"GARBO: An Automated Garbage Classification App"

A Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of

Bachelor of Technology
in
Computer Science & Information Technology

Submitted by:

Raj Borasi (0827CI201151)

Raj Vishwakarma(0827CI201152)

Tanu Patel (0827CI201189)

Vasu Upadhyay(0827CI201198)

Guided by:

Prof. Garima Joshi

Prof. Vandana Kate

CSIT Department

Acropolis Institute of Technology & Research, Indore

Jan- June 2024

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

2020-2024

DECLARATION

We hereby declare that the work, which is being presented in this project entitled "**GARBO: An Automated Garbage Classification App**" in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology in Computer Science and Information Technology**, is authentic record of work carried out by me.

Place : CSIT, Indore

Date:

Signature of Student

Raj Borasi (0827CI201151)
Raj Vishwakarma (0827CI201152)
Tanu Patel (0827CI201189)
Vasu Upadhyay (0827CI201198)

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

2020-2024

RECOMMENDATION

This is to certify that the work embodied in this project entitled **"GARBO: An Automated Garbage Classification App""** submitted by **Raj Borasi (0827CI201151), Raj Vishwakarma (0827CI201152), Tanu Patel (0827CI201189), Vasu Upadhyay (0827CI201198),** is a satisfactory account of the bonafide work done under the supervision of **Prof.Vandana Kate, Prof.Garima Joshi** , is recommended towards partial fulfillment for the award of the Bachelor of Technology in Computer Science & Information Technology degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

Project Guide :-	**Project Coordinator :-**
Prof. Garima Joshi	**Prof. Vandana Kate**

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

2020-2024

CERTIFICATE

The Project entitled **"GARBO: An Automated Garbage Classification App""** submitted by **Raj Borasi (0827CI201151), Raj Vishwakarma (0827CI201152), Tanu Patel (0827CI201189), Vasu Upadhyay (0827CI201198),** has been examined and is hereby approved towards partial fulfillment for the award of **Bachelor of Technology** in **Computer Science & Information Technology,** for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

Internal Examiner : **External Examiner**

Date : **Date:**

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

2020-2024

STUDENT UNDERTAKING

This is to certify that project entitled **"GARBO: An Automated Garbage Classification App"" "** has developed by us under the supervision of **Prof. Garima Joshi**. The whole responsibility of work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.

Signature of Student :-

Raj Borasi (0827CI201151)

Raj Vishwakarma (0827CI201152)

Tanu Patel (0827CI201189)

Vasu Upadhyay (0827CI201198)

ACKNOWLEDGEMENT

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach us on shore safely.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide **Prof. Garima Joshi, Project Guide** and mentor **Prof. Vandana Kate**, **Project Coordinator**, AITR, Indore for their motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Prof.** (**Dr.) Shilpa Bhalerao HOD CSIT**, AITR Indore for her support, suggestion and inspiration for carrying out this project. We would be failing in our duty if do not acknowledge the support and guidance received from **Prof.** (**Dr.) S. C. Sharma, Director**, AITR, Indore whenever needed. We take opportunity to convey my regards to the **Management of Acropolis Institute, Indore** for extending academic and administrative support and providing us all necessary facilities for project to achieve our objectives.

We are grateful to our parent and family members who have always loved and supported us unconditionally.

Signature of Student :-

Raj Borasi (0827CI201151)

Raj Vishwakarma (0827CI201152)

Tanu Patel (0827CI201189)

Vasu Upadhyay (0827CI201198)

TABLE OF CONTENTS

DECLARATION i
RECOMMENDATION ii
CERTIFICATE iii
STUDENT UNDERTAKING iv
ACKNOWLEDGEMENT v
ABSTRACT vi
CONTENTS vii
List of Figures viii
List of Tables ix
List of Abbreviations x
Chapter 1: Introduction xi
1.1 Overview 1
1.2 Existing System 1
1.3 Problem Statement 1
1.4 Proposed System 1
1.5 Need and Scope 2
1.6 Report Organization 2
Chapter 2: Literature Survey 4
2.1 Study 5
2.2 Problem Methodology 5
2.3 Software Engineering Paradigm 6
2.4 Software Development Life Cycle: 7
2.5 Technology Methodology 7
Chapter 3: Analysis 10
3.1 Identification of System Requirements 11
3.2 Functional Requirements 11
3.3 Non-Functional Requirement 12
3.4 Feasibility Study 12
3.4.1 Technical Feasibility 12

3.4.2 Financial Feasibility 12
3.4.3 Operational Feasibility 13
Chapter 4: Project Planning 14
Chapter 5: Design 17
5.1 Introduction to UML 18
5.2 UML Diagrams 18
5.2.1 Use Case Diagram 18
5.2.2 Class Diagram 20
5.2.3 Sequence Diagram 20
5.2.4.ER Diagram 21
5.2.5 Activity Diagram 22
Chapter 6: Implementation 23
6.1 Coding (Main Module) 24
6.2 Results: Screen Shots 25
Chapter 7: Testing 29
7.1 Testing Objectives 30
Chapter 8: Conclusion 33
8.1 Conclusion 34
8.2 Future Work 35
References 37
Web Sites 38
Appendix 39

LIST OF FIGURES

Figure No. and Title	Page No.
Figure 2-1: Workflow Diagram	17
Figure 5-1: Use Case Diagram	27
Figure 5-2: Class Diagram	28
Figure 5-3: Sequence Diagram	29
Figure 5-4: ER Diagram	30
Figure 5-5: Activity Diagram	31
Figure 6-1: Screenshot 1	33
Figure 6-2: Screenshot 2	33
Figure 6-3: Screenshot 3	33
Figure 6-4: Screenshot 4	34
Figure 6-5 : Screenshot 5	34
Figure 6-6: Screenshot 6	35
Figure 6-7: Screenshot 7	35
Figure 6-8: Screenshot 8	36
Figure 6-9: Screenshot 9	36
Figure 4-13: Test Case 2 Output 1	38
Figure 4-14: Test Case 2 Output 2	39

LIST OF ABBREVIATIONS

Abbreviations

Abbr1: CNN- Convolutional Neural Networks

Abbr2: APC-Air Pollution Control

Abbr3: API-Application Programming Interface

Abbr4: ML- Machine Learning

Abbr5: AI-Artificial Intelligence

Abbr6: CSV-Comma-Separated Values

Abstract

Garbo: The Automated Garbage Classification App

This report of EOI is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (MP), India for partial fulfillment of Bachelor of Engineering in Computer Science & Engineering branch under the sagacious guidance and vigilant supervision of **Prof. Vandana Kate**.

Waste management is a pressing concern in modern urban environments, and effective waste classification is a key aspect of this challenge. This project introduces a Garbage Classification App that harnesses the power of image recognition technology to classify waste into two distinct categories: biodegradable and non-biodegradable. Additionally, the app provides users with valuable information about the waste they encounter, including disposal methods and environmental impact. By bridging technology and environmental responsibility, this app promises to revolutionize waste management practices and promote informed decision-making among users.

