Bahir Dar University
Bahir Dar Institute of Technology
Faculty of computing

Industrial project on Web based Agricultural Raw Material Distribution and Order System (ARMDOS) for Amhara Region agricultural input supply directorate

Submitted to the faculty of computing in partial fulfillment of the requirements

for the degree of Bachelor of

Science in

INFORMATION TECHNOLOGY

Group members

Name	**ID Number**
1. Alamin Esayas ………………………………	1200851
2. Endeabay Feleke ………………………………....	1201557
3. Kidist Woreta ……………………………………	1109272

Advisor :MRs. Birtukan Abebe

JANUARY 2023

BAHIR DAR UNIVERSITY, BAHIR DAR INSTITUTE OF TECHNOLOGY

Declaration

The Project is our own and has not been presented for a degree in any other university and all the sources of material used for the project have been duly acknowledged.

Alamin Esayas

Name **Signature**

Endeabay Feleke

Name **Signature**

Kidist Woreta

Name **Signature**

Faculty: Computing

Program: information technology

Project Title: Web based agricultural raw materials distribution and order for amhara region input supply directorate

MRs. Birtukan Abebe

Name of Advisor **Signature**

Examining committee members **signature** **Date**

1. Examiner1 ……………………………………… ____________________ ……………………
2. Examiner2 ……………………………………… ____________________ ……………………

It is approved that this project has been written in compliance with the formatting rules laid down by the faculty.

Roles and Responsibilities of the Group Members

List of Tasks	List members		
	Alamin Esayas	Endeabay Feleke	Kidist Woreta
Requirement gathering			✓
Use case drawing and documentation	✓	✓	
Activity diagram and sequence diagram drawing	✓	✓	
Analysis class model	✓		
CRC drawing		✓	
Component diagram drawing			✓
Deployment diagram drawing		✓	
Drawing design class diagram	✓		
user interface design	✓	✓	✓

Table of Responsibility of Team Members

Acknowledgment

First and foremost, we would like to thank almighty that permits us to do and enable us to be successful in our final year project and to complete without any problem (actually there were a lot of problems) as his permission. We also greatly thanks to our advisor Birtukan Abebe, who have given timely advice as well as encouragement to the completion of our project and her continuous support in every phase of this project. Then we would like to thanks instructor of faculty of computing who helps as by giving information about the project. Finally, we would like to thank our classmates and friends.

List of acronyms

ARMDOS: Agriculture Raw Material Distribution and Order System
FREQ: Functional Requirement
UCD: Use Case Diagram
UC: Use Case
BR: Business Rule
UML: Unified Modeling Language
NRF: Non-Functional Requirement
CRC: Class Responsibility Collaboration
UI: User Interface

List of Figures

List of Tables

Table of Contents

Contents

Abstract

This document contains the brief description of background information of the bureau and project, method of data collection and analysis, description of the existing system and modeling and design of the proposed system.

This delivery system currently has many problems, even if it is semi-automated system. Due to this we are going to develop online system to support the delivery of raw materials for farmers in urban and rural areas of Amhara region which means farmers can order the raw materials from the website using their mobile phones or personal computers through internet.

Chapter One: Introduction

1.1 Background

Raw material distribution for farmers generally involves the delivery of necessary supplies. Delivery services usually include bringing the materials to the farm, loading them onto vehicles and distribution within the farm. Most delivery services combine their services with the supply of related information regarding the proper methods and applications, providing agronomic tips or market information.

Amhara agricultural input supply directorate is one of the biggest directorates in the amhara agriculture bureau which is involved in raw material distribution. Currently the institution comprises of 230 employees. It is located at the capital city of amhara regional state, bahirdar city kebele 03 near grand hotel.

Mission

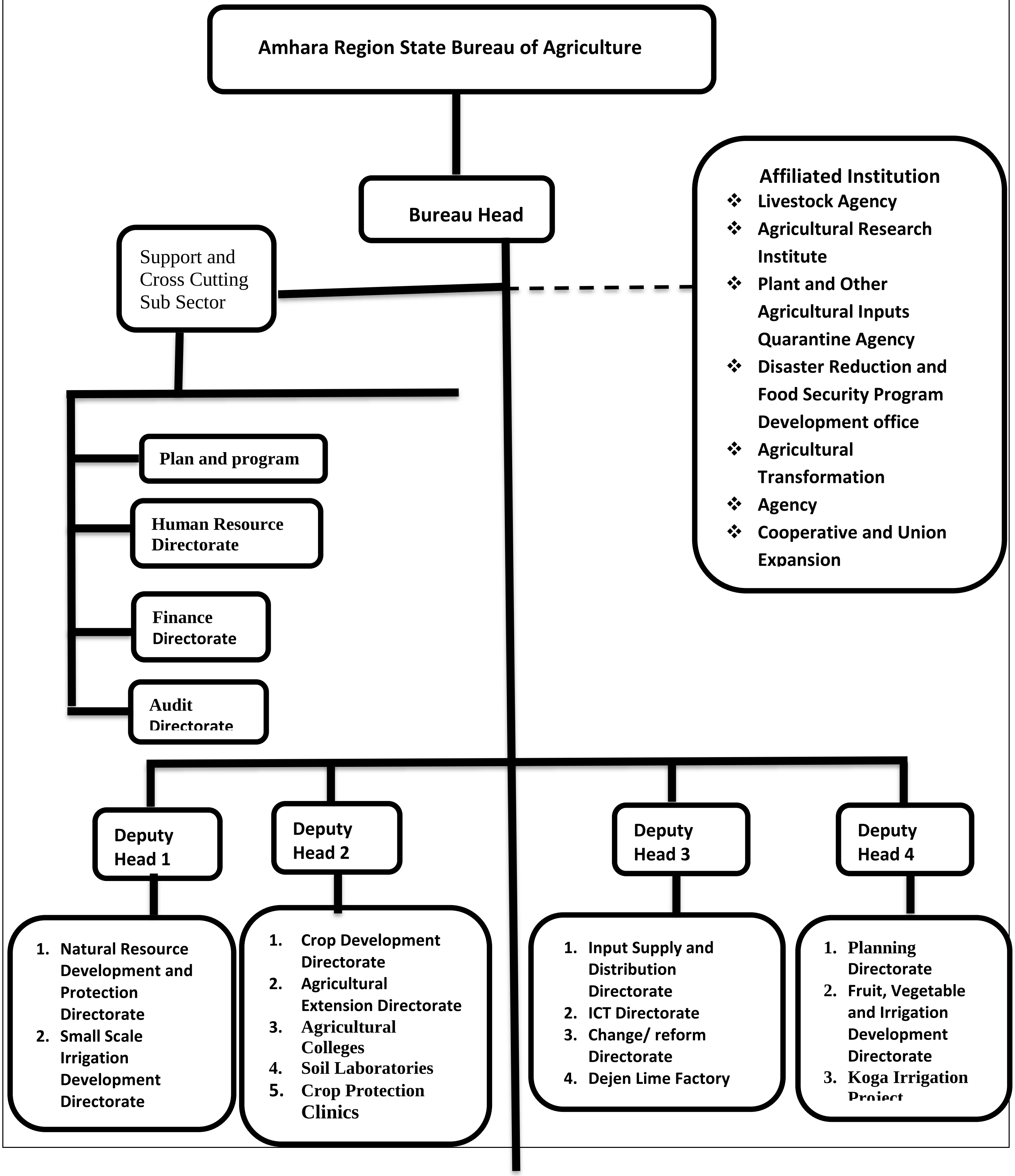
Based on integrated water shed development, in agro-ecological friendly manner, widely using the labor force of farmers; providing the extension services which is mainly characterized by modern agricultural inputs and maximize the benefit of farmers at a large extent.

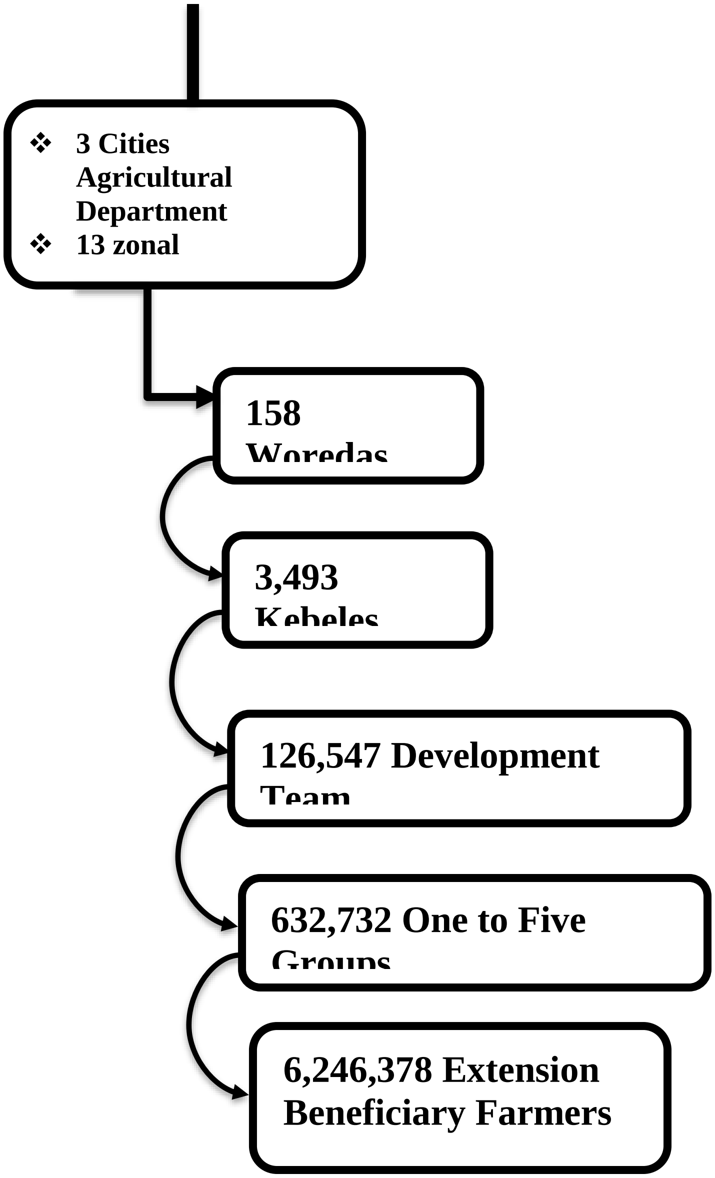
Vision

- Working to achieve our mission and vision.
- Working hard to fulfill customers need.
- Make a society which have a culture of saving and investment.
- We believe agriculture is pillar for development.
- Make sure societies participation and usability.

Organizational structure

The organizational structure of Amhara Region State Bureau of Agriculture is as follows:


Amhara Region State Bureau of Agriculture
Bureau Head
Support and Cross Cutting Sub Sector
Plan and program
Human Resource Directorate
Finance Directorate
Audit Directorate
Affiliated Institution
❖ Livestock Agency
❖ Agricultural Research Institute
❖ Plant and Other Agricultural Inputs Quarantine Agency
❖ Disaster Reduction and Food Security Program Development office
❖ Agricultural Transformation
❖ Agency
❖ Cooperative and Union Expansion
Deputy Head 1
Deputy Head 2
Deputy Head 3
Deputy Head 4
1. Natural Resource Development and Protection Directorate
2. Small Scale Irrigation Development Directorate
1. Crop Development Directorate
2. Agricultural Extension Directorate
3. Agricultural Colleges
4. Soil Laboratories
5. Crop Protection Clinics
1. Input Supply and Distribution Directorate
2. ICT Directorate
3. Change/ reform Directorate
4. Dejen Lime Factory
1. Planning Directorate
2. Fruit, Vegetable and Irrigation Development Directorate
3. Koga Irrigation Project



❖ 3 Cities Agricultural Department
❖ 13 zonal
158 Woredas
3,493 Kebeles
126,547 Development Team
632,732 One to Five Groups
6,246,378 Extension Beneficiary Farmers

Figure 1 organizational structure

Raw materials delivery for farmers may include the following:

1. Seeds,
2. Fertilizers,
3. Crop protection products,
4. Irrigation systems and supplies,
5. Farm equipment and supplies,
6. Animal feed and bedding materials,
7. Veterinary products and services,
8. Business consulting services,
9. Technology tools to assist in production and marketing of crops (smart phones/computer software/application

1.2 Statement of the problem

The traditional ARMDOS have unreliable information about farmers need and their land and it also involves several inefficient and costly processes. Farming produces a large variety of products, making it difficult to efficiently source the correct materials from various different places. Many farmers are forced to rely on manual labor and inefficient transportation methods due to limited resources and funds. This leads to a number of problems including

- Unreliable and unorganized information about the farmer needs
- high production costs,
- slow supply chain speeds, and
- Difficulty in meeting customer demands based on their need and reliable information.

Additionally, farmers may not have access to necessary information about current raw materials in the directorate prices or technology that could help them manage their operations better. Unreliable information gathering systems also present further challenges and create sustainability concerns as crop damage can occur from delays in distribute or inappropriate storage conditions during transport and it may send for unnecessary places. Finally, rural areas often lack access to adequate transportation infrastructure which further hinders efficient delivery of raw materials. And the directorate also vulnerable to unexpected wastage of raw materials.

1.3 Objectives

1.3.1 *General Objective*

The main objective of this project is to design Web based ARMDOS for amhara agricultural input supply directorate.

1.3.2 *Specific Objectives*

The specific objectives of our project are: -

- Determining what to build and understand the problem domain of the system using low fidelity models consisting of transforming user requirements in to system requirements.
- Analyze the limitations of the existing system
- Suggest alternative problems and selecting the best among the alternative
- Analysis and design a system which is secured, efficient and user friendly
- Design the system that will solve the current problems and provide reliable functionalities.
- Implementation of the project using programming language according to the design specification in efficient way.
- Giving recommendation on further studies to be conducted on the areas of raw material distribution system

1.4 Methodology

1.4.1 *Requirement gathering methods*

There are different data gathering technique which helps us to understand the present systems general activities and its procedures. We used three methodologies to gather the needed information. The methods we use for data collection are:

- **Interviewing**: As a method for the collection of data about the activities of directorate we use interviewing method to understand peoples who belongs to the current system also we raised questions that helps us to develop the new system.
- **Direct Observation:** We observe the current system and identify the problems regarding to the working process. So, it helps us as easy way to understand the system and to develop the project. By observing their system structure of the directorate and we design a project to solve the problems.
- **Document Analysis**: we analyze some documents and forms found in the office to get information about the current system.

1.4.2 *Analysis and design Methodology*

System Analysis: is the process of breakdown an entire system into module, analyzing each module separately, and determining the relationship between them. In system development process we use system modeling i.e. creating model of the system. System modeling is the process of creating a model of system by analyzing and organizing the system elements. This is the phase where deeply understanding of the existing system problems and finding alternative solution. Models are:

Use case diagram: Use cases define interactions between external actors and the system to attain particular goals. There are three basic elements that make up a use case:

- **Actors**: Actors are the type of users that interact with the system.
- **System**: Use cases capture functional requirements that specify the intended behavior of the system.
- **Goals**: Use cases are typically initiated by a user to fulfill goals describing the activities and variants involved in attaining the goal

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 (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

Activity diagram: An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent.

Sequence diagram: Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. Like

- ✓ Model high-level interaction between active objects in a system
- ✓ Model the interaction between object instances within a collaboration that realizes a use case
- ✓ Model the interaction between objects within a collaboration that realizes an operation
- ✓ Either model generic interactions (showing all possible paths through the interaction) or specific instances of interaction (showing just one path through the interaction)

System Design*:* - to design the system the project team has choose Object Oriented Modeling techniques and Unified modeling language tools. Design models are:

Deployment diagram: are used to visualize the topology of the physical components of a system, where the software components are deployed.

