BACHELOR OF SCIENCE IN INFROMATION TECHNOLOGY

PROJECT: BIT 3205

WATER RESERVOIR MANAGEMENT SYSTEM FOR MÙRERA WARD

BY

NJOROGE MORRIS KIMANI

morrisnjorogekimani@gmail.com

16/05387

PROGECT FINAL DOCUMENT IN PARTIAL FULFILMENT FOR THE REQUIREMENTS FOR THE AWARD OF A DEGREE IN INFORMATION TECHNOLOGY

PRESENTED TO: Mr. Joseph Kuria

Declaration

I declare that this project is my original work and has not been presented in any other college or university for the award of a Diploma or a Degree.

Student

Name……Morris Kimani N. ……. Date…13th December 2018…

Signature……  ……..

Table of Contents

Contents

Project Proposal

1.0 Background

Water is life. This is a famous saying that is globally known in as many languages as you can imagine in existence today, including the language spoken by machines; binary.

Water reservoirs are storage spaces for water and are very important to the community. They are in many forms: dams, tanks and lakes. Some may be even found underground like one currently lying more than 339 meters beneath Turkana which is estimated to be enough to quench our country's thirst for seventy years.

The reservoirs however always face one common problem; managing the water and maintaining accurate data on how much water is being pumped out into homes and industries.

2.0 Introduction

Mùgùtha Ward, like many other wards has a water project meant to supply water to the many homes that exist within it. It was started in the year …….. and till today, it continues to function very efficiently because the water it supplies from its reserve tanks is from a borehole.

Even so, the project has faced a lot of challenges during the period which it has been active. They include:

i. Water shortages due to sudden fluxes in consumption.
ii. Imbalance in the bill payments to the amount of water that has been pumped out of the reserve tanks.
iii. Over reliance on mechanical water level control systems which may malfunction causing a drain in the reserves hence posing a great risk of damage to the water pumps due to overheating.
iv. Lack of a proper computerized system to analyze the usage statistics that is collected after water meters are read.
v. Lack of real-time data on usage which may be critical in detecting damage to the infrastructure being used to deliver the water to customers.
vi. Poor management of past records which makes it difficult to plan for future expansion and while maintaining good service to already existing customers.

The main objective is to have 24/7 automated systems monitoring the levels of water in the reserve tanks while recording the data in real time and transmitting it for processing and storage.

The specific objectives are to:

i. Create a system that will be used to remotely manage the water being pumped into and out of the reserve tanks.
ii. Obtain data on the amount of water being used on a day to day basis, analyze and store it for future reference.
iii. Eliminate the need for purely mechanical valves and implement electronic ones that will be controlled from the remote command center.

iv. Provide a redundant system that can be used for controlling the valves and receiving the data being collected in case the main communication network connection is lost.
v. Ensure that the system can alert the users incase of any predefined or unexpected changes, e.g. water level drops to a certain level or the tanks are almost full.

The system I am developing will be an added advantage to the Ward's water project because it will:

i. Provide details on usage statistics down to almost second by second time period.
ii. Reports will be ready for analysis on a daily basis.
iii. It will reduce the labor needed in running the water project.
iv. After analysis, the data can be stored for future reference and be an important tool in the future. In case of shortages, reasonable rationing can be done based on the statistics before other solutions can be provided.
v. The imbalance between the units of water paid for by the customers against the amount the system will say was pumped will show the real losses hence prompt proper maintenance hence improved services.

2.1 Requirements

To undertake this project, I will require the following things (subject to change during development)

1. A computer of 2GB RAM and 2.4GHz processor
2. MySQL database to store the data
3. Visual Basic .NET for the user interface
4. A microcontroller to operate all the sensors and communications
5. A sonar module for getting depth of the water
6. GSM module for a redundant communication link
7. Ethernet/Wi-Fi module for communicating with the user interface

2.2 Logical Design

A logical design is a conceptual design that comes before the physical implementation of a system. It shows how information will flow within a system and shows the relationship between objects in the system.

This system will comprise of three main parts:

1. **The Microcontroller:** It will be programmed to collect data via the sonar module of the water levels in the reserves, package it and send it out via the networking modules as well as control the pumps that control water flow into and out of the reserve tanks. It will also receive instructions from the user interface in the remote command center on how to handle events such as low water volumes. In case of no response from the remote command center, it will also have default settings on how to handle events.
2. **User Interface:** It will be used to send commands to the microcontroller as well as receive and process data collected and sent by the microcontroller. It will then push the information into the database.
3. **The Database:** It will be used to store the data obtained by the microcontroller.

2.3 Development

The development of this system will be quite challenging considering it will involve creation of a hardware component and programming it to perform as expected.

2.4 Testing and Implementation

Testing takes nearly half the cost budget of development of systems. This is to ensure that the system is working fine and is ready to be implemented into the environment it was made for. Implementation is the release of a system for use by the client after verifying that it has met all requirements.

3.0 Literature Review

Water reservoirs have been in existence for many years and have increasingly been used by people for a great number of purposes which include construction, farming and even as an educational tool and entertainment purposes. This also means that management of how the water is dispersed has also changed a lot.

One of the first methods of managing water reserves was by the use of **cisterns** which can be traced back to the 4th millennium BC in North Eastern Lebanon. These are waterproof structures used to hold rainwater which today vary in size.

Currently, mechanical float valves are used in many parts of the world to control water reservoirs with the exception of big water projects such as dams which have a consistent supply of water from the river that flows behind it. Reserve tanks are fitted with a mechanical float valve with the recent technology ensuring that the valve controls and electric float switch that powers on/off a water pump that is filling the reserves depending on the level of the water in the tank.

From my experience during a weekend out where the float switch slipped from where it was fixed in contact with the water's surface and was no longer taking up the level of the water thus the water pump of a client kept working the entire night and many litres of water were wasted and the pump overheated due to depleting the water available in the well.

Currently, the Mùgùtha Ward water project implements this floatswitch to control the flow of the water going into the tank hence there is no data being collected on the water

4.0 Methodology

Methodology is a concept that is used to offer a theoretical understanding of which process is the best to apply in a specific case to ensure that the task is completed.

For this project, I have chosen to use the **Prototyping Methodology** due to the following reasons:

- ✓ It increases user involvement hence they will accept it more openly because they will have a sense of ownership of the project.
- ✓ The product will have gone through enough changes to ensure that it has met user requirements
- ✓ Reduces time of development and costs because one can predict early exactly what the users eventually want to have.
- ✓ This project having production of new hardware involved, it will be easy to change designs depending on user requirements without incurring heavy costs.

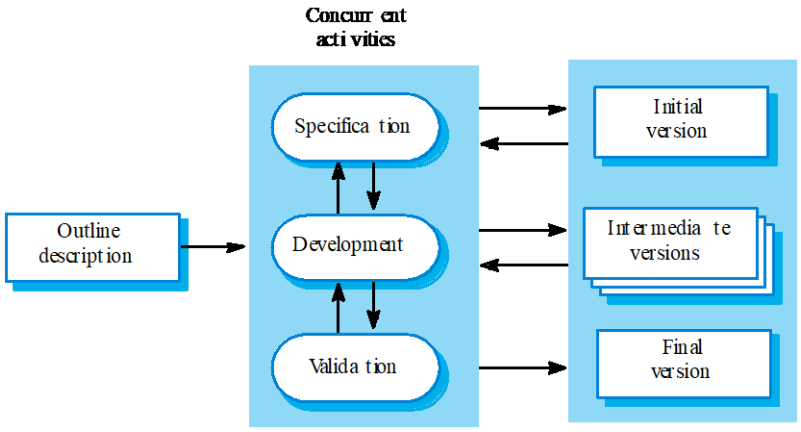


Figure 1: The Prototyping Process

5.0 Schedule

			START DATE		COMPLETION DATE	
	TASK ID	Task Duration	PLANNED START TIME	ACTUAL START DATE	PLANNED COMPETION DATE	ACTUAL COMPLETION DATE
1	Writing Project Proposal	1 week	13/5/2018	20/5/2018	19/5/2018	26/5/2018
2	Proposal Presentation	1 day	28/5/2018	28/5/2018	28/5/2018	28/5/2018
3	Feasibility study	2 weeks	29/5/2018		10/6/2018	
4	System Design	2 weeks	11/6/2018		26/6/2018	
5	Construction	8 weeks	1/7/2018		19/8/2018	
6	System Testing	2 weeks	20/8/2018		9/9/2018	
7	Documentation	2 weeks	10/9/2017		24/9/2018	
8	Presentations	1 day				

6.0 Budget

Item	**Quantity**	**Price per Unit (Ksh)**

2GB RAM 2.4GHz Processor Computer	1	-
Microprocessor set (All sensors and modules included)	1	7500
TOTALS	**2**	**7500**

7.0 References

1. Richard Swancott (2001) ICT Project Management
2. https://docs.oracle.com/cd/A80142_01/DOC/Server.816/a76004/logical.html
3. https://en.m.wikipedia.org

System Requirement Specification (SRS)

1.0 Introduction

1.1 Purpose

Software development is a process by which one develops a system based on studies on the existing systems, their strengths and limitations and working to ensure that one produces an improved product that solves unique problems.

The purpose of the system requirements specifications document is to show system designers the needs of the clients.

1.2 Intended Audience

This document will be available to the top management of Mùgùtha Ward, users of the system being developed, future developers, and project management committee of KCA University and will also serve as a reference to other students interested in the field of software development.

1.3 Aims of SRS

The purpose of the Software Requirements Specification is to outline the requirements for The Mùgùtha Ward Water Reservoir management system. It will be built on Visual Basic.Net for the user interface, MySQL for the database and Arduino programming language for the microcontroller. It will be operating system dependent and accessible only to the connected workstations.

1.4 Project Scope

The project is aimed at creating a platform for effectively developing a computerized system that will meet the needs for the Mùgùtha Ward Water Project.

The new system is going to run on two servers i.e. an application server that will host the user interface and the database sever that will be used to store and retrieve data. The data will be stored in a MySQL database within the ward's water project offices. The data will be collected by a microcontroller with sensors for water levels and will also control the pumps for filling and emptying the reservoir.

1.5 References

https://www.t4tutorials.com/srs-software-requirement-specification/

Software Engineering, A practitioner's Approach,

Harvard Publishers, USA, 1985.

2.0 Overall Description

The Mùgùtha Ward Water Reservoir Management is a relatively new product which will function on a client-server model. The user interface which will also hold some of the processing logic will be installed in individual workstations within the offices while the database will be installed in one server and will be available to all the workstations when needed.

2.1 Product Features

2.1.1 Data Capturing

Data will be collected from the reservoirs using an Arduino microcontroller which will have a water depth sensor to tell how much water is currently available in the reservoirs. This information will then be relayed to the workstation controlling the microcontroller remotely via Ethernet or any wireless communication means available for processing and storage.

2.1.2 Data Processing

Data will be processed by the application in real time to provide usage statistics and advice on actions that can be taken in predefined situations such as: low water volume, running pumps but no change in water levels or even the reservoirs being full to capacity.

2.1.3 Data Output

After data capturing and processing, output will be in the form that the user requires.

2.1.4 Data Validation and Integrity

There will be control mechanisms that will check the validity of the data being captured in accordance to the rules provided.

2.1.1 Monitoring

The system will be able to oversee the various actions being taken within the system and documenting them. It will also inform users of any errors that may occur during the system's runtime and provide possible solutions on how to solve the error(s).

2.2 User Classes and Characteristics

The users of this new system will require computer-literacy training for them to be able to fully and effectively use the new system. To that effect, a user guide will be provided to the users in friendly language to help them have optimal use of the system.The system will be easy to use as it will have simple forms which will be simple to navigate. The user interface will run on specified workstations.

