DESIGN PROJECT REPORT

on

IOT BASED SMART ROOM

Submitted by

STEPHY JOHN TOC16CS054

RANJANA RAJU TOC16CS044

ELSA THOMAS TOC16CS022

In partial fulfillment of the requirements for the award of the degree of
Bachelor of Technology
In

COMPUTER SCIENCE AND ENGINEERING

Of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

November 2018

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Toc H INSTITUTE OF SCIENCE & TECHNOLOGY
Arakkunnam P.O, Ernakulam District, Kerala – 682 313

Toc H INSTITUTE OF SCIENCE & TECHNOLOGY

Arakkunnam P.O, Ernakulam District, Kerala – 682 313

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the design project titled “IOT BASED SMART ROOM” **submitted by** STEPHY JOHN (TOC16CS054),RANJANA RAJU (TOC16CS044),ELSA THOMAS (TOC16CS022) **of Semester V is a bonafide account of the work done by them/him/her under our supervision, during the academic year 2018-‘19.**

PROJECT GUIDE **HEAD OF THE DEPARTMENT**

HEAD OF THE INSTITUTION

ACKNOWLEDGEMENT

It is our proud privilege and duty to acknowledge the kind of help and guidance received from several people in preparation of this report. It would not have been possible to prepare this report in this form without their valuable help, cooperation and guidance.

First and foremost, we wish to record our sincere gratitude to **Management of this college** and to our beloved **Principal, Prof. Dr. Preethi Thekkath,** Principal, Toc-H Institute of Science and Technology, Arakkunnam, for her constant support and encouragement in preparation of this report and for making available library and laboratory facilities needed to prepare this report. Our sincere thanks to **Prof. Dr. R. Ravindran Nair**, Dean, Academics and Student Affairs for his valuable support and encouragement in preparation of this report.

Our sincere thanks to **Asso. Prof. Sreela Sreedhar**, Head of the Department of Computer Science & Engineering, for her valuable suggestions and guidance throughout the period of this report. Our sincere thanks to project coordinators **Asst. Prof. Mithu Mary George, Asst. Prof. Mima Manual** and **Asst. Prof. Anuraj C.K** for having supported the work related to our project. Our sincere thanks to project guide **Asst. Prof. Jesna Anver** for her valuable suggestions and guidance.Their contributions and support in preparing this report are greatly acknowledged.

We wish to thank our parents for financing our studies in this college as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

ABSTRACT

In the recent decades urbanization has increased tremendously. This lead to more commercial buildings & houses which means consumption of more electricity by negligent use by consumers, less security of products and people, inefficient food waste and water management and difficulty in irrigation. In the proposed 'IOT based Smart Room' system there is a mobile application with many features. The proposed system aims to solve the mentioned problems. Smart Room includes a complete automatic security system by door locking/unlocking through face recognition technology. Smart Room ensures a fully automatic lighting facility using sensors for corridor, staircase and garage. Some of the main Features of this system is performed through a Smart room application. This app displays the humidity & temperature values of a room. It contains door lock/unlock buttons and bulb on/off buttons. Sensors to measure soil conditions based on which a proper Wi-Fi enabled irrigation possible through pump on/off buttons. A fully functioning automatic water storing and monitoring facility is available. A smart waste management solution through IOT fill – level check sensors is built. It also includes an efficient route tracking technique for the waste containers requiring services. Data readings from different sensors with an alert receiving and sending facility is proposed. From this project, we aim to solve problems faced in buildings considering user convenience and easiness.

TABLE OF CONTENTS

Chapter	Contents	Page No.
1.0	**INTRODUCTION**	**1**
	1.1 BACKGROUND	3
2.0	**ANALYSIS AND STUDY OF EXISTING SYSTEMS**	**4**
	2.1 BUILDTRACK HOME AUTOMATION	5
	2.2 CURIOUSFLY	6
	2.3 DRIP IRRIGATION	7
	2.4 SMART BIN	8
3.0	**PROBLEM IDENTIFICATION AND OBJECTIVES OF THE PROJECT**	**10**
	3.1 PROBLEM STATEMENT	11
	3.2 OBJECTIVES OF THE SYSTEM	11
4.0	**SYSTEM DESIGN**	**13**
	4.1 ARCHITECURE	
	4.1.1 HARDWARE COMPONENTS	14
	4.1.2 CLOUD SERVICES	14
	4.1.3 MOBILE APPLICATION	15
	4.2 DETAILED DESIGN	15
	4.2.1 MODULE DESCRIPTION	15
	4.2.1.1 SMART ROOM	15
	4.2.1.2 SMART BIN	15
	4.2.1.3 GARDEN/FARM	15
	4.3 USECASES	18
	4.3.1 USECASE DIAGRAM	18
	4.3.2 LOGIN	19
	4.3.3 CONTROL HOME APPLIANCES	20
	4.3.4 VIEW TEMPERATURE & HUMIDITY VALUE	21
	4.3.5 DOOR LOCK\UNLOCK	21

4.3.6 VIEW LEVEL OF WASTE IN BIN 22

4.3.7 SEND ALERT & RECEIVE ALERT 23

4.3.8 VIEW MOISTURE CONTENT OF SOIL 24

4.3.9 CHECK AVAILABILITY OF WATER IN TANK 24

4.4 UI DESIGN 25

4.5 LOGIN 25

4.6 SIGN UP 26

4.7 HOME 26

4.8 SMART ROOM 27

4.9 SMART BIN 27

4.10 GARDEN/FARM 28

4.11 WATER LEVEL 28

4.5 DATABASE DESIGN 29

5.0 IMPLEMENTATION AND PRESENTATION OF DATA 30

5.1 HARDWARE REQUIREMENTS 31

5.2 SOFTWARE REQUIREMENTS 31

5.3 TESTING 32

5.3.1 UNIT TESTING 32

5.3.2 INTEGERATION TESTING 33

5.3.3 FUNCTIONAL TESTING 33

5.3.4 TEST CASES 34

5.1 LOGIN 34

5.2 SMART ROOM 35

5.3 GARDEN/FARM 36

5.4 SMART BIN 37

6.0 CONCLUSION 39

7.0 FUTURE WORKS SUGGESTED 41

8.0 REFERENCES 43

APPENDIX I – CODE SNIPPETS 46

APPENDIX II – SCREEN SHOTS 56

LIST OF FIGURES

Figure	Title	Page No.
Figure 4.1	Block Diagram	14
Figure 4.2	Class Diagram	16
Figure 4.4	Use case Diagram	18
Figure 4.5	Login	25
Figure 4.6	Sign Up	26
Figure 4.7	Home	26
Figure 4.8	Smart Room	27
Figure 4.9	Smart Bin	27
Figure 4.10	Garden/Farm	28
Figure 4.11	Water Level	28

LIST OF TABLES

Table	Title	Page No.
Table 4.3	Class Table	17
Table 4.12	User Table	29
Table 4.13	Owner Table	29
Table 5.1	Login	34
Table 5.1	Smart Room	35
Table 5.1	Garden/Farm	36
Table 5.1	Smart Bin	37

INTRODUCTION

1. INTRODUCTION

The Internet of things (IoT) is the network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these things to connect, collect and exchange data Recent advances in intelligent computer systems and communications have created the necessary conditions for the networking of a wide variety of heterogeneous devices. SMART ROOM is a simple but quite versatile home automation system which aims to solve problems like wastage of electricity due to negligent use by consumers, inefficient and time consuming irrigation for garden/farm and improper waste management seen in residential buildings. The proposed system consists of a mobile application that enables user to solve these issues.

1.1. BACKGROUND

The main feature of this technology is the integration of heterogeneous sensing and action elements (actuators) in a distributed system which performs different actions based on the information gathered by the sensors combined with the requirements of the particular application. Intelligent information systems enable the processing of multimodal data collected by the sensors, so as to reconcile heterogeneous information and safe conclusions on the facts giving rise to the activation of the necessary actions to address the consequences of these events. Moreover, the availability of new (smart) energy meters allows for real-time monitoring of energy consumption and provides a unique opportunity of using energy more efficiently.

There are various techniques for home automation such as IOT based home automation over the cloud, home automation under Wi-Fi through android apps from any smartphone, Arduino based home automation, home automation by android application based remote control, home automation using digital control. The proposed system is a home automation with additional features of like automatic irrigation and waste management.

ANALYSIS AND STUDY OF EXISTING SYSTEMS

2. ANALYSIS AND STUDY OF EXISTING SYSTEMS

.

2.1. BUILDTRACK HOME AUTOMATION

Build Track is a company working in Smart Automation and Internet of Things (IoT) solutions for Homes, Enterprises, Hotels, Hospitals, Senior Living and Warehouses.

FEATURES

Build track consists many modern features like CCTV cameras and Door Lock Safety and Security sensors like smoke, fire, Gas etc. Build track curtain/Blind control system .Build track panic buttons for elders. Motion sensors for energy savings. Product enables access and control through company's mobile applications. Switches to control lights, fans, curtains and local alarms and App notifications.

PROS

Offer solutions for Energy Efficiency that enable ZERO energy wastage. It guarantees productive management of Residential and Commercial Buildings and Campuses. It is more convenient to use. Proper monitoring and security. Offers both remote and voice enabled control. Customization-Allows choice of both wired and wireless deployment. User comfort is prior. It's quick and easy to use.

