Smart Water Management for Domestic Use

Group Members

Hamza Mansoor Ali(01-131172-008)

Jahangeer Khan (01-131172-038)

Supervisor*:* Dr. Adeel M.Syed

A Final Year Project submitted to the Department of Software Engineering, Faculty of Engineering Sciences, Bahria University, Islamabad in the partial fulfillment for the award of degree in Bachelor of Software Engineering

July 2021

THESIS COMPLETION CERTIFICATE

Student Name: **Hamza Mansoor Ali** Enrolment No: **01-131172-008**

Student Name: **Jahangeer Khan** Enrolment No: **01-131172-038**

Programme of Study: **Bachelor of Software Engineering**

Project Title: **Smart Water Management for Domestic Use**

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at **10%** that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature: 

Date: **09-06-2021** Name: **Dr Adeel M.Syed**

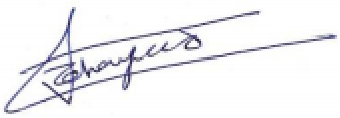
CERTIFICATE OF ORIGINALITY

This is certify that the intellectual contents of the project **Smart Water Management for Domestic Use** are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Name of the Student: **Hamza Mansoor Ali**

Signature:  Date: **09-06-2021**

Name of the Student: **Jahangeer Khan**

Signature:  Date: **09-06-2021**

Bahria University Islamabad
Department of Software Engineering

Dated: 17th June, 2021

CERTIFICATE

We accept the work contained in the report titled **Smart Water Management for Domestic Use** as a confirmation to the required standard for the partial fulfillment of the degree of BSE -8.

Engr. Sadaf Farhan
Project Coordinator
Date: 22/06/2021

Dr.Adeel M Syed
Supervisor
Date: 22/06/21

Dr.Tamim Ahmad Khan
Internal Examiner
Date:

Dr.Arslan Shaukat
External Examiner
Date: 17/6/21

Dr. Awais Majeed
Head of Department (SE)
Date: 24/6/2021

Abstract

We assessed the problem of water in the society and ways to conserve them, described a system to manage and control water for the domestic use. This study is based upon the project for water management Smart Water Management for domestic Use. Smart Water Management for domestic Use is management strategy of home automation. This system aids in the distribution and management of water in a society or flat system. Since this system is applicable in flats, hostels and many residential areas to conserve water, the methods to gather data and to manipulate are widely researched. The necessary research was done on how to check and manage water through ideal solutions and techniques. Smart Water Management for domestic Use uses an android application for the operation of management of water resource. We researched the existing strategies to identify water levels to perform necessary actions and provided an upgrade by connecting the devices with the android application to further perform operations. The water tanks are installed in a series in which water will be supplied sequentially as per requirement through a motor after processing through necessary checks. The flow will be disrupted after getting the required data from the sensors to turn on/off motor. The user will be notified to perform an action regarding the supplying of water to destinations and will be viewing in real time on the application. There was a need to conserve water, so a study was developed to provide a functionality of water sharing. It will allow the user to manage water within the tanks. This assisted in smart management of water with the fulfilment of the conservation purpose. The core achievement of this project was to design a system which fulfils the purpose of water supplying by replacing the manual work to an automated work. The purpose of water conservation and managing was promising. The integration of hardware with software was achieved and whole process is managed over a mobile application. Finally, we implemented necessary hardware in which devices are installed and which are operated through the android application on which options along with visualization is available.

Dedication

We would like to dedicate this project to our parents, siblings, friends and loved ones whose continuous support and encouragement led us to right direction.

Acknowledgments

All praise is to Almighty Allah who presented to us a moment part of His unfathomable learning by righteousness of which we had the capacity to achieve this difficult task. We are enormously obliged to our supervisor "Dr. Adeel M.Syed". Without his own supervision, counsel and important direction, this project would not have been successful. We are profoundly obliged to them for their support and constant help in this work. Furthermore, we are grateful to our folks and family who have been a steady source of consolation for us and taught to us the values of hard work and time management.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we will discuss about the project **Smart Water Management for Domestic Use.** It will describe for what purpose the project is developed along with its objectives. Firstly, we will discuss the problem for which the system is built and how the problem is resolved by the project. What motivation we got for making this project and the benefits which made the project more valuable for the society or world. The report of the project will not be comprehendible unless we organise it in a more understandable way, in a sequential way by figures defining the description at each phase. This is done for the reader to analyse the summary of the whole report.

1.1. Motivation

Currently, in most cases, the water is supplied to destination through the manual work by toggling the motor on/off and then supervising it to execute the next action. This involves time and labour of the person who is assigned to do this job. Moreover, manual work causes human errors and result in consequences such as water loss.

The motivation of this project is to substitute the human work to an automated work. Water is the biggest resource for every being on the planet and it should be conserved. In our daily life, water is being wasted and there are ways to reduce this loss but no physical implementation. To overcome this problem of water along with assisting the people, this project was built. The water supplying system in our homes, flats or any other domestic place requires proper timing along with supervision to avoid water loss. The negligence causes water loss, for which this system is developed, which will allow the user to make it more possible to reduce the manual work by an application to supply water to destination with the exception of prompt supervision of water situation.

1.2. Problem statement or research questions

The problem statement we identified for our system is stated below:

"*A perfect balance that meets the society water demands, without exceeding the unmanageable threshold of water supplies is crucial for the effective management of the resources. When a common water source is being shared among multiple users, then weak water-sharing systems regularly results in over-use and are typically associated with social, domestic and economic issues*"

In this era of 21st century, where climate change is a debate on every table. The scarcity of resources, whether it be power, water, agriculture, mineral or other natural resources, is of prime importance. To play our part, we have indulged ourselves in finding the common resource deprivation face by every inhabitant. As we analysed the water conservation in the domestic sector is not satisfactory, so we came up with a solution to overcome this issue and help the domestic sector conserve more water which is beneficial not only for them but for the environment as well. We built a project named Smart Water Management for Domestic Use which helps managing the water in supplying to the destinations along with substituting the manual work with an automated system. The purpose of the system is to supply water and supervising it over an application, which will be time-efficient and resource-friendly.

1.3. Objectives

Smart Water Management for Domestic Use will replace the manual work of humans to an automated work. It will allow the users to automate the water supply with the aid of an application which is operated through Wi-Fi. This serves the purpose of reducing the water-loss in addition to being time-effective.

1.3.1. Supply Water

One of the prime objectives of the system is to supply water to the required destinations. In most of the cases, people usually turn motor on/off by themselves and fulfils their purpose. To substitute this manual work, users can operate their mobile application to supply water for a specific time and it will assist in the water supplying management. The water is supplied through a motor which is connected to a series of tanks. The water will be supplied on following conditions:

- If the main tank has already enough water to be supplied
- If there is already sufficient water in the tanks, it will not supply water
- If the internet connection is not established, it will not supply water

1.3.2. Generates water supplying schedule

Another objective of this project is to generate schedule of water supplying which will allow the users to manage their water schedule according to their daily lifestyles. In most of the cases, people usually turn on the motor and due to some reason, the motor is not turned off. The water keeps on filling the tanks and result in water loss. The schedule will allow the users to set up a time and duration of the water supplying by the motor which is installed at the other end. The water will be supplied during the set time and will be turned off as soon as the duration ends. In other case, if the tanks are filled before the assigned time, the motor will be turned off automatically. It will be done by the ultrasonic sensors which are installed over the tanks.

1.3.3. Real-time visualization

Another objective of this system is to provide a real-time interface of the current states of the motor. This will allow the user to set a schedule or turn on/off the motor according to the need of user. The real-time visualization will assist the user to monitor the current situations which will be a prime objective in effective usage of water in the domestic use.

1.3.4. Water sharing among the tanks

One of the key aspects of this system is the concept of water sharing. The objective of this function is to manage the water among the tanks rather than demanding more water to be supplied. It will allow the water to be shared depending upon the statuses of the tanks. If there is less water in one tank and the adjacent tanks have sufficient water, then instead of turning on the motor again the water is being shared from the adjacent tank having more water. This will be done by the solenoid valve installed between the tanks which will be operated from the application. By this objective, the purpose of water conservation is fulfilled.

1.3.5. Establishing connection

One of the primary objectives is to establish a connection between the hardware and software. As this system has two sections, hardware and software. Hardware consists of the microcontrollers such as Arduino and NodeMCU which are further attached to the ultrasonic sensors for retrieving the information. The information is sent to the real-time database Firebase which stores the necessary information of current states of the hardware.

The firebase is a real-time cloud database which is established by Google. It is connected to the mobile application developed at the other end. The user can get all the necessary information and view it on the application to make the further decision.

1.3.6. Alert Generation

Alerts are generated accordingly for the user to be aware of the current states of water. Alerts are generated at every action taken by the user, for instance if the schedule is set it will prompt the user that the time is set for the motor to be turned on. In the same way when the duration completes, the user will be received a notification on the application that motor has been turned off. In some cases, where the main tank will be empty, the user will be made aware that It cannot supply water as the main tank has not enough water. It will also send alerts the user to set a time or turn on the motor as the water level in the tanks are low. The key objective of this function is to make aware the user to monitor the current water states and it helps in the water management of the particular area.

1.4. Main contributions

- Research have been done in the automation regarding the water distribution system, the project offers a platform where all the functions can be performed concurrently. Moreover, to provide a real-time supervision of the water state which aids the user for better monitoring. The new thing in this project is concept of water sharing. It allows the user to allow the flow of water from adjacent filled tanks rather than starting the supply entirely for one tank.
- The purpose of this project is water conservation and to reduce water-loss. By using this system, the users/flat managers will be able to schedule their water distribution which is time-effective and serves the purpose.

1.5. Report organisation

Chapter 1 discusses the introduction of the report in which we have discussed about the project. It reflects that what motivated us to create this project, what problem we analysed which formed the basis of the project. The need of the project which is defined by the objectives of the project which will be implemented. Mainly it covers the brief introduction of the whole project.

Chapter 2 discusses the current state of the work done related to the water management systems in automation. The work in chapter 2 reflects the foundations of smart water management in the automation systems. It also discusses various work done related to these and background of the already performed work. Moreover, literature review is also included in this chapter which formed the base to perform various work related to our project. It also explains various concepts, methodologies and tools related to implement the project. Finally the problem of integration of software with hardware is discussed in detail with an overview and critique of existing work.

Chapter 3 discusses the system requirements of our project. As our system is based upon some requirements on the basis of which it is implemented. It is best described by the use case diagram. In use case diagram, each functionality is described in detail in the description which shows the functionality of the system. Moreover, the functional requirements added indicates what the system actually performs. Functional requirements confirm the outcome of the whole system. These are followed by the non-functional requirements which describes the limitations to the project. For our project, security, reliability, consistency, performance and modifiability are included. The project is of no use if it is not feasible. Different types of feasibilities are also described in this chapter which confirms the implementation of the project. Finally, the analysis of the system is added which is best described by the state diagram which shows the state of the system at each phase. Furthermore, to elaborate the flow of the system, flow diagram is included to show the whole flow of the system.

Chapter 4 comprises of the system architecture. It discusses in detail about the design approach which is used for the system. Design approach cannot be described without the design constraints. Constraints makes easier for understanding of the design. System architecture is also discussed in detail such as there are two types of architectures which are used for the system. As being in the design section, it described the logical model of the system through the class and domain model in which the whole structure of the system is described. Furthermore, to show the dynamic view of the system, sequence diagram of each function of the system is described in detail. Followed by state diagram, where every state is described at each phase of the system. While describing the architecture, it is necessary to discuss the components and their interaction. As our system comprises of hardware and other entities as well, deployment diagram is used to demonstrate this.

Chapter 5 discusses the implementation of the system. The methods and means to apply the system to function it in the desired way. This comes under the strategy that how we have managed the implementation along with the major tools and techniques. Hardware and software tools are discussed which will be implemented with the coordination. The software and hardware are equipped in such a way that they follow some algorithms which are also discussed. Algorithms such as retrieving information and manipulation of such data is discussed. Since algorithms are used in such a way that they form a method so major methodologies of the system is also discussed. The system has vast methodologies which defines the working and performance of the overall system.

Chapter 6 consists of discussion related to the evaluation of the system. Evaluation process is carried out by performing the test cases. There are multiple techniques of testing and evaluating the system which are discussed in this chapter. In the first phase, unit testing is performed which is testing the system and its units individually before using it for the system. Then integration testing is performed which is the technique used to evaluate the integrated system. We used top-down strategy for the integration testing by testing the system as a whole and then hierarchically to the lower end components. Finally, the system is tested overall by System testing in which each module of the system is tested by generating test cases depending upon the requirement.

Chapter 7 consists the conclusion of the whole document in which we concluded our system and its shortcomings. Along with these, the effort and the reflection of each work is discussed. Concluding the system, it is necessary to define the future work which can be carried on lately. As we have made a flexible system which can be extended depending upon the requirement so further additions to our system is also discussed. It is necessary to mention the contributions and work performed by the members involved in the completion of the system which is also discussed in this chapter.

The structure of the report is illustrated in **figure 1.1.**

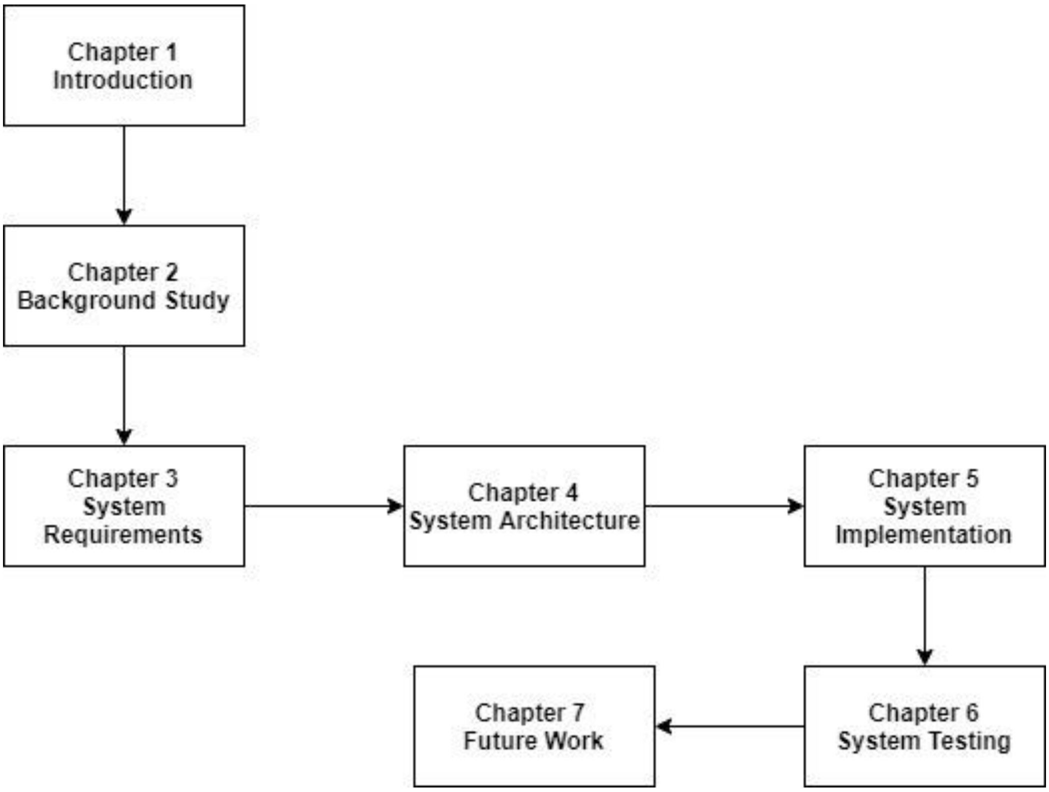


Figure 1.1: Thesis Organization

Chapter 2

Background Study/Literature Review

2.1. Organize Related Work

In the field of IoT based automation systems, there are many projects being developed every year as it is a vast area. Automation includes many fields from which we have selected the branch of home automation such as water management. Many projects have been made regarding smart water management using different techniques and in different areas such as agriculture, homes, and in designated areas.

Similarly, water management techniques are also used in home automation where the whole house is being operated smartly. Automation has been increasing every year due to its time-effectiveness and cost-effectiveness by replacing heavy machinery with some compact systems. Automation involves a range of technologies such as robotics, expert systems, wireless applications, test measurement and cybersecurity.[1]

In water management systems, different areas have been highlighted such as some researches have been made to determine the situation of water in the agricultural land, where some are made to control the overflow of water tanks. In a system where it was made possible to make a control unit which can monitor and control the overflow in tank. This system was connected to the mobile application via Bluetooth. The user sends the on/off control signal whenever an overflow is detected. Bluetooth module is also used which is used to send signals coming from Arduino. The purpose of the system is also to reduce the water loss as it is a major concern.

Most of the systems use many microcontrollers to receive the necessary information. In some systems, Raspberry Pi is used as a main microcontroller to be used for processing the information which is coded according to the need. Arduino is also used as a microcontroller to make the processes happen. For the transfer of information from the hardware to mobile applications, different technologies are used. For Bluetooth, GSM chip is used which is coded with Arduino or Raspberry Pi for the transfer of information. In Wi-Fi systems, NodeMCU, ESP8266 and ESP32 are mainly used. These are coded according to the requirements and hardware which is implemented.

Comparison of existing water automation systems are given as below:

Research Works	Hardware	Technology	Features	Lacking
Automated Water Billing with Detection and Control of Water Leakage using Flow Conservation [1]	GSM Module, Transceiver, Microcontroller	Zigbee, GSM	• Reduces the loss of water by detecting leakages. • Use of Automatic Meter Reading system to confirm water auditing • No need of human effort because of transceiver	No detail discussion related to mobile application interaction with the system
Automated Water Tank Overflow Control Unit Integrated with Mobile Application [2]	Arduino, Bluetooth Device, Water Pump, Sensor	Bluetooth	The system works for manual and automatic, people having no smart phones have to choose the automatic system only and for the manual monitoring operations a display board could be switched in the place of the mobile application	No detail discussion related to mobile application interaction with the system
Android Based Smart Water Pump Controller With Water Level Detection Technique[3]	Sensor, Transmitter, Router	Wi-Fi Network	It can be used to fill the water tank and in return it can prevent wastage of water	
Fully Automated System for Monitoring Water Usage Using SMS and Android Application[14]	Arduino, GSM Module, Ultrasonic Sensor, Water Flow Sensor, Solenoid Valve	Wi-Fi Network	• It can check water level • Check water usage • Cut-Off water supply	
Automation in Farming Using Android Application[5]	Arduino, Bluetooth Device, Sensor	Wi-Fi Network	• It can remotely on/off water pump with the help of bluetooth. • It can remotely on/off cultivation process • It can remotely on/off seeding process. • It can remind the farmer about their scheduled activity.	No detail discussion related to mobile application interaction with the system

Automated Irrigation System using Solar Power[6]	Microcontroller, Water Pump	Solar power system	• The model ensures the sufficient water level in paddy field avoiding the under-irrigation and over-irrigation. • Farmers can turnON/OFF the motor by using cell phone even far away • The system is protected with password for the specific number of users. • Solar power provides sufficient amount of power to drive the system.	There is no detail discussion about how the mobile application interacts with the system
Study of Arduino for Irrigation Based Control Using Android App[7]	GSM Module, Arduino, Sensor, Water pump	GSM, UMTS, Zigbee	• It can check electricity • Motor status can be monitored • Motor condition is monitored	

Table 2-1: Comparison of existing water automation systems

2.2. Key Concepts of System

The IoT based automation system aim is to use the control of operating the daily routine appliances or equipment at your ease and mobile. The main purpose is to control all the devices through internet protocols or through cloud computing. It allows ease of use and avoids complexity. The following figure illustrates the process of IoT based automation system which became the basis of our project:

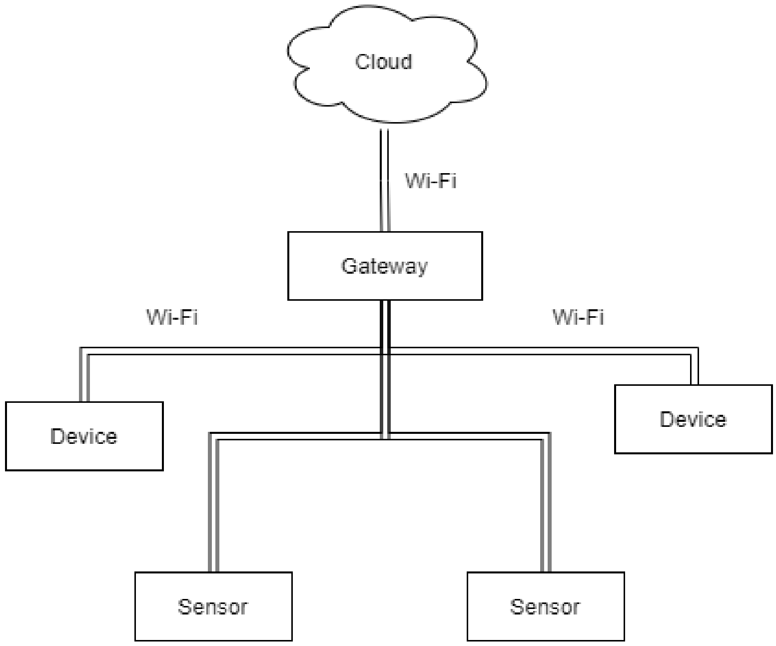


Figure 2.1: Cloud computing

The main concept of the system is to provide a cloud computing interface which can store the information which can be circulated to and fro from sensors to mobile devices.

2.2.1. Difference between internet protocols and cloud-based computing

Internet	Cloud
Internet is referred as global network of interconnected computers	Cloud computing is referred as the technology of delivering services or resources over the internet.
It allows software/hardware Infrastructure to maintain and establish connectivity	The resources which are transferred between software and hardware is done by cloud computing
Internet has protocols such as TCP/IP, UDP, FTP, HTTP, HTTPS, SMTP, POP	Cloud computing has three main models such as IaaS, PaaS and SaaS
Internet requires no subscription	Subscribers pay subscription fees for the transfer of resources
Internet can be used globally	Cloud cannot be used globally
It uses protocols as a transferring medium	It uses internet as a medium to transfer services

Table 2-2: Difference between internet protocols and cloud-based computing

2.3. Research Areas

Our research for project Smart Water Management for Domestic Use is mainly based upon IoT and automation which will give the motivation to develop our system. Moreover, these technologies include collaboration of micro-controlling units connected to the cloud via Wi-Fi in real-time. The research for units is for confining the best possible use for our system. Our research strategy will also include mobile application development to connect and retrieve the real-time data to perform further actions.

2.3.1. Real-time systems

Real time systems state that system is subjected to real time. This means the response which should be recorded or retrieved from any system should meet the specific deadline. There are many crucial systems developed based on real time responses. Most of the work which is done are subjected to provide accurate and prompt responses to take further action. The areas in which real time systems are useful are mostly monitoring or control based such as Air traffic control systems, real time monitors, real time systems in hospitals for medical applications, online gaming and much more. The purpose is acquiring the real time state or conditions of the desired results for further action.

2.3.2. IoT

Internet of Things or IoT is a system of network of physical objects. Things refers to computing devices, digital machines, unique identifiers or specific microcontrollers for controlling purposes and the interconnection and coordination between these components comes under the branch of IoT. IoT is a vast area in which various large-scale projects have been made. From management systems to real time systems, from reporting systems to home automation it has its own application.

IoT comprises of web-enabled smart devices installed and interconnected that use embedded systems. These systems include microprocessors, sensors, Wi-Fi or Bluetooth modules which are required to collect data from the environment. Theses IoT devices are directed to collect information from sensors to either IoT gateway which stores the states or to any cloud if it is used to retrieve data locally. Normally, most of the projects which are operate use cloud storages or real time cloud databases for which processing becomes more efficient by manipulating the data in real time. The flow of data becomes from sensors to the IoT devices, and from IoT device to any cloud storage from where it can be retrieved from any mobile application or web-based application.

