Software Requirement Specifications

FLITS

Version: [1.0]

Project Code	
Supervisor	Sir Shoaib Raza
Co Supervisor	
Project Team	Ibrahim Ali (19K-0289) Ibrahim Ali Asghar (19K-1443) Subhan Farid (19K-1380)
Submission Date	November 25, 2022

Document History

Version	Name of Person	Date	Description of change
1.0	Ibrahim Ali	November 15, 2022	*Document Created*
1.0	Ibrahim Ali Asghar	November 16, 2022	*Added External Interface Requirement*
1.0	Ibrahim Ali Asghar	November 16, 2022	*Added Introduction*
1.0	Subhan Farid	November 19, 2022	*Added Functional Requirements*
1.0	Subhan Farid	November 19, 2022	*Added Use Cases*
1.0	Ibrahim Ali	November 19, 2022	*Overall Project Description*
1.0	Ibrahim AlI	November 23, 2022	*Added Non Functional Requirements*

Distribution List

[Following table will contain list of people whom the document will be distributed after every sign-off]

Name	Role
Sir Shoaib Raza	Supervisor

Document Sign-Off

Version	Sign-off Authority	Sign-off Date

Table of Contents

1. Introduction

1.1. Purpose of Document

This document is a definition of software requirements to facilitate the Pakistani Farmers by providing a Task Scheduling and Management System called “Farming Livestock Inventory and Task Scheduling (FLITS) “. This document will present the functional, non-functional, and design constraint requirements for the system to be developed. Use case models and descriptions are included along with class diagrams to help model and specify the functional requirements and specifications of the system.

1.2. Intended Audience

The potential audiences for this document are design and development team of the FLITS in order to specify software designs. Test team will utilize this software requirements specification document to define test scenarios accordingly.

The intended audience are the farmers of Pakistani Agricultural Industry who will be using this application.

1.3. Abbreviations

FLITS: Farming Livestock Inventory and Task Scheduling

CMMS: Computerized Maintenance Management System

1.4. Document Convention

Font: Arial

Font-size:

Heading: 16px

Subheading: 14px

Description: 12px

2. Overall System Description

2.1. Project Background

Agriculture and farming industry plays a vital role in Pakistan's economy. To keep consistently growing in this industry, we needed a system through which we could manage equipment, workers, and simplify regulatory compliance. Traditionally Pakistani farmers had been working unsystematically and some of the farmers who were doing it systematically relied on paper-based work. Work Orders CMMS (Computerized Maintenance Management System) solution streamlines the management of agricultural and farm maintenance routines. Currently Pakistani Farming industry is not digitally aware due to which the upcoming generations of agricultural families are not interested in doing farming and are trying to find opportunities in different fields of work. FLITS will not only ease the farm management and operations but will also create digital awareness in the upcoming generations of farmers and help them look at it as a business model. We will be creating a system where landlords can monitor the progress of their farms and are responsible to assign supervisor to each of their farm. Supervisor will manage farm operations and assign tasks to workers. Workers will be responsible to update their daily activities

2.2. Project Scope

FLITS is a mobile application that will digitalize farm management operations. This includes farm customization, task scheduling, labor management and evaluation, inventory management, crop progress, etc.

2.3. Not In Scope

Vendors and Service Providers for buying of resources, equipment and animals is not in scope.

2.4. Project Objectives

Our main objectives are:

- Make Pakistani farmers digitally aware
- Digitalize Farm Management Operations

2.5. Stakeholders

- Farm Owners (Landlords)
- Farm Supervisor
- Farm Worker
- App Developer
- Database Developer

2.6. Operating Environment

The application will operate on a mobile application for operational/running purposes.

2.7. System Constraints

- **Software constraints**
  The software is built on NodeJS (server-side JavaScript) framework Which is NestJS and uses MySQL as a Database. And for deployment it requires a cloud service to host for it to be available on Android smartphones.

- **Hardware constraints**
  The application will work on any smartphone which has an internet connection.
- **Cultural constraints**
  The application supports English language.
- **Legal constraints**
  The software cannot be used by any user who has not registered himself.
- **Environmental constraints**
  The software need not be installed.
- **User constraints:** Application is developed for Farm Owners, Farm Supervisors, and Farm Workers
- **Off the shelf components**
  No off the shelf components used.

