POLITECNICO
MILANO 1863

DREAM - Data-dRiven PrEdictive FArMing in Telangana

Requirements analysis and specification document

Version 1.2

Venturini Luca
Vernola Dario
Vittorini Francesco

Release date: December 23rd, 2021

Table of contents

Introduction **4**
- Purpose 4
- Scope 4
  - World Phenomena 5
  - Shared Phenomena 5
  - Goals 6
- Definitions, Acronyms, Abbreviations 7
- Revision history 8
- Reference Documents 8
- Document Structure 8

Overall Description **9**
- Product perspective 9
  - Scenarios 9
  - Class Diagram 14
  - State Diagrams 15
  - Activity diagrams 16
- Product functions 17
  - Manage daily plan 17
  - Interaction farmer - agronomist 17
  - Data collection 17
  - Farmers communication 17
  - Farmers' performances monitoring 18
  - Propose and monitor steering initiatives 18
- User characteristics 18
- Assumptions, dependencies and constraints 19
  - Domain assumptions 19

Specific Requirements **21**
- External Interface Requirements 21
  - User Interfaces 21
  - Hardware Interfaces 21
  - Software Interfaces 22
- Functional Requirements 22
  - Functional Requirements 22
  - Use Cases 27
  - Sequence diagrams 40

Traceability matrices 47
Farmers 48
Performance Requirements 49
Design Constraints 49
Standard compliance 49
Hardware limitations 49
Software System Attributes 50
Reliability 50
Availability 50
Security 50
Maintainability 51
Portability 51
Formal Analysis using Alloy 51
Introduction 51
Code 51
Effort Spent 57
References 58

1. Introduction

1.1. Purpose

The purpose of this document is to specify and illustrate in detail the DREAM application, showing all the functionalities offered to the different users.

DREAM is an application that aims to help farmers of Telangana, a region of India, dense with farmers who every day face different types of problems, starting from an unstable economic situation to changing weather conditions, that put at risk the harvest, and more in general their well-being.

The application will be used in a large area, since Telangana counts 35 million people, part of which are farmers or people involved in agriculture.
Moreover this sector is expanding every year, and so the applications will grow in number of users, and so it is expected to scale and serve more and more people.

The DREAM app has been proposed, in agreement with the government, in order to handle in a smart way the production of food by farmers. In fact in the next few years a huge increment of demand regarding food is expected, and the government wants to face it at the best of its possibilities, safeguarding the economy and, at the same time, the wellness of the farmers.

The objective of this application is to help farmers, involving other figures, like agronomists and policy makers.
Both of these actors will propose steering initiatives in order to improve the production and well being of farmers.
Agronomists will act at the local level, keeping in touch with farmers of the same zone, and helping them on the basis of their requests and their performances.
Policy makers act instead at a higher level, checking that such initiatives are effective and evaluating the farmers' performances.

1.2. Scope

The application will provide a set of services to the different actors involved.

The farmers will be able to interact with their peers, they can access data that can help them in production, like weather forecasting or data retrieved by sensors and they will be encouraged to contact agronomists if needed.
Agronomists will be required to monitor the farmers living in their same zone, visiting the farms multiple times during the year, to check how they are working and answering their requests or helping them if needed.
Finally Policy Makers monitor the data produced by farmers and agronomists, to understand if the steering initiatives are working properly, and intervene when necessary introducing new strategies.

1.2.1. World Phenomena

WP1	The agronomist inspects a farm
WP2	The farmer uses resources for production
WP3	The farmer quantifies the daily production
WP4	The farmer faces a problem in his/her farm

1.2.2. Shared Phenomena

	DESCRIPTION	**CONTROLLED BY**
SP1	The agronomist inserts the area for which she/he is responsible	WORLD
SP2	The agronomist receives requests (messages) from farmers	MACHINE
SP3	The agronomist answers to farmers' messages	WORLD
SP4	The agronomist checks relevant parameters	WORLD
SP5	The agronomist checks the daily plan of farms to be visited	WORLD
SP6	The agronomist updates the daily plan	WORLD
SP7	The agronomist introduces a new steering initiative	WORLD
SP8	The system highlights under-performing farmers	MACHINE
SP9	The system highlights well-performing farmers	MACHINE
SP10	The policy maker marks a farmer as a model	WORLD
SP11	The policy maker marks a farmer as "to-be-helped"	WORLD
SP12	The system sends a notification to a farmer who was marked as a model and invites him/her to publish his/her best practices on the forum	MACHINE
SP13	The system sends a notification to a farmer who was marked as "to-be-helped" inviting him/her to ask for help from the agronomists or other farmers	MACHINE
SP14	The system sends a notification to a farmer involved in an initiative, describing its content	MACHINE
SP15	The policy maker uploads a new steering initiative	WORLD

SP16	The system shows the results of a steering initiative	MACHINE
SP17	The agronomist works on the daily plan, specifying the deviations from the given plan	WORLD
SP18	The agronomist confirms a farm registration	WORLD
SP19	The farmer sends a registration request for his farm	WORLD
SP20	The farmer answers other farmers help requests	WORLD
SP21	The system shows the current available parameters to farmers	MACHINE
SP22	The farmer asks for advice to other users	WORLD
SP23	The farmer uses the forum	WORLD
SP24	The farmer uploads his daily production	WORLD

1.2.3. Goals

Goal	**Description**
G1	The policy makers should be able to identify those farmers who are performing well
G2	The policy makers should be able to identify those farmers who are performing badly
G3	The policy makers and agronomists should be able to understand whether a steering initiative produces significant results
G4	Agronomists should be able to decide and execute the best possible farm visit plan
G5	Farmers and agronomists should be able to access environmental informations to work in the best manner
G6	Farmers should be able to seek help from peers and agronomists to improve their production

1.3. Definitions, Acronyms, Abbreviations

Daily plan: Plan made by agronomists that contains a list of farms to visit on a specific day.
Sensor: We refer with this term to all the embedded devices used to get information from parameters of the surrounding environment
DREAM: It's the name of the application we aim to develop.
Application: The DREAM application.
Steering Initiatives: Initiatives carried out by agronomists and policy makers to help the farmers in performing better.
Metrics: It refers to the values of the monitored parameters
Visit a farm: The action of the agronomist that goes to a farm, checks the conditions and discuss with the farmers
Competence zone: The zone assigned to a specific agronomist, of which she/he is responsible.
Credentials: Username and password required to access to DREAM
Institutional credentials: Particular kind of credentials assigned to a policy maker by the government.
Performance analysis: Computation that takes as input various parameters as level of production, water consumption and other data of a particular farmer and gives as output a numerical score.
Performance analysis results: numerical value computed by the performance analysis.
Performance threshold: numerical value determined in advance by some experts in order to determine a threshold under which the performance analysis results are considered bad. If the results are up to the threshold they are considered good.
Well-performing farmer: Farmer whose performance results are up to the performance threshold.
Under-performing farmer: Farmer whose performance results are under the performance threshold.
Model: A farmer who is performing particularly well can be marked as a model by a policy maker. He/She is considered as an example for all the other farmers.
To-be-helped: A farmer who is performing particularly badly and needs to be helped by agronomists or other farmers.
Topic: a thread of the forum made of a written text open for discussion, published by an user, to which other users can reply to.
Section: sections act as "folders" to collect topics regarding a specific theme.
Moderator: a user of the forum who can suspend users and delete topics and replies
Suspension: a suspension is a penalty given to a user who behaved badly in the forum. It lasts for a variable amount of time, during which such user is not allowed to comment or create new topics on the forum.
UI: it stands for User Interface, and is the graphical interface that a user is presented with and which allows him to interact with the system.
User-friendly: an user-friendly interface is by definition easy to use by almost any kind of user without any specific background.

1.4. Revision history

Version	Description

1.0	Initial version
1.1	Revision of requirements concerning steering initiatives
1.2	Full revision and details definition

1.5. Reference Documents

All the references are presented in **section 6**.

1.6. Document Structure

In **section 1** a general overview of the project is presented; in the purpose and in the scope it is briefly described the context in which the application will be used and the motivations for which an application like DREAM has been commissioned.
Moreover an initial description of World and Shared phenomena is presented, alongside a list of high level goals and a glossary to explain keywords used in the following pages.

In **section 2** the product is presented describing different scenarios that include the main features of the application.
Diagrams like class diagrams and state-charts are used to present more details about the system model.
After a brief description of the main actors the domain assumptions and requirements are listed.

In **section 3** we can find details about functional and nonfunctional requirements.
The interactions between the Actors and the system are represented using Use Cases which describe more in detail what is mentioned in the Scenarios section.
Finally we can find tables describing where and when the different requirements are used.

Section 4 is dedicated to Alloy and contains a formal representation of what is described in the document.

Section 5 and **6** contains, respectively, an estimation of the time spent on the project for each of the components of the group and a list of references used to write this document.

2. Overall Description

2.1. Product perspective

In this section the main possible scenarios are presented, furthermore class diagrams and state diagrams are used to describe the system.

2.1.1. Scenarios

Agronomist

Scenario 1: Insert related area

Raul has just been hired as an agronomist by Telangana government, and he has been assigned an area of which he is responsible.
He visits the DREAM home page, logs in as Agronomist with his username and password given directly by a government subordinate, and visits his personal page.
Here he can add the area to which he has been assigned, selecting among all the areas proposed.
Raul confirms his choice and checks his email, where he will find a summary of the operation.

Scenario 2: Help a farmer

Altaaf, one of the agronomists, receives on his smartphone a notification from DREAM, he reads that the notification is about “messages from farmers”.
He opens the DREAM platform from his pc, logs in and selects, from the menu, the option to see all the incoming messages.
He selects the last one and reads it. He found out that a farmer is writing to him to ask for some advice regarding the type of fertilizer to use, since he has seen that the weather will be quite bad for the next two weeks.
The agronomist, after double checking the weather to get more precise information, checks the historical data looking for farmers performing well in the same situations during the past years and in particular the fertilizers they used.
He notices that the majority of them used the same specific type of water resistant fertilizer.
The agronomist decides to suggest such product to the farmer, and he replies to him with a short message.

Scenario 3: Execute daily plan

Manpreet, a recently hired agronomist, arrives at his office at 8.30.
He opens his laptop and connects to the DREAM platform to visualize his daily plan.
After the usual login phase he clicks the Daily Plan button from the menu and starts reading it.
The DREAM platform shows him the farms he should visit today, with some details regarding the location of the farms and the (good) performance obtained during the last visit.
The agronomist drives to the farms, asks for some data to the owner and inspects the farm, moreover he talks to the owner of the farm giving him some advice based on his knowledge and finally gets back to his office.
The agronomist fills the form proposed by DREAM related to the daily plan.
Manpreet confirms the execution of his daily routine and closes the DREAM page.

Scenario 4: Compose the daily plan

Raul, an agronomist, after his daily routine, wants to prepare the daily plan for the following days.
He opens DREAM and logs in. After having selected the option to create daily plans, the system shows him a list of farms to be visited, highlighting the ones that are outperforming and the ones that aren't performing well, alongside some notes next to the corresponding farm, specifying problems reported by the farmers.
Raul composes the schedule for the following days mixing some farms that will be quickly visited and others that need more attention and saves it, so that it will be proposed the day after.

Scenario 5: Execute daily plan with variations

Gurjeet, an agronomist, after checking the daily plan, discovers that today he has to visit farm A and farm B, and that the first one is performing very badly, and has been marked as urgent by the Policy Maker.
He drives to the first farm and starts looking around and talking to the farmers working there. He finds out that the farmers aren't following the suggestions given by the agronomist during the last visit, additionally, a storm has destroyed some machines used in the fields, as they reported in the DREAM app some days ago.
Gurjeet spends the whole day trying to set-up a new plan to increase the production and showing to the farmers some techniques that may help to improve production without the necessity of machines.
When Gurjeet returns to his office he fills the form regarding the first farm; since the farm is performing badly it is marked as visited but the agronomist already knows he will have to return there more than two times.
He then has to specify a deviation to the current plan, since he hasn't been able to visit the second farm, and marks as unvisited the farm B.

Scenario 5: Confirm farmer registration

Deepa receives a notification on her smartphone, and reading it she discovers that a new farmer, from her competence zone, has been registered to the DREAM application. She sends a message to set an appointment to visit the farm and confirms the registration on the portal.
After some messages exchanged between the agronomist and the farmer a date is decided.
On the day of the agreed date the agronomist visits the farm and, with the help of a technician, they install the appropriate sensors for the farm and, once returned to his office, she confirms the registration of the new farm.
The system sends a notification to the farmer that has now been accepted in the DREAM application.

Scenario 6: Create a new steering initiative

Devaki, one of the agronomists, before creating a daily plan, wants to check some important parameters related to her competence zone.

She opens DREAM, clicks on the button to see all the farmers with some parameters and, among them, she notices, also thanks to a warning, that many of them are in trouble with the production of Rice since this year rain has been scarce, probably due to climate change.
Devaki wants to help those farmers that are facing these problems and so she decides to think of an initiative to promote the purchase of instruments that helps to collect water while raining, and reuse it in an efficient way when needed, without waste.
Devaki creates on DREAM a new steering initiative, she inserts a description, the monitoring metrics, the time limit and the list of farmers involved.
After that she contacts a company that produces the instrumentation required for her scope and she manages to obtain a good discount for all the farmers that will buy it.
At this point Devaki sends a message to all the interested farmers about this opportunity and, when visiting the farms, she will continue to suggest the new instrumentation.

Policy makers

Scenario 1: Check farmers' performances

Mario, a policy maker of Telengana's government, wants to have a look at the results that each farmer is obtaining in a particular period of time.
First, he logs in the DREAM platform with his credentials. From the dashboard he selects the option to visualize the data related to each farmer of a specific area.
DREAM shows the results and highlights all the farmers who are performing well and those who are performing badly.

