

VISVESVARAYA TECHNOLOGICAL UNIVERSITY  
JNANASANGAMA, BELAGAVI - 590018



## ”SMART GARBAGE SYSTEM”

This is submitted in partial fulfillment of the curriculum prescribed  
for the award of the degree of Bachelor of Engineering in  
Computer Science & Engineering by

1CR14CS155	Vani K
1CR14CS061	Keerthy Kumari V
1CR15CS409	Devika R S
1CR15CS428	Ramya R

Under the Guidance of

Ms. Gopika D  
Assistant Professor  
Department of CSE, CMRIT, Bengaluru



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
#132, AECS LAYOUT, IT PARK ROAD, BENGALURU - 560037

2017-18

VISVESVARAYA TECHNOLOGICAL UNIVERSITY  
JNANASANGAMA, BELAGAVI - 590018



## Certificate

This is to certify that the project entitled "SMART GARBAGE SYSTEM" is a bonafide work carried out by **Vani K, Keerthy Kumari V, Devika R S & Ramya R** in partial fulfillment of the award of the degree of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belgaum, during the year 2017-18. It is certified that all corrections / suggestions indicated during reviews have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

----- Signature of Guide <b>Ms. Gopika D</b> Assistant Professor Department of CSE CMRIT, Bengaluru - 37	----- Signature of HoD <b>Dr. Jhansi Rani P</b> Professor & Head Department of CSE CMRIT, Bengaluru - 37	----- Signature of Principal <b>Dr. Sanjay Jain</b> Principal CMRIT, Bengaluru - 37
---	---	--

### External Viva

Name of the Examiners	Institution	Signature with Date
1. -----	-----	-----
2. -----	-----	-----

VISVESVARAYA TECHNOLOGICAL UNIVERSITY  
JNANASANGAMA, BELAGAVI - 590018



## Declaration

We **Ms.Vani K, Ms.Keerthy Kumari V, Ms.Devika R S & Ms.Ramya R**, bonafide students of CMR Institute of Technology, Bangalore, here by declare that the dissertation entitled "**SMART GARBAGE SYSTEM**" has been carried out by us under the guidance of **Ms. Gopika D**, Assistant Professor of CSE Department, CMRIT, Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in **Computer Science Engineering**, of the Visvesvaraya Technological University, Belgaum during the academic year 2017-2018. The work done in this dissertation report is original and it has not been submitted for any other degree in any University.

Vani K  
Keerthy Kumari V  
Devika R S  
Ramya R

# Acknowledgement

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without mentioning people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So,with gratitude we acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

We would like to thank **Dr. Sanjay Jain**, Principal, CMRIT, Bangalore for providing excellent academic environment in the college and his never-ending support for the B.E program.

We would like to thank **Dr. Jhansi Rani**, Professor & HOD , Department of Computer Science, CMRIT, Bangalore who shared their opinions and experiences through which we received the required information crucial for the project.

We would also like to thank **Mr. Sudhakar K N** , Associate Professor & Project Coordinator, Department of Computer Science, CMRIT, Bangalore who shared his opinions and experiences through which we received the required information crucial for the project.

We consider it a privilege and honour to express my sincere gratitude to my internal guide **Ms. Gopika D**, Assistant Professor, Department of Computer Science & Engineering, for his valuable guidance throughout the tenure of this project work.

We would also like to thank all the faculty members who have always been very co-operative and generous. Conclusively, we also thank all the non-teaching staff and all others who have done immense help directly or indirectly during my project

Vani K

Keerthy Kumari V

Devika R S

Ramya R

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>1 PREAMBLE</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Problem Statement . . . . .	2
1.3 Organization of the project report . . . . .	3
<b>2 LITERATURE SURVEY</b>	<b>5</b>
<b>3 THEORETICAL BACKGROUND</b>	<b>8</b>
3.1 Arduino . . . . .	8
3.2 Micro controller . . . . .	9
3.3 Gas Sensor . . . . .	9
3.4 GPS Module . . . . .	10
3.5 DC Motor Driver (L239D) . . . . .	10
3.6 DC Motor . . . . .	11
3.7 LCD . . . . .	11
3.8 PIR Sensor . . . . .	11
3.9 Voltage Transformer . . . . .	12
3.10 Moisture Sensor . . . . .	12
3.11 Embedded C Language . . . . .	12
3.12 Arduino 1.8.5 Software . . . . .	13
<b>4 SYSTEM REQUIREMENT SPECIFICATION</b>	<b>14</b>
4.1 Functional Requirements . . . . .	14
4.2 Non-Functional Requirements . . . . .	15

4.3	Hardware Requirements . . . . .	18
4.4	Software Requirements . . . . .	18
<b>5</b>	<b>SYSTEM ANALYSIS</b>	<b>19</b>
5.1	Overview . . . . .	19
5.2	Feasibility Study . . . . .	19
<b>6</b>	<b>SYSTEM DESIGN</b>	<b>21</b>
6.1	System development methodology . . . . .	21
6.2	Design Using UML . . . . .	23
<b>7</b>	<b>IMPLEMENTATION</b>	<b>25</b>
7.1	Arduino IDE . . . . .	25
7.2	ESP8266 Arduino Core . . . . .	26
7.3	Cloud Computing . . . . .	26
<b>8</b>	<b>TESTING</b>	<b>27</b>
8.1	Testing Methodologies . . . . .	27
8.2	Unit Testing . . . . .	29
8.3	Integration Testing . . . . .	29
8.4	System Testing . . . . .	29
8.5	Quality Assurance . . . . .	30
8.6	Test Cases . . . . .	33
<b>9</b>	<b>RESULTS &amp; EXECUTION</b>	<b>35</b>
9.1	Snapshots . . . . .	35
<b>10</b>	<b>CONCLUSION &amp; FUTURE ENHANCEMENT</b>	<b>41</b>
10.1	Conclusion . . . . .	41
10.2	Future Enhancement . . . . .	42
	<b>REFERENCES</b>	<b>43</b>

# List of Figures

6.1	Waterfall model . . . . .	23
6.2	System Architecture . . . . .	24
9.1	Hardware . . . . .	35
9.2	Login Page . . . . .	36
9.3	Connect Channel Address . . . . .	36
9.4	Home Page . . . . .	37
9.5	Dust Bin Tracker Automation . . . . .	37
9.6	Longitudes . . . . .	38
9.7	Lattitudes . . . . .	38
9.8	Gas . . . . .	39
9.9	Wet Waste Bin . . . . .	39
9.10	Dry Waste Bin . . . . .	40

# List of Tables

1.1	Phase Description . . . . .	3
8.1	Test case table1 . . . . .	34

## Abstract

Many times, in our city we see that the garbage bins or dustbins placed at public places are overloaded. It creates unhygienic conditions for people as well as ugliness to that place leaving bad smell. To avoid all such situations we are going to implement a project called IoT Based Smart Garbage and Waste Collection bins. These dustbins are interfaced with microcontroller based system having IR wireless systems along with central system showing current status of garbage, on mobile web browser with html page by Wi-Fi. Hence the status will be updated on to the html page. Major part of our project depends upon the working of the Wi-Fi module; essential for its implementation. The main aim of this project is to reduce human resources and efforts along with the enhancement of a smart city vision.

# Chapter 1

## PREAMBLE

### 1.1 Introduction

In recent times, garbage disposal has become a huge cause for concern in the world. A voluminous amount of waste that is generated is disposed by means which have an adverse effect on the environment. The common method of disposal of the waste is by unplanned and uncontrolled open dumping at the landfill sites. This method is injurious to human health, plant and animal life. This harmful method of waste disposal can generate liquid leachate which contaminate surface and ground waters can harbour disease vectors which spread harmful diseases and can degrade aesthetic value of the natural environment and it is an unavailing use of land resources.