2.8. Assumptions & Dependencies

User interface and some functionalities can change during the development process of project,. and also new functionalities can be added which is able to change the dependent system requirements.

3. External Interface Requirements

3.1. Hardware Interfaces

Since FLITS will be a mobile application that must run over the internet, the following hardware is necessary for this project: Smartphone, WIFI to run the application.

3.2. Software Interfaces

Database management system is required software product for Carpool system because all data about system for example user and route information must be stored in database for later use and system functionality.

- Operating System: Android
- Database: MySQL
- Libraries and Frameworks: Flutter, Node.js, Nest.js

3.3. Communications Interfaces

There is communication between Owner and Supervisor, and Worker and Supervisor.

4. Functional Requirements

4.1. Functional Hierarchy

The functional requirements of the project include the following:

- Users to be registered on the application.
- Users should be authenticated in order to be able to use the application.
- Farm Owner can Add Farms, Assign Supervisor, Add Fields, Add Plantings, Add Products, Add Customer, Manage Transactions.
- Farm Supervisor can Add and View Workers, View Plantings, View Grow Locations, Add and View Resources, Add and View Equipment, Add and View Tasks, Schedule and Assign Tasks , Add and View Animals.
- Farm Workers can mark their attendance, update task status.

4.2. Use Cases

4.2.1. Farm Owner

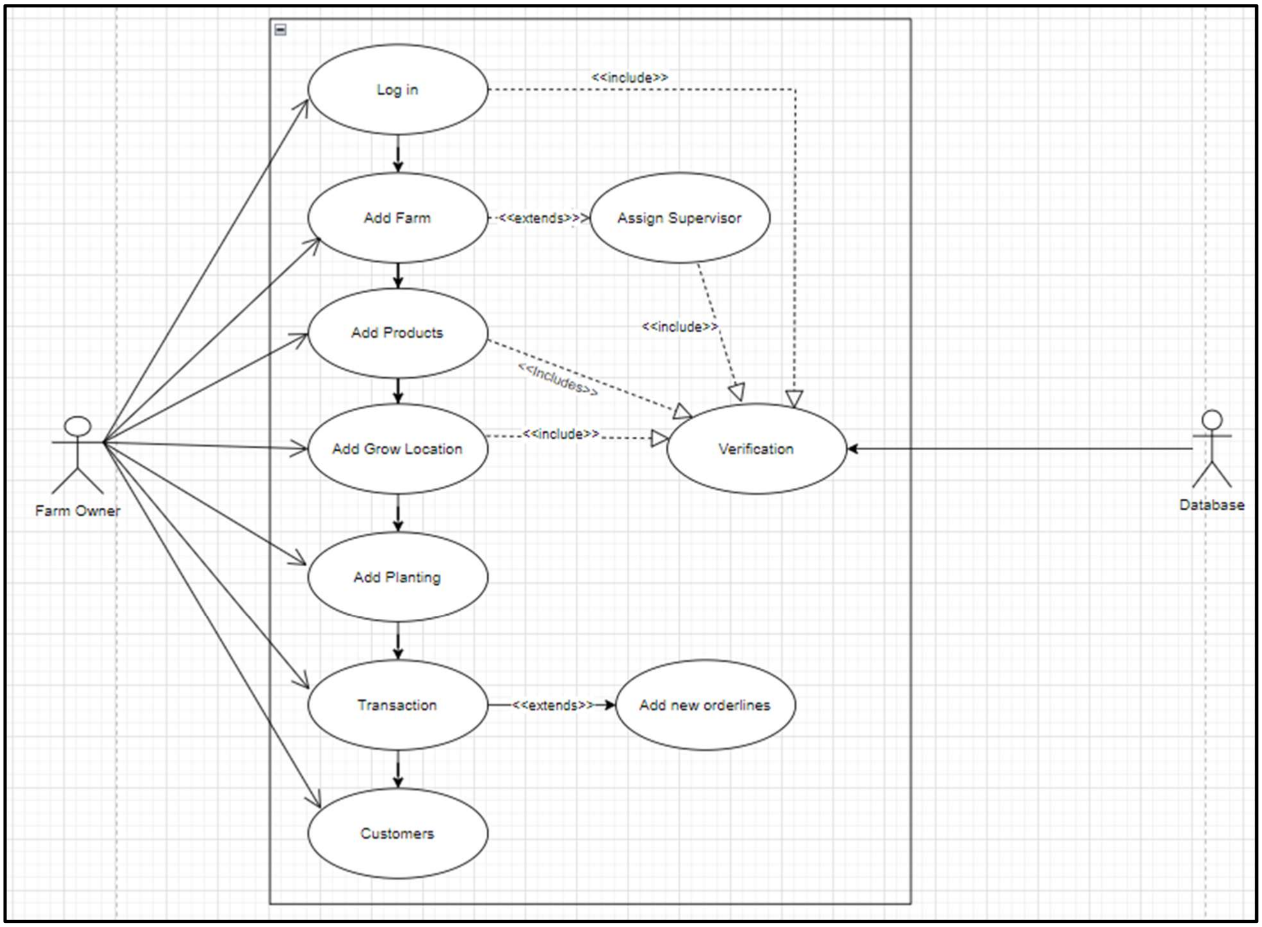

Figure 1: Farm Owner Use Case

Table 1: Add Farm Use Case Description

Use Case Description	
Use Case name:	ADD FARM
Use Case Description: To add new farm	
Primary actor: Farm Owner	**Other actors:** Database
Stakeholders: Farm Owner	
Relationships □ **Includes:** Verify	
Flow of Events: **1. Farm owner will click on add farm.** **2.** He will add the details of the Farms. **3. He will click on assign supervisor.** **4. Database will verify the details and check constraints.**	
Post-conditions: A farm with it's personal supervisor will be added.	

Table 2: Add Product Use Case Description

Use Case Description	
Use Case name:	ADD PRODUCTS
Use Case Description: To add products in the farm	
