

Software Requirements Specification for SmartAgCloud

Prepared by:

Priya Yadav

Jessica Mathias

Ankita Chikodi

Priya Khadke

Professor:

Jerry Gao <jerry.gao@sjsu.edu>

San Jose State University

Spring 2019

Table of Contents

Table of Contents	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Intended Audience	1
1.3 Application Scope.....	1
1.4 Out of Scope	2
2. Overall Description	3
2.1 User Classes and Characteristics	3
2.2 Design and Implementation Constraint	4
2.3 Design and Implementation Constraint	4
2.4 Assumptions and Dependencies	4
3. Requirement Analysis.....	5
3.1 Functional Requirement Analysis.....	5
3.2 Nonfunctional Analysis	6
4. Use Cases.....	7
4.1 User Registration	7
4.2 User Login	7
4.3 Build Smart Farm.....	8
4.4 Sensor Data Management	9
4.5 View Ranch-based dashboard.....	9
4.6 View Cloud-based dashboard	10
4.7 Receive Alerts.....	11
5. System infrastructure and architectures	12
6. System design and Component interaction design	14
6.1 Database Design	15
6.2 Simulation Software Design	17
6.3 Deployment-Oriented System Infrastructure.....	18
Fig 6: Deployment-Oriented System Infrastructure.....	18
7. Plan and schedule.....	19
8. Technology Selection	20
8.1 Cloud Provider.....	20
8.2 MQTT	20
9. References:.....	21
Abbreviation.....	21

1. Introduction

1.1 Purpose

This document describes the system requirements specification for an IoT-based smart agriculture infrastructure service management system as a service on a cloud namely SmartAgCloud. The document describes in detail the requirements and high-level design of the application. The document includes the details on defining scope along with assumptions and dependencies. The IoT based application will be designed and implemented for ranch field farms.

The application supports a connected farm where farm conditions across the ranch can be monitored by the farmer as per his needs. This is achieved by implementing a large-scale on-demand IoT-based agriculture system infrastructure services for farmers. Each farmer could select choose one or more IoT agriculture networks for their ranch fields. The sensors used for monitoring the farm conditions in the ranch communicate with the IoT system through a gateway. The IoT system processes the data and sends data to the farmer's device in the form of notifications. The farmer can also see view the condition of his ranch using the designed dashboard. Using this data collected from the smart ranch will help the farmer decide agricultural solutions.

1.2 Intended Audience

This document is intended to be used by farmers, IoT Supports, IoT supporters, business stakeholders, developers and testers. This document can also be helpful for individual looking for smart farming solutions in agriculture domain. The document includes high level application understanding including application scope and functionality.

1.3 Application Scope

SmartAgCloud is a cloud service solution for smart agriculture which helps farmers to monitor farms which are remotely located; thereby saving time and cost involved. The SmartAgCloud application is expected to achieve goals of implementing the solution on cloud and leverage the cloud benefits of on demand resource allocation, run time scaling, reduced cost, improved coverage. Below are the key features in scope for the SmartAgCloud.

1. Smart farm to monitor below parameters,
 - Temperature,
 - Humidity,
 - Wind Speed and Wind Direction
 - Rain
 - Measures Soil Moisture

The parameters will be generated by simulated software. This software is expected to simulate below two real types of sensors,

- AcuRite
 - Soil Moisture Meter Probe - 24 Inch
2. Allows farmers to configure smart ranch fields within farmlands.
 3. Allows farmers to select their own desired ranches within farmland
 4. Choose the sensors.
 5. Dashboard on web and mobile facilitate for real time monitoring
 6. Receive notifications of monitored parameters on registered devices in real time.

1.4 Out of Scope

Real sensors are out of scope of this application including the connectivity and setup configuration. Only data feeds will be simulated by software.

