, *PIC*, *React-Native*, *C/C++* and *Javascript*. As for the hardware, previous classes such as Digital Circuit and microcontroller labs helped us be more familiar with how to connect hardware pieces together and create a working hardware project.

3 Literate Review

A smart Pet House is quite similar to many smart houses that exist within the scope of the world, some of them have quite the features but also lack the functionality for pets:

One of these applications is called Apple Home The app enables the users to create plans, which lets them perform multiple actions with just one tap on their smartphone. Just like the leaving home plan while leaving home The additional success factors to this app are security and efficiency. The Apple home kit app is also the most secure smart home application, ensuring complete safety for most of its users. The application has the ability to schedule multiple actions at once while ours can only schedule one of the features and cannot control all of them with one button but through multiple.

Another application is called Honeywell which has a range of devices which include thermostats, security cameras, water leakage, and freezing prevention systems. Additionally, one of the most efficient features and unique ones compared to ours is the geofencing feature. Through the geofencing system, the users can set up a smart perimeter around the house. Once they cross that perimeter the automation system would switch to the default settings automatically.

4 Methodology

For our project, we used React-Native library alongside a few dependencies in order to run and create projects throughout, such as *NodeJS* and React Command line.

Within the Arduino application we used multiple libraries, such as the TinyGPS library in order to control the GPS, the HX711 library for the weight sensor to be programmed, DHT library to read the temperature as well as use the CronAlarm library to finally be able to schedule the functions through the system time of the ESP32.

React-Navigate library was provided in order to navigate through the websites and link them to each other, that way the user can change the page they are viewing through the navigation built in the interface. As well as the React-Native Paper that enabled us to utilize premade components inside our mobile app as well as the Map library in order to view the GPS. We utilized *Laravel - PHP* Framework as a backend as well.

4.1 Hardware Component

Name	Quantity
ESP32	1
ESP32-CAM	1
NodeMCU ESP8266 ESP-12F	1
UBLOX NEO-6M GPS6MV2	1
IR	2
Load Cell 20KG Straight bar, with weight sensor (HX711)	1
Micro Servo Motor	2
LDR Sensor module	1
LEDs	5
DHT11	1
Fan	1

Water pump	1
Water level sensor	1
Two channel relay	2
breadboard	2
FTDI	1
Wires	N

4.2 ESP32 Pin Mapping

Device	Pin
HX711	HX711_dout=>16 HX711_sck => 2
LDR Sensor module	36
DHT11	4
Water level sensor	34
IR1 (Door)	35
IR2 (for Pets)	33
DOOR_LED	32
4 LEDs_PIN	22
Fan	21
Pump	18
Camera	19
Micro Servo Motor	Feeding => 14 Door => 15
UBLOX NEO-6M GPS6MV2	1 & 2 on NodeMCU

4.3 System Features and Implementation

4.3.1 Indoor lighting System

This system consists of the following:

- 4 LEDs
- LDR (Light detection sensors)
-

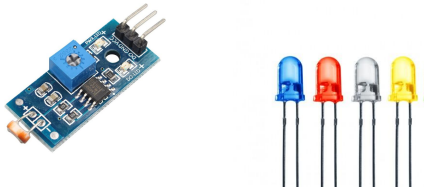


Figure 4.3.1.1: Indoor lighting system

The lighting system can either be a fully automatic system or a manual one through the remote controls of the mobile app. The automatic system can be toggled on and off through the mobile application, if toggled on then the user won't be able to control the LEDs manually until it's toggled off, while toggled on if the sensor reading is below a certain threshold then the lights will turn on and if they are above it they will return to on state.