Primary actor: Farm Owner	**Other actors:** Database
Relationships : **Includes:** Verify	
Stakeholders: Farm Owner	
Flow of Events: **1. Farm Owner will click on add products.** **2.** He will add the name of the product and id will be attached automatically. **3. product will be added and verified by the database.**	
Post-conditions: Farm owner can view the products he is growing on his/her farm.	

Table 3: Add Grow Location Use Case Description

Use Case Description	
Use Case name:	ADD GROW LOCATION
Use Case Description: New fields will be added with proper details.	
Primary actor: Farm Owner	**Other actors:** Database
Stakeholders: Farm Owner	
Relationships □ **Includes:** Verify	
Flow of Events: **1. Farm Owner will click on add Grow location.** **2.** He will add new fields and provide all necessary details required for the field. 3. Database will verify the constraints for each data entry.	
Post-conditions: Details of field will be added/updated in database.	

Table 4: Add Planting Use Case Description

Use Case Description	
Use Case name:	ADD PLANTING
Use Case Description: To add planting details on field.	
Primary actor: Farm Owner	**Other actors: -**
Stakeholders: Farm Owner	
Relationships □	
Flow of Events: **1. Actor will click on Add Planting.** **2.** He will add details of planting which includes methods and other related requirements. 3. Planting details will be added in the database.	
Post-conditions: Farm Owner will be able to view the planting details in his/her field.	

Table 5: Transaction Use Case Description

Use Case Description	
Use Case name:	TRANSACTION
Use Case Description: To view orderliness and add/view transactions.	
Primary actor: Farm Owner	**Other actors:** Database
Stakeholders: Farm Owner	
Relationships □ **Includes: -**	
Flow of Events: **1. Farm Owner will click on view transactions.** **2.** He/she can add new Order lines.	
Post-conditions: The transaction details will be stored in database.	

Table 6: Add Customer Use Case Description

Use Case Description	
Use Case name:	ADD CUSTOMER
Use Case Description: To view and add new customers.	
Primary actor: Farm Owner	**Other actors:**
Stakeholders: Farm Owner	
Relationships □	
Flow of Events: **1. Farm Owner will click on Customers.** **2.** He/She can view and add new customers.	
Post-conditions: Customers will be added in the database.	

