EcoScrap : Real Time Garbage Detection App

ENROLLMENT NO. S	NAME	BATCH
22103005	VANDIT KAUL	B1
22103025	NAVDEEP LADDHA	B1
22103022	ANUBHAV TYAGI	B1

SUPERVISOR- DR. VARUN SRIVASTAVA

SUBMITTED TO- DR. AMARJEET PRAJAPATI, DR. ANUJA SHUKLA

November– 2024

Submitted in partial fulfillment of the Degree of

Bachelor of Technology

in

Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

(I) TABLE OF CONTENTS

(III)

CERTIFICATE

This is to certify that the work titled **“EcoScrap : Real Time Garbage Detection App”** submitted by "**Vandit Kaul, Navdeep Laddha and Anubhav Tyagi”** in partial fulfillment for the award of the degree of **Bachelor of Technology in Computer Science Engineering** of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor ______________________

Name of Supervisor Dr. Varun Srivastava

Designation Assistant Professor (Senior Grade)

Date ______________________

(IV)

ACKNOWLEDGEMENT

Our team members would sincerely like to thank our college Jaypee Institute of Information Technology, Sector 62 Noida who gave us the opportunity to work on a project under the guidance of Dr. Ankit Vidyarthi sir.

We would like to express our sincere thanks and gratitude to our mentor Dr. Varun Srivastava for letting us work on this project. We are very grateful to him for his support and guidance in completing this project.

Your valuable guidance and support helped us in various phases of completion of this project. We will always be thankful to you in this regard.

Signature of the Student ______________________

Name of the Student Vandit Kaul

Enrollment Number 22103005

Date ______________________

Signature of the Student ______________________

Name of the Student Navdeep Laddha

Enrollment Number 22103025

Date ______________________

Signature of the Student ______________________

Name of the Student Anubhav Tyagi

Enrollment Number 22103022 **Date** ______________________

SUMMARY

This project addresses the pressing challenges of improper waste management, which leads to environmental degradation, public health risks, and inefficient recycling practices. The traditional methods of waste segregation and recycling are labor-intensive, error-prone, and lack accessibility to organized recycling systems.

The proposed "EcoScrap: Real-Time Garbage Detection App" leverages advanced machine learning (ML) techniques and mobile technology to automate garbage classification and connect waste generators with recyclers. By integrating a PyTorch-based Convolutional Neural Network (CNN) for accurate material detection and an Android application for user interaction, the system ensures real-time functionality and accessibility.

This solution minimizes user effort by automating waste classification and providing tailored recycling recommendations based on material type and user location. It promotes sustainability by encouraging responsible waste disposal habits and fostering connections between users and recyclers, with a 10% commission revenue model to ensure platform viability.

For the Jaypee Institute of Information Technology and similar institutions, this project demonstrates a scalable, efficient, and user-centric approach to improving waste management. It aligns with global efforts to combat environmental challenges and fosters a sustainable, circular economy by turning waste into a valuable resource.

1. INTRODUCTION

1.1 General Introduction

Waste management is one of the most pressing environmental challenges faced globally and locally. Rapid urbanization, industrial growth, and population expansion have resulted in an exponential increase in waste generation. Globally, around **2 billion tonnes** of solid waste is produced annually, of which a significant portion remains uncollected or improperly disposed of, leading to severe environmental hazards such as air, soil, and water pollution.

In developing nations like India, improper waste segregation remains a critical issue. Waste often ends up in landfills without prior classification, making recycling and recovery processes inefficient. For instance, **only 30% of plastic waste in India is recycled**, while the rest contributes to growing landfill volumes and marine pollution. Additionally, unsegregated waste hampers the recycling of biodegradable materials and increases the demand for virgin raw materials, further straining natural resources.

Economic consequences of poor waste management include higher municipal costs for collection and disposal, reduced economic opportunities in recycling industries, and loss of valuable materials. Socially, unmanaged waste exposes communities to health hazards such as respiratory diseases and vector-borne illnesses.

1.1.1 Objective

The "EcoScrap: Real-Time Garbage Detection App" seeks to address these challenges by leveraging technology to automate waste segregation and streamline recycling processes. The primary objectives of the project include:

1. Utilizing a **machine learning-based garbage detection model** to classify waste materials accurately.
2. Providing users with an easy-to-use mobile application to connect them with recyclers and rag collectors (kabadiwalas).
3. Encouraging users to participate in recycling by simplifying the process and offering a transparent platform for waste management.

The goal is to build a sustainable ecosystem where individuals can contribute to waste reduction while enabling recyclers to access pre-segregated materials. This automation not only enhances recycling efficiency but also promotes environmental responsibility and fosters a circular economy.

1.1.2 Scope

The project aligns with global efforts to combat waste mismanagement and achieve sustainability goals. It contributes to the **United Nations' Sustainable Development Goal 12 (Responsible Consumption and Production)** by encouraging efficient resource use and promoting recycling practices.

By integrating technologies like **machine learning, mobile application development, and real-time data management**, this solution also fits into the framework of **smart city initiatives**. Cities worldwide are adopting smart waste management systems to optimize resources, reduce costs, and lower their carbon footprints.

The "EcoScrap" project offers scalability, as it can be extended to include more waste types, expand geographically, and partner with municipal waste management authorities. With its ability to automate processes, provide real-time recommendations, and foster community participation, the app represents a significant step toward sustainable and efficient waste management solutions in both urban and rural contexts.

1.2 Problem Statement: Key Issues in Waste Management

1. Manual Segregation Inefficiency

- ❖ Waste segregation at the source is predominantly manual and prone to human error.
- ❖ Households and businesses often mix biodegradable and non-biodegradable waste, complicating the process for recyclers to extract reusable materials efficiently.
- ❖ The absence of standardized sorting methods leads to inconsistencies, reducing recycling effectiveness and increasing landfill burden.

2. Lack of Accessibility to Recycling Systems

- Many individuals lack awareness of the environmental and economic benefits of proper waste segregation.
- Even environmentally conscious individuals face challenges in accessing organized recycling systems, such as rag collectors (kabadiwalas) or municipal facilities.
- This inaccessibility discourages active participation in sustainable waste disposal practices.

3. Limited Use of Technology in Waste Management

- Technological advancements, such as AI and real-time data, have minimal application in waste management.
- Lack of technological integration creates a disconnect between waste generators (individuals and businesses) and recyclers.
- This disconnect results in missed opportunities for recycling valuable materials, including plastics, metals, and paper.

1.2.1 Impact

The inefficiencies in waste management systems have far-reaching consequences:

- **Environmental Consequences**
  - Improperly segregated waste contributes to the pollution of land, air, and water bodies. Non-biodegradable waste, such as plastics, persists in the environment for centuries, harming ecosystems and marine life.
  - Decomposing waste in landfills generates methane, a potent greenhouse gas that accelerates climate change.
- **Societal Consequences**
  - Unmanaged waste creates unhygienic living conditions, leading to the spread of diseases such as cholera and dengue.
  - Communities near landfills or waste dumping sites often face health hazards, reduced property values, and compromised quality of life.

- ❖ **Economic Consequences**
  - ➢ The lack of an organized recycling system results in the loss of potentially reusable materials, increasing the demand for virgin raw materials and driving up production costs.
  - ➢ Municipal authorities incur higher costs for waste collection, transportation, and landfill management.
  - ➢ The recycling industry loses opportunities for economic growth due to insufficient waste segregation and recovery.

1.2.2 Need for a Tech-Driven Solution

To address these pressing challenges, there is a need for an innovative, technology-driven approach that bridges the gap between waste generators and recyclers. A robust system that integrates **machine learning (ML)** for accurate waste classification, **real-time mobile applications** for user accessibility, and a **centralized database** for efficient coordination can revolutionize waste management.

Such a solution can reduce manual segregation errors, enhance recycling rates, and contribute to sustainable development. By automating waste segregation and providing easy access to recycling systems, the "EcoScrap" project aims to tackle the inefficiencies of current waste management practices while promoting environmental responsibility and community participation.

1.3 Significance/Novelty of the Problem

1.3.1 Innovation

The "EcoScrap: Real-Time Garbage Detection App" introduces a cutting-edge approach to tackling waste management challenges by leveraging advanced technology and user-friendly design. The following unique aspects set this project apart from conventional methods:

- **Real-Time Garbage Detection with PyTorch**
  - The project utilizes a deep learning model built using PyTorch to classify garbage into categories such as plastic, metal, paper, and more.
  - The real-time image recognition feature automates waste segregation at the source, minimizing human intervention and errors.
  - By deploying this model within the mobile app, the system ensures that users receive instant feedback on the type of waste they are disposing of.
- **Seamless User Experience with Kotlin-Based Android App**
  - The app is designed with a user-centric interface developed in Kotlin, ensuring smooth navigation and functionality.
  - Users can easily capture images of garbage, provide location details, and initiate the selling process without technical knowledge.
  - The integration of Firebase for real-time database management ensures swift communication between users, admins, and recyclers.

