# **ABSTRACT**

An Agribalance Enterprise system is the system developed as a way to manage and store crops details and transaction from village level up to nation level. The system developed aims at addressing the problem by creating a safe database storage for storing all crops details. The main objective was to develop an online Agribalance Enterprise system.

To fulfill this objective of developing an Agribalance Enterprise system, the work was broken down into four measurable(aggregate) steps using waterfall methodology. The first step was requirement gathering (data collection) which was done using questionnaires and interviews. The second step was analysis of the data that was collected. Next to that was design of the system was done using Object Oriented Analysis and Design Methodology(OOADM) specifically the UML (Unified Modelling Language). Lastly, was the implementation of the system where the system was developed and tested against its functional requirements.

The output was the Agribalance Enterprise system.

# **ACKNOWLEDGEMENT**

First, we would like to give thanks to the Almighty God for giving us strength during all this time when conducting our Project, Thanks to all of the people who helped in designing, carrying out, and reviewing our project.

**TABLE OF CONTENT**

[**DECLARATION** i](#_Toc138319014)

[**ABSTRACT** ii](#_Toc138319015)

[**ACKNOWLEDGEMENT** iii](#_Toc138319016)

[**LIST OF FIGURES** vi](#_Toc138319017)

[**LIST OF TABLES** vii](#_Toc138319018)

[**LIST OF ABBREVIATIONS** viii](#_Toc138319019)

[**CHAPTER ONE** 1](#_Toc138319020)

[**1.INTRODUCTION** 1](#_Toc138319021)

[**1.1 GENERAL INTRODUCTION** 1](#_Toc138319022)

[**1.2 STATEMENT OF PROBLEM** 3](#_Toc138319023)

[**1.3 OBJECTIVES** 4](#_Toc138319024)

[**1.3.1 GENERAL OBJECTIVES** 4](#_Toc138319025)

[**1.3.2 SPECIFIC OBJECTIVES** 4](#_Toc138319026)

[**1.4. SIGNIFICANCE OF THE STUDY** 4](#_Toc138319027)

[**CHAPTER TWO** 5](#_Toc138319028)

[**2. LITERATURE REVIEW** 5](#_Toc138319029)

[**2.1 INTRODUCTION** 5](#_Toc138319030)

[**2.2.1 LITERATURES FOR DATA COLLECTION** 5](#_Toc138319031)

[**CHAPTER THREE** 7](#_Toc138319032)

[**3. METHODOLOGY** 7](#_Toc138319033)

[**3.1 INTRODUCTION** 7](#_Toc138319034)

[**3.2 DATA COLLECTION METHODS** 7](#_Toc138319035)

[**3.2.1 LITERATURE REVIEW.** 7](#_Toc138319036)

[**3.2.2 INTERVIEWS** 7](#_Toc138319037)

[**3.2.3 OBSERVATION** 7](#_Toc138319038)

[**3.2.4 QUESTIONNAIRES** 8](#_Toc138319039)

[**3.3 TOOS USED** 8](#_Toc138319040)

[**CHAPTER FOUR** 9](#_Toc138319041)

[**4. SYSTEM DESIGN** 9](#_Toc138319042)

[**4.1 INTRODUCTION** 9](#_Toc138319043)

[**4.2 REQUIREMENT ANALYSIS** 9](#_Toc138319044)

[**4.2.1 USER REQUIREMENTS** 9](#_Toc138319045)

[**4.2.1.1 FUNCTIONAL REQUIREMENTS** 9](#_Toc138319046)

[**4.2.1.2 NON-FUNCTIONAL REQUIREMENTS** 10](#_Toc138319047)

[**4.2.2 SYSTEM REQUIREMENTS** 11](#_Toc138319048)

[**4.2.2.1 HARDWARE REQUIREMENTS TOOLS** 11](#_Toc138319050)

[**4.2.2.2 SOFTWARE REQUIREMENTS TOOLS.** 12](#_Toc138319051)

[**4.3. SYSTEM DESIGNGS** 13](#_Toc138319052)

[**4.3.1 SYSTEM ARCHITECTURE** 13](#_Toc138319053)

[**4.3.2 CLASS DIAGRAM** 13](#_Toc138319054)

[**4.3.3 USE CASE DIAGRAM** 15](#_Toc138319055)

[**CHAPTER FIVE** 16](#_Toc138319056)

[**5. IMPLEMENTATION AND TESTING** 16](#_Toc138319057)

[**5.1 INTRODUCTION** 16](#_Toc138319058)

[**5.2 IMPLEMENTATION** 16](#_Toc138319059)

[**5.2.1 HOMEPAGE** 16](#_Toc138319060)

[**5.2.2** ADMINISTRATOR LOGIN 17](#_Toc138319061)

[**5.2.3 VILLAGE/WARD LEVEL LOGIN** 18](#_Toc138319062)

[**5.2.4 DISTRICT LEVEL LOGIN** 19](#_Toc138319063)

[**5.2.5 REGIONAL LEVEL LOGIN** 21](#_Toc138319064)

[**5.3 TESTING** 22](#_Toc138319065)

[**5.3.1 PERFORMANCE TESTING** 22](#_Toc138319066)

[**5.3.2 USER ACCEPTANCE TESTING** 22](#_Toc138319067)

[**5.3.3 FUNCTIONAL TESTING** 23](#_Toc138319068)

[**CHAPTER SIX** 25](#_Toc138319069)

[**6. SUMMARY, CONCLUSION AND RECOMMENDATION** 25](#_Toc138319070)

[**6.1 INTRODUCTION** 25](#_Toc138319071)

[**6.2 CHAPTER SUMMARIES** 25](#_Toc138319072)

[**6.2.1 INTRODUCTION** 25](#_Toc138319073)

[**6.2.2 LITERATURE REVIEW** 25](#_Toc138319074)

[**6.2.3 METHODOLOGY** 25](#_Toc138319075)

[**6.2.4 SYSTEM DESIGNING** 26](#_Toc138319076)

[**6.3 CONCLUSION** 26](#_Toc138319077)

[**6.4 CHALLENGES AND LIMITATIONS** 26](#_Toc138319078)

[**6.5 RECOMMENDATIONS** 27](#_Toc138319079)

[**REFERENCE** 28](#_Toc138319080)

# **LIST OF FIGURES**

[Figure 1 Class diagram 14](file:///E:\ISM%20II%20GROUP%20PROJECT\ONLINE%20ACADEMIC%20ADVISORY%20SYSTEM.docx#_Toc138237758)

[Figure 2 Use case Diagram 15](#_Toc138237759)

[Figure 3 Homepage 16](file:///E:\ISM%20II%20GROUP%20PROJECT\ONLINE%20ACADEMIC%20ADVISORY%20SYSTEM.docx#_Toc138237760)

[Figure 4 Users Login page 17](file:///E:\ISM%20II%20GROUP%20PROJECT\ONLINE%20ACADEMIC%20ADVISORY%20SYSTEM.docx#_Toc138237761)

# **LIST OF TABLES**

[Table 1 Tools and methodologies 8](#_Toc138237784)

[Table 2 Functional Testing 23](#_Toc138237785)

# **LIST OF ABBREVIATIONS**

1. HTML - Hyper Text Markup Language
2. PHP - Hypertext Preprocessor
3. CSS - Cascade Style Sheet
4. MySQL - My Structured Query Language
5. UML - Unified Modeling Language
6. RAM - Random Access Memory

# **CHAPTER ONE**

# **1.INTRODUCTION**

## 1.1 GENERAL INTRODUCTION

National as national consist of four main category village, district, regional and national level where differs cooperatives groups they are presents and they practice local crop management where data stores on manual papers that is too led way can cause easily way to loss of those data. This project aims to design and implement an online Agribalance Enterprise system to effectively assist those cooperatives to store their data in easy way, longtime and manage all crops at all levels.

