POLITECNICO DI MILANO

Computer Science and Engineering

Requirements Analysis and Specifications Document

DREAM

Software Engineering 2 Project

Academic year 2021 - 2022

23/12/2021

Version 1.0

Authors:
Stefano Staffolani,
Stefano Taborelli,
Matteo Viafora

Professor:
Damian Andrew Tamburri

Revision History

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Stefano Staffolani, Stefani Taborelli e Matteo Viafora
Version:	1.0
Date:	23-December-2021
Download page:	Github
Copyright:	

Contents

1 Introduction

1.1 Purpose

Agriculture plays a pivotal role in India's economy as over 58% of rural households depend on it as the principal means of livelihood, 80% of whom are smallholder farmers with less than 5 acres of farmland. More than a fifth of the smallholder farm households are below poverty.
The COVID-19 pandemic has greatly highlighted the massive disruption caused in food supply chains exposing the vulnerabilities of marginalized communities, small holder farmers and the importance of building resilient food systems. It has become even more important now that we develop and adopt innovative methodologies and technologies that can help bolster countries against food supply shocks and challenges.
The purpose of Data-dRiven prEdictive fArMing (DREAM) is to help Telangana's government in designing, developing, and demonstrating anticipatory models for food systems using digital public goods and the community of agricultural worker. To achieve this DREAM allows the communication between the users, gives the possibility to retrieve data from the sensors in Telangana, to track the evolutions of the work of farmers, and to access the data from farmers or from any other sources.

1.1.1 Goals

The ambition of this project is that the adoption of DREAM will help the sustainability of the agricultural workers in Telangana. This is crucial during a pandemic emergency, like the one that is going on, because the farmers are the ones more afflicted. Below are presented the goals of DREAM.

G.1 Make available the weather forecasts collected by Telangana.

G.2 Create an anticipatory model for Telangana's food system.

G.3 Allow the communication between 2 users.

G.4 Support the agricultural work in Telangana.

- **G.4.1** Support the work of the farmers.
- **G.4.2** Support the work of the agronomists.
- **G.4.3** Support the work of the TPM.

1.2 Scope

Data-dRiven prEdictive fArMing (DREAM) is an easy-to-use and intuitive application which aims to settle for various problems faced by the member of the agricultural community of Telangana.

It allows Telangana's Policy Makers (TPM) to access data and analyze the performance of all farmers in the country, identify which farmers are coping well with adversity and which ones need help, understanding which initiatives are performing best, in order to replicate them in the whole Nation, and they can contact the resilient farmers to give them special incentives.
To improve their cultivations, farmers can get personalized suggestions, report any problem they encountered during their work, request more support by an agronomist or another farmer, discuss in the about any topic in the forum, but they can also provide their data of production.
Agronomists can use this system to increase the effectiveness of their work as they can manage their daily work and keep track of the visits they have planned throughout the year and evaluating the performance of farmers they are responsible of. Furthermore, they have access to the weather forecast and the data collected by the system to better take decisions with farmers or receive requests for help from farmers.

1.2.1 Shared Phenomena

Farmer

SP.1 Farmer logs into DREAM

SP.2 Farmer chats with another user

SP.3 Farmer reports a problem he/she faced

SP.4 Farmer asks for help

SP.5 Farmer replies to a request of help

SP.6 Farmer interacts in the Forum

SP.7 Farmer gets suggestions

SP.8 Farmer inserts data of production

SP.9 Farmer searches the weather forecast

SP.10 Resilient farmer shares information on how he/her overcame a meteorological adverse event

Agronomist

SP.11 Agronomist logs into DREAM

SP.12 Agronomist chats with another user

SP.13 Agronomist reads about a problem faced by a farmer

SP.14 Agronomist replies to a request of help

SP.15 Agronomist gets suggestions

SP.16 Agronomist looks for data of production of a farmer

SP.17 Agronomist analyzes the performances ranking of farmers

SP.18 Agronomist searches the weather forecast

SP.19 Agronomist visualizes his/her daily plan

SP.20 Agronomist updates his/her agenda

SP.21 Agronomist uploads his/her report about a visit to a farmed he/she has just concluded

TPM

SP.22 TPM logs into DREAM

SP.23 TPM chats with another user

SP.24 TPM reads about a problem faced by a farmer

SP.25 TPM looks for data of production of a farmer

SP.26 TPM analyzes the performances ranking of farmers

SP.27 TPM searches for the initiatives taken to help a farmer

SP.28 TPM ask to a Resilient Farmer to provide best practices on how he/she overcame a meteorological adverse event

DREAM

SP.29 System shows the weather forecast for the time slot selected

SP.30 System shows the ranking of the farmers

SP.31 System shows the suggestions requested by a farmer of an agronomist

SP.32 System shows the daily plan for the selected day

SP.33 System shows the production data of the selected user

SP.34 System shows the initiatives taken to help a farmer

1.2.2 Word Phenomena

WP.1 Farmer faces a problem

WP.2 Farmer faces a meteorological adverse event

WP.3 Farmer collects data of the production

WP.4 Agronomist visits a farmer

WP.5 Agronomist chooses which farms to visit

WP.6 Agronomist helps a farmer to improve his/her production

WP.7 TPM provide special incentives to resilient farmers

WP.8 Resilient Farmer receives incentives from Telangana

1.3 Glossary

This section explains the terms used throughout the document.

1.3.1 Definitions

Term	Definition
Good Farmers	Farmers who performed well
Resilient Farmers	Farmers who performed well, despite they faced an adverse weather event
Bad Farmers	Farmers that are not able to sustain their production
Farmer's Performance	It is the average quantity produced per acre by a farmer
Problem faced by farmers	Any difficulty that a farmer faced during his/her work, i.e., a bug infestation of the cultures
Temporal Slot	An arch of time selected by the user to see the weather. E.g., it could be just one day, but also 6 months
Type of Production	How the cultivation is grown, i.e., Shifting Cultivation, intensive, and many more
Data of Production	Information concerning a single cultivation of a farmer. It includes tons produced per acre of land used, fertilizer used, humidity of the soil, type of production, water usage, seed planted and the zone of the cultivation
Credentials	Username and Password given by Telangana to a user to access DREAM
Short-Term Forecast	Weather forecast concerning 7 days or less
Long-Term Forecast	Weather forecast concerning more than 7 days
Zone	It refers to a District and a Mandala
Meteorological Adverse Event	A climate event that caused many damages to a farmer

1.3.2 Acronyms

Acronym	Term
DREAM	Data-dRiven PrEdictive FArMing in Telangana
GUI	Telangana Policy Makers
GPS	Global Positioning System

1.3.3 Abbreviations

Abbreviation	Term
e.g.	Exempli Gratia
i.e.	Id est
G	Goal
WP	World Phenomena
SP	Shared Phenomena
R	Requirement
DA	Domain Assumption
UC	Use Case

1.4 Reference Documents

Below is presented the list of all the documents used to create this document.

- Project assignment specification document
- ISO/IEC/IEEE 29148 - Systems and software engineering
- Course slides on WeBeeP
- UNDP India's Github
- Telangana's Forecast WebPage
- Agriculture farming in Telangana
- Package to insert the Alloy code Github

1.5 Document Structure

This document is presented as it follows:

1. **Introduction** contains a preamble of the given problem and proposes, in a simple way, the system and its goals as solution
2. **Overall Description** gives a general description of the system, focusing on its functions and constraints. Moreover, it provides the domain assumptions of the analyzed world
3. **Specific Requirements** explains in detail the functional and non-functional requirements. It lists the possible interactions with the system in the form of scenarios, use cases and sequence diagrams
4. **Formal Analysis** contains the Alloy model of some critical aspects of the system and an example of the generated world
5. **Effort Spent** keeps track of the time spent to complete this document. The first table defines the hours spent as a team for taking the most important decisions or reviewing contents, the seconds contain the individual hours spent working on this project

2 Overall Descriprion

2.1 Product Perspective

The system will be developed from scratch, and it will be integrated with the other system already present in Telangana State.

2.1.1 Class Diagram

The class diagram in Figure. 1 is a high-level representation of the whole system.
The main elements in the class diagram are:

- **User:** identifies a user with his unique credentials. Users can be farmers, agronomists or TPM
- **Farmer:** he relevant information
- **Problem:** identifies a problem associated to a farmer
- **HelpRequest:** identifies a help request associated to a farmer
- **Replies** identifies the replies to a help request, written either by a farmer or an agronomist
- **Dataproduction:** identifies the data about the entire production of a farmer
- **DataCultivation:** identifies the data about a single cultivation of a farmer
- **Discussion:** identifies a discussion in the forum
- **DiscussionReplies:** identifies the replies to a discussion in the forum
- **Messages:** identifies a message sent using the chat function, with all the relevant information
- **Zone:** identifies a Mandala and the relative District
- **Location:** identifies a specific address
- **WeatherRequest:** identifies a weather forecast in a set time slot
- **WikiFarmRequest:** identifies a suggestion request
- **Agenda:** identifies the agenda associated to each agronomist
- **DailyPlan:** identifies a daily plan with all the relevant information
- **Event:** identifies an event programmed by an agronomist, with all the relevant information
- **Report:** identifies a report written by an agronomist at the end of a visit
- **TPM:** identifies a TPM with all the relevant information
- **Ranking:** identifies the ranking that associates each farmer with his corresponding position

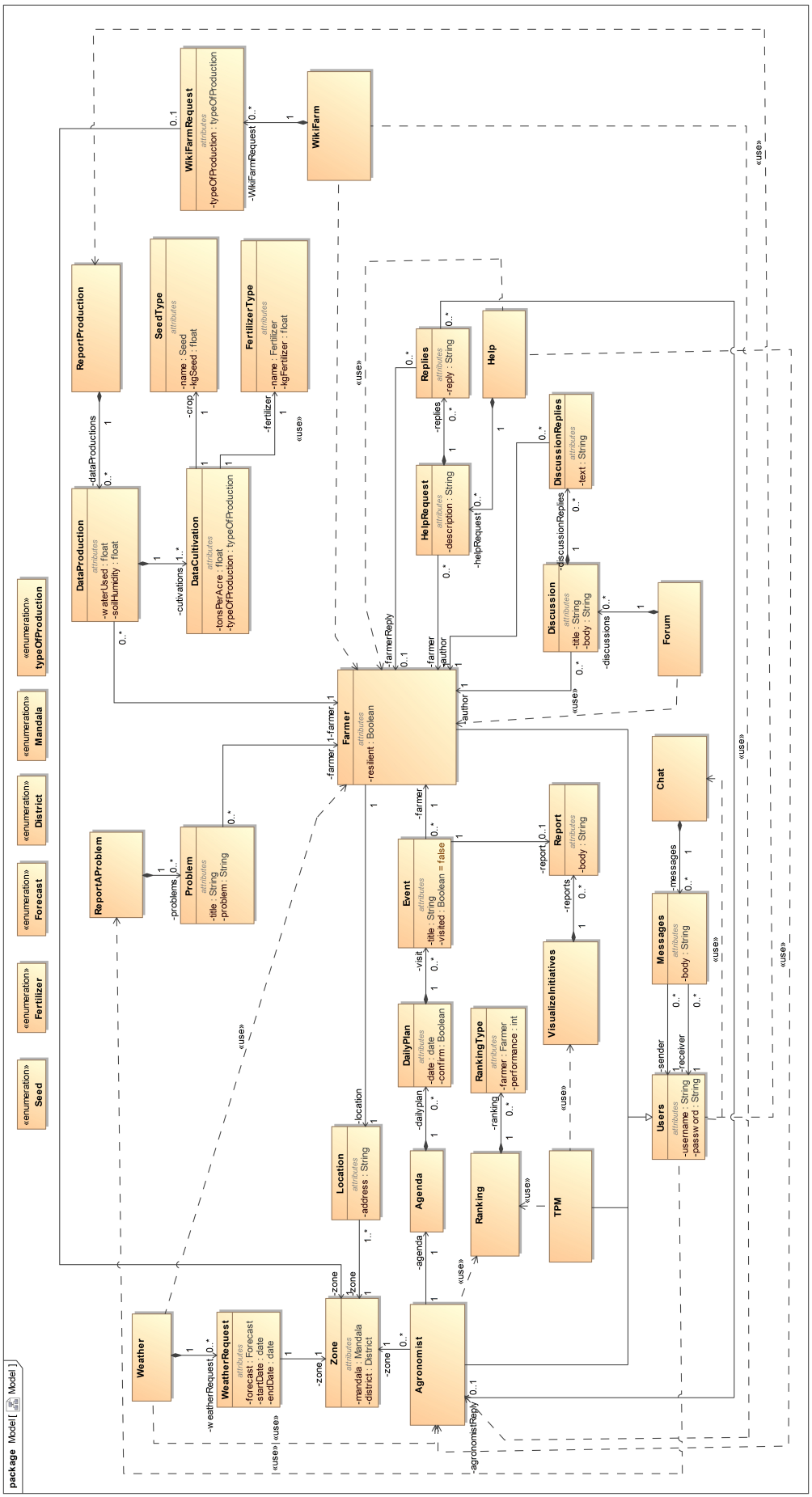

Figure 1: DREAM metamodel.

2.2 Product functions

This section provides a description of the functionalities available in DREAM.

2.2.1 LogIn

LogIn allows users to access DREAM, using their credentials provided by Telangana.

2.2.2 Report a Problem

Report a problem allows a farmer to report any kind of problem he has encountered while growing any of his crops, whether he has solved it himself or has yet to solve it and wants to ask for help via the "Help" section.
TPM and agronomists can access the list of all problems according to the selected area, so that they can keep track of what is happening in their area.

2.2.3 WikiFarm

WikiFarm allows agronomists and farmers to search for information on what seed to plant and what fertilizer to use, according to the zone they have entered and the type of production they have selected.

2.2.4 Weather

Weather allows agronomists and farmers to search for short-term or long-term forecasts for the selected zone. The system accesses the Telangana weather portal and retrieves the data that the user has requested.

2.2.5 Report production

Report Production allows farmers to enter their production data for each of their crops, while agronomists and TPM can retrieve all production data entered by a specific farmer.

2.2.6 Help

Help allows each farmer to create a new help request to overcome a problem he has encountered and was unable to overcome by his own efforts. Each help request can be seen by all farmers or agronomists, who, thanks to their experience, can help and answer the farmer's request.

2.2.7 Forum

Forum is a section of DREAM where farmers can create public discussions on any topic they wish to discuss. Only farmers can participate in each discussion, and it is an attempt to strengthen the community of Telangana farmers.

2.2.8 Agenda

Agenda is the most useful working tool for agronomists. They can see their visits to farmers for the current day or, even, for any day they wish. Agronomists can also create new visits or reschedule them for a new date.
Agenda is also the tool with which agronomists can confirm that they have made a visit and thus upload their report of the initiatives taken with the farmer. In addition, they can report any changes to their initial schedule.

2.2.9 Visualize Initiatives

Visualize Initiatives allows TPM to keep track of the initiatives carried out during visits by agronomists to a farmer. This allows the TPM to retrieve details of initiatives that have been essential in improving the production of farmers who were performing badly or had faced an adverse weather event.

2.2.10 Rankings

Rankings is a tool that allows TPM and agronomists to evaluate in a simple and effective way the performance of farmers in Telangana or in a selected area. The results are presented in descending order according to the performance of the farmers.

2.2.11 Chat

Chat allows two DREAM users to communicate with each other in private, using their username.

2.3 Scenarios

This section provides an overview description of how DREAM helps the agricultural community of Telengana during the everyday life.

2.3.1 Farmer

F.1 Bug infestation: Hansal during the spring season has been afflicted by a bug infestation and she would like to report the event in the systems. She opens "Report a Problem", inserts her type of production and then, she describes the problem she has encountered. She also would like to request for help, as she's not able to defeat the infestation on her own. She opens "Help" and she posts a new request of help that can be seen by agronomists of her area and other farmers.

F.2 The kind farmer: Harishankar is an expert farmer. He opens "Help" and he notices there is an assistance request from Hansal, who is afflicted by a bug infestation. Since he had the same crisis in the past, he knows the solution to this problem, so he decides to reply to the help request and he suggests using a poison that kills the bugs but does not damage the plants. Then, he sees that it's a common topic in the Forum, so he replies to a conversation about bug infestation to leave the same advice for everyone to see.

F.3 The new business of Ayodele: Ayodele is a farmer from Maddur who would like to expand his production, but he does not know what is the best choice to plant. First, he opens "Weather" and he discovers that for the next six month there won't be heavy rainfall and the climate will be good for almost any type of plant, except cotton. Then, he opens "WikiFarm" and he researches for the crops and the fertilizer to use, based on his District, Mandala and type of production. DREAM suggests him to plant cotton or Sesame and it suggests as fertilizer "Happy Soil" and "Grow Better". Based on the information obtained, Ayodele decides to plant sesame and to use "Happy Soil".

F.4 Mrigankshekhar, the resilient farmer: Mrigankshekhar is a farmer whose production grew a lot during the last year, although he faced an extraordinary amount of rainfall due to the monsoon season. At the end of the month, he opens DREAM to insert the data about his production. For each type of his production, the system asks to the farmer the amount of production and the amount of land used for that cultivation.
After a week a TPM, through "Chat", asks him to share information on how he overcame the adverse weather, as he is resulted to be a resilient farmer and he will get extra financial incentives from the State. Mrigankshekhar gladly accepts the request, and he opens the "Forum" and he creates a new discussion describing how he planted a new type of cotton that is resistant to heavy rain.

2.3.2 TPM

TPM.1 Resilient farmers are rewarded: Ganga works for the Ministry of Agriculture in Telangana and is responsible for evaluating the performance of farmers within the country. He first opens "Ranking" and asks the system to generate a ranking of the farmers' performance. He realizes that Mrigankshekhar has managed to survive a monsoon flood and he is one of the most productive farmers of the State. Through "Chat", she notifies Mrigankshekhar that he has been awarded of a government bonus and she asks him to share with the other farmers how he survived the flood.

TPM.2 Nobody left behind: Rao works for the Ministry of Agriculture in Telangana and is tasked with finding the most struggling farmers in the country. He first accesses DREAM and asks the system to generate a ranking of the farmers' performance. Apu is among the last in terms of performance and Rao decides to open "Report Production" to retrieve Apu's data. He notices, from the data collected by sensors in the water distribution, that he used a low quantity of water, which caused the low yield of his soya crop. Finally, Rao opens "Chat" and he reports the case to the agronomist in charge of Apu's area.

TPM.3 For a better Telangana: Chandler works for the Ministry of Agriculture in Telangana and is tasked with evaluating which initiatives have improved the performance of resilient farmers. He first accesses "Ranking" and he asks the system to generate a ranking of farmers' performance, identifying which ones are resilient to climatic adversity. He is interested in the case of Pamir, which has doubled its production thanks to the work of agronomist Raj, who advised him to use 'Happy Soil' as a fertilizer against the magnesium deficiency in the soil caused by last April's drought. Chandler takes note of the initiative to propose it as a possible solution in the future.

2.3.3 Agronomist

A.1 A day as an agronomist: Keffir goes to his office and starts his day by opening "Weather" to access the weather forecast for the current day and for the rest of the week. Then, he opens "Agenda" and he checks if any farmer visits are planned. Today he has to visit Menroosh in Palmela for the first time this year. The agronomist takes the car and drives to Menroosh. During the visit the agronomist and the farmer look at last year's production data and they decide to try planting Green Beans, while they decide to change the fertilizer for cotton production. The agronomist opens "Agenda" and confirms that he has completed his visit. After that, he enters a report of the initiatives taken with the farmer. Once back home, Keffir confirms that he has completed and respected his daily schedule.

