TrashBot - An Artificial Intelligence-based Framework for Automated Waste Segregation

A thesis submitted in partial fulfillment of the requirements for the Degree of Bachelors of Software Engineering

Submitted By

Mahrukh Ali Khan (2019-BSE-037)

Arshiya Saleem (2019-BSE-004)

Hala Ali khan (2019-BSE-007)

Supervisor

Dr. Aamir Arsalan

Co-Supervisor

Dr. Saria Safdar

DEPARTMENT OF SOFTWARE ENGINEERING

FATIMA JINNAH WOMEN UNIVERSITY THE MALL ROAD RAWALPINDI

JULY 2023

Acknowledgements

We would like to express our heartfelt gratitude to Almighty Allah for granting us the strength and perseverance to embark on and complete this research journey. This thesis represents a culmination of our efforts, and we are deeply thankful for the blessings and guidance we have received throughout this process.

We extend our sincere appreciation to our thesis supervisor and co-supervisor, Dr. Aamir Arsalan and Dr. Saria Safdar, for their unwavering support, invaluable guidance, and profound expertise. Their mentorship and insightful feedback have been instrumental in shaping this research. We are grateful for their dedication and commitment to our academic and personal development.

We would also like to express our gratitude to the faculty and staff of the Software department. Their provision of resources, research facilities, and a conducive learning environment have significantly contributed to the successful completion of this thesis.

We acknowledge and appreciate the contributions of all those who have supported us in various ways, directly or indirectly, in making this thesis a reality. We are truly grateful for their presence in our lives. Thank you all for being a part of this remarkable chapter in our academic and personal growth.

Dedication

Our parents' unwavering support, encouragement, and love have been the foundation that has sustained us throughout our lives. Therefore, this thesis is dedicated to them.

Contents

List of Figures

List of Tables

Abstract

Handling heaps of garbage is the most essential issue of today's era. Improper disposal of garbage has a negative impact on the environment and contributes to adverse climatic changes. Harmful substances and pollutants are released from the accumulated heaps of garbage. Food scraps and organic wastes release methane which contributes to climatic change. Toxic chemicals from improper disposal of electronic waste can contaminate soil and water sources which will harm human life. Improper waste management results in filling landfills rapidly, leading to deforestation and the destruction of natural habitats. Contaminated soil and water from garbage will affect those who rely on it for their survival i.e. plants, animals, and humans. Accumulated heaps of garbage release harmful gases into the air which causes air pollution and respiratory problems. The poor air quality will have an effect on human life. Accumulation of garbage in landfills increases global greenhouse gas emissions. It is essential to adopt proper waste management practices to mitigate these negative impacts and have a healthier environment. This includes reducing waste generation through recycling, properly disposing of segregated waste, efficient waste collection, and raising public awareness about proper waste management. We will be able to protect the environment, conserve resources and mitigate adverse climatic changes by managing and disposing of heaps of garbage properly and efficiently. Disposal and recycling of trash is one of the main challenges of the present era because the garbage is accumulated in landfills. The improper disposal of garbage can result in heaps of rubbish which will harm the environment. The TrashBot bin is an efficient solution to the waste management problem. TrashBot bin is an

automated trash sorter that can segregate the trash without any error and with an accuracy of 87% that has been achieved on a trash dataset of seven categories i.e. wood, plastic, glass, metal, cardboard, battery, and others. Previously, the process of sorting garbage was carried out manually, which was tedious and perilous work, since it contained harmful and hazardous wastes which endanger human life. TrashBot bin will fix the problem and prevent the garbage from turning into heaps. Also, sorted trash is much easier to be disposed of. TrashBot bin will contain several bins inside it, in which it will segregate waste into its respective bin. This segregation will prevent the accumulation of garbage in landfills. The segregated waste will be easily disposed of without contaminating soil, air, and water. The TrashBot application will have several features which facilitate the customer in proper waste management. The schedule bin feature will allow the user to book a slot of his choice for the collection of waste. This will prevent the accumulation of waste in landfills. The user will be notified through the application when the TrashBot bin will be full. The news feature will keep the user aware of the latest environmental and adverse climatic changes due to heaps of garbage. This will raise awareness for proper waste management among users. The report generation feature will allow the user to analyze how much waste is produced weekly and monthly. This will encourage users to reduce waste generation.

Chapter 1

Introduction

Heaps of garbage have posed numerous challenges to human life. It is now becoming a significant problem for today's era as the population is increasing day by day. Rapid population growth and urbanization make it difficult to handle large quantities of garbage efficiently. When waste is not managed properly, it causes a lot of garbage to pile up in landfills. The accumulation of garbage leads to air, water, and soil pollution. It contaminates the sources of water which harm the lives of animals, plants, and humans. The accumulation of garbage has an adverse effect on climatic changes due to the release of methane into the air. Deforestation and the destruction of natural habitats are also key problems because of the accumulation of garbage in landfills. It is essential to solve this problem in order to protect the environment and climate. Therefore, the smart AI TrashBot bin has been designed to have segregated waste at the household and commercial levels. Segregated waste can easily be disposed of. The TrashBot application will help the user in managing waste efficiently, reduce waste generation and prevent the accumulation of waste in landfills. The user will be aware of the latest adverse climatic changes due to garbage accumulation to highlight the importance of managing waste efficiently.

1.1 Title

TrashBot - An Artificial Intelligence-based Framework for Automated Waste Segregation

1.2 Problem Statement

Disposal and recycling of trash are one of the main challenges of the present era. No proper disposal can result in heaps of rubbish which can be very harmful to our environment. The proposed solution to this problem is to construct an automated TrashBot bin that can segregate the trash without any error and with much accuracy. Previously, the process of sorting garbage was carried out manually, which was tedious and perilous work, since it contained harmful and hazardous wastes. The trash sorter bin will fix the problem and prevent the garbage from turning into heaps. Also, sorted trash is much easier to be disposed of. We will classify the garbage into different categories and will train the models on those classes. Once trained we can then use it to classify the trash and can prevent it from turning into harmful waste for our environment.

1.3 Aims and Objectives

The TrashBot application is designed to automate the process of waste segregation. TrashBot will generate daily and monthly reports of waste being segregated. These reports will help in recycling and disposing of waste. The aim of TrashBot is to achieve waste classification by automating the waste segregation process. Our goal is to design a system that is time efficient, accurate, and less risky to human life and the environment. This will help in disposing of the segregated waste easily. We will implement deep learning techniques to solve environmental problems like waste accumulation and segregation which can eventually lead to pollution. We will design a model for waste segregation into several categories, which will help in waste fabrication and recyclability. Our main objectives will be:

- To collect a dataset of waste
- To train deep learning model
- To achieve the highest accuracy
- To deploy our model
- To integrate the model with the application

1.4 Thesis Layout

The thesis layout comprises a chapter-wise summary. Basically, the thesis consists of five chapters. Chapter 1: In this chapter, the introduction and overview of the TrashBot application and bin are discussed. Chapter 2: In this chapter, a literature review of the waste management system is discussed. Chapter 3: In this chapter, the software requirement specification of the TrashBot application is discussed. Chapter 4: In this chapter, the software design of the TrashBot application is discussed. Chapter 5: In this chapter, software testing of the TrashBot application is discussed.

Chapter 2

Literature Review

The literature review focuses on exploring the current state of research and advancements in waste management and classification using deep learning techniques. It begins with an overview of the waste management system and highlights the rising challenges associated with waste production and recycling. The review examines various waste classification methods and the role of computer-based image recognition in waste management. It also delves into the concept of smart bins and the application of convolutional neural networks (CNN) for waste classification. Additionally, the review discusses innovative technologies for garbage detection and monitoring, as well as the potential of deep learning in enhancing recycling efficiency. The challenges faced in trash classification and the advancements in deep learning models are also examined. Finally, the review concludes by highlighting the progress made in waste classification and the potential for future advancements in this field.

2.1 Overview of Waste Management System

It provides an overview of the global waste management problem, emphasizing the increasing amount of waste generated each year and the associated environmental issues. It mentions the health problems caused by manual waste segregation and highlights the need for automating the process. [1]

2.2 Rise in Waste Production and Recycling Issues

It highlights the increase in waste production due to urbanization and development. It mentions the improper recycling practices and the associated risks to human and animal life, as well as the contribution to global warming [2].

2.3 Challenges in Waste Management and the Role of Deep Learning

Domestic waste is one of the most common pollutants, which contributes to environmental pollution. By 2025, solid waste will reach up to 2.2 billion, and the corresponding waste treatment cost will amount to 375 billion US dollars. Hence, an effective waste management system is necessary to tackle this issue. Intelligent systems based on deep learning have gained popularity due to their superior performance in various computer vision tasks [3]. In this study, an intelligent system named GarbageNet is proposed, which uses deep convolutional neural networks for trash classification. The system is trained on a dataset consisting of 60,000 images divided into six classes, achieving an impressive accuracy of 94.28 percent. The results demonstrate the potential of deep learning techniques in addressing the challenges of garbage classification and recycling [4].

2.4 Addressing Waste Accumulation and Processing Challenges

In recent times, the accumulation of waste has become a significant issue in our society, particularly in Thailand, where the current waste processing method falls short in managing the staggering 27 million tons of trash generated annually. To address this problem, this paper proposes the development of a device called Alpha-Trash, designed to be mounted on standard curbside trash cans. Alpha-Trash automates the sorting process by utilizing a retrained convolutional neural network, specifically

Inceptionv1, which achieves an impressive 94 percent accuracy in classifying trash within a quick 4.2 seconds per classification [5].

2.5 Computer Vision Techniques for Trash Categorization

Waste management techniques utilizing computer vision to sort garbage into recycling categories have shown promise. This project focuses on categorizing trash images into four categories: glass, paper, metal, and plastic [6]. A garbage image database consisting of approximately 400 images for each class is utilized. The experiments involve employing pre-trained VGG-16, AlexNet, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Random Forest models (RF). Results indicate that using VGG-16, with its potential accuracy of 93 percent, is an effective strategy for addressing the issue. However, CNN approaches typically require better computational resources due to their higher computational complexity. Further research will explore techniques such as augmentation and fine-tuning to improve the accuracy of CNN approaches, which tend to yield better results with larger datasets [7].

2.6 Overview of Waste Classification Methods

It discusses the advantages of using Vision Transformers (ViT) over Convolutional Neural Networks (CNN) for waste classification [8]. It explains the limitations of CNN and the benefits of ViT, such as reduced computational resources and the ability to handle inputs of different sizes. It also mentions the use of ViT for waste classification, where the whole image is divided into patches and processed by transformers [9].

2.7 Deep Learning Applications for Trash Identification and Recycling

The primary objective of this study is to develop a deep-learning application capable of identifying various types of trash and facilitating recycling through a vision system. The training and testing will be conducted on an image dataset containing different classes of garbage. The dataset consists of 2527 images, divided into six classes, with 70 percent used for training and the remaining used for testing [10]. Additionally, transfer learning is employed to achieve shorter training and testing procedures with higher accuracy. Models such as Alexie, VGG16, Googlenet, and Resnet are utilized as refined models. The performance of the classifiers is assessed using two different classifiers, Softmax and Support Vector Machines (SVM). The GoogleNet+SVM combination accurately categorizes six different types of trash images, achieving the highest accuracy of 97.86 percent [11].

2.8 Intelligent Waste Material Classification for Efficient Waste Management

The buildup of solid waste in urban areas poses a significant concern as it can lead to environmental pollution and health risks [12]. An advanced waste management system is crucial for effectively handling diverse waste materials. In this context, an intelligent waste material classification system is proposed, incorporating the ResNet-50 Convolutional Neural Network model as an extractor and Support Vector Machines (SVM) for waste categorization, including glass, metal, paper, plastic, and more. The proposed system achieves an accuracy of 87 percent when tested on the trash image dataset created by Gary Thung and Mindy Yang. Implementing this waste material classification system would expedite and optimize waste separation processes, potentially reducing or eliminating the need for extensive human involvement [13].

2.9 Innovative Technologies for Garbage Detection and Monitoring

Maintaining clean and hygienic public spaces is a challenging task, especially in developing nations[14]. In response to this challenge, a novel smartphone application named SpotGarbage is introduced in this paper. SpotGarbage empowers residents to monitor and report on their communities by detecting and roughly segmenting garbage regions in geo-tagged images clicked by users. The application utilizes fully convolutional networks, an advanced deep architecture, for garbage detection.[4] After being trained on the recently released Garbage In Images (GINI) dataset, the model achieves a mean accuracy of 87.69 percent. Furthermore, the paper suggests network architecture improvements that reduce prediction times by 96.8 percent and memory usage by 87.9 percent, enabling near-real-time garbage image classification. However, there is room for further improvements in prediction time reduction.[15]

2.10 Enhancing Recycling Efficiency through Intelligent Systems

A sustainable economy relies on efficient waste management and recycling practices. This study highlights the effectiveness of contemporary intelligent approaches by experimenting with well-known deep convolutional neural network architectures.[16] Inception-ResNet and Inception-v4 exhibit superior performance, achieving 90 percent test accuracy without pre-trained weights. However, these networks have slightly longer prediction times. To address this, the connection patterns of skip connections within dense blocks were modified to improve prediction outcomes. The RecycleNet model employs a carefully optimized deep convolutional neural network architecture to classify specific recyclable object classes. Through this innovation, the parameters of a 121-layered network were reduced from 7 million to approximately 3 million.[17]

2.11 Deep Learning-Based Classification Model with self-Monitoring Module

Moreover, a deep learning-based classification model for recyclable waste images is proposed in this paper. The model enhances the residual network architecture by incorporating a self-monitoring module.[18] This enhancement allows the model to integrate relevant features from all channel graphs, compress spatial dimension features, and have a global receptive field. By maintaining a constant number of channels, the model enhances the feature map's ability to represent data and automatically extract features from different types of waste-related images. The proposed model is evaluated using the TrashNet dataset, achieving an image classification precision of 95.87 percent, outperforming other algorithms.[19]

2.12 A Deep Convolutional Neural Network for Garbage Classification

Furthermore, a deep convolutional neural network-based garbage classification model called X-DenseNet is developed in this paper to efficiently tackle the garbage classification issue. X-DenseNet is based on the Xception network, incorporating dense connections and multi-scale feature fusion inspired by DenseNet [20]. Experimental results demonstrate the model's effectiveness, achieving an accuracy of 94.1 percent on the testing set, surpassing some traditional classification networks. The proposed X-DenseNet automatic garbage classification model significantly reduces manual efforts and enhances the garbage recovery rate, offering both scientific importance and practical value.[21]

2.13 A Deep Learning Approach for Trash Classification

To address this challenge, this paper proposes a Deep Learning model based on the EfficientNet Architecture, offering high accuracy in classifying various types of trash with a relatively smaller number of parameters compared to existing methods [22]. The objective is to develop a viable solution for the industry. The model is evaluated on the industry-standard TrashNet dataset, yielding an impressive 98 percent accuracy that outperforms other models. To address the absence of a large and diverse trash image dataset, a new dataset containing 8135 images is created by combining different datasets and standardizing them. With EfficientNet B3, a classification accuracy of 92.87 percent is achieved. This study aims to increase the precision of waste sorting and introduce intelligent waste classification using mobile devices and computer vision.[23]

2.14 Comapartive Analysis of Deep Learning Models for Garbage Classification

To further explore the capabilities of deep learning models in trash classification, this paper presents a comparative study of different deep learning models, including Inception-v3, Inception-ResNet-v2, ResNet-50, and DenseNet-201 [24]. The models are trained and evaluated on a garbage dataset consisting of images belonging to six classes. The results show that Inception-v3 achieves the highest accuracy of 93.74 percent, followed closely by Inception-ResNet-v2 with an accuracy of 93.61 percent. The study emphasizes the effectiveness of deep learning models in automating garbage classification and emphasizes the need for further research to optimize and refine these models for real-world applications [25].

2.15 Challenges in Trash Classification for Effective Recycling

In recent years, the growing awareness of individuals about their impact on the environment has highlighted the urgent need for action. Recycling has emerged as a popular approach to mitigate environmental damage. However, the recycling industry has not witnessed significant advancements, and long-standing issues persist. [26] The fundamental problem lies in trash classification since recycling is only possible when waste is properly categorized. Manual classification often leads to misclassification due to subjective judgments based on personal experiences and knowledge. Additionally, direct contact with toxic waste poses physical risks to those involved. It is crucial to find a solution to this issue for the recycling industry to keep pace with the expanding recycling culture.[27]

2.16 Advancements in Waste Classification and Image Recognition Techniques

Solid Domestic Waste is generated in huge amounts daily, on a global level. This waste needs to be discarded and disposed of correctly in order to reduce environmental pollution. But most of the time, the waste isn't attended to properly, which leads to many problems [28]. Discarding waste in the soil means we are leaving our lands and environment in a hazardous situation because inorganic matter takes a lot of time to decompose in the soil.

In order to discard the waste in appropriate places, methods and different technologies have been implemented. RFID method was majorly used for the recycling process, where the RFID connector used Wi-Fi to capture and provide data regarding the trash, which helped the trash management companies in supervising and tracing the waste in a much better way. Since this technique was costly and not very flexible, researchers were looking for an effective technique that would be cost-efficient, flexible,

and reliable. The main goal of this paper is to build a pipeline that is good enough to perform waste segregation without making use of very complex models of computer vision with a large number of parameters [29].

In this paper, GANs were used to perform classification. GANs are networks that are mostly used for data augmentation. They work on two sub-models, the generative model, and the discriminator. The generative model captures the distribution of the dataset, and the discriminator classifies whether the sample is in accordance with the distribution of the dataset or not. GANs were quite effective but difficult to train. Therefore, DC-GAN (Deep Convolutional Generative Adversarial Networks) was introduced, which were standardized and much more stable. The comparison was made between traditional data augmentation techniques and GANs. The dataset on which the techniques were implemented proved that the results of GANs were much more accurate as compared to the traditional data augmentation techniques. With object classification being quite old, a better approach for object and image classification is CNN (Convolutional Neural Network). Image or object classification broadly comprises two components, the model used that will be tested, and the dataset that will be used for training and testing of the model [30].