2.3 Operation Environment

The following will be necessary for optimal performance of the system:

2.3.1 Hardware

- Minimum hardware specifications for the workstations are **2 GB RAM**, **80 GB** hard disk,
- Minimum hardware specifications for the database server are **4 GB RAM**, **500 GB** hard disk,
- Water reservoir controller (Currently under development)
- Networking devices for connection between the microcontroller and the user interface.

2.3.1 Software

- Microsoft windows 2012 server Operating System and windows 10 for the clients/workstations.
- MySQL Database application

2.4 Design and Implementation Constraints

The current constraints are:

- Interfacing different applications is challenging.
- Current work stations do not meet the minimum requirements
- Developing the microcontroller and its components and making them work together is proving to be harder than anticipated

2.5 User Documentation

Users need to be trained on how to use a new system for the project to become complete success and also maximize on how the developed software is being utilized.

The users will have the following documents at their disposal from either the vendor or developer which will be relating to the general issues that users will encounter during use of the system:

- User Manual:
    - Provided by the developer

    - Easy to use and understand

- Knowledge Base
    - Good for quick references.
    - Assumption is that the user knows what they are looking for.
    - Compiled by the vendor
- Command prompts
    - They assume knowledge of a command.
    - Can provide information about correct usage of a command when an error occurs.
    - Designed by the vendor

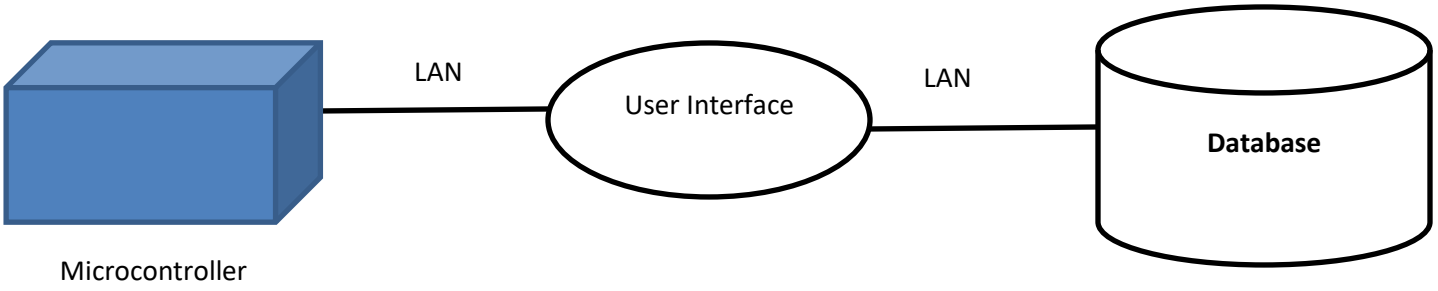
2.6 Assumptions and Dependencies

- Test data provided is assumed to be reliable.
- The MySQL database will continue to be available with existence of support and new updates.
- There will be a dedicated server for optimum performance of the database before production of the system.
- Staff will always be computer literate.
- New versions of the system will be produced to deal with any bugs that may not be captured during testing.
- It is assumed that all the clients/workstations will always run a compatible version of windows operating system.
- It is assumed that the network will always be secured such that no outsiders will gain unauthorized access to either the server or workstations in the enclosed network.
- The microcontroller upon completion will be able to run the system automatically without any human interference unless it needs repairs.
- The microcontroller will synchronize will with the rest of the system, i.e.: the user interface and database which will be located in different computers over the network connection provided.
- In case of a breakdown in the network, none of the systems' components (database, user interface, and microcontroller) will fail, They will all remain active until a connection is re-stablished.

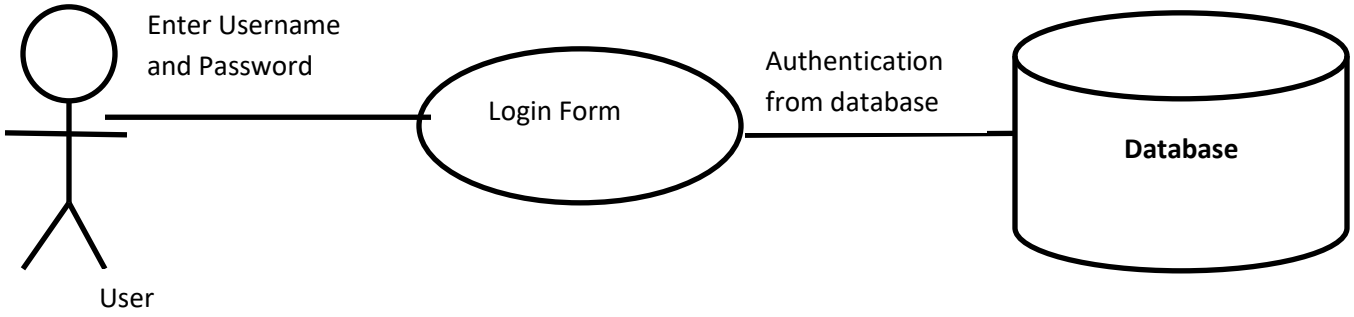
3.0 System Features

The system should be able to capture all data about the water levels automatically by aid of the microcontroller and its components. The data will be streamed in real-time through the user interface and it will be stored directly into the database.

It should also be able to produce quick and detailed reports on, water levels, usage over time and other necessary information very fast hence eliminating the current time wastage experienced in putting together this information manually. The system can generally be depicted as the following image below.



3.1 Access to The System



3.1.1 Description and Priority

To gain any access to the functionality of the system, one's identity and therefore the functions of the systems which they can utilize, this takes place as the system establishes a connection with the database as it is starting yup. The priority of this case is "High".

3.1.2 Stimulus/Response Sequences

1. The user starts the application on their workstation.
2. A connection with the database is established and the user enters their credentials.
3. The credentials entered are compared with those in the database to grant/deny access to the system.

3.1.3 Functional Requirements

REQ-1: Connection to a LAN

It ensures that the database can be accessed by the application as it runs.

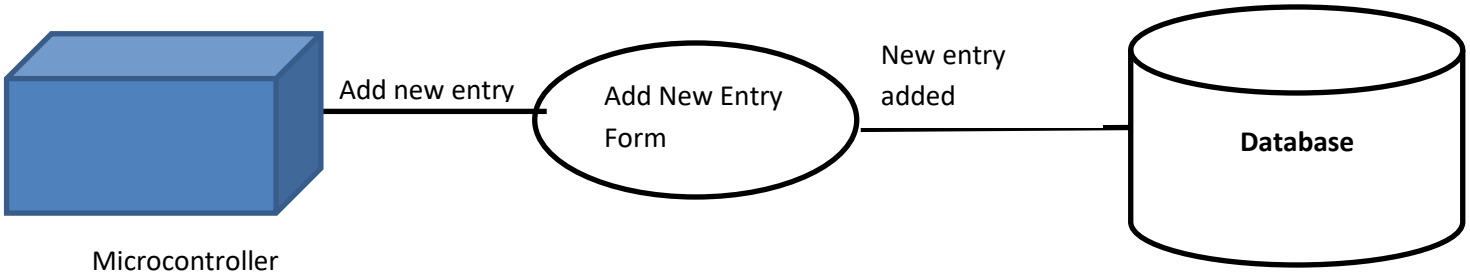
REQ-2: User Interface

It will be implemented as forms. They will be used for data input.

REQ-3: Access Control

It will be enforced by authentication of the credentials obtained from a user.

3.2 Creating New Entries



3.2.1 Description and Priority

It entails the entry of the main data needed for the system; water levels in the reservoirs. It is done automatically by the microcontroller into the database through the interface which can run in the background. No entries by a person can be accepted in this form. It is of "**High**" priority because it encases the core business of the system.

3.2.2 Stimulus/Response Sequences

Connection to the database is necessary in order to use this functionality as well as access to the new entry form(s).

Project Manager

1. The user clicks the "Add New Attendant" button after logging into their account the system.
2. A new form appears with various data fields that are to be filled with information from the new member.
3. Data entry is complete; the data will be saved in the database by clicking "save new record".

Microcontroller

1. Connection to the user interface is established.
2. It starts transmitting the water levels in the reservoir in real time to the user interface.
3. The user interface adds a time stamp to the data streaming into it before sending it into the database for storage.

3.2.3 Functional Requirements

REQ-1: Connection to a LAN

It ensures that the microcontroller can access the user interface and that the database can be accessed by the application as it runs.

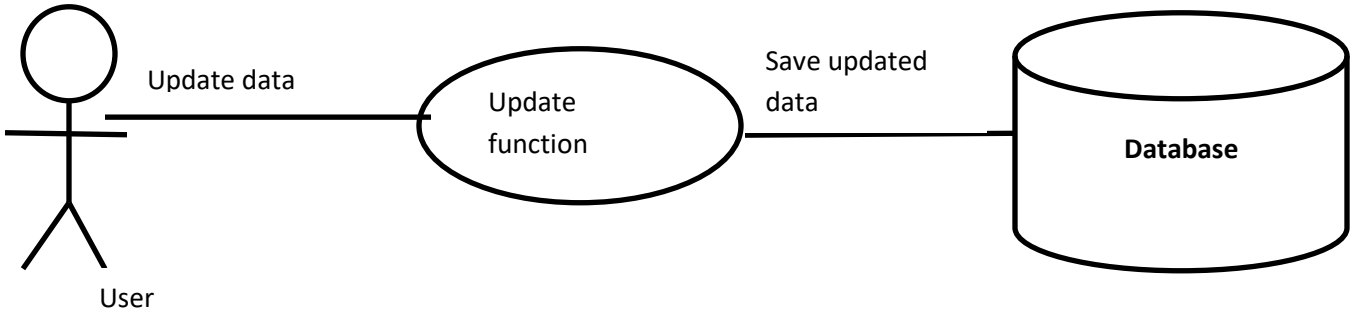
REQ-2: User Interface

It will be implemented as forms. They will be used for data input.

REQ-3: Accurate data

It will be paramount. It can however be changed at any point of time to maintain it's timeliness and accuracy.

3.3 Editing/Updating Data



3.3.1 Description and Priority

There is need to alter the information that has been stored in the database. This may be due to a change in an employee's information or the organization's policy that requires addition/truncation of data. Its priority could be termed as low since the rate of having erroneous data will be very low. Editing of data collected by the microcontroller however will be prohibited.

3.3.2 Stimulus/Response Sequences

Connection to the database is necessary in order to use this functionality as well as access to the new entry form(s) because this is where the editing function is located.

Project Manager

1. The project manager logs into his/her account.
2. Selects “Change Employee Details”
3. Selects the employees details he/she wants to edit
4. Saves the information

3.3.3 Functional Requirements

REQ-1: Connection to a LAN

This will enable it to connect to the database.

REQ-2: User Interface

It will be used to collect the fresh data and send it to the database to update the existing data.

4.0 External Interface Requirements

4.1 User Interfaces

It is a description of how the system will interact with its users and other software as it is being run. This system will use Visual Basic forms as an interface with the users and will be run from any workstation within the ward’s premises.

4.1.1 Graphical User Interface (GUI)

Normal users will use the system through a GUI while the more experienced users will access the database directly using SQL (Structured Query Language) commands on the database’s command line console.

4.1.2 Buttons

They will be easy to see and their use very clear to the user as they will be having simple labels to state their functions. They will be widely used in the system to eliminate the need of menus which take time for one to get accustomed to.

4.1.3 Messages

There will be some specific types of messages:

- *Error messages:* They will alert the user on actions that are not supposed to be taken. For example, entering an alphabetical character into a numerical input field.