In India, rag pickers play an important role in the recycling of urban solid waste. Rag pickers and conservancy staff have higher morbidity due to infections of skin, respiratory, gastrointestinal tract and multisystem allergic disorders, in addition to a high prevalence of bites of rodents, dogs and other vermin. Dependency on the rag-pickers can be diminished if segregation takes place at the source of municipal waste generation. The economic value of the waste generated is not realised unless it is recycled completely. Several advancements in technology has also allowed the refuse to be processed into useful entities such as Waste to Energy, where the waste can be used to generate synthetic gas (syngas) made up of carbon monoxide and hydrogen. The gas is then burnt to produce electricity and steam, Waste to Fuel, where the waste can be utilized to generate bio fuels.

When the waste is segregated into basic streams such as wet, dry and metallic, the waste has a higher potential of recovery, and consequently, recycled and reused. The wet waste fraction is often converted either into compost or methane-gas or both. Compost can replace demand for chemical fertilizers, and biogas can be used as a

source of energy. The metallic waste could be reused or recycled.

Even though there are large scale industrial waste segregators present, it is always much better to segregate the waste at the source itself. The benefits of doing so are that a higher quality of the material is retained for recycling which means that more value could be recovered from the waste. The occupational hazard for waste workers is reduced. Also, the segregated waste could be directly sent to the recycling and processing plant instead of sending it to the segregation plant then to the recycling plant.

Currently there is no system of segregation of dry, wet and metallic wastes at a household level. J.S. Bajaj has recommended that a least cost, most appropriate technological option for safe management should be developed. The purpose of this project is the realization of a compact, low cost and user friendly segregation system for urban households to streamline the waste management process.

We are implementing a smart dustbin which is a cheap, easy to use solution for a segregation system at households, so that it can be sent directly for processing. It is designed to sort the refuse into metallic waste, wet waste and dry waste. The mixed waste is sorted based on the following methods at the industrial level. Larger items are removed by manual sorting. Then the refuse is sorted based on its size by using large rotating drums which is perforated with holes of a certain size. Materials smaller than the diameter of the holes will be able to drop through, but larger particles will remain in the drum. For metallic objects electromagnets or eddy current based separators can be used. Near infrared scanners are used to differentiate between various types of plastics based on the ability of the material to reflect light. X-rays can also be used to segregate materials based on their density.

## 1.2 Problem Statement

1.2 Problem Statement The main sources of waste are industrial and domestic waste. This project mainly concentrate on domestic waste whose value is unrecognized since people dont spend time on segregating waste into their basic streams. The wet waste generated can be used to generate biogas, metallic and dry waste can be send for recycling, if metallic waste is left untreated then it becomes a threat to animal and plant lives. If waste is separated at household level then they can be directly sent for recycling instead of sending them to industries first for segregation which becomes a huge task and the waste does not get segregated accurately. The methods adopted for waste segregation in industries is hazardous to human health since it makes use of

x-rays and infrared rays.

The environmental risks associated with poor waste management are well known and understood. Fly tipped wastes can poison and injure children and animals as well as create an eyesore. Careless disposal of liquid wastes such as solvents can leach into the ground water and contaminate drinking water supplies. Poorly planned and managed landfills will create a significant neighborhood nuisance, and where landfill gas and leachate are not properly treated there will be a serious threat to the safety of local residents. Incinerators operated without adequate pollution abatement equipment will release highly toxic dioxins. Even recycling and composting facilities can be a source of litter and unpleasant odor if not properly regulated. The main aim of the project is to segregate waste at source level to wet, dry and metallic such that waste is not wasted but there value is understood and can be converted to a source of energy, in a cost effective way.

Table 1.1: Phase Description

Phase	Task	Description
Phase1	Analysis	Analyzing the core of the IEEE paper and provide Literature review based on analysis.
Phase2	Literature survey	Collect raw data and elaborate on literature surveys.
Phase3	System analysis	Analyses the requirements of the project and lists the specific requirements needed.
Phase4	Design	Object designing and Functional description .
Phase5	Implementation	Implement the code based on the object specification .
Phase6	Testing	Test the project according to Test Specification .
Phase7	Documentation	Prepare the document for this project with conclusion and future enhancement.

### 1.3 Organization of the project report

The project report is organized as follows:

**Chapter 2: Literature Review** - Gives a brief overview of the survey papers and the research sources that have been studied to establish a thorough understanding of  
Dept Of CSE - Feb - May 2018

the project under consideration.

**Chapter 3: Theoretical Background** - Establishes groundwork for the proposed project by giving a detailed analysis of the project topic, existing research relevant to the project, arguments in favor and against the existing solutions and finally explores the motivation behind the proposed solution.

**Chapter 4: System Requirement Specification** - Discusses in details about the different kinds of requirements needed to successfully complete the project.

**Chapter 5: System Analysis** - gives details about several analysis that are performed to facilitate taking decision of whether the project is feasible enough or not.

**Chapter 6: System Design** - Gives the design description of the project, conceptual and detailed design well supported with design diagrams.

**Chapter 7: Implementation** - Discusses the implementation details of the project and reasons the use of the programming language and development environment.

**Chapter 8: Testing** - Briefs the testing methods used for testing the different modules in the project.

**Chapter 9: Results and Performance Analysis** - Gives the snapshots and graphs of the proposed protocols.

**Chapter 10: Conclusion and Future Scope** - Gives the concluding remarks of the project, throwing light on its future aspects.

# Chapter 2

## LITERATURE SURVEY

Literature survey is mainly carried out in order to analyze the background of the current project which helps to find out flaws in the existing system and guides on which unsolved problems we can work out. So, the following topics not only illustrate the background of the project but also uncover the problems and flaws which motivated to propose solutions and work on this project.

[1] Claudine Capel, **INNOVATIONS IN WASTE**, **Waste management-world**, Volume 11, Issue 2, Mar 2010.

### **Source separation of waste:**

The waste hierarchy as laid out in European law states the ideal chain of events when it comes to waste is reduce, reuse, recycle, energy recovery, and dispose, and it is interesting to look at the wide variation of systems in Europe today for citizens disposing of their household waste. Where some countries such as Germany and the Netherlands have had efficient methods in place for years, other countries still have the majority of residents throwing all their household waste into one bin and leaving it for the local authority to separate it. It seems that more stringent measures need to come into play to ensure that the waste hierarchy is followed wherever possible.

While streams of mixed MSW can be collected and then separated into the various components, i.e. recyclable items and organic waste, it is much better to separate the waste stream at the source. This has several benefits:

- Maintains a higher quality of material for recycling, meaning there is more value to be recovered,
- Decreases the occupational risks for waste workers, and

→ Means that waste can most often be sent straight to the correct place for processing, instead of one facility to be separated and then another to be processed.

There are many separation schemes in effect across the world and it depends on each municipality as to what will work best. The collection of food scraps into a separate bin is one of the most common and has an important role to play in making sure organic waste does not end up in landfill. It also means that biowaste can be turned into compost or biodegraded in a safe manner without emitting harmful gases. Systems for separating glass bottles, aluminium cans and plastics also mean that recycling becomes easier, safer and more efficient.