1.3.2 Importance

The project addresses significant environmental, societal, and economic issues by fostering sustainable behavior and creating a circular economy.

- **Encouraging Sustainable Behavior**
  - The app simplifies waste disposal by automating segregation, motivating users to participate in recycling efforts.
  - By connecting individuals to nearby recyclers or rag collectors, it eliminates barriers to responsible waste management.
  - Awareness about proper waste handling increases as users interact with the platform.
- **Reducing Landfill Pressure**
  - Automated segregation ensures that recyclable materials like plastics and metals are diverted from landfills, decreasing the strain on these already-overburdened systems.

    - ➢ Compostable materials can be directed to appropriate facilities, reducing methane emissions from organic waste decomposition in landfills.

- ❖ **Promoting Recycling**
    - ➢ By enabling easy access to recyclers, the project increases the amount of waste that is recycled instead of discarded.
    - ➢ The seamless integration of material classification with location-based recommendations supports efficient resource recovery and reduces dependency on virgin raw materials.

1.3.3 Novel Features

The "EcoScrap" app incorporates several innovative features to create a comprehensive and efficient waste management system:

- ❖ **Integration of Location-Based Recycler Recommendations**
    - ➢ The app uses geolocation to suggest nearby recyclers and rag collectors, optimizing the collection process.
    - ➢ This feature saves users time and ensures that materials are sent to appropriate recycling facilities promptly.
- ❖ **Revenue Model**
    - ➢ A 10% commission on every transaction generates consistent revenue, making the platform financially sustainable.
    - ➢ Additional revenue streams, such as partnerships with recyclers, can further enhance the platform's profitability.
- ❖ **Real-Time Interaction**
    - ➢ The app bridges the gap between waste generators and recyclers, enabling direct communication and fostering trust within the ecosystem.
    - ➢ Admins can monitor user activity and initialize orders for efficient collection and recycling.
    - ➢ By combining cutting-edge technology with an intuitive design and practical functionality, this project represents a novel solution to the global waste

management problem. It not only promotes environmental sustainability but also creates a scalable and profitable business model.

1.4 Empirical Study

1.4.1 Data Collection

The development of an accurate and reliable machine learning (ML) model for garbage detection required a well-structured dataset comprising diverse images of waste materials.

- ❖ **Dataset Sourcing**
    - ➢ **Publicly Available Datasets**: We utilized datasets from platforms such as Kaggle and ImageNet, which provided labeled images of garbage items like plastic bottles, metal cans, paper, and organic waste.
    - ➢ **Custom Data Collection**: To improve the model's accuracy, we captured real-world images using smartphones. These images were taken in various lighting conditions and backgrounds to account for real-time application scenarios.
    - ➢ **Diversity in Data**: The dataset included different types of waste materials, such as plastics, metals, paper, glass, and organic waste, ensuring the model could classify common categories effectively.
- ❖ **Data Preprocessing**
    - ➢ **Annotation**: Each image in the dataset was labeled manually with its respective material type.
    - ➢ **Augmentation**: Techniques like rotation, scaling, flipping, and brightness adjustment were applied to increase dataset size and improve the model's robustness.
    - ➢ **Normalization**: The images were resized and normalized to ensure consistency during training and reduce computational complexity.
    - ➢ **Splitting**: The dataset was divided into training (70%), validation (20%), and testing (10%) subsets for unbiased model evaluation.

1.4.2 Analysis

Through the data collection and research phase, several insights were gained that guided the project's design and implementation:

- **Waste Segregation Patterns**
  - Research indicated that **plastic waste** constitutes the largest percentage of urban waste, followed by metals and paper.
  - Improper segregation was found to be most common in mixed-use waste containers, where recyclables and non-recyclables were disposed of together.
  - A lack of public knowledge about recyclable and non-recyclable materials emerged as a significant factor contributing to improper segregation.

- **User Needs and Preferences**
  - Surveys conducted among potential users revealed that **ease of use** and **minimal manual intervention** were critical requirements for adopting a waste management app.
  - Users preferred systems that could provide instant classification of waste materials without requiring them to understand complex categories.
  - The importance of location-based recommendations was highlighted, as users preferred knowing the nearest recycler or collection point.

- **Challenges in Manual Systems**
  - Feedback from waste collectors emphasized the inefficiency and inaccuracies of manual segregation, underscoring the need for automated systems to improve recycling rates.

1.4.3 Relevance

The data collected and analyzed during this phase strongly supports the viability of the proposed solution:

- **Model Performance**
  - The diversity of the dataset ensures that the deep learning model can accurately classify waste materials in varied real-world scenarios.
  - Augmentation and preprocessing techniques enhanced the model's ability to generalize, ensuring reliable performance across different environmental conditions.
- **Alignment with User Needs**
  - By incorporating instant image recognition and location-based recommendations, the project addresses user pain points directly identified during the research phase.
  - The app's intuitive design and automated functionality cater to the widespread need for accessibility and simplicity in waste segregation.
- **Practicality and Scalability**
  - The insights into waste patterns enable the app to prioritize the most commonly generated waste types, making it highly practical for urban settings.
  - The modular design of the solution allows for future expansion, such as adding more waste categories or integrating additional features like rewards for recycling participation.

1.5 Brief Description of the Solution Approach

1.5.1 Overview

The "EcoScrap" project provides a comprehensive solution to modernize waste management using technology. The workflow integrates machine learning, geolocation services, and a user-friendly mobile application to facilitate efficient garbage segregation and recycling. The approach is designed to simplify the process for users while ensuring scalability and accuracy.

- ❖ **User Interaction**
  - ➢ Users upload an image of the waste material via the mobile application.
  - ➢ Additional details, such as location, weight, and pickup preferences (date and time), are entered by the user.
- ❖ **Material Detection**
  - ➢ The PyTorch-based ML model processes the uploaded image to identify and classify the waste material into categories such as plastic, metal, paper, or others.
  - ➢ Classification results are displayed instantly within the app, giving users clarity on the waste type.
- ❖ **Recycler Connection**
  - ➢ Based on the user's geolocation, the app identifies and suggests nearby rag collectors or recyclers.
  - ➢ Users confirm the order, and the details are forwarded to the admin panel for processing.
- ❖ **Order Management**
  - ➢ Admins review the submitted orders via a dedicated panel.
  - ➢ Once verified, recyclers are notified to collect the waste from the user's location.
- ❖ **Revenue Model**
  - ➢ A 10% platform fee is charged for each transaction, generating revenue while promoting a circular economy.

1.5.2 Key Components

- ❖ **Mobile Application**
  - ➢ **Platform**: Developed using Kotlin on Android Studio.
  - ➢ **Features**: Intuitive interface, real-time order updates, camera integration, and geolocation support.
  - ➢ **Functionality**: Enables image uploads, user registration, and order tracking.

- **ML Model**
  - **Technology**: A deep learning model built with PyTorch for image recognition.
  - **Functionality**: Classifies garbage materials with high accuracy using a well-prepared dataset.
  - **Integration**: Embedded within the app to process images in real-time, ensuring seamless user interaction.
- **Database**
  - **Technology**: Firebase for real-time data synchronization.
  - **Functionality**: Manages user information, transaction records, and material classifications.
  - **Scalability**: Ensures reliable storage and retrieval of data even with an increasing number of users.

