RASD

Requirements Analysis and Specification Document

DREAM

Salvatore Marragony (10617299)

Alessandro Maranelli (10661029)

Amir Bachir Kaddis Beshay (10659740)

Version 1.0

CONTENTS

1 Introduction

1.1 Context

Most of the Indian rural population (898 millions) relies on agriculture as its main source of income. These people represent over 58% of total population, 80% of whom are smallholder farmers with less than 2 hectares of farmland. More than a fifth of the smallholder farm households are below poverty. As stated in The Harvard Business review, global food demand is expected to increase anywhere between 59% to 98% by 2050. In addition, climate change is already threatening agricultural products and the whole food system chain, and it is predicted to cause a net loss of 4%-25% for farmers by the end of the century. As these weren't enough, COVID-19 highlighted the weaknesses and vulnerabilities that a massive but unorganized community can face. This is due to a lack of communication and centralized intelligence.
For this purpose, Telangana's government aims to profit from the technologies and methodologies developed in the last decades. The long-term goal of Telangana's government is to design, develop and demonstrate anticipatory governance models for food systems using digital public goods and community-centric approaches to strengthen data-driven policy making in the state. Within this context, the DREAM app will be a powerful support to collect data about agriculture in the state.

1.2 Purpose

The purpose of this document is to present a detailed description of the DREAM app. It provides functional and non-functional requirements for the development of the system, including use cases, features, user interaction and system constraints.
This document is addressed to the developers who have to implement the requirements and could be used as an agreement between the customer and the contractors.

1.2.1 Goals

The aim of the DREAM application is to provide a specific interface for each different user (farmers, agronomists and Policy Makers) that allows them to monitor useful information for their purposes but also allows communication, supervision and discussions between the 3 actors.
Below are presented the goals of DREAM.

Gx	Description of the goal
G1	Allow TPMs to have an overview of farmers' performances
G2	Allow TPMs to visualize the result of steering initiatives
G3	Allow agronomists to visualize all relevant data concerning their area
G4	Allow agronomists to answer farmers' help requests
G5	Allow agronomists to manage a daily plan to visit farms
G6	Allow farmers to create discussion forums with the other farmers
G7	Allow farmers to visualize every relevant data they need for their work
G8	Allow farmers to ask for help and suggestions

G9	Allow every user to have a proper account for the type of user
G10	Allow TPMs to create BPAs
G11	Allow agronomists to create steering initiatives

1.3 Scope

DREAM is an application meant to protect and improve agricultural production in the state of Telangana as it will include helpful online production data and organizational and environmental information.
There are 3 types of users: farmers, agronomists and Telangana's policy makers. Farmers, in order to get a successful production, will get through the application updates on the weather, soil and water conditions. The application will also provide them with discussion forums to reach other farmers and agronomists and suggestions on the most suitable crops to plant.
Agronomists are experts whose job is to periodically visit farm households, answer requests from farmers and monitor the environmental conditions. They are supposed to visit farms for which they are responsible for at least twice a year, so DREAM helps them with a digital agenda to confirm or modify their daily plan. They can also promote steering initiatives to enhance farmers' productions.
Finally, Telangana's policy makers need to gather a whole picture of the production system and figure out if the agronomists' initiatives got the expected results. They can visualize the performances of each farmer and the conditions they are operating in to either send incentives to the best ones or give help for those in need.

World Phenomena	Description
W1	Environmental conditions (weather, soil, water)
W2	Farmers produce crop
W3	Farmers use resources (water, fertilizers,...)
W4	Agronomists visit farms
W5	Farmers implement agronomists' steering initiatives

Shared Phenomena	Description
S1	Users log in the application
S2	Users log out of the application
S3	Agronomists insert the area they are responsible of
S4	Agronomists receive information about requests for help
S5	Agronomists answer to farmers who need help
S6	Agronomists visualize weather forecasts

S7	Agronomists create a daily plan to visit farms in the area
S8	Agronomists update the daily plan
S9	Agronomists confirm the execution of the daily plan at the end of each day
S10	Agronomists specify the deviations from the daily plan at the end of each day
S11	Agronomists create new steering initiatives
S12	TPMs write new BPAs
S13	TPMs report bad performing farmers to agronomists
S14	TPMs ask the best farmers to fill a form with their best practices
S15	Farmers visualize relevant data and suggestions
S16	Farmers insert data about their production
S17	Farmers ask for help
S18	Farmers send messages on discussion forums

1.4 Definitions, Acronyms and abbreviations

1.4.1 Definitions

Soil moisture	percentage of water with respect to dry soil
Agronomist	professional in the science, practice, and management of agriculture and agribusiness
Best Practices Article	an article written by a TPM after receiving best practices forms by well-performing farmers. It has a title and a list of topics covered in order to be easily searchable by farmers.

1.4.2 Acronyms

RASD	Requirement Analysis and Specification Document
U.C	Use case
BPA	Best Practices Article
TPM	Telangana Policy Maker

1.4.3 Abbreviations

Gx	Goal number x.
Dx	Domain assumption number x
Rx	Functional requirement number x
SPx	Shared phenomena number x
WPx	World phenomena number x

1.5 Revision history

Version	Date	Description
1.0	19/12/2021	First version

1.6 Reference Documents

- Assignment document A.Y. 2021/2022 ("Requirement Engineering and Design Project: goal, schedule, and rules")
- Course slides

1.7 Document Structure

- Section 1: Introduction: The first section provides an introduction to the purpose of the document and the objectives of the project. Included here is an analysis of the context in which the system will operate, along with a glossary including definitions, acronyms and abbreviations used in this document.
- Section 2: Overall description: This section opens with the description of some scenarios that demonstrate the possible uses of the application, followed by a description of the domain (done through class and state diagrams). Subsequently, the section includes the main functions of the application and the description of user characteristics. At the end of the section, the main assumptions relevant to the functioning of the system are then presented, divided into text and domain assumptions.
- Section 3: Specific requirements: This part of the document provides more details, which may be useful to the development team, on the aspects presented in the previous section. Included here are the system requirements (both functional and non-functional) and their mapping to the goals presented in section 2. The specific use cases, developed here from the scenarios, are then enriched with different UML diagrams.
- Section 4: Formal analysis using Alloy: This section includes the Alloy code accompanied by some of the worlds obtained by running it. A brief introduction clarifies the goal of the modelling activity itself.
- Section 5: Effort spent: This section includes information on the number of hours each group member worked for this document.
- Section 6: References: This section includes the reference documents.

2 Overall Description

2.1 Product Perspective

2.1.1 Scenarios

Here are some of the possible scenarios of usage of the application.

Identify well performing farmers
Ram is a policy maker of the Telangana state. The government informed him that the equivalent of 10.000$ in Indian rupees is available to be given as incentives to 5 particularly well performing farmers of the state. So Ram accesses the DREAM application and goes to the interface where he can visualize the performance score of all farmers' last productions. He notices that the average score is not very high, since the algorithm reports a massive shortage of rainfall over the period of the last productions, so many farmers had problems. He finds 5 farmers with a strong positive deviance though, so he decides that they'll be the ones receiving the incentives. He sends them the best practices forms to fill and after their answers he will manage the delivery of the incentives and possibly write a BPA.

Irrigation based on weather forecasts
Rajesh is a farmer from the Medak district. He started a big production of green grams which is growing very well, so he's very precise with the irrigation. He opens the DREAM application to visualize the weather forecasts and he sees that heavy rain is expected for the next two days, so he switches off the irrigation system so as not to provide too much water to the plants and ruin the crop.

Production problem report
Farid has a medium-sized farm in which for a long time he has mainly grown potatoes. This year, following a new European trend he saw on the web, he decided to halve the ground dedicated to potatoes to try a new fertilizer that doubles the quantity of Nitrogen in its formula. The purpose of this change is to increase crop yields. Everything is going according to plan, but then winter comes: late-season cold weather ruins all the new experimental crops, leaving Farid with only half harvest. He immediately reports the fact on the farmers' forum in the DREAM app, stating that in his area it is not possible to use that new fertilizer due to weather conditions. Then he goes into the current production section and writes a whole page to describe this unfortunate problem that he encountered.

Daily plan
Ishani is an agronomist responsible for the district of Karimnagar. Weekend is ending and she has to schedule her incoming workweek. Tuesday is usually the weekday dedicated to farm visits, so she starts to elaborate her daily plan inside the DREAM app. She absolutely has to visit Bharat's farm, a farm that is under-performing and has reported plenty of problems in the last weeks (it is marked with a "!" in the application), so she schedules this visit in the morning. In the afternoon, since there are no emergencies, she decides to plan a visit to Anand, an expert and very consistent farmer in the area who has not received a visit yet this year. She confirms the daily plan and when Tuesday comes, she visits the two farmers as she programmed. She finishes the two visits and, after noticing it took less time than she predicted, she decides to also stop at Schrute farms, the beets leading producers of the state. At the end of the day Ishani logs in the application and confirms the execution of the daily plan, also adding the visit to Schrute farms.

Creating a steering initiative

It has been a year since the government of the district of Nizamabad changed the main irrigation pipelines and increased the water pressure to make this precious resource flow more efficiently through the country. Patrick, a Telangana state agronomist, wants to exploit this improvement so he visits three among the best farmers in his area to propose to them a steering initiative based on a heavier use of water irrigation in their potato production. They agree so he creates in the DREAM application a new steering initiative with a 3 months-long period and an accurate description of the new approach, inserting the names and the positions of the farmers involved.

2.1.2 Class Diagram

The class diagram is a high-level representation of the system as a whole.
It identifies three types of actors, the Policy Makers, the Farmers and the Agronomists, and it links everyone to the entities they are related with.
Policy makers have a whole vision of the situation; therefore, they can see the farmers and their productions, which are characterized by a type of crop and by the fertilizers used, in addition to the amount of water used. Farmers can also report any problem they face during the production. TPMs also have the possibility to send Best Practice forms to farmers and edit BPAs when they receive answers.
Both farmers and agronomists are linked to a specific area, and within this area, through the application, they can see the weather forecasts.
Furthermore, farmers can send requests and agronomists or other farmers can answer them.
Agronomists can compile and update a daily plan of farm visits, and can promote steering initiatives, whose results are then analysed by policy makers.

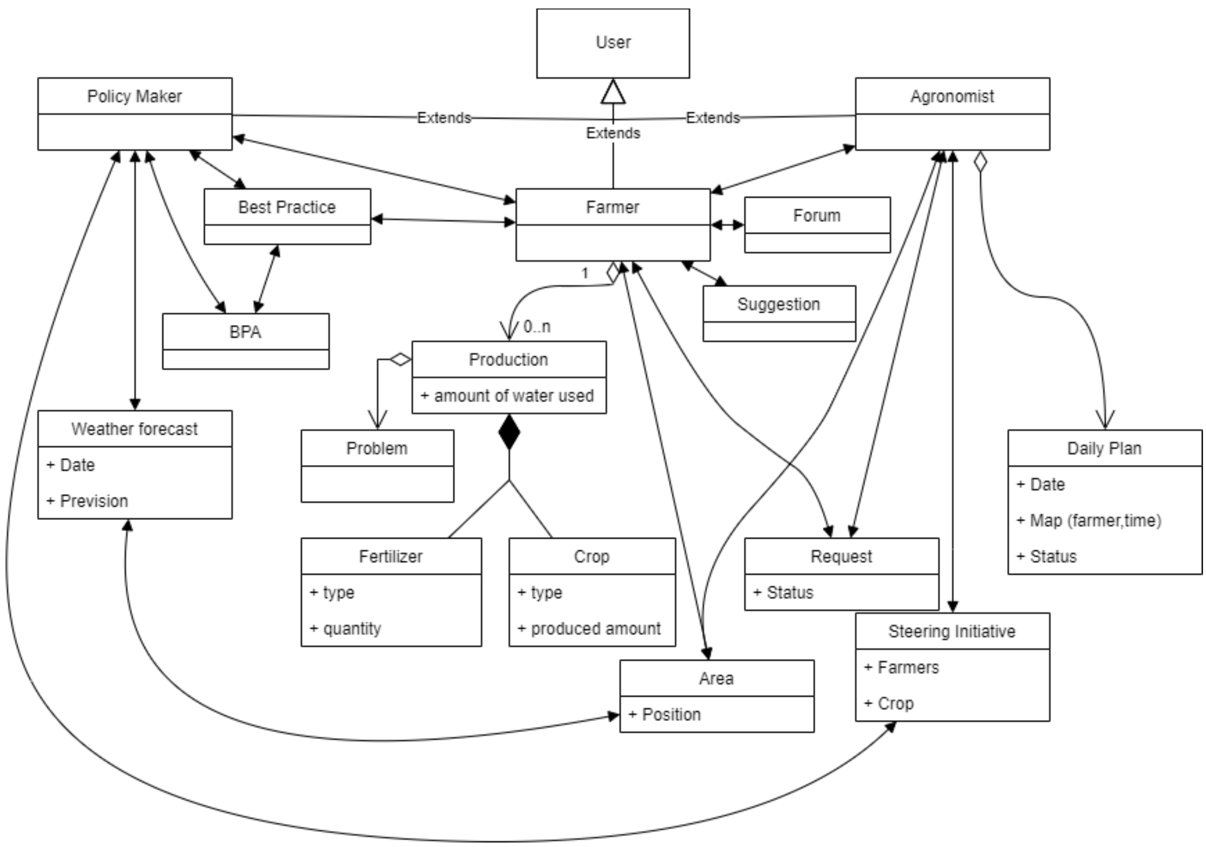


Figure 1 - Class diagram

2.1.3 State Diagrams

State diagrams describe the behaviour of the system while considering all the possible states the objects can have when an event occurs. This analysis helps to understand the most critical aspects of the system.

For clarity, diagrams about harvest and irrigation are included: even though they represent world phenomena, they highlight the actions of the farmers in relation to the application in every part of the harvest cycle and they help to better understand the diagram in fig. 2.3, which describes the steps from the harvest to the evaluation of a production.

