BinHero: An AI-Driven Smart Waste Disposal Mobile App with Object Detection Motion Tracking

A report submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor
of
Computer Science (Hons)

by

ANUJ BHANDARI
0358445

COMPUTER SCIENCE
INSTITUTE OF INTERNATIONAL MANAGEMENT SCIENCE (IIMS)
AFFILIATED TO TAYLOR'S UNIVERSITY, MALAYSIA
Kathmandu, Nepal

April, 2025

CERTIFICATE

This is to certify that **Anuj Bhandari**, Bachelor (Computer Science), has worked on the project entitled **'BinHero: An AI-Driven Smart Waste Disposal Mobile App with Object Detection Motion Tracking '** under my supervision and guidance. The content of this report is original and has not been submitted elsewhere for the award of any academic or professional degree.

April, 2025
IIMS College, Kathmandu

Sachin sir
Lecturer, Department of Computer Science
IIMS College, Kathmandu

ACADEMIC INTEGRITY

I hereby declare that this capstone project is my own work and, to the best of my knowledge, it contains no materials previously published or written by any other person, or substantial proportions of material which have been accepted for the award of any other degree at IIMS College or any other educational institution, except where due acknowledgement is made in the project.

April, 2025
IIMS College

Anuj Bhandari

ACKNOWLEDGEMENT

I would like to sincerely thank Mr. Sachin Shrestha Sir, who has guided me beyond measure, motivated me, and provided me with useful criticism during the progression of this project. I owe the faculty of IIMS College my utmost gratitude in terms of their technical support and informed learning environment that enabled this project. Special thanks to all my team members, who contributed their efforts and support to the project on a two dimensional front in the backend development, the front end, UI/UX design, and the quality assurance front of the project. Lastly, I would also like to show my gratitude to my family and friends, the unconditional support and assistance of which have been invaluable to me throughout this process.

Anuj Bhandari

ABSTRACT

BinHero is a multiplatform mobile application that was conceptualized to address some of the current challenges in waste management by embracing artificial intelligence and gamification. The system combines YOLO11 to detect objects in real-time and Deep-SORT/SORT to track objects, and therefore ensures that waste disposal can be accurately classified and verified. The architecture integrates hybrid AI inference, which allows for on-device and on-server computation, hence keeping the flexibility and responsiveness to a maximum degree. Using React Native, MongoDB, Node.js, and Flask, the application transforms waste disposal into an intelligent and engaging process, all while gathering data on garbage bin locations that can guide community planning. The project is an illustration of the synergistic usage of AI, civic engagement, and behavioral science to create a smart, scalable process of waste disposal.

Contents

List of Symbols & Abbreviations

AI	*Artifical intelligence*
API	*Application Programming Interface*
CNN	*Convolutional Neural Network*
mAP	Mean Average Precision
mAP	*Mean Average Precision*
ML	*Machine learning*
PvP	*Player vs Player*
R-CNN	*Regions with Convolutional Neural Network*
Re-ID	*Re-identification*
TfLite	*TensorFlow Lite*
YOLO	*You Only Look Once*

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Executive Summary

Rapid urbanization and city growth have caused a significant increase in worldwide waste. By 2050, this production is anticipated to increase by 70%, per Kaza et al. (2018). Waste management techniques in the past had certain issues that made it challenging for both people and communities to participate and recycle properly. In order to confront this multifaceted challenge, the application presents a novel approach and tackles the issue using crowdsourcing, gamification, and artificial intelligence as features within a single mobile application.

BinHero is a mobile application that is powered by AI technologies like object detection and motion tracking. These AI technologies help validate trash disposal actions in real-time, while also promoting accurate sorting behavior with fast feedback. Point systems, leaderboards, PvP challenges, and team-based tasks, such as gamified elements, further enhance user engagement and develop long-term habits. The application aims to turn a garbage disposal into a community-driven activity by incorporating those AI features and enabling users to map and verify bin sites through crowdsourcing. This boosts accessibility and aids municipal planning efforts. The app was built using React Native, with backend support from Node.js and MongoDB. Likewise, YOLO11 and DeepSORT algorithms were applied to detect objects and track their motion.

A thorough risk assessment and development roadmap guarantee that challenges from AI accuracy and hardware compatibility to user retention are addressed proactively through continual model refinement, strategic outreach, and inclusive design. Likewise, future improvements will be incorporated, like augmented reality integration, federated learning, multilingual support, and collaboration with government and environmental groups.

In conclusion, the project illustrates some of the ways technology and behavioral science may work together to provide environmental benefits, encourage community involvement, and build a culture of sustainability.

1.2 Project Objectives

The mobile application aims to revolutionize waste management methods by powering them with AI and gamification. Some of the primary objectives are as follows:

- AI for Real-Time Waste Classification and Validation: Make use of AI to track both objects and their movements to confirm proper waste disposal and help address mistakes in sorting and waste added to landfills.
- Promote Community Participation in Sustainability Efforts: Collaboration with environmentally conscious communities by offering team-based challenges, missions, and PvP (Player vs Player) arenas.
- Scalable and Engaging Mobile Platform: The Project aims to provide a robust backend that can handle a high load of traffic and will perform in harsh conditions.
- Data Driven: Locations of dustbins from crowdsourcing can be beneficial for the local government to plan and increase the efficiency of garbage collection.

1.3 Project Description

The project utilizes AI and user-driven engagement and encourages long-lasting environmental change, emerging as an effective solution to the ongoing global problem of inappropriate garbage disposal. **Key Components and Features:**

- Gamification Mechanics:
  - Garbage Disposal
  - Daily and weekly missions.
  - PvP contests.
  - Team challenges.
  - Leaderboards.
  - Badges and achievements.
- AI-powered Object detection and Motion tracking:
  - Custom-trained YOLO11 model to classify wastes as recyclable, biodegradable, or general.
- Crowdsourced Bin Mapping
  - Users pin the location of trash bins to a live map
  - Help the local government collect garbage efficiently and promote public participation.

- Backend Infrastructure
  - A minimal and flexible Node.js framework for server-side operations.
  - MongoDB is a NoSQL database to manage user profiles, bin data, and other interactions.
  - AI models built using the Pytorch framework with cloud-based inference.
- Platform and Accessibility
  - Built with React Native with usability and performance in mind, and optimized for contemporary devices.

1.4 Project Scope

The functional boundaries, features that the team intends to create and execute, are described in the application's project scope. In order to guarantee clarity of purpose and compatibility with technical requirements. Here, both projects' inclusions and exclusions are described.

Inclusion:

1. Mobile app development: Build a full-fledged mobile application from React Native with robust backend support that can provide users with seamless accessibility and functionality.
2. Gamification Rewards Systems: To engage users better, tools like structured points, rankings, battles between players, and challenges in teams are created and placed in the applications.
3. Crowd-source: It helps residents contribute to their community and plan the use of municipal resources by letting them update and check bin locations on the app.

Exclusion

1. Integration with Government Waste Collection Systems: At present, there has been no official integration with the municipal garbage collection or the government, despite the fact that the system might support data that is beneficial to municipal organizations.
2. Hardware Device: The project also does not entail the creation of physical hardware, such as sensor-based disposal devices.
3. Offline Functionality: Features such as real-time data sync, multiplayer interactions these all require internet connectivity, so the app does not have any offline features.

1.5 Project Organization

The project consists of an organizational structure that guarantees effective teamwork and accountability, which is essential for a successful execution of the project. The team comprises five members who contribute to their own specific domains and their areas of expertise.

Group Structure: Under the direction of a group leader, every member leads or contributes to an area of their expertise and works closely with others in a flat but functional organization.

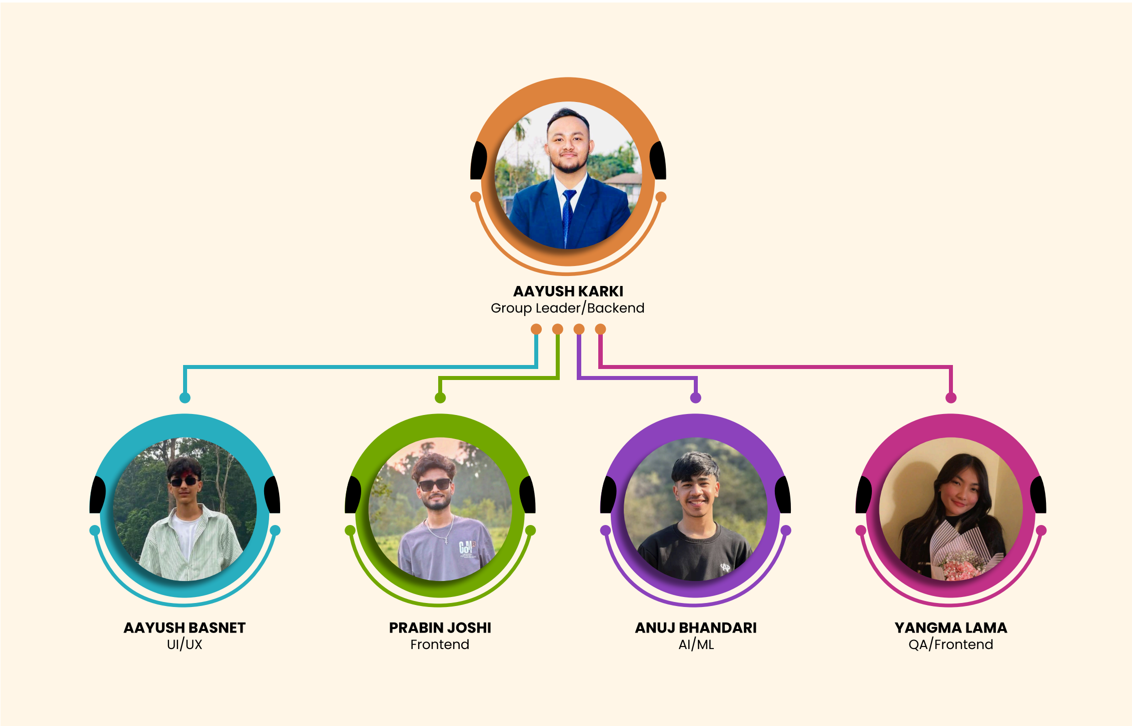

Figure 1.1: Organization Chart

The project organization involves the following roles:

Group Leader: The Group Leader directs the workflow management activities and works to build teamwork among the members. The group leader assigns responsibilities to team members while maintaining constant communication with the supervisor.

Group Members: Each member of the team brings various abilities to the group and is dedicated to a certain component of the project. In order to accomplish job efficiency and system completion, users work together as a team.

1.6 Milestones and Deliverables

Our application's development life cycle has been divided into several phases. It outlines all those critical stages and milestones that are required to successfully complete the application's development. The sole purpose of these plans is to track progress, allocate

Table 1.1: Roles and Responsibilities

Name	Roles	Responsibility
Anuj Bhandari	AI/ML	Development of Object detection model and collaboration with Backend to integrate models.
Aayush Karki	Backend Development	Work on the server-side logic and database connection.
Aayush Basnet	UI/UX	Design an aesthetically pleasing and user-friendly interfaces
Prabin Joshi	Frontend Development	Converts UI/UX designs and implements responsive, interactive interfaces of mobile applications.
Yangma Lama	QA/Frontend Development	Carries out testing and guarantees the functionality of the software while also assisting frontend development.

resources, and on on-time delivery. The significant milestones and anticipated timescale are shown in Table 2.1.

The project timings and key achievement objectives in all tasks are represented in the Figure. The Gantt chart helps to monitor the process in several stages, which include requirement analysis, design development, testing, and deployment. It ranges from March to December. Team members are highlighted with color bars to represent the expected time for their task; it encourages responsibility and cooperation.

Table 1.2: Milestones and Deliverables

No	Milestone	Expected time (*Days*)
1	Topic Research	10
2	Gathering and Analyzing Requirements	10
3	Project Proposal	10
4	UI/UX Design	45
5	Backend Development	55
6	AI model Development	35
7	Frontend Development	55
8	Gamification Implementation	25
9	CroudSource Bin Location	12
10	Testing	15
11	Documentation and Deployment	20

Chapter 2

Related Works

One of the most pressing environmental issues currently is the worldwide waste management catastrophe. According to Kaza et al. (2018), approximately 2.01 billion tons of municipal solid waste is produced annually. Despite increasing environmental awareness, the public has shown inadequate responses to conventional waste management education and infrastructure improvements.

2.0.1 AI Powered Waste Classification using Deep Learning

Artificial intelligence has been widely used in waste management in recent years due to the growing demand for accurate and efficient techniques to handle urban garbage. Object detection and classification are essential components of contemporary waste management systems. Arishi (2025) applied YOLOv8 on her paper to accurately classify household waste. With a mean average precision (mAP) of 89.5%, the YOLOv8-CBAM model outperformed the baseline model by a significant 4.2%. The paper implemented comparative analysis of other state of the art models such as YOLOv7, EfficientDet, Fast R-CNN and MaskR-CNN .In terms of performance and accuracy YOLOv8 was able to outperform all of them. The paper focused on the lightweight networks practical and real-time applications on consumer devices.

