

**SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE**  
**A**  
**PROJECT STAGE-I REPORT ON**  
**“Smart Waste Management and Awareness System”**

**SUBMITTED TOWARDS THE  
PARTIAL FULFILLMENT OF THE AWARD OF THE DEGREE OF  
BACHELOR OF TECHNOLOGY (COMPUTER ENGINEERING)**

**BY**

Miss. Vaishnavi Agre [UCS20F1006]  
Miss. Akanksha Barve [UCS20F1020]  
Mr. Sanket Gadekar [UCS20M1038]  
Mr. Aniket Gangurde [UCS20M1041]

**UNDER THE GUIDANCE OF**

**Dr. A. B. Pawar**



**Department of Computer Engineering  
SANJIVANI COLLEGE OF ENGINEERING,  
KOPARGAON-423603  
(AN AUTONOMOUS INSTITUTE)  
2023-2024**

**[17/2023-2024]**



Sanjivani College of Engineering, Kopargaon-423603

(An Autonomous Institute)

DEPARTMENT OF COMPUTER ENGINEERING

## CERTIFICATE

This is to certify that the project entitled

**“Smart Waste Management and Awareness System”**

Submitted by

Miss. Vaishnavi Agre [UCS20F1006]

Miss. Akanksha Barve [UCS20F1020]

Mr. Sanket Gadekar [UCS20M1038]

Mr. Aniket Gangurde [UCS20M1041]

is a bonafide work carried out by students under the supervision of Dr. A. B. Pawar and it is submitted towards the partial fulfillment of the requirement of Bachelor of Technology (Computer Engineering).

During the Academic Year 2023-24

**Dr. A. B. Pawar**

[Internal Guide]

**Dr. S.R. Deshmukh**

[ Project Co-Ordinator]

**Dr. D. B. Kshirsagar**

[Head of Dept.]

**Dr. A. G. Thakur**

[ Director]

**Signature of Internal Examiner**

**Signature of External Examiner**

# **PROJECT APPROVAL SHEET**

**A**  
**PROJECT STAGE- I REPORT**  
**ON**  
**“Smart Waste Management and Awareness System”**

**Is Successfully Completed By**

|                             |                     |
|-----------------------------|---------------------|
| <b>Miss. Vaishnavi Agre</b> | <b>[UCS20F1006]</b> |
| <b>Miss. Akanksha Barve</b> | <b>[UCS20F1020]</b> |
| <b>Mr. Sanket Gadekar</b>   | <b>[UCS20M1038]</b> |
| <b>Mr. Aniket Gangurde</b>  | <b>[UCS20M1041]</b> |

**At**

**DEPARTMENT OF COMPUTER ENGINEERING**

**SANJIVANI COLLEGE OF ENGINEERING,KOPARGAON – 423603**

**(AN AUTONOMOUS INSTITUTE)**

**SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE**

**ACADEMIC YEAR 2023-24**

**Dr. A. B. Pawar**

[Internal Guide]

**Dr. S.R. Deshmukh**

[ Project Co-Ordinator]

**Dr. D. B. Kshirsagar**

[Head of Dept.]

**Dr. A. G. Thakur**

[ Director]

## **ACKNOWLEDGEMENT**

“Smart Waste Management and Awareness System” has been a subject with tremendous scope to research, which leads one’s mind to explore new heights in the field of Computer Engineering, and its miscellaneous applications. We dedicate all our project work to our esteemed guide **Dr. A. B. Pawar** whose interest and guidance helped us to complete the work successfully as well as he has provided facilities to explore the subject with more enthusiasm.

This experience will always encourage us to do our work perfectly and professionally. We also extend our gratitude to **Dr. D. B. Kshirsagar (H.O.D. Computer Engineering)**.

We express our immense pleasure and thankfulness to all the teachers and staff of the Department of Computer Engineering, Sanjivani College of Engineering, Kopargaon for their cooperation and support. Last but not least, we thank all others, and especially our parents and friends, who in one way or another, helped us in the successful completion of this project.

**Miss. Vaishnavi Agre**

**Miss. Akanksha Barve**

**Mr. Sanket Gadekar**

**Mr. Aniket Gangurde**

**(B.Tech Computer Engineering)**

## **ABSTRACT**

The rapid urbanization and population growth have led to an unprecedented increase in the generation of solid waste, posing significant environmental and logistical challenges for cities worldwide. This project introduces a novel approach to address these challenges by implementing a "Smart Waste Management and Awareness System." The primary objectives of this system are to optimize waste collection and disposal processes, enhance environmental sustainability, and promote public awareness of responsible waste management practices. To achieve these goals, the project employs a combination of Internet of Things (IoT) technology, data analytics, and community engagement. Smart bins equipped with sensors and communication devices are strategically placed throughout the college, enabling real-time monitoring of waste levels. This data is collected and processed to optimize waste collection routes, thereby reducing operational costs and carbon emissions.

The application also raises awareness about the importance of sustainable waste management practices and the impact of individual actions on the environment. Preliminary results from the implementation of the Smart Waste Management and Awareness System demonstrate notable improvements in waste collection efficiency and a positive shift in public behaviour towards responsible waste disposal.

This project not only offers an innovative solution to the growing waste management crisis but also contributes to the creation of cleaner, more sustainable urban environments. In conclusion, the Smart Waste Management and Awareness System represents a promising step towards more efficient waste management, reduced environmental impact, and increased public engagement in promoting responsible waste disposal practices. This project exemplifies the potential for technology to address pressing urban challenges and foster a culture of sustainability within communities.

## Contents

|   |           |
|---|-----------|
| <b>1 INTRODUCTION</b>                           | <b>1</b>  |
| 1.1 Problem Definition . . . . .                | 1         |
| 1.2 Literature Review/Relevant Theory . . . . . | 1         |
| 1.3 Scope . . . . .                             | 3         |
| 1.4 Objectives . . . . .                        | 3         |
| <b>2 REQUIREMENT ANALYSIS</b>                   | <b>4</b>  |
| 2.1 Requirement Specifications . . . . .        | 4         |
| 2.1.1 Normal Requirements . . . . .             | 4         |
| 2.1.2 Expected Requirements . . . . .           | 5         |
| 2.1.3 Exciting Requirements . . . . .           | 5         |
| 2.2 Validation of Requirements . . . . .        | 5         |
| 2.3 Functional Requirement . . . . .            | 6         |
| 2.4 Non-Functional Requirement . . . . .        | 7         |
| 2.5 System Requirements . . . . .               | 8         |
| 2.5.1 Hardware Requirements . . . . .           | 8         |
| 2.5.2 Software Requirements . . . . .           | 18        |
| 2.5.3 Networking Requirements . . . . .         | 24        |
| <b>3 SYSTEM MODEL</b>                           | <b>26</b> |
| 3.1 Process Model . . . . .                     | 26        |
| 3.1.1 Incremental Model . . . . .               | 26        |
| 3.1.2 Why to use Incremental Model . . . . .    | 28        |
| 3.1.3 Advantages . . . . .                      | 28        |
| 3.1.4 Disadvantages . . . . .                   | 29        |
| 3.2 Module Details . . . . .                    | 29        |
| 3.2.1 User . . . . .                            | 29        |
| 3.2.2 Admin Module . . . . .                    | 30        |

|          |   |           |
|----------|---|-----------|
| 3.3      | Project Estimation . . . . .                                | 31        |
| 3.3.1    | Estimation in KLOC . . . . .                                | 31        |
| 3.3.2    | Efforts . . . . .   | 31        |
| 3.3.3    | Development time in months . . . . .                        | 32        |
| 3.3.4    | Total Time Required for Project Development . . . . .       | 32        |
| 3.3.5    | Number of Developers (N) . . . . .                          | 33        |
| <b>4</b> | <b>SYSTEM ANALYSIS</b>                                      | <b>34</b> |
| 4.1      | Project Scheduling and Tracking . . . . .                   | 34        |
| 4.1.1    | Project Work and Breakdown Structure (Analysis) . . . . .   | 34        |
| 4.2      | Project work breakdown structure (Implementation) . . . . . | 35        |
| 4.3      | Technologies Used . . . . .                                 | 40        |
| 4.3.1    | WordPress . . . . .   | 40        |
| 4.3.2    | RASA chatbot . . . . .                                      | 42        |
| 4.3.3    | Implementation details about RASA: . . . . .                | 43        |
| 4.3.4    | Blynk App . . . . .   | 44        |
| 4.4      | Task Identification . . . . .                               | 45        |
| 4.5      | Analysis Modeling . . . . .                                 | 46        |
| 4.5.1    | Behavioral modeling . . . . .                               | 47        |
| <b>5</b> | <b>RISK MANAGEMENT</b>                                      | <b>53</b> |
| 5.1      | Risk Identification . . . . .                               | 53        |
| 5.2      | Strategies Used to Manage Risk . . . . .                    | 54        |
| 5.3      | Risk Projection . . . . .                                   | 55        |
| 5.3.1    | Preparing Risk Table . . . . .                              | 55        |
| 5.4      | Feasibility . . . . .                                       | 56        |
| <b>6</b> | <b>TECHNICAL SPECIFICATION</b>                              | <b>57</b> |
| 6.1      | Software requirement specification . . . . .                | 57        |
| 6.1.1    | Operating System (OS) . . . . .                             | 57        |
| 6.1.2    | Integrated Development Environment (IDE) . . . . .          | 57        |
| 6.1.3    | Blynk App . . . . .   | 57        |
| 6.2      | Hardware Requirement Specifications . . . . .               | 57        |
| <b>7</b> | <b>IMPLEMENTATION DETAILS</b>                               | <b>58</b> |
| 7.1      | Smart Bin . . . . .   | 58        |
| 7.2      | RASA Chatbot . . . . .                                      | 59        |

|  |           |
|--|-----------|
| 7.3 Awareness Website . . . . .            | 61        |
| <b>8 APPLICATIONS OF THE PROJECT</b>       | <b>69</b> |
| <b>9 CONCLUSION &amp; FUTURE SCOPE</b>     | <b>71</b> |
| 9.1 Conclusion . . . . .                   | 71        |
| 9.2 Future Scope . . . . .                 | 71        |
| <b>REFERENCES</b>                          | <b>73</b> |
| <b>Annexure A Weekly Assessment Report</b> | <b>74</b> |
| <b>Annexure B Plagiarism Report</b>        | <b>81</b> |

## **List of Figures**

|     |   |    |
|-----|---|----|
| 2.1 | Ultrasonic Sensor . . . . .               | 11 |
| 2.2 | IR Proximity Sensor . . . . .             | 12 |
| 2.3 | Temperature and Humidity Sensor . . . . . | 15 |
| 2.4 | Gas Sensor . . . . .                      | 18 |
| 3.1 | Incremental Model . . . . .               | 27 |
| 4.1 | System Architecture . . . . .             | 36 |
| 4.2 | Gateway structure . . . . .               | 38 |
| 4.3 | Breakdown Structure . . . . .             | 39 |
| 4.4 | Use Case Diagram . . . . .                | 48 |
| 4.5 | Sequence Diagram . . . . .                | 49 |
| 4.6 | Class Diagram . . . . .                   | 50 |
| 4.7 | Activity Diagram . . . . .                | 51 |
| 4.8 | State Chart Diagram . . . . .             | 52 |
| 4.9 | Deployment Diagram . . . . .              | 52 |
| 7.1 | Smart Bin . . . . .                       | 59 |
| 7.2 | RASA Chatbot Implementation . . . . .     | 60 |
| 7.3 | Home Page . . . . .                       | 62 |
| 7.4 | Service Page . . . . .                    | 63 |
| 7.5 | Awareness Page . . . . .                  | 65 |
| 7.6 | Contact Page . . . . .                    | 66 |
| 7.7 | Chatbot Page . . . . .                    | 66 |
| 7.8 | Real-time Execution . . . . .             | 67 |
| 9.1 | Weekly Assessment Report 1 . . . . .      | 74 |
| 9.2 | Weekly Assessment Report 2 . . . . .      | 75 |

|      |                                      |    |
|------|--------------------------------------|----|
| 9.3  | Weekly Assessment Report 3 . . . . . | 76 |
| 9.4  | Weekly Assessment Report 4 . . . . . | 77 |
| 9.5  | Weekly Assessment Report 5 . . . . . | 78 |
| 9.6  | Weekly Assessment Report 6 . . . . . | 79 |
| 9.7  | Weekly Assessment Report 7 . . . . . | 80 |
| 9.8  | Plagiarism Report . . . . .          | 81 |
| 9.9  | Plagiarism Report . . . . .          | 82 |
| 9.10 | Plagiarism Report . . . . .          | 83 |

## **List of Tables**

|   |    |
|---|----|
| 3.1 Time Required for Project Development . . . . . | 32 |
| 5.1 Risk Management Table . . . . .                 | 55 |

# Chapter 1

## INTRODUCTION

### 1.1 Problem Definition

Waste management has become a critical global issue as urbanization and population growth contribute to the generation of vast amounts of waste. Inefficient waste disposal practices lead to environmental pollution, health hazards, and resource depletion. Moreover, lack of public awareness about responsible waste disposal exacerbates the problem. The existing waste management systems often suffer from inefficiencies, resulting in improper disposal, increased landfill usage, and associated environmental consequences. Additionally, there is a significant lack of awareness among the public regarding the importance of proper waste disposal and recycling practices.

### 1.2 Literature Review/Relevant Theory

Alam and Dey (2019) introduced an IoT-based solid waste management system tailored for smart cities, emphasizing technological innovation and exploration. Additionally, the Central Pollution Control Board in its 2016 annual report (Reference [2]) highlighted the importance of robust waste management strategies at the governmental level.

Gharaibeh et al. (2017) delved into the concept of smart waste management using IoT, contributing insights from the 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT). Kanchan and Sharma (2020) extended this discourse with their study on IoT-based waste management for smart cities, further emphasizing the role of technology in optimizing waste processes.

A comprehensive review by Kumar et al. (2017) explored various technological options for waste-to-energy, shedding light on effective approaches to managing municipal solid waste. Meanwhile, the Ministry of Environment, Forest, and Climate Change in In-

dia provided a regulatory framework through the Solid Waste Management Rules of 2016 (Reference [6]).

Rana and Goyal (2019) contributed to the literature by discussing the challenges and solutions associated with implementing IoT in solid waste management, as published in the International Journal of Computer Applications. Finally, Sharma et al. (2020) addressed the impact of the COVID-19 pandemic on solid waste management, presenting challenges, opportunities, and innovations in the post-pandemic era in their publication in Resources, Conservation, and Recycling.

### **1.3 Scope**

1. Develop a Smart Waste Management and Awareness System.
2. Target audience: Sanjivani college community and beyond.
3. Integration of Smart Bins, chatbot, animated videos, and comprehensive website.
4. Deployment of IoT sensors on waste bins to monitor and transmit real-time data on fill levels.
5. Development of a centralized server infrastructure for efficient storage, processing, and analysis of data collected from sensors.
6. Implementation of a smart routing algorithm to optimize waste collection routes based on fill-level data, minimizing fuel consumption and reducing the environmental impact.
7. Creation of a user-friendly mobile application to provide residents with real-time information on waste levels, recycling locations, and guidelines for responsible waste disposal.

### **1.4 Objectives**

- To Improve waste management existing system to provide real-time data on waste collection and recycling centers.
- To Enhance user engagement through a userfriendly chatbot.
- To Integrate animated videos for informative and interactive content.
- To Complete website development and host for public accessibility.

## Chapter 2

# REQUIREMENT ANALYSIS

Requirements Analysis or requirement engineering is a process of determining user expectations for new software or providing updates for previous products. These core points must be measurable, relevant and detailed. In the software engineering field this term is also called functional specifications. Requirements analysis mainly deals with communication with users or customers to determine system feature expectations, requirements and reduce convicts as demanded by various software users. Energy should be directed towards ensuring that the system or product conforms to user needs rather than attempting to turn user expectations to the requirements.

### 2.1 Requirement Specifications

Requirement specification describes the function and performance of the computer based system and constraints which govern its development. It can be a written document, a set of graphical models, a collection of scenarios, or any combination of above. These are of 3 types:

1. NR: Normal Requirements
2. ER: Expected Requirements
3. XR: Exciting Requirements

#### 2.1.1 Normal Requirements

These are the requirements which are clearly stated by the customer so all these requirements will be present in the project for user satisfaction.

- NR1: The system should target audience for awareness: Sanjivani college community and beyond.
- NR2: Reduces the effect of waste on the environment and health.
- NR3: Enhance user engagement through a userfriendly chatbot.

### **2.1.2 Expected Requirements**

These requirements are expected by the customer but not clearly stated by the customer. These are implicit types of requirements.

- ER1: System should be fast and reliable.
- ER2: System should have a neat and clean user interface.
- ER3: System should provide instruction of usage to the user.

### **2.1.3 Exciting Requirements**

These requirements are not stated by the customer but externally provided by the developer in order to maintain a good relationship with the customer.

- XR1: : It should be able to accurately track the fill level of bins.
- XR2: It should be able to send alert when bins are full.

## **2.2 Validation of Requirements**

Requirements validation is the process of checking that requirements defined for development, define the system that the customer really wants. To check issues related to requirements, we perform requirements validation. We usually use requirements validation to check error at the initial phase of development as the error may increase excessive rework when detected later in the development process.

In the requirements validation process, we perform a different type of test to check the requirements mentioned in the Software Requirements Specification (SRS), these checks include:

- Completeness checks

- Consistency checks
- Validity checks
- Realism checks
- Ambiguity checks
- Verifiability

The output of requirements validation is the list of problems and agreed on actions of detected problems. The lists of problems indicate the problem detected during the process of requirement validation. The list of agreed action states the corrective action that should be taken to fix the detected problem.

The development of software begins once the requirements document is ready. One of the objectives of this document is to check whether the delivered software system is acceptable. For this, it is necessary to ensure that the requirements specification contains no errors and that it specifies the users requirements correctly. Also, errors present in the SRS will adversely affect the cost if they are detected later in the development process or when the software is delivered to the user. Hence, it is necessary to detect errors in the requirements before the design and development of the software begins. To check all the issues related to requirements, requirements validation is performed. In the validation phase, the work products produced as a consequence of requirements engineering are examined for consistency, omissions, and ambiguity. The basic objective is to ensure that the SRS reflects the actual requirements accurately and clearly. The requirement checklist as follows:

1. Are all requirements consistent?
2. Are the requirements really necessary?
3. Is each requirement testable?
4. Does the requirement model properly reflect the information function and behaviour of the system to be built?

### **2.3 Functional Requirement**

- Integrate sensors with waste bins to measure fill levels in real time.

- The system shall be able to generate alerts when waste bins are full.
- The system shall be able to provide real-time data on waste generation and disposal.
- The system shall be able to generate reports on waste trends and patterns.
- The system shall be able to provide educational resources on waste management and recycling.
- The system shall be able to track the location and fill level of waste bins.
- Display comprehensive information on responsible waste disposal practices, recycling guidelines, and the environmental impact of waste.
- Allow users to create accounts with personalized profiles to track their engagement and contributions to waste management initiatives.
- Include interactive content such as videos, infographics, and quizzes to engage and educate visitors on waste management topics.
- Provide a dedicated section for news and updates related to waste management initiatives, events, and success stories.

## 2.4 Non-Functional Requirement

- The system shall be secure and accessible to authorized users only.
- The system shall be reliable and available 24/7.
- The system shall be scalable to accommodate future growth.
- The system shall be easy to use and maintain.
- The system shall be affordable and cost-effective.
- The website should load within three seconds on standard internet connections to ensure a seamless user experience.
- The website should be scalable to accommodate a growing user base and increased content without compromising performance.

