POLITECNICO
MILANO 1863

DREAM project Buttiglione M.D.,
Caffagnini M., Faouzi D.

Requirement Analysis and Specification Document

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Marco Domenico Buttiglione, Martina Caffagnini, Dounia Faouzi
Version:	2.0
Date:	6-February-2022
Download page:	https://github.com/marticaffa/CaffagniniFaouziButtiglione.git
Copyright:	Copyright © 2022, Buttiglione M.D., Caffagnini M., Faouzi D. – All rights reserved

Contents

List of Figures

1 Introduction

1.1 Purpose

The current problems caused by climate change and Covid 19 in the Indian agricultural and economic sector have led to the need for renewal of the entire food supply chain. The goal is to create a versatile and resilient system that adopts innovative methodologies.

DREAM wants to acquire and combine data concerning meteorological short-term and long-term forecasts, farmers production, water irrigation, humidity of soil and data provided by the experts in order to help the main stakeholders.

On one hand, the application wants to monitor and determine who are the best farmers to resist to adverse situations and who are the others to be rewarded. On the other hand, it also defines who are the farmers that need help. Finally, the application is useful to understand if the initiatives carried out by the cooperation of agronomists and farmers bring significant results.

DREAM also aims to be a place for farmers where it is possible to view data on agriculture such as weather forecasts and information about the sowing time of specific plants. DREAM is a place to upload data regarding your production or problems and to ask for help and suggestions from the local agronomists and farmers by creating forums for discussion with colleagues.

DREAM sets to give agronomists the chance to share their knowledge and to answer questions from local farmers. The system allows agronomists to view their daily farm visit plans and may possibly change it at the end of the day.

1.1.1 Goals

The goals of this Application are:

G1	Allowing farmer to share their knowledge and problems with other farmers and help them with their farm by creating and commenting a topic
G2	Allowing farmer to keep track of their production
G3	Showing the farmer personalized advice regarding specific subjects of their need, shared by the agronomist
G4	Allowing farmers to make a request for help to agronomists
G5	Allowing agronomists to help farmers in their area by answering to their request
G6	Allowing agronomists to help farmers in their area by sharing their knowledge and giving them advice
G7	Allowing agronomists to plan visits with farmers to check their progress and if needed, help them
G8	Allowing policy makers to monitor the performance of the farmers to help them or to give them special incentives to encourage them to share their experience
G9	Showing customers weather forecasts allowing them to have a better approach with the climate changes

1.2 Scope

The DREAM service wants to acquire and combine data to manage and to keep under control the farmers' production in Telangana with the goal of building a resilient food system.
DREAM should provide the following main features:

- **Request for Help**: allows farmers to send requests for help to their specific agronomist of the area. The local agronomist will reply by giving some advice.

- **Forum**: allows farmers to communicate with other farmers and to share information and problems they face. It is possible to post a topic, to ask a question, to leave a comment, to open conversations.

- **Calendar for Meetings**: allows farmers and agronomists to check their schedule, to modify and to cancel appointments. The agronomist is supposed to visit the farmer at least twice each year to check how the farmer is managing the farm. The agronomist will perform some routine tests such as retrieving a sample of soil to perform composition soil tests. During the first meeting, the agronomist have to install the sensors to obtain soil data.

- **Display Knowledge and Weather Forecasts**: allows farmers to visualize relevant data as weather forecasts, personalized suggestions, such as specific crops to plant or fertilizers to use. The latter is shared by the agronomists considering his studies and experience.

- **Enter Production Data**: after having registered and logged in, this feature gives farmers the possibility to enter the system data concerning their production. They will be able to enter data such as the type of products they cultivate and the correspondent production.

- **Grading System**: The information added by the farmers and the data retrieved by the agronomists will be used to evaluate farmers resilience and how well they are performing. Finally, the grades are shared with the Policy Makers to allow them to understand whether a farmer is performing well and needs to be rewarded or is performing badly and deserves to be helped by the government.

1.2.1 World Phenomena

World Phenomena	Description
WP1	The farmer counts the amount of production
WP2	The agronomist goes to visit farmer's farm
WP3	The agronomist gets soil data of the farmer's farm
WP4	The agronomist installs soil sensors in farms
WP5	The policy maker gives incentives to help farmers

1.2.2 Shared Phenomena

Shared Phenomena	Description	Control
SP1	The farmer creates a forum to share problems or advices	World controlled
SP2	The farmer sees a forum and comments a topic	World controlled
SP3	The farmer enters data concerning the own production	World controlled
SP4	The farmer sends a request for help to the local agronomist	World controlled
SP5	The agronomist plans a meeting with a farmer	World controlled
SP6	The agronomist creates a Knowledge to share with farmers	World controlled
SP7	The application shows to the farmer all topics created by other farmers	Machine controlled
SP8	The application shows to the farmer personalized Knowledge created by the agronomist	Machine controlled
SP9	The application sends a notification to the farmer to confirm a meeting	Machine controlled
SP10	The application shows to the agronomist all the requests of the farmers	Machine controlled
SP11	The application sends a notification to the agronomist to remind them of the meeting	Machine controlled
SP12	The application sends a notification to the agronomist about a new farmer in their area to plan the first meeting	Machine controlled
SP13	The application shows to the policy maker the performance of the farmers	Machine controlled
SP14	The application shows to the customer the weather forecasts	Machine controlled

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- **Farmer**: the "average" customer of the application that shares their progress and visualizes data relevant to him;
- **Agronomist**: the customer that receives requests for help and answers them, visualizes data concerning weather forecasts and farmers in the area and organizes meetings with farmers;
- **Policy Maker**: the customer that visualizes data uploaded by farmers and agronomists;
- **Customer**: general DREAM customer, can be a farmer or an agronomist or a policy maker;

- **Notification**: a message shown to the customer when he needs to be notified about something;
- **Request**: request for help made by a farmer to a single agronomist;
- **Topic**: anything that farmers want to share with other farmers. They can either ask for help or share something they know that might help others;
- **Knowledge**: information shared by the agronomists. It has the goal to inform the farmers and to increase their awareness to improve their performance;

1.3.2 Acronyms

Acronyms	Description
RASD	Requirements Analysis and Specification Document
UI	User interface
DREAM	Data-dRiven prEdictive fArMing

1.3.3 Abbreviations

Abbreviation	Description
WP	World Phenomena
SP	Shared Phenomena
PM	Policy Maker
Gn	n-th Goal
Rn	n-th Requirement
Dn	n-th Domain Assumption
Un	n-th Use Case

1.4 Revision history

Date	Description
01/11/2021	First Version and goal definitions.
07/11/2021	Purpose and Definitions.
10/11/2021	Scope definition and adding acronyms and abbreviations.
13/11/2021	Software, Hardware and Communication interfaces definitions.
17/11/2021	UML class, words and shared phenomena definitions.
20/11/2021	Domain assumptions, User characteristics.
23/11/2021	Scenarios definitions.
25/11/2021	Product perspective and functions.
30/11/2021	Adding Use Cases, Sequence diagram and their definitions.
04/11/2021	Adding state charts
05/12/2021	Document structure definition.
06/12/2021	Last version of requirements definitions.
10/12/2021	Mapping between goals, requirements, and domain assumptions.
13/12/2021	Use cases mapping.
15/12/2021	Performance Requirements, Design Constraints, and Software System Attributes.
18/12/2021	Alloy integration.
20/12/2021	Final corrections and improvements.
22/12/2021	First release
06/02/2021	Second release: Scope, Software System Attributes

1.5 Reference Documents

- Slides of the course Software Engineering 2
- Requirement Engineering and Design Project: goal, schedule, and rules

1.6 Document Structure

The document is composed of the following chapters:

Chapter 1: In this chapter the purpose and the scope of the software are introduced. At fist there will be a general description of the system and the main goals. After, there are specified the main features the system should provide and then follows the analysis of word and shared phenomena. Lastly, there are shown the list of abbreviations, acronyms, definitions used in the document, the revision history and the reference documents.

Chapter 2: This chapter presents the most relevant functions of the system and a representation with a Class Diagram and some State Charts. Lastly, there is the clarification of the different categories of users of the system, the list of the domain assumptions, and some examples of possible scenarios.

Chapter 3: The chapter describes the interfaces of the system and specifies its functional requirements. Use Cases Diagrams, Sequence Diagrams, mapping between requirements, domain assumptions, and use cases are also presented.

Chapter 4: In this chapter is included the metamodel generated with Alloy and the code used to achieve this goal.

Chapter 5: The chapter shows the amount of time that each member has spent to write the document.

Chapter 6: The last chapter shows the references and online resources used for this document.

2 Overall Description

2.1 Product perspective

The DREAMS service is offered to the customers that want to acquire and to combine data concerning meteorological short-term and long-term forecasts, information provided by the farmers about their production, information obtained by the water irrigation system concerning each farmer, information about the humidity of the soil and information regarding the soil's composition obtained by the governmental agronomists from each farm. The service is thought to put together this information aiming to strengthen data-driven policy making in Telangana.
In the following list, a summary of the problem resolution process is presented.

- After having registered and logged in, the DREAM service gives farmers the possibility to enter the system data concerning their production and possible problems they face. They can enter data such as the type of products they cultivate and the correspondent production. The data are obtained by smart sensors. There is one smart humidity sensor to track the soil humidity, the temperature and the amount of sunlight. Another is a smart water meter. These smart sensors send data to the agronomists.

- Farmers visualize relevant data such as weather forecasts, personalized suggestions ; which specific crops to plant or which specific fertilizers to use.

- Farmers log in and create discussion forums with other farmers to exchange advice concerning their activities or to ask directly to their corresponding agronomist.

- Every time a farmer asks some advice to his agronomist, he is given the possibility to give feedback telling if the advice has been useful and if it has been applied. This feedback system is used not only to verify if the farmer in need has been able to work through his problem showing resilience, but also to show if the agronomist answering the questions have been useful. The latter will be used in the grading system concerning the farmers.

- Since the farmers need to be visited at least twice a year by the agronomists, they will have access to a calendar through which they can accept the appointment requested by the agronomist and have a view of the upcoming appointments.

- Agronomists, on the other hand, will insert data concerning the area they are responsible for, will receive information about requests for help and will answer to these requests. They will have access to a map with all farmers under their examination and will visit each farmer at least twice a year by requesting a visit through a calendar available in their platform. Through such a calendar the agronomist can request, modify, and confirm appointments with the farmers.

- During each visit the agronomist will check on the farmer and will acquire data concerning the soil composition by performing a test to evaluate its PH, phosphorus, total limestone, total nitrogen and organic carbon composition to have information about it in order to give the farmers proper advice.

- Agronomists can also visualize data concerning weather forecasts in the area they are responsible for in order to give the farmers coherent suggestions and advice in case they are asked.

- Agronomists can also visualize the best performing farmers in their area. Each farmer is graded every six months to evaluate his performance and his resilience. The grade the Agronomist gives to the Farmer is based on the farmer's production report, the amount of water they have used, how well they demonstrate to be resilient to meteorological adverse events and how well they show to be transparent towards other farmers by sharing their acquired knowledge through the forum. This information will be also redirected to Telangana's policy makers in order to highlight the best performing farmers that will be eligible to receive special incentives and will be asked to share their best practice to the other farmers. The grading system will also help Telangana's policy makers to find these farmers who need help as they are performing badly and to understand whether the steering initiatives carried out by agronomists with the help of good farmers produce significant results.

2.1.1 Scenarios

Scenario 1 : Marco is a farmer and an usual user of DREAM, while scrolling though different topic, finds a subject that his well informed at. He decide to comment on the topic sharing his knowledge of the topic to help other farmers. This comments during the day get lot of likes and increased Marco position in the list of best resilient farmers.

Scenario 2 : Martina is a farmer, and after a day of collecting crop decides to update the data of her production of the day. In that way she will be able to see the progress of her activity.

Scenario 3 : Virginia is a farmer that earlier in the day finds an infestation of bugs in her farm. She then decides to ask for help to her local agronomist for some advice to how to get rid of them. The agronomist answers to her request by suggesting an insecticide. Virgina tries the agronomist's advice and successfully gets rid of the bugs, then leaves a feedback agronomist's feedback.

Scenario 4 : Matteo is an old farmer, that with the help of his son wants to try to be more self aware of the weather forecasts and trying to have a better approach with the climate changes. His son controls once a day the weather and the knowledge shared by the agronomist and refers them to Matteo.

Scenario 5 : Pietro is a farmer that in a conversation with another farmer gets to know about DREAM, so he decides to register in the application. He then enters his personal data and the data of his farm. The application asks Pietro if he already has some collected data of his production, but Pietro prefers to skip that part and starts scrolling though his advice.

Scenario 6 : Elisabetta is an Agronomist responsible for a determined location who received the notification about a new farmer in her controlled area. She then decides to plan the first meeting with the farmer to add their soil data. She then opens her monthly plan and looks for a free day, selecting the day and the hour she creates the meeting. The application sends a notification to the farmer to confirm the meeting.

Scenario 7 : Carlo is an Agronomist that did some research about a natural way to create an insecticide for bugs. After his research and attempts, he then decide to share his new knowledge with all the farmers adding tags to identify the subject, for example insecticide, natural and bugs.

Scenario 8 : Francesco is a agronomist who, later that day, has a first meeting with a farmer for the first time. The application in the morning sent him a notification to remind him of the meeting. In the evening he went to visit the farmer and during the visit he collects the data about his soil, installs the sensors and gets information about the farmer resilience. After the meeting, Francesco updates the information about the farm and then lets a feedback of the meeting.

Scenario 9 : Dounia is a policy maker that after logging in sees the farmers performance, that is showed by the application after making some computations. She then searches for farmer of a specific location and order them by resilience. After selecting the first farmer she looks at his information and decides to contact him to give him a reward for his work.

2.1.2 Class Diagram

The UML class diagram below depicts the conceptual model of the program that will be developed. Because of its nature, it is possible that it will model items that will not be represented in the final system. Note that the policy maker's table is not related to any other table because the policy maker's task is only to have an overview of the system while he does not directly engage with neither agronomists or farmers. Still, it is important to insert the table in the diagram since their data are necessary to their login into the system.

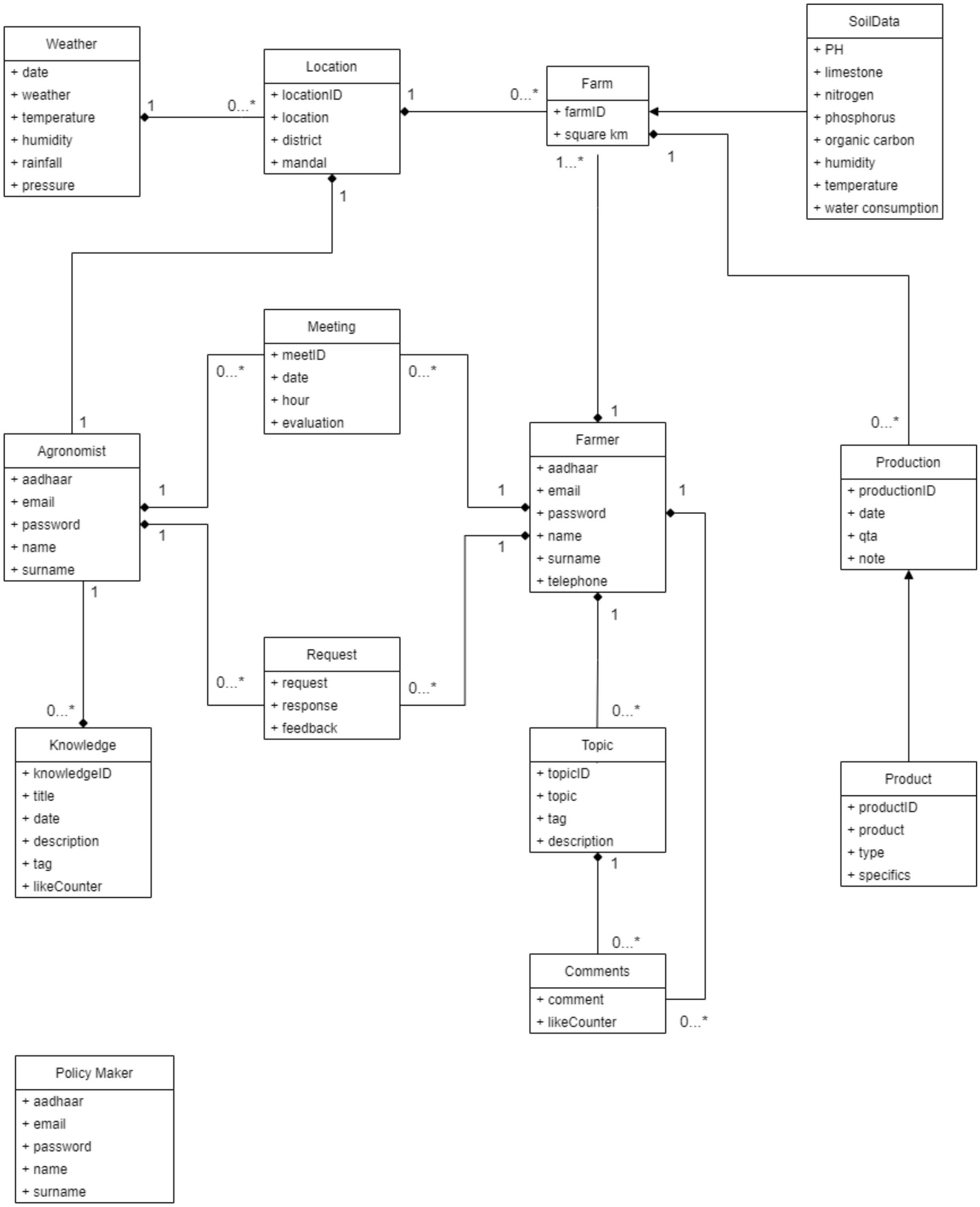


Figure 1: UML Class Diagram

2.1.3 State Charts

In this section the proposed UML State Charts will present the behavior of the above classes in terms of state (state machine), how the main conceptual components of the system respond to external events.