Kuang et al. (2024) study achieved a success rate accuracy of 89.5% as well. The paper addressed the problem of ineffective sorting and classification of waste still remains an inefficient waste management dimension in terms of time and cost consumption in the entire process. The research also identified the potentials as well as the limitations that faced YOLO in practical contexts, where it was sensitive to complicated backgrounds and overtopped objects. Nevertheless, the model demonstrated a high accuracy rate of classification in controlled conditions despite such challenges.

Nafiz et al. (2023) introduced Convowaste, a deep learning-driven system of trash segregation. It applied transfer learning with Inception-ResNet V2 to just label garbage into six categories with 98 percent of accuracy. The accuracy of the model on the available limited labeled data was astounding, which reflects on the success of pre-trained convolutional architectures in ensuring accurate classification in practical settings. Their approach, also, exhibited fast inference which is crucial to low-data real-time systems.

Sayem et al. (2025) suggested strong waste classification and detection models that allow efficient sorting and recycling of waste. The paper presented a with a dataset called WaRP which consists a collection of realistic conditions of waste types (28 in total) with various low light conditions and occlusion. The dataset was trained on GELAN-E detector which obtained 63% mAP@50 and on their dual-stream DenseViT model, which obtained 83.1% accuracy, outperforming the results of the prior datasets like TrashNet. This showed the influence of environment specific data on the model generalization.

Alharbi et al. (2025) addressed authors consider the burning issue of garbage in streets and the lack of efficient waste disposal in the world and proposed a solution called SAWN (Surveillance and Waste Notification) system. An innovatice approach in which real-time video classification and object detection are combined to detect and address cases of littering. Authors emphasized with dataset constraints in relation to garbage detection and public surveillance. Despite understanding that there is a need to personalize datasets with contextual behavioral labels, i.e., the type of littering (throwing, dropping, discarding), in combination with object instances (trash bags, bottles), their model was designed to use multiple detection models (YOLOv8, MoViNet, and Haar Cascades). The SAWN system illustrated high accuracy, it achieved 99.5% accuracy score with MoViNet and 99.5% accuracy score with YOLOv8 to detect license plate.

Besides limitations of datasets, uncontrolled conditions are an essential issue regarding robustness. Abo-Zahhad and Abo-Zahhad (2025) remedied this by cross-comparative analysis of YOLOv5 and YOLOv8 to identify the overflow of bins through the presence and absence of light, occlusions, and clutters. 95% and 96% accuracy were attained in their models respectively. Feasibility of running those models on low-resource and mobile hardware was also confirmed by the study, which means that the models can be used in the field in real-time fashion. The results encourage the foundational application of adaptive models that are well generalizable to the variability of the environmental real world.

2.0.2 Lightweight and Edge-Optimized Models

The core of project's technical orientation is dealing with real-time inference on devices with efficiency.Nedjar et al. (2024) utilized lightweight models like MobileNet and NASNet-Mobile, on systems like real-time solid waste sorter. The model was able to achieve nearly 100% classification accuracy and operated on Raspberry Pi with TensorFlow lite for efficient edge deployment. Likewise, Prasath (2025) verified that YOLO are appropriate for embedded systems by testing YOLOv5 on Jetson Nano and Raspberry Pi. It balanced accuracy and processing performance. Additionally, IoT-based robotic arm was integrated into the system to physically sort into respective bins. Their system was scalable, cost effective and deployed as a framework of AI-based trash management specifically for smart cities and industrial automation in India

White et al. (2020) implemented deep convolutional neural network tailored to edge devices. WasteNet deploys its model on jetson Nano since it is low-resource and has successfully achieved 97% accuracy. The research paper aimed to enhance waste sorting efficiency by using an automated, real-time waste sorting system, which would help solve the problem of ever-expanding loads of garbage and increasing contamination of recycled commodities. The paper also explored the AI-based approach to waste management through the implementation of object detection solution that can be used to categorize the trash. In this regard, pretrained on ImageNet DenseNet model was implemented. The model was evaluated on TrashNet dataset and achieved an accuracy score of 97%.

2.0.3 Motion Tracking and Disposal Verification

Kim and Cho (2022) proposed AIDM-strat which aimed at reducing the illegal garbage dumping, which is of significant concern in the Republic of Korea and is made even worse by the lack of municipal waste management system based on the volume. AIDM-strat implemented a combination of YOLOv4 and OpenPose to track wrist-to-object distances during disposal. This approach allows for accurate behavioral analysis in situations that involve illegal dumping. In order to confirm dumping, the system detects the trash and the distance between the hand and the trash, making it possible to detect behaviors like deliberate dumping. It was trained on GPU and scored a high mAP of 99.38% for trash bag classification. AIDM-Strat offered a more intelligent and precise solution to the problem of urban garbage, since it was able to evaluate the location of an object, in addition to identifying it, to allow a groundbreaking mechanism to recognize behavior depending on the context.

2.0.4 Integrated Detection and Tracking Frameworks

Huang et al. (2023) proposed a study on a real-time automotive sensing system and designed the solution to monitor urban garbage disposal data.The system consists the integration of object detection (YOLOv5) with tracking algorithms like DeepSORT could be used to monitor and analyze waste related activities across time. with the use of MQTT protocols for real-time data handling the system, which was installed in garbage trucks achieved 16.52 FPS and low latency communication (76ms). Similarly, Pathak et al. (2024) has used a multi component AI system that included Tesseract OCR for license plate recognition, and YOLOv5 for object detection. It could identify the amounts of unlawful dumping with the detection rate of 97 percent. The system is an illustration of a comprehensive surveillance system that can identify complex environmental. The article has also discussed dataset construction and preparation in detail in order to enhance replicability and resilience in the dataset.

2.0.5 Comparative Insights

A comparison of previous AI/ML models for smart trash management uncovers few important patterns and trade-offs that would influence the BinHero system's architecture. Among the reviewed object detection algorithm YOLO models were outstanding in real-time classification of garbage. Arishi (2025) and Abo-Zahhad and Abo-Zahhad (2025) record its greater precision and its applicability to edge deployment, thus validating the choice of mobile application to run YOLO11 on its servers and YOLO11n for mobile inference.
DeepSORT provides large performance advantages in terms of re-identification but requires a graphics processing unit (GPU) as opposed to SORT, which is fast and favors tracking on mobile with less opportunity to handle occlusion. In practice, the selective usage of both systems is used in the project where they are selected depending on the requirements of the task in hand.PyTorch allows significant flexibility in model building but adds more complexities to frameworks deployed in resource-constrained platforms.

In conclusion, dual-inference and hybrid tracking strategy integrates the most important aspects of previous research, while balancing accuracy, performance, and platform compatibility.framework

Table 2.1: Summary of Related Works

Study	Problem Addressed	AI Approach	Deployment Approach	Key Result	Relevance to BinHero
Arishi (2025)	accurate classification of household garbage	YOLOv8-CBAM	Consumer devices	89.5% mAP	Ensures the application of YOLO on-device real-time classification
Kuang et al. (2024)	YOLO performance in practical settings	YOLOv8	Controlled vs real environments	High accuracy in clean settings	Endorses the real-time applicability of YOLO
White et al. (2020)	On-device waste classification	WasteNet (CNN)	Jetson Nano	97% accuracy	Evidence of low- resource edge deployment
Nafiz et al. (2023)	Multi-category trash classification	Inception-ResNet V2	Transfer learning on limited data	98% accuracy	Confirms pre-trained CNNs on small datasets
Nedjar et al. (2024)	Light models for sorting on embedded systems	MobileNet, NASNet-Mobile	Raspberry Pi with TFLite	Nearly 100% accuracy	Validates lightweight models for embedded inference
Prasath (2025)	Real-time YOLO use in embedded systems	YOLOv5	Jetson Nano	Balanced accuracy-performance	Affirms the appropriateness of YOLO to smartphones
Kim and Cho (2022)	Behavior analysis in illegal dumping	YOLOv4 + OpenPose	GPU system	99.38% mAP for trash bags	Inspires motion + classification strategy in BinHero
Huang et al. (2023)	Waste monitoring over time	YOLOv5 + DeepSORT	Garbage trucks + MQTT	16.52 FPS, 76ms latency	Demonstrates object tracking pipeline integration
Pathak et al. (2024)	Illegal dumping identification	YOLOv5 + OCR	Surveillance system	97% detection rate	Illustrates multi-model synergy in waste tracking
Sayem et al. (2025)	Dataset realism and robustness	GELAN-E, DenseViT	WaRP dataset	83.1% accuracy	Supports contextual dataset importance
Alharbi et al. (2025)	Behavioral trash detection	YOLOv8, MoViNet, Haar	Public surveillance	Multi-model integration	Emphasizes need for behavioral datasets
Abo-Zahhad and Abo-Zahhad (2025)	Overflow detection in cluttered conditions	YOLOv5 vs YOLOv8	Low-resource/mobile hardware	95–96% accuracy	Validates YOLO under occlusion and low light

Chapter 3

System Analysis

3.1 SWOT analysis

This section aims to provide a SWOT analysis of key artificial intelligence (AI) and machine learning components that underlie the application system. Figure 3.1 consists of all the key strengths, weaknesses, opportunities, and threats of the core AI/ML components used in the project.

Technology	Strengths	Weaknesses	Opportunities	Threats
YOLOv11 (Server)	Fast, accurate detection	Needs GPU; not for mobile	Suitable for cloud-based real-time processing	Impacted by server issues; needs retraining
YOLOv11n (Mobile)	Lightweight; real-time on mobile	Lower accuracy; limited for complex scenes	Offline use; supports TFLite and mobile accelerators	Dependent on device hardware; model drift risk
SORT	Fast tracking; mobile-friendly	Poor with occlusion and re-ID	Enables offline tracking	Weak in crowded or fast-motion scenarios
DeepSORT (Server)	High accuracy with re-ID support	Heavy GPU usage; not mobile-compatible	Useful in multi-user detection scenarios	Performance drops under high load
PyTorch	Full training-to-deployment support	Hard to combine with mobile tech stacks	Exports to ONNX/TFLite; widely adopted	API/version compatibility issues
TFLite	Low-latency mobile inference	Needs tuning; limited custom layer support	Good for offline and older mobile devices	Poor performance on outdated hardware
Dual Inference	Supports both server and mobile processing	Adds system complexity	Improves flexibility and reliability	Output differences may reduce user trust

Figure 3.1: SWOT analysis

3.2 Technical Analysis

The system combines the qualities of both artificial intelligence (AI) and machine learning (ML) by introducing intelligent waste classification and disposal verification on the basis of the motion to the mobile application. The architecture was intentionally designed to achieve the best trade-offs can be made between the high detection accuracy, the low-latency inference, and scalability across both the mobile and the server environments.

3.2.1 AI/ML Algorithms and Model Selection

The project is based on two fundamental machine-learning modules: object detection and motion tracking. To accomplish these objectives, models must be selected that provide the optimal trade offs between accuracy, real-time inference, model size, and resource efficiency. Accordingly, YOLO11 and YOLO11n were selected for object detection, while DeepSORT and SORT to be used as part of the tracking framework, these decision were both technical and strategic in terms of operational goals.

Object Detection

The mobile app implements two state-of-the-art object detection models, YOLO11 and YOLO11n. Since the project uses a hybrid inference approach, YOLO11 is integrated on the server for its accuracy, while YOLO11n is deployed on mobile for its lightweight architecture. These models were able to perform single-stage detection, which allowed real-time classification of two primary categories of waste: biodegradable and non-biodegradable.

Internal Working Principles of YOLO11

- Grid Division: YOLO11 divides the input picture into a grid of S S 2. In this grid, every grid cell has a specific task of predicting bounding boxes and the respective confidence scores when the center of an object has fallen within a grid cell.

- Bounding Box Prediction: Each grid cell predicts bounding boxes, which consist of coordinates (x,y), width(w) ,and height(h). It also consists of a confidence score that indicates the object occurrence and the bounding box accuracy.

- Class Prediction: The model predicts the probabilities of classes for each bounding box.

Internal Architecture of YOLO11

As outlined by Khanam and Hussain (2024), YOLO11 is a state-of-the-art object detector that is based on the YOLO ”You Only Look Once” architecture. It is known for its single forward pass detection, which makes it significantly faster compared to other detectors like R-CNN. The key components of YOLO11 are as follows:

1. **CSP-based Backbone**: The YOLO11 uses feature extraction based on Cross Stage Partial networks. C3K2 block was the main innovation since it was more

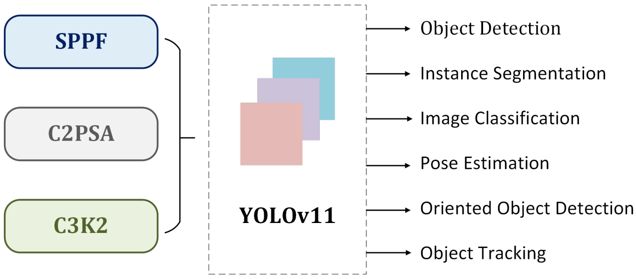

Figure 3.2: YOLO11 Architecture

computationally efficient varient of the CSP bottleneck. This improved precsion and speed while also maintaining parameter efficiency.

