

SMART WASTE MANAGEMENT SYSTEM

COMMON INTEGRATED SOLUTION REPORT

Team member	Student ID	Individual component
Jayawardhane K. G. M. S.	IT22213358	ML-Based Municipal Waste Prediction, Composition Estimation and Circular Economy Revenue Hub
Gamaariya C. C.	IT22202772	Smart Fill-Level and Occupancy Monitoring
Hemsara K. D. S.	IT22548528	IoT-Based Environmental Sensing Smart Bin
Wijesiri A. M.	IT22088482	Mobile Application for Real-Time Bin Detection, Adaptive Route Optimization, and Resident-Driven Coordination

Report submitted in partial fulfilment of the requirements for the

B.Sc. (Hons) Degree in Information Technology

Specialized in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

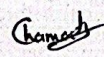
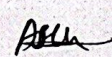
Sri Lanka

October 2026

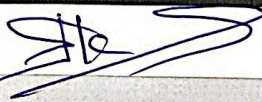
DECLARATION AND SUPERVISOR CERTIFICATION

Candidate declaration. We declare that this common report is our own work and does not incorporate, without acknowledgement, material previously submitted for a degree or diploma at any university or institute of higher learning. To the best of our knowledge, it contains no material previously published or written by another person except where acknowledgement is made in the text. We grant the Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute this report, in whole or in part, in print, electronic, or other media, while retaining the right to use the content in future work.

Academic integrity and AI disclosure. We have disclosed all use of generative AI and other automated tools in accordance with the module policy. We remain responsible for the accuracy, originality, citations, code, data, analyses, and conclusions in this report.

Student name	Student ID	Signature	Date
Jayawardhane K. G. M. S.	IT22213358		09/10/2026
Gamaariya C. C.	IT22202772		09/10/2026
Hemsara K. D. S.	IT22548528		09/10/2026
Wijesiri A. M.	IT22088482		09/10/2026

Supervisor certification. The above candidates carried out this research project under my supervision.

Role and name	Signature	Date
Supervisor: <i>Uditha Dharmakeerthi</i>		10/10/26
Co-supervisor: <i>Kanishka Yapa</i>		

ABSTRACT

Municipal waste management requires current information about bin conditions, advance estimates of collection demand and effective coordination between residents and collection personnel. This project developed a four-component Smart Waste Management System comprising environmental sensing, fill-level and lid-state monitoring, municipal waste forecasting, and mobile collection coordination. ESP32-based units provide bin-condition records, while separate XGBoost models forecast weekly wet and dry waste for four Kalutara Municipal Council zones. Forecasts are converted into fleet, composition and indicative recovery-value estimates. A Flutter application supports pickup requests, critical-bin visibility, and vehicle tracking and arrival notifications.

Evaluation combined prototype sensor tests, a chronological forecasting hold-out and controlled mobile workflows. Environmental telemetry achieved a reported 99.4% delivery rate and 1.84-second mean cloud latency, although alert classification accuracy (96.7%) fell slightly short of its 97% target. On the reported 126-record forecasting test set, dry waste achieved R^2 of 0.9492 and mean absolute error of 1.521 tonnes; wet waste achieved R^2 of 0.8035 and mean absolute error of 3.275 tonnes. Fill-level monitoring was verified through functional boundary and scenario tests. All 12 manual mobile application test scenarios passed, demonstrating critical-bin propagation to the driver map and resident–driver request, tracking and arrival interactions. The project establishes technical feasibility and partial cross-component integration. The forecasting component is not yet connected to bin-level data or the mobile application. Its contribution is a modular architecture that links sensing, forecasting and collection action, and separates measured results from predictions and assumptions. A common field pilot is required to establish municipal service, economic and sustainability benefits.

Keywords: smart waste management, Internet of Things, waste forecasting, collection coordination, decision support

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to our supervisor and co-supervisor at the Department of Information Technology, Sri Lanka Institute of Information Technology (SLIIT), for their guidance, continuous support and constructive feedback throughout this research. Their advice helped shape the technical direction and academic quality of our work. We deeply appreciate their encouragement and patient guidance during the development, evaluation and thesis-writing stages.

We would also like to thank the Department of Information Technology at SLIIT for providing the facilities, learning resources and academic support needed to carry out this project. The knowledge and practical experience gained throughout our undergraduate studies provided a valuable foundation for designing and developing the Smart Waste Management System.

We gratefully acknowledge the commitment, cooperation and contributions of every member of our research team. Working together on environmental sensing, fill-level and lid-state monitoring, waste forecasting, and mobile collection coordination strengthened our understanding of how these components can support municipal waste management. Our discussions, shared problem-solving efforts and work on component integration helped us address technical challenges throughout the project. The willingness of all team members to exchange ideas, provide feedback and support one another made this final-year research experience both productive and rewarding.

TABLE OF CONTENTS

1. INTRODUCTION AND PROJECT CONTEXT	1
1.1 Background and Context.....	1
1.2 Stakeholders and User Needs.....	2
1.3 Alignment with the Sustainable Development Goals	4
1.4 Purpose and Scope of the Common Report	5
2. OVERALL PROBLEM DEFINITION AND FOUR SUBPROBLEMS	7
2.1 Overall Problem Definition.....	7
2.2 Root Causes and Overall Research Gap.....	8
2.3 Subproblem and Component Mapping	9
2.4 Overall Aim and Group Objectives	10
2.5 Scope Assumptions and Constraints	11
3. OVERALL SOLUTION STRUCTURE.....	13
3.1 Integrated Solution Overview	13
3.2 High Level Architecture.....	14
3.3 Component 1 Summary.....	15
3.4 Component 2 Summary.....	16
3.5 Component 3 Summary.....	17
3.6 Component 4 Summary.....	19
3.7 End to End User and Data Flow.....	20
4. COMPONENT INTEGRATION AND COHESIVE OPERATION.....	22
4.1 Integration Strategy.....	22
4.2 Interfaces Data Contracts and Shared Models	23
4.3 Shared Nonfunctional Requirements	24
4.4 Integrated Workflow and Exception Handling	25
4.5 Team Contribution and Dependency Matrix.....	26
5. INTEGRATED IMPLEMENTATION AND PROJECT EXECUTION	28
5.1 Common Technology Stack and Infrastructure	28
5.2 Development and Integration Process.....	29
5.3 Deployment and Operational Configuration.....	30
5.4 Evidence of Team Execution	31
6. INTEGRATED TESTING EVALUATION AND VALIDATION	32
6.1 Evaluation Framework and Success Criteria	32
6.2 Integration and End to End Testing	33
6.3 Overall Results.....	34
6.4 Stakeholder and User Validation	36
6.5 Achievement of Common Objectives	37

6.6 Integrated Limitations and Threats	39
7. COMMERCIALISATION SUSTAINABILITY AND RESPONSIBLE INNOVATION	41
7.1 Value Proposition and Beneficiaries	41
7.2 Market Alternatives and Differentiation	41
7.3 Business Model and Feasibility	43
7.4 Adoption Intellectual Property and Compliance.....	44
7.5 Sustainable Development Goal Contribution	45
8. CONCLUSION AND FUTURE DIRECTIONS	46
8.1 Overall Conclusion.....	46
8.2 Principal Integrated Contribution.....	46
8.3 Future Directions.....	47
REFERENCES.....	49
APPENDICES	51
Appendix A Integrated Interface Specification.....	51
A.1 Identity and Location Registry	51
A.2 Environmental Telemetry Contract.....	52
A.3 Fill and Lid Record Contract.....	53
A.4 Operational Critical Bin Event.....	53
A.5 Forecast Service Contract	54
A.6 Pickup and Collection Records	55
A.7 Common Acceptance Scenarios.....	56
A.8 Evaluation Log and Evidence Reconciliation	57
Appendix B Team Contribution Declaration	59
B.1 Academic Integrity and AI Use Disclosure.....	59

LIST OF FIGURES

Figure 3.1 System Architecture and Current Integration Boundaries 14

Figure 3.2 Collection Workflow and Forecast Integration Extension..... 21

LIST OF TABLES

Table 1.1 Information Needs Across the Collection Cycle	2
Table 1.2 Stakeholders and Evidence of Relevant Capabilities	3
Table 2.1 Subproblem and Component Mapping	9
Table 2.2 Common Objectives and Evaluation Evidence	10
Table 4.1 Interfaces and Current Evidence Status	23
Table 4.2 Exception Handling Requirements for Cohesive Operation	25
Table 4.3 Team Contributions and Dependencies.....	26
Table 5.1 Technology Stack by System Responsibility	28
Table 6.1 Evaluation Areas and Acceptance Basis	32
Table 6.2 Integration Scenarios and Evidence Status	34
Table 6.3 Environmental Results Against Stated Criteria.....	35
Table 6.5 Forecast Performance on the Reported 126 Test Records.....	36
Table 6.6 Achievement of the Common Objectives	37
Table 7.1 Alternatives and the Intended Combined Service	41
Table 7.2 Illustrative Gross Value per Tonne of Dry Waste	43
Table A.1 Recommended Shared Registry Fields.....	51
Table A.2 Environmental Telemetry Fields and Interpretation.....	52
Table A.3 Fill and Lid Field Mapping	53
Table A.4 Recommended Operational Event Contract	54
Table A.5 Recommended Forecast and Planning Record	55
Table A.6 Recommended Collection Audit Fields	56
Table A.7 Common Acceptance Test Matrix.....	56
Table B.1 Individual Research and Product Contributions	59

LIST OF ABBREVIATIONS

Abbreviation	Meaning
API	Application Programming Interface
C1–C4	Components 1 to 4
DHT22	Digital temperature and humidity sensor module
DSR	Design Science Research
ESP32	Espressif 32-bit microcontroller family
GPS	Global Positioning System
HC-SR04	Ultrasonic distance sensor module
HTTPS	Hypertext Transfer Protocol Secure
IoT	Internet of Things
JSON	JavaScript Object Notation
LKR	Sri Lankan rupee
MAE	Mean absolute error
MAPE	Mean absolute percentage error
ML	Machine learning
MQ-4 / MQ-135	Metal-oxide gas sensor modules
OSRM	Open Source Routing Machine
OTA	Over-the-air update
ppm	Parts per million
R ²	Coefficient of determination
REST	Representational State Transfer
RMSE	Root mean squared error
SDG	Sustainable Development Goal
SPIFFS	SPI Flash File System
UI	User interface
UTC	Coordinated Universal Time
XGBoost	Extreme Gradient Boosting

1. INTRODUCTION AND PROJECT CONTEXT

1.1 Background and Context

Municipal waste management brings together collection, transport, processing and communication with the public. Each activity relies on timely information. Collection personnel need to know which containers require attention, supervisors need to estimate the load that vehicles will carry, and residents need a practical way to request collection and follow service activity. We developed the Smart Waste Management System to address these needs through four complementary research components: environmental sensing, fill-level and lid-state monitoring, municipal waste prediction, and mobile collection coordination.

The project combines observation at individual bins with planning at municipal scale. At the bin, environmental sensors collect gas-related readings, temperature and humidity, while an ultrasonic sensor estimates occupied volume and a reed switch identifies lid state. At municipal scale, separate machine-learning models forecast weekly wet and dry waste for four Kalutara Municipal Council zones. The mobile application brings collection information to residents, drivers and administrators through requests, maps, tracking and notifications. These functions cover immediate bin condition, active collection activity and advance planning.

Internet of Things (IoT) systems provide a practical foundation for this approach because sensors can send structured measurements to cloud services for storage and use by applications. Sosunova and Porras describe the roles of sensing, communication, data processing and stakeholders in smart waste-management systems [1]. Zoumpoulis et al. examine smart-bin capabilities in relation to resource recovery and sustainable urban waste practices [2]. Our work applies these ideas to a locally relevant system with documented hardware, software, interfaces and evaluation procedures.

Kalutara Municipal Council provides the operating context for the forecasting component. The selected collection zones have different residential, commercial and coastal characteristics, and their records distinguish wet and dry waste. Separating these streams is useful because they have different generation patterns and recovery

routes. Wet waste informs composting and organic-treatment planning, while dry waste informs recycling and material-recovery planning. Forecasting both streams provides more useful information than a single total-tonnage estimate.