- *Informational:* They will give information on the state of the system at a particular point in time. For example: if a transaction is successful.
- *Warning:* They will inform of sudden situational changes that need immediate action to be taken by the user or if a user is not close, the system will take up the action itself.

4.2 Hardware Interfaces

4.2.1 Local Area Network (LAN)

It will provide a platform on which communication will take place between the system's components. This will include networking equipment such as: routers, cables and switches alongside the topology that will be implemented to enable the system to access the database.

4.2.2 TCP/IP Protocols

They will be used to identify the workstations within the LAN hence giving them access to the database via the server. It will also allow the microcontroller to send data to the available workstation for monitoring and storage.

4.2 Software Interfaces

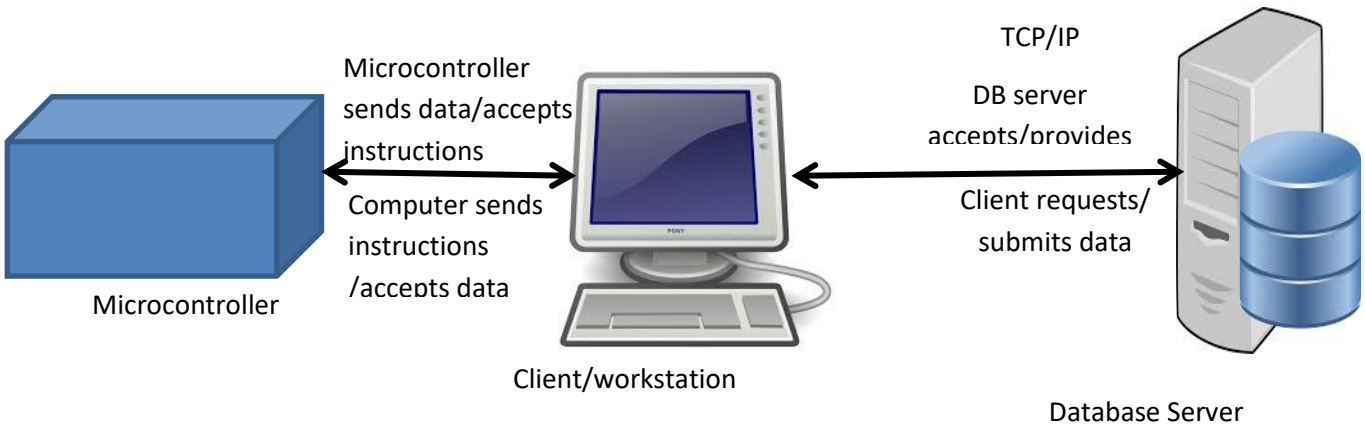
4.2.1 Operating Systems

It is software that facilitates the basic functions of a computer. In this case, it is advised that the operating system be Windows. As for the server, it will be windows 2012.

4.2.1 Database

The database that will be implemented will be MySQL.

4.3 Communication Interfaces



It will utilize a 3-tier processing architecture where there the workstation will hold the user interface and business logic, the microcontroller will collect and process some of its data and the database server will be the data repository.

5.0 Other Non-Functional Requirements

5.1 Performance Requirements

The system will require the following in order to function at it's peak:

1. At least 1GB RAM for workstations and at least 4GB for the server.
2. At least windows 7 and higher for the workstations and Windows Server 2012 for the server.
3. At least 40GB internal storage for the workstation and 200GB for the server.
4. Reliable network connection

5.2 Safety Requirements

Using the system over time will require caution to safeguard the data. The following safety measures will be implemented.

5.2.1 Usernames and Passwords

To access the system, one must have a username and password.

5.2.1 Backup

To ensure that all the data is safe, all data must be backed up in a location that will not be accessible on the network. This is to ensure that in the event the database is corrupted, a backup mechanism is in place.

6.0 Other Requirements

6.1 Reliability

The system will always be available to users at all times.

6.2 Scalability

Upgrading the system will allow growth and able to incorporate even more workstations to monitor more reservoirs and possibly even form a system that is distributed in different geographical areas without affecting performance.

6.3 Maintainability

The system will be open to future upgrades in both features and functionality.

SYSTEM DESIGN SPECIFICATION

Abstract

The motivation behind undertaking this project was the inefficiency of the manual system being used by the Mùgùtha Ward Water Project which resulted in many errors in the data that was in their possession and also degrading the quality of service they were giving their customers which included water shortages and also huge losses in revenue collection.

Therefore, the goal of this project is to automate their manual systems, create a new system that will be able to automate data collection and storage as well as control the functions of the water pumps while giving feedback on how they are performing hence eliminating errors that occur during data collection and computations and overall improvement in service delivery to the residents of Mùgùtha Ward.

1.0 Introduction

A Software Design Document that provides a description of a software product written by a software designer to give a software development team guidance to the architecture of the software product or how the software is to be built. Within this document is a narrative and graphical documentation in the form of use case models, class diagrams, sequence diagrams, activity diagrams and other supporting information that will prove useful to the development of this project as well as wiring diagrams of how the new hardware is to be developed and integrated into the software product.

This Design Document provides in-depth details for the design of the Mùgùtha Ward Water Project Management system for Mùgùtha Ward.

At this stage (system design) the primary objective is producing a design that will meet the expectations of a high-quality information system which are:

(i) Security: It can defend itself against unauthorized personnel trying to access it.
(ii) Reliable: Always available when needed.
(iii) Can integrate easily with other systems.
(iv) Easy to use.
(v) Provides timely and useful data.

In this document, emphasis will be laid on four key areas: data, interfaces, architecture and components of the software under production.

1.1 Purpose

The purpose of this document is to give insight on the design of the system as candidly as possible to enable development of the software and hardware to take off with complete knowledge of what is being built and how.

1.2 Scope

This Software Design Document will cover the basic but critical parts of the system hence acting as a proof of concept that it is feasible for largescale use with emphasis being on data collection, manipulating and storage of information. The system will be used in conjunction with existing systems.

This document contains a complete description of the design of the Mùgùtha Ward Water Project Management System. Its basic structure is a Client-Server paradigm. Basic forms will be created and coded in Visual Basic.

Staff that will be designated to use the system will not be able to make changes to the data that has already been entered into the system unless they have access to the database's command

line interface since the data will be automatically collected by the microcontroller in development.

1.3 System Design Constraints

The user requirements' specifications have directly constrained the system design. This describes the functions that are required by the user which must be implemented as part of the design without leaving out the environmental constraints resulting from the hardware and software environment implementation. These include:

- Integration of data from the current into the new system.
- Implementing an interface between two different applications will be difficult to implement.
- Hardware limitation such as no ownership of a computing device.

1.4 Design Goals and Objectives

The goal of the Mùgùtha Ward Water Project Management System is to design a system which delivers the functions required by Mùgùtha Ward Water Project to support its business. The importance of software design is:

- Allows proper understanding of the system hence requirements can be met.
- Prevents redundancy and increases reusability.
- Helps to mitigate unknown risks.
- The resulting system will be stable with very low probabilities of failure.

Objectives

- **Security:** The system should be able to implement access control methods on who can view which data.
- **Ease of use:** The system should be user –friendly.
- **Flexibility:** The system should enable new requirements of Mùgùtha Ward Water Project to be incorporated without too much complexity.
- **Availability:** The system should always be available for use at any point in time it is needed for use.

1.5 Document Overview

The Software Design Document has been divided into sections and subsections which are:

- **Section 2, Architectural Design:** It specifies all the design entities that collaborate to perform all the functions included in the system.
- **Section 3:** contains File and Data Structure Design
- **Section 4:** Contains Normalization whose purpose is to organize data in a way that optimizes logical storage of data.

- **Section 5:** It is the conclusion of the System Design Document. It indicates how the software will be meeting its user requirements.

2.0 System Architecture

2.1 Introduction

Systems Architecture is a response to the conceptual and practical difficulties of the description and the design of complex systems. It can be viewed as a blueprint of a software system. Here, a software developer analyzes the requirements of the system, determines the components the system will need as well as support the project even after its production is complete.

2.2 Client-Server Model

It is a distributed application structure that partitions tasks between providers of a service (server) and service requesters (clients). It is composed of the following:

(i) **Servers:** they offer services to subsystems. There are many types of servers, examples include: print servers which avail printing services, database servers which provide access to databases and file servers which offer file management services.
(ii) **Clients:** They are software that request services from a server.
(iii) **Network:** It is a connection between two or more computing devices for the purpose of information and resource sharing. It is used by clients to request and obtain services from a server.

Clients access the services provided by a server through remote procedure calls and must be aware of which services a particular server offers. In the case of a server, it must be able to know how many requests it can handle.

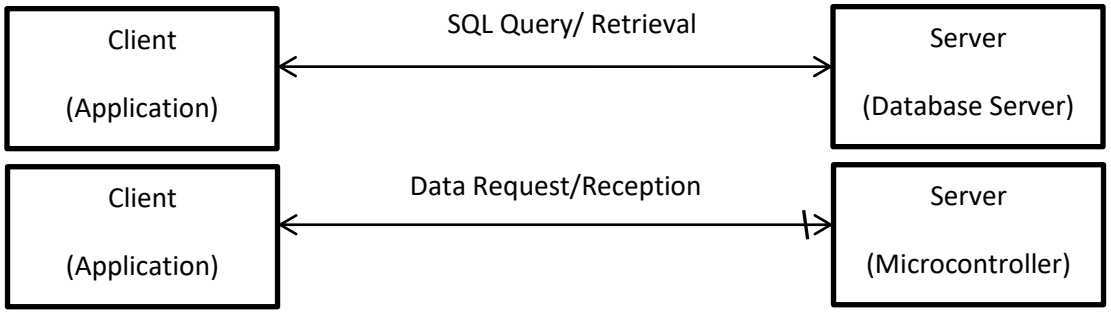
2.3 Design Approach

To solve a problem, it is easier if the said problem is first broken down into smaller and more manageable proportions to better understand it. This is the best design approach there is in developing software also known as modularization. This is breaking down a complex system into smaller subsystems.

Top-down approach design will be implemented as it describes the overall control architecture of the application before designing the individual modules.

2.4 Architectural Design

It's a concept that focuses on the components of a system and unifies them into a functional fully integrated whole. The proposed system will be developed using a two-tier architectural program. The module of the system being run on workstation will hold the bulk of the system as it will house both the user interface and the application logic while the server will run the database only.