CNN can be categorized into two types when talking about object classification, region proposal-based and regression/classification-based. The latter performs better because it has a lower computational cost and its accuracy is somehow similar to other CNN models.[31] In the region proposal model, the image is divided into small blocks, which helps in the recognition of multiple objects of a different kind in one image.

R-CNN has been used for waste sorting and has gained good results. For detecting objects in the image through CNN, the process is divided into several steps where detecting accurate object boundaries is one task. YOLO (You Only Look Once) is an algorithm in CNN that proposes an accurate box boundary for an object in the image. To achieve favorable classification performance with a sizable dataset, one can employ a range of techniques, including data augmentation pipelines, stacking routines, and utilizing feature vectors generated through transfer learning. These approaches serve

to enhance the quality and diversity of the data, ultimately improving the accuracy of classification outcomes.

Data augmentation is one technique to improve the performance of the models, and it is majorly achieved by GANs, where augmentation is done by cropping or rotating images. An even better approach in AC-GAN is where hard labels are changed to soft labels, and the labels are flipped randomly in the dataset to improve the performance of the generator. Deconvolutional layers are suggested to be added at the end of the network because adding them at the start won't help since the starting layers are still learning [32].

2.17 Challenges and Potential of Deep Learning in Waste Classification

In the context of waste management, deep learning techniques have demonstrated significant potential in waste classification. However, there are several challenges that need to be addressed for the effective implementation of these techniques. One major challenge is the availability of large and diverse datasets for training deep learning models. The performance of deep learning models heavily relies on the quality and quantity of data used for training. Therefore, efforts should be made to create comprehensive and representative datasets that cover various waste types and variations in appearance[33].

Another challenge is the computational requirements of deep learning models. Deep neural networks are computationally intensive, requiring substantial processing power and memory resources. To make deep learning models accessible and practical for waste classification tasks, optimization techniques such as model compression, quantization, and efficient hardware utilization should be explored. This will enable the deployment of deep learning models on resource-constrained devices and systems.

Moreover, the interpretability of deep learning models is an ongoing challenge [14]. Understanding the decision-making process of deep learning models is crucial for

building trust and confidence in their classifications. Research efforts should focus on developing explainable AI techniques to provide insights into how deep learning models make predictions and classifications in the context of waste classification[34]. Lastly, the integration of waste classification systems with existing waste management infrastructure is essential for practical implementation. Collaborative efforts between researchers, waste management authorities, and technology developers are necessary to ensure seamless integration, scalability, and sustainability of waste classification systems [25].

2.18 Conclusion: Progress in Waste Classification and Recycling Efforts

In conclusion, deep learning techniques have demonstrated promising results in waste classification and recycling. The utilization of advanced convolutional neural network architectures, such as EfficientNet, Inception, ResNet, and DenseNet, has shown high accuracy in classifying different types of waste materials [11]. These intelligent systems have the potential to revolutionize waste management practices by automating the sorting and segregation process. However, there are several challenges that need to be addressed, including the availability of diverse datasets, computational requirements, interpretability of models, and integration with existing waste management infrastructure. Overcoming these challenges will pave the way for the widespread adoption of deep learning techniques in waste classification and contribute to building a sustainable and environmentally friendly future [35].

Chapter 3

Software Requirement Specification

The subsequent sections of the Software Requirement Specification (SRS) is a critical part of this thesis, as it focuses on defining the specific requirements for the development of TrashBot, a waste management system. The SRS serves as a comprehensive document that outlines the functionalities, features, and constraints of the system. It specifies the desired behavior of TrashBot and acts as a roadmap for its design, development, and implementation. This chapter will detail the user requirements, functional requirements, non-functional requirements, and system constraints of TrashBot, ensuring that the software is designed to effectively address the challenges of waste management. By providing a clear and concise SRS, we aim to ensure that TrashBot meets the needs of waste management authorities, streamlines waste collection processes, and contributes to a cleaner and more sustainable environment.

3.1 Problem description

Disposal and recycling of trash are one of the main challenges of the present era. No proper disposal can result in heaps of rubbish which can be very harmful to our environment. The proposed solution to this problem is to construct an automated trash sorter bin that can segregate the trash without any error and with much ac-

curacy. Previously, the process of sorting garbage was carried out manually, which was tedious and perilous work, since it contained harmful and hazardous wastes. The trash sorter bin will fix the problem and prevent the garbage from turning into heaps. Also, sorted trash is much easier to be disposed of. We will classify the garbage into different categories and will train the models on those classes. Once trained we can then use it to classify the trash and can prevent it from turning into harmful waste for our environment.

3.2 Purpose

The purpose of this chapter is to describe the requirements for TrashBot. TrashBot is an automated bin that will aim to segregate waste into different categories. TrashBot application will generate reports on a daily and monthly basis, fabricate waste, and increase recyclability. The bin is designed for:

- Automating Waste Segregation Process
- Efficient time management
- Increased accuracy for sorting waste
- Minimized human intervention

3.3 Intended Audience and Reading Suggestions

This chapter is intended for developers who aim to design a system dealing with trash bin segregation. Project managers who need to know details about this bin and its application. Marketing staff to highlight the significance and importance of this bin in household use. Users who will use the bin. Documentation writers who aim to write a detailed requirement report.

3.4 Product Scope

TrashBot bin is designed to automate the process of waste segregation. The TrashBot application will generate daily and monthly reports of waste being segregated. These reports will help in recycling and disposing of waste. The aim of TrashBot is to achieve waste classification by automating the waste segregation process. Our goal is to design a bin that is time efficient, accurate, and less risky to human life and the environment. We will implement deep learning techniques to solve environmental problems like waste accumulation and segregation which eventually leads to pollution. We will design a bin for the segregation of waste into several categories which will then help in the fabrication and recyclability of waste. Our main objectives will be:

- To collect datasets of waste
- To train model
- To achieve the highest accuracy
- To deploy our model
- To integrate the bin with the application

3.5 Overall Description

The "Overall Description" section within the "Software Requirement Specification" chapter provides an overarching view of the TrashBot waste management system in relation to the specifications outlined in the SRS. It presents a comprehensive understanding of the system's purpose, its intended users, and the environment in which it will operate. This section describes the high-level functionalities of TrashBot, including its ability to automate waste collection, sorting, and disposal processes. It also outlines the external interfaces and constraints that will influence the system's design and implementation. By capturing the overall context of TrashBot, the "Overall Description" section sets the foundation for the subsequent detailed requirements,

ensuring that the software development process aligns with the specific needs and objectives of the waste management system. Through this comprehensive analysis, the chapter establishes a clear understanding of the TrashBot system and provides valuable insights for the subsequent phases of development and implementation.

3.5.1 Product Perspective

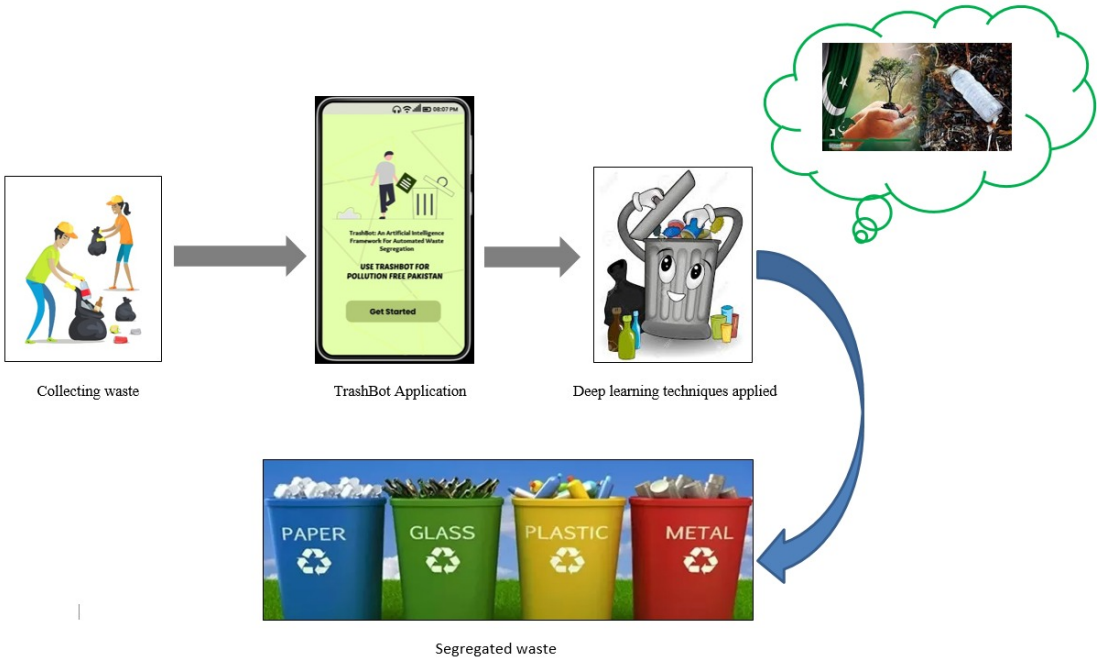

Figure 3.1: The workflow of TrashBot application representing the functionality of an application

TrashBot bin is a new product. The TrashBot application will help the user to know when the bin will be full of garbage by generating an alert message on the phone. Trash details (amount of each waste category produced) will be provided after the segregation of waste into different categories. Daily and monthly trash reports will be generated on the user's request to analyze, control and dispose of the waste properly. The application will show the progress of the waste being segregated. It will show the category of waste produced largely. The application will provide an activity view through the calendar. This view will enable the user to see which type of waste is

produced mostly each day. The application will provide a feature of a scheduled bin, through which the user will be able to book a bin according to their feasibility. Waste will be collected from their home at the scheduled time.

The TrashBot bin itself will segregate the waste into its respective categories. The model will be trained on deep learning techniques to gain the highest accuracy. The waste will be recognized through the camera and then it will be classified, based on the model training.

3.5.2 Product Functions

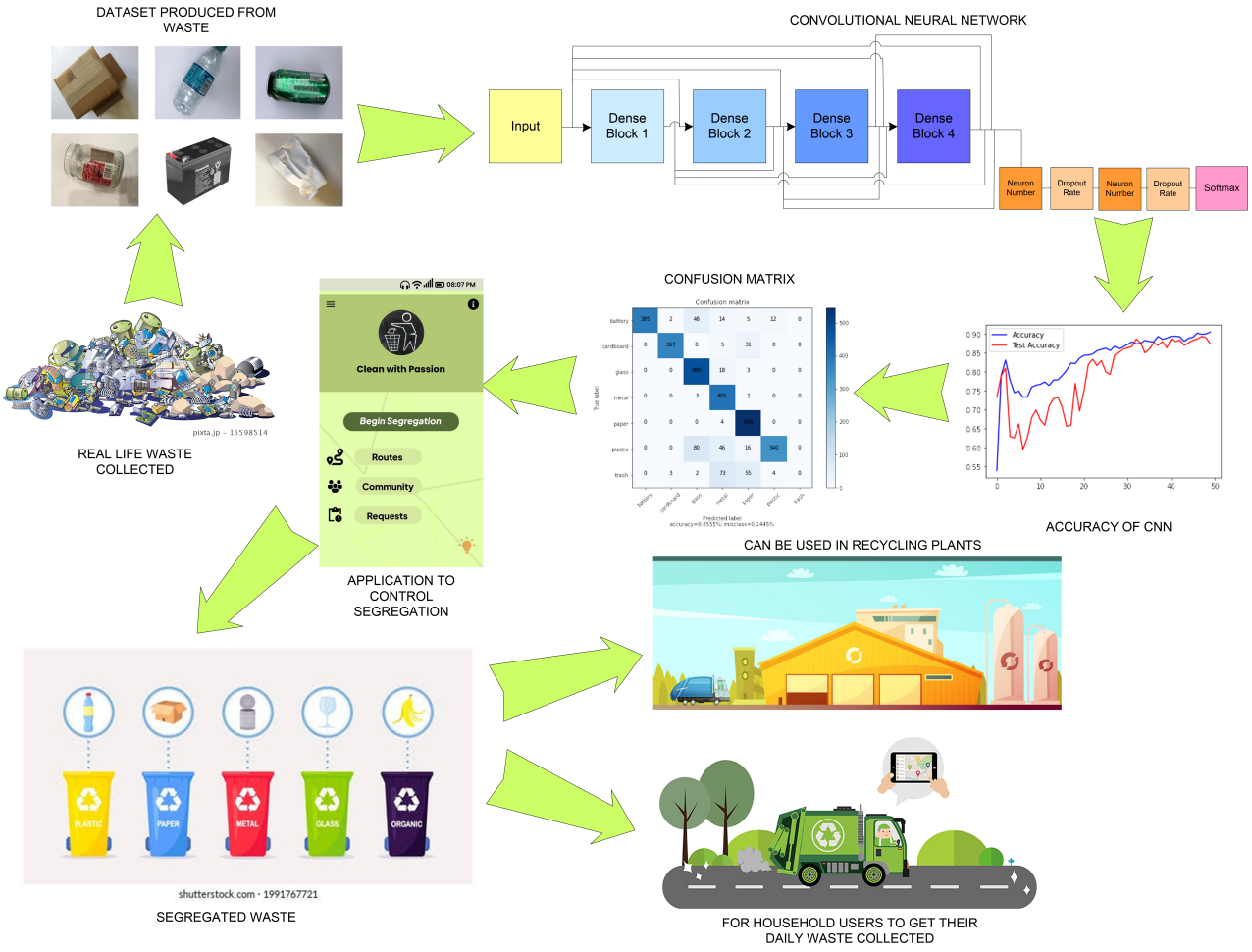

Figure 3.2: Block Diagram of TrashBot application that represents the overall methadology adopted by the application using CNN

The following items outline the primary functionalities that the TrashBot application will execute:

- It will control the process of waste segregation in the bin
- It will segregate waste into seven categories i.e. glass, battery, cardboard, metal, plastic, paper, and trash
- It will provide the daily and monthly reports of the waste being segregated on the user's request
- It will provide the feature of a scheduled bin to facilitate the user
- It will provide the feature of news through which the user will be able to know how the produced garbage is harming our environment. The latest news about the climate and environment will be listed.
- It will provide an activity view feature through the calendar which will enable the user to view the waste produced on each day

3.5.3 User classes and Characteristics

There are two types of users who will use the TrashBot application. These users and their characteristics are discussed below:

- **Type-1 user** will be household users. They will use the TrashBot bin and application to have segregated waste which will be easy to dispose of using the scheduled bin feature. Despite throwing garbage into the open landfill area. Household users will be able to know the latest climatic and environmental harms due to heaps of garbage.
- **Type-2 user** will be the employees of the waste management industry. They can analyze the daily and monthly reports to control the production of waste produced largely. Through these reports, they can take initiatives to dispose of the waste being segregated. The employees can use the activity view feature through the calendar to analyze waste production.

The educational level of the household users should be Matric, FSC, or should be comfortable with the English language. Smartphone users familiar with such types of applications can easily use this TrashBot application. The technical expertise of the waste management industry employee should be of middle or high level. They should have knowledge of using such bins and applications. Type-1 users are more important to satisfy their requirements because they will use the bin for household and commercial users. They will control and manage the bin through the TrashBot application.

3.5.4 Operating Environment

TrashBot application is an Android application. It is created using the Flutter platform. This application is designed for the Windows operating system. My-SQL is used to store all the daily and monthly reports of the segregated waste. Users' data will also be stored in the database. TrashBot application will ensure the successful segregation of the waste given to it. TrashBot will consist of a camera through which it will scan the waste and then it will classify the waste based on its training. TrashBot bin and the application are integrated, they work together with each other.
The user should have the following specifications to use TrashBot bin and its application:

- Connectivity to Wi-Fi or mobile network
- Convenient touch screen or keypad interface
- Continuous power supply
- 600 MHz processor speed
- Utilization of mobile cameras for various functionalities
- User input functionality
- Minimum 530MB RAM requirement
- Compatibility with Android and iOS platforms

3.5.5 Design and Implementation Constraints

Interruption in Wi-Fi or power supply, an error message will be generated as "Connection not available". The segregation process will be paused. Software security should be maintained in order to control unauthorized access to software in the waste management industry. TrashBot application can be used both on Android and iOS smartphones. Devices having RAM less than 530MB RAM can not operate the TrashBot application. The TrashBot application will be developed using Flutter and MySQL databases. CNN model will be used for image classification written in Python language. Basic knowledge of English is required to use the TrashBot application.

3.5.6 User Documentation

For user manuals and help, the user can select the help option from the main menu to see the instructions and guidance. Users can also select the option of contacting us. They can freely contact us if they have any queries regarding the system. They will be entertained timely.

3.5.7 Assumptions and Dependencies

It is assumed that a good internet connection and power supply will be available to run the system otherwise the system will have to suffer. There is no dependency because we are not using any software component of another project.

3.6 External Interface Requirements

This focuses on the interactions and interfaces that the TrashBot waste management system will have with external entities. These entities may include users, hardware devices, software systems, or external services. This section outlines the specific requirements and functionalities of these interfaces, such as the user interface design, communication protocols, data exchange formats, and integration with external systems. By specifying the external interface requirements in the SRS, it ensures that

the TrashBot system can seamlessly interact with its environment, enabling efficient communication, data transfer, and integration. This section plays a crucial role in defining the boundaries and integration points of TrashBot, providing a clear understanding of how the system will interact with its external stakeholders, devices, and systems.

3.6.1 User Interfaces

Household users should have a menu option on the home screen. Through the menu option, they will be able to select different options like news, activity view, schedule bin, and help. The GUI of the TrashBot application should be user-friendly.
A user interface to control the process of waste segregation is required. Error message "Connection Lost" should be displayed if the internet connection is disturbed. The employee should know the details of the waste being segregated through GUI. They should have the option to request daily and monthly reports of the waste being segregated. Options like activity view and help should be available through the menu.

3.6.2 Hardware Interfaces

The hardware part of the TrashBot will be integrated with the TrashBot application. The trained CNN model will be deployed at the back end of the software application. TrashBot will scan the waste, and pass the information to the trained model, the trained model will identify the waste category and will send back the information to TrashBot. Then the TrashBot will segregate the waste respectively.