Keywords: Waste Management, Garbage Classification, Environmental Awareness.

CHAPTER 1

Chapter 1: Introduction

1.1 Overview

Our Garbage Classification App was conceived as a response to these challenges. It harnesses the latest advancements in image recognition technology to transform how waste is managed. By allowing users to capture images of waste items, the app can instantaneously classify them into two primary categories: biodegradable and non-biodegradable. This classification not only facilitates better waste sorting but also promotes the recycling of biodegradable materials and responsible disposal of non-biodegradable items.

1.2 Existing System

The lots of data on paper/register documents are generated in the organization about client complaint and its resolutions by gathering information from various sources needed to categorized and sorted out so that many operations can be performed on it and these data can be processed which takes lots of time due to not having required kind of application system.

1.3 Problem Statement

The aim of this project is to develop a user-friendly and efficient mobile application for garbage classification using image recognition technology. The primary objective is to create an app that can accurately identify and classify various types of garbage waste, distinguishing them into two main categories: biodegradable and non-biodegradable. Additionally, the application should provide users with detailed information about the waste they have captured in the photos, educating and promoting environmentally responsible disposal practices.

1.4 Proposed System

Our Garbage Classification App was conceived as a response to these challenges. It harnesses the latest advancements in image recognition technology to transform how waste is managed. By allowing users to capture images of waste items, the app can instantaneously classify them into two primary categories: biodegradable and non-biodegradable. This classification not only facilitates better waste sorting but also promotes the recycling of biodegradable materials and responsible disposal of non-biodegradable items.

1.5 Need and Scope

1. **Need:**

- Environmental Impact: With increasing waste generation, proper garbage disposal becomes crucial to reduce environmental pollution.
- Efficiency: Manual garbage sorting is time-consuming and prone to errors. An automatic app can streamline this process.

2. **Scope:**

- Image Recognition: The app can use AI algorithms to recognize different types of waste through images captured by the user's device.
- Database Integration: It can integrate with a database containing information on various waste items and their proper disposal methods.

1.6 Report Organization

- Chapter 1 states the overview of the project with discussing about the discussing about the existing systems in today’s scenario. Describing about the problem statement we are facing about the system. We have given our proposed solution considering all the shortcomings of the previously used system.

- Chapter 2 states the literature survey i.e. the background details of our system including the software engineering paradigm and explaining about

the technologies (Software and Hardware requirement) which we have used building the system.

- Chapter 3 states about the Analysis of the whole system i.e. identification of system requirement about the feasibility study-Technical Feasibility, Financial Feasibility, operational Feasibility.

- Chapter 4 states about the Design of the whole system including all the UML diagrams all the tools used with ER Diagram and Data Flow Diagram and Data Dictionary.

- Chapter 5 states about the whole code of the system, java code , typescript ,html ,CSS scripts and its integration and adaptability.

- Chapter 6 states the Testing phase of our system all the different testing methods and strategies and we have run test cases.

- Chapter 7 states the conclusion of the whole system explaining the advancement of our project in future.

- References: The books, websites, journals, blogs which we have referred.

CHAPTER 2

Chapter 2: Literature Survey

2.1 Study

In this section, recent studies on the identification and classification of waste materials in garbage are examined in detail. In their study, Endah and Shiddiq performed garbage classification using the pre-trained VGG16, ResNet-50, and Xception models TrashNet dataset, and an 88% success rate was obtained with the Xception model (Endah and Shiddiq, 2020). Fu et al., in their study, developed an intelligent garbage classification system based on deep learning and Linux system using the Huawei Garbage Classification Challenge Cup dataset. In this developed system, they achieved a success rate of 92.62% in garbage classification.

Chen and Xiong proposed a structure that detects 15 objects based on the Yolov4 model structure, achieving a 64% success rate (Chen and Xiong, 2020). Yang and Li proposed a new garbage classification system in their study. ImageNet, Garbage Classification, and TrashNet datasets were used for the proposed system, and they achieved classification success rates of 64.5%, 82.5%, and 96.10%, respectively (Yang and Li, 2020). Kang et al., proposed a garbage classification system based on the ResNet-34 deep learning algorithm using online and real-life image datasets, and they achieved a 99% classification success rate (Kang et al., 2020). Qin, et al., proposed a structure combining salience detection and image recognition for garbage classification from images with complex backgrounds and obtained better results (Qin, et al., 2022). Meng et al., in their study, proposed a new model (X-DenseNet) by combining Xception and DenseNet deep learning architectures. The proposed model was trained using a dataset consisting of 6 different (glass, paper, cardboard, plastic, metal, and other trashes) garbage and achieved a garbage classification success rate of 94.1%.

2.2 Problem Methodology

Classifying waste using machine learning is a challenging task since determining the recyclability or compostability of waste based on images is

difficult due to the properties of the material being hard to detect from images. Besides, waste can take on various shapes and forms which requires machine learning techniques to handle such variability and the recyclability of waste depends on the local recycling center's capabilities which the app must consider. Taking those considerations into account, the app is designed in such a way that feedback are collected from users and can operate smoothly with or without an internet connection. The waste image is obtained from the gallery or camera and is passed through the waste classification model which is trained to categorize the waste. The classification model is the result of training a specific CNN model on a dataset of labeled images. Several state of the art convolutional neural network methods are tested in this research which included Inception V3 ,MobileNet V2, Inception Resnet V2, Resnet 50, Mobile Net and Xception. The model is then converted into TensorFlow Lite model as they are highly optimized, efficient and versatile making them ideal for running real-time predictions on mobile. The workflow of Garbo app is shown in Figure 1.

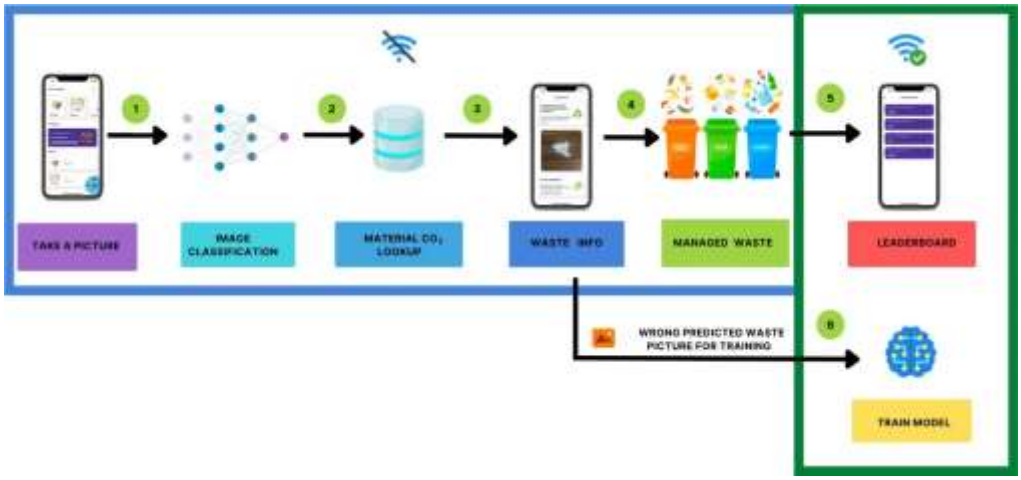


Figure 2-1: Work Flow Diagram

2.3 Software Engineering Paradigm

The principle of software design mainly includes the following:

1. Reliability: The reliability of the software design must be determined. The reliability of the software refers to the ability to avoid fault occurred in the

process of system running, as well as the ability to remedy troubles once the fault occurs.