Meeting

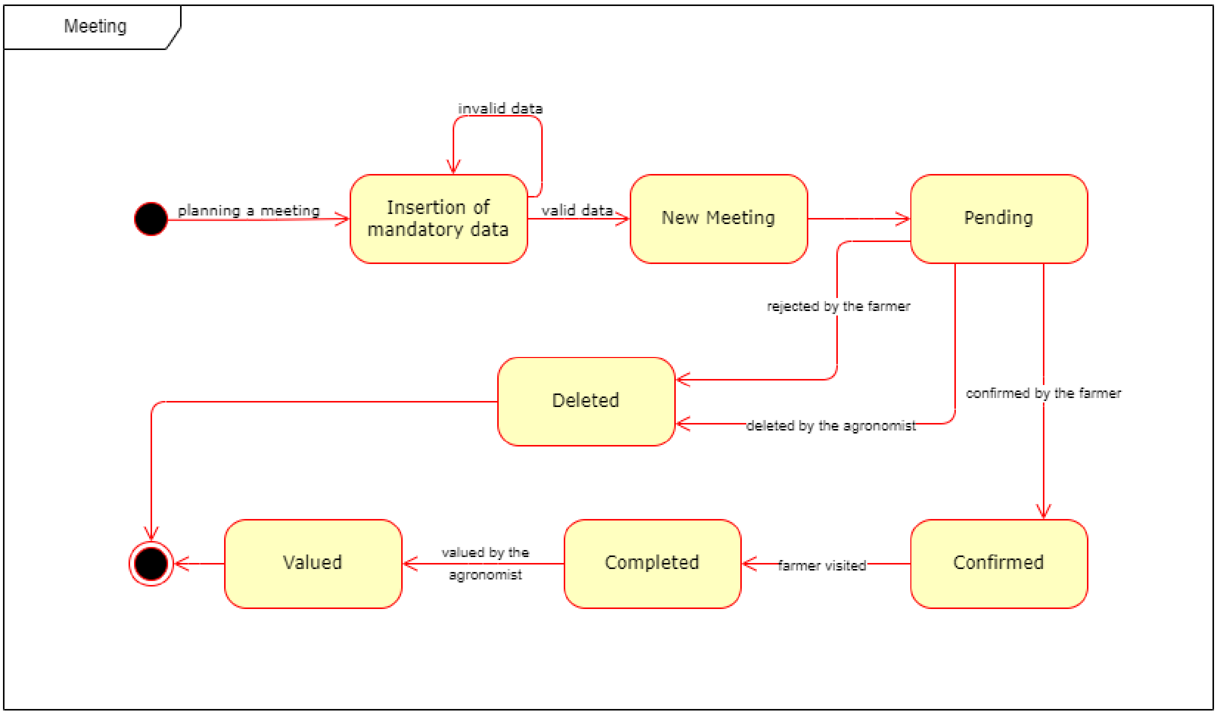


Figure 2: State Chart – Meeting

The State Chart above shows the steps and states of planning a meeting. They are:

- *Insertion of mandatory data*: the agronomist is organizing a new meeting and starts filling out the form by selecting the farmer to visit and entering the mandatory data such as date and time. A check of the correct data entry is also carried out;
- *New Meeting*: if the data is entered correctly then a new meeting has been created;
- *Pending*: in this state the meeting is waiting for an action by the farmer or the agronomist;
- *Confirmed*: if the farmer has confirmed the meeting then he is ready to be visited by the agronomist;
- *Completed*: the agronomist visited the farmer;
- *Valued*: the agronomist evaluates the farmer's work;
- *Deleted*: if the farmer rejects the meeting or the agronomist deletes it then it is removed from the calendar.

Help Request

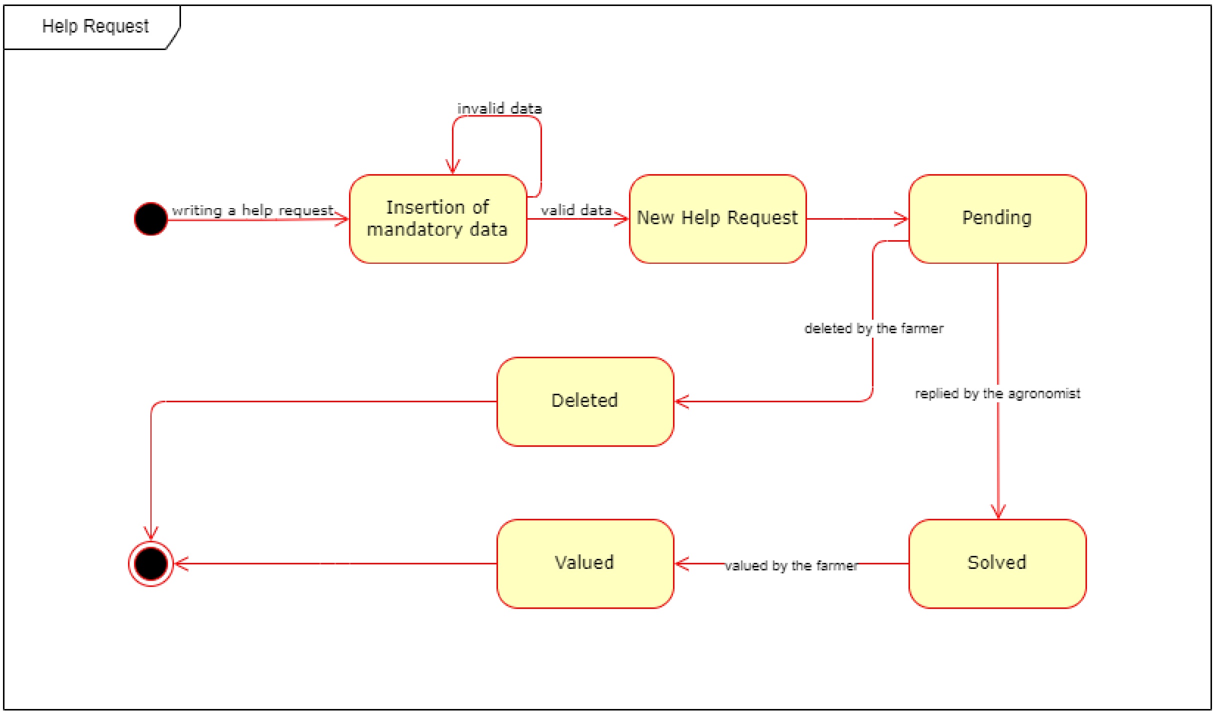


Figure 3: State Chart – Help Request

The State Chart above shows the steps and states of creating a help request. They are:

- *Insertion of mandatory data*: the farmer is writing a help request and starts filling out the form by entering the mandatory data. A check of the correct data entry is also carried out;
- *New Help Request*: if the data is entered correctly then a new help request has been created;
- *Pending*: in this state the help request is waiting for an action by the farmer or the agronomist;
- *Solved*: the agronomist has responded to the help request;
- *Valued*: the farmer gives feedback to the agronomist's response;
- *Deleted*: the farmer eliminates the help request.

2.2 Product functions

As previously described in this document, DREAM offers a system that helps the government of Telangana to have a view over the situation of their farmers. In the following, a list of the most basic and advanced functions is presented.

- **Request for Help.** The system allows to:
  - send a help request to the agronomist;
  - answer a help request made by a farmer;
  - show a list with all the help requests;
- **Forum.** The system allows to:
  - create a new topic to the farmer;
  - add tags to the topic;
  - answer to a topic to a farmer;
  - show a list with all the topics to all farmers;
- **Calendar.** The system allows to:
  - show a calendar with planned appointment to farmers and agronomists;
  - add a new appointment to the agronomist;
  - modify a scheduled appointment to the agronomist;
  - cancel a scheduled appointment to the agronomist;
- **Display Knowledge and Weather Forecasts.** The system allows to:
  - create a new knowledge to the agronomist;
  - add tags to the knowledge to the agronomist;
  - show a list with all the knowledges to farmers and agronomists;
  - view the weather forecasts to farmers and agronomists;
- **Insert Production Data.** The system allows to:
  - insert data for a specific production to the farmer;
  - save the inserted data to the farmer;
  - insert a new product to then add the amount produced;
- **Grading System.** The system allows to:
  - evaluate the farmer;
  - share the given grade with the policy Makers;

Requirement	Description
R.1	The system allows the farmer to add data concerning his production
R.2	The system allows the agronomist to view the data concerning their farmers' production
R.3	The system allows the agronomist to book an appointment with the farmer by showing all available slots
R.4	The system allows the agronomist to modify a booked appointment with the farmer
R.5	The system allows the agronomist to cancel a booked appointment with a farmer
R.6	The system check that the agronomist booked at least two appointments per year with each farmer
R.7	The system alerts the farmer before the appointment
R.8	The system alerts the agronomist before the appointment
R.9	The system allows the agronomist to view the calendar
R.10	The system allows the farmer to view the calendar
R.11	The system allows the farmer to send a help request to the agronomist
R.12	The system allows the agronomist to answer to the help requests that are sent to him
R.13	The system allows the agronomist to view all the help requests sent to him
R.14	The system allows the farmer to view all the help requests sent by him
R.15	The system allows the farmer to publish a topic on the forum
R.16	The system allows all farmers to view the forum
R.17	The system allows all farmers to publish an answer to a topic on the forum
R.18	The system allows the farmers to check the weather forecasts for their area
R.19	The system allows the agronomists to share knowledge with the farmer
R.20	The system allows the farmers to visualize the knowledges shared by their agronomist
R.21	The system allows all agronomists to check the weather forecasts for their area
R.22	The system allows the agronomists to publish a grade for each farmer after each visit
R.23	The system allows the policy makers to view the grades of all farmers of Telangana
R.24	The system requires a sign up and a login to access to the data
R.25	The system shows to each agronomist the list of all farmers they are responsible for
R.26	The system notifies the agronomists when a new farmer has registered under their area
R.27	The system allows the farmer to insert their position when they first register

2.3 User characteristics

The actors of the application are the following:

- **Farmer**: the farmer is a person who has a farm and is registered to DREAM. The farmer is a person who owns a farm and provides his data by explicitly writing them or indirectly by sensors that are placed around the farm that collect the data and send them to the system. The user can see his data on the application. The farmer provides data concerning his production and gives access to the system to acquire data concerning the soil's composition, humidity, PH and water consumption. The farmer can take part in forums with other farmers to exchange infos, can ask suggestions to agronomists, can look up the upcoming weather forecasts and can check the upcoming appointments with the agronomist through a calendar.

- **Agronomist**: the agronomist is a person working for the Talengana's government having access to DREAM. The agronomist is in charge of a certain area of Telangana and his job is to check on the farmers periodically (at least twice a year). The agronomist has access to a map with all the farmers they are responsible for. The agronomist has access to the farmer's information and can help them by answering their help requests. The agronomist has access to a calendar to plan, modify and confirm appointments with farmers. The agronomist grades the farmer based on the farmer's results and resilience.

- **Talengana's policy maker**: the Telangana's policy maker is a person working for the Talengana's government and has access to all information retrieved by the agronomists and to the ranking of farmers. They decide, based on the farmers' ranking, the farmers that need to be helped because they are performing badly and those that need to be rewarded because they are performing well.

2.4 Assumptions, dependencies and constraints

2.4.1 Domain Assumptions

The Domain Assumptions for this Application are:

D1	The internet connection works properly without failure
D2	The agronomist places the sensors during the first meeting with the farmer
D3	The tags of a topic must be coherent to it
D4	When a customer shares something on the platform the language must be adequate and unoffensive
D5	When a farmer updates his data production he must insert reliable and truthful data
D6	The Knowledge shared by agronomists must be true
D7	The tags of a Knowledge must be coherent to it
D8	The help requests sent by a farmer must be about farm related topics
D9	When a customer answers something on the platform, the answer must be related to the question
D10	The meeting schedule must be followed unless previously canceled either by the farmer or the agronomist
D11	After each meeting the agronomist must complete the farmer's evaluation
D12	At least two meetings a year per farmer must be performed
D13	After each meeting the agronomist must perform the soil composition's test and update the results on the corresponding farmer's platform
D14	The sensors that acquire the soil composition data provide information with an error of at most 10-15 percentage (for each type of data) compared to the real values
D15	The sensors that acquire the humidity data provide information with an error of at most 4 percentage compared to the real value
D16	The sensors that acquire the water consumption data provide information with an error of at most 5 percentage compared to the real value
D17	Each Aadhaar is unique
D18	Each farmer that is registered to the system has the sensors correctly placed at all times

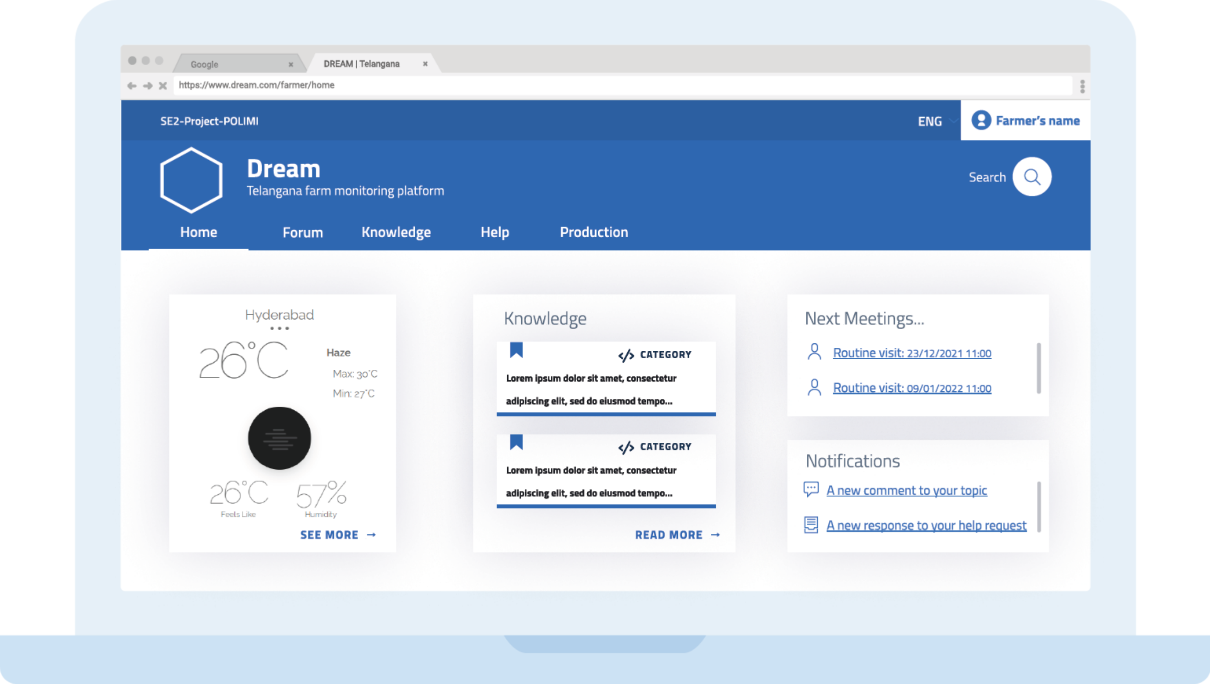
The DREAM system is thought to be distributed in India and that is the reason the system uses the Aadhaar, a 12-digit code each Indian resident has, the Indian Social Security Number, to identify and distinguish the farmers registered in the system. Lastly, the system needs to be compliant with the normative for the privacy of the country.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User interfaces

The system allows the three customers to interact with it in different ways such as creating meetings, looking at the calendar, observing the weather forecast, creating forums and requests for help and much more. For this reason it is highly recommended to use a device with a large screen to view all this information. So a web app turns out to be the best option for all customers to access the platform being able to run on almost all devices with a web browser.
The UI must therefore consist of a landing page where it will be possible to access the main functions of the system after login. Overall it will have to be user friendly and with a unified theme, due possibly not expert users, with large buttons and tabs organized by functionality. The display of the contents must follow a similar view, with primary geometric shapes such as square and circle to represent the interfaces, large space left to the texts, using the shadows that allow the elements of the interface to be hierarchically structured, one on top of the other, and focuses his attention on the important elements of the page such as buttons or navigation menus.



