Developing a Design

Agricultural Sensor System

# About

All text created for this assignment is held within this document. Images (i.e. the class diagram) are held in their own files within this folder.

Sections where there were images created for it contain a summary in a place of them that provides information on the image(s). For example:

**Filename:** *\*\*This is the name of the image file that is within this folder \*\**

**Extension:** *\*\*This is the extension of the image in question (JPG, PNG, or SVG)\*\**

**Description:** *\*\*This is a brief description of the image and what it shows\*\**

The term ‘*AMS*’ is used throughout this document; this stands for ‘*Agricultural Management System*’. This refers to the Agricultural Sensor System.

Table of Contents

[About 1](#_Toc467859551)

[Class Diagram 2](#_Toc467859552)

[Class Diagram Explanation 2](#_Toc467859553)

[Sensor 2](#_Toc467859554)

[Location 2](#_Toc467859555)

[SensorReading 2](#_Toc467859556)

[SensorMonitor 3](#_Toc467859557)

[StaticFieldStation 3](#_Toc467859558)

[PortableFieldStation 3](#_Toc467859559)

[AutomatedMachineryMonitor 3](#_Toc467859560)

[AutomatedMachinery 3](#_Toc467859561)

[FieldArea 4](#_Toc467859562)

[Server 4](#_Toc467859563)

[Field 4](#_Toc467859564)

[Planting 4](#_Toc467859565)

[Harvest 4](#_Toc467859566)

[AgricultureManagementSystemGUI 5](#_Toc467859567)

[User 5](#_Toc467859568)

[Order 5](#_Toc467859569)

[Sequence Diagrams 5](#_Toc467859570)

[Personas 6](#_Toc467859571)

[Production Line Manager 6](#_Toc467859572)

[Farmer (Large-scale) 6](#_Toc467859573)

[Farmer (Small-scale) 7](#_Toc467859574)

[Scenarios 7](#_Toc467859575)

[Scenario #1 (Dave) 7](#_Toc467859576)

[Scenario #2 (John) 8](#_Toc467859577)

[Scenario #3 (Craig) 8](#_Toc467859578)

[Concept Designs 8](#_Toc467859579)

[Storyboards 9](#_Toc467859580)

# 

# Class Diagram

**Filename:** Class-Diagram

**Extension:** SVG

**Description:** This class diagram covers the major classes used (and the relationships between them) within our system. The scopes of attributes/methods are not defined on the classes.

All methods are *public* and all attributes are *private* (even in the case of base classes used for inheritance).

# Class Diagram Explanation

This section better explains the choices that were made when designing the main classes in the system, and related class diagram.

Classes that have not been specifically explained below and the reason as to why are explained here:

* The specific Sensors (i.e. SoilTemperatureSensor) were not included as they were not deemed as main classes of the system as they are only derived classes and hold no specific attributes or methods
* The specific pieces of AutomatedMachinery (i.e. Feeder) as, again, they are only derived classes from AutomatedMachinery and only have specific methods to perform their specific crop treatment
* The sets of individual classes (i.e. SetOfSensorMonitors) as they are only an array of the individual and do not provide enough functionality to be considered a main class

## Sensor

This can take a SensorReading of the crop area it is in. The SensorReading is then relayed back to the SensorMonitor it is attached to (1-to-1 relationship).

Sensor has a Location as it needs to know where it is. Sensor also has a ‘dataUnit’ attribute which will represent what units the readings are read in (i.e. "pH").

## Location

This class provides a way for classes to have X and Y (latitude and longitude) coordinates associated with it.

## SensorReading

This class holds data related to a reading taken from a sensor.

SensorReading has a ‘timeTaken’ attribute that tells the Sensor what time the crop reading was taken. SensorReading stores a dataUnit relating to the sensor’s dataUnit. The ‘readingType’ attribute is there to show the type of reading that has been measured (i.e. Air Temperature). This was needed as well as ‘dataUnit’ as 2 sensors (Air Temperature and Soil Temperature) have the same unit of measurement and if ‘readingType’ was not included, there would be no way to differentiate between the two.

## SensorMonitor

This has the responsibility of controlling the Sensor it relates to, and relaying information between the Sensor and the StaticFieldStation.

