جامعة جدة
University of Jeddah

The Robot "Nybble" Cat

Rayan Dahlawi, Khalid Alshalawi, Aseel Alghamdi

November 2022

Contents

1 - Chapter 1 -

1.1 Introduction

In this chapter, we will discuss an overview of the project, in addition to the problems in the project, ways to solve them, and what we aim for in the future and the project plan.

1.2 General Context

Nybble is a brand-new type of pet that isn't just a walking robotic cat. It's an advanced robotic kitten with a tiny computer, customized software, that can be programmed in our preferred language, and any upgrades you can imagine. We will use a Raspberry Pi. When adding a Raspberry Pi, it is considered as adding a brain to the cat, in addition to the Arduino, which in turn allows the cat to be moved using the servo motor.

1.3 Problem Definition

The Nybble cat has no ability to recognize the voice in its surroundings. The problem is that the cat is not developed and only walks; it does not know the sounds and cannot distinguish them from the voice of the owner so that he can understand his owner's orders.

1.4 Aim and the Objective of Project

Making a robotic cat (Nybble Cat) that behaves like a real cat and recognize the owner's voice to receive commands from the owner to do some action like; come, do a push-up, stand up, etc., when hearing a voice different.

1.5 Proposed Solution

Voice recognition to do commands - We need a microphone to be connected and defined in the raspberry pi by downloading the python on the raspberry pi and downloading the libraries that help in recognizing voice commands.

1.6 Report Outline

The report contained four different chapters in total. Each consists of a number of pieces that have been designed to present them in an understandable manner. In the first chapter, there is a presentation that is more general in nature regarding the project. The second chapter will focus on the works that are connected to the project that has been finished and will discuss their significance. As a direct consequence of this, a comprehensive description of all the important topics and the existing systems will be produced. In the third chapter, we will start collecting the data and prepossessing it. In addition, we will discuss the benefits and cons of the already available systems, as well as the requirements for our project, both in terms of its functionality and its non-functional aspects. In the fourth chapter, we will do some diagrams to explain our project graphically.

1.7 Project Plan

The following Figure 1 is going to be the structure of the project plan that we will be working on

Week	Phases	Tasks	During time	Responsibilities
1 & 2	Selection phase	choose team members	1 day	Rayan
		looking for topics	5 days	Rayan – Khalid – Aseel - Dr. Sajjad
		finalize project topic	5 days	Rayan – Khalid – Aseel - Dr. Sajjad
3 & 4	Planning phase			
		the problem to solve	5 days	Rayan – Khalid – Aseel - Dr. Sajjad
		information gathering	4 days	Rayan – Khalid – Aseel
		Aims of the project	1 day	Rayan – Khalid – Aseel
		project objectives	1 days	Rayan – Khalid – Aseel
		feasibility study	3 days	Rayan – Khalid – Aseel
		project plan	1 day	Rayan – Khalid – Aseel
5 & 6	Problem understanding			
		Defining stakeholders	1 day	Rayan – Khalid – Aseel
		background of the project domain	3 days	Rayan – Khalid – Aseel
		literature review	5 days	Rayan – Khalid – Aseel
		Comparison results with feasibility study	3 days	Rayan – Khalid – Aseel
7 & 8	Analysis phase			
		functional & non-functional requirements	4 days	Rayan – Khalid – Aseel
		hardware requirement	4 days	Rayan – Khalid – Aseel
9 & 10	Drawing phase			
		Use case Diagram	3 days	Rayan
		Sequence Diagram	3 days	Rayan
		ER Diagram	3 days	Khalid
11 & 12	Design phase			
		Writing ch1,2,3,4	10 days	Rayan – Khalid – Aseel
		Review	2 days	Dr. Sajjad
		Editing	3 days	Rayan
		Final design of the paper	3 days	Rayan

Figure 1: Project Plan

2 - Chapter 2 -

2.1 Introduction

This chapter will act as an introduction to the subsequent discussion of works that are linked to our project. In addition, this chapter is broken up into three distinct sections. In the first section," Motivations," we will provide a comprehensive description and illustration of the project's domains. In the second section, we will demonstrate how the project domain must be related to critical activities in order to effectively and appropriately deliver project outcomes. In the final portion, a comparison of the technical features and the details will be made.

2.2 Motivations

Today, Saudi Arabia has become a base station in the field of artificial intelligence and robotics, and it is among the plans of Vision 2030 and future life depends on this robotic technology. NEOM is the most important example of an integrated city based on artificial intelligence and robotics. That is why he was chosen to work on this project that relies on robotics and artificial intelligence to create a robotic cat that behaves like a real cat and fun of using it.

2.3 Background

In this section, we will take a quick look at each domain and outline how each the domain will be involved in the process of constructing this project.

2.3.1 Robotics

Robotics is a branch of engineering that involves the conception, design, manufacture, and operation of robots. The objective of the robotics field is to create intelligent machines that can assist humans in a variety of ways.

2.3.2 Artificial Intelligence

Artificial intelligence allows machines to model, and even improve upon, the capabilities of the human mind. From the development of self-driving cars to the proliferation of smart assistants like Siri and Alexa, AI is a growing part of everyday life. As a result, many tech companies across various industries are investing in artificially intelligent technologies.

2.4 Existing Systems

Petoi Robot Dog Bittle (Figure 2)
The Petoi Bittle Robot Dog is a tiny but powerful robot that can play tricks like real animals. Every bit has been fine-tuned to fit agile maneuverability

into a palm-sized robot pet. Bittle is a DIY, servo-based robot dog from Petoi, controllable via Bluetooth, infrared, and WI-FI. It comes disassembled in kit form. Once assembled, it's remarkably agile, operating via remote control, the app, or a number of different programming options. You can run demo codes before writing your own. The robot is built on Petoi's OpenCat open-source platform and features a customized Arduino board to coordinate movements. And, being open source, users can add on different smart sensors, accessories, or even AI chips.

