LOW LEVEL DESIGN AND IMPLEMENTATION DOCUMENT

Smart Waste Segregation

UE17CS490B – Capstone Project Phase – 2

Submitted by:

Guruprasad Hadimani	**PES1201700980**
Nagesh K J	**PES1201701528**
Laxman	**PES1201701632**
Sanathkumar G	**PES1201701771**

Under the guidance of

Prof. Charanraj B R
Assistant professor
PES University

January - May 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

TABLE OF CONTENTS

1. Introduction

1.1. Overview

The low-level design of the system is explained thoroughly in the subsequent sections. It consists of a master class diagram spanning the entire system and also contains design details of every module along with its associated UML diagrams. This is intended to provide the logical design of the actual code that is being used in the system.

1.2. Purpose

- The purpose of a low-level design document is to give the internal logical design of the actual program code.
- A low-level design document describes various types of diagrams in detail to ensure a thorough understanding of the workflow and is designed using the high-level document design.

1.3. Scope

This document consists of the design considerations, a detailed design description for each discrete module of the system using use case diagrams, class diagrams, and deployment diagrams of the system. It also includes the proposed methodology along with some relevant information in the appendix part.

2. Design Constraints, Assumptions, and Dependencies

Constraints:

Garbage mixed and given for separation may not be classified into the correct category. For example, a plastic material dipped in water may not be classified as plastic due to the limitations of the hardware used.

Assumptions:

We need to place the garbage one by one for separation rather than dumping it all at once. This is because of the limitations and accuracies of the hardware used.

Dependencies:

- The dependency type of this system is Finish-to -Start i.e., the first task should be completed before the second task can start.
- In our system, the separation of wet waste from dry waste needs to be completed first before further classification of dry waste into different categories.

- Likewise, the metallic waste needs to be segregated first from non-metallic waste before further classification of non-metals into plastic or glass.

3. Design Description

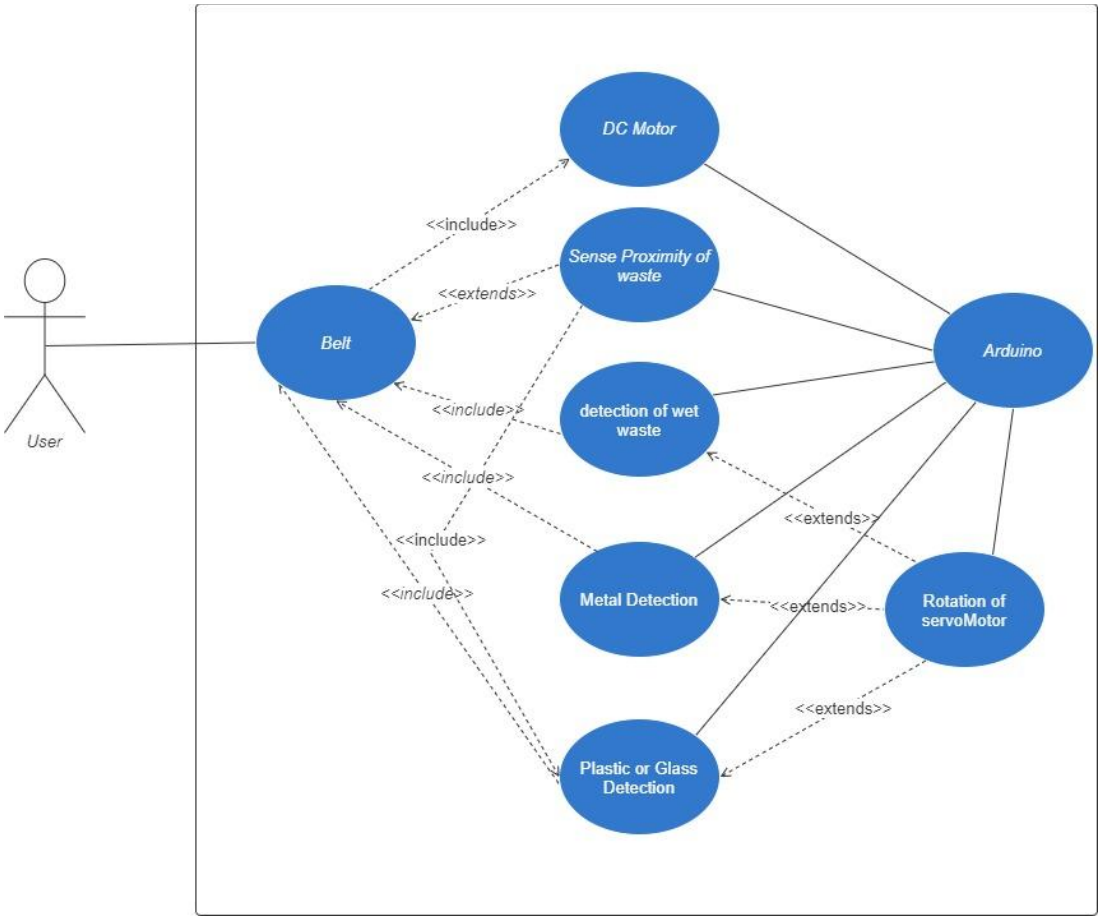
The system mainly consists of the following modules. They are:

- Conveyor belt module
- Moisture detection module
- Metal detection module
- Glass and Plastic separation module

3.1. Usecase Diagram

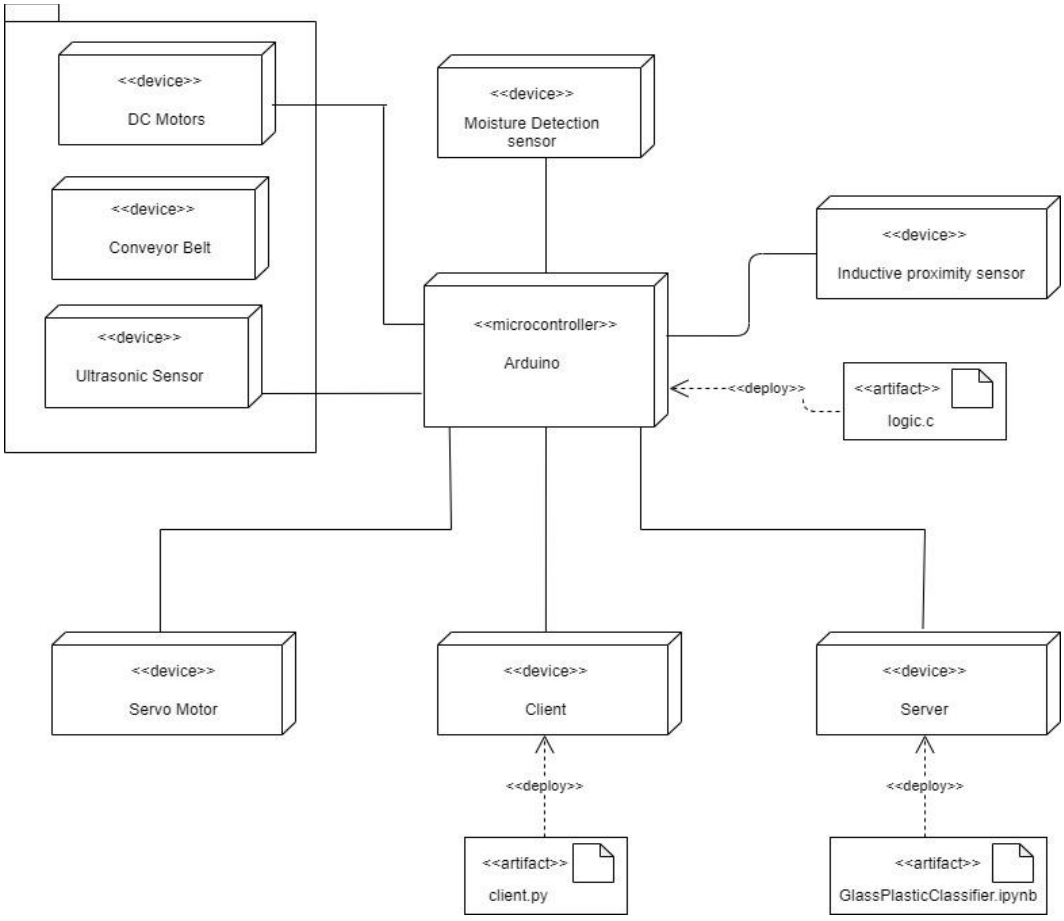
The below diagram represents the use case diagram for the entire system.

