UNIVERSITY OF DAR ES SALAAM

COLLEGE OF INFORMATION AND COMMUNICATION TECHNOLOGIES

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CS 499- FINAL YEAR PROJECT

PROJECT TITLE: DESIGN AND IMPLEMENTATION OF DAWASA WATER SUPPLY REGULATION SYSTEM

Project Report in Partial Fulfillment for the Award of Bachelor of Science in Computer Engineering and Information Technology

NAME OF CANDIDATE: MERO SAMSON R

REGISTRATION NUMBER: 2015-04-02456

SUPERVISOR: DAUDI CHARLES

SUPERVISOR'S SIGNATURE: ……

SUBMISSION DATE: 24th June 2019

STATEMENT OF AUTHORSHIP AND ORIGINALITY

I, MERO, SAMSON R declare that this report and the work described in it are my own work, with any contributions from others expressly acknowledged and/or cited.

I declare that the work in this report was carried out in accordance with the Regulations of the University of Dar es Salaam and has not been presented to any other University for examination either in Tanzania or overseas.

Signature: ……

Date: ……

This report may proceed for submission for assessment for the award of B.Sc. in Computer Engineering and Information Technology at the University of Dar es Salaam.

ABSTRACT

Water usage tends to vary from time to time throughout the day, it tends to be high in the morning when people are preparing themselves to go to their workplaces and at certain times in the afternoon but becomes low during the night when people are sleeping. This necessitates the need to control water supply based on users' demand. This is one of the challenges that DAWASA as the authority that is responsible for distributing water services is facing. When the supply is too high it leads to the bursting of pipes which in turn leads to much losses in terms of both water and energy.

The project aims at developing a system that will help DAWASA regulate their supply based on demand. The system will be able to measure demand on the distribution network and adjust the supply accordingly. This will enable DAWASA to reduce losses in terms of water and energy.

This report does not cover the whole project since the project work is required to be covered in two semesters, rather the report gives an introduction which provide a background to the work. Also the report includes the description of the problem statement, objectives, literature review, methodology used to solve the problem, work done so far, time schedule and the budget for the project.

The designing and implementation process in this project will be covered in the final project report. This part will be accomplished through literature review, data collection and analysis and consultation from the supervisor and other college staffs.

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to the administration of UDSM COICT for providing me with an opportunity to do this wonderful project, which is under the challenge driven education. Special thanks goes to Dr. Kissaka who is the principal of the college for all of his support and contributions towards the project.

I sincerely thank my final year project supervisor Mr. Daudi Charles for all of his support, guidance and the inputs he gave me towards making this project successful. I also thank Madam Mwase, Madam Eva and Mr. Ishengoma our CDE coordinators for their commitment and hardworking in guiding us throughout this project.

Furthermore, I wish to appreciate the contribution made by other supervisors and extend my gratitude to the panel of supervisors for their valuable advice and comments that have improved the project goals.

Last but not least, I thank my fellow group member Kamando, Frank whom I worked with in this project. I thank him for his commitment, positive attitude and hardworking.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CDE	Challenge Driven Education
COICT	College of Information and Communication Technologies
CSS	Cascaded Style Sheet
DAWASA	Dar es salaam Water and Sewarage Authority
DAWASCO	Dar es salaam Water and Sewarage corporation
HTML	Hypertext Makeup Language
PHP	Hypertext Preprocessor
UI	User interface
UML	Unified Modelling Language

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Introduction

DAWASA is a government authority that is responsible for generating and distributing water and sewerage services in Dar es salaam, Pwani and some parts of coastal region such as Bagamoyo and Chalinze. DAWASA as a new body formed from the merging of the previous DAWASCO and DAWASA owns the infrastructures and performs all the operations needed to ensure the society gets proper and reliable water and sewerage services. In the course of its operations DAWASA encounters a number of challenges that cause it not to effectively serve the society the way it is supposed to. These challenges leads to much losses such as water and energy losses causing the society to lack reliable water and sewerage services.

As an academic institution, the University of Dar es salaam in partnership with Dar es salaam institute of technology (DIT) and KTH University in Sweden decided to undertake a project under challenge driven education (CDE) which tries to address some of these challenges. The challenges which were presented by DAWASA were lack of Infrastructure and Asset Management System, Inefficient Customer Care, billing and revenue Collection, Low sanitation coverage and Access to good quality water in unplanned areas see figure 1-1. The challenges were distributed to groups of students for which each group was given one challenge. Later on the challenges were further divided into sub-challenges and students' groups divided into sub-groups such that each sub-group was assigned one sub-challenge. In this project I am addressing lack of Infrastructure and Asset Management System challenge under the sub-challenge lack of efficient demand based water distribution system. See figure 1-2.

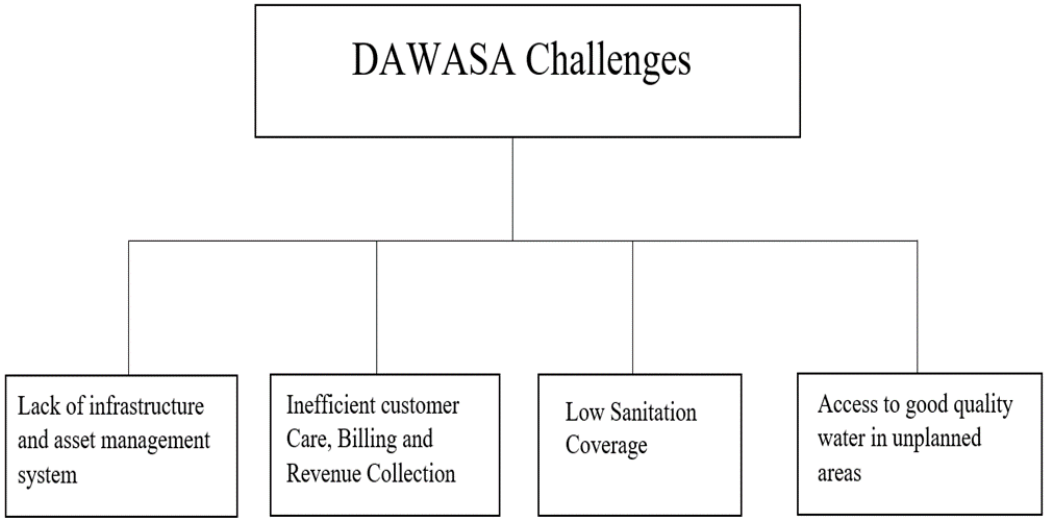

Figure 1-1: Main challenges

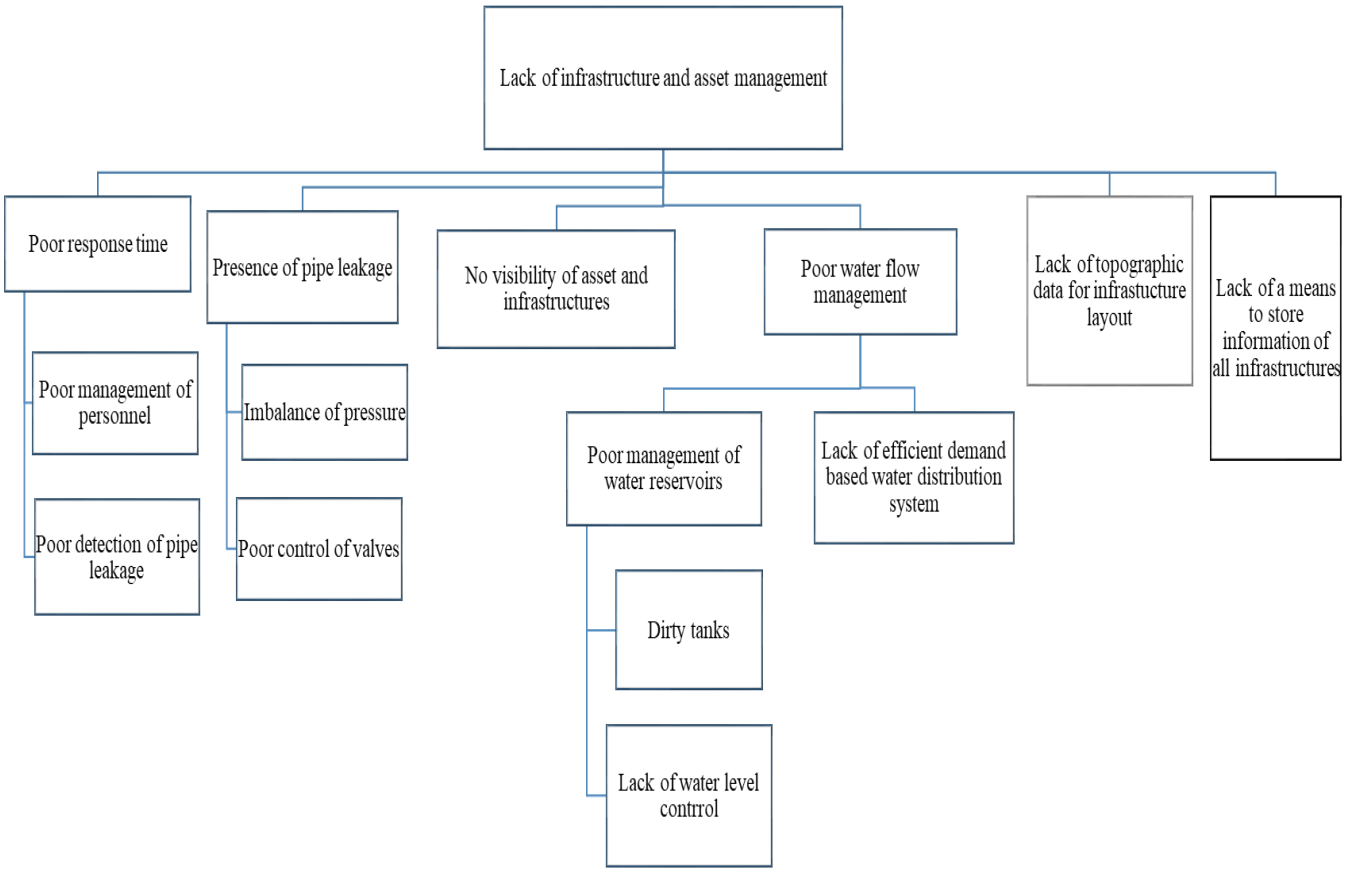

Figure 1-2: Challenge breakdown

1.2 Statement of the problem

Water demand in the distribution network is not a fixed phenomenon, it tends to vary from time to time throughout the day. At some point it is high like in the morning when people are preparing themselves to go to their workplaces and at some other point it is low for example in the night when people are sleeping. Therefore, supplying the same amount of water throughout the day as if the demand is constant could lead to lots of losses in terms of both water and energy. Furthermore, it could lead to bursting of pipes in cases when the supply is higher than the demand due to high amount of pressure accumulating in the pipes. Therefore, there got to be a way to regulate supply in accordance with the varying demand in the distribution network. This calls for an automated system that will be able to detect the varying demand in the distribution network and adjust the supply accordingly.

1.3 Objectives

1.3.1 Main objective

To ensure water supply to customers is based on demand that is the supply should be high when there is high demand and it should be decreased when the demand decreases.

1.3.2 Specific objectives

a) To collect and analyze system requirements.
b) To design and implement an interface to the system.
c) To design and implement a database for the system.
d) To design and implement a mechanism to receive input from the sensors and flow meters.
e) To design and implement a mechanism to send control signals to the actuators.

1.4 Significance of the project

Accomplishing the goals of this project will help DAWASA regulate supply based on needs that is supply amount of water that is actually needed and not otherwise. This will help to reduce unnecessary losses in terms of both energy and water. On the other hand, customers will be ensured with reliable and enough water supply as water loss would be minimized.

1.5 Scope and limitation

The project aims at developing a system that will help DAWASA regulate supply based on demand in the distribution network. The system is developed for and will be used only by DAWASA and not any other water authority in this country or any other country.

CHAPTER TWO

LITERATURE REVIEW

2.1 Literature review

This part gives a description of the existing system that is used to solve the challenge and its weakness. Also it gives a brief description of the proposed system.

2.1.1 Existing system

Currently DAWASA are able to regulate supply based on demand mainly through experience. They know for how long they should operate the pumps to ensure their customers get enough water, they know times of the day when the demand is low and times when the demand is high. This method is mainly dependent on how efficient their workers are as a minor mistake of not shutting down the treatment plant at the required time could lead to much losses in terms of water and energy.

Weaknesses of the system

a) It is costful as a large number of people have to be employed to make sure everything goes well
b) Human weakness. Human beings are not perfect as machines and therefore they can make mistakes which could lead to much losses in terms of water and energy.
c) Retirement of experienced workers is a problem especially if it happens they don't pass the experience/knowledge to the new workers
d) If number of customers in the distribution network increases, it may take some time to be able to adjust to the new demand as there is no a real time system that detects changes in demand.

2.1.2 Related work

Chaos Logic-based Demand Prediction Method

Is a control method implemented by MEIDENSHA corporation in Japan in which fluctuation of water distribution volume is absorbed as fluctuation of water level at the water reservoir. Future water level of water reservoir is predicted by this method and water supply amount is automatically

controlled to prevent deviation of water level from preset level. (Automated Water Water Supply Control System (Chaos Logic-based Demand Prediction Method)).

2.1.3 Proposed system

The proposed solution is to design an automatic system that will control water supply based on demand in the distribution network. The solution will involve the measurement of flowrate in the main outlet pipes of reservoirs and at each branch in the distribution network whose readings will be used to adjust the level of water "H" in the reservoirs, See figure 2-1. As the water level in the reservoirs is altered it will alter the pressure as they are directly proportional which in turn will adjust the supply. This is based on the fact that high flow rate would mean the demand is high and low flow rate the demand is low. All this will be made possible with the aid of a water supply regulation system (a web application) which will be taking in data from the sensors and flow meters and based on these data the system will make decisions and give some outputs which upon being sent to the actuators will alter the supply accordingly. See figure 2-2.