- The website must be available 99.9
- Implement secure socket layer (SSL) encryption to protect user data during transmission, and ensure compliance with industry standards for data security.
- The website should be compatible with major web browsers (Chrome, Firefox, Safari, Edge) and maintain consistent functionality across different devices and screen sizes.

## 2.5 System Requirements

Requirements specify what features a product should include and how those features should work. They help to define the test criteria, which is vital for verification and validation.

### 2.5.1 Hardware Requirements

1. RAM: 4GB (min)
2. Ultrasonic Sensor

An ultrasonic sensor is a device that uses ultrasonic sound waves to measure distance or detect the presence of an object. These sensors are widely used in various applications, including robotics, industrial automation, automotive, and even in some consumer electronics. Here are key features and information about ultrasonic sensors:

**Principle of Operation:** Ultrasonic sensors work based on the principle of sending and receiving ultrasonic waves (sound waves with a frequency above the human audible range). The sensor typically has a transducer that emits ultrasonic waves, and another transducer that receives the waves after they bounce off an object.

**Distance Measurement:** Ultrasonic sensors can be used to measure the distance between the sensor and an object by calculating the time it takes for the sound waves to travel to the object and back. The formula for distance (D) is often given by  $D = (\text{speed of sound} \times \text{time of flight}) / 2$ .

**Components:** Transducer: The transducer is the core component that converts electrical energy into ultrasonic waves and vice versa.

**Signal Processing Unit:** Some sensors have a built-in microcontroller or signal processing unit to handle the generated signals and calculate distances.

**Trigger and Echo Pins:** Many ultrasonic sensors have separate trigger and echo pins for initiating the measurement and receiving the echo, respectively.

**Operating Range:** Ultrasonic sensors have a specific operating range, which is the distance within which they can accurately measure distance. This range can vary depending on the sensor model.

**Accuracy and Resolution:** The accuracy of ultrasonic sensors depends on factors such as the sensor design, frequency of ultrasonic waves, and environmental conditions. Resolution refers to the smallest change in distance that the sensor can detect.

**Applications:** Ultrasonic sensors find applications in various fields, including:

**Obstacle Detection:** Used in robotics and vehicles for avoiding obstacles.

**Distance Measurement:** Commonly used for measuring liquid levels, object distances, etc.

**Proximity Sensing:** Detecting the presence or absence of objects within a certain range.

**Limitations:** Ultrasonic sensors may face challenges in certain environments, such as areas with a lot of sound interference. The accuracy of measurements can be affected by the material and surface characteristics of the objects being detected.

**Mounting and Installation:** Proper mounting and installation are crucial for accurate readings. The sensor should be positioned to avoid false readings caused by reflections or interference.

**Variants:** There are different types of ultrasonic sensors, including single transducer sensors, array transducer sensors, and even some with multiple transducers for improved accuracy.

**Integration with Microcontrollers:** Ultrasonic sensors are often integrated into projects using microcontrollers like Arduino or Raspberry Pi. These microcontrollers can process the sensor data and trigger specific actions based on the measurements. When working with ultrasonic sensors, it's important to refer to the specific datasheet and documentation provided by the manufacturer for accurate details regarding the sensor's specifications, usage, and recommended operating conditions.

Ultrasonic sensors are commonly used for distance measurement and object detection by emitting ultrasonic waves and measuring the time it takes for the waves to bounce

back after hitting an object. The basic ultrasonic sensor module typically has three pins. Here's a breakdown of the usage of each pin:

VCC (or VDD, Power, +5V):

Function: This pin is used for providing power to the ultrasonic sensor. Voltage: Most ultrasonic sensors operate at a voltage of +5V, but it's essential to check the specifications of the specific sensor you are using. Some sensors might operate at different voltage levels, such as 3.3V.

Trig (Trigger):

Function: The Trig pin is used to trigger the ultrasonic sensor to send out a short ultrasonic pulse. Operation: When a pulse (usually a high-level pulse) is applied to this pin, the sensor emits a series of ultrasonic pulses.

Echo:

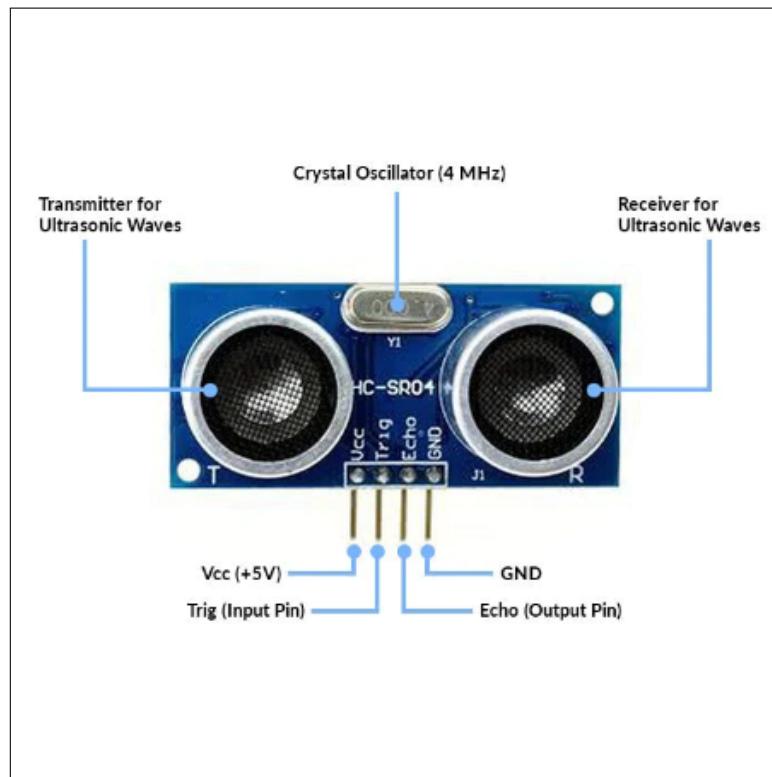
Function: The Echo pin is used to receive the echo of the ultrasonic pulse. Operation: When the ultrasonic pulse emitted by the sensor reflects off an object and returns, the Echo pin produces a pulse that is proportional to the time it took for the ultrasonic waves to travel to the object and back.

GND (Ground):

Function: This pin is the ground or common reference for the electrical circuit. Voltage: It is connected to the ground or 0V reference of the power supply.

Here's a basic sequence of how an ultrasonic sensor operates:

The Trig pin is given a short pulse (typically 10 microseconds) to initiate the transmission of ultrasonic waves. The ultrasonic waves travel through the air and hit an object. The waves are then reflected back to the sensor. The Echo pin produces a pulse whose duration is proportional to the time it takes for the waves to travel to the object and back. The microcontroller or other processing unit connected to the sensor can measure the duration of the Echo pulse and use it to calculate the distance to the object using the speed of sound in the air. The actual usage might vary slightly depending on the specific model and manufacturer, so it's always recommended to refer to the datasheet or documentation provided by the sensor manufacturer for accurate and detailed information.



**Figure 2.1:** Ultrasonic Sensor

### 3. IR Proximity Sensor

Infrared (IR) proximity sensors are devices that can detect the presence or absence of an object within a certain range by using infrared radiation. These sensors work based on the principles of reflection, absorption, and emission of infrared light. Here's a general overview of how IR proximity sensors work:

**Emitter and Detector:** IR proximity sensors consist of two main components: an infrared emitter and an infrared detector. The emitter sends out infrared light, and the detector receives the reflected or emitted light.

**Infrared Light:** The emitter typically emits infrared light in the form of invisible radiation. This light is in the infrared spectrum, beyond the range of human vision.

**Reflection:** When there is no obstacle in the proximity of the sensor, the emitted infrared light reflects off a surface and is detected by the receiver. The sensor is designed to detect the reflection and determine that there is no obstacle in the detection zone.

**Obstacle Presence:** If an object comes into the detection range of the sensor, it inter-

rupts the path of the emitted infrared light. The infrared light reflects off the object in a different direction or gets absorbed by the object, and less light reaches the detector.

**Signal Processing:** The detector measures the amount of reflected or received infrared light. A decrease in the received light indicates the presence of an object. This change in the received signal triggers the sensor to generate a response, such as activating an output signal or triggering an alarm.

**Distance Measurement:** Some advanced IR proximity sensors can also be used for distance measurement. This is done by analyzing the time taken for the infrared light to travel to the object and back. By calculating the time of flight, the sensor can estimate the distance to the object.

**Applications:** IR proximity sensors are commonly used in various applications, including automatic faucets, paper towel dispensers, mobile phones (for proximity sensing during calls), robotics, security systems, and many other scenarios where the detection of nearby objects is essential.

IR proximity sensors are popular due to their reliability, cost-effectiveness, and versatility in different applications. They are widely used for object detection and distance measurement in both industrial and consumer electronics.



**Figure 2.2:** IR Proximity Sensor

#### 4. Temperature and Humidity sensor

Integrating temperature and humidity sensors into a smart bin can offer additional functionalities and data insights. Here's how these sensors can be implemented in a smart bin:

**Environment Monitoring:** Temperature and humidity sensors can monitor the ambient conditions inside and around the smart bin. This information can be useful for assessing the environment where the bin is placed, especially in areas where temperature and humidity levels may affect waste decomposition or the performance of electronic components.

**Optimal Waste Storage:** By monitoring temperature and humidity, the smart bin can provide insights into the optimal conditions for waste storage. For example, certain types of waste may benefit from specific temperature or humidity levels to control odors or decomposition rates.

**Preventing Odor Issues:** The sensors can be used to detect conditions that might contribute to unpleasant odors. When the humidity level rises or specific temperature thresholds are reached, the system could trigger alerts or actions to address potential odor issues, such as activating ventilation or sealing mechanisms.

**Real-Time Monitoring:** Users can access real-time temperature and humidity data through a mobile app or other interfaces. This feature allows them to stay informed about the environmental conditions around the smart bin.

**Smart Ventilation:** Integration with a ventilation system can be based on the readings from the temperature and humidity sensors. When certain conditions are met, the smart bin could activate a ventilation mechanism to control internal conditions and minimize odors.

**Waste Quality Monitoring:** The combination of temperature and humidity data can be used to assess the quality of the waste inside the bin. For instance, high humidity levels may indicate the presence of wet or organic waste.

**Custom Alerts and Notifications:** Users can set up custom alerts based on temperature and humidity thresholds. This allows them to receive notifications when conditions deviate from the desired range or when specific events occur.

**Energy Efficiency:** Implementing energy-efficient algorithms based on sensor data can optimize the operation of additional components, such as ventilation systems or lid mechanisms, contributing to overall energy efficiency.

**Data Analytics and Insights:** Collecting historical data from temperature and humidity sensors enables users to analyze trends and make informed decisions about waste

management practices. This data could be valuable for municipal waste management authorities or businesses looking to optimize waste collection and disposal.

**Integration with Other Sensors:** Temperature and humidity data can be correlated with data from other sensors, such as weight sensors, to provide a more comprehensive understanding of waste conditions and optimize waste management strategies.

**Safety Features:** Implement safety features to ensure that the sensors are protected from moisture and other environmental factors to maintain accurate readings over time. Integrating temperature and humidity sensors into a smart bin enhances its capabilities and provides valuable environmental data. It's important to consider factors such as sensor placement, calibration, and the algorithms used to interpret the data to ensure accurate and meaningful results.

#### VCC or VDD (Power Supply Pin):

**Function:** This pin is used to supply power to the temperature and humidity sensor.

**Voltage:** The voltage level required may vary based on the sensor model, but commonly it operates at +5V. Some sensors are compatible with +3.3V.

#### Data (or Signal, Out):

**Function:** The data pin is used to transmit the digital signal carrying temperature and humidity information. **Output Type:** The data is usually in the form of a digital signal, often a pulse-width modulation (PWM) signal or a serial data stream.

#### GND (Ground Pin):

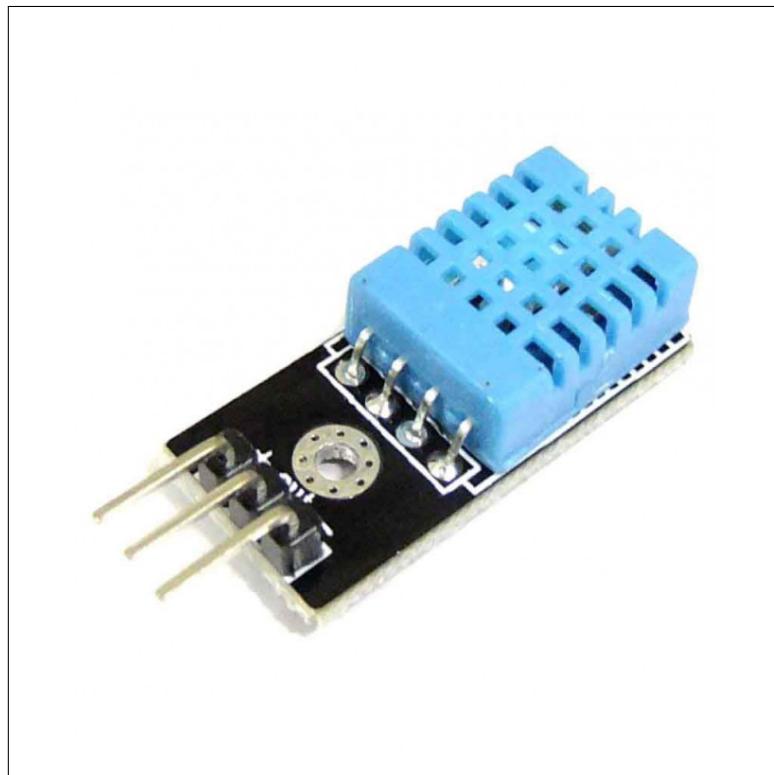
**Function:** The ground pin is the reference point for the electrical circuit. **Voltage:**

Connected to the ground or 0V reference of the power supply.

Here's a basic sequence of how a digital temperature and humidity sensor operates:

The sensor is powered by applying the appropriate voltage to the VCC pin. The microcontroller or data-logging device initiates communication by sending a signal to the sensor through the data pin. The sensor responds by transmitting a digital signal containing the current temperature and humidity data. The microcontroller interprets the digital signal to obtain temperature and humidity values. It's important to note that different sensor models may have additional pins or slightly different functionalities. For example, some sensors may include pins for calibration or for connecting to an external reference voltage. Always refer to the datasheet or product documentation

provided by the sensor manufacturer for accurate and detailed information on the pin configuration and functionality of a specific temperature and humidity sensor.



**Figure 2.3:** Temperature and Humidity Sensor

## 5. Gas Sensor

Gas sensors are devices that detect the presence of specific gases in the surrounding environment. These sensors are commonly used for monitoring air quality, detecting leaks, and ensuring safety in various applications. The working principle of gas sensors depends on the type of sensor technology employed. Here are a few common types of gas sensors and their working principles:

Chemiresistive Gas Sensors:

**Working Principle:** These sensors operate based on changes in electrical resistance when exposed to target gases. **Structure:** The sensing element typically consists of a material whose resistance changes in the presence of specific gases. This material is often a metal oxide semiconductor. **Operation:** When the target gas interacts with the sensing material, it causes a change in the material's conductivity or resistance. This

change is then measured, and the concentration of the gas is determined based on the degree of resistance change.

#### **Electrochemical Gas Sensors:**

**Working Principle:** Electrochemical sensors detect gases through chemical reactions that produce an electric current. **Structure:** These sensors have a cell containing electrodes and an electrolyte. The electrodes are coated with materials that react with specific gases. **Operation:** When the target gas comes into contact with the electrode, a chemical reaction occurs, generating ions and electrons. This reaction produces a current that is proportional to the gas concentration, allowing for the measurement of gas levels.

#### **Infrared Gas Sensors:**

**Working Principle:** Infrared sensors detect gases by analyzing the absorption of infrared light at specific wavelengths. **Structure:** These sensors typically consist of an infrared light source and a detector. The detector measures the amount of infrared light absorbed by the target gas. **Operation:** Each gas absorbs infrared light at unique wavelengths. By analyzing the absorption spectrum, the sensor can identify and quantify the concentration of the gas.

#### **Photoionization Gas Sensors:**

**Working Principle:** Photoionization sensors detect gases by measuring the ionization of gas molecules when exposed to ultraviolet (UV) light. **Structure:** These sensors have a UV light source and a detector. When the UV light interacts with gas molecules, it ionizes them, producing a measurable current. **Operation:** The strength of the current is proportional to the concentration of the target gas, allowing for the determination of gas levels.

#### **Metal Oxide Semiconductor (MOS) Gas Sensors:**

**Working Principle:** MOS sensors operate based on changes in the electrical conductivity of a metal oxide semiconductor when exposed to gases. **Structure:** The sensing material is often tin dioxide ( $\text{SnO}_2$ ) or another metal oxide. When the material comes into contact with specific gases, its conductivity changes.

**Operation:** The sensor measures the change in conductivity, and this change is correlated with the concentration of the target gas. Gas sensors are crucial in various

applications, including industrial settings, environmental monitoring, and safety systems, helping to detect and respond to the presence of hazardous gases. The selection of a specific type of gas sensor depends on the target gases, sensitivity requirements, and environmental conditions.

common pins found in gas sensors and their typical functions:

**VCC or VDD (Power Supply Pin):**

Function: This pin is used to supply power to the gas sensor. Voltage: The voltage level required may vary based on the sensor model, so it's essential to check the datasheet or product documentation. Common voltage levels include +5V or +3.3V.

**GND (Ground Pin):**

Function: The ground pin is the reference point for the electrical circuit. Voltage: Connected to the ground or 0V reference of the power supply. Analog Output (AO) or Digital Output (DO):

Function: Gas sensors often have an output pin that provides an analog or digital signal related to the detected gas concentration. Analog Output (AO): Provides a continuous voltage or current signal that varies proportionally with the gas concentration. Digital Output (DO): Provides a digital signal (typically high or low) based on a threshold level. This pin is often used with a comparator to set a specific concentration threshold.

**Digital Trigger Input (DT) or Analog Voltage Input (AVI):**

Function: Some gas sensors may have a pin for triggering specific actions or modes. Digital Trigger Input (DT): Used for providing a digital trigger signal to initiate a specific operation. Analog Voltage Input (AVI): Used for providing an external reference voltage to calibrate or adjust the sensor.

**Heater Pins (H or H1, H2):**

Function: Gas sensors with metal oxide semiconductor (MOS) technology often include heater pins to heat the sensing element. Purpose: Heating the sensing element improves the sensor's sensitivity and response time. The heater pins are used to control the temperature of the sensing element.

**Reference (REF) or Bias (B):**

Function: Some gas sensors may have a reference or bias pin for setting a reference

voltage or biasing the sensor. Purpose: This pin helps optimize the sensor's performance and stability.



**Figure 2.4:** Gas Sensor

6. Processor: Intel core i3 8th Gen
7. 64-bit CPU

### 2.5.2 Software Requirements

- Operating system: Windows/Linux/Mac
- Cloud Services
- Any compatible Browser
- WordPress Tool

WordPress is a popular content management system (CMS) and website creation tool that allows users to build and manage websites without extensive coding knowledge.

Here is some information about WordPress:

#### Open Source Platform:

WordPress is an open-source platform, meaning its source code is freely available for anyone to use, modify, and distribute. This has contributed to its widespread adoption and a large community of developers.

