



# User Requirements Document (URD)

## 2IPEO SOFTWARE/WEB ENGINEERING PROJECT

November 8, 2019

### GROUP 1

L. van de Beek	1015669
K.K.J. Burgers	1004884
M. Ghanem	1036435
A.N. Groote Woortmann	1000308
M.J.B. Keizer	0988425
B. van Leeuwen	0996101
L.A. Roozen	0948743
W.J.A. van Santvoort	1010775
H.P.A. Swinkels	1005900

**TU/e**

version: 2.0

supervisor: Serguei Roubtsov

customer: Dimas Satria



### **Abstract**

This user requirements document describes the user requirements for CloudFarmer, a web-based user interface, which shows the data of sensors that are placed in the fields of farmers. This application will help users get better insights into crop and field conditions. CloudFarmer is part of the Software Engineering Project (2IPE0) at the Technical University of Eindhoven. This document complies with the ESA software standards. [1]



# Contents

<b>I</b>	<b>Document Status Sheet</b>	<b>1</b>
I.I	General . . . . .	1
I.II	Document History . . . . .	1
<b>II</b>	<b>Document Change Records</b>	<b>2</b>
<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Scope . . . . .	3
1.3	List of Definitions . . . . .	4
1.3.1	Definitions . . . . .	4
1.3.2	Abbreviations and Acronyms . . . . .	6
1.4	List of References . . . . .	6
1.5	Overview . . . . .	6
<b>2</b>	<b>General Description</b>	<b>7</b>
2.1	Product Perspective . . . . .	7
2.2	General Capabilities . . . . .	7
2.2.1	User Interface . . . . .	7
2.2.2	User Management . . . . .	9
2.2.3	Import/Export . . . . .	10
2.3	General Constraints . . . . .	10
2.3.1	User Interface . . . . .	10
2.3.2	Usability . . . . .	10
2.3.3	Security . . . . .	11
2.3.4	Performance . . . . .	11
2.3.5	Environment . . . . .	11
2.4	User Characteristics . . . . .	11
2.4.1	General User . . . . .	12
2.4.2	Researcher . . . . .	12
2.4.3	Farmer . . . . .	12
2.4.4	Farm admin . . . . .	13
2.5	Environment Description . . . . .	14
2.6	Dependencies and Assumptions . . . . .	15
2.6.1	Dependencies . . . . .	15
2.6.2	Assumptions . . . . .	16
<b>3</b>	<b>Requirements</b>	<b>17</b>
3.1	Capability Requirements . . . . .	17
3.1.1	User Interface . . . . .	17
3.1.2	User Management . . . . .	23
3.1.3	Import/Export . . . . .	28
3.2	Constraint Requirements . . . . .	30
3.2.1	Usability . . . . .	30
3.2.2	Security . . . . .	30

3.2.3	Performance . . . . .	31
3.2.4	Environment . . . . .	31
<b>A</b>	<b>Appendix A</b>	<b>34</b>
A.1	General user use cases . . . . .	34
A.1.1	Creating an account . . . . .	34
A.1.2	Logging in . . . . .	35
A.1.3	Logging out . . . . .	36
A.1.4	Searching for a (public) farm . . . . .	37
A.1.5	Viewing details of a farm . . . . .	38
A.1.6	Viewing details of a field . . . . .	39
A.1.7	Viewing details of a crop field . . . . .	40
A.1.8	See observation data of a crop field . . . . .	41
A.1.9	Viewing details about equipment . . . . .	42
A.2	Researcher use cases . . . . .	43
A.2.1	Editing observation data . . . . .	43
A.2.2	Deleting observation data of a crop field . . . . .	44
A.3	Farmer use cases . . . . .	45
A.3.1	Editing a field . . . . .	45
A.3.2	Add field . . . . .	46
A.3.3	Deleting a field . . . . .	47
A.3.4	Editing a crop field . . . . .	48
A.3.5	Deleting a crop field . . . . .	49
A.3.6	Editing equipment . . . . .	50
A.3.7	Adding equipment . . . . .	51
A.3.8	Deleting equipment . . . . .	52
A.4	Farm admin use cases . . . . .	53
A.4.1	Editing a farm . . . . .	53
A.4.2	Adding a new farm . . . . .	54
A.4.3	Deleting a farm . . . . .	55
A.4.4	Viewing user right details . . . . .	56
A.4.5	Editing user right details . . . . .	57
<b>B</b>	<b>Appendix B</b>	<b>58</b>
B.1	Signing page . . . . .	58

## I Document Status Sheet

### I.I GENERAL

<b>Document title:</b>	User Requirements Document (URD)	
<b>Document identifier:</b>	URD/2.0	
<b>Authors:</b>	L. van de Beek	1015669
	K.K.J. Burgers	1004884
	M. Ghanem	1036435
	A.N. Groote Woortmann	1000308
	M.J.B. Keizer	0988425
	B. van Leeuwen	0996101
	L.A. Roozen	0948743
	W.J.A. van Santvoort	1010775
	H.P.A. Swinkels	1005900
<b>Document status:</b>	Version 2.0	

### I.II DOCUMENT HISTORY

Version	Date	Authors	Reason
0.1	04-09-2019	Everyone	Initial version.
0.2	04-09-2019	Everyone	Implement requirements.
0.3	05-09-2019	Everyone, but luuk	Added more specific requirements.
1.0	06-09-2019	Everyone, but luuk	The First draft to be sent to the customer for inspection.
1.1	08-09-2019	Luuk	Feedback was given by the quality manager.
1.2	09-09-2019	Everyone	Added the use cases, and made changes to other sections pertaining to the writing style.
1.3	10-09-2019	Everyone	More use cases and requirements were added. Chapter 2 was written.

1.4	11-09-2019	Everyone	Finished Use Cases, revamped requirements to SMART standard, reworked chapter 2 and 3.
2.0	12-09-2019	Kas, Mo-hamed, Arthur, Bram, Han	Implemented review feedback from the second draft.

## II Document Change Records

Version	Date	Section	Reason
0.1	03-09-2019	All	Initial version.
0.2	04-09-2019	All	Initial content added.
0.3	05-09-2019	All	Added more specific requirements.
1.0	06-09-2019	All	Added list of acronyms and added Use cases appendix.
1.1	08-09-2019	All	Created changes based on the feedback from the quality manager.
1.2	09-09-2019	All	Improvements were needed.
1.3	10-09-2019	All	Improvements were needed.
1.4	11-09-2019	All	Improvements were needed.
2.0	12-09-2019	All	Improvements were needed.

# 1 Introduction

## 1.1 PURPOSE

The purpose of this document is to create an agreement between the customer and the CloudFarmers for all the requirements that the customer wants to have for the application called "CloudFarmer". These requirements are the result of several conversations between the customer and the project group.

## 1.2 SCOPE

CloudFarmers is a team of Computer Science Bachelor students working on a Software Engineering Project for the TU/e and Dimas Satria. Dimas Satria is a representative of *Praktijkcentrum Voor Precisie Landbouw*, a knowledge and practise centre founded to speed up the adoption of precision agriculture (PA) in the Netherlands.

Precision agriculture is a farming management concept based on observing, measuring and responding to inter and intra-field variability in crops. The goal of precision agriculture research is to define a decision support system (DSS) for the whole farm management with the goal of optimising returns on inputs while preserving resources.

The goal of the CloudFarmers is to develop an application that gives users of the application insights into the information of specific fields. These users might be farmers, researchers or even just curious people. This information will be accessible via an easy to use user interface that is able to display data gathered from a number of different data sources in an orderly manner. The process to deploy the application on a server should be straightforward.



## 1.3 LIST OF DEFINITIONS

### 1.3.1 Definitions

Name	Definition
Active farm	An active farm is the farm to which the map, history, live, field and farm settings view are currently referencing.
Active field	An active field is the field to which the field view is currently referencing.
Active crop field	An active crop field is the crop field to which the field view is currently referencing.
Available farm	A farm of which the user is allowed to see the information.
CloudFarm	Database server for the application.
CloudFarmer	Application as defined by the project group and the customer.
CloudFarmers	The project group tasked to create the aforementioned application.
Crop	The virtual representation of a physical crop.
Crop field	The virtual representation of a physical crop field. A field can contain multiple crop fields.
Crop field information/ the information of a crop field	All the information about a specific crop field; location, crop type, crop year, crop season, area.
Customer	The person who assigned the task to create the application to the project group. In this case Dimas Satria.
Dacom	The company that provided us with their crop database and their API for extracting information from this database.
Data sources	List of all the sources of data that are known by the application.
Equipment	The sensors in a field
Farm	The virtual representation of a physical farm. A user is allowed to have more than one virtual farm.
<i>Farmer</i> (Role)	A person who works on the physical farm.