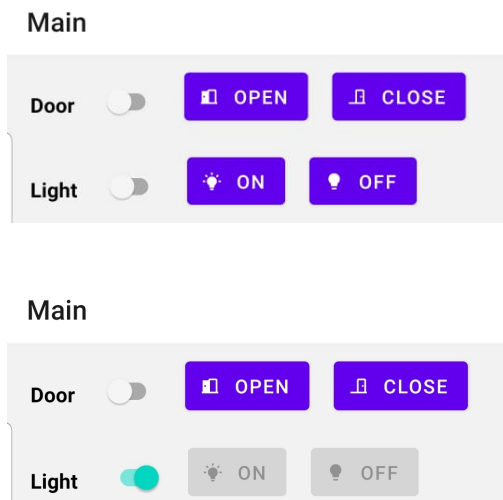


Figure 4.3.1.2.: Light Control by the application

4.3.2 Water refilling system

This system consists of the following:

- Water pump
- Water level sensor
- Relay 12V
- Voltage source 12V

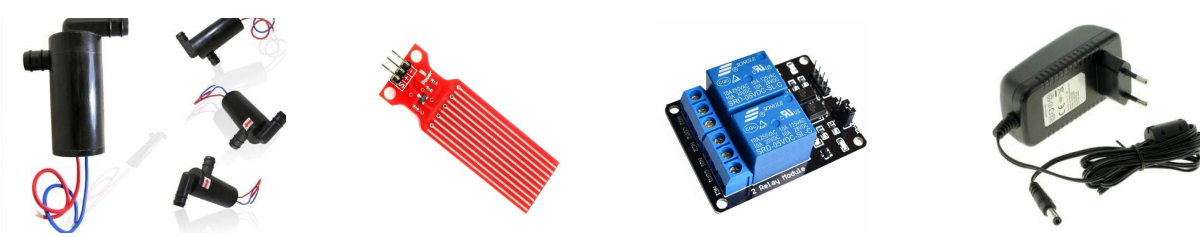


Figure 4.3.2.1: Water refilling system

This system works similar to the previous one except the reading is done through the water level sensor, the sensor will read the water level in the cup and once the threshold of water provided to the pet drops below a certain level, the relay will receive a HIGH signal and therefore turning on the water pump, the water will keep pumping until the water reaches another level where it will stop pumping. The pump can also be controlled manually so long the user toggles off the sensor through the mobile application

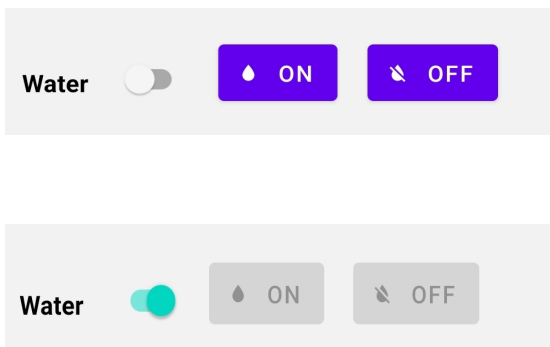


Figure 4.3.2.2.: Water Control by the application

4.3.3 Cooling System

This system consists of the following:

- Fan
- Temperature sensor
- Relay 12V
- Voltage source 12V

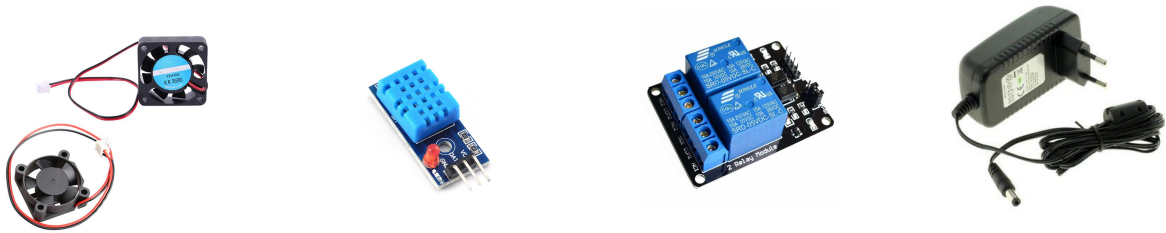


Figure 4.3.3.1: Cooling system

This system will require the DHT library in order to turn the temperature reading sensor reading into ones that can be easily understood in the celsius degree. This system is automatic as it will work on cooling the house as the user will enter the value of the temperature inside the mobile application in order for the sensor to turn the fan on and off based on that value. The temperature sensor will seek to keep the heat within that threshold and that sensor can also be turned off in order to enable manual control.