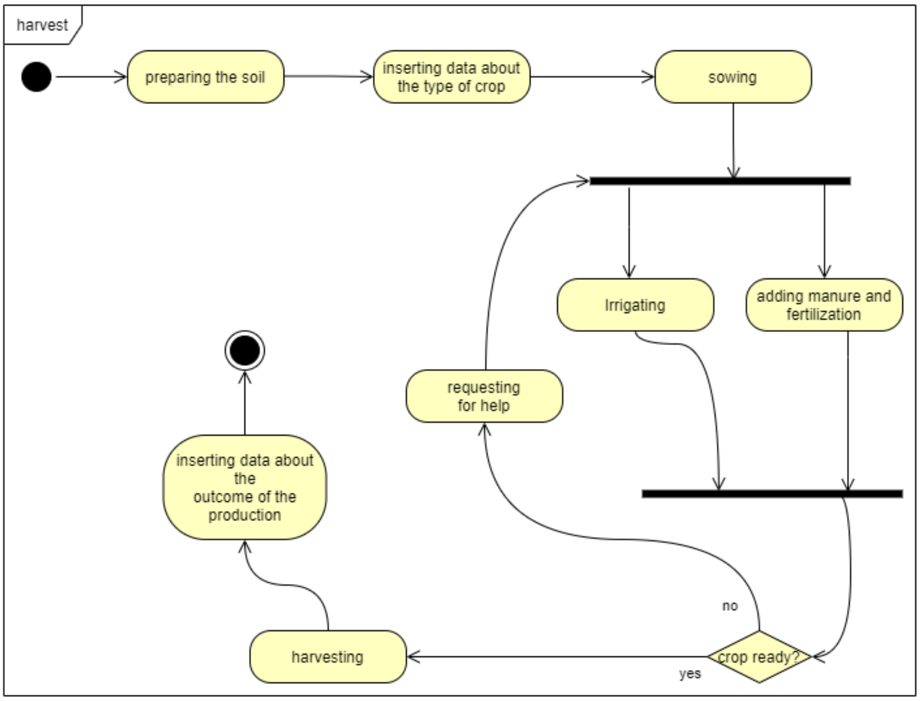


Figure 2.1 - Harvest state diagram

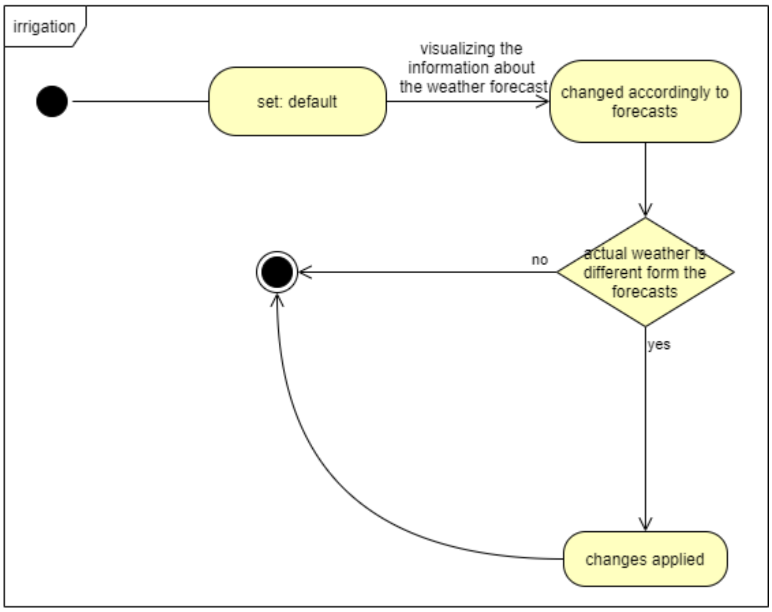


Figure 2.2 - Irrigation state diagram

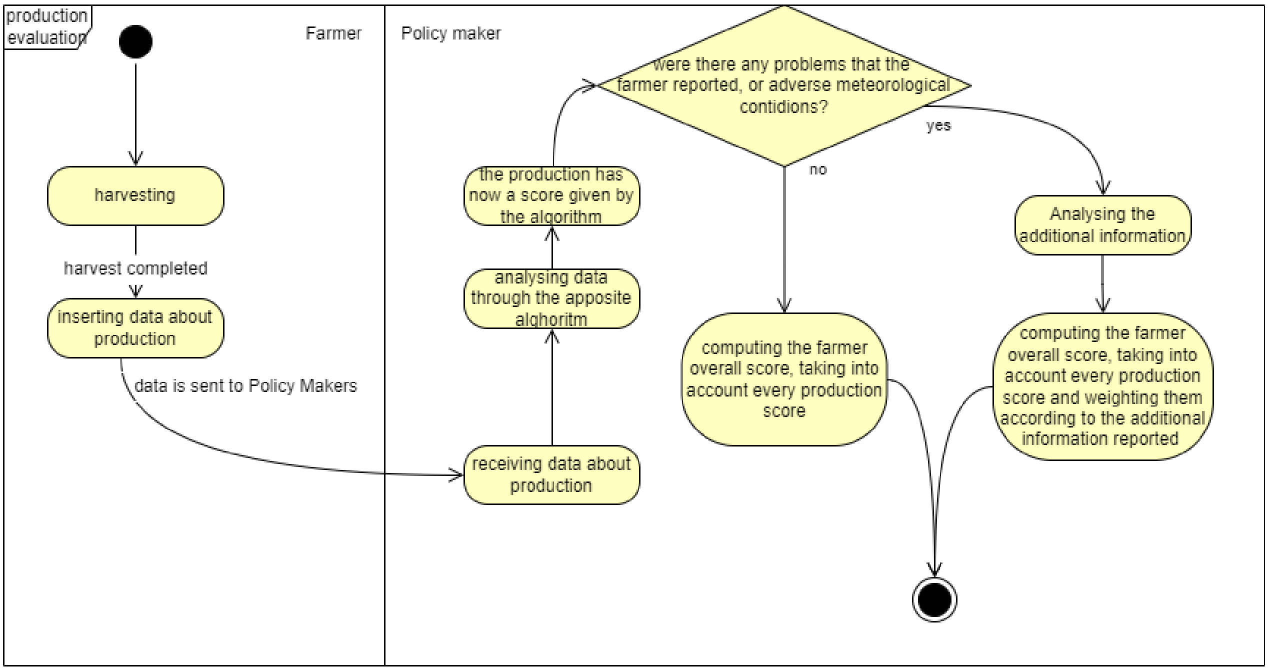


Figure 2.3 - Production evaluation state diagram

The diagram in fig 2.4 represents the various states a daily plan can be in before being confirmed and, eventually, executed.

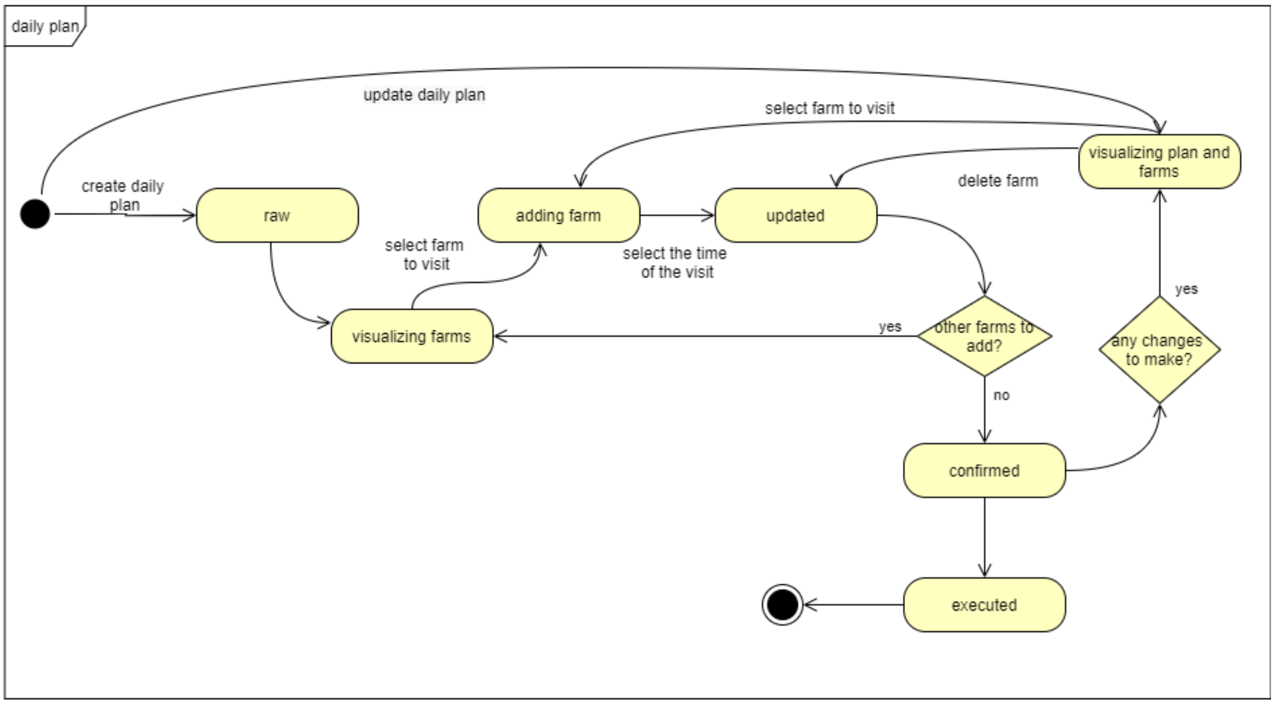


Figure 2.4 - Daily plan state diagram

2.2 Product functions

The application provides different functions for the actors who can use it.

2.2.1 Functions for policy makers

Performance analysis

The main goal of policy makers is to understand whether the different farmers are performing well or not. The application helps them with an algorithm that takes in account many factors, such as crop productivity, crop diversification, soil health, soil moisture, soil organic carbon, amount of water used, and produces as outcome a score for each production of each farmer. Thanks to this score, the policy makers will be able to find the farmers who are performing extremely well (the ones with Positive Deviance) and the farmers who are performing badly (the ones with Negative Deviance). Every farmer has an overall score, based on the scores of the different productions, and this score is updated by the TPM because it is not only an average of the past production scores, but it also takes into account exceptional weather conditions during the productions since the aim of the application is to help finding new best practices to create food productions which are resilient to climate changes. So, productions can have the same score but the ones with a high score during meteorological adverse events (such as shortage of rainfall or high temperatures) will affect the farmer's overall score more heavily. On the other hand, productions with a bad score during adverse conditions will not decrease the overall score as much as the ones during normal conditions. An algorithm shows the TPM if there were adverse weather conditions during that production. To do so, it retrieves weather forecasts from the Telangana state website (https://www.tsdps.telangana.gov.in/aws.jsp) for each day in the production period, and then computes the data to understand if there were exceptional adverse conditions (for instance if the temperature raised over a certain value for more than a certain number of days). The TPM can also report farmers who need help (for instance because they have a negative trend in the last production scores) to agronomists, who will visualize them marked with a “!”. More information on the algorithms will be provided in the Design Document.

Rephrase best practices

Based on the performance analysis, the TPM chooses among the best performing farmers the ones who have to fill in a form to know their best practices for a specific production. When the TPM receives those forms, he/she extracts valid information in order to create a new BPA, assigning it a title and one or more topics to allow farmers to search BPAs easily. The BPAs will be visible to all farmers.

Steering initiatives analysis

The application will produce a score also for the steering initiatives carried out by agronomists with the help of good farmers. The score of an initiative will be based on the differential between the performance score of the productions affected by the initiative and the previous ones of the same productions before the steering initiative.

2.2.2 Functions for farmers

Visualize relevant data

The application will show farmers relevant data for the production they're making. A farmer will visualize:

- weather forecasts, obtained from the Telangana state website, based on the position of the farmer
- suggestions (concerning specific crops to plant or specific fertilizer to use), obtained by an algorithm which takes into account past productions done by the farmers in the same period and their scores, and data such as the humidity of soil (obtained by sensors on the territory), long term forecasts, suggestions from agronomists, best practices suggested by other farmers. More information on the algorithm will be provided in the Design Document.

Insert production information

An interface will be provided to the farmers to insert their production. At the beginning of a production, they insert the start date, the type of crop and the expected amount.
During a production, they can insert problems they face during the production.
At the end of every production, they will insert information such as the end date of the production, the amount produced, the problems faced during the production.

Manage help requests

At any time, a farmer can create a new help request to ask for help or suggestions on a specific issue. The request will be answered by agronomists and other farmers. The farmer can also check the status of an old request, and when there's a new answer which solves the issue, he can decide to close the request. The farmer can also answer other farmers' requests.

Access to the discussion forums

In this section of the application every farmer can visualize discussion forums on specific topics, start a new discussion or add comments on existing discussions.

2.2.3 Functions for agronomists

Answer to help requests

An agronomist can visualize the open requests from the farmers of his area and choose whether to answer to some of them.

Visualize data

Every agronomist visualizes the weather forecasts for his area and the farmers with positive deviance in the area (the best performing ones).

Visualize and update daily plan

Every agronomist has a daily plan to visit farms in the area. The agronomist has an interface in which he visualizes all the future daily plans. The agronomist can update a daily plan, inserting or removing farms that will be visited that day. The agronomist also visualizes a list of all the farmers in the area, with the number of visits each farmer has already received. Since all farms must be visited at least twice a year, the farmers with less than 2 visits are highlighted in red. Moreover, since the under-performing farmers (the ones with negative deviance) should be visited more often, they are marked with a "!".

Confirm the daily plan

At the end of the day farmers need to confirm the execution of the daily plan. When doing so, the system increases the number of visits for all the farms which have been visited that day by the agronomist. If the agronomist has had some deviations from the scheduled daily plan, it is updated and then confirmed by the agronomist.

Promote steering initiatives

An agronomist can promote steering initiatives finalized to enhance production in a particular area. These actions can be made as a result of an analysis of the soil for example, or after a release of funds for agriculture by the government. A function in the application, then, lets the agronomist insert the time frame and the farms to which this action is directed, with an exhaustive description of the initiative.

2.3 User characteristics

The actors of the system are the following:

- Policy maker: the one who analyses the performances of the farmers and decides whether to give special incentives if they're performing well or on the other hand to send agronomists to help them if they're performing badly. The policy maker also evaluates the steering initiatives carried out by the agronomists with the help of good farmers.
- Farmer: the one who produces and the one who should take the most advance from the development of this application. A farmer visualizes relevant data to have a better production, inserts data about the production and the problems faced, requests for help and suggestions and answers to other farmers' requests, interacts with other farmers through discussion forums.
- Agronomist: the one responsible for visiting and helping farmers in a certain area. An agronomist answers help requests by farmers, visualizes useful data for his area and manages a daily plan with all the farms to be visited. The agronomist can update a daily plan and at the end of each day he/she has to confirm its execution. The agronomist can also create new steering initiatives with the help of good farmers.

2.4 Domain assumptions

D1	Data coming from the water irrigation system and from the soil sensors are correct
D2	Each user has a device with an internet connection
D3	Weather forecasts are always available on the Telangana state website

D4	Algorithms which provide the scores are considered correct
D5	Farmers always insert data of their productions (as soon as they are available)
D6	Agronomists are responsible of only one area
D7	There is only one agronomist per area
D8	Agronomists update daily plans to satisfy the constraint of at least 2 visits per year
D9	Agronomists never confirm a daily plan which is not correct
D10	Daily plans are reset at the beginning of every year

3 Specific Requirements

In this section it is given a complete overview of the functional requirements of the system.

