



**POLITECNICO**  
MILANO 1863

M.Sc. Computer Science and Engineering  
Software Engineering 2 Project

# DREAM – Data dRiven PrEdictive FArMing in Telangana

## Requirement Analysis and Specification Document



Censuales Simone, Giovia Giuseppe, Meli Giuseppe

23rd December 2021

Version 1.0

GitHub Repository - <https://github.com/Censu08/CensualesGioviaMeli>

## Contents

<b>Chapter 1: Introduction.....</b>	<b>4-8</b>
1. Purpose.....	4-5
1.1 Goals.....	5
2. Scope .....	6
2.1 World and Shared phenomena.....	6
3. Definitions, Acronyms, Abbreviations.....	7
3.1 Definitions .....	7
3.2 Acronyms .....	7
3.3 Abbreviations .....	7
4. Revision history.....	8
5. Reference Documents .....	8
6. Document Structure.....	8
<b>Chapter 2: Overall Description .....</b>	<b>9-15</b>
1. Product perspective .....	9
1.1 Scenarios .....	9
1.2 Class diagram .....	10
1.3 State Chart.....	11
2. Product Functions .....	12-13
3. User characteristics .....	14
4. Assumption, dependencies and constraints .....	15
4.1 Domain Assumption.....	15
4.2 Dependencies .....	15
4.3 Constraints .....	15
<b>Chapter 3: Specific Requirements.....</b>	<b>16-74</b>
1. External Interface Requirements .....	16-40
1.1 User Interface .....	16-39
1.2 Hardware Interfaces.....	40
1.3 Software Interfaces .....	40
1.4 Communication Interfaces.....	40
2. Functional Requirements .....	41-72
2.1 Use Case Diagrams.....	41-42
2.2 Use Case Analysis.....	43-53
2.3 Sequence Diagrams .....	54-66

2.4 Requirements.....	67-69
2.2 Traceability Matrix.....	70-72
3. Performance Requirements.....	73
4. Design Constraint .....	73
4.1 Standards compliance .....	73
4.2 Hardware limitations .....	73
4.3 Any other constraint .....	73
5. Software System Attributes .....	74
5.1 Reliability.....	74
5.2 Availability .....	74
5.3 Security .....	74
5.4 Maintainability .....	74
5.5 Portability .....	74
<b>Chapter 4: Formal Analysis using Alloy.....</b>	<b>75-85</b>
1. Alloy model.....	76-40
1.1 Signatures.....	76-78
1.2 Facts .....	79-80
1.3 Assertions .....	81
1.4 Analysis results.....	82
1.4 Alloy generated Worlds.....	83-85
<b>Chapter 5: Effort Spent.....</b>	<b>86-87</b>
<b>Chapter 6: References .....</b>	<b>88</b>

# Chapter 1

## Introduction

The requirement analysis and specification document (RASD) main purpose is to specify the domain, the entities, including their relationships, concerning the application as well as to analyse the requirements and explain the goals, the constraints and all the features implemented in the software to be.

### 1. Purpose

With the arrival of the global pandemic caused by COVID-19, the agricultural sector really realized the necessity to implement a software solution to manage the relationship between the personnel of the sector itself.

DREAM has a clear objective: the development of management model that is able to predict the food requirements of the entire nation and build a resilient food system, exploiting digital goods, adopting innovative methodologies and techniques, and using a community-centred approach that would strengthen the data driven policy across the country.

The software will allow its users to send Telangana different kinds of data and information, such as:

- Weather forecast, both short and long term
- Farmer productions
- Measurements and monitoring provided by sensors in the ground and water irrigation system

In doing so, DREAMS will allow the management and the interaction between three different kinds of users:

- **Farmers:** they can visualize and/or insert relevant data concerning:
  - Locations and the typology of product
  - Weather forecast
  - Crop-related suggestions
  - Furthermore, they are able to interact with both other farmers, through discussion and exchanging of knowledge, and agronomist, who are trying to solve some issue they are facing
- **Policy makers:** they identify those farmers who are either particularly well-performing or in need of help.
- **Agronomists:** they can visualize, manage data about the region they are responsible of. Moreover, they can both confirm and update daily plan about farm to visit during a particular day, especially those which are under-performing, assuming that every farm in their region must be visited at least twice a year.

### **1.1. Goals**

- [G.1] Allows policy makers to identify well performing farmers
- [G.2] Allows policy makers to identify farmers in need of help
- [G.3] Allows policy makers to evaluate Agronomist's steering initiative
- [G.4] Allows farmers to visualize relevant data based on location and type of production
- [G.5] Allows farmers to insert in the system data about their production and any problem they face
- [G.6] Allows farmers to ask for help and suggestions
- [G.7] Allows farmers to reply to help suggestions
- [G.8] Allows farmers to create discussions forum
- [G.9] Allows farmers to answer to discussions forum
- [G.10] Allows farmers to search specific discussions forum
- [G.11] Allows farmers to participate steering initiative
- [G.12] Allows farmers to insert final result of steering initiative
- [G.13] Allows agronomists to insert the region they are responsible of
- [G.14] Allows agronomists to receive and answer requests for help
- [G.15] Allows agronomists to visualize data concerning both weather forecast and best performing farmer in a region
- [G.16] Allows agronomists to visualize, update and confirm daily plans to visit farms

## 2. Scope

In accordance with the World and Machine model, conceived by M. Jackson and P. Zane, we can find a correlation between the Machine with the DREAM system to be developed, and the World with Telangana's context. Thanks to this separation it is possible to identify two different categories of phenomena.

### 2.1. World and Shared phenomena

**World Phenomena**, events that take place in the real world and that the machine cannot observe.

- An agronomist is going to visit a farm
- A farmer is in need of help
- A farmer receives special incentives, as he/she is performing particularly well

**Shared Phenomena**, events that involve both the real world and the machine. They can happen in real world and be observed by the machine, or they could take place inside the machine and have an impact in the real world.

**Shared Phenomena: controlled by the world and observed by the machine**

- A farmer signs up or logs in to the DREAM application
- A farmer requests for help
- A farmer answers the suggestion of help
- A farmer inserts data about his/her production or any problem he/she is facing
- A farmer visualizes and checks data relevant to him/her, such as weather forecast and suggestion on specific crops and/or fertilizer to use
- A farmer creates discussion forums with other farmers
- A farmer answers discussion forums
- A farmer searches for specific discussion forum
- A farmer marks a discussion as favourite
- A farmer visualizes all the discussion marked as favourite
- A farmer visualizes and participates to steering initiatives
- A farmer inserts final result of steering initiative to which he has participated
- A farmer visualizes data measured by sensors
- An agronomist signs up or logs in to the DREAM application
- An agronomist updates and/or confirm daily plan to visit farm
- An agronomist answers help request
- An agronomist creates
- A policy maker signs up or logs in to the DREAM application
- A policy maker identifies well performing farmers
- A policy maker identifies bad performing farmers in need of help
- A policy maker evaluates steering initiative

**Shared Phenomena: controlled by the machine and observed by the world,**

- The system notifies an agronomist or a farmer about a farmer asking for help
- The system notifies a farmer about response of an agronomist
- The system notifies a farmer if some other farmer has published an answer on his discussion

### 3. Definitions, Acronyms, Abbreviations

#### 3.1 Definitions

- **Telangana's Policy Makers:** a person who has access to the DREAM Application data and who manages farmers and interacts with agronomists
- **Farmers:** a person who has access to the DREAM Application data and is able to manage data about his production, create and interact with discussion, send help request to agronomist and other farmers, participate to steering initiative carried out by agronomist
- **Agronomists:** a person who is able to access DREAM Application data, answer help request, create steering initiative and visit farmer according to his/her daily plan
- **DREAM System:** refers to the whole system to be developed
- **DREAM Services:** refers to the functionalities offered by the DREAM System
- **DREAM Application:** refers to the application that makes DREAM Services available everywhere. In this document, this term is intentionally used in a generic way
- **External Services:** refers to weather forecast services and geo-localization services that the System accesses from third parties
- **Sensors:** refers to specific devices located in the ground, sending data about soil humidity and water irrigation to the System

#### 3.2 Acronyms

- **RASD:** Requirement Analysis and Specification Document
- **UML:** Universal Modelling Language, it is a way of designing application based on its graphic class representation
- **GPS:** Global Positioning System

#### 3.3 Abbreviations

- **[G.n]:** Goal number #n
- **[D.n]:** Domain assumption number #n
- **[R.n]:** Requirement number #n
- **[C.n]:** Constraint number #n

## 4. Revision history

Version	Date	Authors	Summary
1.0	18/12/2021	Censuale Simone Giovia Giuseppe Meli Giuseppe	First release

## 5. Reference Documents

- Specification document: Assignment RDD A.Y. 2021/2022.pdf
- Software engineering 2 course slide
- Previous project example (given by professors):
  - o Specification document: Project assignment A.Y. 2020/2021.pdf
  - o RASD to be analyzed.pdf
- IEEE Standard on requirement engineering (ISO/IEC/IEEE 29148)

## 6. Document Structure

The document contains the following sections:

- **Introduction:** A brief yet exhaustive introduction to the main purposes and scope of the software-to-be
- **Overall Description:** introductory section to the more detail explanatory section 3, concerning a general description of the final product
- **Specific requirements:** scenarios, use cases, diagrams and activity diagrams are given to provide an explanation of all kind software requirements (including functional and non-functional)
- **Formal analysis using alloy:** this section shows the interaction between parties, using alloy code, in order to prove both the correctness and validity
- **Effort spent:** section that shows the total amount of effort spent by each member of the group on every topic of the document
- **References:** references to all the documents used as a model for the realization of the project.

# Chapter 2

## Overall Description

### 1 Product perspective

#### 1.1 Scenarios

- *Mohammed, the farmer in need of help:*

The harvest of Mohammed has been badly compromised by the climatic change in his region. Feeling his desperation, a farmer friend of his introduce him to the DREAM platform. Once registered and logged in, Mohammed is able to start a help request, sent to both the local agronomist and the farmers nearby.

- *The generous policy maker:*

Vanessa, a policy maker, decides to reward the best performing farmers of the past six months, especially considering the averse climatic conditions they are experiencing. So, she logs in and, by analysing the data collected through the platform, she is able to identify the best three performing farmers and select them as candidates for the special incentives.

- *A reminder from the software:*

Giovanni, a young agronomist located in the south of India, logs in the DREAM platform and starts visualizing the data of the best performing farmers in his local region. Suddenly, the system sends an alert about the daily plan expected for that particular day. Giovanni confirms the daily plan, closes the system and [starts to go](#).

- *Strength comes with Union:*

Johnny, a senior policy maker, needs to compile the list of best and worst performing farmers, on a month basis. Once done compiling the list, the system will notify the local agronomists in due time, about the under-performing farmers in their region of competence. Now, each one of them can schedule a visit in order to try and solve any possible problem farmers are facing.

- *Do not google it, DREAM it:*

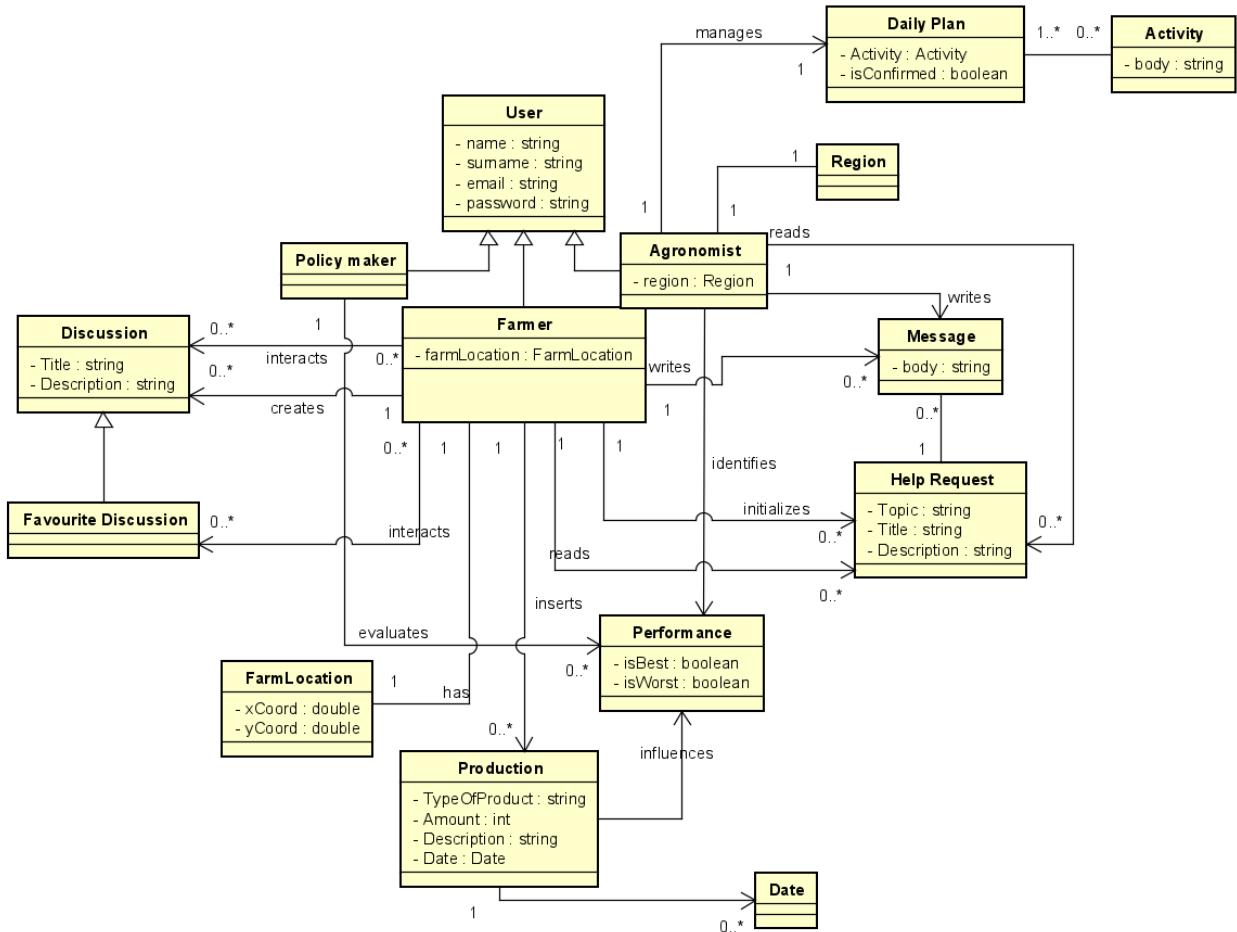
Giulio wants to know what is the best fertilizer for his soil type, so that he can improve his rice production in the winter. Googling it, he cannot find anything relevant for his concern, so he remembers that DREAM platform offers the possibility to create a discussion in the forum, allowing the exchange of data between farmers and agronomists. So, he logs in and starts a discussion, waiting for anybody to answer.

- *Success comes from analysis of trends:*

Marcello is a policy maker in charge of analysing the outcome of steering initiatives carried out by farmers and/or agronomists. He is able to do so by comparing the data collected by the DREAM platform before the start of the initiative, and after its completion, filtering data in specific periods. If positive significant results are produced, he can broadcast the outcome to the farmers interested in the topic.

## 1.2 Class diagram

This is the section where the Class Diagram of the System is displayed in details, presenting a high-level view of the main actors and concepts involved in the software-to-be.



### 1.3 State Chart

In this section, state charts related to the state transitions of a new help request, a new steering initiative and an evaluation of a farmer.



Figure 1: State Chart Help Request

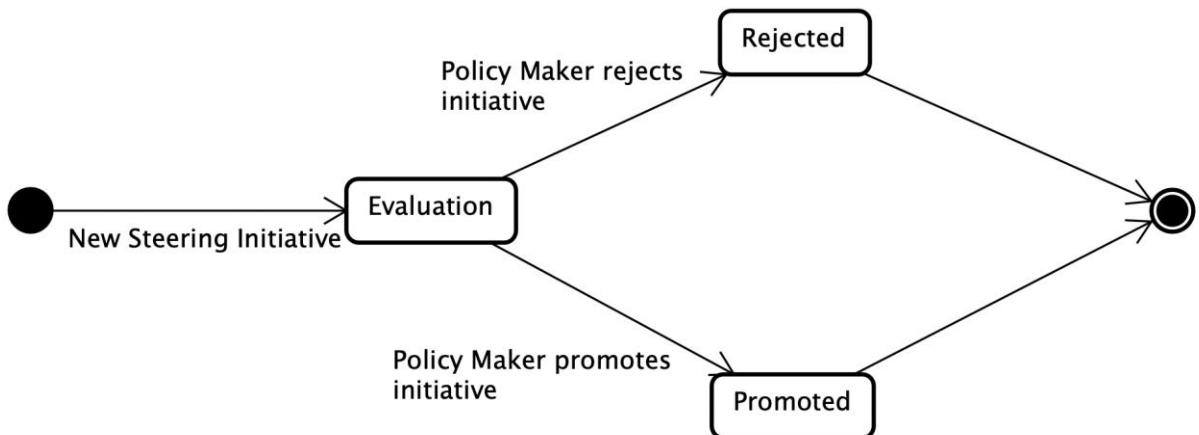


Figure 2: State Chart Steering Initiative

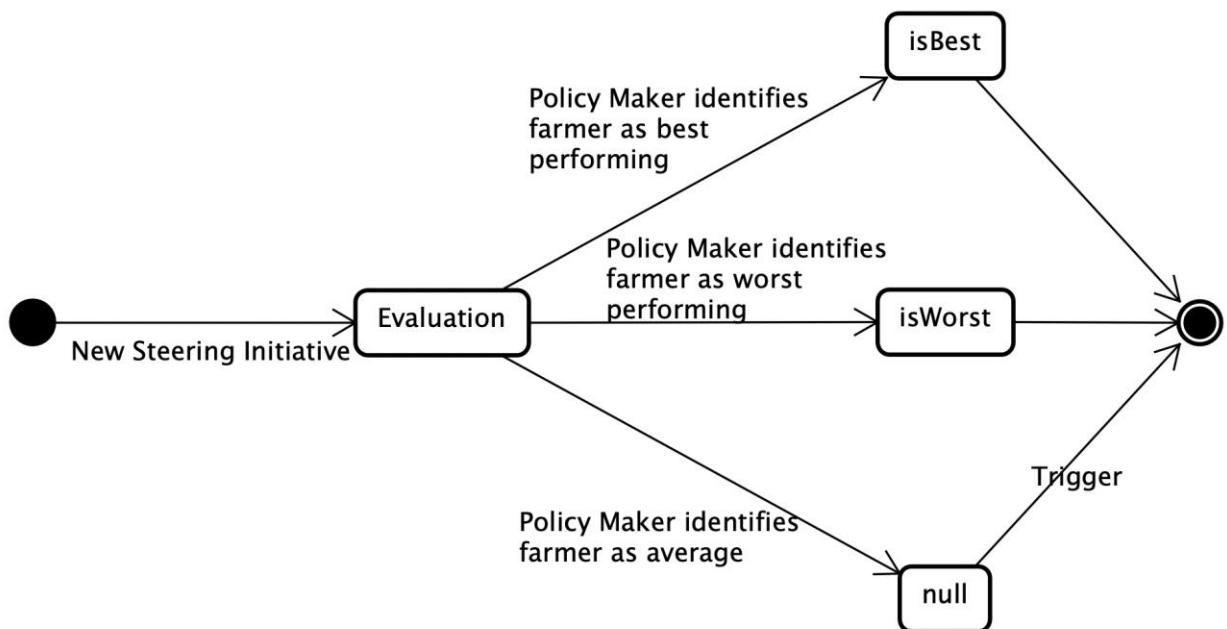


Figure 3: State Chart Farmer Evaluation

## 2 Product Functions

- **Farmer registration:** This feature will enable Farmer to register and take full advantage of all services offered by DREAM Application. The registration will consist of the compilation of a form. It will specially be designated for Farmer, once has been selected “Farmer” box. Then, to complete the registration stage, he/she will have to enter all the required information. It is necessary that the Farmer fills the form with all his/her personal requested data. Finally, it will be necessary to accept the Term of Use and Privacy Policy of the Application to complete registration.
- **Agronomist registration:** This feature will enable Agronomist to register and to take full advantage of all services offered by DREAM Application. The registration will consist of the compilation of a form. It will specially be designated for Agronomist, once has been selected “Agronomist” box. Then, to complete the registration stage, he/she will have to enter all the required information. It is necessary that the Agronomist fills the form with all his/her personal requested data. Finally, it will be necessary to accept the Term of Use and Privacy Policy of the Application to complete registration.
- **Policy Maker registration:** This feature will enable Policy Maker to register and to take full advantage of all services offered by DREAM Application. The registration will consist of the compilation of a form. It will specially be designated for Policy Maker, once has been selected “Policy Maker” box. Then, to complete the registration stage, he/she will have to enter all the required information. It is necessary that the Policy Maker fills the form with all his/her personal requested data. Finally, it will be necessary to accept the Term of Use and Privacy Policy of the Application to complete registration.
- **Data insertion:** This feature will enable Farmer to insert data about their production and any problem they face in the System. There's going to be a dedicated section where Farmer will be capable of inserting data.
- **Farmer requests for help:** This feature will enable Farmer to request for help and suggestions by agronomists and other farmers. Particularly, when help request form is used, the agronomist of competence will be notified, so he will be able to assess support.
- **Farmer initializes a discussion:** This feature will enable Farmer to create a new discussion by submitting all the mandatory field in new discussion form.
- **Farmer answers a discussion:** This feature will enable Farmer to answer an already existing discussion on the “Discussion forum” section by submitting the response in the text input area.
- **Farmer looks for a specific discussion:** This feature will enable Farmer to search for a specific discussion through the insertion of a keyword. The System will perform the match of the string and Farmer will visualize every discussion whom title contains that keyword.
- **Farmer marks a discussion as favourite:** This feature will allow Farmer to select a specific discussion and mark it as favourite, so that he will be able to visualize it fast in the “Favourite discussion” section.
- **Farmer visualizes all favourite discussions:** This feature will allow Farmer to visualize those discussions previously marked as favourite.
- **Farmer visualizes steering initiative:** This feature will enable Farmer to visualize all the steering initiative active in the “Steering Initiative” section.
- **Farmer participates to steering initiative:** This feature will allow Farmer to participate to a visualized and chosen steering initiative in the “Steering Initiative” section. Once joined, the Farmer will have to provide the initial data on which the steering initiative will be partially based on.
- **Farmer inserts final result of steering initiative:** This feature will enable Farmer to insert the final result obtained by applying the procedure shown in the steering initiative selected. Farmer has to provide the final result before the expiration time of the steering initiative itself.
- **Agronomist manages daily plan:** This feature will enable Agronomist to manage a daily plan to visit farms in the region. Particularly, Agronomist, by means of a visit list, can decide where to go; recalling that all farms must be visited at least twice a year, but those that are under-performing

should be visited more often. List will be managed by DREAM Application. Agronomist can modify the list to change his daily plan through the dedicated buttons.

- **Agronomist answers help request:** This feature will enable Agronomist to view all help request, select one of them and answer Farmer help request.
- **Agronomist publishes new steering initiative:** This feature will enable Agronomist to create new steering initiative filling out a form and publishing it.
- **Policy Maker reports performance:** This feature will enable Policy Maker to identify those farmers who are performing well, especially when they demonstrate to be resilient to meteorological adverse events. DREAM Application will show a list, where Farmers' stats are explained. Policy Maker, through the dedicated buttons, can select farmers who are performed well.
- **Policy Maker evaluates steering initiative:** This feature will enable Policy Maker to view all steering initiative, select one of them, view all its descriptions and promote or reject it.

### 3 User characteristics

1. **Farmer:** a single person who is registered to the DREAM Application and has access to all Services the Application offers to manage data, participate to steering initiative, interact with discussion forum and send help request
2. **Agronomist:** a single person who is registered to the DREAM Application and has access to all Services the Application offers to answer help request, manage daily plan and carry out steering initiative
3. **Policy maker:** a single person who is registered to the DREAM Application and has access to all Services the Application offers to evaluate both steering initiative outcome and farmer performance
4. **Unregistered user:** a single person who has not yet registered and is not allowed to interact with the DREAM Application

## 4 Assumption, dependencies and constraints

### 4.1 Domain Assumption

- [D.1] Farmers are supposed to insert only correct data
- [D.2] Sensors are supposed to be accurate and precise
- [D.3] Every user's email is unique
- [D.4] Farmers are supposed to own an internet capable device
- [D.5] Agronomists are supposed to own an internet capable device
- [D.6] Policy makers are supposed to own an internet capable device
- [D.7] The external service used by the system to check the weather forecast is supposed to be accurate
- [D.8] Agronomists are supposed to confirm, in the daily plan, only those farms they have visited
- [D.9] Agronomist is supposed to visit a farmer in need of help in his region
- [D.10] The internet connection is supposed to work flawlessly, without any failure
- [D.11] Agronomists are supposed to insert the region they are responsible of correctly
- [D.12] Farmers are supposed to insert final result of steering initiative before expiration time

### 4.2 Dependencies

- The system will use an external service to retrieve information about weather forecasts
- The system will use an external service to manage data coming from sensors
- The system will use the Internet connection of the Farmer's device
- The system will use the Internet connection of the Agronomist's device
- The system will use the Internet connection of the Policy maker's device
- The system will use an external service to take the time at the moment of any data insertion

### 4.3 Constraints

- [C.1] Every agronomist is supposed to visit at least twice a year every farm in his region
- [C.2] Each region is assigned to only one agronomist
- [C.3] Agronomist is supposed to confirm his daily plan before midnight

# Chapter 3

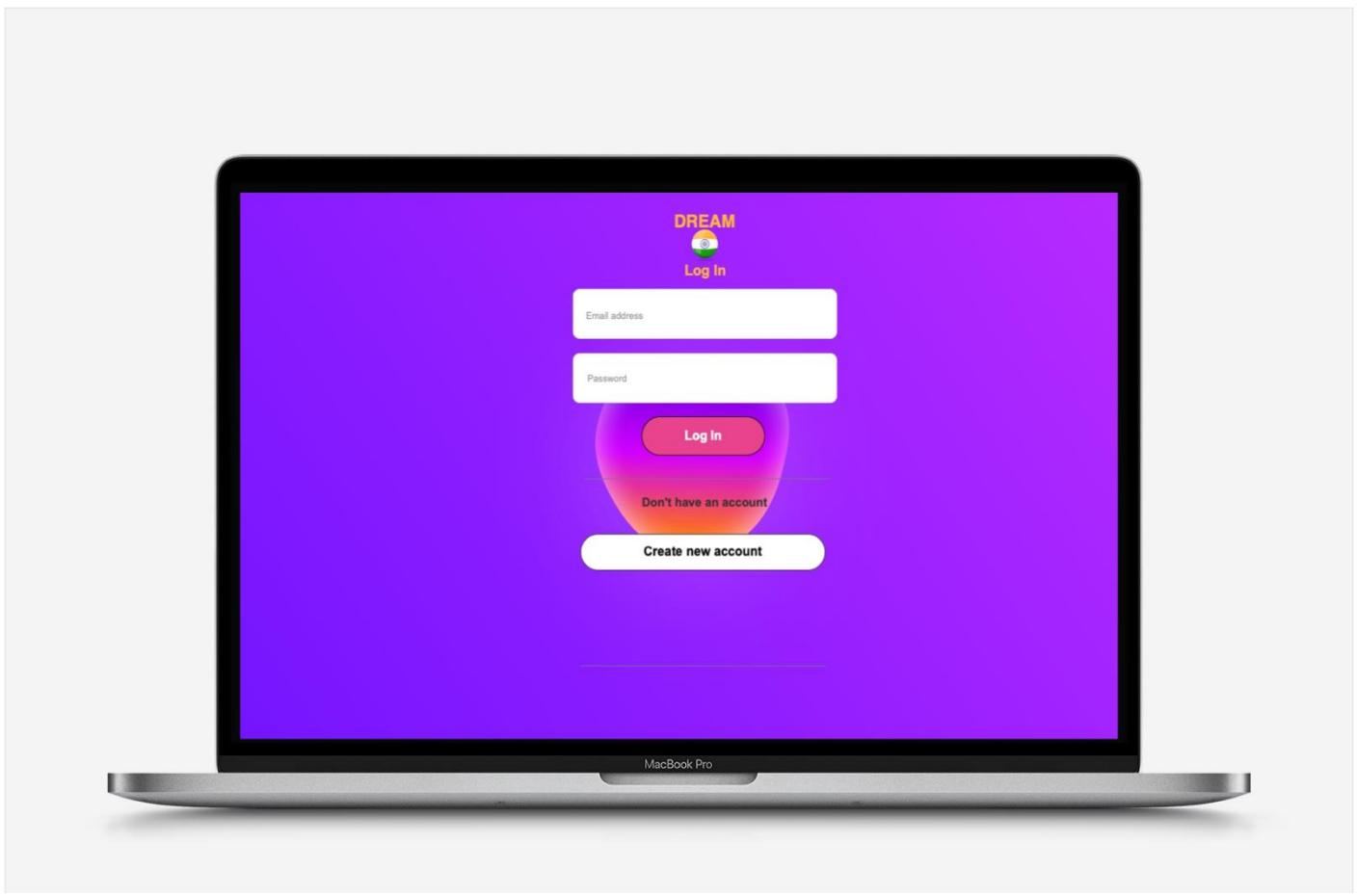
## Specific Requirements

### 1 External Interface Requirements

#### 1.1 User Interface

- **User Login Interface**

Users (Farmers, Agronomists and Policy Makers) can access DREAM application through the Login interface shown in *Figure 4*. To benefit DREAM Services, User must first sign in by inserting Email and the password entered during the registration. If User does not yet have a registered profile, he can create a new account by clicking on the “Create new account” button.



