AY 2021/2022

Politecnico di Milano

RASD: Requirement Analysis and Specification Document

Ottavia Belotti Alessio Braccini Riccardo Izzo

Professor
Elisabetta Di Nitto

Version 1.1
December 9, 2021

Contents

1 Introduction **1**
- 1.1 Purpose . . . 1
  - 1.1.1 Goals . . . 1
- 1.2 Scope . . . 4
- 1.3 Definitions, acronyms, abbreviations . . . 5
- 1.4 Revision history . . . 6
- 1.5 Reference documents . . . 6
- 1.6 Document structure . . . 6

2 Overall Description **8**
- 2.1 Product perspective . . . 8
  - 2.1.1 Scenarios . . . 8
  - 2.1.2 Class diagram . . . 12
- 2.2 Product functions . . . 15
  - 2.2.1 Sign-up and shared functions . . . 15
  - 2.2.2 Policy makers functions . . . 15
  - 2.2.3 Farmers functions . . . 16
  - 2.2.4 Agronomists functions . . . 16
- 2.3 User characteristics . . . 17
- 2.4 Assumptions, dependencies and constraints . . . 19

3 Specific Requirements **20**
- 3.1 External Interface Requirements . . . 20
  - 3.1.1 User Interfaces . . . 20
  - 3.1.2 Hardware Interfaces . . . 20
  - 3.1.3 Software Interfaces . . . 20
  - 3.1.4 Communication Interfaces . . . 21
- 3.2 Functional Requirements . . . 21
  - 3.2.1 List of Requirements . . . 21
  - 3.2.2 Mapping . . . 23
  - 3.2.3 Use Cases . . . 32
  - 3.2.4 Sequence Diagrams . . . 55
- 3.3 Performance Requirements . . . 74
- 3.4 Design Constraints . . . 74
  - 3.4.1 Standards compliance . . . 74

3.4.2 Hardware limitations 74
3.5 Software System Attributes 74
3.5.1 Reliability 74
3.5.2 Availability 75
3.5.3 Security 76
3.5.4 Maintainability 76
3.5.5 Portability 76

4 Formal Analysis using Alloy 77
4.1 Formal Analysis Purpose 77
4.2 Alloy Code 78

5 Effort Spent 94

6 References 94

1 Introduction

Data-dRiven PrEdictive FArMing, also known as *DREAM*, is a project presented by UNDP India and Healthsites initiative, promoted by Telangana's government. The aim of the project is to enhance the farm system and the entire food supply chain with an IT supporting application. This arises from modern challenges like climate change and the foreseen population growth that have underlined the critical issues of the modern system making necessary a complete revision.

1.1 Purpose

DREAM aims to support the work categories involved into the farming industry by providing them relevant and up-to-date data about farms activity's performance. The main stakeholders are: Telangana policy makers, farmers and agronomists. The goal is to develop a data-driven application with the help of IT partners. Telangana's state already collect important data concerning wheather forecast; these data are publicly available with a live rainfall map on the official government website. Other data can be collected through humidity sensors deployed all over the territory and through the water irrigation system.

Agriculture has a main role in India's economy, more than half of the population depends on it and about a fifth is below the poverty line. Furthermore, as a significant increment in population is expected by 2050 (*UN*'s esteem), food demand is going to significantly increase. Telangana needs an efficient application to increase the general productivity of the farm system.

The user base is expected to include a large part of Telangana's population, bringing together people with different knowledge backgrounds, such as people directly working in the agricolture field up to political figures.

1.1.1 Goals

Shared goals

1. **(G1) Reach out to farmers in need of help**
   Farmers that struggle with their cultivation wish to be helped and turn around their business in a positive way, since their income depend on that. At the same time, given the forseen food supply demand, policy

makers and agronomists strive for reaching the maximum potential of every suppliers.

Telangana's policy makers

1. **(G2) Identification of well-performing and under-performing farmers**
   Main goal of the policy makers is to identify farmers that are resilient to meteorological adverse events. These farmers will receive special incentives as a compensation for being asked to share their own best practices with the policy maker and agronomist that might apply them on a larger scale later.

