POLITECNICO
MILANO 1863

Software Engineering 2 Project

Requirement Analysis and Specification Document (RASD)

DREAM

Data-dRiven PrEdictive FArMing in Telangana

Authors:
Marra, Miceli, Mora

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Marra Gabriele, Miceli Matteo, Mora Destiny
Version:	2.0
Date:	09-January-2022
Download page:	GitHub
Copyright:	Copyright © 2022, Marra, Miceli, Mora – All rights reserved

Contents

List of Figures

List of Tables

1 Introduction

1.1 The Problem

The world's food supply chain is threatened by climate change and an unsustainable rate of rising population. Since the Telangana region largely participates in the world's food economy, the Telangana government is interested in pursuing initiatives aimed at mitigating the effect of these issues. More specifically, the Telangana government aims to reform the way they build their policies related to food production. Their goal is to utilize digital public goods and community-centric approaches in order to build resiliency against these dynamic challenges by building policies that are more agile and data-driven.

1.2 Purpose

1.2.1 Purpose of the Document

The purpose of this document is to present a comprehensive description of the requirements and specifications for the project. This document will specify all the information needed to understand exactly the requirements that should be satisfied.

1.2.2 Purpose of the Product

The purpose of the product is to provide a solution to address the problem previously outlined. This initiative, named **D**ata-d**R**iven Pr**E**dictive F**A**r**M**ing, or DREAM, intends to provide a solution that focuses on serving the needs of three stakeholders: farmers, agronomists, and policy makers as shown in **Figure 1**.

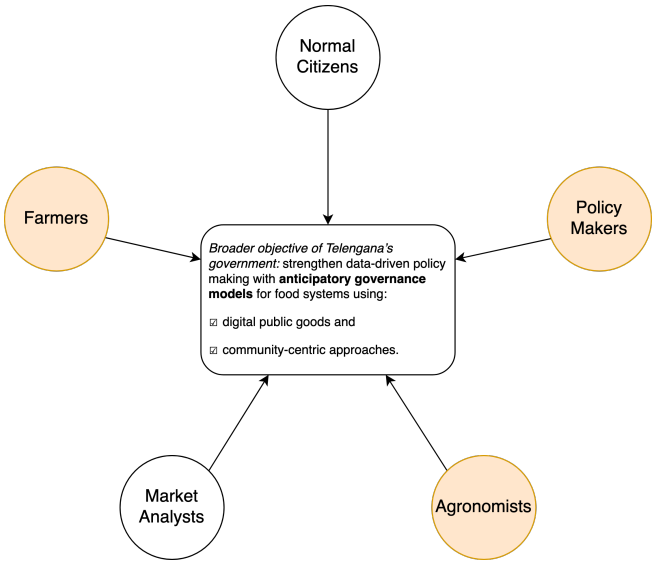

Figure 1: Objective and stakeholders.

The general aim is to take advantage of the present-day abundance of data and the potential for platforms to streamline communication. By utilizing data analysis, farmers can have access to automated solutions, therefore tackling their issues with agility and confidence. Additionally, by offering a means of communication, farmers can leverage their own community as a resource for information and support. Regarding policy makers and agronomists, the product will serve as a tool to aid their respective work-flows.

1.2.3 Goals

The DREAM system is intended to provide a solution for the many issues that arise when the food production sector is strained by slow communication and ineffective policies. Regarding the utility for farmers, purpose of the DREAM system is to: serve as a tool for farmers to input and manage the data relevant to their farms, provide a discussion forum for farmers to draw support from their community, provide a messaging tool to make support from agronomists more accessible, provide farmers custom and automated suggestions from the raw data. For agronomists, the DREAM system serves as a tool to manage their work-flows such as: managing their responsible area, managing their daily plans, providing a messaging tool to easily communicate with farmers directly, and providing a visualization of the data regarding the farmers in their responsible area. Finally, DREAM serves policy makers by providing a tool to visualize the data and quantify the performance of the farmers from the entire Telangana region. **Table 1** provides an aggregate list of the specific goals that are to be accomplished by the DREAM system.

Table 1: Goals of the system.

ID	Goals
G1	Farmers can visualize relevant data and suggestions based on their location and type of production.
G2	Agronomists and farmers can view weather forecast data.
G3	Farmers can interact with others farmers and agronomists by requesting for help and suggestions.
G4	Farmers can create discussion forums with other farmers.
G5	Agronomists can supervise a sub-area inside the region.
G6	Agronomists can visualize the performance of the farmers in their sub-area.
G7	Agronomists can visualize and update a daily plan to visit farms in their area.
G8	Agronomists can specify the deviations from their daily plan and confirm the execution of their daily plan at the end of each day.
G9	Telangana's policy makers can view the performance of the farmers and the ranking of the farmers.
G10	Telangana's policy makers can determine if support from agronomists and well-performing farmers produces significant results.

1.3 Scope

Considering the three users in the scope of this initiative, farmers, agronomists, and policy makers, the design of the system must first consider the following phenomena to define the context in which the system will operate. The world phenomena specifies phenomena or attributes of the environment that merely exist. For example, WP2 in **Table 2** indicates that farmers will have issues on the farm, whether the system exists, or not. The machine phenomena specifies the phenomena that occur inside the system and because of the system. For example, MP5 in **Table 4** indicates that the best path connecting all the farmers in the daily plan will be calculated. This real-world event would not occur if the system did not exist. The shared phenomena, however, specifies the phenomena that satisfy the descriptions for both the world and machine phenomena. For example, SP2 in **Table 3** indicates that a farmer sends a message to the agronomist. This real-world event would still happen if the system did not exist, but due to the functionality offered by the DREAM product, this event occurs within the system.

1.3.1 World Phenomena

Table 2: World Phenomena.

World Phenomena	Description
WP1	An agronomist visits a farm
WP2	Farmer has an issue with the farm

1.3.2 Shared Phenomena

Table 3: Shared Phenomena.

Shared Phenomena	Description	Controlled By
SP1	An agronomist confirms a plan and send all the data about the visits he performed	W
SP2	A farmer sends a message to an agronomist	W
SP3	A farmer creates a forum discussion	W
SP4	An agronomist responds to a farmer help or suggestion request	W
SP5	A user inspects data	W
SP6	Policy maker flags poor performing and well performing farmers	W

1.3.3 Machine Phenomena

Table 4: Machine Phenomena.

Machine Phenomena	Description
MP1	An agronomist visits a farm
MP2	Farmer has an issue with the farm
MP3	Data analysis is performed
MP4	Statistics are created based on data analyzed
MP5	The system computes the best path connecting all farmers an agronomist has to visit
MP6	The system recommend farmers to be visited by an agronomist

1.4 Definitions, Acronyms, Abbreviations

Term	Definition
DREAM	The system described in this document; Data-dRiven PrEdictive FArMing
User	Farmer, agronomist, or policy user; anyone who uses the system.
Policy Maker	Member of the Telangana government who deploys and manages different agriculture-related policies.
Agronomist	Professional who specializes in agriculture sciences.

Term	Definition
Farmer	A user who uses DREAM to help manage data relating to their farms and fields.
Field	One enclosed area that corresponds to one crop. Many fields can make up a farm. The locations of the various fields do not need to be co-located.
Farm	A set of one or many fields that are managed by one farmer.
Production yields	The amount of crop harvested compared to the amount of crop planted. Measured comparatively by percentage or numerically by weight.
Flag	A marker on a farmer that signals the system to increase the priority for the farmer to get visited by an agronomist.
TSDPS	Telangana State Development Planning Society which manages the automated weather stations around the state.
World	A graphical representation of an instance of the Alloy model.
UML	Unified Modeling Language
MTTF	Mean Time To Failure
MTTR	Mean Time To Recovery

1.5 Revision History

Revision	Date	Description
1.0	23 December 2021	Initial Release.
2.0	09 January 2022	Second Release with minor fixes.

1.6 Reference Documents

- Assignment RDD A.Y. 2021-2022
- *Software Abstractions: Logic Language, and Analysis* by Daniel Jackson
- ISO/IEC/IEEE 29148 dated 2018, Systems and software engineering - Life cycle processes - Requirements engineerings

1.7 Document Structure

The document is structured with the following sections:

- **Introduction**: This sections outlines the problem statement, the purpose of the document and of the project, the scope of the domain, and introduces the main goals of the system.
- **Overall Description**: This section goes into more detail about the main functions of the system. This description is aided with diagrams such as a class diagram for the system and flowcharts for select processes. The users of the system are also defined in this section. The system's domain is then described along with any dependencies and constraints.
- **Specific Requirements**: This section describes use cases further with a general use case diagram, use case tables, scenarios, and sequence diagrams. The individual requirements describing the entire system are listed and mapped to the goals and domain assumptions defined in the Introduction and in the Overall Description. Performance requirements and other design constraints are described at the end of the section.

- **Formal Analysis Using Alloy**: The alloy code modeling the system is organized by general function. The facts are co-located with their associated signatures. The predicates and assertions are at the end of the code. Four different worlds have been generated, each one describing a different situation.
- **Effort Spent**: For the purposes of the project assignment, this section itemized the time each participate allotted to different phases of the project.

2 Overall Description

2.1 Product Perspective

2.1.1 Class Diagram

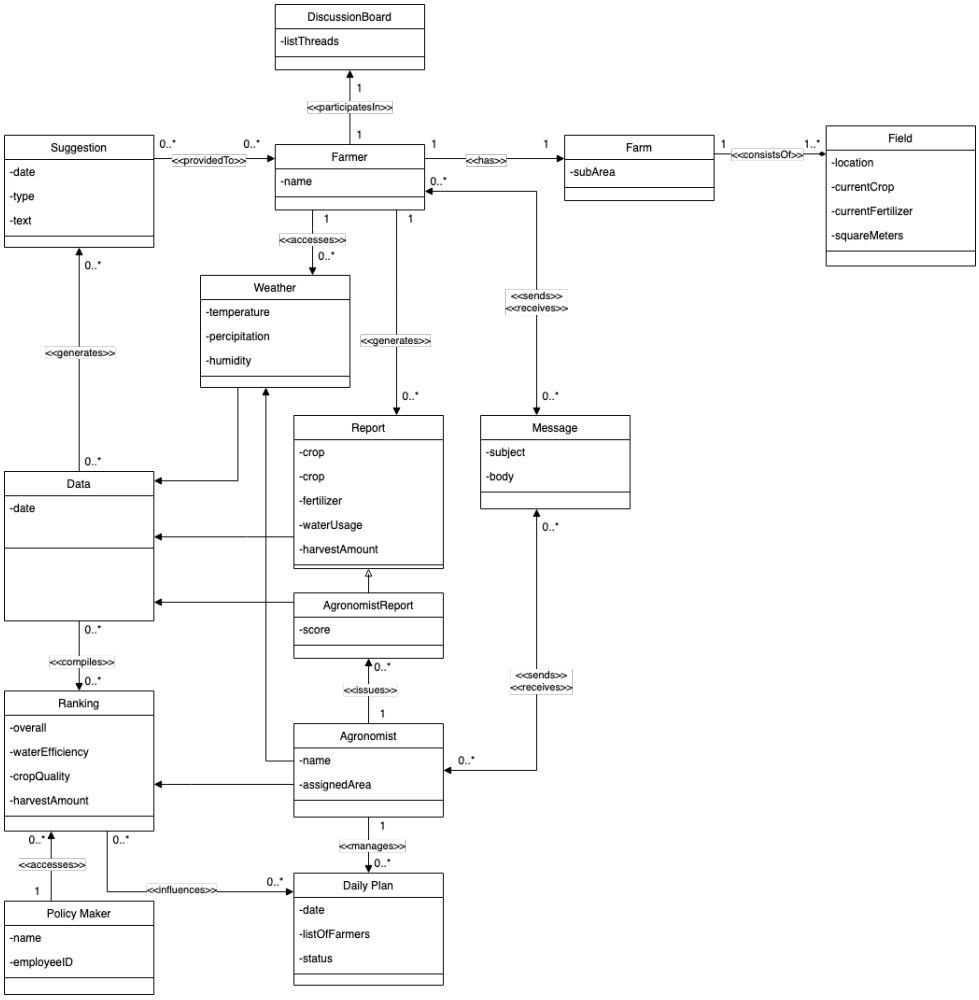

Figure 2: Class diagram.

The UML class diagram shown in **Figure 2** shows a simplified view of the classes needed to support the functions expected from the system. This is not a comprehensive view of all the classes needed but rather a focused view to capture the general decomposition of the different classes and the different relations between them. Considering the goals described in **Section 1.2.3**, the main features that DREAM delivers include a discussion board for farmers to seek support, a messaging interface for farmers and agronomists to communicate, a report submission interface for farmers and agronomists to log data, a daily plan interface for agronomists to manage their visits, a visualization of performance for agronomists and policy makers to access, and a suggestion generator for farmers. The classes and the relations included in **Figure 2** support these main features. The classes are described in detail:

- **Farmer**
  The Farmer class corresponds to a farmer user. The Farmer class should have access to all the functions necessary to support the use cases related to a farmer user.

- **Farm**
  The Farm class corresponds to a farm. As demonstrated by the class diagram, each instance of the Farm class is managed by one Farmer. The Telangana region is split into multiple subareas therefore the Farm class is associated with a subarea instead of GPS location.

- **Field**
  The Field class is distinct from the Farm class to enable a farmer to manage physically disjoint properties. Each farm is made up of one or more fields. Each field has its own GPS location as well as other agriculture-specific attributes.

- **Report**
  The Report class is related to the Farmer class because farmers can submit reports with data describing the status of their fields in order to maintain a historical record of their progress.

- **DiscussionBoard**
  The Farmer class also interacts with the DiscussionBoard class to indicate that the farmer user can interact with the discussion board. For simplicity, other classes that support the implementation of the forum, such as a Thread class or a Post class, are abstracted into DiscussionBoard.

- **Agronomist**
  The Agronomist class corresponds to an agronomist user. The Agronomist class should have access to all the functions necessary to support the use cases related to an agronomist user.

- **AgronomistReport**
  The AgronomistReport class extends from the FarmerReport class that is used by the Farmer class. The difference is that the AgronomistReport also includes a numerical score issued by the agronomist user.

- **DailyPlan**
  The DailyPlan class is related to the Agronomist class because the plans created are specifically for agronomists. This class includes all the functions necessary to support an agronomist user managing their daily plan such as viewing the route, modifying the farmers included in the plan, or entering reports for each farmer visit. Again, the other classes necessary to support the implementation of these functions are abstracted into DailyPlan.

- **Weather**
  The Weather class is related to both the Farmer class and the Agronomist class because both the farmer user and agronomist user need to access weather information. Again, other classes necessary to implement an interface to visualize weather data such as in a map view are abstracted out for simplicity.

  Since the Agronomist class is associated with an assigned subarea, the class diagram omits an Area class and relates the Agronomist class with the Weather class directly. Similarly, since the Farmer is associated with the GPS location through their Farm and Field classes, the Farmer class can relate to the Weather class directly.

- **Message**
  The Message class is related to the Farmer class and the Agronomist class because both the farmer users and the agronomist users can send and receive messages to each other.

Again, other classes that may be necessary to implement the messaging feature are ommited from this diagram for simplicity.

- **Data**
  The Data class aggregates data from the Weather class, the Report class, and the AgronomistReport class. This class performs some data processing and sends it to the Suggestion and Ranking classes, hence the relations in the diagram.
- **Suggestion**
  The Suggestion class is related to the Farmer class because personalized suggestions are auto-generated and issued to farmers based on the data aggregated in the Data class. As noted prior, the Suggestion class is receives the data from the Data class.
- **Ranking**
  The Ranking class is related to the PolicyMaker class because policy makers must access ranking views of the farmers. The Ranking class is also related to the Agronomist class because agronomists must also access ranking views of the farmers, but their view is limited to the farmers in their subarea. These configurable ranking views are based on the data aggregated in the Data class. This ranking influences the priority farmers are given when scheduling the daily plans for agronomists, therefore the Ranking class is related to the DailyPlan class.
- **PolicyMaker**
  The PolicyMaker class corresponds to a policy maker user. The PolicyMaker class should have access to all the functions necessary to support the use cases related to a policy maker user.

2.2 Product Functions

Below is a detailed description of all the main functions of the product. The product functions are generalizations the goals. The functions are further specified in the requirements section and select functions are demonstrated in the scenarios section.

