ENSE 483 Project Requirements Specification

October 10, 2019

Daniel Shevtsov (SID: 200351253)

# Table of Contents

[ENSE 483 Project Requirements Specification 1](#_Toc21457631)

[1. Table of Contents 2](#_Toc21457632)

[Executive Summary 3](#_Toc21457633)

[1.1 Project Overview 3](#_Toc21457634)

[1.2 Purpose and Scope of this Specification 3](#_Toc21457635)

[2. Product/Service Description 4](#_Toc21457636)

[2.1 Product Context 4](#_Toc21457637)

[2.2 User Characteristics 4](#_Toc21457638)

[2.2.1 Farmer 4](#_Toc21457639)

[2.2.2 Worker of the slaughterhouse 4](#_Toc21457640)

[2.2.3 Expert 5](#_Toc21457641)

[2.3 Assumptions 5](#_Toc21457642)

[2.4 Constraints 5](#_Toc21457643)

[2.5 Dependencies 5](#_Toc21457644)

[3. Requirements 6](#_Toc21457645)

[3.1 Functional Requirements 6](#_Toc21457646)

[3.2 User Interface Requirements 7](#_Toc21457647)

[3.3 Usability 7](#_Toc21457648)

[3.4 Performance 7](#_Toc21457649)

[3.4.1 Capacity 7](#_Toc21457650)

[3.4.2 Availability 7](#_Toc21457651)

[3.5 Manageability/Maintainability 7](#_Toc21457652)

[3.5.1 Monitoring 7](#_Toc21457653)

[3.5.2 Maintenance 7](#_Toc21457654)

[3.6 System Interface/Integration 7](#_Toc21457655)

[3.7 Security 8](#_Toc21457656)

[3.7.1 Protection 8](#_Toc21457657)

[3.7.2 Authorization and Authentication 8](#_Toc21457658)

[4. User Scenarios/Use Cases 9](#_Toc21457659)

# Executive Summary

## Project Overview

The project covers an implementation of *Meat UC5.1: Pig Farm Management* as specified in the *Internet of Food & Farm 2020 Use Case Architectures and Overview of the Related IoT Systems*. In particular, it is an Internet of Things (IoT) application that automates the management of pig farms via sensors that collect data regarding individuals and groups in a pig farm and report relevant and actionable information to farmers, workers, and experts.

The relevant pig health parameters to be monitored are water and food consumption, weight, temperature, humidity, and light intensity. In addition, batch data is collected by farmers and experts including pen and characteristics and boar taint. Finally, RFID tags are physically attached to pigs’ ears to track their individual identities and associate it with the other collected parameters for each pig.

The data for all the monitored parameters is fed into a cloud for data storage and analysis. Relevant information is extracted and displayed to farmers, slaughterhouse workers, and experts. Farmers monitor the displayed information and optimize their pig farms by taking action on sub-optimal parameters. In addition, users are given access to a dashboard from which they may provide the batch data.

## Purpose and Scope of this Specification

This specification defines the project’s problem statement, functional and non-functional requirements, user types, and any additional information that is required. Note that there are several requirements that are out of the scope of this project. This project is focused on the data acquisition aspect, that is, the involved sensors and the means by which information is collected from them and aggregated. As such, the data analysis that occurs in the cloud for the purposes of providing actionable information to users is considered out of the scope of this project. The requirements that are in and out of scope are listed below.

**In scope**

* Requirements pertaining to the functionality of the various sensors and the data that is collected.
* The aggregation and storage of information collected by a group of sensors.

**Out of scope**

* Requirements pertaining to the cloud and data analysis.

# Product/Service Description

There are several factors that affect this software’s requirements. These factors are listed below.

## Product Context

This software is independent and self-contained. However, the implementation of this project requires the incorporation of an existing Eclipse IoT open source project.

## User Characteristics

Below is a specification of the characteristics of each of the identified users of the software.

### Farmer

**Description**

The farmer is directly responsible for managing the health of pigs in the farm. As such, they are interested in having as much information as possible available for them to make decisions as to parameters to tweak. The farmer is highly knowledgeable on pig care methodology but their technical skill is expected to be lower than that of experts.

**Experience**

Very high.

**Technical Skill**

Low.

**Goals and Needs**

* Maximize health of pigs in the farm.
* Manage feed, water, and other barn characteristics for pigs individually and in groups.

**Frustrations**

* Required to tediously enter data manually into software systems.
* Receives no relevant information or information that is outdated and not actionable.

### Worker of the slaughterhouse

**Description**

The worker of the slaughterhouse is the user of the farmer’s produce, i.e., the pigs. Their role is to provide feedback as to the quality of the pigs and report on various imperfections such as boar taint or inappropriate fat levels. The worker has less experience than the farmer with respect to managing pig quality, but they can be expected to have a similar level of technical skill.