<i>Farm admin</i> (Role)	A user with unrestricted access to all data associated with a specific farm. (E.g farm information/permissions, field information/permissions ... etc)
Farm information/ the information of a farm	All the information about a specific farm; ID number, name, address, postal code, country, email, phone number, website.
Farm permissions	A set of rules that determine which users are able to read, write or delete farm information.
Field	The virtual representation of a physical field. A farm can have multiple virtual fields.
Field information/ the information of a field	All the information about a specific field; location, field name, area, size in hectares, soil type.
Field permissions	A set of rules that determine which users are able to read, write or delete field information.
<i>General user</i> (Role)	A user that is a member of a farm without permissions.
Member	A user is a member of a farm, if they have a role assigned to them on that farm.
Physical crop field	A physical crop field is a physical field that contains at least one type of crop.
Physical farm	A physical farm that has a physical building.
Physical field	A physical field is a patch of dirt, clay, or some type of rock that is owned by a farmer, on which they are able to grow crops.
<i>Researcher</i> (Role)	A person who carries out academic or scientific research.
Selected farm	A farm that the user has selected.
User	A person that interacts with the application.
User role	A role that can be assigned to a user within a farm which defines all the field permissions and farm permissions of the user. These different roles are defined in the definitions and have the tag (role) at the end of them.
Views	The map view, history view, live view, fields view, settings view and personal settings view.
Widgets	A component of an interface that enables a user to perform a function or access a service.

### 1.3.2 Abbreviations and Acronyms

Acronym	Meaning
CSV	Comma-Seperated Values (file format)
ESA	European Space Agency
ID	Identity
JSON	JavaScript Object Notation
JWT	JSON Web Tokens
OAuth	Open authorization
PNG	Portable Network Graphics (file format)
SEP	Software Engineering Project
TU/e	Eindhoven University of Technology
URD	User Requirements Document

## 1.4 LIST OF REFERENCES

- [1] *E.B. for software Standardisation and Control*. ESA software standards, 1991.
- [2] *WolkyTolky API Description*. WolkyTolky, 2019.
- [3] *Dacom API Docs*. URL: [crop-r.com/api/v3/docs/](http://crop-r.com/api/v3/docs/).
- [4] *Data Storage for Information Discovery in Precision Agriculture*. URL: [app.swaggerhub.com/apis-docs/dimsc/DataStorage/1.0.0/](http://app.swaggerhub.com/apis-docs/dimsc/DataStorage/1.0.0/).
- [5] Steve Messenger. *DSDM Agile Project Framework Handbook*. URL: [https://www.agilebusiness.org/page/ProjectFramework\\_10\\_MoSCoWPrioritisation](https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation). (accessed: 09.04.2019).

## 1.5 OVERVIEW

Chapter 2 gives a general description of the application. It begins with the product perspective followed by the *general capabilities* of the application and the *general constraints* on the application. After this, the chapter continues with the *User Characteristics*, the *Environment Description*, and the *Assumptions and Dependencies*. Chapter 3 is more specific, it lists each user requirement with its description and assigned priority. This chapter includes both capability and constraint requirements. Appendix A provides a list of relevant use cases.

## 2 General Description

### 2.1 PRODUCT PERSPECTIVE

In recent years, sensors have become a highly adopted utility in the world of agriculture. Sensors can measure a variety of different variables of a farm field (e.g. crop yield, terrain features/topography, organic matter content, moisture levels, nitrogen levels, acidity, Electric Conductivity, Magnesium, Potassium, and others). Specialised software is needed to collect, store and combine the different measurements for (collective) use by agricultural businesses and knowledge institutions to get a better understanding of the different variables involved in the quality and quantity of crops. This ultimately makes more precise decisions, which will result in a better harvest.

Yearly, hundreds of students from the Netherlands go to third-world countries to set up such sensors in fields of local farmers to study the data. They would like to have an easy to use application to collect and read all the data from those sensors. When they return home, they leave behind the sensors for the local farmers to use, which will also need such an application to get value out of those sensors.

CloudFarmer will satisfy these needs, as it will be an information discovery application for precision agriculture. CloudFarmer will be open source and thus available to everyone around the world for free. CloudFarmer also aims to lower the knowledge gap required to read, edit, and delete the data created by such sensors through the use of an easy user interface.

### 2.2 GENERAL CAPABILITIES

CloudFarmer will have its capabilities described in this section. The roles *farm admin*, *farmer*, *researcher* or *general user* in this section refer to the relation between the actor and a specific farm. So an actor can be a *farmer* for one farm, a *general user* for another farm, and no member on another farm.

#### 2.2.1 User Interface

##### **General**

The first view that is presented to the user upon logging in is the main view of the CloudFarmer interface. This view shows the location of all accessible farms for this user on an interactive map. From each view a user can switch to another view if they have the correct permissions.

## Map view

The map view gives a general, interactive overview of available farms to the user. The user is able to select farms based on their location on the map, or by searching their name. The user can receive additional information about a certain farm in a simple view, which shows the farm's general information and a small description.

## History view

The history view of an active farm shows the user the collected farm data during a specific time period of that farm. This data is visible in easily digestible graphs, so the user can quickly see the changes that occurred. There are also multiple filter options, so that the user can quickly see different parts of the available data.

## Live view

The live view contains the most recent data from the available data sources of the active farm. This view also supplies the user with a summary of the current data, so that it may be shared with other users. The user may also customise the view with the use of widgets, so that the view can be customised to the user's desire. The widgets consist of a collection of useful data from the available data sources.

## Fields view

The fields view has a collection of all the (crop) fields on the active farm. The fields view allows the user to view and manage the different crop types and their location. The user can draw the fields that are part of their farm, after which they can draw the crop fields inside the fields and add the corresponding crops. If the user has any sensors placed on the field, they may add them as well.

## Farm settings view

The *farm admin* can use the farm settings view to get an overview of all the users that are able to see the information and fields of their farm. This view also manages the roles of the users and researchers currently subscribed to the farm. The general information of the farm can also be edited by the *farm admin* on this view. Besides that, the *farm admin* can also change the visibility of the farm to either private or public.

## Personal settings view

The personal settings view enables the user to view their personal details, such as their email address and full name. The user can also change all of their personal details, as well as delete their account.

### 2.2.2 User Management

#### Account

To access the CloudFarmer interface, a user needs to create an account. That can be done by filling in their corresponding email address, first name, last name, and a personalised password. Once an account is created the user can log-in with its email address and password.

#### Farm admin

When the user creates an account they also have the option to create a farm. To create a farm, a user has the possibility to give the farm its name, address, postal code, country, email, phone number, and website. When they select the option to create a farm, they will automatically become a farm admin of that farm. This gives them the right to assign other users access to their farm as a *farm admin* or *researcher*, via their email address. The *farm admin* also has the option to put information about the farm online as private or public for general users. A *farm admin* can add fields to its farm by providing the field name, field area, field size in hectares, and soil type of the field.

A *farm admin* can also add crop fields to its fields. This can be done by giving the location, crop type, crop season and area of the crop field. The *farm admin* has the rights to edit or delete the field and crop field information. The *farm admin* can read, edit, and delete the observed data from its data sources on that farm. The *farm admin* can also add sensors to the farm by giving the following information about the sensors: name, model, description, manufacturing date, serial number, and owner ID. The *farm admin* is able to edit and delete the data sources on that farm once they are created.

#### Farmer

A user can be given the role and rights of a *farmer* on a farm by the *farm admin*. Once a user is a *farmer*, that user also has right to read the farm information regardless if it is public or private. The *farmer* has the rights to read, edit, and delete the field, crop field, observation data, and sensor information.