2.3.3. Microcontroller units

A microcontroller is defined as a compact integrated circuit which is designed to perform a specific function in an embedded system. IoT project works with the aid of microcontrollers. Main microcontrollers which are used in the project for processing are

- Arduino
- Raspberry Pi
- NodeMCU
- GSM modem

Arduino

Arduino is one of the microcontroller boards based on ATmega328P. It is highly used in projects where processing of information is required at a particular time. It is coded with the Arduino IDE according to the requirements. There are many boards which can be used under the Arduino branch such as Arduino Uno, Arduino Mega etc depending upon the inputs and outputs of the devices attached to it.[8] In an agriculture sector in Indonesia, Arduino is used as a main microcontroller to get the information regarding soil and supplying water according to the requirement. The purpose of this is to implement the application of this microcontroller to be used as a control unit to manage the necessary action.

Rasberry Pi

Raspberry Pi is another form of microcontroller which can be used to process information by coding it. It is coded in Python language which helps a lot in making the processing more time efficient. It consists of a SD card which stores the current code installed into it.

NodeMCU

NodeMCU is another microcontroller used for the processing purposes. In most projects, it is used for serving the purpose of connecting the smart devices as it contains a Wi-Fi module known as ESP-8266. Projects using cloud storages or real time databases require readily connections over Wi-Fi so NodeMCU servers the purpose.

Some of the main projects in which these are used are home automation, security purposes to monitor home security and garage systems. Based on the functionality, in an agricultural project where it used to monitor the greenhouse effects by remote monitoring the situation of the plants.[9] These are monitored through an application which uses the data sent by the NodeMCU. The middle component is firebase which stores the real time data in the database for readily information on the application user.

Solenoid Valve

Solenoid valve is a smart valve which is used to control the water flow among the different equipment. It consists of a magnetic coil which electrifies on the current supply to change the state to on/off to make the water, gas or air through it. These are of different sizes with its operation environment of AC 220V or 12V depending upon the requirement. There are many applications of solenoid valve being used in different automated systems to control of water or gas as to control smartly by any device.

In a research based on the dynamics of vehicles where solenoid valve has its applications in the braking system.[10] The valve operates to produce a pressure on the fluid which is used for the braking of the vehicles. The braking process for vehicles is crucial and the response should be in milliseconds to allow the flow of fluid for brakes which is performed by the solenoid valve.

2.3.4. Automation

Automation is the coordination of different technologies to produce an output which can minimize human intervention. Automation serves the purpose by replacing the manual work of humans by automated process to carry out a specific task. It is a growing field and with the advancement of technology, more and more processes are shifting to automation. Automation has many types such as

- Space automation
- Home automation
- Self-driving vehicles
- Hospitality Events Processing
- IVR (Interactive voice response systems)
- Smart Home notifications
- Self-healing

2.3.5. Android Application Development

Android SDK

Android SDK is a development kit issued by the Google for the implementation of android programming. This SDK allows to program easily by using specific files. In android studio, it comes bundled with it. It is actually a collection of libraries and tools which are used to develop the android applications. There are multiple versions released timely by the Google for its implementation for the programmers. The SDK version for our project was selected according to the requirement. It comes with packages to be installed automatically by android studio or can be imported manually by downloading it.

Java JDK

It is an environment provided to develop and make the android applications. Java Development Kit comes under the package including JRE, interpreter/loader, archiver, compiler, documentation generator and the related tools for the development of applications.

Services

Services are those components of android which are running in the background even if the application is destroyed. It runs in the background to perform long running operations. A service has many states which can be operated depending upon the requirement. It mainly has two states:

Started: Service which is started when component of application, such as activity, start initially by starting itself. If it is started, it will run and cannot be destroyed even if the application is destroyed.

Bound: Usually a service is bound when application component binds it with operation by calling **bindService().** It allows an interface which allows a client-service interface that allows to get and send requests, retrieve results and even across processes with inter-process communication.[12]

Broadcast Receivers

Android application can send or receive broadcast receivers depending upon the event occurred at any instance. Usually these broadcasts are sent when a specific event occurs. These events are such as system booting up, mobile or laptop which are charging. There is also a category of it in which custom broadcasts can be sent which notifies the other application.[11]

Fragments

A fragment represents usable portion of the application UI. It defines its own layout and has its lifecycle. There may be multiple fragments which are used in the application which are placed in a hierarchical order. View fragment which is visible for the user at that moment comes at the top when called and the others respectively. A key aspect of the fragments is modularity. In this, the UI is divided into different chunks depending upon the requirement of the user. It allows to use the activities in the same plane by arranging them into different discrete sections.

SQLite

As to include the database, comes the SQLite in which data can be stored. It is an open-source database which can store data to a text file on the device being used. It comes built in with the android applications development. Key feature of SQLite is that it supports all kinds of relational databases, there is no need to add special libraries for its connection.

2.4. Project Background

In our daily life, every house, domestic flats, hostels or any residential sector requires water for their necessary purposes. The conventional way of serving this purpose is by turning on the motor for supplying and after a time it is turned off. In between it is monitored by peeking into the tanks or after water starts overflowing from the tanks. This causes too much human effort to manually perform these actions.

Moreover, sometimes the water supply is left unattended and the water keeps overflowing from the tanks which results in water loss. To overcome this water loss situation and to substitute the human effort with an automated system, this project was proposed.

The main idea and motivation of this project is to replace the manual work with an automated application and keeping in view the current global problem which is water loss so that water can be monitored and managed. To replace the human manual effort, an application is developed which allows the user to schedule a time of water supply with readily available notification keeping the user aware of the current states of water in the tanks. For domestic usage, there are multiple tanks installed and it will be easier to monitor the water states and then decide to set up a time for supply. The control of this application will be over to the caretaker of the residence and will be managed accordingly.

Another functionality which can increase the effectiveness in managing the water is water sharing in which water is shared among the tanks installed depending upon the states of the adjacent tanks. Instead of turning on the supply for one tank, water is distributed among the tanks.

2.5. Strong and Weak Points

There might be no such global limitations in the project but there are few other constraints which include the application which is only operated on Android phones. As such applications require no framework to be implemented on to make a cross platform application due to its less complexity. Otherwise, the application works perfectly on android devices. The hardware constraints include some equipment such as if the tanks used have irregular surface, then the sensors installed at each tank might give some dubious information at a particular point. So before installing any tank with irregular surface, there is a rare chance of any ambiguity because it basically gets the information from the water surface. Moreover, the water should be in a closed surface so that the information is retrieved in a perfect manner as shown in the figure 2.2.

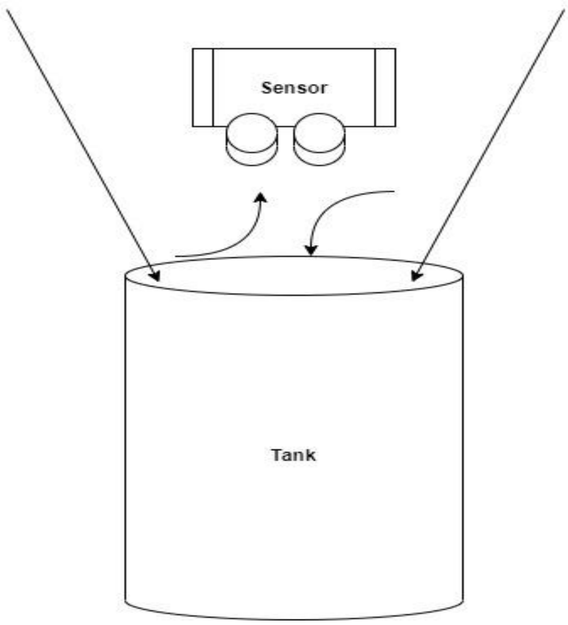


Figure 2.2: Information Retrieval

The data on the firebase which is uploaded by the NodeMCU is connected over Wi-Fi. When the sensors pass the information to the microcontrollers to be sent over the cloud, there is a 1-2 second delay in uploading the information. In the similar way, if the motor state is On/Off by the application, there will be a 1-2 minute delay due to the Wi-Fi connectivity however this may differ depending upon the Wi-Fi connection strength.

One of the plus points of this system is that application can be operated from any mobile location. It is because the application which is connected to the Wi-Fi of the region sends or retrieve information to and from firebase. In the same way, the system which is applied at the location is also connected to the local Wi-Fi which operates the firebase database as illustrated below in figure 2.3:

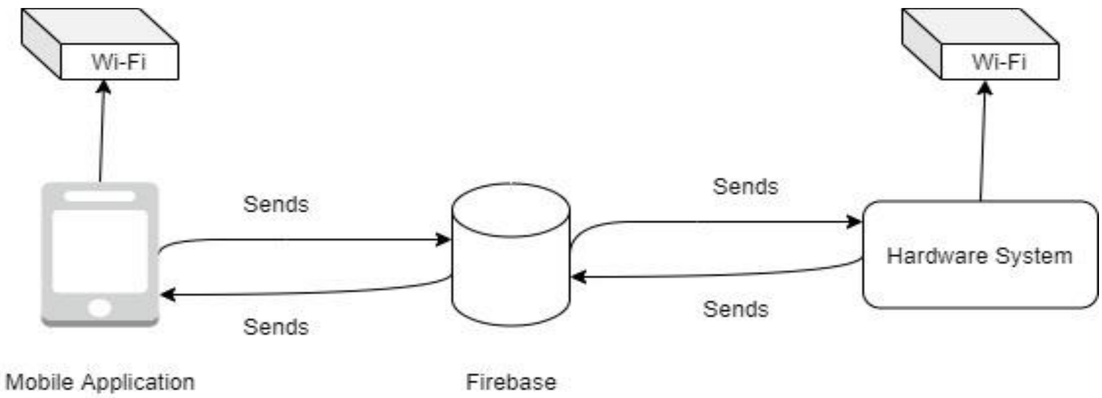


Figure 2.3: Firebase communication

Chapter 3
System Requirements

3.1. Use Case Diagram

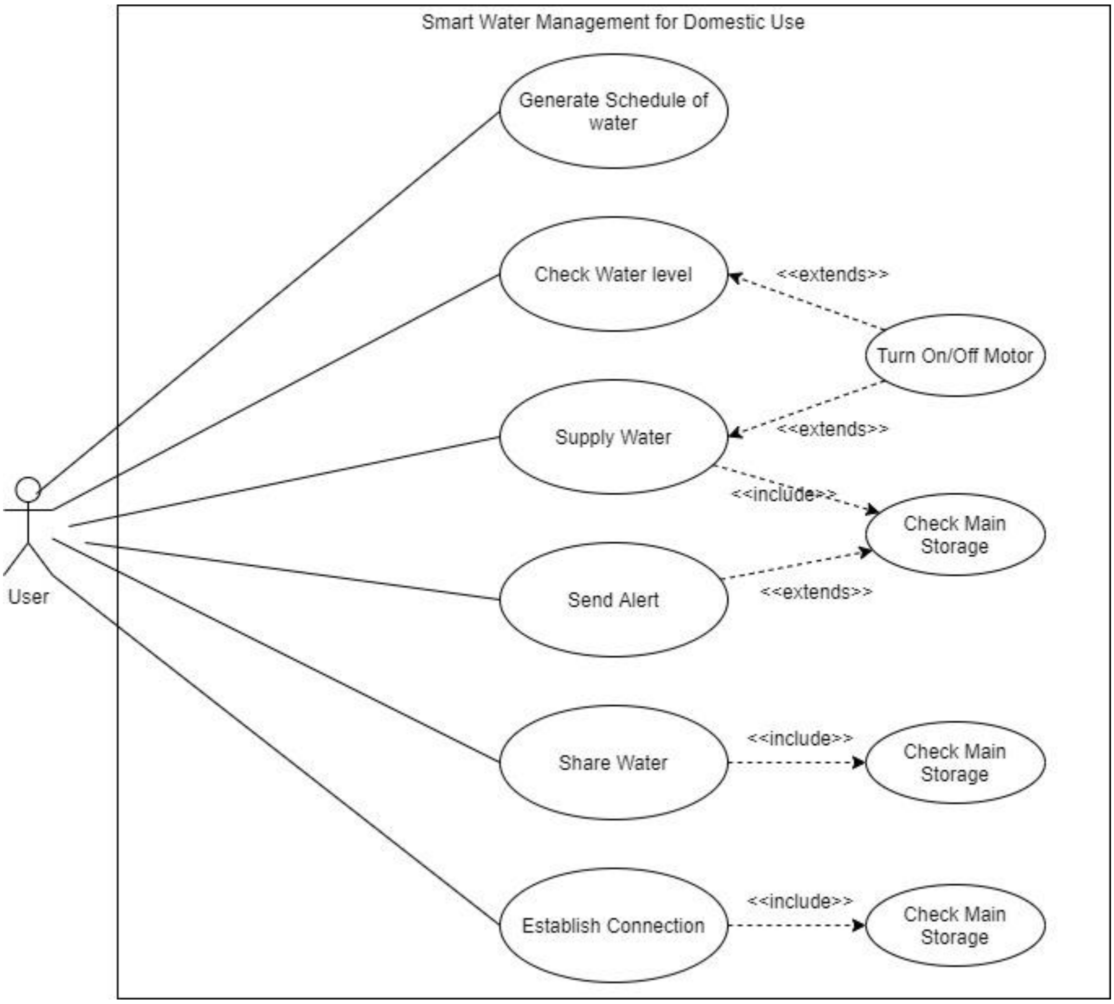


Figure 3.1: Use case diagram

3.1.1. Use case #1: Generate Schedule of Motor

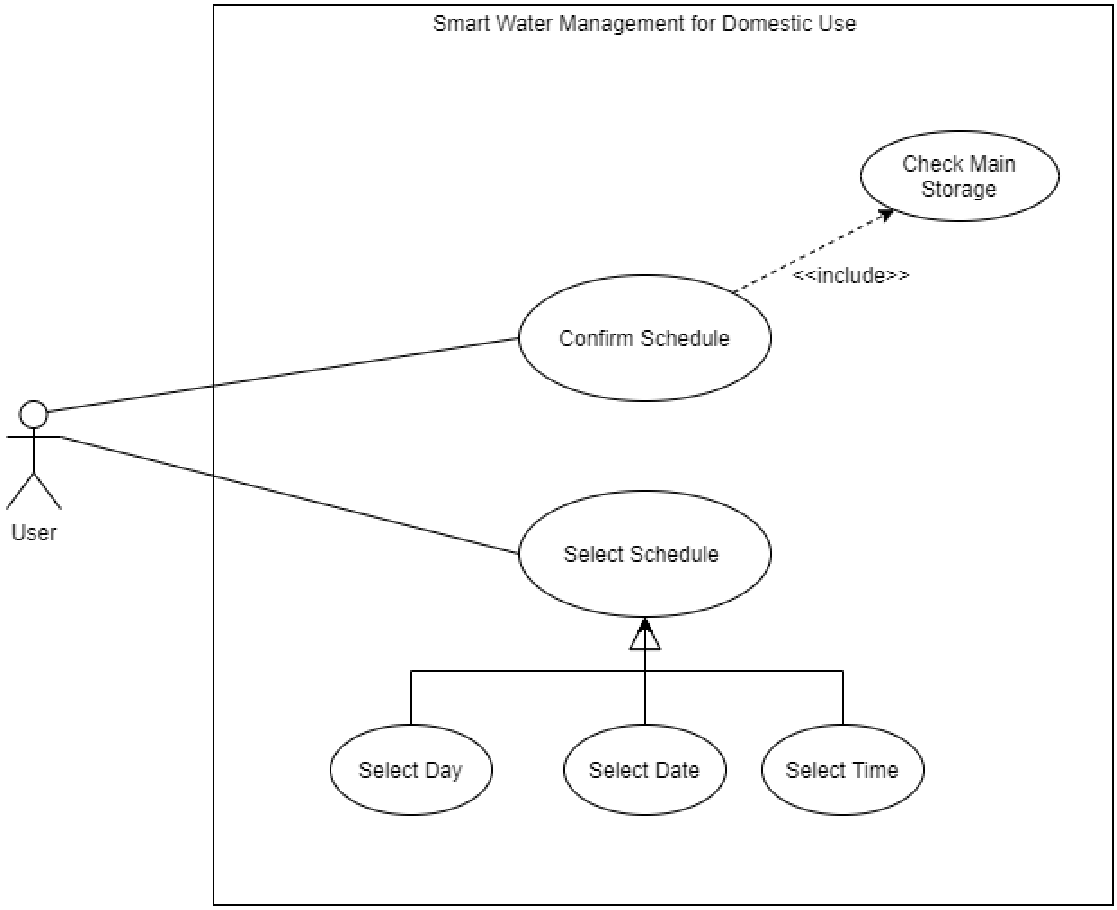


Figure 3.2: Use case 1 - Generate Schedule of Motor

Use case description:

Use Case ID:	UC-1
Use Case Name:	Generate Schedule of Motor
Actor(s):	User
Pre-Conditions:	• Login to the application • Connection with Wi-Fi • Connection with smart devices
Priority:	High
Basic Flow:	The user accesses the application and after connecting to Wi-Fi and the devices, sets a schedule of water supply
Actor Actions	**System Response**
1 User selects the generate schedule of motor from the menu	**2** System displays the scheduling screen with options of timing
3 User selects the respective timings and duration according to need	**4** System displays options for timings

5	User selects "Done" button for confirmation	**6** System adds the schedule of motor in the application
Alternative Course of Action (if any)		
Actor Action		**System Response**
	❖ 3a. Invalid timings selected ❖ 5a. No schedule selected	❖ 2a. No schedule found 2a1- System displays no schedule found and user generates a schedule by selecting add a schedule ❖ 3a1- System gives an error to select valid timing ❖ 5a1- System checks the timings and then ask the user to add some timing for generating a schedule

3.1.2. Use case #2: Check Water Level

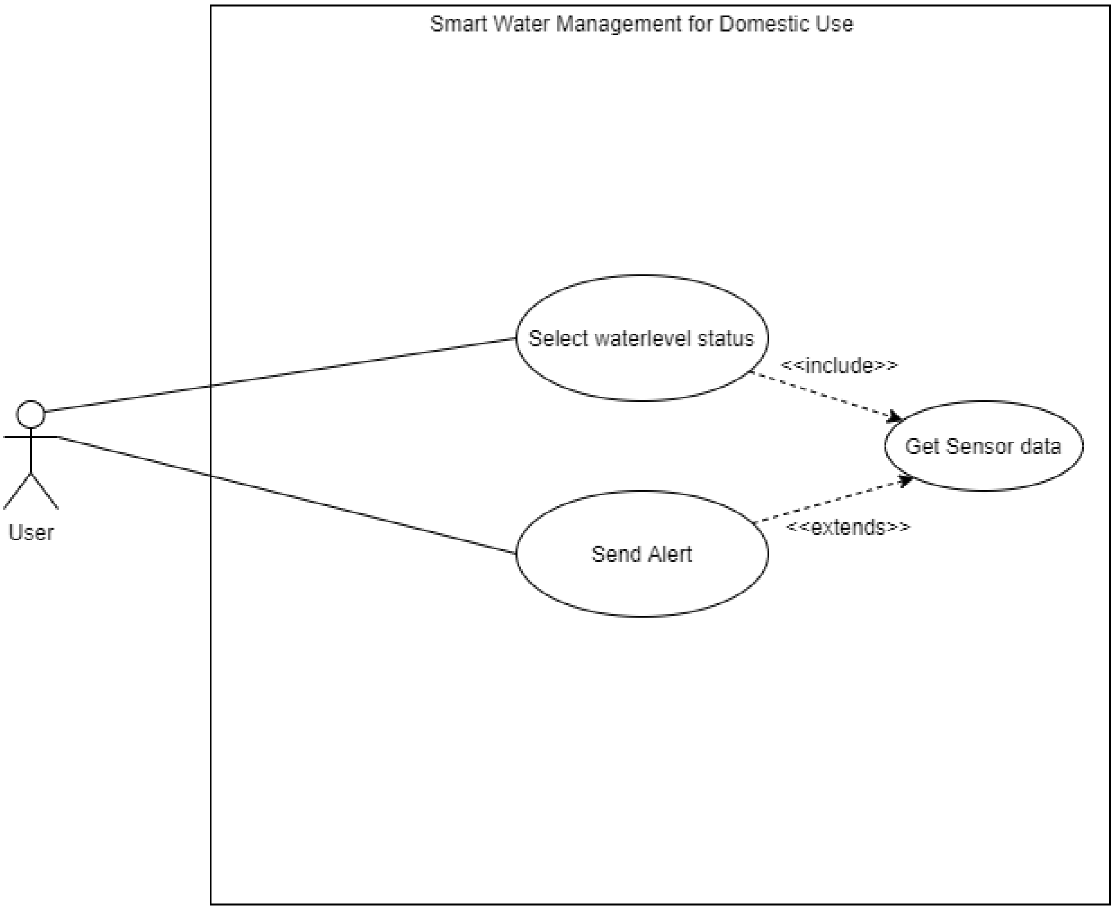


Figure 3.3: Use case 2 - Check Water Level

Use case description:

Use Case ID:	UC-2
Use Case Name:	Check Water Level
Actor(s):	User
Pre-Conditions:	• Login to the application • Connection with Wi-Fi • Connection with smart devices
Priority:	High
Basic Flow:	The system checks the water level in the tanks and generate an alert and automatically turn on/off motor
Actor Actions	**System Response**
1 Motor is turned on manually or through schedule	**2** System checks the water levels in tanks through sensors

3	Water levels are high in the tanks	**4** System generates an alert and turn off the motor automatically
5	Water levels are low in the tanks	**6** System generates an alert and ask user to turn on the motor
6	Motor is already in running state	**7** Checks water levels and generates alert and turn motor off when level gets high
Alternative Course of Action (if any)		
Actor Action		**System Response**
		❖ 6a- Motor is not turned on 6a1- System checks the electricity supply and notifies the user ❖ 7a- Sensor gives wrong water level data 7a1- System checks the sensor data, verifies it and then turn on/off motor and send alert to the user 7a2- System checks the timings and then ask the user to add some timing for generating a schedule