4.2.2. Farm Supervisor

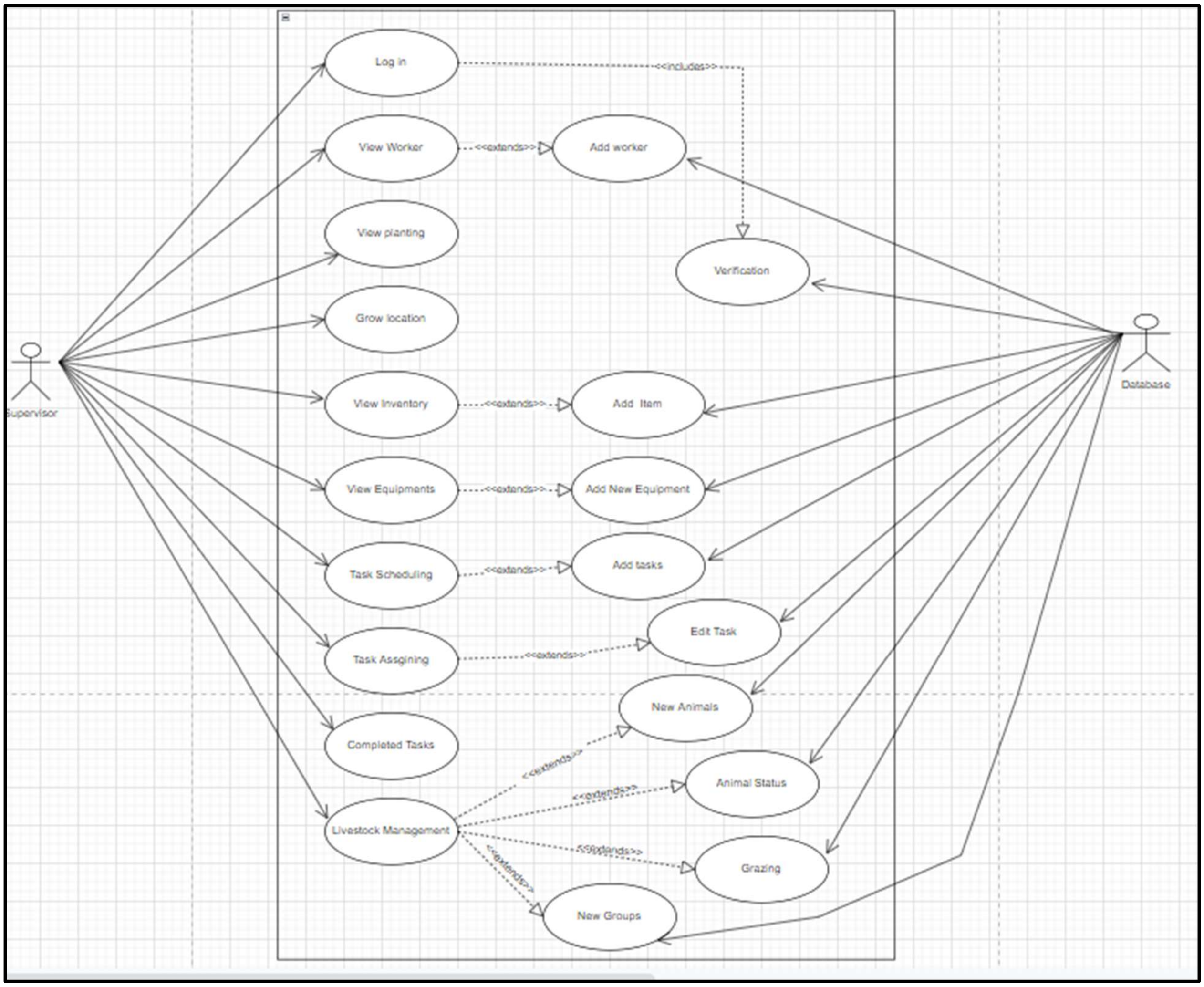

Figure 2: Farm Supervisor Use Case

Table 7: View Worker Use Case Description

Use Case Description	
Use Case name:	View Worker
Use Case Description: Supervisor can view and add workers in the farm.	
Primary actor: Supervisor	**Other actors:** Database
Stakeholders: Supervisor	
Relationships **Extends:** -Add Worker	
Flow of Events:	

1. Supervisor will click on view workers. **2.** He can see and edit the details of all the workers in the farm. 3. He can add new worker in the farm.
Post-conditions: A new worker will be added in the database.