## Researcher

A user can be given the role and rights of a *researcher* on a farm by the farm admin. Once a user is a *researcher*, that user has the rights to read the farm data regardless if it is public or private. The user is also allowed to read the field, crop field, observation data, and sensor information. The *researcher* is also allowed to edit and delete the observation data if the *farm admin* has given permission.

## General user

If a user has no appointed rights by a *farm admin* then that user is classified as a general user. A general user can always see a farm name and the country it is located in. Furthermore, they can see field information, crop field information, observation data, and sensor information. This depends, however, on whether the farm admin opted for the information being private or public.

### 2.2.3 Import/Export

The farm admin and the farmer can upload CSV files manually, researcher can do this as well if the farm admin has given them permission to do so. The APIs can also provide data from the data sensors and the Dacom database. All users that can see the observation data (farm admin, farmer, researcher, and general user if the farm settings are public) can export this data in CSV format.

## 2.3 GENERAL CONSTRAINTS

The constraints that CloudFarmer will take into account are listed below. These are necessary to create a clear picture of what CloudFarmer will not be capable of in terms of functionality and define its limits.

### 2.3.1 User Interface

A user can only select one active farm at the same time, in order to prevent data from different farms being mixed up by the user. If the user can only select a single farm, they will be less likely to edit the farm data incorrectly.

### 2.3.2 Usability

The main users of CloudFarmer will be real life farmers and researchers. However, the application is open to everybody interested in the data that is collected on farms worldwide. Even though these users might not necessarily possess a rigid technical

background, they are still important to the project. Consequently, the user interface of CloudFarmer should be easy and intuitive to utilise without the guide of a user manual. This way, an untrained user should be able to find the requested data without spending too much time in the application.

### 2.3.3 Security

Access to the application is granted only if the user provides correct credentials. To maintain the confidentiality of the data, SQL injection attacks have to be prevented. Therefore, special characters are filtered out before being registered as part of the credentials. Additionally, all sessions associated with CloudFarmer are to be encrypted to prevent any malicious attacks that may compromise the security of the data. An exemplary protocol to realise this goal is HTTPS, for instance. To increase the security of CloudFarmer further it should also be protected against Cross-Site Scripting.

### 2.3.4 Performance

To provide the user with a good user experience, CloudFarmer should perform well. Long waiting times for responses from the server, after a request by the user, will cause frustration to the user. To guarantee good performance, there are requirements set for the maximum waiting time for certain actions. Since not all responses will be possible to conclude within a few seconds, the user will be notified if the response by the server will take longer.

### 2.3.5 Environment

CloudFarmer is a web application that is developed to function primarily on Chrome 72 and later versions. The application may also support other browsers, such as: Firefox, Edge, Safari, and Internet Explorer above version specified. Furthermore, the application might be accessible from mobile devices and tablets, to increase the mobility of the application. CloudFarmer supports the English language as default, but also has the option to add more languages.

## 2.4 USER CHARACTERISTICS

A role describes the permissions a specific user has in relation to a certain farmer in the application. The role of a user can be different for every farm. The four main roles together with their corresponding permissions are listed in sections 2.4.1 - 2.4.4. A summary of these roles and their permissions is illustrated in Figure 1.

### 2.4.1 General User

A *general user* has the following permissions:

- A *general user* can view the farm's name.
- A *general user* can view the country the farm is located.
- A *general user* can view the farm's information.\*
- A *general user* can view the information of specific fields belonging to the farm.\*
- A *general user* can view the observed data from the farm.\*
- A *general user* can view which data sources the farm has.\*

\*Permission can be granted or revoked by an admin of that farm.

### 2.4.2 Researcher

A *researcher* has the following permissions:

- A *researcher* can view the farm's information.
- A *researcher* can view the information of specific fields belonging to the farm.
- A *researcher* can view the observed data from the farm.
- A *researcher* can edit the observed data from the farm.\*
- A *researcher* can delete the observed data from the farm.\*
- A *researcher* can view which data sources the farm has.

\*Permission can be granted or revoked by an admin of that farm.

### 2.4.3 Farmer

A *farmer* has the following permissions:

- A *farmer* can view the farm's information.
- A *farmer* can view the information of specific fields belonging to the farm.
- A *farmer* can edit the information of specific fields belonging to the farm.
- A *farmer* can delete fields from the farm.
- A *farmer* can view the observed data of that farm.
- A *farmer* can edit the observed data of that farm.

- A *farmer* can delete the observed data of that farm.
- A *farmer* can add data sources to the farm.
- A *farmer* can view which data sources the farm has.
- A *farmer* can edit the data sources of the farm.
- A *farmer* can delete data sources from the farm.

#### 2.4.4 Farm admin

A *farm admin* has the following permissions:

- A *farm admin* can view their farm's information.
- A *farm admin* can edit their farm's information.
- A *farm admin* can delete their farm.
- A *farm admin* can add a field to their farm.
- A *farm admin* can view the information of fields belonging to their farm.
- A *farm admin* can edit the information of fields belonging to their farm.
- A *farm admin* can delete their fields from their farm.
- A *farm admin* can view the observed data of their farm.
- A *farm admin* can edit the observed data of their farm.
- A *farm admin* can delete the observed data of their farm.
- A *farm admin* can add data sources to their farm.
- A *farm admin* can view which data sources their farm has.
- A *farm admin* can edit data sources their farm has.
- A *farm admin* can delete data sources from their farm.
- A *farm admin* can view the user permissions of their farm.
- A *farm admin* can edit the user permissions of their farm.
- A *farm admin* can revoke access of a user from their farm.

Role	Farm			Field/CropField			Observation Data			Equipment			Farm User Access		
	Read	Edit	Delete	Read	Edit	Delete	Read	Edit	Delete	Read	Edit	Delete	Read	Edit	Delete
Farm admin	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green
Farmer	Green	Red	Red	Green	Green	Green	Green	Green	Green	Green	Green	Green	Red	Red	Red
Researcher	Green	Red	Red	Green	Red	Red	Green	Yellow	Yellow	Green	Red	Red	Red	Red	Red
General user	Yellow	Red	Red	Yellow	Red	Red	Yellow	Red	Red	Yellow	Red	Red	Red	Red	Red

Green	allowed
Yellow	depending on the permission value set by farm admin
Red	not allowed

Figure 1: An overview of the permissions each role will have.

## 2.5 ENVIRONMENT DESCRIPTION

In figure 2, CloudFarmer's environment is illustrated. The area highlighted in pink represents the scope of CloudFarmer, whereas the area shown in blue represents our client's scope, whose development will run in parallel with the development of CloudFarmer. The remainder is part of the already existing environment(s). That is, CloudFarmer can access data from sensors using the services API provided by the client and from other servers using the cloud API provided by Dacom company. A user will interact with CloudFarmer via their browser.

The front-end part of CloudFarmer is responsible for the proper visualisation of the data. It also handles the user interaction accordingly by sending the appropriate requests to the back-end and provides the user the means to communicate with the APIs as well. The front-end will make use of ReactJS and ExpressJS.

On the other hand, the back-end of CloudFarmer will rely on the APIs that the client uses. These APIs have been provided to be used by CloudFarmer. The back-end also provides the ability to import or export CSV files directly. It allows the farmer to gather the data from the sensors directly and upload them to the database with the use of an external drive. The export functionality of CSV files may be of useful for researchers when analysing multiple data-sets simultaneously. The back-end is connected to the database of Cloudfarmers in order to receive the correct user information and to also modify data in the database according to the user's interaction.