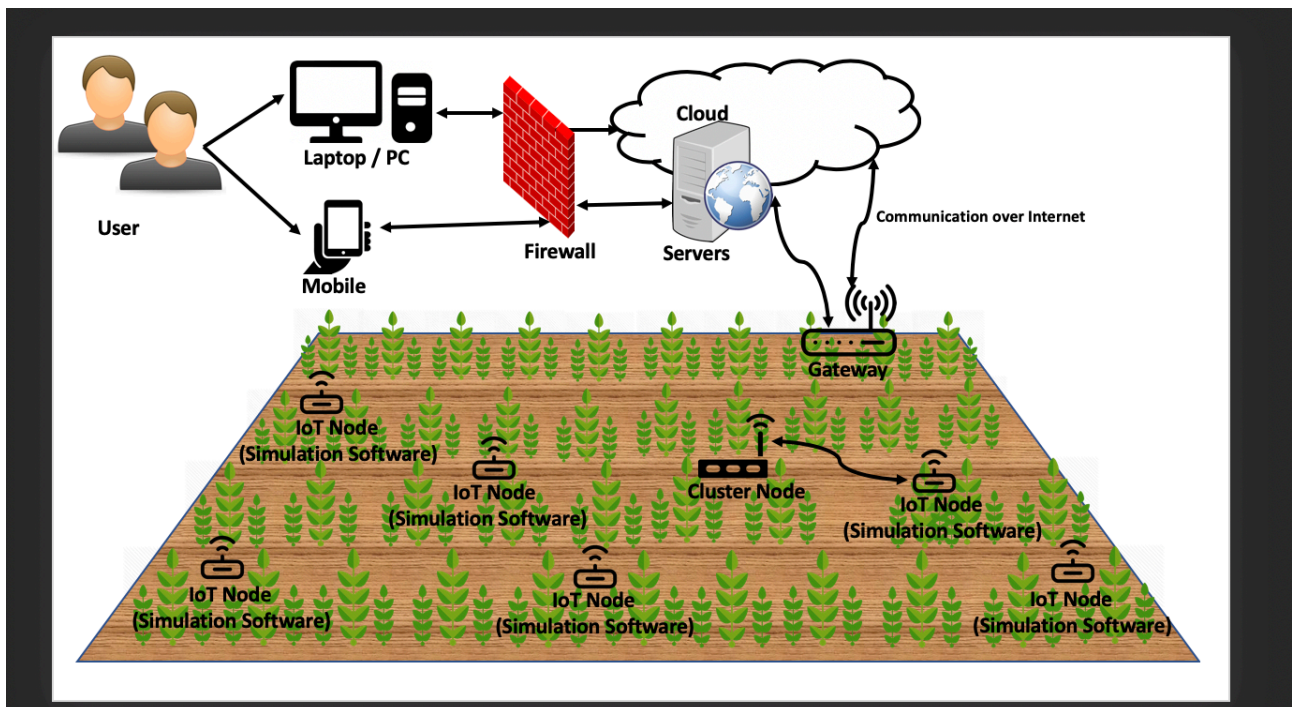


Fig 2: Application overview

2. Overall Description

2.1 User Classes and Characteristics

SmartAgCloud is expected to have three types of users,

1. **Farmers** – The primary user of this application. Farmers can access the system to check and configure their own smart farms.
2. **IoT Supports** – IoT support manager accesses the system to check the status of smart nodes and sensors.
3. **Infrastructure Managers** – People who setup, configure and manage smart nodes, cluster nodes, and sensors, as well as their connectivity. In addition, they could monitor and track the status of smart nodes with diverse sensors.

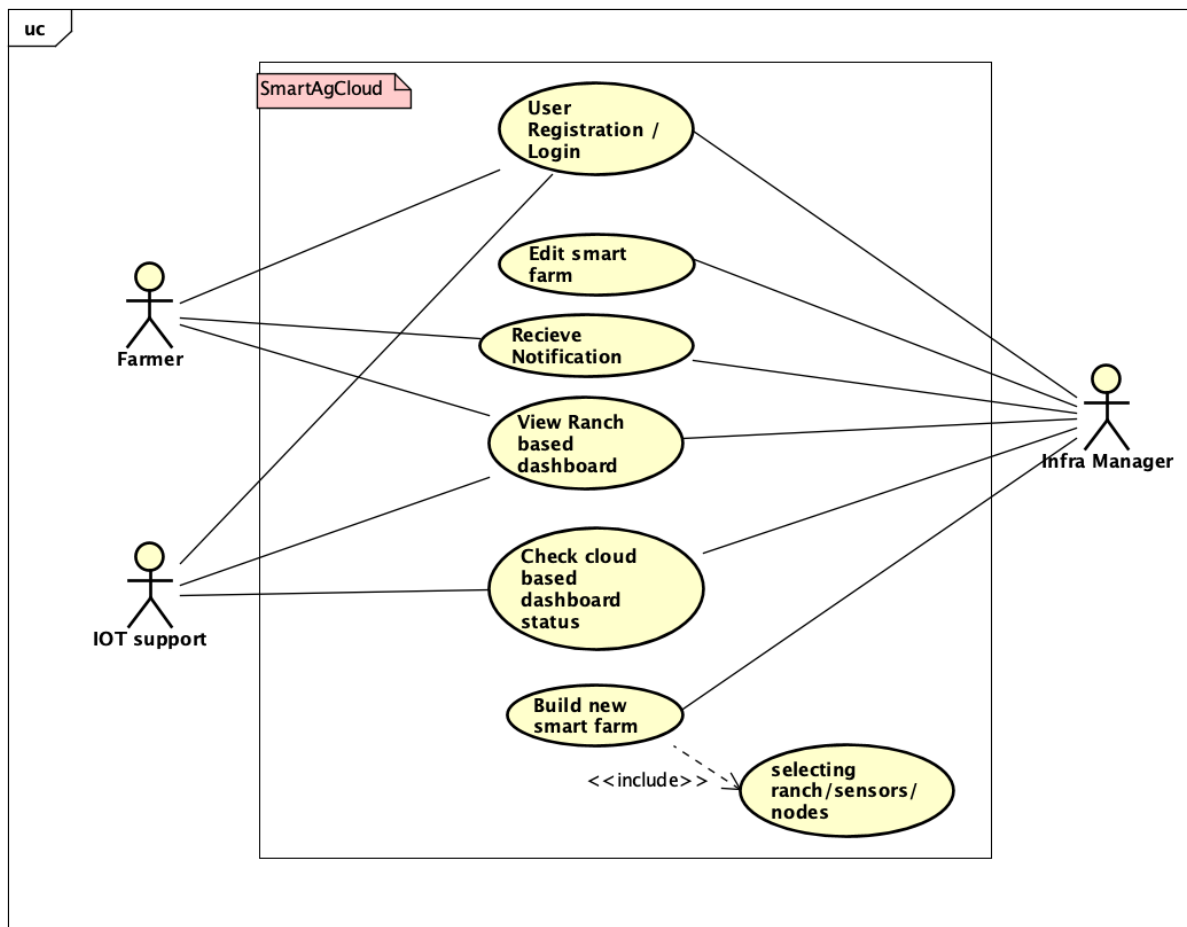


Fig 2: Use case diagram of the SmartAgCloud system

2.2 Design and Implementation Constraint

Linux: Open source OS which will be used to implement the backend of the SmartAgCloud system

Android: Android is the OS which will be used for mobile application.

2.3 Design and Implementation Constraint

It is not possible to use real sensors in the ranch due to time and cost constraints. Simulation software will be used to get real time sensor data.

2.4 Assumptions and Dependencies

- The SmartAgCloud is primarily using cloud service by AWS. The cloud provider AWS is thus a dependency for the SmartAgCloud to function.
- The farmer should have active internet connection in order to use the SmartAgCloud application

3. Requirement Analysis

3.1 Functional Requirement Analysis

High level functional requirements are listed below, (refer [section 4](#) for detailed analysis)

1. User registration and authentication
2. Build Smart Farm in Cloud
3. Smart Farm monitoring
4. Smart sensor and cluster node data management
5. Manage Sensor Profile in Cloud
6. Web and mobile application interface
7. Alerts notification

The interactions within components and end users would be done using API calls. High level communications via API calls are given below as request/response sequence,

