Hamdard University

Department of Computing

Final Year Project

Farmware App

FYP-013/FL24

Software Requirements Specifications

Submitted by

Maryam Nadeem (2437-2021)

Muhammad Mujtaba (1985-2021)

Hamza Bin Asif (1961-2021)

Supervisor(s)

Mr. Faheem Ahmed

Fall 2024

Document Sign off Sheet

Document Information

Project Title	Farmware App
Project Code	FYP-013/FL24
Document Name	Software Requirements Specifications
Document Version	1.0
Document Identifier	FYP-013/FL24-SRS
Document Status	Initial Version
Author(s)	Maryam Nadeem
Approver(s)	Mr. Faheem Ahmed
Issue Date	<Date of issuance of this document>

Name	Role	Signature	Date
Maryam Nadeem	Team Lead		31/07/2024
Muhammad Mujtaba	Team Member		31/07/2024
Hamza Bin Asif	Team Member		31/07/2024
Mr. Faheem Ahmed	Supervisor		31/07/2024
	Co-Supervisor		
	Project Coordinator		

Revision History

Date	Version	Description	Author
28/09/2024	1.0	Initial Version	Maryam Nadeem
02/07/2025	2.0	Feature Enhancement	Maryam Nadeem

Definition of Terms, Acronyms, and Abbreviations

Term	Description
AI	Artificial Intelligence
ML	Machine Learning
NLP	Natural Language Processing
DBMS	Database Management System

Table of Contents

Section Page Number

Introduction

Farmware is an innovative application aimed at improving cattle health management for farmers in Pakistan. By utilizing advanced data collection and machine learning algorithms, it enables farmers to monitor daily health data, detect diseases early, and receive tailored recommendations for recovery.

Purpose of Document

This document provides a detailed description of the functional and non-functional requirements for the Farmware App. It serves as a guide for the development and testing of the system.

Intended Audience

Development Team

Project Supervisors

Testers

Farmers (End-Users)

Overall System Requirements

Project Background

Farmware is designed to assist farmers in managing cow health through daily monitoring and disease prediction. It leverages advanced algorithms to provide early disease detection, tailored recommendations, and recovery tracking.

Problem Statement

Farmers lack efficient tools to monitor cow health, leading to delays in disease detection and increased mortality rates. Farmware automates health monitoring, providing farmers with actionable insights to improve cattle care.

Project Scope

Farmware will:

- Track cow health daily.
- Predict diseases using ML algorithms.
- Offer tailored advice based on symptoms.
- Provide recovery monitoring.

Not in Scope

- Integration with external farm management systems.
- Real-time syncing with external medical databases.
- Replacing professional veterinary diagnosis.

Project Objectives

- Monitor cow health through daily symptom toggles.

- Predict diseases based on collected data.
- Provide actionable advice.

Stakeholders & Affected Groups

- Farmers
- Agricultural organizations
- Veterinary professionals

Operating Environment

Farmware will operate on Android devices with internet access.

System Constraints

Internet connectivity is required for accessing the database.

The system relies on periodic updates to maintain accuracy in disease prediction.

Assumptions & Dependencies

- Farmers will have access to basic digital devices.
- Basic training will be provided to ensure proper usage
- Android mobile devices for hardware interfaces.
- Database: MySQL for managing health records.
- Backend: Node.js, Python.
- Frontend: Ionic Angular Framework
- Internet communication for data synchronization with the cloud.

External Interface Requirements

Hardware Interfaces

- Android Devices:
  - Logical Structure: The application will be installed on smartphones and tablets running Android OS. ○ Physical Addresses: Compatible with devices that have a minimum of 2 GB RAM and Android version 6.0 (Marshmallow) or higher.
  - Expected Behavior: The application should utilize device sensors (e.g., GPS for location tracking) and display health data in a user-friendly manner.

Software Interfaces

- Database Management System:
  - Name of Application: MySQL
  - Interface Details: The application will connect to MySQL for managing health records using SQL queries for data retrieval and storage.
- Backend Framework:
  - Name of Application: Node.js
  - Interface Details: The application will use RESTful APIs to communicate between the frontend and the backend services.