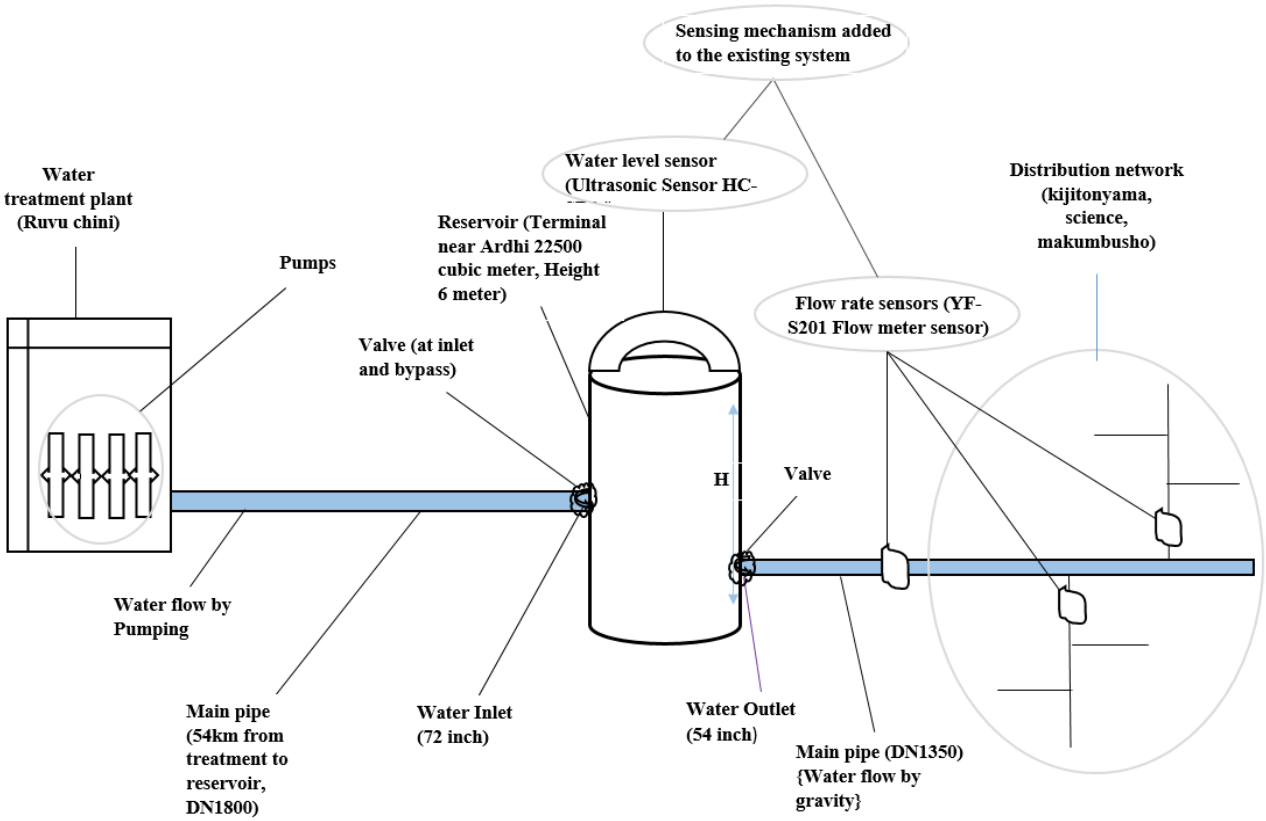

Figure 2-1: Proposed system

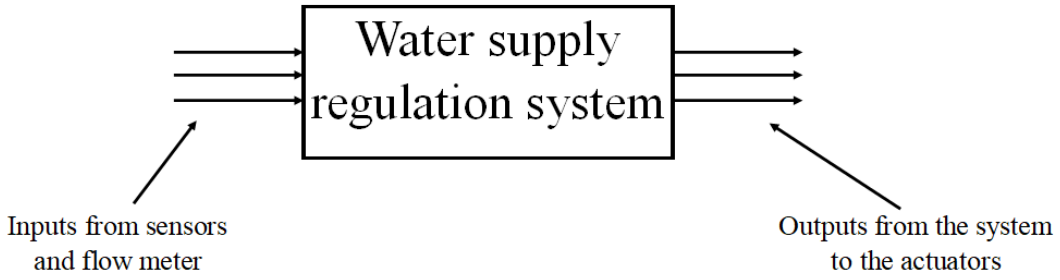

Figure 2-2: Water supply regulation system

CHAPTER THREE

METHODOLOGY

3.1 Methodology

The implementation of the solution will be divided into two parts, hardware and software parts. The hardware part which will be done by my fellow student will mainly be about designing and implementation of the overall circuit, sensors and flow meters that will collect data and send it to the system for manipulation and decision making process. The software part, the part that I will be dealing with, will be the development of that system (water supply regulation system). The following are approaches that will be used in order to develop the system:

3.2. System development

This system development will observe the System Development Life Cycle (SDLC) process which Waterfall Model will be used, see figure 3-1. The selection of the waterfall model was made due to the following advantages:

a) The requirements are clearly and accurately stated, they remain unchanged throughout the entire project development;
b) Detailed documentation of each development stage provides resistance to changes in human resources – a new developer can quickly get all the necessary information;
c) Careful planning of the project development structure reduces the number of problematic issues;
d) The start and end points for each phase are set, which makes it easy to measure progress;
e) The tasks remain as stable as possible throughout the development process;
f) It provides easy control and transparency for the customer due to a strict reporting system;
g) Release date for the finished product, as well as its final cost can be calculated prior to development. (Waterfall Methodology: Advantages, Disadvantages And When to Use It?, 2018)

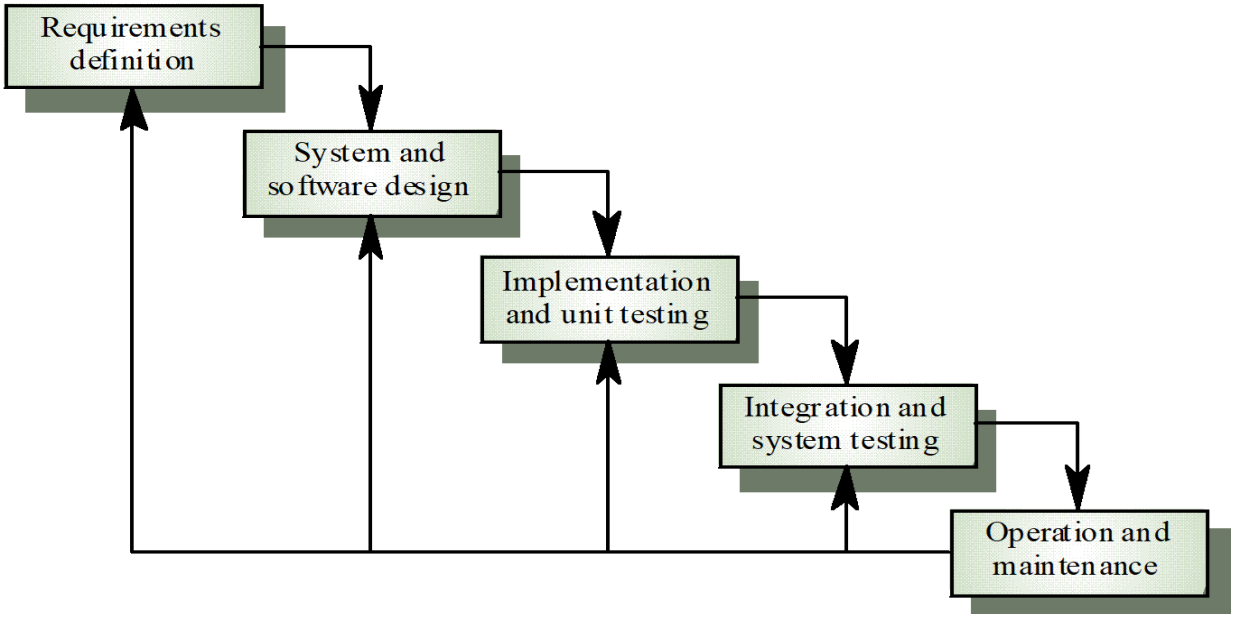

Figure 3-1: Waterfall development model

3.3 System architecture

DISTRIBUTED ARCHITECTURE

The implementation of the system will use distributed architecture where by at each zone/ location there will be a combination of Arduino and raspberry pi where by the Arduino will be interfaced with the sensors and data obtained from the sensors will be sent to the raspberry pi for local processing. After local processing at the raspberry pi some of the required data will be sent to the software system for either further processing or for storage. See figure 3-2.

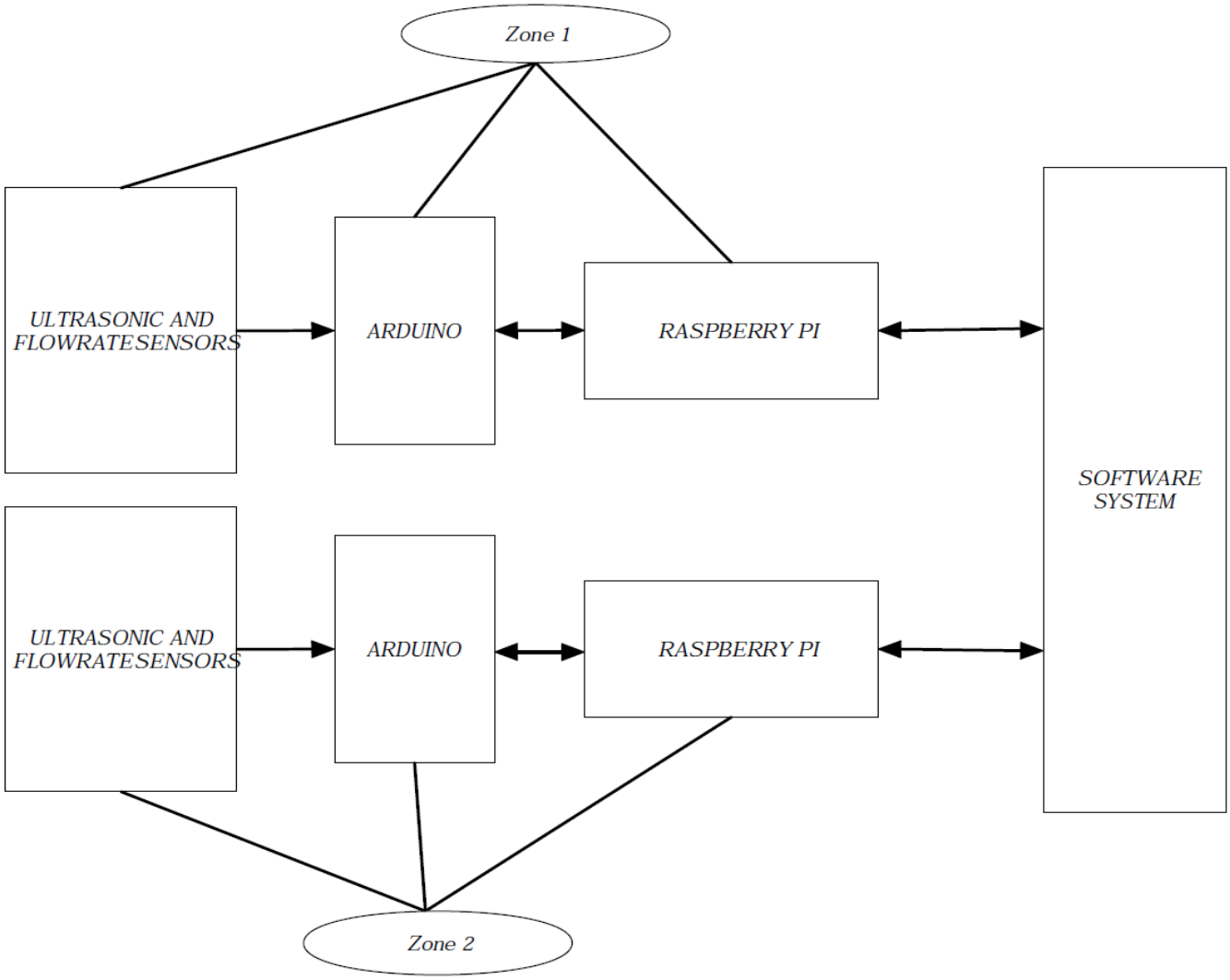

Figure 3-2: Distributed architecture

3.4 System Technologies

This project will use the following web technologies:

a) Web Application development using PHP (laravel framework), CSS, JavaScript and HTML
b) Database Design and Implementation using MYSQL WORKBENCH
c) Object Oriented (OO) system modeling with UML

Advantages of using Laravel PHP framework:

a) **Creating authorization and authentication systems.** Every web app owner needs to make sure that users are who they claim to be and therefore prevent unauthorized users to access the secured resources. Laravel makes authentication implementation very straightforward. Nearly everything is configured out-of-the-box. The framework also provides a simple way of organizing authorization logic as well as control resources access.

b) **Integration with tools to make faster web apps.** It is necessary not only to build an app but to create a fast one as well, which would result to revenue acceleration. Integration with caching back end often is one of the major steps taken to boost the performance of a web app. Laravel supports popular cache back-ends, such as Redis and Memcached out-of-the-box. Through default, the framework is configured to use file cache driver that stores cached objects in a file system. It is even possible to configure numerous cache configurations.

c) **Mail services integration.** Mail service could be used for sending notifications to emails of users after different events. It is difficult to imagine a modern web app, for instance which lack even a simple email notification to a new user regarding their successful registration on a site. The framework offers a simple, clean API over the famous SwiftMailer library. It also provides drivers for Mailgun, SMTP, Mandrill, Amazon SES and more, enabling an app to quickly begin sending email via a local or cloud-based service. It provides support for sending notifications across various delivery channels, such as SMS and Slack.

d) **Handling exception and configuration error.** The manner in which a software app handles errors could have a huge impact on user satisfaction as well as an app's usability.