Scenario 2: Deciding a new steering policy

Deepa, one of the policy makers, wants to analyze the performances and the results of the farmers in the district of Medak, which is, in turn, divided into different zones, assigned to different agronomists.
Deepa notices, looking at the pre-processed statistics, that soybean isn't much considered in that area, and very few farmers are growing that type of crop.
Since an increase of soybean demand is expected, Deepa decides to promote the production of soybean by creating a steering initiative.
Deepa accesses DREAM, she selects the option to create a new steering initiative, she types the problem to solve, she selects the metrics to be observed (the number of tons of soybean produced), sets a time limit (which can be extended) and associates the initiative to all the farmers of the Medak district. All the farmers and agronomists of the Medak district will receive a notification with the new initiative description.

Scenario 3: Understand if a steering policy is having good results

Paul, a policy maker of Telangana's government, wants to check if a particular steering initiative is producing the desired effects.
First, he logs in the DREAM platform with his credentials. From the dashboard he selects the option to check the effects of a particular steering initiative. Then, DREAM displays a list of the steering initiative proposed by agronomists and policy makers. Paul selects the initiative for which he wants to check the results. Finally, DREAM displays a list of the farmers who have carried out that particular initiative with the related results.

Scenario 4: Mark a farmer as a model

John, a policy maker of Telangana's government, has selected the option to check farmers' performances from his DREAM dashboard and now the system is displaying the list of farmers with the relative results. He wants to mark some farmers who are performing particularly well as models for all the other farmers. For every farmer chosen, he selects the option to mark him/her as a model. DREAM returns a confirmation message and sends a notification to all the selected farmers inviting them to publish their practices in the forum with all the other farmers.

Scenario 5: Mark a farmer as "to-be-helped"

Emily, a policy maker of Telangana's government, has selected the option to check farmers' performances from his DREAM dashboard and now the system is displaying the list of farmers with the relative results. He wants to mark some farmers who are performing particularly badly as farmers "to-be-helped". For every farmer chosen, he selects the option to mark him/her as "to-be-helped". DREAM returns a confirmation message and sends a notification to all the selected farmers inviting them to ask for help from agronomists or other farmers.

Farmers

Scenario 1: Registering a farm
Sunita has recently received multiple notes from the government regarding a new platform called DREAMS, which is said to be able to improve and help the production of Telangana farms, and instructions to join the program. She speaks with other nearby farmers and convinces herself to sign in following the given instructions.
She opens the DREAMS page through her laptop and inserts her information and credentials, such as her first and last name, her email, alongside the information of the farm itself, such as its location, the type of produced goods and the soil type. She confirms and sends the information to the system. The registration will be completed after a visit carried out by an appointed agronomist.

Scenario 2: Asking for help
Dinesh, a local farmer, while checking on the crops, notices that an insect that he's never seen before is slowly damaging them. Being a cautious person, before applying a common pesticide, he decides to ask for help using the newly introduced DREAM application. He opens it via his phone, logs in, and checks the forum where he usually interacts with his fellow colleagues. Strangely, no similar topic has been created, so he exits from the forum and sends a help request through the correct interface, describing his problem carefully. He waits a while and then receives a response telling him the procedures to follow to get rid of the insects.

Scenario 3: Answering a help request
Amit, before starting his daily routine in the farm, checks the DREAM application and sees a help request from a colleague and friend, Dinesh, who lives a few miles away from him. He reads the message and realizes that just the day before he experienced the same exact

problem, which he solved by spraying his crops with a product of a certain local brand, so he decides to answer Dinesh by telling him such solution. He then closes the app and starts his routine.

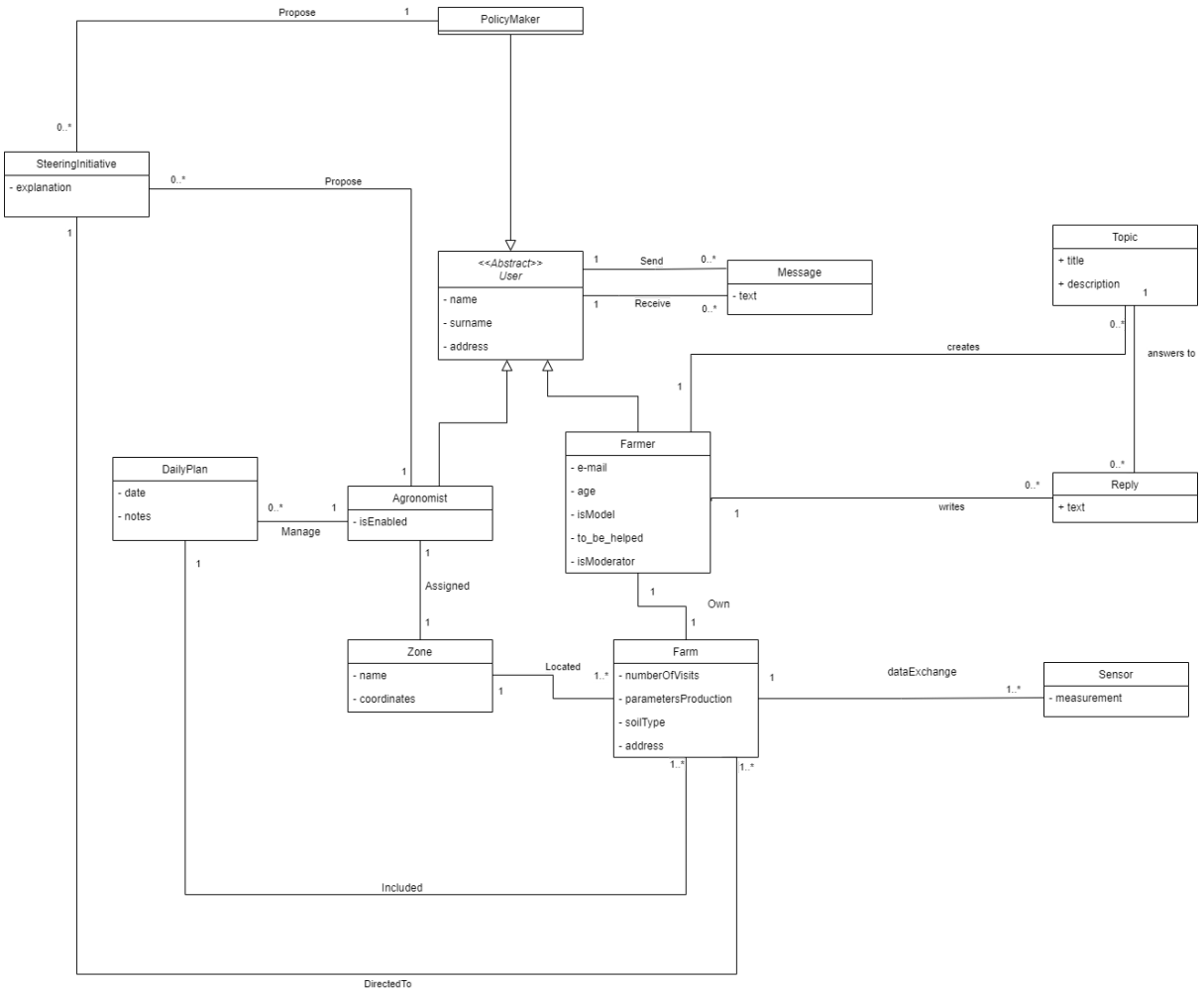
Scenario 4 : Visualizing relevant parameters
Maya has always been very careful and thoughtful about her business so she always tries to plan ahead her work schedule for the week. In order to precisely define the workflow she needs to know what the weather is going to be, so she accesses her profile in the DREAM app through her laptop and selects the section regarding the daily and weekly relevant parameters. Here she is able to visualize not only the weather forecast but also the soil humidity and a list of suggestions provided by the system based on such parameters. She notices that rain is expected for the following week and that the system is suggesting a reduction of 60% of the water used for irrigation for the day so she takes this into account and plans her week accordingly.

Scenario 5 : Using the farmers forum
Ravi is a really sociable person and loves interacting with other people, especially with those who share his same passion for agriculture. For this reason everyday he checks the forum provided by the DREAM system and sees if there are new answers to his favourite threads or if new topics were added. He finds a new topic about the best fertilizers used for sugarcane cultivation which happens to be one of the major products of his farm and decides to leave a reply explaining his experience. Inspired by this thread he decides to create a new topic regarding irrigation machines to see if some of his colleagues have more advanced systems. He then closes the app and goes to work.

Scenario 6 : Sending data
Pankaj just finished a hard working day and is about to go home to his family. Before leaving he checks, as he always does, the daily consumed resources and the collected crop, he then goes to his office, logs in into the DREAM system and selects the option for the daily rundown and proceeds to type the parameters just checked a moment earlier. Before uploading he takes advantage of the moment to write a side note in the correct textbox about a weird color assumed by some of the plants which he cannot link to any phenomena experienced before. He then uploads the data, logs out of the system and goes back home.

Scenario 7 : Moderating the forum
Inesh, a 57 years old soybean farmer, has believed in the DREAM platform since day one and he was one of the first to register himself and his farm to the system. Being one of the elder users, he is automatically appointed as a moderator for the farmers' forum. Conscious of his responsibility, he checks daily the content of the forum, overseeing and verifying that every other user respects the rules. He sees that an angry farmer is heavily cursing in a topic regarding rice cultivation, so he decides to remove his comments and suspend the user from replying for 8 hours. He uses this occasion to write him a message telling him the reason behind the suspension and to kindly ask him to respect the rules next time.

2.1.2. Class Diagram



The UML diagram shown above models the relationship between the main element of the DREAM platform.
The following boolean attributes define statuses of farmers

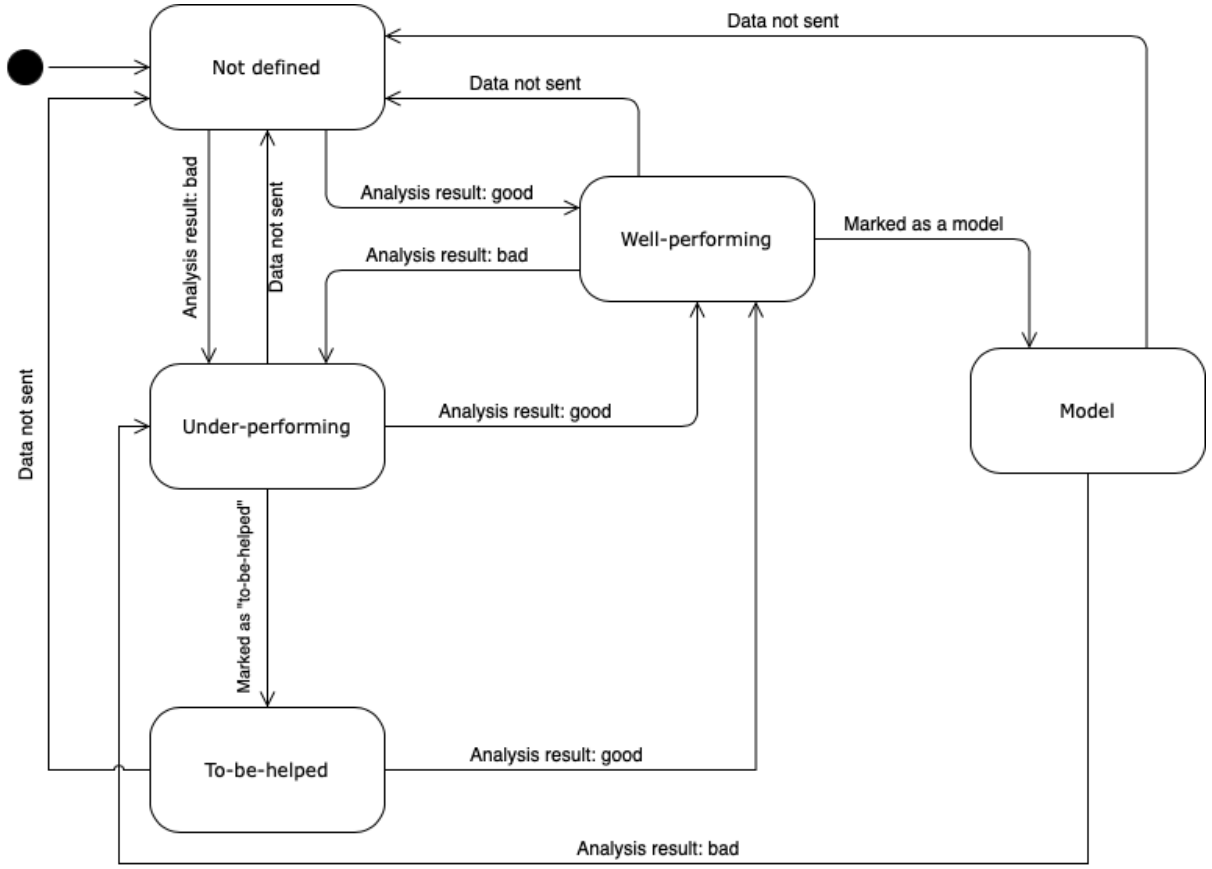
- **isModel:** this attribute defines whether a farmer is marked as a farmer model by a Policy maker
- **to_be_helped:** this attribute defines a farmer whom, after performing particularly badly, is assigned the “to-be-helped” status explained in section 1.3
- **isModerator:** this attribute states whether a given farmer has been appointed moderator of the DREAM forum, based on his seniority. It was designed as an attribute and not as an extension of the “farmer” class to reduce redundancy and to model the case in which a regular farmer is made moderator and vice versa.

The agronomist attribute “**is Enabled**” is set to true as soon as a given agronomist is assigned to a certain zone of the Telagnana region.

The farm attribute “**parametersProduction**” states the kind of production of the given farm (e.g. soybeans, rice, etc…).