3.6.3 Software Interfaces

The TrashBot application will be connected to the MySQL database and will be integrated with the hardware part. The CNN model written in a Python programming language will be used for image classification. The application will receive the image of the waste to be classified. The information regarding the waste category to be

classified will be sent to TrashBot by the application.
Following are the software interfaces used in the system

- Operating System: Android and iOS
- Front End: Dart programming language (Flutter platform)
- Database: MySQL

3.6.4 Communications Interfaces

The system utilizes an internet connection to establish a connection with the database, execute the waste segregation process, and generate daily and monthly reports regarding the segregated waste.

3.7 System Features

Following are the features provided by the TrashBot application. The priority rate of features is rated from a low of 1 to a high of 5.

3.7.1 System Feature 1

Feature 1 of the TrashBot application is that it will **control the process of waste segregation.**

Description and Priority

This feature will enable the employee of the waste management industry to control and monitor the process of waste segregation. Trash details will be displayed on the application during the waste segregation process.
Priority rate of feature: **5**

Stimulus/Response Sequences

The user will select the "Begin segregation" option from the menu. The waste segregation will begin. TrashBot will recognize the waste through the camera and will send this information to the TrashBot application. The trained model will identify the waste category and will send this information back to the TrashBot. The waste will be then thrown into its respective bin.

Functional Requirements

REQ-1: The details of the waste being segregated should be displayed on the screen during the whole process
REQ-2: Employee should be able to pause and stop the segregation process
REQ-3: The waste should be segregated with a minimum accuracy of 95
REQ-4: The process should be efficient

3.7.2 System Feature 2

Feature 2 of the TrashBot application is that it will **segregate the waste into its respective bin**. The categories include glass, plastic, metal, cardboard, battery, paper, and trash.

Description and Priority

The TrashBot application will send information to TrashBot about the category of waste identified through the camera. The TrashBot will then throw the waste into its respective bin accordingly.
Priority rate of feature: **5**

Stimulus/Response Sequences

Users will select the option of "Begin Segregation" from the home page. The process of waste segregation will begin. The TrashBot will identify the waste, classify it, and then it will throw the waste into its respective bin.

Functional Requirements

REQ-1: The waste should be efficiently segregated

REQ-2: Segregated waste should have an accuracy of 95

REQ-3: The database should record all the information about the segregated waste

3.7.3 System Feature 3

Feature 3 of the TrashBot application is **generating daily and monthly reports of the waste** being segregated.

Priority rate of feature: **4**

Description and Priority

This feature will enable the employees to analyze the daily and monthly reports of the waste being segregated. This data will help them to take initiatives to reduce waste and dispose of it in a more good way.

Stimulus/Response Sequences

The employee will select the option of reports from the menu. Another menu box will open in front of the user which will let the employees either select the option of daily reports or monthly reports or both. The user request will be generated and the reports from the database will be fetched. Finally, the employee will get those reports for analysis.

Functional Requirements

REQ-1: Reports generation should not take time more than 5 minutes

REQ-2: Backup of the files should be maintained

REQ-3: The database should be maintained monthly

3.7.4 System Feature 4

Feature 4 of the TrashBot application is the **scheduled bin.**

Priority rate of feature: **5**

Description and Priority

The feature will enable household users to schedule a bin according to their feasibility.

Stimulus/Response Sequences

The user will select the option of schedule bin from the main menu on the home page. The user will then select the time and date on which their waste will be picked from their home by a waste management truck. This feature will make citizens more civilized and will prevent the throwing of waste into open lands.

Functional Requirements

REQ-1: GUI should be user-friendly and easy to use

REQ-2: Waste should be collected at the scheduled time

REQ-3: The database should be maintained monthly

3.7.5 System Feature 5

Feature 5 of the TrashBot application is the **News**

Priority rate of feature: **4**

Description and Priority

This feature will display the latest environmental and climatic changes caused by heaps of garbage. The users will come to know the importance of proper disposal of waste.

Stimulus/Response Sequences

The user will select the option of News from the menu. After selecting this option, a list of the latest harms caused by waste will e displayed. The users can view and read them.

Functional Requirements

REQ-1: Harms displayed should be the latest

REQ-2: It should not take too much time in loading

3.7.6 System Feature 6

Feature 6 of the TrashBot application is the **Activity View.**

Priority rate of feature: **3**

Description and Priority

This feature will allow the user to view the waste produced on a daily basis through a calendar. Different symbols will be used used for different categories of waste. Respective symbols will be shown on the calendar according to their productivity on their respective date. This view will enable users and employees to analyze how the waste is produced on a monthly basis.

Stimulus/Response Sequences

The user/employee will select the option of Activity View from the menu. As a result, the calendar will be opened which will show the waste produced on a daily basis.

Functional Requirements

REQ-1: It should not take too much time in loading

3.8 Other Nonfunctional Requirements

It addresses various aspects of the TrashBot waste management system that are not directly related to its functionality but are crucial for its overall performance, security, usability, and maintainability. This section outlines specific requirements related to performance, scalability, reliability, security, user experience, documentation, and system maintenance. These non-functional requirements define the quality attributes and constraints that the TrashBot system must adhere to. By specifying these requirements in the SRS, it ensures that the TrashBot system meets the desired standards and expectations in terms of performance, security, usability, and other important aspects. This section plays a vital role in shaping the overall design and implementation of the TrashBot system, contributing to its success as a reliable and efficient waste management solution.

3.8.1 Performance Requirements

Feature 1 and Feature 2 should give minimum accuracy of 95.
They should perform efficiently without any delay.

3.8.2 Safety Requirements

Backup of the reports, users' data, and segregated waste should be maintained. Otherwise, everything will be lost due to any mishap.

3.8.3 Security Requirements

Software security is always been an important issue. Password authentication will be required to access the data from the database. Password security methods will block the access of unauthorized users to data.

3.8.4 Software Quality Attributes

The critical quality attributes that need to be met include availability, correctness, flexibility, maintainability, reliability, testability, and usability.

3.8.5 Business Rules

Household users will be able to use the following features of the TrashBot application:

- Schedule Bin
- Activity View
- News

Employees of the waste management industry will use the TrashBot and TrashBot applications to segregate waste into their respective bins.

3.9 Other Requirements

No other requirements still yet.

3.10 Appendix A: Glossary

TrashBot: The hardware part will segregate the waste into their respective bins.
TrashBot application: A software application that controls and monitors the waste segregation process and provides other useful features.

Chapter 4

Software Design

4.1 Introduction

The chapter on "Software Design" in the context of the waste management system, TrashBot, focuses on translating the system requirements into a detailed and comprehensive design. Software design specifies the architecture, components, modules, interfaces, and algorithms that form the foundation of the TrashBot system. It encompasses the overall structure and organization of the software, ensuring that it is scalable, maintainable, and extensible. The design phase considers factors such as data flow, control flow, system behavior, user interactions, and system performance. By documenting the software design for TrashBot, this chapter provides a blueprint for the development team, guiding them in implementing the system and ensuring that it meets the intended functionality, efficiency, and usability goals.

4.1.1 Purpose

The intention of the elaboration is to furnish an extensive depiction of the software system that is to be created. The design chapter captures the architectural and detailed design decisions made during SDLC's planning phase. This chapter provides a blueprint for the software development team. The chapter ensures that the software is developed in a consistent and structured manner, meeting the requirements and

objectives specified in SRS, and ensuring the required quality standards.

4.1.2 Scope

The scope of a software design chapter is typically focused on the design aspects of the software system to be developed. The scope of this chapter includes system architecture, detailed design, user interface design, data management, error handling and recovery, testing, and quality assurance. The software design chapter describes the maintenance and support requirements for the software system, including how updates and upgrades will be handled and how user support will be provided.

4.1.3 Overview

The software design chapter (SDD) is a written description of the software system design to be developed. The chapter outlines the data models, components, architecture, algorithms, and user interface of the software system. It provides a common understanding of software system design among the development team, stakeholders, and users. It helps to ensure that the software system is designed and developed in a structured and organized manner and that it meets the needs of the project stakeholders.

4.1.4 Content Summary

In this section, a comprehensive overview of the system is provided, highlighting the design considerations that have shaped the system architecture. Subsequently, a detailed description of the system architecture is presented, followed by an in-depth explanation of the system design.

4.2 System Overview

The TrashBot software application is designed to provide an integrated solution to manage waste, segregate it and dispose of it properly. The system will offer features

such as segregating waste into four categories, controlling the process of waste segregation, generating daily and monthly reports of segregated waste, scheduling bins, and news, activity view, security and maintenance, and support.

Segregating waste feature will allow the TrashBot bin to segregate the waste into its respective bin. The classified waste will be easy to recycle and dispose of. Controlling the process of waste segregation will allow the user to control the waste being segregated without any human involvement. The user will get an alert message when the bin is full of waste.

The daily and monthly reports feature will enable users to see the type and amount of waste generated by different cities or areas. This information will be used to calculate the total amount of waste generated over time, which will help organizations identify opportunities for waste reduction and recycling.

The scheduling bin feature will allow users to schedule waste collection based on predefined routes and schedules. The system will automatically generate collection schedules based on the type and amount of waste generated, as well as the location of the waste. This feature will help organizations optimize their waste collection processes and minimize waste disposal costs.

The activity view feature will allow the user to view which type of garbage is produced mostly on a daily basis. The user will view this through the calendar.

The news feature will keep the user up to date about the latest climatic and environmental changes caused due to huge garbage production and its improper disposal.

The security feature will ensure that sensitive waste management data is protected. The system will include user authentication, data encryption, and access control to prevent unauthorized access to data.

The maintenance and support feature will ensure that the software is up-to-date and running smoothly. This may include automatic updates, bug fixes, and user support. This feature will help the waste management industry to maintain the efficiency and effectiveness of their waste management processes over time.

4.3 System Architecture

It focuses on defining the overall structure and organization of the TrashBot system. It establishes the foundation for the system's components, modules, and their interconnections. The system architecture specifies the high-level design decisions, such as the choice of architectural style, the arrangement of subsystems, and the communication mechanisms between them. This section outlines how the different modules of TrashBot interact with each other, including the flow of data, control, and communication. By presenting the system architecture, this chapter provides a clear understanding of the system's structure, enabling the development team to implement and integrate the various components effectively to ensure the seamless functioning of TrashBot.

4.3.1 Architectural Design

The core architecture of the waste management system comprises several interconnected subsystems. The waste collection subsystem focuses on the collection of waste and placing it into the respective bin, monitoring its fill levels, and managing the waste collection activities. The scheduled bin subsystem ensures the timely collection of waste from the user. The controlling and monitoring subsystem allows users to view the progress of the waste segregation process. The information management subsystem stores and analyzes data to generate reports for analysis and decision-making. The user interface subsystem provides interfaces for the user and stakeholders to interact with the TrashBot application, facilitating communication and engagement. The system architecture of the TrashBot application aims to provide sustainability and ensure compliance with environmental standards and regulations.
The following UML diagrams represent high-level subsystems and the responsibilities/roles assigned to them. It is represented how the subsystems work together to achieve the desired functionality.

The use case diagram of the TrashBot application depicts various functionalities and

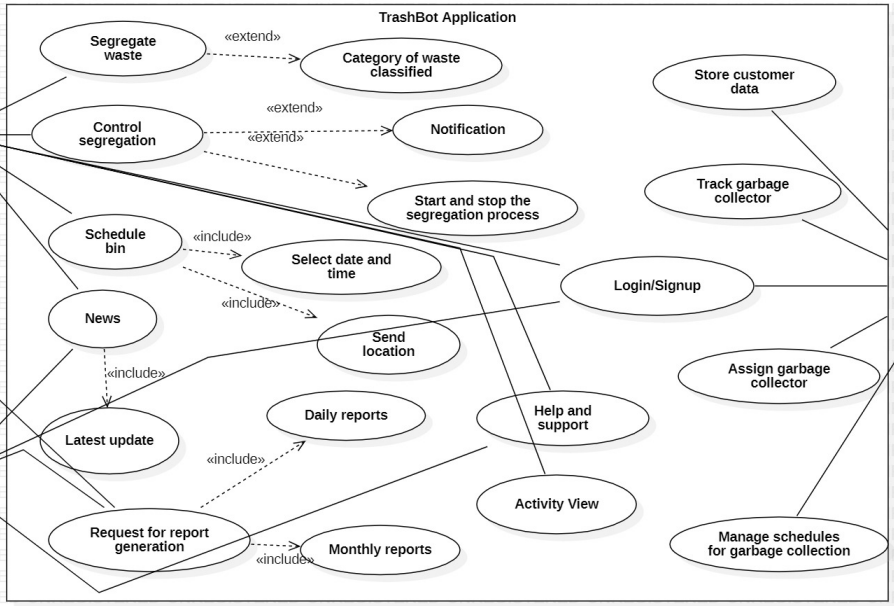

Figure 4.1: Use Case Diagram of TrashBot application describing how the users interact with the application through proper steps for waste segreagtion purpose

interactions of the system from its user perspective. The diagram typically includes actors such as household users, waste management industry employees, and IT managers. The main use cases include segregating waste, control segregation, scheduling bins, news, request for report generation, activity view, help and support. The user will schedule a bin for waste collection by providing the necessary information. IT manager will control the whole process, manage user accounts and handle any issues that arise. The use case diagram illustrates the flow of interactions between the actors and the system. It provides a high-level overview of how the TrashBot application works.

The class diagram of the TrashBot application shows different classes involved in the system and the relationships among them. The diagram includes classes such as Customer, Report generation, Control segregation, Address, Activity view, schedule bin, database, and news. Each class represents a different role in the TrashBot application. Each class has its attributes and methods, representing the behavior

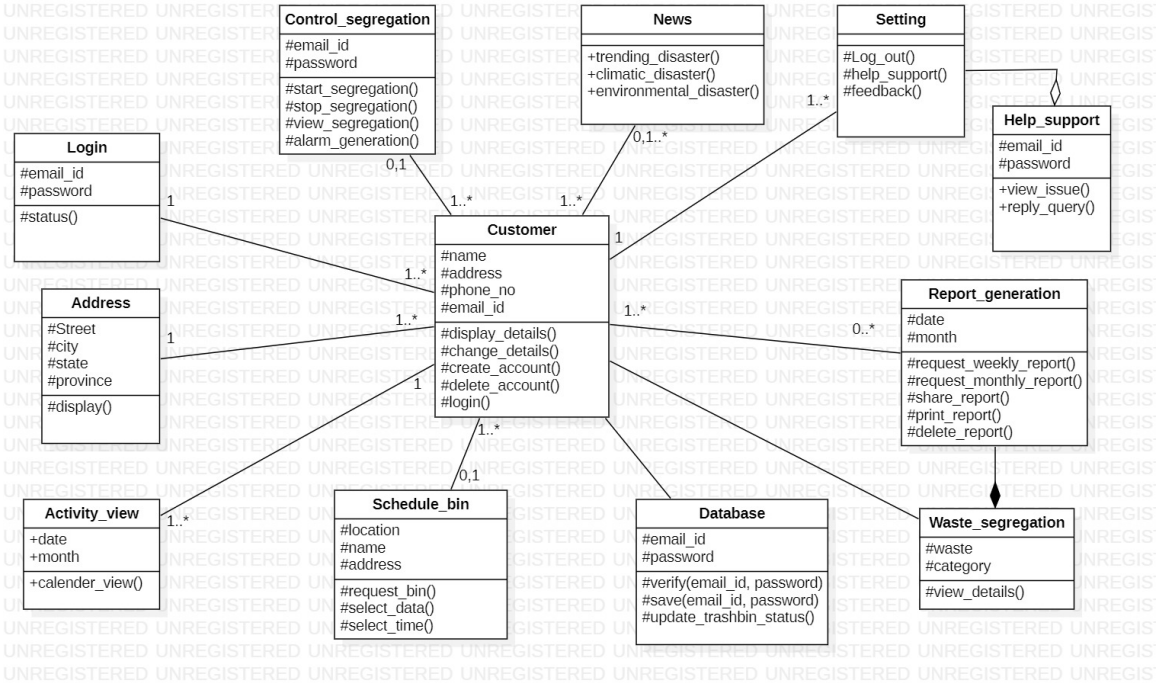

Figure 4.2: Class Diagram of TrashBot representing the different modules of the application

and properties associated with them. This diagram also highlights the relationship between classes such as association, aggregation, and inheritance.

The package diagram of the TrashBot application provides a high-level overview of the system's structure and organization. It shows the different modules of the system and how they are related to each other. The main packages include customer, user interface, and TrashBot bin. Each package in the diagram encapsulates related classes, interfaces, and other components. The package diagram shows a clear understanding of the system's organization and components.