3.1 External interface Requirements

3.1.1 User Interfaces

The following mock-ups show the login interface and the main page for the three different types of users. More mock-ups will be provided in the Design document.

Figure 3.1 - Login page

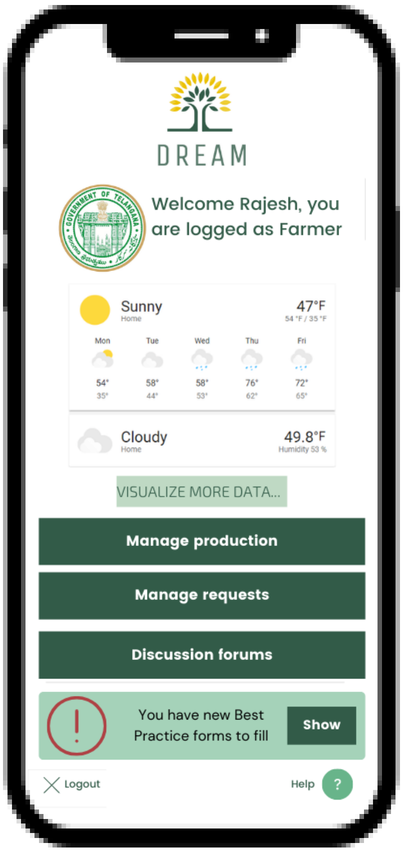


Figure 3.2- Farmer main page

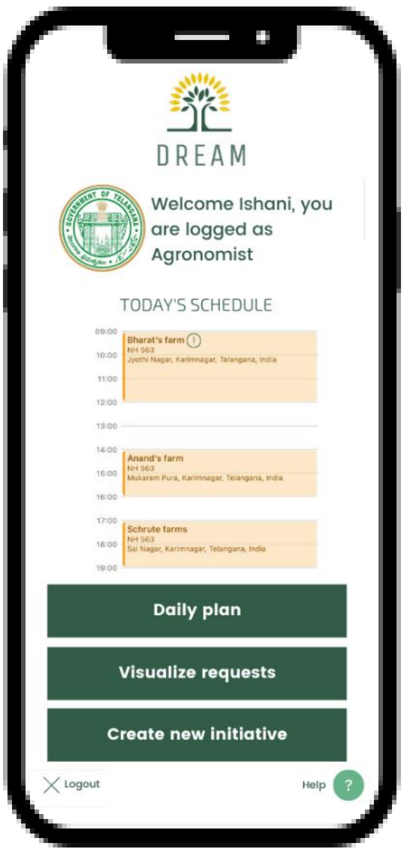


Figure 3.3- Agronomist main page

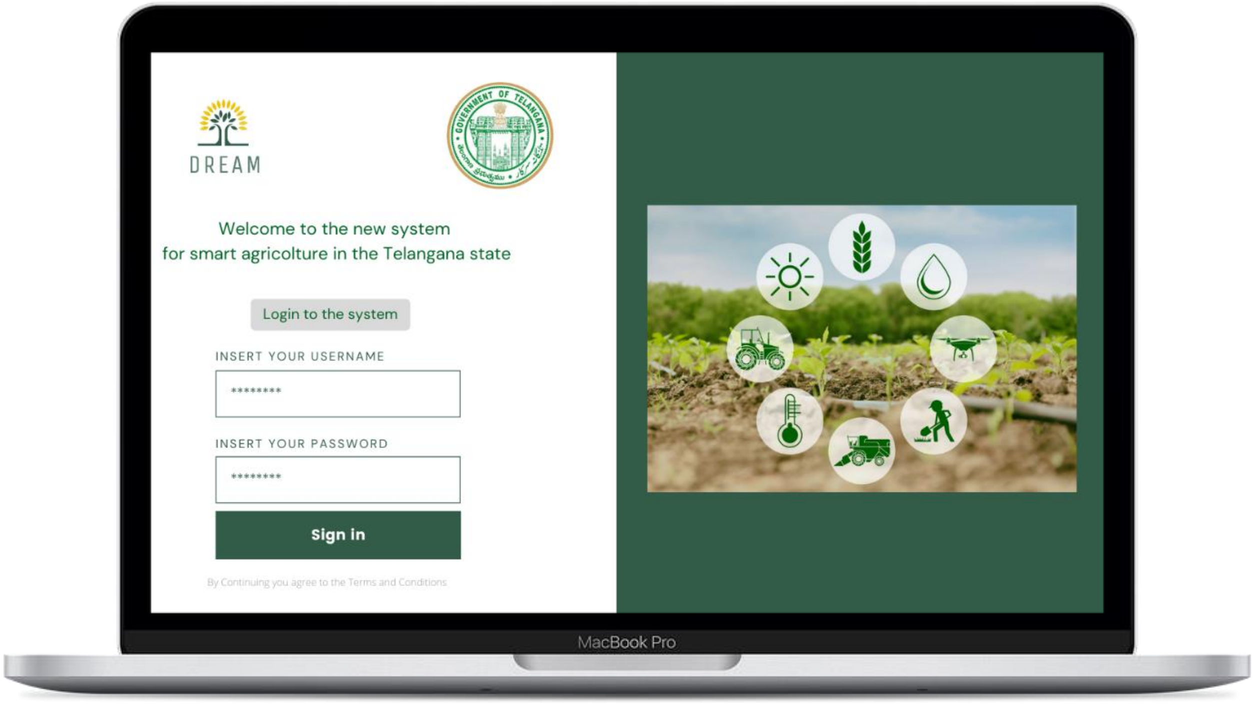


Figure 3.4. - Website login page

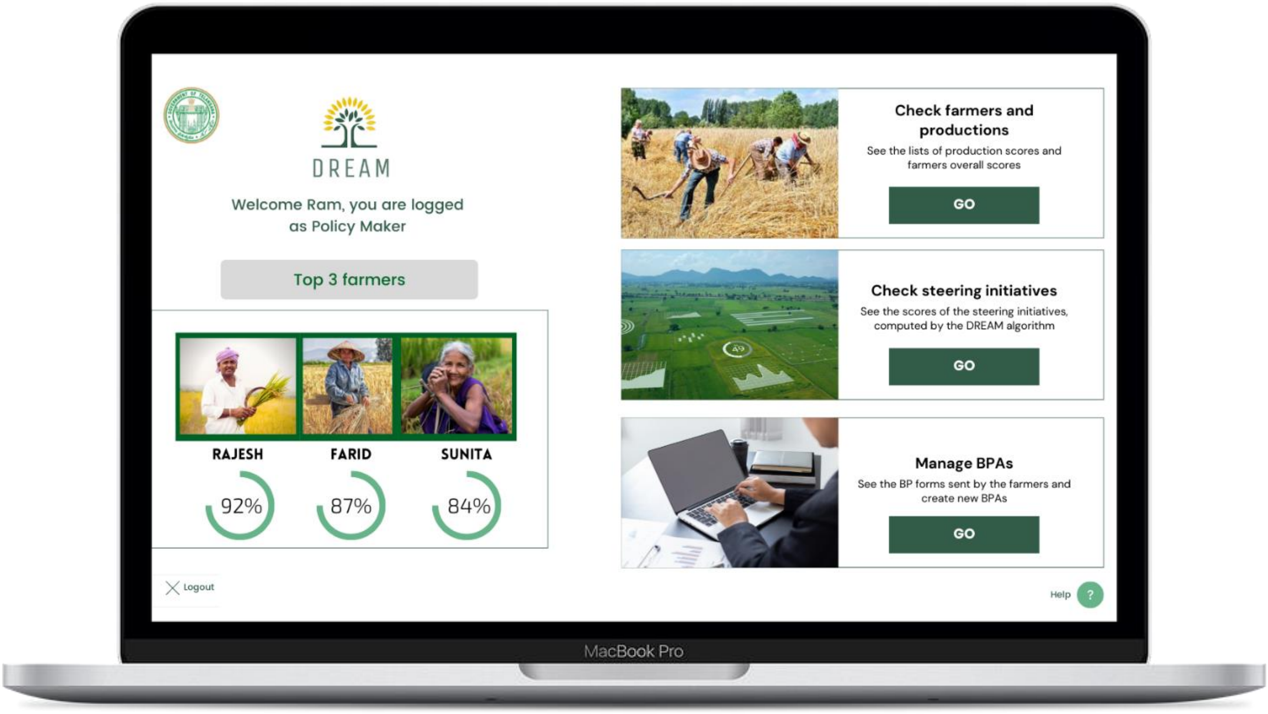


Figure 3.5 - TPM main page

3.1.2 Hardware Interfaces

In order to operate correctly, the system needs every user for which this app is meant to be equipped with either a smartphone/tablet or a computer.

The use with a computer is meant for TPMs while agronomists and farmers should have a smartphone or a tablet to use the mobile application.

3.1.3 Software Interfaces

The system interfaces map with the Telangana website in order to provide users updated weather forecasts.

3.1.4 Communication Interfaces

All communications within the Dream system are made via HTTPS.

3.2 Functional requirements

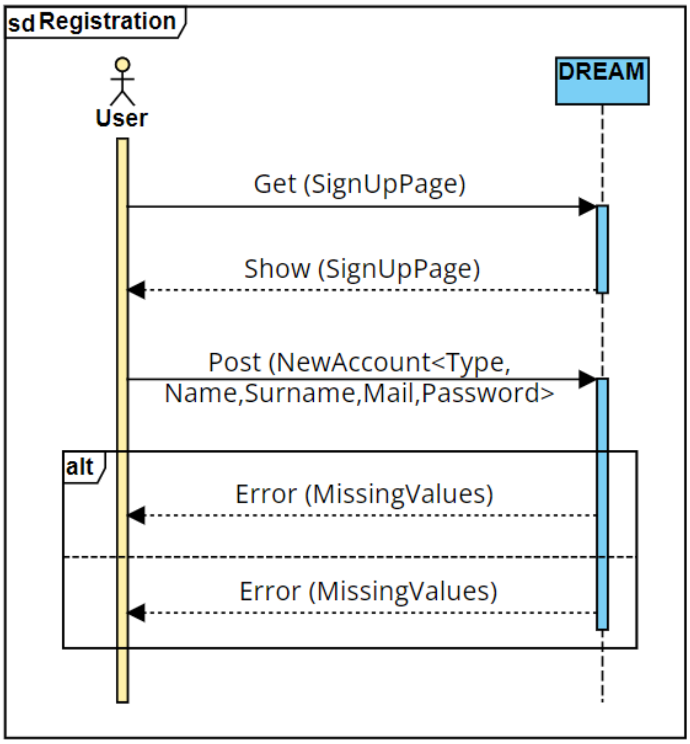
Rx	Description
R1	The system shall allow an app customer to login
R2	The system shall allow an app customer to logout
R37	The system shall allow unregistered users to sign up using their ID
R38	The system shall allow to set a user profile type
Telangana Policy Maker	
R3	The system shall allow a TPM to visualize an ordered list with the farmers and their performances
R4	The system shall allow a TPM to update the overall score of a farmer
R5	The system shall allow a TPM to visualize a list with the production performances
R6	The system shall allow a TPM to send a form to the top performing farmers in order to collect their best practices
R7	The system shall allow a TPM to notify an agronomist with the worst performing farmers who need to be visited as soon as possible
R8	The system shall allow a TPM to visualize an ordered list with steering initiatives carried out by agronomists associated with their performances
R9	When a farmer inserts the last update and closes a production, his/her production performance is computed by the algorithm
R10	When a production performance is computed, the ordered list of the performances is updated
R11	The system shall allow a TPM to receive the best practices
R12	The system shall allow a TPM to create BPAs
R13	The system shall send out the BPAs to all farmers
Farmer	

R14	The system shall allow farmers to insert their location
R15	The system shall allow farmers to fill in the form sent by TPM with their best practices
R16	The system shall allow farmers to visualize weather forecasts
R17	The system shall compute personalized suggestions based on the designed algorithm.
R18	The system shall allow farmers to visualize personalized suggestions
R19	The system shall allow farmers to insert data about a new production
R20	The system shall allow farmers to insert information on problems they face during the production
R21	The system shall allow farmers to close a production when they finish harvesting
R22	The system shall allow farmers to create help requests
R23	The system shall allow farmers to create discussion forums
R24	The system shall allow farmers to send messages in existing discussion forums
Agronomist	
R25	The system shall allow an agronomist to insert the area he is responsible of
R26	The system shall allow an agronomist to receive information about requests for help
R27	The system shall allow an agronomist to answer to requests for help
R28	The system shall allow an agronomist to visualize data concerning weather forecasts in the area
R29	The system shall allow an agronomist to visualize the best performing farmers in the area
R30	The system shall allow an agronomist to create a daily plan to visit farms in the area
R31	The system shall allow an agronomist to update the daily plan
R32	The system shall allow an agronomist to visualize highlighted in red the farms that have not been visited at least twice during the year
R33	The system shall allow an agronomist to visualize with a “!” the most under-performing farms
R34	The system shall allow an agronomist to confirm the execution of the daily plan
R35	The system shall allow an agronomist to specify deviations from a daily plan previously created
R36	The system shall allow an agronomist to insert a place, a period and a description for a new steering initiative

3.2.1 Use cases description

Use cases capture functional requirements of a system from the users' perspective.

