Software Requirements Specification

for

SmartAgCloud

Version 1.0 approved

Prepared by

Priya Yadav

Jessica Mathias

Ankita Chikodi

Priya Khadke

San Jose State University

19th Feb 2019

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the 4 that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

<make sure to include the cloud benefits like reduced cost due to reuse, elastic dynamic good coverage, reliability etc>benefits over normal sensor networks

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 will discuss 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 It supports large-scale on-demand IOT-based agriculture system infrastructure services for farmers. Each farmer could select install and deploy one or more IOT agriculture networks for their greenhouses and ranch fields.

## Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

## Intended Audience and Reading Suggestions

This document is intended to be used by farmers, IoT managers , IoT supporters, business stakeholders, developers, architect 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 product scope and functionality .It is recommended to follow the sections in order as presented by document to get high level understanding followed by low level details.

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## Application Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

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 OnDemand resource allocation, run time scaling , reduced cost , improved coverage. Below are the key features in scope for the SmartAgCloud.

## References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# Overall Description

## Product Perspective

Technologies and IoT have the potential to transform agriculture in many aspects. Namely, there are five ways IoT can improve agriculture:

Data, tons of data, collected by smart agriculture sensors, e.g. weather conditions, soil quality, crop’s growth progress or cattle health. This data can be used to track the state of your business in general, as well as staff performance, equipment efficiency, etc.

Better control over the internal processes and, as a result, lower production risks. The ability to foresee the output of your production allows you to plan for better product distribution. If you know exactly how much crops you are going to harvest, you can make sure your product won’t lie around unsold.

benefits-of-smart-farming

Cost management and waste reduction thanks to the increased control over production. Being able to see any anomalies in the crop growth or livestock health, you will be able to mitigate the risks of losing your yield.

Increased business efficiency through process automation. By using smart devices, you can automate multiple processes across your production cycle, e.g. irrigation, fertilizing, or pest control.

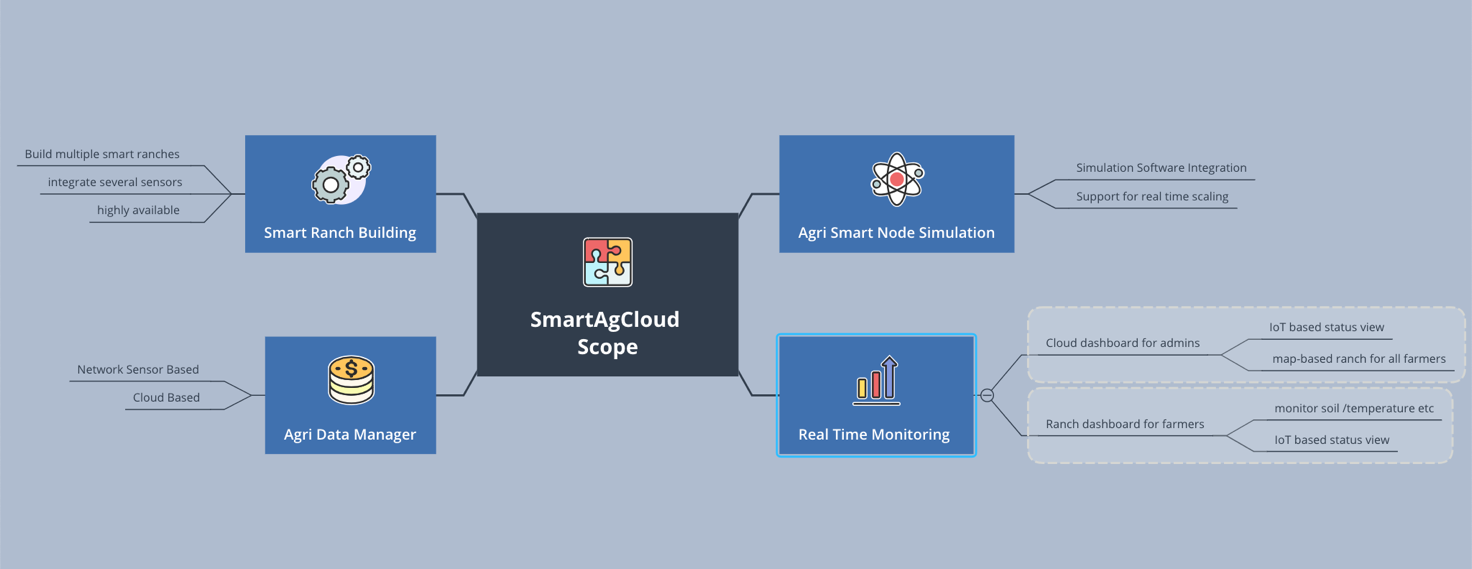
Enhanced product quality and volumes. Achieve better control over the production process and maintain higher standards of crop quality and growth capacity through automation.