Figure 2: Robot Dog Bittle

2.5 Comparison Table

In this section, we will talk about the differences and similarities between our project and the related systems. Figure 3 will show the comparison between a Robot Dog Bittle and a Robot Cat Nybble since they are made by the same company.

		Robot Dog Bittle	Robot Cat Nybble
Robot	Appearance	Dog	Cat
	Software supported	CodeCraft (Scratch-based) Arduino IDE A Python API sending serial commands	Arduino IDE A Python API sending serial commands
	Assembly time	1 hour	4 hour
	Dimensions	200 x 110 x 110mm(7.9 x 4.3 x 4.3inch)	250 x 107 x 140mm(10 x 4.2 x 5.5inch)
	Weight	290g(10.2oz)	320g(11.2oz)
	Number of joints	9	11
	Frame material	Plastic	Wood
	Colors	Black & Yellow	N/A
Controller	Controller board	NyBoard V1	
	CPU	ATMega328PA	
	External EEPROM	64Kbit	
	IMU	6-Axis MPU6050	
	PWM channels	16	
	Grove	4	
	RGB LED	7	
	Built-in connectivity & features	Serial UART I2C network Infrared receiver Buzzer	
	Bluetooth/WiFi connectivity	Official dongles	
	Raspberry Pi support	Yes	
Servo	Servo	P1S	DS031
	Quantities in pack	10	11
	Max voltage	8.4V	
	Wall Material	Plastic	
	Gear	Alloy	
	Motor	Coreless	Cored
Battery	Battery Type	Li-ion battery pack	
	Capacity	Typ. 7.4V 1000mAh	
	Current typ./max.	2A/5A	
	Battery life	1 hour	
	Charger	USB 5V 1A	
	Charge time	2 hours	
	Charger cable	Regular micro-USB	

Figure 3: Table of Comparison between a Robot Dog Bittle and a Robot Cat Nybble.

3 - Chapter 3 -

3.1 Introduction

In the beginning, we will talk in this chapter about several topics, namely, requirements gathering, in which we relied on people's opinions and suggestions so that we can improve and create new ideas. We also talked about functional requirements and non-functional requirements, which were divided based on the importance of each piece in this project. Finally, it summarized the benefit of this project.

3.2 Requirements Gathering

Based on our idea, we reviewed a lot of people's opinions on this topic and discovered some features that will be great for creating new ideas with artificial intelligence using Raspberry Pi, such as voice recognition.

3.3 Functional Requirements

There are three levels of functional requirements, which are: basic-level, middle-level, and high-level requirements for an infrared remote.

• Basic-level
Programming the servomotor with the Arduino to move the cat forward, backward, sitting and getting up by entering the command in the NyBoared (enhanced Arduino) to be executed.

• Middle-level
Added sensors, such as the ultrasonic sensor when approaching a specific target, the cat stops so as not to hit any objects.

• High-level
Make the cat able to self-control enough to roam the whole house in addition to the things mentioned earlier in the level (basic and middle).

3.4 Non-functional Requirements

• The frame kit is non-functional
• Rubber skin for a cat's arm
• Zip Tie tape for attaching wires and improving the appearance

3.5 Conclusion

In conclusion, we benefited greatly from the work of this project in terms of assembling and programming, and we also learned about Arduino and about Raspberry Pi and how to program them and we have also benefited from the use of artificial intelligence by making our information increase on this topic and linking it to life matters we aspire in the near future to work on developing the cat by making it useful in order to be more than an ordinary cat.

4 - Chapter 4 -

4.1 Introduction

The topic of sprint 1 development will be the focus of the discussion that takes place in Chapter 4, these sections are the Use-case Diagram, in which we will create a use-case diagram to represent all of the functional requirements; the Sprint 1 Static Aspect Design, in which we must create an E-R diagram to design all of the functional requirements; the Sprint 1 Architectural Aspect Design, in which we must create a sequence diagram to represent the various hardware components of the functional requirements; and the Sprint 1 Dynamic Aspect Design, in which we represent how the functional requirements will be implemented.

4.2 Use Case

As we see in figure 4 we will program the NyBoard by using Arduino IDE to program the servo motors to sit, walk, etc. If the owner gives the cat commands like; go, sit, push up, etc., the Raspberry Pi will detect the voice and then send it to the Nyboard to obey the commands and do actions. Also, we will program the ultrasonic sensor to prevent any crashes.

