PDE3413 Systems Engineering for Robotics

Individual Project Draft Proposal Report

Automated Multi Pet Feeder (MPF)

2021 Oct Intake

Muddathir Muhammad Ibney Noorani Joomun

(M00830556)

Group B1

Date of Submission: 02/04/2024

Lab tutor Name:

Contents

1. Abstract

Robotic technology has become widely adopted across the world, and its applications have expanded into a wide range of industries. One area where robotics is poised to have a significant impact is social robots. These robots are designed to emulate human-like characteristics and are capable of interacting with their surroundings. One such application is in the development of an automatic pet feeder that can provide food and water for pets while their owners are away.

Many families have pets but also lead busy lives that involve travel, making it challenging to ensure their pets are fed and hydrated. Pet owners typically need to find someone to take care of their animals while they are away, which can be costly. To address this challenge, the Multi Pet Feeder (MPF) concept was developed to provide an automated solution to pet feeding.

The MPF is designed to utilize a variety of sensors, cameras, and motors to dispense food and reduce food waste. It is an improvement over existing pet feeders, which often fail to meet the needs of pet owners with multiple pets. The MPF's key objective is to autonomously feed various pets from a single robot while minimizing food waste.

The MPF is equipped with computer vision technology, which allows it to detect and recognize different pets. The camera identifies the presence of a pet and sends an image to the system for analysis. The system uses algorithms to recognize the pet and administer food accordingly.

One of the essential features of the MPF is its ability to provide portioned feedings to pets. This feature helps to maintain a pet's ideal weight and prevents overfeeding, which can lead to obesity and other health problems. Furthermore, the MPF allows pet owners to set up a customized feeding schedule for their pets through a mobile app, ensuring their pets are fed at specific times even when the owner is not around.

In summary, the MPF is an innovative and highly functional automatic pet feeder designed to provide a convenient and effective solution to pet owners with multiple pets. Its computer vision technology, portion control features, and customizable scheduling capabilities make it a highly attractive option for pet owners who lead busy lives and struggle to keep up with their pets' feeding requirements.

2. Introduction

A social robot is a physical-mechanical system that uses a variety of robotic techniques to interact with its surroundings.

They are created with the purpose of demonstrating human-like characteristics, such as hearing, seeing, and object identification using sensors and cameras. They may alter their movements based on the information they collect, such as hand gestures or vocal soundwaves.

One of the most recent developments of social robots is an automatic pet feeder. The latter is intended to aid pet owners with pet care while they are away from home.

The purpose of having sensors in a system like this is to automate the feed process completely with less/no human interference. (Jayaram, automatic pet feeder using internet of things,2019, pp.1)

Human engagement shall only occur when a refill is required or when the user physically operates the pet feeder.

2.1 Aims & Main Objectives of the Multi Pet Feeder (MPF)

2.1.1 Multiple animal detection (maximum 2)

When a pet approaches the robot, the MPF will be able to identify it using computer vision and administer food accordingly. Given that some individuals own numerous pets, this will be a smart addition.
The MPF can be set up to recognize two different animals, such as a dog and a cat, and it can also be used to identify a single animal if the owner has just one.

2.1.2 Pre-sets in terms of quantity

Approximately 55% of cats and dogs are overweight, which poses major health hazards such diabetes, kidney illness, heart, and lung disease. (McDowell, n.d)

The portioned feedings your pet needs are provided by an automatic feeder, which helps to maintain the right weight. The same will guarantee that the animals are fed adequately.

2.1.3 No earlier wake-up for Users

Many pet owners struggle with their pet waking them up in the early morning or late at night. When needed, MPF will feed the animal.

2.1.4 Food Available even in absence of Owners

We all lead busy lives, and sometimes our schedules prevent us from keeping up a regular feeding schedule for our animals. MPF provides the reassurance that your pet will be fed while you are away or at work. It will not be difficult to fit a late meeting into your schedule if you are confident that your pet will consume the same meals as always.

2.1.5 Computer Vision Technology

The MPF is equipped with computer vision technology that enables it to detect and recognize different pets. The camera detects the presence of a pet and sends an image to the system for analysis. The system uses algorithms to recognize the pet and administer food accordingly.

2.1.6 Portioned Feeding

The MPF has pre-set portion control features that ensure that your pet is fed the right amount of food. The owner can set up the desired amount of food, and the MPF will

dispense the exact amount needed. This helps to prevent overfeeding, which can lead to obesity and other health problems.

2.1.7 Customizable Scheduling

The MPF allows the owner to set up a customized feeding schedule for their pet. This feature ensures that the pet is fed at specific times, even when the owner is not around. The owner can set up the feeding schedule through a mobile app, which can be accessed remotely.

2.1.8 Easy to Clean

The MPF is designed for easy cleaning, with detachable parts that can be easily cleaned and reassembled. This feature ensures that the feeder is kept clean and hygienic, which is important for the pet's health.

2.1.9 Low Power Consumption

The MPF is designed with energy-efficient features that ensure low power consumption. The feeder is equipped with a rechargeable battery that can last for several days, ensuring that the pet is fed even when there is a power outage.

3. Market Size

The Automatic feeder market size was valued at $344.6 million in 2020 and is projected reach $998.7million by 2030, registering a CAGR of 8.0% from 2021 to 2030s

From the above figures and diagram there is an existing market for Pet Feeders and same is expected to grow over the year.

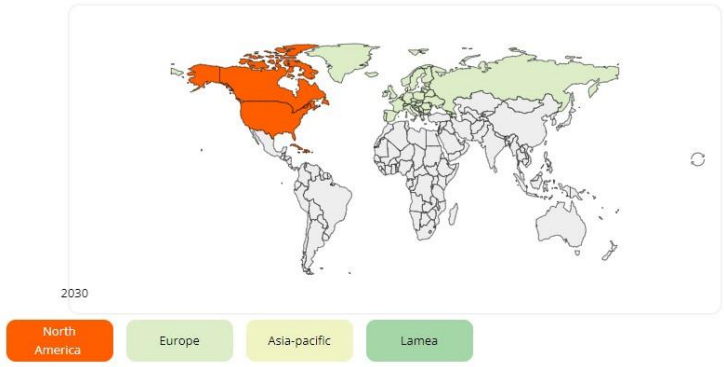

Figure 1. Market Sector (Alliedmarketresearch, Sept 2021)

3.1 Target Audience

The Pet Feeder is targeted for:

- Pet Owners who travel a lot • Pet with food disorder
- Pet Owners who have a rough time keeping up with the frequent food requirements of their pets.

3.2 How project will fit the market?

Seeing the increase in demand and that now several residences have multiple pet a new compact solution will be suited for the homes. The main selling point of this product is that it can be used for multiple pet.

4. Background Research

4.1 Review and short analysis-Existing Projects

The Microchip Pet Feeder (McPT) was made to recognize either the Sure Petcare RFID Collar Tag (one is included - one not) or the microchip that has previously been put in your pet allowing your cat to access their food while maintaining out other creatures.

Surepetcare has created a pet feeder as Figure 2.

Figure 2. Microchip Pet Feeder (Surepetcare, n.d)

4.2 Features and Structural of system

4.2.1 Feature Analysis 3.2.1 Stock Levels

This system lacks a stock level control that would alert the user when supplies are running low, and a refill is necessary.

4.2.1 Tags/Collars

Collar has a unique ID that is saved in the MPT's memory permanently. It can tell if a pet is on the pad of the pet feeder, which has both advantages and disadvantages.
It can tell if a pet is on the pad of the pet feeder, which has both advantages and disadvantages.
But as a drawback, it raises ethical and practical concerns. For instance, many proponents of animal rights oppose the wearing of collars, even though they serve an important purpose and that, given the nature of the animals we deal with, the collar tag is prone to loss or damage and thus increases user costs. Since it has a pricey and distinctive method of recognizing the collars, user will have to purchase one directly from the manufacturer.

4.2.1 Retrieving remaining food by pet for later use

Unfortunately, this pet feeder has no way to put remaining food back to the bowl which might lead to food wastage.

4.2.1 Identification of pet

The McPF does not include a method of identifying the animal because some households may have more than one cat or pet, for example, one household may have dogs and cats, necessitating the purchase of two separate devices, which will confuse the pet.

4.2.1 Food holder in metal

Because of the way the product is designed, it can hold both wet and dry food. Bowl capacity is 400 mL, which is equivalent to two pouches of wet food.

4.2.1 Battery powered

This product has a 6-month battery life, which is ideal if the user is away for an extended period and there is no power in the home.

4.2.2 Structural Analysis

4.2.2.1 Button Placement and function control

Pros:

1. A minimalist approach that is not tampered with by the pet.
2. Easily accessible from the side

Figure 3. Button Placement and interaction board (Surepetcare, n.d)

Cons:

1. There are not many options, and the interface is not very user-friendly.
2. The pet ID is permanently stored and must be returned to the factory for hard reset. There is no leeway.
3. There is no variety of options available, and the same is not aimed at people who are less familiar with technology.