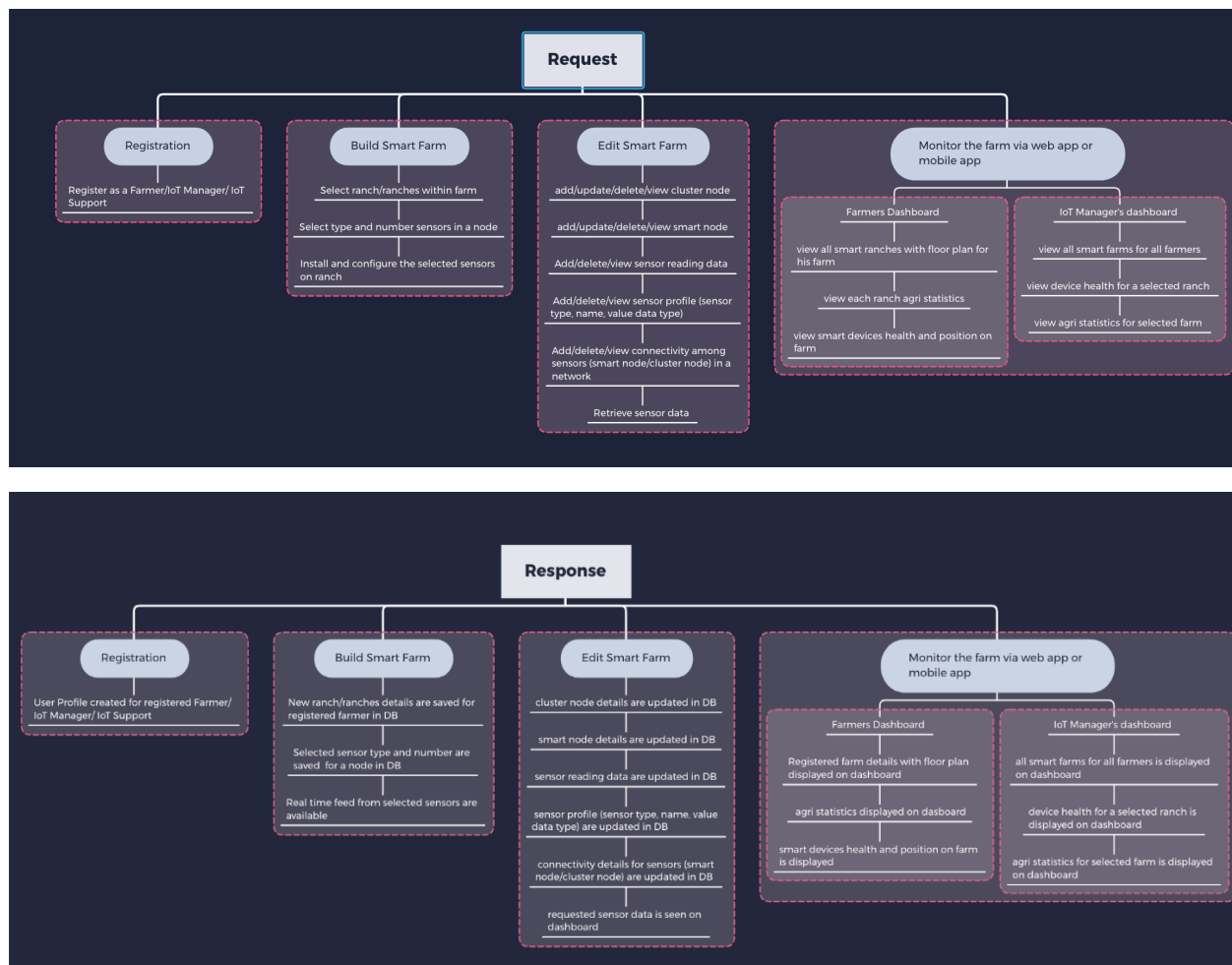


Fig 3: Communication Diagram

3.2 Nonfunctional Analysis

3.2.1 Performance Requirements

The SmartAgCloud application is expected to give real-time or near to real-time data feed from IoT devices – software sensors. The online transaction is expected to not exceed 10 secs. (standard performance for online transactions response time)

3.2.2 Security Requirements

The SmartAgCloud application should strictly adhere to user privacy by implementing features specific to user roles. SmartAgCloud will use secure protocol HTTPS communication for communication over the internet. Registered users with secure login will be enabled to take appropriate actions.

3.2.3 Software Quality Attributes

Usability: The SmartAgCloud interface should be easy to understand for the primary user – farmer. This will be achieved using simple and easy layout of UI.

Reliability/Availability: The application is expected to 99.99% available to the end user.

Scalability: The application should be able to sustain and dynamically handle the surge in traffic without affecting the application performance. The application should be able to scale in real time to serve any increase in traffic.

Extensibility: The application should be adaptive to the future changes with minimal changes within application design or interface.

4. Use Cases

4.1 User Registration

Use case: User Registration	
Users: Farmer/IoT Support /Infrastructure Manager	
Purpose: Register a user in SmartAgCloud Application	
Overview: Allows users to register in SmartAgCloud and assign respective roles to the users	
Type: Essential	
Preconditions: NA	
Postconditions: The user must be able to register to the SmartAgCloud application via web or mobile App and appropriate role must be assigned.	
Special Requirements: NA	
Flow of Events	
Actor Action	System Response
1. This use case begins after the new user opens the SmartAgCloud site or mobile application and selects Register User option 2. Users fills in the registration form and registers in the system.	1. The application will save the user details and assigns appropriate roles (Farmer/IoT Support / Infrastructure Manager) to the user.
Alternative Flow of Events	
Line 1: If the user is already registered, the system should throw error and redirect to login page.	

4.2 User Login

Use case: User Login	
Users: Farmer/IoT Support /Infrastructure Manager	
Purpose: Logins a user in SmartAgCloud Application	
Overview: Allows users to login in SmartAgCloud and use application as per assigned role	
Type: Essential	
Preconditions: The user must be registered in the application	
Postconditions: The user must be able to login to the SmartAgCloud application via web or mobile App and appropriate role must be assigned.	
Special Requirements: NA	
Flow of Events	
Actor Action	System Response

<ol style="list-style-type: none"> 1. This use case begins after the new user opens the SmartAgCloud site or mobile application and selects Login option 2. Users enters the username and password and logs in to the system. 	<ol style="list-style-type: none"> 1. Application prompts the to enter username and password. 2. The application will authenticate the user details, and Displays the home page.
Alternative Flow of Events	
Line 1: if username/password is invalid, return error message "Invalid username/password" to Output and provide with forgot password option.	