Component diagram: are describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function which covered by planned development.

Generally, the model that we are describing above can:

- ➢ Enables us to comprehensively model a system before we develop it.
- ➢ Modification of the object implementation is easy because objects are loosely coupled.
- ➢ Understanding of the structure is easy because object-oriented modeling and tools used to represent real world entities.
- ➢ Direct manipulation of architectural components is possible because several object-oriented programming languages exist.

1.4.3 *Implementation Methodology*

1.4.3.1 Software tools

The software development tools that we use are:

- **MERN STACK**
  - **React JS**: - It is a java script library that Facebook built. It is used to build the UI elements that go into a single page web application's user interface.

    - It is used to develop **front end** or **client –side** script.
  - **MongoDB**: -The application data is stored in this document-oriented, No-SQL database.

    -MongoDB's document model is simple for developers to learn and use, while still providing all the capabilities needed to meet the most complex requirements at any scale.
  - **Node JS**: This is the JavaScript runtime environment that is used to run the JavaScript code on the machine itself, instead of a browser.
  - **Express JS**: It is a framework that sits atop Node JS and is used to create a website's **backend** or **server-side** using Node JS functions and structures. Node JS was created to run JavaScript on computers, not to create websites, so Express JS was created to fill that gap.
  - **MS-word** and **Power point**: For document preparation and presentation.
  - **Browser**- chrome.
  - **Draw io** -to draw the use cases and other diagrams.
  - **Figma** - for user interface design.

1.4.3.2 Hardware tools

The hardware tools that we use are:

- PC (laptop): to perform our task like writing documentation and codes.
- Flash disk: to store file temporarily and used to file transfer computer to computer
- Mobile camera: to capture image that we need to the project.

1.5 Feasibility

Feasibility study is the process of determination of whether or not a project is worth doing. There are many types of feasibility study.

1.5.1 Economic Feasibility

The system which we are going to develop will have economic benefit. Those economic benefits may be tangible or intangible.

1.5.1.1 Tangible Benefits

This is the concrete benefit that can be expressed in terms of money. So, the system proposed to develop will mainly decrease corruption from distributor and raw material wastage, time lost by farmers and also a lot of money that was expend to buy paper and other materials. Also reduce the loss of data that means if it reduces the loss of data, it also reduces the cost of replacing the lost data. In addition

- Enabling on time delivery and distribution
- Easy file management.
- Reduce cost for manual data management (Reduced expenses)
- Easy update & retrieval on stored records.

1.5.1.2 Intangible Benefits

Intangible benefits are those benefits that cannot be expressed in terms of birr. Intangible benefit that the system will give is the following:

- Current time information about the farmer need
- Reliable service to the office
- Fast and reliable service for the farmers
- Little job burden to employees of office
- Small response time and many services
- Knowledge gain by project developer

1.5.2 Technical Feasibility

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available users. Technical feasibility is concerned with specifying equipment and software that will satisfy the user requirement.

This project is feasible on technical because the proposed system can run on any computer with internet access so it would be technically feasible.

1.5.3 Operational Feasibility

Operational feasibility is a measure of how well the solution will work in the organization. Operational feasibility is dependent up on the human resources available for the system. This online system for raw material distribution and order for amhara region agricultural input supply directorate will attain its desired objectives. It can solve the problems in distributing raw materials to the farmers; therefore, it will minimize the amount of effort to do all through manually.

1.6 Beneficiaries or significant of the project

The main beneficiaries of the project are

- Rural and urban farmers in the region
- Employees of the directorate
- The directorate itself

1.7 Limitations of the project

Limitations in this project will be

- The farmers will only use website i.e. We don’t make alternative mobile app.
- The farmers have less knowledge about the current technologies.
- Limited internet access in rural areas.
- The farmers pay for the raw materials manually.

1.8 Scope of the project

The scope of this project is focus on developing Web-based ARMDOS for amhara region agricultural input supply directorate.

The web-based system performs the following tasks:

- Manage raw material on the system by the directorate
- Manage land admin and create account for them
- Manage farmer and create account for them.
- Enable Farmer Order the raw material from the system
- Building web application for the users.
- Security in the form of account, password and permission.

1.9 Organization of the project

In these project documents we are discussing system details in each chapter. Before we begin to write chapter one the document includes acknowledgment, list of acronyms, abstracts, table of contents, list of figures, and list of tables.

The first chapter of this document includes the Introduction part of the project which includes sub topics like background, existing system study, proposed system, general objective, and specific objective, significant of the system, scope and methodologies and tools.

The second chapter of the document includes system feature like existing system, problem of existing system, proposed system, requirement analysis like functional requirement system use case, use case documentation, business rules, user interface prototype, activity diagram, sequence diagram, analysis class model, logical model, nonfunctional requirement and system requirement like software requirement and hardware requirement.

Chapter Two: System features

2.1 The Existing System

The traditional system of ARMDOS involves several inefficient and costly processes vulnerable to corruption and also unreliable information. Farming produces a large variety of products based on variety season and year, making it difficult to efficiently source the correct materials from various different places, season and time. The directorate has difficulty to get the real time information about the farmer need and also many farmers are forced to rely on manual labor and inefficient transportation methods due to limited resources and funds. This is difficult to the directorate to distribute the inputs to farmer in real time information based on the current season and farmer need and also leads to a number of problems including high production costs, slow supply chain speeds, and difficulty in meeting customer demands. Additionally, farmers may not have access to necessary information about current market prices or technology that could help them manage their operations better. Unreliable information systems also present further challenges and create sustainability concerns as crop damage can occur from delays in delivery or inappropriate storage conditions during transport. Finally, rural areas often lack access to adequate transportation infrastructure and unorganized information which further hinders efficient distribute of raw materials.

2.2 Proposed System

The ARMDOS aims to provide farmers with an efficient, reliable and real time information and cost-effective distribution system and also enables the Farmers to purchase the necessary raw materials they need in order to produce goods. Through this project, the directorate can distribute raw materials for the farmer based on reliable information for the current season and year, farmers will be able to access quality and sustainable supplies of raw materials on a regular, reliable basis. The project will include the following components:

1. Establishing a supply and chain: This part of the project will focus on establishing the necessary infrastructure for a reliable and consistent raw material supply and order chain. A network of suppliers and cooperatives will be established in order to ensure availability of

different types of materials like organic fertilizers, grains, and more.

2. Promotion & Marketing: This component focuses on creating awareness about the new system among farmers in different regions, as well as identifying potential adopters who could benefit from the service. Relevant media outlets along with other marketing channels like social media platforms such as Facebook, Twitter, Telegram etc., will be used in order to reach out to a larger target audience.

3. Technology Integration: In order to create a cost-effective yet efficient process for tracking materials being delivered to different farmers, mobile technology solutions such as smartphone apps or dedicated platforms will be developed which can help users get real time updates regarding their orders (e.g.: when orders are made/ received at specified locations or if there are any delays/ glitches).

4. Personnel Training & Support: The end user base needs to be educated about how they can effectively utilize the new delivery system in order to obtain best results from it. Accordingly personnel training sessions along with support resources (manuals, FAQs etc.,) need to be prepared in order facilitate smooth usage of the system/ services by potential adopters for maximum utilization/ effectiveness gains from it.

Overall this project seeks to develop an efficient and cost-effective raw materials delivery system that can provide solutions for both urban and rural based farmers across Ethiopia.

2.3 Requirement Analysis

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people.

2.3.1 Functional requirement

The functional requirement refers to the functionality of the new system, i.e. extends to what are Services it will provide to the user. Statement of service is that system should provide how the system should react to particular inputs and how the system should behave in particular situations. The following table shows functional requirement of the proposed system.

Requirement id	FREQ-1
Source	farmer
Requirement	The system shall allow farmer to order the raw materials
Description	Farmer orders raw materials from the directorate
Category	Ordering materials
Priority	High

Table FREQ-1 order material

Requirement id	FREQ-2
Source	All user
Requirement	The system shall allow users to login
Description	User login to the system to perform their task
Category	login
Priority	High

Table FREQ-2 login

Requirement id	FREQ-3
Source	Land admin
Requirement	The system shall allow land admin to manage user account
Description	Admin manage account for user
Category	Manage user account (create, reset, deactivate, activate)
Priority	High

Table FREQ-3 manage user account

Requirement id	FREQ-4
Source	farmer
Requirement	The system shall allow farmer to view response
Description	farmer view response from the system added by the directorate
Category	view response
Priority	high

Table FREQ-4 view response

Requirement id	FREQ-5
Source	Land admin
Requirement	The system shall allow land admin to add farmer
Description	farmer registers to the system
Category	add farmer
Priority	Medium

Table FREQ-5 add farmer

Requirement id	FREQ-6
Source	Directorate
Requirement	The system shall allow land admin to add notice
Description	Directorate add notice to the system for farmers and distributors
Category	add notice
Priority	High

Table FREQ-6 add notice

Requirement id	FREQ-7
Source	Directorate
Requirement	The system shall allow directorate to add material
Description	Directorate add material to the system
Category	add material
Priority	high

Table FREQ-7 add material

Requirement id	FREQ-8

Source	All user
Requirement	The system shall allow all user to logout
Description	All user leaves the system after finish his/her task
Category	Logout
Priority	Medium

Table FREQ-8 logout

2.3.2 System Use case

2.3.2.1 Use case Diagram

A UML use case diagram is the primary form of system/software requirements for a new software program under developed. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (such as UML). A key concept of use case modeling is that it helps us design a system from end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

The use case diagram shows the overall activities of these system users, it represents the functions that the actor should perform and the diagram also shows the relationships of actors that makes them common actors on specific use case. The diagram consists of the system area in order to hold all use cases, which shows the boundaries of the system.

In the proposed system there are actors who make interact with the system via the use case. We include 6 actors in the proposed system

- ✓ **Directorate:** Is an actor who has the privilege of Registering land admin, and manage (create account, delete account and add) the land admin and have right to add, manage the raw material in the system based on the season and farmers need, and also view farmer’s information when needed, sending farmers list to the distributor and has a big authority.

- ✓ **Land admin:** the actor that have the privilege to register farmer, update Land holding right, view customer’s information when needed, send information about farmer to the directorate and manages the user.
- ✓ **Distributor:** the actor who uses the system to get the list of farmers to distribute raw materials to the farmers based on information and to send the report to the directorate.
- ✓ **Farmer:** Person who uses the system for ordering the raw materials.
- ✓ **Zone:** is an entity which have the authority of approval the information sends from each woreda under it and report.
- ✓ **Woreda:** is an entity which have the authority of approval the information sends from each land admin and report.

The following figure shows use case diagram of the proposed system.

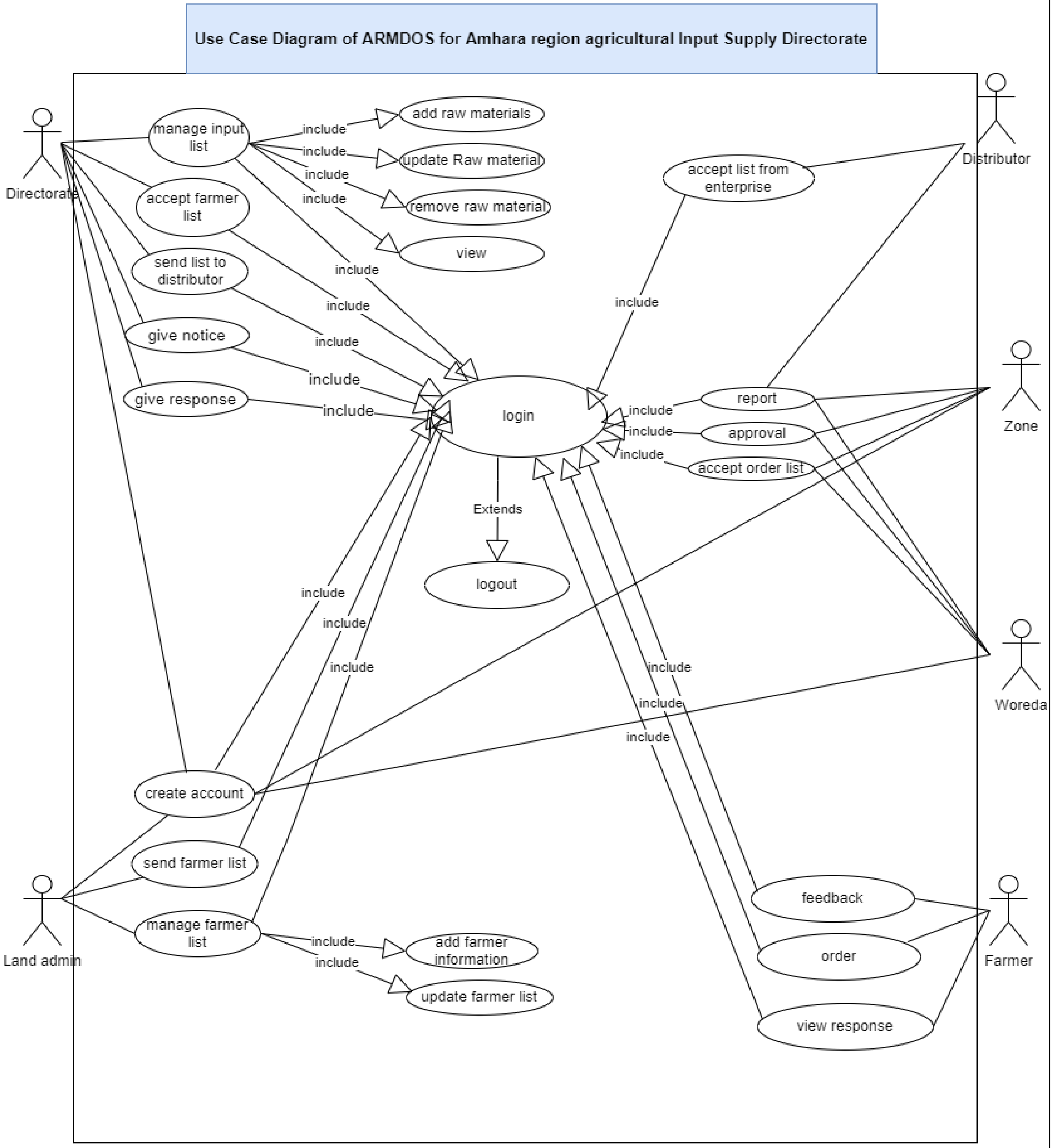


Figure 2 Use Case Diagram

2.3.2.2 Use Case documentation

This sub topic describes in detail about the use case and actors in the use case diagram, it includes the name of use case, description about the use case, the name of the actor who act on the use case, what precondition before the actor acts on the use case, post condition after the actor acts on the use case, basic course of action which describes the interaction between the actor and the system when the actor acts on the use case, and finally alternative course of action.

The following table shows use case documentation of the system.

Use case number	UC 01	
Use case name	Login	
Actor	Land admin, directorate, distributor, farmer	
Description	Checking the intended user is authorized or not	
Precondition	The user must have username and password	
Post condition	The users successfully login.	
Basic course of action	User action	System response
	1.The user opens the system 3. The user enters user name and password then click login button. 6. End use case.	2. The system displays the login page. 4. The system checks the username and password. 5. The system opens the users' home page.
Alternative course of action	If the username and password is invalid, the system displays an error message, then go back to step 3 of basic course of action.	

Table UCD of login

Use case number	UC 02	
Use case name	Logout	
Actor	Land admin, directorate, distributor, farmer	
Description	After doing any private activity in the system the user log out from the system.	
Precondition	The user should be in private page.	
Post condition	The user is in public page.	
Basic course of action	User action	System response
	1. The user clicks the logout button.	