2.5 Logical Design

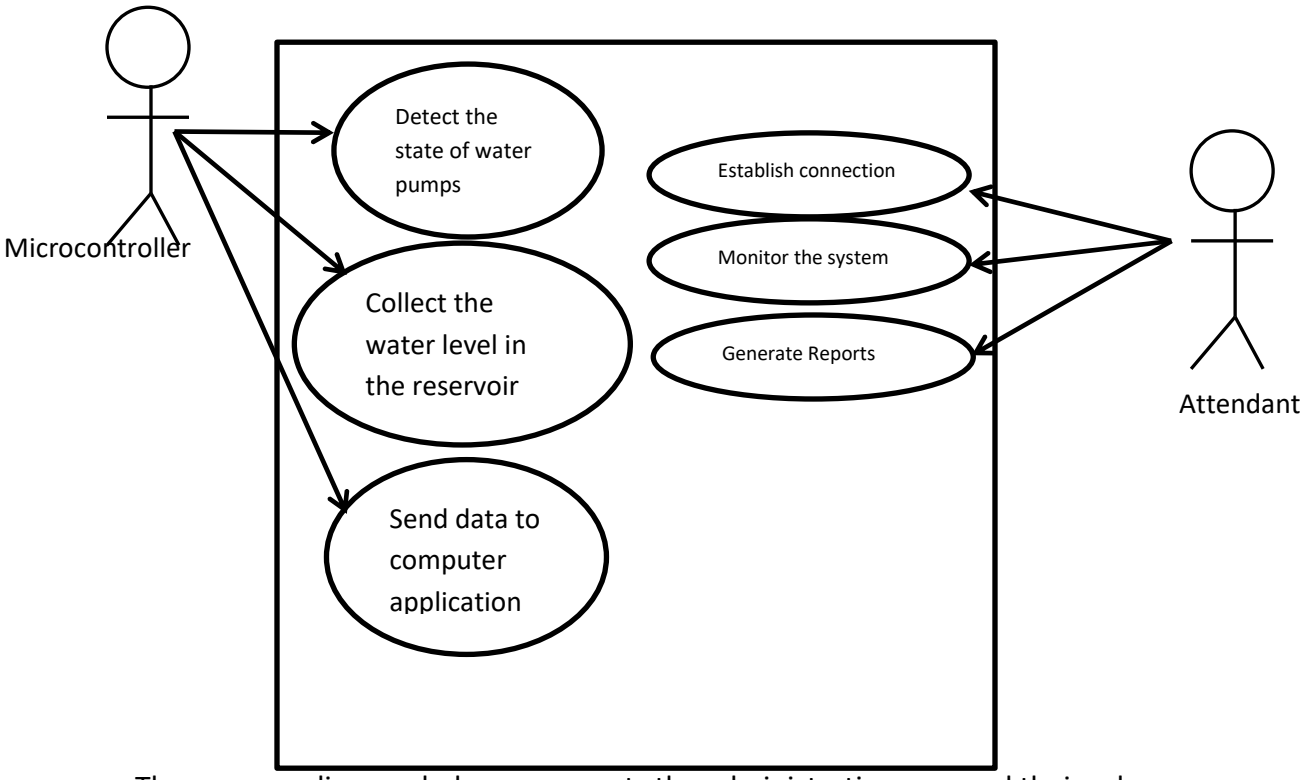
Its main concern is the processes being performed by the system. It describes what is to be done and how it will be done and is not in any way associated with hardware and software of the system.

2.6 Use Case Diagrams

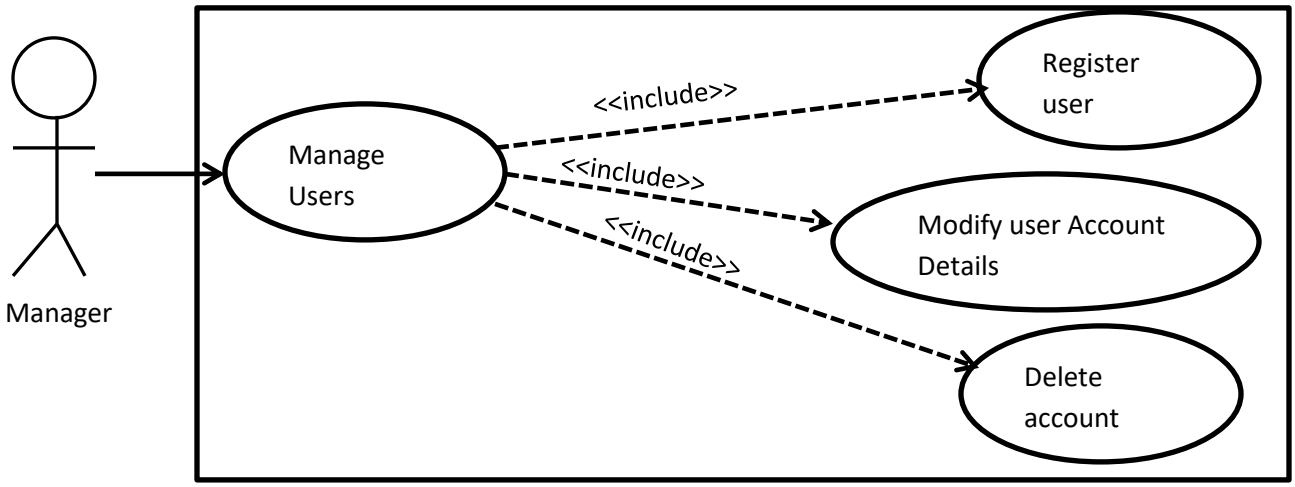
They are at times referred to as behavior diagrams used to describe a set of actions (use cases) that a system (subject) should or can perform in collaboration with one or more external users of the system (actors). There are three main actors in this system namely:

- **Manager:** Holds an administrative role in the system with the full ability to oversee the various users and accounts.
- **Attendant:** Will be in charge of overseeing that the system is running smoothly and that the data is being received by the application from the microcontroller and being sent to the database.
- **Microcontroller:** Will act as the data entry clerk as it will be collecting data on the state of the pumps (Either they are pumping water into the reservoir or pumping water to the residents) and the water levels in the reservoir and sending this data to the computer application for storage in a database.

Below is a use case diagram depicting the attendant and microcontroller actors and their roles:



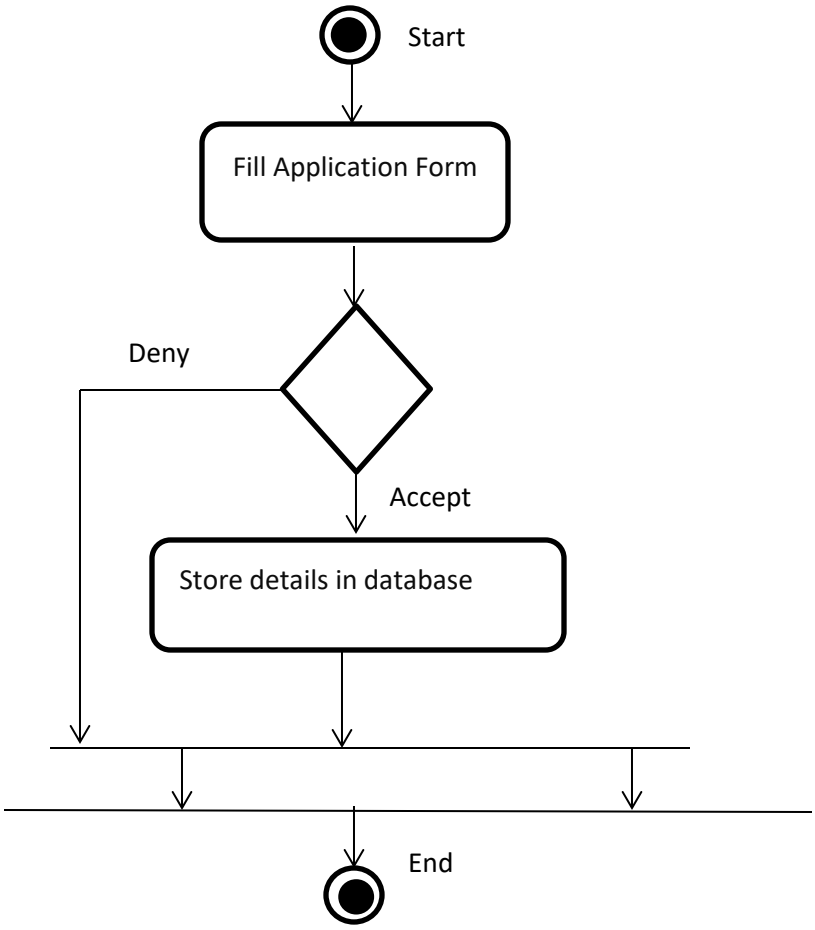
The use case diagram below represents the administrative user and their rolves



2.7 Activity Diagrams

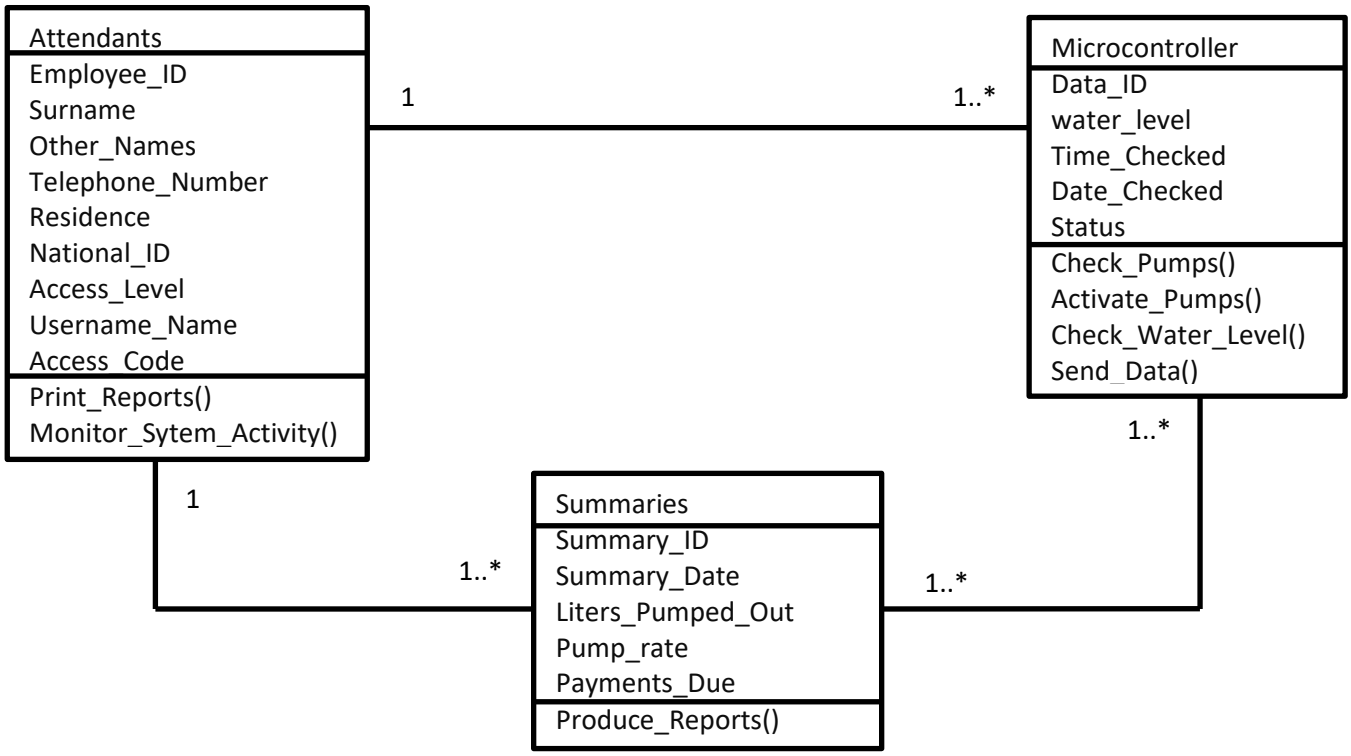
They are basically similar to flowcharts and depict flow from one activity to another which can be described as an operation of the system. The activity could either be sequential, branched or concurrent. Its purpose is to capture the dynamic nature of a system.