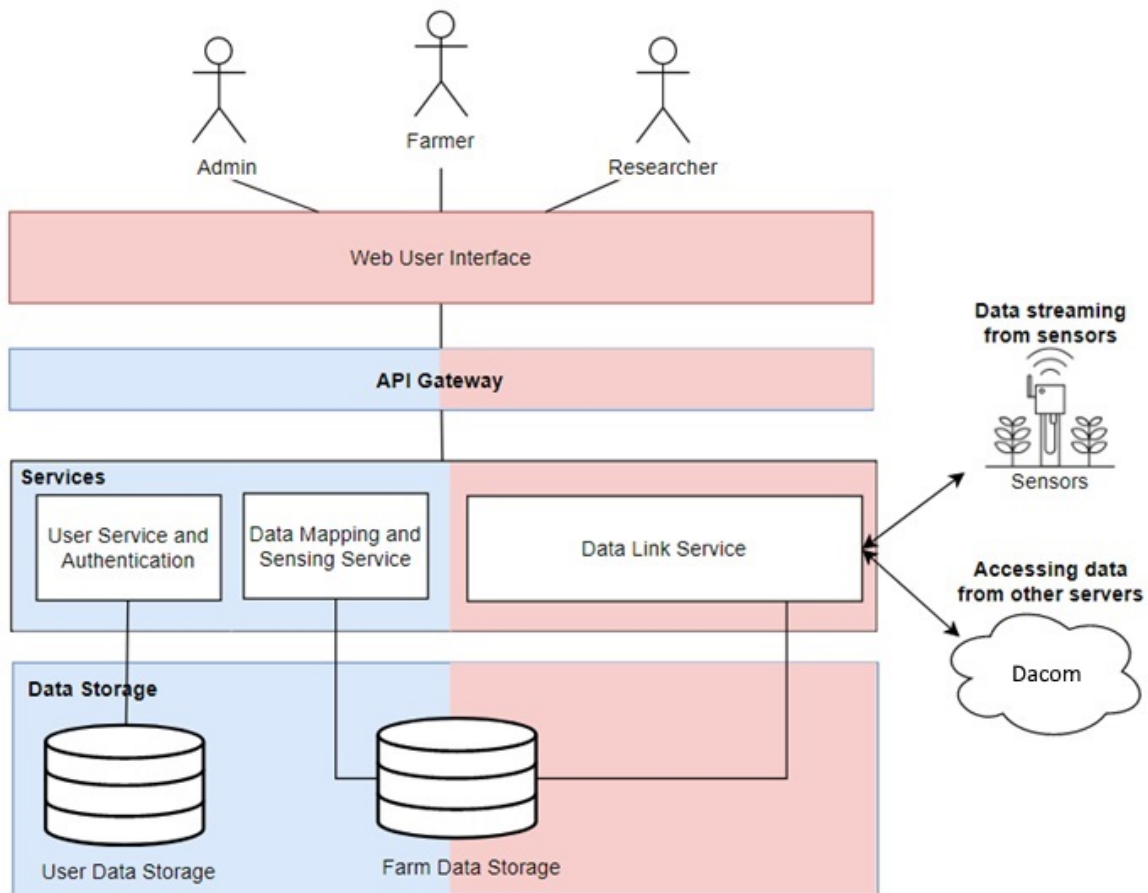


Figure 2: CloudFarmer's Environment

## 2.6 DEPENDENCIES AND ASSUMPTIONS

CloudFarmer is dependent on third party services as well as services provided by the client. These services are listed in sections 2.6.1 and 2.6.2. It is assumed that these services will work as expected during the finalisation of CloudFarmer.

### 2.6.1 Dependencies

CloudFarmer depends on the following items in order to function according to the specifications:

- WolkyTolky provides the WolkyTolky API. [2]
- Dacom company provides the Dacom Farm Cloud API. [3]



- The client provides the services API specified on SwaggerHub. [4]
- TU/e provides a server to host CloudFarmer.

### 2.6.2 Assumptions

The following assumptions must hold for CloudFarmer to function according to the specifications:

- The APIs provided are stable and implemented in a proper fashion and adhere to the specifications in their respective documents.
- The server provided by TU/e is online and accessible all the time.

## 3 Requirements

The requirements for the application are listed in the following sections. They indicate the requisites of the customer and the capabilities of the end product. The requirements are separated into *capability requirements (functional)*, denoted by FR, and *constraint requirements*, denoted by CR. Both of these are prioritised according to the MoSCoW model as shown below.[5]

Acronym	Meaning
Must have	Requirements that will be implemented.
Should have	Requirements that should ideally be implemented
Could have	Requirements that will be implemented if there is still time left after implementing the above-mentioned requirements.
Won't have	Requirements that will not be implemented but could be implemented in a later project.

### 3.1 CAPABILITY REQUIREMENTS

This section describes the requirements that add to the functionality of the application; restrictions on the application are mentioned in the constraints requirements.

#### 3.1.1 User Interface

In this section, the requirements are described that apply to the general user interface of the application. Keep in mind that the application might offer a functionality to the general user, which, based on their permissions, might not be accessible. Whether the user has access to the functionality is mentioned under the section 3.1.2. The paragraph "General" gives requirements that specify which views are available. The ensuing paragraphs give the specific requirements for these views.

##### General

FR1	<i>Must have</i>
A user is provided with a personal settings view.	
FR2	<i>Should have</i>
A user is provided with a map view.	

<b>FR3</b>	<i>Must have</i>
For every farm a user is a member of, the user is provided with a history view.	
<b>FR4</b>	<i>Must have</i>
For every farm a user is a member of, the user is provided with a live view.	
<b>FR5</b>	<i>Must have</i>
For every farm a user is a member of, the user is provided with a field view.	
<b>FR6</b>	<i>Must have</i>
For each farm that a user has the role <i>farm admin</i> on, the user is provided with a farm settings view.	
<b>FR7</b>	<i>Must have</i>
A user that is a member of a farm, is able to select it as the active farm.	
<b>FR8</b>	<i>Must have</i>
A user is able to switch to their personal settings view.	
<b>FR9</b>	<i>Must have</i>
A user that is a member of a farm, is able to switch to the map view of that farm.	
<b>FR10</b>	<i>Must have</i>
A user that is a member of a farm, is able to switch to the history view of that farm, when it is the active farm.	
<b>FR11</b>	<i>Must have</i>
A user that is a member of a farm, is able to switch to the live view of that farm, when it is the active farm.	
<b>FR12</b>	<i>Must have</i>
A user that is a member of a farm, is able to switch to the field view of that farm, when it is the active farm.	
<b>FR13</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on the active farm is able to switch to the farm settings view of that active farm.	
<b>FR14</b>	<i>Should have</i>
A user is informed when no connection can be established to a data source.	
<b>FR15</b>	<i>Must have</i>
A user is informed when a view is loading.	

---

<b>FR16</b>	<i>Should have</i>
A user is informed when a view is updating.	

### **Map view**

---

<b>FR17</b>	<i>Should have</i>
A user is able to become a member of a farm that is selected on the map view.	

---

<b>FR18</b>	<i>Should have</i>
A user is able to remove their membership from a farm on the map view.	

---

<b>FR19</b>	<i>Should have</i>
A user is able to create a new farm on the map view.	

---

<b>FR20</b>	<i>Should have</i>
A list of available farms can be accessed by the user on the map view.	

---

<b>FR21</b>	<i>Should have</i>
A user is able to select any of the available farms in the list of available farms on the map view.	

---

<b>FR22</b>	<i>Should have</i>
The map view provides the user with the farm information of a selected farm.	

---

<b>FR23</b>	<i>Should have</i>
A user is able to access a search function to find farms based on a farm's name on the map view.	

### **History view**

---

<b>FR24</b>	<i>Should have</i>
From the history view, a user is able to view each distinct observation data set of the active farm that CloudFarm and Dacom's database offer.	

---

<b>FR25</b>	<i>Should have</i>
A user is able to filter the visible data based on the data sets on the history view.	

---

<b>FR26</b>	<i>Should have</i>
A user is able to filter the visible data based on the data source on the history view.	

---

<b>FR27</b>	<i>Should have</i>
A user is able to filter the visible data based on the field on the history view.	

---

FR28	<i>Should have</i>
A user is able to filter the visible data based on the crop field on the history view.	

---

FR29	<i>Should have</i>
From the history view a user is able to view each distinct data set that the database offers of the active farm, in the form of at least one graph.	

---

FR30	<i>Could have</i>
A user can customise how data of the active farm is displayed using widgets.	

---

### **Live view**

---

FR31	<i>Must have</i>
A user should be provided with an overview of the current data sources, present on the active farm.	

---

FR32	<i>Must have</i>
A user should be provided with an overview of the current data from the active farm.	

---

FR33	<i>Should have</i>
A user is able to filter the visible data based on the data sets on the live view.	

---

FR34	<i>Should have</i>
A user is able to filter the visible data based on the data source on the live view.	

