DREAM project by Federica
Tommasini & Alessandro Verosimile

POLITECNICO
MILANO 1863

Requirement Analysis and Specification Document

Deliverable:	RASD
Title:	Requirement Analysis and Specification Document
Authors:	Federica Tommasini, Alessandro Verosimile
Version:	1.0
Date:	23-December-2021
Download page:	https://github.com/federicatommasini/TommasiniVerosimile.git
Copyright:	Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

Contents

Table of Contents 3

List of Figures 5

List of Tables 5

1 Introduction 7
1.1 Purpose 7
1.1.1 Goals 7
1.2 Scope 8
1.2.1 State of the art 8
1.2.2 World Phenomena 9
1.2.3 Shared Phenomena 9
1.3 Definitions, Acronyms, Abbreviations 12
1.3.1 Definitions 12
1.3.2 Acronyms 12
1.3.3 Abbreviations 12
1.4 Revision history 13
1.5 Reference documents 13
1.6 Document structure 13

2 Overall Description 14
2.1 Product perspective 14
2.1.1 Scenarios 14
2.1.2 Class diagram 16
2.1.3 Statecharts 17
2.2 Product functions 19
2.3 User characteristics 20
2.4 Assumptions, dependencies and constraints 20

3 Specific Requirements 22
3.1 External Interface Requirements 22
3.1.1 User Interfaces 22
3.1.2 Hardware Interfaces 25
3.1.3 Software Interfaces 25
3.1.4 Communication Interfaces 25
3.2 Functional Requirements 25
3.2.1 List of requirements 25
3.2.2 Mapping on goals 27
3.2.3 Use case diagrams 28
3.2.4 Use cases 31
3.2.5 Sequence diagrams 35
3.3 Performance Requirements 41
3.4 Design Constraints 41
3.4.1 Standards compliance 41
3.4.2 Hardware limitations 41
3.5 Software System Attributes 41
3.5.1 Reliability 41
3.5.2 Availability 41

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

3.5.3 Security 41
3.5.4 Maintainability 42
3.5.5 Portability 42

4 Formal Analysis Using Alloy 43
4.1 Code 43
4.2 Results and Generated instances 46
4.2.1 First run 46
4.2.2 Second run 50
4.2.3 Third run 52

5 Effort Spent 53

6 References 54

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

List of Figures

1 Class Diagram. 16
2 Statechart that descibes farmers' interaction with the system. 17
3 Statechart that describes policy makers' interaction with the system. 18
4 Sign in interfaces for farmers (on the left) and policy makers (on the right) of DREAM. . 22
5 Home interface (on the left) and Profile interface (on the right) for a farmer of Dream. . . 23
6 Home interface for a policy maker of DREAM. 24
7 Use case diagram referred to the unregistered farmer. 28
8 Use case diagram referred to the policy maker. 29
9 Use case diagram referred to the application farmer. 30
10 Sequence diagram referred to the 1st use case. 35
11 Sequence diagram referred to the 2nd use case. 36
12 Sequence diagram referred to the 3rd use case. 37
13 Sequence diagram referred to the 4th use case. 37
14 Sequence diagram referred to the 5th use case. 38
15 Sequence diagram referred to the 6th use case. 38
16 Sequence diagram referred to the 7th use case. 39
17 Sequence diagram referred to the 8th use case. 39
18 Sequence diagram referred to the 9th use case. 40
19 Sequence diagram referred to the 10th use case. 40
20 Alloy code. 45
21 Result obtained by running the Alloy model. 46
22 Instance of the world obtained by running the Alloy model. 46
23 Highlighting of the relation between farmers and product cultivated. 47
24 Highlighting of the relation between the product and the correspondent ranking. 47
25 Highlighting of the relation between the rankings and the respective ranking positions. . 48
26 Highlighting of the relation between the ranking positions and the farmers to which they are referred to. 48
27 Highlighting of the relation between the ranking positions and the correspondent numbers. 49
28 Highlighting of the relation between the farmers and the policy maker who evaluates them. 49
29 Highlighting of the relation between the farmers and the discussion forums in which they write. 50
30 Second instance of the world obtained by running the Alloy model. 50
31 Highlighting of the relation between a message an the respective sender and receiver. . . 51
32 Highlighting of the relation between a HelpRequest and the farmer who addresses it. . . 51
33 Highlighting of the relation between a suggestion and the farmer who uploads it. 52
34 Third instance of the world obtained by running the Alloy model. 52

List of Tables

1 Table of Goals 7
2 Table of World Phenomena 9
3 Table of Shared Phenomena 11
4 Table of Machine Phenomena 12
5 Table of definitions 12
6 Table of acronyms 12
7 Table of abbreviations 13
8 Table of versions 13
9 Table of domain assumptions 21

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

10 Table of requirements 27
11 Table of the mapping of requirements and domain assumptions on goals 27
12 Table of the 1st use case 31
13 Table of the 2nd use case 32
14 Table of the 3rd use case 32
15 Table of the 4th use case 32
16 Table of the 5th use case 33
17 Table of the 6th use case 33
18 Table of the 7th use case 34
19 Table of the 8th use case 34
20 Table of the 9th use case 34
21 Table of the 10th use case 35
22 Effort spent by student 1 53
23 Effort spent by student 2 53

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

1 Introduction

1.1 Purpose

The purpose of this document is to analyze the context of interest of DREAM, exhibiting the problem and specifying how the software to be has to interface with the external environment. To these means, it will be defined a list of goals of the project, followed by the requirements of the system and by the assumptions made regarding the external world. Through these elements, it will be proposed a solution to the problem. This solution, exposed in the document, will be used by different stakeholders and by the software engineers that are in charge of the project's development, in order to be able to understand the detailed specifications without any ambiguity.

DREAM is a project that aims to improve the productivity of the agricultural system in Telengana, by means of creating a software that makes simpler for the users to access to the information collected by different sources, such as agronomists, Telengana's government, sensors deployed on the territory and the water irrigation system. In this way, farmers can easily optimise their work looking at meteorological forecasts, environmental conditions and personalized suggestions; in addition, they can easily evaluate their production with respect to the others'. Whilst, policy makers, not only can precisely identify the farmers who perform well and the ones that are facing problems and, therefore, need some help, but can also understand if the steering initiatives carried out by agronomists are useful. Furthermore, the communication between the characters becomes more clear and fast.

1.1.1 Goals

Goals	Description	
G1	Improve the quality of farmers' production in Telengana.	
	G1.1	Allow a farmer to visualize information about the weather forecasts.
	G1.2	Allow a farmer to visualize suggestions about production based on his geographical area.
	G1.3	Allow a farmer to visualize suggestions about production based on the type of products which he uses.
	G1.4	Allow a farmer to visualize suggestions about production based on the environmental conditions in which he works.
	G1.5	Allow a farmer to visualize an evaluation of his work.
	G1.6	Allow a farmer in a critical situation to easily request help to other farmers.
	G1.7	Allow a farmer in a good situation to answer to a help request dispatched to him by the system.
	G1.8	Allow a farmer who performs well to give suggestions.
	G1.9	Allow a farmer to discuss some topic of interest in a forum.
G2	Automate part of the work of policy makers who are in charge of managing the agricultural system.	
	G2.1	Allow a policy maker to see rankings about the farmers' production through different criteria.
	G2.2	Allow a policy maker to let a farmer know how well his production is with respect to the specific methodology that he had previously established.
	G2.3	Allow a policy maker to communicate with a farmer in order to give him some indications.
	G2.4	Allow a policy maker to see the farmers' water consumption.

Table 1: Table of Goals

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

1.2 Scope

Agriculture plays a fundamental role in the economy of Telengana's state, not only because of the contribution it gives to the wealth of the Indian country, but also because of the large amount of people involved in it. For these reasons, the government undertakes to promote initiatives aimed at improving the quality of production. In order to reach this goal, the DREAM system has been designed to be used by farmers and policy makers with two different modalities. A farmer is a person who owns a land and cultivates it producing a certain amount of food; while a policy maker is a person who is in charge of evaluating the activities of the farmers and managing practices to improve the agricultural system.

The system must allow a farmer to visualize several information useful to plan his work in the most efficient way. These information regard meteorological forecasts and personalized suggestions concerning specific crops to plant or specific fertilizers to use. What a farmer can see is filtered by his location, the type of products that he cultivates or the environment conditions in which he works. In order to create a proper communication network they should also be able to upload specific information about their production, such as the products they cultivate with the respective amount, the fertilizers they use and the practices they put in action.

Another functionality provided by the system is the possibility to request help for a farmer who is in critical conditions. The system should manage such requests dispatching them to other farmers who are able to give them adequate suggestions. In the end, the farmer should be able to confront ideas with other farmers through the means of a discussion forum.

For what concerns policy makers, they can visualize rankings of farmers based on the quantity of their production filtered by the geographical area, meteorological and soil conditions and on the improvements during a certain period of time. Through these rankings, policy makers can identify which farmers are performing well and which aren't, and decide to send them, respectively, some incentives or the help of agronomists. In addition, policy makers can evaluate the farmers' work, relying on the rankings, and assign them a status that represent the quality of their work. Analyzing the improvements that have been made by the farmers through the rankings, policy makers are also able to understand if the mechanism of help carried out by good farmers and agronomists is working as desired. Finally, they can also visualize the farmers' water consumption and, if necessary, ask them to reduce it.

1.2.1 State of the art

Before proceeding with the design of the system, it is important to analyze the features of similar applications that reflects the current state of the art, in order to understand which are the new additional functionalities provided by DREAM. Searching on the web for applications that are used by several farmers among the Indian country, there have been found various results; some of them will be mentioned below.

- **IFFCO Kisaan App:** an application that provides customized information to farmers, such as short-term weather forecasts, products' price trends in the market, experts' advisories about cultivation and others.

- **Krishe-Farming & agriculture app for Kisan Krishi:** an application that provides personalized crop calendars, that indicates to the farmer the specific time in which he has to start every activity regarding cultivation. These calendars are generated using data uploaded by the farmer and other scientific parameters provided by research of experts.

- **Kisan Suvidha:** an application that provides weather conditions, market prices, information about available nearby services and other information about agricultural practices.

Some of the functionalities of these apps are addressed also in DREAM. For what concerns the IFFCO Kisaan App, it is more focused on providing information about the market and experts' advisories, the Krishe app is mainly aimed at facilitating the planning of farmers' work and the Kisan Suvidha app is focused on advising farmers with useful information about available products. While DREAM, not only

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

aims to provide relevant information to the farmers, but it also intends to build a system of cooperation between farmers and it allows policy makers to evaluate the trend of production in Telengana and take measures as a consequence.

In conclusion, DREAM differs from others existing application for the above specified reasons and, once implemented, it will fit well in the current state of the art.

1.2.2 World Phenomena

World Phenomena	Description
WP1	The farmer chooses the plants to cultivate.
WP2	The farmer chooses the products to use (for example fertilizers).
WP3	The farmer cultivates in favourable weather conditions.
WP4	The farmer cultivates in bad weather conditions.
WP5	The farmer quantifies his production.
WP6	A climate-related disaster happens.
WP7	The policy maker establishes a methodology to detect which farmers perform well especially considering the resilience to weather conditions.
WP8	The policy maker sends incentives to certain farmers.
WP9	Sensors measure the humidity of soil.
WP10	The government collects data concerning meteorological forecasts.
WP11	The water irrigation system collects data concerning the amount of water used by each farmer.
WP12	The agronomists obtain information by visiting the farmers.
WP13	The policy maker asks to an agronomist to help a farmer in difficulty.

Table 2: Table of World Phenomena

1.2.3 Shared Phenomena

Shared Phenomena	Description	Control
SP1	The farmer registers into the application.	World controlled
SP2	The farmer logs into the application.	World controlled
SP3	The policy maker logs into the application.	World controlled
SP4	The farmer inserts his farm's location in the profile section of the application.	World controlled
SP5	The farmer inserts the types of products he produces in the profile section of the application.	World controlled
SP6	The farmer inserts data about the amount of his production in the profile section of the application.	World controlled
SP7	The farmer inserts the products that he uses in the profile section of the application.	World controlled
SP8	The farmer visualizes the weather forecasts regarding his location.	Machine controlled
SP9	The farmer publishes information into the application regarding the best practices to use according to his geographical area.	World controlled
SP10	The farmer visualizes the suggestions, filtered by his geographical area, that other farmers have previously published into the application.	Machine controlled

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

SP11	The farmer visualizes the suggestions, filtered by the type of products which he uses, that other farmers have previously published into the application.	Machine controlled
SP12	The farmer visualizes the suggestions, filtered by the environmental conditions in which he works, that other farmers have previously published into the application.	Machine controlled
SP13	The farmer fills in a form to request help to other farmers indicating the problem that he is facing.	World controlled
SP14	The farmer receives a notification containing a request for help written by another farmer in similar conditions.	Machine controlled
SP15	The farmer answers to a request for help sending a message with adequate suggestions.	World controlled
SP16	The farmer visualizes answers to his requests for help.	Machine controlled
SP17	The farmer creates discussion forums about some topic.	World controlled
SP18	The farmer replies to a discussion forum.	World controlled
SP19	The farmer visualizes an evaluation of the quality of his work.	Machine controlled
SP20	The application provides to the policy makers a ranking of the farmers based on the quantity produced of a specific product with respect to the size of their land classified by different geographical areas.[1]	Machine controlled
SP21	The application provides to the policy makers a ranking of the farmers based on the quantity produced of a specific product with respect to the size of their land classified by different meteorological conditions.	Machine controlled
SP22	The application provides to the policy makers a ranking of the farmers based on the quantity produced of a specific product with respect to the size of their land classified by different soil conditions detected by the sensors.	Machine controlled
SP23	The application provides a ranking of the farmers based on the percentage increase of the quantity produced for a specific product with respect to the size of their land in a certain period of time.	Machine controlled
SP24	The policy maker identifies the farmers who perform well through the rankings.	World controlled
SP25	The policy maker identifies the farmers who perform bad through the rankings.	World controlled
SP26	The policy maker updates the evaluation of the farmer's work.	World controlled
SP27	The policy maker determines, through the rankings, whether the mechanism of help carried out by the agronomists and the good farmers works.	World controlled
SP28	The policy maker visualizes the environmental conditions.	Machine controlled
SP29	The policy maker visualizes the farmers' water consumption.[2]	Machine controlled

[1]The size of the land is taken into consideration inserting into the rankings the quantity produced per square meter.
[2]The farmers' water consumption is considered as the quantity of water utilized per square meter of land.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

SP30	The policy maker identifies a farmer who is consuming too much water visualizing his water consumption.	World controlled
SP31	The policy maker sends a notification to a farmer who is performing bad letting him know that he can ask for help through a form.	World controlled
SP32	The policy maker sends a notification to a farmer who is performing well asking him to upload specific information about his production.	World controlled
SP33	The policy maker sends a notification to a farmer who is performing bad warning him that he will receive a visit from an agronomist who will help him improving his work's quality.	World controlled
SP34	The policy maker sends a notification to a farmer who is consuming too much water asking him to reduce its usage.	World controlled
SP35	The farmer visualizes a message previously sent by the policy maker.	World Controlled
SP36	Data detected by the sensors are written into the database of the application.	Machine controlled
SP37	Data collected by the government are copied into the database of the application.	Machine controlled
SP38	Data collected by the water irrigation system are copied into the database of the application.	Machine controlled
SP39	Data collected by agronomists are copied into the database of the application.	Machine controlled

Table 3: Table of Shared Phenomena

Machine Phenomena	**Description**
MP1	The machine obtains data regarding the humidity of soil from the database
MP2	The machine obtains data concerning the amount of water used by each farmer from the database.
MP3	The machine obtains data concerning the meteorological forecasts from the database.
MP4	The machine obtains data collected by agronomists about the farmers' production from the database.
MP5	The machine computes a value for the amount of production of every farmer, based on the information inserted by him and the data collected by the agronomists.
MP6	The machine stores in the database data inserted in the application by the farmers.
MP7	The machine stores in the database data inserted in the application by the policy makers.
MP8	The machine dispatches the requests for help to good farmers with similar work conditions.
MP9	The machine predisposes a discussion forum when it is requested by a farmer.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

MP10	The machine elaborates rankings about the production based on the data allocated in the database.