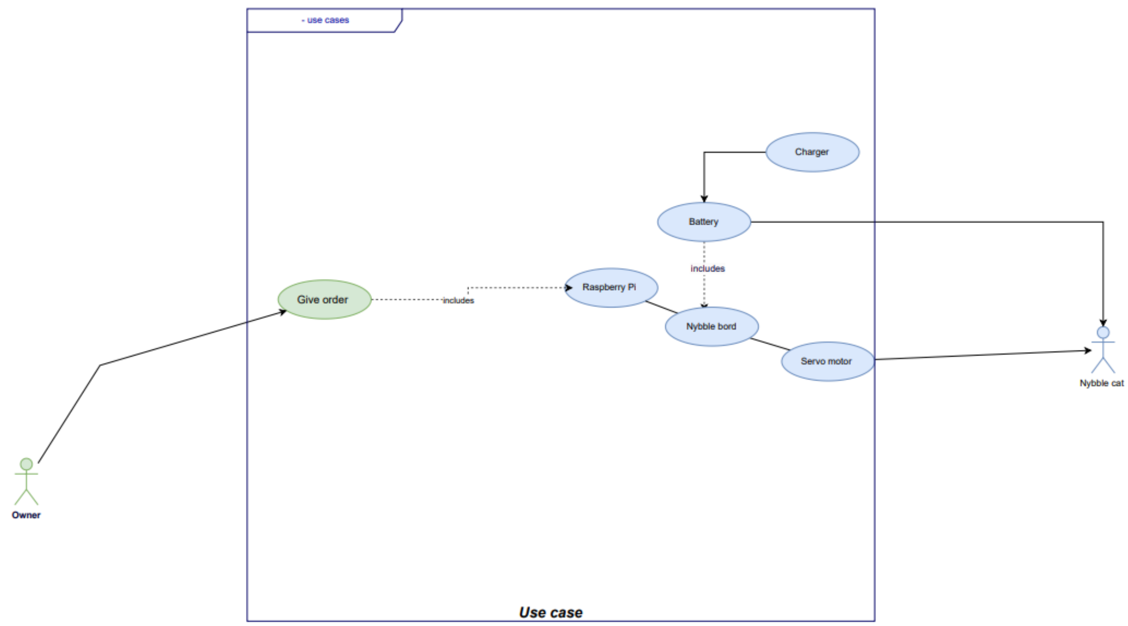


Figure 4: Use Case Diagram.

4.3 ER Diagram

As showen on Figure 5 The owner is associated in a relationship with the table raspberry pi is optional-many And the relationship between the raspberry pi and the owner is mandatory-one,so that there may be more than one order issued by the owner, and there may not be, the same applies to the rest of the tables and the relationships between them.

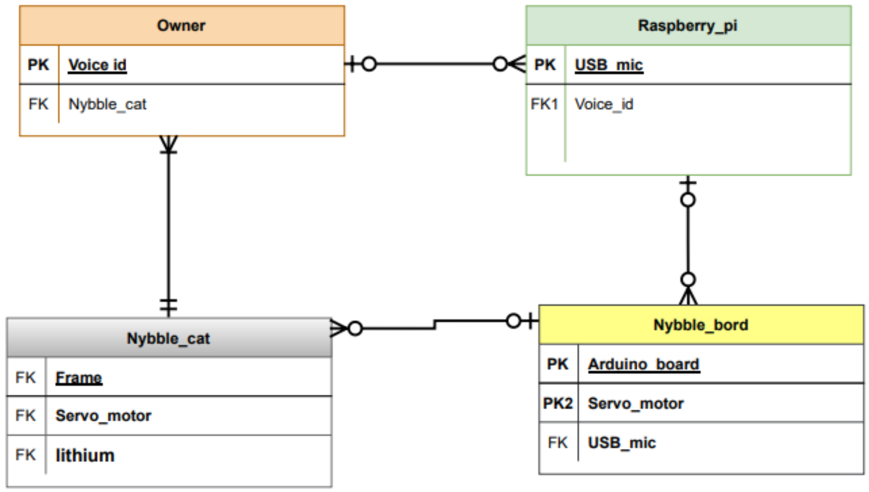


Figure 5: ER Diagram.

4.4 Sprint 1 Architectural Aspect Design

In addition to the necessary hardware for the configuration, we have compiled a table that describes each component and the reasons why we require it.

Hardware	Explanation
NyBoard	Programming and linking with other pieces to do action
Ultrasonic sensor	It used to know the distance
servomotor	Used to measure the position
Lithium battery	It is responsible for powering the other parts

Figure 6: Hardware Explanation

4.5 Sequence Diagram

As shown in Figure7 the owner will command the cat to do an action for example (come), then the sensor sends a signal to the Raspberry Pi will analyze then send it to Nyboared and then the cat will come to the owner.

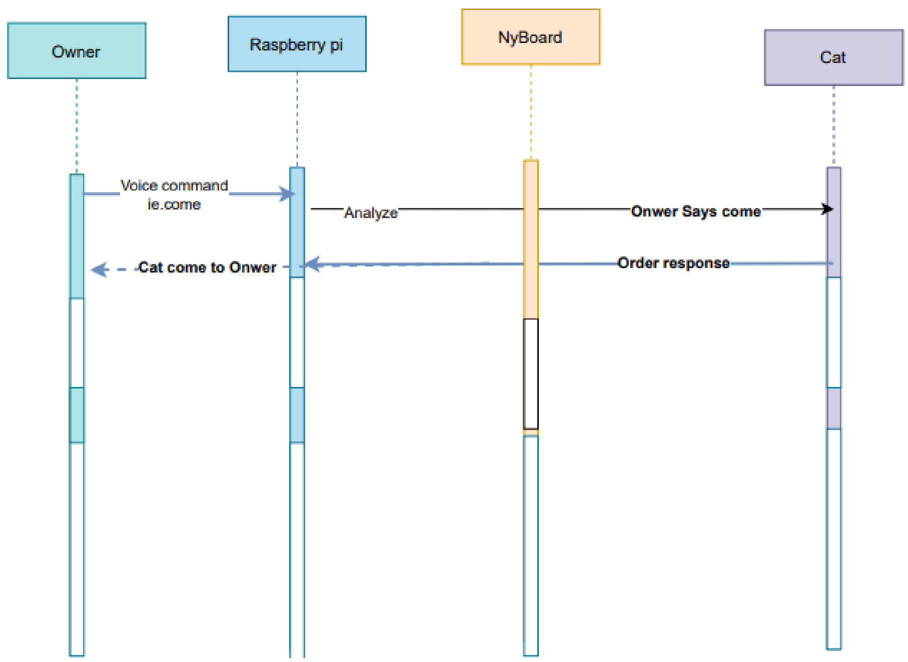


Figure 7: Sequence Diagram

4.6 Sprint 1 Testing Evaluation

We will buy a robot cat, ride it properly, and develop a program for it so that, as they mentioned on their page ("Petoi"), the cat will do is it will walk like a real cat and move its head like the original Cat movement, and also sit. You can also move any part of the object according to your choice, and the cat can be stable when falling from a low place (if it falls from a high place it will disintegrate), and the amazing thing is that the cat can do push-ups and can only stand on two legs and can only stand on its four legs. This is a summary of what the robot cat does. We will develop a program that will help the robot cat distinguish sounds and receive commands from its owner. (These are just beginning ideas that may improve and change to suit our work.)

