A PROJECT REPORT

ON

"WATER QUALITY MONITORING & NOTIFICATION SYSTEM USING IOT"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

MOMIN JUNAID DAWOOD ZAHIDA	**14CO35**
SHAIKH RAHEEN NAEEMUDDIN SANJIDA	**14DCO65**
ANSARI MOHD ARKAM SOHEL ANJUM RAISA	**14CO15**

UNDER THE GUIDANCE OF
PROF. MUBASHIR KHAN

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2018-2019

AFFILIATED TO

UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

"WATER QUALITY MONITORING & NOTIFICATION SYSTEM USING IOT"

Submitted to
UNIVERSITY OF MUMBAI
Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

MOMIN JUNAID DAWOOD ZAHIDA	**14CO35**
SHAIKH RAHEEN NAEEMUDDIN SANJIDA	**14DCO65**
ANSARI MOHD ARKAM SOHEL ANJUM RAISA	**14CO15**

UNDER THE GUIDANCE OF
PROF. MUBASHIR KHAN

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2018-2019
AFFILIATED TO

UNIVERSITY OF MUMBAI

Anjuman-i-Islam's Kalsekar Technical Campus

Department of Computer Engineering
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

CERTIFICATE

This is certify that the project entitled

"WATER QUALITY MONITORING & NOTIFICATION SYSTEM USING IOT"

submitted by

MOMIN JUNAID DAWOOD ZAHIDA	**14CO35**
SHAIKH RAHEEN NAEEMUDDIN SANJIDA	**14DCO65**
ANSARI MOHD ARKAM SOHEL ANJUM RAISA	**14CO15**

This is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at ***Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai*** under the University of MUMBAI. This work is done during year 2018-2019, under our guidance.

Date: / /

Prof. MUBASHIR KHAN
Project Supervisor

Prof. KALPANA BODKE
Project Coordinator

Prof.TABREZ KHAN
HOD, Computer Department

DR. ABDUL RAZAK HONNUTAGI
Director

External Examiner

Acknowledgements

We would like to take the opportunity to express my sincere thanks to my guide **Prof. Mubashir Khan**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout my project research work. Without his kind guidance & support this was not possible.

We am grateful to him/her for his timely feedback which helped me track and schedule the process effectively. His/her time, ideas and encouragement that he gave is help me to complete my project efficiently.

We would like to express deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. Tabrez Khan**, Head of Department of Computer Engineering and **Prof. Kalpana Bodke**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

Momin Junaid Dawood Zahida
Shaikh Raheen Naeemuddin Sanjida
Ansari Mohd Arkam Sohel Anjum Raisa

Project I Approval for Bachelor of Engineering

This project entitled ***Ẅater Quality Monitoring & Notification System Using IoT¨*** by ***Momin Junaid Dawood Zahida, Shaikh Raheen Naeemuddin Sanjida, Ansari Mohd Arkam Sohel Anjum Raisa*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering.***

Examiners

1. .....

2. .....

Supervisors

1. .....

2. .....

Chairman

.....

Declaration

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Momin Junaid Dawood Zahida
14CO35

Shaikh Raheen Naeemuddin Sanjida
14DCO65

Ansari Mohd Arkam Sohel Anjum Raisa
14CO15

ABSTRACT

In overall monitoring of water, standard takes a lot of time and it appears to be a little complex as it undergoes several laboratory testing, which results in time-consuming and for this a real time monitoring of water goodness by using IoT has been proposed. IoT together with sensors is quite effective. Some other technologies had certain drawbacks which it couldn't reveal it. So to remove this short coming and providing an efcient and socio-economic solution has been the main focus of this project. Here we are using sensors which will monitor water on the bases of its level, conductivity, ph value, ow control, and the most important to detect if any presence of micro-organisms is found. The accessed data are controlled by the use of the micro-controller. By using IoT, the information can be collected and the water pollution can be inquired to the public or the person in charge of it. This can make atmosphere to adapt the good quality if water.

Keywords: IoT, Sensors, socio-economic, ph-value, micro-organisms, bacteriophage.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is a trendy work to check water quality regularly using agile technologies and in fact, we common people cannot detect that. Mostly it is found that water tanks are controlled manually to check the ow of water. If the person is found absent, the water keeps on owing despite the loss of water. So this problem can be controlled by using solenoid valve which can be controlled by the user from anywhere on getting the indication of water level through the ultra-sonic sensor.

This wireless sensor technologies are been developed for data acquisition, building a better environment, to enhance the livelihood, to improve one's better life. In other projects, they have used sensors which controls the ow of water or either detects the temperature of water or maybe to check the level of chlorine present in it. In this project, the user can not only detect the ow of the water but can also detect the quality of water based on its ph value, present of any micro-organisms. On getting such indication the user can decide to stop the ow of water. Because the water quality that is present in the tank tend to be contaminated with either of these contaminations.

Ensuring the safety of water is a challenge due to the excessive sources of pollutants, most of which are man-made. Water quality is affected by both point nonpoint sources of pollutants, which include entry of soil through leak in pipeline, over stored of water which leads to rise in micro-organisms, over rate of salinity in water, etc. Poor quality of water spreads disease; causes death hampers scion economic progress.

Water quality monitoring is defined as the collection of information at set locations at regular interval of time in order to provide data which may be used to define current condition of water. Main objective of water quality monitoring include measurement of critical water quality parameters such as microbial, physical chemical properties provide early warning identification of hazards

1.1 Purpose

Today every human man kind and the environment in the world is facing the main problem with the concept of Smart Adaptation. To keep up with the requirements this new Smart Adaptation, conservation of environment and improvement in human health require new technique to improve the quality of the society. In this aspect, science and technology provide hands to this field. Scientific research provided data to improve the quality whereas technology approach provides control on the quality of the society. In this system, we try to improve the quality of water for the society.

1.2 Project Scope

Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here. An SRS that specifies the next release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision.

1.3 Project Goals and Objectives

The proposed project aims at monitoring the various analogue parameters such as pH value, level of water, level of clarity,etc. The system should control the affection of impurity of water in human mankind.
The user will also be able to control the flow of water if desired. The processed data will be sent over wireless network to the owner to remotely access from anywhere in the world. To tackle the problems that effect the health of human life through water contamination, we have designed a system which will determine present environmental conditions.
According to the conditions of purity of water the threshold values will be set, if any of the parameter's value goes beyond or below the set threshold value, then the proposed design will sense the changing in parameter's and it will be monitored continuously.
Entire data will be transmitted to the end user, who will accordingly will take the controlling decision and send the action to the system. With the help of the the actuator the control of the system can take place.

1.3.1 Goals

Our goal is to control the effect ion of contaminated water on human life and the wastage of water supply in the environment. The water quality monitoring system

will do the desired problem tackling and compressing of the health issues related to water diseases. As indicated by pre-requisite of the product the edge will be set, if any of the natural condition goes below or above the limit value, then the proposed venture will detect the change in parameters.

1.3.2 Objectives

The objective of the proposed system is that with help of IoT concept we can detect the parameters of contamination in the water such as pH value or maybe conductivity, we can implement an automated smart system. The processed data will be sent over the wireless network to the owner or the end user to remotely access from anywhere.
To tackle problems that contaminated the water in environment we have proposed a design which will determine present purity level in water. Entire data will be transmitted o the end user, according to that the user can take the action of flow control of water.

1.4 Organization of Report

In chapter 1: We have considered Project overview under which we have explained various important terminologies like Introduction of the project, Motivation for the project Problem definition, About the current system, Advantages over the current system, Goals, Objectives, Scope Application.

In chapter 2: We have discussed about various paper that we have referred for our project. We have mentioned the description, pros and cons and how to overcome the problem under every paper. Total of three paper have been referred.

In chapter 3: We have discussed about the requirement analysis under it we have consider about the requirement for the system, the requirement supporting for the OS of the software nd hardware.

In chapter 4: We can see the system design and architecture various diagram can be seen in this chapter which represents the software, diagrams including our system architecture, use case diagram, activity diagram, DFD diagram, data flow diagram, sequence diagram, etc.

In chapter 5 : We have seen the methodology, here we have explain the project in detail by dividing into modules. Various modules or priority based cab search are explained with the help of few diagram.

In chapter 6: We have discussed about the implementation details, the assumptions and dependencies, this part contains details of the implementation of methodology that we have discuss earlier.

In chapter 7: We have show the test cases and the result along with analytic discussion, this part consist of the result of the output of the project.

In chapter 8: We have concluded the whole project and future scope along with the limitation followed up by reference and chapter 9 consist of Appendix.

Chapter 2

Literature Survey

2.1 Water Tank Monitoring and Visualizing System Using Smart Phones

In 2013, Haesung.Tak, Daegeon.Kwon and Hwan.Gue.Cho; "Water Tank Monitoring and Visualizing System Using Smart Phones". This paper proposed a water tank monitoring and visualization system using smart-phones known as "Tank boy." Monitoring sensor networks that consist of many valves, pumps and tanks is an important task for managing large ships. Existing water tank monitoring systems are only provided for PC environment. In this paper they propose a new method to monitor water tanks on a smart phone. Before presenting their system, they analyze the water tank system and define the discussed problem formally. Based on this analysis, they develop their monitoring system and elaborate on out the implementation. They also show that their system visualizes sensor data using a simple and intuitive user interface.

2.1.1 Advantages

a. This system is a real-time monitoring system using a database to establish a marine communication system.

b. Its implementation on a smart-phone environment is novel.

c. Monitored the real-time data processed for water tanks in ships.

2.1.2 Disadvantages

a. The first is the delay in accessing the Web page.

b. The second is due to the monitoring of the parsing information.

c. The control of the water in the tank was made by accessing valve manually.

2.1.3 How to overcome the problems mentioned in Paper

a. The delay in accessing the data can be overcome by either implementing a web based application or and android application.

b. The monitoring of water parameters can be obtain on real time basis.

c. By implementing the system for major issues in the environment.

2.2 Smart Water Management Using IoT

In 2016, Sayali. Wadekar, Vinayak. Vakare, Ramratan Prajapati, Shivam Yadav, Vijaypal Yadav; "Smart Water Management Using IoT".This paper presents an IOT device which help to manage and plan the usage of water. This system can be easily installed in residential societies. Sensors placed in the tank which continuously informs the water level at the current time. This information will be updated on the cloud and using an android application, user can visualize the water level on a Smartphone anywhere that is connected to Internet. According to the level of water in the tank the motor functioning will be automatically controlled, at low level of water motor will automatically turn on and when tank is about to fill up it will cut off.

2.2.1 Advantages

a. It is a robust system and small in size.

b. Motor can be controlled automatically; full smart automation is achieved.

c. Shows the user detail of previous values too.

2.2.2 Disadvantages

a. This device was implemented at personal level.

b. They could just work on monitoring the level of water present in tank.

c. This system was implemented in android application only.

2.2.3 How to overcome the problems mentioned in Paper

a. System could have been implementd for major level of use.

b. With monitoring of level of water, system could have covered other issues too.

c. Application could have been made for multi platform.

2.3 IoT Based Smart Water Tank With Android Application

In 2017, Priyen. P. Shah, Anjali. A. Patil and Subodh. S. Ingleshwar; "IoT Based Smart Water Tank With Android Application" Implemented an efficient automated water level monitoring and controlling system.In this paper, they introduce the project of water level monitoring as well as controlling with IoT and android application. Wastage of water in the current scenario, merely due to overflowing tanks is not affordable. Conventional water tanks can neither monitor nor control the water level in tank, leading to large amount of wastage. Some other technologies had certain drawbacks in some or the other way. The need of removal of these short-comings and providing an efficient and economical solution has been the main focus of this project.

2.3.1 Advantages

a. This project doesn't require special different tank for it, existing water tanks can be used.

b. A portable system which can solve our water wastage problem.

c. Making decisions with percentage proves to be easier to implement the logic in programming.

2.3.2 Disadvantages

a. They could just work on monitoring the level of water present in tank.

b. They could just work on the process of identifying the level of water and to control the flow of water.

2.3.3 How to overcome the problems mentioned in Paper

a. The system can be implemented at commercial or at medical level.

b. Flow control could have been made by some valve rather than using motor.

2.4 Smart Water Tank Management System for Residential Colonies Using Atmega128A Micro-controller

In June 2014, Yogita Patil, Ramandeep Singh,In this paper they represented the idea of smart water tank management system operated with Atmega 128A microcontroller, which is the prime component of this project. A prototype has been made for this project. So, in this way manual intervention is not required for continuous water supply.This system can also be used for any other fluids in chemical industries

or factories. The main aim of this project is to provide optimal water distribution and moreover reduce manpower involved in it.

2.4.1 Advantages

a. This project was made with the moto of implementing at other level of factors.

b. The distribution of water and manpower was tackle.

2.4.2 Disadvantages

a. This system could only monitor the level of water in the tank.

b. The system was implemented for android user.

c. Other aspects of parameters were missing.

2.4.3 How to overcome the problems mentioned in Paper

a. System can be made for multi-platform user.

b. More quality parameters can be involved.

2.5 Water Quality Monitoring with Internet of Things (IoT)

In 2017, Kamarul Hafiz Kamaludin and Widad Ismail,worked on the project called " Water quality monitoring with Internet of things (IoT). The paper suggests an Internet of Things (IoT) based system implementation by embedding the Radio Frequency Identification (RFID) system, Wireless Sensor Network (WSN) platform and Internet Protocol (IP) based communication into a single platform for water quality monitoring (WQM) purpose. The suggested radio frequency for the proposed WSN communication to be deployed in vegetation area is 920MHz. The measured water parameter in this proposed system is pH level by using an analog pH sensor. The ambient temperature is captured during pH measurement by using an analog temperature sensor. All the WSN nodes are deployed in a real environment at the lake in the campus area of Universiti Sains Malaysia (USM) for performance evaluation. Instead of using 2.4GHz ZigBee protocol, the 920MHz Digi Mesh protocol is proposed to be implemented for water quality monitoring in vegetation area due to its ability to surpass the signal attenuation. This novel proposed system prototype was evaluated in a real environment to ensure that the main functionality on pH measuring process is following the design requirements. Several experimental analysis were conducted including the energy analysis and communication read range analysis to study the overall performance of the proposed system.

2.5.1 Advantages

a. This system was implemented by introducing other different parameters too.

b. This proposed system is reliable for a real environment deployment.

2.5.2 Disadvantages

a. For making the system run over 900m, another wifi setup was required.

b. The system was made with purpose for android user.

c. The flow control of the water was yet not considered.

2.5.3 How to overcome the problems mentioned in Paper

a. We can overcome the problem of wifi setup by implementing wifi module.

b. The end user application must be made by considering the multi-platform user.

c. The flow of water can be monitored and control by implementing autonomous valve.

2.6 Technical Review

Our System is based on Internet of Things, Which includes

* Sensors.

* Micro-controller called as Arduino UNO.

* Cloud server to fetch and dumb data.

For User End or for the front end we have worked on multi-platform application. Which includes of Django (python).

For Backend or for storing data on cloud we have used Thingspeak, Which communicate very well with arduino.

2.6.1 Advantages of Technology

a. Machine to Machine communication.

b. Extensive use of autonomous and control.

c. Monitoring and access information.

d. Minimizes labour cost and manual work.

e. Customer able to define the quality of water.

2.6.2 Reasons to use this Technology

a. Improve decision making.

b. Improve and optimize operations.

c. to have information and access from anywhere.