#### Content Management System (CMS):

WordPress functions as a CMS, allowing users to create, edit, organize, and publish content on their websites. It provides an intuitive interface for managing text, images, multimedia, and more.

#### Themes:

WordPress allows users to change the appearance of their websites easily through the use of themes. Themes are pre-designed templates that control the layout, design, and overall look of a site.

#### Plugins:

Plugins extend the functionality of WordPress. Users can add features such as contact forms, e-commerce capabilities, SEO tools, social media integration, and more by installing and activating plugins.

#### User-Friendly Interface:

WordPress is known for its user-friendly interface, making it accessible to users with varying levels of technical expertise. Its "What You See Is What You Get" (WYSIWYG) editor allows users to create and edit content visually.

#### Community Support:

The WordPress community is vast and active. Users can find support through forums, online documentation, tutorials, and community events. Additionally, developers contribute to the improvement of WordPress through core updates.

#### Customization:

WordPress is highly customizable. Users can customize the appearance and functionality of their sites by modifying themes and using custom CSS. Developers can create custom plugins and themes to meet specific needs.

#### Multilingual Support:

WordPress has built-in support for multiple languages, allowing users to create websites in their preferred language. This feature is useful for reaching a global audience.

#### Regular Updates:

WordPress undergoes regular updates to improve security, fix bugs, and introduce new features. Users are encouraged to keep their installations up to date to benefit from the latest improvements.

#### Responsive Design:

Modern WordPress themes are designed to be responsive, ensuring that websites display correctly on various devices, including desktops, tablets, and smartphones.

#### E-commerce Capabilities:

With plugins like WooCommerce, WordPress can be turned into a full-fledged e-commerce platform, enabling users to set up online stores and sell products or services. Whether you're a blogger, small business owner, or developer, WordPress provides a flexible and powerful platform for creating and managing websites. Its versatility and extensive ecosystem of themes and plugins make it suitable for a wide range of web projects.

- **Blynk App**

Blynk is a popular Internet of Things (IoT) platform that enables users to create custom mobile applications to control and monitor connected hardware projects. It provides a drag-and-drop interface for building mobile apps without the need for extensive programming knowledge. Here are some key aspects of the Blynk app:

**IoT Platform:** Blynk serves as a platform for building IoT applications. It allows users to connect hardware devices to the internet and control them remotely through a mobile app.

**Mobile App Builder:** Blynk offers a mobile app builder that simplifies the process of creating custom interfaces for IoT projects. Users can design the app's layout, add buttons, sliders, graphs, and other widgets using a visual interface.

**Widgets:** Blynk provides a variety of widgets that users can add to their mobile apps. These widgets include buttons, sliders, displays, gauges, and more, allowing for the creation of interactive and dynamic interfaces.

**Supported Platforms:** Blynk supports a wide range of hardware platforms, including Arduino, Raspberry Pi, ESP8266, ESP32, and others. This flexibility allows users to connect various sensors, actuators, and devices to their IoT projects.

**Communication Protocols:** Blynk supports multiple communication protocols such as Wi-Fi, Bluetooth, Ethernet, and USB. This enables users to connect their hardware devices to the internet and control them wirelessly.

**Cloud Connectivity:** Blynk uses cloud services to facilitate communication between the mobile app and connected hardware. Users can remotely monitor and control their devices from anywhere with an internet connection.

**Open-Source Library:** Blynk provides open-source libraries for popular hardware platforms and programming languages. These libraries simplify the process of integrating Blynk into your hardware projects.

**Energy System:** Blynk uses an "energy" system to manage resources within the app. Each widget and feature consumes a certain amount of energy, and users are allocated a specific amount of energy for their projects. Additional energy can be purchased if needed.

**Security:** Blynk takes security seriously and provides features such as secure connections (SSL/TLS) and authentication tokens to protect communication between the app and connected devices.

**Community and Documentation:** Blynk has an active community where users can share their projects, seek help, and collaborate. The platform also provides comprehensive documentation and tutorials to help users get started with building their IoT projects. Whether you're a hobbyist or a professional developer working on IoT projects, Blynk offers a user-friendly and versatile platform for building custom mobile apps to control and monitor connected hardware.

- **RASA**

RASA is an open-source platform for building conversational AI applications, including chatbots and virtual assistants. It provides a set of tools and libraries for developing natural language understanding (NLU) and dialogue management capabilities. RASA is designed to be flexible, allowing developers to create custom, context-aware chatbots that can understand and respond to user inputs in a more sophisticated manner.

Here are some key components and features of RASA:

**RASA NLU (Natural Language Understanding):** This component focuses on understanding user inputs by extracting intent and entities from natural language text. Developers can define training data that includes examples of user messages, intents (the user's intention), and entities (specific pieces of information within the user's message).

**RASA Core:** RASA Core is responsible for managing the dialogue and deciding how the chatbot should respond to user inputs. It uses machine learning to learn from conversations and predict the next best action based on the current context.

**Domain Configuration:** The domain configuration file (domain.yml) defines the actions the bot can take, the intents it can understand, and the entities it can extract.

**Custom Actions:** Developers can define custom actions that the bot can take in response to user inputs. These actions can include making API calls, querying databases, or performing other tasks.

**Training Data:** RASA chatbots are trained on user conversations and labeled data. Developers provide examples of conversations and correct responses, which are used to train the models for NLU and dialogue management.

**Flexible Deployment:** RASA chatbots can be deployed on various platforms, including local servers, cloud platforms, and containers (e.g., Docker). This flexibility allows developers to choose the deployment option that best fits their needs.

**Community and Ecosystem:** RASA has an active community of developers and a growing ecosystem. This community support includes forums, documentation, and contributions from developers worldwide.

**Open Source:** RASA is open-source software, meaning that its source code is freely available for developers to view, modify, and contribute to. This openness encourages collaboration and innovation.

**Integration:** RASA can be integrated into various messaging platforms, such as Slack, Facebook Messenger, or custom chat interfaces, enabling developers to deploy their chatbots where their users are. Overall, RASA provides a comprehensive framework for building intelligent and context-aware chatbots, making it a popular choice for developers working on conversational AI applications.

- Implementation details about RASA:

Installation: You can install RASA by using pip. The recommended way is to create a virtual environment and install RASA within that environment.

- pip install rasa

Project Initialization: Create a new RASA project using the following command:

- rasa init

This will create a basic project structure with necessary files, including domain.yml, config.yml, and data files.

Training Data: RASA uses training data to understand user inputs and learn how to respond. The training data includes conversations, intents, entities, and responses. Conversations are typically stored in files like stories.md or data/nlu.md. Intent examples, entity examples, and responses are defined in these files.

Domain Configuration: The domain.yml file defines the domain of your assistant, including intents, entities, actions, and responses. It's crucial for training the model to understand the context and generate appropriate responses.

Configuration File: The config.yml file contains configuration details for training the RASA models. It includes information about the pipeline, policies, and other training-related parameters.

Training Models: Train the RASA NLU model using the following command:

- rasa train nlu

Train the RASA Core model using:

- rasa train

Running the Bot: Once the models are trained, you can run the RASA server using:

- rasa run

This will start a server, and you can interact with your bot through a REST API.

Interactive Learning: RASA provides a feature called interactive learning, where you can have a conversation with your bot and correct its responses in real-time. This helps improve the model iteratively.

Integration: RASA can be integrated into various channels such as Slack, Facebook Messenger, or custom chat interfaces. You can find integration details in the RASA documentation.

**Deployment:** Deploy your RASA bot using various deployment options, including Docker, cloud platforms like AWS or Azure, or any server that supports Python applications.

**Continuous Training:** As your bot interacts with users and receives new data, it's essential to retrain your models periodically to improve performance.

### 2.5.3 Networking Requirements

- **Internet Connectivity:** The system requires reliable and high-speed internet connectivity for real-time data transmission between waste bin sensors, the centralized server, and the mobile application.
- **Security Protocols:** Implement secure communication protocols, such as HTTPS, to encrypt data transmitted over the network and ensure the confidentiality and integrity of sensitive information.
- **Firewall Configuration:** Set up firewalls to protect the system from unauthorized access and potential cyber threats, allowing only essential and secure network traffic.
- **Redundancy and Failover:** Establish redundant network connections and failover mechanisms to ensure continuous data transmission even in the event of a network failure or disruption.
- **Quality of Service (QoS):** Implement Quality of Service measures to prioritize network traffic, ensuring that critical data related to waste level monitoring and route optimization receives higher priority.
- **Low Latency:** The network must provide low-latency communication to support real-time data updates from waste bin sensors and enable timely response in waste collection route optimization.
- **Scalability:** Design the network infrastructure to be scalable, accommodating an increasing number of sensors, users, and data traffic as the system expands.
- **IPv6 Compatibility:** Ensure compatibility with IPv6 to accommodate the growing number of connected devices and address the limitations of IPv4 addressing.

- Network Monitoring: Implement network monitoring tools to continuously monitor network performance, detect anomalies, and troubleshoot issues promptly.
- Bandwidth Management: Implement bandwidth management mechanisms to optimize data transfer efficiency and prevent network congestion, especially during peak usage periods.
- Mobile Network Compatibility: Ensure compatibility with mobile networks to support data transmission to and from the mobile application, considering different network technologies (3G, 4G, 5G).
- Geographic Redundancy: Consider geographic redundancy for the server infrastructure to enhance system resilience and minimize downtime in case of a localized network failure.
- Remote Access Security: If remote access is required for system maintenance or updates, implement secure remote access protocols and authentication mechanisms to prevent unauthorized access.
- Network Segmentation: Segment the network into different zones with appropriate access controls to isolate sensitive components, such as the centralized server, from less critical parts of the system.
- Energy-Efficient Networking: Implement energy-efficient networking practices to minimize the environmental impact of the system and reduce power consumption.
- Integration with Municipal Networks: Ensure seamless integration with existing municipal networks, allowing for efficient communication with waste management authorities and other relevant stakeholders.
- Compliance with Data Protection Laws: Adhere to data protection regulations when transmitting and storing data across networks, ensuring compliance with privacy laws.
- Network Documentation: Maintain comprehensive documentation of the network architecture, configurations, and protocols for ease of management, troubleshooting, and future expansions.

## Chapter 3

### SYSTEM MODEL

Software process model is an intellectual demonstration of a process. It presents an explanation of a process. Process models may contain activities that are part of the software process. All software process models work on the five generic framework activities such as communication, planning, modelling, construction, and deployment. Each and every activity has its own functionality. The goal of the process model is to provide guidance for systematically coordinating and monitoring the tasks that must be accomplished in order to achieve the end product and the project objective.

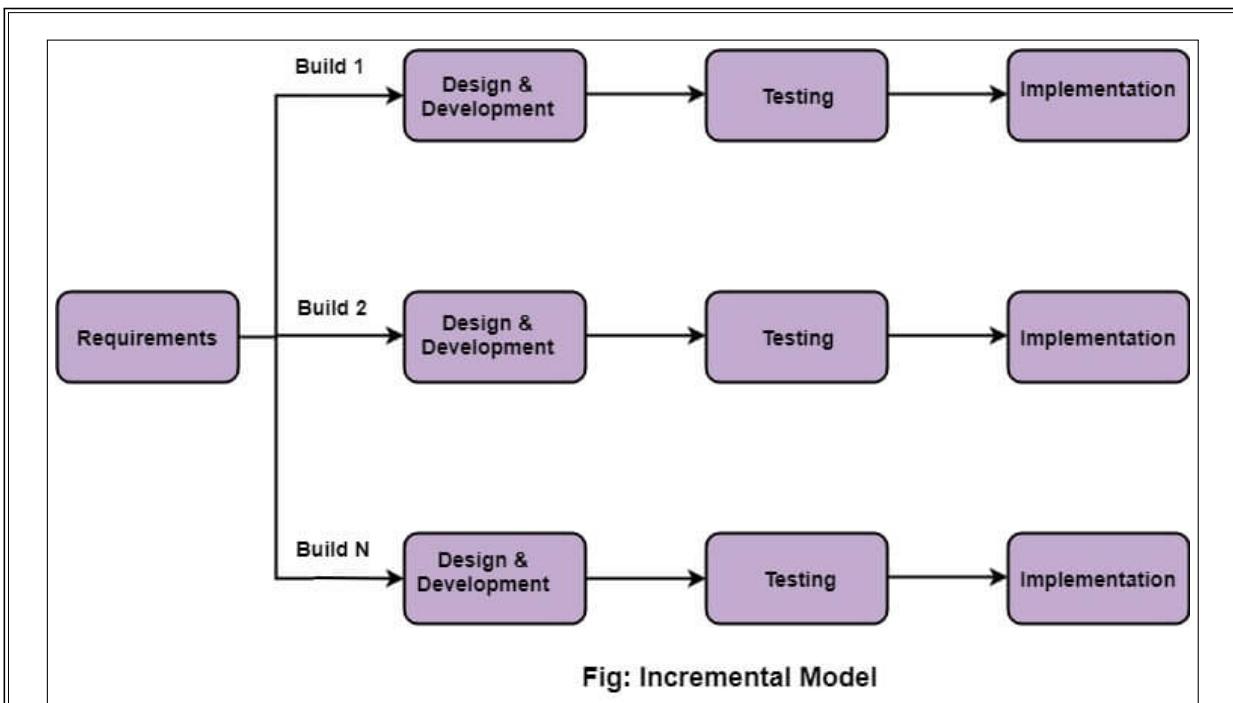
#### 3.1 Process Model

Software process model is an abstract representation of a process. The goal of process model is to provide guidance for systematically coordinating and controlling the tasks that must be performed in order to achieve the end product and the project objective. Incremental model is used as the process model in our system.

##### 3.1.1 Incremental Model

Incremental model in software engineering is a one which associates the elements of waterfall model in an iterative manner. It delivers a series of releases called increments which provide progressively more functionality for the user as each increment is delivered. In the incremental model of software engineering, the waterfall model is frequently applied in each increment. The incremental model applies linear sequences in a required pattern as calendar time passes. Each linear sequence produces an increment in the work.

A, B, C are modules of Software Product that are incrementally developed and delivered. Every subsequent release of a module adds function to the previous release. This process is

**Figure 3.1:** Incremental Model

continued until the complete system is achieved

### Phases of Incremental Model

1. Requirement analysis: In the very first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.
2. Design Development: In this phase of the development, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.
3. Testing: In the incremental model, the testing phase checks the performance of each and every existing function as well as additional functionality. In the testing phase, the various methods are used to test the behaviour of each task.
4. Deployment: Once the product is tested, it is deployed in the production environment or first User Acceptance Testing (UAT) is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing.

5. Maintenance: After the deployment of a product on the production environment, maintenance of the product i.e., if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

### **3.1.2 Why to use Incremental Model**

We used Incremental model for our system design because of reasons mention below:

- Major requirements must be defined: however, some detail can evolve with time.
- There is a need to get a product to market early.
- The software will be produced quickly during the software life cycle.
- It is flexible and less luxurious to change requirements and scope.
- Though the development stages changes can be done.
- This model is less costly than others.
- A customer can answer back to each building.

### **3.1.3 Advantages**

1. Initial product delivery is faster.
2. Lower initial delivery cost.
3. Core product is developed leading i.e., main functionality is added in the first increment.
4. After each iteration, regression testing should be conducted. During this testing, faulty elements of the software can be quickly recognized because few changes are made within any single iteration.
5. It is generally easier to test and debug than other methods of software development because comparatively smaller changes are made during each iteration. This allows for more targeted and difficult testing of each element within the overall product.

6. With each release, a new article is added to the product.
7. Customers can reply to features and review the product.
8. Risk of changing requirements is reduced.
9. Workload is less.

#### **3.1.4 Disadvantages**

1. It requires good planning and designing.
2. Problems might be caused due to system architecture as such not all requirements are collected up front for the entire software lifecycle.
3. Each iteration phase is rigid and does not overlap each other.

Rectifying a problem in one unit requires correction in all the units and consumes a lot of time. It may arise affecting to system architecture because not all necessities are gathered up front for the entire software life cycle.

### **3.2 Module Details**

These are the following modules which will be executed in this system.

1. Users
2. System

#### **3.2.1 User**

- User can give the input for particular parameter.
  - User can do real time monitoring of waste
1. Registration and Authentication: Users need to register with the system, providing necessary details such as name, contact information, and address. Authentication mechanisms ensure secure access to the system, preventing unauthorized users from manipulating or accessing sensitive information.

2. User Dashboard: Upon login, users are presented with a personalized dashboard. This dashboard typically includes relevant information about their waste disposal activities, statistics, and any alerts or notifications. Users may also have access to historical data and trends related to their waste generation and disposal habits.
3. Waste Bin Monitoring: Users can view the status of waste bins in their vicinity. This may include real-time information on whether the bins are full, partially full, or empty. Smart sensors installed in waste bins transmit data to the system, allowing users to make informed decisions about when and where to dispose of their waste.
4. Waste Collection Scheduling: The system can provide information about scheduled waste collection days and times. Users may receive notifications reminding them to place their waste bins at designated collection points on specific days.
5. Reporting and Feedback: Users can report issues such as damaged bins, overflowing bins, or missed collections through the system. This feedback helps authorities address problems promptly. The system may also allow users to provide feedback on the overall waste management services, contributing to continuous improvement.
6. Educational Content and Awareness: The user module may include educational content on sustainable waste management practices, recycling guidelines, and the environmental impact of proper waste disposal. Awareness campaigns and tips on reducing waste generation can be integrated to promote a more eco-friendly approach.
- .

### **3.2.2 Admin Module**

- Dashboard and Overview: A centralized dashboard provides administrators with an overview of the entire waste management system, including the status of waste bins, collection schedules, user activities, and system health.
- User Management: Admins have the authority to manage user accounts, including creating new accounts, modifying user details, and handling account permissions. User authentication and authorization processes are typically managed through the admin module to ensure secure access.

- Waste Bin Monitoring and Control: Real-time monitoring of waste bin statuses, including fill levels, locations, and any sensor data. Admins can remotely control certain aspects of waste bins, such as locking or unlocking bins, triggering special collections, or adjusting sensor sensitivity.
- Alerts and Notifications: The admin module can generate and manage alerts for various events, such as bin overflows, system malfunctions, or missed collections. Notifications may be sent to users, collection teams, or maintenance personnel to address issues promptly.
- Maintenance and Repairs: Tracking and scheduling maintenance activities for waste bins, sensors, and other hardware components. Automated alerts for routine maintenance and notifications for urgent repairs.
- Software Updates and Maintenance: Facilities for updating the software, applying patches, and ensuring the overall health and stability of the system.

### 3.3 Project Estimation

In this section various calculation and estimations related to project has been calculated. The figure shows the system modules. The number of lines required for implementation of various modules can be estimated as follows

#### 3.3.1 Estimation in KLOC

Estimation is the process of finding an estimate, or approximation, which is a value that can be used for some purpose even if input data may be incomplete, uncertain, or unstable. Estimation determines how much money, effort, resources, and time it will take to build a Specific system or product. The number of lines required for implementation of various modules can be estimated as follows.

#### 3.3.2 Efforts

The Efforts required in person/month for implementation can be estimated as follows:

$$Effort = a * (LOC)^b$$

Where,  $a = 2.5$  and  $b = 1.1$ .

