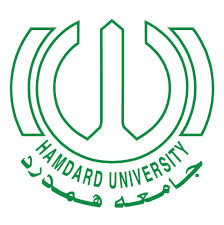
Hamdard University

Department of Computing

Final Year Project



**Smart Cattle Monitoring System with IoT and AI/ML**

**(FYP-033/FL24)**

**Software Requirements Specifications**

Submitted by

Zunair Shahid (2422-2021)

Hina Roshan (1389-2019)

Supervisor(s)

Sir Iqbal Uddin Khan

**Fall 2024**

**Document Sign off Sheet**

### **Document Information**

|  |  |
| --- | --- |
| **Project Title** | Smart Cattle health monitoring with IoT and AI/ML |
| **Project Code** | FYP-033/FL24 |
| **Document Name** | Software Requirements Specifications |
| **Document Version** | 2.0 |
| **Document Identifier** | FYP-033/FL24-SRS |
| **Document Status** | Final |
| **Author(s)** | Zunair Shahid, Hina Roshan |
| **Approver(s)** | Sir Iqbal Uddin Khan |
| **Issue Date** | 5-Jan-2025 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Signature** | **Date** |
| Zunair Shahid | Team Lead |  | 16-Jan-2025 |
| Hina Roshan | Team Member 2 |  | 16-Jan-2025 |
| Sir Iqbal Uddin Khan | Supervisor |  | 16-Jan-2025 |
|  | Co-Supervisor |  |  |
| Sir Sulaman Ahmed Naz | Project Coordinator |  |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 5-Jan-2025 | 1.0 |  | Hina, Zunair |
| 3-July-2025 | 2.0 |  | Hina, Zunair |
|  |  |  |  |
|  |  |  |  |

**Definition of Terms, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Description** |
| IoT | Internet of Things |
| AI | Artificial Intelligence |
| ML | Machine Learning |
| HTTP | HyperText Transfer Protocol |
| HTTPS | HyperText Transfer Protocol Secure |
| MQTT | Message Queuing Telemetry Transport |
| TLS | Transport Layer Security |
| GPS | Global Positioning System |

# **Table of Contents**

[**1.** Introduction 8](#_Toc4677804)

[1.1 Purpose of Document 8](#_Toc4677805)

[1.2 Intended Audience 8](#_Toc4677806)

[1.3 Abbreviations 8](#_Toc4677807)

[**2.** Overall System Description 9](#_Toc4677808)

[2.1 Project Background 9](#_Toc4677809)

[2.2 Problem Statement 9](#_Toc4677810)

[2.3 Project Scope 9](#_Toc4677811)

[2.4 Not In Scope 9](#_Toc4677812)

[2.5 Project Objectives 9](#_Toc4677813)

[2.6 Stakeholders & Affected Groups 9](#_Toc4677814)

[2.7 Operating Environment 9](#_Toc4677815)

[2.8 System Constraints 9](#_Toc4677816)

[2.9 Assumptions & Dependencies 9](#_Toc4677817)

[**3.** External Interface Requirements 10](#_Toc4677818)

[3.1 Hardware Interfaces 10](#_Toc4677819)

[3.2 Software Interfaces 10](#_Toc4677820)

[3.3 Communications Interfaces 10](#_Toc4677821)

[**4.** System Functions / Functional Requirements 11](#_Toc4677822)

[4.1 System Functions 11](#_Toc4677823)

[4.2 Use Cases 12](#_Toc4677824)

[4.2.1 List of Actors 12](#_Toc4677825)

[4.2.2 List of Use Cases 12](#_Toc4677826)

[4.2.3 Use Case Diagram 13](#_Toc4677827)

[4.2.4 Description of Use Cases 13](#_Toc4677828)

[**5.** Non - Functional Requirements 16](#_Toc4677829)

[5.1 Performance Requirements 16](#_Toc4677830)