*Figure 4*

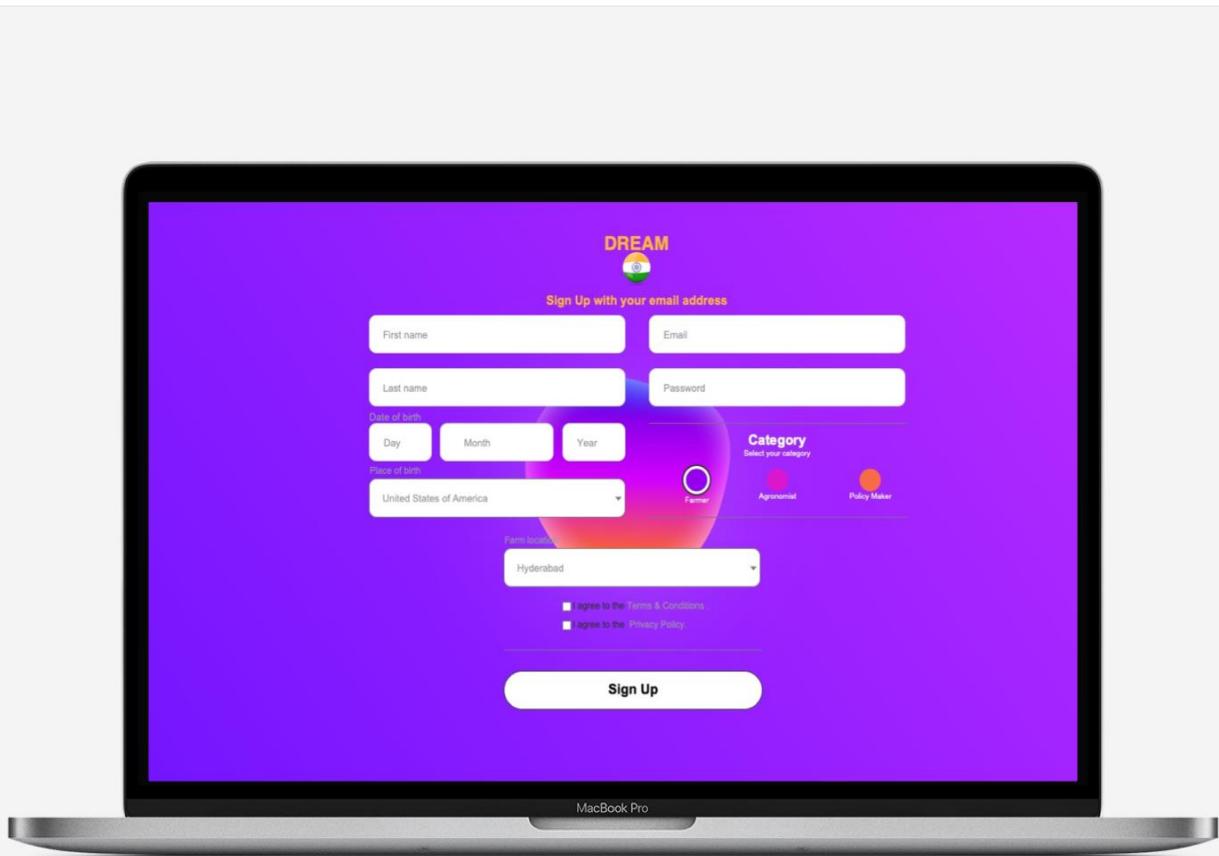
- **User Registration Interface**

Through the registration interface is possible to create a new account to access DREAM Services. There are three different registration interfaces through which can it be ensured the User registration as Farmer, Agronomist or Policy Maker. In order to register, by clicking on the appropriate category button, three different Registration Interfaces will be displayed. So User, to complete registration, can enter its personal data.

If User clicks “Farmer” button, he shall enclose: First name, Last name, Date of birth, Place of birth, Email, Password, Farm location and he must also accept the Terms and Conditions and the Privacy Statement through the appropriate controls (*Figure 5*).

If User clicks “Agronomist” button, he shall enclose: First name, Last name, Date of birth, Place of birth, Email, Password, Area of Responsibility and he must also accept the Terms and Conditions and the Privacy Statement through the appropriate controls (*Figure 6*).

If User clicks “Policy Maker” button, he shall enclose: First name, Last name, Date of birth, Place of birth, Email, Password and he must also accept the Terms and Conditions and the Privacy Statement through the appropriate controls (*Figure 7*).



*Figure 5*

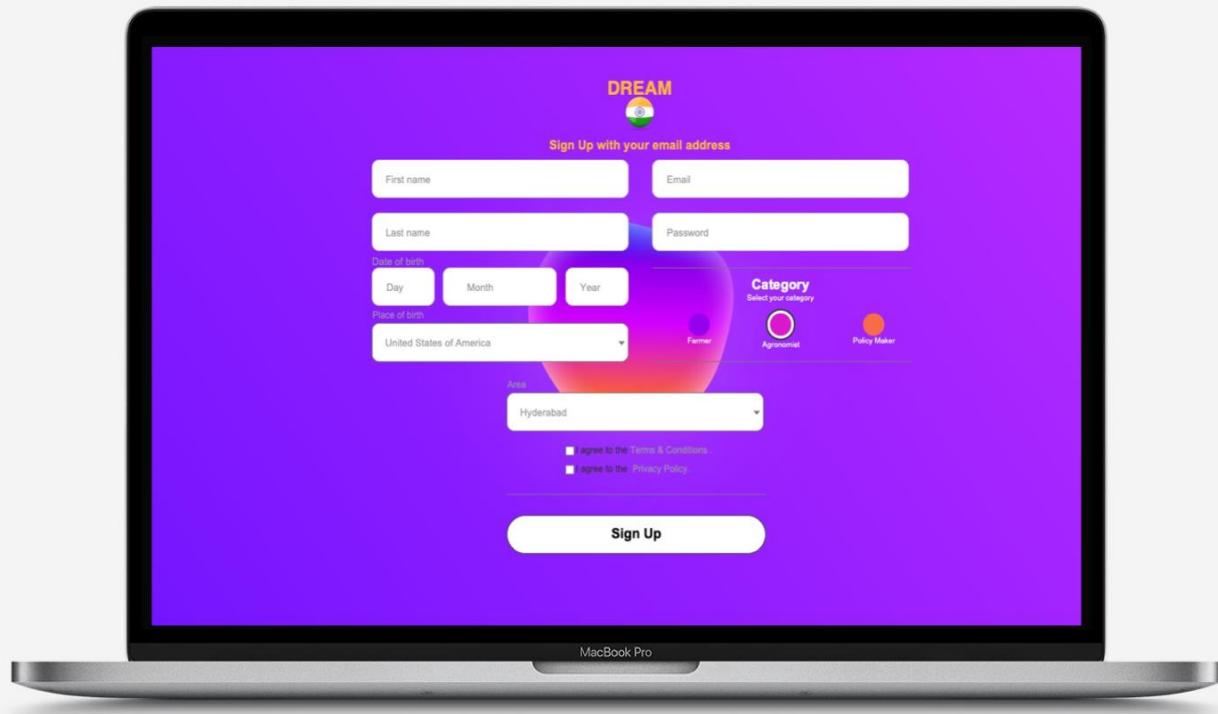


Figure 6

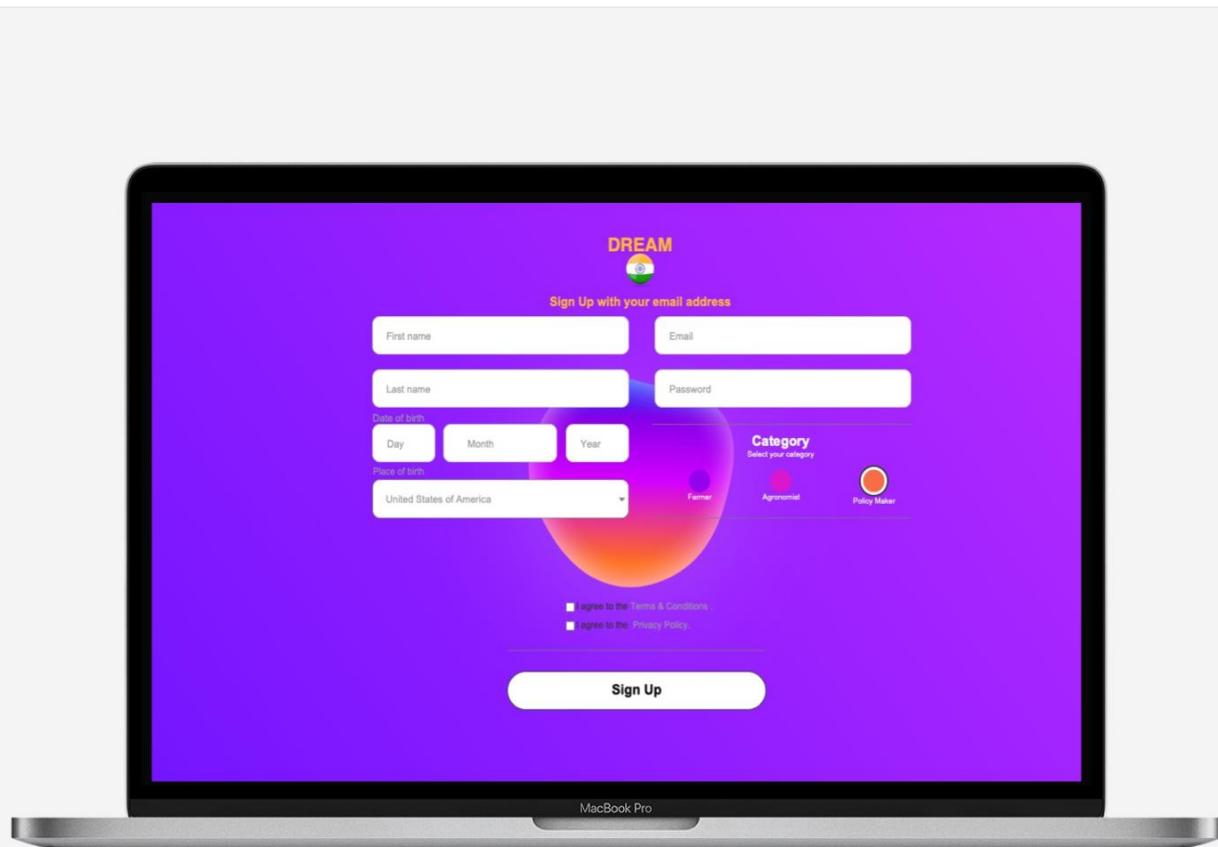
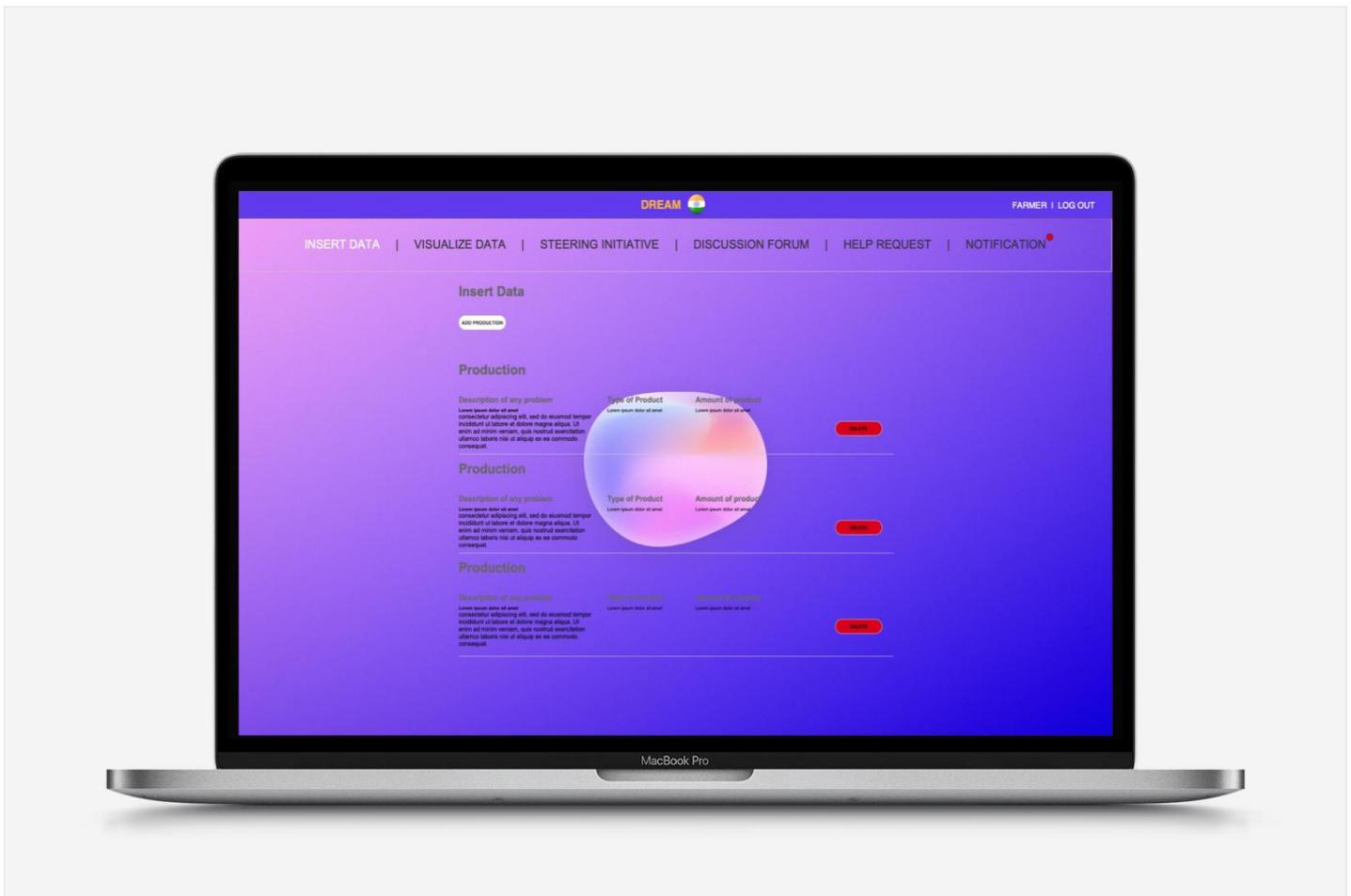


Figure 7

- **Farmer Insert Data Interface**

This Interface allows Farmers to access the services offered by DREAM application. Interacting with the menu, Farmer can access the different interfaces that will ensure, if he clicks on “Insert Data” button, to visualize data previously entered, to delete these or to add a production (*Figure 8*). If Farmer clicks on “Add Production” button he will be redirected to a Insert Data subsection (*Figure 9*) where by entering type of product, amount of product produced and description of any problem about the product he can insert new data. In this subsection there’s a button enables to add more data about more products.



*Figure 8*

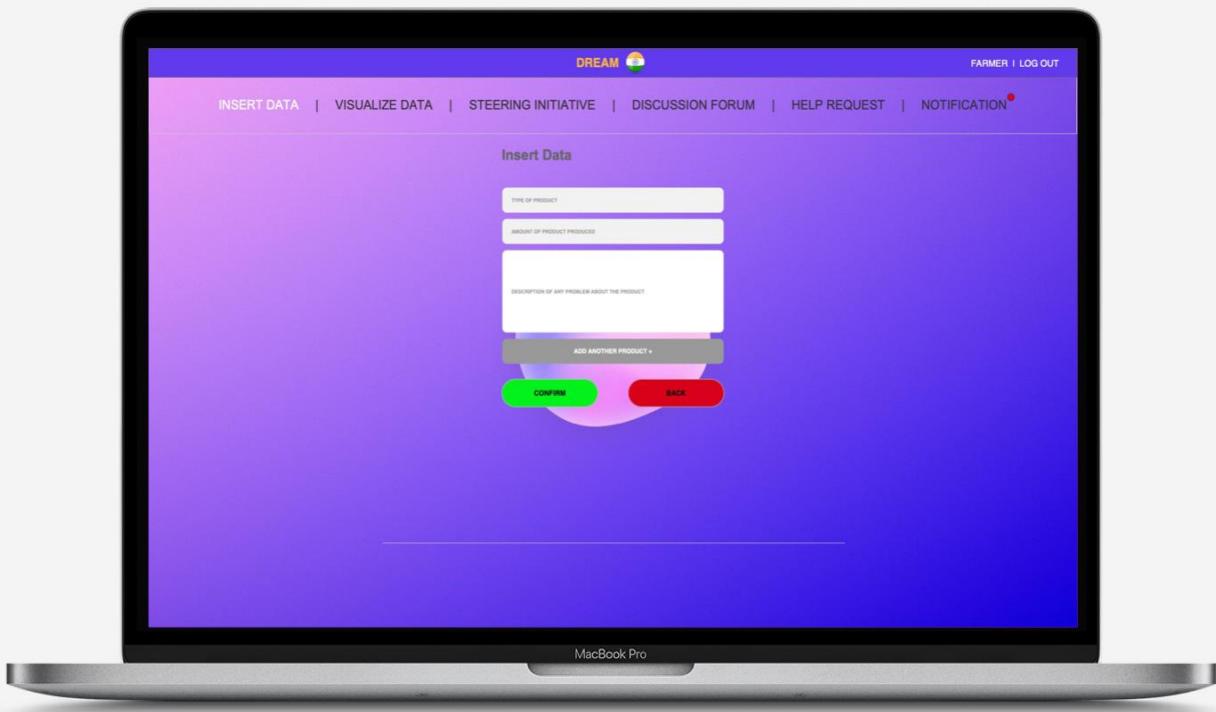


Figure 9

- **Farmer Visualize Data Interface**

This Interface allows Farmers to access the data offered by DREAM application. Interacting with Visualize Data Box, Farmer can visualize Weather Forecast data (*Figure 10*), Sensor Data (*Figure 11*), Water Irrigation data, Humidity Soil data and Fertilizer Suggestion.

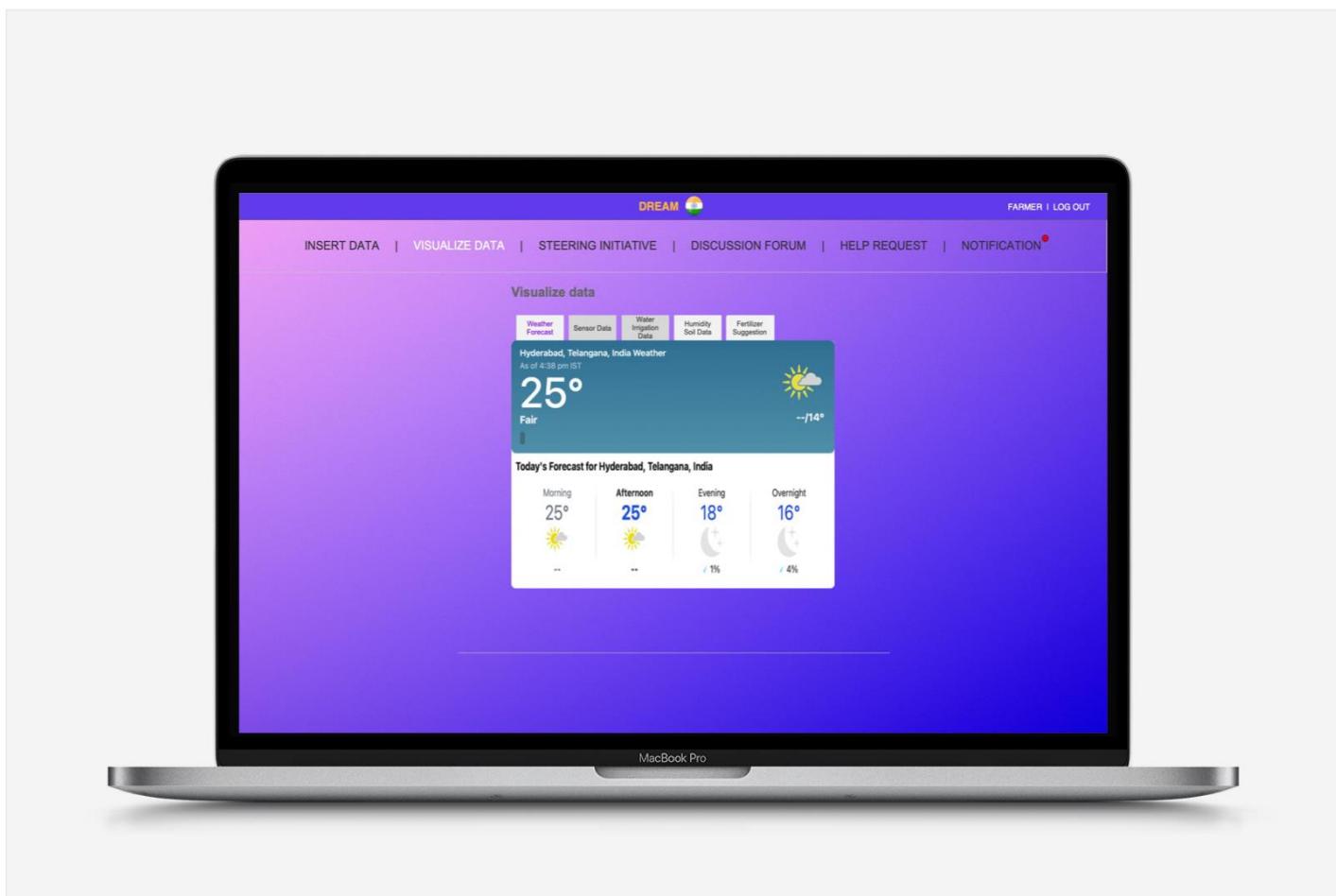


Figure 10

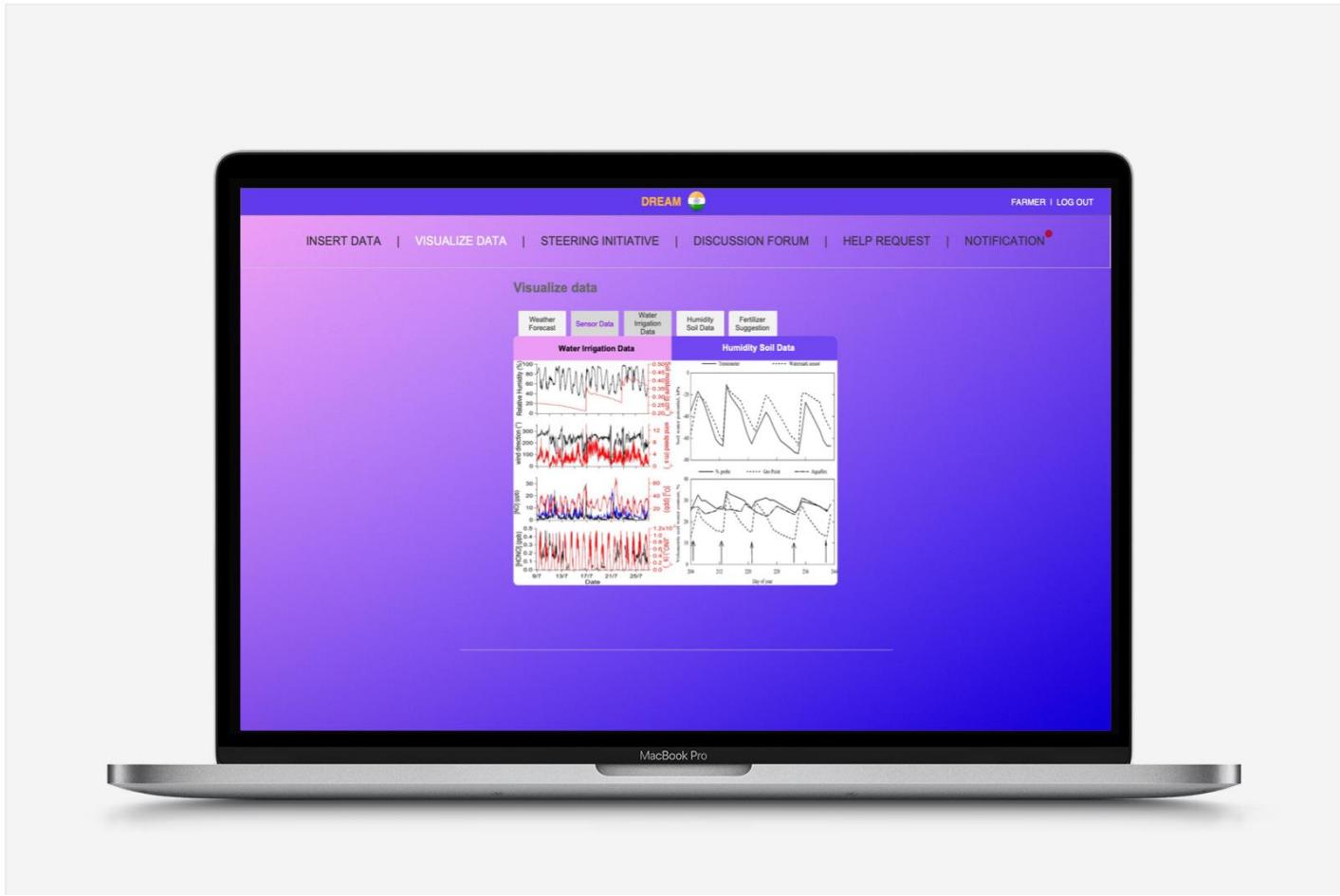
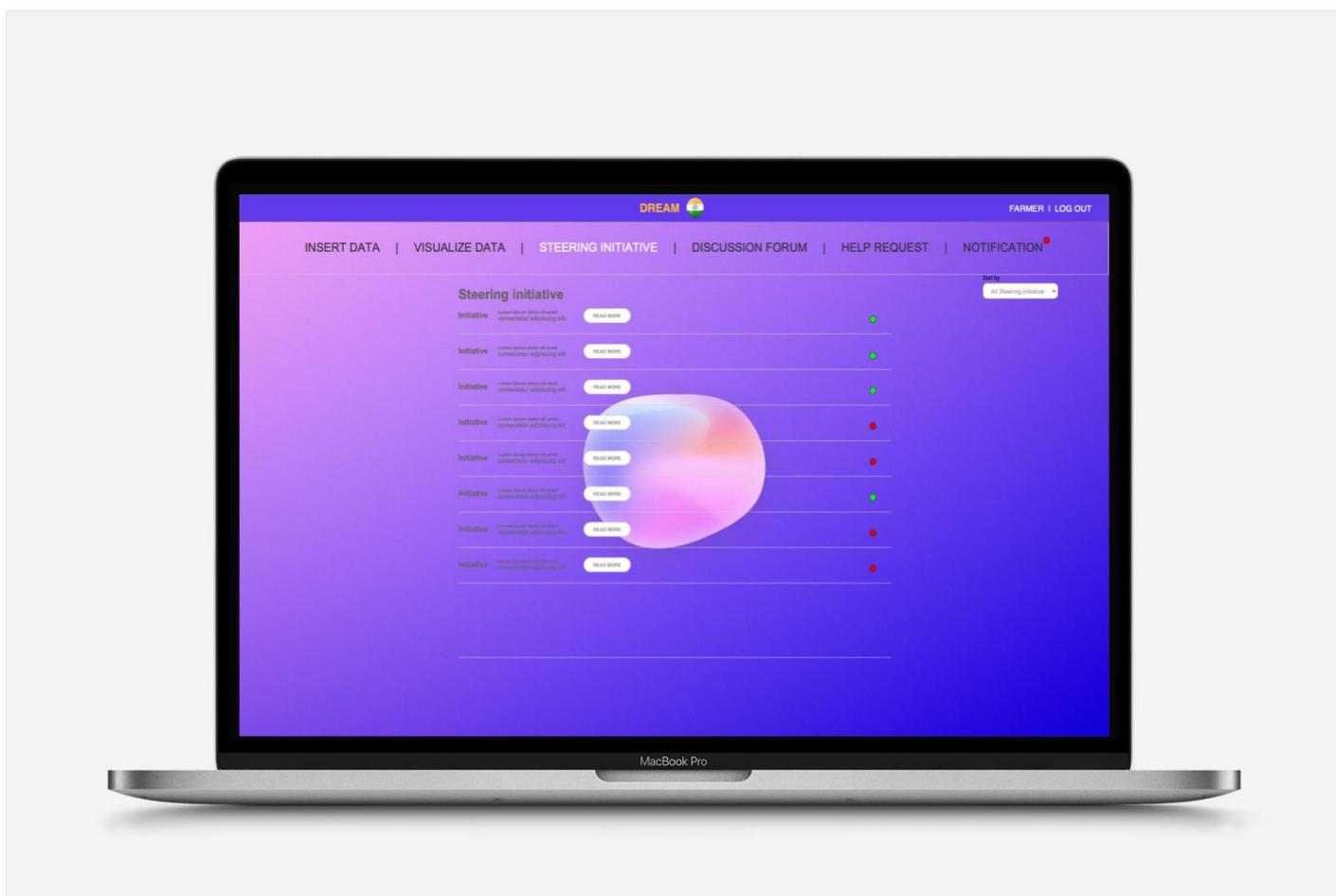


Figure 11

- **Farmer Steering Initiative Interface**

This Interface allows Farmers to access the steering initiative offered by DREAM Agronomist. Interacting with filter button “Sort by”, Farmer can visualize all steering initiative (active or inactive, *Figure 12*) and personal steering initiative. By clicking “Read More” button, if this steering initiative is active, Farmer can participate to this one by clicking on the appropriate field and entering the amount of product (*Figure 13*). After the established time is elapsed it will be possible to register final data about steering initiative by clicking on “Read More” button, in steering initiative interface (*Figure 14*).