### **Waste to fuel:**

Given the oil crisis and the ever-increasing price of fossil fuel, turning waste into fuel is a fantastic solution. Bio fuel is the most common form, and the term encompasses a range of different fuels derived from organic matter, including biowaste. Biofuel can be solid, liquid or gas and be used to power vehicles or used to enhance other types of fuel. Biogas a product of anaerobic digestion and syngas which is produced during gasification are both types of bio fuel.

Landfill gas also has an up-and-coming role in this field. Most landfill-gas-to-energy projects involve turning otherwise harmful emissions into electricity to power homes. But it is also being increasingly used as a vehicle fuel or as a substitute for mains household gas supply.

### **Sorting technology:**

People will always, either through ignorance or carelessness, throw their waste in the wrong bin every now and then. So, however good our separate collection schemes may be and let us remember that it is not always practical to have them in place; we need a way to take a mixed waste stream and divide into reusable, separate waste streams. Enter one of the greatest innovations in waste technology the sorter. When mixed waste is fed into a single stream recycling facility the process will include some or all of these processes:

→ Removal of larger items by hand

→ Separation of items by weight, which means metals, plastics, paper, glass etc. are sorted from each other

→ Use of screens to separate items by size

→ Magnetic separation of metals, such as eddy current separators for aluminium

→ Ultraviolet optical scanners combined with targeted air jets that send items of certain types in separate collection bins e.g. PET and non PET plastics.

## [2] Claudine Capel, WASTE SORTING - A LOOK AT THE SEPARATION AND SORTING TECHNIQUES IN TODAY'S EUROPEAN MARKET

Separating the different elements found in waste streams is essential for enabling the recovery of useful materials, minimizing the amount of material sent to landfill and allowing recyclable materials to find a new incarnation. Companies sort and recycle materials in order to extract value. Separation technologies:

Waste disposal companies dealing with the sorting of materials will commonly use one or more of these five methods:

→ **Drum screens:** These separate materials according to their particle size. Waste is fed into a large rotating drum which is perforated with holes of a certain size. Materials smaller than the diameter of the holes will be able to drop through, but larger particles will remain in the drum.

→ **Eddy current separator:** This method is specifically for the separation of metals. An eddy current occurs when a conductor is exposed to a changing magnetic field. Put simply, it is an electromagnetic way of dividing ferrous and non-ferrous metals.

→ **Induction sorting:** Material is sent along a conveyor belt with a series of sensors underneath. These sensors locate different types of metal which are then separated by a system of fast air jets which are linked to the sensors.

→ **Near infrared sensors (NIR):** When materials are illuminated they mostly reflect light in the near infrared wavelength spectrum. The NIR sensor can distinguish between different materials based on the way they reflect light.

→ **X-ray technology:** X-rays can be used to distinguish between different types of waste based on their density.

# Chapter 3

## THEORETICAL BACKGROUND

Theoretical background highlighting some topics related to project work. The description contains several topics which are worth to discuss and also highlight some of their limitation that encourage going on finding solution as well as highlights some of their advantages for which reason these topics and their features are used in this project.

### 3.1 Arduino

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

There are various types of arduinos. Of all the type of arduinos we are using NodeMCU10 in this project. This Node arduino consist of 12pins. It also has an in-built wiFi module in it. It has a storage of 32kb of memory.

## 3.2 Micro controller

The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable and erasable read only memory (PEROM). The on-chip Flash allows the program memory to be reprogrammed in-system or by a Conventional nonvolatile memory programmer.

→ Compatible with MCS-51 Products 8K Bytes of In-System Reprogrammable Flash Memory.

→ Endurance: 1,000 Write/Erase Cycles.

→ Fully Static Operation: 0 Hz to 24 MHz.

→ Three-level Program Memory Lock.

→ 256 x 8-bit Internal RAM.

→ 32 Programmable I/O Lines.

→ Three 16-bit Timer/Counters.

→ Full Duplex Serial port.

## 3.3 Gas Sensor

This module is made using Gas Sensor . It is a low cost semiconductor sensor which can detect the presence of alcohol gases at concentrations from 0.05 mg/L to 10 mg/L. The sensitive material used for this sensor is SnO<sub>2</sub>, whose conductivity is lower in clean air. Its conductivity increases as the concentration of gases increases. Its conductivity increases as the concentration of gases increases. It has high sensitivity to alcohol and has a good resistance to disturbances due to smoke, vapour and

gasoline. Gas sensor module can be easily interfaced with Microcontrollers.

This gas sensor is suitable for detecting gas concentration on the air, just like your common breathalyzer. It has a high sensitivity and fast response time. Sensor provides an analog resistive output based on gas concentration. The drive circuit is very simple, all it needs is one resistor. A simple interface could be a 0-3.3V ADC.

## 3.4 GPS Module

The Global Positioning System (GPS), originally Navstar GPS, is a satellite-based radionavigation system owned by the United States government and operated by the United States Air Force. It is a global navigation satellite system that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. Obstacles such as mountains and buildings block the relatively weak GPS signals.

The GPS does not require the user to transmit any data, and it operates independently of any telephonic or internet reception, though these technologies can enhance the usefulness of the GPS positioning information. The GPS provides critical positioning capabilities to military, civil, and commercial users around the world. The United States government created the system, maintains it, and makes it freely accessible to anyone with a GPS receiver.

## 3.5 DC Motor Driver (L239D)

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors. To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included. This device is suitable for use in switching applications at frequencies up to 5 kHz.

The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heat sinking The L293D is assembled in a 20 lead surface

mount which has 8 center pins connected together and used for heat sinking.

## 3.6 DC Motor

A DC motor is any of a class of electrical machines that converts direct current electrical power into mechanical power. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor. Most types produce rotary motion, a linear motor directly produces force and motion in a straight line. The very basic construction of a dc motor contains a current carrying armature which is connected to the supply end through commutator segments and brushes and placed within the north south poles of a permanent or an electro-magnet.

## 3.7 LCD

LCD (Liquid Crystal Display) screen is an electronic display module and a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical, easily programmable; have no limitation of displaying special and even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

## 3.8 PIR Sensor

A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. They work entirely by detecting infrared radiation emitted by or rejected from objects. The term passive in this instance refers to the fact that PIR devices do not generate or radiate energy for detection purposes. They work entirely by detecting infrared radiation emitted by or rejected from objects. They

do not detect or measure "heat". PIR sensor is placed in the front and rear doors of the public bus to monitor the passengers entering and exiting the bus.

### 3.9 Voltage Transformer

Voltage transformer gets used in electrical power system for stepping down the system voltage to a safe value which can be fed to low ratings meters and relays. Commercially available relays and meters used for protection and metering, are designed for low voltage. This is a simplest form of potential transformer definition.

The Voltage Transformer can be thought of as an electrical component rather than an electronic component. A transformer basically is very simple static (or stationary) electro-magnetic passive electrical device that works on the principle of Faradays law of induction by converting electrical energy from one value to another. The transformer does this by linking together two or more electrical circuits using a common oscillating magnetic circuit which is produced by the transformer itself. A transformer operates on the principals of electromagnetic induction, in the form of Mutual Induction.

### 3.10 Moisture Sensor

The rain sensor module is an easy tool for rain detection. It can be used as a switch when raindrop falls through the raining board and also for measuring rainfall intensity. The module features, a rain board and the control board that is separate for more convenience, power indicator LED and an adjustable sensitivity though a potentiometer. The analog output is used in detection of drops in the amount of rainfall. Connected to 5V power supply, the LED will turn on when induction board has no rain drop, and DO output is high. When dropping a little amount water, DO output is low, the switch indicator will turn on. Brush off the water droplets, and when restored to the initial state, outputs high level.

