DREAM

Data-dRiven PrEdictive FArMing in Telangana

Requirement Analysis and Specification Document - RASD

POLITECNICO
MILANO 1863

Christian Grasso - Filippo Lazzati - Chiara Magri
Year: 2021/2022

Contents

List of Figures

List of Tables

1 Introduction

A RASD is a document that aims to present all the requirements of the system to be developed, explaining the domain in which it has to operate. A RASD should work as baseline for the following tasks in software development, in particular in project planning, software evaluation and change control. Such document has a wide audience, and hence it has to be written as clear as possible.

1.1 Purpose

The main goal that Telangana's government wants to achieve with `DREAM` is to help policy makers formulating policies in the field of agriculture. In order to accomplish this objective, Telangana's government is asking for predictive models for food systems that can drive decisions exploiting huge amount of data. Therefore, `DREAM` must be a software system that gathers data from some sensors and from the farmers and the agronomists and performs data analysis over it in order to help policy makers doing their job. Moreover, `DREAM` aims to help farmers by putting them in contact with each other so that they can exchange advice and aids. Finally, `DREAM` also schedules the daily work of agronomists and their visits to farmers.
Consequently, the goals of this project are:

Table 1: List of goals.

Goals	
G1	The `policy makers` are able to identify the best-performing farmers and the worst-performing farmers[1].
G2	The `policy makers` are able to understand whether the initiatives involving agronomists and best-performing farmers have a good impact on the work of the farmers.
G3	`Farmers` visualize weather forecasts regarding their piece of land.
G4	The `farmers` receive personalized suggestions about crops to plant and fertilizers to use.

[1] See section 1.3

G5	The `farmers` ask for and receive help from agronomists and other farmers.
G6	The `agronomists` can plan farm visits based on the farmers performances.
G7	The `farmers` can interact with other farmers exchanging opinions about agriculture.
G8	The `agronomists` visualize weather forecasts regarding the area they are responsible of.
G9	The `agronomists` are able to identify the best-performing farmers and the worst-performing farmers[1] of the area they are responsible of.

1.2 Scope

`DREAM` is a software system that has to work in a *World*[2] where the following phenomena occur[3]:

Table 2: List of world phenomena.

World phenomena	
WP1	In the mandal X, in the day YYYY/MM/DD, the **weather** is WW[4].
WP2	In the mandal X, the maximun, minimum and average **temperatures** in the day YYYY/MM/DD are TMAX, TMIN and TAVG.
WP3	In the mandal X, the millimeters of **rain** fallen in the day YYYY/MM/DD are RR.

[2]The world is the portion of the real-world affected by the machine. See section 1.3

[3]Here World-only phenomena are listed, that is the World phenomena which are not shared with the Machine. If W is the set of World phenomena and M is the set of Machine phenomena, here elements of the (W - M) set are listed.

[4]WW is a label that, from now on, will be used to denote one of the following attributes: sunny, partially cloudy, cloudy, foggy, rainy, stormy, tornado, hurricane. See section 1.3

WP4	In the mandal X, the average **wind** of the day YYYY/MM/DD has a speed of VV km/h and a direction DR[5]
WP5	In the mandal X, the average **humidity** of the day YYYY/MM/DD rate is HR.
WP6	In the mandal X, the average atmospheric **pressure** of the day YYYY/MM/DD rate is AP millibars.
WP7	The soil moisture[6] on a terrain T is SM.
WP8	The water irrigation system of a terrain TT spreads a quantity WI of water during the day YYYY/MM/DD.
WP9	A farmer works in a plot of land of area A (and/or lives in the corresponding farm) in the mandal X.
WP10	A terrain of area A is cultivated with product P.
WP11	A farmer during the day YYYY/MM/DD plants an area A of product PR.
WP12	A farmer during the day YYYY/MM/DD seeds an area A of product PR.
WP13	A farmer during the day YYYY/MM/DD harvests a quantity of K kgs of product PR.
WP14	A farmer during the day YYYY/MM/DD uses a quantity FQ of fertilizer on a plant P.
WP15	A farmer manually irrigates one of his fields.

[5]DR is a label that, from now on, will be used to denote one of the following directions: N, W, S, E, NW, NE, SE, SW. See section 1.3

[6]The soil moisture is defined as the mass of water/mass of solid particles in the terrain – or volume, depending on the definition

WP16	A farmer faces a problem during his work.
WP17	A policy maker is given a password by the Telangana administration.
WP18	An agronomist is given a password by the Telangana administration.
WP19	An agronomist visits a farm.
WP20	An agronomist is responsible for an area A of terrains.

The shared phenomena[7] are:

Table 3: List of shared phenomena.

Shared phenomena	
SP1	The forecast indicates that in the mandal X, in the day YYYY/MM/DD, the weather will be WW.
SP2	The forecast indicates that in the mandal X, in the day/week/month the maximun, minimum and average temperatures will be TMAX, TMIN and TAVG.
SP3	The forecast indicates that in the mandal X, in the day/week/month the millimeters of rain fallen will be RR.
SP4	The forecast indicates that in the mandal X, in the day/week/month, the average wind will have a speed of VV km/h and a direction DR.
SP5	The forecast indicates that in the mandal X, in the day/week/month, the average humidity rate will be HR.
SP6	The forecast indicates that in the mandal X, in the day/week/month, the average pressure will be AP millibars.
SP7	The weather reports indicate that in the mandal X, in the day YYYY/MM/DD, the weather was WW.
SP8	The weather reports indicate that in the mandal X, in the day/week/month the maximun, minimum and average temperatures in the day YYYY/MM/DD were TMAX, TMIN and TAVG.
SP9	The weather reports indicate that in the mandal X, in the day/week/month the millimeters of rain fallen were RR.

[7]Shared phenomena are the intersection between World phenomena W and Machine phenomena M: W$\cap M$

SP10	The weather reports indicate that in the mandal X, in the day/week/month, the average wind had a speed of VV km/h and a direction DR.
SP11	The weather reports indicate that in the mandal X, in the day/week/month, the average humidity rate was HR.
SP12	The weather reports indicate that in the mandal X, in the day/week/month, the average atmospheric pressure was AP millibars.
SP13	The soil moisture sensor detects a soil moisture on a terrain T of SM%.
SP14	The water irrigation system sensor of a terrain TT measures that a quantity WI of water has been spread during the day YYYY/MM/DD.
SP15	A farmer creates an account.
SP16	A farmer logs in.
SP17	A farmer inserts in `DREAM` his personal data (name, surname, contacts).
SP18	A farmer inserts in `DREAM` the location and the area of the plot of land that he owns/is responsible of.
SP19	A farmer inserts in `DREAM` that in his piece of land, an area A is devoted to the cultivation of product P.
SP20	A farmer inserts in `DREAM` that during the day YYYY/MM/DD he has sowed an area A of product PR.
SP21	A farmer inserts in `DREAM` that during the day YYYY/MM/DD he has planted an area A of product PR.
SP22	A farmer inserts in `DREAM` that during the day YYYY/MM/DD he has harvested a quantity of K kgs of product PR.

SP23	A farmer inserts in DREAM during the day YYYY/MM/DD that he has used a quantity FQ of fertilizer on a plant P.
SP24	A farmer inserts in DREAM that that during the day YYYY/MM/DD he has manually irrigated a piece of land PL with a quantity of water WI.
SP25	A farmer visualizes the weather forecasts relative to its farm's location.
SP26	A farmer visualizes a suggestion regarding a crop to plant or a fertilizer to use in its land.
SP27	A farmer inserts into DREAM data regarding a problem he has or he faced.
SP28	A farmer issues a request for a visit of an agronomist.
SP29	A farmer issues a request for help to an agronomist.
SP30	A farmer issues a request for help to another farmer.
SP31	A farmer responds to a request for help received by another farmer.
SP32	A farmer shares a post on the discussion forum.
SP33	A farmer creates a new discussion thread on the forum.
SP34	A farmer receives the notification of the visit of an agronomist.
SP35	A policy maker logs in.
SP36	DREAM shows statistics regarding farmers (type of products, amount produced, fertilizers, watering, location...).
SP37	DREAM shows a list of the best-performing farmers.

SP38	DREAM shows a list of the worst-performing farmers.
SP39	DREAM shows correlations between agronomists and good farmers interventions and farmers statistics
SP40	A policy maker selects a farmer from a list of best performing farmers and saves it as well performing farmer .
SP41	A policy maker selects a farmer from a list of worst performing farmers and saves it as badly performing farmer .
SP42	An agronomists logs in in the DREAM system.
SP43	An agronomist inserts the area A of terrains he is responsible of into the DREAM system.
SP44	An agronomist receives a request for help from a farmer.
SP45	An agronomist replies to a request for help from a farmer.
SP46	An agronomist visualizes the weather forecast for the area he is responsible of.
SP47	An agronomist visualizes a list of the best-performing farmers in the area he is responsible of.
SP48	DREAM computes a daily plan for an agronomist to visit the farms he is responsible of.
SP49	An agronomist updates his daily plan for visiting farms.
SP50	At the end of a work day, an agronomist confirms the execution of his daily plan for visiting farms.

SP51	At the end of a work day, an agronomist specifies a deviation for his daily plan for visiting farms.
SP52	An agronomist inserts into the system information regarding a visit to a farm.
SP53	An agronomist visualizes data about a farmer he has to visit: the type of plants he grows, the seeded or planted quantities and the harvested quantities for each cultivation starting from the agronomist's last visit to the farmer and the problems inserted by the farmer since the last visit of the agronomist.

1.3 Definitions, Acronyms, Abbreviations

- WW = sunny — partially cloudy — cloudy — foggy — rainy — stormy — tornado — hurricane
- DR = North — West — South — East — North-West — North-East — South-East — South-West
- farmer = citizen of Telangana who works on a piece of land
- agronomist = expert of agriculture in Telangana
- policy-maker = devises policies to regulate the agricultural production of Telangana
- best/worst -performing farmer = farmer chosen by a policy-maker (or an agronomist) taking into account its performances
- world[8] = portion of the real-world affected by the machine.
- TSDPS = Telangana State Development Planning Society
- machine = the portion of system to be developed.
- weather report = historical meteorological data (so, referred to the past)
- weather forecast= predicted meteorological data (so, referred to the future)

[8]Michael Jackson. 1995. The world and the machine. In Proceedings of the 17th international conference on Software engineering (ICSE '95) Association for Computing Machinery, New York, NY, USA, 283–292. DOI:https://doi.org/10.1145/225014.225041

1.4 Revision history

revision	changes
1.0	initial version

1.5 Reference Documents

- *IEEE 29148-2018 Requirements engineering*, the IEEE specification document that "provides details for the construct of well-formed textual requirements, to include characteristics and attributes, in the context of system and software engineering";
- course slides;
- assignment document.

1.6 Document Structure

This document complies with the SRS[9] standard structure as it is defined in the *IEEE 29148-2018 Requirements engineering*, section 9.6. Nevertheless, the order of the contents has been slightly changed in order to facilitate the readers in the reading of this specific RASD[10] and an additional section about Alloy[11] formal analysis has been added. Therefore, the document is divided in 4 main parts:

1. the first part (to which this section belongs) provides an introduction to the system to-be, `DREAM`, making clear which are the goals it is required to achieve and in which context it is going to operate;
2. the second part provides a more detailed description of the functions that `DREAM` has to implement relating them to the main concepts of the system and to the user needs; it also provides the main assumptions under which `DREAM` will work properly;
3. the third part contains the out-and-out requirements of the system, from both the functional and the non-functional points of view;
4. the fourth and last part contains a detailed analysis using a first order logic language (Alloy) of the trickiest requirements of the system.

It should be remarked that the structure of this document does not follow a logic or temporal order, but whoever is interested in the reading can jump from a section to another, because the purpose of it is to be a reference document.

[9] Software Requirements Specification
[10] Requirements Analysis and Specification Document
[11] `https://alloytools.org/`.

2 Overall description

2.1 Product perspective

The requirements-level class diagram of DREAM is:

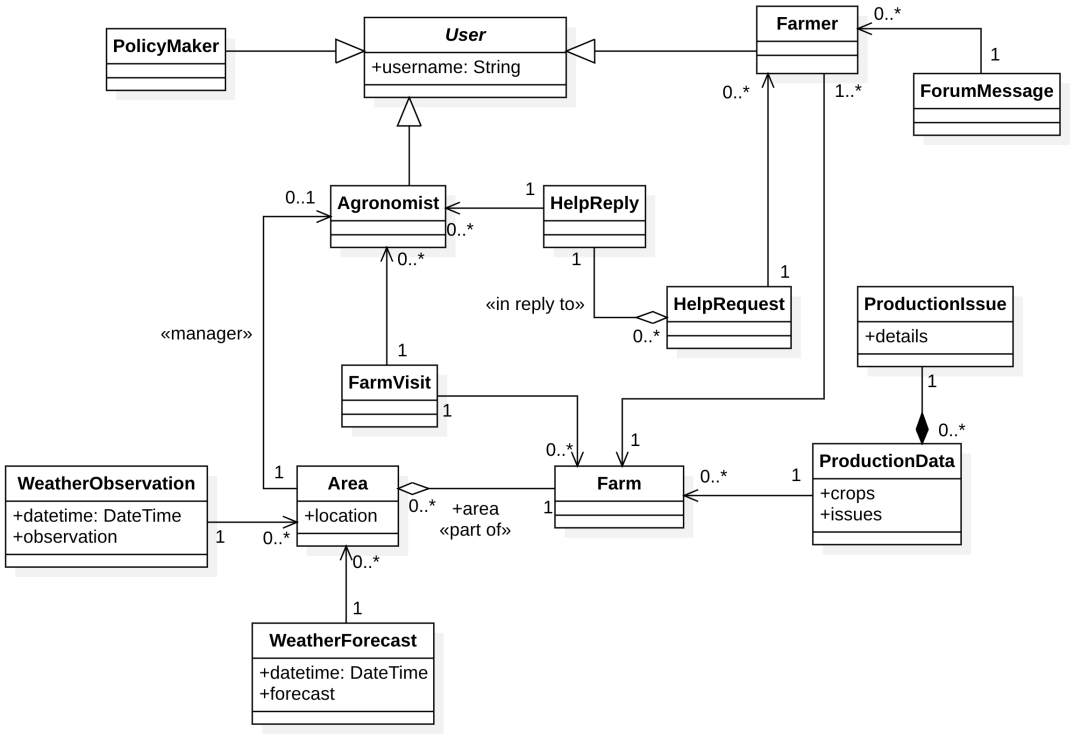

Figure 1: Requirements-level class diagram of DREAM.

The activity diagram for the creation of a new help request is:

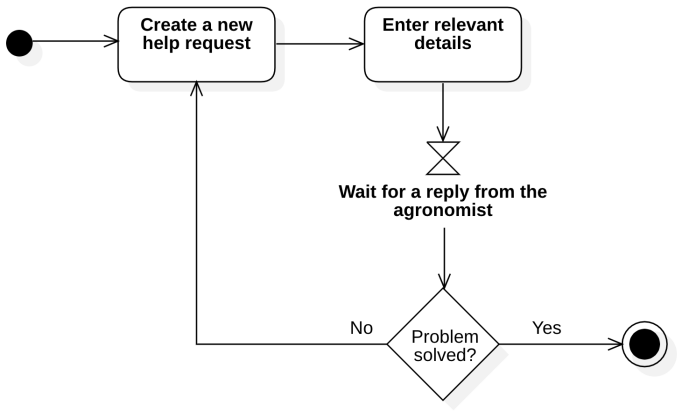

Figure 2: Activity diagram for the creation of a new help request.

The activity diagram for the generation of a new daily plan is:

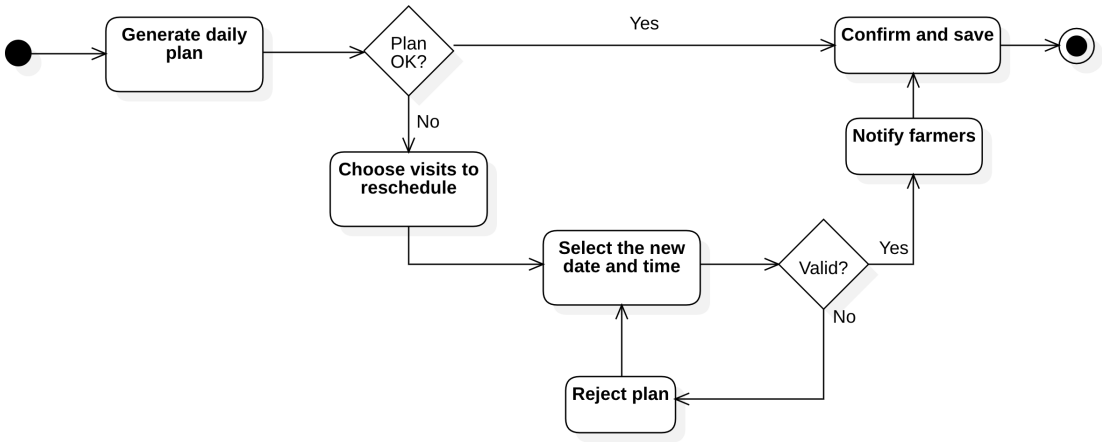

Figure 3: Activity diagram for the generation of a new daily plan.

The activity diagram for the creation of a new production report is:

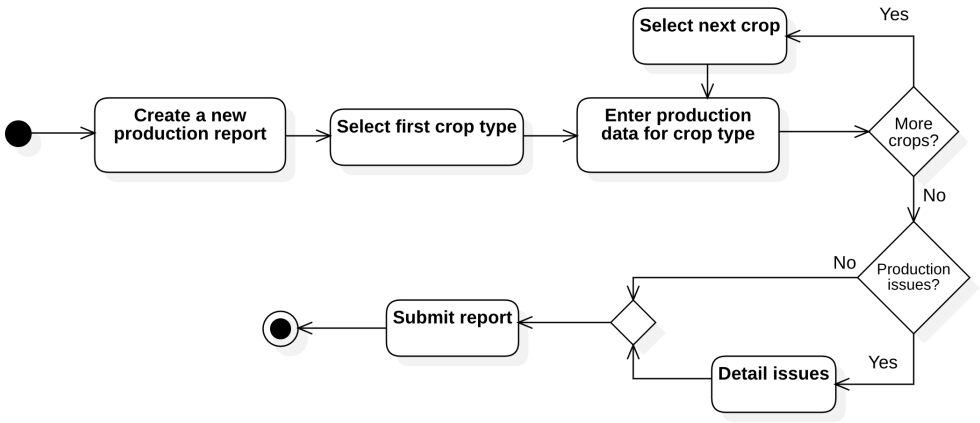

Figure 4: Activity diagram for the creation of a new production report.

2.2 Product functions

In this section, a list of the most important requirements of the system is provided; notice that they are just briefly described, since they will be analysed in-depth in chapter 3.

2.2.1 Data collection

DREAM must be able to manage different kinds of data coming from different sources:

1. agronomists and farmers insert into the system standard kind of information about the farms they respectively visit and own, but also other kinds of information like, for example, the types of products cultivated and the production volume for each product (kinds of information that allow the analytics to profile these users). DREAM also allows them to insert less-structured data, for instance full-text feedback about the inserted data (farmers can insert data about any problem they face);

2. data gathered by sensors deployed on the territory and by the water irrigation system are automatically collected by DREAM through effective interfaces interacting with such systems;

3. already existing systems are integrated with DREAM;

4. the last but not the least, farmers can interact through discussion forums, that are stored by DREAM.

2.2.2 Data analysis

The raw data collected by DREAM must be processed before being delivered to the end-user. Therefore, starting with the big volume of information "ingested", various kinds of analytics are performed in order to provide a more aggregate version of the data:

- the suggestions provided to the farmers are customized according to their information;
- information about farmers' production is used to distinguish among farmers who are performing well and those who are not;
- patterns and relations are identified among the data (for instance between weather forecasts and production volumes);
- analytics are also used for quantifying the improvement (if present) in the crop after having adopted agronomists' suggestions.

2.2.3 Forum

As previously mentioned, farmers are allowed to open discussion forums on DREAM with other farmers. This allows to keep in touch with other people doing the same job and, hence, to ask for advice in case of needing.

2.2.4 Request and supply of help

There are four different ways for a farmer to ask for help: ask for it directly to other farmers, ask in the forum, ask to the agronomists or ask for a visit of an agronomist. It should be remarked that a farmer can contact only agronomists in the same area.

2.2.5 Daily plan

DREAM automatically devises a daily plan for each agronomist consisting in a list of farms that must be visited and provides a possible time schedule for such meetings. Every daily plan must be actively accepted by the agronomist before the involved farmer is notified. Anyway, the agronomist can apply some changes to the schedule, but the system will not approve them if some constraints are violated.

2.3 User characteristics

With regards to the possible actors of DREAM, three different main user classes can be identified:

Policy makers: their job is to devise policies to regulate the agricultural production. DREAM helps them providing aggregate data. Policy makers access the system and then receive high-level information about which

farmers are performing well and which not, and possible reasons to this fact. Furthermore, DREAM helps to understand the impact of agronomists' clues in farmers' productions;

Farmers: they have a farm and some plots of land to cultivate. They access the system and insert data about themselves, helping DREAM to profile them. They insert data about their productions, the area where they work, eventual problems they must face, questions in the forum ... and so on. On the other side, DREAM puts them in touch with someone who can help them and provides data that may be relevant to them;

Agronomists: an agronomist has to find methods for increasing the production and the quality of the harvest. They insert the area they are responsible of and the they receive data about the best performing farmers and weather forecasts. They can answer to farmers' questions and they receive a daily plan about which farmers they should visit.

2.4 Assumptions, dependencies and constraints

Table 4: List of domain assumptions.

Domain assumptions	
D1	The location of the sensors is known to the DREAM system.
D2	The location of the water irrigation systems is known to the DREAM system.
D3	The TSDPS provides weather forecasts and reports which can be accessed by DREAM.
D4	The weather reports provide sufficiently accurate data.
D5	The water irrigation system provides data with a sufficiently accurate precision.
D6	Soil moisture sensors provide data with a sufficiently accurate precision.

D7	Each farmer owns at least one device connected to Internet.
D8	Each farmer has the competences for properly accessing the DREAM system and registering to it.
D9	All the data the farmers insert is correct.
D10	(Best-performing) farmers answer to help requests sent to them.
D11	When a farmer does something to a certain area (for example, applies a fertilizer or harvests), he covers the whole area.
D12	Each policy maker owns a personal computer, connected to the Internet network and with a browser installed.
D13	Each policy maker has the competences for properly accessing the DREAM system and using it.
D14	Every policy maker is given a password – associated to his e-mail – by the Telangana agriculture department, and such account is known to DREAM.
D15	Each agronomist owns a device connected to the Internet network and with a browser installed.
D16	The agronomists are given a password – associated to their mail – by the Telangana agriculture department and such account is known to DREAM.
D17	Each agronomist has the competences for properly accessing the DREAM system and using it.
D18	All the data the agronomists insert is correct.
D19	Agronomists reply to help requests issued by farmers.