**Experience**

High.

**Technical Skill**

Low.

**Goals and Needs**

* Receive high quality pigs from the farmer that can then be sold to customers.
* Provide useful feedback to the farmer without a complicated process.

**Frustrations**

* Inability to verify the farmer’s pig quality before purchasing them.
* Lack of a convenient method to send feedback to farmer.

### Expert

**Description**

The expert is highly knowledgeable in the IoT system and serves as a sort of administrator. They are mostly removed from the farming process physically so they lack experience of the subject matter, but they are expected to have the highest level of technical skill out of the identified users.

**Experience**

Medium.

**Technical Skill**

Very high.

**Goals and Needs**

* Manage the IoT system and trace problems down to the individual pig and sensor.
* Provide technical support to both the farmer and worker of the slaughterhouse.

**Frustrations**

* Systems that do not have granular controls and/or gather data that is not relevant.
* Systems that are difficult to maintain and automate.

## Assumptions

Following is a listing of the assumptions made in this project:

* The cloud component (data analysis, presentation, controls) is developed separately from this project.
* Input readings are simulated to mimic expected conditions.

## Constraints

Following is a list of constraints for this project:

* Sensors are virtual and no physical sensors are provided.
* No live animals are used as part of this project.

## Dependencies

Following is a list of dependencies of this project:

* The selected Eclipse IoT open source project and its dependencies.

# Requirements

Priority Definitions

Requirements in this project are prioritised according to the following scheme:

* Priority 1 – High or “must have” requirement. Without it, the software will not fulfill its core functions and meet its stated goal.
* Priority 2 – Medium or “should have” requirement. A requirement that is expected to be present but its absence does not cause the software to fail to meet its stated goal.
* Priority 3 – Low or “nice to have” requirement. A requirement that is not expected to be present but its inclusion improves the user’s experience.

## Functional Requirements

| ID | Requirement | Priority |
| --- | --- | --- |
| **R01** | The user should browse the list of pigs. | 1 |
| **R02** | The user should uniquely identify the pig based on an RFID tag. | 1 |
| **R03** | The user should view a particular pig and its current health. | 1 |
| **R04** | The expert should add pigs to the list. | 2 |
| **R05** | The expert should remove pigs from the list. | 2 |
| **R06** | The user should browse all sensors associated with a particular pig. | 1 |
| **R07** | The user should view the pig’s feed consumption based on a weight sensor reading. | 1 |
| **R08** | The user should view the pig’s water consumption based on a weight sensor reading. | 1 |
| **R09** | The user should view the pig’s barn temperature based on a sensor reading. | 1 |
| **R10** | The user should view the pig’s barn humidity based on a sensor reading. | 1 |
| **R11** | The user should view the pig’s boar taint status based on entered batch data. | 2 |
| **R12** | The user should view the pig’s pen and barn characteristics based on entered batch data. | 2 |
| **R13** | The expert should add sensors to the list. | 3 |
| **R14** | The expert should remove sensors from the list. | 3 |
| **R15** | The worker of the slaughterhouse should enter the pig’s boar taint status as batch data. | 2 |
| **R16** | The farmer should enter the pig’s pen and barn characteristics as batch data. | 2 |
| **R17** | The user should view an average of pig health at a particular barn. | 3 |
| **R18** | The user should export data to a file for personal storage. | 3 |

## User Interface Requirements

In addition to functional requirements, there are non-functional requirements relating to user interface and user experience.

## Usability

There are several requirements in terms of usability:

* There should be simple documentation provided to help users interacting with the software.
* The system should be simple to learn.
* It should be easy to understand information presented on the dashboard at a glance.

## Performance

* Under typical conditions, the system should be responsive. That is, there should not be significant delays in the user interface that would impede the user’s experience.
* User requests should produce responses from the software at a reasonable speed. Request-response cycles should not exceed 1 second.

### Capacity

* At a minimum, the system should support 3 simultaneous users to cover one of each user type. It is expected that the system be able to support multiple of each user type simultaneously. A reasonable number of simultaneous connections is 10 users.
* Given the scale of the project, it is expected that the software system be able to handle a moderate number of pigs, each of which with a number of sensors attached to them. It is not required or tested whether the system scales to hundreds or thousands of pigs.

### Availability

* The system is expected to remain functional at all times, i.e., 24/7.
* All sensors are expected to be active and reporting at all times.
* In case of unexpected outages, gaps in data storage should be clearly distinguishable.

## Manageability/Maintainability

### Monitoring

* Logging should be in place for purposes of debugging and historical system stability recording. Logs should be rotated to prevent overflowing storage on the deployment machine. At a minimum, logs should be kept for 7 days before being rotated out.