4.3.2 Decomposition Description

The "Decomposition Description" section within the "Software Design" chapter of the waste management system thesis delves into the breakdown of the TrashBot system into smaller, more manageable components. It focuses on decomposing the overall

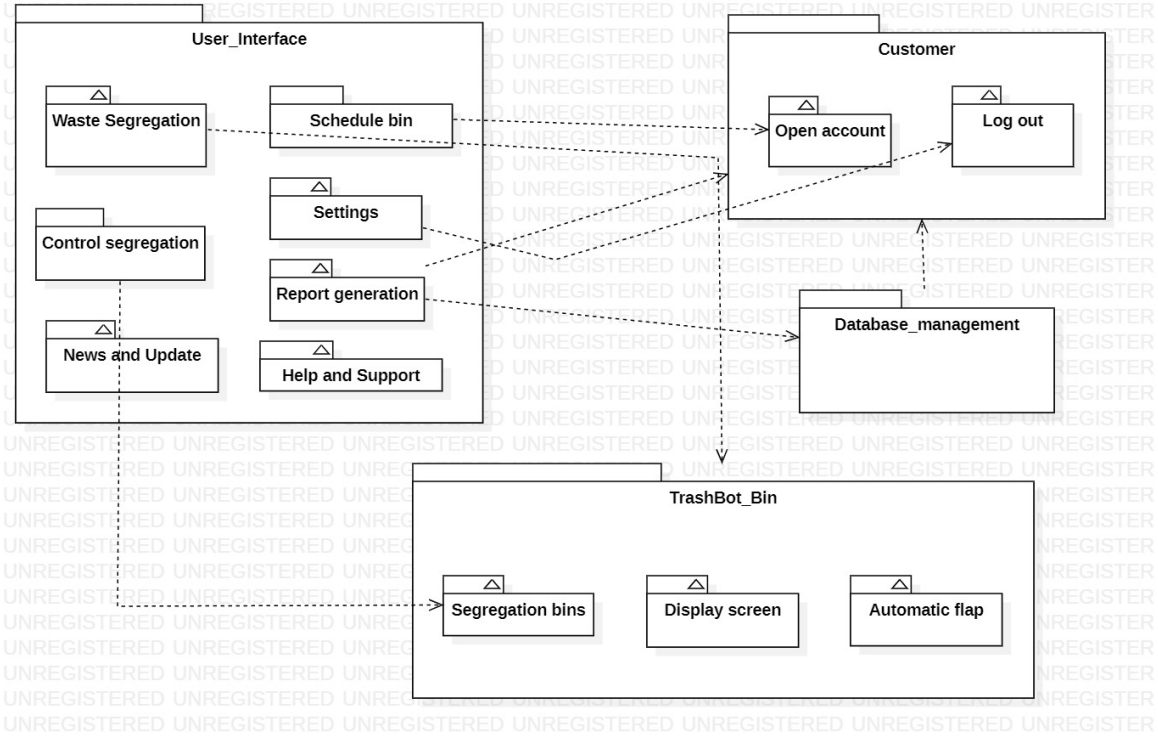

Figure 4.3: Package Diagram of TrashBot application representing the different modules of the application and their relation

system into subsystems, modules, and functions, each with its specific responsibilities and interactions. This section aims to provide a detailed description of how the system is divided into smaller units, allowing for easier development, testing, and maintenance. By decomposing the system, the design team can assign clear responsibilities to each component, facilitating better organization and modularity. This decomposition enhances the overall understandability and maintainability of the TrashBot system, ensuring that each component can be developed and tested independently while still working together cohesively towards the common goal of efficient waste management.

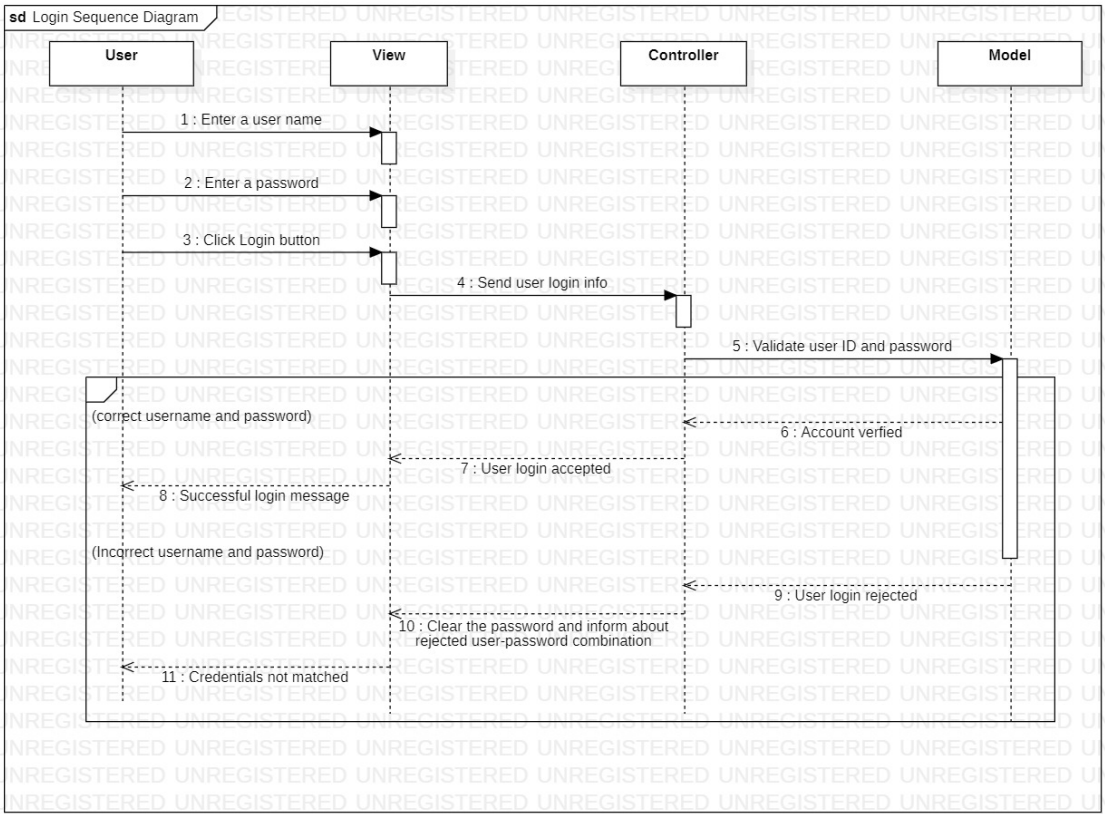

Figure 4.4: Login Sequence Diagram of TrashBot application representing the work flow of the application among different modules and the user

4.3.3 Design Rationale

We have used Model-View-Controller (MVC) architecture for our software design which separates our application into three components: the Model, the View, and the Controller. This architecture promotes independent module development and ensures easy maintenance of the application. In the MVC architecture, the user triggers some action/event by interacting with a View. The View forwards the user input to the Controller. The Controller performs some operations on the Model after receiving user input. The Model updates its state and notifies the View. The View updates its display according to the update received from the Model. The user is then able to see the updated View and continue interacting with the application.

The selection of Model-View-Controller (MVC) architecture for the TrashBot appli-

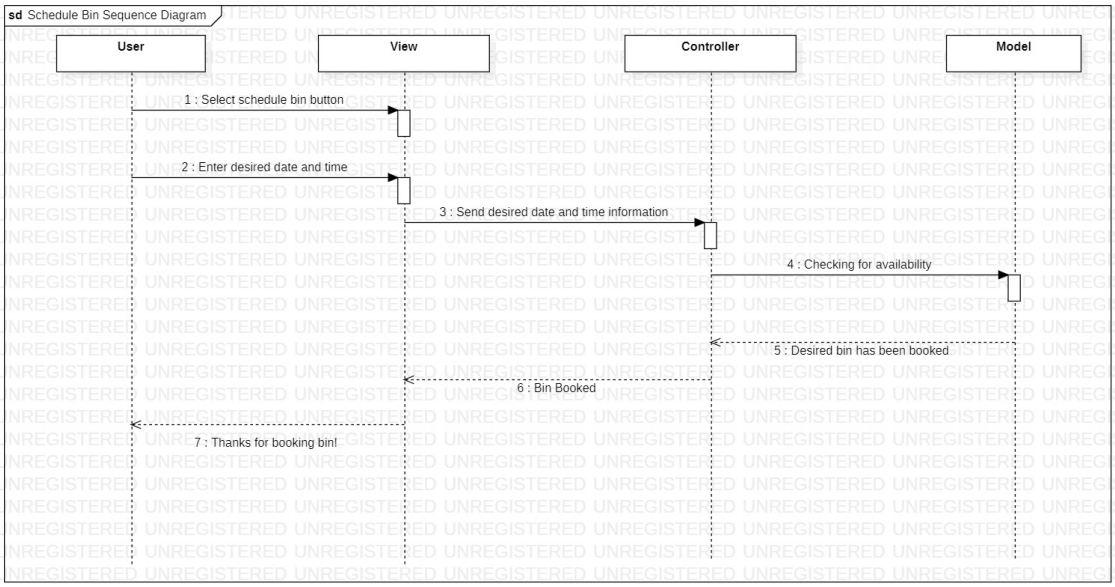

Figure 4.5: Schedule Bin Sequence Diagram of TrashBot application representing the work flow of scheduling the bin by the user

cation is driver by several factors, including the following rationale:

- **Separation of Concerns:** This architecture promotes separation concerns between different modules. The Controller handles the user input and the interaction between the View and Model. The View is responsible for the user interface. The Model encapsulates business logic and data.
- **Modularity and Reusability:** This architecture ensure the usability of the components.
- **Scalability and Maintainability:** This architecture allow one to replace or modify components without affecting other components. This flexibility makes maintenance easy.
- **User Interface Flexibility:** This architecture ensures flexible user interface.

The following are the critical issues and trade-offs while considering this architecture:

- Increased Complexity: Additional interactions and layers can increase the complexity of the system. It is very important to ensure that the interactions are

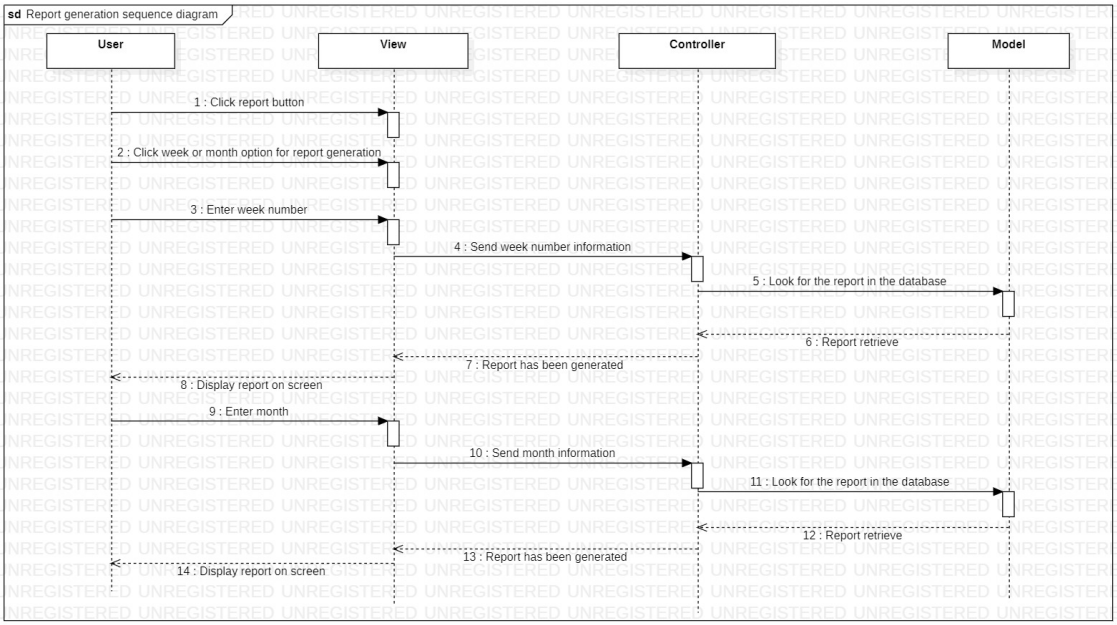

Figure 4.6: Report Generation Sequence Diagram representing the work flow of report generation by the user

well-defined and effectively managed.

- Learning Curve: This architecture requires a clear understanding of the responsibilities and interactions between the components for those who are new to this architecture.

- Performance Considerations: To minimize any impact on performance, efficient data transfer should be ensured.

- Synchronization and Data Consistency: It is important to implement data consistency and proper synchronization in the interaction.

4.4 Data Design

Data design of the TrashBot application involves structuring and organizing data entities and their relationships to efficiently manage all the information related to waste and users. The key components of TrashBot data design are:

1. **Data Entities:** The main entities in the TrashBot application include waste type, customer, and service providers. Each entity has related attributes associated with them.

2. **Data Relationships:** Establishing relationships between different entities to represent dependencies and interactions among them.

3. **User Profiles and Authentication:** To store application user information and their authentication credentials.

4. **Reporting and Analysis:** It includes the feature to generate weekly and monthly reports of the waste being segregated for analysis.

5. **Data Storage and Security:** It includes selecting a database management system (DBMS) to store and manage the TrashBot application's data.

4.4.1 Data Description

The data description for the TrashBot application is as follows:

1. **User Profile Data**

    - User ID: Unique ID of each user
    - Name: User's full name
    - Address: User's house address
    - Email: User's email address
    - Password: Password for user authentication

2. **Trash Bins Data**

    - Bin ID: Unique ID of each waste bin
    - Type: Type of waste bin like glass waste, paper waste, etc
    - Capacity: Maximum capacity of the waste bin
    - Current Fill Level: percentage or volume of waste bin

3. **Scheduled Waste Collection Data**

   - Schedule ID: Unique ID for waste collection schedule
   - Collection Date: Date on which waste collection is scheduled
   - Collection Time: Time on which waste collection is scheduled

4. **Reports Data**

   - Weekly Report ID: Unique ID for weekly report
   - Monthly Report ID: Unique ID for the monthly report
   - Month: Month for which the report is generated
   - Total Weekly Waste: Total waste segregated during the week
   - Total Monthly Waste: Total waste segregated during the month

   This data description highlights the key entities and associated attributes with them of the TrashBot application. The data will be used to manage TrashBot application operations.

4.4.2 Data Dictionary

Following are the major entities along with their types and descriptions.

User

- Type: Object
- Description: It represents the user of the TrashBot application
- Attributes: User ID, Name, Address, Email, Password

Bin

- Type: String
- Description: It represents the waste bin that will segregate the waste into its respective bin like paper waste, glass waste, etc

- Attributes: Bin ID, Capacity, Current Fill Level

Collection Schedule

- Type: Object
- Description: It represents the scheduled date and time for the waste collection from the user
- Attributes: Scheduled ID, Collection Date, Collection Time

Reports

- Type: Object
- Description: It represents the generation of weekly and monthly reports of waste being segregated for analysis
- Attributes: Weekly Report ID, Monthly Report ID, Total Weekly Waste, Month, Total Weekly Waste, Total Monthly Waste

4.5 Component Design

The component design of the TrashBot application includes a bin component, a schedule of waste collection component, a report component, and a user component. Below is the pseudocode for the TrashBot components.

1. **User Component**
   class User {
   String userID;
   String name;
   String address;
   String email;
   String password;
   function createUser(name, email, password) {

```
//Code to create a new user profile with the provided details
}

function authenticateUser(email, password) {
// Code to authenticate the user based on the provided email and password
}

function updateUserDetails(userID, name, email) {
// Code to update the details of a specific user profile
}

function deleteUser(userID) {
// Code to delete the specified user profile
}
}
```

2. **Bin Component**

```
class Bin {
String binID;
Integer capacity;
Float currentFillLevel;

function segregateWaste(wasteItem) {
// Code to segregate the waste item into the appropriate bin category
}

function updateFillLevel() {
// Code to calculate and update the current fill level of the bin
```

```
}

function isFull() {
// Code to check if the bin is full based on its current fill level and capacity
}
}
```

3. **Report Generation Component**

```
class Reporting {
String WeeklyReportID;
String MonthlyReportID;
Float TotalWeeklyWaste;
String Month;
Float TotalWeeklyWaste;
Float TotalMonthlyWaste;

function generateWeeklyReport(WeeklyReportID) {
//Code to generate a weekly report of segregated waste for the specifiedperiod
}

function generateMonthlyReport(MonthlyReportID,Month) {
//Code to generate a monthly report of segregated waste for the specified month
}
}
```

4. **Scheduling Bin Component**

```
class WasteCollectionSchedule {
String scheduleID;
Date collectionDate;
```

```
Time collectionTime;

function createSchedule( collectionDate, collectionTime) {
// Code to create a new collection schedule for the specified bin on the given
date and time
}

function cancelSchedule(scheduleID) {
// Code to cancel the specified collection schedule
}

function getScheduleDetails(scheduleID) {
// Code to retrieve the details of a specific collection schedule
}
}
```

4.6 Human Interface Design

The TrashBot application focuses on creating a user-friendly interface that is intuitive, visually appealing, and easy to navigate. Some key considerations for the human interface design of the application are clean and modern visual design, user-friendly navigation, responsive and mobile-friendly design, intuitive controls and interactions, feedback and notifications, consistent layout and organization, and accessibility considerations.

4.6.1 Overview of User Interface

The TrashBot application provides users with a range of features to control the smart TrashBot bin, schedule the bin, generate reports, and stay updated on the adverse

effects of waste on the environment and climate change. Following is the description of how users can use the system to complete the expected features and the feedback information displayed for them:

1. **TrashBot Bin Control:**

    - User can open the TrashBot application and access the bin control feature.
    - User can interact with the application interface to open or close the bin lid.
    - User can initiate the segregation process, allowing the bin to sort waste into the respective categories.
    - Feedback Information: The application provides real-time updates on the bin's status, such as the fill level of each compartment. It visually displays the bin's current state, including the segregation process in progress, ensuring the user is aware of the actions taken.

2. **Waste Collection Scheduling:**

    - User can navigate to the scheduling feature within the application.
    - User can schedule a bin for waste collection.
    - User can specify the date and time for the waste collection.
    - Feedback Information: The application confirms the scheduling request, providing a notification or confirmation message to the user. It may display the scheduled collection details, including the selected bin, collection date, and time, ensuring the user has a record of the scheduled collection.

3. **Reports Generation:**

    - User can access the reporting section of the application.
    - User can choose to generate weekly or monthly reports.
    - User can specify the desired reporting period (e.g., specific week or month).

- Feedback Information: The application generates the requested report and presents it to the user in a readable format. The report includes information on the total amount of waste segregated and a breakdown of waste by type or category. The user can view, download, or print the report for analysis or record-keeping purposes.

4. **News:**

5. User can access the environmental impact section within the application.

6. User can view the adverse effect of waste on the environment and on climate.

7. User can browse news articles, tips for waste reduction, recycling guidelines, and educational content.

8. Feedback Information: The application displays relevant information on the adverse effects of waste on the environment and climate. It provides updated news, informative content, and actionable tips to help the user stay informed and make efficient decisions about waste management.

   Throughout the user's interaction with the TrashBot application, feedback information is provided in real-time or through notifications. The application displays confirmation messages, success or error notifications, and progress indicators to keep the user informed about the status of their actions, ensuring a seamless user experience. Additionally, the application may include contextual help, or a dedicated help section to provide further guidance and support for users.

