



**UNIVERSITY OF  
SIALKOT**  
A CHARTERED UNIVERSITY

**Faculty of Computing & IT**  
**University of Sialkot**

**SMART GARBAGE MANAGEMENT FOR**  
**SUSTAINABLE CITY LIFE**

**Session: BS-SE Fall 2018-2022**

**Project Advisor: Dr. Sabeen Javaid**

**Submitted By**

*hammad arshad* 18101001-072

*seher jamil* 18101001-074

*abdul wahab* 18101001-086

---

Faculty of Computing & IT  
University of Sialkot

---

## **DECLARATION**

I certify that project title Smart garbage management for sustainable city life is under my supervision with students of Bachelor of Science (Software Engineering), Faculty of Computing & Information Technology, University of Sialkot, Pakistan, worked under my supervision.

### **Dr. Sabeen Javed**

---

Supervisor's Name  
Department of Software Engineering  
Faculty of Computing & Information Technology  
University of Sialkot, Punjab, Pakistan.  
Dated: Wednesday, June 15, 2022

## **ACKNOWLEDGMENT**

We truly acknowledge the cooperation and help by our supervisor, Dr. Sabeen Javed Faculty member of the Department of Software Engineering, University of Sialkot, Sialkot. She has been a constant source of guidance throughout the course of this project. We are also thankful to our friends and families whose silent support led us to complete our project. We are also thankful to our friends and families whose silent support led us to complete our project.

1. Hammad Arshad
2. Seher Jamil
3. Abdul Wahab

Date: Wednesday, June 15, 2022

## ABSTRACT

Pakistan, with a population of 227 million, is the world's fifth-most populous country. The heap of garbage is a common sight in the urban areas of Pakistan. Mismanagement of garbage management causes soil erosion, air, and water contamination which ultimately creates health and environmental hazards for the citizens of Pakistan. Diarrhea, pneumonia, and the release of methane are a few examples of these hazards. It is, therefore, important to devise an efficient mechanism for waste management. With the help of modern and emerging technologies, the concerned authorities can reform the process of waste management in the right direction in Pakistan.

This project proposes a comprehensive solution for waste management. It is a software and hardware-based solution that will facilitate municipal corporations, residents of urban areas, and other concerned authorities for the effective management of waste. To reap the benefits of an integrated system for waste management, this project utilizes the concepts of the Internet of Things (IoT) and Deep Learning. The system will be a hybrid application (i.e., usage of web and mobile-based technologies). It will identify the piles of garbage on the streets and the roads. For this purpose, sensors such as cameras will be installed in the various locations of a city. These cameras will send images to a web server through microcontrollers where a deep learning-based trained model will identify the presence of the piles of garbage. After the identification of the garbage, the system will generate alerts for the authorities and janitorial staff of that area. It will also send the exact location of the piles of garbage to the mentioned authorities and the staff. The tasks of the waste collection can be assigned automatically by identifying the janitorial staff on duty. On the other hand, the head of the janitorial staff can also assign a task if needed. The system keeps checking the status of the assigned task (if the piles of garbage are collected or not) after the interval set by the head. After the task completion, it generates a report. Moreover, the system also keeps an eye on the performance of the staff in each area. It also identifies the staff of the month and the year which are selected on the basis of the cleanliness of the assigned area. The residents of an area can also lodge complaints regarding the cleanliness of their areas through the mobile application. Consequently, making a contribution to maintaining cleanliness in their city. In short, the proposed system will bring the stakeholders in one place for achieving the sustainable goal of a neat and clean Pakistan.

## TABLE OF CONTENTS

<b>CHAPTER 1: PROJECT FEASIBILITY REPORT .....</b>	<b>1</b>
.....	1
1.0. INTRODUCTION .....	2
1.1. PROBLEM STATEMENT.....	2
1.2. OBJECTIVES.....	2
1.3. PROJECT MOTIVATION .....	2
1.4. PROJECT/PRODUCT FEASIBILITY REPORT .....	2
1.4.1. Technical Feasibility.....	3
1.4.2. Operational Feasibility.....	3
1.4.3. Economic Feasibility .....	3
1.4.4. Schedule Feasibility.....	4
1.4.5. Specification Feasibility.....	4
1.4.6. Information Feasibility .....	4
1.5. PROJECT/PRODUCT SCOPE.....	4
1.6. PROJECT/PRODUCT COSTING .....	5
1.7. TASK DEPENDENCY TABLE .....	8
1.8. CPM - CRITICAL PATH METHOD .....	8
1.9. GANTT CHART .....	9
1.10. INTRODUCTION TO TEAM MEMBER AND THEIR SKILL SET .....	10
1.11. TASK AND MEMBER ASSIGNMENT TABLE .....	10
1.12. TOOLS AND TECHNOLOGY WITH REASONING .....	11
<b>CHAPTER 2: SOFTWARE REQUIREMENT SPECIFICATION .....</b>	<b>12</b>
.....	12
2.1 INTRODUCTION: .....	13
2.1.1 Systems Specifications .....	13
2.1.2. Identifying External Entities .....	15
2.1.3. Capture "shall" Statements .....	16
2.1.4. Allocate Requirements .....	17
2.1.5. Prioritize Requirements .....	18
2.1.6. Requirements Trace-ability Matrix.....	20
2.2. USE CASE DIAGRAMS .....	22
2.2.1. High Level Use Case Diagram .....	22
.....	23
2.2.2. Analysis Level Usecase Diagram: .....	24
2.2.3. Usecase Description .....	28
<b>CHAPTER 3: SOFTWARE DESIGN &amp; ARCHITECTURE .....</b>	<b>44</b>
.....	44
3.1 INTRODUCTION .....	45
3.2 SEQUENCE DIAGRAM.....	45
3.3. CLASS DIAGRAM.....	46
3.4. STATE TRANSITION DIAGRAM.....	46
3.5. ENTITY RELATION DIAGRAM .....	47
.....	47
3.6. DOMAIN MODEL DIAGRAM.....	48
3.7. DATA FLOW DIAGRAM.....	48

<b>CHAPTER 4: USER INTERFACE DESIGN .....</b>	<b>49</b>
.....	49
4.1. <i>INTRODUCTION</i> .....	50
4.2. <i>SITE MAPS</i> .....	50
4.3. <i>STORY BOARDS</i> .....	50
<b>CHAPTER 5: SOFTWARE TESTING.....</b>	<b>66</b>
5.1. <i>INTRODUCTION</i> .....	67
5.2. <i>BLACK BOX PLAN/WHITE BOX PLAN</i> .....	67
5.2.1. <i>Black Box Testing</i> .....	67
5.2.2. <i>White Box Testing</i> .....	85

## TABLE OF FIGURES

<i>Figure 1 : Network Diagram .....</i>	9
<i>Figure 2 : Gantt Chart.....</i>	10
<i>Figure 3 : Organizational Chart.....</i>	14
<i>Figure 4 : High Level Use Case .....</i>	23
<i>Figure 5 : Live View Garbage .....</i>	24
<i>Figure 6 : Get Alert.....</i>	24
<i>Figure 7 : Manage Area Supervisor .....</i>	25
<i>Figure 8 : Employee of the year .....</i>	25
<i>Figure 9 : Status Clear .....</i>	26
<i>Figure 10 : Garbage Detection.....</i>	26
<i>Figure 11 : Employee of the month.....</i>	27
<i>Figure 12 : Check Clear Status .....</i>	27
<i>Figure 13 : Register .....</i>	28
<i>Figure 14 : Sequence Diagram.....</i>	45
<i>Figure 15 : Class Diagram .....</i>	46
<i>Figure 16 : State Transition Diagram .....</i>	46
<i>Figure 17 : Entity Relation Diagram.....</i>	47
<i>Figure 18 : Domain Model Diagram.....</i>	48
<i>Figure 19 : Data Flow Diagram.....</i>	48
<i>Figure 20 : Site Map .....</i>	50
<i>Figure 21 : Login Screen .....</i>	51
<i>Figure 22 : Register Screen .....</i>	51
<i>Figure 23 : Welcome Screen.....</i>	52
<i>Figure 24 : Live Stream Recyclable Detection .....</i>	53
<i>Figure 25 : Live Stream Pile Detection .....</i>	53
<i>Figure 26 : Live Stream No Pile Detection .....</i>	54
<i>Figure 27 : Our Services Screen.....</i>	54
<i>Figure 28 : About Screen .....</i>	55
<i>Figure 29 : Dashboard Screen .....</i>	55
<i>Figure 30 : Profile Screen .....</i>	56
<i>Figure 31 : API Screen for Authentication .....</i>	57
<i>Figure 32 : API Screen for Blogs .....</i>	57
<i>Figure 33 : API Screen for Contacts .....</i>	58
<i>Figure 34 : Blogs Screen .....</i>	58
<i>Figure 35 : Login Screen .....</i>	59
<i>Figure 36 : Splash Screen.....</i>	59
<i>Figure 37 : Home Screen .....</i>	60
<i>Figure 38 : Register Screen .....</i>	60
<i>Figure 39 : Profile Screen .....</i>	61
<i>Figure 40 : Camera Screen.....</i>	61
<i>Figure 41 : Map Screen .....</i>	62
<i>Figure 42 : Get Location .....</i>	62
<i>Figure 43 : Mark Complete Status.....</i>	63
<i>Figure 44 : Employee of Month .....</i>	63
<i>Figure 45 : Contact Us .....</i>	64
<i>Figure 46 : Employee of Year .....</i>	64
<i>Figure 47 : About 2 .....</i>	65
<i>Figure 48 : About 1 .....</i>	65
<i>Figure 49 : Contact Us Test Pass .....</i>	86

<i>Figure 50 : Contact Us Test Fail (Wrong Credentials) .....</i>	87
<i>Figure 51 : Blog Test Fail (Wrong Credentials) .....</i>	87
<i>Figure 52 : Selenium to Test Register.....</i>	88
<i>Figure 53 : Selenium to Test Contact .....</i>	88
<i>Figure 54 : Selenium to Test Login.....</i>	89
<i>Figure 55 : Inspection of Login Screen (Email Key) .....</i>	89
<i>Figure 56 : Inspection of Login Screen (Password Key).....</i>	90
<i>Figure 57 : Inspection of Register Screen (Empty Fields) .....</i>	90
<i>Figure 58 : Inspection of Home Screen .....</i>	91
<i>Figure 59 : Inspection of Bottom Navigation &amp; Profile Screen .....</i>	91
<i>Figure 60 : Inspection of Logout .....</i>	92
<i>Figure 61 : Inspection of Map Screen .....</i>	92

## LIST OF TABLES

Table 1 : Scope of Project.....	4
Table 2 : Total Count .....	5
Table 3 : Complexity Adjustment Factor.....	6
Table 4 : Modes of COCOMO Model .....	7
Table 5 : Parameters of Different Modes.....	7
Table 6 : Task Dependency Table .....	8
Table 7 : Sequence of the Activities .....	9
Table 8 : Task and Member Assignment Table .....	10
Table 9 : System Scope.....	14
Table 10 : Capture Shall Statements .....	16
Table 11 : Allocate Requirements.....	17
Table 12 : Prioritize Requirements .....	19
Table 13 : Requirements Trace-ability Matrix.....	21
Table 14 : UC-1 User Login .....	28
Table 15 : UC-2 Validate Email .....	29
Table 16 : UC-3 Validate Password.....	29
Table 17 : UC-4 Authenticate User .....	30
Table 18 : UC-5 Register User.....	30
Table 19 : UC-6 Validate User .....	31
Table 20 : UC-7 Update Profile.....	31
Table 21 : UC-8 Detect Garbage .....	32
Table 22 : UC-9 Label Piles of Garbage.....	32
Table 23 : UC-10 Location Coordinates.....	33
Table 24 : UC-11 Identify Recyclables .....	33
Table 25 : UC-12 Send Alert .....	34
Table 26 : UC-13 Receive Alert .....	34
Table 27 : UC-14 Completed Status .....	35
Table 28 : UC-15 Maximum Alerts .....	35
Table 29 : UC-16 Minimum Alerts.....	36
Table 30 : UC-17 Status Clear List.....	36
Table 31 : UC-18 Status Update .....	37
Table 32 : UC-19 Employee of Month .....	37
Table 33 : UC-20 Employee of Year .....	38
Table 34 : UC-21 User Details.....	38
Table 35 : UC-22 Add Area Supervisor .....	39
Table 36 : UC-23 Remove Area Supervisor .....	39
Table 37 : UC-24 Update Area Supervisor .....	40
Table 38 : UC-25 View Area Supervisor.....	40
Table 39 : UC-26 Add TMA Employee .....	41
Table 40 : UC-27 Remove TMA Employee .....	41
Table 41 : UC-28 Update TMA Employee.....	42
Table 42 : UC-29 View TMA Employee.....	42
Table 43 : UC-30 Logout.....	43
Table 44 : Test Case 1.....	67
Table 45 : Test Case 2.....	68
Table 46 : Test Case 3.....	68
Table 47 : Test Case 4.....	69
Table 48 : Test Case 5.....	70
Table 49 : Test Case 6.....	70

Table 50 : Test Case 7.....	71
Table 51 : Test Case 8.....	72
Table 52 : Test Case 9.....	72
Table 53 : Test Case 10.....	73
Table 54 : Test Case 11.....	74
Table 55 : Test Case 12.....	74
Table 56 : Test Case 13.....	75
Table 57 : Test Case 14.....	76
Table 58 : Test Case 15.....	76
Table 59 : Test Case 16.....	77
Table 60 : Test Case 17.....	78
Table 61 : Test Case 18.....	78
Table 62 : Test Case 19.....	79
Table 63 : Test Case 20.....	80
Table 64 : Test Case 21.....	80
Table 65 : Test Case 22.....	81
Table 66 : Test Case 23.....	82
Table 67 : Test Case 24.....	82
Table 68 : Test Case 25.....	83
Table 69 : Test Case 26.....	84
Table 70 : Test Case 27.....	84
Table 71 : Branch Testing Login Module.....	85
Table 72 : Branch Testing Register Module .....	86



## **Chapter 1: Project Feasibility Report**

---

## **1.0. Introduction**

One of the biggest problems that municipalities around Pakistan have to deal with is waste management. Everyday waste leaves the homes of citizens and then arrives in the designated waste collection areas, which in turn have to be transported by waste workers to waste collection and treatment centers. Unfortunately, in this process, various problems arise which have an impact on the environment and on people's quality of life. In Pakistan, there are a lot of things that need to solve to make Pakistan a well-developing country. This is one of the major problems that need to be solved. This makes Pakistan dirty & unhealthy for our health because that thing creates a lot of diseases which make people ill. As Countries are going smart day by day there is a need to develop the intelligent & most efficient way to solve this problem. For this problem, this system enables Pakistan to become neat & clean. The System detects garbage through cameras & generates alerts to the system. The department of Refinement sends the team to the particular area & cleans the whole garbage/garbage from that area. This makes us safe from many of the major diseases like Malaria & Dengue caused by mosquitos.

### **1.1. Problem Statement**

Piles of garbage in Pakistan is the major issue which causes many major problems like pollution & diseases like Dengue, Malaria that causes from mosquitoes. Everyday waste leaves from home & in Pakistan, most of the time peoples throw trash in the corner of the street which mainly cause of piles of garbage. There's a need to fill this lack in our country to make our country neat & clean.

### **1.2. Objectives**

Smart Garbage Management for sustainable city life is a system used to detect garbage from the streets & generates an alert to the system. To Makes our cities clean there's a dire need for this project. This project is better than others because the previous project that was done only detects some piece of garbage or sensors is applied in the lid of the dustbins. but there didn't exist such a system that could help detect the whole accumulation of garbage.

### **1.3. Project Motivation**

The Piles of garbage present at different places of city which inspires us to fix that problem. The team members inspired by this problem. Moreover, some failures during the development of this project also inspired our team which lead us to great achievement in the form of completion of project and successfully solving that problem.

### **1.4. Project/Product Feasibility Report**

Feasibility means the extent to which appropriate data and information are readily available or can be obtained with available resources such as staff, expertise, time, and equipment. It is used to measure of how practical or beneficial the developed software for us or an organization. Following are some of the types of feasibility.

- Technical
- Operational
- Economical

- Schedule
- Specification
- Information
- Motivational
- Legal and ethical

#### **1.4.1. Technical Feasibility**

Technical feasibility is concerned with the Important question that weather the software or system can be developed. as the technical feasibility demands the expertise and technology that are important for the project so in development of our system, we are capable of making this system and we have the expertise to develop this system. we are going to make flutter application and web panel by using Django platform.

#### **1.4.2. Operational Feasibility**

Operational feasibility checks the ability of a staff to operate the system or how easily can the staff use our possessed system. We are going make our app and web easy to use. We will try our best to produce such a product that users can easily operate without facing any technical fault. In case of any issue, the whole team will be readily available to solve the issue as soon as possible. We considered that our team has enough capability of tackling with the technical issues.

#### **1.4.3. Economic Feasibility**

Our system is economically feasible as our systems does not exceeds the expected cost. In order to check the economic feasibility of a system, cost benefit analysis is done which tells the expected cost and benefits are evaluated.

##### **1. Cost estimation:**

###### **A) Development cost:**

In the development of the app and web of our system, we didn't need any cost on the development of both of them because we had the developer in our own team.

###### **B) Maintenance cost:**

Maintenance is very important for the success of any software. As the developer belongs to our own team so there will also not any maintenance cost required if we need to maintain our system in future.

##### **2. Benefits Estimates:**

###### **A) Tangible Benefit:**

Our foremost benefit is that as we are working in team so we are initially developing our whole system so in this we save a lot of money that are required for the development of the system and increased our revenue. Moreover, we also Think to generate revenue by:

- 1) Brand advertisement
- 2) Through government funds.
- 3) Google ad sons and pro features.

#### **B) Non-Tangible Benefit:**

Non-tangible benefits of our project are that it provides complete satisfaction to our customers and users which will lead to an increase in the use of our project and that will eventually increase our profits.

#### **1.4.4. Schedule Feasibility**

Schedule Feasibility is the most important in the timely completion of the project. In order to check the schedule Feasibility, we developed a Gantt chart to show the completion date of the project. we will work for 7 to 8 hours daily and according to our Gantt Chart it was clearly shown that our project will be completed 15 days earlier. Our schedule is quite flexible and it is least probable that we will be late in achieving our milestones.

#### **1.4.5. Specification Feasibility**

All requirements are completed within the time. All requirements are clear. We have acquired requirements for our projects on focusing the demands of the public users and the problems they faced. These Requirements can be found in the requirement elicitation part of our document. We had mentioned all the constraints and Conditions of the project. Specification feasibility also included hardware and software specification required for completion of our project.

#### **1.4.6. Information Feasibility**

All the given information in our system is meaning full and reliable. All the mentioned requirements are clear and didn't any ambiguity in our requirements. Information Quality is very good and true information is given. As the scope of our system defines that our system is the good opportunity to launch our app so that it will facilitate the Municipality department.

#### **1.4.7. Motivational Feasibility**

Our supervisor motivates us a lot so that our all group is trying our best to perform the donating task of development of the system. We realize that how big this is going to be and we as a team are the source of motivation for ourselves.

#### **1.4.8. Legal & Ethical Feasibility**

This project is exclusively Being developed for learning and providing a platform to municipal committee to enhance the cleanliness issue of our country. The services provided by "The smart garbage Detection" is not only available on a single platform. This project is feasible legally and ethically.

### **1.5. Project/Product Scope**

**Table 1 : Scope of Project**

For	Municipality Committee (Govt. or privately operated)
What	To Make Cities neat & clean from piles of garbage & protect peoples from various types of diseases
The	Smart Garbage Management for Sustainable City Life (SGM)

Is	Web Application Android Mobile Application
That	Help To remove piles of garbage from the streets & make cities look neat & clean

### 1.6. Project/Product Costing

The following five information domain properties are determined and counts are written below here.

**Total external Inputs** = 1  
**Total external Outputs** = 4  
**Total user inquiries** = 5  
**Total internal logical files** = 1  
**Total external interface files** = 1

**Table 2 : Total Count**

Information Domain Value	Low values	Average Values	High values	FP Count
External Input	$0*3=0$	$4*1=4$	$0*6=0$	4
External Output	$0*4=0$	$4*5=20$	$0*7=0$	20
User Inquiries	$0*3=0$	$5*4=16$	$0*6=0$	20
Internal Logical Files	$1*7=7$	$0*10=0$	$0*15=0$	7
External Interface Files	$1*5=5$	$0*7=0$	$0*10=0$	5
Total Count	-	-	-	<b>56</b>

### Complexity Adjustment Factors:

The value adjustment factor is based on 14 general system characteristics (GSC's) that rate the general functionality of the application being counted. For each of the General System Characteristics a rating from 0 to 5 can be provided which represents

- 0=No influence
- 1=Incidental influence
- 2=Moderate influence
- 3=Average influence
- 4=Significant influence
- 5=Strong influence throughout

**Table 3 : Complexity Adjustment Factor**

S#	Factor	Value (0-5)
1	<b>Ease of Installation</b>	<b>5</b>
2	<b>Ease of Operational</b>	<b>5</b>
3	<b>Multiple Installation</b>	<b>4</b>
4	<b>Transaction Rate</b>	<b>4</b>
7	<b>End User Efficiency</b>	<b>5</b>
8	<b>Live Upgradation</b>	<b>2</b>
9	<b>Complex Processing</b>	<b>3</b>
10	<b>Reusability</b>	<b>4</b>
11	<b>Data Communication/Flow</b>	<b>5</b>
12	<b>Distributed Data Processing</b>	<b>4</b>
13	<b>Performance</b>	<b>5</b>
14	<b>Changes/Substitutions</b>	<b>4</b>
	<b>Total =</b>	<b>46</b>

$$\text{CAF} = [0.65 + 0.01 * 0]$$

$$= [0.65 + 0.01 * (46)] = 1.09$$

$$\text{FP est.} = \text{Count Total} * \text{CAF}$$

$$\text{FP est.} = 56 * 1.09$$

$$\text{FP est.} = \mathbf{61.04}$$

For Our Project

Average Productivity = 26 FP/PM

Labor Rate (LR) = 36,400 RS/Month

**Total Effort = FP / productivity**

= $61.04 / 26 = 2.34$  pm

Cost = LR / productivity

=  $36400 / 26 = 1400$  Rs/FP

**Total Project Cost = FP est. \* (Cost/FP)**

=  $61.04 * (1400)$

=**85,456RS.**

### Modes of COCOMO Model

**Table 4 : Modes of COCOMO Model**

Parameters	Organic	Semi-Detached	Embed
Size	2-50 KLOC	50-300 KLOC	300KLOC or above
Team	Small	Medium	Large
Developer	Experienced	Average	Little Experience
Environment	Familiar	Less familiar	Changed
Innovation	Little	Medium	Major Innovation
Deadline	Flexible	Medium	Tight deadline

### Parameters of Different Modes

**Table 5 : Parameters of Different Modes**

Mode	A	B	C	D
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embed	3.6	1.20	2.5	0.32

#### Effort:

$$E = A(KLOC)^B$$

$$E = 2.4(3)^{1.05}$$

$$E = 7.94 \text{ Person/Month}$$

#### Development Time:

$$\text{Dev Time} = C(Effort)^D$$

$$\text{Dev Time} = 2.5(7.94)^{0.38}$$

$$\text{Dev Time} = 3.112 \text{ Months}$$

---

$$\text{Average Staff Size} = \text{Effort} / \text{Dev Time}$$

---

Average Staff Size = 7.94 / 3.112

Average Staff Size = 2.551

**Productivity** = KLOC/Effort

Productivity = 3/7.94

Productivity = 0.377

### **1.7. Task Dependency Table**

Task Dependency is the relationship in which a task or a milestone are relies on another task to be performed. This is also referred to as a logical relationship. A logical relationship can be a dependency between the project task.

**Table 6 : Task Dependency Table**

<b>Task</b>	<b>Duration(weeks)</b>	<b>Dependencies</b>
T1	2	None
T2	2	A(Feasibility)
T3	2	B (Requirement Specification)
T4	3	B, C (Requirement Specification, Design Document)
T5	6	C (Design Document)
T6	20	D, E (web panel design, model training)
T7	6	F (App Development)
T8	2	G (Implementation)
T9	10	H(Testing)

### **1.8. CPM - Critical Path Method**

#### **1. Specify the Individual Activities**

In smart garbage detection using sustainable city life. These activities are performed on the weekly bases and there Gantt chart is also developed.

- 1) Analysis of system
- 2) Planning
- 3) Requirement Gathering
- 4) Detail Design
- 5) Implementation
- 6) Testing
- 7) Deployment
- 8) Final report and closing

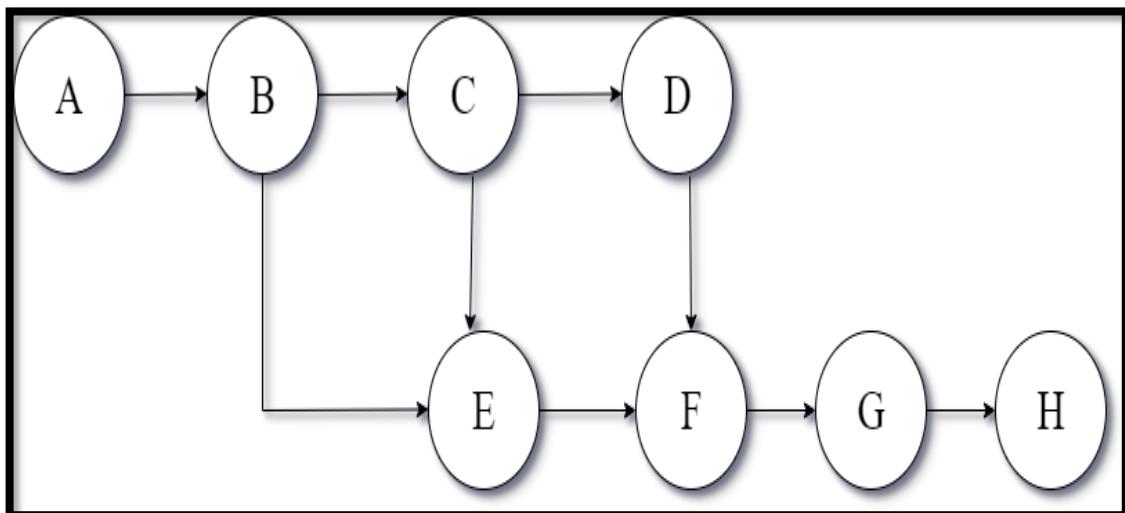
#### **2. Determine the Sequence of the Activities**

Sequence Activities is the process of identifying and documenting Relationships among the project activities.