2. **SPPF (Spatial Pyramid Pooling-Fast**: SPPF captures contextual information at multiple receptive fields hence allowing the network to process objects form different scales.

3. **C2PSA (Cross Stage Partial with Spatial Attention)**: C2PSA block enhances the model by introducing spatial attention mechanism. Utilization of spatial information helps a model allocate computation power towards the significant parts of an image, a fact which intensifies the detection performance, particularly with small or close targets.

4. **Neck Feature Aggregation**: YOLO11 is an evolutionary development of the initial YOLO architecture. In addition to utilizing optimized neck architecture with C3k2 block to serve as an aggregation and fine-tuning of the multi-scale feature representation, it also integrates self-attention mechanism further to become a fusion and a feature distinguish modal.

5. **Detection Head**: Multiple C3k2 blocks is utilized to process the aggregated feature and output bounding box coordinates, objectness scores and class probabilites.

6. **Loss Function**: YOLO11 utilizes CIoU losses for precise bounding box regression and focal loss.

YOLO11n for Mobile inference

YOLO11n is a quantized and compressed version of the YOLO11. It consists of:

- Smaller convolutional kernels and less number of layers.
- Diminished width and depth multipliers.
- TensorFlow compatibility which allows for acceleration.

Justification for BinHero

The selection of these algorithms and models are based on:

- The project is dependent on real-time classification as a core business in order to provide real-time feedback to its users.
- YOLO11 provides high quality, reliable performance in closed, server-based conditions where GPU adequate resources are available.
- YOLO11n makes mobile classification possible even for mid-range devices. While also providing offline support and a lower server burden.
- The application's requirement to handle dynamic trash disposal actions in a matter of seconds matches with its anchor-based infrastructure and the fast inference capabilities.

Motion Tracking

While waste classification is an important factor for validating disposal behavior, it is still insufficient on its own. The application empolys motion tracking with two tracking algorithms, DeepSORT and SORT, that links object IDs across frames, to make sure that the waste item is physically disposed of in the bin.

Internal mechanism of SORT

SORT which stands for Simple Online and Real-time Tracking is a light weight tracking algorithm which utilizes classic computer vision methods.

- Input: It takes bounding boxes coordinates from YOLO models as input.
- Kalman Filter: Based on the object's previous state, it predicts its future position.
- Hungarian Algorithm: Utilizes intersection over union (IoU) to match detection to predicted tracks.
- Track Management: Addition of new tracks and modify old ones and eliminate lost ones.

Internal mechanism of DeepSORT

Unlike SORT, deepSORT incorporates appearance-based re-identification to increase the tracking robustness. It includes:

- Feature Extraction: Utilizes a CNN that encodes a visual embedding of each object.
- State Estimation: Implements Kalaman filter similar to SORT.
- Data Association: combines visual apperance (consine distance) and motion (Mahalanobis distance) in the cost matrix to match detection between frames

Justification for BinHero

- SORT is a lightweight tracking alogrithm which requires minimal computational resources and is able to perform efficiently in real-time.This allows it to provide an immediate user feedback in real time when users perform an action of disposal without overloading or slowing down.
- DeepSORT is more computationally heavy algorithm which is deployed on a server. It can cope with complex tasks with even greater accuracy and robustness (like multiple users or occlusion) using appearance-based Re-ID.

Table 3.1: Algorithm and Model Selection Summary and Justification

Requirement	**Solution**	**Reason for Selection**
Real-time Object Detection	YOLO11, YOLO11n	Rapid single-pass detection with high accuracy
Mobile Inference	YOLO11n	supoorts offline usage via tflite, operate on mid-range mobile phones.
Motion verification for Disposing actions	SORT, DeepSORT	Verifies object movement into the bin with a consistent temporal pattern.
Occlusion robustness	DeepSORT	manages scenarios with several users and increases the precision of behavioral analysis

3.2.2 Mathematical Formulation of YOLO11

YOLO11 (You Only Look Once, version 11) is an object detection model, which can predict bounding and class probability of an object using full image and having only one evaluation. Although the mathematical foundations remain a modification of the previous YOLO architecture design, it maintains the main detection and loss concept that made its precursors so particular.

3.2.3 Model Output

Given an input image, YOLO11 divides it into an $S \times S$ grid. For each grid cell, the model predicts B bounding boxes, each with:

- Center coordinates (x, y)
- Width w and height h
- Objectness confidence score C

- Class probabilities for C classes: $p_1, p_2, \ldots, p_C$

The output tensor per grid cell is:

$$\text{Output} = [x_1, y_1, w_1, h_1, C_1, \ldots, x_B, y_B, w_B, h_B, C_B, p_1, \ldots, p_C]$$

3.2.4 Loss Function

The overall loss L in YOLO11 is a sum of localization, objectness, and classification losses:

$$\begin{aligned}
L = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 \\
& + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\
& + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}$$

Where:

- $1_{ij}^{obj} = 1$ if an object appears in cell i and bounding box j is responsible for the prediction, otherwise 0.
- $\mathbb{1}_{ij}^{noobj} = 1$ if no object appears in cell i for bounding box j
- λ_{coord} and λ_{noobj} are hyperparameters to balance the loss terms
- $(x_i, y_i, w_i, h_i, C_i, p_i(c))$ are the predicted values
- $(\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i, \hat{C}_i, \hat{p}_i(c))$ are the ground truth values

3.2.5 Performance Metrics

The standard object detection metrics are used to assess YOLO11.

- **Precision**, **Recall**, **F1 Score**
- **Average Precision (AP)**
- **Mean Average Precision (mAP)**: commonly reported as mAP@0.5 and mAP@0.5:0.95

3.2.6 Performance Considerations and Trade-offs

The models selected illustrate a harmony between operational limitations and performance. Table 3.2 consists of a comparative discussion of the key components of AI that are incorporated into the BinHero platform

Table 3.2: Comparison of AI components in BinHero

Factor	**YOLO11 (Server)**	**YOLO11n (Mobile)**	**DeepSORT (Server)**	**SORT (Mobile)**
Accuracy	High	Moderate	High	Low
Inference Speed	Fast, GPU usage	30 FPS	Real-Time	Very Fast
Model Size	Large	<10MB	Medium	Very small
Resource Demand	High	Low	High	Very low

3.3 Requirement specification

3.3.1 Functional Requirement

1. **Waste Classification Accuracy**
   To provide feedback to the systems with a high correlation, and to allow implementation of an effective system of scoring users, an object detection model would need to perform classification with an accuracy of at least 85% on the biodegradable, non-biodegradable, and general waste bins.

2. **Real-Time Object Detection**
   The server-side version of YOLO11 is expected to show inference rates of at least 30 frames per second, whereas the mobile version of YOLO11n is required to reach at least 20 frames per second so that the on-device training could be responsive. This performance profile ensures that the integration with the camera-initiated disposal events takes place without any problems.

3. **Motion Based Trash Disposal Validation**
   The system should ensure that a waste item sensed in the hands of the user proceeds to the bin. This is done through the use of motion tracking (using SORT on a mobile device and DeepSORT on a server) to explicitly connect the identity of the objects across frames by checking their location trajectories.

4. **Preprocessing pipeline**
   A set of preprocessing manipulations, i.e., resizing, normalization, and motion-blur

filtering, is required on each captured frame before inference. This is necessary to maintain the quality of data and ensure consistency in inference when deploying mobile devices and servers.

5. **Hybrid Inference Architecture**
   The system must enable the execution of two modes of AI:

   - Mobile: On device YOLO11n and SORT.
   - Server: YOLO11 + DeepSORT inference in the cloud.

   The backup mechanism to ensure continuity of inference in the event of unreachability at the server should be adopted.

6. **Action based Reward Triggering**
   Upon confirmation by the tracking algorithm that an object has been properly disposed of, multiple actions should follow: the user should have their point tally automatically updated, positive feedback should be triggered, and the event of successful disposal should be recorded in order to be analyzed later.

3.3.2 Non-Functional Requirements

1. **Inference Speed**
   On the server-side, inference cannot have a latency greater than 200 milliseconds, and mobile-side inference must be less than 100 milliseconds per frame. These limitations ensure that feedback can be rendered in real time in disposal activities.

2. **Model Precision and Recall Thresholds**
   The precision has to be higher than 0.85, and the recall must be 0.80 to implement the use of an object detection model in practice. These metrics are crucial to prevent false positives and guarantee confidence in AI-generated feedback.

3. **Scalability of AI Pipeline**
   The architecture has to support the horizontal scalability of the cloud-based inference API such that multiple requests from different users can be enabled to run concurrently. The server is recommended to enable support for batch inference and GPU parallelism.

4. **Data Privacy and Security**
   All images captured by the user will be processed fully in memory, except where the user specifies the permanent storage permission. Their interface should use both an authentication scheme and thorough input sanitization against adversarial attacks.

Chapter 4

System Design

4.1 System Architecture Diagram

The architecture of this application is designed as a modular, AI-powered mobile application that integrates with user interface, real-time location services, real-time communications, cloud-based services, and smart inference pipelines. It is developed to support a hybrid AI inference model with the ability to perform computation both on mobile devices and dedicated servers to maximize the performance, accuracy, and responsiveness. This section provides a comprehensive analysis of all the parts, the AI pipeline, model workflows, and the ability of systems to interact.

4.1.1 High-Level Overview

At the top level, the entire system is a cross-platform mobile app which we initially conceptualized in Figma and then built using React Native. The service is connected to several modules: the Google Maps API to geolocate bins, the WebSockets library to interact in real-time, a dual-mode AI inference stack (mobile and server), a Flask-based AI service, an Express.js back-end, and a MongoDB database. The AI modules become the real-time motion-tracking and object-detection components of the system, thus the smart core of the system.

4.1.2 Frontend and User Interaction Layer

The mobile application serves as the main user interface, allowing users to:

- record waste disposal in the form of a video.
- View AI validation in real-time.
- Keep track of points and take part in gamified activities.
- Geotag bin locations

Built using React Native, ensuring cross-platform compatibility with Android and iOS. Figma is used to design the prototypes of the design and the user journey with a focus on its accessibility and the readability of interaction.

4.1.3 Geolocation and Real-Time Communication

Google Maps API allows bin mapping, route tracking, and spatial data visualization. At the same time, WebSockets provide low-latency communication between the application and the backend services. The architecture allows processing motion updates along with AI detection triggers and the related inference feedback in real time.

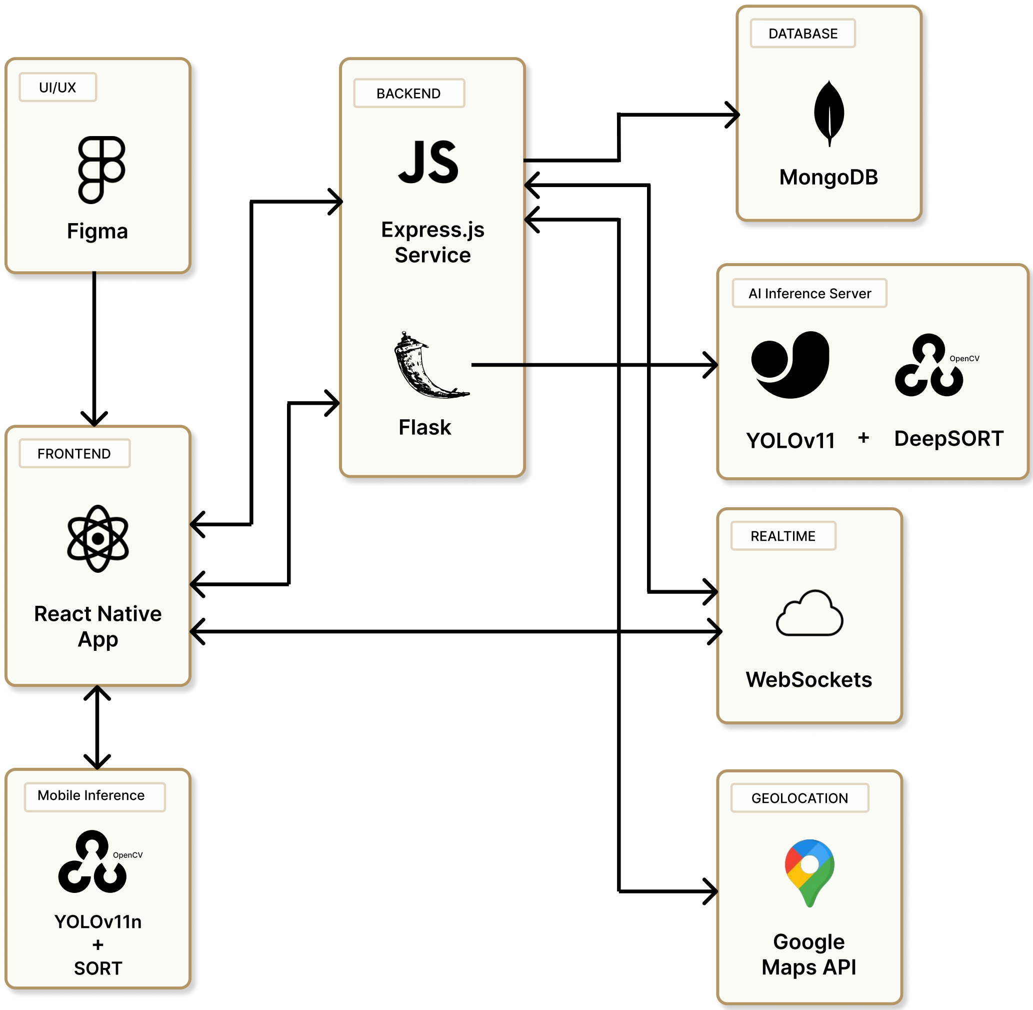

Figure 4.1: System Architecture

4.1.4 AI Inference Architecture

The AI Module of the project utilizes PyTorch to perform as the intelligence engine, and is designed in two modes, including mobile-side and server-side. OpenCV is completely integrated within both operating modes to simplify acquisition of frames, preprocessing, and visualization on a real-time basis.