4.6.2 Screen Images

Here are some screen images of TrashBot application that shows the user interaface of an application. It shows the funtions and services that a user can take through the application. Major screens of TrashBot application are given below:

1. Main Window
2. Calendar Window
3. Register Window
4. Login Window
5. Setting Window

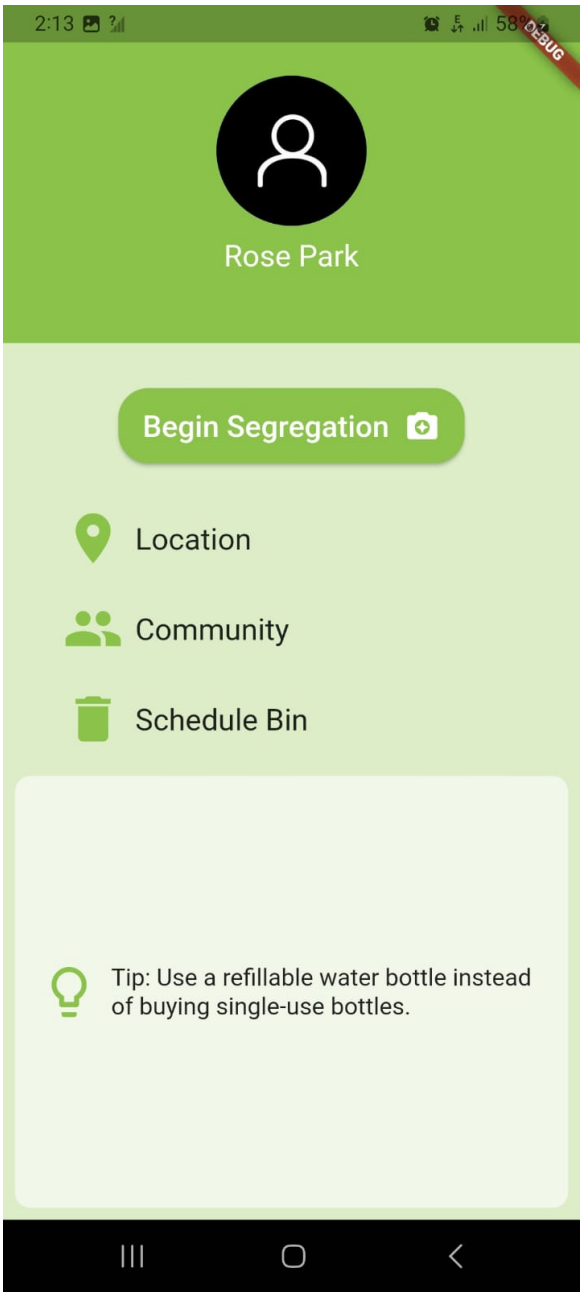

Figure 4.7: Main Window

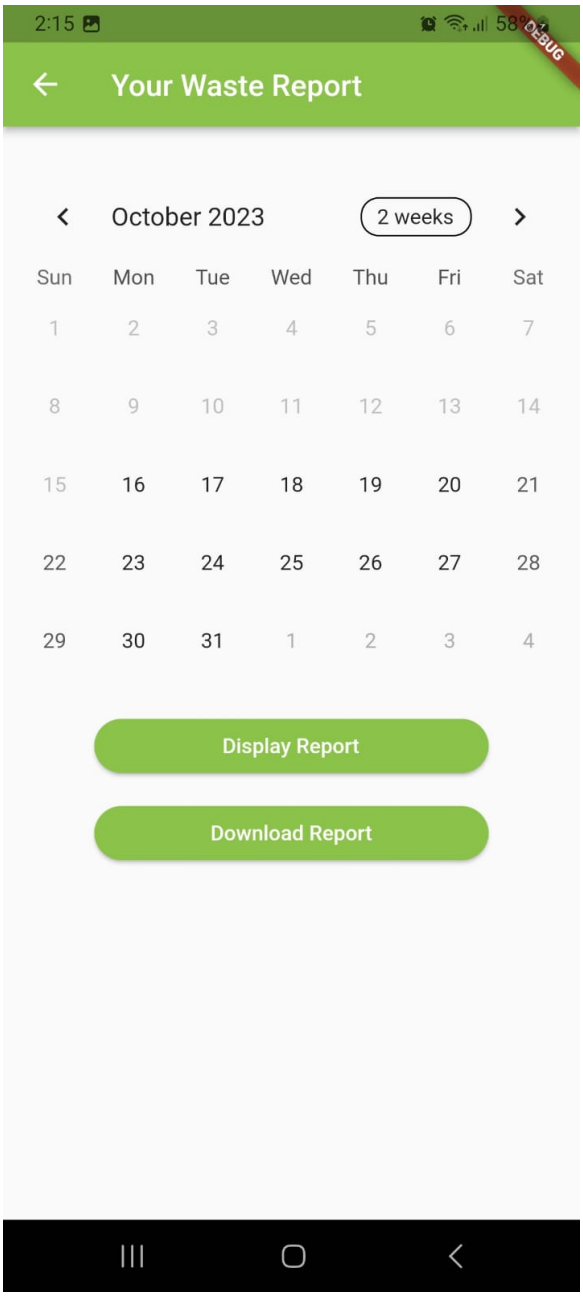

Figure 4.8: Calendar Window

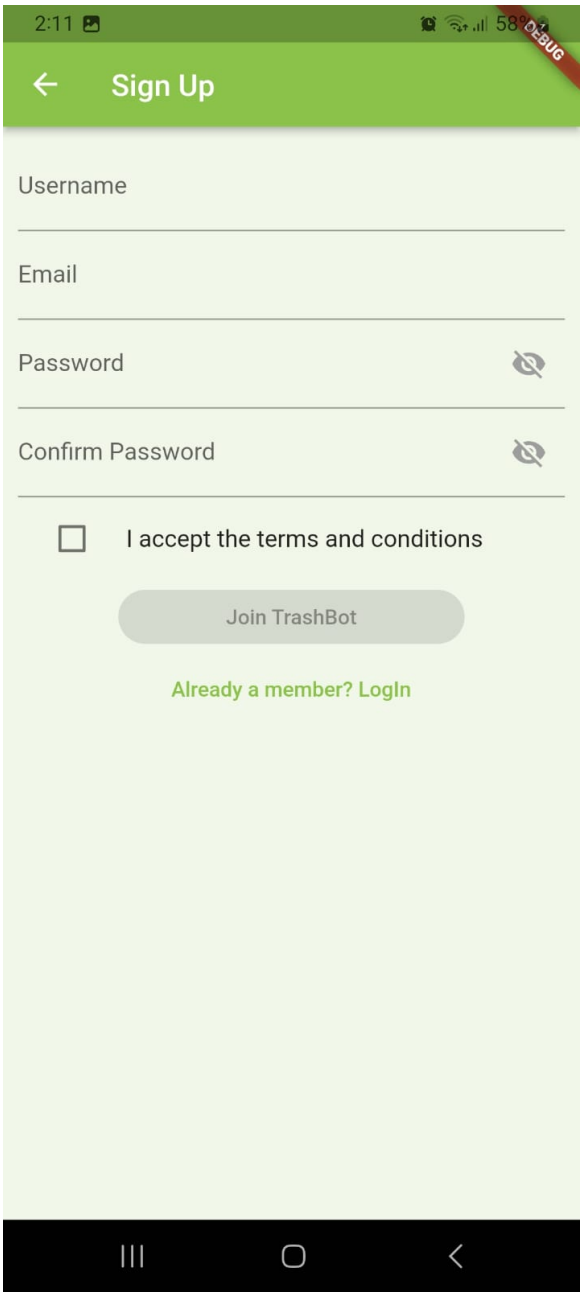

Figure 4.9: Register

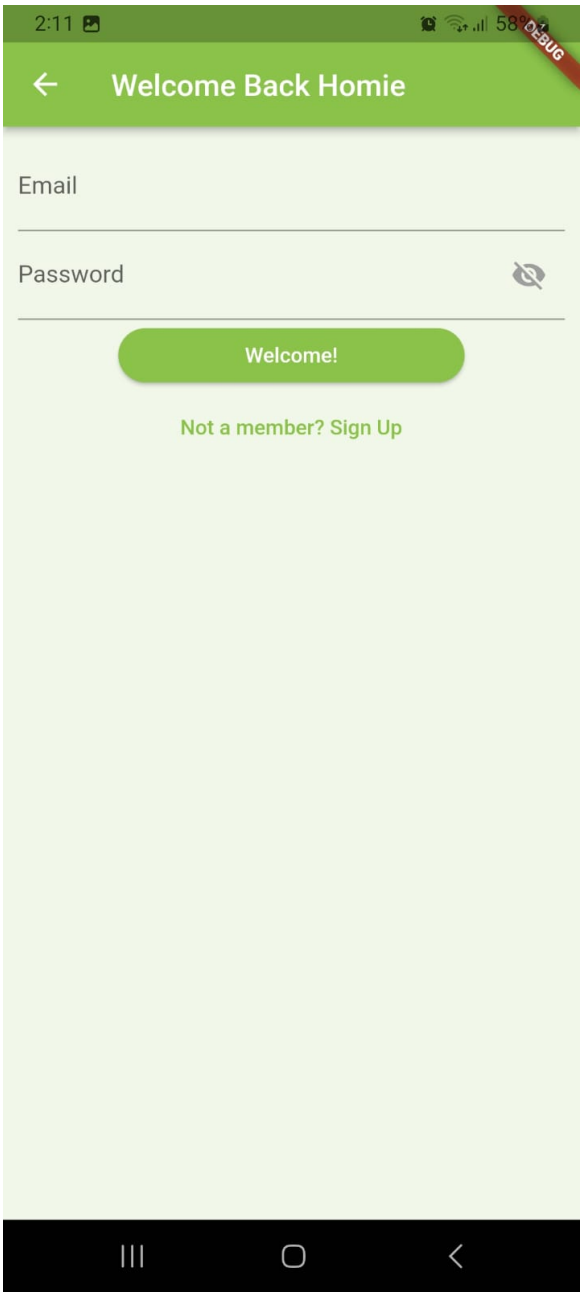

Figure 4.10: Login

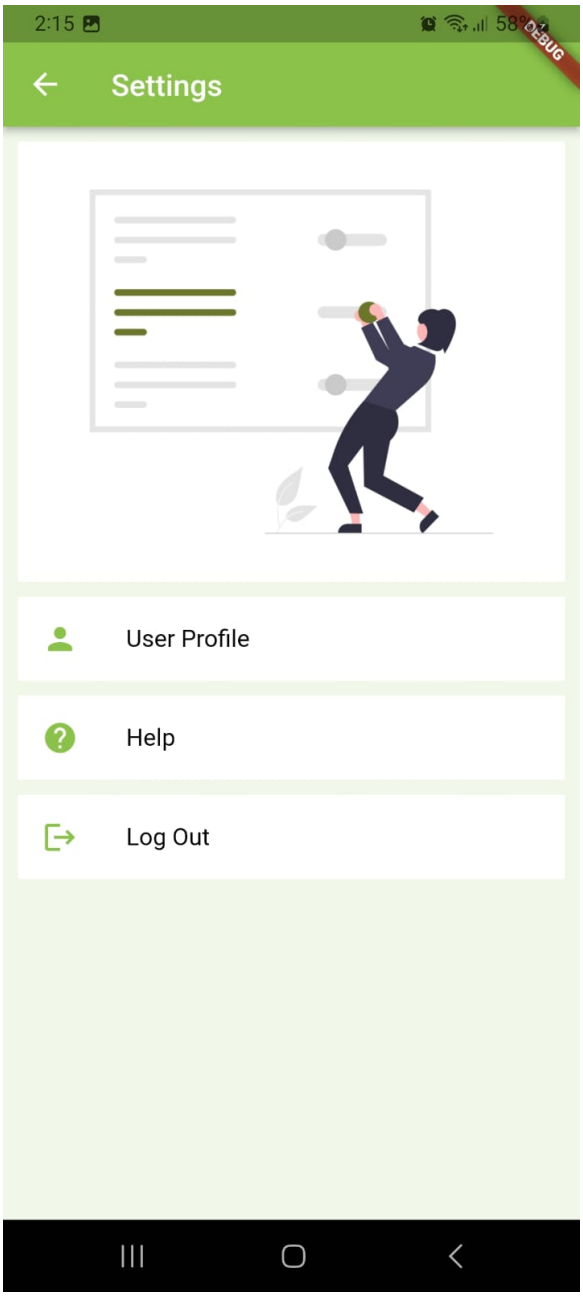

Figure 4.11: Setting

4.6.3 Screen Objects and Actions

In the TrashBot application, several screen objects and associated actions contribute to the overall user experience. Here are some key screen objects and their corresponding actions:

1. **TrashBot Control Screen:**
    - Objects:
        - Bin Lid Control Button: Allows the user to open or close the lid of the TrashBot bin.
        - View Segregation Process: This enables the user to view the waste being segregated.
    - Actions:
        - Clicking on the Bin Lid Control Button triggers the corresponding action of opening or closing the bin lid.
        - Clicking on the Segregation Control Button initiates the AI-driven waste segregation process.
2. **Waste Collection Scheduling Screen:**
    - Objects:
        - Bin Selection Dropdown: Allows the user to choose the desired bin for waste collection.
        - Date and Time Pickers: Enable the user to specify the collection date and time.
        - Schedule Collection Button: Triggers the action of scheduling the selected bin for waste collection.
    - Actions:
        - Selecting a bin from the Bin Selection Dropdown sets the bin for collection.

- Choosing a date and time using the Date and Time Pickers specifies the collection schedule.
- Clicking the Schedule Collection Button confirms the scheduling action.

3. **Reports Generation Screen:**

   Objects:Report Type Selection (Weekly/Monthly): Allows the user to choose the type of report to generate. Period Selection (Week/Month Picker): Enables the user to specify the reporting period. Generate Report Button: Triggers the action of generating the selected report.

   Actions:

   - Selecting the Report Type (weekly or monthly) determines the type of report to be generated.
   - Choosing a specific period using the Week/Month Picker defines the reporting period.
   - Clicking the Generate Report Button initiates the report generation action.

4. **Environmental Impact Updates Screen:**

   Objects:News Articles Section: Displays relevant news articles related to waste management and environmental impact. Tips and Guidelines Section: Provides tips and guidelines for waste reduction and recycling. Educational Content Section: Presents educational content on the adverse effects of waste on the environment and climate.

   Actions:

   - Users can scroll through the News Articles Section to read the latest news updates.
   - They can browse the Tips and Guidelines Section to access actionable tips for waste reduction and recycling.

– Users can explore the Educational Content Section to gain knowledge about the impact of waste on the environment and climate. These screen objects and associated actions work together to provide users with control over the TrashBot bin, the ability to schedule a waste collection, generate reports, and stay informed about environmental impact. The actions associated with each screen object empower users to interact with the application and accomplish their tasks effectively.

Chapter 5

Software and System Testing

The "Software and System Testing" chapter focuses on the critical phase of testing the TrashBot system. This chapter emphasizes the importance of verifying and validating the software and system components to ensure they meet the specified requirements and perform as intended. Software and system testing play a crucial role in identifying and resolving any defects, errors, or inconsistencies that may arise during the development process. The chapter outlines the various testing techniques and methodologies employed, including unit testing, integration testing, system testing, and acceptance testing. By conducting thorough testing, the design team aims to ensure the reliability, functionality, and performance of the TrashBot system, ultimately enhancing its effectiveness in waste management operations. The results of the software and system testing phase provide valuable insights into the system's quality, highlighting areas for improvement and facilitating the delivery of a robust and reliable waste management solution.

5.1 Overview

The software testing section dedicated to the TrashBot application will provide a comprehensive overview of the testing methodology, test blueprint, and specific test scenarios devised for the software. Its primary objective is to verify that the TrashBot application operates as intended and fulfills the requirements specified in the design

section. In detail, the testing strategy will outline the general approach adopted for evaluating the TrashBot application. Additionally, it will define the roles and responsibilities of the testing team and elaborate on how the testing process will be supervised and organized.

5.1.1 Scope

The software testing section dedicated to the TrashBot application encompasses a comprehensive range of testing activities aimed at ensuring compliance with the requirements specified in the design chapter. This entails conducting various types of tests, such as functional testing, regression testing, performance testing, user acceptance testing, integration testing, security testing, as well as maintenance and support testing. The objective is to thoroughly assess the software application across multiple dimensions and ensure its adherence to the defined criteria.

5.1.2 Purpose

The software testing section dedicated to the TrashBot application serves the purpose of verifying that the application aligns with the requirements specified in the design chapter. It acts as a guide for conducting comprehensive testing of all software aspects prior to release. The chapter presents a detailed plan for testing the TrashBot application, encompassing the testing approach, methodologies, and tools employed to ensure the system functions as intended. It also delineates the roles and responsibilities of the testing team, establishes the testing schedule, and provides an elaborate description of the test plan, test cases, and expected outcomes for each testing category. By leveraging the software testing chapter, any defects or issues within the TrashBot application can be identified and addressed before its release. Moreover, it serves as a valuable record of testing activities and results, facilitating future reference and contributing to the enhancement of the software development process. In essence, the software testing chapter for the TrashBot application endeavors to achieve thorough and meticulous testing, ultimately ensuring that the software application

adheres to the required quality standards for its designated purpose.

5.1.3 Test objectives

The TrashBot application has specific testing objectives to guarantee its quality and performance. These objectives are as follows:

Functional Testing: This objective aims to verify that the application effectively performs all the functions and features outlined in the software design chapter.

Usability Testing: The usability testing objective focuses on assessing the user-friendliness of the application, ensuring that it can be easily used by all users.

Performance Testing: This objective involves testing the application's ability to handle a substantial volume of users and transactions without encountering any performance issues.

Security Testing: Security testing ensures that the application is well-protected against unauthorized access and data breaches, guaranteeing the safety of user information.

Compatibility Testing: Compatibility testing verifies that the application functions seamlessly across different devices, browsers, and operating systems.

Integration Testing: This objective ensures the smooth integration of the application with other systems and applications within the TrashBot ecosystem.