Hence, we get:

$$\begin{aligned} \text{Effort} &= 2.5 * (\text{KLOC})^{1.1} \\ \Rightarrow \text{Effort} &= 2.1 * (3.3)^{1.1} \\ \Rightarrow \text{Effort} &= 8 \text{ persons/month} \end{aligned}$$

### 3.3.3 Development time in months

We know that:

$$\begin{aligned} \text{Time required} &= \frac{\text{Efforts}}{\text{No. of Developers}} \\ \Rightarrow \text{Time required} &= \frac{8}{4} \\ \Rightarrow \text{Time required} &= 2 \text{ months} \end{aligned}$$

### 3.3.4 Total Time Required for Project Development

The total time required can be calculated as follows:

**Table 3.1:** Time Required for Project Development

| Task                            | Time Required |
|---------------------------------|---------------|
| Requirement Analysis and Design | 1 month       |
| Implementation and Testing      | 2 months      |
| <b>Total</b>                    | 3 months      |

Hence, total time required is nearly 3 months.

### **3.3.5 Number of Developers (N)**

The project is assigned to a group of 4 people (developers). Hence, the number of developers is taken 4.

The developers are as follows:

- D1: Vaishnavi Agre.
- D2: Akanksha Barve.
- D3: Sanket Gadekar.
- D4: Aniket Gangurde.

## Chapter 4

### SYSTEM ANALYSIS

#### 4.1 Project Scheduling and Tracking

Project Preparation and Tracing is important because in order to build a complex system, many software engineering tasks occur in parallel, and the result of work performed during one task may have a reflective effect on work to be conducted in another task. The inter dependencies are very difficult to understand without a detailed schedule.

##### 4.1.1 Project Work and Breakdown Structure (Analysis)

The project work is decomposed into the following work breakdown structure as a part of the analysis phase.

- **T1 : Communication** Software development process starts with the communication between customer and developer. According to need of project, we gathered the requirements related to project. Requirement gathering is an important aspect as the developer will come to know what customer expects from the project and also he can help a customer to know more features that can be added to project as he is a technical person. The most important thing needed is that communication should be smooth and clear that means developer should easily understand the demands of customer.
- **T2 : Planning** It includes complete estimation and scheduling (complete time line chart for project development). Before starting the project tasks should be scheduled that means there should be starting and ending date assigned for each and every task and developer should work harder to complete the required task within time chosen at the time of scheduling.

- **T3 : Modeling** It includes detailed requirements analysis and project design (algorithm, flowchart etc). Flowchart shows complete pictorial view of the project and algorithm is step by step solution of problem. Both flowchart and algorithm will be helpful in knowing the overall view of project and serve as a base for development of whole project.
- **T4 : Risk Management** It is a process of identifying, organizing, assessing and controlling threats to some organizations' capitals and earnings which assets overall or partial software product or performance. These threats, or risk, could stem from a wide variety of sources, including financial uncertainty, legal liabilities, strategies, management errors, accidents and natural disasters.

## 4.2 Project work breakdown structure (Implementation)

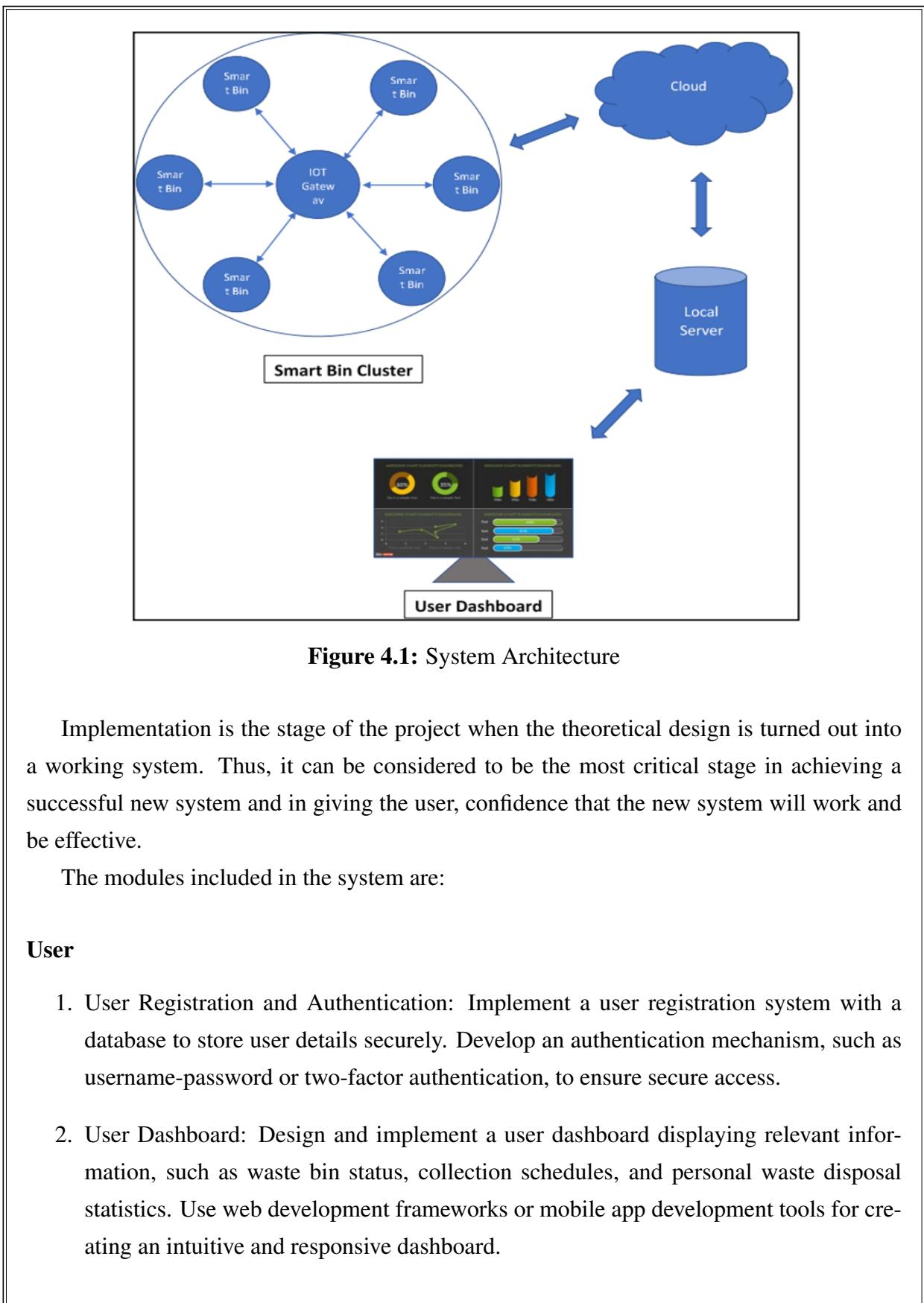
Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The proposed smart waste collection and management system for urban India consists of four main components: smart waste bins, waste collection vehicles, a centralized management dashboard, and a mobile application. The architecture is designed to facilitate seamless data collection, analysis, and communication between the components.

1. Smart Waste Bins: IoT-enabled waste bins equipped with sensors to monitor waste levels, bin temperature, and other environmental parameters. The bins transmit the collected data to the central management dashboard through wireless communication.

2. Waste Collection Vehicles: GPS-enabled waste collection vehicles that receive optimized route information from the central management dashboard. The vehicles are equipped with RFID readers to record waste collection events and update the central dashboard.

3. Centralized Management Dashboard: A cloud-based platform that collects, stores, and analyzes data from the smart waste bins and waste collection vehicles. The dashboard provides real-time insights into waste generation patterns, collection efficiency, and overall system performance.

4. Mobile Application: A user-friendly mobile app for citizens to access waste collection schedules, report issues related to waste bins or waste collection, and participate in waste reduction initiatives.



**Figure 4.1:** System Architecture

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The modules included in the system are:

### User

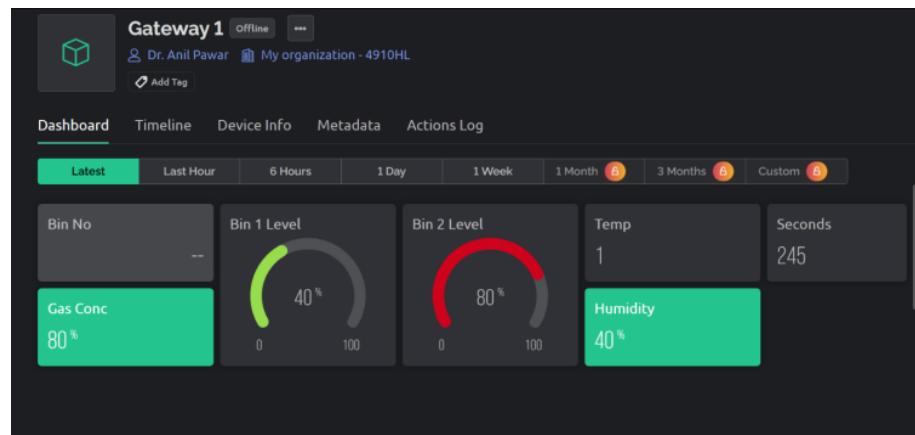
1. User Registration and Authentication: Implement a user registration system with a database to store user details securely. Develop an authentication mechanism, such as username-password or two-factor authentication, to ensure secure access.
2. User Dashboard: Design and implement a user dashboard displaying relevant information, such as waste bin status, collection schedules, and personal waste disposal statistics. Use web development frameworks or mobile app development tools for creating an intuitive and responsive dashboard.

3. Waste Bin Monitoring: Integrate sensors in waste bins to collect real-time data on fill levels. Implement a user interface to visualize the status of nearby waste bins, possibly using maps or a list view.
4. Waste Collection Scheduling: Develop a scheduling system that notifies users about upcoming waste collection days and times. Implement push notifications or emails to remind users to place their bins for collection.
5. Reporting and Feedback: Develop a reporting mechanism where users can provide feedback on the system, report issues, or request assistance. Implement a ticketing system for issue resolution and feedback management.
6. Educational Content and Awareness: Create a section for educational content within the user interface, offering tips on sustainable waste management. Implement features like quizzes, articles, or videos to enhance user awareness.
7. Dashboard and Overview: Create a centralized dashboard for administrators using web development frameworks. Implement data visualization tools for presenting key metrics and system status.
8. User Management: Develop an admin interface for managing user accounts, permissions, and roles. Implement functionality for creating, editing, and deactivating user accounts.

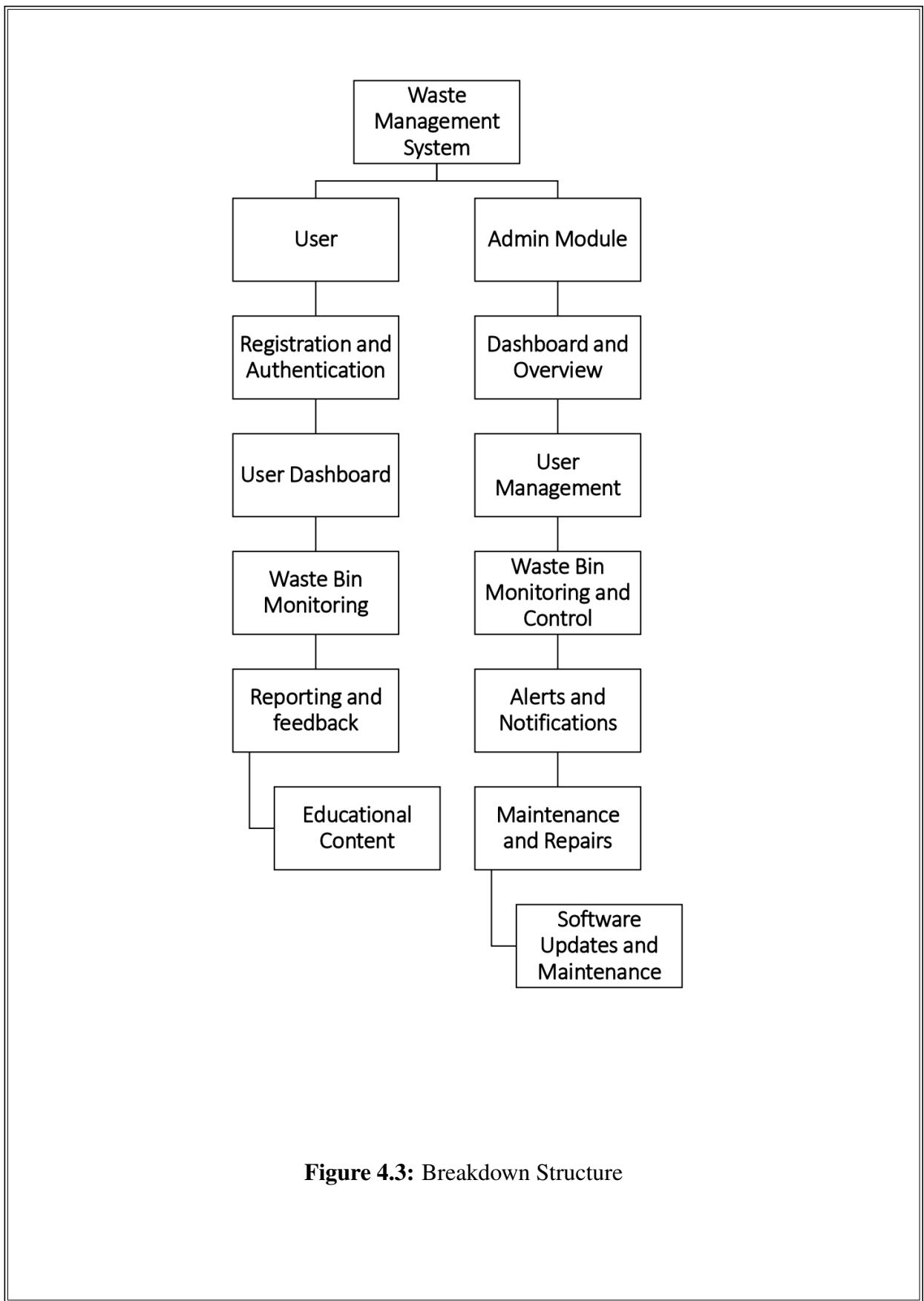
### **Admin**

1. Waste Bin Monitoring and Control: Integrate tools to monitor and control waste bins remotely. Implement a map-based interface or a dashboard displaying the real-time status of all waste bins.
2. Route Optimization and Planning: Develop algorithms for optimizing waste collection routes based on historical data and current conditions. Implement tools for manual route adjustments and planning.
3. Alerts and Notifications: Implement an alert system that notifies administrators of critical events. Develop a notification management system for configuring and customizing alert preferences.

4. Maintenance and Repairs: Integrate a maintenance module for scheduling and tracking routine maintenance tasks. Implement automated alerts for maintenance schedules and urgent repairs.
5. System Configuration and Settings: Create an admin interface for configuring system settings and parameters. Implement functionality for adjusting thresholds, collection schedules, and other parameters.
6. Security and Access Control: Implement secure authentication mechanisms for admin users. Develop role-based access control (RBAC) to restrict access based on user roles.
7. Integration with External Systems: Develop APIs or connectors to integrate with external systems such as GIS, CRM, or ERP. Ensure data consistency and integrity during integration.
8. Software Updates and Maintenance: Implement a system for deploying software updates, patches, and conducting maintenance tasks. Develop a rollback mechanism in case of issues during updates.



**Figure 4.2:** Gateway structure



**Figure 4.3:** Breakdown Structure

## 4.3 Technologies Used

### Internet of Things (IoT)

The Internet of Things (IoT) refers to a network of interconnected devices, vehicles, buildings, and other objects that collect, process, and exchange data through the internet. IoT technology enables real-time data collection, remote monitoring, and automation, transforming various industries, including transportation, agriculture, healthcare, and waste management.

IoT devices, such as sensors and actuators, are embedded in physical objects and can transmit data to a central system for analysis and decision-making. The use of IoT in waste management can lead to significant improvements in efficiency, sustainability, and cost effectiveness by providing accurate, real-time data on waste generation, collection, and disposal patterns.

The Internet of Things (IoT) has the potential to revolutionize various sectors in India, including agriculture, healthcare, and waste management. The Indian government has recognized the importance of IoT technology and has launched initiatives such as the "Digital India" and "Smart Cities Mission" to promote the adoption of IoT and other digital technologies.

IoT adoption in India faces challenges, such as inadequate internet infrastructure, low digital literacy, and concerns about data privacy and security. However, ongoing investments in telecommunications and digital infrastructure are expected to improve the IoT ecosystem, making it more accessible and affordable for various industries, including waste management.

#### 4.3.1 WordPress

WordPress is a popular open-source content management system (CMS) that allows users to create and manage websites and blogs with ease. Here are some key aspects of WordPress:

**Installation and Setup:** WordPress can be installed on a web server either manually or through one-click installations provided by many hosting providers. Once installed, users can customize their site through a web-based interface.

**Themes:** WordPress uses themes to control the visual appearance of a website. Users can choose from thousands of free and premium themes or create custom themes to suit their specific needs.

**Plugins:** Plugins extend the functionality of WordPress. There are thousands of plugins available for various purposes, such as SEO optimization, e-commerce, security, social media

integration, and more.

**Content Creation:** WordPress comes with a user-friendly editor that allows users to create and format content without any knowledge of HTML or CSS. Users can easily add text, images, videos, and other multimedia elements.

**Pages and Posts:** Content in WordPress is organized into two main types: pages and posts. Pages are typically used for static content like the homepage, while posts are used for dynamic content, such as blog entries.

**Media Management:** WordPress has a built-in media library where users can upload and manage images, audio files, and videos. The media library simplifies the process of adding multimedia content to pages and posts.

**User Management:** WordPress allows multiple users with different roles and permissions. Administrators can control what each user role can and cannot do, ensuring a secure and collaborative environment.

**Search Engine Optimization (SEO):** WordPress is inherently SEO-friendly, and there are plugins available, such as Yoast SEO and All in One SEO Pack, that help users optimize their content for search engines.

**Customization:** Users can customize their WordPress site using widgets, menus, and the Customizer tool. Additionally, developers can create custom themes and plugins to add advanced features.

**Updates and Security:** WordPress regularly releases updates to enhance features, improve security, and fix bugs. Users are encouraged to keep their installations up to date to benefit from the latest improvements.

**Community and Support:** WordPress has a vast and active community of developers, designers, and users. There are forums, documentation, and tutorials available to help users troubleshoot issues and learn more about WordPress.

**E-commerce:** With plugins like WooCommerce, WordPress can be transformed into a powerful e-commerce platform, allowing users to set up online stores and sell products. WordPress is a versatile platform that caters to a wide range of users, from beginners creating simple blogs to developers building complex websites and applications. Its user-friendly interface and extensive ecosystem of themes and plugins make it one of the most widely used CMSs on the internet.

### 4.3.2 RASA chatbot

RASA is an open-source platform for building conversational AI applications, including chatbots and virtual assistants. It provides a set of tools and libraries for developing natural language understanding (NLU) and dialogue management capabilities. RASA is designed to be flexible, allowing developers to create custom, context-aware chatbots that can understand and respond to user inputs in a more sophisticated manner. Here are some key components and features of RASA:

**RASA NLU (Natural Language Understanding):** This component focuses on understanding user inputs by extracting intent and entities from natural language text. Developers can define training data that includes examples of user messages, intents (the user's intention), and entities (specific pieces of information within the user's message).

**RASA Core:** RASA Core is responsible for managing the dialogue and deciding how the chatbot should respond to user inputs. It uses machine learning to learn from conversations and predict the next best action based on the current context.

**Domain Configuration:** The domain configuration file (domain.yml) defines the actions the bot can take, the intents it can understand, and the entities it can extract.