3 Specific requirements

3.1 External Interface Requirements

3.1.1 User interfaces

DREAM is provided to the users, namely farmers, agronomists and policy makers, as a webapp, accessible through a browser on the web. Therefore, DREAM is not given with a CLI[12] but only with a GUI[13]. The rest of this subsection contains some wireframes that show an example of suitable user interface for DREAM[14].

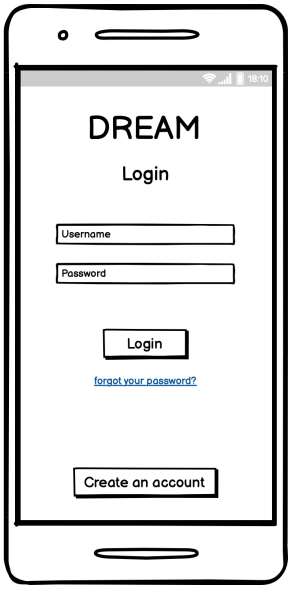

Figure 5: Wireframe of the initial page of the system.

[12] Command Line Interface
[13] Graphical User Interface
[14] According to the "IEEE 29148-2018 Requirements engineering" document, section 9.6.4.2, "A style guide for the user interface can provide consistent rules for organization, coding and interaction of the user with the system".

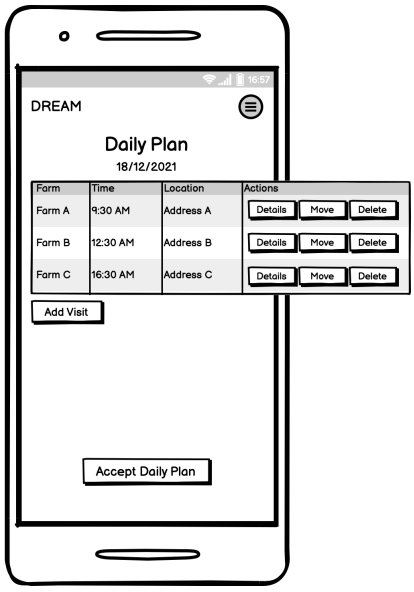

Figure 6: Wireframe of the page that allows to manage the daily plan.

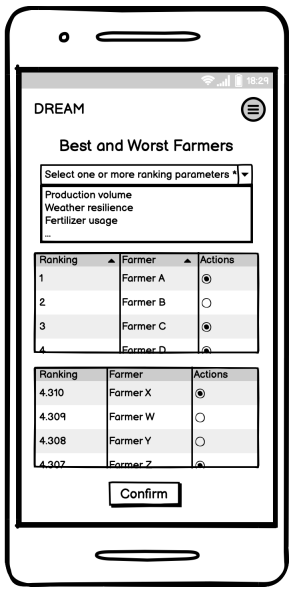

Figure 7: Wireframe of the page that allows to rank the farmers.

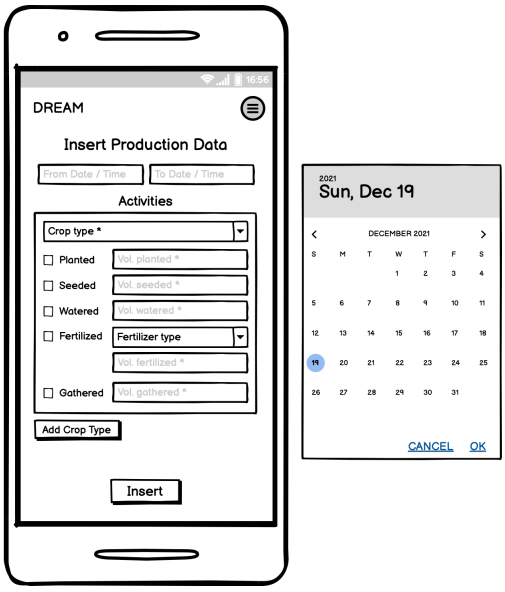

Figure 8: Wireframe of the page that allows the insertion of production data.

3.1.2 Hardware interfaces

- Because DREAM is to be implemented as a webapp, every user can access it through the device he prefers, that is personal computers, smartphones, tablets ... and there is no requirement on the size of the device since the webapp has to be responsive (scale properly to different devices' sizes). Every device of this kind suffices to achieve the goals.
- DREAM has to interact with various hardware components to gather the data it will use. In particular, DREAM has to interact with soil sensors and a water irrigation system. Both these two systems are able to gather data and share it through a third already existing system (an IoT hub), to which they are connected through cables.

3.1.3 Software interfaces

The following software interfaces are required to make DREAM work properly:

1. every user's device must have a browser installed on it through which the user can access the DREAM's website; no other software requirements are requested to these kind of devices;
2. DREAM requires the APIs to the Telangana's government website[15] in order to retrieve weather reports and meteorological short-term and long-term forecasts;

[15] https://www.tsdps.telangana.gov.in/aws.jsp.

3. DREAM requires also an interface to the system that provides access to the data gathered by soil sensors and by the water irrigation system.

3.1.4 Communication interfaces

As far as the communication interfaces are concerned, DREAM uses:

1. the HTTP protocol at the application layer (layer 7 of the stack) to exchange information (web pages) with the clients (users' devices) and to access the data offered by the Telangana's government systems;
2. specific protocols for the communication to the sensor hub.

3.2 Functional Requirements

DREAM allows its users to perform many tasks, and DREAM itself is able to interact with other different systems to achieve some results (see section 2.2). In order to provide the main requirements of the system and a summary of the possible situations in which DREAM is involved and used, this paragraph first lists all the requirements of the system, then it lists some concrete scenarios [16] and finally it abstracts from details and specificities showing the corresponding use cases.

3.2.1 Requirements

Table 5: List of requirements.

Requirements	
R1	DREAM collects soil moisture data from the soil moisture sensors;
R2	DREAM accesses weather reports data provided by the TSDPS system;
R3	DREAM collects watering data from the water irrigation system;
R4	DREAM shall allow agronomists to insert feedback about the farmers they visited;

[16] "A narrative description of what people do and experience as they try to make use of computer systems and applications" [M. Carrol, Scenario-based Design, Wiley, 1995]

R5	DREAM shall allow farmers to insert the location and the area of their plot of land;
R6	DREAM shall allow farmers to insert the type of products they grow in a certain area of their farms;
R7	DREAM shall allow farmers to insert data about their sowing activity;
R8	DREAM shall allow farmers to insert data about their planting activity;
R9	DREAM shall allow farmers to insert data about their harvesting activity;
R10	DREAM shall allow farmers to insert data about the irrigation of their plantations;
R11	DREAM shall allow farmers to insert descriptions of problems they face;
R12	DREAM shall allow farmers to request a visit of an agronomist when inserting a problem;
R13	When a farmer requests a visit of an agronomist, DREAM inserts the visit to the farmer into the daily plan of an agronomist assigned to the area of the farmer;
R14	When a farmer requests a visit of an agronomist, if no agronomist is available in the area of the farmer during the seven days after the request (excluding the day when the request is made), DREAM notifies the farmer about this situation.
R15	DREAM can use different criteria (namely, productivity in terms of harvested quantity over sowed quantity, productivity in presence of adverse meteorological events, productivity in presence of drought, and so on) and/or combinations of them to rank farmers;

R16	DREAM shall allow policy makers and agronomists to choose the criteria to use for ranking farmers and the associated weights (for combinations of them);
R17	DREAM shall allow policy makers and agronomists to choose the time period and the area to consider for ranking the farmers;
R18	DREAM allows policy makers and agronomists to mark/unmark a farmer as best-performing or worst-performing;
R19	DREAM can compare different time periods of a farmer work according to various production criteria (e.g. production volume, fertilizers adopted, fraction of harvested plants over sowed ones, etc);
R20	DREAM can compare different time periods of a farmer work with respect to environmental factors (e.g. weather reports data, soil moisture, ...);
R21	DREAM allows policy makers to choose an initiative taken by an agronomist or a farmer to help a farmer - i.e., visit to the farm or reply to a question - and two time periods of the farmer to compare the two time periods;
R22	DREAM can show the impact of a certain initiative taken by an agronomist or a farmer to help a farmer - i.e., visit to the farm or reply to a question - during a certain time period.
R23	DREAM is able to find correlations among environmental factors, fertilisers adopted and crops planted with the volume of production of the farmers;
R24	DREAM is able to send suggestions to farmers about fertilizers to use;
R25	DREAM is able to send suggestions to farmers about crops to plant;
R26	DREAM allows farmers to choose the date of the forecasts to visualize;

R27	DREAM is able to connect to the Telangana government website to fetch forecasts for the chosen date;
R28	DREAM can show weather forecast data for a certain location and date.
R29	DREAM shall allow agronomists to insert the area they are responsible of;
R30	DREAM shall allow agronomists to choose the date of the weather forecasts to visualize;
R31	DREAM allows farmers to choose the (best-performing) farmer or agronomist (assigned to his area) who to issue a request;
R32	DREAM allows the farmers to send a request to a best-performing farmer or agronomist;
R33	DREAM notifies (best-performing) farmers and agronomists of the requests of help from other farmers;
R34	DREAM allows best-performing farmers and agronomists to insert a message of response to the farmers who have made a request for help to them;
R35	DREAM sends the response to the farmer who issued the corresponding request;
R36	DREAM allows farmers to initiate a new thread in the forum;
R37	DREAM allows the farmers to add a post to a thread in the forum;
R38	DREAM is able to generate a daily plan of visits for each agronomist evenly distributing (with exceptions to bad-performing farmers) over the year the number of visits to each farmer;

R39	DREAM enforces that every farmer is assigned a visit in the daily plan of an agronomist at least twice a year;
R40	DREAM enforces that no agronomist in a certain area is assigned more than twice of the visits than an agronomist of the same area;
R41	DREAM enforces that the daily plan of agronomists includes more visits to worst-performing farmers than to other ones;
R42	DREAM allows agronomists to accept or modify -i.e.: remove, add or move a visit - the automatically generated daily plan;
R43	DREAM allows agronomists to confirm a daily plan at the end of the working day;
R44	DREAM allows agronomists to specify deviations -i.e.: remove, add or move a visit- from a daily plan at the end of the working day;
R45	DREAM is able to notify farmers involved by a visit of an agronomist;
R46	DREAM allows agronomists to visualize data about farmers to visit - such as name, surname and location of the farm - and their production since their last visit to the farm;
R47	DREAM allows agronomists to visualize the problems inserted by a farmer to be visited since their last visit to the farm;
R48	DREAM shall allow users to log in the system by using an email and a password;
R49	DREAM shall allow farmers to visualize data inserted by them about their production.

3.2.2 Mapping $R \wedge D \vDash G$

This section is about showing that the previously listed requirements and domain assumptions can be used to satisfy the goals. In table 6 you can see a summary of the implications, while in the rest of the section you can find some explanations of such entailments.

Table 6: Traceability matrix of requirements and domain assumptions to goals.

$R \wedge D \vDash G$									
	G1	G2	G3	G4	G5	G6	G7	G8	G9
D1	X	X		X					X
D2	X	X		X					X
D3	X	X	X	X				X	X
D4	X	X		X					X
D5	X	X		X					X
D6	X	X		X					X
D7	X	X	X	X	X	X	X		X
D8	X	X	X	X	X	X	X		X
D9	X	X	X	X	X	X			X
D10					X				
D11	X	X		X					X

	G1	G2	G3	G4	G5	G6	G7	G8	G9
D12	X	X							
D13	X	X							
D14	X	X							
D15	X	X			X	X		X	X
D16	X	X			X	X		X	X
D17	X	X			X	X		X	X
D18	X	X			X	X		X	X
D19					X				
R1	X	X		X					X
R2	X	X		X					X
R3	X	X		X					X
R4	X	X							X
R5	X	X	X	X	X	X			X
R6	X	X		X					X
R7	X	X		X					X
R8	X	X		X					X

	G1	G2	G3	G4	G5	G6	G7	G8	G9
R9	X	X		X					X
R10	X	X		X					X
R11	X	X			X	X			X
R12					X	X			
R13					X	X			
R14					X	X			
R15	X								X
R16	X								X
R17	X								X
R18	X				X	X			X
R19		X							
R20		X							
R21		X							
R22		X							
R23				X					
R24				X					

	G1	G2	G3	G4	G5	G6	G7	G8	G9
R25				X					
R26			X						
R27			X	X				X	
R28			X					X	
R29					X	X		X	X
R30								X	
R31					X				
R32					X				
R33					X				
R34					X				
R35					X				
R36							X		
R37							X		
R38						X			
R39						X			
R40						X			

	G1	G2	G3	G4	G5	G6	G7	G8	G9
R41						X			
R42						X			
R43						X			
R44						X			
R45						X			
R46						X			
R47						X			
R48[1]	X	X	X	X	X	X	X	X	X
R49[1]									

Now an explanation of why a certain goal is entailed by certain requirements and certain domain assumptions is provided.

G1: The policy makers are able to identify the best-performing farmers and the worst-performing farmers.

- R1: `DREAM` collects soil moisture data from the soil moisture sensors;
- R2: `DREAM` accesses weather reports data provided by the TSDPS system;
- R3: `DREAM` collects watering data from the water irrigation system;

[17] This requirement is mapped with every goal, as in order to allow policy makers, agronomists and farmers to interact with the system, logging in is necessary. This detail is omitted for brevity in the following explanations of the mapping between goals and requirements.

[18] This requirement is not mapped with any goal, thus to be rigorous it should be removed from here; however, it describes a useful functionality that `DREAM| shall provide, therefore we maintain it.`

- R4: DREAM shall allow agronomists to insert feedback about the farmers they visited;
- R5: DREAM shall allow farmers to insert the location and the area of their plot of land;
- R6: DREAM shall allow farmers to insert the type of products they grow in a certain area of their farms;
- R7: DREAM shall allow farmers to insert data about their sowing activity;
- R8: DREAM shall allow farmers to insert data about their planting activity;
- R9: DREAM shall allow farmers to insert data about their harvesting activity;
- R10: DREAM shall allow farmers to insert data about the irrigation of their plantations;
- R11: DREAM shall allow farmers to insert descriptions of problems they face;
- R15: DREAM can use different criteria (namely, productivity in terms of harvested quantity over sowed quantity, productivity in presence of adverse meteorological events, productivity in presence of drought, and so on) and/or combinations of them to rank farmers;
- R16: DREAM shall allow policy makers and agronomists to choose the criteria to use for ranking farmers and the associated weights (for combinations of them);
- R17: DREAM shall allow policy makers and agronomists to choose the time period and the area to consider for ranking the farmers;
- R18: DREAM allows policy makers and agronomists to mark/unmark a farmer as best-performing or worst-performing;
- R48: DREAM shall allow users to log in the system by using an email and a password;
- D1: The location of the sensors is known to the DREAM system;
- D2: The location of the water irrigation systems is known to the DREAM system;
- D3: The TSDPS provides weather forecasts and reports which can be accessed by DREAM;
- D4: The weather reports provide sufficiently accurate data;
- D5: The water irrigation system provides data with a sufficiently accurate precision;

- D6: Soil moisture sensors provide data with a sufficiently accurate precision;
- D7: Each farmer owns at least one device connected to Internet;
- D8: Each farmer has the competences for properly accessing the DREAM system and registering to it;
- D9: All the data the farmers insert is correct;
- D11: When a farmer does something to a certain area (for example, applies a fertilizer or harvests), he covers the whole area;
- D12: Each policy maker owns a personal computer, connected to the Internet network and with a browser installed;
- D13: Each policy maker has the competences for properly accessing the DREAM system and using it;
- D14: Every policy maker is given a password – associated to his e-mail – by the Telangana agriculture department, and such account is known to DREAM.
- D15: Each agronomist owns a device connected to the Internet network and with a browser installed.
- D16: The agronomists are given a password – associated to their mail – by the Telangana agriculture department and such account is known to DREAM.
- D17: Each agronomist has the competences for properly accessing the DREAM system and using it.
- D18: All the data the agronomists insert is correct.

In order to achieve this goal, DREAM must be able to collect all the data that may be related to the performances of farmers (R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11); moreover, it must be able to order farmers according to specified criteria - possibly restricting the scope of ordering spatially or temporally - (R15, R16, R17). Finally, policy makers and agronomists are able to mark/unmark farmers as best- or worst- performing (R18). Data used by DREAM to rate the farmers should be correct.

G2: The policy makers are able to understand whether the initiatives involving agronomists and best-performing farmers have a good impact on the work of the farmers.

- R1: DREAM collects soil moisture data from the soil moisture sensors;
- R2: DREAM accesses weather reports data provided by the TSDPS system;

- R3: DREAM collects watering data from the water irrigation system;
- R4: DREAM shall allow agronomists to insert feedback about the farmers they visited;
- R5: DREAM shall allow farmers to insert the location and the area of their plot of land;
- R6: DREAM shall allow farmers to insert the type of products they grow in a certain area of their farms;
- R7: DREAM shall allow farmers to insert data about their sowing activity;
- R8: DREAM shall allow farmers to insert data about their planting activity;
- R9: DREAM shall allow farmers to insert data about their harvesting activity;
- R10: DREAM shall allow farmers to insert data about the irrigation of their plantations;
- R11: DREAM shall allow farmers to insert descriptions of problems they face;
- R19: DREAM can compare different time periods of a farmer work according to various production criteria (e.g. production volume, fertilizers adopted, fraction of harvested plants over sowed ones, etc);
- R20: DREAM can compare different time periods of a farmer work with respect to environmental factors (e.g. weather reports data, soil moisture, ...);
- R21: DREAM allows policy makers to choose an initiative taken by an agronomist or a farmer to help a farmer - i.e., visit to the farm or reply to a question - and two time periods of the farmer to compare the two time periods;
- R22: DREAM can show the impact of a certain initiative taken by an agronomist or a farmer to help a farmer - i.e., visit to the farm or reply to a question - during a certain time period;
- R48: DREAM shall allow users to log in the system by using an email and a password;
- D1: The location of the sensors is known to the DREAM system;
- D2: The location of the water irrigation systems is known to the DREAM system;
- D3: The TSDPS provides weather forecasts and reports which can be accessed by DREAM;

- D4: The weather reports provide sufficiently accurate data;
- D5: The water irrigation system provides data with a sufficiently accurate precision;
- D6: Soil moisture sensors provide data with a sufficiently accurate precision;
- D7: Each farmer owns at least one device connected to Internet;
- D8: Each farmer has the competences for properly accessing the `DREAM` system and registering to it;
- D9: All the data the farmers insert is correct;
- D11: When a farmer does something to a certain area (for example, applies a fertilizer or harvests), he covers the whole area;
- D12: Each policy maker owns a personal computer, connected to the Internet network and with a browser installed;
- D13: Each policy maker has the competences for properly accessing the `DREAM` system and using it;
- D14: Every policy maker is given a password – associated to his e-mail – by the Telangana agriculture department, and such account is known to `DREAM`.
- D15: Each agronomist owns a device connected to the Internet network and with a browser installed.
- D16: The agronomists are given a password – associated to their mail – by the Telangana agriculture department and such account is known to `DREAM`.
- D17: Each agronomist has the competences for properly accessing the `DREAM` system and using it.
- D18: All the data the agronomists insert is correct.

Goal G2 is achieved `DREAM` collecting all the data that may be involved in correlations between initiatives and impact on work (R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11); furthermore, `DREAM` must be able to compare different periods of farmers work (R19, R20) according to the choice of the policy maker (R21) and to show the results (R22). Data used by `DREAM` regarding the farmers' work, the agronomists' interventions and the natural phenomena influencing agriculture (weather, soil moisture, water irrigation) should be correct for the goal to be achieved.

G3 Farmers visualize weather forecasts regarding their piece of land.

- R5: DREAM shall allow farmers to insert the location and the area of their plot of land;
- R26: DREAM allows farmers to choose the date of the forecasts to visualize;
- R27: DREAM is able to connect to the Telangana government website to fetch forecasts for the chosen date;
- R28: DREAM can show weather forecast data for a certain location and date.
- R48: DREAM shall allow users to log in the system by using an email and a password;
- D3: The TSDPS provides weather forecasts and reports which can be accessed by DREAM.
- D7: Each farmer owns at least one device connected to Internet;
- D8: Each farmer has the competences for properly accessing the DREAM system and registering to it;
- D9: All the data the farmers insert is correct;

In order to achieve this goal, DREAM must fetch the forecasts (R27) according to the area of the farmer (R5) and the selected date (R26) and it must be able to show the results (R28).

G4 The farmers receive personalized suggestions about crops to plant and fertilizers to use.

- R1: DREAM collects soil moisture data from the soil moisture sensors;
- R2: DREAM accesses weather reports data provided by the TSDPS system;
- R3: DREAM collects watering data from the water irrigation system;
- R5: DREAM shall allow farmers to insert the location and the area of their plot of land;
- R6: DREAM shall allow farmers to insert the type of products they grow in a certain area of their farms;
- R7: DREAM shall allow farmers to insert data about their sowing activity;
- R8: DREAM shall allow farmers to insert data about their planting activity;
- R9: DREAM shall allow farmers to insert data about their harvesting activity;
- R10: DREAM shall allow farmers to insert data about the irrigation of their plantations;

- R23: DREAM is able to find correlations among environmental factors, fertilisers adopted and crops planted with the volume of production of the farmers;
- R24: DREAM is able to send suggestions to farmers about fertilizers to use;
- R25: DREAM is able to send suggestions to farmers about crops to plant;
- R27: DREAM is able to connect to the Telangana government website to fetch forecasts for the chosen date;
- R48: DREAM shall allow users to log in the system by using an email and a password;
- D1: The location of the sensors is known to the DREAM system;
- D2: The location of the water irrigation systems is known to the DREAM system;
- D3: The TSDPS provides weather forecasts and reports which can be accessed by DREAM;
- D4: The weather reports provide sufficiently accurate data;
- D5: The water irrigation system provides data with a sufficiently accurate precision;
- D6: Soil moisture sensors provide data with a sufficiently accurate precision;
- D7: Each farmer owns at least one device connected to Internet;
- D8: Each farmer has the competences for properly accessing the DREAM system and registering to it;
- D9: All the data the farmers insert is correct;
- D11: When a farmer does something to a certain area (for example, applies a fertilizer or harvests), he covers the whole area;

Goal G4 requires that all the relevant data is known to DREAM (R1, R2, R3, R5, R6, R7, R8, R9, R10), that DREAM is able to retrieve weather forecasts (R27) and that DREAM is able to find correlations among the production of the farmers, environmental factors and fertilizers using past data (R23) in order to make predictions for the future, provided to farmers in the form of suggestions (R24, R25).

G5 The farmers ask for and receive help from agronomists and other farmers.