4.2.2.2 Lid opening

Pros:

1. It is a better sanitary option because food does not spill everywhere.
2. Proper food control if you have multiple pets. Only pets with tags will be responded to by the lid.

Figure 4. Lid Mechanism (Surepetcare, n.d)

Cons

1. The mechanical lid should take some time to open, allowing the pet to leave.
2. The opening of the lid is determined by the tags. If two tags are too close together, a conflict will occur, and the machine will need to be restarted.

4.2.2.3 Movement of lid details

Pros:

1. The way the lid will use data from previous opening and closing to adapt to the pet, which is an adequate method of learning the animal to optimize optimal use.

Figure 5. Lid Adaptation (Surepetcare, n.d)

Cons

1. The initial adaptation will be difficult and unpleasant for the pet.

4.3 Review and short analysis-Existing Technologies

4.3.1 Contextual Info

Many animal species are still becoming extinct because of hunting, and governments can only do so much by enacting laws and conducting regular surveys. However, conducting a survey is a challenging task, especially without the assistance of technology.

This project's smart camera employs Python code with pre-trained TensorFlow models for forest monitoring and survey. The models assist us in matching and mapping the image in frame with the trained data of the model, and thus in identifying those animals.

Figure 1 shows images of elephants being detected using the author's prototype. Figure 2 depicts the author's prototype, which employs a Raspberry Pi (or RPi) with a camera module.

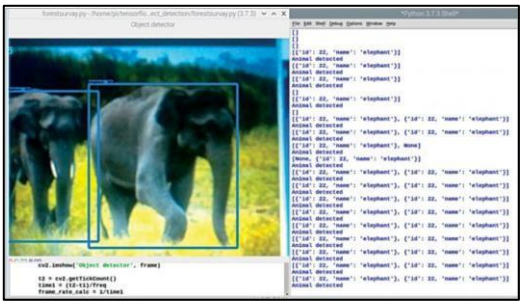

Figure 6. Elephant being captured by prototype (Kumar Sinha, Sept 2021)

Figure 7. Pi Camera for Survey (Kumar Sinha, Sept 2021)

4.3.2 Technological Analysis + Costing for Pi Camera Implementation

The use of Raspberry Pi with a camera for animal species recognition is a cost-effective solution that can be implemented for various applications. The cost of the full prototype at INR 5030 or approximately Mur 2515 makes it an affordable option compared to other expensive animal recognition systems. However, it is important to note that the accuracy of the system depends on the quality and quantity of existing data used to train the detection model. Therefore, the system may require periodic updates to improve its accuracy.

In terms of the application to a pet feeder, implementing this technology can help the feeder to recognize different animals and adjust feeding portions accordingly. This can help to prevent overfeeding and ensure that each pet is receiving the correct amount of food. Additionally, the system can be integrated with other features such as customized scheduling and mobile app control to provide a comprehensive pet feeding solution for pet owners.

Bill of Materials — Table 1

Component name	Quantity	Description	Approx. Cost (₹)
Raspberry Pi 4/Zero W	1	Any version of RPi	4,500
RPi camera	1	Camera module	300
HDMI/RCA cable	1	For video output	200
Wires	Depends	For connection	30
Total cost			**5,030**

Figure 8. Costing Table (Kumar Sinha, Sept 2021)

Overall, the use of Raspberry Pi with a camera and OpenCV for animal recognition is a promising technology that can be applied to various applications, including pet feeders. While the system may require periodic updates to maintain its accuracy, its cost-effectiveness and low maintenance make it an attractive option for businesses looking to implement animal recognition technology.

5. Proposal

5.1 Description and concept behaviour.

A Multi-Pet Automated Feeder that distributes and monitors food. The same will be able to store food and identify which pet is visiting its base and will provide appropriate food. It will also have an interface for human to react directly on it. If food is left over after a while, it must be returned to the food storage. This MPF must also have a water dispenser. The concept behaviour will be in details below. (Refer Section 4.2.2 Flow of System)

The Multi-Pet Automated Feeder will have the following flow of system:

1. Pet detection: The MPF will use computer vision technology to detect and recognize pets. When a pet approaches the feeder, the camera will detect the pet and send an image to the system for analysis.
2. Pet identification: The system will use algorithms to identify which pet is approaching the feeder. Based on the identification, the system will dispense the appropriate food from its storage container
3. Food dispensing: The MPF will dispense the appropriate amount of food for each pet based on the pre-set portion control feature. The pet owner can set up the desired amount of food through the mobile app or interface.
4. Water dispenser: The MPF will also have a water dispenser that will provide fresh water to pets as needed. The water dispenser will be connected to a water source or container and will be easily accessible for pets.
5. Interface for human interaction: The MPF will have an interface that allows pet owners to interact with the system directly. Through the interface, the owner can monitor the feeding schedule, adjust the portion sizes, and receive notifications when the food or water is dispensed.
6. Cleaning and maintenance: The MPF will be designed for easy cleaning and maintenance, with detachable parts that can be easily cleaned and reassembled.

5.2 Review and short analysis-Proposed projects

5.2.1 System Architecture

The Multi-Pet Automated Feeder system architecture will consist of the following components:

1. Raspberry Pi with Camera: The Raspberry Pi with a camera will be used for pet detection and identification using computer vision technology.
2. Food Storage Container: The food storage container will store the food and dispense the appropriate amount based on the identification of each pet.
3. Water Dispenser: The water dispenser will be connected to a water source or container and will provide fresh water to pets as needed.
4. Interface: The interface will allow pet owners to interact with the system directly, monitor the feeding schedule, adjust the portion sizes, and receive notifications.

The System architecture connection shall be as per diagram 9. More details is be given in a flow chart and the flow of data shall be given in full detail refer to Section 4.2.2.

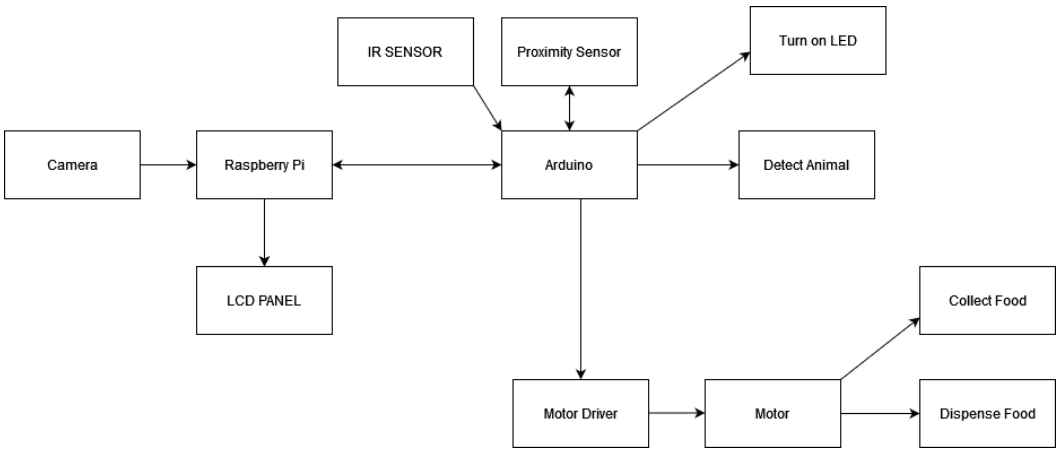

Figure 9. Block Diagram

5.2.2 Flow of System

The flow for the Multi-Pet Automated Feeder system will consist of the following steps:

1. Pet detection: When a pet approaches the feeder, the camera will detect the pet and send an image to the system for analysis.
2. Pet identification: The system will use algorithms to identify which pet is approaching the feeder based on the image received.
3. Food dispensing: The MPF will dispense the appropriate amount of food for each pet based on the pre-set portion control feature.
4. Water dispensing: The water dispenser will provide fresh water to pets as needed.
5. Human interaction: The interface will allow pet owners to interact with the system directly, monitor the feeding schedule, adjust the portion sizes, and receive notifications.
6. Cleaning and maintenance: The MPF will be designed for easy cleaning and maintenance, with detachable parts that can be easily cleaned and reassembled.

The flow of data in the system will follow the same steps as above, with data being collected from the camera and analyzed by the system to identify and dispense food and water accordingly. The interface will allow for data to be monitored and controlled by the pet owner, while the cleaning and maintenance features will ensure that the system remains hygienic and functional.