The primary goal of the Agribalance Enterprise system is to ensure that all crops data are being managed and stores in safe way through system.

## 

## 1.2 STATEMENT OF THE PROBLEM

At all national level, most of cooperatives and all agriculture companies crops data are being store manual on paper based that can easily to be destroyed or loss out. Consequently, to address this issue, there is a need for an Agribalance Enterprise system specifically designed to cater to the needs of agriculture companies who face challenges in storing and manage their crops. This system should provide a supportive and comfortable environment for cooperatives and agriculture companies to store and manage and store all crops at all level through an online system.

## 1.3 OBJECTIVES

### **1.3.1 GENERAL OBJECTIVES**

The general objective of this project is to develop an Agribalance Enterprise system.

### **1.3.2 SPECIFIC OBJECTIVES**

1. To gather requirements from farmers and differs agriculture company.
2. To design an Agribalance Enterprise system.
3. To implement an Agribalance Enterprise system.
4. To deploy and test the System

## 1.4. SIGNIFICANCE OF THE STUDY

The implementation of Agribalance Enterprise system is agriculture company and cooperative facing challenges and difficulties in storing and manage crops data at all national level. By examining how Agribalance Enterprise system can address these challenges, the system contributes to creating a supportive environment that can store and manage crops through system.

# **CHAPTER TWO**

# **2. LITERATURE REVIEW**

## **2.1 INTRODUCTION**

A literature review is a goal, critical summary of published research literature relevant to a topic under consideration for the research. Its purposes are to create familiarity with current thinking and research on a particular topic and may justify a future in a previously overlooked or understudied area. In this chapter which is literature review outlines the current state of the existing Agribalance Enterprise system that exists in use. It shows the way these systems manage information and data relating to its users. The literature then justifies the importance of adopting a new advanced system and how the system will be beneficial to the users and it will justify if the information system is needed**.** The conclusions of this review will identify the required system, identifying the purpose of the new system being developed.

### **2.2.1 LITERATURES FOR DATA COLLECTION**

According to Ward (1979) He stated that one of the main responsibilities of any institution of higher education involves academic advisement of students. The university adviser is the student’s principal faculty adviser. His prime concern with the student, and the student’s with him is the identification of the student’s aims and plans. His interest and abilities and the planning of coherent education that builds upon the student’s interest and awareness of both his strength and weaknesses. The adviser does not plan for the student but helps the student plan for himself.

According to Omiros Latrellis (2017) explains that in order to support the educational process, a software system is needed that will handle the advisory process in an efficient and effective way. An academic advisory system can serve as a strategic partner responsible for the process of supporting, motivating student’s study plan and assisting them in the achievement of their educational goals. He also recommends the importance of making academic advisory systems that are more than data repositories and including more intelligence to the systems which are able to provide reliable advice when needed.

According to Smith Johnson (2022) the role of academic advisory systems in supporting students in higher education, particularly those who face challenges and have difficulty expressing themselves physically to their lecturers. Thus the academic advisory systems are of great significance in facilitating student success, addressing individual needs and providing a supportive environment for students to seek guidance.

# **CHAPTER THREE**

# **3. METHODOLOGY**

## 3.1 INTRODUCTION

A methodology is a standard process followed by an organization to conduct all step necessary to analyze, design, implement and maintain information systems. It describes the activities involved in defining, building and implementing the system. One of the methodology used in accomplishing an Agribalance Enterprise system is literature review.

## 3.2 DATA COLLECTION METHODS

These are methods used in the process of gathering and analyzing accurate data from various sources to find answers to research problems, trends and probabilities. In this case data collection methods such as interview and a questionnaire was developed in which the concept of the information with relationship between peasants, and agriculture cooperatives would be evaluated from

**3.2.1 LITERATURE REVIEW**

This is an overview of the existing produced system based on online document storing. The data for this system as extracted from online source done by other people concerning tucas

### **3.2.2 INTERVIEWS**

The peasants and agriculture cooperatives group will be interviewed, questions addressed to them will ask for their view on how the current system works and whether the introduction of the proposed system will be of value or will just a mere impact to them.

### **3.2.3 OBSERVATION**

Allows the study of the dynamics of a situation, Systematic observation, recording, description, analysis and interpretation of people’s behavior. It is concerned with collecting primary data through observing and noting.

## 3.3 TOOLS USED

Table 1 Tools and methodologies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S/N | Specific objectives | Methodology | Tools | Deliverables |
| 1 | Gathering system and user requirement | Interview  Observation | Face to face visits  Phone recording | User and system requirements |
| 2 | Analysis and design | Use Case diagram and Class diagram | Draw io | Use Case and class diagram system design |
| 3 | System implementation | Scrum methodology | HTML, PHP, CSS, JAVASCRIPT, MySQL | Agribalance Enterprise system |
| 4 | System testing | Beta test | Laptop | System evaluation and result |

# CHAPTER FOUR

# 4. SYSTEM DESIGN

## 4.1 INTRODUCTION

System design is a process of collecting, interpreting, identifying the problems, interpreting facts, and decomposition of a system into its components. This chapter holds the requirement for the system depending on the user’s needs as well as architecture of the system.

## 4.2 REQUIREMENT ANALYSIS

Requirement analysis is the process of determining user expectations for a new or modified product. It is also known as Requirement engineering. It involves analyzing documents, validating documents, and manages all of the identified requirements while considering the possibility that there are conflicting requirements among stakeholders (Peter Landau. 2019). The user requirement is categorized into two parts which are user requirement and system requirement.

### **4.2.1 USER REQUIREMENTS**

User requirement is a type of functional requirement that defines the end-user requirements for a system. It is an integral part of information systems design. Also describes the business needs for what users require from the system. User Requirements Specifications are written early in the validation process, typically before the system is created (Mark Kraeling. 2019). There are two types of user requirement which are functional and non-functional requirements.

### **4.2.1.1 FUNCTIONAL REQUIREMENTS**

Refers to description of the service that the software must offer. It explains a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs.

The follows below are functional requirement of the Online Academic Advisory System.

1. The system shall allow company admin to log in using username and password.
2. The system shall allow company admin to register village, ward, district and national users.
3. The system shall allow registered users to insert and manage all crops data.
4. The system shall allow users to change their passwords.

### **4.2.1.2 NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirement refers to attributes such as performance, security, reliability, maintainability and usability. They serve as constraints or restrictions on the design of the system across the different backlogs (Addison Wesley. 2010). Functional requirements define the system’s fundamental behavior, non-functional requirements set out how the system will carry out this function

The following are non-functional requirements for the Agribalance Enterprise system

1. **User interface**

For better user interaction, the interactive inputs screen is well designed. They would make data entry simple and easy for the user and crops data will be accepted in the same manner as it is done manually but in a much attractive way.

1. **Updating and Deletion of records**

This system will provide features to update records in the system, in updating user and crops data could change any field of the record except, the key field of the existing records.

The system also provides deletion of data. The user can delete any of the record if he/she is authorized.

1. **Data validation**

Various checks should be provided in the database for the data entry updating to ensure data security and validity. Data validation checks would be provided to ensure that there are no duplicates.

1. **Simple to use**

The System requires no special training for its operation, because it is user friendly and communicates with users in simple English language. Thus any person who has some knowledge of computers and the English language can use it.

1. **Data handling**

The present manual system requires stationary for the entry of data. There is therefore a greater chance of mistakes leading to errors, which result in system inefficiency and in adequacy. In the proposed system data validity checks have been incorporated to minimize the chances of entering incorrect data.