**Table 7 : Sequence of the Activities**

<b>Activity</b>	<b>Name (Activity)</b>	<b>Immediate Predecessor</b>	<b>Duration (weeks)</b>
A	Analysis of system	None	2
B	Planning	A	2
C	Requirement Gathering	B	3
D	Detail Design	C	2
E	Implementation	B, C	3
F	Testing	D, E	3
G	Deployment	F	2
H	Final report and closing	G	2

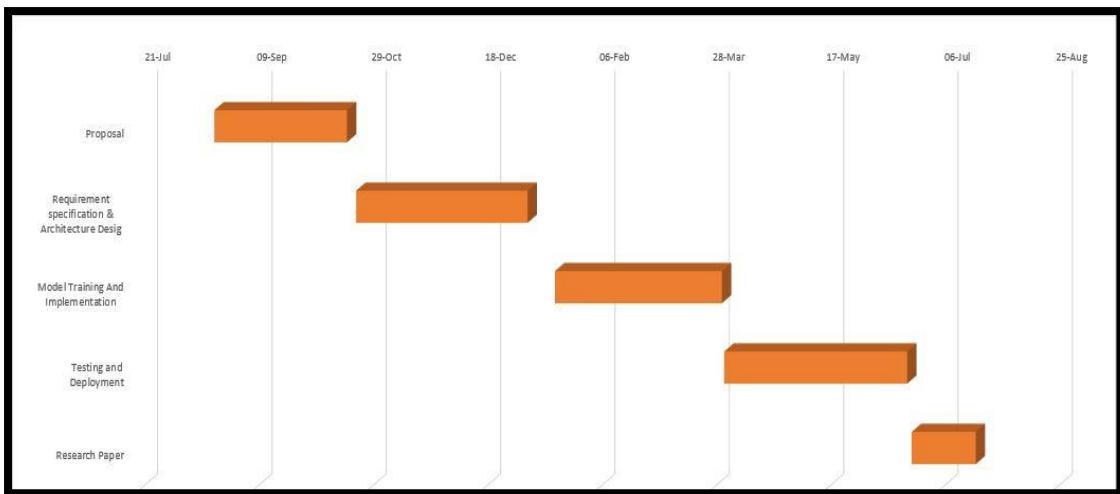
### 3. Draw the Network Diagram



**Figure 1 : Network Diagram**

### 1.9. Gantt chart

The Gantt chart enumerates the activities to be performed on the vertical axis and their corresponding duration on the horizontal axis. It is possible to schedule activities by either early start or late start logic. In the early start approach, each activity is initiated as early as possible without violating the precedence relations. In the late start approach, each activity is delayed as much as possible as long as the earliest finish time of the project is not compromised. The Gantt Chart of Smart Garbage System for Sustainable City Life is as follows:



*Figure 2 : Gantt Chart*

### **1.10. Introduction to Team member and their skill set**

- Hammad Arshad (Major work in development and in Document as well)
- Abdul Wahab (Major Work in Document Designing and some work in development)
- Seher Jamil (Works in Document Designing)

### **1.11. Task and Member Assignment Table**

**Table 8 : Task and Member Assignment Table**

Task	Duration (Weeks)	Dependencies
Proposal	2	M1, M2, M3
SRS	2	M2, M3
SDA	3	M2, M3
Web & Mobile Application Development	2	M1
Model Training & Binding	3	M1
Testing	3	M2
Deployment	2	M1

## ***1.12. Tools and Technology with reasoning***

### **Software**

**Technologies:** (Python, Django 4.0.3, Flutter 2.10.5, Firebase/SQLite Rest API, Html 5, CSS 3, Bootstrap 5, JavaScript ES6, Machine Learning Libraries, YOLO V4)

**Tools:** (PyCharm, Android Studio, Google Colab, VS code)

### **Hardware**

- Arduino UNO R3
- Camera
- GPS Module
- Mobile Phone or Laptop
- Jumper wires (Male, Female)
- Bread Board
- WIFI module ES-8266

### **Sensors**

- Camera

## **Chapter 2: Software Requirement Specification**

---

## **2.1 Introduction:**

Requirements engineering process provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirements as they are transformed into an operational system. The task of capturing, structuring, and accurately representing the user's requirements so that they can be correctly embodied in systems which meet those requirements (i.e., are of good quality).

Here, requirements specification is to be discussed. Requirement's specification would lead to the following four steps:

- Identify external interfaces
- Development of context diagram
- Capture “shall statements
- Allocate requirements
- Prioritize requirements
- Development of requirements traceability matrix

### **2.1.1 Systems Specifications**

The following are the clauses that must be included while describing the system specifications.

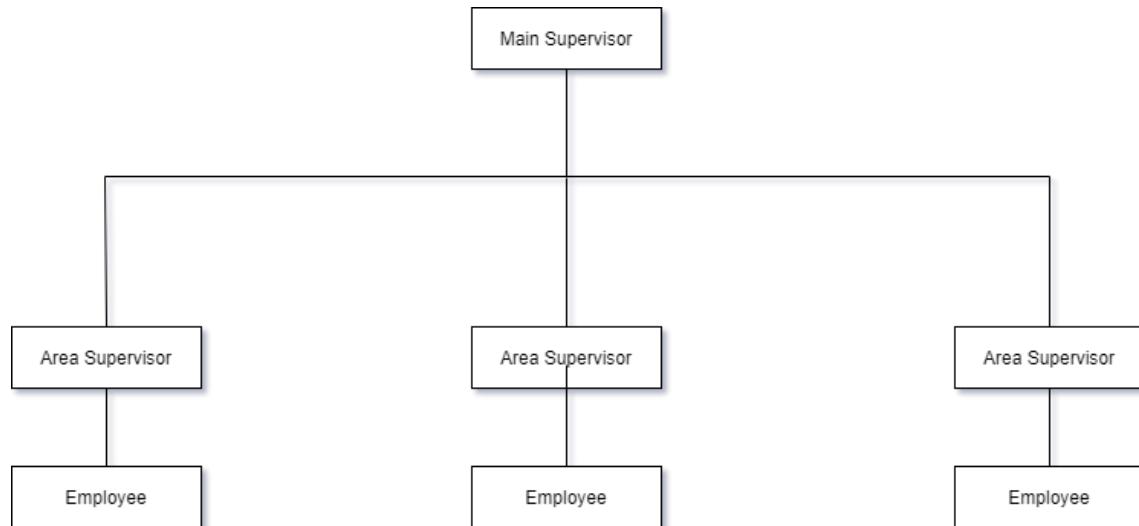
#### **Introduction**

Early Garbage detection systems are not very efficient. The sensors were placed inside the lid of trash bins which measures the levels of garbage by calculating the difference between the objects & the sensor but the sensors didn't perform very well because whenever any piece of garbage comes over the sensor it shows the trash bin full even it's not full. The previous work on problem statement is the detection of garbage through cameras using deep learning model but the model only detects the single/classified garbage items like plastic bottles, wrappers as garbage & there's no system that detects piles of garbage. Another work on this problem statement was to detect the classified pieces of garbage & robot collects it & through it into its back side where trash bin is placed the system is good but very costly to achieve at this stage

#### **Existing System**

Piles of garbage is one of the major issues in smart cities. There's a dire need of system for these types of problems. There are few systems that already developed to solve this type of problems but for specific or single garbage piece as we mentioned earlier. The existing system was developed using sensors on lid of dustbins where the sensor was placed & then generates an alert whenever the dustbins are fully completed. Another existing system was based on image processing which takes input of single waste materials & label as garbage. There's a need/shortage of system that detects piles of garbage & identify recyclables through cameras.

## Organizational Chart



*Figure 3 : Organizational Chart*

### Scope of the System

This Garbage detection system detects the piles of garbage through deep learning model based on computer vision technique. This system use camera to detect piles of garbage. After detection the real time alert will be send to android app used by municipality comities members.

**Table 9 : System Scope**

<b>For</b>	Municipal committee members
<b>What</b>	Garbage detection Send alert message
<b>The</b>	Smart garbage management for sustainable city life
<b>Is</b>	Hybrid App (Android Application & Web Application)
<b>That</b>	Facilitates the authorities to detect the piles of garbage & identify recyclables in real time with exact location

### Summary of Requirements: (Initial Requirements)

#### Functional Requirements

- The system shall detect piles of garbage.
- The system shall label the piles of garbage.
- The system shall label the recyclables & non-recyclables.

- The system shall make user to receive notification.
- The system shall authenticate the user.
- The system shall send alert.
- The system shall show live view of piles of garbage.
- The system shall show work completed status.

### **Non-Functional Requirements**

- **Availability**

By providing all the major functionalities that a system requires with ease. We take feedbacks at the end of month from employee and perform all the updating that they demand.

- **Reliability**

The response time of our system is about 2s to 3s during login. The system will take about 10s for the detection of the coming detected image of the garbage.

- **User Friendly**

This system is easy to use, there is no special type of training required for handling the system. All the things are easily understandable to any type of user.

#### **2.1.2. Identifying External Entities**

The External entities are classified from the given abstract of the project. This identification is done after two phases. We use our given abstract given above at the start of the document for the identification of the external entities by using two phases mentioned below.

##### **a. Over Specify Entities from Abstract**

- TMA (Tehsil Municipal Committee) Employees
- Camera Operators
- Supervisors
- Area Supervisors

##### **b. Perform Refinement**

- **User:**

- Users shall register or login with android application.
- User shall get the coordinates where piles of garbage detected.
- User shall receive alerts for tasks.

### 2.1.3. Capture "shall" Statements

Identify "shall" statements, as they would be all functional requirements.

**Table 10 : Capture Shall Statements**

Para#	Initial requirements
R1	The system Shall allow user to get login
R2	The system Shall validate email address
R3	The system Shall validate password
R4	The system Shall authenticate the user
R5	The system Shall Allow user to get registered
R6	The system Shall validate user name
R7	The system Shall update profile
R8	The system Shall detect garbage.
R9	The system Shall label the piles of garbage.
R10	The system Shall show exact garbage location
R11	The system Shall identify recyclables.
R12	The system Shall send an alert on app
R13	The system Shall receive an alert on application
R14	The system Shall show work completed status
R15	The system Shall check maximum number alerts of specific employee
R16	The system Shall check minimum number alerts of employee
R17	The system Shall show the status clear list
R18	The system Shall send the clear status to admin
R19	The system Shall announced employee of month
R20	The system Shall employee of the year
R21	The system Shall allow supervisor to see user details
R22	The system Shall allow main supervisor to add Area Supervisor.
R23	The system Shall allow main supervisor to remove Area Supervisor.
R24	The system Shall allow main supervisor to update Area Supervisor.
R25	The system Shall allow main supervisor to view Area Supervisor.
R26	The system Shall allow area supervisor to add TMA employee
R27	The system Shall allow area supervisor to remove TMA employee
R28	The system Shall allow area supervisor to update TMA employee
R29	The system Shall allow area supervisor to view TMA employee
R30	The system Shall allow users to get logout

#### 2.1.4. Allocate Requirements

Allocate the requirements in the use cases.

**Table 11 : Allocate Requirements**

Para#	Initial requirements	Use case ID	Use Case Name
R1	The system Shall allow user to get login	UC_1	UC_User_Login
R2	The system Shall validate email address	UC_2	UC_Validate_email
R3	The system Shall validate password	UC_3	UC_Validate_password
R4	The system Shall authenticate the user	UC_4	UC_Authenticate_user
R5	The system shall allow user to registered	UC_5	UC_Registered_user
R6	The system Shall validate user name	UC_6	UC_Validate_username
R7	The system Shall update profile	UC_7	UC_Completed_status
R8	Shall detect garbage through camera	UC_8	UC_Detect_Garbage_Cam
R9	The system Shall label the piles of garbage	UC_9	UC_Piles_Garbage
R10	The system Shall show exact location	UC_10	UC_Location_Corrdinates
R11	Shall identify recyclables through camera	UC_11	UC_Recycleables_Garbage_Cam
R12	Shall send an alert on app	UC_12	UC_Send_Alert_App
R13	The system Shall receive an alert	UC_13	UC_Receive_Alert
R14	The system Shall Update completed status	UC_14	UC_Completed_status
R15	The system Shall check maximum number alerts of specific employee	UC_15	UC_Max_Alerts
R16	The system Shall check minimum number alerts of employee	UC_16	UC_Min_Alerts

R17	The system Shall show the status clear list	UC_17	UC_Clr_Status_List
R18	The system Shall Send clear status to admin	UC_18	UC_Clr_Status_update
R19	The system Shall announced employee of month	UC_19	UC_Employee_month
R20	The system Shall announce employee of the year	UC_20	UC_Employee_Year
R21	The system Shall allow Area Supervisor to see user details	UC_21	UC_See_User
R22	The system Shall allow main supervisor to add Area Supervisor	UC_22	UC_Add_Area Supervisor
R23	The system Shall allow main supervisor to remove Area Supervisor	UC_23	UC_Remove_Area Supervisor
R24	The system Shall allow main supervisor to update Area Supervisor	UC_24	UC_Update_Area Supervisor
R25	The system Shall allow main supervisor to view Area Supervisor	UC_25	UC_View_Area Supervisor
R26	The system Shall allow area supervisor to add TMA employee	UC_26	UC_Add_TMA_Employee
R27	The system Shall allow area supervisor to remove TMA employee	UC_27	UC_Remove_TMA_Employee
R28	The system Shall allow area supervisor to update TMA employee	UC_28	UC_Update_TMA_Employee
R29	The system Shall allow area supervisor to view TMA employee	UC_29	UC_View_TMA_Employee
R30	The system Shall allow users to get logout	UC_30	UC_User_Logout

### 2.1.5. Prioritize Requirements

Requirements must be prioritized as this will help achieve tasks easily. Rank them as “highest, medium, and lowest”.

**Table 12 : Prioritize Requirements**

Para#	Rank	Initial requirements	Use case ID	Use Case Name
R1	Medium	The system Shall allow user to get login	UC_1	UC_User_Login
R2	HIGH	The system Shall validate email address	UC_2	UC_Validate_email
R3	HIGH	The system Shall validate password	UC_3	UC_Validate_password
R4	HIGH	The system Shall authenticate the user	UC_4	UC_Authenticate_user
R5	HIGH	The system shall allow user to registered	UC_5	UC_Registered_user
R6	Medium	The system Shall validate user name	UC_6	UC_Validate_username
R7	Medium	The system Shall update profile	UC_7	UC_Update_profile
R8	HIGH	Shall detect garbage through camera	UC_8	UC_Detect_Garbage_Cam
R9	HIGH	The system Shall label the piles of garbage	UC_9	UC_Piles_Garbage
R10	HIGH	The system Shall show exact location	UC_10	UC_Location_Corrdinates
R11	HIGH	Shall identify recyclables through camera	UC_11	UC_Recycleable_Garbage_Cam
R12	HIGH	Shall send an alert on app	UC_12	UC_Send_Alert_App
R13	HIGH	The system Shall receive an alert	UC_13	UC_Receive_Alert
R14	Medium	The system Shall Update completed status	UC_14	UC_Completed_status
R15	Medium	The system Shall check maximum number alerts of specific employee	UC_15	UC_Max_Alerts
R16	HIGH	The system Shall check minimum number alerts of employee	UC_16	UC_Min_Alerts
R17	Medium	The system Shall show the status clear list	UC_17	UC_Clr_Status_List
R18	HIGH	The system Shall update clear status to admin	UC_18	UC_Clr_Status_update

R19	HIGH	The system Shall announced employee of month	UC_19	UC_Employee_month
R20	HIGH	The system Shall announce employee of the year	UC_20	UC_Employee_Year
R21	HIGH	The system Shall allow admin to see user details	UC_21	UC_See_User
R22	HIGH	The system Shall allow main supervisor to add Area Supervisor	UC_22	UC_Add_Area Supervisor
R23	HIGH	The system Shall allow main supervisor to remove area supervisor	UC_23	UC_Remove_Area Supervisor
R24	HIGH	The system Shall allow main supervisor to update area supervisor	UC_24	UC_Update_Area Supervisor
R25	HIGH	The system Shall allow main supervisor to view area supervisor	UC_25	UC_View_Area Supervisor
R26	HIGH	The system Shall allow area supervisor to add TMA employee	UC_26	UC_Add_TMA_Employee
R27	HIGH	The system Shall allow area supervisor to Remove TMA employee	UC_27	UC_Remove_TMA_Employee
R28	HIGH	The system Shall allow area supervisor to update TMA employee	UC_28	UC_Update_TMA_Employee
R29	HIGH	The system Shall allow area supervisor to view TMA employee	UC_29	UC_View_TMA_Employee
R30	HIGH	The system Shall allow user to get logout	UC_30	UC_User_Logout

### 2.1.6. Requirements Trace-ability Matrix

The requirements trace-ability matrix is a table used to trace project life cycle activities and work products to the project requirements. The matrix establishes a thread that traces requirements from identification through implementation.

**Table 13 : Requirements Trace-ability Matrix**

<b>Para#</b>	<b>System Specification Text</b>	<b>Build</b>	<b>Use Case Name</b>	<b>Category</b>
R1	The system Shall allow user to get login	B1	UC_User_Login	Business
R2	The system Shall validate email address	B2	UC_Validate_email	Business
R3	The system Shall validate password	B3	UC_Validate_password	Business
R4	The system Shall authenticate the user	B4	UC_Authenticate_user	Business
R5	The system shall allow user to registered	B5	UC_Registered_user	Business
R6	The system Shall validate user name	B6	UC_Validate_username	Business
R7	The system Shall update profile	B7	UC_Update_profile	Business
R8	Shall detect garbage through camera	B8	UC_Detect_Garbage_Cam	Business
R9	The system Shall label the piles of garbage	B9	UC_Piles_Garbage	Business
R10	The system Shall show exact location	B10	UC_Location_Cordinates	Business
R11	Shall identify recyclables through camera	B11	UC_Recycleable_Garbage_Cam	Business
R12	Shall send an alert on app	B12	UC_Send_Alert_App	Business
R13	The system Shall receive an alert	B13	UC_Receive_Alert	Business
R14	The system Shall Update completed status	B14	UC_Completed_status	Business
R15	The system Shall check maximum number alerts of specific employee	B15	UC_Max_Alerts	Business
R16	The system Shall check minimum number alerts of employee	B16	UC_Min_Alerts	Business
R17	The system Shall show the status clear list	B17	UC_Clr_Status_List	Business

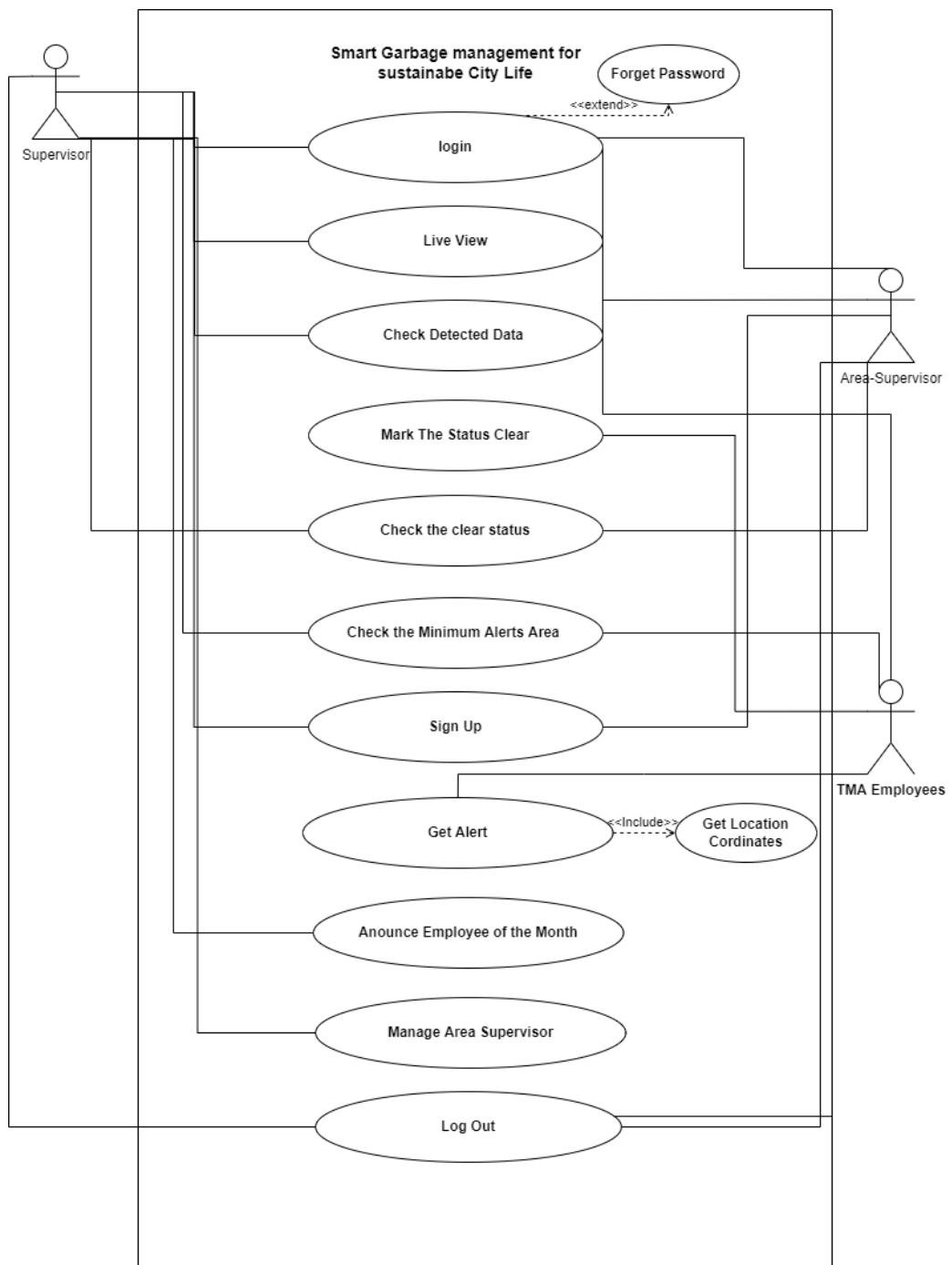
R18	The system Shall update clear status to admin	B18	UC_Clr_Status_update	Business
R19	The system Shall announced employee of month	B19	UC_Employee_month	Business
R20	The system Shall announce employee of the year	B20	UC_Employee_Year	Business
R21	The system Shall allow admin to see user details	B21	UC_See_User	Business
R22	The system Shall allow main supervisor to add Area Supervisor	B22	UC_Add_Area_Supervisor	Business
R23	The system Shall allow main supervisor to remove area supervisor	B23	UC_Remove_Area_Supervisor	Business
R24	The system Shall allow main supervisor to update area supervisor	B24	UC_Update_Area_Supervisor	Business
R25	The system Shall allow main supervisor to view area supervisor	B25	UC_View_Area_Supervisor	Business
R26	The system Shall allow area supervisor to add TMA employee	B26	UC_Add_TMA_Employee	Business
R27	The system Shall allow area supervisor to remove TMA employee	B27	UC_Remove_TMA_Employee	Business
R28	The system Shall allow area supervisor to update TMA employee	B28	UC_Update_TMA_Employee	Business
R29	The system Shall allow area supervisor to view TMA employee	B29	UC_View_TMA_Employee	Business
R30	The Shall Allow user to get Logout.	B30	UC_User_Logout	Business

## 2.2. Use Case Diagrams

### 2.2.1. High Level Use Case Diagram

High level visual representations of user requirements and they can act into the use cases. The use cases are the ovals with which the actor interacts. The box is separating some top-

level internals of the system from the external actors. High level of our project is shown in figure below.

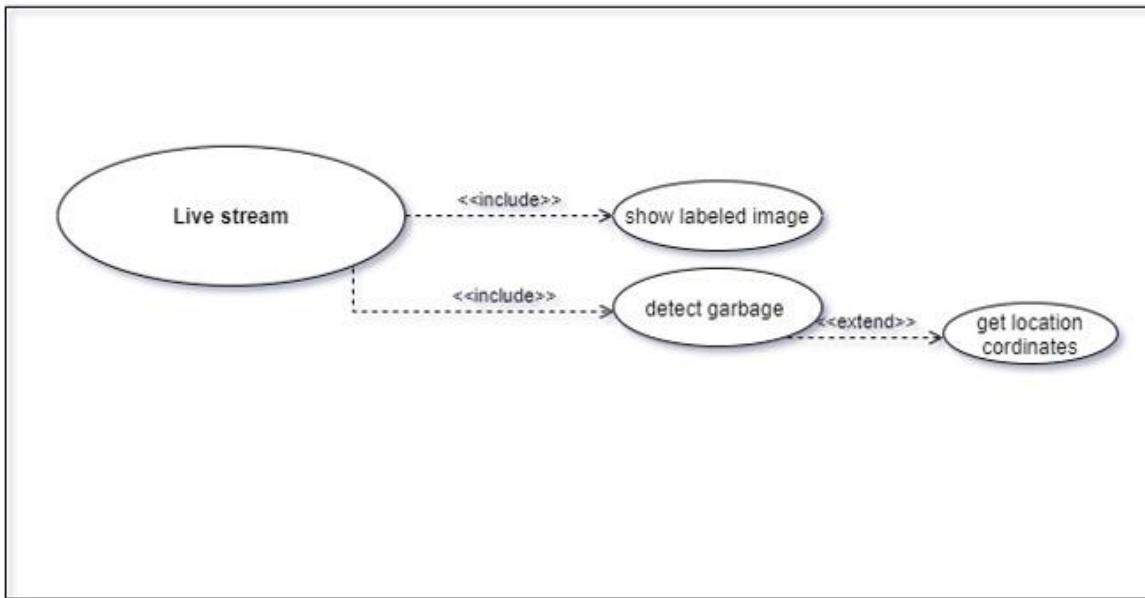


**Figure 4 : High Level Use Case**

### 2.2.2. Analysis Level Usecase Diagram:

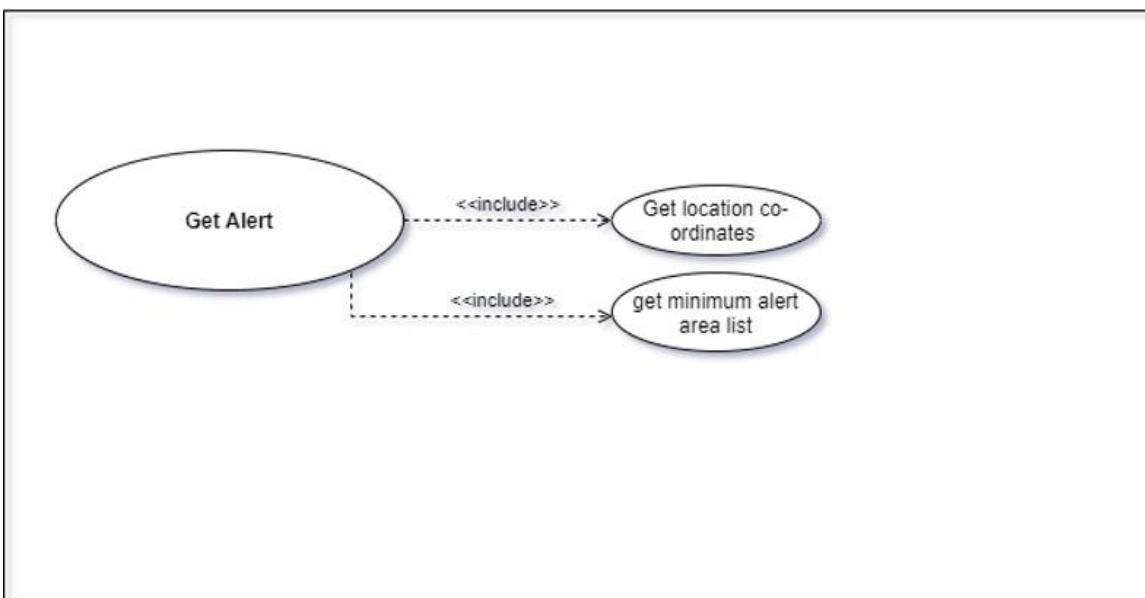
A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