1.5.3 Workflow

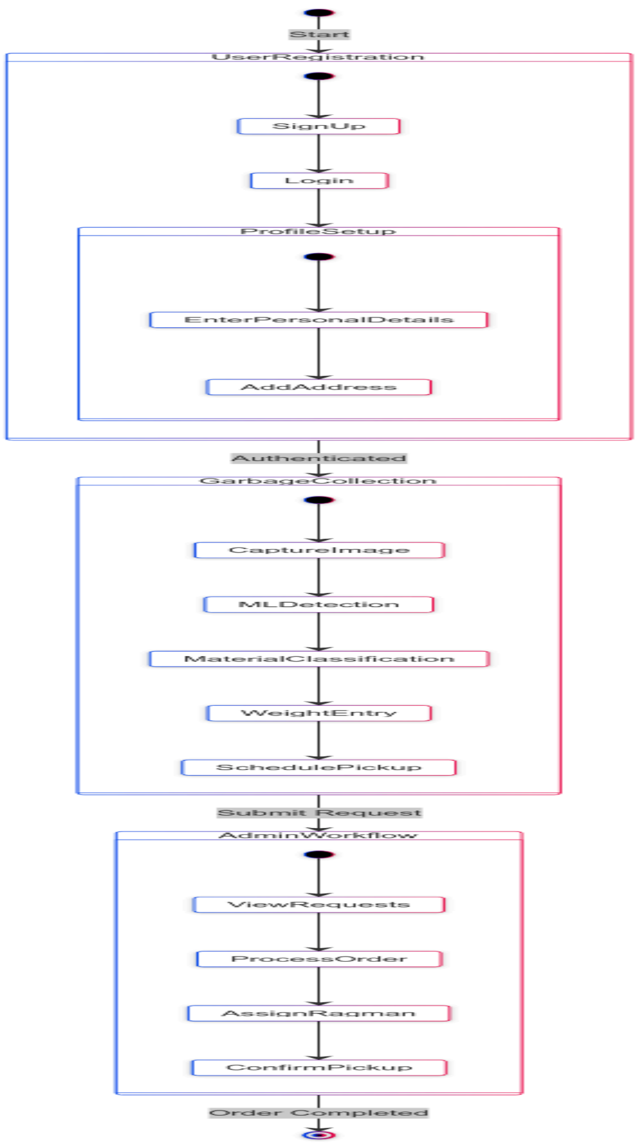
To provide a clear picture of the system's operation, a step-by-step description of the workflow is provided below:

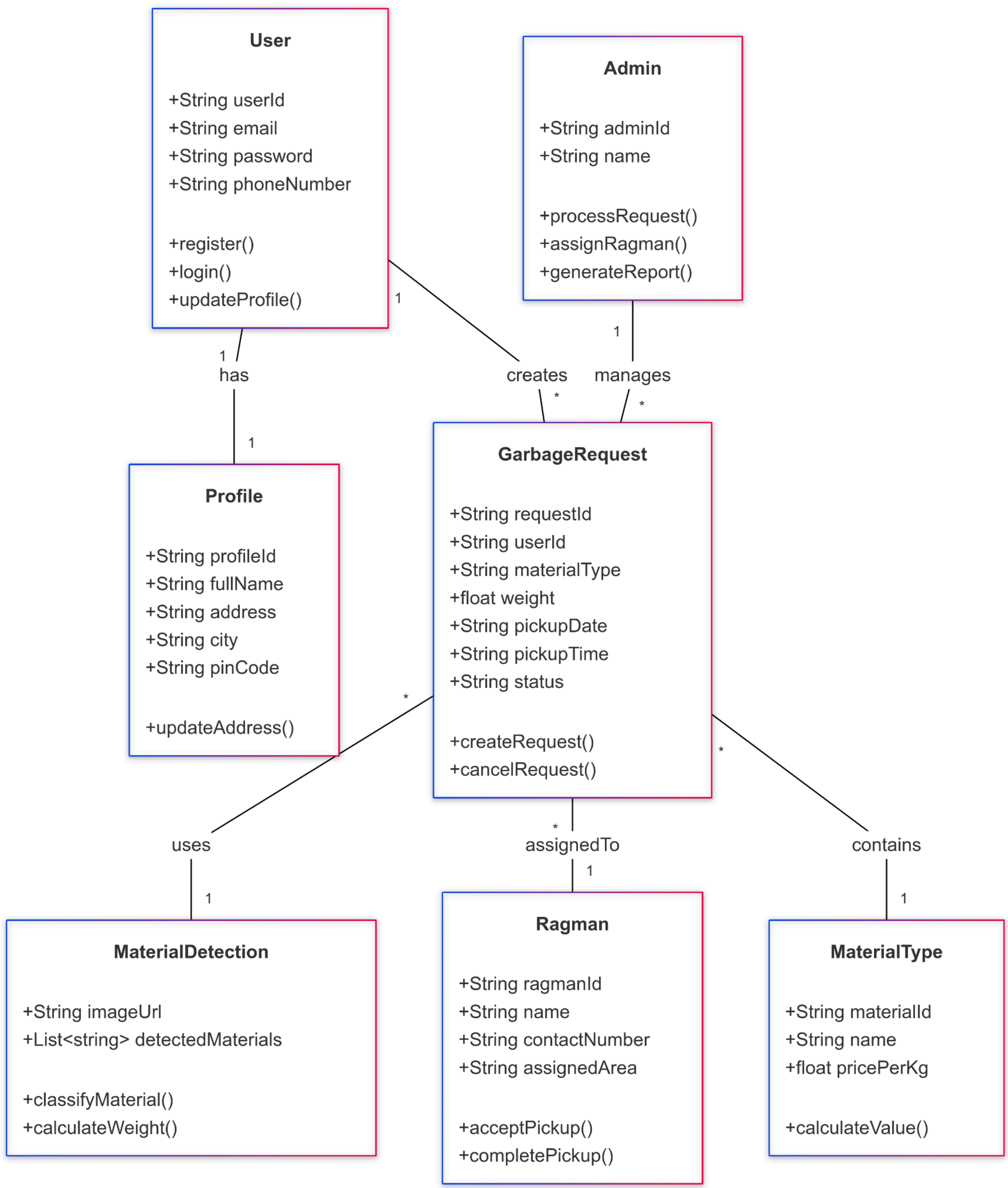
- **User Input**
  - Users launch the app, register, or log in.
  - They capture and upload an image of the waste material through the app.
- **Material Classification**
  - The ML model processes the image and identifies the waste type.
  - Results are displayed along with material-specific recycling suggestions.
- **Order Initialization**
  - Users provide additional details such as weight, location, and pickup preferences.
  - The app uses geolocation to recommend nearby recyclers.
- **Admin Processing**
  - Admins review user requests and confirm orders.
  - Once confirmed, recyclers are notified with the collection details.

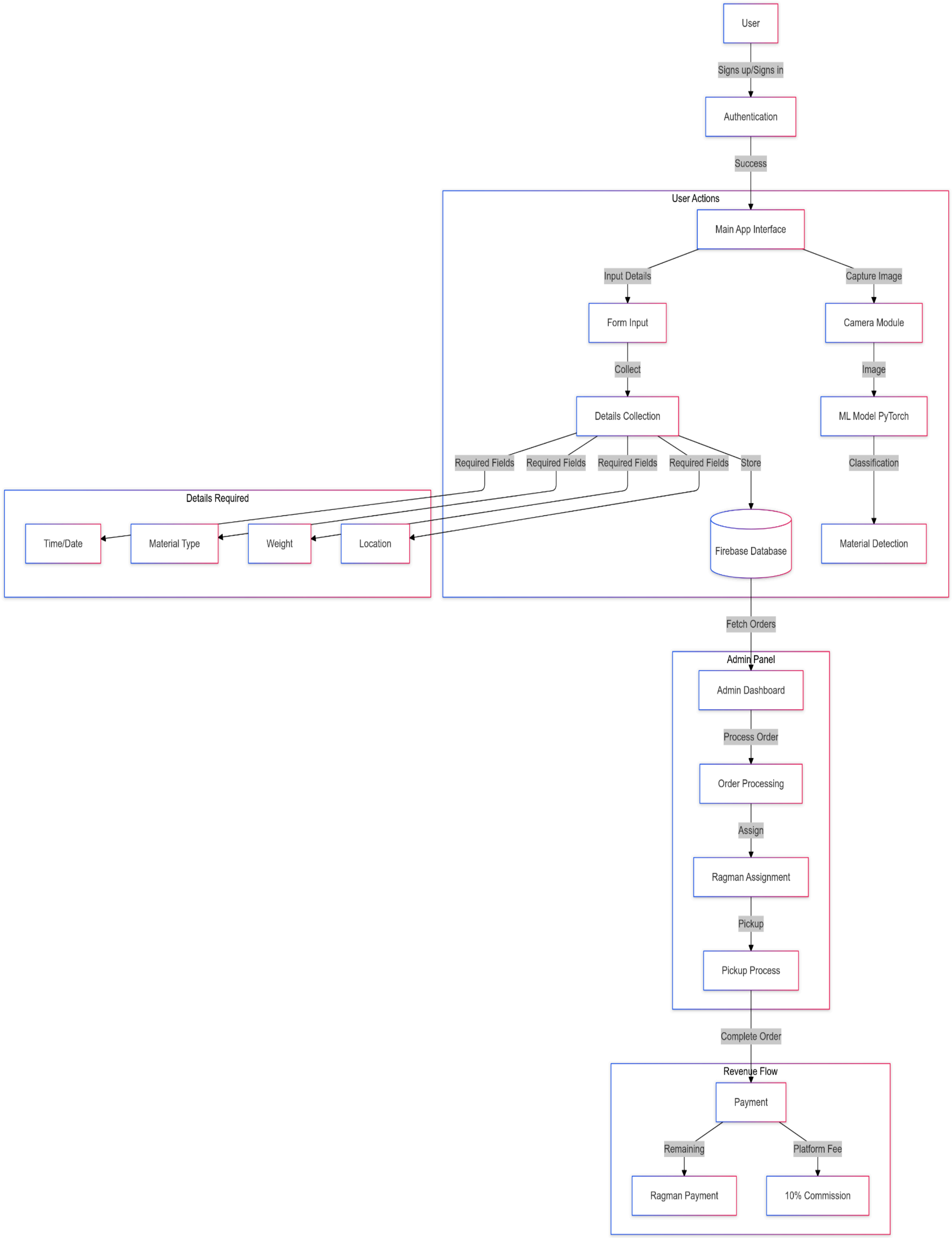
- ❖ **Collection and Recycling**
  - ➢ Recyclers visit the user's location to collect the waste.
  - ➢ The collected waste is sent for appropriate recycling or reuse.