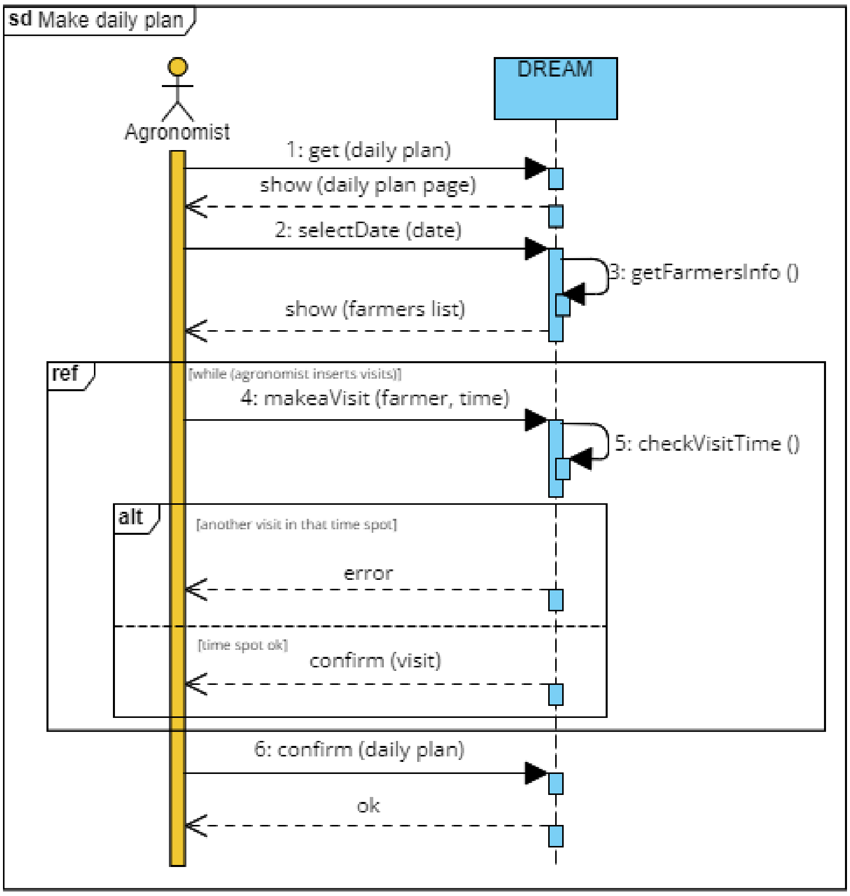
Name	Registration
ID	UC. 1
Actors	Agronomist, Farmer
Entry Conditions	• The user has access to the application on his device
Flow of Events	1. The user clicks the "sign-up" button 2. The user inserts his name, surname, area, ID code, email password
Exit Conditions	All inserted data is valid and the user clicks "confirm"
Exceptions	• A field is empty or not correctly compiled • the email is already associated to another account



Sequence Diagram UC. 1

Name	Make daily plan
ID	UC. 2
Actors	Agronomist
Entry Conditions	• The agronomist has access to the application on his device • The agronomist wants to schedule a daily plan

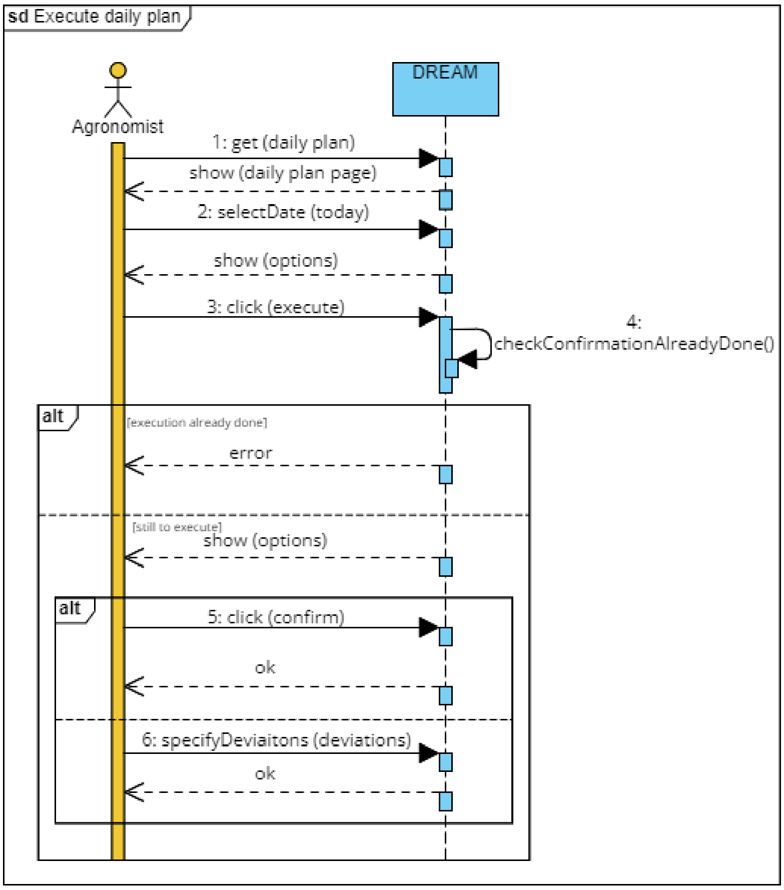
Flow of Events	1. The agronomist clicks on "Daily plan" on the homepage 2. The agronomist selects a date 3. The agronomist visualizes the farmers in the area (he sees highlighted in red the ones who still have not been visited twice, and those who are under-performing are marked with a "!") 4. The agronomist selects the farmers he wants to visit, and the approximate time of the visits 5. The agronomist confirms the daily plan
Exit Conditions	The agronomist has created a daily plan
Exceptions	If the agronomist selects the same time for two different farm visits, an error is prompted in the application



Sequence Diagram UC.2

Name	Execute daily plan
ID	UC. 3
Actors	Agronomist
Entry Conditions	• The agronomist has access to the application on his device • It is the day specified in the daily plan to execute

Flow of Events	1. The agronomist clicks on "Daily Plan" on the homepage 2. The agronomist selects "today" in the app and clicks on "execute" 3. The system shows the actor two possibilities, "confirm" or "specify deviations" 4. The agronomist chooses the option that suits his workday and, in case of any deviations, he specifies them
Exit Conditions	The agronomist has confirmed the execution of the daily plan
Exceptions	If the agronomist has already confirmed the execution of the daily plan, the application prompts an error if the agronomist clicks again on "execute" If the day ends and the daily plan is not confirmed, an alert will be prompted when the agronomist opens the application, asking him/her to confirm it before modifying any other daily plan



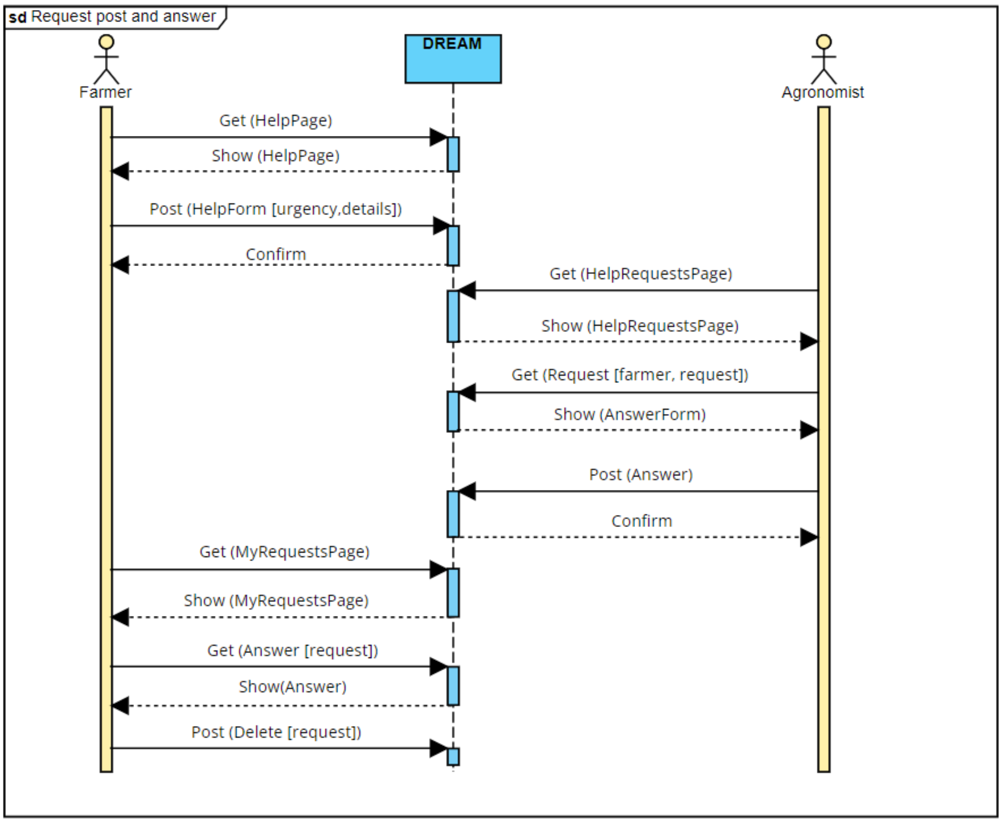
Sequence Diagram UC. 3

Name	Answer help request
ID	UC. 4
Actors	Agronomist
Entry Conditions	• The agronomist has access to the application on his device • There are open help requests

Flow of Events	1. The agronomist clicks on “Help requests” 2. The agronomist selects the one he/she wants to answer to 3. A text field is opened 4. The agronomist answers providing a solution to the problem or proposing a visit to the farm as soon as possible to understand better the problem
Exit Conditions	Answer correctly submitted
Exceptions	The help request is closed while he is answering

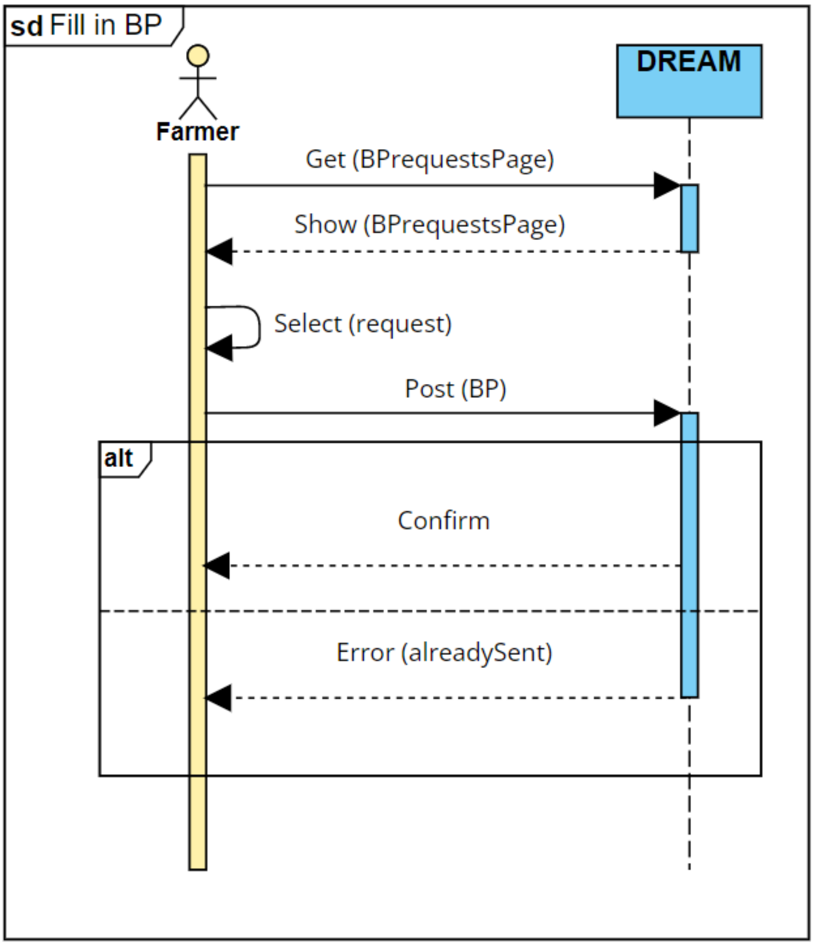
Name	Create a new help request
ID	UC. 5
Actors	Farmer
Entry Conditions	• The farmer has access to the application on his device
Flow of Events	1. The farmer clicks on the “Manage requests” button 2. The farmer clicks on the “Create new request” button 3. The farmer inserts a title which shortly describes the problem 4. The farmer inserts a precise description of the problem faced 5. The farmer can insert an image to attach to the description by clicking on the “Insert an image” button 6. The farmer clicks on the “Send” button
Exit Conditions	The new help request is correctly inserted in the system
Exceptions	The farmer doesn’t insert all the information correctly, so the system shows an error message

Name	Close a help request
ID	UC. 6
Actors	Farmer
Entry Conditions	• The farmer has access to the application on his device • There is at least one answer to the help request
Flow of Events	1. The farmer clicks on the “Manage requests” button 2. The farmer clicks on the “Visualize open requests” 3. The farmer selects the request he/she wants to visualize from the displayed list of all the open requests 4. The farmer visualizes the answers to the request 5. The farmer clicks on the “Close request” button, if he/she is satisfied by the answer/s
Exit Conditions	The help request is now closed so nobody can answer it anymore



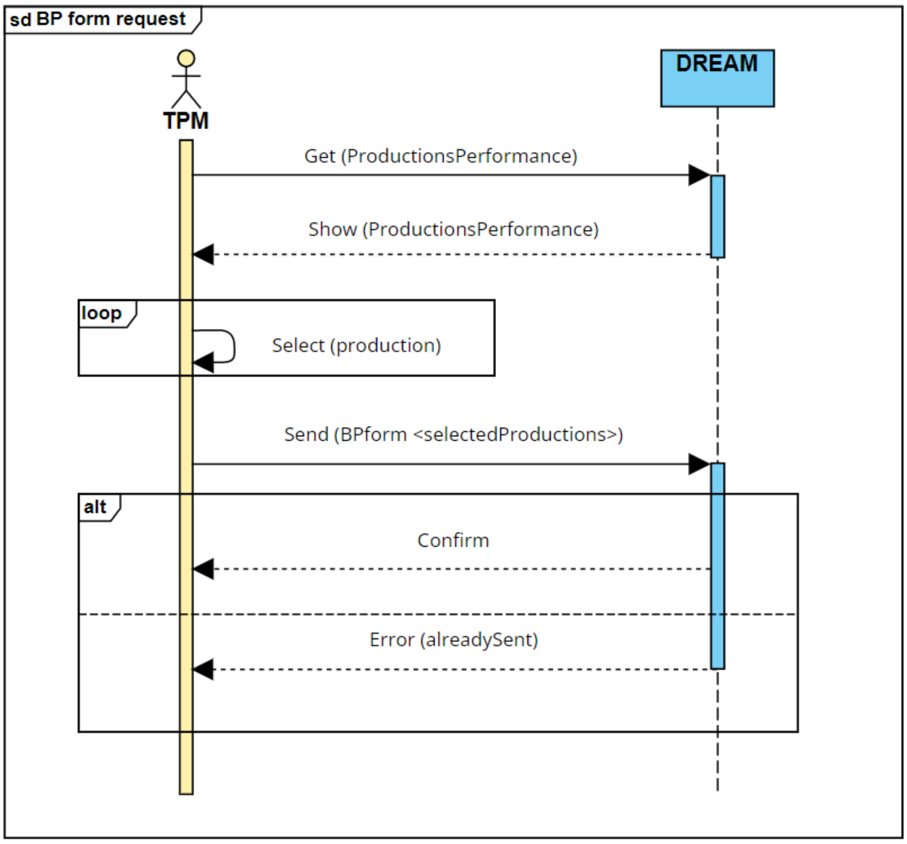
Sequence Diagram UC.4-UC.5-UC.6

Name	Fill in BP form
ID	UC. 7
Actors	Farmer
Entry Conditions	• The Farmer has access to the application on his device • The farmer has at least one BP request
Flow of Events	1. The farmer opens his list of BP requests from the Home Page 2. The farmer selects the one he wants to answer to 3. The farmer fills in the form 4. The farmer confirms and submits
Exit Conditions	BP form correctly submitted
Exceptions	The farmer sends an empty answer