	3. End use case.	2. The system displays the login page.
Alternative course of action	If connection is failed, try again.	

Table UCD of logout

Use case number	UC 03	
Use case name	Order raw material	
Actor	Farmer	
Description	Ordering raw materials	
Precondition	The farmer must be login and the raw material must be uploaded before	
Post condition	The farmer successfully orders the raw materials	
Basic course of action	User action	System response
	1. The farmer clicks raw material link. 3. The farmer select and click "order" link. 5. End use case.	2. The system displays the material option. 4. The system orders the raw materials.
Alternative course of action	If there is no raw materials, the system display "there is no raw material" message.	

Table UCD of order

Use case number	UC 04	
Use case name	give notice	
Actor	Directorate	
Description	The user looks all the information added by the directorate.	
Precondition	The notice must be posted before.	
Post condition	The user observes the notices.	
Basic course of action	User action	System response
	1.The user opens the system 3. The user click on view notice link. 5. the user view the posted notice 6. End use case.	2. The system displays the home page. 4. The system displays available notice that is posted by the directorate.
Alternative course of action	If connection is failed before open the notice, the system displays connection fail. Then turn back to step 3 basic course of action.	

Table UCD of give notice

Use case number	UC 05	
Use case name	View response	
Actor	Farmers	
Description	The user looks response added by the directorate in the system.	
Precondition	The user must have user name and password and the response must be added before.	
Post condition	The user access and know the response.	
Basic course of action	User action	System response
	1. The user login to the system. 3. The user click on view response links. 5. the user view response 6. End use case.	2. The system displays the farmers home page. 4. The system displays available response that is added by the directorate.
Alternative course of action	If response is not added, the system displays the message "there is no response yet".	

Table UCD of view response

Use case number	UC 6	
Use case name	Add raw material	
Actor	Directorate	
Description	Adding raw material in the system	
Precondition	Directorate must log into the system	
Post condition	The raw material successfully added in the system by directorate	
Basic course of action	User action	System response

	1.The directorate login to the system 3. The user clicks add raw material link. 5. The user fills the form and click add button. 7. End use case.	2. The system displays the directorate home page. 4. The system displays raw material registration form 6. The system adds raw material
Alternative course of action	If the user misses some information to fill, error message display and turn back to step 5 basic course of action	

Table UCD of add raw material

Use case number	UC 7	
Use case name	give response	
Actor	Directorate	
Description	give order response in the system	
Precondition	The farmer must have user name and password	
Post condition	The directorate successfully add response in the system	
Basic course of action	User action	System response
	1.The directorate login to the system 3. The directorate clicks add response link. 5. The directorate fills the information and fills each farmer response, then click "add" button. 7. End use case.	2. The system displays the directorate home page. 4. The system displays the response form. 6. The system add response to farmer
Alternative course of action	If user misses some information, system generates error message and back to step 5 to try again.	

Table UCD of give response

Use case number	UC 8
Use case name	Create account
Actor	Land admin, directorate, zone, woreda

Description	The land admin creates user account to farmers	
Precondition	Users registered to the system	
Post condition	Users account Created.	
	User Action	System Response
Basic course of action	1. The land admin login to the system 3.The land admin click on create account link 5. land admin fill create account form. 6. click on create button 8. End of use case.	2. The system display land admin home page. 4. The system display creates account form. 7. The system displays create successful message.
Alternative course of action	If the land admin fills incorrect information, the system displays error message, and go back to step 5 of basic course of action.	

Table UCD of create account

Use case number	UC 9	
Use case name	Add farmer	
Actor	Land admin	
Description	Adding of farmer to the system	
Precondition	Land admin must be login to the system	
Post condition	The farmer is added in to the system by land admin	
Basic course of action	User action	System response
	1. The user clicks add link. 3. The farmer fills and click add button. 5. End use case.	2. The system displays the registration form. 4. The system adds the farmer
Alternative course of action	If the user misses some information to fill, error message	

	display and turn back to step 3 basic course of action

Table UCD of add farmer

Use case number	UC 9	
Use case name	update farmer	
Actor	Land admin	
Description	updating farmer from the system	
Precondition	Land admin must be login to the system	
Post condition	The farmer is updated from the system by land admin	
Basic course of action	User action	System response
	1. The land admin log into the system.	
		2. The system displays land admin homepage.
	3. The land admin click update button.	
		4. The system updates the farmer.
	5. End use case.	
Alternative course of action	If the user misses some information to fill, error message display and turn back to step 2 basic courses of action	

Table UCD of update farmer

Use case number	UC 10	
Use case name	manage input list	
Actor	Directorate	
Description	Adding or removing of farmer to/from the system	
Precondition	directorate must be login to the system	
Post condition	The farmer is added/removed into/from the system by directorate	
Basic course of action	User action	System response
	1. The user clicks add/delete link.	
		2. The system displays the registration form.
	3. The farmer fills and click button.	
		4. The system adds/removes the farmer
	5. End use case.	

Alternative course of action	If the user misses some information to fill, error message display and turn back to step 3 basic courses of action	

Table UCD of Manage raw material

2.3.3 Business Rule Documentation

A business rule defines or constrains one aspect of the business that is intended to assert business structure or influence the behavior of the business. Business rules often focus on access control issues. Business rules may also pertain to business calculations.

In business rule documentation the idea includes

Name: The name should give you a good idea about the topic of the business rule.

Identifier: it shows the id of the business rules

Description: The description defines the rule exactly.

Source (optional): The source of the rule is indicated so it may be verified (it is quite common that the source of a rule is a person, often one of the project stakeholders, or a team of people).

- The following table shows business rule of the project.

Identifier	BR 001
Name	every farmer must provide his/her province to register
Description	Any applicant who wishes to join the system should provide the necessary province information.
Source	The directorate

Table BR of provide province to register

Identifier	BR 002
Name	Determine the need of username and password
Description	Any farmer who is using the system should have a username and password.
Source	The directorate

Table BR of determine the need of username and password

Identifier	BR 003
Name	User must have his/her account
Description	Every user must have his/her user account in order to access the service and perform their task.
Source	The directorate

Table BR of user account

Identifier	BR 004
Name	farmers may access the materials
Description	farmer must login to the system with the correct user name and password in order to order the raw materials and receive the materials
Source	Distributor

Table BR of accessing material

Identifier	BR 005
Name	The directorate is allowed to upload raw material.
Description	The directorate must be login first in order to upload raw materials for the farmers in the system
Source	The directorate

Table BR of upload raw material

2.3.4 User Interface prototype

The user interface prototype is built early, before the whole system is analyzed, designed and implemented. The main purpose of creating a user-interface prototype is to be able to expose and test both the functionality and the usability of the system before the real design and development starts. This way, you can ensure that you are building the right system, before you spend too much time and resources on development.

The figure below shows user interface prototype of the system.

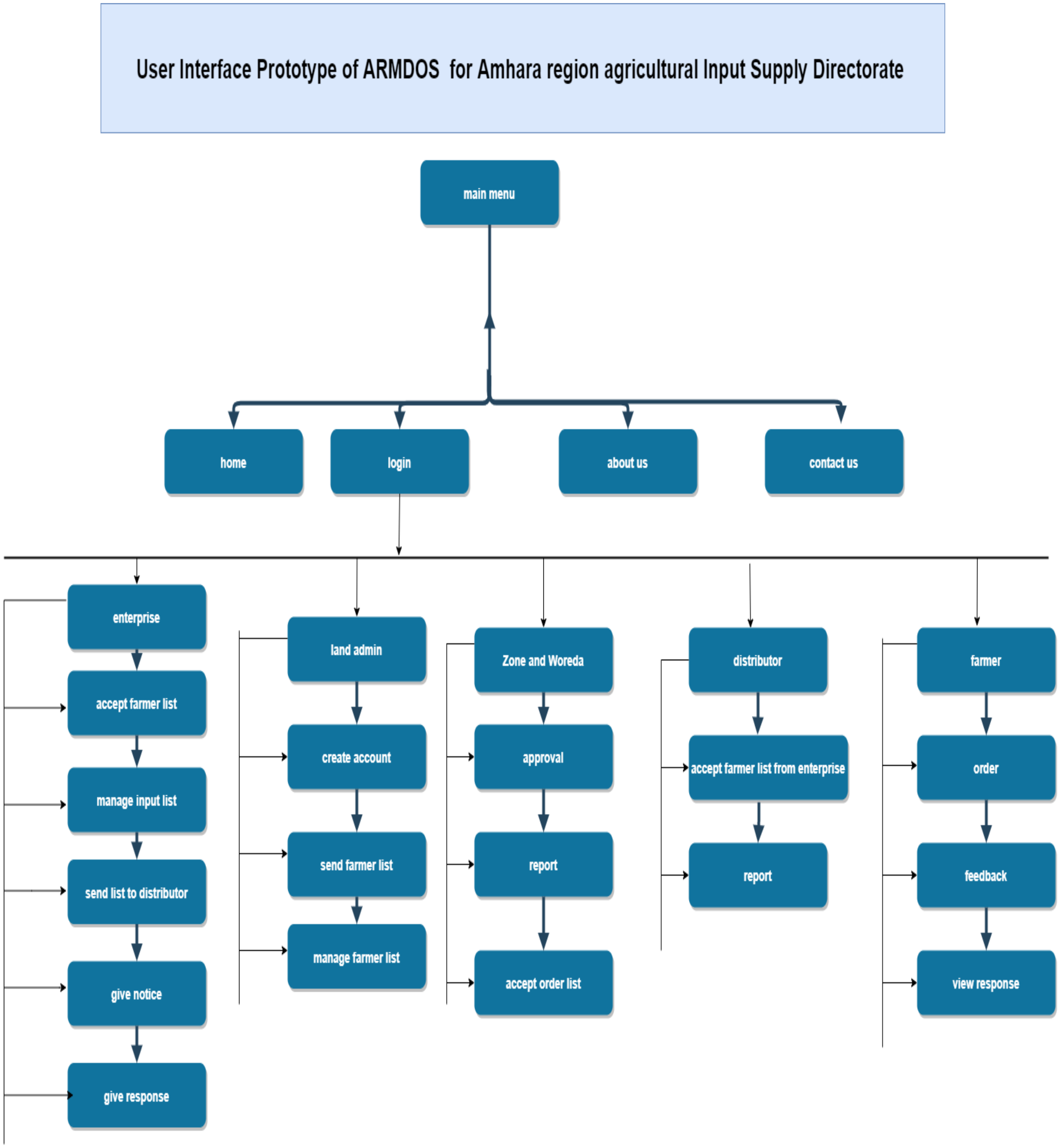


Figure 3: user interface prototype

2.3.5 State chart diagram

Draw state chart diagram for each class showing the states the object (class) passes through in its life. This models how objects change from one state to another and when the change happens.

2.3.6 Activity Diagram

UML Activity diagrams are used to document the logic of a single operation or method, a single use case (may be the basic course of action or the alternate course of action) or the flow of a logic of a business process. They are the object oriented equivalent of flow charts and data flow diagrams in the structured development approach.

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- rounded rectangles represent actions;
- diamonds represent decisions;
- bars represent the start (split) or end (join) of concurrent activities;
- a black circle represents the start (initial node) of the workflow;
- An encircled black circle represents the end (final node).

The following figure shows activity diagram of each activity in the system.