The common project follows a modular structure. Each component performs a defined task and produces information for a particular decision. Environmental status describes the condition around waste, fill status describes occupied bin volume, predictions describe expected weekly tonnage, and mobile records describe requests and vehicle activity. Table 1.1 summarises these decision horizons. This structure lets us relate the four components while retaining the units, meaning and purpose of their outputs.

Table 1.1 Information Needs across the Collection Cycle

Decision horizon	Required information	Primary response
Immediate bin condition	Environmental status, fill class and freshness	Review condition and prioritize appropriate attention
Active collection trip	Critical-bin locations, pickup requests and driver position	Coordinate movement and collection interactions
Weekly municipal planning	Wet and dry tonnage by zone and available fleet capacity	Plan trips, crews and processing requirements
Resource recovery planning	Estimated material quantities, recovery assumptions and prices	Evaluate possible recovery routes and gross value

1.2 Stakeholders and User Needs

Municipal supervisors and logistics personnel need advance estimates that can be translated into collection capacity. The prediction dashboard provides wet, dry and total waste estimates for a selected week, together with trip calculations, material breakdowns, recovery values and recommendations. The same dashboard includes model-validation and prediction-history views, so supervisors can examine both the forecast and the recorded performance of the models. The planning interface is designed around the practical questions of how much waste to expect, how many trips to allocate and which recovery routes to prepare.

Collection drivers need accessible information while operating in the service area. The mobile application displays critical bins and resident requests on a map, supports

collection-point ordering and presents road-route geometry. Driver-location updates allow the resident interface to show relevant vehicles. This makes the application a coordination tool throughout an active collection workflow, from identifying demand to approaching the pickup point.

Residents need a convenient way to communicate collection requirements and receive useful service information. The resident interface supports registration, pickup requests, truck tracking, notifications, reports and announcements. The pickup service uses a one-to-three-hour request window, and an arrival notification is generated when the driver reaches the configured proximity range. These functions provide a direct relationship between a resident's request and the collection activity visible on the application.

Facility managers and environmental officers need information about the condition of waste containers. The environmental unit supplies gas-related and ambient readings together with a GREEN, YELLOW or RED status. The fill unit supplies a percentage and an OK, WARNING or CRITICAL class, as well as lid state. Local indicators and cloud records give operators two ways to inspect the same bin condition: at the installation and through the application or database.

Material-recovery facilities and recycling partners use advance estimates to plan labour, storage and expected inputs. The forecast service translates predicted tonnage into composition and recovery-route estimates using configured shares, yields and rates. Administrators support the continuity of the service through account-management functions, while technical maintainers manage device settings, calibration, connectivity and software configuration. Table 1.2 relates these stakeholder roles to the project functions.

Table 1.2 Stakeholders and Evidence of Relevant Capabilities

Stakeholder	Principal need	Project function
Municipal supervisor	Weekly demand and capacity planning	Forecast dashboard, trip estimates and prediction history
Collection driver	Visible collection points and route support	Critical-bin map, pickup requests and road-route display

Resident	Convenient requests and service visibility	Pickup request, live tracking and arrival notification
Facility or environmental officer	Current bin condition	Environmental readings, fill class, lid state and local indicators
Recovery facility or recycler	Expected incoming materials	Composition estimates and recovery-route calculations
Administrator and maintainer	Reliable account and device operation	User management, recovery workflows and device configuration

1.3 Alignment with the Sustainable Development Goals

The project is aligned with Sustainable Development Goals 11, 12 and 13 in the United Nations 2030 Agenda [3]. These goals connect the technical functions of the system with urban service quality, responsible resource use and environmental planning. The relationship is expressed through the activities supported by the project: observing waste conditions, coordinating collection, planning capacity and identifying material-recovery opportunities.

SDG 11 concerns sustainable cities and communities. Our bin-monitoring functions give operators more current information, while the mobile application supports communication between residents and collection drivers. The critical-bin workflow turns a sensor-derived condition into a visible collection point on the driver's map. The forecasting component complements this immediate response by supplying zone-level estimates for weekly planning. Together, these capabilities support a more informed approach to municipal waste collection.

SDG 12 concerns responsible consumption and production. The prediction service estimates waste composition and allocates predicted wet and dry loads to recovery routes. These include composting, energy recovery, plastics, paper, glass and metals. Presenting material quantities and indicative values makes recovery a visible part of the planning process. It helps supervisors consider waste as a set of material streams with different handling requirements and possible uses.

SDG 13 concerns climate action. Collection planning and organic-waste diversion are relevant to transport activity and disposal-related environmental impacts. The project

supports these decisions through trip estimates, route assistance and recovery planning. Environmental sensing also provides a record of changing conditions around waste, which can help operators identify situations needing attention. The sustainability value therefore arises from informed operational decisions and the ability to examine data over time.

We treat these SDGs as a framework for the intended contribution of the project. Section 7.5 explains how the sensing, forecasting, recovery and coordination functions support the corresponding goals, while Chapter 8 identifies directions for extending their use in municipal operations. The common report connects the goals to concrete system functions rather than using them only as general statements of motivation.

1.4 Purpose and Scope of the Common Report

This report presents the common Smart Waste Management System developed by Research Group R26-IT-126. We bring together the problem definitions, methods, implementations and findings of the four component studies under one solution structure. The report explains how the components contribute to the overall waste-management process, how information moves through their interfaces, and what the evaluation demonstrated.

The technical scope includes ESP32-based environmental monitoring, ultrasonic fill estimation, lid-state monitoring, cloud storage, weekly wet and dry forecasting, fleet and recovery calculations, and mobile collection workflows. The forecasting study uses Kalutara Municipal Council records from four collection zones. The sensing studies use prototype installations and controlled scenarios. The mobile study uses resident, driver and administrator workflows with practical functional and integration testing.

The common scope also includes the architecture, technology stack, team responsibilities, deployment configuration and business considerations. We describe operational sensor-to-cloud and critical-bin-to-mobile paths, together with the forecasting service's connection to its models, weather inputs, database and dashboard. These interfaces place each component within the project and show how its output supports a particular user decision.

Chapter 2 presents the overall problem and four subproblems. Chapters 3 and 4 explain solution structure and cohesive operation. Chapter 5 describes implementation and execution, and Chapter 6 presents evaluation and findings. Chapters 7 and 8 discuss commercialisation, sustainability, conclusions and future directions. Appendix A records the principal interfaces and technical parameters; Appendix B records the contributions of the team.

2. OVERALL PROBLEM DEFINITION AND FOUR SUBPROBLEMS

2.1 Overall Problem Definition

The overall problem is the difficulty of making coordinated waste-management decisions when current bin conditions, future waste demand and collection activity are handled separately. Manual inspection provides a limited view of changing bin conditions. Historical municipal totals provide useful records but require analysis before they can guide advance planning. Residents and drivers also need communication that follows the progress of a collection request.

This problem affects different decisions within the same service. A facility manager may need to inspect a container because it is nearly full or because environmental readings have increased. A supervisor may need to allocate collection trips for the following week. A driver may need to combine critical bins with active resident requests, while a resident may need to know which vehicle is approaching. These decisions require related information, but the information has different units and update rates.

Our solution addresses the problem through complementary functions. The environmental unit records gas-related and ambient conditions. The fill unit measures the available space in a bin and identifies its lid state. The prediction service estimates weekly wet and dry tonnage, and the mobile application supports request and vehicle coordination. The resulting structure covers observation, planning and action within one research project.

The common problem also includes the organisation of information. Sensor records require a bin identifier and meaningful status. A municipal forecast requires a zone and target week. A pickup request requires a location, time window and user reference. Consistent records and defined interfaces allow these functions to be understood within the same service process. Chapter 4 explains the corresponding data contracts and operational workflows.

2.2 Root Causes and Overall Research Gap

The first root cause is limited observation of conditions at the bin. Manual inspection and fixed collection schedules provide intermittent information, while fill-only monitoring does not characterise the environmental state of decomposing waste. Environmental sensing adds gas-related, temperature and humidity information, and fill monitoring adds a direct indication of occupied volume. The two signals describe different properties and should be retained separately.

The second root cause is a separation between historical records and operational planning. Weekly waste records are useful after collection, but a council must allocate vehicles and processing resources before the next collection period. The forecasting component addresses this gap by combining historical wet and dry tonnage with rainfall and other contextual features. It converts forecasts into trip and recovery estimates, while keeping supervisors responsible for the final decision.

The third root cause is weak coordination during collection. A resident may need to know whether a truck is approaching, and a driver may need to identify critical bins and active requests together. The mobile component links these interactions through requests, map information, live vehicle updates and arrival notifications. Its contribution is the integration of established capabilities into a tested operational workflow, rather than novelty in global positioning or map technology.

A further cause appears at the interfaces. The sensing records use different status names and timestamp conventions; the mobile application consumes a binary critical/normal event; and the forecasting service operates on zone-week tonnage. These differences can create incorrect joins, stale displays or misleading interpretations unless they are explicitly managed. A shared bin registry, clear time semantics and a distinction between bin-level and zone-level information are therefore central integration requirements.

The overarching research gap is the practical combination of these information sources into a coherent waste-management decision process. The literature reviews in the theses support the relevance of sensing, forecasting and mobile support, but do not justify claiming that no existing system combines them. The contribution is instead a

localised, modular implementation and an assessment of the evidence and remaining dependencies needed for cohesive operation.

2.3 Subproblem and Component Mapping

The four subproblems are complementary and have distinct owners. Component 1 addresses insufficient visibility of the environmental condition inside a bin. Component 2 addresses fill-state visibility and the behavioural context associated with lid activity. Component 3 addresses uncertainty in future municipal waste load and recovery planning. Component 4 addresses the practical coordination of residents, drivers and current collection demand. Table 2.1 maps these subproblems to the individual work.

Table 2.1 Subproblem and Component Mapping

Component and owner	Subproblem	Primary output
C1 Environmental sensing - Hemsara K. D. S. (IT22548528)	Bin environmental conditions are not visible in fill-only monitoring	Timestamped environmental telemetry and GREEN/YELLOW/RED status
C2 Fill and occupancy - Gamaariya C. C. (IT22202772)	Fill state and lid behaviour are not consistently represented	Fill class, lid state and critical-status transition
C3 Prediction and recovery - Jayawardhane K. G. M. S. (IT22213358)	No localised advance estimate links municipal load to planning	Weekly wet/dry forecasts, trip estimates and indicative recovery value
C4 Mobile coordination - Wijesiri A. M. (IT22088482)	Residents and drivers lack one coordinated collection workflow	Pickup requests, critical-bin map, tracking and arrival information

The sensing components operate at the scale of an individual bin. They supply current readings and statuses, and their records can be retained as a history of changing conditions. The forecasting component operates at the zone-week scale and uses municipal records to estimate tonnage. The mobile component operates at the scale of requests, collection points and vehicle locations. These scales are complementary because they correspond to different decisions in waste management.

The principal cross-component collection interaction is the critical-bin event. A status change produced by the fill-monitoring path is sent to the collection-service backend

and represented in the driver application. The prediction service supports a separate planning workflow through its dashboard. Environmental and fill histories provide a structured monitoring resource, while prediction outputs supply advance estimates. This mapping establishes the role of each component in the common architecture.

We use the component numbering consistently throughout the report. Occupancy in Component 2 refers to occupied bin volume and the context supplied by lid activity. The route function in Component 4 is described as route support: it orders points with a nearest-neighbour method and obtains road geometry from OSRM. These terms reflect the functions developed and evaluated in the individual work.

2.4 Overall Aim and Group Objectives

The overall aim is to develop and evaluate a modular smart waste-management solution that combines bin-condition monitoring, municipal waste forecasting and mobile collection coordination to support informed collection and resource-recovery decisions. The aim is expressed as decision support because the system does not replace operator judgement or independently certify the safety of waste-handling conditions.

The group objectives are defined in Table 2.2. Each objective identifies an observable capability and the evidence required to assess it. This mapping makes the evaluation explicit: developing a data-producing component is different from proving that another component uses its data, and demonstrating a route display is different from proving a reduction in collection distance.