5.2.2 Flow of the System

The sensors will interact between each other and with the Arduino & raspberry pi to generate action exactly like human using sensors to the brain. The flowchart below will give more detail about the flow of data and how sensors react between each other

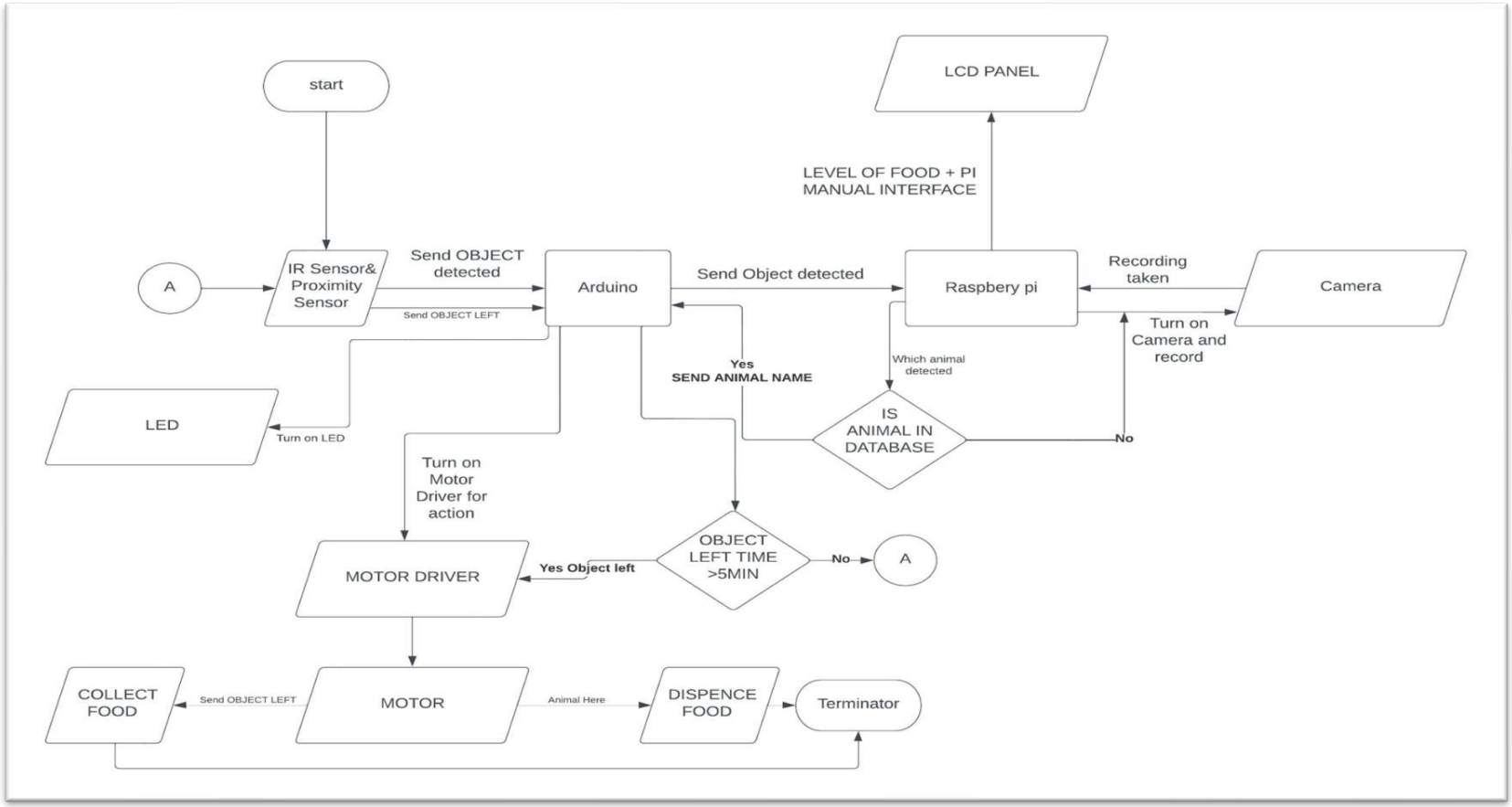

Figure 10. Flowchart

6. System Breakdown

To create an effective robot, we must first divide the system into smaller pieces and investigate its technological needs, which will serve as a guide for testing the system and identifying potential flaws.

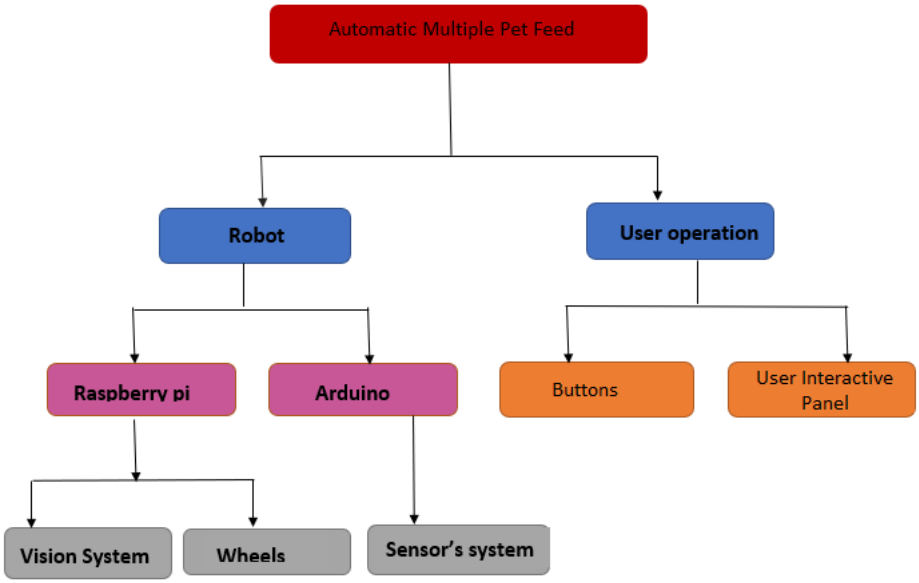

Figure 11. System Breakdown

Each component of the system will have its technological needs and requirements, such as the camera resolution and processing power needed for pet detection and identification, the portion control mechanism for food storage and dispensing, and the mobile app's compatibility and security requirements. Testing each component separately and together will help to identify any potential flaws and improve the system's overall performance.

In summary, breaking down the Multi-Pet Automated Feeder system into smaller components and investigating its technological needs is essential for developing an effective robot. It will help to ensure that each component performs its intended function and work together seamlessly to provide a convenient and user-friendly experience for pet owners.

7. System Requirements

General Requirements

General requirements	
ID	Requirement
R01	The system should provide food to the animal in front of it
R02	The system should be easy to interact
R03	The system must be user friendly.
R04	The system should be lightweight for user to place it and plug in into ac supply
R05	The system's components should be easily accessible for maintenance.
R06	The system should be ready by latest March.

Figure 12. General Requirements

Robotic System		
Sub Systems	**ID**	Requirement
Raspberry Pi	Rasp01	The system should be powered by AC
	Rasp02	The raspberry pi should communicate with the camera to allow for vision of Object coming close to the MPF
	Rasp03	The Raspberry Pi will send signal to the Servor Motors & Water Oump
	Rasp04	The Raspberry Pi Should be the relay between the camera and the sensors
	Rasp05	The servor Motor & DC motors gets instruction for the Raspberry Pi to Open Valve to dispense food and move the cart
Arduino	A01	The system should be powered by AC as the raspberry Pi
	A02	A full check is done upon start to check if all sensors are good and operation
	A03	The Ultrasonic sensor,IR sensor,Servo Motors& DC motors connect to the arduino
Vision	V01	The Vision of the robot consist of Camera
	V02	The camera is made sure that its pointed toward the entrance of the Pet Feeder
	V03	The Camera Will start recording as soon as IR & Ultrasonic Sensor Detect Object Near the Robot
	V04	OpenCV is used to analyse the data to check if it's an animal or not
	V05	The camera operation of turning on and off will be autonomous
Pulley	P01	The pulley will be connected to a motor together with a belt which is connected on both side of the food moving storage internally
	P02	The Pulley needs to be position such that it is horizontal with the centre of mass of the cart
	P03	The Pulley rotation need to stop when food container is reached for delivery and resume when delivery is done to move the food kart
Sensor System	S01	All Sensors need to be properly positioned on the robot
	S02	All the motors need to be properly connected to the arduino with its driver to be able to be powered on
	S03	The Sensors should be configured to read appropriate data and react based on input data
	S04	Data read for sensors need to be communicated to respective part of the robot as shown in the flow chart

Figure 13. Robotic System & Sub Systems

User Operations		
Mode	**ID**	Requirement
User Interactive Panel	UIP01	The UIP will be implemented using simple ROS for user to interact
	UIP02	The Panel should provide an interface for pre-set amount of food to be delivered
	UIP03	The Panel should provide user with current stock and level
	UIP04	The panel should give warning message if there is any error
	UIP05	The Panel also will provide the user with a manual feed mode
	UIP06	It should provide an option to select which animals are present in the house for easy processing of data by the robot

Figure 14. User Operations Table

8. Component for data collection and action creation

All social robot-like human- have different sensors to collect data from the environment and process it to do an action.