## StaticFieldStation

This holds a SetOfSensorMonitors attached to it and stores a SetOfSensorReadings taken by each Sensor attached to it. This has 2 methods; to update the SensorReadings, and to send the SetOfSensorReadings it has collected to the Server. The StaticFieldStation invokes these 2 methods based on the tick of a timer. When ‘updateSensorReadings’ method is invoked, it gets the ‘lastReadingTime’ attribute and interval from the SensorMonitor. Based on a calculation of these 2 variables, the StaticFieldStation will tell the SensorMonitor to get a new SensorReading. A local store of a SetOfSensorReadings is held as, if there is no connection to the Server, the readings cannot always be pushed up to the Server when they are immediately taken.

## PortableFieldStation

A PortableFieldStation is a handheld device that the farmer takes out into the field, puts a Sensor in the ground and selects to take a SensorReading. A PortableFieldStation is not associated with a SetOfSensorMonitors because the Sensors attached to the PortableFieldStation can change. It does not have to use the same Sensors every time. Like a StaticFieldStation, a PortableFieldStation also has a local store of a SetOfSensorReadings that will be pushed to the Server.

If the PortableFieldStation loses connection to the server then the reading will fail and the Farmer must try again another time.

## AutomatedMachineryMonitor

This keeps track of the AutomatedMachinery connected to it. The AutomatedMachineryMonitor polls the Server - on the tick of a timer - to ask whether the FieldArea (stored in the ‘fieldAreaCovered’ attribute) that the AutomatedMachinery covers requires treatment when the ‘checkIfTreatmentNeeded’ method is invoked.

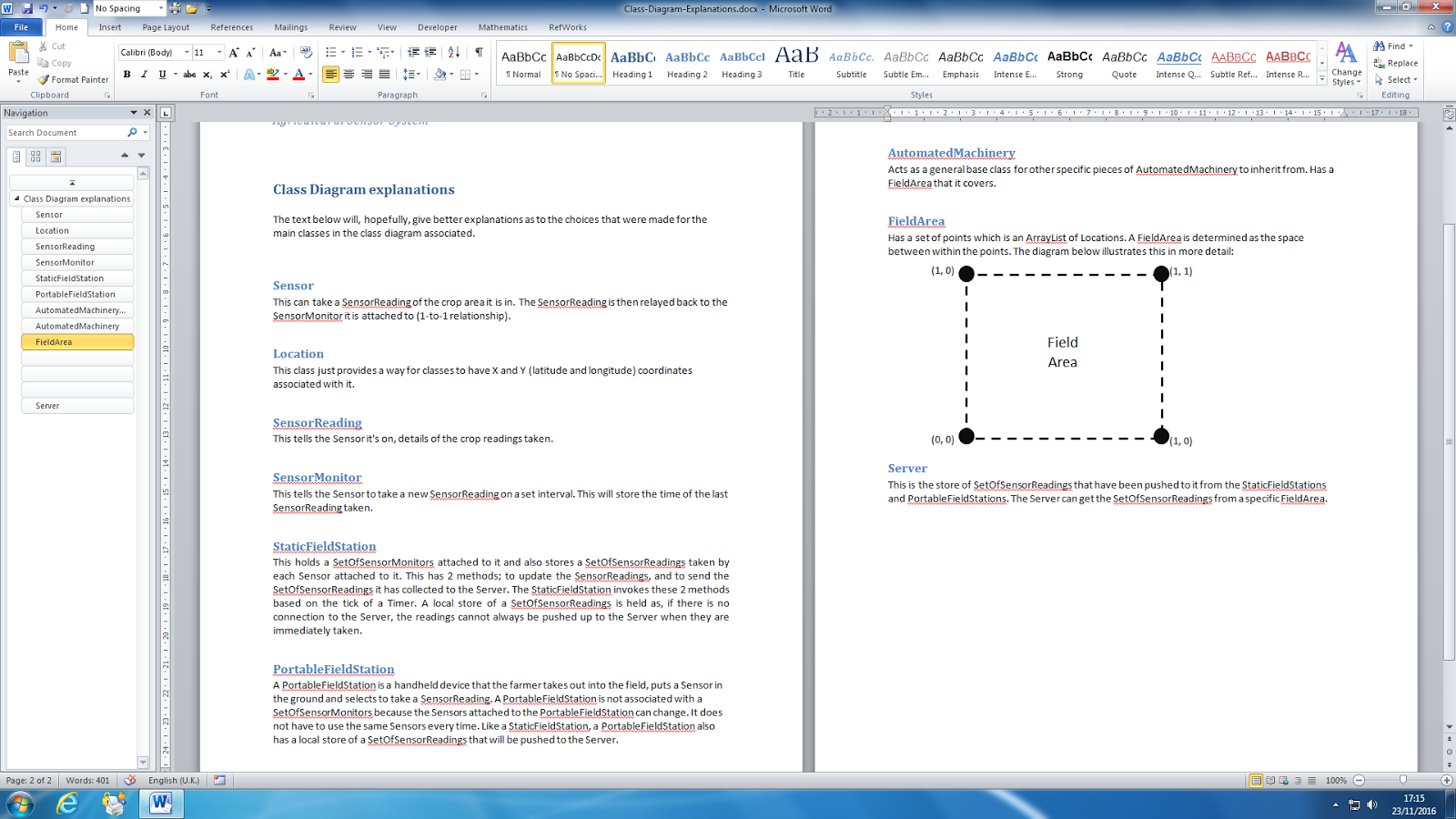
The AutomatedMachineryMonitor knows about the Planting that is in the FieldArea that it covers. From this, the AutomatedMachineryMonitor can get the ideal values (i.e. ‘preferredSoilMoistureLevel’). The AutomatedMachineryMonitor can then use the SetOfSensorReadings retrieved from the Server and the ideal values received for the Planting to determine what AutomatedMachinery to use if treatment is required.

