|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MAKERERE**  **UNIVERSITY**      **A SMART IRRIGAION AND CONTROL SYSTEM**  By  BSE 21-05  IoT APPLICATION  DEPARTMENT OF NETWORKS  SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY  A Project Report Submitted to the School of Computing and Informatics Technology  for the Study Leading to a Project in Partial Fulfillment of the Requirements for the Award of the Degree of Bachelor of  Science in Software Engineering of Makerere University.  Supervisor  Dr. Mary Nsabagwa  Department of Networks  School of Computing and Informatics Technology, Makerere University  mnsabagwa@cis.ac.ug , +256-701-124388  OCTOBER, 2021 | | | | | |
| # | Names | Registration Number | Student Number |
| 1 | OKWE ISAAC | 17/U/18975 | 217020401 |
| 2 | KIJJAMBU HASSAN | 17/U/44437 | 217023193 |
| 3 | BYAKATONDA BENARD | 17/U//3802/PS | 217008692 |
| 4 | NIWANDIINDA JOHN MARTINE | 17/U/9138/PS | 217010696 |

**Table of contents**

Contents

[1. Introduction 4](#_Toc84913212)

[1.1. Purpose 4](#_Toc84913213)

[1.2. Scope 4](#_Toc84913214)

[1.3. Document overview 5](#_Toc84913215)

[1.4. Reference Material 5](#_Toc84913216)

[1.5. Definition of acronyms 5](#_Toc84913217)

[2. System overview 6](#_Toc84913218)

[3. System Architecture 7](#_Toc84913219)

[3.2. 3.1. Architectural design 7](#_Toc84913220)

[3.2. Decomposition Description 8](#_Toc84913221)

[3.2.1. Embedded system module 9](#_Toc84913222)

[3.2.2. Web app module 9](#_Toc84913223)

[3.2.3. Data storage module 9](#_Toc84913224)

[4. Data design 10](#_Toc84913225)

[4.1. Data Description 10](#_Toc84913226)

[4.2. Data dictionary 12](#_Toc84913227)

[5. Component Design 14](#_Toc84913228)

[5.1. Data Input component 14](#_Toc84913229)

[5.2. Data transfer component 14](#_Toc84913230)

[5.3. Output component 14](#_Toc84913231)

[5.4. Irrigation component 15](#_Toc84913232)

[5.5. Analysis component: 15](#_Toc84913233)

[5.6. Data storage 15](#_Toc84913234)

[6. Human Interface Design 16](#_Toc84913235)

[6.1. Over view of Human Interface 16](#_Toc84913236)

[6.2. Screen Images 16](#_Toc84913237)

[6.3. Screen objects and actions 19](#_Toc84913238)

# Introduction

## Purpose

The software design document specifies the architecture and system design for the “smart” irrigation and control system. The system is used by farmers to carry out automatic irrigation based on the moisture levels of the soil. It should be noted that irrigation is the artificial application of water to the soil through various systems such as tubes, pumps and sprays. Irrigation therefore, assists in the growing of agricultural crops, watering of livestock, maintenance of vegetation of distributed soils and dry areas during periods of inadequate rainfall[1]. The system therefore is aimed at saving the farmers’ time and resources by making it possible to irrigate their farm lands irrespective of their physical distance from the farms. The intended audience for this system includes, farm managers, farming engineers and agricultural practitioners.

## Scope

The system shall provide a mechanism to trigger irrigation by the farmers irrespective of their destinations. The project will encapsulate the following functionalities, moisture detection from which the irrigation would be based on, storage of irrigation data for further use, analysis based on the previous records from where the decisions would be made, user-interface control mechanism for whether or not trigger irrigation among other controls.

Following are the objectives of this project

1. To design a moisture detection mechanism which will determine whether or not trigger irrigation.
2. To design a mechanism to control water logging in the different parts of the farm land that fall under a victim.
3. To design an interfacing for the farmer to trigger, control and monitor irrigation as well as managing other activities.
4. To provide a data storage mechanism from which analysis can be based upon.

The “smart” irrigation and control system is therefore of the following benefit to its intended audience

1. Farmers will be able to carry out irrigation with ease irrespective of the physical distance from their farm lands, this saves their time.
2. Farmers will use the smart system to make analysis based on the previous records, this makes it possible to know the costs incurred and also determine under what circumstances do they need to prepare in advance given the quantity of water used for certain periods of time
3. System authentication from the user interface is to provide a security mechanism for only the intended user to be accepted to use the system services.