CONS

It does not offer any techniques for solving: - Food waste, water control and monitoring, irrigation of plants/garden. Another disadvantage is high cost. Complex technology which makes it difficult for common layman to understand. The user interface design of apps is not easy and simple in appearance which makes sometimes difficult for the customers to use**.**

2.2. CURIOUSFLY

Curiousfly is a home automation company in cochin.'Curiousfly’ smart switches make home automation simple with less complicated wiring and simple UI.

FEATURES

Curiousfly is a leader & game-changer in transforming the electrical switches with smart switches. Its "smart switches" does not require any CAT cabling or home automation DB.Curiousfly is accessed using mobile app which enables access to home appliances even with a normal wiring. Together with Amazon echo & Google home, it creates wonders with voice commands to daily habits. Provides easy integrations with gates, curtains, dimmable LED's, sprinklers & much more.

PROS

Curiousfly enables user to check status and control of home appliances from anywhere. A user can schedule appliances as per his routines. User can monitor the energy consumption of appliances. This system sets limit and get alerts. Set custom scenes according to user’s mood for the weekend party or a movie. Timer- Set timer for instant use of home appliances.App Sharing: Extend app control to family.

CONS

This system does not introduce any solutions related to irrigation of garden/farm. Being a home automation product facilities are confined inside the home only. It does not give solutions to issues like improper waste management .i.e. it does not offer any techniques for solving: -water storing and monitoring, irrigation of plants/garden, for the improper heap of waste. Another disadvantage is high cost. Complex technology which makes it difficult for common layman to understand.

2.3. DRIP IRRIGATION

Drip Irrigation is a type of micro-irrigation system also known as trickle irrigation that has the potential to save water and nutrients by allowing water to fall drop by drop just at the position of roots. Water is delivered slowly to the roots of plants, either from above the soil surface or buried below the surface. The goal is to place water directly into the root zone and minimize evaporation. Drip irrigation methods range from very high-tech and computerized to low-tech.

FEATURES

In drip irrigation systems, pump and valves manually operated by a controller. Most large drip irrigation systems employ some type of filters to prevent clogging of the small emitter flow path by small waterborne particles. Backwash controller is used to prevent backflow. Emitting devices like spray head is used. Regulations typically do not permit spraying water through the air that has not been fully treated to potable water standards.

PROS

Enhances seed germination as sufficient amount of water can be supplied**.** Drip irrigation may help achieve water conservation by reducing evaporation and deep drainage. Drip Irrigation is improved and better crop production*.* It is easy to control and cover whole area of the garden or field.

CONS

Drip Irrigation can be used for only specific crops which are closely planted like wheat. It cannot be used for plant germination unless distance between drips is short. It needs manual monitoring for checking water level in soil. Needs manual monitoring for checking water level for different crops.

2.4 SMART BIN

Smart Bin is a waste automation company which offers intelligent remote monitoring solutions for collectors of waste. Smart Bin has been making collection and distribution operations smarter, more efficient and less wasteful for many clients across the world. This company offers its facility for clients in Chicago and US.

FEATURES

This system contains an Ultrasonic fill-level check sensor. Geo-location & temperature measure facility. It is a completely wireless system with 3+ year's battery life. Weighing less than 9 ounces & easily mounted to any container. Zero maintenance with a non-corrosive protective shell. Measures up to 3m in depth.

PROS

User can always know the fill-level of waste containers. This system sends efficient routes directly to drivers. Eliminate over-filling and emergency call-outs. Monitors waste containers and plan optimized routes from anywhere. Smart Bin is the IoT game changer for collectors and distributors around the world.

CONS

This system mainly focuses the entire city not a single user or residential building. It is a foreign waste automation company so its facilities are not available in India. It focuses only metropolitan cities i.e. for small town this kind of system is not presently setup.

2.5 ANALYSIS

In **buildtrack** security through CCTV near doors enable users to record and see footage for viewing only i.e. it will not stop theft or crime when it is in progress and thus allow criminals to escape. In proposed system CCTV cameras will detect face and will lock/unlock door based on pattern matching and security alerts will also be send when needed**.**

Curiousfly does not offer any techniques for solving: - Food waste, water level monitoring while proposed system aims to overcome these limitation s using app alert system and sensors.Curiousfly uses more complex technologies but proposed system will not include complex technologies like Amazon echo. In **drip irrigation**, manual monitoring is needed for checking the moister level or water level in soil and after checking has to be manually operated. To overcome this problem we introduce an automatic irrigation system which will check the water level in soil automatically and if the soil is very dry an alert will be send. In drip irrigation, there are no techniques for identifying the water level in the water tank when it is low. But in proposed system when low level of water is identified. Water will be automatically filled by the proposed IOT system. **Smart bin** is an automatic waste collection service. But it is limited in metropolitan cities. In India Smart bin's service is not available. Proposed system we will introduce solutions to manage this problem by implementing. In this study, existing products in the chosen domain were analyzed. Design of new system will include all necessary features of existing system with needed modification. New system will be combination of all four products studied.

PROBLEM IDENTIFICATION & OBJECTIVES OF THE PROJECT

3. PROBLEM IDENTIFICATION & OBJECTIVES OF THE PROJECT

3.1. PROBLEM STATEMENT

In the recent decades urbanization has increased tremendously which lead to more commercial buildings & houses. This results in the following problems:

- Wastage of electricity due to negligent use by consumers.
- Inefficient and time consuming irrigation methods for garden/farm.
- Improper waste management at buildings.

3.2. OBJECTIVES OF THE SYSTEM

The proposed system presents an approach for smart automation using Internet of Things technology. This project aims to give the following solutions through mobile application

Smart Security System:

- Automatic and Semi-Automatic unlocking/locking facility based on face recognition.
- Alert sending facility

Energy Saving Solutions

- Sensor control lighting which can be used at front door, corridors, garage.
- Monitor data generated from analog sensors (humidity and temperature).
- Easy control of lights/devices through app.

Waste Management Solution:

- Iot fill – level check sensors and efficient route tracking technique for the containers requiring services.

Wi-Fi enabled Irrigation method:

- Sensor based moisture level test.
- Monitor data and activate pump through our ‘Smart Room’ app. Automatic water checking and filling.
- Alert receiving**.**

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1. ARCHITECURE

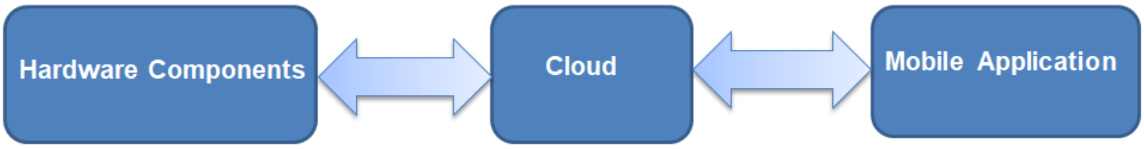


Figure 4.1-Block Diagram

4.1.1. Hardware Components

This includes all sensors and microcontroller chip used. Once the code is uploaded through the microcontroller chip. It will be stored in the memory based on which sensor will activate .Here data values from different sensors will be sensed and will be displayed in serial monitor. These data values will be uploaded to cloud for storing.

4.1.2. Cloud Services

Thingspeak and ubidots are the two cloud services used for uploading data values from sensors and downloading data values from mobile application. ThingSpeak is an open data platform for the Internet of Things. Sensors and application can communicate with ThingSpeak using a restful API.It can either keep data private, or make it public. In addition, Thing Speak can be used to analyze and act on data. This channel can be used to connect with mobile application for downloading data values. Ubidots is an Internet of Ubidots is used in the proposed system for setting conditions for certain data values from sensors based on which email, message and alert can be send to user.

4.1.3. Mobile Application

Mobile application consist of a login page and registration page. On successful login user will be directed to home page. This home page is divided into three modules. Smart room, Garden/Farm and Smart bin. Here smart room consist of features like measuring room conditions and controlling appliances at home. In second module garden/farm based on moisture level identified user can pump water through buttons in application. Automatic water level check in storage tank can be done through app. Third module is Smart bin which consist of text to display level of waste in waste container based on which alert can be send to municipality from the mobile application.

4.2. DETAILED DESIGN

4.2.1 MODULE DESCRIPTION

4.2.1.1 Smart Room

User can login and register through sign in and sign up buttons. They can turn on or off bulbs and monitor the state of the bulbs, can lock and unlock doors through servo motors and monitor whether the door is locked or unlocked. User get notified through email, if anyone tried to open the door with face image as the camera captures it. They can view temperature and humidity values of a room through the app. Security alert can be send to police if an intruder is identified.

4.2.1.2 Smart Bin

User can view the waste level with a notification of present status and can send alert to municipality if an emergency is occurred. Municipality will inform about the waste level and can collect waste before it fills.

4.2.1.3 Garden/Farm

User can view moisture level and present status of the soil. Based on that value user can turn on/off pump and water will flow through pipes. User is notified about automatic activation of motor to fill the tank.