You shall find the sensor & component list below which will be required for the proposed project.

• Object Detection

1. **Proximity Sensors (Sharp GP2Y0A41SK0F)**
A PSD (position sensitive detector), an IRLED (infrared emitting diode), and a signal processing circuit are merged to form the distance measurement sensor unit. The adoption of the triangulation approach

prevents the diversity of the object's reflectivity, the ambient temperature, and the operation length from having a significant impact on the distance detection. The voltage output by this gadget is determined by the detection distance.

Figure 15.Analog Distance Sensor (Pololu, 2022)

Same will be used to detect any incoming object
In Short, the Distance is directly proportional to the voltage output by the sensor

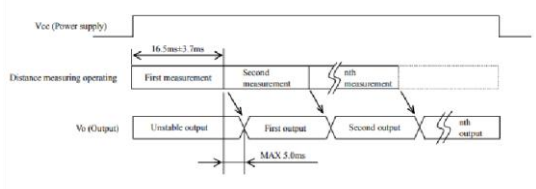

Figure 16. Timing Chart (Global Sharp, 2022)

Operating Voltage of 4.5 to 5.5 V
Minimum Time on to get 1st Output time: 16.5ms±3.7ms(refer to Figure 12).

• Machine Vision

2. **Raspberry Pi Camera Module 2** The 8-megapixel Sony IMX219 sensor in the v2 Camera Module (compared to the 5-megapixel OmniVision OV5647 sensor of the original camera).
Both still images and high-definition video can be captured using the Camera Module 2.
Same used to capture the object detected and will be sent to the raspberry pi for further processing
Recommended Input Voltage 5V and Input Current 2A

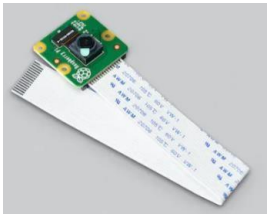

Figure 17. Pi Camera Module 2 (Raspberry Pi Foundation, 2022)

• Water Level Control

3. Water Level Sensor ADV00089
An inexpensive, simple-to-use water level recognition sensor, the water level sensor is made by having several exposed parallel wire traces that measure the volume of the water droplets.

The sensor will provide analogue values straight to the analogue input pins on your Arduino board, where they may be read and processed further.
Ideal for any upcoming project using a water level alarm.
3.3V to 5V is the operating voltage and less than 20mA.

Figure 18. Pi Camera Module 2 (Raspberry Pi Foundation, 2022)

As Figure 15 shows the Water
Level is Indirectly proportional to the resistance.
This will be our control for us to check the level of water

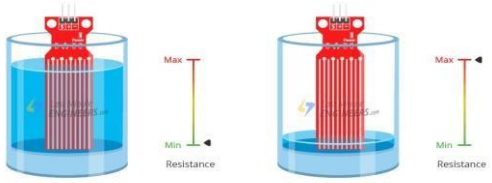

Figure 19. Water Level Sensor in scenario (Lastminuteengineers, n.d)

9. Feature Analysis

8.1 Detection & Identification of Pet

As soon as pet come close to the sensor the Robot will detect the animal thus starting the identification process. Computer vision will be implemented to Identify which pet shall be used using camera. The Project use camera vision linked to capture the face& body of the pet and shall compared that to a database as explained above in Section 1.1.1. Base on the outcome the respected food shall be delivered

8.2 Selection of Food

After pet has been identified a plate shall be move on the x plane to collect proper food. (More details in Structural Analysis). After selection of food, the plate shall move on the Y plane down next to the out-source to the bowl. (More details in Structural Analysis). When the plate come to the outside it shall be tiled by a motor to empty the plate into the food bowl. After that same shall be reset to initial position.

8.3 Level of Food in Stock

Stock shall be monitored like a being in low stock same shall light a red bulb next to the slot for the corresponding food. (An on-board interface shall be implemented depending on the time frame).User will have to interact with it and refill same

8.4 Food storage

Food will be stored internally to avoid contamination of food and risk of heal issue of the pet

8.5 Control Panel

The MPF shall have a control panel where user will be able to interact and have full control on the system manually.

10 Structural Analysis

10.1 Outside structure

The MPF Shall be a box like structure. Below you shall find the plan of the Robot.

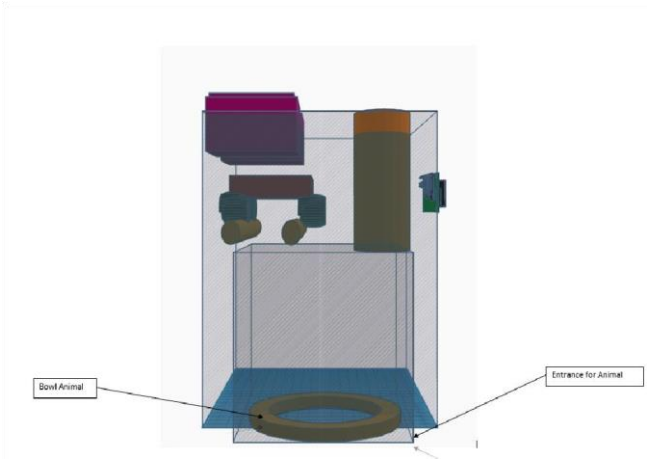

Figure 20. Front View of MPF

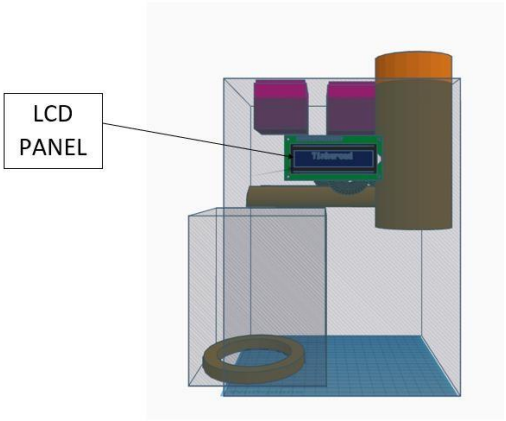

Figure 21. Side View of MPF

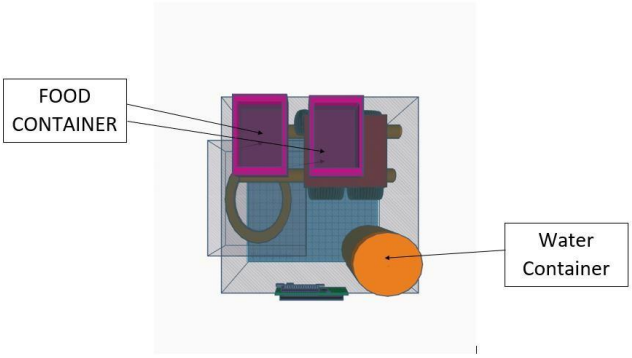

Figure 22. Top View of MPF

Proximity sensor & IR shall be placed on 4 sides of the MPF so as it can detect the animal coming from all areas. The Pi Camera will be placed only on the Front View since the bowl of food is there thus the MPF will dispense food when the animal is in front of the ball. LED will be in the small partition where the food will be given to create a cosy environment for the animal.

10.2 Sensor Placement & Component Placement

7.2.1 IR & Ultrasonic Sensor

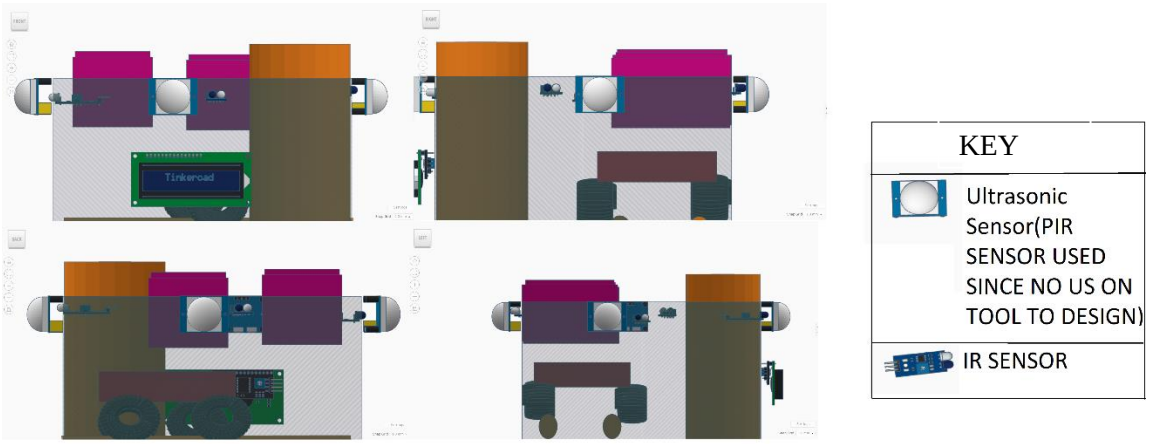

Figure 23 Side View of movement system

10.1.1 Movement of plate for food dispense

The plate shall be place on wheels. Same structure will be put on two rods and a chain shall be attached to both end of structure.