Mobile-side inference

A quantized and lightweight version of YOLO11. YOLO11n is applied at the final stage, locally, to make inferences. It is optimized for edge inference, having the following characteristics:

- Smaller convolutional layers with diminished depth multipliers.
- Tensorflow compatibility for mobile acceleration.
- Real-time detection at approximately 30 FPS.

Bewley et al. (2016) introduced SORT (Simple Online and Realtime Tracking). It uses a Kalman filter and Hungarian matching algorithm to relate bounding boxes across frames, thus making local motion analysis (a key requirement in detection of disposal actions, even where connectivity is low. It was utilized for motion tracking on the client-side. Likewise, Libraries like OpenCV is utilized to capture frames from the device's cameras, and apply various preprocessing techniques such as resizing, normalizing, etc.

Server-side inference

Computationally more demanding work is redirected towards a cloud-based AI Inference Server that accommodates:

- YOLO11 for a better precision on object detection.
- DeepSORT for a robust multi-target tracking, it uses both spatial and visual features, suggested by Re-ID, and embeddings.
- A Flask microservice to handle stages such as preprocessing, batch inference, and post-processing of some end-to-end workflow.

The frames acquired by the mobile application get sent to the server with the help of WebSockets or REST endpoints. To maintain the interaction, the inference server maps the AI pipeline and provides classification scores, motion vectors, and confidence scores. These results are then channelled out to the front-end to justify action and feedback.

4.1.5 AI Pipeline Workflow

The system can be considered modular, and it includes the following AI blocks:

- **Image Acquisition:** During the disposal action mobile device camera is used to record frames. It would be triggered through the application's UI. The OpenCV is utilised to capture each frame of a video stream in real-time and then process it.
- **Pre-processing**: All the captured frames undergo various preprocessing steps such as normalization and resizing. Frames are queued for server-based inference to maximize the GPU batch processing. The step also filters out frames with motion

blur or bad lighting to preserve the input quality. OpenCV fulfills these tasks on both the mobile and server sides and thus makes the frame quality and model compatibility consistent.

- **Object Detection**: The YOLO models classify the items into three classes: biodegradable and non-biodegradable. Bounding boxes, confidence scores, and probabilities are examples of detection outputs. The module runs inference at about 30 frames per second, guaranteeing real-time performance for server and mobile versions.

- **Tracking**: The detections received by the system are sent to SORT/DeepSORT, and the movement vectors are obtained to check whether the object passed or not the user's hands to the bin. Using a tracker, each object in subsequent frames is given a unique identity, thus providing consistency of time in the motion analysis process.

- **Decision Logic**: Disposal validation logic is a compliance rule that checks an identified waste to determine whether it meets a reasonable disposal trajectory. The reasoning uses spatial thresholds and velocity approximations to prevent false confirmations and rewards points to the score, activates responsive feedback, or causes corrective directions.

4.1.6 Backend and Database

Express.js back-end acts as a middleware, which performs the following main functions:

- Authentication and API routing
- Flow of data between frontend, AI server, and database.
- User session management.

Information about the users, including their profile, is stored in the MongoDB database along with metatags of bin locations, the classification history, and gamification data. It is flexible in processing the outputs of AI and heterogeneous data types in virtue of its NoSQL architecture.

The combination of real-time artificial intelligence with an optimally designed backend framework and its responsive user interface will make the proposed architecture deliver scalable, accurate, and community-driven waste disposal solutions. The AI pipeline of hybrid inference allows the system to dynamically match the flexibilities to user situational contexts, preserving its performance and reliability aspect in smart waste management.

4.2 Use Case Diagram, UML Diagram, and Other Visual Representations

Unified Modeling Language (UML) diagrams are a mandatory tool in ensuring a seamless integration of artificial-intelligence-based modules into the system structure. UML can facilitate this because it provides a graphical framework through which the relationship between algorithmic development and the requirements of the end-user can be articulated, and the capabilities of functions like motion tracking and object identification can be brought into line with functional need and the practicality of actual implementations. UML diagrams also help determine trigger points of artificial intelligence inference, data flow, and feedback mechanisms, and hence serve as a pillar in the system planning and implementation.

4.2.1 Use Case Diagram

The figure 4.2 provides extensive information about the relations between the actors of the application and the embedded AI parts. The following are some of the primary actors:

- User: The system connects to artificial-intelligence elements, such as the ones that scan the waste objects, receive the results of object detection, and give feedback on the accuracy of the process in real-time. The interactions trigger AI pipelines, which are exemplified by YOLO11 in object classification and DeepSORT or SORT in motion prove.

- Admin: The Administrator has the management of the backend, which should involve the approval of bin placements, administration of users, and moderation of content. Even though these administration roles do not have any direct interaction with the AI processes, they still build on the given model by investigating the results of the detection and measuring the effectiveness of user input.

- Contributor: The task of contributing bin locations, verifying them, which provides the imprint of proper behavior during removal/disposal within specific areas, that will increase the accuracy of detection within geographical locations.

A Use Case Diagram helps identify important exchanges where the system initiates or consumes AI services. The use case of Scan Waste, in its turn, calls the object detection model, and conversely, Receive Real-Time Feedback is closely connected with the verification logic of motion tracking. Moreover, the transformations of specific gamification use-cases, e.g., the “Earn Points and Rewards” and the “Track Progress,” have begun to be powered by AI feedback loops to evaluate the discarding precision and frequency to translate your behavioral information into effective motivators to be recognized by the users.

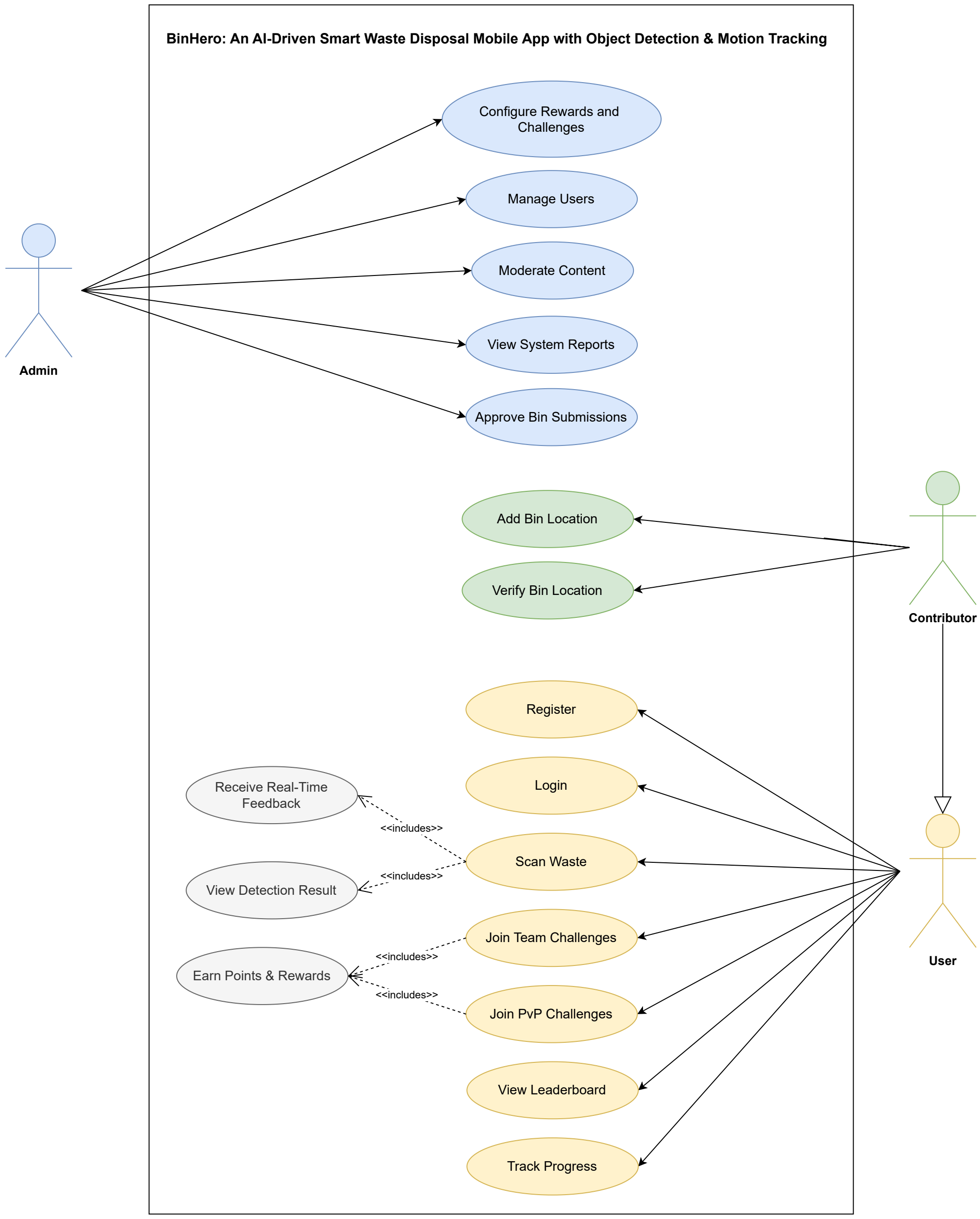

Figure 4.2: Use Case Diagram

4.3 Class Diagram

The Figure 4.3 represents the system's class diagram, which follows a modular, layered design that clearly distinguishes the fields of user-interaction, artificial-intelligence processing, game mechanics, and data management. At the core of the user structure is an abstract User class, which serves as the base, and functionalities that are added to the contributor and Admin, respectively, to have functionalities on role-specific duties. The main classes defining the architecture of the application include the Bin, Challenge, and GamificationEngine, although the latter is by far the most important, as it is the agent system used to encourage user action. In cooperation with these entities, there are AI-centered classes, in particular AIModel and WasteDetection, that recognize and get image inputs and award points to the user with the help of waste recognition detection. Supporting components such as DatabaseManager and MapService cultivate the fundamental functionalities: data persistence and location-based services. Through inheritance and associations, the object-oriented methodologies get threaded into the design, hence ensuring the reuse of code and the sustainability of systems.

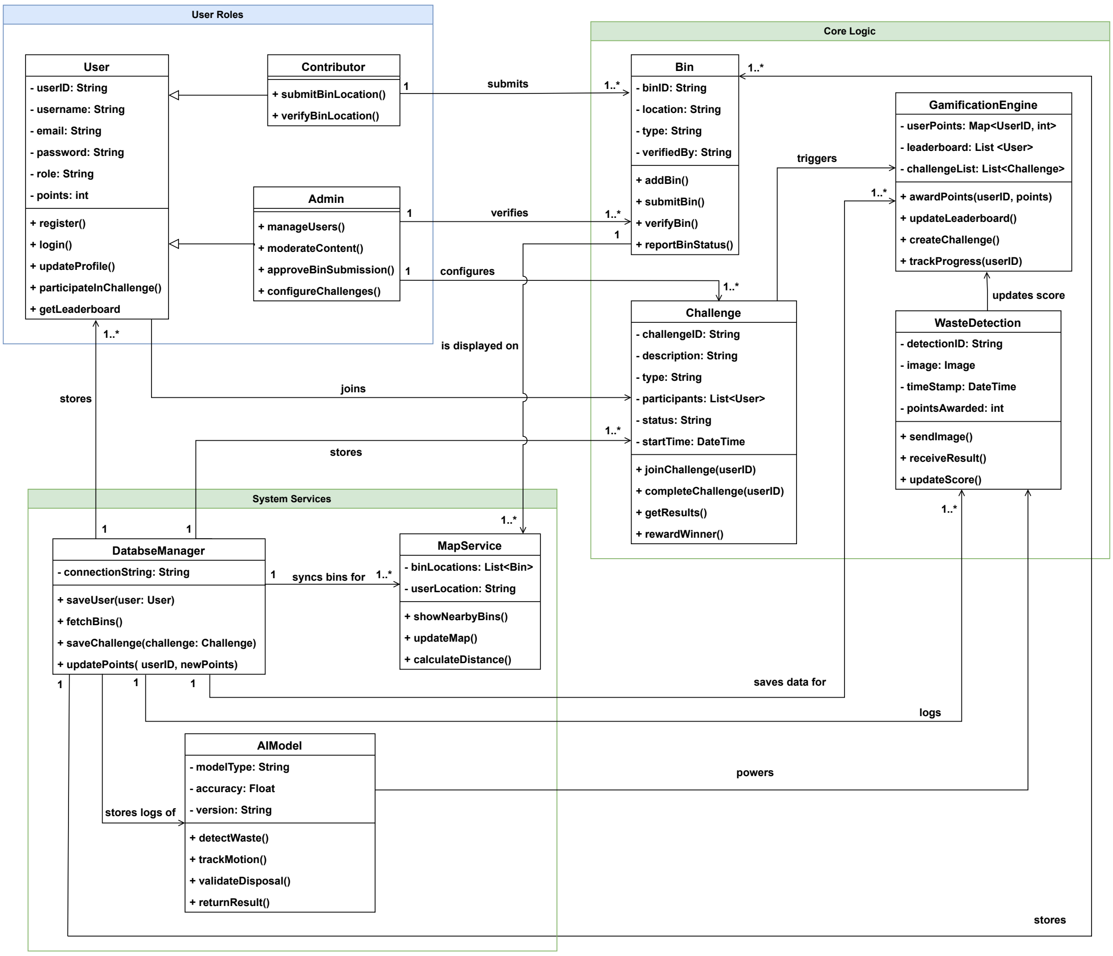

Figure 4.3: Class Diagram

4.4 Activity Diagram

The given activity diagrams outline the main processes within the project system. Figure 4.4 illustrates the user authentication process, detailing secure registration and logging procedures. Likewise, Figure 4.5 depicts PvP challenge flow, where the user interaction is guided by AI functionality, such as object detection and real-time score.

