DECLARATIONS

This is to certify that this project is an original work and was done by us and it has not been submitted elsewhere for the requirement of any other purposes.

Signature of the Students

……	……	…….
Syed Shafaet Hossain	**Md. Rasel Miah**	**MD. Tanvir Ahmed**

ACCEPTANCE

This project is entitled "**An Android Based IoT System for Garbage Monitoring**" submitted by Syed Shafaet Hossain, ID no. 2014-1-60-087 and Md Rasel Miah, ID no. 2014-2-60-013 and MD. Tanvir Ahmed, ID no. 2014-2-60-032 to the Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh is accepted as satisfactory for partial fulfillment of the requirement for the degree of Bachelor of Computer Science and Engineering on December 2018.

.....

Shakila Mahjabin Tonni
Lecturer
Department of Computer Science and Engineering
East West University, Dhaka, Bangladesh.

……

Dr. Ahmed Wasif Reza
Associate Professor and Chairperson
Department of Computer Science and Engineering

East West University, Dhaka, Bangladesh.

Abstract

In recent years, one of the major problems that the developing countries are facing is waste management. With rapid urbanization, industrialization of nations throughout the world, waste has become a great concern for all of us. In developed countries technology researchers have introduced IoT based Smart Waste Management solutions but unfortunately, developing countries are not being able to implement those existing solutions due to many factors like socio-economic environment. Therefore, in this research we have concentrated our thought on developing "An Android based IoT System for garbage monitoring " for developing countries like Bangladesh that's ensure cost-effecting system which will provides the location of dustbin, notify by the level of garbage filled, time-schedule of cleaning for each dustbin etc.

Acknowledgment

Shakila Mahjabin Tonni, Lecturer of the Department of Computer Science and Engineering, East-West University for her precious contribution, consistent advice and support for the completion of this project. Without her support, it was not possible for us to complete this task.

We would like to express our gratitude to all the faculty members of the Computer Science and Engineering Department for the support they had given.

But above everything, we like to thank Almighty Allah for giving us the ability, patience, and stamina to complete this task successfully.

Table of Contents

Abbreviation and Acronyms

IoT: Internet of Things

SMS: Short Message Service

API: Application Program Interface

APP: Application

IDLE: Integrated Development Environment

SoC: System on a Chip

CPU: Central Processing Unit

USB: Universal Serial Bus

Bit: Binary Digit

GB: GigaByte

RAM: Random Access Memory

GPIO: General Purpose Input / Output

SD Card: Secure Digital Card

LAN: Local Area Network

BLE: Bluetooth Low Energy

List of Figures

Chapter 1

Introduction

If we want to stay healthy, we need a healthy environment. But in our daily lives, we see the garbage baskets that are overflowing with garbage and resulting in pollution. The number of diseases is increasing because of the reproduction of the large number of insects and mosquitoes. At present waste management is the biggest challenge for urban cities not only in Bangladesh but for most of the countries in the world. For collecting waste in a city, there are large amount of garbage baskets are deployed and workers are kept particularly for this task. But this old system of hiring people for cleaning filled garbage basket is unsuccessful because of human error and

negligence. Additionally, a basket might get filled in advance and sometimes it might be empty for a long period of time because in different areas baskets are used different number of times. So the present system of garbage monitoring is costly in respect to time, fuel and money. This project presents the development of an Android based IOT system for garbage monitoring in order to measure the waste level in the garbage basket in real-time and send notification to the administration via SMS. The proposed system is consisted by the ultrasonic sensor to measure the waste level, the TWILIO sms api to send the SMS, ANDROID APP for user interface and RASPBERRY PI which controls the system operation. This chapter contains the introduction part of the system. The motivation part is described in section 1.3. The objective behind this project is discussed in section 1.4. Overview of the proposed system is discussed in the 1.5 section. Contribution of this project is discussed on the section 1.6 and in section 1.7, outlines are discussed.

1.1 About IoT

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.[1] “An Android Based IoT System for Garbage Monitoring” is a system that can measure the level of garbage in a garbage basket. When the garbage level is crossing a certain benchmark it will send an SMS notification through Twilio SMS API. An android mobile app will be using as a user interface. [rest of part will write later…..]

1.2 Motivation

Bangladesh is the ninth most populous and twelfth most densely populated country in the world. [7] In particular, the projected urban population growth rate from 2010 to 2015 is 3%. [8] With this population growth, there is an increasing problem of waste management particularly in the larger cities. [7] Currently, according to an UNFPA report, Dhaka is one of the most polluted cities in the world and one of the issues concerned is the management of municipal waste. [7] In our country, we are very poor in proper waste management because of an enormous number of population growth. The negative impact of poor municipal waste management is the incidence of diseases such as malaria, respiratory problems and other illnesses by the pollution of groundwater. As a result, people are passing days with illnesses and spending a lot of money for buying medicines. So, by thinking this we become motivated to develop a system which will provide a smart, user-friendly and secure way to monitoring garbage and sending a notification.

1.3 Objective

Actually, this system can be run by City Corporation who has several garbage bins throughout the city. They need to follow our mechanism to set up this service. Users of this service can use it through any Android device which is connected to the internet. In the Android app, the user can add new bin, tune initial bin height and edit name, sensor pin number, notification level, bin status. The user can see the level of garbage in the bin, bin location, the number of times bin cleaning and last bin cleaning time and date in the Android app. If the garbage level is crossing a particular benchmark then a notification SMS will be sent in the given mobile number.

1.4 Overview of the proposed system

Write later

1.5 Contribution

In this work, we present Raspberry Pi3 small single-board computer by getting instruction given in the python IDLE, which is written in the Python programming language. The Android mobile application is developed by Java language and the database in RethinkDB cloud real-time database. To connect the android app with RethinkDB database we use Node.js as a server side. We have shown how a mobile app command can handle a hardware system. Basically, this work belongs to IOT (Internet of Things).

1.6 Outline of this project

Chapter 2 describes the entire hardware architecture of this project and the necessary steps of interfacing.

Chapter 3 describes the entire software system of this project and the necessary functions.