This is further explained at the module level in subsequent sections.



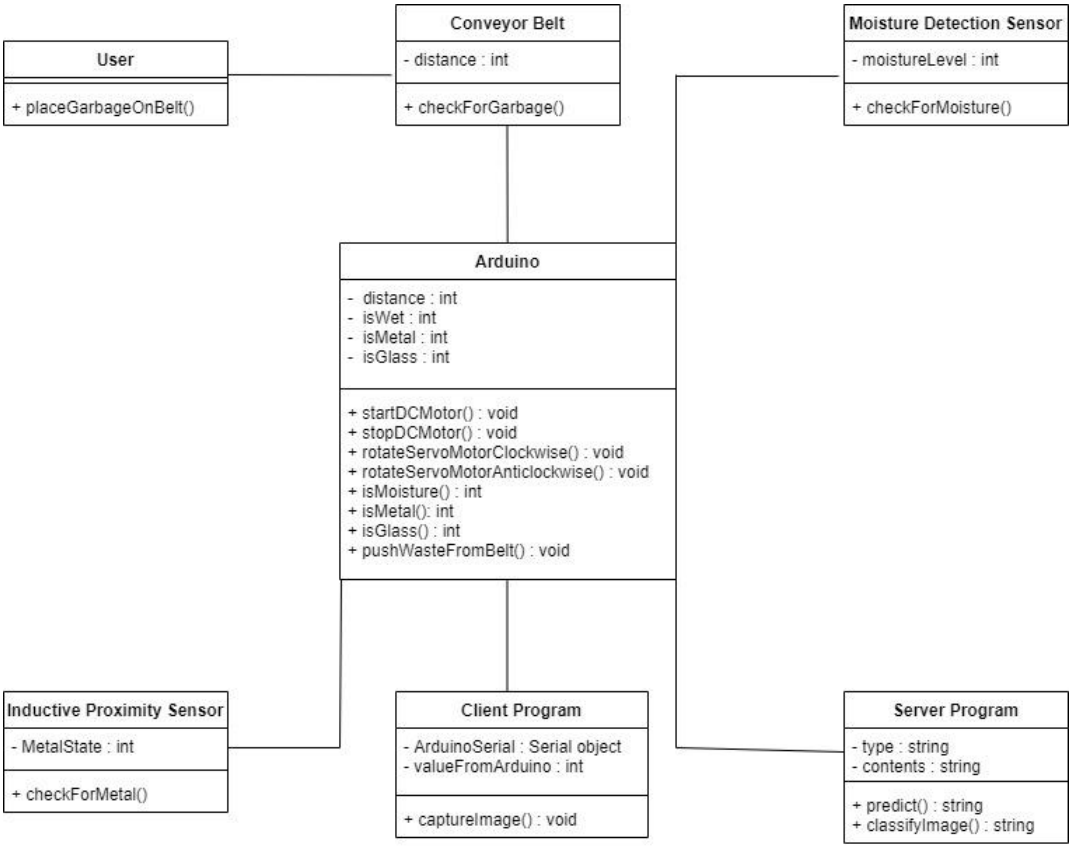
3.2. Deployment Diagram

The below diagram shows the deployment diagram for the entire system. This is further explained at the module level in subsequent sections.



3.3. Master Class Diagram

The class diagram for the system is as follows. This shows how each module interacts with other modules to enable the working of the system as intended. This diagram also shows the dependencies between various components of the system.



Class Name: User

Class Description: This class represents the user, who will use the system to segregate waste automatically.

Class Method: placeGarbageOnBelt() This is a representative method, which represents the user's action of placing the waste on the belt.

Class Name: Conveyor Belt

Class Description: This class represents the belt along with the ultrasonic sensor and DC motors.

Data Member: distance: It is an integer variable that indicates the distance of the waste placed on the belt from the sensor.

Class Method: checkForGarbage(): This method checks if the waste is placed on the belt or not. This is achieved by checking if the waste is within a certain distance from the sensor.

Class Name: Moisture Detection Sensor

Class Description: This class represents the moisture detection sensor along with the servo motor.

Data Member: moistureLevel: It is a variable that indicates the moisture level of the material.

Class Method: checkForMoisture(): This function internally uses the above variable to check if the waste is wet or not using a threshold value for the variable.

Class Name: Inductive Proximity Sensor

Class Description: This class represents the inductive proximity sensor along with the servo motor.

Data Member: MetalState: It is an integer variable that indicates whether the waste is metallic or not.

Class Method: checkForMetal(): This function uses the output of the inductive proximity sensor and returns waste is metallic or not.

Class Name: Client Program

Class Description: This class represents the python program that is responsible for capturing the image and requesting the category of the image from the server.

Data Member: Arduino Serial: It is an instance of the Serial object that is used to communicate with the Arduino and write values to Arduino.

valueFromArduino : It is an integer variable used to instantiate the process of capturing images.

Class Method: captureImage(): This function is responsible for capturing the image using the IP Webcam android app and saves it in a jpeg file. This file is then sent to the server

to get the category of the captured image and writes the response to the Arduino for further actions.

Class Name: Server Program

Class Description: This class represents a python program that is responsible for classifying the image as plastic or glass from the image file sent by the client using a pre-trained classifier model.

Data Member: type: It is a string that contains the value that represents whether the image is plastic or glass.

contents: It is a string that contains the image in binary format.

Class Methods: predict(): This function takes the image as input and predicts its category using the classifier model and returns the category.

classifyImage(): This function fetches the image using REST API, saves it in a jpeg file, and then invokes the predict function.

Class Name: Arduino

Class Description: This class represents the Arduino microcontroller which is the heart of the entire system. It coordinates with all the other classes to ensure the system works as intended.

Data Member: distance: It is an integer variable that indicates the distance of the waste placed on the belt from the sensor.

isWet: It is an integer variable that contains value 0 or 1, where 0 indicates dry waste and 1 indicates wet waste.

isMetal: It is an integer variable that contains value 0 or 1, where 0 indicates metallic waste and 1 indicates non-metallic waste.

isGlass: It is an integer variable that contains value 0 or 1, where 0 indicates plastic waste and 1 indicates glass waste.