4.2.2 CLASS DIAGRAM

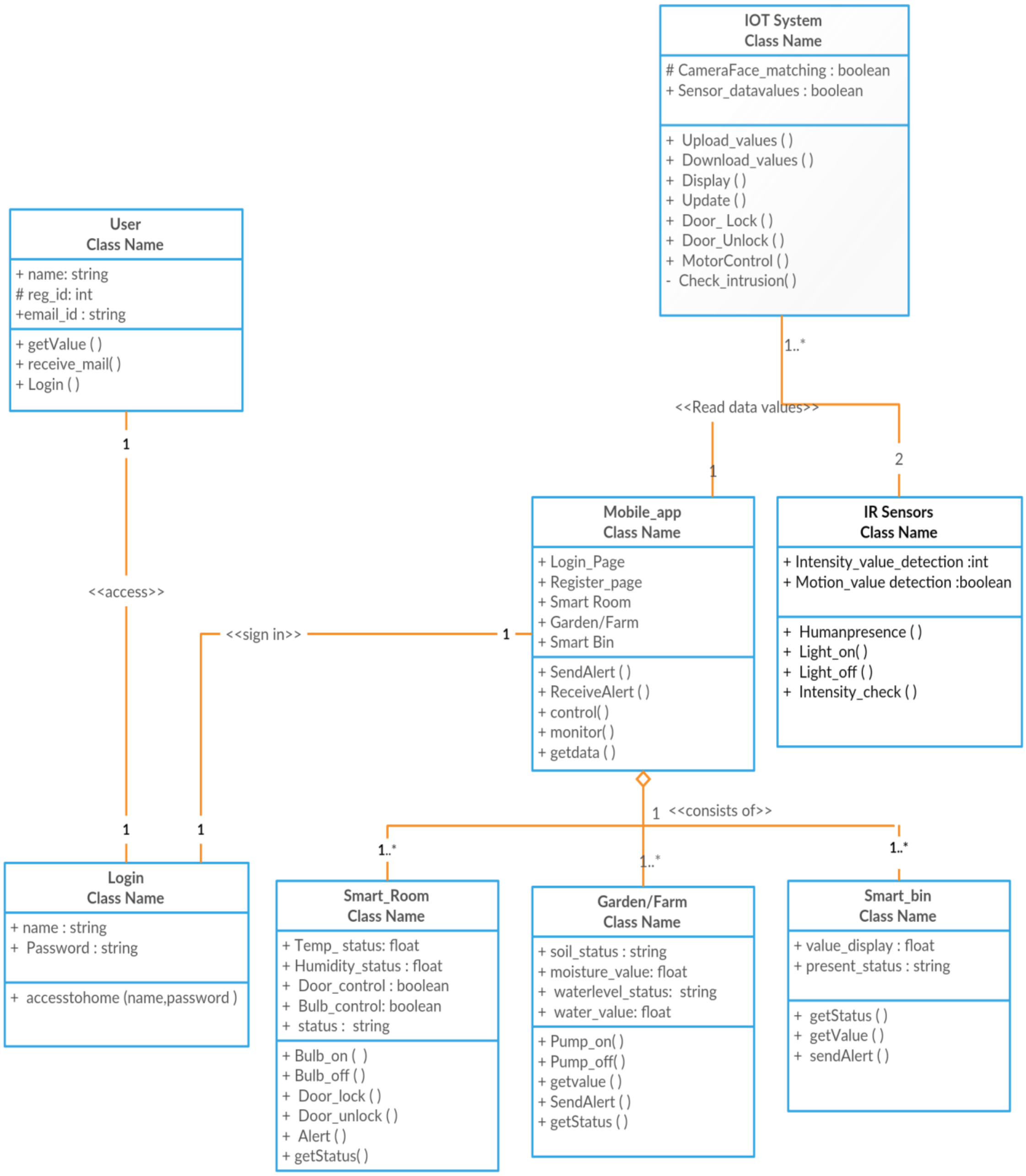


Figure 4.2-Class Diagram

4.2.3 CLASS TABLE

Class	**Purpose**
IOT system	It consists of face recognition based locking system and all sensors that uploads data values to cloud, so that it can be received in Mobile_app.
IR sensors	It provides automatic lighting facility.
Mobile_app	All data values are received with present conditions. Switch control and monitoring is possible.
User	User can access the application to view, control and monitor the entire system.
Login	A login is to ensure security.
Smart_Room	It allows viewing status of room, send alert, control appliances and door lock/unlock.
Garden/Farm	User can water garden/farm through pump on/off. It allows user to check and monitor soil condition and water storage.
Smart Bin	It allows user to check waste level in containers and he can send alert to municipality based on the present status.

Table 4.3-Class table

4.3. USE CASES

4.3.1 Use case Diagram

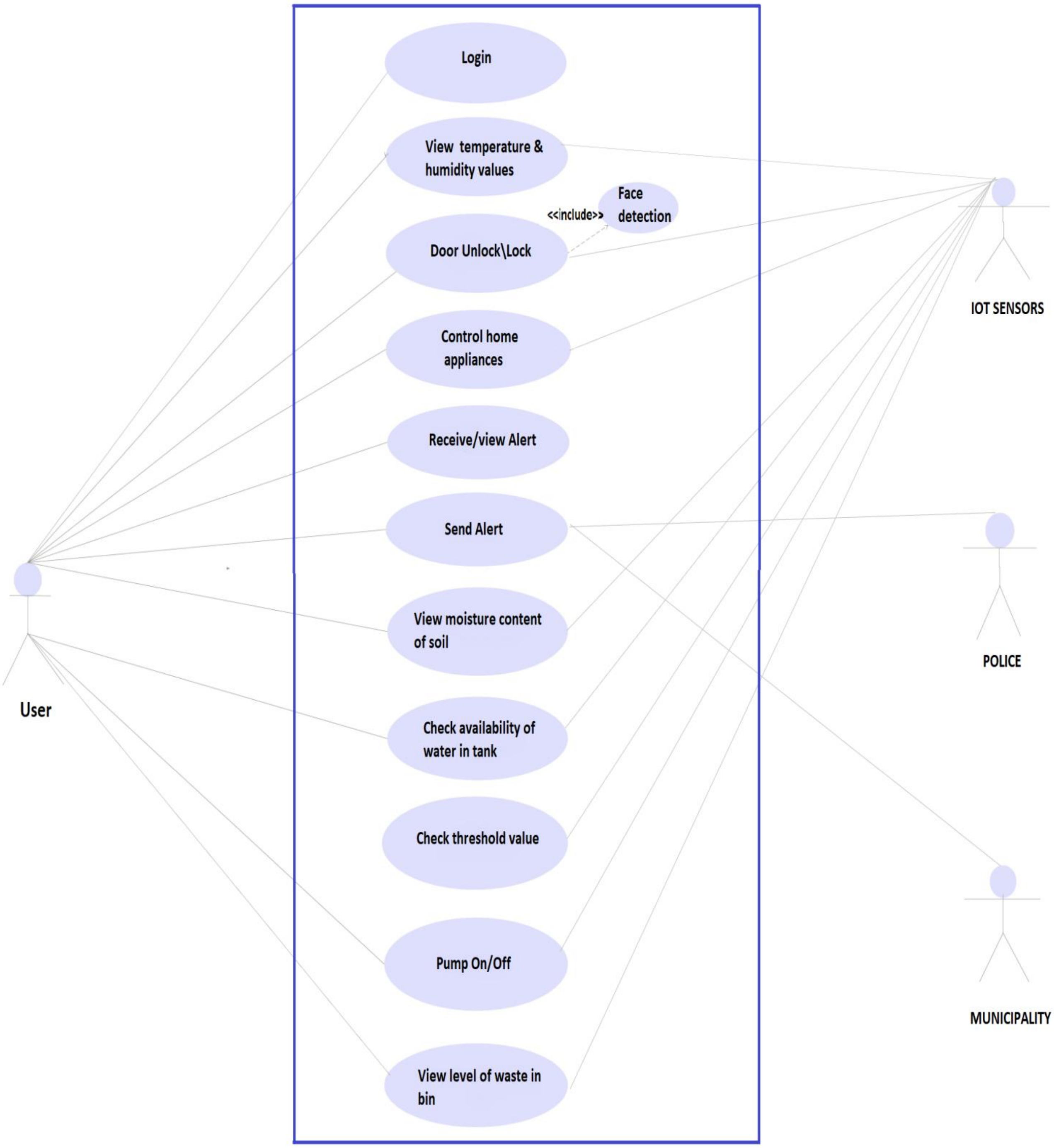


Figure 4.4- Use case diagram

4.3.2 Use case 1

- Use Case ID:ID1
- Use Case Name Login
- Description:

  Allows user to login and access the services of the mobile application.
- Actor(s): User
- Flow of Events (Main Flow & Alternate Flows)

Main flow:

1. User enters his/her user name.
2. User enters his/her password.
3. Click sign in button
4. Password and user id will be verified.
5. Click sign up for new user registration.
6. User enters their user name and email id.
7. Enter valid unique password key.
8. New user can access the service.

Alternate flow:

An incorrect combination of user name and password result in the user being redirected to the login page with an error message “an incorrect user name or password have been entered”.

- Pre-Conditions (optional)

A fully functioning automation system has been successfully installed and powered on.

- Post Conditions (optional)

Data values from sensors will be displayed in the app. Easy and fully automated system as per user need is achieved.

4.3.3 Use case 2

- Use Case ID:ID2
- Use Case Name Control home appliances
- Description: Allows user to control home appliances like: fan, light from mobile app. Based on the motion detected and intensity of light automatic lighting is provided (by IOT sensors).
- Actor(s) User, IOT sensors
- Flow of Events (Main Flow & Alternate Flows)

Main flow:

1. During daytime lights will be switched off.
2. During night it will be switched on by IOT sensor.
3. Motion will be detected and lights will automatically switched on by sensor.
4. User can click ‘bulb on’ when needed.
5. User can click ‘bulb off’ when lighting not needed.