### 3.11 Embedded C Language

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems. There are many popular

programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability. An Embedded System can be best described as a system which has both the hardware and software and is designed to do a specific task. A good example for an Embedded System, which many households have, is a Washing Machine.

## 3.12 Arduino 1.8.5 Software

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them. Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

### Summary

This chapter mainly concentrates on the basic theoretical background related to the topic of focus. It gives information about the platform on which this application has been developed in this chapter.

# Chapter 4

## SYSTEM REQUIREMENT SPECIFICATION

Software requirement Specification is a fundamental document, which forms the foundation of the software development process. It not only lists the requirements of a system but also has a description of its major feature. An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and nonfunctional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be.

### 4.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may

include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

- Input test case must not have compilation and runtime errors.
- The application must not stop working when kept running for even a long time.
- The application must function as expected for every set of test cases provided.
- The application should generate the output for given input test case and input parameters.
- The application should generate on-demand services.

## 4.2 Non-Functional Requirements

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:-

- Product Requirements.
- Organizational Requirements.
- Basic Operational Requirements.

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. This should be contrasted with functional requirements that define specific behaviour or functions. The plan for implementing non-functional requirements is detailed in the system architecture. Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of system shall do {requirement<sub>i</sub>}, an individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of system shall be {requirement<sub>j</sub>}, an overall property of the system as a whole or of a particular aspect and not a specific function. The systems' overall properties commonly mark the difference between whether the development project has succeeded or failed.

Non-functional requirements of our project include:

→ **Response time** The time the system takes to load and the time for responses on any action the user does.

→ **Processing time** - How long is acceptable to perform key functions or export / import data?

→ **Throughput** The number of transactions the system needs to handle must be kept in mind.

→ **Storage** - The amount of data to be stored for the system to function.

→ **Architectural Standards** The standards needed for the system to work and sustain.

### 4.2.1 Product Requirements

→ **Correctness:** It follows a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.

→ **Ease of Use:** The front end is designed in such a way that it provides an interface which allows the user to interact in an easy manner.

Non functional requirements are also called the qualities of a system. These qualities can be divided into execution quality & evolution quality. Execution qualities are security & usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

### 4.2.2 Organizational Requirements

**Process Standards:** IEEE standards are used to develop the application which is the standard used by the most of the standard software developers all over the world.

**Design Methods:** Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs.

### 4.2.3 Basic Operational Requirements

Operational requirement is the process of linking strategic goals and objectives to tactic goals and objectives. It describes milestones, conditions for success and explains how, or what portion of, a strategic plan will be put into operation during a given operational period, in the case of, a strategic plan will be put into operation during a given operational period, in the case of commercial application, a fiscal year or another given budgetary term. An operational plan is the basis for, and justification of an annual operating budget request. Therefore, a five-year strategic plan would typically require five operational plans funded by five operating budgets.

Operational plans should establish the activities and budgets for each part of the organization for the next 1-3 years. They link the strategic plan with the activities the organization will deliver and the resources required to deliver them.

An operational plan draws directly from agency and program strategic plans to describe agency and program missions and goals, program objectives, and program activities. Like a strategic plan, an operational plan addresses four questions:

→ Where are we now?

→ Where do we want to be?

→ How do we get there?

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points:

**Mission profile or scenario:** It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.

**Performance and related parameters:** It points out the critical system parameters to accomplish the mission

**Utilization environments:** It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.

**Operational life cycle:** It defines the system lifetime.

### 4.3 Hardware Requirements

→ Microcontroller (Arduino Mega) → IR sensors → Moisture sensor → DC Motor → DC Motor driver → Conveyor belt → LCD → GPS → Wi-fi module → Gas sensor

### 4.4 Software Requirements

→ Arduino IDE

→ MC Programming Language: Embedded C

#### Summary

This chapter gives details of the functional requirements, non-functional requirements, resource requirements, hardware requirements, software requirements etc. Again the non-functional requirements in turn contain product requirements, organizational requirements, user requirements, basic operational requirements etc.

# Chapter 5

## SYSTEM ANALYSIS

### 5.1 Overview

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

### 5.2 Feasibility Study

All systems are feasible when provided with unlimited resource and infinite time. But unfortunately, this condition does not prevail in practical world. So it is both necessary and prudent to evaluate the feasibility of the system at the earliest possible time. Months or years of effort, thousands of rupees and untold professional embarrassment can be averted if an illconceived system is recognized early in the definition phase. Feasibility & risk analysis are related in many ways. If project risk is great, the feasibility of producing quality software is reduced. In this case there are three primary areas of interest:-

#### 5.2.1 Performance Analysis

For the complete functionality of the project work, the project is run with the help of healthy networking environment. Normally, the OS is windows 7. The main theme of this project is to design a system that correctly identifies plant diseases and generates output specific data. Performance analysis is done to find out whether the

proposed system is time efficient and accurate. It is essential that the process of performance analysis and definition must be conducted in parallel.

### 5.2.2 Technical Analysis

System is only beneficial only if it can be turned into information systems that will meet the organizations technical requirement. Simply stated this test of feasibility asks whether the system will work or not when developed & installed, whether there are any major barriers to implementation. Regarding all these issues in technical analysis there are several points to focus on:-

**Changes to bring in the system:** All changes should be in positive direction, there will be increased level of efficiency and better customer service.

**Required skills:** Platforms & tools used in this project are widely used. So the skilled manpower is readily available in the industry.

**Acceptability:** The structure of the system is kept feasible enough so that there should not be any problem from the users point of view.

### 5.2.3 Economical Analysis

Economical analysis is performed to evaluate the development cost weighed against the ultimate income or benefits derived from the developed system. For running this system, we simply need a computer. All the features in this system run even on the other Operating Systems. So the system is economically feasible enough.

#### Summary

The main aim of this chapter is to find out whether the system is feasible enough or not. For these reasons different kinds of analysis, such as performance analysis, technical analysis, economical analysis etc is performed.

# Chapter 6

## SYSTEM DESIGN

### Overview

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customers requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

### 6.1 System development methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

#### 6.1.1 Model phases

The waterfall model is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of

Requirement initiation, Analysis, Design, Implementation, Testing and maintenance.

**Requirement Analysis:** This phase is concerned about collection of requirement of the system. This process involves generating document and requirement review.

**System Design:** Keeping the requirements in mind the system specifications are translated in to a software representation. In this phase the designer emphasizes on:- algorithm, data structure, software architecture etc.

**Coding:** In this phase programmer starts his coding in order to give a full sketch of product. In other words system specifications are only converted in to machine readable compute code. **Implementation:** The implementation phase involves the actual coding or programming of the software. The output of this phase is typically the library, executables, user manuals and additional software documentation .

**Testing:** In this phase all programs (models) are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation.

**Maintenance:** The maintenance phase is the longest phase in which the software is updated to fulfill the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, enhance the efficiency of the software.

### 6.1.2 Reason for choosing Waterfall Model as development method

1. Clear project objectives.
2. Stable project requirements.
3. Progress of system is measurable.
4. Strict sign-off requirements.
5. Helps you to be perfect.
6. Logic of software development is clearly understood.
7. Production of a formal specification.
8. Better resource allocation.
9. Improves quality. The emphasis on requirements and design before writing a single line of code ensures minimal wastage of time and effort and reduces the risk of schedule slippage.