The absence of proper error handling, a user may decide that the issue is in the app and move away from it forever. Nobody wants to lose customers or disappoint them. With Laravel, error and exception handling is configured already for any new Laravel-based project. Additionally, it is integrated with the Monolog logging library that offers support for various powerful log handlers.

e) **Automation testing work.** Without testing and verifying what a developer has built, in what way could anyone make sure that the software would do without any bugs, errors, crashes and in adherence to the initial app requirements? Automation testing is less time consuming and in most instances more exact compared to a manual one. Laravel is created with testing in mind. As a matter of fact, support for testing with PHP Unit is included out-of-the-box. It also ships with convenient helper methodologies, enabling expressive testing of apps.

f) **Separation of business logic code from presentation code.** Such separation allows HTML layout designers to easily change the look of a webpage without interacting with developers. Possible feature requests and bug fixes will be done through developers faster if all programming code of the app has the right separation at the early part of the development phase. Since Laravel is an MVC framework, separation is done already.

g) **Fixing most common technical vulnerabilities.** Vulnerabilities in security go hand-in-hand with developing any web app. OWASP Foundation, a non-profit American organization describes the most vital web app security vulnerabilities, which include cross-site request forgery, SQL injection, cross-site scripting and more. A developer should not forget and should be able to fix all before delivery. Laravel helps secure the web app through shielding it against the most serious security dangers. The code base is guarded fanatically, and the code has been vetted by several individuals.

h) **Scheduling tasks configuration and management.** Whether it's important to send out emails to subscribers every morning or automatic database tables cleanup at the end of the day, any web application needs a task scheduling mechanism to handle the tasks when it's time. Before, developers generated a Cron entry for every tasks they have to schedule, but this is a headache. The task schedule is not in source control anymore, and developers should SSH into the server to add Cron entries. The command scheduler of Laravel allows

expressive and fluent defining of command schedule within the framework itself, and only one Cron entry is required for the server. (Advantages of using the Laravel framework)

Advantages of using Mysql:

a) **It's Easy to Use**

MySQL is very easy to install, and thanks to a bevy of third-party tools that can be added to the database, setting up an implementation is a relatively simple task. In addition, it's also an easy database to work with. So long as you understand the language, you shouldn't run into too many problems.

b) **Support Is Readily Available Whenever Necessary**

Although Oracle's history of supporting its customers can be spotty at best, the nature of MySQL – which got its start as an open-source platform – means that there's a large and thriving community of developers and enthusiasts to which one can turn for help. This is due in large part to the popularity of the solution, the end result of which is no shortage of experts.

c) **It's Open-Source (Sort Of)**

Oracle's purchase of Sun Microsystems (and by association, MySQL) was met with some contention from the development community. The general fear was that Oracle would transform the tool into a closed, proprietary ecosystem. Thankfully, though Oracle has tightened its grip on MySQL somewhat, it can still be considered an open-source database option, as the code is still available for free online.

d) **It's Incredibly Inexpensive**

Depending on what you plan to use it for, a MySQL implementation could range in price from free to $10,000 or more. Either way, it's significantly less expensive than most other database options on the market (save for MySQL's open-source competitors).

e) **It's an Industry Standard (And Still Extremely Popular)**

Although MySQL's popularity has waned somewhat in recent years, it remains one of the most-used database systems in the world. It's compatible with virtually every operating system, and is more or less an industry standard. This is, of course, in spite of all the folks who say it's on the way out. (Mack, 2014)

CHAPTER FOUR

REQUIREMENTS CAPTURE AND ANALYSIS

4.1 Requirements capture

Requirements are descriptions or statements describing functions that the system must perform. These are the features that a user is looking forward to see once the system is complete. In this project requirements have been categorized into two categories functional and non-functional requirements.

4.1.1 Functional requirements

Are the functions that relates directly to the functioning of the system. These are aspects of the system the client is mostly likely to recognize. See table 4-1.

Table 4-1: Functional requirements

Ref. #	**Description**		**Category**
F1	Interfacing with the microcontroller		
	F1.1	The system should be able to receive data from the microcontroller such as flowrate and water level in the reservoirs	Hidden
	F1.2	The system should be able to send data/control signals to the microcontroller	Hidden
F2	Data processing		
	F2.1	The system should be able to give corresponding value of height h (water level in the reservoirs) based on the received flow rate.	Hidden
	F2.2	The system should be able to calculate the required number of pumps to meet the proposed water level h of the reservoir.	Hidden
	F2.3	The system should be able to calculate the overall water supply on daily basis based on the received flow rate.	Hidden
F3	The system should have a mechanism to store the calculated data for the daily water supply.		Evident

F4	The system should allow the user to see all data (report) for the daily supply based on the specified interval of time.	Evident
F5	The system should be able to display system status such as current water level in the reservoir, number of pumps that are currently operating etc.	Evident
F6	The system should allow the user to manually operate the system if he/she wishes to do so.	Evident
F7	The system should be able to give alerts for the case of any abnormalities in the system.	Evident
F8	The system should allow the user to log in and out of the system	Evident
F9	The system should allow the system admin to register users of the system	Evident
F10	The system should allow the system admin to deactivate users of the system whenever it is necessary to do so.	Evident

4.1.2 Non-functional requirements

These are requirements that define how a system is supposed to behave. They are qualities of the system. See table 4-2.

Table 4-2: Non-functional requirements

REF#	ATTRIBUTE	CONSTRAINTS
R1	Security	The system should authenticate all users before allowing them to interact with its functionalities.
R2	Response time	The system response time should be small not exceeding 5 seconds.
R3	Usability	The system shall be easy to use and self-explanatory such that the operator can be able to operate the system within 30 minutes of encountering the system for the first time.
R4	Compatibility	Accessible in all web browsers

R5	Availability	The system shall be available for use all the time that is for 24 hours and for all days of the week.

4.2 Requirements analysis

This is the most important stage to be taken in order to provide the clear description and understanding of the system's functional and nonfunctional requirements. It defines functional and non-functional requirements to a level of detail sufficient for system design using use case diagrams and use case descriptions.

4.2.1 Use case diagram

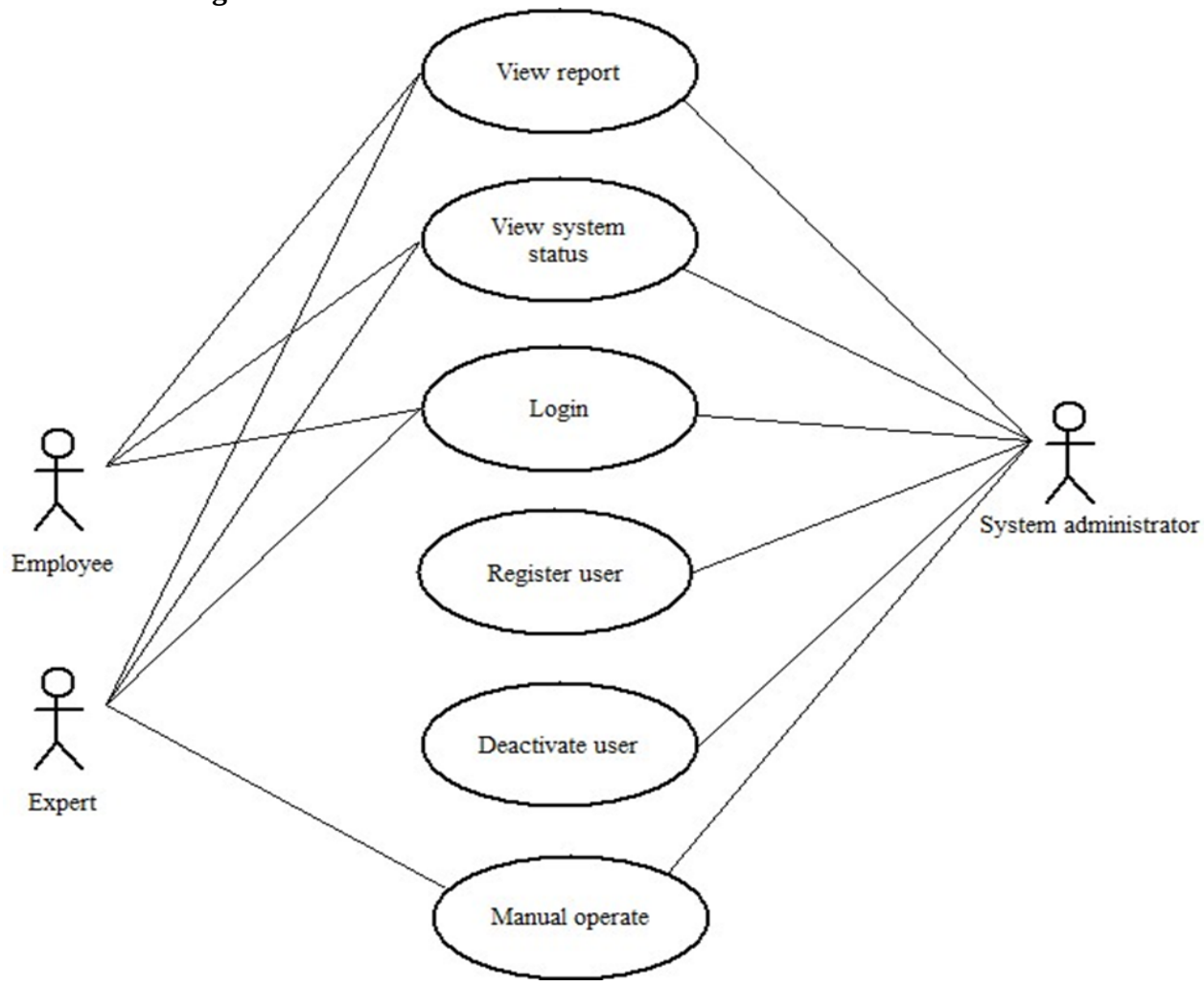

Figure 4-1: Use case diagram for the system

Table 4-3: Description of the actors of the system

No.	ACTOR	DESCRIPTION	USE CASE
1	Employee	This is a normal DAWASA employee who can log in to the system and have access to some of the functionalities but not all	view report, view system status and login
2	Expert	This is also a DAWASA employee but has more technical expertise or has higher position as compared to a normal employee such that he/she is given access to more system functionalities such as being able to manually control the system.	view report, view system status, login and manual operate
3	System administrator	This is also a DAWASA employee but who has specialized in the IT sector he/she is given access to all system functionalities	view report, view system status, login, manual operate, register user and deactivate user.

4.2.2 Use case descriptions

Table 4-4: View report use case description

Field	Description
Use case	View report
Actors	System administrator, employee and expert
Short Description	Allow actors of this use case to view daily water supply report
Pre-condition	Report data must have already been stored in the database, the interval of time for the report must have been specified by the user.
Post-condition	The actor will be presented with daily water supply report based on the specified interval of time
Main flow	The actor log in to the system The system provides the actor with a list of menu items to choose from The actor selects “reports” menu item The system provides the actor with a form to fill in the interval of time for the report to be requested The actor fills in the interval and submit the form The system presents the actor with the requested report.
Alternative flow(s)	-
Exception flow	5. If the actor enters invalid interval the system gives error message requesting the user to repeat again

Table 4-5: View system status use case description

Field	Description
Use case	View system status
Actors	System administrator, employee and expert
Short Description	Allow actors of this use case to view status of the system such as current flowrate, current water level in the reservoirs etc.
Pre-condition	Data for the system status must be available in the system
Post-condition	The actor will be presented with the status of the system
Main flow	The actor logs in to the system The system provides the actor with a list of menu items to choose from The actor selects “system status” menu item The system presents the actor with the current status of the system.
Alternative flow(s)	-
Exception flow	4. If data for the system status is not for any reason available, the system presents the actor with an error message.

Table 4-6: Login use case description

Field	Description
Use case	Login
Actors	System administrator, employee and expert
Short Description	Authenticate the actor before he/she can be allowed to access system functionalities
Pre-condition	Actor must have been registered in the system
Post-condition	The actor will be given access to system functionalities
Main flow	The system provides the actor with a login form. The actor fills in the form with his/her credentials and submits the form. The actor is directed to the homepage of the system.
Alternative flow(s)	-
Exception flow	2. If the entered credentials does not match with the existing records for that actor or if the user enters invalid credentials the system will give error message requesting the actor to try again

Table 4-7: Register user use case description

Field	Description
Use case	Register user
Actors	System administrator
Short Description	Allows the actor to add new users to the system
Pre-condition	The actor must have been registered in the system as an administrator
Post-condition	New user(s) will be added to the system
Main flow	The system presents the actor with a login form The actor enters his/her login credentials and submits the form The actor is redirected to administrator dashboard where he/she is presented with various options to choose from The actor selects “Add user” option. The system provides the actor with a form to fill in details of the new user. The actor fills in the form with credentials of the new user and submits the form The new user is added successfully into the system.
Alternative flow(s)	-

Exception flow	6. If the actor does not fill in all the required fields in the form the system gives an error message requesting him/her to fill in all the required fields.

Table 4-8: Deactivate user use case description

Field	Description
Use case	Deactivate user
Actors	System administrator
Short Description	Allows the actor to deactivate users so as they may no longer be able to use system functionalities.
Pre-condition(s)	For a user to be deactivated he/she must have been registered in the system before
Post-condition	User(s) will be deactivated
Main flow	The system presents the actor with a login form The actor enters his/her login credentials and submits the form The actor is redirected to administrator dashboard where he/she is presented with various options to choose from The actor selects “Deactivate user” option. The system provides the actor with a list of all the users of the system The actor selects the desired user and clicks “deactivate user” button The selected user is deactivated and can no longer have access to system functionalities.

Alternative flow(s)	-
Exception flow	6. When the actor clicks the “deactivate user” button the system asks the actor if he/she is sure of his/her decision if yes the selected user will be deactivated otherwise the process is terminated.

Table 4-9: Manual operate use case description

Field	Description
Use case	Manual operate
Actors	System administrator and expert
Short Description	Allow actors of this use case to manually operate the system when there is a need to do so. For example the actor will be able to alter number of pumps operating or closing some valves.
Pre-condition	The actors must have been registered as either system administrator or the expert for this functionality to be accessible to them
Post-condition	The actor will have achieved manual control of the system

Main flow	The actor logs in to the system. The system provides the actor with a list of menu items to choose from. The actor selects “system status” menu item The system presents the actor with the current status of the system and a “manual control” button (This button does not appear if the actor is not either the expert or the system administrator). The actor clicks the “manual control” button. The system gives the actor the ability to alter some parameters of the system. The actor alters the parameters of the system according to his or her need and confirms the changes made. The system responds accordingly.
Alternative flow(s)	-
Exception flow	7. If the actor upon altering the system parameters does not confirm or clicks the “cancel” button all changes made will be discarded and the system will continue to run in the same manner it was running before the attempt to alter its operation.

CHAPTER FIVE

SYSTEM DESIGN

5.1 System design

System design is a process of converting users' needs into a suitable form, which helps the programmer in software coding and implementation. This is done after the purpose and specifications of the system have been determined.

5.2 Sequence diagram

A sequence diagram is an interaction diagram that details how operations are carried out, what messages are sent and when. Sequence diagrams are normally organized according to time. The time progresses as you go down the page. The objects involved in the operation are listed from left to right according to when they take part in the message sequence.

5.2.1 Login sequence diagram

The software system has a login section. This section is for all users whose credentials are in the database. When a user provides the correct username or email and password, the system will verify if the entered credentials matches with the credentials stored in the database.