2.1.3. State Diagrams

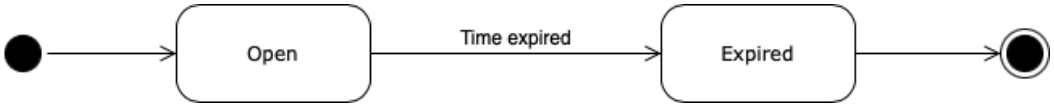
2.1.3.1. Farmer



Initially, when the farmer has not uploaded his/her production yet, his/her state is “Not defined” because the system cannot compute the analysis to determine if his/her results are good or bad. When the farmer uploads his/her production for the first time and every time the farmer uploads new data, the system performs the analysis on the data and determines if he/she is performing well or badly. Based on this result the new farmer state is “Well-performing” or “Under-performing” respectively. When a farmer is well-performing, if a policy maker marks him/her as a model, the new state is “Model”. When a farmer is bad-performing, if a policy maker marks him/her as “to-be-helped”, the new state is “To-be-helped”. We assume that a farmer marked as a model remains in this state until the performance analysis results become bad, so the farmer will move to the state “Under-performing”, not being a model anymore. We assume also that a farmer marked as “to-be-helped” remains in this state until the performance analysis results become good, so

the farmer will move to the state “Well-performing” and he/she is not marked as “to-be-helped” anymore. Regardless of the current farmer state, if he/she has not uploaded the updated data on the production for a long period of time, his/her state becomes “Not defined”.

2.1.3.2. Steering initiative

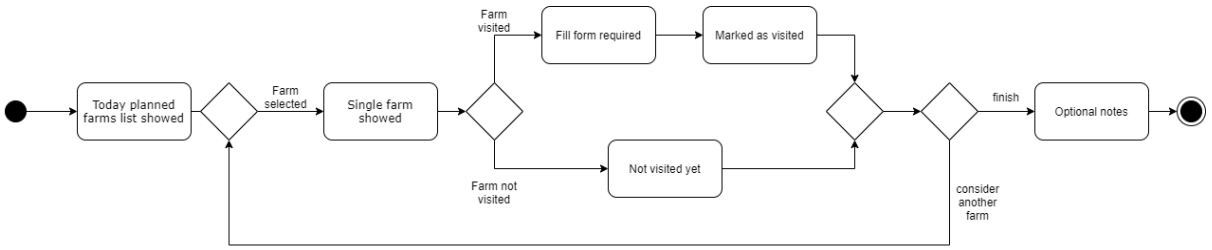


A steering initiative is considered open for a limited period of time (specified in the creation process). When this period expires, the steering initiative is considered expired and will be destroyed.

2.1.4. Activity diagrams

2.1.4.1. Daily plan confirmation

The following diagram describes how the agronomist interacts with the DREAM platform to confirm the execution of his daily plan.
The agronomist selects each “planned to visit” farm and executes the appropriate actions.



2.2. Product functions

This section provides a summary of the major functions that the software will perform w.r.t. the goals already described in section 1.2.3

2.2.1. Manage daily plan

The function allows the agronomists to handle the plan for the following days.

Agronomists can use the daily plan as a sort of agenda on which they add or remove, given a specific date, the farms to be visited, considering the statistics regarding the specific farm.
Moreover the daily plan must be confirmed once executed, describing if some deviations from the plan happened.

2.2.2. Interaction farmer - agronomist

A farmer can interact with the agronomists sending messages to them. The farmer selects as the recipient the agronomist of his area. He can write a short message to explain his problems or ask for suggestions, then he can send the message.
The agronomist, in the messages section of the DREAM application, can check all the messages, read them and, when he is ready, he can reply to the farmer. Both farmers and agronomists will receive a notification highlighting new messages incoming.
The same can happen also inverting the roles, in fact agronomists can directly contact a farmer if they have to suggest practices or solve problems.

2.2.3. Data collection

DREAM has to be able to collect data from across the region regarding multiple environmental parameters, such as soil moisture, texture and depth, weather forecasts and wildfires but also data regarding the farms and the farmers registered to the platform. The data of interest is composed mainly of the quantity and quality of harvested crops, water consumption and daily weather conditions. The combination of these data is required in order to improve political decision making and help the farmers of the region.

2.2.4. Farmers communication

The application is required to stimulate communication between farmers in order to solve problems quickly if they were already solved by some farmer or just to improve an already existing reality.
This feature is guaranteed through a forum section, allowing farmers to interact and stimulate conversations regarding their work and their situation and to solve minor issues, but also through a “help request” system which allows farmers in need to send help/advice requests to nearby farmers and agronomists.

2.2.5. Farmers’ performances monitoring

DREAM is really useful for policy makers in order to monitor the results obtained by farmers in Telangana. The platform collects all the relevant data and, by a suitable processing, is able to determine if a farmer is performing under or up to the performance threshold, classifying farmers in two classes: well-performing and under-performing. In this way, a policy maker can understand quickly and easily what is the current general situation in the various geographical areas.

2.2.6. Propose and monitor steering initiatives

The main goal of Telengana's government is to help farmers to increase their production and to be resilient to climate changes. To this end, policy makers and agronomists can decide to carry out steering initiatives in particular areas of Telangana. DREAM allows these people to upload steering initiatives on the platform and monitor their effects through the time.

2.3. User characteristics

The Actors that will interact with the DREAM application are the following:

1. **Farmers:** The farmer is a person who owns a farm in the Indian region of Telangana. As shown by statistics, farmers living in Telangana often are in a difficult economic situation. Thanks to this application, they will be supported in taking the best decisions and improving their life.
   Farmers usually come from different technological backgrounds, and not all of them are familiar with PCs or smartphones. They will need an easy-to-use system that requires little time to be learned.

2. **Policy Makers:** Policy makers are part of the Telangana government. They have a good background in economics and IT, and they will use DREAM as a support to make appropriate choices, combining the data provided by the application with their knowledge in the field.

3. **Agronomists:** Agronomists are experts in the science of soil management and crop production, they are assigned to a zone, of which they must monitor the situations. Most of their time spent at work consist of visiting farms, they will use the DREAM application just to schedule their plans, hence the application will have to be immediate to use but at the same time detailed enough to display all the relevant data they may need.

2.4. Assumptions, dependencies and constraints

2.4.1. Domain assumptions

Agronomists

NUMBER	ASSUMPTION

AD1	Every actor involved has access or is provided with a smartphone and/or PC
AD2	The agronomist answers to the famers in a reasonable amount of time
AD3	The weather forecasts are in general quite accurate and reliable
AD4	Everyday the agronomist checks his daily plan
AD5	Everyday the agronomist works according to his daily plan
AD6	The agronomist confirms everyday the work done, without lying
AD7	The Agronomist visits a farm at least twice a year
AD8	At any given time each farmer is associated to at most one steering initiative
AD9	The credentials are provided to the agronomist at the moment of their nomination
AD10	An area has to be assigned to only one agronomist
AD11	The agronomist selects the right area to which he was assigned by the policy makers

Policy makers

Assumption	**Description**
PD1	Each policy maker has his/her own institutional credentials to log into the platform
PD2	Each agronomist is assigned by a policy maker to a certain zone of the region

Farmers

NUMBER	**Description**
FD1	Every farm is equipped with the necessary technology (e.g. sensors).
FD2	Each farmers uploads truthfully the designated parameters at most once a day
FD3	Every farmer knows how to collect relevant data
FD4	Farms can be registered to the system only after a careful inspection carried out by an agronomist
FD5	Every farmer owns exactly one farm

FD6	If a farmer is associated with a steering initiative, he/she will carry out that initiative

Sensors can be pretty expensive and it is not expected that the farms involved are properly equipped with such devices, for this reason it is assumed that the government of Telangana, being this a really sensitive issue, provides farms with the correct equipment once a farm is registered.
For this reason, farms will also need to be inspected by qualified personnel (e.g. agronomists in this case) before completing the registration process, so that the correct devices can be installed and relevant information (such as soil and production type) is inserted correctly.
Daily uploads of parameters are not mandatory (it wouldn't be reasonable) but they can only be carried out once per day, preventing multiple uploads which may result in bad and incorrect reads on the algorithm used for decision making.
It is expected, of course, that all of the farmers involved know how to collect data which cannot be read automatically by sensors, such as production quantity.
The choice to associate at most one steering initiative at time with each farmer is motivated by the fact that in this way we can clearly understand the effects of that particular initiative, without being influenced by other concurrent initiatives.
We assume that every person has at least a technological device to execute the operation needed, since we have people with different economic situations we don't assume stricter conditions.
Finally we assume that each actor behaves correctly with respect to the insertion of data, since this part can't be controlled by the system but we must rely on people's honesty.

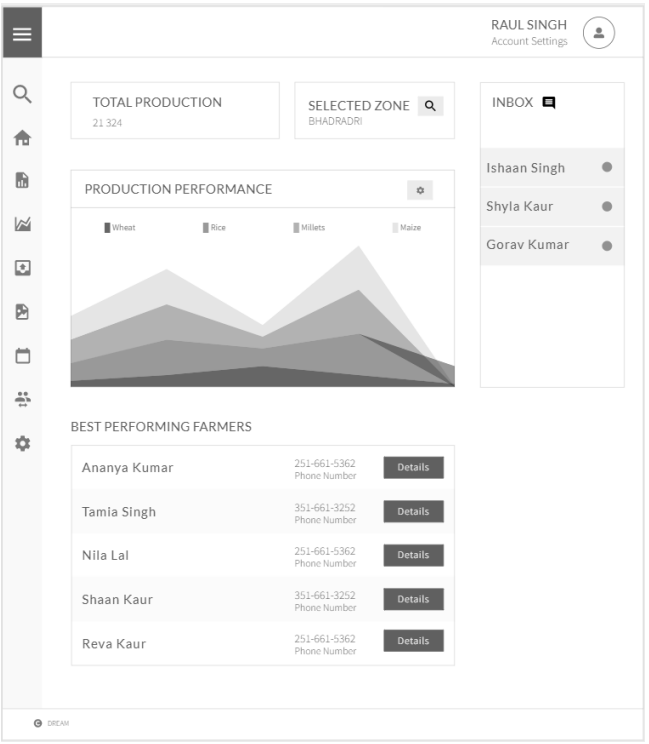
3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

The user will interact with the system using a Graphical Interface, available both on Pc and smartphones/tablets.

Since the actors involved in the use of DREAM are various and heterogeneous, they may not have a strong technological background, hence the UI needs to be as user-friendly as possible.
The following image represents a mockup of the application interface, specifically the dashboard of a policy maker or agronomist user. It shows relevant data, such as productions, best performing farmers and most recent messages.



More precise and detailed representation of other UIs will be presented in the Design document.

3.1.2. Hardware Interfaces

DREAM will necessarily require access, whenever available, to sensors dispatched all over the region territory and, most importantly, on the registered farms. The main sensors of interest would be water volume and soil humidity sensors, in order to provide farmers and policy makers with the most precise kind of information in an autonomous and automatic way, possibly better than a “manual” measurement carried out by the farmers.
It would be better also to have access to a GPS module, in order to understand where the user is located and show him the weather forecast related to his zone.
Obviously all the users will have the possibility to access DREAM through devices like PCs, smartphones or tablets connected to the internet.

3.1.3. Software Interfaces

DREAM will have to communicate, through specific APIs, to data collection centers, mainly regarding parameters constantly checked by regional systems. In particular, a connection is required to the region's weather forecast, soil humidity and wildfire data centres. These connections are fundamental to the main scope of the application, being a very important support to farmers in the crops production process.

It will also be required to communicate with a geo-localization service, in order to efficiently map farms on the territory upon registration and to better provide information regarding the parameters used by the farmers (e.g. weather forecasts).

3.2. Functional Requirements

3.2.1. Functional Requirements

Functional requirements are divided by actors to improve readability.

3.2.1.1. Agronomists

ID	FUNCTIONAL REQUIREMENT
AR1	The system shall allow an agronomist to log in
AR2	The system shall allow an agronomist to log out
AR3	The system shall send a confirmation email to an agronomist once they select the area of responsibility
AR4	The system shall check the consistency of data inserted by users
AR5	The system shall allow the agronomist to send messages to farmers
AR6	The system shall let the Agronomist to confirm the registration of a farmer, after the Agronomist has visited the related farm
AR7	The system shall show the agronomist all and only the farms he can visit during the daily plan creation process
AR8	The system shall correctly store each update of the daily plan
AR9	The system shall allow Agronomist compare farmers performances
AR10	The system shall let Agronomists to delete farms from their future daily plan
AR11	The system shall let Agronomists to add farms to their future daily plan
AR12	The system shall let Agronomist to indicate deviations from their “pre-compiled” daily plan
AR13	The system shall let Agronomists to confirm the execution of a daily plan
AR14	The system shall keep track of how many times a farm has been visited
AR15	The system shall show always the more updated values retrieved from sensor
AR16	The system shall allow Agronomist to monitor created steering initiatives

3.2.1.2. Policy makers

ID	FUNCTIONAL REQUIREMENT
PR1	The system shall allow a policy maker to log into the system with his/her institutional credentials