4.7 Conclusion

The project diagrams were generated, and this chapter discusses the construction of the diagrams as well as the coverage they provide. Because of this, it is now a lot easier for us to track the process from the use case diagram all the way up to the sprint. This process includes the Static Aspect Design to create a detailed class diagram for our functional requirements, the Architectural Aspect Design to determine the hardware required for this project, the Dynamic Aspect Design, Implementation, Testing, and Evaluation, where we estimate scenarios to test and evaluate the project, and the Architectural Aspect Design to determine the hardware required for this project. In addition, this procedure includes the Architectural Aspect of Design.

5 - Chapter 5 -

5.1 Introduction

The topic of sprint 2 development will be the focus of the discussion that takes place in Chapter 5. These sections are the use-case diagram, in which we will create a use-case diagram to represent all of the intermediate functional requirements; the sprint 2 static aspect design, in which we must create an E-R diagram to design all of the intermediate functional requirements; the sprint 2 architectural aspect design, in which we must show the different hardware components of the intermediate functional requirements; and the sprint 2 dynamic aspect design, in which we represent any suitable diagram or another detailed diagram to reflect the internal dynamic of the intermediate functional requirements; the sprint 2 implementation; and the sprint 2 testing and evaluation will be implemented; the sprint 2 testing and evaluation will be implemented; the sprint 2 testing and evaluation will be implemented.

5.2 Use-case diagram

As shown in figure 8 Nybble works in three ways: remote control, and its app, also with Arduino. As for the voice command, the sound is captured using a microphone built into the Raspberry Pi, after that it sends a signal to the board so that the motors interact and the cat follows the command. the same way happens with the other three, but without the intervention of the Raspberry Pi, in order for the board to interact, it must work The built-in battery, so that interaction occurs,

5.3 Sprint 2 Static Aspect Design

As shown in figure 9 the owner is in a relationship with the table raspberry pi optional - many and the relationship between raspberry pi and the owner is mandatory, so that there is more than one command issued by the owner, and there may not be, in addition to the existence of a relationship between the board and the battery, which indicates that it is mandatory for operation, and it is a one-to-one relationship. The board has Prime Key in addition to foreign Key and the same applies to the rest tables and the relationships between them.