- Pre-Conditions (optional)

A fully functioning automation system has been successfully installed and powered on.

- Post Conditions (optional)

Data values from sensors will be displayed in the app. Easy and fully automated system as per user need is achieved.

4.3.4 Use case 3

- Use Case ID:ID3
- Use Case Name View temperature and humidity values
- Description: Allows user to view temperature and humidity inside a room.

- Actor(s): User, IOT sensors
- Flow of Events (Main Flow & Alternate Flows)

Main flow:

1. Temperature values from sensors will be displayed.
2. Humidity values from sensors will be displayed.
3. User can view values in display.
4. User can send alert to concerned person when a value crosses the desired level.

- Pre-Conditions (optional)

A fully functioning automation system has been successfully installed and powered on

- Post Conditions (optional)

Data values from sensors will be displayed in the app. Easy and fully automated system as per user need is achieved

4.3.5 Use case 4

- Use Case ID:ID4
- Use Case Name: Door lock /unlock
- Description: Ensures security to user's home through face recognition technology. Based on recognized face automatic door lock/unlock will be performed.
- Actor(s) User, IOT sensors
- Flow of Events (Main Flow & Alternate Flows)

Main flow:

1. When a new person or user tries to open the door, security system activates with help of face recognition technology.
2. If face matches then door opens automatically.

3. User is notified of who tried to open the door as the camera captures the face image and sends it to user via email.

Alternate flow:

1. If new person is acquaintance, user can unlock the door through app.

- Pre-Conditions (optional).

A fully functioning automatic face pattern matching system through camera has been successfully installed and powered on.

- Post conditions:

Door will open if face recognized match.

4.3.6 Use case 5

- Use Case ID:ID5
- Use case name: View level of waste in bin
- Use case Description: The proposed IOT system consist of a sensor. This sensor is set with a limit. When waste crosses the set level automatic message will be send to user. So that user can send alert to municipality.

- Actors: User,Municipality,IOT Sensors
- Flow of Events (Main Flow & Alternate Flows)

Main Flow:

1. User can dump waste in smart bin.
2. Height of waste will be displayed in app.
3. Status (full) will be displayed when waste crosses the limit set.
4. Status (empty) will be displayed when no waste or level of waste below the level set.

- Pre-Conditions (optional):

A fully functioning automatic system has been successfully installed and powered on.

- Post-Conditions (optional):

Data values from sensors with current status of waste will be displayed in app.

4.3.7 Use case 6

- Use Case ID:ID6
- Use case name: Send Alert and Receive/view Alert
- Description: When system detects an emergency sends data values and present status from different sensors. User can send alert to the correct authority (municipality and police) through app.
- Actors: User, Municipality, Police, IOT sensors
- Flow of Events (Main Flow & Alternate Flows)

Main Flow:

1. When the system (IOT sensors) detects emergency it will send data to user.
2. The mobile application allows user to send alert message to municipality if needed.
3. The system notifies municipality location of container where the waste needs to be collected.
4. System (IOT sensors) will send alert SMS and photo of new person near front door to user.
5. App allows user to send alert to police.

- Pre-Conditions (optional):

System and its sensors are monitoring front door and waste bin.

- Post-Conditions (optional):

User has notified the correct authority.

4.3.8 Use case 7

- Use Case ID:ID7
- Use case name: View moisture content of soil and pump on/off
- Description: Moisture level of soil will be send from sensors with present condition of soil. Based on which user can pump water through buttons in app.
- Actors: User, IOT sensors
- Flow of Events (Main Flow & Alternate Flows)

Main Flow:

1. Sensors will check the moisture content in soil.
2. Moisture level value will be displayed in the app.
3. Status of soil moisture condition (dry/wet) will be displayed in app.
4. User can pump on/off based on soil conditions.
5. Water will flow through pipes.

- Pre-Conditions (optional):

A fully functioning Plant Watering System has been successfully installed and powered on.

- Post-Conditions (optional):

Data values from sensors with current status of soil will be displayed in app.

4.3.9 Use case 8

- Use Case ID:ID8
- Use case name: Check availability of water in the tank
- Description: An automatic water storing using sensor .
- Actors: User, IOT sensors, tank
- Flow of Events (Main Flow & Alternate Flows)

Main flow:

1. Water level will be detected by sensors
2. Value will be displayed with present status of water tank.
3. System will automatically fill tank with water.

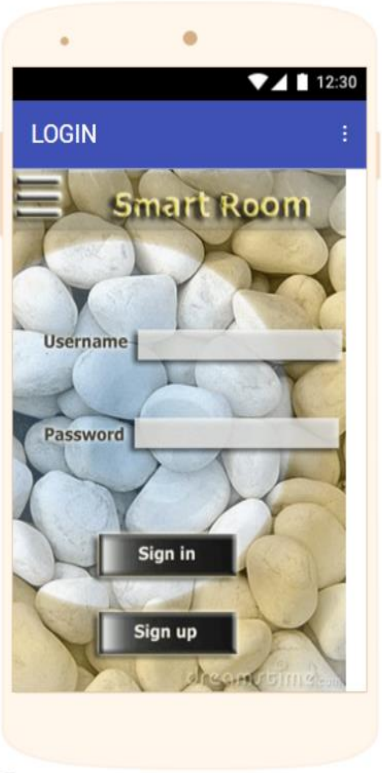
- Pre-Conditions (optional):

A fully water storing System has been successfully installed and powered on.

- Post-Conditions (optional):

Water will automatically fill in the storage tank.

4.4 UI DESIGN



In **Login** User enters his username and password. Then he clicks sign in button which allows user to access the services in mobile application.

Figure: 4.5- Login

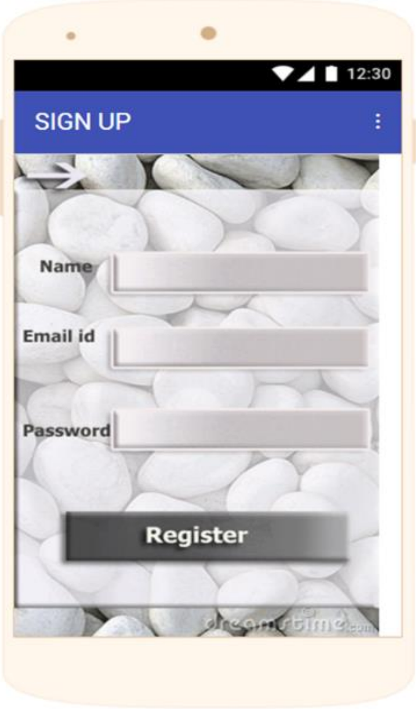


Figure: 4.6-Sign Up

New user can register by clicking **Sign up** button in login page. User will be directed to register page where he\she can register by entering email id, username and password.

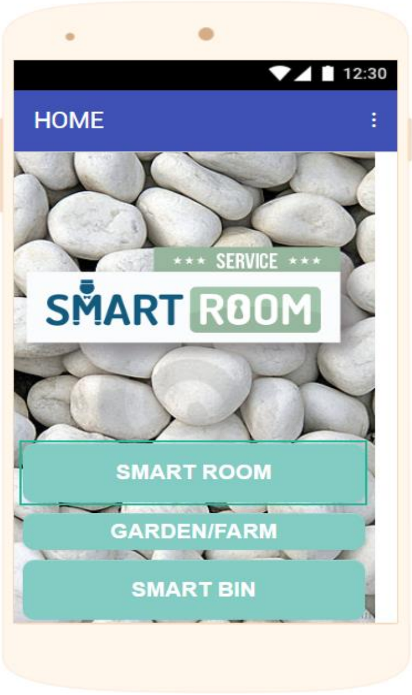
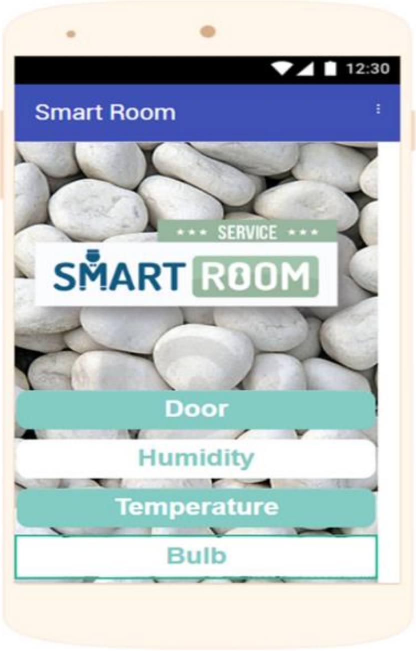


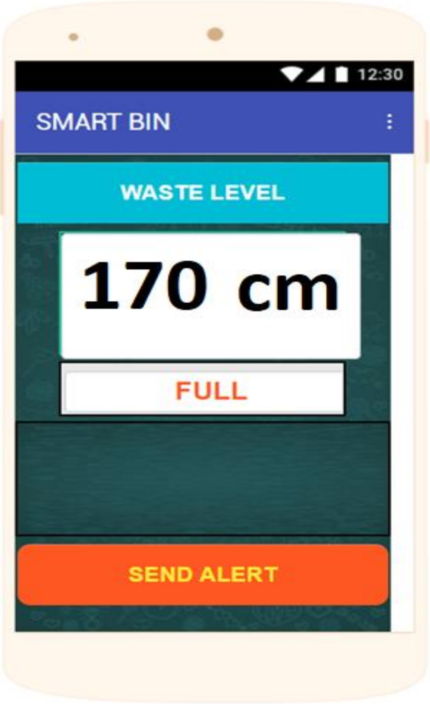
Figure: 4.7-Home

When login is successful user will be directed to this **Home** page. Where he can select any option among the three.