## Document overview

The document has seven sections. The first section is an introduction and contains the purpose of the document, scope, reference material, definitions and acronyms. The second section gives system overview and it describes the proposed system. Section three is the system architecture which describes the architectural design as well as decomposition description of the system. Section four is the data design, which has data description and dictionary. Section five is component design which has an overview of the components of the proposed system. Section six describes the human interface design which contains an overview of the human interfaces, screen images, screen objects and actions of the system intended to be developed. Section seven is the requirements matrix which is used to trace system requirements.

## Reference Material

[1] U. Ighrakpata, “Automation of Irrigation Systems and Design of Automated Irrigation Systems,” *Int. J. Water Resour. Manag. Irrig. Eng. Res.*, vol. 50, no. February, pp. 3–10, 2012, [Online]. Available: file:///F:/ali/payanname/Article/و هﺮﯿﻐﺘﻣ ﺪﻨﭼ ﯽﻄﺧ نﻮﯿﺳﺮﮔر يﺎﻫ شور ﺞﯾﺎﺘﻧ ﻪﺴﯾﺎﻘﻣ ؛ ﻞﻧﻮﺗ ﺮﻔﺣ ﻦﯿﺷﺎﻣ ذﻮﻔﻧ خﺮﻧ ﯽﻨﯿﺑ ﺶﯿﭘ ﯽﺒﺼﻋ يزﺎﻓ ﯽﻘﯿﺒﻄﺗ جﺎﺘﻨﺘﺳا ﻢﺘﺴﯿﺳ يﺮﯿﺼﺑ ﻦﯿﺴﺣ ﺪﻤﺤﻣ ، 2 يﺰﯾر.pdf.

[2] V. Sridharan *et al.*, “The Impact of Climate Change on Crop Production in Water and Energy Implications,” *Water*, no. 2019, p. 24, 2019, [Online]. Available: https://res.mdpi.com/d\_attachment/water/water-11-01805/article\_deploy/water-11-0.

## Definition of acronyms

1. IoT: Internet of Things
2. Db: Database
3. Y: Moisture level of the crop field
4. X: Amount of water flowing or used
5. V: threshold value

# System overview

As farmers continue to experience variations in seasons where there are longer dry seasons as compared to the rainy seasons, irrigation is now becoming a must to every farmer whether small, medium or large-scale farmer [2]. It’s of no doughty that there has been a number of irrigation measures taken such as sprinklers, drip-system and use of water canes, however, all these have been subjected to use of great energy, high costs, time consuming and water logging to victim areas on the farm lands among other limiting factors.

Because of such challenges to the farmers yet there is continued demand for the agricultural products which has caused an increase in the number of agricultural practitioners, there’s need to have a mechanism that could help farmers carry out irrigation at their own convenience and with ease.

It’s by this reason that, we propose the design of an automated “smart” irrigation and control system to improve timely irrigation and control of water logging on the farm lands. The design of this system follows two main modules which are Arduino module and the web-based module.

The Arduino module acts as the main module from which all other components are connected. The working of the system entirely depends on moisture sensor which detects the moisture content of the soil. The results read by the moisture sensor determine whether or not trigger irrigation by the farmer.

The web-based module follows a client-server arrangement from which data read can be stored locally on the client machine, thereby fetching it at the time whenever need be to perform other actions. The data can then be maintained on the cloud for recovery and consistency. All user interactions are through this module. Such user interactions include start or stop irrigation, view irrigation records, analyze farm data among other actions.

# 3. System Architecture

## 3.1. Architectural design

The Arduino module of the system follows a repository architectural style. The sensors (i.e., soil moisture sensors and water flow sensors) send their data to the Arduino. The water pump and solenoid valve contain motors that receive instruction from Arduino.

The web module of the system follows the client server architecture where the client request data from the local server which then process and send reply to the querying client.

The Arduino module communicates to the web module by means of message passing. When the client clicks the ‘on’ toggle button to start irrigation, the instruction is sent to the Arduino through the Wi-Fi module (wireless sensor module). The Arduino then issues commands to the water flow pump and the inlet solenoid valve to irrigate the crop field.



Figure 3.1.1 system architecture for a smart irrigation information system.

The system shall consist of the following modules:

* Embedded systems modules:

This module shall be used to capture data from the crop field through the sensors. It shall also transfer the captured data to the cloud storage through the Wi-Fi module. And finally respond to the command sent by the farmer to irrigate the crop field.

* Web app module:

This module shall be used to perform irrigation to the crop field remotely. It shall also be used by the farmer to perform analysis on the irrigation data.

* Data storage module:

This module shall be used to store irrigation data to the cloud storage. It shall also synchronize the cloud storage data with the one on the local storage.

All the modules combined form a smart irrigation control and monitoring system.