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Junaid Momin	Arduino programming, Server connection, Documentation
2	Raheen Shaikh	Arduino programming, End user application
3	Arkam Ansari	Database, Arduino programming

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Junaid Momin	Team Leader	connectivity, data fetching
2	Raheen Shaikh	End user developer	Frontend UI
3	Arkam Ansari	End user developer	Backend

3.3 Assumptions and Constraints

Assumption is that the data that is going to be uploaded on server and the end user application will receive that data from the server. Similarly for controlling the system with it.

3.4 Project Management Approach

Spiral model is a combination of a waterfall model and iterative model. Each phase in spiral model begins with a design goal and ends with the client reviewing the progress. We have use this model for version control.

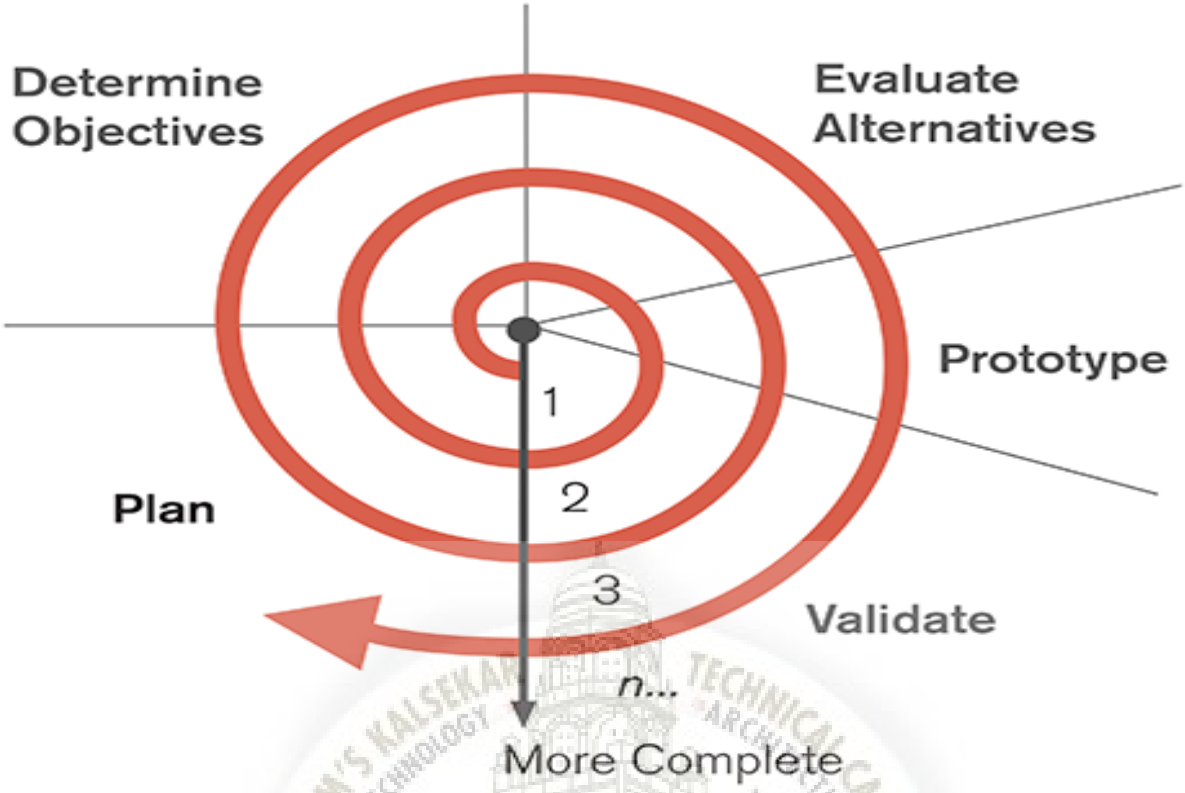

Figure 3.1: Spiral Model

Determine Objectives of the system is to detect the quality of water based on real time monitoring and to provide the result to the user based on its purity level.

Evaluation of alternatives can be found in case of any failure in one of the working of the sensor in our system. This can result in difficulties in completing the process of the system.

Prototype of our system is mainly an Arduino UNO which connects and communicate the sensors to the internet and the user.

Validation of the system can depend on the working conditions of the system based on the proper connection and accurate current supply as well as the internet.

Planning of the system result in smooth functioning of the system on either ways right from sensors to cloud and the user.

3.5 Ground Rules for the Project

1. We treat each other in the team with respect and dignity.
2. We intend to develop better relationships to enhance our trust and have a better communication.

3. We value constructive feedback. We will avoid being defensive and give feedback in a constructive manner.
4. As a team members, we will pitch in to help wherever necessary to help solve the problems.
5. Additional meetings can be scheduled to discuss critical issues or tabled we should create and adopt written notes for help. No responsibilities be assigned unless the person who is being assigned the job accepts it.

3.6 Project Budget

Table 3.3: Project Budget

SR. No	Project Budget	Quantity	Price
1	Arduino UNO	1	350
2	Arduino ESP8266 wifi module	1	550
3	Ultra sonic sensor	1	100
4	Conductivity sensor	1	1300
5	PH tester sensor	1	1800
6	Solenoid valve	1	360
7	Wires Male to Male	15	80
8	Wires Male to Female	10	60
9	Wires Female to Female	10	60
10	Bread Board	1	90
11	USB cable	1	80
		Total	4,740

3.7 Project Timeline

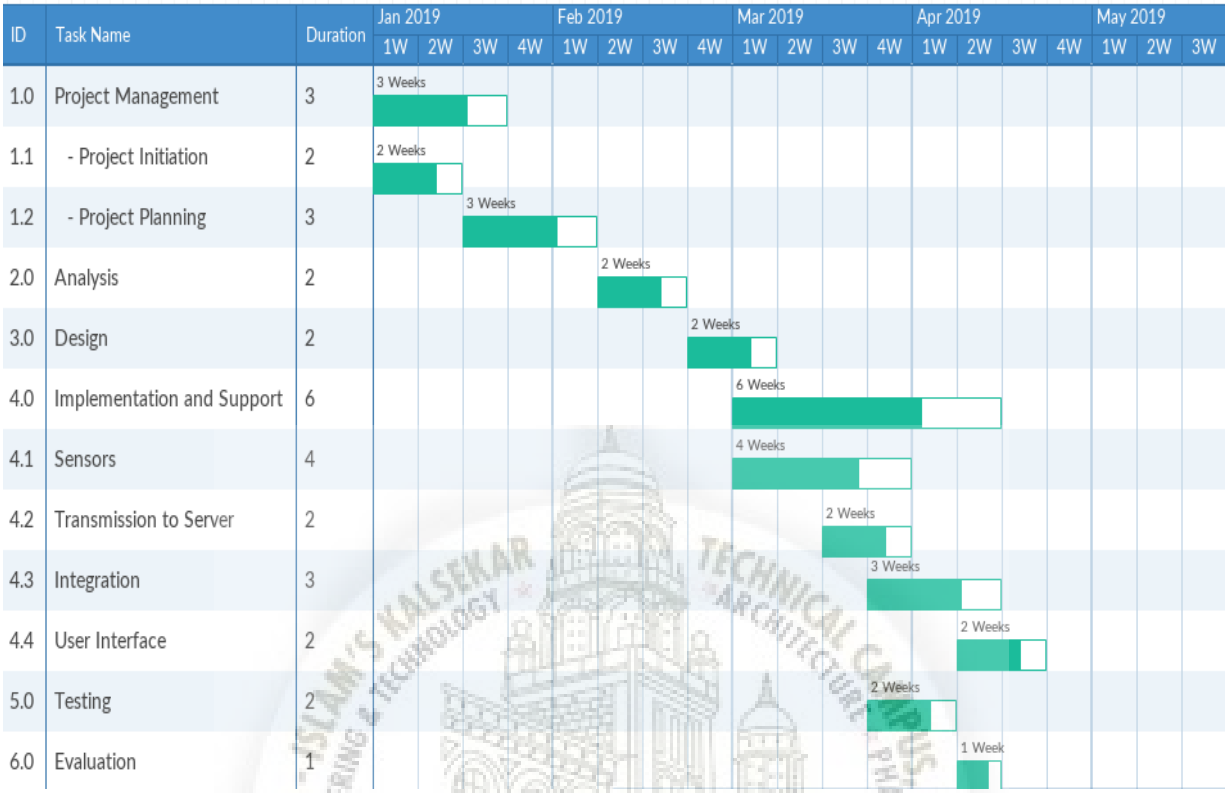

Figure 3.2: Gantt chart of the project

Chapter 4

Software Requirements Specification

4.1 Overall Description

This software requirement specification is the requirement work product that formally shows the user the quality of water based on different parameters. The objective of this system is to minimize the loss or wastage of water and to preserve the health of human mankind.

4.1.1 Product Perspective

The primary point of these project is to enable ranchers to even in inverse ecological conditions like in contamination or shortage of supply of water. A water management is an important factor for our daily consumption in every aspect of field. It is used in day to day life for various purpose.
Automatic monitoring is done, through which water quality can be monitored which can result in detection of any impurities of water. The main aim of this project is to help in reduction of health issues of the society that rises because of water contamination. Here controlling will be done by arduino and the monitoring part is handled by IOT. We proposed water quality monitoring system that consist of sensors which detect the purity of the water.
The sensors we proposed in the project are ultra-sonic sensor, conductivity sensor, ph tester sensor and solenoid valve. An Arduino UNO is used to operate on these sensors and to detect availability of parameters.
Internet of things makes monitoring of system accessible everywhere having net access on any smart phones. The product is supposed to be an open source, it is hardware and software based implementation.

4.1.2 Product Features

Internet of things makes observing of framework available wherever having net access on any smart device. The item should be an open source, it is equipment and programming based execution. The accompanying fundamental component gave by

an application are:

4.1.3 Operating Environment

This is an android system hence, will require the operating environment for a client and server GUI. The product has an user based framework and subsequently will require the working condition for a customer and server GUI. This will work in the accompanying working condition.

- Our project mainly rely on hardware.
- Our software mainly dependent on the internet, without wifi use the application can't work.

4.1.4 Design and Implementation Constraints

1. Speed accuracy as compared to wired network
2. Use of necessary internet.
3. Gets distracted by various elements

4.2 System Features

We have arduino uno to control the environment of water. We have used ESP8266 to send and receive the data over internet. We have used application class for user interface and similarly server class to send and receive data to our server. We have utilized ESP8266 to send and receive information over the web based application.

4.2.1 System Feature

1. Low cost
2. Web based application to be used on multi-platform.
3. Web based application to get information and control the flow of water.
4. Efficient framework
5. Extensive amountof automation and control.
6. Monitoring and access of information
7. Minimization of labour cost in controlling the flow of water and to test parameters in laboratories.
8. Digitization in conservation of human life and water.

Stimulus/Response Sequences

This is the sequence diagram of the system in which modules are sensor, gateway, middleware server. The water quality monitoring system using IoT the sensor data will be transmitted through a set of different devices to the cloud where it get processed, if a measurement crossed the threshold value the server sends a command to a specified parameter in order to perform the suitable action. The diagram below describes the system

Functional Requirements

1. Sensor should sense the parameters of the water.
2. The correct details of the sensor should be fetched by server.
3. The server should respond quickly.
4. Immediate output on the screen.

4.3 External Interface Requirements

4.3.1 User Interfaces

Daily consuming water people are the main audience of our system. This interface should be very user friendly and easily understandable to the user. All the transactions and requirements are merged up in the interface. The entire system focuses on this user and tries to provide the best utility of this application.

4.3.2 Hardware Interfaces

Sensors:

1. Ultra-sonic sensor : It is a sensor that generates the level of water in the tank or any storage. It can be interface with arduino, raspberry pi, etc and get instantaneous result. Ultra-sonic sensor is low cost level detector which provides high realibility and long term stability.
2. Conductivity sensor: This is an easy to use digital ourity checker of the water. Just insert the sensor in water and it can measure the haziness level present in water, which result in clarity of water present. It gives digital output.
3. pH tester sensor : It is a tester that indicates the ph value of water. The name itself indicates the purity of water based on ph is measured. It calculates the ph value based on condition, if the level is below 7 then the water is acidic or if the level is more than 7 then the water is basic else the water is neutral.

4. Solenoid valve: This sensor act as an actuator in the system. When the parameters calculated by each sensor results in impurity of the water, the user can control the flow of water by controlling the switching of solenoid valve. Hence, solenoid valve act as flow controller of the water in the system.

Wifi module ESP8266: It connects the sensor and the arduino with the internet to user application. It updates the information on both sides that is the server and the database. We are using ESP8266 to connect the arduino with the server.

Arduino: It is a controller which is used for particular small application such as water quality monitoring system. In this project the sensors are sensing the purity of water such as ph value. These sensed value are passed to arduino which decides the activation of actuators. The arduino UNO is a micro controller board based on ATMega328. It has 14 digital i/o pins in which 6 are analog inputs.

4.3.3 Software Interfaces

User Interface : In this system we have worked on focusing the user of different OS. Hence, the user interface is based on web based application where user can access this system by using API of the application. For making this application as a web based application we have used Xamarine so as to deliver and focus on the user using android, ios or windows devices.
Arduino programming can b divided into three parts: structure, values(variables and constants) and functions. A versatile application is a PC programming intended to keep running on a cell phones.We are using web based application so as to get to the user remotely. Versatile application improvement requirement.
FUNCTIONS: For controlling the arduino board and performing computations.
VARIABLES: Arduino data types and constants are used to hold the value in variable.
STRUCTURE: The elements of arduino (C++) code.

4.3.4 Communications Interfaces

L2CAP: Logical Link Control Adaptation Protocol. Wi-fi security: WPA-PSK, WEP-64, open. Routing ADVOC: Ad hoc on-demand. Distance vector DSR: Dynamic source routing. BATMAN: better approach to mobile application network. Embedded and electronics, s/w, sensor, actuator sand network connectivity that enables devices to communicate to other or exchange of data.

4.4 Non-functional Requirements

4.4.1 Performance Requirements

1. Data retrieving speed : The speed of retrieving data from server should be faster as possible, more delay may cause the wrong solution.
2. Data manipulation speed: Manipulating the fetched data and provided database must be faster, it may lead to fault result. notification: Notification time should be normal, that user can get the understanding of the purity of water. timing: After completing all the processes, user should know that what parameter of the water has result in reduction of purity level in water.

4.4.2 Safety Requirements

1. Use-Case: Use case is used to prevent the circuit of the system from external circumstances, the circuit includes arduino, wifi module, sensors, wires, etc. They all should be protected from external affairs. Maintenance of the circuit is done by use-case.
2. IOT: Iot includes all the safety parameters regarding to the internet such as http, https, and other security parameters such as communication protocols.
3. Database: It must be secure, because it contains all the information related to the purity of water for the user. It should be prevented from external users, otherwise a small change in database may lead to dangerous failure of the system.

4.4.3 Security Requirements

1. Availability : The system should be available on any part of the earth with internet connection as it supports iot.
2. Correctness: the data send and receive should be always accurate and correct.
3. Flexibility: The system should work with any user application devices.
4. Reliability : The system should always be able to work according to the need of the user.
5. Maintability : The system doesn't need much maintenance ths it can be done easily.