- R5: DREAM shall allow farmers to insert the location and the area of their plot of land;

- R11: DREAM shall allow farmers to insert descriptions of problems they face;
- R12: DREAM shall allow farmers to request a visit of an agronomist when inserting a problem;
- R13: When a farmer requests a visit of an agronomist, DREAM inserts the visit to the farmer into the daily plan of an agronomist assigned to the area of the farmer;
- R14: When a farmer requests a visit of an agronomist, if no agronomist is available in the area of the farmer during the seven days after the request (excluding the day when the request is made), DREAM notifies the farmer about this situation.
- R18: DREAM allows policy makers and agronomists to mark/unmark a farmer as best-performing or worst-performing;
- R29: DREAM shall allow agronomists to insert the area they are responsible of;
- R31: DREAM allows farmers to choose the (best-performing) farmer or agronomist (assigned to his area) who to issue a request;
- R32: DREAM allows the farmers to send a request to a best-performing farmer or agronomist;
- R33: DREAM notifies (best-performing) farmers and agronomists of the requests of help from other farmers;
- R34: DREAM allows best-performing farmers and agronomists to insert a message of response to the farmers who have made a request for help to them;
- R35: DREAM sends the response to the farmer who issued the corresponding request;
- R48: DREAM shall allow users to log in the system by using an email and a password;
- D7: Each farmer owns at least one device connected to Internet;
- D8: Each farmer has the competences for properly accessing the DREAM system and registering to it;
- D9: All the data the farmers insert is correct;
- D10: (Best-performing) farmers answer to help requests sent to them;
- D15: Each agronomist owns a device connected to the Internet network and with a browser installed;

- D16: The agronomists are given a password – associated to their mail – by the Telangana agriculture department and such account is known to DREAM;
- D17: Each agronomist has the competences for properly accessing the DREAM system and using it;
- D18: All the data the agronomists insert is correct;
- D19: Agronomists reply to help requests issued by farmers;

Farmers can ask for help through the DREAM system in two ways: by requesting an agronomist's visit when inserting a problem they are facing (R5, R11, R12, R13, R14, R15, R29) or by asking a question to an agronomist or a best performing farmer (R5, R18, R29, R31, R32, R33, R34, R35) .

G6 The agronomists can plan farm visits based on the farmers performances.

- R5: DREAM shall allow farmers to insert the location and the area of their plot of land;
- R12: DREAM shall allow farmers to request a visit of an agronomist when inserting a problem;
- R13: When a farmer requests a visit of an agronomist, DREAM inserts the visit to the farmer into the daily plan of an agronomist assigned to the area of the farmer;
- R18: DREAM allows policy makers and agronomists to mark a farmer as best-performing or worst-performing;
- R29: DREAM shall allow agronomists to insert the area they are responsible of;
- R38: DREAM is able to generate a daily plan of visits for each agronomist evenly distributing (with exceptions to bad-performing farmers) over the year the number of visits to each farmer;
- R39: DREAM enforces that every farmer is assigned a visit in the daily plan of an agronomist at least twice a year;
- R40: DREAM enforces that no agronomist in a certain area is assigned more than twice of the visits than an agronomist of the same area;
- R41: DREAM enforces that the daily plan of agronomists includes more visits to worst-performing farmers than to other ones;
- R42: DREAM allows agronomists to accept or modify -i.e.: remove, add or move a visit - the automatically generated daily plan;

- R43: DREAM allows agronomists to confirm a daily plan at the end of the working day;
- R44: DREAM allows agronomists to specify deviations -i.e.: remove, add or move a visit- from a daily plan at the end of the working day;
- R45: DREAM is able to notify farmers involved by a visit of an agronomist;
- R46: DREAM allows agronomists to visualize data about farmers to visit - such as name, surname and location of the farm - and their production since their last visit to the farm;
- R47: DREAM allows agronomists to visualize the problems inserted by a farmer to be visited since their last visit to the farm;
- R48: DREAM shall allow users to log in the system by using an email and a password;
- D7: Each farmer owns at least one device connected to Internet;
- D8: Each farmer has the competences for properly accessing the DREAM system and registering to it;
- D9: All the data the farmers insert is correct;
- D15: Each agronomist owns a device connected to the Internet network and with a browser installed;
- D16: The agronomists are given a password – associated to their mail – by the Telangana agriculture department and such account is known to DREAM;
- D17: Each agronomist has the competences for properly accessing the DREAM system and using it;
- D18: All the data the agronomists insert is correct;

In order to achieve this goal, DREAM must be able to identify farmers belonging to the area of the agronomist (R5, R29) and among them the ones who are not performing well, that are the ones marked as worst-performing by policy makers and agronomists (R18) and the ones who requested for a visit by an agronomist (R12). Then, DREAM must be able to generate a proposal of daily plan (R13, R38, R39, R40, R41) which can be modified or accepted by the agronomist (R42, R43, R44). Finally, DREAM must be able to notify the involved farmer of the meeting (R45). Moreover, agronomists must be able to prepare themselves for the visit by visualizing data about farmers and problems inserted by them (R46, R47)

G7 The farmers can interact with other farmers exchanging opinions about agriculture.

- R36: DREAM allows farmers to initiate a new thread in the forum;
- R37: DREAM allows the farmers to add a post to a thread in the forum;
- R48: DREAM shall allow users to log in the system by using an email and a password;
- D7: Each farmer owns at least one device connected to Internet;
- D8: Each farmer has the competences for properly accessing the DREAM system and registering to it;

To achieve goal G7, DREAM is required to put in contact different farmers through a forum (R36, R37).

G8 The agronomists visualize weather forecasts regarding the area they are responsible of.

- R27: DREAM is able to connect to the Telangana government website to fetch forecasts for the chosen date;
- R28: DREAM can show weather forecast data for a certain location and date;
- R29: DREAM shall allow agronomists to insert the area they are responsible of;
- R30: DREAM shall allow agronomists to choose the date of the weather forecasts to visualize;
- R48: DREAM shall allow users to log in the system by using an email and a password;
- D3: The TSDPS provides weather forecasts and reports which can be accessed by DREAM;
- D15: Each agronomist owns a device connected to the Internet network and with a browser installed;
- D16: The agronomists are given a password – associated to their mail – by the Telangana agriculture department and such account is known to DREAM;
- D17: Each agronomist has the competences for properly accessing the DREAM system and using it;
- D18: All the data the agronomists insert is correct;

Goal G8 imposes that DREAM is able to fetch weather forecasts (R27) according to the area of an agronomist (R29) and the date he has chosen (R30). Of course it shall be able to show the results (R28).

G9 The agronomists are able to identify the best-performing farmers and the worst-performing farmers of the area they are responsible of.

- R1: DREAM collects soil moisture data from the soil moisture sensors;
- R2: DREAM accesses weather reports data provided by the TSDPS system;
- R3: DREAM collects watering data from the water irrigation system;
- R4: DREAM shall allow agronomists to insert feedback about the farmers they visited;
- R5: DREAM shall allow farmers to insert the location and the area of their plot of land;
- R6: DREAM shall allow farmers to insert the type of products they grow in a certain area of their farms;
- R7: DREAM shall allow farmers to insert data about their sowing activity;
- R8: DREAM shall allow farmers to insert data about their planting activity;
- R9: DREAM shall allow farmers to insert data about their harvesting activity;
- R10: DREAM shall allow farmers to insert data about the irrigation of their plantations;
- R11: DREAM shall allow farmers to insert descriptions of problems they face;
- R15: DREAM can use different criteria (namely, productivity in terms of harvested quantity over sowed quantity, productivity in presence of adverse meteorological events, productivity in presence of drought, and so on) and/or combinations of them to rank farmers;
- R16: DREAM shall allow policy makers and agronomists to choose the criteria to use for ranking farmers and the associated weights (for combinations of them);
- R17: DREAM shall allow policy makers and agronomists to choose the time period and the area to consider for ranking the farmers;
- R18: DREAM allows policy makers and agronomists to mark/unmark a farmer as best-performing or worst-performing;

- R29: DREAM shall allow agronomists to insert the area they are responsible of;
- R48: DREAM shall allow users to log in the system by using an email and a password;
- D1: The location of the sensors is known to the DREAM system;
- D2: The location of the water irrigation systems is known to the DREAM system;
- D3: The TSDPS provides weather forecasts and reports which can be accessed by DREAM;
- D4: The weather reports provide sufficiently accurate data;
- D5: The water irrigation system provides data with a sufficiently accurate precision;
- D6: Soil moisture sensors provide data with a sufficiently accurate precision;
- D7: Each farmer owns at least one device connected to Internet;
- D8: Each farmer has the competences for properly accessing the DREAM system and registering to it;
- D9: All the data the farmers insert is correct;
- D11: When a farmer does something to a certain area (for example, applies a fertilizer or harvests), he covers the whole area;
- D15: Each agronomist owns a device connected to the Internet network and with a browser installed;
- D16: The agronomists are given a password – associated to their mail – by the Telangana agriculture department and such account is known to DREAM;
- D17: Each agronomist has the competences for properly accessing the DREAM system and using it;
- D18: All the data the agronomists insert is correct;

This goal forces the system to own all the relevant data for the analysis (R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11) and to be able to compare such data (R15) according to choices of the agronomists (R16, R17). Finally, DREAM shall allow agronomists to mark farmers as worst-performing or best-performing (R18).

3.2.3 Scenarios

A storm ruined the harvest Arin is a farmer who mainly cultivates potatoes, onions and tomatoes in the fields of his family. The harvest of March was really good, and Arin hoped that the same volume of production might have been repeated in April as well. Unfortunately, the North Indian Ocean cyclone season starts in April, and in the second week of this month Tauktae storm, the strongest storm of of the season, violently hit Arin's lands, halving the crop.
Consequently, Arin wants to share in the system (`DREAM`) the outcome of the storm. Arin opens its browser and accesses the system using its credentials. Next, he selects from a menu in the incoming webpage the entry for inserting data about production and problems. Then he fills the form with a description about the storm and what it caused to its fields, and finally he sends it.
Arin hopes to receive some aid.

Great year for the potatoes Like many other farmers in India, Ikbal knows that the best period for cultivating potatoes is during the months from October to December. The weather is neither hot nor cold and the monsoons are nearly over at this time. Therefore, last October Ikbal planted many potatoes, and in February, with the help of some peasants who work for him, has picked up all of them.
In order to publish in `DREAM` the results of this year potatoes' harvest, he access the system through its computer, opens the form for inserting data about the production and then starts to fill it. The form is quite structured, and requires some data for each entry. After having inserted all the objective information about the crop, Ikbal would like to share something he has found out during this year, namely that planting the potatoes at depth x allows to retrieve substantial crops of this vegetable. Thus, Ikbal fills in also the optional text box and then sends it.

Registering to the system as an agronomist Dalbir worked for many years as an agronomist in Bihar, but recently he has been moved to Telangana. Telangana uses this innovative system, `DREAM`, to keep track of the harvests of farmers, to collect interesting data from sensors and many other things. All the agronomists in Telangana must work with `DREAM`, therefore Dalbir has to create an account.
Dalbir opens the webpage of `DREAM` and inserts the credentials he previoulsy received by `DREAM`'s administrators. Next, Dalbir has to insert some data about his job in order to finish the registration. Among the others, Dalbir is asked to insert the area where he works.

Visiting a farm The first task of Dalbir is to visit the farm of Arin. The date and hour of the meeting has been established by `DREAM`, which has notified also Arin about the meeting. Arin has been chosen because last month he performed very bad.

During the meeting, Arin has the opportunity of explaining all the problems he faced during the past month, one for all Tauktae (the strongest storm of the North Indian Ocean cyclone season). Arin takes Dalbir to the fields, where Dalbir can gather some sample from the terrain. At the end of the meeting, Dalbir gives Arin some advices, and then he moves to another farm.
Once in the office, Dalbir inserts in the system a report about his visit to Arin's farm.

A bad schedule Like every evening, Dalbir accesses `DREAM` for downloading the daily plan of the next day. Every daily plan provided by the system contains time and place of every meeting, in addition to some information about the farmer. Dalbir notices that Ikbal's farm is closer to his parents' house, and Dalbir knows that they would be really happy to if he had lunch with them.
However, the meeting with Ikbal is scheduled for the 3p.m o' clock. Therefore, Dalbir decides to exchange the meetings with Ikbal and Tarak, visiting Ikbal at 11.30a.m. and Tarak at 3p.m.. Through the user interface of `DREAM`, Dalbir exhanges the two meetings and once he has confirmed, `DREAM` notifies the farmers of the change.

Starting a new cultivation The farmer Ikbal has been cultivating potatoes in his piece of land for ten years. As he has recently bought some acres of land next to his one, he would like to start the cultivation of a new type of product. As he has some doubts about which product to seed, he decides to ask for help through the use of the `DREAM` system. He logs in the `DREAM` website and selects the "My Requests" section in the home page. Then, he clicks on the "New Request" button and a menu appears, where he can choose whether to issue the request to an agronomist or to another farmer, selected among the others because of his good results. Ikbal decides to contact an agronomist, and so selects the "Request to Agronomist" option. On the screen a form appears, where Ikbal can select a farmer among the ones assigned to the area where his plot of land is located and insert a title and some text to express his question to the agronomist. Ikbal chooses to issue his request to the agronomist Dalbir, as he has already met him and received some precious advice from him. Ikbal fills the text input and clicks on the button "Send Request". After sending the request, the screen shows him a confirm that the request has been sent, the agronomist which the request has been sent to and the text of the request. Ikbal returns to the home page, and decides that he also wants to contact one of the best performing farmers, in order to compare different opinions. So he selects again the "My Requests" section, chooses to issue a new request and selects the "Request to Farmer" option. A form analogous to the previous one appears, and Ikbal chooses to ask farmer Arin, and inserts the text of his request. He finally clicks on the "Send Request" button and visualizes the confirm of his request.
The next day Arin opens the `DREAM` website to insert some data about his production, and sees a notification about a request sent to him. He enters the

"My Replies" section of the website, where all requests issued to him are shown in descending temporal order. He sees that there is a new request from Ikbal, and clicks on it to read the entire text of the request. After reading the question from Ikbal, he writes down his reply in a text area, telling Ikbal that, from his point of view, the best cultivation to start in the new plot of land is wheat, as it grows very well in the area where Ikbal works and the seeding season for wheat starts in just one month. Arin clicks on the "Send Reply" button and sees a confirm that the reply has been sent to Ikbal.
In the same day, also the agronomist Dalbir opens the DREAM website and sees a notification about a request sent to him. Just as Arin did, he enters the "My Replies" section of the website and opens the request sent by Ikbal. He writes down his answer in the text input area, suggesting Ikbal to plant either some sunflowers or some sesamum, as the demand for these sunflower oil and sesamum oil is increasing, and some companies have asked him where they could find more cultivations of these products. Finally, he sends the request to Ikbal and sees a confirm message.
After a full workday in his field, Ikbal returns to his house and opens the DREAM website to check whether Arin or Dalbir have answered to his question. He is glad to see the notifications about the two replies, and opens the "My Requests" section of the website. On the top of the list of the requests issued, he sees that the first two requests have been answered. First, he clicks on the request sent to Arin, and the request together with the reply is shown on the screen. He writes down a thanks message to Arin in a text box and clicks on the "Send" button. Then he goes back to "My Requests" and selects the request sent to Dalbir. After reading the answer provided by Dalbir, he thanks him just as he did with Arin and tells him that he has decided to start a cultivation of sunflowers, as he is attracted by the earnings perspective.

A chat about fertilizers for sunflowers The farmer Ikbal has started his new cultivation of sunflowers, and he feels insecure about which fertilizer to choose among the several ones available. He would like to discuss a bit with some other farmers that grow sunflowers, and learn from their experience. So he decides to open the DREAM website and enter the "Forum" section. He reads the list of the already present discussion threads, and sees that there isn't any discussion on fertilizers for sunflowers. Therefore, he decides to create a new thread selecting the "New Thread" button. He inserts the topic of the forum and creates the thread. Then he clicks on the "New Post" button and writes the first post of the forum, asking for advice about which fertilizer to use on a sunflowers field in the spring season.
After a couple of hours, the farmer Monisha opens the "Forum" section of the DREAM website, as she does every evening, to keep up with the latest conversations between her colleagues. She notices that a new thread has been created, and selects it from the list of discussion threads to visualize the posts inside it. She reads the post written by Ikbal, and as she is quite expert in the cultivation of sunflowers, she decides to give some advice to Ikbal. She creates a new post,

where she explains Ikbal that every fertillizer with a good amount of nitrogen and phosphorus is good for sunflowers, and points to him a couple of fertilizer which she has successfully used for her sunflowers.
The next day, Ikbal opens again the "Forum" section of the DREAM website, curious to see if someone wrote on the thread that the created. He is glad to see not only the post from Monisha, but also a couple of other posts from other two farmers, discussing about the fertilizers. He writes another post to thank Monisha and the other farmers.

A successful policy Shyla is a policy maker in the Telangana region, whose main area of interest is agriculture. One of the policies she worked on involved giving incentives to farmers in order to try and prevent damages to crops caused by storms (e.g. by financing the construction of protective infrastructure).
A year after the approval of her policy, Shyla logs into DREAM and accesses the Analysis section. She accesses the production data statistics for the current year, and compares them to last year's data. Thanks to the data analysis capabilities provided by DREAM, she is able to confirm her policy helped significantly reduce damage caused by adverse weather conditions.

3.2.4 Use cases

In this section we show some important use cases. It should be noticed that, in the sequence diagrams, we have decided to do not represent all the possible branches and situations, but only the most relevant. In particular, in many cases the underlying assumption that the user has already logged in is made.
As far as the use cases model is concerned, the main actors are highlighted. Furthermore, in order to make the diagram easier to read, we have decided to exploit generalizations of actors (e.g. User) and of use cases (e.g. ViewData) in a little improper way to reduce the number of associations.

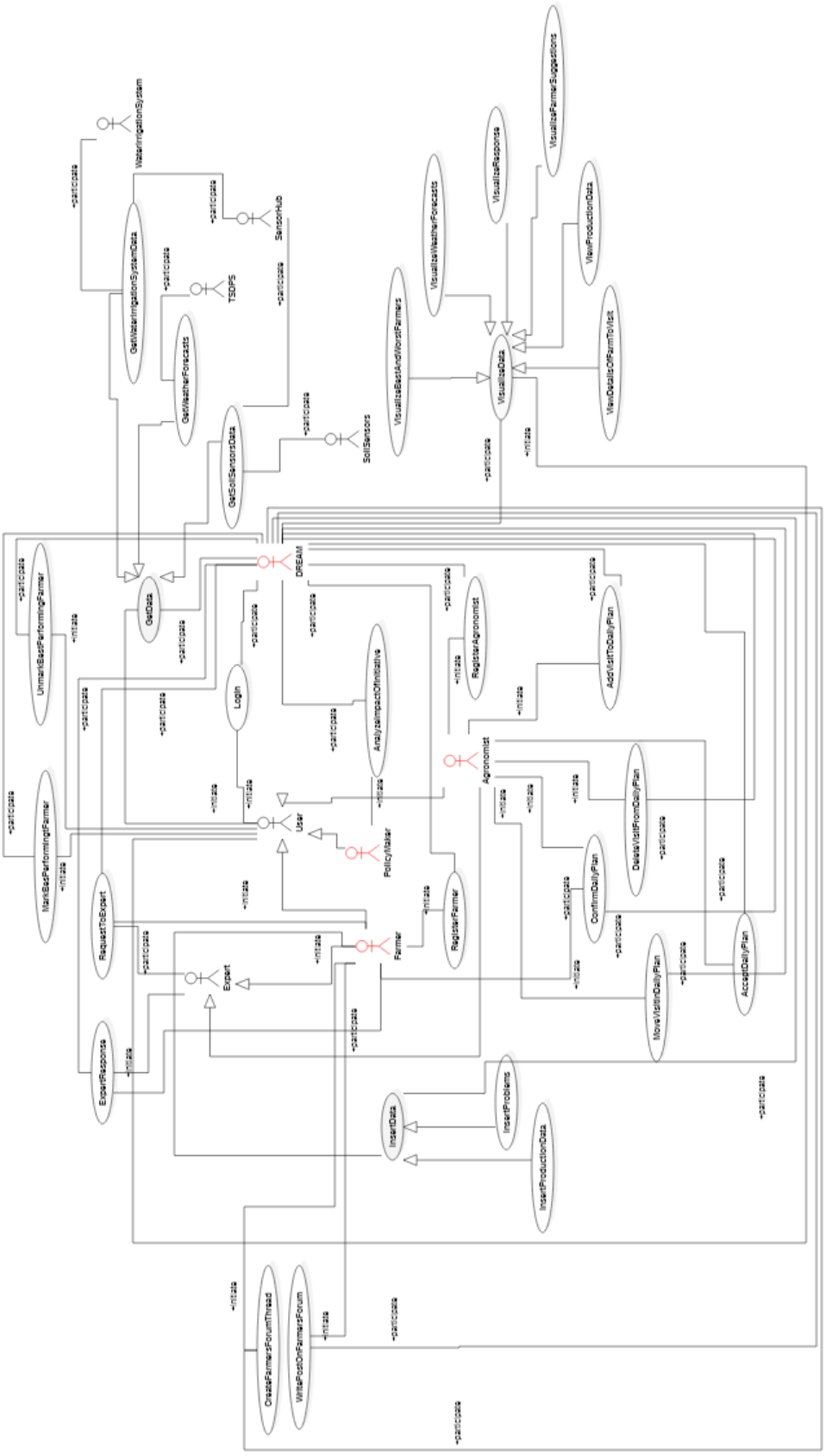

Figure 9: This is the use cases model of the system.

Table 7: Use case 1: RegisterFarmer

USE CASE 1	
Name	RegisterFarmer
Actor	Farmer
Entry condition	The farmer does not have an account and wants to enrol in the `DREAM` system.
Event flow	1. The farmer accesses to the `DREAM` landing page. 2. `DREAM` shows a form for logging in and a button "Sign Up Farmer". 3. The farmer selects the "Sign Up Farmer" button. 4. `DREAM` shows a form to insert the farmer's personal data (name, surname, birthdate), his/her e-mail, a password and the location of his/her farm. 5. The farmer inserts his name, surname, birthdate, e-mail, password and the location of his/her farm. 6. The farmer clicks on the "Sign Up" button. 7. `DREAM` checks the validity of the data, and then it sends an e-mail to the inserted e-mail address, containing a confirmation link to complete the enrolling. 8. `DREAM` shows a message advising the farmer to check his e-mail and click on the confirm link. 9. The farmer clicks on the confirm link sent in the e-mail.

Exit condition	DREAM shows a confirm of having created the account correctly and a link to the farmer's homepage.
Exceptions	• If the e-mail inserted by the farmer is already used in the DREAM system, an error message is shown. • If one of the data inserted by the farmer is empty, an error message is shown. • If the password is shorter than 10 characters, an error message is shown. • If the farmer 24 hours after having inserted his data has not clicked on the confirm link, the signing up process is canceled and if the farmer clicks on the link an error message is shown.
Special requirements	• The e-mail with the confirm link should be sent by DREAM in less than 10 seconds. • After the farmer has clicked on the confirm link, the confirm message should be shown in less than 3 seconds.