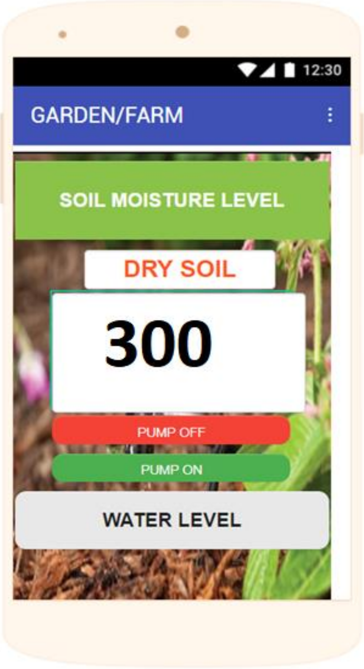
When user clicks **Smart Room** button he will be directed to this screen. User can click any button among the the four to view the respective screens related to each button.

Figure: 4.8-Smart Room



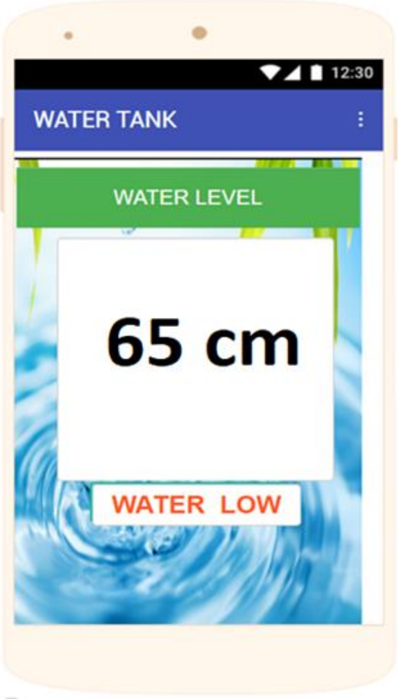
When user clicks **Smart Bin** button in home screen he will be directed to this smart bin screen. Where user has the provision to view waste level with present condition of bin. Based on which user can send alert to municipality by clicking send alert button.

Figure: 4.9-Smart Bin



On clicking **Garden/Farm** button in home screen user will be directed to garden/farm screen. Here user has the provision for automatic watering of plants by clicking pump on/off buttons based on the moisture level and present condition of soil. For viewing water level in storage tank user can click water level button.

Figure: 4.10-Garden\Farm



On clicking **Water level** button in garden/farm screen user will be directed to water tank screen. Here user has the provision to view level of water in storage tank along with that present condition is also shown.

Figure: 4.11-Water level

4.5 DATABASE DESIGN

TABLE NAME

User		
#	**Column Name**	**Type**
1	**User_name**	**varchar(50)**
3	**Password**	**varchar(50)**
4	**Email**	**varchar(50)**

Table 4.12-User

Owner		
#	**Column Name**	**Type**
1	**User_name**	**Varchar(50)**
2	**Password**	**Varchar(50)**

Table 4.13-Owner

IMPLEMENTATION AND PRESENTATION OF DATA

5. IMPLEMENTATION AND PRESENTATION OF DATA

5.1. HARDWARE REQUIREMENTS

- Core i3 and above
- Jumper wires
- Breadboard
- Nodemsu ESP8266 microcontroller chip
- 8GB DDR4-2400 SDRAM
- Servo SG 90
- Ultrasonic sensor HC-SR04
- PIR Motion sensor (Hc-Sr501)
- LDR
- Soil moisture sensor
- L298D motor driver
- DC Motor
- 9V Battery

5.2. SOFTWARE REQUIREMENTS

- Windows 10 OS
- Arduino IDE version:1.8.4
- Android
- Libraries
- Thingspeak.com
- Ubidots
- Firebase DB
- Open CV library

5.3. TESTING

System testing is normally carried out in a planned manner according to the system test plan document. The system test plan identifies all testing-related activities that must be performed, specifies the schedule of testing, and allocates resources. It also lists all the test cases and the expected outputs for each test case. Here the modules are integrated in a planned manner.

Each module must conform to a list of basic tests. This list is generic enough to apply to all modules, but rigid enough to provide a high-level start to the testing process in an accurate manner. The testing will be divided into the following sections:

1. Unit Testing
2. Integration Testing
3. Functional Testing

5.3.1. UNIT TESTING

Unit testing is the responsibility of the developer. Each module must be tested against a basic list of tests. This verifies that each module satisfies the minimum requirements for it to be accepted into the branch. Considering that each module may be significantly different, this list is only a starting place and each developer is required to expand the test list to fully verify the integrity of each module. Before development of a module, each developer is required to provide a list of basic test cases that are specific to their module. These test cases should encompass the main aspects of the module and test them thoroughly. Upon completion of development, the developer is required to create a separate list of test cases that will further verify the integrity of the module. This ensures that the module is thoroughly tested and all results are documented for the code reviewers.

Basic Unit Testing List

- Input validation and handling implemented

- Output format and standards correct
- Coding and documentation standards used
- Error handling and data handling implemented
- Does it satisfy all of the given requirements for the module

5.3.2 Integration Testing

Integration testing requires that the associated modules be completed and unit tested. Integration testing will verify the integration of the various modules. This requires a review of the unit testing documentation, the long-term requirements, and the standards that apply to the modules. With these documents in hand, the testers can begin to test the integration of the modules. This is done by starting with a pre-defined list of generic integration test cases and then defining further test cases based on the specific module functionality.

Basic Integration Testing List:

- Does the output of module A correctly match the inputs of module B.
- Do both modules conform to integration standards.
- Do both modules have the required documentation.
- Does module A call the correct functions in module B for the required functionality.
- Is the output of the combined system within the requirements specified.
- Is the input of the system a correct mapping to the output of the integrated system.
- Does the integration create any new bugs or unintended behaviour.

5.3.3 Functional Testing

Functional testing will be broken down into two sections; the preliminary functional test design early in the development process and the execution of the designed functional tests when the bulk of the system has been completed. This tests the entire system and its specifications. The design of the functional testing will focus on the actual functionality of the system. That is, they will test the inputs and outputs of the system to verify that they conform to the system specifications. Designing the functional tests early in the development process allows for early detection of specification problems that might be not be discovered until the actual execution of the functional testing. The second section, the execution of the functional tests, requires the completion of the designed functional tests. Each test is executed and the result is either a pass or fail. If all functional tests pass, the system is said to pass the functional testing.

5.3.4. TEST CASES

A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not**.**

PROJECT NAME	IOT BASED SMART ROOM
MODULE NAME	LOGIN

TESTCASE ID	TEST CASE	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_LOGIN_1	Verify the login in application.	1.Enter valid username 2.Enter valid password 3.Click sign in button	<Valid Username> <Valid Password>	Successful login	Successfully logged in and displayed the home page.	PASS
TC_LOGIN_2	Verify the login credentials.	1.Enter valid username 2.Enter invalid password 3.Click sign in button	<Valid Username> <Invalid Password>	A message "Enter correct username and password"	Redirected to the same login page.	PASS

TC_REGISTRATION_1	Verify the sign up in login page of application.	Click sign Up button.	<No data required>	Successful directed to register page.	Directed to register page.	PASS
TC_REGISTRATION_1	New registration.	1.Enter new username. 2.Enter email id. 3.Enter new password.	<User name> <Email id> <Password>	Successful registration.	Redirected to the login page.	PASS

Table 5.1 Login

PROJECT NAME	IOT BASED SMART ROOM
MODULE NAME	SMART ROOM

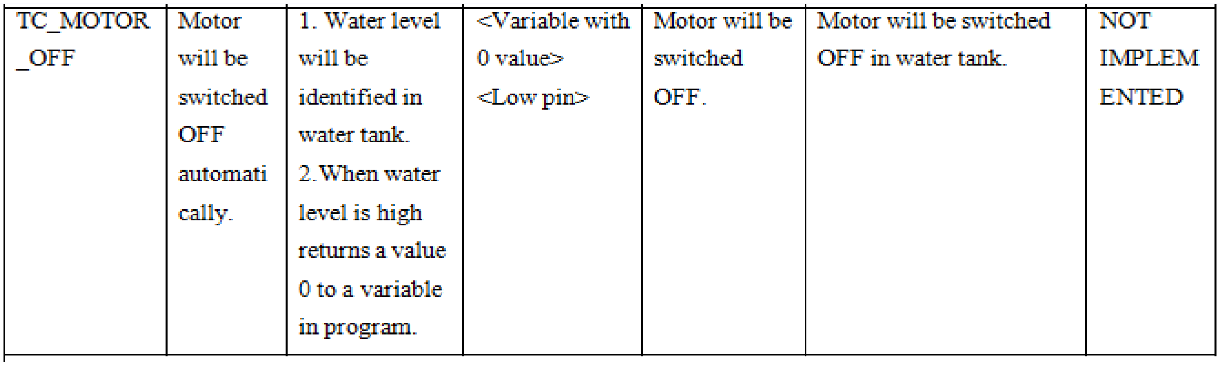
TESTCASE ID	TEST CASE	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_HOME_1	To access different services.	Click smart room button.	<No data>	Directed to smart room screen.	Successfully Directed to smart room screen.	PASS
TC_TEMPERATURE	View present temperature value.	Value displayed in textbox.	<Data from cloud>	Can view temperature value in Celsius.	Temperature value of room displayed.	PASS
TC_BULB ON	Switch ON bulb through app.	Click ON button to switch ON light.	<High(1) will be downloaded to cloud>	Can switch ON bulb in app.	Bulb will be switched ON.	PASS