1. **Backup Functionality**

A backup data is provided which helps in making the duplicate copy of the data weekly. In case of any damage to data by virus (these are malicious programs which attach itself to other programs) or power failure the whole record can be fed into the computer again

### **4.2.2 SYSTEM REQUIREMENTS**

## **Tools used****, System requirements tools**

System requirement tools are the tools that performs the functionality that is needed by a system in order to satisfy the customer's requirements. Also refers to what things required for the system to be accomplish or for built (Addison Wesley. 2012). There are three types of system requirements which include hardware, software and data requirement.

### **4.2.2.1 HARDWARE REQUIREMENTS TOOLS**

Hardware requirements tools these are physical and tangible requirements of a hardware device that are used during built system (Addison Wesley. 2012). The following are hardware requirement of the district contact archive system: -

1. Processor (Central Processing Unit).

Processor is the logic circuitry that responds to and processes the basic instructions that drive a computer. Processor was used to responds instruction that drive a computer during system design and coding such as running process of the system and execution of the system.

1. Random Access Memory (RAM).

Random access memory refers to a temporary memory bank where your computer stores data it's currently working on. RAM was used for temporal storage of our data such as codes for our system, because processor does not have to go into long-term storage to complete immediate processing tasks.

1. Hard disk.

Hard disk is used to store operating systems, software programs and other files using magnetic disks. More specifically, hard disk drives control the reading and writing of the hard disk that provides data storage. HDDs was used as primary or secondary storage device during system design and implementation and design

1. Compatible mouse

Mouse is a handheld hardware input device that controls a cursor in a GUI (graphical user interface) for pointing, moving and selecting text, icons, files, and folders on your computer. On our system mouse was used to draw graphical user interface such us first login page, also was used to pointing and moving files such as database file and codes files.

### **4.2.2.2 SOFTWARE REQUIREMENTS TOOLS.**

Software requirement tools these are also known as Software Requirements Tools Specification or is a document or set of documentation that describes the features and behavior of a system or software application. The following are software requirement tools.

1. PHP language

PHP refers to hyper pre-processor this is the back-end language which is used for design database. To our system PHP was used to connect user interface and the data base of the system.

1. My Structure query language (MySQL)

This is the language that used to create database in our system.

1. Operating System.

Operating system Refers to a piece of software that manages the allocation of computer hardware. On the system was used forexecute our system DCAS in a convenient and efficient manner.

1. Windows

Windows is a graphical operating system developed and published by Microsoft.it was used in the computer we used to create our system our system as well as to display the system.

## 4.3. SYSTEM DESIGNGS

### **4.3.1 SYSTEM ARCHITECTURE**

System architecture refers to the overall design and structure of a complex system, encompassing its components, relationships, and principles. It involves determining how different parts of a system interact, communicate, and function together to achieve specific objectives. Systems architecture takes into account various factors such as performance, scalability, reliability, security, and flexibility to create an efficient and effective system design. It typically includes components like hardware, software, network infrastructure, data storage, and interfaces, ensuring they align with the system's goals

The purpose of systems architecture is to ensure that a system meets its intended objectives and requirements. It allows for a clear understanding of how various elements within the system interact, communicate, and work together. By defining the structure and interfaces, systems architecture facilitates efficient integration and interoperability of system components. It also aids in identifying potential risks, optimizing performance, and supporting system evolution and maintenance. (Bass, 2012)

### **4.3.2 CLASS DIAGRAM**

A class diagram is a type of static structural diagram in software engineering and UML (Unified Modeling Language) that depicts the classes, attributes, operations, and relationships within a system. It provides a visual representation of the classes and their associations, giving insights into the structure and behavior of the system.

In a class diagram, classes are represented as rectangles with three compartments: the top compartment contains the class name, the middle compartment lists the attributes (variables or data members) of the class, and the bottom compartment represents the operations (methods or functions) that the class can perform.

The relationships between classes are represented using various types of connectors, such as association, inheritance, aggregation, and composition.

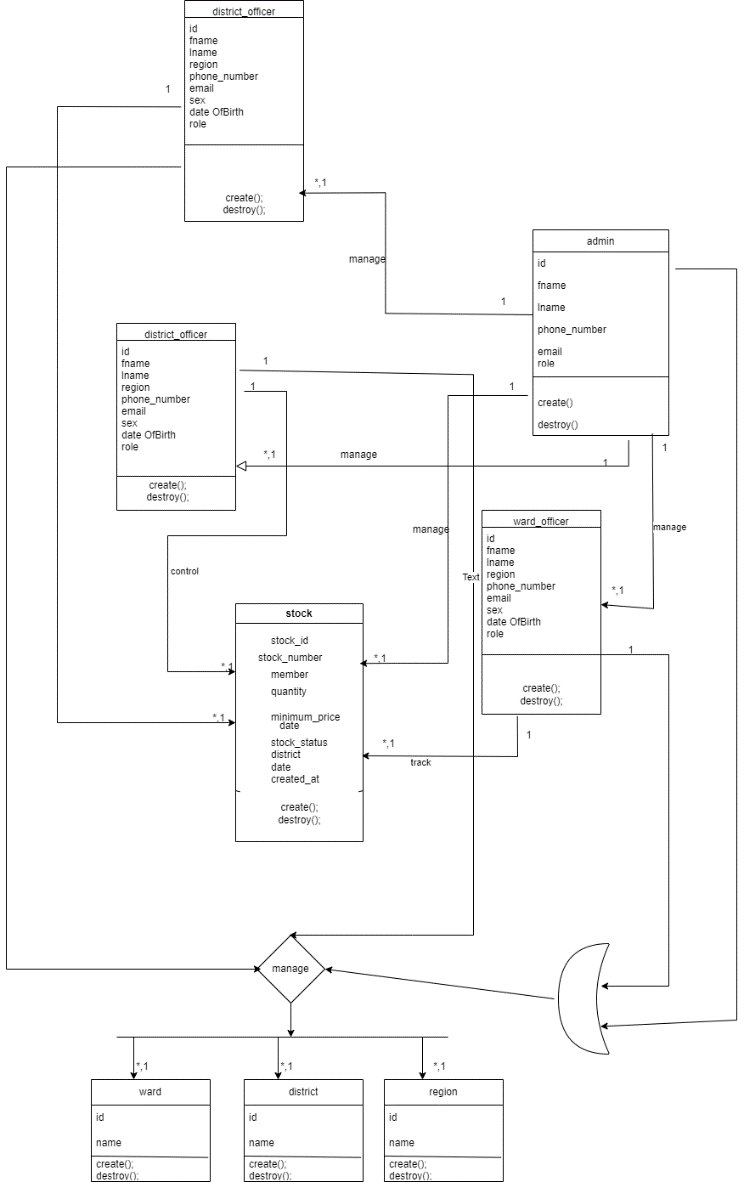


Figure Class diagram

CHAPTER FIVE

# 5. IMPLEMENTATION AND TESTING

## 5.1 INTRODUCTION

This chapter hold an overview of the implementation and testing of the system including part of the system and how they function, generally this chapter explain in details what the system is able to perform.