Sequence Diagram UC.7

Name	Best practice request
ID	UC. 8
Actors	TPM
Entry Conditions	• The TPM has access to the application on his device • At least one farmer has closed a production
Flow of Events	1. The TPM opens a list with the farmers production performances 2. The TPM selects those who in his/her opinion have performed particularly well and can provide useful suggestions for others 3. The TPM sends a form to them to be filled with their best practices
Exit Conditions	BP forms are correctly sent to farmers
Exceptions	The form has already been sent to the selected farmers for those productions

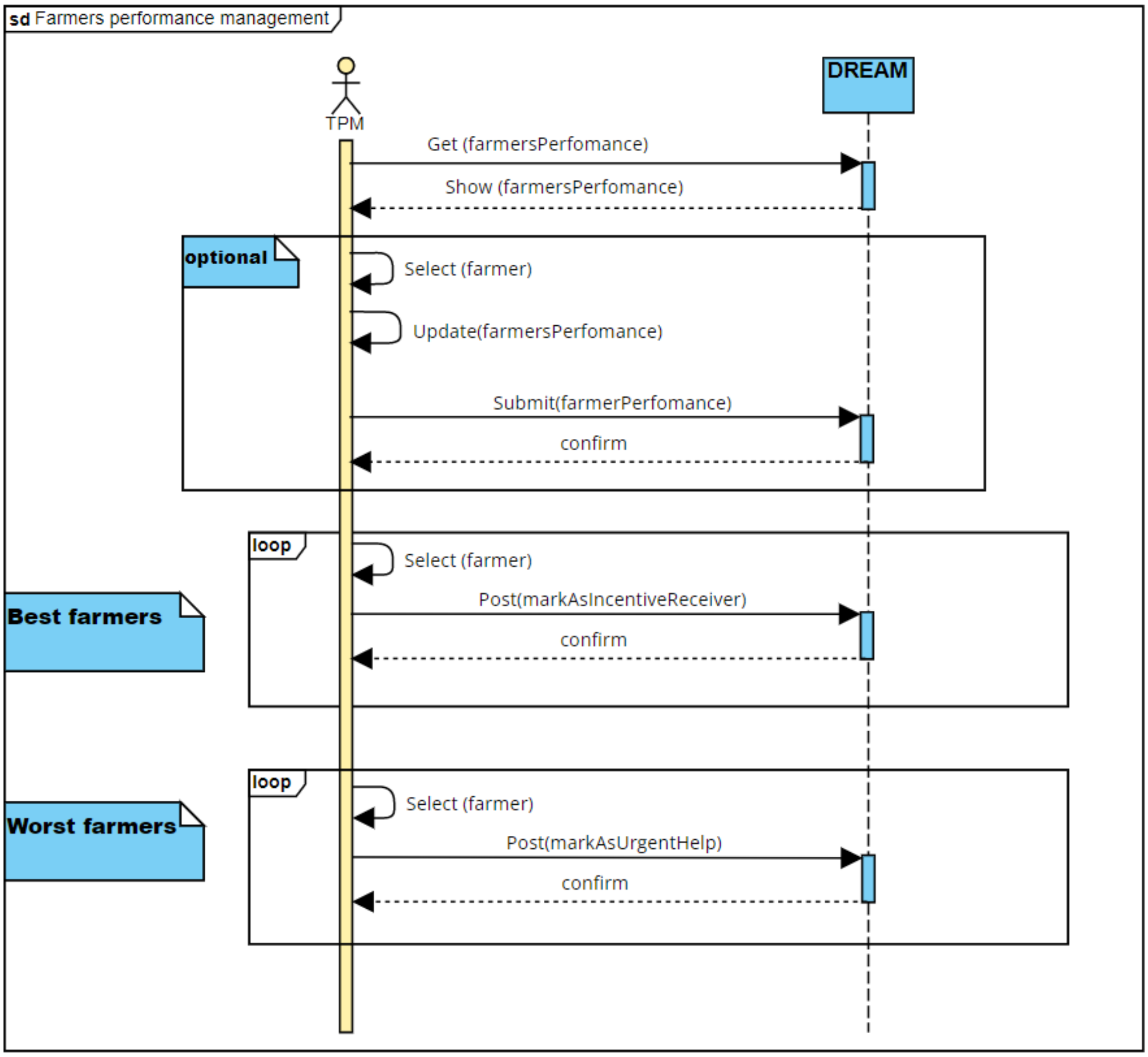


Sequence Diagram UC.8

Name	Help for worst Farmers
ID	UC. 9
Actors	TPM
Entry Conditions	• The TPM has access to the application on his device • At least one farmer has closed a production
Flow of Events	1. The TPM opens a list with the farmers' performances 2. The TPM selects those who, in his/her opinion, need urgent help
Exit Conditions	The TPM confirms his/her choice

Name	Farmers overall performance update
ID	UC. 10
Actors	TPM
Entry Conditions	• The TPM has access to the application on his device • At least one farmer has closed a production

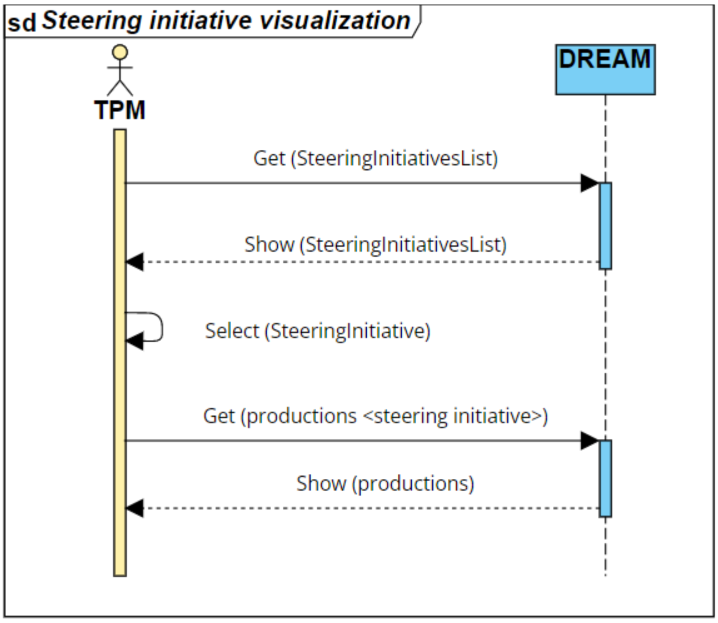
Flow of Events	1. The TPM opens farmers performance list 2. The TPM selects one farmer 3. The TPM updates his score on the basis of the eventually particular weather conditions and other relevant factors
Exit Conditions	The TPM confirms his/her choice



Sequence Diagram UC.9 - UC.10

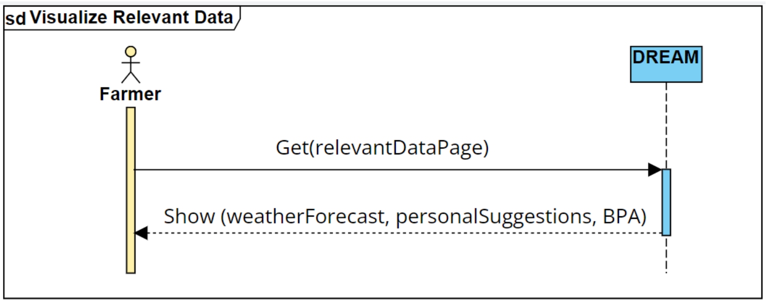
Name	Visualize Steering Initiatives
ID	UC. 11
Actors	TPM
Entry Conditions	• The TPM has access to the application on his device • At least one steering initiative has been brought by an agronomist • At least one farmer has closed a production linked to a steering initiative

Flow of Events	1. The TPM opens a list with the steering initiatives performances computed by an algorithm 2. The TPM clicks on the desired steering initiative 3. A list of production performances is shown
Exit Conditions	The TPM closes the list window



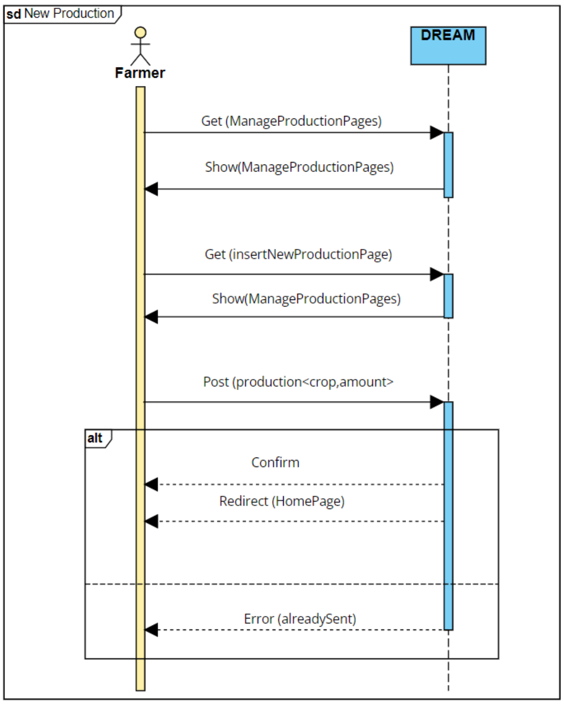
Sequence Diagram UC.11

Name	Visualize relevant data
ID	UC. 12
Actors	Farmer
Entry Conditions	• The farmer has access to the application on his device
Flow of Events	1. The farmer clicks on the “visualize data” button 2. Weather forecasts, personalized suggestions are shown. Also a list of BPAs is shown and the farmer can open one of them or search a BPA based on some topics. 3. The farmer clicks on the “Back” button
Exit Conditions	The farmer is back to the main page



Sequence Diagram UC.12

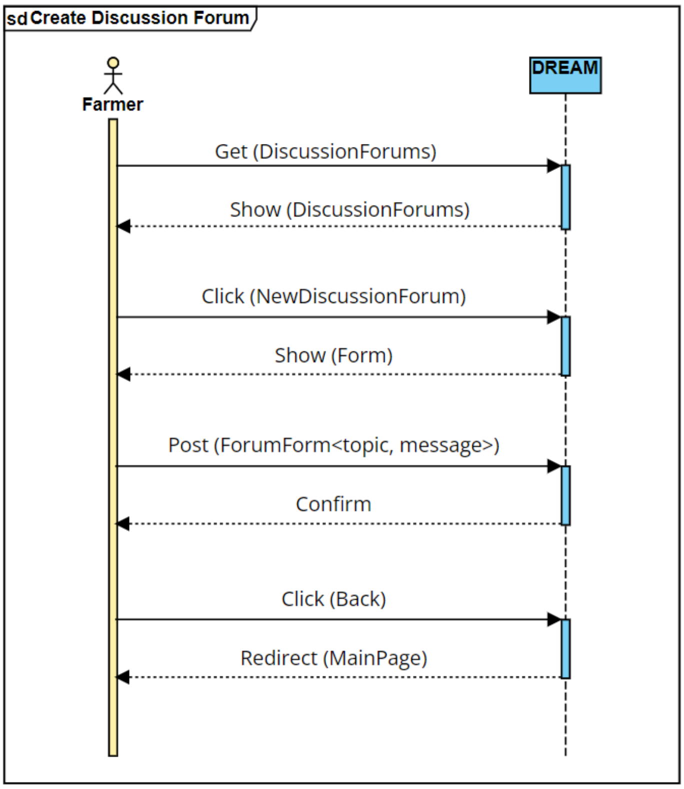
Name	Insert a new production
ID	UC. 13
Actors	Farmer
Entry Conditions	• The farmer has access to the application on his device
Flow of Events	1. The farmer clicks on the "Manage productions" button 2. The farmer clicks on the "Insert new production" button 3. The farmer inserts in a table the type of crops in the production (chosen from a list), the expected produced amount and the date of start of the production. 4. The farmer clicks on the "Confirm" button 5. The system confirms the insertion and redirects the farmer to the main page
Exit Conditions	The new production is correctly inserted in the system
Exceptions	The farmer doesn't insert all the information correctly, so the system shows an error message A type of crop is not in the list. The farmer selects "other" in the list and inserts the type of crop manually.



Sequence Diagram UC.13

Name	Create a new discussion forum
ID	UC. 14
Actors	Farmer
Entry Conditions	• The farmer has access to the application on his device
Flow of Events	1. The farmer clicks on the "Discussion forums" button 2. The farmer clicks on the "Create new discussion forum" button 3. The farmer inserts the topic of the discussion forum 4. The farmer writes the first question in the forum 5. The farmer clicks on the "Send"

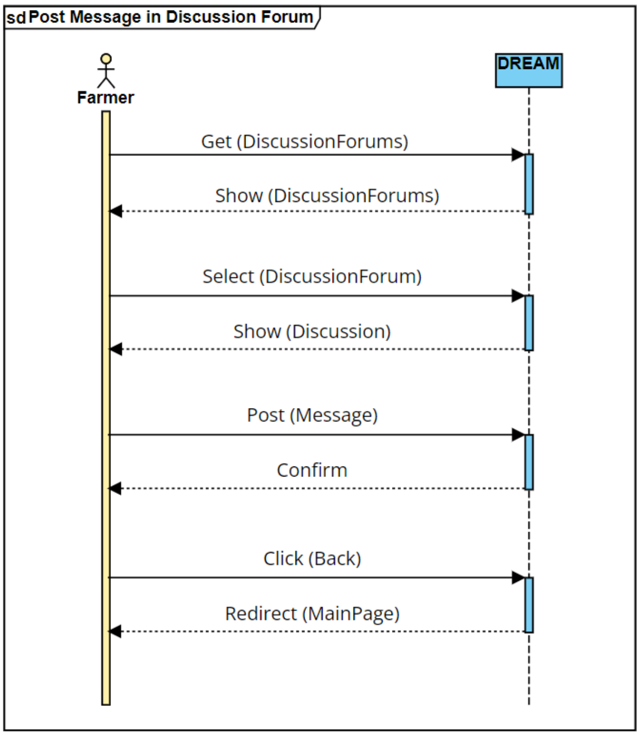
	6. The farmer clicks on the "Back" button
Exit Conditions	The new discussion forum is correctly inserted in the system
Exceptions	The farmer doesn't insert all the information correctly, so the system shows an error message