As a result, all of these factors can eventually lead to higher revenue.

Now that we have outlined how IoT can be advantageously applied in the sphere of agriculture, let’s take a look at how the listed benefits can find their application in real life.

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

Sample figure



## Product Functions

Your ***SmartAgCloud*** is a IOT-based smart agriculture infrastructure service management system on a cloud to provide farmers with on-demand large-scale services in building their own IOT infrastructure networks by booking, configuration, installation, and deployment of desired IOT-based agriculture sensor networks in their ranches.

The key features of the application are stated below:

1. Monitors the agricultural conditions using data feed from smart sensors at real time
2. Allows farmers to configure smart ranch fields within farmlands.
3. Dashboard on web and mobile facilitate real time monitoring

* Allows farmers to select their own desired ranches within farmland
* Book the resources (different types of sensors)

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

## User Classes and Characteristics

**System users:**

This system includes the following types of users:

* - ***Farmers*** – They can access the system to check and configure their own IOT networks with smart nodes (with diverse sensors) online by accessing SmartAgCloud to check their IOT sensor status and statistic data,
* - ***IOT Supports*** – They can access the system to find out the status of smart nodes and sensors.

- **Infrastructure Managers** – The persons 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 their smart nodes with diverse sensors.

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

<NO real sensors used -only simulation software used for project implementation>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

<real sensors underlying in the field are connected with low network>

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

<list of sensors> -- tentative – with cost

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

|  |  |
| --- | --- |
| **Use case:** User Registration | |
| **Actors:** Farmer/IoT Manager /Infrastructure Manager | |
| **Purpose: Register 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 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** |
| 1. This use case begins after the new user opens the SmartAgCloud site or mobile application and selects RegisterUser option 2. Users fills in the registration form and registers to the system. | 1. System prompts the new user registration form.  2. The application will save the user details and assigns  appropriate roles (from Farmer/IoT Manager /  Infrastructure Manager) to the user. |
| 1. **Alternative Flow of Events** |  |
| Line 1: If the user is already registered, the user can login to the application by selecting login option. | |

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2

|  |  |
| --- | --- |
| **Use case:** Build smart farm | |
| **Actors:** Farmer | |
| **Purpose:** Build smart farms to enable monitoring via sensors within selected farm areas | |
| **Overview: Allows farmers to build the one or more smart ranches within his**  **farm with farmers choice of sensors** | |
| **Type: Essential** | |
| **Preconditions**: The user is registered in SmartAgCloud application as a farmer | |
| **Postconditions:** The smart farm is setup by farmers choice and farmer is able to view  the selected farms in the dashboard | |
| **Special Requirements:**   1. ***Cluster node management*** – It allows farmers to add/update/delete/view cluster nodes for an agriculture IOT sensor network, and track cluster node status. 2. ***Smart node management –*** It allows farmers to add/update/delete/view smart nodes controlled by a cluster, and track node status. | |
| **Flow of Events** | |
| **Actor Action** | **System Response** |
| 1. This use case begins after the farmer logins to the SmartAgCloud application with his username and password and selects Build Smart Farm option 2. Farmer selects one ranch at a time from the provided farm map and then selects the smart sensors and required numbers of sensors 3. Farmer may choose to build multiple unique ranches within this use case | 1. Verify the entered username and password and  allow user to login on authentication.   1. Triggers request to IoT managers to deploy and   validate the connections of the sensors. Display the selected smart ranch details and smart sensors details on farmers profile. |
| 1. **Alternative Flow of Events** |  |
| Line 1: if username/password is invalid, return error message "Invalid username/password" to Output  and terminate this Use Case.  Line 2: | |