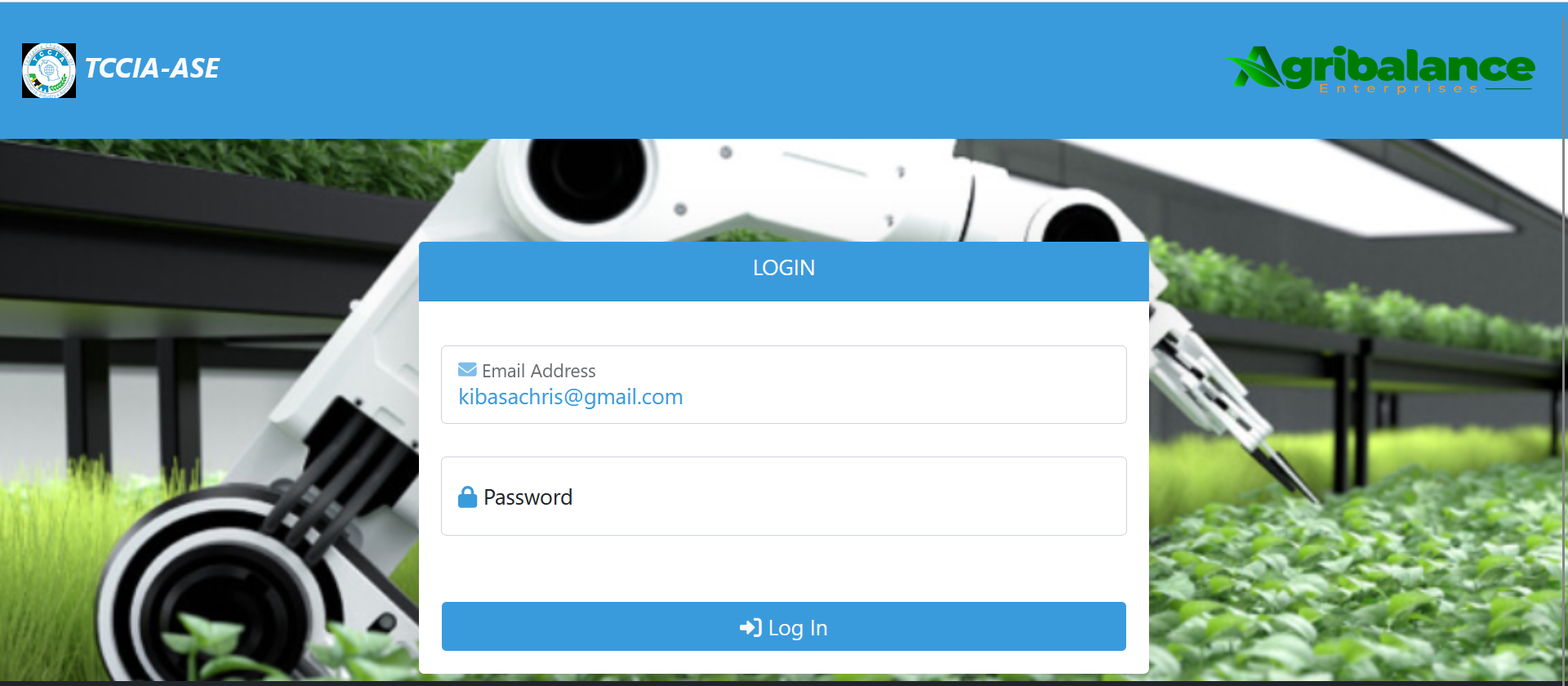
## 5.2 IMPLEMENTATION

Is the process of putting a planned system into action. It involves the actual development, installation and configuration of the system to make it operational and functional within an organization.

### **5.2.1 HOMEPAGE**

Is the first part of the system with all features that allow users to login to the system.

Figure Homepage



### **5.2.2 ADMIN DASHBOARD**

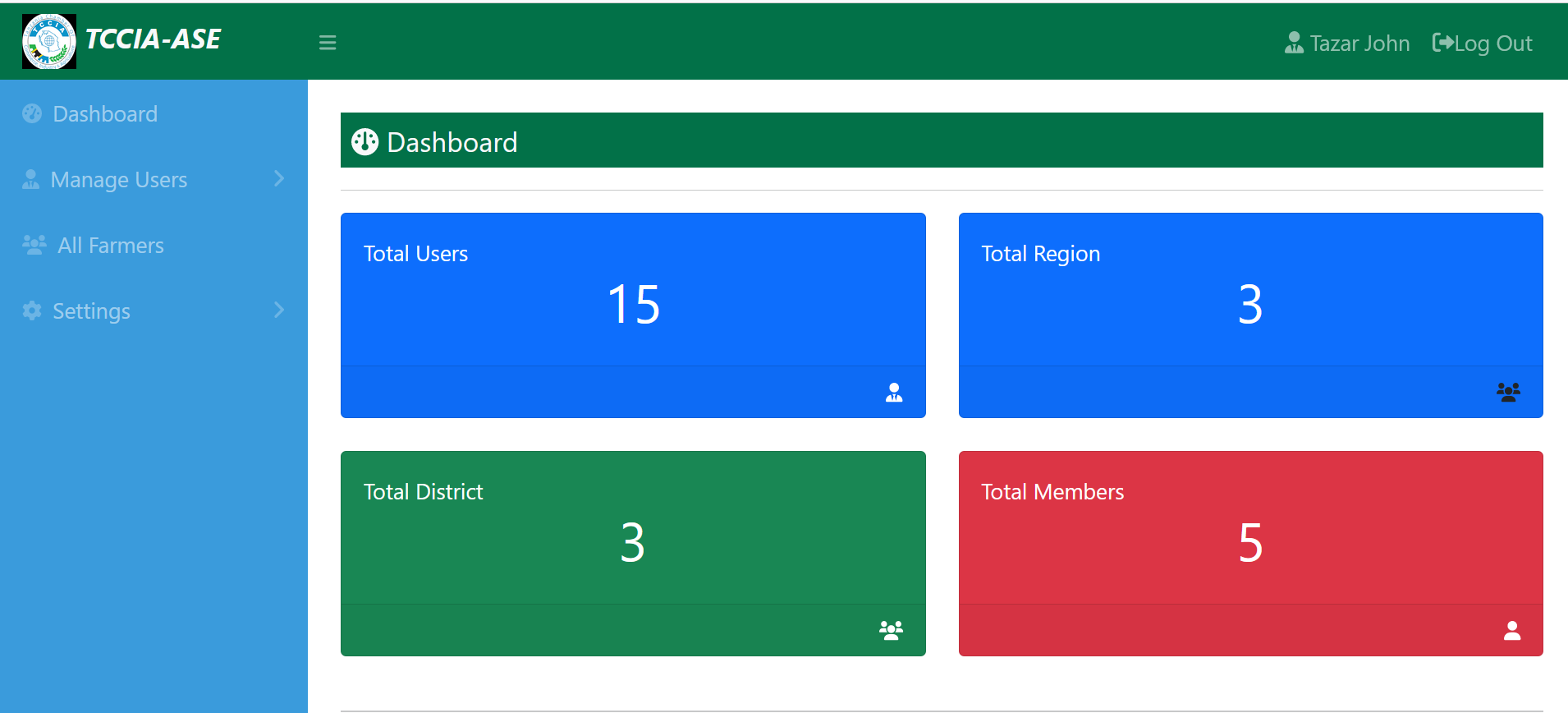
On this page allow only Admin to login to the system by filling their credentials (Email and Password) in the login form. If the credentials (email and password) match as assigned.