A.2 A bad day for Chameli: After completing her visit to the farmer Lalitchandra, Chameli has to go and visit the next farmer on her agenda. Unfortunately, when she gets into the car, she notices that the engine of her car has a problem, because a light has turned on in the dashboard of her car. So Chameli decides to postpone her planned visit until next week and go to the mechanic. She opens "Agenda", updates her daily schedule, in order to report that she has not been able to visit the second farmer on her schedule today and she postpones the visit to the following Tuesday. She marks her work activity as finished, then closes the portal and goes to the mechanic.

A.3 The work office of Nandakishor: Nandakishor opens "Agenda" and checks his daily schedule, but he notices that he has no visits scheduled for today. Then he opens "Report a Problem" and notices that Bamoosh has written a post. He opens the report and finds that Bamoosh has an unused land, and he would like to use it, but he doesn't know what to plant. Nandakishor decides to help him and first he opens "Weather", he enters the farmer's District and Mandala, and discovers that the area is suitable for rice or corn production. In order to decide he looks the humidity of the soil in the "WikiFarm" and discovers that rice is the best option, combined with 'Super Growth 80' as fertilizer. The agronomist answers Bamoosh through "Chat" advising him to plant basmati rice and to use 69g 'Super Growth 80' per acre. Since there are no requests for help, he decides to

schedule his visits for next Monday. He enters his working area and looks at the list of farmers' performances. Nandakishor notices that Tahir, despite having already been visited twice this year, is performing badly with sesame cultivation. So the agronomist plans to visit Tahir on Monday.

2.4 User Characteristics

This section describes all the users that can access DREAM.

2.4.1 Farmers

Farmers are the people in charge of taking care of the administration of a farm. They can be of all ages and don't necessarily have a complex device to access the system. For this reason, the system should be as easy as possible to access, and it should have an intuitive GUI.

2.4.2 Agronomists

Agronomists are the people in charge of help the agricultural work of Telangana's farmers and they have to report their work to the State, in order to help TPM. They could work from anywhere they want; they only need a device to access DREAM and a stable internet connection in all their work area.

2.4.3 TPM

TPM are the people of Telangana who are responsible of creating new Policies for the agricultural system of Telangana, they have to keep track of the work of agronomists and how farmers can improve their farms. They are also in charge of creating all the credentials of DREAM's users and to associate the location of each farmer to their username.

2.5 Constraints

In this topic it is put on paper a general description about considerations, boundaries and items that will limit the system's options.

2.5.1 Structure Limitation

The credentials of all users are created and sent by TPM during the setup phase of DREAM and, also, each farmer is also associated with the address of his/her farm. This is made to keep DREAM accessible only to people of really are part of this project.
DREAM request a user to insert a zone by selecting it through the territorial division of Telangana, first he/she selects the District and then he/she selects the Mandala. This choice is made DREAM accessible to everyone, even with an obsolete device without using a GPS sensor.

2.5.2 Hardware Limitation

In order to access DREAM a user needs:

- 2G/3G/4G/5G connection or Wi-Fi available
- web browser like Firefox, Chrome, Safari

2.5.3 Interfaces to other applications

The proper functioning of the system is strictly subordinated to an external distributed web service. This is required to keep the system update and accessible all the time, even if a server has a failure and the accesses will be redirected to another server.

2.6 Domain Assumptions

The properties that hold in the analyzed world will be listed below.

DA.1 All Telangana's people, who interact with system, have an internet connection

DA.2 All Telangana's people, who interact with system, have access to a browser

DA.3 Every time a farmer incurs into a problem, he/she inserts the details in the system

DA.4 At the end of each month, farmers insert into the system the result of their production

DA.5 Agronomists have a vehicle to visit farmers

DA.6 Agronomists visit only the farmers who are in their daily schedule

DA.7 An agronomist could not complete his/her daily schedule

DA.8 Agronomists visit more than two times a year only the farmers who are performing badly

DA.9 During the visit of a farmer by an agronomist, he/she proposes suggestions to improve the production based on the data collected since the last visit

DA.10 All the farmers have a banking account where they can receive incentives

DA.11 All resilient farmers receive special incentives from TPM

DA.12 All good farmers provide best practices to TPM

DA.13 Farmers are ranked based on their productivity per acre

DA.14 All lands in Telangana have a sensor of humidity of the soil

DA.15 Water distribution system has sensor to measure the amount of water given to each land

DA.16 Water distribution sensors never stop to work

DA.17 Humidity sensors in a land never stop to work

DA.18 Weather forecast for a specific zone is always available

DA.19 Agronomists know the number of times each farmer has been visited

DA.20 Farmers always insert correct data about their production

DA.21 The quantity produced by a farmer is quantified in tons

DA.22 The lands are measured in acres

DA.23 All the users know their credentials given by Telangana state

DA.24 Any user knows the correct username he/she wants to contact using "Chat"

DA.25 All the users do not forget their credentials given by Telangana state

DA.26 Each farmer is associated with only one farm

DA.27 Every farmer stays in the same location

Furthermore, we assume the credentials to LogIn are already given by Telangana State to all the agents acting with the system, in order to exclude that external actor could have access to the platform. Each farmer user is already associated to his/her location.

3 Specific Requirements

This section is devoted to a specific description of every kind of requirement the system has to deal with in order to achieve all the functionalities described.

3.1 External interface Requirements

3.1.1 User Interfaces

DREAM is divided into sections that could be accessed by the users of the system. Below there is the list of the section and the user that can access the section. When there are two divisions in the users it means that the section is the same, but it will show different functionalities.

LogIn
Login is the first page that opens each time any user opens DREAM. After typing the credentials given by Telangana's state, the user has to click the "Log In" button to verify his/her credentials.

Report a Problem
Report a Problem allows farmers to report every problem they encounter during their activity. It also allows agronomists and TPM to visualize the farmer's reported problems, either to prepare for the visits or to record the most common issues.

WikiFarm
WikiFarm is used by farmers and agronomists. After the user provides his/her zone and type of production, the system shows a list of the suggested crops and a list of the suggested fertilizers.

Weather
Weather is used by farmers and agronomists to visualize weather forecasts. The user has to provide his/her zone and a timeslot to visualize, then the system retrieves and shows the requested forecast.

Report Production
Report Production is used by farmers at the end of each month to report their production. It allows the farmers to insert data about each of their cultivation, then the system retrieves the data about soil humidity and water usage from the sensors and a complete report is published. The page can also be accessed by agronomists and TPM, to visualize the reports published by the farmers.

Help
Help is used by farmers to post help requests, which can be answered by other farmers and agronomists. To publish a new help request a farmer has to provide a description of his/her problem.

Forum
Forum allows farmers to create new discussions and reply to existing ones. In order to create a new discussion, a farmer has to insert a title and the topic of the discussion.

Agenda
Agenda is used by agronomists to visualize, change, and confirm their daily plan. It allows agronomists to open an event, either to visualize it, postpone it or publish a report after the visit is completed.

Visualize Initiatives
Visualize Initiatives allows TPM to visualize the reports associated to a farmer, in order to look at the initiatives taken by the agronomists who visited them. This is used by TPM to find the steering initiatives.

Rankings
Rankings is used by agronomists and TPM to create a ranking of the farmer's performances in a zone. If the selected zone is "All", the system will create a ranking of every farmer registered in DREAM.

Chat
Chat can be accessed by every user in order to create a new chat or reply to an open one. To create a new chat a user has to insert the username associated to the user he/she wants to contact.

3.1.2 Hardware Interfaces

To operate properly DREAM needs information about water usage and soil humidity of every farm, so two hardware components shall be provided to each farmer:

- A humidity sensor, installed on the farm ground of every farmer
- A sensor to measure the water supplied to each farm through the water irrigation system

Furthermore, to access DREAM any user is required to own a device connected to internet.

3.1.3 Software Interfaces

The system interfaces with external IT providers in order to acquire the weather forecasts and to combine the information acquired by each farmer.

3.1.4 Communication Interfaces

All the communications from and to DREAM are made through the HTTPS protocol.

3.2 Use Cases Description

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

Name	Platform login
ID	UC.1
Actors	TPM, Farmer, Agronomist
Entry conditions	• The platform is running • The user is running the application on his/her device
Flow of events	i. The user opens DREAM ii. The user inserts their username and password iii. The user clicks on "Log In"
Exit conditions	The actor has successfully logged into the web platform.
Exceptions	• If the username is not recognized by the system, the credentials are not registered, or the username is incorrect. The system notifies the actor, and the procedure is aborted. The actor is redirected to the login page. • If the inserted password is wrong, the system notifies the actor, and the procedure is aborted. The actor is redirected to the login page.

Table 1: *Platform login* use case description.

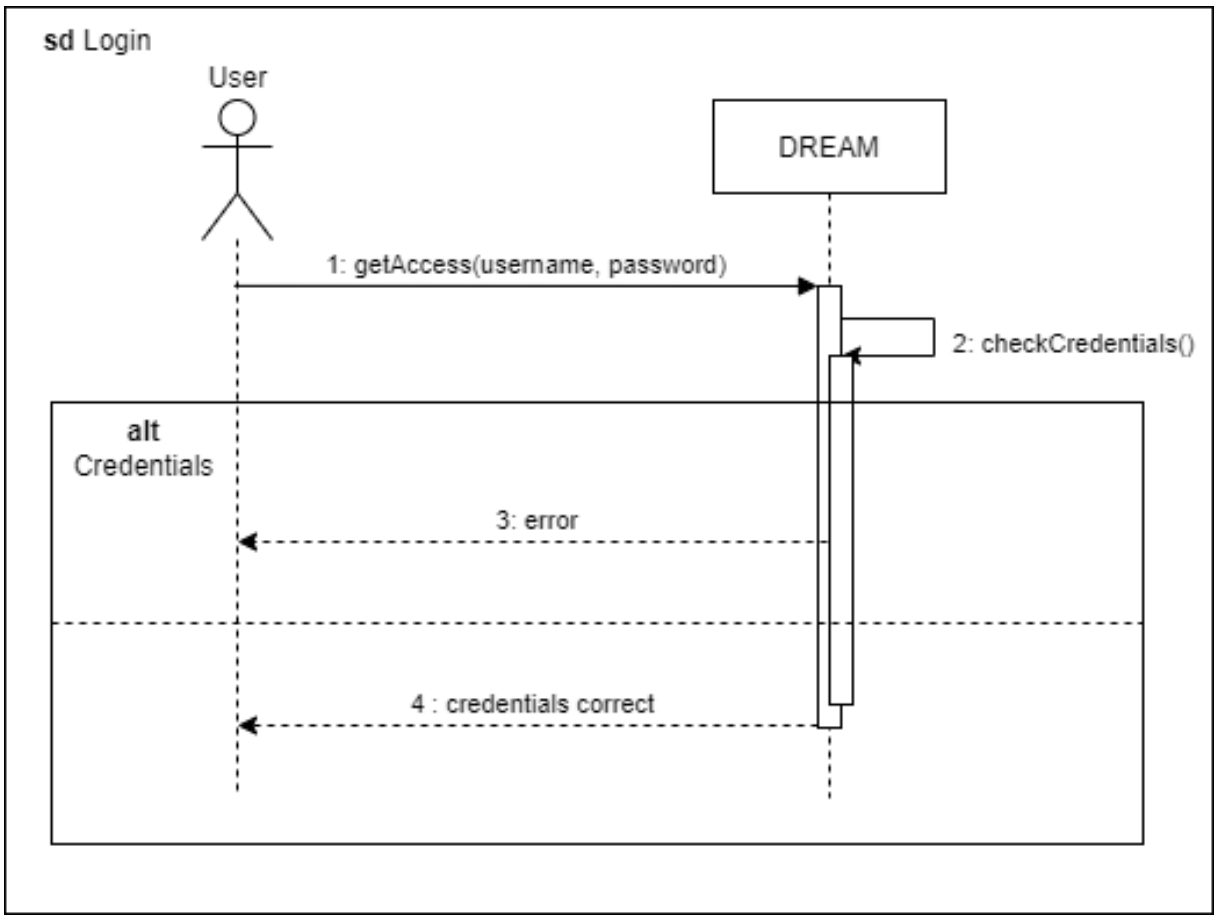

Figure 2: *Platform login* sequence diagram.

Name	Interact with another user
ID	UC.2
Actors	TPM, Farmer, Agronomist
Entry conditions	• The platform is running • The user is running the application on his/her device • The user is logged into the system
Flow of events	i. The user opens "Chat" ii. The user clicks on "New Chat" iii. The user types the username of the other user he/she wants to contact iv. The user clicks on "Start Conversation" v. System creates a new chat between the 2 users vi. The user types the message he/she wants to send
Exit conditions	The user clicks on "Send"
Exceptions	• If the username of a user or the text of the message is missing, the system does not submit the message and highlights the missing parts.

Table 2: *Interact with another user* use case description.

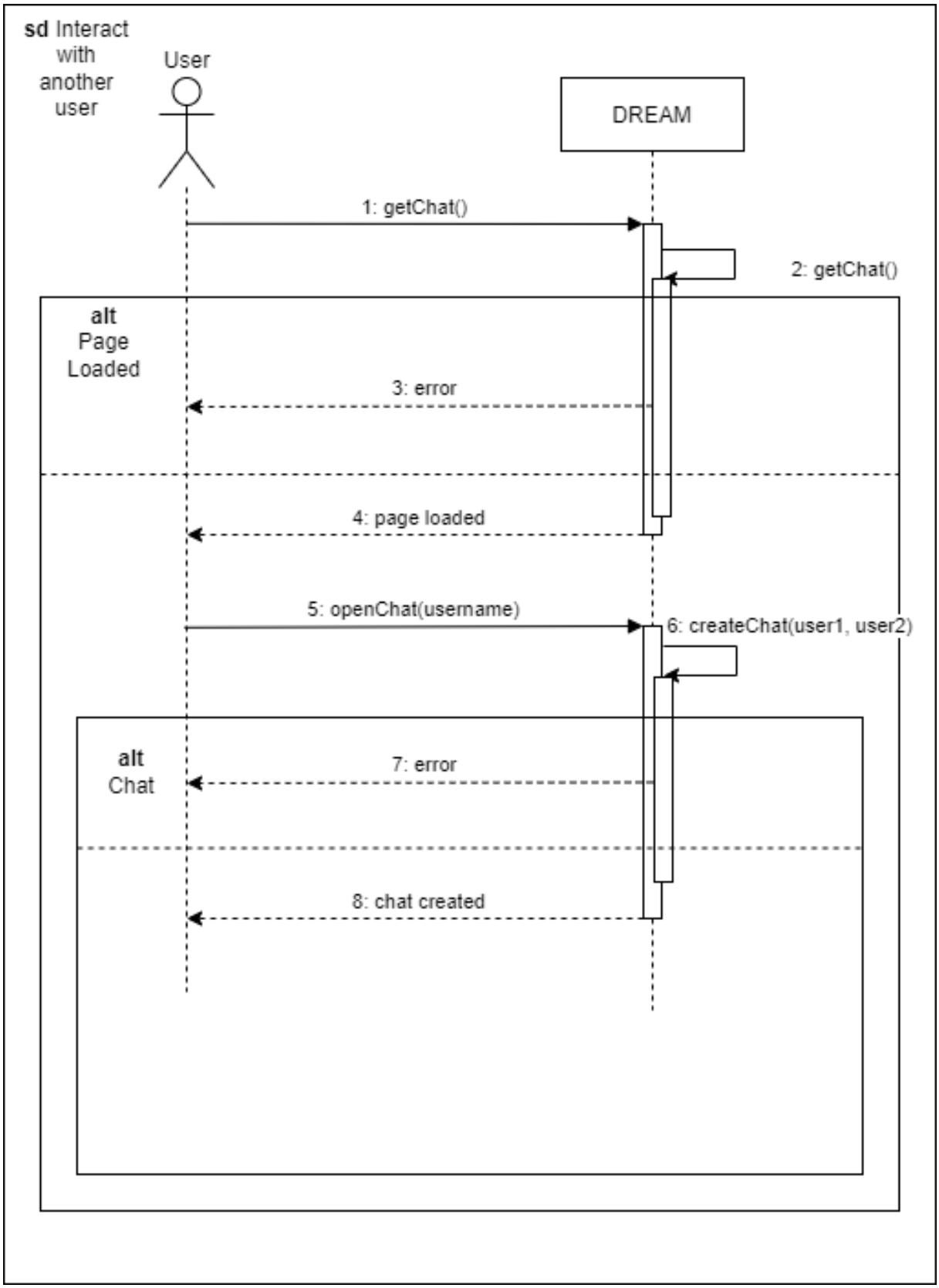

Figure 3: *Interact with another user* sequence diagram.

Name	Reply to a new message
ID	UC.3
Actors	TPM, Farmer, Agronomist
Entry conditions	• The platform is running • The user is running the application on his/her device • The user is logged into the system
Flow of events	i. The user opens "Chat" ii. System shows the list of all messages that the user received and highlights the message/s he/she has not read iii. The user selects the conversation he/she wants to reply iv. System shows the conversation between the 2 users v. The user types the text of the message
Exit conditions	The user clicks on "Send"
Exceptions	• If the text of the message is missing, the system does not submit the message and shows an error message.

Table 3: *Reply to a new message* use case description.

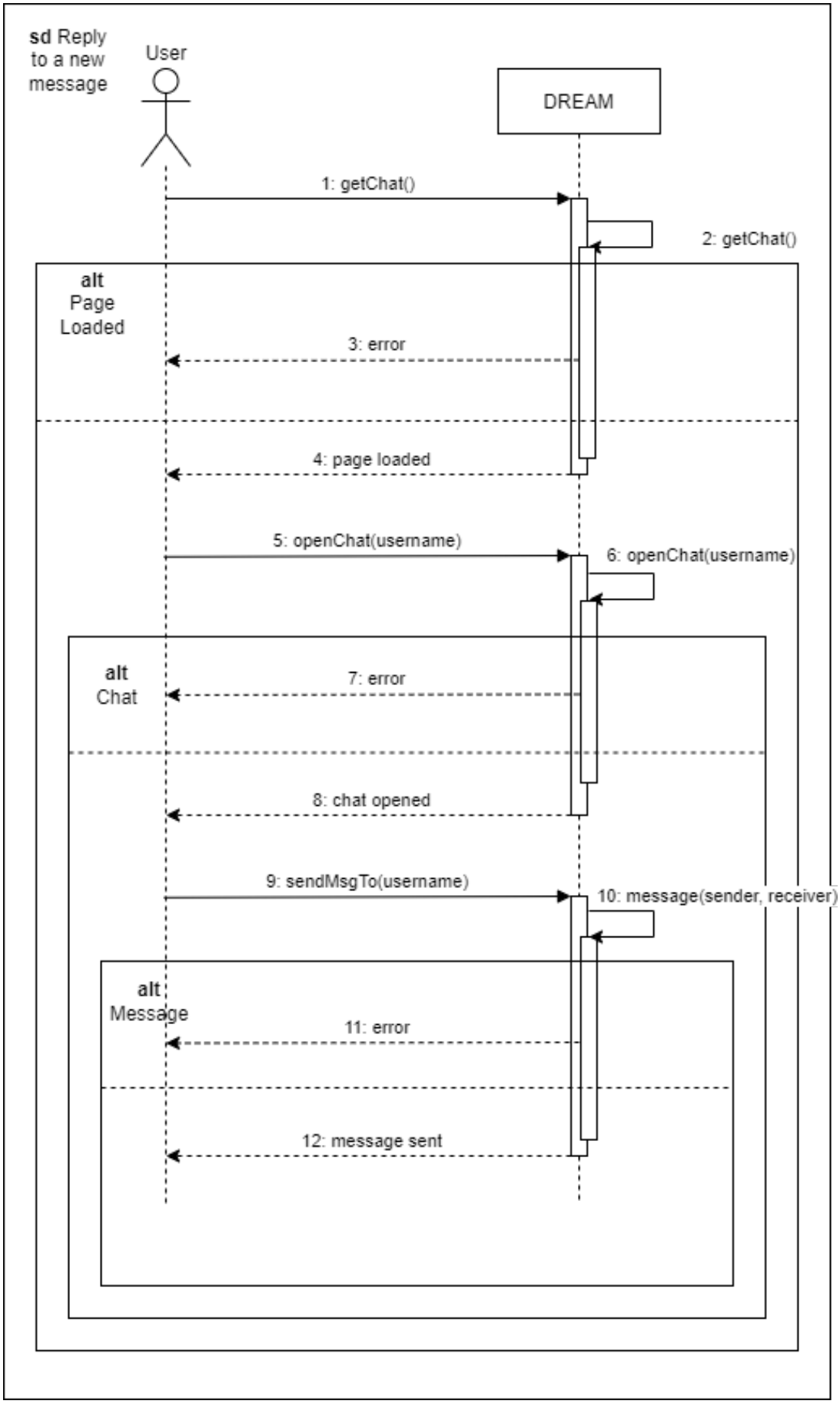

Figure 4: *Reply to a new message* sequence diagram.