#### 2.2.2.1 Live View Garbage



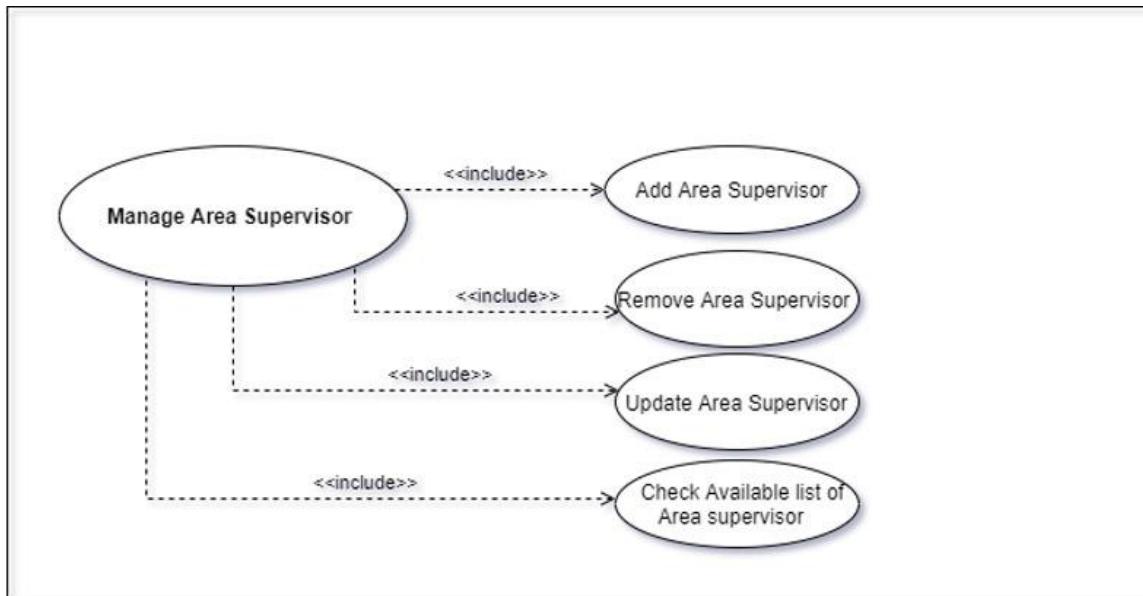
*Figure 5 : Live View Garbage*

#### 2.2.2.2 Get Alert



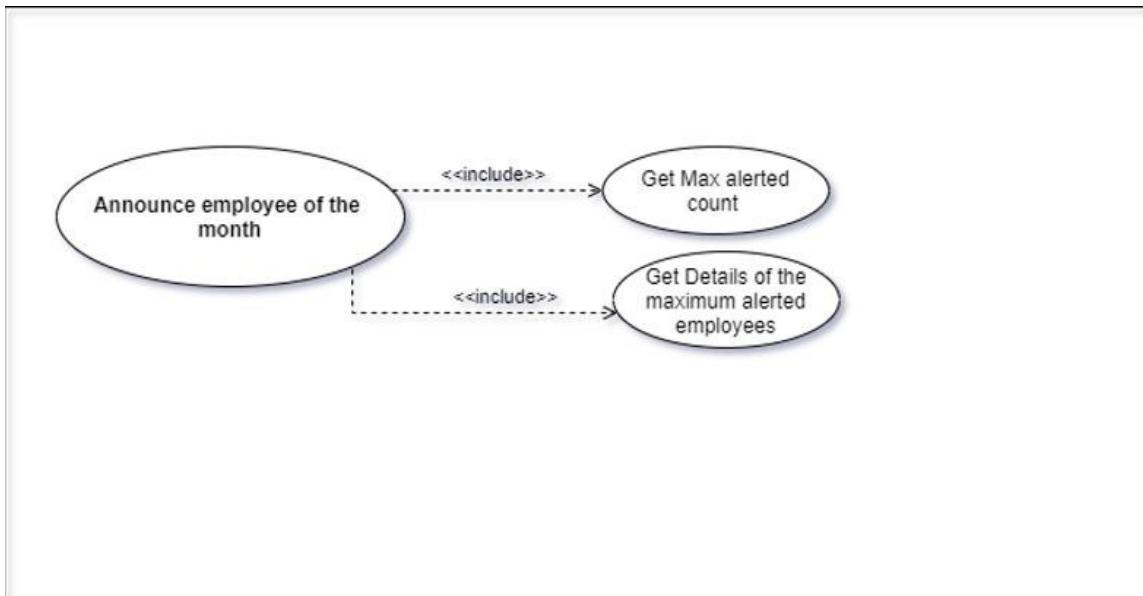
*Figure 6 : Get Alert*

### 2.2.2.3 Manage Area Supervisor



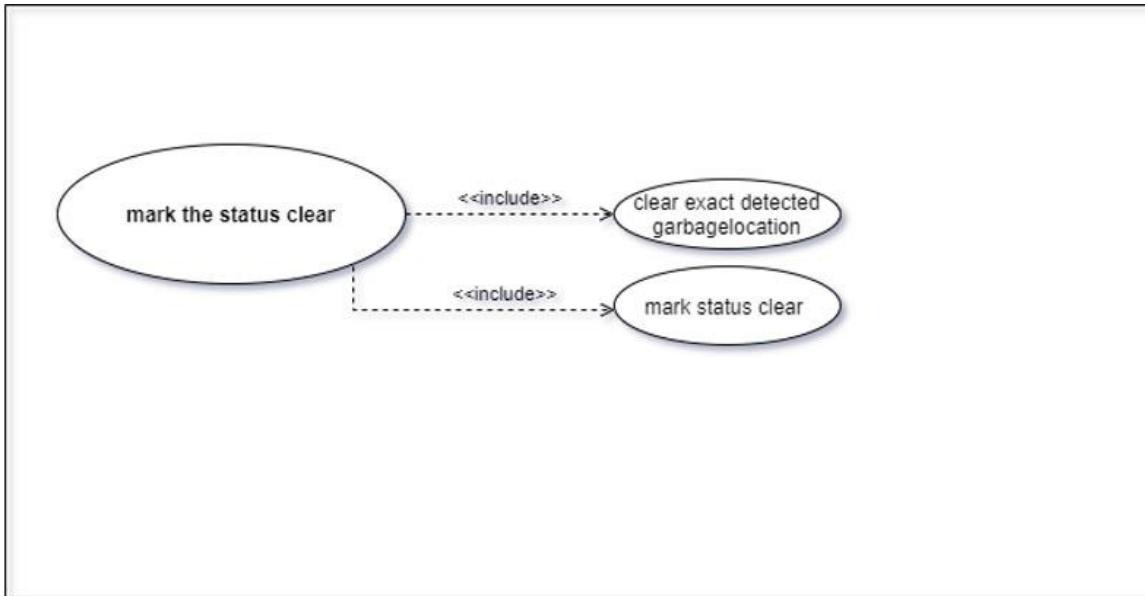
*Figure 7 : Manage Area Supervisor*

### 2.2.2.4 Employee of the year



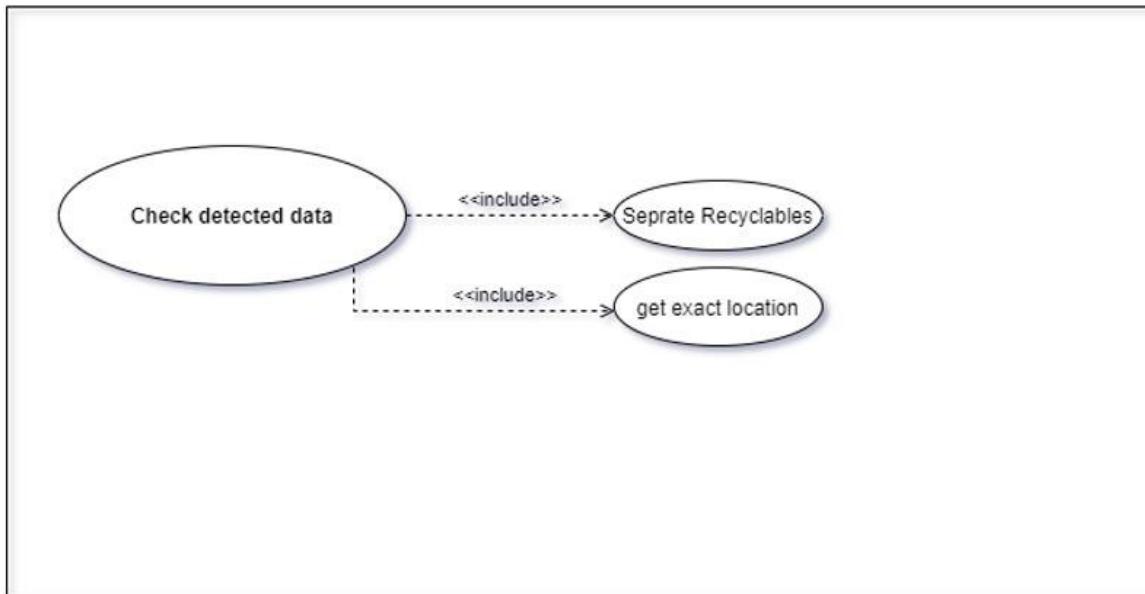
*Figure 8 : Employee of the year*

#### 2.2.2.5 Status Clear



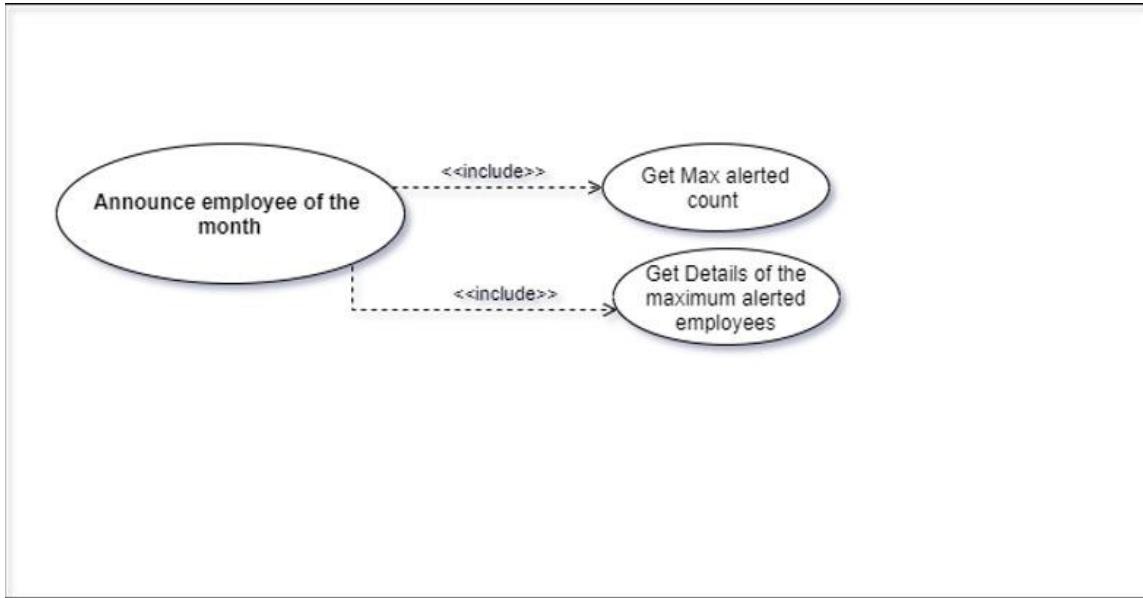
*Figure 9 : Status Clear*

#### 2.2.2.6 Garbage Detection



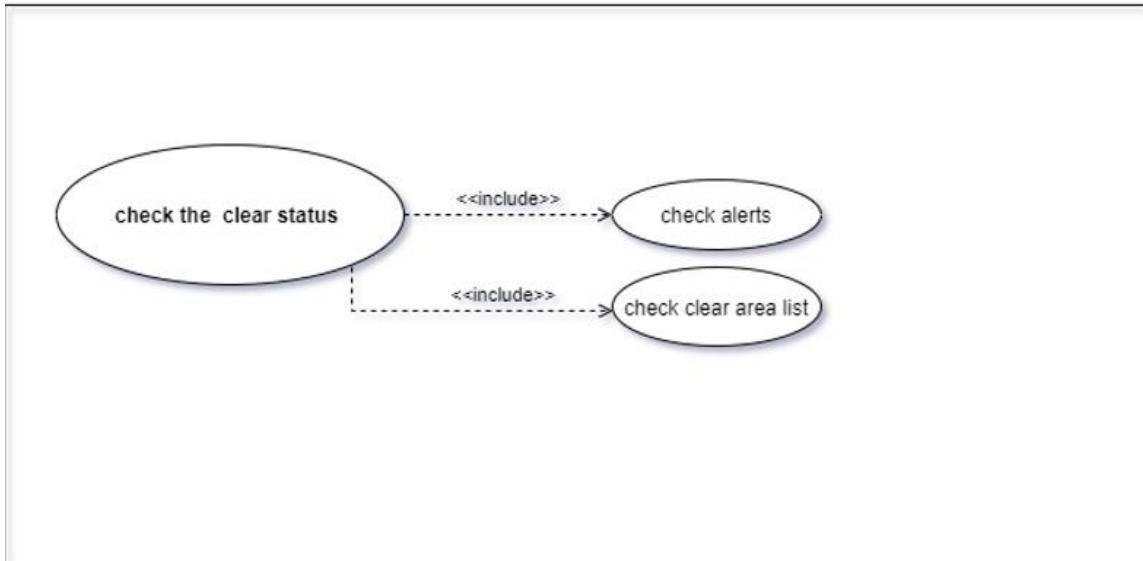
*Figure 10 : Garbage Detection*

#### 2.2.2.7 Employee of the month



*Figure 11 : Employee of the month*

#### 2.2.2.8 Check Clear Status



*Figure 12 : Check Clear Status*

### 2.2.2.9 Register

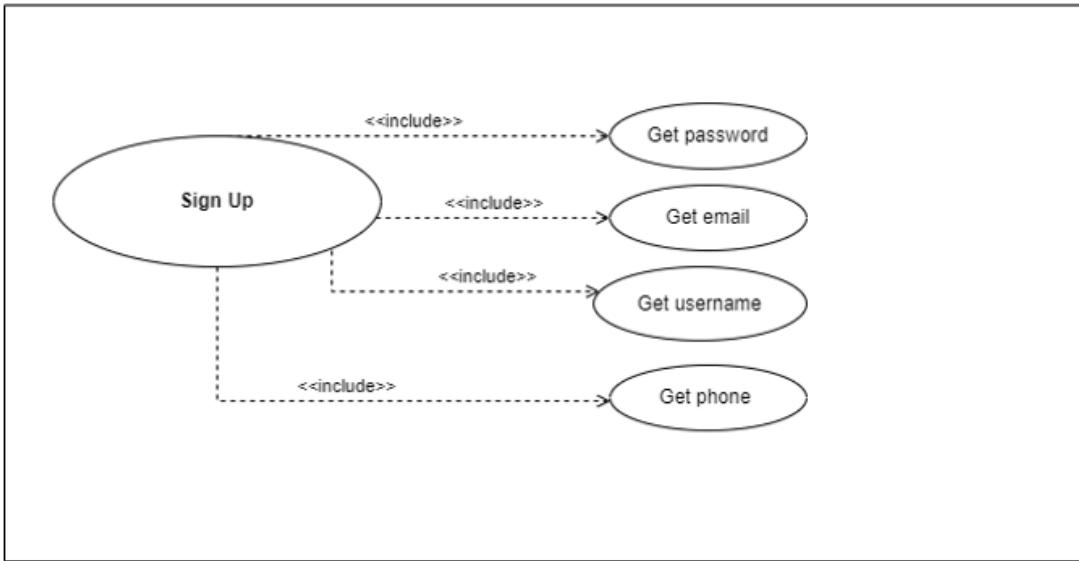


Figure 13 : Register

### 2.2.3. Usecase Description

- UC\_1 describes what is considered the happy path the functionality that occurs when use case to login & executes without errors. It can include critical conditions when the execution of detection occurs.

Table 14 : UC-1 User Login  
Use Case Description

Use Case Description	
Use Case ID:	UC_1
Use Case Name:	UC_User_Login
Goal/Purpose:	Fill the Login Form & allow user to get access to the admin panel/account
Actors:	Admin
Pre-Conditions	Home Screen
Post-Conditions:	Get Access to The Admin Panel
Basic Flow:	By Filling Valid Credentials
Alternate Flow(s):	Register as an Admin

- **UC\_2** describes what is considered the happy path the functionality that occurs when use case to Validate Email Address & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 15 : UC-2 Validate Email**

Use Case Description	
<b>Use Case ID:</b>	UC_2
<b>Use Case Name:</b>	UC_Validate_email
<b>Goal/Purpose:</b>	To Check Entered email address is valid or not
<b>Actors:</b>	System Itself
<b>Pre-Conditions</b>	The Email Address field should be filled
<b>Post-Conditions:</b>	Show Error If Not Validate
<b>Basic Flow:</b>	User Should Fill the Login Form & Click on the Submit Button
<b>Alternate Flow(s):</b>	None

- **UC\_3** describes what is considered the happy path the functionality that occurs when use case to Validate Password & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 16 : UC-3 Validate Password**

Use Case Description	
<b>Use Case ID:</b>	UC_3
<b>Use Case Name:</b>	UC_Validate_password
<b>Goal/Purpose:</b>	To Check Entered Password is valid or not
<b>Actors:</b>	System Itself
<b>Pre-Conditions</b>	The Password field should be filled
<b>Post-Conditions:</b>	Show Error If Not Validate E.g., Weak Password
<b>Basic Flow:</b>	User Should Fill the Login Form & Click on the Submit Button
<b>Alternate Flow(s):</b>	None

- **UC\_4** describes what is considered the happy path the functionality that occurs when use case to Authenticate user & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 17 : UC-4 Authenticate User**

Use Case Description	
<b>Use Case ID:</b>	UC_4
<b>Use Case Name:</b>	UC_Authenticate_user
<b>Goal/Purpose:</b>	To Validate the User That is It Available
<b>Actors:</b>	System Itself
<b>Pre-Conditions</b>	The User Must Should Registered
<b>Post-Conditions:</b>	Open User Page If Valid Otherwise Show Error Message
<b>Basic Flow:</b>	User Should Fill the Login Form & Click on the Submit Button

- **UC\_5** describes what is considered the happy path the functionality that occurs when use case to Register & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 18 : UC-5 Register User**

Use Case Description	
<b>Use Case ID:</b>	UC_5
<b>Use Case Name:</b>	UC_Registered_user
<b>Goal/Purpose:</b>	To Register as a User
<b>Actors:</b>	User/Staff
<b>Pre-Conditions</b>	Registration Screen
<b>Post-Conditions:</b>	Registers
<b>Basic Flow:</b>	User Should open the android app/web & Fill the Registration form to get register as user
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet error

- **UC\_6** describes what is considered the happy path the functionality that occurs when use case to Authenticate user & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 19 : UC-6 Validate User**  
**Use Case Description**

<b>Use Case ID:</b>	UC_6
<b>Use Case Name:</b>	UC_Validate_username
<b>Goal/Purpose:</b>	To Check that Username is Already not Available
<b>Actors:</b>	User/Staff
<b>Pre-Conditions</b>	User Shall pick the username
<b>Post-Conditions:</b>	Register with Username if Already not Available
<b>Basic Flow:</b>	Fill the registration form & submit the form
<b>Alternate Flow(s):</b>	None

- **UC\_7** describes what is considered the happy path the functionality that occurs when use case to Update Profile & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 20 : UC-7 Update Profile**  
**Use Case Description**

<b>Use Case ID:</b>	UC_7
<b>Use Case Name:</b>	UC_Update_profile
<b>Goal/Purpose:</b>	User can Update his/her Profile
<b>Actors:</b>	User/Staff
<b>Pre-Conditions</b>	User Should Login to the System
<b>Post-Conditions:</b>	Update the Profile by Changing Details
<b>Basic Flow:</b>	Login to the System & Go to the Profile & Update the details the user desire
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet error, Not Login to the System

- **UC\_8** describes what is considered the happy path the functionality that occurs when use case Detect piles of garbage through camera executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 21 : UC-8 Detect Garbage**

Use Case Description	
<b>Use Case ID:</b>	UC_8
<b>Use Case Name:</b>	UC_Detect_Garbage_Cam
<b>Goal/Purpose:</b>	To Detect Garbage by using in order to remove pollution from our society & make city neat & clean
<b>Actors:</b>	Drone or Camera Operator/User
<b>Pre-Conditions</b>	Live screen
<b>Post-Conditions:</b>	Piles of Garbage Detected
<b>Basic Flow:</b>	Shows live video in order to detect garbage.
<b>Alternate Flow(s):</b>	Detect Garbage Using Ultrasonic or IR Sensor
<b>Use Case Name:</b>	UC_Detect_Garbage_Cam

- **UC\_9** describes what is considered the happy path the functionality that occurs when use case to Label the Piles of Garbage & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 22 : UC-9 Label Piles of Garbage**

Use Case Description	
<b>Use Case ID:</b>	UC_9
<b>Use Case Name:</b>	UC_Piles_Garbage
<b>Goal/Purpose:</b>	Label the Piles of Garbage
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Shall detect piles of garbage
<b>Post-Conditions:</b>	Label the detected image of piles of garbage
<b>Basic Flow:</b>	Login to the System & see detection & see the label image
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error

- **UC\_10** describes what is considered the happy path the functionality that occurs when use case send location coordinates & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 23 : UC-10 Location Coordinates**

Use Case Description	
<b>Use Case ID:</b>	UC_10
<b>Use Case Name:</b>	UC_Location_Corrdinates
<b>Goal/Purpose:</b>	To Know the Exact Location of Piles of Garbage
<b>Actors:</b>	Drone/Camera Operator
<b>Pre-Conditions</b>	Live screen
<b>Basic Flow:</b>	The System Detects the Garbage & Send the Current Location Coordinates to an Android Application
<b>Alternate Flow(s):</b>	None
<b>Use Case Name:</b>	UC_Location_Corrdinates
<b>Goal/Purpose:</b>	To Know the Exact Location of Piles of Garbage

- **UC\_11** describes what is considered the happy path the functionality that occurs when use case identify recyclables through camera executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 24 : UC-11 Identify Recyclables**

Use Case Description	
<b>Use Case ID:</b>	UC_11
<b>Use Case Name:</b>	UC_Recycleable_Garbage_Cam
<b>Goal/Purpose:</b>	To Classify Recyclables products from the piles of garbage through cameras
<b>Actors:</b>	Drone or Camera Operator/User
<b>Pre-Conditions</b>	Live screen
<b>Post-Conditions:</b>	Classify the Plastic Bottles
<b>Basic Flow:</b>	Shows live video in order to detect garbage.
<b>Alternate Flow(s):</b>	To Classify Plastic Bottles Manually During Cleaning

- **UC\_12** describes what is considered the happy path the functionality that occurs when use case send alert to app executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 25 : UC-12 Send Alert**

Use Case Description	
<b>Use Case ID:</b>	UC_12
<b>Use Case Name:</b>	UC_Send_Alert_App
<b>Goal/Purpose:</b>	To Receive an Alert to an android app as notification & on main screen with coordinates in order to take immediate actions.
<b>Actors:</b>	User/Employee/Staff
<b>Pre-Conditions</b>	Live screen
<b>Post-Conditions:</b>	Alert & Pop-up Notification
<b>Basic Flow:</b>	Receive Notification as Well as Alert to android app on main screen
<b>Alternate Flow(s):</b>	Open android application
<b>Use Case Name:</b>	UC_Send_Alert_App

- **UC\_13** describes what is considered the happy path the functionality that occurs when use case to Receive Alert & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 26 : UC-13 Receive Alert**

Use Case Description	
<b>Use Case ID:</b>	UC_13
<b>Use Case Name:</b>	UC_Receive_Alert
<b>Goal/Purpose:</b>	Generates Alert to Admin Panel if Garbage Detected
<b>Actors:</b>	User/Staff
<b>Pre-Conditions</b>	Garbage Should be Detected
<b>Post-Conditions:</b>	Show an Alert or pop-up Message on Admin Panel
<b>Basic Flow:</b>	Detects Garbage & Show Alert to admin panel
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error, Username Already Available

- **UC\_14** describes what is considered the happy path the functionality that occurs when use case to Authenticate user & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 27 : UC-14 Completed Status**

Use Case Description	
<b>Use Case ID:</b>	UC_14
<b>Use Case Name:</b>	UC_Completed_status
<b>Goal/Purpose:</b>	User/Staff Update the Work Status
<b>Actors:</b>	User/Staff
<b>Pre-Conditions</b>	User Should Login to the System & get Task
<b>Post-Conditions:</b>	Update The task to Completed
<b>Basic Flow:</b>	Get Task Alert. After Completing the Task Update the Status by tapping on Task done.
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error

- **UC\_15** describes what is considered the happy path the functionality that occurs when use case to Know which area gets maximum numbers of task alerts. & Executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 28 : UC-15 Maximum Alerts**

Use Case Description	
<b>Use Case ID:</b>	UC_15
<b>Use Case Name:</b>	UC_Max_Alerts
<b>Goal/Purpose:</b>	To Know which area gets maximum numbers of task alerts.
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Admin should be login to admin panel
<b>Post-Conditions:</b>	Max no of task in specific area
<b>Basic Flow:</b>	Check max no task a lot to specific area through admin panel
<b>Alternate Flow(s):</b>	Calculate Minimum
<b>Exception Flow(s):</b>	Internet Error, Database Error

- **UC\_16** describes what is considered the happy path the functionality that occurs when use case to Know which area gets minimum numbers of task alerts & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 29 : UC-16 Minimum Alerts**

Use Case Description	
<b>Use Case ID:</b>	UC_16
<b>Use Case Name:</b>	UC_Min_Alerts
<b>Goal/Purpose:</b>	To Know which area gets minimum numbers of task alerts.
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Admin should be login to admin panel
<b>Post-Conditions:</b>	Max no of task in specific area
<b>Basic Flow:</b>	Check min no task a lot to specific area through admin panel
<b>Alternate Flow(s):</b>	Calculate Maximum
<b>Exception Flow(s):</b>	Internet Error, Database Error

- **UC\_17** describes what is considered the happy path the functionality that occurs when use case to Show the Status list of clear status & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 30 : UC-17 Status Clear List**

Use Case Description	
<b>Use Case ID:</b>	UC_17
<b>Use Case Name:</b>	UC_Clr_Status_List
<b>Goal/Purpose:</b>	Show the Status list of clear status
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Admin should be login to admin panel & open specific page
<b>Post-Conditions:</b>	Status Clear list Table
<b>Basic Flow:</b>	Login to the Admin Panel & see the details list
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error, Database Error

- **UC\_18** describes what is considered the happy path the functionality that occurs when use case to Update Status & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 31 : UC-18 Status Update**

Use Case Description	
<b>Use Case ID:</b>	UC_18
<b>Use Case Name:</b>	UC_Clr_Status_update
<b>Goal/Purpose:</b>	After completing work & clearing status the status on admin panel should be update
<b>Actors:</b>	User
<b>Pre-Conditions</b>	User should click the update status button
<b>Post-Conditions:</b>	Task done on admin panel
<b>Basic Flow:</b>	Complete the Task & click the update status button
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error

- **UC\_19** describes what is considered the happy path the functionality that occurs when use case to announce the Employee of the Month at the end of Month & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 32 : UC-19 Employee of Month**

Use Case Description	
<b>Use Case ID:</b>	UC_19
<b>Use Case Name:</b>	UC_Employee_month
<b>Goal/Purpose:</b>	Announce the Employee of the Month at the end of Month
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Check the performance according to task a lot
<b>Post-Conditions:</b>	Best Employee of Month
<b>Basic Flow:</b>	Check the No of tasks a lot & announce Employee of the month
<b>Alternate Flow(s):</b>	Check the no completed task or by attendance
<b>Exception Flow(s):</b>	Database error

- **UC\_20** describes what is considered the happy path the functionality that occurs when use case to announce the Employee of the Year at the end of Year & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 33 : UC-20 Employee of Year**  
**Use Case Description**

<b>Use Case Description</b>	
<b>Use Case ID:</b>	UC_20
<b>Use Case Name:</b>	UC_Employee_Year
<b>Goal/Purpose:</b>	Announce the Employee of the Year at the end of Year
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Checking the Minimum Number of Tasks, A lot to the Employee
<b>Post-Conditions:</b>	Announce Employee of the year
<b>Basic Flow:</b>	Check the no completed task or by attendance
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Database error

- **UC\_21** describes what is considered the happy path the functionality that occurs when use case to See User Details & executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 34 : UC-21 User Details**  
**Use Case Description**

<b>Use Case Description</b>	
<b>Use Case ID:</b>	UC_21
<b>Use Case Name:</b>	UC_See_User
<b>Goal/Purpose:</b>	To See User Details
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Login into Web Panel
<b>Post-Conditions:</b>	Logout from Admin Panel
<b>Basic Flow:</b>	Click on Show User Detail
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error

- **UC\_22** describes what is considered the happy path the functionality that occurs when use case to Allow Supervisor to Add Area Supervisor executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 35 : UC-22 Add Area Supervisor**

Use Case Description	
<b>Use Case ID:</b>	UC_22
<b>Use Case Name:</b>	UC_Add_Area_Supervisor
<b>Goal/Purpose:</b>	To Add New Admins in the System
<b>Actors:</b>	Area Supervisor
<b>Pre-Conditions</b>	Area Supervisor Should be Login
<b>Post-Conditions:</b>	New Area Supervisor Add
<b>Basic Flow:</b>	The Supervisor Assign New Area Supervisor in the system by submitting form
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error, Limit Exceed

- **UC\_23** describes what is considered the happy path the functionality that occurs when use case to Allow Supervisor to Remove Area Supervisor executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 36 : UC-23 Remove Area Supervisor**

Use Case Description	
<b>Use Case ID:</b>	UC_23
<b>Use Case Name:</b>	UC_Remove_Area_Supervisor
<b>Goal/Purpose:</b>	Allow Supervisor to Remove Employee
<b>Actors:</b>	Area Supervisor
<b>Pre-Conditions</b>	Login into Admin Panel
<b>Post-Conditions:</b>	Logout from Admin Panel
<b>Basic Flow:</b>	Click on delete on Admin Panel
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error

- **UC\_24** describes what is considered the happy path the functionality that occurs when use case to Allow Supervisor to Update Area Supervisor executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 37 : UC-24 Update Area Supervisor**

Use Case Description	
<b>Use Case ID:</b>	UC_24
<b>Use Case Name:</b>	UC_Update_Area_Supervisor
<b>Goal/Purpose:</b>	To Update Area Supervisor
<b>Actors:</b>	Area Supervisor
<b>Pre-Conditions</b>	Supervisor Should be Login
<b>Post-Conditions:</b>	Update Area Supervisor
<b>Basic Flow:</b>	The Admin Update the Area Supervisor Details By selecting specific admin
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error, Limit Exceed

- **UC\_25** describes what is considered the happy path the functionality that occurs when use case to Allow Supervisor to View Area Supervisor executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 38 : UC-25 View Area Supervisor**

Use Case Description	
<b>Use Case ID:</b>	UC_25
<b>Use Case Name:</b>	UC_View_Area_Supervisor
<b>Goal/Purpose:</b>	View the Area Supervisor
<b>Actors:</b>	Area Supervisor
<b>Pre-Conditions</b>	Login to the Admin Panel
<b>Post-Conditions:</b>	View the Area Supervisor Details
<b>Basic Flow:</b>	Select the Specific Area Supervisor & view details
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error, Database Error

- **UC\_26** describes what is considered the happy path the functionality that occurs when use case to Allow Area Supervisor to Add TMA Employees executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 39 : UC-26 Add TMA Employee**

Use Case Description	
<b>Use Case ID:</b>	UC_26
<b>Use Case Name:</b>	UC_Add_TMA_Employee
<b>Goal/Purpose:</b>	Allow Admin to Add TMA Employee
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Login into Admin Panel
<b>Post-Conditions:</b>	Logout from Admin Panel
<b>Basic Flow:</b>	Click on the Add TMA Employee
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error

- **UC\_27** describes what is considered the happy path the functionality that occurs when use case to Allow Area Supervisor to Remove TMA Employees executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 40 : UC-27 Remove TMA Employee**

Use Case Description	
<b>Use Case ID:</b>	UC_27
<b>Use Case Name:</b>	UC_Remove_TMA_Employee
<b>Goal/Purpose:</b>	Allow Admin to Remove TMA Employee
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Login into Admin Panel
<b>Post-Conditions:</b>	Logout from Admin Panel
<b>Basic Flow:</b>	Click on delete on Admin Panel
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error

- **UC\_28** describes what is considered the happy path the functionality that occurs when use case to Allow Area Supervisor to Update TMA Employees executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 41 : UC-28 Update TMA Employee**

Use Case Description	
<b>Use Case ID:</b>	UC_28
<b>Use Case Name:</b>	UC_Update_TMA_Employee
<b>Goal/Purpose:</b>	Allow Admin to Update TMA Employee
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Login into Admin Panel
<b>Post-Conditions:</b>	Logout from Admin Panel
<b>Basic Flow:</b>	Click on Update on Admin Panel
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error

- **UC\_29** describes what is considered the happy path the functionality that occurs when use case to Allow Area Supervisor to View TMA Employees executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 42 : UC-29 View TMA Employee**

Use Case Description	
<b>Use Case ID:</b>	UC_29
<b>Use Case Name:</b>	UC_View_TMA_Employee
<b>Goal/Purpose:</b>	Allow Admin to View TMA Employees
<b>Actors:</b>	Admin
<b>Pre-Conditions</b>	Login into Admin Panel
<b>Post-Conditions:</b>	Logout from Admin Panel
<b>Basic Flow:</b>	Click on View Employee Button
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error

- **UC\_30** describes what is considered the happy path the functionality that occurs when use case to Allow User to Logout executes without errors. It can include critical conditions when the execution of detection occurs.