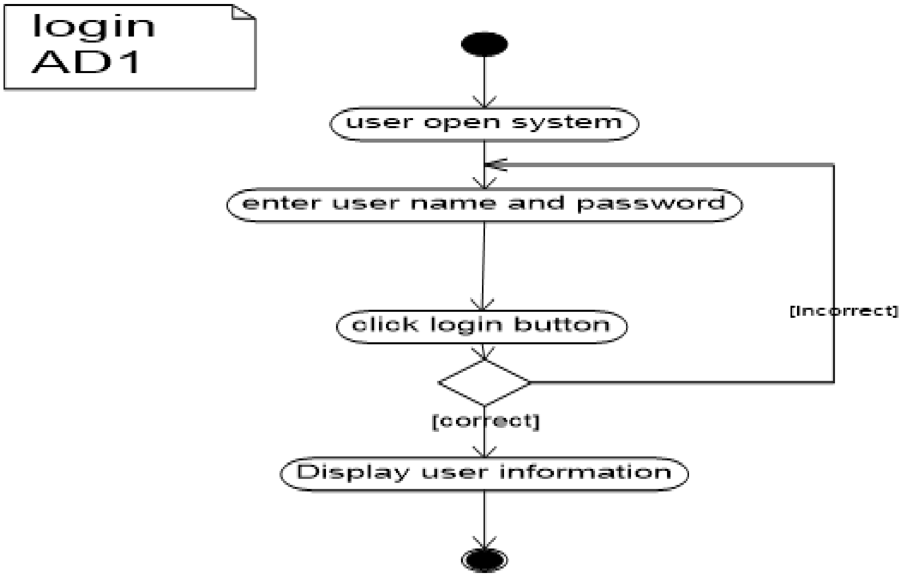


Figure 4: activity diagram login

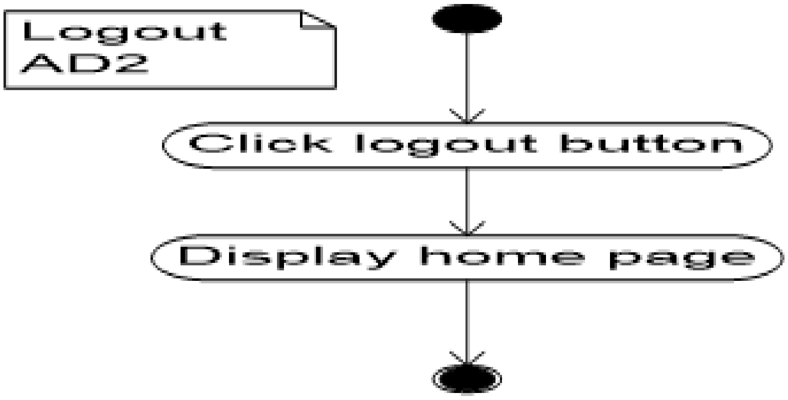


Figure 5: activity diagram logout

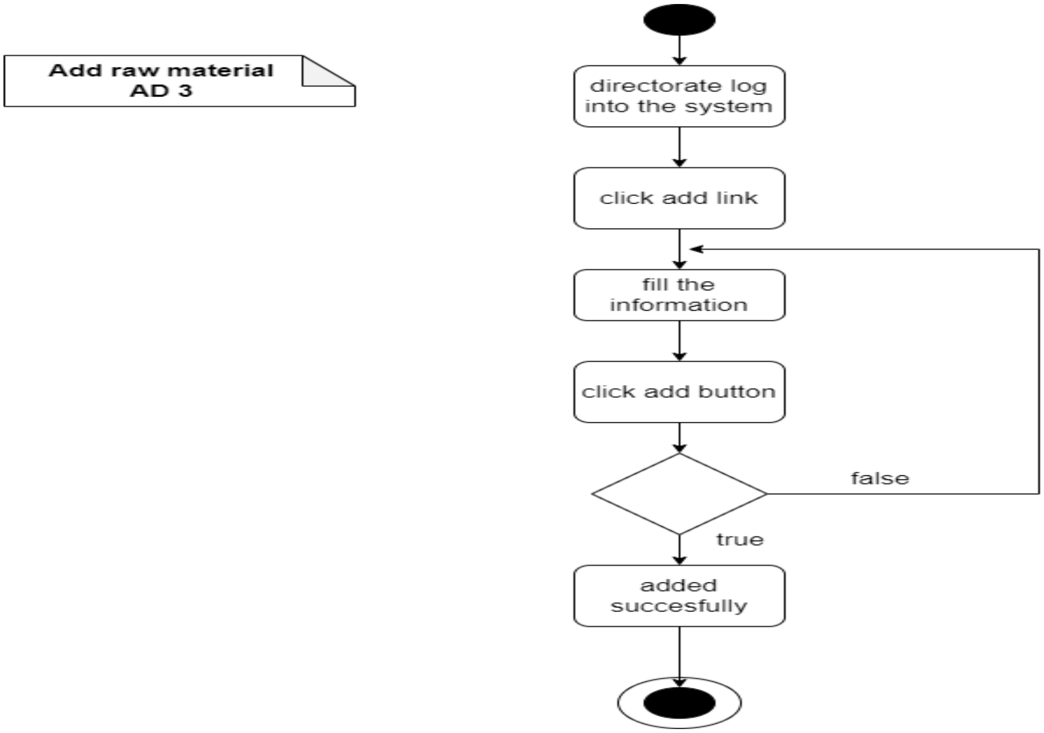


Figure 6: activity diagram to add raw material

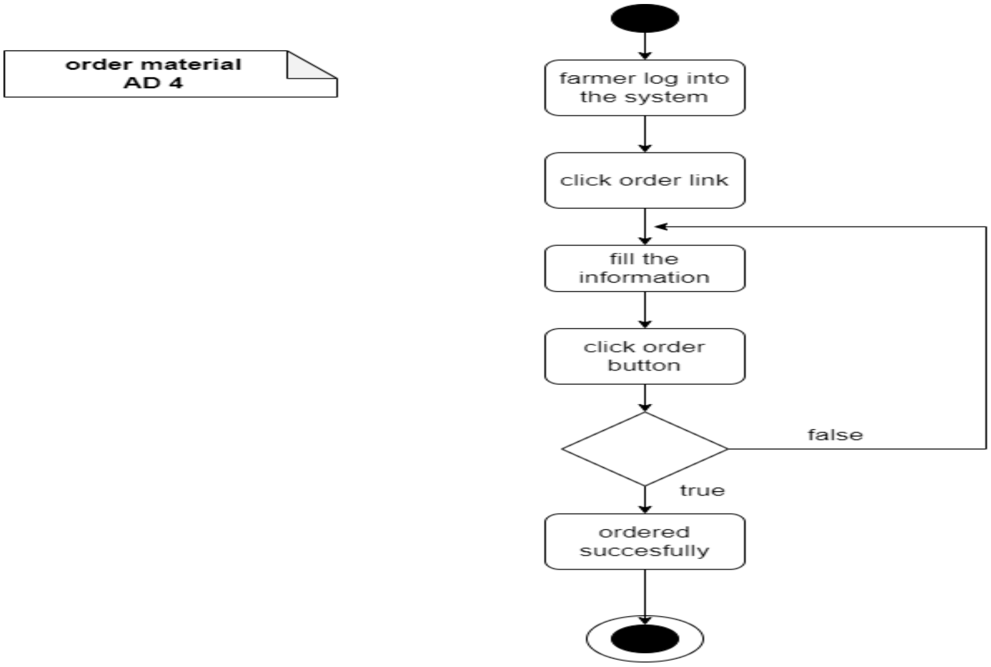


Figure 7: activity diagram order material

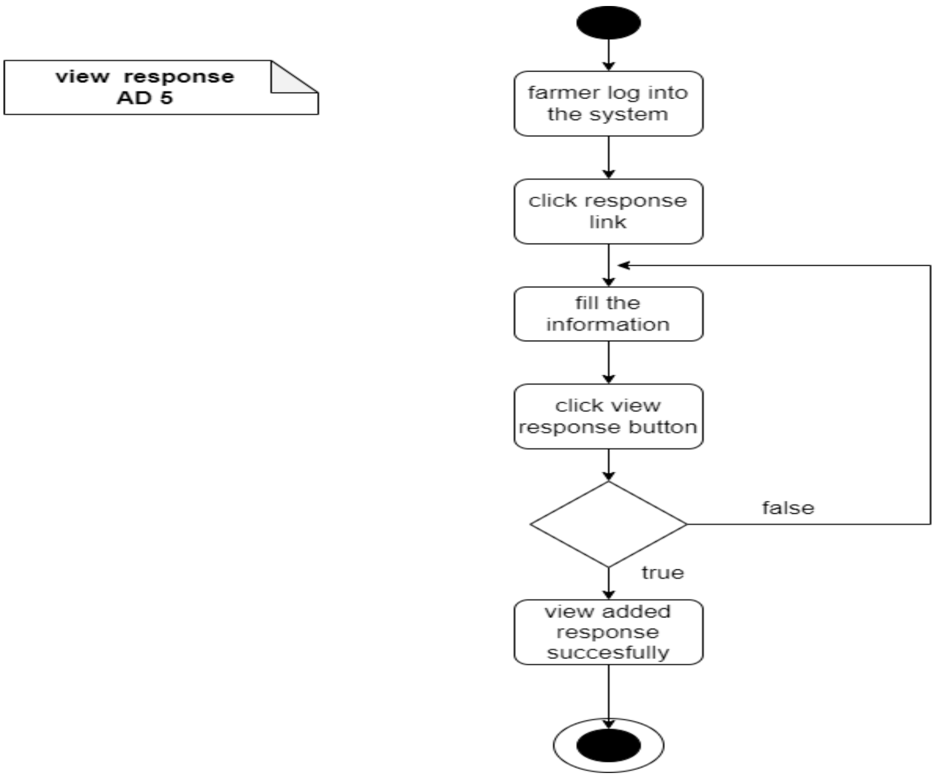

view response
AD 5
farmer log into the system
click response link
fill the information
click view response button
false
true
view added response succesfully

Figure 8: activity diagram view response

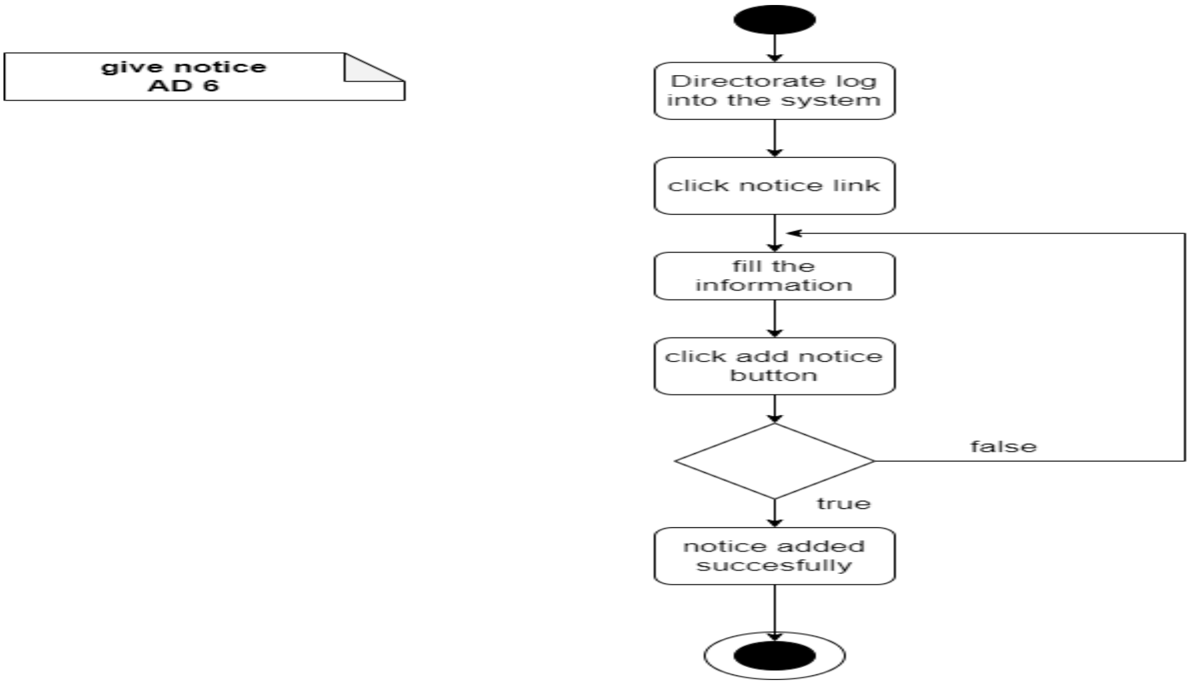


Figure 9: activity diagram give notice

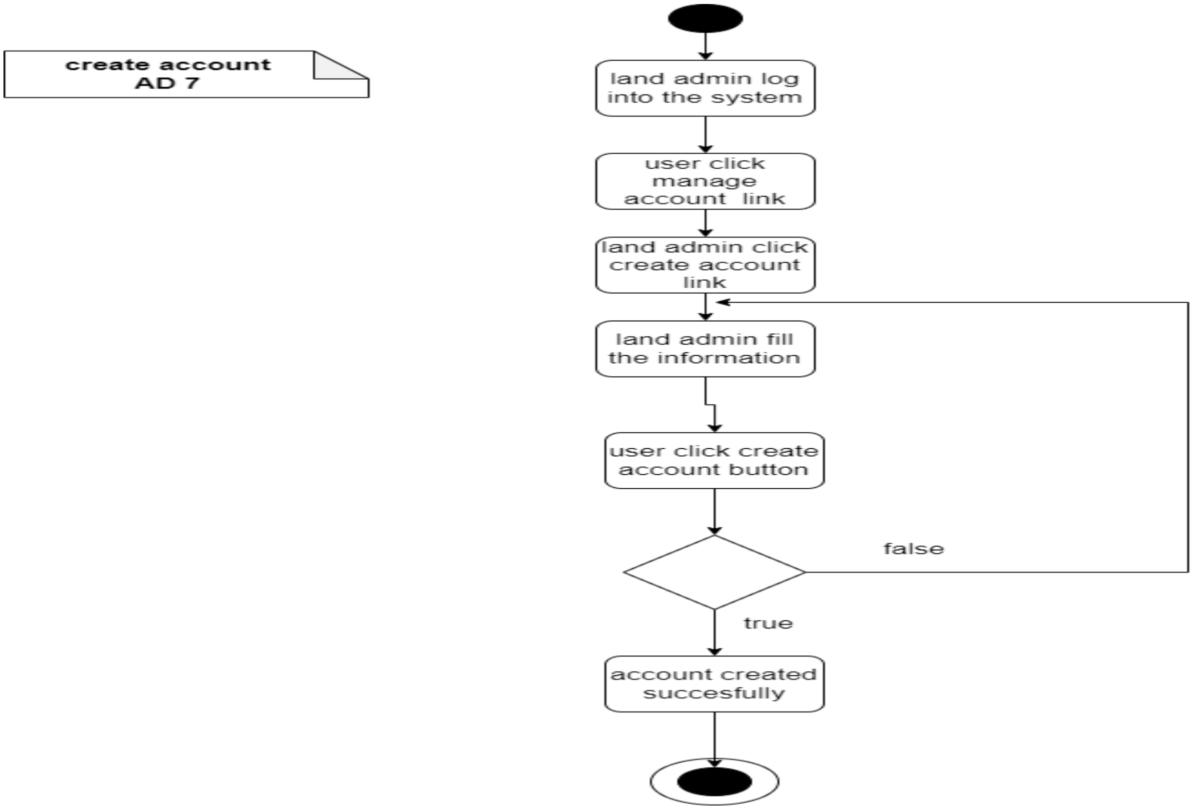

create account
AD 7
land admin log into the system
user click manage account link
land admin click create account link
land admin fill the information
user click create account button
false
true
account created succesfully

Figure 10: activity diagram create account

2.3.7 *Sequence diagram*

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

A sequence diagram 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 diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

The following figure shows the main sequence diagrams.