Chapter 2

Overview of the Existing Technology

In this chapter, we describe the entire proposed system of our project where all the necessary components name, their significance in the project are described elaborately.

2.1 Overview of Raspberry Pi-3

The Raspberry Pi is a series of small single-board, low cost, credit-card sized computer. It can plug into a computer monitor or TV, and uses a standard keyboard and mouse. It is competent with a little device that enables people of all ages to discover computing and to learn how to program in languages like Scratch and Python. It's capable of doing the whole thing you'd

imagine in a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

What is more, the Raspberry Pi has the capability to interact with the outside world and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting bird houses with infrared cameras. We want to see the Raspberry Pi being used by kids all over the world to learn to program and understand how computers work.

The Raspberry Pi is open hardware, with the exception of the primary chip on the Raspberry Pi, the Broadcom SoC (System on a Chip), which runs many of the main components of the board–CPU, graphics, memory, the USB controller, etc. Many of the projects made with a Raspberry Pi are open and well-documented. In our proposed system, we have utilized Raspberry Pi-3.

2.2 Raspberry Pi-3

The Raspberry Pi 3 Model B is a single-board computer. It has quad-core 64 bit CPU, 1 GB RAM, 100 base Ethernet, 40 GPIO pin, 4 USB 2 port, micro SD card for loading an operating system and storing data, BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board.

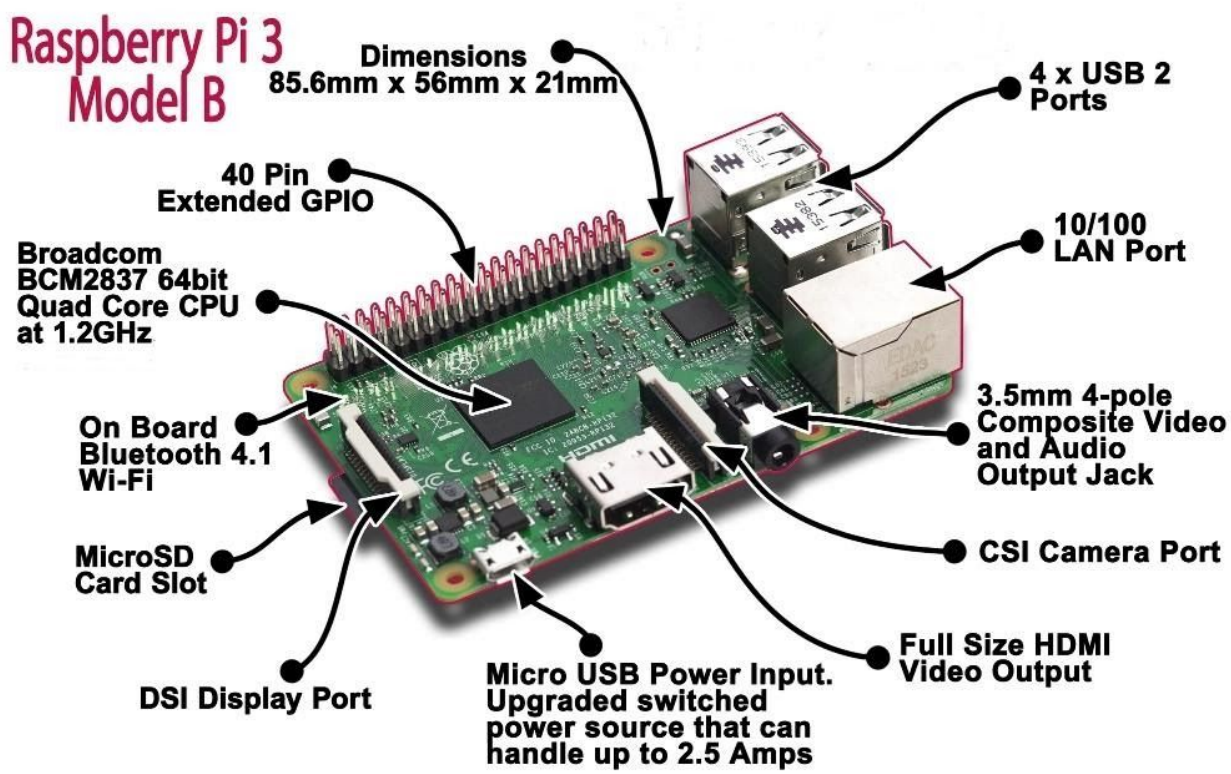


Figure 2. 1: Raspberry Pi 3 Model B

The Raspberry Pi 3 Model B is the earliest model of the third-generation Raspberry Pi. It replaced the Raspberry Pi 2 Model B in February 2016.

- Quad-Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full-size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

2.3 Processor / SoC (System on a chip)

The Raspberry Pi 3 Model B has a Broadcom BCM2837 system on Chip module. It has an ARM Cortex-A53 processor with 512KB shared L2 cache. The chip operates at 700 MHz by default with a quad-core processor and having ten times the performance of a Raspberry Pi 1. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelized tasks.

2.4 Power Source

- 300 mA (1.5 W) average when idle, 1.34 A (6.7 W) maximum under stress (monitor, keyboard, mouse and Wi-Fi connection.
- 459 mA (2.295 W) average when idle, 1.13 A (5.661 W) maximum under stress (monitor, keyboard, mouse and Wi-Fi connected).
- It is powered by a MicroUSB charger or the GPIO header. Any good charger with 5 V via Micro USB or GPIO header will do the work of powering the PI. In our project, we connect Power bank with RPi 3 Model B so that it can be put easily in structure.

2.5 SD Card

The Raspberry Pi does not have any onboard storage available. The operating system is loaded on an SD card which is inserted on the SD card slot on the Raspberry Pi. The operating system can be loaded on the card using a card reader on any computer.

2.6 GPIO – General Purpose Input Output

General – purpose input/output (GPIO) is a generic pin on an integrated circuit whose behavior, including whether it is an input or output pin, can be controlled by the user at runtime.

GPIO capabilities may include:

- GPIO pins can be configured to be input or output
- GPIO pins can be enabled/disabled
- Input values are readable (typically high=1, low=0)
- Output values are writable / readable
- Input values can often be used as IRQs (typically for wakeup events)

The Raspberry Pi 3 Model B board contains a single 40-pin expansion header labeled as 'J8' providing access to 28 GPIO pins as well as +3.3 V, +5 V and GND supply lines.

