IEEE SRS FORMAT

Software Requirements Specification (SRS) for AgroCare

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for AgroCare, an IoT-based smart farming solution. The system aims to optimize water usage, provide crop recommendations, and enhance sustainable agriculture practices through real-time soil monitoring and automated irrigation.

1.2 Scope

AgroCare is a smart farming system designed to address challenges in modern agriculture, such as resource efficiency and water scarcity. The system integrates sensors, IoT devices, and machine learning algorithms to analyze soil health, predict irrigation schedules, and recommend crops. It provides farmers with actionable insights via a user-friendly web interface, improving productivity and sustainability.

1.3 Definitions, Acronyms, and Abbreviations

- **IoT:** Internet of Things
- **ML:** Machine Learning
- **NPK:** Nitrogen, Phosphorus, and Potassium (soil nutrients)
- **ESP8266:** A Wi-Fi microcontroller module
- **AWS:** Amazon Web Services

1.4 References

- IEEE SRS Guidelines
- Documentation on ESP8266 and IoT protocols
- Research on ML algorithms for crop prediction
- UN Sustainable Development Goals (SDGs)

2. Overall Description

2.1 Product Perspective

AgroCare is a modular IoT-based system that integrates soil sensors, cloud computing, and machine learning models. The product operates as an end-to-end solution for farmers,

leveraging data to automate irrigation and recommend crops, thus improving resource utilization and productivity.

2.2 Product Features

1. **Real-Time Monitoring:** Tracks soil conditions (moisture, pH, NPK) and weather data.
2. **Automated Irrigation:** Controls water distribution based on real-time moisture levels.
3. **Crop Recommendations:** Suggests high-yield crops tailored to soil health and environmental conditions.
4. **Predictive Analytics:** Forecasts irrigation schedules and crop yields using historical and real-time data.
5. **Remote Access:** Provides farmers with web and SMS updates for easy system management.

2.3 User Classes and Characteristics

- **Farmers:** Primary users seeking to optimize farming operations.
- **Agricultural Experts:** Analysts using data insights for large-scale farming recommendations.
- **Developers:** Technical personnel maintaining and scaling the system.

2.4 Constraints

- System must adhere to data privacy regulations.
- Requires stable internet connectivity for cloud-based processing.
- Hardware must operate effectively under diverse environmental conditions.

3. Specific Requirements

3.1 Functional Requirements

1. **Soil Data Collection:**
    - Sensors measure moisture, pH, temperature, and NPK levels.
    - Data is transmitted to the cloud for processing.
2. **Crop Prediction:**
    - ML algorithms recommend crops based on soil and environmental data.
3. **Automated Irrigation:**

    - ESP8266 triggers irrigation mechanisms when moisture falls below thresholds.
4. **User Interface:**
    - Farmers access system insights via a web dashboard or SMS alerts.
5. **Pilot Testing:**
    - Initial deployment on small farms to validate functionality and gather feedback.

3.2 Non-Functional Requirements

1. **Performance:**
    - Real-time data processing with a latency of less than 2 seconds.
    - Scalable architecture supporting up to 1,000 sensors.
2. **Security:**
    - End-to-end encryption for data transmission.
    - Secure authentication for web access.
3. **Usability:**
    - Intuitive interface with multilingual support.
4. **Reliability:**
    - System uptime of 99.5% with failover mechanisms.

4. System Models

4.1 Flowchart

1. Sensor collects soil and environmental data.
2. Data is sent to the cloud for analysis.
3. ML model processes the data and provides insights.
4. Automated irrigation system regulates water flow.
5. Farmers receive real-time updates via web or SMS.

4.2 System Architecture

- **Frontend:**
    - Web-based interface developed with ReactJS for real-time monitoring.
- **Backend:**

    - Python-based server using Flask/Django.
    - Cloud hosting on AWS.
- **Hardware:**
    - Sensors (pH, NPK, moisture) and ESP8266 microcontroller.

5. Other Requirements

5.1 Legal and Ethical Concerns

- Compliance with agricultural and environmental regulations.
- Ethical AI practices ensuring unbiased crop recommendations.

5.2 Assumptions and Dependencies

- Farmers have basic internet-enabled devices.
- Hardware components (sensors, controllers) are readily available.
- Stable internet connectivity is maintained in deployment regions.