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempo...

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempo...

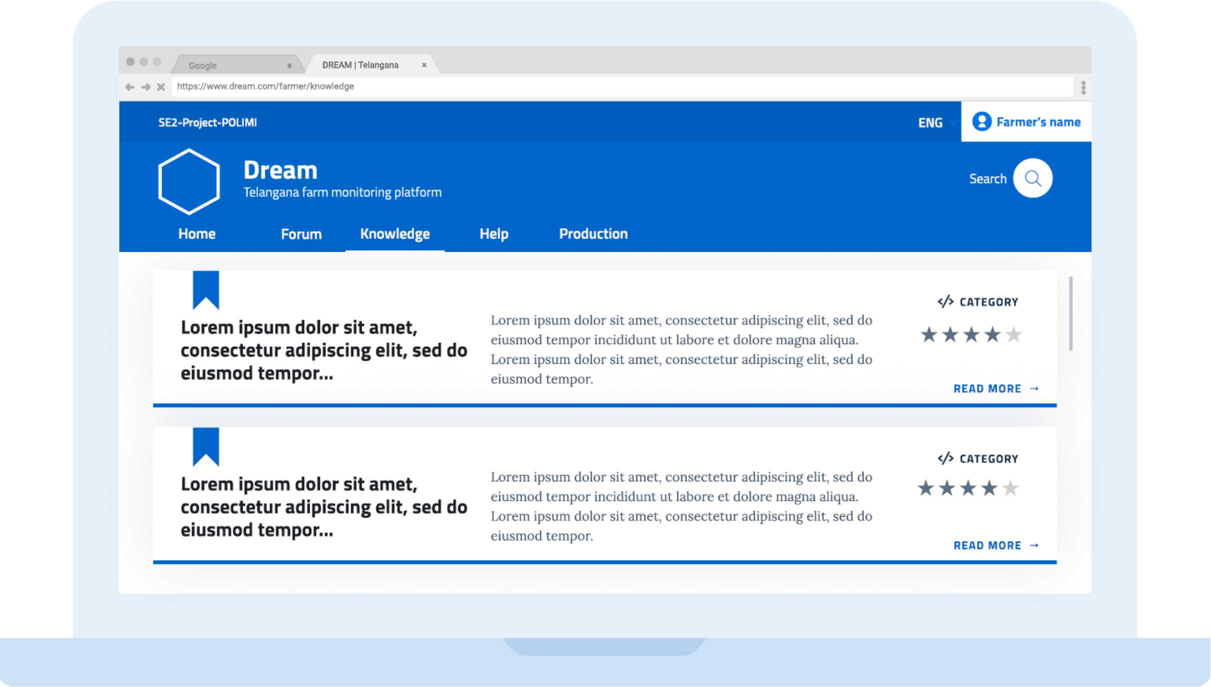
Routine visit: 23/12/2021 11:00

Routine visit: 09/01/2022 11:00

A new comment to your topic

A new response to your help request

Figure 4: Farmer – Homepage



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor...

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor.

★★★★☆

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor...

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor.

★★★★☆

Figure 5: Farmer – Knowledge

3.1.2 Hardware interfaces

To use the system, all users need to have a device where they can download a web browser and carry out all the tasks (ex. signing up, logging in, inserting data, chatting on the forum, asking questions, answering questions, grading farmers...). Due to the presence of numerous elements on the screen, a tablet or laptop turns is considered to be the recommended choice since it would improve the user experience. In order to exploit those features, the users need a device with a 3G/4G/5G or WiFi connection.
Some services offered by the system require some specific hardware to be installed around the farm. Each farmers needs to have sensors for the detection of data concerning the soil (soil moisture sensors) and for the detection of water consumption.

3.1.3 Software interfaces

In order to offer information regarding weather forecasts, the system needs API interfaces offered by the local forecast service. It is used by agronomists and farmers to obtain precise weather information based on the work area of the two users. This API is also used to detect adverse weather phenomena that are reported to agronomists and farmers.
The sensors for the acquisition of data concerning the soil and the water consumption need to be connected with the rest of the system with some API.

3.1.4 Communication interfaces

The system requires a constant connection to the internet. It is essential for the correct functioning of the application, for the reception of data from the weather forecast system and the population of the tables. It is also necessary for communications between farmers and agronomists and viewing calendars.
The supported sensor set also need internet communication to interact with the system.

3.2 Functional Requirements

3.2.1 Use cases Diagram

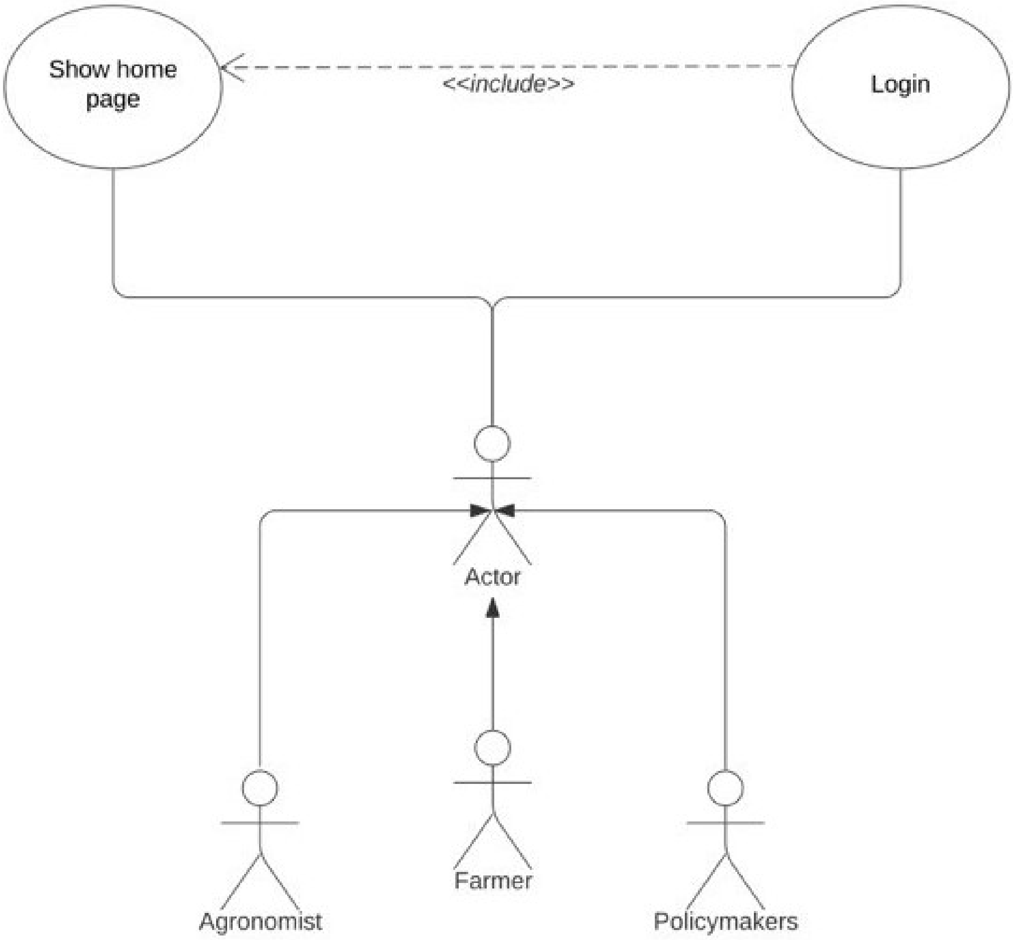


Figure 6: Use cases Diagram (a)

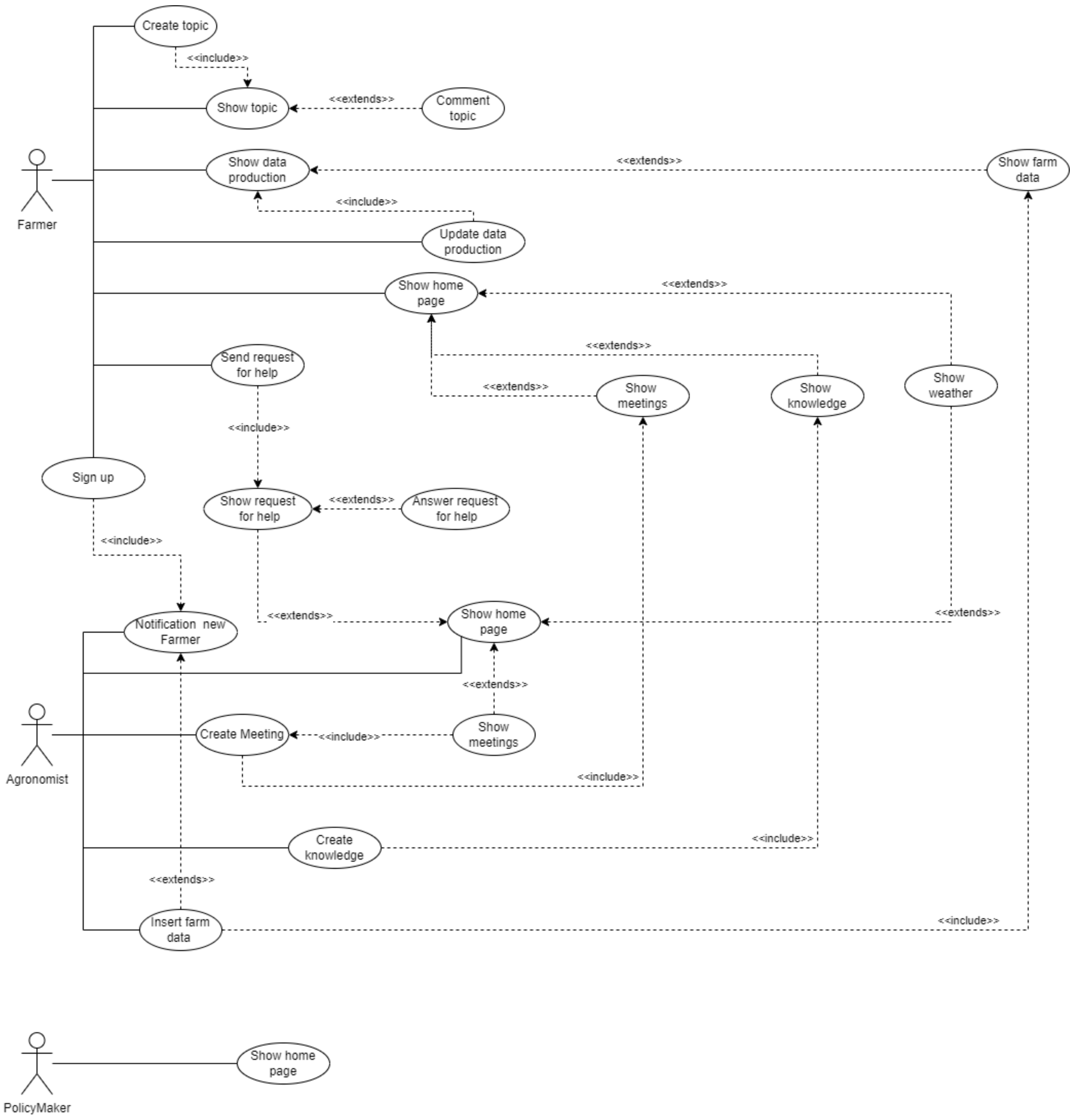


Figure 7: Use cases Diagram (b)

3.2.2 Use cases

1. Show Farmer's Home Page

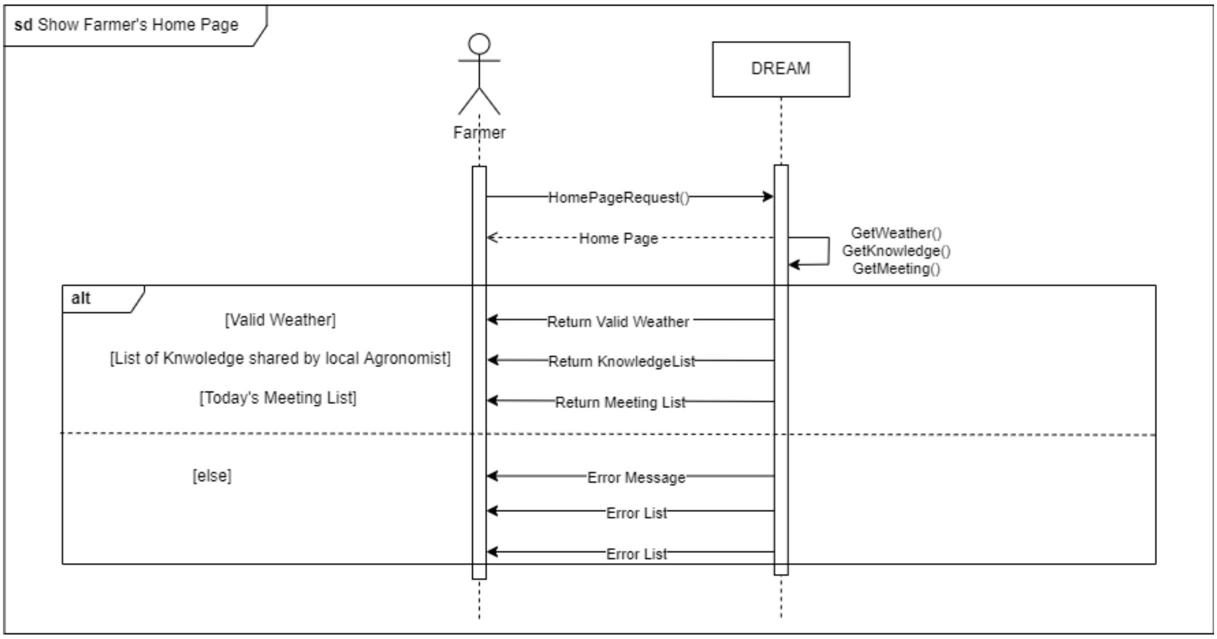


Figure 8: Sequence Diagram - Show Farmer's Home Page

Name	Show Farmer's Home Page
Actors	Farmer
Entry Condition	The actor is already logged into the system
Event Flow	1. The actor requires the Home Page 2. The system shows the Home Page to the actor 3. The system retrieves today's weather, local Agronomist's knowledge list and Farmer's meetings 4. The system displays the weather, knowledge and meetings
Exit Condition	The Home Page with weather, knowledge and meeting is displayed
Exception	1. Loss of internet connection 2. The actor cancels the operation 3. Problem with connection to the weather API 4. The list of Knowledge is empty 5. The list of meetings is empty