[5.2 Safety Requirements 16](#_Toc4677831)

[5.3 Security Requirements 16](#_Toc4677832)

[5.4 Reliability Requirements 16](#_Toc4677833)

[5.5 Usability Requirements 16](#_Toc4677834)

[5.6 Supportability Requirements 16](#_Toc4677835)

[5.7 User Documentation 16](#_Toc4677836)

[**6.** References 17](#_Toc4677837)

# **Introduction**

## **Purpose of Document**

## The objective of this Software Requirement Specifications (SRS) document is to provide a comprehensive overview of the design principles, system architecture, and functionality of the Smart Cattle Monitoring System, developed using the Waterfall methodology. This document acts as a guide for the development team, detailing the technical requirements and specifications essential for the successful deployment of the system. It emphasizes the integration of IoT sensors and AI/ML algorithms to facilitate effective cattle health monitoring, thereby addressing the challenges commonly faced by livestock farmers.

## **Intended Audience**

## The primary audience for the Smart Cattle Monitoring System includes:

1. **Livestock Farmers**:

* Farmers seeking efficient, real-time monitoring of their cattle’s health and behavior.
* Those aiming to reduce operational costs and improve farm productivity through data-driven decisions.

1. **Dairy Farm Owners**:

* Owners focused on maximizing milk production by ensuring the health and fertility of their cattle.

1. **Agricultural and Livestock Industry Stakeholders**:

* Industry professionals interested in integrating technology to improve livestock management.

1. **Veterinarians**:

* Professionals requiring accurate, real-time data for diagnosis and treatment of livestock health issues.

1. **Government and Non-Governmental Organizations (NGOs)**:

* Institutions promoting technological advancements in agriculture for rural development and economic uplift.

1. **Tech Enthusiasts and Researchers**:

* Individuals or organizations interested in exploring the use of IoT and AI in the agricultural sector.

The system is particularly designed to benefit farmers and livestock owners in Pakistan, considering the challenges faced by this demographic in traditional cattle monitoring practices.

# Overall System Description

## **Project Background**

## Livestock is a crucial sector in driving the rural economy, contributing significantly to the national GDP with a 14.36% share and making up 62.68% of the agricultural GDP [1]​. Despite its importance, traditional methods of livestock monitoring rely heavily on manual observations and record-keeping. These outdated practices often lead to inaccuracies in health tracking.

## The absence of real-time monitoring and data analysis can delay the detection of diseases or injuries, which may result in cattle fatalities and reduced productivity.

## **Problem Statement**

## Traditional livestock monitoring methods relying on manual observation are inefficient and error-prone, delaying the detection of diseases. This results in reduced productivity, increased mortality, and higher costs. A smart, real-time system integrating IoT and AI is essential to enhance cattle health, and farm management efficiency.

## **Project Scope**

**Design and Development of Hardware Components**:  
Selection and integration of sensors (e.g., temperature and heart rate) and GPS such modules with microcontrollers.

**Software Development**:  
Development of firmware for microcontrollers, a cloud-based data storage system, and mobile applications for real-time monitoring.

**AI and ML Algorithm Development**:  
Design and implementation of machine learning algorithms for health trend analysis.

**System Integration**:

Ensuring smooth integration between hardware components, software, and cloud services for effective data transmission and storage.

**User Interface Design**:

Creation of user-friendly interfaces enabling farmers to access real-time data and receive alerts, with support for multiple languages for better usability.

**Initial Testing and Validation**:

Conducting tests on a small-scale farm or a limited number of cattle to validate the system's performance and functionality.

## **Not in Scope**

**Extensive Field Testing:** The project will not encompass large-scale field trials or extended performance assessments beyond the initial proof-of-concept validation within the current project timeframe.

**Custom Hardware Development:** The development of specialized hardware components beyond commercially available sensors and microcontrollers is not part of the project scope.