3.1.3. Use case #3: Turn On/Off motor

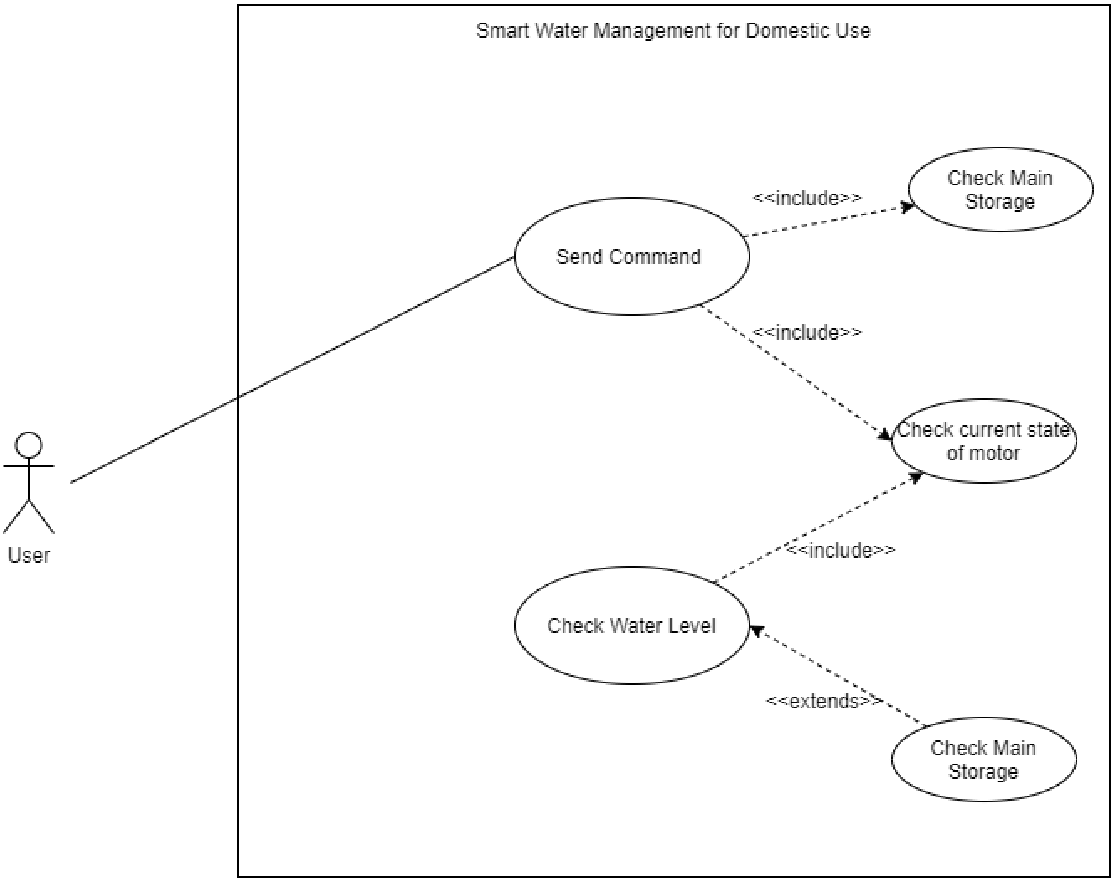


Figure 3.4: Use case 3: Turn On/Off motor

Use case description:

Use Case ID:	UC-3
Use Case Name:	Turn On/Off motor
Actor(s):	User/System
Pre-Conditions:	• Application send a command for supplying of water • Application send a command to turn ON motor
Priority:	High
Basic Flow:	On sending the command of supply water or turning ON the motor, the motor is turned ON and after getting filled it turns OFF

Actor Actions		**System Response**
1	User selects turn ON motor	**2** System displays the turn ON motor screen
3	User selects the confirmation	**4** System sends signal to motor to turn ON and on filling, turns OFF

Alternative Course of Action (if any)	
Actor Action	**System Response**
	❖ 4a-System checks and if water is low then sends an alert to user ❖ 4b-Signal not sent. System asks user to try again

3.1.4. Use case #4: Supply Water

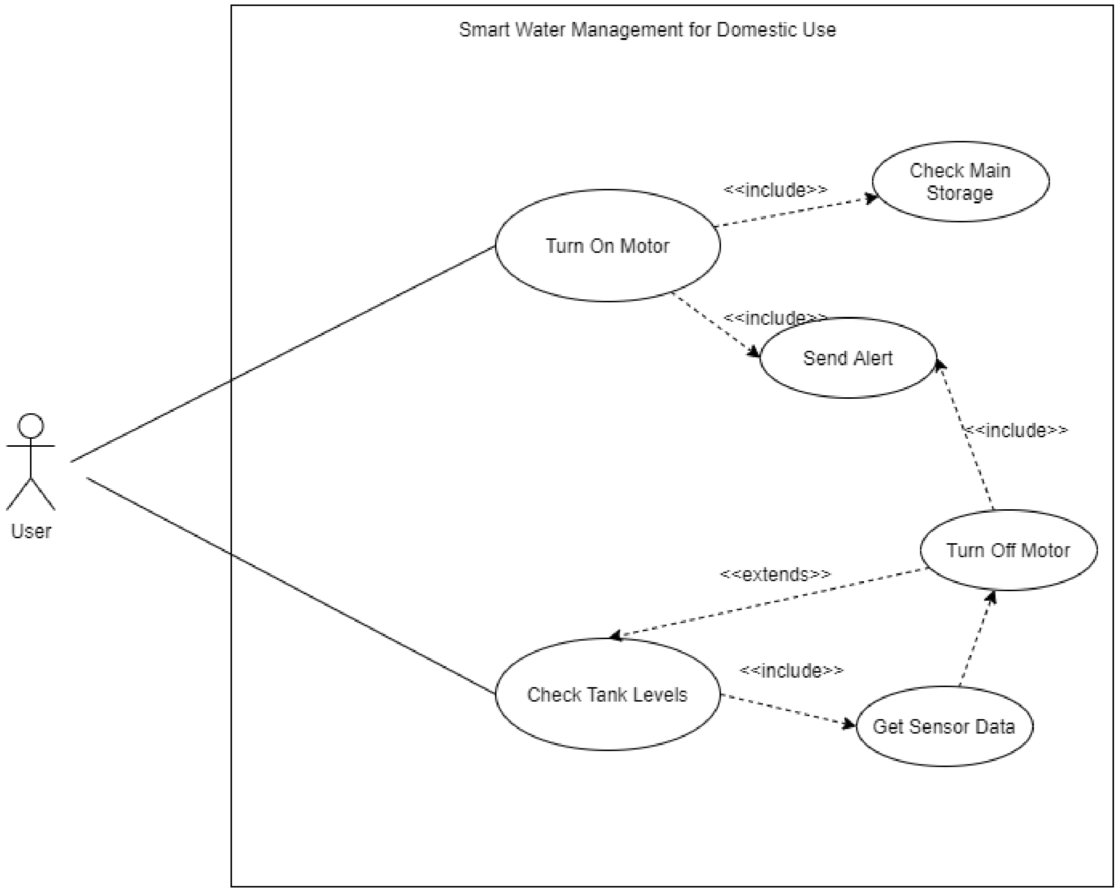


Figure 3.5: Use case 4 - Supply Water

Use case description:

Use Case ID:	UC-4
Use Case Name:	Supply Water
Actor(s):	User
Pre-Conditions:	• Login to the application • Connection with Wi-Fi • Connection with smart devices
Priority:	High
Basic Flow:	The user manually turns on the motor
Actor Actions	**System Response**
1 User selects the supply water from the menu	**2** System displays the options to user
3 User selects the supply water button provided for confirmation	**4** System turn on the motor and water is supplied

Alternative Course of Action (if any)		
Actor Action		**System Response**
	❖ 2 On selecting supply water button	❖ 2a- No water present in the underground main tank 2a1- System checks the main tank and gives alert of “No water present” in the main tank ❖ 2b- The motor is already running 2b1- System checks the motor state and gives alert of already “Motor is On” ❖ 2c- Water level is already high in the tanks 2c1- The system checks the water level condition in the tanks and notify the user and motor is not turned on ❖ 4a- Electricity supply turned off 4a1- System sends an error to the user using application and turn off the motor

3.1.5. Use case #5: Check main storage

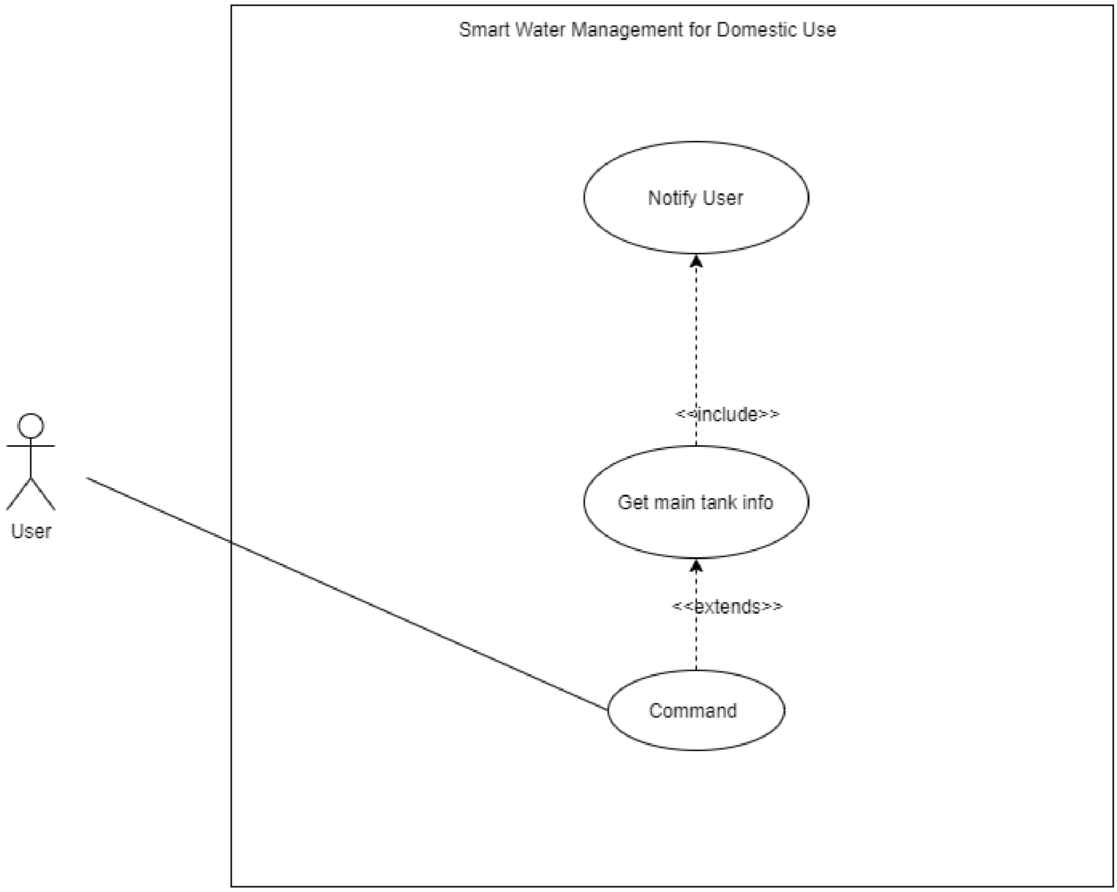


Figure 3.6: Use case 5 - Check main storage

Use case description:

Use Case ID:	UC-5
Use Case Name:	Check main storage
Actor(s):	User
Pre-Conditions:	• Application send a command for supplying of water • Application send a command to turn ON motor
Priority:	High
Basic Flow:	On sending the command of supply water or turning ON the motor, the main storage is checked first then it allows it to flow water

Actor Actions		**System Response**
1	User selects the supply water	**2** System displays the supply water screen

3	User selects the confirmation	**4** System checks the main storage and allows flow of water by turning ON motor
Alternative Course of Action (if any)		
Actor Action		**System Response**
		❖ 4a-System checks and if water is low then sends an alert to user

3.1.6. Use case #6: Share Water

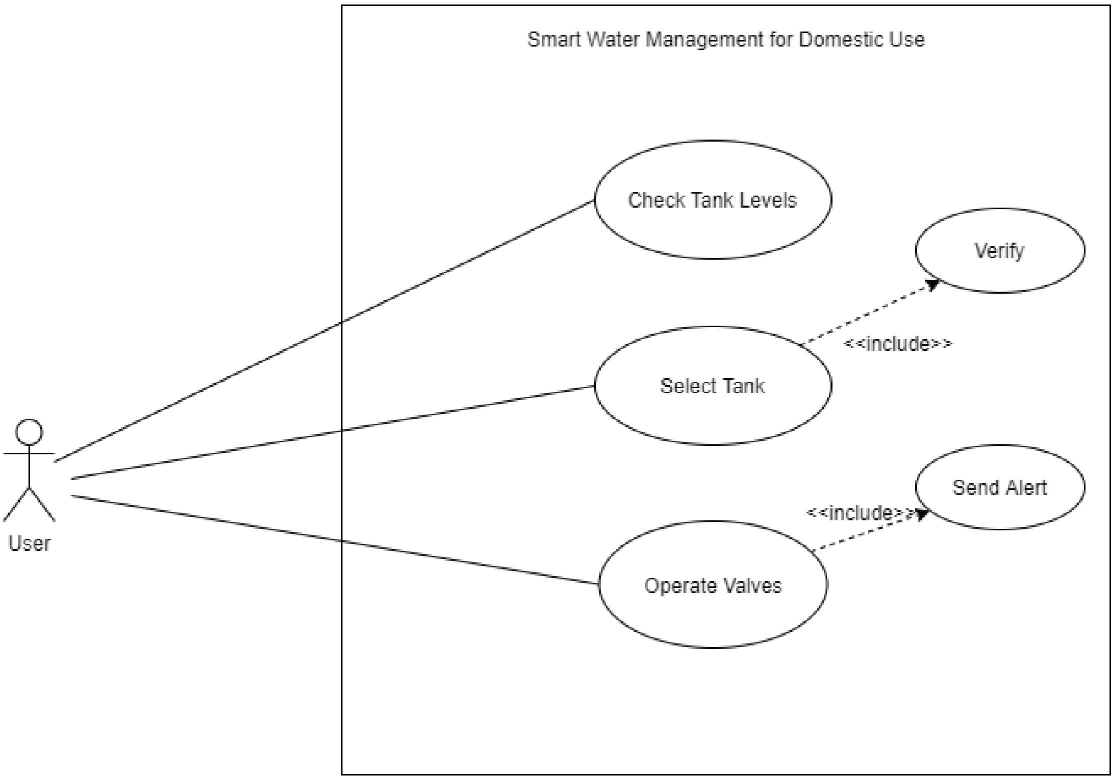


Figure 3.7: Use case 6 - Share Water

Use case description:

Use Case ID:	UC-6	
Use Case Name:	Share Water	
Actor(s):	User	
Pre-Conditions:	• Login to the application • Connection with Wi-Fi • Connection with smart devices • Schedule is generated • Tanks are supplying water accordingly • Alert is generated regarding any low level water	
Priority:	Medium	
Basic Flow:	The system will check the adjacent tanks and will allow sharing of water among the water tanks	
Actor Actions		**System Response**
1	User selects the share water button from the menu	**2** System displays the screen for share water

3	User checks the tank situations and allow sharing of water from the desired tank to the required tank	**4** System displays confirmation and shares water
Alternative Course of Action (if any)		
Actor Action		**System Response**
		❖ 3a- Low level of water in the adjacent tanks 3a1- System checks the adjacent tanks and then send alert regarding the supply of water

3.1.7. Use case #7: Operate Valves

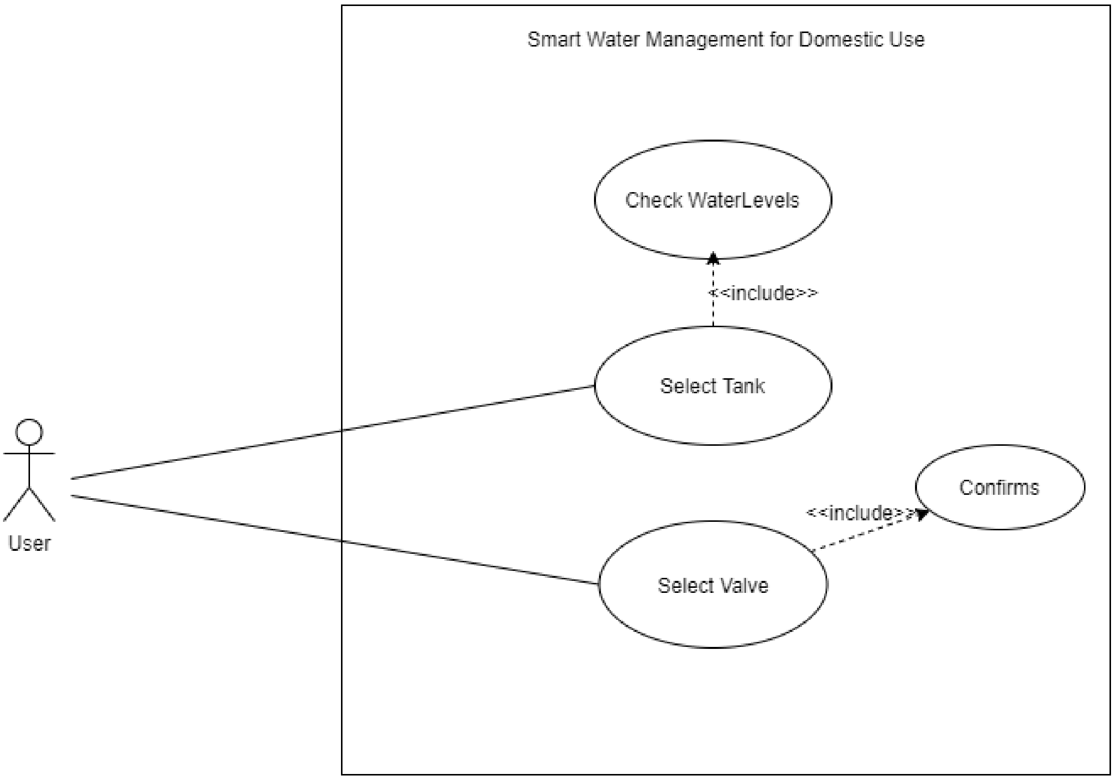


Figure 3.8: Use case 7 - Operate Valves

Use case description:

Use Case ID:	UC-7
Use Case Name:	Operate Valves
Actor(s):	User
Pre-Conditions:	• Login to the application • Connection with Wi-Fi • Connection with smart devices • Schedule is generated • Tanks are supplying water accordingly • Alert is generated regarding any low level water
Priority:	Medium
Basic Flow:	This will trigger the sensors on the valves and allow it to operate during the water sharing

Actor Actions		**System Response**
1	User selects the share water button from the menu	**2** System displays the screen for share water along with visualization of tanks
3	User checks the tank with more water and select it for sharing of water	**4** System asks for operation of required valves

5	User selects the valves and grants permission	**6** System gives confirmations and carries out process
Alternative Course of Action (if any)		
Actor Action		**System Response**
		❖ 2a- System shows error while displaying visual representation 2a1- System generates an alert regarding error and refreshes the current states ❖ 4a- System did not ask for the operation of valves 4a1- System gives an error, refreshes and again show the option of operate valves ❖ 5a- System does not grant permission for operation of valves 5a1- System gives an error and gives a busy state and then repeats extension 4 ❖ 5b- System gives error of wrong tank selected for sharing on clicking the grant permission 5b1- System checks the tank selected and notifies user to select the other tank with appropriate quantity of water

3.1.8. Use case #8: Establish connection

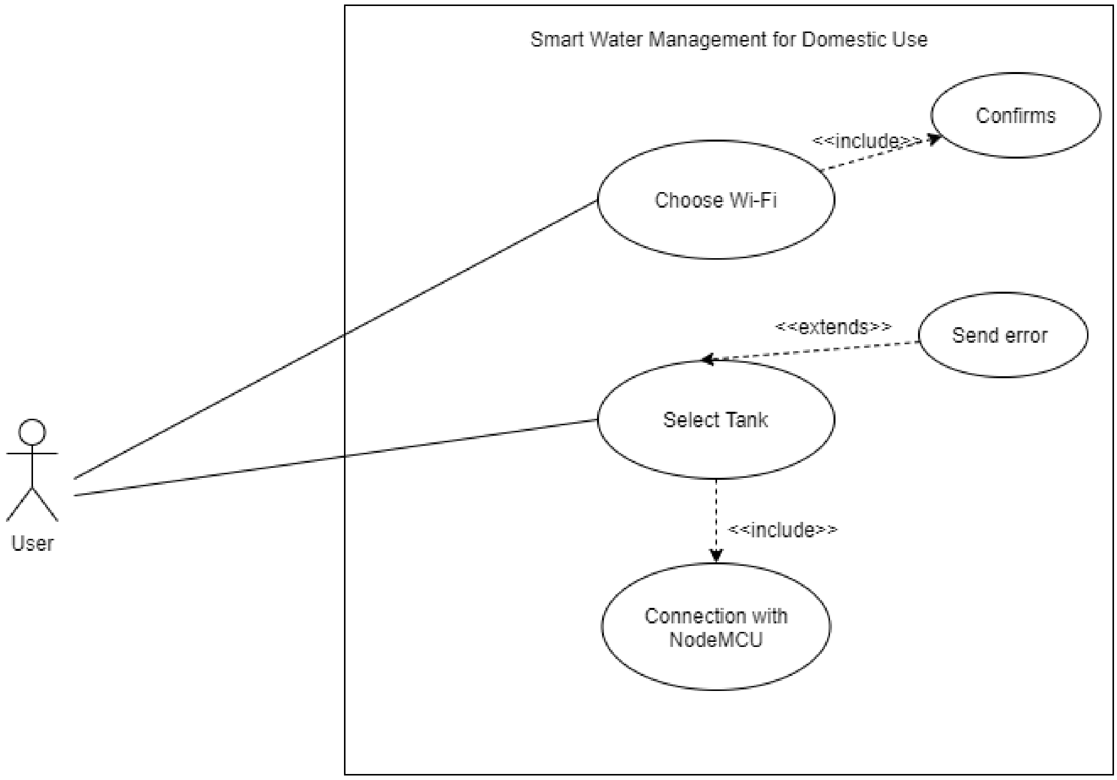


Figure 3.9: Use case 8 - Establish connection

Use case description:

Use Case ID:	UC-8	
Use Case Name:	Establish Connection	
Actor(s):	User	
Pre-Conditions:	Application should be logged in and it should be running	
Priority:	High	
Basic Flow:	On turning the application ON, the application connects and configures the smart devices connected with the applications	
Actor Actions		**System Response**
1	User opens the application	**2** System displays the login screen
3	User logs in the application	**4** System asks for id/password
5	User select the Wi-Fi connection	**6** System establishes connection with database and devices
Alternative Course of Action (if any)		
Actor Action		**System Response**

		❖ 3a- Login not successful 3a1- System asks the user to sign-up or login with credentials ❖ 5a- Unable to connect with Wi-Fi 5a1- System gives the other Wi-Fi devices

3.2. Functional Requirements

3.2.1. Generate Schedule of motor

Identifier	FR-01
Title	Generate Schedule of Motor
Requirement	The admin will set a schedule to turn on the motor for water supply
Source	supervisor
Rationale	The selected schedule by the user will be added to the list which will turn the water supply by turning on the motor
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.2.2. Check Water Level

Identifier	FR-02
Title	**Check water level**
Requirement	The system shall be able to check the data coming from the sensor to notify user
Source	supervisor
Rationale	To collect the data coming from the sensors about the water tanks and then notifying user and system for planning
Restrictions and Risk	N/A
Dependencies	FR-01
Priority	High

3.2.3. Supply Water

Identifier	FR-03
Title	Supply Water
Requirement	The system shall be able to turn on the motor after going through necessary checks to supply water
Source	supervisor
Rationale	To confirm the instruction given by the user and then turning on the motor for supplying water
Restrictions and Risk	N/A
Dependencies	FR-02
Priority	High

3.2.4. Share Water

Identifier	FR-04
Title	Share Water
Requirement	The system shall be able to operate the valves which allow the flow of water between the tanks
Source	supervisor
Rationale	To send command to operate valve after suitable checking of water condition
Restrictions and Risk	N/A
Dependencies	FR-02
Priority	High

3.2.5. Login and Registration

Identifier	FR-05
Title	Login and Registration
Requirement	The system shall be able to login the user to use the application or display the sign up page if the user is new
Source	supervisor
Rationale	To authenticate credentials and storing the user information in case of new user
Restrictions and Risk	N/A
Dependencies	N/A

Priority	Medium

3.2.6. Establish Connection

Identifier	FR-06
Title	Establish connection
Requirement	The system shall be able to connect to firebase by Wi-Fi.
Source	supervisor
Rationale	To check the Wi-Fi at both ends and establish a connection for command transfer
Restrictions and Risk	N/A
Dependencies	N/A
Priority	High

3.2.7. Operate Valves

Identifier	FR-07
Title	**Operate Valves**
Requirement	The system shall be able to turn on or off the valve depending upon the water condition of the tanks involved
Source	supervisor
Rationale	To operate valve and allow flow of water
Restrictions and Risk	N/A
Dependencies	FR-04
Priority	High

3.2.8. Turn On/Off motor

Identifier	FR-08
Title	**Turn On/Off Motor**
Requirement	The system shall be able to turn on or off motor depending upon the condition of tanks and schedule
Source	Supervisor
Rationale	To supply or distort the voltage for operating the motor on command from the application

Restrictions and Risk	N/A
Dependencies	FR-03
Priority	High

3.2.9. Check main storage

Identifier	FR-09
Title	**Check Main Storage**
Requirement	The system shall be able to check the main storage before turning on the motor
Source	Supervisor
Rationale	To collect and send sensor data of the main tank to notify system and user
Restrictions and Risk	N/A
Dependencies	FR-02
Priority	High

3.3. Interface Requirements

Interface requirements are the dependencies of the system with which it is interacting. The system needs to communicate with the outside world with the help of interfaces. The requirements of the system which it needs to fulfil in order to run the system are of three types. User Interface, Hardware and Software which are defined as follow.