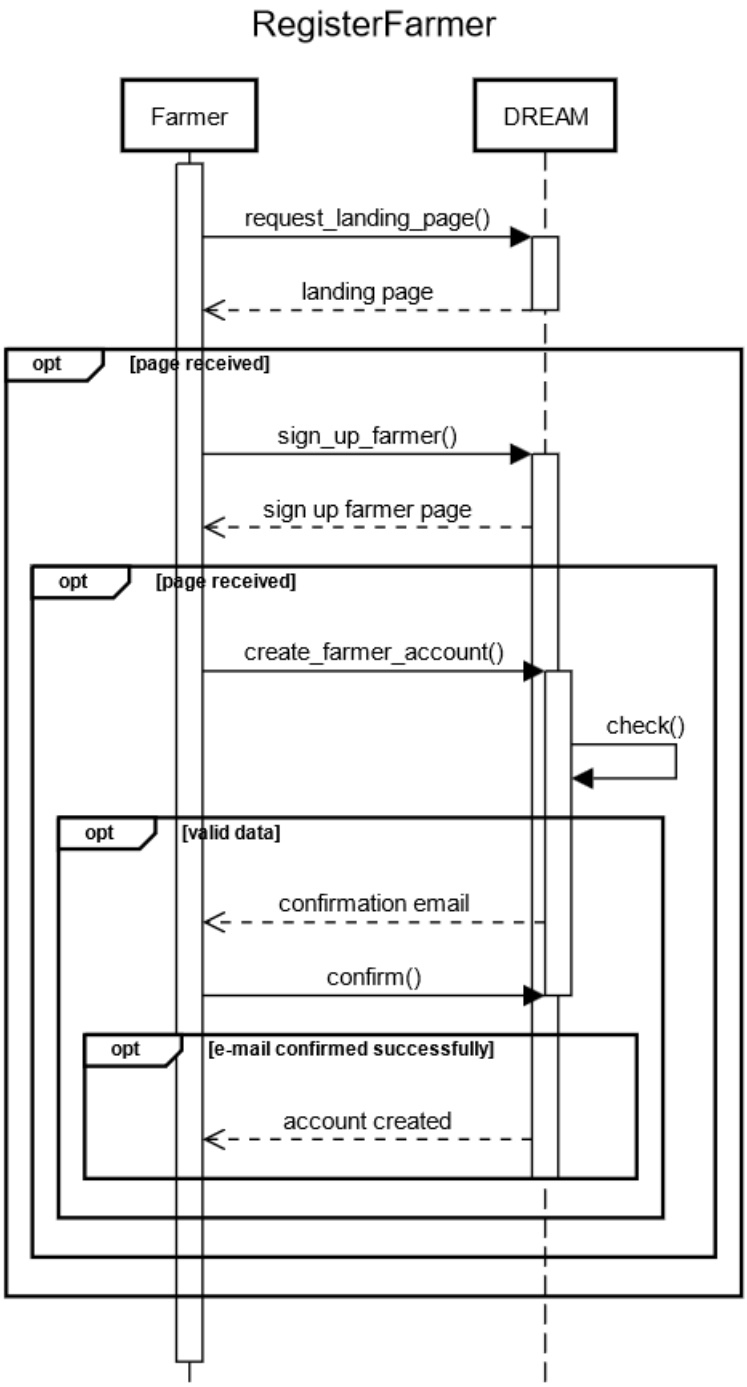

Figure 10: Sequence diagram for the RegisterFarmer use case.

Table 8: Use case 2: RegisterAgronomist

USE CASE 2	
Name	RegisterAgronomist
Actor	Agronomist
Entry condition	The agronomist logs in the `DREAM` system for the first time with the credentials he has already received.
Event flow	1. `DREAM` shows a form for inserting the agronomist's personal and working data (name, surname, birthdate, optionally a specialization) and the area he is reponsible of. 2. The agronomist inserts his personal and working data. 3. The agronomist selects the area he is responsible of. 4. The agronomist clicks on the "Done" button.
Exit condition	The agronomist is shown the agronomist's homepage of the `DREAM` system.
Exceptions	• If the agronomist inserts an empty name, surname or birthdate, an error is shown. • If the agronomist does not select the area which he/she is assigned to, an error is shown.

Special requirements	• After clicking on the "Done" button, the agronomist must be shown the homepage in less than 3 seconds.

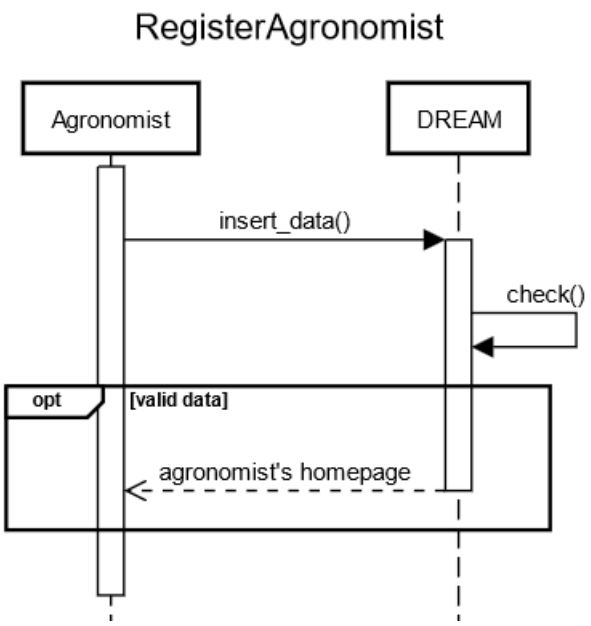

Figure 11: Sequence diagram for the RegisterAgronomist use case.

Table 9: Use case 3: LogIn

USE CASE 3	
Name	LogIn
Actor	User (either a Farmer, an Agronomist or a Policy Maker)
Entry condition	The user is not logged in the system and wants to log in.
Event flow	1. The user accesses to the `DREAM` landing page. 2. `DREAM` shows a form for logging in. 3. The user inserts his/her e-mail, password and selects his role (either farmer, agronomist or policy maker). 4. The user clicks on the "Log In" button.
Exit condition	The user is logged in and is shown the homepage of the `DREAM` system.
Exceptions	• If the credentials inserted by the user are not correct, an error message is shown on the landing page and the log in is not allowed.

Special requirements	• After clicking on the "Log In" button, the user must be shown the homepage in less than 3 seconds.

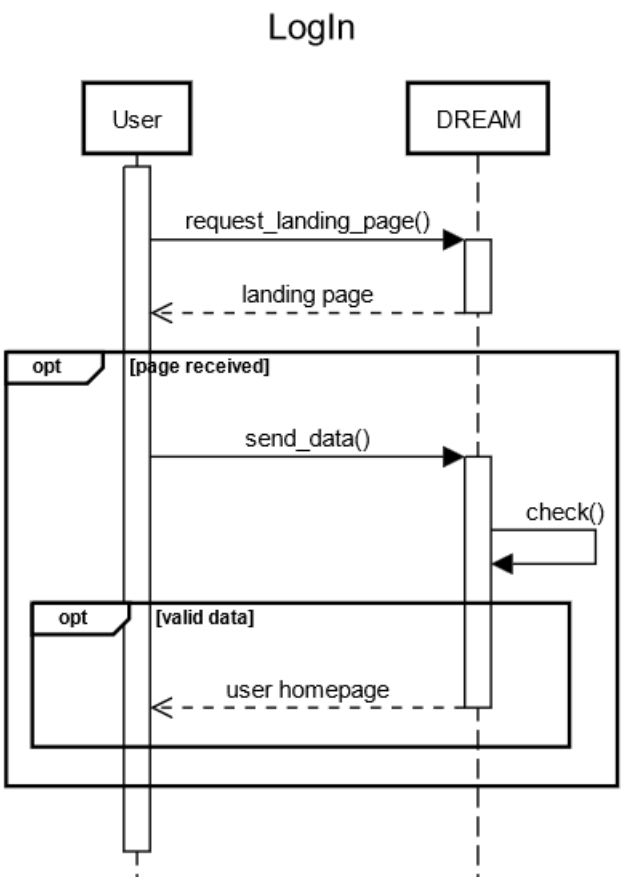

Figure 12: Sequence diagram for the LogIn use case.

Table 10: Use case 4: ViewProductionData

USE CASE 4	
Name	ViewProductionData
Actor	Farmer
Entry condition	The farmer is logged in the `DREAM` system and wants to visualize his/her past production data.
Event flow	1. The farmer selects the "My Productions" entry in the homepage. 2. `DREAM` shows a list of production data items, each one of them containing a date and the corresponding activities performed by the farmer in that date (seeding, planting, fertilizing, watering and harvesting) with the relative details, the comments about these activities inserted by the farmer and, potentially, the problems inserted by the farmer relative to that date Moreover, an "Insert Production Data" button and a "Select Dates" form are shown. 3. The farmer inserts in the "Select Dates" form the start and end date for which he/she wants to visualize his production data. 4. The farmer clicks on the "Select Dates" button.
Exit condition	The user is shown a list of production data relative to the dates between the start and end dates - included - inserted.

Exceptions	• If no production data are relative to the inserted start and and dates, a warning message is shown.
Special requirements	• After clicking on the "My Production" button, the user must be shown the view in less than 3 seconds. • After clicking on the "Select Dates" button, the user must be shown the view in less than 3 seconds.

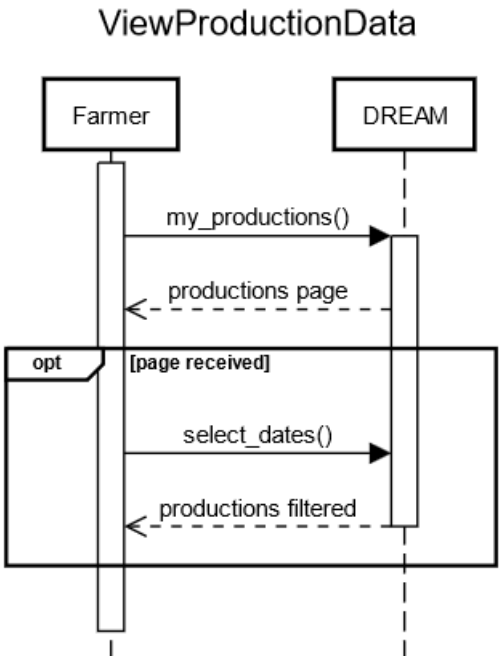

Figure 13: Sequence diagram for the ViewProductionData use case.

Table 11: Use case 5: InsertProductionData

USE CASE 5	
Name	InsertProductionData
Actor	Farmer
Entry condition	The farmer wants to insert some data about his/her production activity. He/she is already logged in the `DREAM` system.

Event flow	1. In the homepage, the farmer selects from the menu the item "My Production". 2. DREAM shows a list of production data items and a "Insert Production Data" button. 3. The farmer clicks on the "Insert Production Data" button. 4. DREAM shows a form that allows the farmer to insert the date the production data is referred to; for each of his plantations, the actions (seeding, planting, fertilizing, watering, harvesting) the farmer has performed on a single day of work and for each action some more details (for seeding, planting and harvesting the quantity of product involved, for fertilizing the type and quantity of the fertilizer used , for watering the amount of water used). Moreover, the farmer can also insert a textual description of the activity performed. 5. The farmer inserts the date which the production data are referred to. 6. For each product type, the farmer selects the action(s) he performed and the details related to that action(s). 7. Optionally, the farmer inserts a brief description of the activity performed or some observations about it. 8. The farmer clicks on the "Insert" button. 9. DREAM checks the validity of the data.
Exit condition	The farmer is shown a confirm of having inserted the data into the system together with the data just inserted.

Exceptions	• If the kind of product is not present, the farmer can add the new product together with the area dedicated to it.
Special requirements	• When the farmer clicks on the "Insert" button, the confirmation is shown in less than 3 seconds.

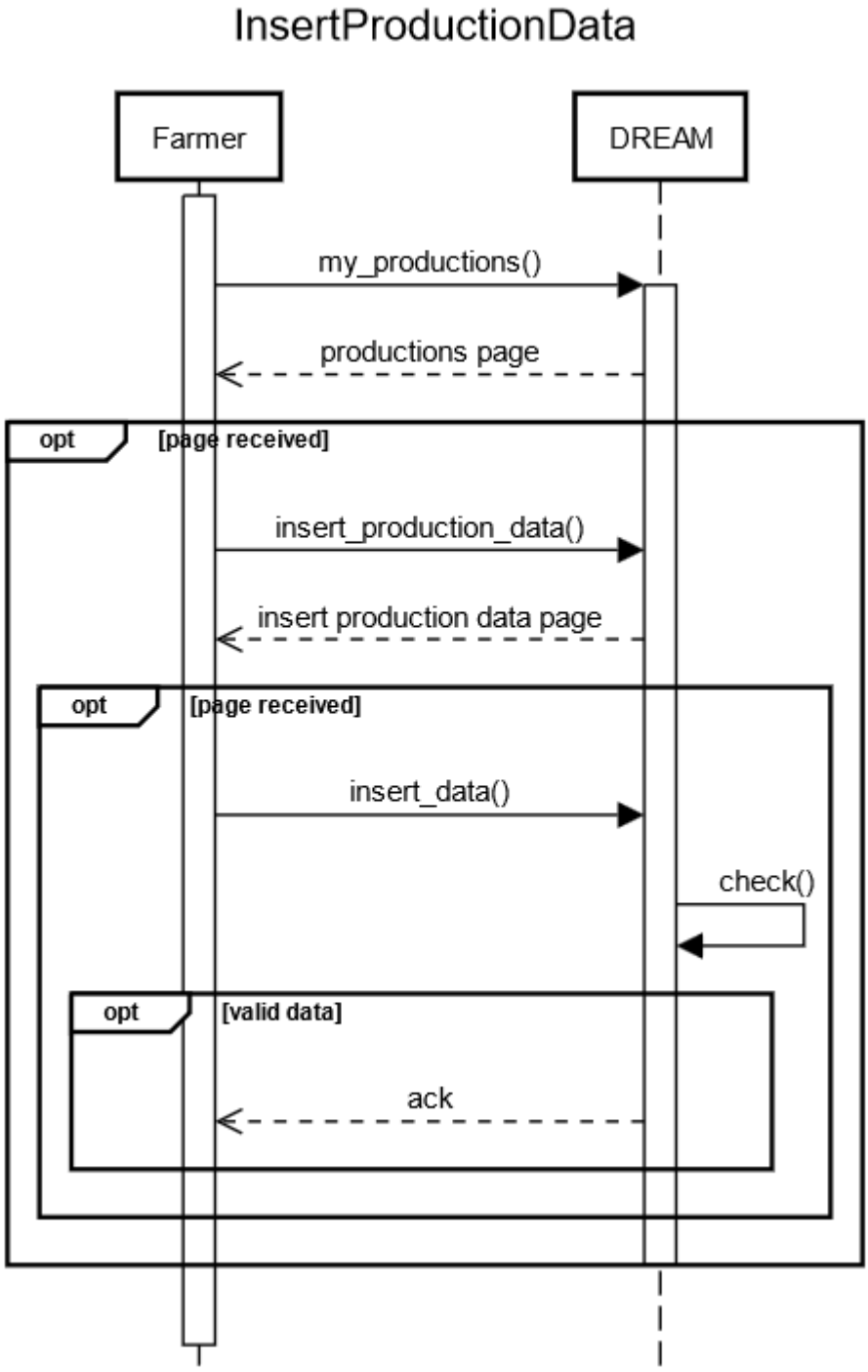

Figure 14: Sequence diagram for the InsertProductionData use case.

Table 12: Use case 6: InsertProblems

USE CASE 6	
Name	InsertProblems
Actor	Farmer
Entry condition	The farmer faces a problem and decides to communicate it on `DREAM`. He/she is already logged in the `DREAM` system.
Event flow	1. In the homepage, the farmer selects from the menu the item "Insert Problem". 2. `DREAM` shows a form to insert a problem. 3. The farmer selects the kind of problem among a list of available ones, when the problem occurred and how much it lasted. 4. The farmer gives a brief description of the problem. 5. The farmer selects whether he would like to meet as soon as possible an agronomist. 6. The farmer clicks on the "Insert" button.
Exit condition	The farmer is shown the problem he inserted with a confirm of having successfully inserted it into the system. If the farmer requested a visit from an agronomist, an agronomist assigned to the area of the farmer is scheduled the visit in his daily plan during the following days and the farmer is notified about the date and time of the visit.

Exceptions	• If the kind of problem is not present, the farmer can add it by its own. • In case no agronomist can be scheduled for the current week, the farmer is notified about the problem and an agronomist will be found for the next week.
Special requirements	• After the farmer clicks on the "send" button, the problem is inserted in the system (and the visit of the agronomist is scheduled, if requested) in less than 3 seconds.

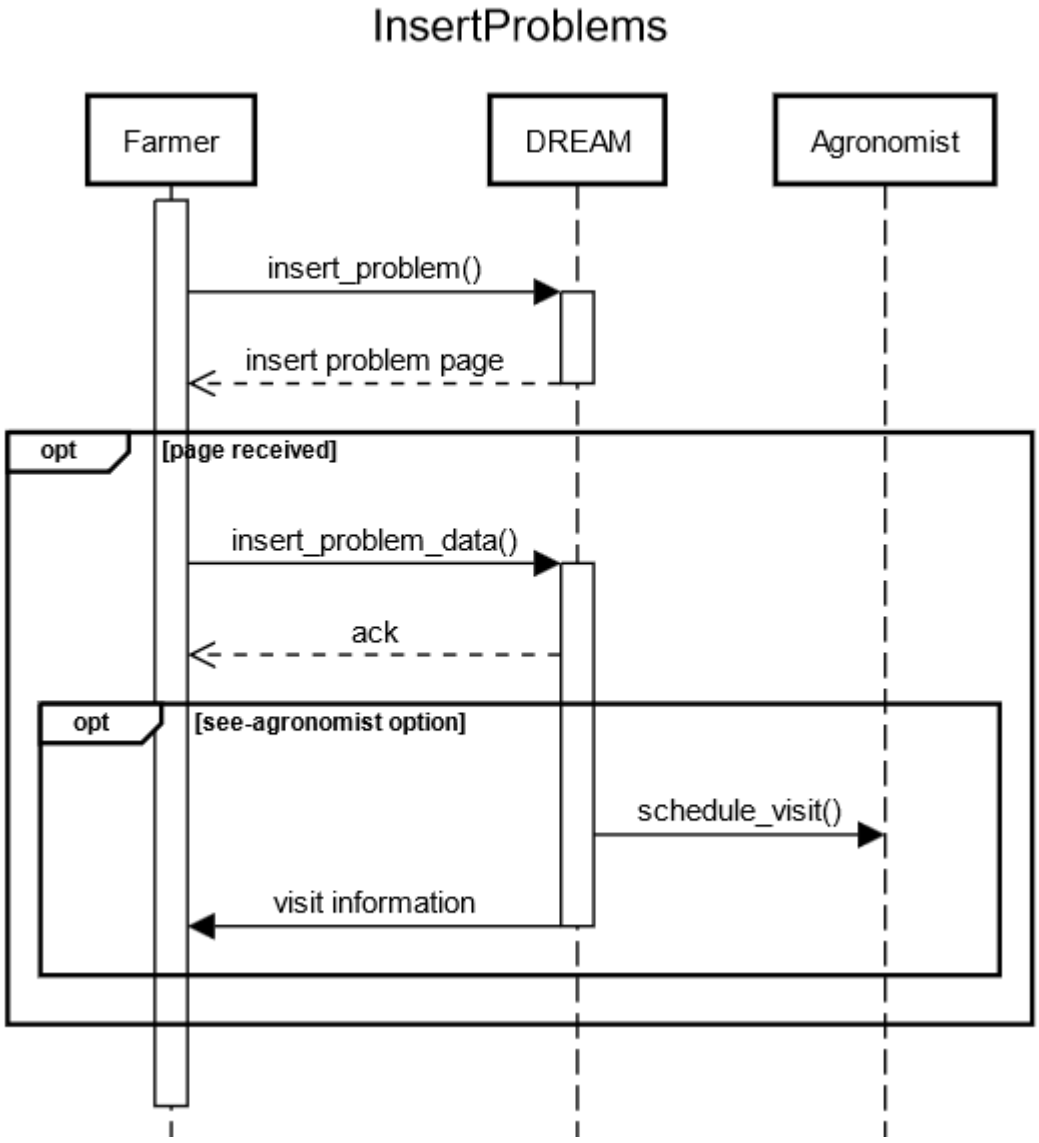

Figure 15: Sequence diagram for the InsertProblems use case

Table 13: Use case 7: AcceptDailyPlan

USE CASE 7	
Name	AcceptDailyPlan
Actor	Agronomist
Entry condition	The agronomist is logged in the `DREAM` system and wants to visualize the daily plan for the next working day, which he has not accepted yet.
Event flow	1. The agronomist accesses the "Daily Plan" area from the homepage. 2. `DREAM` shows a list of 7 dates, starting with the present day and including only working days. 3. The agronomist selects the date of his/her next working day. 4. `DREAM` shows the daily plan of the agronomist for the selected date: it is a list of visits to farms. For each visit, the name and surname of the farmer, the location of the farm, the starting hour of the visit are shown; and there are a "Farm Details" button (view UC8 for more details), a "Move" button (view UC9 for more details) and a "Delete" button (view UC10 for more details). Moreover, an "Add Visit" button (view UC11 for more details) and a "Accept Daily Plan" button are present. 5. After possibly visualizing details about the farms to visit (UC8) or modifying the daily plan (UC9, UC10, UC11), the agronomist clicks on the "Accept Daily Plan button"

Exit condition	The agronomist is shown the daily plan for the selected date without the "Move", "Delete", "Add Visit" and "Accept Daily Plan" buttons, and all the farmers involved in the visits in the accepted daily plan are notified about the visit.
Exceptions	• If the agronomist has modified the daily plan and the daily plan is not acceptable, the daily plan is shown to the agronomist with all the buttons to modify it, together with an error message.
Special requirements	• When selecting the daily plan date, the agronomist must be shown the daily plan in less than 3 seconds.