2.2.1 Data Visualization

DREAM helps the farmers to visualize and analyze data about their production: providing simple graphs and statistics to the users, the application allows them to better understand their needs. In the section named “My Farm” each user can find personalized suggestions: thanks to the data provided by the farmers, DREAM can recommend the type of crop to plant or the best fertilizer to use. DREAM also provides the users a precise weather forecast that is updated daily.

2.2.2 Support for Farmers and Discussion Forums

With DREAM, each farmer can easily ask for help or suggestions directly from professional agronomists: inside the “Ask experts” section, the user can send messages to agronomists and obtain vital information to improve their production. Another opportunity is the “Farmer Forum” where each farmer can ask for help from other farmers or read answers to already posted questions.

2.2.3 Performance Visualization and Evaluations

The DREAM system offers agronomists and policy makers a visualization of the performance of the farmers by means of a ranked list view. Agronomists only visualize the performance of

the farmers inside their responsible area whereas policy makers can visualize the performance of all the farmers in the Telangana region involved in the DREAM initiative. By default, this visualization of the ranking organizes farmers by their overall score which combines data such as crop quality, production yields, water usage, and evaluations by the agronomists. Users can configure their ranking view to filter by one of these metrics, by location, or by other attributes of the farmers.

In addition to a visualization of performance, policy makers can also access evaluations of the farmers issued by agronomists that enable the policy makers to determine the efficacy of the support provided to the farmer.

2.2.4 Daily Plan

The DREAM system provides the agronomists a tool for managing their daily plans with functions such as plan creation, modification, and confirmation. During the creation of the daily plan, the system recommends farmers to visit based on the performance of the farmers and the and frequency of visits. The system also displays a map containing all the fields of the farmers. As the agronomist modifies their list of farmers to visit as part of the daily plan, the DREAM system updates the travel path connecting the fields. During the confirmation of the daily plan, the system allows the agronomist modify the data about the visits they performed during the day.

2.3 User Characteristics

Users of the system fall into one of the following categories: farmer, agronomist, or policy maker.

2.3.1 Farmer

A farmer is the type of user that intends to use the DREAM system to visualize data relevant to them such as weather forecasts or personalized suggestions. Personalized suggestions generated by the DREAM product may consist of new crops to consider planting, different fertilizers to use, or other cultivation methods to try. Farmers can also use the DREAM system to ask professional agronomists and other nearby farmers for help and suggestions.

2.3.2 Agronomist

An Agronomist is a type of user that intends to use the DREAM system as a tool to manage some of their job responsibilities. Agronomists want to receive requests for help from the farmers; answer requests from farmers; inspect rankings of the performance of farmers; provide evaluations regarding farmers performance; and utilize a tool to create, modify and confirm daily plans to manage farms visits.

2.3.3 Policy Maker

A Telangana policy maker is a type of user that intends to use the DREAM system to drive policy decisions. Policy makers are mainly interested in accessing rankings and evaluations of all the farmers in the entire area as well as identifying broader trends in the data such as relating the community-provided support to production outcomes. Since policy makers have a more holistic view of the region, they use the DREAM system to configure metrics that are used to classify ”well-performing” and ”poor-performing” farmers based on rankings, evaluations, and data.

2.4 Flowcharts

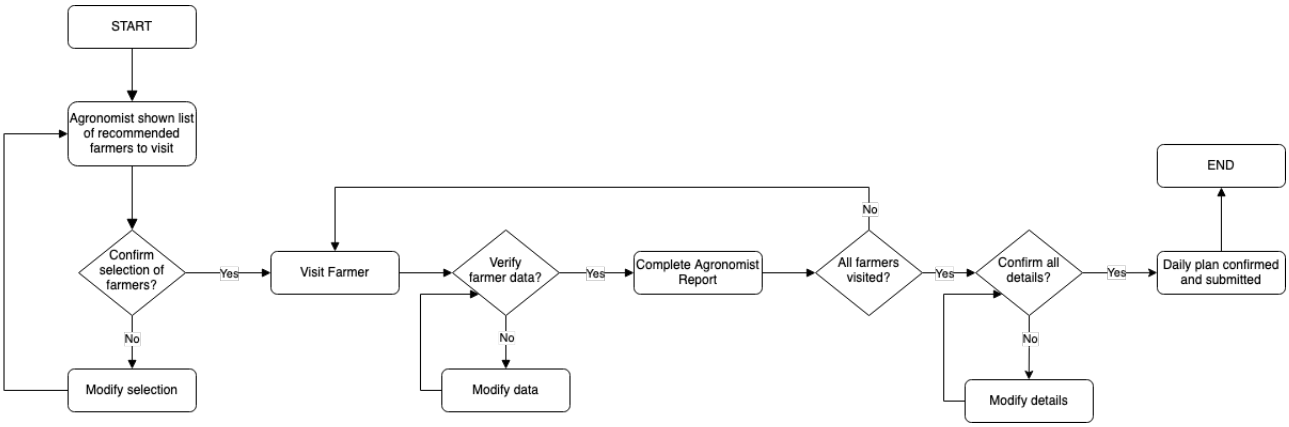

Figure 3: Daily Plans Flowchart.

The UML flowchart in **Figure 3** describes the correlation between the agronomist user and the farmer user when the agronomist launches the interface to start a new daily plan. It shows the process of the agronomist entering their reports as they visit the farmers in their plan. In the beginning of the flow, the agronomist can manually modify the farmers they are scheduled to visit that day. Then, for each farmer the agronomist visits, the agronomist generates a report to capture a snapshot of the farmer's status. After visiting all the farmers scheduled, the agronomist must confirm all the details such as ensuring that all the reports are complete and all the farmers that were in fact visited are included. After confirming, the daily plan is submitted into the system.

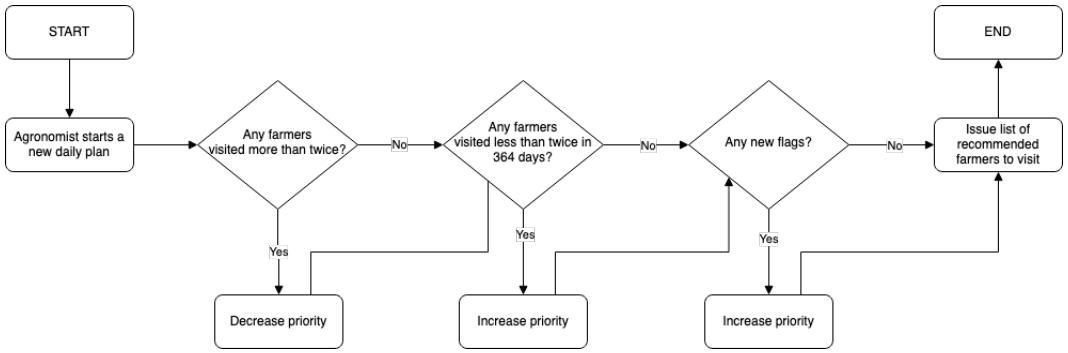

Figure 4: Farmer priority flowchart.

The UML flowchart in **Figure 4** goes into more detail about how the other elements in the system affect the method in which farmers are chosen when the agronomist launches a new daily plan. The general mandate is that farmers must be visited at least twice a year, therefore the first pass deprioritizes farmers who have been visited more than twice in the last year. Then, there is a check to increase the priority for farmers that have not yet met the twice-a-year minimum. Specifically, this pass looks for farmers where it would be the last day to satisfy the mandate. The last pass increases the priority for farmers who have been flagged, either manually by policy makers or automatically by the triggers set in the system. This flowchart does not detail the specific way in which the farmer visiting queue is generated but instead gives a high-level view of the other elements of the system that contribute to the calculation of the priority.

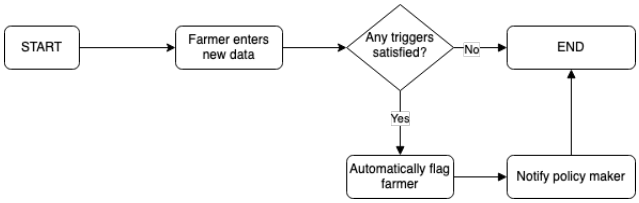

Figure 5: Checking triggers flowchart.

The UML flowchart in **Figure 5** provides a simple view of how the farmer user is indirectly related to the policy maker user. The policy maker is the responsible user for generating triggers. When the farmer user enters new data, the system checks if any of the triggers are satisfied. If so, the system automatically flags the farmer and notifies the policy makers.

2.5 Assumptions, dependencies and constraints

2.5.1 Domain Assumptions

The list of the domain assumptions in **Table 5** are relevant to the context in which the DREAM system operates.

Table 5: Domain Assumptions.

ID	Domain Assumption
D1	Users must have a device connected to internet.
D2	To access to the system the user must have valid credentials.
D3	The data about weather forecast, the farmers and their production, the sensors, the agronomist are correct, complete and stored by the application.
D4	The user has granted permission for GPS, notifications and disk usage.
D5	Farmers have an existing system to quantify, track, and organize their production yields.
D6	Users can successfully operate an interactive application.
D7	Business competition will not influence the farmers' willingness to help.
D8	Farmers are willing to ask for help from other farmers and/or agronomists.
D9	Farmers have industry knowledge about fertilizers, crops, etc.
D10	Farmers are willing to interact with other farmers.
D11	Farmers can recognize issues and production abnormalities.
D12	Agronomists are assigned an area by their superiors.
D13	Agronomists can effectively manage an area assigned to them (ie, the agronomist is not overworked).
D14	Agronomists are experts in their field.
D15	Agronomists will be effective in addressing issues farmers face.
D16	Farmers and Agronomists are not interested in meteorological changes that occur in less than 5 minutes.
D17	Agronomists are effective in determine performance based on various data points.
D18	Modifications to the daily plan are simple.
D19	If a plan is flagged as confirmed, it has actually been performed by the agronomist.
D20	Policy makers want to see the success of farmers in the form of production yields and crop quality.

2.5.2 Dependencies

To generate the travel itinerary when agronomists build their daily plan, the DREAM back-end must utilize the Google Maps API.

The weather data is retrieved from the data collected by the Telangana State Development Planning Society. The TSDPS publishes temperature, rainfall, and humidity data retrieved by over 1000 automated weather stations scattered in the region. The DREAM system will use the weather data from the weather station closest to the GPS location of the field and the currentness of the data is limited to how often the TSDPS publishes the data; once a day.

The personalized suggestions provided to the farmers rely on a database with agricultural data built by agronomists. It is assumed that this database exists and our system can access it.

2.5.3 Constraints

Users must consent to allow the software to use their GPS location otherwise the location data must be entered manually. This includes farmers when they are accessing weather data or entering the location of their individual fields as well as agronomists when providing the start point for the travel itinerary in the daily plan interface.

The system must have effective security measures in place to secure the farmers' sensitive data such as location and credentials. Additionally, since policy makers have special privileges to view farmers' data from the entire region, their credentials must be secured, as well.

Policy makers may need training on how to configure different filters and triggers because they will have access to a large amount of raw data. The filters and triggers can be very powerful in configuring different views of the data and the rankings but training may be required to enable policy makers to take full advantage of the tools.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 App Mockups

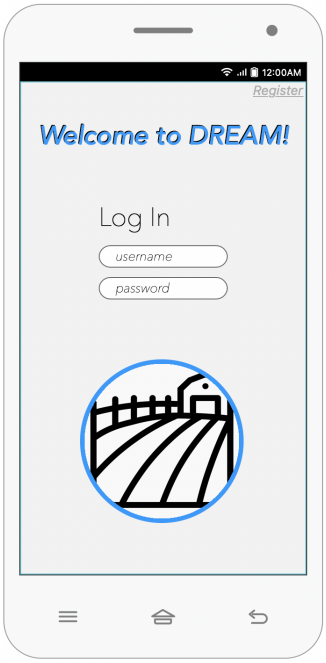

Figure 6: Login Mock Up.

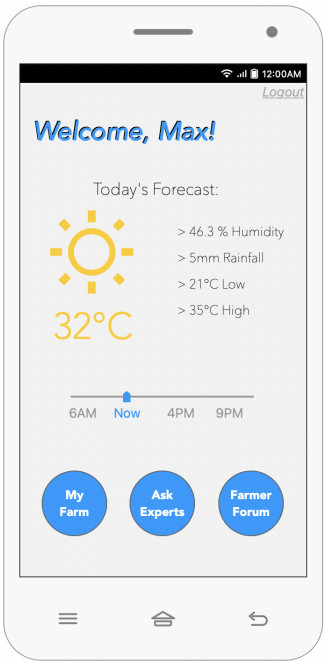

Figure 8: Welcome Farmer Mock Up.

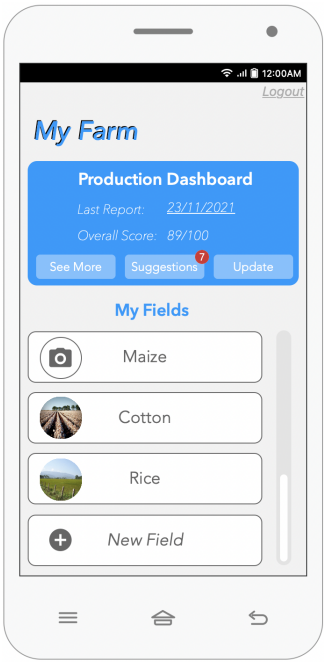

Figure 10: My Farm Mock Up.

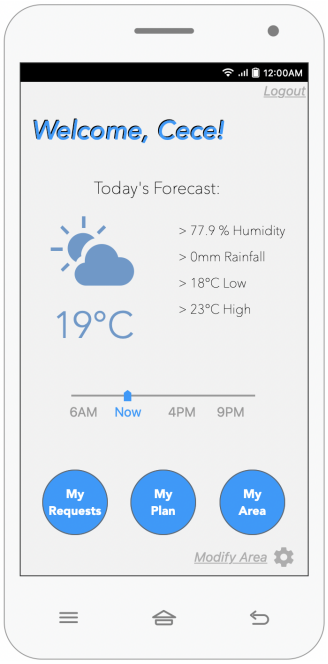

Figure 7: Welcome Agronomist Mock Up.

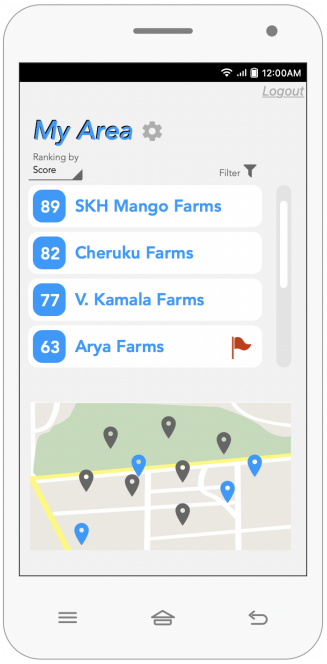

Figure 9: My Area Agronomist Mock Up.

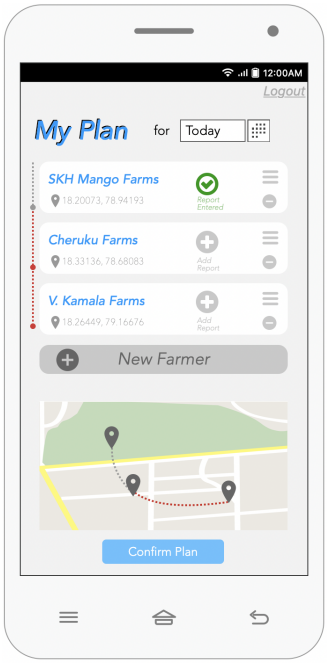

Figure 11: My Plan Agronomist Mock Up.

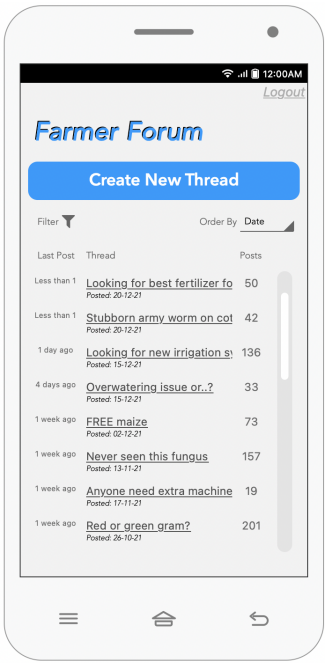

Figure 12: Farmer Forum Mock Up.