Flowchart of the Workflow

A flowchart can effectively visualize the process. Below is a textual representation of the flowchart for this section (can be converted into a diagram later):




User
+String userId
+String email
+String password
+String phoneNumber
+register()
+login()
+updateProfile()
Admin
+String adminId
+String name
+processRequest()
+assignRagman()
+generateReport()
1
has
1
1
creates
*
1
manages
*
Profile
+String profileId
+String fullName
+String address
+String city
+String pinCode
+updateAddress()
GarbageRequest
+String requestId
+String userId
+String materialType
+float weight
+String pickupDate
+String pickupTime
+String status
+createRequest()
+cancelRequest()
*
uses
1
*
assignedTo
1
*
contains
1
MaterialDetection
+String imageUrl
+List<string> detectedMaterials
+classifyMaterial()
+calculateWeight()
Ragman
+String ragmanId
+String name
+String contactNumber
+String assignedArea
+acceptPickup()
+completePickup()
MaterialType
+String materialId
+String name
+float pricePerKg
+calculateValue()


User
Signs up/Signs in
Authentication
Success
User Actions
Main App Interface
Input Details
Capture Image
Form Input
Camera Module
Collect
Image
Details Collection
ML Model PyTorch
Required Fields
Required Fields
Required Fields
Required Fields
Store
Classification
Details Required
Time/Date
Material Type
Weight
Location
Firebase Database
Material Detection
Fetch Orders
Admin Panel
Admin Dashboard
Process Order
Order Processing
Assign
Ragman Assignment
Pickup
Pickup Process
Complete Order
Revenue Flow
Payment
Remaining
Platform Fee
Ragman Payment
10% Commission

1.6 Comparison of Existing Approaches to the Problem Framed

1.6.1 Existing Solutions

Several apps and platforms address waste management; however, most have notable limitations that hinder their effectiveness in solving the problem comprehensively:

- ❖ **Apps Without ML-Based Classification**
  - ➢ These apps rely heavily on users to manually classify the type of waste, leading to inaccuracies and user frustration.
  - ➢ Examples: General waste collection apps that focus on logging waste pickup requests but lack automated classification.
  - ➢ **Limitation**: High dependency on user input and prone to errors, making the process inefficient and less user-friendly.
- ❖ **Platforms Focusing Only on Specific Waste Categories**
  - ➢ Many existing platforms focus exclusively on specialized categories like **e-waste** (electronics) or non-recyclable materials, neglecting broader waste types like plastics, metals, or paper.
  - ➢ Examples: Apps designed to collect only e-waste for safe disposal.
  - ➢ **Limitation**: Limited scope of functionality, reducing their impact on general waste management needs.
- ❖ **Lack of Integration Between Waste Generators and Recyclers**
  - ➢ Few solutions establish direct communication between users (waste generators) and recyclers, often requiring intermediaries to manually match waste with recycling facilities.
  - ➢ Examples: Platforms that offer collection services but do not recommend recyclers based on user location or waste type.
  - ➢ **Limitation**: Increased time and cost due to inefficiencies in connecting stakeholders.

- ❖ **Minimal Use of Technology**
  - ➢ Many waste management systems operate with minimal technological integration, relying on manual processes or basic features like GPS-based pickup services.
  - ➢ Examples: Traditional waste pickup services provided by municipalities or local collection agencies.
  - ➢ **Limitation**: Lack of innovation leads to slower processing, lower recycling rates, and poor user engagement.

1.6.2 Advantage of Your Solution

The "EcoScrap" project introduces a unique, tech-driven approach that overcomes these limitations and provides several advantages:

- ❖ **Real-Time, Automated Garbage Detection**
  - ➢ By integrating a PyTorch-based ML model, the app automates waste classification.
  - ➢ This reduces dependency on user inputs and ensures higher accuracy, even in real-world conditions with diverse waste materials.
  - ➢ **Advantage**: Saves time and improves the reliability of waste segregation compared to manual methods.
- ❖ **Comprehensive Material Classification**
  - ➢ Unlike other solutions, "EcoScrap" supports a broad range of waste types, including plastics, metals, paper, glass, and organics.
  - ➢ The ML model is trained on a diverse dataset, ensuring adaptability to real-life scenarios.
  - ➢ **Advantage**: Addresses the needs of a wide user base, from households to industries, making it a versatile solution.
- ❖ **Seamless User-to-Recycler Connectivity**
  - ➢ The app uses geolocation to recommend the nearest recyclers or rag collectors to users.

- ➢ It also provides a platform for recyclers to interact with users directly, minimizing the need for intermediaries.
- ➢ **Advantage**: Enhances efficiency in waste collection and recycling, reducing costs and environmental impact.

❖ **User-Friendly Mobile Application**

- ➢ Built using Kotlin, the app offers a seamless user experience with easy navigation, real-time updates, and integration of camera features for image capture.
- ➢ **Advantage**: Encourages adoption by simplifying the waste management process for users of all technical backgrounds.

❖ **Holistic Integration**

- ➢ Unlike platforms that focus on isolated aspects (e.g., only waste collection or disposal), "EcoScrap" provides an end-to-end solution:
  - ■ Waste detection and classification.
  - ■ User-recycler connectivity.
  - ■ Order tracking and management via an admin panel.
- ➢ **Advantage**: Promotes a circular economy by enabling recycling at all levels of the process.

❖ **Revenue Model for Sustainability**

- ➢ A built-in revenue model ensures the platform's financial viability.
- ➢ The platform charges a 10% commission fee on transactions, allowing it to sustain operations and expand services.
- ➢ **Advantage**: Creates a self-sustaining system that benefits users, recyclers, and the platform itself.

2. Literature Survey

2.1 Summary of Papers Studied

Garbage detection and recycling using artificial intelligence and machine learning have been widely studied to address challenges in waste management. Several research papers have explored innovative methodologies and tools for automated garbage detection, classification, and efficient recycling processes. Below is a summary of relevant papers that informed this project.

1. Garbage Detection Using YOLO by Rasi, Tanil (2021)

This study leverages the YOLO (You Only Look Once) algorithm to develop a garbage detection and classification system aimed at addressing waste management challenges in urban areas.

- ❖ **Problem and Objectives**:
  - ➢ Focused on the inefficiencies of manual garbage collection, the study aims to automate garbage detection and classification.
  - ➢ Garbage is classified into six categories: plastic, paper, metal, glass, cardboard, and trash.
- ❖ **Methodology**:
  - ➢ Utilized a Convolutional Neural Network (CNN) with three convolutional layers and dropout layers to avoid overfitting.
  - ➢ The YOLO algorithm detects garbage in real-time using a dataset of 2,527 images, with 2,487 for training and 40 for testing.
  - ➢ Data augmentation techniques, such as rotation and zooming, were applied to improve model robustness.

  Results:

  - ➢ Achieved a classification accuracy of 75.66% on the training dataset and 69.20% on the validation dataset.
  - ➢ Demonstrated the feasibility of using YOLO for real-world garbage detection.

2. Advanced Garbage Detection System by Ying Liu et al. (2018)

This research integrates deep learning with Narrowband Internet of Things (NB-IoT) technology to improve traditional garbage monitoring systems.