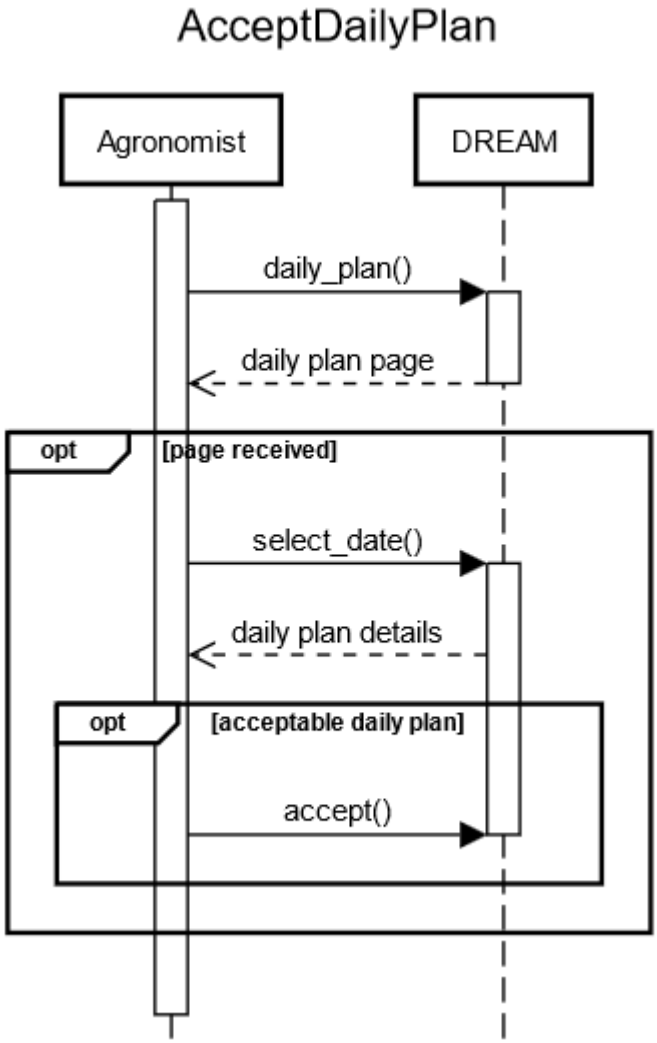

Figure 16: Sequence diagram for the AcceptDailyPlan use case.

Table 14: Use case 8: ViewDetailsOfFarmToVisit

USE CASE 8	
Name	ViewDetailsOfFarmToVisit
Actor	Agronomist
Entry condition	The agronomist is logged in the `DREAM` system and is visualizing the daily plan of a certain work day. He/She wants to see more details about a farm present in the daily plan.
Event flow	1. The agronomist clicks on the "Farm Details" button relative to a farm to visit. 2. `DREAM` shows a list of production data items and a list of problems inserted by the farmer owning the farm to visit, starting from the day after the last visit of the agronomist (or from the subscription of the farmer into the system, if the agronomist has never visited the farmer) until the present day.
Exit condition	The agronomist can read the details about the farmer, and clicking on a "Back" button, returning to the daily plan view.
Exceptions	• If there are no production data relative to the farm, a message is shown. • If there are no problems relative to the farm, a message is shown.

Special requirements	• When selecting the "Farm Details" button, the agronomist must be shown the farm details in less than 3 seconds.

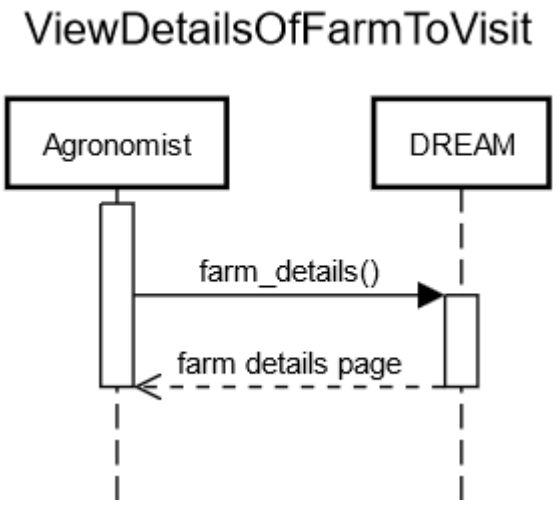

Figure 17: Sequence diagram for the ViewDetailsOfFarmToVisit use case.

Table 15: Use case 9: MoveVisitInDailyPlan

USE CASE 9	
Name	MoveVisitInDailyPlan
Actor	Agronomist
Entry condition	The agronomist is logged in the DREAM system and is visualizing the daily plan of a certain work day. He/She wants to move a visit from a starting hour to another one, either because the work day is ended and he/she performed the visit in a different hour than the selected one or because he/she wants to modify the daily plan for a future work day before accepting it.
Event flow	1. The agronomist clicks on the "Move" button relative to a certain visit. 2. DREAM shows an input field for inserting the starting hour of the visit and an "Insert" button. 3. The agronomist inserts the starting hour of the visit. 4. The agronomist clicks on the "Insert" button.
Exit condition	The agronomist is shown the daily plan of the work day, exactly like it was before the update, except for the fact that the starting date of the selected visit has been updated with the inserted one

Exceptions	• If the inserted starting date is outside the working hours of the agronomist, the daily plan is not updated and an error message is shown.
Special requirements	• When selecting the "Insert" button, the agronomist must be shown the updated daily plan in less than 3 seconds.

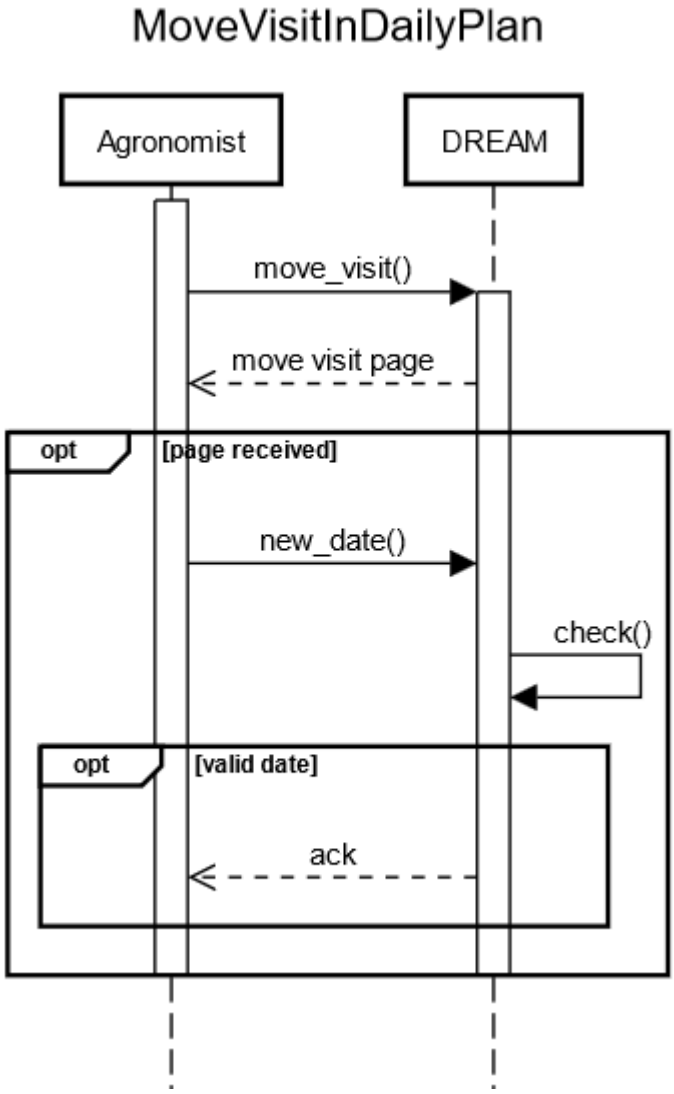

Figure 18: Sequence diagram for the MoveVisitInDailyPlan use case.

Table 16: Use case 10: DeleteVisitFromDailyPlan

USE CASE 10	
Name	DeleteVisitFromDailyPlan
Actor	Agronomist
Entry condition	The agronomist is logged in the `DREAM` system and is visualizing the daily plan of a certain work day. He/She wants to delete a visit, either because the work day is ended and he/she did not perform the visit or because he/she wants to modify the daily plan for a future work day before accepting it.
Event flow	1. The agronomist clicks on the "Delete" button relative to a certain visit.
Exit condition	The agronomist is shown the daily plan of the work day, exactly like it was before the update, except for the fact that the selected visit has been deleted.
Exceptions	-
Special requirements	• When selecting the "Delete" button, the agronomist must be shown the updated daily plan in less than 3 seconds.

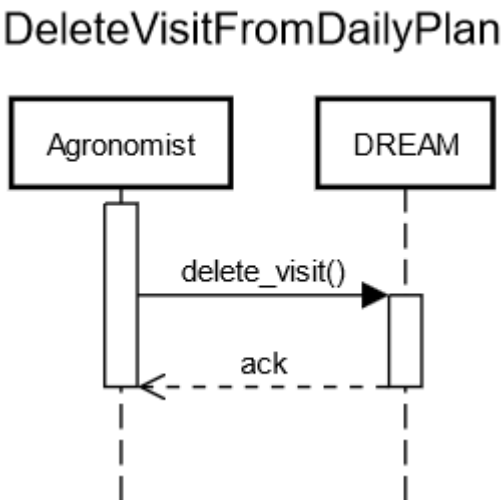

Figure 19: Sequence diagram for the DeleteVisitFromDailyPlan use case.

Table 17: Use case 11: AddVisitToDailyPlan

USE CASE 11	
Name	AddVisitToDailyPlan
Actor	Agronomist
Entry condition	The agronomist is logged in the `DREAM` system and is visualizing the daily plan of a certain work day. He/She wants to add a visit to the plan, either because the work day is ended and he/she performed a visit which was not planned or because he/she wants to modify the daily plan for a future work day before accepting it.
Event flow	1. The agronomist clicks on the "Add Visit" button. 2. `DREAM` shows a list of all farmers in the area of the agronomist and an input field for the starting hour of the visit. 3. The agronomist selects a farmer from the list of farmers. 4. The agronomist inserts a starting hour for the visit. 5. The agronomist clicks on the "Add" button.
Exit condition	The agronomist is shown the daily plan of the work day, exactly like it was before the update, except for the fact that the new visit has been added.

Exceptions	• If the inserted starting date is outside the working hours of the agronomist, the daily plan is not updated and an error message is shown.
Special requirements	• When selecting the "Add" button, the agronomist must be shown the updated daily plan in less than 3 seconds.

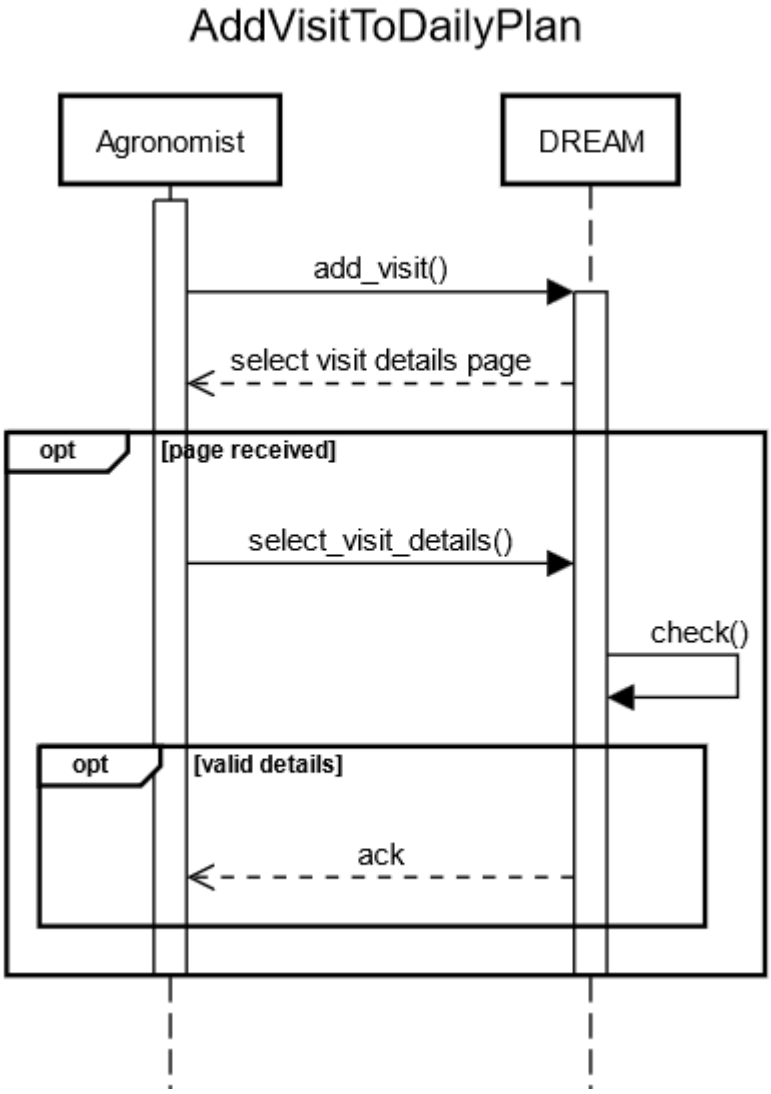

Figure 20: Sequence diagram for the AddVisitToDailyPlan use case.

Table 18: Use case 12: ConfirmDailyPlan

USE CASE 12	
Name	ConfirmDailyPlan
Actor	Agronomist
Entry condition	The working day of an agronomist has ended - more precisely, it is passed at least half an hour from the starting hour of the last visit planned for the agronomist - and the agronomist, who is already logged in the system, wants to confirm the daily plan of the day, possibly specifying some deviations.

Event flow	1. The agronomist accesses the "Daily Plan" area from the homepage. 2. DREAM shows a list of 7 dates, starting with the present day and including only working days. 3. The agronomist selects the date of the present work day. 4. DREAM shows the daily plan of the agronomist for the present work day. For each visit, the name and surname of the farmer, the location of the farm, the starting hour of the visit are shown; and there are a "Farm Details" button (view UC8 for more details), a "Move" button (view UC9 for more details), a "Delete" button (view UC10 for more details) and an input field for inserting a report about the visit. Moreover, an "Add Visit" button (view UC11 for more details) and a "Confirm Daily Plan" button are present. 5. The agronomist writes the reports for all the visits performed. 6. After possibly visualizing details about the farms to visit (UC8) or modifying the daily plan (UC9, UC10, UC11), the agronomist clicks on the "Confirm Daily Plan button"
Exit condition	The agronomist is shown the daily plan for the present work day without the "Move", "Delete", "Add Visit", "Insert Report" and "Confirm Daily Plan" buttons.

Exceptions	• If the agronomist does not insert the report (UC13) for one of the performed visits, an error message is shown and the system does not let the agronomist confirm the plan. • If the agronomist does not confirm the daily plan during the present day, the "Daily Plan" page will contain in the list of dates for the daily plans also the date of the unconfirmed daily plan, together with a message warning the agronomist to confirm the daily plan.
Special requirements	After having selected the "Confirm Daily Plan" button, the agronomist must be shown the daily plan in less than 3 seconds.

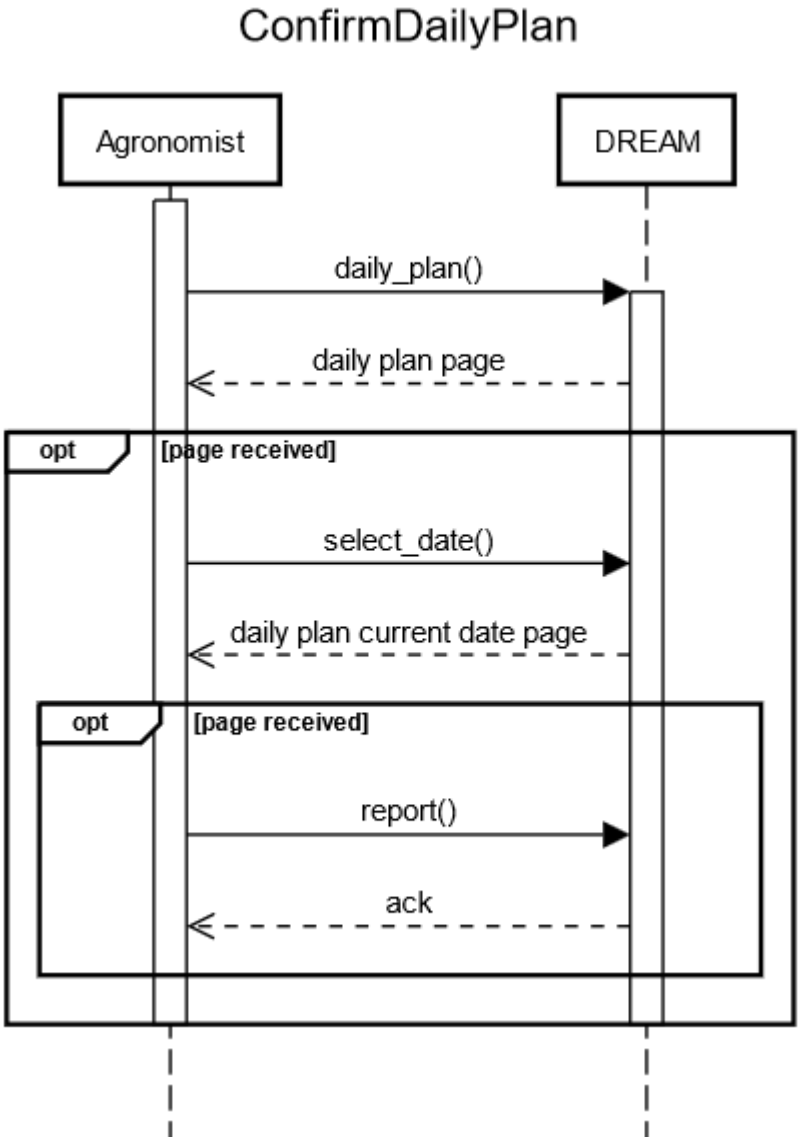

Figure 21: Sequence diagram for the ConfirmDailyPlan use case.

Table 19: Use case 13: GetSoilSensorsData

USE CASE 13	
Name	GetSoilSensorsData
Actor	SoilSensorsHub
Entry condition	It is 06:00, or 14:00, or 22:00 [19].
Event flow	1. `DREAM` sends the SoilSensorsrHub a request for data about soil moisture. 2. The SoilSensorsHub replies with a list of <sensorId, currentSoilMoisture> items, one for each sensor managed by the hub.
Exit condition	`DREAM` either receives the data or the maximum number of attempts is reached.
Exceptions	• In case the SoilSensorsHub is not reachable, `DREAM` retries to contact it 3 times, with an interval of 30 minutes between them; if after the third try the SoilSensorsHub has not been contacted successfully yet, `DREAM` will contact it in 6:30 hours (that is, at the next scheduled time for contact the SoilSensorsHub).
Special requirements	-

[19] All times are referred to the UTC+5:30 time zone, which is the one used in India

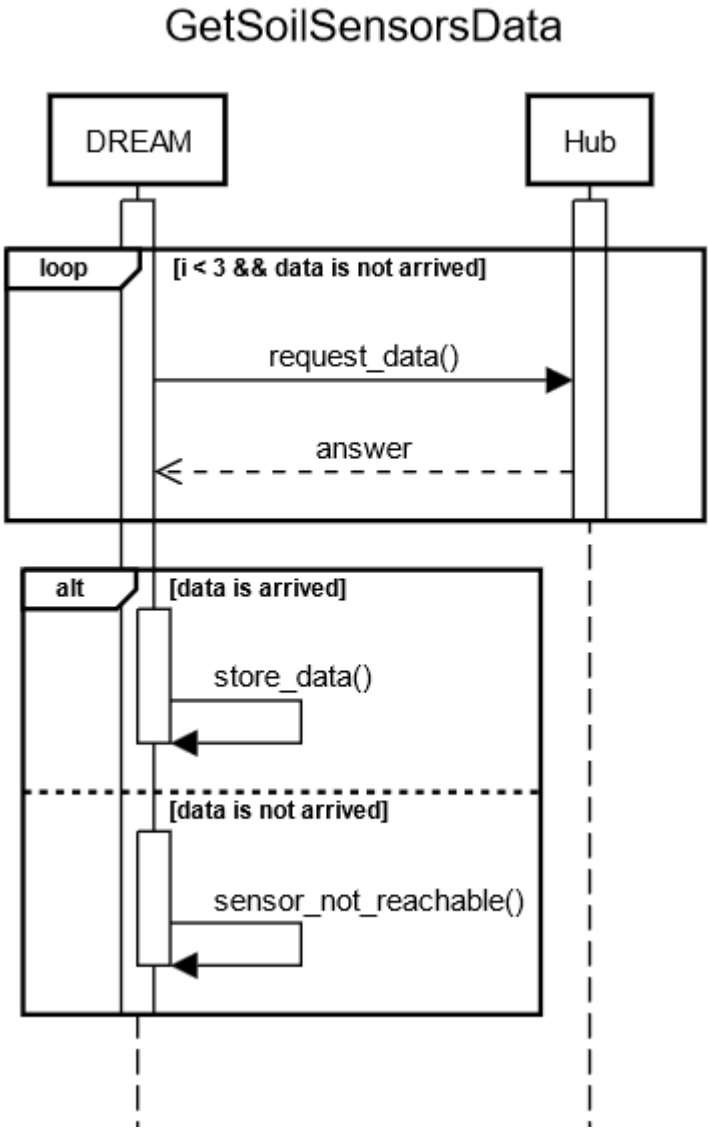

Figure 22: Sequence diagram for the GetSoilSensorsData use case.