Table 4: Table of Machine Phenomena

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Definition	Description
Status	An evaluation of the quality of a farmer's work established by a policy maker.
Suggestion	Information inserted by a farmer which has a good status and that regard the best practice to use in different situations.
Form	A form is a section of the application that contains some fields that have to be filled in with textual information concerning a help request.
Discussion forum	A discussion forum is a section of the application in which the farmers can exchange opinions about a particular topic of interest.
Conflicting data	The email address inserted during the registration process by a farmer is defined as a conflicting data if there are already others users of the DREAM application that have registered with the same address.
Valid credentials	The term credentials is referred to the email and the password inserted by a farmer during the login process; they are said to be valid if they correspond to the one inserted by such farmer during the successful registration process.
Valid recognition code	The code inserted by the policy maker during the log in process is said to be valid if correspond to one of the codes that have been previously used to register the policy makers.

Table 5: Table of definitions

1.3.2 Acronyms

Acronym	Description
RASD	Requirement Analysis and Specification Document
API	Application Programming Interface
GPS	Global Positioning System
HTTPS	Hypertext Transfer Protocol Secure
TLS	Transport Layer Security

Table 6: Table of acronyms

1.3.3 Abbreviations

Abbreviation	Description
Gn	n-th goal
WPn	n-th world phenomenon
SPn	n-th shared phenomenon
MPn	n-th machine phenomenon
Rn	n-th functional requirement

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

Dn	n-th domain assumption

Table 7: Table of abbreviations

1.4 Revision history

Version	Date	Description
1.0	23/12/2021	First version

Table 8: Table of versions

1.5 Reference documents

- Specification document: R & DD Assignment A.Y. 2021-2022
- course slides
- RASDs of the previous academic years
- https://krishe.co.in/
- http://iffcokisan.com/
- https://kisansuvidha.gov.in/
- Github repository Data4Policy (https://github.com/UNDP-India/Data4Policy.git)

1.6 Document structure

- **Section 1:** In this section it is presented an introduction of the project. Initially, it is discussed the purpose of the system and the goals that it aims to reach. Then, it is described its scope and it is analyzed the current state of the art. Moreover, the problem it has been examined by means of a list of world, shared and machine phenomena that represent the system. Finally, it is specified a list of definitions, acronyms and abbreviations that are used in the document.
- **Section 2:** Here it is presented an overall description of the project, analyzing multiple scenarios and presenting the UML class diagram that describes the different entities of the system. Furthermore, the use of some statecharts helps to understand what are the actions that can be made through the system and how they happen in a sequence. Finally, it is described more in detail which are the functionalities of the system and the characteristics of the users that interact with it.
- **Section 3:** This section contains the specification of the interface requirements, i.e. the definition of the user, hardware, software and communication interfaces needed for the system. Moreover, the functional requirements necessary for the system to work as desired and to be consistent are defined. It is also specified which of these requirements are needed for each goal to be reached. Then, the requirements are analyzed through the use of use cases and their respective diagrams that represent all the possible scenarios described in the previous section.
- **Section 4:** This section includes the alloy model and the discussion of its purpose. The results obtained by running the model are properly justified, presenting screens of the generated worlds with the respective comments.
- **Section 5:** This section indicates the effort spent by each group member while working on the project.
- **Section 6:** this section includes the references.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

2 Overall Description

2.1 Product perspective

2.1.1 Scenarios

1. Search for environmental information by an unregistered farmer

Joseph is a farmer who wants to use the system to see information about meteorological forecasts regarding his geographical zone. He is not registered yet, therefore he chooses the registration option in the login-page. The system asks him to insert his personal data and some information about his property, such as his farm's name, the location and size of his land. At the end of this process, the system sends an email to the farmer with the link to ultimate the registration. The farmer accesses to his mail and clicks on the link. Consequently, the system shows him a confirmation page and redirects him to the homepage. At this point, Joseph chooses on the menu the section regarding meteorological forecasts in his geographical zone.

2. Visualization of status and upload of suggestions by an application farmer

Mark is a farmer who is already registered in the system. He logs in to see if there have been changes in his status. He accesses the profile section in which the status is shown and he sees an evaluation labeled "good" marked by a green background. Among the messages sent by policy makers he notices a request for uploading some information about the practices that he uses, in particular regarding the types of products he utilizes to cultivate. These information will be helpful to other farmers in similar work conditions. To do so, he clicks on the section labeled "upload suggestions" and he leaves an advice about the topic, as requested by the policy maker.

3. Evaluation and rewarding of a good farmer by a policy maker

Alexander is a policy maker. To log in the system, he utilizes a recognition code, previously provided by the government, that allows him to access as a policy maker. Then, he goes to the section where the rankings of the farmers are shown, divided into categories. He visualizes the category in which the farmers are classified according to the improvements they have recently made. Consequently, considering also their positions in the other rankings in which they participate, he decides whether to update or not the status of the farmers to an higher one for the ones who have achieved the greater improvements. Where appropriate, he updates the status using the section of the system dedicated to farmers' evaluation and starts the external process of sending them the incentives for their good work.

4. Visualization of water consumption by a policy maker

Frank logs in the system using a recognition code that identifies him as a policy maker. He accesses to the section where the water consumption of the farmers is shown and he visualizes it in order to identify the farmers who are consuming too much. Then, he goes to the message section and sends a notification to such farmers asking them to decrease their water usage.

5. Discussion in a forum between farmers

Claire logs in the system as a farmer. Then she goes to the section dedicated to forums and she starts a new discussion regarding the problems she is facing due to bad meteorological conditions in her geographical zone. She asks some advice about the best products to utilize in such conditions, since the ones that she is currently using aren't working properly. Other farmers reply to the forum suggesting her to replace the products with more efficient ones considering the circumstances. They also suggest her to fill in an help request through the system, in order to receive more specific information. Therefore, she goes in the section in which the form is available and she fills it in.

6. Bad farmer invited to request help by a policy maker

Laurel logs in the system as a policy maker. She visualizes the meteorological conditions among different areas of Telengana and notices that in a certain zone the weather has been critical for a long

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

period of time. Thus, she accesses to the rankings regarding the farmers' production in that geographical area and notices that the farmer who is producing the least is in an evident critical situation with respect to the others. Finally, she sends a message to such farmer in order to letting him know that he necessarily needs help and he can ask for it through the form.

7. Visualization of suggestions by an application farmer

Matthew logs in the system as a farmer. Then he goes to the section dedicated to suggestions to see relevant information filtered, firstly, by environmental conditions and, secondly, by the products he uses to cultivate, in order to improve the quality of his work.

8. Upload of data about production by an application farmer

Jenna logs in the system to upload the quantity of rice that has been produced in her farm during the last year. To this end, she goes to the profile section and she inserts the amount of harvested rice in the field dedicated to production information. The latter contains also information regarding the products utilized for the cultivation and while reading them, she realizes that she has been using a different fertilizer with respect to the one that she had previously written in the system, hence she updates such information. Once she has finished this process, she logs out.

9. Evaluation of steering initiatives by a policy maker

Billy is a policy maker. He logs into the system to visualize the rankings of the farmers. In particular he looks at the one based on the improvements in order to understand if the mechanism of help implemented through the system is working well. He notices that in the last period farmers who are registering improvements are much more than ones who are registering a worsening. So he concludes that the work of good farmers and agronomists aimed to help farmers in difficulty is very useful.

10. Agronomist invited to visit a bad farmer by a policy maker

Ophelia logs into the system as a policy maker in order to visualize the working conditions of the farmers in a specific geographical zone in which there have been problems caused by bad meteorological conditions. She notices that the farmer at the bottom of the ranking is in a particularly bad situation. So she contacts an agronomist in order to putting him in charge of helping the farmer in difficulty and subsequently, she sends a message to such farmer to notify him that he will receive a visit by the agronomist.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

2.1.2 Class diagram

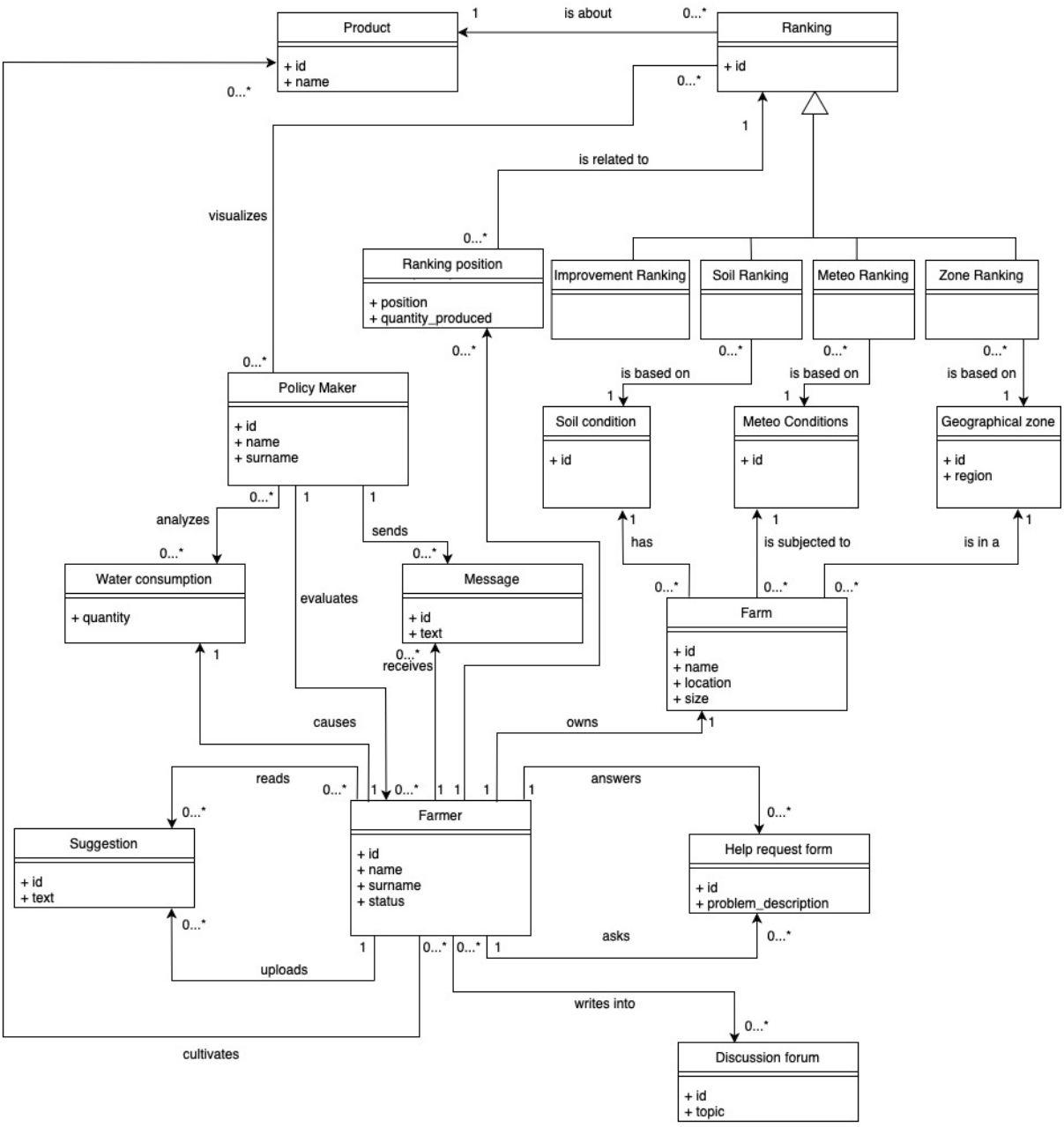

Figure 1: Class Diagram.

In order to avoid any ambiguity into the class diagram, it can be useful do briefly describe such diagram and the choices made to create it.

- It is assumed that a farmer owns only one farm and a farm is owned by only one farmer; this choice justifies the cardinalities of the association.
- A farmer is associated to some ranking positions, each of these related to a specific ranking. There are different rankings for each product.
- A farm can be associated to only one meteorological condition and one soil condition because, even if these factors can change with the course of the time, it will be considered a set of different ranges that will correspond to the percentage of time in which the conditions have been good.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

- A farmer is associated with only one policy maker who evaluates him, because we are interested in considering only the last assignment of a status for each farmer, which represents his current status.

Other cardinalities are more intuitive and it is not needed to explain them in detail in order to avoid to be unnecessarily verbose.

2.1.3 Statecharts

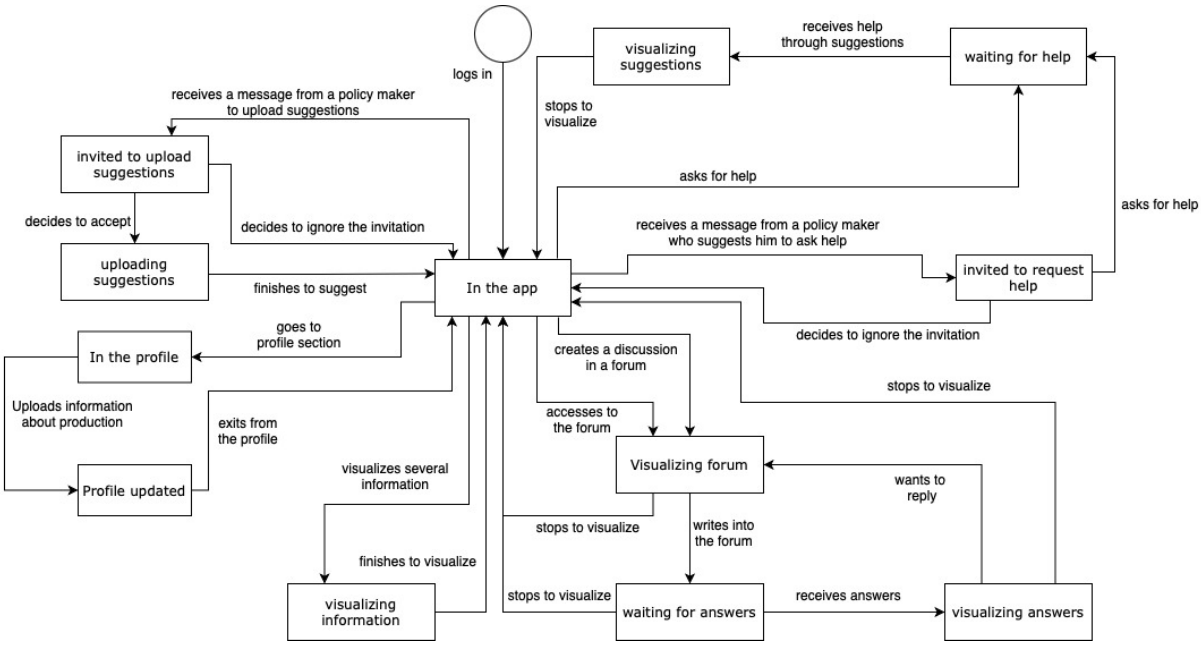

Figure 2: Statechart that descibes farmers' interaction with the system.