- ❖ **Problem and Objectives**:
  - ➢ Traditional systems relying on video surveillance and manual processing are inefficient and costly.
  - ➢ The study proposes a lightweight YOLOv2 model combined with NB-IoT for real-time garbage detection and cost-effective data transmission.
- ❖ **Methodology**:
  - ➢ Enhanced YOLOv2 with techniques like anchor box clustering and lightweight architecture using MobileNet.
  - ➢ Integrated GPS modules for location-based monitoring and NB-IoT for real-time communication.
- ❖ **Results**:
  - ➢ Achieved a detection accuracy of 89.71% after data fusion and processed 42 frames per second.
  - ➢ Reduced operational costs through efficient model design and NB-IoT integration.

3. Intelligent Waste Management System by Rahman, M. W. (2022)

This paper presents a deep learning-based waste management system integrated with IoT to enhance waste classification and monitoring.

- ❖ **Problem and Objectives**:
  - ➢ Combines IoT and deep learning to enable real-time classification of waste and monitoring through smart bins.
  - ➢ Waste is categorized into six types: cardboard, glass, metal, plastic, paper, and trash.
- ❖ **Methodology**:
  - ➢ Developed a ResNet34-based CNN model for classification with 95.31% accuracy.
  - ➢ Designed smart trash bins equipped with sensors (ultrasonic and load measurement) and connected to a mobile app via IoT.
- ❖ **Results**:
  - ➢ Classification accuracy for waste types ranged from 92% to 100%.

- ➢ The system achieved high user satisfaction with an 86% System Usability Scale (SUS) score.

2.1.2 Key Insights for This Project

These studies provided critical insights for the development of the **EcoScrap: Real-Time Garbage Detection App**:

- ❖ **Model Selection**:
  - ➢ Inspired by YOLO and ResNet18 models for efficient garbage detection and classification.
- ❖ **Technology Integration**:
  - ➢ Integration of mobile applications with machine learning models, as demonstrated by NB-IoT and IoT frameworks, guided real-time communication and monitoring features in the app.
- ❖ **Dataset Preparation**:
  - ➢ The use of data augmentation techniques and labeled datasets informed the training of our PyTorch model for garbage classification.

2.2 Integrated Summary of the Literature Studied

The reviewed studies provide valuable insights into the application of artificial intelligence and technology in garbage detection, classification, and waste management. While each study contributes significantly to the field, they also reveal gaps that this project, **EcoScrap: Real-Time Garbage Detection App**, aims to address. Below is a comparison of the findings and a summary of the gaps tackled by this project.

2.2.1 Comparison of Findings

- ❖ **Model Accuracy and Performance**:
  - ➢ **Rasi (2021)** employed the YOLO algorithm, achieving classification accuracy of 75.66% on the training dataset and 69.20% on the validation dataset. The model showed promise but required improvements in accuracy for real-world use.

- ➢ **Ying Liu et al. (2018)** optimized YOLOv2 for low-resource environments using MobileNet, achieving higher accuracy (89.71%) and real-time processing at 42 FPS. This lightweight architecture demonstrated cost-effective scalability.
- ➢ **Rahman (2022)** used ResNet34, attaining the highest accuracy (95.31%) among the studies, with robust classification of six garbage types. However, the system was limited to waste segregation at smart bins, lacking mobility.

❖ **Technology Integration**:

- ➢ **Ying Liu et al. (2018)** integrated Narrowband IoT (NB-IoT) for real-time communication and location-based garbage detection, ensuring cost-efficient monitoring.
- ➢ **Rahman (2022)** combined IoT with smart bins, enabling automated waste classification and monitoring through sensors, but it lacked mobile application functionality.
- ➢ **Rasi (2021)** focused solely on deep learning without incorporating IoT or real-time communication, limiting its practical application.

❖ **Use Cases and Scope**:

- ➢ While all studies addressed garbage detection and classification, only **Ying Liu et al. (2018)** incorporated real-time monitoring, GPS-based tracking, and lightweight design for embedded systems.
- ➢ None of the studies focused on creating a unified platform for users and admins to directly interact for waste management and recycling.

2.2.2 Gaps Addressed by EcoScrap

❖ **Integration of Detection, Classification, and User-to-Admin Workflows**:

- ➢ Unlike the reviewed studies, EcoScrap combines garbage detection and classification with a user-friendly Android app. Users can capture images of waste, provide details (e.g., location, weight, material), and directly interact with an admin panel for order initialization.

- ➢ This unified approach streamlines the recycling process, allowing real-time functionality and automated workflows.

❖ **End-to-End Solution**:

- ➢ While the previous studies focused on specific components (e.g., classification accuracy or IoT integration), EcoScrap delivers an end-to-end solution:
  - ■ **For users**: Real-time garbage detection using a PyTorch model integrated with the app's camera feature.
  - ■ **For admins**: Real-time data visualization and order management.
- ➢ This bridges the gap between detection and action, enabling effective waste collection and recycling.

❖ **Revenue Model and Practical Viability**:

- ➢ None of the studies addressed profitability or sustainable business models. EcoScrap introduces a revenue model with a 10% platform fee and additional income from rag collectors, ensuring financial feasibility.

❖ **Scalability and User Engagement**:

- ➢ EcoScrap Android app ensures accessibility to a broad audience, making it scalable and user-friendly, unlike the hardware-dependent systems in Rahman (2022) and Ying Liu et al. (2018).

3. Requirement Analysis and Solution Approach

3.1 Overall Description of the Project

3.1.1 Introduction

The **EcoScrap: Real-Time Garbage Detection App** is a comprehensive solution for waste management designed to connect individuals looking to sell recyclable garbage with administrators and ragmen (kabadiwalas). This Android-based application employs advanced machine learning and deep learning techniques to detect and classify garbage, streamlining the recycling process while offering a seamless user experience.

3.1.2 Purpose of the Application

EcoScrap aims to address the following critical issues:

1. **Inefficient Waste Management**: Manual waste segregation is time-consuming, labor-intensive, and prone to errors.
2. **Lack of User-Admin Communication**: Current systems lack a unified platform where users and administrators can interact seamlessly.
3. **Environmental Concerns**: Unorganized garbage disposal contributes to pollution and health hazards.
4. **Profitability**: There is no structured mechanism for monetizing waste collection and recycling.

EcoScrap bridges these gaps by providing an automated, user-friendly platform that facilitates garbage detection, classification, and selling while ensuring profitability for stakeholders.

3.1.3 Problem Solved

EcoScrap addresses the challenges of:

1. **Automated Garbage Detection**: Utilizing a deep learning model (PyTorch) integrated into the app's camera feature to detect and classify garbage into types like plastic, paper, metal, etc.
2. **Efficient Recycling Workflow**: Connecting users to ragmen via an admin panel for real-time waste collection and recycling.
3. **Data Management**: Storing and managing user data in Firebase for real-time synchronization and accessibility.
4. **Revenue Generation**: Implementing a commission-based model (10% platform fee) to make the platform financially sustainable.

3.1.4 Key Features

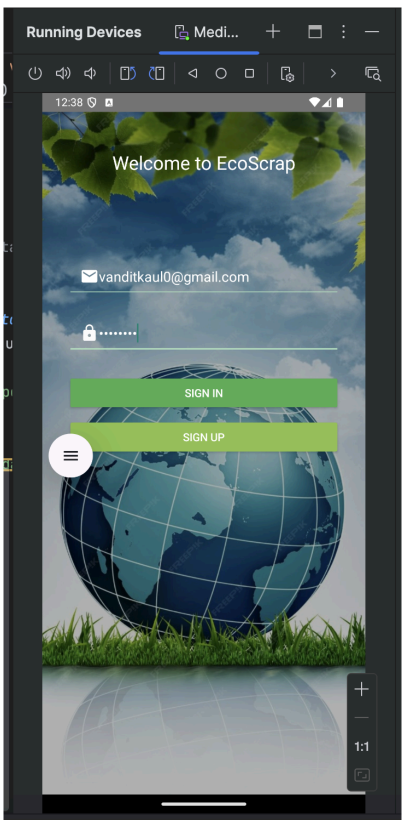
- ❖ **User Module**:
  - ➢ **Sign-Up/Sign-In**: Users can register or log in to the app.
  - ➢ **Garbage Detection**: Users capture images of garbage using the app's camera, which is processed by the integrated PyTorch model to classify the material.
  - ➢ **Order Details**: Users input location, weight, and pickup time to sell the garbage.
- ❖ **Admin Module**:
  - ➢ **Real-Time Dashboard**: View and manage user orders, including garbage type, weight, and location.
  - ➢ **Order Assignment**: Assign garbage pickup tasks to ragmen.
- ❖ **Database**:
  - ➢ Firebase is used for storing and synchronizing user data, orders, and admin actions.