2. Show Agronomist’s Home Page

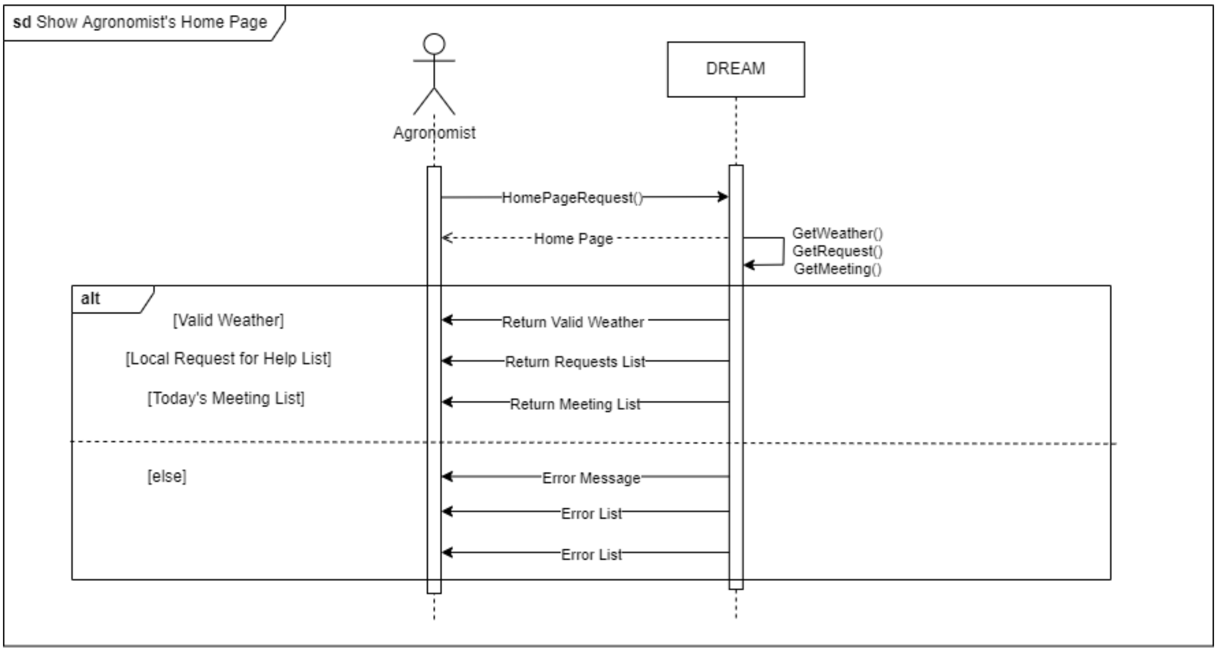


Figure 9: Sequence Diagram - Show Agronomist’s Home Page

Name	Show Agronomist’s Home Page
Actors	Agronomist
Entry Condition	The actor is already logged into the system
Event Flow	1. The actor requires the Home Page 2. The system shows the Home Page to the actor 3. The system retrieves today’s weather, requests for help from local Farmer and today’s Meetings 4. The system displays weather, requests and meetings
Exit Condition	The Home Page with weather, request and meeting is displayed
Exception	1. Loss of internet connection 2. The actor cancels the operation 3. Problem with connection to the weather API 4. The list of Request is empty 5. The list of meetings is empty

3. Show Topic Page

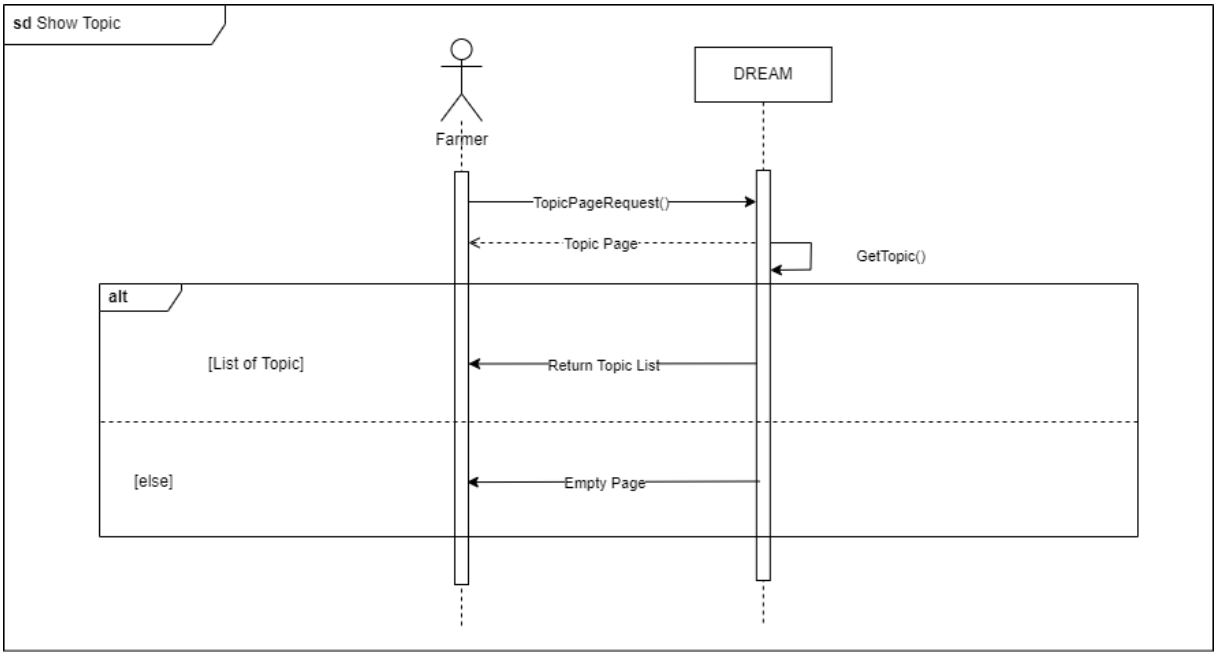


Figure 10: Sequence Diagram - Show Topic Page

Name	Show Topic
Actors	Farmer
Entry Condition	The actor is already logged into the system
Event Flow	1. The actor requires the Topic Page 2. The system shows the Topic Page to the actor 3. The system retrieves Topic list 4. The system displays the topics
Exit Condition	The Topic Page is displayed
Exception	1. Loss of internet connection 2. The actor cancels the operation 3. The list of Topic is empty

4. Show Production

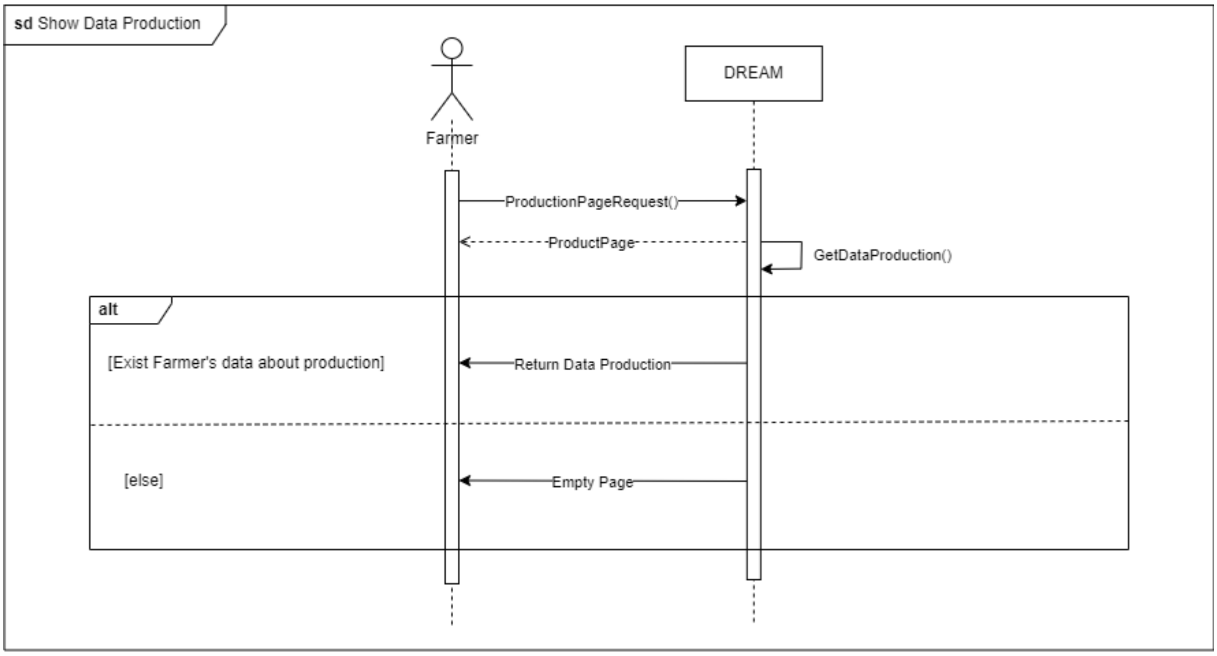


Figure 11: Sequence Diagram - Show Production

Name	Show Production Page
Actors	Farmer
Entry Condition	The actor is already logged into the system
Event Flow	1. The actor requires the Production Page 2. The system shows the Production Page to the actor 3. The system retrieves data about production 4. The system displays the data
Exit Condition	The Production Page is displayed
Exception	1. Loss of internet connection 2. The actor cancels the operation 3. There is no data about production

5. Show Farm Info

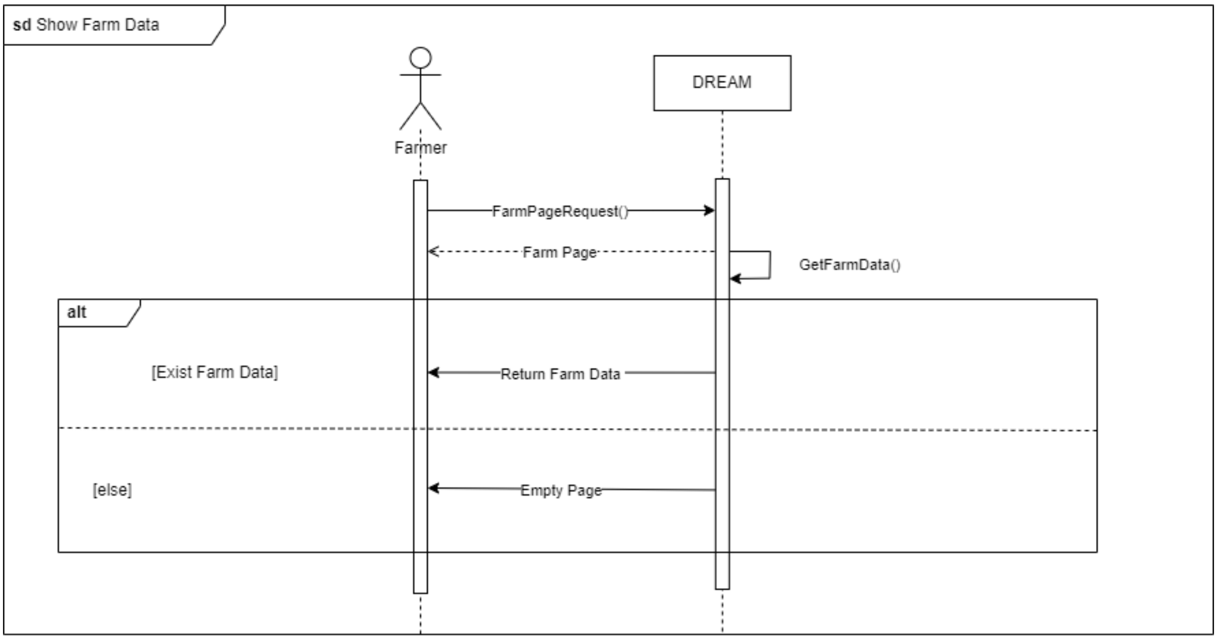


Figure 12: Sequence Diagram - Show Farm Info

Name	Show Farm Info Page
Actors	Farmer
Entry Condition	The actor is already logged into the system
Event Flow	1. The actor requires the Farm Info Page 2. The system shows the Farm Info Page to the actor 3. The system retrieves data about farm 4. The system displays the data
Exit Condition	The Farm Info Page is displayed
Exception	1. Loss of internet connection 2. The actor cancels the operation 3. There is no data about the farm

6. Send Request for Help

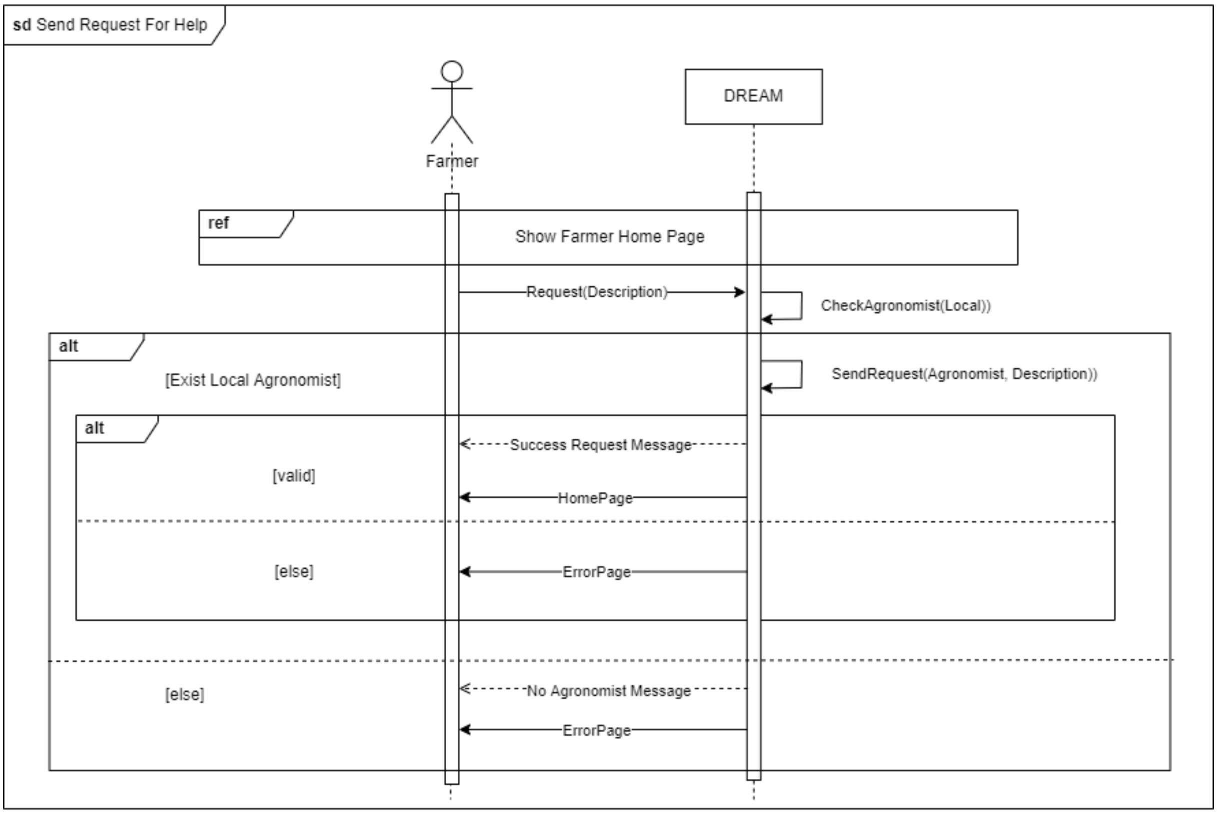


Figure 13: Sequence Diagram - Send Request for Help

Name	Send Request for Help
Actors	Farmer
Entry Condition	The actor is already logged into the system
Event Flow	1. The actor requires the Help Page 2. The system shows the Help Page to the actor 3. The actor require Help from Agronomist 4. The system retrieves data about local Agronomist 5. The system send the request to the local Agronomist 6. The system displays a success Message 7. The system displays Home Page
Exit Condition	The Request is sent to the Agronomist
Exception	1. Loss of internet connection 2. The actor cancels the operation 3. There is no Local Agronomist

7. Sign Up

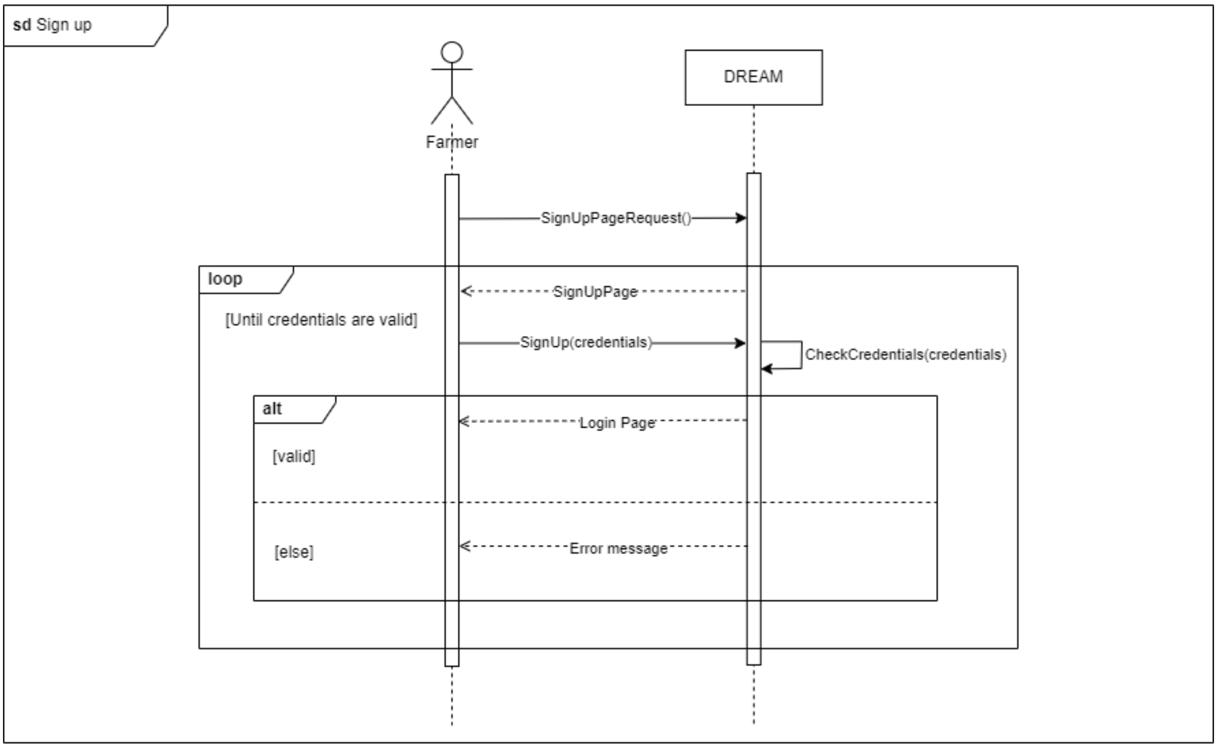


Figure 14: Sequence Diagram - Sign Up

Name	Sign Up
Actors	Farmer
Entry Condition	The actor is not already registered into the system
Event Flow	1. The actor requires the Sign Up Page 2. The system shows the Sign Up Page to the actor 3. The actor inserts all the credentials 4. The system checks the credentials 5. The system saves the credentials 6. The system shows the Login Page at the actor
Exit Condition	The actor is successfully registered into the system and the Login Page is shown
Exception	1. Any mandatory fields are not filled 2. Password and Confirm Password fields are different 3. The actor is already present into the system