The above statechart describes how a farmer can interact with the system. The first mandatory action for him is to log into the application. Then, he can perform several actions:

- He can create a discussion in a forum or access to it in order to write something and then waiting for answers by other farmers; if some answers arrive he visualizes them and can reply. In every moment he can decide to stop to visualize the forum section.
- He can visualize several information such as meteorological forecasts, his status or every type of suggestions. All these are included in one unique ramification of the statechart without distinguish between the different type of information because the sequence of actions doesn't depend from it.
- He can receive a message from a policy maker who asks him to upload suggestions because he is working well; in this case he can ignore the invitation or accept it and upload some suggestions based on how he works to obtain good results.
- He can receive a message from a policy maker who suggests him to ask for help because he is in a critical situation. Then he can ignore the invitation or accept it and decide to request help through the form. Therefore he receives some suggestions from another farmer who works well in similar conditions and visualizes them.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

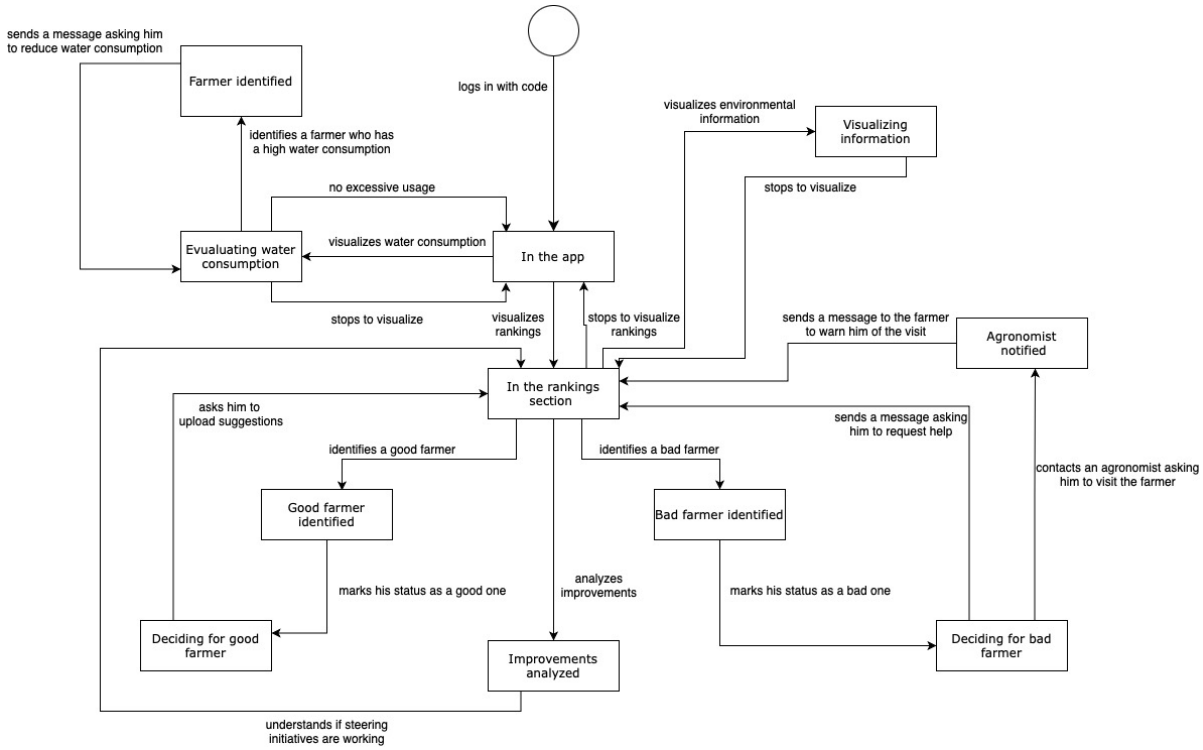

Figure 3: Statechart that describes policy makers' interaction with the system.

The above statechart describes how a policy maker can interact with the system. The first mandatory action for him is to log into the application with a recognition code that identifies him as a policy maker. Then, he can perform several actions:

- He can visualize environmental information through different geographical zones.
- He can visualize data about water consumption of the farmers and evaluate them; if there is no excessive usage by any farmer he stops to visualize, if he identifies a farmer who has a high water consumption he sends him a message asking him to reduce it and continues to evaluate data. In every moment he can stop to visualize this section.
- He can visualize rankings about production of farmers based on different factors. Through this rankings he can identify a farmer who performs particularly good, consequently mark his status as a good one and eventually ask him to upload suggestions. Alternatively, he can identify a farmer who performs particularly bad, consequently mark his status as a bad one and eventually send him a message asking him to request help or contact an agronomist asking him to visit the farmer. In the rankings section the policy maker can also analyze the improvements made by the farmers in order to understand if the steering initiatives carried out by good farmers and agronomists are working well.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

2.2 Product functions

The system provides to the farmers the following functionalities:

- **Visualize meteorological forecasts:** A farmer can visualize both short and long term meteorological forecasts relevant to the geographical area where his farm is placed. By doing so, he can plan his work in a more efficient way.

- **Exchange suggestions:** A farmer can visualize suggestions, previously posted in the system by other farmers, regarding the best practices to use in different situations, filtered by the geographical area, the utilized products or the environmental conditions. In addition, he can also post a suggestion in the system, in order for it to be visualized in a second moment by another farmer.

- **Visualize an evaluation:** A farmer can receive an evaluation, expressed through the definition of a status, which indicates if he's working well or not, with respect to other farmers in similar conditions. This status can be visualized by the farmer, who, in such way, becomes aware of his work's efficiency and behaves accordingly.

- **Request for help:** A farmer can easily request for help describing in detail the problems he is facing. Such help request will be addressed to another farmer through the system.

- **Answer to help requests:** A farmer can answer to a help request made by another farmer when he is requested to do so by the system.

- **Discuss a topic:** A farmer can either access or create a discussion forum about some topic of his interest. In such way, he can take inspiration from other farmers' opinions or write his own.

The system provides to the policy makers the following functionalities:

- **Visualize environmental information:** A policy maker can visualize the meteorological and the soil conditions of different geographical zones in Telengana.

- **Visualize rankings:** A policy maker can visualize rankings of the farmers that cultivate a specific product based on the amount produced and filtered by different factors, such as the geographical area, the weather conditions, the soil conditions or the improvements made in a certain period of time. Through these last rankings he can also understand if the steering initiatives carried out by agronomists and good farmers are working. In particular, each ranking is referred to a specific product and orders the farmers that cultivate it from the one who has produced the greater quantity per square meter to the one that has produced the least, accordingly with what is published by the farmers in the system. In the end, each farmer will participate in four different rankings for each product that he cultivates, corresponding respectively to each of the above specified factors, that represent a working condition shared by all the farmers in the ranking.

- **Visualize water consumption:** A policy maker can visualize the quantity of water consumed by each farmer to cultivate the land, in order to manage any excesses in its use.

- **Evaluate farmers' work:** A policy maker can evaluate the work of a farmer assigning to him a status. He can base its assessment on the different types of rankings that the system provides, analyzing them by means of a methodology previously decided.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

2.3 User characteristics

The system involves the following actors:

- **Unregistered farmer**
  A person who owns a farm and is not registered into the DREAM application yet, hence to use its functionalities has to create an account.
- **Application farmer**
  A person who owns a farm, is already registered into the application, and therefore is able to use all its functionalities.
- **Policy maker**
  A person who is in charge of evaluating and monitoring the activities of the farmers and, to this end, interacts with the DREAM application logging into it with a recognition code to identify himself.

2.4 Assumptions, dependencies and constraints

Domain assumption	Description
D1	The farmers have access to a device on which they can run the DREAM system.
D2	The recognition code that policy makers use to login is known only by them, so that the identity of who accesses as a policy maker can be assumed to be correct.
D3	Each farmer owns only one farm.
D4	The personal information inserted during the registration process by a farmer are assumed to be truthful.
D5	The information about production inserted by the farmers in the system are assumed to be truthful.
D6	The data provided by external services regarding the meteorological forecasts are assumed to be reliable.
D7	The data provided by external services regarding the meteorological forecasts are assumed to be updated, once the time is passed, in order to record meteorological events that happened in the past and use them for the rankings' establishment.
D8	Policy makers have all the knowledge necessary to evaluate farmers' work through the rankings, following a precise methodology.
D9	Policy makers have all the knowledge necessary to establish, using the rankings, whether the mechanism of help carried out by agronomists and good farmers works.
D10	Policy makers have all the knowledge necessary to establish, visualizing the data provided by the system about water consumption, whether a farmer is consuming too much water with respect to the availability of the water irrigation system.
D11	Data collected by the sensors deployed on the territory and by the water irrigation system are reliable thanks to the periodic maintenance done by experts, who are external to the system.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

D12	Data collected by agronomists are assumed to be truthful, because only reliable experts are chosen for the role.

Table 9: Table of domain assumptions

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interfaces showed in this document are useful to have a first idea of the structure of the application, both from the side of a farmer and from the side of a policy maker. The mockups are realized for a iOS mobile application, even if the realization is planned for both mobile and web devices. Furthermore, only sign in and home interfaces will be showed; other more specific ones and other for the web devices will be presented in the DD document.

- **Sign in interfaces**

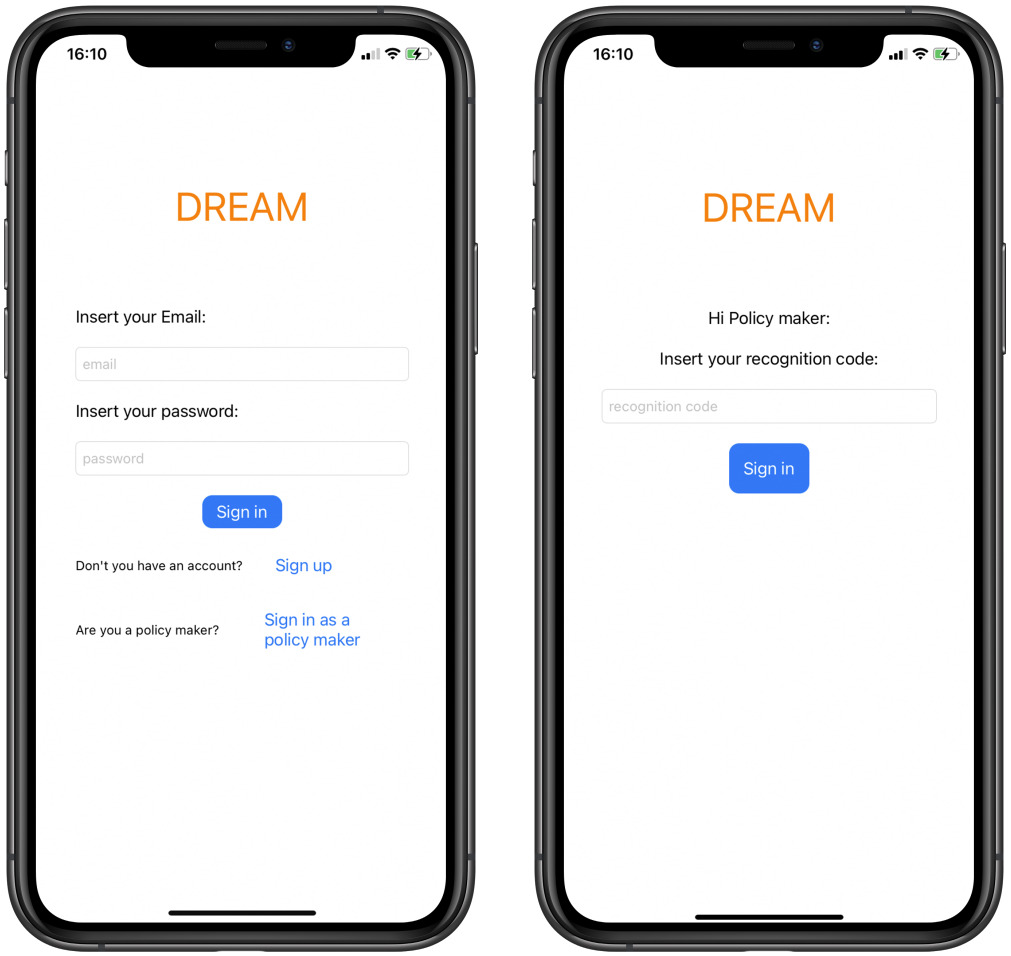

Figure 4: Sign in interfaces for farmers (on the left) and policy makers (on the right) of DREAM.

These interfaces show the very first screen of the application in which a farmer can sign into the application inserting his credentials. For what concerns the policy maker he can choose to sign in as a policy maker inserting then his recognition code in the next view.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

- **Farmer interfaces**

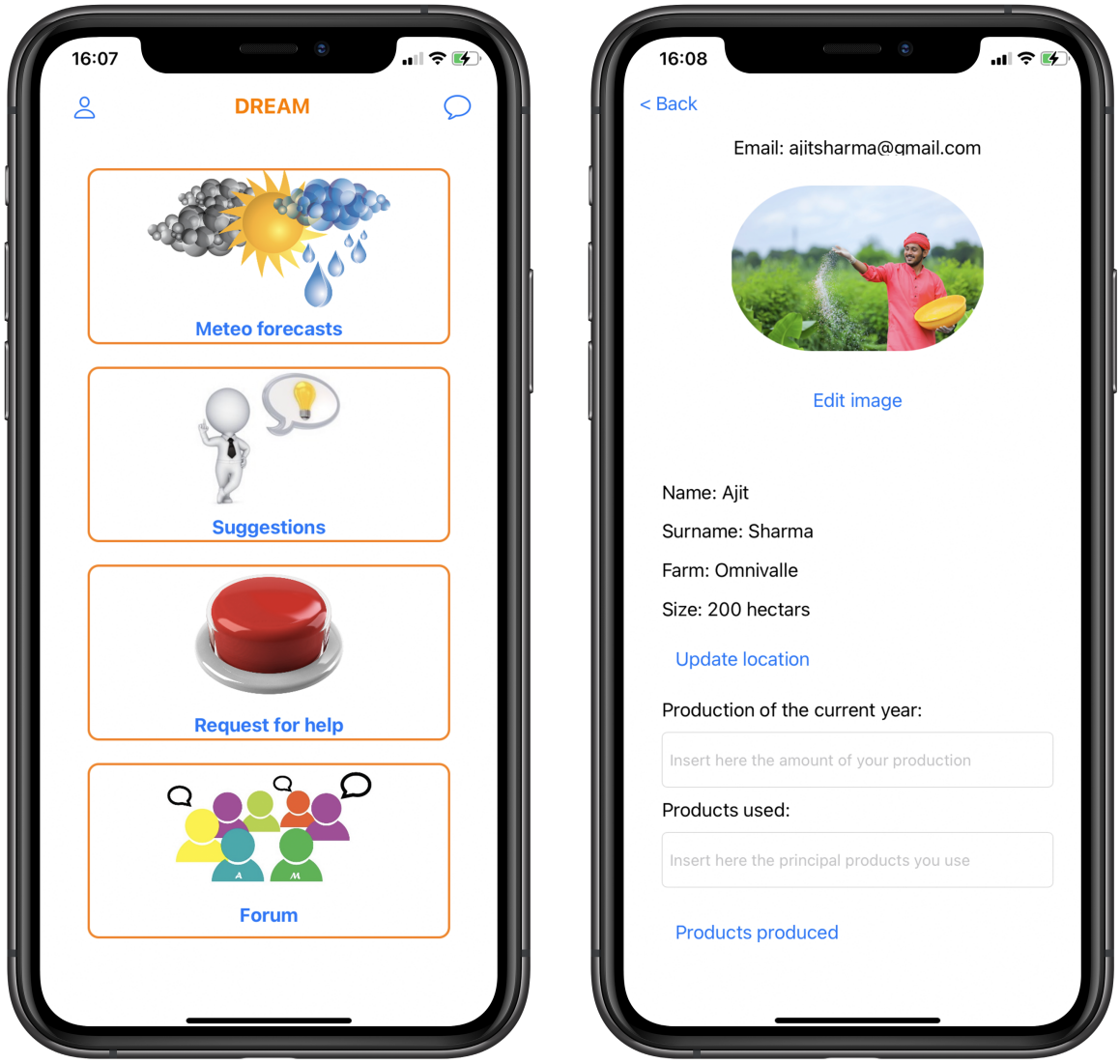

Figure 5: Home interface (on the left) and Profile interface (on the right) for a farmer of Dream.

The first screen shows the home page of the application from the point of view of a farmer. Here he can access to several sections: one for visualizing meteorological forecasts, one for giving and reading suggestions, one for requesting help and one for the forum. Finally, through the buttons at the top of the application he can access to his profile and to the section dedicated to the messages in order to visualize notices sent by policy makers.
The second interface shows the profile section. Here he can visualize his personal information and update some information about his production, such as the production of the current year and the products he uses to cultivate. Finally, he can click on some buttons which allow him, in other sections, to update the location of his farm and the products he cultivates.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

- **Policy maker interfaces**

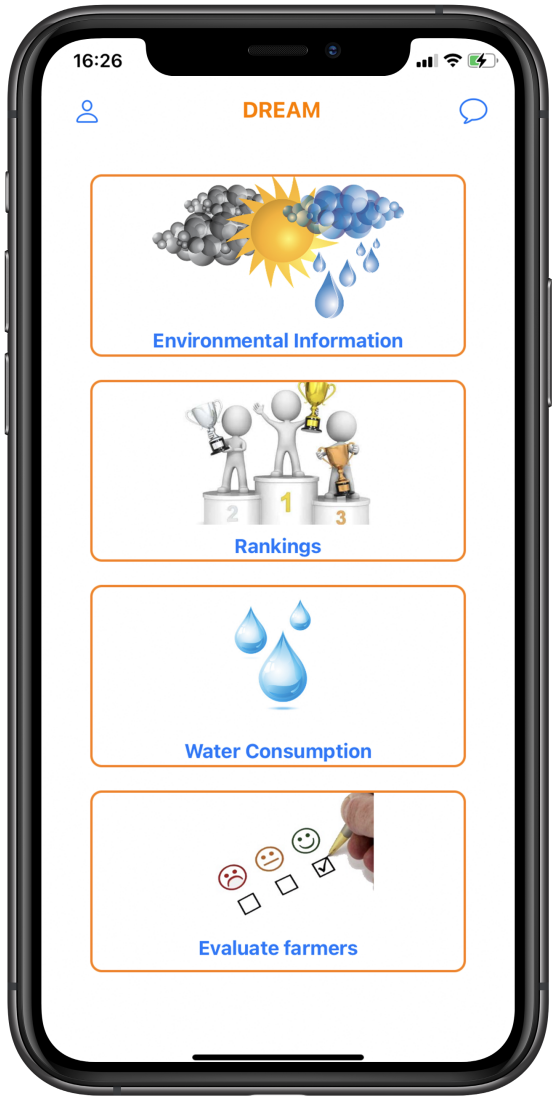

Figure 6: Home interface for a policy maker of DREAM.