3.3.1. User Interface

User interface of the application consists of many screen depending upon the options provided for the user. On launching, there will be a loading screen to let user know the launching process of application is underway. Then there will be a user login page for user to get login. If the user is using for the first time, it will display a registration page for the user to get registered.

3.3.1.1. Login/Registration Page

After logging in, the user is navigated to the home page. Home page consists of different choices available for the user to select. The options are actual buttons which will navigate the user to the respective page i.e. Generate Schedule, Supply water, check water levels etc. For keeping the user logged in and to check account. The user can see a side navigation bar from where the current user is shown. From there, user can logout or update the profile i.e. email and password.

3.3.1.2. Update Profile

On clicking update profile, the user will be navigated to a screen with text fields for the user to enter the updated email and password for updating. Once entered, a confirmation will be given that the credentials are updated and verified.

3.3.1.3. Generate Schedule

In the generate schedule screen, a list of already set schedule is displayed with their information. On clicking add new button, another screen opens which allows user to select day, date and duration for the motor.

3.3.1.4. Supply Water

On supplying motor screen, the user is shown the current states of the water tanks along with the toggle to turn on or off the motor.

3.3.1.5. Check Water Level

Check water levels is another screen where the complete view of the real time situations of the tanks is shown. The three tanks are shown and their current water level statuses are visible at each rate.

3.3.1.6. Share Water

On clicking Share water, a screen is opened with the tanks. From where user can select the respective tank from which water will be shared and to the tank where the water will be transferred. There will also be a toggle to operate valves.

3.3.1.7. Establish Connection

Clicking establish connection will display the current Wi-Fi signals available. User will select and then the application will be configured.

3.3.2. Hardware Interfaces

Hardware will include many components and smart devices which will be connected and related with each other to produce a result. Main hardware interfaces are included as below:

3.3.2.1. Arduino

Arduino will be used as a microcontroller to retrieve all the information and to transfer the specific instruction. It has input pins and output pins, along with these ground and Vcc pins are also present to be supplied to the sensors attached at the outputs.

Figure 3.10: Arduino [15]

Arduino is attached with other components such as NodeMCU, water sensors, relay module and related LED's which have to be placed precisely because every pin on which any device is attached is defined and coded. The pin configuration of Arduino UNO board is as follows:

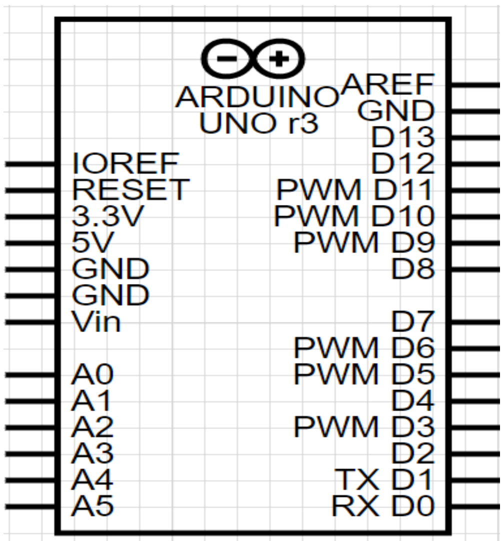


Figure 3.11: Arduino Pin Configuration [16]

The Output pins from D0 to D13 are used as the output pins for the water sensors with the exception of D3 and D5. These are used as TX and RX pins for serial communication with NodeMCU which is the Wi-Fi module. The 5V and GND are supplied for the sensors and NodeMCU respectively.

3.3.2.2. NodeMCU

NodeMCU is also a microcontroller which can be used to process different tasks. It is used as a Wi-Fi module to be connected to firebase and transfer the information coming from Arduino and relay module. It also has set of input and output pins. The relay module is operated from NodeMCU when it receives instruction from firebase.

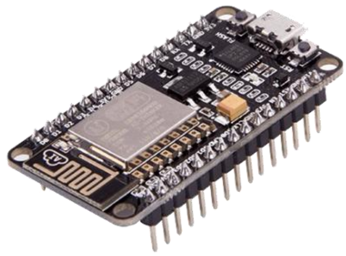


Figure 3.12: NodeMCU [17]

The pin configuration is as follows:

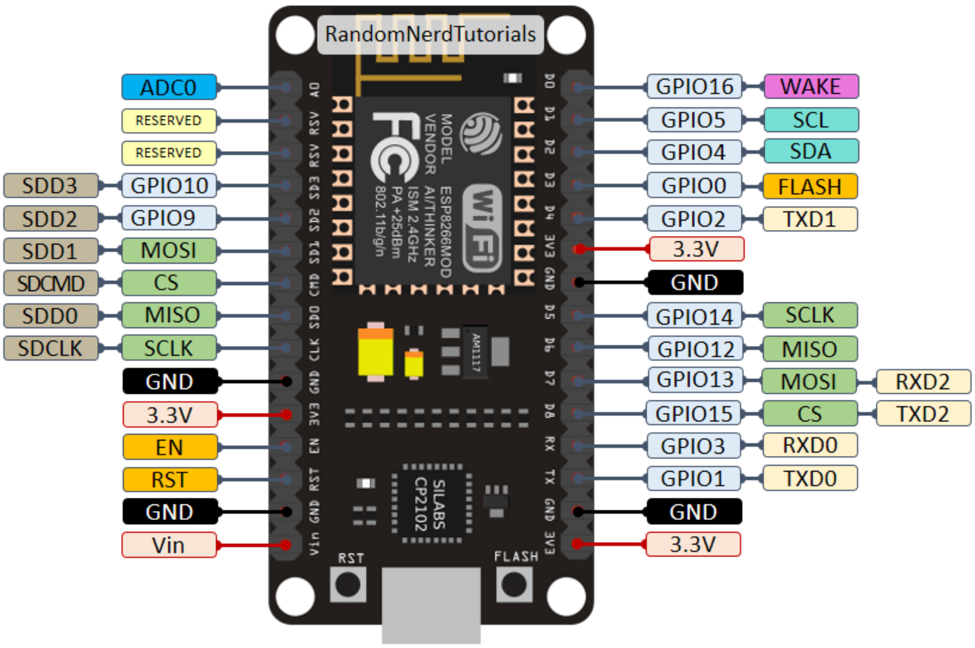


Figure 3.13: NodeMCU Pin Configuration [18]

3.3.2.3. Ultrasonic water level sensors

The water sensors which are used to detect the water level in the tanks. It is connected with Arduino which retrieves the information. These are placed above the water tank level to get the current condition of water. The Trig and Echo pins are used to send and receive the sound waves which are stroke on the water surface to get the required information.

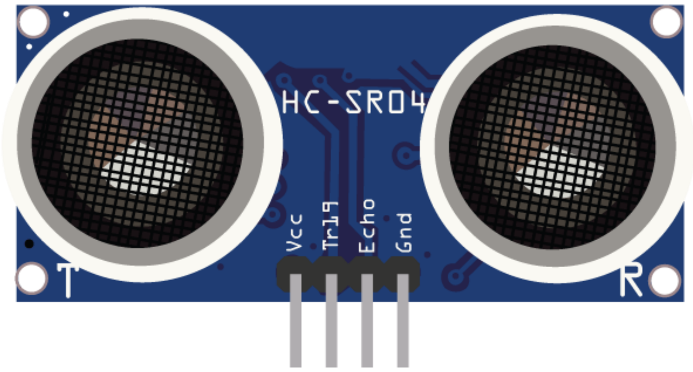


Figure 3.14: Ultrasonic Sensor [19]

3.3.2.4. Relay Module

Relay module is a hardware component which is used to shift the power supply of different appliances attached to it. It has three attachments, one for ground and one for the input and the remaining for output. Motor, solenoid valves and light is attached to relay module which turns on/off the component as per the requirement. It receives the 220V current from the main power source and breaks the circuit to cut the power supply. 8-channel relay module is used which is connected to the NodeMCU and receives the ground and Vcc from it.

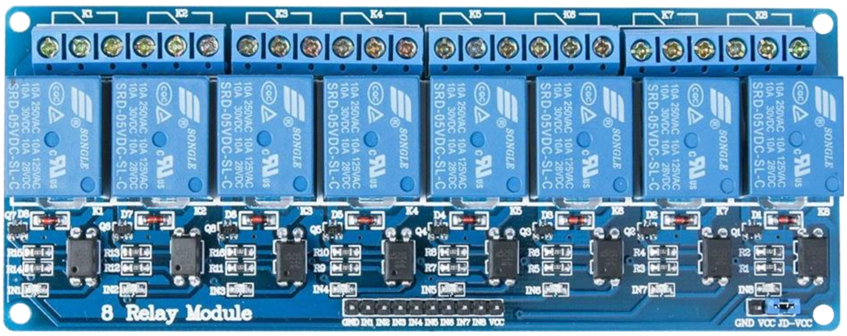


Figure 3.15: Relay Module [20]

3.3.2.5. Water tanks

Water tanks relate to the whole system in which water will be supplied and results will be analysed by water level sensors attached to their tops. The information is recorded and transferred to the other components.

3.3.3. Software Interfaces

Mobile application is developed on Android Studio of the latest version. For database management, DB queries are used for storing data for scheduling. Arduino IDE is used for the programming of Arduino UNO and NodeMCU. It involves many libraries to configure Wi-Fi and to perform serial communication between Arduino and NodeMCU. Operating system should be Windows 10 because Arduino IDE and firebase integrates with it easily.

Firebase is used as a cloud database for storing real time information. It consists of hierarchical structure in which data is stored in nodes or leaves. From there, information is retrieved by the application. The data is sent to firebase in the hierarchical form by the NodeMCU.

3.4. Database Requirements

There are two types of databases which are used at different ends of the system. One database which is used is at the scheduling end in which the user selects a particular schedule to be added for the water supply. This schedule is added to a list and stored in the database. The database in this case is used is SQLite. Using this, the schedule's duration, day and date is selected which is stored and then is retrieved at a particular time for a specific duration.

The other database which is cloud based is Firebase. It stores the real time information retrieved from the sensors applied at the tanks. The statuses are sent by the sensors through NodeMCU which is the Wi-Fi module and is retrieved by the firebase which is accessed by the mobile application. The cloud database has nodes in which information regarding a specific element is stored and then retrieved.
Using firebase, authentication of the user is also performed. Each user can login at a particular time and the information regarding the user is stored. While logging in, the user is authenticated to ensure privacy.

3.5. Non-Functional Requirements

3.5.1. Performance

The Smart Water Management for Domestic Use must respond quickly as it is to replace the human work. The response time of the commands sent through the application to the hardware should be less enough to perform the action as fast. Moreover, user has to visualize the tank status so it should be giving the real time statuses updating it every 5 milliseconds.

3.5.2. Reliability

Reliability of the system depends upon the reliability of its components. It is expressed as quality of the system. As far as reliability is concerned, Smart Water Management for Domestic Use must not compromise on the data which is provided to the user. The user should get the corrected information so that it can take further action. If the data attained is not reliable and ambiguous, then purpose of the system is not fulfilled.

3.5.3. Security

The Smart Water Management for Domestic Use must be kept safe from the normal staff. This means if it is installed in any flats or hostel then the admin worker should be handed over the authority to use and manage the water supplying system in the area. If the access is given to others, then there will be disruption in managing the water and the application becomes unsafe to use.

3.5.4. Consistency

The Smart Water Management for Domestic Use must be consistent in manner i.e. there must not be any internal contradiction in its performance. The System must be capable enough to manage the load. It must provide similar result each time tested with the same instruction given. The information communication between the components should not be distorted and results should be accurate.

3.5.5. Modifiability

The Smart Water Management for Domestic Use should be modifiable according the user’s requirement. If the components which are installed have to be displaced to one's need, then the system should be capable to adapt the change. If the system is to be expanded, then this modification must not affect the system.

3.5.6. Usability

The Smart Water Management for Domestic Use must be easy to use and the user interface of the application must be comprehendible for the common user. The navigation and different options which are available should be well explained and processes must be performed without the ambiguity.

3.6. Project Feasibility

Feasibility study is being performed in order to determine whether the project is feasible within required time and cost constraints.

3.6.1. Technical Feasibility

Smart Water Management for Domestic Use is technically feasible. It will provide required services for the user. By using different high-performance hardware components, the information regarding status of water is recorded and the firebase real time database is maintained. Little training is required for the user to understand the components and hazards as it involves power supplies. The cost of the software is kept minimum, only the hardware requires cost to be implemented.

3.6.2. Operational Feasibility

Operational feasibility usually includes the process through which proposed system will solve the problems. The Smart Water Management for Domestic Use is going to replace the human manual work of managing the water areas in their residential areas to an automated system. It will assist in supplying water according to schedule to turn on/off motor to reduce the water loss. The chances of any mistakes are very less except for the hardware anomaly such as short circuiting. The application is interactive and performs the action given by the user. It will also notify the user of the current actions being performed along with the current states.

The motor and the solenoid valves attached will consume 220V whereas the hardware components such as valves and chipsets will require 5V maximum to operate.

The software will be developed on Android studio and will be operated on android phones. This application will be connected to the firebase which is cloud source for storing real time data in database. The Android application will be needing the latest version of JDK to be operated.

3.6.3. Legal & Ethical Feasibility

Smart Water Management for Domestic Use is legally and ethically feasible.

- It is not going to harm any user unless guides and instructions are followed for operating it.
- Guide will be issued with the system.
- Privacy of the information is maintained.
- The user logged in credentials are not breached.
- The sensors emit no harmful rays.
- Use of this project is legal in our states.
- It is just software that is going to help large number of people with the assistance of hardware.
- Water management will get better with the presence of this product.

3.7. Analysis Models

In the following flow diagram, the complete overall process view of the Smart Water Management for Domestic Use is explained as shown in the figure 3.16:

3.7.1. Flow Diagram

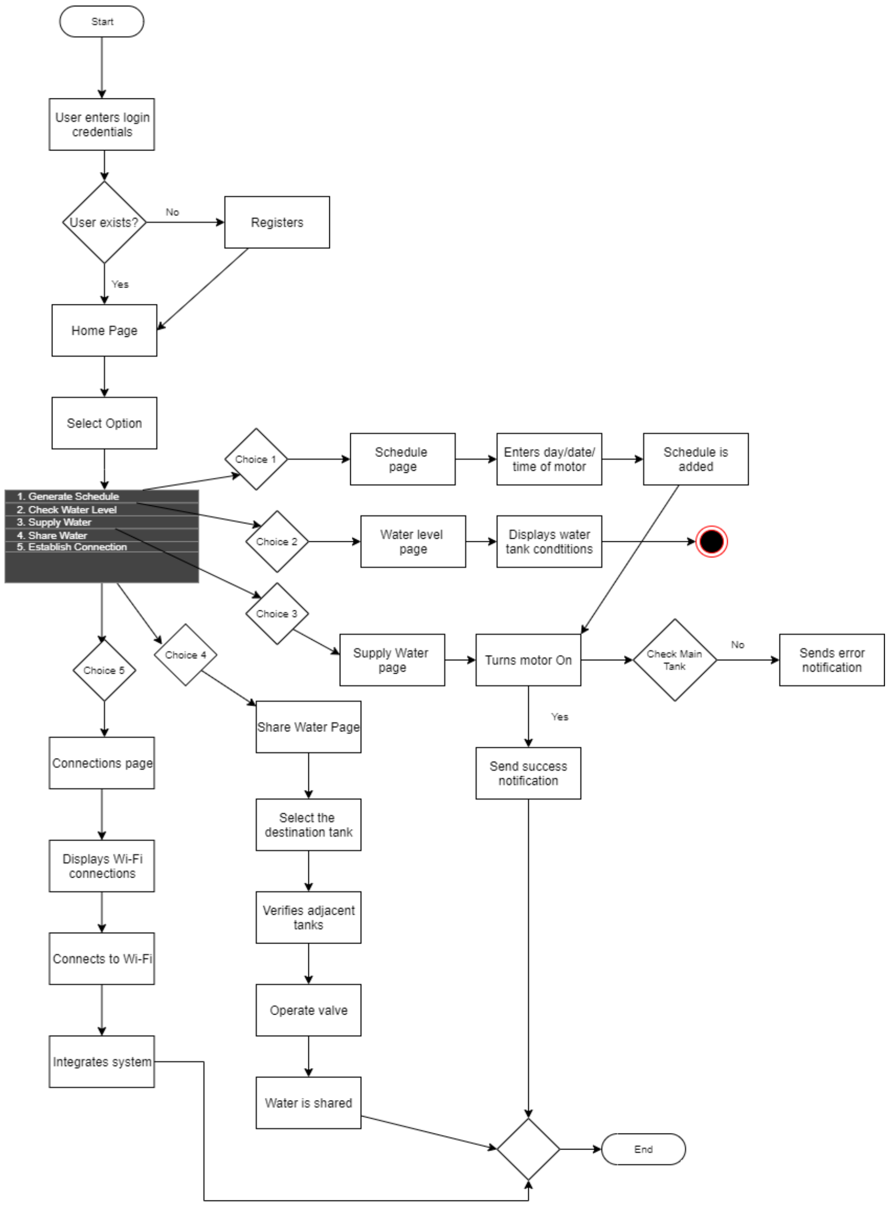


Figure 3.16: Flow Diagram

3.8. Conclusion

In this chapter, the key aspects of this system is defined and explained. This chapter explains the main functionality and modularization of the whole system of how the system is divided into different modules and these coordinate to become a working system. The use cases which are explained describes the actual flow of the system on the whole. Each use case defines a functional requirement which is further explained as breakdown use cases. Moreover, non-functional requirements which are the compulsory aspect of the system are explained. The use cases or functionality are achieved through some interfaces such as software, hardware and user interface for user is described. This system has its applications in the environment if it is feasible to be operational. For this purpose, project feasibility is also discussed in the chapter where the feasible aspects of the project are discussed. The whole overview of the system is explained in this chapter.

Chapter 4
System Design

4.1. Design Approach

We used the bottom-up design strategy to achieve our project. In this strategy, we formalized our individual modules, which then combine to become a system forming a hierarchy as shown below.

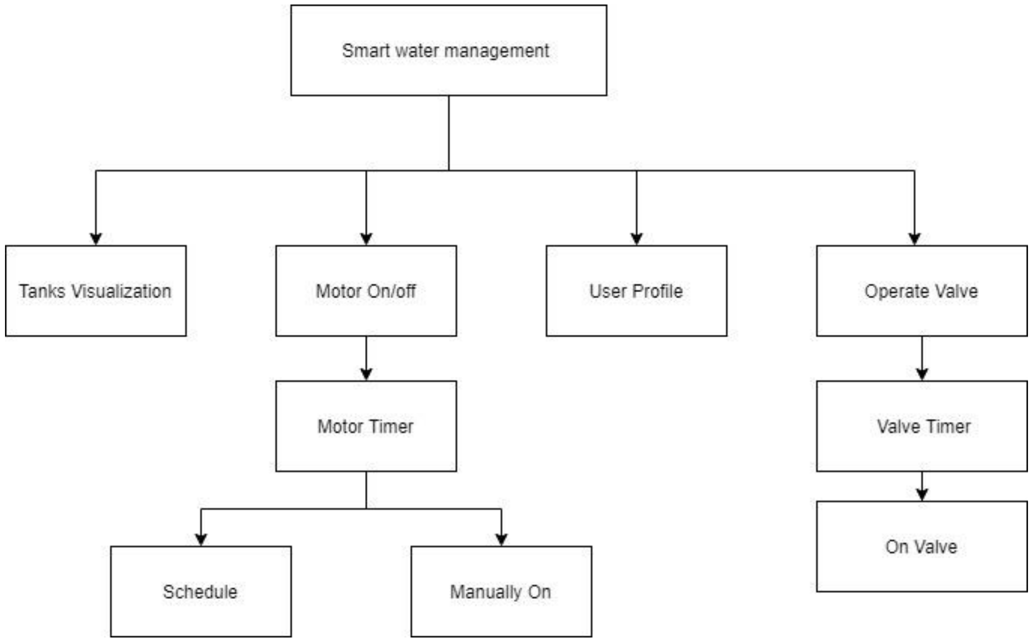


Figure 4.1: Design Approach

4.2. Design Constraints

Arduino and NodeMCU uses languages which are operated in Arduino IDE. These are serial languages used to processing and manipulation purposes. For implementing the mobile application, we have used Java language for android programming. The version selected is of 4.2. For local database, we used SQLite for data manipulation.

4.3. System Architecture

4.3.1. Data-flow architecture:

This kind of architecture is used when input data to be transformed into output data through a series of components. The data will flow from start to end going through components and will be executed at each point depending upon the requirement.

In our project, the data will be flown through the main components which will be applied at each end thus creating an action. Our main components are:

- Application
- Real time Database

- Microcontrollers
- Sensors
- Hardware devices (tanks, motor, valves)

The data will flow from application on request of the user, then it will be passed on to Firebase which is our real time database. From there, it will be processed and sent to

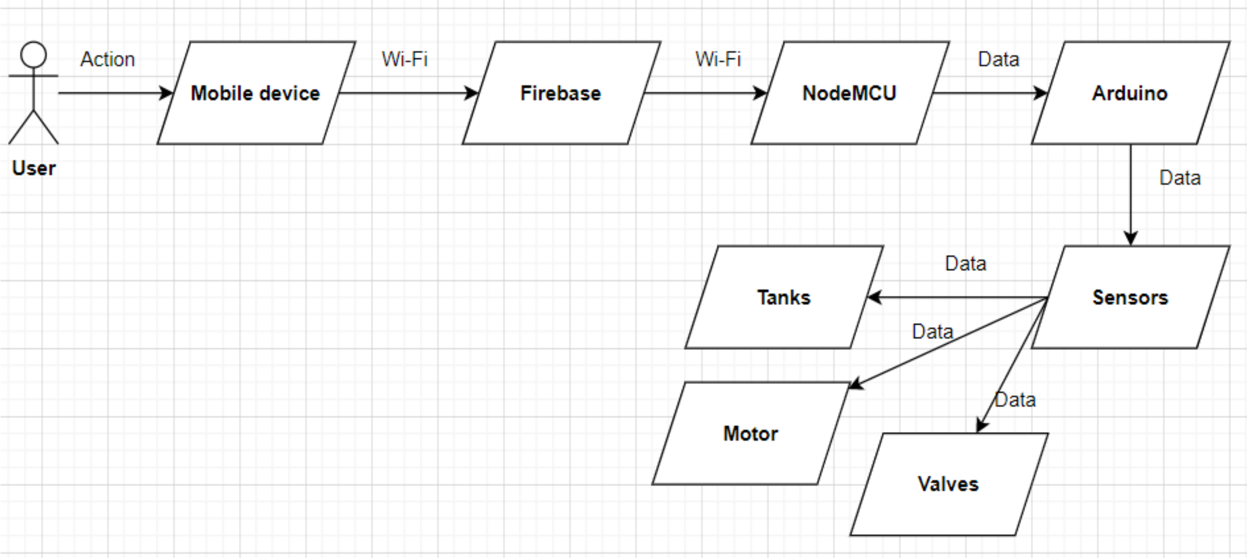


Figure 4.2: Data Flow architecture

NodeMCU which is our Wi-Fi module, it will carry the instruction from firebase and pass it to Arduino chip. The Arduino will carry out the operation coordinating with the sensors and then make changes on the hardware devices. In the same way, the response from the hardware devices or sensors will be passed in the reverse direction and this data is passed between components as shown in the figure.