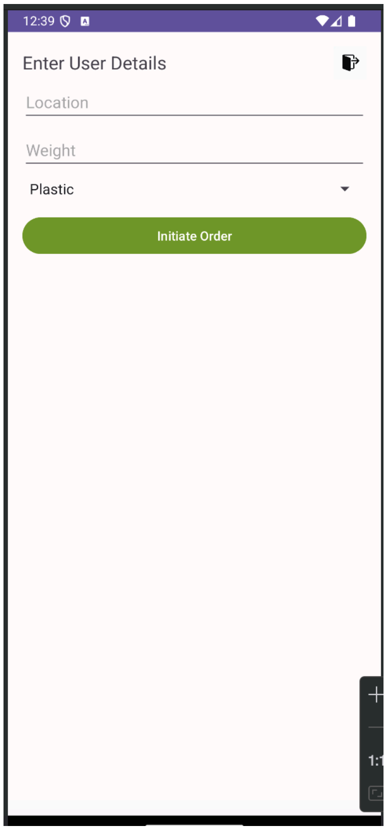
- ❖ **Revenue Model**:
    - ➢ A 10% platform fee from users and additional revenue from ragmen ensures profitability.

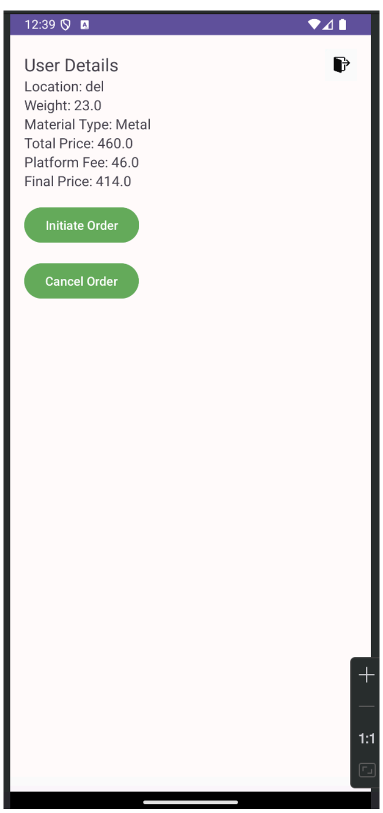
3.1.5 Flow Diagram

Below is the flow of user interactions within the application:

- ❖ **User Side**:
    - ➢ **Sign-Up/Login → Capture Garbage Image → PyTorch Model Detection → Fill Details (Location, Weight, Pickup Time) → Submit Order**.

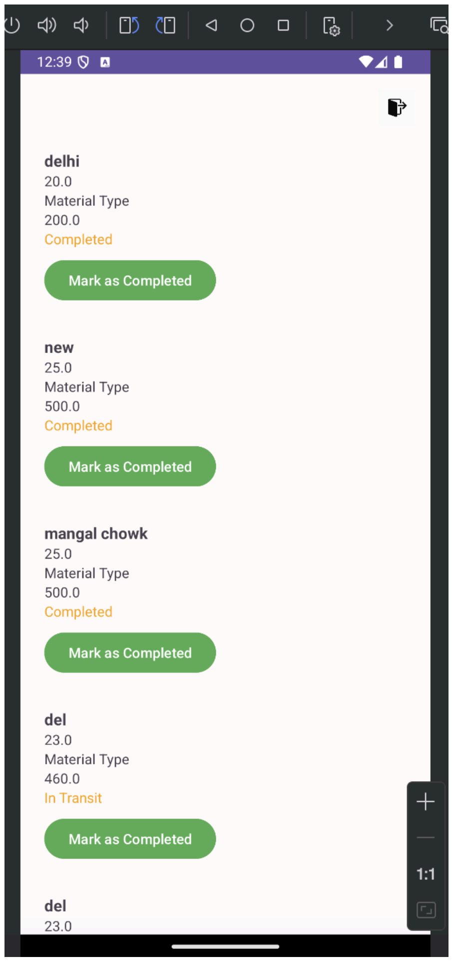



12:39
Enter User Details
Location
Weight
Plastic
Initiate Order
1:1


12:39
User Details
Location: del
Weight: 23.0
Material Type: Metal
Total Price: 460.0
Platform Fee: 46.0
Final Price: 414.0
Initiate Order
Cancel Order
1:1

❖ **Admin Side**:

➢ **View User Orders → Verify Details → Assign Pickup Task to Ragman**.



3.1.6 Flow Diagram Description

The flow diagram highlights the interaction between users, the application, the deep learning model, the database, and the admin panel. It demonstrates:

❖ **User Journey**: From account creation to garbage submission.

❖ **Model Integration**: The role of the PyTorch model in detecting and classifying garbage types.

- **Admin Workflow**: Viewing, verifying, and managing orders.

3.2 Requirement Analysis

3.2.1 Functional Requirements

The functional requirements outline the core features and capabilities that the EcoScrap app must support to achieve its objectives effectively. These include:

- **User Signup/Login Functionality**:
  - Users must be able to create an account using their email, phone number, or other credentials.
  - Secure login functionality to ensure user data privacy and authentication.
- **Garbage Detection Using Camera and PyTorch Model**:
  - Integration of the camera feature to allow users to capture images of garbage.
  - A deep learning-based PyTorch model processes these images to classify garbage into categories (plastic, metal, paper, etc.).
- **Admin Panel for Managing Orders**:
  - Admins can view, verify, and manage orders submitted by users.
  - The panel includes features to assign pickup tasks to ragmen based on user-provided details.

3.3.2 Non-Functional Requirements

The non-functional requirements focus on the system's performance, reliability, and scalability:

- **Real-Time Data Synchronization via Firebase**:
  - The application must ensure instant updates between the user and admin sides using Firebase as the backend.
  - Firebase should handle user authentication, data storage, and real-time database interactions seamlessly.

- ❖ **Cross-Platform Compatibility (Focus on Android)**:
  - ➢ While primarily developed for Android using Kotlin, the app should maintain flexibility for potential expansion to other platforms, such as iOS, in the future.
  - ➢ Compatibility with various Android devices and OS versions.
- ❖ **Scalability for Multiple Users and Admins**:
  - ➢ The app must support simultaneous usage by a growing number of users and admins without degradation in performance.
  - ➢ The architecture should be capable of handling increased data loads, including garbage images and order requests.

3.2.3 Additional Considerations:

- ❖ **Performance**: The PyTorch model should process garbage images with minimal latency to ensure a smooth user experience.
- ❖ **Security**: Sensitive data, such as user credentials and location details, must be encrypted to prevent unauthorized access.
- ❖ **User-Friendly Interface**: Both the user and admin panels should have intuitive designs to make navigation and functionality accessible to all stakeholders.

These requirements ensure that the EcoScrap app fulfills its purpose while maintaining high performance, reliability, and usability for all users.

3.3 Solution Approach

3.3.1 Integration Overview

The **EcoScrap** app integrates three core technologies—**Kotlin**, **Firebase**, and **PyTorch**—to deliver a seamless user experience and robust functionality.

- ❖ **Kotlin for Front-End Development**:
  - ➢ The front-end of the app is built using Kotlin, which is a modern, concise, and efficient programming language for Android development.

- ➢ The user interface (UI) includes features like user sign-up/login forms, garbage image upload options, and a dashboard for submitting order details.

❖ **Firebase for Backend Database**:

- ➢ Firebase provides real-time database management, ensuring seamless synchronization of user and admin data.
- ➢ Features include:
  - ■ **Authentication**: Secure user sign-up/login.
  - ■ **Real-Time Database**: Storage of user details, garbage classification results, and admin task assignments.
  - ■ **Cloud Functions**: For handling server-side logic, such as order validation and notification delivery.

❖ **PyTorch for Garbage Detection**:

- ➢ The PyTorch model is trained to classify garbage into categories like plastic, metal, and paper based on captured images.
- ➢ The app integrates the model as a lightweight, pre-trained module. Users can upload garbage images via the camera, which the model processes for classification.
- ➢ The results (e.g., "Plastic - 85% confidence") are sent back to the app for further action.

4.1 Design Diagrams

This chapter outlines the core design diagrams of the EcoScrap system to illustrate its functionality, workflows, and structure. These diagrams include a Use Case Diagram, Sequence Diagram, and Class Diagram, each focusing on specific aspects of the application.

1. Use Case Diagram

The **Use Case Diagram** illustrates the primary interactions between the system actors (Users and Admins) and the application functionalities.