Figure Admin dashboard

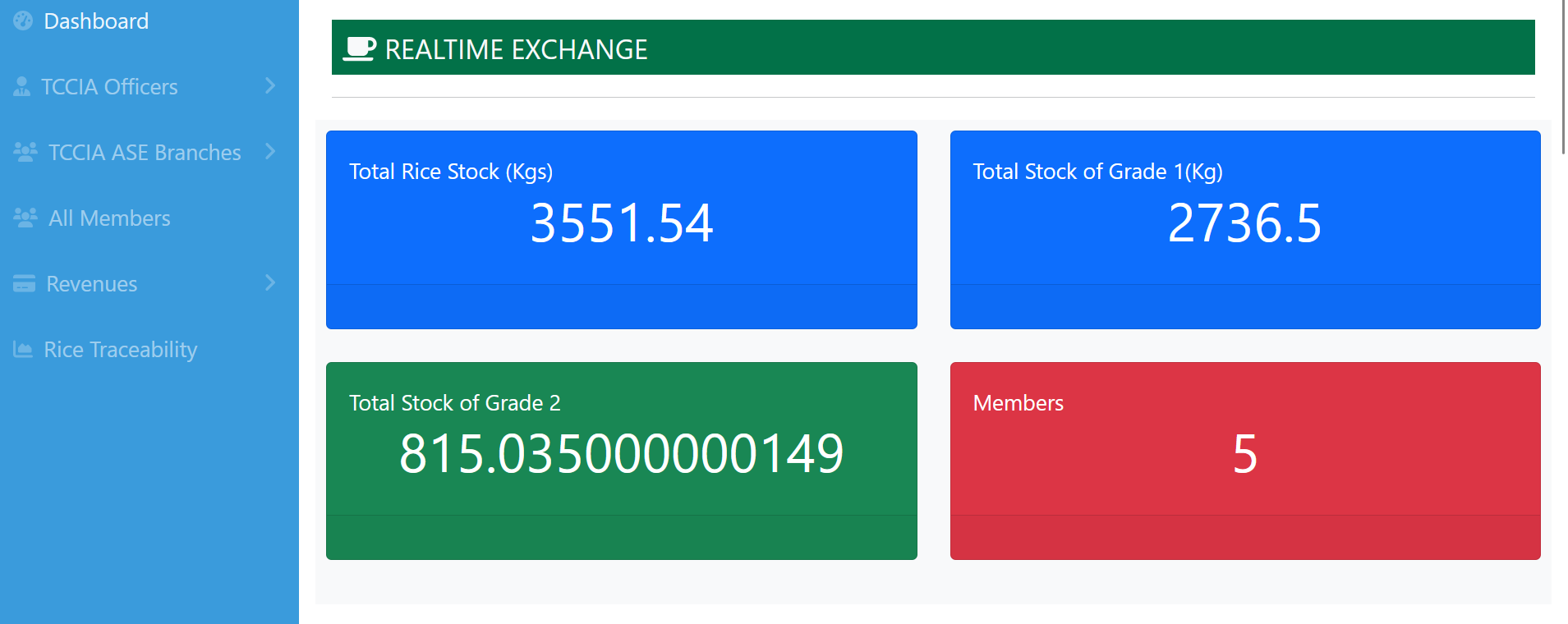
This is a page where Admin to view and track all real time exchange.

Figure Advisor Login Page

### **5.2.3 STUDENT LOGIN PAGE**

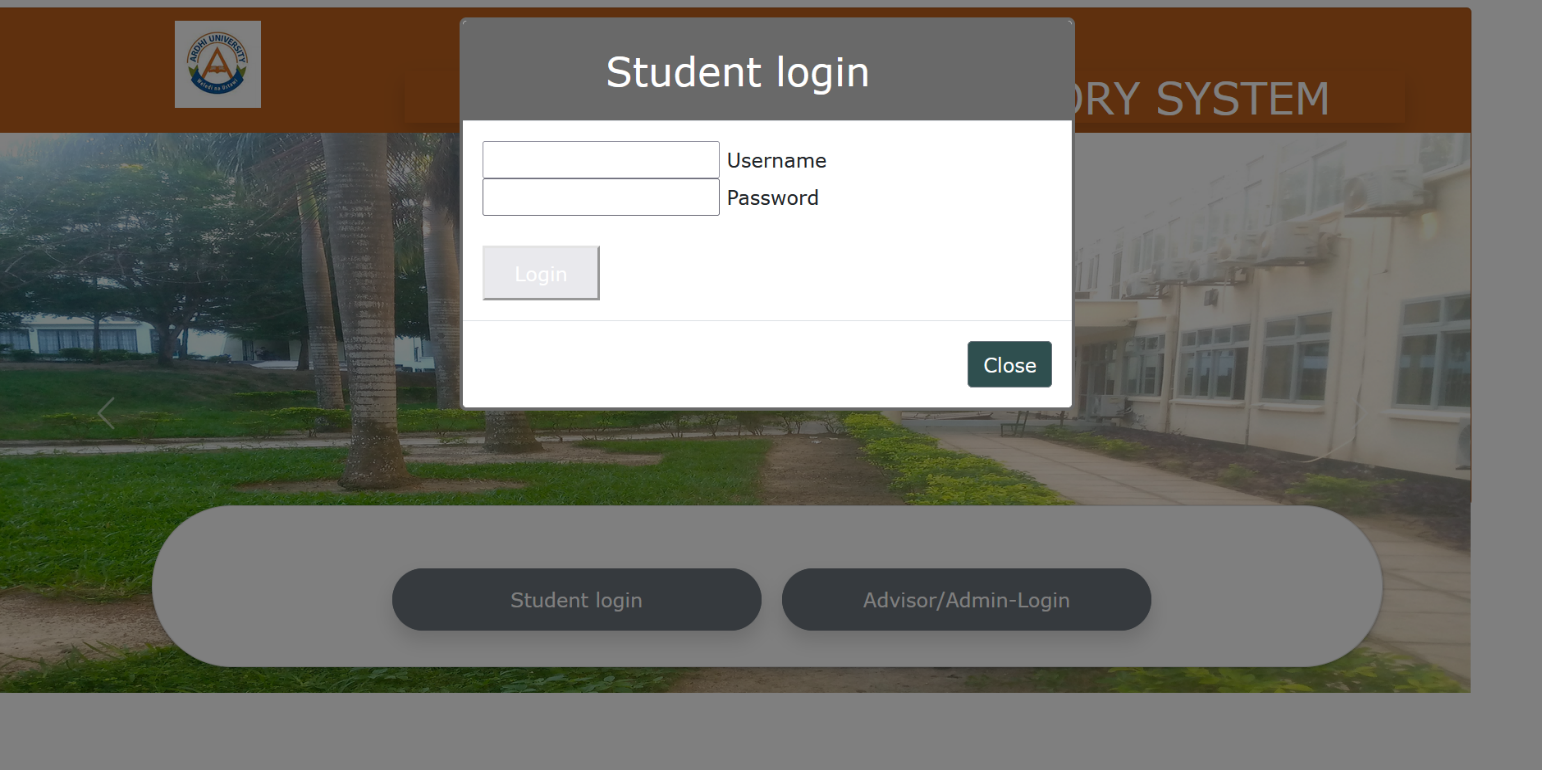
This is a page where Student with an account yet in the system can login by filling their email and password in the form, and access services based on their specific assigned group and Supervisor in the system.

Figure Student Login Page

### **5.2.4 DEPARTMENT COORDINATOR HOMEPAGE**

After the Department Coordinator logs in to the system, he or she will be directed to this page, where he or she will be able to assign Advisors to their specific group; and Students to their specific group and Advisors.