For the movement of the plate same shall act as a pulley system to move right and left (x-axis) refer to Figure 19

For the Movement up and down system shall be same as x-axis put just on y-axis

The pulley will work with a motor. Refer to Figure 19

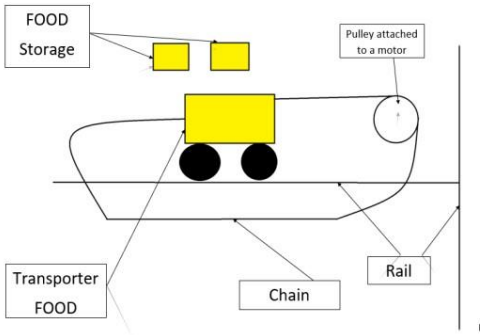

Figure 24. Side View of movement system

11. Bill of Materials(Proposed Costing)

Some projects might turn out costly when all cost is taken into considerations. Below you shall see a breakdown of items require

Qty	Component	Model No.	Price(Rs) incl VAT
Already Bought			
1	Raspbery PI	R-PI4-4	9250
1	MicroSd Card	KI0-MSD16	240
1	Water Level Sensor	A-WLS	35
6	Obstacle Sensor PhotoElectric	KIT-FC51	45
4	Ultrasonic Distance Measuring	HC-SR04	60
1	ABS Plastic Enclosure Rasp Pi	A-ACRY PI4	150
1	Submarine Pump	SUBMP10 85	85
40	Dupont Wire Female Female	DPWF-F20	2
1	LCD Touch Screen 5 Inch	A-LCD5	1600
1	Raspberry PI4 Camera	R-PI-CAM	250
1	Arduino Uno	ARD	500
1	Arduino Sheild	-	450
		Supplied by Transcom Total	12667
Not Bought Yet			
8	Acrylic Pane	-	3200
4	Metallic Rode	-	400
		Supplied by General Store Total	3600
		Gross Total	16267

Figure 25. Component Bills

12. Testing the System

12.1. Component Testing

Proximity Sensors (Sharp GP2Y0A41SK0F)

For this test we shall require 1 Arduino Uno. Sharp Proximity Sensor measuring tape and three breadboard cables. The test consists of setting predefined distance from sensor to object and check if the sensor can detect the object at the pre-defined distance. The circuit will be as below.

If the object can detect the object at predefined distance the sensor is working properly.

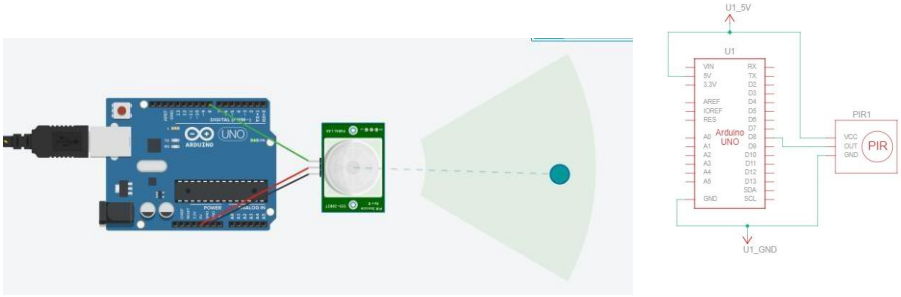

Figure 26. Connection Arduino with Sensor & Pin Out diagram

Raspberry Pi Camera Module 2

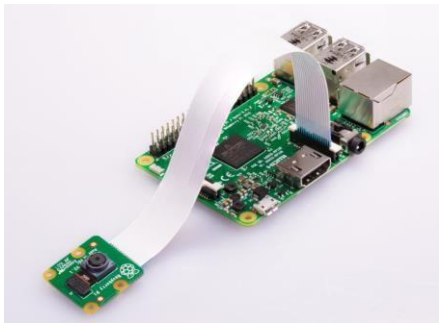

Figure 27. Connection Piboard with PiCam(Rasberrypi Foundation,2022)

For this test we will require The Raspberry Pi board, Raspberry Pi Camera Module 2 and object. We can test if it can identify object using OpenCV sample test

We will have to connect the Pi Camera to the Pi Board as shown in Figure 20. In the System configuration of the PI board, we will have to enable the camera.

If we get an output image the camera is working with proper identification the test is successful

Motor Testing

For this test we will require a DC motor together with an Arduino. This will just test If the motor is working. You can also vary the amount of Power supply to it to see if it has a good speed range since calibrating on the rod/rail require precision

For the Arduino to run the motor a motor driver is Required Model L293D

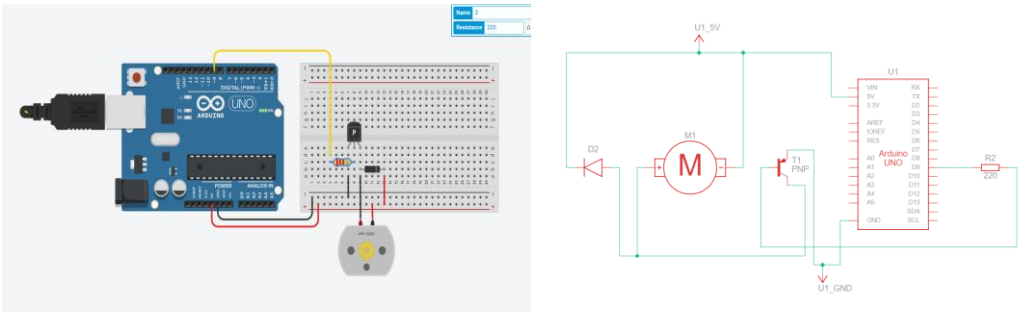

Figure 28. Motor Connection and Pin Out Diagram

Ultrasonic Sensor

For this test we will require a an ultrasonic sensortogether with an Arduino. The sensor can be moved to test if it's taking the data correctly

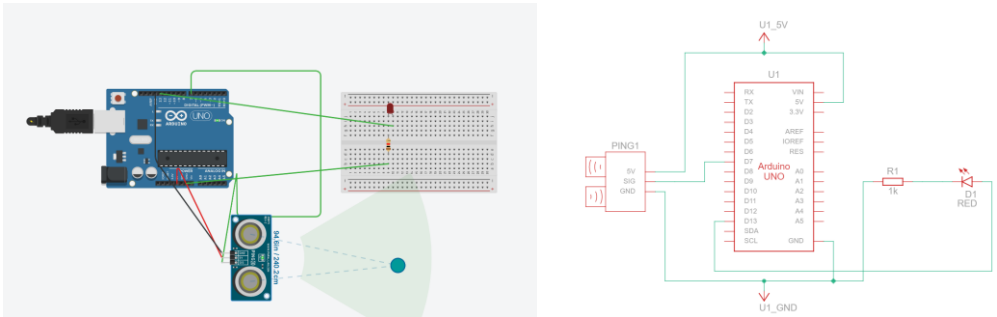

Figure 29. Ultrasonic Sensor and Pin Out Diagram

Below you shall find a snippet of code for the testing phase.

```
/* Ping))) Sensor

   This sketch reads a PING))) ultrasonic rangefinder and returns the
   distance to the closest object in range. To do this, it sends a pulse
   to the sensor to initiate a reading, then listens for a pulse
   to return.  The length of the returning pulse is proportional to
   the distance of the object from the sensor.

   The circuit:
    * +V connection of the PING))) attached to +5V
    * GND connection of the PING))) attached to ground
    * SIG connection of the PING))) attached to digital pin 7

   http://www.arduino.cc/en/Tutorial/Ping

   created 3 Nov 2008
   by David A. Mellis
   modified 30 Aug 2011
   by Tom Igoe

   This example code is in the public domain.

 */

// this constant won't change.  It's the pin number
// of the sensor's output:
const int pingPin = 7;
const int ledPin = 13;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters:
  long duration, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING))): a HIGH
  // pulse whose duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  // convert the time into a distance
  cm = microsecondsToCentimeters(duration);

  // Print the distance
  Serial.print("Distance: ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  // Turn on the LED if the object is too close:
  if(cm < 100) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }

  delay(100);
}


long microsecondsToCentimeters(long microseconds) {
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}
```