This last screen represents the home page of the application from the point of view of a policy maker. Here he can access to several sections: one for environmental information, one for visualizing rankings, one for visualizing water consumption and one for evaluating the farmers. Finally, through the button at the top of the application he can access to his profile and to the section dedicated to the messages in order to send notices to the farmers.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

3.1.2 Hardware Interfaces

The DREAM system, to be correctly used by its users, will require, a smartphone with enough storage to run the mobile application on it or a computer which supports the usage of a web browser. In both cases, the device should be connected to an internet connection in order to properly deliver the functionalities of the application.

3.1.3 Software Interfaces

The DREAM system interacts with multiple external services in order to retrieve fundamental data for the scope of the application. To this end, it is required the usage of different APIs. First of all, it is needed an API to let the application communicate with its database. Then, it is also important the usage of a Geolocation API for detecting the precise position of the farmer's device during the registration process; in this way the insertion of his farm's location is facilitated. Furthermore, the registration and login processes can be managed by an external API, such as the Google Sign-in API, in order to avoid the implementation of encrypting of data.

3.1.4 Communication Interfaces

Both the mobile and web application that compose the system are going to communicate with the server using the HTTPS protocol. In fact, in order to obtain security and reliability, with HTTPS it is used the Transport Layer Security (TLS) for the encryption.

3.2 Functional Requirements

3.2.1 List of requirements

Requirement	Description
R1	The system shall allow an unregistered farmer to create an account in the system.
R2	The system shall allow an unregistered farmer to insert his personal data in the registration page.
R3	The system shall check if the personal data inserted by an unregistered farmer in the registration page are conflicting with respect to other existing accounts.[3]
R4	The system shall send an email containing a link to the unregistered farmer once he has inserted his data.
R5	The system shall show a confirmation page to the farmer once he clicks on the link contained in the email.
R6	The system shall redirect the farmer, who is now registered, to the homepage.
R7	The system shall allow an application farmer to log in the system.
R8	The system shall check if the credentials inserted by an application farmer in the log-in page are valid.[4]
R9	The system shall show to an application farmer only the weather forecasts relevant to his geographical zone.
R10	The system shall allow the application farmer to select the parameter by which the visualization of the suggestions will be filtered.
R11	The system shall show to the application farmer the suggestions about production regarding his geographical area.
R12	The system shall show to the application farmer the suggestions about production filtered by the products used for the cultivation.
R13	The system shall show to the application farmer the suggestions about production filtered by the environmental conditions in which he works.
R14	The system shall allow the application farmer to access to the profile section.

[3] The concept of conflicting data is defined in the section 1.3.1.
[4] The concept of valid credentials is defined in the section 1.3.1.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

R15	The system shall show to the application farmer his status in the profile section.
R16	The system shall allow the application farmer to upload in the profile section the quantity produced during the last year of a certain type of product.
R17	The system shall allow the application farmer to upload in the profile section the type of products that he uses to cultivate.
R18	The system shall allow to upload a suggestion only to an application farmer who has the status of good farmer.
R19	The system shall allow the application farmer to create a discussion forum.
R20	The system shall allow the application farmer to access a discussion forum.
R21	The system shall allow the application farmer to search for a topic in the discussion forum.
R22	The system shall allow the application farmer to write into a discussion forum.
R23	The system shall allow the application farmer to fill in a form to request for help.
R24	The system shall dispatch the help requests only to application farmers who have a good status.
R25	The system shall dispatch the help requests only to application farmers who work in similar work conditions.
R26	The system shall allow an application farmer to answer to a help request, which has been dispatched to him.
R27	The system shall generate the rankings of the farmers based on the quantity produced of a specific product with respect to the size of their land classified by different geographical areas.[5]
R28	The system shall generate the rankings of the farmers based on the quantity produced of a specific product with respect to the size of their land classified by different meteorological conditions.
R29	The system shall generate the rankings of the farmers based on the quantity produced of a specific product with respect to the size of their land classified by different soil conditions detected by the sensors.
R30	The system shall generate the rankings of the farmers based on the percentage increase of the quantity produced for a specific product with respect to the size of their land in a certain period of time.
R31	The system shall allow a policy maker to log in the system through a recognition code.
R32	The system shall check if the recognition code inserted by a policy maker is valid. [6]
R33	The system shall allow the policy maker to visualize the rankings of the farmers based on the quantity produced of a specific product with respect to the size of their land classified by different geographical areas.
R34	The system shall allow the policy maker to visualize the rankings of the farmers based on the quantity produced of a specific product with respect to the size of their land classified by different meteorological conditions.
R35	The system shall allow the policy maker to visualize the rankings of the farmers based on the quantity produced of a specific product with respect to the size of their land classified by different soil conditions detected by the sensors.
R36	The system shall allow the policy maker to visualize the rankings of the farmers based on the percentage increase of the quantity produced for a specific product with respect to the size of their land in a certain period of time.

[5] The size of the land is taken into consideration inserting into the rankings the quantity produced per square meter.
[6] The concept of valid recognition code is defined in the section 1.3.1.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

R37	The system shall allow a policy maker to assign a status to an application farmer according to a methodology previously established based on the result shown by the rankings.
R38	The system shall allow a policy maker to visualize the farmers' water consumption.
R39	The system shall allow a policy maker to send a notification to an application farmer who has a bad status to invite him to request for help through the system.
R40	The system shall allow a policy maker to send a notification to an application farmer who as a good status in order to ask him to upload a suggestion in the system.
R41	The system shall allow a policy maker to send a notification to an application farmer who has a bad status in order to tell him that he will receive the visit of an agronomist.
R42	The system shall allow a policy maker to send a notification to an application farmer asking him to reduce his water consumption.
R43	The system shall periodically obtain data from the database regarding measures collected by the sensors.
R44	The system shall periodically obtain data from the database regarding measures collected by the water irrigation system.
R45	The system shall periodically obtain data from the database regarding meteorological forecasts collected by the government.
R46	The system shall periodically obtain data from the database regarding information collected by the agronomists.

Table 10: Table of requirements

3.2.2 Mapping on goals

Goals	Domain assumptions	Requirements
G1.1	D1, D3, D4, D6	R1, R2, R3, R4, R5, R6, R7, R8, R9, R45
G1.2	D1, D3, D4	R1, R2, R3, R4, R5, R6, R7, R8, R10, R11
G1.3	D1, D3, D4, D5	R1, R2, R3, R4, R5, R6, R7, R8, R10, R12, R17
G1.4	D1, D3, D4, D5, D6, D7, D11, D12	R1, R2, R3, R4, R5, R6, R7, R8, R10, R13
G1.5	D1, D3	R1, R2, R3, R4, R5, R6, R7, R8, R14, R15, R16
G1.6	D1, D3	R1, R2, R3, R4, R5, R6, R7, R8, R23
G1.7	D1, D4, D5, D12	R1, R2, R3, R4, R5, R6, R7, R8, R16, R24, R25, R26
G1.8	D1, D3, D5	R1, R2, R3, R4, R5, R6, R7, R8, R14, R16, R17, R18
G1.9	D1	R1, R2, R3, R4, R5, R6, R7, R8, R19, R20, R21, R22
G2.1	D2, D3, D4, D5, D6, D7, D9, D11, D12	R16, R27, R28, R29, R30, R31, R32, R33, R34, R35, R36, R43, R44, R45, R46
G2.2	D2, D5, D8, D12	R16, R27, R28, R29, R30, R31, R32, R33, R34, R35, R36, R37
G2.3	D2, D4, D5, D6, D7, D8, D10	R31, R32, R39, R40, R41, R42
G2.4	D2, D3, D11	R31, R32, R38, R44

Table 11: Table of the mapping of requirements and domain assumptions on goals

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

3.2.3 Use case diagrams

Use case diagrams are used in the document in order to give a graphical description of users' interaction with the system. In particular, it is provided a use case diagram for each possible user, one for the unregistered farmer, one for the application farmer and one for the policy maker.

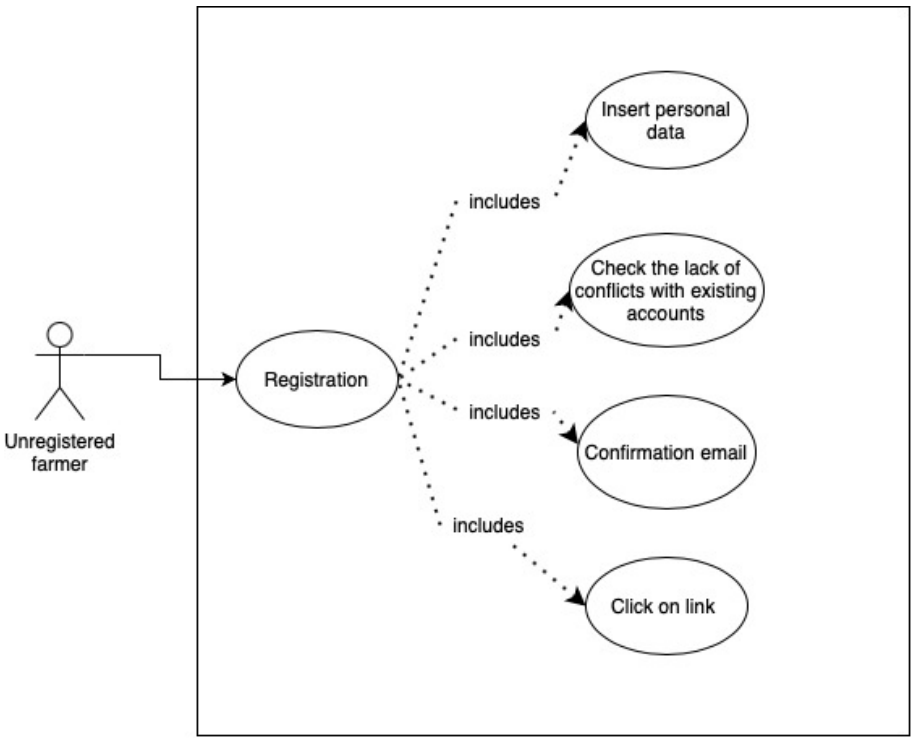

Figure 7: Use case diagram referred to the unregistered farmer.

An unregistered farmer can only register himself into the system. The registration includes several phases, i.e. insertion of personal data by the farmer, checking the lack of conflicts with existing accounts by the system, sending of a confirmation email by the system and the clicking on the confirmation link by the farmer. All these phases are mandatory.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

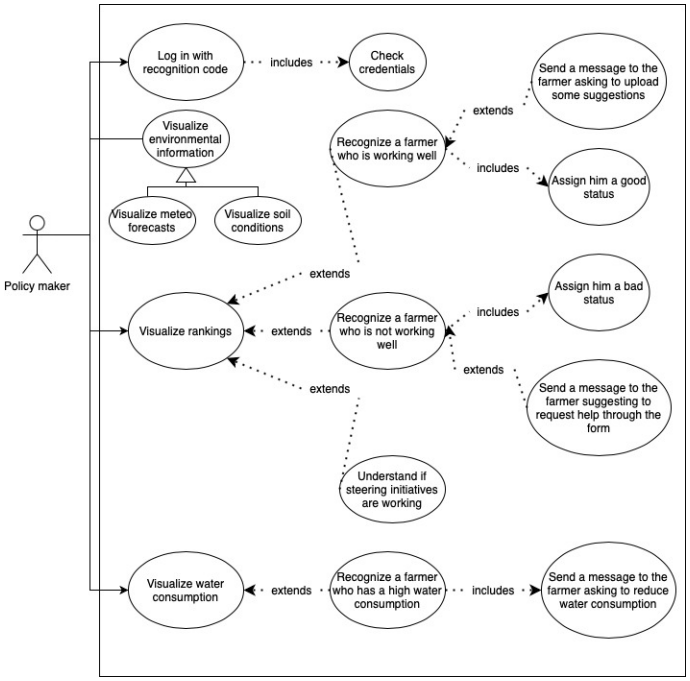

Figure 8: Use case diagram referred to the policy maker.

A policy maker can perform four principal actions: logging in, visualization of rankings and visualization of water consumption and visualization of environmental information.

- For what concerns the process of logging in, it includes the check of the credentials' correctness by the system, that is mandatory.
- For what concerns the visualization of rankings, it can includes also the recognition of a farmer who is working well or not and the evaluation of the productivity of the steering initiatives carried out by good farmers and agronomists (all these operations are not mandatory, a policy maker has also the possibility of only visualizing rankings without doing any other thing). The evaluation of a farmer who is working well or not necessarily includes the assignment to him of a respectively good or bad status. During this process, the policy maker has the possibility of sending a message to the good or bad farmer inviting him respectively to upload some suggestions or to request for help through the form.
- For what concerns the visualization of water consumption it can include the recognition of a farmer who has an high water consumption (it is not mandatory, a policy maker has also the possibility of only visualizing water consumption without doing any other thing). However, the recognition of a farmer who has an high water consumption includes sending a message to him asking to reduce it.
- For what concerns the visualization of environmental information, the policy maker can visualize both the meteorological forecasts and the soil conditions.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

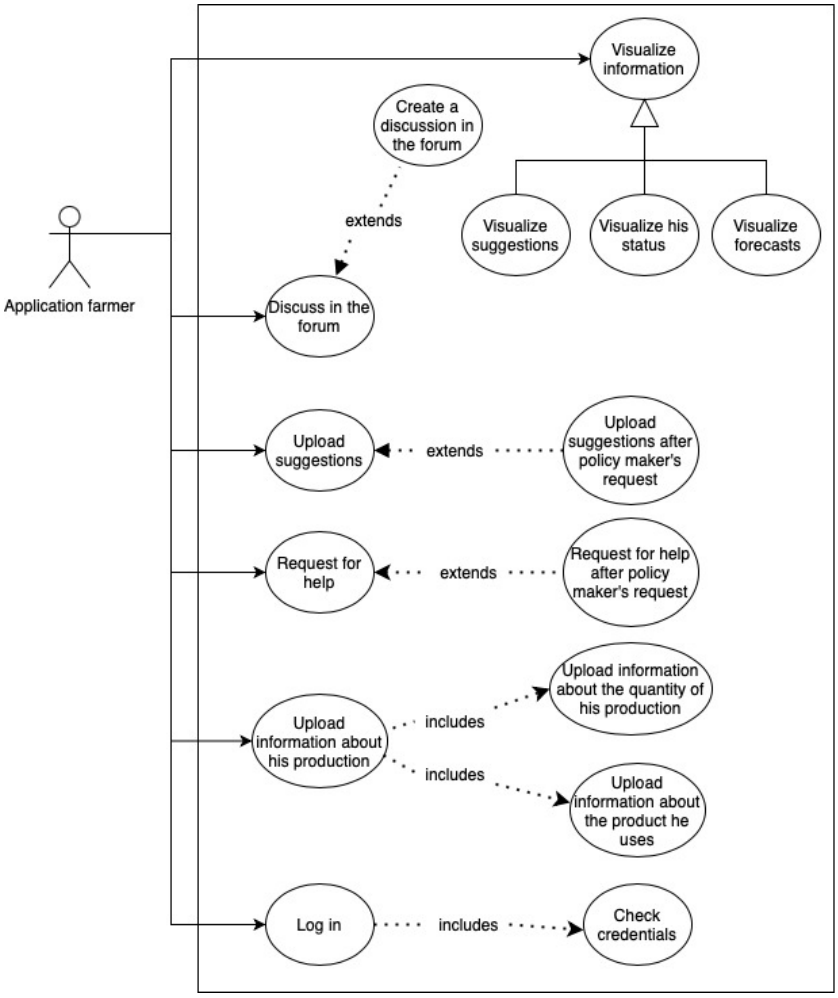

Figure 9: Use case diagram referred to the application farmer.