Below is an activity diagram of adding an Attendant or user into the system:



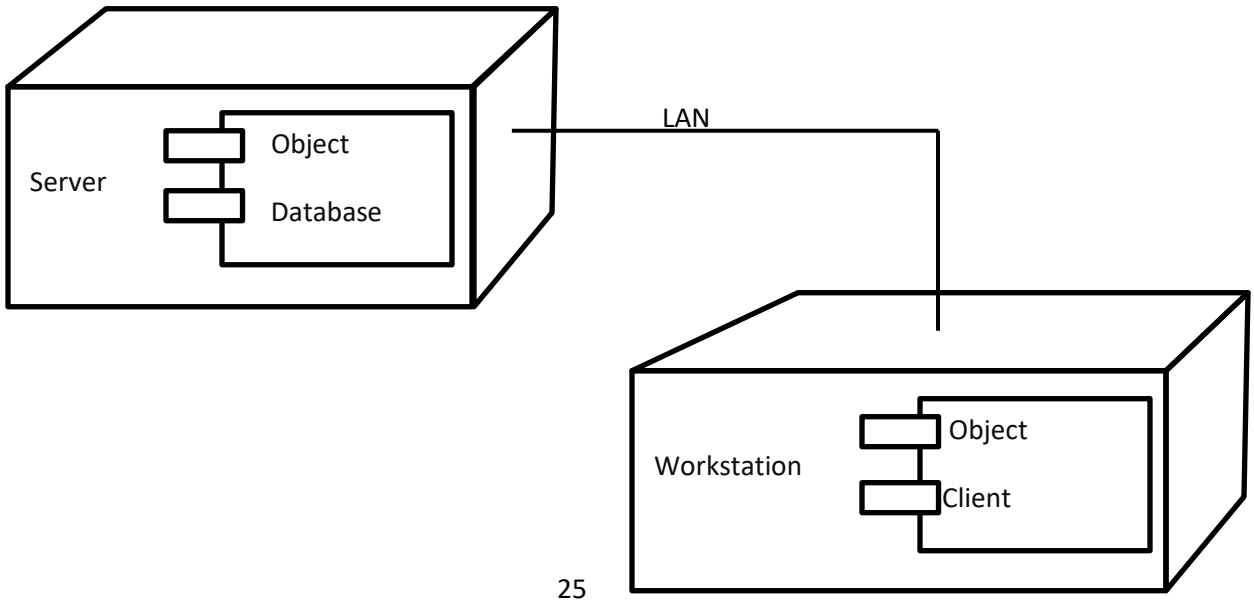
2.8 Class Diagrams

They depict a static state of the system by use of classes, their attributes, methods/operations and relationships between objects.



2.9 Deployment Diagrams

They are used to visualize the topology of the physical components of a computer system where the software components are deployed.



3.0 Database Design

3.1 Introduction

A database is an organized collection of data stored in a way that minimizes duplication of information and provides consistency of data and access for users. The standard application program interface for many relational databases is Structured Query Language (SQL). These are commands/statements that are used to create databases, insert and modify information into the database, generate reports and even delete a database.

The database for this system will be implemented by use of MySQL 5.6.

Reasons for choice of MySQL 5.6

- On-demand scalability
- It is globally recognized as one of the most secure database management systems globally.
- Assurance of 24/7 availability
- Low total cost of ownership due its easy management which saves time during trouble-shooting.
- Meets the needs of demanding applications while providing optimum speed and unique memory caches for enhanced performance.

A good database must have the following qualities:

- **Atomicity:** If one part of a transaction fails, the whole transaction fails and the state of the database remains unchanged.
- **Consistency:** any data entered into the database is valid according to all the defined rules.
- **Isolation:** An incomplete transaction remains invisible to other transactions until it is complete.
- **Durability:** Once a transaction is committed to the database, it will remain so even in the event of a failure or errors in the database.

3.2 Normalization

It is a process of decomposing tables to eliminate data redundancy and unwanted characteristics such as insert, update and delete anomalies. There are several steps involved where data is put into tabular form and duplicated data is eliminated form related tables.

The table below shows the whole process from the un-normalized for to the third normal form.

UNF	1st NF	2nd NF	3RD NF
Employee_ID	Employee_ID	**Employee_ID**	Employees
Names	Sirname	**Data_ID**	**Employee_ID (PK)**
Telephone_Number	Other_Names	**Summary_ID**	Sirname
Residence	Telephone_Number	Sirname	Other_Names
National_ID	Residence	Other_Names	Telephone_Number
Access_Level	National_ID	Telephone_Number	Residence
Username	Access_Level	Residence	National_ID
Access_Code	Username	National_ID	Access_Level
Data_ID	Access_Code	Access_Level	Username
Water_Levels	Data_ID	Username	Access_Code
Time_Checked	Water_Levels	Access_Code	
Status	Date_Checked	Water_Levels	Water_Level
Summary_ID	Time_Checked	Date_Checked	**Data_ID (PK)**
Summary_Date	Status	Time_Checked	Water_Levels
Liters_Sold	Summary_ID	Status	Date_Checked
Pump_Rate	Summary_Date	Summary_Date	Time_Checked
Payments_Due	Liters_Sold	Liters_Sold	Status
	Pump_Rate	Pump_Rate	
	Payments_Due	Payments_Due	**Data_Summary**
	Payments_Received	Payments_Received	**Summary_ID (PK)**
	Payement_Deficit	Payement_Deficit	Summary_Date
			Liters_Sold
			Pump_Rate

Payments_Due

Payments_Received

Payement_Deficit

3.3 Database Description

Employees Table

Field Name	Data Type	Size	Remark
Employee_ID	Integer	5	Primary Key
Sirname	varchar(20)	20	
Other_Names	varchar(100)	100	
Telephone_Number	varchar(13)	13	
Residence	varchar(20)	20	
National_ID	varchar(10)	10	
Access_Level	varchar(20)	20	
Username	varchar(40)	40	
Access_Code	varchar(100)	100	

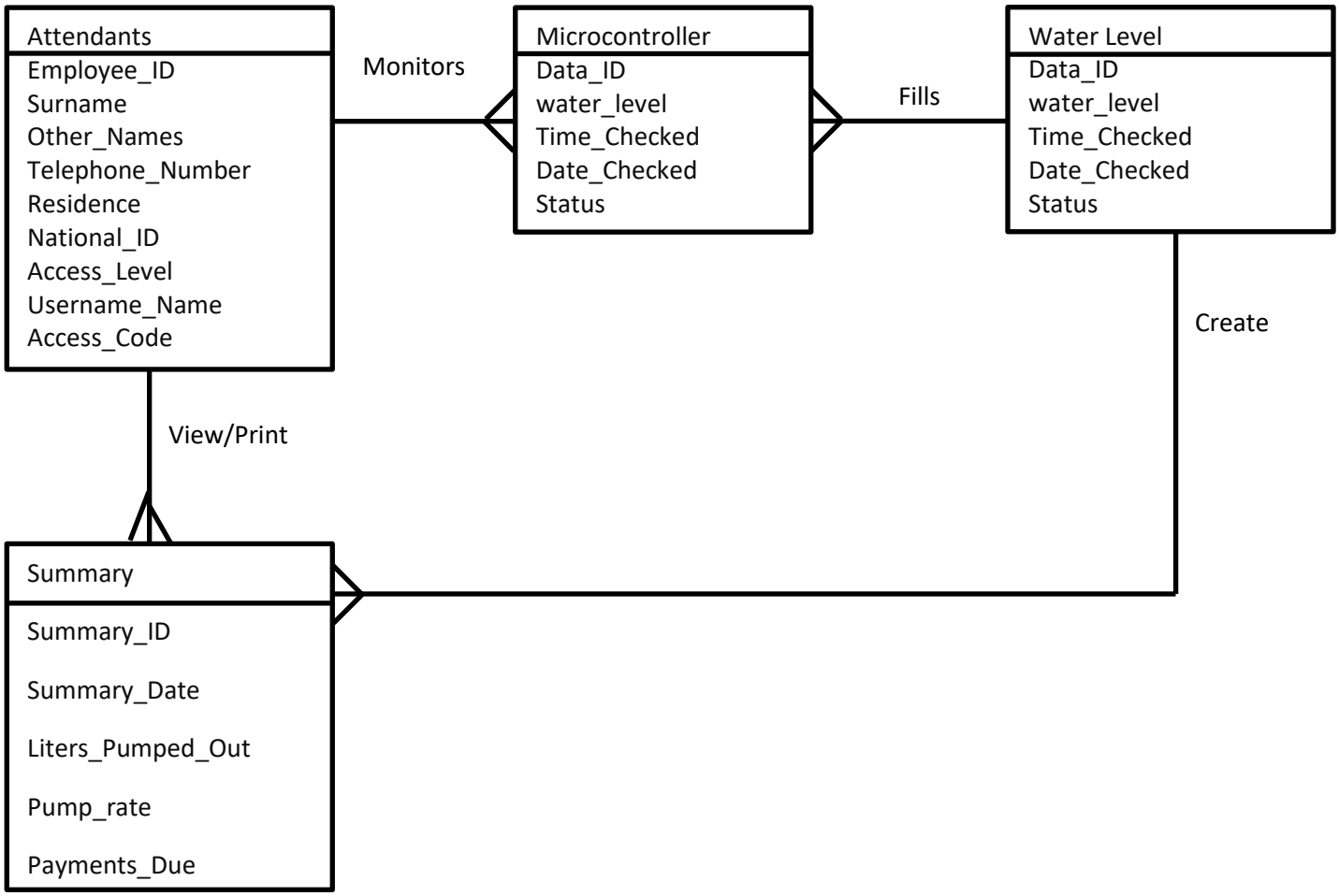
Water Levels Table

Field Name	Data Type	Size	Remark
Data_ID	Int	100	Primary Key
Water_Levels	Double	6,2	
Date_Checked	Date		
Time_Checked	time		
Status	Varchar	20	

Summaries Table

Field Name	Data Type	Size	Remark
Summary_ID	Int	100	Primary Key
Summary_Date	Date		
Liters_Sold	float	6,2	
Pump_Rate	float	6,2	
Payments_Due	float	6,2	
Payments_Received	float	6,2	
Payement_Deficit	float	6,2	

3.3 Entity Relationship Diagram



4.0 User Interface Design

4.1 Internal Machine Interfaces

A client-server network will first need to be created before the system is installed in both the client and server. The database will be installed in the server and the user interface in the client machines. Records added or updated by the user will automatically reflect on server.

The microcontroller will also act as a server as it will be collecting and sending data to the workstations for processing and storage in the database.

4.2 External System Interfaces

Resources such as printers will be available on the network while computer accessories such as a keyboard and mouse will only be usable on one computer only. The server and workstations should have antivirus installed to protect them against malware, viruses and other malicious software that will threaten their operations. The antivirus should be frequently updated online hence internet access will be necessary. The microcontroller must have its sensors to be able to operate and send data. These include: a motor driver for controlling the pumps, and a sonar module for determining the height of the water in the reservoir.

4.3 Human Interfaces

All the workstations will allow the users to interact with the system after they have been authenticated to do so. The server however will not be accessible to them. This is because it will have a command-line interface which only an expert will be able to handle. It will also minimize the possibility of unknown errors from occurring. Users with the level “Attendant” will also not be able to enter any data since data entry will be handled by the microcontroller but they will interact with the microcontroller using buttons on the interface.

4.4 Interface Design

This depicts how users will be able to interact with the system to perform the actions they require. They will do so via forms made using Visual Basic.