Table 20: Use case 14: GetWaterIrrigationSystemData

USE CASE 14	
Name	GetWaterIrrigationSystemData
Actor	WaterIrrigationSystem
Entry condition	It is 00:00. [20]
Event flow	1. `DREAM` requests the WaterIrrigationSystem for the data relative to the day which has just ended. 2. The WaterIrrigationSystem replies with a list of <deviceId, waterUsedAmount> items, one for each water irrigation device managed by the WaterIrrigationSystem.
Exit condition	`DREAM` will send the next request to the WaterIrrigationSystem in 24 hours.
Exceptions	• In case the WaterIrrigationSystem is not reachable, `DREAM` retries to contact it 3 times, with an interval of 30 minutes between them; if after the third try the WaterIrrigationSystem has not been contacted successfully yet, `DREAM` will contact the next day at 00:00.
Special requirements	-

[20] Time referred to the UTC+5:30 time zone

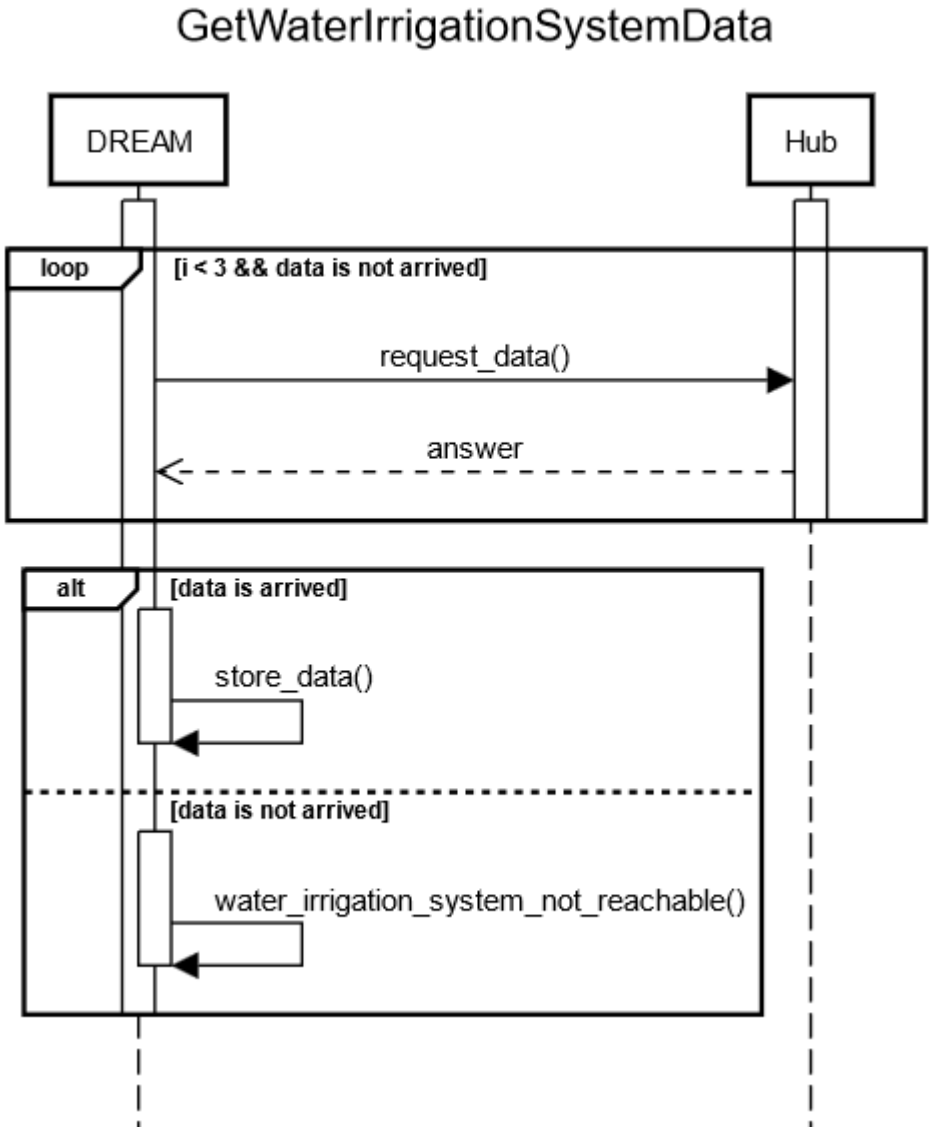

Figure 23: Sequence diagram for the GetWaterIrrigationSystemData use case

Table 21: Use case 15: VisualizeBestAndWorstFarmers

USE CASE 15	
Name	VisualizeBestAndWorstFarmers
Actor	Policy Maker, TSDPSSystem
Entry condition	The policy maker is logged in the system and wants to visualize best and worst performing farmers according to some certain criteria.

Event flow	1. From the homepage, the policy maker accesses the Farms section. 2. DREAM shows: • a form for selecting one or more criteria to rank the farmers; • a form for (optionally) inserting some weights associated to these criteria; • a form for (optionally) inserting the number N of best farmers and the number M of worst farmers to display according to the selected criterion; • a form for (optionally) selecting the area of the farmers to consider; • a form for (optionally) selecting the time period to consider for ranking the farmers. 3. The policy maker selects one or more criteria to rank the farmers among the list of available ones. 4. Optionally, the policy maker inserts some weights associated to the selected criteria. 5. Optionally, the policy maker inserts the number N of best farmers to display. 6. Optionally, the policy maker inserts the number M of worst farmers to display. 7. Optionally, the policy maker selects the area where the farms to rank should be located. 8. Optionally, the policy maker inserts the start and end date of the period to consider for ranking the farmers. 9. The policy maker clicks on the "Rank" button. 10. Depending on the criteria selected, DREAM may fetch from the TSDPSSystem some data from the weather reports (view UC23 for more details)

Exit condition	If no limits on the number of farmers to display were inserted, `DREAM` shows a list of all farmers (belonging to the selected area, if there was one) ordered according to the selected criteria, together with a "Mark As Best-Performing" button for each farmer in the first half of the list and a "Mark As Worst-Performing" button in the second half of the list. If limits on the farmers to display were specified, `DREAM` shows the list of the first N farmers and the list of the last M farmers according to the selected criteria (and belonging to the selected area, if there was one). For each farmer in the first list, there is a "Mark As Best-Performing" button; for each farmer in the latter list, there is a "Mark As Worst-Performing" button. In both cases, if farmer was already marked as best- or worst-performing, the "Mark As Best-Performing" and "Mark As Worst-Performing" are not present, and there is a flag that shows that they are already marked, and a button "Unmark". Moreover, for each farmer there is a "FarmDetails" button. By clicking on it, the policy maker can see a table with monthly seeded, planted, watered, fertilized and harvested amounts for each crop grown in the farm.
Exceptions	• If there is no data in the system about farmers that respect the constraints selected by the policy maker (area and time period), a warning message is displayed.
Special requirements	• After the policy maker clicks on the "Rank" button, the lists should be shown in at most 10 seconds.

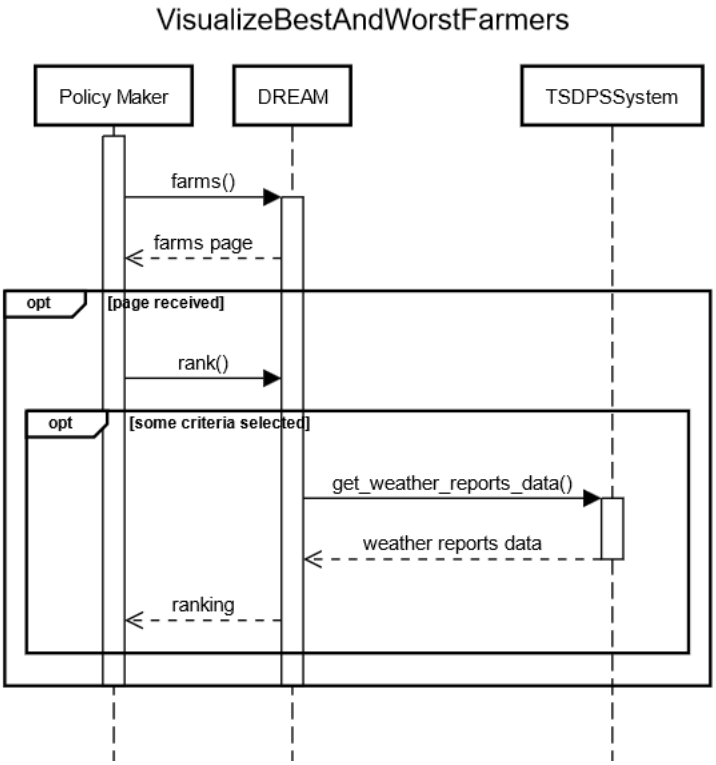

Figure 24: Sequence diagram for the VisualizeBestAndWorstFarmers use case.

OBS.: The use case for the agronomist is almost equivalent to this one; the only difference is that an agronomist is not allowed to choose the area of the farmers to consider, as the farmers he/she can visualize are only the ones located in the area the agronomist is assigned to (event 7 not present).

Table 22: Use case 16: MarkBestPerformingFarmer

USE CASE 16	
Name	MarkBestPerformingFarmer
Actor	User (Policy Maker or Agronomist), Farmer
Entry condition	The user is logged in the system and is visualizing a list of farmers ranked according to some criteria he/she has chosen. He/she wants to mark a certain farmer as best-performing.
Event flow	1. The user clicks on the "Mark As Best-Performing" button related to a certain farmer. 2. `DREAM` shows the user a confirm of having marked the farmer as best-performing. 3. `DREAM` sends the farmer a notification about the fact he/she was selected as best-performing
Exit condition	When policy makers and agronomists of the area of the farmer will visualize the farmer in a list in the "Farms" section, he/she will be shown as best-performing. If the farmer was never marked as best-performing, the homepage of the farmer now displays a "My Replies" section, as other farmers are able to ask him/her questions.
Exceptions	-

Special requirements	• After the policy maker clicks on the "Mark As Best-Performing" button, the confirm should be shown and the notification should be sent in less than 10 seconds.

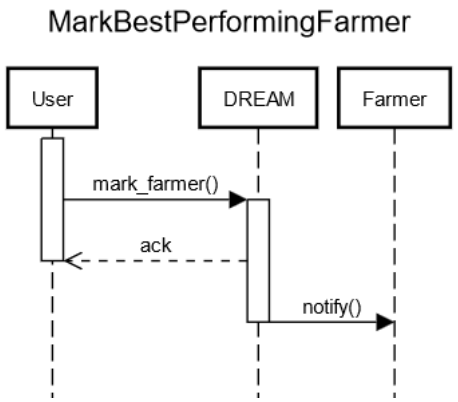

Figure 25: Sequence diagram for the MarkBestPerformingFarmer use case

OBS.: the use case MarkWorstPerformingFarmer is exactly the same.

Table 23: Use case 17: UnmarkBestPerformingFarmer

USE CASE 17	
Name	UnmarkBestPerformingFarmer
Actor	User (Policy Maker or Agronomist), Farmer
Entry condition	The user is logged in the system and is visualizing a list of farmers ranked according to some criteria he/she has chosen. He/she wants to unmark a best-performing farmer.
Event flow	1. The user clicks on the "Unmark" button related to a best-performing farmer. 2. `DREAM` shows the user a confirm of having unmarked the farmer. 3. `DREAM` sends the farmer a notification about the fact he/she has been unmarked.
Exit condition	When policy makers and agronomists of the area of the farmer will visualize the farmer in a list in the "Farms" section, he/she will no more be shown as best-performing. Other farmers are not able anymore to send questions to the farmer - the My Replies section in the homepage of the farmer stays there, to display past replies.
Exceptions	-

Special requirements	• After the policy maker clicks on the "Unmark" button, the confirm should be shown and the notification should be sent in less than 10 seconds.

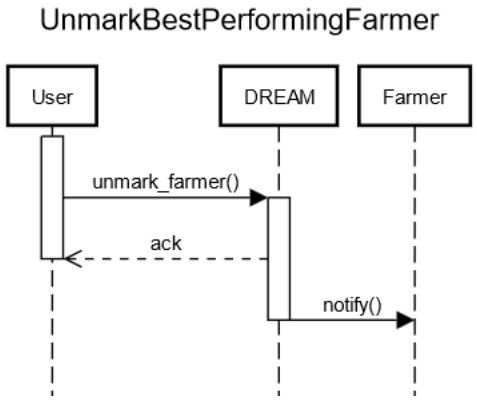

Figure 26: Sequence diagram for the UnmarkBestPerformingFarmer use case.

OBS.: the use case UnmarkWorstPerformingFarmer is exactly the same.

Table 24: Use case 18: AnalyzeImpactOfInitiative

USE CASE 18	
Name	AnalyzeImpactOfInitiative
Actor	Policy Maker, TSDPSSystem
Entry condition	The policy maker is logged in the system and wants to analyze the impact of an initiative - i.e., visit to the farm or reply to a question - on the production of a certain farmer.
Event flow	1. From the homepage, the policy maker accesses the "Initiatives Analysis" section. 2. `DREAM` shows a list of initiatives - either visit to a farm or reply to a question of a farmer -, and some input fields to filter the initiatives according to the type (visit or reply), the date, the area of the farmer, the agronomist or best-performing farmer involved. 3. The policy maker selects one of the initiatives. 4. `DREAM` fetches weather reports data from the TSDPS system
Exit condition	`DREAM` shows details about the initiative and indicators comparing the farmer production data 30 days before and 30 days after the initiative, such as percent difference between the production volume or percent difference between the amount harvested over the amount seeded for each crop, resilience to adverse meteorological events. Moreover, there are two input fields to change the time scope for computing the indicators.

Exceptions	-
Special requirements	• After the policy maker selects the initiative, the result should be shown in less than 10 seconds.

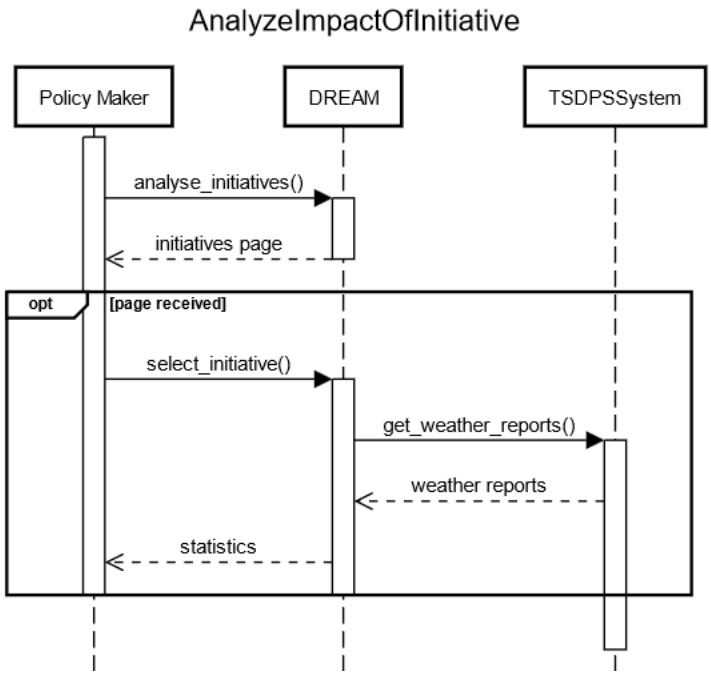

Figure 27: Sequence diagram for the AnalyzeImpactOfInitiative use case.

Table 25: Use case 19: VisualizeWeatherForecasts

USE CASE 19	
Name	VisualizeWeatherForecasts
Actor	User (either Farmer or Agronomist), TSDPSSystem
Entry condition	The user is logged in the system and wants to visualize the weather forecast for the area of his/her farm (if he/she is a farmer) or the area he/she is assigned to (if he/she is an agronomist).
Event flow	1. From the homepage, the user accesses the WeatherForecasts section. 2. `DREAM` shows an input field for inserting the date for which the forecasts should be shown, and if the user is an agronomist, the list of the mandals in the area he/she is responsible of. 3. If the user is an agronomist, he/she selects the mandal in his area for which he/she wants to visualize the forecasts. 4. The user selects the date for which he/she wants to visualize the forecasts. 5. `DREAM` fetches from the TSDPSSystem the weather forecast for the selected date and for the chosen mandal (if the user is an agronomist) or for the mandal of the farmer's farm (if the user is a farmer) (view UC23 for more details).
Exit condition	`DREAM` shows a table containing the weather forecasts for the chosen date and mandal.

Exceptions	• If the Telangana government website is currently unreachable, an error message is shown to the user. • If no weather data is available for the chosen date, an error message is shown to the user.
Special requirements	• The weather forecasts should be shown in less than 10 seconds.

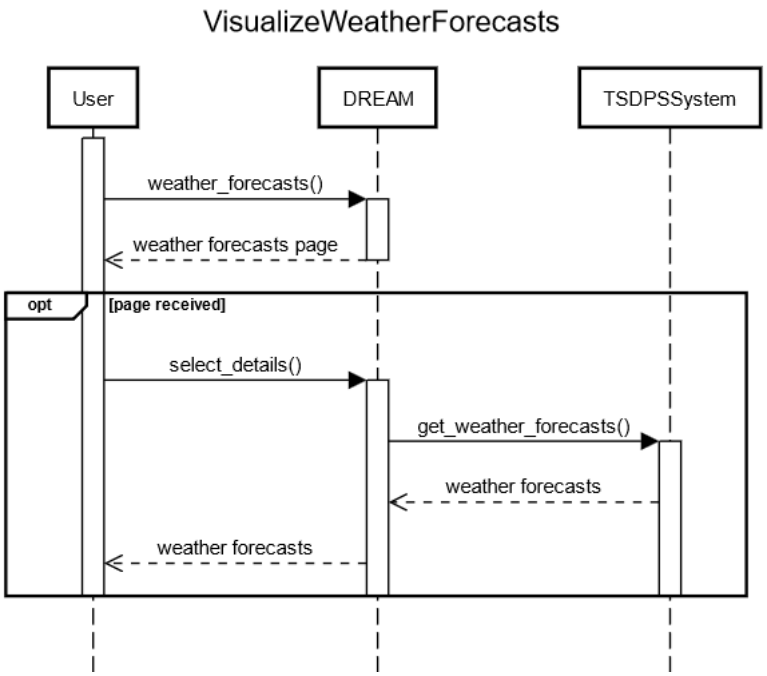

Figure 28: Sequence diagram for the VisualizeWeatherForecasts use case.

Table 26: Use case 20: VisualizeFarmerSuggestions

USE CASE 20	
Name	VisualizeFarmerSuggestions
Actor	Farmer, TSDPSSystem
Entry condition	The farmer is logged into the system and wants to check if there are any suggestions for him/her.
Event flow	1. From the homepage, the farmer accesses the Suggestions section; 2. `DREAM` retrieves weather forecasts and reports necessary for computing the suggestions from the TSDPSSystem (view UC23 for more details). 3. `DREAM` shows a list of suggestions for the farmer.
Exit condition	The farmer views the list of suggestions.
Exceptions	• If there are no suggestions for the farmer, a message is shown.
Special requirements	The farmer should be shown the suggestions in less than 20 seconds.

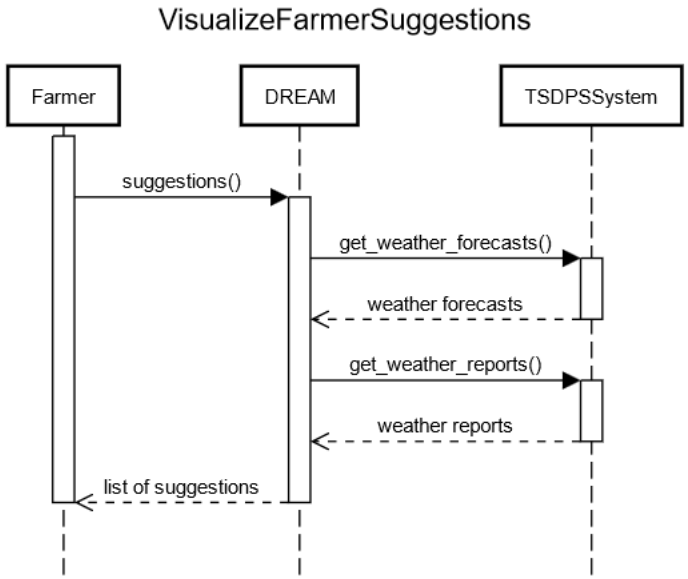

Figure 29: Sequence diagram for the VisualizeFarmerSuggestions use case.

Table 27: Use case 21: GetWeatherForecasts

USE CASE 21	
Name	GetWeatherForecasts
Actor	TSDPSSystem
Entry condition	Either (a) a user requests weather data; or (b) `DREAM` needs to access weather forecasts for computing suggestions for some farmer.
Event flow	1. `DREAM` requests the TSDPSSystem the needed weather forecast 2. The TSDPSSystem replies with the required weather forecasts
Exit condition	The weather forecast are available to the `DREAM` system.
Exceptions	• If the TSDPSSystem is currently unreachable, the process is aborted and an error is returned to the user.
Special requirements	-

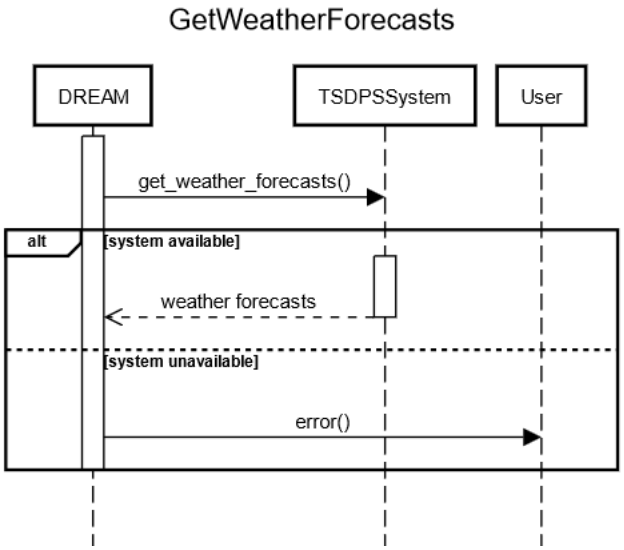

Figure 30: Sequence diagram for the GetWeatherForecasts use case.

The interaction is analogous for retrieving weather reports.

Table 28: Use case 22: RequestToExpert

USE CASE 22	
Name	RequestToExpert
Actor	Farmer, Expert (either an Agronomist or a best-performing Farmer)
Entry condition	The Farmer is logged in the DREAM system and wants to ask a question to an Agronomist or to another Farmer.
Event flow	1. From the homepage, the Farmer enters the "My Requests" section. 2. The Farmer clicks on the "New Request" button and selects the type of expert he wants to ask his question. 3. DREAM responds showing the list of experts of that kind available. 4. The Farmer selects from the list the expert to contact, then inserts in a form all the data of the request (title, description) and sends the request. 5. DREAM forwards the request to the chosen expert – that is, it inserts the request in the list of the new requests issued to the expert. 6. DREAM shows a confirmation of having sent the request, containing the name of the expert, the title and the description of the request.

Exit condition	The request is present in the list in the "My Requests" section of the Farmer view, shown as a closed request, and selecting it from the list the Farmer can see the name of the Expert, the timestamp, title and text of the request, the timestamp and text of the reply, and the feedback message.
Exceptions	• If the device of the Expert is not connected to the Internet when he should receive the request, then `DREAM` shows the details of the request and a message saying that it will be sent as soon as the device will be connected to the Internet. When the disconnected device reconnects, the message is sent.
Special requirements	• The messages between the Farmer and the Expert are delivered in less than one minute.