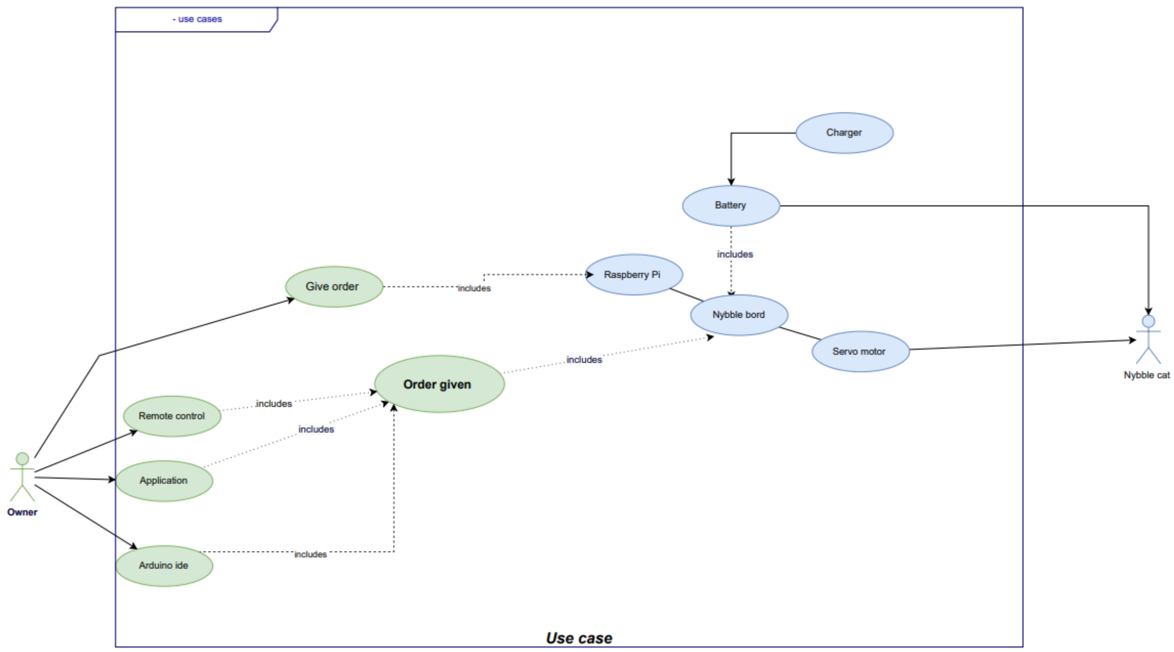


Figure 8: Use case diagram

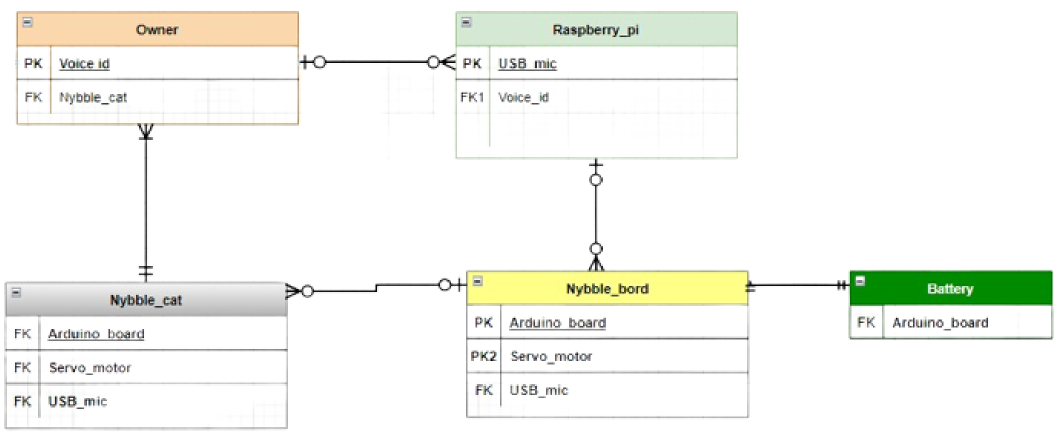


Figure 9: ERD diagram

5.4 Sprint 2 Architectural Aspect Design

The figure 10 is showing the different hardware components of the intermediate functional requirements:

- Wooden frame: It contains pieces of pressed wood that make up the structure for the cat.
- Battery Holder: The load is the battery, which in turn is the power connector for the cat.
- Sanding Block: movement barrier .
- Rubber toe: The rubber is attached to the cat's limbs to prevent it from slipping.
- NyBoard: It is the brain of the cat, as it is the one that receives orders and carries them out.
- Sketch uploader: It is a piece that connects to the board and enters programming commands for the board.
- Ultrasound sensor: Distance sensor to prevent the cat from colliding with barriers.
- Servo Motor: that make the cat move
- Infrared Remote: The controller is the one that gives the cat movement commands.

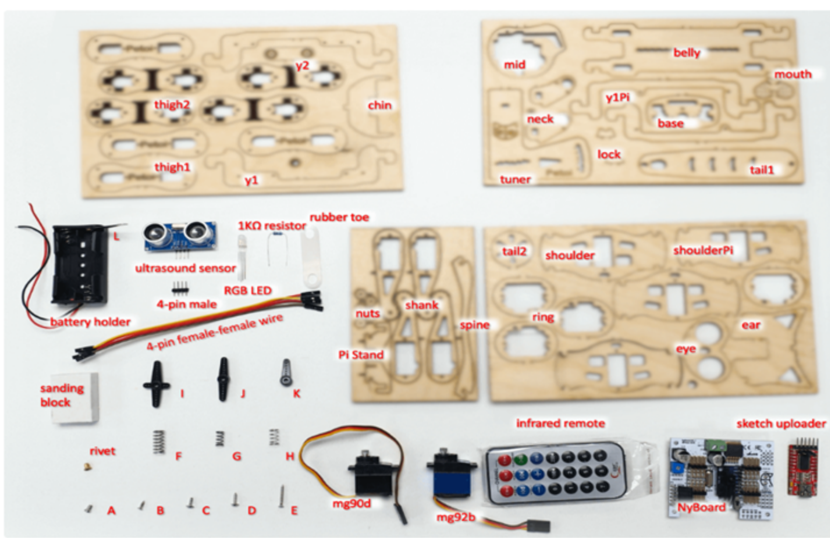


Figure 10: Hardware component

5.5 Sprint 2 Dynamic Aspect Design

As shown in figure 11, the owner issues a command to the cat, either via a voice command or a remote control, and these commands are sent via the Raspberry Pi to the board in order for the cat to take this action by sending a command to the motors to operate and carry out the required action.

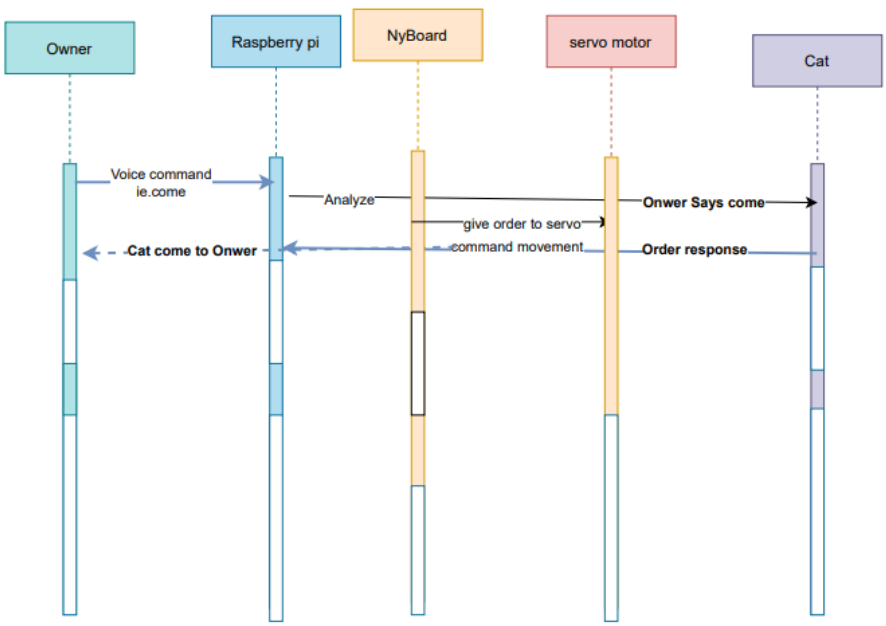


Figure 11: Sequence Diagram

5.6 Sprint 2 Implementation

After assembling and connecting the wires between Nyboard and servo motors on the Nybble cat, we should calibrate the servo motors. The calibration routine will send a calibration signal to the main board and rotate all the servo motors to known positions. Then we will attach the legs close to the reference posture (Figure **??**) and fine-tune the angles using the software.