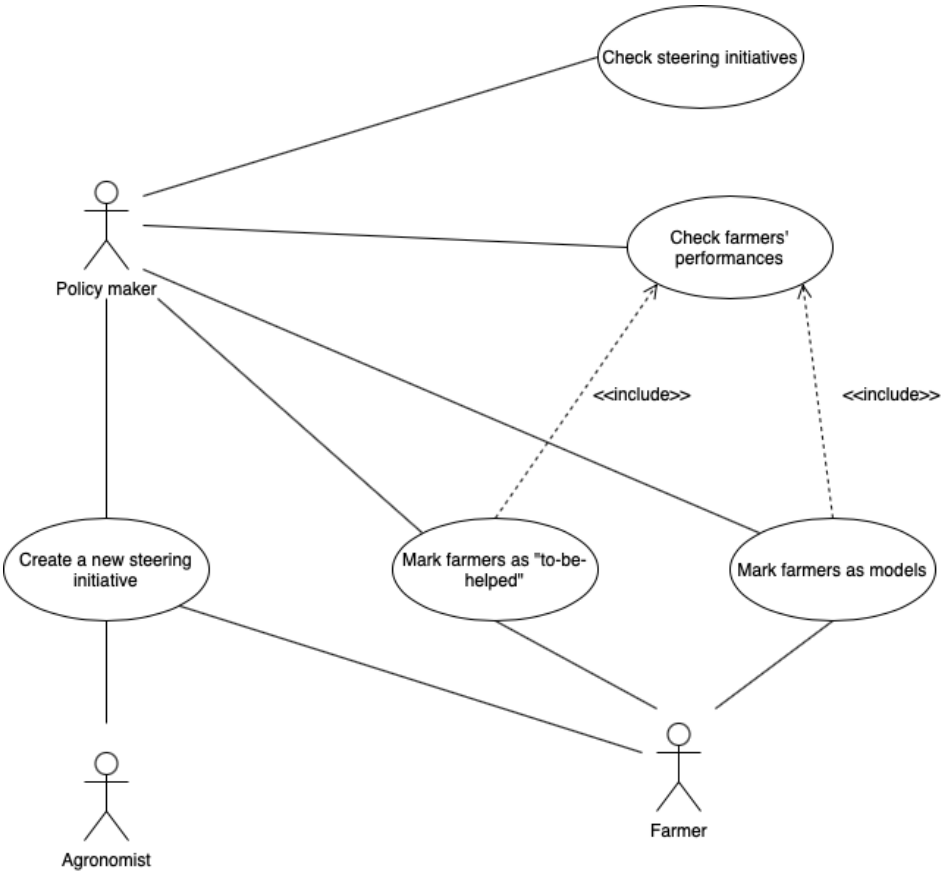
PR2	The system shall allow a policy maker to logout from the system
PR3	The system shall allow a policy maker to upload a new steering initiative
PR4	The system shall show to a policy maker the list of well-performing farmers
PR5	The system shall show to a policy maker the list of under-performing farmers
PR6	The system shall allow a policy maker to mark a well-performing farmer as a model
PR7	The system shall allow a policy maker to mark an under-performing farmer as “to-be-helped”
PR8	The system shall show to a policy maker the results of the farmers involved in a steering initiative
PR9	When a farmer is marked as a model, the system shall send a notification to that farmer inviting him/her to publish him/her best practices on the forum
PR10	When a farmer is marked as “to-be-helped”, the system shall send a notification to that farmer inviting him/her to ask for help from an agronomist or other farmers
PR11	Each time a farmer uploads new data about his/her production, the system performs the performance analysis and marks the farmer as well-performing (or model if he/she is already marked so) if the results are up to a certain threshold, under-performing otherwise (or to-be-helped if he/she is already marked so)
PR12	If a farmer has not uploaded new data about production for a month, the system shall mark his/her state as not defined regardless of the current state

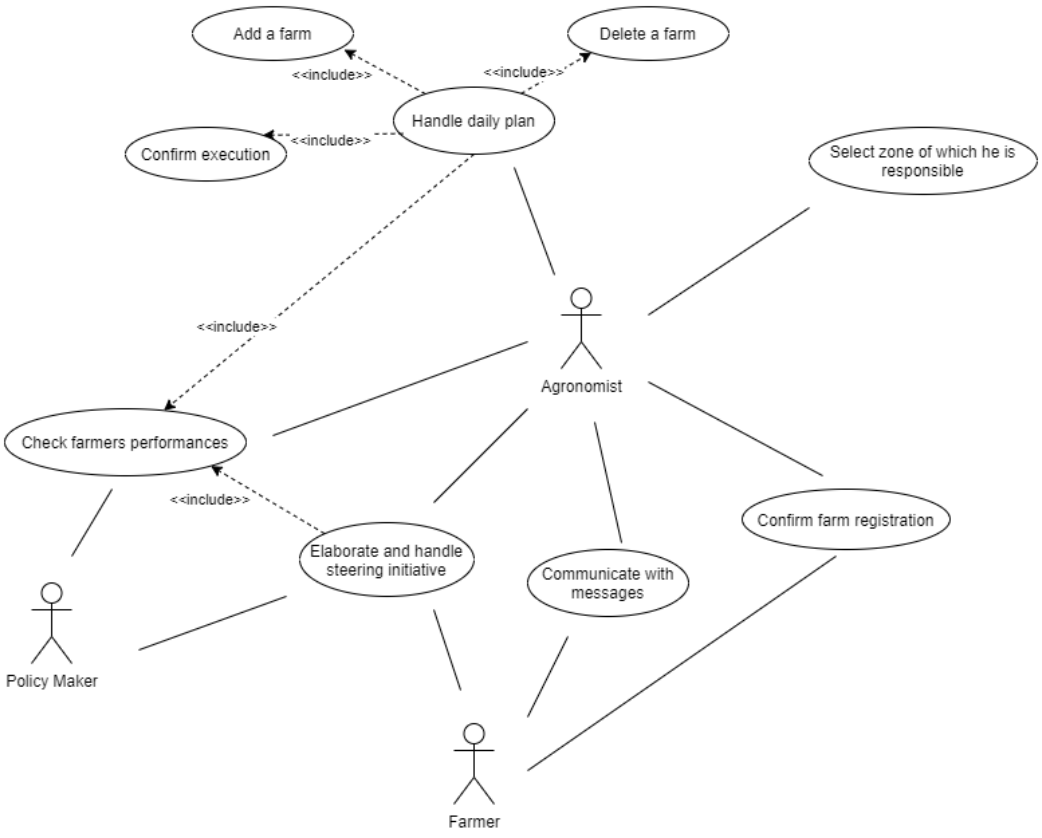
3.2.1.3. Farmers

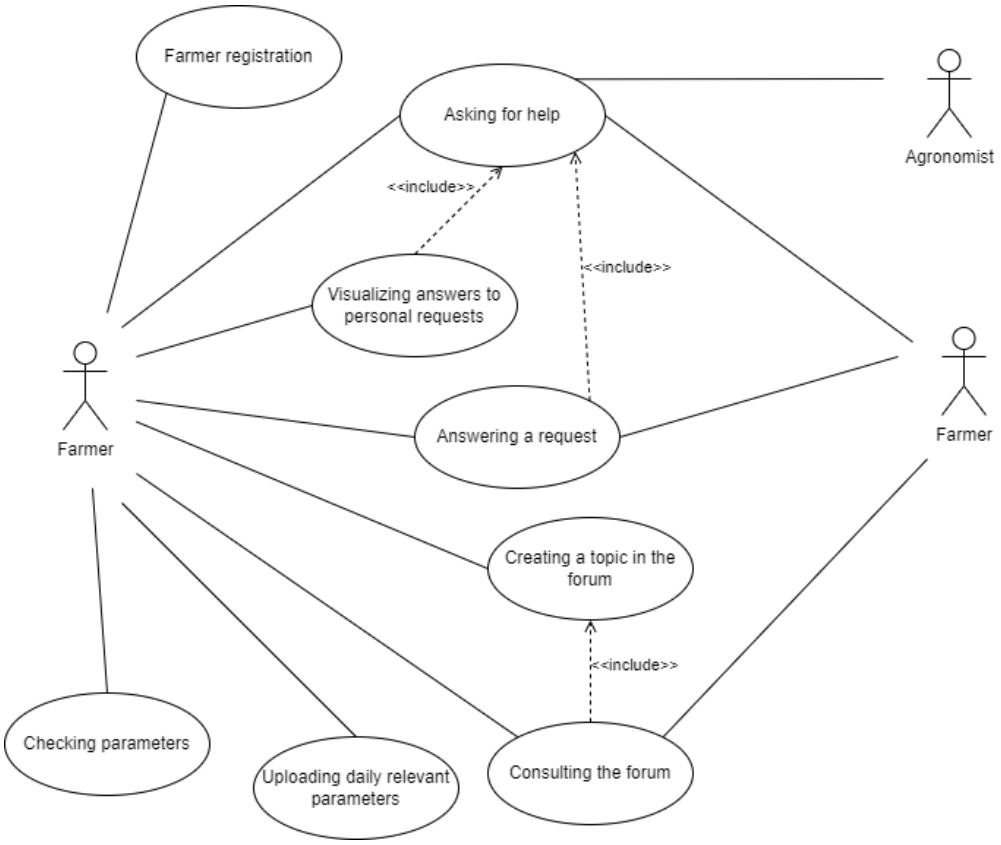
ID	FUNCTIONAL REQUIREMENT
FR1	The system shall allow farmers to register themselves
FR2	The system shall allow registered farmers to register their farms
FR3	The system shall send a confirmation email after the registration process
FR4	The farmer shall be able to check the weather forecast
FR5	The farmer shall be able to check soil humidity values
FR6	The farmer shall be able to check water consumption values
FR7	The farmer shall be able to send help requests to other farmers

FR8	The farmer shall be able to send help requests to agronomists
FR9	The system shall send a summary email to the issuer after the help request
FR10	The farmers shall be able to reply to issued help requests
FR11	The system shall send a summary email to the farmer who responds to a help request
FR12	The farmers shall be able to check the replies given to every given issued request
FR13	The farmers shall be able to close an help request
FR14	The system shall notify those who replied if the issue is still open after the issuer checks the replies
FR15	The farmer shall be able to upload the daily production information to the system
FR16	The farmer shall be able to automatically upload daily water consumption values if available
FR17	The system shall prevent multiple daily uploads
FR18	The system shall provide a discussion forum to all registered farmers with non-customizable sections
FR19	The farmers shall be able to create new topics in any given section
FR20	The farmers shall be able to reply to any given forum topic
FR21	The system shall automatically appoint senior farmers as forum moderators
FR22	The moderators shall be able to delete topic replies
FR23	The moderators shall be able to delete replies
FR24	The moderators shall be able to suspend users
FR25	The system shall send a notification every time a new reply has been added to a recently topic to which an user has replied
FR26	The farmers shall be able to log into the system
FR27	The farmers shall be able to log out of the system

3.2.2. Use case diagrams


Check steering initiatives
Policy maker
Check farmers' performances
<<include>>
<<include>>
Create a new steering initiative
Mark farmers as "to-be-helped"
Mark farmers as models
Agronomist
Farmer


Add a farm
Delete a farm
<<include>>
<<include>>
Handle daily plan
Confirm execution
<<include>>
Select zone of which he is responsible
<<include>>
Agronomist
Check farmers performances
<<include>>
Confirm farm registration
Elaborate and handle steering initiative
Communicate with messages
Policy Maker
Farmer



3.2.3. Use Cases

The following section explains in more detail what is presented in the diagram above.

3.2.3.1. Agronomist - AU1 - Use case 1

Name	Add an agronomist's zone
Actor	Agronomist
Entry condition	An agronomist needs to add a zone of which he has been assigned
Event flow	1. The agronomist opens Dream 2. The agronomist select the button to add a zone from his dashboard 3. The agronomist inspect all the proposed zones and he selects the one to which he has been assigned 4. The agronomist confirms his selection 5. The agronomist receive a summary email
Exit condition	The summary email is confirmed
Exception	The system block the user since the login data are incorrect The agronomist doesn''t find the area he is looking for

	The agronomist doesn't confirm the summary email The agronomist selects the area but he doesn't confirm him selection and closes the browser
Special Requirements	The agronomist must be logged in DREAM

3.2.3.2. Agronomist - AU2 - Use case 2

Name	Agronomist messages handling
Actor	Agronomist, Farmers
Entry condition	The agronomist receive a request message
Event flow	1. The agronomist receive a message from the agronomist 2. The agronomist read the message 3. The agronomist looks for further information a. The agronomist select the option to see weather forecasts b. The agronomist can check the weather forecasts of the farmer's zone c. The agronomist can compare the performances of the farmers who is writing with the performances of the other farmers near him 4. The agronomist write a reply message 5. The agronomist sent the reply
Exit condition	The farmer receive the reply message
Exception	The farmer selects the wrong agronomist who is interested in another zone, so the message arrives to the wrong person, who doesn't reply. The message is written using an unsupported charset, so it is discarded. The user (farmer or agronomist) closes the app without sending the message. The connection drops and the message isn't sent till when the app will be reopened. The agronomist ignores the message from the farmer and doesn't reply to him.
Special requirements	The messages should be encrypted when sent through the internet. The latency between the send action and the receiving should be around 1 second

3.2.3.3. Agronomist - AU3 - Use case 3

Name	Daily plan scheduling
Actor	Agronomist
Entry condition	The agronomist wants to create a new daily plan and is in the dashboard
Event flow	1. The agronomist click on “create new daily plan” 2. The agronomist insert the date to which is referred the daily plan 3. The agronomist check the lists of farms 4. The agronomist scrolls the list, paying attention to the highlighted ones, and to the number of times a farm has already been visited. 5. The agronomist can view more in detail infos of previous notes related to a specific farm 6. The agronomist selects the farms he would visit that day. 7. He clicks on the button to add them to the schedule.
Exit condition	The agronomist saves the daily plan.
Exception	All the farms are well performing and they have already been visited enough time during the year, in this case the agronomist won’t proceed in creating the plan. The agronomist selects a date which is not existing, in this case the plan can’t be created.
Special requirements	The loading of the farms to be displayed should not require more than 3 seconds.

3.2.3.4. Agronomist - AU4 - Use case 4

Name	Confirm daily plan execution
Actor	Agronomist
Entry condition	The agronomist wants to confirm the daily plan execution and is on the dashboard
Event flow	1. The agronomist clicks on the button to confirm the daily plan 2. For each farm visited in the daily plan the agronomist does the following operations: a. The agronomist selects the farm b. The agronomist fill a report form 3. For each farm not visited, due to deviations the agronomist marks the farm as not visited, clicking on a specific button. 4. The agronomist adds, if needed, some notes related to the work done during the day, or the problems he has faced.

Exit condition	The agronomist confirm and clicks on the button to save the report
Exception	There are no farms planned to be visited during the day, in this case the user sees a warning message.
Special requirements	All the needed data should be loaded and displayed in less than 3 seconds.