Figure 30. Source Code for Testing

13. System Testing

- The robot must be able to use the proximity sensor to detect the pet and start turn on the machine. Same need to be calibrated
- The robot must be able to capture the photo and process it to give the correct food. The raspberry pi should give the correct answer.
- Calibration is required for the rod & the food transporter so as it moves exactly under the boxes and move exactly into the bow
- Water container should be tested for any leaks since we are dealing with electronic component.
- Software testing shall be done on all the whole system and the inference system also need to be tested and see if all required function is working correctly.
- The robot shall be able to resist small bumps and crashes since we are dealing with a pet and some pet can be aggressive.
- Do a full testing with all the component together and see if the sensors are giving the proper data.
- The output of the whole system shall be as below
- Proper food and amount of food delivered
- Proper Water Level display
- Proper Alert Sign when low in stock
- Proper timing to collect the food back in the storage section

14. Integration of Project

14.1 Building the physical Architecture.

When building the MPF the design changed from Proposed design due to functionality and implementation issue. Below is the New Design of the MPF

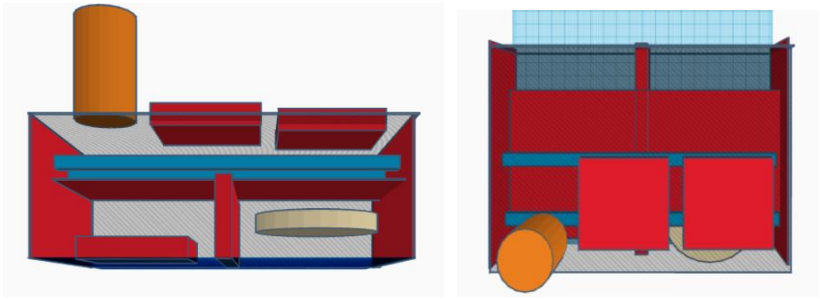

Figure 31. Front and Top View

The 1st step was to cut the acrylic sheets based on the dimension cut out.

Figure 31. Cutting of Acrylic sheets

The 2nd Step was to assemble the whole Physical Part using Glue Gun. Below is the front View and Side view of the MPF

Figure 31. Front View of Water Artifact

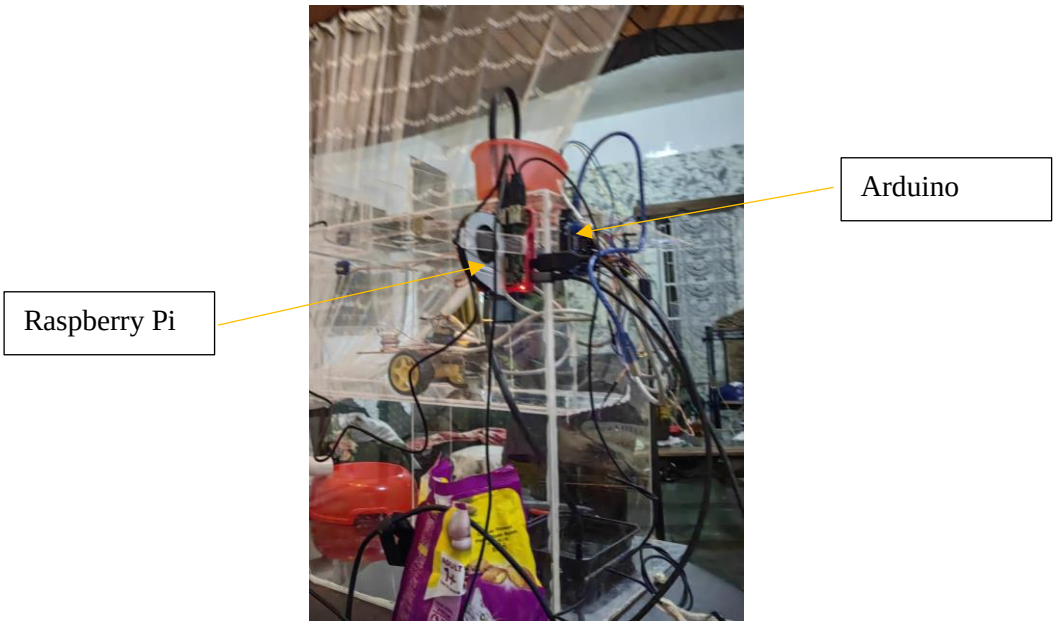

Figure 31. Side View of the Artifact

14.2 Building the Food Truck.

The Food truck will be place in Artifact. Glue gun as use to fix several parts on the Truck Same consist of 2 DC motors, 1 Ode-metric Sensor,1 Driver ,1 battery Pack, servo & a platform.

Below you shall have a detail view of the side and bottom view of the food-truck.

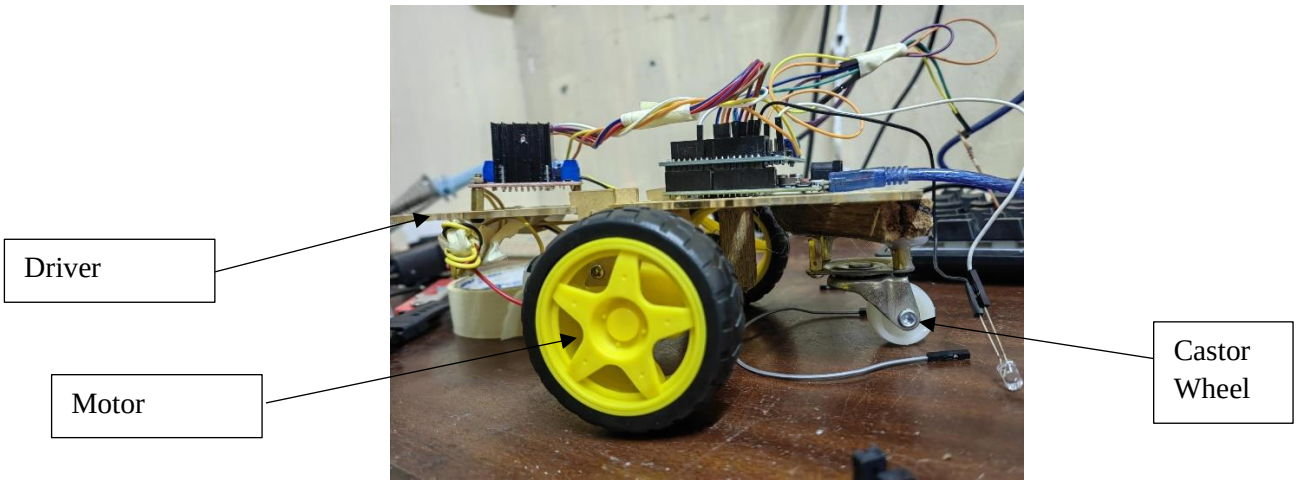

Figure 31. Side View of truck

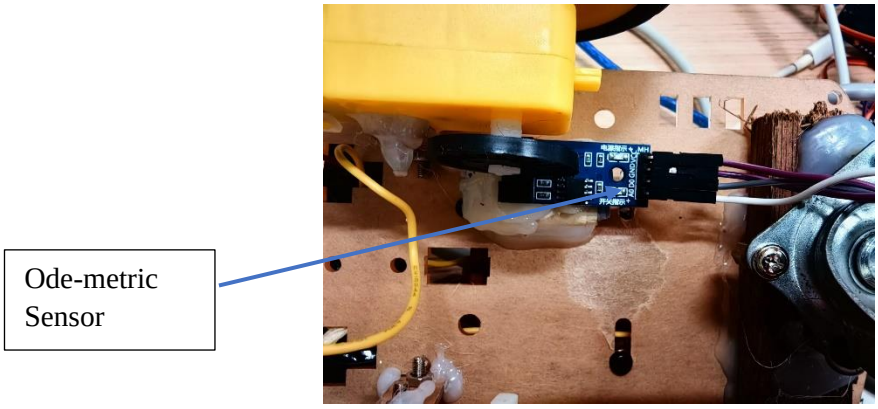

Figure 31. Bottom View of truck

14.3 System Configuration.

Pin Layout Arduino

Below is a tabulated pin layout for the Arduino Uno, L298N motor driver, servo motor, water level sensor, and distance sensor. Same is required to establish proper connections between the driver and the Arduino which will be the brain of the sensors.

