



ADDIS ABABA UNIVERSITY  
FACULTY OF NATURAL AND COMPUTATIONAL  
SCIENCES  
DEPARTMENT OF COMPUTER SCIENCE  
**AGRICULTURAL INFORMATION SHARING  
SYSTEM (AISS)**

COSC 4411: FINAL PROJECT I

BY

No. <u>Name</u>	ID No
1.Melkamu Zinabu	UGR/2909/12
2.Yonas Kebede	UGR/8855/12
3.Samuel Kibret	UGR/5875/12
4.Sewlesew Biazen	UGR/2148/12

Project Advisor: Mr. Solomon.T

Date of submission: 19<sup>th</sup> February, 2023

## **DECLARATION**

This is to declare that this project work which is done under the supervision of Mr. Solomon T. and having the title “Agricultural Information Sharing System” is the sole contribution of:

1. Melkamu Zinabu
2. Yonas Kebede
3. Samuel Kibret
4. Sewlesew Biazen

No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: \_\_\_\_\_

### **Group Members:**

Full Name

Signature

---

---

---

---

---

---

---

---

## **CERTIFICATE**

I certify that this BSc final project report entitled “Agricultural Information Sharing System” by:

1. Yonas Kebede
2. Melkamu Zinabu
3. Samuel Kibret
4. Sewlesew Biazen

Is approved by me for submission. I certify further that, to the best of my knowledge, the report represents work carried out by the students.

---

Date

---

Name and Signature of Supervisor

## **ACKNOWLEDGEMENT**

First of all, we would like to thank the Almighty 'God' who helped us in all aspects and gave us good health to do our works effectively and to complete this project documentation successfully. Secondly, we would like to express our deepest appreciation and gratitude to our advisor Mr. Solomon T. for his advice on our project what we have to do timely and efficiently. We have very much appreciated his support, his numerous advices throughout the project, and his careful proofreading. And thirdly we would like to thank Computer Science academic staff for their guideline preparation which helps us as a framework to develop our project.

Finally, we would like to thank our friends who have helped us with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

## **ACRONOMY AND ABBRIVATION**

AISS - Agricultural Information Sharing System

API - Application programming Interface

SMS - Short Message Service

VCS - Version Control System

GDP - Gross Domestic Product

DA - Department Worker

ICs - Information Clerks

RBAC - Role Based Access Control

MOA - Minister of Agriculture

UML - Unified Modeling Language

## Table of Contents

Declaration.....	I
Certificate.....	II
Acknowledgement.....	III
Acronym And Abbreviation .....	IV
List of Figure.....	VIII
List of Table.....	IX
1 CHAPTER ONE: PROJECT PROPOSAL .....	1
1.1 Introduction .....	1
1.2 Problem statement.....	3
1.3 Project Objective .....	3
1.3.1 General Objective .....	3
1.3.2 Specific Objective.....	3
1.4 Scope of the project.....	4
1.4.1 Scope.....	4
1.4.2 Limitation.....	4
1.5 Methodology .....	5
1.5.1 System development methodology .....	5
1.5.2 Investigation (Fact-Finding) Methods .....	5
1.5.3 System analysis and design methodology.....	6
1.6 The significance of the project.....	7
1.7 The beneficiaries of the project.....	7
1.8 Cost Analysis.....	8
1.8.1 Hardware Cost .....	8
1.8.2 Software Cost.....	8
1.9 Time scheduling .....	9

2	CHAPTER TWO: REQUIREMENT ANALYSIS .....	10
2.1	Introduction .....	10
2.2	Current System.....	10
2.2.1	Problems in the existing system.....	11
2.3	Requirement Gathering .....	11
2.3.1	Requirement Gathering Methods .....	11
2.3.2	Results found .....	12
2.4	Proposed System .....	12
2.4.1	Overview.....	12
2.4.2	Functional Requirements .....	12
2.4.3	Non – Functional Requirement.....	13
2.4.3.1	User Interface and Human Factors .....	13
2.4.3.2	Documentation.....	13
2.4.3.3	Hardware Consideration .....	13
2.4.3.4	Performance Characteristics .....	13
2.4.3.5	Error handling and Extreme conditions.....	13
2.4.3.6	Quality Issue .....	14
2.4.3.7	System Modification.....	14
2.4.3.8	Physical environment .....	14
2.4.3.9	Security Issues .....	14
2.4.3.10	Resource Management .....	14
2.4.3.11	Constraints or Pseudo Requirements .....	15
2.5	System Model.....	15
2.5.1	Use case Model .....	15
2.5.1.1	Use Case Diagram .....	15
2.5.1.2	Use Case Description.....	17
2.5.2	Use Case Scenario.....	29

2.5.3	Object Model .....	38
2.5.3.1	Class diagram .....	42
2.5.4	Dynamic Modelling .....	43
2.5.4.1	Sequence Diagram .....	43
2.5.4.2	Activity Diagram .....	49
2.5.5	User Interface.....	57
3	CHAPTER 3 SYSTEM DESIGN .....	61
3.1	Introduction .....	61
3.2	Design Goals .....	61
3.3	Proposed Software Architecture.....	62
3.3.1	Overview.....	62
3.3.2	Subsystem Decomposition.....	63
3.3.3	Hardware/Software Mapping.....	71
3.3.4	Persistent Data Management.....	72
3.3.5	Access Control and Security .....	75
3.4	Detailed Class diagram.....	78
3.5	Packages .....	81
	References .....	83

## List of Figures

Figure 1-1 Time schedule .....	9
Figure 2-1 Use case Diagram of the system .....	16
Figure 2-2 Class Diagram of the System .....	42
Figure 2-3 Login Sequence Diagram .....	43
Figure 2-4 Create Account sequence diagram .....	44
Figure 2-5 Logout sequence Diagram.....	45
Figure 2-6 Manage Jobs Sequence Diagram .....	46
Figure 2-7 Manage Newsfeed Sequence Diagram.....	47
Figure 2-8 Manage Agri-product Sequence Diagram.....	48
Figure 2-9 View Newsfeed Sequence Diagram.....	48
Figure 2-10 Create Account Activity Diagram.....	49
Figure 2-11 Login Activity Diagram .....	50
Figure 2-12 Logout Activity Diagram .....	51
Figure 2-13 Manage DA workers and IC's account Activity Diagram .....	52
Figure 2-14 Manage Agri-products Activity Diagram .....	53
Figure 2-15 Manage Jobs Activity Diagram.....	54
Figure 2-16 Manage Newsfeed Activity Diagram.....	55
Figure 2-17 Q&A Activity Diagram .....	56
Figure 2-18 Login User interface.....	57
Figure 2-19 Homepage User Interface.....	58
Figure 2-20 Manage Newsfeed Page User Interface .....	59
Figure 2-21 View Newsfeed Page User Interface.....	60
Figure 3-1 System Architecture .....	63
Figure 3-2 System Decomposition.....	64
Figure 3-3 User Interface Subsystem.....	65
Figure 3-4 User Management Subsystem .....	67
Figure 3-5 Newsfeed Management Subsystem.....	68
Figure 3-6 Agri-products Management Subsystem .....	69
Figure 3-7 Jobs Management Subsystem .....	69
Figure 3-8 Data Management Subsystem .....	70
Figure 3-9 Deployment Diagram of the system.....	71

Figure 3-10 Object Diagram of the system.....	71
Figure 3-11 Detailed Class Diagram of jobs.....	78
Figure 3-12 Detailed Class Diagram of Newsfeed .....	78
Figure 3-13 Detailed Class Diagram of Answer.....	79
Figure 3-14 Detailed Class Diagram of Question.....	79
Figure 3-15 Detailed Class Diagram of Agri-product .....	80
Figure 3-16 Detailed Class Diagram of Farmers .....	80
Figure 3-17 Packages of the system and their dependency .....	82

## List of Tables

Table 2-1 Manage jobs use case description.....	17
Table 2-2 Manage Agri-products use case description.....	18
Table 2-3 View newsfeed use case description .....	20
Table 2-4 Create account use case description .....	21
Table 2-5 government financial aid policy use case description .....	22
Table 2-6 Q&A session use case description.....	23
Table 2-7 Login use case description.....	24
Table 2-8 Manage newsfeed use case description .....	25
Table 2-9 Logout use case description.....	26
Table 2-10 Manage DA &IC account use case description.....	27
Table 2-11 Manage jobs use case scenario .....	29
Table 2-12 Manage Agri-product use case scenario.....	30
Table 2-13 View newsfeed use case scenario .....	31
Table 2-14 Create account use case scenario.....	32
Table 2-15 Apply for government financial aid policy scenario .....	33
Table 2-16 Q&A session scenario .....	34
Table 2-17 Login use case scenario .....	35
Table 2-18 Manage newsfeed use case scenario.....	36
Table 2-19 Manage DA &IC account use case scenario .....	37
Table 2-20 Logout use case scenario .....	37
Table 2-21 User.....	38
Table 2-22 Newsfeed .....	39
Table 2-23 Product.....	39
Table 2-24 Job.....	40
Table 2-25 Question.....	40
Table 2-26 Answer.....	41
Table 3-1 User collection description .....	72
Table 3-2 Agri-products collection description .....	73
Table 3-3 Jobs collection description .....	73
Table 3-4 Questions collection description.....	74
Table 3-5 Answer collection description .....	74
Table 3-6 Access control of the system .....	77

# 1 CHAPTER ONE: PROJECT PROPOSAL

## 1.1 Introduction

Ethiopia's economy is predominately centered on agriculture, which generates 85% of all employment and 46.6% of the country's GDP. Marketing, processing, and export of agricultural goods are only a few of the many additional economic activities that rely on agriculture. Cereal production makes up the second-largest portion of Ethiopia's economy from the agricultural sector. Cereals are grown on more than 80.0% of agricultural land, employing 60.0% of the rural labor force. Despite having an abundance of rich land for farming, the country fails to produce large yields of cereals because of poor infrastructure, outdated technology, and a lack of machinery and irrigation facilities [\[1\]](#).

The agricultural sector offers opportunities to farmers, industrialists, exporters, aggregators, and other important actors, and the MOA is committed to working with everyone engaged to ensure that the sector will develop at a much faster pace than in the past. Before, throughout the growing season, different pathogens in the crops grown on land, but could not find a viable way to protect the crops produced from those diseases, farmers used to go to the market pesticides merchants, who always suggested about primary medicine and most of them. Hence, crop loss is typically caused by a delay in receiving the appropriate care.

Similar to this, Ethiopia's Meteorological Department is likewise disseminating data on future agricultural-related weather characteristics. Yet as of right now, the influence of such vital information has not effectively reached the end users, the farmers. That is why every year, our farmers still inquire as to whether or not they will receive an adequate harvest if they plant a particular crop or vegetable. Effective analysis of the relevant soil and weather data can provide a valuable response to this important topic, enabling the farmer to make insightful decisions. The proposed AISS will be made to deliver instantaneous solutions to such queries as well as constructive criticism to the framers, even in the most remote regions of our nation.

Even though our nation has made remarkable technological progress, soil quality and climatic conditions still have a significant impact on agricultural productivity. Hence, in order to undertake effective agriculture, the nation's constantly growing population requires a rigorous assessment of its soil and climate resources.

Smart Phones are now widely available even in far-off villages as a result of the enhanced communication technology reaching the remote nodal regions of our nation. But sadly, the true advantages of smartphones have not yet trickled down to the locals, particularly the farming sector, which is the engine of our nation's economy. This is because most online applications for smartphones were created only for urban environments. For the nodal Ethiopian farmers, the proposed initiative will concentrate on developing web application that provide agricultural information.

After carefully examining these problems, we began developing the "AISS" online web application, which was largely inspired by the difficulties faced by farmers. Farmers frequently struggle to obtain crop and weather reports, but many are unable to do so because they are unsure of where to look. The farmers will find our new web application useful in this regard, but they'll need a smartphone. Given that there are more than 120 million people living in Ethiopia, many people will use the application [2]. Farmers might get the assistance they require from these. Our app will be useful in this situation because farmers must pay a commission to sell their goods. It boosts self-assurance and calls for the creation of a beneficial app for rural residents, particularly farmers that will not only improve their standard of living but also serve to support and grow our farmer's business, allowing our application to assist the farmer in making a larger profit. The suggested app will generally ensure that farmers gain by providing information on the crops, facilitating the market for their products, and enabling them to hire the staff they require.

## **1.2 Problem statement**

Ethiopian farmers have little to no knowledge of the government policy on their work. When the MOA and other concerned bodies want to share agricultural information and other related information, they use antiquated and time-consuming methods like calling meetings, using the radio, or posting on notice boards. They lack information on how next season's climate will fare, they have no guarantee if the crop they produced will even recover the cost they incurred. When the crops are ripe, farmers find it difficult to harvest because doing so requires a lot of man power and equipment. These are just the tip of the ice berg. Farmers can't even sell their crop at the right price, almost all of them never had enough and up-to-date information.

## **1.3 Project Objective**

### **1.3.1 General Objective**

The primary goal of the system is to design and create web-based applications that will aid in modernizing or boasting the agricultural sector and ensuring the benefit of the farmers by supplying information about the crops, facilitating the market information for their products, and enabling them to hire the human resources they need.

### **1.3.2 Specific Objective**

To accomplish the above general objective, our system the following specific objectives:

- Develop responsive user interface
- Develop web API to connect front-end with back-end
- Develop a SMS based service for farmers
- Make the system search engine optimizable for easy access
- Design a database system in order to provide a persistent data to users.
- Implement a role-based access software solution that will achieve functional and non-functional needs of our users.

## **1.4 Scope of the project**

### **1.4.1 Scope**

The project is focused on creating a web based application that enables the minister of agriculture to communicate with farmers and provide them with information about crops, weather, government policy, and related information. It also enables farmers to request agricultural information and receive responses, without having to be present or wait for a formal meeting between DA employees and them. The farmers can sell their products without the involvement of third party and find the labor they require. On the other hand, the system enables job seekers to find employment opportunities and buyers who wish to purchase the products of the farmer to do so. For the farmer as well as the buyers, it also offers market information.

### **1.4.2 Limitation**

The system has been designed to address the issues farmers faced regarding to lack of information related to agriculture, with some limitations noted.

- The system is limited to some languages
- Only accessible by smart phone or computer with an internet connection but some information is delivered to farmers through a SMS message.
- The system also lack of an online payment integration when farmers sells or the consumer buys products
- Doesn't have an instant mobile application.

## **1.5 Methodology**

### **1.5.1 System development methodology**

In this project, an object-oriented system analysis and design methodology is applied because it has got many advantages compared with structured analysis and design. The most important ones are flexibility to change, easy transition from one phase to the next, easy traceability and it suggest greater user involvement. Waterfall Model software development methodology is the software model of this project. It will be broken down in to numerous parts; The Requirement Analysis, System Design and Implementation and Testing.

### **1.5.2 Investigation (Fact-Finding) Methods**

We used different data collection techniques like observation, discussion with our friend and families and also internet to gather accurate information about the existing system and the requirements for the new system.

To get precise data, our team used the following data collection techniques-

1. **Interview:** to get the basic information and background information about the existing system our team asked different students, comes from different corner of the country, about their means of getting information related to farming. Plus, we have asked farmers their livelihood depends on farming. Moreover we asked agricultural experts that engage in that field of area.
2. **Direct observation:** even though interview is very important to gather information, direct observation is simple and helpful to get real problems on the ground. As we are from countryside, we have observed how our families get crop information and government schemes.
3. **Document Analysis:** - To gain some insight as to how the need for a system arises to identify the part of the commission associated with the problem. Before we plan to develop this system we have referred existing system records, and read the way of data gathering they follow manually.

### **1.5.3 System analysis and design methodology**

The system will try to solve the problem using web technologies and hence the system shall have a website that is capable of solving the problems listed above. The system shall be implemented using the following technologies.

#### **Tools for written documentation we intend to use**

- Microsoft office Word 2016
- Edraw Max
- MS Visio
- Adobe XD: Online User Interface design maker

#### **Tools for the implementation phase**

##### **For front-end**

- HTML5: - Is currently the most stable and improved version that has a lot of support by the developers.
- CSS: - Is also the most stable version that have many tutorials and can be used to add beauty to other user interface.
- JavaScript: - All the validation task and animation will be developed by JavaScript.
- Bootstrap5:- We use it in the front-end development because of its cross-platform compatibility and ability to build beautiful user interface
- React:-It is a free and open-source front-end JavaScript library for building user interfaces based on UI components.

##### **For Back-end**

- MongoDB is a source-available cross-platform document-oriented database program.
- Express.js or simply Express, is a back end web application framework for building RESTful APIs with Node.js

#### **Tools for modeling**

- UML: - During the analysis and design phase, Unified Modeling Language (UML) will be used for the graphical representation of some system concepts of the project using Microsoft Visio 2016.

## **Code editing tool**

- Visual Studio Code: - We use this text editor because it is very powerful and has an easy to use interface. In addition, we will be VCS to work together online.

## **Operating systems on which this project will be configured**

- Windows: - This project can easily be configured on Windows operating system.  
For running this project on Windows operating system.
- Linux: - We can run this project also on all version of Linux operating system.

## **1.6 The significance of the project**

Since more than 80% of the country population is engaged in the agricultural sector, solving the problems that stem from lack of information will increase their productivity [1]. Helping solve some of these problems will no doubt have a pronounced effect both to the lives of the farmers and to the overall economy of the country. It means that the agriculture sector will do its part to bring the country to a better stage.

## **1.7 The beneficiaries of the project**

The project will have multiple beneficiaries ranging from farmers to consumers. Each with their needs taken into account. The farmers, the main beneficiaries, will gain multiple advantages which were not possible before now. They will access expert information anytime anywhere without making appointment with a DA worker. In addition to these they will get a detailed analysis of the MOA's policy towards the sector, they will get seasonal information and also they will have an idea in which direction the price of the crops he/she farmed will swing. Our project benefits reach beyond farmers and unto Buyers, Labor workers and MOA. Buyers can now directly communicate with farmers to buy agricultural products. The MOA will have a direct channel to link up with farmers. Our project takes in mind labor workers too. Labor workers can use the website as a job application posting site. It is safe to say the project we have in mind will have a range of grantees.

## 1.8 Cost Analysis

### 1.8.1 Hardware Cost

Number	Material	amount	Price /ETB	Total price/ETB
1	HP laptop	1	20,000	20,000
2	A4 size paper	1 destin	200	200
3	Pen	5	15	75
4	Pencil	5	10	50
5	Flash	1	500	500
total	20,825			

Table 1.1 Hardware Cost

### 1.8.2 Software Cost

Number	Software	Cost/ETB
1	Microsoft Office 2016	Free
2	Microsoft Visio 2016, Edraw Max	Free
3	Adobe XD, VS Code	Free

Table 1.2 Software Cost

## 1.9 Time scheduling

ID	Task	Start	Duration	Finish	Oct 2022	Nov 2022	Dec 2022	Jan 2023	Feb 2023	Mar 2023	Apr 2023	May 2023
1	<b>Project proposal</b>	03/10/2022	30d	11/11/2022								
2	Problem identification	03/10/2022	10d	14/10/2022								
3	Setting the project scope	17/10/2022	5d	21/10/2022								
4	Selecting the methodology	24/10/2022	5d	28/10/2022								
5	Proposal document draft	31/10/2022	7d	08/11/2022								
6	Proposal document final	09/11/2022	3d	11/11/2022								
7	<b>Requirement Analysis</b>	14/11/2022	30d	23/12/2022								
8	Requirement gathering	14/11/2022	8d	23/11/2022								
9	Requirement organizing	24/11/2022	6d	01/12/2022								
10	System model	02/12/2022	10d	15/12/2022								
11	Requirement analysis draft	16/12/2022	4d	21/12/2022								
12	Requirement analysis final	22/12/2022	2d	23/12/2022								
13	<b>System design</b>	26/12/2022	40d	17/02/2023								
14	System decomposition	26/12/2022	5d	30/12/2022								
15	Hardware and software mapping	02/01/2023	5d	06/01/2023								
16	Persistent data management	09/01/2023	8d	18/01/2023								
17	Access control and security	19/01/2023	5d	25/01/2023								
18	Subsystem services	26/01/2023	5d	01/02/2023								
19	Detailed class diagram	02/02/2023	5d	08/02/2023								
20	System design document draft	09/02/2023	5d	15/02/2023								
21	System design document final	16/02/2023	2d	17/02/2023								
22	<b>System development</b>	20/02/2023	70d	26/05/2023								
23	Frontend development	20/02/2023	30d	31/03/2023								
24	Backend development	03/04/2023	24d	04/05/2023								
25	Integration and finalize	05/05/2023	15d	25/05/2023								

Figure 1-1 Time schedule

## **2 CHAPTER TWO: REQUIREMENT ANALYSIS**

### **2.1 Introduction**

Agricultural Information Sharing System at its heart strives to rid away the problems faced by farmers in their day to day activities. Most of these problems stem from lack of scalable and adequate information sharing platforms. By designing a web application, the project at its core objective intends to make the flow and access of information both easy and applicable in the Ethiopian context. AISS, by utilizing the technological advancements the world has reached, has a goal of building a knowledge sharing platform. The project's main goal is to develop a system that enables farmers inquire about the crop, weather, laws, and other relevant sectors without being obliged to wait for a formal meeting with a sanctioned authoritative body. The platform will have different actors each with their own responsibilities and privileges to help farmers get reliable information. By preventing the participation of a third party between farmers and consumers, it also acts as a bridge to link producers to the market. And also it act as bridge to link farmers to laborers. This will have a profound effect on reaching the market and cutting the middlemen. The project will take into consideration the technological inefficiency and backwardness that plagues the Ethiopian countryside. To that end the web application will have an SMS integration.