**Custom Actions:** Developers can define custom actions that the bot can take in response to user inputs. These actions can include making API calls, querying databases, or performing other tasks.

**Training Data:** RASA chatbots are trained on user conversations and labeled data. Developers provide examples of conversations and correct responses, which are used to train the models for NLU and dialogue management.

**Flexible Deployment:** RASA chatbots can be deployed on various platforms, including local servers, cloud platforms, and containers (e.g., Docker). This flexibility allows developers to choose the deployment option that best fits their needs.

**Community and Ecosystem:** RASA has an active community of developers and a growing ecosystem. This community support includes forums, documentation, and contributions from developers worldwide.

**Open Source:** RASA is open-source software, meaning that its source code is freely available for developers to view, modify, and contribute to. This openness encourages collaboration and innovation.

**Integration:** RASA can be integrated into various messaging platforms, such as Slack, Facebook Messenger, or custom chat interfaces, enabling developers to deploy their chatbots

where their users are. Overall, RASA provides a comprehensive framework for building intelligent and context-aware chatbots, making it a popular choice for developers working on conversational AI applications.

#### **4.3.3 Implementation details about RASA:**

**Installation:** You can install RASA by using pip. The recommended way is to create a virtual environment and install RASA within that environment.

- pip install rasa

**Project Initialization:** Create a new RASA project using the following command:

- rasa init

This will create a basic project structure with necessary files, including domain.yml, config.yml, and data files.

**Training Data:** RASA uses training data to understand user inputs and learn how to respond. The training data includes conversations, intents, entities, and responses. Conversations are typically stored in files like stories.md or data/nlu.md. Intent examples, entity examples, and responses are defined in these files.

**Domain Configuration:** The domain.yml file defines the domain of your assistant, including intents, entities, actions, and responses. It's crucial for training the model to understand the context and generate appropriate responses.

**Configuration File:** The config.yml file contains configuration details for training the RASA models. It includes information about the pipeline, policies, and other training-related parameters.

**Training Models:** Train the RASA NLU model using the following command:

- rasa train nlu

Train the RASA Core model using:

- rasa train

**Running the Bot:** Once the models are trained, you can run the RASA server using:

- rasa run

This will start a server, and you can interact with your bot through a REST API.

**Interactive Learning:** RASA provides a feature called interactive learning, where you can have a conversation with your bot and correct its responses in real-time. This helps improve the model iteratively.

**Integration:** RASA can be integrated into various channels such as Slack, Facebook Messenger, or custom chat interfaces. You can find integration details in the RASA documentation.

**Deployment:** Deploy your RASA bot using various deployment options, including Docker, cloud platforms like AWS or Azure, or any server that supports Python applications.

**Continuous Training:** As your bot interacts with users and receives new data, it's essential to retrain your models periodically to improve performance

#### 4.3.4 Blynk App

Blynk is a popular Internet of Things (IoT) platform that enables users to create custom mobile applications to control and monitor connected hardware projects. It provides a drag-and-drop interface for building mobile apps without the need for extensive programming knowledge. Here are some key aspects of the Blynk app:

**IoT Platform:** Blynk serves as a platform for building IoT applications. It allows users to connect hardware devices to the internet and control them remotely through a mobile app.

**Mobile App Builder:** Blynk offers a mobile app builder that simplifies the process of creating custom interfaces for IoT projects. Users can design the app's layout, add buttons, sliders, graphs, and other widgets using a visual interface.

**Widgets:** Blynk provides a variety of widgets that users can add to their mobile apps. These widgets include buttons, sliders, displays, gauges, and more, allowing for the creation of interactive and dynamic interfaces.

**Supported Platforms:** Blynk supports a wide range of hardware platforms, including Arduino, Raspberry Pi, ESP8266, ESP32, and others. This flexibility allows users to connect various sensors, actuators, and devices to their IoT projects.

**Communication Protocols:** Blynk supports multiple communication protocols such as Wi-Fi, Bluetooth, Ethernet, and USB. This enables users to connect their hardware devices to the internet and control them wirelessly.

**Cloud Connectivity:** Blynk uses cloud services to facilitate communication between the mobile app and connected hardware. Users can remotely monitor and control their devices from anywhere with an internet connection.

**Open-Source Library:** Blynk provides open-source libraries for popular hardware platforms and programming languages. These libraries simplify the process of integrating Blynk into your hardware projects.

**Energy System:** Blynk uses an "energy" system to manage resources within the app. Each widget and feature consumes a certain amount of energy, and users are allocated a specific amount of energy for their projects. Additional energy can be purchased if needed.

**Security:** Blynk takes security seriously and provides features such as secure connections (SSL/TLS) and authentication tokens to protect communication between the app and connected devices.

**Community and Documentation:** Blynk has an active community where users can share their projects, seek help, and collaborate. The platform also provides comprehensive documentation and tutorials to help users get started with building their IoT projects. Whether you're a hobbyist or a professional developer working on IoT projects, Blynk offers a user-friendly and versatile platform for building custom mobile apps to control and monitor connected hardware.

#### 4.4 Task Identification

Following analysis and design tasks are to be carried out in the process of analysis and design of the project. All project modules are divided into following tasks.

- T1: Project Definition Searching
- T2: Project Definition Preparation
- T3: Literature Collection
- T4: Project Definition Finalization
- T5: Dataset Collection
- T6: Synopsis Preparation
- T7: Requirement Analysis and Validation
- T8: Determine Process Model (Incremental Model)
- T9: System Breakdown Structure
- T10: Project Estimation in KLOC
- T11: Project Scheduling and Tracking

- T12: Analysis Modelling using Behavioral, Functional and Architectural Modelling
- T13: Feasibility Management of Project using Mathematical Modelling
- T14: Risk Analysis, Project Management and Risk Management
- T15: Report Preparation
- T16: Data Preprocessing using Preprocessing Techniques
- T17: Extracting the Features from the Preprocessed Dataset
- T18: Creating Graphical User Interface
- T19: Model Training using Linear Regression, Ridge Regression, Lasso Regression, SVC, Decision Tree, Random Forest Algorithms
- T20: Real time Monitoring Of Waste
- T21: Testing of the Generated Model
- T22: Deployment of Project

## 4.5 Analysis Modeling

The system analysis model is made up of class diagram, sequence or collaboration diagrams and state chart diagrams. Between them they constitute a logical, implementation free view of computer system that includes a detail definition of every aspect of functionality. Analysis model contains following modeling:

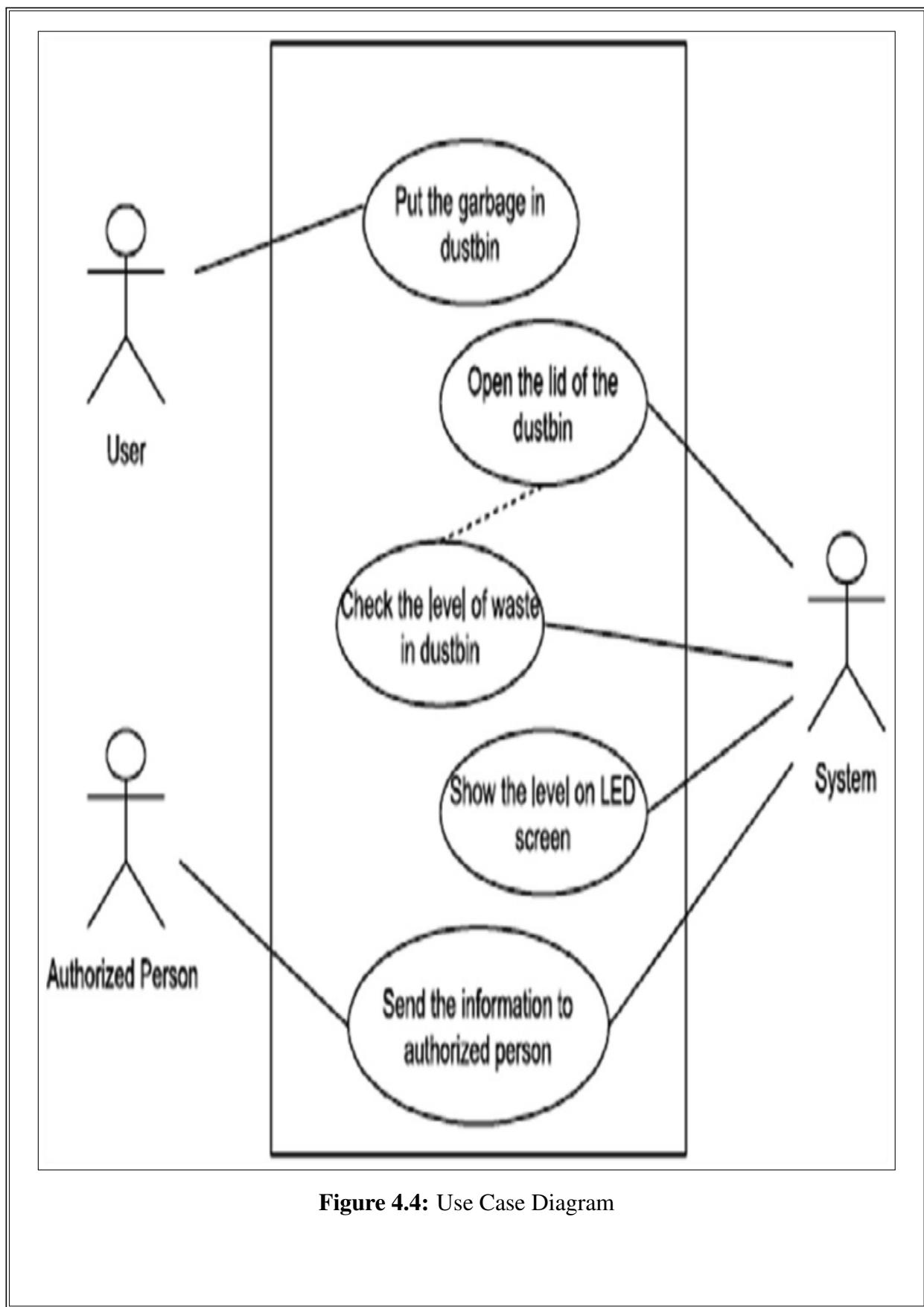
1. Behavioral modeling.
2. Functional modeling.
3. Architectural modeling.

Analysis modeling uses a combination of text and diagrammatic form to depict requirement for data, function and behaviour in a way that is relatively easy to understand and more important, straightforward to review for correctness, completeness and consistency.

#### **4.5.1 Behavioral modeling**

##### **Use case diagram**

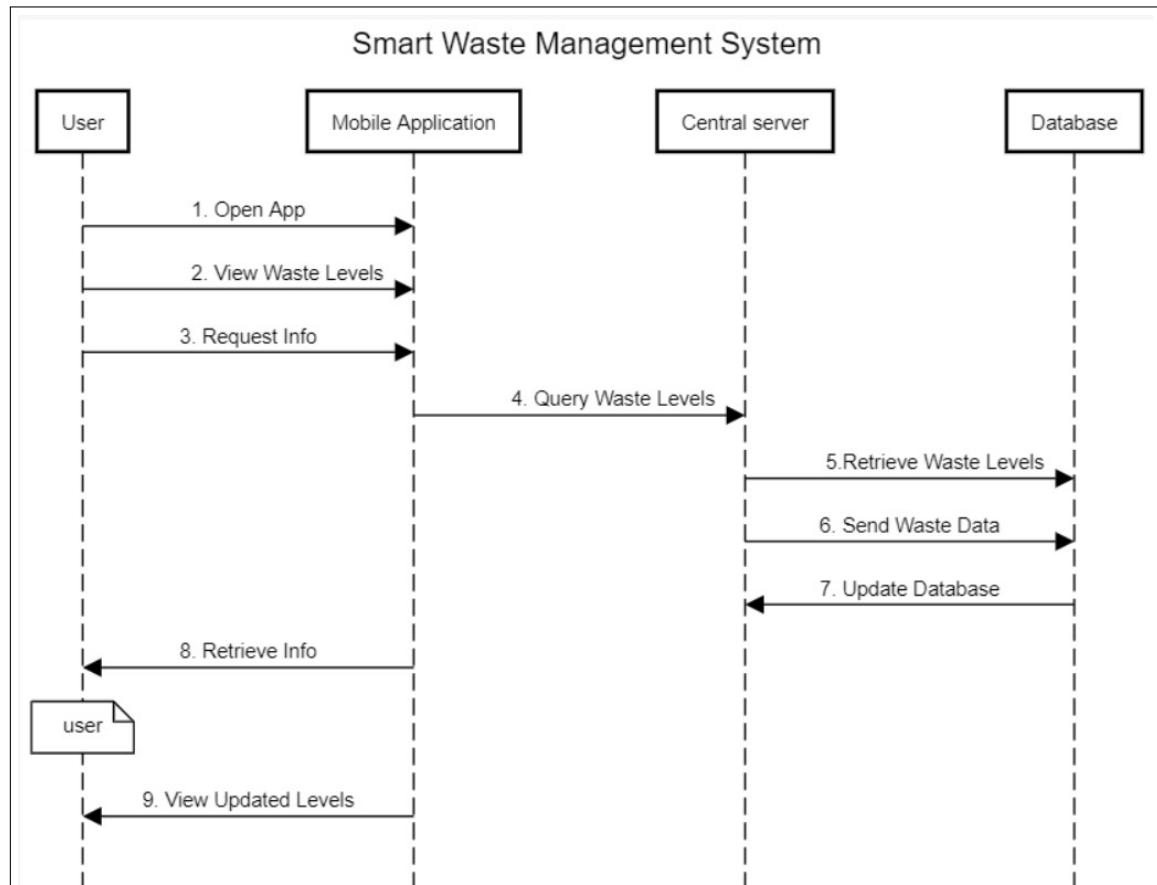
A use case involves a sequence of interactions between the motivator and the system, possibly including other actors.



**Figure 4.4:** Use Case Diagram

## Sequence Diagram

A sequence diagram is a graphical view of a state that shows object interface in a time-based sequence.



**Figure 4.5:** Sequence Diagram

## Class Diagram

A Class is an Explanation of a set of objects that have the same attributes and methods Attribute.

- Attribute
- Operation
- Relationship

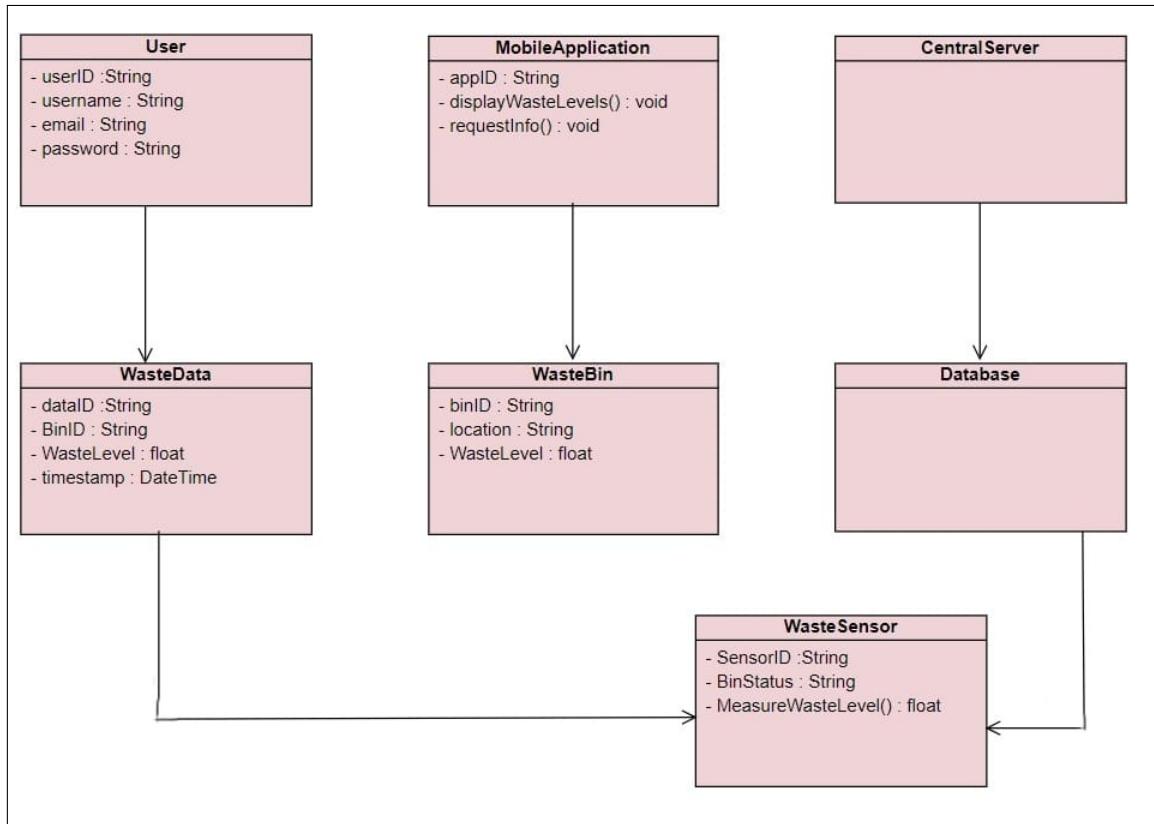
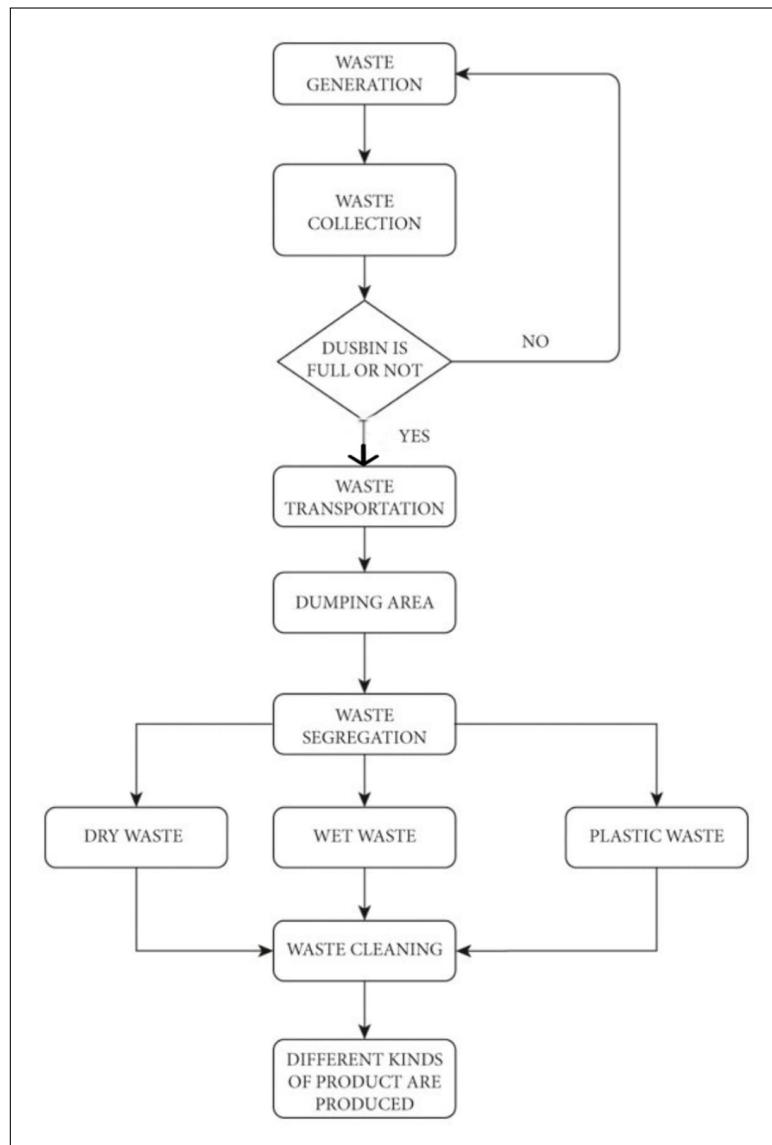


Figure 4.6: Class Diagram

## Activity Diagram

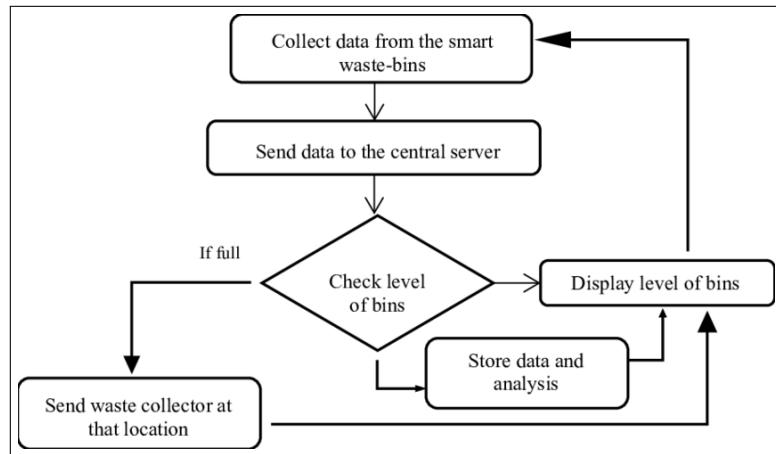
Activity diagram is an important diagram to describe the dynamic aspects of the system. Activity diagram is essentially a flowchart to represent the flow from one activity to another activity.