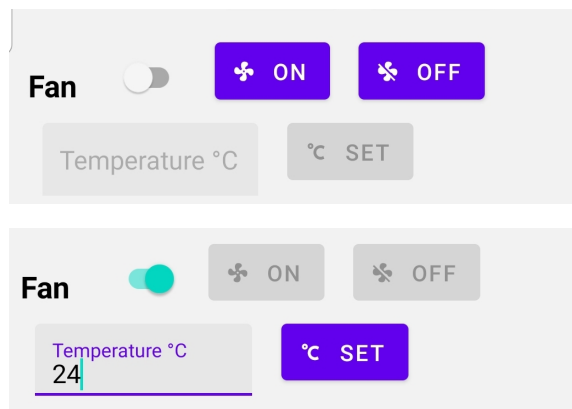


Figure 4.3.2.2: Fan Control by the application

4.3.4 Gate Control System

This system consists of the following:

- Servo motor
- IR sensors X2
- Voltage source 5V

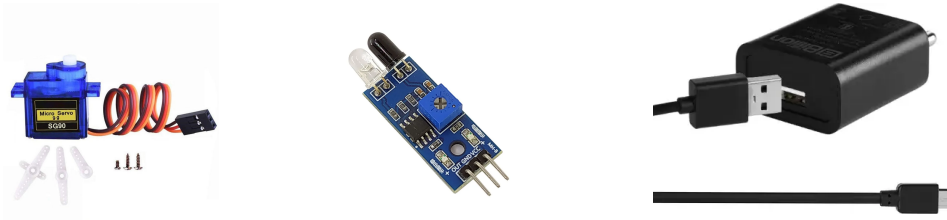


Figure 4.3.4.1: Gate Control system

This system will use the Servo library that's made for ESP32 in order to control the Servo motor and attach it, the gate system functions as the following. If the IR gives a positive reading within a certain threshold then that means it detected an object or a person within the range of the gate and the door will open if the upper IR is triggered. If the bottom IR is triggered and the upper is not triggered within 2 seconds then that means that it's a pet that's being detected and not a human and the user will be notified about that, pressing the notification will lead the user into the next system we will talk about. This door sensor can be turned off in order to control it manually.