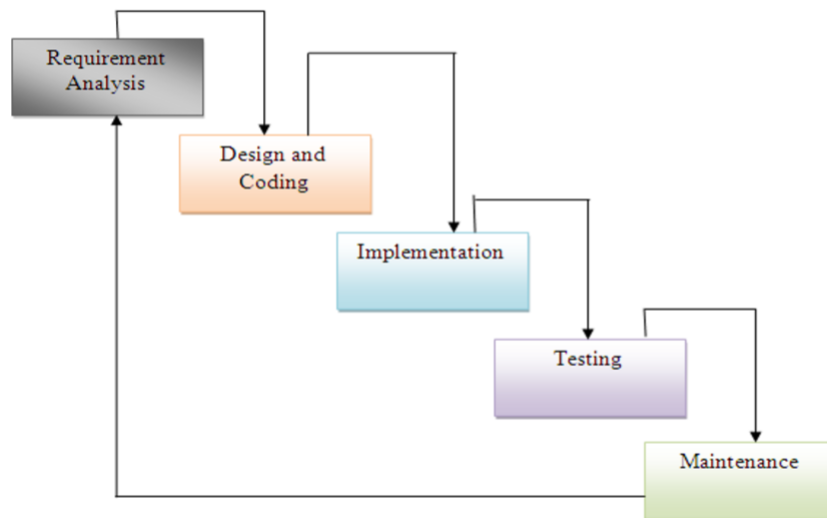


Figure 6.1: Waterfall model

## 6.2 Design Using UML

Designing UML diagram specifies, how the process within the system communicates along with how the objects within the process collaborate using both static as well as dynamic UML diagrams since in this ever-changing world of Object Oriented application development, it has been getting harder and harder to develop and manage high quality applications in reasonable amount of time. As a result of this challenge and the need for a universal object modeling language every one could use, the Unified Modeling Language (UML) is the Information industries version of blue print. It is a method for describing the systems architecture in detail. Easier to build or maintains system, and to ensure that the system will hold up to the requirement changes.

### 6.2.1 Architectural Design

The overall logical structure of the project is divided into processing modules and a conceptual data structure is defined as Architectural Design.

From the above figure 6.2 we can observe that the module is designed to control the accidents caused because of drunken driver and inconvenience of passengers is reduced to some extent.module is designed in such a way that alcohol level is set in

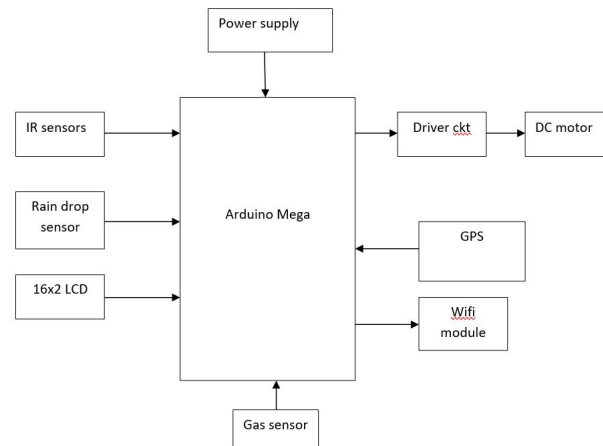


Figure 6.2: System Architecture

alcohol sensor and notification will be sent to authority when the level increases compared to set level of alcohol and moving vehicle is stopped and availability of seats is calculated based on people entering and exiting the bus which is monitored by PIR sensor and switch placed near door and information is sent to passenger throw GSM when it is requested.

### Summary

This chapter mainly concentrates on few fundamental design concepts such as system development methodology, system architecture, class diagram, flowchart, sequence diagram, data flow diagram etc.

# Chapter 7

## IMPLEMENTATION

The implementation phase of the project is where the detailed design is actually transformed into working code. Aim of the phase is to translate the design into a best possible solution in a suitable programming language. This chapter covers the implementation aspects of the project, giving details of the programming language and development environment used. It also gives an overview of the core modules of the project with their step by step flow. The implementation stage requires the following tasks.

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Evaluation of the changeover method.
- Correct decisions regarding selection of the platform.
- Appropriate selection of the language for application development.

### 7.1 Arduino IDE

The NodeMcu is the wifi module required for uploading the APP data to the Smatrix display. NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the dev kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and spiffs.

## 7.2 ESP8266 Arduino Core

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate tool chains to allow Arduino C/C++ to be compiled down to these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file down to the target MCU's machine language. Some creative ESP8266 enthusiasts have developed an Arduino core for the ESP8266 WiFi SoC. This is what is popularly called the "ESP8266 Core for the Arduino IDE" and it has become one of the leading software development platforms for the various ESP8266 based modules and development boards, including NodeMCUs. The nodemcu will provide power to the matrix display. The matrix display will upload its reading in the wireless module, the wireless module will upload the data to the app.

## 7.3 Cloud Computing

Cloud computing is an information technology (IT) paradigm that enables ubiquitous access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a public utility.

Third-party clouds enable organizations to focus on their core businesses instead of expending resources on computer infrastructure and maintenance. Advocates note that cloud computing allows companies to avoid or minimize up-front IT infrastructure costs. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and that it enables IT teams to more rapidly adjust resources to meet fluctuating and unpredictable demand. Cloud providers typically use a "pay-as-you-go" model, which can lead to unexpected operating expenses if administrators are not familiarized with cloud-pricing models.

# Chapter 8

## TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all the system elements have been properly integrated and perform allocated functions. The testing process is actually carried out to make sure that the product exactly does the same thing what is supposed to do. In the testing stage following goals are tried to achieve:-

- To affirm the quality of the project.
- To find and eliminate any residual errors from previous stages.
- To validate the software as a solution to the original problem.
- To provide operational reliability of the system.

### 8.1 Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

#### 8.1.1 White box testing

White box testing (clear box testing, glass box testing, and transparent box testing or structural testing) uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. While white box testing is applicable at the unit, integration and system levels of the software testing process, it is typically applied to the unit. While it normally tests paths within a unit, it can also

test paths between units during integration, and between subsystems during a system level test

Though this method of test design can uncover an overwhelming number of test cases, it might not detect unimplemented parts of the specification or missing requirements, but one can be sure that all paths through the test object are executed. Using white box testing we can derive test cases that:

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structure to assure their validity .

### 8.1.2 Black box testing

Black box testing focuses on the functional requirements of the software. It is also known as functional testing. It is a software testing technique whereby the internal workings of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs.

The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specifications. It enables us to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing is an alternative to white box technique. Rather it is a complementary approach that is likely to uncover a different class of errors in the following categories:-

- Incorrect or missing function.
- Interface errors.
- Performance errors.
- Initialization and termination errors.

#### Advantages

- The test is unbiased as the designer and the tester are independent of each other.
- The tester does not need knowledge of any specific programming languages.
- The test is done from the point of view of the user, not the designer.

## 8.2 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

## 8.3 Integration Testing

Upon completion of unit testing, integration testing begins. Individual modules are combined and tested as a group. Integration testing is black box testing. The purpose of integration testing is to ensure distinct components of the application still work in accordance to user requirements. Integration testing is considered complete, when actual results and expected results are either in line or differences are explainable based on client input. It concentrates on data transfer between modules. Integration testing is a logical extension of unit testing. Two units that have already been tested are combined into a component and the interface between them is tested. Integration testing identifies problems that occur when units are combined .The errors that arise can be attributed to those occurring due to the combination of modules, resulting from errors across interface.

## 8.4 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "interassemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specifications.

The following examples are different types of testing that should be considered during System testing:

- Graphical user interface testing.
- Usability testing .
- Software performance testing.
- Compatibility testing .
- Exception handling .
- Load testing .
- Volume testing.

Although different testing organizations may prescribe different tests as part of System testing, this list serves as a general framework or foundation to begin with.