## AutomatedMachinery

Acts as a general base class for other specific pieces of AutomatedMachinery to inherit from. Has a FieldArea that it covers.

## FieldArea

Has a set of points which is an array list of Locations. A FieldArea is determined as the space between within the points. The diagram below illustrates this in more detail:



It also contains a method for determining if a Location passed in is within the area that it covers.

## Server

This is the store of SetOfSensorReadings that have been pushed to it from the StaticFieldStations and PortableFieldStations. The Server can be polled to get a SetOfSensorReadings relating to a specific FieldArea.

When the AutomatedMachineryMonitor polls the Server, the Server gets the FieldArea that the AutomatedMachinery covers. The Server checks whether that FieldArea requires treatment based on the SetOfSensorReadings it has been given in that location by passing the location into the ‘isLocationInArea’ method on the FieldArea.

## Field

This represents a field that the Farmer owns in his farm. It holds the current crop (held in the ‘currentPlanting’ attribute) that's being grown in the field (i.e. Peas). A Field also holds a FieldArea which is the entire field. A Field also has information about the history of Harvests that have been done within that field. Finally, a Field holds a collection of the AutomatedMachineryMonitors for that field.

Fields can record a new harvest (‘recordNewHarvest’ method). This method is invoked when the Farmer has recently completed a full harvest of the field whereby the Farmer specifies; the time the harvest was done and the yield collected. By invoking this method, the ‘currentPlanting’ attribute is then set to 'null' as there will be no Planting.

## Planting

This represents 1 type of crop that is grown within a field. Each Planting has many attributes including levels of ideal variables which are used to determine if the crop requires treatment.

## Harvest

This holds the ‘createHarvest’ method which is used to record a harvest and store it in the ‘harvestHistory’ attribute of the related Field.

## AgricultureManagementSystemGUI

This acts as the default GUI that a user of the system will interact with. Production Line Managers and Farmers will have a different UI presented to them as they require different functionality. For example, a Farmer will want to check the readings in his fields whereas the Production Line Manager would not need to know this.

The AgricultureManagementSystemGUI knows about many other classes as the users will want to interact with many other parts of the system.

This GUI can be refactored into many separate GUI classes based on the developer(s) preferences.

## User

This is the base class representing a user that will interact with the system. It has common attributes to describe a user (i.e. username, firstName, surname…).

There is also an attribute called 'orders' which holds a SetOfOrders. An order is added to both users SetOfOrders when it is created (by the Production Line Manager); the Production Line Manager's SetOfOrders as they made the order, and the Farmer's SetOfOrders as they are the supplier of the order.