**Post-Deployment Support and Maintenance:** Ongoing support, maintenance, and updates after the initial deployment phase will not be included.

**Scalability and Redundancy Solutions:** The design and implementation of solutions for scalability and redundant systems for large-scale deployments fall outside the current scope of the project.

## **Project Objectives**

The Objective of the Smart cattle monitoring system using IoT sensors to collect and AI is to analyze, and provide understanding into cattle health, activities, ultimately improving farm management decisions and farm productivity.

**“Empowering Farmers with Intelligent Cattle Care”**

## **Stakeholders & Affected Groups**

## The primary users of the system are farmers, who will benefit from improved cattle health and productivity through real-time monitoring, though they may face a learning curve. Veterinary experts will use the data for faster and more accurate diagnoses, while adjusting to AI-driven recommendations. Tech developers and engineers will design and integrate the system but may encounter integration challenges. IoT hardware suppliers will benefit from increased demand, though supply fluctuations may pose challenges. Government bodies will support better cattle management but may face compliance issues. Local agricultural communities, especially farmers, will experience improved economic stability, but resistance from older generations may occur. Investors and funding bodies will provide financial backing but face risks if adoption is slow. App developers will create the mobile app, facing challenges in ensuring user-friendliness. Cattle will benefit from better health and management with minimal risk due to continuous monitoring.

## **Operating Environment**

## The smart cattle monitoring system will be deployed in rural farming areas with varying infrastructure. It will rely on cloud services for data processing, requiring stable internet connectivity, but will store data locally in areas with limited network access. Hardware components like sensors and GPS tracker must be durable enough to withstand weather conditions. The app will be designed in both English and Urdu to accommodate local farmers, offering a user-friendly interface. Data security will be prioritized to protect sensitive cattle health information.

## **System Constraints**

## The smart cattle monitoring system may face several constraints, including limited internet connectivity in rural areas, which could affect real-time data transfer. The accuracy and durability of sensors may be impacted by harsh environmental conditions. Power supply reliability is another challenge, especially in remote locations where grid access may be unstable. Additionally, the system’s reliance on GPS could be affected by poor satellite coverage in certain areas. Cost constraints may limit the scale of the system’s deployment initially, and ensuring data security while making the app accessible in both English and Urdu may require additional resources.

## **Assumptions & Dependencies**

## The system assumes stable internet connectivity on farms for effective data transmission. It is designed based on standard cattle farming practices, with no consideration for extreme operational variations. The project will operate within a strict budget, utilizing low-cost hardware and open-source software solutions. Additionally, it is assumed that farmers will possess basic technical literacy to interact with the mobile app for monitoring and receiving alerts.

# External Interface Requirements

## **Hardware Interfaces**

**Sensors**:

* Includes health monitoring sensors such as temperature, heart rate and movement sensors to collect real-time physiological data from cattle.

**Microcontrollers**:

* Utilizes Arduino for processing sensor data and managing communication between hardware components.
* Utilizing the Esp32 module for collecting the data from the Arduino wirelessly and send it to cloud storage.

**GPS Modules**:

* Enables real-time location tracking of cattle to monitor movements and location.

**Communication Modules**:

* Wi-Fi modules facilitate wireless data transmission to the cloud, ensuring real-time updates and accessibility.

**Power Supplies**:

* Employs batteries or other power sources to provide consistent power to all hardware components, ensuring uninterrupted operation.

These interfaces work together within a logical structure where sensors gather data, microcontrollers process it, and communication modules transmit it to a cloud-based system. The physical addresses and connections are designed to align with industry-standard protocols, ensuring reliability and ease of maintenance. The expected behavior includes real-time data collection, processing, and seamless transmission to support effective cattle monitoring.

## **Software Interfaces**

**1. Microcontroller Programming Tools**

**Name:** Arduino

**Owner:** Arduino AG (founded by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, and David Mellis)

**Details:** Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is widely used for creating interactive projects, such as robotics, IoT devices, and automation systems. Arduino boards are programmable microcontrollers that enable users to write and upload code using the Arduino IDE (Integrated Development Environment).