4.4.1 Description of the user interface

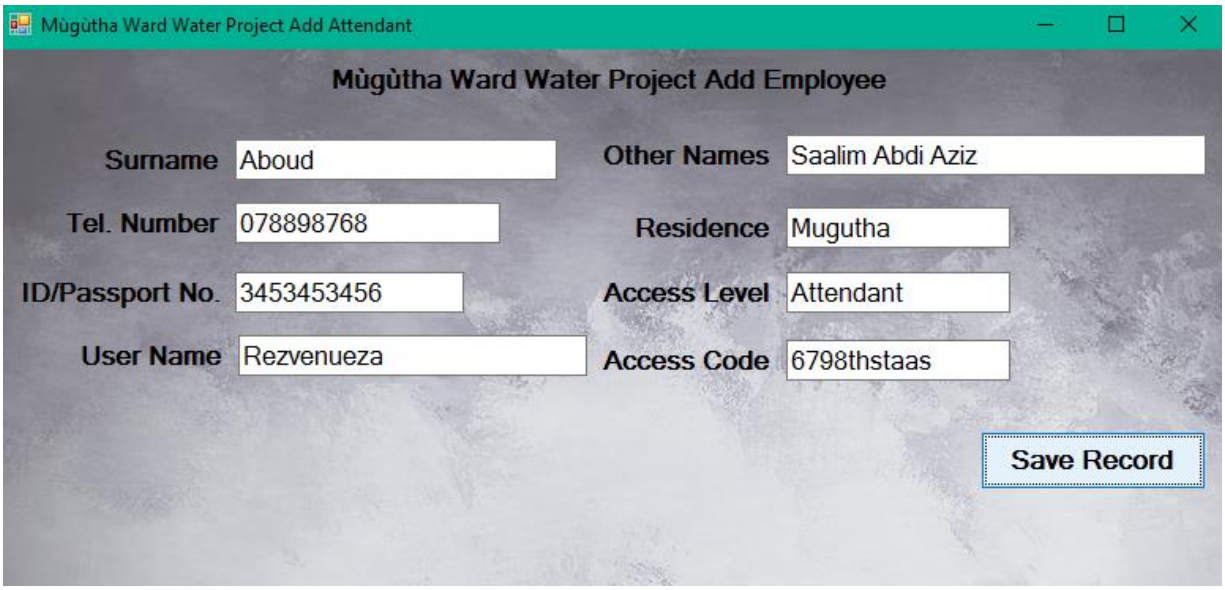
- **Command buttons:** They will issue commands to the system such as: open, save, edit, delete, search etc.
- **Command line:** The user can write SQL statements to query the database.
- **Pop-ups:** They will serve as error messages telling the user what error is occurring and why.

4.4.2 Description of the user interface

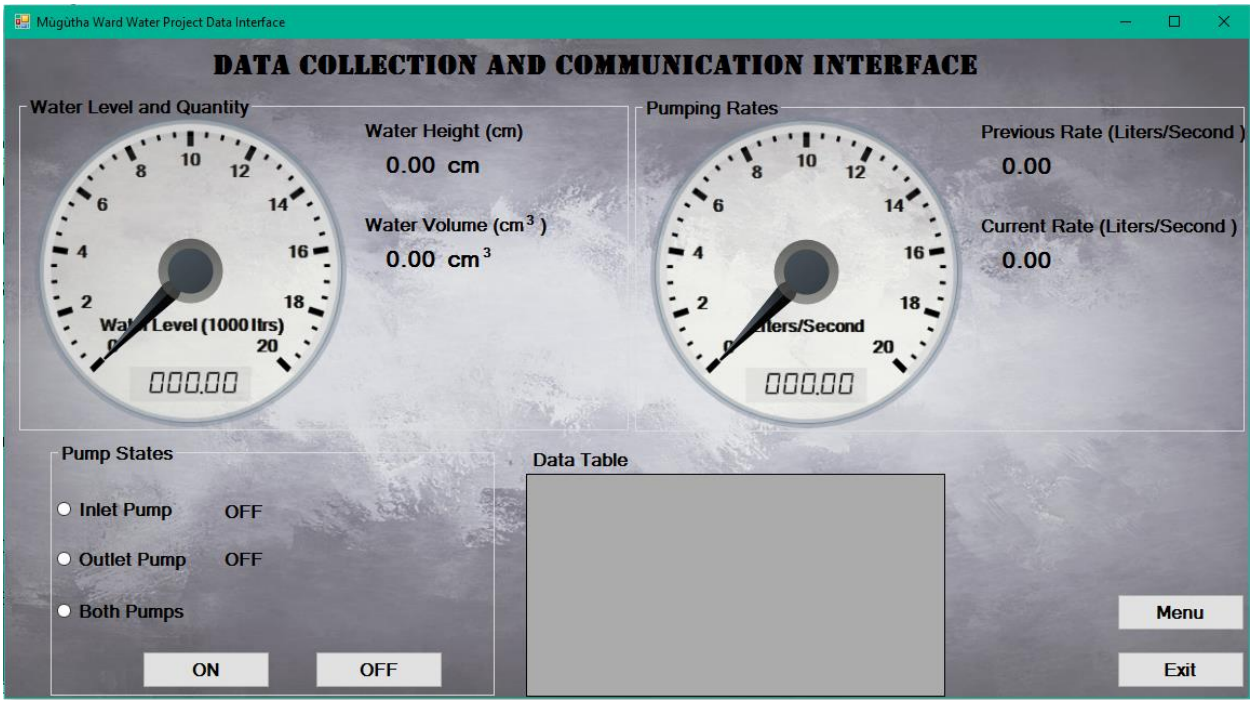
Screen images depict what the user will see when performing certain tasks in the system. Below are some of those designs that will be used in the system.

4.4.2.1 Login Form

4.4.2.2 Add New Employee Form



4.4.2.3 Data Collection Form



4.4.3 Components Available

- **Save Button:** Used to save records.
- **Search button/textbox:** Searches for a specified record from within the database.
- **Validate button:** Used to cross-check data entered with that in the database.
- **Delete Button:** Removes unwanted information from the database.
- **Update Button:** Saves a record that has been successfully edited.
- **Exit button:** Closes down a form and automatically opens the previous form or closes down the interface completely. Hence restart is required.

4.4.4 Software Context

The system will include two main modules: Attendant and Manager modules. Each module will have specified functionality with each being different from the other or sharing some degree of similarity in some sub-modules.

4.4.5 Expected Software Response

The system is expected to run smoothly without any fatal errors that may cause its operation to halt completely or crash. This is because the system has undergone vigorous testing by the system designer and will be handed to qualified team of testers. But in case of any unexpected errors during testing, they will be corrected before implementation by the client.

4.4.6 Packaging and installation

The system on completion will be packaged neatly into a compact disc and copyrighted. It will be accompanied by its documentation.

It will also be accompanied by the hardware required for data collection i.e. the microcontroller and its various components as one unit.

5.0 Conclusion

The system will be able to fulfil the described user requirements and tests plan documents will be designed to help verify this argument.

6.0 References

1. Frank B. Watts (2004) Engineering documentation control handbook
2. En.tekstenuitleg.net/articles/software/database-design-tutorial.html
3. www.tutorialspoint.com
4. Alan Dennis, Barbara Harley Wixom, Ronerta Marie, Roth Hardcover (2012) Systems Analysis and Design 5th Edition

TEST PLAN

Abstract

This document shows in detail all the testing that will be done on the new system for Mùgùtha Ward so as to ensure that the system meets the users' requirements. A test plan will be drawn to show testing schedules of testing and the methods that will be implemented to ensure successful module testing.

1.0 Introduction

Software testing is the process of running software with the aim of finding bugs within its code, checking to ensure that the software meets its requirements and is running as expected. In this case, the hardware developed will also be tested to ensure that it is working and integrating well with the software.

With increase of the complexity and size of software each day, it is paramount that an organization assesses the new software products it produces will be accepted by the intended users, purchasers and other stakeholders. Software testing is the process of attempting to make this assessment.

1.1 Goals and Objectives

The goal of the test plan is to:

- Ensure the end result meets the business and user requirements.
- Ensure the system satisfies the stated functions in the SRS (System Requirement Specification).
- Find and eliminate errors in the system.
- Delivering a high quality product.

1.2 Purpose

This document serves as a draft test approach for the Water Reservoir Management System. There will be three stages in the preparation for the test:

- **Test Approach:** Sets the scope of system testing, the overall strategy, the activities to be completed, resources needed and the methods to be used to test the release.
- **Test Planning:** Details the activities, dependencies and effort needed to conduct the test.
- **Test Conditions/Cases:** States the tests to be applied, data to be processed and the expected results

1.2 Purpose

This document serves as the draft test approach for the Mùgùtha Water Reservoir Management System.

Preparation for the test consists of the following stages:

1. **Test approach:** It sets the scope of testing the system, general resources needed, strategy to be taken up, activities to be completed and the methods to be employed to test the release.

2. **Test Planning:** It profiles the activities, dependencies and effort required to conduct the system test.
3. **Test Conditions/Cases:** documents the tests to be applied, the data to be processed, the automated testing coverage and the expected results.

1.3 Statement of the scope

- Before granting a user access to the system it must first prompt the user for a username and password.
- The system to eliminate redundancy of unique records by implementing the use of primary keys.
- Data searches and processing should be fast and efficient.
- The system will be resistant to failure and will correct itself in case of any errors during operation.

1.4 Major Constraints

- **Time:** The system is very extensive hence full testing will require more time than is available.
- **Funds:** Thorough and satisfactory testing will require a large amount of money which may not be practically possible.
- **Cooperation:** The testing team and the users of the system need to be ready to work together during this exercise. Their availability is key but they may not avail themselves at the required time hence important parts will be overlooked.

1.5 Formal Reviewing

There will be several formal review points before and during system tests which is very important to ensure that the product is of required quality.

1.5.1 Formal Review Points

1. Design Documentation
2. Testing Approach
3. Unit Test Plans
4. Unit Test Conditions and Results
5. System Test Conditions
6. System Test Progress
7. Post System Test Review

2.0 Test Plan

It is an organized approach to testing a specific hardware or software product. It contains a detailed understanding of the eventual workflow. With significant input from test engineers,

the test plan is a strategy that is used to check whether the product (either hardware or software) meets design specifications among other requirements.

The goal here is to make sure that the product does what the stakeholders and users intended it to do and that the products does not have any defects.

2.1 Software and Hardware to be tested

The software to be tested is the Mùgùtha Ward Management System for Mùgùtha Ward along with the special hardware developed to integrate with the system and aid it achieve its primary objectives.

2.2 Testing Strategy

The system test will be done in the following steps:

- Unit testing
- Integration testing
- Validation testing
- High-Order testing

2.2.1 Unit Testing

The system is composed of multiple modules. In unit testing, each individual module is tested to verify and validate its results. This is done by the programmer because it requires detailed knowledge of the internal structure of the code.

Area Being Tested	Expected Results	Actual Results	Comments
Passwords	Should mask characters		
Username	Correct usernames		
Numeric fields	Input restricted to numeric		
Alphanumeric fields	Input alphanumeric		
Date	Only specified date structure is accepted.		

Communication with the microcontroller	Should connect to, send and receive data from the microcontroller
Data Processing	Received data is processed and stored.

2.2.2 Integration Testing

This is the phase in software testing where the individual components of the software are combined and tested as a unit. This testing is very necessary in client-server systems because it ensures that both systems are communicating. The process of integration is done by adding together the modules and then running them together.

Area Being Tested	Expected Results	Actual Results	Comments
Add one module after another to the system and check if it interacts well with other modules.	All the modules should interact without any problems.		
Install the system into the client's network to check for collision with other applications also running on the client's network by running the system concurrently with other applications.	There should be no collision between the various applications running on the same network and computers.		

2.2.3 Validation Testing

The software must also be checked to see if it meets the specifications of its intended purpose.

Area Being Tested	Expected Results	Actual Results	Comments
	Forms		
Add	Insert a new record to the database		
Delete	Removes a specified record		
Menus	Must open the required areas.		
Update	Change an existing record		
Save	Enforce changes done to data or commit new data to the		

	database.
Data	
Data should be received from the microcontroller.	Data received should reflect on the concerned interface.
Input data with known output will be entered into the system for processing	The output should match the expected output.