## 8.5 Quality Assurance

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confidence that the product quality is meeting its goals. This is an umbrella activity that is applied throughout the engineering process. Software quality assurance encompasses:-

- Analysis, design, coding and testing methods and tools.
- Formal technical reviews that are applied during each software engineering.

- Multitier testing strategy.
- Control of software documentation and the change made to it.
- A procedure to ensure compliance with software development standards.
- Measurement and reporting mechanisms.

Quality Assurance (QA) is a way of preventing mistakes or defects in manufactured products and avoiding problems when delivering solutions or services to customers. QA is applied to physical products in pre-production to verify what will be made meets specifications and requirements, and during manufacturing production runs by validating lot samples meet specified quality controls. QA is also applied to software to verify that features and functionality meet business objectives, and that code is relatively bug free prior to shipping or releasing new software products and versions. Quality Assurance refers to administrative and procedural activities implemented in a quality system so that requirements and goals for a product, service or activity will be fulfilled. It is the systematic measurement, comparison with a standard, monitoring of processes and an associated feedback loop that confers error prevention. This can be contrasted with quality control, which is focused on process output.

Two principles included in Quality Assurance are: "Fit for purpose", the product should be suitable for the intended purpose; and "Right first time", mistakes should be eliminated. QA includes management of the quality of raw materials, assemblies, products and components, services related to production, and management, production and inspection processes. Suitable quality is determined by product users, clients or customers, not by society in general. It is not related to cost, and adjectives or descriptors such as "high" and "poor" are not applicable. For example, a low priced product may be viewed as having high quality because it is disposable, where another may be viewed as having poor quality because it is not disposable.

Software quality assurance (SQA) consists of a means of monitoring the software engineering processes and methods used to ensure quality. The methods by which this is accomplished are many and varied, and may include ensuring conformance to one or more standards, such as ISO 9000 or a model such as CMMI. SQA encompasses the entire software development process, which includes processes such as requirements definition, software design, coding, source code control, code reviews, software configuration management, testing, release management, and product integration.

### 8.5.1 Quality Factor

An important objective of quality assurance is to track the software quality and assess the impact of methodological and procedural changes on improved software quality. The factors that affect the quality can be categorized into two broad groups:

1. Factors that can be directly measured.
2. Factors that can be indirectly measured.

These factors focus on three important aspects of a software product

- Its operational characteristics.
- Its ability to undergo changes.
- Its adaptability to a new environment.
- Effectiveness or efficiency in performing its mission.
- Duration of its use by its customer.

In the context of software engineering, software quality refers to two related but distinct notions that exist wherever quality is defined in a business context: Software functional quality reflects how well it complies with or conforms to a given design, based on functional requirements or specifications. That attribute can also be described as the fitness for purpose of a piece of software or how it compares to competitors in the marketplace as a worthwhile product; Software structural quality refers to how it meets non-functional requirements that support the delivery of the functional requirements, such as robustness or maintainability, the degree to which the software was produced correctly.

Structural quality is evaluated through the analysis of the software inner structure, its source code, at the unit level, the technology level and the system level, which is in fact how its architecture adheres to sound principles of software architecture outlined in a paper on the topic by OMG. In contrast, functional quality is typically enforced and measured through software testing. Historically, the structure, classification and terminology of attributes and metrics applicable to software quality management have been derived or extracted from the ISO 9126-3 and the subsequent ISO 25000:2005 quality model, also known as SQuaRE. Based on these models, the Consortium for IT Software Quality (CISQ) has defined five major desirable structural characteristics needed for a piece of software to provide business value: Reliability, Efficiency, Security, Maintainability and (adequate) Size.

Software quality measurement quantifies to what extent a software or system rates along each of these five dimensions. An aggregated measure of software quality can be computed through a qualitative or a quantitative scoring scheme or a mix of both and then a weighting system reflecting the priorities. This view of software quality being positioned on a linear continuum is supplemented by the analysis of "critical programming errors" that under specific circumstances can lead to catastrophic outages or performance degradations that make a given system unsuitable for use regardless of rating based on aggregated measurements.

Such programming errors found at the system level represent up to 90 percent of production issues, whilst at the unit-level, even if far more numerous, programming errors account for less than 10 percent of production issues. As a consequence, code quality without the context of the whole system, as W. Edwards Deming described it, has limited value.

To view, explore, analyze, and communicate software quality measurements, concepts and techniques of information visualization provide visual, interactive means useful, in particular, if several software quality measures have to be related to each other or to components of a software or system.

## 8.6 Test Cases

The following test cases were generated.

### Summary

The chapter discusses the tests that are done on the system to check its functionality. Testing is carried out at three different levels from the module level to the system level checking for errors at each stage. The remarks have also been documented.

Table 8.1: Test case table1

Test id	Input	Expected O/p	Actual O/p	Status
1	Connect wet and dry Sensor to Arduino	Sensor Values are read by microcontroller	Conveyer belt is running	Pass
2	Connect ir sensors to Arduino	Sensor Values are read by microcontroller	Dustbin status	Pass
3	Connect app	Read the Location	Location Sent to android application	Pass
4	Connect	A new form will be displayed	User going to connect through wifi module	Pass
5	Connected with device	Wifi module is connected to the microcontroller	Wifi module connected	Pass
6	Connected with android app	Wifi module will be connected to android app	App and microcontroller are connected	Pass

# Chapter 9

## RESULTS & EXECUTION

The following snapshots and graphs define the results or outputs that we will get after step by step execution of each proposed system.

### 9.1 Snapshots

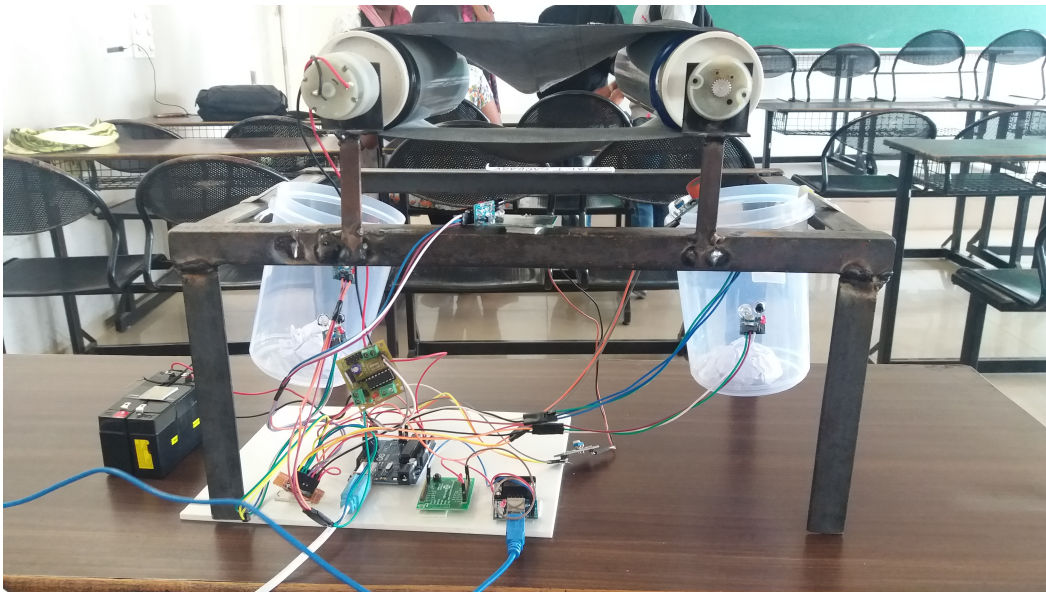


Figure 9.1: Hardware

#### Summary

Assistance to public transport is easy to implement and use daily. It is economic and is reliable as it can be implemented to be used in bulk.