*Figure 12*

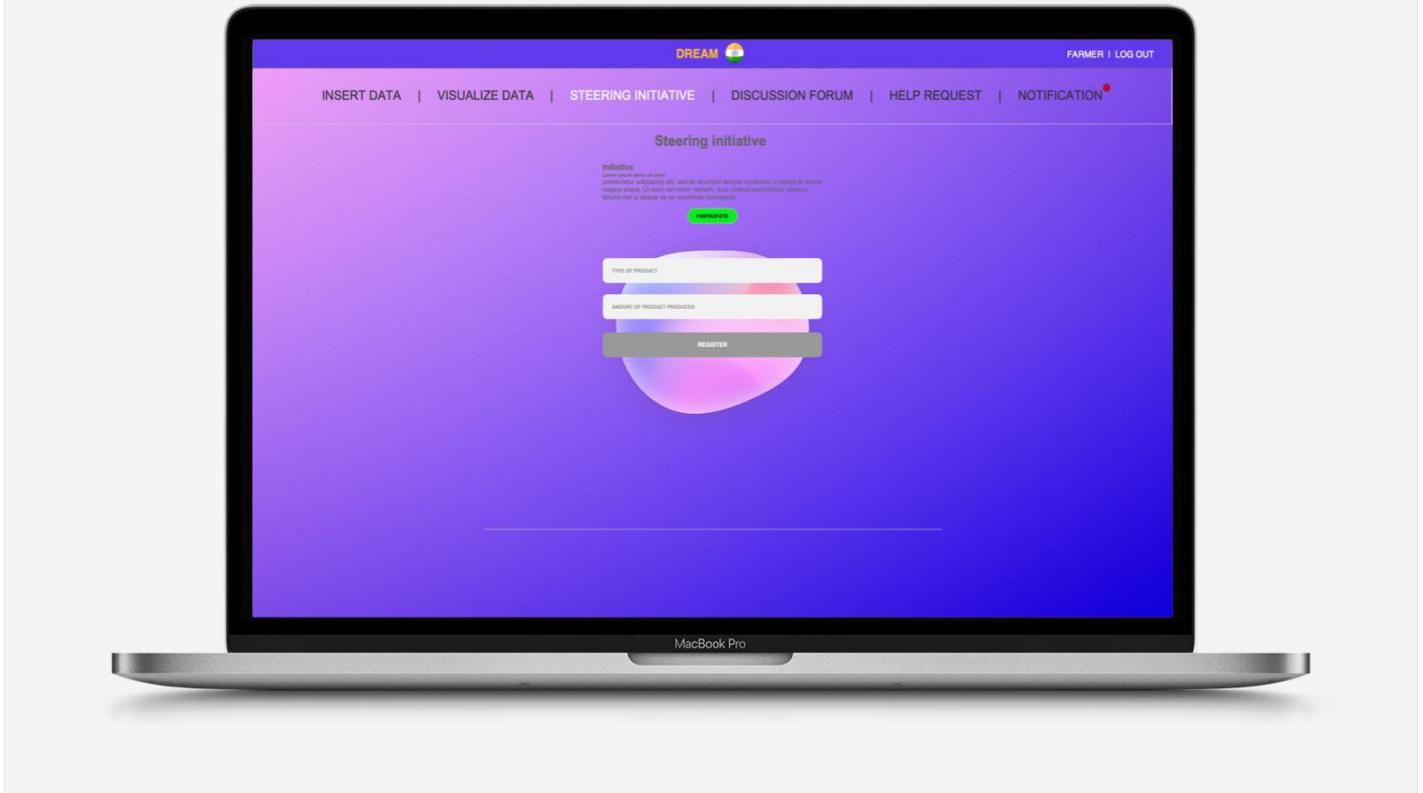


Figure 13

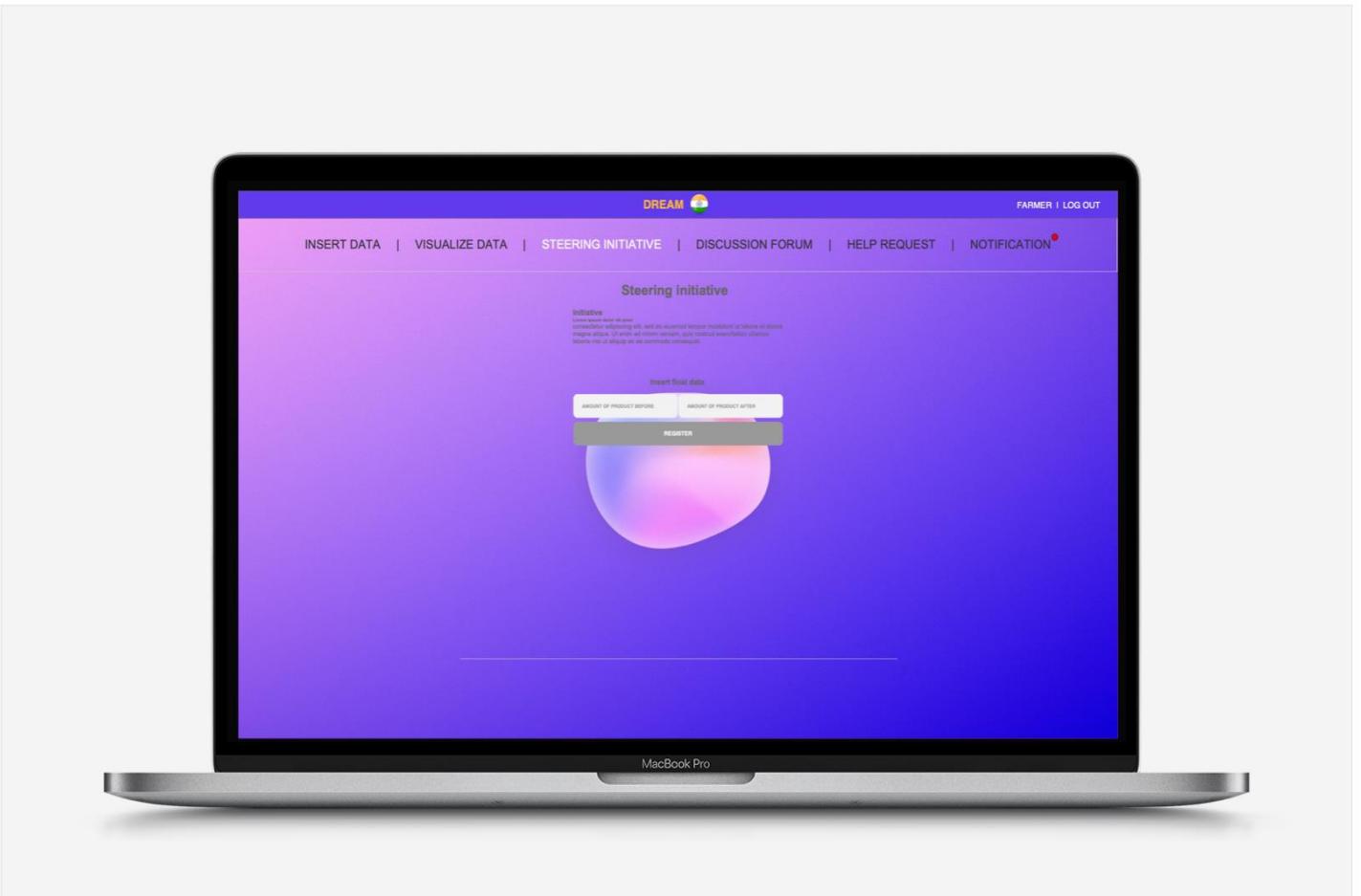
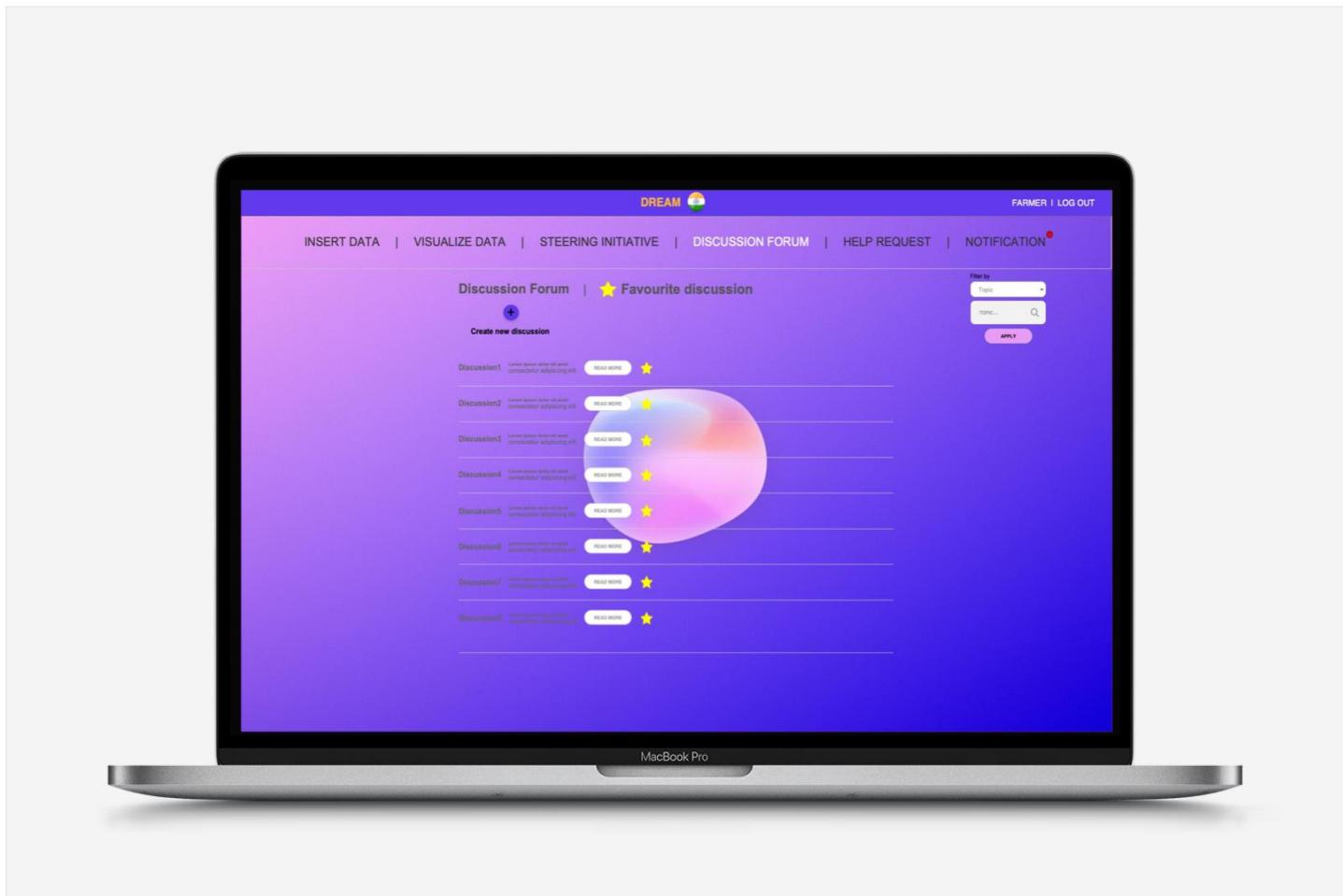


Figure 14

- **Farmer Discussion Forum Interface**

This Interface allows Farmers to access discussion forum section offered by DREAM Application. Interacting with interface, Farmer can visualize all discussion (*Figure 15*) or using filter option he can filter discussion by topic or by favorite discussion (farmer can add his favorite discussion using star button). By clicking on create a new discussion, farmer will be redirected to the create new discussion subsection where by inserting topic, title and description he can create a new discussion and publish it clicking on “Create” button (*Figure 16*). Farmer can answer a discussion clicking on “Read More” button and later texting his answer in the box below (*Figure 17*).



*Figure 15*

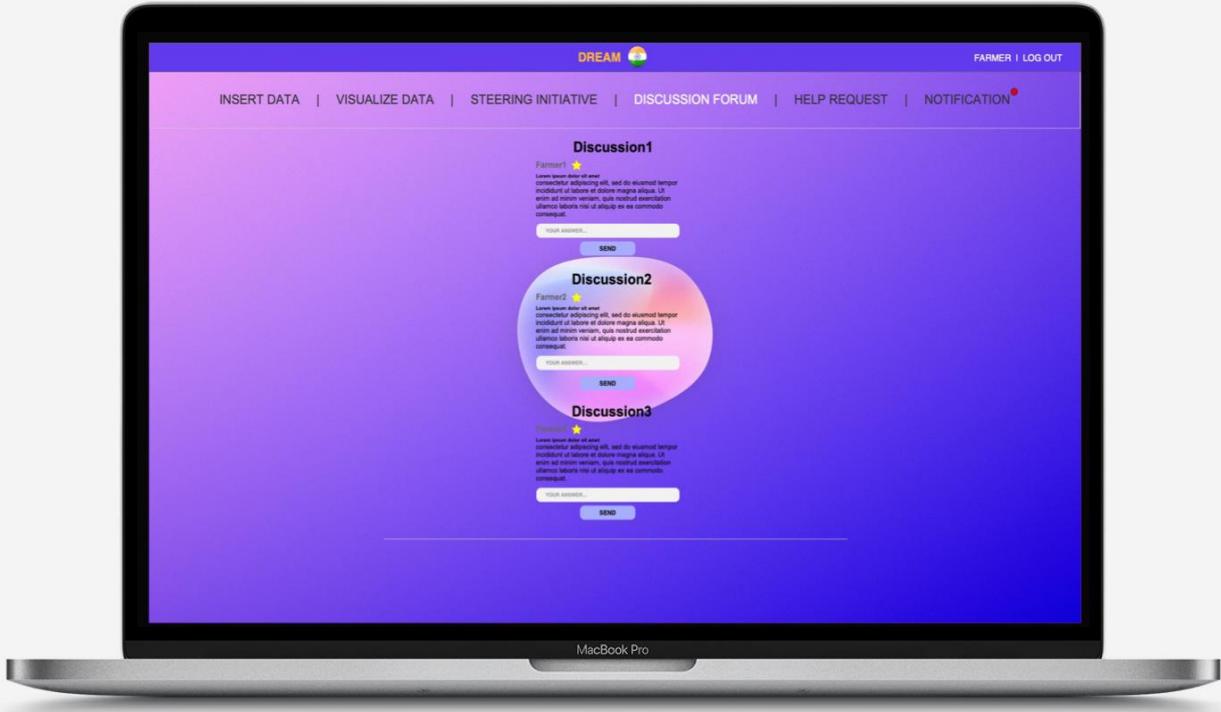


Figure 16

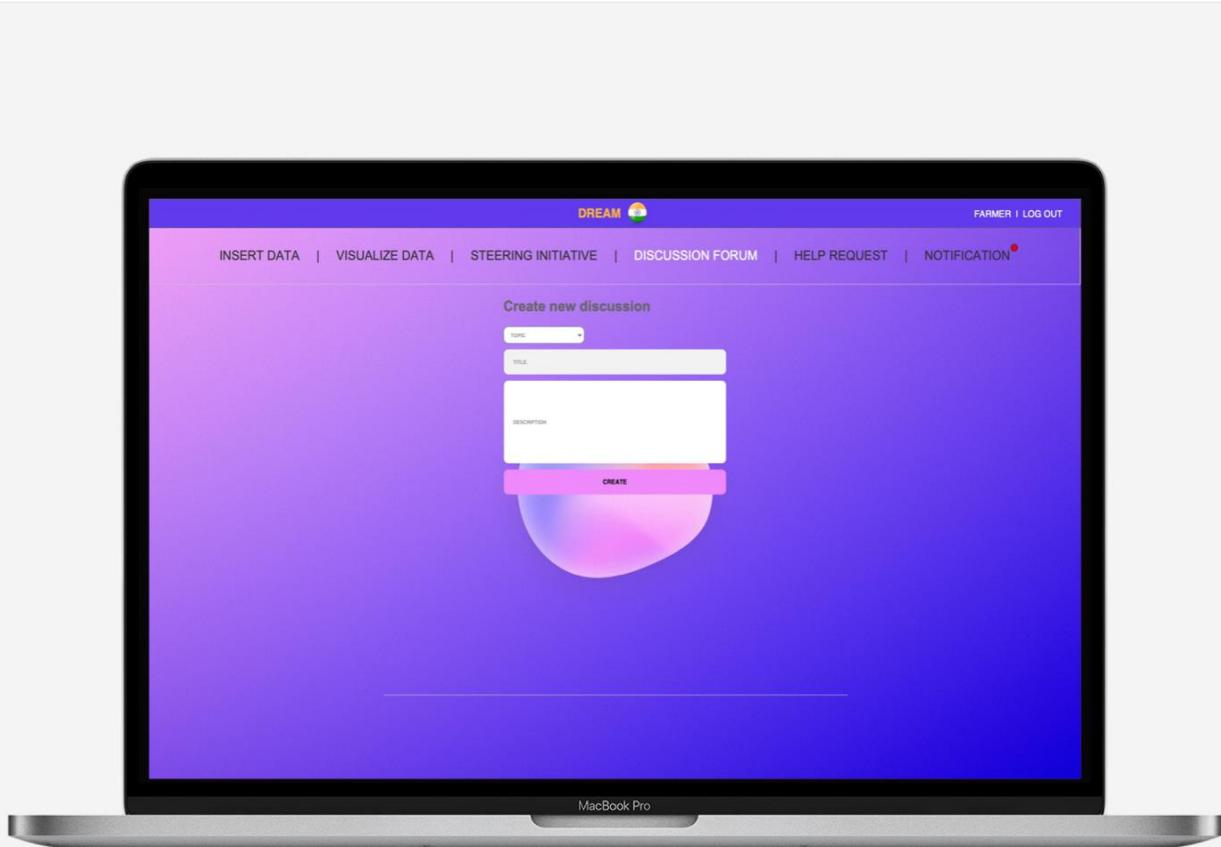
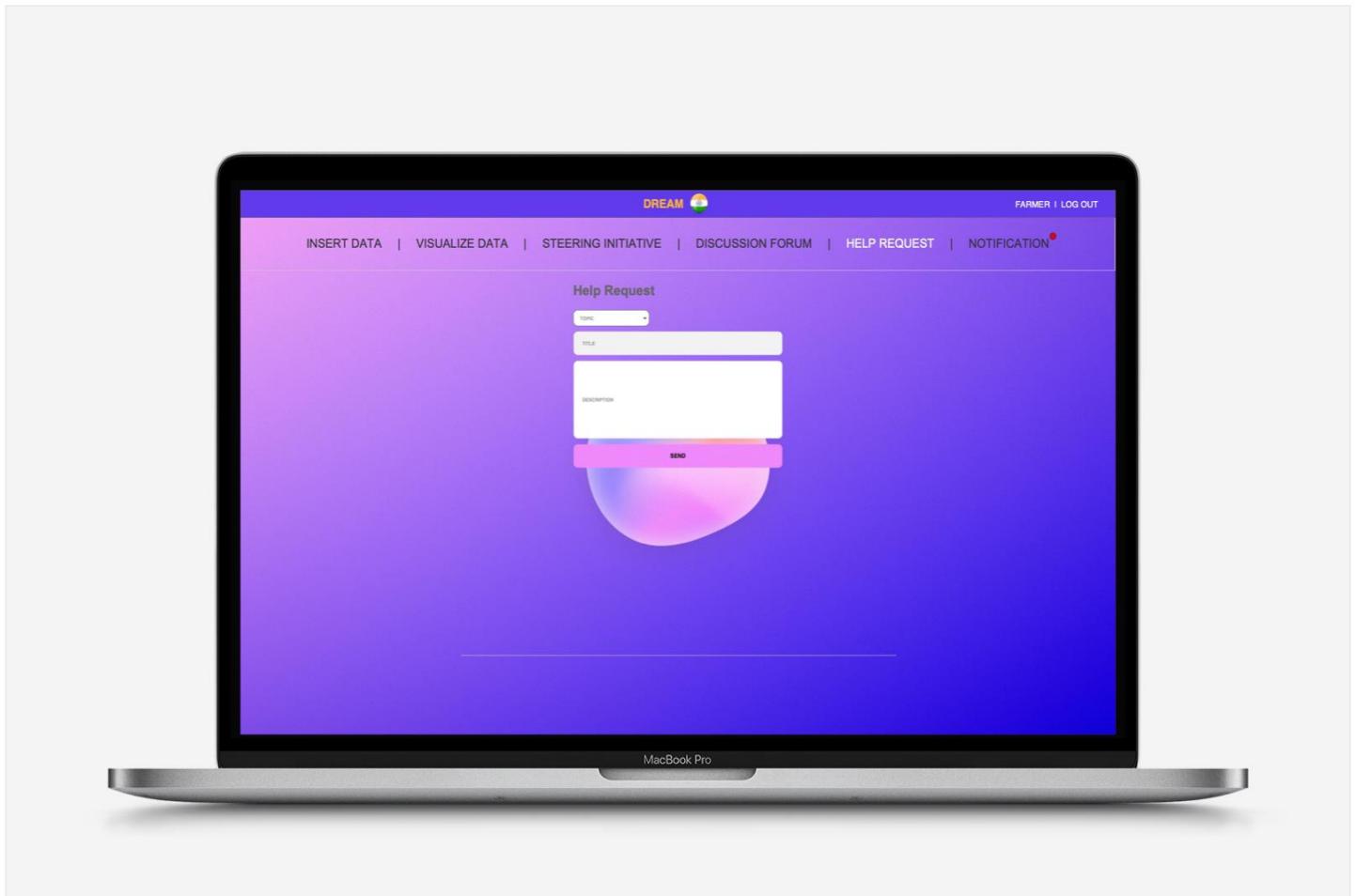


Figure 17

- **Farmer Help Request Interface**

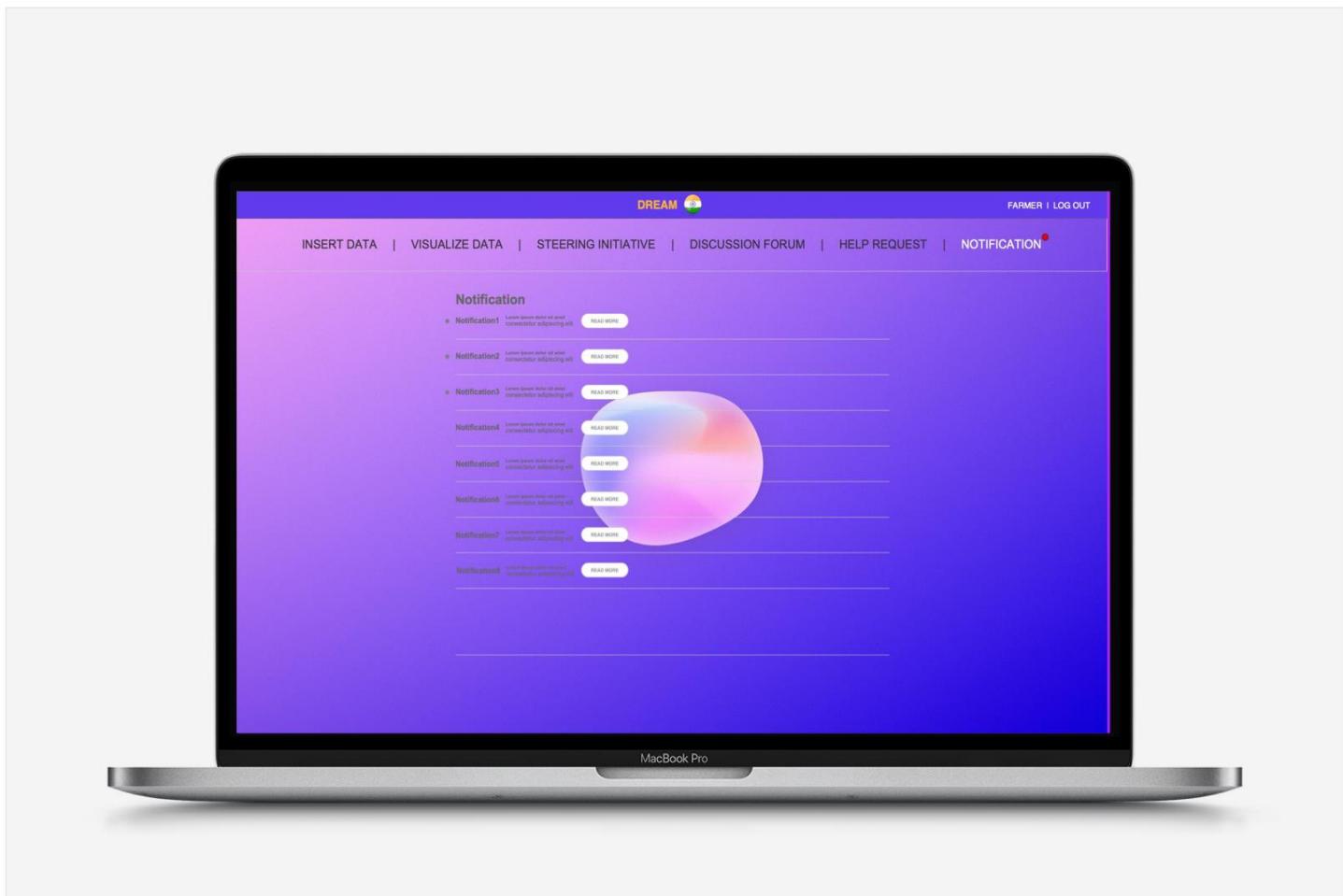
This Interface allows Farmers to access help request section offered by DREAM Application. Interacting with interface, Farmer can submit an help request to Agronomist. By inserting topic, title and problem description and clicking on “Send” button, Farmer sends Help Request (*Figure 18*).