4.3.2. Data-centric Architecture:

In this architecture, there is a data store which acts as the central hub of the system. This data store could be any database or file system. Various functionalities can be carried out using this data store as it allows to add, modify or pass data between different components.

In our project, there are different components. Some of these are present at the hardware end and at the other end, there is an application which sends it instructions by interacting with the database which is Firebase in our case. So firebase acts as a data store which has all the instructions. For instance, in our case of generating a schedule for water supply, the application will send it to the real time database, and it will trigger the Arduino to carry out the operation at the required time. In this case, the data is stored in the form of a schedule on the database and it is passed to the hardware components which interact with it over the Wi-Fi as shown in the figure.

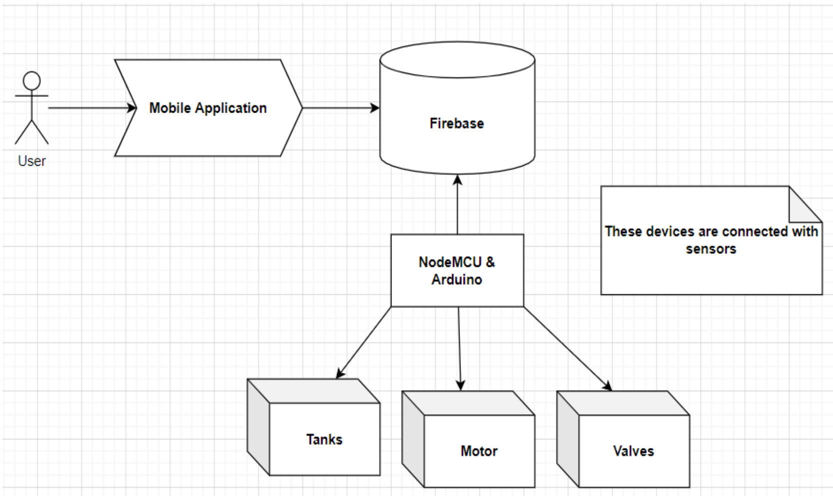


Figure 4.3: Data stored as schedule

4.4. Logical Design

4.4.1. Domain Model

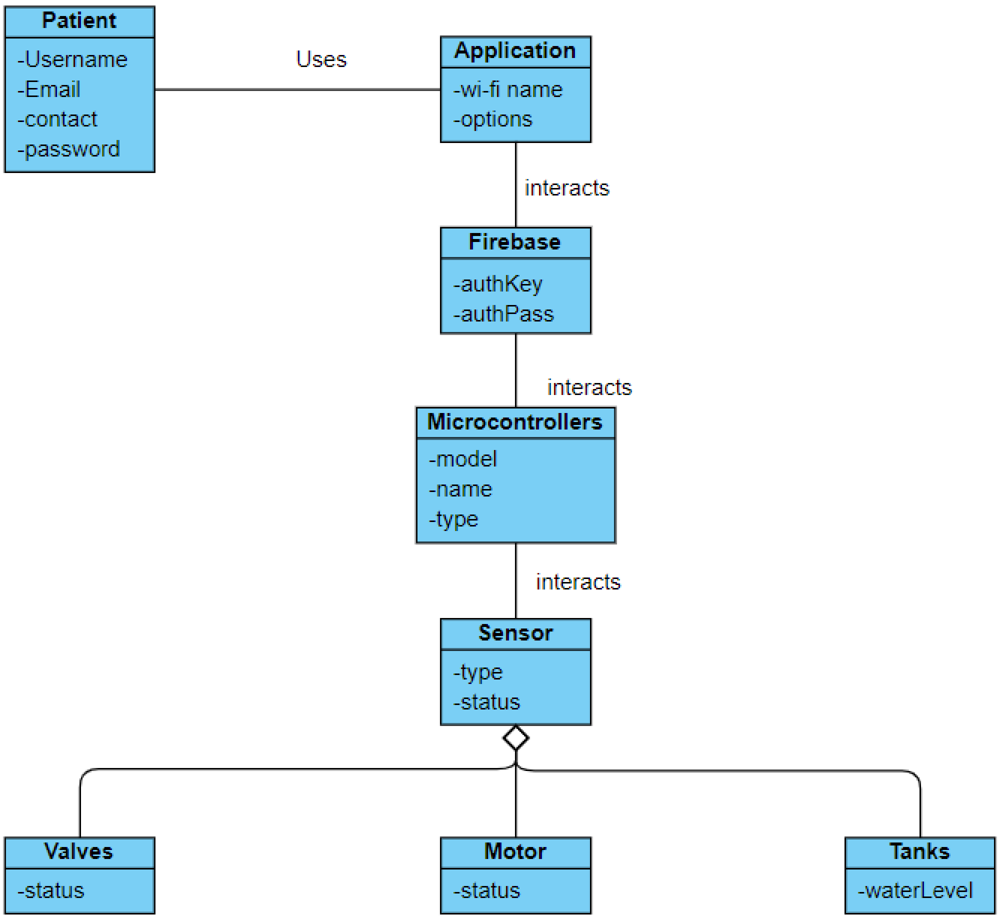


Figure 4.4: Domain Model

4.4.2. Class Diagram

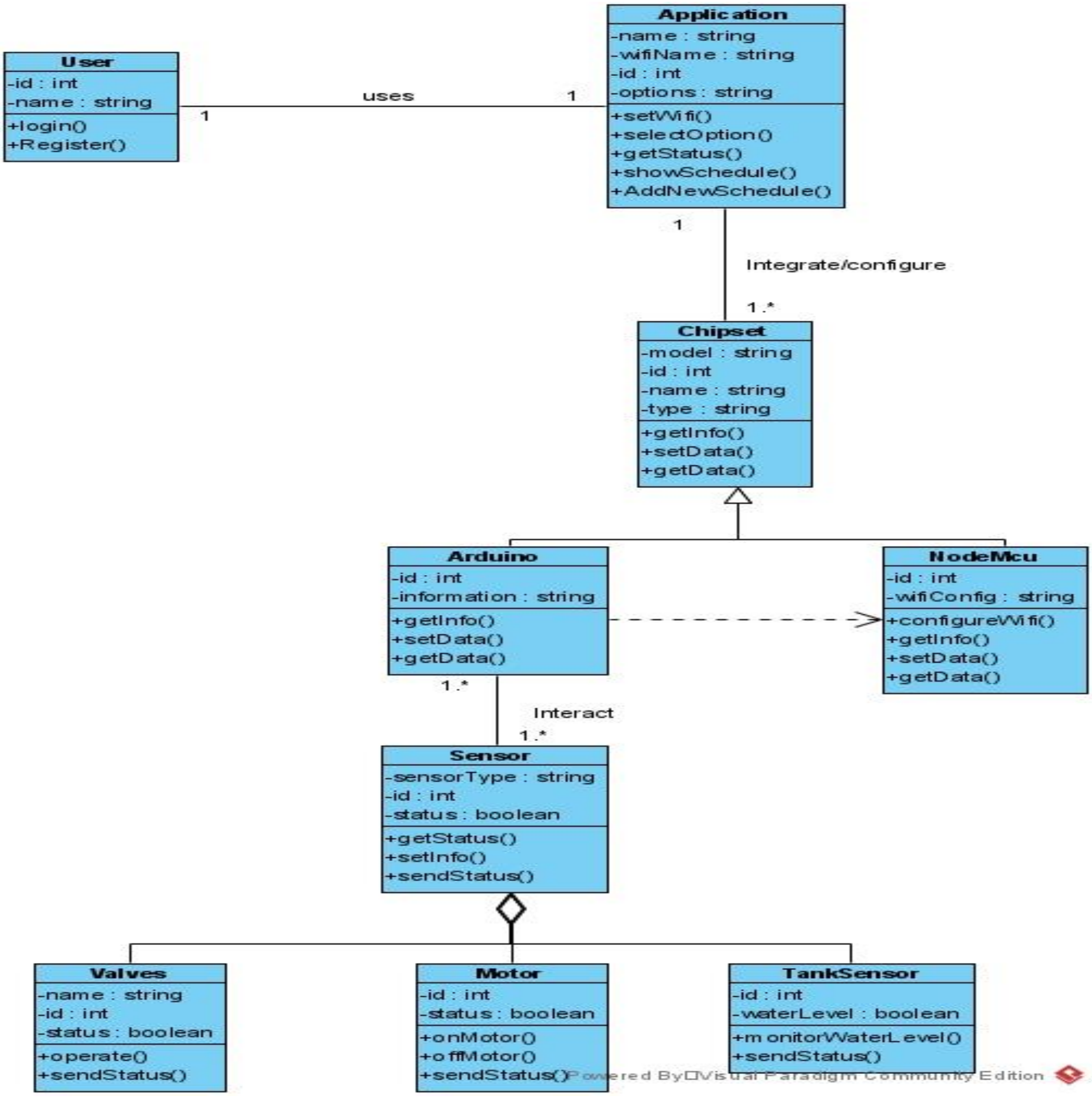