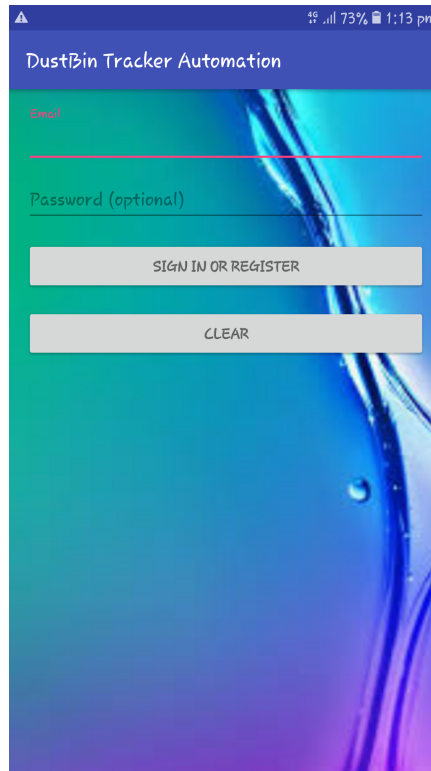


Figure 9.2: Login Page

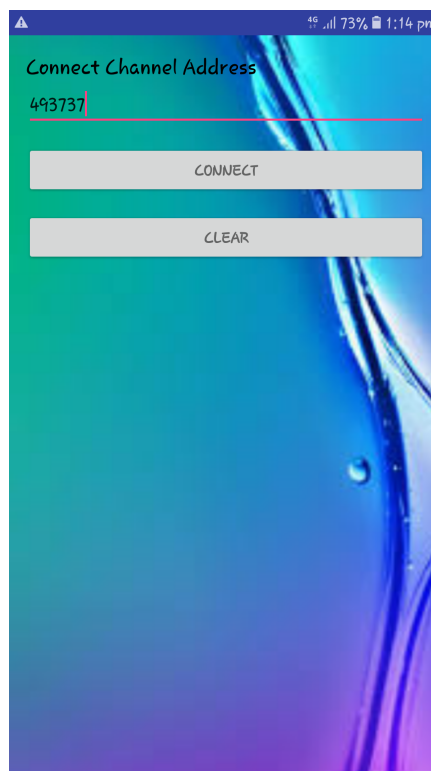


Figure 9.3: Connect Channel Address

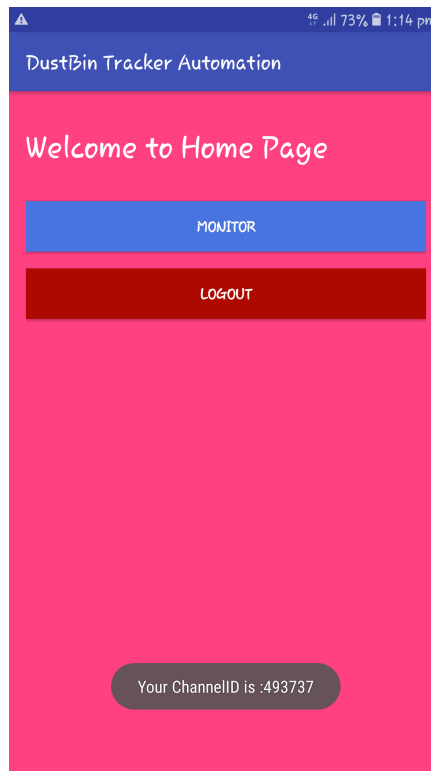


Figure 9.4: Home Page



Figure 9.5: Dust Bin Tracker Automation

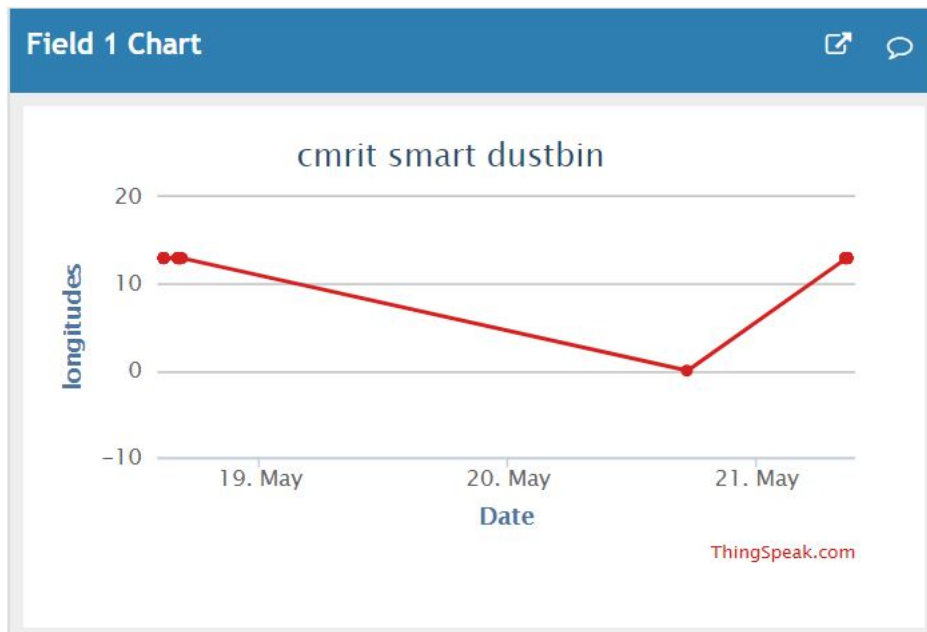


Figure 9.6: Longitudes



Figure 9.7: Lattitudes

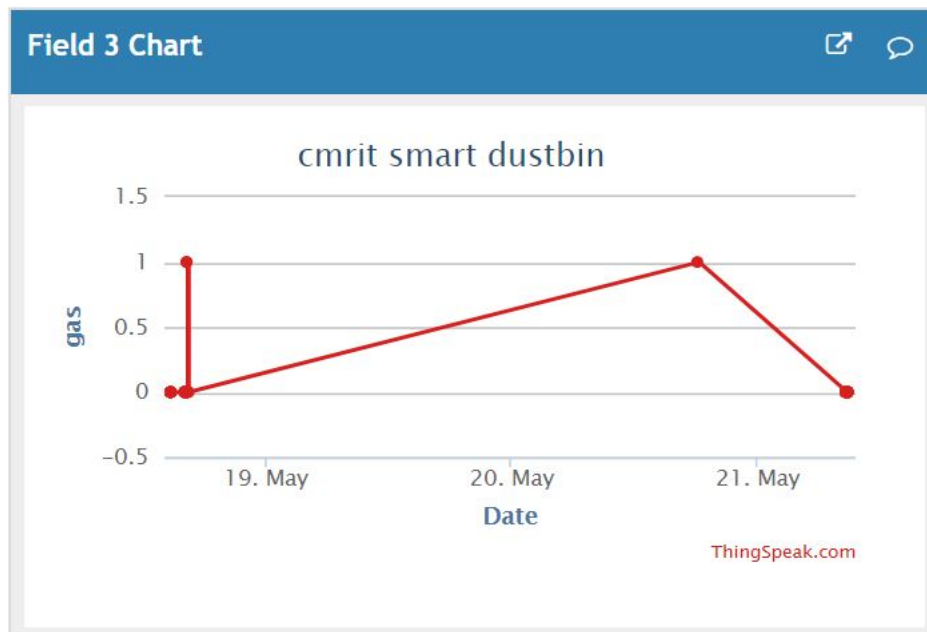


Figure 9.8: Gas

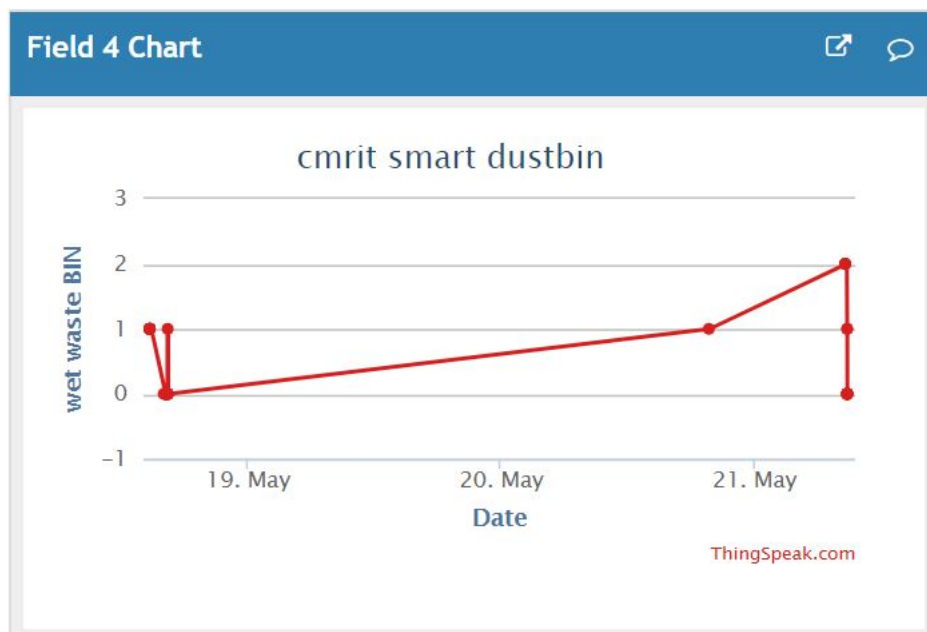


Figure 9.9: Wet Waste Bin

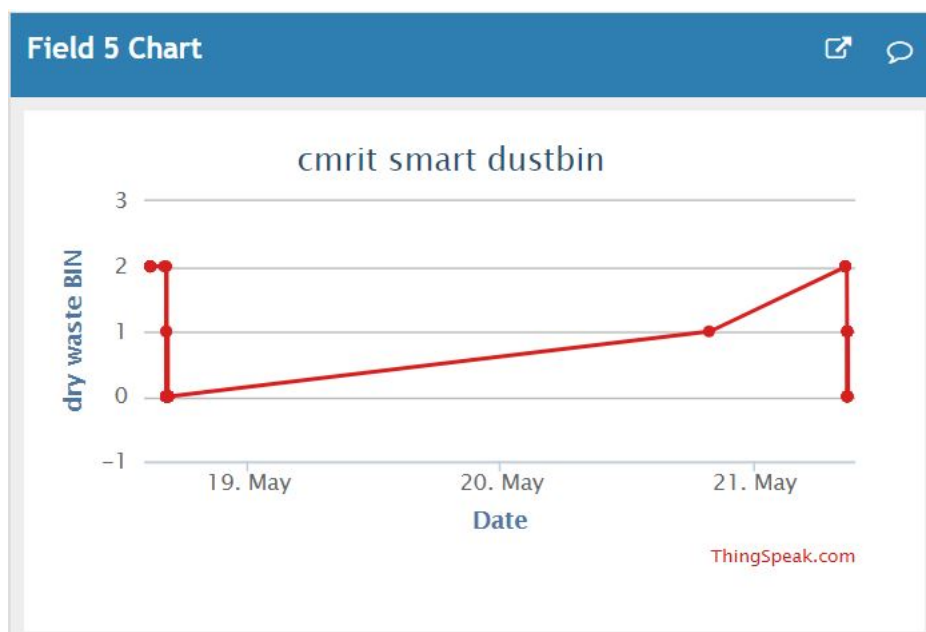


Figure 9.10: Dry Waste Bin

# Chapter 10

## CONCLUSION & FUTURE ENHANCEMENT

### 10.1 Conclusion

Waste Segregation using smart dustbin has been successfully implemented for the segregation of waste into dry and wet waste at root source. One of several environmental problems is bad waste management practices which can result in land and air pollution and can cause respiratory problems and other adverse health effects as contaminants are absorbed from the lungs into other parts of the body. The method presented provides a fruitful way to come out of this problem by making entire system automated. The components used in smart dustbin are economical, environmental friendly and gives accurate results for separating three different types of wastes which are generally produced at places like shopping malls, offices, houses, schools/colleges etc.

Presently there is no device/product available for segregation of waste at root source other than manual separation. Probably the biggest advantage of smart dustbin is the safety it provides. This device carefully separates all three types of waste and not only increases the economic value of waste but also gives a healthy and beautiful environment at lesser cost.

Segregating waste manually is not accurate and many of us don't like to do that. Due to open dumping of solid waste, it emits bad smell due to presence of dead animal waste and biodegradable components. Rodents and dogs are feeding on such dumping place and they may bite people present in those areas. Such dumping sites are spoiling environment of nearby villages surrounding the dumping site.