Name	Report of a problem
ID	UC.4
Actors	Farmer
Entry conditions	• The platform is running • The farmer is running the application on his/her device • The farmer is logged into the system
Flow of events	i. The farmer goes to "Report a Problem" ii. The farmer types a title for his/her problem iii. The farmer inserts the description of the problem faced iv. The farmer clicks on "Submit" v. System retrieves the farmer's location vi. System adds the location of the farmer to the form
Exit conditions	The farmer successfully submits the form
Exceptions	• If the form is not submitted, the page is reloaded with all the text inserted by the farmer. • If the form is empty, the system does not submit the data and highlights the missing part.

Table 4: *Report of a problem* use case description.

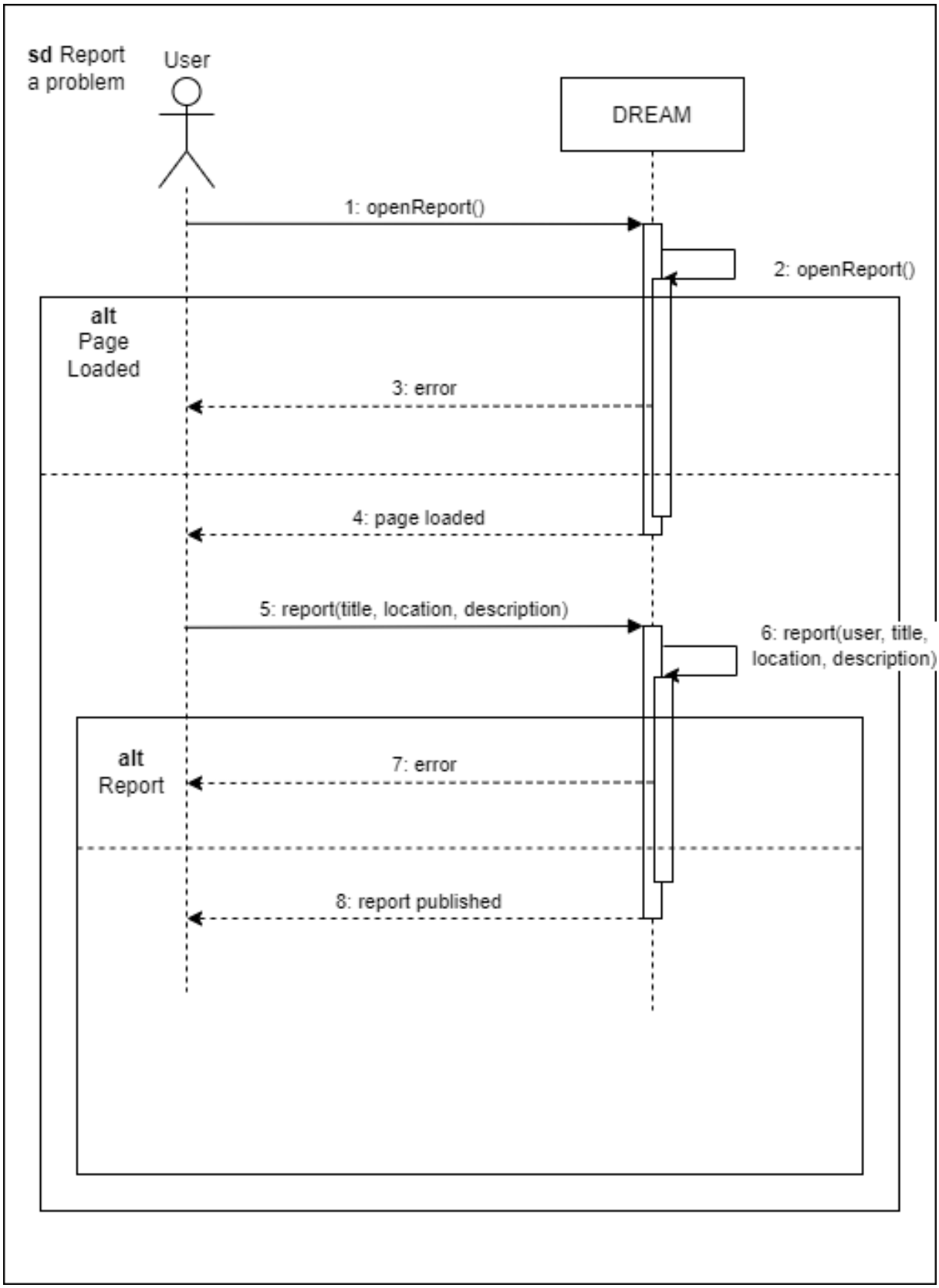

Figure 5: *Report of a problem* sequence diagram.

Name	Read about a problem
ID	UC.5
Actors	TPM, Agronomist
Entry conditions	• The platform is running • The user is running the application on his/her device • The user is logged into the system
Flow of events	i. The user opens “Report a Problem” ii. The user selects the area of interest iii. The user clicks on “Search” iv. System shows the list of titles of all problems in the area selected v. The user selects one in the lists to see the description vi. System shows the report selected
Exit conditions	System shows the description of the problem selected by the user
Exceptions	• : If in the area there are any report a message is shown: “For the area selected there are no reports”

Table 5: *Read about a problem* use case description.

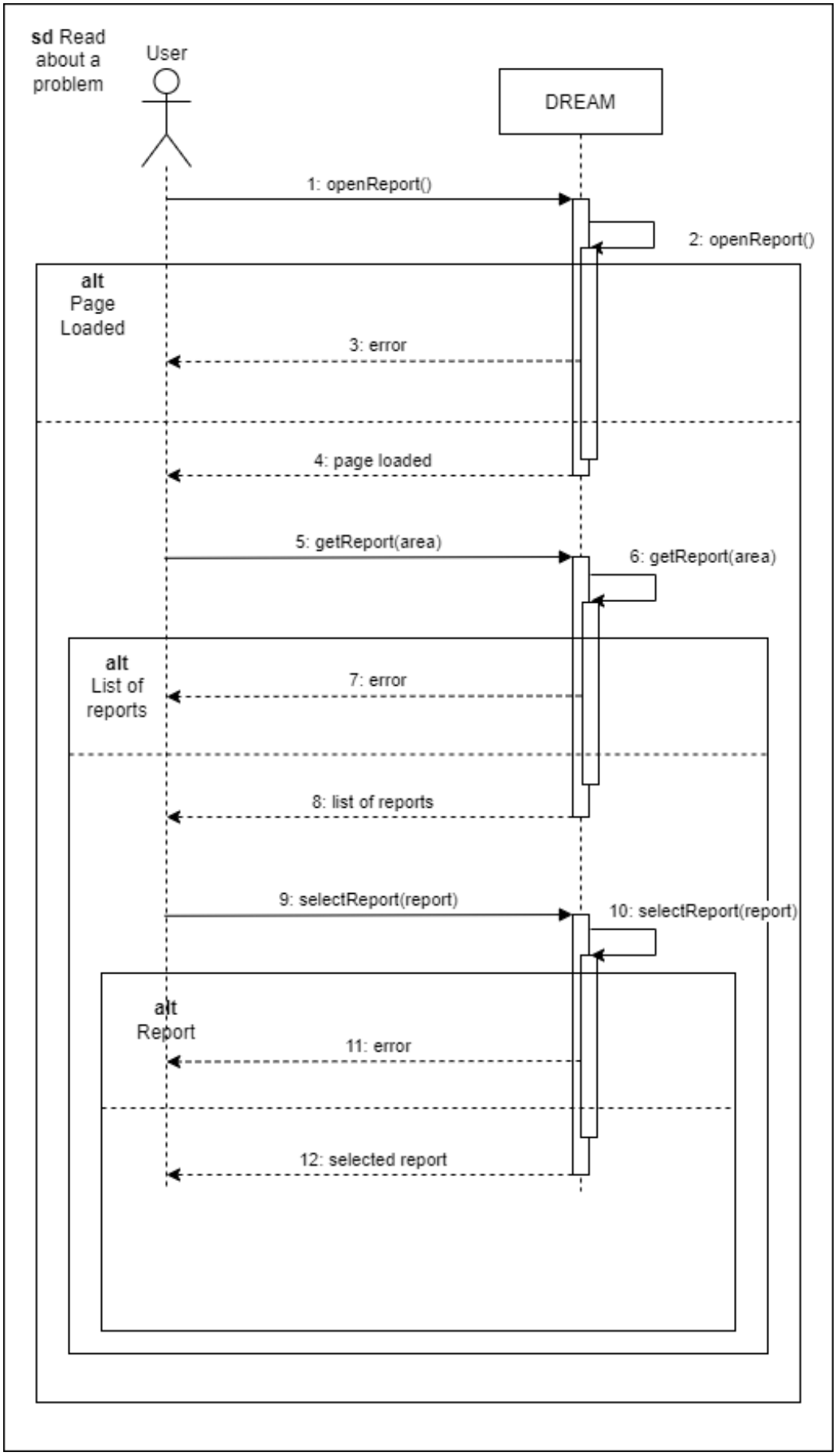

Figure 6: *Read about a problem* sequence diagram.

Name	Ask for help
ID	UC.6
Actors	Farmer
Entry conditions	• The platform is running • The farmer is running the application on his/her device • The farmer is logged into the system
Flow of events	i. The farmer accesses "Help" ii. The farmer clicks on "Get support" iii. The farmer describes how he/she wants to be helped iv. The farmer clicks on "Submit Request" v. The system retrieves the farmer's location vi. System adds the location of the farmer to the form
Exit conditions	The farmer successfully submits the form
Exceptions	• If the form is not submitted, the page is reloaded with all the text inserted by the farmer. • If a field of the form is empty, the system does not submit the data and highlights the missing parts.

Table 6: *Ask for help* use case description.

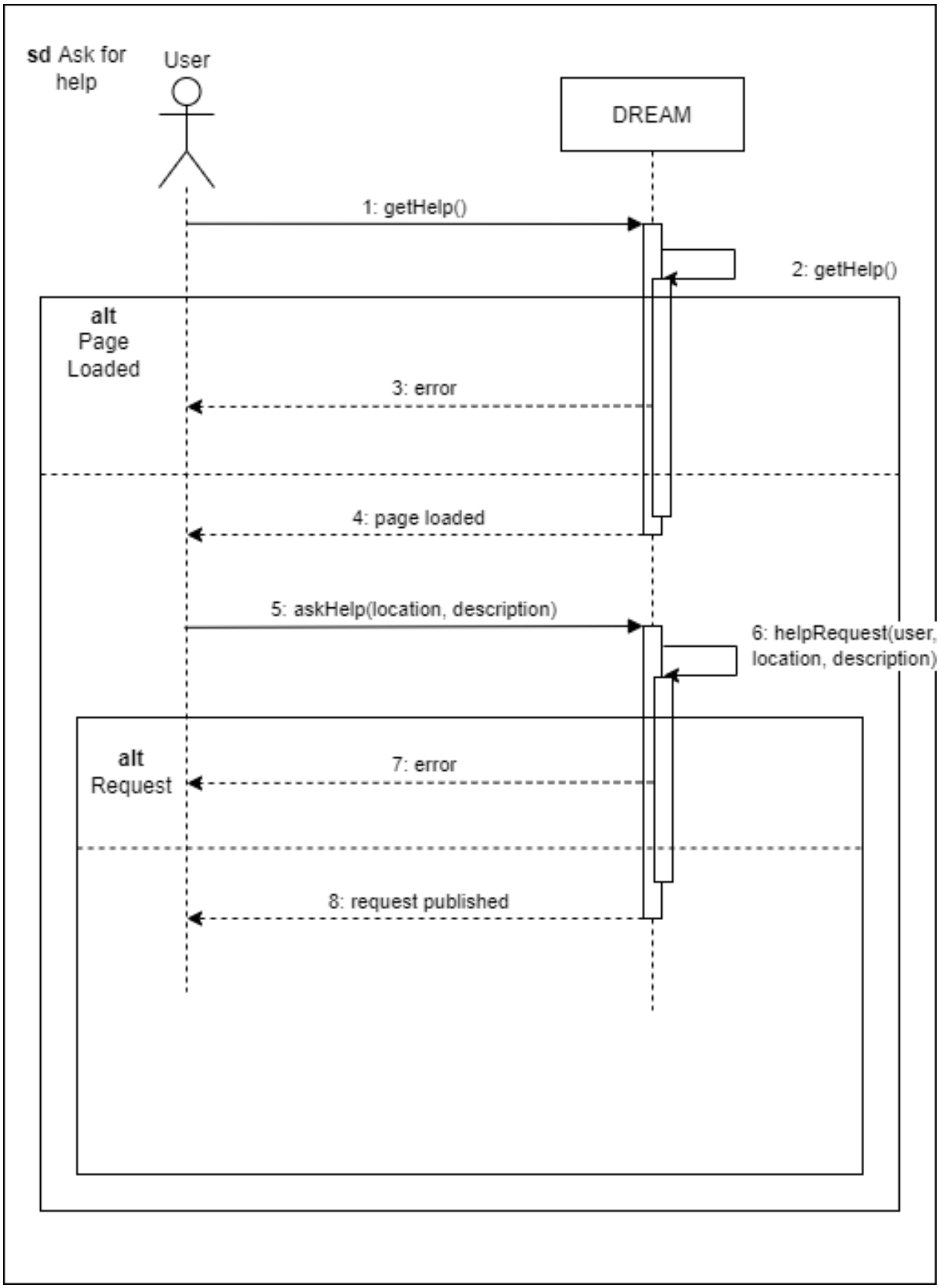

Figure 7: *Ask for help* sequence diagram.

Name	Reply to a request of help
ID	UC.7
Actors	Farmer, Agronomist
Entry conditions	• The platform is running • The user is running the application on his/her device • The user is logged into the system
Flow of events	i. The user accesses “Help” ii. System shows the list of all requests of help iii. The user browses through the requests of help iv. The user clicks on the request he/she wants to reply v. System shows the description of the request vi. System shows the username and location of the farmer who posted the request vii. The user types his/her solution to the problem viii. The user clicks “Reply”
Exit conditions	The user successfully submits his/her reply
Exceptions	• If the reply is not sent, the page and the reply will be reloaded. • If the form is empty, the system does not submit the data and highlights the missing part.

Table 7: *Reply to a request of help* use case description.

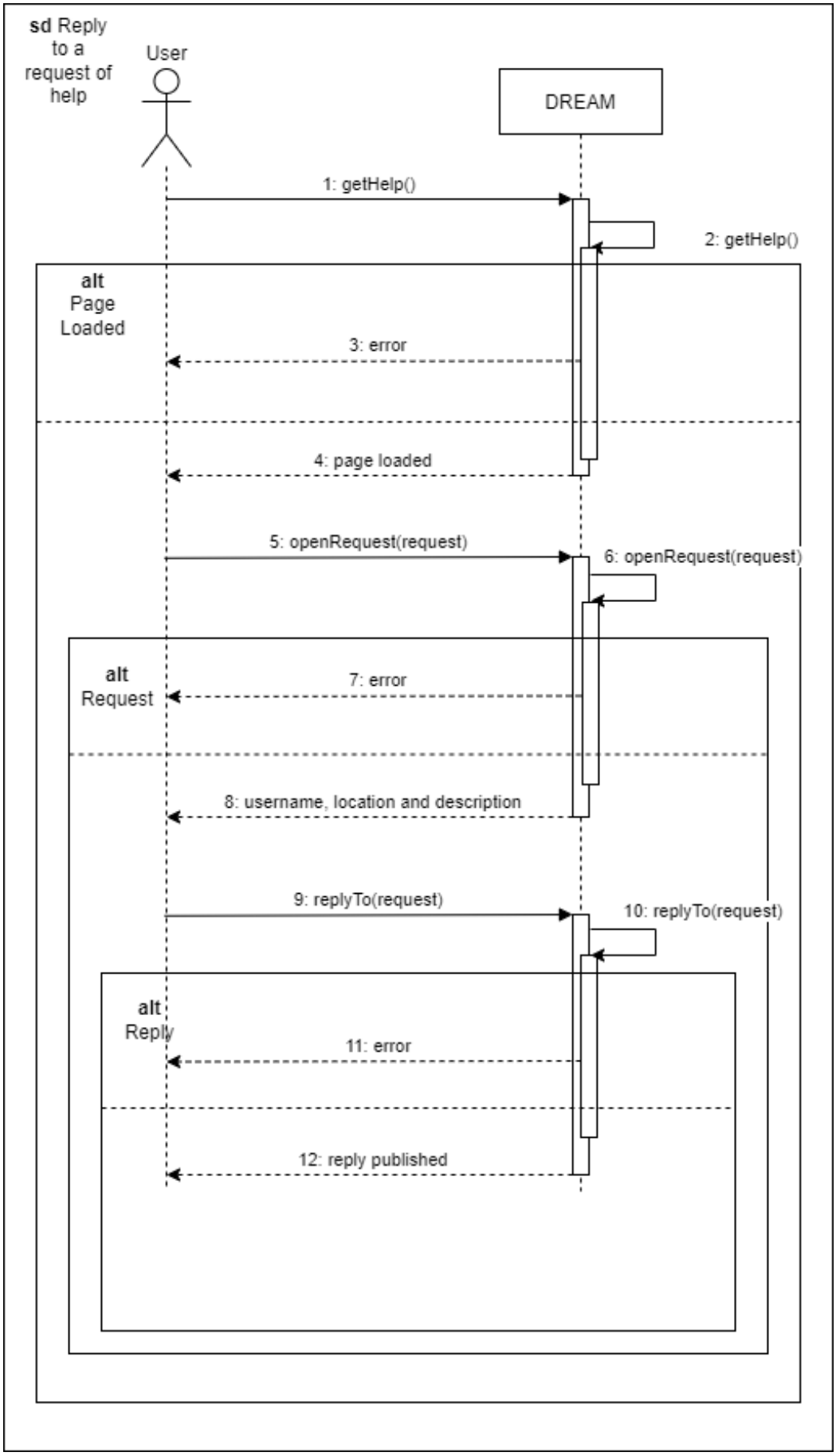

Figure 8: *Reply to a request of help* sequence diagram.

Name	Create a new discussion in the Forum
ID	UC.8
Actors	Farmer
Entry conditions	• The platform is running • The farmer is running the application on his/her device • The farmer is logged into the system
Flow of events	i. The farmer opens "Forum" ii. The farmer clicks on "Create new Discussion" iii. The farmer inserts a title for the discussion iv. The farmer types the topic of the discussion v. The farmer clicks on "Submit Discussion" vi. System creates the discussion
Exit conditions	The discussion is successfully submitted and the system updates "Forum"
Exceptions	• If the discussion is not submitted, the page and the form is reloaded. • If a field of the form is empty, the system does not submit the data and highlights the missing parts.

Table 8: *Create a new discussion in the Forum* use case description.