1. **User Authentication Flow**
   User Authentication Flow forms the key mechanism with which individuals will log into the application. On starting the application, the user is shown the choice of either creating a new account or logging in to an existing one. A user must fill in a registration form and deliver all the required personal data at registration, but only valid credentials are necessary in the process of following the subsequent login process. The system matches the received data with existing data and, based on the results, enables access or sends the error message to correct. After the authentication procedure has been executed, a user goes through the process of setting up permissions and arranging his or her profile, then goes to the main interface. The flow ensures an orderly and secure onboarding of all users, thus preserving the privacy and integrity of the application. While this flow does not incorporate AI functionality, it still serves as a fundamental mechanism of controlling access to the more advanced functionalities of the system. The authentication process is also imperative in the formation of user-specific sessions, thereby making it easy to accurately track participation, scoring, and behavior in the system. Therefore, the user Authentication Flow is crucial for customized and responsible user experiences in the larger ecosystem.

2. **PvP Challenge Flow**
   PvP Challenge Flow reflects the gamified procedure of waste disposal that encourages users to follow environmentally responsible practices through interactive gameplay, making it an essential component of the application. Users can navigate to various challenge sections once they are authenticated. The system first examines the eligibility of an applicant according to set standards; once this is approved, the system assigns the applicant a suitable opponent and also indicates that the challenge has begun. At the same time, AI-enabled waste detection units will monitor and track the user's waste disposal actions in real time, and then they will score the user in real time based on accuracy and overall performance. The figure 4.5 depicts how the integration of AI into a structured interaction cycle is smooth. The system still observes the session, determines whether the exchange has ended based on either time limits or scoring points then it declares a winner using automated reasoning. After the challenge is over, rewards are calculated, and the scoreboard position is updated. The PvP challenge flow shows how the project implements AI to observe user behavior data in real time, offer continuous feedback

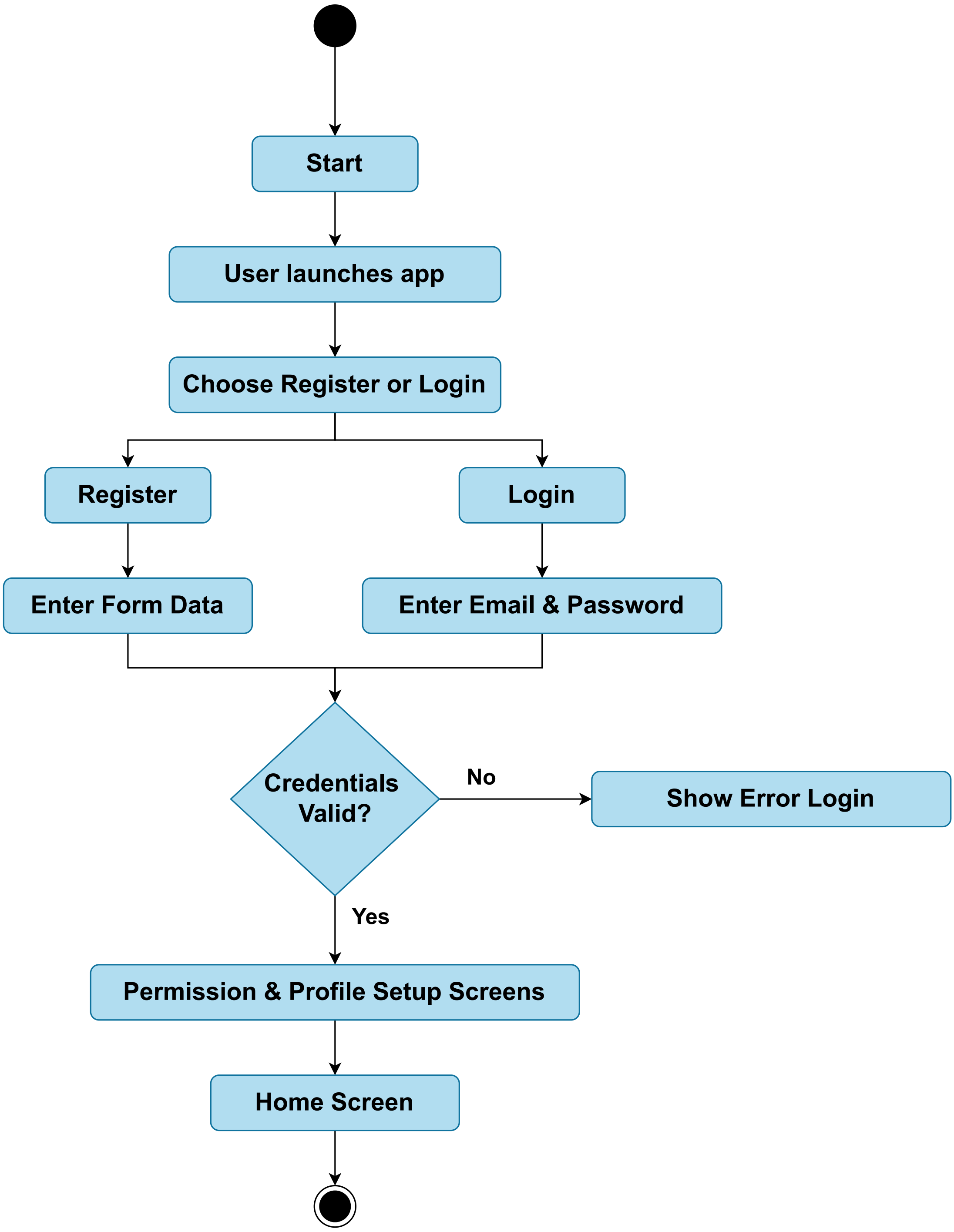

Figure 4.4: User Authentication Activity Diagram

to users, and keep them engaged with gamification. It reinvents the very process of waste management by transforming all the following steps of disposing of waste matter into a rather interactive process moderated by intelligent automation and the sense of behavioral responsibility.

3. **Waste Disposal Validation Flow**

   The figure 4.6 illustrates an organized sequence of interactions that incorporates AI features into every phase of user involvement. The sequence starts once a user steps in front of a trash bin, which triggers a system to send out its camera and capture an image of a waste object, and send the obtained visual data to the backend server. An artificial-intelligence model on a server or in a mobile device performs the task of waste classification by determining the object to be in one of two categories: biodegradable or non-biodegradable. This classification gives the framework for making informed decisions, making sure that each action includes an understanding of its context. Simultaneously, they use a motion-tracking module that tracks the user as he or she performs the disposal action to ensure that the detected object is dropped into the bin. Once this verification is done, the system evaluates the result. Upon valid disposal, the system will grant the corresponding points, increment the user's score and reward, and the disposal will be recorded to be analysed later. In case the disposal does not pass verification, no reward is received by the user, as a consequence, promoting accountability and adequate behavior. The results of the user input are shown within the system as soon as the process is performed, with the prevalent gamification features of the achievement badges or progress notifications. According to the diagram, artificial-intelligence abilities, including classification, tracking, and decision-making, have been well assimilated into the system workflow. The given integration enables automatic reaction to a great range of stimuli, provision of immediate feedback, and the assuring level of user involvement. All of its components not only improve the performance of the operations but also lead to a responsive, smart, and user-oriented system architecture.

4.5 Sequence Diagram

The sequence diagram highlights the use of AI technology to address real-time decision-making and user experience by documenting the past interactions of users, backend services, and AI-enabled devices. This process starts by either logging in or registering through the mobile application. This application directly contacts the Auth API and the associated database to confirm the credentials entered. After a user is validated, he/she is placed in a Player-versus-Player game, where the participant creates communications with the Matchmaker Service, the WebSocket Hub, and an Opponent. In the course of gameplay, the user spots a waste item and scans it. The picture is sent to the backend,