4.3 Build Smart Farm

Use case: Build smart farm	
Users: Infrastructure Managers	
Purpose: Build smart farms to enable monitoring via sensors within selected farm areas	
Overview: Allows <i>Infrastructure Managers</i> to build the one or more smart ranches within farm with farmers choice	
Type: Essential	
Preconditions: The users are registered in SmartAgCloud application as an Infrastructure Managers	
Postconditions: The smart farm is setup by Infrastructure Managers by the choice of farmer is able to view the selected farms in the dashboard	
Special Requirements: <ol style="list-style-type: none"> 1. Cluster node management – It allows Infrastructure Managers to add/update/delete/view cluster agriculture IoT sensor network, and track cluster node status. 2. Smart node management – It allows Infrastructure Managers to add/update/delete/view smart nodes controlled by a cluster, and track node status. 	
Flow of Events	
Actor Action	System Response
<ol style="list-style-type: none"> 1. This use case begins after the Infrastructure Managers logins to the SmartAgCloud application with his username and password and selects Build Smart Farm option 2. Infrastructure Managers selects one ranch at a time from the provided farm map and then selects the smart sensors and required numbers of sensors 3. Infrastructure Managers may choose to build multiple unique ranches within this use case 	<ol style="list-style-type: none"> 1. Verify the entered username and password and allow user to login on authentication. 2. Triggers request to IoT Supports to deploy and validate the connections of the sensors. 3. Display the selected smart ranch details and sensors detail on farmers profile.
Alternative Flow of Events	
Line 1: if username/password is invalid, return error message "Invalid username/password" to Output and provide with forgot password option.	

4.4 Sensor Data Management

Use case: Sensor Data Management	
Users: IoT Support	
Purpose: To allow IoT Support to manage the sensors data and connectivity.	
Overview: Allows IoT Support to update sensors data reading, sensor's profile and retrieve sensors data for a specified time interval	
Type: Essential	
Preconditions: The user is registered in SmartAgCloud application with IoT Support role	
Postconditions: The IoT Support is able to update sensors data reading, add/delete/update the sensor's profile and retrieve the sensor's data. The updated sensor profile is available for farmers to build the smart farm.	
Special Requirements: none	
Flow of Events	
Actor Action	System Response
<ol style="list-style-type: none"> 1. This use case begins after the IoT Support logs in to the SmartAgCloud application with his username and password and selects Build Smart Farm option 2. IoT Support selects one ranch at a time from the provided farm map and then selects the smart sensors and required numbers of sensors 3. IoT Support may choose to build multiple unique ranches within this use case 	<ol style="list-style-type: none"> 1. Verify the entered username and password and allow user to login on authentication. 2. Display the selected smart ranch details and smart sensors detail on farmers profile.
Alternative Flow of Events	
Line 1: if username/password is invalid, return error message "Invalid username/password" to Output and provide with forgot password option.	

4.5 View Ranch-based dashboard

Use case: View ranch-based dashboard of the monitored smart farm
Users: Farmer/IoT Support
Purpose: To allow farmers to view real time status of monitored smart field via dashboard on Mobile app Or web interface
Overview: Allows farmers to view ranch-based dashboard via mobile app or online interface
Type: Essential
Preconditions: The user is registered in SmartAgCloud application with farmer role
Postconditions: The farmer is able to view ranch-based dashboard for all built smart farms.

Special Requirements: none	
Flow of Events	
Actor Action	System Response
1. This use case begins after the farmer logs in to the SmartAgCloud application with his username and password 2. Farmer selects view dashboard option. 3. Farmer selects a ranch to monitor	1. Verify the entered username and password and allow user to login on authentication. 2. Displays the dashboard for logged in farmer profile 3. Displays the farm stats for the set period of selected
Alternative Flow of Events	
Line 1: if username/password is invalid, return error message "Invalid username/password" to Output and provide with forgot password option. Line 2: Farmer may edit the default the time duration for the dashboard; the system will display the stats As per edited duration.	

4.6 View Cloud-based dashboard

Use case: View Cloud-based dashboard of the monitored smart farm	
Users: Infrastructure /IoT Support	
Purpose: To allow Infrastructure /IoT Support to view real time status of all monitored smart field via dashboard on Mobile app or web interface	
Overview: Allows Infrastructure /IoT Support to view cloud-based dashboard via mobile app or web interface	
Type: Essential	
Preconditions: The user is registered in SmartAgCloud application with Infrastructure /IoT Support role	
Postconditions: The Infrastructure /IoT Support is able to view ranch-based dashboard for all built smart farms and devices	
Special Requirements: none	
Flow of Events	
Actor Action	System Response
1. This use case begins after the user logs in to the SmartAgCloud application with his username and password 2. user selects view dashboard option. 3. user selects a ranch to monitor	1. Verify the entered username and password and allow user to login on authentication. 2. Displays the dashboard for logged in user profile 3. Displays the farm stats for the set period of interval he selected
Alternative Flow of Events	

Line 1: if username/password is invalid, return error message "Invalid username/password" to Output and provide with forgot password option.

Line 2: user may edit the default the time duration for the dashboard; the system will display the stats As per edited duration.

4.7 Receive Alerts

Use case: Receive alerts on monitored	
Users: Farmer /IoT Support/ Infrastructure	
Purpose: Notify Farmer/Infrastructure Managers of any action required on all monitored smart field via dashboard on Mobile app or web interface	
Overview: Allows Farmer /IoT Support/ Infrastructure to receive alerts on monitored smart farms in case of emergency	
Type: Essential	
Preconditions: The smart farm must be monitored, and users must be registered as farmers/IoT Support	
Postconditions: Farmers/IoT Supports are notified of any action required or monitored parameter crossing minimum/maximum threshold	
Special Requirements: none	
Flow of Events	
Actor Action	System Response
1. This use case begins after the user logs in to the SmartAgCloud application with his username and password 2. user selects view dashboard option. 3. user selects a ranch to monitor	1. Verify the entered username and password and allow user to login on authentication. 2. Displays the dashboard for logged in user profile 3. Displays the farm stats for the set period of selected
Alternative Flow of Events	
Line 1: if username/password is invalid, return error message "Invalid username/password" to Output and provide with forgot password option. Line 2: user may edit the default the time duration for the dashboard; the system will display the stats As per edited duration.	