2. Reusability: Commonness of similar codes and come out new method abstractly and reasonably. Prefer a more generic design.

3. Understand ability: The understand ability of software not only require clear and readable document, but the simplified structure of the software itself, which requires the designer possess keen insight and creativity, and know well about the design objects.

4. Simple program: To keep the program simple and clear, good programmers can use simple program to solve complex problems.

5. Testability: Testability means that the created system has a proper data collection to conduct a comprehensive test of the entire system.

2.4 Software Development Life Cycle:

A software development methodology is an outline or framework to plan and control the process of developing a software application. The methodology for every project is designed to suit the specific needs of each project. The Modified Waterfall Model allows a return to a previous phase for verification or validation, ideally confined to connecting steps. So this project's purpose, we made use of a modified waterfall model for software development methodology.

The systems development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application.

2.5 Technology Methodology

2.5.1 Hardware Requirements

The used hardware for development of application system:

1. RAM: 16 GB

2. Processor: i5 8th generation

2.5.2 Software Requirement

The required software for development environment:

- Windows XP onwards (Used Windows 10 Pro)
- PostgreSQL for database support (Used version 14)
- Java Development Kit (JDK)(Used version 11.0.13.0)
- Eclipse IDE Environment (Used version 2021-09 (4.21.0))
- Spring Tool Suite (STS) plug-in for the Eclipse IDE that is designed to give powerful, integrated environment which build Java Spring Boot Applications.
- Android Studio Code (Used version 1.63.2) for Angular UI development.

2.5.3. Java Programming Language

Java is a high level, robust, object-oriented and secure programming language. It is used to develop mobile apps, web apps, desktop apps, games and much more.

2.5.3.1 Java Advantages:

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)

  It is secure, fast and powerful.

- It is open-source and free.

2.5.4 Spring Boot Framework

Spring Boot is an open-source Java-based framework used to create a micro-Service. Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can just run.

2.5.4.1 Features:

- Create stand-alone Spring applications.
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' dependencies to simplify your build configuration
- Automatically configure Spring and 3rd party libraries whenever possible
- Provide production-ready features such as metrics, health checks, and externalized configuration.
- Absolutely no code generation and no requirement for XML configuration

2.4.5.2 Advantages:

There are few advantages to its developers -

- Easy to understand and develop spring applications.
- Increases productivity.
- Reduces the development time.
- It provides a flexible way to configure Java Beans, XML configurations, and Database Transactions.
- It provides powerful batch processing and manages REST endpoints.
- In Spring Boot, everything is auto configured; no manual configurations are needed.
- It offers annotation-based spring application.
- Eases dependency management.

CHAPTER 3

Chapter 3: Analysis

3.1 Identification of System Requirements

System requirements are the requirements at the system level that describe the functions which the system as a whole should fulfil to satisfy the user's needs and requirements and is expressed in an appropriate combination of textual statements, views and non-functional requirements also ensuring the safety. security, reliability, etc. that will be necessary.

System requirements play major roles such as:

- From the basis of system architecture and design activities.
- From the basis of system integration and verification activities.
- Act as reference for validation and stakeholder acceptance.
- Provide a means of communication between the various technical staff that interact throughout the project.

System requirements are considered in detail during System Definition. Neither can be considered complete until consistency between the two has been achieved as demonstrated by traceability, for which a number of iterations may be needed.

3.2 Functional Requirements

To implement the above goals, the following requirement needs to be followed:

- Specifying the application and various components of the architecture.
- Specifying the bindings between the tasks and the resources either manually or by the design tools.
- Specifying the port interconnections between the resources.
- Analysis: Extracting the data required for analysis and the doing the analysis.

3.3 Non-Functional Requirement

In system and requirement engineering, non-functional requirements are a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirements.

Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of "System shall do< requirement>", an individual action or part of the system, perhaps explicitly in the sense of a function, a box description input, output, process and control functional model.

3.4 Feasibility Study

The Feasibility study aims to uncover the strengths and weakness of a proposed venture, opportunities and threats present in the environment and the resources required to be carried through. The Feasibility study evaluates the project's success.

3.4.1 Technical Feasibility

To design application system which meets the basic requirement, deep understanding of Java language, the Eclipse ide, PostgreSQL databases, the micro services architecture, application of framework and other technical knowledge are needed (framework is the core of the application and pattern that all programmers uses in developing).Based on the related digital technology information, resources and spirit of willing to learn, the digital technology is feasible.

3.4.2 Financial Feasibility

The design of application based on Java framework requires elaborate designs of screens in application, by adopting typescript and Java language as technical support of this application, with plug-in tools, and combination of Java JDK11.0.13 version lead to comprehensive and smoothly design and

development of application. Therefore, the whole process of development doesn't need to spend any money that is economic feasibility.

3.4.3 Operational Feasibility

Strengthened the criteria for primary key specification, primary key be unique across whole database (a so-called enterprise key i.e. a primary key whose value is unique across all relations.), not just unique within the relational table to which it applies. This criterion makes a primary key more like in object-oriented databases called an object identifier. In this project, we have used complaint number as so-called primary key.

CHAPTER 4

Chapter 4: Project Planning

The Automatic Garbage Classification App aims to simplify and streamline the process of waste disposal by automatically identifying and categorizing different types of garbage. This app will utilize image recognition technology to analyze images of waste items captured by users' smartphones and provide real-time feedback on the proper disposal category. By promoting proper waste segregation, the app contributes to environmental sustainability and waste management efforts.

1. **Image Recognition:** The app will employ machine learning algorithms to recognize and classify various types of garbage items based on images captured by users.

2. **Garbage Categorization:** Once the image is analyzed, the app will categorize the garbage into different waste types such as recyclable, non-recyclable, organic, and hazardous.

3. **Real-Time Feedback:** Users will receive immediate feedback on the correct disposal category for each item, helping them make informed decisions about waste management.

4. **Educational Resources:** The app will provide educational materials and tips on waste management practices, including the importance of recycling, composting, and proper disposal of hazardous materials.

5. **User Engagement:** To encourage user participation and engagement, the app may include features such as gamification, challenges, and rewards for proper waste disposal behavior.