---

FR35	<i>Should have</i>
A user is able to filter the visible data based on field on the live view.	

---

FR36	<i>Should have</i>
A user is able to filter the visible data based on crop field on the live view.	

---

FR37	<i>Could have</i>
A user has the ability to display the current data from the active farm using widgets.	

---

### **Field view**

---

FR38	<i>Must have</i>
A user is able to add a field to the active farm on the fields view.	

---

FR39	<i>Must have</i>
A user is able to select a field of the active farm to be the active field.	

---

<b>FR40</b>	<i>Must have</i>
A user is able to see the field information of the active field on the fields view.	
<b>FR41</b>	<i>Should have</i>
A user is able to change the field information of the active field on the fields view.	
<b>FR42</b>	<i>Must have</i>
A user is able to delete a field from the active farm on the fields view.	
<b>FR43</b>	<i>Must have</i>
A user is able to add a crop field to the active field on the field view.	
<b>FR44</b>	<i>Must have</i>
A user is able to select a crop field of the active field to be the active crop field.	
<b>FR45</b>	<i>Must have</i>
A user is able to see the crop field information of the active crop field on the fields view.	
<b>FR46</b>	<i>Should have</i>
A user is able to change the crop field information of the active crop field on the fields view.	
<b>FR47</b>	<i>Must have</i>
A user is able to delete a crop field from the active field on the fields view.	
<b>FR48</b>	<i>Must have</i>
A user is able to add a data source to the active farm.	
<b>FR49</b>	<i>Should have</i>
A user is able to edit the data sources on the active farm.	
<b>FR50</b>	<i>Must have</i>
A user is able to delete a data source of the active farm.	
<b>FR51</b>	<i>Must have</i>
A user is able to view the data sources of the active farm.	

### **Farm settings view**

<b>FR52</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is able to see which users can see the information of that active farm in the farm settings view.	



<b>FR53</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to see which users can see the information from which fields of that active farm in the farm settings view.	
<b>FR54</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to change the role a certain user has of that active farm in the farm settings view.	
<b>FR55</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to change the information of that active farm in the farm settings view.	
<b>FR56</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to change whether every user can see the farm information of that active farm in the farm settings view.	
<b>FR57</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to change whether no user can see the farm information of that active farm in the farm settings view.	
<b>FR58</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to change whether <i>general users</i> can read the farm information of that active farm in the farm settings view.	
<b>FR59</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to change whether <i>general users</i> can read the field information of that active farm in the farm settings view.	
<b>FR60</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to change whether <i>general users</i> can read the observed data information of that active farm in the farm settings view.	
<b>FR61</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to change whether <i>general users</i> can read the data sources information of that active farm in the farm settings view.	
<b>FR62</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to change whether <i>researchers</i> can edit observed data of that active farm in the farm settings view.	

<b>FR63</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is able to change whether <i>researchers</i> can delete observed data of that active farm in the farm settings view.	

### Personal settings view

<b>FR64</b>	<i>Should have</i>
A user is able to read their email address on the user settings view.	
<b>FR65</b>	<i>Should have</i>
A user is able to edit their email address on the user settings view.	
<b>FR66</b>	<i>Should have</i>
A user is able to edit their password on the user settings view.	
<b>FR67</b>	<i>Should have</i>
A user is able to read their first name on the user settings view.	
<b>FR68</b>	<i>Should have</i>
A user is able to edit their first name on the user settings view.	
<b>FR69</b>	<i>Should have</i>
A user is able to read their last name on the user settings view.	
<b>FR70</b>	<i>Should have</i>
A user is able to edit their last name on the user settings view.	
<b>FR71</b>	<i>Should have</i>
A user is able to delete their account on the user settings view.	

## 3.1.2 User Management

### Accounts

<b>FR72</b>	<i>Must have</i>
A user is able to register for an account.	
<b>FR73</b>	<i>Must have</i>
A user is able to log in to the application.	
<b>FR74</b>	<i>Must have</i>
A user, that is logged in, is able to log out of their account.	

<b>FR75</b>	<i>Must have</i>
A user account must have an email address.	
<b>FR76</b>	<i>Must have</i>
A user account must have a password.	
<b>FR77</b>	<i>Must have</i>
An email address must be unique.	
<b>FR78</b>	<i>Should have</i>
A user is able to reset their password if they cannot log in.	
<b>FR79</b>	<i>Must have</i>
A user is able to have different user roles in different farms.	

### **Farm admin**

<b>FR80</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to view the information of that farm.	
<b>FR81</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to edit the information of that farm.	
<b>FR82</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to delete the information of that farm.	
<b>FR83</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to view the information of fields belonging to that farm.	
<b>FR84</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to edit the information of fields belonging to that farm.	
<b>FR85</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to delete the information of fields belonging to that farm.	
<b>FR86</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to view the information of all crop fields belonging to a field, which belongs to that farm.	

<b>FR87</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to edit the information of all crop fields belonging to a field, which belongs to that farm.	
<b>FR88</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to delete the information of all crop fields belonging to a field, which belongs to that farm.	
<b>FR89</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to view the observed data from the data sources of that farm.	
<b>FR90</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to edit the observed data from the data sources of that farm.	
<b>FR91</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to delete the observed data from the data sources of that farm.	
<b>FR92</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to view the data sources of that farm.	
<b>FR93</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to edit the data sources of that farm.	
<b>FR94</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to delete the data sources of that farm.	
<b>FR95</b>	<i>Could have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to see all the permissions that other users have on the same farm.	
<b>FR96</b>	<i>Should have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to change the role of other users on that farm to, either <i>farm admin</i> , <i>farmer</i> , <i>researcher</i> or <i>general user</i> .	
<b>FR97</b>	<i>Must have</i>
A user that has the role <i>farm admin</i> on a farm is allowed to add new users to that farm.	

---

**FR98** *Should have*  
A user that has the role *farm admin* on a farm is allowed to make the information of that farm public.

---

**FR99** *Should have*  
A user that has the role *farm admin* on a farm is allowed to make the information of that farm private.

---

**FR100** *Should have*  
A user that has the role *farm admin* on a farm is allowed to give permission to all users with the role *researcher* on that farm, to edit the information of the observation data on that farm.

---

**FR101** *Should have*  
A user that has the role *farm admin* on a farm is allowed to give permission to all users with the role *researcher* on that farm, to delete the information of the observation data on that farm.

---

**FR102** *Should have*  
A user that has the role *farm admin* on a farm is allowed to give permission to all users with the role *general user* on that farm, to view the information of that farm.

---

**FR103** *Should have*  
A user that has the role *farm admin* on a farm is allowed to give permission to all users with the role *general user* on that farm, to view the information of a specific field belonging to that farm.

---

**FR104** *Should have*  
A user that has the role *farm admin* on a farm is allowed to give permission to all users with the role *general user* on that farm, to view the observed data by that farm.

---

**FR105** *Should have*  
A user that has the role *farm admin* on a farm is allowed to give permission to all users with the role *general user* on that farm, to view the data sources of that farm.

---

**FR106** *Should have*  
A user that has the role *farm admin* on a farm is allowed to remove a user from their farm.

---

## **Farmer**

---

**FR107** *Must have*  
A user that has the role *farmer* on a farm is allowed to view the information of that farm.

---

---

**FR108** *Must have*

A user that has the role *farmer* on a farm is allowed to view the information of fields belonging to that farm.

---

**FR109** *Must have*

A user that has the role *farmer* on a farm is allowed to edit the information of fields belonging to that farm.

---

**FR110** *Must have*

A user that has the role *farmer* on a farm is allowed to delete the information of fields belonging to that farm.

---

**FR111** *Should have*

A user that has the role *farmer* on a farm is allowed to view the values of the observed data, that is produced by all the data sources of that farm.

---

**FR112** *Should have*

A user that has the role *farmer* on a farm is allowed to edit the values of the observed data, that is produced by all the data sources of that farm.

---

**FR113** *Should have*

A user that has the role *farmer* on a farm is allowed to delete the values of the observed data, that is produced by all the data sources of that farm.

---

**FR114** *Must have*

A user that has the role *farmer* on a farm is allowed to view the data sources of that farm.