**Figure 4.7:** Activity Diagram

### State Chart Diagram

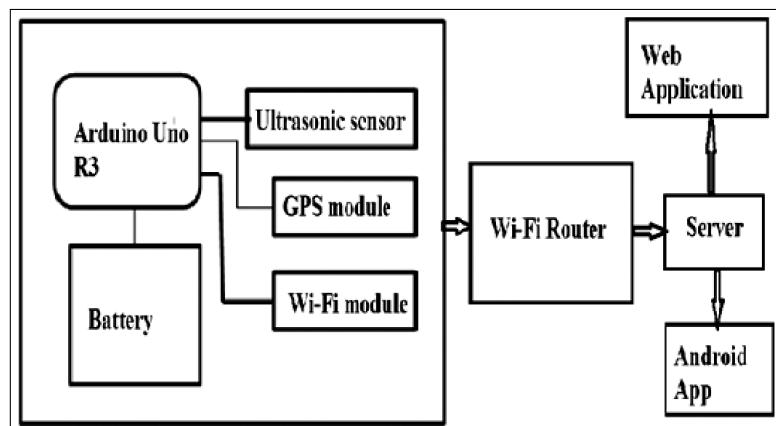
State chart Diagram shows the state machine that consists of states, transitions, events and activities. This diagram shows how the system makes the transition from one state to another state on occurrence of a particular event.



**Figure 4.8:** State Chart Diagram

### Deployment Diagram

Deployment figures are used to visualize the topology of the physical components of a system where the software components are organized. Deployment diagrams are used for describing the hardware components where software components are deployed.



**Figure 4.9:** Deployment Diagram

## Chapter 5

### RISK MANAGEMENT

Project risk management is the process of identifying, analyzing and then responding to any risk that arises over the life cycle of a project to help the project remain on track and meet its goal. Managing risk isn't reactive only, it should be part of the planning process to figure out risk that might happen in the project and how to control that risk if it in fact occurs. A risk is anything that could potentially impact your projects timeline, performance or budget. Risks are potentialities, and in a project management context, if they become realities, they then become classified as issues that must be addressed. So risk management, then, is the process of identifying, categorizing, prioritizing and planning for risks before they become issues .This article gives us ten golden rules to apply risk management successfully in our project.

#### 5.1 Risk Identification

- **Product size risk**

R1: Is the development team able to decompose the required program into smaller, highly cohesive modules.

R2: Are there enough development team members available relative to the size of the project.

- **Business impact**

R3: Delay in project delivery (violation in time constraints) can hamper the customer economically.

R4: If system is not more efficient than the existing system, it will cause economic losses.

- **Customer related risk**

R5: User or service provider is a non-technical person; if proper guide-lines were not mentioned then it will create ambiguity.

R6: If a user wants any modifications that lead to changing the entire System.

- **Process risk** R7: The risk of technology errors or security incidents that disrupt or invalid processes.

R8: Quality of a process itself that leads to failures. A low quality process may not properly work and may break down the system.

- **Technical Risk**

R9: Lack of database stability and concurrency.

R10: Module integration fails.

R11: Wrong trained data may lead to wrong results.

- **Development Environment Related risk**

R12: Lack of proper training and less knowledge of programming leads to a moderate risk. It will delay product development and deployment.

## 5.2 Strategies Used to Manage Risk

- S1: Formulation and follow up of the project plan on a regular basis.
- S2: Keep assigned work under certain deadlines.
- S3: Web Development cycle should be used.
- S4: Regular meetings with users reduce the risk to some extent, design systems with flexibility and maintain necessary documentation for the same.
- S5: Re-defined software process at higher degree.
- S6: Proper training on required technical tools for development of projects reduces risk.
- S7: Make certain rules that each one the members are taking part in the design.

- S8: Study and understanding of project definition, programming language.
- S9: Take a look at model development and all its associated used software.
- S10: Time constraints must be followed to avoid economical risks.
- S11: Each and every module must be tested for its functioning.
- S12: After unit testing, the system must be integrated and validated accordingly.
- S13: Integration testing for authentication hierarchy.
- S14: Use of standard database technology which supports concurrency more.

## 5.3 Risk Projection

### 5.3.1 Preparing Risk Table

The risk table shown below lists all possible risks which may occur at any stage during development of a project. Table also clearly shows the impact of the risks and RMMM (Risk Mitigation Monitoring and Management) plan to deal with any such risks.

**Table 5.1:** Risk Management Table

| Risk | Category                             | Probability | Impact | Plan        |
|------|--------------------------------------|-------------|--------|-------------|
| R1   | Product Size Risk                    | More        | High   | S1,S3,S4    |
| R2   | Product Size Risk                    | Less        | Less   | S4,S6,S1    |
| R3   | Business Impact Risk                 | More        | High   | S2          |
| R4   | Business Impact Risk                 | More        | High   | S4          |
| R5   | Customer Related Risk                | More        | High   | S11,S12,S13 |
| R6   | Customer Related Risk                | More        | Less   | S11,S12,S13 |
| R7   | Process Risk                         | More        | High   | S14         |
| R8   | Process Risk                         | More        | High   | S14         |
| R9   | Technical Risk                       | Less        | High   | S1,S6       |
| R10  | Technical Risk                       | More        | High   | S7          |
| R11  | Development Environment Related Risk | Less        | Less   | S4,S5,S7,S8 |
| R12  | Product Size Risk                    | More        | High   | S9,S10      |

## 5.4 Feasibility

Feasibility is defined as an evaluation or analysis of the potential impact of a proposed project.

- Technical feasibility: - It is disturbed with specifying equipment and software that will successfully preserve the task required.
- SAT (Satisfiability): - Boolean formula is satisfiability if there exists at least one way of assigning value to its variable so as to make it true and we denote it by using SAT. The problem of deciding whether a given formula is satisfiability or not.
- Facility to produce output in given times.
- Response time under certain conditions.
- Operational Feasibility: -It is related to human organization.
- What changes will be brought in with the system.
- How organizational Structure will be distributed.
- What new skills are required.
- Economic Feasibility: -It is the most frequently used technique for evaluating the effectiveness of proposed systems. Most usually identified as cost/benefit analysis.

## **Chapter 6**

### **TECHNICAL SPECIFICATION**

#### **6.1 Software requirement specification**

##### **6.1.1 Operating System (OS)**

- Windows 10 or higher.

##### **6.1.2 Integrated Development Environment (IDE)**

- WordPress Tool.

##### **6.1.3 Blynk App**

#### **6.2 Hardware Requirement Specifications**

The algorithms used in the project are computationally intensive and thus require high computing capabilities in terms of hardware. For the testing and demo purpose we will need good hardware usually found in desktop type of computers. Thus, to reduce the computation time of the project we recommend to use high performance system.

- Ultrasonic Sensor - Non-Invasive Sonic Eye™ SL-630
- Ram: 2GB
- Processor: Intel core i5 9th Gen
- 64-bit CPU
- IR Proximity Sensor
- Temperature and Humidity Sensor - DHT 11

## Chapter 7

### IMPLEMENTATION DETAILS

In this chapter implementation details are explained as follow:

#### 7.1 Smart Bin

To implement the waste bin monitoring system, IoT-enabled waste bins with ultrasonic sensors are deployed to measure waste levels. These bins also contain temperature and humidity sensors to monitor environmental conditions. The data from the sensors is transmitted to the central waste management dashboard using wireless communication modules, such as Wi-Fi, LoRaWAN, or cellular networks. The waste bin monitoring system enables real-time tracking of waste levels and identification of bins that require immediate collection, ensuring efficient waste collection and management.

**Sensor Technology:** Smart bins are equipped with sensors that can detect the fill level of the bin in real-time. This data is crucial for optimizing waste collection routes and schedules, reducing unnecessary pickups, and improving overall operational efficiency.

**IoT Connectivity:** Internet of Things (IoT) technology enables smart bins to communicate wirelessly. This connectivity allows for remote monitoring and management, facilitating quick response to issues such as overflow or malfunction.

**Data Analytics:** The collected data from smart bins can be analyzed to derive insights into waste generation patterns. This information can be valuable for city planners and waste management authorities to make informed decisions about resource allocation and infrastructure development.

**Energy Efficiency:** Some smart bins are designed with energy-efficient features, such as solar panels for powering sensors and communication devices. This enhances sustainability by reducing reliance on traditional power sources.



**Figure 7.1:** Smart Bin

User Interaction: Many smart bins come with user interfaces or mobile apps that allow citizens to interact with the waste management system. This can include features like reporting issues, receiving alerts, or accessing information about recycling practices.

## 7.2 RASA Chatbot

```

C:\Windows\System32\cmd.exe x + ~
2023-12-05 22:44:10 WARNING rasa.shared.utils.common - The Unexpected Intent Policy is currently experimental and might change or be removed in the future.
  Please share your feedback on it in the forum (https://forum.rasa.com) to help us make this feature ready for production.
2023-12-05 22:44:17 INFO  root - Rasa server is up and running.
Bot loaded! Type a message and press enter (use '/stop' to exit).
Your input -> hey
Hello! How can I assist you today?
Your input -> I need to report a problem
Thank you for reporting the issue. Please describe the problem in detail so that we can assist you better.
Your input -> Give me sustainability tips
Certainly! For sustainability tips, here are a few suggestions:
1. **Workplace:** Implement energy-efficient appliances and reduce water consumption.
2. **Office:** Encourage paperless practices and encourage recycling among colleagues.
3. **Outdoor Activities:** Choose eco-friendly products and clean up after yourself to preserve nature.
Your input -> Can you collect my trash?
Absolutely! To schedule a waste pickup, please provide your complete address along with the preferred date and time for the pickup. We'll take care of the rest!
Your input -> How much waste have I produced?
To view your usage statistics, simply log in to your account on our website or app. If you encounter any issues, feel free to let us know, and we'll assist you promptly.
Your input -> I need to contact support
For support, you can reach our team by sending an email to support@smartwastemanagement.com or by calling our helpline at +1-800-123-4567. We're here to help!
Your input -> Tell me about your impact on the environment
Absolutely! Our sustainability efforts have made a positive impact on the environment by reducing waste and promoting eco-friendly practices. If you'd like more details, let us know!
Your input -> Where are the nearest bins?
To find the nearest waste collection bins, use our mobile app or website. Alternatively, you can search for bin locations by entering your address. It's quick and convenient!
Your input -> Recycling instructions
For comprehensive recycling guidance, visit our website's recycling guide. It outlines which items are recyclable and provides tips on proper recycling practices.
Your input -> Give feedback
We highly value your feedback! Share your thoughts and suggestions by sending an email to feedback@smartwastemanagement.com. Your input helps us improve our services.
Your input -> How to order bins?
To order waste bins, simply log in to your account or contact our customer support. We offer a variety of bin sizes for you to choose from.
Your input -> I have payment problems
If you're facing payment issues, reach out to our billing department at billing@smartwastemanagement.com. They will assist you in resolving the problem.
Your input -> Change my pickup schedule
To change your waste pickup schedule, log in to your account or contact our support team. Provide your details, and we'll assist you with the necessary changes.
Your input -> How can I donate?
We offer donation options for items that can be reused or repurposed. Let us know what you'd like to donate, and we'll guide you through the donation process.
Your input -> Mobile app assistance
For mobile app support, visit our app's support section or reach out to our mobile app support team at mobilesupport@smartwastemanagement.com. They'll assist you promptly.
Your input -> I forgot my password
If you've lost your password, reset it by clicking the 'Forgot Password' option on our website or app. Follow the instructions, and you'll regain access to your account.
Your input -> Teach me about sustainability
Explore educational resources on sustainability on our website. Gain insights and knowledge to contribute to a more sustainable future.
Your input -> Urgent support
For emergency situations, please call 911 or your local emergency services number. If you need assistance related to our services during emergencies, contact our support team.
Your input -> Tell me about your products
To get detailed information about our products, visit the 'Products' section on our website, or reach out to our sales team at sales@smartwastemanagement.com.
Your input -> thanks
You're welcome! If you need any more assistance, don't hesitate to ask.
Your input -> bye
Goodbye! If you have more questions in the future, feel free to ask.
Your input -> |
```

**Figure 7.2: RASA Chatbot Implementation**

Rasa is fully open source, providing developers with the flexibility to customize and extend the functionality of their chatbots according to specific needs. This allows for greater control and adaptability in comparison to closed-source alternatives.

**Natural Language Processing (NLP):** Rasa incorporates natural language understanding (NLU) capabilities, enabling chatbots to comprehend and respond to user inputs in a more contextually aware and natural manner. Rasa's NLU engine can be trained with domain-specific language data.

**Intent Recognition:** Rasa allows developers to define and train the chatbot to recognize user intents. Intents represent the user's goal or purpose behind a particular message, and accurate intent recognition is crucial for generating relevant responses.

**Dialogue Management:** Rasa provides tools for managing the flow of conversations and maintaining context. Developers can design interactive and dynamic conversations by defining dialogue trees, handling user responses, and managing the overall chatbot behavior.

**Entity Recognition:** In addition to intents, Rasa supports entity recognition. Entities are specific pieces of information within a user's message, such as dates, locations, or names. Extracting entities is important for understanding the details of user requests.

**Slot Filling:** Rasa allows developers to define slots, which are pieces of information the chatbot needs to collect from the user to fulfill a request. The chatbot can prompt users for missing information and fill these slots during the conversation.

**Integration:** Rasa chatbots can be integrated with various messaging platforms, websites, and applications. This flexibility allows developers to deploy their chatbots on platforms such as Facebook Messenger, Slack, or a custom web interface.

**Machine Learning:** Rasa leverages machine learning techniques for intent recognition and entity extraction. Developers can continually improve and train the model based on real user interactions, enhancing the chatbot's performance over time.

**Community and Documentation:** Rasa has a strong community of developers and a comprehensive set of documentation. This community support and documentation make it easier for developers to get started, troubleshoot issues, and collaborate with others using Rasa.

**Scalability:** Rasa can be deployed on various infrastructure, making it scalable to handle different levels of user interactions. Whether it's a small-scale prototype or a large-scale production deployment, Rasa can be adapted accordingly.

### 7.3 Awareness Website

1. **Interactive Dashboard:** Our homepage features an intuitive and interactive dashboard that provides real-time insights into waste management statistics. From the total amount of waste collected to the percentage recycled, you can witness the impact of our system on the environment at a glance.

2. **Live Sensor Data:** Explore the cutting-edge technology behind our system with live sensor data. See how our smart sensors are deployed across cities to monitor waste levels in bins, optimizing collection routes and reducing carbon footprints. The dynamic visuals make



**Figure 7.3:** Home Page

understanding our system's efficiency a seamless experience.

3. Community Initiatives: Discover how we're building a sense of community around waste reduction. Our homepage showcases ongoing initiatives and success stories from individuals and organizations that have embraced a sustainable lifestyle. From recycling challenges to community clean-up events, get inspired to join the movement.

4. Educational Resources: Knowledge is power, and our website is committed to empowering users with valuable insights into waste management and environmental conservation. Engage with informative articles, infographics, and video content that break down complex concepts into easily digestible pieces, fostering a sense of responsibility and informed decision-making.

5. User-Friendly Navigation: Our homepage is designed with user experience in mind. Whether you're a first-time visitor or a regular user, our streamlined navigation ensures that you can effortlessly explore different sections of the website, from the latest news on sustainable practices to FAQs about our smart waste management system.

6. Social Media Integration: Connect with us beyond the website through our social media links. Stay updated on the latest developments, share your thoughts, and participate in discussions that amplify the impact of our collective efforts towards a cleaner planet.

7. Join the Movement: Empower yourself by becoming a part of the solution. The homepage prominently features a call-to-action, inviting visitors to join the Smart Waste Management and Awareness System movement. Whether through volunteering, spreading awareness, or adopting eco-friendly practices, every action counts.

The screenshot shows a service page with a light gray background. At the top center, the word "SERVICES" is written in bold capital letters above a horizontal red line. Below this, there are two main sections: "Chat Bot" on the left and "System Dashboard" on the right. The "Chat Bot" section features an illustration of a laptop screen displaying a friendly AI bot with a speech bubble saying "HI!". Below the illustration, the text reads: "Let's have interaction to understand the need of Smart Waste Management System" and includes a "TRY NOW →" button. The "System Dashboard" section features an illustration of a smartphone and a tablet displaying various charts and graphs, with a small gear icon nearby. Below the illustration, the text reads: "Our dashboard gives you insights into the most important metrics." and includes a "DASHBOARD →" button. In the bottom right corner of the page, there is a red circular icon with a white speech bubble containing a red dot.

**Figure 7.4:** Service Page

1. Smart Sensor Deployment: Discover the magic of smart sensors! Our cutting-edge technology involves strategically deploying sensors across cities to monitor real-time waste levels in bins. This data-driven approach optimizes collection routes, minimizes operational costs, and significantly reduces the ecological footprint of waste management.
2. Data Analytics and Insights: Dive into the world of data analytics as we turn raw information into actionable insights. Our system generates detailed reports, charts, and graphs that not only reflect the efficiency of waste management but also highlight areas for improvement. Harness the power of data to make informed decisions and drive positive environmental change.
3. Customized Solutions for Municipalities: Tailored to meet the unique needs of municipalities, our services include the design and implementation of customized waste management solutions. From optimizing collection schedules to enhancing recycling programs, we collaborate closely with local governments to create sustainable, efficient, and cost-effective systems.
4. Community Engagement Programs: Explore our community-focused initiatives designed to instill a sense of environmental responsibility. We offer comprehensive programs that educate communities on sustainable living practices, conduct workshops, and organize events to promote active participation in waste reduction and recycling efforts.
5. Educational Workshops and Training: Knowledge is key to driving change. Our service page details educational workshops and training sessions aimed at individuals, businesses, and organizations. Equip yourself with the tools and know-how to make eco-conscious

decisions in your daily life and contribute to a more sustainable future.