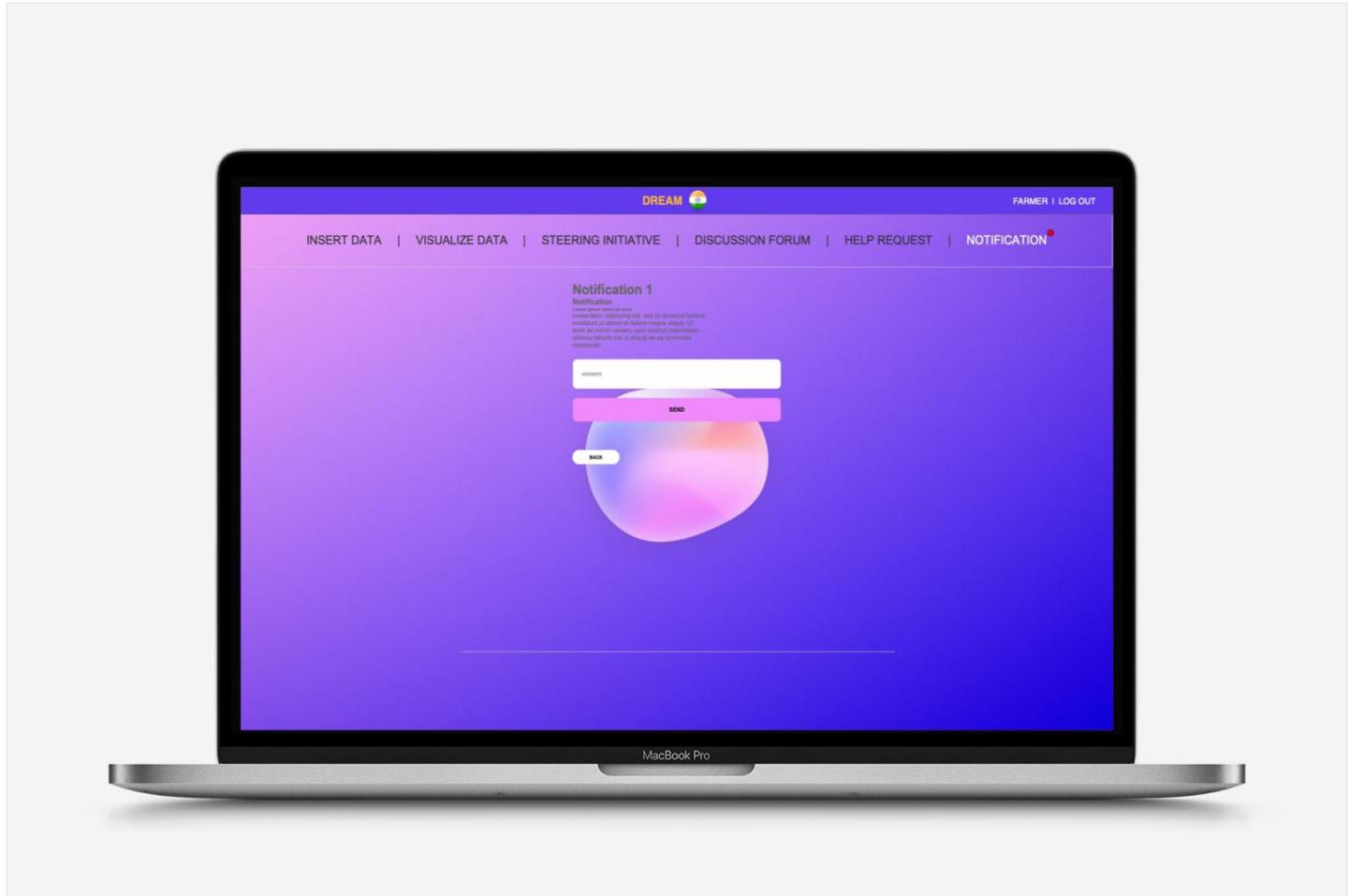
*Figure 18*

- **Farmer Notification Interface**

This Interface allows Farmers to receive notification in DREAM Application. Entering in this section, Farmer can visualize all received notification (*Figure 19*). By clicking on “Read More” button, Farmer can visualize notification description and if the notification was received by agronomist, it’s possible to send an answer by typing in the appropriate box and pushing “Send” button (*Figure 20*).



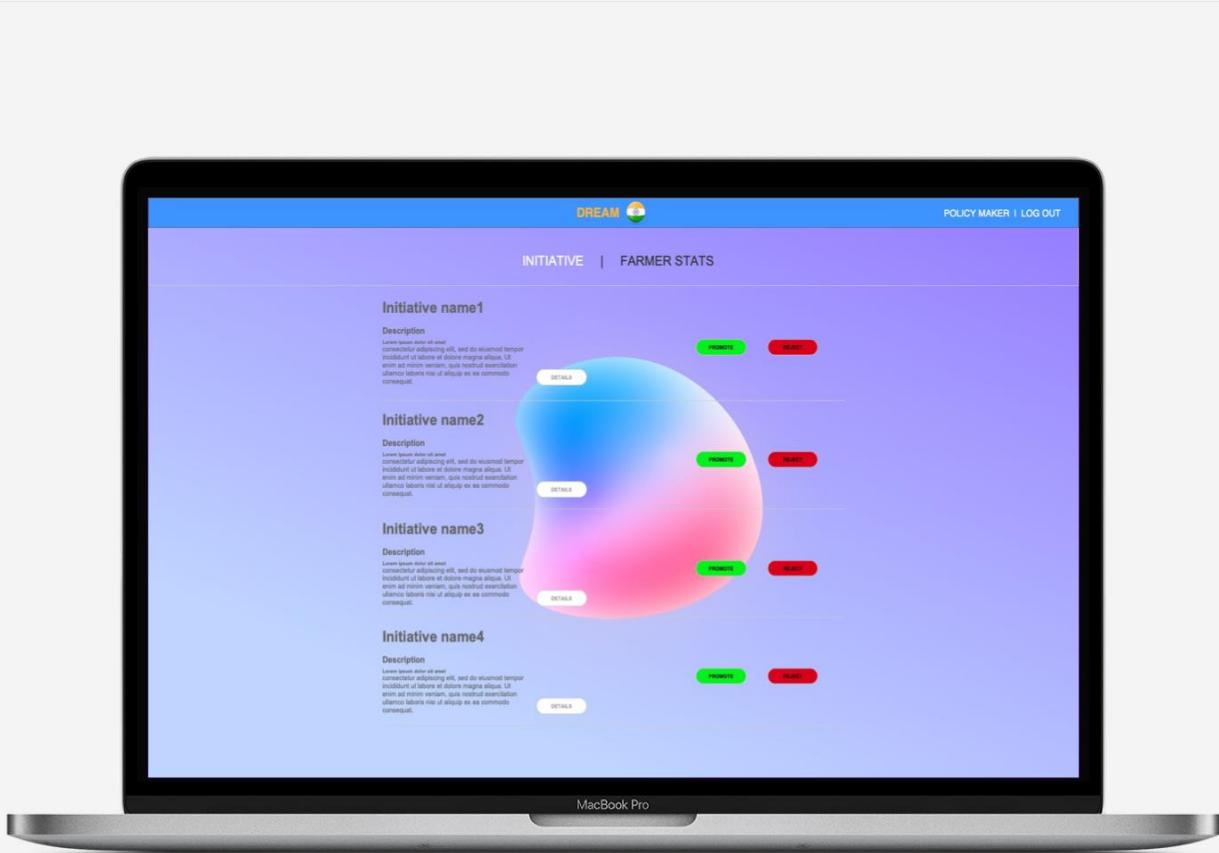
*Figure 19*



*Figure 20*

- **Policy Maker Initiative Interface**

This Interface allows Policy Maker to promote or reject steering initiative in DREAM Application. Entering in this section, Policy Maker can visualize all active steering initiative (*Figure 21*). By clicking on “Promote” or “Reject” button, Policy Maker can promote or reject a steering initiative. By clicking on “Details” button, Policy Maker can visualize data about this steering initiative, particularly those farmers who adopted this, type of product, amount of product produced and their performance index at the beginning and end of initiative (*Figure 22*).



*Figure 21*

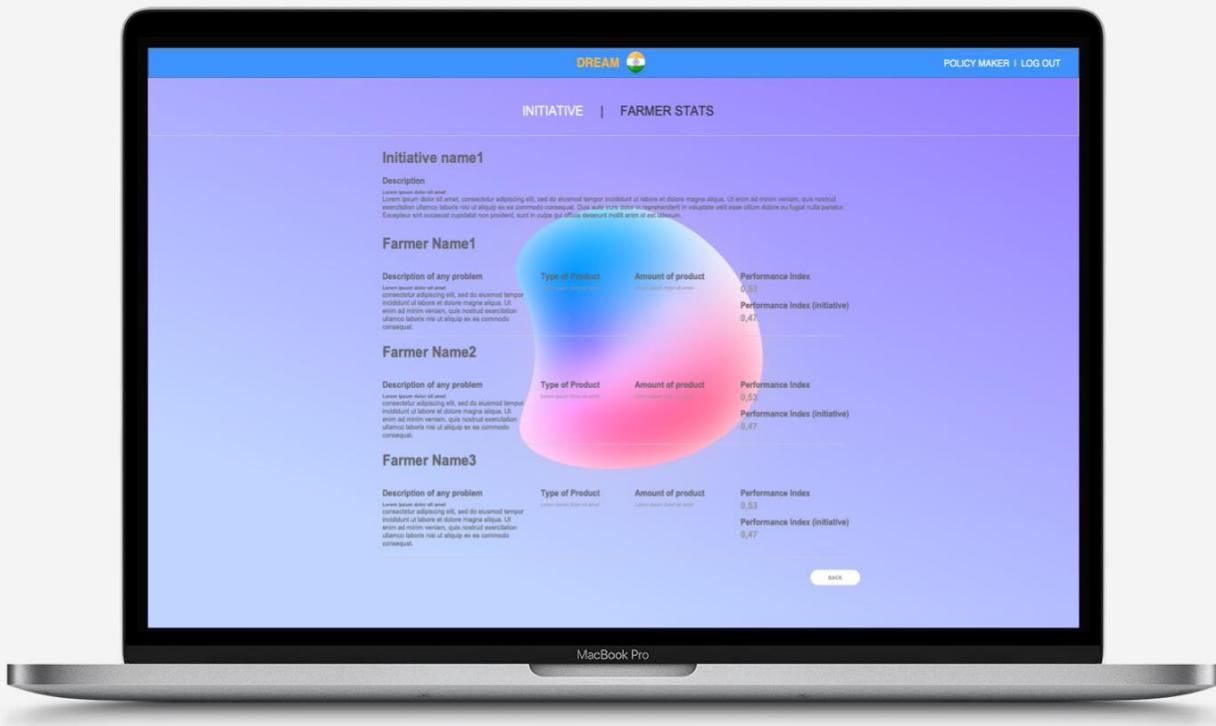
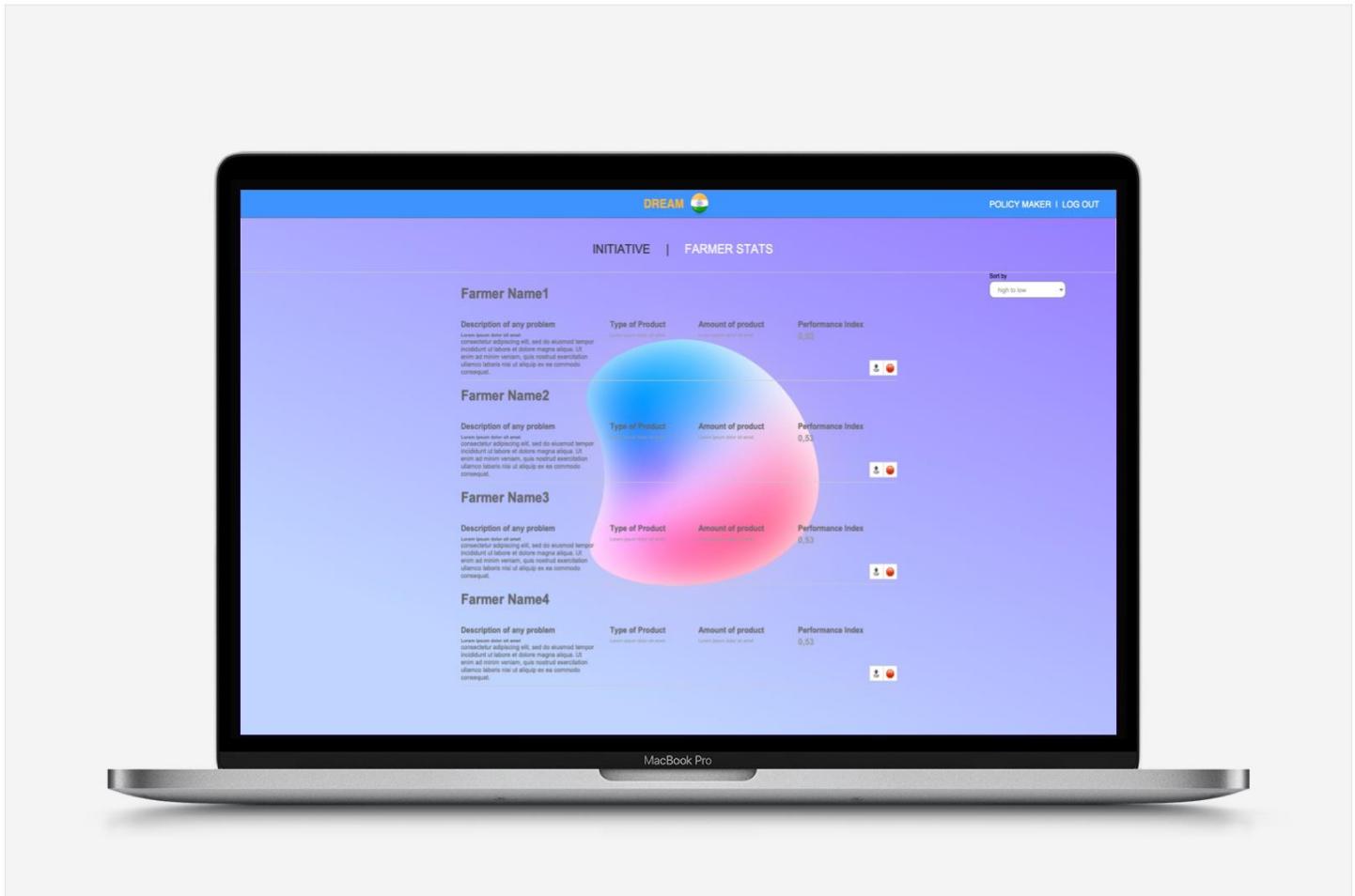


Figure 22

- **Policy Maker Farmer Stats Interface**

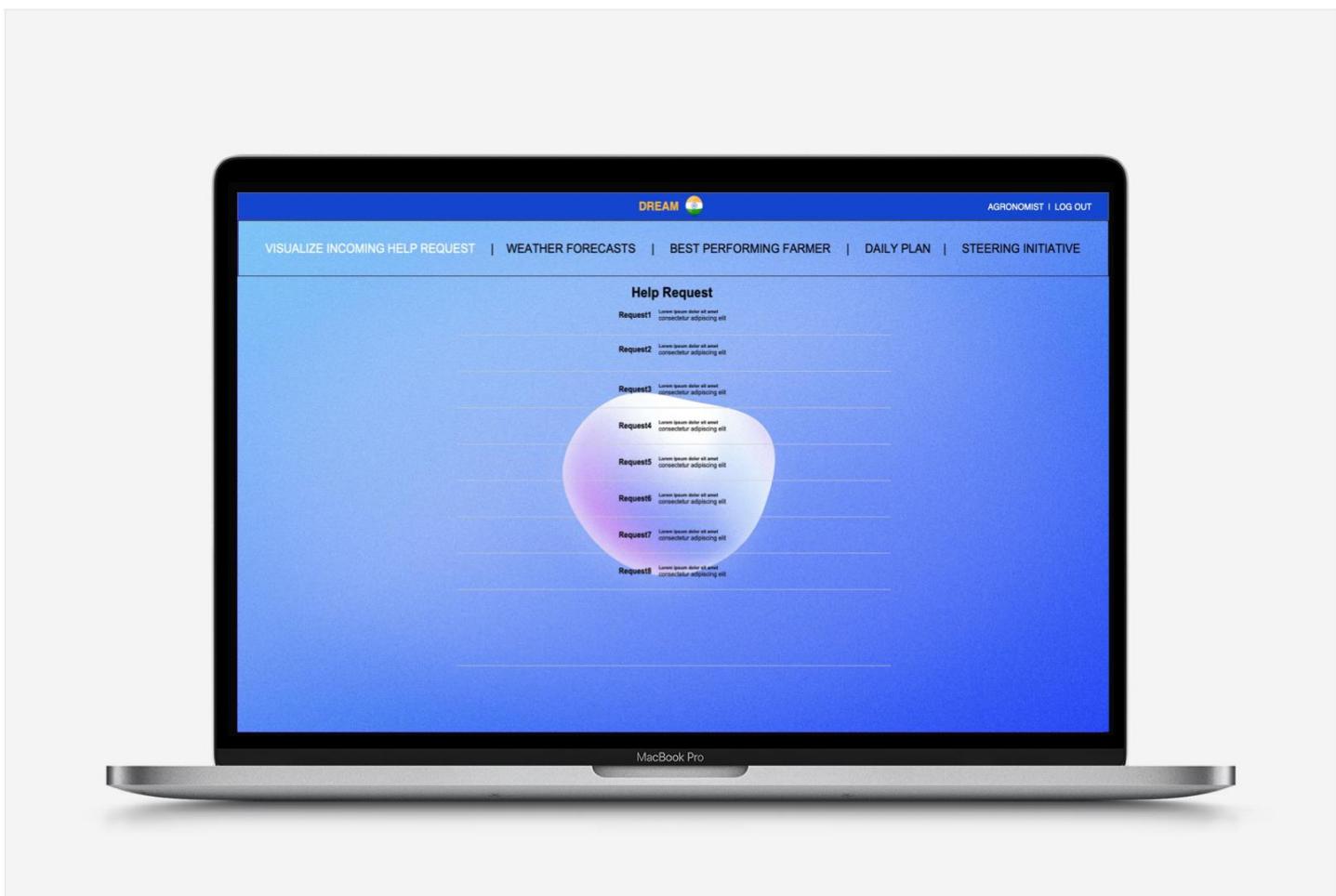
This Interface allows Policy Maker to evaluate Farmer stats in DREAM Application. Entering in this section, Policy Maker can visualize all farmer stats (*Figure 23*). By clicking on “Sort by” button, Policy Maker can order Farmer stats list by high-to-low or low-to-high performance index. By clicking on “top” button, Policy Maker can evaluate a farmer as the best performer. By clicking on “stop” button, Policy Maker can evaluate a farmer as the worst performer.



*Figure 23*

- **Agronomist Visualize incoming help request Interface**

This Interface allows Agronomist to visualize Farmers help request list in DREAM Application (*Figure 24*). By clicking on box request, Agronomist can provide help sending a message directly to relative Farmer (*Figure 25*). In addition, Agronomist can visualize all of the earlier discussion.



*Figure 24*

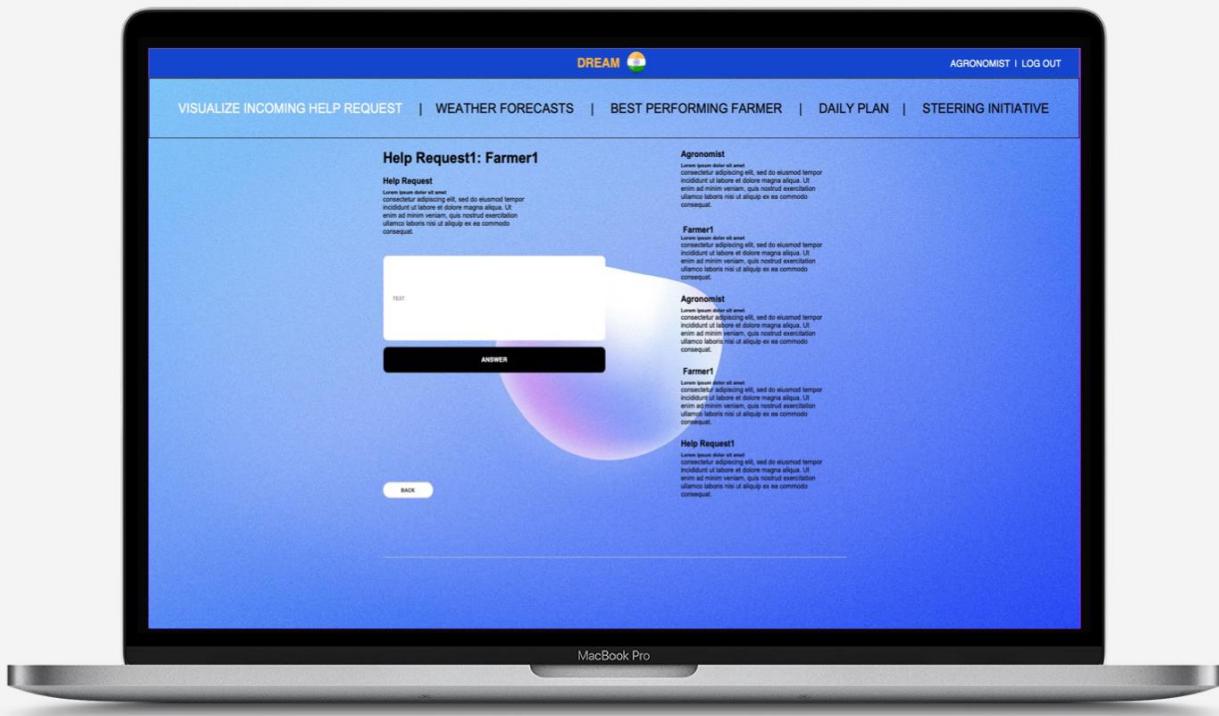
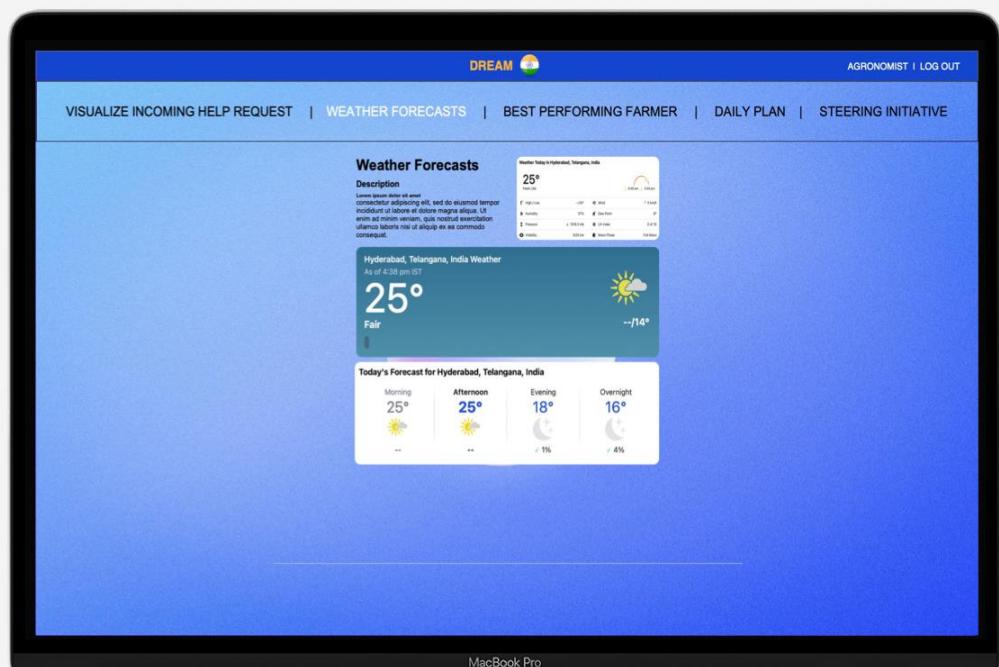


Figure 25

- **Agronomist Weather forecast Interface**

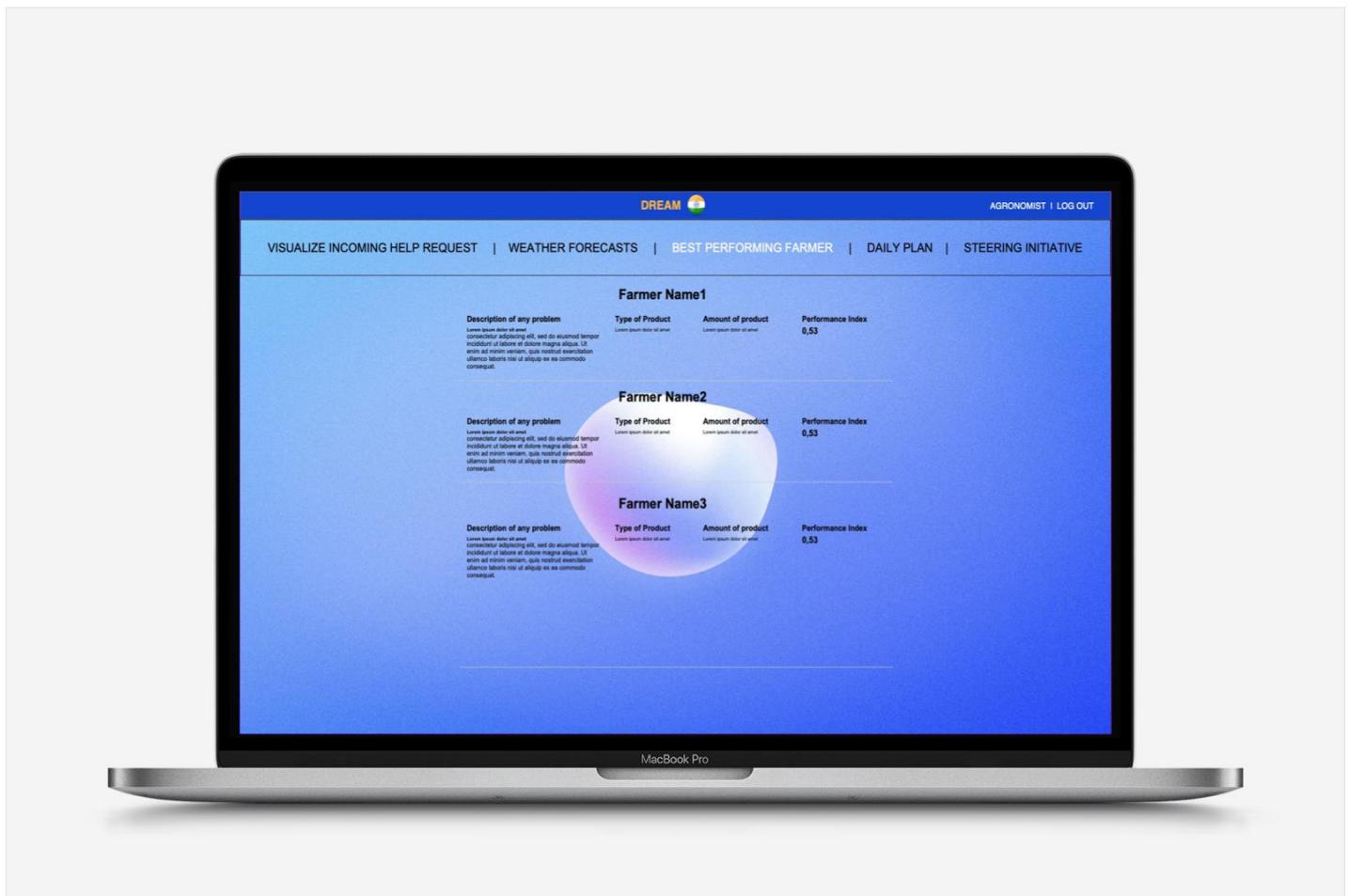
This Interface allows Agronomist to visualize Weather Forecasts Data offered by DREAM Application (*Figure 26*).



*Figure 26*

- **Agronomist Best performing Farmer Interface**

This Interface allows Agronomist to visualize best performing farmer in his professional area inserted in DREAM Application (*Figure 27*). Agronomist can visualize Farmer Name, Description of any problem, Type of product, amount of product produced and performance index.