Chapter 5

System Design

5.1 System Requirements Definition

Arduino is open source Hardware Software based development platform. The arduino micro-controller is easy to use yet powerful single board computer. It's combination of micro-controller based Arduino boards, Arduino programming Language and the Arduino software for development and compilation. All the sensors actuator will be connected to the arduino. A wifi module will be used to connect the Arduino to the Internet. Cloud server will provide an interface between the user and the database. Examples for cloud server are AWS, ThinkSpeak,etc. There are Arduino Shields, which are fix on board. We can use them externally as well through ESP8266, ESP13 Wifi shield, GSM module and zigbee. To reach long distance over internet, Zigbee devices use mesh network of intermidiate devices. Zigbee has a defined rate of 250 kbit/s,best suited for intermediate data trasmission from a sensor or input device. Where as GSM modules are used to send a message or miscall over a network. The ESP8266 or ESP13 shield is low cost wifi chip woth on load full TCP/IP stack. It's a transceiver that can be added to any existing micro-controller based setups via UART (serial link) to enable the system to communicate over the Internet via Wifi. Commands will be sent from an android application through the internet and to Wifi network to ESP8266 with Arduino Mega.

5.1.1 Functional requirements

Every one of the sensors actuator will be associated with the Arduino. A wifi module will be utilised to associate the Arduino to the Internet. Cloud server will give an interface between the client and the database. Cases for cloud server are AWS, ThinkSpeak and so forth. There are Arduino shields, which are settle on board. We can utilise them remotely too through ESP8266 or ESP13 shield, GSM module and Zigbee.

Use-case Diagram

By using the Use-Case diagram, the user can monitor the quality of water (view the parameters, current values, history, actuator state purity condition of water), perform a manual action (turn on off the actuator) and view the notifications. The following diagram describes the functional requirements of water quality monitoring system using IOT.

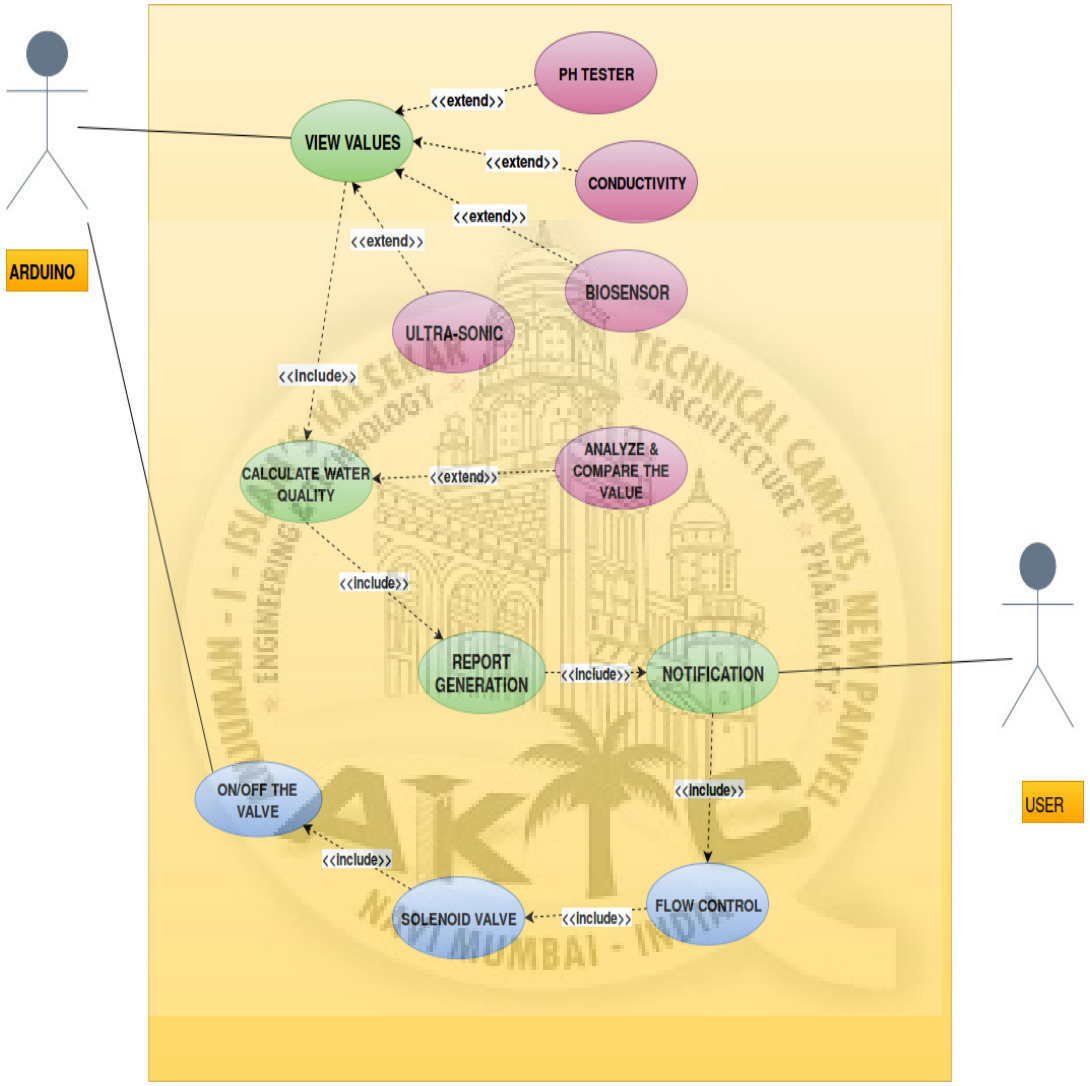

Figure 5.1: Use Case diagram for Water Quality Monitoring and Notification System

Data-flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short labels or texts to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to-in depth, multilevel DFD's that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one.

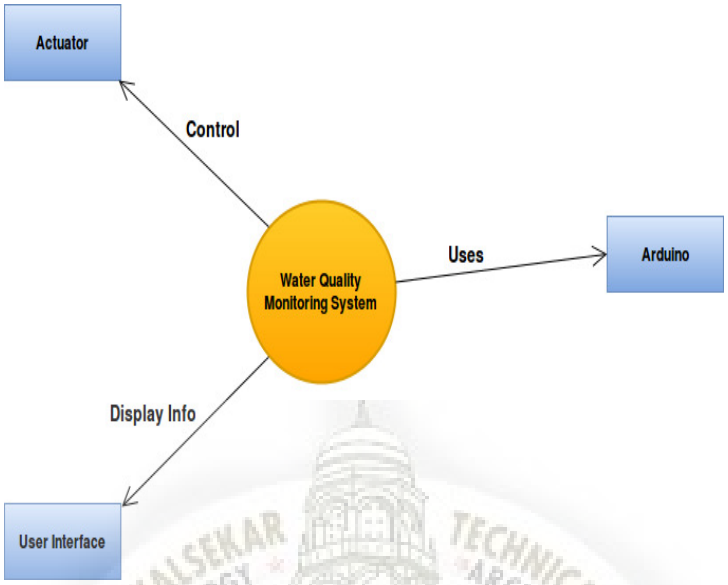

Figure 5.2: Data-Flow Diagram-0

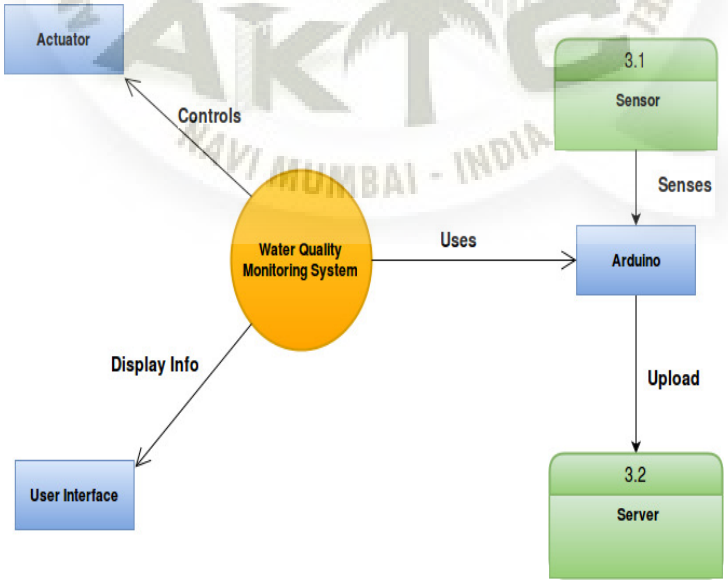

Figure 5.3: Data-Flow Diagram-1

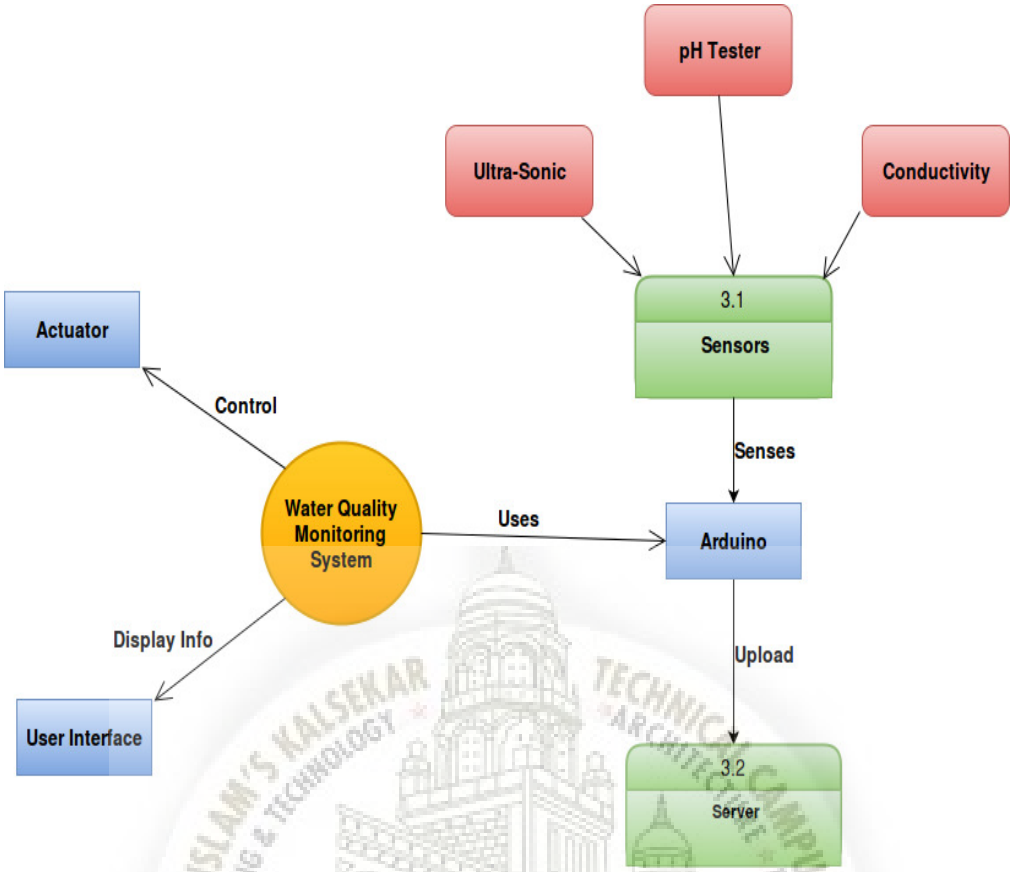

Figure 5.4: Data-Flow Diagram-2

5.1.2 System requirements (non-functional requirements)

These are non-functional system properties such as availability, performance and safety. They define functions of a system, services and operational constratints in detail.

5.2 System Architecture Design

In the proposed system, it consist of an micro-controller that is Arduino which is connected with different kinds of sensors such as pH tester, conductivity, bio-sensor, etc, which are placed or induced inside the water to monitor its quality. This system uses this commercial hardware to allow the acquisition of data.

After the data acquisition, with a defined sampling period this data is calculated and then analyzed and compared with the stored data. When the comparison is completed it generates the reports based on the historical acquired data. This report is then forwarded to the user as a notification.

Based on the notification and the report of water quality obtained, the user can decide to control the flow of the water, if the value obtained is beyond the threshold

value. This flow control of water is made with the help of solenoid valve which is a sensor which controls the flow of water. All this features can be accessed at any time.

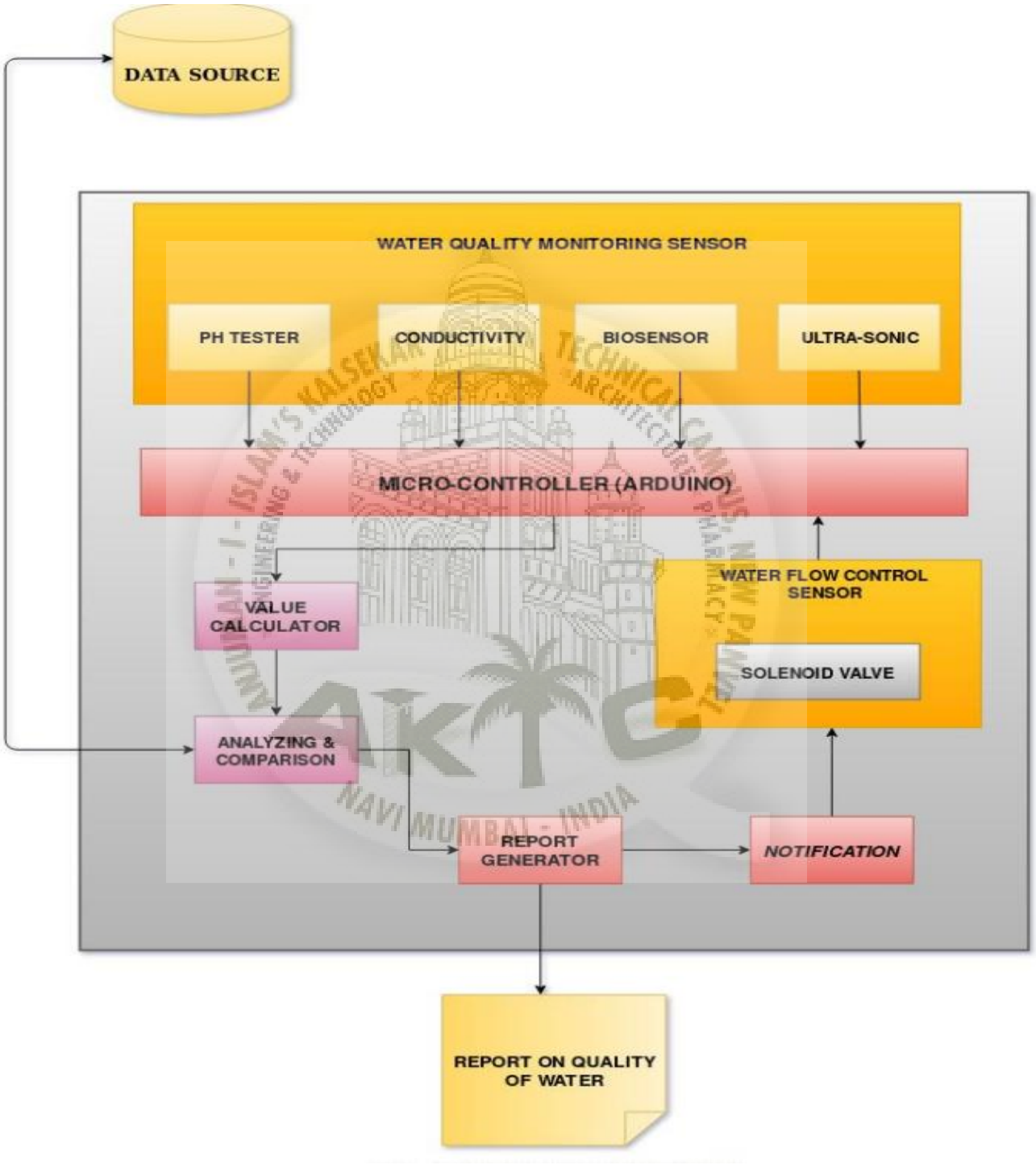

Figure 5.5: Architecture of Water quality Monitoring and Notification System using IoT

5.3 Sub-system Development

Our system is basically divided in to many parts but the main two are hardware and software. However is further divided into sensors and transmission of data. We have explained in modules as follows:

5.3.1 Sensor testing

We tried to test all the sensors. We connected each sensor individually to the arduino. For eg we connected Ultrasonic sensor for indicating the level of water. We understood the pin configuration of Arduino UNO and Wifi shield. We associated every sensor independently to the arduino. We associated ultrasonic for level indication. We comprehended the pin layout of arduino UNO and wifi shield. We relate each sensor autonomously to the arduino.