Table 2.2 Common Objectives and Evaluation Evidence

ID	Objective	Evidence required
O1	Acquire and publish interpretable bin-condition information	Environmental telemetry, fill/lid records and calibration or functional tests
O2	Turn critical-bin events into collection-side visibility	Traceable condition-to-backend-to-driver-map scenario
O3	Forecast weekly wet and dry municipal waste	Chronological hold-out metrics and zone-level error analysis

O4	Convert forecasts into useful planning estimates	Working trip, composition and revenue calculations with assumptions
O5	Coordinate resident requests and driver activity	Request propagation, tracking, nearest-truck and arrival scenarios
O6	Connect monitoring, prediction and collection through consistent interfaces	Shared identifiers, implemented adapters and end-to-end cross-component logs
O7	Assess readiness for responsible field adoption	Stakeholder evaluation, operational baseline and reconciled evidence package

The objectives cover three types of work. Research activities include sensor evaluation and forecast-performance analysis. Engineering activities include firmware, APIs, databases and interfaces. Operational activities include translating bin status into driver information and supporting resident–driver interaction. Treating these activities together provides a common account of what the group developed and how the results relate to its aim.

We also use the objectives to organise the evaluation in Chapter 6. Bin-condition tests address O1, the critical-bin path addresses O2, the forecasting test set addresses O3, and dashboard calculations address O4. Mobile scenarios address O5. The architecture and contribution records address O6, while Chapter 7 addresses O7. This gives the report a consistent relationship between problem, implementation and result.

2.5 Scope Assumptions and Constraints

The environmental and fill components are prototype systems using ESP32 controllers and Wi-Fi connectivity. Their installation depends on sensor placement, suitable power and calibration for the physical bin. The environmental unit records gas-related readings, temperature and humidity. The fill unit uses the measured distance to the waste surface and calibrated empty and full distances. These choices define the practical operating conditions of the sensing work.

The forecasting scope is four Kalutara Municipal Council zones and 628 weekly zone-level records. The dataset contains calendar, rainfall, zone, area-type, population-

density, and event and lagged-tonnage information. We use 502 records for training and 126 for testing in chronological order. The fleet-planning example uses a five-tonne truck capacity and three vehicles. Material shares, recovery yields and benchmark rates are configuration values used to calculate planning estimates.

The mobile scope includes resident, driver and administrator roles. A resident request contains location and timing information, while driver-position records support tracking and proximity functions. The pickup service uses a one-to-three-hour request window. Route support uses nearest-neighbour ordering based on Haversine distance and OSRM road geometry. Its evaluation uses controlled device movement and predefined manual scenarios.

These assumptions establish a workable development and evaluation setting. Bin geometry affects fill measurement; weather availability affects prediction inputs; and network and location services affect the application workflow. We describe how the implementations handle these dependencies in Chapters 4 and 5. The common evaluation reports sensor measurements, model errors and observed workflow behaviour within this defined scope.

3. OVERALL SOLUTION STRUCTURE

3.1 Integrated Solution Overview

The Smart Waste Management System is organised into observation, planning and collection layers. The observation layer contains environmental sensing and fill/lid monitoring. The planning layer contains the municipal prediction service and its decision functions. The collection layer contains the mobile application and the backend services used for requests, vehicle locations, accounts and critical-bin events. Together, these layers cover the principal information needs described in Chapter 1.

Component 1 collects readings from MQ-135, MQ-4 and DHT22 sensors. The ESP32 processes the measurements and produces a multi-parameter environmental status. Component 2 measures ultrasonic distance, calculates fill percentage, reads the lid switch and publishes bin records. Its critical-status transition provides an operational signal for the collection-service backend. The sensor functions therefore create both measurements and interpretable statuses.

Component 3 combines municipal records with automated rainfall retrieval and separate wet and dry XGBoost models. The service converts predictions into total loads, trips, material composition and recovery values. A React dashboard gives supervisors a planning interface with forecasts, backtesting, recommendations and history. This layer supplies an advance view of municipal demand.

Component 4 presents collection information to residents, drivers and administrators. It consumes critical-bin events through the collection backend, displays requests on maps, shares vehicle positions and generates proximity-based arrival information. Nearest-neighbour ordering and OSRM geometry support driver movement between collection points. The mobile functions provide the operational interaction needed to use current collection information.

The architecture is modular because the input, timescale and responsibility of each function differ. Sensor cloud records support monitoring, the prediction database supports planning runs, and application records support live user workflows. Defined interfaces let these functions contribute to the common service while retaining their own data models and development environments.

3.2 High Level Architecture

Figure 3.1 shows the main component boundaries, cloud services, users and data paths. At the edge, two ESP32-based functions observe environmental and physical bin conditions. Firebase Real-time Database stores sensing records as a live and historical data resource. The collection API receives status changes and updates operational records used by the mobile application.

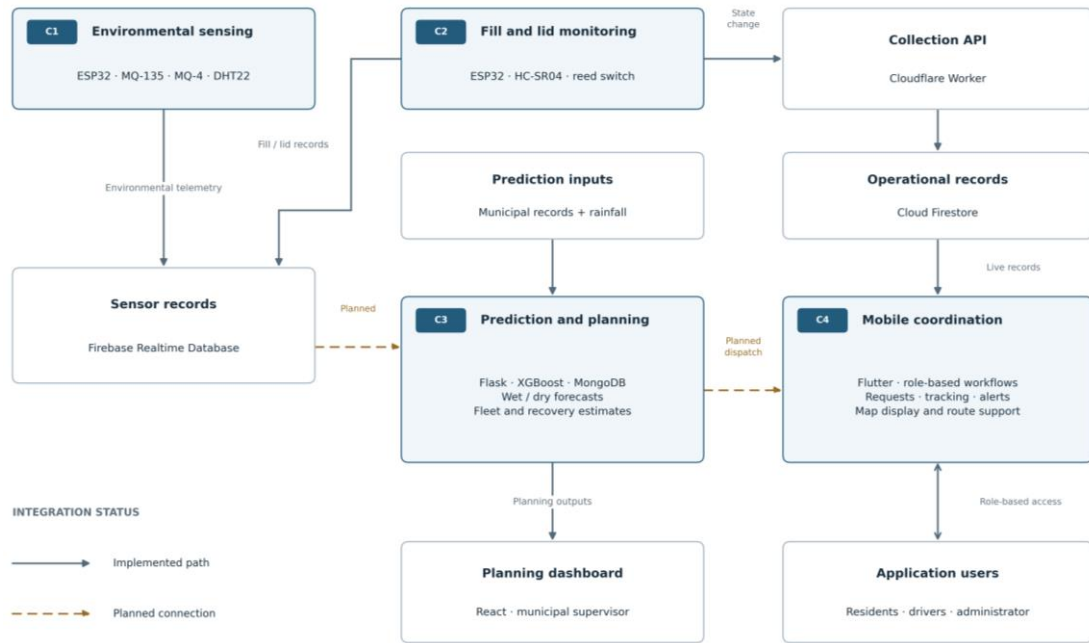


Figure 3.1 System Architecture and Current Integration Boundaries

Firebase Realtime Database provides JSON-based storage and synchronisation [4]. Cloud Firestore provides document-based application records and real-time listeners [5]. Cloudflare Workers supplies the collection-service API and supporting backend functions [6]. In our architecture, sensor records and application records have distinct purposes: the first preserves readings, while the second supports requests, users, vehicles and operational events.

The prediction boundary includes the municipal dataset, preprocessing pipelines, trained models, rainfall retrieval, Flask API and MongoDB Atlas. The dashboard communicates with this service to obtain predictions and planning outputs. Municipal supervisors use the dashboard to select a week, inspect the results and review previous

runs. This boundary also includes the backtesting interface used to compare predictions withheld-out observations.

The mobile boundary includes Flutter application logic, role-based navigation, authentication, Firestore streams, location services and map presentation. Drivers use critical-bin and request information, residents use request and tracking functions, and administrators use account-management functions. External map and routing services provide geographic presentation and road geometry.

Solid connections in Figure 3.1 represent component services and the operational bin-to-mobile workflow. Dashed connections identify the architecture's extension paths between sensor histories, forecasting and mobile planning. The design provides a common view of the project while keeping current measurement, weekly estimation and live collection interaction as distinct information flows. The interfaces described in Chapter 4 explain the records used at these boundaries.

3.3 Component 1 Summary

Component 1 is the IoT-Based Environmental Sensing Smart Bin developed by Hemsara K. D. S. Its purpose is to provide an environmental view of waste containers through gas-related readings and ambient measurements. The sensing unit combines an ESP32, MQ-135, MQ-4 and DHT22. Espressif's Arduino ESP32 core provides the firmware-development foundation for the controller [7]. The prototype uses Wi-Fi communication to publish structured records to Firebase Realtime Database.

The MQ-135 provides a gas-related air-quality indication, while the MQ-4 provides a methane-related indication. DHT22 measures temperature and relative humidity. These measurements complement fill-level information because they describe conditions around the waste rather than the occupied volume of the container. The controller processes the readings and evaluates a threshold matrix to produce GREEN, YELLOW or RED status.

The firmware has six principal layers: sensor abstraction, data processing, threshold classification, JSON serialisation, communication, and watchdog/recovery. Sensor functions initialise and read the devices. Processing applies calibration, moving-average filtering and unit conversion. The classification function chooses the highest

severity among the status-determining measurements. The communication layer manages Wi-Fi and message dispatch, while recovery functions support reconnection and unattended operation.

The environmental threshold design uses MQ-135 bands below 35 ppm, 35–100 ppm and above 100 ppm; MQ-4 bands below 1,000 ppm, 1,000–5,000 ppm and above 5,000 ppm; and temperature bands below 45 °C, 45–60 °C and above 60 °C. Humidity is retained as environmental context and for compensation. These are prototype monitoring settings used to classify the observed condition.

The payload contains bin identity, message identity, timestamp, location, the sensor values and status. Current readings, history, alert transitions and threshold configuration are organised under the bin hierarchy in the cloud database. Routine telemetry is sent at sixty-second intervals, with immediate reporting when status changes. This provides a current indication together with a record suitable for inspection over time.

Evaluation included calibration, functional status tests, communication tests and continuous operation. Reported results include 99.4% payload delivery, 1.84-second mean cloud latency, 0.82 °C temperature RMSE and 2.74 percentage-point humidity RMSE. A thirty-day run produced a history with 2.1% missing readings. The component contributes a documented environmental-sensing unit, its firmware and calibration method, and structured cloud telemetry for the common project.

3.4 Component 2 Summary

Component 2 is Smart Fill-Level and Occupancy Monitoring developed by Gamaariya C. C. It measures the space occupied in a bin and records whether its lid is open or closed. The prototype uses an ESP32, HC-SR04 ultrasonic sensor and magnetic reed switch. Three bins were used in the study, with an empty distance of thirty centimetres and a full reference of two centimetres.

The ultrasonic sensor measures the time taken for an emitted pulse to return from the waste surface. The controller converts this echo time into distance and then into fill percentage using the calibrated empty and full references. The reported fill is limited to the range from zero to one hundred per cent. Classification uses OK below 80%,

WARNING from 80% to 94%, and CRITICAL from 95% to 100%. Green, yellow and red indicators present the class locally.

The reed switch supplies a physical lid-state input. The prototype reads the switch in its main loop and records OPEN or CLOSED alongside the fill record. A React web page displays lid state. The firmware posts distance, fillLevel, binStatus, lidStatus and a device-uptime timestamp to Firebase Realtime Database. The documented prototype loop samples approximately every five seconds.

The critical-bin path communicates a change of collection status to the collection-service Worker. This makes the sensing result available as a location-based event for the mobile application. The cloud record preserves the measured and classified values, while the collection event represents the operational critical/normal state used by drivers. The two records support monitoring and collection coordination respectively.

The component design also specifies five rule-based interpretations of lid and fill activity: a large fill increase during a lid opening, a rapid increase without a lid event, frequent lid activity, an extended open lid and rapidly increasing critical fill. These rules provide an interpretable basis for extending activity analysis. The common implementation account centres on the prototype's distance measurement, fill classification, lid-state display, cloud records and critical-state communication.

Functional evaluation examined calibrated distances, empty and full conditions, the boundaries at 79%, 80%, 94% and 95%, lid states and cloud data exchange. The boundary tests produced the expected status classes. The web page displayed OPEN and CLOSED, and stored records could be retrieved for review. These results establish the component's practical monitoring contribution within the common system.

3.5 Component 3 Summary

Component 3 is the ML-Based Municipal Waste Prediction, Composition Estimation and Circular Economy Revenue Hub developed by Jayawardhane K. G. M. S. It forecasts weekly wet and dry waste for Kalutara North, Kalutara South, Katukurunda 1 and Katukurunda 2. The service uses municipal records and contextual features to supply advance information for collection and recovery planning.