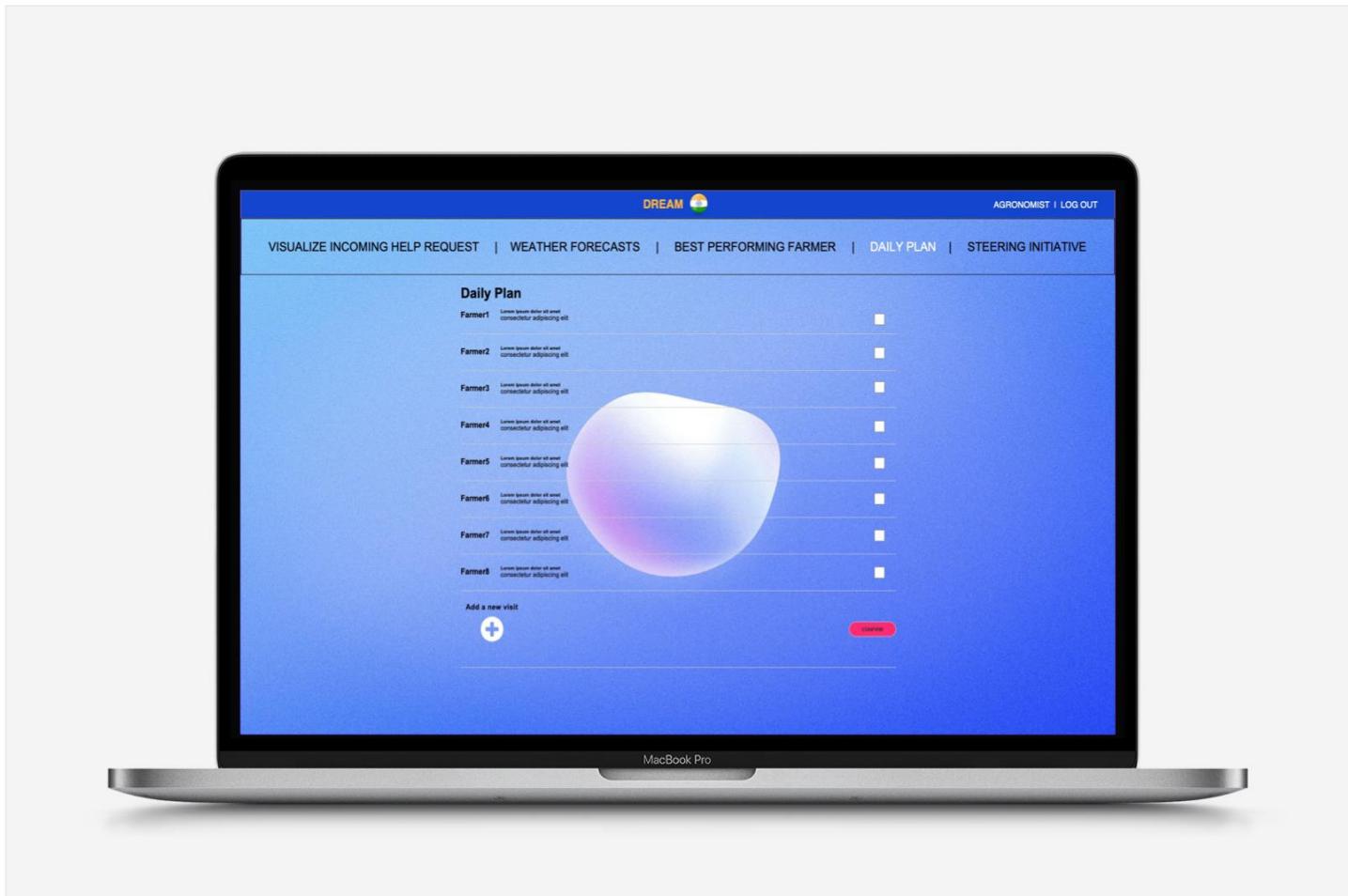


*Figure 27*

- **Agronomist Daily Plan Interface**

This Interface allows Agronomist to visualize his visits daily plan offered daily by DREAM Application (*Figure 28*). Interacting with interface, Agronomist can check the box, at the end of the workday, if he has completed it. Later Agronomist can confirm daily farmer visits by clicking on “Confirm” button.

This Interface allows Agronomist to add a new visit. By clicking on “Add a new Visit” button, Agronomist will be redirected to the add a new visit page where by selecting farmer and date, he can add a new visit, by clicking on the appropriate button (*Figure 29*).



*Figure 28*

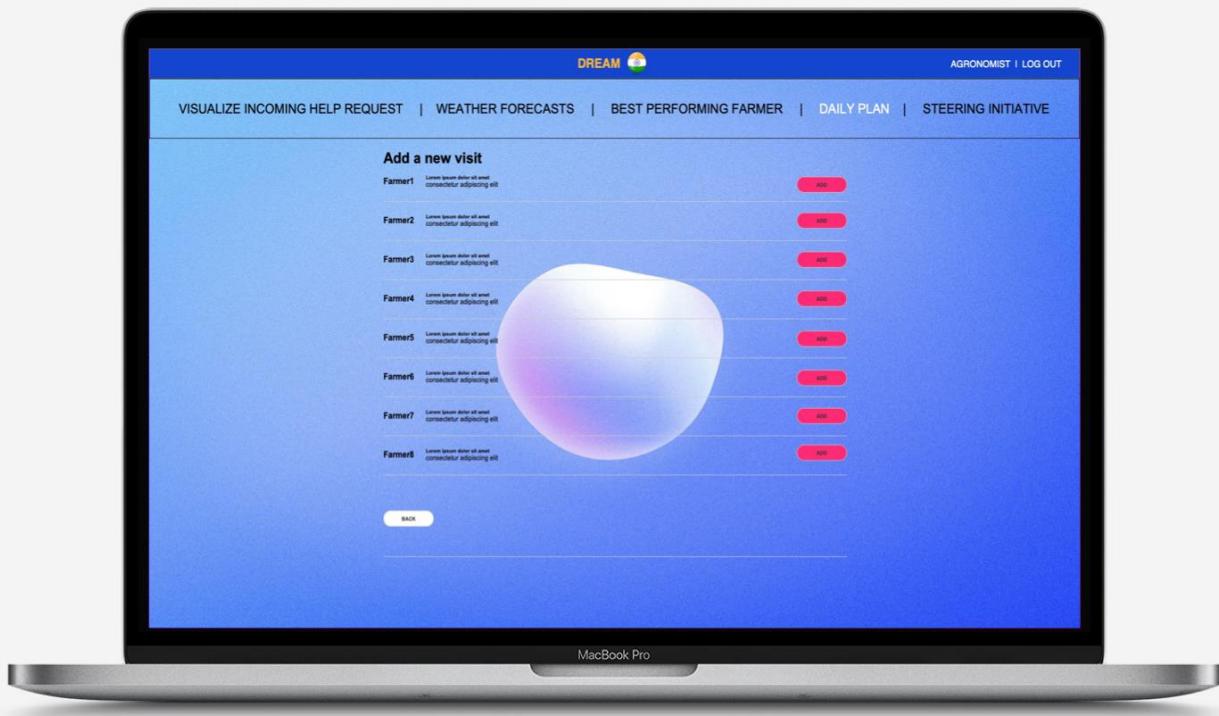
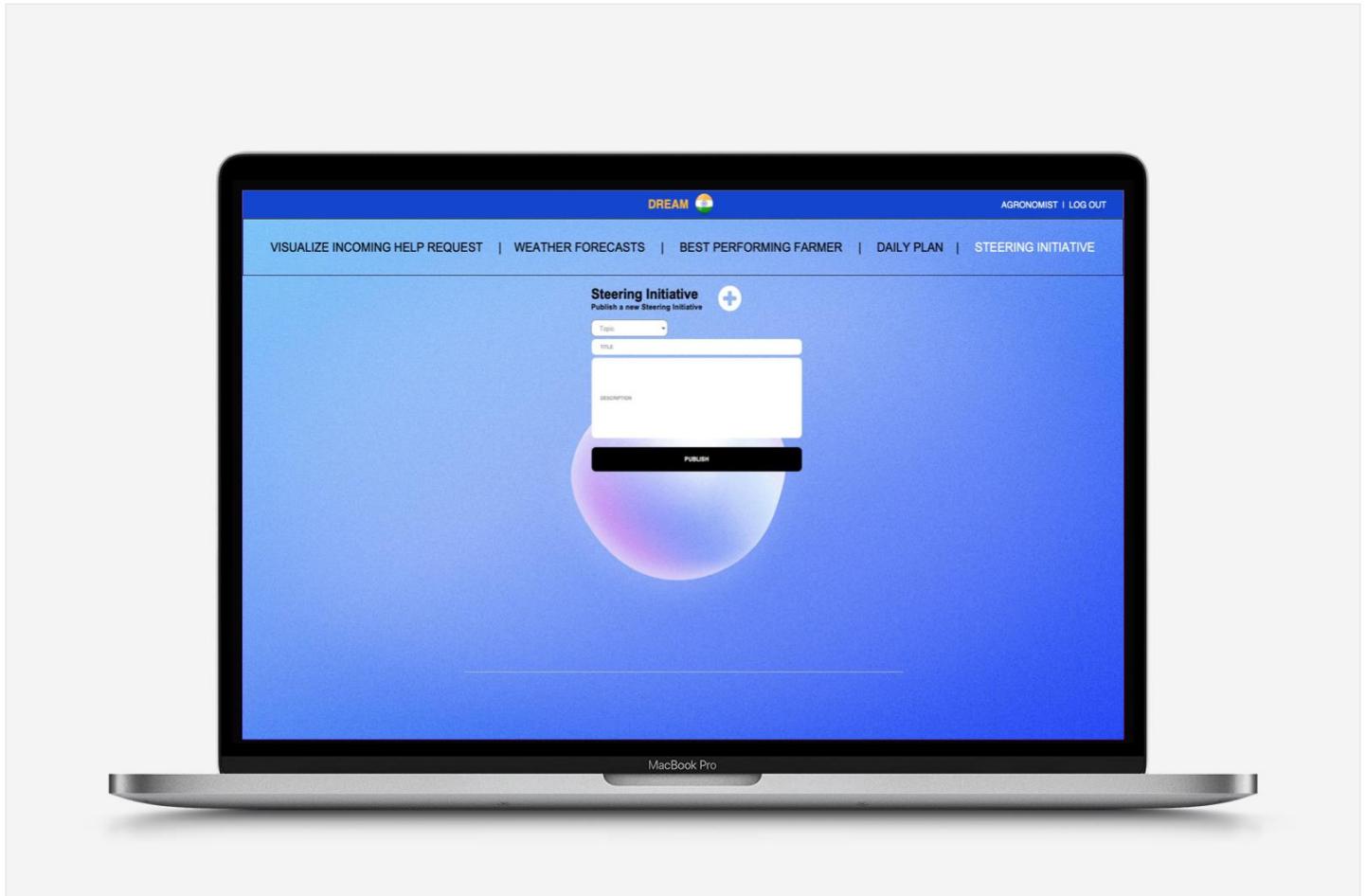


Figure 29

- **Agronomist Steering Initiative Interface**

This Interface allows Agronomist to create new Steering Initiative in DREAM Application (*Figure 30*). Interacting with interface, Agronomist can select topic, insert title and description. So, Agronomist can publish it clicking on the “Publish” button.



*Figure 30*

## 1.2 Hardware Interfaces

To allow the collection of specific types of data, the DREAM software must integrate an automatic way of communication with all the hardware pieces located in farms.

To meet this requirement, each **Farmer** should dispose of:

- a *set of sensors*, to be located in the soil;
- A *water irrigation system*, able to collect data concerning the amount of water used by each **Farmer**;
- an *internet capable device*, to interact with the DREAM application

Furthermore, both **Policy Makers** and **Agronomist** must have an *internet capable device*, to join the application, analyse the data coming from farmers and manage interventions and interactions.

## 1.3 Software Interfaces

The DREAM application relies on several external services.

Among them, we can mention:

- **Weather forecast:** the software collects data concerning weather forecast from other platforms;
- **Maps location service:** DREAM application relies on specific map API service to locate region all around the country

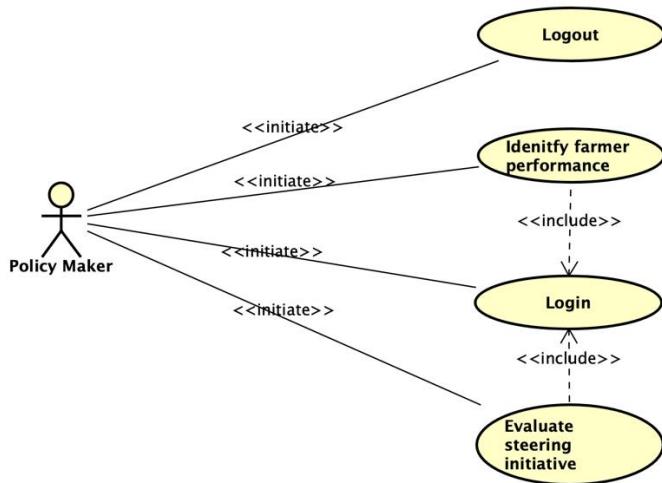
## 1.3 Communication Interfaces

All the device connected to the DREAM software must dispose of an internet connection to work properly.

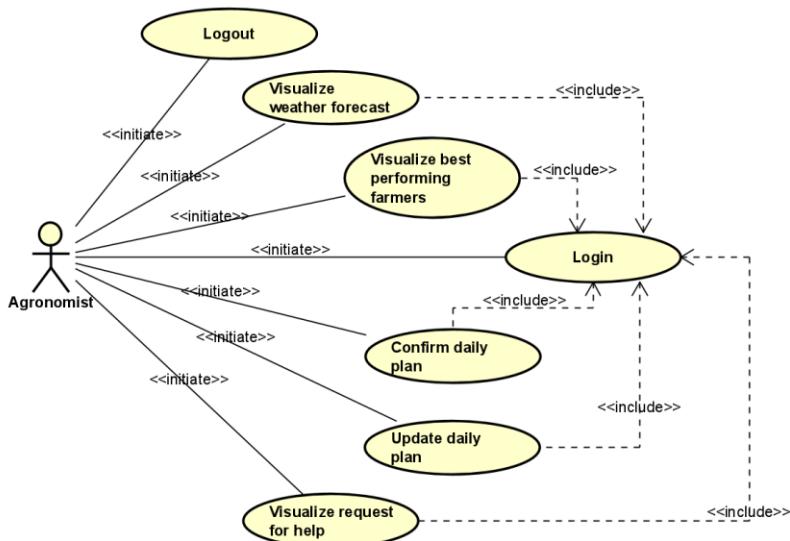
## 2 Functional Requirements

### 2.1 Use Case Diagrams

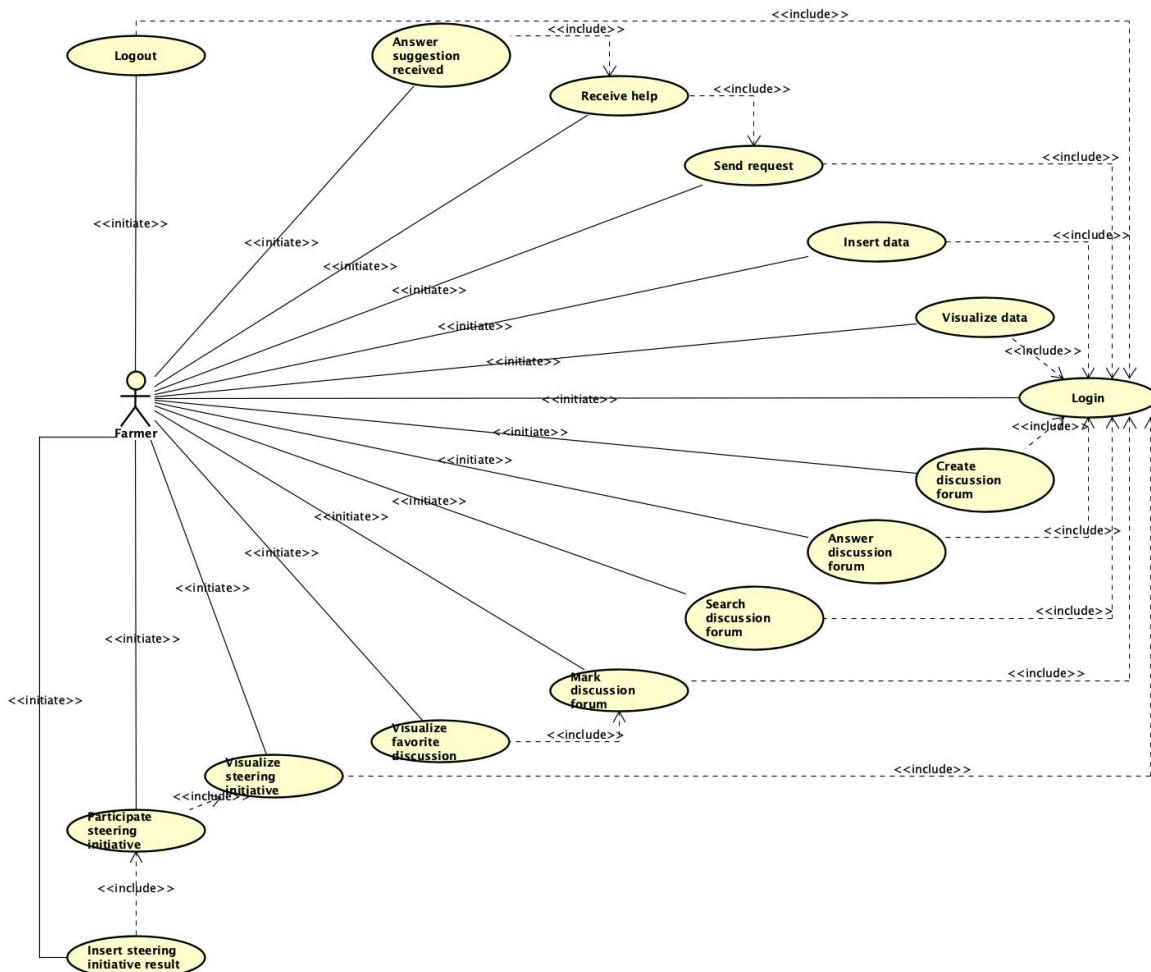
- Policy Maker



- Agronomist



- Farmer



## 2.2 Use Case Analysis

Name	[U.C.1] User Registration
Actors	Unregistered User
Entry Conditions	<ul style="list-style-type: none"> <li>• Unregistered User wants to register to DREAM</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. Unregistered User visualizes the application login forum and clicks the link to register</li> <li>2. Unregistered User enters his/her Name</li> <li>3. Unregistered User enters his/her Surname</li> <li>4. Unregistered User enters his/her Date of Birth</li> <li>5. Unregistered User enters his/her Place of Birth</li> <li>6. Unregistered User enters his/her Email</li> <li>7. Unregistered User enters a Password</li> <li>8. Unregistered User chooses his/her Role <ul style="list-style-type: none"> <li>8.1. If “Agronomist” is chosen, then Unregistered User choose an Region he/she will be responsible of</li> <li>8.2. If “Farmer” is chosen, then Unregistered User enters the location of the farm</li> </ul> </li> <li>9. Unregistered User checks the “Terms &amp; Conditions” document and agree to it clicking on the checkbox</li> <li>10. Unregistered User checks the “Privacy Policy” document and agree to it clicking on the checkbox</li> <li>11. Unregistered User clicks the “Register” button</li> <li>12. The System process and saves the information</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• Unregistered User registered correctly</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>• Not all the mandatory fields have been correctly filled</li> <li>• The inserted password does not respect the security standards imposed by the application</li> <li>• For the “Farmer” case, the location entered is not located in India</li> <li>• “Terms &amp; Conditions” checkbox has not been checked</li> <li>• “Privacy Policy” checkbox has not been checked</li> </ul> <p>An error message is shown</p>

Name	[U.C.2] Login
Actors	Registered User
Entry Conditions	<ul style="list-style-type: none"> <li>User is correctly registered to the platform</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>User opens the login page</li> <li>User enters his email in the appropriate field</li> <li>User enters his password in the appropriate field</li> <li>User clicks the Login button</li> <li>The System checks the credential</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>User successfully logged in</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>Inserted email does not exist in the database</li> <li>Inserted password is wrong</li> </ul> <p>An error message is shown</p>

Name	[U.C.3] Logout
Actors	Registered user
Entry Conditions	<ul style="list-style-type: none"> <li>User is logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>User clicks on the Logout button</li> <li>The system logs user out</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>User successfully logged out</li> </ul>
Exceptions	None

Name	[U.C.4] Policy Maker identifies farmer performance
Actors	Policy Maker
Entry Conditions	<ul style="list-style-type: none"> <li>Policy Maker is logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Policy Maker clicks on the Farmer stats button</li> <li>Policy Maker visualizes Farmer stats list             <ol style="list-style-type: none"> <li>If Policy Maker selects high-to-low option                     <ol style="list-style-type: none"> <li>Policy Maker visualizes Farmer stats page (ordered by best performance)</li> <li>Policy Maker clicks on the button to report that farmer as a best farmer</li> </ol> </li> <li>If Policy Maker selects low-to-high option                     <ol style="list-style-type: none"> <li>Policy Maker visualizes Farmer stats page (ordered by worst performance)</li> <li>Policy Maker clicks on the button to report that farmer as a worst farmer or</li> </ol> </li> </ol> </li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>Policy Maker successfully identified well performing farmers or underperforming farmers</li> </ul>
Exceptions	None

Name	[U.C.5] Policy Maker evaluate steering initiative
Actors	Policy Maker
Entry Conditions	<ul style="list-style-type: none"> <li>Policy Maker is logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Policy Maker clicks on the steering initiative button</li> <li>Policy Maker selects the initiative to analyse</li> <li>Policy Maker visualizes Farmer statistics list</li> <li>Policy Maker selects promote initiative or reject button             <ol style="list-style-type: none"> <li>If Policy Maker selects promote, the System notifies relative farmers to adopt it</li> <li>If Policy Maker selects delete</li> </ol> </li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>Policy Maker successfully checked the initiative efficiency</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>Initiative list is empty In this case an error message is shown</li> </ul>

Name	[U.C.6] Farmer inserts data
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>Farmer is correctly logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Farmer clicks on the “Insert data” section</li> <li>Farmer visualizes the page with the list of all product inserted by that moment and clicks the “Add Production” button and visualize the form</li> <li>Farmer enters the Type of product</li> <li>Farmer enters the Amount of product produced</li> <li>Farmer enters the Description of any problem his/she is facing about the product</li> <li>Farmers can decide to confirm the insertion or add another product with the respective buttons</li> <li>Farmer clicks on “Confirm” button, the System will process and save the information</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>Data about the product correctly stored in the System</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>Not all the mandatory fields ("Type of product" and "Amount of product produced") have been correctly filled</li> <li>"Amount of product produced" is not a number</li> </ul> <p>An error message is shown</p>

Name	[U.C.7] Farmer visualizes data
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>Farmer is correctly logged in</li> <li>Data information have been correctly stored</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Farmer clicks on the "Visualize data" section</li> <li>Farmer visualizes the page in which several options are given             <ol style="list-style-type: none"> <li>If farmer clicks on "Weather Forecast", he will visualize a map concerning his region with the relative weather forecast of the current day and the week</li> <li>If farmer clicks on "Sensor Data", he will visualize a submenu with two further options to choose between                     <ol style="list-style-type: none"> <li>If farmer clicks on "Water Irrigation Data", he will visualize data coming from the water irrigation system, such as the amount of water used every day, week, and month</li> <li>If farmer clicks on "Humidity Soil Data", he will visualize data coming from sensors deployed on the territory</li> </ol> </li> <li>If farmer clicks on "Crop Suggestion", he will visualize an input text field                     <ol style="list-style-type: none"> <li>Farmer enters the type of crop he wants suggestion about</li> <li>Farmer visualizes a list of suggestions about that crop</li> </ol> </li> <li>If farmer clicks on "Fertilizer Suggestion", he will visualize an input text field                     <ol style="list-style-type: none"> <li>Farmer enters the type of crop to fertilize he wants suggestion about</li> <li>Farmer visualizes a list of suggestions about fertilizers to use</li> </ol> </li> </ol> </li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>Farmer has successfully visualized data he was looking for</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>Farmer has not found any suggestion regarding the crop to plant</li> <li>Farmer has not found any suggestion regarding the fertilizer to use</li> </ul> <p>The System will suggest the farmer to open a discussion on the forum</p>

Name	[U.C.8] Farmer requests for help
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>Farmer is correctly logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Farmer clicks on the "Help request" section</li> <li>Farmer visualize the page and the form to request help</li> <li>Farmer chooses the Topic from those available</li> <li>Farmer enters the Title</li> <li>Farmer enters the Description</li> <li>Farmer clicks on the "Send" button</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>The help request is successfully sent to the agronomist responsible for that region</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>"Topic" has not been chosen</li> <li>"Title" has not been entered</li> </ul> <p>an error message is shown</p>

Name	[U.C.9] Farmer creates discussion forum
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>Farmer is correctly logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Farmer clicks on the "Discussion forum" section</li> <li>Farmer visualizes the page where all the discussions are presented, and clicks on the "Create new discussion" button</li> <li>Farmer visualizes the form to create a new discussion</li> <li>Farmer chooses the Topic from those available</li> <li>Farmer enters the Title</li> <li>Farmer enters the Description</li> <li>Farmer clicks the "Create" button</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>The discussion is correctly created and displayed in the "Discussion forum" section</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>"Topic" has not been chosen</li> <li>"Title" has not been entered</li> </ul> <p>an error message is shown and the form is shown again</p>

Name	[U.C.10] Farmer answers discussion forum
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>Farmer is correctly logged in</li> <li>A discussion is currently live in the section</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Farmer clicks on the "Discussion forum" section</li> <li>Farmer visualizes the page where all the discussions are presented and chooses to interact with one</li> <li>Farmer enters the answer to the discussion's topic in the relative input text area, positioned in the right corner of the discussion</li> <li>Farmer clicks "Send" to effectively reply</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>The answer is successfully registered in the discussion</li> </ul>
Exceptions	None

Name	[U.C.11] Farmer marks discussion as favourite
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>Farmer is correctly logged in</li> <li>A discussion is currently live in the section</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Farmer clicks on the "Discussion forum" section</li> <li>Farmer visualizes the page where all the discussions are presented and chooses to interact with one</li> <li>Farmer wants to save the discussion, as he/she is particularly interested in the topic, he/she clicks on the "star" button</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>The discussion is correctly saved in the "Favourite discussion" section</li> </ul>
Exceptions	None

Name	[U.C.12] Farmer answers his discussion's reply by another farmer
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>• Farmer is correctly logged in</li> <li>• His discussion is currently live in the section</li> <li>• Other Farmer has previously answered to his discussion</li> <li>• A notification has been shown in Farmer "Notification" section</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. Farmer clicks on the "Notification" section</li> <li>2. Farmer visualizes the list of all notification and clicks on the one related to the answer</li> <li>3. Farmer visualizes the section where the answer has been published</li> <li>4. Farmer writes his answer in the text area input</li> <li>5. Farmer submits the answer</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• The answer is successfully registered in the discussion</li> </ul>
Exceptions	None

Name	[U.C. 13] Farmer search for a discussion
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>• Farmer is correctly logged in</li> <li>• A discussion is currently live in the section</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. Farmer clicks on the "Discussion forum" section</li> <li>2. Farmer visualizes the page where all the discussions are presented and interacts with the "Filter menu"</li> <li>3. Farmer chooses the Topic</li> <li>4. Farmer enters a keyword to match the Title</li> <li>5. Farmer clicks the "Apply" button</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• Discussions matching the parameters chosen by the farmer are shown in the page</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>• There is no discussion that matches the parameters chosen</li> </ul> <p>"Not Found" error message is shown</p>

Name	[U.C. 14] Farmer visualizes favourite discussions
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>• Farmer is correctly logged in</li> <li>• Farmer has at least one favourite discussion</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. Farmer clicks on the "Discussion forum" section</li> <li>2. Farmer visualize the page and clicks on the "Favourite discussion" subsection</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• Farmer visualizes the page where all the discussion that he has previously marked as favourite</li> </ul>
Exceptions	None

Name	[U.C. 15] Farmer answers help suggestion
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>Farmer is correctly logged in</li> <li>Farmer has sent a “Help request”</li> <li>A notification has been shown in Farmer “Notification” section</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Farmer clicks on the “Notification” section</li> <li>Farmer visualizes all the notification</li> <li>Farmer clicks on the notification related to the answer</li> <li>Farmer visualizes the answer</li> <li>Farmer clicks in the input text area, types the answer and clicks the send button</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>The answer is registered and published in the section</li> </ul>
Exceptions	None