5.3.2 Wifi Connectivity

In this module tried to test ESP13 shield by sending sensors value to our server. We used ESP13 wifi module to connect our Arduino to the internet so that it can send data to out server. We tried to connect every sensor individually to the server separately in this module. Second module is spending the sensors values to our server i.e. thinkspeak. We can ESP13 wifi shield module to connect our arduino to the internet so that it can send data to our server.

5.3.3 Status Shown

Third module of the system is data fetching in which the data about the sensors is fetch from the respective servers, all this module are integrated in to one whole app module using user interface application.

5.3.4 Joining Together

We integrated all the sensors while the sensors uploading the value to our server. We used web based application for our server and we used shields for sensors as they were getting lose.

5.4 Systems Integration

First module of the system is connecting each sensors to the arduino individually and its pin configuration of arduino uno and wifi shield. Second module is sending the sensors values to our server. We used ESP8266 wifi shield to connect our arduino to the internet so that it can send data to our server. Third module of the system is

data fetching in which the data about the sensors is fetch from the respective servers. All this modules are integrated in to one whole user application using web based technology.

5.4.1 Sequence Diagram

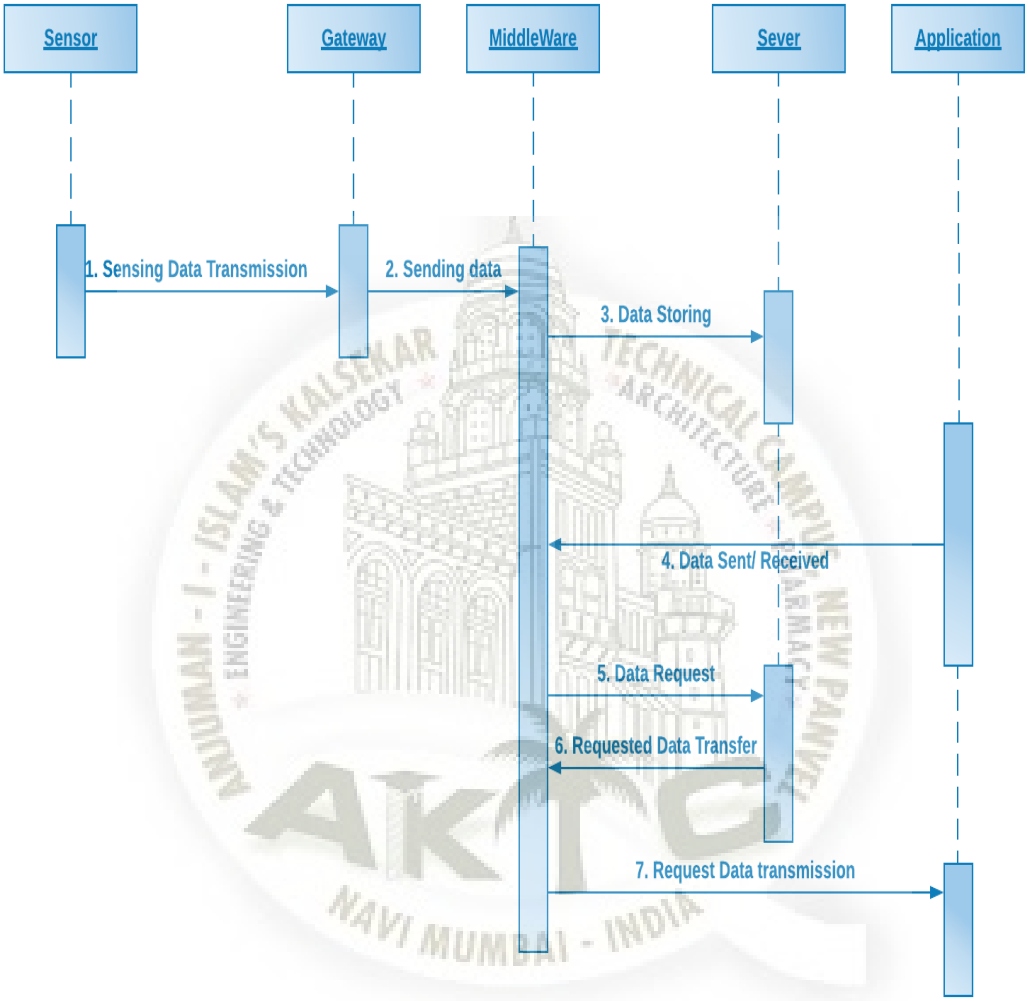

Figure 5.6: Sequence Diagram

In the water quality monitoring system using IoT, the sensors data will be transmitted through a set of different devices, to the cloud where it get processed, if a measurement crossed the threshold value, the cloud sends a command to the smart water system in order to perform the suitable action. The diagram below describes the system:

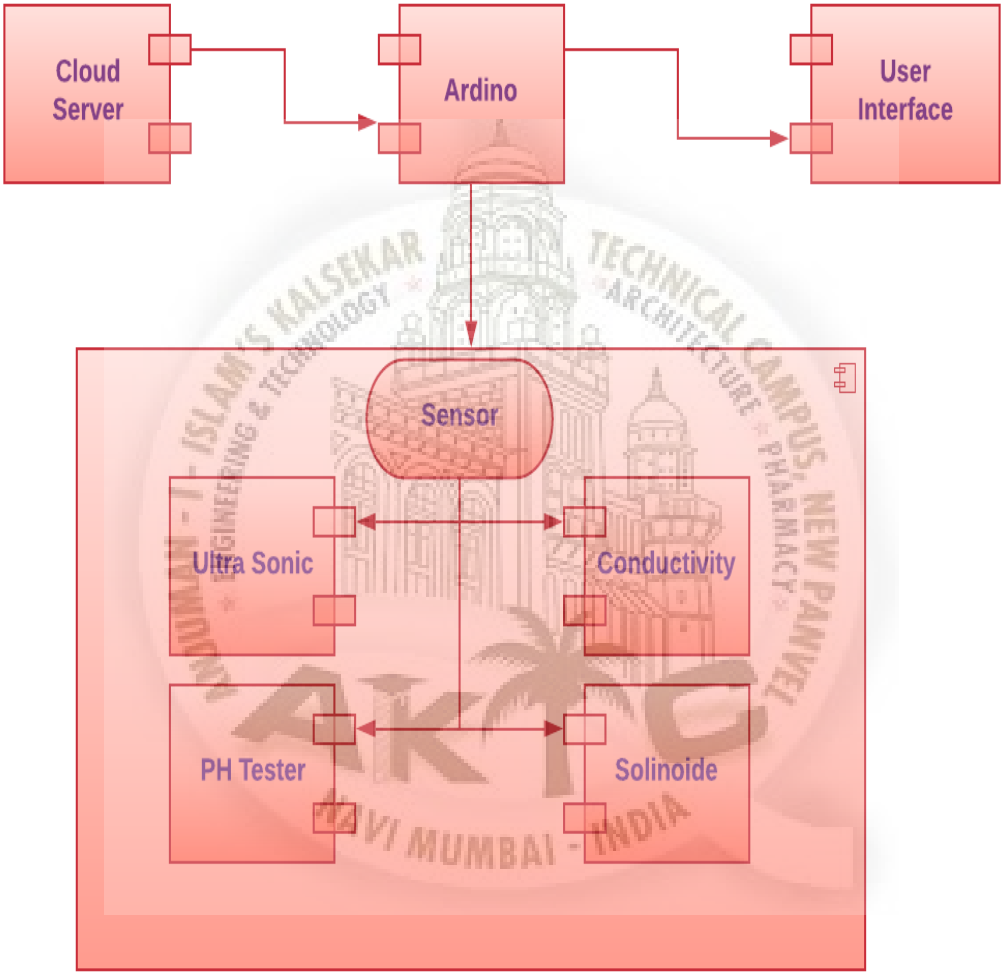

Figure 5.7: Component Diagram

5.4.2 Component Diagram

In component diagram it helps to understand the each module in better way forming its integrated sub module likewise shown in the diagram. IoT gateway allows a device to report data using its sensors to a remote location. Every one of these smart water system is connected to the same gateway forming a network, this gateway collects the data from all the sensors through arduino, filters it(depending on message priority) and sends it periodically to the cloud, which leads to a low cost as IoT cloud performs use message count as a payment unit.

5.4.3 Deployment Diagram

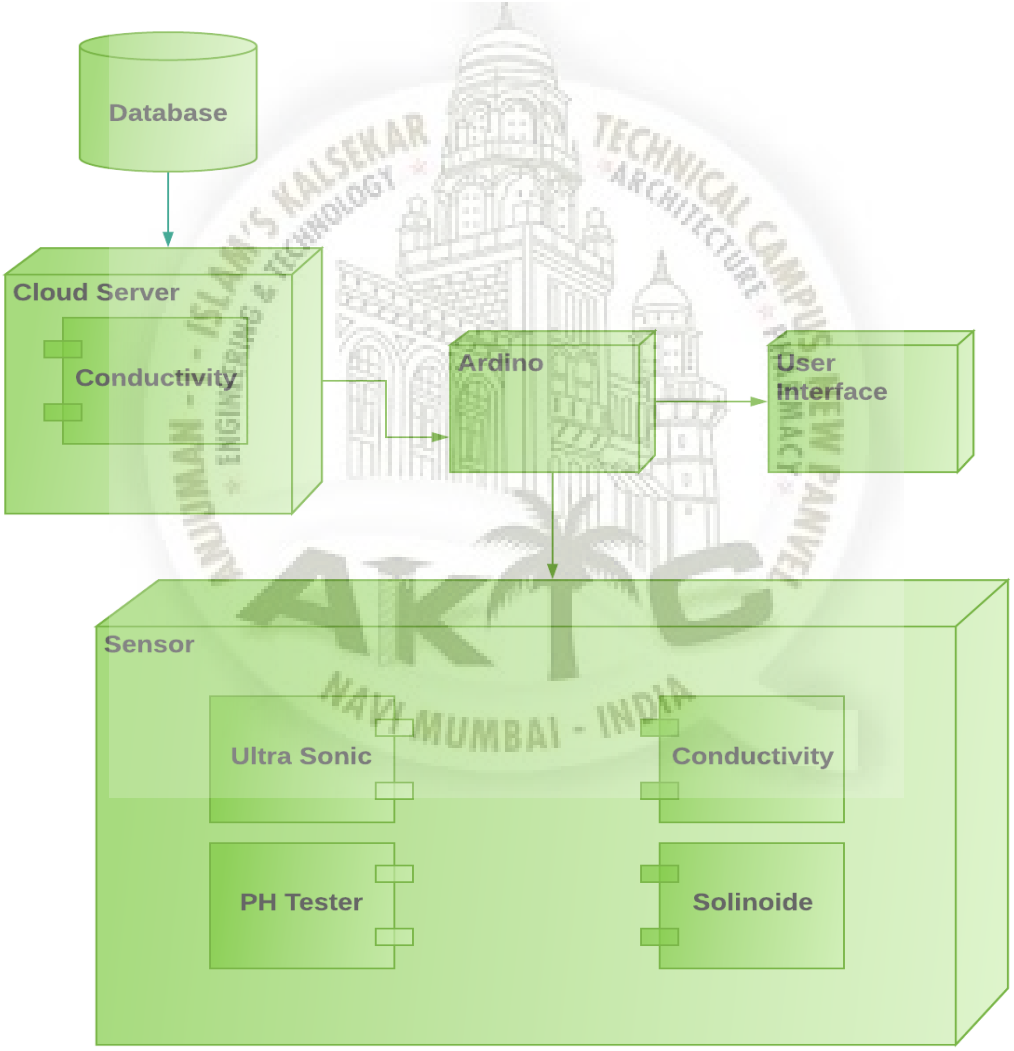

Figure 5.8: Deployment Diagram

Because the application has three major module like arduino, user interface, cloud server each one for a specific goal. The following wireframe represent the result and

explains the different parts that the application will have like the mobile application, web based application should offer a remote monitoring ability, information about the purity of the water, ability to control the actuator. This framework shows its functionality.

5.4.4 State Chart

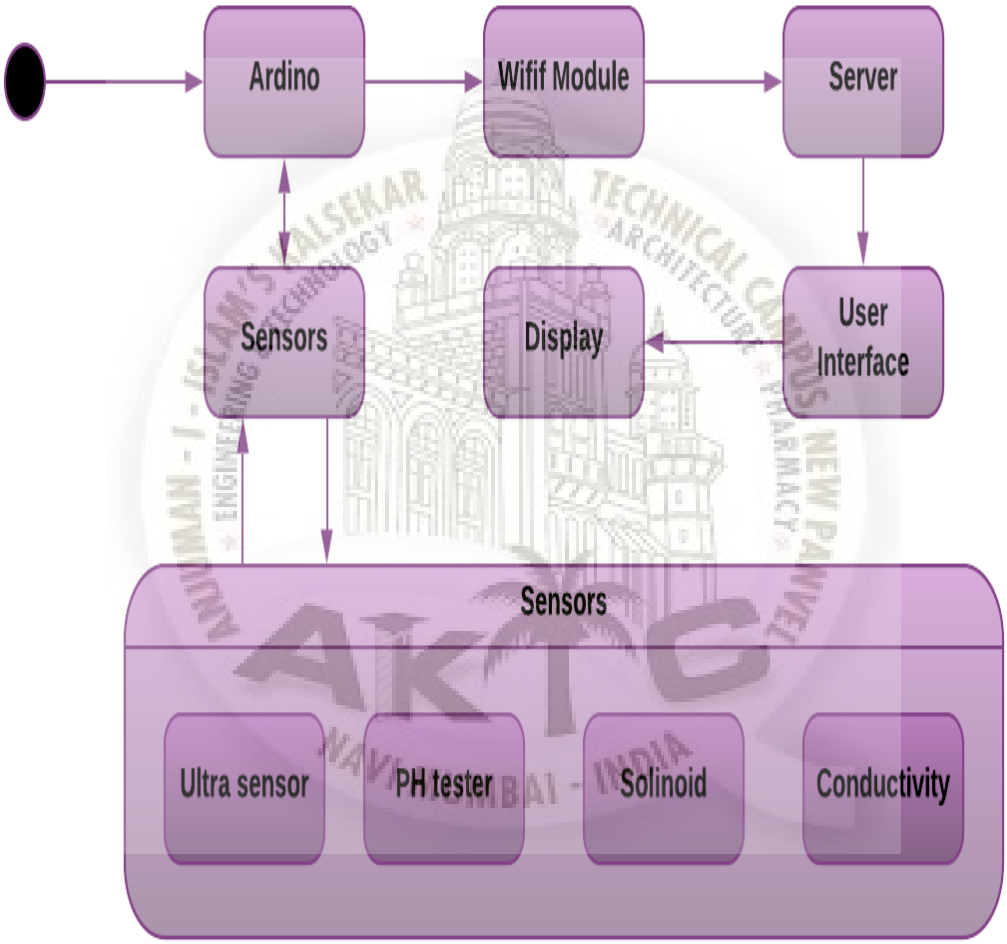

Figure 5.9: State Chart Diagram

The state chart diagram below represents the overall smart water quality diagram scenario. The idea behind the smart water system based on IoT is to allow the user to identify the purity of water as well as control the flow of water using IoT. By using web based application indicates the use of multi-platform user to access the control and data over the system.