8. Login

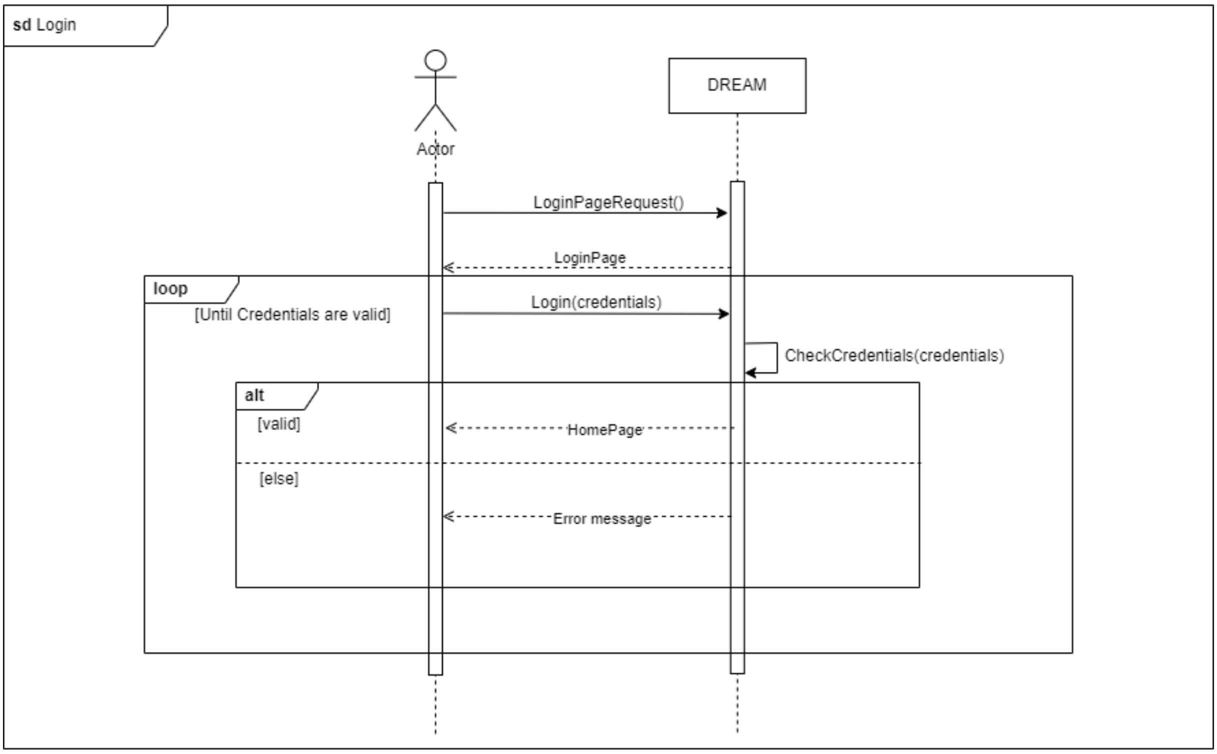


Figure 15: Sequence Diagram - Login

Name	Login Page
Actors	Farmer, Agronomist, Policy Maker
Entry Condition	The actor is already registered in the system
Event Flow	1. The actor requires the Login Page 2. The system shows the Login Page to the actor 3. The actor inserts and sends the credentials 4. The system checks the information 5. The system shows the Home Page to the actor
Exit Condition	The actor has successfully logged into the system and the Home Page is displayed
Exception	1. Th username is not valid 2. The password is not valid 3. Loss of internet connection 4. The actor cancels the operation before confirming

9. Create Meeting

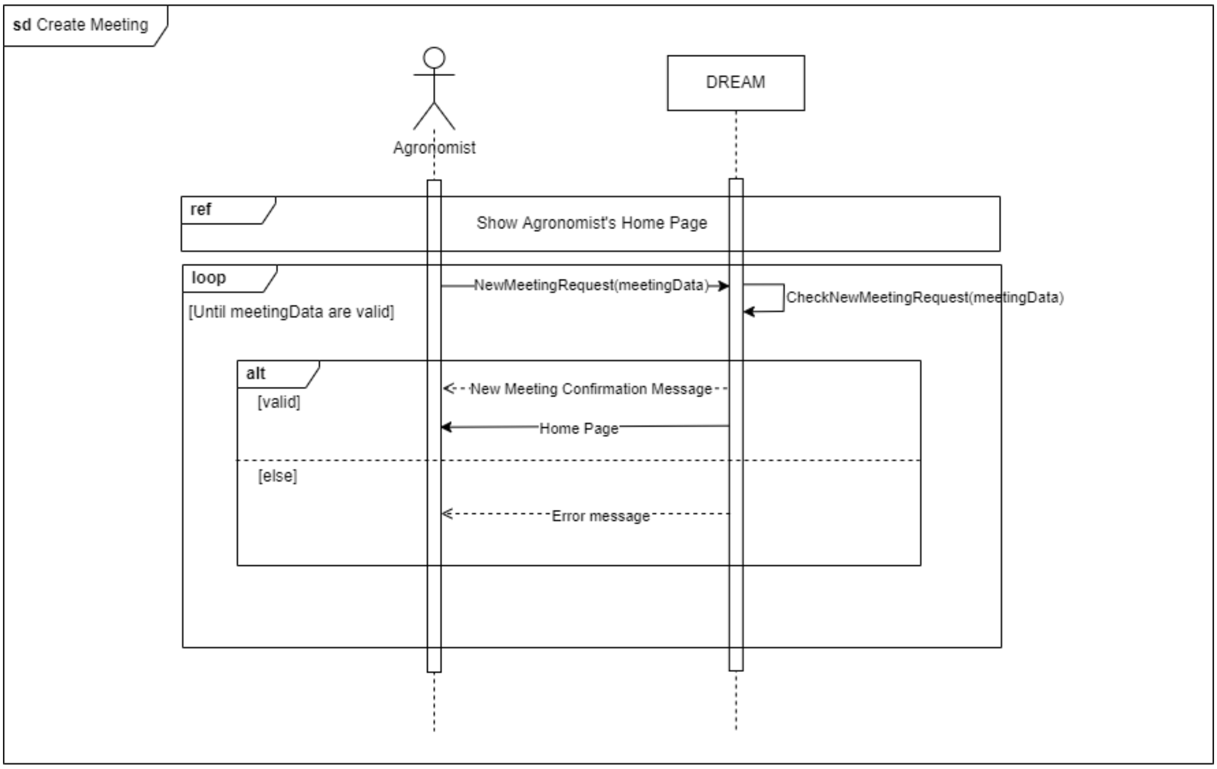

sd Create Meeting
Agronomist
DREAM
ref
Show Agronomist's Home Page
loop
[Until meetingData are valid]
NewMeetingRequest(meetingData)
CheckNewMeetingRequest(meetingData)
alt
[valid]
New Meeting Confirmation Message
Home Page
[else]
Error message

Figure 16: Sequence Diagram - Create Meeting

Name	Create Meeting
Actors	Agronomist
Entry Condition	The actor is already logged into the system and he is in the Home Page
Event Flow	1. The actor inserts the Meeting Data 2. The system checks the Meeting Data 3. The system saves the Meeting Data 4. The system displays a success Message 5. The system displays Home Page
Exit Condition	The Meeting is successfully saved into the system and the Home Page is shown
Exception	1. Any mandatory fields are not filled 2. The Meeting Date is in the past 3. The actor has already another Meeting in the same period 4. The Farmer has already another Meeting in the same period

10. Create Knowledge

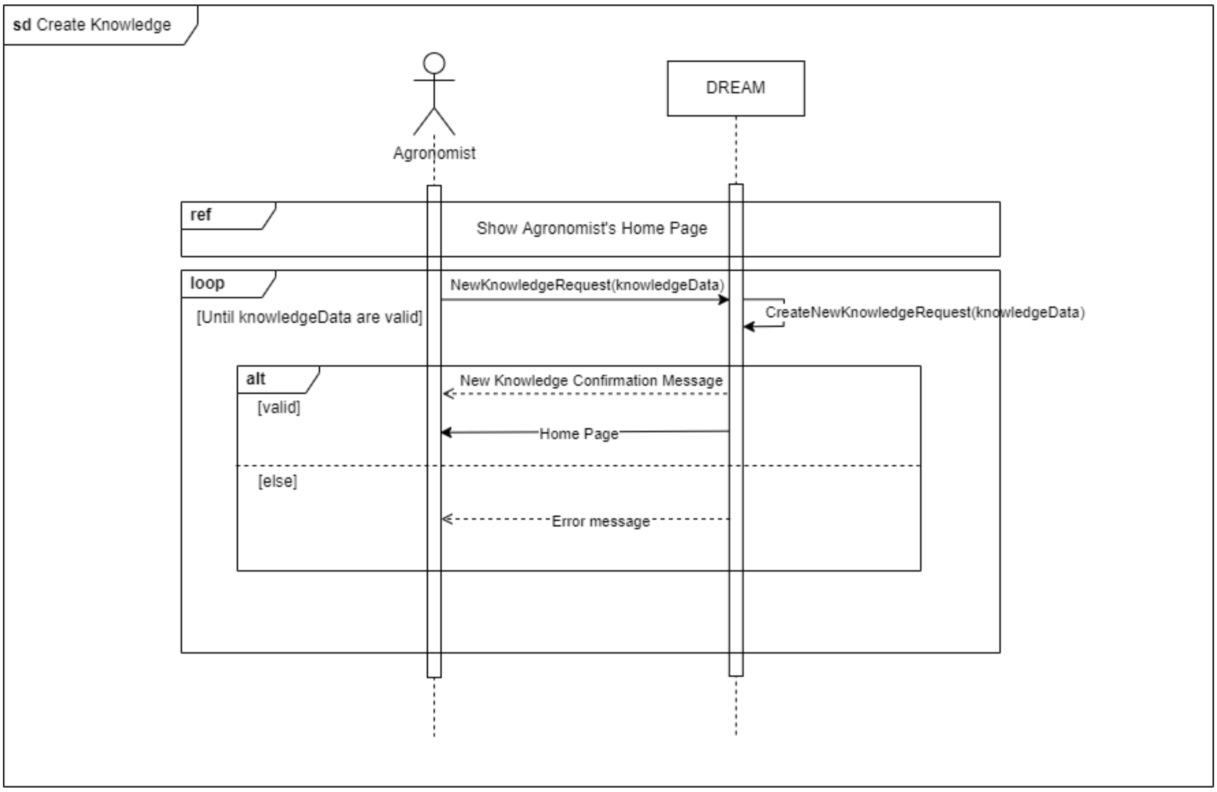

sd Create Knowledge
Agronomist
DREAM
ref
Show Agronomist's Home Page
loop
[Until knowledgeData are valid]
NewKnowledgeRequest(knowledgeData)
CreateNewKnowledgeRequest(knowledgeData)
alt
[valid]
New Knowledge Confirmation Message
Home Page
[else]
Error message

Figure 17: Sequence Diagram - Create Knowledge

Name	Create Knowledge
Actors	Agronomist
Entry Condition	The actor is already logged into the system and he is in the Home Page
Event Flow	1. The actor inserts the title of the Knowledge 2. The actor inserts the body of the Knowledge 3. The actor inserts the tags of the Knowledge 4. The system checks the Knowledge Data 5. The system saves the Knowledge Data 6. The system displays a success Message 7. The system displays Home Page
Exit Condition	The Knowledge is successfully saved into the system and the Home Page is shown
Exception	1. Any mandatory fields are not filled

11. Insert Farm Data

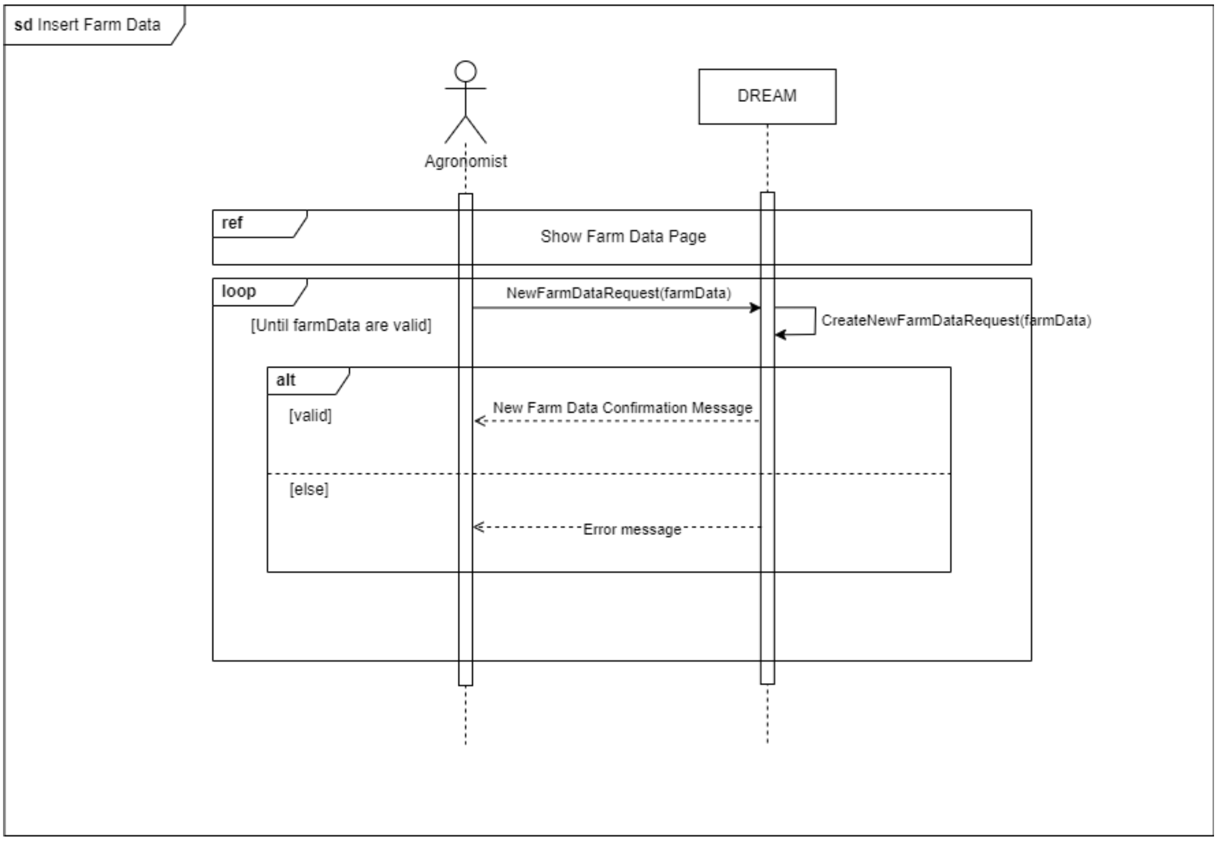

sd Insert Farm Data
Agronomist
DREAM
ref
Show Farm Data Page
loop
[Until farmData are valid]
NewFarmDataRequest(farmData)
CreateNewFarmDataRequest(farmData)
alt
[valid]
New Farm Data Confirmation Message
[else]
Error message