An application farmer can perform several actions:

- Logging into the application: it includes the check of the credentials' correctness by the system, that is mandatory.
- Uploading of information about his production: it includes the uploading of information about the quantity of his production and about the products he uses. These operations are mandatory.
- Requesting for help: it may happen after a request of a policy maker who notices the critical situation of the farmer.
- Uploading of suggestions: it may happen after a request of a policy maker who notices the good work of the farmer.
- Discussing in a forum: in the forum he can also create a new discussion about a topic of interest.
- Visualizing several information such as suggestions by other farmers, his status or meteorological forecasts.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

3.2.4 Use cases

Name	**Search for environmental information by an unregistered farmer**
Actors	Unregistered farmer, Application farmer
Entry conditions	The unregistered farmer has installed the application on his device. The unregistered farmer is on the registration screen of the application.
Event flow	- The unregistered farmer chooses the option to sign up. - The system asks him to insert personal data and information about the farm. - The unregistered farmer inserts all these information. - The system sends a confirmation link to the unregistered farmer's email address to confirm it. - The system warns the unregistered farmer to check his mail in order to continue the registration. - The unregistered farmer accesses to his mail and clicks the link. - The system saves the information. - The system redirects the farmer (who is now an application farmer) to the home page. - The application farmer chooses on the menu the section regarding meteorological forecasts. - The system shows to the application farmer the page regarding meteorological forecasts in his geographical zone.
Exit conditions	The farmer has successfully registered and now he's able to use the application. The application farmer visualizes meteorological forecasts he was interested into.
Exceptions	- The farmer is already signed up. - The farmer's email is already used. These exceptions are handled by warning the farmer with an error message and redirecting him to the log in page. - The unregistered farmer does not fill one or more fields. This exception is handled by warning the farmer with an error message which asks to fill them. - The confirmation email does not arrive to the farmer. This exception is handled by giving the possibility to the unregistered farmer to click on a button in order to receive the email again.

Table 12: Table of the 1st use case

Name	**Visualization of status and upload of suggestions by an application farmer**
Actors	Application farmer, Policy maker
Entry conditions	The application farmer is already registered. The application farmer has previously uploaded information about his production to take part in the rankings.
Event flow	- The application farmer inserts his email and password to log in. - The application farmer clicks on the "Log in" button. - The system redirects him to the home page. - The application farmer accesses to the profile section. - The application farmer visualizes his status. - The application farmer receives a message from a policy maker who asks him to upload suggestions. - The application farmer accesses to the suggestions' section.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

	- The application farmer uploads a suggestion as requested by the policy maker.
Exit conditions	The application farmer has successfully visualized his status. The application farmer has successfully uploaded the suggestions.
Exceptions	- The application farmer inserts invalid email. - The application farmer inserts invalid password. These exceptions are handled by warning the farmer with an error message and asking him to insert them again.

Table 13: Table of the 2nd use case

Name	**Evaluation and rewarding of a farmer by a policy maker**
Actors	Policy maker
Entry conditions	No entry conditions
Event flow	- The policy maker logs into the application using his recognition code. - The system redirects him to the home page. - The policy maker accesses to the rankings' section. - The policy maker selects the improvement rankings. - The policy maker identifies a farmer with good improvements. - The policy maker visualizes other rankings in order to understand if it is reasonable to change the status of the farmer. - If it is reasonable the policy maker accesses to the evaluation section. - The policy maker updates farmer's status. - The policy maker starts the external process of sending him incentives for his good work.
Exit conditions	The policy maker has successfully visualized rankings. The policy maker has successfully uploaded the status of a farmer.
Exceptions	- The policy maker inserts an invalid recognition code. This exception is handled by warning the policy maker with an error message saying that the recognition code is invalid and asking him to insert it again.

Table 14: Table of the 3rd use case

Name	**Visualization of water consumption by a policy maker**
Actors	Policy maker
Entry conditions	The policy maker is already logged in.
Event flow	- The policy maker accesses to the section of the application dedicated to water consumption. - The policy maker visualizes farmers' water consumption and identifies the ones who are consuming too much. - The policy maker accesses to the message section. - The policy maker sends a message to farmers who are consuming too much asking them to decrease their usage.
Exit conditions	The policy maker has successfully visualized water consumption. The policy maker has successfully sent messages to farmers who are consuming too much water.
Exceptions	No exceptions

Table 15: Table of the 4th use case

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

Name	**Discussion in a forum between farmers**
Actors	Application farmer
Entry conditions	The application farmer is already logged in.
Event flow	- The application farmer accesses to forum section. - The application farmer starts a discussion about problems caused by bad meteorological conditions. - The application farmer asks in the forum about the best products to use in such conditions. - Other application farmers answer to her giving suggestions. - The application farmer, if she considers it necessary, accesses also to the "Request help" section. - The application farmer fills in a form to request a more specific help.
Exit conditions	The application farmer has successfully discussed the topic in the forum. The application farmer has successfully requested help.
Exceptions	No exceptions

Table 16: Table of the 5th use case

Name	**Bad farmer invited to request help by a policy maker**
Actors	Policy maker
Entry conditions	The policy maker is already logged in.
Event flow	- The policy maker accesses to the section of the application dedicated to environmental conditions. - The policy maker visualizes meteorological forecasts. - The policy maker identifies a geographical zone in which there has been very bad meteorological conditions. - The policy maker accesses to the section dedicated to the rankings. - The policy maker visualizes rankings in that specific geographical zone. - The policy maker identifies a farmer who is in critical conditions. - The policy maker accesses to the section dedicated to messages. - The policy maker sends a message to such farmer suggesting him to request help through the form.
Exit conditions .	The policy maker has successfully visualized meteorological forecasts. The policy maker has successfully visualized rankings. The policy maker has successfully sent a message to a farmer in a critical situation.
Exceptions	No exceptions

Table 17: Table of the 6th use case

Name	**Visualization of suggestions by an application farmer**
Actors	Application farmer
Entry conditions	The application farmer is already logged in. The application farmer has previously uploaded data about the products he uses to cultivate. The application farmer has previously uploaded data about the location of his farm.
Event flow	- The application farmer accesses to the section dedicated to suggestions. - The application farmer chooses to filter suggestions by environmental conditions (determined by his location) and visualizes them.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

	- The application farmer filters suggestions by the products he uses and visualizes them.
Exit conditions	The application farmer has successfully visualized suggestions.
Exceptions	- The application farmer has not inserted in his profile the products he uses. - The application farmer has not inserted in his profile the location of his farm. These exceptions are handled by warning the farmer with an error message and redirecting him to his profile in order to upload such information.

Table 18: Table of the 7th use case

Name	**Upload of data about production by an application farmer**
Actors	Application farmer
Entry conditions	The application farmer is already logged in.
Event flow	- The application farmer accesses to the profile section. - The application farmer upload the quantity of rice produced. - The system saves such data. - The application farmer upload data about the products he utilizes to cultivate. - The system saves such data.
Exit conditions	The application farmer has successfully uploaded data about his production and the products he uses.
Exceptions	No exceptions

Table 19: Table of the 8th use case

Name	**Evaluation of steering initiatives by a policy maker**
Actors	Policy maker
Entry conditions	The policy maker is already logged in.
Event flow	- The policy maker accesses to the section dedicated to the rankings. - The policy maker chooses to see the rankings based on improvements. - The policy maker notices that several improvements have been made by many farmers and understands that the steering initiatives are working well.
Exit conditions	The application farmer has successfully visualized improvements rankings.
Exceptions	No exceptions

Table 20: Table of the 9th use case

Name	**Agronomist invited to visit a bad farmer by a policy maker**
Actors	Policy maker
Entry conditions	The policy maker is already logged in.
Event flow	- The policy maker accesses to the section dedicated to the rankings. - The policy maker visualizes rankings in a specific geographical zone in which there have been bad meteorological conditions. - The policy maker identifies a farmer who is in critical conditions. - The policy maker contacts an agronomist in order to putting him in charge of helping the farmer. - The policy maker accesses to the section dedicated to messages. - The policy maker sends a message to such farmer warning him that he will receive a visit by an agronomist.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

Exit conditions	The policy maker has successfully visualized rankings. The policy maker has successfully sent a message to the farmer in difficulty.
Exceptions	No exceptions.

Table 21: Table of the 10th use case

3.2.5 Sequence diagrams

The following figures represent the sequence diagrams, each of them is referred to one of the use cases studied in the previous section.

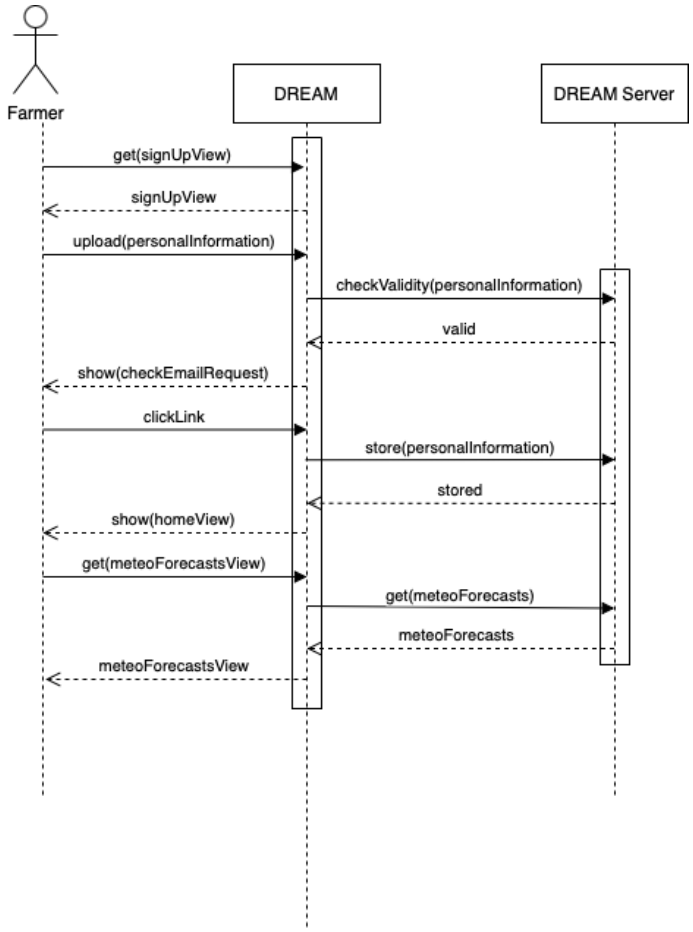

Figure 10: Sequence diagram referred to the 1st use case.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

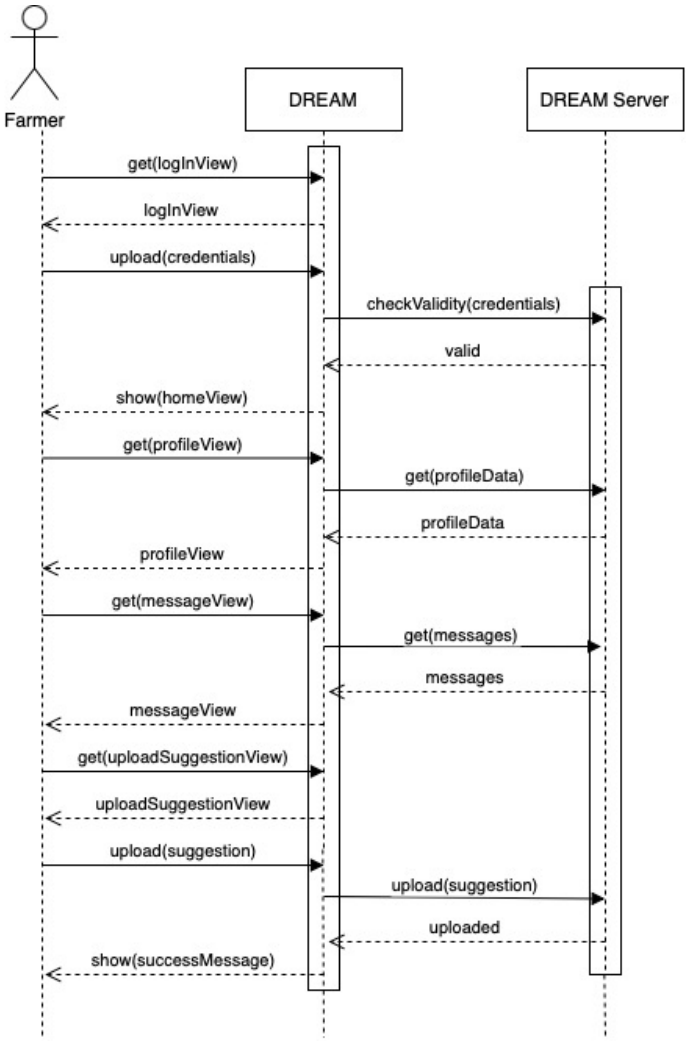

Figure 11: Sequence diagram referred to the 2nd use case.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

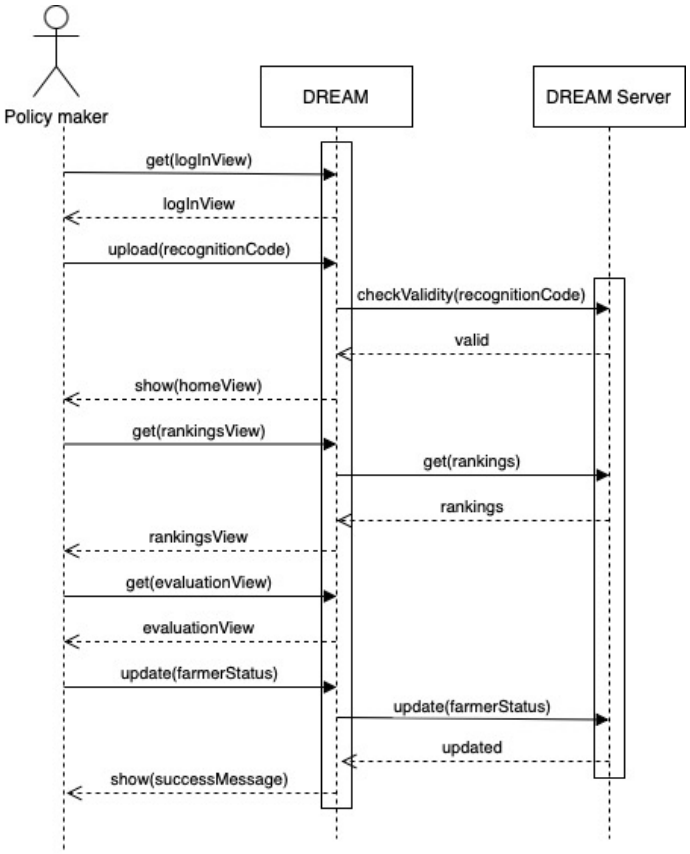

Figure 12: Sequence diagram referred to the 3rd use case.