6. **Community Integration:** Users can share their experiences, challenges, and success stories related to waste management through social features within the app, fostering a sense of community and collective responsibility.

7. **Data Collection and Analysis:** The app will collect anonymized data on waste disposal habits and trends, which can be used for research purposes and to improve waste management strategies.

8. **Accessibility:** The app will be designed to be user-friendly and accessible to people of all ages and backgrounds, with multilingual support and intuitive interfaces.

CHAPTER 5

Chapter 5: Design

5.1 Introduction to UML

Unified modelling language (UML) is a general-purpose modelling language. The main aim of UML is defining a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behavior and structure of system. UML helps software engineers, businessmen and system architects with modelling, design and analysis.

5.2 UML Diagrams

Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them.

Businessmen do not understand code, so UML becomes essential to communicate with non-programmers essential requirements, functionalities and processes of the system. A lot of time is saved down the line when teams are able to visualize processes user interactions and static structure of the system.

UML is linked with object-oriented design and analysis. UML makes the use of elements and from associations between them to form diagram. Diagram in UML can be broadly classified as:

- **Structural Diagrams:** Capture static aspects or structure of a system. Structural
- **Diagrams include**: Component Diagrams, Objects Diagram, Class Diagrams and Deployment Diagrams.
- **Behaviour Diagram:** Capture dynamic aspects or behaviour of the system. Behaviours
- **Diagram include:** Use Case Diagram , State Diagram, Activity Diagram and Interaction Diagram.

5.2.1 Use Case Diagram

The purpose of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system. Use case diagrams consist of 3 objects.

- Actor: Actor in a use case diagram is any entity that performs a role in one given system. This could be a person, organization or an external system and usually drawn like skeleton.
- Use Case: A use case represents a function or an action within the system. It's drawn as an oval and named with the function.
- System: The system is used to define the scope of the use case and drawn as a rectangle. This an optional element but useful when you're visualizing large systems.

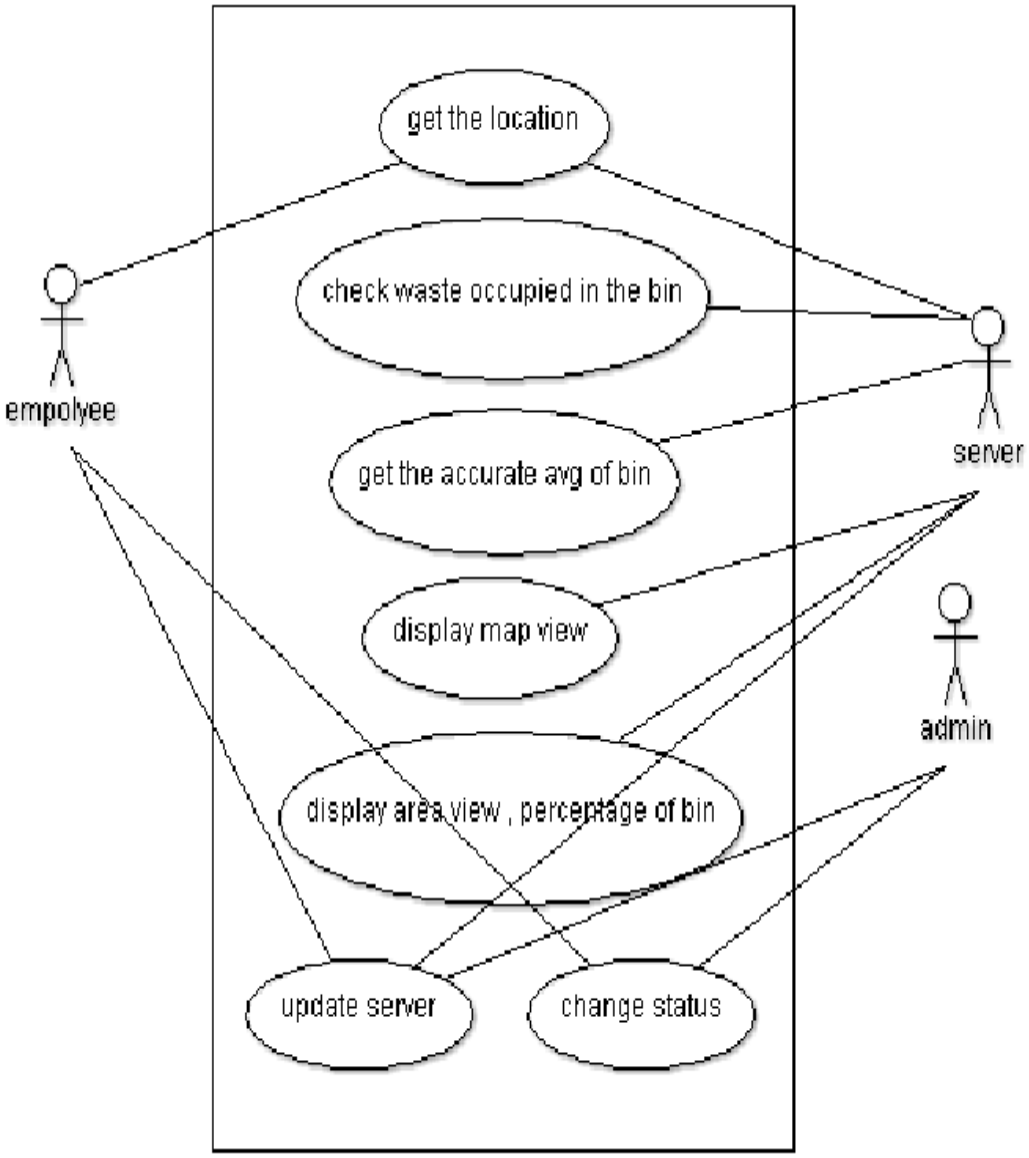


Figure 5-1: Use Case Diagram

5.2.2 Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's:

- classes,
- their attributes,
- operations (or methods),
- And the relationships among objects.