Component	Pin / Terminal	Connected to	Arduino Pin
Water Level Sensor	VCC	5V	-
	GND	GND	-
	Signal	-	A0
Distance Sensor	VCC	5V	-
	GND	GND	-
	Signal	-	3
L298N Motor Driver	ENA	-	5
	IN1	-	6
	IN2	-	7
	ENB	-	10
	IN3	-	9
	IN4	-	8
	OUT1	Motor 1 A	-
	OUT2	Motor 1 B	-
	OUT3	Motor 2 A	-
	OUT4	Motor 2 B	-
Servo Motor	VCC	5V	-
	GND	GND	-
	Signal	-	13

Figure 31. Pin Layout

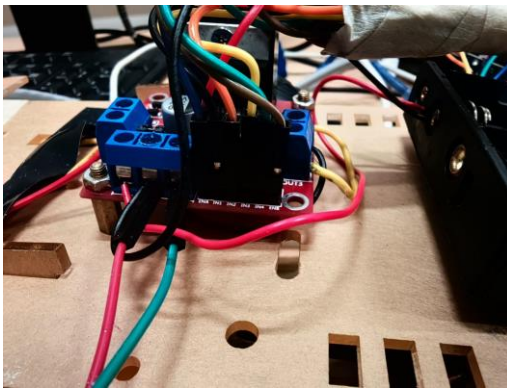

Figure 31. Motor Driver Pin Layout

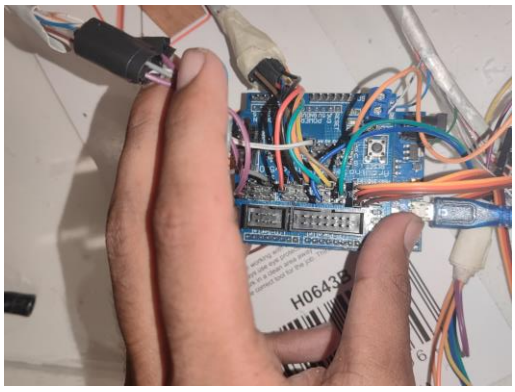

Figure 31. Arduino Pin Lay out

This table shows the connections between each component and the Arduino Uno pins, as well as power and ground connections. Use this table as a reference when setting up your project on a breadboard or Arduino shield.

Code Snippets based on Features- Ardunio

1. Libraries & Pin declarations & Variable declarations:: The code includes the Servo.h library to control a servo motor.
   The code declares pins for various purposes such as sensor input, motor control, and servo control. The code declares variables including a volatile distance variable that will be modified in an interrupt service routine.

```
#include <Servo.h>
#define SENSOR_PIN A0

volatile long distance = 0; // distance counter, declared as volatile because it is modified in an interrupt
const int ENA = 5; // PWM pin for motor 1 speed
const int IN1 = 6; // Control pin 1 for motor 1 direction
const int IN2 = 7; // Control pin 2 for motor 1 direction
const int ENB = 10; // PWM pin for motor 2 speed
const int IN3 = 9; // Control pin 1 for motor 2 direction
const int IN4 = 8; // Control pin 2 for motor 2 direction
const int servoPin = 13;

Servo servo;
```

Figure 31. Code Snippet-1

2. Setup function: This function is called once when the program starts and sets the pinMode for all pins, attaches an interrupt to a specified digital pin, attaches the servo to its control pin, and initializes the serial communication.

```
Servo servo;
void setup() {

  pinMode(13, OUTPUT); // Set pin 13 as an output
  pinMode(8, INPUT); // Set pin 8 as an input

  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  pinMode(servoPin, OUTPUT);
  pinMode(SENSOR_PIN, INPUT);
  attachInterrupt(digitalPinToInterrupt(3), incrementDistance, RISING); // attach interrupt to digital pin 0
  servo.attach(servoPin);
  Serial.begin(19200);
}
```

Figure 31. Code Snippet-2

3. Loop function: This function is called repeatedly during the program execution. The loop reads the signal from the serial port and performs actions based on the received signal. It also moves the motors, servo, and sensors and prints the readings to the serial monitor.

```
void loop() {
  if (Serial.available() > 0) {
      char signal = Serial.read();
      if (signal=='1'){// If '1' is received on pin 8
        digitalWrite(13, HIGH); // Turn on the built-in LED on pin 13
        Serial.print("CAT DETECTED:");
        moveMotors(250, 250, true, true);
        delay(5000); // Move both motors forward

      }else { // If '0' is received on pin 8
          digitalWrite(13, LOW); // Turn off the built-in LED on pin 13
          Serial.print("CAT NOT DETECTED:");
          stopMotors(); // Stop both motors
      }
  }
  moveMotors(200,200,true,true);
  if (distance==200){
      stopMotors();
      moveServo(100, 5000);
      delay(5000); // Wait for 1 second before moving the servo back
      moveServo(-100, 0); // Move the servo back to -90 degrees
      moveMotors(200,200,true,true);
      if (distance == 500){
      stopMotors();
      moveBackward(200);     }
  }
  int level = analogRead(SENSOR_PIN);
}
```

Figure 31. Code Snippet-3

4. incrementDistance function: This function is an interrupt service routine that increments the distance variable when the interrupt is triggered by a rising edge on a specified digital pin.

```
// Interrupt service routine to increment the distance counter
void incrementDistance() {
  distance++;
}
```

Figure 31. Code Snippet-4

5. moveMotors function: This function moves two motors in the specified directions and at the specified speeds using PWM control.

```
// Function to move both motors
void moveMotors(int speed1, int speed2, bool forward1, bool forward2) {
  // Set motor 1 direction
  if (forward1) {
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
  } else {
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
  }

  // Set motor 2 direction
  if (forward2) {
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
  } else {
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
  }

  // Set motor speeds
  analogWrite(ENA, speed1);
  analogWrite(ENB, speed2);
}
```

Figure 31. Code Snippet-5

6. stopMotors function: This function stops both motors by setting the control pins to LOW and PWM signals to 0.

```
// Function to stop both motors
void stopMotors() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW);
  analogWrite(ENA, 0);
  analogWrite(ENB, 0);
}
```

Figure 31. Code Snippet-6

7. moveServo function: This function moves a servo to a specified angle and for a specified duration using the servo.write() function.

```
void moveServo(int angle, unsigned long duration) {
  unsigned long startTime = millis();
  servo.write(angle);

}
```

Figure 31. Code Snippet-7

8. moveBackward function: This function moves both motors backward by setting one motor in forward direction and the other in reverse direction at the specified speed.

```
void moveBackward(int speed) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);
  analogWrite(ENA, speed);
  analogWrite(ENB, speed);
}
```

Figure 31. Code Snippet-8

Layout Raspberry PI

For the raspberry Pi connection is simple. Since we shall ne suing the USB Camera & Serial Connection to transfer data to the Arduino we just connect the data cable & the usb cable from both Arduino and Camera to USB ports of the Raspberry Pi

The Raspberry Pi is mainly for computer vision. We used OpenCV together with pre install model haarcascades to Enable same

Code Snippets

1. Import of Libraries
   Below is a list of libraries we used for this project.

Libraries	Function
cv2	To use OpenCV for camera vision
threading	Since we are using raspberry pi we need to separate the load on separate threads
time & numpy	this is used to record the time a pet has come to the MPF such that we can create a delay for the next serving
serial	to enable communication between the Pi and the Arduino
tkinter	to provide and interface for user to interact

Figure 31. Table of Libraires

```
import cv2
import threading
import numpy as np
import time
import serial
import tkinter as tk
```