- Machine Learning Libraries:
    - Name of Application: Scikit-learn or TensorFlow
    - Interface Details: These libraries will be used for disease prediction and analysis, interacting through Python scripts.

Communication Interfaces

- Internet Connectivity:
    - The application will require a stable internet connection for real-time data exchange between the mobile app and the cloud servers. This will ensure timely access to health records and disease predictions.
- APIs:
    - The application will use RESTful APIs for communication between the frontend and backend services to ensure smooth data flow and user experience.

System Functions / Functional Requirements

Ref #	Functions	Category	Attribute	Details & Boundary Constraints
R1.1	Record daily cow health data input by farmers	Evident	System Response Time	Data entry and saving must occur within 5 seconds of submission.
R1.2	Predict diseases based on daily health data	Hidden	Prediction Accuracy	Predictions must be at least 85% accurate based on symptom analysis.
R1.3	Provide recommendations	Evident	Usability	Recommendations should be clearly presented and easy to understand.
R1.4	Secure login for farmers	Evident	Security	User authentication must utilize secure passwords and data encryption.

Details & Boundary Constraints

Attribute	Details and Boundary Constraints	Category
Response Time	When recording health data, the entry should be saved and visible within 5 seconds.	Optional

Concurrent User Load	The application must support a minimum of 10 users connected simultaneously.	Mandatory
User Interface	The interface should be intuitive and user-friendly to accommodate farmers with limited technical skills.	Mandatory
Language Support	The application must support both English and Urdu, with the possibility to add more regional languages if needed.	Optional
Security Measures	The application must implement TLS encryption for data transmission and secure user authentication protocols.	Optional
Database Scope	The system will use a static database of diseases and symptoms relevant to local conditions, with periodic updates. Real-time integration with global health databases is not in scope.	Mandatory
User Training	The system will provide basic training through documentation and support resources. Comprehensive in-person training sessions are not in scope but could be added in future phases.	Mandatory
Expansion Flexibility	The application should be designed to easily incorporate new features, such as additional livestock, crop management, or other animal breeds.	Optional