## Decomposition Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Stage 1**  **Input/sensing** | **Stage 2**  **Storage** | **Stage 3**  **irrigation** | **Stage 4**  **Analysis** |
| Sensing crop field | 1  3  2  Sending data to database | Performing irrigation  4  444455 | Analysing irrigation data |

Figure 3.2.1 stages of the module functions.

Table3.2.1. Description of the different stages of the systems functions.

|  |  |
| --- | --- |
| Arrow number | Description |
| 1 | After sensing the soil of the crop field, the system captures the reading of the soil moisture sensor. |
| 2 | The data is then sent to the cloud storage which is then synchronized with the local storage of the farmer. The notification may be sent to the client about the new update in the field data. |
| 3 | If the soil moisture reading is less than the threshold value, the farmer can perform irrigation of the field by clicking the ‘toggle’ button on the web interface. |
| 4 | Using the irrigation data stored in the database over several irrigation dates, the farmer can analyze the data so as to make appropriate decision out of it. |

### Embedded system module

The module consists of three main components as explained below.

* Data input (sensor) component: this component uses the soil moisture sensor to measure the humidity of the soil. It also uses the water flow sensor to measure the volume of the water used during irrigation.
* Data transfer component: this component uses the wi-fi module to transfer data from Arduino to the cloud storage and local storage. Data transferred includes, date and time, quantity of water used and the thresh hold value read by the sensor for each irrigation.
* Output component: this component controls the actual irrigation by regulating the flow of water from the water flow pump and the inlet solenoid valve.

### Web app module

The module consists of two major components, that is, irrigation and analysis components

* Irrigation component: this component trigger irrigation through a clickable on/off button on the web interface. The on/off button on the web interface sends a command to the Arduino to perform the actual irrigation in the crop field.
* Analysis component: this component allows the farmer to view irrigation records, analyze the irrigation data by plotting graphs, charts etc.

### Data storage module

The module consists of two major components, that is, Data storage and data synchronization components.

* Data storage: this component stores irrigation data to the cloud and provide access to this data to the farmer through the web interface.
* Data synchronization. This module synchronizes data between cloud storage and the local storage to ensure consistency in the irrigation data.

# Data design

## Data Description

The IoT project being implemented involves a user who is curious about the different information to be manipulated, retrieved and analyzed. It is by this reason that, in this section, we give a description of the underlying data structures of the entities identified.

Table 4.1.1 List of entities and their attributes

|  |  |  |
| --- | --- | --- |
| Entity Name | Description | Attribute(s) |
| User | The person who interacts with the system through the web interface | 1. User\_id (Primary key 2. Field\_id (foreign key) 3. FirstName 4. LastName 5. OtherName 6. Email 7. Role\_id (foreign key) |
| UserRole | Different users play different roles on the farm e.g engineer, manager,… | 1. Role\_id(primary key) 2. roleName |
| SensorPattern | This is based on the sensor data from the field | 1. s\_id (primary key) 2. date\_time 3. moist\_value |
| irrigationPattern | Data captured is based on the trigger of the irrigation action | 1. irp\_id (primary key) 2. s\_id (foreign\_key) 3. date 4. startTime 5. endTime 6. waterUsed |

Below is a description of the database structure which is composed of the different entities described in the above table.

SensorPattern

User

field\_id (pk)

field\_name

date

time

moist\_read

usr\_id (pk)

field\_id (fk)

fName

lName

email

role\_no (fk)

Date

0, \* View 1, 1

Role\_no (pk)

roleName

UserRole

IrrignPatterrn

irrp\_id (pk)

field\_id (fk)

timeStart

timeEnd

Duration

waterUsed

1,1 Trigger

1, \*

Has

0,1 0,1

**Figure 4.1.1 ERD**

## Data dictionary

The section describes the different entities, attributes and their variable data each is expected to store in the database.

Table 4.2.1 User-entity for creating an account

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Maximum size | Description |
| usr\_id | Integer | 10 | Unique identifier for each user |
| field\_id | Integer | 10 | Foreign Key referencing sensor pattern |
| role\_no | Integer | 10 | Foreign key referencing user role |
| fName | Varchar | 20 | User’s first Name |
| lName | Varch | 20 | User’s last Name |
| Email | Varchar | 50 | User’s email address |
| password | Varchar | 20 | User’s twisted string |

Table 4.2.2 User role

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Maximum size | Description |
| role\_no | Integer | 10 | Unique identifier for each role |
| roleName | Varchar | 50 | Role the user plays |