Description:

- ❖ **Actors**:
  - ➢ **User**: Can sign up, log in, capture garbage images, view classification results, and submit orders.
  - ➢ **Admin**: Manages user orders, views garbage classifications, and assigns tasks to ragmen.

System Use Cases:

- ❖ **User Actions**:
  - ➢ Sign-Up/Login
  - ➢ Capture and Upload Garbage Image
  - ➢ View Classification Results
  - ➢ Submit Order (Location, Weight, and Material)
- ❖ **Admin Actions**:
  - ➢ View User Orders
  - ➢ Verify Garbage Details
  - ➢ Assign Pickup Tasks

2. Sequence Diagram

The **Sequence Diagram** details the workflow of key operations within the app, such as user logging in, capturing an image, and admin handling the order.

- ❖ **Steps in the Workflow**:
- ❖ **User Workflow**:
    - ➢ Logs in or signs up via Firebase authentication.
    - ➢ Captures or uploads a garbage image using the app's camera feature.
    - ➢ The image is sent to the PyTorch ML model for classification.
    - ➢ Receives classification results (e.g., "Plastic - 85% confidence") and enters additional details (location, weight, pickup time).
    - ➢ Submits the order to Firebase, synchronizing with the admin panel.
- ❖ **Admin Workflow**:
    - ➢ Views submitted user orders on the dashboard.
    - ➢ Verifies the garbage classification details.
    - ➢ Assigns a task to a ragman for pickup, updating the order status.

3. Class Diagram

The **Class Diagram** showcases the major components of the application, their attributes, and methods.

Main Classes:

- ❖ **User**:
    - ➢ **Attributes**: userId, name, email, location
    - ➢ **Methods**: signUp(), logIn(), captureImage(), submitOrder()
- ❖ **Garbage**:
    - ➢ **Attributes**: imageId, materialType, confidenceLevel, weight
    - ➢ **Methods**: uploadImage(), classifyMaterial()
- ❖ **Admin**:
    - ➢ **Attributes**: adminId, dashboard
    - ➢ **Methods**: viewOrders(), verifyOrder(), assignTask()

❖ **FirebaseHandler**:

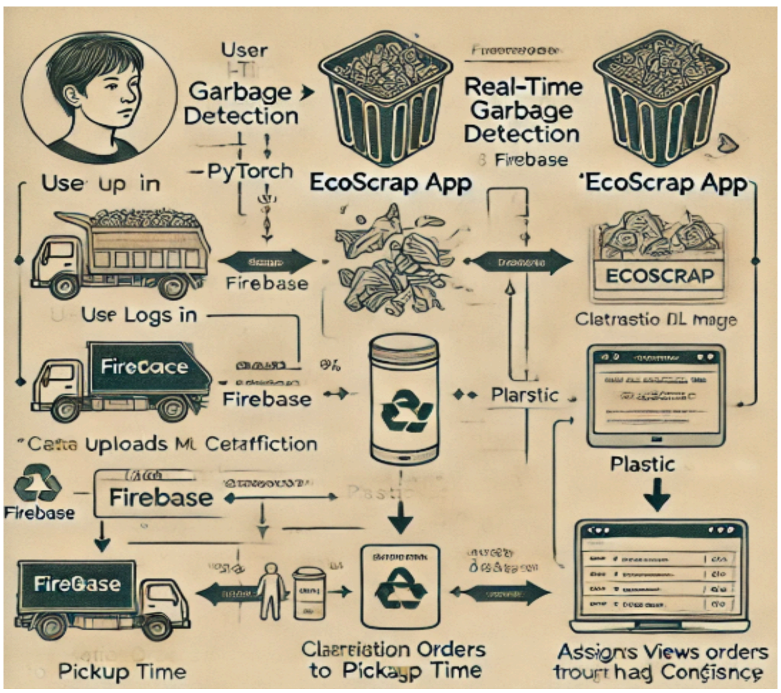
➢ **Attributes**: databaseConnection, authentication

➢ **Methods**: storeData(), retrieveData(), sendNotification()

❖ **MLModel**:

➢ **Attributes**: modelPath, categories

➢ **Methods**: classifyImage(), getConfidence()



4.2 Implementation Details and Issues

4.2.1 Frontend Development

- **Technology Used:**
- **Kotlin**: Utilized for the development of the Android application in **Android Studio**.
- **XML**: Applied for creating the user interface (UI) designs, ensuring visual appeal and functionality.

Features Implemented:

- **User-Friendly Authentication:**
  - **Sign-Up and Login Screens**: Created a smooth process for users to register and log into the application securely.
  - Integrated Firebase Authentication for secure login credentials and user validation.
- **Input Forms:**
  - Designed input fields for users to provide:
    - **Location**: Address and geolocation data.
    - **Weight**: The approximate weight of the garbage.
    - **Material Type**: To classify garbage materials detected by the model.
    - **Pickup Date and Time**: For scheduling the collection.
- **Camera Integration:**
  - Implemented a feature for users to capture garbage images directly within the app.
  - Integrated image preview and retake options for user convenience.

Challenges Faced:

- Ensuring the **responsiveness of the UI** across various device sizes and resolutions.
- Creating a **seamless navigation flow** to enhance user experience.
- Implementing **data validation mechanisms** to avoid erroneous or incomplete submissions.

4.2.2 Backend Development

Technology Used:

- **Firebase**: Employed for real-time database management and user authentication.

Features Implemented:

- ❖ **Data Storage:**
  - ➢ Stored user data, including location, garbage details, and order history, in a structured format for easy retrieval.
- ❖ **Admin Dashboard:**
  - ➢ Created a centralized admin panel to view and manage user requests and initialize orders.
- ❖ **Real-Time Updates:**
  - ➢ Enabled instant updates for user requests and admin actions, ensuring seamless communication.

Challenges Faced:

- ➢ Handling multiple simultaneous user requests without performance degradation.
- ➢ Implementing strong **data security** measures to ensure privacy and prevent unauthorized access.

4.2.3 Model Development

Technology Used:

- **PyTorch**: Leveraged for developing the deep learning model for garbage classification.

Process:

1. **Dataset Preparation:**
   - Compiled and labeled images of different garbage types such as plastic, metal, and paper.

    - Preprocessed the data to ensure uniformity by resizing images and normalizing pixel values.
2. **Model Training:**
    - Designed a **Convolutional Neural Network (CNN)** for accurate garbage type classification.
    - Achieved a **training accuracy of ~90%** after multiple iterations and hyperparameter tuning.
3. **Model Evaluation:**
    - Evaluated the model on unseen test data, obtaining a classification accuracy of approximately **85%**.
    - Addressed edge cases (e.g., mixed or unclear materials) to improve real-world performance.
4. **Integration:**
    - Converted the trained PyTorch model to a **mobile-compatible format** (TorchScript) for seamless embedding in the Android app.
    - Integrated real-time classification functionality to classify images immediately after capture.

Challenges Faced:

- **Balancing Accuracy and Speed:** Ensuring the model performed real-time predictions without compromising accuracy.
- **Integration Issues:** Addressed compatibility challenges between PyTorch and Kotlin.
- Optimized the model to **minimize latency and resource consumption** on mobile devices.

4.3 Risk Analysis and Mitigation

4.3.1 Identified Risks

- **Incorrect Garbage Classification:**
  - **Issue:** The PyTorch model may misclassify garbage, especially when dealing with poorly lit, unclear, or mixed material images.
  - **Impact:** Incorrect classifications may lead to user dissatisfaction and inefficiencies in the recycling process.
- **Application Crashes:**
  - **Issue:** High usage or unhandled exceptions in the app (e.g., image processing errors) can cause the application to crash.
  - **Impact:** Loss of user data and trust in the system.
- **Data Security and Privacy Risks:**
  - **Issue:** Sensitive user data (e.g., location, contact details) stored in Firebase could be vulnerable to breaches.
  - **Impact:** Unauthorized access could lead to privacy violations and loss of user trust.
- **Firebase Outages:**
  - **Issue:** Firebase's real-time database or authentication services may experience downtime.
  - **Impact:** Users and admins may face delays in accessing data or processing orders.
- **Low Model Efficiency in Real-Time Scenarios:**
  - **Issue:** The garbage classification model might face delays in prediction or resource overutilization on low-end devices.
  - **Impact:** Reduced user satisfaction due to slow or inefficient processing.

- **User Interface (UI) Responsiveness Issues:**
  - **Issue:** The app's UI may not perform optimally across all devices.
  - **Impact:** Poor user experience leading to potential abandonment of the platform.