**Table 43 : UC-30 Logout**

Use Case Description	
<b>Use Case ID:</b>	UC_30
<b>Use Case Name:</b>	UC_User_Logout
<b>Goal/Purpose:</b>	Allow User to Logout
<b>Actors:</b>	Admin, TMA_Employees
<b>Pre-Conditions</b>	Login into Admin Panel
<b>Post-Conditions:</b>	Logout from Admin Panel
<b>Basic Flow:</b>	Click on Logout Button
<b>Alternate Flow(s):</b>	None
<b>Exception Flow(s):</b>	Internet Error

## **Chapter 3: Software Design & Architecture**

---

### 3.1 Introduction

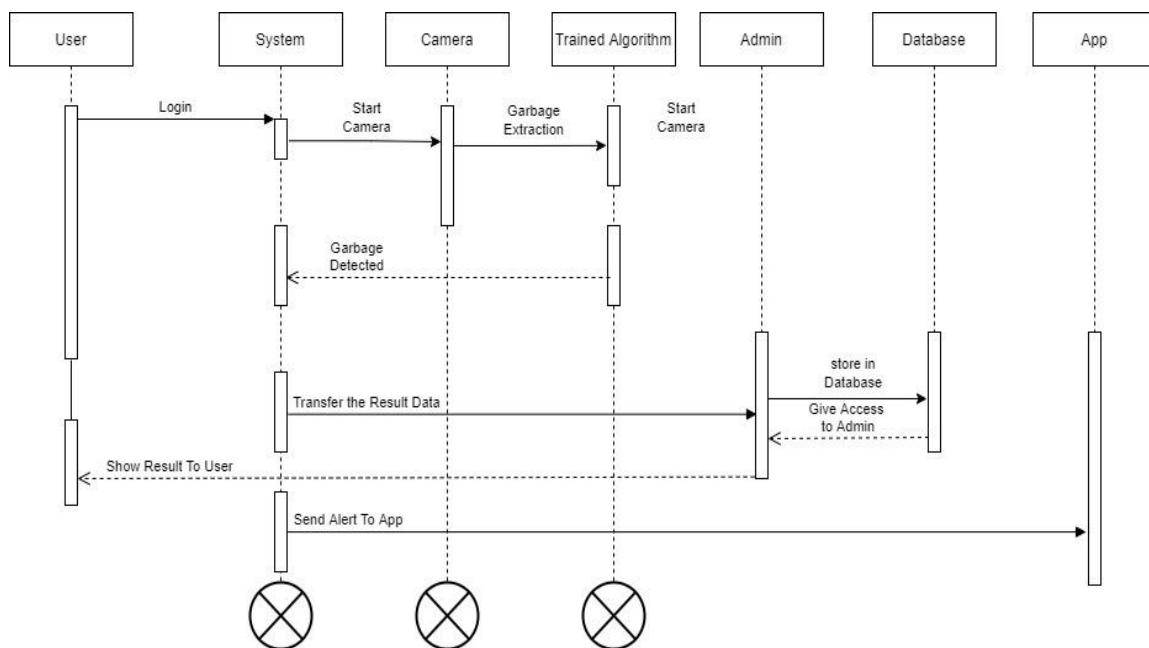
Third deliverable is all about the software design. In the previous deliverable, analysis of the system is completed. So, we understand the current situation of the problem domain. Now we are ready to strive for a solution for the problem domain by using object-oriented approach. Following artifacts must be included in the 3<sup>rd</sup> deliverable.

1. Sequence Diagram
2. Entity Relationship diagram
3. Design Class Diagram
4. State Transition Diagram
5. Data Flow Diagram
6. Domain Model Diagram

Now we discuss these artifacts one by one as follows

### 3.2 Sequence Diagram

The UML system sequence diagram (SSD) illustrates events sequentially input from an external source to the system. The SSD will define the system events and operations. System sequence diagrams are a timeline drawing of an expanded use case. Events are related by time with the top events occurring first. System events are the important items. These are events that cause a system response. As in the purposed system there are two actors Admin & User/Employees. A sequence diagram of our system named as “Smart Garbage Management for Sustainable City life” is shown as follows:



*Figure 14 : Sequence Diagram*

### 3.3. Class Diagram

A class diagram in the (UML) is a type of stagnant structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among those objects.

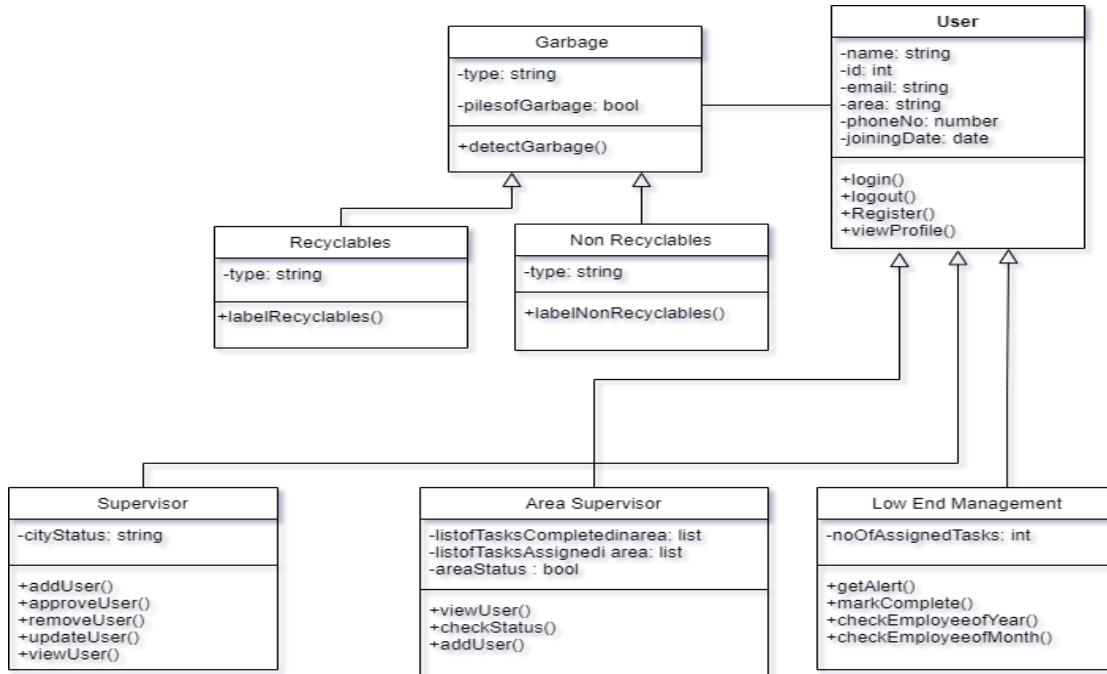


Figure 15 : Class Diagram

### 3.4. State Transition Diagram

State Transition Diagram is developed to represent the behavior of the system under consideration.

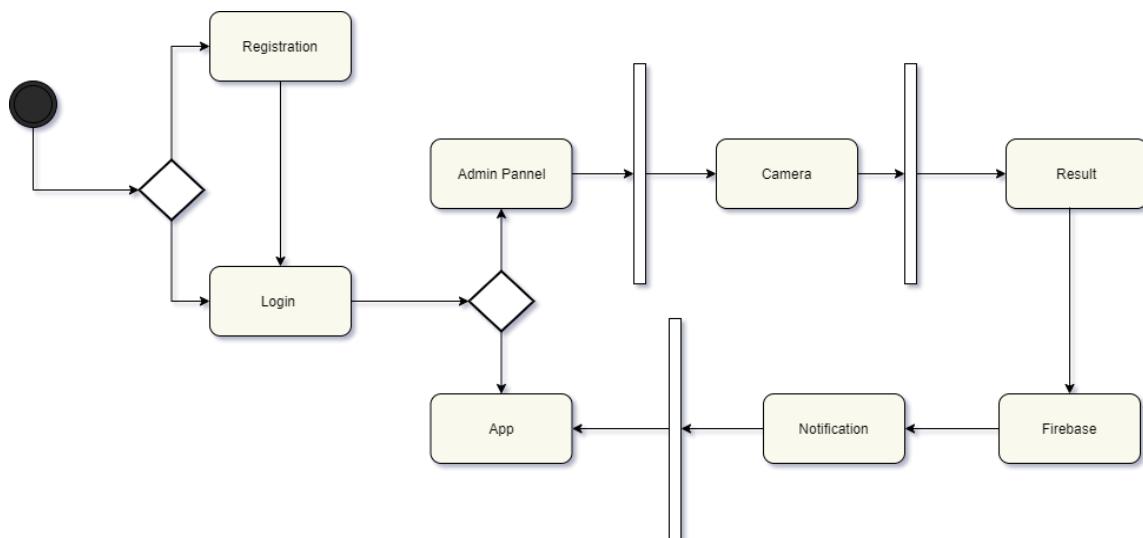


Figure 16 : State Transition Diagram

### 3.5. Entity Relation Diagram

Entity relationship diagram consists of three things

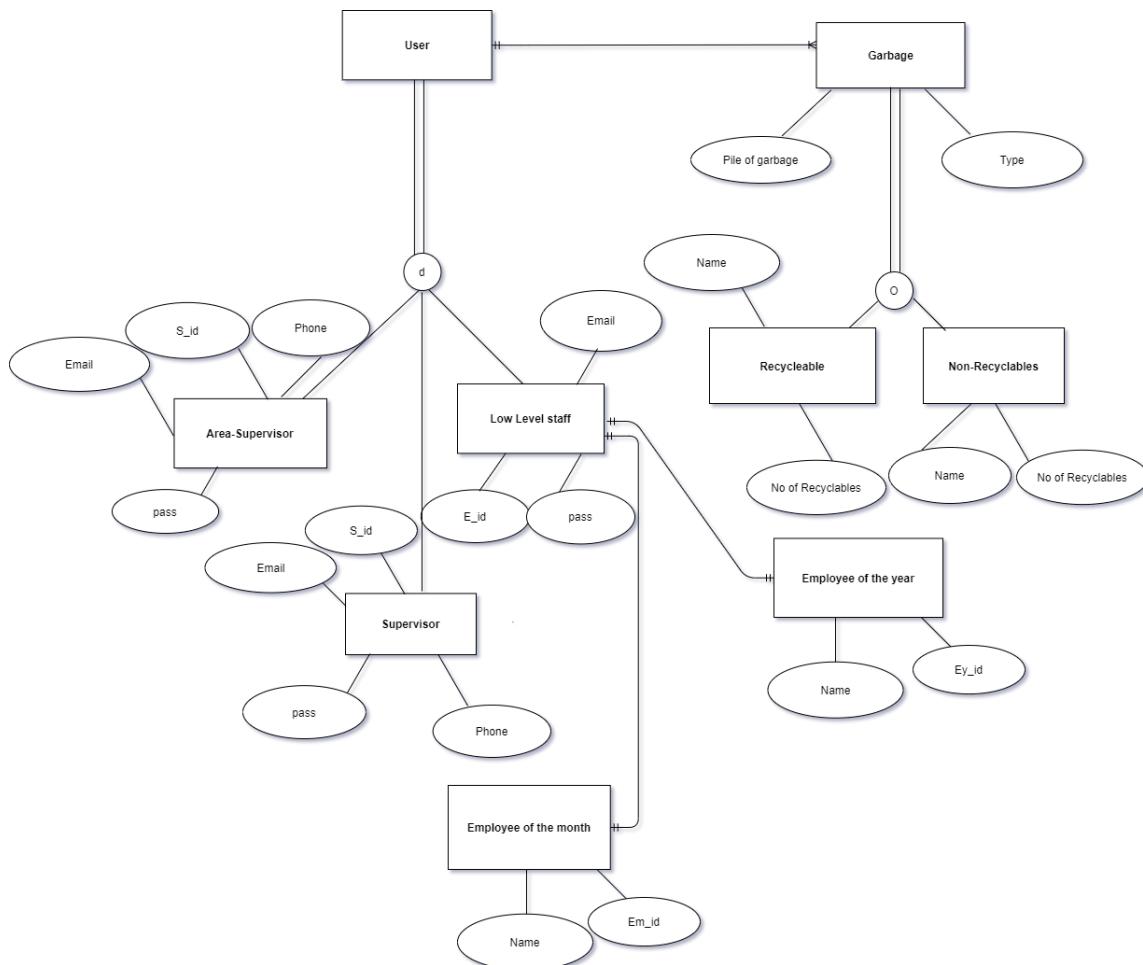
- **Entities** in system
- **Relationship** between entities
- **Attributes** of entities

#### Main Entities in our system

1. Camera Operator / User
2. Municipal committee

#### Associated Entities:

1. Operating Cameras/Drone
2. Receive alert notification



*Figure 17 : Entity Relation Diagram*

### 3.6. Domain Model Diagram

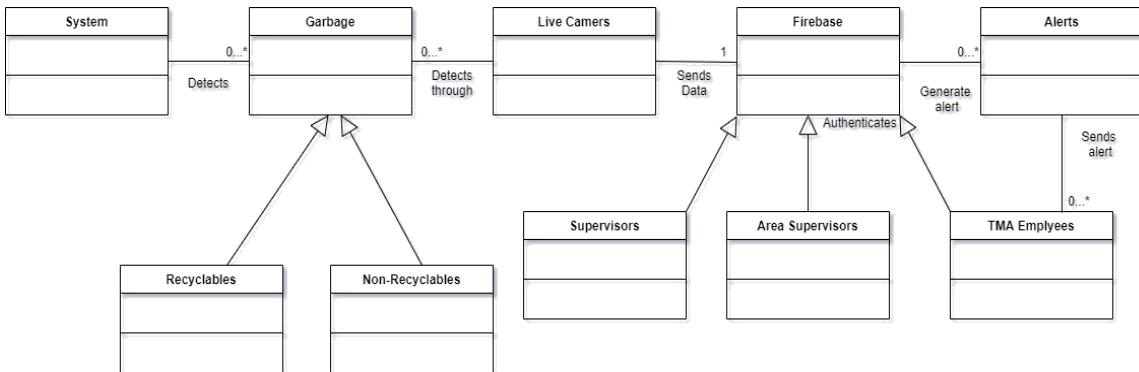


Figure 18 : Domain Model Diagram

### 3.7. Data Flow Diagram

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. Dataflow Diagram of Smart Garbage Management for Sustainable City Life Shown as Follows:

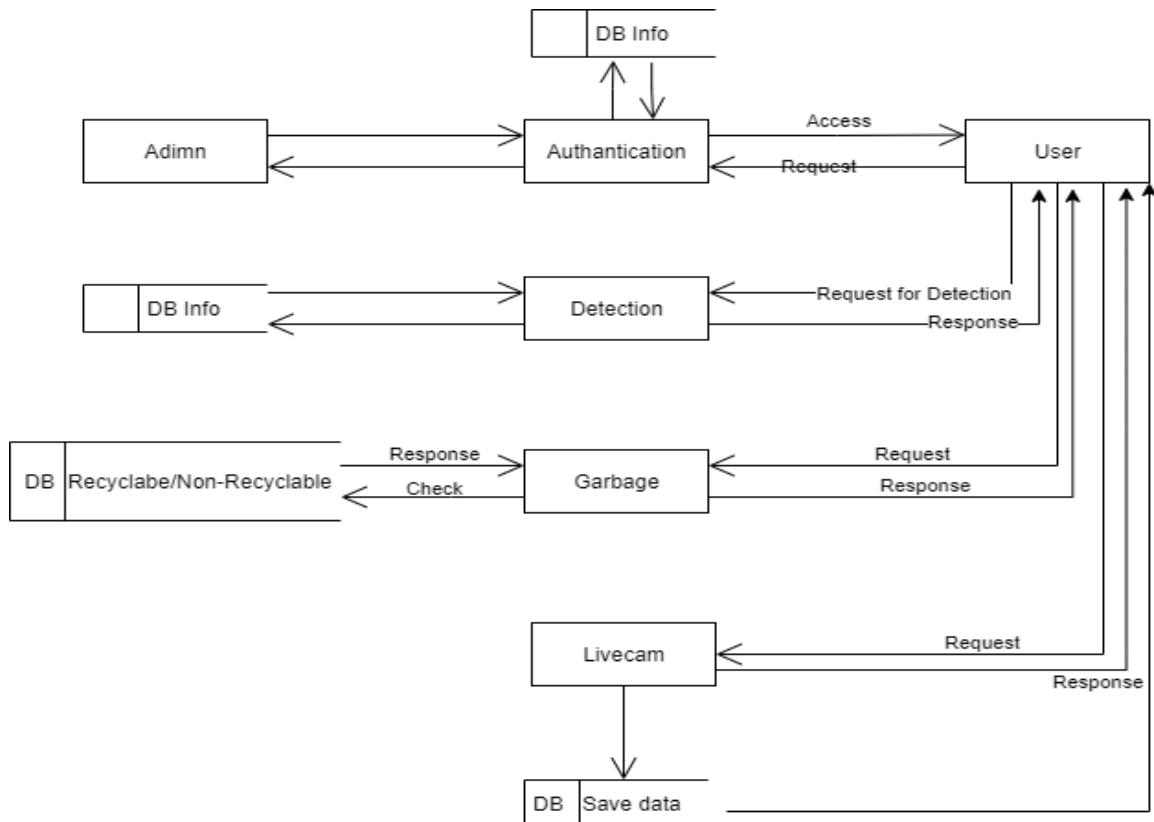


Figure 19 : Data Flow Diagram

## **Chapter 4: User Interface Design**

---

## 4.1. Introduction

A user interface design consists of three main parts:

Page elements should be visualized on paper before building them in the computer. Just as you draw a site map to plan the site, use cartoons and storyboards to begin blocking out the site's appearance and navigational scheme.

1. Site maps
2. Storyboards
3. Navigational maps
4. Traceability Matrix

## 4.2. Site Maps

A site map's main benefit is to give users an overview of the site's areas in a single glance by dedicating an entire page to a visualization of the information architecture. If designed well, this overview can include several levels of hierarchy, and yet not be so big that users lose their ability to grasp the map. The site map should include.

- Home page
- Primary Navigation
- Secondary Navigation
- Tertiary Pages

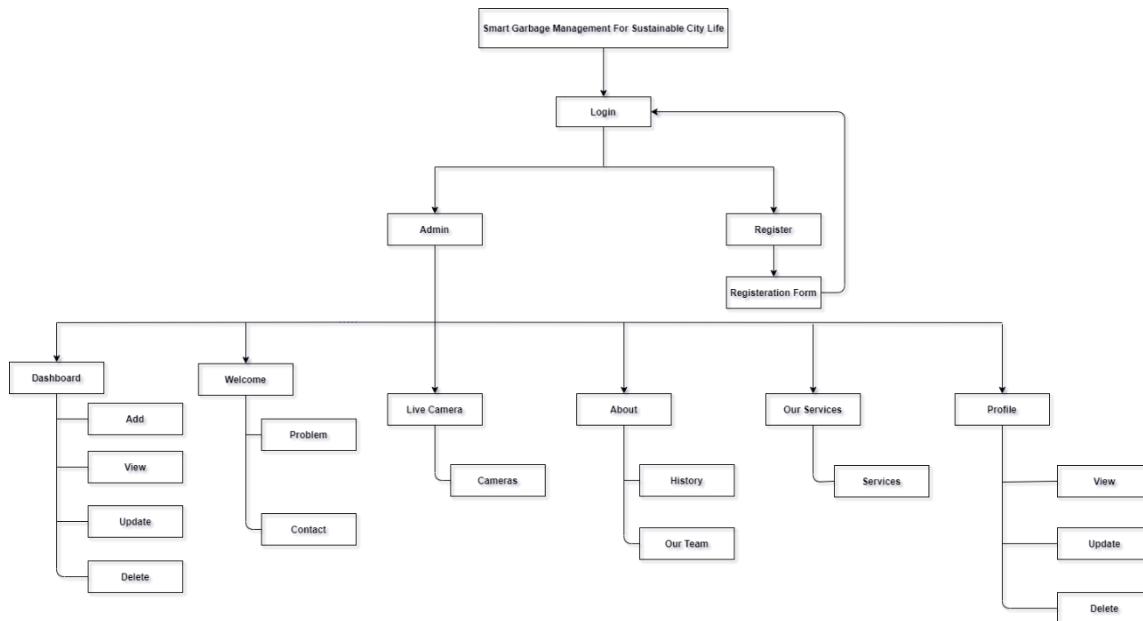
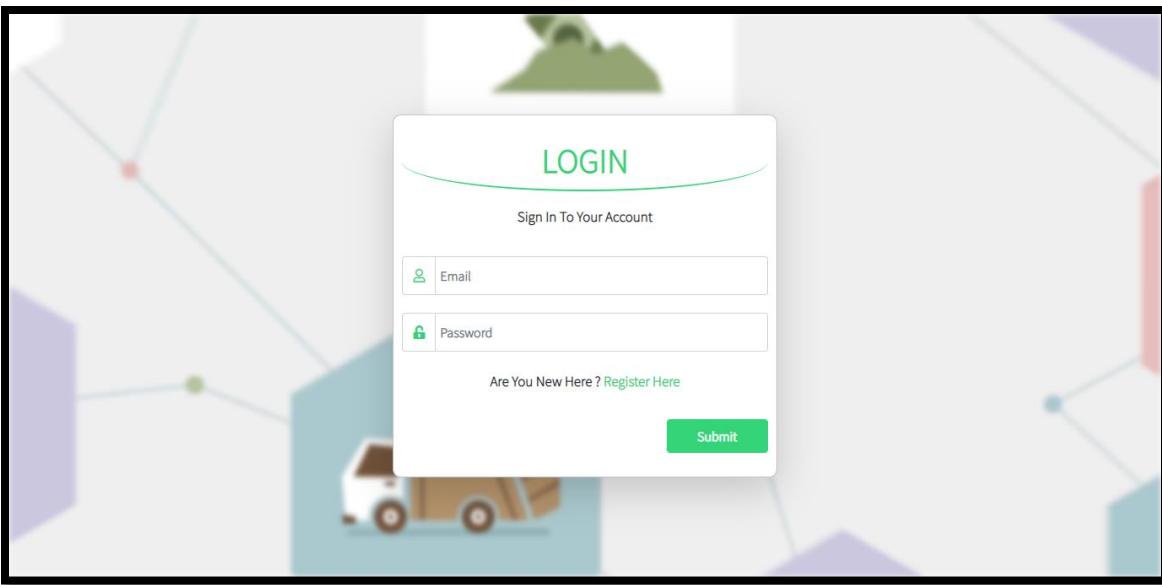


Figure 20 : Site Map

## 4.3. Story boards

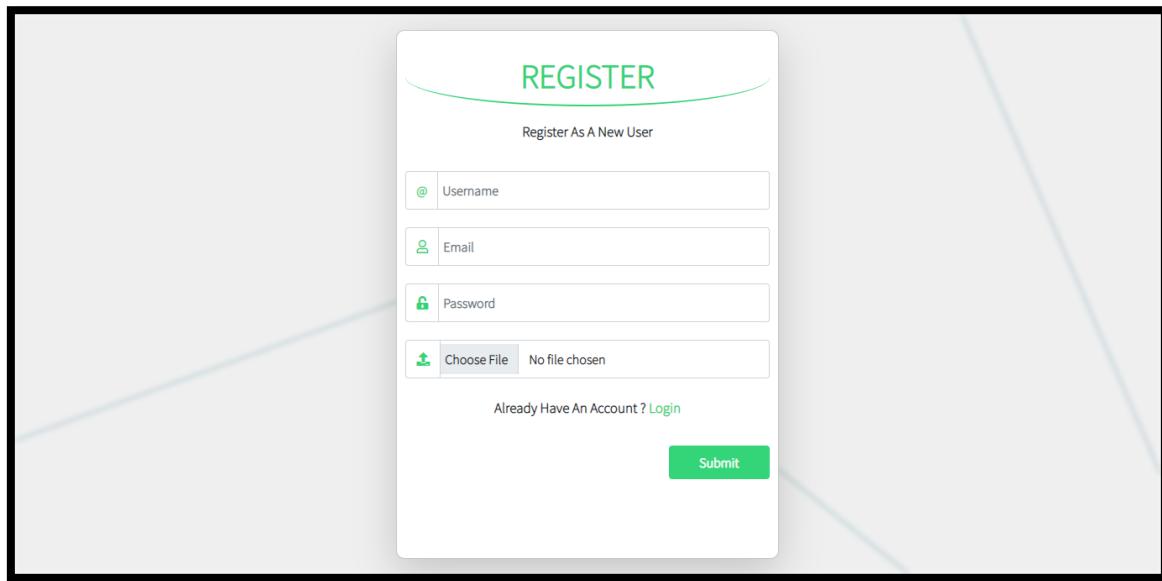
A storyboard is a sequence of single images, each of which represents a distinct event or narrative. It is also a visual representation of the script illustrating the interaction between the user and the machine.