### **2.2 Current System**

The agriculture sector in Ethiopia currently does possess platforms for sharing and accessing information. These platforms are archaic and old. But they are still the main and in most places they are the only methods of sharing and accessing information. Most of these platforms require from small to none financial costs. Also they are mobile, adding for their success among the farmers. Some of these platforms include Radio Programs, Advertisement Boards, DA sanctioned meeting and they are still popular by the farmers. When farmers want to sell their products or hire people, they contact them through middlemen. They did not have the opportunity to find a direct way.

### **2.2.1 Problems in the existing system**

Thought these platforms are popular, sometimes not without fault. Platforms' reliance on old technology has made it difficult to keep pace with modern technological developments. The platforms made farmers use only outdated gadgets. If we take radio programs for example, one of the problems we face is that the information presented is strictly audio. One of the most important symbols of information sharing is visual. Radio programs don't work on demand either. Farmers should take time to listen to the programs. Another problem they face is that direct interaction takes place using two different technologies. If a farmer wants to ask a question, they should call to the radio station. Advertisement boards' have problems ranging from easy accessibility to almost none existent direct Q&A session. SMS based systems still have not made a significant headway into their customers mainly because the information they impart is not rich and diverse enough. DA sanctioned meetings, while they have a direct Q&A session with farmers and the information are reliable, the big problem is once the session is over farmers must wait for the next regular meeting before they ask questions. This is very critical since most of these issues need to be answered right away. In addition as we have said now the farmers do not have direct link with laborers and buyers, they need to pay a commission in order to sell their product and also to get laborers.

## **2.3 Requirement Gathering**

### **2.3.1 Requirement Gathering Methods**

We have followed different methods to provide us with some insights into the system that we are developing. The methods we followed in gathering information to develop our projects include phone interviews and physical interviews. The phone interviews were carried out with farmers. We inquired about how they get information that are necessary for their day to day activities. We have also asked the advantages and disadvantages they face when engaging on these information gathering methods. And we have asked if there are any issues that are overlooked and not given the proper due emphasis when projects like these are pursued. The physical interviews while we have asked and investigated how they gain information, we focused on observing how farmers get information. We wanted to know how much our project would be technologically applicable and socially accepted in countryside Ethiopia. The other method we used to gather the requirements is by reading

the documents and browsing in the internet to get more insights about the system that we are developing.

### **2.3.2 Results found**

After analyzing and investigating the information we collected, we have found some interesting and new ideas. One of the ideas that were repeatedly raised by the farmers was market integration. The crops they produce do not reach market. The other problem was lack of both internet access and the price of smart phones. These means feature phones are mainstream among the farmers. To that end we have to make our site compatible with feature phones. The other important issues raised were a Q&A session with DA workers, a method by which they would have knowledge of the prices of their crops and weather forecast. One very interesting suggestions raised was labor. Farmers need big number of labor during the time of harvesting. They have asked a way to hire labor workers.

## **2.4 Proposed System**

### **2.4.1 Overview**

The proposed system is developing a web based application system that, at its core objective is helping farmers get information that is integral in their day to day activities.

### **2.4.2 Functional Requirements**

In this part of the section we will outline the functional requirement of the proposed project. The functional requirement is a set of tasks that are implemented by the programmer to help users accomplish their intended goal.

1. The system administrator will manage DA and Information clerk accounts
2. The farmer can manage available agricultural products information
3. The farmer can manage available job information for labor workers
4. The farmer can see new agricultural technology information
5. The buyers can see available products and contact the seller
6. The farmer can see weather information
7. The farmer and buyers can see market information
8. The farmer can engage in a Q&A session with DA workers online using the system
9. The farmer can see government policies and directives

10. Labor workers can see available jobs and contact the employer
11. The information clerk can manage newsfeed information like weather, crop detail, new agricultural technology, government policies and market information

### **2.4.3 Non – Functional Requirement**

Non-Functional requirements are quality attributes. They focus on the user expectation and experience. They help to enhance the performance of the product in development.

#### **2.4.3.1 User Interface and Human Factors**

The user interface of the product will be responsive. This will mean the website will be accessible both from desktop and mobile phones. As the main users are farmers, not adept at technology, The UI will be easy to navigate and simple to accomplish tasks

#### **2.4.3.2 Documentation**

Each step of the process development cycle will be documented based on industry standards. The implementation will have the necessary explanation for what each module stands to implement. This will be done by using the appropriate commenting method. This will also make it easy for code maintainers and new developers to get started.

#### **2.4.3.3 Hardware Consideration**

The client-side services shall work on all devices and all web browsers. The website is implemented using responsive method. Meaning, both cellphones, laptops and desktop computers will be able to use it. Most feature phones can access it too. To a limited number of feature phones that can't access the internet the website will have SMS integration.

#### **2.4.3.4 Performance Characteristics**

The website will be built to make it lightweight. This means the use of animation and videos during the development of the website will be limited. This will help users who have low internet access speed to load and access the site in a fairly small amount of time.

#### **2.4.3.5 Error handling and Extreme conditions**

For an error data input the website will throw the appropriate message to the user, indicating the mistake he/she made during inserting data. Incorrect data entry and data destruction during processing will be managed by the system. This will eliminate the system crashing during an invalid data input. If appropriate, alert box will be utilized to notify the user the error committed.

#### **2.4.3.6 Quality Issue**

The system is expected to run 24 hours and 7 days a week. In rare moments the site may be taken down to handle extreme situations. Power outages could become a major problem keeping the site running all the time. Since the site is developed taking the RBAC (Role Based Access control system) principles, one user can not cause a system crash.

#### **2.4.3.7 System Modification**

The product for now is limited to web application. But in the near future we intend to build a mobile application version of it. With the popularity of open source projects we will enlist the help and talent of anyone interested. The merge and pull request will be handled by the development team. The mobile application will be built for both android and IOS mobile operating systems. The service provided by the SMS integration at the time of writing this is limited. We intend to make that part of the product deploy rich and diverse services.

#### **2.4.3.8 Physical environment**

The physical environment for the machines on the client side do not require special treatment. The environments that are applicable for any mobile gadget is perfectly suitable for our case. The server machines will have to be put in a temperature that is within 19 to 21centigrade to run efficiently.

#### **2.4.3.9 Security Issues**

Each user will have a strong password. The password must meet the required strength. This is measured based on the length of the character, the alphanumeric characters used. The system will warn users not to have passwords that can be cracked with dictionary attacks. The website will take the necessary measure to guard against SQL injection attacks. To defend against brute force attacks each user will only be given 3 trials before their account is locked for the next 24 hours.

#### **2.4.3.10 Resource Management**

The product requires users to have varieties of devices. Smart phone, feature phones, laptops or Desktop are required in order to access the platform. The people hosting the site will need to have a power that runs 24 hours. Clients are required to have an internet connection that is at least 2G to access the website. But for feature phones 1G connection suffices. We save a cookie file on our customers' browsers in order to have a personalized experience.

#### **2.4.3.11 Constraints or Pseudo Requirements**

Power is needed to keep both client side and server side machines running. We need an internet connection to load the site on the client machines. Political and Social unrest sometimes cause the government to restrict the internet access. Our main customer, farmers, most times do not own a Smart phone and rarely, they have a laptop.

### **2.5 System Model**

#### **2.5.1 Use case Model**

Use case is a technique for capturing the functional requirements of a system. Use cases work by describing the typical interaction between the users of a system and the system itself, providing a narrative of how a system is used.

##### **2.5.1.1 Use Case Diagram**

Figure 2-1 is used to illustrate how to capture the dynamic aspect of the system. It's representation of user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. The use case model for our proposed system is described below:

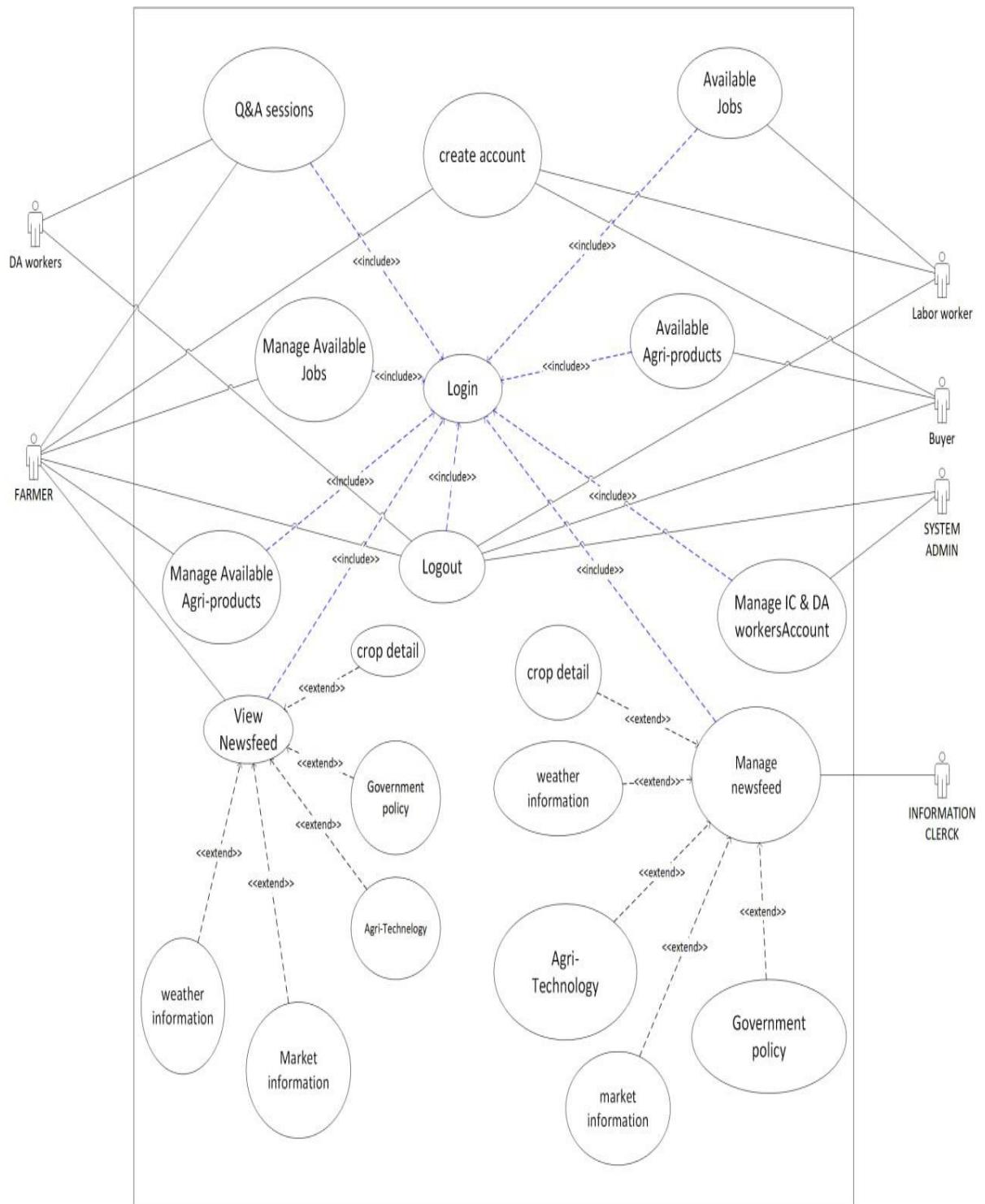


Figure 2-1 Use case Diagram of the system

### 2.5.1.2 Use Case Description

Table 2-1 Manage jobs use case description

Use case id	001
Use case name	Manage jobs
Participating actors	Farmer
Use Case description	Administering available jobs by posting, updating and deleting.
Pre-conditions	The farmer have to logged in
Flow of events	<ol style="list-style-type: none"> <li>1. The user navigates the system home page</li> <li>2. The user go to manage jobs page</li> <li>3. The system redirects and displays the list of jobs that the farmer formerly posted and add new job button.</li> <li>4. The user selects one of the jobs</li> <li>5. The system redirects to the appropriate page with deleting and editing buttons</li> <li>6. The user clicks edit button</li> <li>7. The system makes the job content editable</li> <li>8. The user edits the job and clicks save button</li> <li>9. End of Use Case</li> </ol>
Alternate flow	<ol style="list-style-type: none"> <li>4. [If the farmer wants to add new job]             <ol style="list-style-type: none"> <li>4.1. The farmer clicks add new job button</li> <li>4.2. The system redirects and displays job filling form</li> <li>4.3. The Farmer enter necessary contents of new job and saves</li> <li>4.4. The system displays a confirmation pop-up message to the farmer and redirects to manage jobs page.</li> <li>4.5. End of use case</li> </ol> </li> </ol>

	<p>6.[if the Farmer wants to delete the job]</p> <p>6.1.The farmer clicks delete job button</p> <p>6.2. The system a pop up message to confirm the consent of the farmer</p> <p>6.3. The farmer confirms to delete</p> <p>6.4.The system deletes the job ,displays a confirmation pop-up message and redirects to manage jobs page</p> <p>6.5. End of use case</p>
Post-conditions	The job is added, updated or deleted.

Table 2-2 Manage Agri-products use case description

Use case id	002
Use case name	Manage Agri-products
Participating actors	Farmer
Use Case description	Administering available Agri-products by posting, updating and deleting.
Pre-conditions	The farmer have to logged in
Flow of events	<ol style="list-style-type: none"> <li>1. The user navigates the system home page</li> <li>2. The user go to manage Agri-products page</li> <li>3. The system redirects and displays the list of Agri-products that the farmer formerly posted and add new product button.</li> <li>4. The user selects one of the Agri-products</li> <li>5. The system redirects to the appropriate page with deleting and editing product buttons</li> <li>6. The user clicks edit button</li> <li>7. The system makes the Agri-product content editable</li> <li>8. The user edits the Agri-product and clicks save button</li> </ol>

	9. End of Use Case
Alternate flow	<p>4. [If the farmer wants to add new Agri-product]</p> <p>    4.1. The farmer clicks add new product button</p> <p>    4.2. The system redirects and displays Agri-product filling form</p> <p>    4.3. The Farmer enter necessary contents of new Agri-product and clicks save/Post button</p> <p>    4.4. The system displays a confirmation pop-up message to the farmer and redirects to manage Agri-products page.</p> <p>    4.5. End of use case</p> <p>6. [if the Farmer wants to delete the Agri-product]</p> <p>    6.1.The farmer clicks delete product button</p> <p>    6.2. The system a pop up message to confirm the consent of the farmer</p> <p>    6.3. The farmer confirms to delete</p> <p>    6.4.The system deletes the Agri-product, displays a confirmation pop-up message and redirects to manage Agri-products page</p> <p>    6.5. End of use case</p>
Post-conditions	The (new) product is added/updated or deleted.

Table 2-3 View newsfeed use case description

Use case id	003
Use case name	View newsfeed
Participating actors	Farmer, labor worker and buyer
Use case description	The user view the information related to agriculture as newsfeed.
Pre-conditions	The user have to logged in.
Flow of events	<ol style="list-style-type: none"> <li>1. The user navigates the system home page</li> <li>2. The users hovers on view newsfeed button from the navigation bar</li> <li>3. The user selects the type of information he/she wants to view from the drop down list of newsfeed</li> <li>4. The system redirects and display available newsfeed according to the user interest.</li> <li>5. The user view and comment to the newsfeed he/she is looking</li> <li>6. The user leaves the page by clicks "close" button or return back to step 2</li> <li>7. End of use case</li> </ol>
Alternate flow	None
Post-conditions	The user viewed a newsfeed he/she want.

Table 2-4 Create account use case description

Use case id	004
Use case name	Create account
Participating actors	Farmer, buyer and labor worker
Use case description	The user creates an account in the system
Pre-conditions	The user to the system login page
Flow of events	<ol style="list-style-type: none"> <li>1. The user clicks sign up button</li> <li>2. The system redirects to "create account" page with appropriate form</li> <li>3. The user enters the necessary credentials in the displayed form</li> <li>4. The user clicks submit button</li> <li>5. The system sends user verification code</li> <li>6. The user enters the verification code and click verify</li> <li>7. The system approves and shows success message to the user</li> <li>8. End of use case</li> </ol>
Alternate flow	<p>5 [if the user entered invalid credential(s)]</p> <p>5.1a. The system refreshes with an error message (invalid credentials)</p> <p>5.2a. The user returns to step 3</p> <p>6 [if the user enters invalid confirmation code]</p> <p>6.1a. The system refreshes with an error message (invalid verification code)</p> <p>6.2a. The user clicks "send verification code" again</p>
Post-conditions	New user is registered

Table 2-5 government financial aid policy use case description

Use case id	005
Use case name	Apply for government policy
Participating actors	Farmer
Use case description	Apply for government policy like bank loans, subsidies and other infrastructure opportunity
Pre-conditions	The farmer have to logged in
Flow of events	<ol style="list-style-type: none"> <li>1. The user navigates the system home page.</li> <li>2. The farmers hovers on newsfeed and clicks government policy from the drop down list.</li> <li>3. The system redirects to government policy newsfeed page</li> <li>4. By dragging the up and down the farmer can view government policy information and apply for it by clicking "apply" button.</li> <li>5. The system redirects and displays an application form</li> <li>6. The farmer fills the form with information asked and clicks "submit" button.</li> <li>7. The system displays success message to the farmer.</li> <li>8. End of use case</li> </ol>
Alternate flow	<p>7 [if the farmer fills the form with invalid data]</p> <p>    7.1a. The system refreshes with an error message          (invalid data filled)</p> <p>    7.2a. The farmer go to step 6</p>
Post-conditions	The farmer applied for government policy.

Table 2-6 Q&A session use case description

Use case id	006
Use case name	Q&A session
Participating actors	Farmer & DA worker
Use case description	Asking and answering question about crop information and related to it
Pre-conditions	The user has to logged in
Flow of events	<ol style="list-style-type: none"> <li>1. The user navigates the system home page</li> <li>2. The user clicks on Q&amp;A session button from navigation bar</li> <li>3. The system redirects to Q&amp;A page based on user role</li> <li>4. The farmer fills question form with necessary information and clicks “submit”</li> <li>5. The system sends a success message back to the farmer</li> <li>6. End of use case</li> </ol>
Alternate flow	<p>4 [if the user is a DA worker]</p> <p>    4.1a The DA worker search for specific question to answer</p> <p>    4.2a The system enables to answer the question</p> <p>    4.3a The DA worker write the answer and submits</p> <p>    4.4a The system displays the question and its answer as newsfeed</p> <p>    4.5a End of use case</p>
Post-conditions	The farmer asked a question or the DA worker answered a question

Table 2-7 Login use case description

Use case id	007
Use case name	Login
Participating actors	Farmer, Labor worker, buyer, DA worker, IC, System administrator
Use Case Description	Signing(logging) in to the system
Pre-conditions	None
Flow of events	<ol style="list-style-type: none"> <li>1. The user navigates to the site's Homepage.</li> <li>2. The system displays an option for login.</li> <li>3. The user fills the required information and clicks login button [Alternate 1].</li> <li>4. The system redirects the user to his/her respective page.</li> <li>5. End of use case</li> </ol>
Alternate flow	<p>4 [if the user fills the wrong username and/or password]</p> <p>    4.1a. The system displays the wrong message dialogue and resets the login form empty.</p> <p>    4.2a go to step 3</p>
Post-conditions	The user is logged into the system.

Table 2-8 Manage newsfeed use case description

Use case id	008
Use case name	Manage information
Participating actors	IC
Use case description	Adding ,editing and deleting different information in the system
Pre-conditions	The user must be logged in to the system
Flow of events	<ol style="list-style-type: none"> <li>1. The user clicks on manage information from the page</li> <li>2. The system redirects to manage information page and displays the list of types of information.</li> <li>3. The user clicks one of the type of information listed</li> <li>4. The system redirects and displays a list of posted information and “add new information” button</li> <li>5. The user clicks “add new information” button from the page</li> <li>6. The system redirects to adding information page and displays a new information form</li> <li>7. The user(IC) fills the form and clicks “post” button</li> <li>8. The system displays a success message with information id</li> <li>9. End of use case</li> </ol>
Alternate flow	<p>5 [if the user(IC) wants to edit an existing information]</p> <p>5.1a. The user selects an information he/she wants to edit from the list of information</p> <p>5.2a. The system displays the message with “edit” and “delete” button</p> <p>5.3a. The user clicks “edit” button</p> <p>5.4a The system makes the information selected editable</p> <p>5.5a. The user performs edition and clicks save</p>

	<p>5.6a. The system displays a success pop-up message to the user</p> <p>5.7a. End of use case</p> <p>5.3a [If the user wants to delete an existing information]</p> <p>5.31a. The user clicks “delete” tab</p> <p>5.32a. The system deletes the information and displays a success pop-up message to user</p> <p>5.33a. End of use case</p>
Post-conditions	The (new) information is added, updated or deleted.

Table 2-9 Logout use case description

Use case id	010
Use case name	Logout
Participating actors	Farmer, Labor worker, buyer, System administrator, IC, DA workers
Use case description	Allow user to logout from the system.
Pre-conditions	None
Flow of events	<ol style="list-style-type: none"> <li>1. Once an actor is ready to exit the system, he/she clicks on logout button.</li> <li>2. The system ask user to confirm logout process.</li> <li>3. The system redirects him/her to the homepage and terminates his/her session.</li> </ol>
Alternate flow	<p>2 [if the user does not confirm logout]</p> <p>2.1. The system cancels logout process.</p>
Post-conditions	The user is logged out from the system.