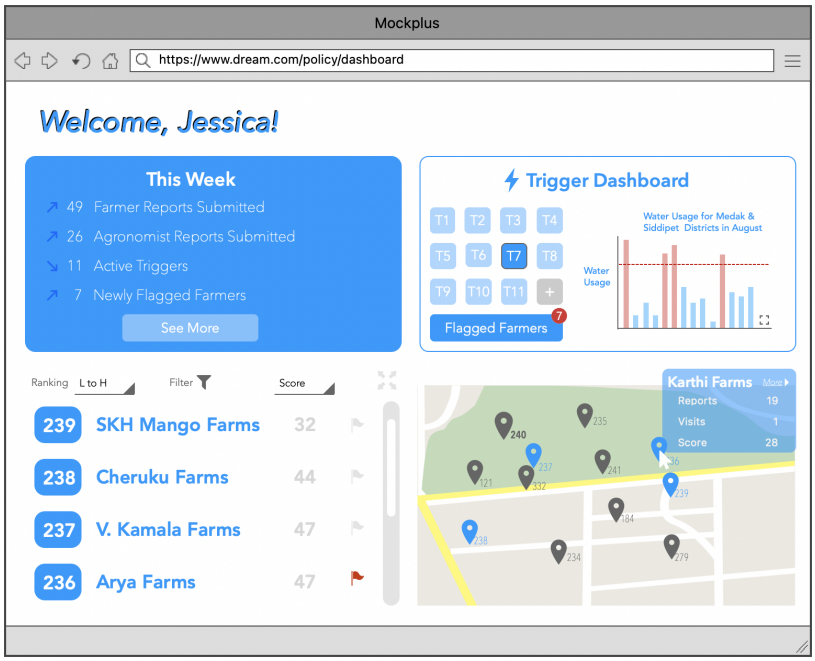

Figure 13: Policy Maker Dashboard Mock Up.

3.1.2 Hardware Interfaces

All users will access the application with their own devices: mobile device, tablet device, or computer device. The farmer's device needs the following in order to have full functionality of DREAM:

- Internet connection [required]: when the farmer is submitting their data while in their fields, their device will need to have an internet connection in order to access DREAM servers. In the event that an area has limited internet connection, the data will be temporarily saved locally on the user's device. The size of the data should be negligible with respect to the memory offered by commonly held devices.
- Camera [optional]: farmers can submit pictures to aid their textual descriptions when posting in the discussion board or when messaging an agronomist. Although this is not required to use the application, it is highly recommended.
- GPS sensor [optional]: farmers can either manually enter their location or allow the application to utilize the GPS sensor from their device.

The agronomist's device needs the following in order to have full functionality of DREAM:

- Internet connection [required]: when the agronomist is visiting farmers their device will need to have internet access in order for the agronomist to submit their reports to the DREAM servers throughout the day and to follow the navigation path calculated by DREAM.
- GPS [optional]: since the agronomist will use a navigation plan generated by DREAM, sharing their GPS location is preferred for an optimal user experience. Without this device functionality, the agronomist will not be able to receive live turn-by-turn navigation guidance.

The policy maker's device needs the following in order to have full functionality of DREAM:

- Internet connection [required]: in order to access the rankings and evaluations of the farmers throughout the region, internet connection is required to access the DREAM servers.

3.1.3 Software Interfaces

This product is accessible through a mobile-friendly application. Farmers will spend the majority of their time accessing the application through a mobile device; agronomists will access the application with both a mobile device and a computer device; policy makers will access the application almost exclusively with a computer device. Due to the mix of mobile and computer devices, the web application must be suitable for both interfaces.

3.1.4 Communication Interfaces

The mobile-friendly web application will communicate with DREAM servers over an internet connection.

3.2 Functional Requirements

The use case diagram in **Figure 14** shows how the different actors expect to use the system. Each use case in the diagram can be further disseminated into other more specific use cases as described in the subsequent **Tables 8 - 23**.

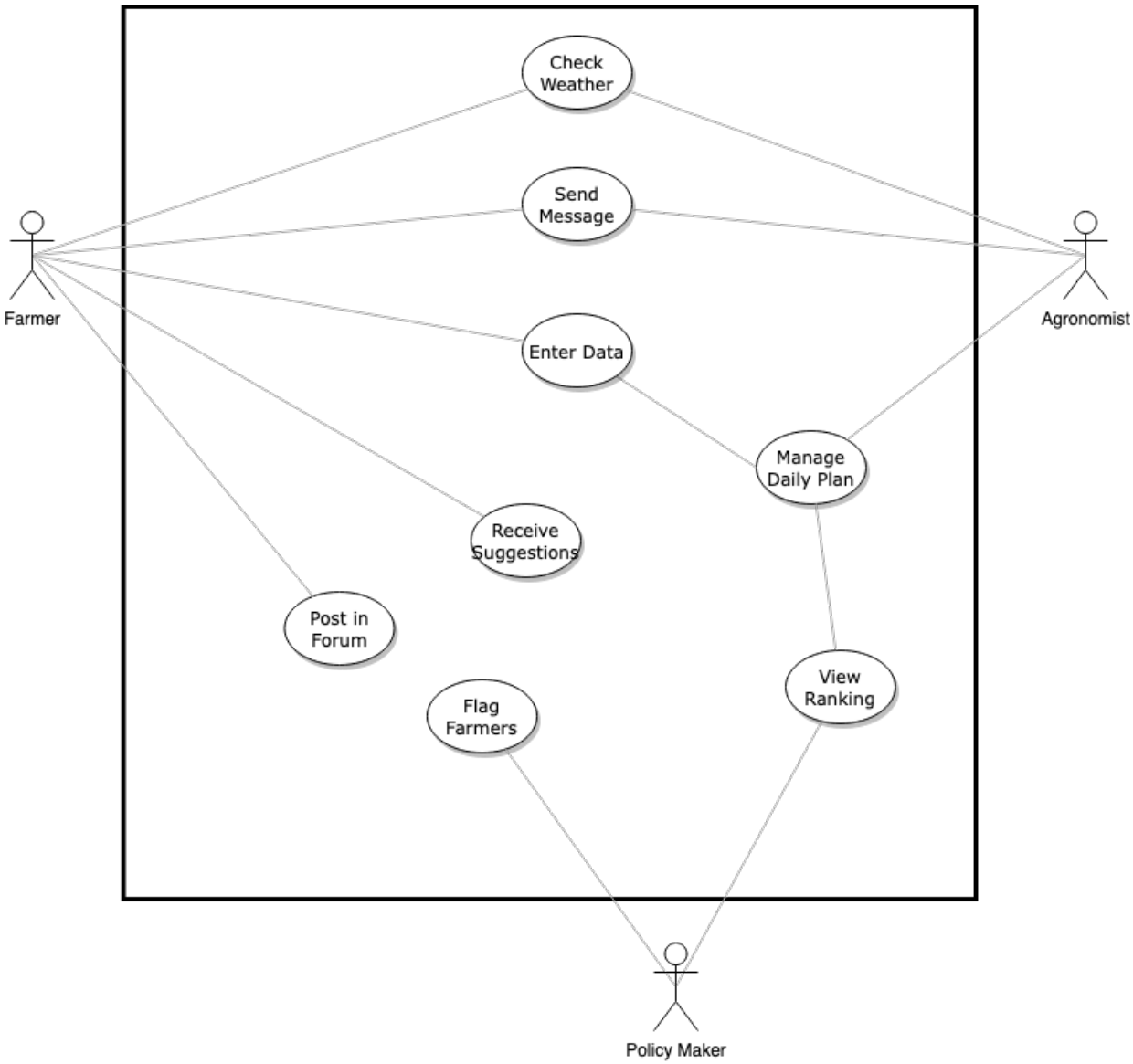

Figure 14: Use case diagram.

Table 6: All users register.

Name	User Registration
Actor	Farmer, Agronomist
Entry Conditions	The user has navigated to the application website.
Events Flow	1. The user register on the site using their email. 2. The user gets an appointment to verify their identity. 3. The user profile is validated and obtains credentials.
Exit Conditions	• The user has credentials of a validated account.

Table 7: All users log in.

Name	User log-in
Actor	Farmer, Agronomist, Policy maker
Entry Conditions	The user opens the application.
Events Flow	1. The user clicks the "log-in" option on the screen. 2. The user inserts their credentials. 3. The user clicks the "log-in" button.
Exit Conditions	The system accepts the credentials.
Exceptions	• The credentials are not valid, therefore the user will be asked to check their input. • The password is not correct so the user will be asked to insert the correct password. • After three tries, the account is temporarily blocked and a reset mail is sent.

3.2.1 Farmer

Scenario 1: Max is a farmer who cultivates some fields near his house in Telangana state. Max has been planting the same plant species over the last few years; he refrains from trying different plants because he is unfamiliar with the cultivation methods. Now, with more children and grandchildren to feed, he would like to change the crop with something more productive. Max learned about the DREAM initiative from a friend and, through the Association of Farmers, he received his credentials to log into the website. Within minutes of accessing the site, he skimmed through the discussion forums and discovered that there are thousands of small farmers like him with the same doubts and fears. He learned which species are more productive and which fertilizer to use. Now he can feed his entire family and even sell some food to the local market.

Scenario 2: Caroline has a big farm in Telangana with 50 hectares of land and different varieties of plants. Last year, during the monsoon season, her fields were flooded, and almost no plants survived. In addition, this summer, the hot temperatures killed some other species. After consulting an expert, she decided to join the DREAM initiative to ask for direct support

from agronomists. Next year she will plant more resilient crops and take some precautions against flooding. Furthermore, frequently uploading information about her production, she can even monitor the overall performance of her fields and try to obtain some incentives from the central government.

Table 8: Farmer Use Case: New Thread, related to Scenario 1.

Name	Create a thread in the discussion forum
Actor	Farmer
Entry Conditions	The user logs into the application with valid credentials.
Events Flow	1. The user opens the “Forums” section. 2. The user clicks on the ”Create Thread” button. 3. The user provides a valid title and message. 4. The user clicks on the ”Publish” button. 5. The user can answer messages published in their thread.
Exit Conditions	• The user exits the ”Forums” section. • The user close the entire application.
Exceptions	• The server is not available.

Table 9: Farmer Use Case: Accessing Forum, related to Scenario 1.

Name	Visit the discussion forum
Actor	Farmer
Entry Conditions	The logs into the application with valid credentials.
Events Flow	1. The user opens the ”Forums” section. 2. The user opens a thread and reads the conversation. 3. The user submits a post in an existing thread.
Exit Conditions	• The user exits the ”Forum” section. • The user closes the entire application.
Exceptions	• The server is not available. • There are no existing threads.

Table 10: Farmer Use Case: Help Request, related to Scenario 2.

Name	Asking for advice from agronomists
Actor	Farmer
Entry Conditions	The user logs into the application with valid credentials.
Events Flow	1. The user opens the "Ask Experts" section. 2. The user selects their assigned agronomist. 3. The user writes a request for help. 4. The user submits the written request. 5. After receiving notification of an answer, the user can see the response from the agronomist. 6. The user can continue the conversation by asking more questions.
Exit Conditions	• The conversation is closed by the farmer. • The conversation is closed by the agronomist.
Exceptions	• The server is not available. • The agronomist does not answer to the message.

Table 11: Farmer Use Case: Submit Report, related to Scenario 2.

Name	Upload information about production
Actor	Farmer
Entry Conditions	The user logs into the application with valid credentials and has new data to upload.
Events Flow	1. The user opens the "My Production" section. 2. The user clicks on the "Update Production Data" button. 3. The user selects a field to update from a list of fields. 4. The user adds the desired information. 5. The user clicks on the "Submit" button.
Exit Conditions	The user submits the information.
Exceptions	• The server is not available. • The provided information is not valid. • Some required information is left empty.

Table 12: Farmer Use Case: New Field.

Name	Add field location with GPS
Actor	Farmer
Entry Conditions	The user logs into the application with valid credentials and is physically in the field.
Events Flow	1. The user opens the "My Farm" section. 2. The user clicks on the "Add New Field" button. 3. The user is prompted to allow the application to use their GPS location. 4. The user clicks on the "Allow" button, allowing the application to locate them using GPS. 5. The user adds information about the size of the field, crop currently planted, the fertilizer in use, and any other information they deem relevant. 6. The user clicks on the "Submit" button.
Exit Conditions	The user submits the information about the new field.
Exceptions	• The server is not available. • The application is not able to use the GPS. • The user doesn't allow the application to use GPS. • The provided information is not valid. • Some required information is left empty.

Table 13: Farmer Use Case: Accessing Weather.

Name	Access weather forecast data
Actor	Farmer
Entry Conditions	The user logs into the application with valid credentials and has at least one field registered with the application.
Events Flow	1. The user opens the "My Farm" section. 2. The user clicks on the "Weather" tab. 3. The user selects a field from a list of fields. 4. The user can see the weather associated with the location for the selected field.
Exit Conditions	• The user closes the weather section. • The user navigates "Back" to the list of fields. • The user closes the entire application.
Exceptions	• The server is not available.

3.2.2 Agronomist

Scenario 3: Schmidt is an agronomist. His job is based on knowing farmers' information and helping them to improve and solve problems. Before installing DREAM he was receiving various emails from farmers. All the emails contained important information about them and help or suggestion requests. He then needed to combine the content from the emails with the information he provided about latest visits of their farm. Sometimes data were not in the same format and not standardized; this slowed his work. Now with DREAM he can easily inspect all of the data in the same place from the convenience of his device. Data are more readable and available. He can also now consult them in any place with an internet connection.

Scenario 4: Cece is an agronomist. Part of her job is to visit farms in the area. She has to establish a daily plan for her visits and then send all the data she collects to the Telangana authorities. The daily plan is very hard to create: she has to search for farmers that need more help than others and find an itinerary that connects all their locations. She has to mind the distance between farms and how to reach them. She installs DREAM and now her work is much easier. All of the information about location, performance, and needs is in one place. The system also recommends farmers to visit and automatically suggests a path connecting all the farms. At the end of a day of visits, she has to compile the forms about farmers' performance that are sent automatically to the Telangana authorities.

Table 14: Agronomist Use Case: Respond to Requests, related to Scenario 3.

Name	Respond to help requests
Actor	Agronomist
Entry Conditions	The user has opened the application, logged in with valid credentials, specified their responsible area, and received at least one help request.
Events Flow	1. The user clicks on the "Help Requests" button. 2. The user views the list of the the different help requests received from different farmers. 3. The user can select a farmer from the list of help request and inspect data about the farmer such as location, ranking, and evaluation. 4. The user reads the help requests and can then reply with solutions and suggestions.
Exit Conditions	• The user has answered to the farmer and the response has been sent to the farmer. • The user navigates out of the "Help Requests" section.
Exceptions	• The messages are not accessible at the moment the user is asked to try again later. • A message cannot be sent in this moment the user is asked to try again later.

Table 15: Agronomist Use Case: Create Daily Plan, related to Scenario 4.

Name	Creating a daily plan
Actor	Agronomist
Entry Conditions	The user has opened the application and logged in with valid credentials.
Events Flow	1. The user clicks on the daily plans interface. 2. The user clicks on the "Create New Plan" button. 3. The user selects a day from a calendar interface. 4. The user can then choose multiple farmers to visit that day from a list of recommended farmers. 5. Whenever a farmer is added to the daily plan the system recalculates the best path that connects all selected farmers and updates the map.
Exit Conditions	The daily plan is created and inserted in the "Your Daily Plans" section.
Exceptions	• The server is not responding. • The Google API is not working.

Table 16: Agronomist Use Case: Confirm Plan, related to Scenario 4.

Name	Confirming a daily plan
Actor	Agronomist
Entry Conditions	The user has opened the application, logged in with valid credentials, specified their responsible area, and created a daily plan for that specific day.
Events Flow	1. The user click on the daily plan interface. 2. The user clicks on the "Your Daily Plans." 3. The user can then choose a specific daily plan and view or modify the information.
Exit Conditions	The plans are correctly modified and saved by the system
Exceptions	• The database with daily plans data is not accessible at the moment the user is asked to try again later.

Table 17: Agronomist Use Case: View Rank.