**UI-1:** Login Screen



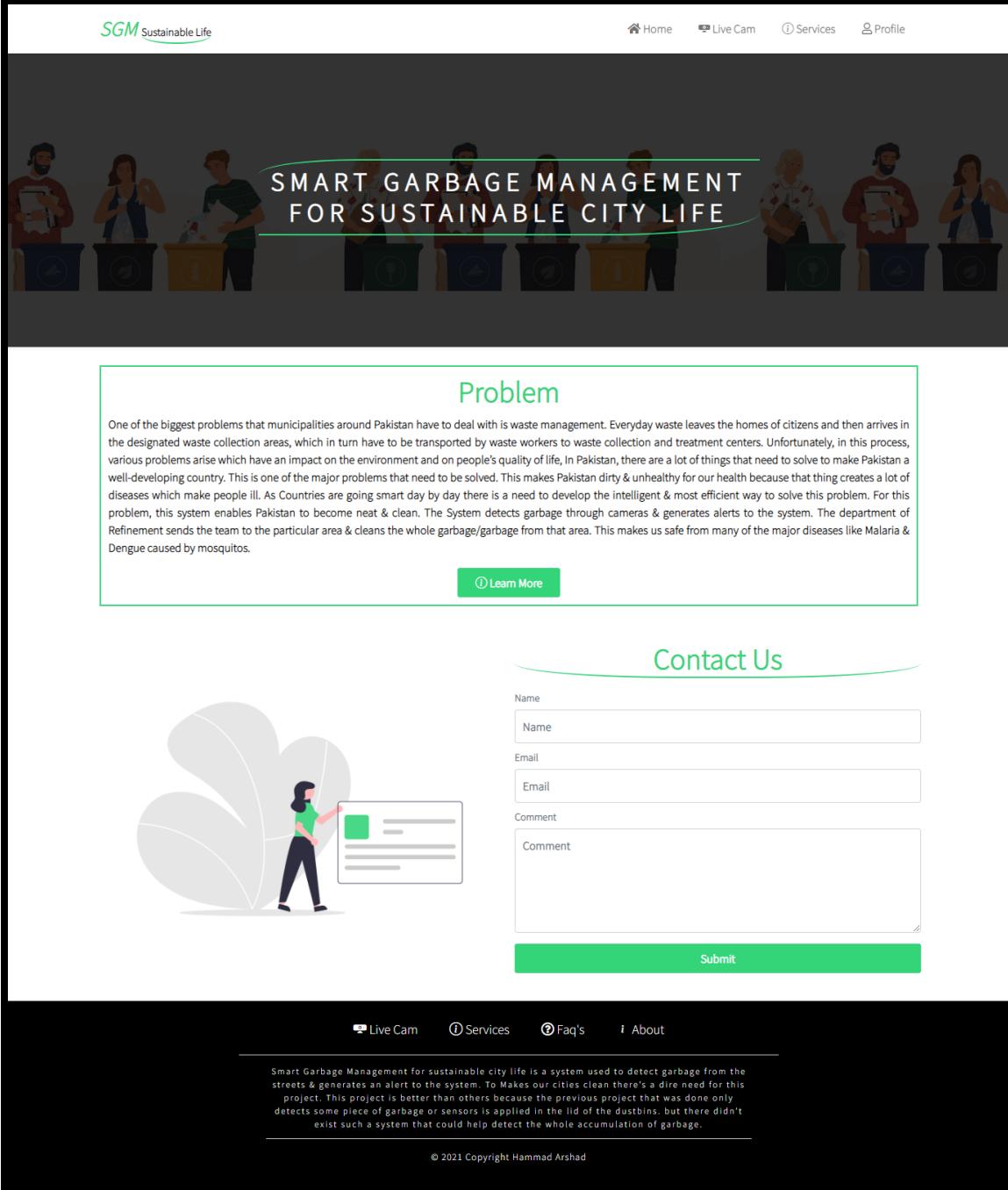
*Figure 21 : Login Screen*

**UI-2:** Register Screen



*Figure 22 : Register Screen*

### UI-3: Welcome Screen



The image shows the welcome screen of a web application for 'Smart Garbage Management for Sustainable City Life'. The header features the logo 'SGM Sustainable Life' and navigation links for Home, Live Cam, Services, and Profile. The main banner has a dark background with silhouettes of people and recycling bins, and the text 'SMART GARBAGE MANAGEMENT FOR SUSTAINABLE CITY LIFE'. Below the banner, a green-bordered box contains the heading 'Problem' and a detailed paragraph about waste management challenges in Pakistan. A 'Learn More' button is at the bottom of this box. To the right, there's a 'Contact Us' section with a form for Name, Email, and Comment, and a 'Submit' button. A decorative graphic of a person interacting with a large leaf and a document is on the left. The footer includes links for Live Cam, Services, Faq's, and About, along with a copyright notice for 2021.

SGM Sustainable Life

Home Live Cam Services Profile

SMART GARBAGE MANAGEMENT FOR SUSTAINABLE CITY LIFE

**Problem**

One of the biggest problems that municipalities around Pakistan have to deal with is waste management. Everyday waste leaves the homes of citizens and then arrives in the designated waste collection areas, which in turn have to be transported by waste workers to waste collection and treatment centers. Unfortunately, in this process, various problems arise which have an impact on the environment and on people's quality of life. In Pakistan, there are a lot of things that need to be solved to make Pakistan a well-developing country. This is one of the major problems that need to be solved. This makes Pakistan dirty & unhealthy for our health because that thing creates a lot of diseases which make people ill. As Countries are going smart day by day there is a need to develop the intelligent & most efficient way to solve this problem. For this problem, this system enables Pakistan to become neat & clean. The System detects garbage through cameras & generates alerts to the system. The department of Refinement sends the team to the particular area & cleans the whole garbage/garbage from that area. This makes us safe from many of the major diseases like Malaria & Dengue caused by mosquitos.

① Learn More

**Contact Us**

Name

Email

Comment

Comment

Submit

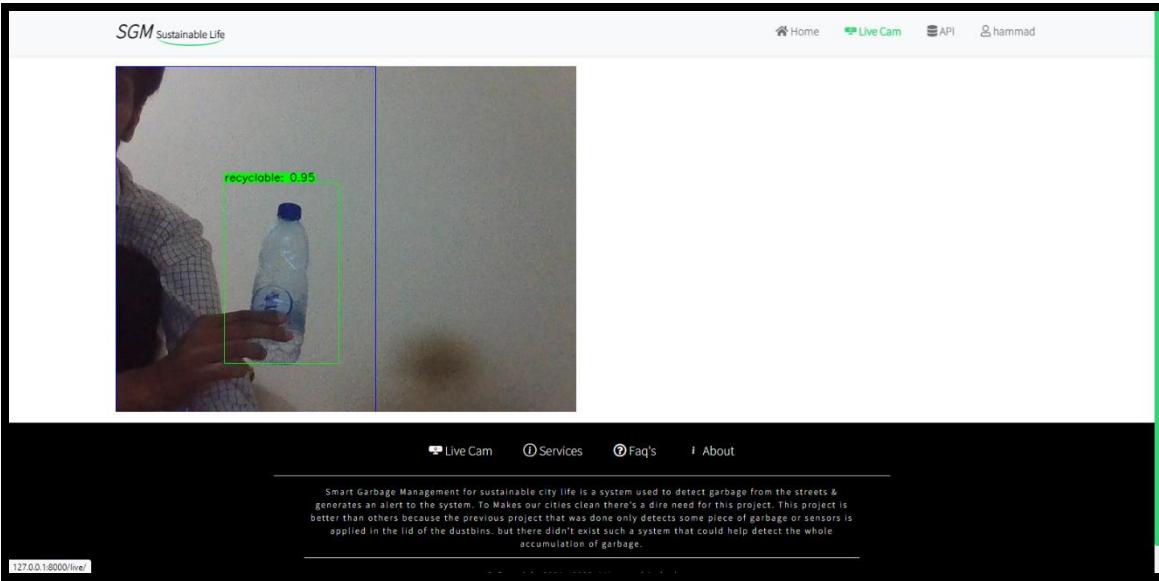
Live Cam Services Faq's About

Smart Garbage Management for sustainable city life is a system used to detect garbage from the streets & generates an alert to the system. To Makes our cities clean there's a dire need for this Project. This project is better than others because the previous project that was done only detects some piece of garbage or sensors is applied in the lid of the dustbins, but there didn't exist such a system that could help detect the whole accumulation of garbage.

© 2021 Copyright Hammad Arshad

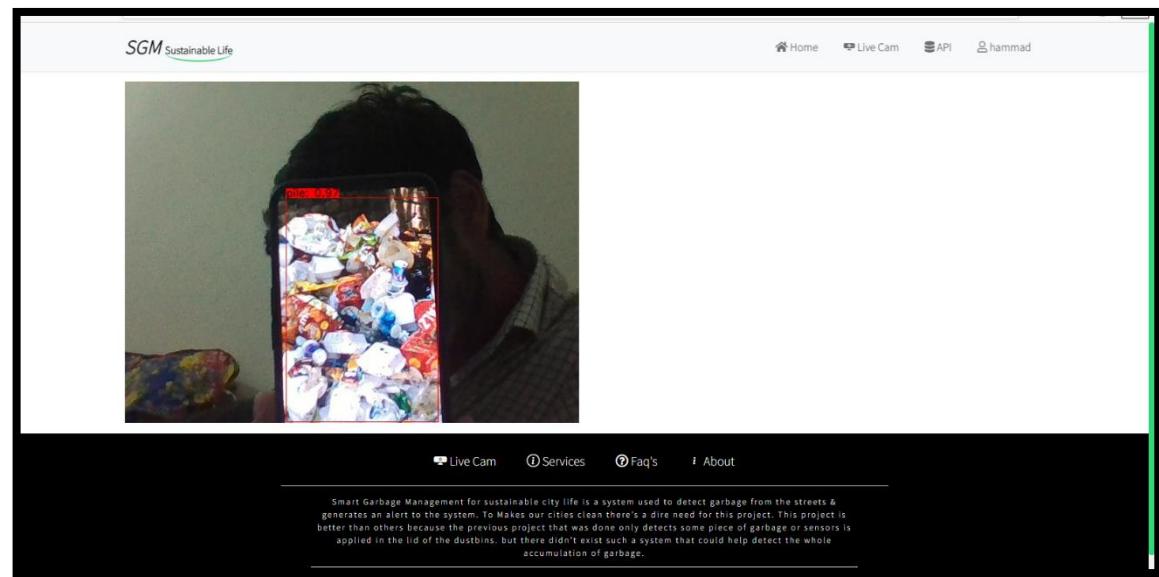
**Figure 23 : Welcome Screen**

#### UI-4: Live Camera Screen



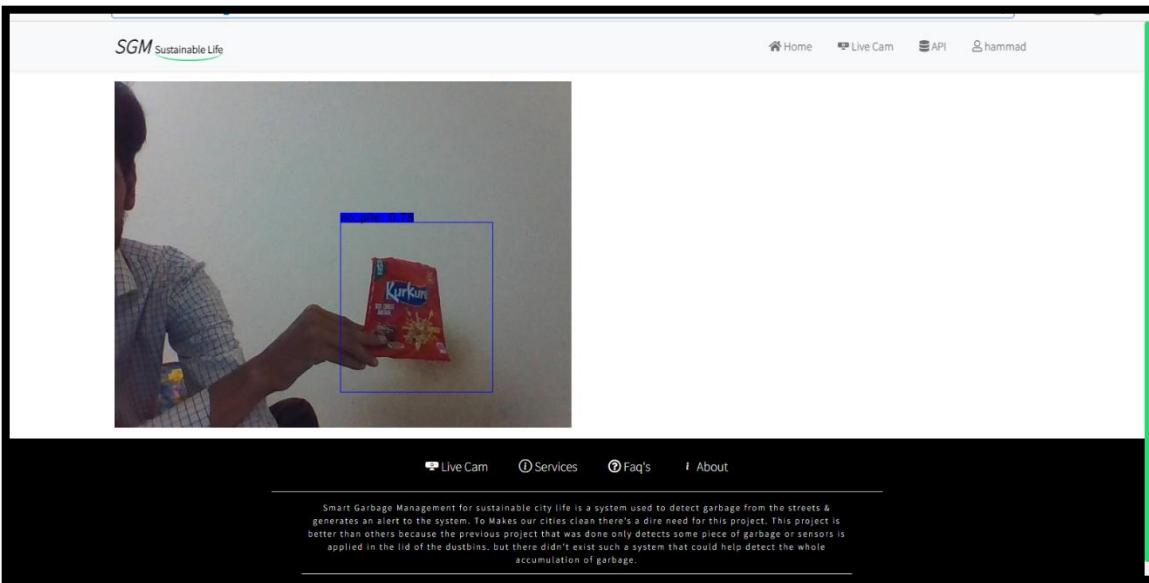
*Figure 24 : Live Stream Recyclable Detection*

#### UI-4: Live Camera Screen 2



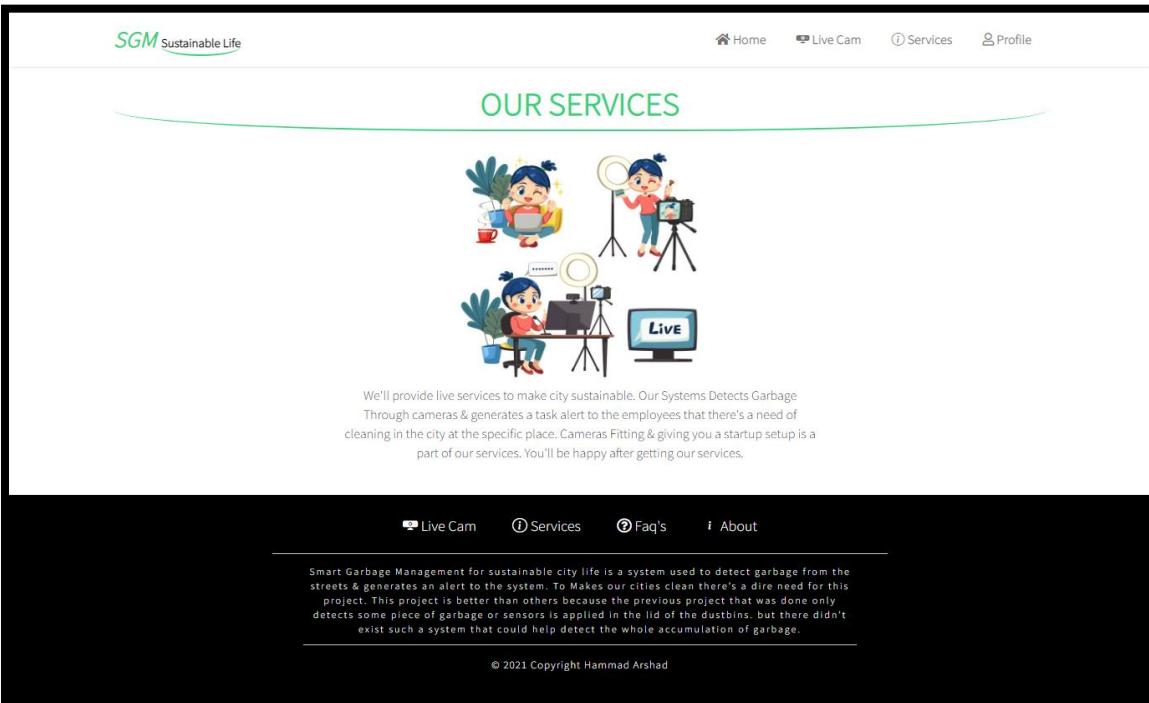
*Figure 25 : Live Stream Pile Detection*

## UI-4: Live Video Screen 3



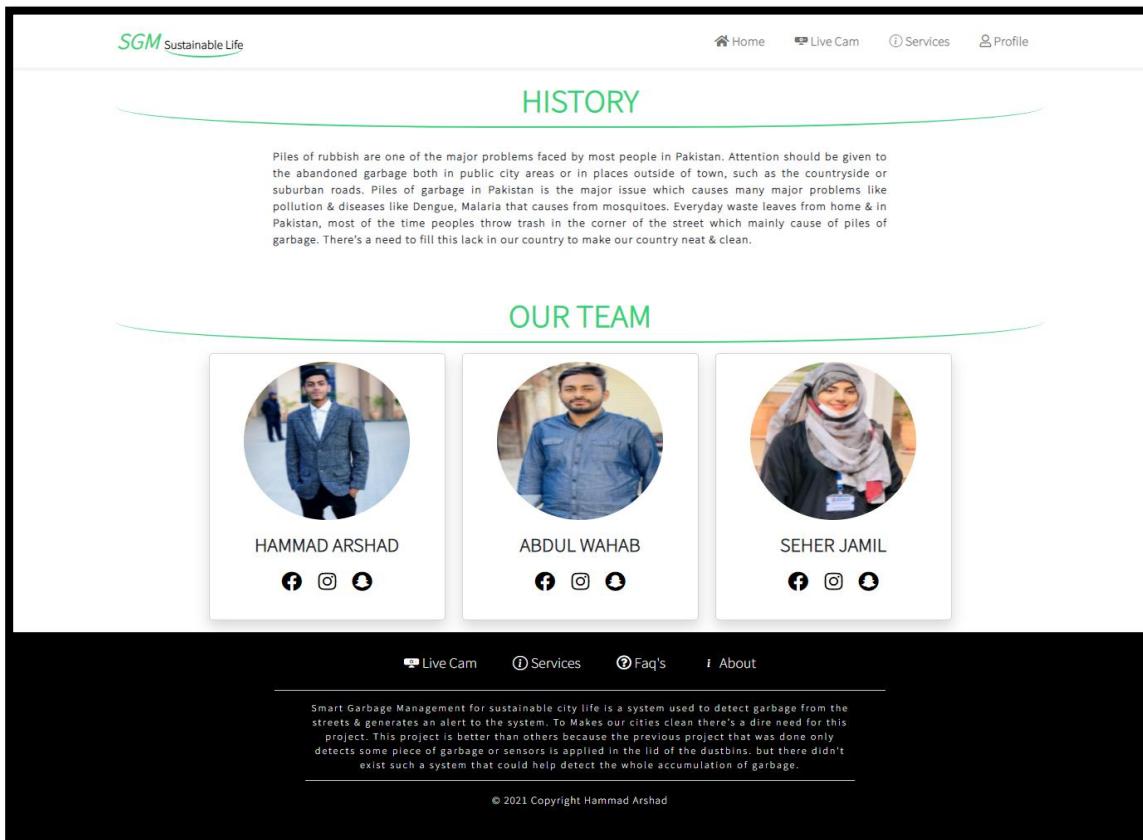
**Figure 26 : Live Stream No Pile Detection**

## UI-5 Services



**Figure 27 : Our Services Screen**

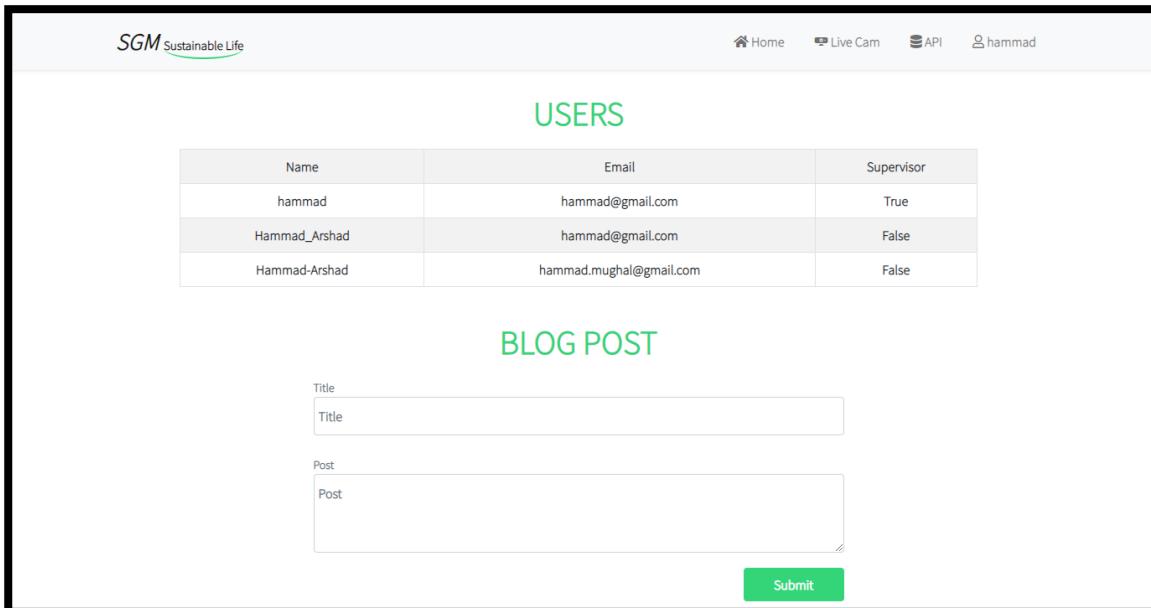
## UI-6: About Screen



The screenshot shows the 'About' section of the SGM Sustainable Life website. At the top, there is a navigation bar with links for Home, Live Cam, Services, and Profile. Below the navigation, a section titled 'HISTORY' contains a paragraph about the issue of piles of rubbish in Pakistan and its impact on the environment and public health. A section titled 'OUR TEAM' follows, featuring three team members: Hammad Arshad, Abdul Wahab, and Seher Jamil, each with a profile picture and social media links. At the bottom, there is a footer with links for Live Cam, Services, Faq's, and About, along with a copyright notice for 2021.

Figure 28 : About Screen

## UI-7: Dashboard Screen

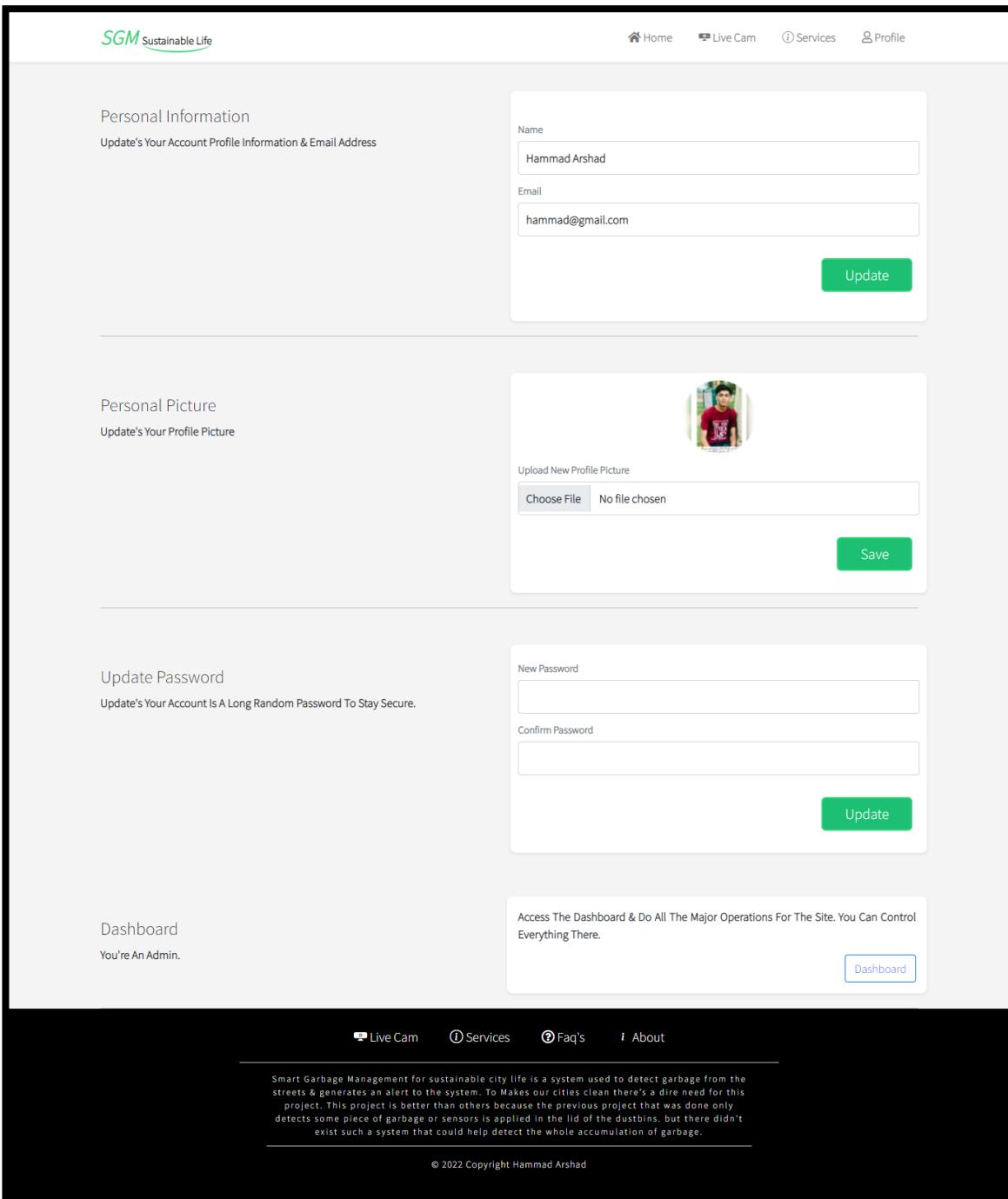


The screenshot shows the 'Dashboard' section of the SGM Sustainable Life website. At the top, there is a navigation bar with links for Home, Live Cam, API, and a user profile for 'hammad'. Below the navigation, a section titled 'USERS' displays a table with four user entries. A section titled 'BLOG POST' follows, containing fields for 'Title' and 'Post' with a 'Submit' button.

Name	Email	Supervisor
hammad	hammad@gmail.com	True
Hamad_Arshad	hammad@gmail.com	False
Hammad-Arshad	hammad.mughal@gmail.com	False

Figure 29 : Dashboard Screen

## UI-8: Profile Screen



The image shows the 'Profile Screen' of the Smart Garbage Management System. The top navigation bar includes 'Home', 'Live Cam', 'Services', and 'Profile'. The main content area is divided into four sections: 'Personal Information', 'Personal Picture', 'Update Password', and 'Dashboard'. The 'Personal Information' section contains fields for 'Name' (Hammad Arshad) and 'Email' (hammad@gmail.com), with an 'Update' button. The 'Personal Picture' section shows a placeholder image and a file upload input ('Choose File: No file chosen'), with a 'Save' button. The 'Update Password' section has fields for 'New Password' and 'Confirm Password', with an 'Update' button. The 'Dashboard' section displays a message about controlling the site and a 'Dashboard' button. The bottom navigation bar includes 'Live Cam', 'Services', 'FAQ's', and 'About'. A copyright notice for Hammad Arshad is at the bottom.

SGM Sustainable Life

Home Live Cam Services Profile

Personal Information

Update's Your Account Profile Information & Email Address

Name

Hammad Arshad

Email

hammad@gmail.com

Update

Personal Picture

Update's Your Profile Picture

Upload New Profile Picture

Choose File No file chosen

Save

Update Password

Update's Your Account Is A Long Random Password To Stay Secure.

New Password

Confirm Password

Update

Dashboard

You're An Admin.

Access The Dashboard & Do All The Major Operations For The Site. You Can Control Everything There.

Dashboard

Live Cam Services FAQ's About

Smart Garbage Management for sustainable city life is a system used to detect garbage from the streets & generates an alert to the system. To Makes our cities clean there's a dire need for this project. This project is better than others because the previous project that was done only detects some piece of garbage or sensors is applied in the lid of the dustbins, but there didn't exist such a system that could help detect the whole accumulation of garbage.