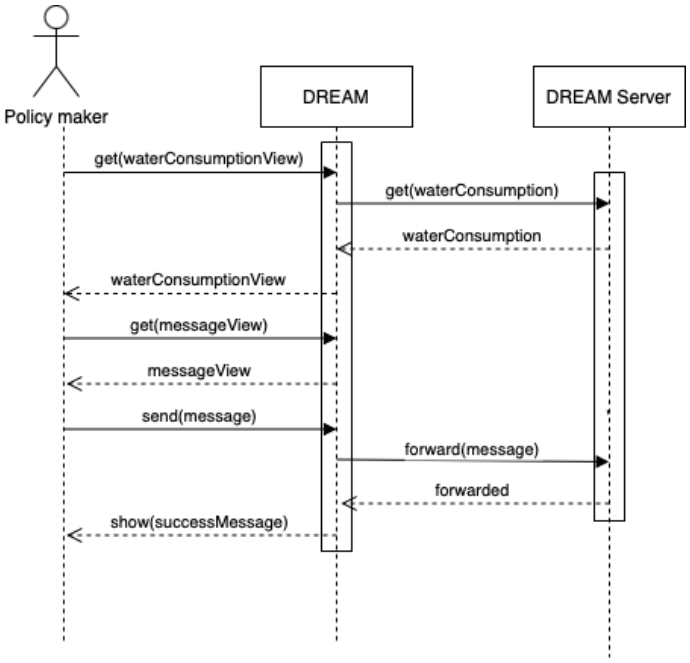

Figure 13: Sequence diagram referred to the 4th use case.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

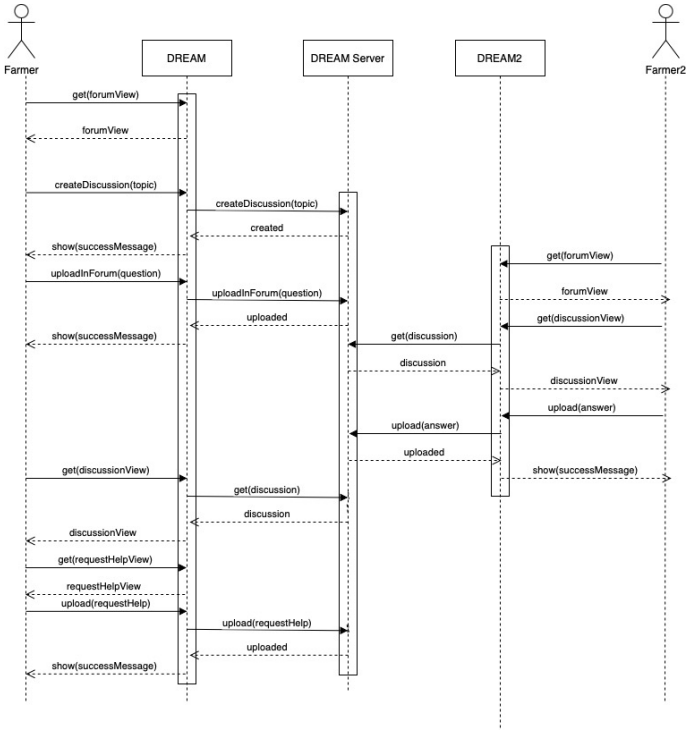

Figure 14: Sequence diagram referred to the 5th use case.

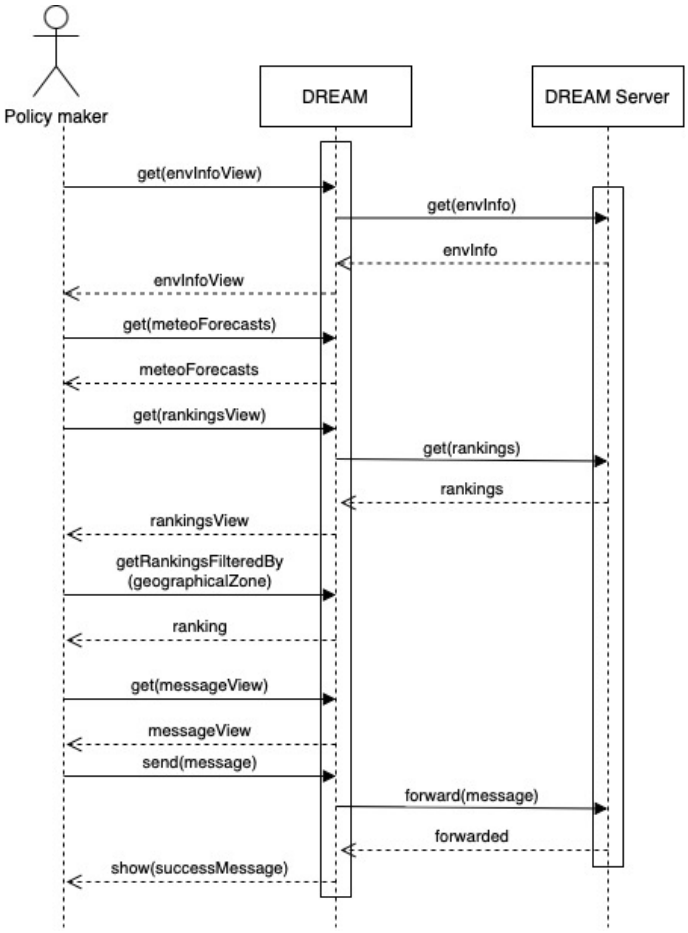

Figure 15: Sequence diagram referred to the 6th use case.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

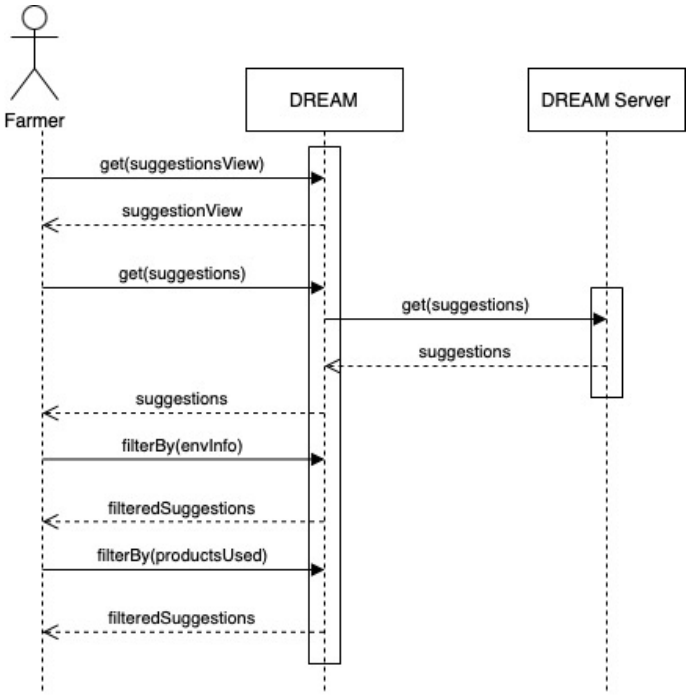

Figure 16: Sequence diagram referred to the 7th use case.

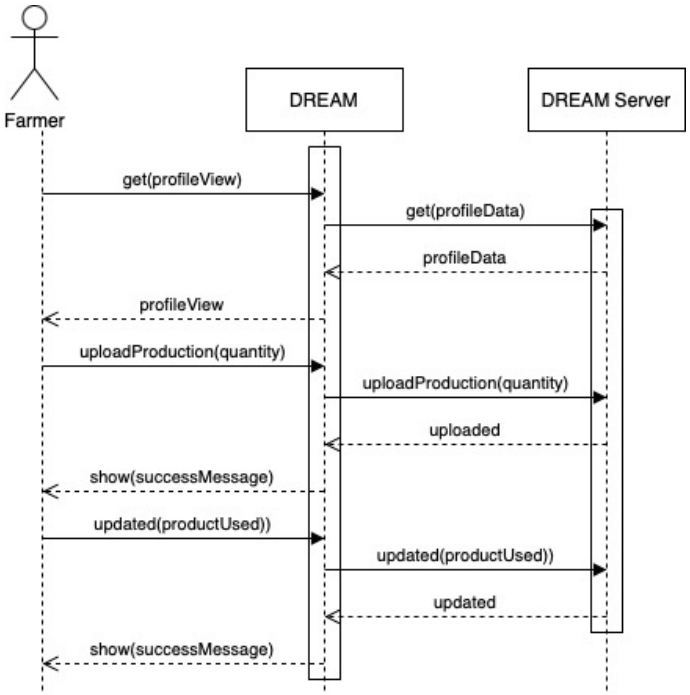

Figure 17: Sequence diagram referred to the 8th use case.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

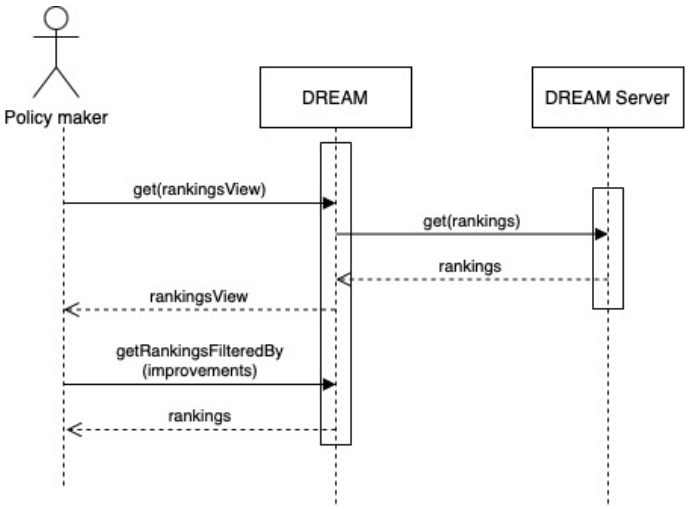

Figure 18: Sequence diagram referred to the 9th use case.

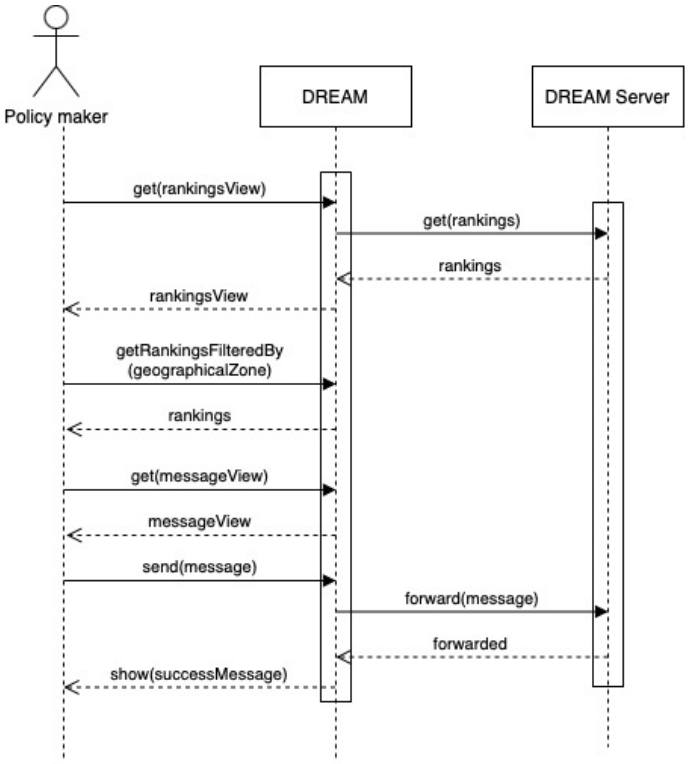

Figure 19: Sequence diagram referred to the 10th use case.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

3.3 Performance Requirements

The system should be able to provide its functionalities to several users simultaneously. In fact, the agricultural sector in Telengana employs the 55.6% of the population and the DREAM projects aims to engage the greatest possible number of farmers. In addition, the system should be available for the 99.99% of the time. Finally, the system should be able to respond to the users' request in less than 5 seconds.

3.4 Design Constraints

3.4.1 Standards compliance

The system should be subjected to the standards of a public digital good; thus implies, for example, the software to be open source and to adhere to privacy and other applicable laws and best practices. In addition, the system should ask the farmers' permission for the use of their personal information for rankings' analysis and data sharing. Finally, the system must ask the farmers' permission to retrieve information about their farm's location using the GPS.

3.4.2 Hardware limitations

- **Farmer's device:**

  The system, to be used with all the functionalities needed by a farmer, requires a mobile device with an active internet connection and a GPS service or a computer which supports the usage of a web browser and it is connected to a wi-fi network.

- **Policy maker's device:**

  Equivalently to the above case, the system to be used with the functionalities needed by a policy maker, requires the use of a computer which supports the usage of a web browser or a mobile device. In both cases, the device must be connected to an internet network in order to work properly.

- **Server:**

  The system requires the presence of a server, which has to manage the multiple tasks of the application, responding to several requests simultaneously.

3.5 Software System Attributes

3.5.1 Reliability

The software should be reliable, meaning that it should face possible failures in order to provide its functionalities without interruptions. To do so, a duplication of the data present in the database is needed to deal with any loss of information.

3.5.2 Availability

The software should be available for the 99.99% of the time, in order to guarantee the continuity of its services. To do so, the system should implement multiple servers, in order to deal with possible malfunctioning of one of them.

3.5.3 Security

The personal information inserted by the users in the application should be protected by encryption in order to respect their privacy and avoid the leak of sensitive information.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

3.5.4 Maintainability

The system should be developed in a way that easily permits its update and maintenance. For example the software should be implemented by the usage of subcomponents that leads the software developers to be able to easily modify them.

3.5.5 Portability

The software should be designed to be accessible by different devices with different operating systems. This aim should be reached by both the mobile version used by the farmers and the desktop version used by the policy makers.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

4 Formal Analysis Using Alloy

In this section it is analyzed the Alloy model. The main aim of this analysis is to create a model of the system, defining all its entities with the respective properties and imposing the constraints that make the model consistent. In such way, it will be possible to create some instances of the model using the Alloy tool Analyzer and to visualize them through a graphical notation, which help to better understand if the imposed constraints are valid and consistent with the real world domain associated with the system.

For simplicity, it has been analyzed a model that contains only one type of rankings. In this way, without any loss of generality, it is possible to see if the constraint related to the rankings are respected, but at the same time the model is more easy to understand. The rankings that have been considered in the model are the ones related to the improvements reached by the farmers in a certain period of time.

4.1 Code

```
open util/integer

// as previously specified, to one farmer is associated only one farm
sig Farmer{
property: one Farm,
waterConsumption: one WaterConsumption,
status: one Status,
evaluatedBy: one PolicyMaker
}

// for simplicity reasons, the size is expressed as an Int representing the square meters, with the
//hypothesis of rounding the non integer values
sig Farm{
location: one Address,
size: one Int,
owner: one Farmer,
}{size>0}

sig Address{
referredTo: one Farm
}

sig Product{
cultivatedBy: some Farmer
}

sig Suggestion{
uploadedBy: one Farmer,
visualizedBy: set Farmer
}

sig HelpRequest{
requestedBy: one Farmer,
answeredBy: one Farmer
}{!(requestedBy = answeredBy)}

sig DiscussionForum{
writers: some Farmer
}

sig Message{
receiver: one Farmer,
sender: one PolicyMaker
}

// for simplicity reasons,  the quantity is expressed as an Int representing the liter
//for sqaure meters, with the hypothesis of rounding the non integer values
sig WaterConsumption{
quantity: one Int,
causedBy: one Farmer,
analyzedBy: set PolicyMaker
}{quantity > 0}
```

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

```
enum Status{
GoodStatus,
NormalStatus,
BadStatus
}

sig PolicyMaker{
visualizedRankings: set Ranking
}

abstract sig Ranking{
relatedTo: one Product,
includes: set RankingPosition
}

sig ImprovementRanking extends Ranking{}

sig RankingPosition {
position: one Int,
relatedTo: one Ranking,
referredTo: one Farmer
}{position>0}

//if to a farmer is associated a water consumption, this water consumption has to be associated
//only with the same farmer
fact singleFarmerWaterConsumptionAssociation{
	all w: WaterConsumption, f:Farmer | w.causedBy = f iff f.waterConsumption = w
}

//if to a farmer is associated a farm, this farm has to be associated only with the same farmer
fact singleFarmerFarmAssociation{
	all farmer: Farmer, farm:Farm | farmer.property = farm iff farm.owner = farmer
}

//if to a farm is associated an address, this address has to be associated only with the same farm
fact singleFarmerWaterConsumptionAssociation{
	all f: Farm, a:Address | a.referredTo = f iff f.location = a
}

//A suggestion can be uploaded only by a farmer who has a good status
fact UploadedOnlyByGoodFarmers{
	all s:Suggestion, f: Farmer | s.uploadedBy=f
	implies f.status=GoodStatus
}

//A help request can be  answered only by a farmer who has a good status
fact AnsweredOnlyByGoodFarmers{
	all h:HelpRequest, f: Farmer | h.answeredBy=f
	implies f.status=GoodStatus
}

//every Improvement Ranking is linked to a different product
fact onlyoneImprovementRankingForProduct{
	no disj r1,r2: ImprovementRanking | (r1.relatedTo = r2.relatedTo)
}
```