Table 2-10 Manage DA &IC account use case description

Use case id	009
Use case name	Manage DA &IC account
Participating actors	System administrator
Use case description	Creating ,updating and removing DA and IC accounts from the system
Pre-conditions	The user have to logged in to the system
Flow of events	<ol style="list-style-type: none"> <li>1. The System administrator clicks “manage account” tab from the page</li> <li>2. The system redirects to manage account page with the list of available accounts and “add new account” button</li> <li>3. The System administrator clicks “add new account” button</li> <li>4. The system redirects to create account page with account creation form</li> <li>5. The System administrator fills all required information and clicks “Add”</li> <li>6. The system shows success message</li> <li>7. End of use case</li> </ol>
Alternate flow	<p>3 [if the user wants to update an existing account]</p> <p>3.1.The System administrator selects the account he/she wants to update and clicks “update” tab</p> <p>3.2 The system allows the user to edit the selected account and makes it editable</p> <p>3.3 The System administrator edits the account and clicks “save”</p> <p>3.4 The system shows success message</p> <p>3.5 end of use case</p> <p>3 [if the user wants to delete an existing account]</p>

	<p>3.1. The System administrator selects the account he/she wants</p> <p style="padding-left: 40px;">to remove and clicks “remove” tab</p> <p>3.2 The system removes the account from the system</p> <p>3.3 The system shows success message</p> <p>3.4 end of use case</p> <p>6[if the user enters an invalid information]</p> <p style="padding-left: 40px;">6.1 The system displays an error message</p> <p style="padding-left: 40px;">6.2The System administrator go to step 5</p>
Post-conditions	The (new) DA worker or IC account is created, updated or removed.

### 2.5.2 Use Case Scenario

A scenario is a hypothetical series of actions and experiences that individuals may have when attempting to use computer systems and applications. A series of scenarios that explain the system's primary functions and event flow have been used to depict the system in this part. The chosen cases that follow describe some actual uses of the suggested system.

Table 2-11 Manage jobs use case scenario

Use case id	001
Use case name	Manage jobs
Participating actors	Farmer(Alemu)
Use case description	Administering available jobs by posting, updating and deleting.
Flow of events	<ol style="list-style-type: none"><li>1. Alemu navigates to the site's homepage.</li><li>2. Alemu go to manage jobs page</li><li>3. The system redirects and displays the list of jobs that the farmer formerly posted and add new job button.</li><li>4. Alemu selects harvesting</li><li>5. The system displays the details of harvesting job that can be deleted and edited.</li><li>6. Alemu clicks edit button</li><li>7. The system makes the job content editable</li><li>8. Alemu edited harvesting job</li><li>9. Alemu clicked save button and left the page.</li></ol>

Table 2-12 Manage Agri-product use case scenario

Use case id	002
Use case name	Manage Agri-products
Participating actors	Farmer (Alemu)
Use case description	Administering available Agri-products by posting, updating and deleting.
Flow of events	<ol style="list-style-type: none"> <li>1. Alemu navigates to the site's homepage.</li> <li>2. Alemu go to "manage Agri-products page"</li> <li>3. The list of Agri-products that Alemu formerly posted and a button to add new product button.</li> <li>4. Alemu selects a wheat product that posted recently</li> <li>5. The details of the post is displayed with edit and delete button on the right of the page</li> <li>6. Alemu clicks edit button</li> <li>7. Alemu update the post by entering new detail of the post.</li> <li>8. Alemu clicks save button and left the page.</li> </ol>

Table 2-13 View newsfeed use case scenario

Use case id	003
Use case name	View newsfeed (information)
Participating actors	Farmer(Alemu), user(labor (Demelash)and consumer(Zinash))
Use case description	The targeted user can access the information
Flow of events	<ol style="list-style-type: none"> <li>1. Alemu, Demelash or Zinash navigates the system's landing page</li> <li>2. Alemu, Demelash or Zinash hovers on view newsfeed button from the navigation bar</li> <li>3. Alemu ,Demelash or Zinash selects crop information from the drop down list of newsfeed</li> <li>4. Crop information(s) is(are) displayed in chronological order according to their time they were posted</li> <li>5. Alemu, Demelash or Zinash view and comment to the crop information viewed.</li> <li>6. Alemu, Demelash or Zinash left the page by clicking "close" button</li> </ol>

Table 2-14 Create account use case scenario

Use case id	004
Use case name	Create account
Participating actors	Farmer(Alemu), User(consumer(Zinash) and labor(Demelash))
Use case description	The user creates an account in the system
Flow of events	<ol style="list-style-type: none"> <li>1. Alemu, Zinash or Demelash clicks sign up button(link)</li> <li>2. A page with create account form is displayed</li> <li>3. Alemu, Zinash or Demelash enters full name, age, gender, password, and others</li> <li>4. Alemu, Zinash or Demelash clicks submit button</li> <li>5. Alemu, Zinash or Demelash received a confirmation message with a code on it.</li> <li>6. Alemu, Zinash or Demelash enters the code and clicks verify</li> <li>7. Alemu, Zinash or Demelash receives success message.</li> </ol>

Table 2-15 Apply for government financial aid policy scenario

Use case id	005
Use case name	Apply for government financial aid policy
Participating actors	Farmer (Alemu)
Use case description	Apply for government schema like bank loans, subsidies and other infrastructure opportunity
Flow of events	<ol style="list-style-type: none"> <li>1. Alemu navigates the system landing page</li> <li>2. Alemu hovers on newsfeed and clicks government aid from the drop down list</li> <li>3. Government policy newsfeed page is displayed.</li> <li>4. By dragging the up and down Alemu views government policy information and apply for it by clicking "apply" button to the right of the information.</li> <li>5. The application form is displayed.</li> <li>6. Alemu fills the form with information asked and clicks "submit" button.</li> <li>7. Alemu receives success message from the system.</li> </ol>

Table 2-16 Q&A session scenario

Use case id	006
Use case name	Q&A session(Asking Question)
Participating actors	Farmer (Alemu)
Use Case Description	Asking question about crop information and related to it
Flow of events	<ol style="list-style-type: none"> <li>1. Alemu navigates the system landing page</li> <li>2. Alemu clicks on Q&amp;A session button from navigation bar</li> <li>3. A page with asking question form is displayed.</li> <li>4. Alemu fills question form with necessary information and clicks “submit”</li> <li>5. Alemu receives success message.</li> </ol>

Table 2-17 Login use case scenario

Use case id	007
Use case name	Login
Participating actors	Farmer(Alemu), Labor(Chala), Consumer(Zinash), DA worker(Mr Samuel), IC(Ms Almaz), System administrator(Mr Natnael)
Use case description	Signing(logging) in to the system
Flow of events	<ol style="list-style-type: none"> <li>1. Alemu, Chala, Zinash, Mr Samuel, Ms Almaz, or Mr Natnael navigates to the site's Homepage.</li> <li>2. The system displays an option for login.</li> <li>3. The user enters wrong email (example alemug12@gmail.com) and password</li> <li>4. The system shows an error message to the user and make the login form empty</li> <li>5. The user enter email (example <a href="mailto:alemu12@gmail.com">alemu12@gmail.com</a>) and password correctly</li> <li>6. The system lets the user enter the respective page.</li> </ol>

Table 2-18 Manage newsfeed use case scenario

Use case id	008
Use case name	Manage newsfeed
Participating actors	IC(Ms Almaz)
Use case description	Adding, editing or deleting agricultural newsfeeds
Flow of events	<ol style="list-style-type: none"> <li>1. Ms Almaz clicks on manage newsfeed from the page</li> <li>2. The system redirects to manage newsfeed page and displays the list of types of information.</li> <li>3. Ms Almaz clicks “crop” to manage crop newsfeed</li> <li>4. A list of posted information and “add new information” button are displayed</li> <li>5. Ms Almaz clicks “add new information” button from the page</li> <li>6. The system leads Ms. Almaz to adding newsfeed page and displays a newsfeed form</li> <li>7. Ms Almaz fills the form and clicks “post” button</li> <li>8. The system displays a success message.</li> </ol>

Table 2-19 Manage DA &IC account use case scenario

Use case id	009
Use case name	Manage DA workers and IC account(Adding new account)
Participating actor	System Administrator(Mr Natnael)
Use case description	Creating, Updating and Removing DA and IC workers
Flow of events	<ol style="list-style-type: none"> <li>1. Mr Natnael clicks “manage account” tab from the page</li> <li>2. The system redirects to manage account page with the list of available accounts and “add new account” button</li> <li>3. Mr Natnael clicks “add new account” button</li> <li>4. The system redirects to create account page with account creation form</li> <li>5. Mr Natnael fills all correct required information and clicks “Add”</li> <li>6. The system shows success message</li> </ol>

Table 2-20 Logout use case scenario

Use case id	010
Use case name	Logout
Participating actors	Farmer(Alemu), Labor(Chala), Consumer(Zinash), DA worker(Mr Samuel), IC(Ms Almaz), System administrator(Mr Natnael)
Use case description	Allow user to logout from the system.
Pre-conditions	None
Flow of events	<ol style="list-style-type: none"> <li>1. Alemu, Chala, Zinash, Mr Samuel, Ms Almaz or Mr Natnael clicks on logout button.</li> <li>2. The system ask user to confirm logout process.</li> <li>3. The system redirects user to the homepage and terminates user’s session.</li> </ol>

### 2.5.3 Object Model

#### 2.5.3.1. Data dictionary

A Data Dictionary is a collection of names, definitions, and attributes about data elements that are being used or captured in a database, information system, or part of a research project. It describes the meanings and purposes of data elements within the context of a project, and provides guidance on interpretation, accepted meanings and representation. A Data Dictionary also provides metadata about data elements. The metadata included in a Data Dictionary can assist in defining the scope and characteristics of data elements, as well the rules for their usage and application.

Table 2-21 User

Table Name	User			
Description	To Store the information of the member			
Fields	Data Type	Length	Description	Example
User_ID	Int	10	Unique identification	0012
User_Name	String	30	Login name	Famuser
User_Password	String	30	Login password	zero@123
User_phone	Int	20	Phone number of user	+251900631287
User_email	String	30	Email of user	Melkamu@gmail.com
User_age	Int	20	The age of the user	20
User_gender	String	10	The sex of user	male
User_Address	String	30	Address of user	Addsalem, ethoipia
Role	String	30	Restrictions and permissions of user	Farmer

Table 2-22 Newsfeed

<b>Table Name</b>	News			
<b>Description</b>	To post the latest news at the website's main page			
<b>Fields</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>	<b>Example</b>
Id	Int	10	Unique Identification for news	01
NF_type	string		The category of news	weather
NF_image	object		The image of news	-
NF_Date	Date	30	The date the news posted	1/01/2000
NF_description	String	Max	The content of the news	Agricultural input is now available!!!

Table 2-23 Product

<b>Table Name</b>	product			
<b>Description</b>	To store the information of the product			
<b>Fields</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>	<b>Example</b>
P_id	Int	10	Unique identification of product	01
P_name	String	20	The name of product	wheat
P_image	object		The image of product	url
P_Description	String	100	Additional information about the image	-

Table 2-24 Job

<b>Table Name</b>	Job			
<b>Description</b>	To store the information of the jobs available in the website			
<b>Fields</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>	<b>Example</b>
J_id	Int	10	Unique identification of jobs	01
J_type	String	20	The category of the job	harvesting
J_salary	Double	20	Salary the employ earned	5000
J_hr	String	20	The hours the employ invest on the work per day	8
J_duration	String	20	It tells the jobs last	5 month
J_description	String	100	It gives additional information	-

Table 2-25 Question

<b>Table Name</b>	Question			
<b>Description</b>	To store Question in the website			
<b>Fields</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>	<b>Example</b>
Question_id	Int	10	Unique identification of government policy	2341
Question_title	String	20	It is the title of the question/answer	crop
Date_created	date			
Q&A_details	String	30	It is about some notes on the question/answer	-

Table 2-26 Answer

<b>Table Name</b>	Answer			
<b>Description</b>	To store answer for the question in the website			
<b>Fields</b>	<b>Data Type</b>	<b>Length</b>	<b>Description</b>	<b>Example</b>
Answer_id	Int	10	Unique identification of government policy	2341
Answer_title	String	20	It is the title of the answer	crop
Answer_details	String	30	It is about some notes on the answer	-
Date_created	date	64	The date in which the answer posted	2/5/2023

### 2.5.3.1 Class diagram

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (methods) and the relationships among objects [4].

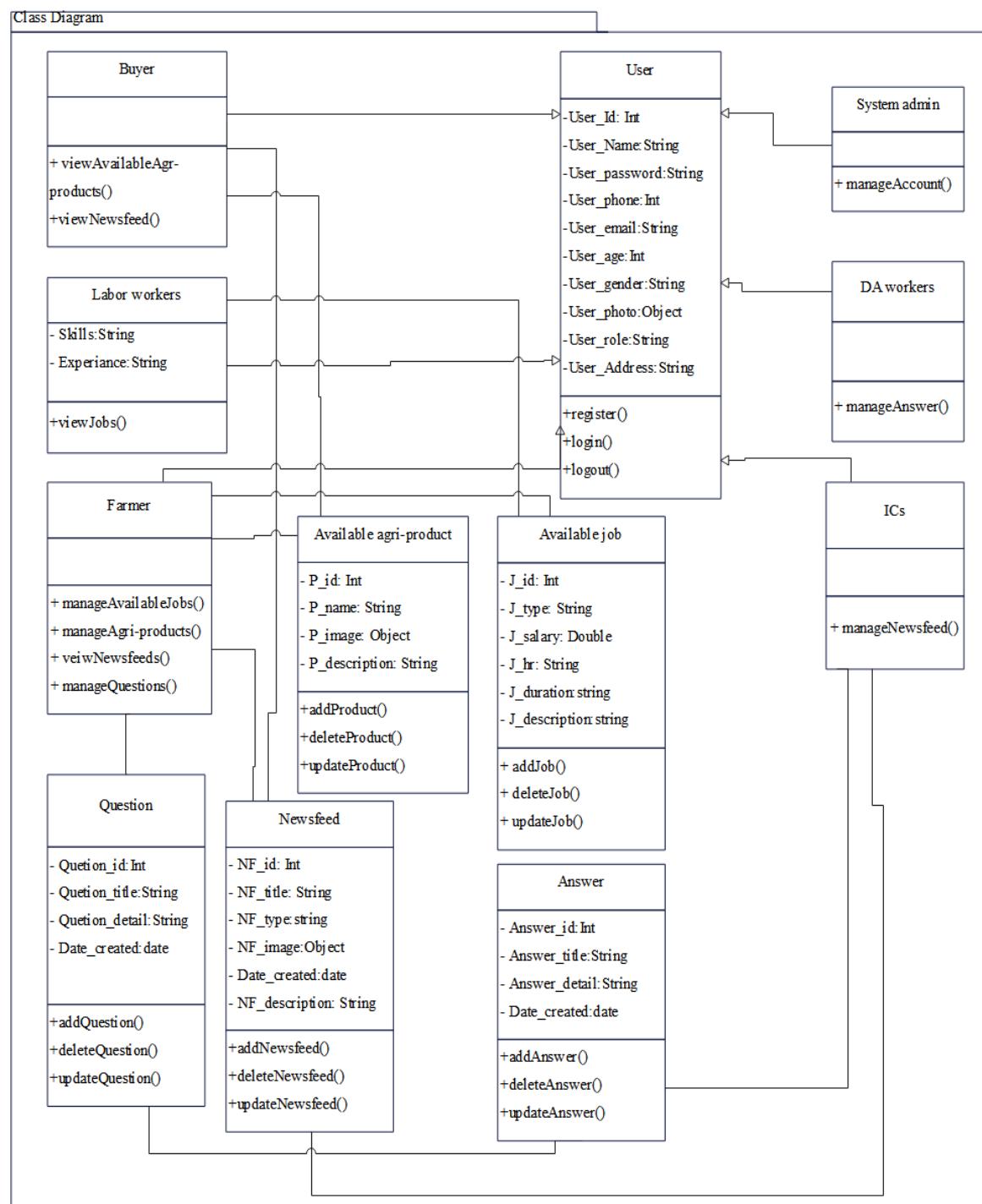


Figure 2-2 Class Diagram of the System

## 2.5.4 Dynamic Modelling

### 2.5.4.1 Sequence Diagram

A sequence diagram is an interaction diagram that shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagram are typically associated with use case realizations [4].

Accordingly, some selected to show sequence diagrams of our proposed system are as follows.