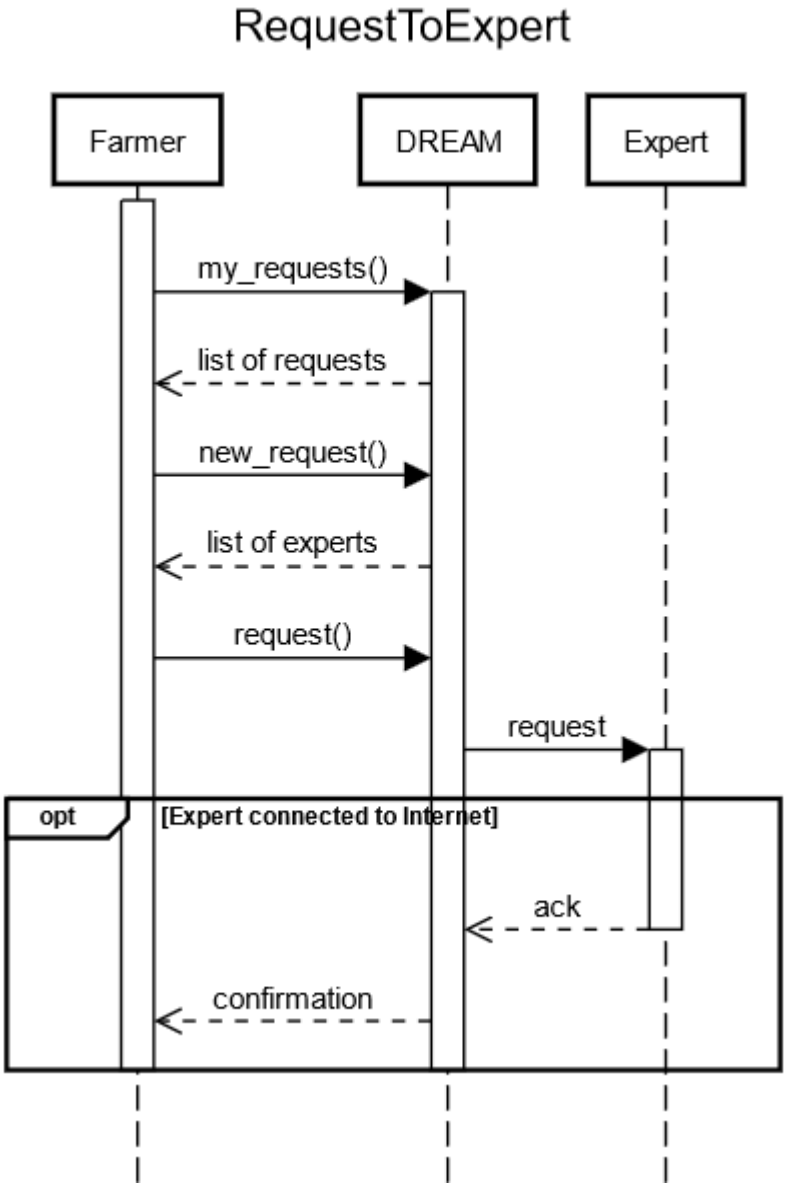

Figure 31: Sequence diagram for the RequestToExpert use case.

Table 29: Use case 23: ExpertResponse

USE CASE 23	
Name	ExpertResponse
Actor	Farmer, Expert (either an Agronomist or a best-performing Farmer)
Entry condition	The Expert is logged in the DREAM system.
Event flow	1. From the homepage, the Expert enters the "My Responses" section. 2. The Expert selects one request from the list of non-answered requests. 3. DREAM shows the name of the Farmer, the timestamp, the title and the description of the request, and a form to insert the reply. 4. The Experts fills the form with the response and clicks on the "Send Reply" button. 5. DREAM forwards the reply to the Farmer – that is, it associates the reply to the request of the Farmer. 6. DREAM shows a confirmation of having sent the reply, containing the name of the Farmer, the title and the text of the request, and the text of the reply.
Exit condition	The request is present in the list in the "My Responses" section of the Expert, shown as a closed request.

Exceptions	• If the device of the Farmer is not connected to Internet when the Expert sends the response, then `DREAM` shows a message saying that it will be sent as soon as the device will be connected to Internet. When the disconnected device reconnects, the message is sent.
Special requirements	• The messages between the Farmer and the Expert are delivered in less than one minute.

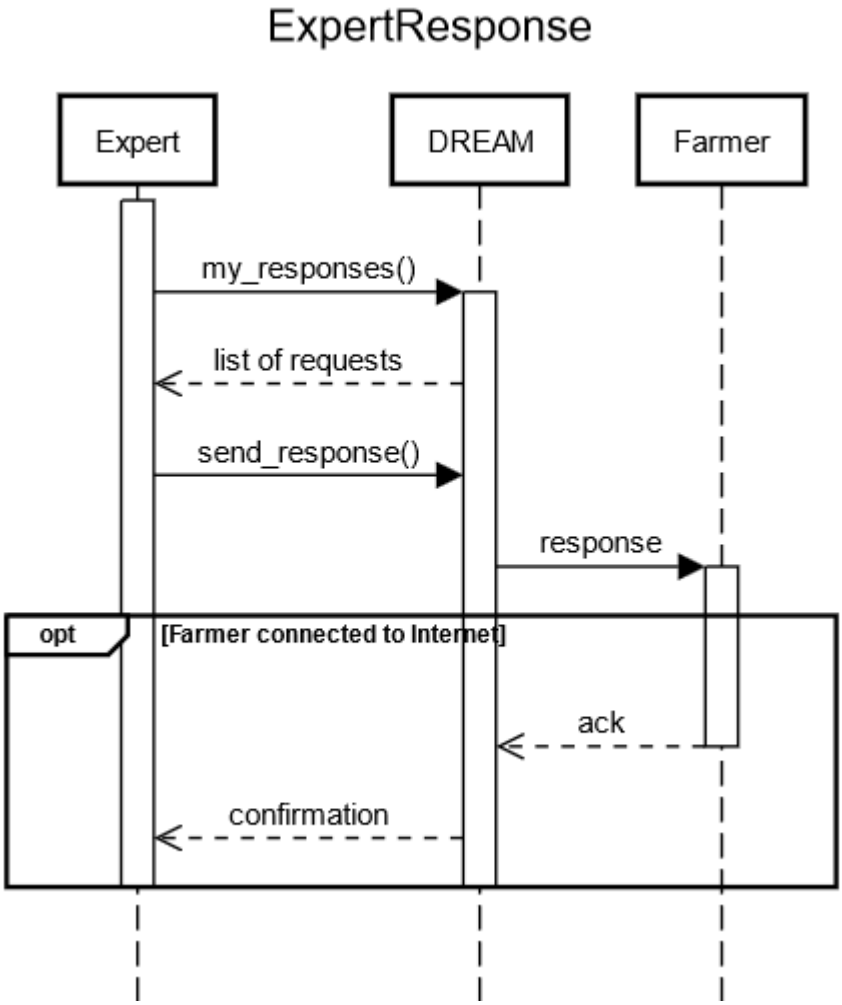

Figure 32: Sequence diagram for the ExpertResponse use case.

Table 30: Use case 24: VisualiseResponse

USE CASE 24	
Name	VisualiseResponse
Actor	Farmer
Entry condition	The Farmer is logged in the `DREAM` system.
Event flow	1. From the homepage, the Farmer enters the "My Requests" section. 2. `DREAM` shows a list of the requests issued by the Farmer, where the ones which the response hasn't been read yet are highlighted. 3. The Farmer selects the request the Expert has replied to. 4. `DREAM` shows the timestamp, title and text of the request, the name of the Expert who has replied, and the timestamp and text of the reply, and a text input to provide a feedback to the Expert. 5. The Farmer writes down his feedback to the Expert. 6. The Farmer clicks on the "Send" button. 7. `DREAM` sends the feedback to the expert – that is, it associates the feedback to the request-reply thread.
Exit condition	The farmer visualises the response of the expert and the expert has received a feedback from the farmer.

Exceptions	-
Special requirements	-

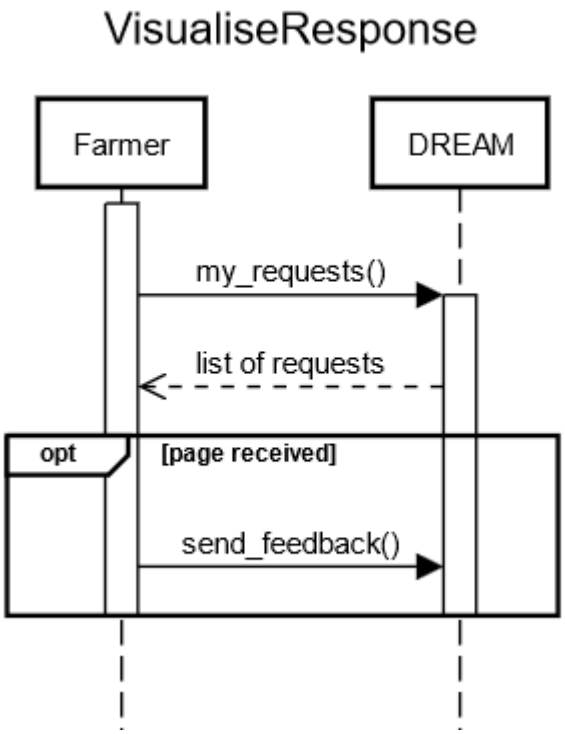

Figure 33: Sequence diagram for the VisualiseResponse use case.

Table 31: Use case 25: CreateFarmersForumThread

USE CASE 25	
Name	CreateFarmersForumThread
Actor	Farmer
Entry condition	The Farmer is logged in the `DREAM` system, and visualizes the home page.
Event flow	1. From the homepage, the Farmer enters the Forum section. 2. `DREAM` replies showing a list of the already present discussion threads and a "New Thread" button. 3. The Farmer clicks on the "New Thread" button. 4. `DREAM` shows a text input area to insert the name of the "New Thread", and a "Create" button. 5. The Farmer inserts the title of the discussion thread. 6. The Farmer clicks on the "New Thread" button.
Exit condition	A new thread of discussion has been created, which can be seen by all farmers when entering the "Forum" section. The Farmer visualizes the list of messages in the thread – which is empty – together with a "New Post" button.

Exceptions	• If the title inserted by the Farmer is not valid (empty or already present), an error message is shown and the new thread is not created.

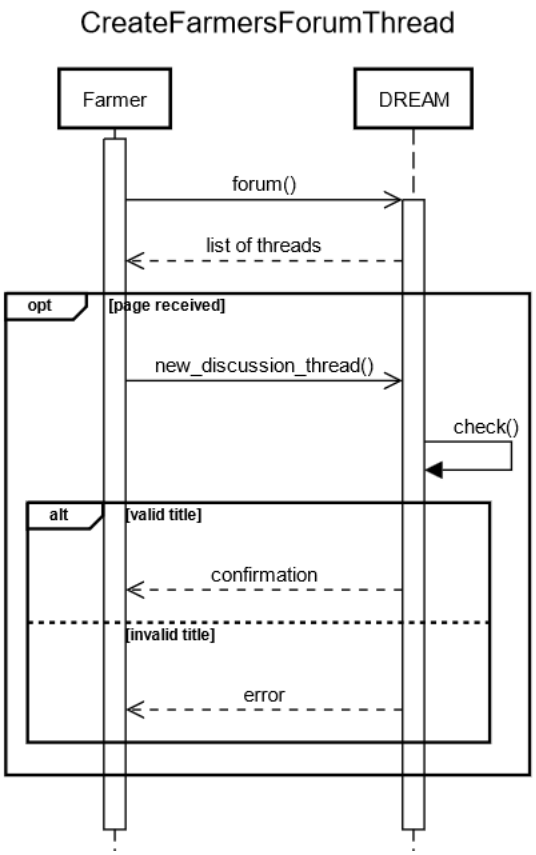

Figure 34: Sequence diagram for the CreateFarmersForumThread use case.

Table 32: Use case 26: WritePostOnFarmersForum

USE CASE 26	
Name	WritePostOnFarmersForum
Actor	Farmer
Entry condition	The Farmer is logged in the DREAM system, and visualizes the home page.
Event flow	1. From the homepage, the Farmer enters the Forum section. 2. DREAM replies showing a list of the already present discussion threads and a "New Thread" button. 3. The Farmer clicks on one of the already present discussion threads. 4. DREAM shows a list of the posts in the forum, an input text field and a "New Post" button. 5. The Farmer inserts the text of the post. 6. The Farmer clicks on the "New Post" button.
Exit condition	DREAM shows a confirmation of having created the new post, and the post is shown together with the others in the list.
Exceptions	• If the text of the post is empty, DREAM shows an error message and doesn't create any new post.

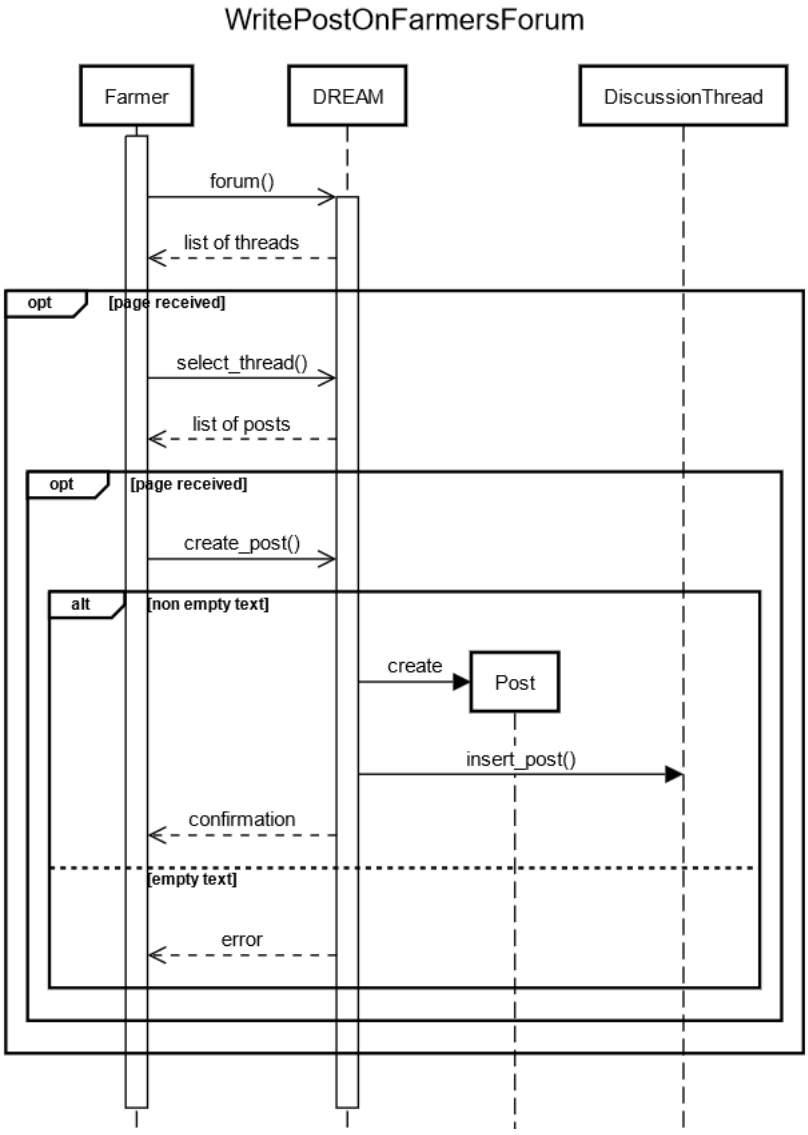

Figure 35: Sequence diagram for the WritePostOnFarmersForum use case.

3.2.5 Mapping between use cases and requirements

This section shows how the DREAM system implements requirements by mapping them to use cases.

Table 33: Mapping between use cases and requirements.

UC1: RegisterFarmer	• R5: DREAM shall allow farmers to insert the location and the area of their plot of land;
UC2: RegisterAgronomist	• R29: DREAM shall allow agronomists to insert the area they are responsible of;
UC3: LogIn	• R48: DREAM shall allow users to log in the system by using an email and a password;
UC4: ViewProductionData	• R49: DREAM shall allow farmers to visualize data inserted by them about their production;

UC5: InsertProductionData	• R6: DREAM shall allow farmers to insert the type of products they grow in a certain area of their farms; • R7: DREAM shall allow farmers to insert data about their sowing activity; • R8: DREAM shall allow farmers to insert data about their planting activity; • R9: DREAM shall allow farmers to insert data about their harvesting activity; • R10: DREAM shall allow farmers to insert data about the irrigation of their plantations;
UC6: InsertProblems	• R11: DREAM shall allow farmers to insert descriptions of problems they face; • R12: DREAM shall allow farmers to request a visit of an agronomist when inserting a problem; • R13: When a farmer requests a visit of an agronomist, DREAM inserts the visit to the farmer into the daily plan of an agronomist assigned to the area of the farmer; • R14: When a farmer requests a visit of an agronomist, if no agronomist is available in the area of the farmer during the seven days after the request (excluding the day when the request is made), DREAM notifies the farmer about this situation.

UC7: AcceptDaily-Plan	• R39: DREAM enforces that every farmer is assigned a visit in the daily plan of an agronomist at least twice a year; • R40: DREAM enforces that no agronomist in a certain area is assigned more than twice of the visits than an agronomist of the same area; • R41: DREAM enforces that the daily plan of agronomists includes more visits to worst-performing farmers than to other ones; • R42: DREAM allows agronomists to accept or modify -i.e.: remove, add or move a visit - the automatically generated daily plan; • R45: DREAM is able to notify farmers involved by a visit of an agronomist;
UC8: ViewDetailsOf-FarmToVisit	• R46: DREAM allows agronomists to visualize data about farmers to visit - such as name, surname and location of the farm - and their production since their last visit to the farm;

UC9: MoveVisitInDailyPlan	• R39: DREAM enforces that every farmer is assigned a visit in the daily plan of an agronomist at least twice a year; • R40: DREAM enforces that no agronomist in a certain area is assigned more than twice of the visits than an agronomist of the same area; • R41: DREAM enforces that the daily plan of agronomists includes more visits to worst-performing farmers than to other ones; • R42: DREAM allows agronomists to accept or modify -i.e.: remove, add or move a visit - the automatically generated daily plan; • R45: DREAM is able to notify farmers involved by a visit of an agronomist;
UC10: DeleteVisitFromDailyPlan	• R39: DREAM enforces that every farmer is assigned a visit in the daily plan of an agronomist at least twice a year; • R40: DREAM enforces that no agronomist in a certain area is assigned more than twice of the visits than an agronomist of the same area; • R41: DREAM enforces that the daily plan of agronomists includes more visits to worst-performing farmers than to other ones; • R42: DREAM allows agronomists to accept or modify -i.e.: remove, add or move a visit - the automatically generated daily plan; • R45: DREAM is able to notify farmers involved by a visit of an agronomist;

UC11: AddVisitToDailyPlan	• R39: DREAM enforces that every farmer is assigned a visit in the daily plan of an agronomist at least twice a year; • R40: DREAM enforces that no agronomist in a certain area is assigned more than twice of the visits than an agronomist of the same area; • R41: DREAM enforces that the daily plan of agronomists includes more visits to worst-performing farmers than to other ones; • R42: DREAM allows agronomists to accept or modify -i.e.: remove, add or move a visit - the automatically generated daily plan; • R45: DREAM is able to notify farmers involved by a visit of an agronomist;
UC12: ConfirmDailyPlan	• R4: DREAM shall allow agronomists to insert feedback about the farmers they visited; • R43: DREAM allows agronomists to confirm a daily plan at the end of the working day; • R44: DREAM allows agronomists to specify deviations -i.e.: remove, add or move a visit- from a daily plan at the end of the working day;
UC13: GetSoilSensorsData	• R1: DREAM collects soil moisture data from the soil moisture sensors;

UC14: GetWaterIrrigationSystemData	• R3: DREAM collects watering data from the water irrigation system;
UC15: VisualizeBestAndWorstFarmers	• R15: DREAM can use different criteria (namely, productivity in terms of harvested quantity over sowed quantity, productivity in presence of adverse meteorological events, productivity in presence of drought, and so on) and/or combinations of them to rank farmers; • R16: DREAM shall allow policy makers and agronomists to choose the criteria to use for ranking farmers and the associated weights (for combinations of them); • R17: DREAM shall allow policy makers and agronomists to choose the time period and the area to consider for ranking the farmers;
UC16: MarkBestPerformingFarmer	• R18: DREAM allows policy makers and agronomists to mark/unmark a farmer as best-performing or worst-performing;
UC17: UnmarkBestPerformingFarmer	• R18: DREAM allows policy makers and agronomists to mark/unmark a farmer as best-performing or worst-performing;

UC18: AnalyzeImpactOfInitiative	• R19: DREAM can compare different time periods of a farmer work according to various production criteria (e.g. production volume, fertilizers adopted, fraction of harvested plants over sowed ones, etc); • R20: DREAM can compare different time periods of a farmer work with respect to environmental factors (e.g. weather reports data, soil moisture, ...); • R21: DREAM allows policy makers to choose an initiative taken by an agronomist or a farmer to help a farmer - i.e., visit to the farm or reply to a question - and two time periods of the farmer to compare the two time periods; • R22: DREAM can show the impact of a certain initiative taken by an agronomist or a farmer to help a farmer - i.e., visit to the farm or reply to a question - during a certain time period.
UC19: VisualizeWeatherForecasts	• R27: DREAM is able to connect to the Telangana government website to fetch forecasts for the chosen date; • R28: DREAM can show weather forecast data for a certain location and date; • R30: DREAM shall allow agronomists to choose the date of the weather forecasts to visualize;

UC20: VisualizeFarmerSuggestions	• R23: DREAM is able to find correlations among environmental factors, fertilisers adopted and crops planted with the volume of production of the farmers; • R24: DREAM is able to send suggestions to farmers about fertilizers to use; • R25: DREAM is able to send suggestions to farmers about crops to plant;
UC21: GetWeatherForecasts	• R27: DREAM is able to connect to the Telangana government website to fetch forecasts for the chosen date;
UC22: RequestToExpert	• R31: DREAM allows farmers to choose the (best-performing) farmer or agronomist (assigned to his area) who to issue a request; • R32: DREAM allows the farmers to send a request to a best-performing farmer or agronomist.
UC23: ExpertResponse	• R33: DREAM notifies (best-performing) farmers and agronomists of the requests of help from other farmers; • R34: DREAM allows best-performing farmers and agronomists to insert a message of response to the farmers who have made a request for help to them.

UC24: VisualizeResponse	• R35: DREAM sends the response to the farmer who issued the corresponding request.
UC25: CreateFarmersForumThread	• R36: DREAM allows farmers to initiate a new thread in the forum.
UC26: WritePostOnFarmersForum	• R37: DREAM allows the farmers to add a post to a thread in the forum;

3.3 Performance Requirements

This section is dedicated to "specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole"[21]. Static numerical requirements:

- according to the estimates[22], Telangana has a population of 36 millions of people and a total number of farm holdings equal to 55.54 lakhs, namely more or less 5 555 400 people in Telangana are farmers; assuming that all of them are potential users of `DREAM`, it must support at least 5.7 million non-simultaneously connected terminals, since also the agronomists and policy makers shall be allowed to access the system;
- given the huge number of (possible) users and the purpose of the system, `DREAM` must be able to support at least 10% of the total number of users simultaneously; this requirement is more important than speed performances;
- all the weather reports and forecasts are not directly stored on `DREAM`; nevertheless, `DREAM` must store all the sensors and water irrigation system data as well as the users data and the forums data. We can assume that the system stores 1KB of personal information for each user:

  $$usersData = 5.7 * 10^6 users * 1KB \approx 5.7GB$$

  Assuming an average of 1000 posts a day, the data required for 1 year is (assumption: 1KB for each post, no images allowed):

  $$forumData = 1KB * 1000 posts/day * 365 days/year \approx 365MB$$

  Plus 32 bytes for each sensor measurement (16 bytes to recognize the sensor, 8 bytes for the timestamp and 8 bytes for the soil moisture value; we estimate 10 000 humidity sensors in Telangana whose data is retrieved 3 times a day):

  $$sensorsData = 32B * 10000 sensors * 3 measurements/day * 365 days/year \approx 350MB$$

  Plus 32B additional bytes for each farmer because of the water irrigation system:

  $$waterIrrigationSystemData = 32B * 12 measurements/year * 5.5 * 10^6 farmers \approx 2GB$$

  Plus 1KB for each production report stored in `DREAM` (assuming that each farmer uploads the production data once for each month):

  $$productionData = 1KB * 12 reports/year * 5.5 * 10^6 farmers \approx 66GB$$

[21] IEEE 29148-2018 Requirements engineering, section 9.6.14
[22] professor Jayashankar Telangana state agricultural university

Plus some additional data for the daily plans, that we can neglect because of the small number of agronomists with regards to the number of farmers.
This means that the amount of information to be effectively stored must be in the rank of:

$$totalData = \sum_{i=1}^{5} data[i] \approx 74GB$$

In this calculation is not considered all the space required to perform analysis over the stored data and the data accessed over the Telangana's government website (namely, weather reports and forecasts). Therefore, we can assume that a database of $100GB$ can suffice for the first year.