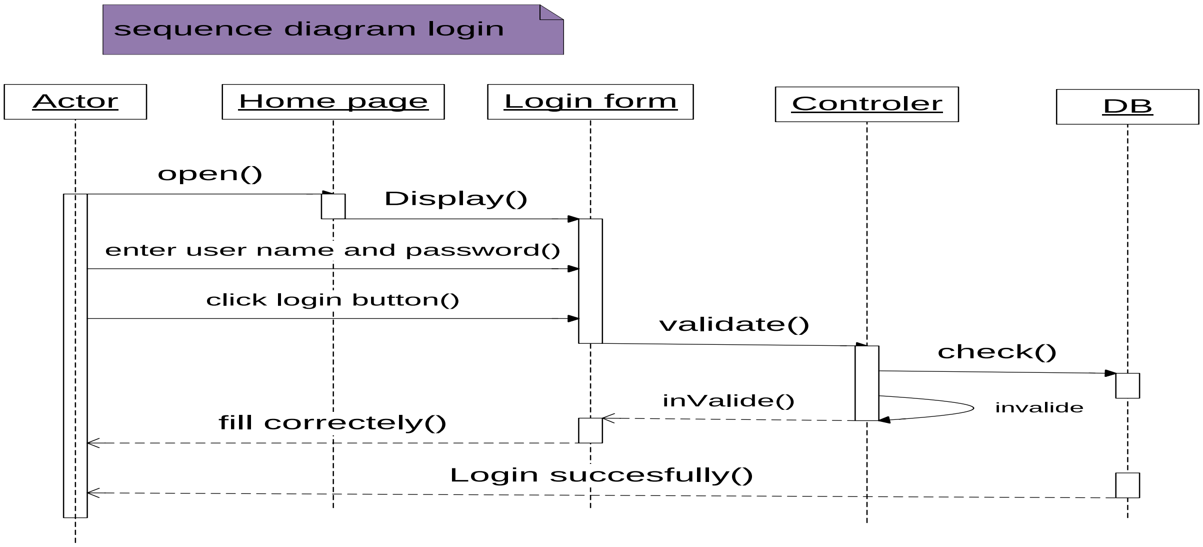


Figure 11: sequence diagram login

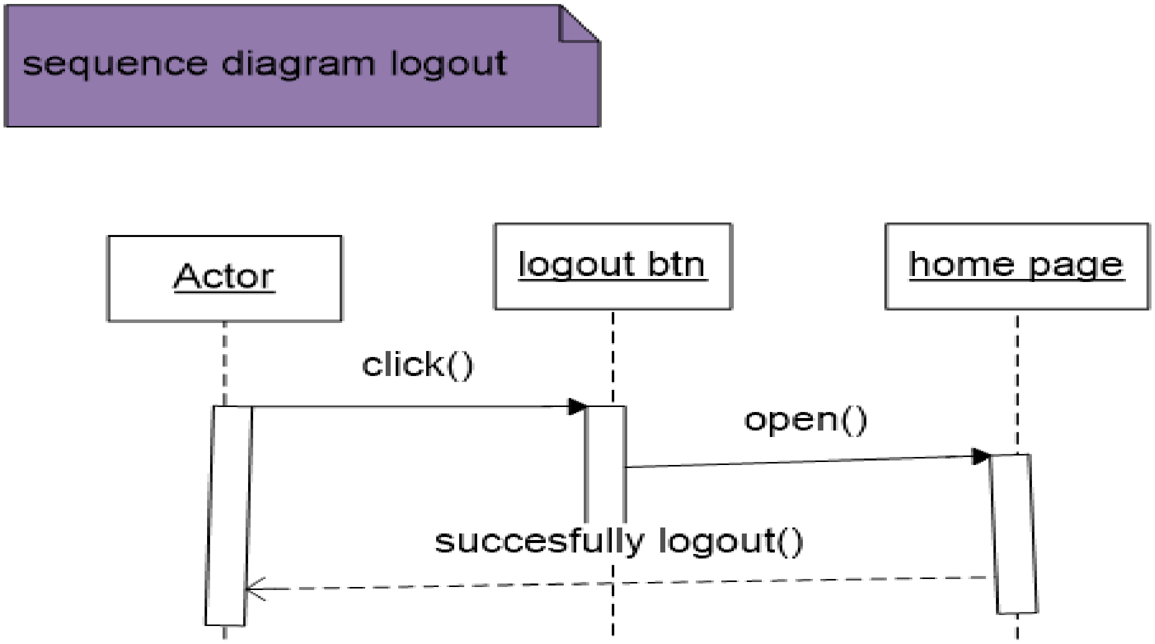


Figure 12: sequence diagram logout

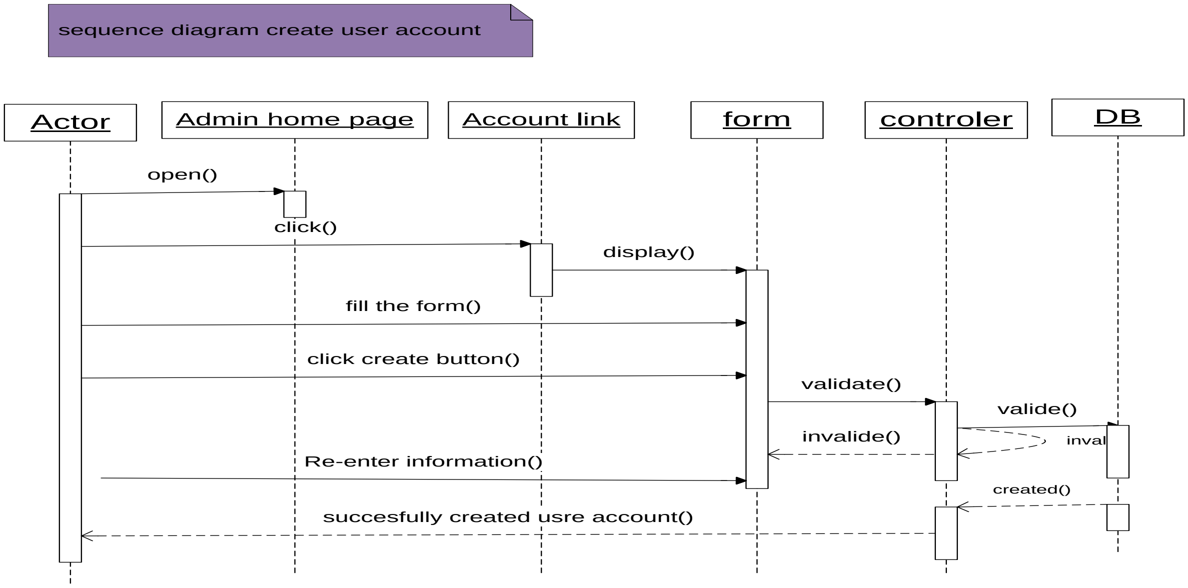


Figure 12: sequence diagram create account

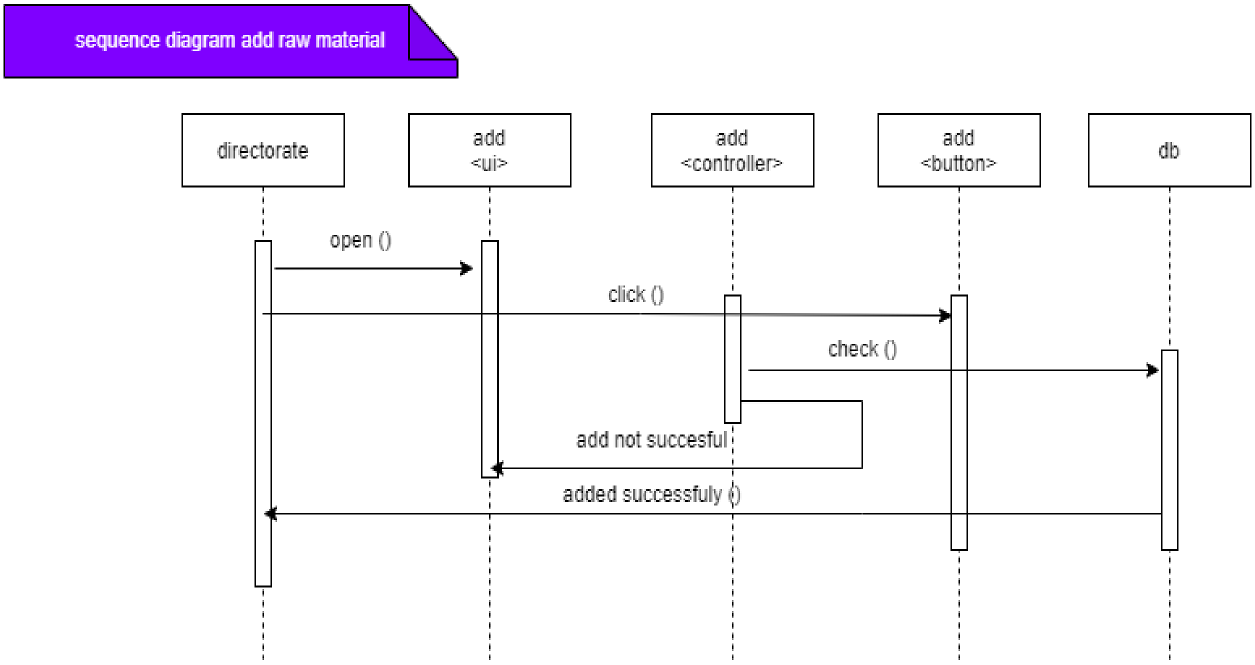


Figure 13: sequence diagram add raw material

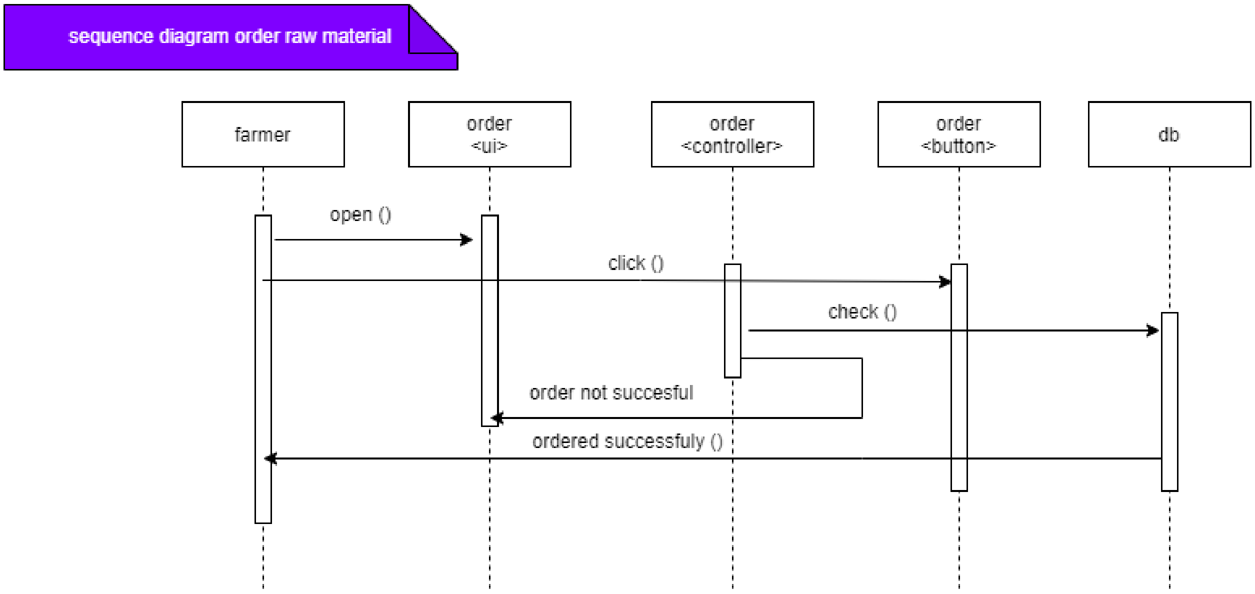


Figure 14: sequence diagram order raw material

2.3.8 *Analysis Class Model*

Analysis Class model/diagram is static model that shows the classes and the relationships among classes that remain constant over the time. Class is the main building block of class diagram, which stores and manages information in the system.

The figure below shows analysis class model of the system.

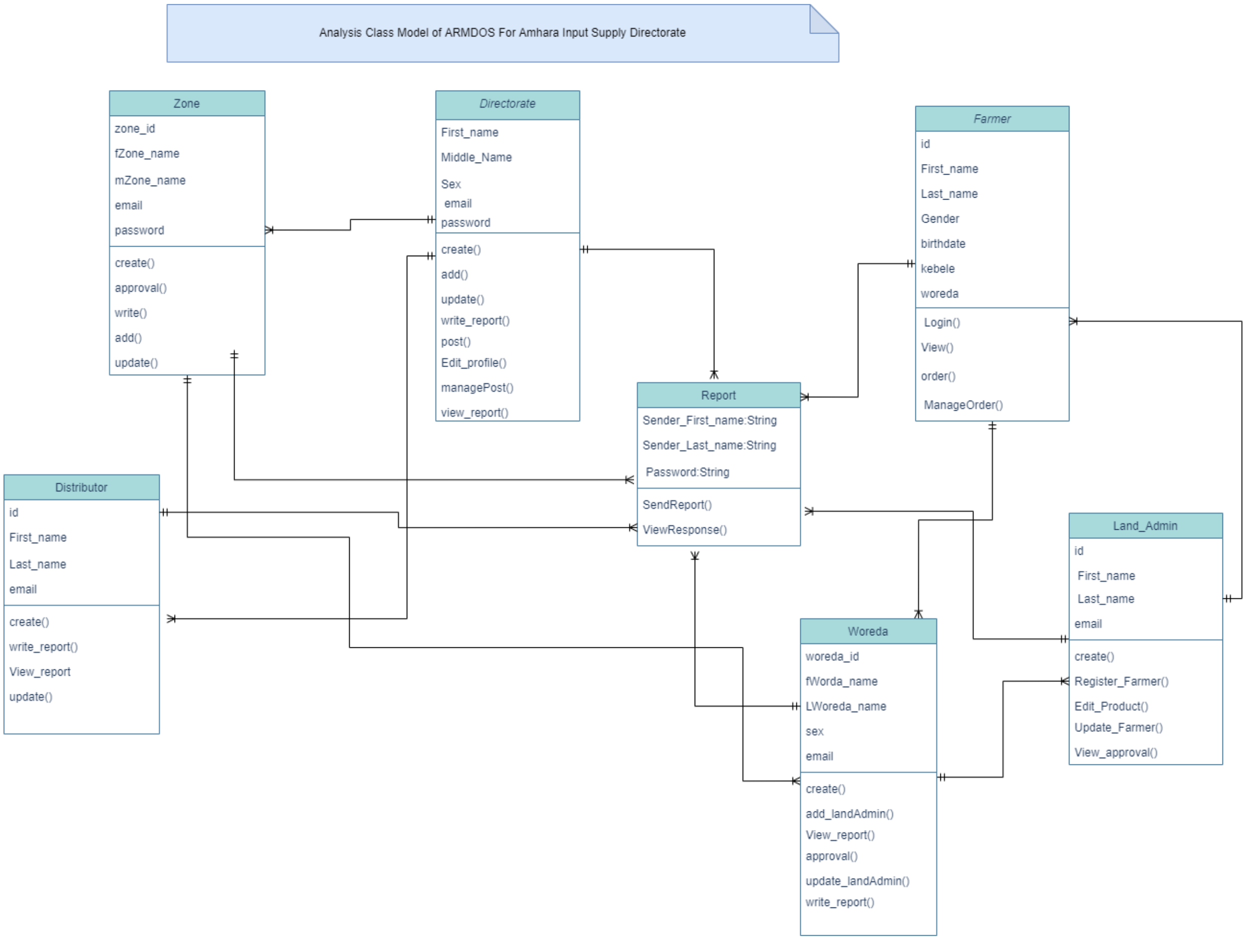


Figure 15: analysis class model

2.3.9 *Logic model*

Pseudo code is a simple way of writing programming code in English. Pseudo code is not actual programming language. It uses short phrases to write code for programs before you actually create it in a specific language. Once you know what the program is about and how it will function, then you can use pseudo code to create statements to achieve the required results for the program.

ID: AL 01

Name: Pseudo code **for Login**

```
Login ()
Begin
{
Enter user name;
Enter password;
Connect to database
If (user name&&password==true)
Display home page;
Else
Display error massage;
}
End
```

ID: AL 02

Name: Pseudo code **for Create Account**

```
Create user account ()
Begin
{
Enter user information;
Validate input
Connect to Database
If (information==true)
Insert to Database;
Display success massage;
Else
Display error message
}
End
```

ID: AL 03

Name: Pseudo code **for add raw material**

```
Add material ()
Begin
{
Enter material information;
Connect to database
If (material information==true)
Display message “successfully added”;
Else
Display error massage;
}
End
```

ID: AL 04

Name: Pseudo code **for Remove notice**

```
Remove notice ()
     Begin
      {
      Login ()
      Enter a key word to search the notice;
      Check if the key word is valid;
      If (keyword==true) {
            Click on delete button;
            If (delete success) {
                  Message 'deleted';
            }
            Else {
                  Message 'not deleted' try again;
            }
            }
         Else
```

```
            {
              Message 'please enter valid key word';
            }
            }
     End
```

2.4 Nonfunctional requirement

Non-functional requirements are requirement that can support and give more quality for the system. The proposed system has the following Non-Functional Requirements to achieve its functionality.

The following table shows nonfunctional requirement of the system.

ID	NFR-1
Source	user
Requirement	Usability
Description	The system is easy to learn and operate. The User interface for this system will be simple and clear. Web based distance learning services are easy to gain and use i.e. the service doesn’t require special training.
Priority	Medium

Table NFR usability

ID	NFR-2
Source	user
Requirement	Availability
Description	This system is available in everywhere (where internet/intranet service reach) and at all time for those who have access to use the system.
Priority	Medium

Table NFR availability

ID	NFR-3
Source	user
Requirement	Performance
Description	The system will have good performance i.e., fast response time and optimal workload.
Priority	Low

Table NFR performance

ID	NFR-4
Source	Developer
Requirement	Security
Description	Use very strong user name and password in order to secure the system. And also encrypts user's password on database. So it is designed to be very secure by providing a login feature which authenticates the user by means of a user name and password with which user will be able to login to his/her respective pages and use the system as required.
Priority	High

Table NFR security

ID	NFR-3
Source	Developer
Requirement	Documentation
Description	The system contains the required documents needed to implement the project.
Priority	Low

Table NFR documentation

2.5 System Requirement

2.5.1 Hardware requirements

Hardware requirements needed for the system:

- ✓ Processor speed=3.0 GHZ and above
- ✓ RAM=4 GB and above
- ✓ Hard disk space=250GB and above

2.5.1.1 Software requirements

Software requirements needed for the system:

- ✓ Web server: Apache HTTP server
- ✓ Database: MongoDB is a NoSQL Server
- ✓ Operating system: Windows (7, 8, 10), Linux.

✓ Client-side application (Browser): currently available/functional browser (Google chrome).

✓

2.6 Key abstraction with CRC analysis

A Class Responsibility Collaborator (CRC) model is a collection of standard index cards that have been divided into three sections. They are class, responsibility and collaboration. A class represents a collection of similar objects, a responsibility is something that a class knows or does, and a collaborator is another class that a class interacts with to fulfill its responsibilities.

The table shows the CRC of a class.