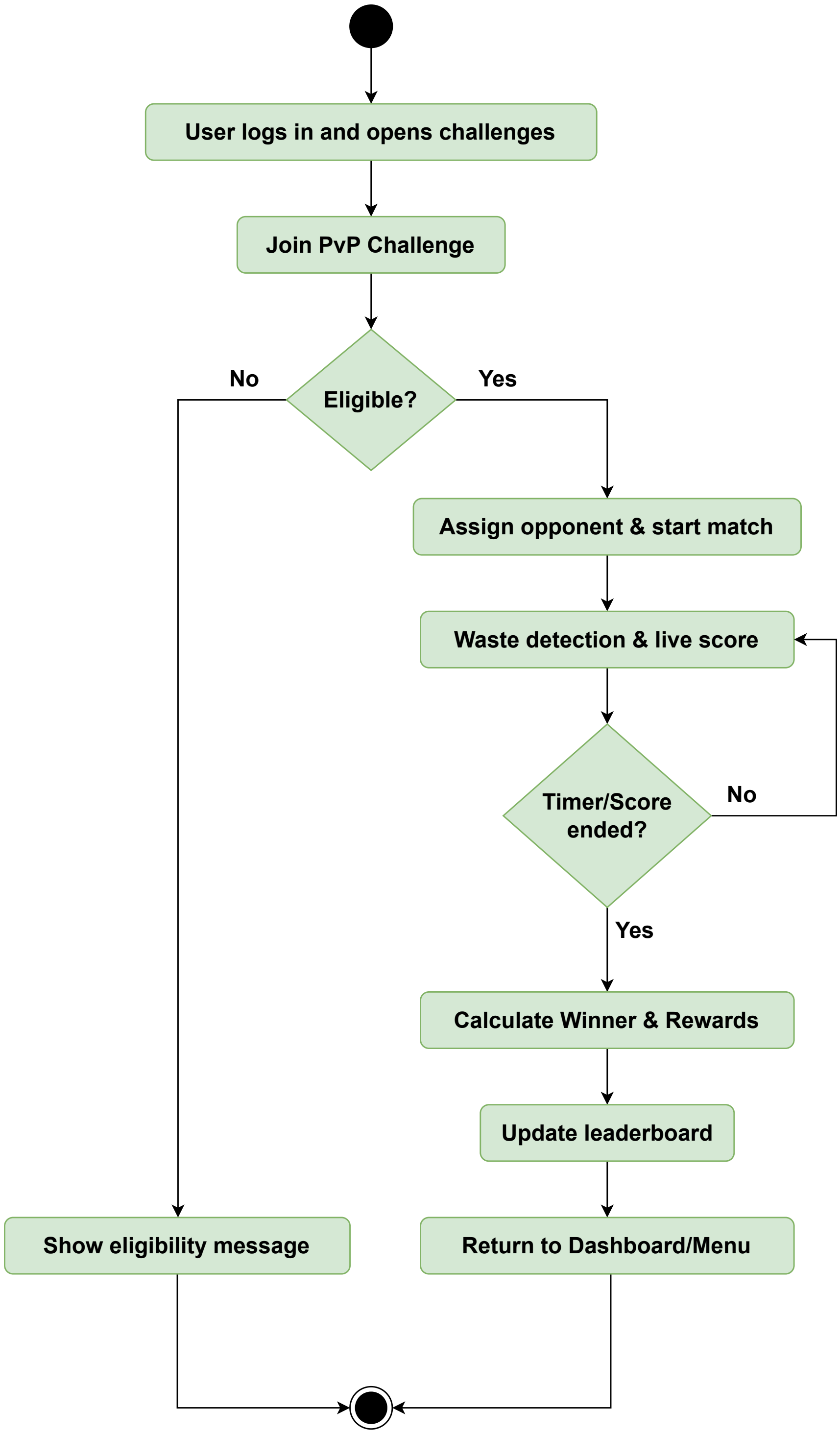

Figure 4.5: PvP Challenges Activity Diagram

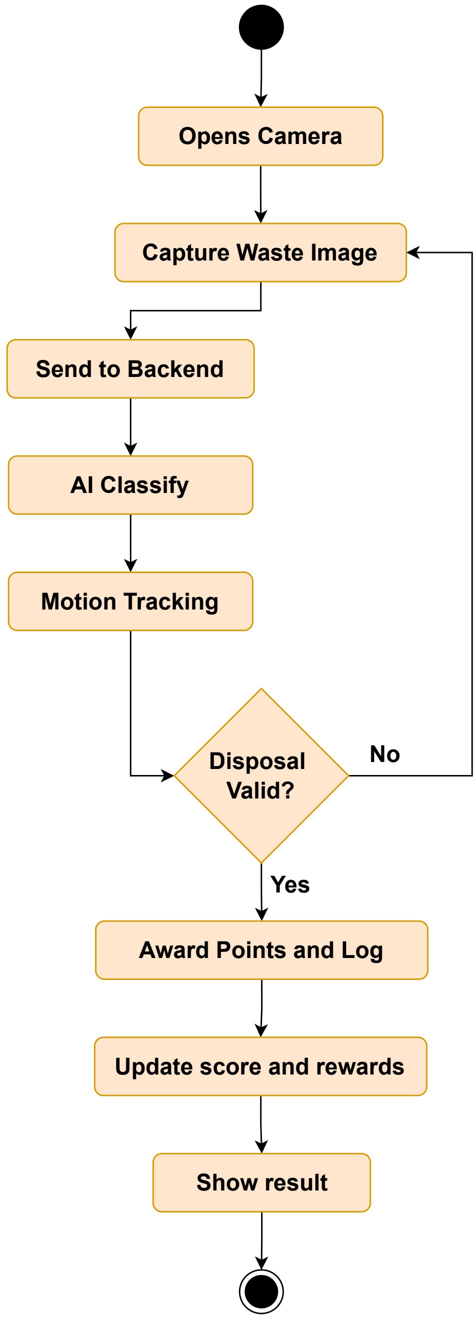

Figure 4.6: Waste Disposal Validation Activity Diagram

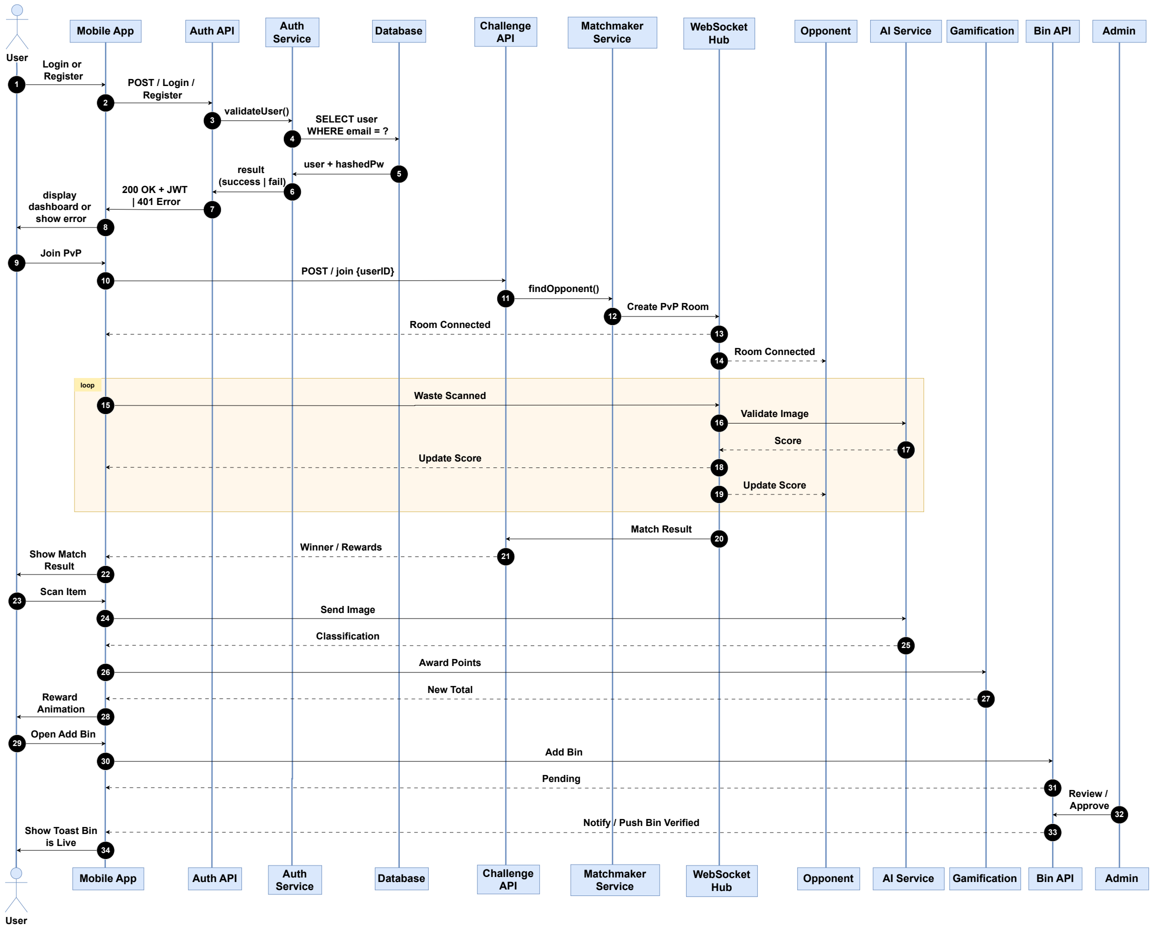

Figure 4.7: Sequence Diagram

where it is categorized using AI. The system checks the input and computes a score and posts a user profile through the Gamification service. The application presents performance analytics, awards, and outcomes by using a fun animated presentation. The users can also use the registration flow of bin, a nd the submissions are checked and verified by the Admin and the Bin API before making the content live. This diagram shows the smooth combination of all AI functions of classification and scoring into the larger and receptive architecture of the service to support the real-time and gamified waste disposal experience.

4.6 Pseudocode

This section gives a breakdown of the key algorithms that lie beneath the main components of our application, namely object detection based on the YOLO11 model and real-time tracking that relies on SORT and DeepSORT. The pseudocode details the main sequential steps by which the YOLO11 model can be trained to distinguish different types of waste, pre- and exported to an edge device with support for real-time tracking to maintain the object identity across frames. Collectively, they comprise the main part of the AI in the work of the system, as it allows detecting waste items and garbage bins with high accuracy in dynamic settings and monitoring them over time.

The *pseudocode 1* discusses fine-tuning the YOLO11 object-detection model on one of the custom datasets that contain such classes as biodegradable waste, non-biodegradable waste, and garbage bins. It involves downloading a pre-trained model, preparing the dataset to achieve the objectives of the detection task, and then training the model for a few epochs to fine-tune it to its purpose.

The *pseudocode 2* provided below leads to exporting the pre-trained model (YOLO11n) to the TensorFlow Lite version to enable the use of the model on an edge device and depicts the reloading procedure using on-device inference. This process is mobile-friendly, and the model still has the capability of detecting.

The next *pseudocode 3* represents the pseudocode of the implementation of the SORT (Simple Online and Realtime Tracking) method of simultaneous and multiple object tracking. It shows how the detections that are produced by a real-time object detector, including the YOLO11n, are passed to the SORT tracker. The algorithm processes all frames of the video stream and constantly updates and displays the tracking data of every object.

Algorithm 1 YOLO11 Fine-Tuning for Waste Classification

This *pseudocode 4* describes how DeepSORT, a more complex algorithm that improves the SORT algorithm by adding deep appearance feature embedding, works. It outlines the inference of detections of a model like YOLO11, where the feature extractor generates embeddings that are then used by the tracker in aligning and maintaining track of the identity of the objects across adjacent frames. This kind of use increases tracking accuracy, particularly in highly populated or partially obstructed scenes.

Input: Pre-trained model, dataset config
Output: Trained weights, metrics

```
                                                ▷ Import YOLO module
Import YOLO from ultralytics
                                                ▷ Load model
MODEL ← YOLO("yolo11.pt")
                                                ▷ Set dataset config
DATA_CONFIG ← "dataset/data.yaml"
                                                ▷ Set training params
NUM_EPOCHS ← 20
IMAGE_SIZE ← 640
DEVICE_ID ← 0
                                                ▷ Start training
TRAINING_RESULTS ← MODEL.train(
data=DATA_CONFIG,
epochs=NUM_EPOCHS,
imgsz=IMAGE_SIZE,
device=DEVICE_ID
)
Print("Training completed. Check output folder.")
```

Algorithm 2 YOLO11n Export to TensorFlow Lite

```
Input: Custom pre-trained YOLO11n model
Output: TFLite model ready for inference
                                                ▷ Import YOLO
Import YOLO from ultralytics
                                                ▷ Load model
MODEL ← YOLO("yolo11n.pt")
                                                ▷ Set export format
EXPORT_FORMAT ← "tflite"
                                                ▷ Export model
MODEL.export(format=EXPORT_FORMAT)
                                                ▷ Load TFLite model
TFLITE_MODEL ← YOLO("yolo11n_float32.tflite")
                                                ▷ Ready for inference
```

Algorithm 3 Object Tracking with YOLO11 and SORT

```
Input: Video stream, pre-trained detector
Output: Tracked object visualizations
                                                ▷ Import modules
Import video_stream
Import object_detector
Import SORTTracker
                                                ▷ Initialize detector
DETECTOR ← LoadYOLO11(weights="YOLO11n.pt", classes=3)
                                                ▷ Initialize tracker
TRACKER ← SORT()
                                                ▷ Open video stream
VIDEO ← OpenVideoStream("input_video.mp4")
while VIDEO.has_frames() do
    FRAME ← VIDEO.read_next_frame()
                                                ▷ Run detection
    DETECTIONS ← DETECTOR.detect(FRAME)
                                                ▷ Update tracker
    TRACKED_OBJECTS ← TRACKER.update(DETECTIONS)
                                                ▷ Draw tracking results
    for all object in TRACKED_OBJECTS do
        Draw bounding_box on FRAME
        Draw track_id on FRAME
    end for
    DISPLAY FRAME
end while
                                                ▷ Release resources
VIDEO.release()
Close display_window
```

Algorithm 4 Object Tracking with YOLO11 and DeepSORT

```
Input: Video stream, detector, appearance model
Output: Tracked object visualizations
                                                    ▷ Import modules
Import video_stream
Import object_detector
Import DeepSORTTracker
Import AppearanceFeatureExtractor
                                                    ▷ Initialize detector
DETECTOR ← LoadYOLO11("YOLO11n.pt", classes=3)
                                                    ▷ Initialize DeepSORT
APPEARANCE_ENCODER ← LoadFeatureExtractor("cnn_encoder.onnx")
TRACKER ← DeepSORT(appearance_model=APPEARANCE_ENCODER)
                                                    ▷ Open video
VIDEO ← OpenVideoStream("input_video.mp4")
while VIDEO.has_frames() do
    FRAME ← VIDEO.read_next_frame()
                                                    ▷ Run detection
    DETECTIONS ← DETECTOR.detect(FRAME)
                                                    ▷ Extract features
    FEATURES ← [ ]
    for all bbox in DETECTIONS do
        CROP ← CropImage(FRAME, bbox)
        FEATURE ← APPEARANCE_ENCODER.extract(CROP)
        Append FEATURE to FEATURES
    end for
                                                    ▷ Combine data
    DETECTIONS_WITH_FEATURES ← Combine(DETECTIONS, FEATURES)
                                                    ▷ Update tracker
    TRACKED_OBJECTS ← TRACKER.update(DETECTIONS_WITH_FEATURES)
                                                    ▷ Draw results
    for all object in TRACKED_OBJECTS do
        Draw bounding_box on FRAME
        Draw track_id on FRAME
    end for
    DISPLAY FRAME
end while
                                                    ▷ Release resources
VIDEO.release()
Close display_window
```

Chapter 5

Conclusion

The BinHero project is an example of how AI systems can be used with practical effectiveness on pressing environmental issues, as it provides real-time waste classification and confirms behavioral patterns. The combination of motion tracking and object detection allows the system to authenticate the activities of users and promote the appropriate disposal practice. The integration of game elements, such as PvP and reward systems, takes user activity to new levels and supports achieving sustainable environment behavior as well. The architecture is scalable, mobile, and data privacy-friendly, and offers a community that finds a responsive solution to urban waste management. Future advancements will incorporate features like government cooperation, linguistic support, and AR integration, which may increase its usefulness.

References

Mohammed M Abo-Zahhad and Mohammed Abo-Zahhad. Real time intelligent garbage monitoring and efficient collection using yolov8 and yolov5 deep learning models for environmental sustainability. *Scientific Reports*, 15(1):1–26, 2025.