5.4.5 Activity Diagram

Once authentication is successfully done, the application gets the data history and start sensing and displays it as a chart. When the user performs a manual action (enable / disable actuator of his choice), the application will send the command to the cloud, which will process it and send it to the gateway, then the gateway sends it to the specified smart water system.

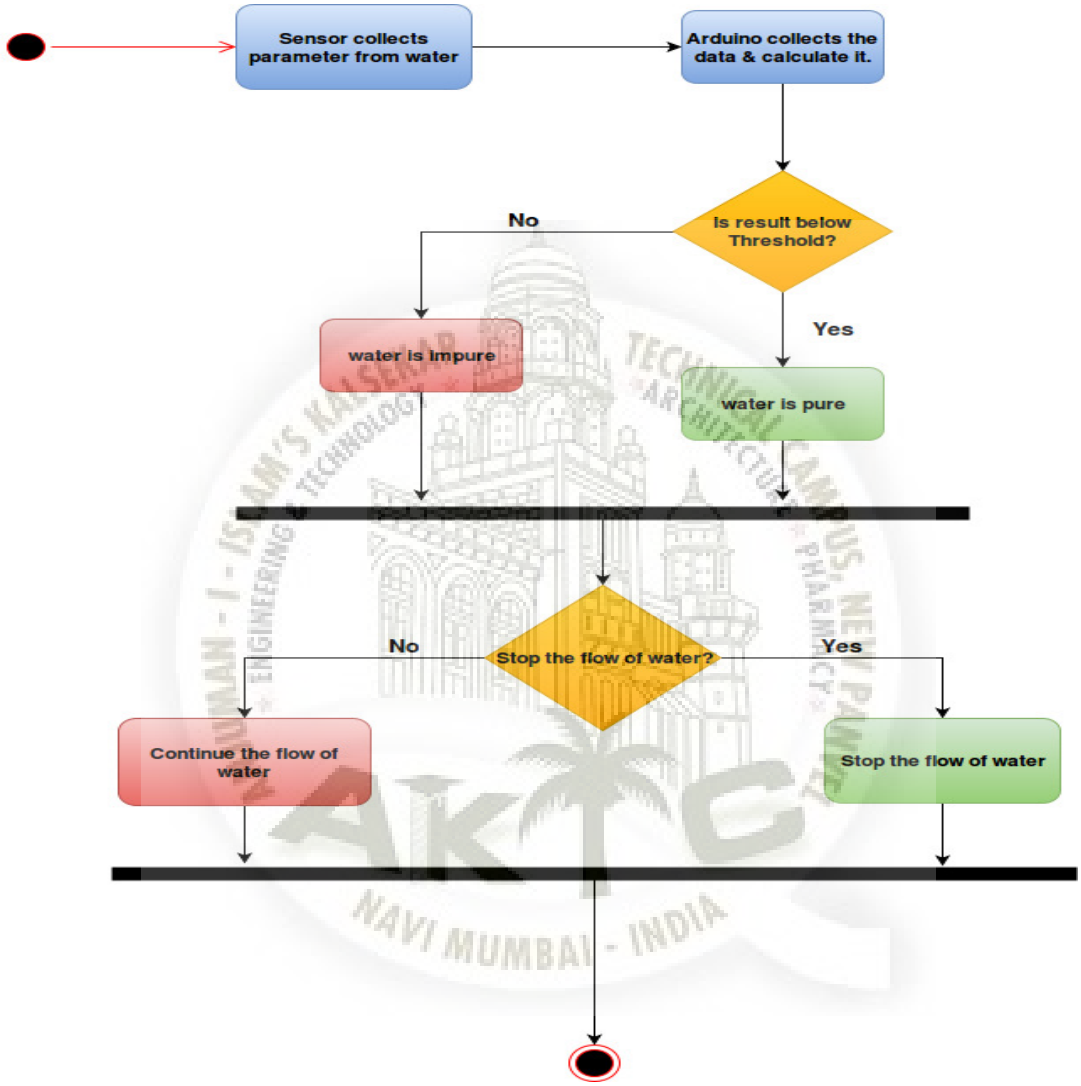

Figure 5.10: Activity Diagram

Chapter 6

Implementation

6.1 Sensors

In this module, the values of different sensors are viewed and read. Based on the individual sensor the value are obtained. The sensors that are used and their value obtained are:-

6.1.1 pH Tester

The pH tester test the purity value of water based on the pH value. The pH value are compared based on the range

* If pH value is less then 7, then the water is Acidic in nature.

* If pH value is greater then, then the water is Basic in nature.

* If pH value is equal to 7, then the water is neutral (pure).

Technical specification of pH tester

Module Power	5 V
pH measuring range	43mm x 32mm
Response time	¡=1min

Table 6.1: Technical specification of pH tester

Figure 6.1: pH Tester

```
#define SensorPin A0          //pH meter Analog output to Arduino Analog Input 0
#define Offset 0.00           //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth  40     //times of collection
int pHArray[ArrayLenth];   //Store the average value of the sensor feedback
int pHArrayIndex=0;
void setup(void)
{
  pinMode(LED,OUTPUT);
  Serial.begin(9600);
  Serial.println("pH meter experiment!");     //Test the serial monitor
}
void loop(void)
{
  static unsigned long samplingTime = millis();
  static unsigned long printTime = millis();
  static float pHValue,voltage;
  if(millis()-samplingTime > samplingInterval)
  {
      pHArray[pHArrayIndex++]=analogRead(SensorPin);
```

```
        if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
        voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
        pHValue = 3.5*voltage+Offset;
        samplingTime=millis();
    }
    if(millis() - printTime > printInterval)    //Every 800 milliseconds, print a
        numerical, convert the state of the LED indicator
    {
      Serial.print("Voltage:");
          Serial.print(voltage,2);
          Serial.print("    pH value: ");
      Serial.println(pHValue,2);
          digitalWrite(LED,digitalRead(LED)^1);
          printTime=millis();
    }
}
double avergearray(int* arr, int number){
    int i;
    int max,min;
    double avg;
    long amount=0;
    if(number<=0){
      Serial.println("Error number for the array to avraging!/n");
      return 0;
    }
    if(number<5){   //less than 5, calculated directly statistics
      for(i=0;i<number;i++){
        amount+=arr[i];
      }
      avg = amount/number;
      return avg;
    }else{
      if(arr[0]<arr[1]){
        min = arr[0];max=arr[1];
      }
      else{
        min=arr[1];max=arr[0];
      }
      for(i=2;i<number;i++){
        if(arr[i]<min){
          amount+=min;        //arr<min
          min=arr[i];
        }else {
          if(arr[i]>max){
            amount+=max;      //arr>max
            max=arr[i];
          }else{
            amount+=arr[i];  //min<=arr<=max
          }
        }//if
      }//for
      avg = (double)amount/(number-2);
    }//if
    return avg;
}
```

6.1.2 Ultra-Sonic

This sensor measures the distance between any two objects or elements. Similarly in water monitoring it will monitor the level of water in tank. If the level of water is high,low or medium the same information is informed to the user and on the basis of the notification user can control the flow of water.

Technical specification of Ultra-sonic Sensor

power supply	5V DC
Working current	15mA
Distance range	2cm- 400cm
Resolution	0.3cm

Table 6.2: Technical specification of Ultra-sonic Sensor

Figure 6.2: Ultra-sonic

```
1
2  // defines pins numbers
3  const int trigPin = 9;
4  const int echoPin = 10;
5  // defines variables
6  long duration;
7  int distance;
8  void setup() {
9  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
10 pinMode(echoPin, INPUT); // Sets the echoPin as an Input
11 Serial.begin(9600); // Starts the serial communication
12 }
13 void loop() {
14 // Clears the trigPin
15 digitalWrite(trigPin, LOW);
```

```
16
17 delayMicroseconds(100);
18 // Sets the trigPin on HIGH state for 10 micro seconds
19 digitalWrite(trigPin, HIGH);
20
21 delayMicroseconds(100);
22 digitalWrite(trigPin, LOW);
23 // Reads the echoPin, returns the sound wave travel time in microseconds
24 duration = pulseIn(echoPin, HIGH);
25 // Calculating the distance
26
27 distance= duration*0.034/2;    //(timeXspeedOfSound/2) 0.034=speedofsound of
       ultra, 2=formula for ultra cuz it travels and echo backs.
28
29 // Prints the distance on the Serial Monitor
30 Serial.print("Distance: ");
31 Serial.println(distance);
32 if(distance >=199){
33 Serial.print("LOW");
34 }
35 else if(distance >=99){
36 Serial.print("Medium");
37 }
38 else if(distance >=25)
39 {
40 Serial.print("HIGH");
41 }
42 else{
43  Serial.print("OVERFLOW");
44 }
45
46 }
```

6.1.3 Conductivity

The conductivity detects the level of tubidity in water. If the turbid level is higher then the usual level then the user is informed about it with the help of report which is generated after analyzing and comparing the value. This sensors helps in identifying the level of chlorine in water.

Technical specification of Conductivity Sensor

conductivity reading accuracy	0.5 percent
Measurement span	0 to 2 siemens/cm
Measure conductivity	up to 82 feet depth

Table 6.3: Technical specification of Conductivity sensor

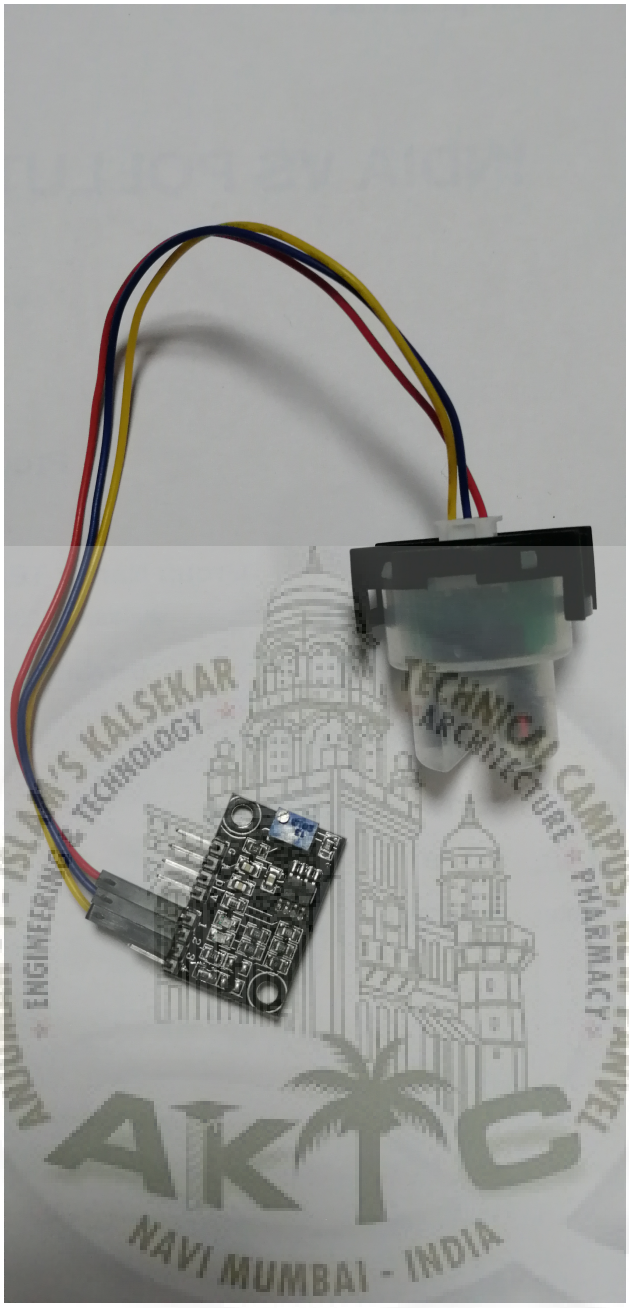

Figure 6.3: Conductivity

```
  void setup() {
  Serial.begin(9600); //Baud rate: 9600
}
void loop() {
  int sensorValue = analogRead(A0);// read the input on analog pin 0:
  float voltage = sensorValue * (5.0 / 1024.0); // Convert the analog reading (
      which goes from 0 - 1023) to a voltage (0 - 5V):
  Serial.println(voltage); // print out the value you read:
  delay(500);
}
```

6.1.4 Solenoid valve

A solenoid valve is an electro-mechanical device in which the solenoid uses an electric current to generate a magnetic field and thereby operate a mechanism which regulates the opening of fluid flow in a valve.
Solenoid valves are the most frequently used control elements in fluidics. Their tasks are to shut off, release, dose, distribute or mix fluids. They are found in many application areas. Solenoids offer fast and safe switching, high reliability, long service life, good medium compatibility of the materials used, low control power and compact design.