© 2022 Copyright Hammad Arshad

Figure 30 : Profile Screen

## UI-9: API

```
Django REST framework
hammad

Get Users
OPTIONS GET

GET /api/
HTTP 200 OK
Allow: GET, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 1,
    "password": "pbkdf2_sha256$320000$05iskqaesy@0U2wIxFxhVbd$hevK7qAtR6ED5hGeCzYFmzVr@KaG9jhPrHsFhByrIdU=",
    "last_login": "2022-06-14T13:25:52.187652Z",
    "is_superuser": true,
    "username": "hammad",
    "first_name": "",
    "last_name": "",
    "email": "hammad@gmail.com",
    "is_staff": true,
    "is_active": true,
    "date_joined": "2022-03-15T20:01:57.765176Z",
    "groups": [],
    "user_permissions": []
  },
  {
    "id": 4,
    "password": "pbkdf2_sha256$320000$p7P+YECR0bg7UbRzX4IwOF$YNNKBh0ZHxY+hd5bApUnxV1UFVh522Ej39i2RafuqQY=",
    "last_login": "2022-04-01T18:58:37.483753Z",
    "is_superuser": false,
    "username": "Hammad_Arshad",
    "first_name": "Hammad",
    "last_name": "Arshad",
    "email": "hammad@gmail.com",
    "is_staff": false,
    "is_active": true,
    "date_joined": "2022-03-29T15:27:20Z",
    "groups": [],
    "user_permissions": []
  },
  {
    "id": 7,
    "password": "pbkdf2_sha256$320000$K1woHlyRvHXBak80R580YVSdE89hFzP31cHr61KUzgqo4RH5/vcxMitXi6NaTHgNo=",
    "last_login": "2022-06-12T07:46:00.860604Z",
    "is_superuser": false,
    "username": "hammad-Arshad",
    "first_name": "",
    "last_name": "",
    "email": "hammad.mughal@gmail.com",
    "is_staff": false,
    "is_active": true,
    "date_joined": "2022-06-11T13:48:58.054768Z",
    "groups": [],
    "user_permissions": []
  }
]
```

Figure 31 : API Screen for Authentication

## UI-10: API/blogs

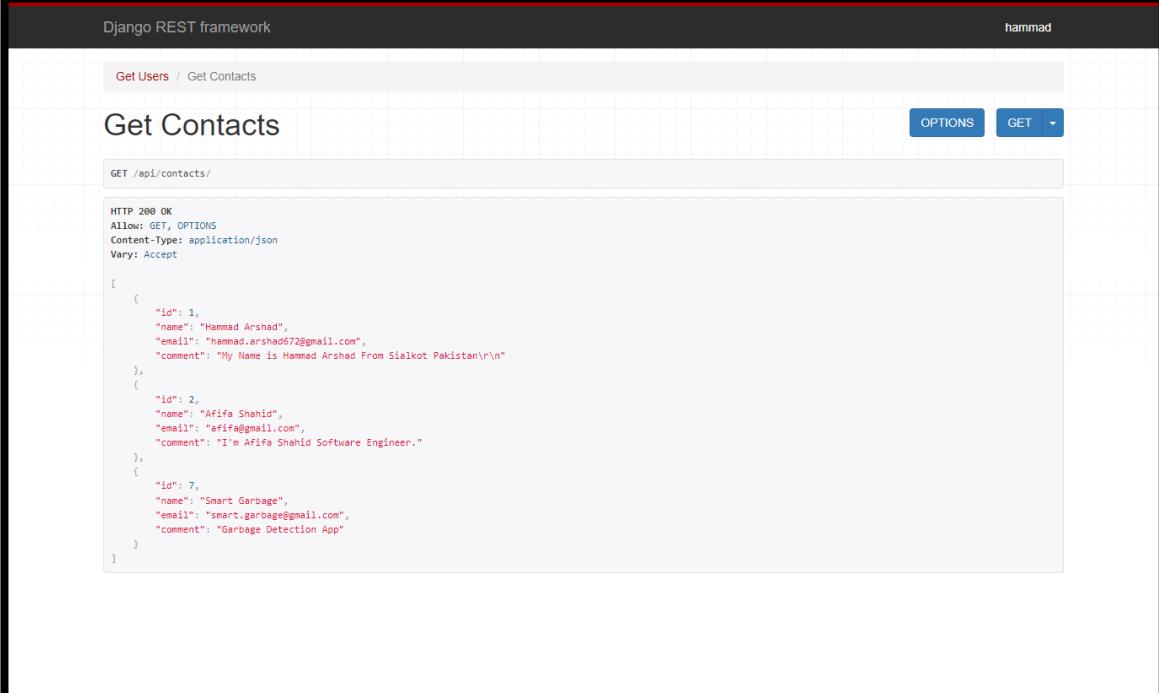
```
OPTIONS GET

GET /api/blogs/
HTTP 200 OK
Allow: GET, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 5,
    "title": "Smart Garbage Management",
    "postDetails": "Smart Garbage Management for sustainable city life is a system used to detect garbage from the streets & generates an alert to the system. To Makes"
  }
]
```

Figure 32 : API Screen for Blogs

## UI-11: API/contacts



```
HTTP 200 OK
Allow: GET, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 1,
    "name": "Hammad Arshad",
    "email": "hammad.arshad672@gmail.com",
    "comment": "My Name is Hammad Arshad From Sialkot Pakistan\n\n"
  },
  {
    "id": 2,
    "name": "Afifa Shahid",
    "email": "afifa@gmail.com",
    "comment": "I'm Afifa Shahid Software Engineer."
  },
  {
    "id": 7,
    "name": "Smart Garbage",
    "email": "smart.garbage@gmail.com",
    "comment": "Garbage Detection App"
  }
]
```

Figure 33 : API Screen for Contacts

## UI-12: Blogs

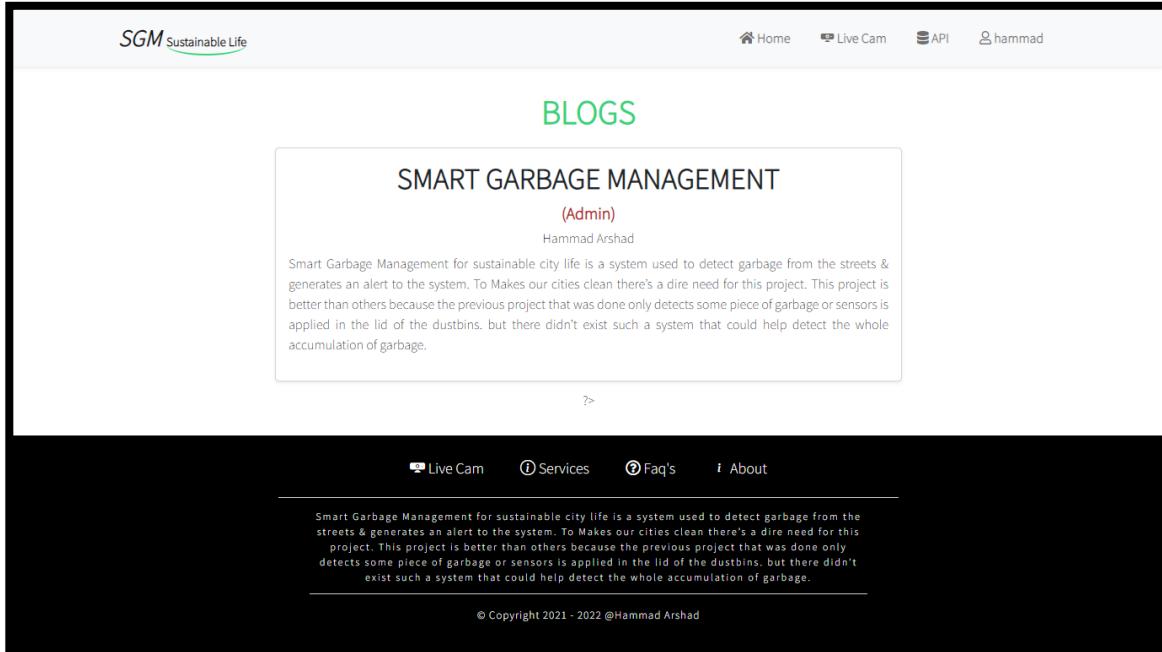
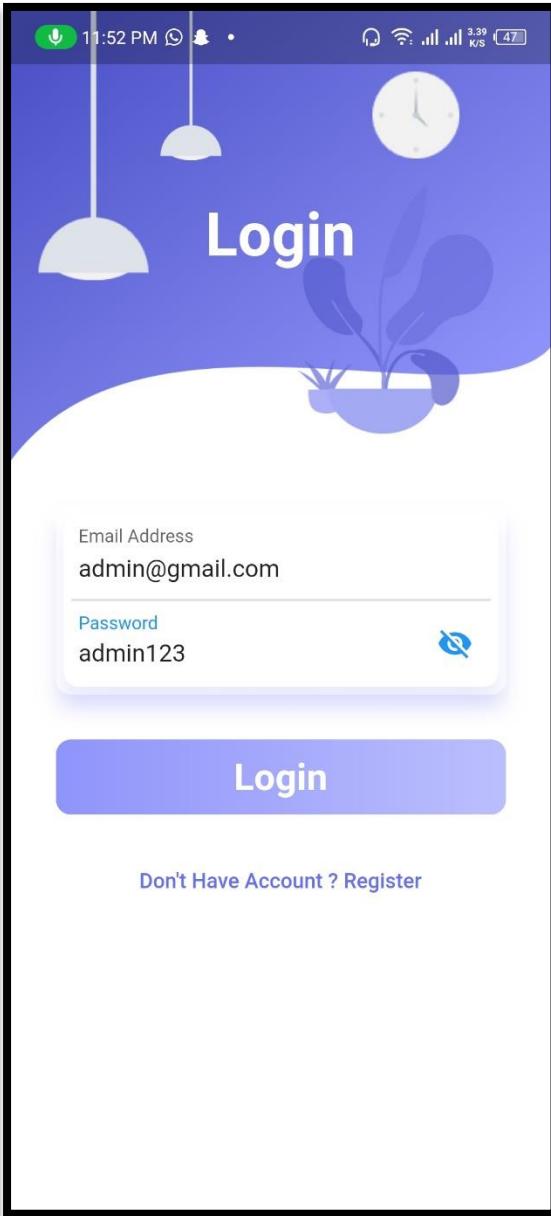


Figure 34 : Blogs Screen



*Figure 36 : Splash Screen*



*Figure 35 : Login Screen*

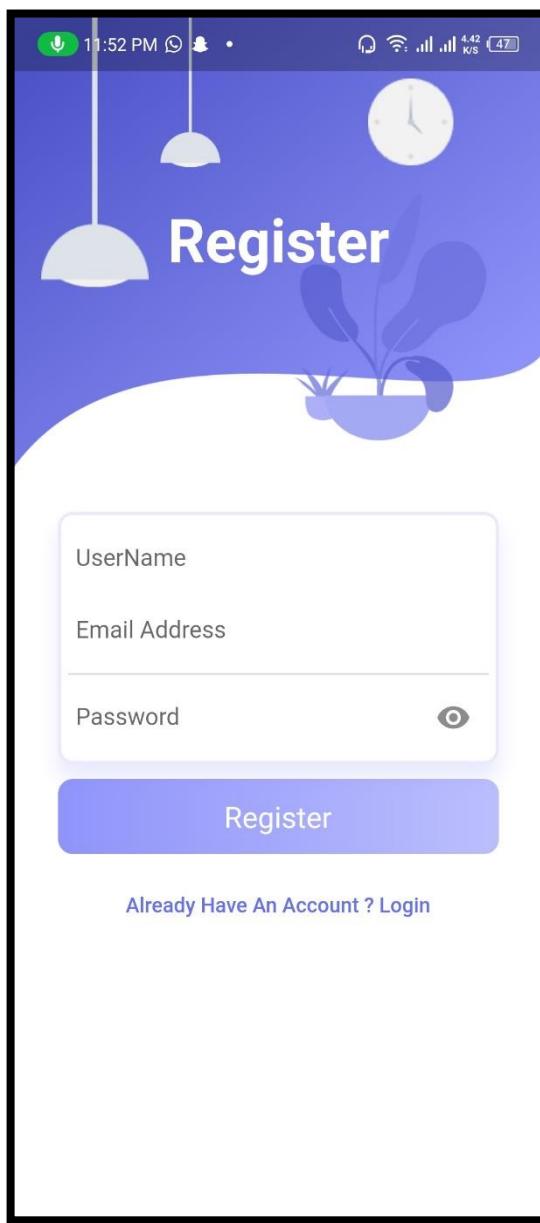


Figure 38 : Register Screen

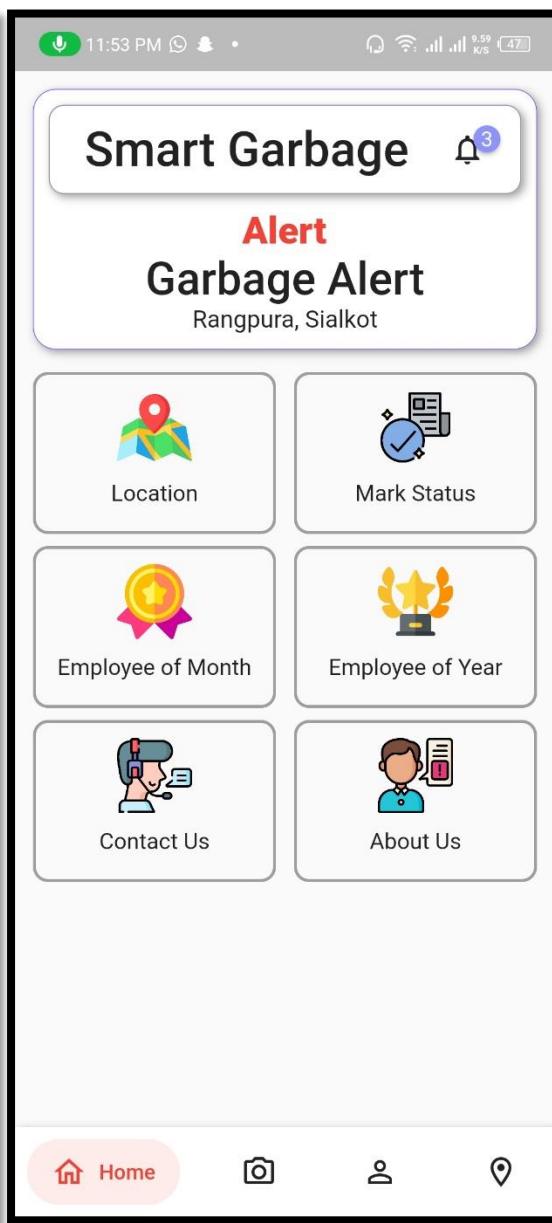


Figure 37 : Home Screen

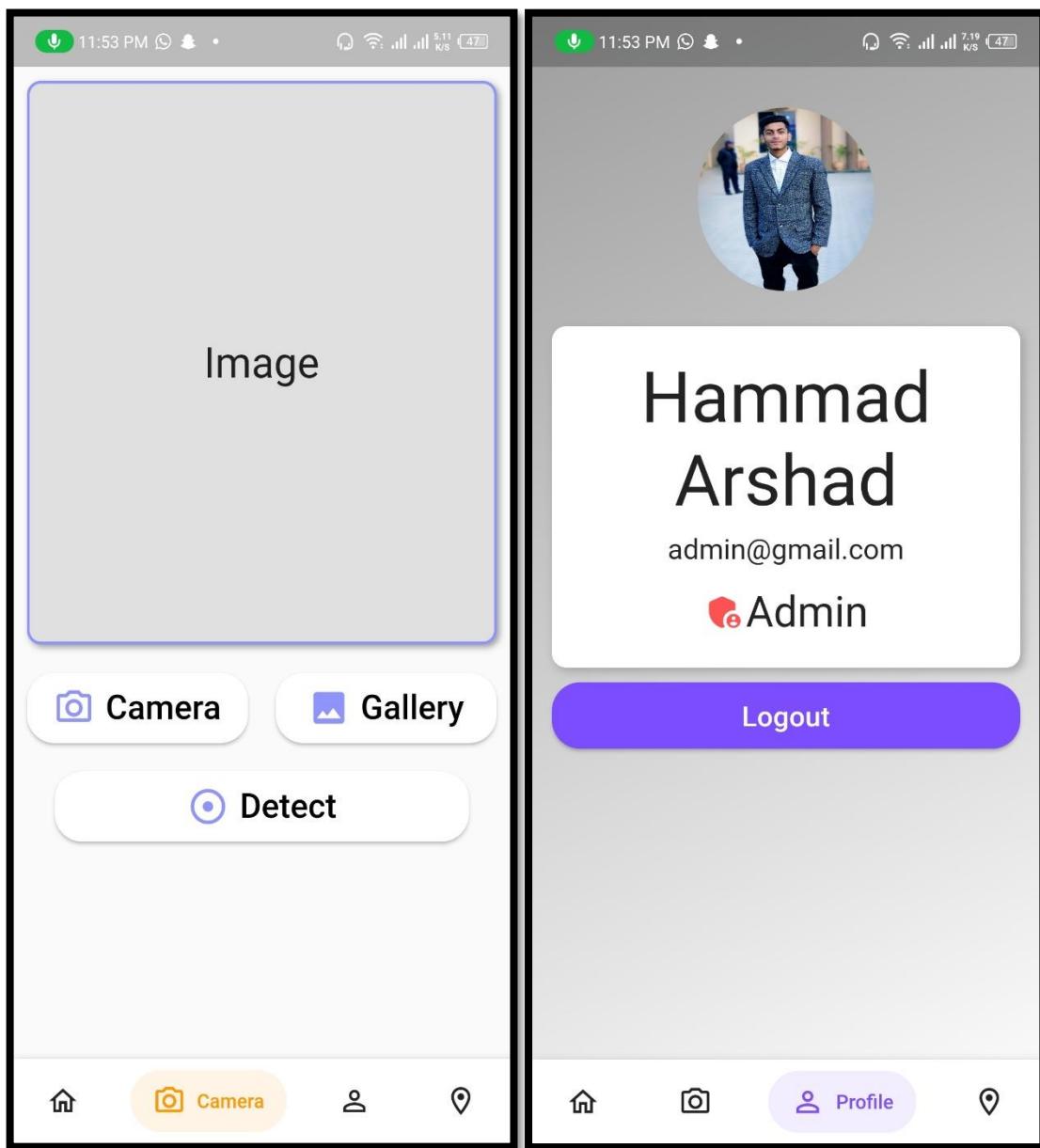


Figure 40 : Camera Screen

Figure 39 : Profile Screen

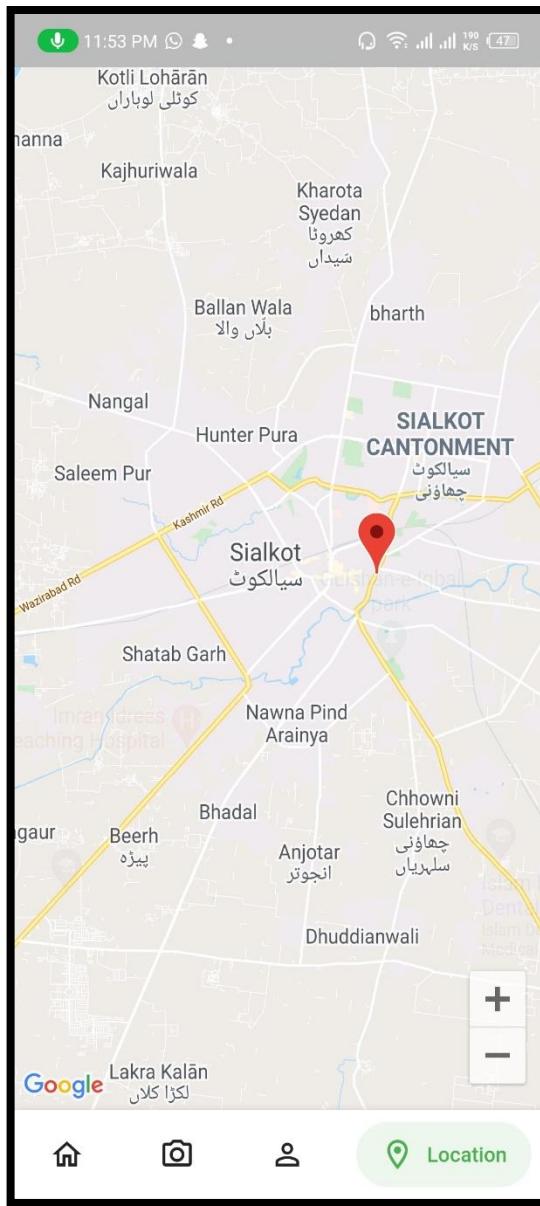


Figure 41 : Map Screen

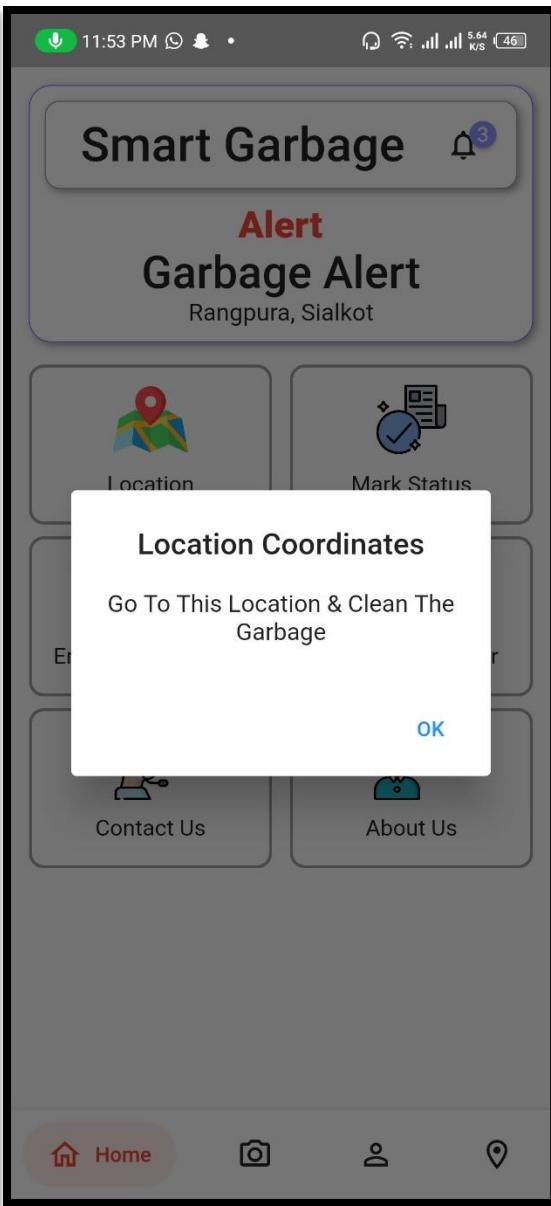


Figure 42 : Get Location

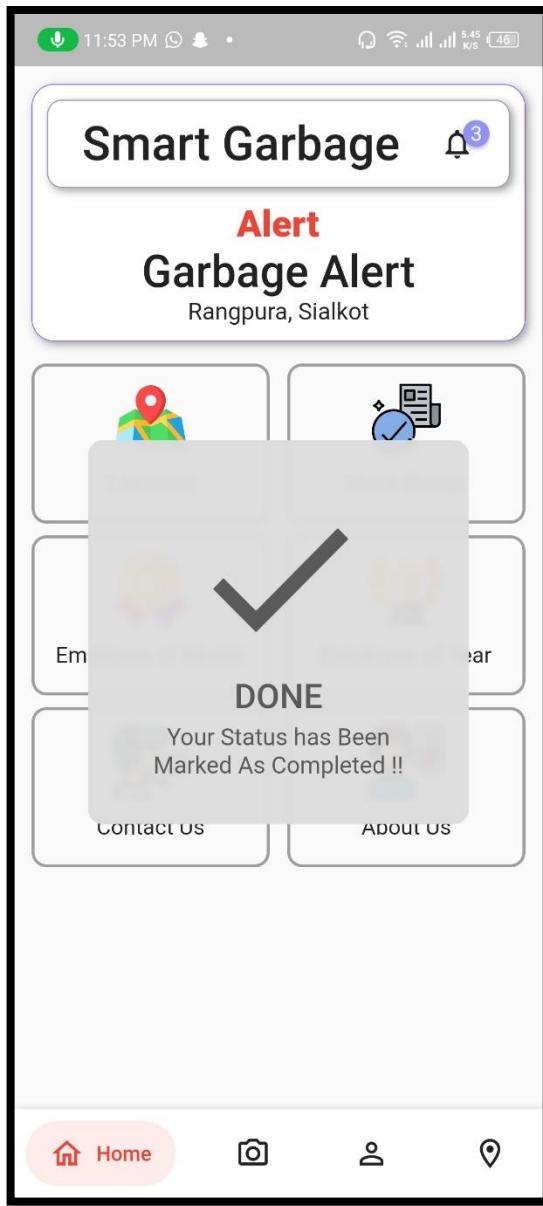


Figure 43 : Mark Complete Status

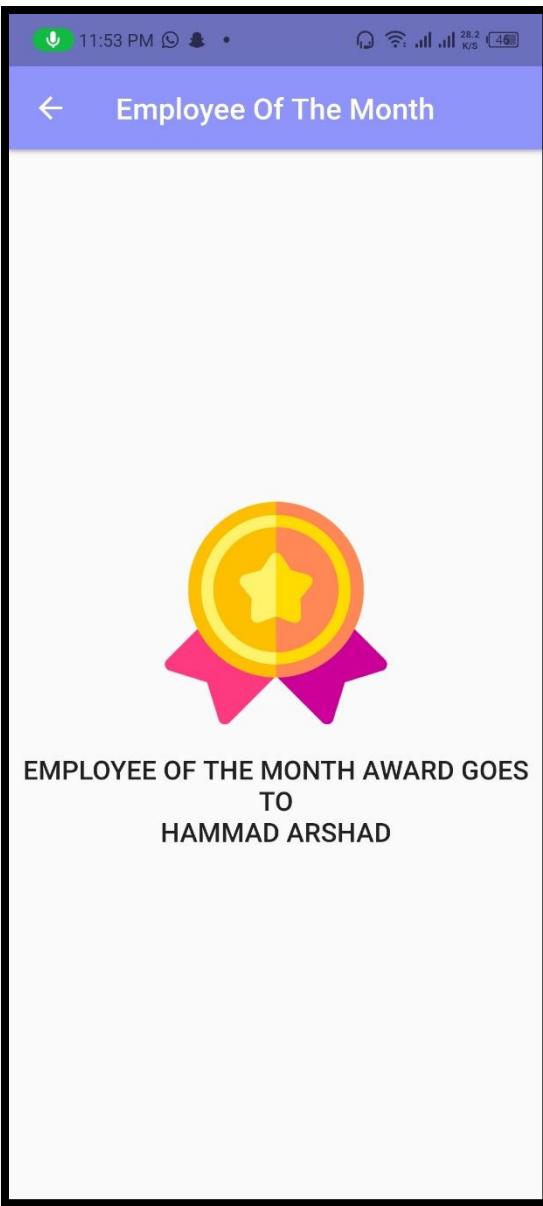


Figure 44 : Employee of Month

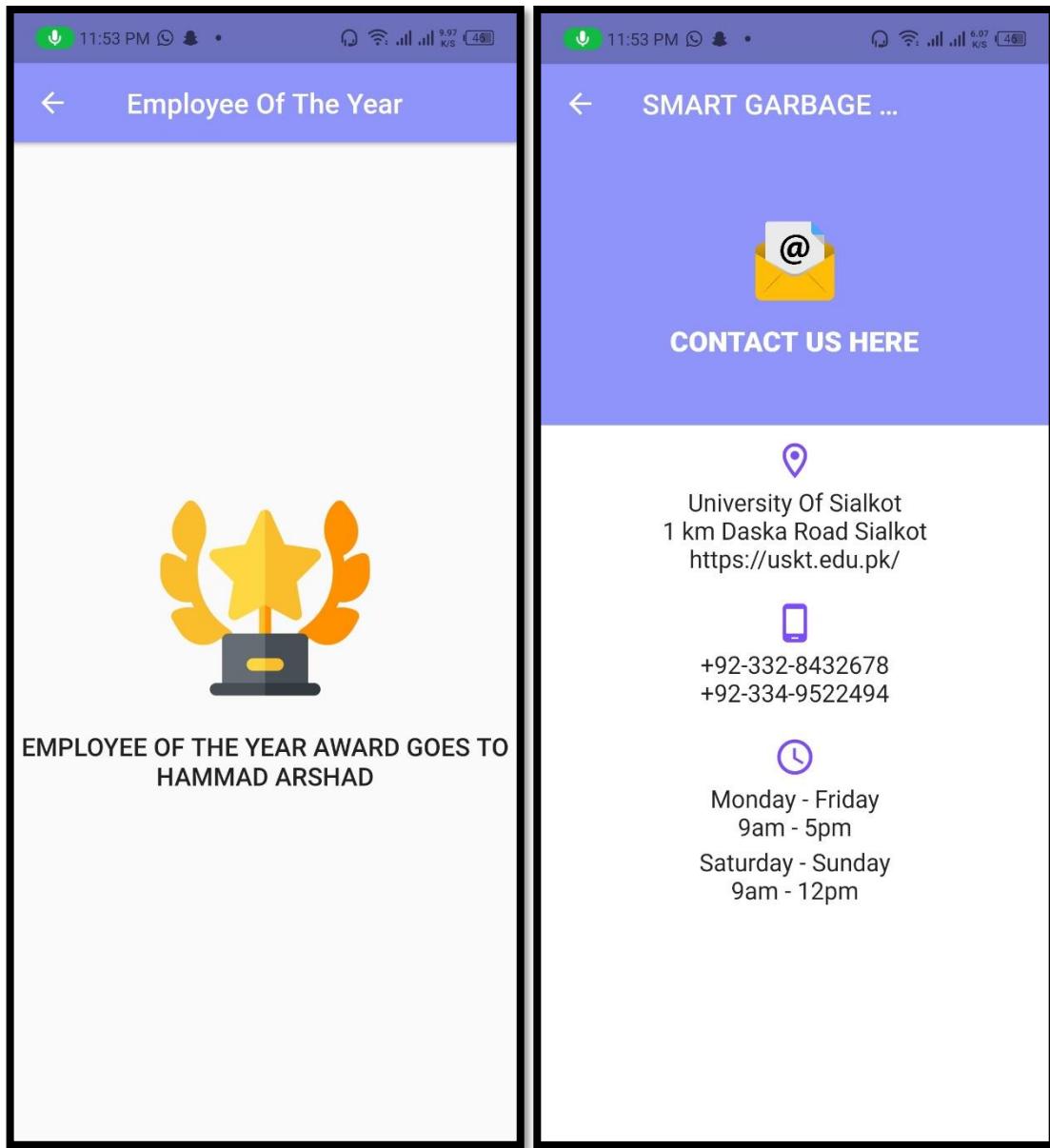


Figure 46 : Employee of Year

Figure 45 : Contact Us



*Figure 48 : About 1*



*Figure 47 : About 2*

## **Chapter 5: Software Testing**

---

## 5.1. Introduction

From the start of software engineering different software testing schemes and testing templates have been invented. These testing templates are used to test both UI and unit testing. Every credible software house developed its testing documentation to develop efficient systems. To provide a common set of standardized documents the IEEE developed the 829 Standard for Software Test Documentation for any type of software testing, including User Acceptance Testing. In our testing module, we are going to test different features and modules of the system using a documentation template supported by IEEE829. There are mainly three types of testing in which we perform only two.