Eaman Alharbi, Ghadah Alsulami, Sarah Aljohani, Waad Alharbi, and Somayah Albaradei. Real-time detection and monitoring of public littering behavior using deep learning for a sustainable environment. *Scientific Reports*, 15(1):3000, 2025.

Ali Arishi. Real-time household waste detection and classification for sustainable recycling: A deep learning approach. *Sustainability*, 17(5):1902, 2025.

Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. *CoRR*, abs/1602.00763, 2016. URL http://arxiv.org/abs/1602.00763.

Wenhao Huang, Kazuhiro Mikami, Yin Chen, and Jin Nakazawa. Real-time image-based automotive sensing: A practice on fine-grained garbage disposal. In *Proceedings of the 13th International Conference on the Internet of Things*, pages 138–145, 2023.

Silpa Kaza, Lisa Yao, Perinaz Bhada-Tata, and Frank Van Woerden. *What a waste 2.0: a global snapshot of solid waste management to 2050*. World Bank Publications, 2018.

Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

Yeji Kim and Jeongho Cho. Aidm-strat: augmented illegal dumping monitoring strategy through deep neural network-based spatial separation attention of garbage. *Sensors*, 22(22):8819, 2022.

Everest Z Kuang, Kushal Raj Bhandari, and Jianxi Gao. Optimizing waste management with advanced object detection for garbage classification. *arXiv preprint arXiv:2410.09975*, 2024.

Md Shahariar Nafiz, Shuvra Smaran Das, Md Kishor Morol, Abdullah Al Juabir, and Dip Nandi. Convowaste: An automatic waste segregation machine using deep learning. In *2023 3rd International conference on robotics, electrical and signal processing techniques (ICREST)*, pages 181–186. IEEE, 2023.

Imane Nedjar, Mohammed M'hamedi, and Mokhtaria Bekkaoui. Real-time solid waste sorting machine based on deep learning. *International journal of electrical and computer engineering systems*, 15(7):581–589, 2024.

Nupur Pathak, Gangotri Biswal, Megha Goushal, Vraj Mistry, Palak Shah, Fenglian Li, and Jerry Gao. Smart city community watch—camera-based community watch for traffic and illegal dumping. *Smart Cities*, 7(4):2232–2257, 2024.

C Arun Prasath. Automated waste classification and segregation using yolov5 and robotic arm. *Frontiers in Mathematical and Computational Research*, pages 8–15, 2025.

Faizul Rakib Sayem, Md Sakib Bin Islam, Mansura Naznine, Mohammad Nashbat, Mazhar Hasan-Zia, Ali K Ansaruddin Kunju, Amith Khandakar, Azad Ashraf, Molla Ehsanul Majid, Saad Bin Abul Kashem, et al. Enhancing waste sorting and recycling efficiency: robust deep learning-based approach for classification and detection. *Neural Computing and Applications*, 37(6):4567–4583, 2025.

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

Appendices

Appendix A

Gantt Chart and Current Progress

A.1 Gantt Chart

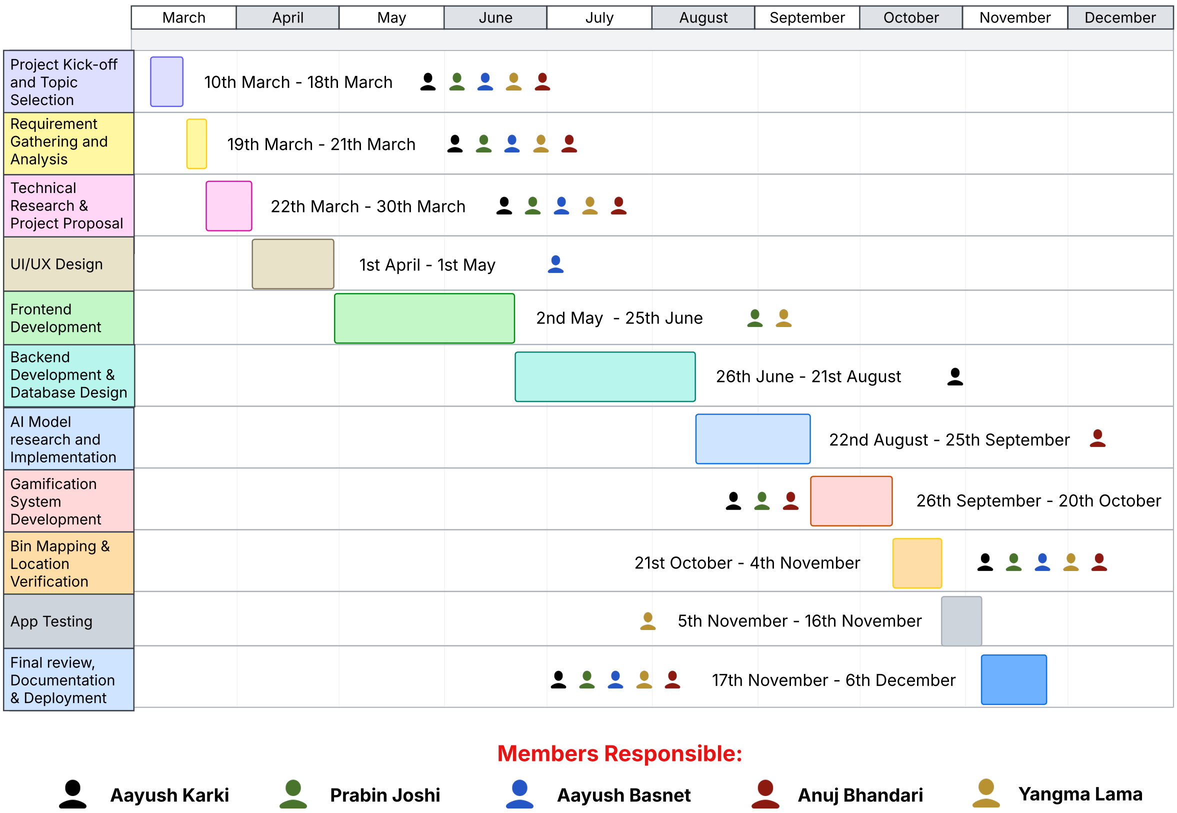

Figure A.1: Project Milestone Gantt Chart

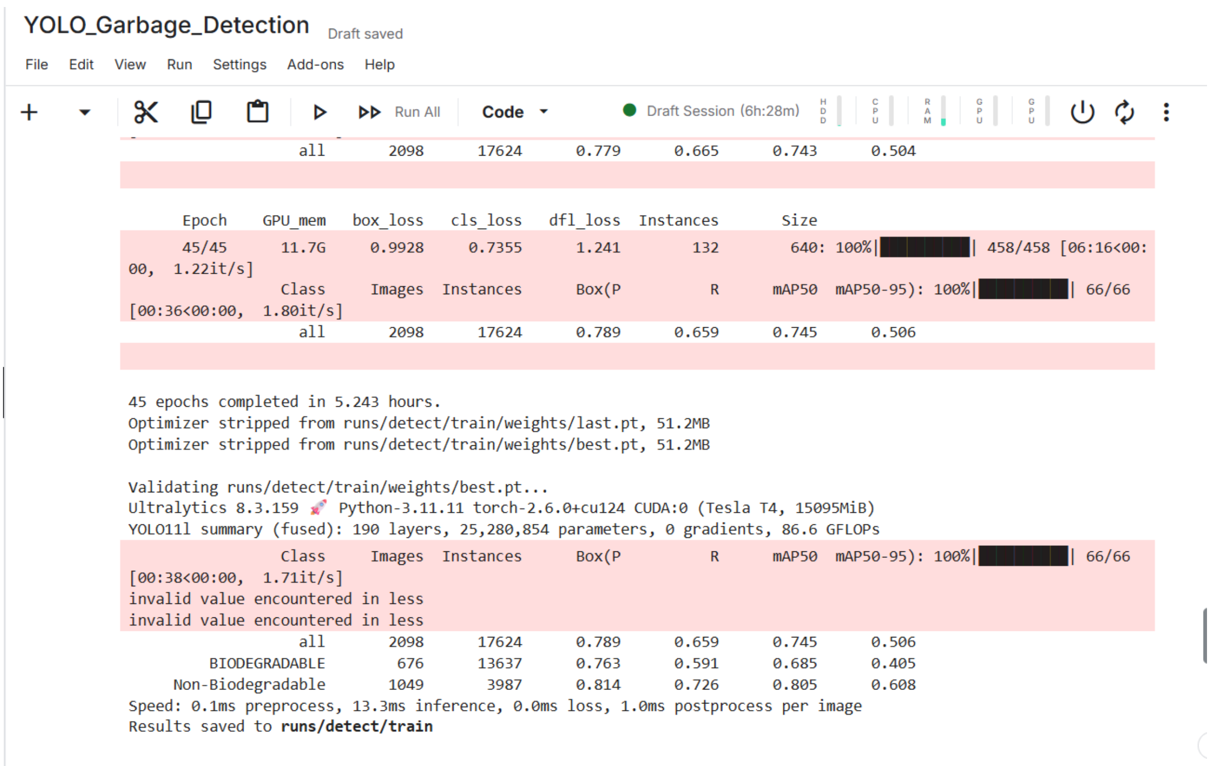

Figure A.2: YOLO11 Training

A.2 Current Progress and Preliminary Results

Table A.1: Current Performance of Detection Model

Class	Images	Instances	Precision	Recall	mAP@0.5	mAP@0.5:0.95
All	2098	17624	0.789	0.659	0.745	0.506
Biodegradable	676	13637	0.763	0.591	0.685	0.405
Non-Biodegradable	1049	3987	0.814	0.726	0.805	0.608

Inference speed: 13.3ms per image.

Appendix B

Log Sheet

IIMS COLLEGE

Bachelors of Computer Science (Hons)
Capstone Project Log Sheet

Notes on use of the project log sheet:

1. Minimum TEN (10) for Capstone/FYP Part 1 and TEN (10) for Capstone 2
2. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
3. A log sheet is to be brought by the STUDENT to each supervisory session.
4. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant sections of the form.
5. The student should make copies of the Log Sheet and must be inserted in the documentation (under appendices) in the softbound report.
6. It is recommended that students bring along log sheets of previous meetings during each supervisory session.
7. The log sheet is an important deliverable for the research and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the documentation, with sheets dated and numbered consecutively.

Student Name: Anuj Bhandari **Group:2** **Date:** 09/03/2025 **Meeting No: 1**

Project Title : BinHero

Supervisor's Name: Sachin Shrestha	**Supervisor's Signature:** Sachin

Item for Discussion: (Noted by Student **Before** Supervisory meeting)

- Introduction to Capstone module.
- Overview of deliverables and expectations
- Understanding how ideas are evaluated.

Record of Discussion: (Noted by Student **During** Supervisory meeting)

- Supervisor explained module structure and evaluation process.
- Emphasis on innovation, problem-solving, and technical depth
- Discussed importance of weekly logs, proposal, and final report

Action List (To be attempted or Completed by Student by the **Next** Supervisory meeting)

- Research potential project ideas.
- Search for team members.
- Prepare questions about the technical feasibility for ideas.

Note: A student MUST attend all weekly meetings as scheduled by the supervisor. In the event Supervisor could not be met, the student must immediately inform the Academic Coordinator and supervisor so that a meeting can be subsequently arranged. Under NO situation should a student miss a weekly appointment with the supervisor except during exceptional circumstances.

Figure B.1: Meeting 1

IIMS COLLEGE

Bachelors of Computer Science (Hons)
Capstone Project Log Sheet

Notes on use of the project log sheet:

8. Minimum TEN (10) for Capstone/FYP Part 1 and TEN (10) for Capstone 2
9. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
10. A log sheet is to be brought by the STUDENT to each supervisory session.
11. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant sections of the form.
12. The student should make copies of the Log Sheet and must be inserted in the documentation (under appendices) in the softbound report.
13. It is recommended that students bring along log sheets of previous meetings during each supervisory session.
14. The log sheet is an important deliverable for the research and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the documentation, with sheets dated and numbered consecutively.

Student Name: Anuj Bhandari **Group:**2 **Date:** 18/03/2025 **Meeting No:** 2

Project Title : BinHero

Supervisor's Name: Sachin Shrestha | **Supervisor's Signature:**

Item for Discussion: (Noted by Student **Before** Supervisory meeting)

- Initial discussion of possible project htemes
- AI development interests

Record of Discussion: (Noted by Student **During** Supervisory meeting)

- I was assigned with AI developer role for this project.
- Superviso advised role-specific learning plans.

Action List (To be attempted or Completed by Student by the **Next** Supervisory meeting)

- Confirmed responsibilites and milestones per role.

Note: A student MUST attend all weekly meetings as scheduled by the supervisor. In the event Supervisor could not be met, the student must immediately inform the Academic Coordinator and supervisor so that a meeting can be subsequently arranged. Under NO situation should a student miss a weekly appointment with the supervisor except during exceptional circumstances.