Figure 31. Code Snippet Pi-1

2. Code for Camera Vision, time delay & interface

```
import cv2
import threading
import numpy as np
import time
import serial
import tkinter as tk


# Set up the serial connection to the Arduino
ser = serial.Serial('/dev/ttyUSB0', 19200)

# Load the Haar cascade for cat detection
cat_cascade =
cv2.CascadeClassifier('/home/joomun/Desktop/opencv/da
ta/haarcascades/haarcascade_frontalcatface_extended.xml
')


# Initialize the camera
camera = cv2.VideoCapture(0)

# Define a flag for indicating if a frame has been captured
frame_captured = False

# Define a lock for accessing the captured frame
lock = threading.Lock()

# Define a flag for indicating if the program should quit
quit_program = False

# Define a variable for storing the interval for last detection
interval = 2 * 60 * 60  # default value: 2 hours

# Define a function to quit the program
def quit():
    global quit_program
    quit_program = True

# Define a function to handle the interval input from the user
def set_interval():
    global interval
    interval = int(interval_entry.get()) * 60 * 60  # convert hours
to seconds
    interval_window.destroy()


# Create a window for inputting the interval
interval_window = tk.Tk()
interval_window.title("Interval Input")
interval_window.geometry("200x100")
interval_label = tk.Label(interval_window, text="Interval for
last detection (hours):")
interval_label.pack()

interval_entry = tk.Entry(interval_window)

interval_entry.pack()

interval_button = tk.Button(interval_window, text="OK",
command=set_interval)
interval_button.pack()


# Define a function for capturing frames from the camera
def capture_frame():
    global frame_captured
    global lock
    global quit_program

    while not quit_program:
        # Capture frame-by-frame
        ret, frame = camera.read()
        # Acquire the lock to access the captured frame
        lock.acquire()
        # Set the captured frame and release the lock
        frame_captured = frame
        lock.release()


# Define a function for processing frames and detecting cats
def process_frame():
    global frame_captured
    global lock
    global quit_program

    global interval

    last_detection_time = time.time() - interval  # set last detection time
to the specified interval ago

    while not quit_program:
        # Wait until a frame has been captured
        while not isinstance(frame_captured, np.ndarray):
            if quit_program:
                break

        if quit_program:
            break

        # Acquire the lock to access the captured frame
        lock.acquire()
        # Make a copy of the captured frame and release the lock
        frame = frame_captured.copy()
        frame_captured = False
        lock.release()

        # Convert the frame to grayscale
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        # Get the current time
        current_time = time.time()

        # Only detect a cat if it's been at least the specified interval since the
last detection
        if (current_time - last_detection_time) >= interval:

            # Detect cats in the frame
            cats = cat_cascade.detectMultiScale(gray, scaleFactor=1.3,
minNeighbors=5)

            # Draw a rectangle around each cat
            for (x, y, w, h) in cats:
                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
                ser.write(b'8H') # Send a signal to pin 8 on the Arduino board to
set it high
                ser.write(b'13H') # Send a signal to turn on the onboard LED on
the Arduino board
                ser.write(b'0') # Send a signal to pin 8 on the Arduino board to
set it low
                ser.write(b'13L') # Send a signal to turn off the onboard LED on
the Arduino board
                print("Cat found")
                last_detection_time = current_time


        # Display the frame
        cv2.imshow("Cat Detection", frame)

        # Check if the 'q' key has been pressed
        if cv2.waitKey(60) & 0xFF == ord('q'):
            quit_program = True

    # Release the camera and close the window
    camera.release()
    cv2.destroyAllWindows()

# Create and start the threads
capture_thread = threading.Thread(target=capture_frame)
process_thread = threading.Thread(target=process_frame)
capture_thread.start()
process_thread.start()

# Wait for the threads to finish
capture_thread.join()
process_thread.join()

ser.close()
```

Figure 31. Code Snippet Pi-2

14.3 Final Costing Budget

Qty	Component	Model No.	Price(Rs) incl VAT
Already Bought			
1	Raspbery PI	R-PI4-4	9250
1	MicroSd Card	KI0-MSD16	240
1	Water Level Sensor	A-WLS	35
6	Obstacle Sensor PhotoElectric	KIT-FC51	45
4	Ultrasonic Distance Measuring	HC-SR04	60
1	ABS Plastic Enclosure Rasp Pi	A-ACRY PI4	150
1	Submarine Pump	SUBMP10 85	85
40	Dupont Wire Female Female	DPWF-F20	2
1	Battery pack, Board,2 DC motor	A-LCD5	1600
1	Raspberry PI4 Camera	R-PI-CAM	250
1	Arduino Uno	ARD	500
1	Arduino Sheild	-	450
		Supplied by Transcom	
		Total	12667

Not Bought Yet			
8	Acrylic Pane	1	3500
4	Metallic Rode	-	400
		Supplied by General Store	
		Total	3900
		Gross Total	16567

15. Functionality Test Cases

To check if robot is successful and efficient in terms of technicality, these test cases can be ally

COMPONENT/ID	Test Requirements
Rasp01/A02	Input voltage should be properly checked for not to burn out the board
Rasp02/V01	Check & Test calibration of camera
Rasp03	Test if RaspPi Sending correct into to motors and check for power
A02	Test if all sensors are powered on
A03	Check if all connection is ok and run some dummy test
V04	Connect Monitor to Robot to visualise and check if accurate data is being collected
P01	Check if pulley well mounted to motors and if spinning correctly
P03	Check if pulley on kart is moving horizontally
S01	Test Ultrasonic sensors with different distances
S02	Test motors if moving in clockwise and anti clockwise direction
S03	Check if all codes are correct for robot to function efficiently
S04	Check if data is well received between the arduino and the raspberry pi and the sensors by running dummy test

Figure 31. Test Cases

16. Difficulties Encountered

- All the components were not available on the local market.
- Cutting the acrylic was difficult. Same is a tough material to work with without adapted tools.
- Cable management the cable management used was too heavy thus creating a drag force on the food truck.
- Low battery life for the motors
- Since Ubuntu was install on my pi had issue configuring the camera
- Power supply was not enough that's why a battery pack was added.
- Required long wires and had lots of soldering to be done.
- Due to time management issue interface was not designed fully.
- The initial designed was not adapted for the functionalities thus we had to change the design

17. What are the potential pitfalls of your system?

- Power Supply- The Robot should have enough power to supply to the two motors.
- The size of the entrance for the animal should be enough
- High speed movement animal cannot be detected using this model
- It can only feed 1 pet at a time
- If knocked down, internal part may move
- Since we are dealing with water, circuits may get damage.
- No cold storage in case some food needs to be refrigerated.

18. Future implementation?

- Same can be altered to serve animal in zoos such as ass wheel to the robot for it to move to different section in a zoo.
- A cooler can be install in the storage section can be implemented to store cold food.
- The project can be made in bigger version so as 4 animal can eat at the same time at the different side of the robot.
- Internal calibration can be more refine and it auto calibrate every time the pet feeder restarts.
- A mobile app/webapp can be developed for the user to interact with it
- Add rechargeable batteries as a power supply for the robot.

Reason why above features not implemented right now.

i. Budget. This implementation requires better component and more accurate one and these component are more expensive
ii. Time constraints. Since the project is on a whole scale of 4 month the implementation of all these features will be really challenging

16. Project Planning

Project planning is a key aspect for the success of this project. Below you shall find a Gantt Chart for the Project Plan.

PDE3413: Multi Pet Feeder

	January				Febuary				March
	Week1	Week2	Week3	Week4	Week1	Week2	Week3	Week4	Week1
Planning and gathering of materials									
Purchasing components									
Alternatives									
Testing each component									
Construction									
Assembling Pet Feeder									
Building Cart Platform for Movement									
Assembling kart together with sensors									
Put all side panels together and add sensors & LCD Panel									
Coding									
Coding of all components									
Coding ultrasonic and servo motor.									
Coding Camera Vision									
Coding User Interface									
Testing									
System testing (each sensor)									
Integration testing acceptance									
Completion of project									
Project done									
Presentation report									

Figure 32. Gantt Chart

16. Conclusion

To conclude this social robot is a small prototype that has a great utility in our live. Robots are taking the spotlight in every industry. The outcomes not only demonstrate a significant advancement in the social robot technology's pet monitoring system, but also satisfy pet owners' needs.

17. Reference List

1. McDowell (no date) *Benefits of an Automatic Pet Feeder*. Available at: https://www.petsafe.net/learn/benefits-of-an-automatic-pet-feeder#:~:text=Automatic%20feeders%20help%20provide%20proper,over%20a%2015%2D minute%20period. (Accessed: 13 October 2022).
2. Jetir (Year of publication) *Automatic Pet Feeder USING INTERNET OF THINGS*. Available at: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwibg_qT7eL6AhXchf0HHVR7Bn8QFnoECBAQAw&url=https%3A%2F%2Fwww.jetir.org%2Fdownload1.php%3Ffile%3DJETIR1904I61.pdf&usg=AOvVaw2CFB_jiGKtpcLKbonqhQN- (Accessed: 15 October 2022).
3. Surepetcare (2022) *Microchip Pet Feeder*. Available at: https://www.surepetcare.com/engb/pet-feeder/microchip-pet-feeder (Accessed: 15 October 2022).
4. Surepetcare (2022) *Microchip Pet Feeder*. Available at: https://www.surepetcare.com/image.php?file=media/products/1/mpf-1.png (Accessed: 15 October 2022).
5. Surepetcare (2022) *Microchip Pet Feeder*. Available at: https://www.surepetcare.com/image.php?file=media/products/1/how-it-works-trainingmode-feeder.png (Accessed: 15 October 2022).
6. Raspberrypi (2022) *Getting started with the Camera Module*. Available at: https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/1 (Accessed: 17 October 2022).
7. Pikaeducation (2022) *FLOAT SWITCH or FLOAT SENSOR WITH ARDUINO*. Available at: https://pijaeducation.com/arduino/sensor/floatswitch/#:~:text=A%20float%20sensor%20or%20float,an%20electromagnetic%20ON%2FOFF %20switch. (Accessed: 17October 2022).
8. Kumar Sinha, A. (2021) *Animal Classification With RPi Camera For Survey And Monitoring*. Available at: https://www.electronicsforu.com/electronics-projects/animal-classification-rpicamera-survey-monitoring (Accessed: 04 November 2022).
9. *How Water Level Sensor Works and Interface it with Arduino* (2022) Available at: https://lastminuteengineers.com/water-level-sensor-arduino-tutorial/ (Accessed: 05 November 2022).