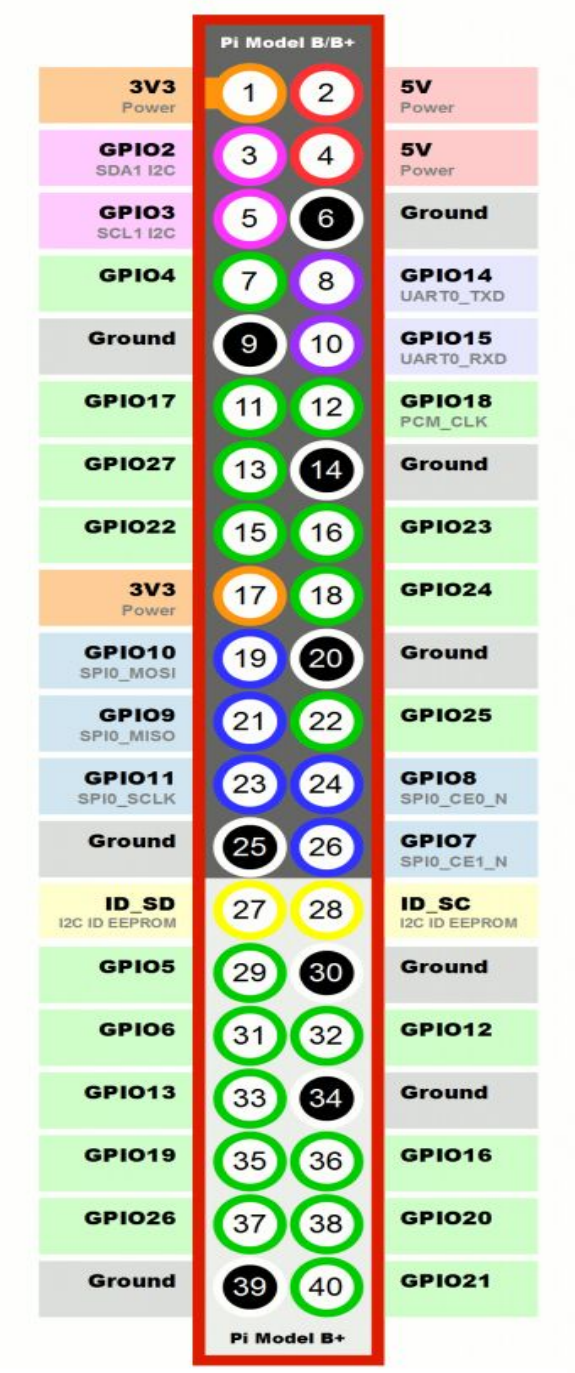


Figure 2. 3: Pin Diagram

2.7 USB 2.0 Port

USB 2.0 ports are the means to connect accessories such as a mouse or keyboard to the Raspberry Pi. There are 4 ports on RPi 3 Model B. The number of ports can be increased by using an external powered USB hub which is available as a standard Pi accessory.

2.8 Ethernet

Ethernet port is available on RPi 3 Model B. It can be connected to a network or internet using a standard LAN cable on the Ethernet port. The Ethernet ports are controlled by Microchip BCM43438 controller chip.

2.8 Ultrasonic Sensor

This sensor is a high-performance ultrasonic range finder. It is compact and measures an amazingly wide range from 2 cm to 4m. This ranger is perfect for any robotic application or any other projects requiring accurate ranging information. This sensor can be connected directly to the digital I/O lines of your microcontroller and distance can be measured in the time required for traveling of a sound signal using simple formula as below. The module works on 5VDC input and also gives an output signal directly for detection of an obstacle up to 4M. As soon as the signals are transmitted the “Echo” pin goes to high level and remains in high level until the same sound waves are received by the receiver. If the received sound waves are same as what the same sensor transmitted then the Echo pin goes to a low level. If no object is detected within 5M after 30ms the Echo signal will automatically go to low level.

Figure 2. 4: Ultrasonic Sensor

2.9 Jumper:

In our proposed system, we have used two types of wires. Male-to-Male jumper and Male-to-Female jumper. Both are shown below:

Figure 2. 5: Male to Male Jumper

Figure 2. 6: Male to Female Jumper

Chapter 3

Overview of the Proposed System

3.1 Structure

The basic structure of the Python language is fairly simple and fully oop supported and single threaded programming language just like other languages. Python features a dynamic type system and automatic memory management.it supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library to interact with GPIO.

The basic structure of the Python class is :

```
class MyClass(object):
     #class variable
     common = 10
     #constructor
     def __init__(self):
         self.myvariable = 3
     #method
     def myfunction(self, arg1, arg2):
         return self.myvariable

#instantiation
classinstance = MyClass()
classinstance.myfunction(1, 2)
```

Python classes facilitate to build user-defined modules. For a specific device, we used a class to build a module like a sensor.py, lcd.py, geocode.py etc which is so handy to use and build projects.

To run sensors we mostly used the infinite loop like this :

```
While True:
      Statements
```

This snippet helps to run a device or component to emit data continuously.

3.2 Main Modules

Our project's main module is **main_app.py** which runs the app.py as a subprocess. This is the only module which runs all the trash sensor. Each time app.py runs inside the run_bin_sensor(trash_id) by calling :

```
subprocess.call(['python3', 'app.py', trash_id])
```

and we used here multiprocessing to run each trash simultaneously. Our another important module is **app.js** written with node.js.And its a server which is serving to the Android App 'The Garbage Collection Notifier'.

3.3 Standard libraries

- **math**: This module is always available. It provides access to the mathematical functions defined by the C standard. These functions cannot be used with complex numbers; use the functions of the same name from the cmath module. Lots of functions are provided by this module such as math.ceil(x), math.round(), math.floor(), math.pow(x, y), math.sqrt(x) etc.
- **sys:** This module provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter.
- **threading:** This module constructs higher-level threading interfaces on top of the lower level _thread module.
- **time:** This module provides various time-related functions.
- **datetime:** The datetime module supplies classes for manipulating dates and times in both simple and complex ways. While the date and time arithmetic are supported, the

focus of the implementation is on efficient attribute extraction for output formatting and manipulation.