Figure 4.5: Class Diagram

4.5. Dynamic View

4.5.1. State Machine Diagram

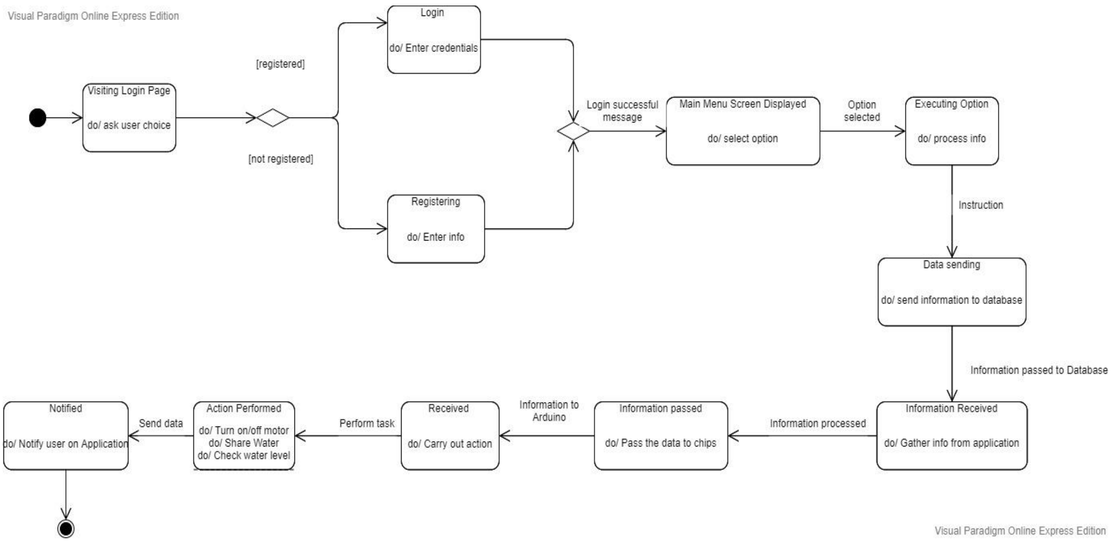


Figure 4.6: State Machine Diagram

4.5.2. Sequence Diagram

4.5.2.1. Check Water Level

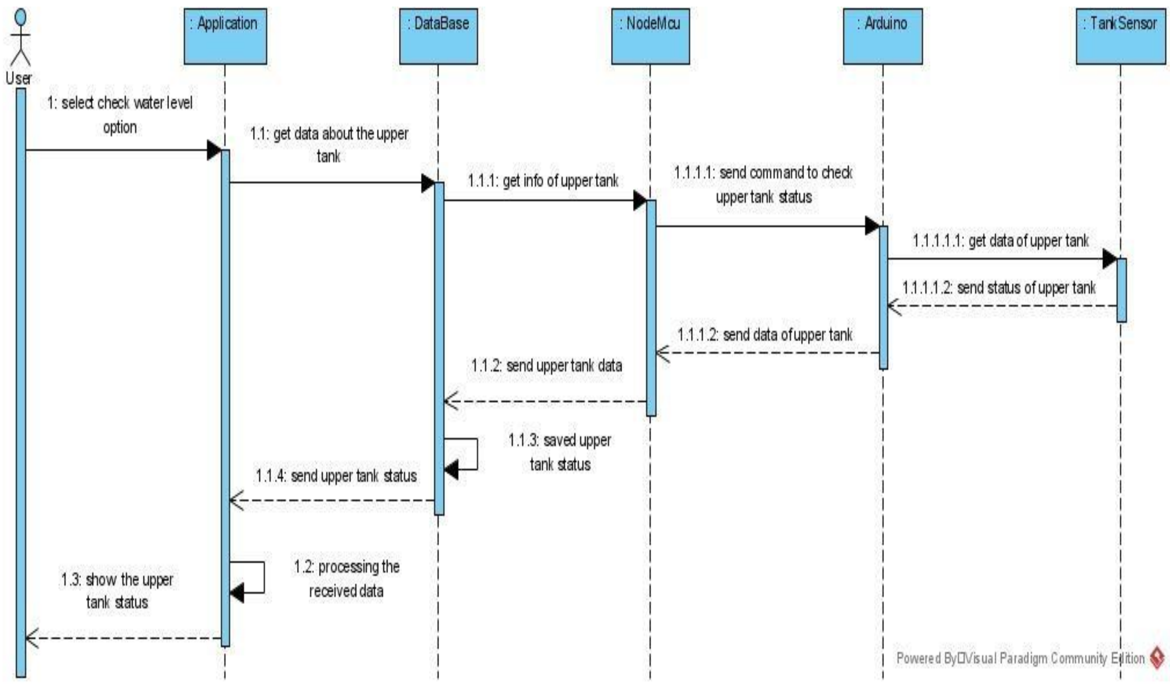


Figure 4.7: Sequence Diagram - Check Water Level

4.5.2.2. Motor On

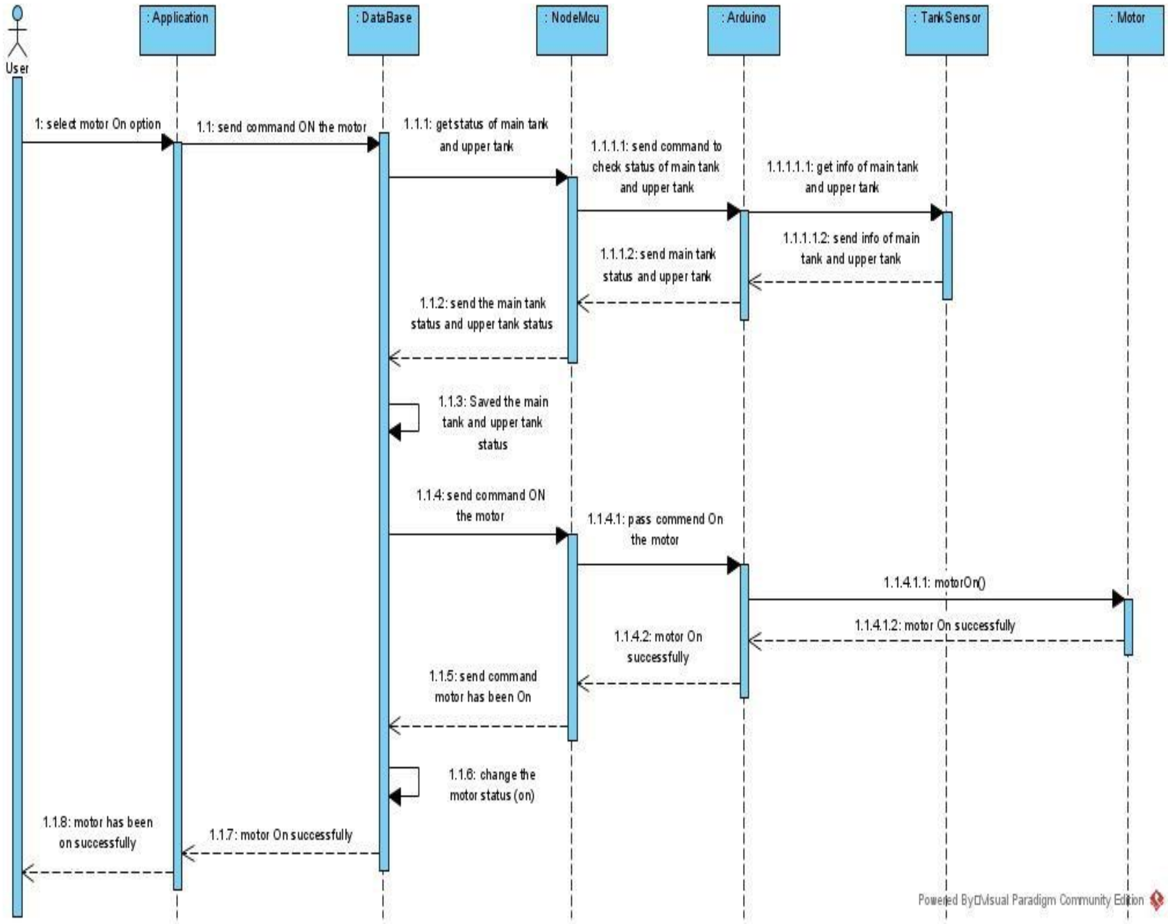


Figure 4.8: Sequence Diagram - Motor On

4.5.2.3. Operate Valves

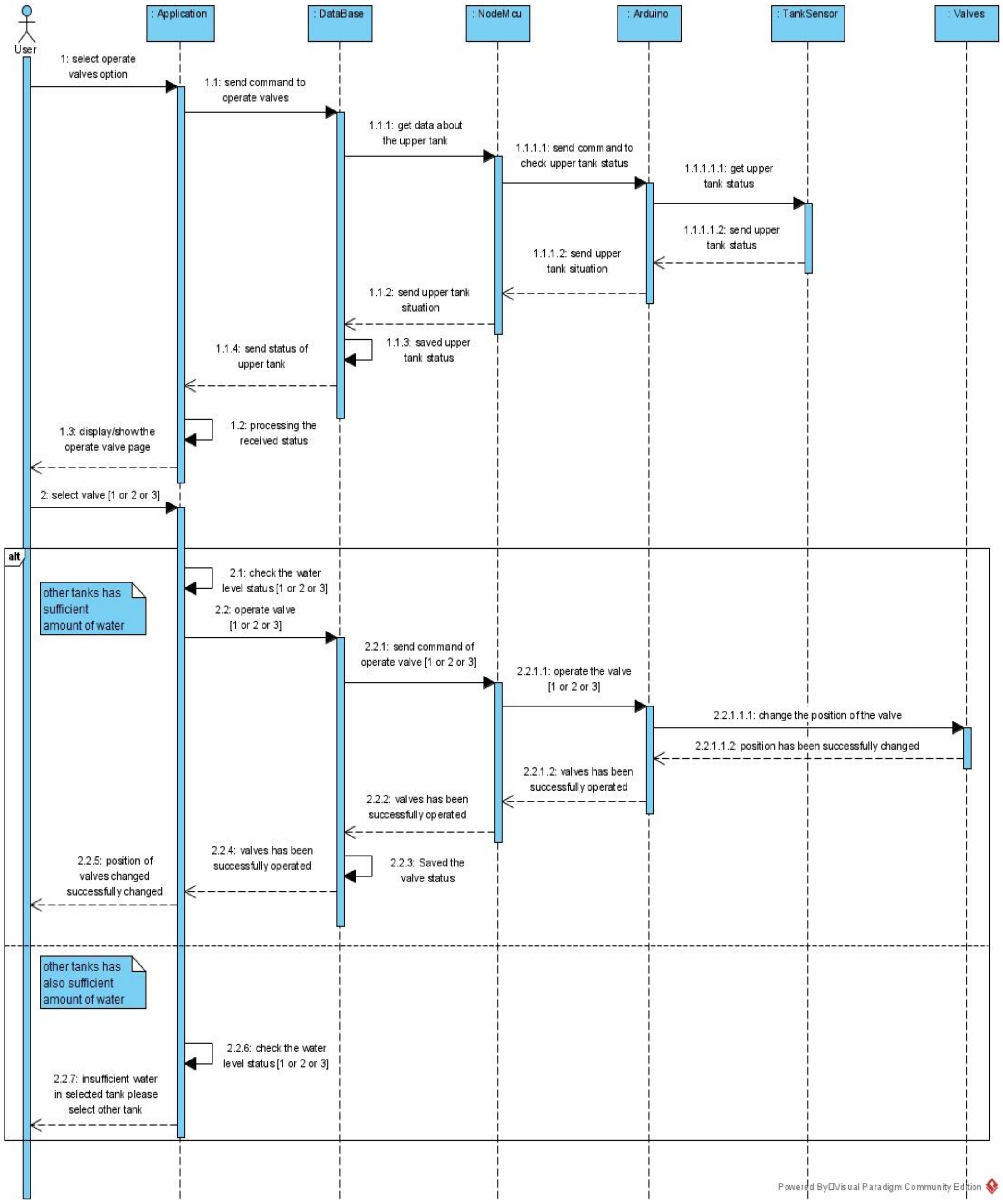


Figure 4.9: Sequence Diagram - Operate Valves

4.5.2.4. Generate Schedule

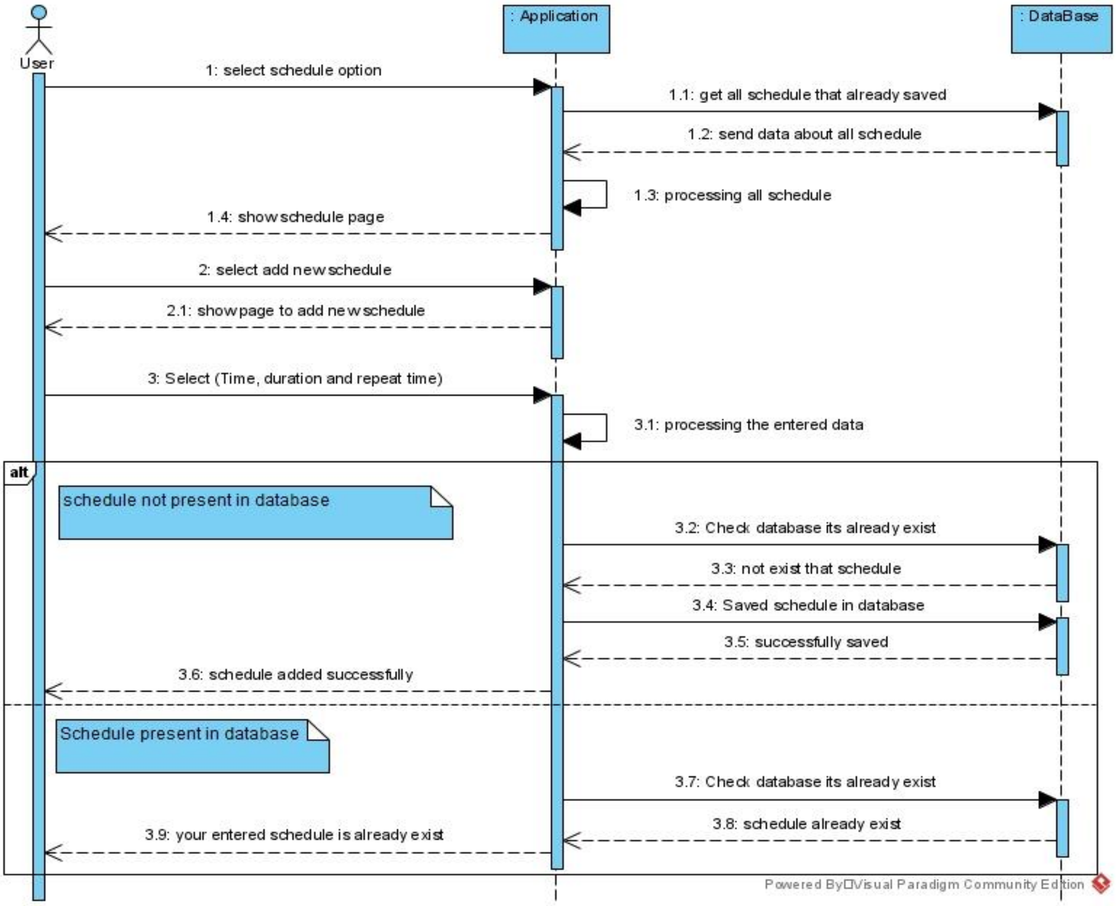


Figure 4.10: Sequence Diagram - Generate Schedule

4.5.2.5. Login and SignUp

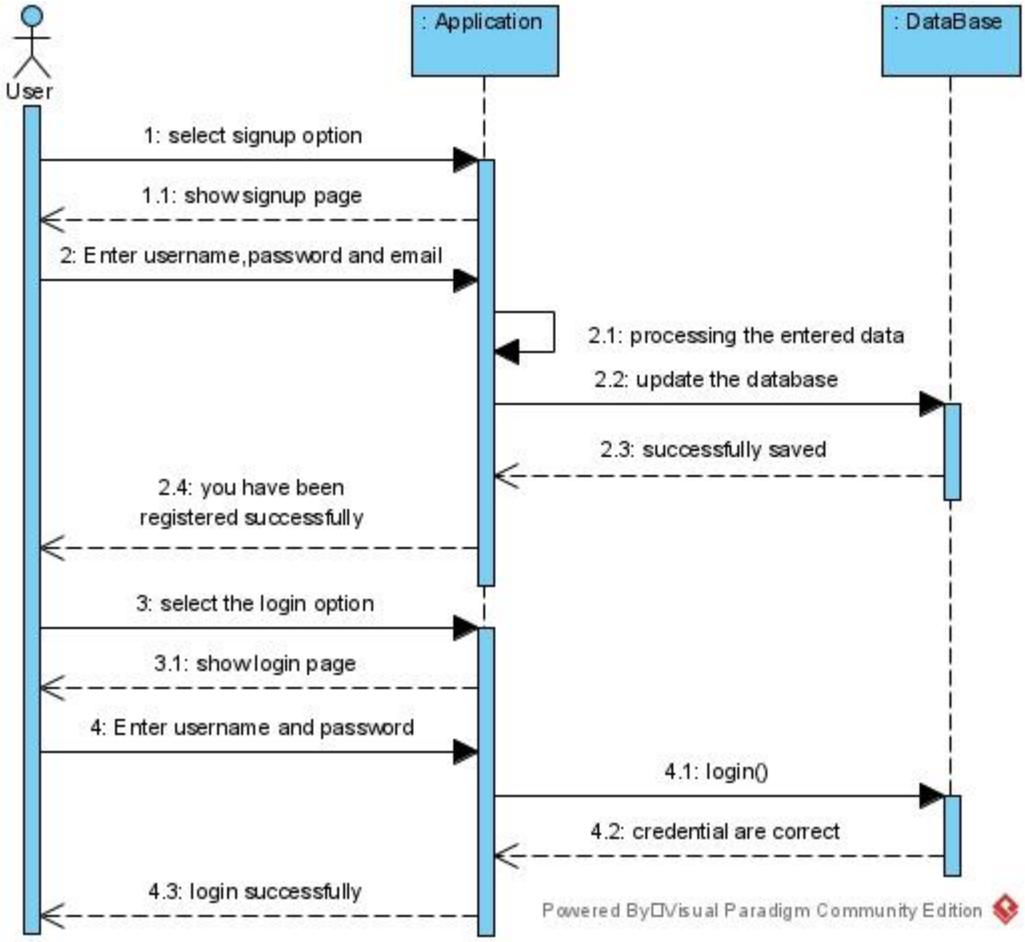


Figure 4.11: Sequence Diagram - Login and SignUp

4.5.2.6. Motor Off

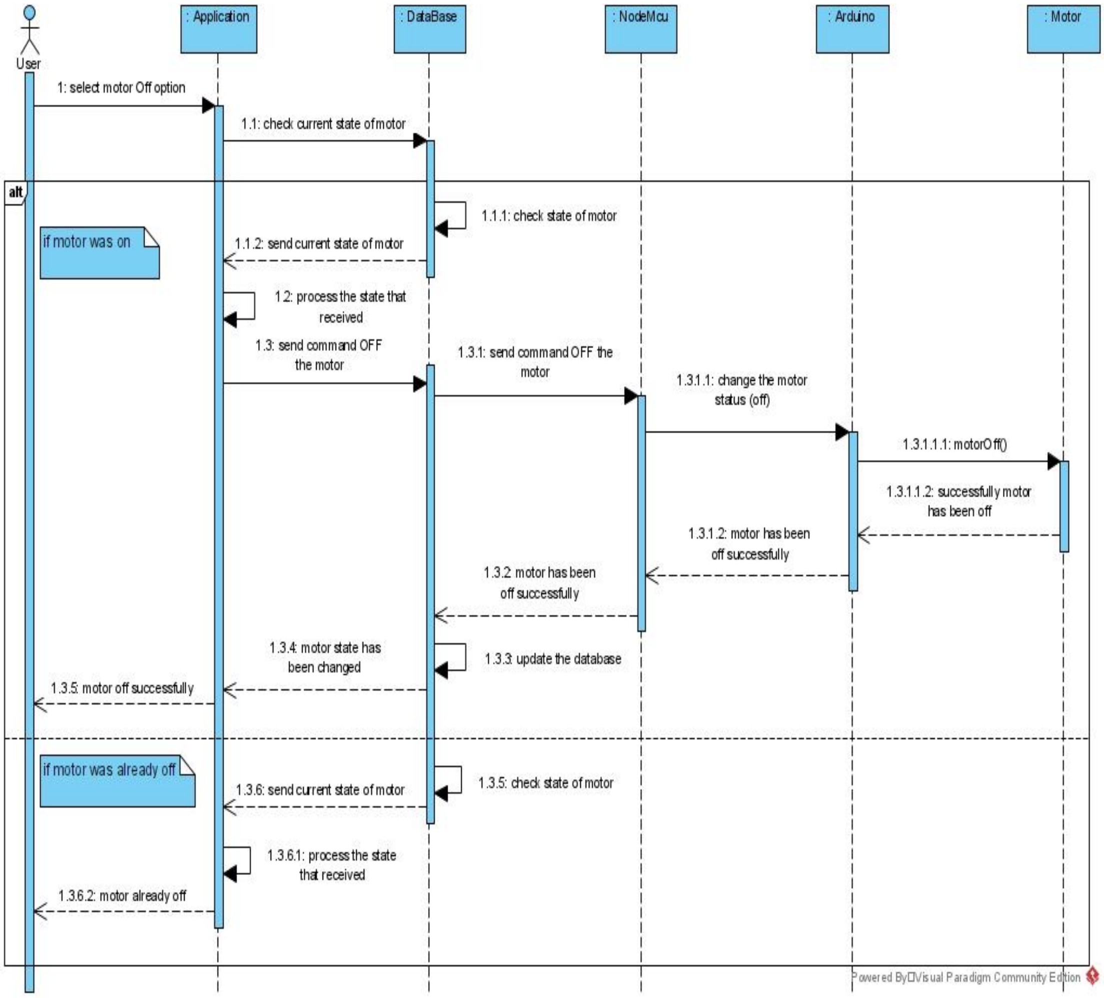


Figure 4.12: Sequence Diagram - Motor Off

4.5.2.7. Establish Connection

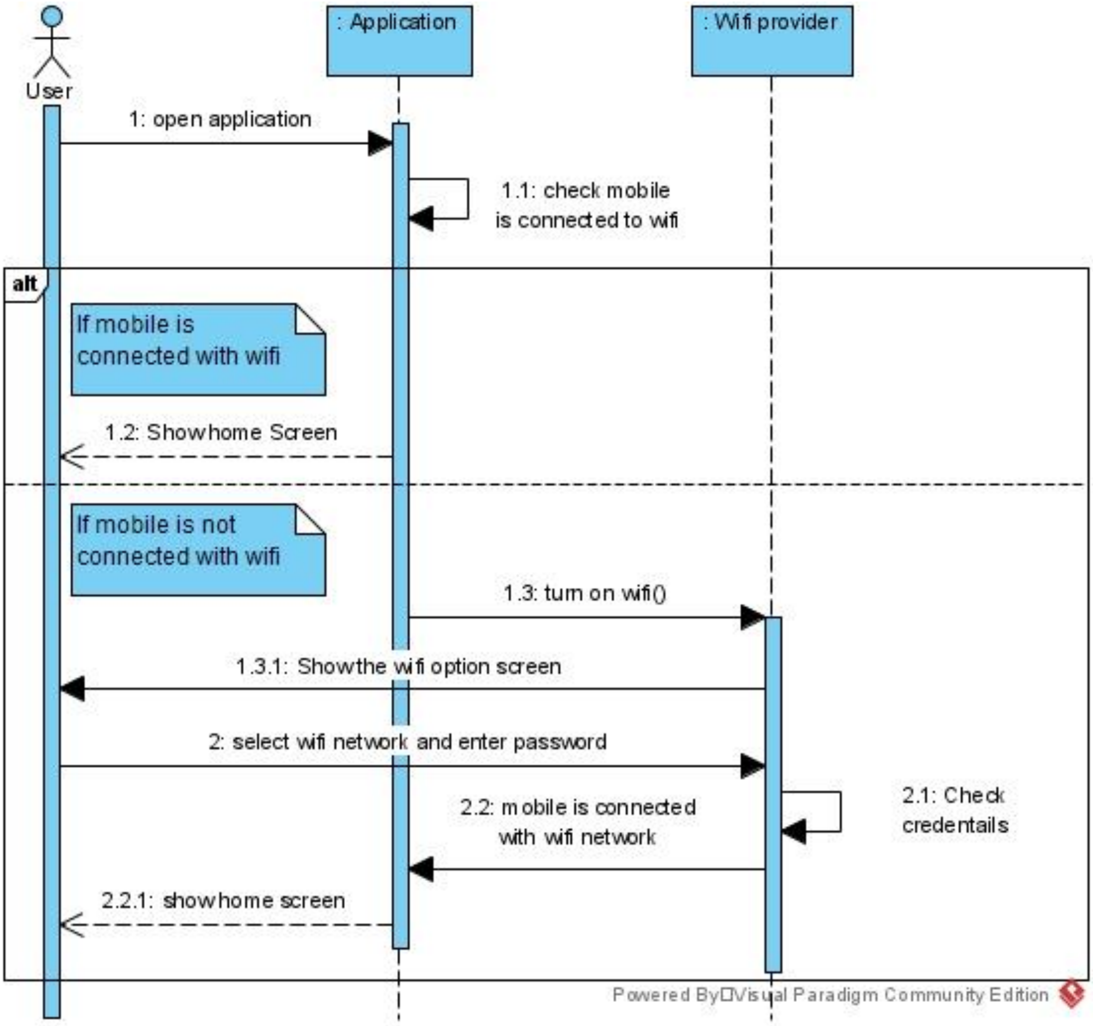


Figure 4.13: Sequence Diagram - Establish Connection

4.6. Component Design

4.6.1. Component Diagram

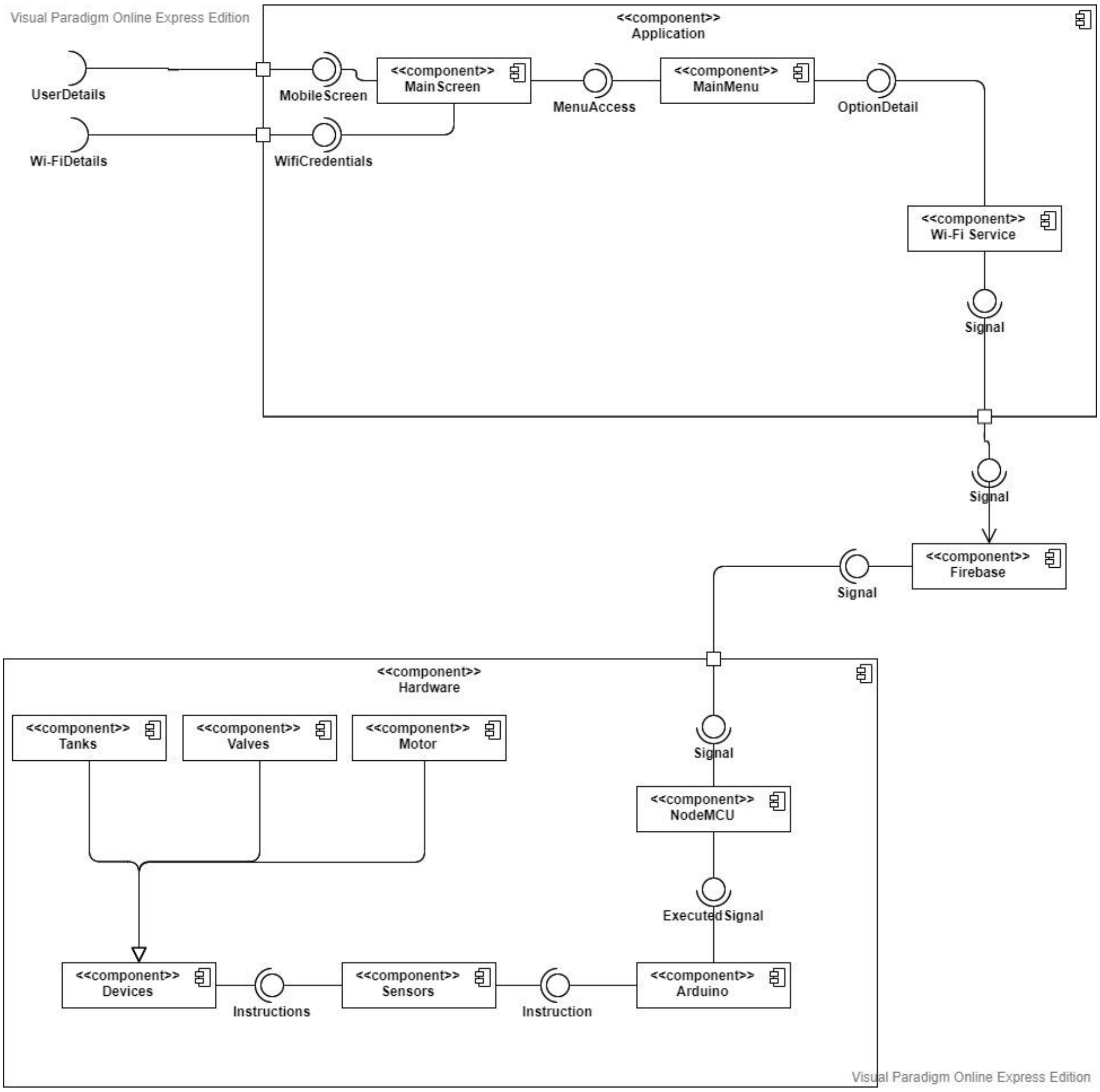


Figure 4.14: Component Diagram

4.6.2. Deployment Diagram

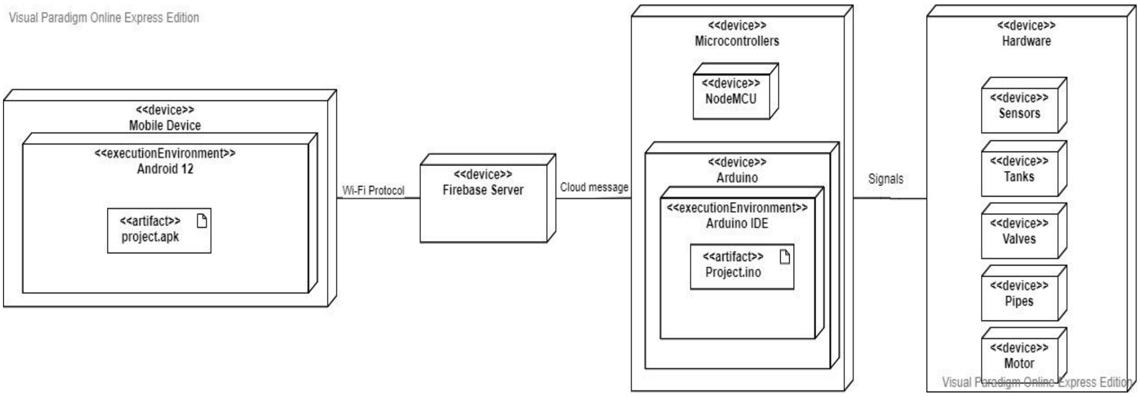
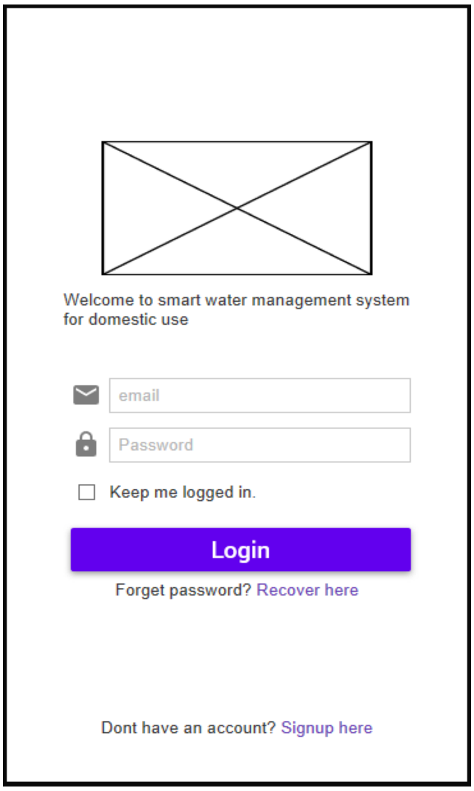
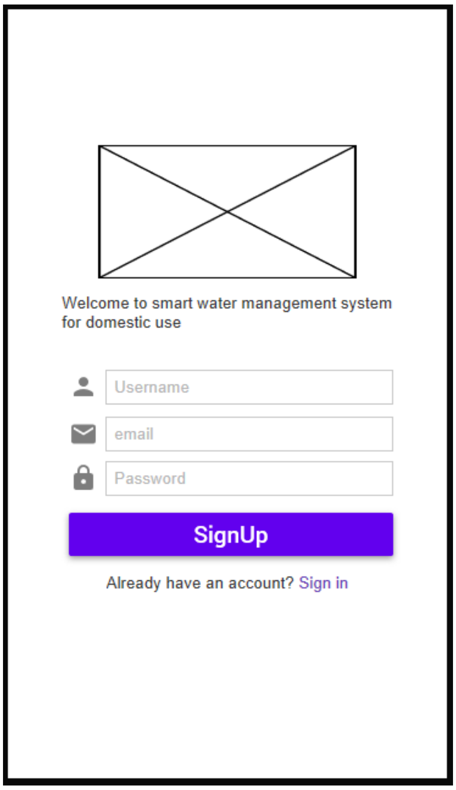