3.2.3.5. Agronomist - AU5 - Use case 5

Name	Update daily plan - deleting items
Actor	Agronomist
Entry condition	The agronomist wants to update the daily plan, she wants is on the page showing all the plans
Event flow	1. The agronomist select the daily plan and click the update button 2. The agronomist visualizes the list of all the farms he will visit that day 3. The agronomist clicks the trash icon near the farms he wants to remove from the daily plan 4. A message of warning appears 5. The agronomist confirm his choice
Exit condition	The agronomist clicks on save to exit from the daily plan update page
Exception	The agronomist change his mind and doesn't confirm his choice of deleting, in this case nothing append and he stays on the page showing the daily plan The agronomist has selected the wrong day, he can return to the previous page by clicking the apposite button The agronomist doesn't save the entire daily plan before exiting from the update-delete page The date selected is in the past, the user can't delete any items, he can just see the daily plan
Special requirements	The system should respond in less than 3 seconds

3.2.3.6. Agronomist - AU6 - Use case 6

Name	Update daily plan - add farms
Actor	Agronomist
Entry condition	The agronomist wants to update the daily plan, she wants is on the page showing all the plans

Event flow	1. The agronomist visualizes the list of all the farms he will visit that day 2. The agronomist clicks the plus icon on the top of the page 3. A new form appear showing a list of all the farms 4. The agronomist selects from the list the farm he wants 5. The agronomist inserts the hour 6. The agronomist click on the save button 7. A warning message appears 8. The agronomist confirm the message
Exit condition	The agronomist clicks on save to exit from the daily plan update page
Exception	The agronomist has selected the wrong daily plan, he can return to the previous page using the specific button. The agronomist can't find the farm he wants to add, in this case he can close the form and exit. The agronomist doesn't confirm the warning message, in this case nothing happens and he returns to the page showing the daily plan. The date selected is in the past, he can't insert a new farm, he can just see the daily plan without updating it.
Special requirements	

3.2.3.7. Agronomist - AU7 - Use Case 7

Name	Confirmation of new farm in DREAM
Actor	Agronomist, Farmer
Entry condition	The agronomist receive a notification showing a new registered farm in is zone
Event flow	1. The agronomist reads the notification 2. The agronomist opens the DREAM app 3. The agronomist sends and receive messages to agree on the date of the visit 4. The agronomist visit the farm 5. The agronomist assist to the installation of the sensors 6. The agronomist confirm the registration of the farm
Exit condition	The system send a notification to the farmer, that has been confirmed
Exceptions	The farmer doesn't want to be visited, in this case the registration is deleted The farm doesn't exist when the agronomist shows up on the agreed date, in this case the registration is deleted

Special requirements	The system should send the notification to the agronomist as soon as a new farmer sign up

3.2.3.8. Agronomist - AU8 - Use case 8

Name	Create a new steering initiative for the zone
Actor	Agronomist, Farmer
Entry condition	The agronomist wants to create a new initiative and is on the DREAM dashboard
Event Flow	1. The agronomist clicks of the button to insert a new initiative 2. The agronomist fill the form with a description of the initiative 3. The agronomist selects the metrics to track 4. The agronomist selects the farmers to which the initiative is proposed
Exit condition	The agronomist save the initiative
Extension	The agronomist can check the metrics of the farmers if needed during the composition of the initiative The agronomist can set an alert when some of the monitored metrics become higher or lower than a specific parameter
Exception	The agronomist exits without saving, in this case all the changes are lost
Special requirements	The monitored metrics must be updated as soon as they are available

3.2.3.9. Policy maker - PU1 - Use case 1

Name	Check farmers' performances
Actor	Policy maker
Entry condition	The policy maker chooses the option to check farmers' performances from his/her dashboard
Event flow	1. The system shows a list of the farmers and put in evidence those farmers who are performing well and those farmers who are performing badly
Exit condition	The policy maker comes back to the dashboard
Special requirements	The system shall answer in maximum 5 seconds

3.2.3.10. Policy maker - PU2 - Use case 2

Name	Create a new steering initiative
Actor	Policy maker, agronomist, farmers
Entry condition	The policy maker clicks on the button to create a new initiative
Event Flow	1. The policy makers insert a description of the initiative in the apposite text box 2. The policy maker selects the metrics to track 3. The policy maker selects all the farmers to which the initiative is related 4. The policy maker set the expiration date 5. The policy maker saves the initiative 6. The system send a message to all the interested agronomists and farmers
Exit condition	The agronomists receive the message presenting them the initiative to carry out
Extensions	The policy maker, while creating a new initiative, can always consult the metrics of the farmers, in case he needs it.
Exception	The policy maker doesn't save the initiative, in this case the work done is lost.
Special Requirements	The automated messages directed to the Agronomist must be sent by the end of the day

3.2.3.11. Policy maker - PU3 - Use case 3

Name	Check steering initiatives
Actor	Policy maker (or Agronomist)
Entry condition	The policy maker chooses the option to check the steering initiatives from his/her dashboard
Event flow	1. The system shows a list of the steering initiatives and asks to the policy maker to select the steering initiative that he/she wants to check 2. The policy maker selects a steering initiative and clicks on the "check" button 3. The system shows a list of all the farmers who have carried out the steering initiative selected by the policy maker, with the relative results
Exit condition	The policy maker comes back to the dashboard
Special	The system shall answer in maximum 5 seconds

requirements	

3.2.3.12. Policy maker - PU4 - Use case 4

Name	Mark farmers as models
Actors	Policy maker, farmers
Entry condition	The policy maker chooses the option to check farmers' performances from his/her dashboard
Event flow	1. The system shows a list of the farmers and put in evidence those farmers who are performing well and those farmers who are performing badly 2. For each farmer who is performing particularly well the policy maker clicks on the button "Mark as a model" that is placed next to the corresponding farmer's name 3. The system sends a notification to each farmer who has been selected to invite him/her to publish his/her best practices on the forum with the other farmers 4. The system shows a confirmation message to the policy maker
Exit condition	All the farmers have been notified and the policy maker comes back to the dashboard
Exceptions	If a particular farmer is already marked as a model, the corresponding button "Mark as a model" is disabled
Special requirements	The system shall answer in maximum 5 seconds The notification shall arrive to the farmers in maximum 1 minute

3.2.3.13. Policy maker - PU5 - Use case 5

Name	Mark farmers as "to-be-helped"
Actors	Policy maker, farmers
Entry condition	The policy maker chooses the option to check farmers' performances from his/her dashboard
Event flow	1. The system shows a list of the farmers and put in evidence those farmers who are performing well and those farmers who are performing badly 2. For each farmer who is performing particularly badly the policy maker clicks on the button "Mark as "to-be-helped"" that is placed next to the corresponding farmer's name 3. The system sends a notification to each farmer who has been selected to invite him/her to ask for help from agronomists

	4. The system shows a confirmation message to the policy maker
Exit condition	All the farmers have been notified and the policy maker comes back to the dashboard
Exceptions	If a particular farmer is already marked as “to-be-helped”, the corresponding button “Mark as “to-be-helped”” is disabled
Special requirements	The system shall answer in maximum 5 seconds The notification shall arrive to the farmers in maximum 1 minute

3.2.3.14. Farmer - FU1 - Use case 1

Name	Farmer registration
Actors	Farmer
Entry condition	The farmer wants to register to DREAMS
Event flow	1. The farmer selects the sign-in option on the DREAM homepage 2. The farmer inserts his credentials, being the email, his first and last name and his date and place of birth in the correct boxes 3. He confirms his choice 4. DREAM displays a new view used for farm registration 5. The farmer fills the boxes with information regarding his farm such as the location, the soil type and the production type 6. He commits the information by clicking the “confirm” button 7. The systems checks the information given and sends a confirmation mail to the farmer
Exit condition	The summary email is confirmed
Exceptions	Any of the information provided by the farmer may be signed as already used (email, farm location) or incorrect (the information provided is simply not correct) after a system check.
Special requirements	The system check will require some human validation so it will not be an automatic process and will require some time

3.2.3.15. Farmer - FU2 - Use case 2

Name	Asking for help
Actors	Farmers, agronomist
Entry condition	The farmer requires advice and is already logged in
Event flow	1. The farmer opens the help request page from the appropriate button 2. he fills the text box with all the information, details and descriptions regarding his problem 3. he confirms clicking the confirmation button 4. data is sent to all farmers sharing the similar land and weather parameters and to the agronomist appointed to the area of the farm 5. the system sends an email with an information summary regarding his request
Exit condition	The farmer receives the summary email.
Special requirements	The feedback email shall be received in no more than 5 seconds after the request confirmation.

3.2.3.16. Farmer - FU3 - Use case 3

Name	Checking parameters
Actors	Farmer
Entry condition	The farmer is logged in the system
Event flow	1. In the DREAMS dashboard, the farmer clicks on the “Check parameters” option button 2. Through a list of selectable elements, the farmer chooses the parameter he wants to check between weather forecast, soil humidity, water consumption (if available to the logged in farm) 3. The system connects to the database storing the requested information and queries it 4. The system shows on the “Check parameters” based on his choice: a. if “weather forecast” was chosen, the application will show a map of the area relevant to the farm and a brief description based on it b. if “soil humidity” was chosen, the application will show the parameter with a brief description c. if “water consumption” was chosen, the application will show the water consumption of the previous day 5. The farmer visualizes the information for whatever time as he pleases

Exit condition	The farmer clicks on the exit button
Exceptions	A service between the selectable ones may be unavailable due to connection or hardware malfunctioning, in this case a message will be shown explaining the error that occured.
Special requirements	The system shall show the requested information in less than 1 s.

3.2.3.17. Farmer - FU4 - Use case 4

Name	Answering a request
Actors	Farmers
Entry condition	The farmer receives a help request notification
Event flow	1. The farmer clicks on the notification opening a page summarizing and explaining the request problem 2. The farmer carefully reads the information and decides whether to answer the request or not: a. if he decides not to answer, he clicks the close button. The notification will remain available until it is answered by someone. b. if he decides to answer he clicks on the “Answer” button 3. DREAM shows a textbox which can be filled by the farmer to write his solution 4. The farmer writes his solution in the textbox 5. He confirms by clicking the “send” button 6. The system stores the response and sends it to the issuer
Exit condition	An email with a summary of the response is sent to the farmer
Special requirements	The confirmation email is sent in less than 5 minutes after the confirmation.

3.2.3.18. Farmer - FU5 - Use case 5

Name	Visualizing answers to personal requests
Actors	Farmer
Entry condition	The farmer is logged in the system and has issued a request
Event flow	1. On the DREAMS dashboard a notification is shown regarding the previously issued request 2. The farmer gets to the request answers page by clicking on the notification 3. A list of request issued by the user is shown, each of them

	with a marker showing if new replies have arrived 4. The farmer clicks on a list item with a new reply 5. A view with a summary of the original issue is shown by the application, alongside a list of replies to such request and their respective user 6. The farmer reads the replies and based on whether the advice was useful or not he either clicks on the “solved” button or the “not solved” button a. if the “solved” button is clicked, the system closes the request issue and it cannot be replied to anymore b. if the “not solved” button is clicked, the system sends a notification to all the users who replied. The farmer and the other interested users will still be able to reply to the request
Exit condition	The farmer is redirected to the “issued request” page
Special requirements	The “replies” view has to be shown to the user in less than 1 s after its selection.

3.2.3.19. Farmer - FU6 - Use case 6

Name	Consulting the forum
Actors	Farmer
Entry condition	The farmer is logged in the system
Event flow	1. The farmer clicks on the “forum” button 2. The system shows the forum interface, with a list of different sections 3. The farmer clicks on the section of interest 4. A list of topics regarding the chosen section is shown by the app 5. The farmer selects a topic by clicking on it 6. The system shows the first message of the topic on top of the page alongside the original author’s name and a list of replies divided in pages if they exceed a certain amount 7. The farmer reads the messages 8. The farmer decides to reply to the topic by selecting the “reply” button 9. A textbox is displayed, allowing the farmer to write a reply 10. The farmer writes the reply and clicks the “confirm” button 11. The message is uploaded to the topic and is shown as the last reply. 12. A message is issued to all the other users who participated to the conversation If the user consulting the forum is a moderator he may also: a. suspend a user, by selecting the proper button on its personal page

	b. delete a reply, by selecting the proper button after clicking on the reply c. delete a topic, by selecting the proper button after clicking on the topic
Exit condition	The farmer clicks on the exit button
Exceptions	An user may be suspended or banned when trying to write a reply to a given topic. In this case, instead of a textbox, the user will be shown a message saying he has been suspended for a certain amount of time.