6. Waste Audit Services: For businesses and institutions aiming to enhance their waste management practices, our waste audit services provide a detailed analysis of current waste generation and disposal processes. Armed with this information, we collaborate to develop targeted strategies to minimize waste and maximize recycling.

7. 24/7 Customer Support: Our commitment to service excellence extends to our customer support. Explore the various channels through which we provide assistance, answer queries, and address concerns. Whether you're a municipality, business, or an individual, our dedicated support team is here to ensure a seamless experience with our smart waste management system.

8. Sustainability Consultation: Embark on your sustainability journey with our consultation services. Our experts provide guidance on eco-friendly practices, waste reduction strategies, and the adoption of sustainable technologies. Together, we can create a roadmap for a greener future.

#### TODAY'S NEED

The population of the cities worldwide is growing at an accelerating rate, imposing significant socioeconomic and environmental issues. One of the most critical issues in cities nowadays is the production of millions of tons of waste every day, and the cities do not have appropriate methods to manage waste properly. As per The World Bank, by 2050, global waste production will grow by approximately 3.40 billion tons, pushing the world to the edge of an alarming situation.

[NEWS ARTICLE →](#)



### OUR MISSION

- Optimizing Waste Collection and Management
- Creating Awareness About Environmental Impact
- Empowering Communities for Cleaner Cities
- Fostering Innovation in Green Technology
- Encouraging Public Participation in Environmental Conservation



**Extraordinary Experiences**

**Our Core Values**

Since the deployment of these smart waste management system across our campus, we have witnessed a noticeable improvement in waste management practices. The reduced overflow of bins has led to cleaner surroundings. The data analytics provided by the smart waste management system have been particularly valuable in identifying peak usage periods and optimizing resource allocation accordingly.

MESSAGE

### THE BENEFITS OF SMART WASTE MANAGEMENT

**Prevention of Overflowing Bins**

Sensors on waste containers can detect fill levels and notify waste management teams when bins are reaching capacity. This helps prevent overflowing bins, reducing litter and the risk of environmental contamination.

EXPLORE



**Figure 7.5:** Awareness Page

Awareness Page is a space dedicated to fostering environmental consciousness and inspiring positive change. Here, we illuminate the importance of awareness in shaping a sustainable future and provide resources to empower individuals, communities, and organizations to become stewards of the environment.

Delve into a wealth of knowledge through our Environmental Education Hub. Access engaging articles, informative videos, and interactive infographics that cover a spectrum of topics, from the basics of waste management to the global impact of sustainable practices. Empower yourself with the understanding needed to make informed, eco-conscious decisions.

Explore inspiring narratives of individuals and communities making a difference. Our Awareness Page showcases success stories from around the world, highlighting innovative

Department of Computer Engineering, Sanjivani College of Engineering, Kopargaon 423603

65

initiatives, grassroots movements, and the transformative power of collective action. Witness the impact of small changes and be inspired to embark on your own journey towards sustainability.

**CONTACT US**

**WE'RE READY, LET'S TALK.**

**SEND MESSAGE**

**CONTACT INFO**

**Address**  
Sanjivani Group of Institutes At Post, Sahajanand Nagar, Singnapur, Kopargaon, Maharashtra 423603

**Email Us**  
[smartcampus.sanjivani@gmail.com](mailto:smartcampus.sanjivani@gmail.com)

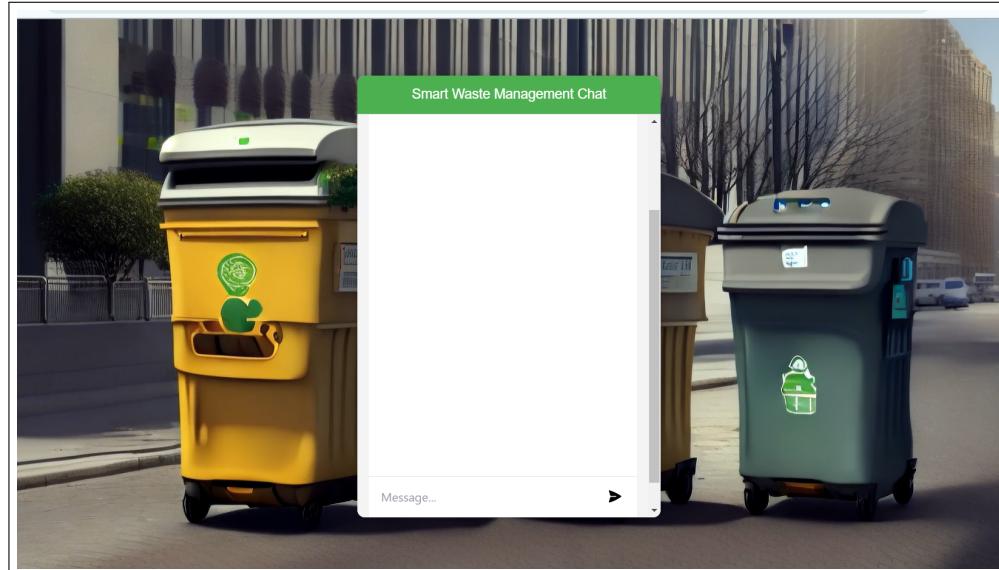
**Call Us**  
+91 7219747737

**Follow Us**

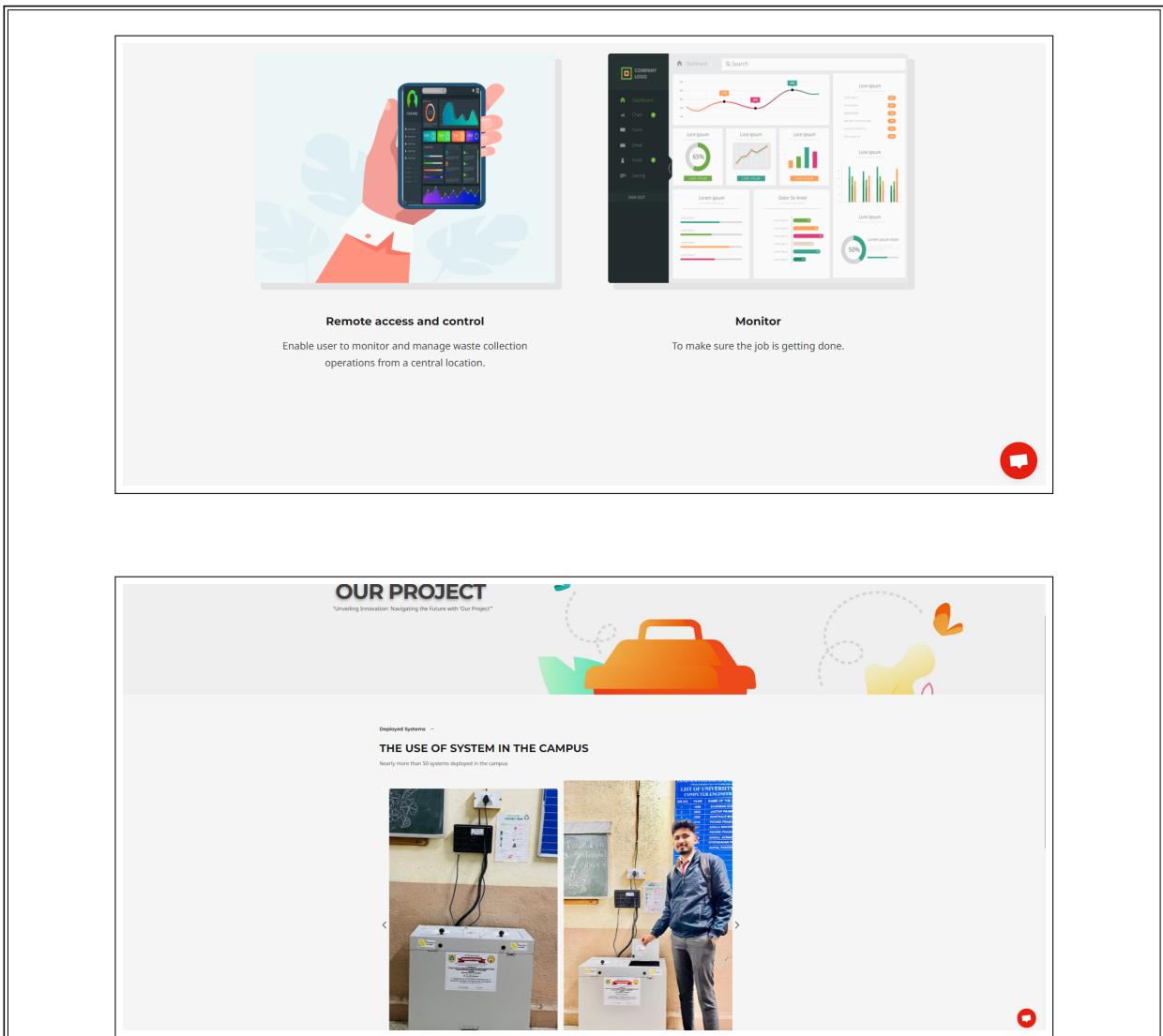
Sanjivani Group of Institutes, Kopargaon, Maharashtra 423603 | Phone: +91 7219747737 | Email: smartcampus.sanjivani@gmail.com

Copyright © 2023 Smart Waste Management System

**Figure 7.6:** Contact Page



**Figure 7.7:** Chatbot Page



**Figure 7.8:** Real-time Execution

Contact Page is the gateway to direct communication and collaboration. We value your interest, questions, and feedback, and our Contact Page is designed to provide a seamless and efficient way for you to connect with us.

1. **Contact Form:** Our user-friendly contact form is the quickest and most convenient way to reach out to us. Simply fill in your details, type your message, and hit send. Whether you have inquiries about our services, want to collaborate, or have general questions, our team is ready to respond promptly.
2. **General Inquiries:** For general inquiries, we provide a dedicated email address where you can send your questions or comments. Our team of experts is committed to providing

comprehensive and informative responses to address your needs.

4. Customer Support: Our customer support information is prominently displayed for any assistance you may require. Whether you're a municipality, a business, or an individual using our services, our support team is here to address technical issues, guide you through processes, and ensure a smooth experience with our smart waste management system.

7. Feedback and Suggestions: We value your feedback as it helps us improve and adapt to better serve your needs. Use our contact page to share your thoughts, suggestions, or experiences. Your input is crucial in our ongoing efforts to enhance our services and make a positive impact on the environment.

## Chapter 8

### APPLICATIONS OF THE PROJECT

- Optimized Waste Collection: Route Optimization: Smart waste management systems use sensors and data analytics to optimize waste collection routes, ensuring that garbage trucks take the most efficient paths. This reduces fuel consumption, lowers emissions, and saves operational costs. Fill-Level Monitoring: Sensors installed in waste bins monitor the fill levels in real-time. This information helps municipalities schedule collections based on actual needs, preventing unnecessary pickups of partially filled bins.
- Cost Reduction and Resource Efficiency: Predictive Maintenance: Smart sensors on waste collection vehicles can monitor their health and performance. This allows for predictive maintenance, reducing downtime and overall maintenance costs. Resource Allocation: Data analytics help authorities allocate resources more efficiently, directing personnel and equipment to areas with the highest waste generation, reducing unnecessary expenditures.
- Environmental Impact Reduction: Emission Reduction: Optimized routes and schedules lead to reduced fuel consumption and lower emissions from waste collection vehicles, contributing to a greener environment. Landfill Optimization: Smart waste systems can help in monitoring landfill capacity and diverting waste to recycling or composting facilities, reducing the environmental impact of landfills.
- Public Awareness and Education: Mobile Apps and Notifications: Smartphone applications provide users with information on waste disposal practices, collection schedules, and recycling locations. Push notifications can remind users to put out their bins or inform them of changes in collection schedules. Education Campaigns: Smart systems enable municipalities to launch targeted education campaigns, using data analyt-

ics to identify areas with low recycling rates and implementing initiatives to improve public awareness and participation.

- Recycling Incentives: Reward Systems: Smart waste management systems can implement reward programs for residents who actively participate in recycling efforts. This can be tracked through user accounts linked to waste disposal practices. Data-Driven Policies: Authorities can use data analytics to identify trends and patterns in waste generation, enabling them to develop policies that encourage sustainable practices and reduce waste.
- Community Engagement: Community Portals: Online platforms and community portals allow residents to engage in discussions, share tips, and participate in local waste management initiatives. Interactive Events: Smart systems can facilitate community events, workshops, and programs to further engage residents in sustainable waste management practices.
- Compliance Monitoring: Regulatory Compliance: Authorities can use smart systems to monitor and enforce waste disposal regulations, ensuring that businesses and individuals adhere to proper waste management practices.

## Chapter 9

# CONCLUSION & FUTURE SCOPE

### 9.1 Conclusion

In conclusion, a smart bin and awareness system harnesses the power of technology to transform traditional waste management. It not only optimizes waste collection processes through real-time data monitoring but also raises environmental awareness by involving users in the management of their waste. This innovative approach promises improved efficiency, reduced costs, and a more sustainable and environmentally conscious society.

### 9.2 Future Scope

1. Integration of IoT and AI: Sensor Advancements: Continued improvement in sensor technologies will enhance real-time monitoring capabilities, enabling more accurate and widespread data collection on waste levels, composition, and environmental factors. Machine Learning Algorithms: Advanced AI algorithms will be employed to analyze data patterns, predict waste generation trends, and optimize collection routes with even greater efficiency.
2. Blockchain for Transparency and Accountability: Supply Chain Traceability: Blockchain technology may be integrated to create transparent and traceable supply chains for waste, ensuring accountability from generation to disposal. This can be particularly valuable in tracking recycled materials.
3. Waste-to-Energy Technologies: Energy Harvesting: Innovations in waste-to-energy technologies may become more integrated into smart waste management systems, allowing for the harvesting of energy from disposed materials to power sensors or contribute to the local energy grid.

4. Advanced Recycling Technologies: Automated Sorting Systems: Implementation of robotic and automated sorting systems at recycling facilities will improve the efficiency and accuracy of material separation, increasing the overall effectiveness of recycling processes. Chemical Recycling: Advancements in chemical recycling technologies may offer more sustainable solutions for complex materials that are challenging to recycle through traditional methods.
5. Augmented Reality (AR) and Virtual Reality (VR) in Education: AR/VR Education Platforms: Future systems may leverage AR and VR technologies to create immersive educational experiences, allowing users to virtually explore waste management processes, recycling facilities, and environmental impact scenarios.
6. Enhanced Public Engagement: Gamification: The incorporation of gamification elements in waste management apps and platforms could incentivize and engage the public, turning responsible waste disposal into a more interactive and rewarding experience. Social Media Integration: Increased integration with social media platforms can amplify awareness campaigns and encourage community participation in sustainable practices.
7. Autonomous Vehicles for Waste Collection: Drone and Robot Collectors: Autonomous vehicles, including drones and robots, may be employed for waste collection in areas that are difficult to reach or where traditional collection methods are inefficient.
8. Global Collaboration and Standardization: International Standards: The development of standardized protocols and international collaboration in smart waste management can enhance interoperability and facilitate the exchange of best practices among different regions and countries.
9. Circular Economy Initiatives: Closed-Loop Systems: The integration of circular economy principles may become more widespread, emphasizing the reduction of waste through design, reuse, and recycling, ultimately creating closed-loop systems for various materials.
10. Data Security and Privacy Measures: Secure Data Handling: As the amount of sensitive data collected increases, future systems will need to prioritize robust cybersecurity measures to protect user information and maintain public trust.

## References

- [1] Alam, M., Dey, A. (2019). IoT-Based Solid Waste Management System for Smart Cities. International Journal of Innovative Technology and Exploring Engineering, 8(10), 3024-3027.
- [2] Central Pollution Control Board. (2016). Annual report on solid waste management. Government of India.
- [3] Ghabraibeh, K., Salahuddin, M. A., Hussini, S. J., Khreishah, A. (2017). Smart Waste Management using Internet of Things (IoT). 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), 1-6.
- [4] Kanchan, D., Sharma, P. (2020). IoT Based Waste Management for Smart Cities. International Research Journal of Engineering and Technology, 7(3), 1349-1353.
- [5] Kumar, A., Samadder, S. R., Chakraborty, S. K. (2017). A review on technological options of waste to energy for effective management of municipal solid waste. Waste management, 69, 407-422.
- [6] Ministry of Environment, Forest and Climate Change. (2016). Solid Waste Management Rules, 2016. Government of India.
- [7] Rana, R., Goyal, R. (2019). Internet of Things for Solid Waste Management: Challenges and Solution. International Journal of Computer Applications, 182(36), 0975-8887.
- [8] Sharma, H. B., Vanapalli, K. R., Cheela, V. R. S., Ranjan, V. P., Jaglan, A. K., Dubey, B., Bhattacharya, J. (2020). Challenges, opportunities, and innovations for effective solid waste management during and post COVID-19 pandemic. Resources, Conservation and Recycling, 162, 105052.

## Annexure A

### Weekly Assessment Report

#### **PROBLEM IDENTIFICATION AND DEFINITION**

Week 1 Progress Report – Smart Waste Management and Awareness System

##### **Introduction:**

The Smart Waste Management and Awareness System project aims to address critical issues related to waste management and environmental sustainability. This report outlines the progress made during the first week of the project. This project is a dynamic initiative focused on improving waste management practices and increasing environmental awareness in our community. This report highlights our achievements during the first week of the project and provides insights into the problem at hand.

##### **Problem Identification:**

Our initial analysis has revealed a significant challenge in efficient waste management and a lack of awareness about the environmental impact of improper waste disposal in our community. Individuals and organizations often struggle to dispose of waste responsibly, leading to environmental pollution and resource wastage. There is a noticeable lack of access to reliable information and resources for proper waste disposal and recycling.

##### **Problem Definition:**

Our primary goal is to develop a comprehensive Smart Waste Management and Awareness System that caters to the needs of the general public. This system will provide information, resources, and guidance on efficient waste disposal, recycling, and environmental sustainability. We aim to design a user-friendly platform that offers real-time data on waste collection, recycling centers, and environmental impact assessments. Our definition of the problem involves creating a platform that bridges the current gaps in waste management and equips individuals and organizations with the tools to contribute to a cleaner and more sustainable environment.

In the coming weeks, our team will focus on data gathering and analysis to ensure that our solution aligns with the needs and preferences of the community. We are committed to making a significant impact on waste management and environmental awareness in our community.

[25/09/2023]

**Figure 9.1:** Weekly Assessment Report 1

## REQUIREMENTS GATHERING

Week 2 Progress Report - Smart Waste Management and Awareness System

**Requirements Gathering:**

During this phase, our team diligently collected responses from the entire Sanjivani college community, including students and staff members. The survey, conducted using Google Forms, centered on questions related to the Smart Bin implementation on our campus and the state of waste management awareness among the community.

- Survey Participants: Our survey included a comprehensive representation of the college community, ensuring diversity in age, background, and roles.
- Survey Questions: The questionnaire covered various aspects, including the usage and perception of the Smart Bins, general health habits, and awareness of health services available on campus.
- Response Rate: We received responses from a significant portion of the college community, showcasing a high level of engagement and interest in the project.