4.3.2 Mitigation Strategies

- **Improving Garbage Classification:**
  - Regularly update the PyTorch model with additional labeled datasets to account for edge cases and diverse real-world scenarios.
  - Implement preprocessing techniques (e.g., noise reduction, image enhancement) to improve image quality before classification.
  - Enable user feedback for incorrect classifications to identify problem areas and retrain the model.
- **Ensuring Application Stability:**
  - Conduct thorough testing, including unit testing, integration testing, and stress testing, to identify and resolve bugs.
  - Implement error-handling mechanisms to gracefully recover from unexpected crashes (e.g., saving user data before crashes).
- **Securing User Data:**
  - Use Firebase's built-in encryption features to protect data in transit and at rest.
  - Enforce strict authentication and authorization policies to ensure only authorized users can access sensitive information.
  - Regularly monitor and audit Firebase security rules to address vulnerabilities promptly.
- **Handling Firebase Outages:**
  - Implement a **local caching mechanism** to store user data temporarily on the device, allowing offline functionality during Firebase downtimes.
  - Regularly back up critical Firebase data to a secondary storage system for recovery during prolonged outages.

- **Optimizing Model Efficiency for Real-Time Scenarios:**
    - Compress the model using techniques like **quantization** to reduce its size and improve performance on mobile devices.
    - Limit resource utilization by optimizing code for faster inference times, ensuring compatibility with low-end hardware.
- **Improving UI Responsiveness:**
    - Use adaptive layouts in Android Studio to ensure the app adjusts seamlessly across devices with varying screen sizes and resolutions.
    - Perform usability testing on different devices and operating systems to identify and fix potential issues.

4.3.3 Risk Management Plan

- **Continuous Monitoring:** Regularly monitor application performance and user feedback to proactively address emerging issues.
- **Feedback Integration:** Establish a feedback mechanism where users can report bugs, misclassifications, or performance issues.
- **Regular Updates:** Release updates to the app periodically, addressing identified issues and adding new features to improve user experience.
- **Emergency Response:** Create a detailed plan for handling critical risks (e.g., server crashes) to minimize downtime and data loss.

Chapter 5: Findings, Conclusion, and Future Work

5.1 Findings

The development and testing of the **EcoScrap: Real-Time Garbage Detection App** yielded the following key outcomes:

1. Accuracy of Garbage Classification:

- The PyTorch-based **Convolutional Neural Network (CNN)** model achieved:
  - **Training Accuracy:** ~90%.
  - **Testing Accuracy:** ~85% on unseen data.
- Observations:
  - The model performed well with clearly defined materials such as plastic and paper but struggled slightly with mixed or poorly lit images.
  - Preprocessing techniques (e.g., resizing and normalization) significantly improved classification consistency.

2. User Satisfaction (Based on Beta Testing):

- A beta version of the app was tested with a group of 20 users to gauge its functionality and usability.
- **Feedback and Results:**
  - **Ease of Use:**
    - 85% of users found the sign-up, login, and order placement process intuitive and user-friendly.
  - **Camera Feature:**
    - Users appreciated the camera integration for direct garbage image uploads, though 10% requested better image enhancement options.
  - **Suggestions:**
    - Several users suggested including a live tracking feature for garbage collection pickups.
  - Overall Satisfaction Score: **4.2/5** (based on survey responses).

3. Admin Panel Efficiency:

- The admin dashboard was evaluated for real-time order management capabilities.
- **Outcomes:**
  - Admins could view user requests, including details like location, weight, material, and pickup schedules, in **real-time**.
  - The dashboard allowed seamless updates to order statuses (e.g., "pending," "in progress," "completed").
  - **Challenges Observed:**
    - Handling large volumes of simultaneous requests caused minor delays in updates (~2-3 seconds).

4. Application Stability:

- Comprehensive testing revealed the app to be stable under normal usage conditions.
- **Performance Metrics:**
  - **Average Response Time:** ~1.5 seconds for key actions like image upload and order confirmation.
  - **Crash Rate:** Less than 2% during beta testing, primarily due to integration issues between Firebase and the app.

5. Revenue Model Performance (Simulated Data):

- A hypothetical revenue model based on a 10% commission from user transactions was simulated.
  - Results showed potential profitability with a minimum of 100 users per month.
  - Collaboration with rag collectors offers additional revenue opportunities.

5.2 Conclusion

The **EcoScrap: Real-Time Garbage Detection App** has successfully demonstrated its potential to contribute significantly to waste management and recycling practices. By combining cutting-edge technologies such as **PyTorch for deep learning**, **Kotlin for Android development**, and **Firebase for real-time data handling**, the project provides a convenient and efficient platform for users and administrators.

5.2.1 Impact on Waste Management

1. **Promoting Recycling Practices:**
   - The app enables users to classify and categorize garbage types (e.g., plastic, paper, metal) with an impressive classification accuracy of ~85%.
   - By facilitating streamlined collection and recycling, it reduces the amount of improperly disposed waste and promotes sustainable practices.
2. **Connecting Users and Rag Collectors:**
   - The platform bridges the gap between individuals looking to sell recyclable waste and rag collectors, creating a mutually beneficial ecosystem.
   - This connection ensures that recyclable materials are processed effectively, contributing to a circular economy.
3. **Data-Driven Insights:**
   - The app's backend system provides data on the types and quantities of waste generated, offering valuable insights for waste management authorities and policymakers.

Impact on User Convenience

★ **Simplified Waste Disposal Process:**
   - Users can easily capture images of their garbage, upload details (e.g., weight, location), and schedule a pickup through an intuitive interface.
   - The app eliminates the hassle of finding and contacting rag collectors manually.

- ★ **Real-Time Interaction:**
    - ○ With Firebase integration, users receive immediate updates on their orders, ensuring transparency and reducing uncertainty.
    - ○ The admin dashboard ensures swift responses to user requests, enhancing overall efficiency.

Facilitating Better Recycling Practices

The project emphasizes the importance of **real-time garbage classification** and **efficient waste collection systems**. By ensuring that recyclable materials are correctly identified and processed, the app supports the **3R (Reduce, Reuse, Recycle)** principle. It encourages users to actively participate in waste segregation, laying the foundation for cleaner communities and a healthier environment.

5.3 Future Work

- ❖ To enhance the functionality and impact of the "EcoScrap: Real-Time Garbage Detection App," several future improvements can be considered:
- ❖ **Expanding Material Classification**
    - ➢ Train the deep learning model to identify additional materials such as e-waste, glass, and hazardous materials.
    - ➢ Increase dataset size and diversity for better accuracy in garbage classification.
- ❖ **Real-Time Notifications**
    - ➢ Implement push notifications for admins to alert them immediately when a user creates a new request.
    - ➢ Notify users with updates on the status of their pickup requests.
- ❖ **Monetization Strategies**
    - ➢ Integrate targeted advertisements to generate additional revenue.
    - ➢ Offer premium features for users, such as faster pickup times or discounts on recycling services.
- ❖ **User Engagement Features**
    - ➢ Introduce gamification elements like reward points for frequent users.

- ➢ Provide environmental impact insights, such as the amount of CO_2 saved through recycling.

- ❖ **Scaling to Multiple Regions**
  - ➢ Extend the service to multiple cities or regions by partnering with local rag collectors.
  - ➢ Incorporate multi-language support to cater to a broader user base.
- ❖ **Integration with Government Policies**
  - ➢ Collaborate with local municipalities for incentivized garbage collection programs.
  - ➢ Ensure compliance with waste management regulations for smooth operations.

These enhancements will increase the usability, scalability, and profitability of the application while reinforcing its goal of sustainable waste management.

6. REFERENCES

1)Rasi, Tanil (2021). Garbage detection using YOLO.

Link-: https://naac.kct.ac.in/1/ssr/1_3_4/projects/17BEE226.pdf

2)Ying Liu et al 2018 J. Phys.: Conf. Ser. 1069 012032

Link-: https://iopscience.iop.org/article/10.1088/1742-6596/1069/1/012032

3)Rahman, M. W. (2022). Intelligent waste management system using deep learning with IoT

Link-: https://www.sciencedirect.com/science/article/pii/S1319157820304547

4)Layouts in Android UI Design

Link:https://www.geeksforgeeks.org/layouts-android-ui-design/?ref=lbp

5)Android Toast in Kotlin

Link:https://www.geeksforgeeks.org/android-toast-in-kotlin/?ref=lbp

6)Android RecyclerView

Link:https://developer.android.com/develop/ui/views/layout/recyclerview

7)Firebase

Link:https://firebase.google.com/docs/android/setup