Use Cases

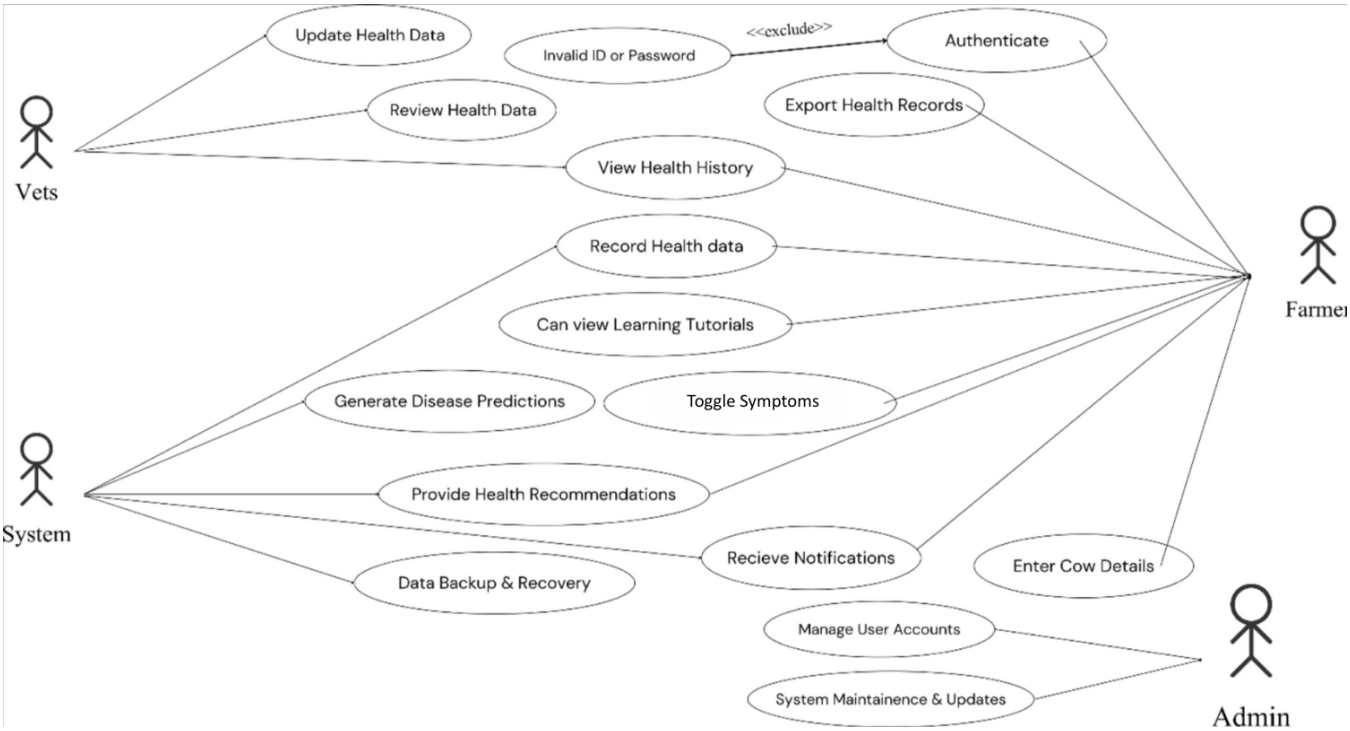
Actors

1. Farmer: The primary user of the application who inputs health data for cows, views health reports, and receives disease predictions and recommendations.
2. Veterinary Expert: Provides expert reviews and insights into the cow health data, assisting in validating disease predictions and updating health recommendations.
3. System (Farmware Application): Responsible for processing input data, analyzing health trends, generating predictions, and storing data.
4. Admin: Manages user accounts, application settings, and oversees data accuracy in the system.

Use Cases Table

Use Case ID	Use Case	Description	Actors Involved
UC1	Record Health Data	Farmer inputs daily health data for each cow, including symptoms and any observed behavior changes.	Farmer, System
UC2	Generate Disease Predictions	Based on the input data, the system uses ML algorithms to predict possible diseases.	System
UC3	Provide Health Recommendations	The system provides actionable health recommendations and preventive measures based on disease predictions.	System, Farmer
UC4	Display Health History	The farmer can view historical health data for each cow, allowing for trend analysis and monitoring over time.	Farmer

UC5	Review and Update Health Data	Veterinary expert reviews data and provides insights, updating health recommendations as needed.	Veterinary Expert
UC6	Manage User Accounts	Admin can create, edit, or delete farmer accounts and manage permissions.	Admin
UC7	Data Backup and Recovery	The system automatically backs up data to ensure no data loss, allowing for recovery in case of a failure.	System
UC9	Receive Notifications and Alerts	The system sends notifications to the farmer about critical health alerts, predictions, and reminders for data entry.	System, Farmer
UC10	System Maintenance and Updates	Admin performs regular system maintenance and updates, ensuring data accuracy and system performance.	Admin
UC11	Access Health Insights	Farmers and veterinary experts can access insights generated by the system, showing disease trends and risk levels across different cow groups.	Farmer, Veterinary Expert



System Boundary

The Farmware system is designed to monitor and manage cow health, providing disease predictions and health recommendations based on daily health data input by farmers. The system boundary includes functionalities like data entry, disease prediction, recommendations, user management, and notifications.

Actors and Their Roles

1. Farmer
    a. Role: The primary user who interacts with the system to manage cow health.
    b. Responsibilities: Records daily health data for cows, views health history, receives health alerts and recommendations, and exports health reports.
    c. Associated Use Cases: Record Health Data, View Health History, Receive Notifications and Alerts, Export Health Reports.