**2. AI/ML Frameworks**

* **Name: Scikit-learn**
* **Owner: Open Source (Community-driven; initially developed by David Cournapeau)**
* **Details: Used in the project to build and deploy machine learning models (e.g., Logistic Regression) that analyze cattle health data like temperature and heart rate to predict whether the animal is healthy or unhealthy.**

**3. Mobile App Development**

* **Name**: Flutter (Dart).
* **Owner**: Google.
* **Details**: Builds cross-platform apps for real-time monitoring and alerts.

**4. Cloud Services**

* **Name: Firebase**
* **Owner: Google**
* **Details: Used for real-time database storage, user authentication, and backend hosting of the mobile application. It stores cattle sensor data and receives health predictions from the ML model.**

**Communications Interfaces**

**1. Wireless Communication**

* **Wi-Fi Modules**: Transmit data from Esp32 to the cloud via standard Wi-Fi protocols.

**2. Cloud Connectivity**

* **Protocols**: HTTP/HTTPS ensure secure and efficient data transfer between the system and cloud servers.

**3. Mobile App Integration**

* **Type**: Data is sent from the cloud to mobile apps using RESTful APIs or WebSocket protocols for real-time updates.

**4. GPS Modules**

* Provide real-time location data via serial or UART protocols.

**5. Sensor-Microcontroller Communication**

* Use I2C, or UART protocols to connect sensors with the microcontroller.

**6. Bluetooth Communication:**

* Wireless communication between Arduino with a help of a Bluetooth module and the Esp32 for sharing the data.

# System Functions / Functional Requirements

## **System Functions**

1. **Real-Time Health Monitoring**
   * Sensors (temperature, heart rate, activity) continuously collect data from cattle.
   * The system analyzes the data for anomalies (e.g., signs of illness) using AI/ML models.
2. **Location Tracking**
   * GPS modules provide real-time tracking of cattle movements and locations.
3. **Alerts and Notifications**
   * Immediate notifications for critical conditions (e.g., abnormal health metrics) via the mobile app.
4. **Historical Data Analysis**
   * Display historical trends and patterns for cattle health, enabling predictive analytics.
5. **User Management:**
   * Secure login for users (e.g., farmers, veterinarians), with role-based access to system features.
6. **Data Storage and Backup:**
   * Cloud-based storage ensures redundancy and secure backup of all collected data.
7. **Multilingual Support**
   * Mobile application available in English and Urdu for user convenience.
8. **Integration with IoT Hardware**
   * Reliable communication between sensors, microcontrollers (e.g., Arduino, Esp32), and cloud services.
   1. **Use Cases**
9. **List of Actors:**