Table 8:View Planting Use Case Description

Use Case Description	
Use Case name:	View Planting
Use Case Description: Display planting details to Supervisor.	
Primary actor: Supervisor	**Other actors:**
Stakeholders: Supervisor	
Relationships	
Flow of Events: 1. Actor will click on View planting. **2.** Crop name and grow locations will be displayed to the Supervisor.	

Table 9:Grow Location Use Case Description

Use Case Description	
Use Case name:	Grow Location
Use Case Description: Display the details of growing locations.	
Primary actor: Supervisor	**Other actors:**
Stakeholders: Supervisor.	
Relationships	
Flow of Events: • Supervisor will click on grow location. • details of specific location and growth stage will be displayed on the screen.	

Table 10:View Inventory Use Case Description

Use Case Description	
Use Case name:	View Inventory
Use Case Description: Supervisor can display and add new tools in the inventory.	

Primary actor: Supervisor	**Other actors:** Database
Stakeholders: Supervisor	
Relationships **Extends:** Add Item-	
Flow of Events: • Supervisor will click on View inventory. • All the products/tools in inventory will be displayed to supervisor. • Supervisor can add new item in inventory by clicking on "add item".	
Post-conditions: The details of new item added in the inventory will be saved in the database.	

Table 11:View Equipment Use Case Description

Use Case Description	
Use Case name:	View Equipment
Use Case Description: To display, edit and add new equipment.	
Primary actor: Supervisor	**Other actors:** Database
Stakeholders: Society	
Relationships **Extends:** Add New Equipment -	
Flow of Events: • Supervisor will click on view equipment. • List of equipment will be displayed to supervisor. • Supervisor can edit Equipment details. • Supervisor can add new equipment by clicking on "Add Equipment".	
Post-conditions: The new and updated details of equipment will be added/updated in the database.	

Table 12:Task Scheduling Use Case Description

Use Case Description	
Use Case name:	Task scheduling
Use Case Description: Tasks will be schedule with respect to name.	
Primary actor: Supervisor	**Other actors:** Database

Stakeholders: Supervisor	
Relationships **Extends:** Add Tasks -	
Flow of Events: • Supervisor will click on Task Scheduling. • He can view all scheduled tasks. • He can also add new task in the List. • He can also edit and delete task.	
Post-conditions: Once the task is scheduled, they can be assigned.	

Table 13:Task Assigning Use Case Description

Use Case Description Task Assigning.	
Use Case name:	Task Assigning.
Use Case Description: Task will be assigned to workers.	
Primary actor: Supervisor	**Other actors:** Database
Stakeholders: Supervisor	
Relationships **Extends:** -Edit Tasks	
Flow of Events: • Supervisor will click on Task Assigning. • He can select the schedule task he wants to assign. • He can select Edit Task in order to edit or re-Assign the task.	
Post-conditions: After the Task is Assign, he can view the Assign task on his screen**.**	

Table 14:Completed Task Use Case Description

Use Case Description	
Use Case name:	Completed Task
Use Case Description: To view Assign tasks which are completed.	
Primary actor: Supervisor	**Other actors:** Database
Stakeholders: Supervisor	

Flow of Events: 1.Supervisor will click on delete Completed Task. 2.List of Tasks which are completed will be display on Screen.	
.	
Use Case Description	
Use Case name:	New Animals.
Use Case Description: To view add new animals in the farm.	
Primary actor: Supervisor	**Other actors:** Database
Stakeholders: Supervisor.	
Relationship **s** **Extends:** -	
Flow of Events: 1.Supervisor will click on Livestock Management. 2.Then he select New animals. 3.He can view and add new animals in the farm.	
Post-conditions: After adding new animals, the updated list of the animals will be displayed to User.	