Figure 4.15: Deployment Diagram

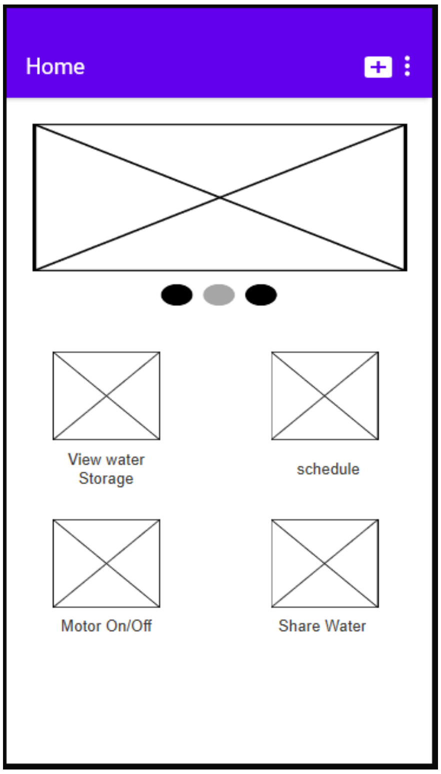
4.7. User Interface Design

4.7.1. Low Fidelity Prototype

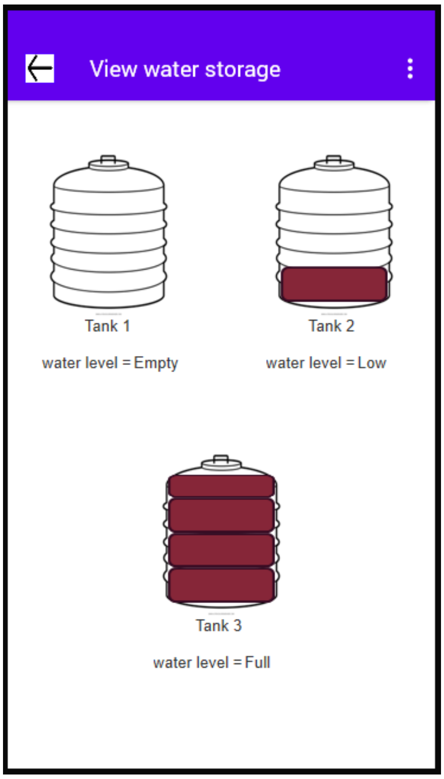




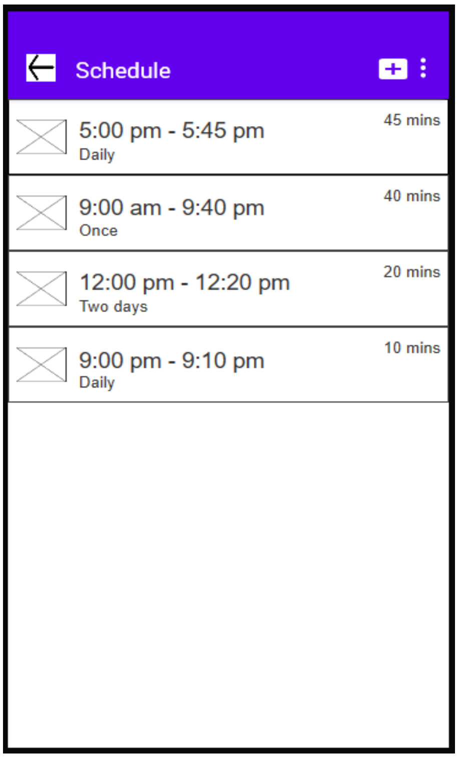
Login Page **Registration Page**



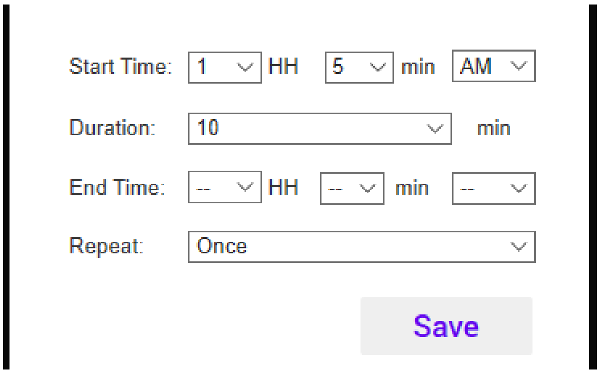
Home Page



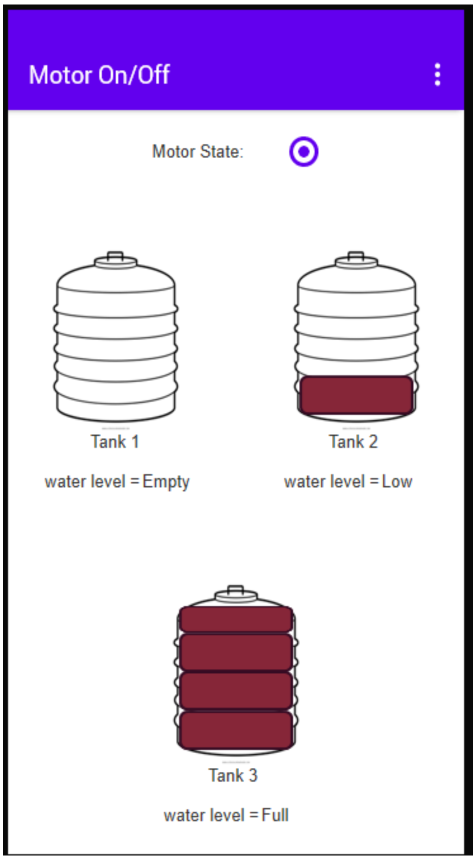
Water Visualization Page



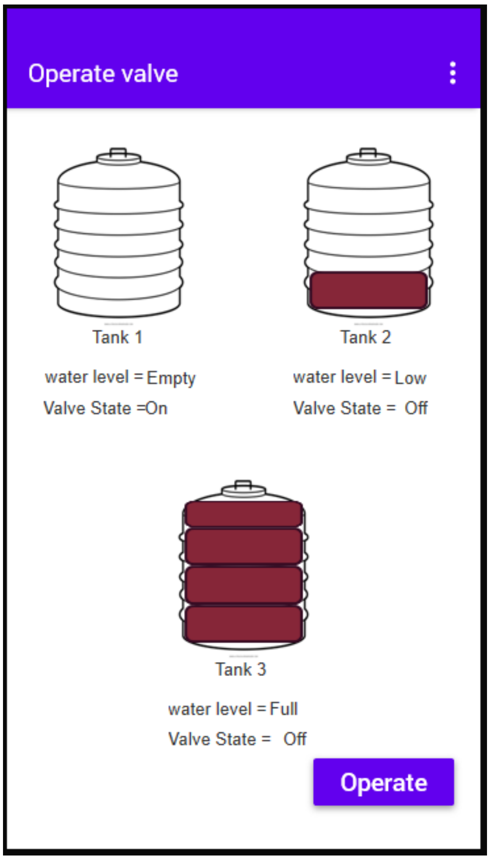
Scheduler Page



Schedule Selection

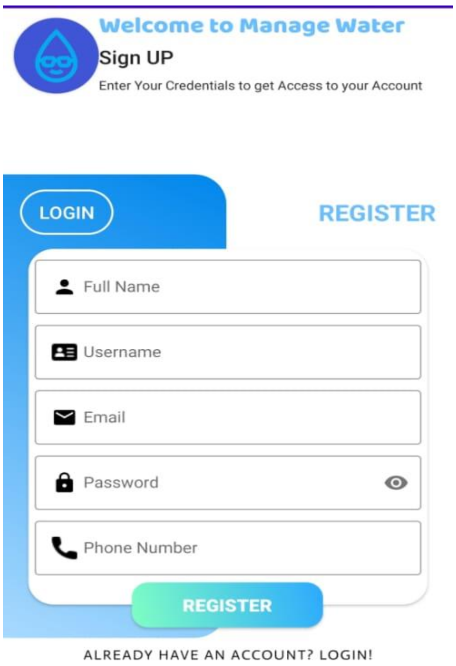


Motor On/Off

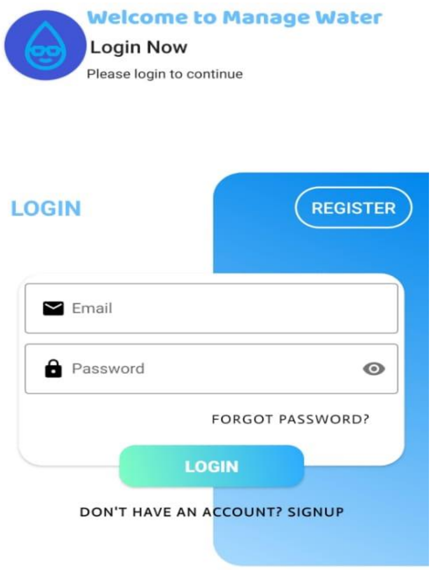


Operate Valve

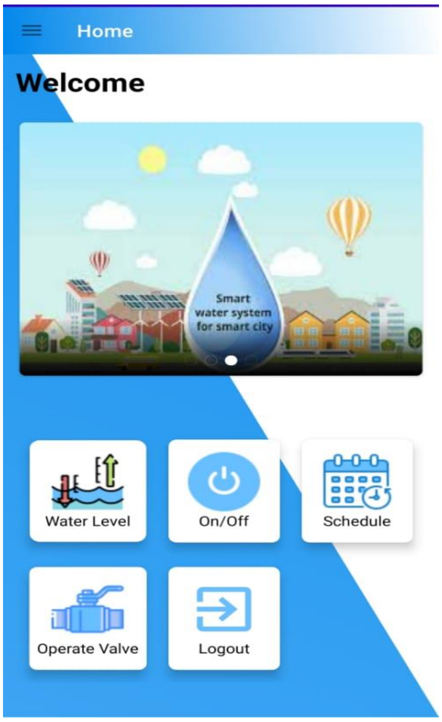
4.7.2. High Fidelity Prototype



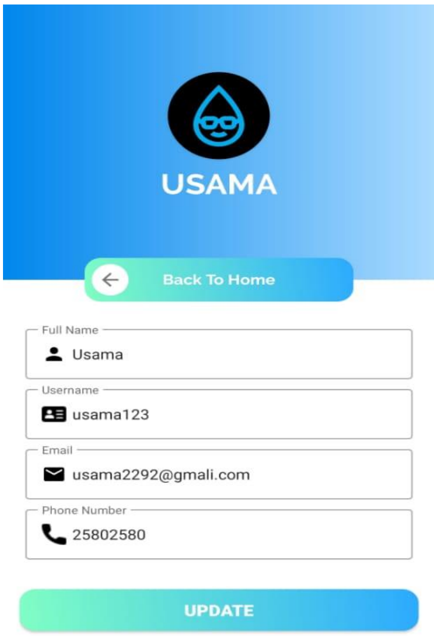
Registration Page



Login Page



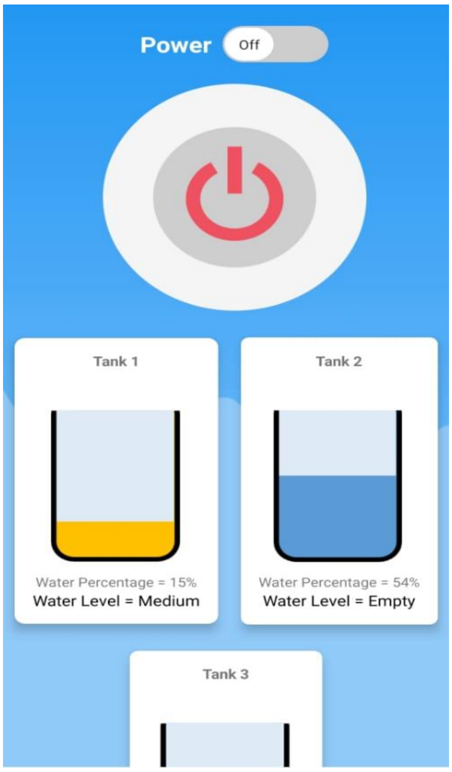
Home Page



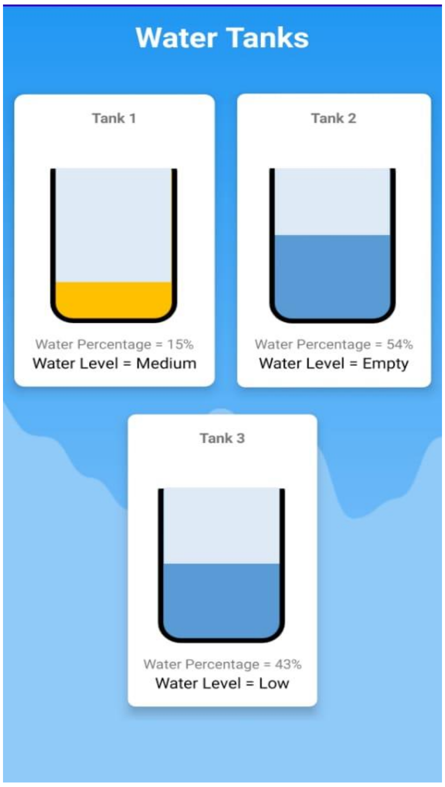
Update Profile

Schedule List

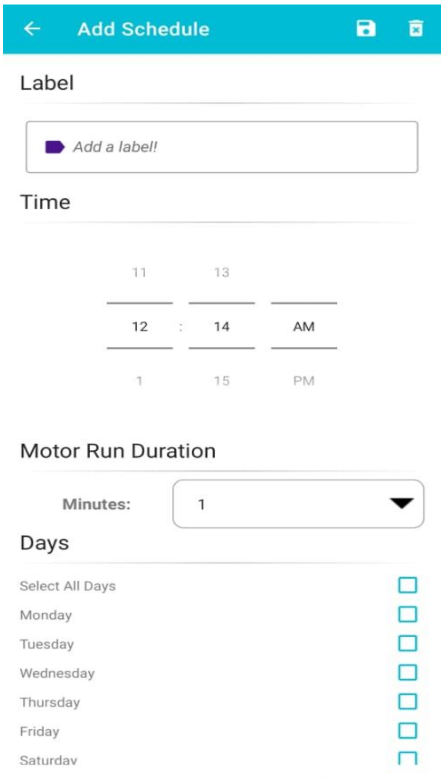
Reset Password



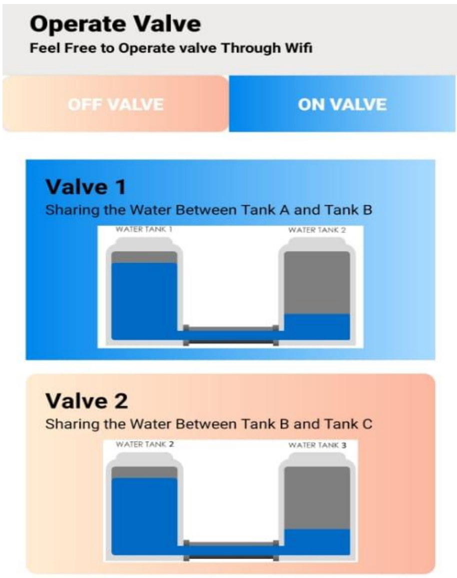
Water Visualization



Motor On/Off



Add Schedule



Operate Valve

4.8. Conclusion

In this chapter, the system architecture is discussed in detail and how the system will be operated using specific design styles. The architecture of the system defines how the operations will be performed in different scenarios. Different views of the system are also discussed. System is logically presented in domain model where it is clearly identified that in which domain this system lies. The class diagram defines the entities as classes of the whole system. The class diagram is identified which shows the significance of the communication between different classes. Furthermore, in dynamic view we present the nature of the system. The significance of presenting the system dynamically is to show the sequence of each process which goes with the coordination. The sequence diagrams are shown in the section where it is clearly identified that how system will behave in particular scenario at a particular time. The system is formed with the conjunction of different components and how the components behave is also discussed in this chapter in the form of component diagram. In component diagram, each component has many artefacts which merge to form a component. Among the component lies the communication that how components interact. As our system is made up of hardware components as well, to represent the hardware entities deployment view is designed to further explain the inclusion of hardware and how the whole system is collaborated.

In system architecture and design, it is necessary to show the system in each process. For this purpose, low and high-fidelity prototypes are included to create a view of the proposed interface and the result which is obtained. The final view is high fidelity prototype which lies in some standard keeping in view the human computer interaction.

Chapter 5
System Implementation

5.1. Strategy

The basic strategy used in the project is to integrate the components with the software application which is developed. As this system contains many modules which is to be completed in order to achieve the results. We have developed an application which is tested separately with the configuration of hardware applied at the other end. The hardware is deployed at the other end which provides the results of water states, from where it is sent to cloud database to be retrieved by the mobile application. This integration led us to identify each minor mistake in every step of the project.

5.2. Tools Used

This project has two development areas i.e. software section and hardware section. The tools used in each section are described as below:

5.2.1. Software Tools

The tools which are used for programming purposes are as follows:

- Android Studio for Mobile Application Development
- Arduino IDE for Arduino and NodeMCU programming
- Firebase for real-time database

5.2.2. Hardware Tools

The other side of the system consists of hardware components. It includes hardware technologies which will be used for the processing and manipulation of information. These will be integrated with the equipment which will be installed.

5.2.2.1. Hardware Technologies

The hardware technologies include the technologies which will be used in processing. These are described as below:

- Arduino UNO for retrieving information from sensors and processing it.
- NodeMCU as a Wi-Fi module attaining information from Arduino and sending it over Firebase which is a cloud database through Wi-Fi.
- Relay Module for changing the states as On/Off by receiving 220V.
- Ultrasonic water level sensor
- Solenoid valve for behaving a valve used in water sharing

5.2.2.2. Hardware Equipment

The hardware equipment will be installed with hardware technologies for receiving results

- Water tanks
- Wires and power supplies
- Pipes for transferring water
- Water source

5.3. Algorithms

The basic algorithm of the project revolves around the water states. The whole project depends upon the water condition for the system to take the next action. The main features of the project which are Scheduling, Water supply On/Off , Real time water visualization and Water Sharing which will be notified along with the notifications. If any ambiguous state is received from the sensors, then the system will either respond incorrect or prompt a notification indicating the user of invalid result.

There are algorithms used at the both software and hardware ends depending upon the requirement. These algorithms will work collaboratively to produce a result. While retrieving information from the sensors, the sensors will use the sound waves which will be depending upon the surface of the water. For this purpose, it will send sound wave via echo and then will be retrieved by trig. In the same way for processing the information in Arduino it will be manipulated by using equation. It will be sent to NodeMCU via Serial Communication.

5.3.1. Water Level Detection

Water will be detected from the sensor through sound waves. The echo sensor will send a sound wave and will be received by the trig sensor. In this way the water level distance will be calculated in this way. It will give ambiguous result (in any other measurement) if not set according to the requirement. For this purpose, following equation will be used for retrieving the current status of water:

duration = pulseIn(echopin,HIGH);

distance = (duration / 2) / 29.1;

In the above equation, duration is an integer variable sending the echo wave through the sensor and receiving via trig sensor. It will be converted into centimetres for our purpose and stored in the distance integer variable. This will be applied for all the sensors applied for the use.

5.3.2. Scheduling

Scheduling is done so that user can set a time and day for motor to operate. The schedule will be generated and added to the list as soon as the user selects a particular day and time along with duration. Moreover, the motor will start at the selected time and will be turned off as soon as the duration is completed. The algorithm is designed in such a way that if the water conditions in the tank are full before the duration, the motor will be turned off and will be notified to the user. Each schedule will be different and if there are two schedules overlapping, it will prompt the user that the new schedule is invalid. The duration will be added according to the user requirement by using the following:

```
time.set(Calendar.MINUTE, ViewUtils.getTimePickerMinute(mTimePicker));
time.set(Calendar.HOUR_OF_DAY, ViewUtils.getTimePickerHour(mTimePicker));
time.set(Calendar.SECOND,0);
time.set(Calendar.MILLISECOND,0);
alarm.setTime(time.getTimeInMillis());
```

In the same way, the duration which is set is also set for the particular day which is performed using the following method:

```
alarm.setDay(Alarm.MON,mMon.isChecked());
alarm.setDay(Alarm.TUES,mTues.isChecked());
alarm.setDay(Alarm.WED,mWed.isChecked());
alarm.setDay(Alarm.THURS,mThurs.isChecked());
alarm.setDay(Alarm.FRI,mFri.isChecked());
alarm.setDay(Alarm.SAT,mSat.isChecked());
alarm.setDay(Alarm.SUN, mSun.isChecked());
```

5.3.3. Water Sharing

One of the key concept is water sharing in our system. It will be done by the solenoid valves placed between the tanks. The tank with the least water in it will be requiring water from the adjacent tanks. The adjacent tanks will be checked and then tank with more water will allow the flow of water through the solenoid valve. The checks implemented which will operate the valves are as follows:

5.3.4. Water Visualization

The water level visualization is performed as the sensors detect the real time state of water in the tanks. As there are multiple tanks installed along with the main tank which is the source. The sensors retrieve the data and transfer into the Arduino. Arduino processes the data and sends it to firebase through NodeMCU. From firebase, the real time states of water tanks is accessed by the mobile application.

5.4. Methodologies

There are many methods used in the project which will be used in attaining a result. As this system has software and hardware ends, there requires method for the transfer of information to and fro. Moreover, modules like connection establishing with firebase, checking the water level and comparing to get a state, Serial communication between Arduino and NodeMCU, and state transition using relay module.

5.4.1. Serial Communication

Serial communication is the process of sending data one bit at a particular time, which is sent over a communication channel sequentially. In Arduino, Serial is a library which is used for communication between Arduino board and other boards. In our case, serial communication is done between Arduino and NodeMCU so that NodeMCU sends the data coming from Arduino to firebase through Wi-Fi.[13]

In Arduino, pins 3 and 5 will be used as Serial ports for transferring data as Tx and Rx. For this purpose, SoftwareSerial variable ‘s’ will be used as a reference. It will send the information as soon as it will be available.

Another library which will be used is ArduinoJson.h. This is useful for sending multiple information from Arduino to NodeMCU. It will create a data package for transferring data and in the same way will be retrieved by NodeMCU by ArduinoJson object as a reference. As given:

StaticJsonBuffer<1000> jsonBuffer;

```
JsonObject& root = jsonBuffer.createObject();
```

The object ‘root’ is created through this method and will be used throughout where information of each sensor will be stored in this object such as:

```
root["Level"] = "High";
```

The object root will create a sequence containing the states depending upon the data from sensor and will be sent accordingly. After getting all the entries of the states, it will be sent to NodeMCU through SoftwareSerial variable ‘s’ such as:

```
if(s.available()>0)
 {
  root.printTo(s);

 }
```

The information is sent as a packet using these libraries, the pins 3 and 5 of Arduino will be connected to Rx and Tx of NodeMCU respectively to receive information. In NodeMCU programming, the root node will be used as a reference to retrieve the information in the packet such as:

```
JsonObject& root = jsonBuffer.parseObject(Serial);

 if(root == JsonObject::invalid())
 return;
 root.prettyPrintTo(Serial);

String state = root["Level"];
```

The circuit diagram through which serial communication will be performed is as follows:

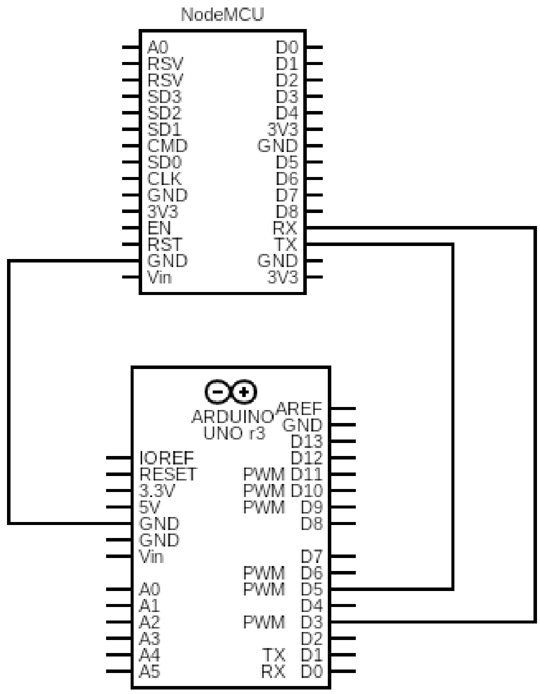


Figure 5.1: Circuit diagram of serial communication

5.4.2. Establishing Wi-Fi Connection

While implementing the system, it is necessary to establish a network connection throughout the system. The Wi-Fi is the primary source of connection of mobile application with firebase and firebase with NodeMCU. Wi-Fi connection of mobile application will be established through the available networks in the mobile device. It will configure with one of the connection and the application will establish a connection ultimately with the firabase.

The NodeMCU will also require Wi-Fi connection to establish a connection to firebase. It will be provided with the desired SSID and password and connection will be established by the following method:

```
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.println("Connected");
```

5.4.3. Establishing connection with Firebase

The next purpose after having a Wi-Fi connection is to establish a connection between mobile application and firebase and firebase to NodeMCU. As firebase is the real-time database for storing the sensors state and the instructions coming from application. The connection with firebase for mobile application will be done by following method:

FirebaseDatabase rootNode;
DatabaseReference reference;
rootNode= FirebaseDatabase.getInstance();
reference=rootNode.getReference("MotorStat");

Where in this case we are retrieving the state of motor which can be on off depending upon the condition.

In the same way, NodeMCU will be coded so that it establishes a connection with the firebase. Firebase has a host name with its secret name for configuration. We will fetch the secrets and code it in our NodeMCU for the configuration. The credentials on firebase are as follows:

Database Host:

https://sensordata-4bc4b-default-rtdb.firebaseio.com/

sensordata-4bc4b-default-rtdb

Database Secret:

Database	Secret
sensordata-4bc4b-default-rtdb	wk5IYgD6SLYmJ5XKW9hTqyN9ozX4VzJg3k7FuH6O

These credentials will be provided while coding for NodeMCU and with the following method it will be configured:

Firebase.begin(FIREBASE_HOST, FIRBASE_AUTH);

The Firebase keyword is derived from the header file included as <Firebase.h>. It allows the functions related to firebase to be called and used.