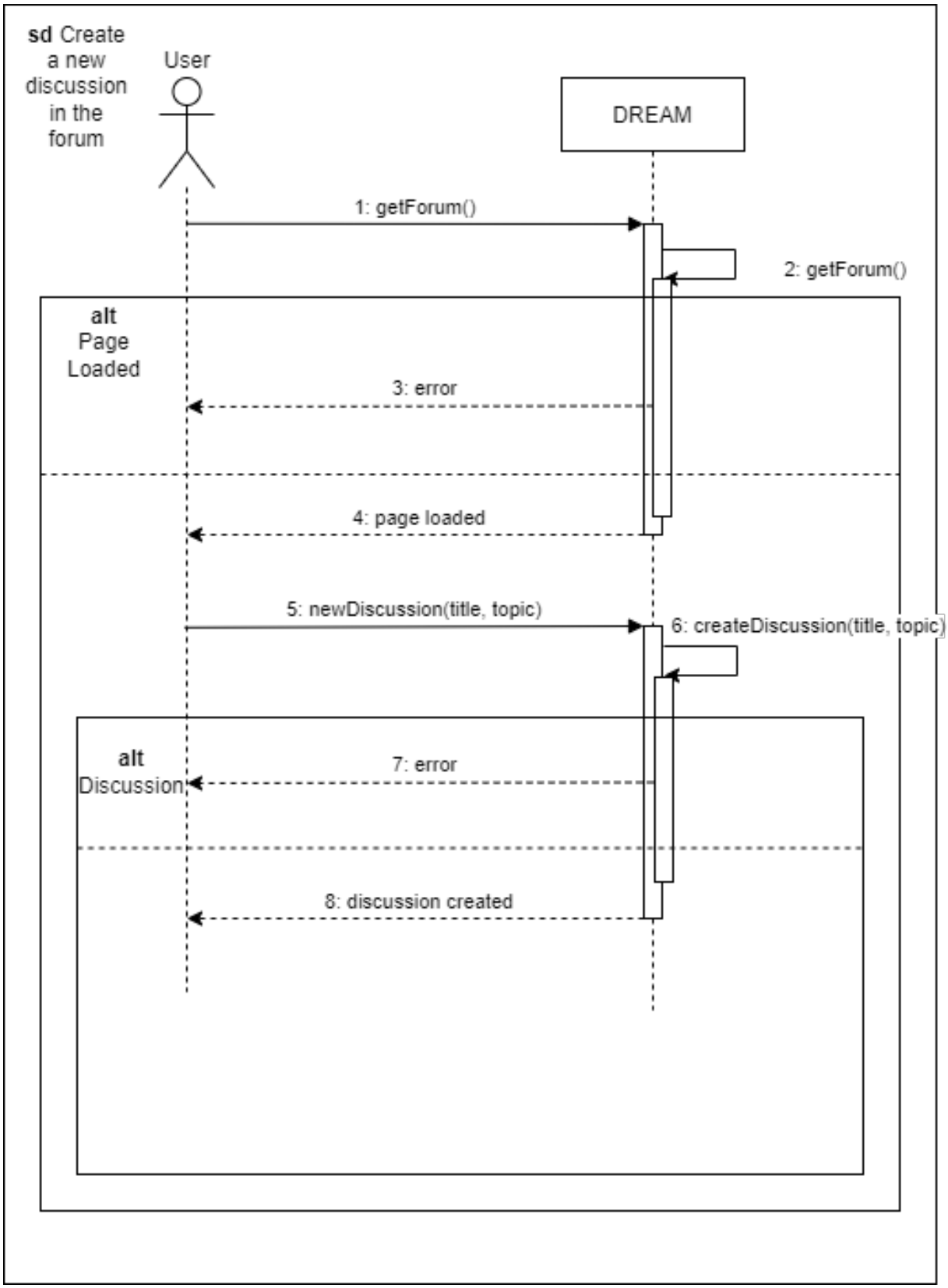

Figure 9: *Create a new discussion in the Forum* sequence diagram.

Name	Reply to a discussion in the Forum
ID	UC.9
Actors	Farmer
Entry conditions	• The platform is running • The farmer is running the application on his/her device • The farmer is logged into the system
Flow of events	i. The farmer opens “Forum” ii. System shows the list of all discussions in the forum iii. The farmer browses through the discussions already created iv. The farmer selects the topic he/she wants to reply v. System shows the replies to the topic vi. The farmer types his/her reply vii. The farmer clicks on “Send”
Exit conditions	The reply to the topic is successfully submitted
Exceptions	• If reply is not submitted, the discussion and the reply are reloaded. • If a field of the form is empty, the system does not submit the data and highlights the reply area.

Table 9: *Reply to a discussion in the Forum* use case description.

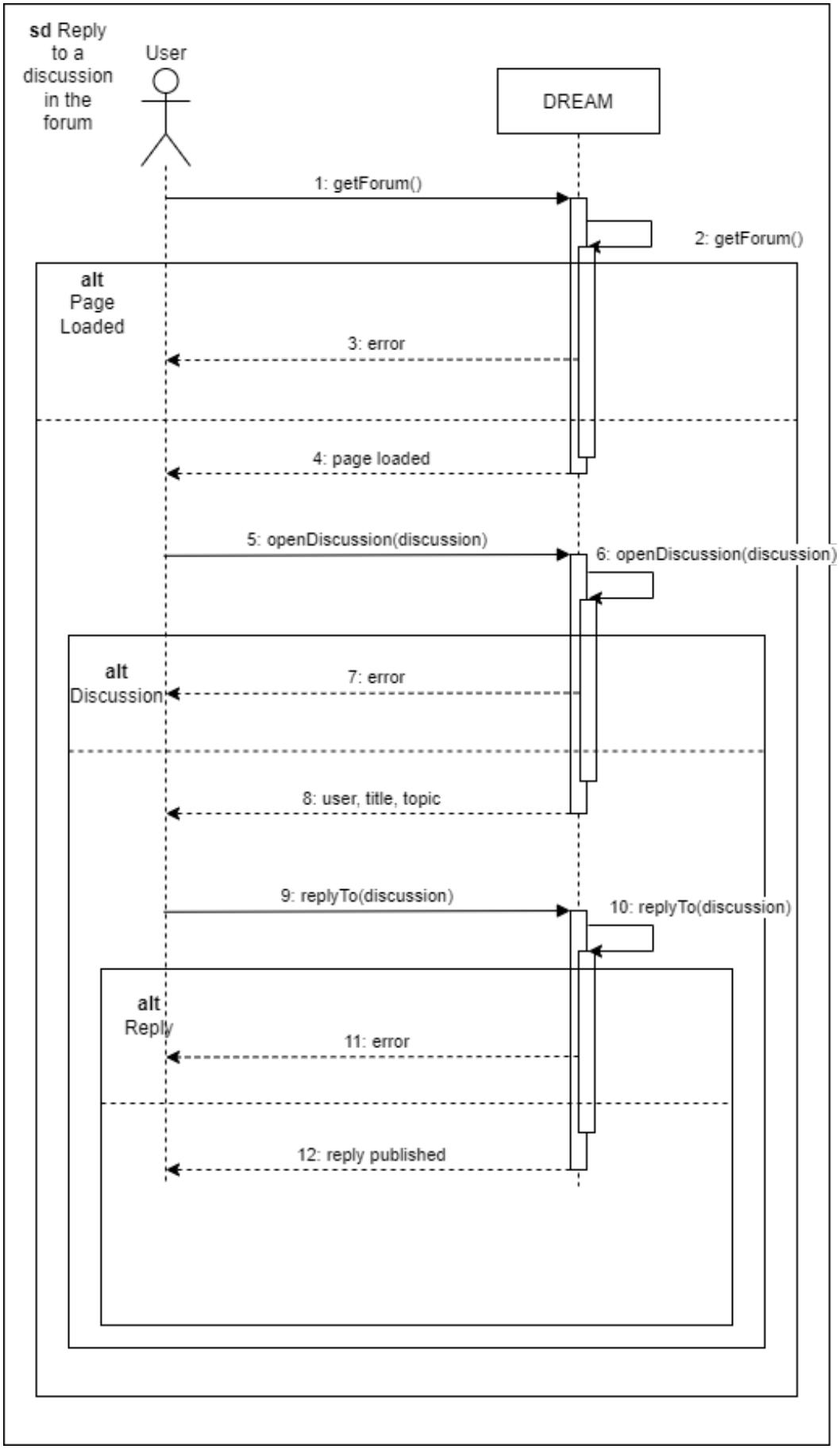

Figure 10: *Reply to a discussion in the Forum* sequence diagram.

Name	Get suggestions
ID	UC.10
Actors	Agronomist, Farmer
Entry conditions	• The platform is running • The user is running the application on his/her device • The user is logged into the system
Flow of events	i. The user opens "WikiFarm" ii. The user selects the zone iii. The user inserts the type of production iv. The user clicks on "Get Suggestions" v. System shows a list of all the compatible fertilizer, regarding the data inserted vi. System shows a list of all the compatible crops, regarding the data inserted
Exit conditions	The user closes DREAM or the user opens another section
Exceptions	• If for the request there is no solution, an error message is displayed, and the farmer is brought to the section of suggestions. • If a field of the form is empty, the system does not submit the data and highlights the missing parts.

Table 10: *Get suggestions* use case description.

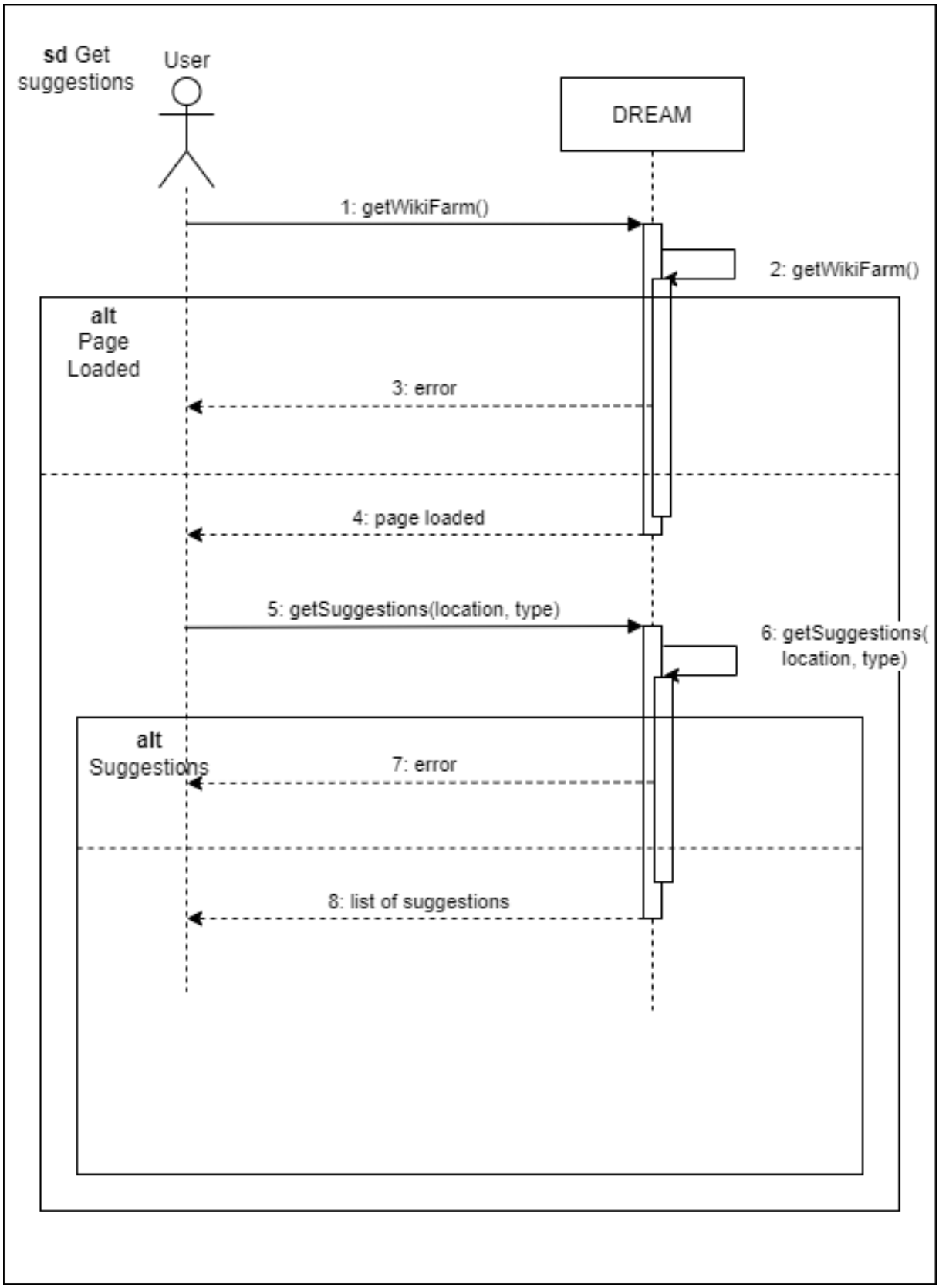

Figure 11: *Get suggestions* sequence diagram.

Name	Insert data of production
ID	UC.11
Actors	Farmer
Entry conditions	• The platform is running • The farmer is running the application on his/her device • The farmer is logged into the system
Flow of events	i. The farmer opens "Report Production" ii. The farmer types the type of production iii. The farmer types the crop planted iv. The farmer inserts the amount produced v. The farmer inserts the amount of land dedicated to the production vi. The farmer clicks on "Finish" vii. Using the sensors in the distribution system, DREAM retrieves the amount of water used by farmer viii. Using the sensors in the land, DREAM retrieves the humidity of the soil ix. System retrieves the farmer's location
Exit conditions	System combines data inserted by the farmer with the ones retrieved and it stores them into the database with the correct date.
Exceptions	• If a field of the form is empty, the system does not submit the data and highlights the missing parts. • If the farmer clicks on "Next", the system stores the data typed and loads a new page to insert another production report.

Table 11: *Insert data of production* use case description.

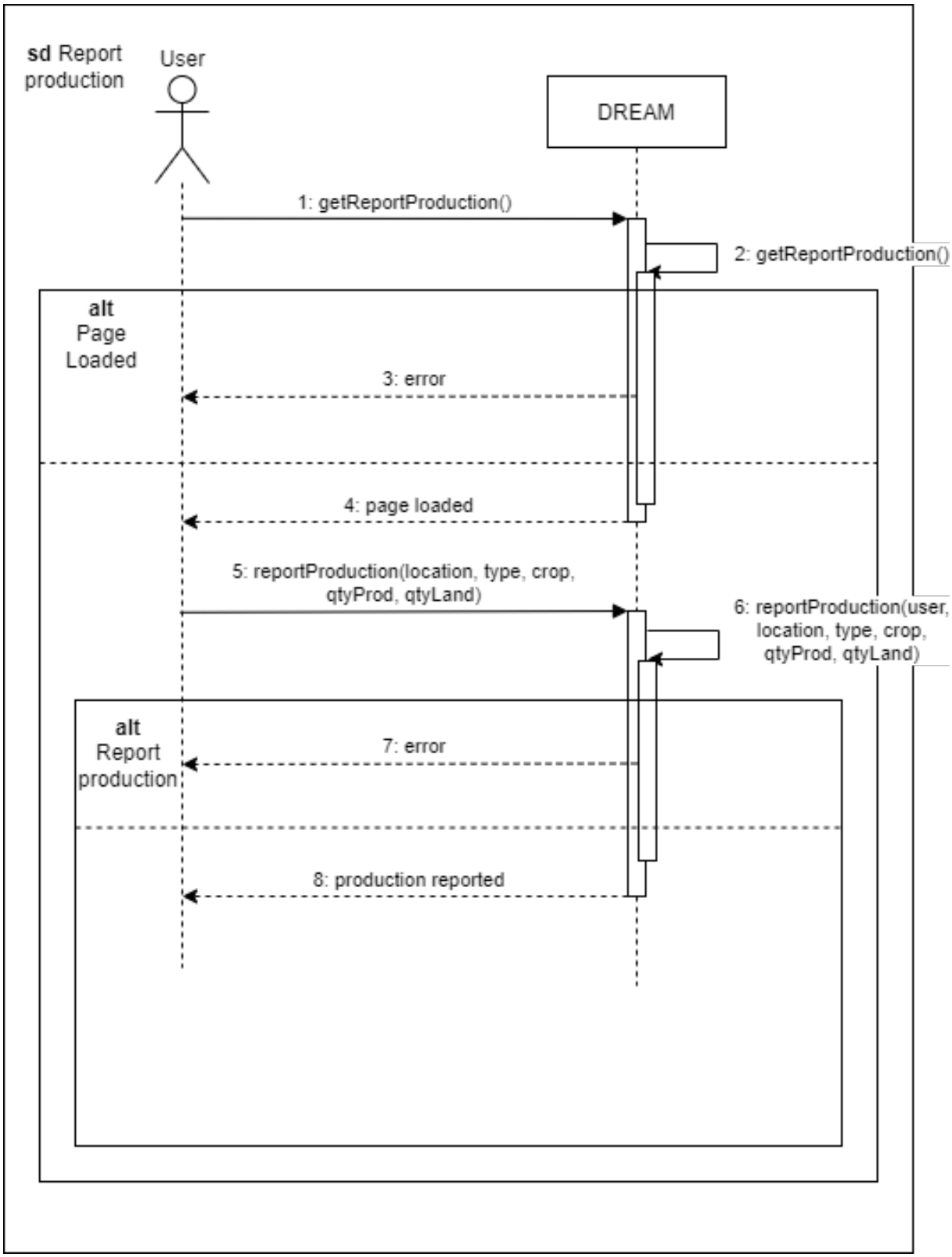

Figure 12: *Insert data of production* sequence diagram.

Name	Search data of production
ID	UC.12
Actors	TPM, Agronomist
Entry conditions	• The platform is running • The user is running the application on his/her device • The user is logged into the system
Flow of events	i. The user opens “Report Production” ii. The user types the username of the farmer he/she wants to visualize iii. System shows search in the database all the data inserted by the farmer iv. System shows search in the database all the data inserted by the farmer
Exit conditions	The user closes DREAM or opens another section of the system.
Exceptions	• If the farmer selected has not already reported data of production, the system shows a message: “The farmer selected has no data of production inserted”

Table 12: *ISearch data of production* use case description.