User Acceptance Testing: User acceptance testing validates that the application meets the requirements and expectations of end-users and stakeholders.

Regression Testing: Regression testing aims to ensure that any modifications or updates to the application do not adversely affect existing functionality and features.

Accessibility Testing: Accessibility testing ensures that the application is accessible to users with disabilities and complies with relevant accessibility guidelines.

Disaster Recovery Testing: This objective tests the application's ability to recover from disasters or data loss, ensuring timely restoration of normal operations.

These testing objectives collectively ensure that the TrashBot application undergoes thorough testing, meeting the required quality standards before its release.

5.1.4 Intended Audience

The audience of the software testing chapter for the TrashBot application can vary depending on the stakeholders involved in the project. Some potential audiences could include:

Project Managers: They are responsible for overseeing the testing process and ensuring that it is completed according to schedule.

Developers: They need to understand the test cases and expected results to fix any issues found during testing.

Quality Assurance (QA) Engineers: They are responsible for carrying out the testing process and ensuring that the application meets the required quality standards.

End-users: They need to know that the application has been thoroughly tested and is reliable to use.

Business Owners and Stakeholders: They need to ensure that the application meets the required business requirements and objectives.

System Administrators: They need to understand the test cases and expected results to ensure that the application runs smoothly in the production environment.

The testing chapter is written in a clear and concise manner so that it can be easily understood by all of the stakeholders involved in the project.

5.1.5 Disclaimer

The testing chapter is based on the current state of the application and may not reflect any future changes or updates. The results of the testing process may be influenced by a range of factors such as the testing environment, hardware and software configurations, and the skills and expertise of the testing team. The testing chapter is not a guarantee of the application's performance, reliability, or suitability for any particular purpose. The testing chapter should not be relied upon as the sole means of evaluating the application's quality or fitness for purpose. The testing chapter may contain errors or omissions.

5.1.6 Limitations

A testing chapter may not identify all possible defects or issues with the software. Some defects may only become apparent during actual usage, which may not be captured by the testing document.

5.2 Definitions and abbreviations

5.2.1 Definitions

Anomaly: Any observed deviation in the software's documentation or operation from the expected behavior based on previously validated software products, reference documents, or other sources of indicative behavior.

Component testing: The testing of individual hardware or software components.

Feature: A distinctive characteristic of a system item, encompassing both functional and non-functional attributes such as performance and reusability.

Integrity level: The extent to which software complies or must comply with a set of software and/or software-based system characteristics selected by stakeholders to reflect the software's importance.

Test case: A collection of test inputs, execution conditions, and anticipated results created for a specific objective.

Test class: A designated grouping of test cases.

Test plan: A document that outlines the scope, approach, resources, and schedule of intended test activities. It identifies the test items, features to be tested, testing tasks, responsible individuals, and any risks that necessitate contingency planning.

Validation: The process of assessing a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

Verification: The process of evaluating a system or component to determine whether the products generated in a particular development phase meet the conditions established at the start of that phase.

5.2.2 Abbreviations

SDC Software Design Document

SRS System Requirements Specification

5.3 Test Items

The test items cover various functionalities and scenarios to make sure that the application works as intended. Following is the list of possible test items:

1. **User Registration and Login:**
   To verify and test that users can create an account, register successfully, and can access their accounts.

2. **Bin Scheduling:**
   To test the bin scheduling feature to ensure that the user can set the desired date and time for the trash pickup

3. **Bin Notification:**
   To test that the user will get a timely alert when the bin gets full

4. **Segregated Waste reports:**

   To verify and test that the user accurately generates reports on waste segregation

5. **News Updates:**

   To verify and test that the user access the latest updates on the adverse effects of garbage on the environment and climate

6. **Integration with TrashBot Bin:**

   To verify and test the communication and integration between the TrashBot application and the TrashBot bin

7. **User Interface:**

   To verify and test the user interface to ensure it is intuitive, responsive, and user-friendly

8. **Performance and Reliability:**

   To test the application to ensure that it performs well and is stable, reliable, and responsive

9. **Security:**

   To test user authentication, data encryption, and protection against common vulnerabilities

10. **Compatibility:**

    To test the application on different devices, operating systems, and browsers to ensure compatibility and consistent functionality.

5.4 Features to be Tested

The features to be tested for the TrashBot application include:

- **Notification Feature:** Testing the functionality of the application to send notifications to the user when the trash bin reaches its capacity.

- **Scheduling Feature:** Testing the ability of the application to allow the user to schedule the disposal of the trash bin at their preferred date and time.
- **Report Generation Feature:** Testing the functionality of the application to generate reports on waste segregation for analysis purposes.
- **News Feature:** Testing the application's capability to provide the user with updates on the latest adverse effects of waste on the environment and climate through new features.

These features will be thoroughly tested to ensure their proper functioning and adherence to the requirements of the TrashBot application.

5.5 Features not to be Tested

The features that will not be tested for the TrashBot application are:

- **Hardware Components:** As the focus of the testing chapter is on the software aspect of the application, the testing of hardware components, such as the Smart AI TrashBot Bin itself, will not be included.

- **Integration with External Systems:** The TrashBot application integration with external systems, such as waste management systems or environmental databases, the testing of these external systems will not be covered in the software testing chapter.

- **User's Device Compatibility:** The application is designed to work on specific platforms or operating systems like Android and iOS, the compatibility testing of various user devices is not included in the software testing chapter.

- **Network Infrastructure:** The testing of the network infrastructure, including the internet connection or mobile network, will not be specifically addressed in the software testing chapter. This aspect is typically handled by network administrators or infrastructure testing teams.

These features fall outside the scope of the software testing chapter as they involve aspects that are not solely related to the software functionality of the TrashBot application.

5.6 Approach

In our endeavor to uphold and ensure the quality of the TrashBot application, we have diligently adhered to a comprehensive approach grounded in manual testing. This methodology has been adopted in recognition of its capacity to meticulously analyze each aspect of the application and to provide a binary pass/fail judgment that offers clear, actionable outcomes.

Our approach to the testing procedure for the TrashBot application is a blend of both functional and non-functional testing, a combination formulated to thoroughly investigate the application's performance and compliance with set standards. Our testing process emphasizes an in-depth understanding of both the 'functional' attributes, i.e., how well the application performs its intended tasks, as well as 'non-functional' attributes such as usability, performance, reliability, and security. This comprehensive approach ensures that the application not only does what it is intended to but does so in a user-friendly and reliable manner.

Central to our manual testing methodology is the development and execution of a well-structured test plan, meticulously crafted to include a diverse set of test cases. Each of these test cases has been designed to probe distinct aspects of the TrashBot application, from its core functionalities to its performance under various conditions, to its interactions with users.

In addition, to leverage the power of manual testing fully, our test plan has been

constructed to include exploratory testing. This style of testing emphasizes the personal freedom and responsibility of the individual tester to continually optimize the quality of their work, hence supplementing our structured test cases and providing an additional layer of assurance. This approach is particularly effective in detecting issues related to usability and user experience, which are often overlooked in a purely scripted testing approach.

Moreover, our testing methodology also takes into consideration the need for regression testing. Whenever a new feature is added or an existing one modified, regression tests are conducted to ensure that the changes have not unintentionally disturbed or degraded the performance of the application. This assists in ensuring the stability and reliability of the TrashBot application throughout its development cycle.

Lastly, we acknowledge the importance of documenting our test procedures and results. Detailed test reports, comprising observed outputs, identified defects, and recommendations for corrective actions, are created and reviewed after each test cycle. These reports facilitate tracking of identified issues until their resolution and serve as a point of reference for future testing activities, enhancing overall transparency and traceability.

In summary, our robust and comprehensive manual testing approach enables us to thoroughly scrutinize every facet of the TrashBot application, from its performance of designated tasks to its usability and reliability. This approach, coupled with our meticulously crafted test plan, rigorous execution of test cases, exploratory and regression testing, and diligent documentation practices, ensures that the TrashBot application not only meets but surpasses the required standards.

5.7 Item Pass/Fail Criteria

We'll develop test plan to test the functionality of TrashBot application through test cases. In the test plan, we'll test different parts of the application and specify them as pass or fail on the basis of their function. The test plan is shown in figure 5.1 :

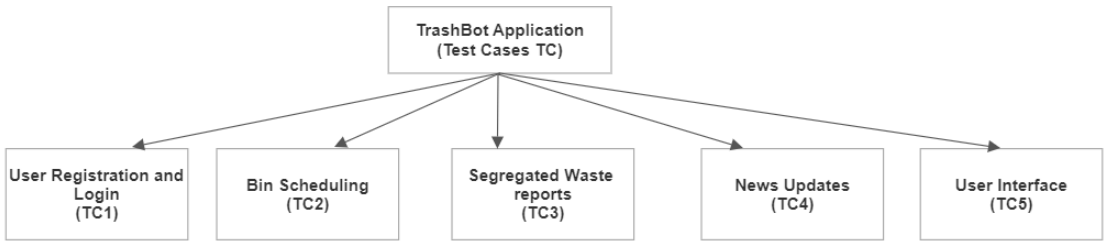

Figure 5.1: Test Plan of TrashBot application representing the features that need to be tested with the help of test cases

5.7.1 User Registration and Login:

User Registration and Login testing feature verifies the functionality and security of account creation and login processes. It ensures successful registration, valid inputs, and secure access for users. Test cases cover inputs, error handling, password encryption, session management, and authentication.

Test Case ID	TC001
Test Case Description	New users provide personal details for registration, while returning users log in with email and password.
Test Steps	1. New users register by providing personal details. 2. Returning users login with email and password. 3. System verifies registration and saves data in the database. 4. Users login with provided credentials.
Test Data	User personal details like email, phone number, address, date of birth, name etc.
Expected Results	1. Data is valid. 2. Registered successfully. 3. Logged in successfully.
Actual Results	1. Data is valid. 2. Registered successfully. 3. Logged in successfully.
Pass/Fail	Pass

Table 5.1: User Registration and Login (TC001)

5.7.2 Bin Scheduling:

Bin scheduling is a testing feature that focuses on verifying the functionality and accuracy of the scheduling system for waste bins. The objective is to ensure that bins are scheduled according to calendar. Test cases are designed and executed to validate the proper creation, assignment, and updating of bin schedules. This includes scenarios such as adding new bins to the schedule by chosing date and time on the calendar.

Test Case ID	TC002
Test Case Description	User will login and can schedule bin by choosing date and time on calendar to collect the waste and segregate it.
Test Steps	1. Enter email and password to login. 2. Select date from calendar and time from clock to schedule the bin for waste collection. 3. Check whether the user is able to chose the date and time to schedule the bin. 4. Check whether the system is saving the schedule or not.
Test Data	Select date and time for bin scheduling and save it.
Expected Results	Notification will be generated about bin schedule.
Actual Results	Bin scheduled successfully.
Pass/Fail	Pass

Table 5.2: Bin Scheduling (TC002)

5.7.3 Segregated Waste Reports

Segregated Waste Reports is a testing feature that focuses on verifying the accurate generation and presentation of reports based on the segregation of waste. The objective is to ensure that the system correctly captures and analyzes data regarding the different types of waste that have been segregated and provides comprehensive reports. Test cases are designed and executed to validate the proper generation of reports, including aspects such as data accuracy, report formatting etc. This feature aims to provide valuable insights and statistics on the effectiveness of waste segregation efforts for monitoring and decision-making purposes.

Test Case ID	TC003
Test Case Description	Are the system is generating segregating reports or not on daily/monthly basis that how much waste is segregated and what kind of waste it is such as plastic, metal,battery etc.
Test Steps	1. Enter email and password to login. 2. Select Reports to see the report. 3. Check whether the user is able to see the correct reports or not. 4. Check whether the system is saving the reports according to waste segregation. 5. Check whether the system is generating reports dynamically when the waste is segregated again and again.
Test Data	Segregate the waste and see the reports in Reports section.
Expected Results	Waste Segregation percentage along with waste type.
Actual Results	Waste Segregation percentage along with waste type.
Pass/Fail	Pass

Table 5.3: Segregated Waste Reports (TC003)

5.7.4 News Updates:

News Updates is a testing feature that focuses on validating the functionality of delivering timely and accurate news updates related to waste management. The objective is to ensure that the system retrieves and displays the latest news articles or notifications pertaining to waste management practices, regulations, or industry updates. Test cases will be designed to verify the proper retrieval, display, and formatting of news updates, as well as the ability to filter and sort the updates based on relevance or date. This feature aims to provide users with up-to-date information and promote awareness and engagement in waste management activities.

Test Case ID	TC004
Test Case Description	Are the system is displaying dynamic news or not.
Test Steps	1. Enter email and password to login. 2. Select News to see the news. 3. Check whether the user is able to see the dynamic news or not. 4. Check whether the system is showing the dynamic news about the waste effect on environment. 5. Refresh it.
Test Data	Select the news section. Refresh it again and again to see whether the news is changing or not.
Expected Results	System is displaying the dynamic news.
Actual Results	System is displaying the dynamic news.
Pass/Fail	Pass

Table 5.4: News updates (TC004)

5.7.5 User Interface:

User Interface testing feature involves validating the visual appearance, functionality, and usability of the system's interface. It focuses on testing elements such as navigation, layout, responsiveness, and user interaction. Test cases are designed to ensure a seamless and intuitive user experience, identifying and resolving any issues related to the interface design and usability.

Test Case ID	TC005
Test Case Description	User Interface. Is the system is user-friendly such as clear icons, buttons, font size, page navigation?
Test Steps	1. Enter email and password to login. 2. Check every icon, navigation of page, display size, quality of content. 3. Check whether the user is able to easily use the system or not.
Test Data	System's screen
Expected Results	User friendly system's components like navigation, font, buttons, icons etc.
Actual Results	User friendly system's components like navigation, font, buttons, icons etc.
Pass/Fail	Pass

Table 5.5: User Interface (TC005)

5.8 Suspension Criteria and Resumption Requirements

Suspension Criteria and Resumption Requirements are an essential aspect of the TrashBot system, which ensures the proper handling and management of suspensions in its operations. This feature defines the conditions and criteria that determine when the system should be temporarily suspended and the necessary steps for resuming its functionality. By establishing clear suspension criteria, such as system errors or maintenance requirements, the system can pause its operations to prevent any further issues or potential risks. The resumption requirements outline the specific actions and procedures that need to be followed to safely and efficiently restore the system's normal functioning. Proper testing and validation of these criteria and requirements are crucial to ensure the reliability and stability of the TrashBot system.

5.8.1 Suspension Criteria

Suspension criteria are conditions or situations that would lead to the temporary halt of testing activities. In the context of the TrashBot application, suspension criteria can be identified based on various factors that may impact the testing process.

- **Critical Defects:** If critical defects are identified during testing that significantly impact the functionality or usability of the TrashBot application, testing activities shall be suspended until these defects are resolved. Critical defects may include errors that prevent proper waste segregation, inaccurate report generation, or failure to schedule waste collection.

- **Unstable Test Environment:** If the test environment for TrashBot becomes unstable, with server unavailability, network disruptions, or other technical issues that hinder proper testing, testing shall be temporarily halted until the environment stabilizes. An unstable test environment may affect the reliability and accuracy of test results.

- **Requirement Changes:** In the event of substantial changes to the application's requirements or functionality, testing activities shall be suspended until the test plan and test cases are updated to align with the new requirements. This ensures that testing accurately reflects the intended functionality and avoids testing against outdated specifications.

- **Resource Unavailability:** If essential testing resources, such as required test data, test environments, or necessary hardware/software, are unavailable or inaccessible, testing shall be paused until these resources become available. Adequate resources are vital to conducting comprehensive and meaningful tests.

- **External Dependencies:** If TrashBot relies on external systems, services, or APIs that are unavailable or undergo significant changes, testing shall be suspended until the dependencies are resolved or stabilized. This ensures that testing can accurately evaluate the integration and interaction of TrashBot with external components.

5.8.2 Resumption Requirements

Resumption requirements outline the conditions that must be met to resume testing activities after a temporary suspension. These requirements ensure that testing can proceed effectively and efficiently. Here's the resumption requirements for the TrashBot application:

- **Defect Resolution:** Testing activities shall be resumed once critical defects identified during the suspension period have been addressed, fixed, and validated. The development team must resolve these defects to ensure the proper functionality and reliability of the TrashBot application.

- **Stable Test Environment:** Testing can be resumed when the test environment for TrashBot is stable. This includes ensuring that all required servers, databases, network connectivity, and supporting infrastructure are operational and functioning as expected. A stable test environment is essential for conducting accurate and reliable tests.

- **Updated Test Plan:** If there have been significant changes to the application's requirements, functionality, or testing scope, testing shall resume after the test plan and associated test cases have been updated accordingly. This ensures that testing aligns with the latest specifications and accurately covers

the intended functionality of the TrashBot application.

- **Availability of Resources:** Testing can recommence once all necessary testing resources, such as test data, test environments, and any required hardware or software components, are available and accessible to the testing team. Adequate resources are crucial for conducting thorough and comprehensive tests.

- **Regression Testing:** Before resuming the testing process, regression testing should be performed to ensure that previously tested functionalities remain unaffected by changes or fixes implemented during the suspension period. This helps identify any unintended side effects and ensures the overall stability of the application.

- **Communication and Coordination:** Prior to resuming testing, effective communication and coordination among the testing team, development team, and relevant stakeholders should take place. This ensures that everyone involved is aware of the resumption of testing activities and any necessary adjustments to timelines or priorities.

5.9 Test Deliverables

Test deliverables are the documents and artifacts produced during the testing process, which provide valuable information about the testing activities and results. These deliverables serve as evidence of the testing effort and help stakeholders assess the quality and reliability of the TrashBot application. Here are the test deliverables for the TrashBot application:

- **Test Plan:** A comprehensive document that outlines the overall approach, objectives, scope, and schedule of the testing activities for the TrashBot application. It includes details about the testing strategy, test environment, test techniques, and the identified test suites.

- **Test Cases:** Detailed test cases that cover various scenarios and functionalities of the TrashBot application. These test cases specify the input data, expected results, and steps to be executed during testing. They provide a systematic approach to validate the application's behavior.