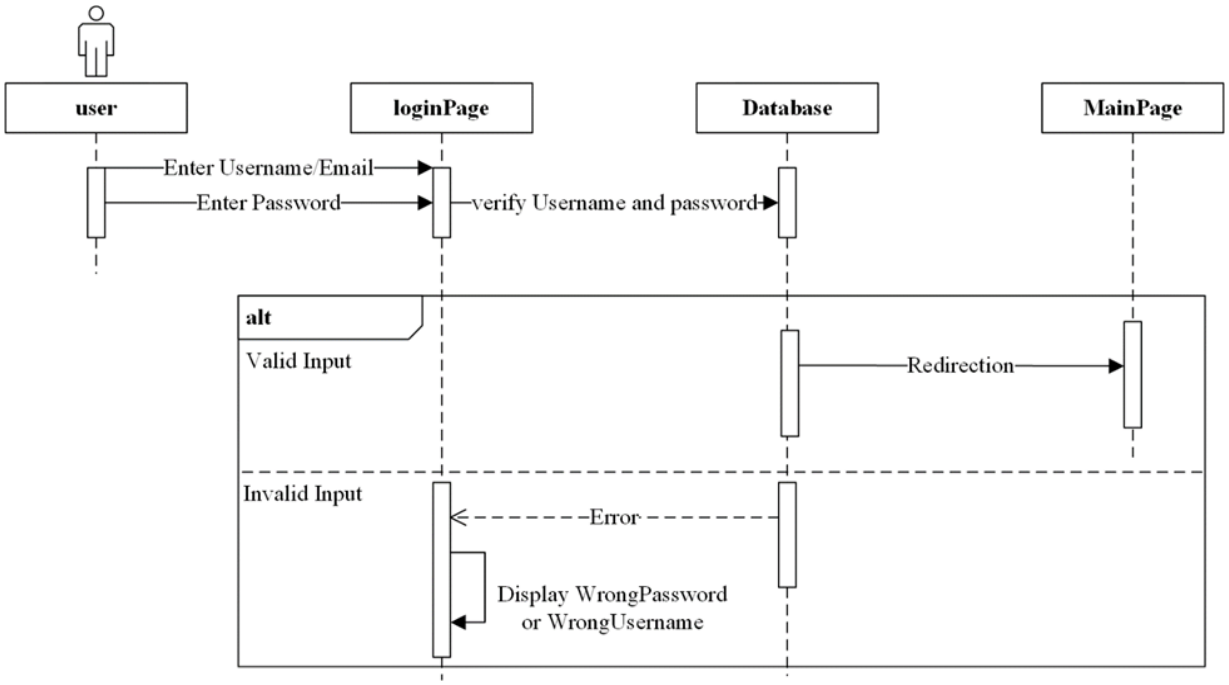

Figure 5-1: login sequence diagram

If they match, the system will direct the user to the main page (Homepage) of the system. If the username or email doesn't match, the system will display an error to the user as shown in figure 5-1.

5.2.2 User registration sequence diagram

The system has a register section, in which the user, in this case, the administrator needs to register another system user and assign them permission to the system. The administrator before registering a user needs to have user's credentials such as username and other credentials. The username for the user needs to be unique. This means that the username should not match with any other username in the database. If the username does not match with the ones already stored in the database, the user will be successfully registered. See figure 5-2.

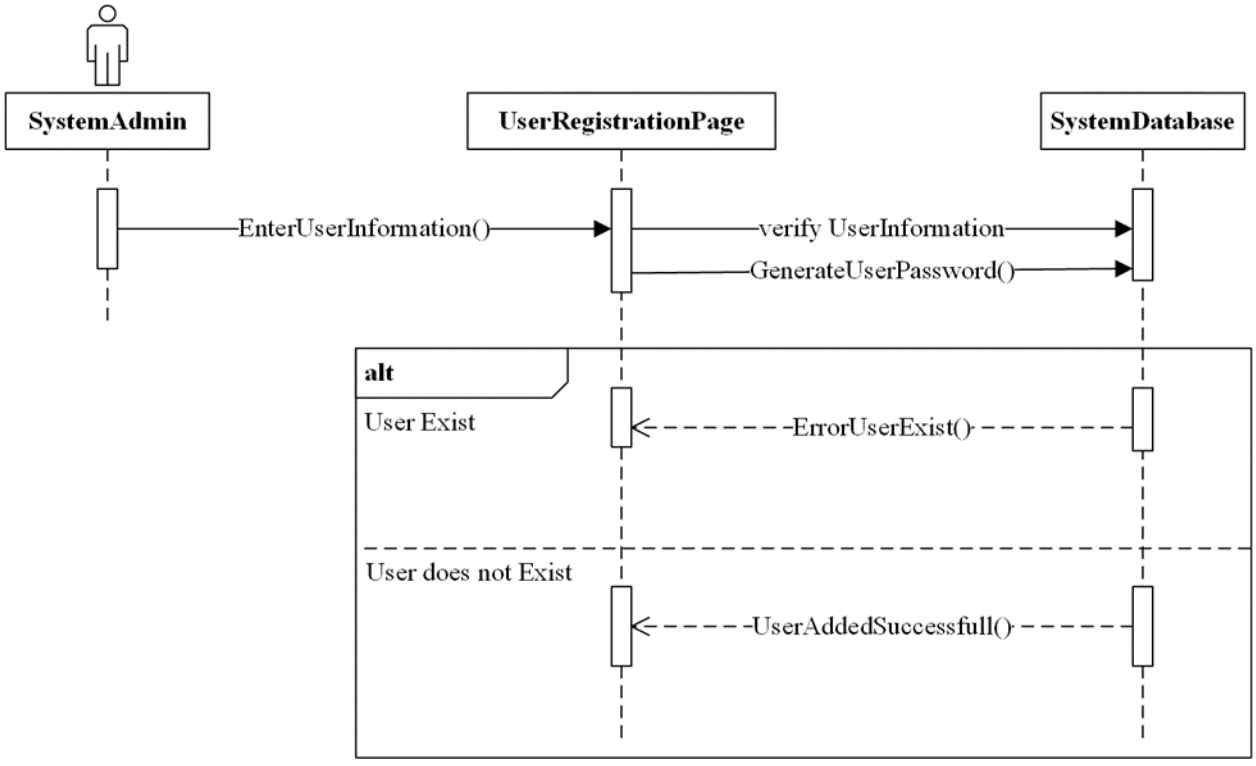

Figure 5-2: User registration sequence diagram

5.2.3 View report system sequence diagram

The system has a functionality whereby the user can view various reports. The user is required to specify the interval of time for the report to be produced in terms of start and end dates. Once the user specify the start and end dates the system will respond by providing the user the report which he/she requested, see figure 5-3.

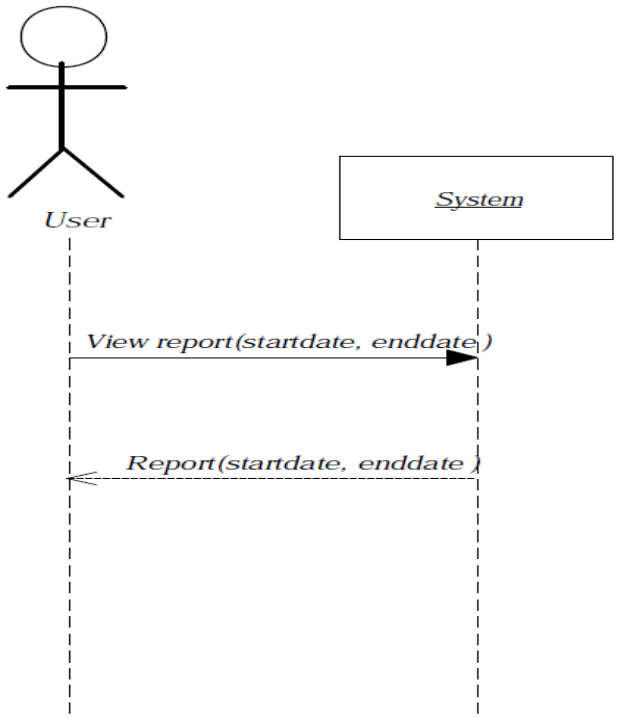

Figure 5-3: System sequence diagram for view report

5.2.4 Contract for view report

Table 5-1: Contract for view report

Name	View report(start date, end date)
Responsibilities:	Display report based on the specified interval of time (start and end dates)
Type:	System
Cross references:	System function: F4 Use case: View report
Note:	
Exceptions:	If the entered dates are invalid or exceed the allowed range indicate error
Output:	
Pre-conditions:	Data for the report must have been stored already in the system
Post-conditions:	-Report was created (instance creation) -The report was associated with system administrator, employee and expert (association formed). - Name and date attributes were set to respective values (attribute modification).

5.3 Class diagram

A class diagram shapes or models the static structure of a system. It shows relationships between classes, objects, attributes, and operations. See figure 5-4.

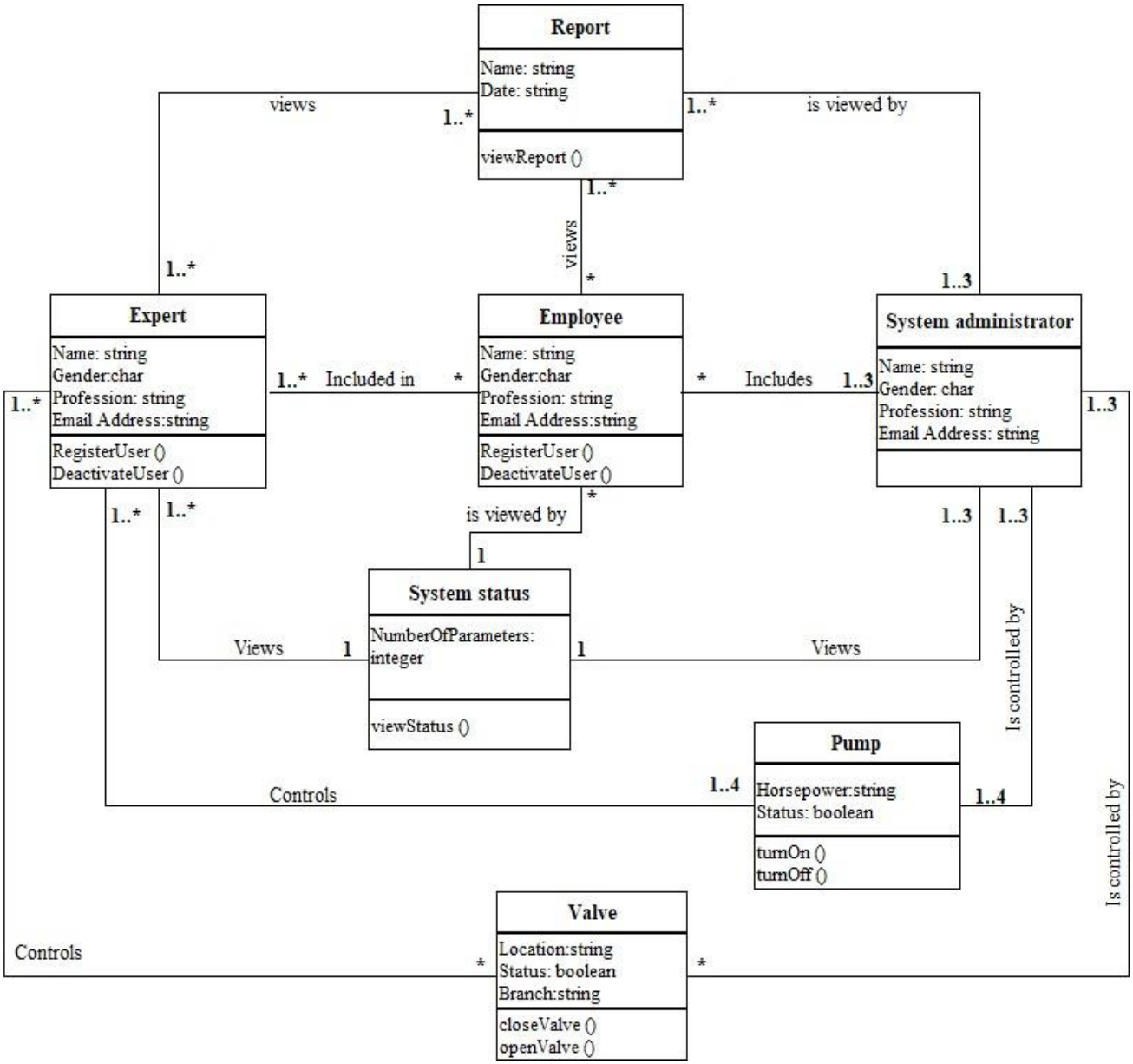

Figure 5-4: Class diagram for the system

5.4 Database design diagram

A database is a collection of tables, schemas, queries, reports, views, and procedures. Also, is defined as an organized collection of data. A database designed for the system is for model the process of storing and processing data stored on it. See figure 5-5.

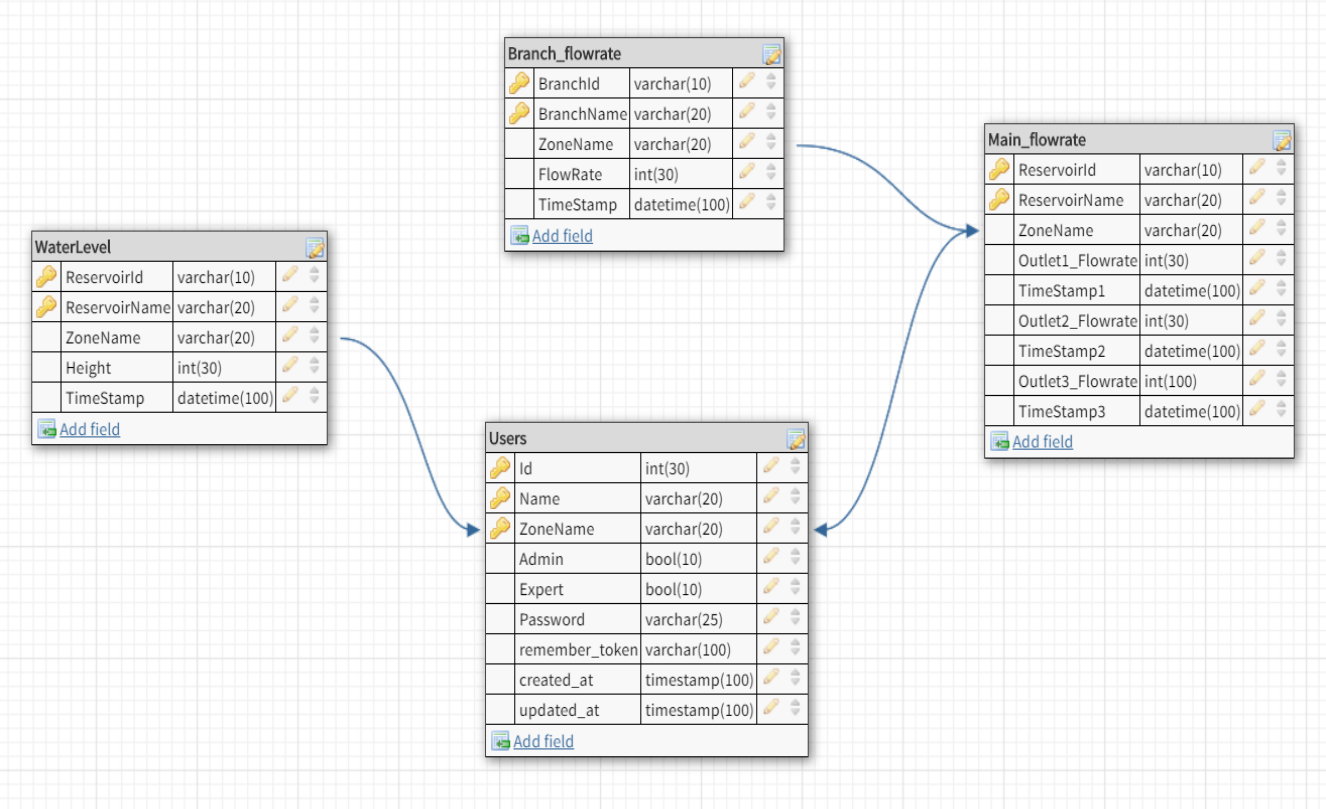

Figure 5-5: Class diagram for the system

CHAPTER SIX

SYSTEM IMPLEMENTATION

6.1 System implementation

DAWASA water supply regulation system is implemented using Laravel PHP framework which uses the MVC (Model-View-Controller). The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects. See figure 6-1. (Codes, 2015)