Name	Viewing the ranking
Actor	Agronomist
Entry Conditions	The user has opened the application, logged in with valid credentials, and specified their responsible area.
Events Flow	1. The user clicks on the farmers ranking interface. 2. The user can view general statistics of the area. 3. The user can click on any farmer to view their evaluation and statistics.
Exit Conditions	The user selects another tab to exit the ranking view.
Exceptions	• The database of farmers performance is not accessible.

Table 18: Agronomist Use Case: Submit Report, related to Scenario 4.

Name	Sending a report
Actor	Agronomist
Entry Conditions	The user has opened the application, logged in with valid credentials, and specified their responsible area.
Events Flow	1. The user clicks on the daily plan interface. 2. The user click on the daily plan of the day and selects a farmer also based on the GPS position. 3. The user can fill a form about the farmer's data, evaluation and
Exit Conditions	The user has correctly sent the information to the server and the data have been saved.
Exceptions	• The server is not accessible at the moment.

Table 19: Agronomist Use Case: Defining Area.

Name	Choosing the location
Actor	Agronomist
Entry Conditions	The user has the application open and has logged in with valid credentials.
Events Flow	1. The user opens the "My Area" section of the application. 2. The user selects the area in which they are responsible for on the map.
Exit Conditions	The system saves the area chosen.
Exceptions	• The site is not accessible at the moment so the user is asked to try again later.

Table 20: Agronomist Use Case: Accessing Weather.

Name	Inspecting weather forecasts
Actor	Agronomist
Entry Conditions	The user has opened the application, logged in with valid credentials, and specified their responsible area.
Events Flow	1. The user clicks on the weather forecast interface. 2. The user clicks on a section of their responsible area. 3. The user inspects the weather forecast for the selected area.
Exit Conditions	The user exits the weather forecast section.
Exceptions	• The data about weather forecast is not accessible.

3.2.3 Policy Maker

Scenario 5: Setting Triggers

Winston is a policy maker within the Telangana government tasked with providing a mid-season status report on the progress of the farmers. Winston configures his view to organize the ranking of the farmers by water usage. Winston notices a group of outliers over-consuming water but producing below-average yields. Winston sets a trigger to automatically flag farmers that surpass the average water usage while yielding below-average production. The flagged farmers have increased priority in the queue for recommended farmers to visit on the agronomist interface.

Scenario 6: DREAM helping drive policy decisions

The Telangana government partners with a start-up that is developing a new agriculture tool aimed at reducing the manual labor involved in cultivating castor. Jessica, a Telangana policy maker, rolls out a beta program steered by agronomists to issue this new product to various farmers across the region and observe how the product affects castor crop quality and production yields. Jessica observes that farmers participating in the beta program are generating much greater production yields compared to farmers not in the beta program. The increased production yields qualifies the program to get rolled out to the rest of the region. Jess uses the percent-increase data to determine how much the Telangana government can subsidize the cost of the tool.

Table 21: Policy Maker Use Case: Flagging Farmers.

Name	Flagging Poor-Performing Farmers
Actor	Policy Maker
Entry Conditions	The user has logged in with valid credentials and configured a ranking view.
Events Flow	1. The user scrolls the low-ranking section of the list. 2. The user clicks on the grey ”flag” icon for each respective farmer they would like to flag.
Exit Conditions	The flag icon turns red.
Exceptions	The database is unavailable.

Table 22: Policy Maker Use Case: Set Trigger.

Name	Set Trigger
Actor	Policy Maker
Entry Conditions	The user is logged in with valid credentials.
Events Flow	1. The user clicks on “Create Trigger” button. 2. The user is navigated to the “Create Trigger” wizard. 3. The user selects the parameters and limits for the trigger. 4. The user clicks on the ”Submit” button.
Exit Conditions	Sufficient information is provided to set the trigger and the trigger is entered into the system.
Exceptions	• Limits are missing for the parameters selected. • The database is unavailable.

Table 23: Policy Maker Use Case: Filtering Ranking.

Name	Configure and view overall ranking of farmers
Actor	Policy Maker
Entry Conditions	The user is logged in with valid credentials.
Events Flow	1. The user clicks on “View Rankings” button. 2. The user is navigated to list-view of all the farmers in order of their overall ranking. 3. The user clicks on the ”Filter” button to configure the ranking (ie, ordering by water usage, production quantity, or production quality). 4. The user views a new ranking of the farmers. 5. The user can scroll up and down the list of farmers.
Exit Conditions	• The user navigates out of the “View Rankings” section. • The user closes the application.
Exceptions	• There are no farmers whose parameters satisfy the filters and settings specified. • The database is unavailable.

3.3 Requirements Table

Table 24: Requirements Table.

ID	Requirement
R1	The system must allow the farmer to set the production types of their fields.
R2	The system must allow the farmer to set the position of their fields manually.
R3	The system must allow the farmer to set the position of their fields through their devices' GPS.
R4	The system must keep track of the data about farmers.
R5	The system must provide an interface to visualize data.
R6	The system must be able to analyze data and show statistics.
R7	The system must enable farmers to modify their production type.
R8	The system must enable farmers to report issues they may face.
R9	The system must allow the farmer to report production data at a frequency chosen by the farmer.
R10	The system must retrieve the weather forecast data from the data that the Telangana government collects.
R11	The system must show updated weather forecast data at most 5 minutes from which the data has been published by the Telangana government.
R12	The system must provide weather data that forecasts at least 3 days ahead.
R13	The system must allow agronomists to access weather forecast data specific to their responsible area.
R14	The system must allow farmers to access weather forecast data based on their GPS location or from the location of their farm on record.
R15	The system must provide an interface for farmers to request help and suggestions from other farmers.
R16	The system must provide an interface for farmers to receive help requests and receive suggestions sent to them from other farmers.
R17	The system must provide an interface for farmers to provide suggestions to other farmers.
R18	The system must provide an interface for farmers to respond to help requests sent to them from other farmers.
R19	The system must provide an interface for farmers to request help and suggestions from other agronomists.
R20	The system must provide an interface for agronomists to receive help requests sent to them from other farmers.
R21	The system must provide an interface for agronomists to respond to help requests sent to them from other farmers.
R22	The system must provide an interface for agronomists to provide suggestions to other farmers.
R23	The system must provide a forum interface.
R24	The system must allow the farmer to create discussion forums.
R25	The system must allow farmers to view all posts in the discussion forum.

ID	Requirement
R26	The system must allow farmers to post replies in the discussion forum.
R27	The system must keep track of all the forum discussion.
R28	The system must allow agronomists to specify their responsible geographic area.
R29	The system must allow agronomists to modify their responsible geographic area.
R30	The system must allow agronomist to view the list of all farmers in their area.
R31	The system must provide an evaluation of farmers such that the evaluation reflects the quality and quantity of their crop production.
R32	The system must enable agronomists to access farmer evaluations from their specific area.
R33	The system updates farmers' evaluation when new data is available.
R34	The system must provide an interface for daily plans.
R35	The system must recommend which farmers should be included in the agronomist's daily plan.
R36	The system must generate recommendations such that farmers are visited by their respective agronomists at least twice a year.
R37	The system must generate recommendations such that farmers with low evaluation are visited more often than twice a year.
R38	The system must allow agronomist to view the list of all farms to visit on a specific day.
R39	The system must allow agronomists to modify which farmers they visit in their plan.
R40	The system must allow agronomists to specify and modify the duration of the visits in their plan.
R41	The system must maintain a record of farmers who have been visited by their respective agronomists.
R42	The system must allow agronomists to modify the daily plan at the end of the day.
R43	The system must allow agronomists to confirm that the daily plan was executed that the end of that day.
R44	The system must not allow anymore modifications to the plan after the plan is confirmed by the agronomist.
R45	The system must only generate a new plan for a new day after the plan from the preceding day was confirmed by the agronomist.
R46	The system must allow Telangana's policy makers to view the list of all farmers.
R47	The system must allow Telangana's policy makers to view the performance and evaluation of the farmers.
R48	The system must allow Telangana's policy makers to view the ranking of the farmers.
R49	The system must allow Telangana's policy makers to view well-performing and poor-performing farmers.
R50	The system must allow Telangana's policy makers to flag the farmers that need to be helped based on their performance.
R51	The system must designate each farmer a measure of support received by agronomists and other well-performing farmers.

ID	Requirement
R52	The system must allow policy makers to view the history of farmers' performance/ evaluation.
R53	The system must allow Telangana policy makers to view this measure of support designated to each farmer.

3.4 Goal mapping

Requirements must ensure satisfaction of the goal given the context of the domain assumption.

[G1] **Farmers can visualize relevant data and suggestions based on their location and type of production.**

- [D1] Users must have a device connected to internet.
- [D2] To access to the system the user must have valid credentials.
- [D3] The data about weather forecast, the farmers and their production, the sensors, the agronomist are correct, complete and stored by the application
- [D4] The user has granted permission for GPS, notifications and disk usage
- [D5] Farmers have an existing system to quantify, track, and organize their production yields
- [D6] Users can successfully operate an interactive application
- [R1] The system must allow the farmer to set the production types of their fields.
- [R2] The system must allow the farmer to set the position of their fields manually.
- [R3] The system must allow the farmer to set the position of their fields through their devices' GPS.
- [R4] The system must keep track of the data about farmers
- [R5] The system must provide an interface to visualize data
- [R6] The system must be able to analyze data and show statistics
- [R7] The system must enable farmers to modify their production type.
- [R8] The system must enable farmers to report issues they may face.
- [R9] The system must allow the farmer to report production data at a frequency chosen by the farmer.
- [R10] The system must retrieve the weather forecast data from the data that the Telangana government collects.

[G2] **Agronomists and farmers can view weather forecast data.**

- [D1] Users must have a device connected to internet.
- [D2] To access to the system the user must have valid credentials.
- [D3] The data about weather forecast, the farmers and their production, the sensors, the agronomist are correct, complete and stored by the application
- [D4] The user has granted permission for GPS, notifications and disk usage
- [D6] Users can successfully operate an interactive application

[D7] Business competition will not influence the farmers' willingness to help

[D8] Farmers are willing to ask for help from other farmers and/or agronomists

[D9] Farmers have industry knowledge about fertilizers, crops, etc

[D10] Farmers are willing to interact with other farmers

[D12] Agronomists are assigned an area by their superiors

[D16] Farmers and Agronomists are not interested in meteorological changes that occur in less than 5 minutes

[R2] The system must allow the farmer to set the position of their fields manually.

[R3] The system must allow the farmer to set the position of their fields through their devices' GPS.

[R4] The system must keep track of the data about farmers

[R5] The system must provide an interface to visualize data

[R10] The system must retrieve the weather forecast data from the data that the Telangana government collects.

[R11] The system must show updated weather forecast data at most 5 minutes from which the data has been published by the Telangana government.

[R12] The system must provide weather data that forecasts at least 3 days ahead

[R13] The system must allow agronomists to access weather forecast data specific to their responsible area

[R14] The system must allow farmers to access weather forecast data based on their GPS location or from the location of their farm on record.

[G3] **Farmers can interact with others farmers and agronomists by requesting for help and suggestions.**

[D1] Users must have a device connected to internet.

[D2] To access to the system the user must have valid credentials.

[D4] The user has granted permission for GPS, notifications and disk usage

[D5] Farmers have an existing system to quantify, track, and organize their production yields

[D6] Users can successfully operate an interactive application

[D7] Business competition will not influence the farmers' willingness to help

[D8] Farmers are willing to ask for help from other farmers and/or agronomists

[D9] Farmers have industry knowledge about fertilizers, crops, etc

[D10] Farmers are willing to interact with other farmers

[D11] Farmers can recognize issues and production abnormalities

[D12] Agronomists are assigned an area by their superiors

[D14] Agronomists are experts in their field

[D17] Agronomists are effective in determine performance based on various data points

[R5] The system must provide an interface to visualize data

[R6] The system must be able to analyze data and show statistics

[R8] The system must enable farmers to report issues they may face.

[R9] The system must allow the farmer to report production data at a frequency chosen by the farmer.

[R10] The system must retrieve the weather forecast data from the data that the Telangana government collects.

[R15] The system must provide an interface for farmers to request help and suggestions from other farmers

[R16] The system must provide an interface for farmers to receive help requests and receive suggestions sent to them from other farmers

[R17] The system must provide an interface for farmers to provide suggestions to other farmers

[R18] The system must provide an interface for farmers to respond to help requests sent to them from other farmers

[R19] The system must provide an interface for farmers to request help and suggestions from other agronomists

[R20] The system must provide an interface for agronomists to recieve help requests sent to them from other farmers

[R21] The system must provide an interface for agronomists to respond to help requests sent to them from other farmers

[R22] The system must provide an interface for agronomists to provide suggestions to other farmers

[G4] **Farmers can create discussion forums with other farmers**

[D1] Users must have a device connected to internet.

[D2] To access to the system the user must have valid credentials.

[D4] The user has granted permission for GPS, notifications and disk usage

[D5] Farmers have an existing system to quantify, track, and organize their production yields

[D6] Users can successfully operate an interactive application

[D7] Business competition will not influence the farmers' willingness to help

[D8] Farmers are willing to ask for help from other farmers and/or agronomists

[D9] Farmers have industry knowledge about fertilizers, crops, etc

[D10] Farmers are willing to interact with other farmers

[D11] Farmers can recognize issues and production abnormalities

[R5] The system must provide an interface to visualize data

[R8] The system must enable farmers to report issues they may face.

[R23] The system must provide a forum interface

[R24] The system must allow the farmer to create discussion forums.

[R25] The system must allow farmers to view all posts in the discussion forum.

[R26] The system must allow farmers to post replies in the discussion forum

[R27] The system must keep track of all the forum discussion

[G5] **Agronomists can supervise a sub-area inside the region**

[D1] Users must have a device connected to internet.

[D2] To access to the system the user must have valid credentials.

[D3] The data about weather forecast, the farmers and their production, the sensors, the agronomist are correct, complete and stored by the application

[D4] The user has granted permission for GPS, notifications and disk usage

[D5] Farmers have an existing system to quantify, track, and organize their production yields

[D6] Users can successfully operate an interactive application

[D12] Agronomists are assigned an area by their superiors

[D13] Agronomists can effectively manage an area assigned to them (ie, the agronomist is not overworked)

[R5] The system must provide an interface to visualize data

[R6] The system must be able to analyze data and show statistics

[R9] The system must allow the farmer to report production data at a frequency chosen by the farmer.

[R10] The system must retrieve the weather forecast data from the data that the Telangana government collects.

[R13] The system must allow agronomists to access weather forecast data specific to their responsible area

[R28] The system must allow agronomists to specify the geographic area in which they are responsible for.

[R29] The system must allow agronomists to modify their responsible geographic area.

[G6] **Agronomists can view the ranking of farmers' performance in their specific area.**

[D1] Users must have a device connected to internet.

[D2] To access to the system the user must have valid credentials.

[D3] The data about weather forecast, the farmers and their production, the sensors, the agronomist are correct, complete and stored by the application

[D4] The user has granted permission for GPS, notifications and disk usage

[D5] Farmers have an existing system to quantify, track, and organize their production yields

[D6] Users can successfully operate an interactive application

[D12] Agronomists are assigned an area by their superiors

[D13] Agronomists can effectively manage an area assigned to them (ie, the agronomist is not overworked)

[D14] Agronomists are experts in their field

[D15] Agronomists will be effective in addressing issues farmers face

[R5] The system must provide an interface to visualize data

[R6] The system must be able to analyze data and show statistics

[R8] The system must enable farmers to report issues they may face.

[R9] The system must allow the farmer to report production data at a frequency chosen by the farmer.

[R10] The system must retrieve the weather forecast data from the data that the Telangana government collects.

[R13] The system must allow agronomists to access weather forecast data specific to their responsible area

[R28] The system must allow agronomists to specify the geographic area in which they are responsible for.

[R29] The system must allow agronomists to modify their responsible geographic area.

[R30] The system must allow agronomist to view the list of all farmers in their area.

[R31] The system must provide an evaluation of farmers such that the exaluation reflects the quality and quantity of their crop production

[R32] The system must enable agronomists to access farmer evaluations from their specific area

[R33] The system updates farmers' evaluation when new data is available (ie, new farmer event entries or after an agronomist visit, etc)

[G7] **Agronomists can visualize and update a daily plan to visit farms in their area**

[D1] Users must have a device connected to internet.