Class Methods:

startDCMotor(): This function is responsible for starting the DC motors.

stopDCMotor(): This function is responsible for stopping the DC motors.

rotateServoMotorClockwise(): This function takes a servo object, start degree, and end degree. It then rotates the servo motor between the start degree and end degree in the clockwise direction.

rotateServoMotorAnticlockwise(): This function takes a servo object, start degree, and end degree. It then rotates the servo motor between the start degree and end degree in the anticlockwise direction.

isMoisture(): This function is responsible for checking if the waste is wet or dry.

isMetal(): This function is responsible for checking if the waste is metallic or non-metallic.

isGlass(): This function is responsible for checking if the waste is glass or plastic.

pushWasteFromBelt(): This function accepts a servo object as the parameter and internally invokes startDCMotor(), stopDCMotor(), rotateServoMotorClockwise() and rotateServoMotorAnticlockwise()

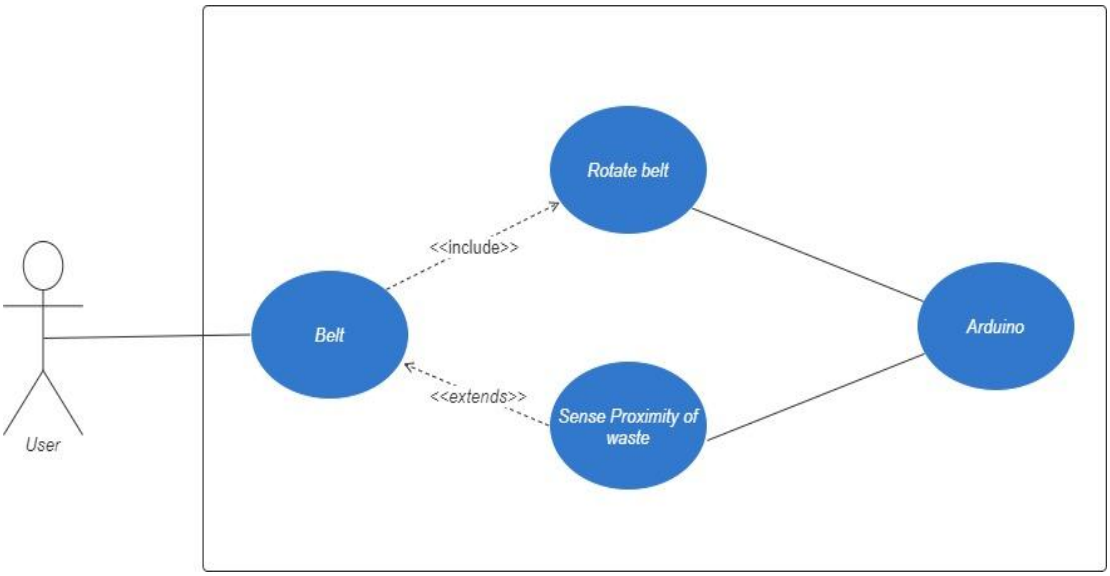
to push the waste off the belt.

3.4. Conveyor Belt Module

3.4.1. Description

The conveyor belt module is responsible for moving the waste placed on the belt to the different sensing modules. This module is driven by two DC motors which are powered by a microcontroller and a 12V DC power supply. It also contains an ultrasonic sensor. This module initiates the movement of the conveyor belt only if the ultrasonic sensor detects the waste on the belt or else it continues to stay in a stationary state. This is done to achieve a less power-consuming design.

3.4.2. Use Case Diagram

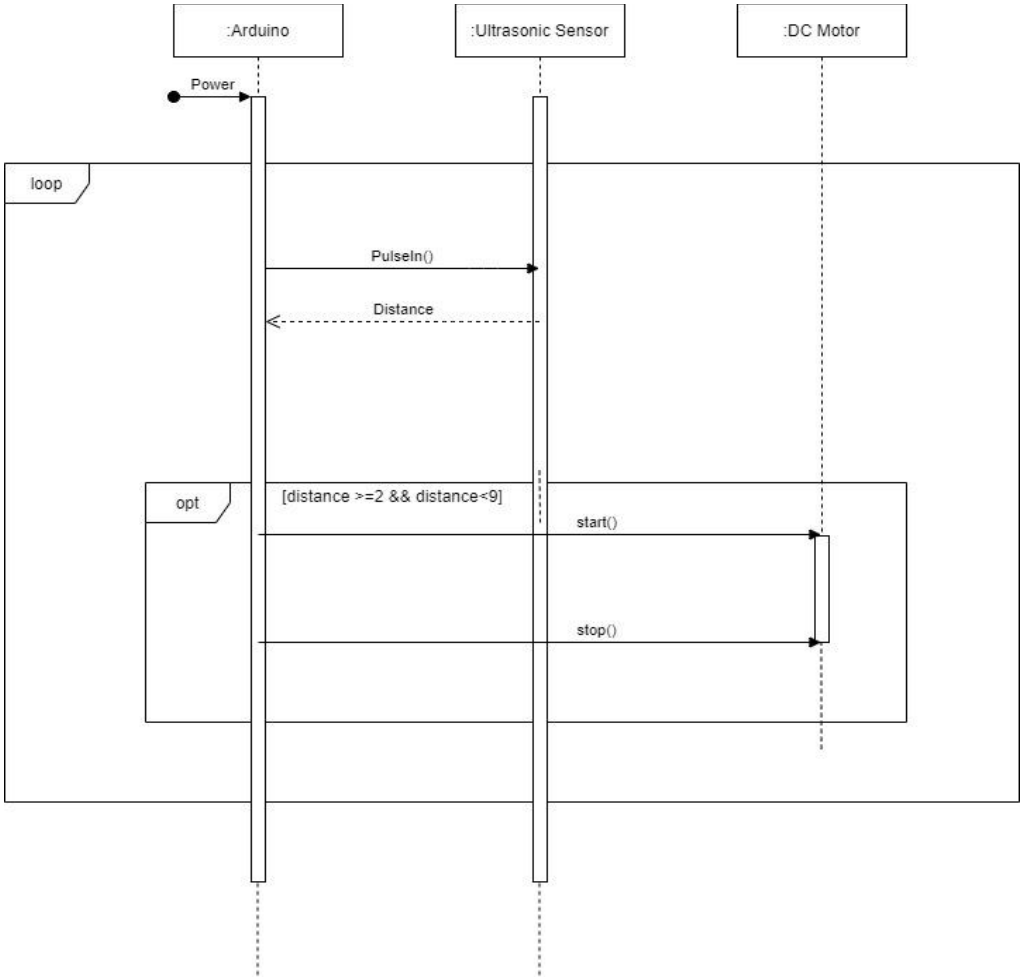


Use Case Item	Description
Belt	It includes the conveyor belt made of rexine material used to move the waste from the source to respective bins.
Rotate Belt	It includes two 12V DC motors used to drive the conveyor belt.