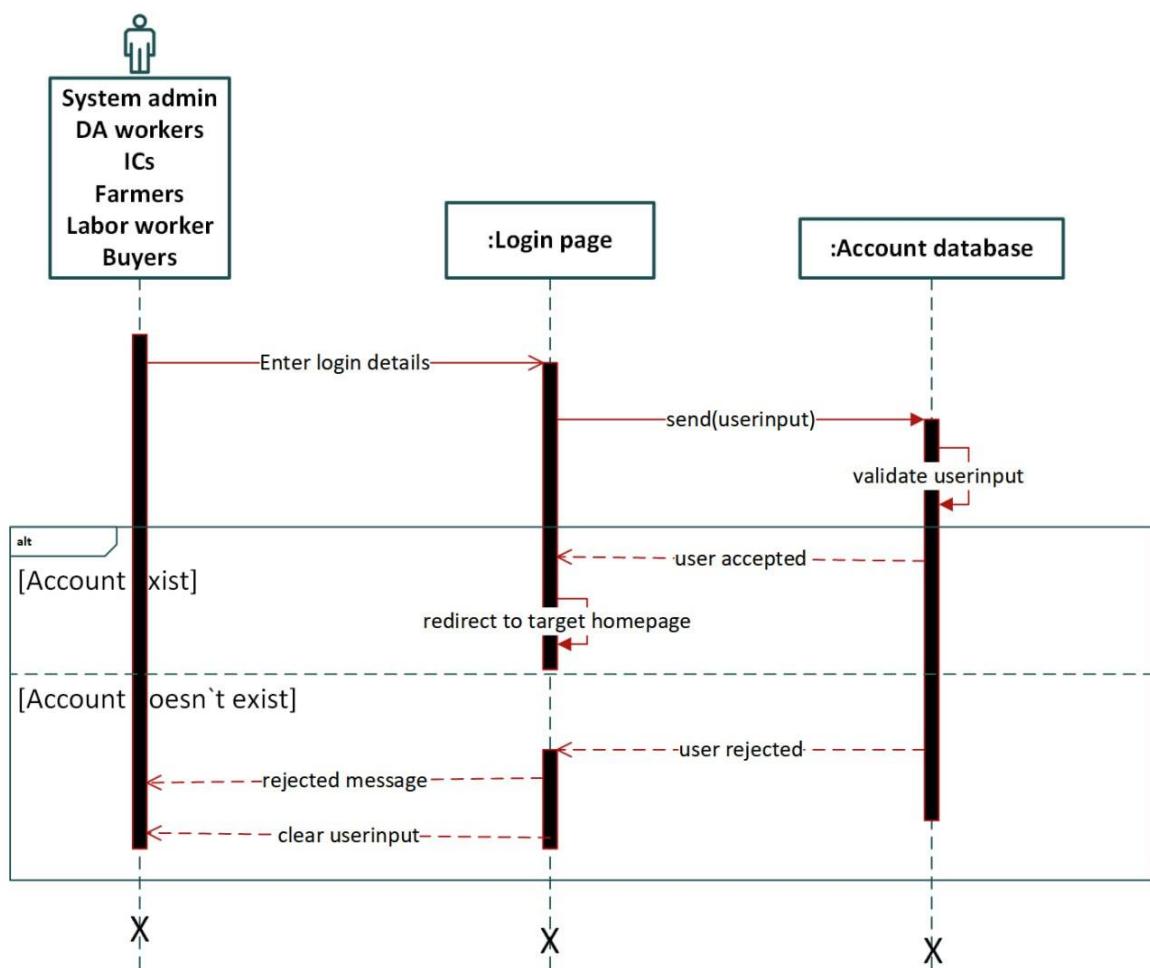


Figure 2-3 Login Sequence Diagram

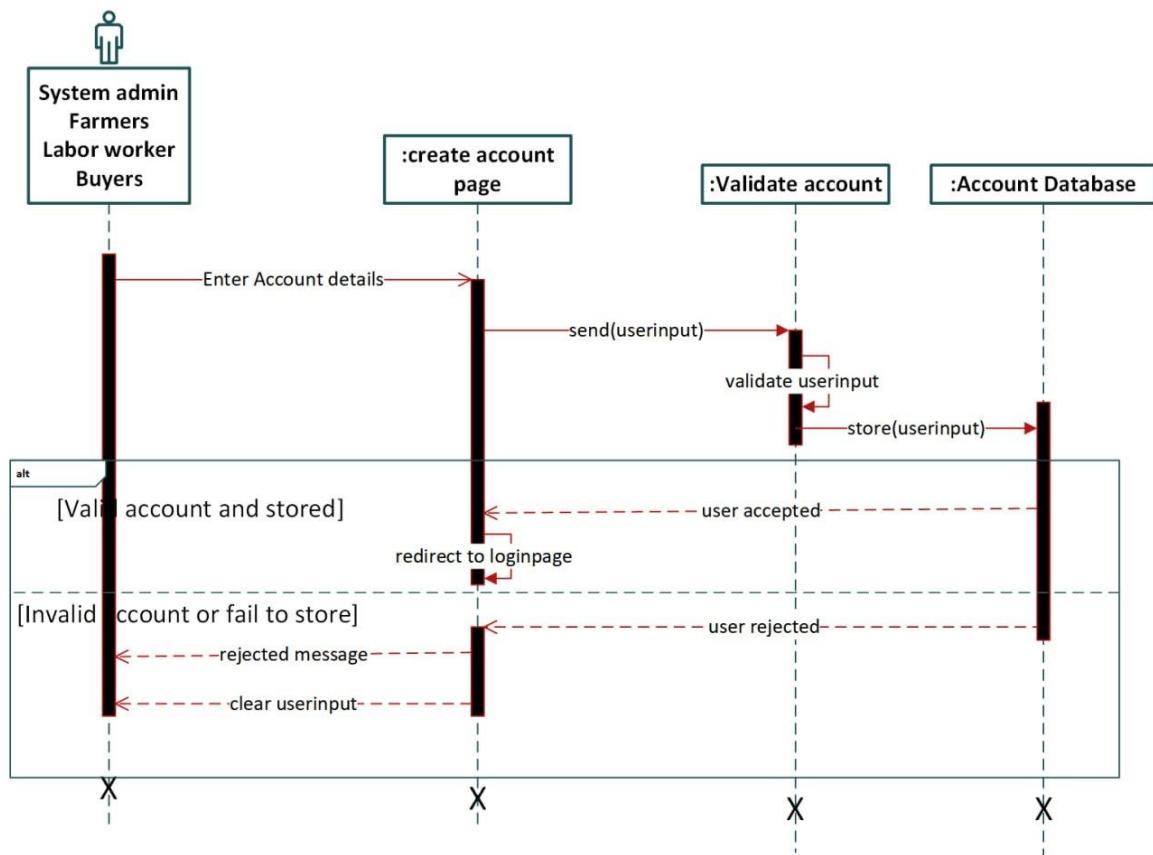


Figure 2-4 Create Account sequence diagram

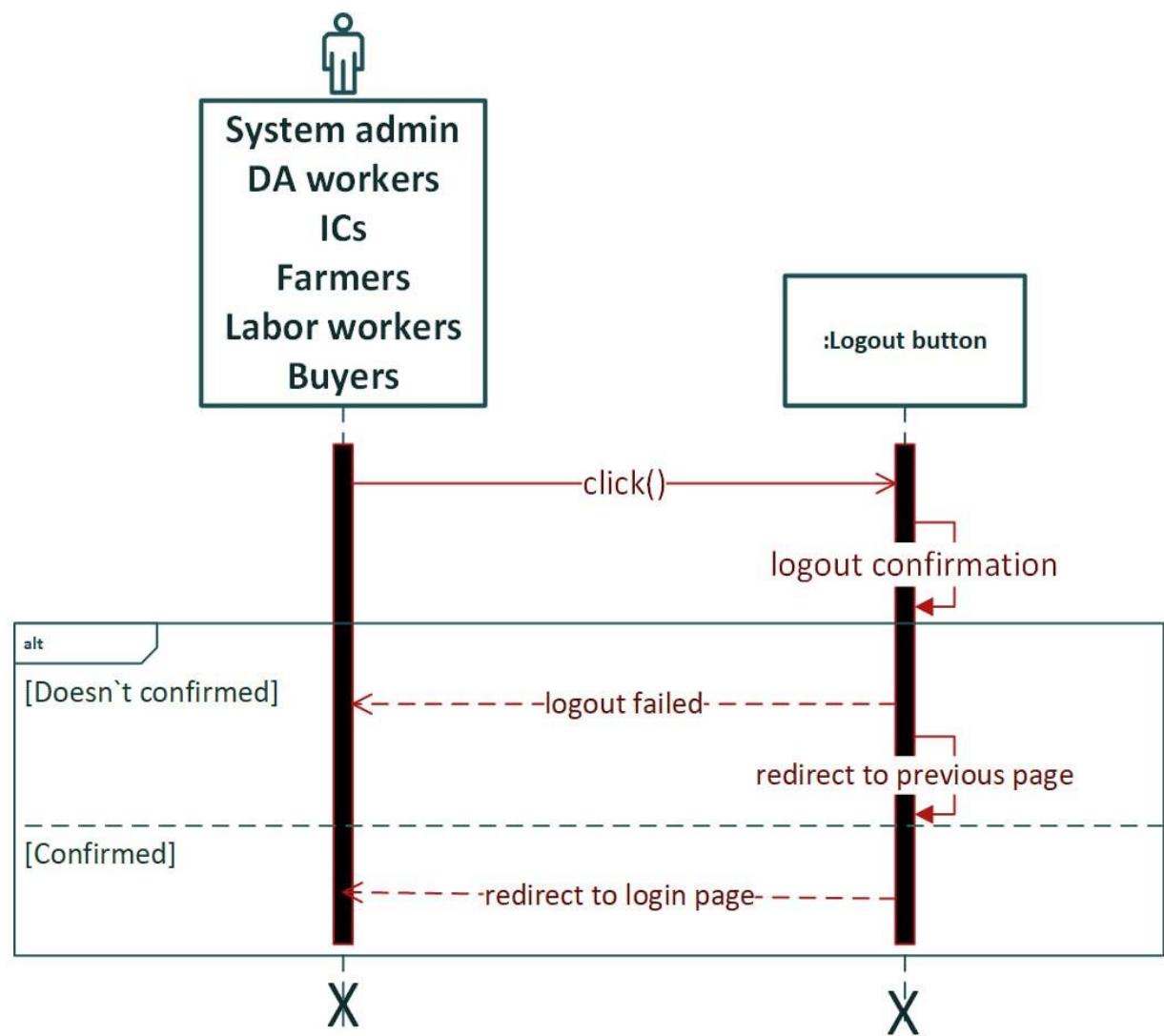


Figure 2-5 Logout sequence Diagram

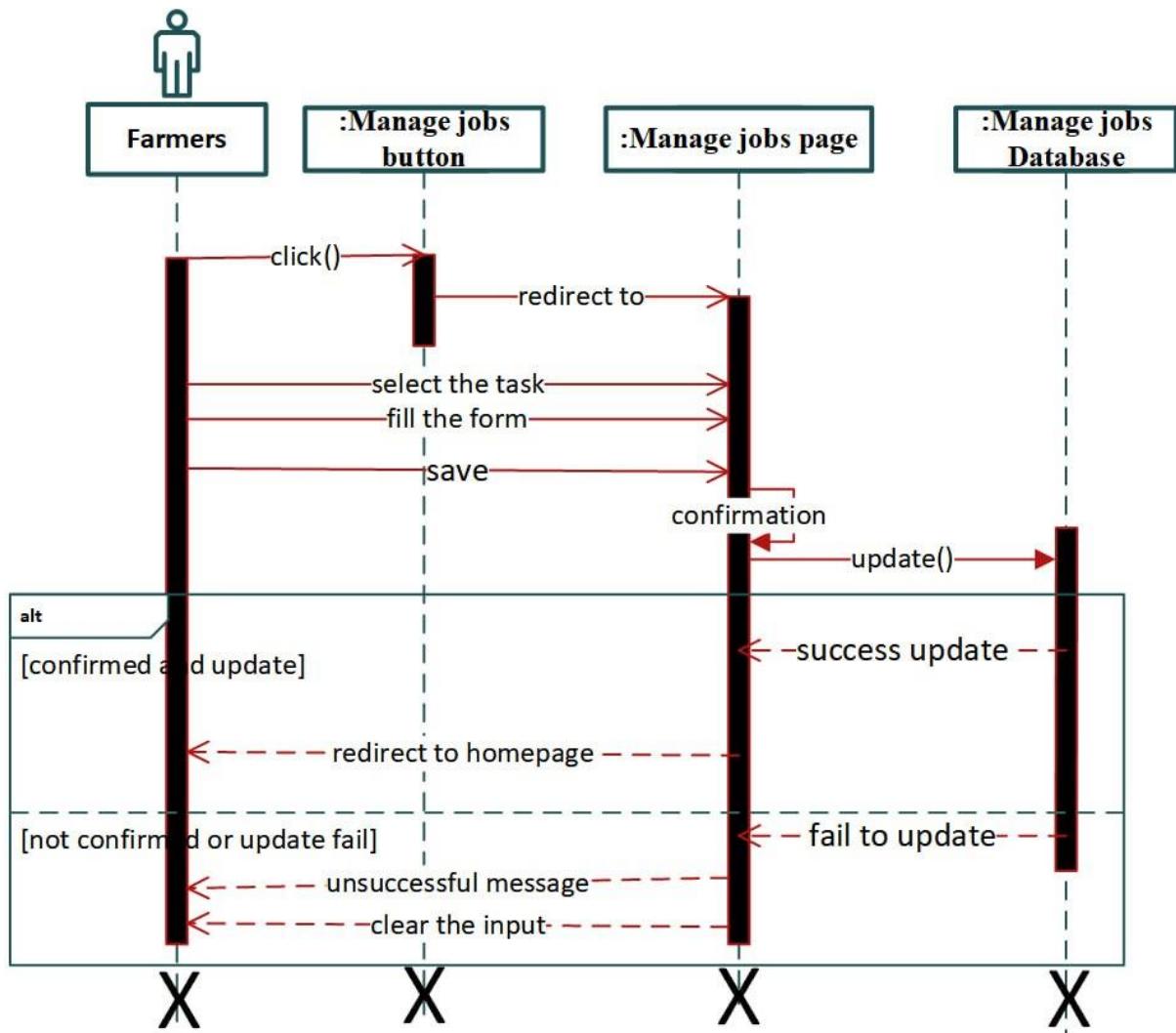


Figure 2-6 Manage Jobs Sequence Diagram

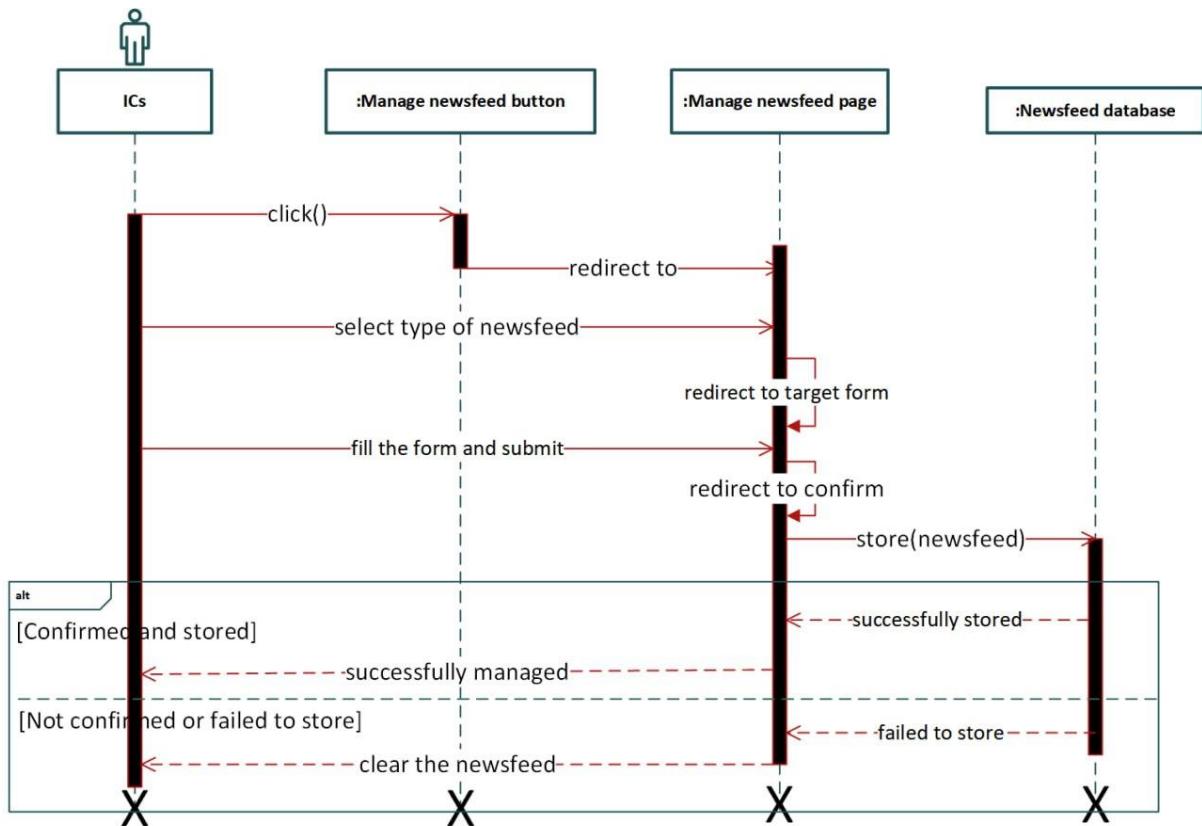


Figure 2-7 Manage Newsfeed Sequence Diagram

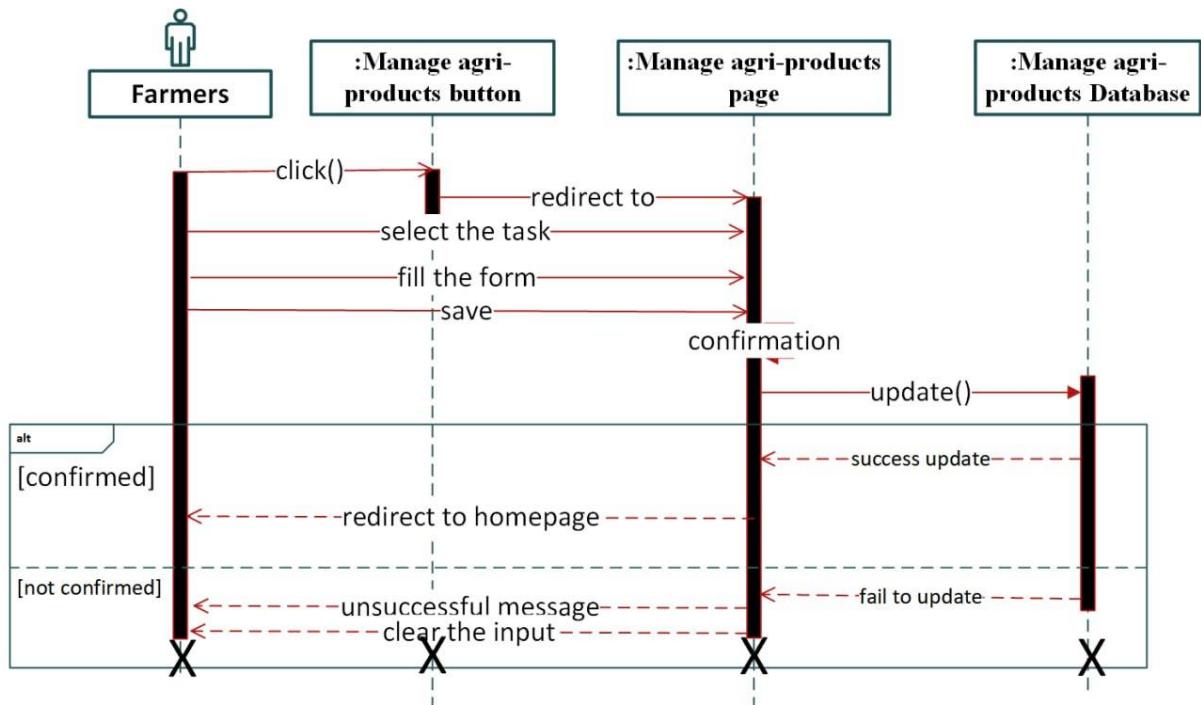


Figure 2-8 Manage Agri-product Sequence Diagram

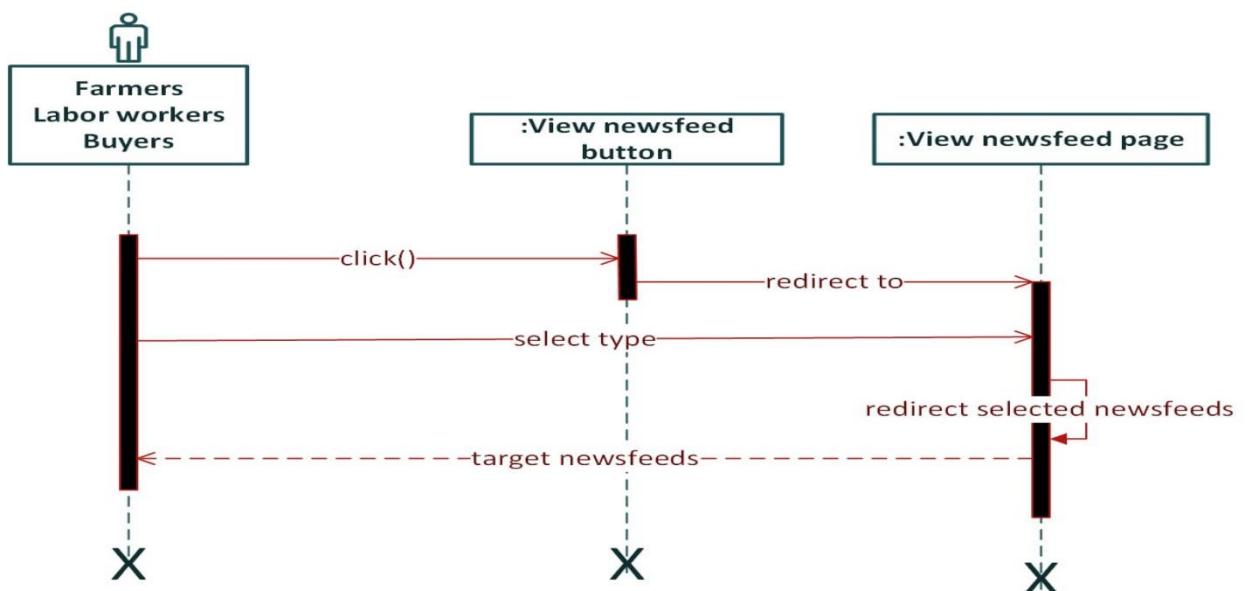


Figure 2-9 View Newsfeed Sequence Diagram

#### 2.5.4.2 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

Accordingly, some selected to show activity diagrams of our proposed system are as follows.