Register Farmer The roll is registere farmer	
Register farmer list and information	farmer woreda land admin

Write Report	
Write a Report what it does?	land admin farmer distributor zone woreda

Send raw material information

Writing and sending of raw material information	land admin directorate distributor

Approval	
Approve or disprove the request	farmer directorate woreda zone

Create account	
This is giving privilege and put restriction for user	Directorate Zone Distributor Woreda Land admin Farmer

Table CRC of a class

2.6.1 Identifying change cases

I. There might be some changes on the class and diagram.

II. For the future we will be add current information related to input/raw material

III. Authentication and authorization may be changed to support more features like email accounts.

IV. The UI/UX that we designed should change and continuously evolve to achieve more eye catching and smooth experience.

V. Multiple language support maybe added.

VI. We will be develop mobile app to the system

VII. If there is time and the features are feasible, we can add the above features to our system

Chapter 3: System Design

Systems design is the process of defining elements of a system like architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or an organization.

3.1 Architectural Design

Systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can comprise system components, the expand systems developed, that will work together to implement the overall system.

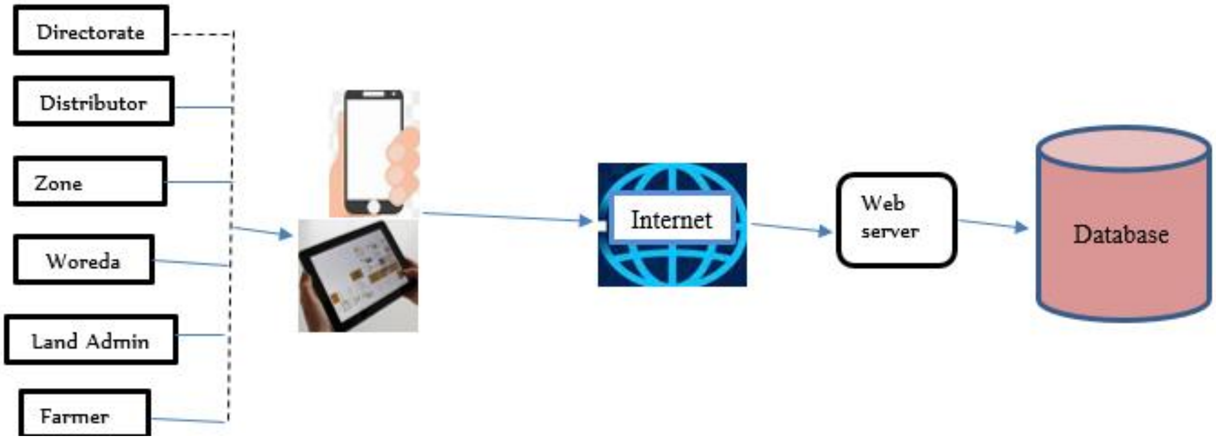


Figure 16:Architecture Design

3.1.1 Component modeling

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams. It does not describe the functionality of the system but it describes the components used to make those functionalities.

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components or software systems.

The figure below shows the component diagram of the system.

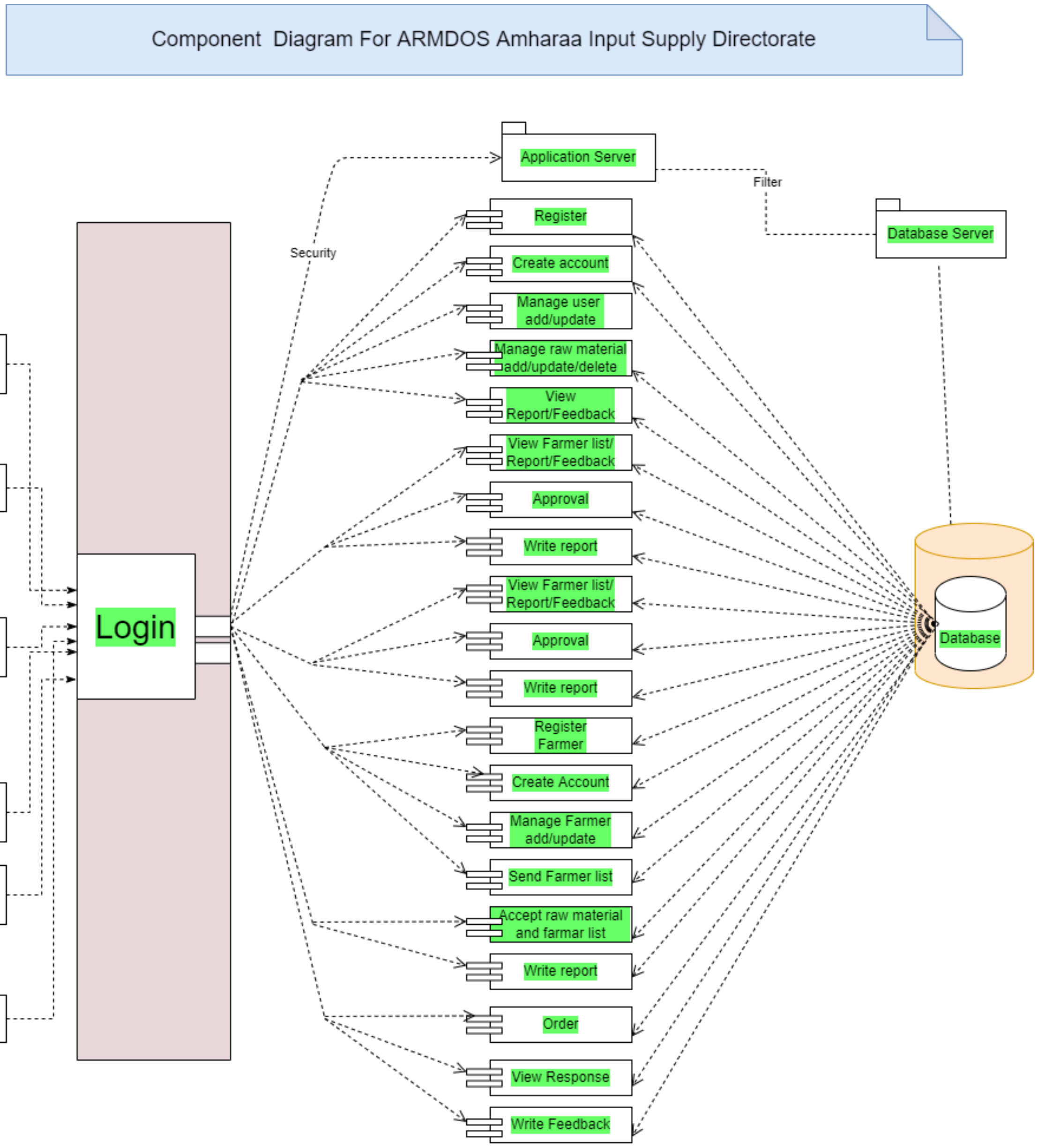


Figure 17: component diagram

3.1.2 *Deployment Modeling*

Deployment diagram is structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Artifacts represent concrete elements in the physical world that are the result of a development process.

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes.

The figure below shows deployment diagram of the system.

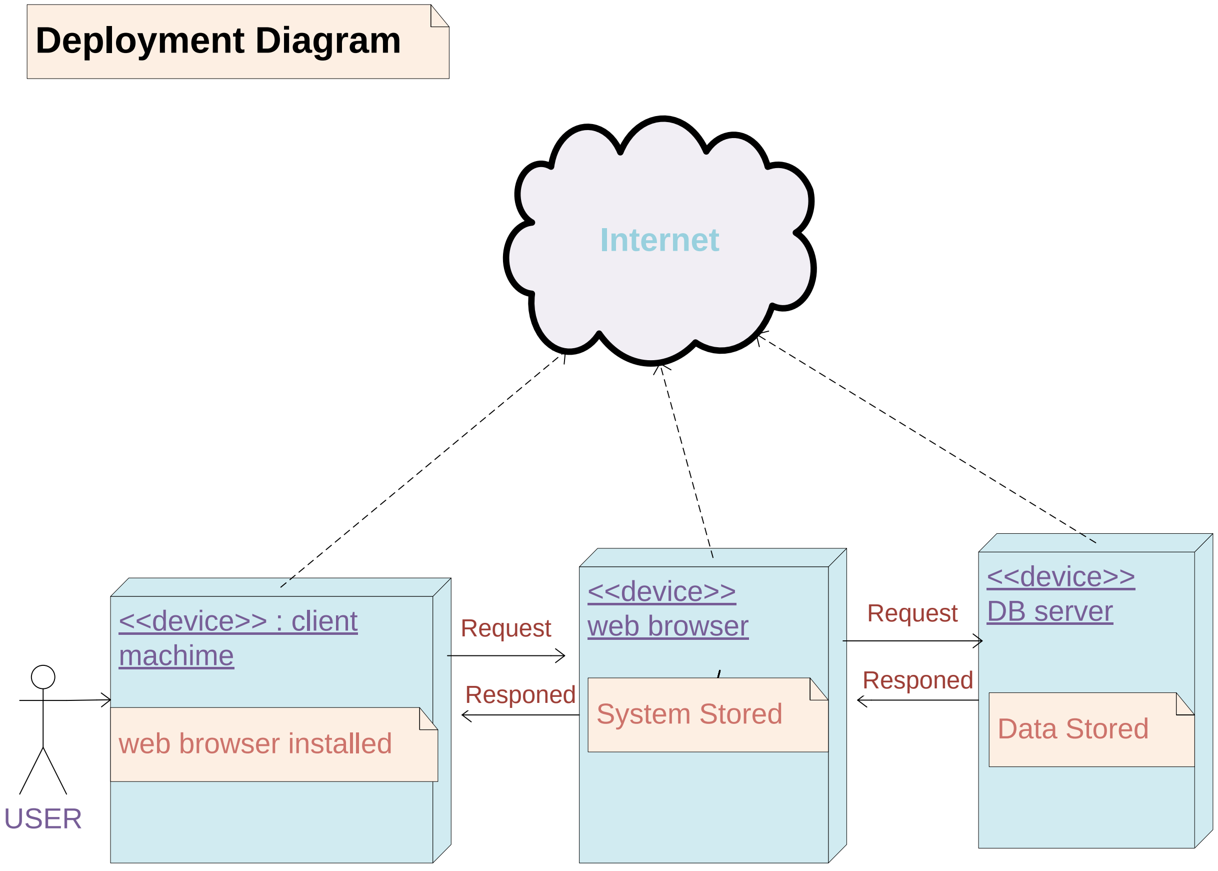


Figure 18: deployment diagram

3.2 Detail Design

3.2.1 Design class model

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP). The concept is several years old but has been refined as OOP modeling paradigms have evolved.

In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.

The figure below shows design class diagram of the system.

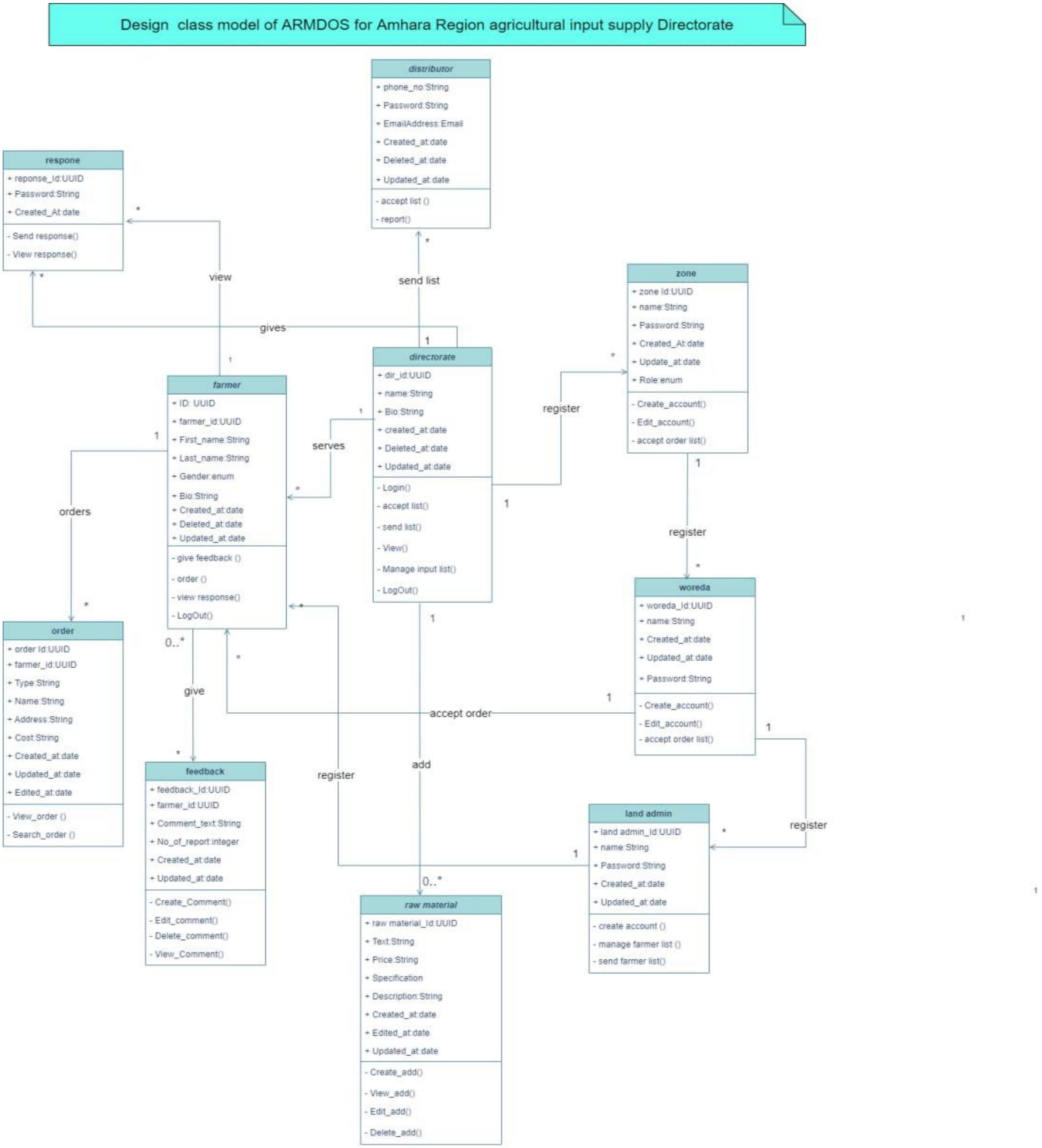


Figure 19: design class model diagram

3.2.2 Persistent model

A persistent diagram in software engineering is a type of diagram that represents the persistent data in a software system. It shows the relationships between the different data entities and how they are stored in a database or other forms of data storage.

Designing of a Persistence Model can be accomplished through model-driven design or database modeling.

The figure below shows Persistent model of the system.