The dataset contains 628 zone-level weekly observations. Features include month, zone, area type, population density, week type, special-event status, seven-day rainfall and the previous week's wet and dry tonnage. The targets are wet-waste tonnes and dry-waste tonnes. We use separate XGBoost regression pipelines because the two waste streams have different generating processes and patterns. XGBoost provides the tree-boosting method described by Chen and Guestrin [8].

The models use a learning rate of 0.05 and one hundred estimators. The wet-waste model uses maximum depth three, while the dry-waste model uses depth five. Categorical inputs are one-hot encoded within the pipeline, with unknown categories ignored at inference. Numeric inputs are passed through. The training and test sets are ordered chronologically, with 502 training records and 126 test records.

Rainfall is resolved automatically through four tiers: the historical dataset, the Open-Meteo forecast/past-days service, the historical archive, and a monthly climatological mean. Open-Meteo provides rainfall and precipitation variables through its weather API [9]. The hierarchy allows the prediction workflow to obtain a rainfall value appropriate to the requested week. The rainfall source forms part of the interpretation of the planning input.

The Flask service retrieves the input values, runs both models, applies the planning functions and stores the prediction in MongoDB Atlas. The React dashboard displays zone and council totals, five-tonne truck-trip estimates, trips per vehicle, capacity utilisation, composition, recovery values and recommendations. It also provides actual-versus-predicted comparisons and a history of earlier runs.

On the 126-record test set, dry waste achieved R^2 of 0.9492, MAE of 1.521 tonnes and RMSE of 2.273 tonnes. Wet waste achieved R^2 of 0.8035, MAE of 3.275 tonnes and RMSE of 4.826 tonnes. The different results show the value of evaluating the streams separately. The component contributes trained pipelines, automated rainfall retrieval, a planning API and a dashboard that translates forecasts into municipal decision information.

3.6 Component 4 Summary

Component 4 is the mobile application for real-time bin detection, route support and resident-driven coordination developed by Wijesiri A. M. It provides resident, driver and administrator functions in a Flutter/Dart application. Its approximately thirty routes/screens cover authentication, pickup requests, maps, tracking, reports, notifications, account management and recovery.

Resident functions include creating a pickup request, viewing relevant collection vehicles, following movement, receiving arrival information and using report and announcement features. A request contains the resident reference, coordinates, creation time, expiration time and status. The one-to-three-hour pickup window gives the request a defined operating period. Driver functions include critical-bin visibility, active requests, and route support and collection interactions.

Firebase Authentication provides the account-authentication service [10]. Cloudflare Workers supports protected operations and OTP-related backend processing, while Firestore stores application records and supplies live streams. Firebase Cloud Messaging provides the notification infrastructure [11]. Provider manages application state and GoRouter manages role-based navigation. These services connect the interface to user, request and vehicle data.

Route support uses a greedy nearest-neighbour method based on Haversine distance. Starting from the driver's position, the application selects nearby remaining collection points to create an ordering. OSRM then supplies road geometry for the ordered coordinates. Its Route service finds road routes between coordinates in the supplied order [12]. Flutter_map and OpenStreetMap provide the map display, while a configured route-deviation threshold supports rerouting.

Tracking uses driver-location updates and a proximity condition for arrival. The resident-side nearest-truck display can change as relative vehicle positions change. Manual evaluation used resident and driver devices, physical movement to simulate a vehicle, and a two-driver scenario. All twelve principal manual test scenarios were recorded as passing, including pickup visibility, tracking, arrival, and critical-bin display and account-management workflows.

The component contributes the application and the operational interaction between IoT-derived collection status and user-facing collection activity. Its critical-bin integration was demonstrated by a red indication on the driver's map. The resident–driver workflow demonstrated request creation, movement visibility, nearest-vehicle changes and an arrival notification in one coordinated application process.

3.7 End to End User and Data Flow

The bin-triggered flow begins with the fill-monitoring unit. The controller measures the distance to the waste surface, calculates fill percentage and assigns a status class. When the collection state changes, it reports the event to the collection-service Worker. The backend stores the operational state, and the driver application receives the record through the application data service. The result is a visible critical-bin point that the driver can inspect.

The resident-triggered flow begins with an authenticated resident creating a pickup request. The request supplies a location and time window. It becomes visible to the driver, who can use the map and route-support functions. Driver movement updates the location record, allowing the resident interface to follow the relevant vehicle and update its nearest-truck display. Entry into the arrival range produces the corresponding notification.

The planning flow begins when a supervisor selects a target week in the prediction dashboard. The service obtains rainfall and previous-week tonnage, builds model inputs and runs the wet and dry pipelines. It then calculates zone and council totals, vehicle-trip requirements, composition and recovery estimates. The prediction is stored in MongoDB and returned to the dashboard with the derived planning information.

Figure 3.2 presents the collection paths and the planning extension within the common architecture. The two operational collection paths converge in the driver interface. The forecast service contributes a separate advance-planning view, with zone-week outputs suitable for subsequent coordination with collection planning. Each flow therefore has a clear trigger, data transformation and user-facing result.

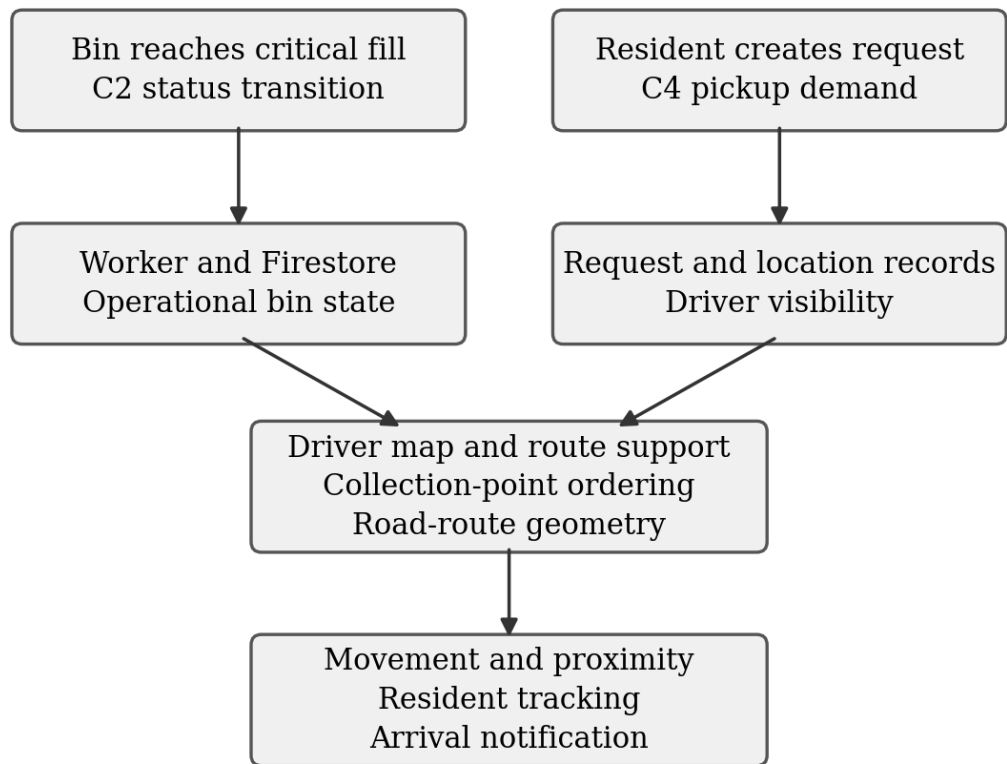


Figure 3.2 Collection Workflow

The flow links the same service cycle at different timescales. Sensor status supports current attention at the bin, requests support active demand, location records support movement and arrival, and forecasts support weekly preparation.

4. COMPONENT INTEGRATION AND COHESIVE OPERATION

4.1 Integration Strategy

We organised the solution around two complementary needs: current collection coordination and advance municipal planning. Current coordination uses bin status, resident demand and vehicle position. Advance planning uses historical waste quantities, calendar information and rainfall. This arrangement allows each component to retain the data resolution required for its purpose while contributing to the same waste-management process.

The environmental component produces time-stamped condition observations. The fill component produces capacity and lid-state observations, together with collection-relevant status transitions. The mobile component presents operational events to residents, drivers and administrators. The forecasting component transforms zone-level weekly records into expected waste quantities and planning estimates. Integration therefore begins with clear ownership of observations, predictions and collection activities.

The implemented cross-component collection path connects the fill-monitoring prototype to the mobile driver's critical-bin display. An ESP32 status message reaches the Cloudflare Worker, which records the event in Firestore. The application then presents the bin as an identifiable collection point. Resident requests and driver location updates use the application backend to support tracking and arrival interactions. This path gives the common solution a practical connection between physical monitoring and the collection interface.

The environmental cloud path and the forecasting dashboard contribute separate views of the same municipal service. Environmental records support attention to bin conditions, while the dashboard supports weekly preparation. Their modular interfaces provide a basis for extending the operational view as the system develops. We use the architecture in Figure 3.1 to show these deployment boundaries and distinguish the implemented collection path from the planned extension of planning information into dispatch.

This strategy also supports development by four team members. Sensor acquisition, model fitting and mobile interaction can be developed with different tools, then related through identifiers, locations, status classes and documented service interfaces. The integration work concerns the meaning and movement of the information, rather than requiring every component to use one programming language or database.

4.2 Interfaces Data Contracts and Shared Models

Table 4.1 records the principal current interfaces and their maturity. It distinguishes sensor storage, event delivery, forecast calculation and planned operational exchange. These distinctions are necessary because the current components use Firebase Realtime Database, Cloud Firestore and MongoDB for different purposes.

Table 4.1 Interfaces and Current Evidence Status

Interface	Representation	Current status
Environmental unit to Realtime Database	Bin ID, message ID, timestamp, location, readings and status	Telemetry integration reported
Fill unit to Realtime Database	distance, fillLevel, binStatus, lidStatus and uptime timestamp	Prototype storage and retrieval reported
Fill unit to collection worker	Device identifier, critical/normal status and coordinates	State-change path documented
Worker and Firestore to mobile client	Operational bin state and real-time application records	Critical-bin-to-map scenario reported
Forecast service to web dashboard	Zone-week predictions and derived planning outputs	Internally integrated
Forecast service to dispatch or mobile client	Versioned forecast and planning record	Not implemented in the forecasting thesis

The two sensing components use different status vocabularies because they address different questions. Environmental green, yellow and red classes indicate condition severity. Fill-monitoring OK, WARNING and CRITICAL classes indicate remaining capacity. The mobile collection interface uses the critical or normal state relevant to driver attention. Maintaining these meanings prevents a collection-priority marker from being mistaken for a calibrated environmental hazard assessment.

Time also has different meanings across the components. Environmental records use synchronised time for cloud history. The fill firmware includes a millisecond device timestamp for its readings. Mobile requests use creation and expiry times to define the request window. Forecast records represent an entire zone-week. These fields should be interpreted within their originating service when records are displayed or combined. The application and forecasting services retain their respective storage models. Firestore supports operational collection records and user workflows. MongoDB stores forecasting application records and prediction history. Firebase Realtime Database stores sensor observations. This division reflects the implemented applications and allows the common architecture to describe a realistic service boundary rather than forcing unrelated records into an artificial common table.

4.3 Shared Nonfunctional Requirements

The common solution depends on reliable data delivery, responsive interfaces and maintainable services. For environmental sensing, reliability includes surviving network interruptions and resending buffered observations. For fill monitoring, it includes repeatable classification at the configured capacity boundaries. For the mobile application, it includes consistent request visibility, authenticated access and location updates. For forecasting, it includes reproducible feature preparation and consistent treatment of input categories.

Network availability affects both sensing and mobile coordination. The environmental firmware provides a local SPIFFS buffer and reconnection logic, allowing observations collected during an outage to be transmitted after connectivity returns. The fill prototype communicates through Wi-Fi and HTTP requests. Mobile users require network access for shared requests and current vehicle positions. The operational configuration therefore includes device connectivity, endpoint configuration and access credentials as part of deployment.

Location access is a specific requirement of vehicle tracking. The Flutter application uses Geolocator to obtain device location and manage the associated platform permissions [13]. Driver operating mode determines the location information relevant

to resident tracking. Coordinates support a service interaction and should be handled with the same role restrictions as the corresponding account and request records.