Figure B.2: Meeting 2

IIMS COLLEGE

Bachelors of Computer Science (Hons)
Capstone Project Log Sheet

Notes on use of the project log sheet:

15. Minimum TEN (10) for Capstone/FYP Part 1 and TEN (10) for Capstone 2
16. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
17. A log sheet is to be brought by the STUDENT to each supervisory session.
18. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant sections of the form.
19. The student should make copies of the Log Sheet and must be inserted in the documentation (under appendices) in the softbound report.
20. It is recommended that students bring along log sheets of previous meetings during each supervisory session.
21. The log sheet is an important deliverable for the research and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the documentation, with sheets dated and numbered consecutively.

Student Name: Anuj Bhandari **Group:2** **Date:** 20/03/2025 **Meeting No:** 3

Project Title : BinHero

Supervisor's Name: Sachin Shrestha | **Supervisor's Signature:**

Item for Discussion: (Noted by Student **Before** Supervisory meeting)
- Finance Management App idea pitiching.
- Supervisor feedback on orignality and feasiblity

Record of Discussion: (Noted by Student **During** Supervisory meeting)
- Idea was considered common and lacking innovation
- Suggested focusing on a more impactful or real-time app.

Action List (To be attempted or Completed by Student by the **Next** Supervisory meeting)
- Finance management idea rejected

Note: *A student MUST attend all weekly meetings as scheduled by the supervisor. In the event Supervisor could not be met, the student must immediately inform the Academic Coordinator and supervisor so that a meeting can be subsequently arranged. Under NO situation should a student miss a weekly appointment with the supervisor except during exceptional circumstances.*

Figure B.3: Meeting 3

IIMS
COLLEGE

Bachelors of Computer Science (Hons)
Capstone Project Log Sheet

Notes on use of the project log sheet:

22. Minimum TEN (10) for Capstone/FYP Part 1 and TEN (10) for Capstone 2
23. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
24. A log sheet is to be brought by the STUDENT to each supervisory session.
25. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant sections of the form.
26. The student should make copies of the Log Sheet and must be inserted in the documentation (under appendices) in the softbound report.
27. It is recommended that students bring along log sheets of previous meetings during each supervisory session.
28. The log sheet is an important deliverable for the research and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the documentation, with sheets dated and numbered consecutively.

Student Name: Anuj Bhandari **Group:2** **Date:** 25/03/2025 **Meeting No: 4**

Project Title : BinHero

Supervisor's Name: Sachin Shrestha	**Supervisor's Signature:** Sachin

Item for Discussion: (Noted by Student **Before** Supervisory meeting)
- Personal safery app idea pitched..
- Potential use of GPS, alerts etc.
- Revview of past similar projects.

Record of Discussion: (Noted by Student **During** Supervisory meeting)
- Emphasis on including AI real-time components.
- Encouraged team to focus on somehting novel idea.

Action List (To be attempted or Completed by Student by the **Next** Supervisory meeting)
- Dropped Safety App idea.
- Ideating next topics

Note: A student MUST attend all weekly meetings as scheduled by the supervisor. In the event Supervisor could not be met, the student must immediately inform the Academic Coordinator and supervisor so that a meeting can be subsequently arranged. Under NO situation should a student miss a weekly appointment with the supervisor except during exceptional circumstances.

Figure B.4: Meeting 4

IIMS COLLEGE

Bachelors of Computer Science (Hons)
Capstone Project Log Sheet

Notes on use of the project log sheet:

29. Minimum TEN (10) for Capstone/FYP Part 1 and TEN (10) for Capstone 2
30. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
31. A log sheet is to be brought by the STUDENT to each supervisory session.
32. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant sections of the form.
33. The student should make copies of the Log Sheet and must be inserted in the documentation (under appendices) in the softbound report.
34. It is recommended that students bring along log sheets of previous meetings during each supervisory session.
35. The log sheet is an important deliverable for the research and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the documentation, with sheets dated and numbered consecutively.

Student Name: Anuj Bhandari **Group:**2 **Date:** 01/04/2025 **Meeting No:** 5

Project Title : BinHero

Supervisor's Name: Sachin Shrestha | **Supervisor's Signature:**

Item for Discussion: (Noted by Student Before Supervisory meeting)

- Smart trash disposal app idea pitched..
- Potential use of GPS, notificationetc.
- Supervisor feedback on technical potential.
- Revview of past similar projects.

Record of Discussion: (Noted by Student During Supervisory meeting)

- Emphasis on including AI real-time components.
- Suggested to clearly outline AI+ gamification concepts.
- Proposal writing initiated.

Action List (To be attempted or Completed by Student by the Next Supervisory meeting)

- Proposal writing started.

Note: A student MUST attend all weekly meetings as scheduled by the supervisor. In the event Supervisor could not be met, the student must immediately inform the Academic Coordinator and supervisor so that a meeting can be subsequently arranged. Under NO situation should a student miss a weekly appointment with the supervisor except during exceptional circumstances.

Figure B.5: Meeting 5

IIMS COLLEGE

Bachelors of Computer Science (Hons)
Capstone Project Log Sheet

Notes on use of the project log sheet:

36. Minimum TEN (10) for Capstone/FYP Part 1 and TEN (10) for Capstone 2
37. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
38. A log sheet is to be brought by the STUDENT to each supervisory session.
39. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant sections of the form.
40. The student should make copies of the Log Sheet and must be inserted in the documentation (under appendices) in the softbound report.
41. It is recommended that students bring along log sheets of previous meetings during each supervisory session.
42. The log sheet is an important deliverable for the research and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the documentation, with sheets dated and numbered consecutively.

Student Name: **Anuj Bhandari** **Group:2** **Date:** 08/04/2025 **Meeting No:** **6**

Project Title : **BinHero**

Supervisor's Name: Sachin Shrestha	**Supervisor's Signature:** Sachin

Item for Discussion: (Noted by Student **Before** Supervisory meeting)
- Proposal review and feedback.
- Tech stack confirmations.

Record of Discussion: (Noted by Student **During** Supervisory meeting)
- **Decided to use Pytorch and YOLOv11.**

Action List (To be attempted or Completed by Student by the **Next** Supervisory meeting)
- **Finalize YOLOv11 for object detection from ultralytics**

*Note: A student **MUST** attend all weekly meetings as scheduled by the supervisor. In the event Supervisor could not be met, the student must immediately inform the Academic Coordinator and supervisor so that a meeting can be subsequently arranged. Under NO situation should a student miss a weekly appointment with the supervisor except during exceptional circumstances.*

Figure B.6: Meeting 6

IIMS COLLEGE

Bachelors of Computer Science (Hons)
Capstone Project Log Sheet

Notes on use of the project log sheet:

43. Minimum TEN (10) for Capstone/FYP Part 1 and TEN (10) for Capstone 2
44. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
45. A log sheet is to be brought by the STUDENT to each supervisory session.
46. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant sections of the form.
47. The student should make copies of the Log Sheet and must be inserted in the documentation (under appendices) in the softbound report.
48. It is recommended that students bring along log sheets of previous meetings during each supervisory session.
49. The log sheet is an important deliverable for the research and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the documentation, with sheets dated and numbered consecutively.

Student Name: Anuj Bhandari **Group:**2 **Date:** 23/04/2025 **Meeting No:** 7

Project Title : BinHero

Supervisor's Name: Sachin Shrestha	**Supervisor's Signature:** Sachin

Item for Discussion: (Noted by Student **Before** Supervisory meeting)

- Proposal submision confirmation.
- Dataset search.

Record of Discussion: (Noted by Student **During** Supervisory meeting)

- Supervisors feed back on dataset.

Action List (To be attempted or Completed by Student by the **Next** Supervisory meeting)

- Research on Trash datasets.

__Note:__ A student MUST attend all weekly meetings as scheduled by the supervisor. In the event Supervisor could not be met, the student must immediately inform the Academic Coordinator and supervisor so that a meeting can be subsequently arranged. Under NO situation should a student miss a weekly appointment with the supervisor except during exceptional circumstances.

Figure B.7: Meeting 7

IIMS COLLEGE

Bachelors of Computer Science (Hons)
Capstone Project Log Sheet

Notes on use of the project log sheet:

50. Minimum TEN (10) for Capstone/FYP Part 1 and TEN (10) for Capstone 2
51. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
52. A log sheet is to be brought by the STUDENT to each supervisory session.
53. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant sections of the form.
54. The student should make copies of the Log Sheet and must be inserted in the documentation (under appendices) in the softbound report.
55. It is recommended that students bring along log sheets of previous meetings during each supervisory session.
56. The log sheet is an important deliverable for the research and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the documentation, with sheets dated and numbered consecutively.

Student Name: Anuj Bhandari **Group:2** **Date:** 20/05/2025 **Meeting No: 8**

Project Title : BinHero

Supervisor's Name: Sachin Shrestha | **Supervisor's Signature:**

Item for Discussion: (Noted by Student **Before** Supervisory meeting)

- Faced dataset imbalanced issues.
- Dataset problems faced.
- Introduction of overleaf and LaTeX.

Record of Discussion: (Noted by Student **During** Supervisory meeting)

- Supervisor suggested datasets like taco dataset, Roboflow and Trashnet
- Suggested data augmentation procedures.

Action List (To be attempted or Completed by Student by the **Next** Supervisory meeting)

- **Decided to use taco dataset and Roboflow.**
-

Note: A student MUST attend all weekly meetings as scheduled by the supervisor. In the event Supervisor could not be met, the student must immediately inform the Academic Coordinator and supervisor so that a meeting can be subsequently arranged. Under NO situation should a student miss a weekly appointment with the supervisor except during exceptional circumstances.

Figure B.8: Meeting 8

IIMS COLLEGE

Bachelors of Computer Science (Hons)
Capstone Project Log Sheet

Notes on use of the project log sheet:

57. Minimum TEN (10) for Capstone/FYP Part 1 and TEN (10) for Capstone 2
58. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
59. A log sheet is to be brought by the STUDENT to each supervisory session.
60. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant sections of the form.
61. The student should make copies of the Log Sheet and must be inserted in the documentation (under appendices) in the softbound report.
62. It is recommended that students bring along log sheets of previous meetings during each supervisory session.
63. The log sheet is an important deliverable for the research and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the documentation, with sheets dated and numbered consecutively.

Student Name: **Anuj Bhandari** **Group:2** **Date:** 5/06/2025 **Meeting No: 9**

Project Title : BinHero

Supervisor's Name: Sachin Shrestha | **Supervisor's Signature:**

Item for Discussion: (Noted by Student **Before** Supervisory meeting)
- Motion tracking algorithms discussion..

Record of Discussion: (Noted by Student **During** Supervisory meeting)
- Supervisor suggested to research on DeepSORT and SORT algorithm
- Revied citation formatiing on Overleaf template.

Action List (To be attempted or Completed by Student by the **Next** Supervisory meeting)
- Report writing and refining individual report.
- Utilize DeepSORT and SORT for the project.

Note: A student MUST attend all weekly meetings as scheduled by the supervisor. In the event Supervisor could not be met, the student must immediately inform the Academic Coordinator and supervisor so that a meeting can be subsequently arranged. Under NO situation should a student miss a weekly appointment with the supervisor except during exceptional circumstances.

Figure B.9: Meeting 9

IIMS COLLEGE

Bachelors of Computer Science (Hons)
Capstone Project Log Sheet

Notes on use of the project log sheet:

64. Minimum TEN (10) for Capstone/FYP Part 1 and TEN (10) for Capstone 2
65. The student should prepare for the supervisory sessions by deciding which question(s) he or she needs to ask the supervisor and what progress has been made (if any) since the last session and noting these in the relevant sections of the form, effectively forming an agenda for the session.
66. A log sheet is to be brought by the STUDENT to each supervisory session.
67. The actions by the student (and, perhaps the supervisor), which should be carried out before the next session should be noted briefly in the relevant sections of the form.
68. The student should make copies of the Log Sheet and must be inserted in the documentation (under appendices) in the softbound report.
69. It is recommended that students bring along log sheets of previous meetings during each supervisory session.
70. The log sheet is an important deliverable for the research and an important record of a student's organisation and learning experience. The student **must** hand in the log sheets as an appendix of the documentation, with sheets dated and numbered consecutively.

Student Name: **Anuj Bhandari** **Group:2** **Date :** 16/06/2025 **Meeting No: 10**

Project Title : BinHero

Supervisor's Name: Sachin Shrestha	**Supervisor's Signature:** Sachin

Item for Discussion: (Noted by Student **Before** Supervisory meeting)
- Finalizing indivdual report.
- Supervisor feedback on individual report.

Record of Discussion: (Noted by Student **During** Supervisory meeting)
- Suggested minor formatting and report refinement.
- Logsheets and architecture diagram approval.

Action List (To be attempted or Completed by Student by the **Next** Supervisory meeting)
- Submission of the Final individual report.
- Finished 40% of the model development.

*__Note:__ A student **MUST** attend all weekly meetings as scheduled by the supervisor. In the event Supervisor could not be met, the student must immediately inform the Academic Coordinator and supervisor so that a meeting can be subsequently arranged. Under NO situation should a student miss a weekly appointment with the supervisor except during exceptional circumstances.*

Figure B.10: Meeting 10