[D2] To access to the system the user must have valid credentials.

[D3] The data about weather forecast, the farmers and their production, the sensors, the agronomist are correct, complete and stored by the application

[D4] The user has granted permission for GPS, notifications and disk usage

[D6] Users can successfully operate an interactive application

[D12] Agronomists are assigned an area by their superiors

[D13] Agronomists can effectively manage an area assigned to them (ie, the agronomist is not overworked)

[R5] The system must provide an interface to visualize data

[R6] The system must be able to analyze data and show statistics

[R9] The system must allow the farmer to report production data at a frequency chosen by the farmer.

[R13] The system must allow agronomists to access weather forecast data specific to their responsible area

[R28] The system must allow agronomists to specify the geographic area in which they are responsible for.

[R29] The system must allow agronomists to modify their responsible geographic area.

[R30] The system must allow agronomist to view the list of all farmers in their area.

[R31] The system must provide an evaluation of farmers such that the exaluation reflects the quality and quantity of their crop production

[R32] The system must enable agronomists to access farmer evaluations from their specific area

[R33] The system updates farmers' evaluation when new data is available (ie, new farmer event entries or after an agronomist visit, etc)

[R33] The system updates farmers' evaluation when new data is available (ie, new farmer event entries or after an agronomist visit, etc)

[R34] The system must provide an interface for daily plans

[R35] The system must recommend which farmers should be included in the agronomist's daily plan

[R36] The system must generate recommendations such that farmers are visited by their respective agronomists at least twice a year

[R37] The system must generate recommendations such that farmers with low evaluation are visited more often than twice a year

[R38] The system must allow agronomist to view the list of all farms to visit on a specific day.

[R39] The system must allow agronomists to modify which farmers they visit in their plan

[R40] The system must allow agronomists to specify and modify the duration of the visits in their plan

[R41] The system must maintain a record of farmers who have been visited by their respective agronomists

[G8] **Agronomists can specify the deviations from their daily plan and confirm the execution of their daily plan at the end of each day.**

[D1] Users must have a device connected to internet.

[D2] To access to the system the user must have valid credentials.

[D4] The user has granted permission for GPS, notifications and disk usage

[D6] Users can successfully operate an interactive application

[D12] Agronomists are assigned an area by their superiors

[D13] Agronomists can effectively manage an area assigned to them (ie, the agronomist is not overworked)

[D14] Agronomists are experts in their field

[D17] Agronomists are effective in determine performance based on various data points

[D18] Modifications to the daily plan are simple

[D19] If a plan is flagged as confirmed, it has actually been performed by the agronomist

[R5] The system must provide an interface to visualize data

[R28] The system must allow agronomists to specify the geographic area in which they are responsible for.

[R29] The system must allow agronomists to modify their responsible geographic area.

[R30] The system must allow agronomist to view the list of all farmers in their area.

[R31] The system must provide an evaluation of farmers such that the exaluation reflects the quality and quantity of their crop production

[R32] The system must enable agronomists to access farmer evaluations from their specific area

[R33] The system updates farmers' evaluation when new data is available (ie, new farmer event entries or after an agronomist visit, etc)

[R35] The system must recommend which farmers should be included in the agronomist's daily plan

[R36] The system must generate recommendations such that farmers are visited by their respective agronomists at least twice a year

[R37] The system must generate recommendations such that farmers with low evaluation are visited more often than twice a year

[R38] The system must allow agronomist to view the list of all farms to visit on a specific day.

[R39] The system must allow agronomists to modify which farmers they visit in their plan

[R40] The system must allow agronomists to specify and modify the duration of the visits in their plan

[R41] The system must maintain a record of farmers who have been visited by their respective agronomists

[R42] The system must allow agronomists to modify the daily plan at the end of the day.

[R43] The system must allow agronomists to confirm that the daily plan was executed that the end of that day

[R44] The system must not allow anymore modifications to the plan after the plan is confirmed by the agronomist

[R45] The system must only generate a new plan for a new day after the plan from the preceding day was confirmed by the agronomist.

[G9] **Telangana's policy makers can view the performance of the farmers and the ranking of the farmers.**

[D1] Users must have a device connected to internet.

[D2] To access to the system the user must have valid credentials.

[D3] The data about weather forecast, the farmers and their production, the sensors, the agronomist are correct, complete and stored by the application

[D4] The user has granted permission for GPS, notifications and disk usage

[D5] Farmers have an existing system to quantify, track, and organize their production yields

[D6] Users can successfully operate an interactive application

[D17] Agronomists are effective in determine performance based on various data points

[R5] The system must provide an interface to visualize data

[R6] The system must be able to analyze data and show statistics

[R9] The system must allow the farmer to report production data at a frequency chosen by the farmer.

[R10] The system must retrieve the weather forecast data from the data that the Telangana government collects.

[R31] The system must provide an evaluation of farmers such that the exaluation reflects the quality and quantity of their crop production

[R33] The system updates farmers' evaluation when new data is available (ie, new farmer event entries or after an agronomist visit, etc)

[R46] The system must allow Telangana's policy makers to view the list of all farmers.

[R47] The system must allow Telangana's policy makers to view the performance and evaluation of the farmers.

[R48] The system must allow Telangana's policy makers to view the ranking of the farmers.

[R49] The system must allow Telangana's policy makers to view well-performing and poor-performing farmers.

[R50] The system must allow Telangana's policy makers to flag the farmers that need to be helped based on their performance.

[R52] The system must allow policy makers to view the history of farmers' performance/ evaluation (score over time)

[G10] **Telangana's policy makers can use the system to determine if support from agronomists and well-performing farmers produces significant results.**

[D1] Users must have a device connected to internet.

[D2] To access to the system the user must have valid credentials.

[D3] The data about weather forecast, the farmers and their production, the sensors, the agronomist are correct, complete and stored by the application

[D4] The user has granted permission for GPS, notifications and disk usage

[D5] Farmers have an existing system to quantify, track, and organize their production yields

[D6] Users can successfully operate an interactive application

[R9] The system must allow the farmer to report production data at a frequency chosen by the farmer.

[D12] Agronomists are assigned an area by their superiors

[D14] Agronomists are experts in their field

[D20] Policy makers want to see the success of farmers in the form of production yields and crop quality.

[R5] The system must provide an interface to visualize data

[R6] The system must be able to analyze data and show statistics

[R9] The system must allow the farmer to report production data at a frequency chosen by the farmer.

[R10] The system must retrieve the weather forecast data from the data that the Telangana government collects.

[R28] The system must allow agronomists to specify the geographic area in which they are responsible for.

[R29] The system must allow agronomists to modify their responsible geographic area.

[R31] The system must provide an evaluation of farmers such that the evaluation reflects the quality and quantity of their crop production

[R33] The system updates farmers' evaluation when new data is available (ie, new farmer event entries or after an agronomist visit, etc)

[R46] The system must allow Telangana's policy makers to view the list of all farmers.

[R47] The system must allow Telangana's policy makers to view the performance and evaluation of the farmers.

[R48] The system must allow Telangana's policy makers to view the ranking of the farmers.

[R49] The system must allow Telangana's policy makers to view well-performing and poor-performing farmers.

[R50] The system must allow Telangana's policy makers to flag the farmers that need to be helped based on their performance.

[R51] The system must designate each farmer a measure of support received by agronomists and other well-performing farmers.

[R52] The system must allow policy makers to view the history of farmers' performance/ evaluation (score over time)

[R53] The system must allow Telangana policy makers to view this measure of support designated to each farmer.

3.5 Sequence Diagrams

3.5.1 Farmer

As demonstrated in the sequence diagram in **Figure 15**, when the farmer user access the "Ask Experts" area in the application, the farmer will send a request for help to their assigned agronomist in the form of a message. The agronomist and the farmer can continue to exchange messages until either user chooses to end the conversation.

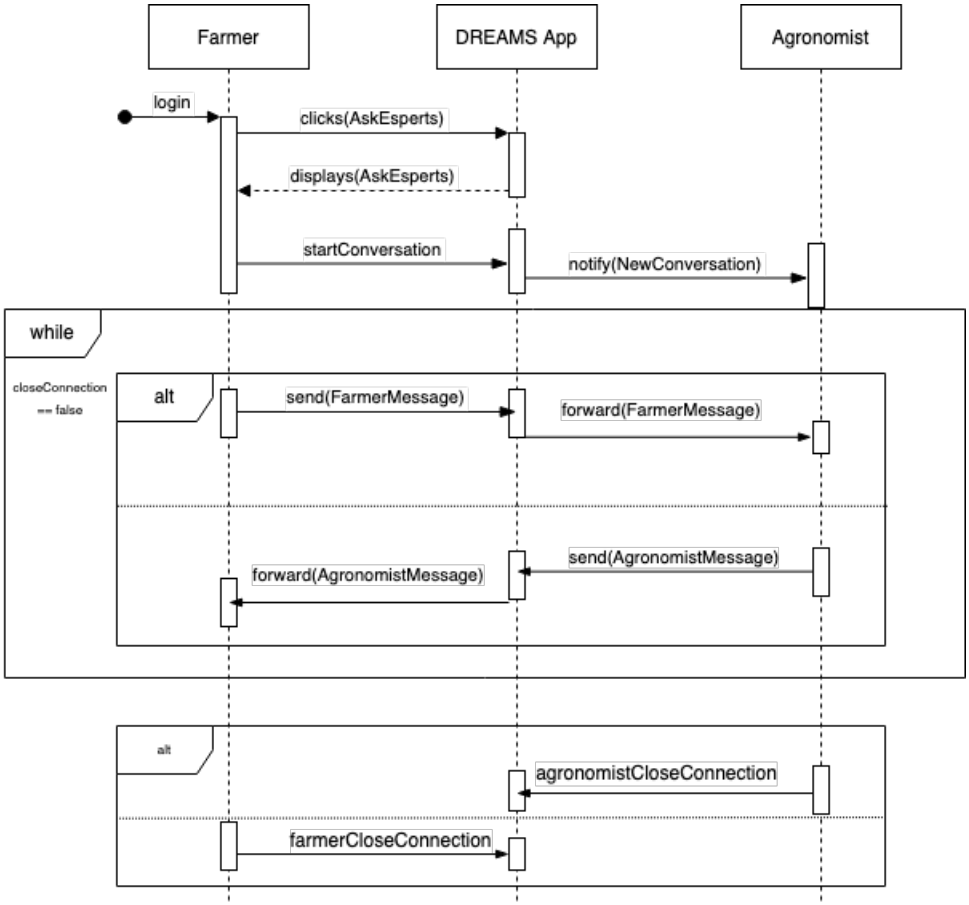

Figure 15: Sequence Diagram for Farmer Requesting Help.

The sequence diagram in **Figure 16** shows the exchange of information between the user and the DREAM application when the farmer user creates a new thread. After the user publishes the new thread, the DREAM application saves the submitted form and updates its servers.

The sequence diagram in **Figure 17** shows how the farmer user interacts with the DREAM application when adding a new field to their farm. Part of adding a new field to their farm involves providing the GPS location (if the user chooses to consent the DREAM application to access their location) and entering some information about the field such as the crop planted and the fertilizer used. After submitting the necessary information, the DREAM application saves and updated its servers and the farmer can then view the new field in their "My Farm" interface.

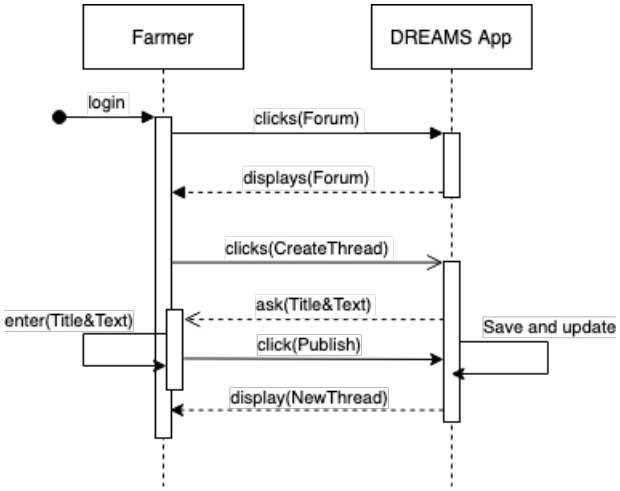

Figure 16: Sequence Diagram for Farmer Creating Thread.

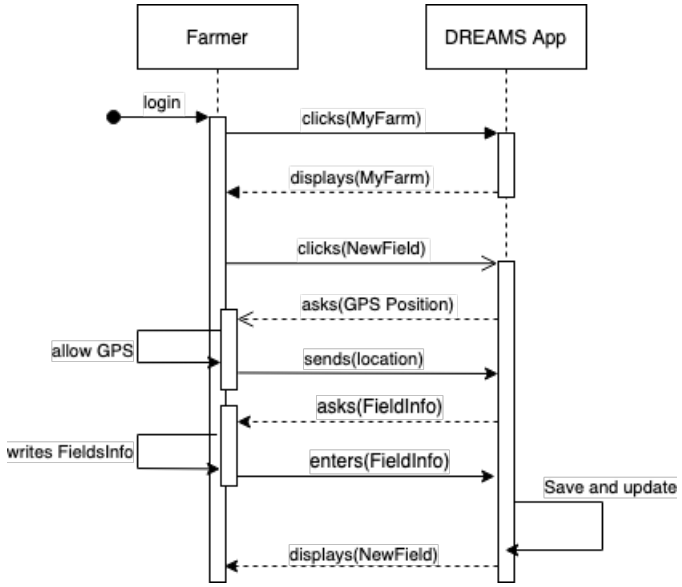

Figure 17: Sequence Diagram for Farmer Adding Field.

3.5.2 Agronomist

The sequence diagram in **Figure 18** shows how the agronomist user interacts with the DREAM servers when modifying the area that they are responsible of with a map display interface.

The sequence diagram in **Figure 19**shows how the agronomist user interacts with the DREAM servers when creating a new daily plan in the daily plan interface including some of the specific calls that occur between the DREAM servers and the user's device. As the agronomist modifies the list of farmers to visit for the day, the agronomist's map is updated to show the new navigation path. From the list of farmers to visit, the agronomist can include or remove each farmer from the plan. Then, at the end, the agronomist can confirm the daily plan and the data is updated on the DREAM servers.