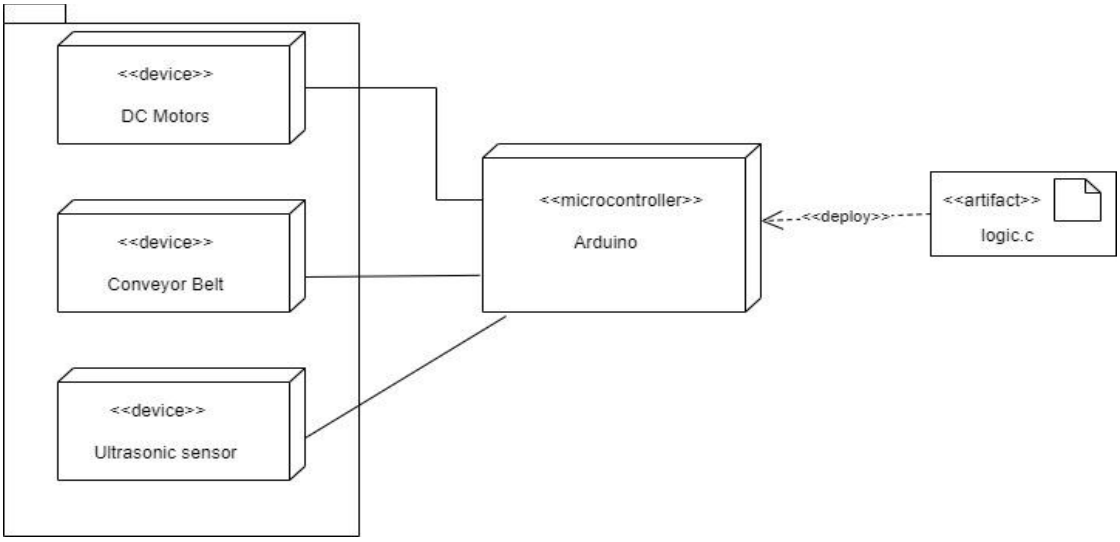
Sense Proximity of waste	This includes an ultrasonic sensor which is used to detect the proximity of the waste on the belt.
Arduino	It is the microcontroller that contains the logic for driving DC motors and checking the proximity of waste that is placed on the belt.

3.4.3 Sequence Diagram

The below is the sequence diagram for the conveyor belt module.



3.4.4 Deployment Diagrams

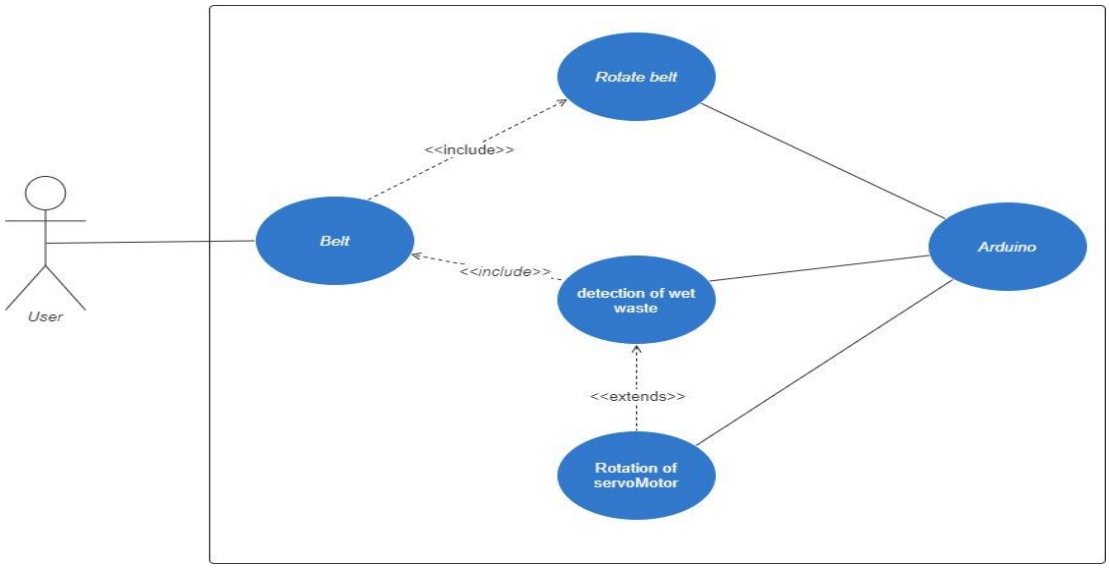


3.5. Moisture Detection Module

3.5.1. Description

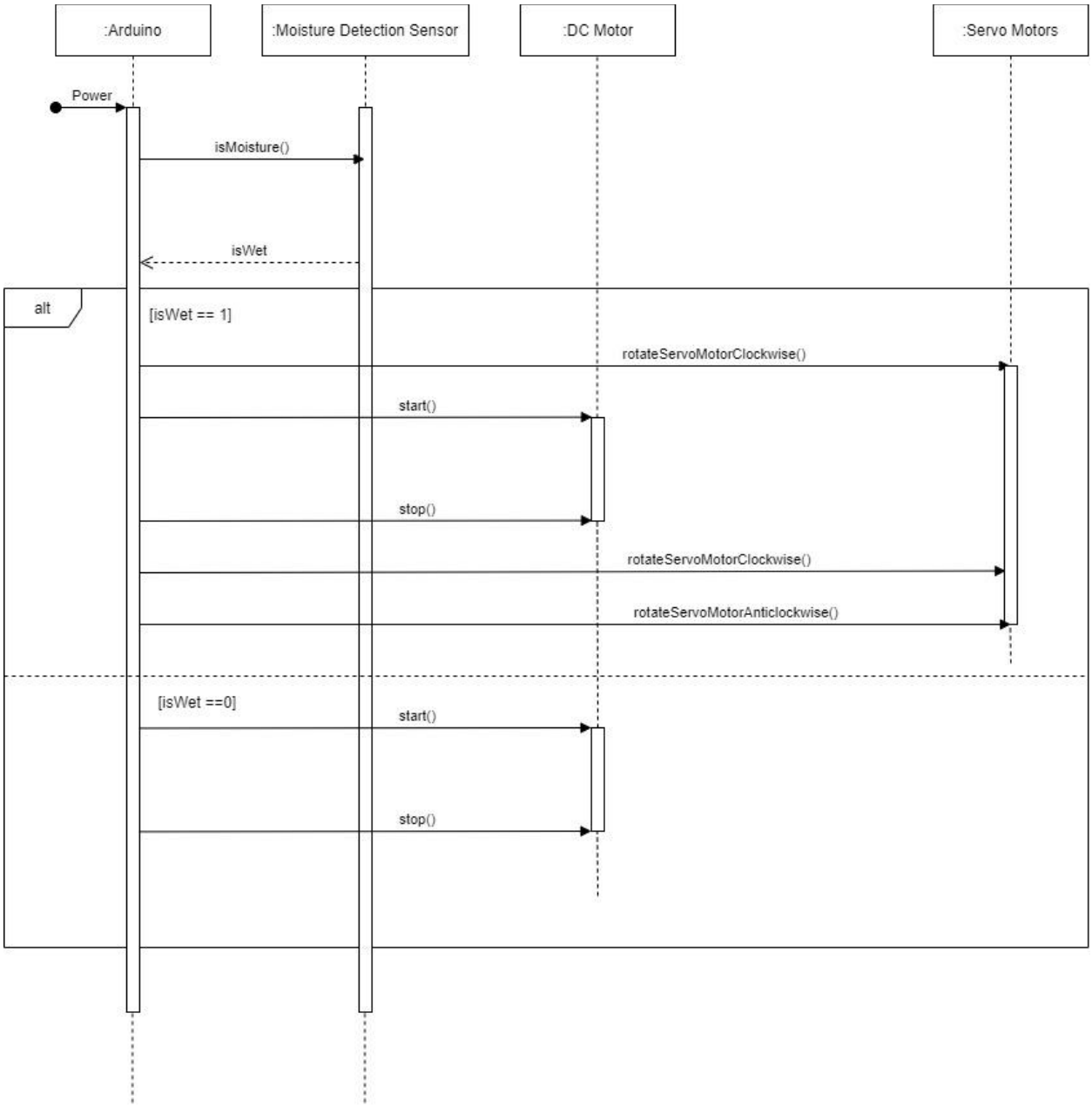
The moisture detection module consists of a moisture detection sensor along with the servo motor with flaps which is used to push the waste off the belt. When the moisture detection sensor detects the waste to be wet, the waste is pushed off the belt using the flap-like mechanism else, the waste is moved further on the belt to the next module for further segregation.