**Farmer:** The primary user managing cattle and viewing health data.

**System Admin:** Manages user accounts, sensors, and AI model updates.

**Cloud Server:** Stores, processes, and displays sensor data.

**Sensor Hardware:** Collects cattle health data and sends it to the cloud.

**AI Model:** Processes data and predicts cattle health status.

1. **List of Use Cases:**

#### **User Management (Farmer):**

* **Sign Up**
* **Login**
* **Password Reset**

#### **Cattle Management (Farmer):**

* **Add Cattle**
* **Edit/Delete Cattle**

#### **Health Monitoring (Farmer & Cloud Server):**

* **Show Health Status on Dashboard**
* **Display Sensor Data in Charts**

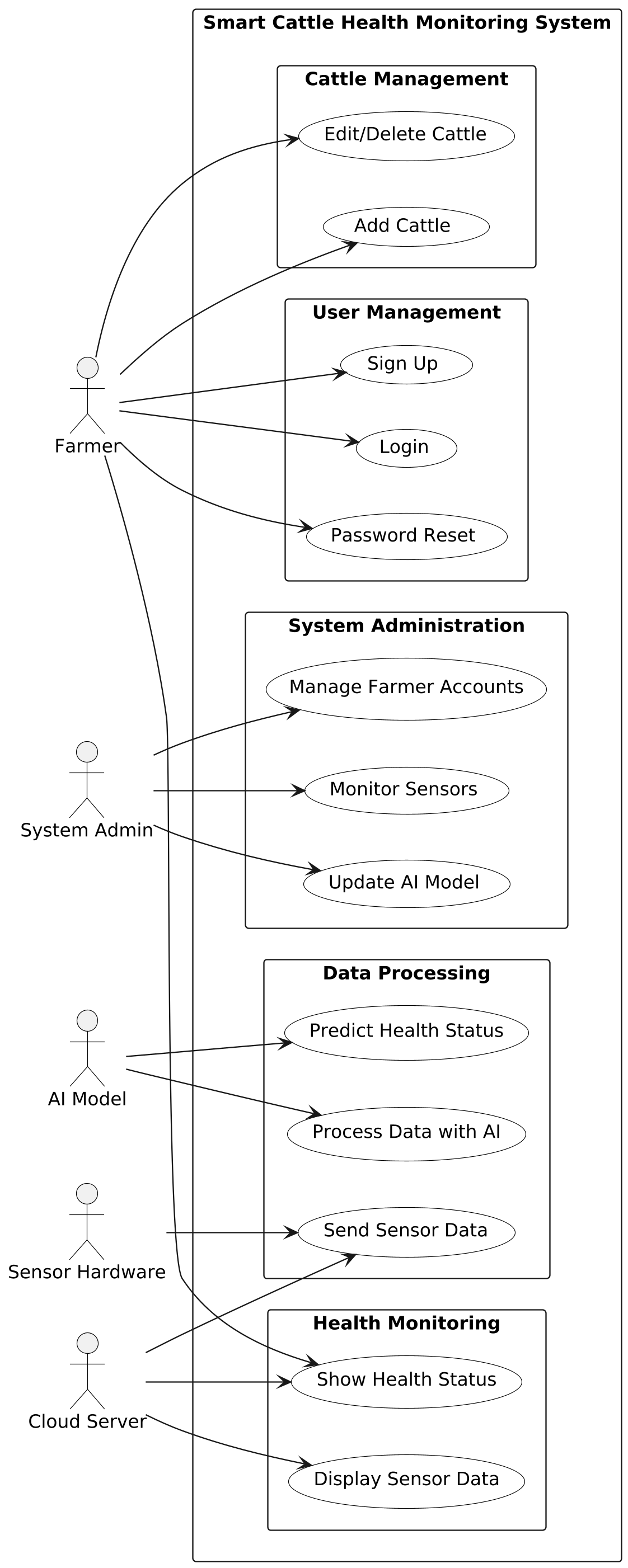
#### **Data Processing (Sensor Hardware & AI Model):**

* **Send Sensor Data to Cloud**
* **Process Data Using AI Model**
* **Predict Cattle Health Status**

#### **System Administration (System Admin):**

* **Update AI Model**
* **Manage Farmer Accounts**
* **Monitor Sensor Connectivity**

1. **Use Case Diagram:**



1. **Description of Use Cases:**

**User Management:**

* **Sign Up:** Allows farmers to create an account by providing credentials.
* **Login:** Authenticates farmers to access their dashboard.
* **Password Reset:** Enables farmers to reset forgotten passwords.

**Cattle Management:**

* **Add Cattle:** Farmers can register a new cattle profile.
* **Edit/Delete Cattle:** Farmers can update or remove cattle details.

**Health Monitoring:**

* **Show Health Status on Dashboard:** Displays AI-predicted cattle health conditions.
* **Display Sensor Data in Charts:** Shows real-time sensor readings visually.