3.2.3.20. Farmer - FU7 - Use case 7

Name	Creating a topic in the forum
Actors	Farmer
Entry condition	The farmer is logged in and is in the forum page of DREAMS
Event flow	1. The farmer clicks on the section he wants to create a topic in 2. The farmer clicks on the “create topic” button 3. The farmer inserts a title, a description (which will be shown in the main section page) in the relative box and writes the main argument of the topic he wants to create in the relative textbox 4. He clicks on the “Create” button 5. The system updates the forum section
Exit condition	The forum section is updated with the new topic
Exceptions	In case the user is suspended or banned he will not be able to find the “create topic” button.
Special requirements	The system shall update the section in less than 1 s after the confirmation

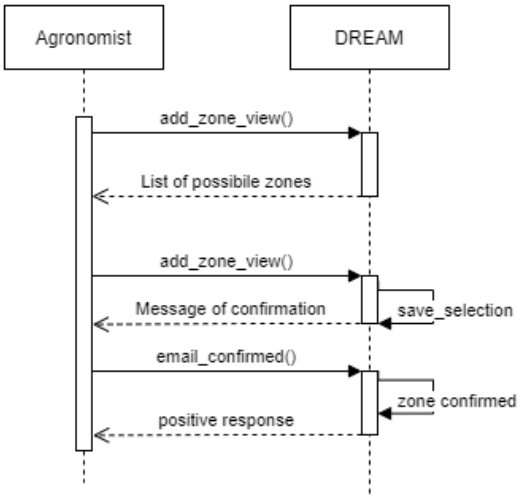
3.2.3.21. Farmer - FU8 - Use case 8

Name	Uploading daily relevant farm parameters
Actors	Farmer

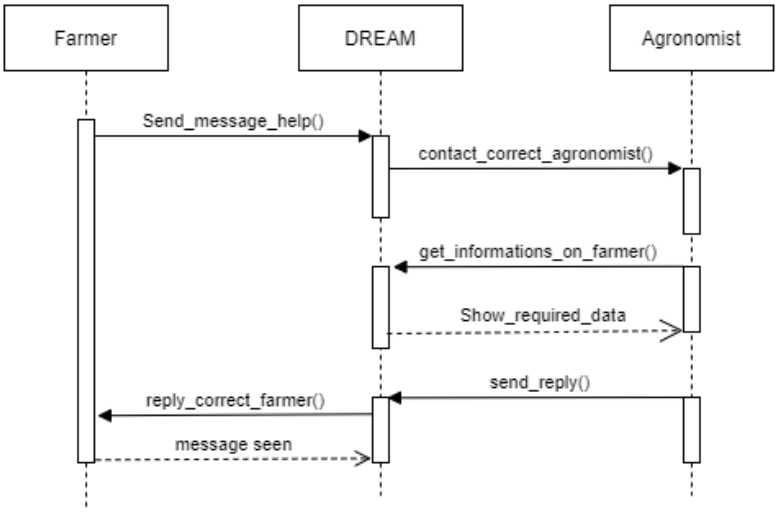
Entry condition	The farmer is logged in the system
Event flow	1. The farmer clicks on the “upload parameters” button 2. He is redirected to a page which presents fixed boxes with fixed labels representing relevant data regarding the production, such as amount of crop harvested and water consumption 3. The farmer inserts the data regarding the daily harvest manually 4. The farmer can automatically fill the field by clicking on the “fill” button a. the system queries the sensors and copies the value in the box 5. The farmer can fill a “notes” box with additional notes regarding the daily production 6. The farmer clicks on the “upload” button 7. The system sends the data to the servers
Exit condition	The farmer is redirected to the dashboard with a confirmation message.
Exceptions	An error message is issued if the same farmer tries to upload data twice in the same day or in an early hour of the day.