Table 4.2.3 SensorPattern

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Maximum size | Description |
| field\_id | Int | 10 | Unique identifier for each sensor data read by the field sensor |
| fieldName | Varchar | 40 |  |
| Date | dateTime | longDate | Date for which the data was read |
| Time | dateTime | shortTime | Time for which the reading was taken by the sensor |
| Moist\_read | Double |  | Moisture value read by the field detector |

Table 4.2.4 Irrigation pattern

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data type | Maximum Size | Description |
| Irp\_id | Integer | 5 | The unique identifier for irrigation data |
| field\_id | Integer | 10 | Foreign Key |
| Date | dateTime | Short date | Date for which irrigation has been triggered |
| timeStart | dateTime | Short Time | The time for which the irrigation started |
| timeEnd | dateTime | Short Time | Irrigation stopping time |
| waterUsed | Double |  | The quantity of water used during the whole process |
| Duration | dateTime | Short Time | The amount of time taken for the irrigation process |

# Component Design

The section provides a detailed description of the different components that constitute the smart irrigation and control system prior to the architecture. The flow of data and control as per each component has been described using the pseudo code.

## Data Input component

Two different sensors constitute the data input component, that is, soil moisture sensor and water flow sensor.

Pseudo code

*While the moisture sensor is inserted in the soil*

*Read Y (moisture content of the crop field.)*

*If Y < V (The thresh hold value)*

*Notify the user*

*Else*

*Proceed with detection*

*While irrigation is going on*

*Read X (amount of water flowing)*

*Record X*

## Data transfer component

*While X and Y are recorded*

*Transfer readings from the Arduino to Db*

*Record the dates and time*

## Output component

*If the toggle button changed to On*

*While Y < V*

*Irrigate the crop field*

*Else*

*Stop irrigation*

## Irrigation component

*If the on button is pressed*

*Send a command to the Arduino*

*Start irrigation*

*Else*

*Do nothing*

## Analysis component:

*While the on button is pressed and irrigation is going on,*

*Capture the soil moisture readings and the amount of water flowing through the pipe from the moisture sensor and water flow sensor respectively*

*Record the soil moisture readings and the amount of water flowing through the pipe in the data store*

*Display the soil moisture readings and the amount of water flowing through the pipe on the web component*

*Record and display the number of times irrigation is done*

*Generate and display a graph of the number of irrigations against time on the web component.*

## Data storage

*While irrigation process is taking place*

*Record the Capture the soil moisture readings and the amount of water flowing through the pipe in the data store*

# Human Interface Design

## Over view of Human Interface

The system shall provide a web-based user interface to its users in the web app module. The farm manager will be responsible for adding new users of the system. Every user shall login before accessing the system functions.

The action performed by the user includes the following: viewing moisture level as indicated by “Moisture Level”, view previous irrigation record indicated by “View Data”, perform irrigation indicated by “Irrigate” and Analyze data as shown in the figures below in succession.

When logged into the system the default page displays the moisture level provided by the check boxes shown in figure 6.2. When the moisture level is very low the farmer can navigate to irrigate function and click the toggle button to start the irrigation as shown in figure 6.3.

The previous irrigation record can be viewed by navigating through the ‘view data’ page which display the table listing of the previous irrigation record.

Finally, the user can analyze the irrigation date by plotting graphs and charts as shown in the ‘analyze data’ page in figure 6.5 below.

## Screen Images

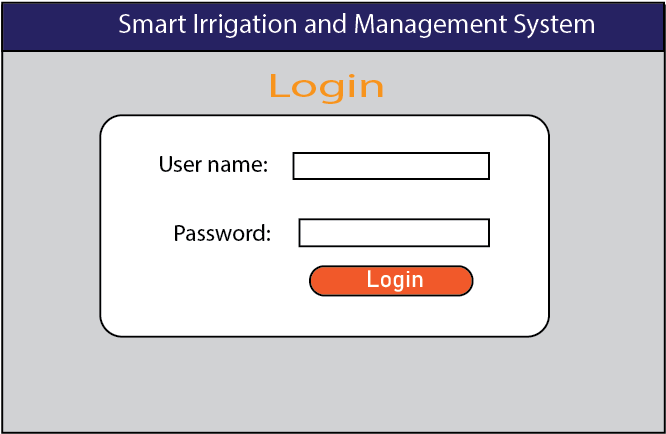


Figure 6.2.1 log in the system.