There are three methods to calibrate Nybble cat:

- Use the mobile app Petoi.
- Use the Petoi Desktop App.
- Use the Arduino IDE.

Because of modifying Nybble cat, we will use Arduino IDE.

After we have done the calibration the Nybble cat is ready for use and control. There are 5 methods to control Nybble cat:

- Use the IR Remote Controller.
- Use the mobile app Petoi.
- Use the Petoi Desktop App.
- Use the Arduino IDE.
- Use Python.

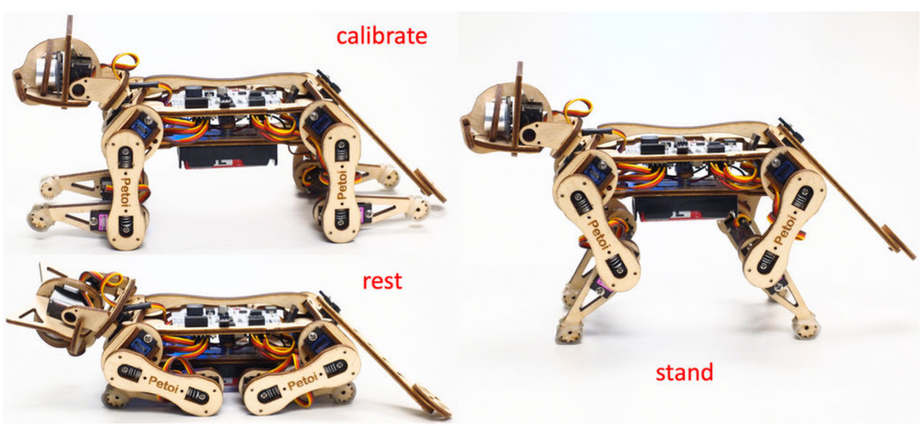


Figure 12: The rationale for calibration

5.7 Sprint 2 Testing and Evaluation

As the unit testing each leg is one unit so we did the calibration. Before Integrating the whole cat we checked each leg and servo motor separately and checked if it is responding or not by using Arduino IDE to send the commands to the Nyboard. After that, we Integrated and then checked all the legs, head, and tail together which are working very well.

5.8 Conclusion

The diagrams for the project were created, and this chapter covers how they were made as well as the coverage they provide. As a result, tracking the process from the use case diagram all the way up to the sprint is now much simpler for us. The Architectural Aspect Design determines the hardware needed for this project, the Static Aspect Design, Implementation, Testing, and Evaluation where we estimate scenarios to test and evaluate the project, and the Architectural Aspect Design, Testing, and Evaluation.

6 - Chapter 6 -

6.1 Introduction

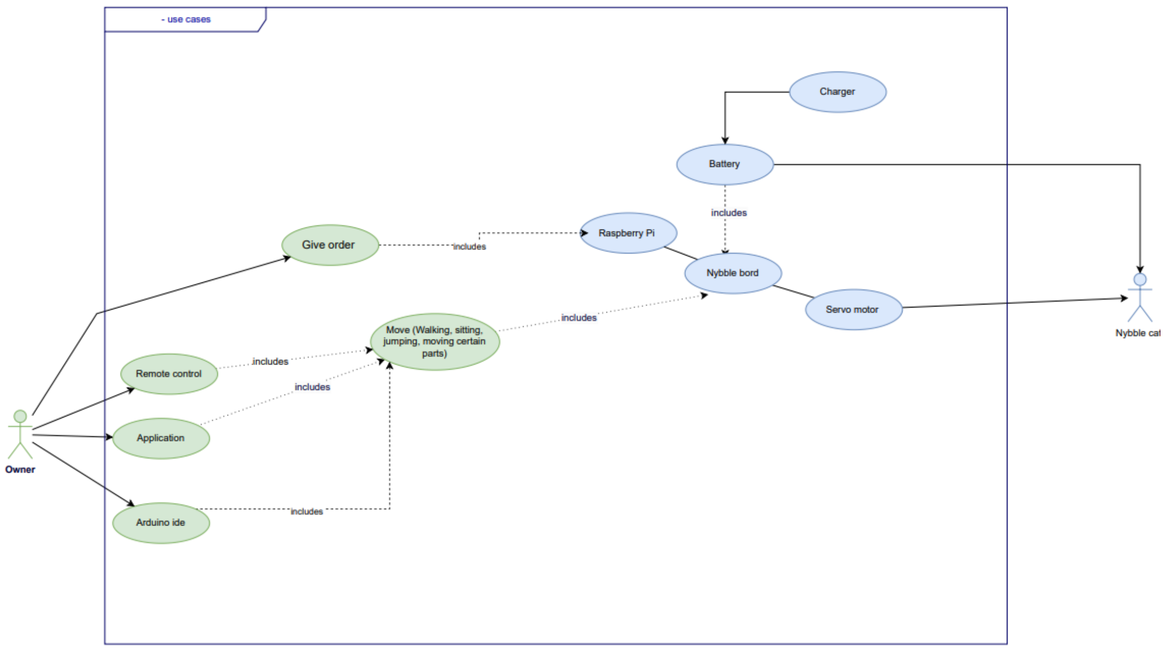
The topic of sprint 2 development will be the focus of the discussion that takes place in Chapter 6. These sections are the use-case diagram, in which we will create a use-case diagram to represent all of the advanced functional requirements; the sprint 2 static aspect design, in which we must create an E-R diagram to design all of the advanced functional requirements; the sprint 2 architectural aspect design, in which we must show the different hardware components of the advanced functional requirements; and the sprint 2 dynamic aspect design, in which we represent any suitable diagram or another detailed diagram to reflect the internal dynamic of the advanced functional requirements; the sprint 2 implementation; and the sprint 2 testing and evaluation will be implemented; the sprint 2 testing and evaluation will be implemented; the sprint 2 testing and evaluation will be implemented.