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

```
//Each farmer has to be accociated only to one ranking position for each ranking in which he participates
fact onlyOneRankingPositionForAFarmerInARanking{
	all disj rp1,rp2: RankingPosition | rp1.relatedTo = rp2.relatedTo
	implies rp1.referredTo != rp2.referredTo
}

// in this way it is assured that if a ranking position is related to a ranking, it appears also in the
//set of RankingPosition associated to the Ranking
fact associationBetweenRankingAndRankingPosition{
	all rp:RankingPosition, r:Ranking | rp.relatedTo = r
	implies rp in r.includes
}

// in this way it is assured that a ranking position appears in the set of only a ranking
fact noOverlappingRankingPosition{
	all rp: RankingPosition, disj r,r1: Ranking | rp.relatedTo = r
	implies rp not in r1.includes
}

// in this way two RankingPositions related to the same ranking cannot have the same position number
fact consistentPositions{
	(all disj p1,p2: RankingPosition | (p1.relatedTo = p2.relatedTo)
	implies p1.position != p2.position)
}
// in this way the number of elements in an improvement ranking related to a product is the same of the
//number of farmers who cultivates that product.
fact consistentSizeOfImprovementRanking{
	(all p: Product, r:ImprovementRanking | r.relatedTo = p
	implies #r.includes = #p.cultivatedBy)
}

//in this way every ranking position related to a specific ranking (and then to a specific product) can be
//associated only to a farmer who produces that product.
fact consistentLinkBetweenFarmerAndPosition{
	(all p: Product, r:ImprovementRanking, f: Farmer | (r.relatedTo = p and f in p.cultivatedBy)
	implies f in r.includes.referredTo )
}

//in this way it is mandatory to associate an improvement ranking for each product
fact oneImprovementRankingForProduct{
	all p:Product | (let ir = {r:ImprovementRanking | r.relatedTo = p} | #ir = 1)
}

//in this way the value of a position related to an ImprovementRanking is smaller or equal to the number
//of farmers who cultivates that product (for instance, if 10 farmers cultivate tomatoes, none of them
//can be in the 11th position in the respective ranking)
fact consistentPositionsUpperBoundImprovementRankings{
	all p: Product, rp:RankingPosition | rp.relatedTo.relatedTo = p
	implies rp.position<= #p.cultivatedBy
}


pred show{}

run show for 10 but exactly 4 Farmer, exactly 2 Product
```

Figure 20: Alloy code.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

4.2 Results and Generated instances

4.2.1 First run

Initially, the predicate "show" it has been run for a specific number of farmers and products in order to visualize a significant representation of the world.

```
Executing "Run show for 10 but exactly 4 Farmer, exactly 2 Product"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20
  26777 vars. 1800 primary vars. 59005 clauses. 41ms.
  Instance found. Predicate is consistent. 131ms.
```

Figure 21: Result obtained by running the Alloy model.

Below is reported an instance of the world obtained with the above specified execution of the Alloy model. Subsequently, in order to analyze better the validity of the constraints imposed by the definition of facts, there are provided some relevant zoomed screens.

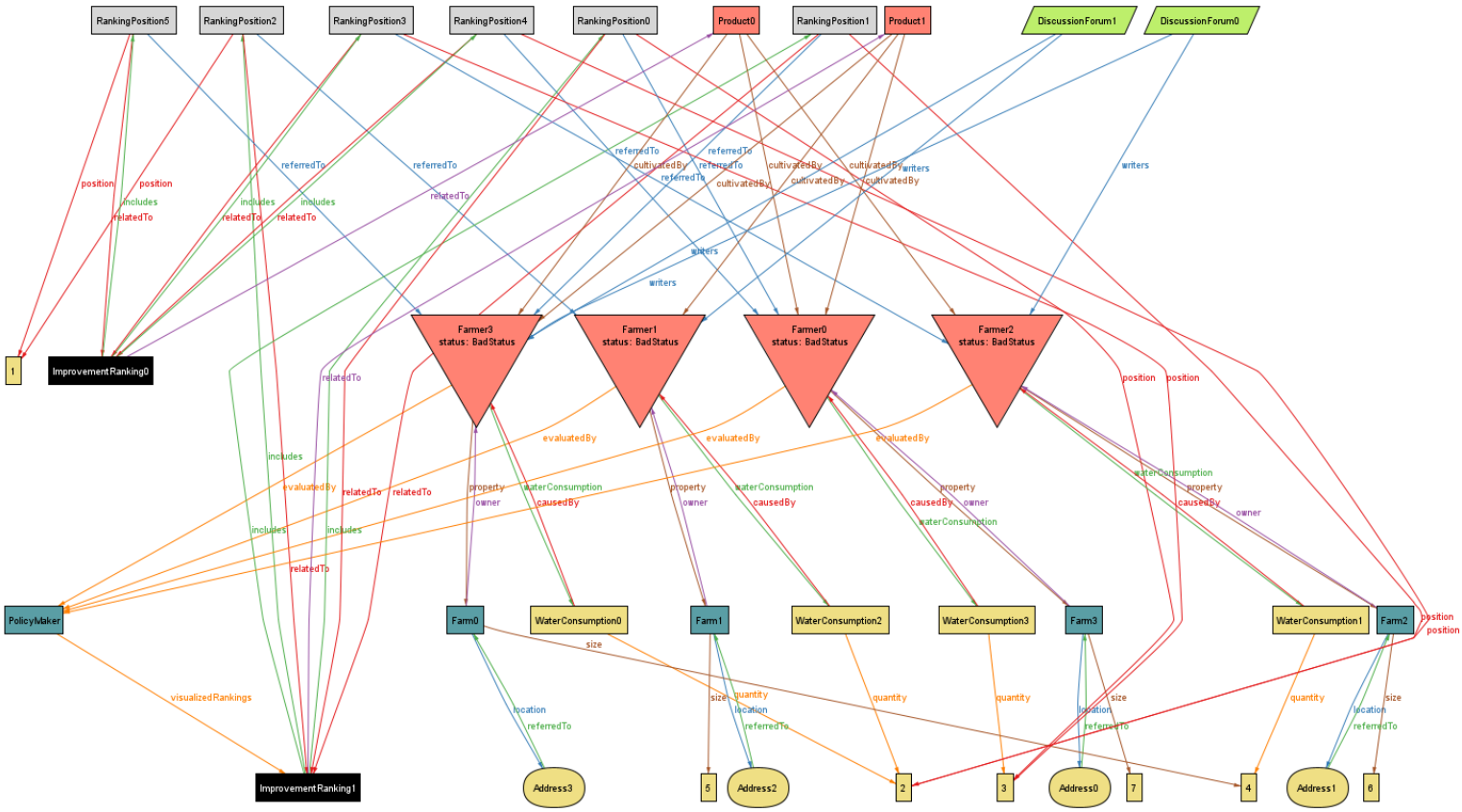

Figure 22: Instance of the world obtained by running the Alloy model.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

In the figure 23 it is highlighted that the product0 is cultivated by three farmers: farmer0, farmer2, and farmer3. While, in the figure 24, it is shown that such product is related to the ranking denominated as improvementRanking0.

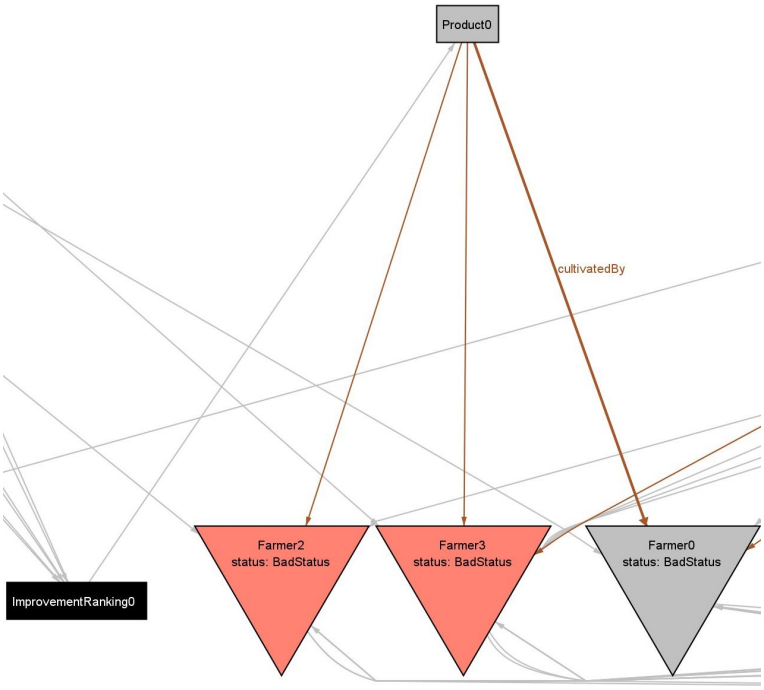

Figure 23: Highlighting of the relation between farmers and product cultivated.

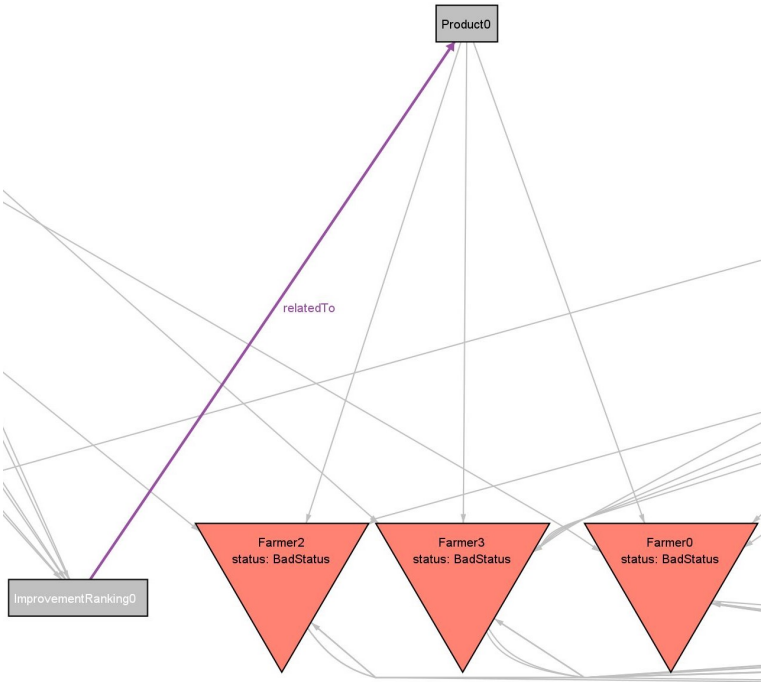

Figure 24: Highlighting of the relation between the product and the correspondent ranking.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

It can also be noticed in the figure 25 that this ranking is rightly associated to three ranking positions, which are referred to the three farmers mentioned above that cultivate the Product0 (figure 26).

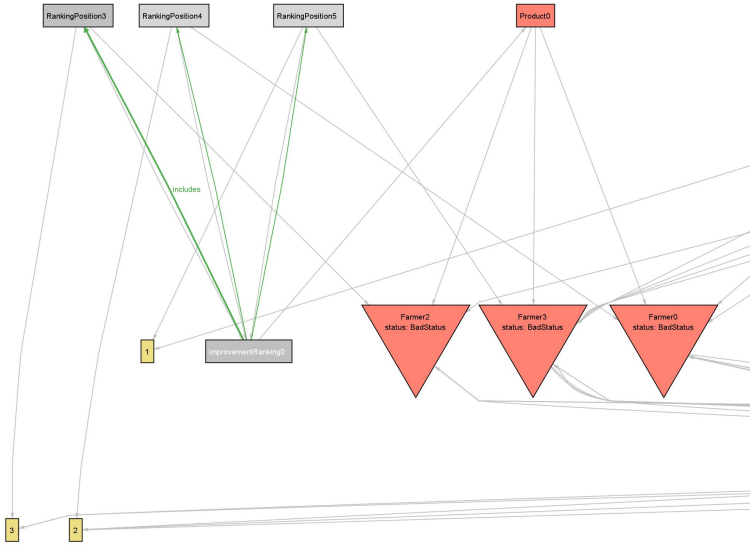

Figure 25: Highlighting of the relation between the rankings and the respective ranking positions.

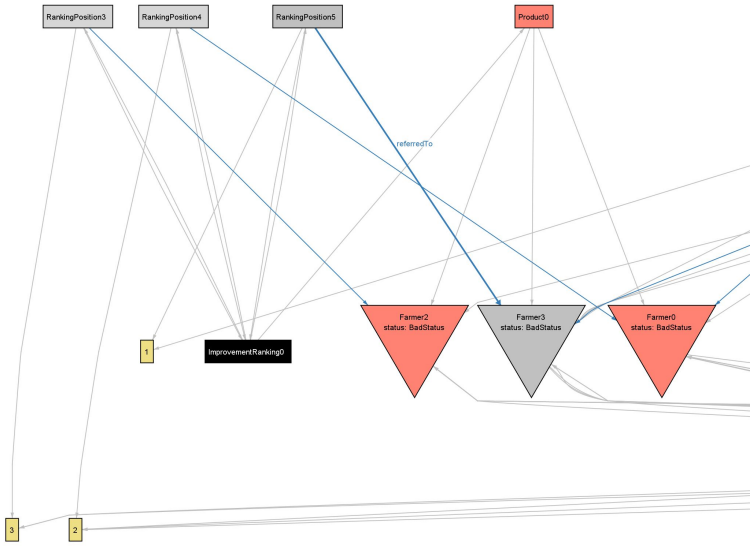

Figure 26: Highlighting of the relation between the ranking positions and the farmers to which they are referred to.

Finally, in the figure 27 it is shown that the ranking positions correspond to the numbers from 1 to 3, this is consistent with the fact that the farmers that cultivate that product are only three.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

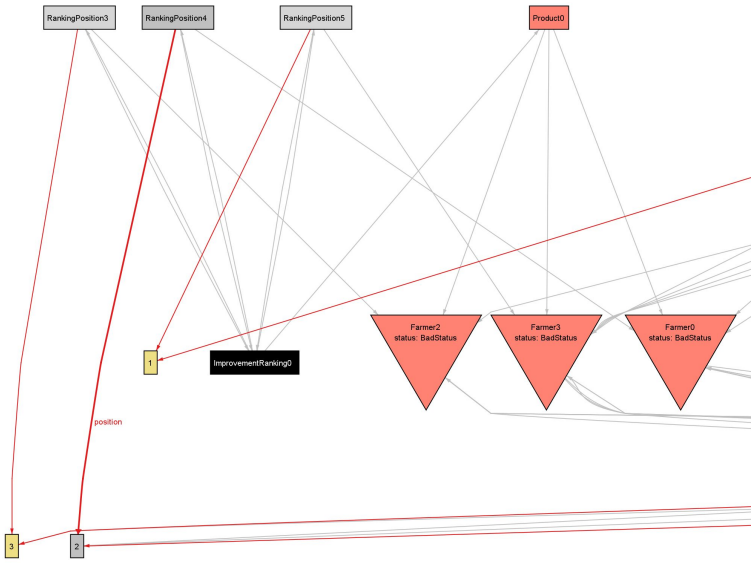

Figure 27: Highlighting of the relation between the ranking positions and the correspondent numbers.

In conclusion, it can be observed that all the constraints stated in the facts regarding the rankings are respected in the world generated and make the world consistent with the specifications of the RASD document.

For what concerns other aspects of the system, the figure 28 highlights the fact that a policy maker evaluates the farmers assigning to them a status. While, in the figure 29 it is shown how different farmers can write into a discussion forum.