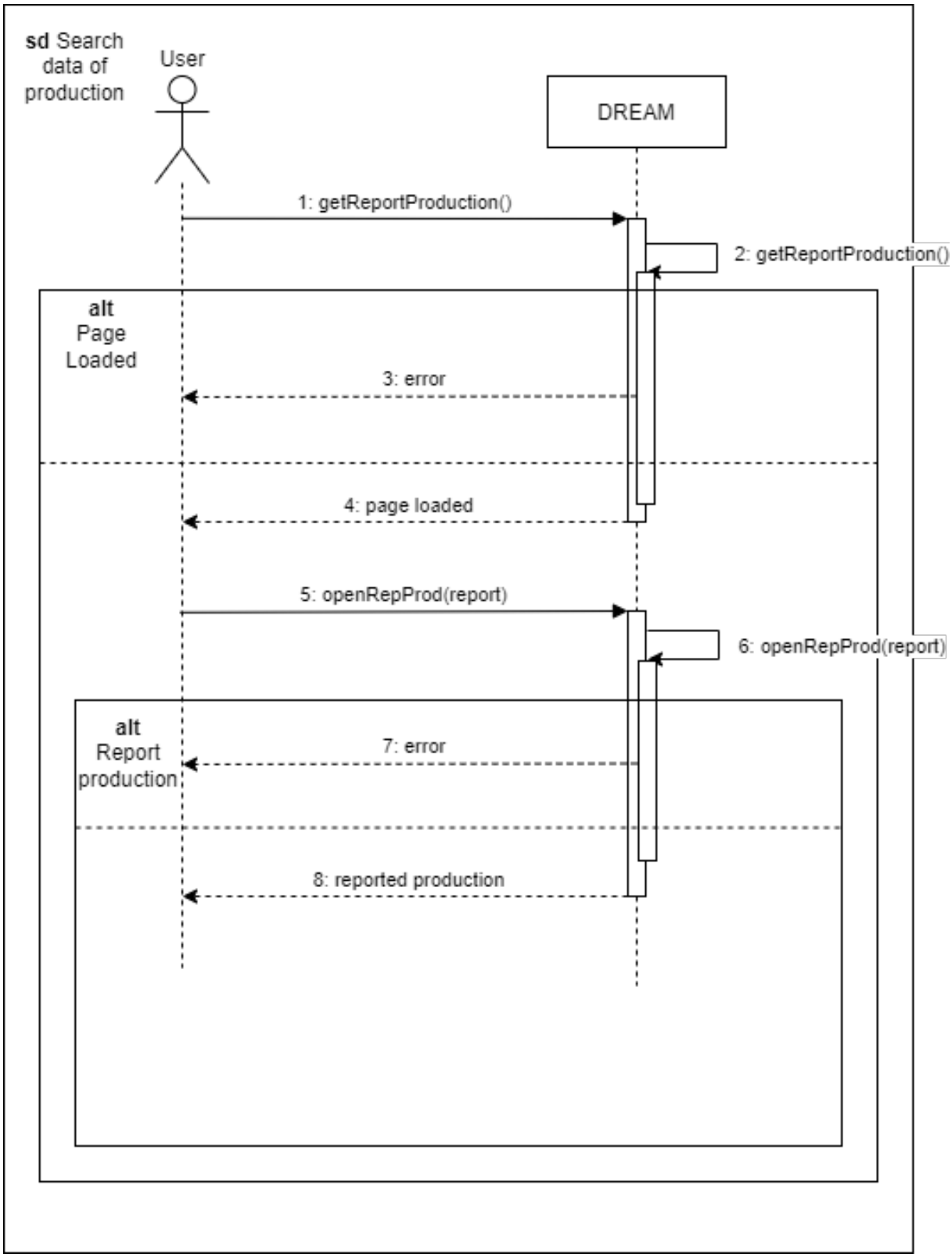

Figure 13: *Search data of production* sequence diagram.

Name	Generation of the ranking of the farmers
ID	UC.13
Actors	TPM, Agronomist
Entry conditions	• The platform is running • The user is running the application on the user's device • The user is logged into the system
Flow of events	i. The user opens "Ranking" ii. The user selects the zone of interest iii. The user clicks "Search" iv. System computes the evaluation of the performance of all the farmers in the selected area v. System shows the list of all farmers from the best performing to the worst
Exit conditions	The user closes DREAM or opens another section of the system
Exceptions	• In case of ties, farmers are ordered alphabetically. • If no area is selected, the ranking will be based on all the farmers in Telangana.

Table 13: *Generation of the ranking of the farmers* use case description.

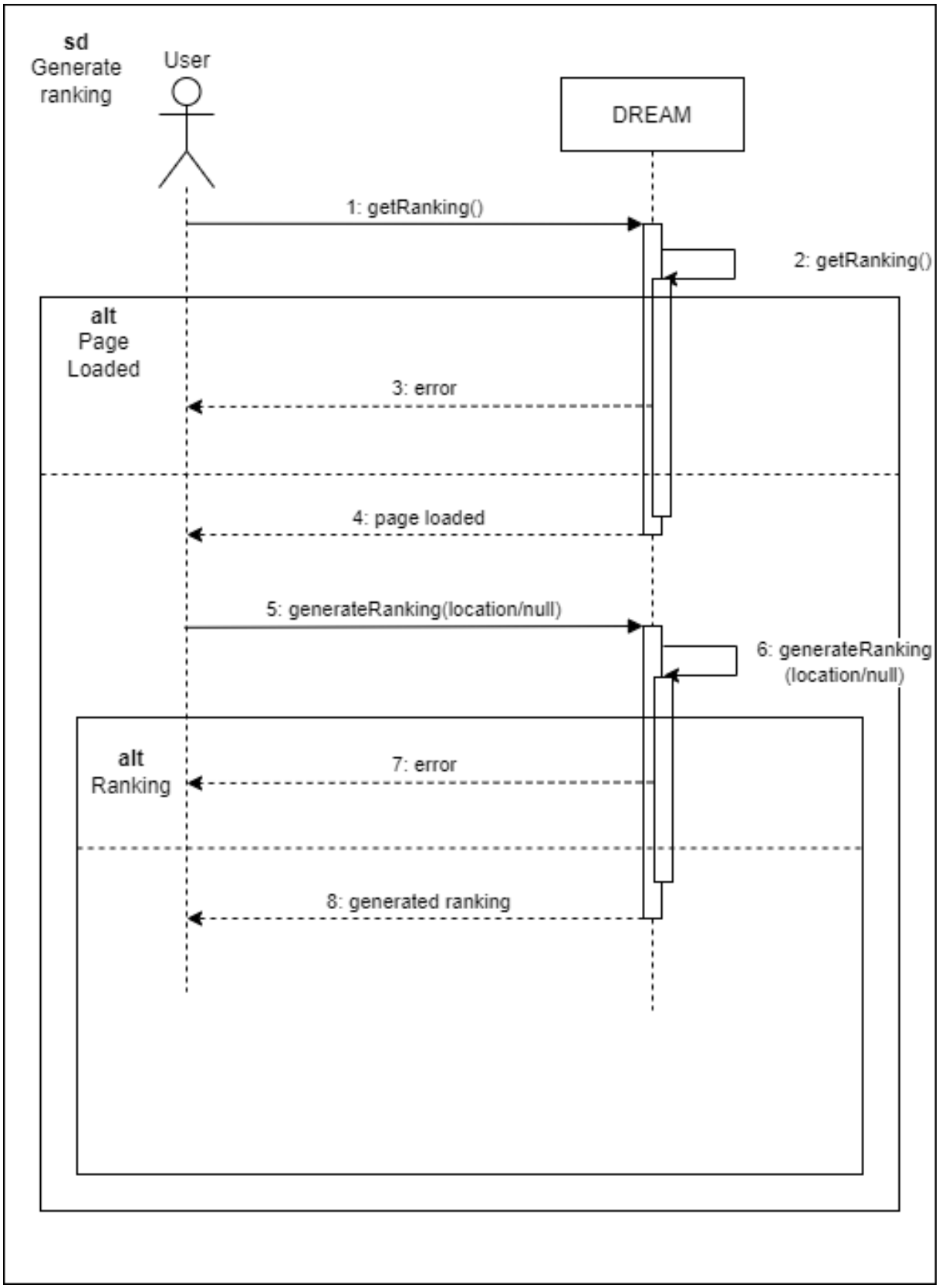

Figure 14: *Generation of the ranking of the farmers* sequence diagram.

Name	Checking the weather forecast
ID	UC.14
Actors	Farmer, Agronomist
Entry conditions	• The platform is running • The user is running the application on the user's device • The user is logged into the system
Flow of events	i. The user opens "Weather" ii. The user selects the zone of interest iii. The user selects the temporal slot for the forecast iv. The user clicks "Search" v. System accesses the Telangana's weather portal vi. System requests forecast for the time slot and zone chosen by the user vii. System shows the weather forecast chosen
Exit conditions	The user closes the page or decides to do another operation.
Exceptions	• In case of service disruption caused by bad weather, the system shows an error message, and the user is brought to the home page of the system. • If the time slot is greater than 7 days, the system shows the average forecast for the area and period selected.

Table 14: *Checking the weather forecast* use case description.

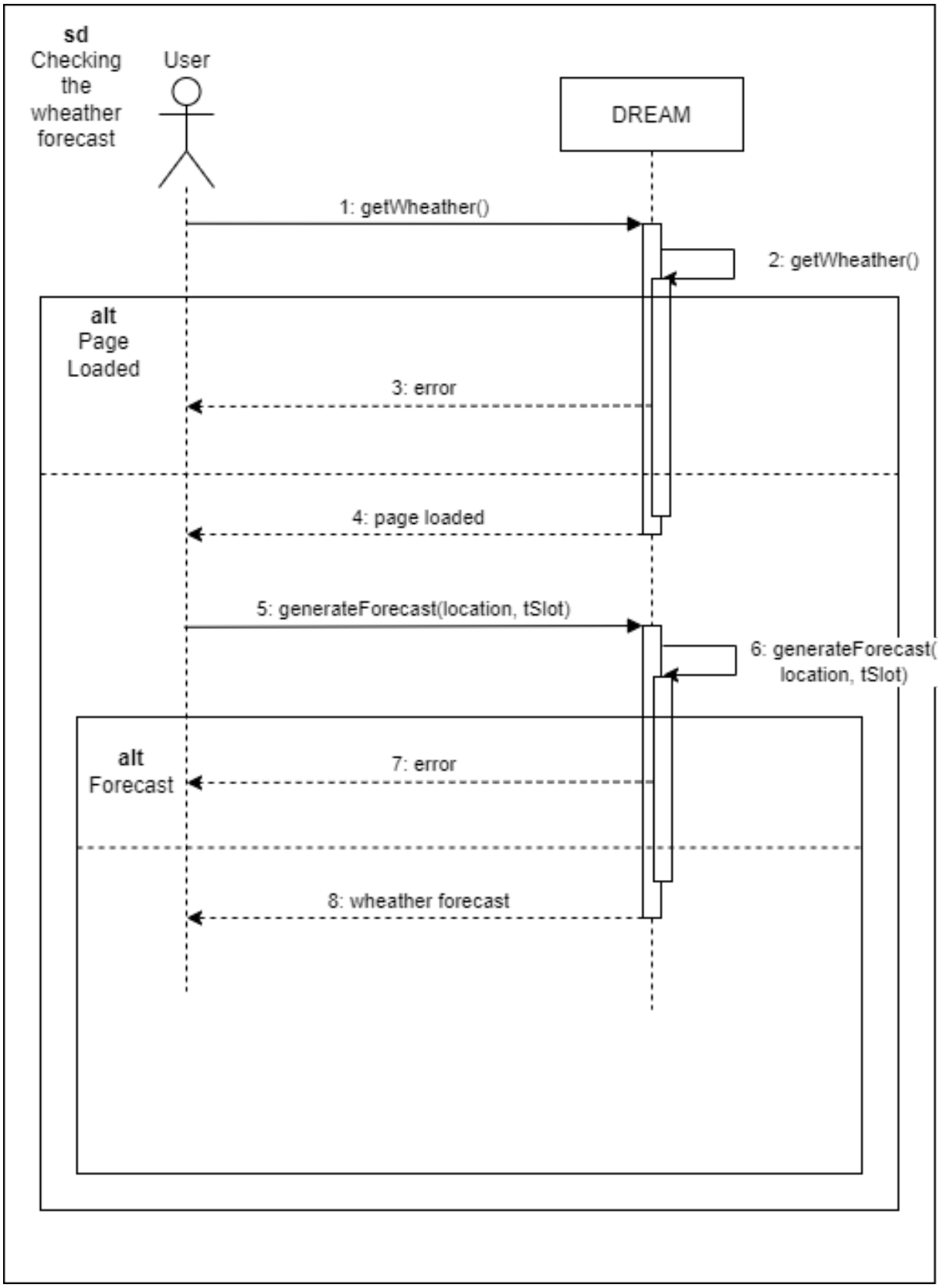

Figure 15: *Checking the weather forecast* sequence diagram.

Name	Checking the schedule
ID	UC.15
Actors	Agronomist
Entry conditions	• The platform is running • The Agronomist is running the application • The Agronomist is logged into the system
Flow of events	i. The agronomist opens “Agenda” ii. The agronomist selects the day of interest iii. The agronomist clicks “Search” iv. System shows the list of all the events planned for that day v. The agronomist selects the event he/she wants to visualize vi. System shows the details for the event selected
Exit conditions	The agronomist closes the page or decides to do another operation.
Exceptions	• If in the date selected there are no events planned, the system will show a blank page with the text “No events for the date selected”.

Table 15: *Checking the schedule* use case description.

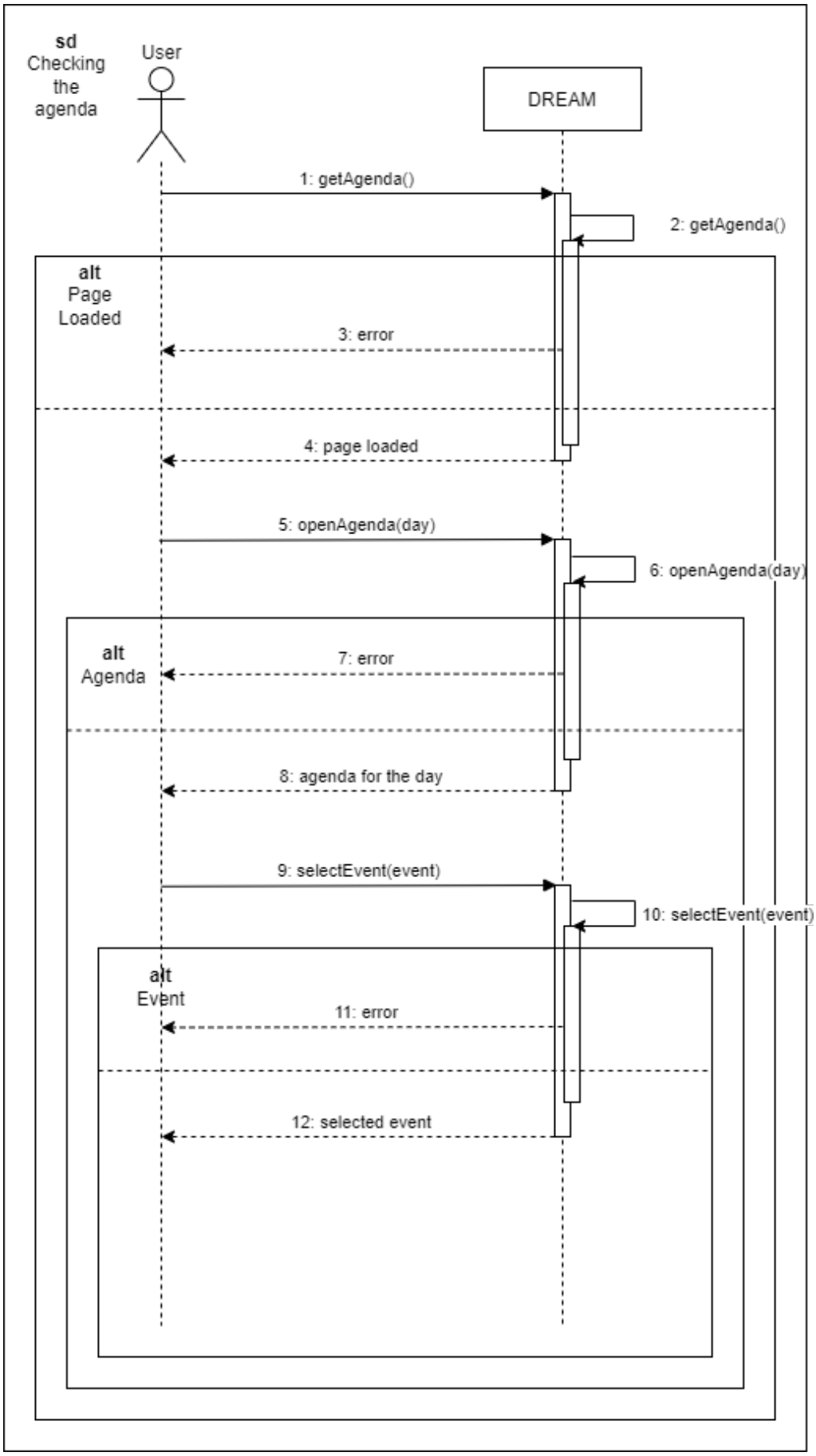

Figure 16: *Checking the schedule* sequence diagram.

Name	Reporting a visit to a farmer
ID	UC.16
Actors	Agronomist
Entry conditions	• The platform is running • The Agronomist is running the application • The Agronomist is logged into the system
Flow of events	i. The agronomist opens “Agenda” ii. The agronomist selects the current day iii. The agronomist clicks “Search” iv. System shows the list of all the events planned for that day v. The agronomist selects the event he/she has just concluded vi. System shows the details of the event selected vii. The agronomist clicks “Upload Report” viii. The agronomist types the report of the visit ix. The agronomist clicks “Visit concluded”
Exit conditions	The report of the visit is successfully stored in the system.
Exceptions	• If the upload of the report fails an error message is shown and the agronomist is asked to repeat the operation.

Table 16: *Reporting a visit to a farmer* use case description.

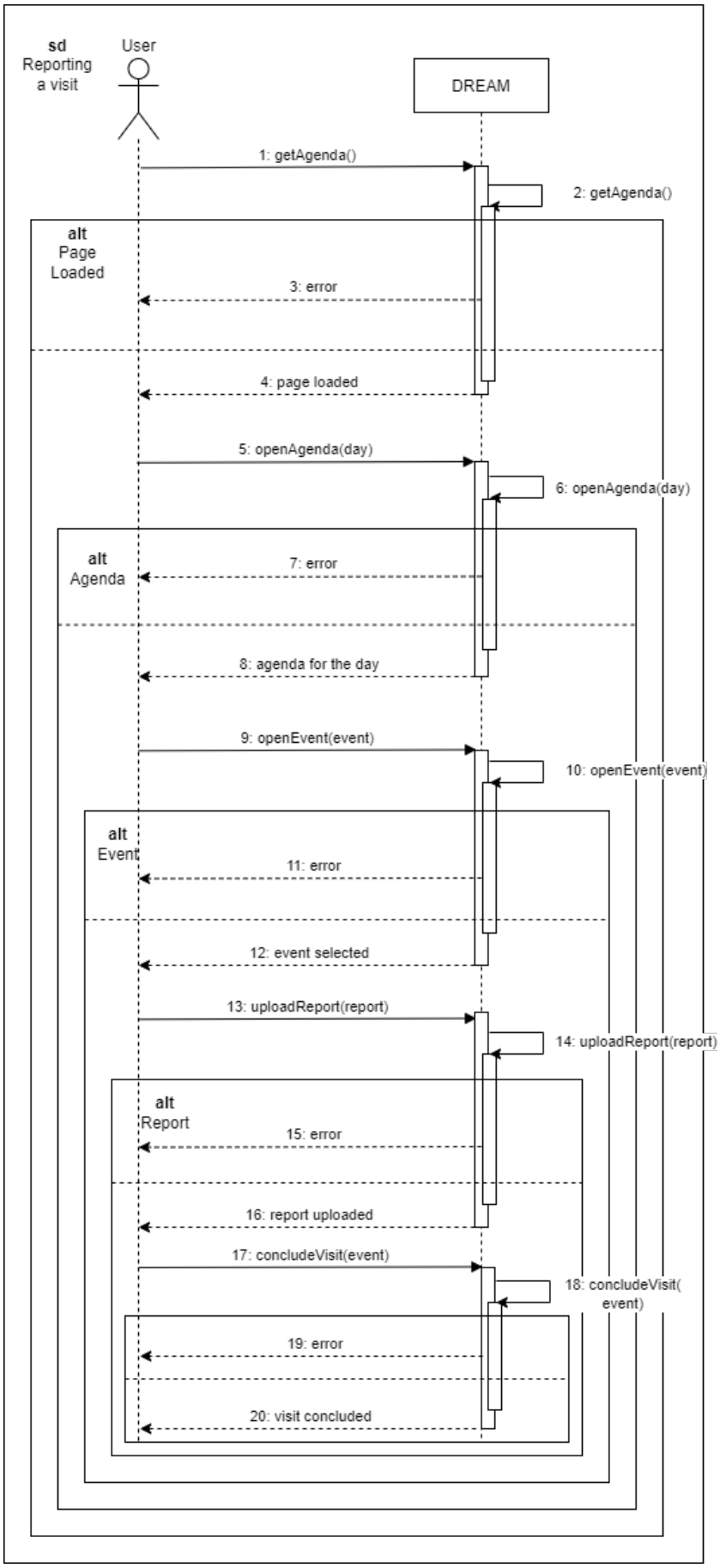

Figure 17: *Reporting a visit to a farmer* sequence diagram.