Figure 18: Sequence Diagram - Insert Farm Data

Name	Insert Farm Data
Actors	Agronomist
Entry Condition	The actor is already logged into the system and he is in the Farm Data Page
Event Flow	1. The actor inserts the Farm Data 2. The system checks the Farm Data 3. The system saves the Farm Data 4. The system displays a success Message
Exit Condition	The Farm Data is successfully saved into the system
Exception	1. Any mandatory fields are not filled

12. Notification New Farmer

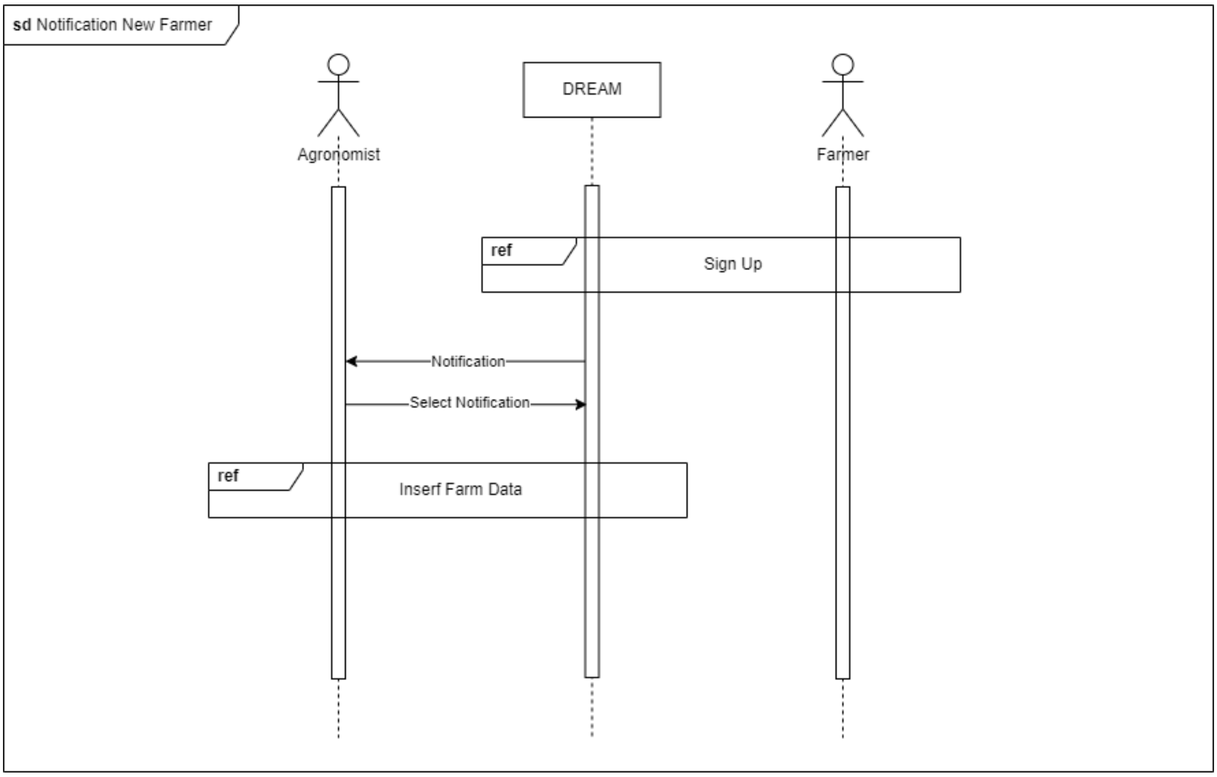


Figure 19: Sequence Diagram - Notification New Farmer

Name	Notification New Farmer
Actors	Farmer, Agronomist
Entry Condition	The Agronomist is already logged into the system and the Farmer has concluded the Sign Up procedure
Event Flow	1. The system sends to the Agronomist a notification 2. The Agronomist selects the notification
Exit Condition	The Agronomist inserts the data of the new Farmer
Exception	1. Loss of internet connection 2. The actor cancels the operation 3. There is no Local Agronomist

13. Show Policy Maker’s Home Page

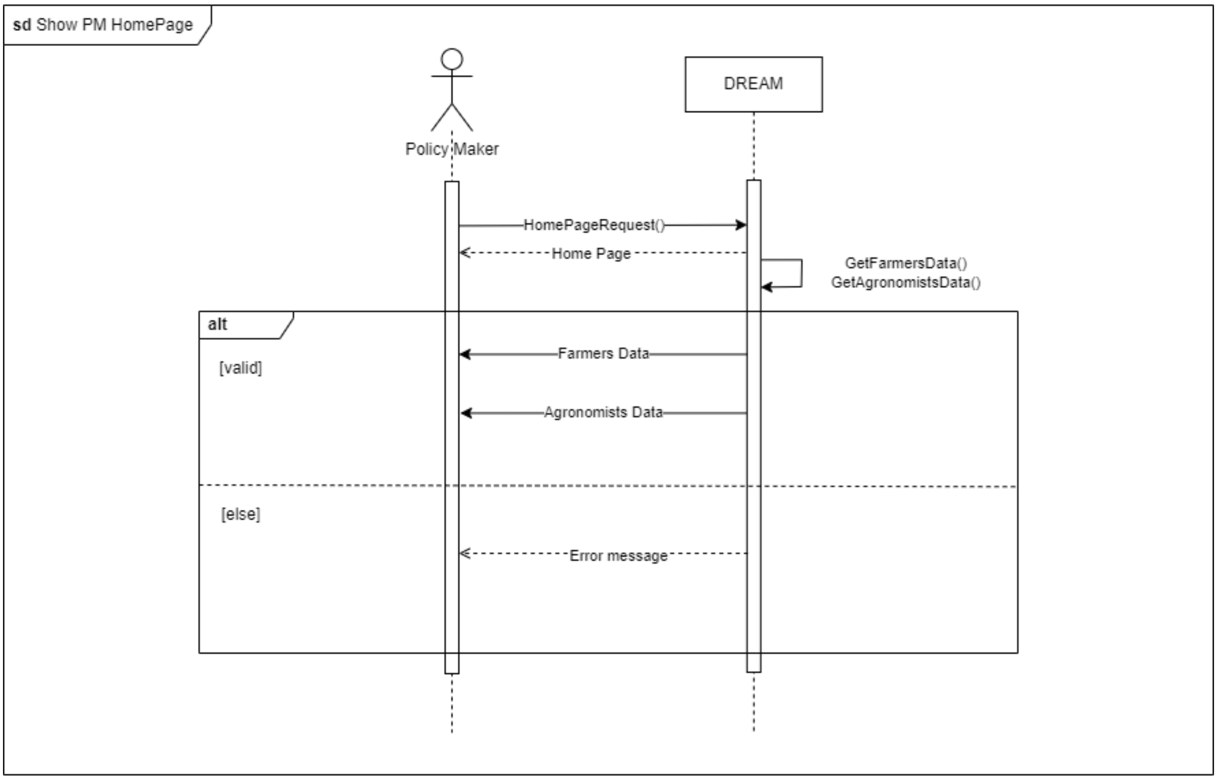


Figure 20: Sequence Diagram - Show Policy Maker’s Home Page

Name	Show Policy Maker’s Home Page
Actors	Policy Maker
Entry Condition	The actor is already logged into the system
Event Flow	1. The actor requires the Home Page 2. The system shows the Home Page to the actor 3. The system retrieves Farmer and Agronomist Data 4. The system displays Farmer and Agronomist Data
Exit Condition	The Home Page with Farmer and Agronomist Data is displayed
Exception	1. The list of Farmer Data is empty 2. The list of Agronomist Data is empty

14. Create New Topic

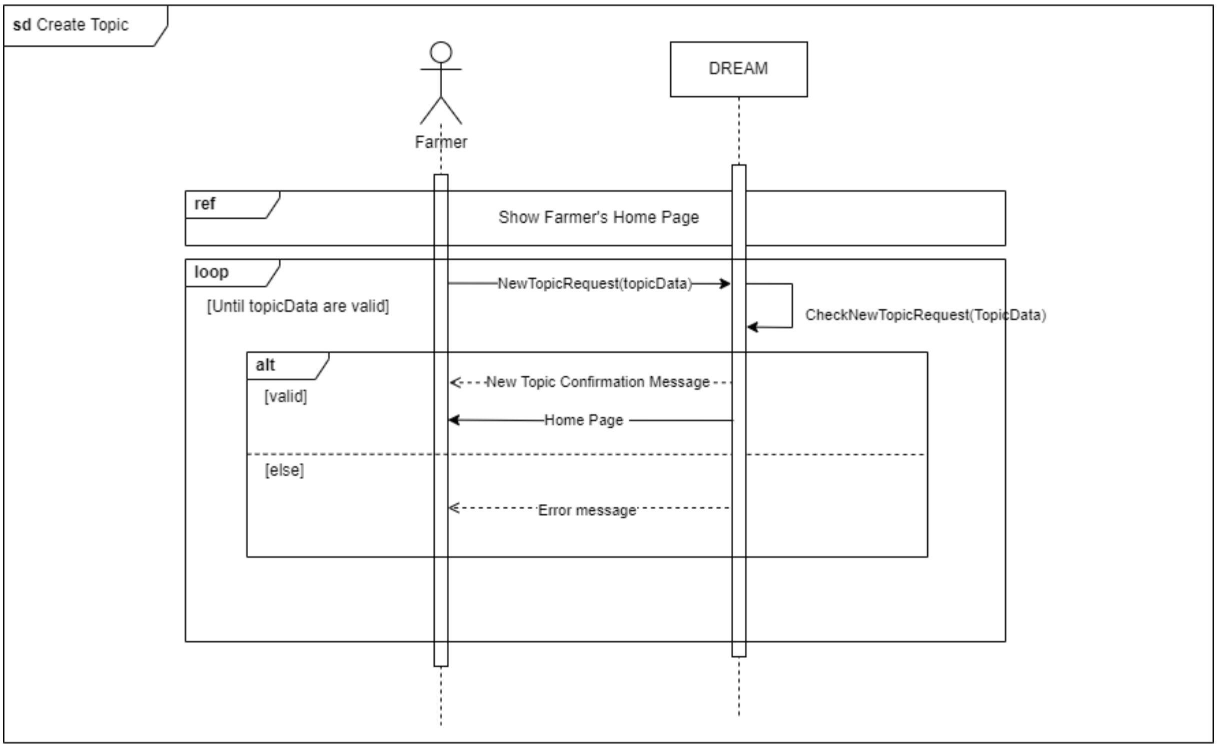

sd Create Topic
Farmer
DREAM
ref
Show Farmer's Home Page
loop
[Until topicData are valid]
NewTopicRequest(topicData)
CheckNewTopicRequest(TopicData)
alt
[valid]
New Topic Confirmation Message
Home Page
[else]
Error message

Figure 21: Sequence Diagram - Create New Topic

Name	Create New Topic
Actors	Farmer
Entry Condition	The farmer is already logged into the system and he is in the Home Page
Event Flow	1. The actor inserts the title of the Topic 2. The actor inserts the body of the Topic 3. The actor inserts the tags of the Topic 4. The system checks the Topic Data 5. The system saves the Topic Data 6. The system displays a success Message 7. The system displays Home Page
Exit Condition	The Topic is successfully saved into the system and the Home Page is shown
Exception	1. Any mandatory fields are not filled

15. Update Production Data

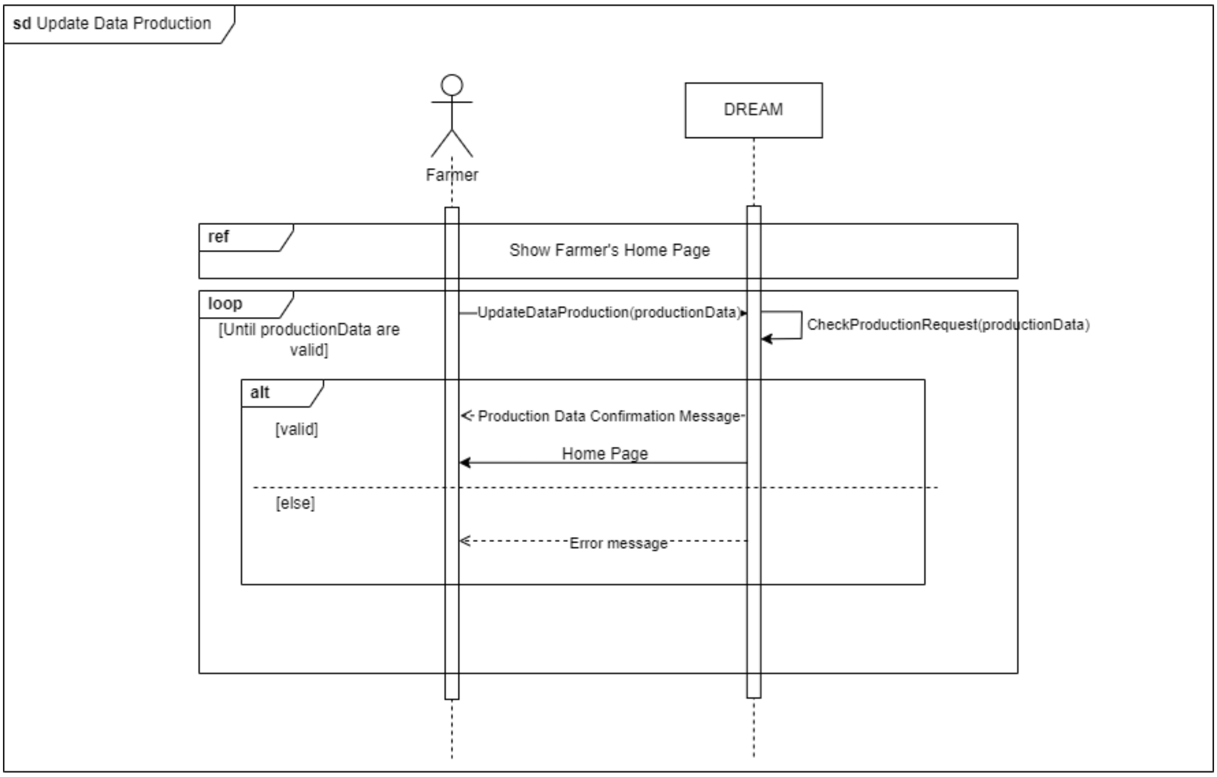

sd Update Data Production
Farmer
DREAM
ref
Show Farmer's Home Page
loop
[Until productionData are valid]
UpdateDataProduction(productionData)
CheckProductionRequest(productionData)
alt
[valid]
Production Data Confirmation Message
Home Page
[else]
Error message

Figure 22: Sequence Diagram - Update Production Data

Name	Update Production Data
Actors	Farmer
Entry Condition	The farmer is already logged into the system and he is in the Home Page
Event Flow	1. The actor inserts the Data of the Production 2. The system checks the Production Data 3. The system saves the Production Data 4. The system displays a success Message
Exit Condition	The Production Data is successfully saved into the system
Exception	1. Any mandatory fields are not filled