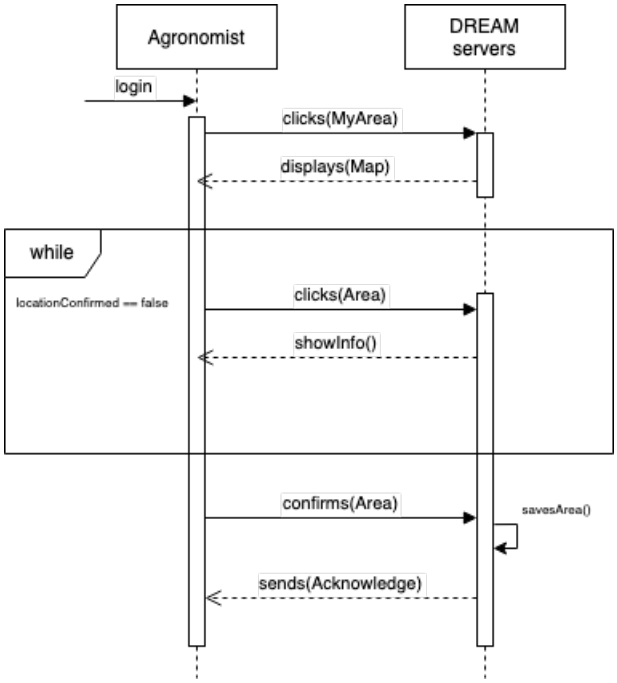

Figure 18: Sequence Diagram for Agronomist Selecting The Area

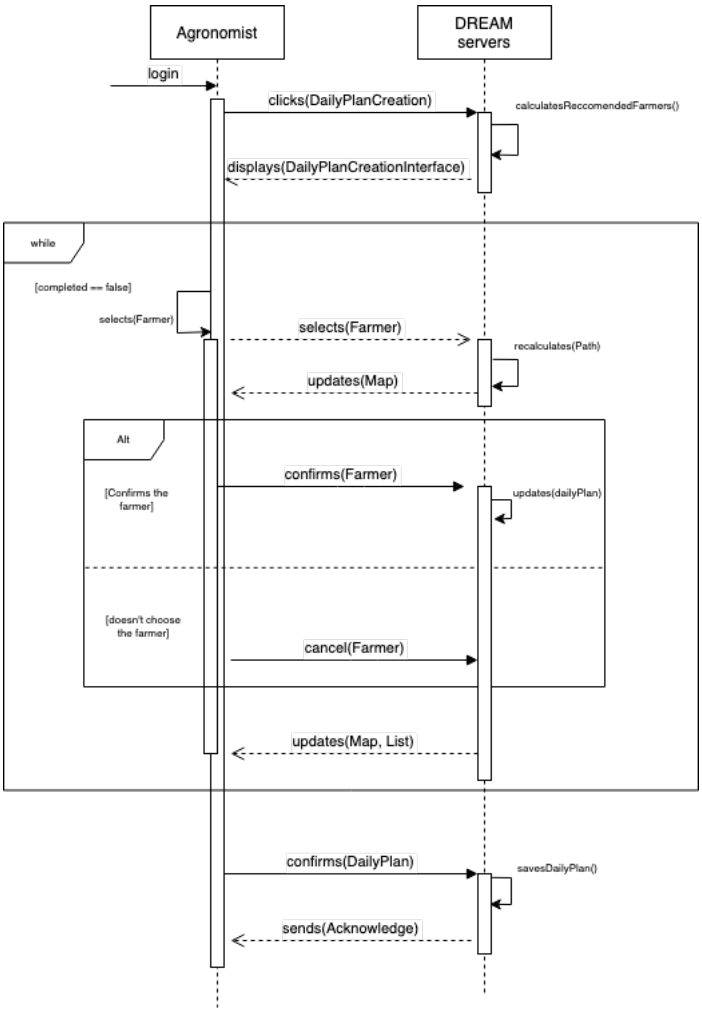

Figure 19: Sequence Diagram for Agronomist Creating Daily Plan

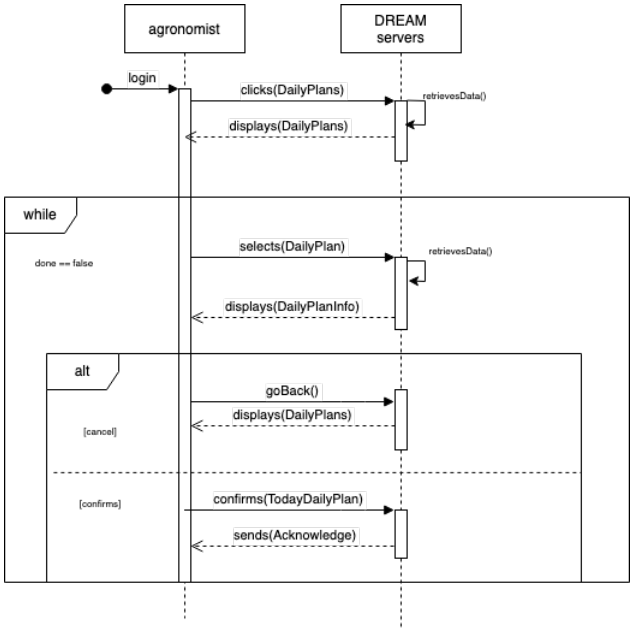

Figure 20: Sequence Diagram for Agronomist Confirming Daily Plan

At the end of a agronomist's day of visiting farmers, the agronomist has to confirm the execution of the plan. **Figure 20** shows how the agronomist user interacts with the DREAM servers when confirming a daily plan. Confirmation of a daily plan should occur after the agronomist has completed all the farmers they intend to visit that day. When confirming, the agronomist can modify the plan so that the plan accurately reflects the visits that the agronomist actually completed that day.

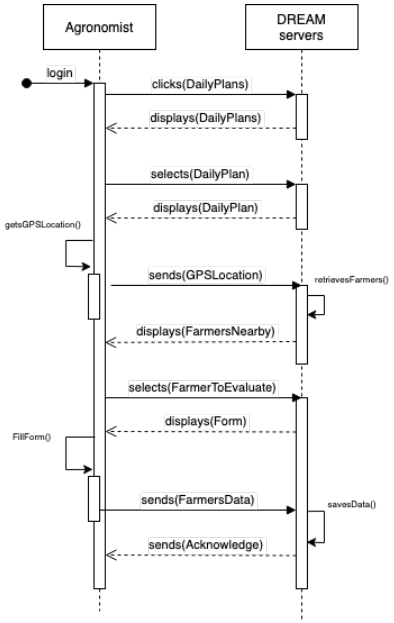

Figure 21: Sequence Diagram for Agronomist Submitting Report

Thee sequence diagram in **Figure 21** shows the interaction between the agronomist user and the DREAM servers when submitting a report evaluating the status of a farmer during a visit.

3.5.3 Policy Maker

The policy maker's use cases revolve around examining the results of the data analysis from a holistic point of view. The sequence diagram in **Figure 22** shows how the policy maker user interacts with the DREAM servers when they set a flag on a farmer. After the policy maker flags a farmer from the ranking list view, the DREAM servers save the flag and send the policy maker user a confirmation that the flag has been set.

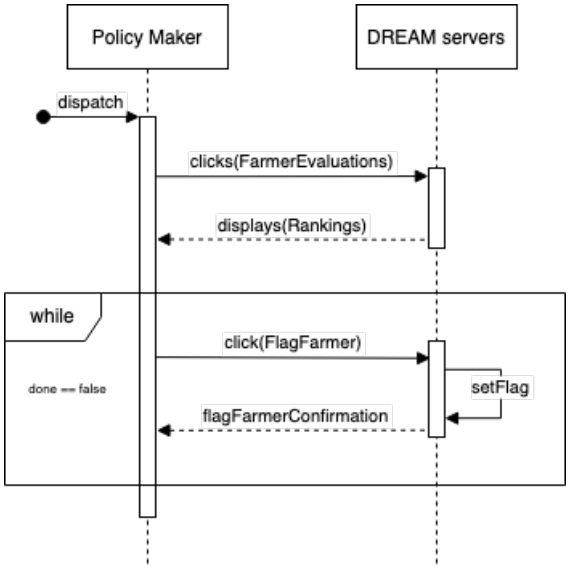

Figure 22: Sequence Diagram for Policy Maker Flagging Farmer

The sequence diagram in **Figure 23** shows how the policy maker user interacts with the DREAM servers when setting a trigger. Once the trigger parameters are set by the policy maker user, the DREAM servers saves them and confirms them with the user. Then, the data that the farmer enters thereafter is checked against the triggers saves on the DREAM servers. If the data entered satisfies a trigger, a notification is sent to the policy maker users.

The sequence diagram in **Figure 24** shows the simple interaction between the policy maker user and the DREAM servers when the policy maker navigates to a ranking view. According to this sequence diagram, the ranking is calculated when the policy maker requests the view. This way, the view considers the most up-to-date data when ranking the farmers. The policy maker can also filter their view such as only considering corn farmers or only considering farmers from specific subareas.

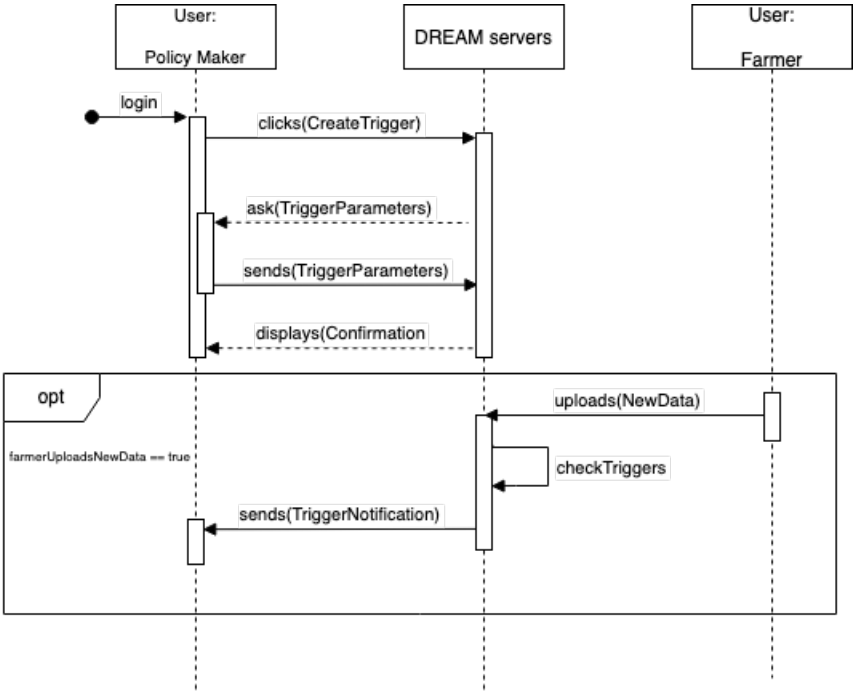

Figure 23: Sequence Diagram for Policy Maker Setting a Trigger

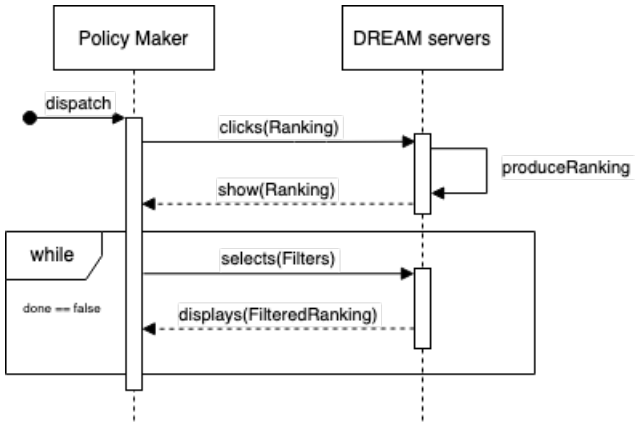

Figure 24: Sequence Diagram for Policy Maker Inspecting the Ranking

3.6 Performance Requirements

The DREAM application is expected to be accessed by users in the Telangana region. The servers should support over 1000 users simultaneously. The large majority of users will be farmers, whereas the number of policy makers and agronomists is strictly the number of government employees that anticipate utilizing DREAM as a tool to manage their job responsibilities. Farmers are expected to interact with the system while managing their fields in the day and agronomists and policy makers are expected to interact with the system strictly during business hours. From around 6:00 to 18:00 DREAM servers should sustain simultaneous requests from multiple users. Due to the large amount of data handled by the system it is crucial to ensure that data loss is minimal in order to preserve the integrity of the results of the data analysis produced by the system. For this reason, data integrity is prioritized over response time.

3.7 Design Constraints

3.7.1 Standards compliance

DREAM application should be as safe and secure as possible: the users and their data must be protected.
The system should be compliant with the ISO/IEC 27001:2013 standard. This standard specifies the requirements for establishing, implementing, maintaining, and continually improving an information security management system within the context of the organization.

3.7.2 Hardware limitations

Since DREAM will be accessed from a mobile-friendly web application, there are few hardware limitations to consider. Instead, the limitations outlined will be easily met with common devices that are probably already used by the users.

Policy makers already utilize a desktop computer for their work so with respect to the DREAM system they would need a screen that is large enough to comfortably view and configure the data visualizations offered by the DREAM application. Since DREAM will be accessed from a web application, the policy maker will need a modern internet browser; the operating system of the computer does not matter.

Agronomists and farmers both may access the DREAM application from a mobile or tablet device. Current iOS/Android devices already come with GPS antennas, internet connectivity, and cameras, therefore no additional limitations need to be considered. Devices without a GPS antenna and without a camera can still be used to access the DREAM system but the user experience will be somewhat compromised: users will have to manually enter location information; users will not have turn-by-turn navigation guidance; and users will not be able to include pictures when describing issues or updates in their reports, forum posts, or help requests.

Regarding the back-end, the system needs a server with constant internet access that can host the web application. A virtual machine on a cloud service, such as AWS or Azure, is a suitable solution to host the web application.

3.7.3 Any other constraints

Since users will have access to data visualizations, the interface should be simple enough as to no overwhelm the user but also ensure that the user has access to all the features, tools, and options when configuring their data views. Additionally, the format offered for farmers to report their data must be defined well enough so that DREAM can effectively parse through it and generate solutions with data analysis but also be flexible enough for farmers to capture qualitative data.

3.7.4 Reliability

Since one of DREAM's primary value-add as a product is the data analysis, preserving the data is very important. In order to recover from potential data loss, offline data backups that are updated weekly are needed for redundancy. Ideally, the DREAM system should have the highest MTTF and the lowest MTTR as possible. However, since DREAM is not dealing with time-critical features, the system can afford a higher MTTR if pending data can be saved on the user's local device while waiting for DREAM system to recover.

3.7.5 Availability

Since users depend on DREAM as a tool to execute their work, it is crucial to ensure the system is characterized by an availability of at least 99.9%. This translates to about 10 minutes of downtime per week. This is tolerable because the system will be used primary during business hours and will have reduced usage or traffic during the weekends; policy makers and agronomists will likely not be accessing the DREAM application during the weekend but farmers might access the DREAM application.

As mentioned, if users try to send data to the DREAM servers when they are unavailable, the data should get saved on the user's device until communication can be reestablished with the DREAM servers. Although this time duration may be perceivable to the end-user, the downtime does not interrupt the user experience since the system will handle the delayed communication. This mitigation technique ensures that the DREAM servers still receive all the data sent to it and the user's experience is not severely interrupted.

3.7.6 Security

Government employees will be accessing the DREAM application with sensitive credentials. Additionally, the data produced by farmer users is associated with sensitive location data. The data around the daily plans for the agronomists contains location-identifying information during an agronomist's work day. Primarily due to these reasons, security should be prioritized in the design of the DREAM system to safeguard from data leaks, exploits, and malicious users.

User credentials should be hashed and encrypted. Transactions, especially those involved with sensitive location data such as the daily plans or field GPS information, should only occur through connections that are secure and encrypted through the most updated version of TLS protocol. This minimizes the risk of data breaches.

3.7.7 Maintainability

During the design, maintainability can only be sustained if good design, documentation, and coding practices are followed. Proper adherence to the standards cited in Section 3.7.1 will also preserve maintainability in the design of the system. Ensuring that the system is designed for maintainability will also enable the system to be easily scaleable; either to more farmers in the region, to other regions in India, or to other nations who wish to use DREAM.

3.7.8 Portability

DREAM's web application should be compatible with a variety of operating systems, both mobile and non-mobile. Since DREAM is serving a market with a wide range of access to devices (from self-funded farmers to government-funded entities), DREAM system and layout should accommodate mobile users with a modern web browser, such as Chrome, Safari, or Firefox. Similarly, for users accessing DREAM from a computer, the OS must support a web browser, such as Chrome, Safari, or Firefox, in order to access the web application.

4 Formal Analysis Using Alloy

4.1 Overview

This section contains a formal model describing the core parts of the system. The model was defined using Alloy, an open-source language and analyzer for software modeling.
The purpose of this section is to validate the consistency of the world generated by the union of the previous assumptions and requirements.

The formal model shows that the system described in the present document is consistent with respect to:

- the utilization of the Farmers' Forum.
- the send of request from a Farmer to an Agronomist.
- the creation and confirmation of an agronomist plan.