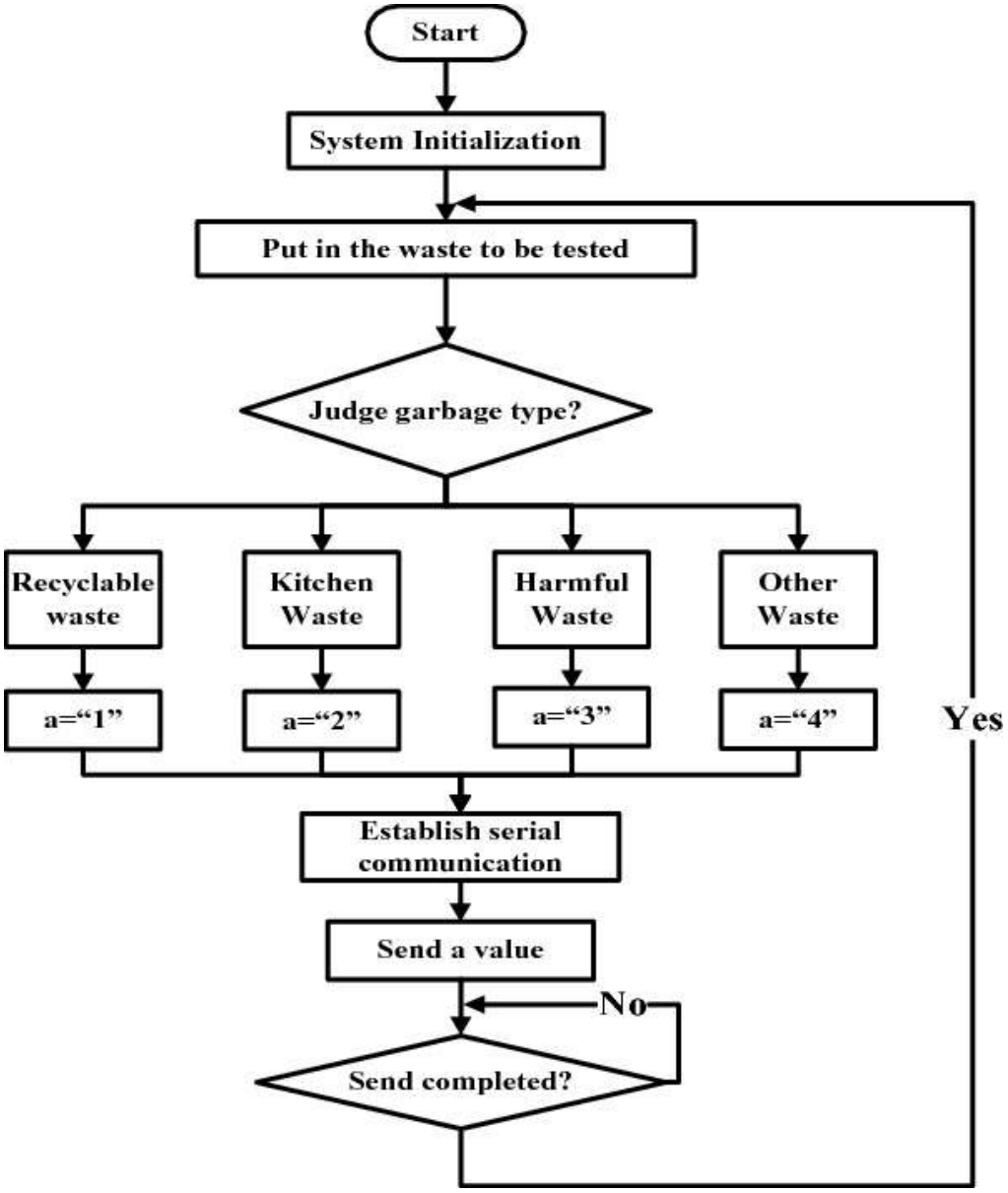


Figure 5-2: Class Diagram

5.2.3 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describe how- and in what order- a group of objects works together.

Sequence diagram are sometimes known as an event diagrams or event scenarios. Sequence diagram are time focus

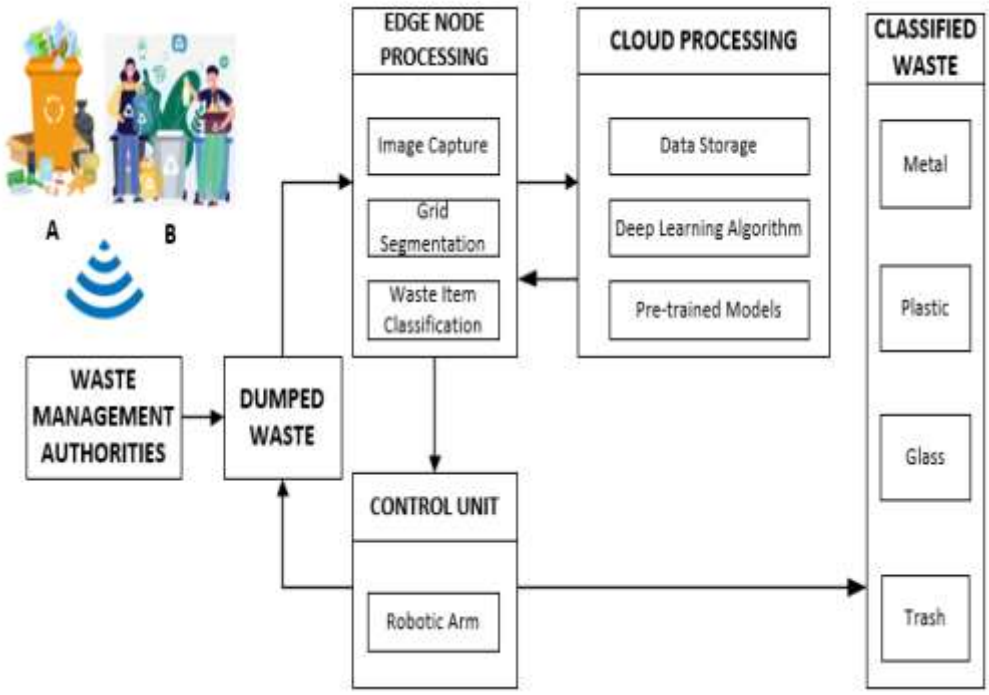


Figure 5-3: Sequence Diagram

5.2.4 ER Diagram

ER Diagram is a visual representation of data that describes how data is related to each other. In ER Model, we disintegrate data into entities, attributes and setup relationships between entities, all this can be represented visually using the ER diagram. ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