- **Test Scripts:** Automated test scripts developed using testing frameworks or tools to execute predefined test cases and scenarios. These scripts help automate repetitive tasks, increase testing efficiency, and enable regression testing.

- **Test Data:** Sample or representative data sets used for testing the TrashBot application. This includes waste types, quantities, scheduling data, and various scenarios to assess the functionality and performance of the application accurately.

- **Test Reports:** Detailed reports summarizing the testing activities, including test execution results, defect tracking, and overall test coverage. These reports provide insights into the testing progress, identified defects, their severity, and recommendations for improvement.

- **Defect Reports:** Documentation of identified defects during testing, including their description, steps to reproduce, severity, and priority. These reports facilitate effective communication between the testing team and development

team to address and resolve the reported defects.

- **Test Completion Report:** A report that signifies the completion of the testing process for the TrashBot application. It summarizes the overall testing effort, including the achieved test coverage, number of executed test cases, identified defects, and recommendations for future testing cycles.

- **Traceability Matrix:** A document that establishes traceability between requirements, test cases, and test results. It ensures that all requirements are adequately covered by corresponding test cases and allows for easy tracking and verification of testing completeness.

- **User Acceptance Test (UAT) Plan and Results:** If applicable, a separate UAT plan outlining the activities and criteria for end-user testing. The UAT results report documents the outcomes of user acceptance testing, including any issues or feedback provided by the end-users.

These test deliverables provide valuable insights into the testing process, results, and overall quality of the TrashBot application. They assist stakeholders in understanding the testing effort, assessing the application's reliability, and making informed decisions about its readiness for deployment.

5.10 Testing Tasks

Testing tasks for the TrashBot application involve specific activities that need to be carried out during the testing process to ensure the quality and reliability of the application. These tasks cover different aspects of testing, from test planning to test

execution and reporting. The testing tasks for TrashBot application are shown in figure 5.2:

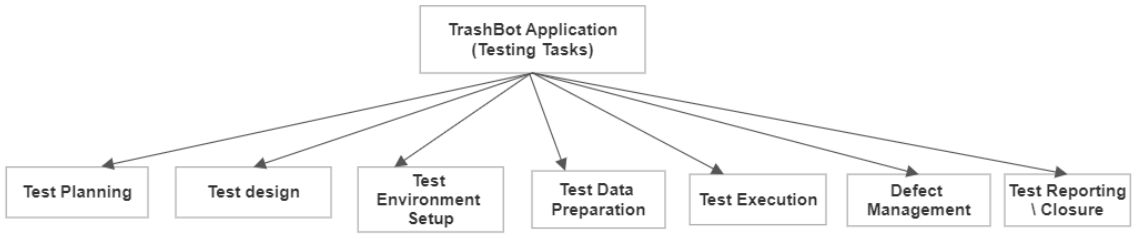

Figure 5.2: Testing tasks of TrashBot representing that which steps should be taken during testing phase of the application

Test Planning: Review project requirements and specifications. Identify the scope and objectives of testing. Define the testing approach and strategy. Create a detailed test plan outlining the testing activities, schedule, and resource requirements.
Test Design: Test design includes the following:

1. Identify testable features and functionalities of the TrashBot application.
2. Develop test scenarios and test cases based on requirements.
3. Determine test data requirements for different test scenarios.
4. Prioritize test cases based on risk analysis and business impact.

Test Environment Setup: Test environment setup includes the following:

1. Set up the necessary hardware, software, and network infrastructure for testing.
2. Prepare the test environment to simulate real-world usage conditions.

3. Install and configure the TrashBot application and any supporting systems or dependencies.

Test Data Preparation: Test data preparation includes the following:

1. Define and gather test data sets representing various waste types, quantities, and scheduling scenarios.
2. Create test data that covers different scenarios, including boundary cases and edge conditions.
3. Ensure the availability and accuracy of test data for comprehensive testing coverage.

Test Execution: Test execution includes the following:

1. Execute test cases according to the defined test scenarios.
2. Record test results, including observed behavior, actual outcomes, and any deviations from expected results.
3. Document defects and issues encountered during test execution.
4. Perform regression testing to ensure that existing functionalities remain unaffected by new changes.

Defect Management: Defect management includes the following:

1. Report identified defects in a structured manner, including detailed descriptions, steps to reproduce, and severity/priority ratings.
2. Collaborate with the development team to investigate and resolve reported defects.

3. Verify defect fixes and retest the impacted functionalities.

Test Reporting and Closure: Test reporting and closure includes the following:

1. Prepare and distribute test progress reports, providing an overview of testing activities, status, and coverage.

2. Document and maintain traceability matrices, linking test cases to requirements.

3. Generate comprehensive test summary reports, including test results, coverage, and recommendations for improvements.

4. Evaluate the completion criteria to determine if the testing objectives have been met.

5. Conduct a test closure meeting to discuss lessons learned and identify areas for process improvement.

6. Archive test artifacts, including test plans, test cases, and reports for future reference and audit purposes.

5.11 Environmental Needs

The environmental needs for the TrashBot application outlines the hardware, software, and infrastructure requirements necessary to conduct the testing activities effectively. This part ensures that the testing environment is adequately set up to simulate real-world conditions and verify the functionality and performance of the application. Here are the environment needs for the TrashBot application:

Hardware Requirements: The requirements for hardware are:

1. A computer system with minimum specifications (e.g., processor, RAM, storage) necessary to run the TrashBot application and associated testing tools.

2. Mobile devices (smartphones, tablets) with various screen sizes and operating systems, if applicable, to test mobile responsiveness.

3. Access to physical waste bins or mockups to simulate waste segregation and collection processes, if required for testing specific functionalities.

Software Requirements: The software requirements are:

1. Operating systems compatible with the TrashBot application (e.g., Windows) to ensure cross-platform compatibility.

2. Web browsers (e.g., Chrome) for testing the TrashBot application's web-based interfaces.

3. Test automation tools (e.g., Selenium) for automated test script execution, if applicable.

4. Database management system (e.g., MySQL, PostgreSQL) for testing data storage and retrieval functionalities.

5. Communication tools (e.g., email clients, messaging platforms) to facilitate collaboration and communication among testing team members.

6. Test management software or spreadsheets to track and manage test cases, test execution results, and defects.

Test Environment Setup: Test environement setup includes the following:

1. Dedicated test servers or cloud-based environments to host the TrashBot application and associated components (e.g., web servers, databases).

2. Internet connectivity to test the TrashBot application's integration with external services or APIs.

3. Virtualization or containerization technologies (e.g., Docker) to create isolated and reproducible testing environments.

4. Test data generation tools or scripts to populate the application with representative test data sets for different waste types, quantities, and scheduling scenarios.

5. Simulated or controlled environments to replicate various waste segregation and collection scenarios, if applicable.

Network Infrastructure: Network infrastructure include the following requirements:

1. Local area network (LAN) or internet connectivity to access the TrashBot application from different devices and locations.

2. Sufficient bandwidth and network stability to ensure uninterrupted testing and reliable communication between testing team members.

3. Firewalls and security configurations to simulate secure network environments and validate the TrashBot application's security features.

5.12 Responsibilities

The responsibilities for the TrashBot application outlines the roles and responsibilities of the individuals or teams involved in the testing process. Clearly defining these responsibilities ensures that all stakeholders understand their roles and contributions, leading to efficient and effective testing. Responsibilities in Testing for TrashBot Application are:

Testing Team: The responsibilities of the testing team are:

1. Develop the test plan, test scenarios, and test cases based on the application requirements.

2. Execute the test cases, record test results, and document any defects or issues encountered during testing.

3. Perform regression testing to ensure the stability of existing functionalities after changes or fixes.

4. Collaborate with the development team to investigate and resolve reported defects.

5. Generate test reports summarizing the testing activities, including progress, coverage, and results.

6. Conduct root cause analysis and provide recommendations for improvement based on test findings.

7. Participate in test closure activities, including lessons learned and knowledge sharing sessions.

Development Team: The responsibilities of the development team are:

1. Provide the testing team with a clear understanding of the TrashBot application's technical architecture and design.

2. Collaborate with the testing team to review and validate the test plan, test scenarios, and test cases.

3. Address and resolve reported defects, ensuring the proper functionality and reliability of the application.

4. Conduct code reviews and unit testing to identify and fix defects at the development level.

5. Assist in providing necessary tools, resources, and access to facilitate testing activities.

6. Collaborate with the testing team to investigate and replicate reported defects and provide necessary information for resolution.

7. Participate in test closure activities, including discussions on lessons learned and process improvements.

Project/Product Manager: The responsibilities of the project or product manager are:

1. Define the overall testing strategy, objectives, and priorities in alignment with project goals.

2. Coordinate and prioritize testing activities based on project timelines and milestones.

3. Ensure clear communication and collaboration between the testing and development teams.

4. Review and approve the test plan, test scenarios, and test cases.

5. Monitor the progress of testing activities and provide necessary support and resources.

6. Review and validate the test reports, ensuring their accuracy and completeness.

7. Make informed decisions regarding the readiness of the TrashBot application for deployment based on testing outcomes.

Domain Experts: The responsibilities of the domain experts are:

1. Provide domain-specific knowledge and insights to guide the testing approach and test case design.

2. Participate in the review and validation of test scenarios and test cases.

3. Assist in identifying critical and high-risk areas for testing based on their understanding of the waste management domain.

4. Validate the TrashBot application's behavior and correctness based on domain requirements.

5. Provide feedback and guidance on the overall quality and suitability of the application from a business perspective.

5.13 Staffing and Training Needs

The staffing and training needs for the TrashBot application outlines the required personnel and their skill sets, as well as any training or knowledge transfer requirements to ensure a competent and efficient testing team. Identifying these needs helps in resource planning and ensuring that the testing team is adequately equipped to perform their tasks effectively. Here are the staffing and training needs for TrashBot application:

Testing Team Roles:

Testing team members and their roles are:

- **Test Lead/Manager:** Responsible for overall test planning, coordination, and supervision of the testing activities. Requires strong leadership and communication skills, as well as expertise in testing methodologies and techniques.

- **Test Analyst/Engineer:** Involved in test design, execution, and reporting. Requires a solid understanding of software testing principles, test case creation, and defect management.

- **Automation Specialist:** If automation is a part of the testing strategy, an automation specialist with experience in relevant testing tools and scripting

languages is required.

- **Domain Expert:** A waste management domain expert who understands the intricacies of waste segregation, collection, and reporting. They contribute domain-specific knowledge and insights to the testing process.

Required Skills and Competencies:

Required skills and competencies are:

- Knowledge of software testing principles, methodologies, and best practices.
- Understanding of waste management processes and concepts.
- Proficiency in test case design, execution, and defect tracking.
- Familiarity with relevant testing tools and frameworks.
- Analytical and problem-solving abilities to identify and report defects accurately.
- Strong communication and collaboration skills to effectively interact with the development team and other stakeholders.
- Attention to detail and ability to work under pressure to meet deadlines.
- Adaptability and willingness to learn new technologies, tools, and domain-specific information.

Training and Knowledge Transfer:

- **Orientation to TrashBot Application:** Provide the testing team with an overview of the TrashBot application, its features, functionality, and the underlying waste management concepts. This ensures a shared understanding of the application and its purpose.

- **Testing Techniques and Tools:** Conduct training sessions to enhance the testing team's knowledge and skills in various testing techniques, such as test case design, boundary value analysis, and regression testing. Provide training on relevant testing tools and frameworks used in the testing process.

- **Waste Management Domain Knowledge:** Organize knowledge transfer sessions or invite waste management domain experts to provide insights into waste segregation, collection, and reporting. This helps the testing team understand the domain-specific requirements and align their testing activities accordingly.

- **Continuous Learning:** Encourage the testing team to stay updated with the latest trends, technologies, and advancements in the software testing field through regular training sessions, workshops, and access to relevant resources.

By identifying the staffing needs and providing necessary training and knowledge transfer, the testing team for the TrashBot application can acquire the required skills and domain knowledge to perform their testing tasks effectively. This ensures that the testing effort is carried out by competent individuals who are well-equipped to validate the quality and functionality of the application.

5.14 Schedule

The schedule for the TrashBot application outlines the timeline and milestones for the testing activities of TrashBot application. It gives an organized plan for when every testing task will take place, ensuring that testing is conducted in an organized and timely manner.. Testing Schedule for TrashBot Application is given below in figure 5.3:

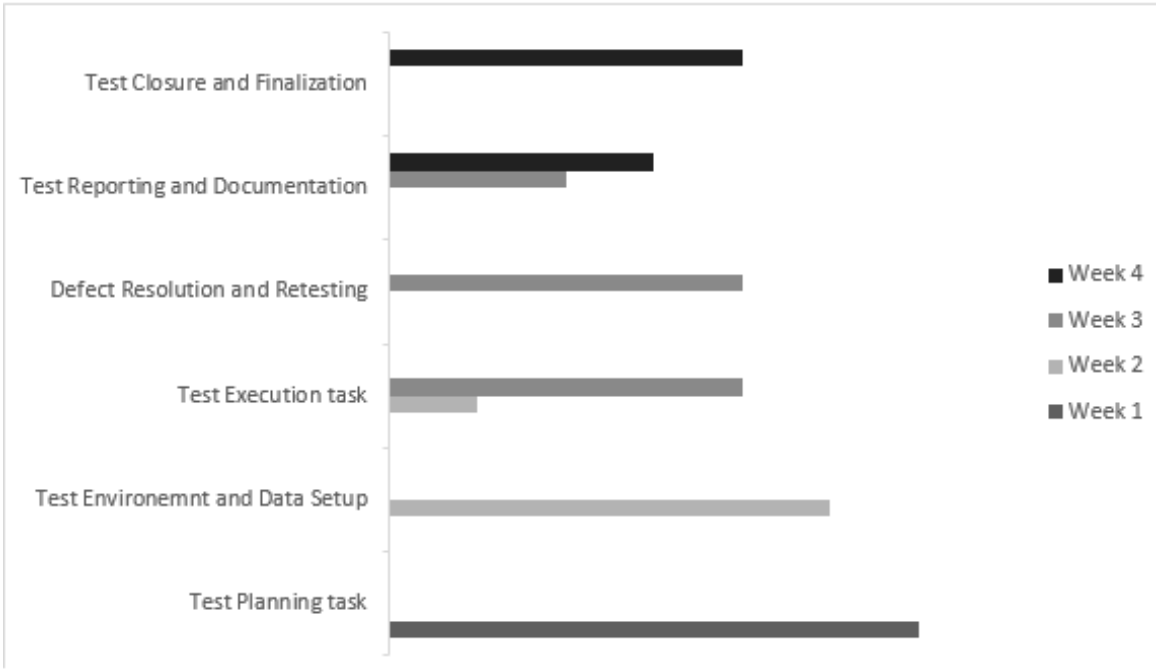

Figure 5.3: Testing Schedule of TrashBot representing the time taken by the tasks included in testing phase of application

5.14.1 Test Planning Task:

Duration: [16 - June - 2023] to [22 - June - 2023]

Activities:

- Gather and analyze project requirements.
- Define testing objectives, scope, and strategy.
- Develop the test plan, including test approach and resource requirements.
- Identify and prioritize test scenarios and test cases.
- Review and finalize the test plan with relevant stakeholders.
- Obtain necessary approvals for the test plan.

5.14.2 Test Environment and Data Setup:

Duration: [23 - June - 2023] to [26 - June - 2023]

Activities:

- Set up the test environment, including hardware, software, and network infrastructure.
- Install and configure the TrashBot application and any supporting systems.
- Prepare representative test data sets covering various waste types, quantities, and scheduling scenarios.
- Validate the test environment and data to ensure readiness for testing.

5.14.3 Test Execution Task:

Duration: [27 - June - 2023] to [2 - July - 2023]

Activities:

- Execute the identified test scenarios and test cases.
- Record test results, including observed behavior and any deviations from expected results.
- Report and track identified defects.
- Conduct regression testing to ensure the stability of existing functionalities.
- Continuously monitor and manage the progress of test execution.

5.14.4 Defect Resolution and Retesting:

Duration: [3- July - 2023] to [7 - July -2023]
Activities:

- Collaborate with the development team to investigate and resolve reported defects.
- Retest the fixed functionalities to verify defect resolution.
- Document and update the status of resolved defects.
- Conduct regression testing after defect fixes to ensure no new issues are introduced.

5.14.5 Test Reporting and Documentation:

Duration: [8 - July - 2023] to [13 - July - 2023]
Activities:

- Prepare and distribute test progress reports, summarizing the testing activities, coverage, and results.
- Document identified defects, including their descriptions, steps to reproduce, and severity/priority ratings.
- Maintain traceability matrices linking test cases to requirements.
- Generate comprehensive test summary reports, including test results, coverage, and recommendations for improvements.

5.14.6 Test Closure and Finalization:

Duration: [14 - July - 2023] to [18 - July - 2023]
Activities:

- Evaluate the completion criteria to determine if the testing objectives have been met.
- Conduct a test closure meeting to discuss lessons learned and identify areas for process improvement.
- Archive test artifacts, including the test plan, test cases, reports, and any relevant documentation.
- Obtain final approvals and sign-offs on the testing deliverables.

5.15 Risks and Contingencies

The risks and contingencies for the TrashBot application addresses potential risks that may impact the testing process and outlines contingency plans to mitigate or respond to those risks. Identifying these risks and having contingency measures in place helps ensure smooth testing operations and minimizes any adverse effects on the project. Here are the risks and contingencies for TrashBot application testing:
Risk: Limited Availability of Test Environment

- **Description:** The test environment may not be consistently available due to server downtime, infrastructure issues, or competing demands from other teams.
- **Contingency:** Maintain a backup test environment or collaborate with the system administration team to prioritize and promptly address any environment-

related issues. Alternatively, leverage cloud-based testing environments for enhanced availability and scalability.

Risk: Inadequate Test Coverage

- **Description:** There is a possibility of insufficient coverage of test cases, leading to potential gaps in testing various features and functionalities of the TrashBot application.

- **Contingency:** Regularly review and update the test plan to ensure comprehensive coverage of critical functionalities. Conduct risk-based analysis to identify high-risk areas and prioritize test case creation and execution accordingly. Leverage automation to increase test coverage and efficiency.