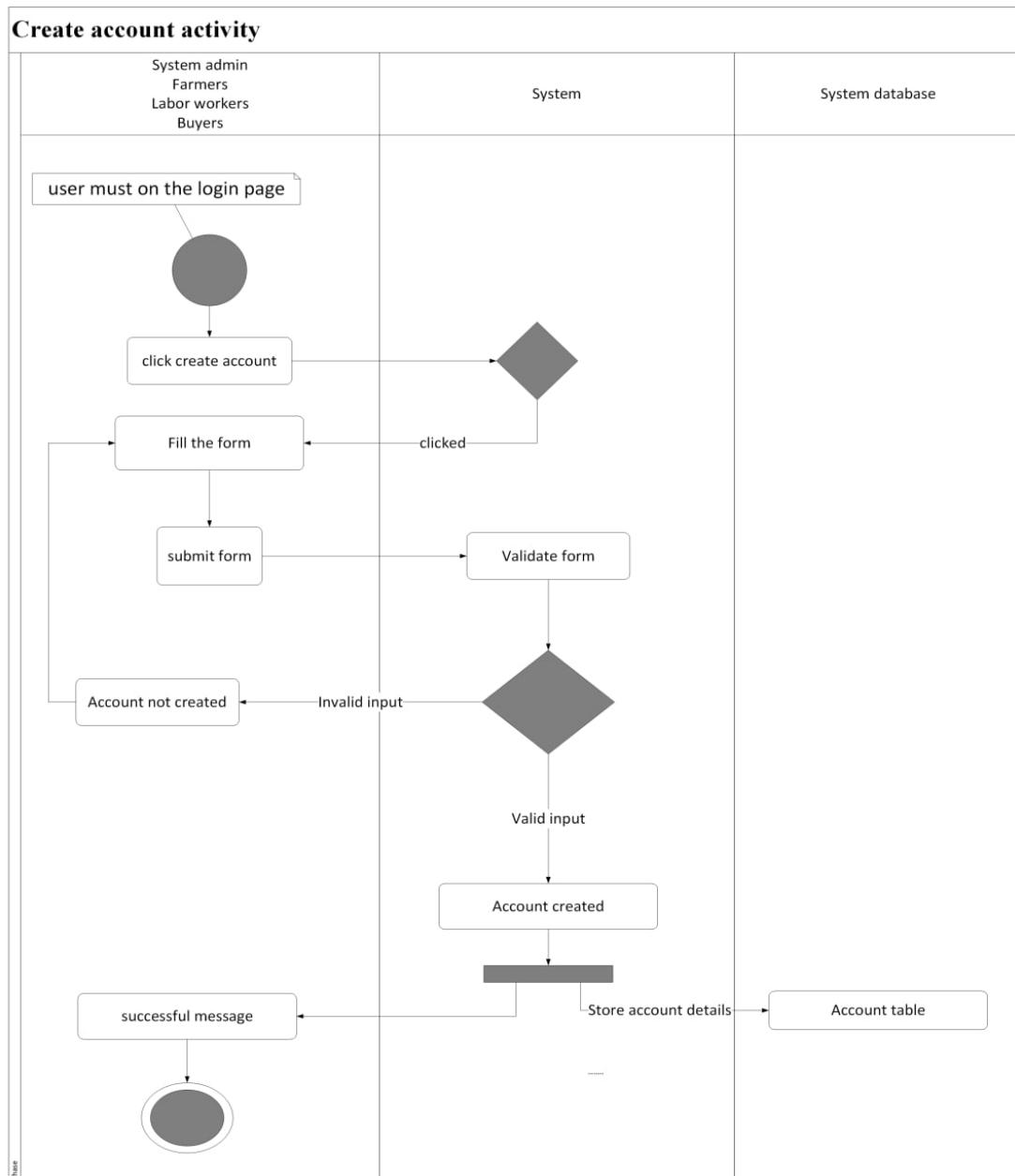


Figure 2-10 Create Account Activity Diagram

## Login activity

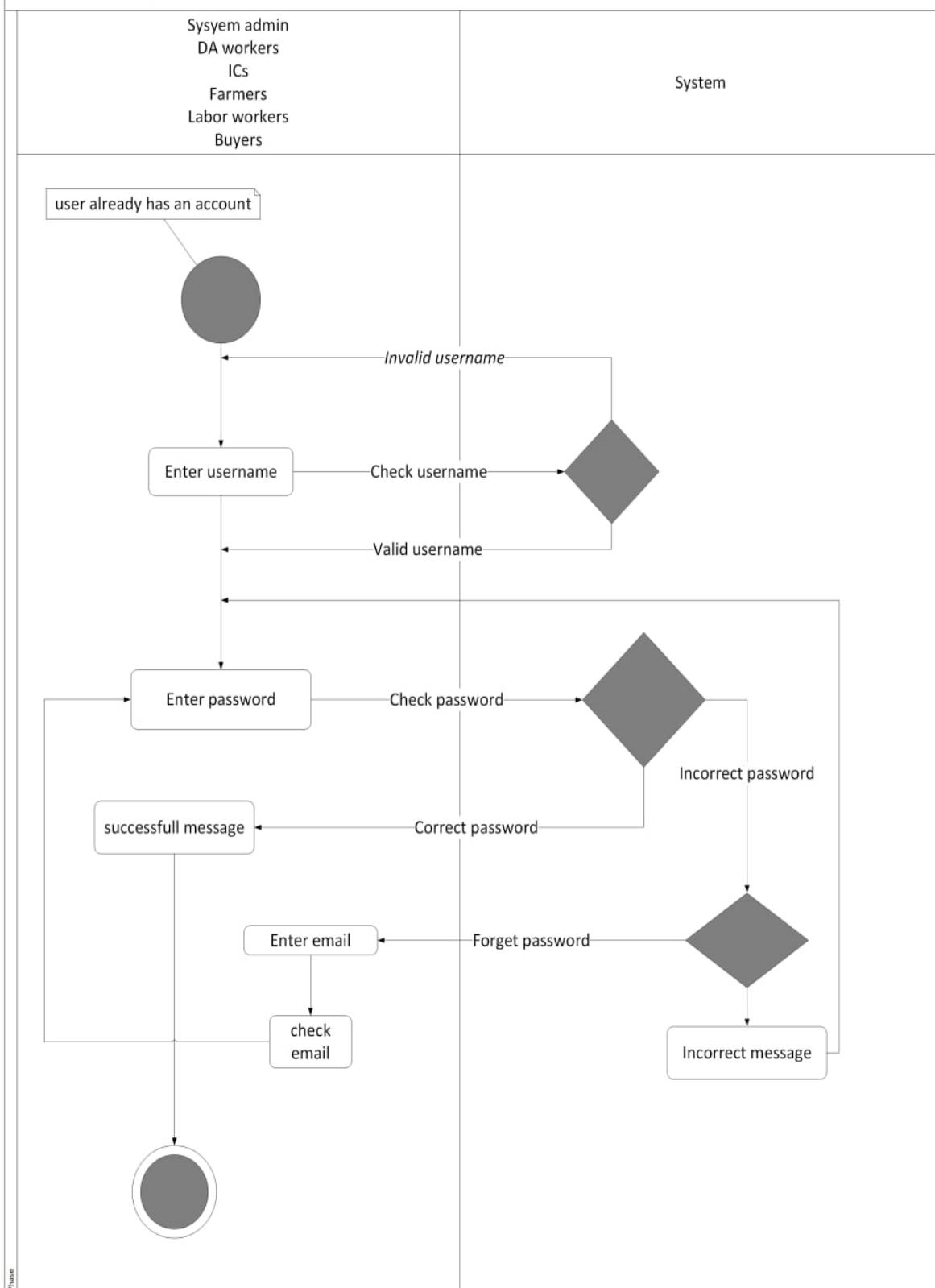


Figure 2-11 Login Activity Diagram

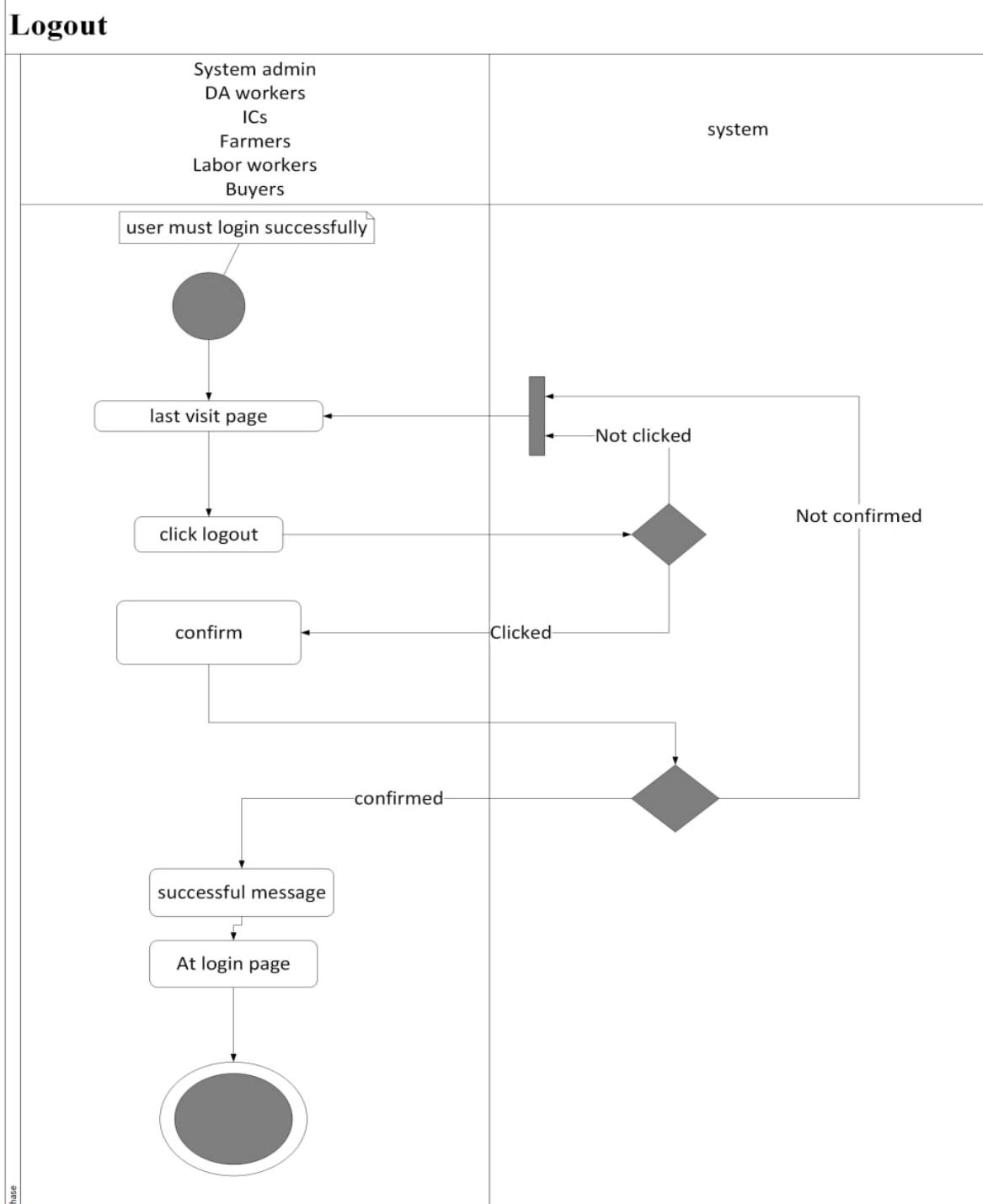


Figure 2-12 Logout Activity Diagram

## Manage account of DA workers and ICs

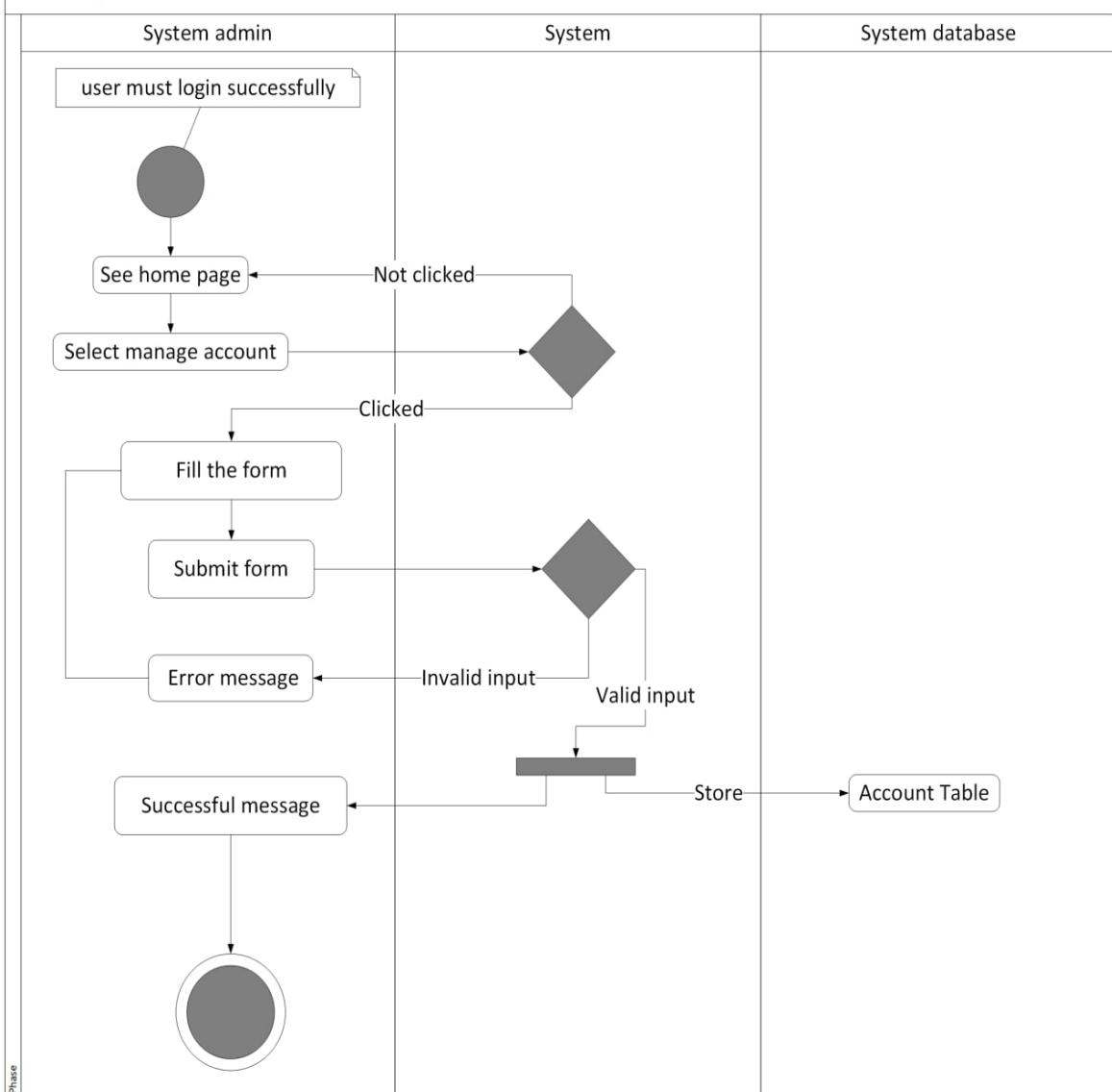


Figure 2-13 Manage DA workers and IC's account Activity Diagram

## Manage agri-products

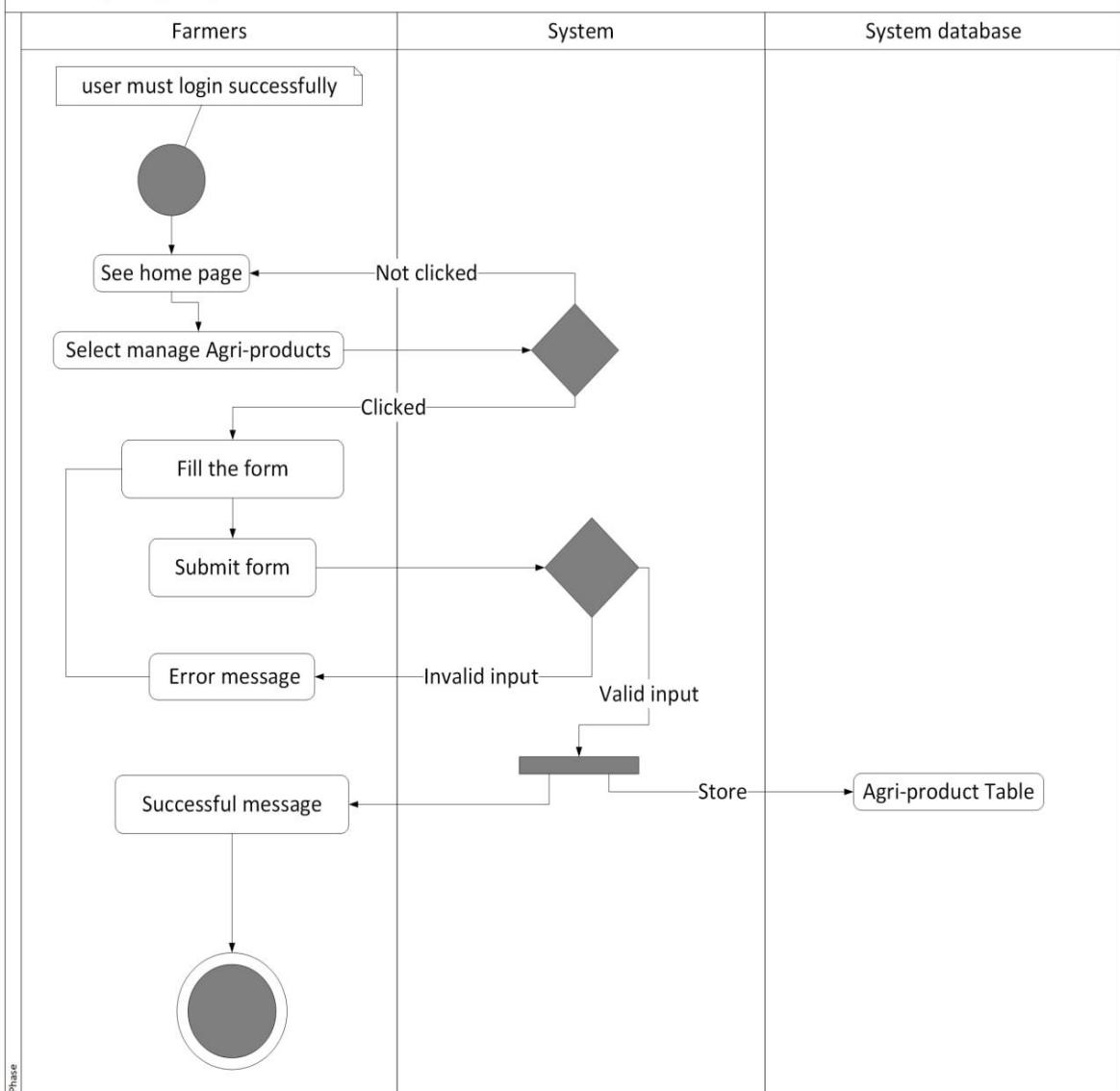


Figure 2-14 Manage Agri-products Activity Diagram

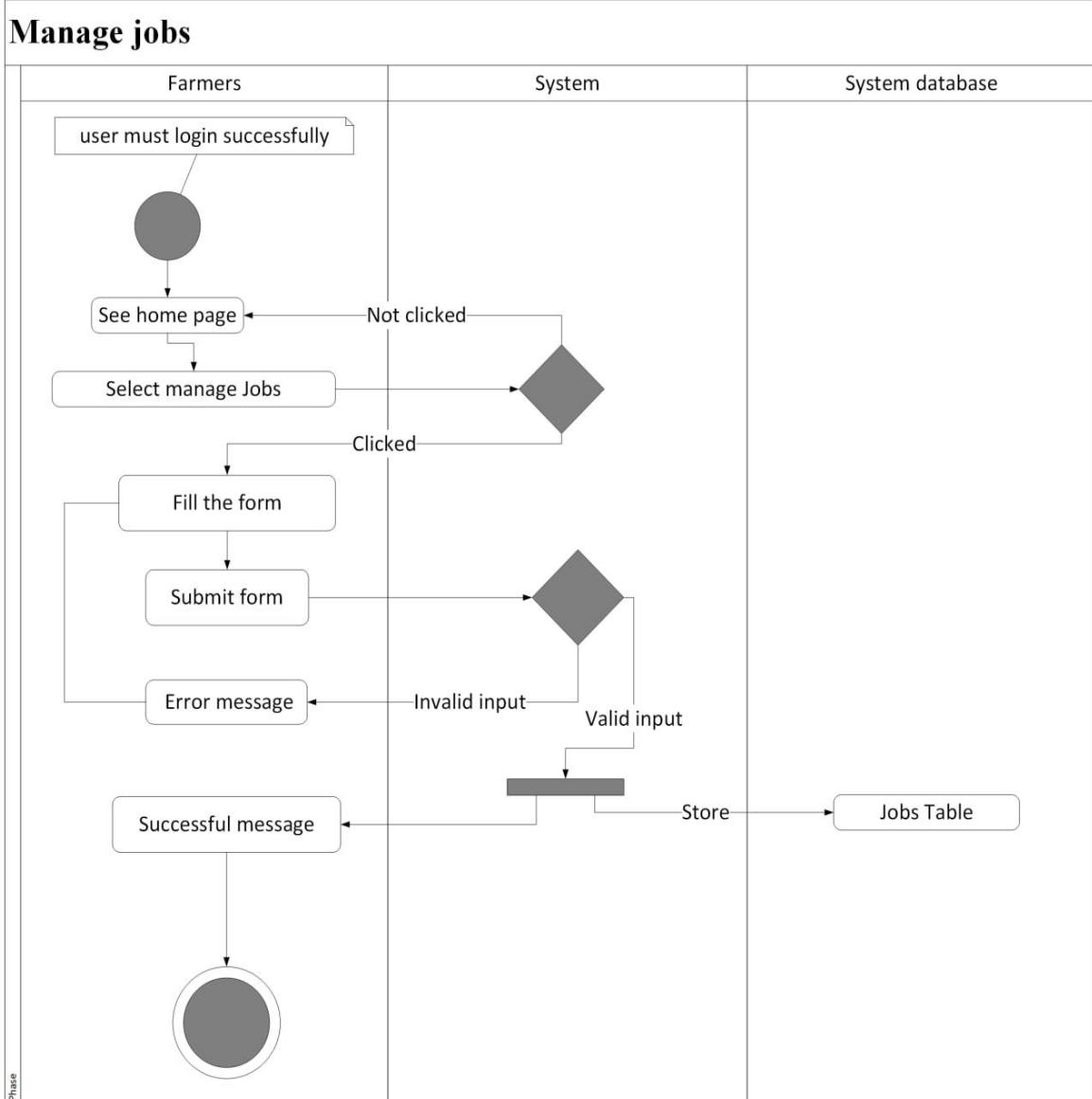


Figure 2-15 Manage Jobs Activity Diagram

## Manage Newsfeeds

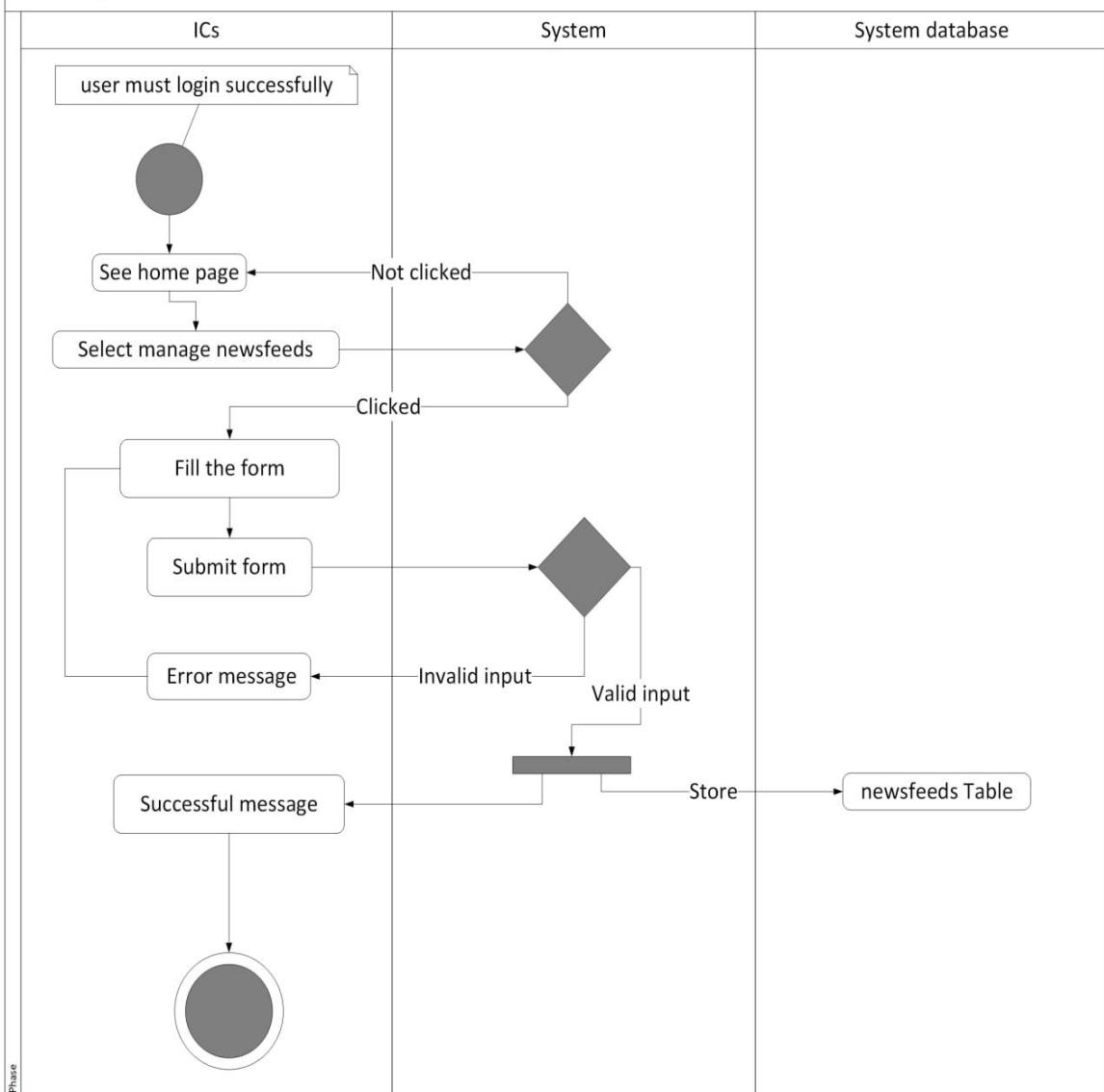


Figure 2-16 Manage Newsfeed Activity Diagram

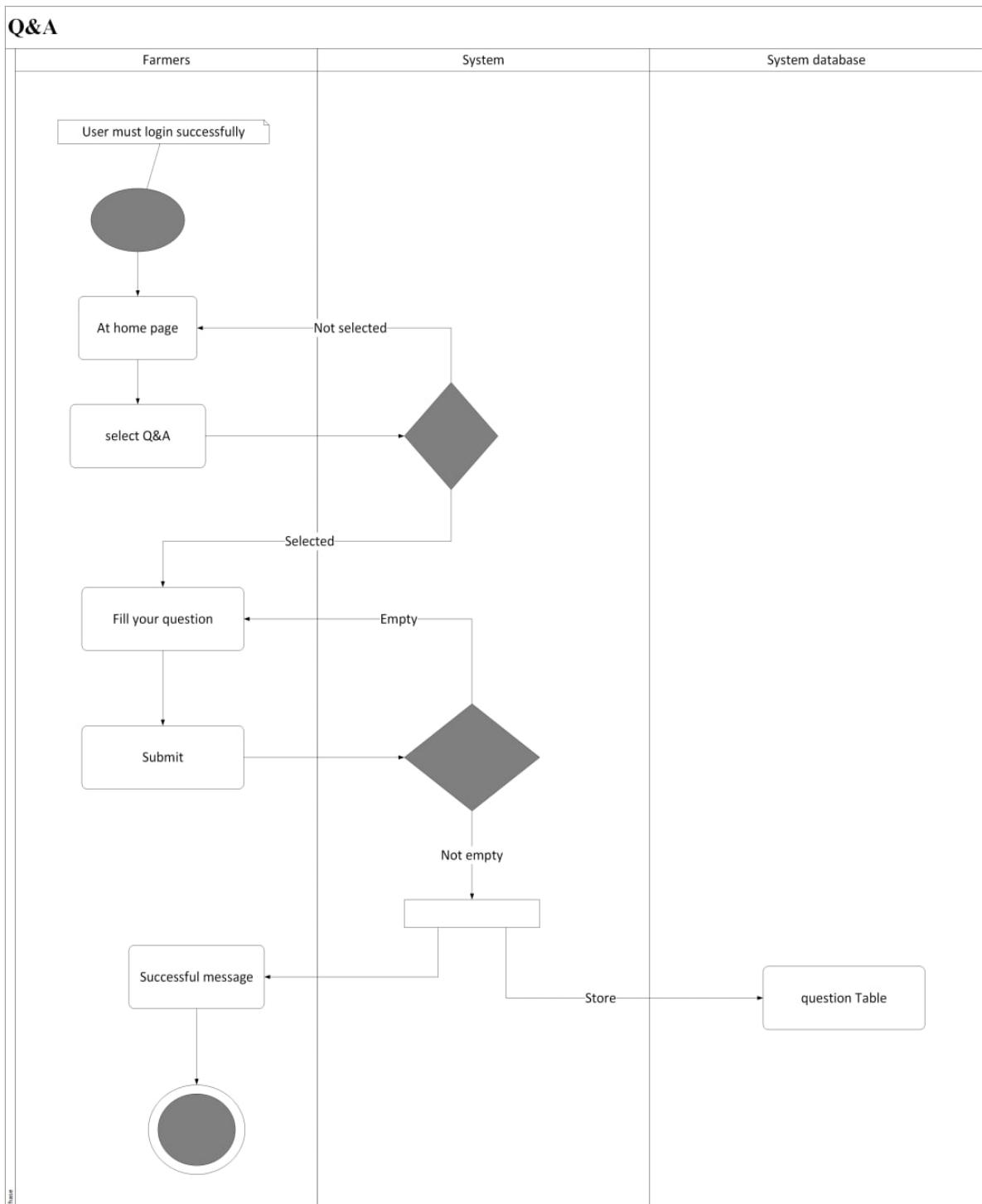


Figure 2-17 Q&A Activity Diagram

### 2.5.5 User Interface

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website. Figure 2.17 - Figure 2-20 are used to illustrate the user interface of the system.

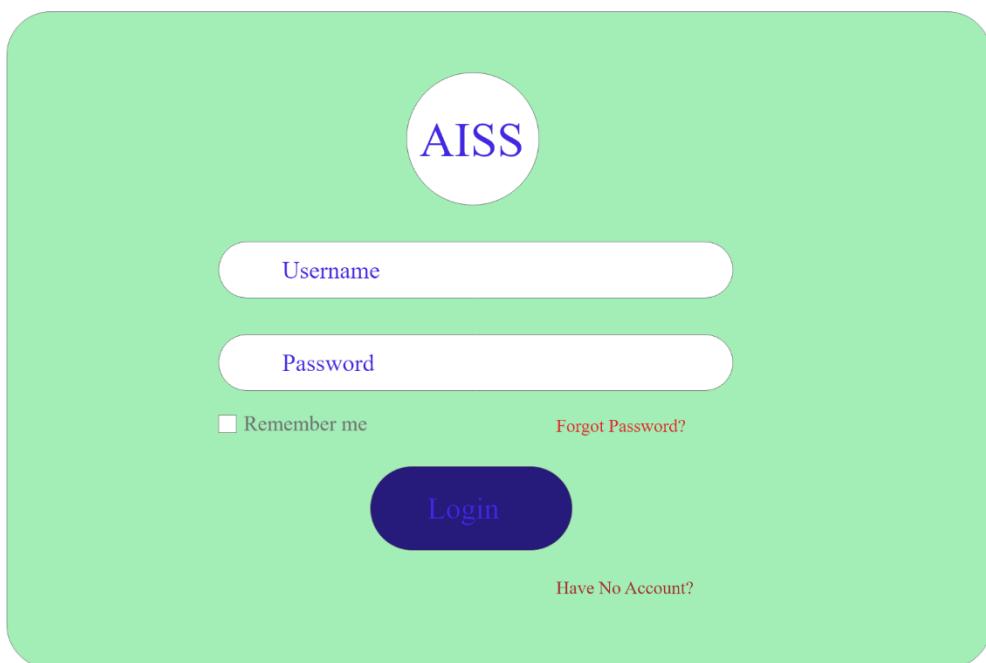


Figure 2-18 Login User interface



Figure 2-19 Homepage User Interface

**AISS**

- [Crop](#)
- [Seasonal Information](#)
- [New Agri-Tech Information](#)
- [Government Policy](#)
- [Price Index](#)
- [Other](#)

IC username ✓

**+ Add New**

**Search Here.....**

**Title of post**

Lore Ipsum is simply dummy text of the printing and typesetting industry. Lore Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lore Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lore Ipsum. [Read More..](#)

**Edit**

**Delete**

**Title of post**

Lore Ipsum is simply dummy text of the printing and typesetting industry. Lore Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lore Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lore Ipsum. [Read More..](#)

**Edit**

**Delete**

**Explore More...**

**AISS** AISS

AISS(Agricultural Information Sharing System) is a platform that provides necessary information to farmers and consumers.

**Contact Us**

Username

Email

**Send**

**Links**

- [Home](#)
- [Newsfeed](#)
- [About us](#)
- [Contact Us](#)
- [Price Index](#)
- [Q&A](#)

AISS©copyright 2023. All Rights Reserved.

Figure 2-20 Manage Newsfeed Page User Interface

AISS
[Home](#)
[Newsfeed](#)
[Q&A](#)
[Management panel](#)
[Price Index](#)
[About Us](#)
username v

**Filters**

**Date**

**Type**

**Sort**

### Newsfeed(AI)

**Title of the post**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. [Read More..](#)

[Comment](#)
[Ask?](#)

**Title of the post**

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum. [Read More..](#)

[Comment](#)
[Ask?](#)

[Read More...](#)

AISS
AISS
Links

AISS(Agricultural Information Sharing System) is a platform that provides necessary information to farmers and consumers.

Contact Us

Home

Newsfeed

About us

Contact Us

Price Index

Q&A

AISS(C)opyright 2023. All Rights Reserved.

Figure 2-21 View Newsfeed Page User Interface

60

## 3 CHAPTER 3 SYSTEM DESIGN

### 3.1 Introduction

The system design will include the overall view of the system from the functional viewpoint. It is the process of defining the components, modules, interfaces, and data for a system to satisfy specified requirements.

System development is the process of creating or altering systems, along with the process, practices, models, and methodologies used to develop the system.

This section provides a clear description of the “Agricultural Information Sharing System” overall design in an organized way.

### 3.2 Design Goals

The design goals of the system are derived mainly from the non-functional requirement. It helps us in identifying the qualities that the system is optimized to satisfy. The following four major quality perspectives are used to identify the design goals of the system.

#### Performance

It describes the speed and space requirement that expected from the system.

- **Response time:** As the system is a web-based, responses to remote user request will be optimized.
- **Throughput:** As the system will be accessed by different users, so it is required to accomplish several users' request.