3.5.2. Use Case Diagram

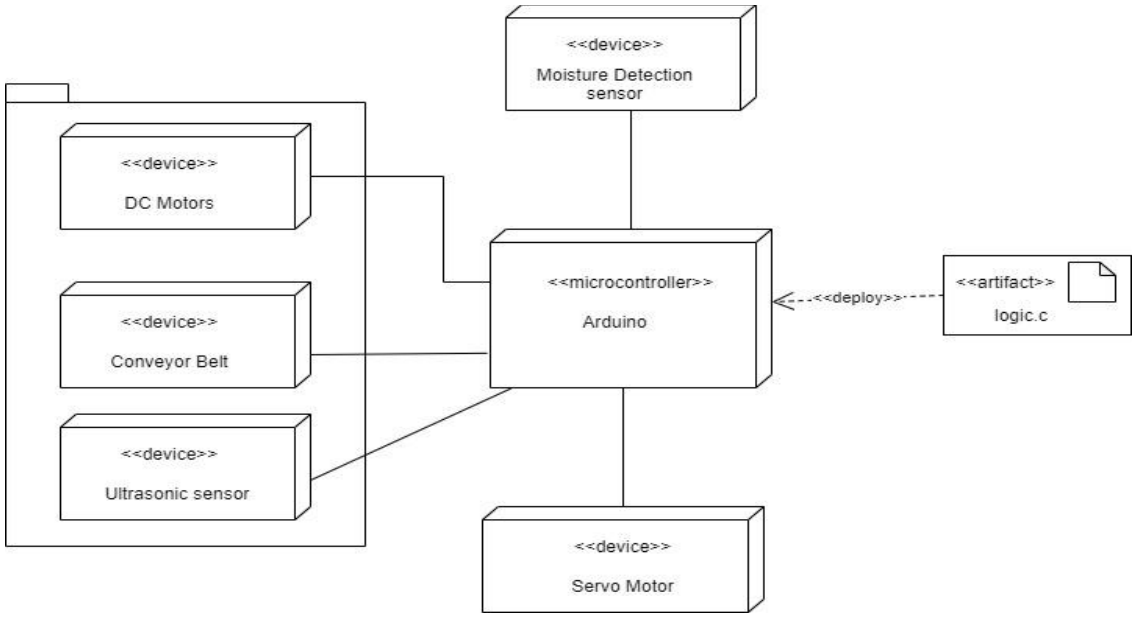


Use Case Item	Description
Belt	It includes the conveyor belt made of rexine material used to move the waste from the source to respective bins.
Rotate belt	It includes two 12V DC motors used to drive the conveyor belt.
Detection of Wet Waste	It includes a moisture detection sensor which when comes in contact with the material detects it as wet or dry.
Rotation of Servo Motor	It consists of a servo motor and flap attached to it which is used to push waste off the belt when detected as wet waste.
Arduino	It is the microcontroller that contains the logic for driving DC motors and coordinates with the moisture detection sensor and respective servo motor.

3.5.3. Sequence Diagram



3.5.4. Deployment Diagrams

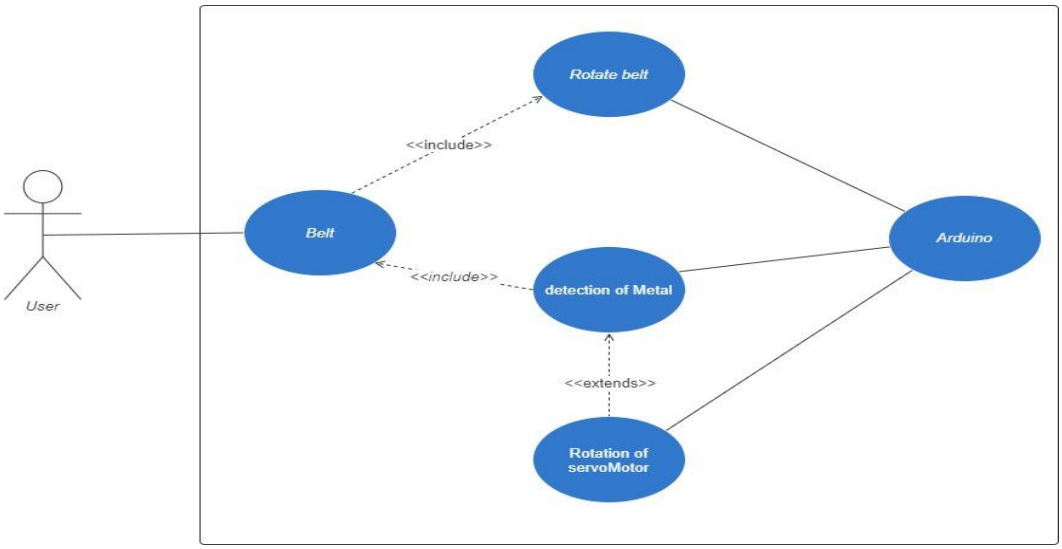


3.6. Metal Detection Module

3.6.1. Description

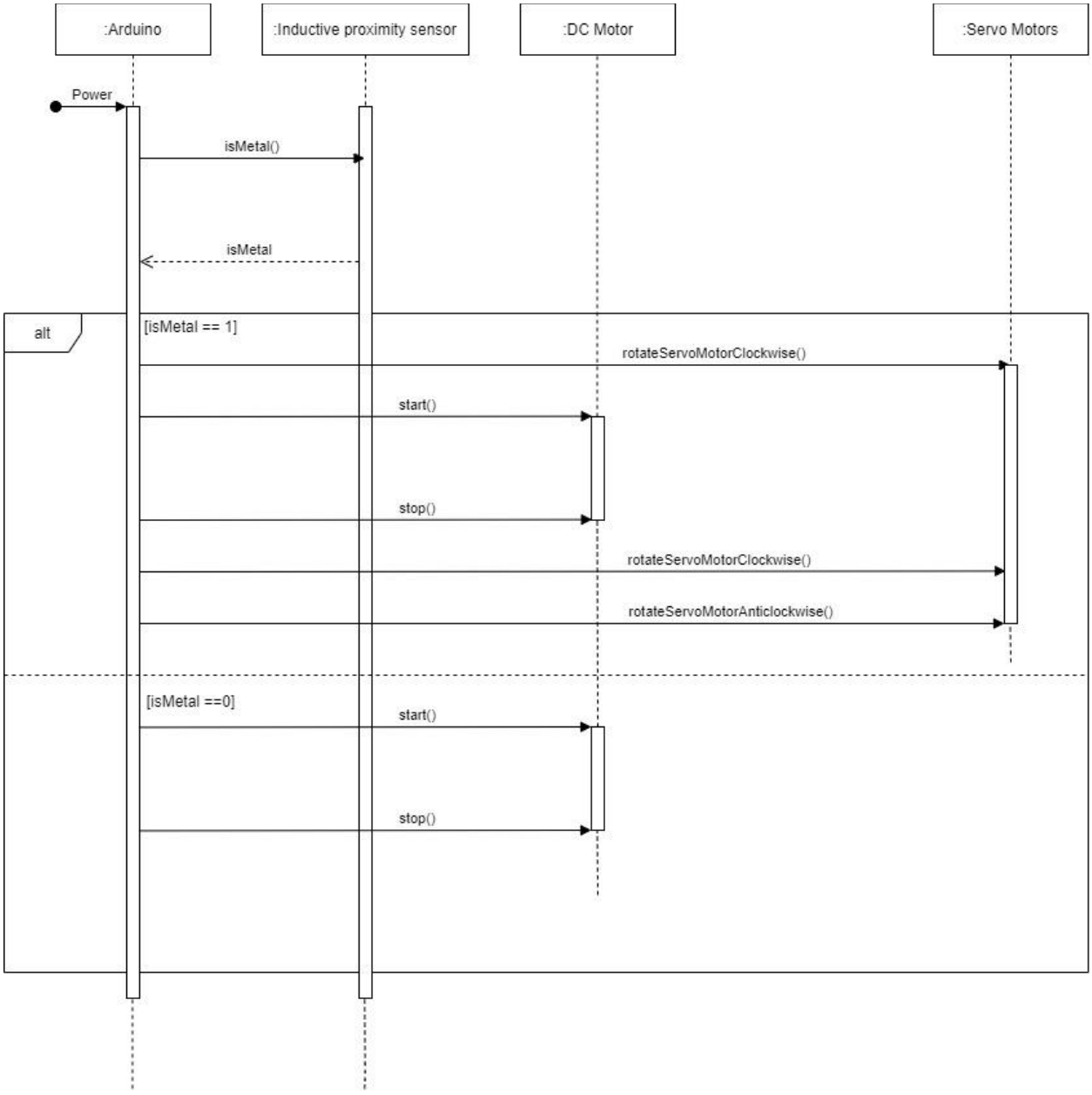
This module consists of an inductive proximity sensor along with a servo motor and a flap-like mechanism attached to the servo motor to push waste off the belt. The servo motor is activated once the inductive proximity sensor detects the waste to be metallic. If it is non-metallic, then the waste is moved further on the belt to classify it as glass or plastic.