Security responsibilities follow the service boundaries. Authentication identifies users and their application roles. Backend access controls protect operational records. Sensor credentials and service settings require controlled configuration. Forecasting account functions use password hashing and token-based sessions. These mechanisms support the prototype workflows and provide a foundation for operational administration.

Maintainability is supported by modular firmware, separate model pipelines, documented API routes and role-specific mobile screens. A change to a fill threshold should be traceable to the capacity rules and tests. A change to the forecast model should be traceable to the stored pipeline and validation results. A change to a mobile workflow should be assessed against the relevant user scenario.

4.4 Integrated Workflow and Exception Handling

Normal collection operation begins with an authenticated user and a current set of collection demands. A critical-bin event identifies a condition requiring attention, while a resident request creates a separate demand with a location and time window. The driver can view these records and use route support. Resident tracking and arrival notification depend on fresh driver location. The demonstrated scenario confirms the principal transitions, but not every possible concurrent or failure condition.

Table 4.2 defines important exceptions and the required response. The responses are an operating specification for fuller integration, not a claim that all have been demonstrated. They identify which component should retain information and which user needs to see the effect of a failure.

Table 4.2 Exception Handling Requirements for Cohesive Operation

Exception	Required response	Responsible boundary
Sensor connection lost	Retain last reading with age and mark current condition unavailable	Device service and monitoring interface
Duplicate critical event	Update one bin state and avoid duplicate active demands	Collection worker and event adapter
Device clock reset	Preserve receipt time and flag uncertain source time	Ingestion service

Weather source unavailable	Use the next tier and disclose the chosen source	Forecast service and dashboard
Previous-week tonnage missing	Identify missing input and apply an explicitly approved policy	Forecast service
Driver location stale	Show last update and avoid confident arrival claims	Mobile location service
Route geometry unavailable	Explain the unavailable road route and retain collection-point visibility	Mobile route service
Pickup window expires	Update the request state and inform the relevant users	Pickup service

The forecast service should also separate operational failure from model uncertainty. An available prediction can still have substantial error, especially for wet waste within a zone. The dashboard should show the model evidence and input provenance rather than presenting a precise value as guaranteed demand. Conversely, a temporary service failure should not be represented as a prediction of zero waste.

Recovery requires traceable state transitions. Buffered sensor records should retain message identity, a repeated request should not create multiple collection demands, and a completed pickup should remain attributable to its request. These controls make the workflow auditable and allow integrated test logs to distinguish recovery from normal operation.

4.5 Team Contribution and Dependency Matrix

The common solution depends on the four individual contributions listed in Table 4.3. The owner of each component is responsible for its detailed research and implementation evidence, while interface work requires agreement between producer and consumer.

Table 4.3 Team Contributions and Dependencies

Member and component	Contribution	Integration dependency
Hemsara K. D. S. - C1	Sensor selection, environmental firmware, calibration, status and telemetry design	Common bin identity, condition semantics and downstream interpretation

Gamaariya C. C. - C2	Fill sensing, lid-state reporting, Firebase records, LEDs and critical-event interface	Collection-worker mapping and future historical feature preparation
Jayawardhane K. G. M. S. - C3	Municipal dataset pipeline, wet/dry models, weather retrieval, planning calculations and dashboard	Future sensor feature mapping and forecast-to-dispatch adapter
Wijesiri A. M. - C4	Flutter workflows, backend functions, tracking, route support and account management	Critical-bin event delivery, request state and location freshness

Shared outputs include the project architecture, component numbering, common problem definition, interface responsibilities and the synthesis of evaluation results. The individual theses provide different levels of detail about collaboration.

5. INTEGRATED IMPLEMENTATION AND PROJECT EXECUTION

5.1 Common Technology Stack and Infrastructure

The system uses several technology groups chosen for different responsibilities. ESP32 and Arduino/C++ support sensing and local processing. Firebase Realtime Database stores prototype sensor information. Secure Hypertext Transfer Protocol (HTTPS) supports device communication. The mobile application uses Flutter/Dart, Firebase Authentication, Firestore and a Cloudflare Worker backend. Forecasting uses a Python Flask service, XGBoost and scikit-learn, MongoDB Atlas and a React web interface. Table 5.1 summarises the stack without implying a single shared database.

Table 5.1 Technology Stack by System Responsibility

Responsibility	Technologies	Purpose
Environmental edge sensing	ESP32, MQ-135, MQ-4, DHT22, Arduino/C++	Acquisition, filtering, classification and telemetry
Fill and lid monitoring	ESP32, HC-SR04, reed switch, Wokwi, LEDs	Distance-derived fill, lid state and prototype verification
Sensor cloud store	Firebase Realtime Database and HTTPS interfaces	Latest readings and prototype histories
Prediction service	Python 3.13, Flask 3.1.1, XGBoost, scikit-learn	Feature preparation, inference and planning calculations
Prediction persistence and UI	MongoDB Atlas, PyMongo, React 18, Vite, Recharts	Prediction history and supervisor dashboard
Mobile client and backend	Flutter, Dart, Provider, GoRouter, Cloudflare Workers	Role-based collection workflows and protected operations
Mobile identity and data	Firebase Authentication, Firestore, Cloud Messaging	Accounts, real-time records and notifications
Location and road display	Geolocator, flutter_map, OpenStreetMap, OSRM	Position, map display and road-route geometry

The architectural rationale is separation of concerns. Low-cost microcontrollers perform the measurements close to the bin. Cloud stores make current information available to applications. The forecasting service keeps preprocessing and models

together using scikit-learn, a Python machine-learning library [14], so inference uses the same transformation as evaluation. The mobile application combines role-specific screens with live operational records. The scikit-learn Pipeline API combines data transformers with a final predictor, supporting consistent preprocessing during model use [15]. Flutter's architecture guide likewise recommends separating user-interface and data responsibilities [16].

This separation also creates maintenance responsibilities. Firebase access rules, Worker credentials and prediction-service tokens do not constitute one uniform authorisation system. Similarly, a deployment of one component does not make the others available. A complete release needs a manifest of device firmware, service versions, database rules, and configuration and external dependencies

5.2 Development and Integration Process

The sensing research follows Design Science Research (DSR) through problem awareness, design, build, evaluation and documentation. This reflects the building and evaluation of an information-systems artefact described by Hevner et al. [17]. The fill work combines simulation with hardware testing and records the difference between the richer design and the implemented prototype. The forecasting work follows data preparation, pipeline development, chronological evaluation and dashboard implementation. Time-series evaluation requires attention to the relationship between training and test periods, as discussed by Bergmeir and Benítez [18]. The mobile work follows design and development with practical functional and end-to-end testing.

These approaches support an iterative group process in which interfaces can be checked before a municipal pilot. A sensor unit should first produce a valid local reading, then a valid cloud record, then an operational event visible to the driver. A forecasting service should first reproduce its held-out results, then return a valid planning record, then connect that record to dispatch. Passing a component test should be a prerequisite for the next integration step, but cannot replace that step.

Version control is described in the individual implementation records, with specific mobile contribution evidence reported as 47 commits. The common evidence does not include an exported group issue history, sprint board, continuous-integration log or

release record. Accordingly, this report does not claim a particular number of sprints, completed reviews or automated release checks. The demonstrated process is prototype development and manual integration testing, supported by component artefacts.

For the next iteration, an interface-first sequence is appropriate. The team should approve the registry and contracts in Appendix A, implement the missing adapters, record test fixtures for each contract, and then execute the shared acceptance scenarios. Schema changes should be reviewed by both producer and consumer owners. Model versions and dataset revisions should be recorded independently from application revisions because a change in training data can alter predictions without changing a mobile screen.

This process makes the remaining integration work concrete while preserving the research contribution already demonstrated. It also creates the records needed to resolve inconsistent metrics and link future outcomes to a specific system version.

5.3 Deployment and Operational Configuration

Current operation requires separate configuration for devices, cloud services, forecasting and mobile clients. Sensing devices need bin identity, calibrated geometry or sensor constants, Wi-Fi settings and an authorised cloud destination. The environmental design includes configurable thresholds and buffering. The fill prototype needs its measured empty and full distances and the endpoint used for critical-state transitions.

The forecast service loads the two trained pipelines, connects to MongoDB and obtains external rainfall information when required. Configuration secrets are supplied through environment variables. Authenticated users select a target week and receive prediction and planning outputs through the web dashboard. Packaging the service for a council-managed server is future work.

The mobile application requires its Firebase project configuration, Worker backend, database rules, messaging services and external map and route access. Drivers must have location services enabled and operate in the relevant mode for residents to see applicable vehicles. Account recovery and administrative restrictions are part of the

application workflow. A field release would additionally need device provisioning, access review and an operational support arrangement.

Monitoring should cover last sensor report, delivery failures, driver-location age, forecast errors, external-service failures and account-management events. Each measurement should identify its time and component so that a fault can be located. Existing component measurements do not constitute a common monitoring dashboard.

Backup and restoration are also common release requirements. Calibration and registry data, prediction history, model artefacts and operational collection records should be backed up according to their role. A restoration test should demonstrate that bin identities, model versions and active request states remain consistent after recovery.

5.4 Evidence of Team Execution

The environmental implementation produced a working sensor-to-cloud path and evaluated it across transmission, latency, alert, reference-sensor and continuity tests. The fill implementation produced distance and lid records, local capacity indications, a monitoring display and collection-state communication. These outputs show the team's work at the physical observation layer.

The forecasting implementation produced two fitted models and an application that turns their outputs into planning information. The dashboard and database functions extend the model beyond an offline experiment by allowing prediction requests, stored histories and inspection of model validation results.

The mobile implementation produced resident, driver and administrator workflows. Its manual test set includes registration, password recovery, request visibility, movement tracking, proximity notification, and critical-bin display and account controls. The use of separate resident and driver devices, including movement tests, supported practical assessment of the user interactions.

6. INTEGRATED TESTING EVALUATION AND VALIDATION

6.1 Evaluation Framework and Success Criteria

We evaluated the common solution through methods matched to the information each component produces. Sensor evaluation concerns readings, delivery and classification. Forecast evaluation concerns numerical estimates of future waste quantities. Application evaluation concerns whether a user task produces the expected visible result. The common evaluation combines these findings around the observation, planning and collection functions.

Environmental tests measured transmission delivery, cloud latency, condition classification, and agreement with reference instruments, calibration drift and operational continuity. Fill-monitoring tests exercised the capacity-class boundaries, lid states and cloud fields. Forecasting used an ordered training and test partition. Mobile evaluation used manual scenarios covering resident, driver and administrator roles.

Forecast performance is expressed using coefficient of determination, mean absolute error, root mean squared error and mean absolute percentage error. MAE and RMSE retain the tonne scale of the target, helping the reader assess typical error and the influence of larger deviations. MAPE expresses relative error and should be interpreted with the magnitude of the actual waste values. Forecast-accuracy measures have different properties, so the metric set is more informative than a single accuracy label [19].

The principal success criteria were practical: acquire and communicate bin conditions; identify capacity and lid states; generate weekly wet and dry estimates; turn predictions into planning outputs; and support collection requests, critical-bin visibility, tracking and arrival interactions.

Table 6.1 Evaluation Areas and Acceptance Basis

Area	Acceptance basis	Evidence currently available
Environmental sensing	Sensor error, alert performance, delivery and latency targets	Quantitative prototype results, with some target shortfalls

Fill monitoring	Correct boundary classes and valid cloud records	Functional tests and inconsistent quantitative summaries
Critical-bin integration	Physical condition becomes an identifiable driver-map event	Controlled IoT-to-driver scenario
Forecasting	Errors on a chronological hold-out and per-zone interpretation	126-record test-set metrics
Planning outputs	Consistent calculations with visible assumptions	Dashboard, fleet and revenue functions
Resident coordination	Request, tracking and arrival transitions operate together	Manual end-to-end scenarios

6.2 Integration and End to End Testing

The critical-bin test starts with a smart-bin condition and ends with a visible red marker in the driver application. This is a meaningful cross-component result because it confirms that a device-derived event can reach an operational user interface. The mobile architecture routes the event through the Worker and Firestore rather than assuming a direct reading of Realtime Database by the mobile client.

The resident–driver scenario tests another complete path: request creation, driver visibility, vehicle movement, resident-side tracking and arrival notification. A two-driver scenario confirms that the nearest-truck display changes when relative positions change. Authentication and administrative deactivation, recovery and reactivation were also exercised.