- Black Box Testing
- White Box Testing
- Grey Box Testing

## 5.2. Black box plan/White box plan

### 5.2.1. Black Box Testing

Black box testing is a software testing technique in which the tester doesn't have any knowledge related to code. The tester is fully unaware of the code. The tester performs different types of tests on the system & gives results. The tester can be any person who can run the system. We can also consider end-user testers. Sometimes we give beta testing to some users to test the functionality of the system & if there are any bugs in the system report them to the admin. This is also an example of black-box testing.

#### 5.2.1.1 Test Case

TC-1

**Table 44 : Test Case 1**

<b>Test Engineer:</b>	Abdul Wahab
<b>Test Case ID:</b>	TC1
<b>Related UC/FR:</b>	UC_1
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	Authenticate the user
<b>Pre-Req:</b>	Internet connectivity, installed browser, Application Installed, at the login screen
<b>Test Data:</b>	Login Credentials
<b>Steps:</b>	<ol style="list-style-type: none"><li>1. Open Application/URL</li><li>2. Fill the login credentials</li><li>3. Click on Login</li></ol>

<b>Expected Result:</b>	If authenticate goes to the home page otherwise redirect to the same page with an error message.
<b>Actual Result:</b>	Login Successfully
<b>Status:</b>	Pass

## TC-2

**Table 45 : Test Case 2**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC2
<b>Related UC/FR:</b>	UC_2
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	To Validate the User That is It Available
<b>Pre-Req:</b>	The User Must Should Registered
<b>Test Data:</b>	User Credential details.
<b>Steps:</b>	<ul style="list-style-type: none"> <li>• Login to admin panel</li> <li>• Check registered users list.</li> </ul>
<b>Expected Result:</b>	Authenticate that the user is registered
<b>Actual Result:</b>	User Unsuccessfully registered
<b>Status:</b>	Fail

## TC-3

**Table 46 : Test Case 3**

<b>Test Engineer:</b>	Abdul Wahab
<b>Test Case ID:</b>	TC3
<b>Related UC/FR:</b>	UC_3

<b>Date:</b>	5-3-2022
<b>Purpose:</b>	To Register
<b>Pre-Req:</b>	Internet connectivity, installed browser, Application Installed, at login screen
<b>Test Data:</b>	Registration Credentials
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Go to registration Page</li> <li>2. Enter Username, Email &amp; Password</li> <li>3. Click on Register.</li> </ol>
<b>Expected Result:</b>	If registered credentials already exist redirect to the same page with an error message otherwise if registered credentials are valid then register the user.
<b>Actual Result:</b>	Successfully registered
<b>Status:</b>	Pass

#### TC-4

**Table 47 : Test Case 4**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC4
<b>Related UC/FR:</b>	UC_4
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	To Check that Username is Already not Available
<b>Pre-Req:</b>	User Shall pick the username
<b>Test Data:</b>	Username
<b>Steps:</b>	<ul style="list-style-type: none"> <li>• Click on Register as user</li> <li>• Enter valid Username</li> <li>• Submit the form</li> </ul>
<b>Expected Result:</b>	Authenticate user is Already Available or not

<b>Actual Result:</b>	User Registered
<b>Status:</b>	Pass

### TC-5

**Table 48 : Test Case 5**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC5
<b>Related UC/FR:</b>	UC_5
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	Update Profile
<b>Pre-Req:</b>	Login to the system
<b>Test Data:</b>	User updated data (Personal information, Password)
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Go to profile Page.</li> <li>2. Enter Update Personal information or password.</li> <li>3. Click on Update.</li> </ol>
<b>Expected Result:</b>	If the user update data, then update it and redirect to the same page with updated information.
<b>Actual Result:</b>	Profile updated successfully.
<b>Status:</b>	Pass

### TC-6

**Table 49 : Test Case 6**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC6
<b>Related UC/FR:</b>	UC_6
<b>Date:</b>	5-3-2022

<b>Purpose:</b>	To Detect Garbage
<b>Pre-Req:</b>	Login, At Live camera Page.
<b>Test Data:</b>	Piles of garbage.
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login</li> <li>2. Click on Live camera</li> <li>3. Check detected piles of garbage</li> </ol>
<b>Expected Result:</b>	If Piles of Garbage are present in the live stream, then Label the detected piles of Garbage otherwise no piles are detected.
<b>Actual Result:</b>	Detect garbage successfully.
<b>Status:</b>	Pass

## TC-7

**Table 50 : Test Case 7**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC7
<b>Related UC/FR:</b>	UC_7
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	Label the Piles of Garbage
<b>Pre-Req:</b>	Shall detect piles of garbage
<b>Test Data:</b>	Piles of garbage
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login</li> <li>2. Click on live screen.</li> <li>3. See the detected labeled images of piles of Garbage.</li> </ol>
<b>Expected Result:</b>	Detects and labels the piles of garbage.
<b>Actual Result:</b>	Piles of garbage labeled successfully.

<b>Status:</b>	Pass
----------------	------

### TC-8

**Table 51 : Test Case 8**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC8
<b>Related UC/FR:</b>	UC_8
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	To Get Location of piles of garbage
<b>Pre-Req:</b>	Alert along with location co-ordinates
<b>Test Data:</b>	Alert generated from firebase
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Open website</li> <li>2. Click on live screen</li> <li>3. Detect Garbage</li> <li>4. send alert to app along with location co-ordinates</li> </ol>
<b>Expected Result:</b>	If piles of garbage are detected then location coordinates will be sent otherwise re directed to the live screen page.
<b>Actual Result:</b>	Did not identify the exact location of garbage.
<b>Status:</b>	Fail

### TC-9

**Table 52 : Test Case 9**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC9
<b>Related UC/FR:</b>	UC_9
<b>Date:</b>	5-3-2022

<b>Purpose:</b>	To Classify Recyclables products from the piles of garbage through cameras
<b>Pre-Req:</b>	Login, At Live camera Page.
<b>Test Data:</b>	Detected labeled Images
<b>Steps:</b>	<ul style="list-style-type: none"> <li>• open website</li> <li>• Click on live screen.</li> <li>• Detect piles of garbage</li> <li>• Label recyclables.</li> </ul>
<b>Expected Result:</b>	If piles of garbage are detected then identify the recyclables otherwise redirect to the live screen page
<b>Actual Result:</b>	Recyclable products identified successfully.
<b>Status:</b>	Pass

## TC-10

**Table 53 : Test Case 10**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC10
<b>Related UC/FR:</b>	UC_10
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	Send alert to the authenticated user after identification of garbage with location
<b>Pre-Req:</b>	Piles of Garbage should be detected, user login to the application
<b>Test Data:</b>	Detected piles of garbage, user login details
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Detects piles of garbage.</li> <li>2. Send alert to User with location.</li> </ol>
<b>Expected Result:</b>	If piles of garbage are detected then an alert will be sent.
<b>Actual Result:</b>	Alert Message received successfully.

<b>Status:</b>	Pass
----------------	------

### TC-11

**Table 54 : Test Case 11**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC11
<b>Related UC/FR:</b>	UC_11
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	Update User's Work Status
<b>Pre-Req:</b>	User Should Login to the System & get Task
<b>Test Data:</b>	List of allotted tasks
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login to application</li> <li>2. Click update work status</li> <li>3. Update the status of cleared areas.</li> </ol>
<b>Expected Result:</b>	If the pile is cleared then allow the user to clear the work status.
<b>Actual Result:</b>	Work status will be updated successfully.
<b>Status:</b>	Pass.

### TC-12

**Table 55 : Test Case 12**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC12
<b>Related UC/FR:</b>	UC_12
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	To Know which area gets maximum numbers of task alerts.

<b>Pre-Req:</b>	Admin should be login to admin panel
<b>Test Data:</b>	List of alerts.
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login to application</li> <li>2. Click of alerts notification icon</li> <li>3. Check the maximum number of alerts.</li> </ol>
<b>Expected Result:</b>	Get the maximum alerting area.
<b>Actual Result:</b>	gets maximum numbers of task alerts successfully
<b>Status:</b>	Pass

### TC-13

**Table 56 : Test Case 13**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC13
<b>Related UC/FR:</b>	UC_13
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	To Know which area gets minimum numbers of task alerts.
<b>Pre-Req:</b>	Admin should be login to admin panel
<b>Test Data:</b>	List of alerts.
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login to application</li> <li>2. Click of alerts notification icon</li> <li>3. Check the maximum number of alerts.</li> </ol>
<b>Expected Result:</b>	Get the minimum alerted area.
<b>Actual Result:</b>	gets minimum numbers of task alerts
<b>Status:</b>	Pass

**TC-14****Table 57 : Test Case 14**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC14
<b>Related UC/FR:</b>	UC_14
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	Show the Status list of clear status
<b>Pre-Req:</b>	Admin should be login to admin panel & open specific page
<b>Test Data:</b>	List of clear status areas.
<b>Steps:</b>	<ul style="list-style-type: none"> <li>• Login to application</li> <li>• Click on clear status.</li> <li>• Check the list of cleared status areas.</li> </ul>
<b>Expected Result:</b>	If a specific area is cleared then the list shows the cleared areas list.
<b>Actual Result:</b>	Cleared status list get successfully
<b>Status:</b>	Pass

**TC-15****Table 58 : Test Case 15**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC15
<b>Related UC/FR:</b>	UC_15
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	After completing work & clearing status the status on the admin panel should be updated
<b>Pre-Req:</b>	User should click the update status button

<b>Test Data:</b>	Updated status list
<b>Steps:</b>	<ul style="list-style-type: none"> <li>• Login to application</li> <li>• Click on the updated status.</li> </ul>
<b>Expected Result:</b>	The work status should be updated on the admin panel.
<b>Actual Result:</b>	Status should be updated successfully.
<b>Status:</b>	Pass

### TC-16

**Table 59 : Test Case 16**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC16
<b>Related UC/FR:</b>	UC_16
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	Announce the Employee of the Month
<b>Pre-Req:</b>	
<b>Test Data:</b>	List of allotted tasks
<b>Steps:</b>	<ul style="list-style-type: none"> <li>• Login to application</li> <li>• Click on the employee of month</li> </ul>
<b>Expected Result:</b>	The employee of the month will be announced.
<b>Actual Result:</b>	Employee of the month announced successfully.
<b>Status:</b>	Pass

**TC-17****Table 60 : Test Case 17**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC17
<b>Related UC/FR:</b>	UC_17
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	Announce the Employee of the Year at the end of the Year
<b>Pre-Req:</b>	Checking the Minimum Number of Tasks, A lot to the Employee
<b>Test Data:</b>	List of allotted tasks
<b>Steps:</b>	<ul style="list-style-type: none"> <li>• Login to application</li> <li>• Click on the employee of year</li> </ul>
<b>Expected Result:</b>	The employee of the year will be announced.
<b>Actual Result:</b>	employee of the year announced unsuccessfully.
<b>Status:</b>	Fail

**TC-18****Table 61 : Test Case 18**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC18
<b>Related UC/FR:</b>	UC_18
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	See User Details
<b>Pre-Req:</b>	Login to admin panel
<b>Test Data:</b>	List of users

<b>Steps:</b>	1. Login to admin panel 2. Click on user details 3. See list of registered users
<b>Expected Result:</b>	If the user is registered then the list of users will be shown.
<b>Actual Result:</b>	The list of users shown successfully
<b>Status:</b>	Pass

## TC-19

**Table 62 : Test Case 19**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC19
<b>Related UC/FR:</b>	UC_19
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	Add Area supervisor to the system.
<b>Pre-Req:</b>	Supervisor Should be Login
<b>Test Data:</b>	Area Supervisor Details
<b>Steps:</b>	1. Login to admin panel 2. Click on Add area supervisor 3. Enter the required credentials 4. Click on submit.
<b>Expected Result:</b>	If credentials already exist redirect to same page with error message otherwise if credentials are valid then register the new area supervisor.
<b>Actual Result:</b>	The area supervisor will be added successfully.
<b>Status:</b>	Pass

**TC-20****Table 63 : Test Case 20**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC20
<b>Related UC/FR:</b>	UC_20
<b>Date:</b>	5-3-2022
<b>Purpose:</b>	Allow Supervisor to Remove Area Supervisor
<b>Pre-Req:</b>	Supervisor Should be Login to admin panel
<b>Test Data:</b>	Area Supervisor Id
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login to admin panel.</li> <li>2. Enter Employee Id of Area supervisor to be removed</li> <li>3. Click on remove Area supervisor</li> </ol>
<b>Expected Result:</b>	If the Area supervisor Id exists then the employee will be removed.
<b>Actual Result:</b>	The area supervisor will be removed successfully.
<b>Status:</b>	Pass

**TC-21****Table 64 : Test Case 21**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC21
<b>Related UC/FR:</b>	UC_21
<b>Date:</b>	12-3-2022
<b>Purpose:</b>	Update Area Supervisor
<b>Pre-Req:</b>	Supervisor Should be Login

<b>Test Data:</b>	Area supervisor id
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login to admin panel.</li> <li>2. Enter Id of Area supervisor to be updated</li> <li>3. Click on update Area Supervisor</li> </ol>
<b>Expected Result:</b>	If the area supervisor id exists then their supervisor details will be updated.
<b>Actual Result:</b>	The Area Supervisor details will not be updated successfully.
<b>Status:</b>	Fail

## TC-22

**Table 65 : Test Case 22**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC22
<b>Related UC/FR:</b>	UC_22
<b>Date:</b>	12-3-2022
<b>Purpose:</b>	View the list of Area Supervisor
<b>Pre-Req:</b>	Login to the Admin Panel
<b>Test Data:</b>	List of Area supervisors
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login to admin panel.</li> <li>2. Click on see list of Area Supervisors.</li> </ol>
<b>Expected Result:</b>	The list of area supervisors will be shown.
<b>Actual Result:</b>	The List of Area Supervisors will be shown successfully.
<b>Status:</b>	Pass

TC-23

**Table 66 : Test Case 23**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC23
<b>Related UC/FR:</b>	UC_23
<b>Date:</b>	12-3-2022
<b>Purpose:</b>	Allow Supervisor to add Employee
<b>Pre-Req:</b>	Login to the Admin Panel
<b>Test Data:</b>	Employee Details
<b>Steps:</b>	<ol style="list-style-type: none"><li>1. Login to admin panel</li><li>2. Click on Add Employee</li><li>3. Enter the required credentials</li><li>4. Click on submit.</li></ol>
<b>Expected Result:</b>	The new Employee will be added.
<b>Actual Result:</b>	The Employee will be added successfully
<b>Status:</b>	Pass

TC-24

**Table 67 : Test Case 24**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC24
<b>Related UC/FR:</b>	UC_24
<b>Date:</b>	12-3-2022
<b>Purpose:</b>	Allow Supervisor to Remove Employee
<b>Pre-Req:</b>	Login to Admin Panel

<b>Test Data:</b>	Employee Id
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login to admin panel.</li> <li>2. Enter Id of employee to be removed</li> <li>3. Click on Remove Employee</li> </ol>
<b>Expected Result:</b>	If the Employee Id exists then the employee will be removed.
<b>Actual Result:</b>	The Employee will be removed successfully
<b>Status:</b>	Pass

## TC-25

**Table 68 : Test Case 25**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC25
<b>Related UC/FR:</b>	UC_25
<b>Date:</b>	12-3-2022
<b>Purpose:</b>	Allow Supervisor to Update Employee
<b>Pre-Req:</b>	Login into Admin Panel
<b>Test Data:</b>	Employee Details
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login to admin panel.</li> <li>2. Enter the Id of Employee To be Updated</li> <li>3. Click on Update Employee</li> </ol>
<b>Expected Result:</b>	The Employee details will be updated.
<b>Actual Result:</b>	If the Employee Id exists then the employee details will be updated
<b>Status:</b>	Pass

**TC-26****Table 69 : Test Case 26**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC26
<b>Related UC/FR:</b>	UC_26
<b>Date:</b>	12-3-2022
<b>Purpose:</b>	Allow Supervisor to View Employees
<b>Pre-Req:</b>	Login into Admin Panel
<b>Test Data:</b>	List of employees
<b>Steps:</b>	<ol style="list-style-type: none"> <li>1. Login to admin panel.</li> <li>2. Click on see list of Employees.</li> </ol>
<b>Expected Result:</b>	The List of Employees will be shown.
<b>Actual Result:</b>	The List of employees will be shown successfully.
<b>Status:</b>	Pass

**TC-27****Table 70 : Test Case 27**

<b>Test Engineer</b>	Abdul Wahab
<b>Test Case ID:</b>	TC27
<b>Related UC/FR:</b>	UC_27
<b>Date:</b>	12-3-2022
<b>Purpose:</b>	Allow User to Logout
<b>Pre-Req:</b>	Login into Admin Panel
<b>Test Data:</b>	Login User Session

<b>Steps:</b>	1. Login to admin panel. 2. Click on Logout
<b>Expected Result:</b>	The User will be Logout.
<b>Actual Result:</b>	The User will Logout successfully.
<b>Status:</b>	Pass

### 5.2.2. White Box Testing

White box testing is software testing technique in which internal structure, design, and coding of software are tested to verify the flow of input-output and to improve design, usability, and security. In white-box testing, code is visible to testers so it is also called Clear box testing, Open box testing, transparent box testing, Code-based testing, and Glass box testing. The tester should know programming language/coding. There are many types of white box testing in which we perform a few of them which are as follows.

- Branch Testing (Web Application)
- Unit Testing (Web Application using selenium & Django itself library)
- Inspection (Mobile Application)

#### 5.2.2.1 Branch Testing

Branch testing is a type of white-box testing that is used to test every possible branch in the control flow graph of a program. In branch testing, every branch in the code is executed at least once.

### Login

**Table 71 : Branch Testing Login Module**

#	Conditions	Input/Test Data	Expected result
1	(Email==Auth()) && (Email!=" ")&& (Password!=" ")	Email=hammad@gmail.com	Login successfully
2	(Password==Auth()) && (Password!=" ")&& (Email!=" ")	Password=123	Login successfully
3	(Email==Auth()) && (Password!=Auth())	Email=wm90402@gmail.com Password=456	Login failed
4	(Email!=Auth()) && (Password==Auth())	Email=wm402@gmail.com Password=123	Login failed

## Register

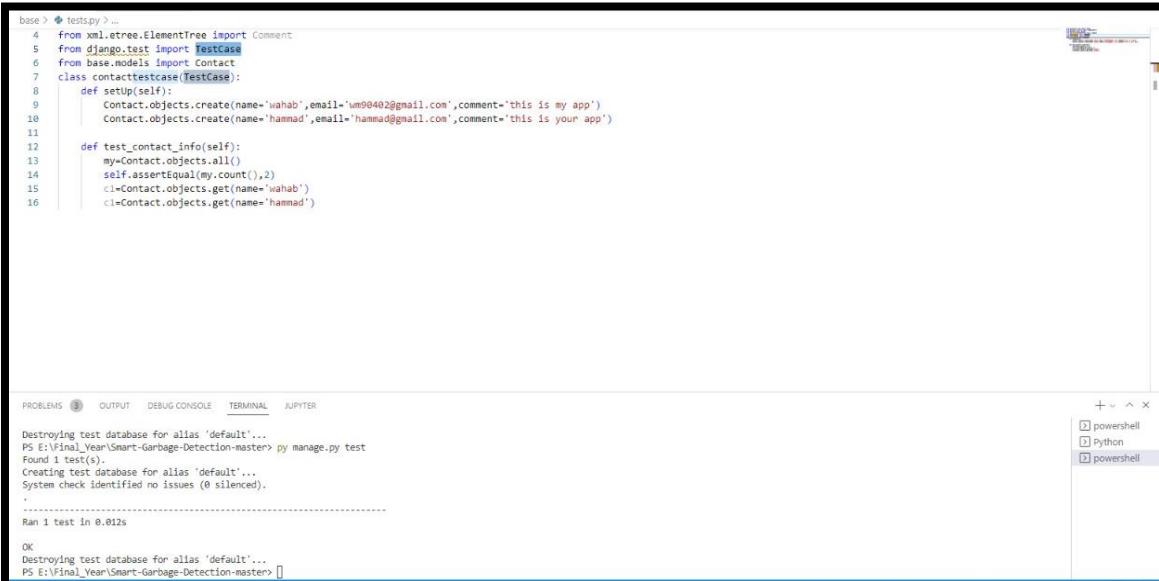
Table 72 : Branch Testing Register Module

#	Conditions	Input/Test Data	Expected result
1	(Email!=Auth()) && (Email!=" ") &&(Username!=" ")&&(password!=" ")	Email=ahmed402@gmail.com Password=1234 username=Ali	Registered successfully
2	(Password!=Auth()) && (Password!=" ")&&(Username!=" ")&&(Email!=" ")	Email=ahmed402@gmail.com Password=1234 username=Ahmed	Registered successfully
3	(Username!=Auth()) && (Username!=" ")&&(Password!=" ")&&(Email!=" ")	Email=wm90402@gmail.com Password=456 username=Arslan	Registered successfully
4	(Username==Auth())    (Username==Auth()) &&(Password==Auth())&&(Email==Auth())	Email=wm90402@gmail.com Password=456 username=Arslan	Not Registered

### 5.2.2.2 Unit Testing

It's often first type of testing. It's performed on each unit or block of code as it developed. It's specially done by the programmer.