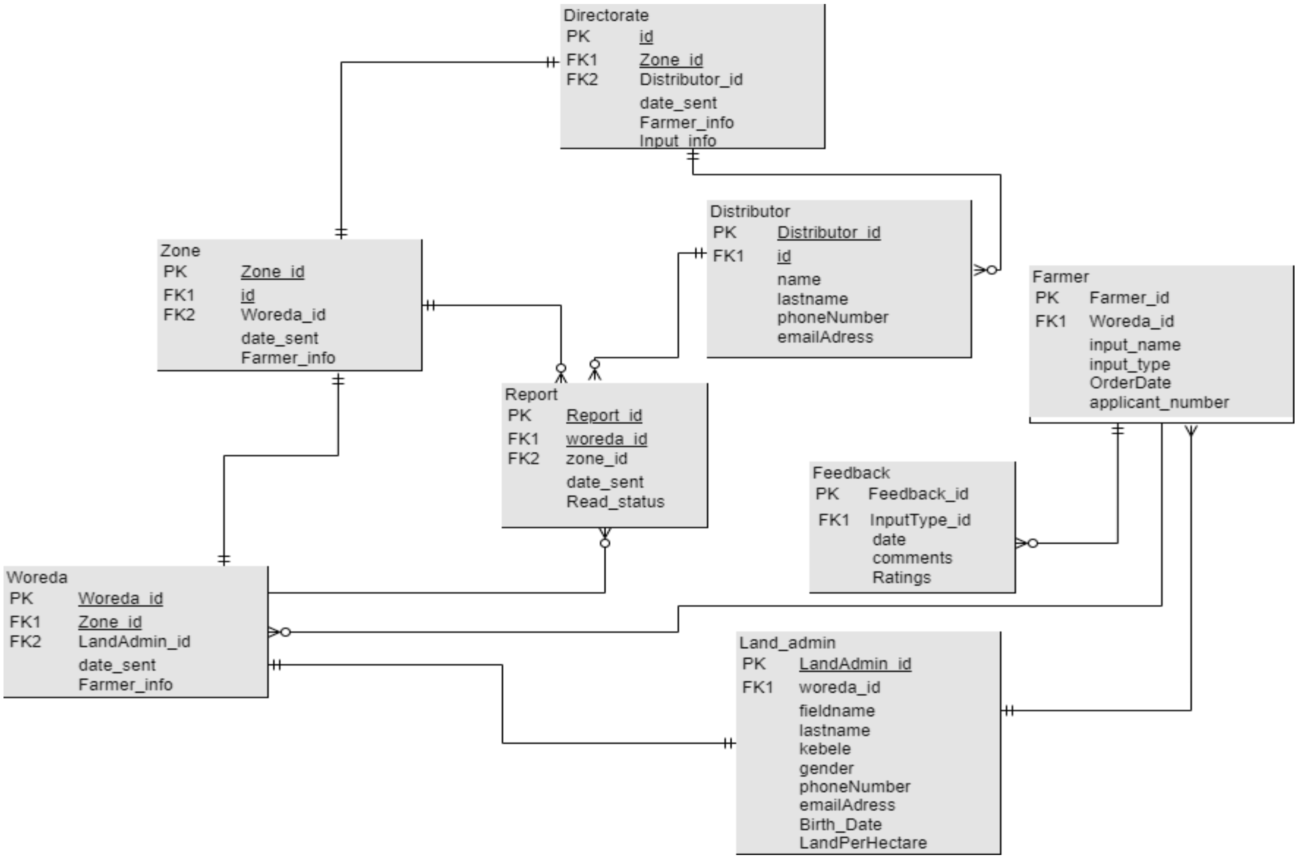


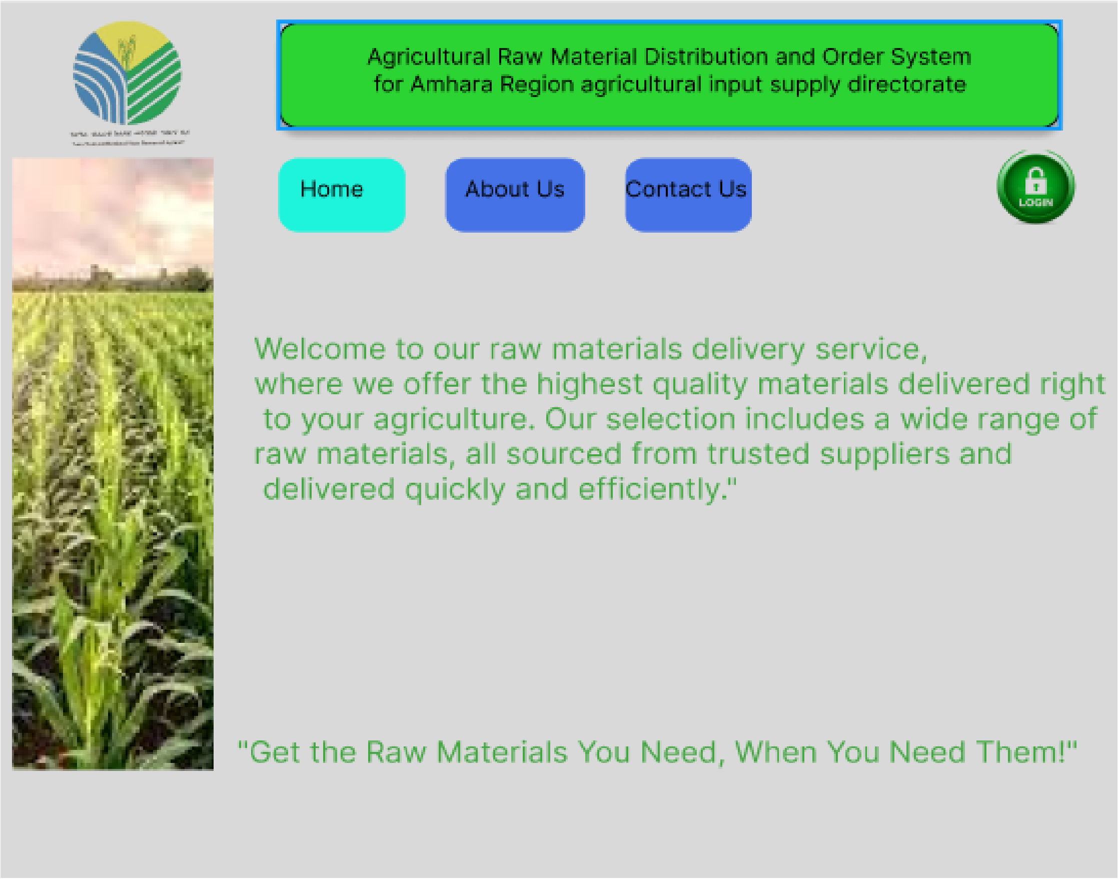
Figure 20: Persistent Model Diagram

3.3 User Interface Design

User interface design (UI design) refers to the design of various types of software and hardware interfaces through which users interact with computers and other technologies.

User interface design (UI) or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goal.

Here are some of the user interfaces.



Agricultural Raw Material Distribution and Order System
for Amhara Region agricultural input supply directorate

Welcome to our raw materials delivery service,
where we offer the highest quality materials delivered right to your agriculture. Our selection includes a wide range of raw materials, all sourced from trusted suppliers and delivered quickly and efficiently."

"Get the Raw Materials You Need, When You Need Them!"

Figure 21: Home Page UI

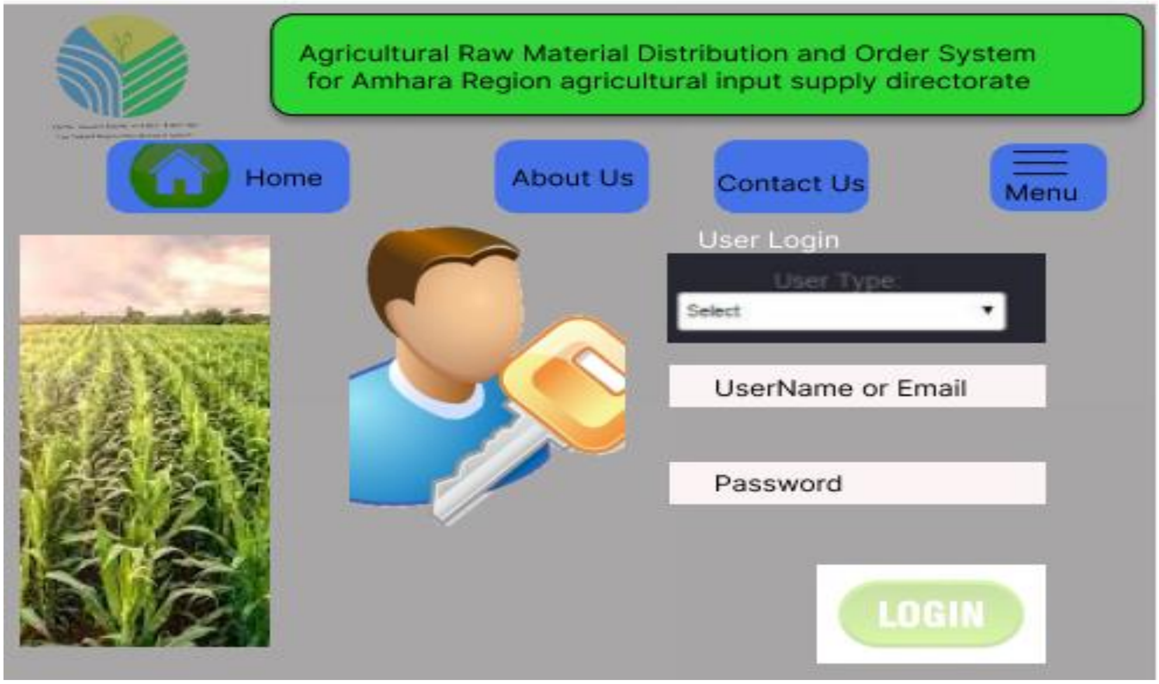

Agricultural Raw Material Distribution and Order System
for Amhara Region agricultural input supply directorate
Home
About Us
Contact Us
Menu
User Login
User Type:
Select
UserName or Email
Password
LOGIN

Figure 22: Login UI

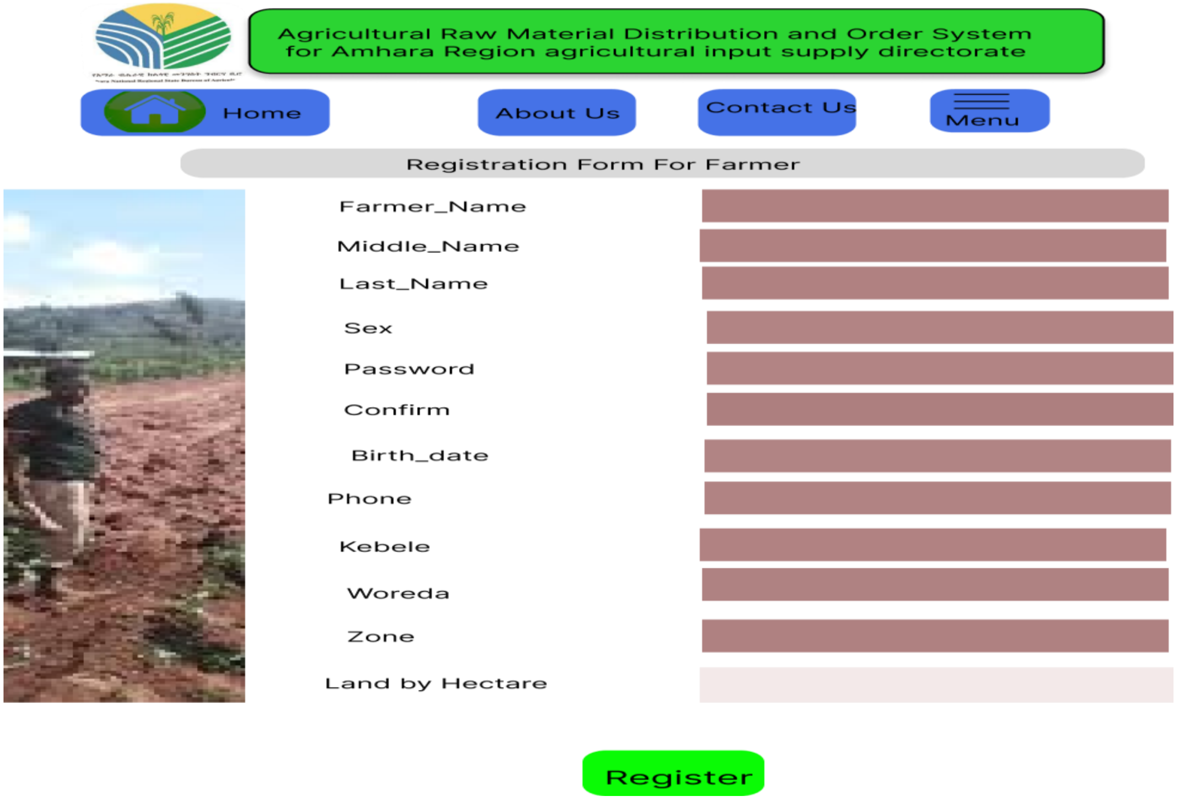
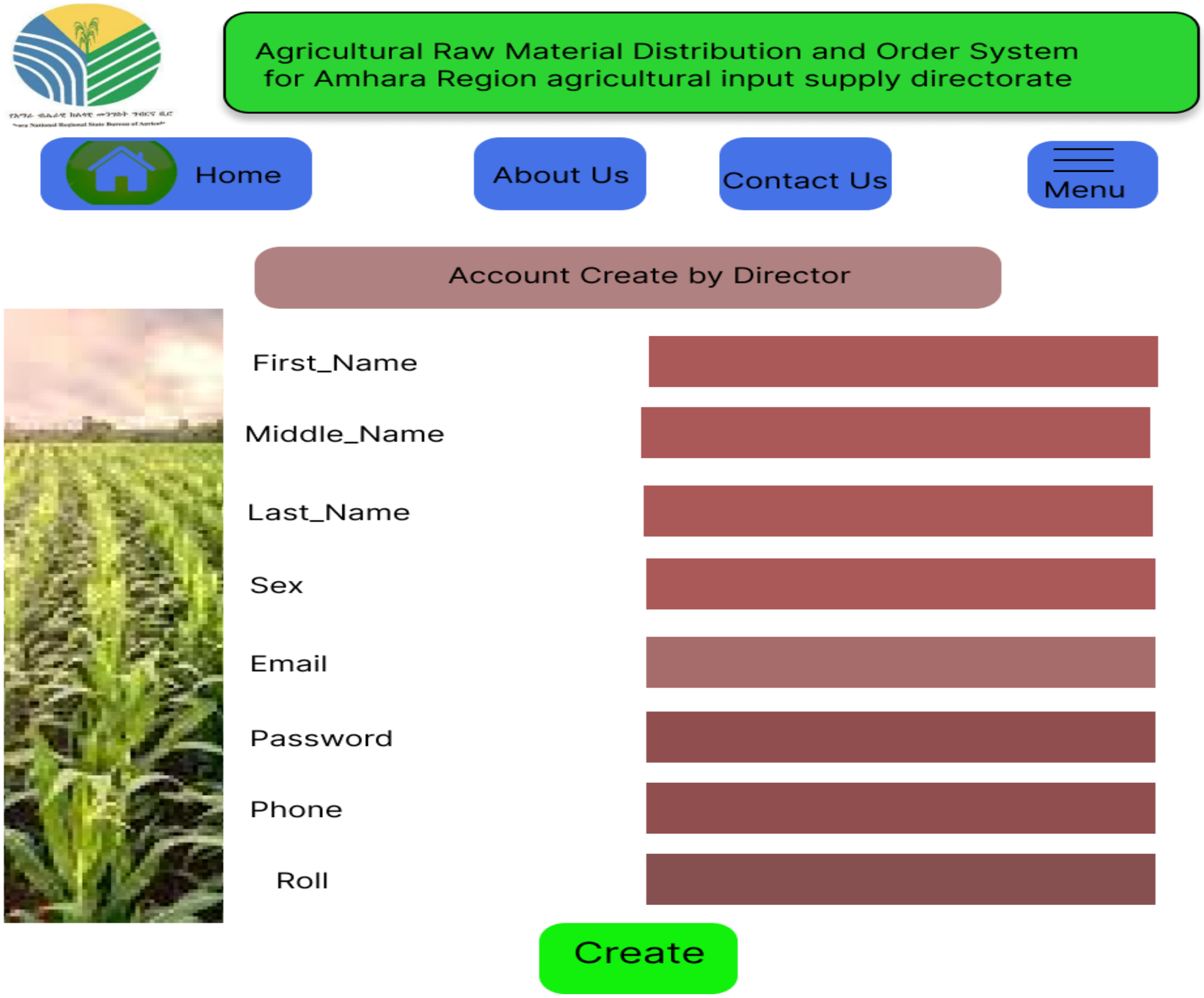
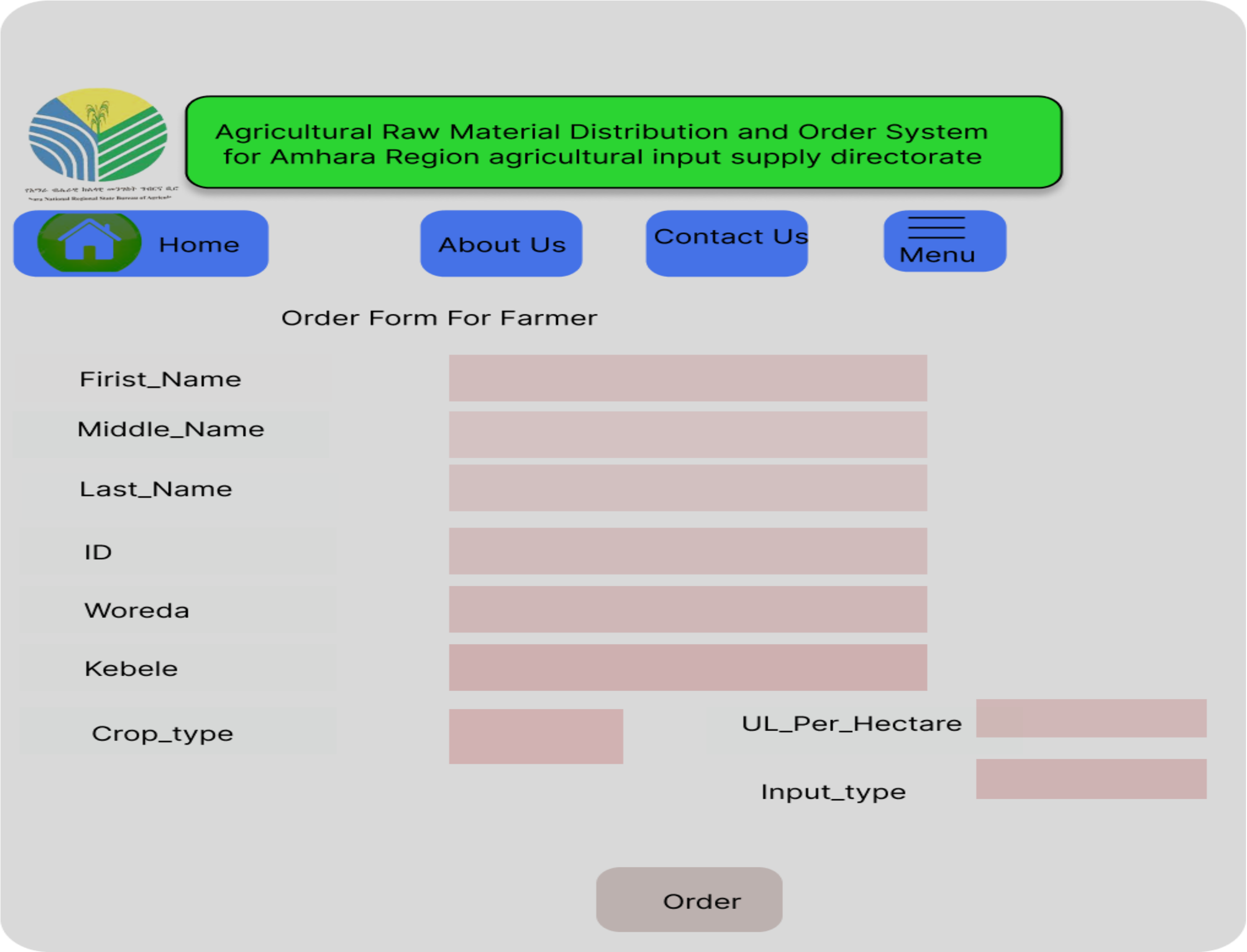