2. Veterinary Expert
    a. Role: A professional who provides additional insights and reviews cow health data for accuracy.
    b. Responsibilities: Reviews recorded health data, updates health recommendations, and accesses health insights for disease trends.
    c. Associated Use Cases: Review Health Data, Access Health Insights.
3. System (Farmware Application)
    a. Role: The automated system that processes input data, generates disease predictions, and provides recommendations.
    b. Responsibilities: Generates disease predictions, provides health recommendations, and backs up data.
    c. Associated Use Cases: Generate Disease Predictions, Provide Health Recommendations, and Data Backup.
4. Admin
    a. Role: The administrator responsible for managing the system, user accounts, and maintenance.
    b. Responsibilities: Creates and manages user accounts, ensures data integrity, and performs regular system updates.
    c. Associated Use Cases: Manage User Accounts, System Maintenance and Updates.

Use Cases Enlisted

Review Health Data

Veterinary experts review collected health data to analyze and provide insights or updated recommendations for better care.

Update Health Data

Vets can modify or refine recorded health data based on new findings, observations, or diagnostic results.

Invalid ID or Password (<<exclude>>) System restricts access to users with invalid credentials and prompts them to retry authentication or recover their password.

Enter Cow Details

Farmers add cow-specific details such as name, breed, age, and other identifiers into the system to maintain unique profiles.

Authenticate

Ensures secure access to the system by verifying users' credentials (username and password).

View Health History

Farmers can access detailed historical data for their cows to observe trends or patterns in health over time.

Record Health Data

Farmers input daily health observations, symptoms, or other relevant information into the system.

Attempt Symptom Toggling

Farmers update cow health records by toggling buttons for observed symptoms each day, ensuring accurate monitoring and timely care.

Can View Learning Tutorials

Farmers can access tutorials that guide them on using the application features or understanding animal health care better.

System Maintenance & Updates

Admin ensures the system is functioning optimally by applying updates, fixing bugs, and maintaining data integrity.

Manage User Accounts

Admin oversees account creation, updates, password resets, and user permissions for both farmers and veterinary experts.

Generate Disease Predictions

The system analyzes collected health data to predict potential illnesses or risks using advanced algorithms.

Provide Health Recommendations

Based on predictions and input data, the system generates customized health advice and preventive care measures.

Description Of Usecases

1. Review Health Data

Name	Review Health Data
Actors	Veterinary Expert
Purpose	To analyze health data and provide insights.
Description	The vet reviews the cow's recorded health data to identify issues and update recommendations.

Name	Review Health Data
Cross References	Functions: Analyze Data, View History
Pre-Conditions	Farmer has submitted health data.
Successful Post-Conditions	Vet provides actionable insights.
Failure Post-Conditions	No insights provided; review fails due to incomplete data.

2. Update Health Data

Name	Update Health Data
Actors	Veterinary Expert, Farmer
Purpose	To modify or refine existing health data.
Description	The actor updates health data entries for accuracy or based on new observations.
Cross References	Functions: Modify Data
Pre-Conditions	Health data already exists in the system.
Successful Post-Conditions	Data is updated successfully in the system.
Failure Post-Conditions	Update fails due to system or input errors.

3. Invalid ID or Password (<<exclude>>)

Name	Invalid ID or Password
Actors	Farmer, Vet, Admin
Purpose	To prevent unauthorized access.
Description	System denies access and prompts users to reattempt or recover passwords if credentials are invalid.
Cross References	Functions: Authentication, Error Handling
Pre-Conditions	User enters invalid credentials.
Successful Post-Conditions	User is prompted to retry or recover credentials.

Name	Invalid ID or Password
Failure Post-Conditions	Access is denied, but the system remains secure.

4. Enter Cow Details

Name	Enter Cow Details
Actors	Farmer
Purpose	To create unique profiles for cows.
Description	Farmer enters details like name, age, breed, and other specifics.
Cross References	Functions: Add Cow Details
Pre-Conditions	Farmer has logged in successfully.
Successful Post-Conditions	Cow details are saved into the system.
Failure Post-Conditions	Data entry fails, and no cow profile is created.

5. Authenticate

Name	Authenticate
Actors	Farmer, Veterinary Expert, Admin
Purpose	To allow secure access to the system.