## 10.2 Future Enhancement

Smart dustbin can segregate only three types of wastes. In future improvement can be done to segregate more number of wastes like plastic, glass, toxic waste, separation of different metals. This can be improved to segregate mix waste dumped at a time using different high accurate sensors and other technologies and segregation of mix waste can also be achieved by using a servo motor to give jerks to conveyor belt so that mix waste gets separated. Size of smart dustbin can be made compact using different technologies and different methodologies. Some additional feature can be added like generating power or biogas generation at root source itself.

# REFERENCES

- [1] Su Lujun. Study on the Mechanism of Service Fairness Influencing to Consumer Loyalty. Journal of Shanxi Finance and Economics University.2010, 07, pp.94-103.
- [2] Zou Jianlan.Customer Loyalty Cultivation Based On Customer Satisfaction J.Journal of Harbin University of Commerceh, 2010, 04, pp.58-60.
- [3] V. Daniel, P. A. Puglia, M. Puglia, RFID-A Guide to Radio Frequency Identification, Technology Research Corporation, 2007.
- [4] A. Flora, "Towards a clean environment: A proposal on sustainable and integrated solid waste management system for university Kebangsaan Malaysia", Report from Alam Flora., 2009.
- [5] L Gogoi, Solid Waste Disposal and its Health Implications in Guwahati City: A Study in Medical Geography, Germany:Lambert Academic Publishing, 2012.
- [6] M. A. Hannan, M. Arebey, H. Basri, "Intelligent Solid Waste Bin monitoring and Management System", Australian Journal of Basic and Applied Sciences, vol. 4, no. 10, pp. 5314-5319, 2010.
- [7] Md. Liakot Ali, Alam Mahbubul, Md. Abu Nayeem Redwanur Rahaman, "RFID based E-monitoring System for Municipal Solid Waste Management", International Conference on Electrical and Computer Engineering, pp. 474-477, 2012.
- [8] H. Rahman, A. Al-Muyeed, "Solid and Hazardous Waste Management", ITN-BUET Center for Water Supply and Waste Management, 2010.
- [9] Twinkle Sinha, K. Mugesh Kumar, P. Saisharan, "SMART DUSTBIN", International Journal of Industrial Electronics and Electrical Engineering, vol. 3, no. 5, 2015.