6.1.1 MVC components

Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

View

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data. (MVC Framework Introduction)

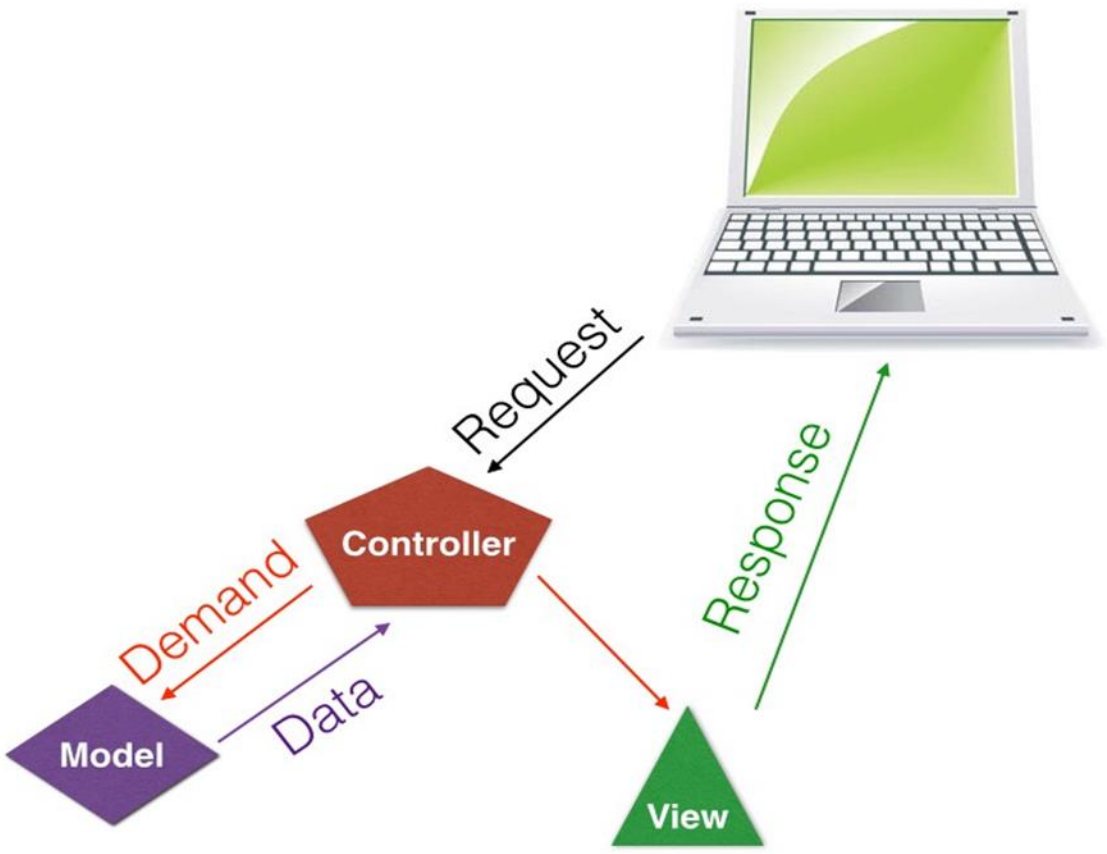

Figure 6-1: MVC architecture

6.2 Database (MODEL)

Database for the system has been created using the design established in CS 498 (see figure 5-5) and the name of the database is "Dawasumores" (DAWASA Water supply and Monitoring System), see figure 6-2.

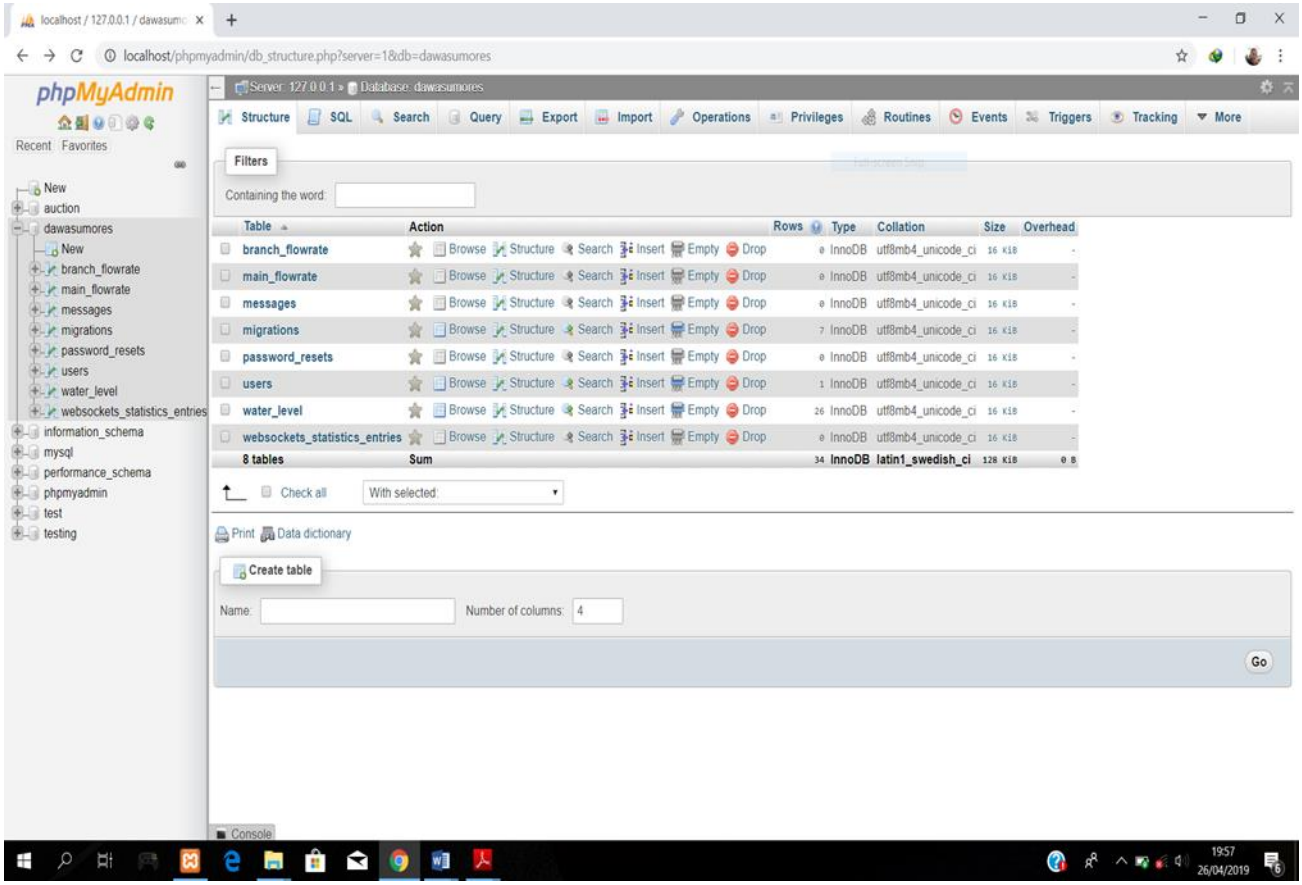

Figure 6-2: Database implementation

Other tables not included in the design such as migrations, password_resets, messages and websockets_statistics_entries are created automatically by the laravel app for operational purposes.

6.3 Controllers

The system uses several controllers to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. Example of a controller used in the system is Change_passwordController, see figure 6-3.

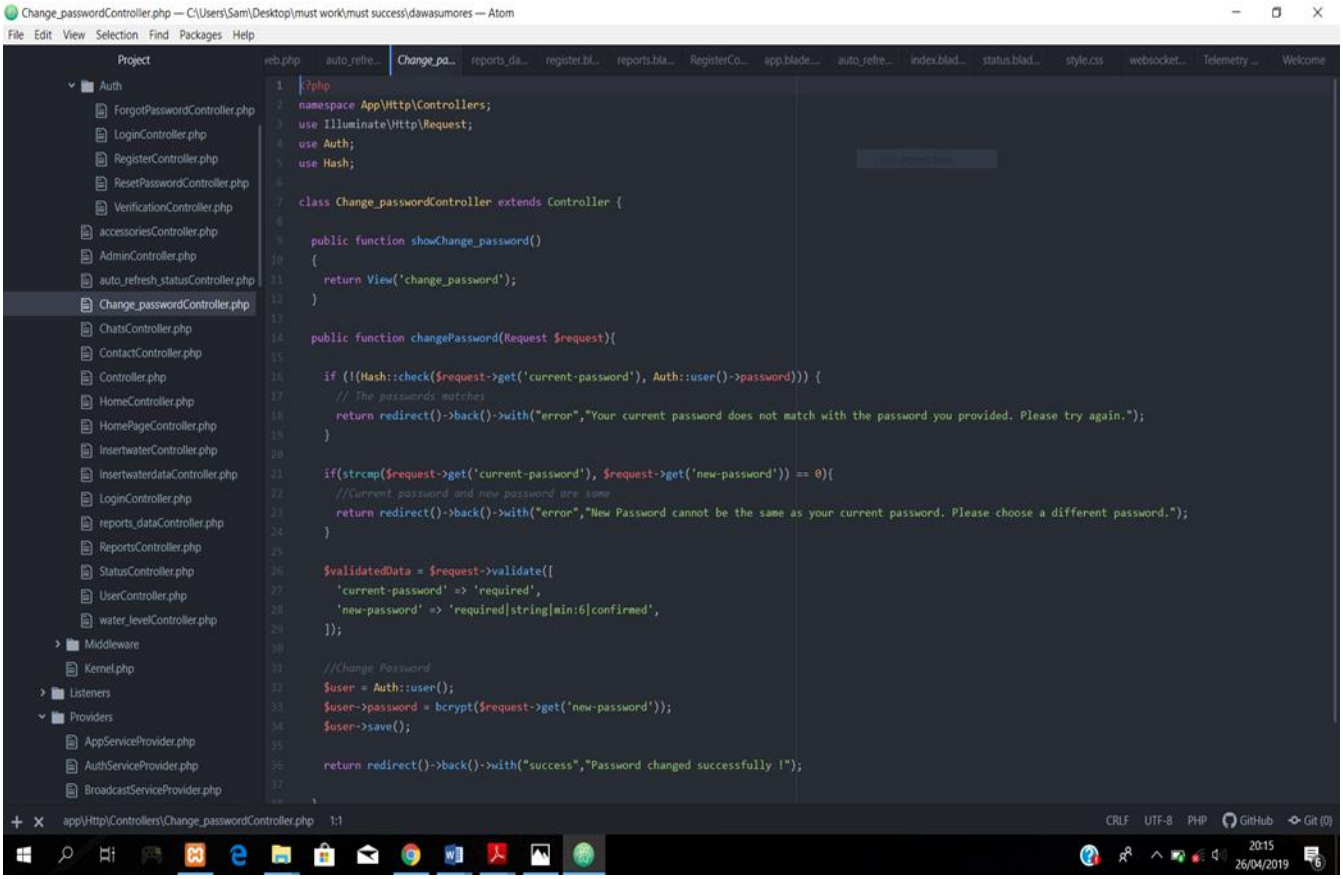

Figure 6-3: Change password controller

6.4 Views

Views are what the user sees, they receive data from the Controller of the MVC and packages it and presents it to the browser for display. In this system there several views and example of a view is a home page view, see figure 6-4.

Figure 6-4: Home page view

6.5 System functionalities

6.5.1 Role based login function

When a user logs in, the system checks what type of user he/she is and based on the type of user the system present him/her with functionalities that suits him/her.

a) Normal employee

For a normal employee the system allows him/her to mainly monitor real-time variation of various parameters of the system such as water level in the reservoir, flowrate, pump status and to query reports without the ability to do any controlling action such as to turn on or off the pump. See figure 6-5.

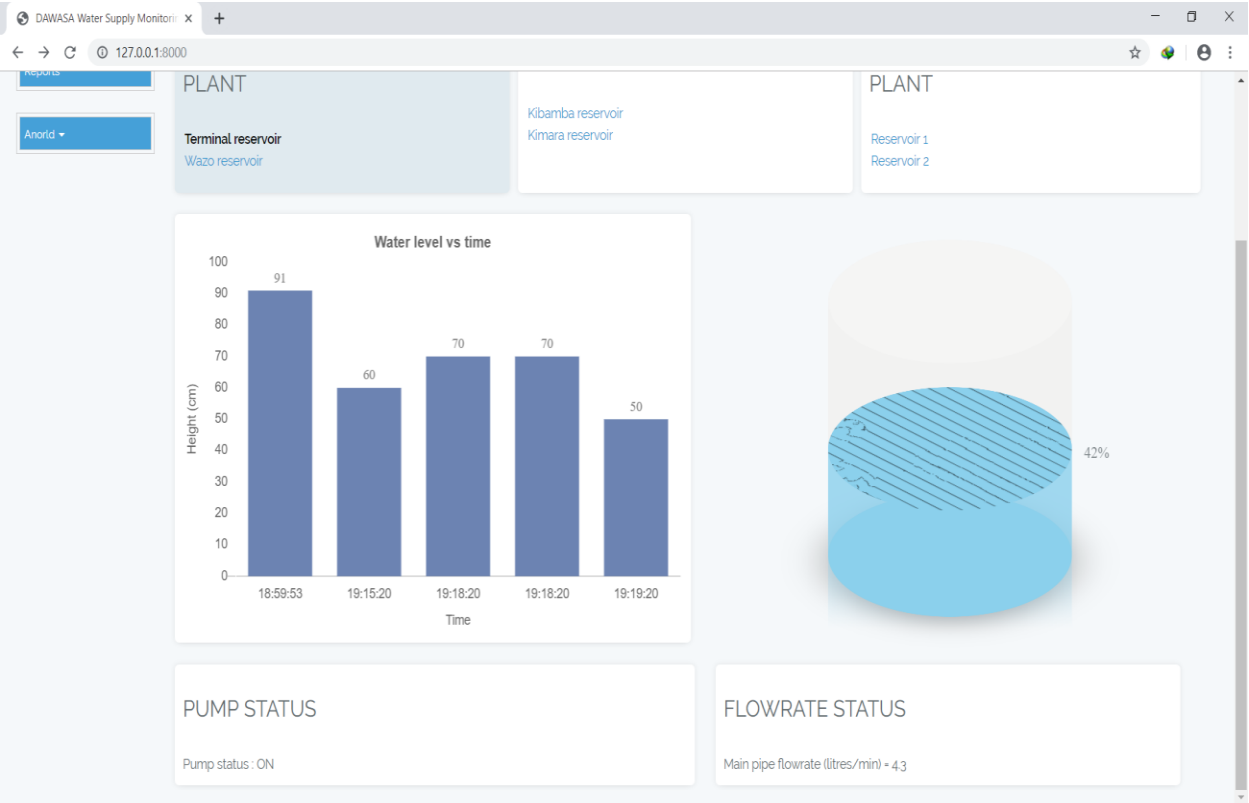

Figure 6-5: Login case for a normal employee

b) Expert

When an expert logs in, the system presents him/her with the same features as the normal employee but with additional ability to turn on or off the pump.

c) Administrator

When a system admin logs in, the system provides all features as the expert with additional ability to add or deactivate users of the system. See figure 6-6.