Sequence Diagram UC.14

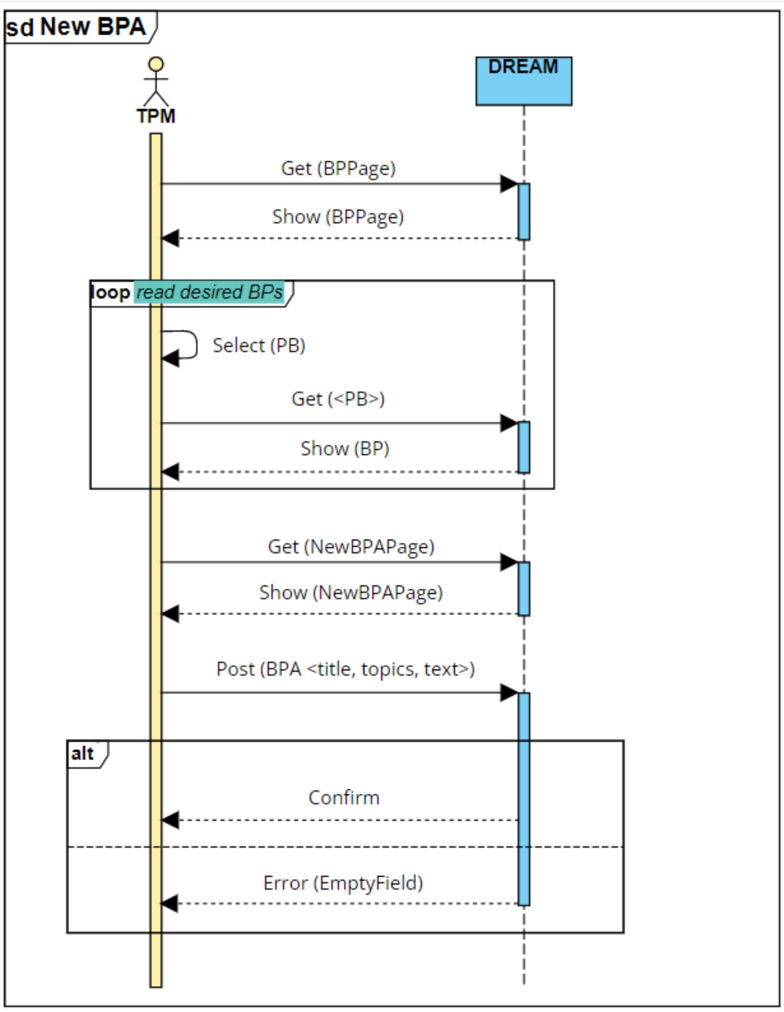
Name	Message in a discussion forum
ID	UC. 15
Actors	Farmer
Entry Conditions	• The farmer has access to the application on his device
Flow of Events	1. The farmer clicks on the "Discussion forums" button 2. The farmer selects the forum to open from a list of discussion forums shown by the system (or he/she can search the topic of a forum from a search bar) 3. The farmer clicks on the "Add a new message" button 4. The farmer inserts the message related to the topic of the forum 5. The farmer clicks on the "Send" button
Exit Conditions	The message is correctly inserted in the system

Exceptions	The farmer doesn't insert all the information correctly, so the system shows an error message



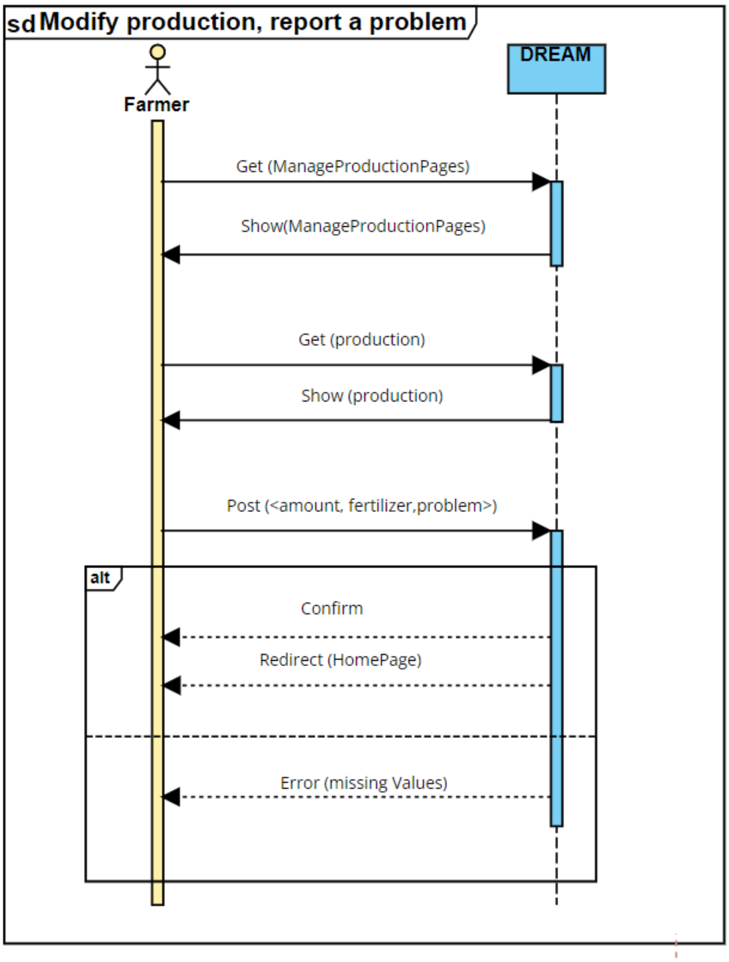
Sequence Diagram UC.15

Name	Write BPA
ID	UC. 16
Actors	TPM
Entry Conditions	• The TPM has access to the application on his device • The TPM has already logged in and is on the main page
Flow of Events	1. The TPM clicks on "Best Practices" section where he/she can find the best practices forms sent by farmers 2. The TPM selects and opens one of them 3. Outside the app, The TPM writes down the useful information found inside the best practice form 4. The TPM repeats 2. 3. for all the best practices he/she wants to read 5. The TPM clicks on "New BPA" 6. The TPM fills the form with Title, topics and text based on the gathered information 7. The TPM clicks "send"
Exit Conditions	BPA correctly inserted in the system
Exceptions	Not all fields are filled in correctly



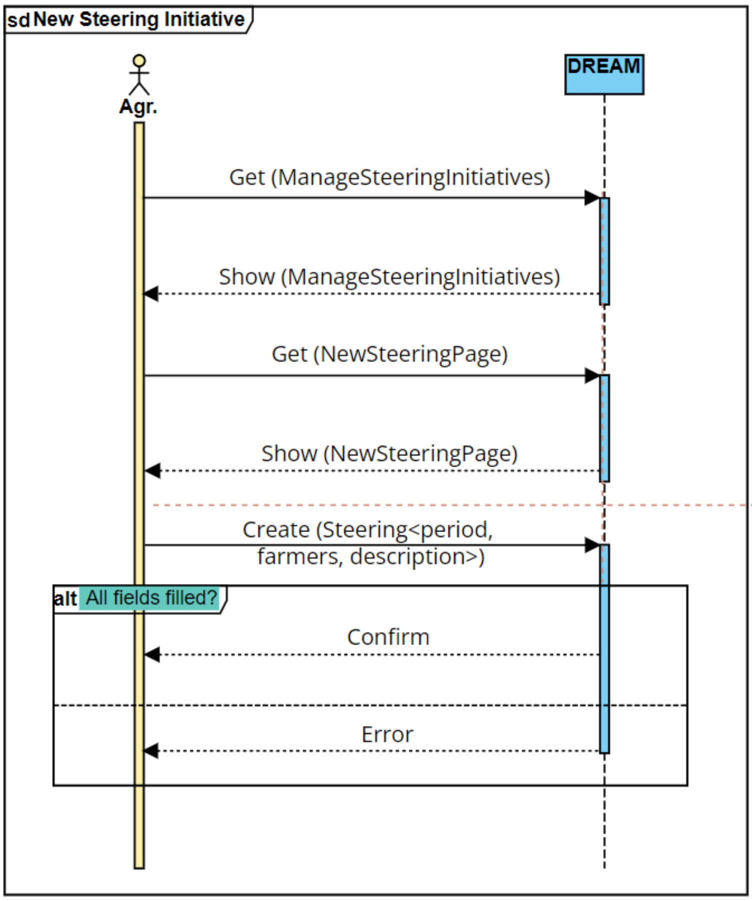
Sequence Diagram UC.16

Name	Update and insert problems in a production
ID	UC. 17
Actors	Farmer
Entry Conditions	• The farmer has access to the application on his device • The farmer has uploaded at least one production
Flow of Events	1. The farmer clicks on the "Manage production" button 2. The farmer selects one production from those shown 3. If the production is not closed, the farmer clicks on the "modify" button 4. The farmer modifies quantity produced, fertilizer so on and so forth 5. Eventually, the farmer adds problems occurred during the production
Exit Conditions	Modifications and problems are correctly submitted
Exceptions	The farmer doesn't insert all the information correctly, so the system shows an error message



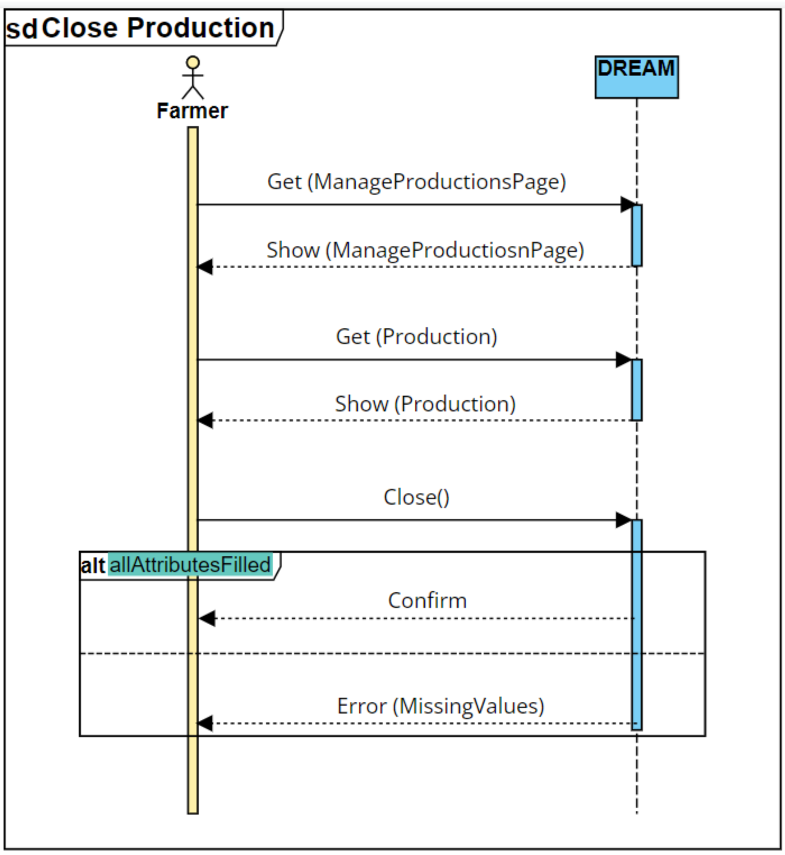
Sequence Diagram UC.17

Name	Create steering initiatives
ID	UC. 18
Actors	Agronomist
Entry Conditions	• The agronomist has access to the application on his device
Flow of Events	1. The agronomist clicks on "new steering initiatives" 2. The agronomist compiles the fields "text" and "period of application" 3. The agronomist selects farmers that have to follow the steering initiative
Exit Conditions	The agronomist confirms and the steering initiative is correctly submitted
Exceptions	• Not all fields are compiled • No farmer is selected, so nobody will be affected by the steering initiative



Sequence Diagram UC.18

Name	Close Production
ID	UC. 19
Actors	Farmer
Entry Conditions	• The farmer has access to the application on his device
Flow of Events	1. The farmer clicks on "manage productions" 2. The farmer clicks on the production he wants to close from the ones shown 3. The farmer clicks on the "close" button
Exit Conditions	The production is filled with all its attributes and is correctly signed as closed
Exceptions	• Not all fields are compiled



Sequence Diagram UC.19

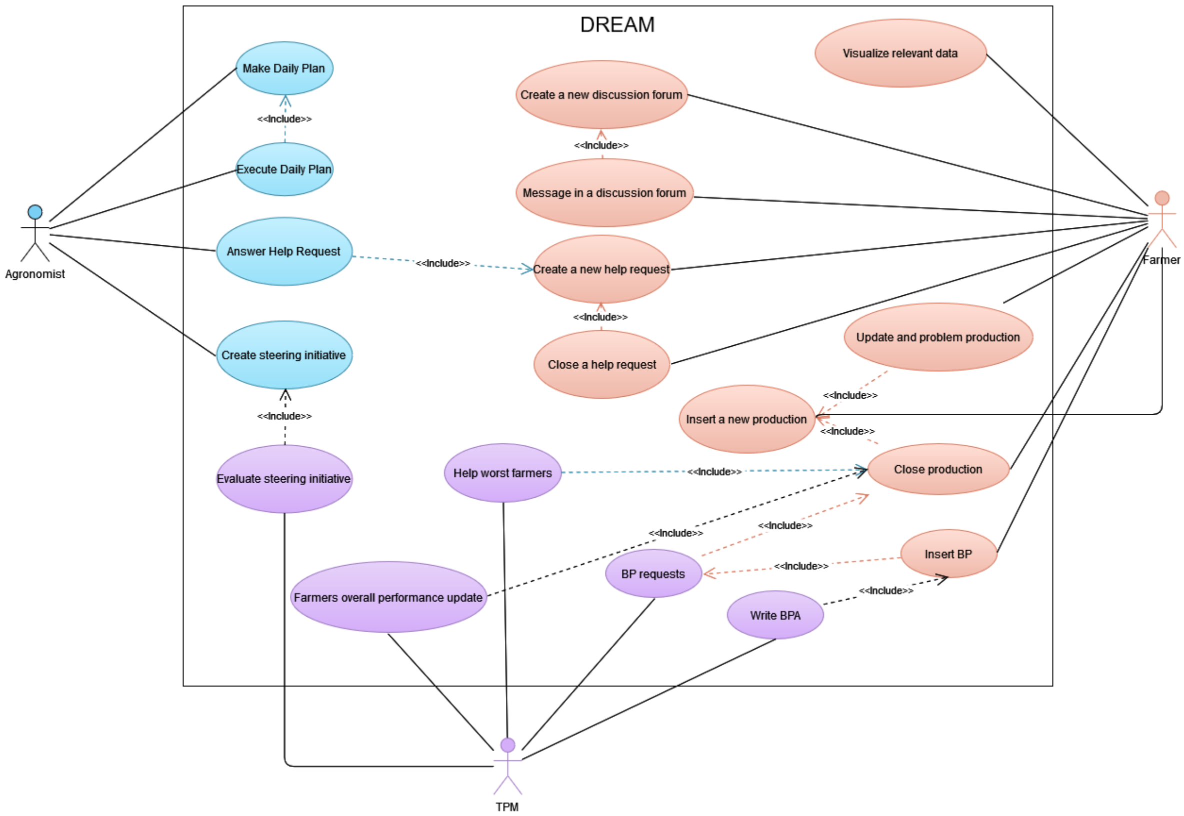


Figure 3.6 - Logged users use case diagram

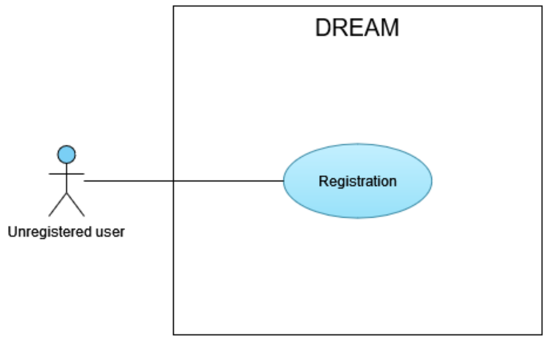


Figure 3.7 - Unregistered user use case diagram

3.2.2 Mapping on requirements

G1	Allow TPMs to have an overview of farmers' performances
R3	The system shall allow a TPM to visualize an ordered list with the farmers and their performances
R4	The system shall allow a TPM to update the overall score of a farmer
R5	The system shall allow a TPM to visualize a list with the production performances
R7	The system shall allow a TPM to notify an agronomist with the worst performing farmers who need to be visited as soon as possible
R9	When a farmer inserts the last update and closes a production, his/her production performance is computed by the algorithm
R10	When a production performance is computed, the ordered list of the performances is updated
R19	The system shall allow farmers to insert data about a new production
R20	The system shall allow farmers to insert information on problems they face during the production
R21	The system shall allow farmers to close a production when they finish harvesting
Brief Description: A TPM needs to be able to visualize the list of all farmers with a score for each production made by each farmer, computed by an algorithm.	