---

**FR115** *Must have*

A user that has the role *farmer* on a farm is allowed to add a data source to that farm.

---

**FR116** *Must have*

A user that has the role *farmer* on a farm is allowed to delete data sources from that farm.

---

## **Researcher**

---

**FR117** *Must have*

A user that has the role *researcher* on a farm is allowed to view the information of that farm.

---

**FR118** *Must have*

A user that has the role *researcher* on a farm is allowed to view the information of all fields belonging to that farm.

---



---

**FR119** *Should have*

A user that has the role *researcher* on a farm is allowed to view observed data from the data sources of that farm.

---

**FR120** *Should have*

A user that has the role *researcher* on a farm is allowed to edit observed data from the data sources of that farm, if a *farm admin* of that farm allows this.

---

**FR121** *Should have*

A user that has the role *researcher* on a farm is allowed to delete observed data from the data sources of that farm, if a *farm admin* of that farm allows this.

---

**FR122** *Must have*

A user that has the role *researcher* on a farm is allowed to view the data sources of that farm.

---

### **General user**

---

**FR123** *Must have*

A user that has the role *general user* on a farm is allowed to view the information of that farm, if a *farm admin* of that farm allows this.

---

**FR124** *Must have*

A user that has the role *general user* on a farm is allowed to view the information of all fields belonging to that farm, if a *farm admin* of that farm allows this.

---

**FR125** *Should have*

A user that has the role *general user* on a farm is allowed to view the observed data from the data sources of that farm, if a *farm admin* of that farm allows this.

---

**FR126** *Must have*

A user that has the role *general user* on a farm is allowed to view the data sources of that farm, if a *farm admin* of that farm allows this.

---

**FR127** *Could have*

A user that has the role *general user* on a farm is allowed to request to become a researcher or farmer at that farm.

---

### **3.1.3 Import/Export**

**FR128** *Could have*

A user that is allowed to view and edit the information of a farm is able to export this information in PNG format.

<b>FR129</b>	<i>Could have</i>
A user that is allowed to view and edit the information of a farm is able to export this information in CSV format.	
<b>FR130</b>	<i>Could have</i>
A user that is allowed to edit the information of a farm can import information to this farm in CSV format.	
<b>FR131</b>	<i>Could have</i>
A user that is allowed to view and edit the information is able to export this information in PNG format.	
<b>FR132</b>	<i>Could have</i>
A user that is allowed to view and edit the information of a field is able to export this information in CSV format.	
<b>FR133</b>	<i>Could have</i>
A user that is allowed to edit the information of a field is able to import information to this field in CSV format.	
<b>FR134</b>	<i>Could have</i>
A user that is allowed to view and edit the observation data of a farm is able to export this data in PNG format.	
<b>FR135</b>	<i>Should have</i>
A user that is allowed to view and edit the observation data of a farm is able to export this data in CSV format.	
<b>FR136</b>	<i>Should have</i>
A user that is allowed to edit the observation data of a farm is able to import data to this farm in CSV format.	
<b>FR137</b>	<i>Could have</i>
A user that is allowed to view and edit the data sources of a farm is able to export a list of these sources in PNG format.	
<b>FR138</b>	<i>Could have</i>
A user that is allowed to view and edit the data sources of a farm is able to export a list of these sources in CSV format.	

---

FR139	Could have
-------	------------

A user that is allowed to edit the data sources of a farm is able to import a list of these sources to this farm in CSV format.

## 3.2 CONSTRAINT REQUIREMENTS

The requirements that constrain the possible functionality are listed below.

### 3.2.1 Usability

CR1	Must have
-----	-----------

A user with the role *farm admin* on a farm is not able to change their role.

---

CR2	Must have
-----	-----------

A user is able to perform an implemented use case without instructions within 5 minutes.

---

CR3	Should have
-----	-------------

A user is able to perform an implemented use case without instructions within 2 minutes.

---

CR4	Could have
-----	------------

A user is able to perform an implemented use case without instructions within 1 minute.

### 3.2.2 Security

CR5	Should have
-----	-------------

All connections between the application and the server must be through HTTPS.

---

CR6	Must have
-----	-----------

Any string containing input from a user must be proof against SQL injection attacks.

---

CR7	Should have
-----	-------------

Any string containing input from a user must be proof against Cross-Site Scripting attacks.

---

CR8	Should have
-----	-------------

The latitude of the location of a data source is only visible up to two digits after the point, for *general user*

---

CR9	Should have
-----	-------------

The longitude of the location of a data source is only visible up to two digits after the point, for *general user*

<b>CR10</b>	<i>Must have</i>
User authorisation and authentication uses OAuth using JWT.	

### 3.2.3 Performance

<b>CR11</b>	<i>Must have</i>
Processing login credentials is performed within 5 seconds.	

<b>CR12</b>	<i>Should have</i>
Processing login credentials is performed within 2 seconds.	

<b>CR13</b>	<i>Could have</i>
Processing login credentials is performed within 1 second.	

<b>CR14</b>	<i>Should have</i>
Generating a PNG file is performed within 20 seconds.	

<b>CR15</b>	<i>Should have</i>
Generating a PNG file is performed within 10 seconds.	

<b>CR16</b>	<i>Could have</i>
Generating a PNG file is performed within 5 seconds.	

<b>CR17</b>	<i>Must have</i>
Generating a CSV file is performed within 30 seconds.	

<b>CR18</b>	<i>Should have</i>
Generating a CSV file is performed within 20 seconds.	

<b>CR19</b>	<i>Could have</i>
Generating a CSV file is performed within 5 seconds.	

<b>CR20</b>	<i>Must have</i>
Processing a request for a new account is performed within 20 seconds.	

<b>CR21</b>	<i>Should have</i>
Processing a request for a new account is performed within 10 seconds.	

<b>CR22</b>	<i>Could have</i>
Processing a request for a new account is performed within 5 seconds.	

### 3.2.4 Environment

#### General

<b>CR23</b>	<i>Must have</i>
CloudFarmer is accessible through a web browser.	

<b>CR24</b>	<i>Must have</i>
CloudFarmer is compatible with PostgreSQL database.	

<b>CR25</b>	<i>Should have</i>
If there is no connection to the farm data storage, live data will be sent to the interface only.	

## **Responsiveness**

<b>CR26</b>	<i>Must have</i>
CloudFarmer can run on Microsoft Windows.	

<b>CR27</b>	<i>Should have</i>
CloudFarmer can run on Mac OS.	

<b>CR28</b>	<i>Should have</i>
CloudFarmer can run on Linux.	

<b>CR29</b>	<i>Could have</i>
CloudFarmer can run on Android.	

<b>CR30</b>	<i>Could have</i>
CloudFarmer can run on iOS.	

## **Browser versions**

<b>CR31</b>	<i>Must have</i>
CloudFarmer supports Chrome version 72 and later.	

<b>CR32</b>	<i>Should have</i>
CloudFarmer supports Firefox version 64 and later.	

<b>CR33</b>	<i>Should have</i>
CloudFarmer supports Edge version 44 and later.	

<b>CR34</b>	<i>Could have</i>
CloudFarmer supports Safari version 12 and later.	

<b>CR35</b>	<i>Could have</i>
CloudFarmer supports Internet Explorer version 11 and later.	

---

<b>CR36</b>	<i>Could have</i>
CloudFarmer supports Internet Explorer version 10 and earlier.	

## **APIs**

---

<b>CR37</b>	<i>Must have</i>
CloudFarmer supports the API documented on SwaggerHub.	

---

<b>CR38</b>	<i>Should have</i>
CloudFarmer supports the Dacom API.	

---

<b>CR39</b>	<i>Must have</i>
CloudFarmer supports the WolkyTolky API.	

---

<b>CR40</b>	<i>Could have</i>
CloudFarmer supports the RainDancer API.	

## **Language**

---

<b>CR41</b>	<i>Must have</i>
CloudFarmer is available in English.	

---

<b>CR42</b>	<i>Should have</i>
CloudFarmer will allow for language localisation.	

---

<b>CR43</b>	<i>Could have</i>
CloudFarmer is available in Dutch.	

## A Appendix A

### A.1 GENERAL USER USE CASES

This section describes the use cases that can be performed by all users. However, the preconditions should still be met.