6.2 Use-case diagram

As shown in figure 13 the NyBoard can take commands comes from the remote control taking commands in detail so that you can move certain parts while the other parts are stopped and add some special movements and the matter applies to voice commands, but the commands must be defined first and then add voice recognition.

6.3 Sprint 3 Static Aspect Design

As shown in figure 14, the owner is in a relationship with the raspberry table pi is optional - many and the relationship between the raspberry pi and the owner Mandatory, so that there is more than one order issued by the owner, and There may not be, in addition to a relationship between The board and battery, indicating that they are mandatory for operation, and It's a one-to-one relationship. The board contains a Prime Key as well as a foreign key and the same thing applies to the rest of the tables and the relationships between them, in addition to two new tables, Control and Command, so that Control is linked to commands, and Commands are linked to Naybord, and the relationship between Control and Commands mandatory-one.



Use case

Figure 13: Use case diagram

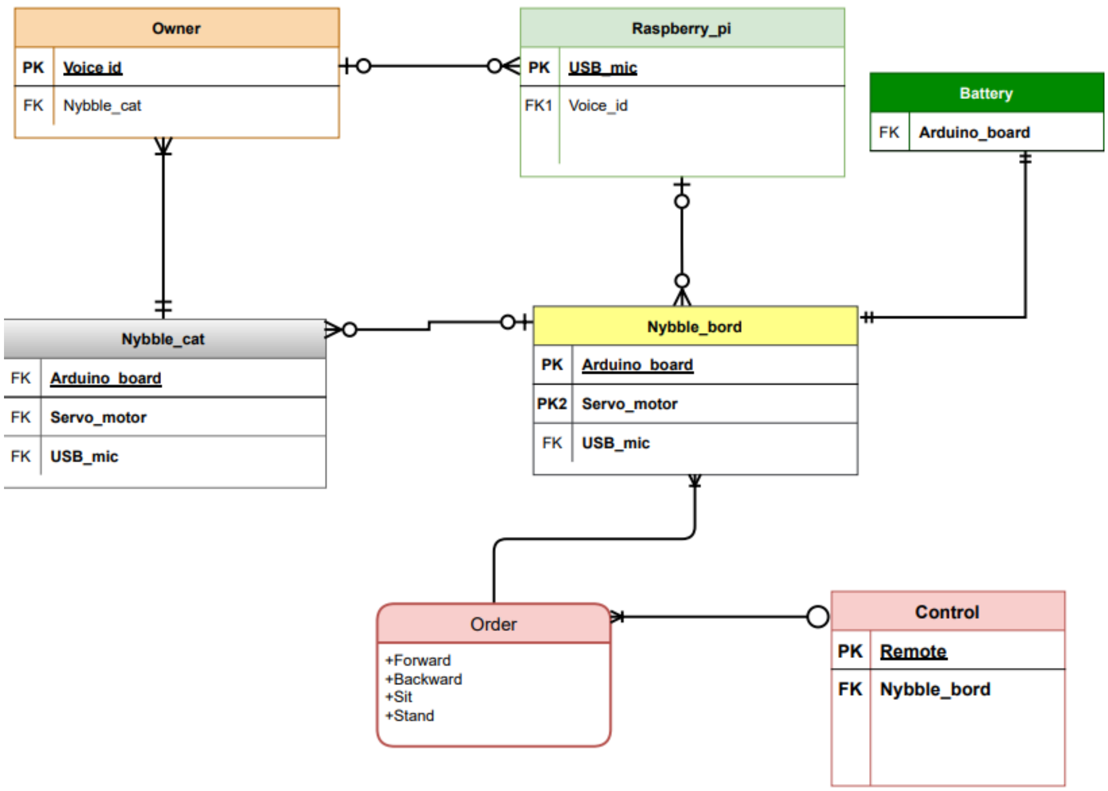


Figure 14: ERD diagram

6.4 Sprint 3 Architectural Aspect Design

In this section, we will know what is Raspberry Pi and USB microphone.

The Raspberry Pi is a low-cost, credit-card-sized computer that connects to a computer monitor or TV and operates with a regular keyboard and mouse. It is a competent tiny device that allows individuals of all ages to experiment with computing and learn to program in languages such as Scratch and Python. It can do everything a desktop computer does, from browsing the internet and watching high-definition videos to creating spreadsheets, word editing, and playing games.

A USB microphone is a microphone that is meant for use in a computer or laptop and makes use of the "plug and play" technology associated with USB ports. Devices that use USB technology are popular since all that is required to use them is to plug them into a USB port, after which the item is rapidly powered up and functioning. There are numerous sorts of these mics available, varying not only in price and quality but also in the number of capabilities they have.