4.2 Alloy Model

4.2.1 Alloy Code

```
// Signatures
abstract sig User {
        username : one Username,
        email : one Email
}

sig Username {}{one u:User | u.username=this}

sig Email {}{one u:User | u.email=this}

sig Farmer extends User {
        farm : one Farm,
        farmerReports : set FarmerReport,
        requests : set Request,
        policyMakerFlag: one Boolean,
        suggestions: set Suggestion
}

sig Farm{
        fields : some Field,
        farmer : one Farmer,
        subarea : one Area
}

// Each Farm belongs to only one farmer and each farmer has only one farm
fact {
        all ff:Farm, f:Farmer|f.farm=ff iff ff.farmer=f
}

abstract sig Report{
        timestamp : one Timestamp
}

sig FarmerReport extends Report{
        field : one Field,
        fieldStatus : one FieldStatus,
        waterUsage : lone WaterUsage,
        harvestAmount : lone HarvestAmount
}

//Each FarmerReport belongs to only one Farmer
fact {
        all f1,f2:Farmer, fr:FarmerReport | ((fr in f1.farmerReports) and (fr in f2.
            ↪ farmerReports)) implies f1=f2
        all fr:FarmerReport | one f:Farmer | fr in f.farmerReports
```



```
}

//The Field inside the FarmerReport belongs to Farmer issuing the Report
fact {
        all f:Farmer, fr:FarmerReport | (fr in f.farmerReports) implies (fr.field in f.
            ↪ farm.fields)
}

sig WaterUsage{}

sig HarvestAmount{}

sig FieldStatus {
        crop : lone Crop,
        fertilizer : lone Fertilizer,
}

sig Field{
        farm : one Farm,
        location : one Location,
        sqMeters : one SqMetersArea,
        currentStatus : one FieldStatus
}

// Each Field belongs to only one farm and the relation is bidirectional
fact {
        all fi:Field, f:Farm | (fi in f.fields) iff (fi.farm=f)
}

sig SqMetersArea{}

sig Suggestion {}

//All suggestion belongs to different farmers
fact {
        all f1, f2: Farmer, s:Suggestion | ((s in f1.suggestions) and (s in f2.
            ↪ suggestions)) implies f1=f2
}


sig Agronomist extends User {
        subarea : one Area,
        plans : set Plan,
        requests : set Request
}

sig Location {}{one f:Field | f.location=this}


sig Area {
        farms : some Farm,
        agronomists : some Agronomist
}

//A farm belongs to only one Area
fact {
        all a1,a2:Area, f:Farm | ((f in a1.farms) and (f in a2.farms)) implies a1=a2
        all a:Area, f:Farm | (f in a.farms) iff (f.subarea=a)
}

//An Agronomist belongs to only one Area
fact {
        all a1,a2:Area, ag:Agronomist | ((ag in a1.agronomists) and (ag in a2.agronomists
            ↪ )) implies a1=a2
        all a:Area, ag:Agronomist | (ag in a.agronomists) iff (ag.subarea=a)
}

sig Plan {
        visits : some Visit,
        date : one Date,
        confirmed : one Boolean
}
```



```
// Each Plan belongs to only one Agronomist
fact {
        all a1,a2:Agronomist, p:Plan | ((p in a1.plans) and (p in a2.plans)) implies a1=
            ↪ a2
        all p:Plan | one a:Agronomist | p in a.plans
}

abstract sig Boolean {}

one sig True, False extends Boolean {}

sig Visit {
        farmer : one Farmer,
        time : one Time,
        duration : one VisitDuration,
        agronomistReport : one AgronimistReport
}

fact {
        all a:Agronomist, p:Plan, v:Visit, f:Farmer | ((p in a.plans) and (v in p.visits)
            ↪  and (f=v.farmer)) implies (a.subarea = v.farmer.farm.subarea)
}

// Each Visit belongs to only one plan
fact{
        all p1,p2:Plan, v:Visit | ((v in p1.visits) and (v in p2.visits)) implies p1=p2
        all v:Visit | one p:Plan | v in p.visits
}

sig VisitDuration{}

sig AgronimistReport extends Report{
        score : one Score
}{one v:Visit|v.agronomistReport = this}

sig Score {}

sig Request{
        messages : some Message,
        farmer : one Farmer,
        agronomist : one Agronomist
}

//One Request belongs to only one Farmer and one Agronomist. Check also if the relation
    ↪ is bidirectional
fact{
        all a1,a2:Agronomist, r:Request| ((r in a1.requests) and (r in a2.requests))
            ↪ implies a1=a2
        all f1,f2:Farmer, r:Request| ((r in f1.requests) and (r in f2.requests)) implies
            ↪ f1=f2
        all r:Request | one a:Agronomist | (r in a.requests) and (r.agronomist=a)
        all r:Request | one f:Farmer | (r in f.requests) and (r.farmer=f)
}

//The Agronomist inside the request must be the one of the subarea of the Farm
fact {
        all r:Request | r.agronomist.subarea = r.farmer.farm.subarea
}


//At least one message send by a farmer
fact {
        all r:Request | some m: Message | (m in r.messages) and (m.sender=r.farmer)
}

sig Message {
        request : one Request,
        messageContent : one MessageContent,
        sender : one User,
        receiver : one User,
        timestamp : one Timestamp
```



```
}{one r:Request | (request = r) and ( this in r.messages)}

//One Message belongs to only one Request and receiver and sender must be the two Users
    ↪ owning the Request
fact {
        all m:Message | (m.sender=m.request.farmer and m.receiver=m.request.agronomist)
            ↪ or (m.sender=m.request.agronomist and m.receiver=m.request.farmer)
        all r1,r2:Request, m:Message | ((m in r1.messages) and (m in r2.messages))
            ↪ implies r1=r2
}

sig MessageContent{}{one m:Message | m.messageContent=this}

sig Crop {
        name : one CropName,
        suggestedFertilizers : set Fertilizer
}

sig CropName{}{one c:Crop | c.name=this}
sig Date {}
sig Time {}
sig Timestamp{
        date : one Date,
        time : one Time
}

sig Fertilizer {
        name : one FertilizerName,
        suggestedCrops : set Crop
}

//If a fertilizer is suggested for a specific crop, then that crop should be also listed
    ↪ in the suggested crop field
fact {
        all f:Fertilizer, c:Crop| (f in c.suggestedFertilizers) iff (c in f.
            ↪ suggestedCrops)
}

sig FertilizerName{}{one f:Fertilizer | f.name=this}

sig PolicyMaker extends User {}

//Only one Forum possible - Singleton
one sig Forum {
        threads: set Thread
}

sig Thread {
        title : one ThreadTitle,
        posts : some Post,
        creator : one Farmer,
        timestamp : one Timestamp
}

//Thread belongs to the FORUM
fact {
        all t:Thread, f:Forum | t in f.threads
}

sig ThreadTitle {}

// Each ThreadTitle has one thread
fact {
        all tt:ThreadTitle | one t:Thread | tt = t.title
}

sig Post {
        thread : one Thread,
        postContent : one PostContent,
        creator : one Farmer,
        timestamp : one Timestamp
}
```



```
// Check bidirectional relation between Post and Thread
fact {
        all p:Post | one t:Thread | (p.thread = t) and (p in t.posts)
        all t1, t2: Thread, p:Post | ((p in t1.posts) and (p in t2.posts)) implies t1=t2
}

// At leas a post by the Thread creator
fact {
        all t:Thread | some p:Post | t.creator = p.creator
}

sig PostContent{}

// Each PostContent has one post
fact {
        all pc:PostContent | one p:Post | pc = p.postContent
}

abstract sig Ranking {
        farmers: some Farmer,
        rankingType: one RankingType
}

sig RankingType{}

sig PolicyRanking extends Ranking{}

//All farmers listed in the Ranking and Rankings
fact {
        all pr:PolicyRanking, f:Farmer | f in pr.farmers
        }

//Two different PolicyRankings with same farmers have different Type
fact {
        all pr1, pr2:PolicyRanking | ((pr1.farmers=pr2.farmers) and (pr1.rankingType=pr2.
            ↪ rankingType)) implies pr1=pr2
}

sig AgronomistRanking extends Ranking{
        allowedAgronomists: set Agronomist,
        area: one Area
}

//Agronomists can see the rankings of only their Area
//Contains only the Farmer in that Area
fact {
        all ar:AgronomistRanking, f:Farmer | (f in ar.farmers) iff (f.farm.subarea=ar.
            ↪ area)
        all ar:AgronomistRanking, a:Agronomist | (a in ar.allowedAgronomists) iff (a.
            ↪ subarea=ar.area)
}

//Two different PolicyRankings with same Area have different Type (check only Area, not
    ↪ farmers and allowedAgronomists)
fact {
        all ar1, ar2:AgronomistRanking | ((ar1.area=ar2.area) and (ar1.rankingType=ar2.
            ↪ rankingType)) implies ar1=ar2
}

pred flagFarmer (f:Farmer){
        f.policyMakerFlag = True
}
//run flagFarmer for 5 but exactly 2 Farmer

//Create a new Post inside a Thread
pred createPost (newPost:Post, t:Thread, pc:PostContent, f:Farmer, ts:Timestamp){
        newPost.thread = t
        newPost.postContent = pc
        newPost.creator = f
        newPost.timestamp = ts
```



```
        t.posts = t.posts + newPost
}

//Create a new Thread inside the Farmers' Forum
pred createThread (newThread:Thread, forum:Forum, f:Farmer, tt:ThreadTitle, firstPost:
    ↪ Post, pc:PostContent, ts: Timestamp){
        newThread.title = tt
        newThread.creator = f
        newThread.timestamp = ts

        createPost[firstPost, newThread, pc, f, ts]

        forum.threads = forum.threads + newThread
}

//Send a Message inside a Request
pred sendMessage (newMessage:Message, req:Request, mc:MessageContent, sen: User, rec:User
    ↪ , ts:Timestamp){
        newMessage.request = req
        newMessage.messageContent = mc
        newMessage.sender = sen
        newMessage.receiver = rec
        newMessage.timestamp = ts

        req.messages = req.messages + newMessage
}

//A Farmer create a new Request sending the first Message
pred sendRequestToAgronomist (newRequest: Request, firstMessage:Message, far:Farmer, agr:
    ↪ Agronomist, mc:MessageContent, ts:Timestamp){
        newRequest.farmer = far
        newRequest.agronomist = agr

        sendMessage[firstMessage, newRequest, mc, far, agr, ts]

        far.requests = far.requests + newRequest
        agr.requests = agr.requests + newRequest
}

// Create an AgronomistReport inside a Visit
pred createAgronomistReport (newReport: AgronimistReport, vis:Visit, ts:Timestamp, sc:
    ↪ Score){
        newReport.timestamp = ts
        newReport.score = sc

        vis.agronomistReport = newReport
}

// Create an agronomist Visit inside a Plan
pred createVisit (newVisit: Visit, p:Plan, f:Farmer, t:Time, dur: VisitDuration, ar:
    ↪ AgronimistReport){
        newVisit.farmer = f
        newVisit.time = t
        newVisit.duration = dur
        newVisit.agronomistReport = ar

        p.visits = p.visits + newVisit
}

// Confirm an existing Plan
pred confirmPlan (p:Plan){
        p.confirmed = True
}

// All the farmers have at least an associated agronomist
assert allFarmersHaveAtLeastAnAgronomist {
        all f:Farmer | some a:Agronomist | f.farm.subarea = a.subarea
}

// All Request sent are properly filled
assert requestSent {
        all r: Request, m:Message, far:Farmer, agr:Agronomist, mc:MessageContent, ts:
```



```
            ↪ Timestamp | sendRequestToAgronomist[r,m,far,agr,mc,ts] implies (r in far.
            ↪ requests and r in agr.requests and r.farmer=far and r.agronomist=agr and m
            ↪  in r.messages and m.messageContent=mc and m.timestamp=ts)
}

// All Visit created are properly filled
assert visitCreated {
        all v: Visit, p:Plan, f:Farmer, t:Time, dur: VisitDuration, ar:AgronimistReport |
            ↪  createVisit[v,p,f,t,dur,ar] implies (v in p.visits and v.farmer=f and v.
            ↪ time=t and v.duration=dur and v.agronomistReport=ar)
}

pred show{}

//RUN
run show for 12
run createThread for 6
run sendMessage for 6
run sendRequestToAgronomist for 6
run createAgronomistReport for 4
run createVisit for 4
run confirmPlan for 3
check allFarmersHaveAtLeastAnAgronomist for 10
check requestSent for 10
check visitCreated for 10
```

4.2.2 Predicates execution and assertions checks

Executing "Run show for 12"
Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20 Mode=batch
248362 vars. 9144 primary vars. 472692 clauses. 1760ms.
Instance found. Predicate is consistent. 788ms.

Executing "Run createThread for 6"
Solver=sat4j Bitwidth=4 MaxSeq=6 SkolemDepth=1 Symmetry=20 Mode=batch
42642 vars. 2449 primary vars. 80697 clauses. 297ms.
Instance found. Predicate is consistent. 168ms.

Executing "Run sendMessage for 6"
Solver=sat4j Bitwidth=4 MaxSeq=6 SkolemDepth=1 Symmetry=20 Mode=batch
42385 vars. 2448 primary vars. 80086 clauses. 248ms.
Instance found. Predicate is consistent. 179ms.

Executing "Run sendRequestToAgronomist for 6"
Solver=sat4j Bitwidth=4 MaxSeq=6 SkolemDepth=1 Symmetry=20 Mode=batch
42613 vars. 2448 primary vars. 80794 clauses. 354ms.
Instance found. Predicate is consistent. 186ms.

Executing "Run createAgronomistReport for 4"
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20 Mode=batch
17356 vars. 1144 primary vars. 31535 clauses. 128ms.
Instance found. Predicate is consistent. 100ms.

Figure 25: Execution of 7 predicates and check of 3 assertions.

4.3 Worlds generated

To better represent the model described above, we decided to show all the elements only in the first World. With World 2 and following, we show different sections of the model more significant to our purpose.

4.3.1 World 1

In World 1 we include one element for each signature. It is possible to see the complexity described by the Alloy and how it is consistent.

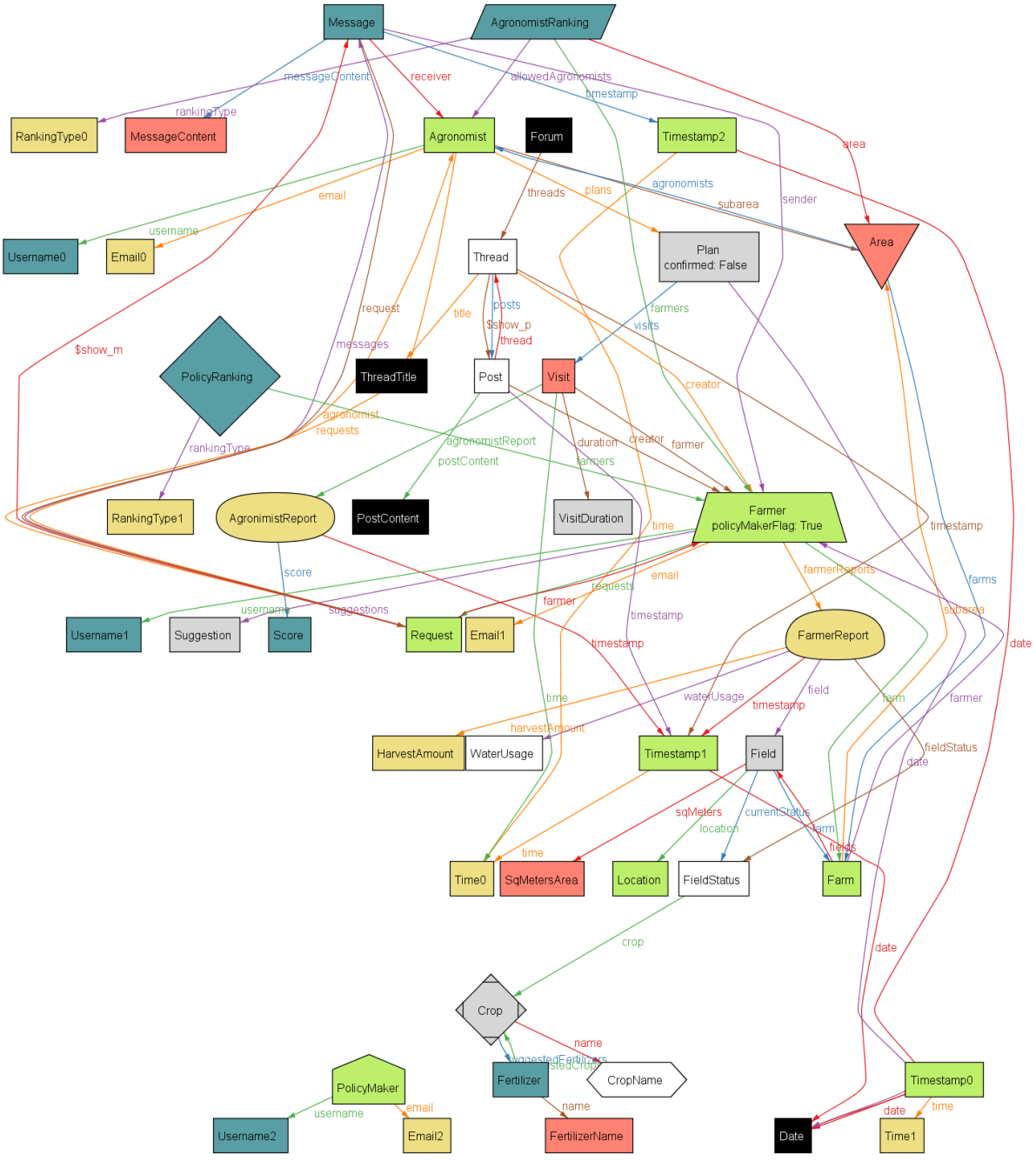

Figure 26: World 1 - Complete model

4.3.2 World 2

In this world we show some farmers sending requests for help to the agronomists and how they can interact with each other exchanging messages.