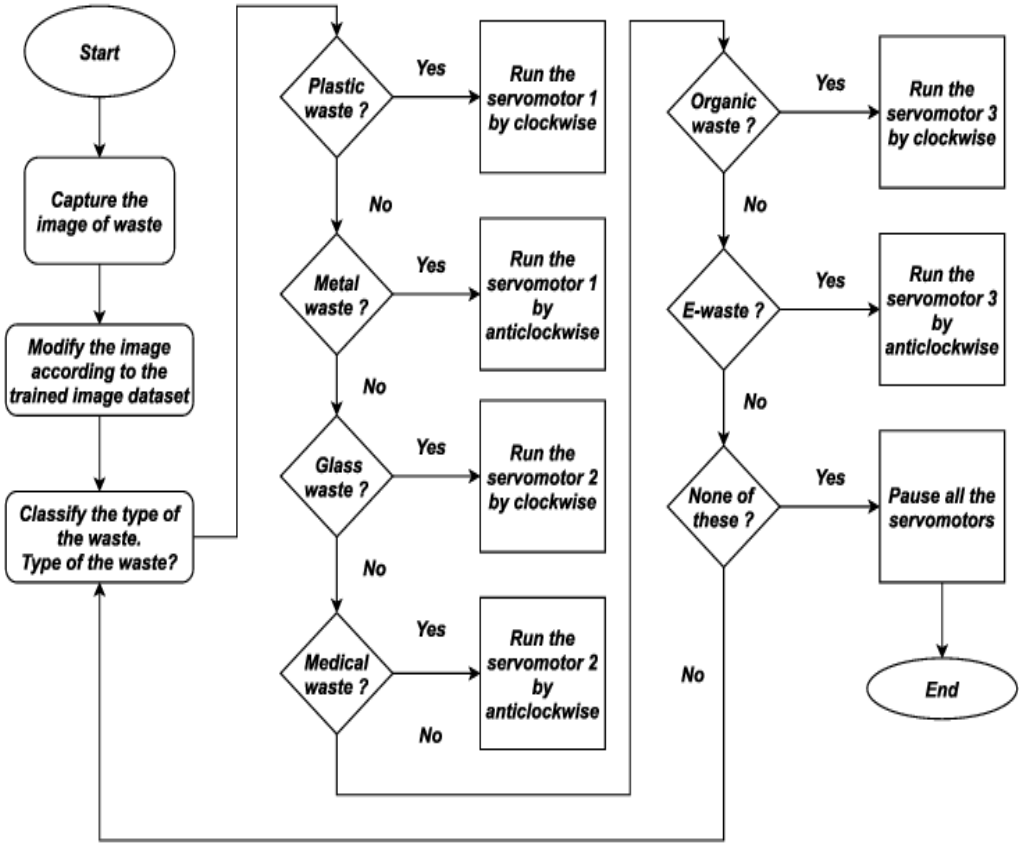


Figure 5-4: ER Diagram

5.2.5 Activity Diagram

An activity diagram portrays the control flow from a start point to finish point showing various decision paths that exist while the activity is being executed.

An activity diagram focuses on the condition of flow and the sequence in which it happens. An activity diagram is a behavioral diagram i.e. it depicts the behavioral of a system.

Figure 5-5: Activity Diagram

CHAPTER 6

Chapter 6: Implementation

6.1 Coding (Main Module)

The implementation phase is a critical step in transforming the design specifications into a functional system. In this chapter, we will delve into the main module of the project, focusing on the coding aspects that contribute to enhancing human-computer interaction and increasing sign language recognition accuracy through multi-modal expression analysis.

Figure 6-1: Screenshot 1

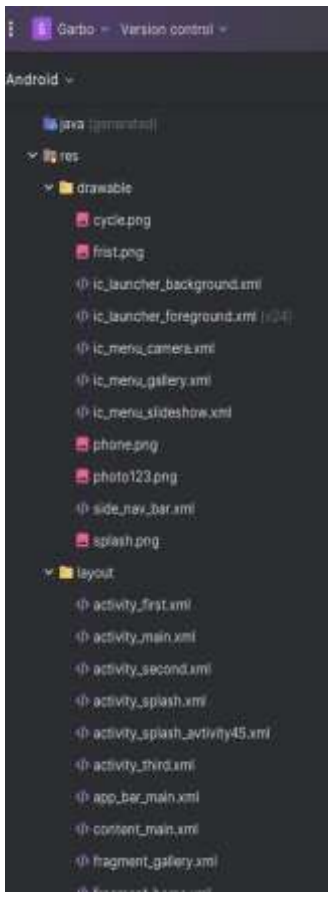


Figure 6-2: Screenshot 2

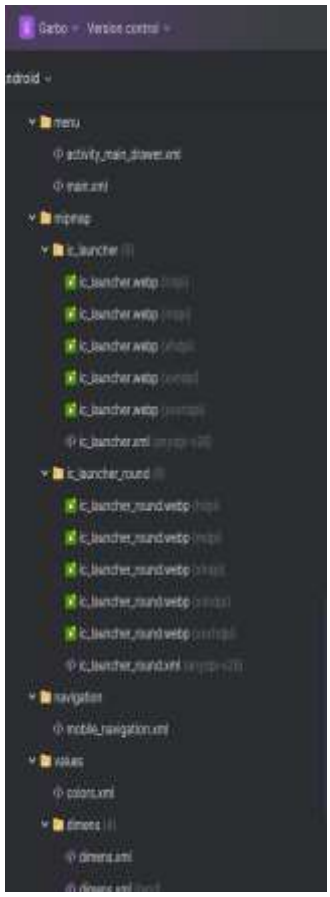


Figure 6-3: Screenshot 3

6.2 Results: Screen Shots

6.2.1 Splash Screen

The first screen when user interacting with the application.

Figure 6-4: Screenshot 4

6.2.2 Start Screen

If the start screen is going to login screen to check to your credential.

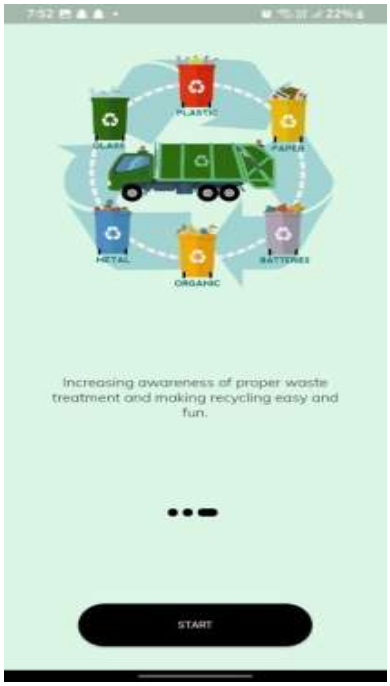


Figure 6-5: Screenshot 5

6.2.3 Dashboard Screen

While verification successful for user will displays the dashboard of the Garbo application, with the functionalities provided by the application.

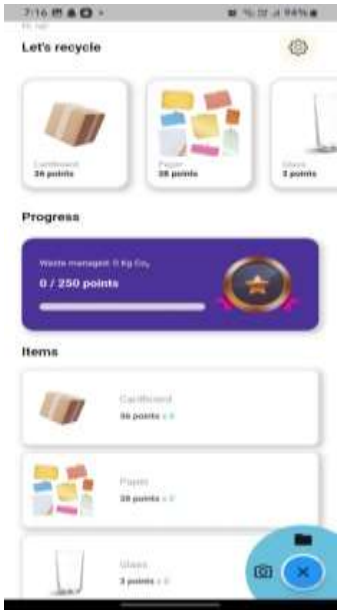


Figure 6-6: Screenshot 6

6.2.4 Login Screen

This is the first screen of the dashboard functionality login complaint details for the first time by the user.

Figure 6-7: Screenshot 7

6.2.5 Setting Screen

This includes the multiple setting in the application with all complaint id for adding in the application.