#### A.1.1 Creating an account

**Goal:** Create a user account with general user account permissions for every farm.

**Actor:** A user.

**Precondition:** There does not already exist an user account with the inserted user credentials.

**Postcondition:** The actor has credentials that give them access to public data on the server.

**Summary:** The user navigates to the create account view and fills in the credentials they want to use. The server stores this data if there does not yet exist an account with these credentials.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to create account view. 2: Actor submits their account credentials.	3: System checks if the credentials are not yet registered for an account. 4.1: If not: system stores the account information in the database. 4.2: If the credentials are already in use: the system gives an error message.

### A.1.2 Logging in

**Goal:** Being logged in to the web interface.

**Actor:** A user.

**Precondition:** The actor already has an existing account.

**Postcondition:** The actor is logged in and at the home page.

**Summary:** An actor with an account logs in on the website.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor enters their credentials. 2: Actor submits the log-in information.	3: System verifies the log-in information. 4.1: Credentials are correct: System redirects the user to the home view. 4.2: Credentials are incorrect: System shows an error message to the actor.



### A.1.3 Logging out

**Goal:** Log out from the website.

**Actor:** A user.

**Precondition:** The actor is logged in.

**Postcondition:** The actor is logged out.

**Summary:** The actor logs out from the website.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor presses the logout button.	2: System shows a popup asking for confirmation whether or not the actor wants to sign out.
3: Actor presses on confirm to log out.	4.1: Confirm: System redirects to the login view. 4.2: Cancel: System hides pop-up.

#### A.1.4 Searching for a (public) farm

**Goal:** Find a specific farm the actor is looking for.

**Actor:** A user.

**Precondition:** The actor is logged in.

The farm the actor is looking for is registered in CloudFarm.

The user account has access to the farm either through permission of the farm admin, or the farm is public.

**Postcondition:** The actor sees the name and/or location of the farm on their screen. They are able to select the farm.

**Summary:** The actor searches for a farm and the system returns farms that the actor could mean.

**Priority:** Should have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor goes to the home view. 2: Actor submits the name of the farm they want to find.	3: System searches database for farm names that (partly) match with the submitted search term. 4.1: If there is at least one match: system returns a list of matches. 4.2: If there are no matches: system returns that there were no farms found.

### A.1.5 Viewing details of a farm

**Goal:** Get an overview of the details of a farm.

**Actor:** A user.

**Precondition:** The actor is logged in.

The logged in account has the permission of a farmer, farm admin, or researcher at the farm they want to select. Or the permission value of the farm details is set to public by its farm admin.

**Postcondition:** The user is presented with a overview of the details of a farm.

**Summary:** Select a farm and view the overall details of a farm.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor selects the farm from the available farms.	2: System provides a view with detailed information about the selected farm.

### A.1.6 Viewing details of a field

**Goal:** Get an overview of the details of a field.

**Actor:** A user.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farmer, farm admin, or researcher at the farm selected and is allowed to view details of the field. Or the permission value of the farm details is set to public by its farm admin.

The actor has selected a farm they wish to view details of a field from.

The farm that has been selected has one or more fields.

**Postcondition:** The actor gets an overview of the details of a field.

**Summary:** Select a field and view the overall details of a field.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor selects the field that they want to view the details of.	2: System shows a page with detailed information about the selected field of the farm.

### A.1.7 Viewing details of a crop field

**Goal:** Get an overview of the details of a crop field.

**Actor:** A user.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farmer, farm admin, or researcher at the farm selected and is allowed to view details of the field. Or the permission value of the farm details is set to public by its farm admin.

The actor has selected a field that has one or more crop fields.

**Postcondition:** The actor gets an overview of the details of a crop field from the selected field of the selected farm.

**Summary:** Select a crop field and view the overall details of a crop field.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor selects the crop field they want to view more details of.	2: System shows a view with detailed information about the selected crop field.

### A.1.8 See observation data of a crop field

**Goal:** Get an overview of the observation data of a crop field.

**Actor:** A user.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farmer, farm admin, or researcher at the farm selected and is allowed to view details of the field. Or the permission value of the farm details is set to public by its farm admin.

The actor has selected a crop field.

**Postcondition:** The actor gets an overview of the observation data from the sensors of a crop field.

**Summary:** Go to the observation data menu to see the observation data.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to the observation data view.	2.1: If the crop field has observation data: the system shows a view in which the observation data is displayed in an organised matter. 2.2: If the crop field does not have observation data: the system shows that this crop field does not have observation data.

### A.1.9 Viewing details about equipment

**Goal:** Get a view of the detailed information of the equipment used to collect the data.

**Actor:** A user.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farmer, farm admin, or researcher at the farm selected and is allowed to view details of the farm. Or the permission value of the farm details is set to public by its farm admin.

The actor has selected a farm they wish to view the equipment information from.

**Postcondition:** The actor gets a view of the detailed information about a piece of specific equipment.

**Summary:** The actor goes to a farm page, field page, or crop field page of a farm and selects a piece of equipment. The system returns a view with the details of that piece of equipment.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to a farm view, field view, or crop field view of a farm.	
	2: System shows a view with detailed information about the selected farm, field, or crop field.
3: Actor selects the piece of equipment they want more information on.	
	4: System shows a view with detailed information about the selected piece of equipment.

## A.2 RESEARCHER USE CASES

These use cases can be performed by all users that have at least the role of a researcher in relation to the selected farm. Hence all users that are either registered as researcher, farmer, or farm admin for that farm can perform these actions.

### A.2.1 Editing observation data

**Goal:** Changing the existing observation data.

**Actor:** Farm admin, farmer, or researcher.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farmer, farm admin, or researcher at the farm selected and is allowed to view details of the observation data.

The actor is on the observation data view.

**Postcondition:** The actor has edited the observation data.

**Summary:** View the observation data and edit it.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to the edit observation data view.	
3.1: Actor saves changes made.	2: System shows a view on which the observation data can be edited.
3.2: Actor cancels changes made.	4.1: System updates observation data and shows the view with the observation data. 4.2: System shows a view on which the observation data can be edited.



### A.2.2 Deleting observation data of a crop field

**Goal:** Delete already existing observation data.

**Actor:** Farm admin, farmer, or researcher.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farmer, farm admin, or researcher at the farm selected and is allowed to edit the observation data.

The actor is on the observation data view.

**Postcondition:** The actor has deleted previously existing observation data.

**Summary:** View the observation data and delete it.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor presses the edit button.	
	2: System shows a view on which the observation data can be edited.
3: Actor presses the delete button.	
	4: System asks whether actor confirms to delete the data.
4.1: Actor confirms the deletion.	
4.2: Actor cancels the deletion.	
	5.1: System deletes observation data and redirects actor to the previous page it was on.
	5.2: System shows page on which the observation data can be edited.

## A.3 FARMER USE CASES

The use cases in this section can only be performed if the role of the user in relation to the selected farm is either a farmer or a farm admin.

### A.3.1 Editing a field

**Goal:** Edit the existing field information.

**Actor:** Farm admin or farmer.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin or farmer at the farm selected.

A field has been selected which the actor wants to change.

**Postcondition:** The actor has edited information of the field.

**Summary:** Select a field and edit its corresponding information.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to the edit page.	
	2: System shows view on which the field can be edited.
3: Actor makes changes to the field.	
4.1: Actor saves changes made.	
4.2: Actor cancels changes made.	
	5.1: System updates information and shows information view of the field.
	5.2: System shows view on which the field can be edited.

### A.3.2 Add field

**Goal:** Add a field to a farm.

**Actor:** Farm admin or farmer.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin or farmer at the farm selected.

A farm has been selected to which the actor wants to add a field.

**Postcondition:** The actor has added a field.

**Summary:** Select a farm and create a new field with its specifications.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: The actor goes to the add field view.	
	2: System asks the actor for the specifications needed for a field.
3: The actor fills in the specifications.	
4.1: The actor saves the specifications.	
4.2: The actor cancels the creation of a new field.	
	5.1: The system saves the new field to the database and shows the field view of that field.
	5.2: The system shows the information view of the farm.

### A.3.3 Deleting a field

**Goal:** Delete a existing field from a farm.

**Actor:** Farm admin or farmer.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin or farmer at the farm selected.

A farm has been selected from which the actor wants to delete a field.