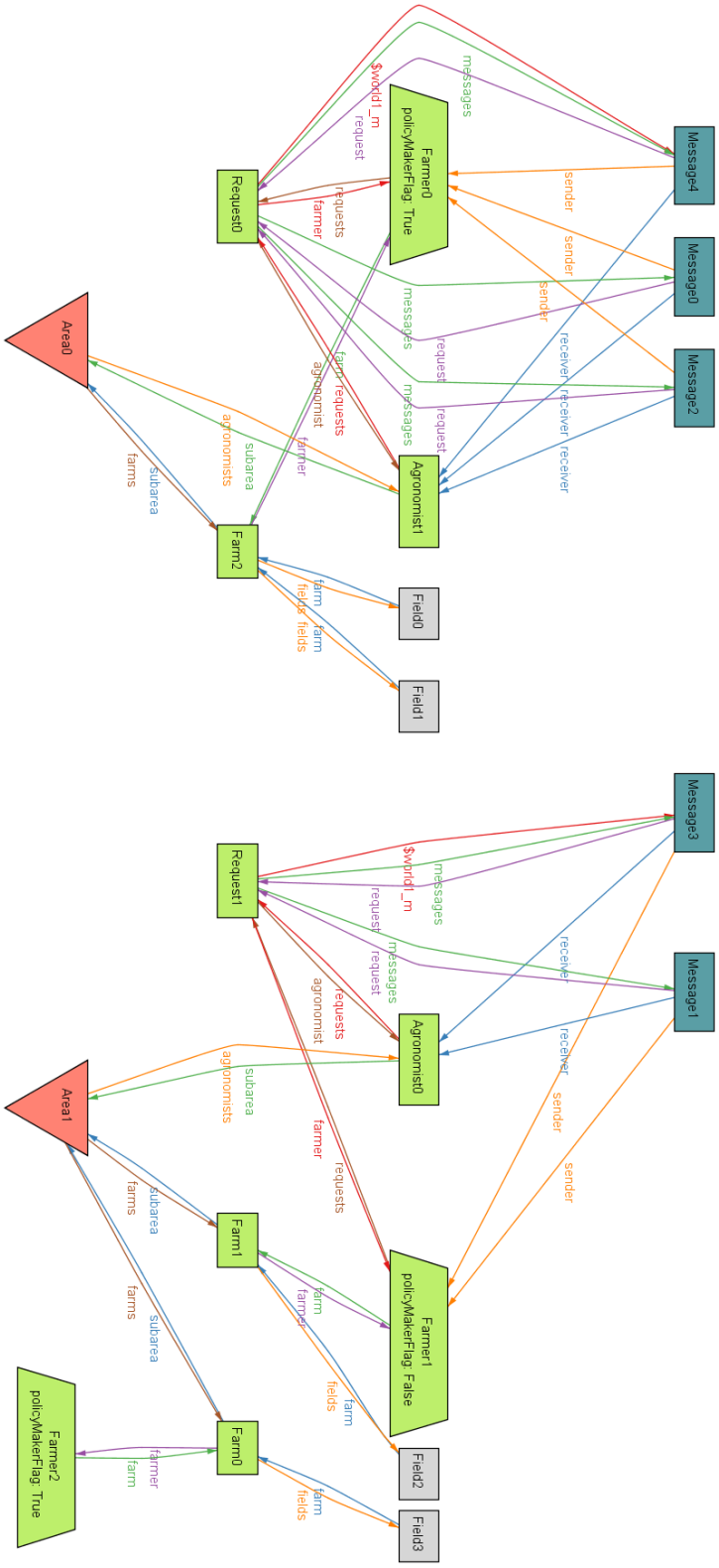

Figure 27: World 2 - Farmer, Agronomist and Request

4.3.3 World 3

In this world instead, we decided to represent many different elements: first of all the daily plan creation and confirmation for the agronomist.
It is possible to see also the reports issued by the agronomist after each visit and both the agronomist and policymaker ranking. The latter contains the list of all the farmers, instead, the agronomist' one is restricted to only the area in which the agronomist belongs.

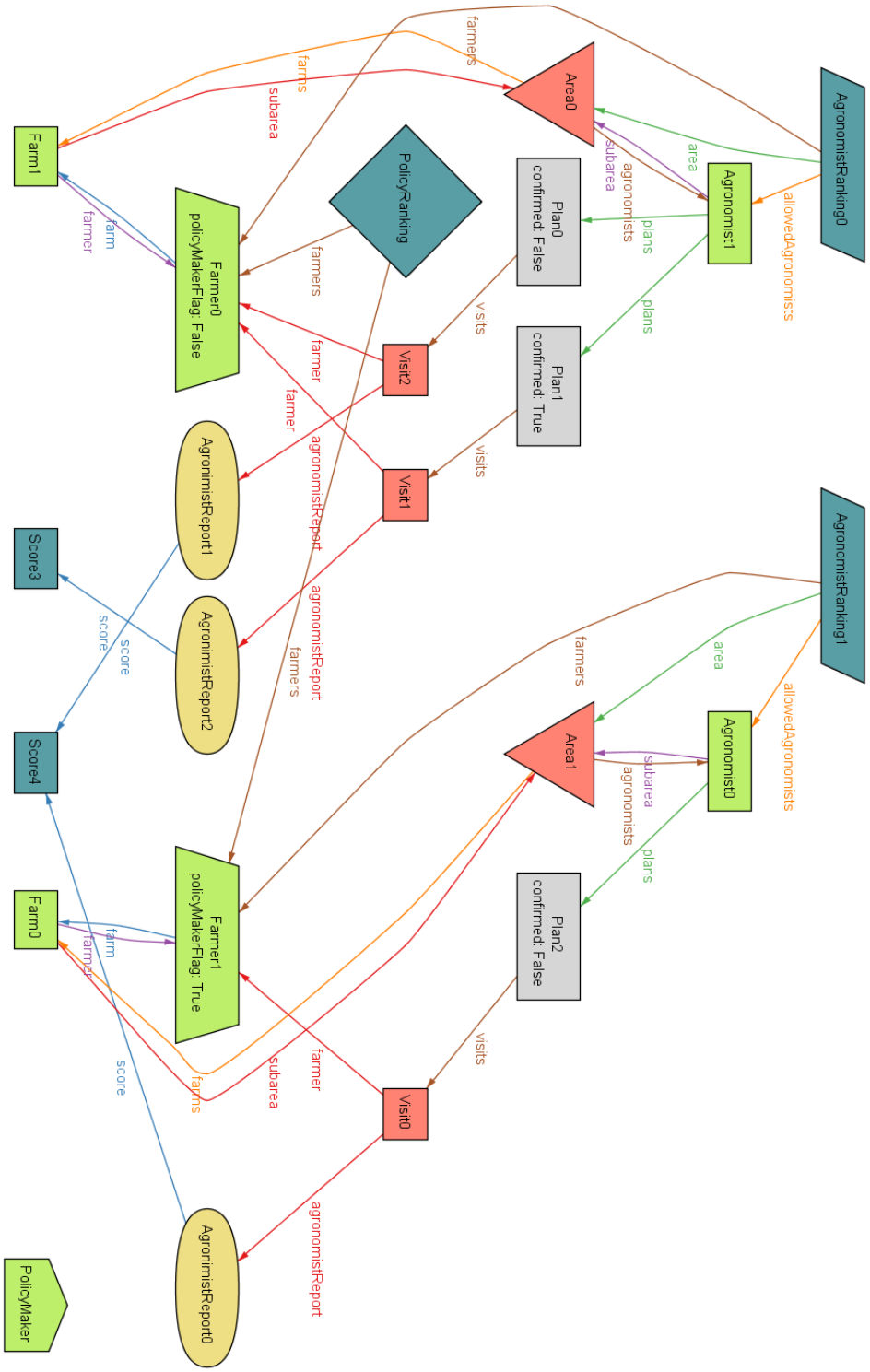

Figure 28: World 3 - Agronomist, Policy Maker and Rankings

4.3.4 World 4

In this last world we wanted to show the functionality of the Farmers Forum with different threads and posts. Each post has its own creator and belongs to only one thread.

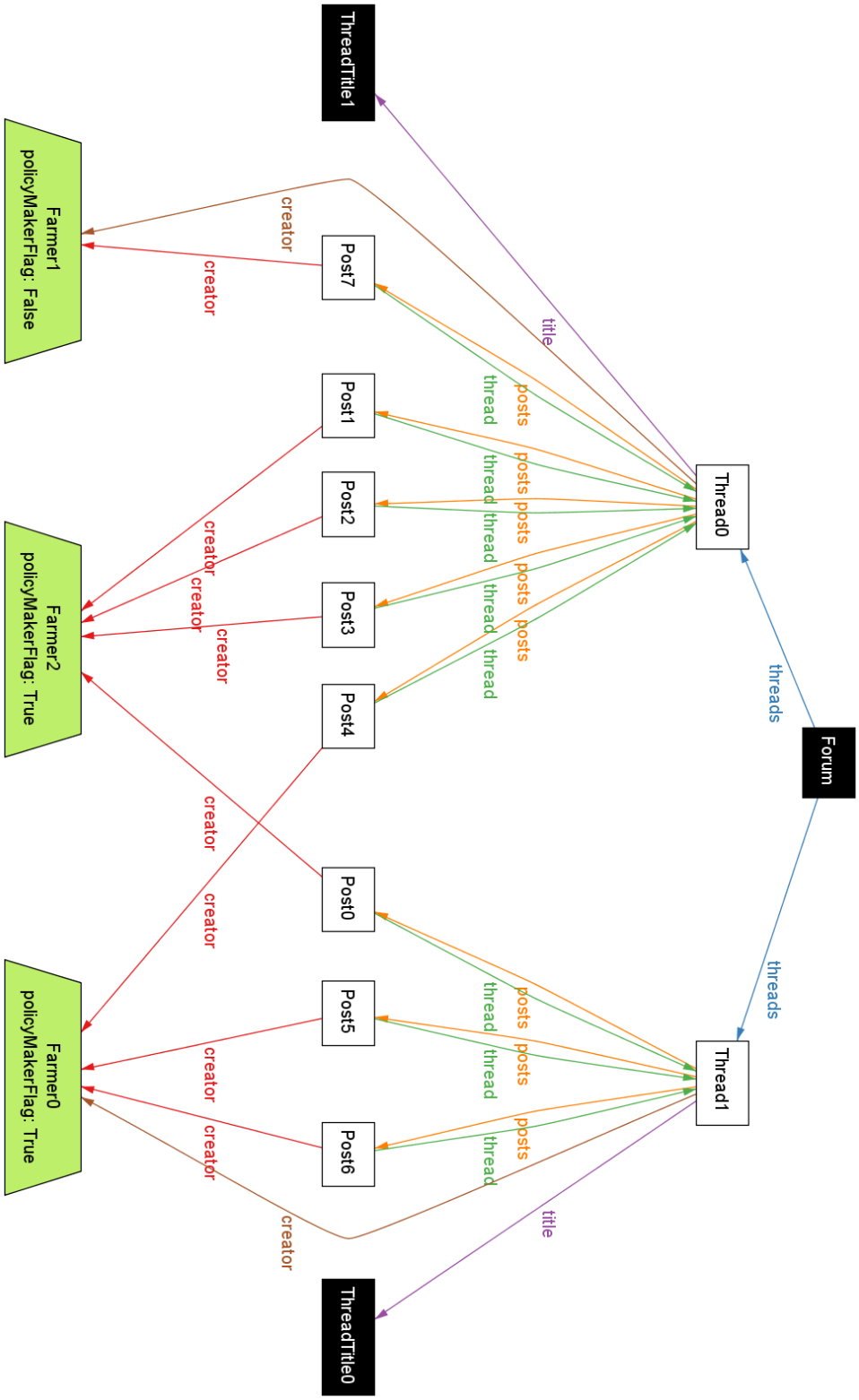

Figure 29: World 4 - Farmer and Forum

5 Effort Spent

Gabriele Marra		
Task	**Time**	**Date**
First Meeting	1.0h	2021/10/20
Goals and phenomena	1.0h	2021/11/20
Goals meeting	1.5h	2021/11/21
Requirements	1.0h	2020/11/23
Working on requirements and domain assumption	3.0h	2021/11/24
Requirements meeting	3.5h	2021/11/25
Scenario and use cases	2.5h	2021/11/29
Scenario and use cases meeting	3.0h	2021/11/30
Use cases meeting	1.5h	2021/12/01
Product function and user description	1.5h	2021/12/03
Alloy meeting	2.0h	2021/12/08
Sequence diagrams	1.5h	2021/12/09
Sequence diagrams meeting	3.0h	2021/12/09
Alloy working	3.0h	2021/12/11
Alloy working	2.5h	2021/12/13
Alloy working	2.5h	2021/12/14
Alloy refinement and testing	3.0h	2021/12/16
Design mockup	1.5h	2021/12/17
Alloy cleaning and description	3.5h	2021/12/18
RASD check	3.0h	2021/12/20
RASD finishing	4.0h	2021/12/21
RASD final review	3.0h	2021/12/22
Total	**52 h**	

Matteo Miceli		
Task	**Time**	**Date**
First Meeting	1.0h	2021/10/20
Goals and phenomena	1.0h	2021/11/20
Goals meeting	1.5h	2021/11/21
Requirements	1.0h	2020/11/23
Working on requirements and domain assumption	3.0h	2021/11/24
Requirements meeting	3.5h	2021/11/25
Scenario and use cases	2.5h	2021/11/29
Scenario and use cases meeting	3.0h	2021/11/30
Use cases meeting	1.5h	2021/12/01
Product function and user description	1.5h	2021/12/03
Alloy meeting	2.0h	2021/12/08
Sequence diagrams	2.0h	2021/12/08
Sequence diagrams meeting	3.0h	2021/12/09
UML and sequence diagrams meeting	2.0h	2021/12/10
Sequence diagrams	2.0h	2021/12/12
Diagrams meeting	1.0h	2021/12/14
Alloy working	2.0h	2021/12/16
Design mockup	1.5h	2021/12/17
Goal mapping	3.0h	2021/12/17
Goal mapping	3.0h	2021/12/18
RASD check	2.0h	2021/12/20
RASD finishing	2.0h	2021/12/21
RASD final review	2.0h	2021/12/22
Total	**47 h**	

Destiny Mora		
Task	**Time**	**Date**
First Meeting	1.0h	2021/10/20
Goals and phenomena	1.0h	2021/11/20
Goals meeting	1.5h	2021/11/21
Requirements	1.0h	2020/11/23
Working on requirements and domain assumption	3.0h	2021/11/24
Requirements meeting	3.5h	2021/11/25
Scenario and use cases	2.5h	2021/11/29
Scenario and use cases meeting	3.0h	2021/11/30
Use cases meeting	1.5h	2021/12/01
Product function and user description	1.5h	2021/12/03
Alloy meeting	2.0h	2021/12/08
Sequence diagrams	1.5h	2021/12/09
Sequence diagrams meeting	3.0h	2021/12/09
UML and other diagrams	2.0h	2021/12/13
Diagrams meeting	1.0h	2021/12/14
Alloy working	1.0h	2021/12/16
Alloy refinement and testing	2.5h	2021/12/16
Design mockup	4.0h	2021/12/17
Section 1 and 2 writing	3.0h	2021/12/19
Section 3 writing	2.0h	2021/12/20
RASD check	3.0h	2021/12/20
RASD finishing	4.0h	2021/12/21
RASD final review	3.0h	2021/12/22
Total	**51.5 h**	