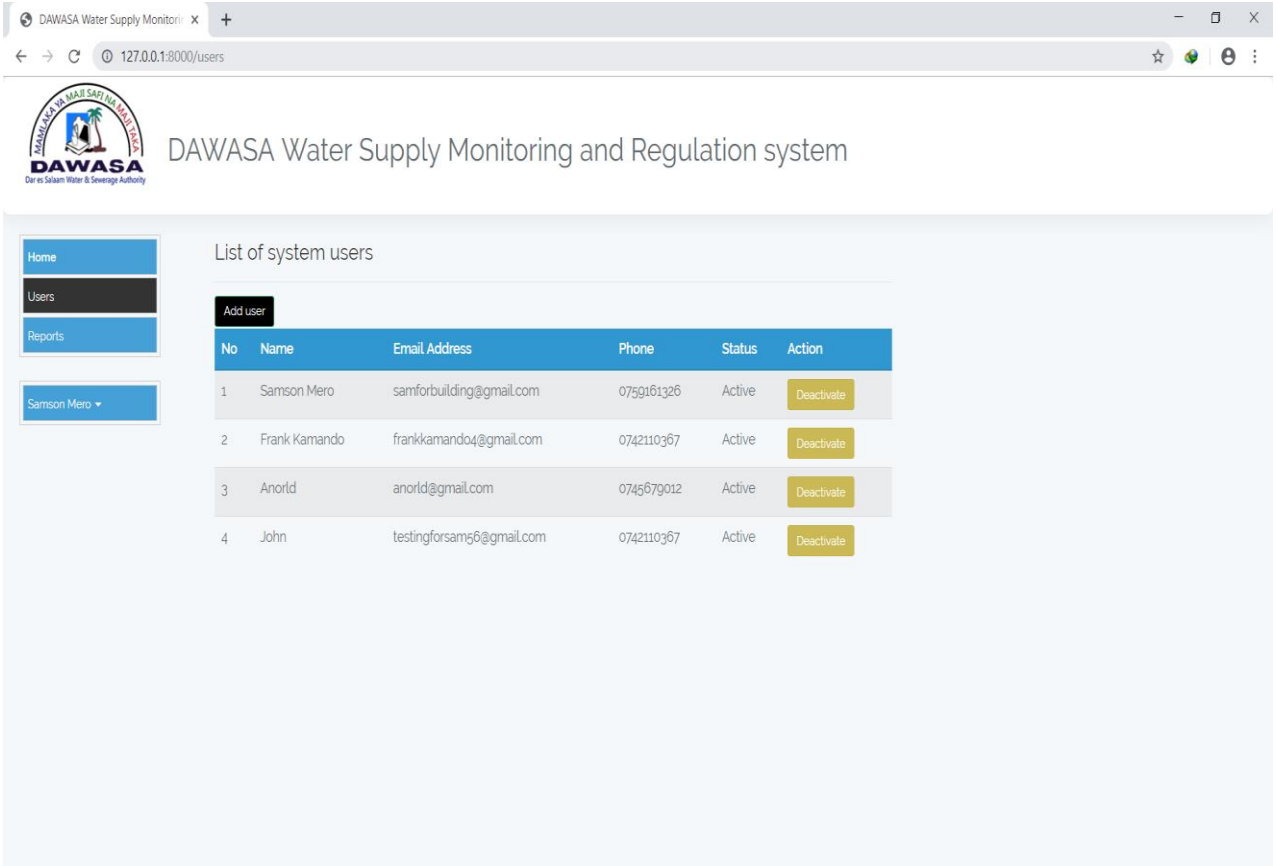

Figure 6-6: List of system users

6.5.2 Reports generation

The system generate report for water level in the reservoirs based on user specified start and end dates, see figure 6-7.

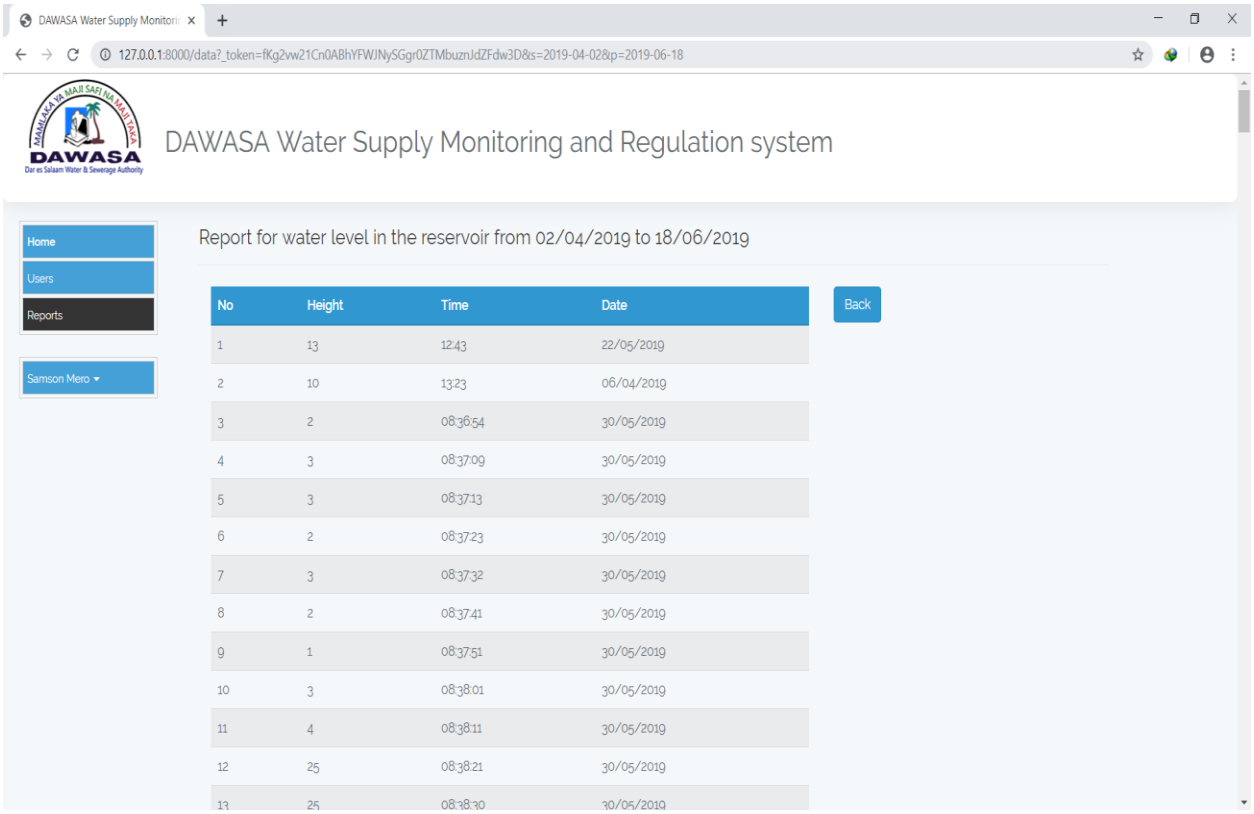

Figure 6-7: Reports

6.5.3 View profile

The system gives user an option to see his/her profile information such as name, email and phone number, see figure 6-8.

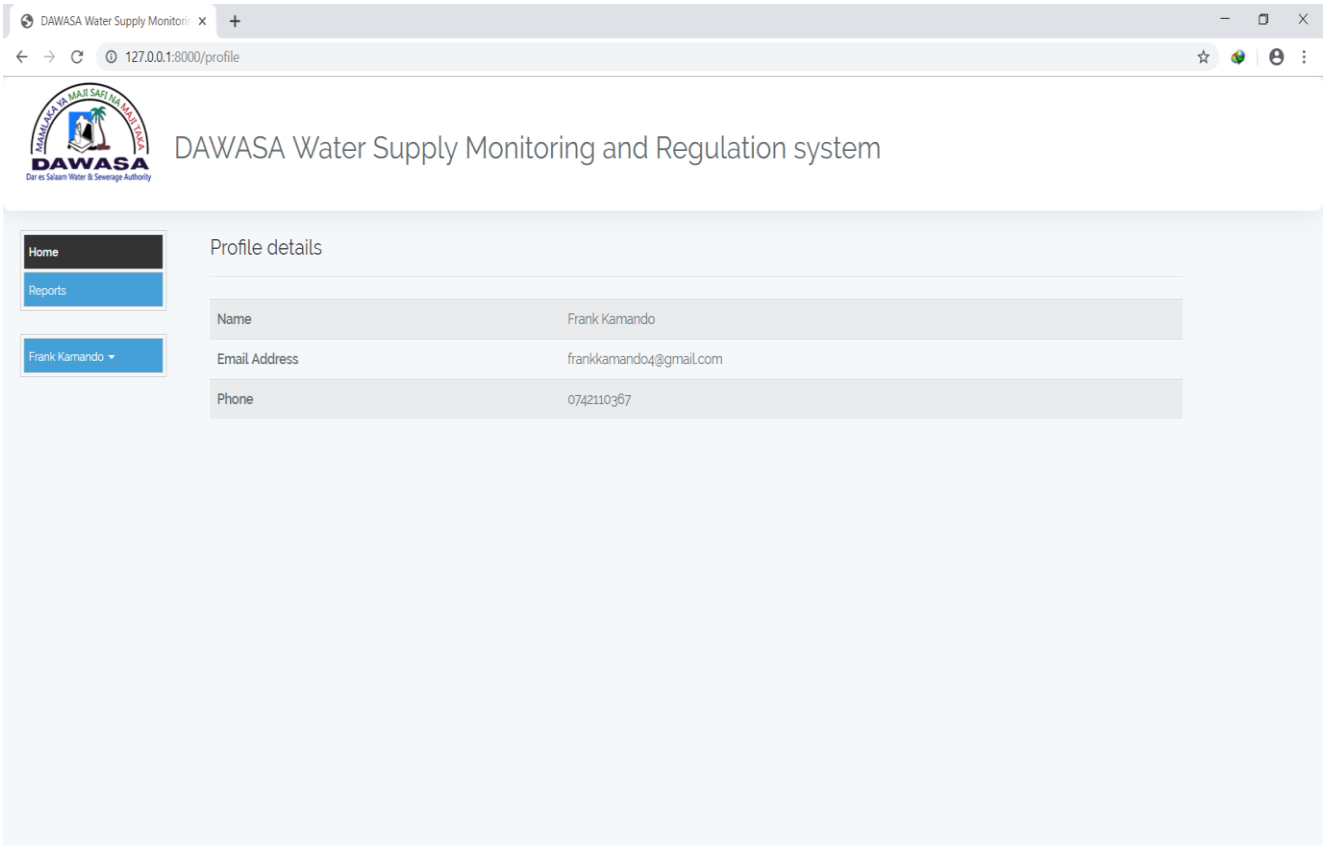

Figure 6-8: View profile

6.5.4 Edit profile

Here the user is given an option to change his/ her particulars such as name, email address and mobile number, see figure 6-9.

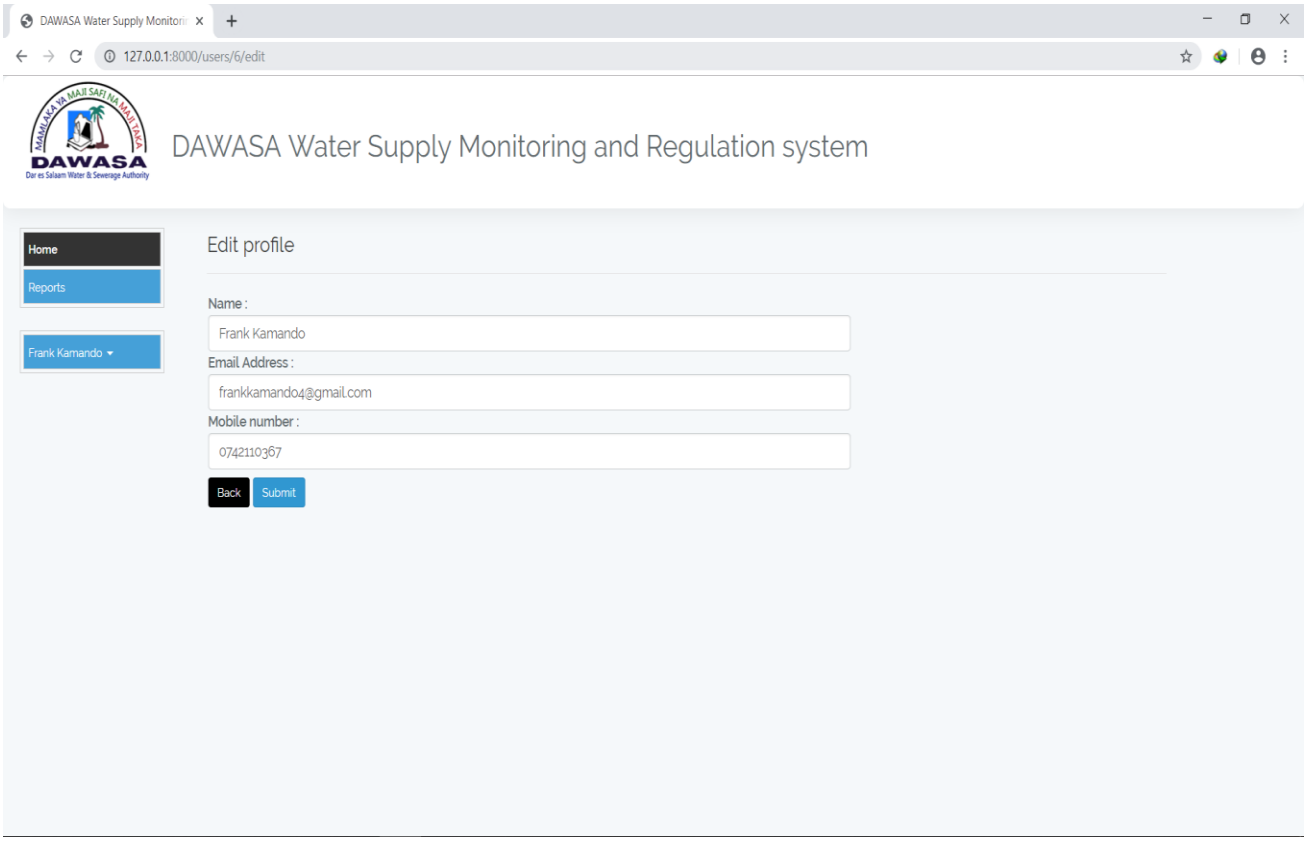

Figure 6-9:Edit profile

6.5.5 Password change

Here the system allows the user to change his or her password. By default, the system assigns a default password “waterforlife” to every new user added to the system and therefore he or she is advised to change it after he/she logs in for the first time, see figure 6-10.