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

|  |  |
| --- | --- |
| **Use case:** Sensor Data Management | |
| **Actors:** An IOT agriculture data manager | |
| **Purpose:** To allow managers to manage the sensors data and connectivity. | |
| **Overview: Allows IoT manager 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 manager role | |
| **Postconditions:** The IOT agricultural data manager 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** |
| 1. This use case begins after the farmer logins to the SmartAgCloud application with his username and password and selects Build Smart Farm option 2. Farmer selects one ranch at a time from the provided farm map and then selects the smart sensors and required numbers of sensors 3. Farmer may choose to build multiple unique ranches within this use case | 1. Verify the entered username and password and  allow user to login on authentication.  2. Display the selected smart ranch details and smart c  Sensors details on farmers profile. |
| 1. **Alternative Flow of Events** |  |
| Line 1: if username/password is invalid , return error message "Invalid username/password" to Output  and terminate this Use Case.  Line 2: | |

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 3

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

|  |  |
| --- | --- |
| **Use case:** Cloud-based sensor data management | |
| **Actors:** An IOT agriculture data manager | |
| **Purpose:** To allow managers to manage the sensors data and connectivity. | |
| **Overview: Allows IoT manager 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 manager role | |
| **Postconditions:** The IOT agricultural data manager 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** |
| 1. This use case begins after the farmer logins to the SmartAgCloud application with his username and password and selects Build Smart Farm option 2. Farmer selects one ranch at a time from the provided farm map and then selects the smart sensors and required numbers of sensors 3. Farmer may choose to build multiple unique ranches within this use case | 1. Verify the entered username and password and  allow user to login on authentication.  2. Display the selected smart ranch details and smart c  Sensors details on farmers profile. |
| 1. **Alternative Flow of Events** |  |
| Line 1: if username/password is invalid , return error message "Invalid username/password" to Output  and terminate this Use Case.  Line 2: | |

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 4

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

|  |  |
| --- | --- |
| **Use case: View ranch-based dashboard of the monitored smart farm** | |
| **Actors:** Farmer | |
| **Purpose:** To allow farmers to view real time status of monitored smart field via dashboard on Mobile app  Or Online 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 logins to the SmartAgCloud application with his username and password  2. Farmer selects view dashboard option. | 1. Verify the entered username and password and  allow user to login on authentication.  2. Displays the dashboard for logged in farmer’s  profile and displays the farm stats for the set period of interval the selected |
| **Alternative Flow of Events** | |
| Line 1: if username/password is invalid , return error message "Invalid username/password" to Output  and terminate this Use Case.  Line 2: Farmer may edit the default the time duration for the dashboard; the system will display the stats  As per edited duration. | |

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 5

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

|  |  |
| --- | --- |
| **Use case: View cloud-based dashboard of the monitored smart farm** | |
| **Actors:** Farmer/ | |
| **Purpose:** To allow farmers to view real time status of monitored smart field via dashboard on Mobile app  Or Online 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 logins to the SmartAgCloud application with his username and password  2. Farmer selects view dashboard option. | 1. Verify the entered username and password and  allow user to login on authentication.  2. Displays the dashboard for logged in farmer’s  profile and displays the farm stats for the set period of interval the selected |
| **Alternative Flow of Events** | |
| Line 1: if username/password is invalid , return error message "Invalid username/password" to Output  and terminate this Use Case.  Line 2: Farmer may edit the default the time duration for the dashboard; the system will display the stats  As per edited duration. | |

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 6

|  |  |
| --- | --- |
| **Use case: View cloud-based dashboard of the monitored smart farm** | |
| **Actors:** Farmer, IoT Manager and Infrastructure Managers | |
| **Purpose:** To allow farmers to view real time status of monitored smart field via dashboard on Mobile app  Or Online 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 logins to the SmartAgCloud application with his username and password  2. Farmer selects view dashboard option. | 1. Verify the entered username and password and  allow user to login on authentication.  2. Displays the dashboard for logged in farmer’s  profile and displays the farm stats for the set period of interval the selected |
| **Alternative Flow of Events** | |
| Line 1: if username/password is invalid , return error message "Invalid username/password" to Output  and terminate this Use Case.  Line 2: Farmer may edit the default the time duration for the dashboard; the system will display the stats  As per edited duration. | |

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements – Reliability

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

<include user auth details>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>