3.6.2. Use Case Diagram

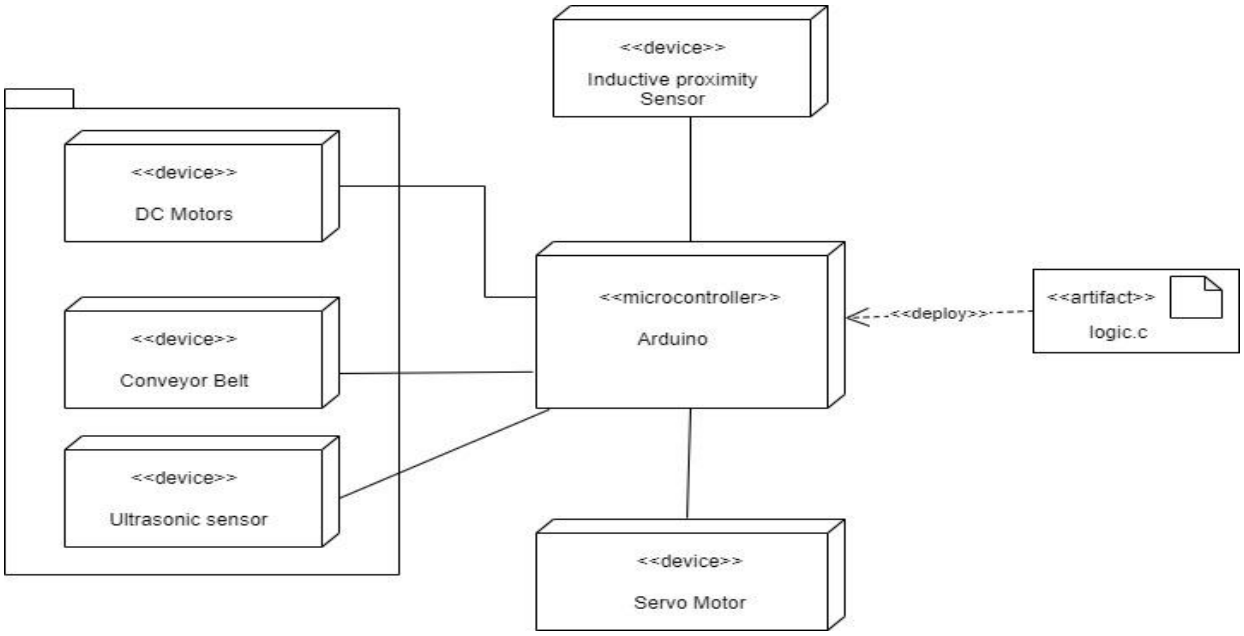


Use Case Item	Description
Belt	It includes the conveyor belt made of rexine material used to move the waste from the source to respective bins.
Rotate belt	It includes two 12V DC motors used to drive the conveyor belt.
Detection of Metal	It includes an inductive proximity sensor that detects the material as metallic or non-metallic when the material comes in the vicinity of the sensor.
Rotation of Servo Motor	It consists of a servo motor and flap attached to it which is used to push waste off the belt when detected as metallic waste.
Arduino	It is the microcontroller that contains the logic for driving DC motors and coordinates with the inductive proximity sensor and respective servo motor.

3.6.3. Sequence Diagram



3.6.4. Packaging and Deployment Diagrams



3.7. Plastic and Glass Detection Module

3.7.1. Description

This consists of an ultrasonic sensor, a python client program, a python server program, a servo motor along with a flap-like mechanism for pushing waste off the belt, and a mobile camera for capturing images.

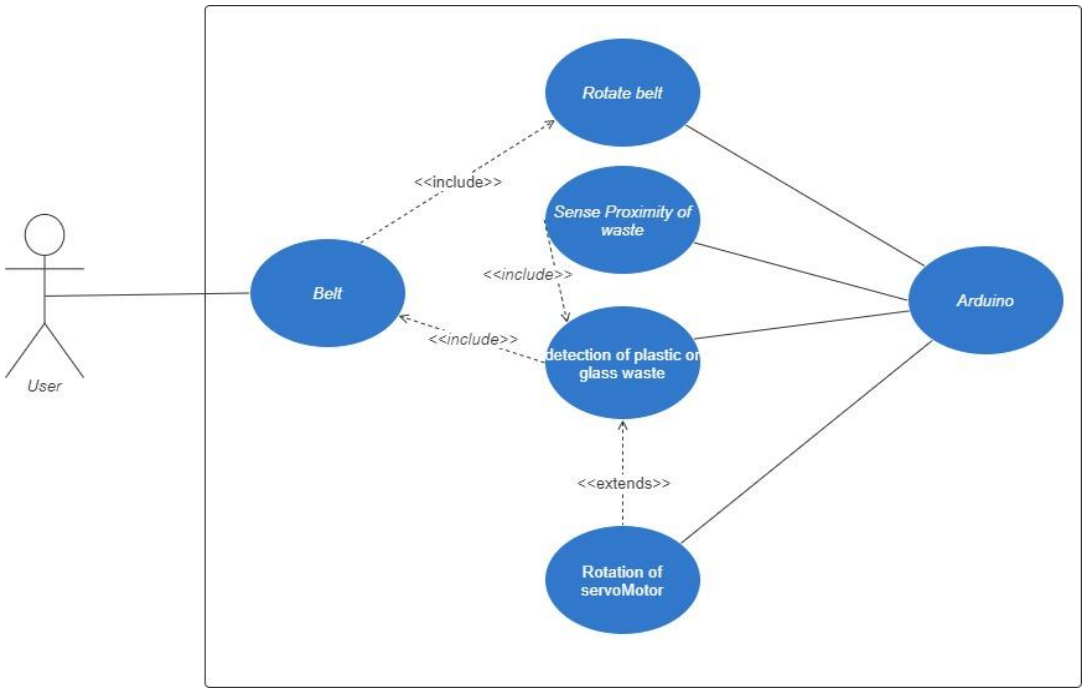
The ultrasonic sensor is connected to Arduino to sense the proximity of the waste. The Arduino is in turn connected to the python client program using a serial port connection.