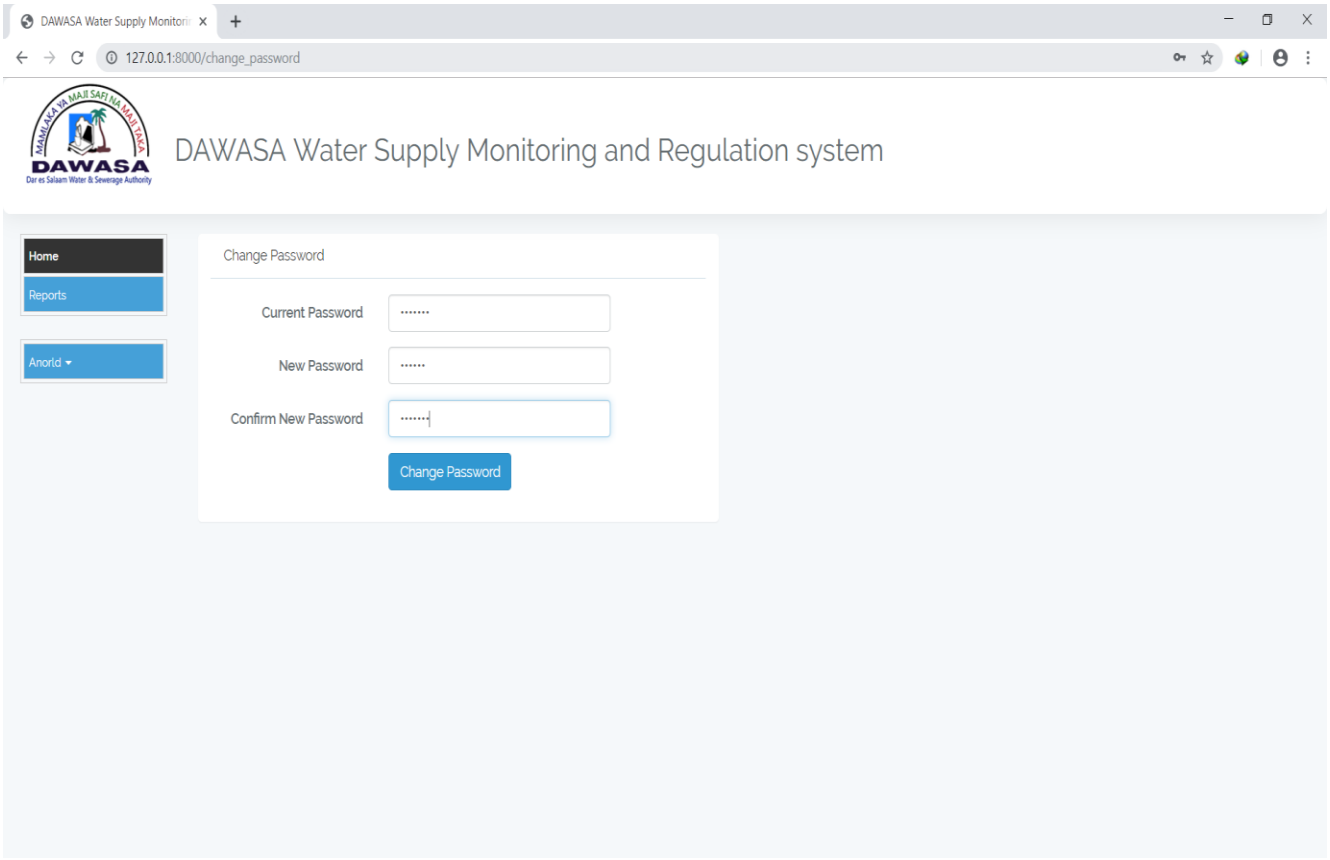

Figure 6-10: Password change

6.5.6 Receiving and sending data to Arduino

The system is able to receive data from Arduino save it to the database and from there display it in the front-end in the form of a graph or a 3d water tank as shown in figure 6-4. The real life tank represented by the system is shown in figure 6-11. Furthermore, the system is able to send data/control signals such as signals to turn off or on the pump, see figure 6-12.

Figure 6-11: Tank whose water level is represented by the system

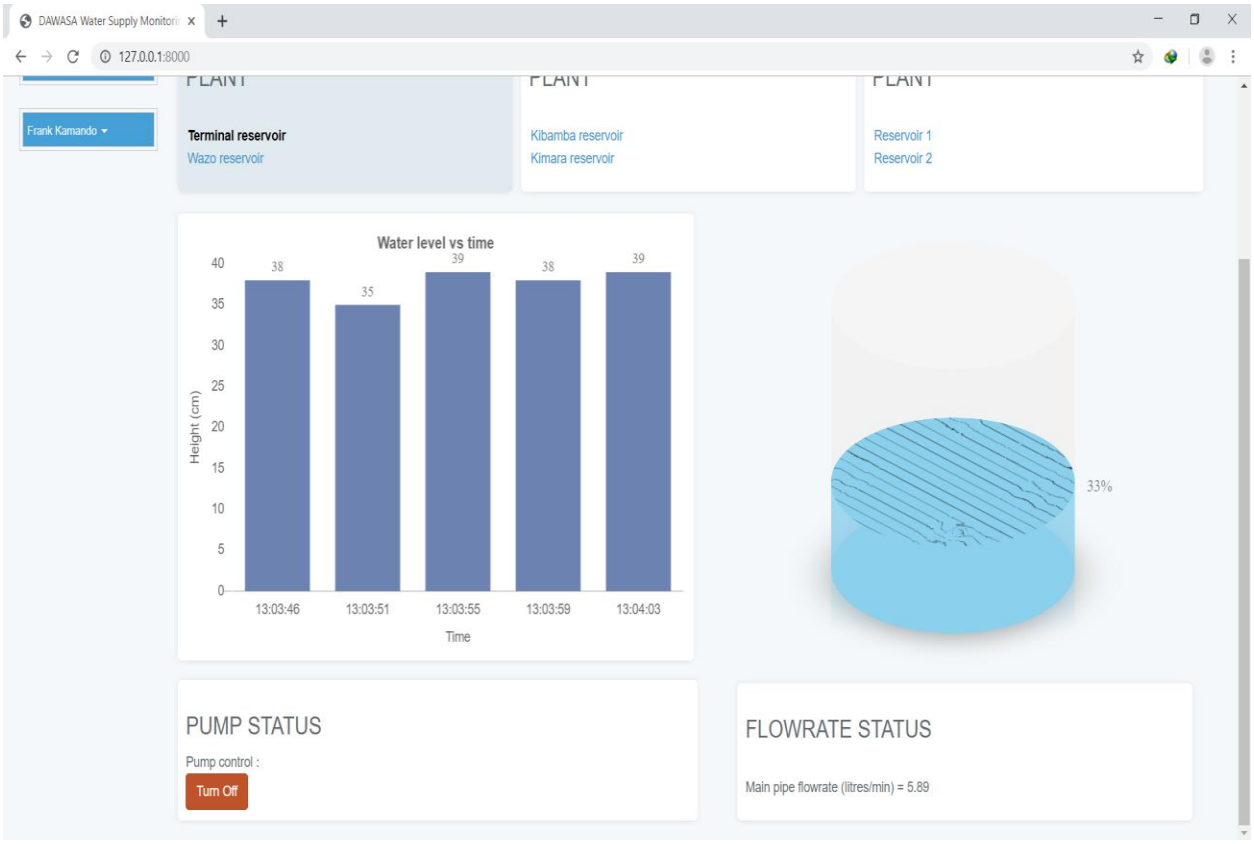

Figure 6-12: Button to turn on/off pump

CHAPTER SEVEN

SYSTEM TESTING

7.1 Definition

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

7.2 Why is software testing important?

Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss, and history is full of such examples.

a) In April 2015, Bloomberg terminal in London crashed due to software glitch affected more than 300,000 traders on financial markets. It forced the government to postpone a 3bn pound debt sale.

b) Nissan cars have to recall over 1 million cars from the market due to software failure in the airbag sensory detectors. There has been reported two accidents due to this software failure.

c) Starbucks was forced to close about 60 percent of stores in the U.S and Canada due to software failure in its POS system. At one point, store served coffee for free as they unable to process the transaction.

d) Some of the Amazon's third party retailers saw their product price is reduced to 1p due to a software glitch. They were left with heavy losses.

e) Vulnerability in Window 10. This bug enables users to escape from security sandboxes through a flaw in the win32k system.

f) In 2015 fighter plane F-35 fell victim to a software bug, making it unable to detect targets correctly.

g) China Airlines Airbus A300 crashed due to a software bug on April 26, 1994, killing 264 innocent live (What is Software Testing? Introduction, Definition, Basics & Types).

7.3 Testing cases of the system

This system has been subjected to a number of tests and results for each test was obtained as it is going to be described in the below.

7.3.1 Login function

In the process of login into the system the system compares the credentials that have been entered by the user with the one stored in the database. Furthermore, the system checks whether the user is active or not active (has been disabled by the system admin) and when the credentials don't match or the user is inactive the system returns an error message as shown in figure 7-1.

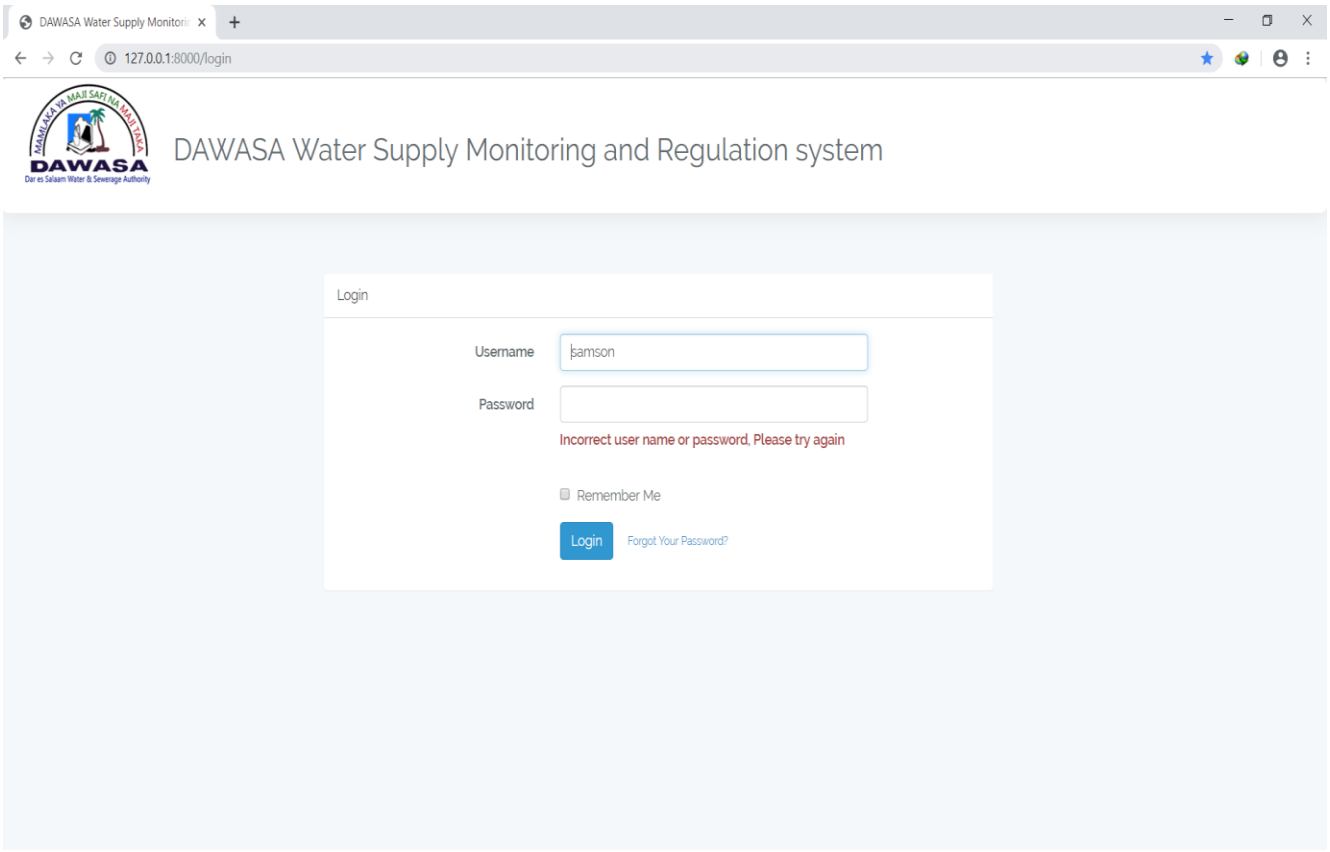

Figure 7-1: Login attempt failure

7.3.2 Form validation

Form validation helps us to ensure that users fill out forms in the correct format, making sure that submitted data will work successfully with our applications. When you enter data, the web application checks it to see that the data is correct. If the information is correct, the application allows the data to be submitted to the server and (usually) saved in a database; if the information isn't correct, it gives you an error message explaining what needs to be corrected (sidd, 2019).

Form validation for this system (for the case of querying reports from the database) is as follows:

a) When start date or end dates are greater than the current date

   The system prevents the form from being submitted and returns error as shown in figure 7-2.