16. Comment Topic

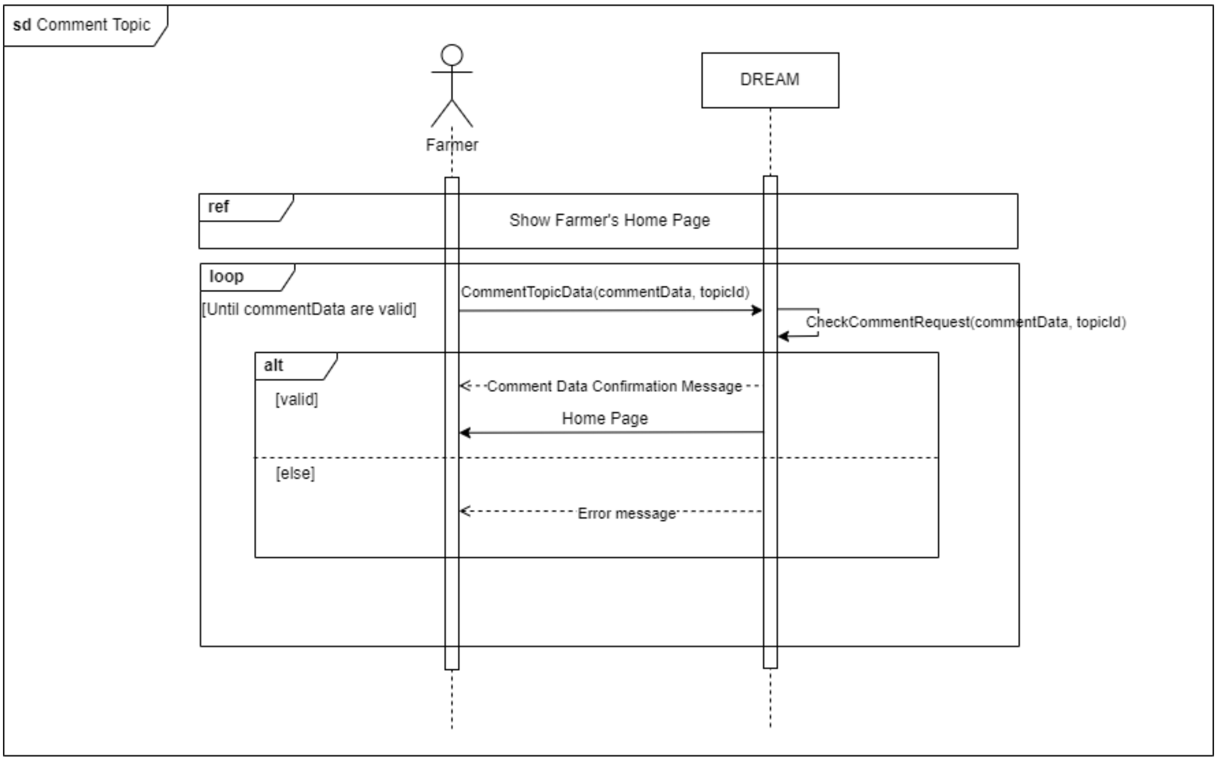

sd Comment Topic
Farmer
DREAM
ref
Show Farmer's Home Page
loop
[Until commentData are valid]
CommentTopicData(commentData, topicId)
CheckCommentRequest(commentData, topicId)
alt
[valid]
Comment Data Confirmation Message
Home Page
[else]
Error message

Figure 23: Sequence Diagram - Comment Topic

Name	Comment Topic
Actors	Farmer
Entry Condition	The farmer is already logged into the system and he is in the Home Page
Event Flow	1. The actor inserts the body of the comment 2. The system checks the Comment Data 3. The system saves the Comment Data 4. The system displays a success Message 5. The system displays Home Page
Exit Condition	The Comment Data is successfully saved into the system
Exception	1. Any mandatory fields are not filled

17. Answer Help Request

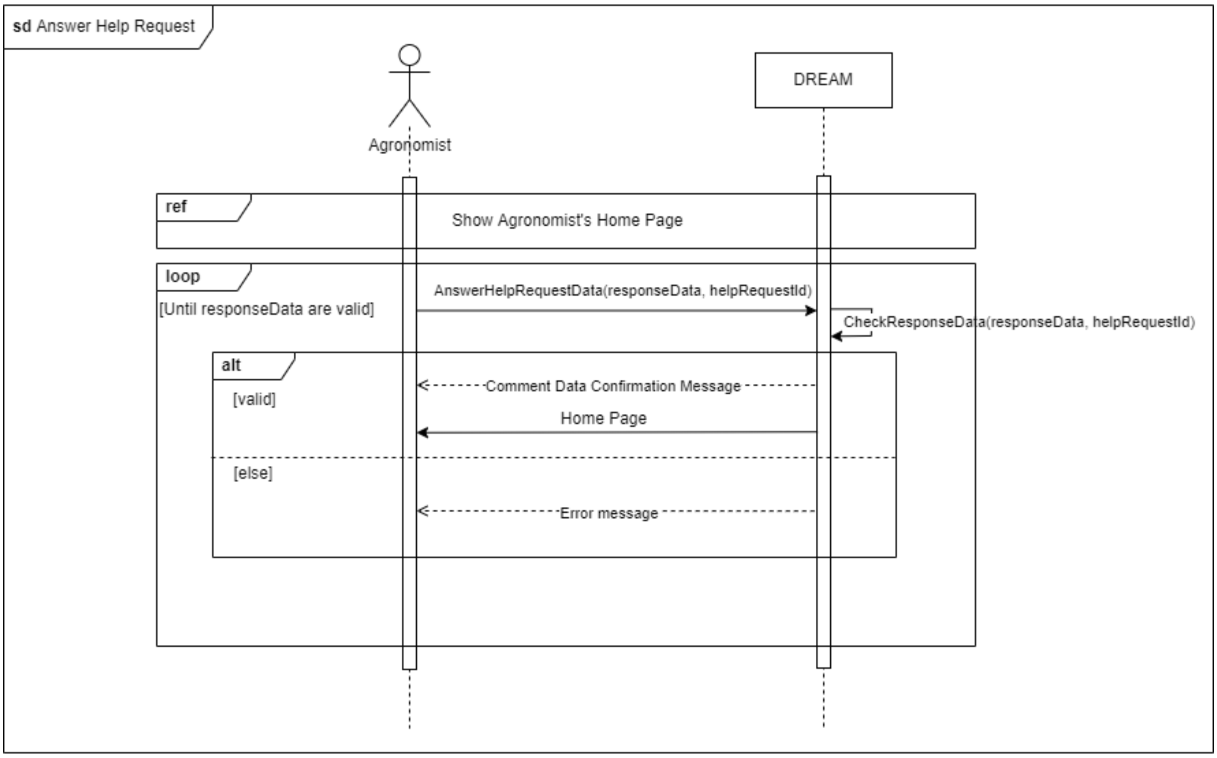

sd Answer Help Request
Agronomist
DREAM
ref
Show Agronomist's Home Page
loop
[Until responseData are valid]
AnswerHelpRequestData(responseData, helpRequestId)
CheckResponseData(responseData, helpRequestId)
alt
[valid]
Comment Data Confirmation Message
Home Page
[else]
Error message

Figure 24: Sequence Diagram - Answer Help Request

Name	Answer Help Request
Actors	Agronomist
Entry Condition	The agronomist is already logged into the system and he is in the Home Page
Event Flow	1. The actor inserts the body of the answer 2. The system checks the Answer Data 3. The system saves the Answer Data 4. The system displays a success Message 5. The system displays Home Page
Exit Condition	The Answer is successfully saved into the systemand the Home Page is shown
Exception	1. Any mandatory fields are not filled

3.2.3 Mapping

Goals	Requirements	Domain Assumptions	Use Cases
G1	R15,R16,R17,R24,R27	D1,D3,D4,D9,D17	U1,U3,U7,U8,U14,U16
G2	R1,R27,R24	D1,D2,D5,D17	U1,U4,U7,U8,U15
G3	R2,R19,R20,R24,R27	D1,D2,D5,D6,D7, D13,D14,D15,D16,D17, D18	U1,U7,U8
G4	R11,R14,R24,R27	D1,D4,D8,D17	U1,U6,U7,U8
G5	R12,R13,R24,R25, R26,R27	D1,D2,D6,D7,D17	U2,U7,U8,U17
G6	R2,R19,R20,R21,R24, R25,R26,R27	D1,D2,D6,D7,D17	U2,U5,U7,U8,U10
G7	R3,R4,R5,R6,R7,R8,R9, R10,R24,R25,R26,R27	D1,D10,D12,D13,D17	U2,U7,U8,U9,U11,U12
G8	R22,R23,R24	D1,D11,D13,D17,D18	U7,U8,U11,U13
G9	R18,R24,R27	D1,D17	U1,U2,U7,U8

G1	Allowing farmer to share their knowledge and problems with other farmers and help them with their farm by creating and commenting a topic
R15	The system allows the farmer to publish a topic on the forum
R16	The system allows all farmers to view the forum
R17	The system allows all farmers to publish an answer to a topic on the forum
R24	The system requires a sign up and a login to access to the data
R27	The system allows the farmer to insert their position when they first register
D1	The internet connection works properly without failure
D3	The tags of a topic must be coherent to it
D4	When a customer shares something on the platform the language must be adequate and unoffensive
D9	When a customer answers something on the platform, the answer must be related to the question
D17	Each Aadhaar is unique
U1	Show Farmer's Home Page
U3	Show Topic Page
U7	Sign Up
U8	Login
U14	Create New Topic
U16	Comment Topic

G2	Allowing farmer to keep track of their production
R1	The system allows the farmer to add data concerning his production
R24	The system requires a sign up and a login to access to the data
R27	The system allows the farmer to insert their position when they first register
D1	The internet connection works properly without failure
D2	The agronomist places the sensors during the first meeting with the farmer
D5	When a farmer updates his data production he must insert reliable and truthful data
D17	Each Aadhaar is unique
U1	Show Farmer’s Home Page
U4	Show Production
U7	Sign Up
U8	Login
U15	Update Production Data

G3	Showing the farmer personalized advice regarding specific subjects of their need, shared by the agronomist
R2	The system allows the agronomist to view the data concerning their farmers' production
R19	The system allows the agronomists to share knowledge with the farmer
R20	The system allows the farmers to visualize the knowledges shared by their agronomist
R24	The system requires a sign up and a login to access to the data
R27	The system allows the farmer to insert their position when they first register
D1	The internet connection works properly without failure
D2	The agronomist places the sensors during the first meeting with the farmer
D5	When a farmer updates his data production he must insert reliable and truthful data
D6	The Knowledge shared by agronomists must be true
D7	The tags of a Knowledge must be coherent to it
D13	After each meeting the agronomist must perform the soil composition's test and update the results on the corresponding farmer's platform
D14	The sensors that acquire the soil composition data provide information with an error of at most 10-15 percentage (for each type of data) compared to the real values
D15	The sensors that acquire the humidity data provide information with an error of at most 4 percentage compared to the real value
D16	The sensors that acquire the water consumption data provide information with an error of at most 5 percentage compared to the real value
D17	Each Aadhaar is unique
D18	Each farmer that is registered to the system has the sensors correctly placed at all times
U1	Show Farmer's Home Page
U7	Sign Up
U8	Login

G4	Allowing farmers to make a request for help to agronomists
R11	The system allows the farmer to send a help request to the agronomist
R14	The system allows the farmer to view all the help requests sent by him
R24	The system requires a sign up and a login to access to the data
R27	The system allows the farmer to insert their position when they first register
D1	The internet connection works properly without failure
D4	When a customer shares something on the platform the language must be adequate and unoffensive
D8	The help requests sent by a farmer must be about farm related topics
D17	Each Aadhaar is unique
U1	Show Farmer's Home Page
U6	Send Request for Help
U7	Sign Up
U8	Login

G5	Allowing agronomists to help farmers in their area by answering to their request
R12	The system allows the agronomist to answer to the help requests that are sent to him
R13	The system allows the agronomist to view all the help requests sent to him
R24	The system requires a sign up and a login to access to the data
R25	The system shows to each agronomist the list of all farmers they are responsible for
R26	The system notifies the agronomists when a new farmer has registered under their area
R27	The system allows the farmer to insert their position when they first register
D1	The internet connection works properly without failure
D2	The agronomist places the sensors during the first meeting with the farmer
D6	The Knowledge shared by agronomists must be true
D7	The tags of a Knowledge must be coherent to it
D17	Each Aadhaar is unique
U2	Show Agronomist’s Home Page
U7	Sign Up
U8	Login
U17	Answer Help Request

G6	Allowing agronomists to help farmers in their area by sharing their knowledge and giving them advice
R2	The system allows the agronomist to view the data concerning their farmers' production
R19	The system allows the agronomists to share knowledge with the farmer
R20	The system allows the farmers to visualize the knowledges shared by their agronomist
R21	The system allows all agronomists to check the weather forecasts for their area
R24	The system requires a sign up and a login to access to the data
R25	The system shows to each agronomist the list of all farmers they are responsible for
R26	The system notifies the agronomists when a new farmer has registered under their area
R27	The system allows the farmer to insert their position when they first register
D1	The internet connection works properly without failure
D2	he agronomist places the sensors during the first meeting with the farmer
D6	The Knowledge shared by agronomists must be true
D7	The tags of a Knowledge must be coherent to it
D17	Each Aadhaar is unique
U2	Show Agronomist's Home Page
U5	Show Farm Info
U7	Sign Up
U8	Login
U10	Create Knowledge

G7	Allowing agronomists to plan visits with farmers to check their progress and if needed, help them
R3	The system allows the agronomist to book an appointment with the farmer by showing all available slots
R4	The system allows the agronomist to modify a booked appointment with the farmer
R5	The system allows the agronomist to cancel a booked appointment with a farmer
R6	The system check that the agronomist booked at least two appointments per year with each farmer
R7	The system alerts the farmer before the appointment
R8	The system alerts the agronomist before the appointment
R9	The system allows the agronomist to view the calendar
R10	The system allows the farmer to view the calendar
R24	The system requires a sign up and a login to access to the data
R25	The system shows to each agronomist the list of all farmers they are responsible for
R26	The system notifies the agronomists when a new farmer has registered under their area
R27	The system allows the farmer to insert their position when they first register
D1	The internet connection works properly without failure
D10	The meeting schedule must be followed unless previously canceled either by the farmer or the agronomist
D12	At least two meetings a year per farmer must be performed
D13	After each meeting the agronomist must perform the soil composition’s test and update the results on the corresponding farmer’s platform
D17	Each Aadhaar is unique

U2	Show Agronomist’s Home Page
U7	Sign Up
U8	Login
U9	Create Meeting
U11	Insert Farm Data
U12	Notification New Farmer

G8	Allowing policy makers to monitor the performance of the farmers to help them or to give them special incentives to encourage them to share their experience
R22	The system allows the agronomists to publish a grade for each farmer after each visit
R23	The system allows the policy makers to view the grades of all farmers of Telangana
R24	The system requires a sign up and a login to access to the data
D1	The internet connection works properly without failure
D11	After each meeting the agronomist must complete the farmer's evaluation
D13	After each meeting the agronomist must perform the soil composition's test and update the results on the corresponding farmer's platform
D17	Each Aadhaar is unique
D18	Each farmer that is registered to the system has the sensors correctly placed at all times
U7	Sign Up
U8	Login
U11	Insert Farm Data
U13	Show Policy Maker's Home Page

G9	Showing customers weather forecasts allowing them to have a better approach with the climate changes
R18	The system allows the farmers to check the weather forecasts for their area
R24	The system requires a sign up and a login to access to the data
R27	The system allows the farmer to insert their position when they first register
D1	The internet connection works properly without failure
D17	Each Aadhaar is unique
U1	Show Farmer's Home Page
U2	Show Agronomist's Home Page
U7	Sign Up
U8	Login

3.3 Performance Requirements

The system should guarantee access to Policy Makers, Agronomists and Customers simultaneously. Since the system is to be considered of a non-critical nature, there is no need to take into account very strict performance requirements. However, the system should provide:

- the list of Topics in the Forum in 10 seconds, or less;
- the list of Help Requests in 10 seconds, or less;
- the list of all the farmers an agronomist is responsible for in 10 seconds, or less;
- the list with the farmers and their grades in 10 seconds, or less;
- the meeting confirmation in 10 seconds, or less;
- the meeting deleting in 10 seconds, or less;
- the loading of the available time slots for a meeting with a farmer in less than 5 seconds;
- push notification with a delay that is imperceptible to the user;
- login and sign up response in less than 5 seconds;

It is important to highlight how a good working internet connection is fundamental to achieve the previously considered performance requirements.

3.4 Design Constraints

3.4.1 Stadards Compliance

All the specifications that have been described in this document must be respected by the system. Moreover, the system uses specific units of measure such as cubic meter for the amount of water consumed, PH has no unit of measure (it is just shown as a number in range 0 to 14) and mg/Kg for all the other parameters that are analyzed during the soil's analysis.
Also, the source code of the application must be commented on and documented adequately.
Finally, since the system is thought to serve the Indian state of Telangana, it should respect the guidelines described by the PDP Bill, India's first law on the protection of personal data.

3.4.2 Hardware Limitations

As previously specified in the "Hardware interfaces" paragraph, all hardware requirements are only considered to improve the user experience. Indeed every kind of device with a stable internet connection and a browser installed can work properly.

3.5 Software System Attributes

3.5.1 Reliability and Availability

In order to guarantee the best service possible, the system needs to work without many interruptions: the system needs to be fault tolerant. The system should offer its functionalities with an availability equal to 99.5% or more so that the system works properly the whole year but two days or less. In order to achieve this goal, the system is tought to guarantee high redundancy for the most critical components. Also, all the scheduled maintenance interventions on the system should be performed during the night in order to achieve the previously described goal.

3.5.2 Security

A fundamental aspect to consider is safety. Since the user provides sensitive information, the connection between the application and the server must be safe. For this purpose, the database containing the data needs to be kept safe from attacks. Also, through encryption techniques, the data need to be encrypted to guarantee consistency and privacy.