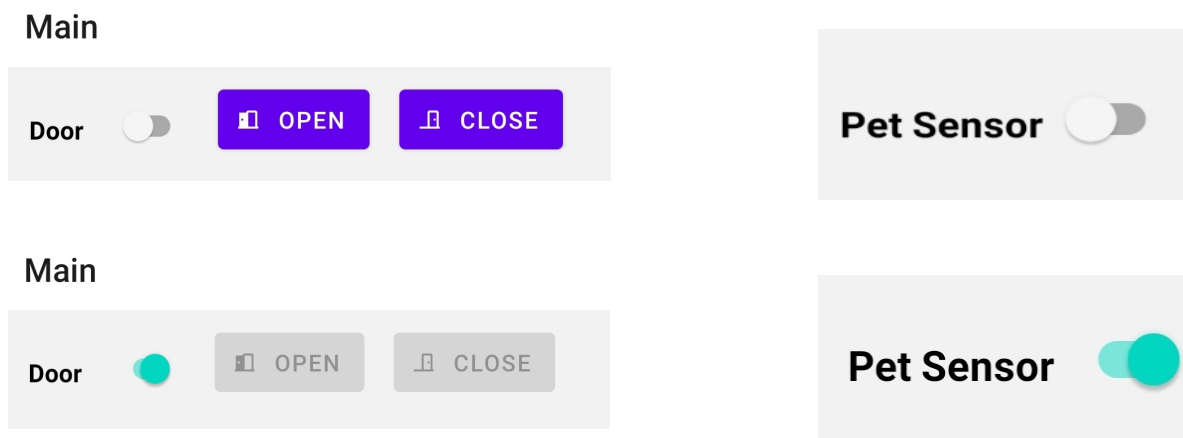


Figure 4.3.4.2: Gate and Pet Sensor Control by the application

4.3.5 Animal tracking System

This system consists of the following:

- ESP8266
- GPS6MV2
- Voltage source 5V



Figure 4.3.5.1: Animal tracking system

This system will be attached to the pet's collar and it will use the TinyGPS library in order to use the GPS and get the coordinates of the pet while it's outside the house in order to provide safety if the pet is lost, making the ability to find them much easier as the GPS coordinates will be displayed on a map within the mobile application.

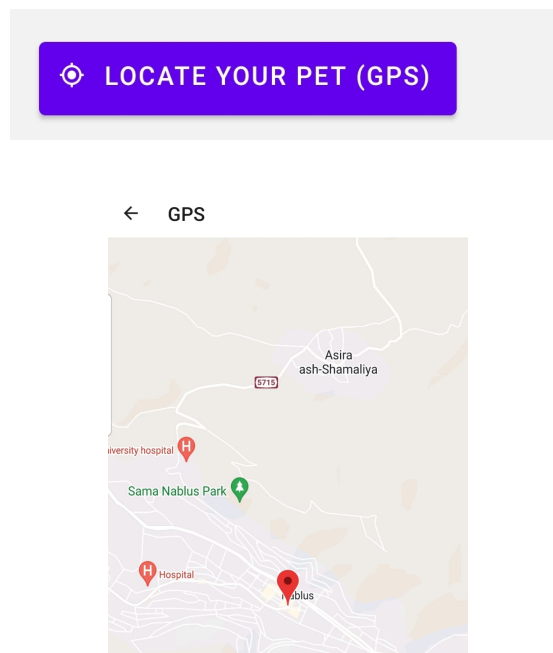
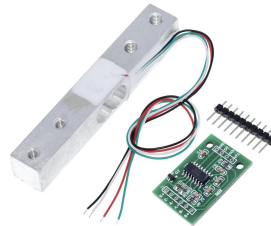


Figure 4.3.5.2: Track Pet by the application

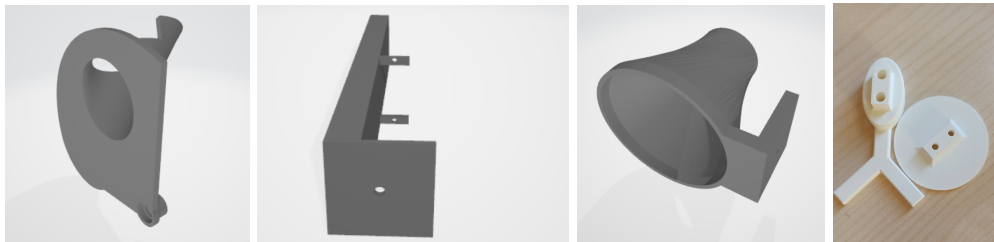
4.3.6 Pet feeding System

This system consists of the following:

- Servo motor
- HX711
- Loadcell 20kg
- 3D printed pieces



(a) The chips



(b) 3D printed pieces

Figure 4.3.6.1: Pet Feeding system

This system consists of an HX711 connected to a load cell. The load cell has two 3D printed pieces attached to it in order to create space between it and the actual weight for an accurate measurement by making sure the weight presses on one edge and not the center. The food will be placed within a cup and the cup and the servo motor will be connected to a lid that opens and closes by being turned at a specific angle through the servo motor and as a result the food will drop.