Agricultural Raw Material Distribution and Order System
for Amhara Region agricultural input supply directorate
Home
About Us
Contact Us
Menu
Registration Form For Farmer
Farmer_Name
Middle_Name
Last_Name
Sex
Password
Confirm
Birth_date
Phone
Kebele
Woreda
Zone
Land by Hectare
Register

Figure 23: Farmer Registration UI



Agricultural Raw Material Distribution and Order System for Amhara Region agricultural input supply directorate

Figure 24: Account UI



Agricultural Raw Material Distribution and Order System
for Amhara Region agricultural input supply directorate

Order Form For Farmer

Firist_Name

Middle_Name

Last_Name

ID

Woreda

Kebele

Crop_type

UL_Per_Hectare

Input_type

Figure 25: Input Order UI

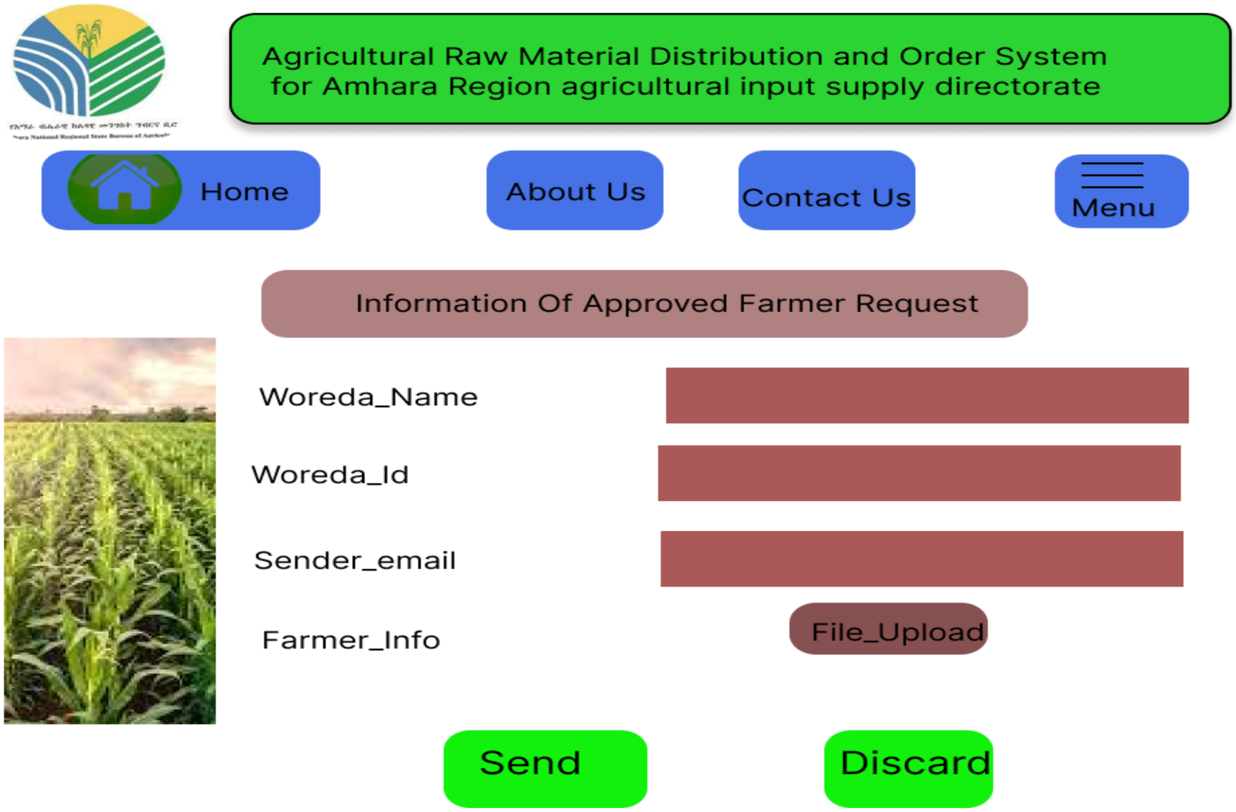
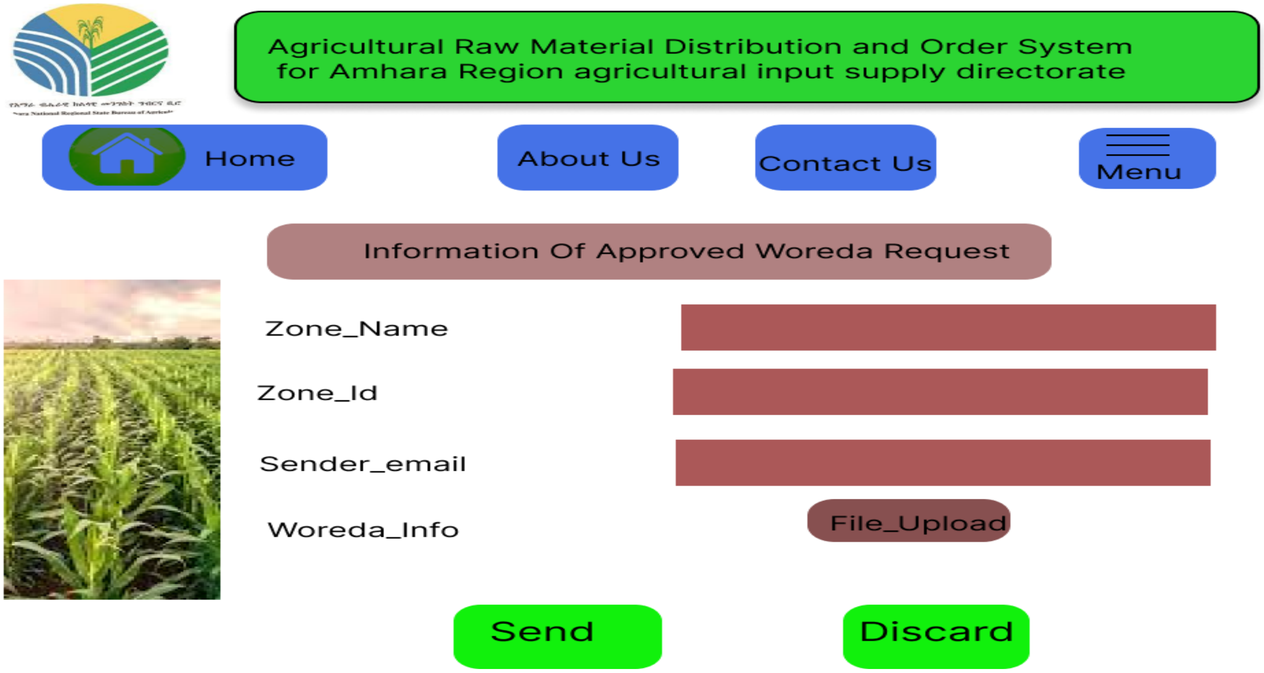
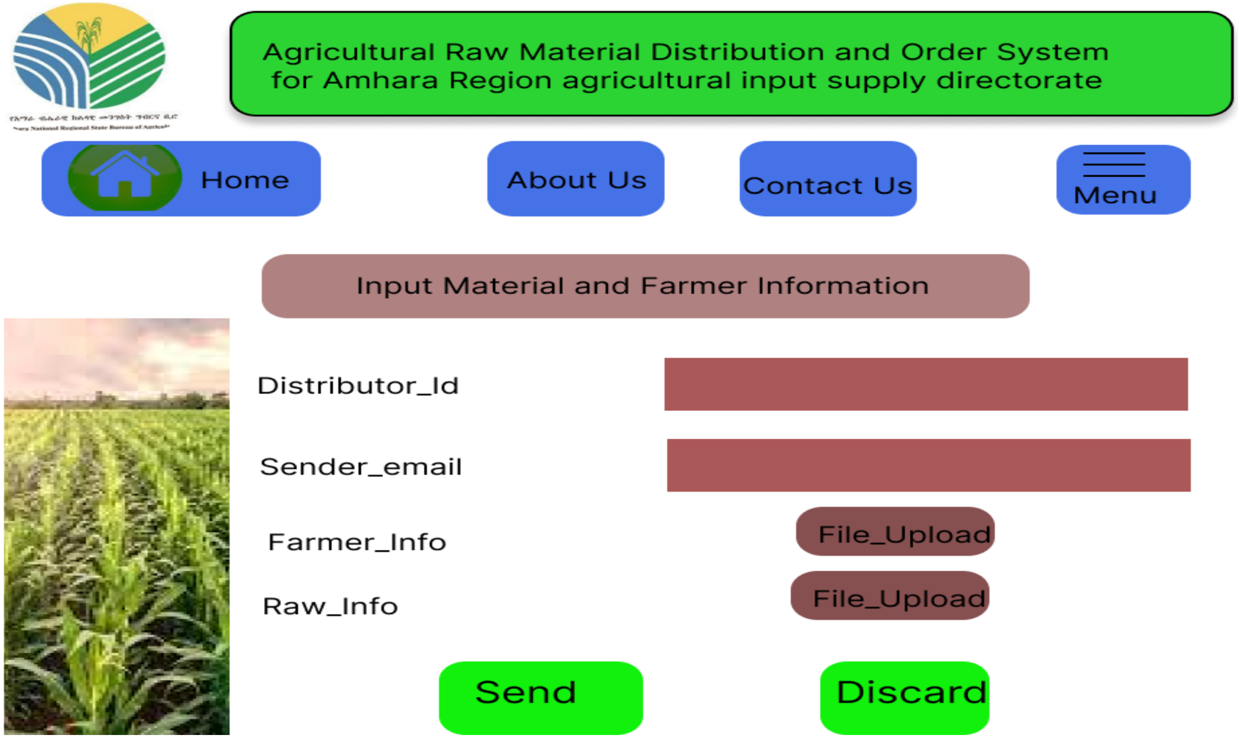

Agricultural Raw Material Distribution and Order System
for Amhara Region agricultural input supply directorate
Home
About Us
Contact Us
Menu
Information Of Approved Farmer Request
Woreda_Name
Woreda_Id
Sender_email
Farmer_Info
File_Upload
Send
Discard

Figure 26: Approval Farmer list UI



Agricultural Raw Material Distribution and Order System for Amhara Region agricultural input supply directorate

Figure 27 : Approval Woreda list UI



Agricultural Raw Material Distribution and Order System for Amhara Region agricultural input supply directorate

Figure 28: Input List_info UI

3.4 Access control and security

Access controls are security features that control how users and systems communicate and interact with other systems and resources. Access is the flow of information between a subject and an object. It is all about the act of ensuring that an authenticated user accesses only what they are authorized to. In the systems, different actors have access to different functionality and data. The project is divided into five modules namely,

1. Land Admin Module

The land admin is the person who is responsible for security issue of the system. He has the authority of manage and control user accounts. He can create new user account, deactivate and activate user account.

2. Directorate Module

Directorate is the entity who is responsible for inserting information into the database like order result, uploading raw material to the farmers.it has also access grant to view information in the system. To do this he/she must have his/her own account user name and password.

3. Distributor Module

A distributor is a user who has access grant to use the system for distributing raw material from the system. it has also the right to view information's about the directorate which concern them. To do this he/she must have his/her own account user name and password.

4. Farmer module

Farmer is the person who is responsible for ordering raw materials. To do this he/she must have his/her own account user name and password.

5. Zone and woreda module

This are the stakeholders which have the authority of approval and reporting. They have the right to approve the current work and report to their respective upper level authority.

References

What Is The MERN Stack? Introduction & Examples. (n.d.). MongoDB. https://www.mongodb.com/mern-stack

Caching a MongoDB Database with Redis — SitePoint. (n.d.). Caching a MongoDB Database With Redis — SitePoint. https://www.sitepoint.com/caching-a-mongodb-database-with-redis/

www.amhboard.gov