3.2.4. Sequence diagrams

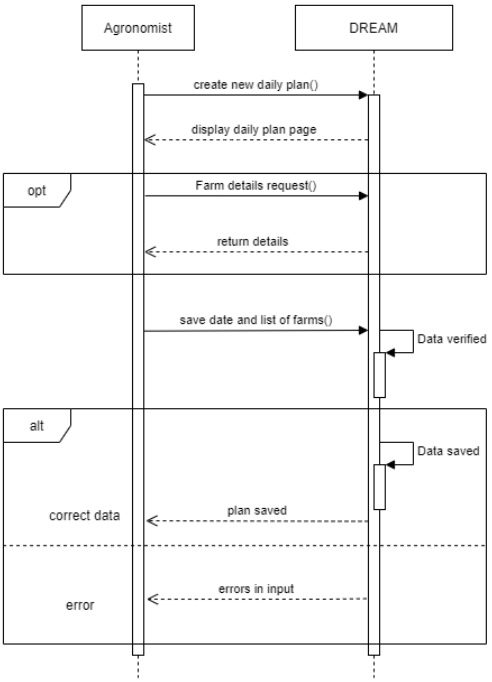
Agronomist - Use case 1



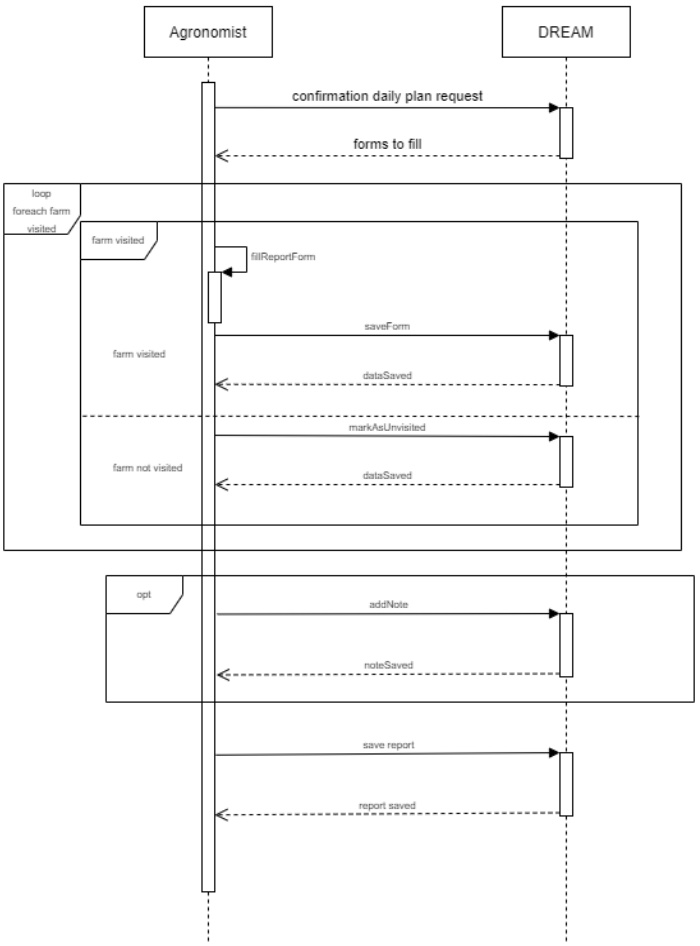
Agronomist - Use case 2



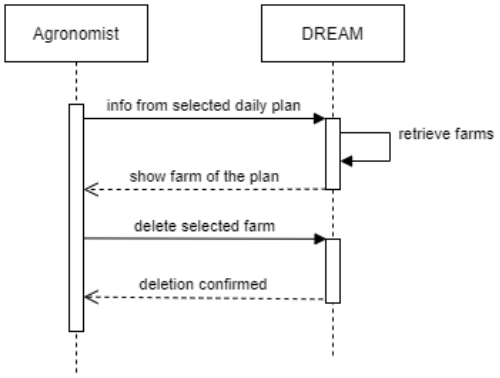
Agronomist - Use case 3



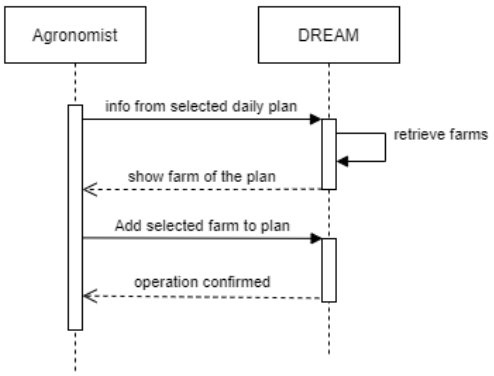
Agronomist - Use case 4



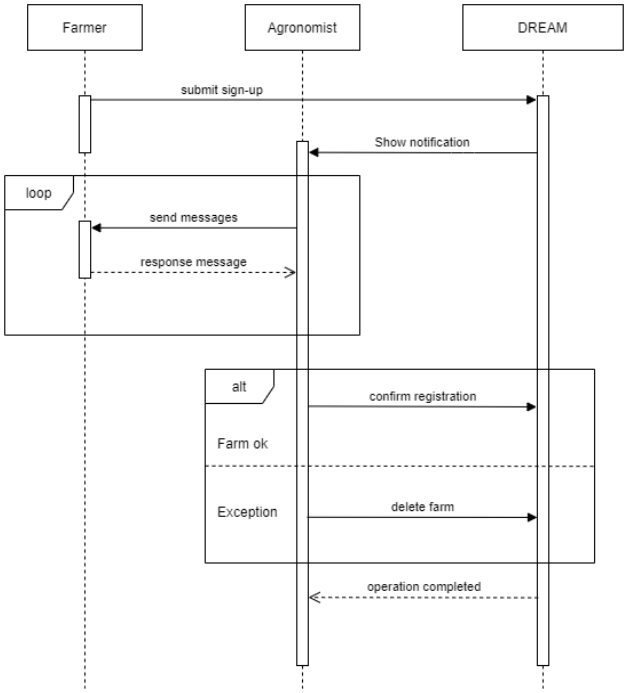
Agronomist - Use case 5



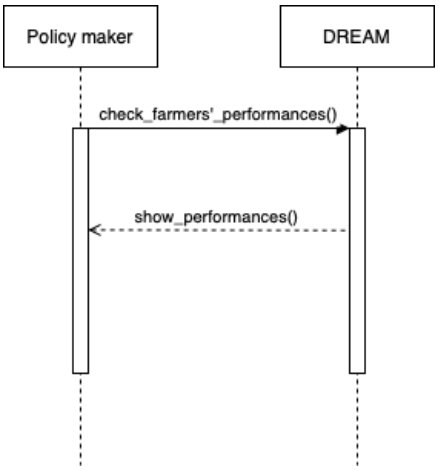
Agronomist - Use case 6



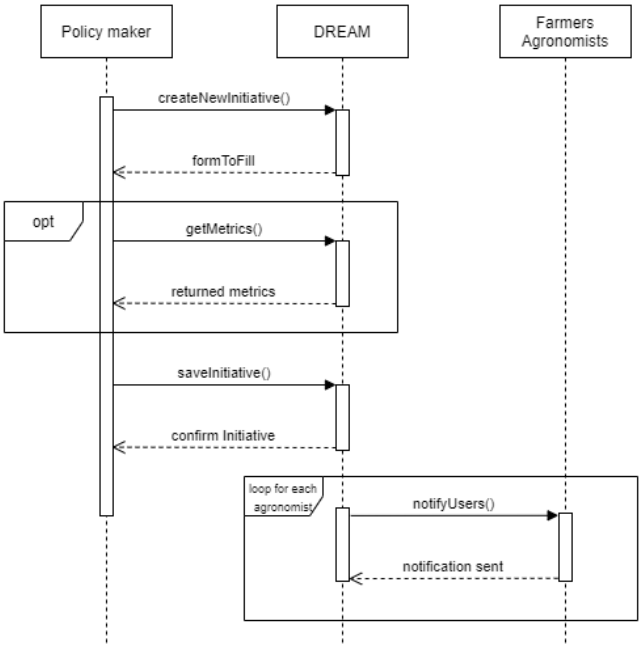
Agronomist - Use case 7



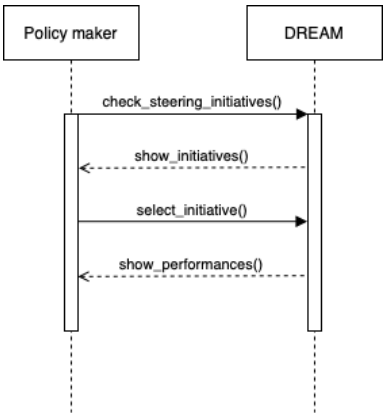
Policy maker - Use case 1



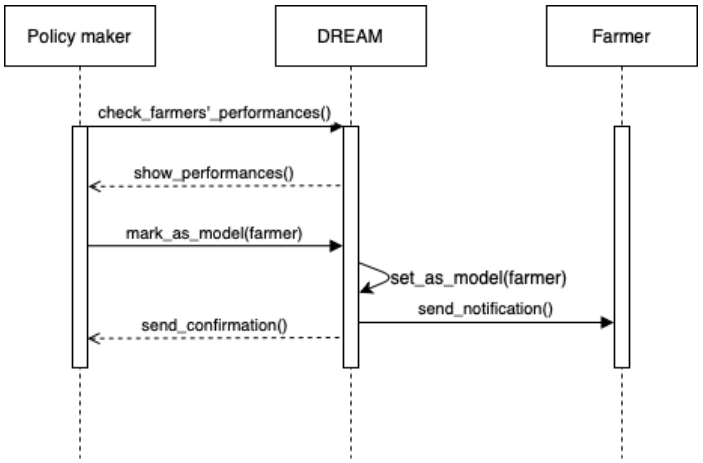
Policy maker - Use case 2



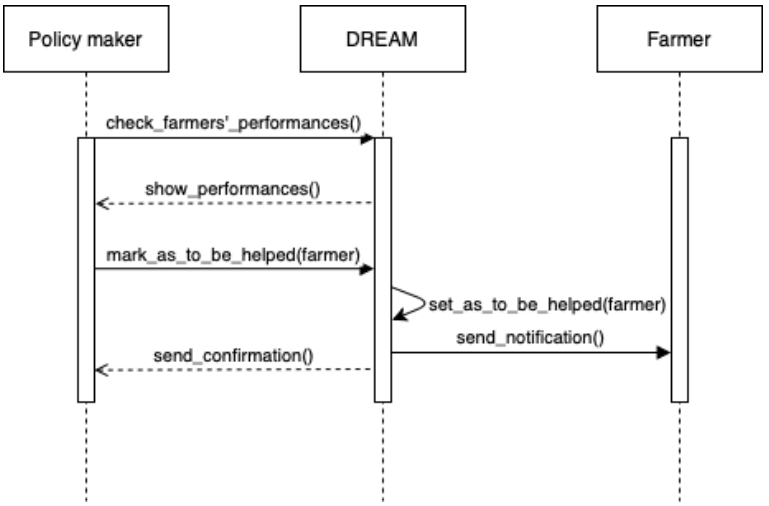
Policy maker - Use case 3



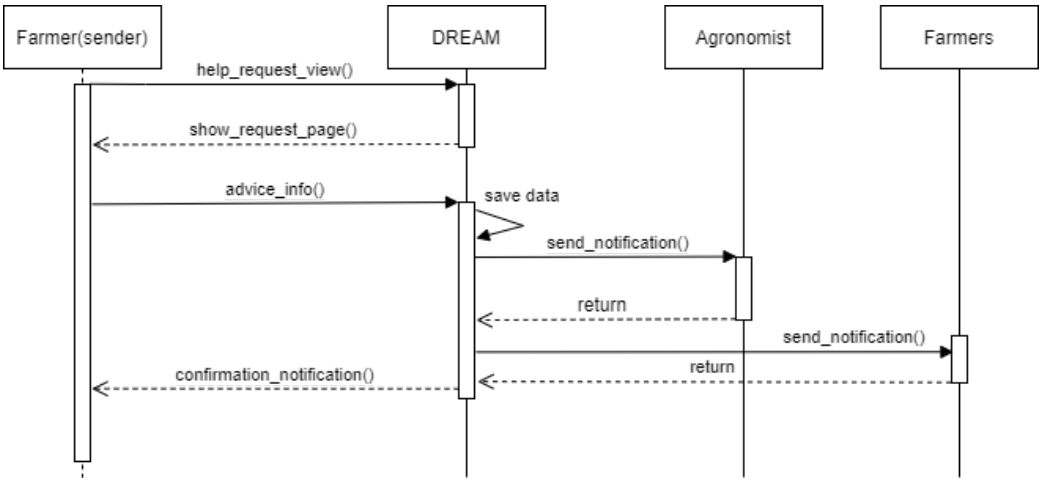
Policy maker - Use case 4



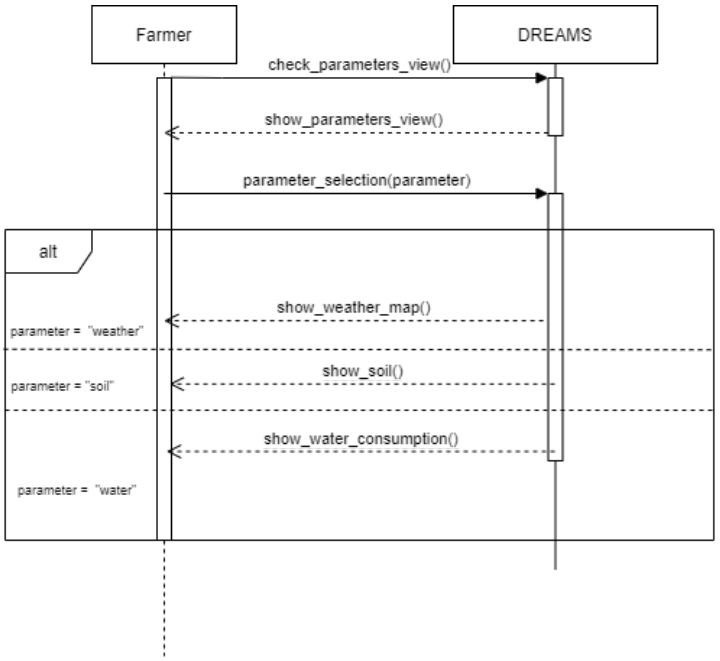
Policy maker - Use case 5



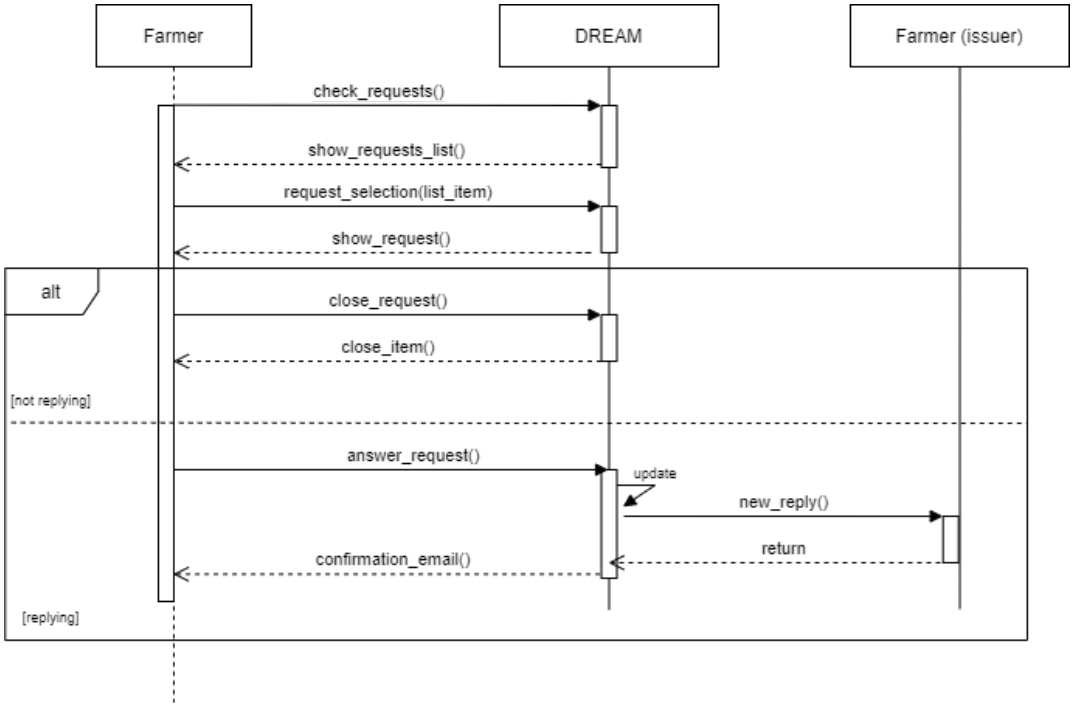
Farmers - Use case 2 (asking for help)



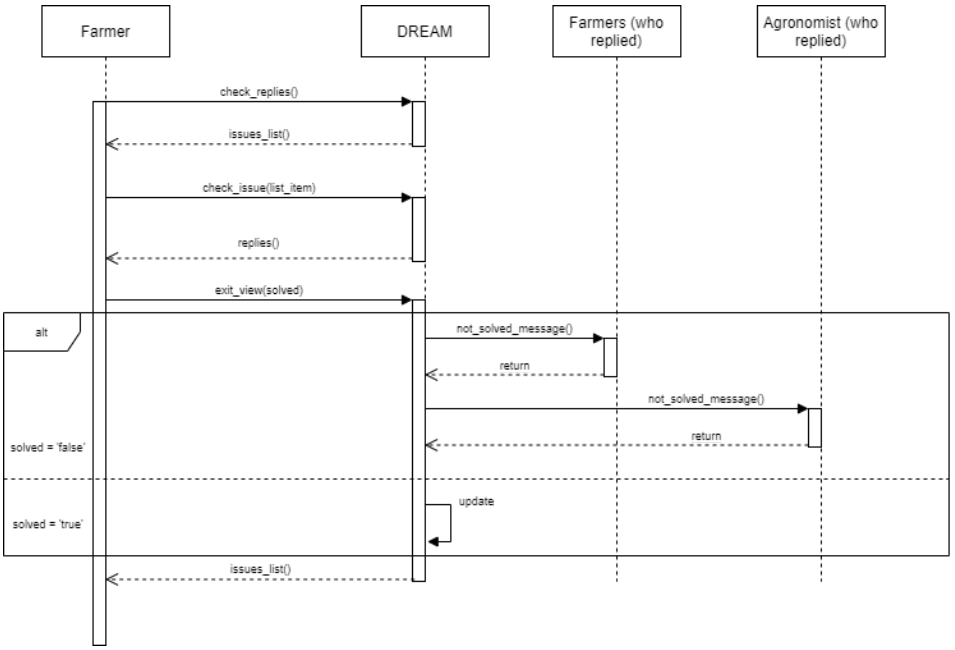
Farmers - Use case 3 (showing parameters)



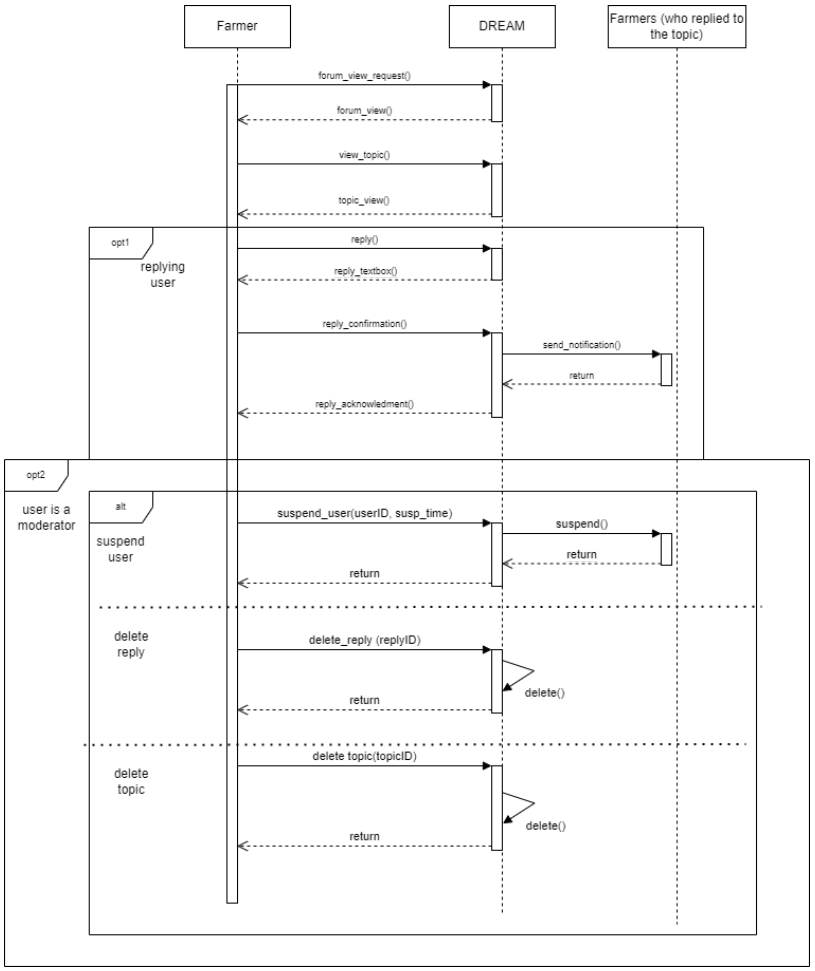
Farmers - Use case 4 (answering help request)



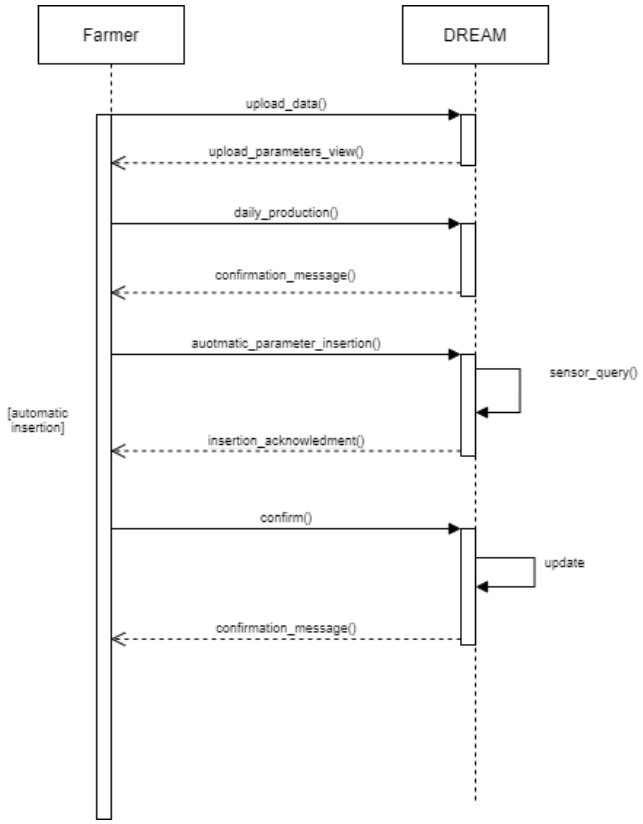
Farmers - Use case 5 (checking help request replies)



Farmers - Use case 6 (consulting the forum)



Farmers - Use case 8 (uploading data)



Sequence diagrams related to FU1 and FU7 are omitted because they are trivial.

3.2.5. Traceability matrices

Agronomist

Goal	Domain assumptions	Requirements
G3	AD1, AD7, AD8	AR1, AR2, AR9 AR14, AR16
G4	AD1, AD3, AD4, AD5, AD6, AD7, AD9, AD10, AD11	AR1, AR2, AR3, AR4, AR7, AR8, AR9, AR10, AR11, AR12, AR13, AR14
G5	AD1, AD3, AD9, AD11	AR1, AR2, AR15
G6	AD1, AD2, AD3	AR1, AR2, AR5, AR6, AR9, AR14, AR15

USE CASE	REQUIREMENTS
AU1	AR1, AR2, AR3
AU2	AR1, AR2, AR5, AR9, AR15
AU3	AR1, AR2, AR4, AR7, AR8, AR9, AR10, AR11, AR14, AR15

AU4	AR1, AR2, AR4, AR8, AR12, AR13, AR14
AU5	AR1, AR2, AR4, AR8, AR10
AU6	AR1, AR2, AR4, AR8, AR11
AU7	AR1, AR2, AR6
AU8	AR1, AR2, AR4, AR9, AR16

Policy makers

Goal	**Domain assumptions**	**Requirements**
G1	PD1	PR1, PR2, PR4, PR6, PR9, PR11, PR12
G2	PD1	PR1, PR2, PR5, PR7, PR10, PR11, PR12
G3	PD1	PR1, PR2, PR3, PR8

USE CASE	**REQUIREMENTS**
PU1	PR4, PR5, PR11, PR12
PU2	PR3
PU3	PR8
PU4	PR4, PR5, PR6, PR9, PR11, PR12
PU5	PR4, PR5, PR7, PR10, PR11, PR12

Farmers

Goal	**Domain assumptions**	**Requirements**
G1,G2	FD1, FD2, FD3, FD4, FD5	FR1, FR2, FR3, FR15, FR16, FR17, FR26
G3	FD6	FR15, FR16,, FR17, FR26
G5	FD1	FR4, FR5, FR6
G6	AD1	FR7, FR8, FR9, FR10, FR11, FR12, FR13, FR14, FR18, FR19, FR20, FR21, FR22, FR23, FR24, FR25, FR26, FR27

USE CASE IDs	REQUIREMENTS IDs
FU1	FR1, FR2, FR3
FU2	FR7, FR8, FR9, FR26
FU4	FR10, FR11, FR26
FU5	FR12, FR13, FR14, FR26
FU3	FR4, FR5 , FR6, FR26
FU6	FR18, FR20, FR21, FR22, FR23, FR24, FR25, FR26
FU7	FR19, FR26
FU8	FR15, FR16, FR17, FR26

3.3. Performance Requirements

The response time of the system is not critical, since the application does not require almost real time replies in order to function. For a pleasant user experience, though, a response time of at most 5 seconds is expected.