Dynamic numerical requirements:

- all the transactions must be processed in less than 3 seconds (as said before, `DREAM` does not require strict temporal requirements);
- the number of transactions the system has to process during peak workload conditions is in the order of 10^4.

3.4 Design Constraints

In this section we "specify constraints on the system design imposed by external standards, regulatory requirements or project limitations"[23].

3.4.1 Standard compliance

`DREAM` stores personal data of the users, like name, surname, job, location of the farm (if he is a farmer) ... . Therefore, this data must be treated according to the privacy regulations in India.

3.4.2 Hardware limitation

Since `DREAM` is a webapp, every device with a browser recent enough can be used to access the system. `DREAM` does not impose any constraint on the sensors and the water irrigation system, as long as the central hub uses a standardized protocol for communication.

3.4.3 Any other constraint

No any other specific constraint is required.

3.5 Software System Attributes

In this section a list of required attributes of the system is provided.

[23] IEEE 29148-2018 Requirements engineering, section 9.6.16

3.5.1 Reliability

DREAM is not a critical system, therefore some limited failures cannot create big problems. For example, an error when accessing the Telangana's website or a failure during the generation of the daily plan can be easily solved just by re-starting the process.

3.5.2 Availability

The system is not critical, as already said, therefore some periods of inactivity are allowed. Because all the tasks it performs are not critical, small periods of inactivity do not create many problems. For instance, when a farmer needs to ask for help, he could make the request even the following day. Another example is the access to the humidity sensors: data of today could be accessed even tomorrow since they are stored in Telangana's website as well. A final example is about the proposal of a schedule plan; agronomists can wait some time before retrieving it. We can impose that DREAM must not be unavailable for more than 1 day per month.

3.5.3 Security

The only information that needs to be protected is the personal data of the users. Therefore, DREAM has to assure data privacy. This can be done by ensuring that all connections are established securely via HTTPS.

3.5.4 Maintainability

The system must be designed in order to facilitate maintainability. Every functionality implemented must be well documented. The system must be decomposed in a certain number of modules to limit the complexity and to make the development process more efficient. The main goal is that each module is coherent and can be (internally) modified without affecting other modules and without changing its interface.

3.5.5 Portability

Since DREAM is a webapp, it does not have strict requirements on the underlying operating system or hardware used. The website will be developed using an interpreted language with a widely available interpreter.
Furthermore, DREAM must have a responsive interface (namely, it must scale properly to different devices' sizes).

4 Formal analysis using Alloy

This section presents the formal specification of some requirements through Alloy. See the comments in the code for an explanation.

```
open util/integer
open util/ordering [DateTime] // adds ordering to DateTime objects

/**** SIGNATURES ****/
sig DateTime {}
abstract sig User {}
sig Farmer extends User {
        farm: one Farm
}
sig Agronomist extends User {
        // For the purposes of this Alloy specification,
        // we assume all Agronomists have already selected their area.
        area: one Area
}
sig Farm {
        area: one Area,
}
sig DailyPlan {
        agronomist: one Agronomist,
        fromDateTime: one DateTime,
        toDateTime: one DateTime
}
sig FarmVisit {
        dailyPlan: one DailyPlan,
        farm: one Farm,
        dateTime: one DateTime
}
sig Area {}
sig ProductionData {
        farm: one Farm,
        fromDateTime: one DateTime,
        toDateTime: one DateTime,
        volume: one Int
}
sig ProductionIssue {
        productionData: one ProductionData
}

/**** FUNCTIONS ****/
fun FarmVisits[fx: Farm]: set FarmVisit {
        { fv: FarmVisit | fv.farm = fx }
```

```
}

fun FarmerVisits[f: Farmer]: set FarmVisit {
        FarmVisits[f.farm]
}

fun FarmVisitAgronomist[fv: FarmVisit]: one Agronomist {
        fv.dailyPlan.agronomist
}

fun AgronomistVisits[a: Agronomist]: set FarmVisit {
        { fv: FarmVisit | FarmVisitAgronomist[fv] = a }
}

fun FarmerIssues[f: Farmer]: set ProductionIssue {
        { i: ProductionIssue | i.productionData.farm = f.farm }
}

fun AgronomistDailyPlans[a: Agronomist]: set DailyPlan {
        { p: DailyPlan | p.agronomist = a }
}

/**** FACTS ****/
// Every farm has exactly one farmer
fact { all fx: Farm | one f: Farmer | f.farm = fx }

// All areas have at least one agronomist
fact { all a: Area | some agr: Agronomist | agr.area = a }

// All areas have at least one farmer
fact { all a: Area | some f: Farmer | f.farm.area = a }

// All farms are visited at least twice
// (Note: For the purposes of this specification, we assume all visits
// in the generated world happen in the same year)
fact { all fx: Farm | (let v = FarmVisits[fx] | #v >= 2) }

// Every Daily Plan has at least one visit
fact { all p: DailyPlan | some v: FarmVisit | v.dailyPlan = p }

// Daily Plan dates are consistent
fact { all p: DailyPlan | lt[p.fromDateTime, p.toDateTime] }

// Farm Visit dates are consistent with their daily plan
fact {
    all fv: FarmVisit | (
```

```
        gte[fv.dateTime, fv.dailyPlan.fromDateTime] and
        lte[fv.dateTime, fv.dailyPlan.toDateTime]
    )
}

// Agronomists only visit farms in their area
fact { all fv: FarmVisit | fv.farm.area = FarmVisitAgronomist[fv].area }

// Farmers who had more production issues are visited more often
// (relative to other farmers in their area)
fact {
    all disj f1: Farmer, f2: Farmer |
        (f1.farm.area = f2.farm.area and #FarmerIssues[f1] > #FarmerIssues[f2])
        implies
        #FarmerVisits[f1] >= #FarmerVisits[f2]
}

// Agronomists in the same area "split" their work,
// i.e no agronomist does more than twice
// the number of visits of any of their colleagues.
fact {
    all disj a1: Agronomist, a2: Agronomist | (
        a1.area = a2.area
        implies
        #AgronomistVisits[a1] <= mul[2, #AgronomistVisits[a2]]
    )
}

// Production volume is nonnegative
fact { all d: ProductionData | d.volume >= 0 }

// Production datetimes are consistent
fact { all d: ProductionData | lt[d.fromDateTime, d.toDateTime] }

/**** PREDICATES (WORLD GENERATION) ****/
pred WorldSimple {
        #Area = 1
        #Farmer = 1
        #Agronomist = 1
}

pred WorldManyAgronomists {
        #Area = 1
        #Farmer = 2
        #Agronomist = 5
        #FarmVisit = 5
```

```
}

pred WorldFarmersIssues {
        #Area = 1
        #Farmer = 2
        #Agronomist = 2
        #ProductionData = 2
        #ProductionIssue = 5
        #FarmVisit = 5
        // Every farmer had at least one issue
        all f: Farmer | #FarmerIssues[f] >= 1
}

/**** ASSERTIONS ****/
// All farmers have a "reference" agronomist
//(i.e. the agronomist they can contact via DREAM)
assert FarmerReferenceAgronomists {
        all f: Farmer | some agr: Agronomist | f.farm.area = agr.area
}

// All agronomists have at least one daily plan
assert AgronomistDailyPlans {
        all a: Agronomist | #AgronomistDailyPlans[a] > 0
}

// All agronomists visit some farms
assert AgronomistFarmVisits {
        all a: Agronomist | #AgronomistVisits[a] > 0
}

/**** EXECUTION ****/
check FarmerReferenceAgronomists for 5
check AgronomistFarmVisits for 5
check AgronomistDailyPlans for 5
run WorldSimple for 5
run WorldManyAgronomists for 10
run WorldFarmersIssues for 5
```

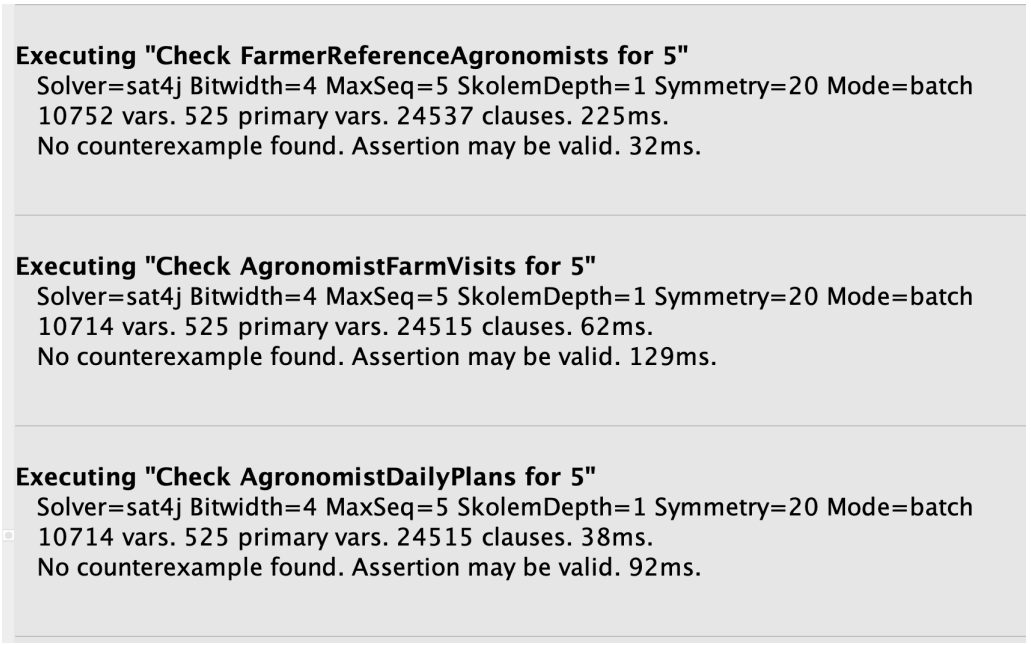

Figure 36: Output of the assertions.

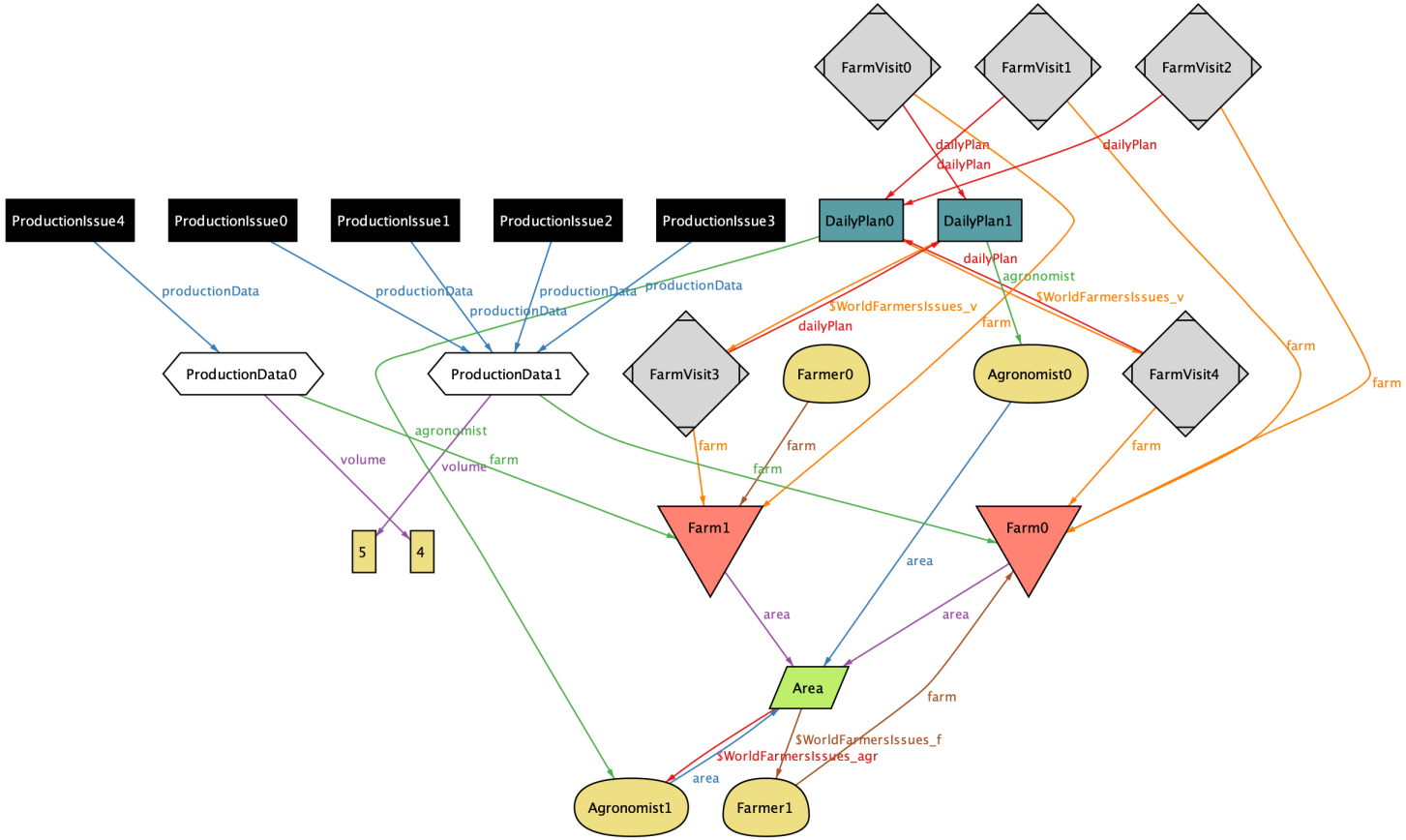

Figure 37: Output of the "WorldFarmersIssues".

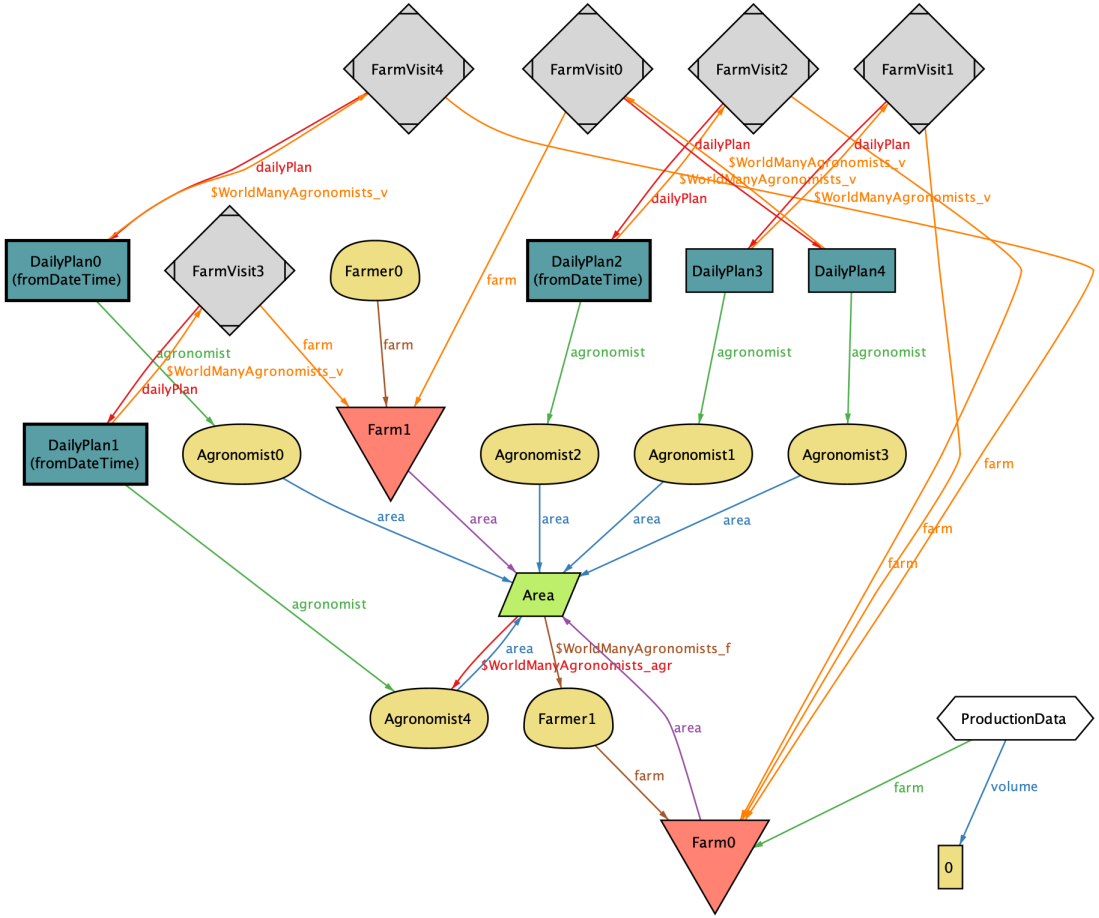

Figure 38: Output of the "WorldManyAgronomists".

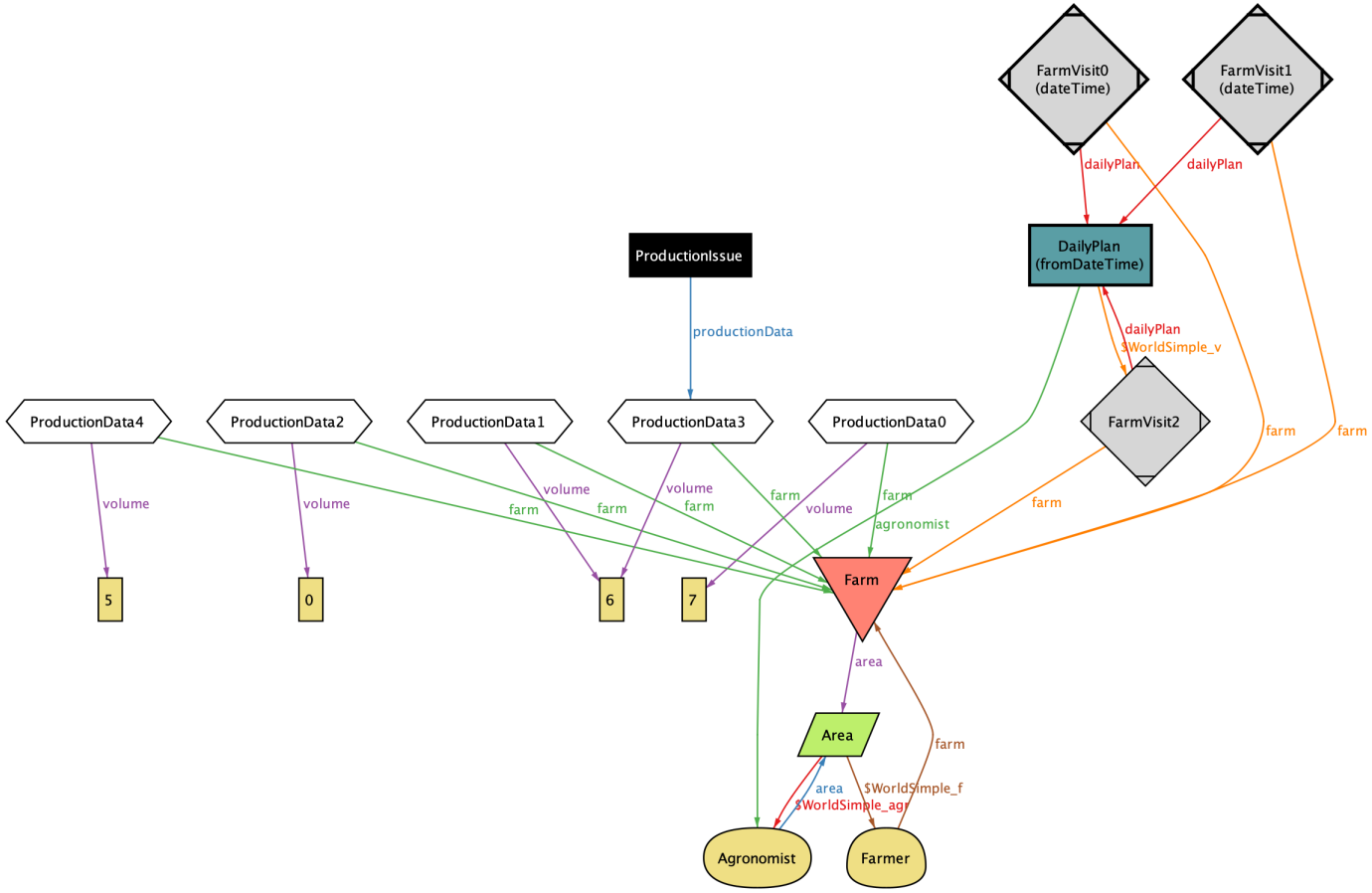

Figure 39: Output of the "WorldSimple".

5 Effort spent

Table 34: The time Christian Grasso has spent on this document.

Effort spent	
Organization[24]	6h
Product perspective	2.30h
Scenarios and use cases	4h
WP SP G R D	4h
Alloy	9h
Review and Mockups	4h
Final recap	2h

Table 35: The time Filippo Lazzati has spent on this document.

Effort spent	
Organization[25]	6h
Overall description	4h
Scenarios and use cases	12.30h

[24] This entry features all the times the group gathered to decide what to do.
[25] This entry features all the times the group gathered to decide what to do.

WP SP G R D	9.30h
Alloy	3h
External interfaces and performance requirements	2.30
Review and Mockups	3.30h
Final recap	2h

Table 36: The time Chiara Magri has spent on this document.

Effort spent	
Organization[26]	6h
Scenarios and use cases	12.30h
Mapping R - UC	5h
WP SP G R D	14h
Alloy	3h
Review	1.30h
Final recap	2h

[26]This entry features all the times the group gathered to decide what to do.

6 References

6.1 Used Tools

- StarUML: to draw UML diagrams;
- draw.io: another tool to draw UML diagrams;
- Alloy: to specify formally some requirements;
- Overleaf: Latex editor;
- Github: to share the files;
- Balsamiq: to design the wireframes.