New Orders that come through to the supplier are put at the end of a queue for the Farmer to do.

## Order

This represents an order. An order just holds all the details about it (i.e. cost, supplier (Farmer), status).

# Sequence Diagrams

We have done 3 sequence diagrams on this case study. Below is a list of them, where to find them, and a brief explanation:

1. **Filename:** SD-Record\_Harvest

**Extension:** JPG

**Description:** This is a sequence to show how a Farmer logs that he has recently completed a harvest into AMS

1. **Filename:** SD-Make\_Order

**Extension:** JPG

**Description:** This sequence diagram shows how a Production Line Manager makes an order on AMS

1. **Filename:** SD-Collect\_Readings

**Extension:** JPG

**Description:** This sequence diagram shows how AMS updates the readings that are collected by the Sensors spread throughout the field(s)

# Personas

The 3 personas listed below represent 3 different kinds of user that will be interacting with the Agricultural Management System (AMS):

1. Production Line Manager
2. Farmer (Large-scale)
3. Farmer (Small-scale)

## Production Line Manager

Dave is a Production Line Manager at Frozen FoodStuffs Ltd and has been with the company for over a decade. Dave is 35 years old and enjoys mixed martial arts in his spare time. Dave finds that being a Production Line Manager is stressful and is always looking for ways to make his job easier.

Dave used to work on the production line itself, doing quality assurance of the products coming in. Since becoming a Production Line Manager, Dave has a much broader picture of the system. He has come to realise that a large amount of low quality peas are being frozen. This is because, when peas come into the company, they are sometimes sat waiting for a long time - whilst still being fresh - before being frozen. This is because Dave doesn't know what product has been ordered and when and, therefore, cannot prepare correctly for when the products come in. Every day, Dave must sacrifice a large amount of his time reviewing what has been ordered and when and estimating when the product is going to be delivered.

Dave would like a system to keep track of past orders. Details like:

* When the order was put through
* Products ordered
* Estimated delivery date

This system would be a huge benefit to Dave as would make more of his time available to be put to good use. This would also result in better quality products entering the freezing process as Dave (and others) would be more aware as to when products are coming in and what proper procedures to employ for their arrival.

## Farmer (Large-scale)

John is a large-scale farmer. He is 28 years old. He is what some people would call, a modern farmer and is new in the farming industry but is very ambitious to succeed. John is very big into modern technology and is fascinated by the rapid growth of it. Due to this, John owns a lot of technology and it's become a part of his daily life.

John owns a lot fields (and many types of crops) and finds it difficult and very time consuming to treat all his crops on his farm. John can’t find any willing employees to work for him as he is new to the industry and lacks experience.  John admits that he needs help and sees that he was in-over-his-head when purchasing such a large farm but sees that he's gone too far to turn back now. John is wondering whether technology can somehow solve his problem. He is willing to pay any amount of money (that he has left) for any system that will make life easier for him. He is considering whether it is possible to install a system that fully automates the care of his farm. Types of care include:

* Irrigation
* Spraying pesticides
* Fertilisation

This would solve John's problem and, hopefully, wouldn't result in any of his crops suffering thus. From this, John can imagine that all he would have to do is install the system, maintain the automated machinery employed, and harvest his crops when ready.

## Farmer (Small-scale)

Craig is a small-scale farmer and grows peas. He is 56 years old has been in the farming industry all his life, since taking over the family farm from his father 30 years ago. Craig prefers stuff the old-fashioned way. Craig believes that if you want a job done right, you must do it yourself and, therefore, doesn't employ anyone/anything else to do his farming work for him. Craig enjoys spending quality time with his two younger children and his loving wife after a hard day's work. Craig believes that technology is a distraction from the beauty of life and, therefore, doesn't own a smartphone or any other technology.

Craig has noticed that from doing business with these new, larger organisations, their demand for products has become a lot more refined. Craig used to sell his peas fresh and in smaller quantities. However, these new companies would like to purchase larger quantities of peas, freeze them and put them into storage. Due to this, the products supplied need to meet new criteria (for example, crop moisture needs to be at a specific level when harvested). Due to Craig only being human, he cannot correctly measure the exact moisture level of crops when harvested. This means that companies do not want to buy his product as they're not at the quality they would like and, thus, Craig is losing money.