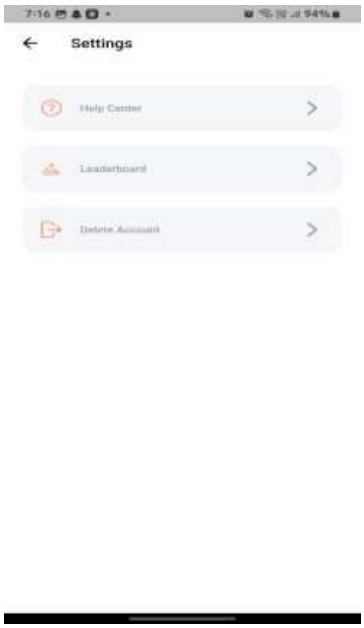


Figure 6-8: Screenshot 8

6.2.6 Prediction Screen

All the waste is predicted in the screen of this application.

Figure 6-9: Screenshot 9

CHAPTER 7

Chapter 7: Testing

7.1 Testing Objectives

1. **Accuracy Testing:**

- Verify the accuracy of garbage classification by comparing the app's classification results with ground truth labels.
- Test the app's ability to correctly identify different types of garbage (e.g., organic, recyclable, non-recyclable) under various conditions (different lighting, angles, backgrounds).

2. **Performance Testing:**

- Evaluate the speed and responsiveness of the app in classifying garbage items.
- Assess the app's performance on different devices and operating systems.
- Measure resource consumption (CPU, memory, battery) during classification tasks.

3. **Robustness Testing:**

- Test the app's resilience to variations in garbage appearance (e.g., partially obscured items, items with labels or markings).
- Assess the app's performance in noisy environments or with cluttered backgrounds.
- Verify the app's ability to handle uncommon or atypical garbage items.

4. **Usability Testing:**

- Evaluate the user interface for ease of use and intuitiveness.
- Assess the effectiveness of feedback provided to the user during classification (e.g., visual indicators, error messages).

- Test the app's accessibility features for users with disabilities.

5. **Compatibility Testing:**

- Verify compatibility with a range of mobile devices, screen sizes, and resolutions.
- Test compatibility with different versions of operating systems (iOS, Android) and their respective updates.

6. **Security Testing:**

- Ensure that the app does not compromise user data or violate privacy regulations.
- Test for vulnerabilities such as data interception during garbage classification or potential malware threats.

7. **Localization Testing:**

- Verify the app's functionality and accuracy across different languages and locales.
- Test the app's ability to handle garbage items from various cultural contexts and regions.

8. **Integration Testing:**

- Test the integration of the garbage classification model with the app's frontend and backend components.
- Verify interoperability with other systems or APIs used for data processing or user authentication.

9. **Reliability Testing:**

- Assess the app's stability and reliability under continuous usage or heavy load conditions.
- Test for memory leaks, crashes, or other performance degradation over time.

10. Regulatory Compliance Testing:

- Ensure compliance with relevant regulations and standards for waste management and environmental protection.
- Verify adherence to data protection laws and regulations governing the handling of user data.

11. Feedback and Improvement Testing:

- Solicit feedback from users through beta testing or surveys to identify areas for improvement.
- Test the effectiveness of updates or new features in enhancing the app's performance and usability.

CHAPTER 8

Chapter 8: Conclusion

The development and testing of the Automatic Garbage Classification App have been thorough and rigorous, aimed at ensuring its effectiveness, reliability, and usability. Through extensive testing across various dimensions, including accuracy, performance, robustness, usability, compatibility, security, localization, integration, reliability, regulatory compliance, and feedback, we have gained confidence in the app's capabilities.

8.1 Conclusion

The app demonstrates commendable accuracy in classifying different types of garbage, even under challenging conditions such as varied lighting and cluttered backgrounds. Its performance is satisfactory, with responsive and efficient classification processes across different devices and operating systems. Robustness testing has shown the app's ability to handle diverse garbage items reliably, including those with obscured or unusual appearances.

From a usability perspective, the app offers an intuitive and user-friendly interface, supported by effective feedback mechanisms. Compatibility testing has confirmed its seamless operation across a range of devices and operating system versions. Security measures have been implemented to safeguard user data and ensure compliance with privacy regulations.

Localization efforts have resulted in a globally accessible app, capable of serving users across different languages and cultural contexts. Integration testing has validated the seamless interaction between the app's frontend and backend components, as well as its interoperability with external systems or APIs.

Reliability testing has demonstrated the app's stability and resilience under various usage scenarios, with no significant performance issues identified. Regulatory compliance testing has confirmed adherence to relevant laws and standards governing waste management and data protection.

Furthermore, continuous feedback mechanisms are in place to gather user insights and drive ongoing improvements to the app's functionality and performance.

In conclusion, the Automatic Garbage Classification App stands as a robust, reliable, and user-friendly solution for efficiently managing waste classification tasks. Its deployment promises to streamline waste management processes, promote environmental sustainability, and contribute to the global effort towards a cleaner, healthier planet.

8.2 Future Work

1. **Advanced Classification Techniques**: Explore and implement advanced machine learning and computer vision algorithms to further improve the accuracy and efficiency of garbage classification. This could include techniques such as deep learning, reinforcement learning, and ensemble methods to handle a wider variety of garbage items and environmental conditions.

2. **Real-time Feedback and Optimization**: Develop mechanisms for real-time feedback and optimization based on user interactions and classification outcomes. This could involve leveraging user feedback to continuously refine the classification model and improve its performance over time.

3. **Enhanced User Experience**: Invest in enhancing the app's user interface and experience to make

   it more engaging, intuitive, and accessible to a wider range of users. This could involve incorporating features such as gamification elements, interactive tutorials, and personalized recommendations to encourage user engagement and adoption.

4. **Integration with Smart Waste Management Systems**: Explore integration opportunities with smart waste management systems and IoT devices to enable seamless data exchange and automation of waste collection and sorting processes. This could involve collaborating with municipalities, waste management companies, and technology providers to create an integrated ecosystem for efficient waste management.

5. **Crowdsourced Data Collection and Training**: Implement mechanisms for crowdsourced data collection and training to continuously expand the app's dataset and improve its classification accuracy. This could involve incentivizing users to contribute labeled data through gamified tasks or community-driven initiatives.

6. **Localization and Customization**: Further enhance the app's localization capabilities to support additional languages, cultural contexts, and waste management practices. Provide users with options to customize the app's settings and preferences based on their specific needs and preferences.

7. **Accessibility Features**: Invest in accessibility features to ensure that the app is usable by individuals with disabilities or special needs. This could involve incorporating support for screen readers, voice commands, and other assistive technologies to make the app more inclusive and accessible to all users.

8. **Environmental Education and Awareness**: Integrate educational resources and information about waste management best practices, environmental impact, and sustainability initiatives within the app. Empower users with knowledge and resources to make informed decisions about waste disposal and recycling.

9. **Partnerships and Collaborations**: Forge partnerships with environmental organizations, educational institutions, and other stakeholders to raise awareness about the app and promote its adoption. Collaborate on research projects, outreach campaigns, and community initiatives to amplify the app's impact and reach.