Name	Authenticate
Description	User logs in using valid credentials.
Cross References	Functions: Authentication
Pre-Conditions	Valid credentials are stored in the system.
Successful Post-Conditions	User gains access to their dashboard.
Failure Post-Conditions	Access is denied.

6. View Health History

Name	View Health History
Actors	Farmer
Purpose	To track past health trends and data.
Description	The farmer views historical data for a specific cow.
Cross References	Functions: Retrieve Data
Pre-Conditions	Historical data is available.
Successful Post-Conditions	Data is displayed on the farmer's screen.
Failure Post-Conditions	Data retrieval fails.

7. Record Health Data

Name	Record Health Data
Actors	Farmer
Purpose	To log daily health details for each cow.
Description	The farmer records symptoms, diet, or observations about the cow's health.
Cross References	Functions: Input Data
Pre-Conditions	Farmer is authenticated and cow details exist.
Successful Post-Conditions	Data is successfully recorded.
Failure Post-Conditions	Data is not recorded; an error is displayed.

8. Symptom Toggle

Name	Attempt Symptom Toggling
Actors	Farmer
Purpose	To update the cow's health status by toggling visible symptoms based on observation.
Description	The farmer selects observed symptoms by toggling buttons, allowing the system to record and analyze the health status of the cow.
Cross References	Functions: Symptom Toggling, Health Monitoring
Pre-Conditions	The farmer is logged in and has access to the cow profile and daily symptom toggling interface.
Successful Post-Conditions	Selected symptoms are saved and used to update health records and generate possible recommendations.
Failure Post-Conditions	No symptoms are toggled; health update is not recorded.

9. Can View Learning Tutorials

Name	Can View Learning Tutorials
Actors	Farmer
Purpose	To provide educational material for better care of livestock.
Description	Farmers access tutorials or guides available in the system.
Cross References	Functions: View Tutorials
Pre-Conditions	Tutorials are uploaded in the system.
Successful Post-Conditions	Tutorials are successfully displayed.
Failure Post-Conditions	Tutorials fail to load; user is notified.

10. System Maintenance & Updates

Name	System Maintenance & Updates
Actors	Admin
Purpose	To ensure system reliability and introduce improvements.
Description	Admin performs regular checks, applies updates, and resolves issues.

Name	System Maintenance & Updates
Cross References	Functions: System Maintenance
Pre-Conditions	System requires maintenance or updates.
Successful Post-Conditions	System operates optimally after updates.
Failure Post-Conditions	Issues remain unresolved; system notifies admin of errors.

11. Manage User Accounts

Name	Manage User Accounts
Actors	Admin
Purpose	To manage profiles of farmers and veterinary experts.
Description	Admin creates, edits, or deactivates user accounts.
Cross References	Functions: User Management
Pre-Conditions	User accounts exist.
Successful Post-Conditions	Accounts are created, updated, or deactivated as intended.
Failure Post-Conditions	Account actions fail, and admin is notified of an issue.

12. Generate Disease Predictions

Name	Generate Disease Predictions
Actors	System
Purpose	To predict potential diseases based on health data.
Description	The system analyzes health data to identify patterns and risks.
Cross References	Functions: Analyze Data, Predictive Analytics
Pre-Conditions	Adequate health data has been recorded.
Successful Post-Conditions	Predictions are displayed to the farmer or vet.
Failure Post-Conditions	Predictions are not generated; user is notified of the error.

13. Provide Health Recommendations

Name	Provide Health Recommendations
Actors	System
Purpose	To suggest preventive measures or treatments.
Description	System provides tailored recommendations based on analysis.
Cross References	Functions: Health Advisory
Pre-Conditions	Predictions have been generated.
Successful Post-Conditions	Recommendations are delivered to the user.
Failure Post-Conditions	No recommendations are provided; an error message is displayed.

14. Data Backup & Recovery

Name	Data Backup & Recovery
Actors	System, Admin
Purpose	To ensure data safety and allow recovery during failures.
Description	System performs regular backups and allows admins to restore data when needed.
Cross References	Functions: Data Management
Pre-Conditions	Backup functionality is configured.
Successful Post-Conditions	Data is backed up or recovered successfully.
Failure Post-Conditions	Data loss or recovery failure occurs.