Data will also be sent from NodeMCU to firebase. It requires .setString() function to be used contained in Firebase library along with the nodes as per requirements. The function uses two arguments, one is the path which will be created and other is the value which is to be sent to firebase. The sensor value of one tank sent to firebase is shown in the following method:

Firebase.setString("SensorsState/WaterLevelUpperTank",state); // Push Sensor condition of upper tank sensor 1 to firebase

Firebase.setString("SensorsState/TankOneDistance",first_distance); // pushing d for first upper tank

The firebase and its entities looks something like this:

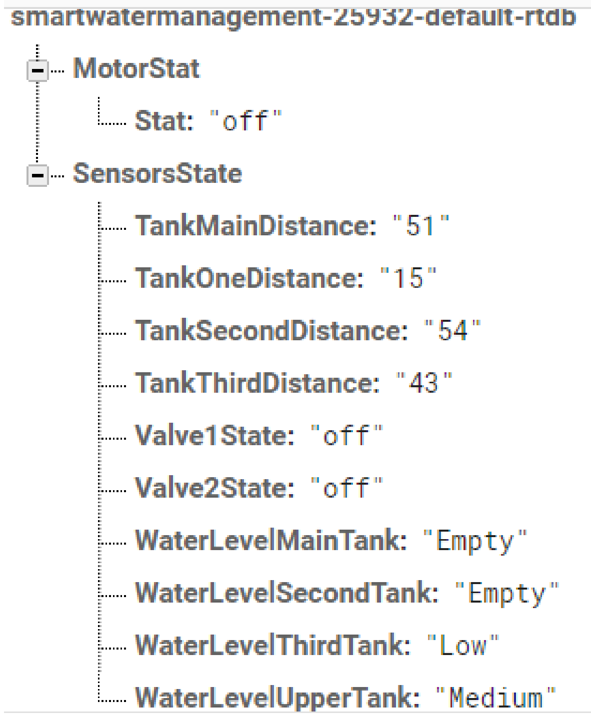


Figure 5.2: Firebase entities

5.4.4. Water level conditions

Water levels which will be retrieved from the ultrasonic sensors will be recorded. These recorded values will be vague unless we set some boundaries to produce a status. We have used three states i.e. High, Medium and Low. The sensors are applied at the top of the tank and the total length of the tank is 100 cm. According to 100 cm, we have set boundaries such that at particular length, the water level state will be given a status let's say High. The following method will be used for each state:

```
if(  (distance >= 0) && (distance <= 20)   )
then High;
else if(  (distance >= 20) && (distance <= 80)   )
then Medium;
else if(  (distance >= 80) && (distance <= 100)   )
then Low;
else
Empty;
```

5.4.5. State transition using Relay Module:

Relay module is used to convert the state of equipment such as in our case is motor and solenoid valve. The relay takes the 220V and it breaks or connects the circuit depending upon the state which is assigned. If the state of motor is Off, then the circuit will be break by relay module and current supply will be turned off and vice versa. The relay is connected to NodeMCU in our case which receives the state from the firebase and then takes action accordingly. The motor is connected to the D2 pin of the NodeMCU and valves are connected to D3 and D4 respectively. The following method will be used to change the state of motor after receiving state from firebase:

```
FirebaseObject object = Firebase.get(path);

String stateMotor = object.getString("MotorStat/Stat");
if(stateMotor == "off")
{
  digitalWrite(MotorPin, HIGH);
}
else
{
  digitalWrite(MotorPin, LOW);
}
```

The connection of NodeMCU with relay modules connected with motor (M) and solenoid valve (S) is illustrated as follows:

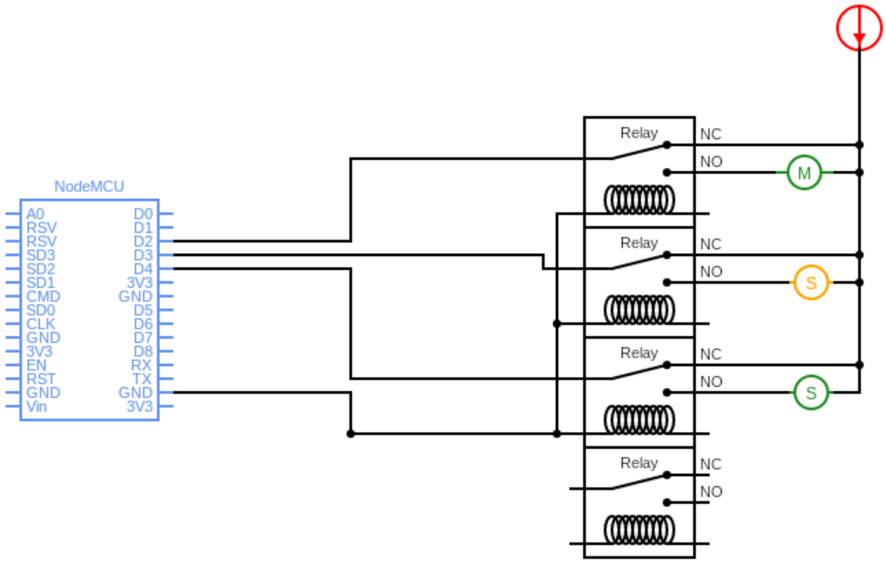


Figure 5.3: NodeMCU and Relay Modules connection

5.5. Conclusion

In this chapter, we have discussed the implementation of the whole system. The implementation is the process in which system is developed and then results are generated. In doing so, there are many algorithms applied, many methodologies are used, and these are applied to some tools for the processing.

Firstly, the strategy is discussed describing through what ways and means we will implement the project. The system requires tools for its implementation, so in our case as there are software and hardware sides the main tools contributing will be collaboratively working.

Furthermore, algorithms which are used are discussed for each module. As there will be specific technique for each module to perform so as there will be an algorithm defining the way. These algorithms will be followed by methods which are discussed. Methodologies include the basic mechanism to achieve the results. It includes circuit diagrams, different checks to observe the water states and state transitions through firebase. As firebase is the main data source, it is connected to the mobile application and the hardware equipment and flow of information takes place between them.

Chapter 6
System Testing & Evaluation

6.1. Test Strategy

The testing was done in different phases. We performed testing at each step by doing unit testing, component testing, integration testing and then to test the overall system we performed system testing. The components were tested at each phase where it is confirmed that how components can individually perform their work. Integration testing involves the testing in a hierarchical way. To test the whole system performance, we have tested the system at the final phase through which errors are removed.

6.2. Component Testing

In component testing, every single component of the system is tested separately. Error in every module is detected and removed and the overall functionality is verified. In our system, there are many components used which have to be integrated with the application. In our systems, we have applied many components which are tested individually before using them for the system. These systems combine to give an output. As in our case, there are software and hardware sides and each consists of components to be tested.

- Arduino is tested by supplying power to it and then uploading some basic code which tests its functionality to be feasible for our system.
- NodeMCU is used as a Wi-Fi module, by providing necessary power supply it is integrated with the Wi-Fi and connection is established,
- Ultrasonic water level sensors are used which is tested by integrating it with Arduino and retrieving the state of water.
- Relay module is used to convert states. It is tested by giving instructions from NodeMCU to change the state of any appliance using 220V.
- Motor is tested to work by turning in On or Off.
- Solenoid valve is used by applying it with water source and providing it power source.
- Firebase real time database is tested by connecting it with the mobile application and the hardware equipment.

6.3. Unit Testing

In unit testing we have we tested some functionalities of the modules. The smallest part of a system is called units and testing is performed on the units of the system.

Test Case No: TC01

QA Test Engineer: Hamza Mansoor

Objective: Water Level detection by sensor

System: Smart Water Management for Domestic Use

Step No.	Description	Result
1	Water level raises in the water tank	**Actual Output:** The water is detected and is variable **Expected Output:** The sensor detects the changing states of water

Comments: The test was successful

Test Case No: TC02

QA Test Engineer: Hamza Mansoor

Objective: Water states to mobile application

System: Smart Water Management for Domestic Use

Step No.	Description	Result
1	Water level change is sent to the mobile application in real time	**Actual Output:** The water states are sent to mobile application in real time **Expected Output:** The water level is detected and sent to mobile application

Comments: The test was successful

Test Case No: TC03

QA Test Engineer: Jahangeer Khan

Objective: Notifies

System: Smart Water Management for Domestic Use

Step No.	Description	Result
1	The system notifies the user at each action	**Actual Output:** The system gives the user a notification of the action performed **Expected Output:** The application sends a

		notification to user to notify the action

Comments: The test was successful

Test Case No: TC04

QA Test Engineer: Hamza Mansoor

Objective: Motor On/Off

System: Smart Water Management for Domestic Use

Step No.	Description	Result
1	System turns on/off the motor	**Actual Output:** The motor is on as soon as the application sends a command to on or off **Expected Output:** The motor turns on/off depending upon the user command sent from mobile application

Comments: The test was successful

Test Case No: TC05

QA Test Engineer: Jahangeer Khan

Objective: Adds schedule to list

System: Smart Water Management for Domestic Use

Step No.	Description	Result
1	User selects the day, date and duration of the motor to supply water	**Actual Output:** The user selects the day, date and duration and press add. The system responds with confirmation message and schedule is added in the list **Expected Output:** The system shows the option to select day, date and duration of water supply which on selecting is added

		to the list with the confirmation

Comments: The test was successful

Test Case No: TC06

QA Test Engineer: Hamza Mansoor

Objective: Real time visualization

System: Smart Water Management for Domestic Use

Step No.	Description	Result
1	The user views the current water states	**Actual Output:** The system shows the real time water states in the tanks **Expected Output:** The system navigates to the page where real time water states are visually displayed

Comments: The test was successful

Test Case No: TC07

QA Test Engineer: Hamza Mansoor

Objective: Application Updates firebase

System: Smart Water Management for Domestic Use

Step No.	Description	Result
1	The mobile application updates the firebase by sending a command to the cloud database	**Actual Output:** Firebase real time database is updated every time user sends a command from application **Expected Output:** When user sends a command from mobile application, firebase real time database is updated

Comments: The test was successful

Test Case No: TC08

QA Test Engineer: Jahangeer Khan

Objective: Sensors update the firebase

System: Smart Water Management for Domestic Use

Step No.	Description	Result
1	The sensors send the real time water states to mobile application through firebase real time database over Wi-Fi	**Actual Output:** The water level detected by sensors is detected and updates the firebase database through Wi-Fi **Expected Output:** The sensor detects the changing states of water and updates the database through Wi-Fi

Comments: The test was successful

Test Case No: TC09

QA Test Engineer: Jahangeer Khan

Objective: Wi-Fi connection

System: Smart Water Management for Domestic Use

Step No.	Description	Result
1	The user selects the Wi-Fi connections available on the mobile device	**Actual Output:** The Wi-Fi connection is established and is notified to the user **Expected Output:** The Wi-Fi connection is made after the user selects one of the available connections

Comments: The test was successful

6.4. Integrated Testing

After unit testing integration testing is done, in which the modules which have been undergone unit testing are then integrated and they are tested. The errors, problems and issues that are faced in the integration of these modules are noted and are removed.

6.4.1. Integration Test Strategy:

We have implemented top-down integration in which the testing flow runs from high level components and moves down to low level. Similarly, as in a tree where the top level is integrated first and then moves down to the lower level of the tree for integration. It has many advantages which includes that most of the functionality is already integrated and working when the prototype is released. The code is maintained easily and errors are eliminated before they are passed to the next level testing. While there are few disadvantages like the lower components are testing but not as much as the higher level components are tested because testing lower components becomes hard, still we have tried our best to make sure that lower level components are tested enough to satisfy the system requirements.

6.5. System Testing

Once the unit testing and integration testing is completed the full system is evaluated thoroughly. This is done to check that whether the system is working on what it is supposed to, keeping in mind the requirements. Also a basis to test the modules to work properly when integrated with the full system.

6.6. Test Cases

The test cases are generated as follows. We have tested the whole functionality of the system by making the test cases.

6.6.1. Test Case#1 Application Installation

Test Case ID	TC-01
Description	Successfully installing the application on a phone
Applicable	For Android Users only
Preconditions	The phone must be powered on and working
Steps	• APK file should be available on the phone's storage • Install the APK file on the phone

Expected Outcome	Application is successfully installed on the phone
Actual Output	Application is ready to use
Status	PASS

Table 6-1 Test Case 1

6.6.2. Test Case#2 Unsuccessful installation

Test Case ID	TC-02
Description	Installation unsuccessful
Applicable	For Android Users only
Preconditions	• Phone is turned on • App installed on an unsupported device
Steps	• APK file downloaded • Install on your device
Expected Outcome	Device prompts the user: “Device is not supported”
Actual Output	Device prompts the user: “Device is not supported”
Status	PASS

Table 6-2 Test Case 2

6.6.3. Test Case #3 Launching Application

Test Case ID	TC-03
Description	Launch the application
Applicable	For Android Users only
Preconditions	• Phone is turned on • App is installed on the device • Permissions are allowed
Steps	• After installation click on the application to use it
Expected Outcome	Application is launched successfully
Actual Output	Application is launched successfully
Status	PASS

Table 6-3: Test Case 3

6.6.4. Test Case #4 Login Successful

Test Case ID	TC-04
Description	Login the application
Applicable	For Android Users only
Preconditions	• Phone is turned on • App is installed on the device • Permissions are allowed
Steps	• After installation, the user opens the application • After loading screen, login page is displayed • User enters the credentials and clicks login
Expected Outcome	User is logged in successfully
Actual Output	User is logged in successfully
Status	PASS

Table 6-4: Test Case 4

6.6.5. Test Case #5 Connect Wi-Fi

Test Case ID	TC-05
Description	Connect Wi-Fi
Applicable	For Android Users only
Preconditions	• Phone is turned on • App is installed on the device • Permissions are allowed
Steps	• After launching application, the user selects the Wi-Fi connection
Expected Outcome	Wi-Fi connection is established
Actual Output	Wi-Fi connection is established
Status	PASS

Table 6-5: Test Case 5

6.6.6. Test Case #6 Add Schedule

Test Case ID	TC-06
Description	Connect Wi-Fi
Applicable	For Android Users only
Preconditions	• Phone is turned on • App is installed on the device

	• Permissions are allowed • User is logged in • Wi-Fi is connected
Steps	• After logging in, user selects the schedule option • User selects day, date and duration of water supply • User clicks on add schedule
Expected Outcome	Schedule is added to the list
Actual Output	Schedule is added to the list
Status	PASS

Table 6-6: Test Case 6

6.6.7. Test Case #7 Get current water levels

Test Case ID	**TC-07**
Description	Get current water levels
Applicable	For Android Users only
Preconditions	• Phone is turned on • App is installed on the device • Permissions are allowed • User is logged in • Wi-Fi is connected
Steps	• After logging in, user selects the check water states
Expected Outcome	Water tanks with their current states are displayed
Actual Output	Water tanks with their current states are displayed
Status	PASS

Table 6-7: Test Case 7

6.6.8. Test Case #8 Turn On Motor

Test Case ID	**TC-08**
Description	Turn on motor
Applicable	For Android Users only
Preconditions	• Phone is turned on • App is installed on the device • Permissions are allowed • User is logged in • Wi-Fi is connected
Steps	• After logging in, user tends to turn the motor on

	• System will check the main tank to check availability of water • System checks the current states of water tanks
Expected Outcome	Motor is turned on
Actual Output	Motor is turned on
Status	PASS

Table 6-8: Test Case 8

6.6.9. Test Case #9 Operate Valve

Test Case ID	TC-09
Description	Operate valves
Applicable	For Android Users only
Preconditions	• Phone is turned on • App is installed on the device • Permissions are allowed • User is logged in • Wi-Fi is connected
Steps	• After logging in, user selects the share water option • User selects the source and destination tanks
Expected Outcome	Valve is operated between tanks and water is shared
Actual Output	Water tanks with their current states are displayed
Status	PASS

Table 6-9: Test Case 9

6.6.10. Test Case #10 Update Profile

Test Case ID	TC-10
Description	Update Profile
Applicable	For Android Users only
Preconditions	• Phone is turned on • App is installed on the device • Permissions are allowed • User is logged in • Wi-Fi is connected
Steps	• After logging in, user selects the update profile from navigation bar • User enters the updated information and clicks confirm

Expected Outcome	Profile is updated on firebase
Actual Output	Profile is updated
Status	PASS

Table 6-10: Test Case 10

6.7. Results & Evaluation

The results show that almost every test which is performed has passed based on the inputs. This indicates and verifies the testability of the system. The system has many modules and components which are tested in component testing. Moreover, to further test the system using system testing which confirms that the system is less faulty based on required input.

6.8. Conclusion

In this chapter, we have performed the testing of our system. The testing was done in different phases. In first phase we have performed the component testing. In component testing, we have tested each component of the system individually before using it in the system and collaborating it with other components. Using this technique, we have identified the errors using testing and errors are removed individually before applying for the system.

In the next phase we have performed unit testing in which each module of the system is tested. As our system has limited modules so these are tested on the basis of their functionalities and errors are removed.

Furthermore, integration testing is performed. We have selected the bottom-up technique to do this kind of testing in which we performed the system testing at the lower level in the hierarchy, which means we have tested each module of the system and then combining these modules we have tested the system which is at the top level of the hierarchy.

The main part of the testing phase is the system testing. In this phase, we test the system overall and each function of the system is tested. In this testing, we observed the overall functionality of the system by providing some input and comparing the actual output with expected output. This helped to identify potential errors which are resolved.

Chapter 7
Conclusion

7.1. Contributions

In our final year project, we have implemented an idea which is comprised of many areas of research. We came up with this project keeping in view the common problem which water loss and management. The research area is IoT which also comes under Automation. We have applied many techniques to integrate our system as it contains two major section which carries out the whole process. There are numerous advantages of the system. The most important is resolving the problem of water loss which was our sole purpose before implementing this system.

While implementing the system, we kept in mind the already methods performed to attain certain information such as using ultrasonic sensor. In different projects, it was used according to their requirements. We made it possible to be used for our system. Another usage of solenoid valve which is used for water sharing, the valve is operated by 220V which is supplied by the relay module by disrupting the main power supply.

In the implementation process, we faced number of issues and problems. Out of which environment dependency was the biggest issue, because our system works on Wi-Fi and sometimes the Wi-Fi connection is not stable or in working condition. Moreover, living in such country where there is an issue of load shedding, the system is affected as it requires power supply for motor and valves even for microcontroller units as well.

7.2. Reflections

Although the system made flexible to some extend but still the research to make these more accurate is still being done throughout the world. The system we implemented has numerous advantages such as multisession, scheduling, automation of work, saving human workload by providing a working application which can be used remotely and innovation of the techniques used in the system.

This project can lead to many other ideas being implemented as very little work is done on this area in our country. As our country is facing a problem of water loss, so our system can be used for each domestic area or residence to conserve and manage water so that water can be saved for ourselves and the future.

To conclude, we have put a lot of effort in this project. To conclude, we have put a lot of effort in this project. From the beginning, the idea was our own and we have worked hard to complete every document, fulfilled every requirement by our department. By the supervision of our supervisor we have been able to progress in this project. This project can also be used in the future as an MS research by extending it for further application in automation, and can be made better and advance with the help of more research and more ideas integrated in it.

7.3. Future work

Finally discuss how this work can be extended in future work. This will normally include linking back to weaknesses, shortcomings or any open research questions that still need to be addressed.

The project idea is not limited to residential areas, such as homes, flats and hostels. It can be extended to be used in buildings or stores. As building, stores and malls also require water and there is a hastier routine their everyday so in order to manage the water supplies in these, our system can be extended for use.

The project we implemented is just based upon the water, furthermore, it can be also extended to gas management as well. The application can be extended by making some more options and applying gas sensors but using the same algorithm for gas usage in domestic areas. Solenoid valve and pipes can also be used for gas as well with the supply coming from the main line, with little adjustments this system can be extended. Research on this area is still under process and gas management can be added to our system for use.

REFERENCES

[1] Megha M Raykar, Parijata Vinod, Parinita Vinod, Preethi K M and Lovee Jain, "Automated Water Billing with Detection and Control of Water Leakage using Flow Conservation", Volume 3, Issue 2, Dept. of Computer Science And Engineering, NIE-institute of Technology, Mysore, India, 2015

[2] S.Gowri, Pola Pranathi, Kodali Sravya, "Automated Water Tank Overflow Control Unit Integrated with Mobile Application", Vol.9 No.2, Faculty of Computing, Dept. of IT., Sathyabama University, Chennai- 600119, India, July 2015

[3] Souvik Paul , Mousumi Das, Anik Sau , Soumyadeep Patra, "Android Based Smart Water Pump Controller With Water Level Detection Technique", Vol. 4, Issue, 2015

[4] Mazharul Islam Nayeem, Mahfida Amjad, "Water Automation for Water PumpController using Android Application – Review", Volume 182 – No. 29, Stamford University Bangladesh Dhaka, Bangladesh, November 2018

[5] Sushil S.Patil , Varsha D. Nikam, "AUTOMATION IN FARMING USING ANDROID APPLICATION", Computer Science and Engineering Department, DMGOI, Vikaswadi (Kagal), 2016

[6] Jia Uddin, S.M. Taslim Reza, Qader Newaz, Jamal Uddin, Touhidul Islam, and Jong-Myon Kim, "Automated Irrigation System Using Solar Power", Dhaka, Bangladesh, 2012

[7] Pooja P. Karande, Prachi N. Sawardekar, Pooja B. Patil, Z.J.Tamboli, "Study Of Arduino For Irrigation Based Control Using Android App", Volume: 04 Issue: 1, Department of Electronics and Telecommunication Engineering Sanjay Ghodawat Institutions, Atigre, 2017

[8] "Design of Automatic Watering System Based on Arduino", Vol. 1, No. 2, 2020

[9] Yuliana Astutik, Murad, Guyup Mahardian Dwi Putra and Diah Ajeng Setiawati, "Remote monitoring systems in greenhouse based on NodeMCU ESP8266 microcontroller and Android", 2019

[10] Viktor SZENTE, Gábor LÓRÁNT, "Computational and Experimental Investigation onDynamics of Electric Braking Systems", Department of Fluid Mechanics Budapest University of Technology and Economics Bertalan Lajos u. 4 - 6., H - 1111 Budapest, Hungary

[11] Broadcast Receivers: https://developer.android.com/guide/components/broadcasts Accessed Date: 3-1-2021

[12]Services in android: https://www.tutorialspoint.com/android/android_services.html Accessed Date: 12-1-2021

[13] Iotguider, "Serial Communication between two Arduino Boards", 2020, https://www.youtube.com/watch?v=9HivniieLvI&t=248s Accessed Date: 15-2-2021.

[14] Aniket Nikam, Nisha Warhade, Rohit Dhawale, Siddhant Prabhu, Ganesh Deshmukh, "Fully Automated System for Monitoring Water Usage using SMS and Android Application", Volume: 04, Dept. of Computer Engineering, PCCoE, Maharashtra, India, 2017

[15] ArdunioUno: https://i1.wp.com/www.teachmemicro.com/wp-content/uploads/2018/06/arduino_uno_r3_top.jpg?zoom=2.625&resize=410%2C205&ssl=1 Accessed Date: 12-12-2020.

[16] Arduino Pin Configuration: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSoSbWKjb5rTQJIyMDHK01L8Vj5hblvaQn-yA&usqp=CAU Accessed Date:12-12-2020

[17] Node MCU: https://imgbin.com/png/Dk6tsRVF/nodemcu-esp8266-wi-fi-lua-usb-png Accessed Date:12-12-2020.

[18] Node MCU Pin Configuration: https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet Accessed Date:12-12-2020.

[19] Ultrasonic : https://www.pngegg.com/en/png-pftry Accessed Date:12-2- 2021.

[20] Relay Module: https://circuits-diy.com/8-channel-5volt-relay-module/ Accessed Date:14-2- 2021

APPENDICES

Appendix A

Costing of our project is as follows:

S. No.	Equipment / Components	Price (PKR.)
1.	Arduino x2	1300
2.	NodeMCU	650
3.	Motor x 2	1144
4.	Relay Module	1050
5.	BreadBoard+Wires	2000
6.	UltraSonic Sensors x 4	880
7.	Smart Valves (Required) x 2	6000
8.	Tanks x 4 (Required)	4000
Total		**17,024**

Appendix B

Following is the datasheet of Ultrasonic sensor HC- SR-04

Vcc- Connects to 5V of positive voltage for power

Trig- A pulse is sent here for the sensor to go into ranging mode for object detection

Echo- The echo sends a signal back if an object has been detected or not. If a signal is returned, an object has been detected. If not, no object has been detected.

GND- Completes electrical pathway of the power.

Plagiarism Report

FYP

ORIGINALITY REPORT

10%	5%	1%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	3%
2	Submitted to University of Greenwich Student Paper	2%
3	www.slideshare.net Internet Source	<1%
4	Submitted to Middlesex University Student Paper	<1%
5	Submitted to Loughborough University Student Paper	<1%
6	Submitted to UT, Dallas Student Paper	<1%
7	Submitted to Nottingham Trent University Student Paper	<1%
8	Submitted to University of Keele Student Paper	<1%
9	irl.umsl.edu Internet Source	<1%