The forecasting path is end to end within its component. A request triggers rainfall retrieval, lagged-input lookup, prediction, planning calculations, storage and dashboard display. This confirms that the models and decision functions operate within one web service. It does not test a forecast reaching a driver or influencing an executed collection trip.

Table 6.2 separates demonstrated scenarios from additional acceptance tests. Detailed input and expected-output specifications appear in Appendix A. Proposed tests should be executed against one recorded group release, with event identifiers and timestamps retained across services.

Table 6.2 Integration Scenarios and Evidence Status

Scenario	Reported outcome	Interpretation
Critical bin to driver map	Red critical-bin indication appeared	Demonstrated operational event path
Resident request to driver visibility	Request appeared on driver map	Demonstrated request propagation
Driver movement to resident tracking	Movement visible; nearest truck changed	Demonstrated controlled location workflow
Proximity to arrival notification	Expected arrival notification appeared	Demonstrated functional trigger; latency unmeasured
Forecast request to dashboard and history	Forecast and planning outputs returned and stored	Demonstrated internal prediction integration
Forecast to mobile dispatch	Not implemented	Required group integration test
Sensor features to trained forecast	Archive access described; model connection unimplemented	Data availability does not establish model use
Duplicate, stale and outage recovery across services	No common test record	Required robustness and regression testing

6.3 Overall Results

The results support a modular system with functioning sensing, prediction and coordination capabilities, together with selected cross-component interactions. They do not support a claim that all four components have been evaluated as one deployed municipal system. The following results retain their component-specific units, denominators and limitations.

Environmental sensing produced a reported payload delivery rate of 99.4% over 10,080 transmissions. Mean sensor-to-cloud latency was 1.84 seconds, with a maximum of 4.63 seconds. The temperature root mean squared error was 0.82 °C and the humidity root mean squared error was 2.74 percentage points of relative humidity. The thirty-day record reports 2.1% missing data and one unplanned restart. These measurements provide evidence of prototype sensing and cloud communication.

Table 6.3 compares environmental results with the stated targets. The classification result of 96.7% does not meet an above-97% target, and a 95.0% true-positive rate

does not meet an above-95% target. Similarly, 720 hours and one restart do not demonstrate the stricter conditions of more than 720 hours and less than one restart per 720 hours. The common evaluation therefore qualifies the source's broad pass statements rather than repeating them.

Table 6.3 Environmental Results against Stated Criteria

Metric	Stated target	Reported result	Assessment
Alert classification	Above 97%	96.7%	Target not met
Gas-alert true-positive rate	Above 95%	95.0%	Strict target not met
Payload delivery	Above 99%	99.4%	Reported target met
Cloud latency	Under 5 seconds	1.84 s mean; 4.63 s maximum	Reported test values meet target
Continuous operation	More than 720 h; fewer than one restart per 720 h	720 h; one unplanned restart	Strict target not demonstrated
Missing data	Below 5%	2.1%	Reported target met
Temperature error	Within ± 1 °C over paired readings	RMSE 0.82 °C	RMSE alone does not prove every reading is within the bound
Humidity error	Within ± 3 %RH over paired readings	RMSE 2.74 %RH	RMSE alone does not prove every reading is within the bound

Fill monitoring demonstrates correct classification at 79%, 80%, 94% and 95%, prototype cloud records and local status indication. Its quantitative evidence is internally inconsistent. The abstract reports 96.5% fill accuracy, 92.0% anomaly accuracy, 4.0% false positives and 2.4-second average alert latency. The results summary states that these percentages and latency were not verified. The detailed tables report a different set of counts and timing values.

The fill confusion matrix contains 100 observations and 96 diagonal entries, which implies 96.0% classification accuracy if those counts are authoritative. The five anomaly-rule rows contain 47 detections in 50 scripted anomalies, implying a 94.0% detection rate; they do not establish overall classification accuracy. The normal-event

denominator needed for a false-positive rate is absent. The delivery counts imply $495/500 = 99.0\%$, while the latency table reports a mean of 1.8 seconds. These are derived interpretations of unreconciled source tables, not independently validated common results.

The forecasting results are internally clearer. Table 6.5 reproduces the reported held-out metrics for the two streams. The quantity labelled predictive accuracy is 100% minus mean absolute percentage error (MAPE); it is not classification accuracy.

Table 6.5 Forecast Performance on the Reported 126 Test Records

Stream	MAE in tonnes	RMSE in tonnes	MAPE	R ²
Wet waste	3.275	4.826	19.15%	0.8035
Dry waste	1.521	2.273	12.33%	0.9492

Dry-waste performance is stronger than wet-waste performance in the reported test. The pooled wet-waste R² does not represent within-zone weekly variation: the zone-level wet-waste R² values between approximately 0.30 and 0.45. Differences in average loads between zones can contribute to the higher pooled score. A supervisor should therefore consider local errors instead of assuming the pooled result applies equally to each zone.

Errors also matter at fleet scale. The mean absolute errors for the pooled streams are below the configured 5-tonne truck capacity, but that does not guarantee the correct number of trips for every week. Rounding at a capacity boundary can change a trip count even for a relatively small error, and the source reports wet-waste mean absolute error of 5.221 tonnes in the highest-volume zone. Forecasts are useful planning inputs, with human review and contingencies.

The mobile results establish successful controlled workflows: request visibility, movement tracking, nearest-truck updates, proximity notification, and critical-bin display and account-management operations.

6.4 Stakeholder and User Validation

The system addresses residents, collection drivers, municipal supervisors and administrators. Their tasks differ, so validation focuses on the information and action

available to each role. Residents require a straightforward request process and a visible indication of vehicle progress. Drivers require identifiable collection points and navigation support. Supervisors require a planning view of expected quantities, fleet needs and waste composition.

The environmental component's stakeholder feedback supported the clarity and suitability of its status presentation, with suggestions for presentation refinements. This feedback complements the numerical telemetry and reference-sensor tests by considering how readings are communicated to users. The status classes make the condition view easier to interpret while retaining the underlying measurements.

The mobile scenarios assess user tasks directly. Registration and recovery establish access to the service. Request creation and driver visibility establish communication of pickup demand. Tracking and arrival interactions establish the resident's view of collection movement. Critical-bin viewing establishes the driver's access to monitored collection points. These outcomes connect technical backend functions with visible user activities.

The forecasting dashboard supports inspection through prediction results, composition, logistics, recovery estimates, validation and history views. Its validation view exposes model metrics, allowing a supervisor to interpret the waste estimate in the context of measured forecasting error. The planning calculations provide quantities that can be discussed against the municipality's operational capacity and processing options.

The validation activities therefore combine feedback, role-based functional testing and numerical model assessment. They show how the components serve their intended users in the prototype setting. A municipal pilot can extend these activities to routine collection work, including service timing, driver workload and supervisor use of weekly estimates.

6.5 Achievement of Common Objectives

The common objectives connect information acquisition, forecasting and collection coordination. Table 6.6 relates each objective to the implemented output and the result

presented in this report. This assessment treats the project as a set of related research artefacts and identifies the contribution of each to the shared municipal problem.

Table 6.6 Achievement of the Common Objectives

Objective	Implemented output	Evaluation result
Observe environmental conditions	ESP32 multi-sensor unit and cloud history	Delivery, latency, reference-sensor and alert measurements
Monitor capacity and lid state	Fill calculation, capacity classes and lid display	Functional boundaries and OPEN/CLOSED scenarios verified
Forecast weekly wet and dry waste	Separate XGBoost pipelines for four zones	Held-out R^2 , MAE, RMSE and MAPE results
Support planning decisions	Fleet, composition and recovery calculations	Dashboard outputs generated from configured rules
Coordinate resident and driver activity	Requests, tracking and arrival functions	Corresponding manual scenarios passed
Link monitored bins to collection	Critical-bin event path and driver map	Critical marker and viewing tests passed
Establish a common architecture	Documented component boundaries and interfaces	Sensing, planning and collection flows organised coherently

The environmental objective is supported by measured delivery and timing performance and by comparison with reference instruments. The fill objective is supported by the correct functional interpretation of capacity and lid inputs. The forecasting objective is supported by an unseen chronological test partition. The coordination objective is supported by the twelve completed manual scenarios.

Decision support extends the predictions into operational quantities without requiring the user to calculate them separately. Fleet estimates relate predicted tonnage to vehicle capacity and available vehicles. Composition estimates relate predicted quantities to the configured area profiles. Recovery estimates relate processing shares and yields to configured rates. These outputs make the planning function practically interpretable.

The common architecture is the organising contribution of the group project. It relates physical observation to collection attention and places weekly prediction beside current operational information. The implementation demonstrates this relationship through the bin-to-mobile workflow and the separate planning application. The future directions build on these existing functions.

6.6 Integrated Limitations and Threats

The evaluation describes prototype and controlled application conditions. Environmental delivery and timing depend on the tested device, network and cloud configuration. Low-cost metal-oxide gas sensors respond to operating conditions and cross-sensitivity, so their readings provide indicative bin-condition monitoring. Reference calibration, enclosure conditions and periodic maintenance remain relevant to their use.

Ultrasonic fill measurement depends on bin geometry and the waste surface. Uneven material, reflections and invalid echoes can affect distance readings. The documented prototype uses calibrated empty and full distances and functional classification tests. Deployment in differently shaped bins requires corresponding calibration and checks under the intended waste conditions.

Forecast findings relate to the four-zone dataset and its chronological partition. Zone-level wet-waste results show that performance differs across targets and areas. Rainfall source selection and longer-horizon fallback assumptions can also affect predictions. Further evaluation over successive operational periods would characterise this behaviour as collection and weather patterns change.

The mobile evaluation establishes completion of twelve manual workflows under controlled conditions. Location behaviour depends on device permissions, position availability and network updates. The route-support method uses nearest-neighbour ordering and road geometry; its operational benefit should be measured against actual collection practice during field use.

Fleet, composition and recovery outputs depend on configured capacity, shares, yields and rates. They are planning estimates, and actual vehicle availability, processing performance and market conditions determine operational outcomes. A municipal

pilot can assess these outputs together with service time, user experience and operating costs. These considerations define the next evaluation setting for the implemented system.

7. COMMERCIALISATION SUSTAINABILITY AND RESPONSIBLE INNOVATION

7.1 Value Proposition and Beneficiaries

The system's value proposition is access to current collection information and advance waste estimates through related digital tools. Residents can create pickup requests and follow the relevant collection vehicle. Drivers can view requests and critical bins and use map-based route support. Municipal supervisors can inspect expected wet and dry quantities with fleet and recovery estimates. Administrators can manage access and account status.

For sensing, the benefit is visibility of bin conditions between physical inspections. Environmental measurements and capacity classes provide different kinds of information: one concerns condition and the other concerns space and collection attention. Presenting these observations through cloud and application interfaces makes them available to personnel away from the bin.

For planning, the benefit is a repeatable weekly estimate with interpretable supporting calculations. Separate wet and dry forecasts reflect different processing and collection needs. The dashboard presents the predicted quantities alongside the assumptions used for fleet, composition and recovery estimates. This allows users to review the planning output in relation to the municipality's available resources.

For collection coordination, the benefit is a common interface for resident demand, monitored collection points and vehicle movement. The manual tests show that these functions can operate within the prototype application. Commercial value would arise from their use in routine service, including improved visibility and coordination across the relevant roles.

7.2 Market Alternatives and Differentiation

Municipal collection can use scheduled routes, manual inspection, telephone requests, and stand-alone vehicle tracking or specialised smart-bin services. These approaches provide useful functions at different points in the service cycle. Our system brings

sensing, zone-level prediction and role-based collection coordination into one common architecture.

Table 7.1 compares the functions represented by these alternatives. It is a functional comparison of approach types, rather than a claim about a named commercial product. The group's differentiation lies in relating low-cost embedded monitoring to a mobile collection interface while providing a separate weekly wet and dry planning service.