5. System infrastructure and architectures

The overview of our SmartAgCloud architecture consists of two components namely cloud infrastructure and users (farmers, IoT Support and Infrastructure Manager).

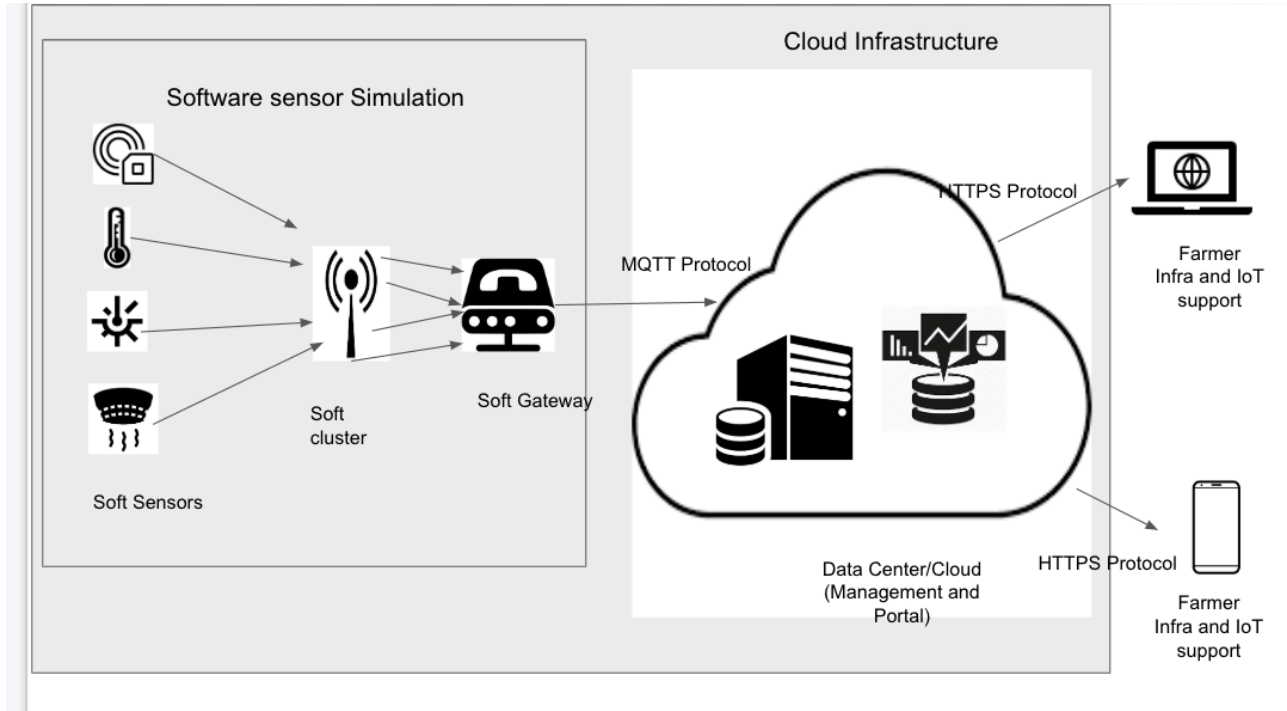


Fig 4: System Infrastructure

1. Cloud Infrastructure

i. Simulation software for Sensors

This software simulates the data generated by various sensors in the network at real time and passes it on to the application server.

ii. Portal Server:

The portal server consists of the dashboard and user authentication module. The users can access the application through the dashboard and customize their dashboard as per their needs.

iii. Management server:

The management server handles load balancing when there is increased demand for sensor data information, resource pooling and scalability (addition and deletion of sensors based on the farmer's needs)

iv. Databases:

The database used here is the Cassandra DB which is a part of the AWS cloud. The sensor data sent from the simulation software is handled by the application server which then stores it in the Database. Whenever there is a request from the entities or notification to be sent to the entities, the application server hits the database, retrieves the information and sends it across to the management server and portal server

2. Users

End user can access the application using web interface or mobile interface. These interfaces will connect to the application over the internet securely using HTTPS.

6. System design and Component interaction design

The infrastructure consists of components such as simulation software, application server, management server, portal server and a database.