Table 15:Animal Status Use Case Description

Use Case Description	
Use Case name:	Animal Status
Use Case Description: To view and edit the status of animals	
Primary actor: Supervisor	**Other actors:** Database
Stakeholders: Supervisor	
Relationships **Extends:** -	
Flow of Events: 1.Supervisor will click on Animal status. 2. All animals status will be displayed to him. 3. Click on edit status, to Edit the current status of animal.	

Post-conditions: The status of the animal will be updated and store in database.

Table 16: Grazing Use Case Description

Use Case Description	
Use Case name:	Grazing
Use Case Description: To assign field locations to animal groups for grazing.	
Primary actor: Supervisor	**Other actors:** Database
Stakeholders: Supervisor	
Relationships	
Flow of Events: 1. Supervisor will click on Livestock Management. 2. Then he will select Grazing. 3. He will select specific fields and assign animals groups on that group.	
Post-conditions: Supervisor can view the fields and animals groups that are allocated for Grazing.	

Table 17:New Group Use Case Description

Use Case Description	
Use Case name:	New Group
Use Case Description: To add new Group for adding animals.	
Primary actor: Supervisor	**Other actors:** Database
Stakeholders: Supervisor	
Relation **ships** **Extends:** -	
Flow of Events: 1. Supervisor will click on Livestock Management. 2. Then he will select New Group. 3. He will enter the details of the group and select animals for that specific group. 4. He can edit the group as well.	

Post-conditions: New groups along with the animals it contains will be stored in Database.