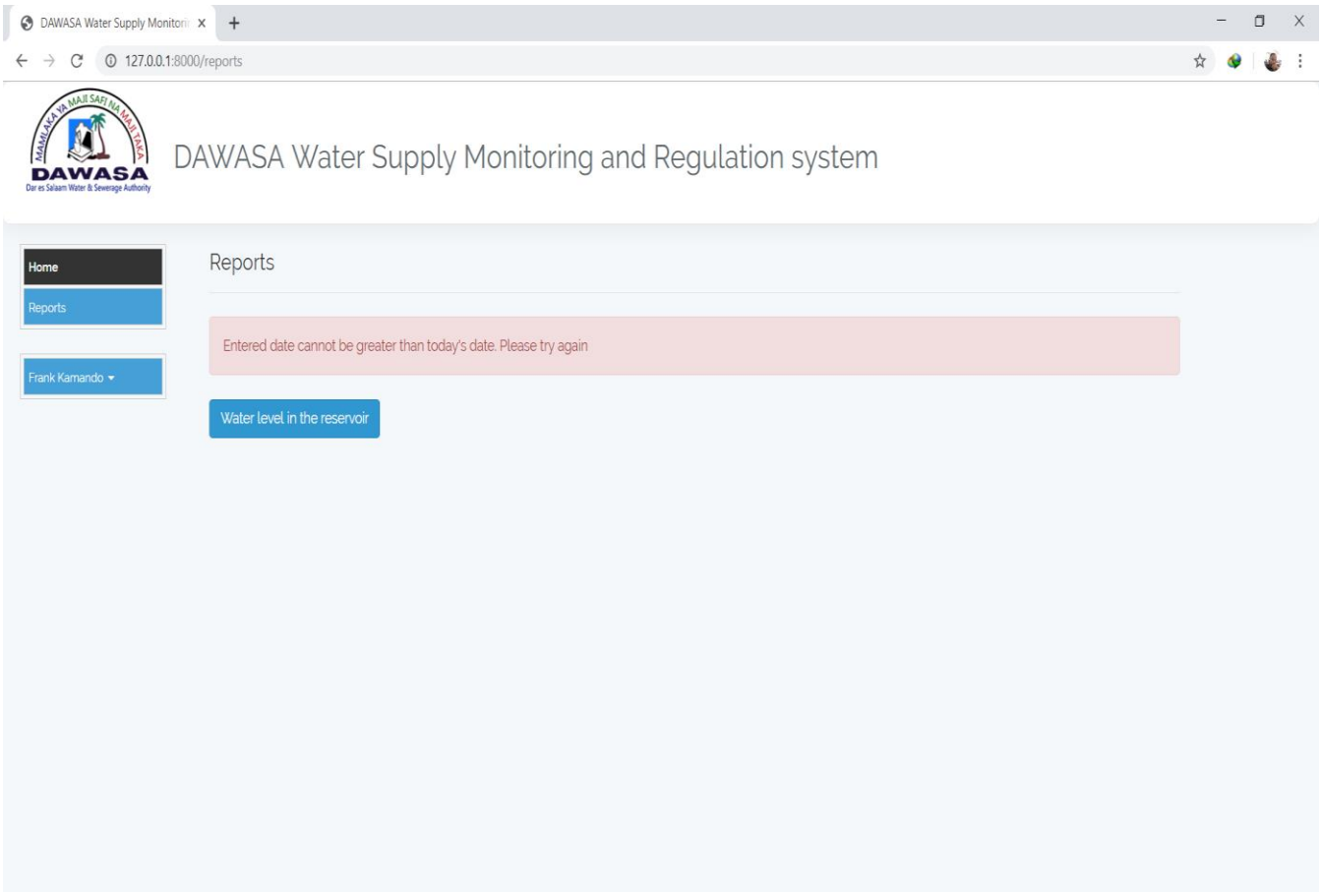

Figure 7-2: Error when start or end date are greater than current date

b) When start date is greater than the end date

When the user selects a start date that is greater than the end date the system responds as shown in figure 7-3.

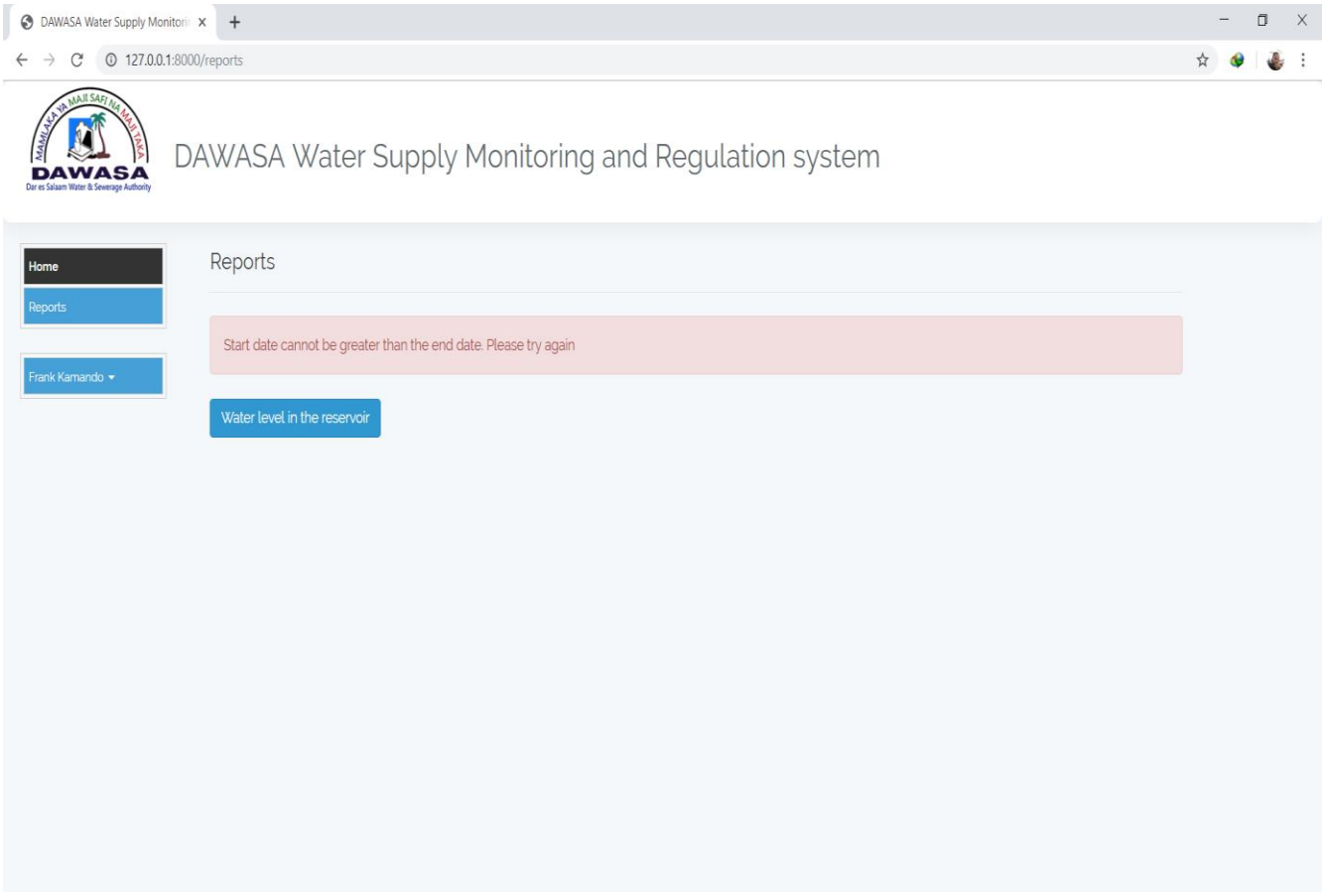

Figure 7-3: Error when start date is greater than end date

c) When the user doesn’t enter any date

When the user doesn’t select any date the system responds as shown in figure 7-4.

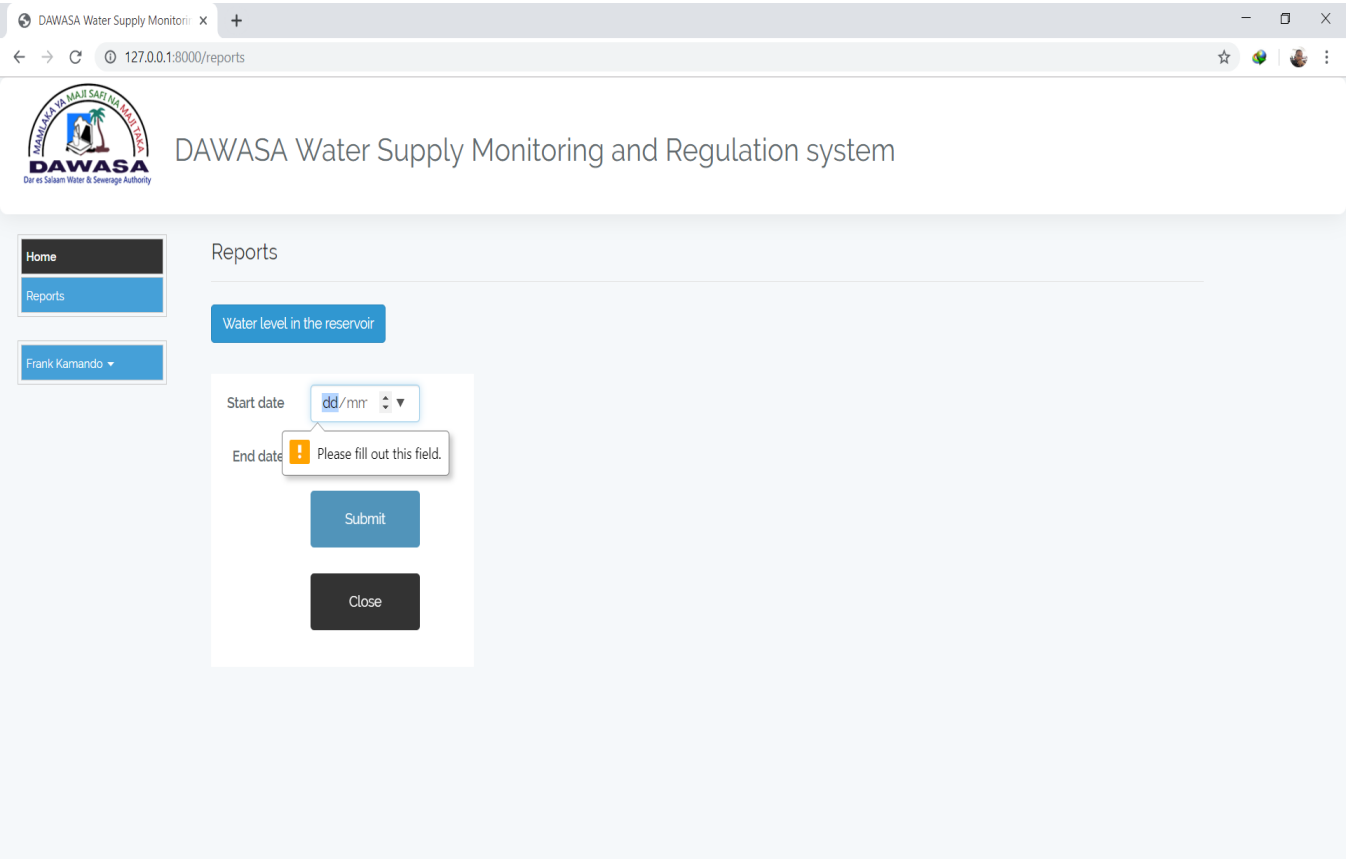

Figure 7-4: System response when the user doesn't select any date

CHAPTER EIGHT

CONCLUSION AND RECOMMENDATIONS

8.1 Conclusion

Changing a university curriculum to be a problem based curriculum which will expose students to different problem solving skills that are required in the industry is a very good thing to do but not so easy as it sounds. This is where challenge driven education comes in, a way to expose students to problem solving skills without changing the entire university curriculum. This is achieved by changing the way of conducting final year projects by engaging students to participate in solving real life societal challenges with the use of various techniques and tools such as design thinking and team working. All these are very beneficial to students as they equip them with different knowledge and skills such as presentation skills, project management skills and team working skills which are the required skills in industry.

In the course of this project the challenge that was encountered was the failure to purchase a raspberry pi which was proposed in the methodology part of this report and this was due to financial constraints and therefore only Arduino was the one used. Apart from that the project went well and it was successful.

Therefore, as being one of the students who got an opportunity to conduct a challenge driven education project, I can say this project has been so helpful to me in many ways. Apart from the skills I have acquired this project has exposed us students to the industry as we have been interacting with various personnel from DAWASA and other stakeholders and hence increases our chances of employment.

For the time schedule and semester budget refer to Appendix A and appendix B respectively.

8.2 Recommendations

I highly recommend the continuation of challenge driven education projects in our university and other universities which have not yet adapted to also adapt due to a lot of benefits that comes with the challenge driven education. By this way students will be equipped with various skills that are required in the industry.

REFERENCES

Advantages of using the Laravel framework. (n.d.). Retrieved from eTatvaSoft - Web & Mobile App Development Company: https://www.etatvasoft.com/blog/advantages-of-using-the-laravel-framework/

Automated Water Water Supply Control System (Chaos Logic-based Demand Prediction Method). (n.d.). Retrieved from MEIDENSHA CORPORATION: http://www.meidensha.com/products/water/prod_03/prod_03_02/index.html

Codes, J. (2015, August 29). How MVC Architecture works. Retrieved from Medium: https://medium.com/@JanlCodes/how-mvc-architecture-works-483254288a45

Mack, J. (2014, May 28). Five Advantages & Disadvantages Of MySQL. Retrieved from Datarealm: https://www.datarealm.com/blog/five-advantages-disadvantages-of-mysql/

MVC Framework Introduction. (n.d.). Retrieved from Tutorialspoint: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

sidd. (2019, May 23). Form data validation. Retrieved from MDN Web docs: https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Form_validation

Waterfall Methodology: Advantages, Disadvantages And When to Use It? (2018, March 15). Retrieved from Lvivity - Custom Software Develoment company. IT outsorcing for business: https://lvivity.com/waterfall-model

What is Software Testing? Introduction, Definition, Basics & Types. (n.d.). Retrieved from Guru99: https://www.guru99.com/software-testing-introduction-importance.html

APPENDIX

Appendix A: Time schedule for the semester

No	Activity	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Literature review															
2	Database implementation as per database design established in the last semester															
3	Implementing front-end views such as homepage, reports page, change password page etc.															
4	Implementing user login functionality															
5	Interfacing with Arduino (receiving data from Arduino)															
6	Progress report for mid-semester II (collection)															
7	Mid-semester II progress presentation (oral delivery)															
8	Implementing admin dashboard															
9	Making the site responsive to different screen sizes															
10	Implementing data processing algorithms															
11	Interfacing with Arduino (sending data from the system to Arduino)															
12	Final report for semester II (collection)															
13	Semester II final Presentation (Oral Delivery)															

Appendix B: Estimated budget for the semester

S/N	ITEMS	QUANTITY	ESTIMATED	ESTIMATED
			PRICE/EACH(Tsh)	COST(Tsh)
1.	Internet bundle		80,000	80,000
2.	Stationary		50,000	50,000
			Estimated total cost	130,000