TC_BULB OFF	Switch OFF bulb through app.	Click OFF button to switch OFF light.	<Low(0) will be downloaded to cloud>	Can switch OFF bulb in app.	Bulb will be switched OFF.	PASS
TC_HUMIDI TY	View present humidity value.	Value displayed in textbox.	<Data from cloud>	Can view humidity value.	Humidity value of room displayed.	PASS
TC_HUMIDI TY ALERT	Send alert when required.	Click send alert button.	<Set message>	Can send alert message to anyone.	Alert message can be send.	PASS
TC_TEMPE RATURE ALERT	Send alert when required.	Click send alert button.	<Set message>	Can send alert message to anyone.	Alert message can be send.	PASS
TC_DOOR LOCK	Lock door through app.	Click lock button in app.	<High(1) will be downloaded to cloud>	Can lock door.	Door will be locked on clicking button.	PASS
TC_DOOR UNLOCK	Unlock door through app.	Click unlock button in app.	<Low(0) will be downloaded to cloud>	Can unlock door.	Door will be unlocked on clicking button.	PASS

Table 5.2 Smart Room

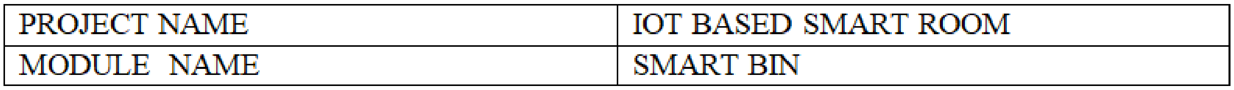
PROJECT NAME	IOT BASED SMART ROOM
MODULE NAME	GARDEN/FARM

TESTCASE ID	**TEST CASE**	**TEST STEPS**	**TEST DATA**	**EXPECTED RESULT**	**ACTUAL RESULT**	**STATUS (PASS/FAIL)**
TC_HOME_2	To access different services.	Click Garden/Farm button.	<No data>	Directed to garden/farm screen.	Successfully Directed to garden/farm screen.	PASS
TC_SOIL MOISTURE LEVEL	View present moisture level value of soil in app.	Value displayed in textbox.	<Data from cloud>	Can view moisture content of soil in mm^3.	Moisture level value of soil will be displayed.	NOT IMPLEMENTED
TC_PUMP ON	Switch ON water pump through app.	Click ON button to switch ON pump.	<High(1) will be downloaded to cloud>	Can switch ON pump in app.	Water pump will be switched ON this enables to water plants.	NOT IMPLEMENTED
TC_PUMP OFF	Switch OFF water pump through app.	Click OFF button to switch OFF pump.	<Low(0) will be downloaded to cloud>	Can switch OFF pump in app.	Water pump will be switched OFF.	NOT IMPLEMENTED
TC_WATER TANK	View present water level value in storage tank.	Value displayed in textbox.	<Data from cloud>	Can view water level value.	Can view water level in water tank.	PASS
TC_MOTOR_ON	Motor will be switched ON automatically.	1. Water level will be identified in water tank. 2.When water level is low returns a value 1 to a variable in program.	<Variable with 1 value> <High pin>	Motor will be switched ON.	Motor will be switched ON in water tank.	NOT IMPLEMENTED

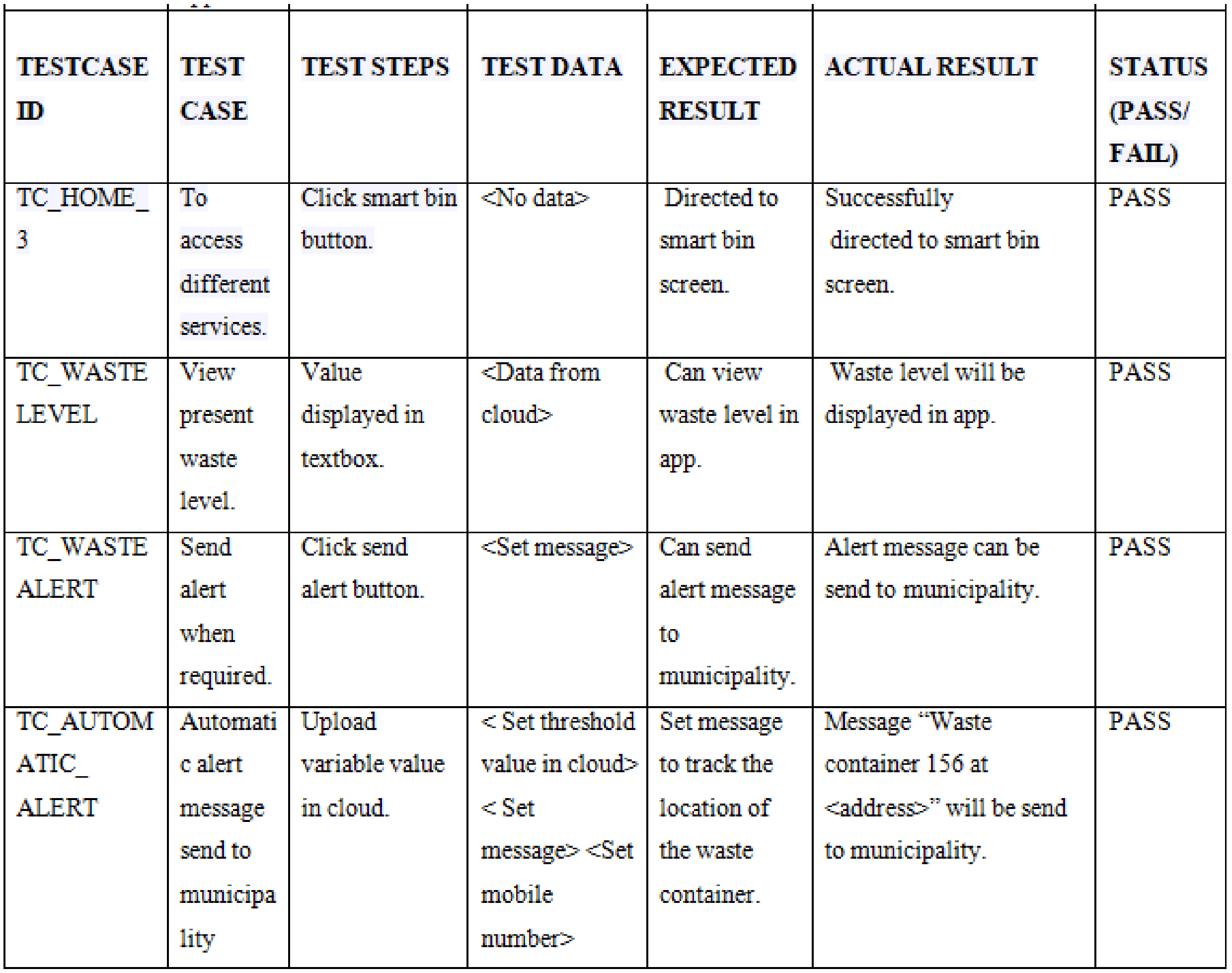


TC_MOTOR _OFF	Motor will be switched OFF automatically.	1. Water level will be identified in water tank. 2. When water level is high returns a value 0 to a variable in program.	<Variable with 0 value> <Low pin>	Motor will be switched OFF.	Motor will be switched OFF in water tank.	NOT IMPLEMENTED

Table 5.3 Garden/Farm



PROJECT NAME	IOT BASED SMART ROOM
MODULE NAME	SMART BIN



TESTCASE ID	TEST CASE	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/ FAIL)
TC_HOME_ 3	To access different services.	Click smart bin button.	<No data>	Directed to smart bin screen.	Successfully directed to smart bin screen.	PASS
TC_WASTE LEVEL	View present waste level.	Value displayed in textbox.	<Data from cloud>	Can view waste level in app.	Waste level will be displayed in app.	PASS
TC_WASTE ALERT	Send alert when required.	Click send alert button.	<Set message>	Can send alert message to municipality.	Alert message can be send to municipality.	PASS
TC_AUTOM ATIC_ ALERT	Automatic alert message send to municipality	Upload variable value in cloud.	< Set threshold value in cloud> < Set message> <Set mobile number>	Set message to track the location of the waste container.	Message "Waste container 156 at <address>" will be send to municipality.	PASS

Table 5.4 Smart Bin

CONCLUSION

6. CONCLUSION

In this study, existing products in the chosen domain were analyzed. Issues that arise in commercial and residential buildings have been examined, and ways of evaluating a project have been discussed. Use case diagram developed with use case specification for various blocks of proposed system. UI design developed as per user convenience and comfort. Database design and class diagrams were developed. Implementation of the proposed system was discussed which included: Hardware and Software requirements. This project has been a really great experience and opportunity to learn and to experiment. The output of this project is an array of home appliances that are controlled over the internet with the help of a mobile application.