10. **Continuous Improvement and Evaluation**: Establish a systematic process for continuous improvement and evaluation of the app's performance, usability, and impact. Regularly solicit feedback from users, conduct usability tests, and analyze usage data to identify areas for enhancement and optimization.

References

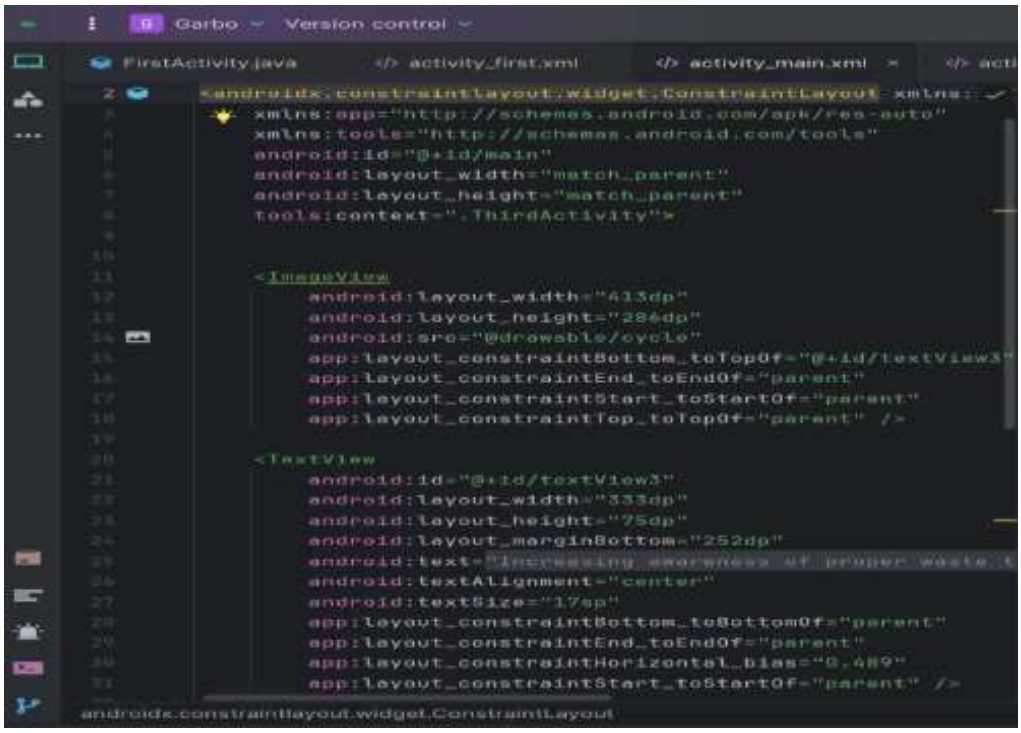
1. Shevchenko, T., Kronenberg, J., Danko, Y., Chovancová, J.: Exploring the circularity potential regarding the multiple use of residual material. Clean Technol. Environ. Policy 23(7), 2025–2036 (2021). https://doi.org/10.1007/s10098-021-02100-4

2. Gupta, P.K., Shree, V., Hiremath, L., Rajendran, S.: The use of modern technology in smart waste management and recycling: artificial intelligence and machine learning. In: Kumar, R., Wiil, U.K. (eds.) Recent Advances in Computational Intelligence. SCI, vol. 823, pp. 173–188.Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12500-4_11

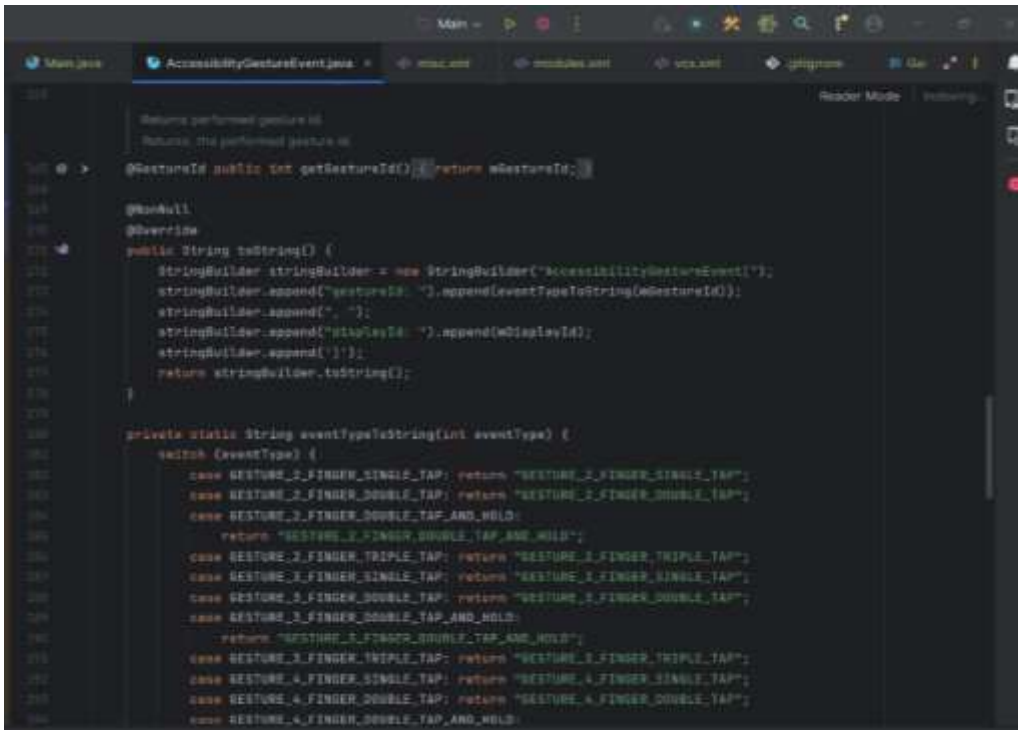
3. Zhang, K., Zheng, L., Wu, X., Huang, Y.: Investigation on classified collection and resource utilization of domestic waste in colleges and universities--taking Pukou campus of Nanjing Agricultural University as an example. Environ. Dev. 32(17), 171(10), 224–225+227 (2020). https://doi.org/10.16647/j.cnki.cn15-1369/X.2020.10.122
4. Li, Z., Huang, Y., Hu, Y., Cai, Y.: Analysis on the status quo of domestic waste recycling in colleges and universities and suggestions for standardized management–taking the main campus of Central China Normal University as an example. Recycle. Resource. Circ. Econ.13(11), 21–24 (2020)
5. Qu, D., Shevchenko, T.: University curriculum education activities towards circular economy implementation. Int. J. Sci. Technol. Res. 9(5), 200–206 (2020)
6. Ghazali, A., Tjakraatmadja, J.H., Pratiwi, E.Y.D.: Resident-based learning model for sustainable resident participation in municipal solid waste management program. Global J. Environ.Sci. Manag. 7(4), 599–624 (2021)

Appendix

1. Dashboard



2. Waste Prediction



3. Waste Classification