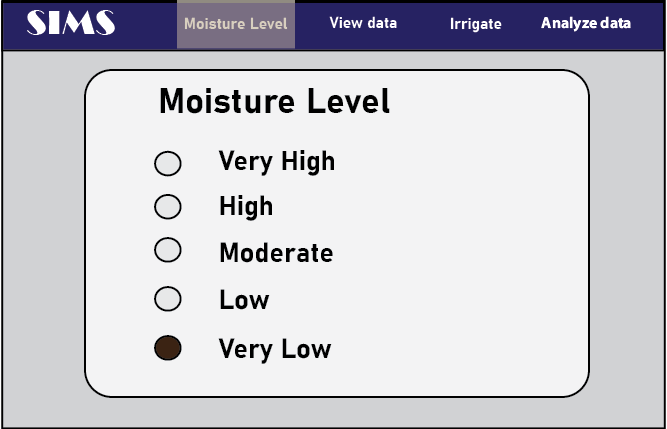


Figure 6.2.2 view the soil moisture level.

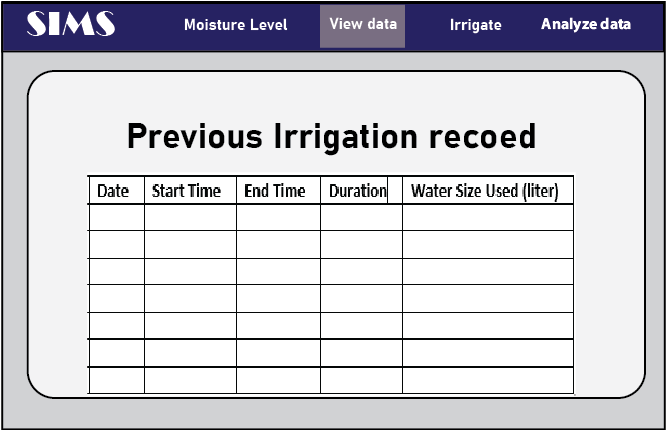


Figure 6.2.3 view previous irrigation record.

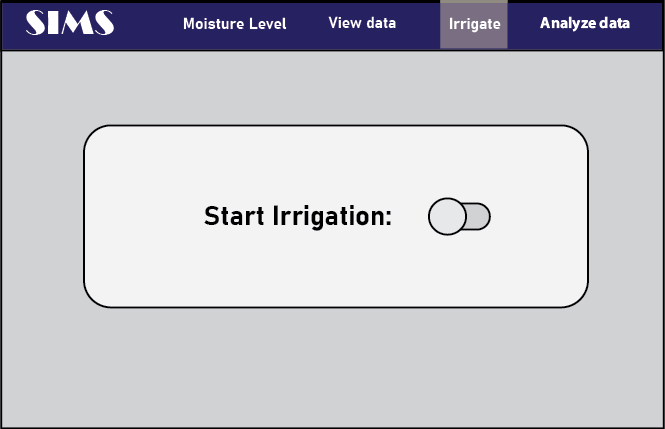


Figure 6.2.4 perform irrigation.

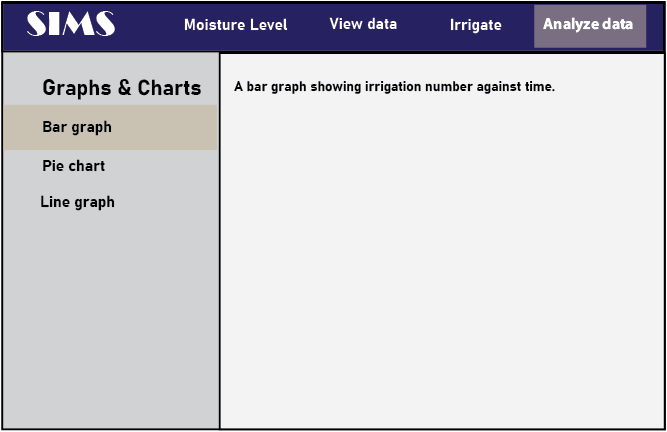


Figure 6.2.5 analyze data.

## Screen objects and actions

|  |  |  |
| --- | --- | --- |
| **Screen Objects** | **Action** | **What happens on performing the action** |
| Moisture Level | Moisture level of the soil in the crop field | Displays the moisture level reading of the soil. |
| View data | Previous records of the irrigation. | Display record of the previous irrigation performed. |
| Irrigate | Perform irrigation | Provides a toggle button to start and stop irrigation. |
| Start irrigation | Toggling button | Starts irrigation when its on and stops irrigation when its off. |
| Analyze data | Analysis on the previous irrigation record. | Display several analysis options on the irrigation record. |
| Bar graph | Analysis of the irrigation data on a bar graph | Display a par graph showing number of irrigations against time. |
| Pie chart | Analysis of the irrigation data on a pie chart | Display a pie chart showing water usage per irrigation |
| Line graph | Analysis of the irrigation data on a line graph | Display a line graph showing annual irrigation pattern. |