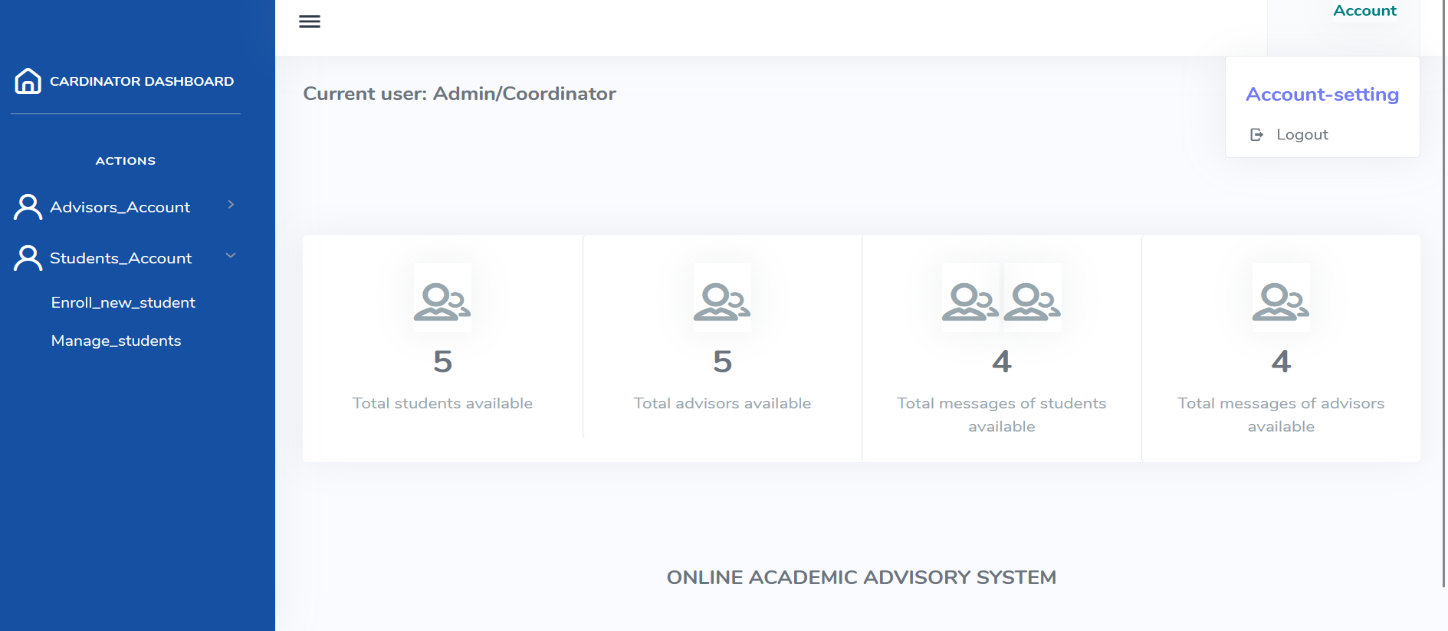
 Also to see and preview the total available Students and Advisors assigned in the system and Total available Messages.

Figure Department Coordinator Homepage

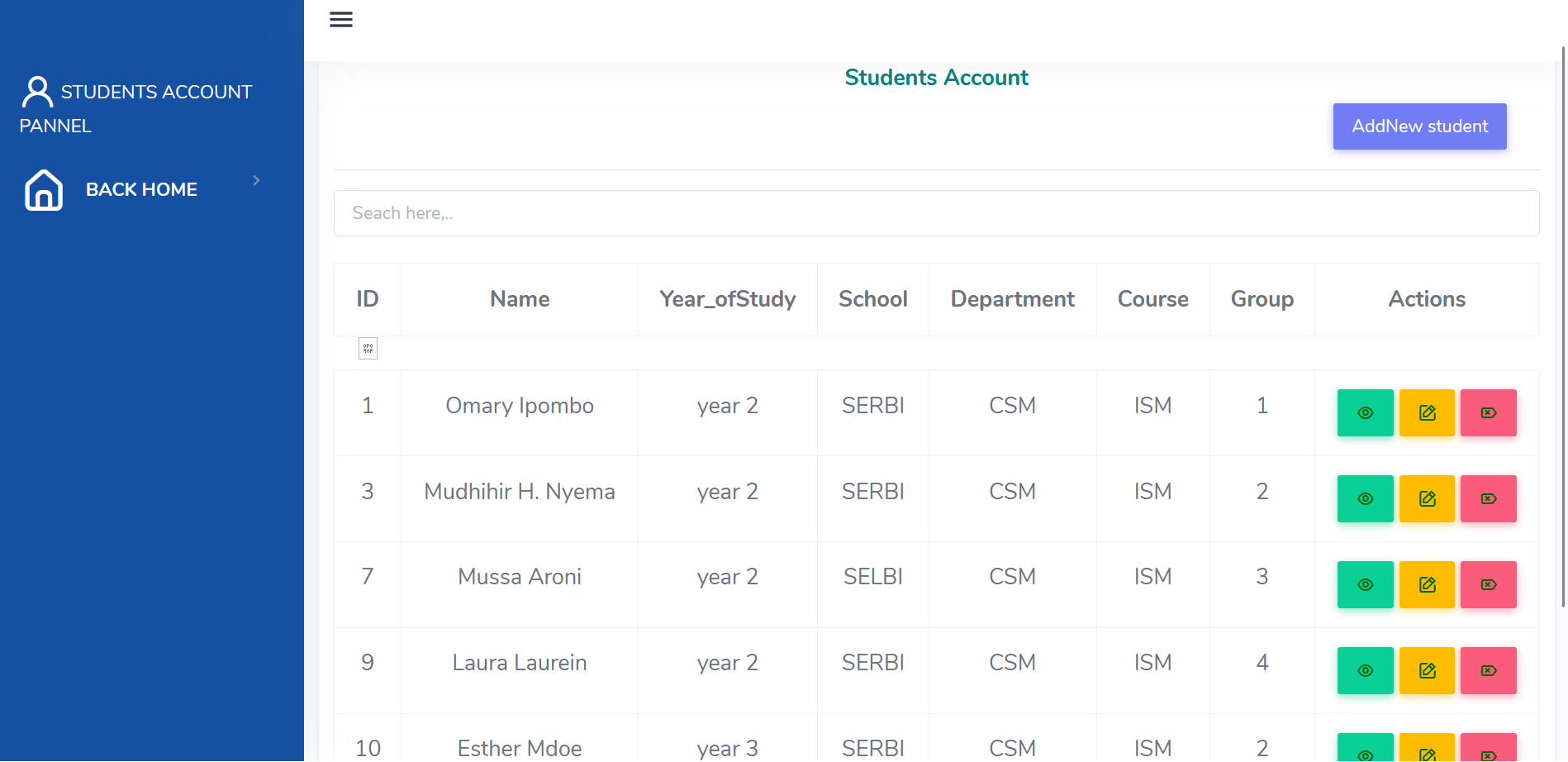
The department coordinator will also be able to add and manage (view, edit and delete) specific Advisor.