Name	[U.C. 16] Farmer visualizes steering initiative
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>Farmer is correctly logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Farmer clicks on “Steering Initiative” section</li> <li>Farmer visualizes the page where all the steering initiatives active are presented</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>Farmer can choose the steering initiative of interest</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>There is no steering initiative active</li> </ul> <p>“Not found” error message is shown</p>

Name	[U.C. 17] Farmer participates to steering initiative
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>Farmer is correctly logged in</li> <li>Farmer in “Steering Initiative” section</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>Farmer clicks on the “Read More” button of the steering initiative of interest</li> <li>Farmer visualizes the details of the steering initiative of interest</li> <li>Farmer clicks on the “Participate” button</li> <li>Farmer visualizes the form to insert the personal data of the relevant production</li> <li>Farmer enters the Type of product</li> <li>Farmer enters the Amount of product at the start of the initiative</li> <li>Farmer clicks the “Register” button</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>Farmer involvement is correctly registered in the system</li> <li>The chosen steering initiative is displayed in the “Personal Steering Initiative” section of the farmer</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>Not all the mandatory fields (“Type of product” and “Amount of product produced”) have been correctly filled</li> <li>Steering initiative is expiring while farmer is clicking on “Participate” button</li> </ul> <p>“Initiative no more available” error message is shown</p>

Name	[U.C. 18] Farmer inserts steering initiative final results
Actors	Registered farmer
Entry Conditions	<ul style="list-style-type: none"> <li>• Farmer is correctly logged in</li> <li>• Farmer is participating to at least one active steering initiative</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. Farmer clicks on “Steering Initiative” section</li> <li>2. Farmer visualizes the page where all the steering initiative active are presented and clicks on “Personal Steering Initiative” subsection</li> <li>3. Farmer visualizes the page where all the personal steering initiative are presented and selects the one, he wants to insert the final result of</li> <li>4. Farmer visualizes the form for the insertion of data of the relevant final production</li> <li>5. Farmer enters the Amount of product at the start of the initiative</li> <li>6. Farmer clicks the “Register” button</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• Final result correctly stored in the system</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>• “Amount of product produced” have been correctly filled</li> <li>• Farmer exceeds the expiration time to insert the final result</li> </ul> <p>“Expiration time exceeded” error message is shown</p>

Name	[U.C.19] Agronomist answer incoming help requests
Actors	Agronomist
Entry Conditions	<ul style="list-style-type: none"> <li>• Agronomist is correctly logged in</li> <li>• Some farmer has sent a help request</li> <li>• Agronomist has received a notification shown on visualize incoming help request section</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. The Agronomist clicks on the “Visualize incoming help request” section</li> <li>2. The Agronomist visualises all the help requests and chooses one by clicking on its title</li> <li>3. The details about the help request and all past conversation are shown, the Agronomist writes his answer in the text box</li> <li>4. The agronomist clicks the “answer” button to send the answer to the Farmer</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• The agronomist has sent the answer to a help request</li> </ul>
Exceptions	none

Name	[U.C.20] Agronomist visualizes data about best performing farmers
Actors	Agronomist
Entry Conditions	<ul style="list-style-type: none"> <li>• Agronomist is logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. Agronomist clicks on the best performing farmer’s button</li> <li>2. Agronomist visualizes his zone’s best performing farmer stats list</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• Agronomist successfully visualizes data about best performing farmers</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>• Best performing farmer stats list is empty An error message is shown</li> </ul>

Name	[U.C.21] Agronomist visualizes weather forecasts
Actors	Agronomist
Entry Conditions	<ul style="list-style-type: none"> <li>• Agronomist is correctly logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. The Agronomist clicks on the "Weather Forecasts" section</li> <li>2. Agronomist visualizes the weather forecast relative to the zone they are responsible of</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• Agronomist successfully visualizes data about the weather forecast</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>• Weather forecast’s provider is unavailable An error message is shown</li> </ul>

Name	[U.C.22] Agronomist inserts a new visit
Actors	Agronomist
Entry Conditions	<ul style="list-style-type: none"> <li>• Agronomist is correctly logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. Agronomist clicks on “daily plan”</li> <li>2. Agronomist clicks on the plus button</li> <li>3. Agronomist selects the Farmer he wants to visit from a list of Farmers</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• Agronomist has visualized the daily plan</li> </ul>
Exceptions	None

Name	[U.C.23] Agronomist confirms daily plan
Actors	Agronomist
Entry Conditions	<ul style="list-style-type: none"> <li>• Agronomist is correctly logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. Agronomist clicks on “daily plan”</li> <li>2. Agronomist checks all the boxes regarding the visits he made that day</li> <li>3. Agronomist clicks on the “confirm” button</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• Agronomist has confirmed the daily plan</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>• Daily plan is empty An error message is shown</li> </ul>

Name	[U.C.24] Agronomist publishes a new steering initiative
Actors	Agronomist
Entry Conditions	<ul style="list-style-type: none"> <li>• Agronomist is correctly logged in</li> </ul>
Events Flow	<ol style="list-style-type: none"> <li>1. Agronomist clicks on “steering initiative”</li> <li>2. Agronomist clicks on the plus button</li> <li>3. Agronomist clicks on Topic</li> <li>4. Agronomist selects a Topic from the list</li> <li>5. Agronomist writes a title in the “Title” section</li> <li>6. Agronomist writes a description in the “Description” section</li> </ol>
Exit Conditions	<ul style="list-style-type: none"> <li>• Agronomist has successfully published a steering initiative</li> </ul>
Exceptions	<ul style="list-style-type: none"> <li>• A steering initiative with the same Title already exists An error message is shown</li> </ul>