15. Receive Notifications

Name	Receive Notifications
Actors	Farmer, Veterinary Expert
Purpose	To provide updates, reminders, or alerts.
Description	System sends notifications for updates on health status, predictions, or tasks.
Cross References	Functions: Notification System
Pre-Conditions	Notifications are enabled in the system.
Successful Post-Conditions	Notifications are delivered successfully.
Failure Post-Conditions	Notifications are not sent; user is alerted of the failure.

Typical Course of Events

Actor Action	System Response
1. This use case begins when a user logs into the system using valid credentials.	System authenticates the user and grants access to the dashboard.
2. Farmer enters cow details like name, age, and breed.	System saves the cow profile into the database.
3. Farmer records health data, including symptoms, diet, or observations.	System validates and stores the health data.

Actor Action	System Response
4. Farmer views historical health data for a specific cow.	System retrieves and displays the cow's health history.
5. Farmer attempts symptom toggling.	System records the responses and analyzes them.
6. Veterinary Expert reviews the recorded health data to analyze trends and identify issues.	System organizes the data and displays it for analysis.
7. Veterinary Expert updates health data entries based on new findings.	System updates the database with the new information.
8. System generates disease predictions based on the recorded health data.	Predictions are displayed to the farmer or Veterinary Expert.
9. System provides tailored health recommendations to the user.	Recommendations are displayed based on the analyzed data.
10. Farmer views educational tutorials on livestock care.	Tutorials are displayed on the farmer's screen.
12. Admin manages user accounts, including creating, editing, or deactivating accounts.	System updates the user account database accordingly.
13. Admin performs system maintenance to resolve issues or apply updates.	System executes maintenance tasks and ensures reliability.
14. System performs data backups or allows admins to recover data during failures.	Data is backed up or restored successfully.
15. System sends notifications to users for updates, alerts, or reminders.	Notifications are delivered to the relevant users.
16. Farmer logs out of the system after completing tasks.	System ends the session and logs out the user securely.

Alternative Course Of Events

Step 1: User enters invalid credentials.

System prompts for retry or password recovery.

Step 2: Farmer enters incomplete health data.

System prompts the farmer to fill in the missing fields.

Step 3: No historical data exists for the cow.

System provides an option to record new data.

Step 4: Farmer abandons the symptom toggle.

System doesn't proceed forward

Non - Functional Requirements

1. Performance Requirements
    - The system should handle the daily input of health data for multiple cows without performance degradation.
    - Disease predictions should be generated within 5 seconds of submitting the health data.
    - The system must support simultaneous usage by up to 1000 farmers without noticeable latency.
2. Safety Requirements
    - The system must ensure that user data, including health records and cow information, is stored securely with regular backups.
    - The system must comply with relevant data protection laws to ensure user privacy and security.
    - The system should have features to handle errors without loss of data (e.g., automated data recovery in case of system failure).
3. Security Requirements
    - All sensitive data must be encrypted both at rest and in transit using industry-standard encryption protocols (e.g., AES- 256).
4. Reliability Requirements
    - It must be capable of automatically recovering from failures, with no data loss during downtime.
    - The system must be scalable to handle growth in the number of users or cows monitored without degrading performance.
5. Usability Requirements
    - The user interface should be simple and intuitive for farmers with limited technical expertise.

- The system should be accessible on both mobile and desktop devices.
- The system should provide real-time feedback on health data entries, with easy-to-understand recommendations.

6. User Documentation
    - The system must include comprehensive user documentation, including a user manual detailing how to enter health data, interpret predictions, and follow health recommendations.
    - Documentation should also cover troubleshooting common issues, FAQs, and step-by-step guides for using the system effectively.
    - Tutorials and educational material about cow health care and disease prevention should be available to farmers within the system.

References

- SRS for Livestock Managment System 1.0
  https://www.studocu.com/row/document/riphah-international-university/computer-sciences/srs-first-final-draft-srs-for-livestock-managment-system-10/42526023