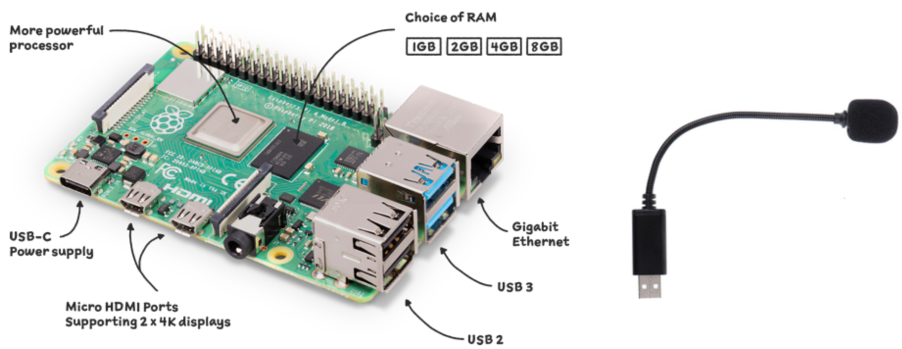


Figure 15: Raspberry pi USB mic

6.5 Sprint 3 Dynamic Aspect Design

As shown in figure 16. We use a flowchart diagram to describe the cycle of voice recognition to work. After the USB Mic detects the voice command the command would enter a main loop would go to analyze the command if the voice detected commands like (stand up). Then will analyze the command if it is valid or not. After that, the action will be executed.

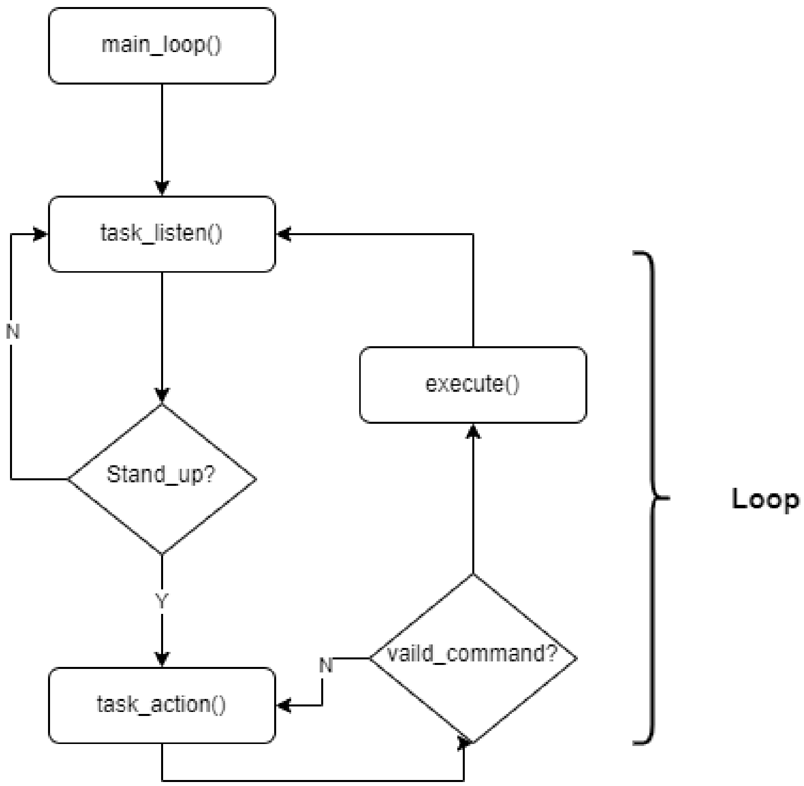


Figure 16: Flowchart diagram

6.6 Sprint 3 Implementation

We will be using a Raspberry Pi 4 and to get it working we need to do the following : Micro SD card - monitor/keyboard/mouse - monitor must have an HDMI input, keyboard and mouse must be USB. Then Install Raspberry Pi OS using Raspberry Pi Imager. This is the place to start. it is an easy installer for the OS, after that we will plug it into the power to install the OS and it will take time to install the system, then it will be ready to use. Then we will plug a USB microphone into Raspberry Pi for receiving voice. After that, we plug Raspberry Pi into Nyboard to control the cat's movement.

6.7 Sprint 3 Testing Evaluation

We used voice control, the first step, we add a Voice recognition feature that changes the voice of to robot into a listening text by using Raspberry Pi, that raspberry pi sends texts to Arduino and then does the required reaction, By directing the motors to do the action and that through his office numba and ibrosa and Ilvumlite . We built data libportaudio and the special voice for us stored because the cat uses the sound pickup to do the desired reaction by Analyzing the sound and comparing it with the data stored in it, and if there is a match between the commands and data, it does the action.

6.8 Conclusion

The project diagrams were generated, and this chapter discusses how they have been created as well as the coverage they provide. As a result, we can now trace the process from the use case diagram all the way up to the sprint lot more easily. The Architectural Aspect Design identifies the hardware required for this project, whereas the Static Aspect Design, Implementation, Testing, and Evaluation estimate scenarios to test and evaluate the project.

References

Nybble Cat - https://www.petoi.com/pages/nybble-cutest-bionic-robot-cat

Nybble Cat on githup - https://github.com/PetoiCamp/OpenCat

Petoi Robot Dog Bittle - https://www.petoi.com/pages/bittle-open-source-bionic-robot-dog

NyBoard - https://www.petoi.com/products/nyboard-customized-arduino-board

Problem Statement - https://www.tutorialspoint.com/arduino/arduino$_temperature_sensor.htm$

$ProblemStatement-ProposedSolutionhttps://github.com/AssemblyAI/youtube-tutorials$

$Whatisrobotics-https://www.techtarget.com/whatis/definition/robotics$

$IntroductiontoAI-https://builtin.com/artificial-intelligence$

$https://www.easytechjunkie.com/what-is-a-usb-microphone.htm$

$https://www.raspberrypi.org/help$