- **Rpi.GPIO:** This package provides a class to control the GPIO on a Raspberry Pi. To import the RPi.GPIO module just we need to write the below line on the main module:

```
import RPi.GPIO as GPIO
```

but before using this module we should install it by using this command:

```
sudo pip install RPi.GPIO
```

- **rethinkdb:** RethinkDB is a free and open-source, distributed document-oriented database originally created by the company of the same name. The database stores JSON documents with dynamic schemas and is designed to facilitate pushing real-time updates for query results to applications. This module should also be installed by this:

```
sudo pip install rethinkdb
```

- **twilio.rest:** Twilio is a cloud communications platform as a service company based in San Francisco, California. Twilio allows software developers to programmatically make and receive phone calls, send and receive text messages, and perform other communication functions using its web service APIs

3.4 Functions

We used a number of library functions and also defined a collection of methods under a specific class to implements the project. So those things are being briefly reviewed below:

3.4.1 Library Functions

- **time.sleep(seconds):** Suspend execution of the calling thread for the given number of seconds. The argument may be a floating point number to indicate a more precise sleep time. The actual suspension time may be less than that requested because any caught

signal will terminate the sleep() following the execution of that signal's catching routine. Also, the suspension time may be longer than requested by an arbitrary amount because of the scheduling of other activity in the system.

- **GPIO.setmode(GPIO.BOARD) :**

To select pins as BOARD numbering this is called. There are two ways of numbering the IO pins on a Raspberry Pi within RPi.GPIO. The first is using the BOARD numbering system. This refers to the pin numbers on the P1 header of the Raspberry Pi board. The advantage of using this numbering system is that your hardware will always work, regardless of the board revision of the RPi. You will not need to rewire your connector or change your code

- **GPIO.setmode(GPIO.BCM):**

To select pins as GPIO numbering this is called. The second numbering system is the BCM numbers. This is a lower level way of working. it refers to the channel numbers on the Broadcom SOC. You have to always work with the diagram of which channel number goes to which pin on the RPi board. Your script could break between revisions of Raspberry Pi boards

- **GPIO.setup(port_or_pin, GPIO.IN/GPIO.OUT) :**

To set port/pin as an input or as an output. For example, to set Echo pin of the Ultrasonic sensor we should call this function providing echo number as the first parameter and the second parameter is GPIO.OUT.

- **GPIO.output(port_or_pin, 1) :**

Using this function set an output port/pin value to 1/GPIO.HIGH/True

- **GPIO.cleanup()** :

At the end of any program, it is good practice to clean up any resources you might have used. This is no different with RPi.GPIO. By returning all channels you have used back to inputs with no pull-up/down, you can avoid accidental damage to your Orange Pi by shorting out the pins. Note that this will only clean up GPIO channels that your script has used. Note that GPIO.cleanup() also clears the pin numbering system in use.

3.4.2 User-defined Functions

- **main():**

  This function creates a trash object from the sensor class and calls the other two functions. The first one is **databage_change(trash_id)** function called inside a thread and the second on is **run_sensor(trash).**

- **run_sensor(trash):**

  It takes a trash object as a parameter and runs the trash sensor. Inside this function tune_sensor() is called to find the bin depth if depth is not tuned before.

- **databage_change(trash_id):**

  This function is run on a thread to find out the database changes. It takes trash id as parameter and pause or resumes a sensor of trash based on the status attribute of the database using the given trash id. It also triggers the tune_now() function to tune sensor immediately when the user presses the tune button on the android app.

- **update_bin_info(trash_id, attr, info):**

  It updates a specific database attribute of particular trash.

- **clean_counter(trash_id, garbage_level):**

  This function will count the total clean time and save the last clean time and date into rethinkdb.

- **track_location():**

  This is the location tracker of trash .it makes a request to the ipinfo.io and with the response, it gives the latitude and longitude of that location.

- **measure_distance(gpio, time):**

  Distance is measured using the ultrasonic sensor in centimeter. It takes GPIO and time library module as a parameter.

❑ garbage_calc():

It calculates the garbage level using the distance which is measured by the measure_distance() function.

❑ tune_sensor():

It measures the distance 10 times and calculate the average of those distances and finally set the average to the bin_depth variable.

❑ send_sms(message, send_from, send_to):

Send SMS to the send_to based on the notify level of the trash. It also embeds the trash location to the message body.

3.5 Design Diagrams

UML is a way of visualizing a software program using a collection of diagrams. The notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar Jacobson, and the Rational Software Corporation to be used for object-oriented design, but it has since been extended to cover a wider variety of software engineering projects. The current UML standards call for 13 different types of diagrams: class, activity, object, use case, sequence, package, state, component, communication, composite structure, interaction overview, timing, and deployment.In our proposed system, we have designed 2 diagrams named Use Case Diagram and another is Activity Diagram**.**

- **Use Case Diagram:** A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. So we can say, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

Our proposed systems Use Case Diagram is given below:

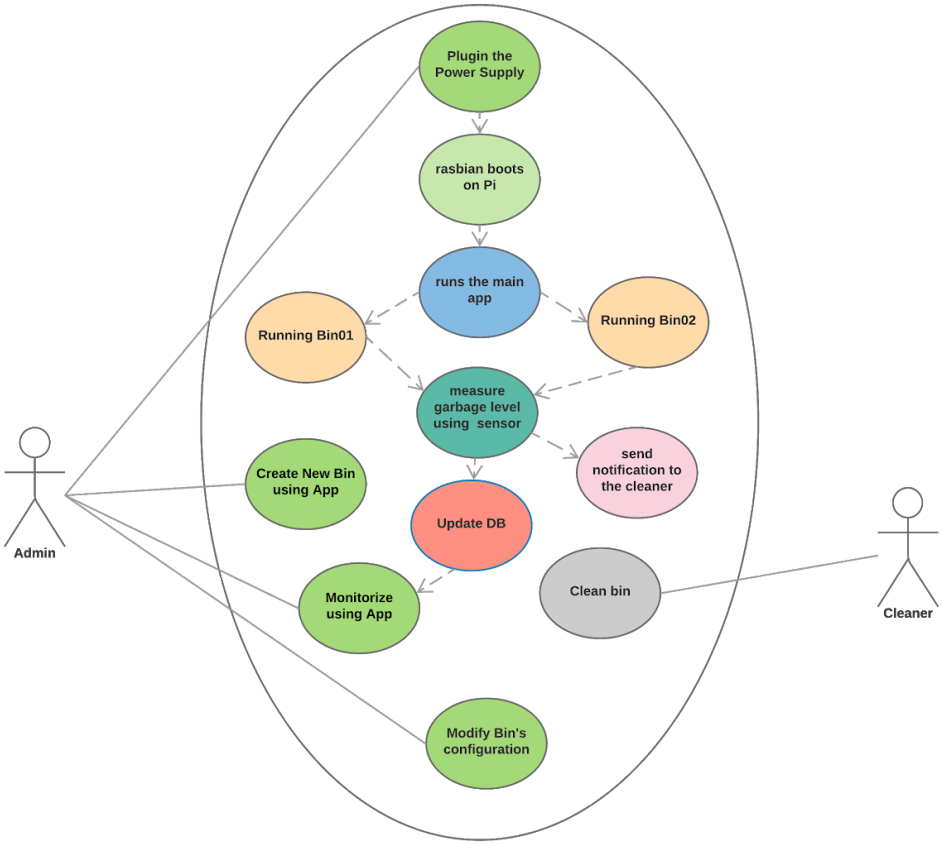
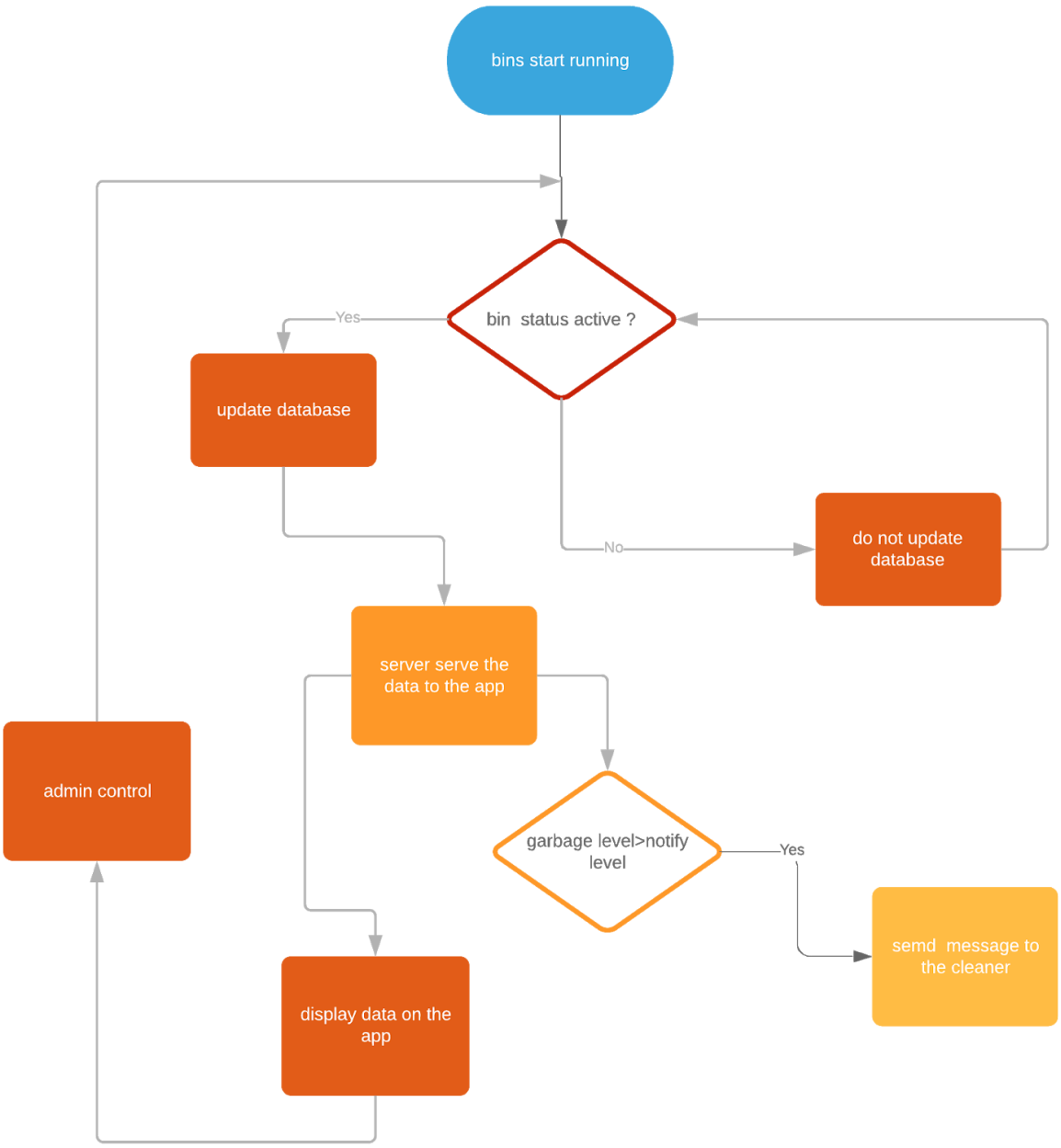