FUTURE WORKS SUGGESTED

7. FUTURE WORKS SUGGESTED

- The proposed system can be used for office/commercial buildings by adding some more additional features since it provides complete security and will save time and expenditure.
- This system can be modified by interfacing it with a battery power supply.
- Features can be extended by adding more sensors: like smoke sensors, child lock facility, voice based control.

REFERENCES

8. REFERENCES

[1]https://www.buildtrack.in

[2] https://youtu.be/B0q-UtJnVRs

[3]https://www.google.co.in/search?q=home+automation&source=lnms

[4] http://en.wikipedia.org/wiki/home_automation

[5] http://www.dumpobin.com/smart-features.htm

[6] https://simpro.world/bins/240l-wheelie-bin

[7] http://www.dailycivil.com/drip-irrigation-types-advantages-disadvantages

[8] Ning Wanga, Naiqian Zhangb, Maohua Wangc, "Wireless sensors in agriculture and food industry-Recent development and future perspective", *science direct*, 2006

[9] Dan Teibel, Pat Bowen, Results from an Agricultural Wireless Sensor Network, IEEE, 2004.

APPENDIX

APPENDIX I - CODE SNIPPETS

```
A. UPLOAD CODE: Data values from different sensors are uploaded
   to thingspeak.

#include <DHT.h>
#include  "DHT.h"            //  including  the  library  of  DHT11
temperature and humidity sensor
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"
#include <SPI.h>

//Thingspeak Channel Field Values Measurement
unsigned long myChannelNumber = 535985;
const char * myWriteAPIKey = "IAT81J440UZ9H95B";

#define WIFI_SSID "AndroidAP"
#define WIFI_PASSWORD "icwu1856"
#define DHTTYPE DHT11   // DHT 11
WiFiClient client;

#define dht_dpin D7
DHT dht(dht_dpin, DHTTYPE);
int ledPin = D3;                // choose the pin for the LED
int inputPin = D4;                // choose the input pin (for
PIR sensor)
int pirState  =  LOW;                //  we  start,  assuming  no
motion detected
int door_state = 0;                // variable for reading
the pin status
int pinSpeaker = D3;  //PIR MOTION SENSOR LED
long duration;//UV
long cm;//UV Distance in cm

void setup() {
pinMode(ledPin, OUTPUT);      // declare LED as output
pinMode(inputPin, INPUT);     // declare sensor as input
pinMode(pinSpeaker, OUTPUT);
pinMode(A0,INPUT);
```

```
  pinMode(D6,OUTPUT);
  pinMode(D1,OUTPUT);//Trigger
  pinMode(D2,INPUT);//Echo
  dht.begin();
  Serial.println("Humidity and temperature\n\n");
  delay(700);
  Serial.begin(9600);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);// connect to wifi.
  ThingSpeak.begin(client);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
}

  void loop(){
  door_state = digitalRead(inputPin);  // read input value
  if (door_state == HIGH) {  // check if the input is HIGH
  digitalWrite(ledPin, HIGH);  // turn LED ON
  //playTone(300, 160);
  delay(300);
  digitalWrite(pinSpeaker, HIGH);
  delay(10);
  ThingSpeak.setField(1,door_state);
  ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
  delay(20000);

  if (pirState == LOW) {
      // we have just turned on
      Serial.println("Motion detected!");
      // We only want to print on the output change, not state
      pirState = HIGH;
    }
  } else {
    digitalWrite(ledPin, LOW); // turn LED OFF
   // playTone(0, 0);
    delay(300);
    digitalWrite(pinSpeaker, LOW);
    delay(10);
    ThingSpeak.setField(1,door_state);
    ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    delay(20000);
    if (pirState == HIGH){
    // we have just turned of
    Serial.println("Motion ended!");
    // We only want to print on the output change, not state
    pirState = LOW;
  }
}
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
  Serial.print("Current humidity = ");
  Serial.print(humidity);
  ThingSpeak.setField(2,humidity);
  ThingSpeak.writeFields(myChannelNumber,myWriteAPIKey);
  delay(20000);
  Serial.print("%  ");
  Serial.print("temperature = ");
  Serial.print(temperature);
  ThingSpeak.setField(3,temperature);
  ThingSpeak.writeFields(myChannelNumber,myWriteAPIKey);
  delay(20000);
  Serial.println("C");
  delay(800);
  int bulb_state=analogRead(A0);
  ThingSpeak.setField(4,bulb_state);
  ThingSpeak.writeFields(myChannelNumber,myWriteAPIKey);
  delay(20000);
            if(bulb_state<1000)
              {
              Serial.println(bulb_state);
              Serial.println("is lower");
              digitalWrite(D6,LOW);
```

```
        delay(1000);
        }
       else
        {
        Serial.println(bulb_state);
        Serial.println("is greater");
        digitalWrite(D6,HIGH);
        delay(1000);
        }
digitalWrite(D1, LOW);  // The PING))) is triggered by a HIGH
pulse of 10 microseconds.
delay(2/1000);    //    Give    a    short    LOW    pulse
(2 microsec) beforehand to ensure a clean HIGH pulse:
digitalWrite(D1, HIGH);
delay(10/1000);
digitalWrite(D1, LOW);

duration = pulseIn(D2, HIGH);// The same pin is used to read
the signal from the PING))): a HIGH pulse
               // whose duration is the time
(in microseconds) from the sending of the ping
               // to the reception of its echo
off of an object.
cm = microsecondsToCentimeters(duration);//waste level in cm
Serial.print(cm);
Serial.println("cm");
delay(1000);
ThingSpeak.setField(8,cm);//Channel field of waste level
ThingSpeak.writeFields(myChannelNumber,myWriteAPIKey);
delay(20000);
}
 long microsecondsToCentimeters(long microseconds) {
 return (duration *.034)/ 2; // The speed of sound is 340 m/s

}
```

B. DOWNLOAD CODE: Values from mobile application is downloaded to thingspeak.

```
#include <ThingSpeak.h>
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"
#include <SPI.h>
#include <Servo.h>
//Channel Field Values Measurement
unsigned long myChannelNumber = 535985;
const char * myReadAPIKey = "OX2UGO92S9NUIC8P";
#define WIFI_SSID "AndroidAP"
#define WIFI_PASSWORD "oqwl6849"
WiFiClient client;
Servo servo;
//Pump pins
int ENA = D3;
int IN1 = D5;
int IN2 = D6;

void setup() {
  Serial.begin(9600);
  pinMode(D8,OUTPUT);
  servo.attach(D5);  //D5
  servo.write(0);
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  delay(2000);
  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  ThingSpeak.begin(client);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);

  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
```

```
 }
 void loop()
 {
intled=ThingSpeak.readFloatField(myChannelNumber,5,myReadAPIKey)
intservo_status=ThingSpeak.readFloatField(myChannelNumber,6,myRe
adAPIKey);
intpump_status=ThingSpeak.readFloatField(myChannelNumber,8,myRea
dAPIKey);
Serial.print(led);
Serial.print(servo_status);
Serial.print(pump_status);
 if (led==1){
   digitalWrite(D8, HIGH);
   delay(1000);}  // turn the LED on (HIGH is the voltage level)
   else
   {
   digitalWrite(D8,LOW);
   delay(1000);
   }
   if (servo_status==1){
     servo.write(90);
     delay(1000);}
   else
   {
     servo.write(0);
     delay(1000);
   }
   if (pump_status==1){
     // turn on pump
       digitalWrite(ENA,  HIGH);   //  set  speed  to  200  out  of
 possible range 0~255
       digitalWrite(IN1, HIGH);
       digitalWrite(IN2, LOW);
     }
   else
   {
      //turn off pump
      digitalWrite(IN1, LOW);
      digitalWrite(IN2, LOW);
```

```
  }

}
```