### Maintenance

* It should be simple to recover from unexpected fatal failures.
* Upon restart, the system should automatically continue processing data without further troubleshooting.

## System Interface/Integration

The system requires integration with several other software:

* Data storage – Either a relational database (e.g., MySQL, PostgreSQL, etc.) or an in-memory data store (Redis, Memcached, etc.).
* Communication between sensors – The selected Eclipse IoT open source project software.

## Security

### Protection

Following is a list of security factors that are considered to protect and maintain the system’s data integrity:

* Users’ activity should be logged in some form.
* User roles should be enforced so as to prevent users from accessing resources that they do not have the permissions to access.
* Unless required by the implementation, data encryption is not considered for this project due to its scale.

### Authorization and Authentication

* At a minimum, a layer of authentication should be provided to ensure that only users who are registered in the system may access it. Users should be presented with a login screen to enter their credentials before being allowed to interact with the system.
* A layer of authorization should be used to enforce user roles and prevent users from accessing restricted resources that they should not be able to access.

# User Scenarios/Use Cases

Below are several major scenarios of the identified users interacting with the system derived from the identified list of requirements.

|  |  |
| --- | --- |
| **ID** | UC01 |
| **Name** | User views a particular pig’s health. |
| **Description** | A user is interested in browsing the list of all pigs, drilling down to a specific pig of interest, and viewing detailed information on their health based on all of the sensors assigned to it and the most recent data they collected. |
| **Actors** | User (any role). |
| **Preconditions** | * User logged into the system. |
| **Postconditions** | N/A |
| **Main Flow** | 1. User browses the list of pigs (see **R01**). 2. User locates the specific pig they are interested in (based on their unique ID in the system (see **R02**)). 3. User views details on the specific pig (see **R03**). 4. The user sees the current readings of all of the pig’s sensors (see **R6**, **EX01**), including:    * Feed consumption (see **R07**)    * Water consumption (see **R08**)    * Barn temperature (see **R09**)    * Barn humidity (see **R10**)    * Boar taint status (see **R11**)    * Pen and barn characteristics (see **R12**) |
| **Alternate Flows** | N/A |
| **Exceptions** | **EX01** Pig has no sensors attached.   1. In place of the sensors, user receives warning message notifying them that no sensors are available for this pig. |

|  |  |
| --- | --- |
| **ID** | UC02 |
| **Name** | Farmer and worker of the slaughterhouse modify batch data. |
| **Description** | The farmer or worker have finished physically inspecting a pig’s barn and physical health and wish to add their batch data into the system manually for bookkeeping purposes. |
| **Actors** | Farmer **OR** worker of the slaughterhouse. |
| **Preconditions** | * User logged into the system. * User followed **UC01** and found their pig of interest. |
| **Postconditions** | Boar taint status **OR** pen and barn characteristics has been updated. |
| **Main Flow** | 1. Farmer edits the pen and barn characteristics. (see **R16**). **OR** Worker of the slaughterhouse edits boar taint status (see **R15**). 2. Edited value is updated and user is returned to view details on the specific pig (see **R03**). |
| **Alternate Flows** | N/A |
| **Exceptions** | N/A |

|  |  |
| --- | --- |
| **ID** | UC03 |
| **Name** | Expert manages pigs and sensors. |
| **Description** | The expert uses their administrative permissions to manually manage the list of pigs and sensors associated with each. The expert is able to add and remove pigs and sensors per pig. |
| **Actors** | Expert. |
| **Preconditions** | * Expert logged into the system. |
| **Postconditions** | List of pigs **AND/OR** list of sensors per pig has been updated. |
| **Main Flow** | 1. Expert browses the list of pigs (see **R01**). 2. Expert clicks a button to add a new pig to the list. 3. Expert fills out the pig’s details including their unique ID (see **R02, R04**). 4. Pig is added and expert is sent to view the pig’s details (see **R03**). 5. Expert sees list of all of the pig’s sensors (see **R6**, **EX01**). 6. Expert clicks a button to add a sensor (see **R13**). 7. Expert selects the sensor to assign to the pig. 8. Selected sensor is associated with the pig and the expert is returned to view the pig’s details (see **R03**). 9. Expert returns to browse the list of pigs (see **R01**). 10. Expert locates a specific pig they are interested in to remove (based on their unique ID in the system (see **R02**)). 11. Expert clicks a button to remove the selected pig (see **R05**). 12. Pig is removed and expert is returned to the list of pigs (see **R01**). |
| **Alternate Flows** | N/A |
| **Exceptions** | **EX01** Pig has no sensors attached.   1. In place of the sensors, user receives warning message notifying them that no sensors are available for this pig. |