Figure Department Coordinator Homepage

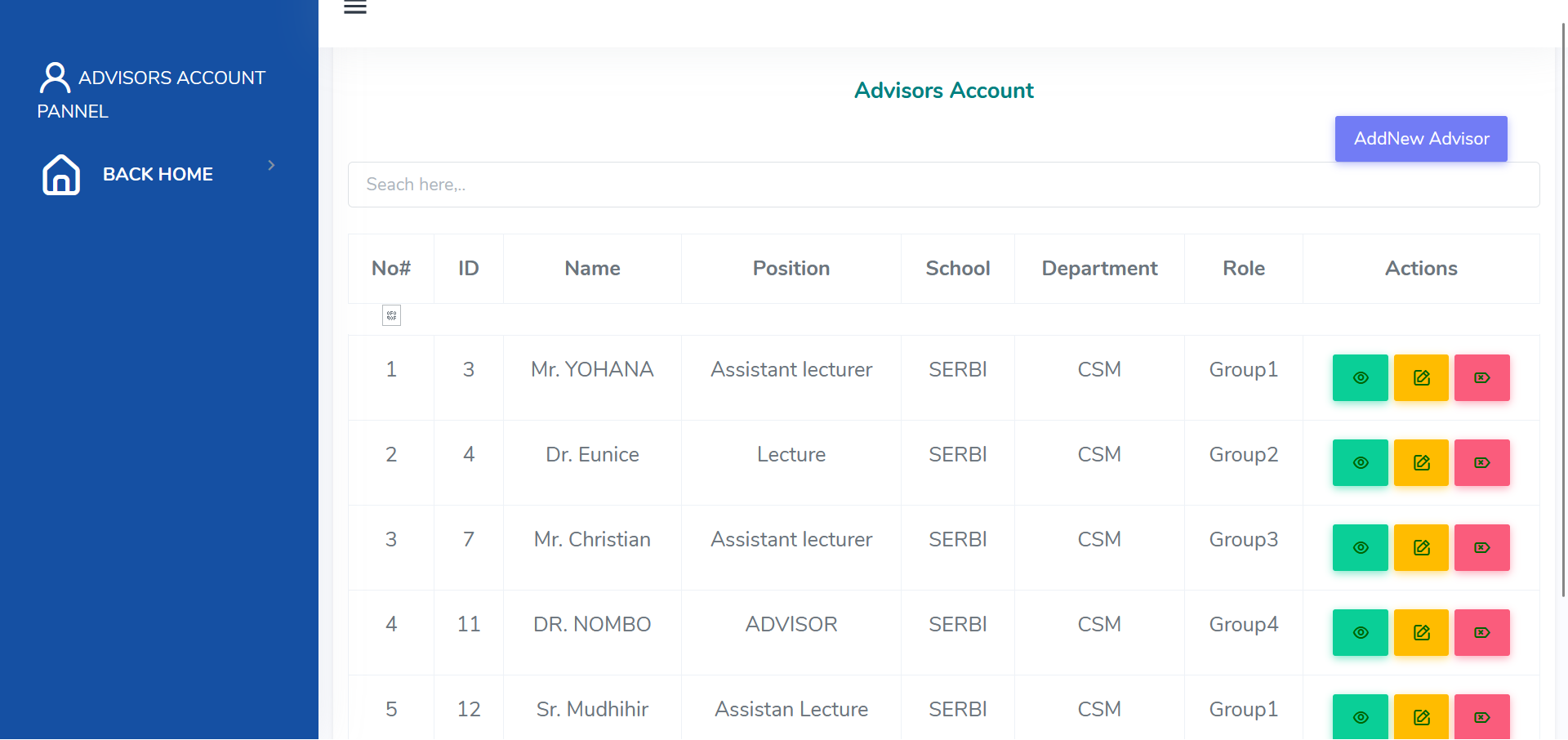
 The department coordinator will also be able to add and manage (view, edit and delete) specific Student.

Figure Department Coordinator Homepage

### **5.2.5 ADVISOR HOMEPAGE**

After Advisor logs in to the system, he or she will be directed to this page according to the assigned group, where Advisor will be able to see total number of students assigned to him/her by Coordinator.

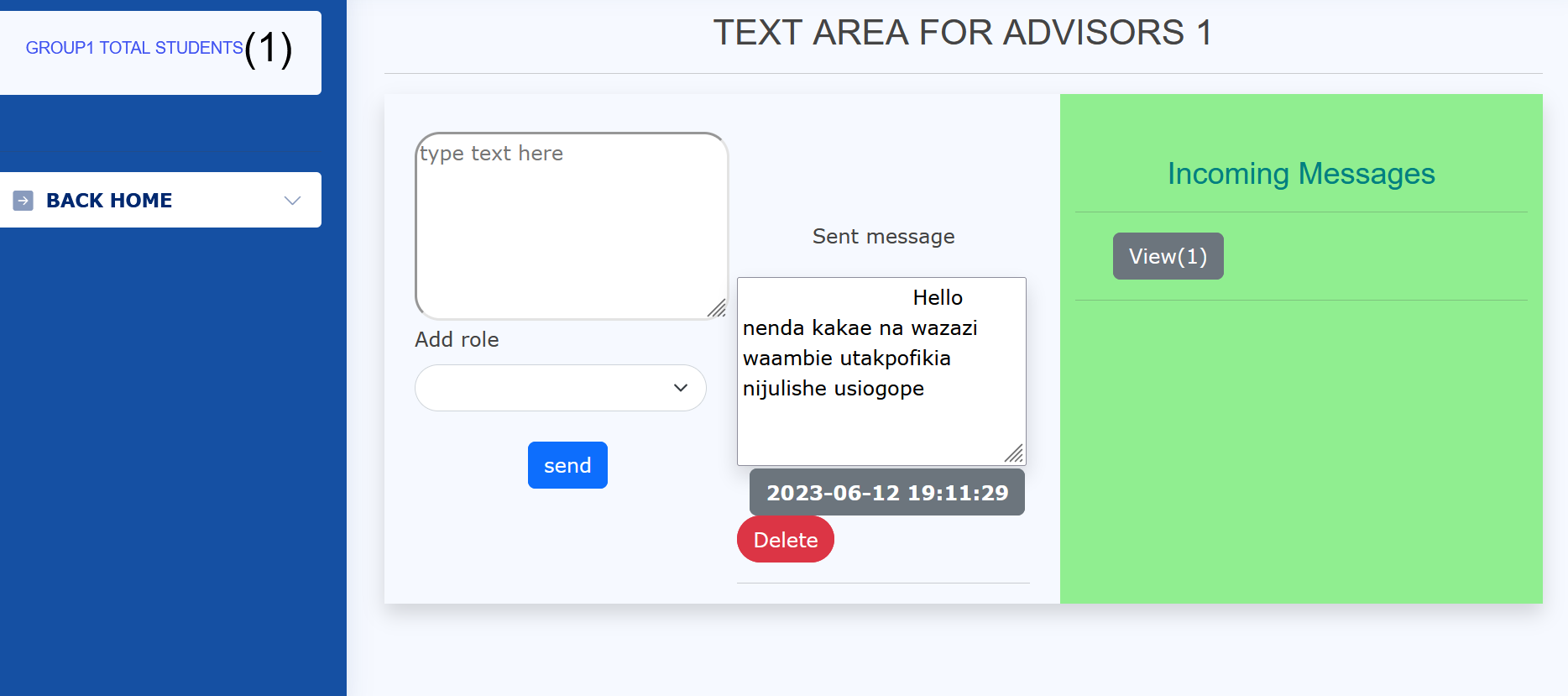
Also he or she will be able to communicate with assigned student by chatting.