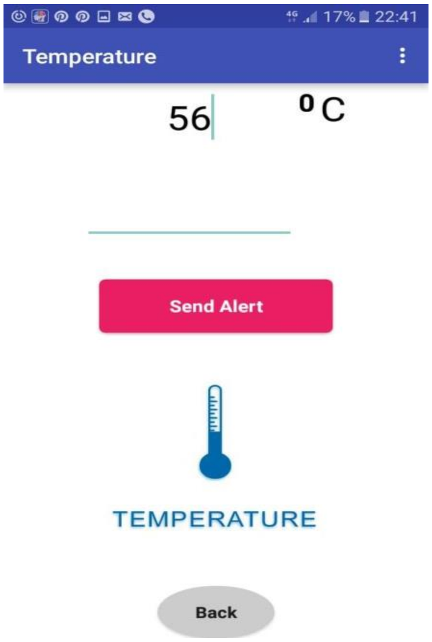
C: UPLOAD CODE: Values from sensors are uploaded to ubidots(for automatic alert message).

```
#include "DHT.h"            // including the library of DHT11
temperature and humidity sensor
#include <ESP8266WiFi.h>
#include "UbidotsMicroESP8266.h"
#include <SPI.h>

#define TOKEN  "A1E-a6eacfcfb1eb50e71841638937e37422c791"  //
Put here your Ubidots TOKEN
//#define ID_1 "Your_variable_ID_here" // Put your variable ID
here
//#define ID_2 "Your_variable_ID_here" // Put your variable ID
here

#define WIFI_SSID "Android AP"
#define WIFI_PASSWORD "icwu1856"
#define DHTTYPE DHT11  // DHT 11
Ubidots client(TOKEN);

#define dht_dpin D7
DHT dht(dht_dpin, DHTTYPE);
int ledPin = D3;                // choose the pin for the LED
int inputPin = D4;                // choose the input pin (for
PIR sensor)
int pirState = LOW;             // we start, assuming no motion
detected
int door_state = 0;                     // variable for reading
the pin status
int pinSpeaker = D3;            //Set up a speaker on a PWM pin
(digital 9, 10, or 11)

void setup() {
  pinMode(D1,OUTPUT);//Trigger
  pinMode(D2,INPUT);//Echo
  pinMode(ledPin, OUTPUT);      // declare LED as output
  pinMode(inputPin, INPUT);     // declare sensor as input
  pinMode(pinSpeaker, OUTPUT);
  pinMode(A0,INPUT);
  pinMode(D6,OUTPUT);
  dht.begin();
  Serial.println("Humidity and temperature\n\n");
  delay(700);
  Serial.begin(115200);
  client.wifiConnection(WIFI_SSID, WIFI_PASSWORD);
}

void loop(){
  door_state = digitalRead(inputPin);  // read input value
  if (door_state == HIGH) { // check if the input is HIGH
    digitalWrite(ledPin, HIGH);  // turn LED ON
    //playTone(300, 160);
    delay(300);
    digitalWrite(pinSpeaker, HIGH);
    delay(10);
    client.add("door_state", door_state);
    client.sendAll(true);


    if (pirState == LOW) {
      // we have just turned on
      Serial.println("Motion detected!");
      // We only want to print on the output change, not state
      pirState = HIGH;
    }
  } else {
      digitalWrite(ledPin, LOW); // turn LED OFF
     // playTone(0, 0);
      delay(300);
      digitalWrite(pinSpeaker, LOW);
      delay(10);
      client.add("door_state", door_state);
    client.sendAll(true);
      if (pirState == HIGH){
```

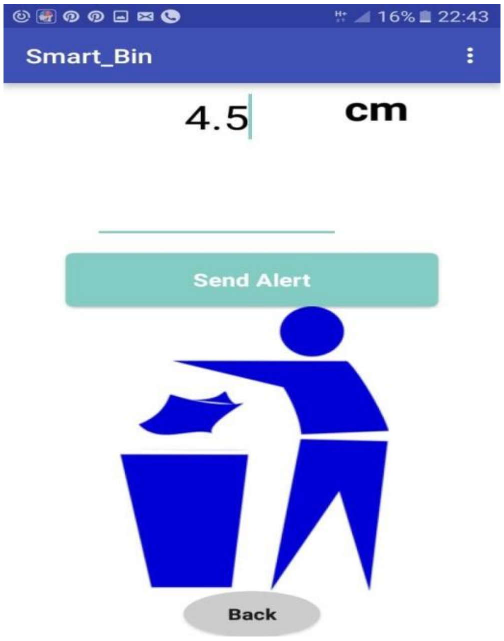
```
    // we have just turned of
    Serial.println("Motion ended!");
    // We only want to print on the output change, not state
    pirState = LOW;
  }
}
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
  Serial.print("Current humidity = ");
  Serial.print(humidity);
  client.add("humidity", humidity);
  client.sendAll(true);
  Serial.print("%  ");
  Serial.print("temperature = ");
  Serial.print(temperature);
  client.add("temperature", temperature);
  client.sendAll(true);
  Serial.println("C  ");
  delay(800);
  int bulb_state=analogRead(A0);
  client.add("bulb_state", bulb_state);
  client.sendAll(true);
            if(bulb_state<1000)
              {
              Serial.println(bulb_state);
              Serial.println("is lower");
              digitalWrite(D6,LOW);
              delay(1000);
              }
            else
              {
              Serial.println(bulb_state);
              Serial.println("is greater");
              digitalWrite(D6,HIGH);
              delay(1000);
              }

   digitalWrite(D1, LOW);
```

```
 // The PING))) is triggered by a HIGH pulse of 10
microseconds.
   delay(2/1000);// Give a short LOW pulse (2 micro sec)
beforehand to ensure a clean HIGH pulse:
   digitalWrite(D1, HIGH);
   delay(10/1000);
   digitalWrite(D1, LOW);
   duration = pulseIn(D2, HIGH
   cm = microsecondsToCentimeters(duration);
   Serial.print(cm);
   Serial.println("cm");
   delay(1000);
   client.add("cm", cm);
   client.sendAll(true);
}
   long microsecondsToCentimeters(long microseconds) {
   return (duration *.034)/ 2; // The speed of sound is 340 m/s
}
}
```

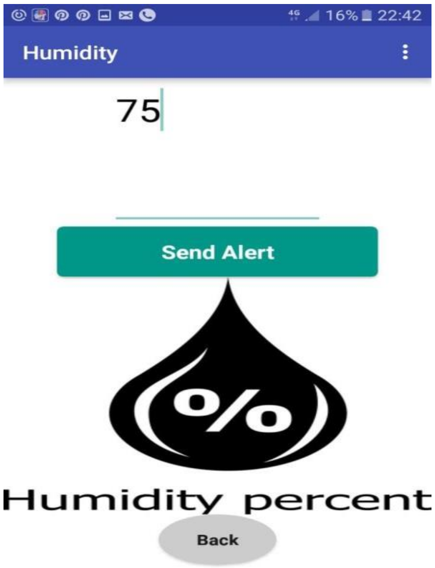
APPENDIX II - SCREEN SHOTS



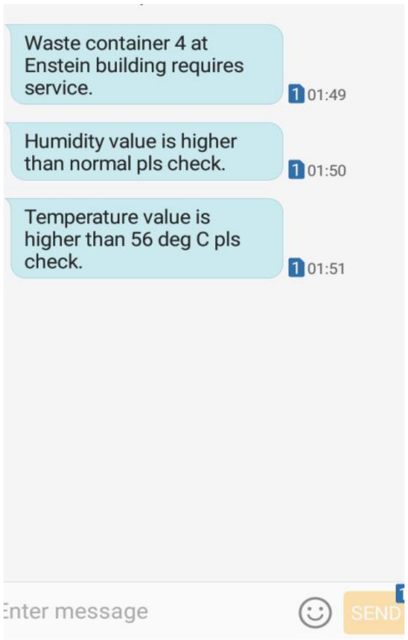
A. TEMPERATURE



B. SMART BIN



C. HUMIDITY

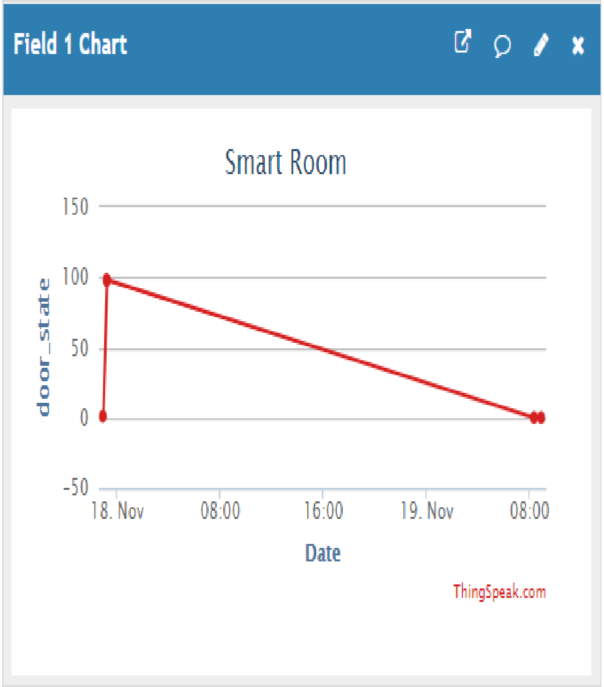


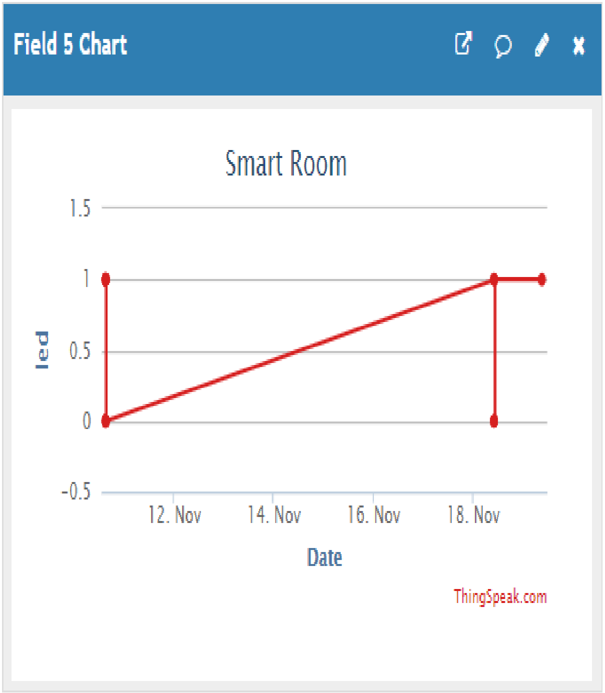
Waste container 4 at Enstein building requires service. 1 01:49

Humidity value is higher than normal pls check. 1 01:50

Temperature value is higher than 56 deg C pls check. 1 01:51

D. RECEIVED ALERT





E. DOOR LOCK/UNLOCK STATUS

F. BULB ON/OFF STATUS