#### Unit Testing: 1



```

base > tests.py ...
 4  from xml.etree.ElementTree import Comment
 5  from django.test import TestCase
 6  from base.models import Contact
 7  class contactTestCase(TestCase):
 8      def setUp(self):
 9          Contact.objects.create(name='wahab',email='wm90402@gmail.com',comment='this is my app')
10         Contact.objects.create(name='hammad',email='hammad@gmail.com',comment='this is your app')
11
12     def test_contact_info(self):
13         myContact.objects.all()
14         self.assertEqual(myContact.count(),2)
15         c1=Contact.objects.get(name='wahab')
16         c2=Contact.objects.get(name='hammad')

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Destroying test database for alias 'default'...  
PS E:\Final\_Year\Smart-Garbage-Detection-master> py manage.py test  
Found 1 test(s).  
Creating test database for alias 'default'...  
System check identified no issues (0 silenced).

-----  
Ran 1 test in 0.012s  
OK  
Destroying test database for alias 'default'...  
PS E:\Final\_Year\Smart-Garbage-Detection-master> []

Figure 49 : Contact Us Test Pass

## Unit Testing: 2



The screenshot shows a Jupyter Notebook interface with the following details:

- Code Editor:** The code in `models.py` is as follows:

```
base > tests.py 1 x ④ models.py 2
base > tests.py ④ contacttestcase > ④ test_contact_info
4  from xml.etree.ElementTree import Comment
5  from django.test import TestCase
6  from base.models import Contact
7  class contacttestcase(TestCase):
8      def setUp(self):
9          Contact.objects.create(name='wahab',email='wm00402@gmail.com',comment='this is my app')
10         Contact.objects.create(name='hammad',email='hammad@gmail.com',comment='this is your app')
11
12     def test_contact_info(self):
13         myContact.objects.all()
14         self.assertEqual(my.count(),2)
15         c1=Contact.objects.get(name='all')
16         c1=Contact.objects.get(name='hammad')
```

- Terminal:** The terminal output shows the execution of the test and its failure:

```
File "C:\Users\WAHAB_MUGHAL_DESKTOP-IQB3B84\000\AppData\Roaming\Python\Python39\site-packages\django\db\models\manager.py", line 85, in manager_method
    return getattr(self.get_queryset(), name)(*args, **kwargs)
File "C:\Users\WAHAB_MUGHAL_DESKTOP-IQB3B84\000\AppData\Roaming\Python\Python39\site-packages\django\db\models\query.py", line 496, in get
    raise self.model.DoesNotExist(
base.models.Contact.DoesNotExist: Contact matching query does not exist.
```

- Output:** The output shows the test ran in 0.020s and failed with an error.
- PROBLEMS:** A list of errors is shown, all related to the `powershell` extension.
- OUTPUT:** The output shows the test ran in 0.020s and failed with an error.
- DEBUG CONSOLE:** The debug console shows the command `PS E:\Final_Year\Smart-Garbage-Detection-master>`.
- TERMINAL:** The terminal shows the command `tests.py 1 x ④ models.py 2`.
- JUPYTER:** The Jupyter interface is visible at the bottom.

**Figure 50 : Contact Us Test Fail (Wrong Credentials)**

## Unit Testing: 3



```
base > tests.py 1 x  models.py 2
base > tests.py > BlogTestCase > test_contact_info
5  from xml.etree.ElementTree import Comment
6  from django.test import TestCase
7  from base.models import Blog, Contact
8
9  class BlogTestCase(TestCase):
10     def setUp(self):
11         Blog.objects.create(title='mybook', postDetails='it is posted on sunday')
12         Blog.objects.create(title='social media ', postDetails='it is posted on monday')
13
14     def test_contact_info(self):
15         myBlog.objects.all()
16         self.assertEqual(myBlog.count(), 2)
17         self.assertEqual(Blog.objects.get(title='mybook', postDetails='it is posted on sunday')
18                         .postDetails, 'it is posted on sunday')
19         self.assertEqual(Blog.objects.get(title='social media ', postDetails='it is posted on monday')
20                         .postDetails, 'it is posted on monday')

PROBLEMS 0 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

File "C:\Users\WAHAB_MUGHAL\Desktop-1\Q83BP4.000\AppData\Roaming\Python\Python39\site-packages\django\db\backends\utils.py", line 89, in _execute
    return self.cursor.execute(sql, params)
File "C:\Users\WAHAB_MUGHAL\Desktop-1\Q83BP4.000\AppData\Roaming\Python\Python39\site-packages\django\db\backends\sqlite3\base.py", line 477, in execute
    return Database.Cursor.execute(self, query, params)
django.db.utils.OperationalError: no such table: base_blog

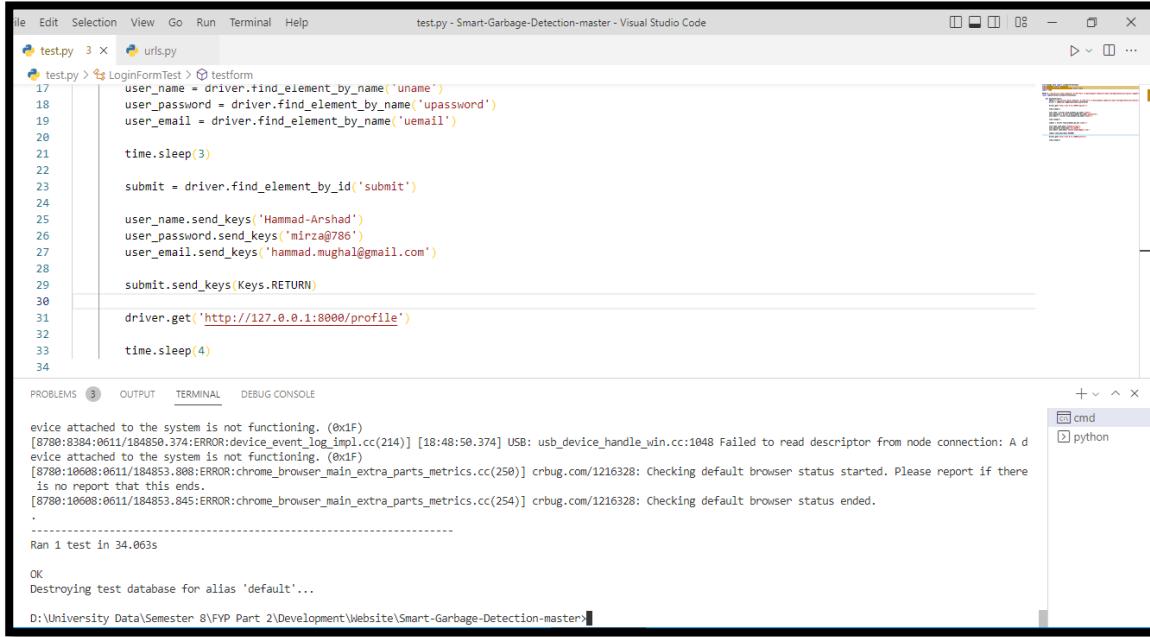
Ran 1 test in 0.014s

FAILED (errors=1)
Destroying test database for alias 'default'...
PS E:\Final_Year\Smart-Garbage-Detection-master |
```

**Figure 51 : Blog Test Fail (Wrong Credentials)**

### 5.2.2.3 Unit Testing using selenium

#### Unit Testing: 4



test.py 3 x urls.py

```
17 user_name = driver.find_element_by_name('uname')
18 user_password = driver.find_element_by_name('upassword')
19 user_email = driver.find_element_by_name('uemail')
20
21 time.sleep(3)
22
23 submit = driver.find_element_by_id('submit')
24
25 user_name.send_keys('Hammad-Arshad')
26 user_password.send_keys('mirza@786')
27 user_email.send_keys('hammad.mughal@gmail.com')
28
29 submit.send_keys(Keys.RETURN)
30
31 driver.get('http://127.0.0.1:8000/profile')
32
33 time.sleep(4)
34
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

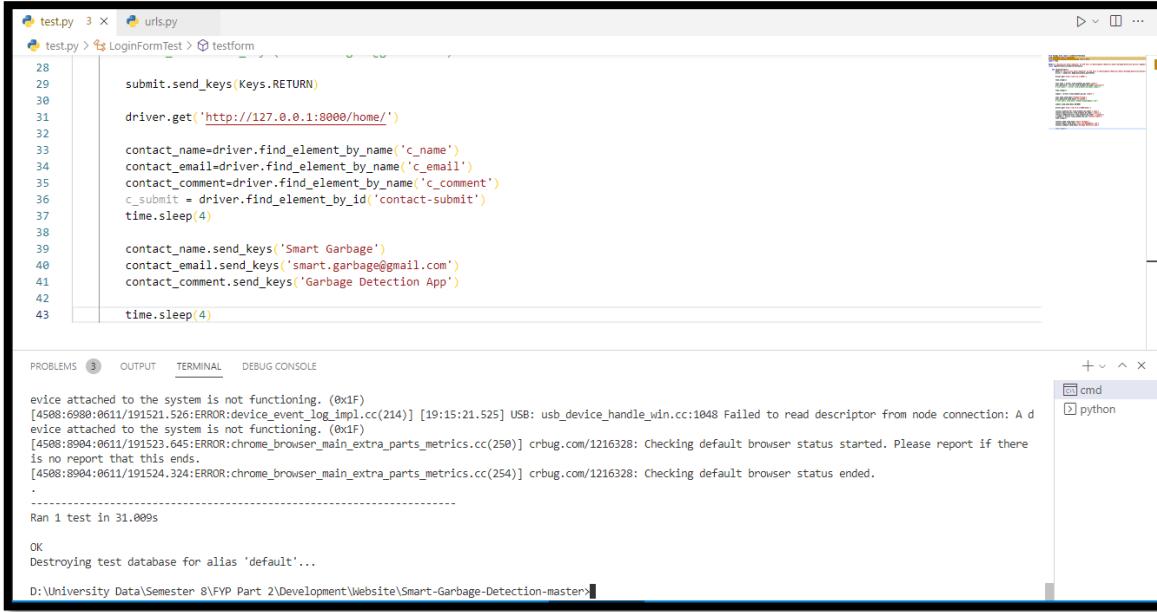
```
device attached to the system is not functioning. (0x1F)
[8780:8384:0611/184850.374:ERROR:device_event_logimpl.cc(214)] [18:48:50.374] USB: usb_device_handle_win.cc:1048 Failed to read descriptor from node connection: A device attached to the system is not functioning. (0x1F)
[8780:10608:0611/184853.808:ERROR:chrome_browser_main_extra_parts_metrics.cc(250)] crbug.com/1216328: Checking default browser status started. Please report if there is no report that this ends.
[8780:10608:0611/184853.845:ERROR:chrome_browser_main_extra_parts_metrics.cc(254)] crbug.com/1216328: Checking default browser status ended.
.
-----
Ran 1 test in 34.063s

OK
Destroying test database for alias 'default'...
```

D:\University Data\Semester 8\FYP Part 2\Development\Website\Smart-Garbage-Detection-master>

Figure 52 : Selenium to Test Register

#### Unit Testing: 5



test.py 3 x urls.py

```
28 submit.send_keys(Keys.RETURN)
29
30 driver.get('http://127.0.0.1:8000/home/')
31
32 contact_name=driver.find_element_by_name('c_name')
33 contact_email=driver.find_element_by_name('c_email')
34 contact_comment=driver.find_element_by_name('c_comment')
35 c_submit = driver.find_element_by_id('contact-submit')
36
37 time.sleep(4)
38
39 contact_name.send_keys('Smart Garbage')
40 contact_email.send_keys('smart.garbage@gmail.com')
41 contact_comment.send_keys('Garbage Detection App')
42
43 time.sleep(4)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

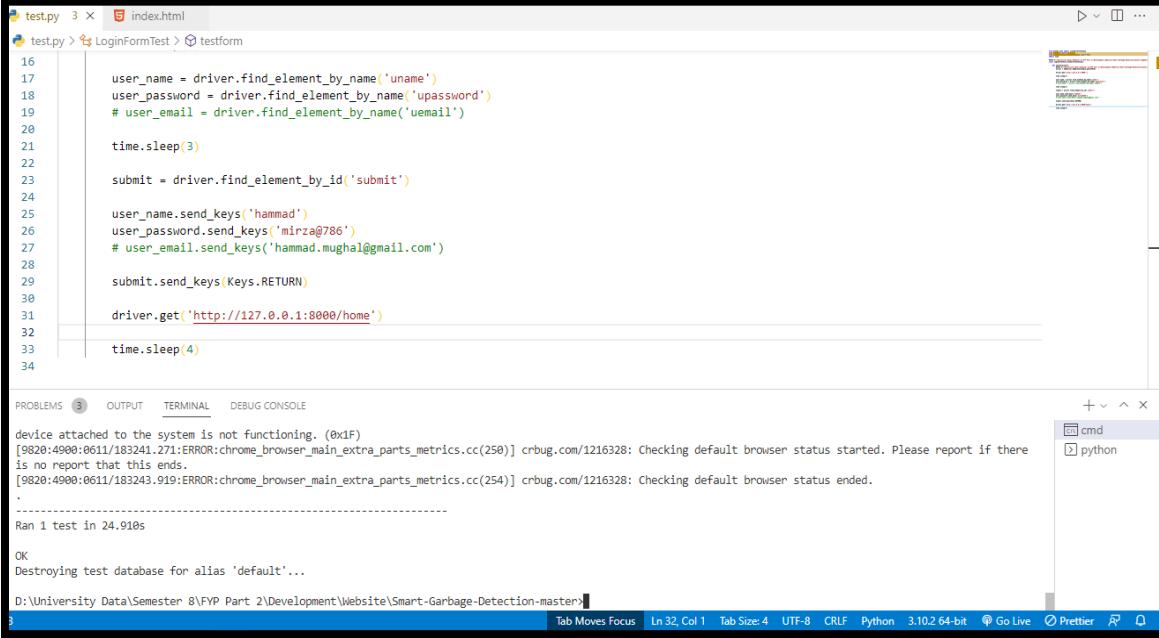
```
device attached to the system is not functioning. (0x1F)
[4508:6900:0611/191521.526:ERROR:device_event_logimpl.cc(214)] [19:15:21.525] USB: usb_device_handle_win.cc:1048 Failed to read descriptor from node connection: A device attached to the system is not functioning. (0x1F)
[4508:6904:0611/191523.645:ERROR:chrome_browser_main_extra_parts_metrics.cc(250)] crbug.com/1216328: Checking default browser status started. Please report if there is no report that this ends.
[4508:6904:0611/191524.324:ERROR:chrome_browser_main_extra_parts_metrics.cc(254)] crbug.com/1216328: Checking default browser status ended.
.
-----
Ran 1 test in 31.009s

OK
Destroying test database for alias 'default'...
```

D:\University Data\Semester 8\FYP Part 2\Development\Website\Smart-Garbage-Detection-master>

Figure 53 : Selenium to Test Contact

## Unit Testing: 6



```
test.py 3 x index.html
test.py > testform
16
17 user_name = driver.find_element_by_name('uname')
18 user_password = driver.find_element_by_name('upassword')
19 # user_email = driver.find_element_by_name('uemail')
20
21 time.sleep(3)
22
23 submit = driver.find_element_by_id('submit')
24
25 user_name.send_keys('hammad')
26 user_password.send_keys('mirza@786')
27 # user_email.send_keys('hammad.mughal@gmail.com')
28
29 submit.send_keys(Keys.RETURN)
30
31 driver.get('http://127.0.0.1:8000/home')
32
33 time.sleep(4)
34
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
device attached to the system is not functioning. (0x1f)
[9820:4900:0611/183241.271:ERROR:chrome_browser_main_extra_parts_metrics.cc(250)] crbug.com/1216328: Checking default browser status started. Please report if there is no report that this ends.
[9820:4900:0611/183243.919:ERROR:chrome_browser_main_extra_parts_metrics.cc(254)] crbug.com/1216328: Checking default browser status ended.

Ran 1 test in 24.918s

OK
Destroying test database for alias 'default'...
```

D:\University Data\Semester 8\FYP Part 2\Development\Website\Smart-Garbage-Detection-master\

Tab Moves Focus Ln 32, Col 1 Tab Size: 4 UTF-8 CRLF Python 3.10.2 64-bit Go Live Prettier

Figure 54 : Selenium to Test Login

### 5.2.2.3 Inspection

Appium inspector is used to inspect the mobile app to check everything is working. Inspector is use to inspect every process line by line to review the code & see if there any error or bug available. We perform inspection on our mobile application on almost all the major required modules.

## Inspection: 1

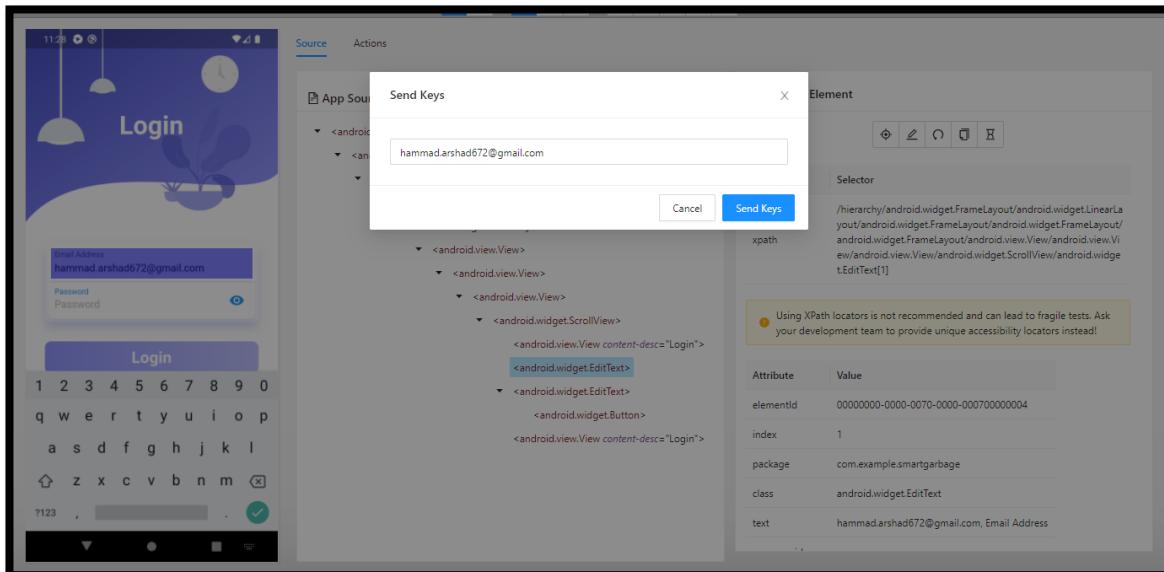


Figure 55 : Inspection of Login Screen (Email Key)

## Inspection: 2

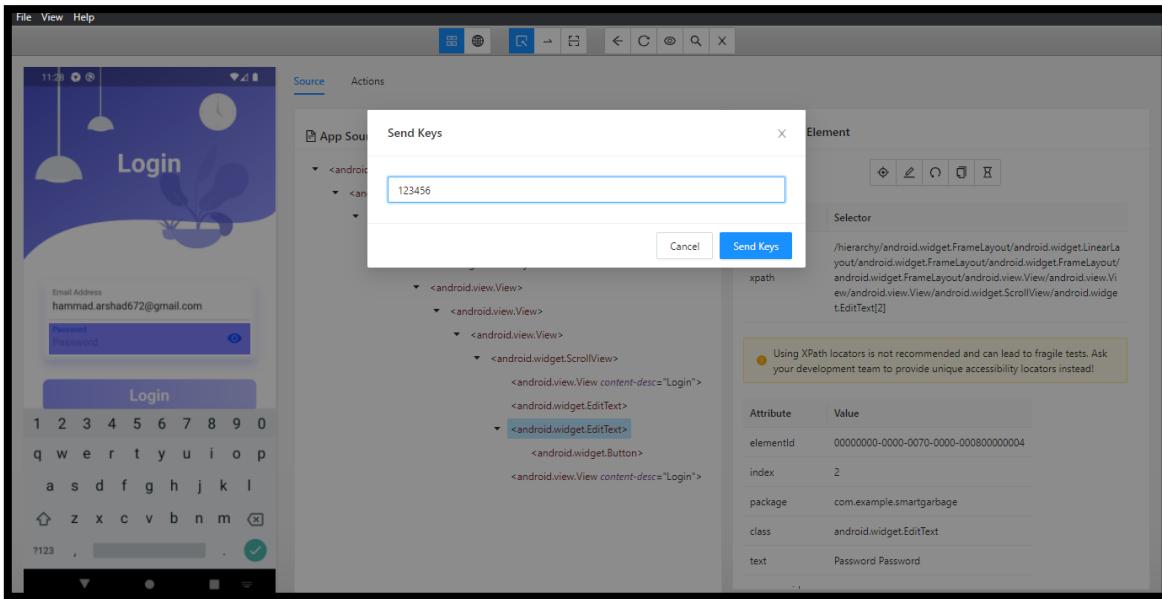


Figure 56 : Inspection of Login Screen (Password Key)

## Inspection: 3

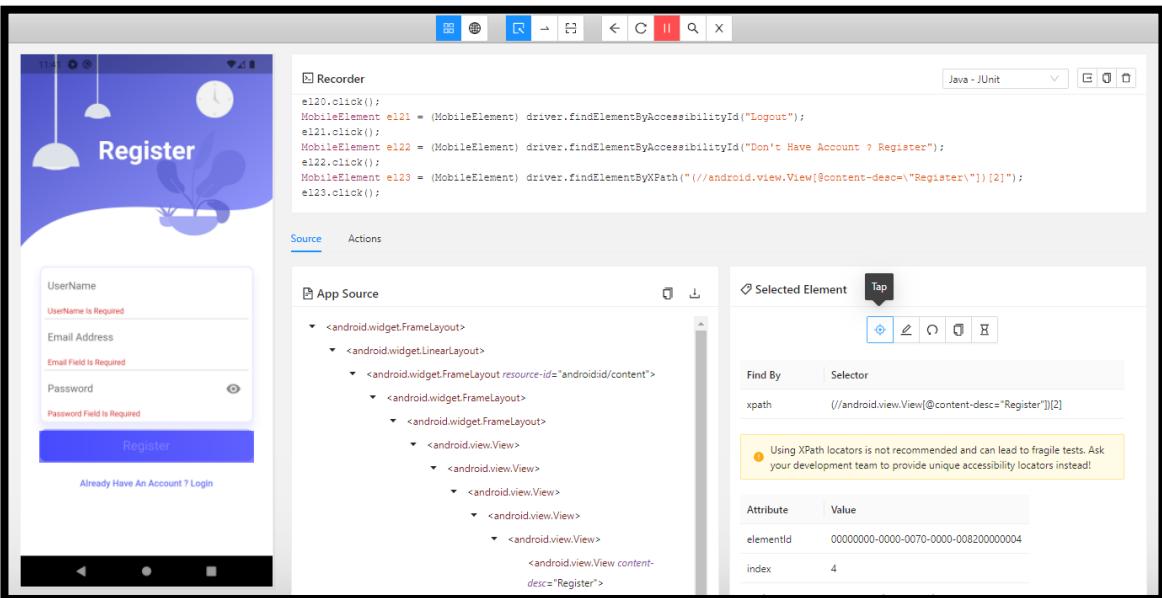


Figure 57 : Inspection of Register Screen (Empty Fields)

## Inspection: 4

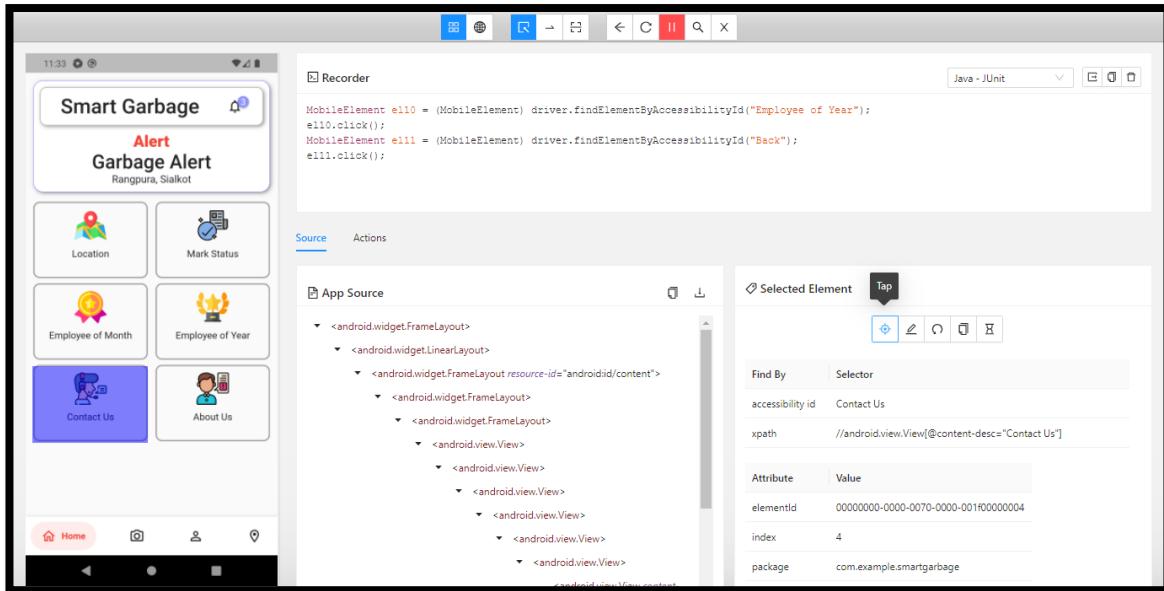


Figure 58 : Inspection of Home Screen

## Inspection: 5

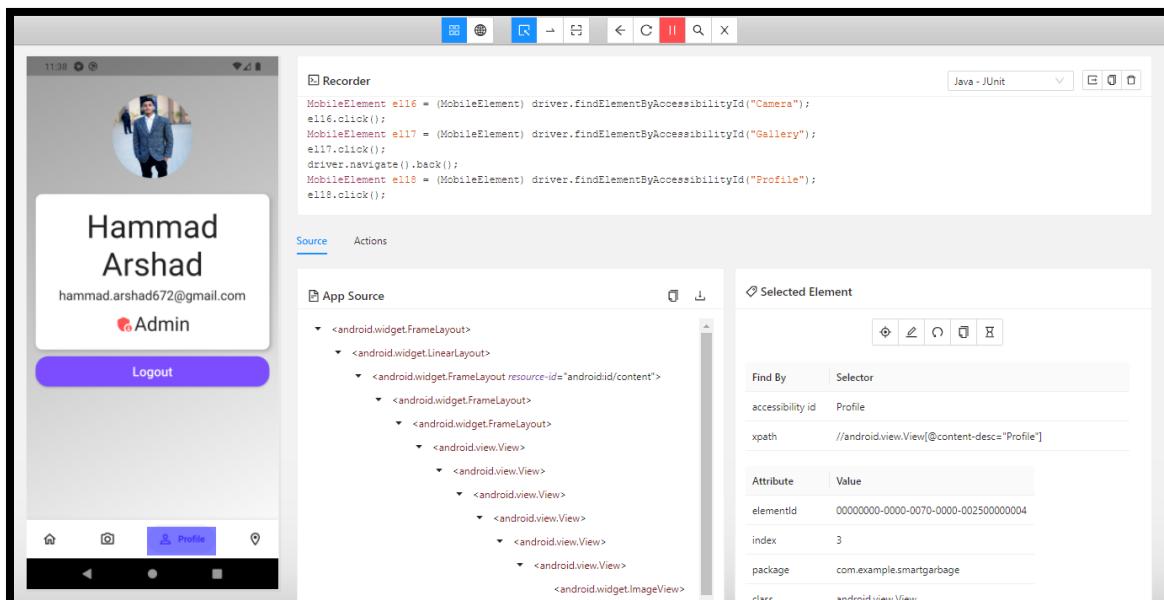


Figure 59 : Inspection of Bottom Navigation & Profile Screen

## Inspection: 6

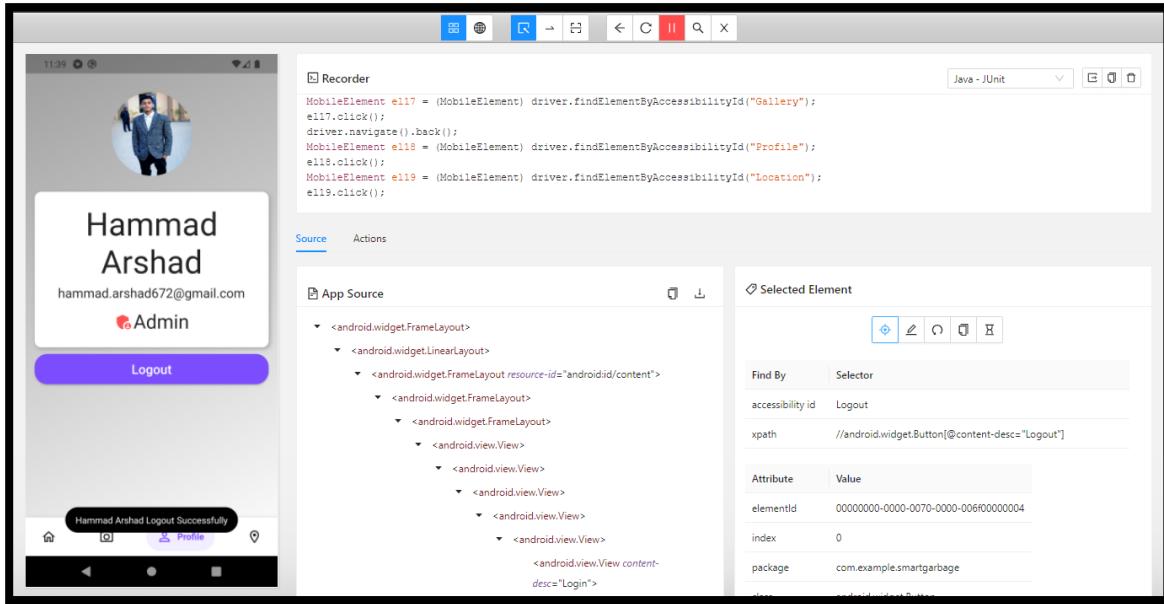


Figure 60 : Inspection of Logout

## Inspection: 7

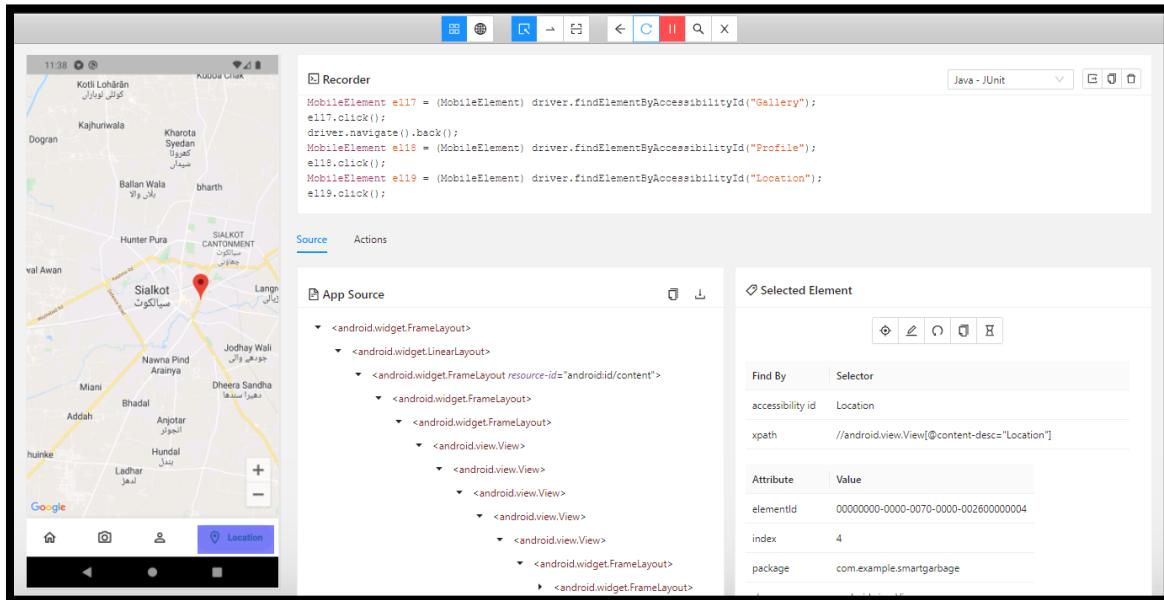


Figure 61 : Inspection of Map Screen