3.5.3 Maintainability

Another fundamental aspect to consider while developing the system is maintainability. In order to achieve this goal, the application must be developed thinking that in the future it needs to be easy to modify and possibly fix to contain costs.The source code must be correlated documentation must be updated during the whole life cycle of the system.
All design patterns and choices will be better explained in the further documentation.

3.5.4 Portability

As already discussed in the "Hardware interfaces" and "Software interfaces", the system is thought to be used by many people and it then needs to be compatible with as many devices as possible, here the choice to make it accessible just through a simple browser. In the first phases of development and launch of the system, the only supported platforms will be web applications, future developments could take into account other platforms such as iOS and Android applications to make it easier to use.

4 Formal Analysis Using Alloy

4.1 Objectives of the analysis

In this section, the purpose is to have a formal notation for the modeling activity, describing properties and imposing some constraints that are not possible to show in the class diagram. Afterwards the following entities will be presented:

- the actors (Farmer, Agronomist and Policy Maker);
- the farms and all the information connected to them;
- the part concerning the forum, the Knowledge section and requests for help.

The main goal is also to model:

- that an agronomist can only have meetings with a farmer in his area of expertise;
- that an agronomist can only insert soil data of a farm in his area of expertise;
- that an agronomist can only response at a request of a farmer in his area of expertise;
- that a policy maker can observe the data concerning all the agronomists and farmers.

Further details are given below as comments.

4.2 Alloy Code

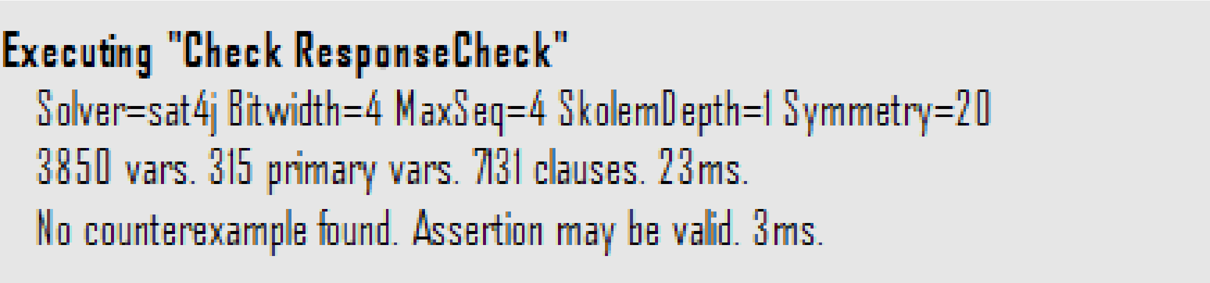
```
// Definition of actors
abstract sig Person{}

sig Farmer extends Person{
        farm: one Farm,
        request: set Request,
        topic: set Topic,
        meeting: set Meeting
}
sig Agronomist extends Person{
        location: one Location,
        response: set Request,
        knowledge: set Knowledge,
        meeting: set Meeting,
        soilData: set SoilData
}
sig PolicyMaker extends Person{
        view: one View, }

//Definition of location
sig Location{
        agronomist: one Agronomist
}
//Definition of farm
sig Farm{
        farmer: one Farmer,
        location: one Location,
        production: set Production,
        soilData: lone SoilData,
}
//Definition of soil data
sig SoilData{
        agronomist: one Agronomist,
        farm: one Farm
}
//Definition of production
sig Production{
        farm: one Farm,
}
//Definition of topic
sig Topic {
        farmer: one Farmer,
        comment: set Comment,
}
//Definition of comment
sig Comment{
        topic: one Topic,
        farmer: one Farmer
}
Definition of request
sig Request{
        farmer: one Farmer,
        agronomist: lone Agronomist,
}
//Definition of meeting
sig Meeting{
        agronomist: one Agronomist,
        farmer: one Farmer,
}
//Definition of Knowledge
sig Knowledge{
        agronomist: one Agronomist,
}
sig View{
        policyMaker: one PolicyMaker,
        agronomist: set Agronomist,
        farmer: set Farmer
}

// Farmers Coherence
// if a "farmer" belongs to a "farm", then that "farm" is owned by that "farmer"
fact FarmerFarmCoherence {
        all f: Farmer, ff: Farm | f.farm = ff iff ff.farmer = f
}
// if a "farmer" create a "request", then that "request" has that "farmer"
fact FarmerRequestCoherence {
        all f: Farmer, r: Request | f.request = r iff r.farmer = f
```

```
}
//two request cannot be made by the same farmer
fact RequestUnique {
	all disj r1,r2: Request | r1.farmer != r2.farmer
}
// if a "farmer" create a "topic", then that "topic" has that "farmer"
fact FarmerTopicCoherence {
	all f: Farmer, t: Topic | f.topic = t iff t.farmer = f
}
// two "farmer" cannot create the same "topic"
fact TopicUnique {
	all disj f1,f2: Farmer | f1.topic not in f2.topic
}
// if a "farmer" partecipate at a "meeting", then that "meeting" has that "farmer"
fact FarmerMeetingCoherence {
	all f: Farmer, m: Meeting | f.meeting = m iff m.farmer = f
}
//two "farmer" cannot participate the same "Meeting"
fact MeetingFUnique {
	all disj f1,fa2: Farmer | f1.meeting not in f2.meeting
}


// Agronomists Coherence
//if an "agronomist" is located in a "Location", then that "Location" is managed by that "agronomist"
fact AgronomistLocationCoherence {
	all a: Agronomist, l: Location | a.location = l iff l.agronomist = a
}
// if an "agronomist" response at a "request", then that "request" has that "agronomist"
fact AgronomistRequestCoherence {
	all a: Agronomist, r: Request | a.response = r iff r.agronomist = a
}
// an agronomist can response at a request of the farmer of the same location
fact RequestLocation{
	all r:Request | r.farmer.farm.location = r.response.location
}
//if an "agronomist" create a "knowledge", then that "knowledge" is created by that "agronomist"
fact AgronomistKnowledgeCoherence {
	all a: Agronomist, k: Knowledge | a.knowledge = k iff k.agronomist = a
}
//two "Agronomist" cannot create teh same "knowldge"
fact KnowledgeUnique {
	all disj a1,a2: Agronomist | a1.knowledge not in a2.knowledge
}
// if an "agronomist" create a "meeting", then that "meeting" is created by that "agronomist"
fact AgronomistMeetingCoherence {
	all a: Agronomist, m: Meeting | a.meeting = m iff m.agronomist = a
}
//two "Agronomist" cannot create the same "Meeting"
fact MeetingAUnique {
	all disj a1,a2: Agronomist | a1.meeting not in a2.meeting
```

```
}
// the agronomist can get a meeting only with a farmer of the same location
fact MeetingLocation{
        all m:Meeting | m.farmer.farm.location = m.agronomist.location
}
//if an "agronomist" update a "SoilData", then that "soilData" is Updated by that "agronomist"
fact AgronomistSoilDataCoherence {
        all a: Agronomist, sd: SoilData | a.soilData = sd iff sd.agronomist = a
}
// the possibilities for the agronomist to insert soil data only with a farmer of the same location
fact SoilDataLocation{
        all sd:SoilData | sd.agronomist.location = sd.farm.location
}

// PolicyMaker Coherence
// if an "policyMaker" have a "view", then that "view" has that "PolicyMaker"
fact PMViewCoherence {
        all pm: PolicyMaker, v: View | pm.view = v iff v.policyMaker = pm
}
//Farm Coherence
// if a "farm" has a "production", then that "production" is about that "farm"
fact FarmProductionCoherence {
        all f: Farm, p: Production | f.production = p iff p.farm = f
}
//two "farmer" cannot have the same "production"
fact ProductionUnique {
        all disj p1,p2: Production | p1.farm != p2.farm
}
// if a "farm" has a "soil data", then that "soil data" is about that "farm"
fact FarmSoilDataCoherence {
        all f: Farm, sd: SoilData | f.soilData = sd iff sd.farm = f
}

// Topic Coherence
// if a "topic" has a "comment", then that "comment" is about that "topic"
fact TopicCommentCoherence {
        all t: Topic, c: Comment | t.comment = c iff c.topic = t
}
// two topic cannot have the same comment
fact CommentUnique {
        all disj t1,t2: Topic | t1.comment not in t2.comment
}

// the PolicyMaker can view all Agronomist and Farmer
fact ViewAllAgronomist {
        all v: View | all a: Agronomist | a in v.agronomist
}
fact ViewAllFarmer {
        all v: View | all f: Farmer | f in v.farmer
}
```

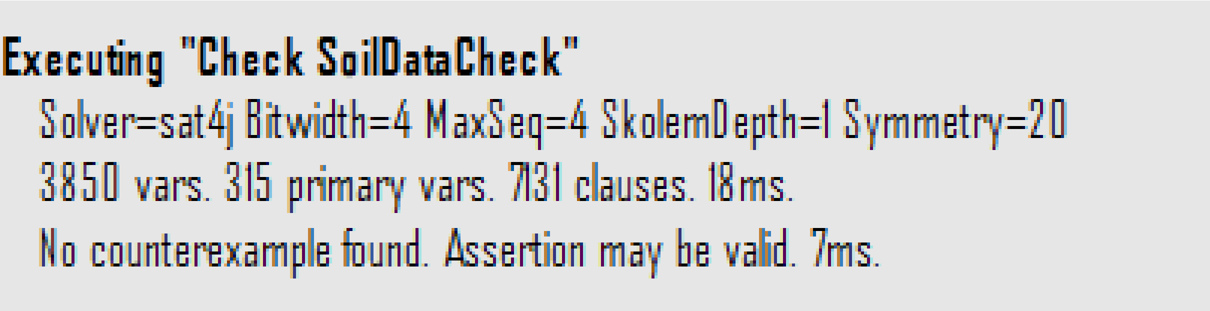
```
pred show { } run show for 5

// Check if two Agronomist can response to the same request
assert ResponseCheck {
        all disj  a1, a2: Agronomist | a1.response not in a2.response
}
Check ResponseCheck

// Check if two Agronomist can insert the same SoilData
assert SoilDataCheck {
        all disj  a1, a2: Agronomist | a1.soilData not in a2.soilData
}
Check SoilDataCheck
```

4.3 Assertions Results



```
Executing "Check ResponseCheck"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   3850 vars. 315 primary vars. 7131 clauses. 23ms.
   No counterexample found. Assertion may be valid. 3ms.
```

Figure 25: Assertions Results - ResponseCheck



```
Executing "Check SoilDataCheck"
   Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
   3850 vars. 315 primary vars. 7131 clauses. 18ms.
   No counterexample found. Assertion may be valid. 7ms.
```

Figure 26: Assertions Results - SoilDataCheck

4.4 Models

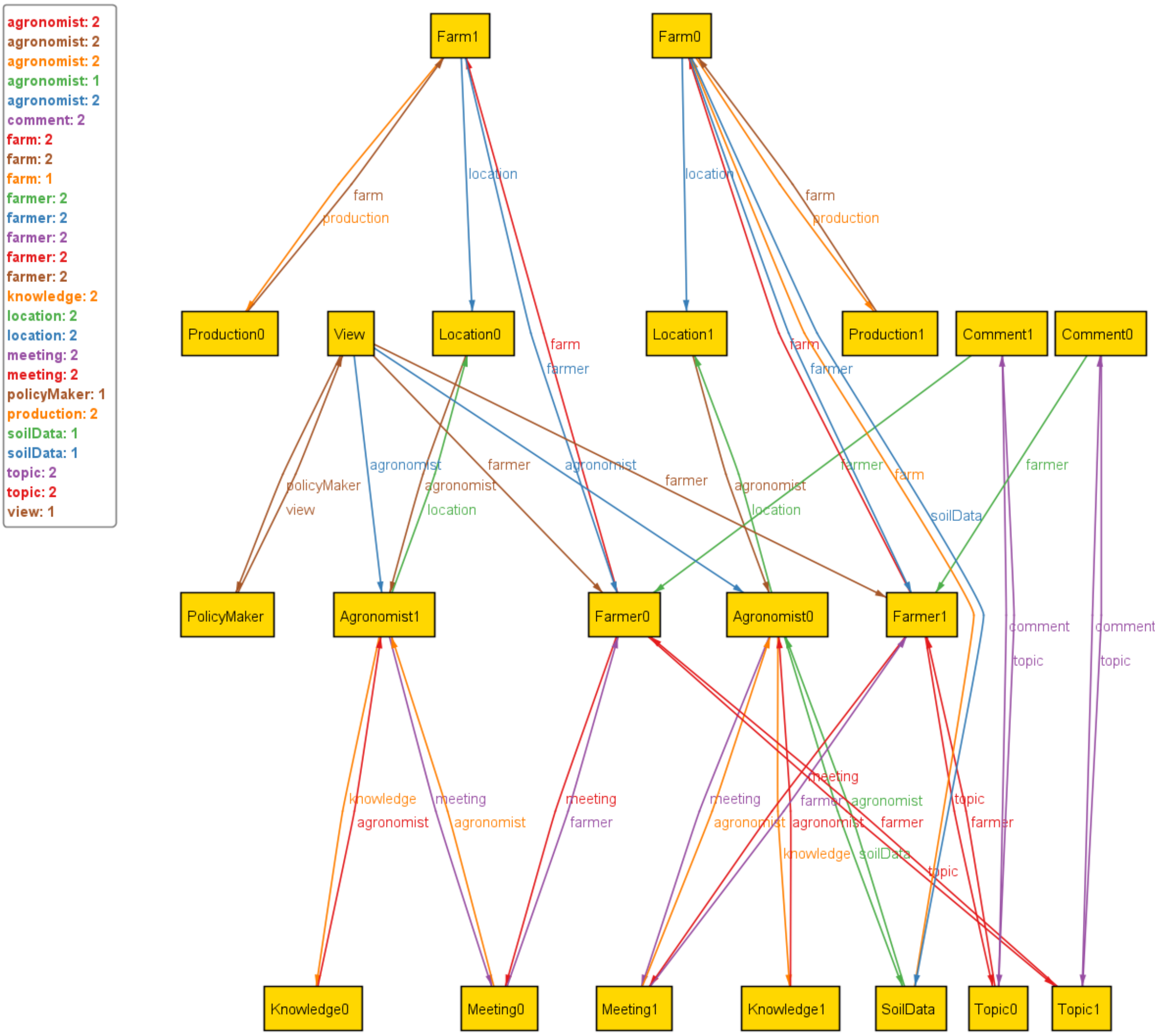


Figure 27: Predicate show

5 Effort Spent

This section shows the amount of time that each member has spent to produce the document. Please notice that each part of the document is the result of coordinated work. The column Member specifies only the main contributor (or contributors, if more than one) for each topic but should not be interpreted as a lack of participation by other team members for that topic.

Topic	**Member**	**Hours**
General initial brainstorming and interpretation of the domain	Buttiglione, Caffagnini, Faouzi	5
Purpose	Buttiglione	4
Scope	Caffagnini	4
Goal	Faouzi	3
Definitions, Acronyms, Abbreviations	Faouzi	1.5
Revision history , Reference Documents and Document Structure	Faouzi	2
Product perspective	Caffagnini	2
Scenarios	Faouzi	5
UML Class Diagram	Faouzi	3
State Charts	Buttiglione	4
Product functions	Caffagnini	3
Requiements, use case and domain assumption analysis	Buttiglione, Caffagnini, Faouzi	7
Requirements and User Characteristics definition	Caffagnini	4
Domain assumption definition	Caffagnini	2
Mock-up and User interface	Buttiglione	8
Hardware, Software and Comunication interfaces	Buttiglione	3
Use case Diagram	Faouzi	4
Sequence Diagrams	Buttiglione, Caffagnini, Faouzi	15
Mapping between goals, requirements and domain assumptions analysis	Buttiglione, Caffagnini, Faouzi	5
Mapping between goals, requirements and domain assumptions definition	Caffagnini	8
Definition of Performance Requirements and Design Constraints	Caffagnini	2
Alloy Modelling	Buttiglione,Caffagnini, Faouzi	8
Alloy improvements	Faouzi	5
Alloy description and corrections, integration into the document, and final adjustments	Buttiglione	4
Final Improvement	Buttiglione , Caffagnini, Faouzi	5
Second Version Modifications	Buttiglione , Caffagnini, Faouzi	5

6 References

- The diagrams have been made with: https://www.visual-paradigm.com
- The Mock-ups have been made with: https://www.figma.com/
- Alloy Language Reference: https://alloytools.org/download/alloy-language-reference.pdf
- Alloy Tools: https://alloytools.org/tutorials/day-course/
- Sequence Diagram Reference: https://www.uml-diagrams.org/sequence-diagrams-reference.html
- Mock-up Design Reference: https://italia.github.io/bootstrap-italia/