G2	Allow TPMs to visualize the result of steering initiatives
R8	The system shall allow a TPM to visualize an ordered list with steering initiatives carried out by agronomists associated with their performances
R36	The system shall allow an agronomist to insert a place, a period and a description for a new steering initiative
Brief Description:	

A TPM needs to be able to visualize the score of steering initiatives to understand whether they were useful or not. This score is computed by the application.

G3	Allow agronomists to visualize all relevant data concerning their area
R25	The system shall allow an agronomist to insert the area he is responsible of
R28	The system shall allow an agronomist to visualize data concerning weather forecasts in the area
R29	The system shall allow an agronomist to visualize the best performing farmers in the area
R33	The system shall allow an agronomist to visualize with a “!” the most under-performing farms
Brief Description: An agronomist needs to be able to visualize in the application all data that can be useful for his job, so the system shall show him all these data	

G4	Allow agronomists to answer farmers’ help requests
R26	The system shall allow an agronomist to receive information about requests for help
R27	The system shall allow an agronomist to answer to requests for help
Brief Description: A farmer should be able to ask for help anytime he needs it and the system shall take care of this scenario; an agronomist should be able to receive these requests and to answer them.	

G5	Allow agronomists to manage a daily plan to visit farms
R30	The system shall allow an agronomist to create a daily plan to visit farms in the area
R31	The system shall allow an agronomist to update the daily plan
R32	The system shall allow an agronomist to visualize highlighted in red the farms that have not been visited at least twice during the year
R33	The system shall allow an agronomist to visualize with a “!” the most under-performing farms
R34	The system shall allow an agronomist to confirm the execution of the daily plan

R35	The system shall allow an agronomist to specify deviations from a daily plan previously created
Brief Description: To make a daily plan is the key feature of Dream app in the version for agronomists. Agronomists should be able to prepare, update and confirm a daily plan to visit farms and the system should ease them on this duty.	

G6	Allow farmers to create discussion forums with the other farmers
R23	The system shall allow farmers to create discussion forums
R24	The system shall allow farmers to send messages in existing discussion forums
Brief Description: Farmers should be able to discuss in a forum with other farmers, exchanging opinions about crops, fertilizers and every agriculture-related topic. Within this forum the system shall permit them to create discussion, post comments and send messages.	

G7	Allow farmers to visualize every relevant data they need for their work
R16	The system shall allow farmers to visualize weather forecasts
R17	The system shall compute personalized suggestions based on the designed algorithm.
R18	The system shall allow farmers to visualize personalized suggestions
Brief Description: A farmer needs to be able to visualize in the application all data that can ease his job, so the system shall show him all these data.	

G8	Allow farmers to ask for help and suggestions
R22	The system shall allow farmers to create help requests
Brief Description: A farmer who wants to ask for help within the app should be facilitated to do so	

G9	Allow every user to have a proper account for the type of user
R1	The system shall allow an app customer to login

R2	The system shall allow an app customer to logout
R14	The system shall allow farmers to insert their location
R25	The system shall allow an agronomist to insert the area he is responsible of
R37	The system shall allow unregistered users to sign up using their ID
R38	The system shall allow to set a user profile type
Brief Description: Dream app should permit every user to use all the provided functions for his role. In order to do so every user should be able to login, logout and sign up, and furthermore to modify all the desired profile settings.	

G10	Allow TPMs to create BPAs
R6	The system shall allow a TPM to send a form to the top performing farmers in order to collect their best practices
R11	The system shall allow a TPM to receive the best practices
R12	The system shall allow a TPM to create BPAs
R13	The system shall send out the BPAs to all farmers
R15	The system shall allow farmers to fill in the form sent by TPM with their best practices
Brief Description: TMPs should have the possibility to ask good farmers their best practices and receive them. Then the app should make possible for them to create articles that explain these best practices	

G11	Allow agronomists to create steering initiatives
R8	The system shall allow a TPM to visualize an ordered list with steering initiatives carried out by agronomists associated with their performances
R36	The system shall allow an agronomist to insert a place, a period and a description for a new steering initiative
Brief Description: The system should ease agronomists to create steering initiatives and permit them to register these actions in the proper way. TPM will then evaluate the result of the initiatives after a certain period.	

3.2.3 Requirements mapping matrix

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11
R1									X		
R2									X		
R3	X										
R4	X										
R5	X										
R6										X	
R7	X										
R8		X									X
R9	X										
R10	X										
R11										X	
R12										X	
R13										X	
R14									X	X	
R15										X	
R16							X				
R17							X				
R18							X				
R19	X										
R20	X										
R21	X										
R22								X			
R23						X					
R24						X					
R25			X						X		
R26				X							
R27				X							
R28			X								
R29			X								
R30					X						
R31					X						
R32					X						
R33			X		X						
R34					X						
R35					X						
R36		X									X
R37									X		
R38									X		

3.3 Performance requirements

This section specifies requirements placed on the software or on human interaction with the software as a whole.
The majority of the computation will take place on the servers of the system. Therefore, the user apps shall be lightweight, not load too much the environments in which they are installed on and occupy little memory on the personal devices.
The system must be able to handle concurrent requests (in the order of tens of thousands) especially during peak times, for example in harvest season or after some extreme weather events.

3.4 Design constraints

3.4.1 Standard compliance

The system will store all the data submitted to it in a standardized form. In this way it will be easier to catalogue, to retrieve, to run queries on data and to detect anomalies.
To provide a better user experience to farmers it is recommended that the Dream system implementation follows stateless protocols.
It is crucial to design modules properly so that ease of use, security and performance will remain the core factors of the system.

3.4.2 Hardware limitations

The client application requires a device able to connect to the internet in order to send and receive data. Farmers and agronomists are meant to use the application through a mobile device (either a smartphone or a tablet), so it needs to have enough memory space for the executable, while TPMs connect through a website. The server needs an internet connection and enough memory for the executable.

3.4.3 Other constraints

Regulatory policies have to be considered for the interaction between Dream and users. Further information will be provided in section 3.6.1.

3.5 Software system attributes

3.5.1 Reliability

In order to guarantee continuity, services are required to be fault tolerant. Error handling and fault containment mechanisms to prevent error propagation and data loss are to be arranged.

3.5.2 Availability

Dream users should be able to access the application anytime, in any case a limited period of downtime is admissible, and preferably at night. A 95% uptime is plausible so the MTTR must be contained in ~18 days a year, since agronomists are the only users who need to use the application daily, but they are not expected to use it more than an hour per day; while other users are expected to use the application a few times a week. In order to comply with the required uptime, the system must be supported by an appropriate infrastructure composed of redundant services.

3.5.3 Security

The system shall perform role-based access control (RBAC), a policy-neutral access-control mechanism defined around roles and privileges. The components of RBAC such as role-permissions, user-role and role-role relationships make it simple to perform user assignments and authorizations.
Common security standards should be implemented in the management and storage of private data. Dream application is not involved in any payment process, so no critical information is stored nor used.

3.5.4 Maintainability

The system shall be characterized by scalable and reusable modules which will be easier to maintain and replace in case of failure. Ordinary maintenance, for bug fixes and improvements will be scheduled during night time, when the user traffic is minimal.
The implementation must be written in a widely used programming language and must ensure a high level of maintainability. Code must be written following standards with a high level of abstraction without hard-code as well and must be highly commented on in any aspect. Code must provide a testing routine that covers the majority of the code.
The core aspects of maintainability and modularity will be addressed in the design document.

3.5.5 Portability

The web platform will be accessible by any web browser. The mobile application for users must be accessible by everyone, therefore it must be developed for the major mobile OSes. Since mobile apps do not demand special functions, a non native approach can be adopted to fasten the development process. The server side has no major requirements for portability.

3.5.6 Usability

The mobile app and the web platform of the system will be designed to be concise and user-friendly, with a graphical interface to help users identify immediately their needs on the screen. It is expected that at least 90% of the users will be able to complete tasks without requiring assistance. Anyway, courses and tutorials, especially for farmers, will be included in the launch of Dreams.

3.6 Other requirements

3.6.1 Privacy requirements

The system shall ensure that the collection and transmission of personal data is handled in accordance with user's expectation and regulations.
The system shall block unauthorized access to implicit information and encrypt data transmission.

4 Formal analysis using ALLOY

This section is dedicated to the Alloy model of the DREAM software. In this model we described the main components of the system together with their relationships and their (most important) properties. In particular, the purpose of this section is to validate the consistency of the world generated by the union of the previous assumptions and requirements, with the help of this specification language. In order to improve readability, we introduced some minor assumptions that we are sure will not interfere with our goal.
For this purpose, we kept only two figures to refer to "year", and the score given to production is comprehended in a range from 0 to 30. Dates only refer to the interval from January 2020 to December 2021.

4.1 Alloy Code

```
open util/integer as intgr

---- Signatures ----

sig Date {
        day: Int,
        month: Int,
        year: Int
}{
        day>1
        day<=30
        month>=1
        month<=12
        year>=20
        year<=21
}

one sig CurrentDate extends Date{}  //for alloy purporses
{
        day=30
        month=12
        year=21
}

sig Time {
        hour: Int,
} {
        hour >= 0
        hour <= 23
}

sig DateTime {
        date: Date,
        time: Time
}
```

```
sig Area {}

sig Usercode {}

sig Password {}

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

abstract sig User{
	usercode: Usercode,
	password: Password
}

sig Farmer extends User{
	productions: set Production,
	area: Area
}

sig PolicyMaker extends User{}

sig Agronomist extends User{
	dailyPlans: set DailyPlan,
	area: Area
}

sig FarmerVisit {
	farmer: Farmer,
	dateTime: DateTime
}

sig DailyPlan {
	date: Date,
	farmerVisits: set  FarmerVisit,
	confirmed: Bool
}

sig Production {
	startDate: Date,
	crop: Crop,
	harvestFinishDate: Date,
	evaluated: Bool,
	score: Int
}{
	score>=-1
	score<=30
}
```

```
sig Crop{}

sig Request {
	date: Date,
	farmer: Farmer
}

sig Answer {
	date: Date,
	request: Request,
	agronomist: Agronomist
}

sig BestPractice{
	production: Production
}

sig Bpa{
	bestPractices: set BestPractice
}

---- Functions ----

//retrieve alll FarmerVisits
fun retrieveFarmerVisits [f: Farmer, y: Int]: FarmerVisit {
	{v: FarmerVisit | v.farmer = f and v.dateTime.date.year=y }
}

//calculate difference in days between two dates
fun differenceBetweenDates [d1: Date, d2: Date]: Int {
	mul[(d1.year-d2.year),365]+mul[(d1.month-d2.month),30]+(d1.day-d2.day)
}

---- Predicates ----

pred SameDate [d1:Date, d2:Date] {
	d1.year=d2.year and d1.month=d2.month and d1.day=d2.day
}

pred datePrecedesAnotherDate [d1: Date, d2: Date] {
	d1.year<d2.year or
	(d1.year=d2.year and d1.month<d2.month) or
	(d1.year=d2.year and d1.month=d2.month and d1.day<d2.day)
}

//determinate whether two productions overlap
pred dateNotInBetweenPeriod [p1: Production, p2: Production] {
```

```
		datePrecedesAnotherDate[p1.harvestFinishDate, p2.startDate] or datePrecedesAnotherDate[p2.harvestFinishDate, p1.startDate]
}

pred sameCrop [p1: Production, p2: Production] {
		p1.crop=p2.crop
}

---- Facts ----

//a username is unique
fact uniqueUsername {
		all p:Usercode | one u:User | p in u.usercode
}

//all productions are associated to a farmer
fact ProductionLinkedToFarmer {
		all p: Production | one f: Farmer | p in f.productions
}

//there cannot be two productions of the same crop in the same period for the same farmer
fact noSameCropSamePeriod {
		all disj p1,p2: Production, f:Farmer | (p1 in f.productions and p2 in f.productions and sameCrop[p1,p2]) implies dateNot-InBetweenPeriod[p1,p2]
}

//one agronomist per area
fact oneAgroOneArea {
		all a: Area | one agro: Agronomist | a in agro.area
}

//no more than one daily plan per day
fact notTwoDailyPlans {
		all disj d1,d2: DailyPlan, a:Agronomist | (d1 in a.dailyPlans and d2 in a.dailyPlans) implies !SameDate[d1.date, d2.date]
}

// a daily plan is always linked to an agronomist
fact notAloneDailyPlan {
		all d:DailyPlan | one a:Agronomist | d in a.dailyPlans
}

//a daily plan always has some farmerVisits
fact noDailyPlanwithNoVisits {
		all d:DailyPlan | some f:FarmerVisit | f in d.farmerVisits
}

//a farmerVisit is always linked to a daily plan
fact noDailyNoVisits {
		all f:FarmerVisit | one d:DailyPlan | f in d.farmerVisits
```

```
}

//every farmerVisit is linked to a farm
fact noFarmerNoVisits {
	all f:FarmerVisit | one farm:Farmer | farm in f.farmer
}

//every datetime is linked to a farmerVisit
fact dateTimeOnlyForVisit {
	all d:DateTime | one f: FarmerVisit | d in f.dateTime
}

//all visits in a dailyplan in the right day
fact visitIntheRightDay {
	all v: FarmerVisit, d: DailyPlan | v in d.farmerVisits implies SameDate[v.dateTime.date, d.date]
}

//in the same dailyPlan the visits do not overlap (for simplicity they are distanced at least by 2 hours)
fact nonOverlappingVisits {
	all disj v1,v2: FarmerVisit, d: DailyPlan | (v1 in d.farmerVisits and v2 in d.farmerVisits) implies
	(v1.dateTime.time.hour<=2+v2.dateTime.time.hour or v2.dateTime.time.hour<=2+v1.dateTime.time.hour)
}

//in the same dailyPlan the visits are always referred to different farmers
fact notTheSameFarmerAgain {
	all disj v1,v2: FarmerVisit, d: DailyPlan | (v1 in d.farmerVisits and v2 in d.farmerVisits) implies v1.farmer != v2.farmer
}

//every production that has finished since at least 2 months has to be evaluated  //for alloy purposes
fact oldProdEvaluated {
	all p: Production | differenceBetweenDates[CurrentDate, p.harvestFinishDate]>=60 implies p.evaluated = TRUE
}

//every production must start before finishing
fact startBeforeFinish {
	all p: Production | datePrecedesAnotherDate [p.startDate,p.harvestFinishDate]
}

//every evaluated production should have a score
fact evaluationWithScore {
	all p: Production | (p.evaluated =TRUE implies (p.score>=0 and p.score<=30)) and (p.evaluated=FALSE implies
	p.score=-1)
}

//a production can not be linked to more than one best practice
fact goodProdBestPract {
	all  disj b1,b2: BestPractice, p: Production  |  p in b1.production implies p not in b2.production
}
```

```
//productions with a score >21 should be linked to best practices
fact goodProdBestPractice {
	all  b: BestPractice, p: Production | p in b.production implies p.score>=21
}

//bpa can not be empty
fact BPAsNotEmpty {
	all b: Bpa | some a: BestPractice | a in b.bestPractices
}

//every answer is referred to an older request
fact answerAfterRequest {
	all a: Answer, r: Request | r in a.request implies datePrecedesAnotherDate [r.date, a.date]
}

//every agronomist can only answer a request once
fact answerOnlyOnePerReqAgronomist {
	all disj a1,a2: Answer, r: Request | (r in a1.request and r in a2.request) implies a1.agronomist!=a2.agronomist
}

//for every finished year every farmer should be visited at least twice (here only 2020 is considered)
fact atLeastTwoVisitsPerYear {
	all f: Farmer| #retrieveFarmerVisits[f,20] > 1
}

pred world1{
	#Area=1
	#Date=6
	#Farmer=2
	#DailyPlan=6
	#Bpa=0
	#Answer=0
	#Request=0
}

pred world2{
	#Area=2
	#Date=6
	#Farmer=4
	#DailyPlan=0
	#Bpa=0
	#Answer=5
	#Request=8
}

pred world3{
	#Area=1
	#Farmer=1
	#DailyPlan=0
```

```
        #Bpa=1
        #Production=4
        #BestPractice=2
        some p:Production | p.evaluated= FALSE
        #Answer=0
        #Request=0
}
run worldI for 7 but 6 Int
```