Name	Changing the schedule
ID	UC.17
Actors	Agronomist
Entry conditions	• The platform is running • The Agronomist is running the application • The Agronomist is logged into the system
Flow of events	i. The agronomist opens “Agenda” ii. The agronomist selects the day iii. The agronomist clicks “Search” iv. System shows the list of all the events planned for that day v. The agronomist selects the event he/she has not done vi. The agronomist clicks “Change Date” vii. The agronomist selects the new date for the visit viii. The agronomist selects the new time for the visit ix. The agronomist clicks “Upload changes”
Exit conditions	The system successfully updates the agronomist’s “Agenda”
Exceptions	• If the visit is overlapping with another event, a message is displayed: “There is another event for that time and date, please choose another slot.” The system brings the agronomist to the page where he/she could modify the new date and time.

Table 17: *Changing the schedule* use case description.

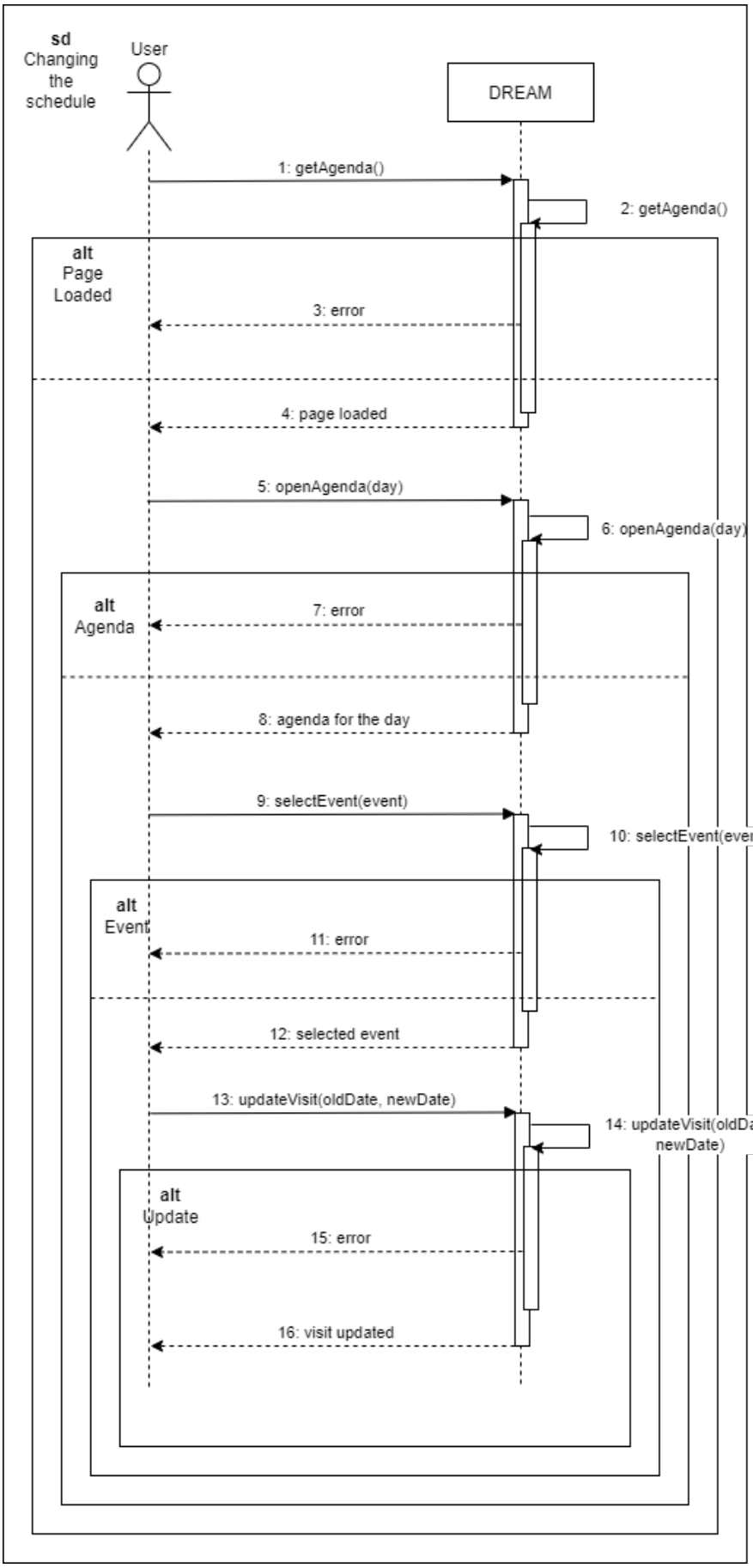

Figure 18: *Changing the schedule* sequence diagram.

Name	Creating a new event in the schedule
ID	UC.18
Actors	Agronomist
Entry conditions	• The platform is running • The Agronomist is running the application • The Agronomist is logged into the system
Flow of events	i. The agronomist opens “Agenda” ii. The agronomist selects a day iii. The agronomist clicks “Search” iv. System shows the list of all the events planned for that day v. The agronomist clicks “New Event” vi. The agronomist inserts a title for the event vii. The agronomist selects the time for the visit viii. The agronomist inserts the username of the farmer he/she has to visit ix. The agronomist clicks “Create event” x. System retrieves the location of the farmer xi. System adds the location to the event
Exit conditions	System successfully creates the event and successfully updates the agronomist’s “Agenda”
Exceptions	• If the visit is overlapping with another event, a message is displayed: “There is another event for that time and date, please choose another slot.” The system brings the agronomist to the page where he/she could modify the new date and time.

Table 18: *Creating a new event in the schedule* use case description.

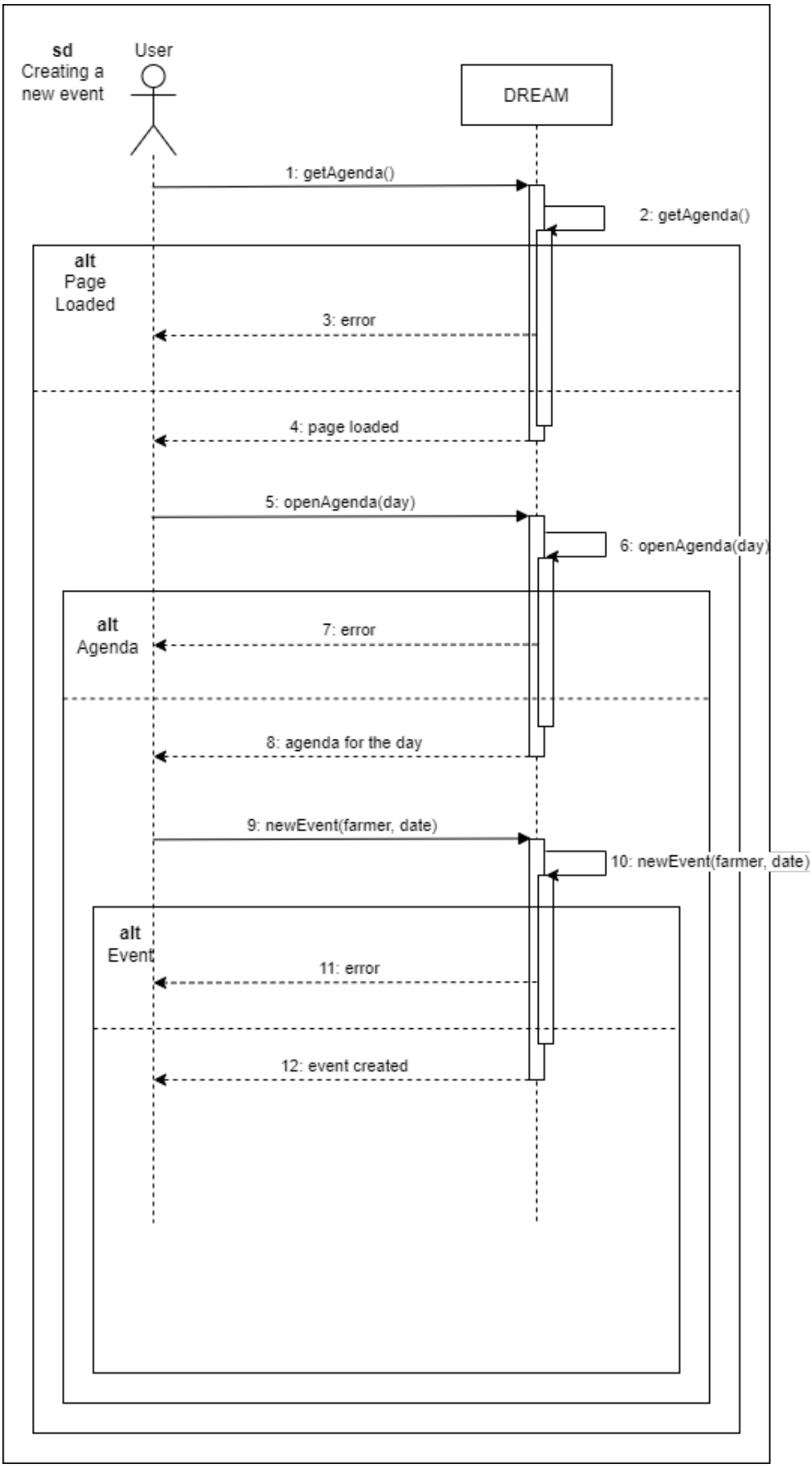

Figure 19: *Creating a new event in the schedule* sequence diagram.

Name	Visualize the report of an agronomist's visit
ID	UC.19
Actors	TPM
Entry conditions	• The platform is running • The TPM is running the application • The TPM is logged into the system
Flow of events	i. The TPM opens "Visualize Initiatives" ii. The TPM types the username of the farmer iii. he TPM clicks "Search" iv. System shows a list with all the reports associated with the farmer selected v. The TPM selects the report he/she wants to visualize vi. System shows the selected report
Exit conditions	The TPM closes DREAM or selects another operation.
Exceptions	• If the farmer selected has not yet associated with a report, the system shows a message:" The farmer selected has no reports".

Table 19: *Visualize the report of an agronomist's visit* use case description.

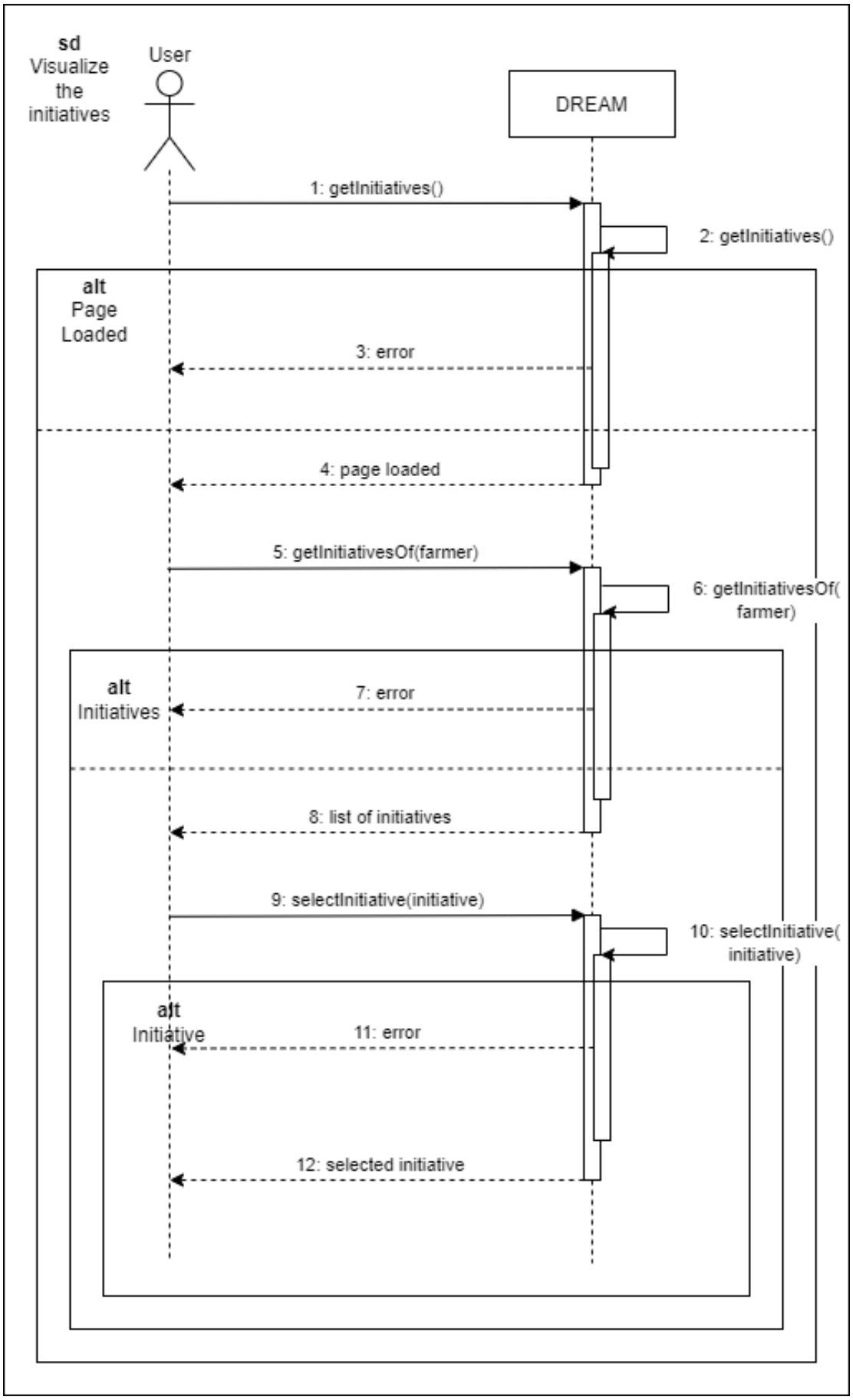

Figure 20: *Visualize the report of an agronomist's visit* sequence diagram.

3.3 Functional Requirements

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

3.3.1 Requirements

R.1 Data in the system are never deleted

R.2 System shall retrieve the data from a sensor of humidity in a land

R.3 System shall update data of humidity in land every hour

R.4 System shall retrieve the amount of water supplied to a farmer

R.5 System shall weekly update the amount of water consumed by a farmers

R.6 System shall compute a ranking of the farmers' performances

R.7 System shall allow any user to communicate with any other user using the chat function

R.8 System shall allow a farmer to insert his/her type of production

R.9 System shall store the location of each farmer

R.10 System shall store all the types of production of each farmer

R.11 System shall allow a farmer to filter the type of data he/she wants to visualize

R.12 System shall receive information about local weather forecast

R.13 System shall allow a farmer to visualize the weather forecast

R.14 System shall store all the fertilizers available in Telangana

R.15 System shall suggest fertilizers based on the retrieved data

R.16 System shall store all crops that can be planted in Telangana

R.17 System shall suggest crops based on the retrieved data

R.18 System shall allow a farmer to insert his/her data of production

R.19 System shall store all data of production of farmers

R.20 System shall allow a farmer to insert data about a problem that he/she faced

R.21 System shall store all data of problems faced by farmers

R.22 System shall allow a resilient farmer to share best practices with other farmers

R.23 System shall allow a farmer to ask for help

R.24 System shall allow a farmer to reply to a request of help

R.25 System shall allow a farmer to create a new discussion in the Forum

R.26 System shall allow a farmer to reply to a discussion in the Forum

R.27 System shall allow an agronomist to select the area he/she is responsible of

R.28 System shall allow an agronomist to visualize a request of help in the area he/she is responsible of

R.29 System shall allow an agronomist to reply to a request of help in the area he/she is responsible of

R.30 System shall allow an agronomist to visualize the weather forecast for the area he/she is responsible of

R.31 System shall allow an agronomist to visualize the performances of farmers in the area he/she is responsible of

R.32 System shall allow an agronomist to visualize data of production inserted by farmers

R.33 System shall allow an agronomist to visualize problems reported by farmers in the area he/she is responsible of

R.34 System shall store the daily plan of all agronomists

R.35 System shall allow an agronomist to visualize his/her daily plan

R.36 System shall allow an agronomist to update his/her daily plan

R.37 System shall allow an agronomist to confirm his/her daily plan

R.38 System shall allow an agronomist to insert the deviations from his/her original daily plan

R.39 System shall store all the initiatives took by a farmer and an agronomist

R.40 System shall allow an agronomist to insert an initiative he/she proposed to a farmer

R.41 System shall allow TPM to visualize the ranking of farmers' performances

R.42 System shall allow TPM to visualize data about the production of a farmer

R.43 System shall allow TPM to visualize the problems reported by farmers

R.44 System shall allow TPM to visualize initiatives taken by agronomist during his/her visits

R.45 System shall allow TPM to ask to a resilient farmer to share best practices

R.46 System shall allow TPM to signal bad farmers to Agronomists

R.47 System shall allow any user to access through a browser

R.48 System shall allow any user to log in using his/her credentials

R.49 System shall give access to any user who have the right permission. For example, a farmer could not access to "Agenda"

3.3.2 Mapping Goals on Requirements and Domain Assumptions

G.1 Make available/Share the weather forecasts collected by Telangana

R1 - Data in the system are never deleted

R11 - System shall allow a farmer to filter the type of data he/she wants to visualize

R12 - System shall receive information about local weather forecast

R13 - System shall allow a farmer to visualize the weather forecast

R30 - System shall allow an agronomist to visualize the weather forecast for the area he/she is responsible of

DA1 - All Telangana's people, who interact with system, have an internet connection

DA2 - All Telangana's people, who interact with system, have access to a browser

DA18 - Weather forecast for a specific zone is always available

G.2 Create an anticipatory model for Telangana's food system

R1 - Data in the system are never deleted

R2 - System shall retrieve the data from a sensor of humidity in a land

R3 - System shall update data of humidity in land every hour

R4 - System shall retrieve the amount of water supplied to a farmer

R5 - System shall weekly update the amount of water consumed by a farmers

R6 - System shall compute a ranking of the farmers' performances

R12 - System shall receive information about local weather forecast

R16 - System shall store all crops that can be planted in Telangana

R19 - System shall store all data of production of farmers

DA1 - All Telangana's people, who interact with system, have an internet connection

DA2 - All Telangana's people, who interact with system, have access to a browser

DA3 - Every time a farmer incurs into a problem, he/she inserts the details in the system

DA4 - At the end of each month, farmers insert into the system the result of their production

DA13 - Farmers are ranked based on their productivity per acre

DA14 - All lands in Telangana have a sensor of humidity of the soil

DA15 - Water distribution system has sensor to measure the amount of water given to each land

DA16 - Water distribution sensors never stop to work

DA17 - Humidity sensors in a land never stop to work

G.3 Allow the communication between 2 users

R7 - System shall allow any user to communicate with any other user using the chat function

R22 - System shall allow a resilient farmer to share best practices with other farmers

R23 - System shall allow a farmer to ask for help

R24 - System shall allow a farmer to reply to a request of help

R25 - System shall allow a farmer to create a new discussion in the Forum

R26 - System shall allow a farmer to reply to a discussion in the Forum

R28 - System shall allow an agronomist to visualize a request of help in the area he/she is responsible of

R29 - System shall allow an agronomist to reply to a request of help in the area he/she is responsible of

R45 - System shall allow TPM to ask to a resilient farmer to share best practices

R46 - System shall allow TPM to signal bad farmers to Agronomists

DA1 - All Telangana's people, who interact with system, have an internet connection

DA2 - All Telangana's people, who interact with system, have access to a browser

DA24 - Any user knows the correct username he/she wants to contact using "Chat"