2.2.4 High Order Testing

Several methods of testing will be combined in this test with the aim to test different conditions by utilizing different test methods. These tests will include:

2.2.4.1 Stress Testing

It's a test that is used to ensure that the system will remain stable and will not breakdown in strenuous conditions.

Area Being Tested	Expected Results	Actual Results	Conclusions
Create many concurrent users.	The system should handle the users		
Reduce system resource to bare minimum, e.g. reduce the system's memory.	The system should keep running normally even with diminished resources.		

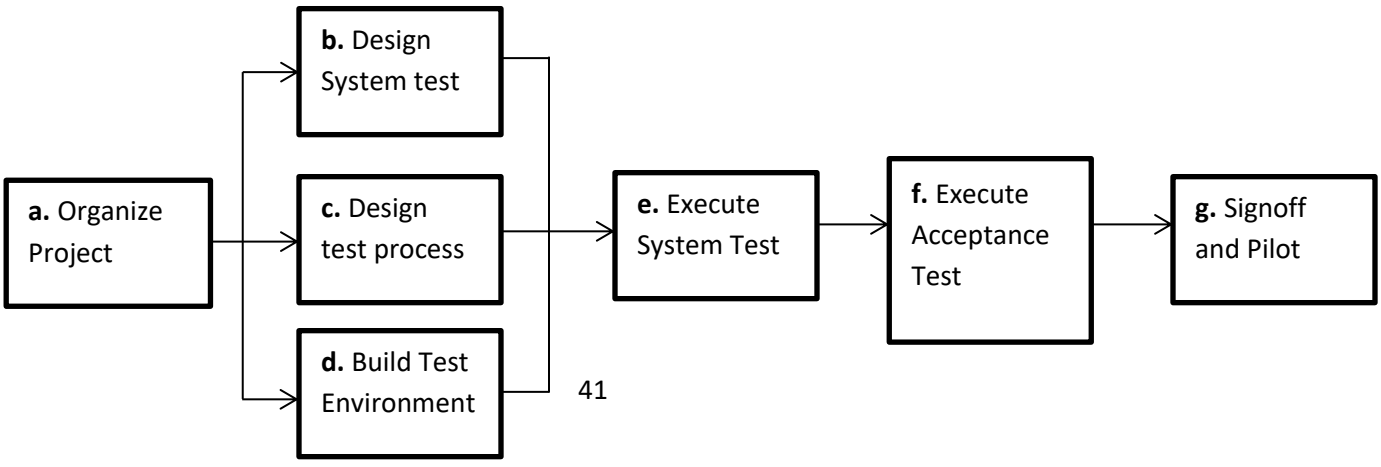
2.2.4.1 Security Testing

It should cover the following areas:

- Confidentiality
- Integrity
- Authentication
- Authorization
- Availability
- Non-repudiation

It will ensure that the security parameters are functional and no unauthorized actions on the data can be performed.

2.3 Testing Process



a. **Organize project:** involves creating a System Test Plan, schedule and test approach and requesting/assigning resources.
b. **Design System Test:** identification of Test Cases, Expected results is done in this stage. This will also include the test data required for this purpose all of which will be done by the test team.
c. **Design Test Procedures:** Procedures such as Error Management, Status reporting and setting up data tables for automate data testing tools.
d. **Build a test environment:** involves requesting hardware, software and data set-ups
e. **Executable integration and acceptance test:** Combination of all the units and ensuring they are functioning well together.
f. **Signoff:** When all requirements are met, the testing can be concluded.

2.4 Test Schedule

Activity	Planned Date From	Planned Date To	Actual Date From	Actual Date to
Design test plan	10th October 2018	15th October 2018		
Design test procedures	17th October 2018	20th October 2018		
Create test environments	27th October 2018	31st October 2018		
Execute predefined tests	1st November 2018	10th November		
Generate test report	15th November 2018	20th November 2018		

2.5 Testing Resources and Funding

A number of resources are mandatory to have in order to test the software entirely. The services of the end users of the system will also be very useful in testing the system thoroughly and complete testing.

2.5.1 Testing Teams

Resource Type	Name	Title
Test Controller	Morris Kimani	Programmer & Developer

Delvin Otunga	Manager
Ismael Bukina	Attendant

3.0 Conclusion

Testing is an integral part of system development because it determines the usability of the software product and ensures that the product is ready for the real world. This therefore means that it has to be planned well and documented with due diligence.

4.0 References

1. Frank B. Watts (2004) Engineering documentation control handbook
2. www.tutorialspoint.com

USER MANUAL

1.0 Introduction

After a system has been successfully produced, there has to be some documentation on how the system works, the actions that can be performed and possible reasons why the new users may face certain errors as they operate the system. The aim of this document is to make all the users conversant with the system with as minimal reading as possible but in an effective way.

1.1 Software and Hardware Requirements

The system will run well on a windows environment with all the workstations connected to a printer. The computers must meet the following requirements in order for the system to run at peak performance:

- 500GB Disk space on a dedicated computer/server
- 4GHz processing speed or higher
- 2GB or more RAM
- CD-ROM installer containing the system ready for installation.
- Water-Level sensor and pump control system.

1.2 Installation

To install the Mùgùtha Water Reservoir Management system, a database administrator and system developer will install and configure the MySQL database in the server followed by installation of the user interface onto the workstation.

For the database there will be a **Read me** text file detailing how the database scripts contained in the CD for creating the database and the tables for the database will be used.

1.3 Getting Started

1.3.1 Login Details

To start the program on the work stations, double click the icon named ***"Reservoir"*** on the desktop, this will launch the system's Graphical User Interface directly to an entry form where the user will be required to click on "**Enter**" on the top left corner.

Figure 2: Welcome Screen

The login form will then appear where a user will be required to enter their details i.e. username and password for validation

Figure 3: Login Form

After logging in as one of the users, a menu will appear showing the actions that the user can perform.

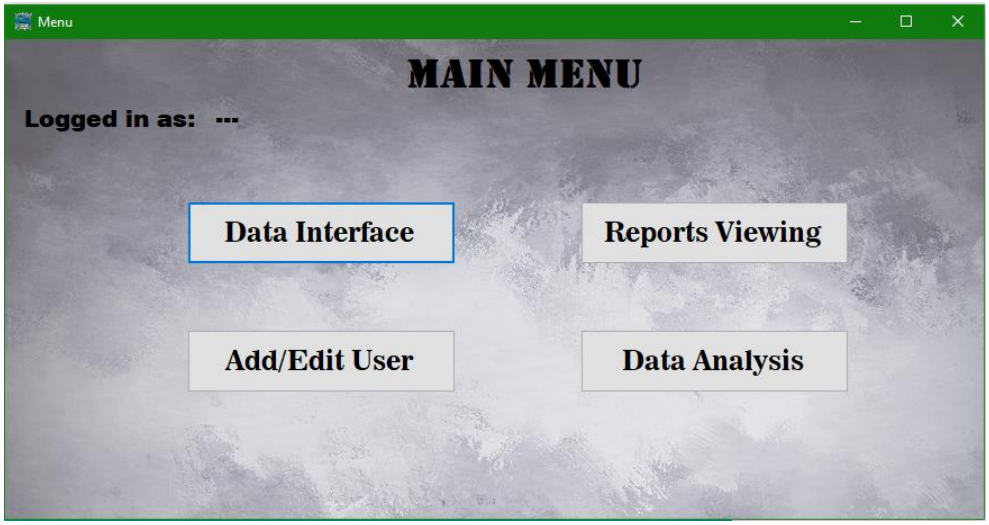


Figure 4: Main Menu

1.3.2 Exit the Program

To exit the program, just click on the small "X" icon at the top right corner of the form.

1.3.3 Add New Records

The system has been divided into separate modules that perform specific actions. In order to add a record into the system:

1. Navigate to the relevant form, in this case, **add/edit user** form.
2. Fill in all the fields then click on the "**Validate**" button to verify the details then proceed to **"save"** to add the record to the database.
3. Click the "**Exit**" button to go back to the menu.

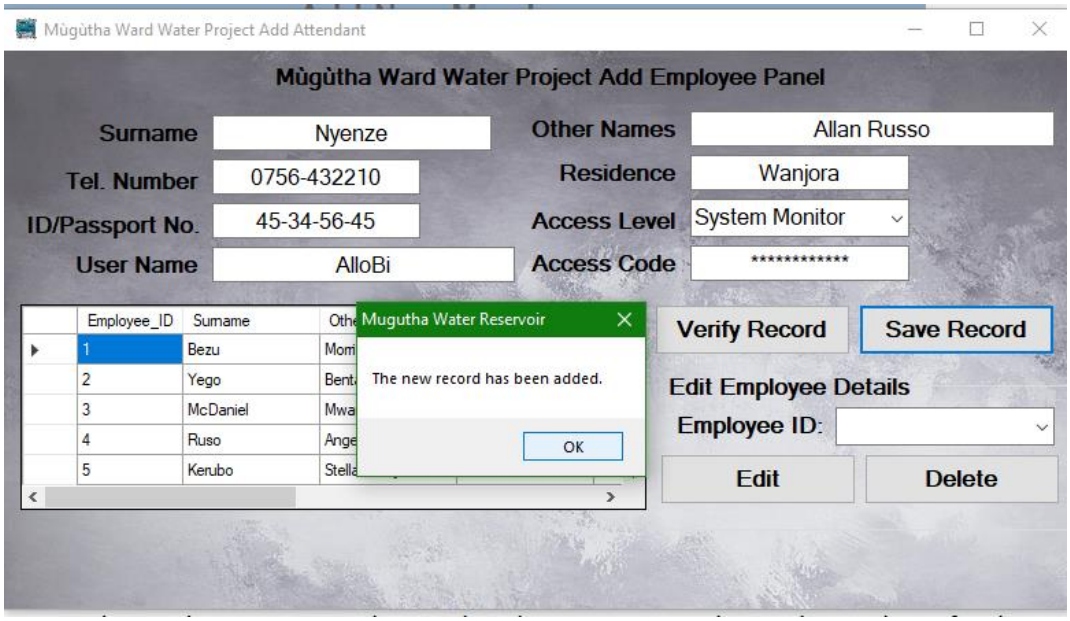


Figure 5: Add new user form

1.3.4 Update Records

The system has a module for altering specific information about a user. This could be due to an error during data entry or a change that the user makes after being entered into the system. To alter a record:

1. Navigate to the **Add/Edit** form from the main menu.
2. Enter the unique ID of the user or select it from the combo box whose details you want to edit.
3. Make the changes to the details then click the "**Validate**" button then the "**Edit**" button.

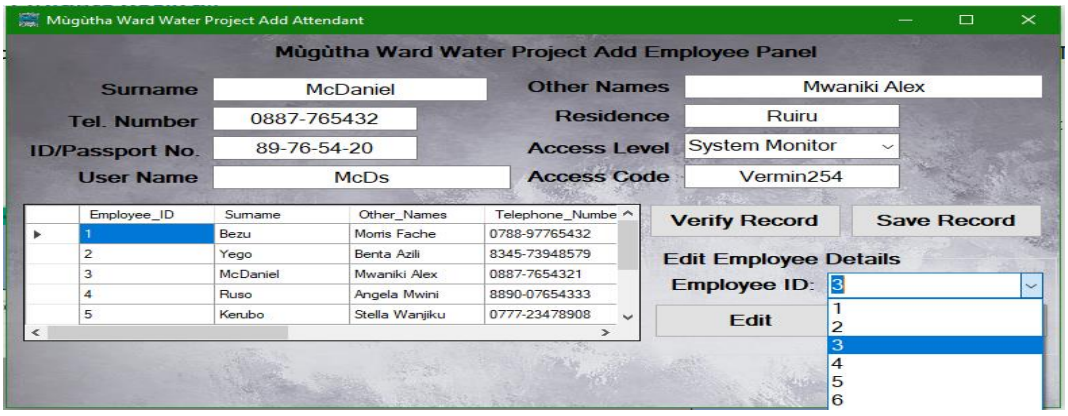


Figure 6: Add/Edit users Form: Editing a user's details

1.3.5 Querying

The system displays specific information on the water level, amount of water pumped and the amount due in the collection interface as well as in the data processing and reports viewing forms, all of which are accessible via the **Main Menu.**

However, the most useful method of searching for data is by using dates; i.e. data that is falling between certain dates. This option is available in the reports and data processing modules. This is due to the immense amount of data being collected per minute on the level of water in the reservoir.