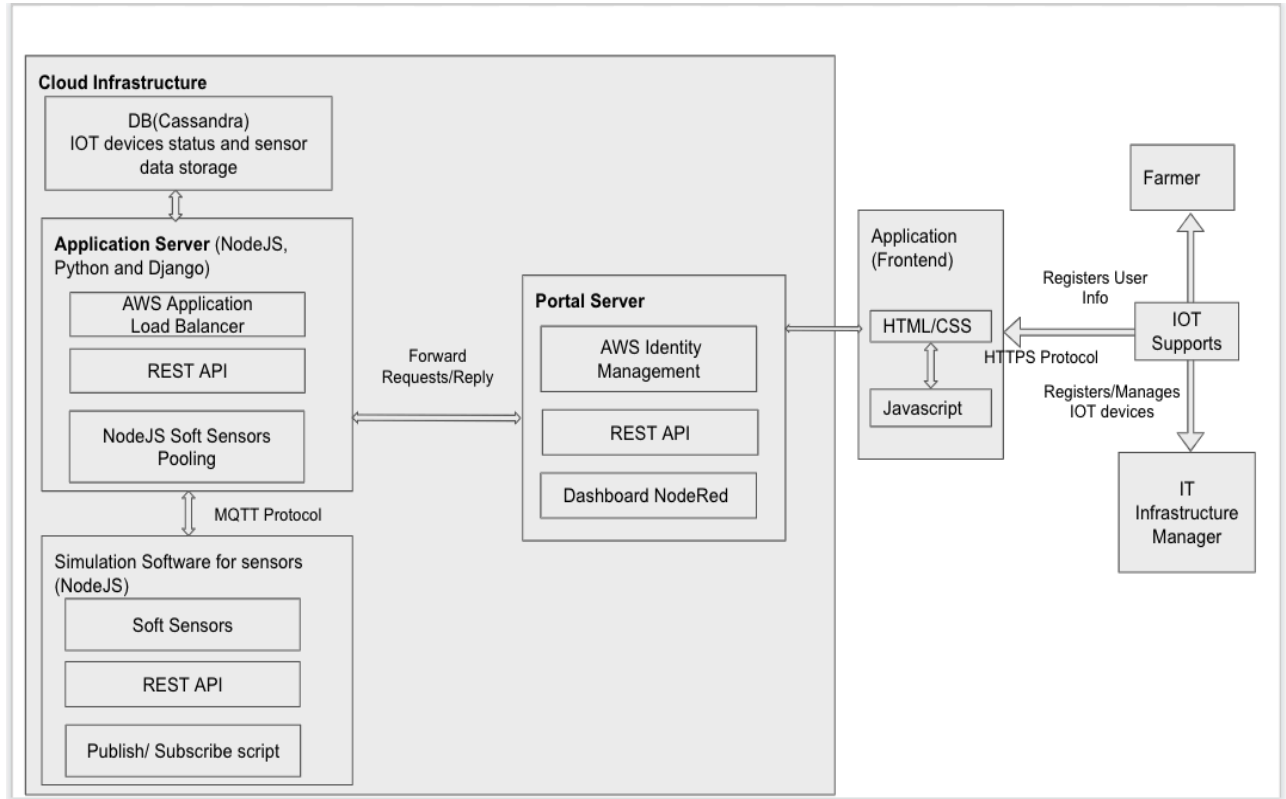


Fig 5: Black-box component design with API and functional description

The functional responsibility for each component is explained below.

1. **Application Interface**
Web and mobile interface will be available for users. The frontend would be implemented using HTML/CSS and JavaScript. While the mobile application will be implemented for Android phones.
2. **Portal Server**
Portal server will handle user registration, authentication using AWS identity management service. In addition to this, the monitoring of real time data feed will also be handled by the portal server using NodeJS – NodeRed framework.
3. **Application Server**

Application server will be responsible to manage the pool of virtual sensors and clusters. This will be implemented using NodeJS on AWS cloud platform. The real time application traffic will be managed using AWS application load balancer.

Application server will manage the data feeds from simulation software to the database.

4. Simulation Software

Data feeds from virtual sensors are generated and transferred to the application server using MQTT queues. While the data feeds would be in JSON format using the RESTful calls. This will be implemented using NodeJS.

5. Database

Data feeds are stored will be stored in highly scalable Cassandra database. The database will primarily include two schemas; sensor data schema and manage schema. These schemas will be hosted on two nodes.

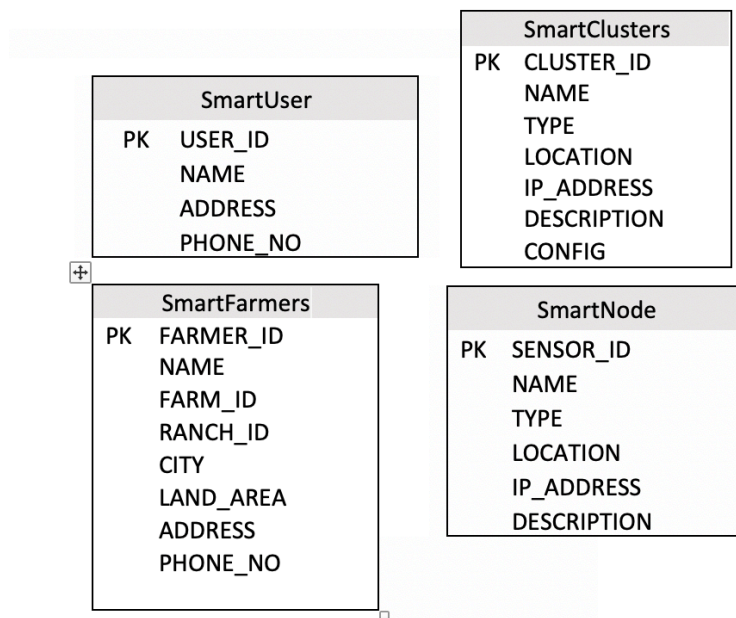
6.1 Database Design

Our application will use the use NoSQL Cassandra database to handle the big data and at the same time to achieve high scalability and availability without sacrificing performance.

Database design for SmartAgCloud:

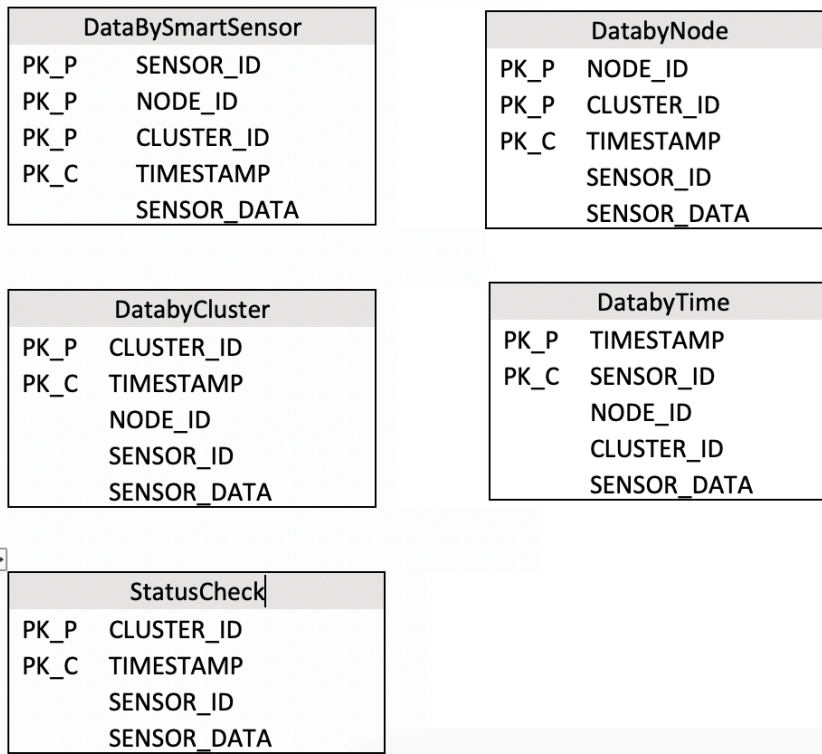
- NoSQL Cassandra database servers will be installed onto 3 nodes.
- Two Database server nodes will store the IOT sensor data replicated across both the nodes. Third database server will store relatively static data related to users and IOT sensors information.
- MQTT protocol will be used to read the data published by soft sensors and insert into the respective column families in Cassandra database,
- The RandomPartitioner technique will be used to generate 128-bit consistent hash key to determine on which node data need to place for load balancing.
- Initially Replication factor will be set to 2.
- Data will get replicated to another node eventually using hinted handoff to make the data consistent across the nodes to provide high availability.
- Read request can be satisfied by reading only one replica either any node.