Whenever the ultrasonic sensor detects the waste in its proximity, it activates the python client, which in turn captures the image from the mobile camera and sends the image to a server program running on google collaborator through a REST request.

The server has a pre-trained classifier that classifies the image as either glass or plastic. The category is sent back to the python client as a response to the REST request. Based on the response from the server, the client program writes values to

Arduino to push waste off the belt if the waste is plastic else, it activates the DC motors so the glass waste can fall at the end of the belt.

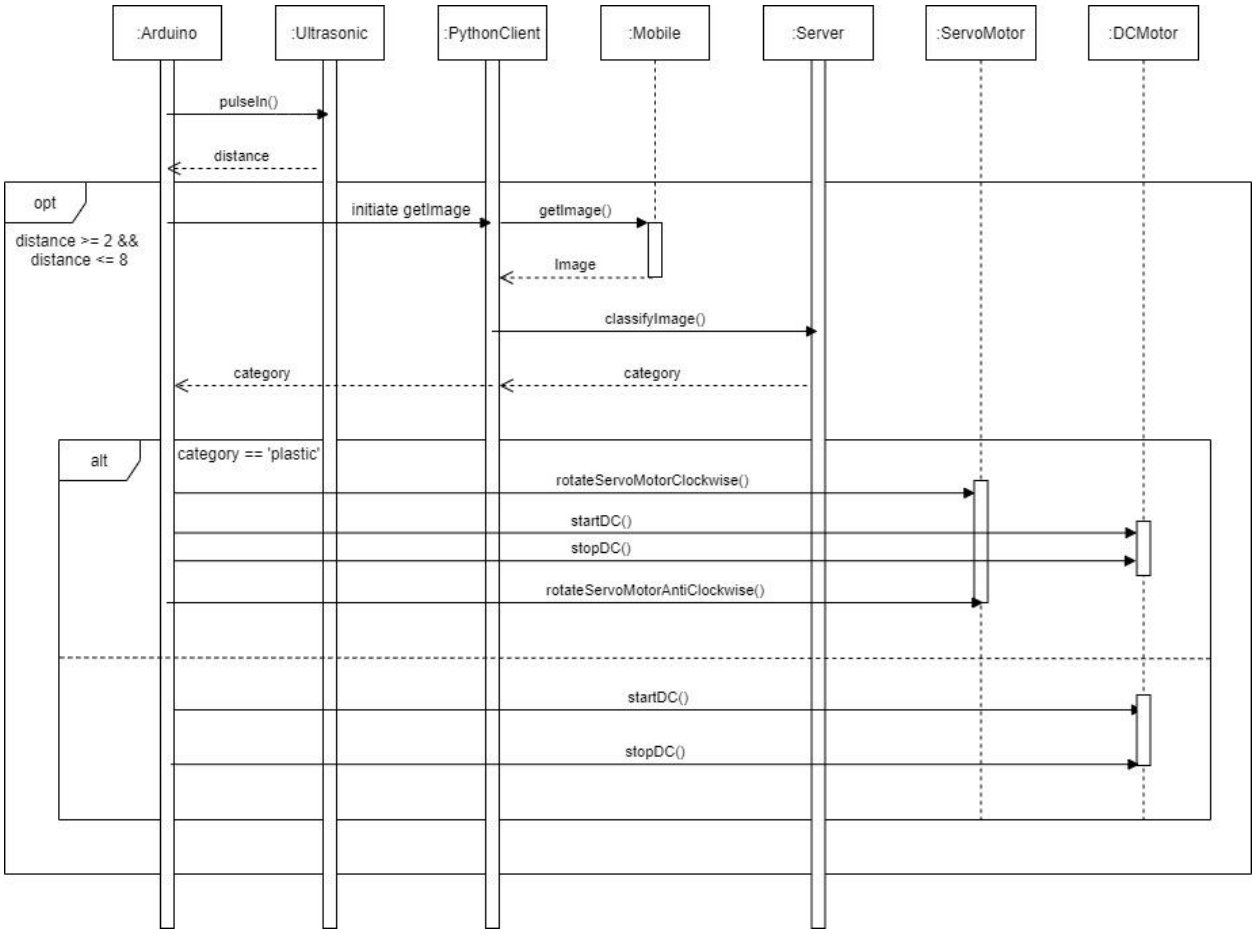
3.7.2. Use Case Diagram



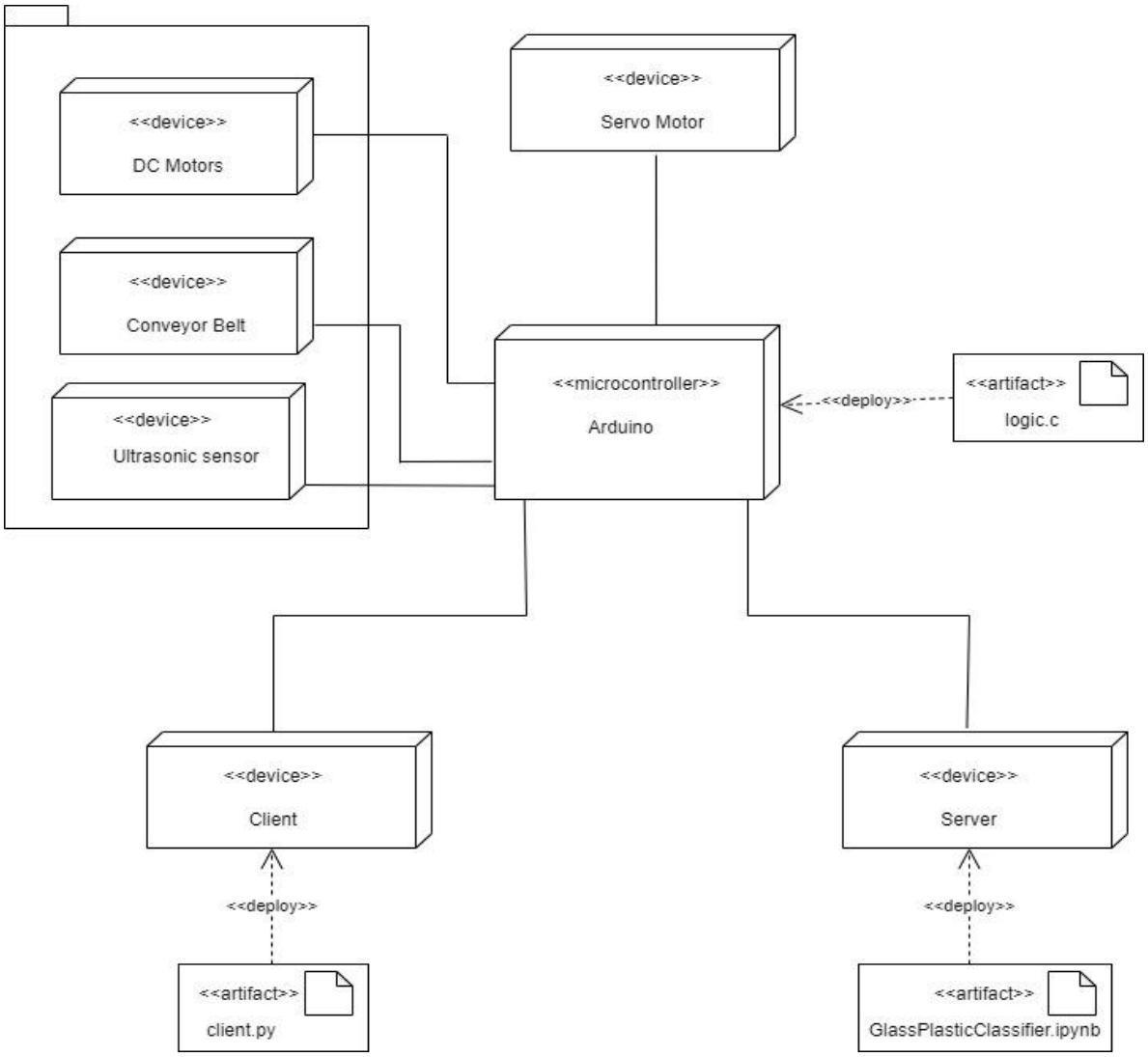
Use Case Item	Description
Belt	It includes the conveyor belt made of rexine material used to move the waste from the source to respective bins.
Rotate belt	It includes two 12V DC motors used to drive the conveyor belt.
Sense Proximity of Waste	This includes an ultrasonic sensor which is used to detect the proximity of the waste on the belt.
Detection of Plastic or Glass Waste	It consists of a python client, a python server, and a mobile camera for capturing images. The python client is responsible for capturing images from the camera and sending them to the server for classification of