**Data Processing:**

* **Send Sensor Data to Cloud:** Sensors upload cattle health data to the cloud.
* **Process Data Using AI Model:** The AI model analyzes sensor readings.
* **Predict Cattle Health Status:** AI determines whether cattle is healthy or sick.

**System Administration:**

* **Update AI Model:** Admins improve AI models for better predictions.
* **Manage Farmer Accounts:** Admins oversee farmer registrations and permissions.
* **Monitor Sensor Connectivity:** Admins ensure sensors are functioning properly.

# Non - Functional Requirements

## **Performance Requirements**

**Real-Time Monitoring:**

The system must process and analyze sensor data (temperature, heart rate, activity, GPS) in real-time to detect anomalies and provide alerts without noticeable delays.

Data transmission from hardware to the cloud should occur within 5-8 seconds.

**Accuracy:**

IoT sensors should provide data with at least 90% accuracy, and AI/ML models should achieve a prediction accuracy of 90% for health or anomaly detection.

**Low Latency:**

The mobile application must retrieve and display updated cattle status within 2-3 second after a request.

## **Safety Requirements**

**Animal Safety:**

* Sensors must be non-invasive, lightweight, and securely fastened to avoid causing discomfort or harm to the cattle.
* Enclosures for hardware must be weatherproof and designed to withstand harsh farm conditions.

**Electrical Safety:**

* All electronic components should operate at low voltages to prevent hazards in case of malfunctions.

**Data Integrity:**

* Prevent loss of critical health and location data by ensuring redundant storage mechanisms (local and cloud).

## **Security Requirements**

**Data Encryption:**

* All data transmitted between devices, the cloud, and the mobile application should be encrypted using secure protocols (e.g., HTTPS, TLS).

**Authentication:**

* Implement secure user authentication for the app with password-protected accounts and optional two-factor authentication (2FA).

**Access Control:**

* Farmers should only access data related to their registered cattle.

**Data Backup:**

* Automated cloud backups should be performed daily to prevent data loss due to system failures or cyberattacks.

## **Reliability Requirements**

**System Uptime:**

The system should maintain a minimum uptime of 99%, ensuring continuous monitoring and functionality.

**Fault Tolerance:**

The system should handle single-component failures (e.g., one sensor or a temporary network issue) without affecting overall functionality.

**Durability:**

Hardware components must function reliably in farm environments, including temperature and varying humidity levels.

## **Usability Requirements**

**User-Friendly App Interface:**

The mobile app should feature a simple and intuitive design in both English and Urdu to accommodate diverse users.

**Clear Alerts and Notifications:**

Farmers should receive clear, concise notifications for critical events (e.g., health issues) through push notifications.

**Minimal Training Required:**

Farmers should require minimal training to use the system, supported by clear in-app instructions and tooltips.

## **Supportability Requirements**

**Regular Updates:**

* Provide software updates to improve functionality, fix bugs, and enhance security.

**Compatibility:**

* Ensure the system is compatible with commonly available Android and iOS devices.

**Hardware Maintenance:**

* Provide clear guidelines for maintaining and replacing hardware components (e.g., cleaning sensors, replacing batteries).

## **User Documentation**

**User Manual:**

* A comprehensive user manual should be provided, including:
  + System setup instructions.
  + Hardware placement guidelines.
  + App login and navigation details.

**Quick Start Guide:**

* A concise, illustrated guide to help farmers quickly install and start using the system.

**Troubleshooting Section:**

* Include common issues and their resolutions (e.g., sensor not transmitting data, app connection errors).

**Video Tutorials:**

* Offer online video tutorials in Urdu and English for better understanding of system usage and troubleshooting.

**FAQs:**

* Include a frequently asked questions section addressing common user concerns.

# References

# [1] Punjab Livestock & Dairy Development Department, "Livestock Contribution," [Online]. Available: https://livestock.punjab.gov.pk/livestock-contribution. [Accessed: 6-Jan-2025].s