## 2.3 Sequence Diagrams

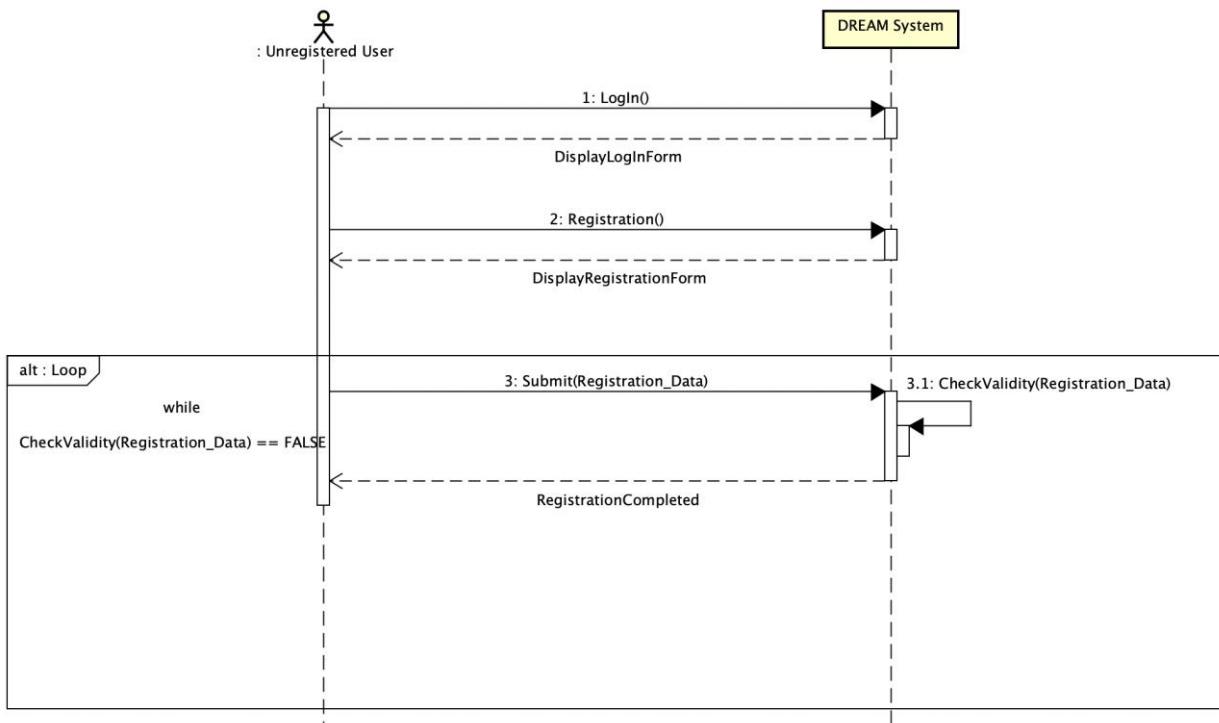


Figure 31: User Registration

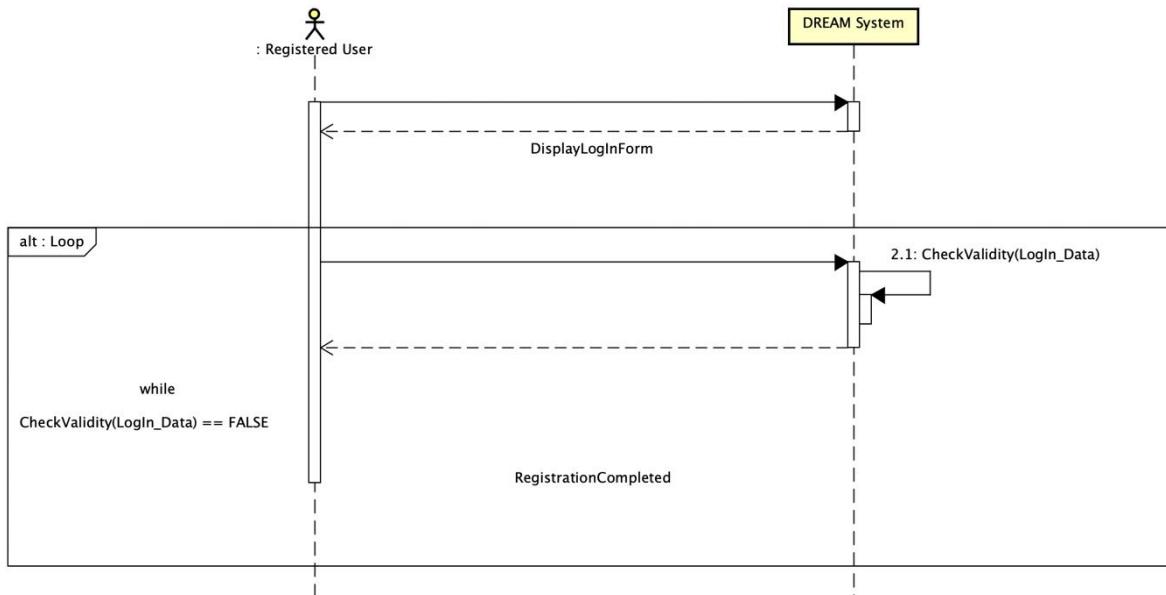


Figure 32: Login

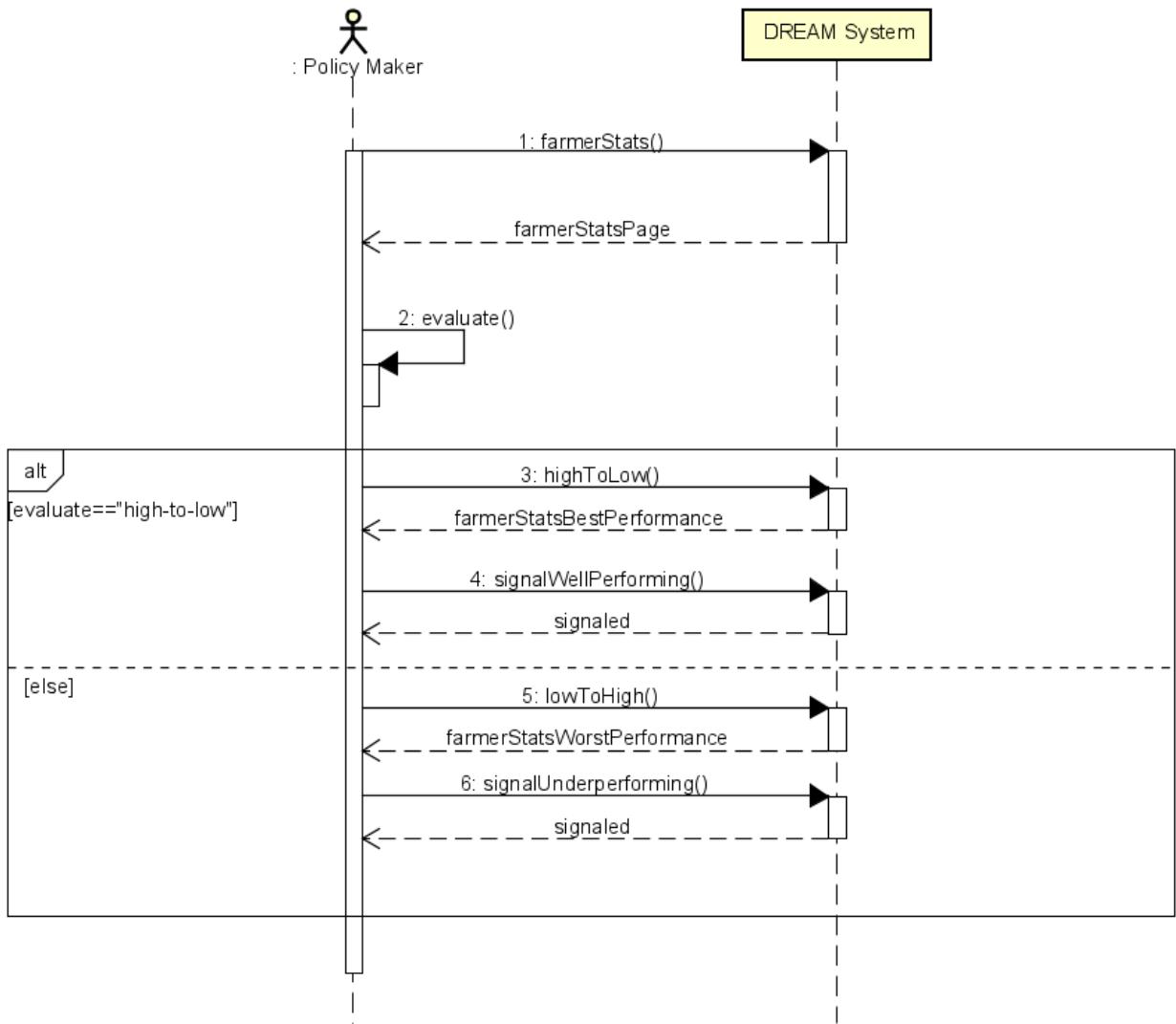


Figure 33: Policy Maker identifies farmer performance

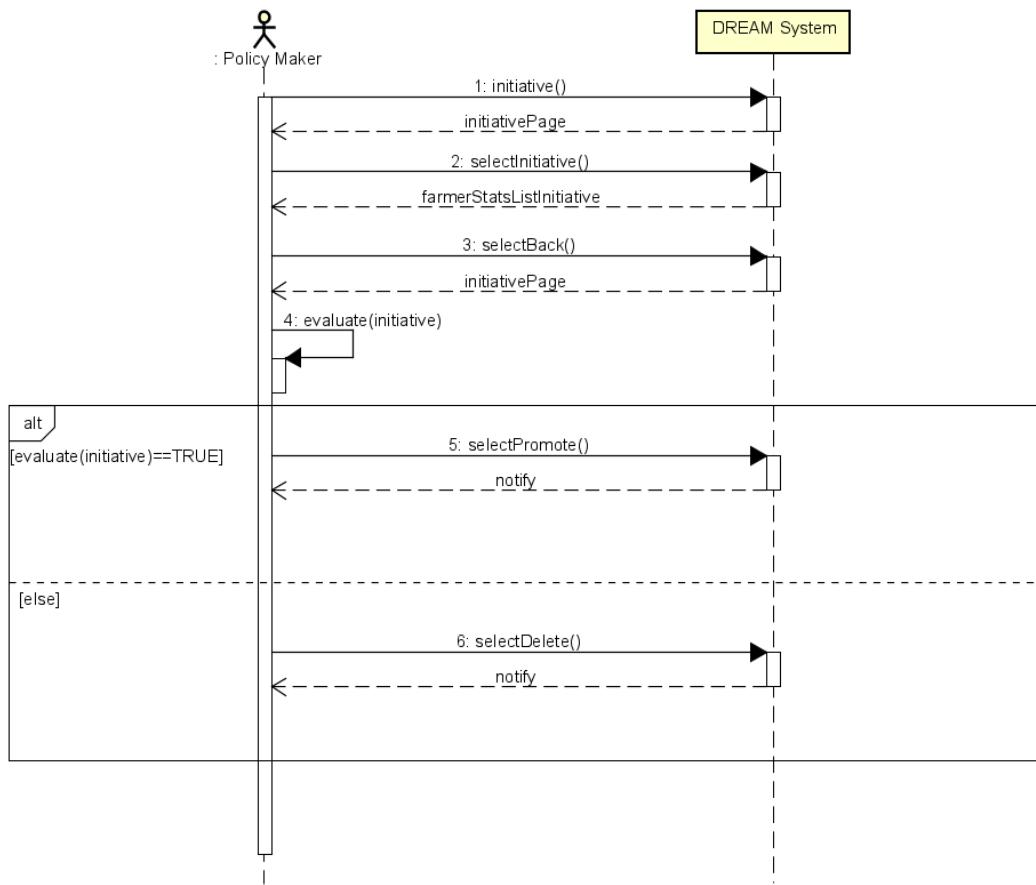


Figure 34: Policy Maker evaluates Steering Initiative

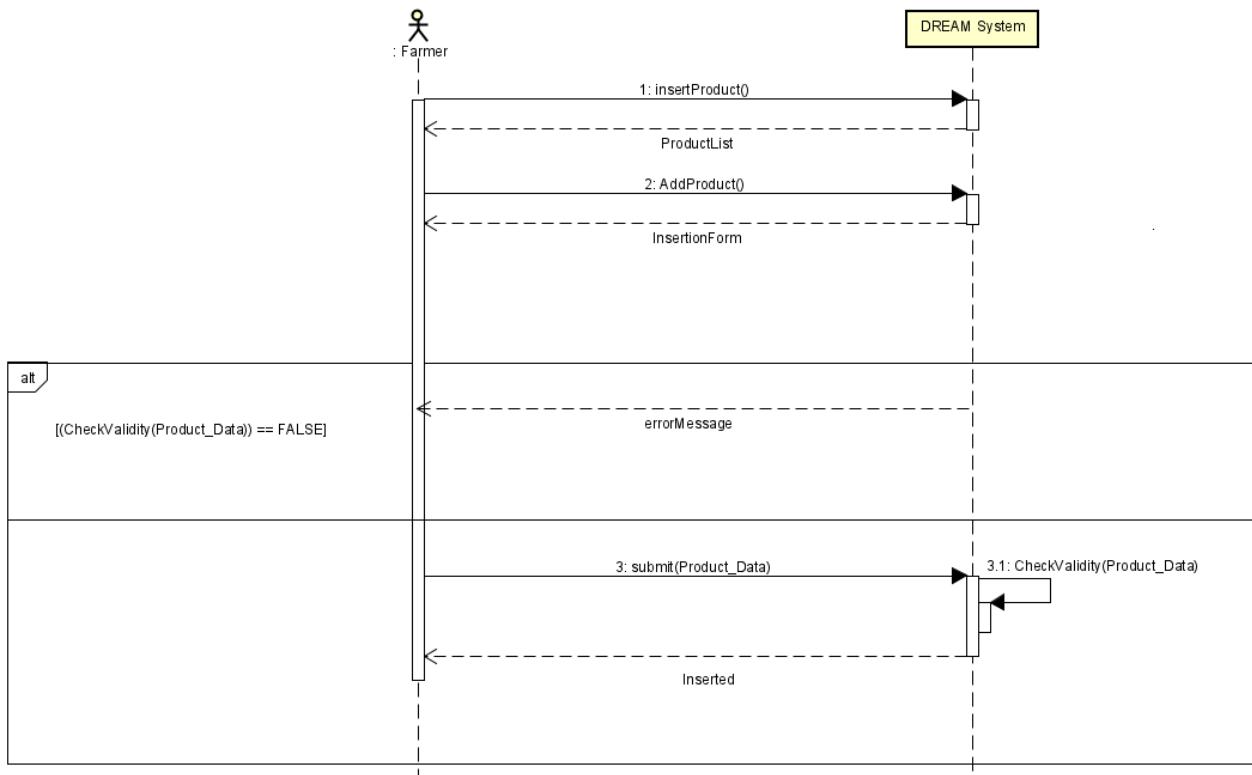


Figure 35: Farmer inserts Production

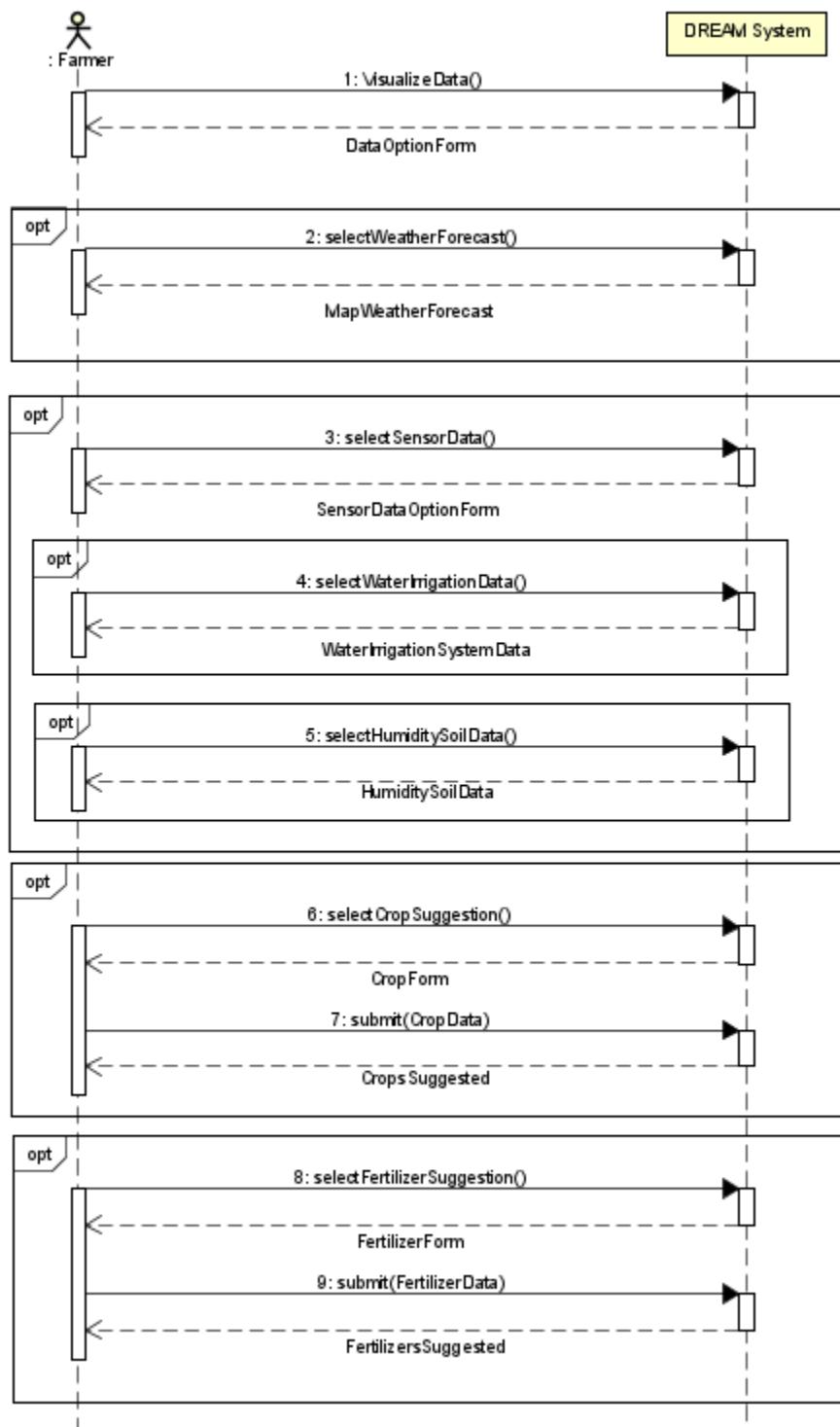


Figure 36: Farmer visualizes data

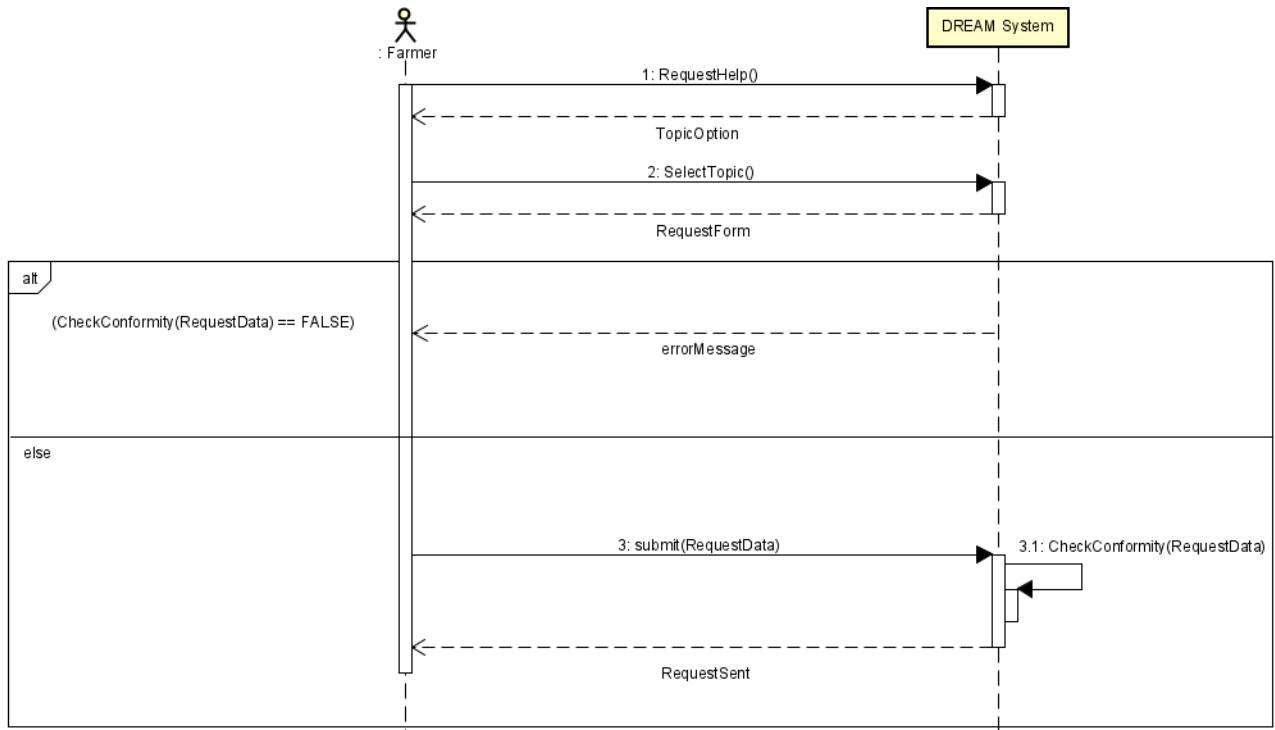


Figure 37: Farmer requests for help

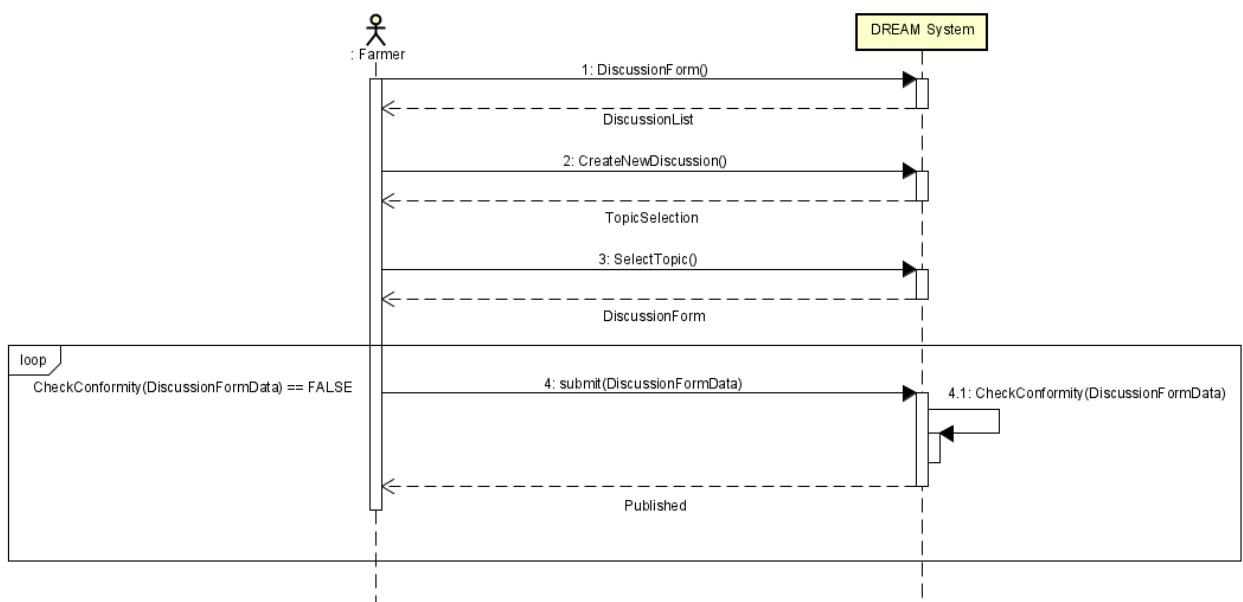


Figure 38: Farmer creates discussion

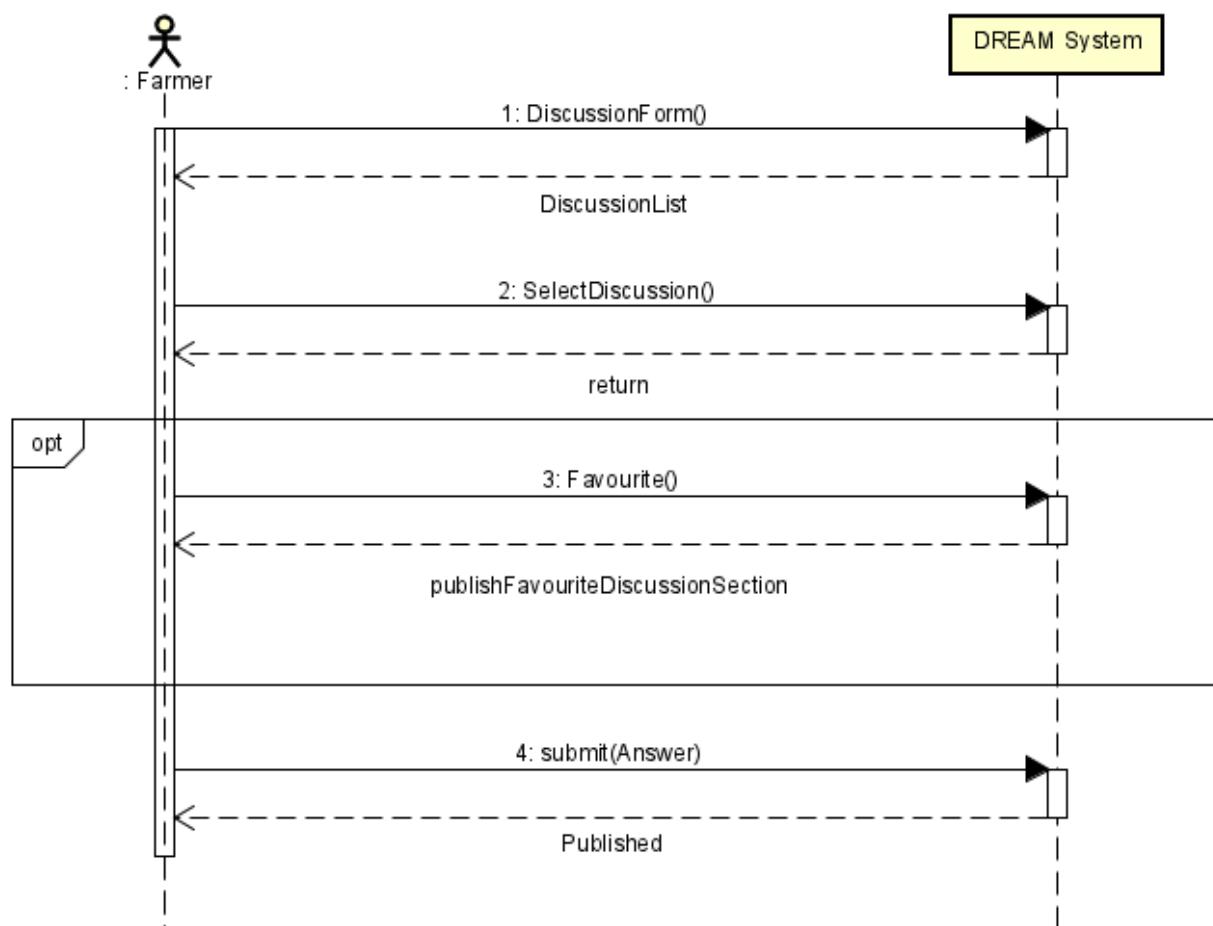


Figure 39: Farmer answers discussion (opt: Farmer marks discussion as favourite)

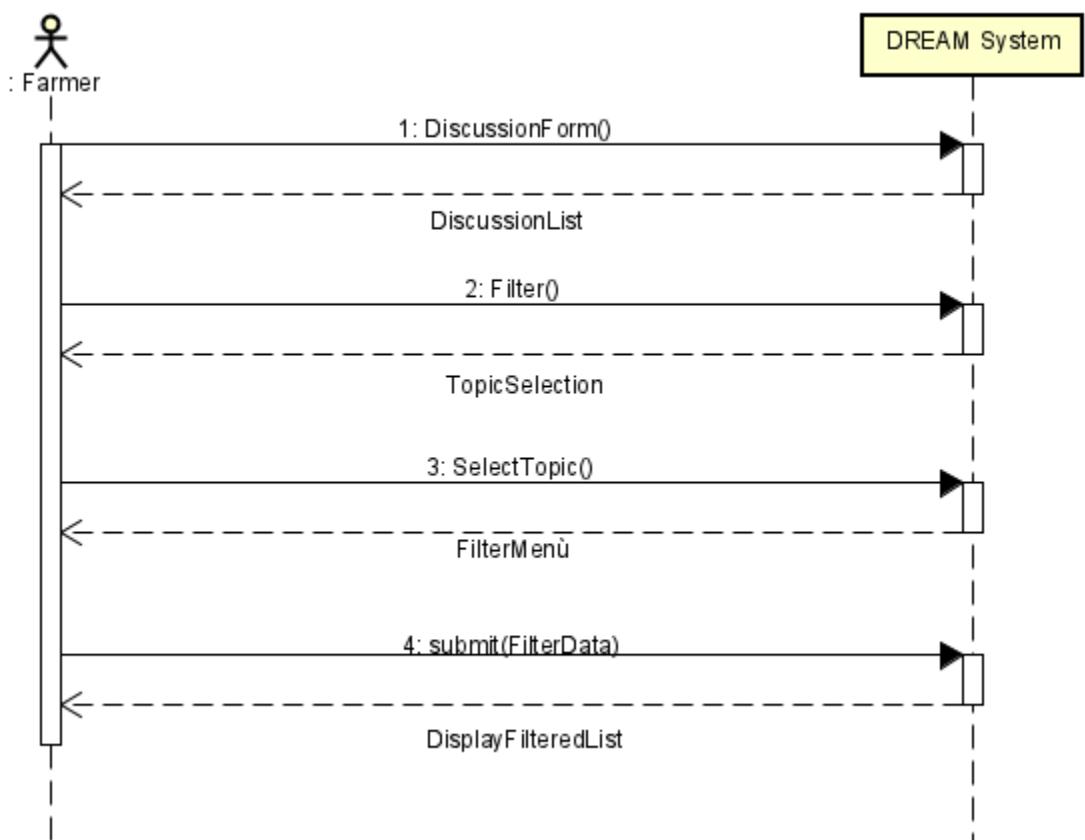


Figure 40: Farmer search for a discussion

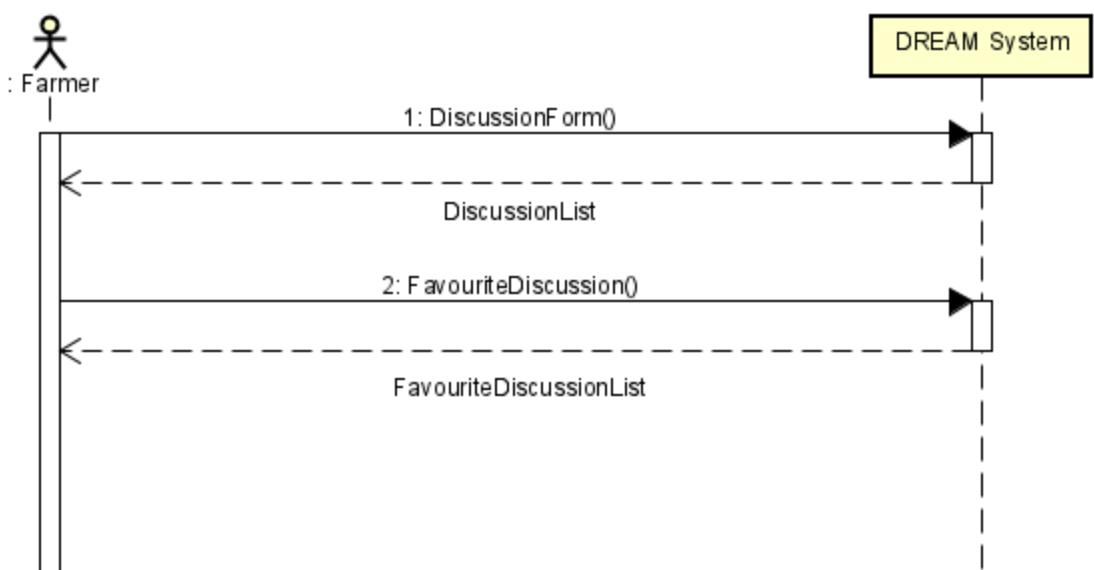


Figure 41: Farmer visualizes favourite discussion

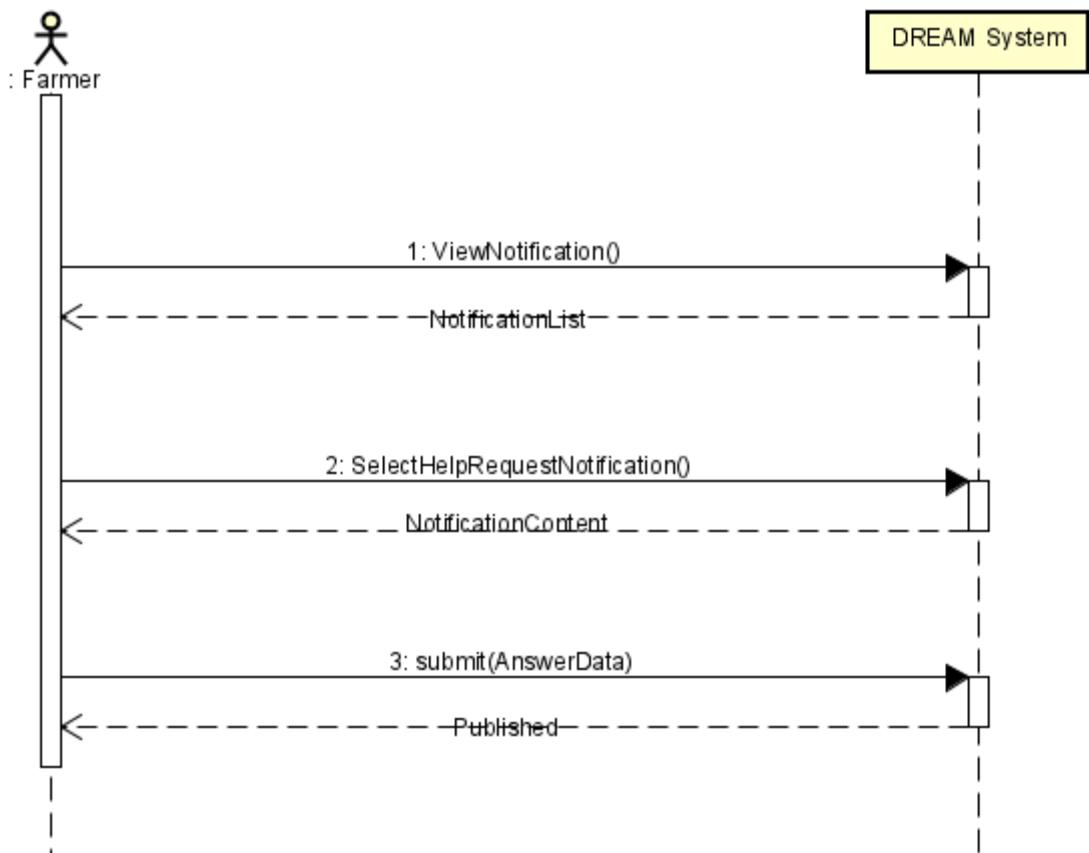


Figure 42: Farmer answers help suggestion

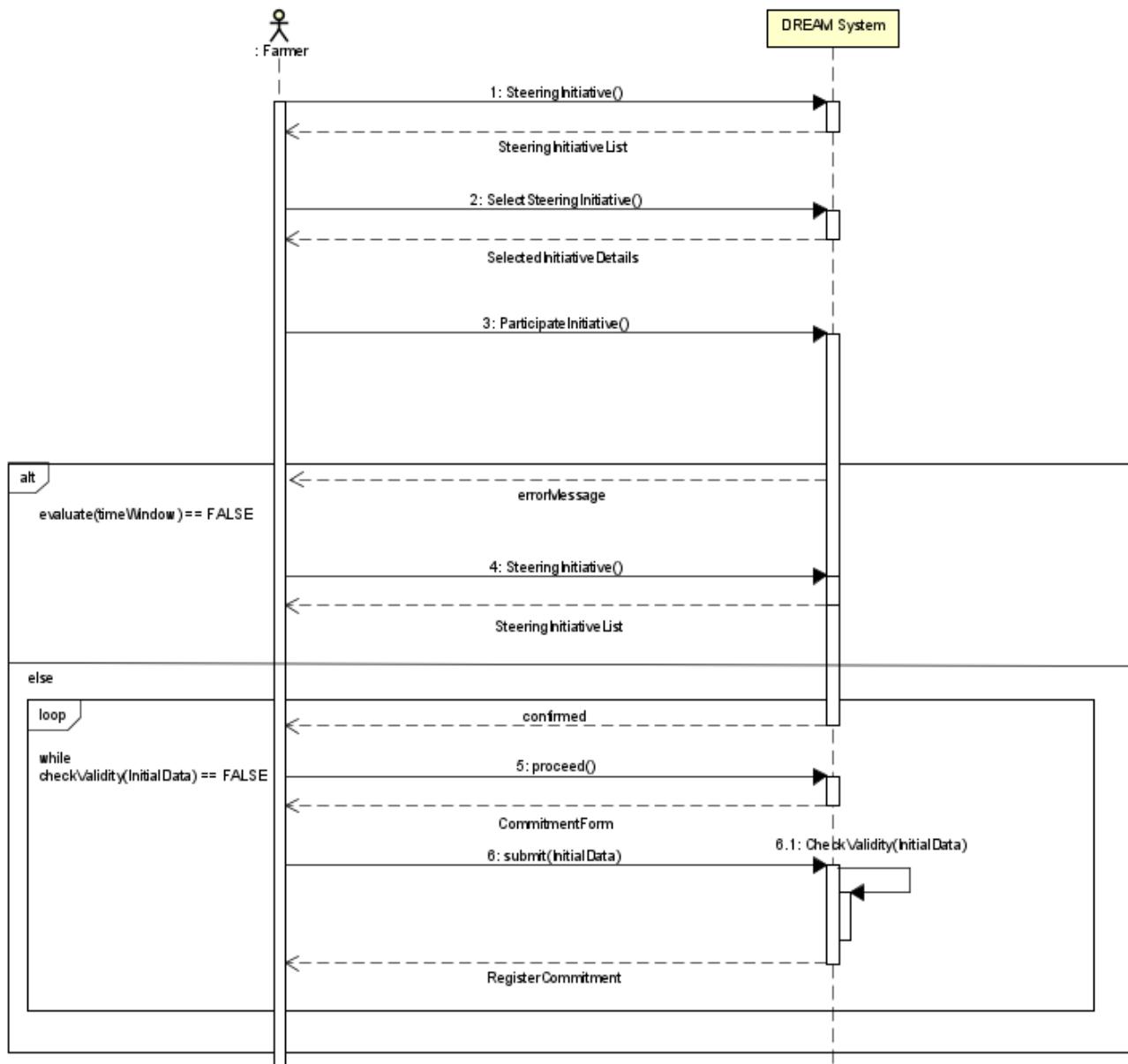


Figure 43: Farmer participates to steering initiative

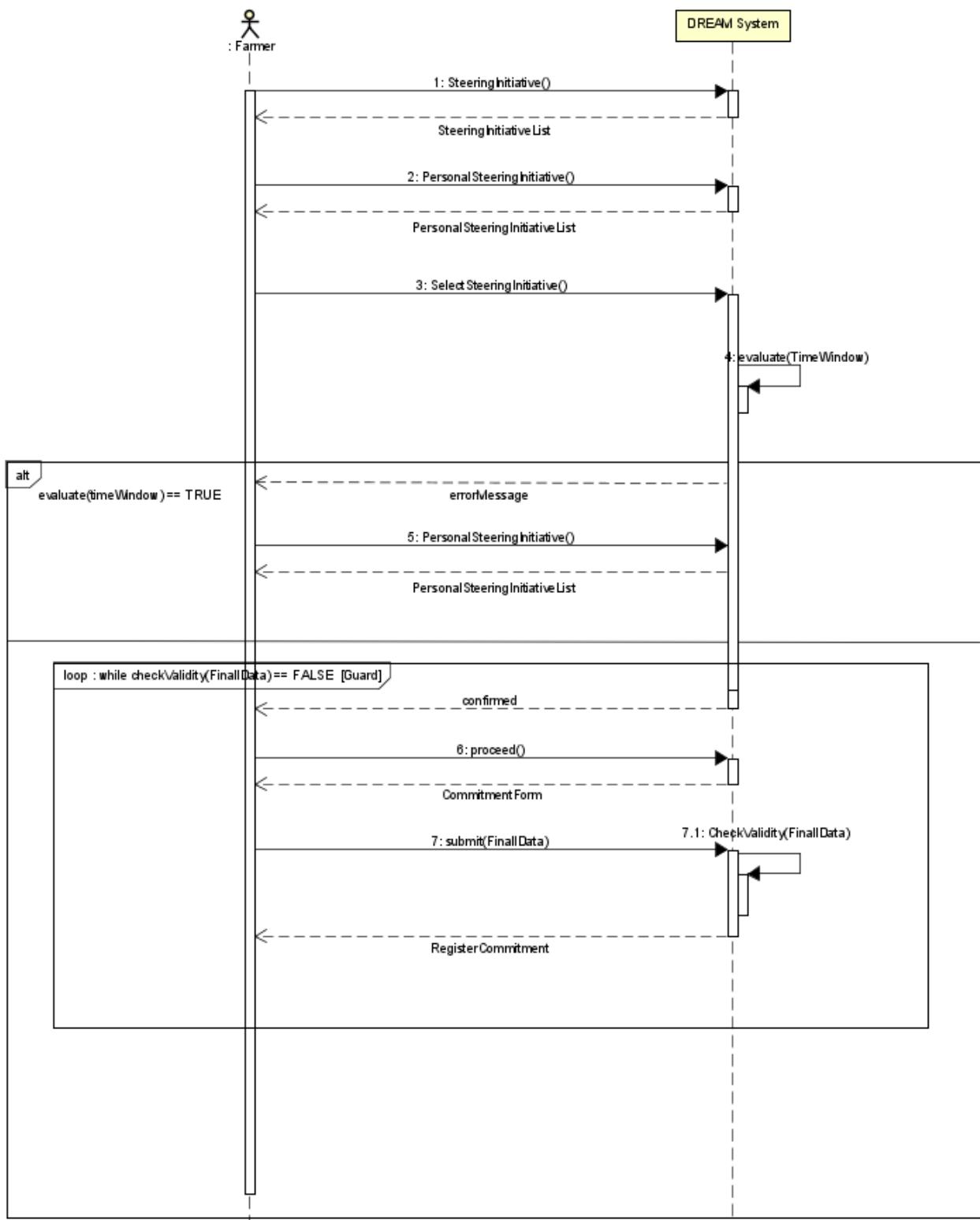


Figure 44: Farmer inserts steering initiative final result

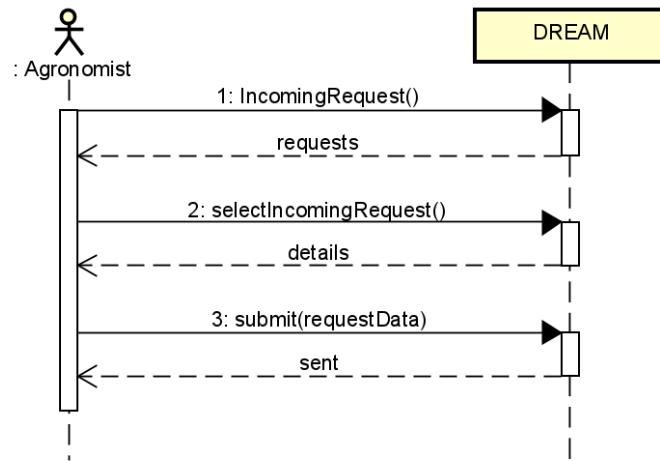


Figure 45: Agronomist answers incoming help request

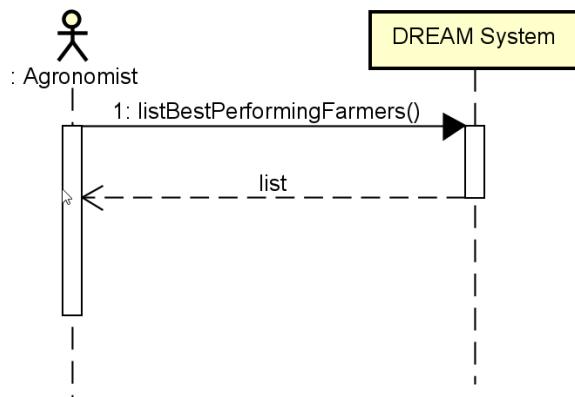


Figure 46: Agronomist visualizes data about best performing farmers

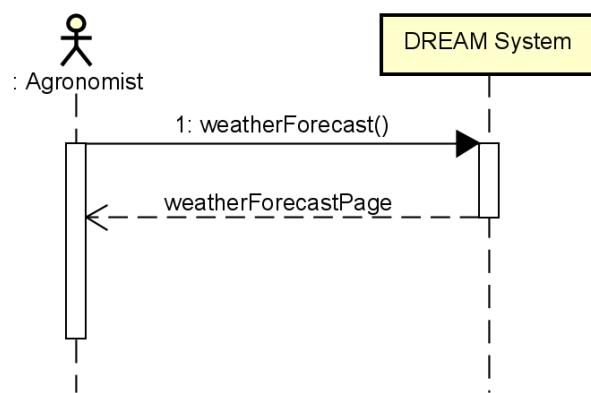


Figure 47: Agronomist visualizes weather forecasts

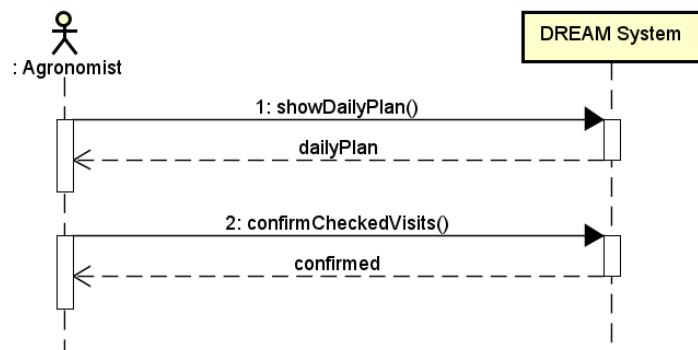


Figure 48: Agronomist inserts a new visit

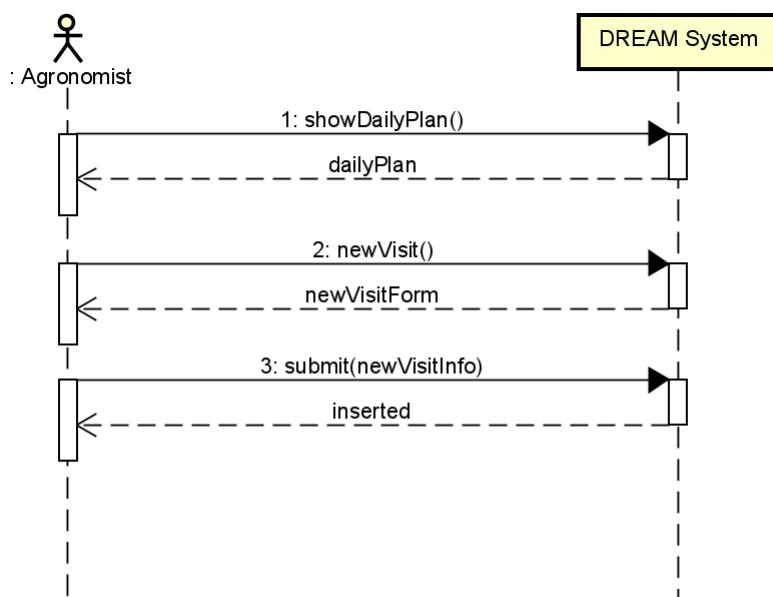


Figure 49: Agronomist confirms daily plan

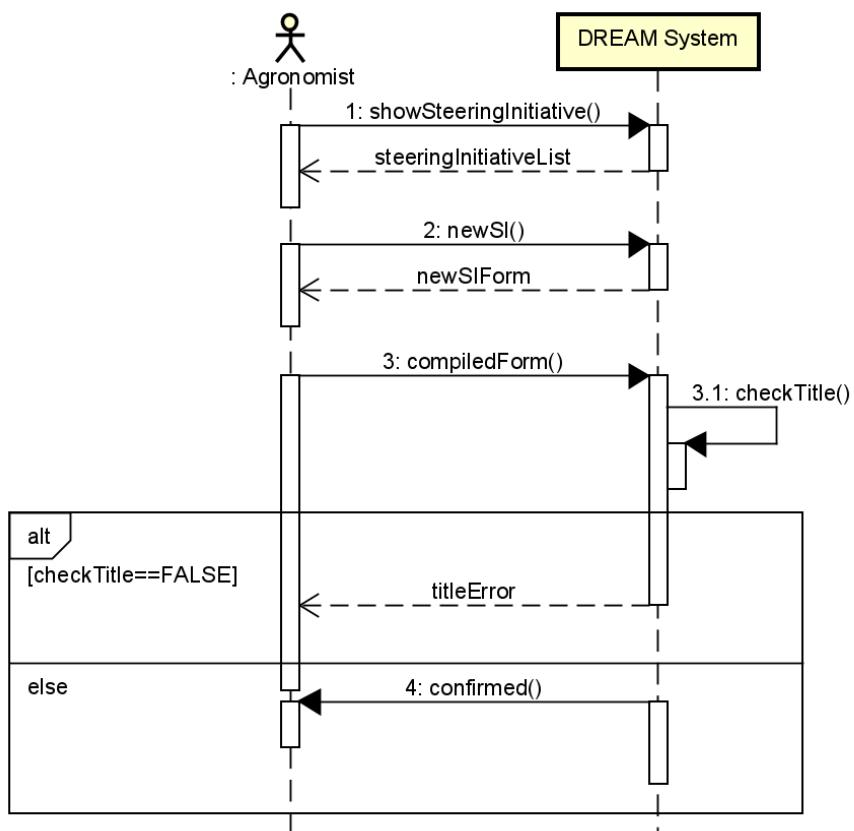


Figure 506: Agronomist publishes a new Steering Initiative

## 2.4 Requirements

- [G.1] Allows policy makers to identify well performing farmers
  - [R.1] The system allows policy maker to visualize list of farmers
  - [R.2] The system allows policy maker to visualize farmer's statistics
  - [R.3] The system allows policy maker to sort farmers depending on their performance
  - [R.4] The system allows policy maker to report the best performing farmers
  - [D.1] Farmers are supposed to insert only correct data
  - [D.6] Policy makers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure
- [G.2] Allows policy makers to identify farmers in need of help
  - [R.1] The system allows policy maker to visualize list of farmers
  - [R.2] The system allows policy maker to visualize farmer's statistics
  - [R.3] The system allows policy maker to sort farmers depending on their performance
  - [R.5] The system allows policy maker to report the worst performing farmers
  - [D.1] Farmers are supposed to insert only correct data
  - [D.6] Policy makers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure
- [G.3] Allows policy makers to evaluate Agronomist's steering initiative
  - [R.6] The system allows policy maker to visualize list of steering initiative
  - [R.2] The system allows policy maker to visualize farmer's statistics
  - [R.7] The system allows policy maker to promote/reject steering initiative
  - [D.1] Farmers are supposed to insert only corrected data
  - [D.4] Farmers are supposed to own an internet capable device
  - [D.5] Agronomists are supposed to own an internet capable device
  - [D.6] Policy makers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure
- [G.4] Allows farmers to visualize relevant data based on location and type of production
  - [R.8] The system allows farmer to visualize weather forecast data
  - [R.9] The system allows farmer to visualize sensor's collected data
  - [R.10] The system allows farmer to visualize suggestion for crop
  - [R.11] The system allows farmer to visualize suggestion for fertilizer
  - [D.1] Farmers are supposed to insert only corrected data
  - [D.2] Sensors are supposed to be accurate
  - [D.4] Farmers are supposed to own an internet capable device
  - [D.7] The external service used by the system to check weather forecast is supposed to be accurate
  - [D.10] The internet connection is supposed to work flawlessly, without any failure
- [G.5] Allows farmers to insert in the system data about their production and any problem they face
  - [R.12] The system allows farmer to visualize the list of all products
  - [R.13] The system must correctly save and store any data inserted by the farmer
  - [D.1] Farmers are supposed to insert only corrected data
  - [D.4] Farmers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure

- [G.6] Allows farmers to ask for help
  - [R.14] The system allows farmer to visualize the list of requests previously done
  - [R.13] The system must correctly save and store any data inserted by the farmer
  - [D.1] Farmers are supposed to insert only corrected data
  - [D.4] Farmers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure
- [G.7] Allows farmers to reply to help suggestions
  - [R.15] The system must correctly display the notification badge
  - [R.16] The system allows farmer to visualize the list of notification
  - [R.13] The system must correctly save and store any data inserted by the farmer
  - [D.1] Farmers are supposed to insert only corrected data
  - [D.4] Farmers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure
- [G.8] Allows farmers to create discussions forum
  - [R.17] The System allows farmer to visualize the list of all discussions forum
  - [R.18] The System must allow Farmer to create discussion forum
  - [R.13] The System must correctly save and store any data inserted by the farmer
  - [D.1] Farmers are supposed to insert only corrected data
  - [D.4] Farmers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure
- [G.9] Allows farmers to answer to discussions forum
  - [R.17] The System allows farmer to visualize the list of all discussions forum
  - [R.19] The System must allow Farmer to answer to discussion forum
  - [R.13] The System must correctly save and store any data inserted by the farmer
  - [D.1] Farmers are supposed to insert only corrected data
  - [D.4] Farmers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure
- [G.10] Allows farmers to search specific discussions forum
  - [R.17] The System allows farmer to visualize the list of all discussions forum
  - [R.20] The System must allow Farmer to choose his favorite topic
  - [R.13] The System must correctly save and store any data inserted by the farmer
  - [R.21] The System allows farmer to visualize the list of a filtered discussions forum
  - [D.1] Farmers are supposed to insert only corrected data
  - [D.4] Farmers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure
- [G.11] Allows farmers to participate steering initiative
  - [R.22] The System allows farmer to visualize the list of all steering initiative
  - [R.23] The System must allow Farmer to participate his favorite steering initiative
  - [R.13] The System must correctly save and store any data inserted by the farmer
  - [D.4] Farmers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure

- [G.12] Allows farmers to insert result of steering initiative
  - [R.22] The System allows farmer to visualize the list of all steering initiative
  - [R.24] The System must allow Farmer to visualize his personal steering initiative
  - [R.25] The System must allow Farmer to read more about a steering initiative
  - [R.13] The System must correctly save and store any data inserted by the farmer
  - [D.1] Farmers are supposed to insert only corrected data
  - [D.4] Farmers are supposed to own an internet capable device
  - [D.10] The internet connection is supposed to work flawlessly, without any failure
  - [D.12] Farmers are supposed to insert final result of steering initiative before expiration time
  
- [G.13] Allows agronomists to insert the region they are responsible of
  - [R.26] The System must ask in the registration form if the new user is an agronomist
  - [R.27] The System must allow agronomist to insert the region they are responsible of during the registration process
  - [R.28] The System must store the data about the agronomist's region
  - [D.5] Agronomists are supposed to own an internet capable device
  - [D.11] Agronomists are supposed to insert the region they are responsible of correctly
  
- [G.14] Allows agronomists to receive and answer requests for help
  - [R.29] The System must allow agronomist to visualize the help requests addressed to them
  - [R.30] The System must allow agronomist to visualize all previous interactions regarding a help request
  - [R.31] The System must allow agronomist to enter some text
  - [R.32] The System must allow agronomist to send some text as an answer
  - [D.5] Agronomists are supposed to own an internet capable device
  
- [G.15] Allows agronomists to visualize data concerning both weather forecast and best performing farmer in an region
  - [R.33] The System must allow agronomist to visualize weather forecasts
  - [R.34] The System must allow agronomist to visualize list of farmers
  - [R.35] The System must allow agronomist to identify best performing farmers
  - [D.7] The external service used by the system to check the weather forecast is supposed to be accurate
  
- [G.16] Allows agronomists to visualize, update and confirm daily plans to visit farms
  - [R.36] The System must allow agronomist to visualize his daily plan
  - [R.37] The System must compile agronomists daily plan for the current day before 6:00 AM
  - [R.38] The system must allow agronomists to add a visit to the daily plan
  - [R.34] The system must allow agronomist to visualize the list of Farmers
  - [R.39] The system must allow agronomist to check which visit they did
  - [R.40] The system must allow agronomist to confirm the daily plan
  - [D.8] Agronomists are supposed to confirm only those farms they have visited as scheduled in daily plan
  - [D.9] Agronomist is supposed to visit a farmer in need of help in his region

## 2.5 Traceability Matrix

Goal	Requirement	Domain
[G.1]	[R.1], [R.2], [R.3], [R.4]	[D.1], [D.6], [D.10]
[G.2]	[R.1], [R.2], [R.3], [R.5]	[D.1], [D.6], [D.10]
[G.3]	[R.2], [R.6], [R.7]	[D.1], [D.4], [D.5], [D.6], [D.10]
[G.4]	[R.8], [R.9], [R.10], [R.11]	[D.1], [D.2], [D.4], [D.7], [D.10]
[G.5]	[R.12], [R.13]	[D.1], [D.4], [D.10]
[G.6]	[R.13], [R.14]	[D.1], [D.4], [D.10]
[G.7]	[R.13], [R.15], [R.16]	[D.1], [D.4], [D.10]
[G.8]	[R.13], [R.17], [R.18]	[D.1], [D.4], [D.10]
[G.9]	[R.13], [R.17], [R.19]	[D.1], [D.4], [D.10]
[G.10]	[R.13], [R.17], [R.20]	[D.1], [D.4], [D.10]
[G.11]	[R.13], [R.22], [R.23]	[D.4], [D.10]
[G.12]	[R.13], [R.22], [R.24], [R.25]	[D.1], [D.4], [D.10]
[G.13]	[R.26], [R.27], [R.28]	[D.5], [D.11]
[G.14]	[R.29], [R.30], [R.31], [R.32]	[D.5]
[G.15]	[R.33], [R.34], [R.35]	[D.7]
[G.16]	[R.34], [R.36], [R.37], [R.38], [R.39], [R.40]	[D.8], [D.9]

<b>Requirement</b>	<b>Use Cases</b>
[R.1]	Policy Maker identifies farmer performance
[R.2]	Policy Maker identifies farmer performance Policy Maker evaluate steering initiative
[R.3]	Policy Maker identifies farmer performance
[R.4]	Policy Maker identifies farmer performance
[R.5]	Policy Maker identifies farmer performance
[R.6]	Policy Maker evaluate steering initiative
[R.7]	Policy Maker evaluate steering initiative
[R.8]	Farmer visualizes data
[R.9]	Farmer visualizes data
[R.10]	Farmer visualizes data
[R.11]	Farmer visualizes data
[R.12]	Farmer inserts data
[R.13]	Farmer inserts data Farmer requests for help Farmer answers agronomist's help suggestion Farmer creates discussion forum Farmer answers discussion forum Farmer search for a discussion Farmer participates to steering initiative
[R.14]	Farmer requests for help
[R.15]	Farmer answers agronomist's help suggestion
[R.16]	Farmer answers agronomist's help suggestion
[R.17]	Farmer creates discussion forum Farmer answers discussion forum Farmer search for a discussion
[R.18]	Farmer creates discussion forum
[R.19]	Farmer answers discussion forum
[R.20]	Farmer search for a discussion
[R.21]	Farmer search for a discussion
[R.22]	Farmer participates to steering initiative Farmer inserts steering initiative final results
[R.23]	Farmer participates to steering initiative
[R.24]	Farmer inserts steering initiative final results
[R.25]	Farmer inserts steering initiative final results
[R.26]	User Registration
[R.27]	User Registration
[R.28]	User Registration
[R.29]	Agronomist answer incoming help requests
[R.30]	Agronomist answer incoming help requests
[R.31]	Agronomist answer incoming help requests
[R.32]	Agronomist answer incoming help requests
[R.33]	Agronomist visualizes data about best performing farmers Agronomist visualizes weather forecasts
[R.34]	Agronomist visualizes data about best performing farmers Agronomist visualizes weather forecasts Agronomist confirms daily plan
[R.35]	Agronomist visualizes data about best performing farmers Agronomist visualizes weather forecasts

[R.36]	Agronomist confirms daily plan
[R.37]	Agronomist confirms daily plan
[R.38]	Agronomist confirms daily plan Agronomist inserts a new visit
[R.39]	Agronomist confirms daily plan
[R.40]	Agronomist confirms daily plan

## **3 Performance Requirements**

The system has a performance requirement with:

- the response time concerning the arrival of the notification of the daily plan with each agronomist.

By the end of the day, the system should be able to compute the daily plan per agronomist.

## **4 Design Constraint**

### **4.1 Standards compliance**

1. The geo-location coordinates must be encoded as longitude and latitude degrees.
2. Dates and times must always be expressed according to the ISO 8601 standard, as the software is meant to be used in India

### **4.2 Hardware limitations**

1. Every user of the application must be able to exchange data through internet connection
2. The devices used by all the users must be internet capable in order to exchange data with the platform
3. The devices used by the farmers and policy makers must be able to interact with sensors placed in the ground, as well as water irrigation system, so that it would be possible to collect all the relevant data

### **4.3 Any other constraint**

- The application should be easy to use and very intuitive for each type of user, to allow a more fluid integration

## 5 Software System Attributes

### 5.1 Reliability

The System must be robust and tolerant to any type of fault

### 5.2 Availability

The system should be available up to 99% of the time.

In practical terms, the percentage points out that the average time within the occurrence of a fault and the recovery from it should not be higher than ~3.65 days every year.

This number results from the calculation of the ***Mean Time To Repair (MTTR)***.

### 5.3 Security

In order to protect sensitive data against malicious attacks, DREAM software uses protocols to encrypt data exchanged in communication.

Passwords are hashed before being stored, and data in the database are protected against intrusion by unauthorized agents trying to access them.

### 5.4 Maintainability

The software is being developed following a well-documented structure and implementing easy to upgrade components, all to guarantee maintainability and immediate integration of new pieces of hardware and software.

### 5.5 Portability

The software must be compatible with the most used modern OS, such as Windows, Mac and Linux for the pc compartment, and Android and iOS for the mobile one.

# Chapter 4

## Formal Analysis using Alloy

In this section the main components of the software are presented and modelled using Alloy language, specialized in software modelling.

### 1 Alloy model

The Alloy model is specialized in showing in details the interaction between main actor: Farmer, Agronomist and Policy Maker. In particular, the model aims at analyzing:

- The insertion of production unit, the participation to steering initiative, the creation and management of discussion, the possibility to send a help request by Farmer
- The evaluation of steering initiatives, the evaluation of farmer's performance by Policy Maker
- The management and update of a daily plan, the creation of a steering initiative, the possibility to answer help request by agronomist

In addition, some aspects of the model must be preserved:

- The agronomist can visit and answer only farmers located in his/her region
- A farmer can not insert two different production which refer to the same steering initiative

## 1.1 Signatures

```
----- User: Agronomist, Farmer Policy Maker
abstract sig User {
    username: one Username,
    password: one Password,
    email: one Email
}

----- Farmer
sig Farmer extends User{

}

----- Agronomist
sig Agronomist extends User{
    region: one Region,
    steeringInitiative: some SteeringInitiative,
}

----- Policy Maker
sig PolicyMaker extends User{

}

----- User's Username
sig Username {}

----- User's password
sig Password {} {
    all p: Password | (some r: User | r.password = p)
}

----- User's email
sig Email {}

----- Farmer's farm location
sig FarmLocation{
    farmer: one Farmer,
    coordinates : one Coordinates,
    region: one Region
}

----- Coordinates farm location
sig Coordinates{}

----- Farmer's production
sig Production{
    date: one Date,
    type : one Type,
    amount : one Amount,
    description : one Description,
    farmer: one Farmer
}

----- Date added production
sig Date{}
```

```

----- Type of product
sig Type{}

----- Amount of product
sig Amount{}

----- Generic Description
sig Description {}

----- Description message
sig Discussion{
    farmer: one Farmer,
    description : one Description,
}

----- Participation in a steering initiative inserting production
sig InitProduction{
    farmer: one Farmer,
    steeringInitiative: one SteeringInitiative,
    beforeProd: one Production,
    afterProd: one Production
} { (all disj p1, p2: Production| p1.type=p2.type)
    and (no disj p3, p4: Production | p3.date=p4.date) }

----- Agronomist's region
----- Farmer's region
sig Region{}

----- Steering Initiative
sig SteeringInitiative{
    startingDate: one Date,
    endingDate: one Date,
    description: one Description,
    farmer: some Farmer
}
{
    no disj f1, f2: Farmer | f2.email = f1.email
}

----- Steering Initiative evaluation status
abstract sig InitiativeStatus{}
one sig PROMOTE extends InitiativeStatus{}
one sig REJECT extends InitiativeStatus{}

----- Help Request
sig HelpRequest{
    agronomist: one Agronomist,
    status: one HRStatus
}

----- Help Request Answer Status
abstract sig HRStatus {}
lone sig Answered extends HRStatus{}
lone sig Pending extends HRStatus{}

```

```

----- Evaluate Farmer
sig EvaluateFarmer{
    farmer: one Farmer,
    upcomingStatus: one SetStatus,
    policyMaker: one PolicyMaker
}

----- Evaluate Steering Initiative
sig EvaluateInitiative{
    steeringInitiative: one SteeringInitiative,
    upcomingDecision: one InitiativeStatus,
    policyMaker: one PolicyMaker
}

----- Notification Help Request (Agronomist)
sig Notification{
    helpRequest: one HelpRequest
}
helpRequest.status in Pending

----- Performance Farmer
sig Performance{
    policyMaker : one PolicyMaker,
    farmer : one Farmer,
    ratio : one Ratio,
}
sig Ratio{ }

----- Evaluate Farmer status
abstract sig SetStatus{}
one sig PAUSE extends SetStatus{}
one sig BEST extends SetStatus{}
one sig WORST extends SetStatus{}

----- Agronomist's daily plan
sig DailyPlan{
    date: one Date,
    agronomist: one Agronomist,
    farmer: some Farmer
}

```

## 1.2 Facts

```
fact{ //Each can not do the same Steering Initiative many times
    one far: Farmer | no disj i1, i2: InitProduction |
        (i1.farmer = far and i2.farmer = far) and
        i1.steeringInitiative = i2.steeringInitiative
}

fact{ //A unique username is associated to each registration
    no disj r1, r2: User | r1.username = r2.username
}

fact{ //A unique password is associated to each user
    no disj u1, u2: User | u1.password = u2.password
}

fact{ //A unique email is associated to each user
    no disj u1, u2: User | u1.email = u2.email
}

fact { //There cannot exist two Farms with the same position
    no disj f11, f12: FarmLocation | f12.coordinates = f11.coordinates
}

fact{ //Production inserted before and production inserted after can not be the same
    all Ip:InitProduction | (Ip.beforeProd.type=Ip.afterProd.type) and
        (Ip.beforeProd.date!=Ip.afterProd.date)
}

fact{ //Policy Maker doesn't evaluate the same initiative many times
    no disj e1,e2:EvaluateInitiative|
        e1.steeringInitiative=e2.steeringInitiative
}

fact { //Before e after production in init production in the same farmer
    all i : InitProduction| i.beforeProd.farmer=i.afterProd.farmer and
        i.afterProd.farmer=i.farmer
}

fact { //No empty daily plan
    no d: DailyPlan | #{f: Farmer | f in d.farmer} = 0
}

fact{ //Two daily plan can not have the same date
    all disj d1, d2: DailyPlan | d1.date != d2.date
}

fact{ //Two notifications can not be created by the same help request
    all disj n1, n2: Notification | n1.helpRequest != n2.helpRequest
}
```

```

fact{ //Two production initiative production can't have the same
starting and ending production
no disj i1, i2: initProduction| i1.beforeProd=i2.beforeProd and
i1.afterProd=i2.afterProd
}

fact{ // Help request must belong to the Farmer with the same region as the
Agronomist
all f:HelpRequest| all fl:FarmLocation| f.farmer=fl.farmer and (all
a:Agronomist| fl.region=a.region)
}

fact{
no disj f1, f2: FarmLocation| f1.farmer=f2.farmer
}

fact { //Each region has at most one agronomist
no disj a1, a2: Agronomist | a1.region = a2.region
}

fact{
no disj s1, s2: SteeringInitiative| s1.farmer=s2.farmer
}

fact{ //Two evaluateFarmer can not be relevant to the same Farmer
no disj e1, e2: EvaluateFarmer | e1.farmer != e2.farmer
}

fact{ //Every steering initiative's ending and starting date must be
different
no si: SteeringInitiative| si.startingDate=si.endingDate
}

fact{ //Matching DailyPlan
no d: DailyPlan | some f: FarmLocation| (f.farmer=d.farmer) and
(d.agronomist.region != f.region)
}

fact{
no f:Farmer| some fl:FarmLocation| f not in fl.farmer
}

```

### 1.3 Assertions

```
//No two Agronomist with the same Farmer
assert noTwoAgronomistSameFarmer{
    no disj a1,a2:Agronomist | one fl:FarmLocation |
    a1.region=fl.region and a2.region=fl.region
}
check noTwoAgronomistSameFarmer

//No two different Farmer with the same Help Request
assert noHelpRequestTwoDifferentFarmer{
    no disj f1,f2:Farmer| one hr:HelpRequest | hr.farmer=f1 and
    hr.farmer=f2
}
check noHelpRequestTwoDifferentFarmer
```

## 1.4 Analysis results

```
//No two Agronomist with the same Farmer
```

```
Starting the solver...
```

```
Executing "Check noTwoAgronomistSameFarmer"
```

```
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
```

```
Mode=batch
```

```
Generating CNF...
```

```
No counterexample found. Assertion may be valid. 3ms.
```

```
//No two different Farmer with the same Help Request
```

```
Starting the solver...
```

```
Executing "Check noHelpRequestTwoDifferentFarmer"
```

```
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
```

```
Mode=batch
```

```
Generating CNF...
```

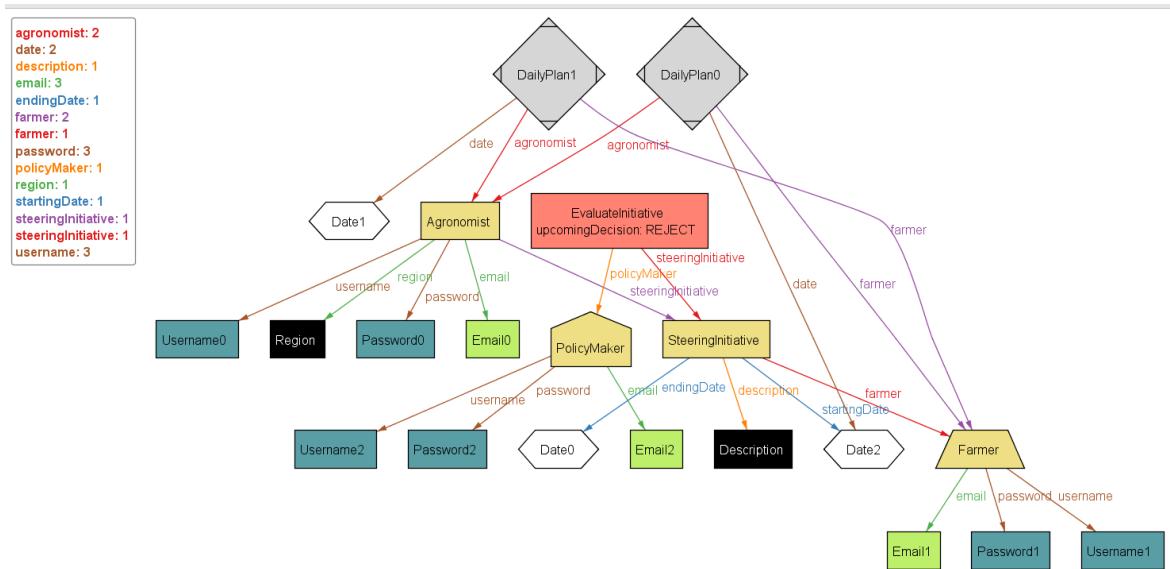
```
No counterexample found. Assertion may be valid. 5ms.
```

## 1.5 Alloy generated Worlds

### Policy Maker Evaluates Initiative

This world models a scenario in which a single Farmer, Agronomist and Policy Maker is registered on DREAM Application. Particularly, this world shows how a Policy Maker evaluates a Steering Initiative made by Policy Maker and adopted by a farmer.

```
pred world1 {
    #Farmer = 1
    #Agronomist = 1
    #PolicyMaker=1
    #EvaluateInitiative=1
}
run world1 for 3
```

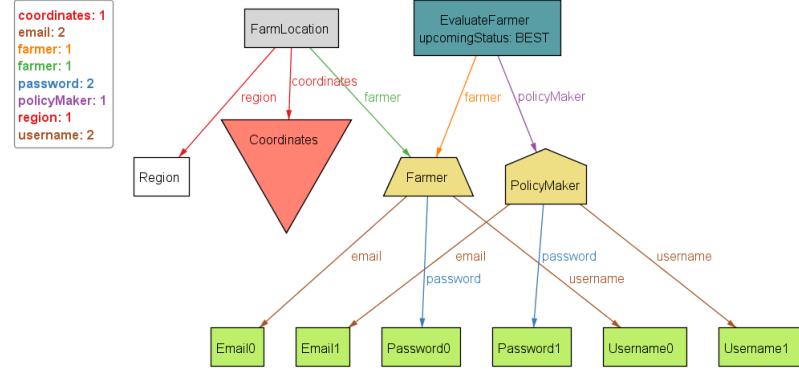


## Policy maker evaluates farmer

This world models a scenario in which is shown how to Policy Maker evaluates Farmer. In the figure, some elements have been hidden for easier reading.

```
pred world2{
    #Notification=0
    #EvaluateFarmer=1
    #HelpRequest=0
    #Discussion=0
    #Production=0
}
```

```
run world2 for 3
```

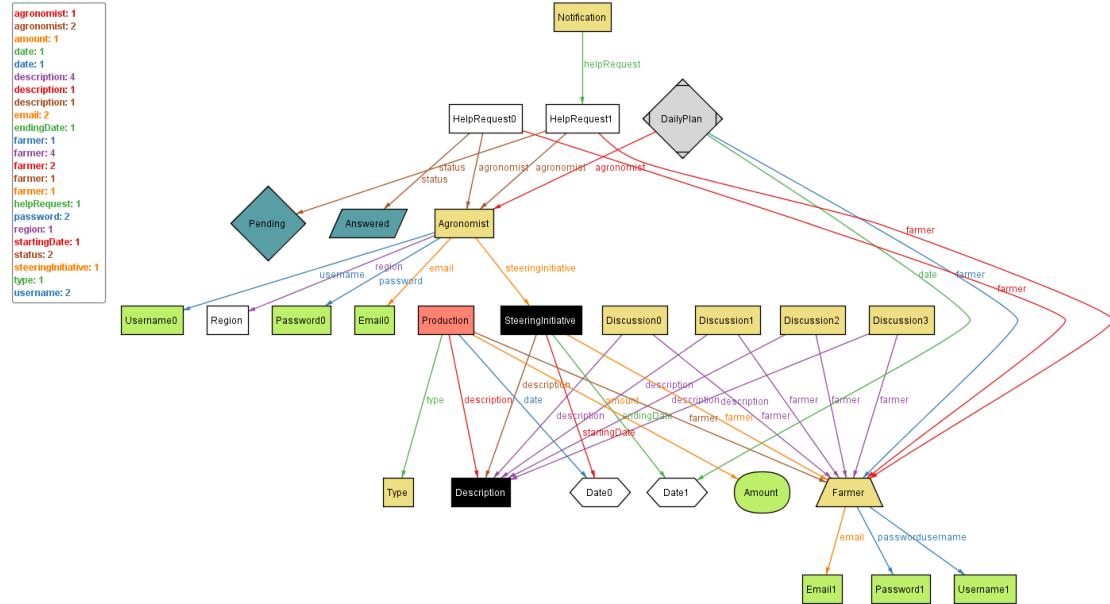


## Agronomist Help request discussion

This world models a more complex scenario with more than one Help Request. It is shown two different types of notification. In fact HelpRequest1 is Pending and is associated a Notification; HelpRequest0 is answered and it's ok.

```
pred AgronomistHelpRequest{
    #Farmer=1
    #Agronomist=1
    #PolicyMaker=0
    #HelpRequest=2
    #Production=1
}
```

```
run AgronomistHelpRequest for 4
```



# Chapter 5

## Effort Spent

### Censuale Simone

Description of the task	Hours
Discussion about the Assignment 2021-2022	1
Document structure and Scope and Purpose	2
First chapter, Definition of Scenarios and Product Functions	3
Actor Scenarios	2
Definition of Use Case and related diagrams	3
Use Case and Sequence diagrams	5
User characteristics and Domain Assumptions	1
User Interfaces	1
UML and State Charts	3
Revision Second chapter	2
Revision Third chapter	3
Alloy: introduction and signatures definition	3
Alloy: facts definition	4

### Giovia Giuseppe

Description of the task	Hours
Discussion about the Assignment 2021-2022	1
Document structure and Scope and Purpose	2
First chapter, Definition of Scenarios and Product Functions	3
Actor Scenarios	1
Definition of Use Case and related diagrams	3
Use Case and Sequence diagrams	5
User characteristics and Domain Assumptions	1
User Interfaces	2
UML and State Charts	3
Revision Second chapter	2
Revision Third chapter	3
Alloy: introduction and signatures definition	4
Alloy: facts definition	3

## Meli Giuseppe

Description of the task	Hours
Discussion about the Assignment 2021-2022	1
Document structure and Scope and Purpose	2
First chapter, Definition of Scenarios and Product Functions	3
Actor Scenarios	2
Definition of Use Case and related diagrams	3
Use Case and Sequence diagrams	5
User characteristics and Domain Assumptions	1
User Interfaces	1
UML and State Charts	3
Revision Second chapter	2
Revision Third chapter	3
Alloy: introduction and signatures definition	3
Alloy: facts definition	4

# **Chapter 6**

## **References**

M. G. Rossi. Lecture Slides. Politecnico di Milano

M. G. Rossi. Project Assignment AY 2020-2021. Politecnico di Milano

ISO/IEC/IEEE 29148:2011. Standard on requirement engineering

<https://standards.ieee.org/standard/29148-2011.html>