G.4 Support the agricultural work in Telangana

G.4.1 Support the work of the farmers

R7 - System shall allow any user to communicate with any other user using the chat function

R8 - System shall allow a farmer to insert his/her type of production

R9 - System shall store the location of each farmerR9 - System shall store the location of each farmer

R10 - System shall store all the types of production of each farmer

R11 - System shall allow a farmer to filter the type of data he/she wants to visualize

R13 - System shall allow a farmer to visualize the weather forecast

R14 - System shall store all the fertilizers available in Telangana

R15 - System shall suggest fertilizers based on the retrieved data

R16 - System shall store all crops that can be planted in Telangana

R17 - System shall suggest crops to farmers based on the retrieved data

R18 - System shall allow a farmer to insert his/her data of production

R19 - System shall store all data of production of farmers

R20 - System shall allow a farmer to insert data about a problem that he/she faced

R21 - System shall store all data of problems faced by farmers

R22 - System shall allow a resilient farmer to share best practices with other farmers

R23 - System shall allow a farmer to ask for help

R24 - System shall allow a farmer to reply to a request of help

R25 - System shall allow a farmer to create a new discussion in the Forum

R26 - System shall allow a farmer to reply to a discussion in the Forum

R47 - System shall allow any user to access through a browser

R48 - System shall allow any user to log in using his/her credentials

R49 - System shall give access to any user who have the right permission. For example, a farmer could not access to "Agenda"

DA1 - All Telangana's people, who interact with system, have an internet connection

DA2 - All Telangana's people, who interact with system, have access to a browser

DA3 - Every time a farmer incurs into a problem, he/she inserts the details in the system

DA4 - At the end of each month, farmers insert into the system the result of their production

DA10 - All the farmers have a banking account where they can receive incentives

DA11 - All resilient farmers receive special incentives from TPM

DA12 - All good farmers provide best practices to TPM

DA18 - Weather forecast for a specific zone is always available

DA20 - Farmers always insert correct data about their production

DA21 - The quantity produced by a farmer is quantified in tons

DA22 - The lands are measured in acres

DA23 - All the users know their credentials given by Telangana state

DA24 - Any user knows the correct username he/she wants to contact using "Chat"

DA25 - All the users do not forget their credentials given by Telangana state

DA26 - Each farmer is associated with only one farm

DA27 - Every farmer stays in the same location

G.4.2 Support the work of the agronomists

R2 - System shall retrieve the data from a sensor of humidity in a land

R3 - System shall update data of humidity in land every hour

R4 - System shall retrieve the amount of water supplied to a farmer

R5 - System shall weekly update the amount of water consumed by a farmers

R6 - System shall compute a ranking of the farmers' performances

R7 - System shall allow any user to communicate with any other user using the chat function

R14 - System shall store all the fertilizers available in Telangana

R15 - System shall suggest fertilizers based on the retrieved data

R16 - System shall store all crops that can be planted in Telangana

R17 - System shall suggest crops based on the retrieved data

R21 - System shall store all data of problems faced by farmers

R27 - System shall allow an agronomist to select the area he/she is responsible of

R28 - System shall allow an agronomist to visualize a request of help in the area he/she is responsible of

R29 - System shall allow an agronomist to reply to a request of help in the area he/she is responsible of

R30 - System shall allow an agronomist to visualize the weather forecast for the area he/she is responsible of

R31 - System shall allow an agronomist to visualize the performances of farmers in the area he/she is responsible of

R32 - System shall allow an agronomist to visualize data of production inserted by farmers

R33 - System shall allow an agronomist to visualize problems reported by farmers in the area he/she is responsible of

R34 - System shall store the daily plan of all agronomists

R35 - System shall allow an agronomist to visualize his/her daily plan

R36 - System shall allow an agronomist to update his/her daily plan

R37 - System shall allow an agronomist to confirm his/her daily plan

R38 - System shall allow an agronomist to insert the deviations from his/her original daily plan

R39 - System shall store all the initiatives took by a farmer and an agronomist

R40 - System shall allow an agronomist to insert an initiative he/she proposed to a farmer

R47 - System shall allow any user to access through a browser

R48 - System shall allow any user to log in using his/her credentials

R49 - System shall give access to any user who have the right permission. For example, a farmer could not access to "Agenda"

DA1 - All Telangana's people, who interact with system, have an internet connection

DA2 - All Telangana's people, who interact with system, have access to a browser

DA3 - Every time a farmer incurs into a problem, he/she inserts the details in the system

DA4 - At the end of each month, farmers insert into the system the result of their production

DA5 - Agronomists have a vehicle to visit farmers

DA6 - Agronomists visit only the farmers who are in their daily schedule

DA7 - An agronomist could not complete his/her daily schedule

DA8 - Agronomists visit more than two times a year only the farmers who are performing badly

DA9 - During the visit of a farmer by an agronomist, he/she proposes suggestions to improve the production based on the data collected since the last visit

DA13 - Farmers are ranked based on their productivity per acre

DA18 - Weather forecast for a specific zone is always available

DA19 - Agronomists know the number of times each farmer has been visited

DA21 - The quantity produced by a farmer is quantified in tons

DA22 - The lands are measured in acres

DA23 - All the users know their credentials given by Telangana state

DA24 - Any user knows the correct username he/she wants to contact using “Chat”

DA25 - All the users do not forget their credentials given by Telangana state

DA26 - Each farmer is associated with only one farm

DA27 - Every farmer stays in the same location

G.4.3 Support the work of the TPM

R2 - System shall retrieve the data from a sensor of humidity in a land

R3 - System shall update data of humidity in land every hour

R4 - System shall retrieve the amount of water supplied to a farmer

R5 - System shall weekly update the amount of water consumed by a farmers

R7 - System shall allow any user to communicate with any other user using the chat function

R21 - System shall store all data of problems faced by farmers

R41 - System shall allow TPM to visualize the ranking of farmers’ performances

R42 - System shall allow TPM to visualize data about the production of a farmer

R43 - System shall allow TPM to visualize the problems reported by farmers

R44 - System shall allow TPM to visualize initiatives taken by agronomist during his/her visits

R45 - System shall allow TPM to ask to a resilient farmer to share best practices

R46 - System shall allow TPM to signal bad farmers to Agronomists

R47 - System shall allow any user to access through a browser

R48 - System shall allow any user to log in using his/her credentials

R49 - System shall give access to any user who have the right permission. For example, a farmer could not access to “Agenda”

DA1 - All Telangana’s people, who interact with system, have an internet connection

DA2 - All Telangana’s people, who interact with system, have access to a browser

DA11 - All resilient farmers receive special incentives from TPM

DA12 - All good farmers provide best practices to TPM

DA13 - Farmers are ranked based on their productivity per acre

DA18 - Weather forecast for a specific zone is always available

DA21 - The quantity produced by a farmer is quantified in tons

DA22 - The lands are measured in acres

DA23 - All the users know their credentials given by Telangana state

DA24 - Any user knows the correct username he/she wants to contact using “Chat”

DA25 - All the users do not forget their credentials given by Telangana state

DA26 - Each farmer is associated with only one farm

DA27 - Every farmer stays in the same location

3.3.3 Mapping Scenarios on Use Cases

Scenario F1 - Bug infestation

UC1 - Platform login

UC4 - Report of a problem

UC6 - Ask for help

Scenario F2 - The kind farmer

UC1 - Platform login

UC7 - Reply to a request of help

UC9 - Reply to a discussion in the forum

Scenario F3 - The new business of Ayodele

UC1 - Platform login

UC10 - Get suggestions

UC14 - Checking the weather forecast

Scenario F4 - Mrigankshekhar, the resilient farmer

UC1 - Platform login

UC2 - Interact with another user

UC3 - Reply to a new message

UC8 - Create a new discussion in the Forum

UC11 - Insert data of production

Scenario TPM1 - Resilient farmers are rewarded

UC1 - Platform login

UC2 - Interact with another user

UC13 - Generation of the ranking of the farmers

Scenario TPM2 - Nobody left behind

UC1 - Platform login

UC2 - Interact with another user

UC12 - Search data of production

UC13 - Generation of the ranking of the farmers

Scenario TPM3 - For a better Telangana

UC1 - Platform login

UC12 - Search data of production

UC19 - Visualize the report of an agronomist's visit

Scenario A1 - A day as an agronomist

UC1 - Platform login

UC12 - Search data of production

UC15 - Checking the schedule

UC16 - Reporting a visit to a farmer

Scenario A2 - A bad day for Chameli

UC1 - Platform login

UC17 - Changing the schedule

Scenario A3 - The work office of Nandakishor

UC1 - Platform login

UC2 - Interact with another user

UC5 - Read about a problem

UC10 - Get suggestions

UC13 - Generation of the ranking of the farmers

UC14 - Checking the weather forecast

UC15 - Checking the schedule

UC18 - Creating a new event in the schedule

3.3.4 Mapping Use Cases on Requirements

Use Case	Requirements
UC1 - Platform login	R48, R49
UC2 - Interact with another user	R7
UC3 - Reply to a new message	R7
UC4 - Report of a problem	R8, R9, R10, R20, R21
UC5 - Read about a problem	R33, R43
UC6 - Ask for help	R8, R9, R10, R23
UC7 - Reply to a request of help	R24, R28, R29
UC8 - Create a new discussion in the Forum	R25
UC9 - Reply to a discussion in the Forum	R26
UC10 - Get suggestions	R8, R9, R10, R14, R15, R16, R17
UC11 - Insert data of production	R2, R3, R4, R5, R8, R9, R10, R19
UC12 - Search data of production	R32
UC13 - Generation of the ranking of the farmers	R6, R31, R41
UC14 - Checking the weather forecast	R12, R13, R30
UC15 - Checking the schedule	R34,R35
UC16 - Reporting a visit to a farmer	R39, R40
UC17 - Changing the schedule	R34, R36, R38
UC18 - Creating a new event in the schedule	R34, R36
UC19 - Visualize the report of an agronomist's visit	R44

Table 20: Mapping Use Cases on Requirements

3.4 Performance Requirements

This section will focus on numerical requirements of the software, which include both computational speed and constraints on the interactions from humans with DREAM. The website shall be fast to load, since all the computation will be done by the server, which means that the users will only retrieve the required data. Since the worktime of both farmers and agronomists is usually from 8AM to 8PM, the load out of this time slot is expected to decrease considerably. DREAM doesn't require any constraints regarding the speed while retrieving or uploading data in order to fulfill its goals, but to provide a more enjoyable experience to the users, at least 95% of the requests shall be processed in less than 2 seconds.

3.5 Design Constraints

The system shall store all data in a standardized form, which will make it easier to store new data and run queries. DREAM system shall use stateless protocols and standard operations to allow components to be managed and updated without affecting the whole system. It's crucial to design modules properly so that ease of use, security and performance will remain the core factors of the system.

3.6 Hardware Limitations

DREAM requires sensors installed in every farm to measure water distribution and the humidity of the soil. Since those components are subjected to degradation, and could stop working properly without any notice, it's expected that some problems will arise. This, however, shouldn't impact on the overall experience, since those malfunctions should occur rarely, and they should be dealt with hastily.

3.7 Software System Attributes

3.7.1 Reliability

In order to avoid disservices, the system has to be fault tolerant. Errors handling and fault containment mechanism have to be arranged.

3.7.2 Availability

The system shall guarantee at least 99% (2-nines) availability, which means at most 3.65 days of down-time per year. This is justified by the fact that, although it's not essential for the system to be always running in order to reach its goals, a high availability should guarantee a better experience for the users. Furthermore, DREAM partners with external IT providers in order to retrieve weather forecasts and combine the collected data. This links the offered service with those external entities, which means that a problem on the server of one of the IT providers could result in a worsened overall experience for DREAM users.

3.7.3 Security

The system shall be managed with Role Based Access Control (RBAC), which grants access rights based on the role of the user. This technology shall be divided into two modules:

- **Authentication:** the system request and verify the identity of every user attempting to login.
- **Authorization:** verify the permission of the logged user before performing an action (e.g., only agronomists can access the daily plan).

3.7.4 Maintainability

The system shall be characterized by scalable and reusable modules, which will be easier to implement and replace in case of failure. Ordinary maintenance and bug fixes shall be scheduled during the night-time, when the traffic is reduced.

3.7.5 Portability

The web app for DREAM should be accessible by any browser. The server instead does not pose any major requirement for portability.

3.7.6 Usability

The web app shall be designed to be clear and easy to use. The graphical interface should help users to access the service required with no assistance required.

3.8 Other Requirements

3.8.1 Privacy Requirement

The system shall ensure that the collection and transmission of personal data is handled in accordance with user's expectation and regulations. Only the necessary data will be requested to the user, and the system shall block unauthorized access to stored information via encryption.

3.8.2 Installation Requirements

For the installation of DREAM, every farmer has to receive a sensor of humidity and a sensor for measuring water distribution. Providing every sensor will require one year at most, while the installation of those sensors is left to the farmer and will take at most 2 working days.

4 Alloy

In this section will be provided a formal model of the problem achieved using Alloy.The model represent only the most important part of the problem, few things have been simplified leaving the more relevant constraints.
For the sake of readability the generated world is split into three sub-worlds which will be explained below.

4.1 Code

```
open util/integer
---------------------------------
-- Enumerations
---------------------------------
sig Seed{}
sig Fertilizer{}
sig Forecast{}
sig District{}
sig Mandala{}
sig TypeOfProduction{}


---------------------------------
-- Typedef
---------------------------------
abstract sig Boolean{}
one sig TRUE extends Boolean{}
one sig FALSE extends Boolean{}

---------------------------------
-- Classes
---------------------------------
abstract sig Users{
        //username: one String
        //password: one String
}

sig Zone{
        mandala: one Mandala,
        district: one District
}

sig Location{
        zone: one Zone
        //address: one String
}

sig WeatherRequest{
        zone: one Zone,
        forecast: one Forecast
        //startDate: one Date
        //endDate: one Date
}

one sig Weather{
        weatherRequest: set WeatherRequest
}

sig TPM extends Users{}

sig Farmer extends Users{
        location: one Location,
        resilient: one Boolean
}

sig Report{
        //body: one String
}

one sig VisualizeInitiatives{
```

```
        reports: set Report
}

sig Event{
        farmer: one Farmer,
        //title: one String
        visited: one Boolean,
        report: lone Report
}

sig DailyPlan{
        visit: set Event,
        //date: one Date
        confirm: one Boolean
}

sig Agenda{
        dailyPlan: set DailyPlan
}

sig Agronomist extends Users{
        zone: one Zone,
        agenda: one Agenda
}

sig SeedType{
        name: one Seed,
        kgSeed: one Int //float
}

sig FertilizerType{
        name: one Fertilizer,
        kgFertilizer: one Int //float
}

sig DataCultivation{
        crop: one SeedType,
        fertilizer: one FertilizerType,
        typeOfProduction: one TypeOfProduction,
        tonsPerAcre: one Int //float
}

sig DataProduction{
        farmer: one Farmer,
        waterUsed: one Int, //float
        soilHumidity: one Int, //float
        cultivations: some DataCultivation
}

one sig ReportProduction{
        dataProductions: set DataProduction
}

sig Problem{
        farmer: one Farmer
        //title: one String
        //problem: one String
}

one sig ReportProblem{
        problems: set Problem
}

sig Replies{
        farmer: lone Farmer,
        agronomist: lone Agronomist
        //reply: one String
}

sig HelpRequest{
        author: one Farmer,
        replies: set Replies
        //description: one String
```

```
}

one sig Help{
        helpRequest: set HelpRequest
}

sig DiscussionReplies{
        farmer: one Farmer
        //text: one String
}

sig Discussion{
        farmer: one Farmer,
        discussionReplies: set DiscussionReplies
        //title: one String
        //body: one String
}

one sig Forum{
        discussions: set Discussion
}

sig Messages{
        sender: one Users,
        receiver: one Users
        //body: one String
}

sig Chat{
        messages: set Messages
}

sig RankingType{
        farmer: one Farmer,
        performance: one Int
}

one sig Ranking{
        ranking: set RankingType
}

sig WikiFarmRequest{
        zone: one Zone,
        typeOfProduction: one TypeOfProduction
}

one sig WikiFarm{
        wikiFarmRequest: set WikiFarmRequest
}


---------------------------------
-- Facts
---------------------------------
-- Famers
fact Location{
        all l: Location | no disj f1, f2 : Farmer | f1.location = f2.location ∧
        f1.location = l
}

fact ReportAllProblems {
        all p: Problem | p in ReportProblem.problems
}

fact AllFarmerInRankingType{
        all f: Farmer | one r: RankingType | f in r.farmer
}

fact Ranking{
        all rt: RankingType | one r: Ranking | rt in r.ranking
}

fact NumberRankingType{
```

```
        (no p: RankingType.performance | p> #Ranking.ranking ∨ p≤0) ∧
        (no disj r1, r2: RankingType | r1.performance = r2.performance)
}

fact DataProduction{
        all d: DataProduction | (one f: Farmer | f in d.farmer) ∧ (d.waterUsed ≥ 0) ∧
        (d.soilHumidity ≥ 0)
}

fact DataCultivation{
        all d: DataCultivation | (one dp: DataProduction | d in dp.cultivations) ∧
        (d.tonsPerAcre ≥ 0)
}

fact ReportProduction{
        all d: DataProduction | one r: ReportProduction | d in r.dataProductions
}

fact SeedTypeFertilizerType{
        (all s: SeedType | (one d: DataCultivation | s in d.crop) ∧ s.kgSeed ≥ 0) ∧
        (all f: FertilizerType | (one d: DataCultivation | f in d.fertilizer) ∧
        f.kgFertilizer ≥ 0)
}

fact MandalaDistrict{
        all m: Mandala | one z: Zone | (m in z.mandala ) ∧ (no z2: Zone | z ≠ z2 ∧ m in z2.mandala)
}

fact Replies{
        all r: Replies | (one h: HelpRequest | r in h.replies) ∧ (#r.farmer ≠ #r.agronomist)
}

fact HelpRequest{
        all hr: HelpRequest | one h: Help | hr in h.helpRequest
}

fact DiscussionReplies{
        all dr: DiscussionReplies | one d: Discussion | dr in d.discussionReplies ∧
        (no d2: Discussion | dr in d2.discussionReplies ∧ d ≠ d2 )
}

fact    Discussion{
        all d: Discussion | (one f: Farmer | f in d.farmer) ∧ (one f: Forum | d in f.discussions)
}

--Agronomist
fact OneAgendaForAgronomist{
        all a: Agenda | one ag: Agronomist | a in ag.agenda
}

fact DailyPlan{
        all d: DailyPlan | one a: Agenda | d in a.dailyPlan
}

fact Event{
        (all e: Event | (one d: DailyPlan | e in d.visit) ∧ (#e.report ≤ #e.visited) ∧
        (#e.report = 0    e.visited = FALSE else e.visited = TRUE))
}

fact Report{
        all r: Report | one e: Event | r in e.report
}

fact NoFarmerRepetitionEvent{
        all d: DailyPlan | no disj e , e2: Event | (e in d.visit ∧ e2 in d.visit ∧
        e.farmer = e2.farmer)
}

fact SameZone{
        all e: Event | (one a:Agronomist, d: DailyPlan | d in a.agenda.dailyPlan ∧
        e in d.visit ∧ e.farmer.location.zone = a.zone)
}
```

```
fact VisualizeInitiatives{
        all r: Report | r in VisualizeInitiatives.reports
}

--Users
fact Messages{
        all m: Messages | (one c: Chat | m in c.messages) ∧ (m.sender ≠ m.receiver)
}

--System
fact Weather{
        all wr: WeatherRequest | one w: Weather | wr in w.weatherRequest
}

fact WeatherRequest{
        all z: Zone | one w: WeatherRequest | z = w.zone
}

fact WikiFarmRequest{
        all wr: WikiFarmRequest | one w: WikiFarm | wr in w.wikiFarmRequest
}


---------------------------------
-- Function
---------------------------------
fun numEvent[f: Farmer] : one Int {
        #(f <: Event.farmer)
}


---------------------------------
-- Predicates
---------------------------------
pred goodFarmer[f:Farmer, i: Int]{
        one rk: RankingType | f in rk.farmer and rk.performance=<i
}
```