4.2.3. Farm Worker

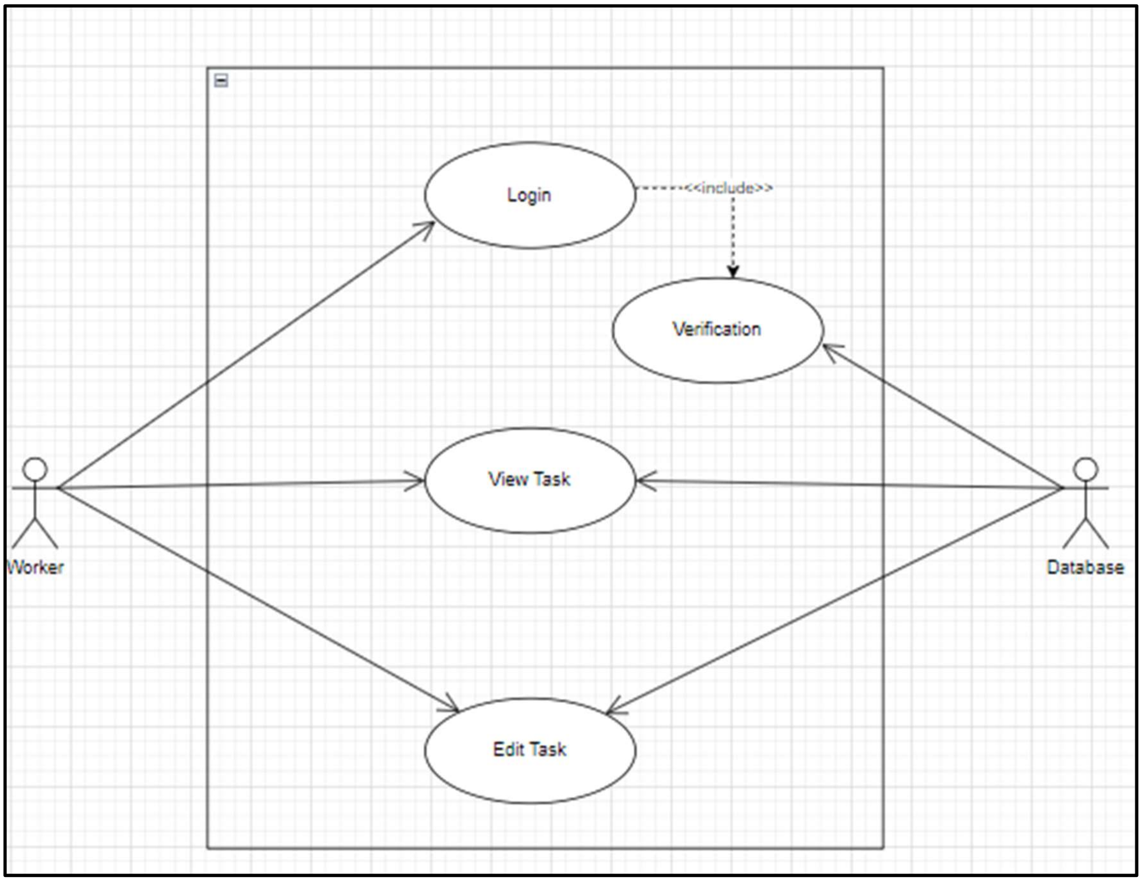

Figure 3: Farm Worker Use Case

Table 18:Login Use Case Description

Use Case Description	
Use Case name:	Login
Use Case Description: To login and use app	
Primary actor: Worker	**Other actors:** Database
Stakeholders: Worker	
Relationships □ **Includes:** Verify	

Flow of Events: **1. Worker will login to App.** **2.** Id and Password will be verified. **3. Re-enter ID and Password if incorrect.**
Post-conditions: Worker will be able to view tasks.

Table 19:View Task Use Case Description

Use Case Description	
Use Case name:	View Task
Use Case Description: To View and complete Task.	
Primary actor: Worker	**Other actors:** Database
Stakeholders: Worker	
Flow of Events: **1. Worker will click on View Task.** **2.** All pending tasks will be fetch from database and display on user screen. **3. Click on Complete Task once the task is done.**	
Post-conditions: Supervisor will be notified once the task is done/completed.	

Table 20:Edit Task Use Case Description

Use Case Description	
Use Case name:	Edit Task
Use Case Description: To edit the expected completion time for the task.	
Primary actor: Worker	**Other actors:** Database
Stakeholders: Worker	

Flow of Events:

1. Worker will click on edit tasks.

2. Task details will be load from database and displayed to worker.

3. Worker can change the expected time completion for the task.

4. Updated information will be stored in database.

Post-conditions:

Supervisor will be notified that task expected time is changed and all other dependent task will be delayed accordingly.

5. Non-functional Requirements

5.1. Performance Requirements

A proper internet connection is needed for the users using the application and the user interface of the website should be easy. The system shall be able to respond several users simultaneously. The system shall be able to keep user information of several users.

5.2. Safety Requirements

The system should be fast else the application will be down and the server files will not work properly, but as such it will not damage, lose or cause harm to the system or data.

5.3. Security Requirements

The system shouldn’t allow any unauthorized person to access the application, until he provides correct username and password.

5.4. User Documentation

- The application will be user-friendly and created in such way that the person having little or no knowledge of the application can use this; therefore, no user manual will be required.

6. References

1. Yang-yangZHENG, Tie-huiZHU, WeiJIA, "Does Internet use promote the adoption of agricultural technology?" ,"Journal of Integrative Agriculture", Vol 21, Issue 1, page 282-292, January 2022.
2. Hadi Balouei Jamkhaneh, Javad Khazaei Pool, Seyed Mohammad Sadegh Khaksar, S. Mohammad Arabzad, Reza Verij Kazemi,"Impacts of computerized maintenance management system and relevant supportive organizational factors on total productive maintenance ", ISSN: 1463-5771, 1 October 2018.
3. ZhiyangShen, SongkaiWang, Jean-PhilippeBoussemart, YuHao "Digital transition and green growth in chinese agriculture", "Technological Forecasting and Social Change", Vol: 181, Article id;121742, August 2022.
4. State Bank of Pakistan, "Pakistan GDP from Agriculture", "www.tradingeconomics.com", year 2022.
5. Janine Russell, "Farm Brite ", " https://www.farmbrite.com/ ".
6. Kizeo, "Kizeo Forms", " https://www.kizeo-forms.com/en/ ".
7. AbdulRehman, LuanJingdonga, AbbasAliChandio, ImranHussain "Livestock production and population census in Pakistan", Information Processing in Agriculture, volume 2, page 168-177, June 2018.

7. Appendices