#### Dependability

It refers to the design goals set to minimize system crushes and their consequences. The following are identified as the main dependability criteria.

- **Reliability:** The system must be reliable by using Operating System provided backup systems.
- **Robustness:** The system should have the ability to withstand invalid user input by validating all the inputs of the user.
- **Availability:** The system should be available to everyone who has smart phone wherever internet connection is available.

- **Fault tolerance:** The system should be fault tolerant by displaying warning messages to the user even when erroneous data is inserted.
- **Security:** The system should have a secure authentication mechanism so as to prevent unauthorized users from accessing non-allowed resources.

## Maintainability

It determines behavior of the system to maintain after deployment when new functionalities will be observed.

- **Extensibility:** New functionalities and classes can easily be added.
- **Modifiability:** The system should be easily modifiable after deployment without affecting the current working system.

## End User Criteria

- **Usability:** The system should be a user-friendly system so that users can easily access and use it.

## 3.3 Proposed Software Architecture

### 3.3.1 Overview

As the application is a web-based application, it will be accessed through a web browser. To implement this online system, the system design proposed to implement Three-Tier client/server architecture, with a database. In the client/server architecture, a subsystem, the server, provides services to instances of other subsystems called the clients, which are responsible for interacting with the user.

Web based applications are accessed by web browser over a network of client/server architecture. Users will use their browser software to communicate with the system. Users will interact with the database system through the graphical user interface. All data entries that are performed on the client PC are transferred to the web server via an appropriate network connection.

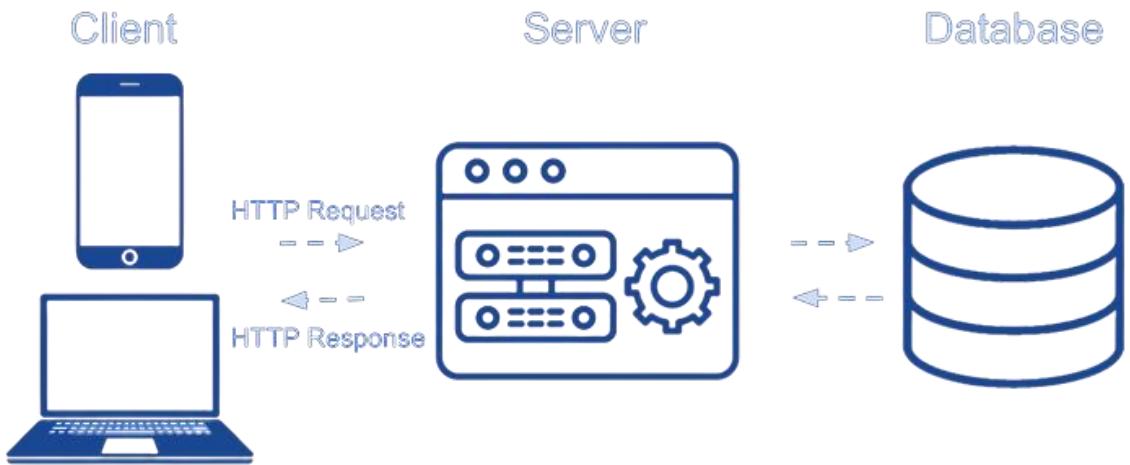


Figure 3-1 System Architecture , Tanenbaum and Van Steen (2017)

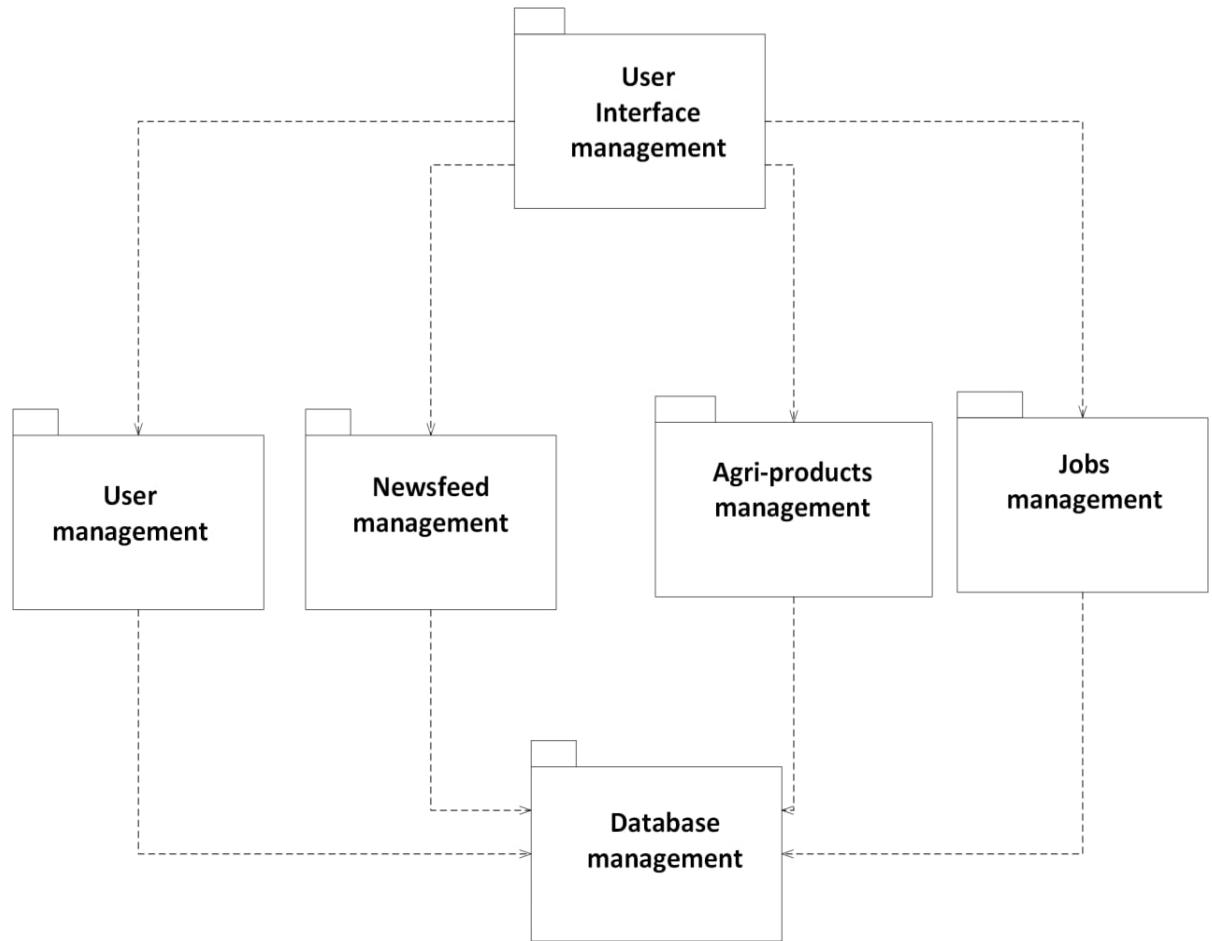
### 3.3.2 Subsystem Decomposition

It describes the division of the system into subsystems that are a collection of classes, associations, operations, events, and constraints that are closely interrelated with each other and the responsibilities of each subsystem.

The “Agriculture Information Sharing System” has the following subsystems.

- User interface subsystem
- User management subsystem
- Information management subsystem
- Agri-products management subsystem
- Jobs management subsystem
- Data management subsystem

The system decomposition and the interaction among the subsystems are shown in figure 3-2.



**Figure 3-2 System Decomposition**

### A. User Interface Subsystem

This subsystem allows interaction with the other subsystems, that is it causes interaction of end user with system.

The interface subsystem includes the following interfaces:

- **User management subsystem interface:** This interface enables users of the system to interact with the user management subsystem.
- **Information management subsystem interface:** This interface enables the authorized user to interact with the information management subsystem.
- **Agri-products management subsystem interface:** This interface enables the authorized user to interact with the agri-products management subsystem.

- **Jobs management subsystem interface:** This interface enables the authorized user to interact with the jobs management subsystem.
- **Data management subsystem interface:** This interface allows interfacing with the data management subsystem.

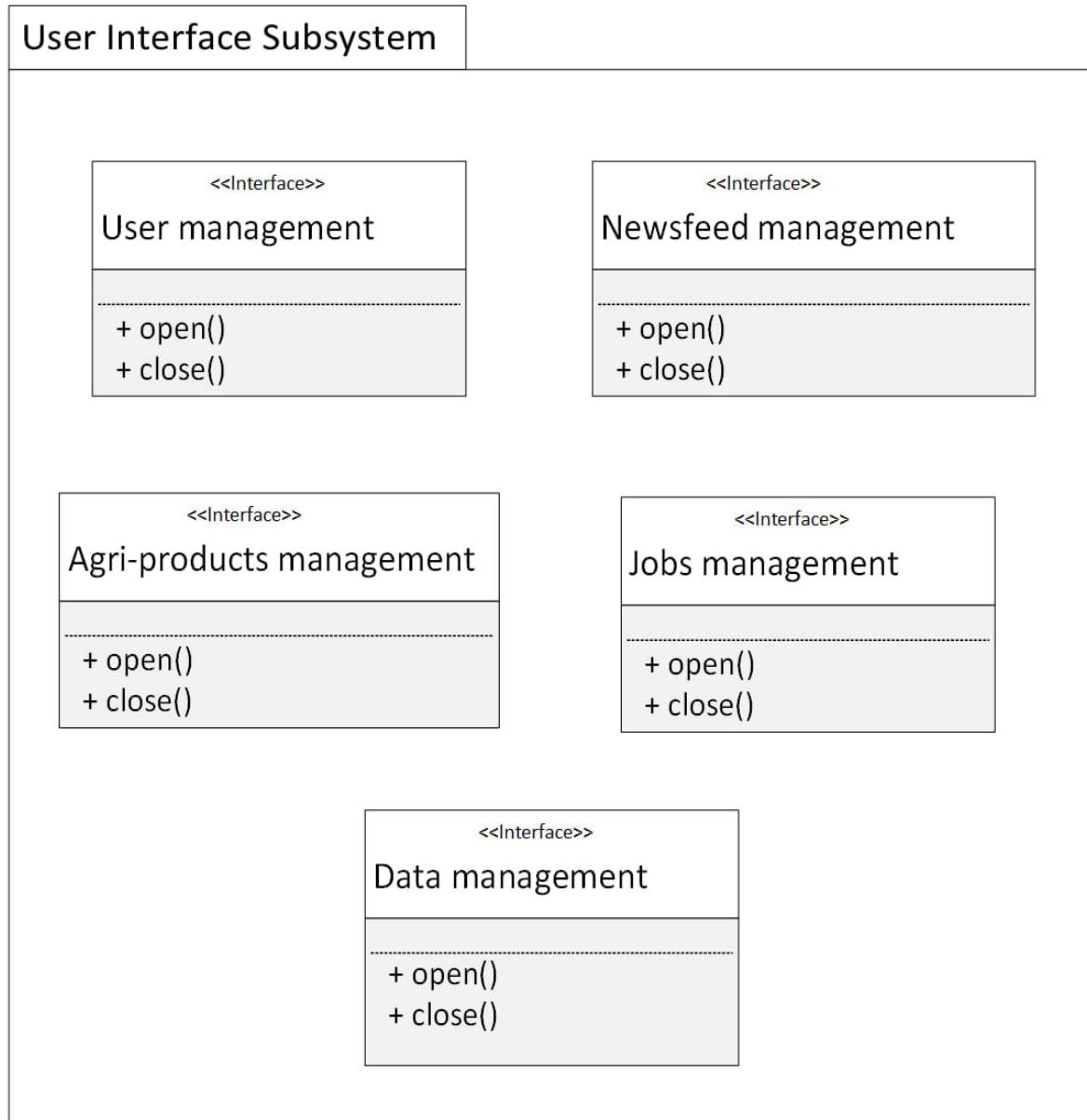


Figure 3-3 User Interface Subsystem

## B. User Management Subsystem

This subsystem is responsible for user login and maintenance of user information. It includes user account creation and edit, update, delete and interaction of users.