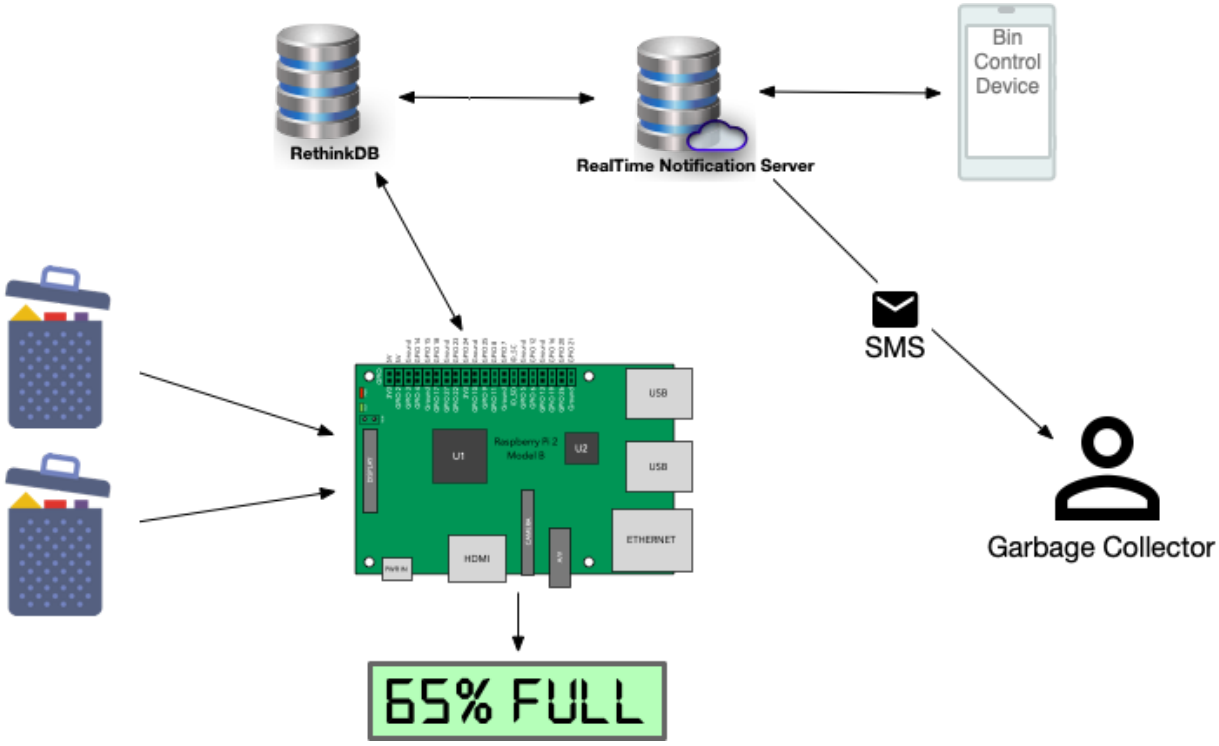
Figure 3.1: Use Case Diagram

- **Activity Diagram:** Activity diagrams illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation.

Our proposed system's Activity Diagram is given below:


bins start running
bin status active ?
Yes
No
update database
do not update database
server serve the data to the app
admin control
garbage level>notify level
Yes
semd message to the cleaner
display data on the app

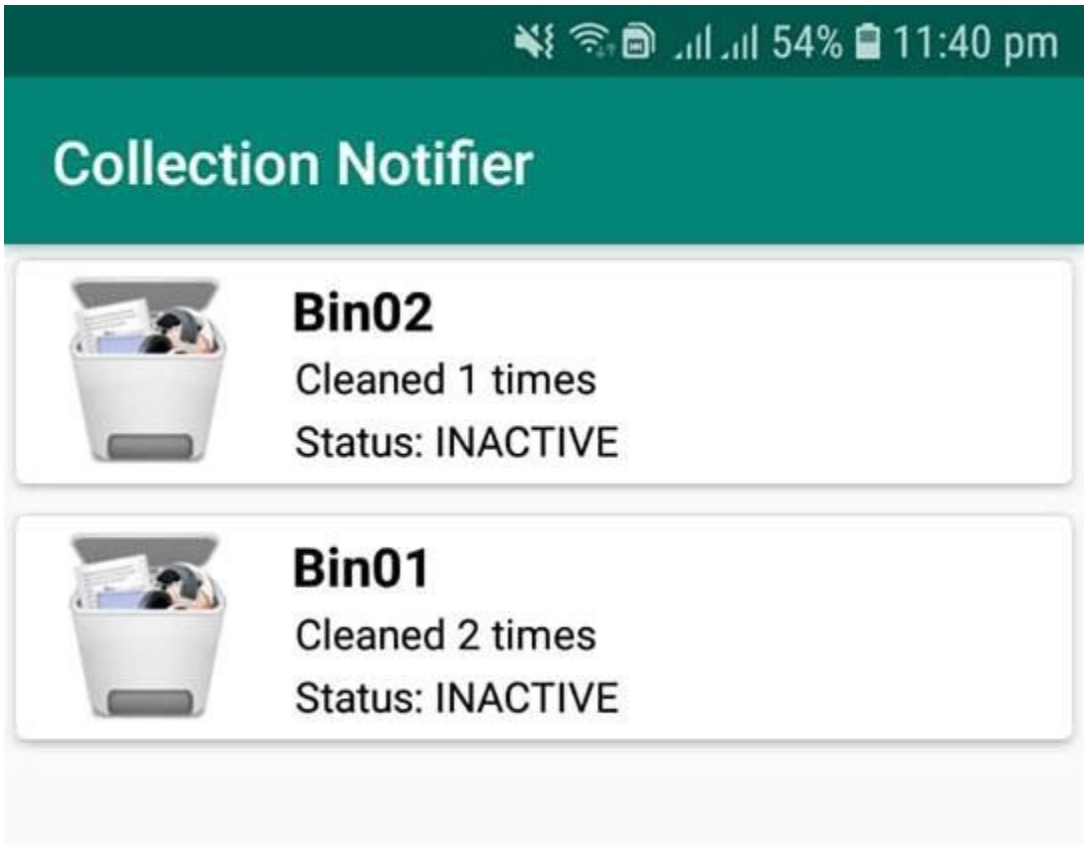
Block Diagram:



Chapter 4

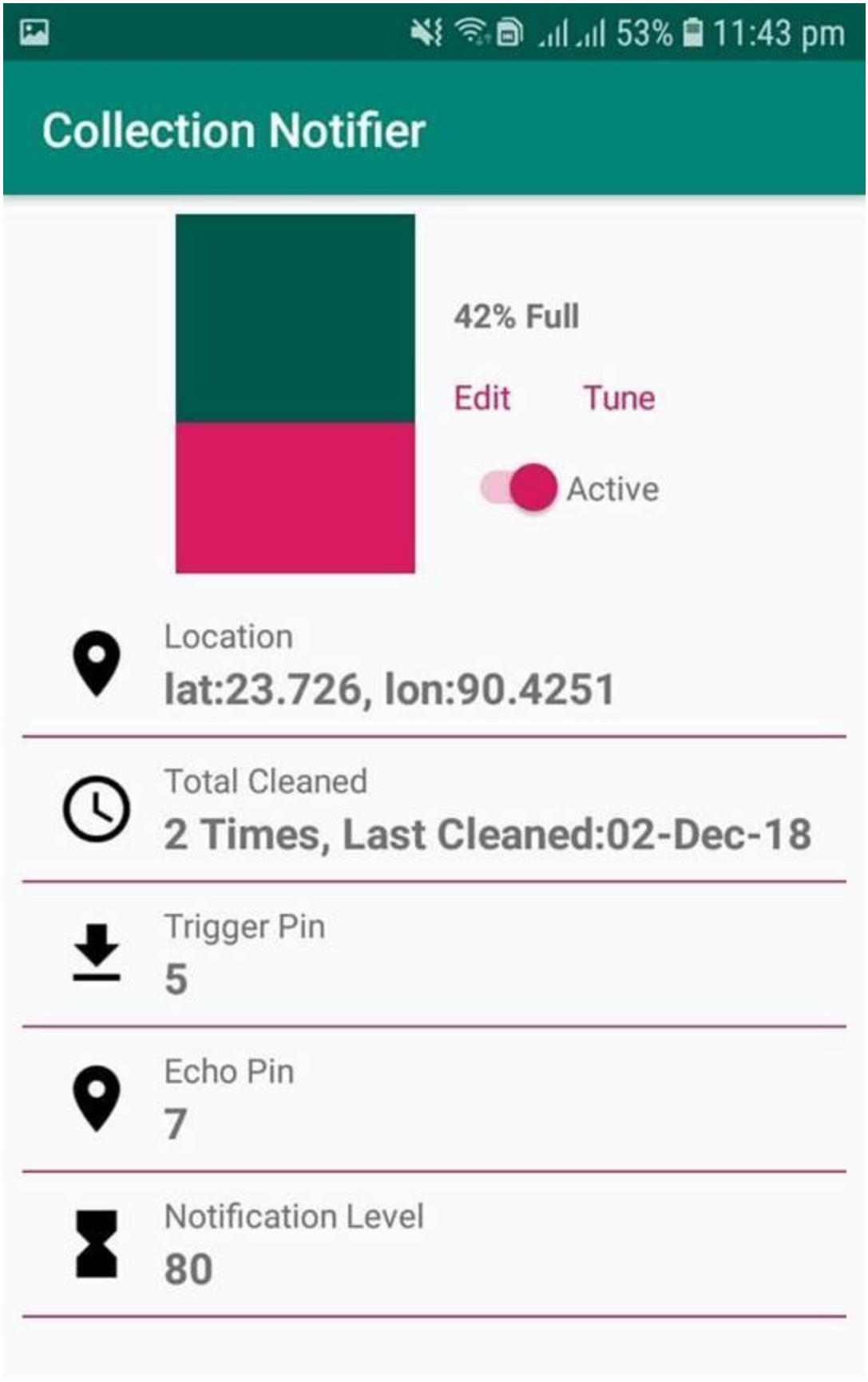
4.1 Home Layout :

In android user workforce or clint can see the details about dustbin.Here is the screenshot of android app.



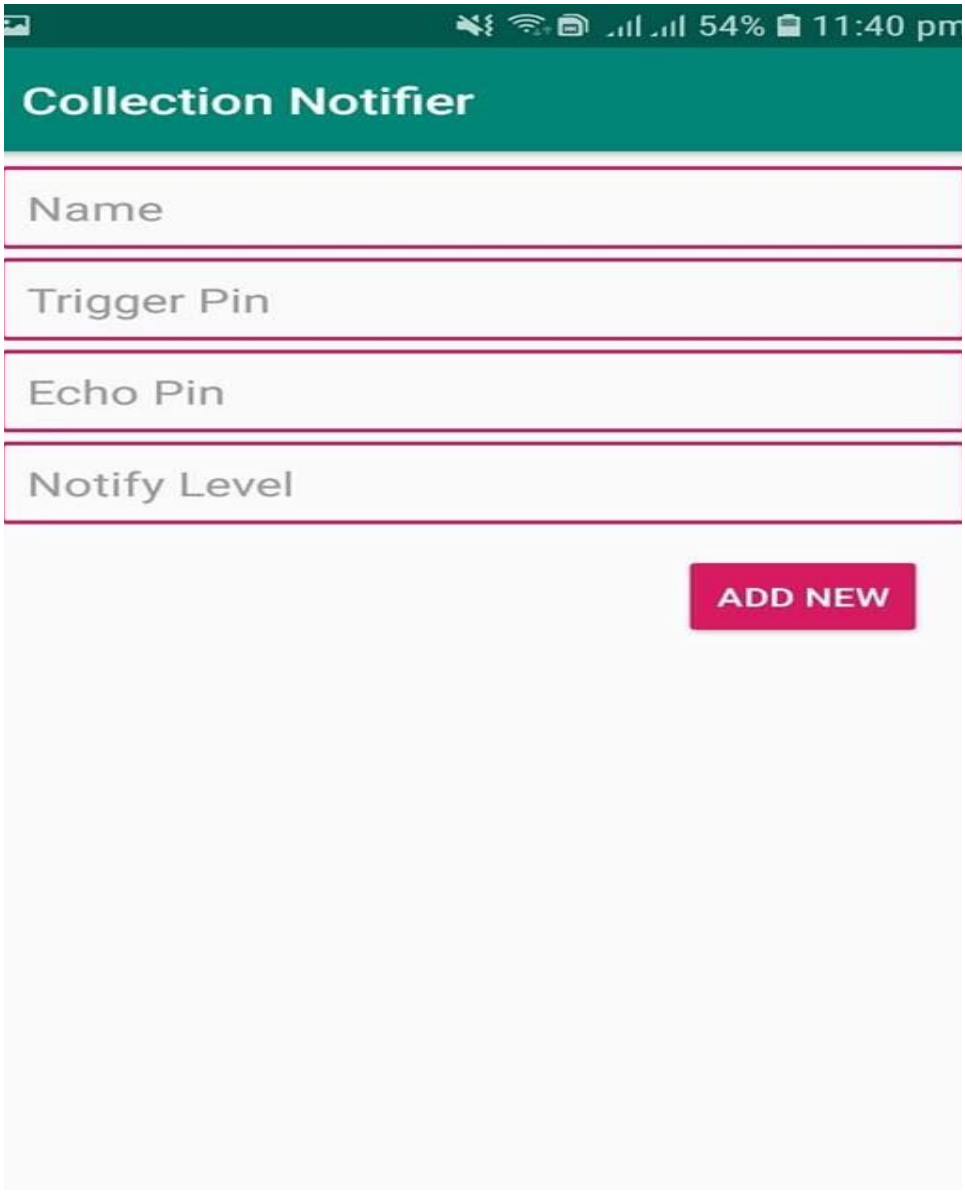
4.2 Bin Details Layout:

In the Bin layout there are few features those provides location,cleaning time and date,trigger pin,eco pin and level of garbage filled in the dustbin.There are also active and inactive button.By using those user/workforce can pause or restart the bin


53% 11:43 pm
Collection Notifier
42% Full
Edit
Tune
Active
Location
lat:23.726, lon:90.4251
Total Cleaned
2 Times, Last Cleaned:02-Dec-18
Trigger Pin
5
Echo Pin
7
Notification Level
80

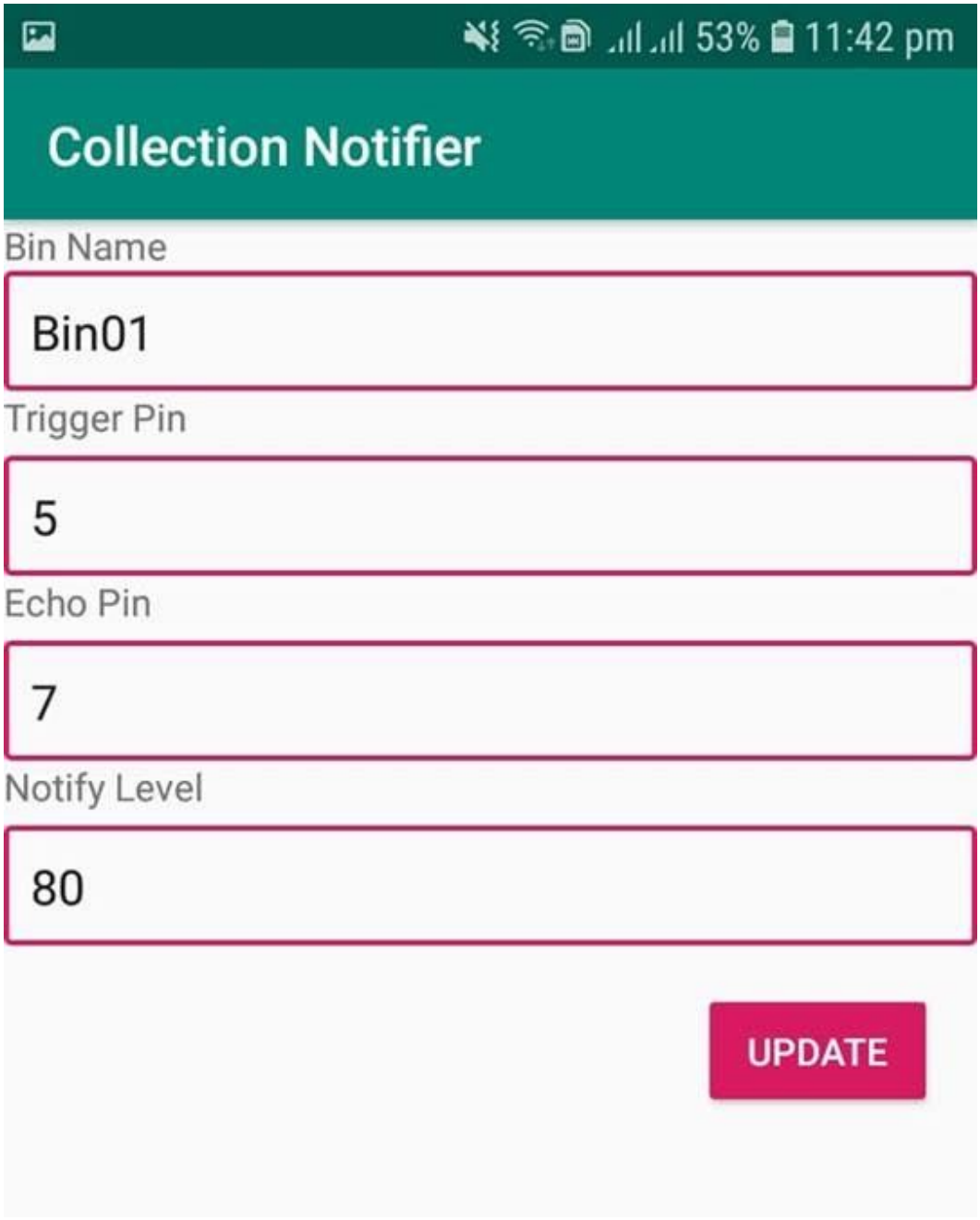
4.3 Add New Bin

If users/workforces need to add multiple dustbin,they can easily add by giving the Bin name ,by setting the trigger and echo pin and they can a notify level so what this system can automatically notify them



4.4 Update Bin Info Layout :

Users/Workforce can simply change/update their bin name,trigger pin,echo pin and level of dustbin filled if needed.


53% 11:42 pm
Collection Notifier
Bin Name
Bin01
Trigger Pin
5
Echo Pin
7
Notify Level
80
UPDATE