Figure Advisor home page

**5.2.6 STUDENT HOMEPAGE**

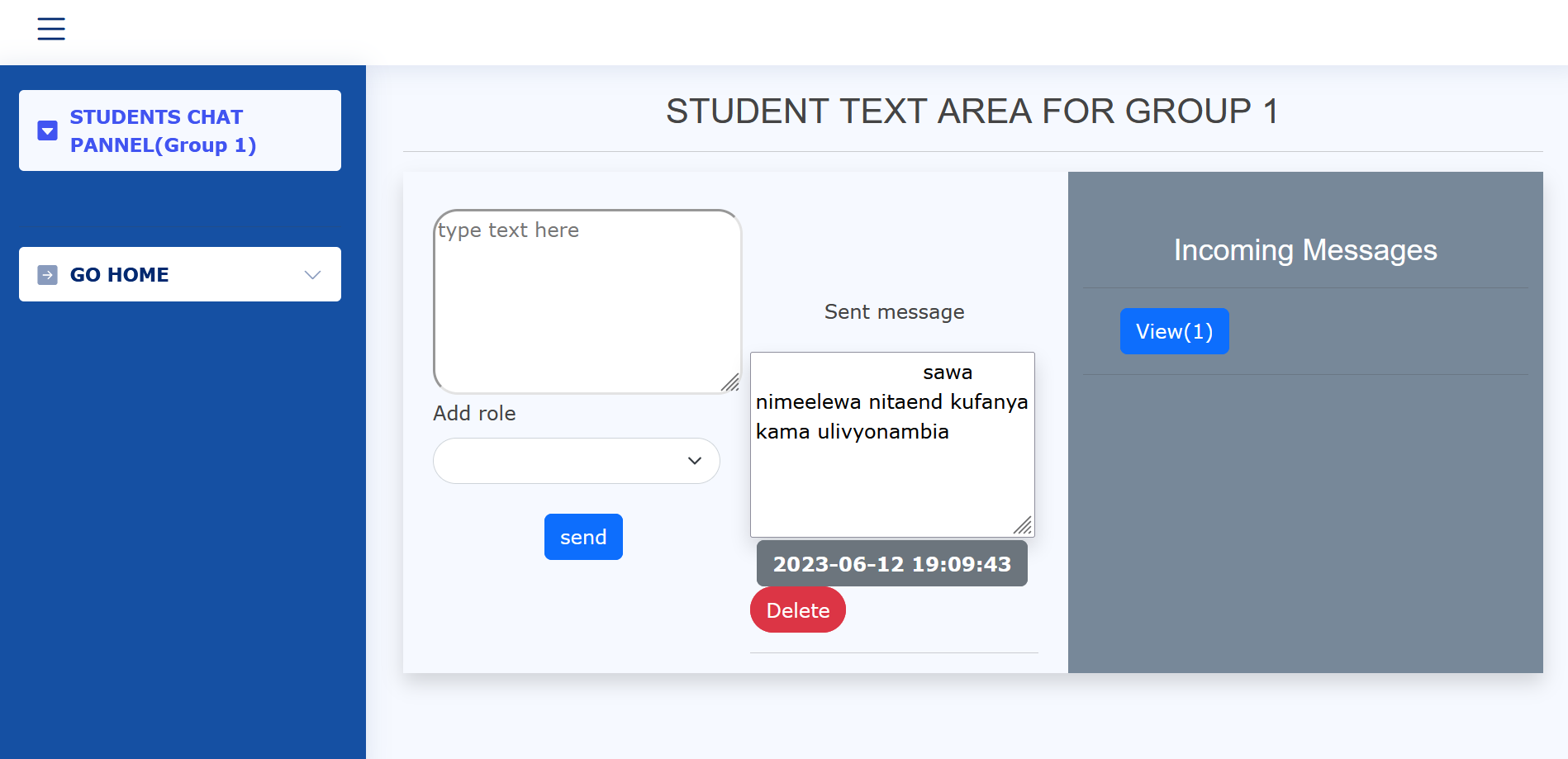
After Student logs in to the system, he or she will be directed to this page according to the assigned group, where he or she will be able to communicate with assigned Supervisor by chatting.

Figure Student home page

Figure Student home page

## 5.3 TESTING

Is the process in which a quality assurance team evaluates how the various components of an application interact together in the full, integrated system or application. The importance of testing, improves the system quality in terms of performance; errors reduction and improving system security since ensures the system does not contain potential vulnerabilities that can put end users and system data at risk of potential threats.

From our system tucas; we have used three Testing methods which are:

1. Performance Testing
2. User Acceptance Testing
3. Functional Testing

### **5.3.1 PERFORMANCE TESTING**

According to (International Software Testing Qualifications Board (ISTQB). (2018).ISTQB Glossary of Testing Terms); Performance Testing is a non-functional testing technique performed to determine the system parameter in terms of responsiveness and stability under various workload.

We have used the performance testing to measures the quality attribute of the system such as scalability, reliability and availability

1. Scalability; from the test the users (Student, Advisor and Coordinator) can easily Perform their activities without changing or destructing the system.
2. Usability; the system is very usable as it includes the use of simple language and good graphical interface that allow users to access the system easily.
3. Availability; the system can be accessed anytime depending on the user choice.

### **5.3.2 USER ACCEPTANCE TESTING**

According to (International Software Testing Qualifications Board (ISTQB). (2018).ISTQB Glossary of Testing Terms); User Acceptance Testing Is a type of testing performed by the end user or client to verify the system before moving the system to the production environment.

We have used Beta test as a method of providing results on the developed system of Online Academic Advisory System. The main users (Student, Advisor and Department Coordinator) have tested the system and self-satisfaction according to his or her category and corresponding requirements.

### **5.3.3 FUNCTIONAL TESTING**

According to (International Software Testing Qualifications Board (ISTQB). (2018).ISTQB Glossary of Testing Terms); Functional Testing is the method that involves testing the system’s functions and features against the documented functional requirements.

We have used the functional testing to verify whether the system behaves correctly and produces the expected outputs.

# CHAPTER SIX

# 6. SUMMARY, CONCLUSION AND RECOMMENDATION

## 6.1 INTRODUCTION

This chapter provide the whole summary of what has been covered in all chapters from the project, conclusion about all ideas discussed in the project report as well as the recommendation for the future improvements to the system that can be done to increase efficiency of the system.

## 6.2 CHAPTER SUMMARIES

### **6.2.1 INTRODUCTION**

Tucas is the system that helps in managing and store crops data at all national level. It provides a good environment for agriculture companies to store and manage their data through system.

### **6.2.2 LITERATURE REVIEW**

A literature review on the tucas helps us to understand existing approaches, identify gaps, and gather valuable insights to inform the design and implementation of the project. It serves as a foundation for acquiring knowledge and understanding the best practices in the field, ultimately contributing to the success and effectiveness of the project.

### **6.2.3 METHODOLOGY**

There are different methods we used in creating our system including the tools used like HTML, CSS, Java script as well as bootstrap for front end part of our system, PHP for back-end implementation of our system for making the interaction between database and front end of the system, MYSQL was also used to create database of our system, as we used SQL database to implement our system database.

### **6.2.4 SYSTEM DESIGNING**

There are various user requirements both functional requirement of our system like system should allow user to login, System should allow user registration, the system should allow students to

Initiate chatting with their academic advisors, the system should allow student enrolment to their respective academic advisors. Also, there are non-functional requirement of the system like usability as well as scalability.

**6.2.5 IMPLEMENTATION AND TESTING.**

The system consists of different parts including homepage of the system which is visible to all users when they interact with the system, it is the part where user can log in depending on their role this can be a administrator, an village manager, ward manager, or district manager.

## 6.3 CONCLUSION

Generally, the main objective of this study was successfully accomplished. An tucas was developed to enable agriculture to effectively and efficiently manage and store crops data.

## 6.4 CHALLENGES AND LIMITATIONS

Though the system is successfully designed there are some challenges and limitation that were faced during the implementation of the Online tucas. Some of the challenges include lack an adequate time for conducting research to get enough data as we expected, lack of capital for live testing our system as we expected to host our system for the time for an intention of testing to see how it will respond for multiple server requests.

## 6.5 RECOMMENDATIONS

An effective recommendation for an online tucas would be to implement a comprehensive and inclusive approach that caters to agriculture companies who face various challenges. One crucial aspect would be to establish easy way of keeps and manage crops data..

# REFERENCE

Bass, L. C. (2012). *Software architecture in practice.* Addision Wesley.

Fowler, M. S. (2003). *a brief Guide to the Standard Object Modeling Language.* Wesley Professional.

Jane K. Drake, P. J. (2012 First Eddition). *Academic Advising Approaches.*

Latrellis, O. (2017). *Academic Advising Systems.*

Ruel, H. J. (2010). *The waterfall Approach Requirement Uncertainity.* International Journal of Information Technology Project Management.

Smith, J. J. (2022). *Supporting Student in Higher Education.*

Virginia N. Gordon, W. R. (Second AEddition). *Academic Advising Approaches.*