Below tables will be created in one database server node to store users, smart clusters and smart:



PK_P - Primary Key Partitioned

- Two database nodes will store replicas of sensor time series data and smart node statuses data.



PK_C - Primary Key Clustered

6.2 Simulation Software Design

The agriculture sensors simulation would be designed using NodeJS backend. All the data communications thus would be in API calls (RESTful call /JSON format).

The simulation software is expected to contain below subcomponents:

1. Smart Sensors:

- Sensor data generator – This refers to the function that generates the simulated sensor's data.
- Sensor data collector – This will collect sensor data from different sensors based on a-predefined schedule.
- Sensor data transfer – This refers to the function that transfer sensor data the back-end server on a cloud.
- Smart node control – This refers to the basic control functions, such as node creation, update, deletion as well as status change.
- Smart node configuration –This supports the configuration of a smart node, including add/delete/update/set-up a list of sensors. For each sensor, it has the following status: Turn-On, Active, Inactive, Turn-Off, and Maintenance

2. Smart Clusters:

A cluster node, which supports the controls and connectivity with other related nodes (required)

This controls and supports a number of smart nodes, and collect sensor data from them, and send sensor data to the back-end server.

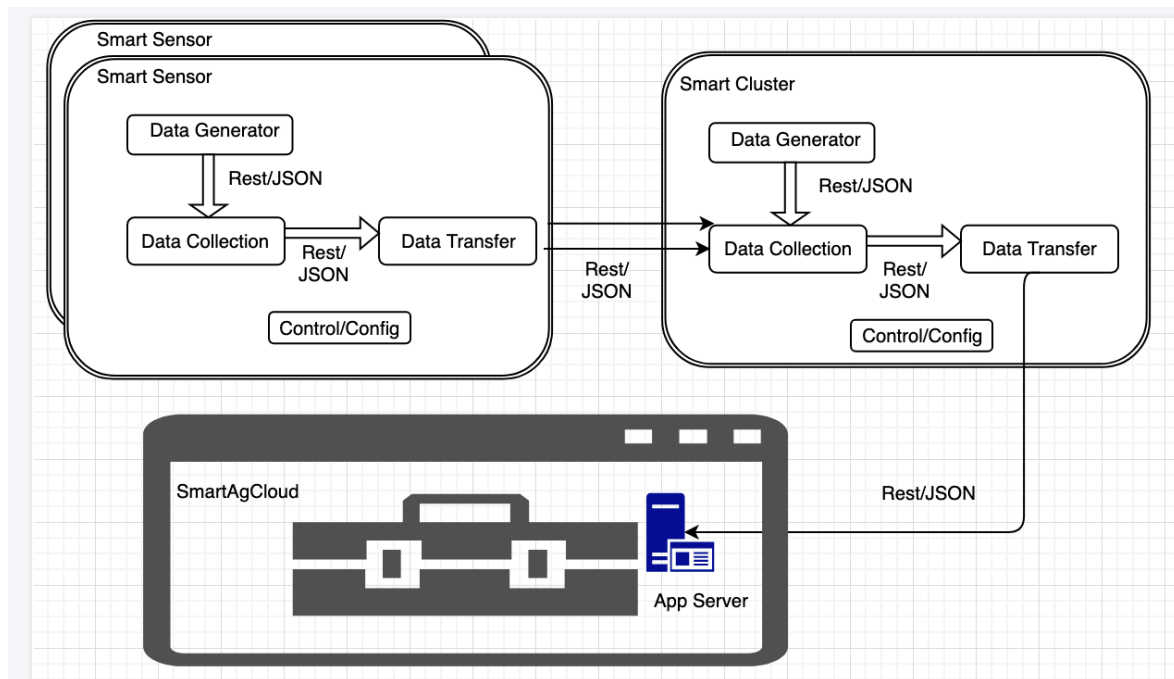


Fig 6: Sensor simulation software design

6.3 Deployment-Oriented System Infrastructure

AWS has selected as cloud infrastructure provider. AWS EC2 instances will get used for Application server, Database server and Web portal server.