Technical specification of Solenoid valve

Body Materials	2S Series: Stainless Steel; 2W Series Brass
Coil Duty	100 %
Operating Pressure 220 AC Coil) , 0 to 120 (12	0 to 150 PSI (110 24 DC, 24VAC coil)
Service	Air, Gas, Liquid, Vacuum

Table 6.4: Technical specification of Conductivity sensor

Figure 6.4: Solenoid valve

```
int solenoidPin = 9;                    //This is the output pin on the Arduino

void setup()
```

```
{
  pinMode(solenoidPin, OUTPUT);           //Sets that pin as an output
}

void loop()
{
  digitalWrite(solenoidPin, HIGH);       //Switch Solenoid ON
  delay(1000);                           //Wait 1 Second
  digitalWrite(solenoidPin, LOW);        //Switch Solenoid OFF
  delay(1000);                           //Wait 1 Second
}
```

6.2 Working of Sensors together

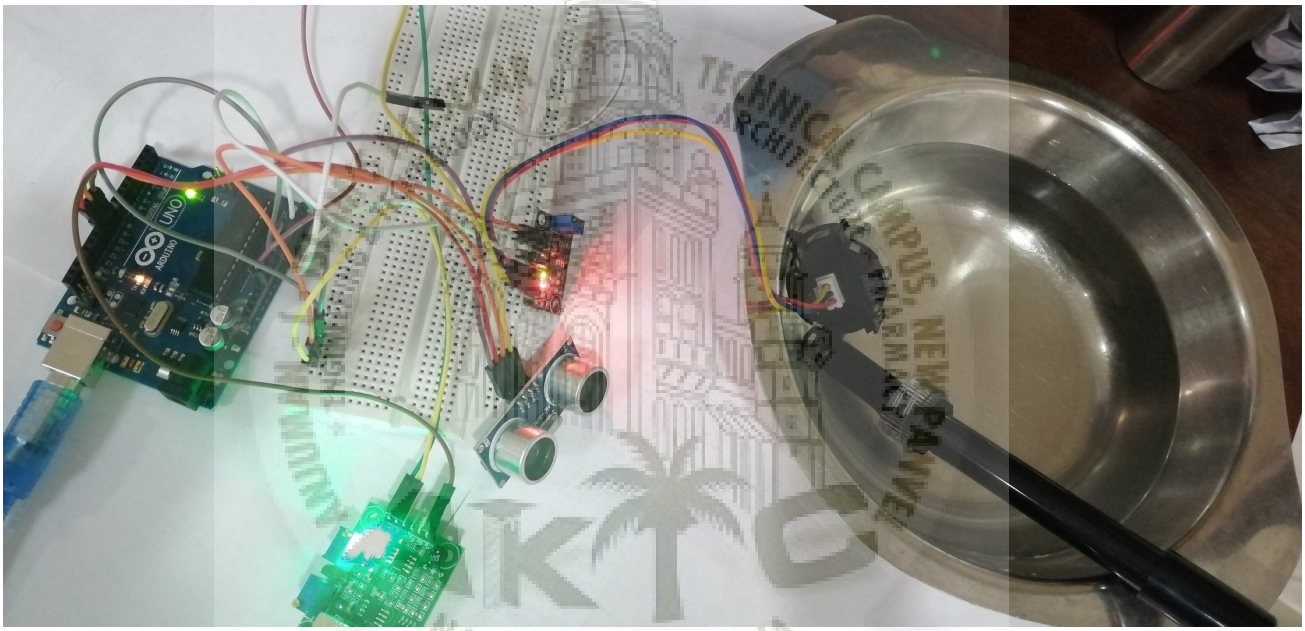

Figure 6.5: Working of Sensors

```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
// defines variables
long duration;
int distance;
float voltage,turbidity,voltage1;
#define SENSOR A0

#define SensorPin A0            //pH meter Analog output to Arduino Analog Input
    0
#define Offset 0.00            //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth  40     //times of collection
int pHArray[ArrayLenth];   //Store the average value of the sensor feedback
int pHArrayIndex=0;
```

```
double avergearray(int* arr, int number){
  int i;
  int max,min;
  double avg;
  long amount=0;
  if(number<=0){
    Serial.println("Error number for the array to avraging!/n");
    return 0;
  }
  if(number<5){   //less than 5, calculated directly statistics
    for(i=0;i<number;i++){
      amount+=arr[i];
    }
    avg = amount/number;
    return avg;
  }else{
    if(arr[0]<arr[1]){
      min = arr[0];max=arr[1];
    }
    else{
      min=arr[1];max=arr[0];
    }
    for(i=2;i<number;i++){
      if(arr[i]<min){
        amount+=min;        //arr<min
        min=arr[i];
      }else {
        if(arr[i]>max){
          amount+=max;      //arr>max
          max=arr[i];
        }else{
          amount+=arr[i]; //min<=arr<=max
        }
      }//if
    }//for
    avg = (double)amount/(number-2);
  }//if
  return avg;
}
void setup() {
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.begin(9600); // Starts the serial communication
pinMode(SENSOR,INPUT);

  pinMode(LED,OUTPUT);
  Serial.begin(9600);
  Serial.println("pH meter experiment!");    //Test the serial monitor
}
void loop() {
  voltage=0.004888*analogRead(SENSOR);  //in V
  turbidity=-1120.4*voltage*voltage+5742.3*voltage-4352.9;  //in NTU
  turbidity=turbidity/1000;
  if((voltage>=2.5)&(turbidity>=0))
  {
    Serial.println("Voltage="+String(voltage)+" V Turbidity="+String(turbidity)+
        " NTU");
    delay(2000);
  }
// Clears the trigPin
digitalWrite(trigPin, LOW);
```

```
78
79 delayMicroseconds(100);
80 // Sets the trigPin on HIGH state for 10 micro seconds
81 digitalWrite(trigPin, HIGH);
82
83 delayMicroseconds(100);
84 digitalWrite(trigPin, LOW);
85 // Reads the echoPin, returns the sound wave travel time in microseconds
86 duration = pulseIn(echoPin, HIGH);
87 // Calculating the distance
88
89 distance= duration*0.034/2;    //(timeXspeedOfSound/2) 0.034=speedofsound of
       ultra, 2=formula for ultra cuz it travels and echo backs.
90
91 // Prints the distance on the Serial Monitor
92 Serial.print("Distance: ");
93 Serial.println(distance);
94 if(distance >=199){
95 Serial.println("LOW");
96 }
97 else if(distance >=99){
98 Serial.println("Medium");
99 }
100 else if(distance >=25)
101 {
102 Serial.println("HIGH");
103 }
104 else{
105  Serial.println("OVERFLOW");
106 }
107
108 static unsigned long samplingTime = millis();
109   static unsigned long printTime = millis();
110   static float pHValue,voltage;
111   if(millis()-samplingTime > samplingInterval)
112   {
113        pHArray[pHArrayIndex++]=analogRead(SensorPin);
114        if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
115        voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
116        pHValue = 3.5*voltage+Offset;
117        samplingTime=millis();
118   }
119   if(millis() - printTime > printInterval)   //Every 800 milliseconds, print a
       numerical, convert the state of the LED indicator
120   {
121     Serial.print("Voltage:");
122          Serial.print(voltage,2);
123          Serial.print("    pH value: ");
124     Serial.println(pHValue,2);
125          digitalWrite(LED,digitalRead(LED)^1);
126          printTime=millis();
127   }
128
129 }
```

6.3 Front End

In our system, the user interface for the front end basically is been implemented with the intention of working for multi-platform. SO the front end mainly compromises of registration of the user and the data representation of the data from sensors with the help of cloud server.

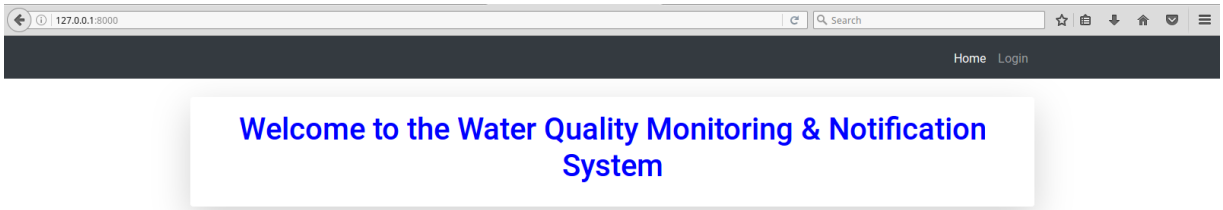

Figure 6.6: Front End

```
//ADMIN.PY

from django.contrib import admin
from .models import values
# Register your models here.
admin.site.register(values)

//APPS.PY

from django.apps import AppConfig
class AccountsConfig(AppConfig):
    name = 'accounts'

//MODELS.PY

from django.db import models
# Create your models here.
class values(models.Model):
    def __str__(self):
        id = self.id
        return '{}'.format(id)

    wl_choice = (('low','low'),('medium','medium'),('high','high'))
    purity_values = (('pure','pure'),('impure','impure'))
    ph_values = (('neutral','neutral'),('alkaline','alkaline'),('acidic','acidic
        '))


```

```
    ph_value = models.CharField(choices=ph_values,default ='neutral',max_length=
        10)
    water_level = models.CharField(choices = wl_choice,default = 'low',
        max_length =10)
    turbidity = models.FloatField(null=True, blank=True, default=0.0)
    purity = models.CharField(choices=purity_values,default ='pure',max_length=
        10)

    //URLS.PY

    from django.urls import include, path
from . import views
app_name = 'accounts'

urlpatterns = [
    path('', views.index, name='home'),
    path('login/', views.login_user, name='login'),
    path('logout/', views.logout_user, name='logout'),
    path('status/', views.status, name='status'),
]

//VIEWS.PY

from django.shortcuts import get_object_or_404, render, redirect
from django.contrib.auth import authenticate, login, logout,
    update_session_auth_hash
from django.contrib import messages
from django.contrib.auth.decorators import login_required
#from django.contrib.auth.forms import UserChangeForm, PasswordChangeForm
#from .forms import EditProfileForm
#from django.db.models import Count
from django.contrib.auth.models import User
#from inventory.models import Product_table, Req_issue_item, Req_maintenance,
    Res_issue_item
from .models import values
def index(request):
    return render(request, 'accounts/index.html')

def login_user(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)

        if user is not None:
            login(request, user)
            messages.success(request, ('You have successfully logged in !'))
            return redirect('accounts:home')
        else:
            messages.success(request, ('You have failed to login !'))
            return redirect('accounts:login')
    else:
        return render(request, 'accounts/login.html')

def logout_user(request):
    logout(request)
    messages.success(request, ('You have successfullylogout...'))
    return redirect('accounts:home')

@login_required(login_url='accounts:login')
def status(request):
```

```
    mt = values.objects.all().last()
    return render(request, 'accounts/status.html', {'mt': mt})

//SETTINGS.PY

"""
Django settings for waterquality project.
Generated by 'django-admin startproject' using Django 2.1.5.
For more information on this file, see
https://docs.djangoproject.com/en/2.1/topics/settings/
For the full list of settings and their values, see
https://docs.djangoproject.com/en/2.1/ref/settings/
"""
import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.1/howto/deployment/checklist/
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'h)+a&a(^#qgib+@xw-_*1tn&m%au61**9e=4u*h_l3ex4mzv0'
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'accounts.apps.AccountsConfig',
    'wc.apps.WcConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'waterquality.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
```

```
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'waterquality.wsgi.application'

# Database
# https://docs.djangoproject.com/en/2.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}

# Password validation
# https://docs.djangoproject.com/en/2.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.
            UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator'
            ,
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator
            ',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.
            NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/2.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.1/howto/static-files/

STATIC_URL = '/static/'

//URLS.PY
```

```
"""waterquality URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/2.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('accounts.urls')),
]

//WSGI.PY

"""
WSGI config for waterquality project.
It exposes the WSGI callable as a module-level variable named ``application``.
For more information on this file, see
https://docs.djangoproject.com/en/2.1/howto/deployment/wsgi/
"""
import os

from django.core.wsgi import get_wsgi_application
os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'waterquality.settings')
application = get_wsgi_application()

//APPS.PY

from django.apps import AppConfig
class WcConfig(AppConfig):
    name = 'wc'

//MANAGE.PY
#!/usr/bin/env python
import os
import sys

if __name__ == '__main__':
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'waterquality.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
```

```
263
264 //BASE.HTML
265
266
267 {% load static %}
268 <!doctype html>
269 <html lang="en">
270     <head>
271         <!-- Required meta tags -->
272         <meta charset="utf-8">
273         <meta name="viewport" content="width=device-width, initial-scale=1,
                shrink-to-fit=no">
274
275         <!-- Bootstrap CSS -->
276         <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/
                bootstrap/4.2.1/css/bootstrap.min.css" integrity="sha384-
                GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS"
                crossorigin="anonymous">
277
278         <title>Water Quality</title>
279
280         <nav class="navbar navbar-expand-lg navbar-dark bg-dark ">
281             <div class="container">
282                 <a class="navbar-brand" href="{% url 'accounts:home'%}"></a>
283                 <button class="navbar-toggler" type="button" data-toggle="
                        collapse" data-target="#navbarSupportedContent" aria-
                        controls="navbarSupportedContent" aria-expanded="false" aria
                        -label="Toggle navigation">
284                 <span class="navbar-toggler-icon"></span></button>
285
286                 <div class="collapse navbar-collapse" id="navbarSupportedContent
                    ">
287                     <!--<nav class="navbar navbar-expand-sm bg-info navbar-dark
                        ">-->
288
289                     <ul class="navbar-nav ml-auto">
290                         <li class="nav-item active">
291                             <a class="nav-link" href="{% url 'accounts:home'
                                %}">Home <span class="sr-only">(current)</span
                                ></a>
292                         </li>
293
294                         {% if user.is_authenticated %}
295
296                     <!--  <li class="nav-item">
297                             <a class="nav-link" href="{% url 'accounts:status'
                                %}">FlowControl ON</a>
298                         </li>
299
300                         <li class="nav-item">
301                             <a class="nav-link" href="{% url 'accounts:
                                status' %}">FlowControl OFF</a>
302                         </li> -->
303
304                         <li class="nav-item">
305                             <a class="nav-link" href="{% url 'accounts:status'
                                %}">Status</a>
306                         </li>
307                         <li class="nav-item">
308                             <a class="nav-link" href="{% url 'accounts:logout'
                                %}">Logout</a>
```

```
                        </li>

                        {% else %}

                        <li class="nav-item">
                            <a class="nav-link" href="{% url 'accounts:login'
                                %}">Login</a>
                        </li>

                        {% endif %}

                    </ul>
                <!--</nav>-->
                </div>
            </div>
        </nav>
    </head>
<!-- <style>
            body {
              background-image: url("{% static 'mysite/images/drop.png'%}");
              background-repeat: repeat;
              background-size: 70px 70px;
            }
    </style> -->

    <body>
    <div class="row">
        <div class="container">
            {% if messages %}
                {% for message in messages %}
                    <div class="alert alert-warning" role="alert">
                        <button class="close" data-dismissable="alert"><small><
                            sup>X</sup></small></button>
                        {{ message }}
                    </div>

                {% endfor %}
            {% endif %}
            {% block content %}
            {% endblock %}
        </div>
    </div>

    <iframe width="425" height="349" src="https://www.youtube.com/watch?v=
        gcnF1LRiMU4" frameborder="0" allowfullscreen></iframe>
        <!-- Optional JavaScript -->
        <!-- jQuery first, then Popper.js, then Bootstrap JS -->
        <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity
            ="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
            abtTE1Pi6jizo" crossorigin="anonymous"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd
            /popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/
            yoi7FRNXMRBu5WHdZYu1hA6ZOblgut" crossorigin="anonymous"></script>
        <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/
            bootstrap.min.js" integrity="sha384-B0UglyR+
            jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k"
            crossorigin="anonymous"></script>
    </body>
</html>

//INDEX.HTML
```

```
360
361
362 {% extends "accounts/base.html" %}
363 {% load static %}
364 {% block content %}
365         <br>
366         <div class="container shadow-lg p-3 mb-5 bg-white rounded">
367                 <h1 class="text-center"><p style="color:blue",background color="
                        color:green";>Welcome to the Water Quality Monitoring &
                        Notification System </h1></p>
368         </div>
369
370         <div class="embed-responsive embed-responsive-4by3">
371                 <iframe class="embed-responsive-item" src="https://www.youtube.
                        com/watch?v=gcnF1LRiMU4"></iframe>
372         </div>
373
374 {% endblock content %}
375
376 //LOGIN.HTML
377
378 {% extends "accounts/base.html" %}
379 {% block content %}
380
381 <div class="container shadow-lg p-3 mb-5 bg-white rounded">
382     <h1 class="text-center">Continue to Login...</h1>
383     <div class="col-md-6 offset-md-3 shadow p-3 mb-5 bg-white rounded">
384         <br>
385         <div class="col-md-8 offset-md-2">
386             <form method="POST">
387             {% csrf_token %}
388                 <div class="form-group">
389                     <input type="text" class="form-control" placeholder="Enter
                            Username" name="username">
390                 </div>
391                 <div class="form-group">
392                     <input type="password" class="form-control" placeholder="
                            Enter Password" name="password">
393                 </div>
394                 <div class="text-center">
395                     <button type="submit" class="btn btn-primary center-block">
                            Login</button>
396                 </div>
397             </form>
398
399             <br>
400             <div class="text-center">
401             <a href="#">Click here to reset password</a>
402             </div>
403             <br>
404         </div>
405     </div>
406 </div>
407
408 {% endblock content %}
409
410 //STATUS.HTML
411
412 {% extends "accounts/base.html" %}
413 {% block content %}
414
```

```
<br>
<div class="container shadow-lg p-3 mb-5 bg-white rounded">
    <h1 class="text-center">Satus</h1>
    <div class="col-md-6 offset-md-3 shadow p-3 mb-5 bg-white rounded">
        {% if mt is not Null %}
            <div class="row">
                <div class="col-6"><h3>Turbidity</h3></div>
                <div class="col-1"><h3>:</h3></div>
                <div class="col-5  "><h3>{{ mt.turbidity }}</h3></div>
            </div>
            <div class="row">
                <div class="col-6"><h3>Water Level</h3></div>
                <div class="col-1"><h3>:</h3></div>
                <div class="col-5"><h3>{{ mt.water_level }}</h3></div>
            </div>
            <div class="row">
                <div class="col-6"><h3>PH Value</h3></div>
                <div class="col-1"><h3>:</h3></div>
                <div class="col-5"><h3>{{ mt.ph_value }}</h3></div>
            </div>
            <div class="row">
                <div class="col-6"><h3>Purity of Water</h3></div>
                <div class="col-1"><h3>:</h3></div>
                <div class="col-5"><h3>{{ mt.purity }}</h3></div>
            </div>
        {% else %}
            <h1>Empty DataSet</h1>
        {% endif %}
    </div>
</div>

<script language="javascript">
setTimeout(function(){
   window.location.reload(1);
}, 333000);
</script>

{% endblock content %}
```