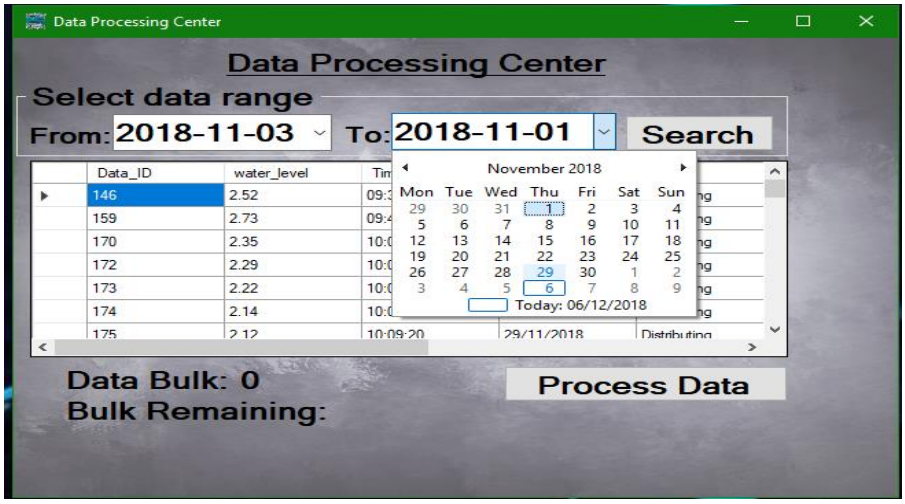


Figure 7: Searching by Dates

1.3.5 Data Collection

This is one of the most important functions of any information system. Data collection in this system is being done by a custom made hardware system which automatically checks the level of water in the reservoir and relays the information in real time to the Graphical User Interface for processing and storage in the database.

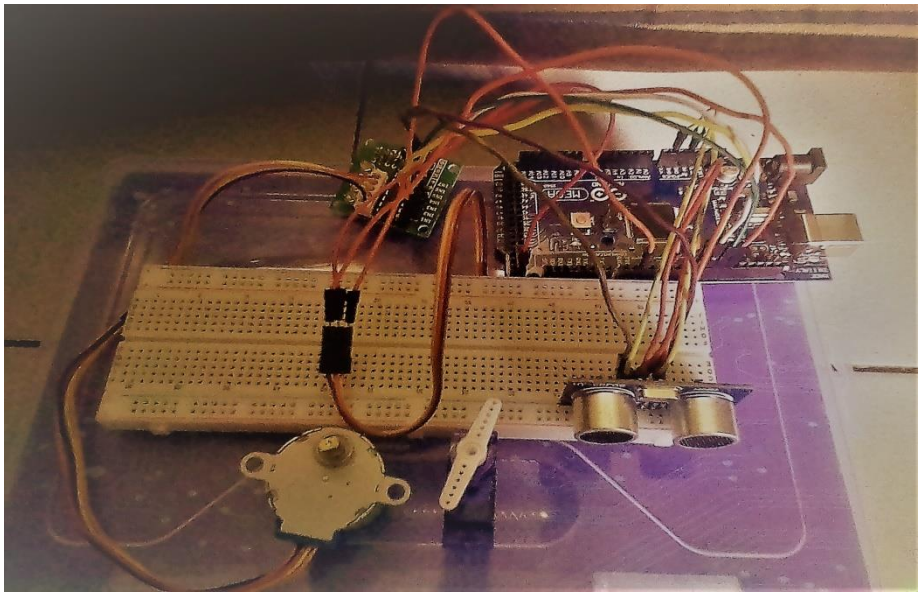


Figure 8: Data Collection and Pump-Control Hardware

After successful connection between the data collection hardware and the user interface by selecting "**COM 12**" on the combo box and clicking the "**Connect**" button beneath the combo box, the interface indicates that it is connected and data can be seen reflected on the interface.

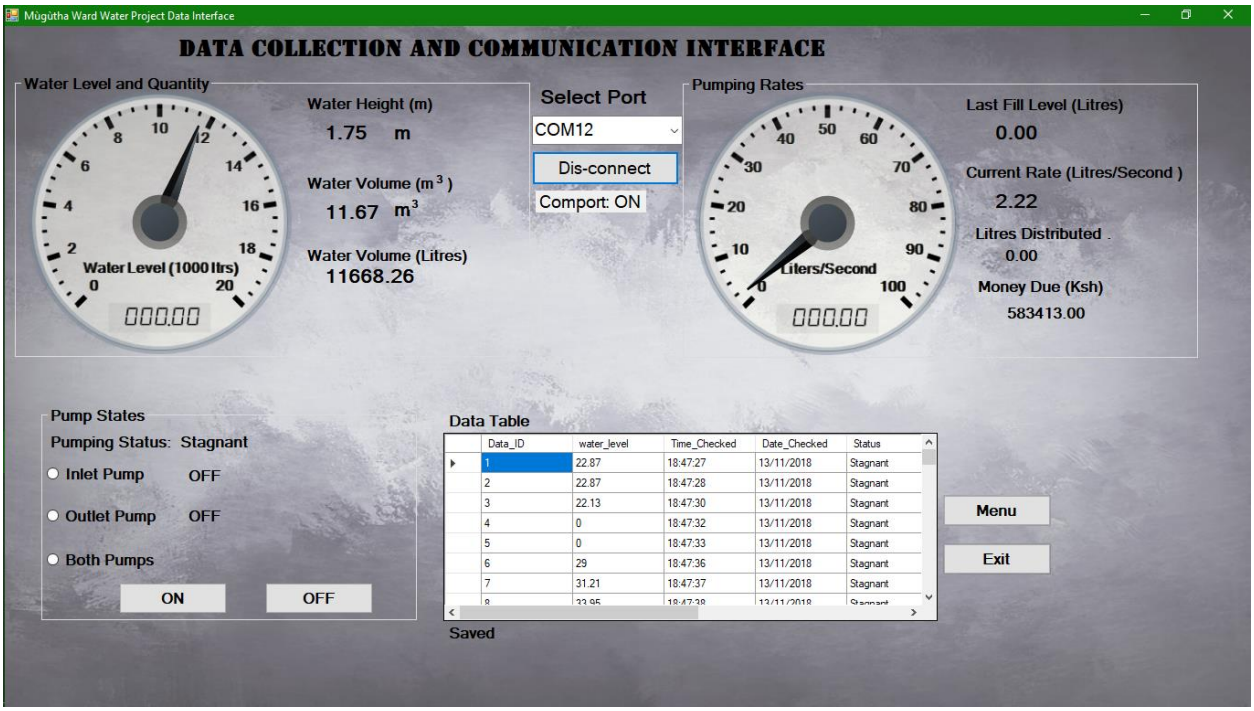


Figure 9: Data Collection Interface while receiving data

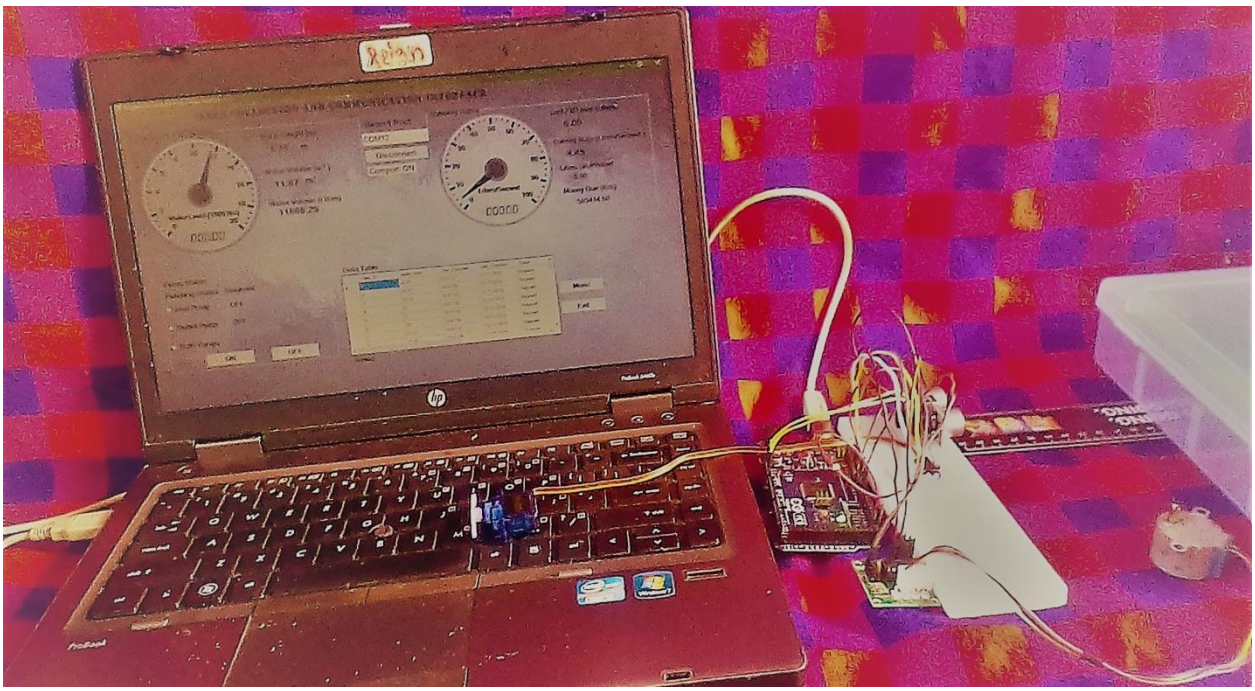


Figure 10: The whole Data Collection setup.

1.4 Threats

1.4.1 Threats to the system

All information systems are surrounded by both internal and external threats which can be intentional or accidental.

Intentional threats are crimes committed with full knowledge of their repercussions. They include:

- Deletion of data
- Destruction of the computers
- Viruses that will attack the systems and corrupt the data.

Accidental threats include:

- Human errors (truncation, omission and transposition)
- Environmental hazards such as fires and storms, power surges and earthquakes.

1.4.2 Control Measures

System developers have managed to mitigate some of the risks by use of authentication to protect the data and verify who is making what changes to the system. Other control measures will include:

- Restricting physical access to the server room
- Anti-virus programs should always be running and up to date.
- Smoke detectors to prevent destruction by fire
- Strong grills and doors to protect the computers against theft.
- Insuring computers against natural and man-made accidents to help the Water Project to recover if any were to occur.
- Maintain a backup for the information within the system.

1.5 Application Maintenance

For Product Maintenance and upgrades, contact:

Morris Kimani

Tel: +254724375433

Email: morrisnjorogekimani@gmail.com

1.6 References

https://www.userfocus.co.uk/articles/usermanuals.html

Douglas Bell (2005) Software Engineering for Students, Fourth Edition