**Data Analysis:**

The data analysis process revealed a valuable range of insights, and it's noteworthy that most of the responses were not entirely positive:

- Smart Bin Utilization: While a significant number of respondents indicated a positive reception of the Smart Bins on campus, there were also some concerns and suggestions for improvements, such as more user-friendly interfaces and better signage.
- Waste Awareness: There's a notable awareness gap among some participants regarding available waste management services and resources on campus. This insight emphasizes the importance of enhancing waste awareness initiatives.
- Interest in waste management Promotion: Many respondents expressed interest in waste management awareness campaigns and educational initiatives, which is a promising sign that the community is open to embracing healthier lifestyles with the right guidance and information.

[03/10/2023]

**Figure 9.2:** Weekly Assessment Report 2

## FRONTEND DEVELOPMENT

Week 3 Progress Report - Smart Waste Management and Awareness System

**Introduction:**

The chatbot development project continues to make strides in our mission to provide efficient and helpful conversational AI. This update report highlights the progress achieved in the third week, with a focus on the development of the chatbot's frontend and the integration of additional features.

**Frontend Development:**

Frontend development is a pivotal aspect of the chatbot development project. During this phase, we concentrated on crafting an appealing and user-friendly interface, leveraging the capabilities of our chosen development framework.

- Framework Integration: We've successfully integrated a robust development framework, enabling the creation of a chatbot with diverse features and customization options.
- Responsive Design: Our chatbot's design is responsive, ensuring a seamless user experience across various devices and screen sizes.
- User-Friendly Interface: The interface is designed to be intuitive, allowing users to easily interact with the chatbot and access the information and services they need.

This frontend development effort establishes the foundation for an interactive and informative chatbot that aligns with our mission to provide efficient assistance in various conversations. We remain committed to making further improvements and enhancing user engagement in the upcoming phases of the project.

**Conclusion:**

The third week of our chatbot development project has witnessed significant progress in frontend development and the integration of valuable user features. Our development framework has empowered us to create an interactive and informative chatbot, fulfilling our mission to provide efficient and helpful conversational AI.

[09/10/2023]

**Figure 9.3:** Weekly Assessment Report 3

## CHATBOT DEVELOPMENT

Week 4 Progress Report - Smart Waste Management and Awareness System

**Introduction:**

The Chatbot Development project continues to make significant strides in our mission to provide innovative and efficient solutions within the awareness system. This report details the progress achieved during the fourth week, with a primary focus on the development and refinement of the chatbot's capabilities.

**Chatbot Development:**

The fourth week of the Chatbot Development project marked a pivotal phase in the creation of an advanced, user-friendly chatbot that can efficiently address user queries and provide valuable assistance.

- Enhanced Natural Language Processing: Our team concentrated on further improving the chatbot's natural language processing capabilities, ensuring it can understand and respond to a wider range of user inputs with greater accuracy.
- Expanded Knowledge Base: Active efforts have been made to expand the chatbot's knowledge base, incorporating the latest information and insights to make it a more valuable resource.
- User Engagement Features: We've integrated interactive features that enhance user engagement, including quick links to relevant resources and the ability to provide feedback to continually enhance the chatbot's performance.

These developments position our chatbot as an advanced and versatile tool that aligns with our mission to provide effective and accessible assistance to our users. As we move forward, we remain committed to refining and expanding the chatbot's capabilities to better serve our community.

**Conclusion:**

The fourth week of our Chatbot Development project has witnessed significant advancements in the chatbot's capabilities, making it a more efficient and user-friendly resource. We are dedicated to continuous improvement to ensure that the chatbot remains a valuable tool for our users within the awareness system.

[16/10/2023]

**Figure 9.4:** Weekly Assessment Report 4

**Integration of Animated Videos for User Engagement**

Week 5 Progress Report - Smart Waste Management and Awareness System

**Introduction:**

The Smart Waste Management and Awareness System has reached its fifth phase, which focuses on enhancing user engagement through the integration of animated videos. The primary objective of this phase is to incorporate animated videos into the system, serving as an educational tool and a means of promoting waste management awareness. These videos will convey important information about sustainable waste disposal, recycling, and other eco-friendly practices, making the platform not only informative but also interactive.

**Key Tasks:**

1. Content Creation: Develop animated videos on various waste management topics, such as recycling, composting, reducing waste, and the environmental impact of different materials. These videos should be clear, concise, and visually appealing.
2. User-Friendly Integration: Ensure seamless integration of the videos into the web application's user interface. Users should easily access and view these videos from their profiles.
3. Video Organization: Implement a well-structured library or section for the videos within the application, categorizing them based on topics for easy navigation.
4. Accessibility: Ensure that the videos are accessible to a broad range of users, including those with disabilities. Provide closed captions and transcriptions to enhance accessibility.
5. User Feedback Mechanism: Implement a feature that allows users to provide feedback on the videos. This feedback can be used to improve the quality and relevance of the content.
6. Scalability: Develop the system with the potential to add new videos and update existing ones as waste management practices and guidelines evolve.

**Conclusion**

The integration of animated videos in the Smart Waste Management and Awareness System represents a significant step in making the platform more informative, engaging, and user-friendly. By providing valuable content in an accessible and appealing format, the system aims to empower users to make environmentally responsible choices and contribute to a cleaner, more sustainable future. This phase aligns with the project's overall goal of fostering waste management awareness and practices.

[23/10/2023]

**Figure 9.5:** Weekly Assessment Report 5

## CHATBOT EARLY TRAINING AND WEBSITE COMPLETION

Week 6 Progress Report - Smart Waste Management and Awareness System

### Introduction:

The sixth week of our Smart Waste Management and Awareness System project marked significant advancements in two key areas: the early training of our chatbot and the completion of the project website. This report outlines the progress made during this week, emphasizing the milestones achieved in both aspects.

### Key Tasks:

This week, our development team focused on initiating the training process, ensuring that the chatbot gains a foundational understanding of waste management, recycling practices, and environmental sustainability.

**Dataset Integration:** We successfully integrated relevant datasets to facilitate the early training of the chatbot. These datasets include information on local waste management practices, recycling guidelines, and environmental impact assessments.

**Initiate RASA :** Our RASA have been deployed to begin the early stages of teaching the chatbot to understand and respond to user inputs related to waste management. This involves refining its language comprehension and contextual understanding.

**User Feedback Loop:** An initial user feedback loop has been established to gather insights on the chatbot's early responses. This iterative process is integral to improving the chatbot's accuracy and user satisfaction.

### Website Completion:

In parallel with chatbot development, our team dedicated efforts to complete the Smart Waste Management and Awareness System website. All project-related content, including information on Smart Bins, waste management resources, and the chatbot's capabilities, has been seamlessly integrated into the website.

### Conclusion:

The sixth week of our project has seen substantial progress in the early training of our chatbot and the completion of the project website. These developments position us well to move forward in delivering a comprehensive and accessible Smart Waste Management and Awareness System to our community. As we continue refining the chatbot's capabilities and ensuring the website's functionality.

[20/11/2023]

**Figure 9.6:** Weekly Assessment Report 6

## WEBSITE HOSTING

Week 7 Progress Report - Smart Waste Management and Awareness System

**Introduction:**

In the seventh week of our Smart Waste Management and Awareness System project, our focus has shifted towards the crucial aspect of hosting the project website. This report outlines the progress made in website hosting, marking a significant step towards making our platform accessible to the wider community.

**Website Hosting:**

Website hosting involves making the project website publicly accessible on the internet. This week, our team has diligently worked on deploying the website to a hosting environment, ensuring seamless performance, security, and accessibility.

**Hosting Platform Selection:** After careful consideration, we selected a reliable hosting platform that aligns with the project's requirements for scalability, performance, and security.

**Domain Configuration:** The project website has been associated with a unique domain name, making it easily identifiable and accessible to users.

**Security Measures:** To safeguard user data and ensure a secure browsing experience, appropriate security measures, such as SSL certification, have been implemented.

**Testing and Debugging:** Rigorous testing and debugging procedures were undertaken to identify and address any issues that may arise during the hosting process, ensuring a smooth user experience.

**Conclusion:**

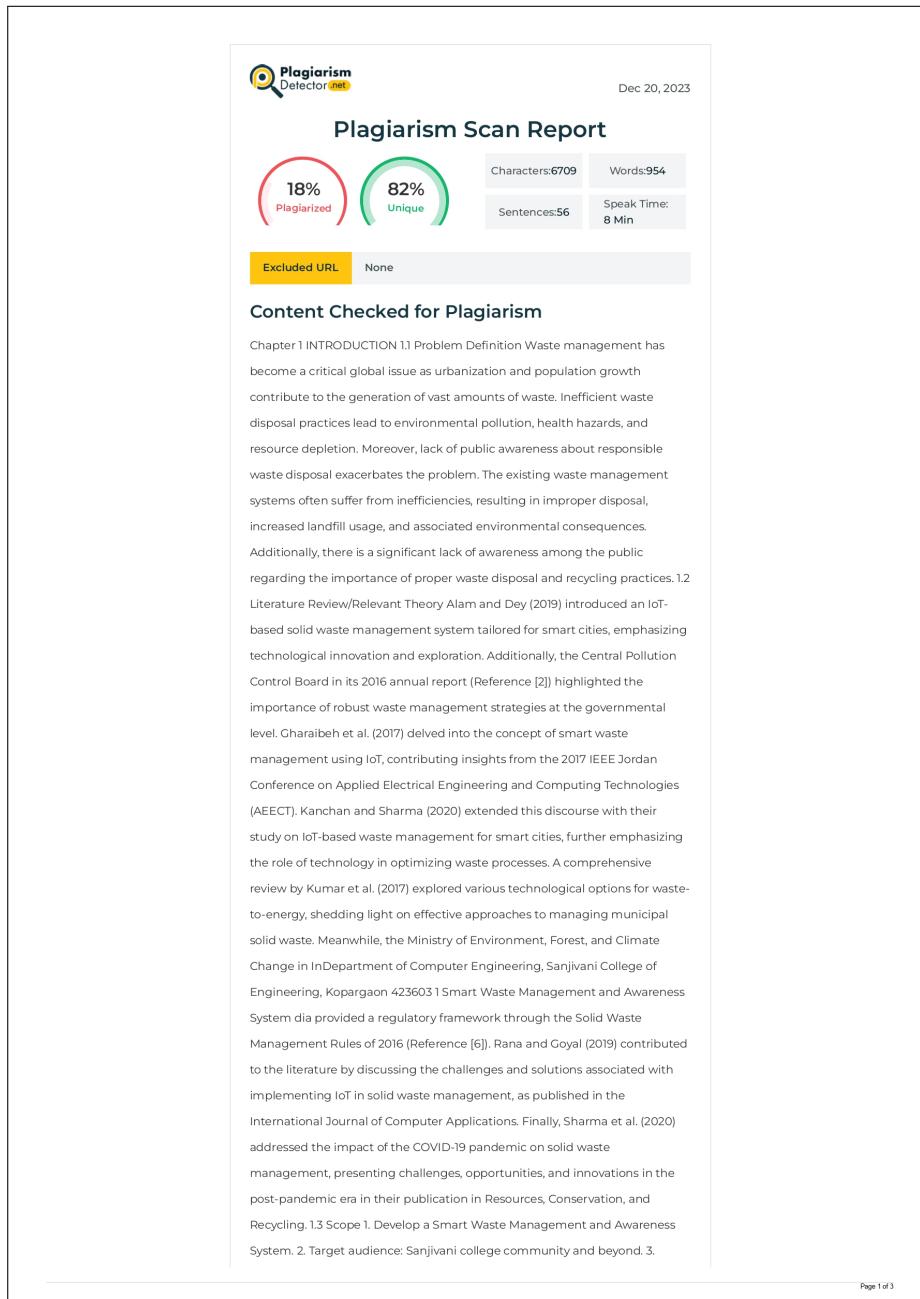
The seventh week represents a pivotal stage in our project, as we successfully hosted the Smart Waste Management and Awareness System website. This achievement brings us one step closer to realizing our goal of providing valuable information and resources to the community. As we enter the next phase, the team remains dedicated to refining the website, addressing user feedback, and continuing to advance our mission of promoting sustainable waste management practices.

[28/11/2023]

**Figure 9.7:** Weekly Assessment Report 7

## Annexure B

### Plagiarism Report



**Figure 9.8:** Plagiarism Report

Integration of Smart Bins, chatbot, animated videos, and comprehensive website. 4. Deployment of IoT sensors on waste bins to monitor and transmit real-time data on fill levels. 5. Development of a centralized server infrastructure for efficient storage, processing, and analysis of data collected from sensors. 6. Implementation of a smart routing algorithm to optimize waste collection routes based on fill-level data, minimizing fuel consumption and reducing the environmental impact. 7. Creation of a user-friendly mobile application to provide residents with real-time information on waste levels, recycling locations, and guidelines for responsible waste disposal.

Objectives - To Improve waste management existing system to provide real-time data on waste collection and recycling centers. - To Enhance user engagement through a userfriendly chatbot. - To Integrate animated videos for informative and interactive content.

Department of Computer Engineering, Sanjivani College of Engineering, Kopargaon 423603

2 Smart Waste Management and Awareness System

Department of Computer Engineering, Sanjivani College of Engineering, Kopargaon 423603

3 Smart Waste Management and Awareness System

Chapter 2 REQUIREMENT ANALYSIS

Requirements Analysis or requirement engineering is a process of determining user expectations for new software or providing updates for previous products. These core points must be measurable, relevant and detailed. In the software engineering field this term is also called functional specifications. Requirements analysis mainly deals with communication with users or customers to determine system feature expectations, requirements and reduce convicts as demanded by various software users. Energy should be directed towards ensuring that the system or product conforms to user needs rather than attempting to turn user expectations to the requirements.

2.1 Requirement Specifications

Requirement specification describes the function and performance of the computer based system and constraints which govern its development. It can be a written document, a set of graphical models, a collection of scenarios, or any combination of above.

These are of 3 types: 1. NR: Normal Requirements 2. ER: Expected Requirements 3. XR: Exciting Requirements

2.1.1 Normal Requirements

These are the requirements which are clearly stated by the customer so all these requirements will be present in the project for user satisfaction.

Department of Computer Engineering, Sanjivani College of Engineering, Kopargaon 423603

4 Smart Waste Management and Awareness System

- NR1: The system should target audience for awareness: Sanjivani college community and beyond.

- NR2: Reduces the effect of waste on the environment and health.

- NR3: Enhance user engagement through a userfriendly chatbot.

2.1.2 Expected Requirements

These requirements are expected by the customer but not clearly stated by the customer. These are implicit types of requirements.

- ER1: System should be fast and reliable.

- ER2: System should have a neat and clean user interface.

- ER3: System should provide instruction of usage to the user.

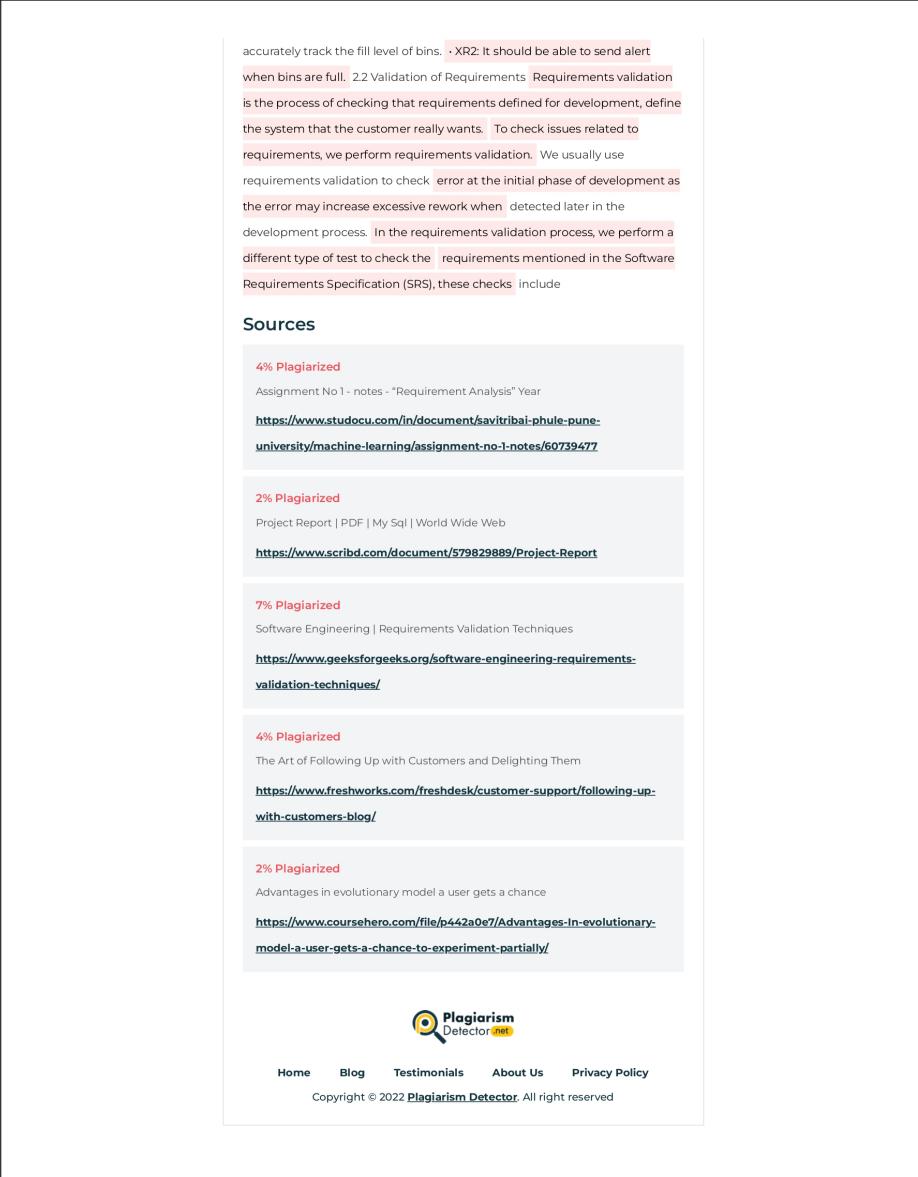
2.1.3 Exciting Requirements

These requirements are not stated by the customer but externally provided by the developer *in order to maintain a good relationship with the customer.*

- XR1: It should be able to

Page 2 of 3

**Figure 9.9:** Plagiarism Report



The screenshot shows a plagiarism report from Plagiarism Detector. The report lists five sources, each with its percentage of plagiarism and a link to the original document.

- 4% Plagiarized**  
Assignment No1 - notes - "Requirement Analysis" Year  
<https://www.studocu.com/in/document/savitribal-phule-pune-university/machine-learning/assignment-no-1-notes/60739477>
- 2% Plagiarized**  
Project Report | PDF | My Sql | World Wide Web  
<https://www.scribd.com/document/579829889/Project-Report>
- 7% Plagiarized**  
Software Engineering | Requirements Validation Techniques  
<https://www.geeksforgeeks.org/software-engineering-requirements-validation-techniques/>
- 4% Plagiarized**  
The Art of Following Up with Customers and Delighting Them  
<https://www.freshworks.com/freshdesk/customer-support/following-up-with-customers-blog/>
- 2% Plagiarized**  
Advantages in evolutionary model a user gets a chance  
<https://www.coursehero.com/file/p442a0e7/Advantages-In-evolutionary-model-a-user-gets-a-chance-to-experiment-partially/>

At the bottom, there is a navigation menu with links to Home, Blog, Testimonials, About Us, and Privacy Policy. A copyright notice states "Copyright © 2022 Plagiarism Detector. All right reserved".

**Figure 9.10:** Plagiarism Report