	material as plastic or glass. The server program has a pre-trained classification model that takes the image as input and sends the category of the waste in the image back to the client. The client coordinates with the Arduino for segregation.
Rotation of Servo Motor	It consists of a servo motor and flap attached to it which is used to push waste off the belt when detected as plastic waste.
Arduino	It is the microcontroller that contains the logic for driving DC motors and coordinates with the python client and respective servo motor.

3.7.3. Sequence Diagram

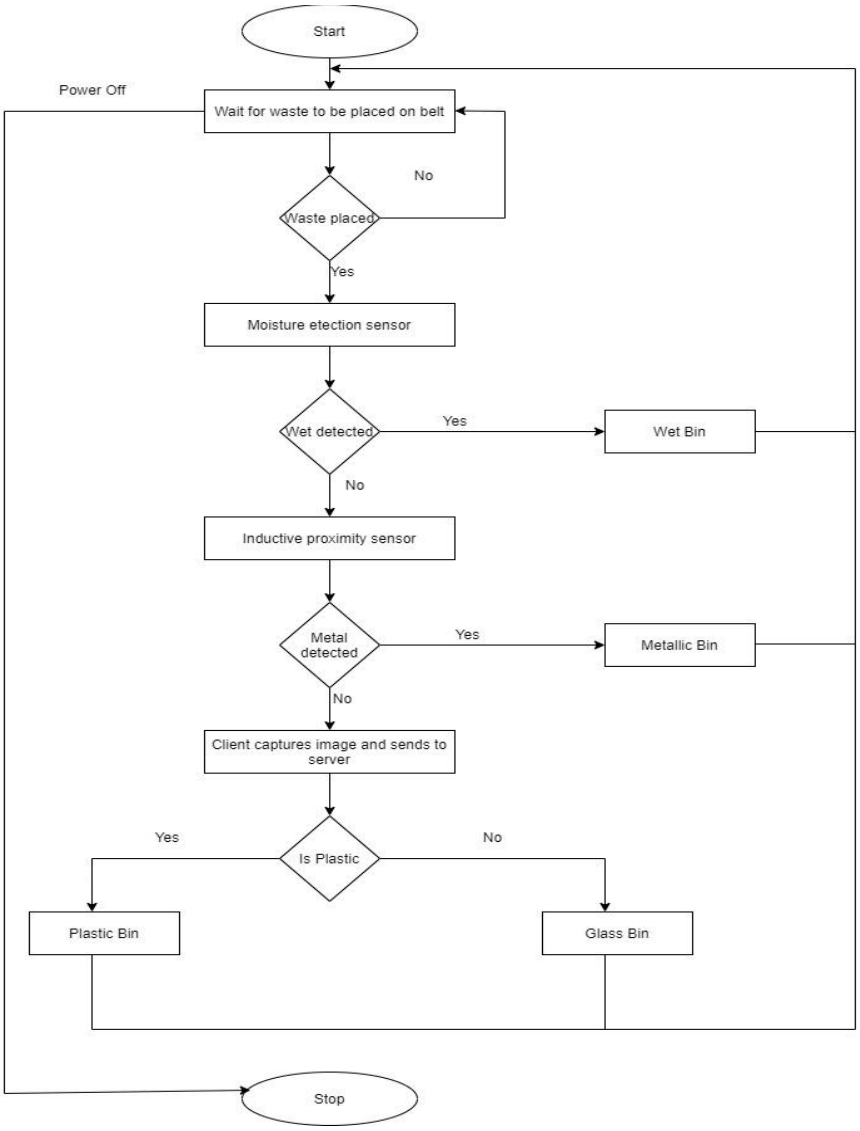


3.7.4. Deployment Diagrams



4. Proposed Methodology / Approach

The below flowchart shows the proposed methodology of the entire system.



Appendix A: Definitions, Acronyms and Abbreviations

Conveyor belt module: This module consists of a conveyor belt, DC motors for driving the belt, proximity sensor for switching on and off the belt.

Moisture detection module: This module consists of a moisture detection sensor along with a mechanism that uses a servo motor to push off the garbage to the respective bin if the garbage is detected as wet.

Metal detection module: This module consists of an inductive proximity sensor along with a mechanism that uses a servo motor to push off the garbage to the respective bin if the garbage is detected as metal.

Plastic and Glass Detection Module: This module consists of an ultrasonic sensor, a python client, a mobile camera, and a python server responsible for classifying the waste as glass or plastic. This also uses a servo motor to push the waste off the belt if detected as plastic.

Appendix B: References

High-Level Design Document that was done as part of the capstone phase-1.

For component diagram:

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/

For deployment diagram:

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/

For state diagram:

https://www.guru99.com/state-machine-transition-diagram.html

https://www.smartdraw.com/state-diagram/

For designing all UML diagrams for the system.

https://app.diagrams.net/

Appendix C: Record of Change History

#	Date	Document Version No.	Change Description	Reason for Change
1.		1	Instead of using the capacitive proximity sensor for segregation of plastic and glass waste image classifier model is incorporated.	The output of the capacitive proximity sensor was varying drastically whenever the materials were brought near to it. Because of this behavior, it was difficult to find out the ranges which could differentiate between glass and plastic.

Appendix D: Traceability Matrix

Project Requirement Specification Reference Section No. and Name.	DESIGN / HLD Reference Section No. and Name.	LLD Reference Section No. Name
3.Functional requirements	6.State diagram	4. Proposed Methodology
3.Functional requirements	5.Design Description	3. Design Description
4.2.Hardware requirements	10.Deployment diagram	3.2. Deployment Diagram