Craig admits that he needs help with determining this level of requirements. Craig is reluctant to use technology to assist him but with the persistence of his supportive family, Craig agrees to employ this new technology to report to him specifically the state of his crops.

This new help will, hopefully, result in companies wanting to purchase his product again.

# Scenarios

The 3 scenarios listed below illustrate issues that might need to be considered in developing a useful and usable interactive design. Each scenario is linked with a different persona that has been created.

## Scenario #1 (*Dave*)

Dave sees that his supply of frozen peas is running low. Dave logs into the Agricultural Management System (AMS) to order some more. On logging in, Dave goes to ‘Make an Order’ from the Menu and is presented with a list of products that he can order. Dave selects ‘Peas’ from the list and nothing more. Dave then proceeds with the order to see which farmers known to AMS supply peas. Dave selects a farmer from the list that meet his requirements (i.e. price they charge/delivery time). If Dave is happy with the order details, he then selects to order the product from that farmer.

## Scenario #2 (*John*)

John sees that one of his crops (Peas) is ready for harvest and so he goes out into the field and harvests them. From checking AMS previously, John knows who the next buyer in the queue is for these crops. He packages up the crop and ships them to the buyer.

Once John has returned to his farm, he knows that he must log into AMS that he has harvested the crop. John logs into AMS with his user credentials. He proceeds to select ‘View Fields’ from the initial Menu he is presented with. From the list of fields shown to him, John selects the one that he has just harvested which takes him through to another screen where he can choose the action performed on that field. John knows that he has harvested the field so clicks the ‘Record Harvest’ button. He can then enter the yield collected (in %) and the date the harvest was taken. When this is complete, AMS now knows that there is not a current planting in that field.

## Scenario #3 (*Craig*)

Craig wants to check the health status of one of his crops. He logs into AMS and selects to ‘View Fields’ from the immediate list presented to him. From here, Craig is displayed with a list of fields to select from. Craig knows what field the crop is in that he wants to view and selected that field. From the list of actions given to him, Craig clicks the button ‘Check Crops’ as that’s what he wants to do. The next screen shown is the levels of the 5 variables that affect the health status/growth of the crop in question. The 5 variables are:

1. Soil Moisture
2. Soil Acidity
3. Soil Temperature
4. Air Temperature
5. Light

These variables are readings that have been taken by the Sensors and relayed back via the FieldStations that the Sensors are attached to. Craig sees that the value for ‘Soil Moisture’ is below a satisfactory level for that crop. He goes out into the field and manually irrigates the crop using a watering can (or something similar). When Craig sees the values again in AMS, he sees that the ‘Soil Moisture’ value for the crop in that field has increased.

# Concept Designs

Below is a list of the 3 high-level concepts designs and where to find them:

1. **Filename:** CD-Seasonal\_Crop\_Calculation

**Extension:** PNG

**Description:** This concept design shows that, if the Farmers needed help (novices), AMS could suggest crops to plant based on the time of year it is.

1. **Filename:** CD-Automated\_Harvester

**Extension:** PNG

**Description:** This concept design describes that AMS can calculate when a crop is ready for harvesting. Select the appropriate type of harvester depending on the crop and that the harvester can do the harvesting based on GPS.

1. **Filename:** CD-Automated\_Greenhouse\_Cover

**Extension:** PNG

**Description:** This concept design suggests that, if the field were in a controlled environment (i.e. a greenhouse), then AMS protect the crops against harsh elements (i.e. strong winds)

# Storyboards

Below is a list of the 3 detailed storyboards we have created for this system and where to find them:

1. **Filename:** SB-Craig\_Waters\_Crops

**Extension:** PNG

**Description:** This storyboard shows how Craig (the small-scale Farmer) interacts with AMS to see the health status of his crops and what can be done about it.

1. **Filename:** SB-John\_Records\_Harvest

**Extension:** PNG

**Description:** This storyboard shows that, when John (large-scale Farmer) has recently delivered stock to a customer, how John logs into AMS that he has completed a harvest

1. **Filename:** SB-Dave\_Makes\_Order

**Extension:** PNG

**Description:** This storyboard shows how Dave (Production Line Manager) goes about making an order to a Farmer through AMS