The food will keep dropping until the load cell reaches a certain weight then the lid will go back to closing. This system supports three meals that can be scheduled and that will be done through the Cronalrams as it will check the system time of ESP32 and then execute the function given to it once the time reaches the one decided by the user through a time picker in the mobile and the user can have the food drop immediately instead of picking a time as well.

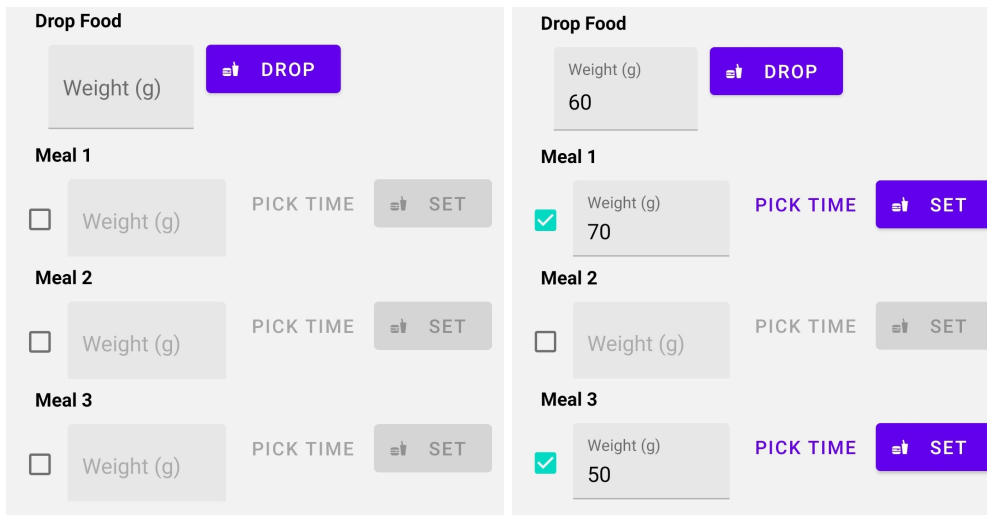


Figure 4.3.6.2: Feeding Pet Control and scheduling by the application

4.3.7 Camera streaming System

This system consists of the following:

- ESP32CAM
- Relay
- Power source 5V

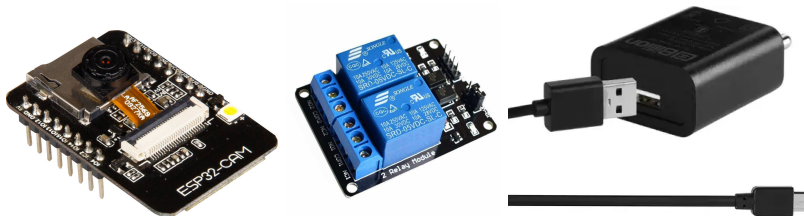


Figure 4.3.7.1: Camera Streaming system

This system uses an ESP32CAM in order to video stream the outside of the house, acting as a security system by providing a link to the stream and saving the link in the database, afterwards that link can be opened within the mobile application in the form of a webview in order to view the surroundings of the house. The relay is there to turn the CAM on and off