Given the high magnitude of users who may use the application simultaneously, the system performance is expected not to change even under a high load of requests.

3.4. Design Constraints

3.4.1. Standard compliance

a. The proposed software should be compliant with the Indian law regarding the personal data treatment, in particular we refer to the law introduced in 2021 named “Personal Data Protection Bill”.
b. The sensors should communicate to the application in a standardized way, in order to keep the application more scalable, allowing the possibility to add new sensors, and as simple as possible (e.g. IEEE 21451-x compliant sensors).
c. For the following implementation the code must be open source and use, if needed, third party libraries which must be open source too, in order to make the application scalable and open to anyone who wants to contribute.

3.4.2. Hardware limitations

The following hardware limitations required to the users in order to use the DREAM application:

- Users devices may have very limited computational power, due to their various nature

- Devices used by users, because of their various nature, may be sensitive to battery consumption
- Internet bandwidth may be limited because of farms different locations and environmental impediments (e.g. very bad weather, mountains)
- Sensors may be subject to corrosion or consumption due to water exposure or environmental agents

3.5. Software System Attributes

3.5.1. Reliability

In order to be as reliable as possible, it is required to have a fault tolerant system.
Data persistence must be ensured taking into account not only the main damage hazards, but also the hostile Indian environment which may very well damage hardware .
Possible errors must not propagate in the system and data loss must be handled properly, so as to restore them.

3.5.2. Availability

An availability of 99.9%, which corresponds to less than 6 hours of downtime per year, should be considered acceptable, since it is the correct trade-off between feasibility and availability constraints in order not to leave farmers without help for a period of time which could be critical to their production.

3.5.3. Security

The system handles different types of data, some of which are personal or sensible data.
In order to guarantee safety for all the actors involved in the application the personal and sensible data stored in the system must be encrypted before saving them with a SHA-3 algorithm.
All messages must be encrypted during transmission using an SSL protocol.

Since we have actors with different responsibilities the system must implement different permissions for each one of them; in this way we can decide the type of user that can access specific parts of the database, and the operations he can do.

3.5.4. Maintainability

The system must be easy to maintain, an important feature is the modularity of the system, that must be scalable, since in the future the application may be expanded and new features could be integrated.

Moreover the code has to be properly commented and the documentation should be as complete as possible.
Sensors placed in the farms should be easy to replace if damaged by the farmers themselves.
The hardware on which DREAM works must be easy to replace in case of fault.

3.5.5. Portability

Due to the various nature of the users, the system has to be available to almost any desktop or mobile device alongside the most used OS run by those systems.
It is also important to guarantee access from all the main web browsers.

4. Formal Analysis using Alloy

4.1. Introduction

In order to check our previous assumptions, and to verify that our goal will actually be achieved, we use a formal language that allows us to inspect in a more structured way our decisions made in the pages above.

4.2. Code

```
open util/integer

abstract sig User{}


sig Agronomist extends User{
	associatedZone: one Zone,
	elaboratedPlan: set DailyPlan,
	checkForecast: set WeatherForecast,
	read: one DataContainer,
}

sig WeatherForecast{
	zone: one Zone,
	forecast: one Weather,
	day: one WeekDay
}

enum Weather {
	Sunny,
	Cloudy,
	Rainy
}
```

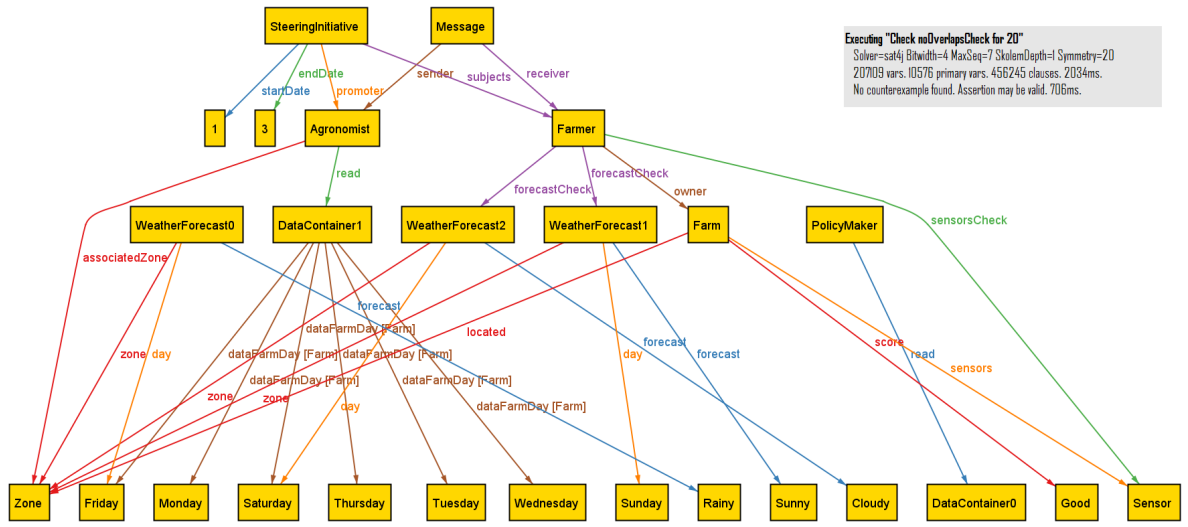
```
sig SteeringInitiative{
	startDate: one Int,
	promoter: one User,
	subjects: some User,
	endDate: one Int
}{startDate > 0  and startDate < endDate}

// Steering initiatives proposed by PolicyMaker involves directly some Agronomists, instead initiatives proposed by Agronomist's involves directly some Farmers of the Agronomist's zone
fact contraintsOnSteeringInitiative{
	all s: SteeringInitiative | (s.promoter in PolicyMaker or s.promoter in Agronomist)
	and (    s.promoter in Agronomist implies (all sj: s.subjects | sj in Farmer and (sj.owner).located = (s.promoter).associatedZone ) )
	and (    s.promoter in PolicyMaker implies (all sj: s.subjects | ( sj in Farmer) ) )
}


// a Farmer can have just one SteeringInitiative associate at time
fact noOverlaps{
	all f: Farmer | no i1, i2: SteeringInitiative | f in i1.subjects and f in i2.subjects and i1 != i2 and ( i1.startDate <= i2.endDate and i2.startDate <= i1.endDate )
}

assert noOverlapsCheck{
	all i:Int | all f: Farmer| no i1, i2: SteeringInitiative |i1 != i2 and f in i1.subjects and f in i2.subjects and i1.startDate <= i and i1.endDate >= i and i2.startDate<= i and i2.endDate >= i
}

// The daily plan is associated to one specific day, that in our alloy is emulated with a week day
sig DailyPlan{
	day: one WeekDay,
	farmsToVisit: some Farm,
	completed: one Bool
}

enum WeekDay {
	Monday,
	Tuesday,
	Wednesday,
	Thursday,
	Friday,
	Saturday,
	Sunday
}
```

```
abstract sig Bool {}
one sig TRUE extends Bool {}
one sig FALSE extends Bool {}

sig Farmer extends User{
	owner: one Farm,
	forecastCheck: set WeatherForecast,
	sensorsCheck: set Sensor,
}

sig PolicyMaker extends User{
	read: one DataContainer,
}


// The performance of a farm is defined, for semplicity, with one of these three options
enum Performance {
	Bad,
	Average,
	Good
}

sig Sensor{
}

// Here are collected all the data inserted by the farmers, that the agronomists and the policy makers can inspect
sig DataContainer{
	dataFarmDay: Farm -> WeekDay
}

sig Farm{
	located: one Zone,
	score: one Performance,
	sensors: some Sensor
}

sig Message{
	sender: one User,
	receiver: one User
}

sig Zone{
}

//The farmer can, obiouvsly, check the values of the sensors that are implemented by the owned farm
fact farmerCheckOnlySensorUsedByHisFarm{
```

```
	all f: Farmer | f.sensorsCheck in (f.owner).sensors
}

//Associate each Agronomist with a different zone
fact oneAgronomistPerZone{
	all a: Agronomist | all a1: Agronomist | a1 != a implies a.associatedZone !=
a1.associatedZone
}

//Each DailyPlan is associated to only one Agronomist
fact maxOneAgronomistPerPlan{
	all a1, a2 : Agronomist |  a1 != a2 implies (no p: DailyPlan | p in a1.elaboratedPlan
and p in a2.elaboratedPlan)
}

//One DailyPlan has exactly one Agronomist who own it
fact minOneAgronomistPerPlan{
	all p: DailyPlan | p in Agronomist.elaboratedPlan
}

//Each farm should have a owner
fact farmOwnedByFarmer{
	all f: Farm | one fm:Farmer | f in fm.owner
}


//The agronomist can visit only the Farms of the same zone
fact farmOfSameAgronomistZone{
	all dp: DailyPlan | (dp.farmsToVisit).located in (elaboratedPlan.dp).associatedZone
}

//In forecast, same zone and same day should correspond to the same forecast
fact oneForecastPerZoneAndDay{
	all w1, w2: WeatherForecast | w1.day = w2.day and w1.zone=w2.zone implies w1 =
w2
}

// Predicate used to confirm the execution of a daily plan
pred confirmDailyPlan[dp: DailyPlan]{
	dp.completed = TRUE
}

// Get all the farmers which are performing good in the zone of the agronomist
fun getBestPerforming[a: Agronomist] :
	set Farmer {
		owner.(score.Good & located.(a.associatedZone))
}
```

```
//check that for each farmer among the ones well performing the score is always Good
assert onlyBestPerforming {
	all f: Farmer | all a: Agronomist | f in getBestPerforming[a] implies (f.owner).score =
Good
}

// Assert to check the predicate confirmDailyPlan
assert confirmPlanAssertion {
	all dp: DailyPlan | confirmDailyPlan[dp] implies dp.completed = TRUE
}


//The messages can't be sent and received by the same user
fact differentSendAndReceiveUsers{
	all m: Message | m.sender != m.receiver
}

//In general all the communications are allowed, except the direct communication from
farmers, that are too many, directed to the Policy makers.
// Moreover the farmer can write only to the agronomist related to him
fact noFramerToPolicyMakers{
	all m: Message |	m.sender in Farmer implies ((m.receiver in Farmer or
m.receiver in Agronomist) and (m.receiver in Agronomist implies m.receiver.associatedZone
= ((m.sender).owner).located) )
}


pred show { #elaboratedPlan.DailyPlan = 1 and #Farm = 1 and #DailyPlan = 2 and
#DataContainer=1 and #dataFarmDay > 1 and #Message = 2 and #SteeringInitiative = 2 }

//check onlyBestPerforming for 40
//check confirmPlanAssertion for 40
run show for 4
//check noOverlapsCheck for 20
```

We used alloy to look for inconsistencies in our model. We decided to represent the main properties and constraints of the system.
When running the model we started considering simple examples, with very few elements, and gradually we increased the number of instances of each signature. We decided to stop when we realized that we had reached a satisfying number of elements, enough to represent the complexity of the system.
We mainly focused on the relations between different elements, for example, regarding steering initiatives and associated farmers, since it was important to have a single initiative associated with a farmer, given a specific date, and we used an assertion to prove the absence of overlapping initiatives related to each farmer.
Other relations were modeled, for example, the one to one mapping between Farm and Farmers, or the association of each zone to an agronomist.



5. Effort Spent

Vittorini Francesco

Date	**Time spent**
Document template, Shared phenomena, assumptions, scenarios	1h 30 min
Use cases	30 min
Use cases, sequence diagrams, state diagrams	1h 45 min
Use case diagram, functional requirements	3h
Non functional requirements	2h
Traceability matrices, document style	3h
Details definition	2h

Luca Venturini

Scenarios	4h
Use cases	3h
Diagrams	3h 45 min
Alloy	4h
Requirements + small fix	1h
General text	2h
Graphic	1h 30 min

Dario Vernola

SP, Goals, scenari 30/11/21	2h 30 min
Domain assumption, use case 1 06/12/21	1h
use cases 07/12/2021	2h 30 min
review of the first 2 chapters, modifications, and scenario updates 13/12/2021	2h
sequence diagrams 14/12/2021	3h
functional req 15/12/2021	1h 20 min

traceability matrix, hardware and software interfaces 16/12/2021	1 h
fixes	1h 30
UML possible fix	1 h

All

First meeting	3h
Revision RASD	6h 30 min
Last revision	2h 30 min

6. References

Agriculture in Telangana
https://www.niti.gov.in/writereaddata/files/Telangana_Presentation_0.pdf

Jackson, M., & Zave, P. (1995). Deriving Specifications from Requirements: an Example.
https://www.fceia.unr.edu.ar/asist/jackson01.pdf

Data meteorological forecasts
https://www.tsdps.telangana.gov.in/aws.jsp

Why India has introduced the new Personal Data Protection Bill
https://www.datacenterdynamics.com/en/opinions/why-india-has-introduced-the-new-personal-data-protection-bill/

Dall'idea al codice con UML 2 - Guida all'utilizzo di UML attraverso esempi
https://www.researchgate.net/publication/344569853_Dall'idea_al_codice_con_UML_2_Guida_all'utilizzo_di_UML_attraverso_esempi

Information technology — Smart transducer interface for sensors and actuators — Part 1: Network Capable Application Processor (NCAP) information model
https://www.iso.org/standard/54357.html

ESA CCI Soil Moisture website
https://www.esa-soilmoisture-cci.org/

Slides course and materials uploaded on WeBeep