Chapter 7

System Testing

First the system will check the quality of water based on the parameters. If it gives the output successfully then the data can be fetched to cloud server through which data will be transmitted to the end user.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Testing of sensors	Working	Loads the value successfully	Successfull
T02	Connection to cloud server	Unable to connect	Unable to communicate	No result
T03	End user	Shows the data	Unable to communicate with cloud server	dependency

7.2 Sample of a Test Case

Title: Monitoring the quality of water with the help of IOT
Description: Before any final implementation, testing is the most important part of the scenario since its give you the conditions which can manipulate the system in case. Our system is based on IOT in which the use of sensors is to detect the quality of water based on parameters and to inform the user the purity level of water. The sensors in these fetch the quality and with the help of arduino and esp8266 wifi module, the data for the same is fetch to cloud server called as THINGSPEAK using API of the server. The user too can get the data that is been monitored and stored on thingspeak by user interface with cloud

server. Once our connection with the cloud server is done successfully the fetching and storing of data can be achieved successfully.

Precondition: User must be connected with the internet so as to get the data.

Assumption: A supported cloud server is been used.

Test Steps:

1. Installation of Arduino IDE and proper libraries and drivers required.
2. Working with the connection of sensors and arduino uno.
3. Connect the ESP8266 wifi module to the server.
4. Connect the end application with the cloud server (THINGSPEAK).
5. Plug the USB with arduino and computer and start the working.
6. System working will begin.
7. User can control the flow of water based on the requirement.

Expected Result: To get the quality parameters through communication betwen arduino and thingspeak to the the user interface.

Actual Result: Communication of the wifi to the cloud server was unable.