Table 7.1 Alternatives and the Intended Combined Service

Approach	Principal information	Relation to this project
Scheduled collection	Established timetable and route	Baseline operational process supported by current information
Manual inspection	Observed condition at inspection time	Complemented by remote sensor records
Telephone or informal pickup requests	Resident-reported demand	Represented through authenticated, time-bounded mobile requests
Vehicle tracking	Vehicle position and movement	Included with requests and arrival interactions
Smart-bin monitoring	Bin capacity or condition	Extended through collection-state visibility on the driver map
Forecasting dashboard	Expected area-level quantities	Included with wet/dry models and planning calculations

The use of separate models for wet and dry waste gives the forecasting service a direct connection to composition and processing decisions. The event path from fill monitoring to the driver map gives the collection application a physical source of priority information. The role-based interfaces make the service accessible according to the activity of residents, drivers and administrators.

A municipal adoption decision would consider compatibility with existing operations, maintenance requirements, communications coverage and the usability of the interfaces. The modular design permits a staged introduction of the functions most useful to a particular operator. Subsequent integration can use the documented interfaces rather than replacing the underlying artefacts.

7.3 Business Model and Feasibility

The individual components discuss service subscriptions, hardware supply or leasing, and support arrangements as possible commercial approaches. A common offering can combine sensor equipment with cloud and application services, while allowing forecasting and mobile coordination to be adopted according to municipal needs. The recurring service would cover hosting, maintenance, calibration support and application updates.

The environmental component reports a prototype hardware cost of LKR 12,400. The fill-monitoring budget reports LKR 7,400 for electronics and supporting items, and a separate development allowance of 480 hours at LKR 150 per hour. These are component-specific prototype budget figures. Table 7.2 preserves their stated basis so that hardware expenditure and development effort are interpreted correctly.

Table 7.2 Reported Prototype Cost Bases

Component	Cost item	Reported amount or basis
Environmental sensing	Prototype hardware	LKR 12,400
Fill and lid monitoring	Electronics and supporting items	LKR 7,400
Fill and lid monitoring	Development allowance	480 hours × LKR 150 = LKR 72,000
Fill and lid monitoring	Combined stated budget	LKR 79,400

Operating costs depend on the number of bins, active users, update frequency and use of weather, mapping and notification services. Hardware support includes sensor checks, calibration, enclosure maintenance and replacement. Application support includes backend administration, security updates, monitoring and user assistance. Forecasting support includes data updates, model review and adjustment of processing assumptions.

The recovery calculation offers a separate planning perspective. It estimates gross value from configured shares, yields and rates for wet and dry materials. It is useful for comparing processing scenarios, but operating costs, contamination, transport and actual sale conditions determine realised financial performance. Appendix A records

the configured calculation inputs so that a reader can understand the basis of the dashboard estimate.

A feasible adoption path begins with a limited municipal service area and a defined set of users and bins. The operator can assess device maintenance, connectivity, driver use and planning value before expanding. The component structure supports this staged approach because monitoring, coordination and forecasting can be evaluated within their own operating boundaries.

7.4 Adoption Intellectual Property and Compliance

Adoption requires alignment with existing collection responsibilities. The municipality determines service areas, authorised personnel, and vehicle availability and collection procedures. Residents require clear instructions for requesting pickup and interpreting vehicle information. Drivers require practical map and request interfaces that fit their work. Administrators require account controls and support processes.

Data stewardship is central to responsible operation. User identity, pickup coordinates and vehicle movement should be available only to the roles that require them. Sensor records should retain their device identity and interpretation. Forecasting data should be used with the permissions and attribution appropriate to the municipal source. Public presentation of planning results should avoid unnecessary disclosure of individual account or location information.

The project combines original component work with third-party software and services. Intellectual-property management should preserve the team's contribution while respecting the licenses and conditions of embedded libraries, application frameworks, mapping data and service providers. OpenStreetMap attribution and the relevant software licenses form part of deployment preparation. The institution's research and intellectual-property arrangements also apply to the group's work.

Responsible presentation includes explaining the meaning of each output. Environmental readings are condition indicators, fill percentages are geometry-based estimates, forecasts are model outputs, and recovery figures are configured financial estimates. The user interfaces and operational guidance should preserve these

distinctions. Doing so supports informed decisions without overstating what a prototype measurement or prediction establishes.

A staged pilot also provides an opportunity to refine accessibility, user support and maintenance arrangements. Feedback from residents, drivers and supervisors can inform screen wording, request procedures and alert presentation. This approach develops the existing functions through their intended use rather than assuming that one interface suits every operating context.

7.5 Sustainable Development Goal Contribution

The project relates to the sustainable urban service and waste-management themes of the United Nations 2030 Agenda [3]. Its functional contribution is the availability of information that can support collection coordination, resource planning and recovery decisions. The relevant Sustainable Development Goals include sustainable cities and communities, responsible consumption and production, and climate action.

For SDG 11, the system supports municipal service visibility through monitored bins, resident requests and vehicle tracking. These functions relate to the organisation of waste collection in urban communities. Their contribution can be assessed in a pilot through indicators such as request handling, overflow incidents and collection-service responsiveness.

For SDG 12, separate wet and dry predictions and composition estimates support discussion of treatment and recovery options. The dashboard's configured recovery calculations make the assumptions visible in planning. Operational assessment can examine the quantities directed to composting, recycling and other processing activities, together with actual recovery yields.

For SDG 13, fleet and route information can support examination of travel and fuel use as collection decisions develop. The prototype demonstrates the relevant information functions. A field evaluation can measure distance, fuel consumption and service outcomes under an agreed comparison procedure. The system's sustainability contribution therefore begins with decision-support capability and develops through measured municipal use.

8. CONCLUSION AND FUTURE DIRECTIONS

8.1 Overall Conclusion

We developed a four-component Smart Waste Management System that addresses current bin information, advance waste estimates and collection coordination. The environmental unit measures bin conditions and transmits classified records. The fill and lid unit converts ultrasonic distance into capacity information and reports lid state. The forecasting service predicts weekly wet and dry waste for four Kalutara Municipal Council zones. The Flutter application supports residents, drivers and administrators through requests, critical-bin visibility, and tracking and arrival interactions.

The evaluation demonstrates the feasibility of these functions. Environmental telemetry achieved a reported 99.4% delivery rate and 1.84-second mean cloud latency, with 96.7% alert classification. Fill monitoring was verified through functional boundary and scenario tests. On the 126-record chronological forecasting test set, dry waste achieved R^2 of 0.9492 and MAE of 1.521 tonnes; wet waste achieved R^2 of 0.8035 and MAE of 3.275 tonnes. All twelve principal mobile manual scenarios passed.

The implemented fill-to-mobile path connects a monitored collection state to a driver-facing map point. The forecasting dashboard adds a weekly planning view with fleet, composition and recovery estimates. Together, these artefacts provide a coherent basis for municipal decision support across observation, preparation and collection activity.

8.2 Principal Integrated Contribution

The principal contribution is a modular architecture that links sensing, forecasting and collection action, and separates measured results from predictions and configured assumptions. The components share a municipal purpose while retaining the data and processing methods appropriate to their functions. This allows the report to relate device readings, zone-week estimates and user workflows within one service framework.

The physical-to-operational link is demonstrated by the critical-bin workflow. A sensor-derived state becomes a stored collection event and a visible driver map point.

The application then provides request, movement and arrival functions around the collection activity. This is a concrete integration result within the group solution.

The analytical contribution is the separate wet and dry forecasting service with interpretable decision-support outputs. The models are evaluated on unseen chronological records, and the dashboard presents both predictions and their planning implications. The recorded fleet and recovery assumptions make the relationship between model output and operational estimate explicit.

The common report consolidates the team's embedded, analytical and application work into a shared account of objectives, architecture, implementation and evaluation. Its findings establish a technical foundation for a field pilot and for further connections between the planning service and operational collection information.

8.3 Future Directions

The next development direction is to connect the forecasting service with bin-level records and the collection application through the documented service interfaces. Zone-week predictions can be presented to supervisors alongside current collection demand, while appropriately aggregated sensor histories can be evaluated as additional predictors. This extends the current modular architecture and its implemented collection-event path.

A common municipal pilot should exercise the sensing units, dashboard and mobile workflows over successive collection cycles. The pilot can assess device maintenance, connectivity, user interaction and the use of weekly forecasts in planning. Service indicators should include request response time, overflow events, collection distance and actual recovered quantities. These measurements would characterise the practical value of the system in routine operation.

Forecasting development can use successive chronological evaluation windows to examine model stability across seasons and changing collection patterns. TimeSeriesSplit provides a documented mechanism for ordered cross-validation [20]. The choice of window and prediction horizon should follow the intended weekly planning use. Additional work can compare alternative models, refine event

information and examine the zone-specific wet-waste behaviour shown in the current results.

Sensing development can broaden calibration and functional tests across bin shapes, waste surfaces and outdoor conditions. Longer deployment periods can assess sensor drift, enclosure performance and recovery after interruptions. The existing threshold and interface specifications provide a reproducible starting point for these tests.

Mobile development can extend operational confirmation and supervisor workflows, then evaluate route support with actual collection vehicles and service constraints. User feedback can guide refinements to request handling, map presentation and notification wording. Financial and sustainability evaluation can then use recorded operating costs and service measurements alongside the dashboard's planning estimates.

REFERENCES

- [1] I. Sosunova and J. Porras, “IoT-Enabled Smart Waste Management Systems for Smart Cities: A Systematic Review,” *IEEE Access*, vol. 10, pp. 73326–73363, 2022, doi: 10.1109/ACCESS.2022.3188308. [Online]. Available: <https://doi.org/10.1109/ACCESS.2022.3188308>; Open full text. Accessed: 10 Oct. 2026.
- [2] P. Zoumpoulis, F. K. Konstantinidis, G. Tsimiklis and A. Amditis, “Smart bins for enhanced resource recovery and sustainable urban waste practices in smart cities: A systematic literature review,” *Cities*, vol. 152, article 105150, 2024, doi: 10.1016/j.cities.2024.105150. [Online]. Available: <https://doi.org/10.1016/j.cities.2024.105150>; Open full text. Accessed: 10 Oct. 2026.
- [3] United Nations General Assembly, “Transforming our world: The 2030 Agenda for Sustainable Development,” Resolution A/RES/70/1, 2015. [Online]. Available: <https://sdgs.un.org/2030agenda>. Accessed: 10 Oct. 2026.
- [4] Google, “Firebase Realtime Database.” [Online]. Available: <https://firebase.google.com/docs/database>. Accessed: 10 Oct. 2026.
- [5] Google, “Cloud Firestore.” [Online]. Available: <https://firebase.google.com/docs/firestore>. Accessed: 10 Oct. 2026.
- [6] Cloudflare, “Cloudflare Workers documentation.” [Online]. Available: <https://developers.cloudflare.com/workers/>. Accessed: 10 Oct. 2026.
- [7] Espressif Systems, “Arduino ESP32 documentation.” [Online]. Available: <https://docs.espressif.com/projects/arduino-esp32/en/latest/>. Accessed: 10 Oct. 2026.
- [8] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785. [Online]. Available: <https://arxiv.org/abs/1603.02754>. Accessed: 10 Oct. 2026.
- [9] Open-Meteo, “Weather forecast API documentation.” [Online]. Available: <https://open-meteo.com/en/docs>. Accessed: 10 Oct. 2026.
- [10] Google, “Firebase Authentication.” [Online]. Available: <https://firebase.google.com/docs/auth>. Accessed: 10 Oct. 2026.
- [11] Google, “Firebase Cloud Messaging.” [Online]. Available: <https://firebase.google.com/docs/cloud-messaging>. Accessed: 10 Oct. 2026.

- [12] Project OSRM, “OSRM API documentation version 5.24.0.” [Online]. Available: <https://project-osrm.org/docs/v5.24.0/api/>. Accessed: 10 Oct. 2026.
- [13] Baseflow, “Geolocator Flutter package documentation.” [Online]. Available: <https://pub.dev/packages/geolocator>. Accessed: 10 Oct. 2026.
- [14] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: <https://jmlr.org/papers/v12/pedregosa11a.html>. Accessed: 10 Oct. 2026.
- [15] scikit-learn developers, “Pipeline API reference.” [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html>. Accessed: 10 Oct. 2026.
- [16] Flutter, “Guide to app architecture.” [Online]. Available: <https://docs.flutter.dev/app-architecture/guide>. Accessed: 10 Oct. 2026.
- [17] A. R. Hevner, S. T. March, J. Park and S. Ram, “Design science in information systems research,” *MIS Quarterly*, vol. 28, no. 1, pp. 75–106, 2004, doi: 10.2307/25148625. [Online]. Available: <https://doi.org/10.2307/25148625>; Open full text. Accessed: 10 Oct. 2026.
- [18] C. Bergmeir and J. M. Benítez, “On the use of cross-validation for time series predictor evaluation,” *Information Sciences*, vol. 191, pp. 192–213, 2012, doi: 10.1016/j.ins.2011.12.028. [Online]. Available: <https://doi.org/10.1016/j.ins.2011.12.028>. Accessed: 10 Oct. 2026.
- [19] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006, doi: 10.1016/j.ijforecast.2006.03.001. [Online]. Available: <https://doi.org/10.1016/j.ijforecast.2006.03.001>. Accessed: 10 Oct. 2026.
- [20] scikit-learn developers, “TimeSeriesSplit API reference.” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html. Accessed: 10 Oct. 2026.