**Postcondition:** The field is no longer in the database of the farm.

**Summary:** Select a field and delete it.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: The actor selects they want to edit the field.	
	2: The system shows the edit view of that field.
3: The actor selects they want to delete the field.	
	4: The system asks the actor to verify that they want to delete the field.
5.1: The actor confirms that they want to delete the field.	
5.2: The actor cancels deleting the field.	
	6.1: The system deletes the field and its information from the database, then shows the farm view.
	6.2: The system shows the edit view of that field.

### A.3.4 Editing a crop field

**Goal:** Edit the information of an existing crop field.

**Actor:** Farm admin or farmer.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin or farmer at the farm selected.

A field has been selected from which the actor wants to change a crop field.

**Postcondition:** The actor has edited information of the crop field in the database.

**Summary:** Select a crop field and edit its corresponding information.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor selects the crop field that they want to edit.	
	2: System shows a view on which the information of the crop field can be edited.
3: Actor makes changes to the crop field.	
4.1: Actor saves changes made.	
4.2: Actor cancels changes made.	
	5.1: System updates information and shows information view of the field.
	5.2: System shows information view of the field.

### A.3.5 Deleting a crop field

**Goal:** Delete an existing crop field from a farm.

**Actor:** Farm admin or farmer.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin or farmer at the farm selected.

A field has been selected from which the actor wants to delete a crop field.

**Postcondition:** The crop field is no longer in the database of the farm.

**Summary:** Select a crop field and delete it.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigate to edit crop field page.	
	2: System shows a view on which the crop field can be edited.
3: Actor selects they want to delete the crop field.	
	4: System asks whether actor confirms to delete the crop field.
5.1: Actor confirms the deletion.	
5.2: Actor cancels the deletion.	
	6.1: System deletes the crop field and redirects actor to the previous view it was on.
	6.2: System shows view on which the crop field can be edited.

### A.3.6 Editing equipment

**Goal:** Edit the equipment information.

**Actor:** Farm admin or farmer.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin or farmer at the farm selected.

The actor is on the information view of the equipment.

**Postcondition:** The actor has edited information of the equipment in the database.

**Summary:** Select a piece of equipment and edit its corresponding information.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to the edit view.	
	2: System shows view on which the equipment can be edited.
3: Actor makes changes to the equipment.	
4.1: Actor saves changes made.	
4.2: Actor cancels changes made.	
	5.1: System updates information and shows information view of the equipment.
	5.2: System shows information view of the equipment.

### A.3.7 Adding equipment

**Goal:** Add equipment to a farm.

**Actor:** Farm admin or farmer.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin or farmer at the farm selected.

The actor is on the information view of a farm.

**Postcondition:** The actor has added equipment.

**Summary:** Select a farm and add new equipment with their specifications.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor goes to the add equipment view.	
	2: System asks the actor for the specifications needed for the equipment.
3: Actor fills in the specifications.	
4.1: Actor saves the specifications.	
4.2: Actor cancels the creation of a new equipment.	
	5.1: System saves the new equipment to the database.
	5.2: System shows the information view of the farm.



### A.3.8 Deleting equipment

**Goal:** Delete existing equipment from a farm.

**Actor:** Farm admin or farmer.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin or farmer at the farm selected.

The actor is on the information view of a piece of equipment.

**Postcondition:** The equipment is no longer in the database of the farm.

**Summary:** Select equipment and delete it.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to the edit view of the equipment.	
3: Actor clicks on delete equipment.	2: System shows a page on which the equipment can be edited. 4: System asks whether actor confirms to delete the equipment.
4.1: Actor confirms the deletion. 4.2: Actor cancels the deletion.	5.1: System deletes the equipment and redirects actor to the previous view they were on. 5.2: System shows view on which the equipment can be edited.

## A.4 FARM ADMIN USE CASES

The following use cases can only be performed by the farm admin of the selected farm.

### A.4.1 Editing a farm

**Goal:** Edit the information and change its specifications, for instance if the farm is public or not.

**Actor:** Farm admin.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin at the farm selected.

The actor is on the information view of the farm.

**Postcondition:** The information and/or specifications of a farm are updated in the database.

**Summary:** Select a farm and edit the data specific to that farm.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to edit view of farm.	
	2: System shows a view on which the farm can be edited.
3: Actor edits the data of the farm and selects they want to save the changes.	
	4: System asks whether actor confirms to keep the changes.
5.1: Actor confirms the changes.	
5.2: Actor cancels the changes.	
	6.1: System updates the data in the database and redirects user to the previous view they were on.
	6.2: System shows view on which the farm can be edited.

#### A.4.2 Adding a new farm

**Goal:** Add a new farm of which the logged in user is the farm admin.

**Actor:** All types of users.

**Precondition:** The actor is logged-in.

The actor is on the farm overview page.

**Postcondition:** A new farm in the database of which the actor is the farm admin.

**Summary:** Navigate to add new farm view, fill in parameters and confirm selection to create farm.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to add new farm view.	2: System shows a view on which multiple parameters to be filled-in are listed and a way to confirm the filled in data.
3: Actor fills in requested parameters.	5: System shows a popup asking for confirmation whether or not the actor wants to add the farm.
4: Actor confirms the filled in parameters.	7.1: System adds a farm with filled in parameters.
6.1: Actor confirms the addition of the farm.	7.2: System cancels the addition of the farm.
6.2: Actor cancels the addition of the farm.	8.1: System shows page on which the details of the new farm are displayed.
	8.2: System redirects the actor to the previous view they were on.

### A.4.3 Deleting a farm

**Goal:** Delete a farm and its information from the database.

**Actor:** Farm admin.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin at the farm selected.

The actor is on the information page of the farm.

**Postcondition:** The data about the selected farm is deleted from the database.

**Summary:** Select a farm and delete it with its information on the database.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to the edit farm view.	
	2: System shows a view in which the farm can be edited.
3: Actor clicks on delete farm.	
	4: System asks whether actor confirms deletion of the farm.
5.1: Actor confirms the deletion.	
5.2: Actor cancels the deletion.	
	6.1: System deletes the data of the farm from the database and redirects user to the previous view they were on.
	6.2: System shows view on which the farm can be edited.

#### A.4.4 Viewing user right details

**Goal:** Get information about the rights of user accounts.

**Actor:** Farm admin.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin at the farm selected.

**Postcondition:** The actor knows the rights a specific user has.

**Summary:** Go to the user management view, selected a user and view its information.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to the user management view.	
	2: System shows a view with a list of users registered in the system.
3: Actor searches for a user.	
	4: System returns a list of users that match the search terms.
5: Actor chooses the user they want to view from the returned list.	
	6: System shows a view with the information on the selected user, including its rights details.

#### A.4.5 Editing user right details

**Goal:** Changing the rights and information of users registered at a farm.

**Actor:** Farm admin.

**Precondition:** The actor is logged in.

The logged-in account has the permission of a farm admin at the farm selected.

**Postcondition:** The user rights of the selected user are changed in the database.

**Summary:** Go to the user management view, select a user and view its information.

**Priority:** Must have.

**Steps:**

<i>Actor actions</i>	<i>System actions</i>
1: Actor navigates to the user management view.	
	2: System shows a view with a list of users registered in the system.
3: Actor searches for a user.	
	4: System returns a list of users that match the search terms.
4: Actor chooses the user they want to view from the returned list.	
	5: System shows a view with the information on the selected user, including its rights details.
6: Actor selects they want to edit the currently viewed user.	
	7: System returns a view where the actor can edit the data of the selected user.
8: Actor changes the data he/she wants to change and selects he/she want to save the data.	
	9: System asks whether actor confirms to keep the changes.
10.1: Actor confirms the changes.	
10.2: Actor cancels the changes.	
	11.1: System updates the data in the database and redirects actor to the previous view.
	11.2: System shows view on which the user can be edited.

# B Appendix B

## B.1 SIGNING PAGE

Hereby the customer and CloudFarmers agree upon the requirements stated in this document.

**Customer**

*Name*

---

*Date*

---

*Signature*

---

**Supervisor**

*Name*

---

*Date*

---

*Signature*

---