Risk: Evolving Requirements

- **Description:** The TrashBot application's requirements may change or evolve during the testing phase, impacting the relevance and validity of existing test cases.

- **Contingency:** Establish a robust change management process to promptly communicate and document any changes to requirements. Regularly review and update the test plan, test cases, and associated artifacts to align with the revised requirements. Collaborate with stakeholders to ensure that testing efforts remain synchronized with evolving project requirements.

Risk: Data Availability and Quality

- **Description:** Insufficient or inaccurate test data may hinder the effectiveness of testing, leading to incomplete or inaccurate validation of TrashBot's functionality.

- **Contingency:** Identify representative and realistic test data sets covering various waste types, quantities, and scheduling scenarios. Develop data generation or anonymization strategies to ensure data privacy and availability. Regularly validate and update the test data to reflect real-world scenarios.

Risk: Integration and Dependency Issues

- **Description:** TrashBot may rely on external systems, APIs, or databases that are prone to instability, changes, or compatibility issues, impacting the overall testing process.

- **Contingency:** Establish effective communication channels with the external system owners or API providers to address any integration issues promptly. Maintain documentation of integration points, dependencies, and their potential impact on testing. Conduct regular compatibility testing with external systems or APIs to identify and resolve any compatibility issues.

Risk: Resource Constraints

- **Description:** Limited availability of testing resources, such as skilled personnel, time, or budget, may affect the testing process's effectiveness and efficiency.

- **Contingency:** Conduct resource planning and allocation early in the project to ensure adequate staffing and availability of necessary infrastructure. Prioritize testing activities based on risk analysis and critical functionalities. Automate repetitive and time-consuming testing tasks to optimize resource utilization.

Risk: Defect Backlog

- **Description:** A large number of unresolved defects may accumulate, impacting the testing progress and potentially delaying the project schedule.

- **Contingency:** Establish a well-defined defect management process to ensure timely tracking, prioritization, and resolution of reported defects. Regularly review and triage the defect backlog, ensuring that critical defects are addressed promptly. Collaborate closely with the development team to resolve defects efficiently.

5.16 Approvals

The approvals for the TrashBot application outlines the individuals or stakeholders who are responsible for reviewing and approving various testing-related activities, documents, and deliverables. These approvals ensure that the testing process aligns with the expectations and requirements of the project and that key stakeholders have provided their endorsement. Here are the approvals for TrashBot application testing:
Test Plan Approval:

- **Project/Product Manager:** Approves the overall test plan, including the testing objectives, scope, approach, and schedule. Ensures that the test plan

aligns with project goals and requirements.

Test Scenarios and Test Cases Approval:

- **Project/Product Manager:** Reviews and approves the test scenarios and test cases to ensure they adequately cover the application's functionalities and align with the project requirements.

- **Business/Domain Experts:** Provide input and review the test scenarios and test cases from a domain perspective to ensure they accurately represent waste management processes.

Test Data Approval:

- **Project/Product Manager:** Reviews and approves the test data sets used for testing different waste types, quantities, and scheduling scenarios. Ensures that the test data accurately reflects real-world scenarios.

Test Execution Approval:

- **Test Lead/Manager:** Oversees the test execution process and provides approval for initiating the test execution phase once all necessary preparations are completed. This includes verifying that the test environment is ready and that all required resources are available.

Defect Reports Approval:

- **Project/Product Manager:** Reviews and approves the defect reports, ensuring they provide accurate and complete information about identified defects. They may prioritize defects based on their severity and impact on the application.

Test Completion Approval:

- **Project/Product Manager:** Reviews the test completion report, which summarizes the overall testing effort, including coverage, executed test cases, identified defects, and recommendations. Provides final approval for concluding the testing process.

5.17 Results and Discussions

The implementation of deep learning techniques, specifically Convolutional Neural Networks (CNN) and object detection, in the TrashBot system has yielded a remarkable accuracy of 87% in trash sorting. This achievement demonstrates the effectiveness of the system in automating the segregation process and reducing human involvement. The utilization of CNN and object detection algorithms has proven to be instrumental in achieving high accuracy levels, enabling TrashBot to accurately classify and sort different types of trash.

By employing the TrashBot system, we have successfully addressed the waste management problem and contributed to the creation of a clean and green environment. The system's efficiency, time effectiveness, and cost efficiency make it a viable solution for waste segregation. TrashBot's automation capabilities significantly reduce the need for manual intervention, streamlining the trash sorting process and promoting a healthier environment.

Furthermore, TrashBot generates daily and monthly reports that provide users with

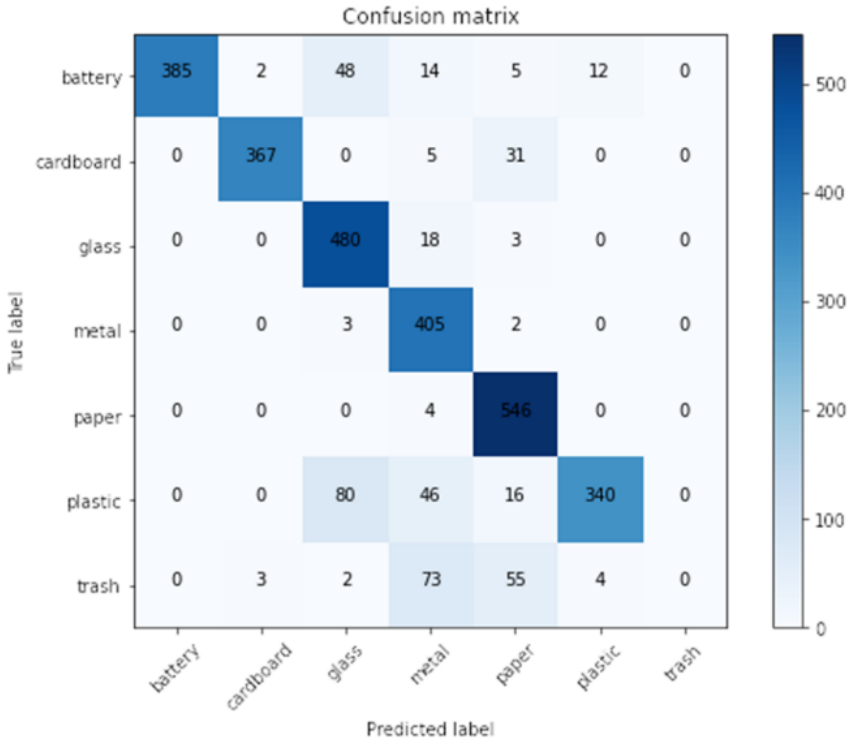

Figure 5.4: Confusion Matrix of TrashBot presenting system's classification performance

up-to-date information on the status and categorization of trash. This feature allows users to stay informed about the environmental situation and the impact of trash on the ecosystem. The reports serve as valuable tools for raising awareness and promoting responsible waste management practices among individuals and communities.

To further evaluate the performance of the TrashBot system, a confusion matrix and accuracy graph can be employed. The confusion matrix as shown in Figure 5.4 provides a detailed breakdown of the system's classification performance, highlighting true positives, true negatives, false positives, and false negatives. This analysis helps in assessing the system's strengths and areas for improvement.

Additionally, an accuracy graph as shown in Figure 5.5 visually represents the system's accuracy over time, allowing for trend analysis and performance monitoring.

In conclusion, the TrashBot system, with its impressive accuracy of 87%, signifies a significant advancement in waste management. Its ability to automate trash segregation, generate informative reports, and contribute to a healthier environment

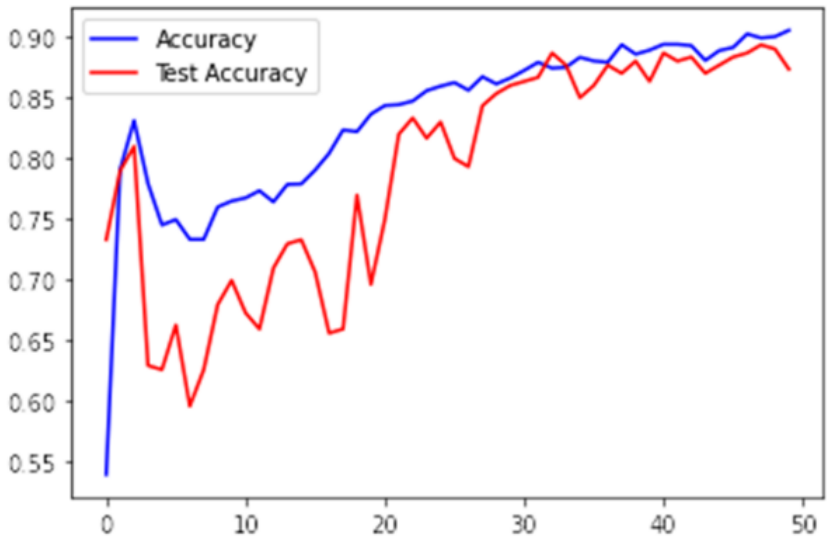

Figure 5.5: Confusion Matrix of TrashBot presenting system's accuracy over time

positions it as a valuable solution for addressing waste-related challenges. Continued research and development in this field will further enhance the capabilities and effectiveness of the TrashBot system, leading to more sustainable waste management practices and a better quality of life for individuals and communities.

5.18 Conclusion

TrashBot application is able to be an effective solution to the waste management problem in order to get clean and green environment. The implementation of the proposed model takes into account all possible economic costs and major environment concerns; and the constraints arise from technical and environmental issues. The system is designed with an objective of developing most efficient ways for waste management that will include less human involvement, less risky, time efficiency, cost efficiency, automation of trash segregation and healthy environment. For now, we have trained the model of our system on seven categories of trash which can further be extended for a dataset of more than seven categories. This method can be successfully implemented to the commercial places like hospitals, schools, malls, railway stations, offices and small organizations. It can be served as the helping tool

to make the clean environment.

References

[1] Shanshan Meng and Wei-Ta Chu. A study of garbage classification with convolutional neural networks. In *2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN)*, pages 152–157. IEEE, 2020.

[2] Ali Bakhshi, Nasimul Noman, Zhiyong Chen, Mohsen Zamani, and Stephan Chalup. Fast automatic optimisation of cnn architectures for image classification using genetic algorithm. In *2019 IEEE congress on evolutionary computation (CEC)*, pages 1283–1290. IEEE, 2019.

[3] Patipol Tiyajamorn, Pollakrit Lorprasertkul, Rawin Assabumrungrat, Warin Poomarin, and Ratchatin Chancharoen. Automatic trash classification using convolutional neural network machine learning. In *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 71–76. IEEE, 2019.

[4] Patipol Tiyajamorn, Pollakrit Lorprasertkul, Rawin Assabumrungrat, Warin Poomarin, and Ratchatin Chancharoen. Automatic trash classification using convolutional neural network machine learning. In *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 71–76. IEEE, 2019.

[5] Umut Ozkaya and Levent Seyfi. Fine-tuning models comparisons on garbage classification for recyclability. *arXiv preprint arXiv:1908.04393*, 2019.

[6] Vella Atienza. Review of the waste management system in the philippines: initiatives to promote waste segregation and recycling through good governance. *Chiba, Japan, Institute of Developing Economies, Japan External Trade Organization*, 2011.

[7] Chun lai Guo, Hong Lan, YingChen Ma, HaNan Zhu, and Kun Sun. The research of garbage classification and recognition based on surf and geometric hashing algorithm. In *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, volume 9, pages 1328–1331. IEEE, 2020.

[8] Jianfei Yang, Zhaoyang Zeng, Kai Wang, Han Zou, and Lihua Xie. Garbagenet: a unified learning framework for robust garbage classification. *IEEE Transactions on Artificial Intelligence*, 2(4):372–380, 2021.

[9] Daniel Otero Gomez, Santiago Cartagena Agudelo, Santiago Isaza Cadavid, Mauricio Toro, Juan Camilo Ramirez, et al. A pipeline for solid domestic waste classification using computer vision. Technical report, Center for Open Science, 2021.

[10] Azni Idris, Bulent Inanc, and Mohd Nassir Hassan. Overview of waste disposal and landfills/dumps in asian countries. *Journal of material cycles and waste management*, 6:104–110, 2004.

[11] Cristian Iorga and Victor-Emil Neagoe. A deep cnn approach with transfer learning for image recognition. In *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–6. IEEE, 2019.

[12] Deyuan Zhang, Zhenghong Liu, and Xiangbin Shi. Transfer learning on efficientnet for remote sensing image classification. In *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pages 2255–2258. IEEE, 2020.

[13] Menti Aditya Gowrisha, Mukesh Kumar Dahlanb, and R Subhashinic. Intelligent waste classification system using vision transformers. *Smart Intelligent Computing and Communication Technology*, 38:331, 2021.

[14] Gary White, Christian Cabrera, Andrei Palade, Fan Li, and Siobhan Clarke. Wastenet: Waste classification at the edge for smart bins. *arXiv preprint arXiv:2006.05873*, 2020.

[15] William Mulim, Muhammad Farrel Revikasha, Novita Hanafiah, et al. Waste classification using efficientnet-b0. In *2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI)*, volume 1, pages 253–257. IEEE, 2021.

[16] Patipol Tiyajamorn, Pollakrit Lorprasertkul, Rawin Assabumrungrat, Warin Poomarin, and Ratchatin Chancharoen. Automatic trash classification using convolutional neural network machine learning. In *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 71–76. IEEE, 2019.

[17] Janusz Bobulski and Mariusz Kubanek. Waste classification system using image processing and convolutional neural networks. In *Advances in Computational Intelligence: 15th International Work-Conference on Artificial Neural Networks, IWANN 2019, Gran Canaria, Spain, June 12-14, 2019, Proceedings, Part II 15*, pages 350–361. Springer, 2019.

[18] Sha Meng, Ning Zhang, and Yunwen Ren. X-densenet: deep learning for garbage classification based on visual images. In *Journal of Physics: Conference Series*, volume 1575, page 012139. IOP Publishing, 2020.

[19] K Sujatha, RC Radha, et al. Technologies for segregation and management of solid waste: A review. In *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, pages 1–4. IEEE, 2016.

[20] Chenyu Lu, Lin Zhang, Yongguang Zhong, Wanxia Ren, Mario Tobias, Zhilin Mu, Zhixiao Ma, Yong Geng, and Bing Xue. An overview of e-waste management in china. *Journal of Material Cycles and Waste Management*, 17:1–12, 2015.

[21] Mahbub Hussain, Jordan J Bird, and Diego R Faria. A study on cnn transfer learning for image classification. In *Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK*, pages 191–202. Springer, 2019.

[22] Qiang Zhang, Xujuan Zhang, Xiaojun Mu, Zhihe Wang, Ran Tian, Xiangwen Wang, and Xueyan Liu. Recyclable waste image recognition based on deep learning. *Resources, Conservation and Recycling*, 171:105636, 2021.

[23] Cenk Bircanoğlu, Meltem Atay, Fuat Beşer, Özgün Genç, and Merve Ayyüce Kızrak. Recyclenet: Intelligent waste sorting using deep neural networks. In *2018 Innovations in intelligent systems and applications (INISTA)*, pages 1–7. IEEE, 2018.

[24] Abhishek Masand, Suryansh Chauhan, Mahesh Jangid, Rajesh Kumar, and Satyabrata Roy. Scrapnet: an efficient approach to trash classification. *IEEE access*, 9:130947–130958, 2021.

[25] Merve Erkinay Ozdemir, Zaara Ali, Balakrishnan Subeshan, and Eylem Asmatulu. Applying machine learning approach in recycling. *Journal of Material Cycles and Waste Management*, 23:855–871, 2021.

[26] Haikel Alhichri, Asma S Alswayed, Yakoub Bazi, Nassim Ammour, and Naif A Alajlan. Classification of remote sensing images using efficientnet-b3 cnn model with attention. *IEEE access*, 9:14078–14094, 2021.

[27] Gaurav Mittal, Kaushal B Yagnik, Mohit Garg, and Narayanan C Krishnan. Spotgarbage: smartphone app to detect garbage using deep learning. In *Pro-*

ceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pages 940–945, 2016.

[28] Sajja Tulasi Krishna and Hemantha Kumar Kalluri. Deep learning and transfer learning approaches for image classification. *International Journal of Recent Technology and Engineering (IJRTE)*, 7(5S4):427–432, 2019.

[29] Li Cao and Wei Xiang. Application of convolutional neural network based on transfer learning for garbage classification. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 1032–1036. IEEE, 2020.

[30] Qiang Zhang, Qifan Yang, Xujuan Zhang, Qiang Bao, Jinqi Su, and Xueyan Liu. Waste image classification based on transfer learning and convolutional neural network. *Waste Management*, 135:150–157, 2021.

[31] Sukmawati Nur Endah, Ilman Nabil Shiddiq, et al. Xception architecture transfer learning for garbage classification. In *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–4. IEEE, 2020.

[32] Maad Ebrahim, Mahmoud Al-Ayyoub, and Mohammad A Alsmirat. Will transfer learning enhance imagenet classification accuracy using imagenet-pretrained models? In *2019 10th International Conference on Information and Communication Systems (ICICS)*, pages 211–216. IEEE, 2019.

[33] Bojan Batinić, Sran Vukmirović, Goran Vujić, Nemanja Stanisavljević, Dejan Ubavin, and Goran Vukmirović. Using ann model to determine future waste characteristics in order to achieve specific waste management targets-case study of serbia. 2011.

[34] Jingyi Liu, Pietro Balatti, Kirsty Ellis, Denis Hadjivelichkov, Danail Stoyanov, Arash Ajoudani, and Dimitrios Kanoulas. Garbage collection and sorting with a mobile manipulator using deep learning and whole-body control. In *2020 IEEE-*

RAS 20th International Conference on Humanoid Robots (Humanoids), pages 408–414. IEEE, 2021.

[35] Nikhil Venkat Kumsetty, Amith Bhat Nekkare, Sowmya Kamath, et al. Trashbox: trash detection and classification using quantum transfer learning. In *2022 31st Conference of Open Innovations Association (FRUCT)*, pages 125–130. IEEE, 2022.