It includes the following classes.

- **System Administrator:** This class is responsible for creating, updating and deleting or terminating DA workers and ICs account.
- **DA workers:** This class is responsible for viewing newsfeeds, answering to user questions, approving report change request from region user.
- **Information Clerks:** This class is responsible for posting, editing and deleting seasonal weather information, crop information, new technology, government schemas and visualization of price index.
- **Farmers:** This class is responsible for viewing newsfeeds, ask questions and posting, editing and deleting available jobs and agri-products.
- **Labour workers:** This class is responsible for viewing newsfeeds, contact to farmers who needs labour worker.
- **Buyers:** This class is responsible for viewing newsfeeds, contact to farmers who has products for sell.
- **Account:** This class is responsible for login and logout all users' accounts.

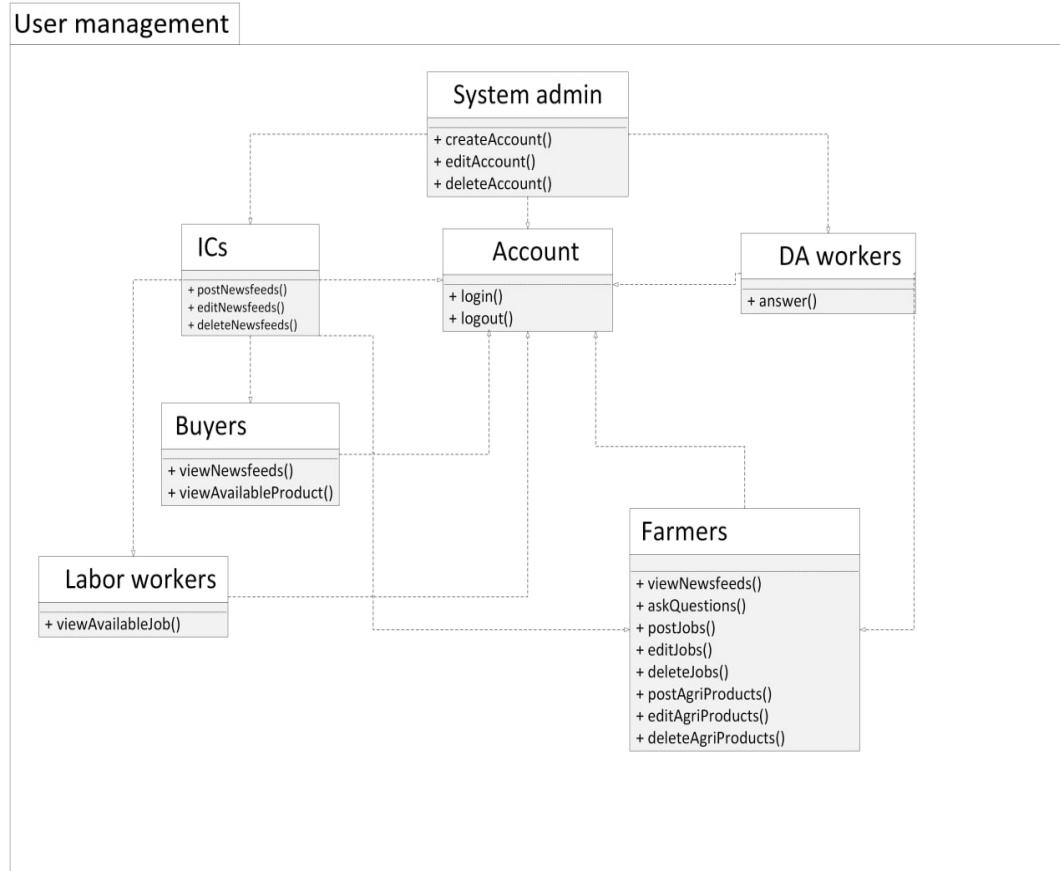


Figure 3-4 User Management Subsystem

### C. Newsfeed Management Subsystem

This subsystem is responsible for the management of newsfeeds including seasonal information, crop information, new technology, government schema and price index.

It contains the following classes.

- **Weather information:** This class is responsible for generating weather information.
- **Crop information:** This class is responsible for getting strain-crops detail information.
- **New agricultural technology information:** This class is responsible for getting new technologies that help the agricultural sector.
- **Government policy:** This class is responsible for getting government policies and directives.

- **Market information:** This class is responsible for generating a graphical price index visualization of crops trend on the market.

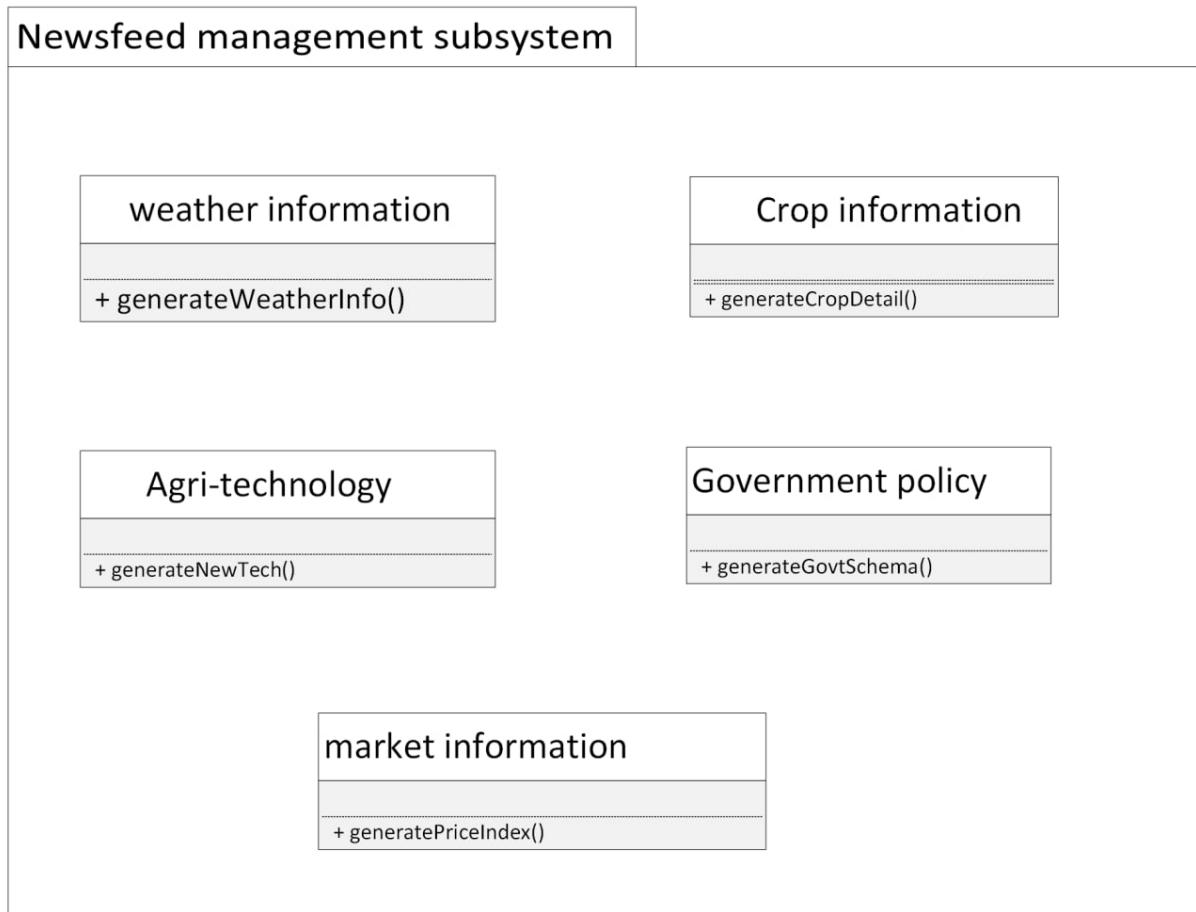


Figure 3-5 Newsfeed Management Subsystem

#### D. Agri-products management subsystem

This subsystem is responsible for agri-products management system tasks like add, update and delete agricultural products.

The Agri-products management subsystem include the following classes:

- **Agri-product:** This is responsible for adding, editing and deleting agricultural products that the farmers wants to sell.

## Agri-products management system

### Agri-products

+ addProduct()  
+ editProduct()  
+ deleteProduct()

Figure 3-6 Agri-products Management Subsystem

### E. Jobs management subsystem

This subsystem is responsible for jobs management system tasks like add, update and delete available jobs.

The Agri-products management subsystem include the following classes:

- **Jobs:** This is responsible for adding, editing and deleting available jobs that the farmers wants to hire.

### Jobs management subsystem

### Jobs

+ addJob()  
+ editJob()  
+ deleteJob()

Figure 3-7 Jobs Management Subsystem

## F. Data Management Subsystem

This subsystem is responsible for data management tasks like storage, update, and retrieval requests triggered by the subsystem.

The data management subsystem includes the following classes.

- Data Storage: This class is responsible to handle persistent data storage activities.
- Data Access: This is responsible to retrieve any information from database.
- System Message: This class is responsible to display error or success message related to data entry and data retrieval.

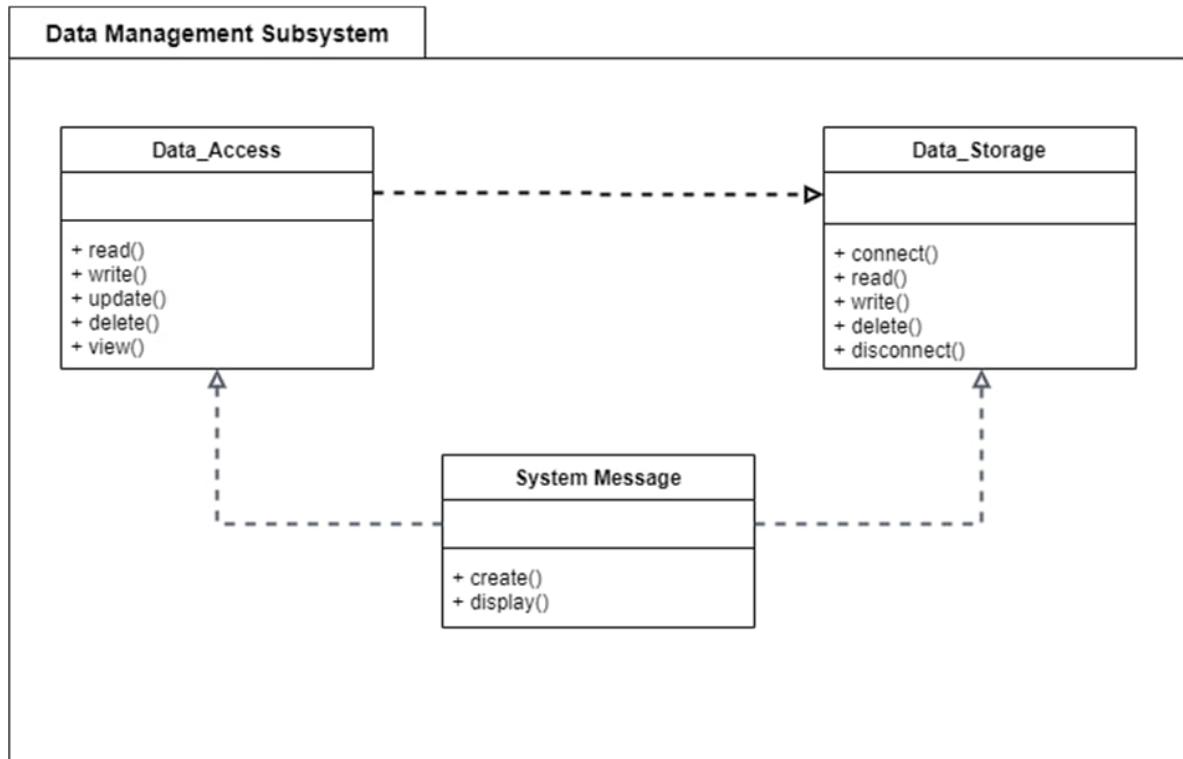


Figure 3-8 Data Management Subsystem

### 3.3.3 Hardware/Software Mapping

It indicates the various hardware devices and equipment used in the system, its interaction with the software components and execution environment, and artifacts deployed on the hardware.

Since the system is web-bases system, which means it is a client-server software, the following are devices that constitute the hardware components.

- **Application Server:** This server holds the subsystems of the system except the data management subsystem and user interface subsystem.
- **Database Server:** This server holds the data management subsystem.
- **Client Machines:** Client machine is a terminal from which the user will access the system.

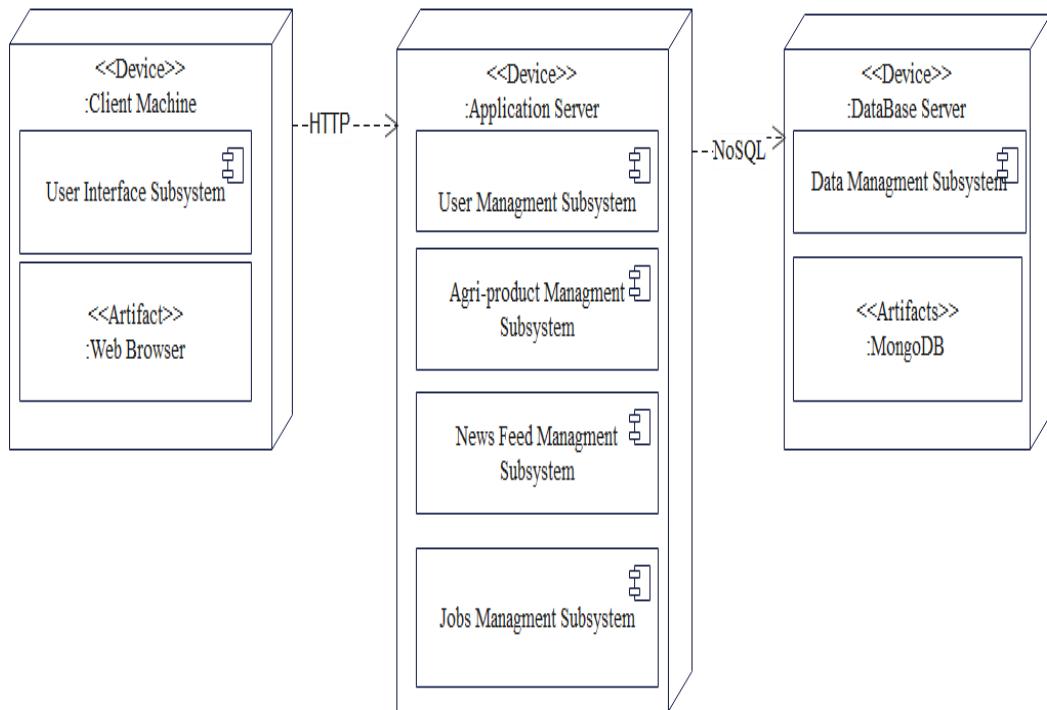


Figure 3-9 Deployment Diagram of the system

### **3.3.4 Persistent Data Management**

Persistent data management describes the persistent data stored by the system and the data management infrastructure required for it. The system uses the MongoDB database, a document-oriented database management system classified as a NoSQL database. It uses JSON-like documents. It provides an extensible, highly scalable, cost-effective way to store data.

The following are collection in the database:

1. User: it stores all the data of the user that includes user id, user name, user password, user gender, user age, user gender, user photo, user role user email, user phone, user address.
2. Agri-products: this collection contains all the data that include product id, user Id, product name, Product image, Product description.
3. Jobs collection contains document like job id, User Id, Job type, Job salary, Job hour, Job duration, Job description.
4. Newsfeed: it used to store information of crop, government policy, weather, agricultural technology and also market information. It has the following attributes News feed id, user Id, Newsfeed type, Newsfeed title, Newsfeed image, Newsfeed description.
5. Question: stores Question id, user Id, Question title, Question detail and also date of question created.
6. Answer: stores Answer id, Question id, user Id, Answer title, Answer detail and also date of Answer created.

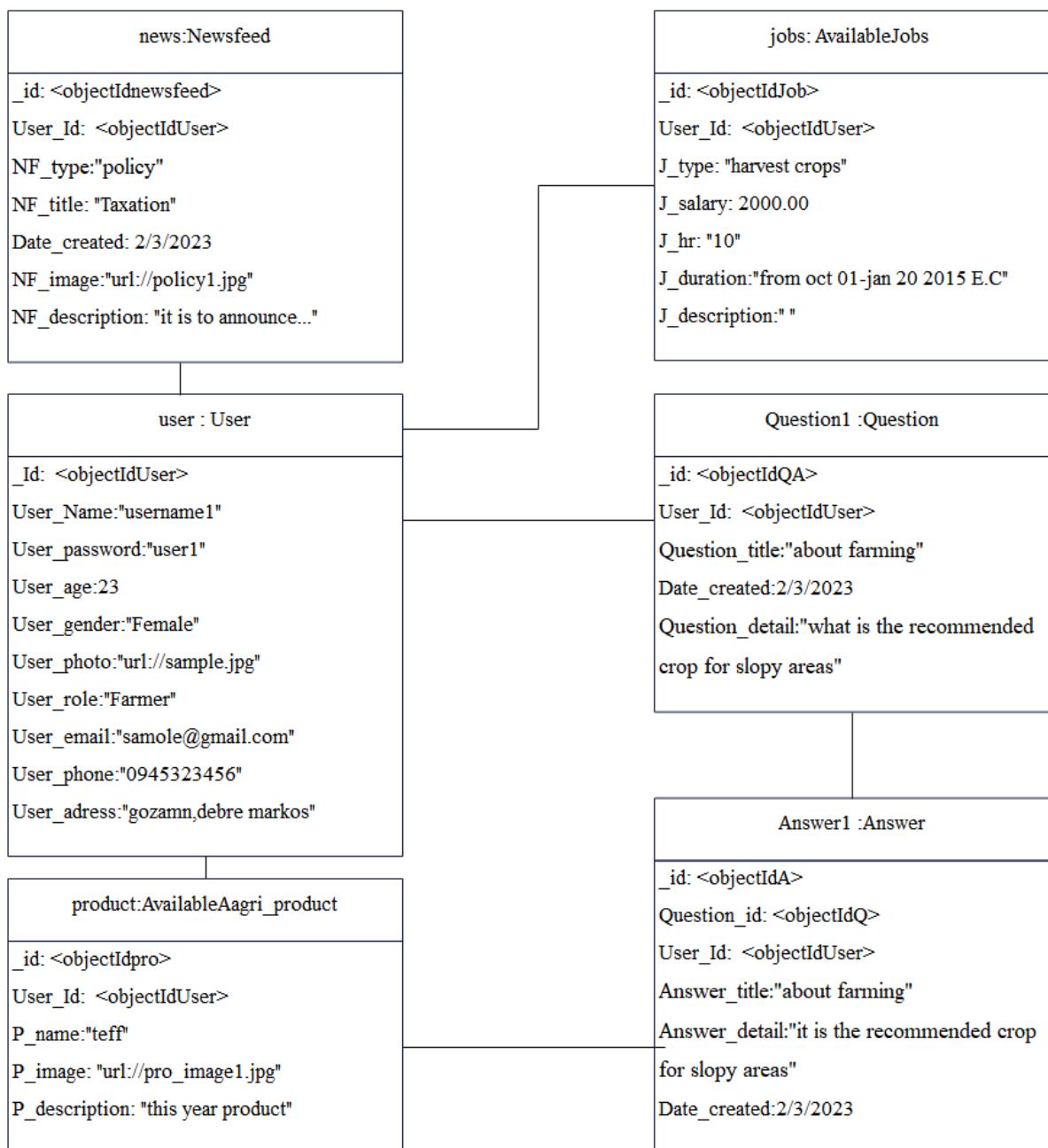


Figure 3-10 Object Diagram of the system

## User collection

Table 3-1 User collection description

Attribute name	Data type	Length	Constraints
User_ID	objectId	15	PRIMARY KEY
User_name	string	20	REQUIRED
User_password	string	8	REQUIRED
User_email	string	15	REQUIRED
User_phone	string	10	REQUIRED
User_age	INT	3	REQUIRED
User_gender	string	5	REQUIRED
User_role	string	25	REQUIRED
User_photo	OBJECT	-	REQUIRED
User_address	string	20	REQUIRED

## **Agri-products collection**

Table 3-2 Agri-products collection description

Attribute name	Data type	Length	Constraints
P_ID	objectId	15	PRIMARY KEY
P_name	string	20	REQUIRED
P_image	OBJECT	-	REQUIRED
User_id	objectId	8	REQUIRED
P_description	INT	6	REQUIRED

## **Jobs collection**

Table 3-3 Jobs collection description

Attribute name	Data type	Length	Constraints
J_ID	objectId	15	PRIMARY KEY
J_type	string	20	REQUIRED
J_salary	DOUBLE	8	REQUIRED
J_duration	DATE	15	REQUIRED
J_hr	string	15	REQUIRED
J_description	String	15	REQUIRED
User_id	objectId	15	REQUIRED

## Questions collection

Table 3-4 Questions collection description

Attribute name	Data type	Length	Constraints
Q_ID	objectId	15	PRIMARY KEY
Q_title	string	25	REQUIRED
Q_description	String	200	REQUIRED
Date	Date	64	REQUIRED
Question_id	ObjectId	15	REQUIRED
User_id	ObjectId	15	REQUIRED

## Answer collection

Table 3-5 Answer collection description

Attribute name	Data type	Length	Constraints
A_ID	ObjectId	15	PRIMARY KEY
A_title	string	25	REQUIRED
A_description	string	200	REQUIRED
Date	Date	64	REQUIRED
Question_id	ObjectId	15	REQUIRED
User_id	ObjectId	15	REQUIRED

### 3.3.5 Access Control and Security

Access control refers which user is granted to access which functionality of the system. Each user has different permissions on each subsystem of the system. The systems is required to have access control policies and security policies to protect the system and data from:

- Unauthorized access
- Malicious modification of data
- Accidental introduction of inconsistency

The following tables describes the functionalities that each user shall allowed to access.