4.2 Description of a general situation in Telangana

The model highlights the case where there are 2 agronomists, agronomist0 and agronomist1, working in the same area. Agronomist0 has two daily plans, the first one is confirmed, despite not meeting the original plan, while the second plan is unconfirmed, but the visit has taken place, this models the fact that the working day is not yet over. Agromist1 has a confirmed daily plan because he confirmed his visit to Farmer2 and has finished his working day. Finally, Farmer0 is both resilient and first in Ranking, so he is qualified to receive incentives from Telangana.
Below is presented the code used for generating the model.

```
pred generalModel{
  #Seed = 0
  #Fertilizer = 0
  #TypeOfProduction = 0
  #DailyPlan = 3
  #Event = 4
  #DataProduction = 0
  #Forecast = 1
  #District = 1
  #Mandala ≥ #District
  #Zone = 2
  #Farmer = 3
  #TPM = 0
  #Agronomist = 2
  #HelpRequest = 0
  #Discussion = 0
}
run generalModel for 15
```

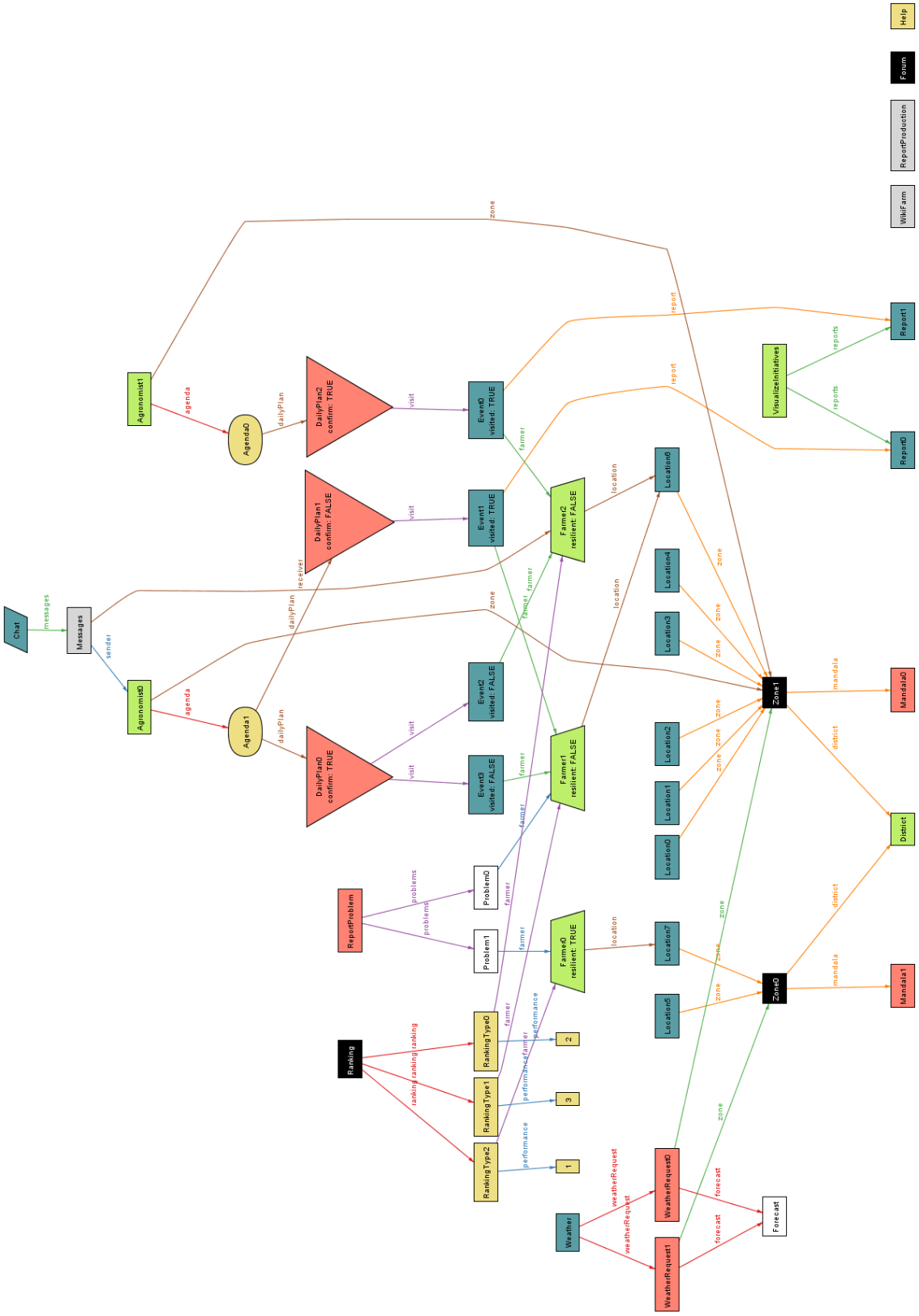

Figure 21: Model of a general situation in Telangana

4.3 Help Requests

This graph represents two farmers requesting help. HelpRequest2 has a response from a farmer, while HelpRequest3 has a response from an agronomist. Both farmers also report a problem they have faced, using the ReportProblem function. In addition, both farmers present their DataProduction, Farmer1 presents two as one describes the most recent production, while the second represents a past situation. Below is presented the code used for generating the model.

```
pred helpRequests{
  #DailyPlan = 0
  #DataProduction = 3
  #Forecast = 1
  #District = 1
  #Mandala ≥ #District
  #Zone = 1
  #Location = 2
  #Farmer = 2
  #TPM = 1
  #Agronomist = 1
  #HelpRequest = 4
  #Discussion = 0
  #WikiFarmRequest = 0
  #Replies.agronomist ≥ 1
  #Replies.farmer ≥ 1
}
run helpRequest for 15
```

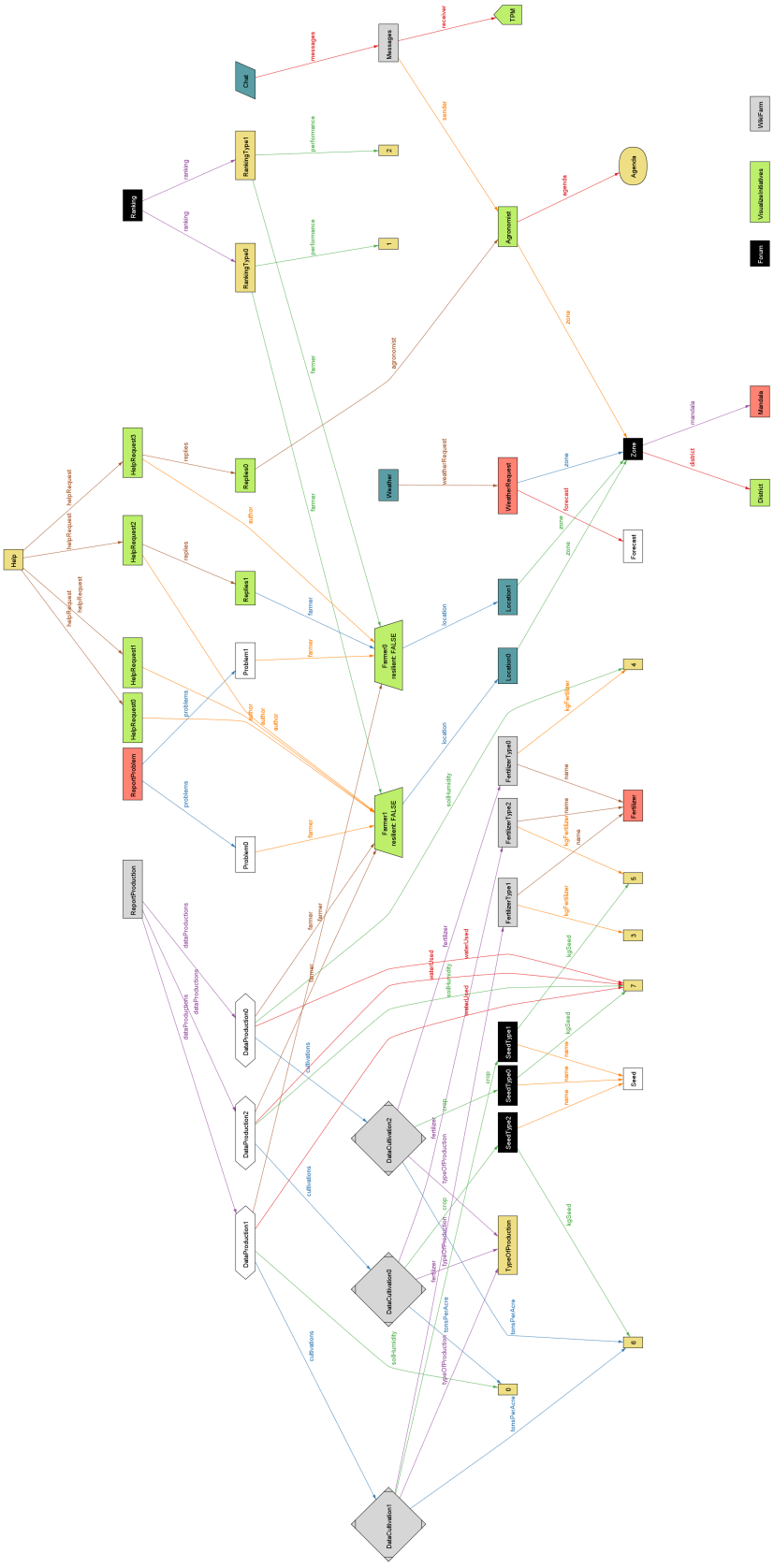

Figure 22: Model of Help requests

4.4 Discussions

This graph shows that a farmer can create a Discussion in the Forum. Any farmer, even the creator, can also reply to the discussions. The most significant Discussion is Discussion5, which has replies from every farmer.
Below is presented the code used for generating the model.

```
pred discussion{
  #Seed = 0
  #Fertilizer = 0
  #TypeOfProduction = 0
  #DailyPlan = 0
  #DataProduction = 0
  #Forecast = 1
  #District = 1
  #Mandala >= #District
  #Zone = 1
  #Farmer = 3
  #TPM = 0
  #Agronomist = 0
  #HelpRequest = 0
  #Discussion = 6
  #DiscussionReplies = 7
  #Problem = 0
  #Messages = 0
  #Location < 5
}
run discussion for 15
```

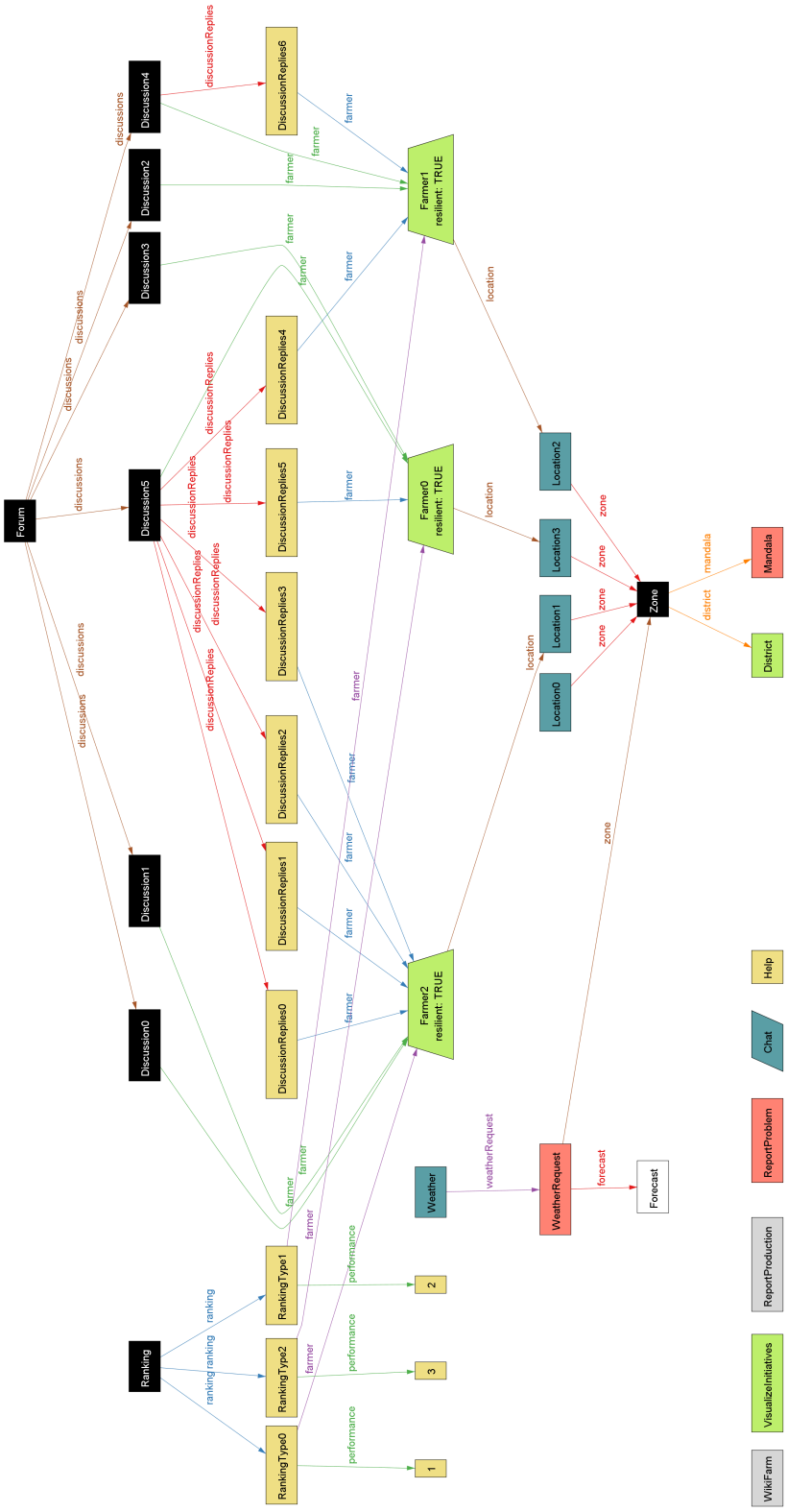

Figure 23: Model of DIscussion

4.5 WikiFarm and Visualize Initiatives

This graph shows the WikiFarm functionality, which allows users to create requests based on their zone and type of production. It also shows the VisualizeInitiatives functionality, which allows TPM to visualize the reports produced by agronomists during their visits.
Below is presented the code used for generating the model.

```
pred wikiFarm{
  #Seed = 0
  #Fertilizer = 0
  #TypeOfProduction = 3
  #DailyPlan = 2
  #DataProduction = 0
  #Forecast = 1
  #District = 1
  #Mandala >= #District
  #Zone = 2
  #Farmer = 3
  #TPM = 0
  #Agronomist = 1
  #HelpRequest = 0
  #Discussion = 0
  #DiscussionReplies = 0
  #Problem = 0
  #Messages = 0
  #Location = 3
  #WikiFarmRequest = 5
  #Report > 2
}
run wikiFarm for 15
```

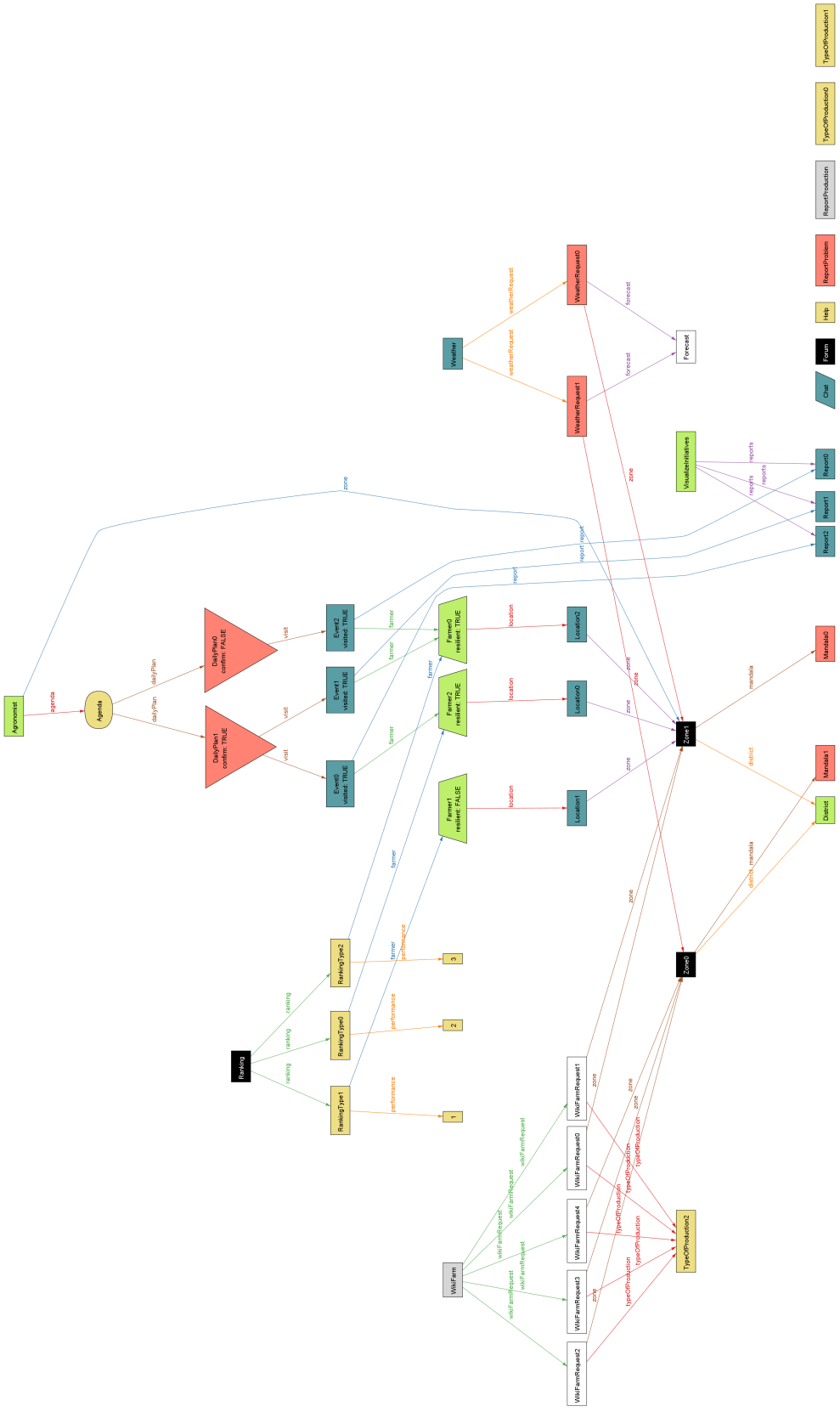

Figure 24: Model of WikiFarm and Visualize Initiatives

5 Effort Spent

In this section is provided an overview of the effort spent to write this document, for more details we invite to read open the file Excel: "Schedule" in our Github page.

5.1 Team Work

Task	Hours
Initial briefing	4,5
Introduction	16,5
UML	2
Document Revision	19

5.2 Stefano Staffolani

Task	Hours
Scenarios	1
Class Diagram	4
Alloy	15

5.3 Stefano Taborelli

Task	Hours
Introduction	4
Use Cases	3
LaTeX	18
Document Revision	5

5.4 Matteo Viafora

Task	Hours
Introduction	1
Scenarios	2,5
Section 3	5
Mapping	3,5
Diagrams	7
Document Revision	1