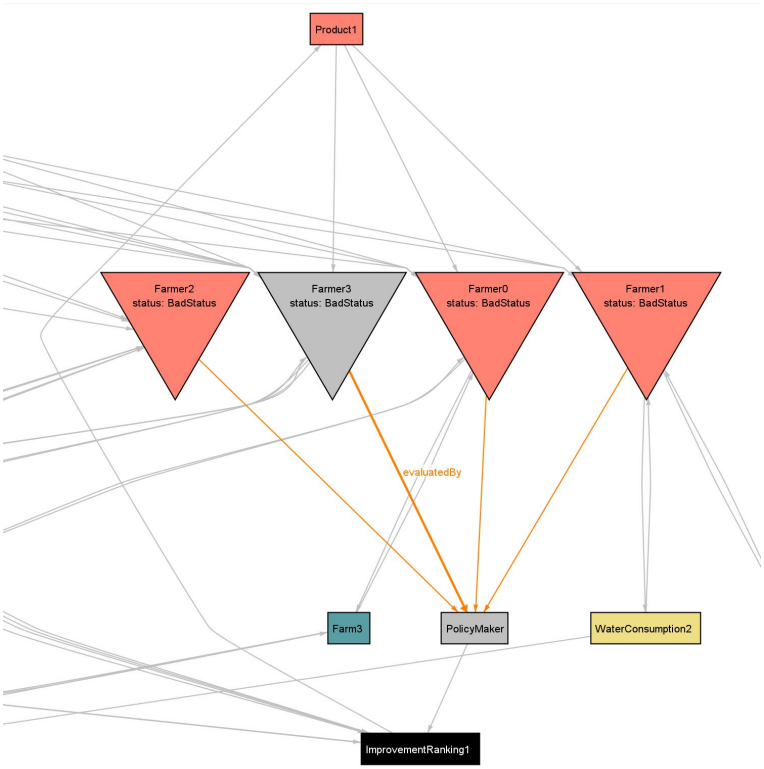

Figure 28: Highlighting of the relation between the farmers and the policy maker who evaluates them.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

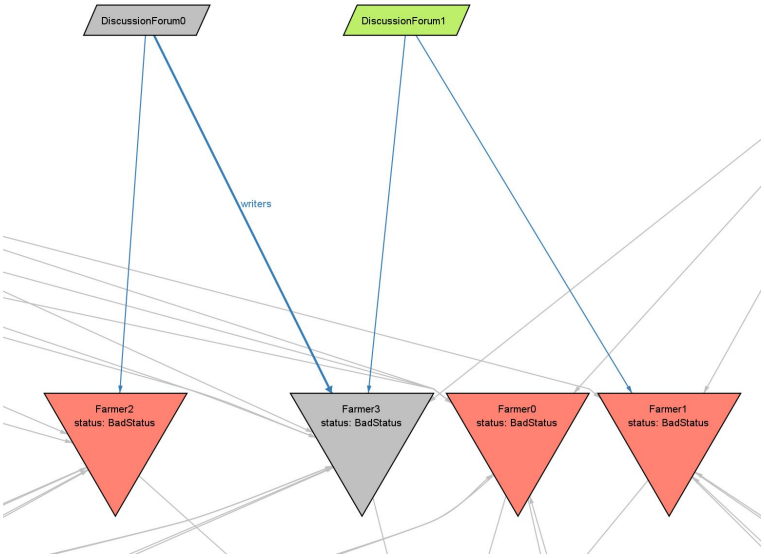

Figure 29: Highlighting of the relation between the farmers and the discussion forums in which they write.

4.2.2 Second run

The figure below represents a second instance of the world, generated by the same code previously described. It can be easily observed that all the constraints discussed in the previous section continue to be valid. In addition, there are a few entities that weren't generated in the first run and that can lead to some observations. Also in this case, it is made use of some zoomed screens in order for the relations between the various objects of the diagram to be more visible.

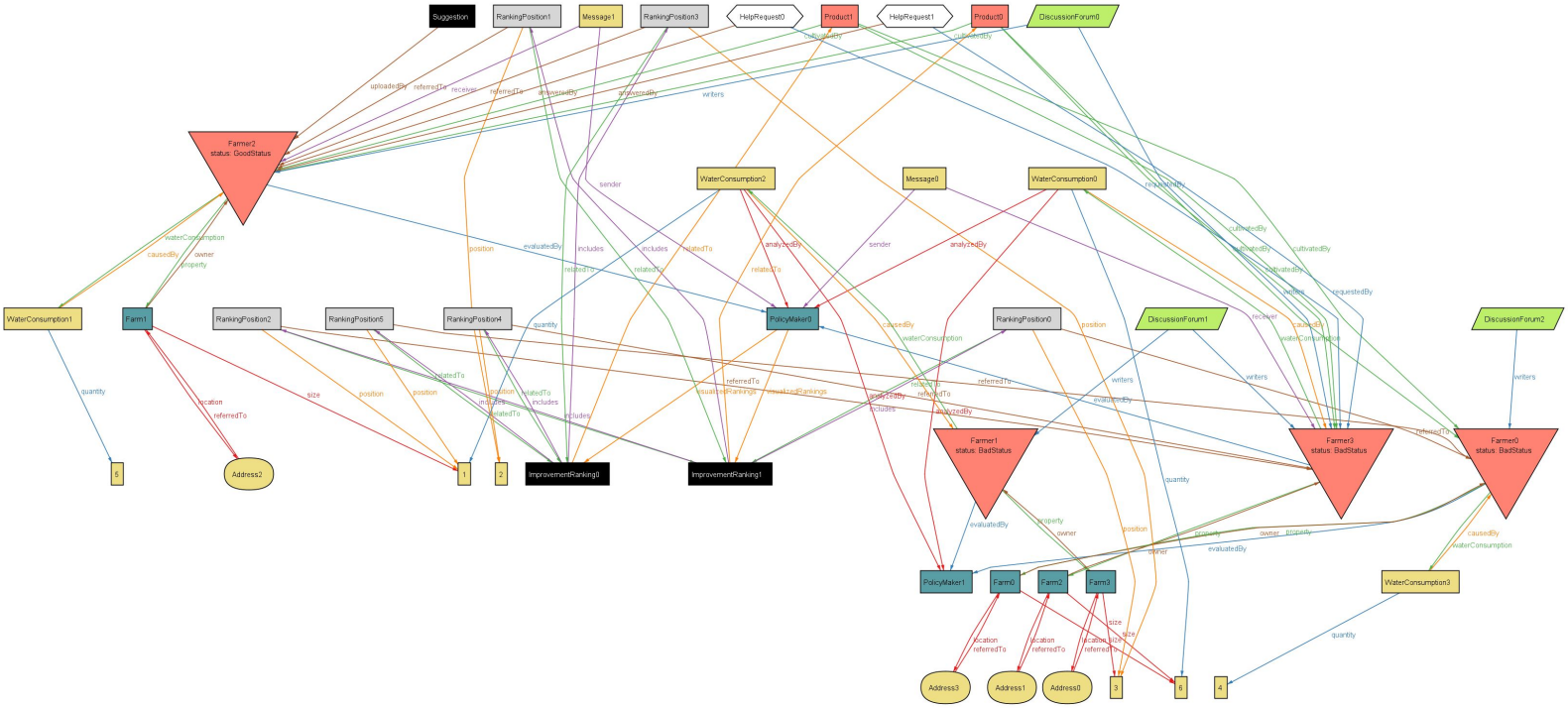

Figure 30: Second instance of the world obtained by running the Alloy model.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

In the figure 31, it can be observed that the message indicated as Message1 is sent by a policy maker (PolicyMaker0) and it is received by a farmer (Farmer2). This result is consistent with what has been designed in the project and it is imposed in the facts of the alloy model.

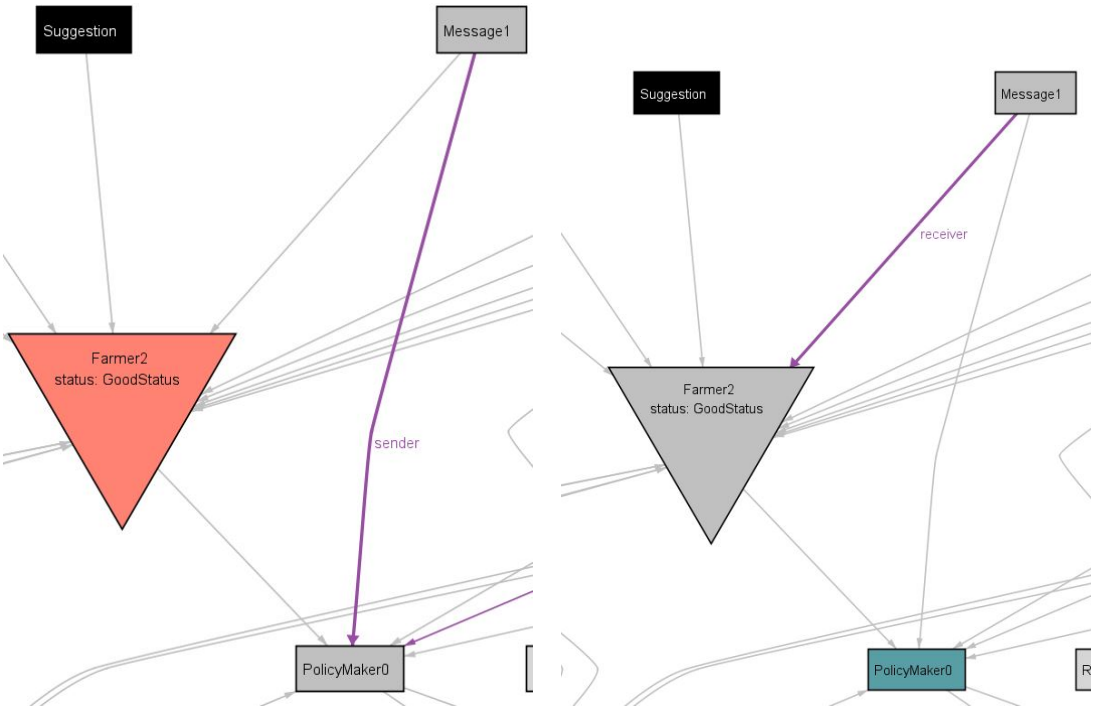

Figure 31: Highlighting of the relation between a message an the respective sender and receiver.

In the figure 32 it is shown that the HelpRequest0 is answered by a farmer who has a good status. This is consistent with the model and respects the imposed constraints.

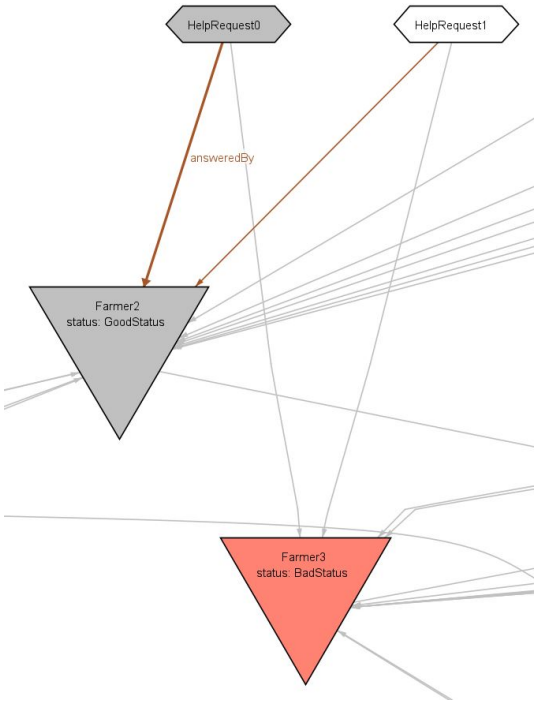

Figure 32: Highlighting of the relation between a HelpRequest and the farmer who addresses it.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

In the figure 33 it is shown that a suggestion is uploaded by a farmer who has a good status; also in this case, the constraints are respected.

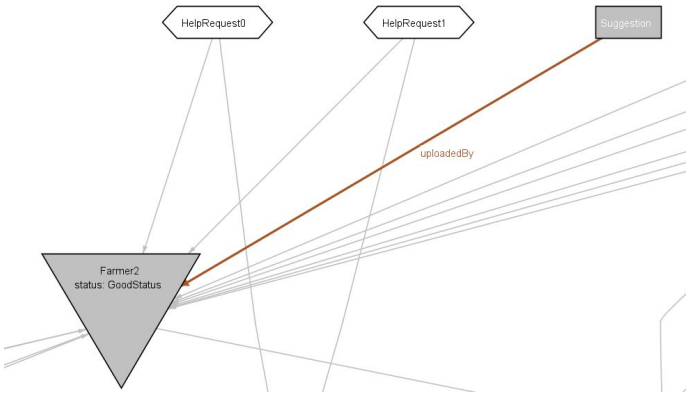

Figure 33: Highlighting of the relation between a suggestion and the farmer who uploads it.

4.2.3 Third run

Below it is reported another instance generated by running the code. This case will not be analyzed in detail, but it can be observed that all the relations are consistent with the model for the reasons already specified.

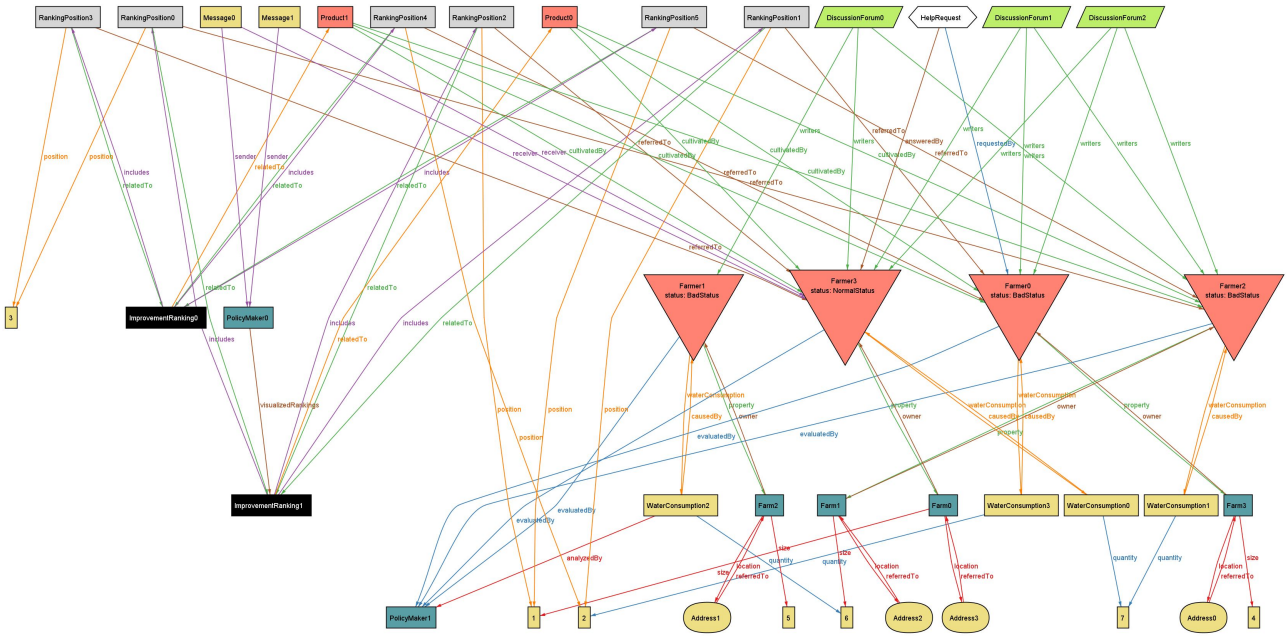

Figure 34: Third instance of the world obtained by running the Alloy model.

In conclusion, after several other executions of the code, it was always obtained as a result an instance of the world consistent with the model and with the features of the system that are described in this specification document.

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

5 Effort Spent

- Student 1

Topic	Hours
General reasoning	6:00h
Purpose & Scope	4:00h
Phenomena	3:00h
Scenarios	1:30h
Class diagram	2:00h
Statechart	2:00h
Product functions	1:00h
Domain assumptions	1:00h
User interfaces	2:00h
Hardware, software and communication interfaces	1h
Functional requirements	1:30h
Use case & Use case Diagram	5:00h
Sequence Diagrams	3:00h
Design constraints and software attributes	2h
Alloy and alloy documentation	8:30
Document organization	4:30
Total effort spent	**48:00h**

Table 22: Effort spent by student 1

- Student 2

Topic	Hours
General reasoning	6:00h
Purpose & Scope	4:00h
Phenomena	3:00h
Scenarios	1:30h
Class diagram	1:00h
Statechart	1:00h
Product functions	2:00h
Domain assumptions	3:00h
User interfaces	1h
Hardware, software and communication interfaces	2:00h
Functional requirements	4:00h
Use case & Use case Diagram	3:00h
Sequence Diagrams	1:30h
Design constraints and software attributes	2h
Alloy and alloy documentation	8:30
Document organization	4:30
Total effort spent	**48:00h**

Table 23: Effort spent by student 2

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved

6 References

- All the diagrams have been made with Draw.io
- Alloy code was made on the specific Analyzer tool
- All the user interfaces have been made with the Storyboard of Xcode

Copyright © 2021, Federica Tommasini and Alessandro Verosimile – All rights reserved