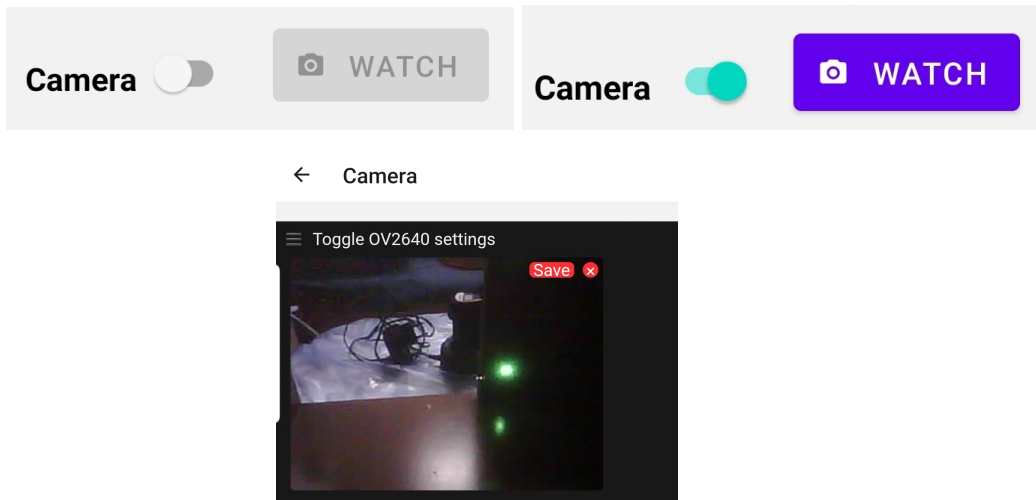


Figure 4.3.6.2: Camera Control/Streaming by the application

5 Results & Discussion

The project ended up being a fully automatic household system in multiple aspects, from lighting, cooling to security as well. The system also provides unique functionalities that take care of your pet which fulfills the “Pet” portion in the name of our project as it’s able to schedule meals and feed pets automatically while also providing the user control over the amounts of food they consume as well as providing the water they need constantly through the water sensor and pump. Tracking the pets and knowing where they are to guarantee their safety at all times by making sure they are not taken or lost through the GPS system and the notification that the user receives.

5.1 Project outcomes

- A react native application that enables the user to control the system
- An indoor lighting system that is handled automatically through sensor or manually.
- An indoor cooling system that is handled automatically through sensor or manually.
- An automatic water refiller to make sure the pet doesn’t run out of water.
- The ability to schedule meals for the pet in order for them to be fed automatically while the person is away
- Control the amount of food the pet is being fed through a weight sensor and a threshold decided by the user
- An automatic gate that can also be turned manually, the gate is able to also detect if the pet leaves and send a notification to the user
- A GPS tracking system to track the pet if they are outside the house to ensure their safety
- Monitoring system through the Camera in order to provide security for the household

6 Conclusion

In summary, we have created an application able to provide the Users with the ability to control their house automatically or manually through an application that can be used remotely through the smartphone. The user will also be able to take care of their pet while they are away from the house and ensure their safety through the tracking and security implementations.

6.1 Things we learnt and future work

- Building a mobile application using React Native and other libraries.
- Connecting the ESP32 to the WiFi and the database.
- How to handle sensors of multiple kinds.
- How to work on the load cell and schedule events.
- Coding the ESP32 to function as a fully-fledged house system.

6.2 Future work

- Enable the user to have more control over the devices of their household
- The ability to easily add new devices onto the household system.
- Image processing and the AI to add up onto the functionality of the camera rather than just monitoring .
- The ability to handle more pet tasks automatically such as cleaning behind the pet and ensuring their litterbox is also clean.
- More features that cover more devices within the household such as fire alarm.
- Better tracking system using GPRS/GSM.

References

Introduction · React Native. (2022, June 22). React Native. Retrieved August 24, 2022, from <https://reactnative.dev/docs/getting-started>

(n.d.). Expo Documentation. Retrieved August 24, 2022, from <https://docs.expo.dev/>

The PHP Framework For Web Artisans. (n.d.). Laravel. Retrieved August 24, 2022, from <https://laravel.com/docs/9.x>

SQLite Documentation. (n.d.). SQLite. Retrieved August 24, 2022, from <https://www.sqlite.org/docs.html>

Technical Documents. (n.d.). Espressif Systems. Retrieved August 24, 2022, from <https://www.espressif.com/en/support/documents/technical-documents>

ESP32 Tutorials. (n.d.). ESP32 Tutorial. Retrieved August 24, 2022, from <https://esp32io.com/esp32-tutorials>