Three different worlds have been generated with alloy. World 1 focuses mainly on the DailyPlan, and shows the constraint by means of which every farmer should be visited at least twice during a year. World 2 describes answers and requests between farmers and agronomists. World 3 shows farmers' productions and their attributes, highlighting the relationships between them and the best practices.
To make it possible to run the last two world, the fact "atLeastTwoFarmersPerYear" should not be run, since for the sake of clarity the number of dailyPlans has been set to 0.
Below are shown the results of the execution and the three different worlds.

World 1

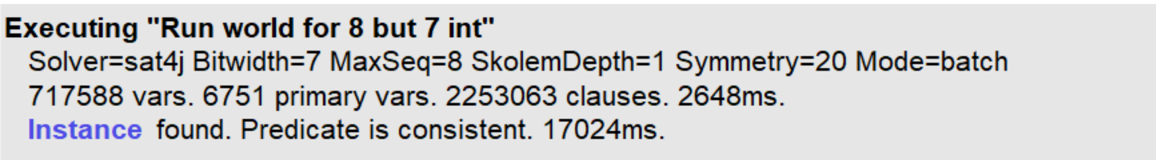

Executing "Run world for 8 but 7 int"
Solver=sat4j Bitwidth=7 MaxSeq=8 SkolemDepth=1 Symmetry=20 Mode=batch
717588 vars. 6751 primary vars. 2253063 clauses. 2648ms.
Instance found. Predicate is consistent. 17024ms.

Figure 4.1 – Execution result of world 1

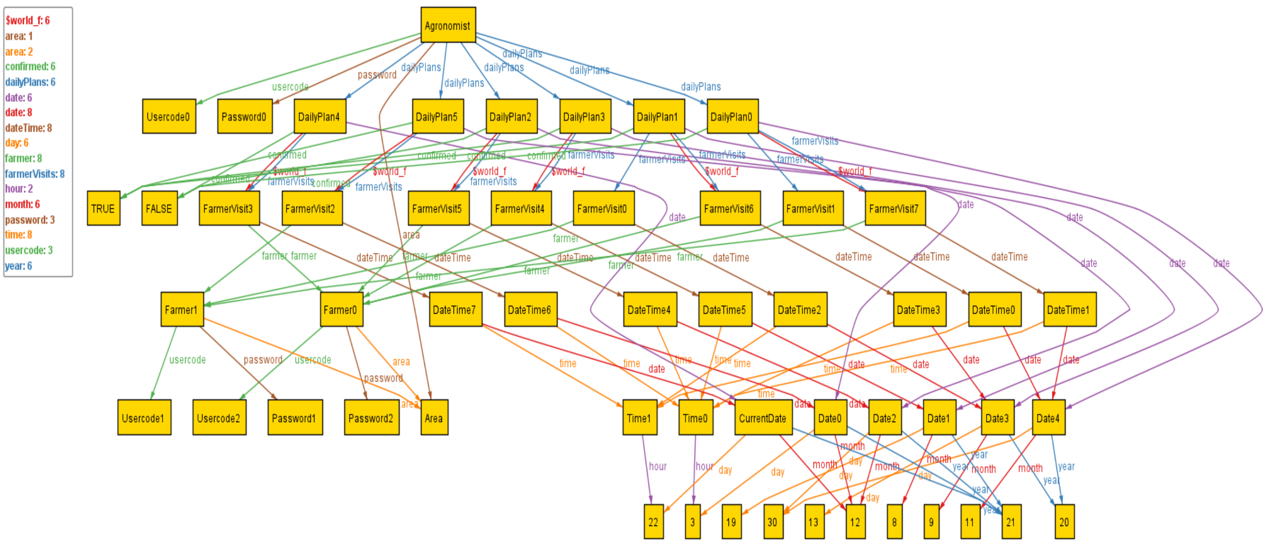


Figure 4.2 – World 1

World 2

Executing "Run world2 for 8 but 7 int"
Solver=sat4j Bitwidth=7 MaxSeq=8 SkolemDepth=1 Symmetry=20 Mode=batch
718388 vars. 6751 primary vars. 2251896 clauses. 2629ms.
Instance found. Predicate is consistent. 8771ms.

Figure 4.3 – Execution result of world 2

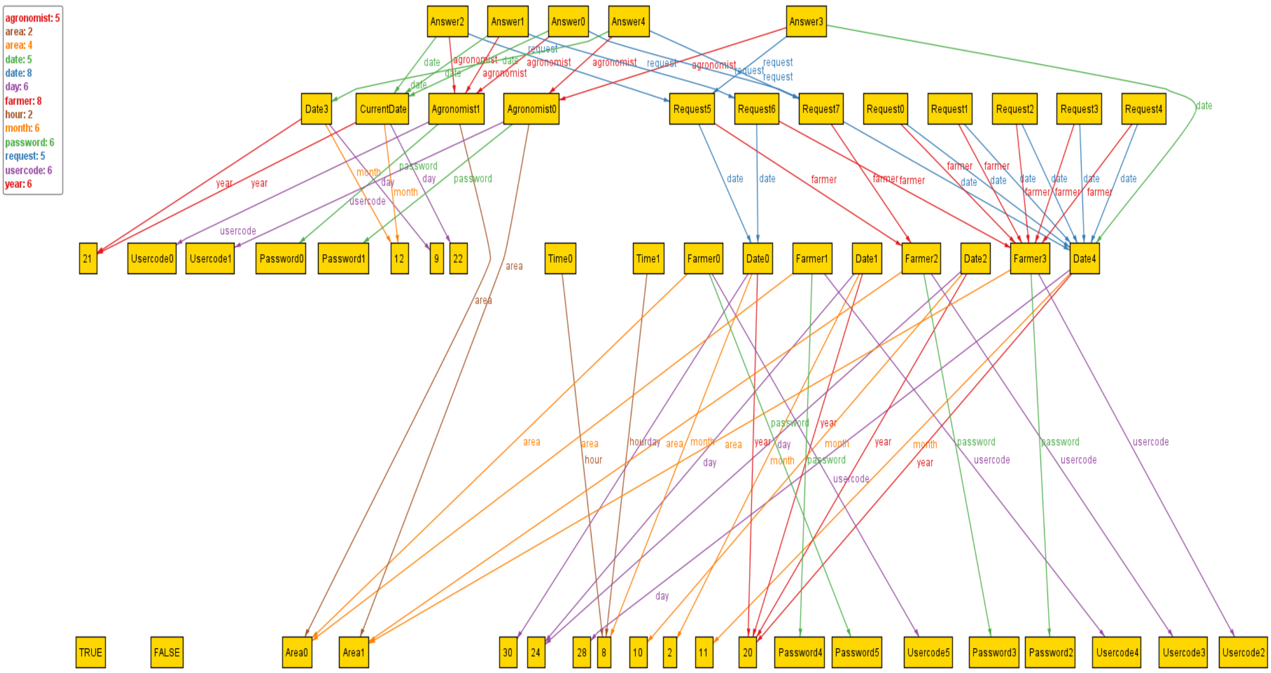


Figure 4.4 – World 2

World 3

Executing "Run world3 for 7 but 6 int"
Solver=sat4j Bitwidth=6 MaxSeq=7 SkolemDepth=1 Symmetry=20 Mode=batch
294746 vars. 3569 primary vars. 858765 clauses. 1100ms.
Instance found. Predicate is consistent. 9778ms.

Figure 4.5 – Execution result of world 3

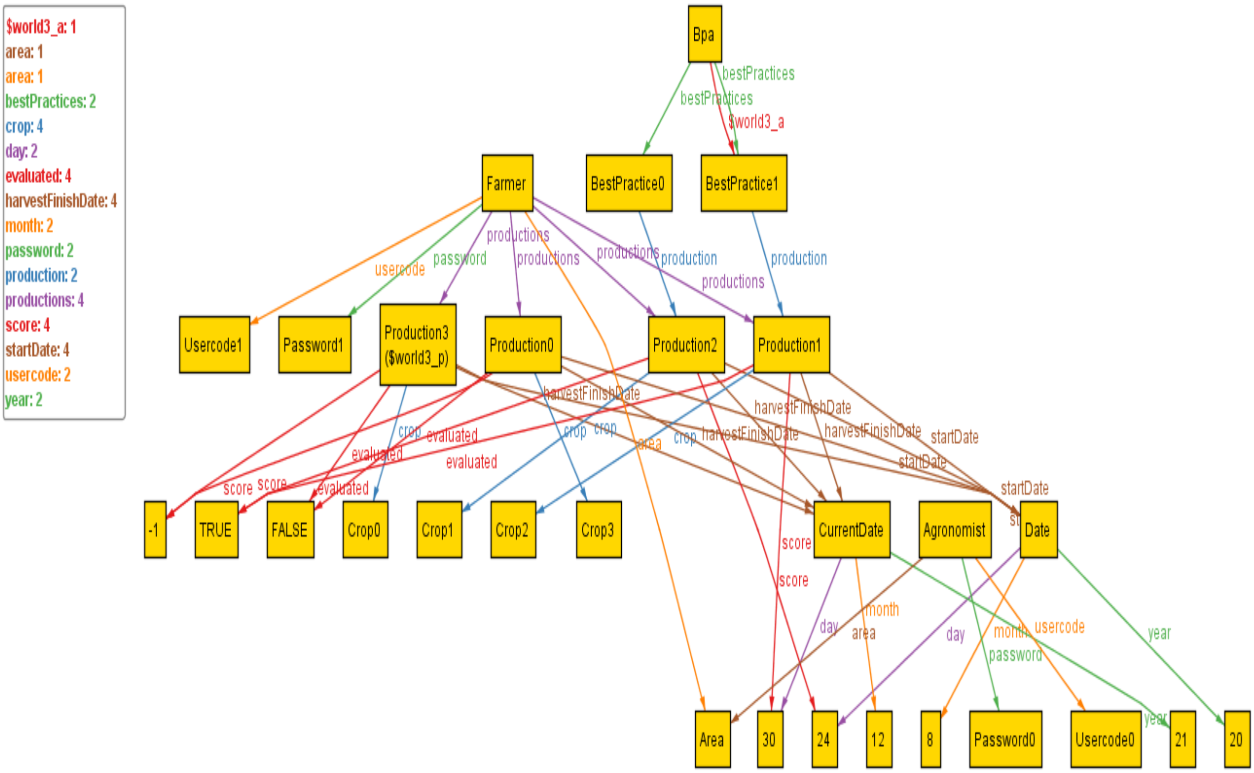


Figure 4.6 – World 3

5. Effort spent

Team

Topic	Hours
Initial briefing	3
General reasoning	10
Domain assumptions and functional requirements	5
Document revision	10
Total	28

Alessandro Maranelli

Topic	Hours
UML diagram	2
State Charts	3
Scenario definition	2

Functional Requirements	4
Use cases and sequence diagrams	2
Performance requirements	2
Alloy	14
Total	29

Amir Bachir Kaddis Beshay

Topic	Hours
Introduction	2
Scenario definition	2
Functional requirements	5
Use cases and sequence diagrams	16
Alloy	2
Total	27

Salvatore Marragony

Topic	Hours
Scenario definition	2
Product functions	3
User characteristics	2
User interfaces	5
Use cases and sequence diagrams	3
Requirements mapping	6
Additional revision	6
Alloy	2
Total	29

6. References

Websites and tools used while writing the RASD document:

- Github with information about the real UN project:
  https://github.com/UNDP-India/Data4Policy

- Telangana government website for weather forecasting:
  https://www.tsdps.telangana.gov.in/aws.jsp

- Tools for diagrams:
  - Visual Paradigm Online - Suite of Powerful Tools (visual-paradigm.com)
  - draw.io

- Tool for alloy:
  https://alloytools.org