Subsystems	Classes	Operations	Users					
			admin	DA worker	IC	Farmer	Labor worker	Buyer
User Interface	User management	Open()	✓	✓	✓	✓	✓	✓
		Close()	✓	✓	✓	✓	✓	✓
	Information management	Open()			✓			
		Close()			✓			
	Agri-products management	Open()				✓		✓
		Close()				✓		✓
	Jobs management	Open()				✓	✓	
		Close()				✓	✓	
	Data management	Open()	✓	✓	✓	✓	✓	✓
		Close()	✓	✓	✓	✓	✓	✓
	System admin	manageAccount()	✓					

Subsystems	Classes	Operations	Users					
			admin	DA worker	IC	Farmer	Labor worker	Buyer
User management	DA workers	Ans()		✓				
	Farmers	manageNewsfeed()			✓			
		manageAgricultri-products()				✓		
		viewNewsfeeds()				✓		
		manageJobs()				✓		
		Ask()				✓		
	Labor workers	viewJobs()					✓	
Information management	Buyers	viewAgricultri-products()						✓
	Weather information	generateWeatherInfo()			✓			
	Crop information	generateCrop()			✓			
New technology information	New technology information	generateTech()			✓			

Subsystems	Classes	Operations	Users					
			admin	DA worker	IC	Farmer	Labor worker	Buyer
Gov't policy	generatePolicy()				✓			
	Market information	generatePriceindex()			✓			
Agri-products management	Agri-products	manageProducts()				✓		
Jobs management	Jobs	manageJobs()				✓		
Data management	Data access	Read()	✓	✓	✓	✓	✓	✓
		Write()	✓	✓	✓	✓		
		Update()	✓	✓	✓	✓		
	Data storage	Connect()	✓	✓	✓	✓	✓	✓
		Read()	✓	✓	✓	✓	✓	✓
		Write()	✓	✓	✓	✓	✓	✓
		Delete()	✓	✓	✓	✓	✓	✓
	System message	Disconnect()	✓	✓	✓	✓	✓	✓
	System message	Create()		✓	✓	✓		
		Display()		✓	✓	✓		

Table 3-6 Access control of the system

### 3.4 Detailed Class diagram

This section describes the classes in a detailed way and their public interfaces. The overview, dependencies, constraints, attributes, and operations of classes of the system are presented below in the following detail.

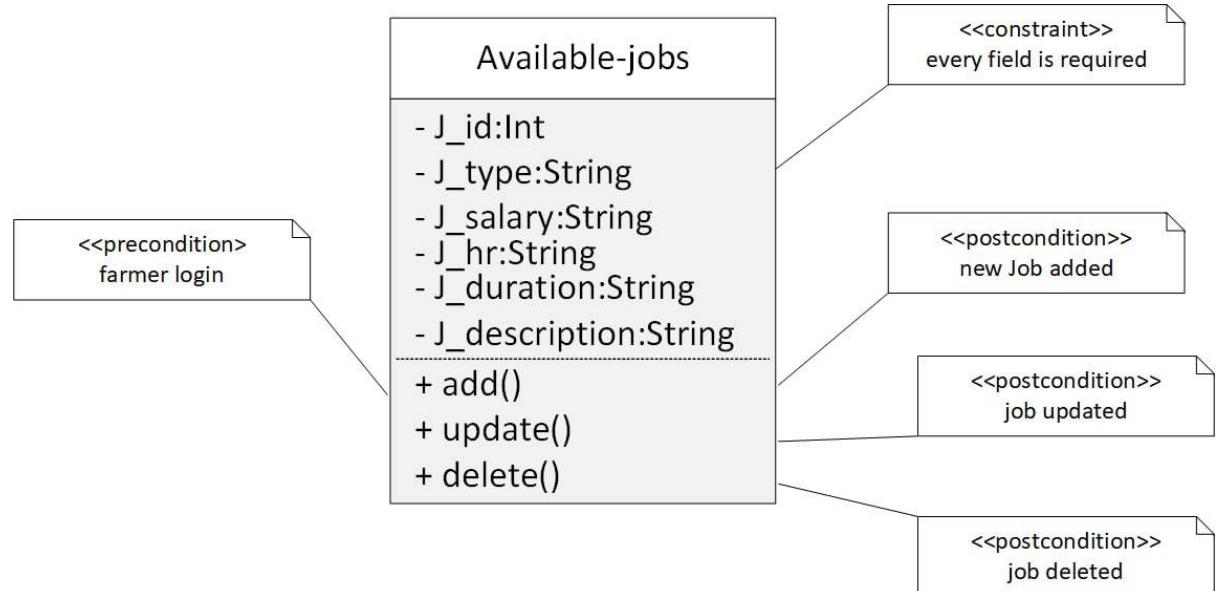


Figure 3-11 Detailed Class Diagram of jobs

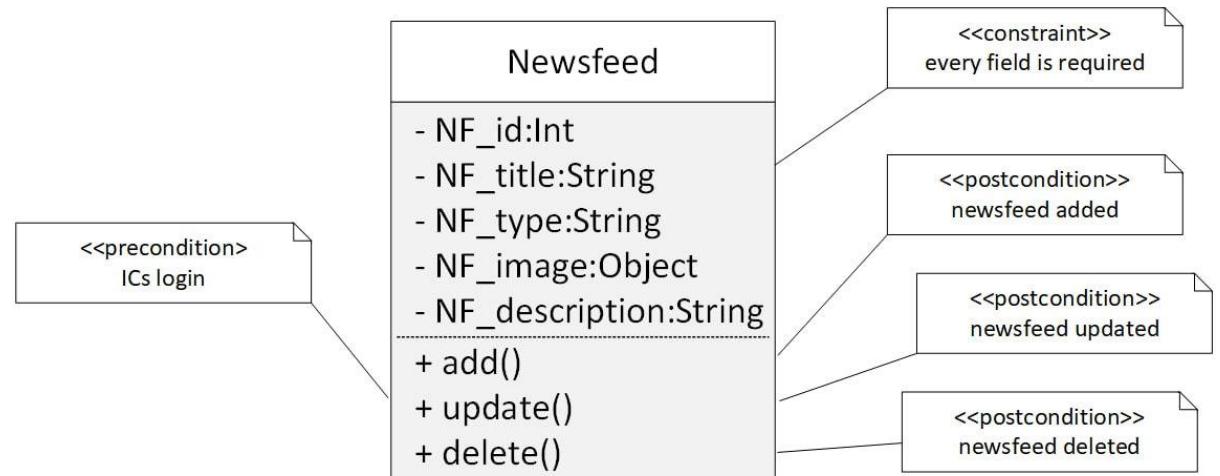


Figure 3-12 Detailed Class Diagram of Newsfeed

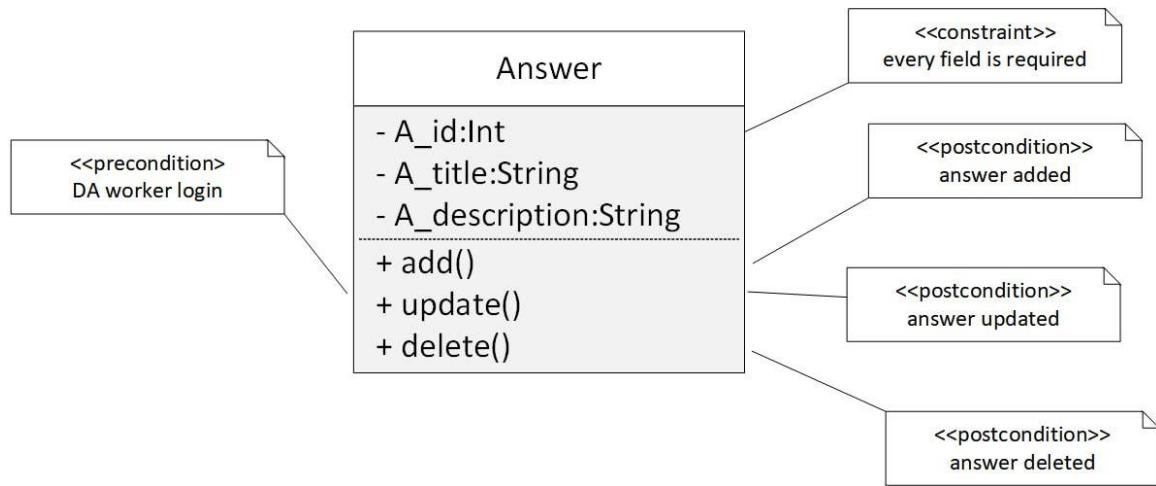


Figure 3-13 Detailed Class Diagram of Answer

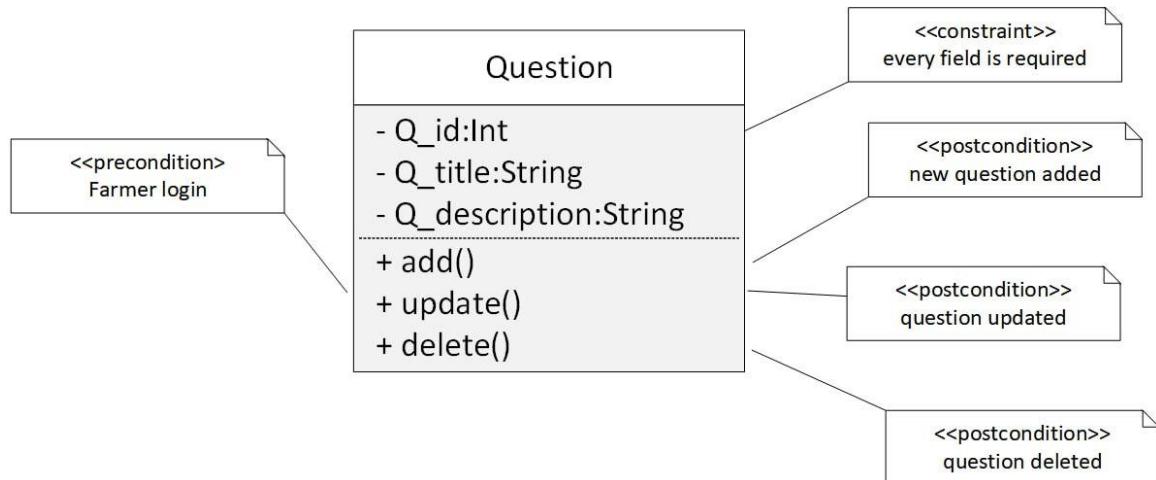


Figure 3-14 Detailed Class Diagram of Question

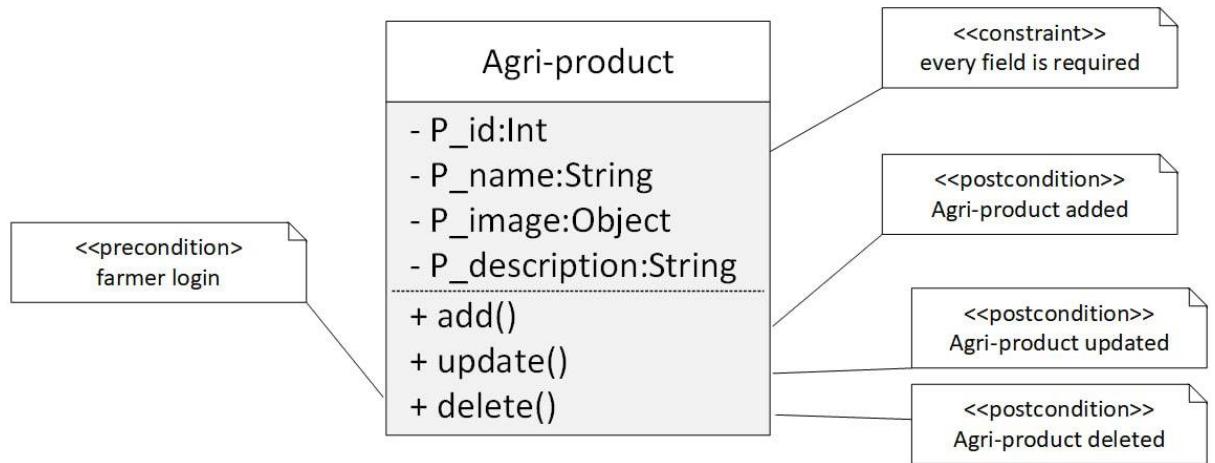


Figure 3-15 Detailed Class Diagram of Agri-product

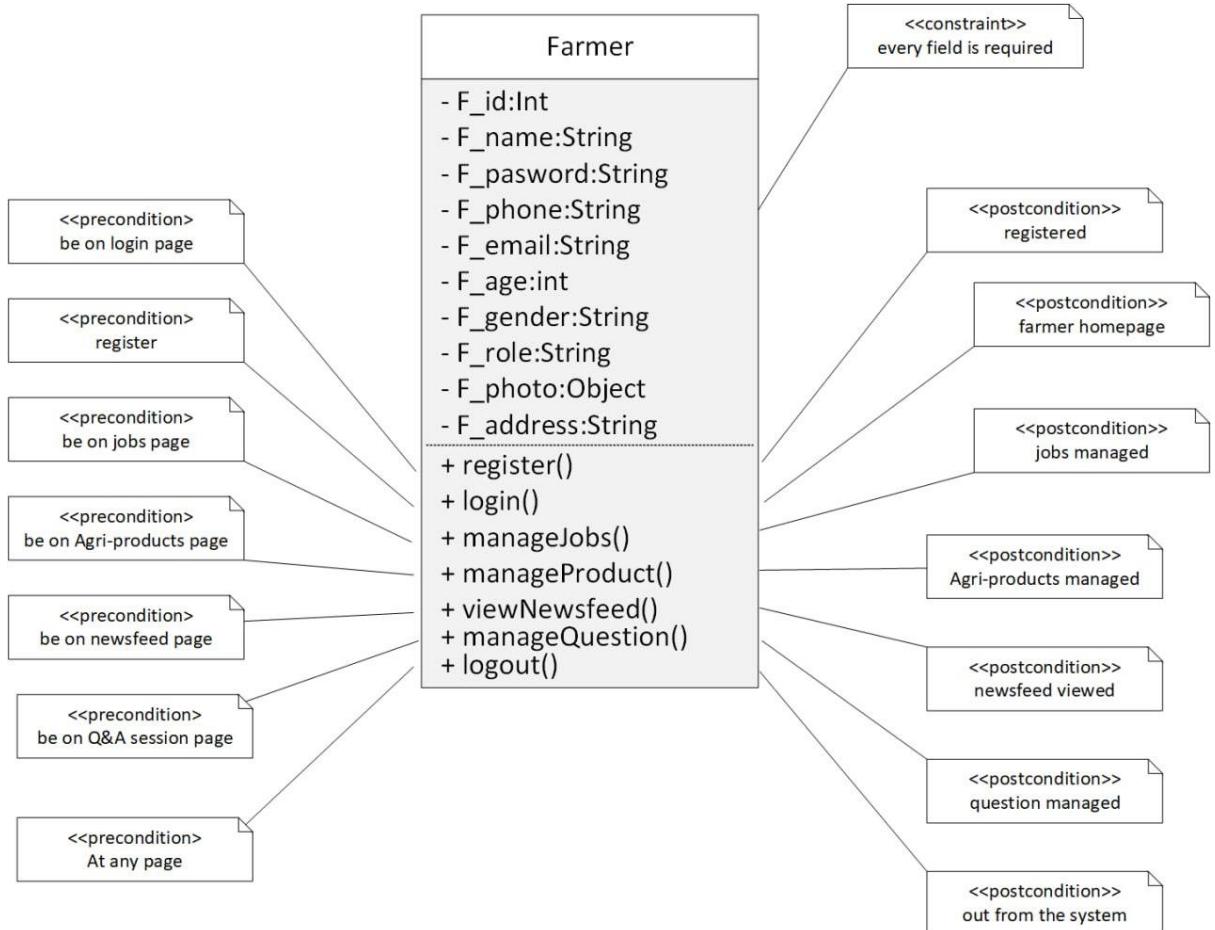


Figure 3-16 Detailed Class Diagram of Farmers

### 3.5 Packages

In order to reduce the complexity of the application domain, smaller parts that are identified during the analysis called classes are organized into packages.

Packages of the system are presented as follows:

- **User Interface Package:** This package contains different classes, which control the loading and unloading of the interfaces with which the users interface different parts of the system.
- **User Management Package:** This package contains classes, which are responsible for functionalities of the system administrator, DA workers, ICs, Farmers, Labour workers and buyers account management.
- **Newsfeed Management Package:** This package contains classes, which are responsible for functionalities of seasonal weather information, crop information, new technology, government schema and price index visualization.
- **Agri-products Management Package:** This package contains classes, which are responsible for functionalities of agri-products that farmers want to sell.
- **Jobs Management Package:** This package contains classes, which are responsible for functionalities of available jobs, the farmers want to hire.
- **Data Management Package:** The data management package contains classes responsible for data storage and information retrieval triggered by the subsystems.
- **Dependency of package:** Users interact with different packages of the systems only through the interfaces designed. To reduce dependencies among packages, each package will interact with the Data Management subsystem directly through their interfaces

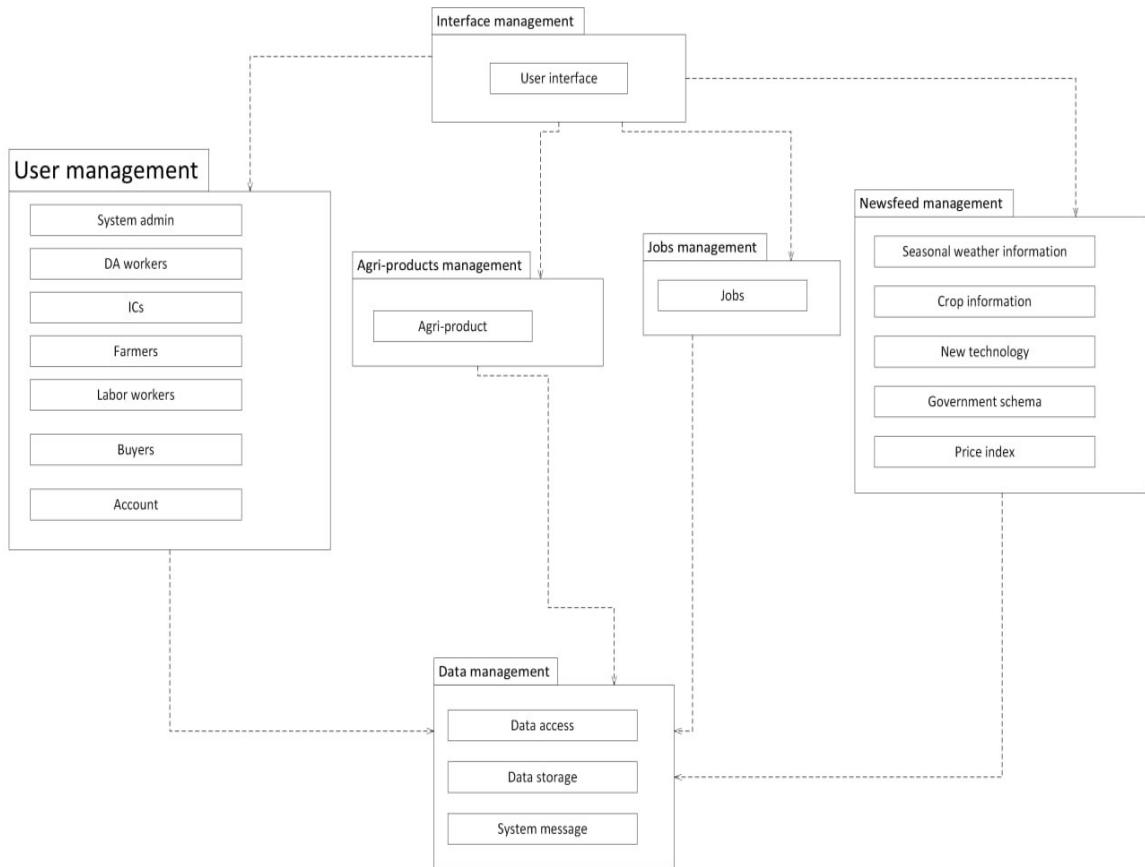


Figure 3-17 Packages of the system and their dependency

## References

- [1] [Ethiopia Economy Ethiopian Economy, business opportunities in Ethiopia \(globaltenders.com\)](#)
  - [2] <https://www.worldometers.info/world-population/ethiopia-population/>
  - [3] Tanenbaum, Andrew S., and Maarten van Steen. Distributed Systems: Principles and Paradigms. Pearson Education, 2017.
  - [4] Bruegge, B. and Dutoit, A.H., 2009. Object--Oriented Software Engineering. Using UML, Patterns, and Java.
- Addis Ababa University, Department of Computer Science, Undergraduate Final Project Guideline