APPENDICES

Appendix A Integrated Interface Specification

This appendix defines the recommended common interface needed to complete the integration described in Sections 3.2, 4.2 and 6.2. Current source fields are identified separately from proposed additions. The specification does not imply that the forecasting-to-dispatch connection or every robustness control has already been implemented.

A.1 Identity and Location Registry

Every physical device should have a unique device identifier linked to one registered bin. The bin record should contain its location, municipal zone, active state and calibration reference. A replacement device should not create a new operational bin unless the physical bin itself changes. Maintaining device and bin identity separately permits replacement and recalibration without losing the bin's collection history.

Table A.1 Recommended Shared Registry Fields

Field	Meaning and validation	Implementation status
bin_id	Stable unique identifier for the physical bin	Present in environmental design; common registry required
device_id	Unique reporting hardware identity	Device identifier used by critical-event path
zone_id	Registered municipal collection zone	Required for sensor-to-zone and forecast joins
latitude and longitude	Validated bin coordinates in decimal degrees	Location present in telemetry and collection event designs
active	Whether the bin is currently in service	Recommended common field
calibration_version	Reference to geometry or sensor constants	Recommended provenance field
registered_at and updated_at	UTC registry timestamps	Recommended audit fields

The registry should reject unknown zones and invalid coordinates before data is used in planning or maps. A device may still retain a local reading during registration failure, but the system should not silently assign it to a default municipal zone. Registry changes should be attributable to an authorised administrator and retained in an audit record.

A.2 Environmental Telemetry Contract

The environmental unit defines `bin_id`, `msg_id`, `timestamp`, `location`, `gas_mq135_ppm`, `gas_mq4_ppm`, `temperature_c`, `humidity_rh` and `status`. Latest readings, history and alerts are described under the bin hierarchy in Realtime Database. Table A.2 specifies how a common adapter should preserve these values and add provenance needed for operational interpretation.

Table A.2 Environmental Telemetry Fields and Interpretation

Field group	Required interpretation	Validation or extension
<code>bin_id</code> and <code>msg_id</code>	Bin identity and per-device message identity	Reject unregistered bin; identify duplicate messages
<code>timestamp</code>	Source observation time	Validate UTC syntax and retain receipt time separately
<code>location</code>	Reported device or bin coordinates	Compare with registered installation location
<code>gas_mq135_ppm</code>	Calibrated gas-related estimate	Retain sensor type; do not relabel as certified species measurement
<code>gas_mq4_ppm</code>	Calibrated methane-related estimate	Retain calibration and quality information
<code>temperature_c</code> and <code>humidity_rh</code>	Ambient readings in stated units	Flag unavailable or invalid measurements
<code>status</code>	GREEN, YELLOW or RED environmental state	Retain determining parameter and threshold version
<code>schema_version</code> and <code>received_at</code>	Contract version and server receipt time	Recommended additions for common ingestion

Invalid readings should be flagged rather than replaced by normal values. A repeated message should update delivery diagnostics without creating a duplicate alert. The

latest record should show both its value and age, while the historical record retains the original payload and quality flags.

A.3 Fill and Lid Record Contract

The prototype stores distance, fillLevel, binStatus, lidStatus and a timestamp derived from the device uptime counter. Its design schema adds fill_readings, lid_events and alerts with richer event information. These are different maturity levels.

Table A.3 Fill and Lid Field Mapping

Prototype value	Recommended common representation	Important condition
distance	distance_cm	Retain measured distance and calibration reference
fillLevel	fill_pct	Constrain to defined range; distinguish volume estimate from mass
binStatus	fill_status	Use OK, WARNING or CRITICAL
lidStatus	lid_state	Use a defined open/closed/unknown enumeration
timestamp from millis	device_uptime_ms	Do not interpret as UTC time
No wall-clock field	received_at; observed_at when synchronised	Server time needed for cross-device comparison
Design event_type	event_type only when captured	Do not claim interrupt-based events from polled state alone

The fill-class boundaries should be tested directly: 79% is OK, 80% is WARNING, 94% is WARNING and 95% is CRITICAL. Invalid geometry or distance should produce a quality condition rather than an unqualified percentage. The richer anomaly contract should be introduced only when the corresponding rules are implemented, versioned and evaluated.

A.4 Operational Critical Bin Event

The implemented collection-service path uses a device identifier, critical/normal status and coordinates, transmitted on a state change. The mobile application consumes this

binary state. The common extension should retain the event's source and reason so that future environmental alerts can be interpreted appropriately instead of appearing as unexplained collection demand.

Table A.4 Recommended Operational Event Contract

Field	Purpose	Rule
event_id	Unique status-transition identity	Use for duplicate suppression and traceability
bin_id and device_id	Relate operational event to physical source	Resolve through registered identity
operational_status	critical or normal	Preserve the current mobile enumeration
source_component	environmental or fill	Required when multiple sources contribute
source_status and reason	Original class and determining condition	Do not discard environmental cause
observed_at and received_at	Event timing and freshness	Flag uncertain source clocks
latitude and longitude	Collection-point location	Use validated registry position
schema_version	Contract compatibility	Reject or quarantine unsupported changes

A normal event from the fill unit should not automatically clear an independent critical environmental condition. A combined operational state requires an explicit aggregation policy. A recommended policy is to retain separate active causes and clear only the cause addressed by its own source event or an authorised review. This policy is a design requirement for multi-source operation, not a measured result.

A.5 Forecast Service Contract

The existing service includes POST /api/predict, GET /api/rainfall-forecast, GET /api/previous-week, GET /api/backtest and GET /api/predictions, with authentication endpoints. The common dispatch consumer should use a versioned response that identifies the target period and preserves the evidence behind the planning calculation. The field names in Table A.5 are proposed common names; they are not asserted as the exact current response schema.

Table A.5 Recommended Forecast and Planning Record

Field group	Content	Validation
prediction_id and generated_at	Run identity and creation time	Unique run and valid UTC time
zone_id and week_start	Target collection area and period	Registered zone and agreed week boundary
wet_tons and dry_tons	Predicted weekly loads	Nonnegative values with explicit tonne units
rainfall_mm and rainfall_source	Input amount and retrieval tier	Distinguish observed, forecast and fallback values
previous_week_reference	Lagged tonnage source	Identify record and missing-input policy
model_version and dataset_version	Prediction provenance	Required for reproducing a run
truck_capacity_tons and fleet_size	Planning configuration	Positive capacity and stated fleet assumption
trips and utilisation	Derived fleet estimates	Calculate from documented aggregation policy
composition and recovery_assumptions	Shares, yields and benchmark rates	Expose assumptions and avoid realised-revenue language

Zone-level trip rounding and council-wide rounding can produce different totals. The service should state which planning policy is used and whether trips can combine loads across zones. A forecast should never be converted into a dispatch instruction without the supervisor's review of capacity, work hours and actual collection conditions.

A.6 Pickup and Collection Records

The mobile design records resident identity, coordinates, creation time, expiration time and status for a pickup request. Driver location records contain coordinates, operating mode and update information. The implemented pickup service uses a one-to-three-hour pickup window. The common record should link these operational events with the relevant bin or zone only where that relationship is valid.

Table A.6 Recommended Collection Audit Fields

Record	Minimum content	Purpose
Pickup request	Request ID, resident reference, location, creation and expiry, status	Trace demand and its valid time window
Driver position	Driver ID, coordinates, operating mode and update time	Interpret tracking and proximity freshness
Collection action	Action ID, driver, request or bin reference, time and outcome	Record what was completed or deferred
Critical-bin review	Bin, active reason, reviewer and decision	Separate collection completion from condition clearance
Planning link	Zone-week and prediction ID when used	Trace whether a forecast influenced the workflow

Arrival notification is not collection confirmation. A driver may enter the proximity radius without completing a pickup. Likewise, a request can expire without being collected. The workflow should preserve these distinctions and prevent expired demand from being presented as an active request.

A.7 Common Acceptance Scenarios

Table A.7 defines the tests needed to establish a complete and resilient common workflow. Entries marked reported refer to the individual component evidence and should still be rerun against the common release. Entries marked required have no established common execution record. Each run should retain input fixtures, relevant logs, observed output and reviewer sign-off.

Table A.7 Common Acceptance Test Matrix

ID	Scenario and expected result	Evidence state
INT01	Set fill to CRITICAL; correct bin appears on driver map	Reported controlled scenario; rerun with trace ID
INT02	Return fill to normal; clear only the relevant fill cause	State-change path documented; multi-cause behaviour required
INT03	Send duplicate event; retain one active operational condition	Required

INT04	Disconnect sensor; show stale or unavailable condition	Required common test
INT05	Reset device clock; preserve receipt time and flag source time	Required
INT06	Create resident request; same request appears to authorised driver	Reported controlled scenario
INT07	Move driver; resident sees fresh updates and nearest-truck change	Reported controlled scenario
INT08	Enter arrival range; notify resident without automatically claiming collection	Arrival reported; state separation needs verification
INT09	Expire request; remove active demand and retain audit record	Expiry implemented; common evidence required
INT10	Fail preferred weather source; show fallback provenance	Hierarchy implemented; provenance test required
INT11	Request forecast; return reproducible zone-week planning record	Internal forecast flow reported
INT12	Publish forecast to dispatch; authorised user sees correct target period	Required; interface not yet implemented
INT13	Complete collection; link outcome to demand and relevant planning run	Required full-system trace
INT14	Change access role; reject unauthorised records and operations	Component controls described; common security test required

A.8 Evaluation Log and Evidence Reconciliation

Every quantitative result should identify the firmware or software release, test period, number of trials, ground-truth procedure and calculation used. A transmission result needs sent and received counts; a false-positive rate needs the number of normal opportunities; and a classification result needs a labelled denominator. A mean latency should identify its start and end events and retain the underlying timings.

For Component 2, the team should reconcile the abstract's figures with the confusion matrix, detection rows and alert table. The anomaly-rule results must identify the firmware or test harness that actually executed the rules because the supplied prototype

description omits them. For Component 1, the team should confirm alert precision and recall separately and correct strict-target pass labels. For Component 3, the team should confirm chronological dates from the records and preserve one authoritative split manifest.

These corrections should precede publication of a combined performance summary. The common report can then be updated with verified figures, while retaining the distinction between component performance and end-to-end service outcomes.

Appendix B Team Contribution Declaration

The contribution record in Table B.1 is based on the component ownership and implementation.

Table B.1 Individual Research and Product Contributions

Member	Student ID	Contribution recorded in the individual thesis
Hemsara K. D. S.	IT22548528	Environmental sensor selection, calibration method, ESP32 firmware, threshold status, telemetry and prototype evaluation
Gamaariya C. C.	IT22202772	Ultrasonic fill sensing, lid-state monitoring, prototype firmware and simulation, Firebase records, LEDs, lid display and critical-event interface
Jayawardhane K. G. M. S.	IT22213358	Municipal feature dataset, wet/dry XGBoost pipelines, weather retrieval, backtesting, fleet and recovery calculations, Flask service and React dashboard
Wijesiri A. M.	IT22088482	Flutter resident/driver/admin workflows, Worker integration, real-time requests and tracking, route support, notifications, authentication and recovery testing

B.1 Academic Integrity and AI Use Disclosure

This thesis is based on our own research, system development, implementation, testing, analysis, and results. ChatGPT and Claude were used only as supporting tools to improve language clarity, sentence structure, and the organization of selected thesis sections. They were not used to create research data, alter experimental results, fabricate references, or make research decisions. All AI-assisted content was reviewed and verified against our original research work, implementation, results, and sources. We take full responsibility for the accuracy, originality, and conclusions presented in this thesis.