Chapter 8

Screenshots of Project

8.1 Working of sensors

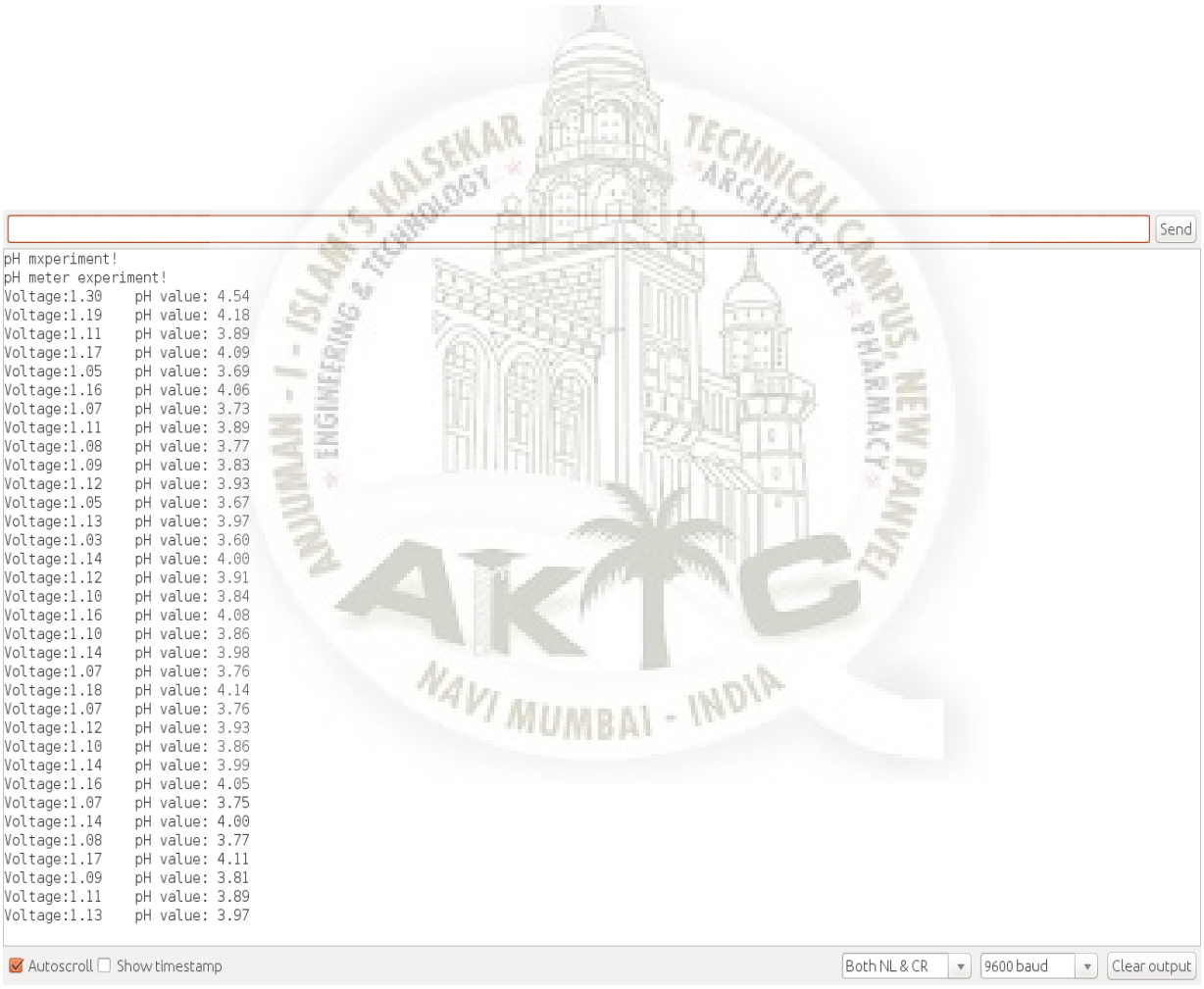

Figure 8.1: Output of pH tester

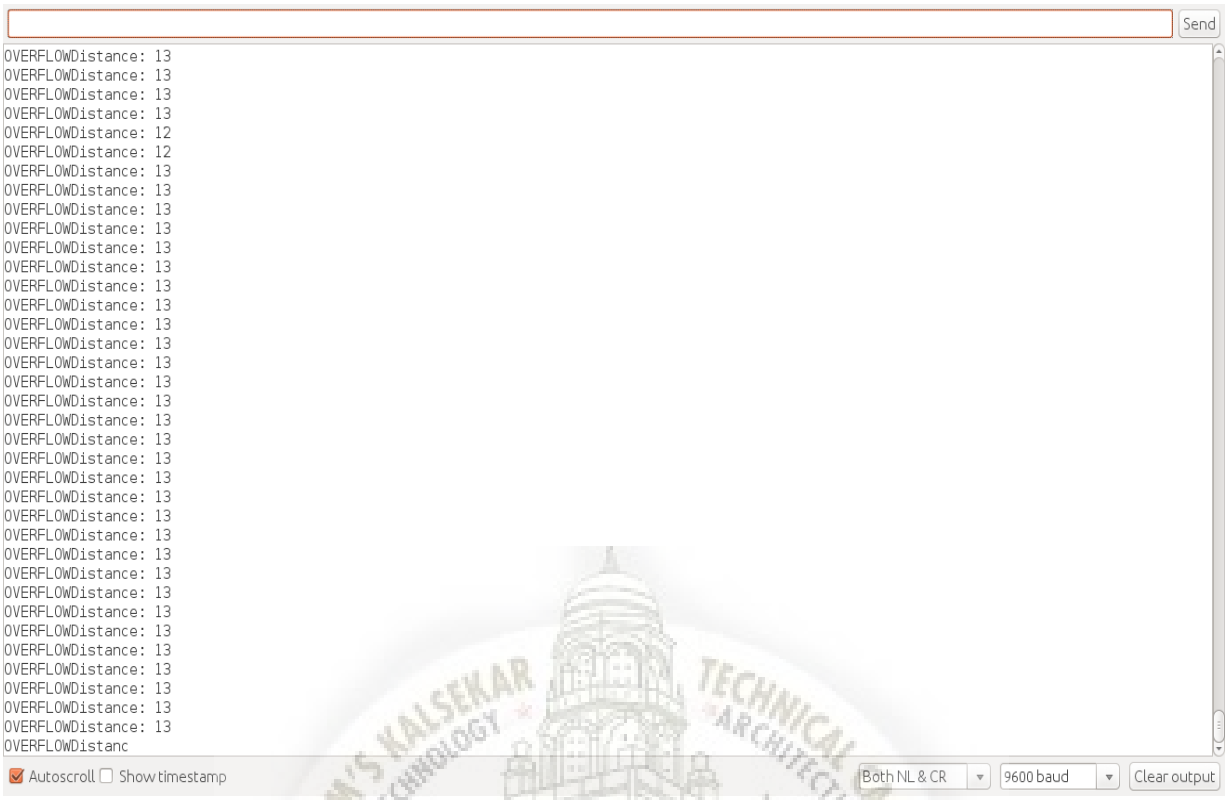

Figure 8.2: Output of Ultra-sonic

Figure 8.3: Output of Conductivity

8.2 Working of Sensors together

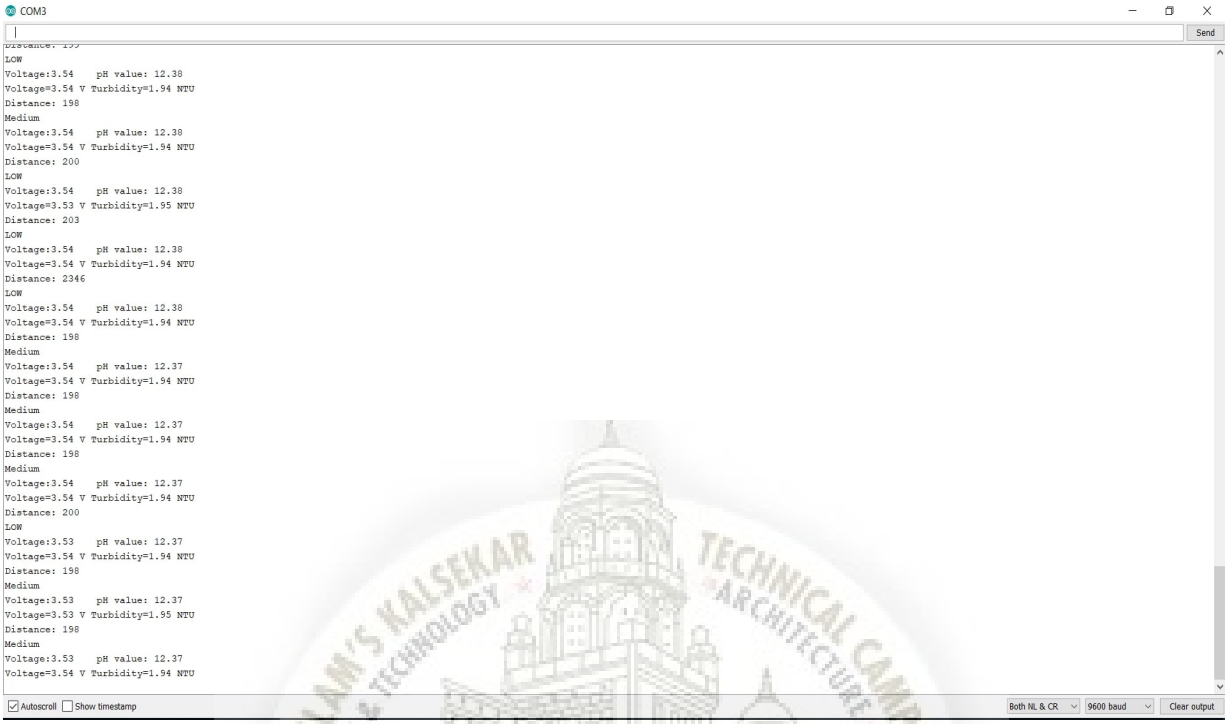

Figure 8.4: Output of All Senors Reading

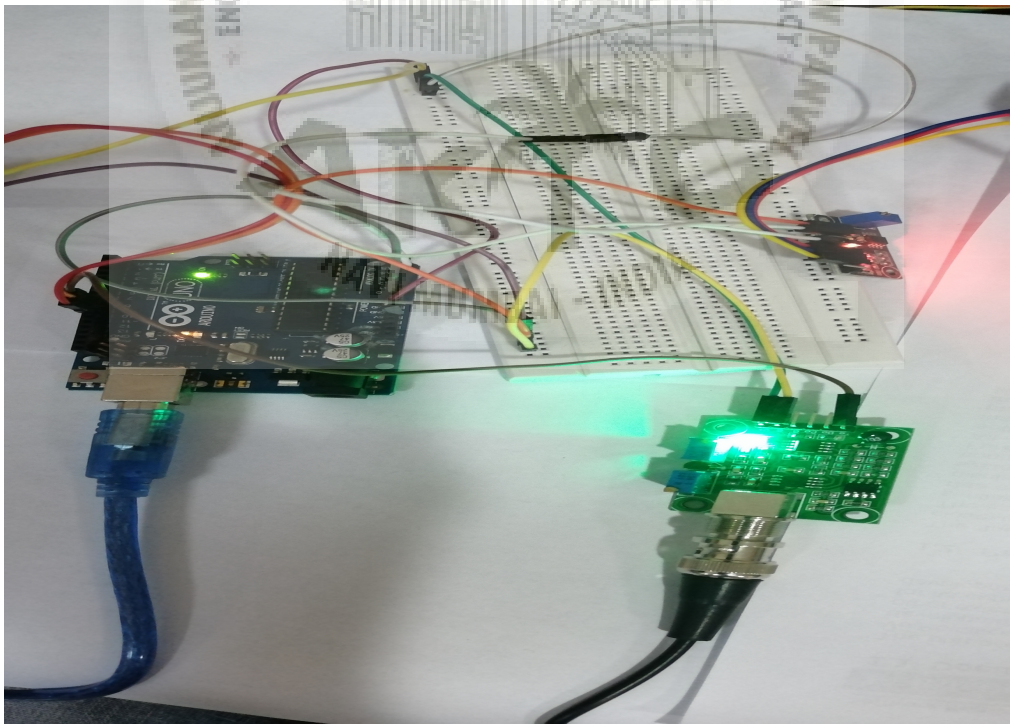

Figure 8.5: Connection To Arduino

8.3 Front End

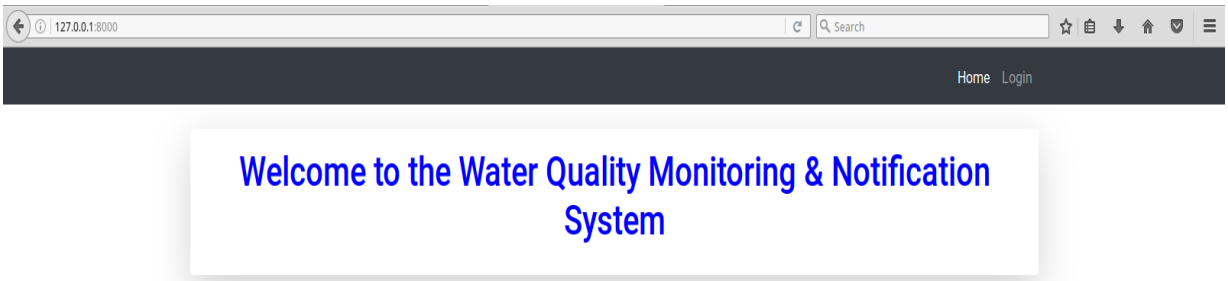

Figure 8.6: Output of Front End

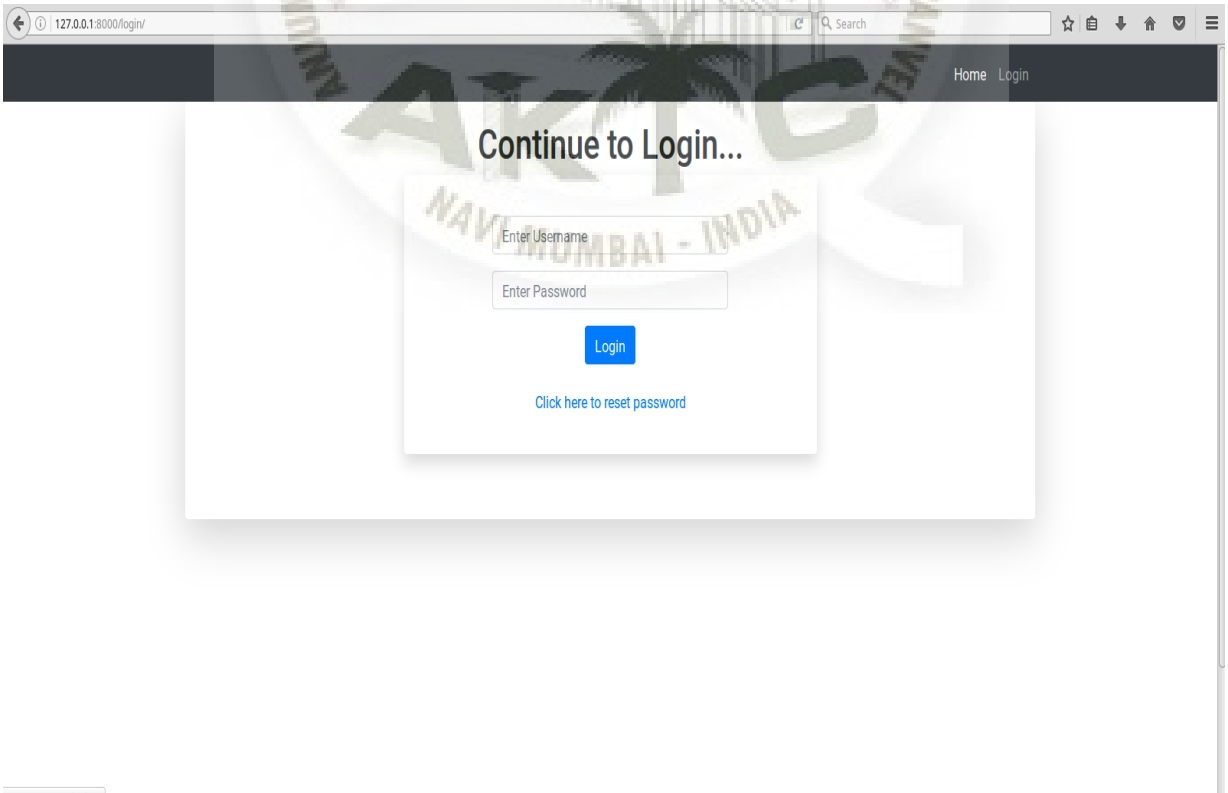

Figure 8.7: Output of Login Page

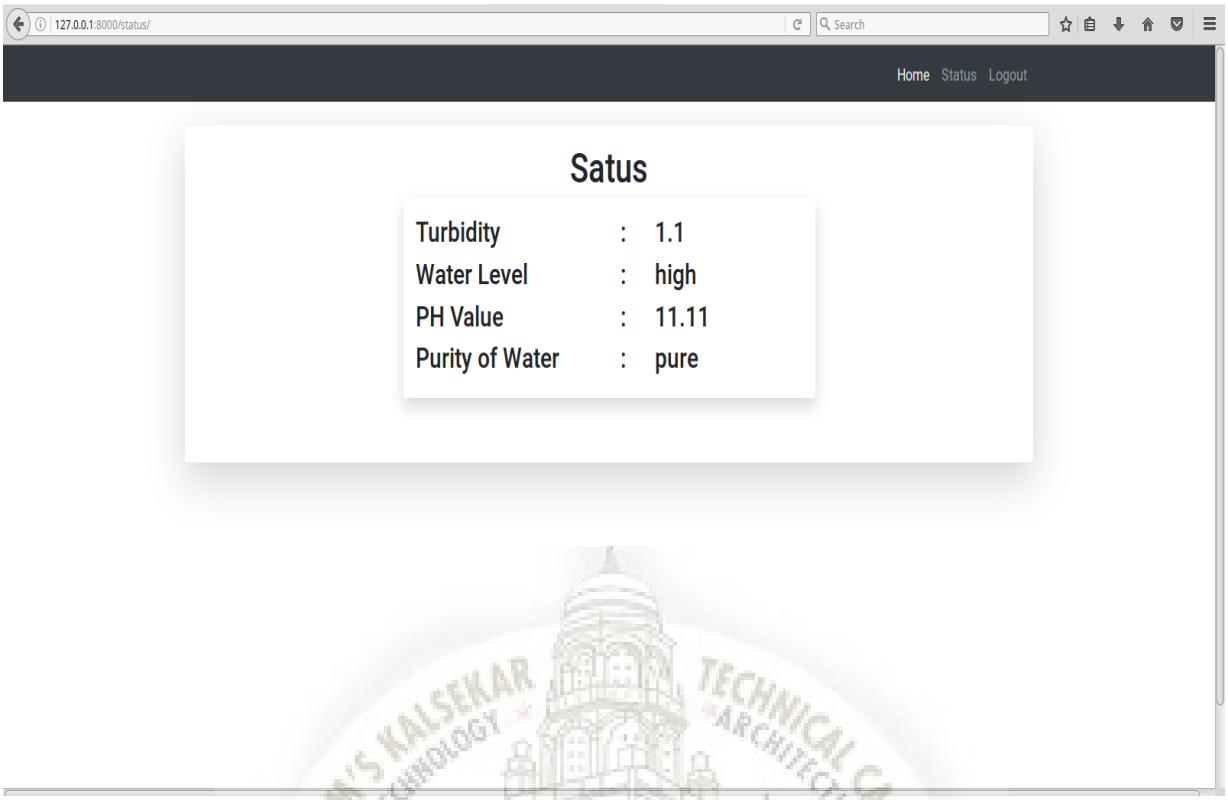

Figure 8.8: Output of Status of Water Quality

Figure 8.9: Output of Log Out

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

This system in based on real-time water quality analyzing which has low cost maintenance and easy to use. With the help of sensors many harmful effects to society can be avoided and this can be well organized if it is governed properly, through which people can be notified or alarmed for the water quality they shall use which can prevent them from getting infected by various harmful chemicals, metals, pollutants and bacteria.

Since it is based on IoT, it has wide range of uses and application which can be taken into demonstration which can result in change in human lifestyle in various aspects, one of them is this system which tries to overcome the problem that arises through water by detecting the quality of water and informing the same to the user. On prior notification a user can decide whether the water must be taken into use or not depending on the quality it has been mentioned. For this system to be operated a user need not have to be well trained or instructed. It is easy to use and operate.

9.2 Future Scope

Looking at its working and compatibility we look forward to Need to work on platform which can give :-

- To work on identifying more quality of parameters such as Nutrients present in water.
- To work on traditional purifier by adding this system with it.
- To work purification of water along with this system

References

[1] *Smart Water Management Using IOT*; Sayali Wadekar, Vinayak Vakare, Ramratan Prajapati, Shivam Yadav, Vijaypal Yadav; Journal of IEEE, June 2016

[2] *Smart Water Tank Management System for Residential Colonies Using Atmega128 Microcontroller*; Yogita Patil, Ramandeep Singh; International Journal of Scientific Engineering Research, Volume 5, Issue 6, ISSN 2229-5518, June 2014

[3] *IoT based Smart Water Tank with Android application.*;Priyen P. Shah, Anjali A. Patil, Subodh S. Ingleshwar; International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), December 2017

[4] *Water Tank Monitoring and Visualizing System Using Smart Phones*; Haesung.Tak, Daegeon.Kwon and Hwan.Gue.Cho; International Journal of Machine Learning and Computing, Vol. 3, No. 1E, February 2013

[5] *Smart Water Tank: an IoT based Android Application*; Prasanna Lakhsmi, Vasavi Mounika, Veda Sri, Pragna, Mr. K. Vikas; IRE Journals — Volume 1 Issue 9 — ISSN: 2456-8880, March 2018

[6] *Arduino Website* (`http://www.arduino.in`)

Achievements

1. Publications
   (a) *Water Quality Monitoring & Notification System using IOT*; Junaid Momin, Raheen Shaikh & Arkam Ansari; IJRAR, April 2019 (http://www.ijmems.in)
2. Conferences
   (a) *Water Quality Monitoring & Notification system using IOT*; Junaid Momin; International Conference on Efficacy of Software Tools for Mathematical Modelling (ICESTMM) ; April,2019 (Venue : Thadomal Sahani Engineering College, Bandra)
3. Project Competitions
   (a) *Water Quality Monitoring & Notification System using IoT*; Junaid momin, Raheen Shaikh, Arkam Ansari; 13-Inter-University Poster cum Presentation- Avishkar , December, 2018(Venue : RAIT, Nerul)
   (b) *Water Quality Monitoring & Notification System using IOT*; Junaid Momin; Bonhomie-Fuerza- Intercollegiate Technical Event; January,2019(Venue : AIKTC, New Panvel)

ANJUMAN - I - ISLAM'S

KALSEKAR TECHNICAL CAMPUS

PLOT #2 & 3, SEC -16, NEAR THANA NAKA, KHANDAGAON, NEW PANVEL, NAVI MUMBAI - 410206

CERTIFICATE OF PARTICIPATION

This certificate is presented to

Junaid Momin

For active participation in *Paper Presentation* event held during BONHOMIE 2019 FUERZA, an intercollegiate technical event held at AIKTC. Your involvement is highly appreciated.

PROF. SAAD SHAIKH
Technical Co-ordinator

PROF. TAIB QURAISHI
Controller, Students' Council

DR. ABDUL RAZAK HONNUTAGI
Director - AIKTC

Scanned by CamScanner

University of Mumbai

DEPARTMENT OF
STUDENTS' DEVELOPMENT

13th Inter-Collegiate/
Institute/Department

Avishkar

Research Convention: 2018-19

(District/Zonal Level Research Project Competition)

Certificate of Participation

This is to Certify that **Mr. Momin Junaid Dawood** of **Anjuman-I-Islam School of Engineering and Technology, New Panvel** Participated and Presented a Research Project Titled **"Water Quality Monitoring & Notification System using IoT"** in **Engineering and Technology** Category and **UG** Level at the **Selection Round** of **13th Inter-Collegiate / Institute / Department Avishkar Research Convention: 2018-19** held at Ramrao Adik Institute of Technology, Nerul, Navi Mumbai on **23 December, 2018** for **All Engineering Colleges of all Districts** zone.

DR. (MRS.) MINAKSHI GURAV
CONVENER
Avishkar Rearch Convention,
University of Mumbai

DR. SIDDHIVINAYAK BARVE
OSD
Avishkar Rearch Convention,
University of Mumbai

Dr. Sunil Patil
DIRECTOR
Department of Students' Development,
University of Mumbai

23 December, 2018
Nerul

DEPARTMENT OF
STUDENTS' DEVELOPMENT

13th Inter-Collegiate/
Institute/Department

Avishkar

Research Convention: 2018-19

(District/Zonal Level Research Project Competition)

Certificate of Participation

This is to Certify that **Ms. Shaikh Raheen N.** of **Anjuman-I-Islam School of Engineering and Technology, New Panvel** Participated and Presented a Research Project Titled **Water Quality Monitoring & Notification System using IoT** in **Engineering and Technology** Category and **UG** Level at the Selection Round of **13th Inter-Collegiate / Institute / Department Avishkar Research Convention: 2018-19** held at Ramrao Adik Institute of Technology, Nerul, Navi Mumbai on **23 December, 2018** for **All Engineering Colleges of all Districts zone.**

DR. (MRS.) MINAKSHI GURAV
CONVENER
Avishkar Reserch Convention,
University of Mumbai

DR. SIDDHIVINAYAK BARVE
OSD
Avishkar Reserch Convention,
University of Mumbai

Dr. Sunil Patil
DIRECTOR
Department of Students' Development,
University of Mumbai

23 December, 2018
Nerul

DEPARTMENT OF
STUDENTS' DEVELOPMENT

13th Inter-Collegiate/
Institute/Department

Avishkar

Research Convention: 2018-19

(District/Zonal Level Research Project Competition)

Certificate of Participation

This is to Certify that Mr. Ansari Arkam S. of Anjuman-I-Islam School of Engineering and Technology, New Panvel Participated and Presented a Research Project Titled **Water Quality Monitoring & Notification System using IoT** in Engineering and Technology Category and UG Level at the Selection Round of 13th Inter-Collegiate / Institute / Department Avishkar Research Convention: 2018-19 held at Ramrao Adik Institute of Technology, Nerul, Navi Mumbai on 23 December, 2018 for All Engineering Colleges of all Districts zone.

DR. (MRS.) MINAKSHI GURAV	DR. SIDDHIVINAYAK BARVE	Dr. Sunil Patil
CONVENER	OSD	DIRECTOR
Avishkar Rearch Convention,	Avishkar Rearch Convention,	Department of Students' Development,
University of Mumbai	University of Mumbai	University of Mumbai

23 December, 2018
Nerul