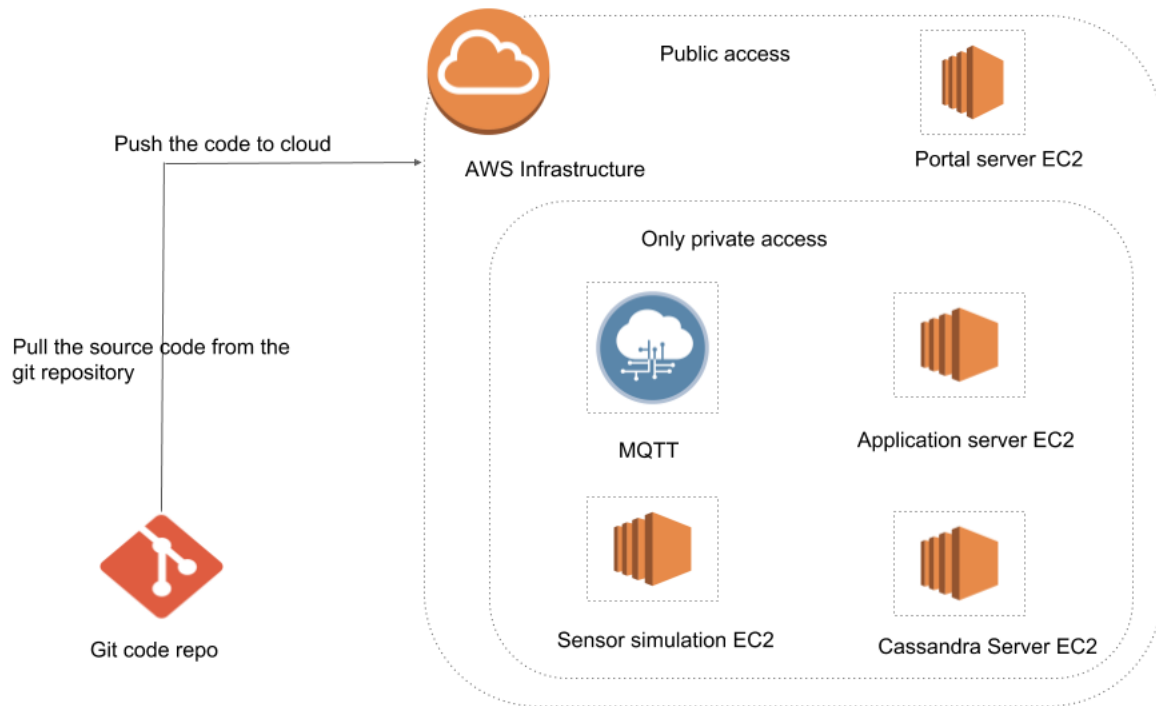


Fig 7: Deployment-Oriented System Infrastructure

The deployment process:

The source code from the git repository will get pulled and SSH to application server EC2 instance.

Security groups:

- Private access group – Application server, Database servers and sensor simulation instance will run in private security group.
- Public access group – Web portal server will run in public access group.

7. Plan and schedule

The below PERT chart describes the high-level planning and task breakdown overview that will be followed throughout the project lifecycle. The project will be delivered in phases in below order,

1. Initiation
2. Project Design and Analysis
3. Project Implementation
4. Final Submission

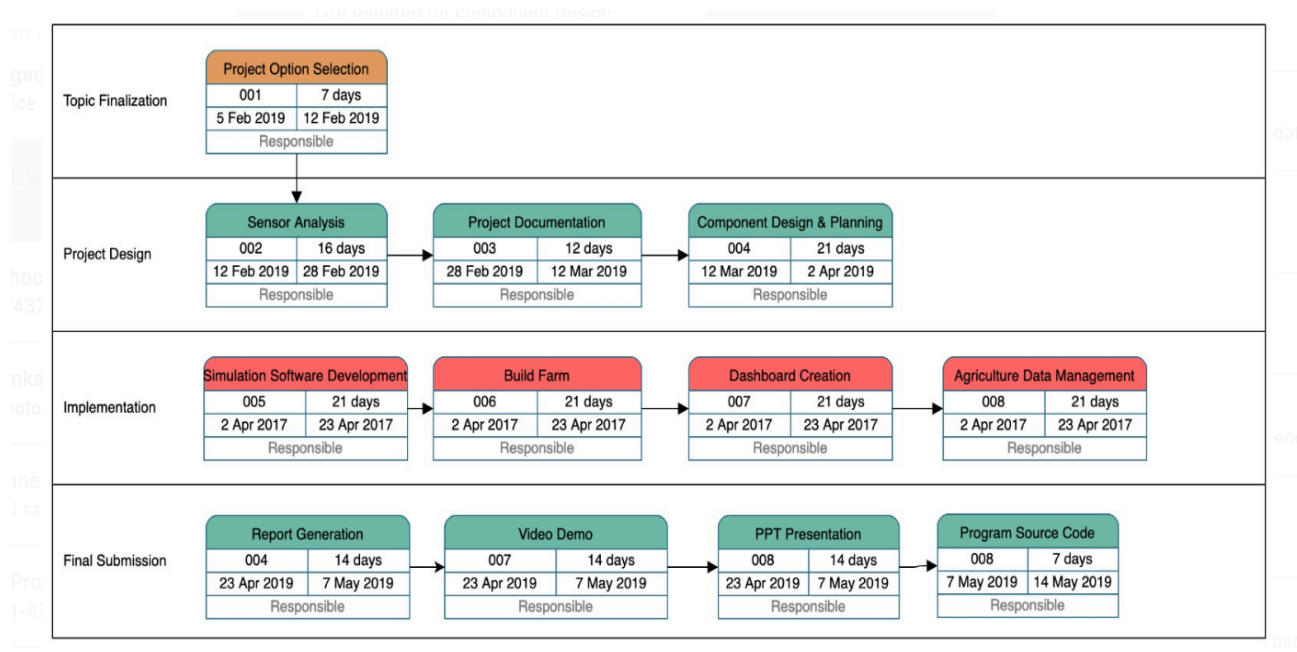


Fig 8: PERT Chart

8. Technology Selection

The below section describes cloud technology selection, usage and validation environments

Technology	Name	Remarks
Language	NodeJS, Java, SQL, Python	Programming language
Communication Protocol	MQTT, HTTPS	To get data from devices
UI	HTML, CSS, JS	Web and mobile interface
DB	Cassandra	To store and retrieve sensor data
MQTT Broker	Mosquitto	Broker to support devices on MQTT Protocol
Cloud	AWS	To build cloud infrastructure
Sensor Simulation	NodeJS	Soft sensors to generate, collect and transfer sensor data to the SmartAgCloud cloud
Mobile App Development	Android Dev Studio	To build the android mobile app
Framework	NodeRed	Used for dashboard implementation

8.1 Cloud Provider

SmartAgCloud application is a solution on cloud, for which AWS (Amazon Web Service) compute service will be leveraged. AWS offers good range of instances. AWS will also provide the real time scaling of configured instances and backend with dynamic change in traffic using application load balancer.

8.2 MQTT

The SmartAgCloud application is expected to give real-time or near to real-time data feed from IoT devices – software sensors. This requirement is expected to be fulfilled using the MQTT data bus transfer

MQTT offers below benefits,

1. MQTT has good integration with NodeJS (the backend logic)
2. MQTT is performance efficient where the monitored data is received with minimal connections

9. References:

1. Cassandra for Internet of Things: An Experimental Evaluation - André Duarte, Jorge Bernardino
2. Cassandra's Performance and Scalability Evaluation - Melyssa Barata and Jorge Bernardino

Abbreviation

Sr No	Acronyms	Description
1	MQTT	Message Queuing Telemetry Transport
2	AWS	Amazon Web Services
3	GUI	Graphical User Interface
4	PERT	Program Evaluation Review Technique
5	IoT	Internet of Things