   Identify farmers that are performing badly in order to seek the help for them to get back on track. In fact, they are the ones that need to be helped by the agronomists and by regulation promoted by policy makers promoting better practices.

2. **(G3) Visualize the results of steering initiatives**
   Visualize and evaluate the results produced by the steering initiatives from agronomists and good farmers in order to assess which are resulting useful and actually improving the food production system, so that those practices can be applied on a larger scale.

Farmers

1. **(G4) Visualize relevant data for the business**
   Visualize important data like weather forecast and personalized suggestion about specific crops or fertilizers. All data are based on location and type of production.

2. **(G5) Keep track of the production**
   Insert data about their production to identify trends of their own doings, especially bad ones undermining their revenue. Register every type of problem they're experiencing for further inspection with the help of experts in the field.

3. **(G6) Request for help/suggestion**
   Farmers can request help with a text message that will be sent directly to the agronomists responsible of the area to better their practice both in critical conditions and for improvement purposes.

4. **(G7) Create and participate in discussions concerning the agricolture field**
   Create forums to discuss with the other farmers. This feature shall allow a farmer-user to choose the name of a new topic they wish to introduce in the forum and to join all already open discussions. The purpose of this is to ease the cultivation-related knowledge spreading from which every farmer would benefit.

Agronomists

1. **(G8) Receive requests for help/suggestion all in one place**
   Here the agronomist can manage all the incoming request for help or suggestion and will have an higher chance of intervening promptly in critical situations before it's too late. In fact, having just one place in which all the requests that need attention are cumulated makes easier for them to not miss any, rather than losing them because of the different means they could come from and in a unstructured way.

2. **(G9) Visualize area statistics**
   Keep track of the area general performance through a list of best-performing farmers and visualize data about whether forecast.

3. **(G10) Easy daily planning procedure**
   The daily plan consists in a list of farms to be visited during the day. Since, by regulations, every farm must be visited at least twice a year, with particular attention to the under-performing ones that should be visited more often, agronomists wish to be suggested of which are the farms that are due to be visited or the ones that have not been visited yet during the year.

1.2 Scope

Phenomena controlled by the Machine

ID	Phenomenom	Shared
M1	Visualize data concerning weather, land, performance	Yes
M2	User is given statistic data	Yes
M3	User is notified of unread requests for help or suggestions	Yes
M4	User is suggested about which farms need to be visited	Yes
M5	User is presented with news that might be relevant to their business	Yes

Phenomena controlled by the World

ID	Phenomenom	Shared
W1	User logs in	Yes
W2	User queries for the farmers ranking	Yes
W3	User queries for statistic data from humidity sensors	Yes
W4	User queries for statistic data from wather irrigation sensors	Yes
W5	User sends a request for help to an agronomist	Yes
W6	User creates topic in forum	Yes
W7	User joins a discussion in forum	Yes
W8	User inserts post on the forum	Yes
W9	User updates daily plan	Yes
W10	User confirms daily plan	Yes
W11	User pays a visit to a farm	No
W12	User receives a visit in his/her own farm	No
W13	User reads help requests	Yes
W14	User replies to request for help	Yes
W15	User checks weather forecast	Yes
W16	User checks news about crop	Yes

1.3 Definitions, acronyms, abbreviations

Definitions

- **Agronomist**: scientist in the agricoluture field. Their main professional role is to make sure of the well-being of crops and recommend solutions to farmers in need of help with their cultivations
- **Daily plan**: a schedule of the activities to be done during the day, each of them planned out with their own starting time and location
- **Farmer**: workers that own or manage a farm with the intention of selling at least a part of their own production
- **Forum**: online spot enabling users to meet and exchange ideas, share tips, express concerns and arise issues related to topics that comply to the purpose of the forum itself
- **Policy Maker**: political figure responsible or involved in formulating policies, specifically food chain supply policies in this document
- **Production**: in this particular document, it's any type of edible result coming from crop
- **Telangana**: Indian state promoting the *DREAM* project
- **User**: either Policy Makers, Farmers or Agronomists that will use the app

Acronyms

- **AES**: Advanced Encryption Standard
- **API**: Application Programming Interface
- **CE**: Conformité Européenne
- **DBMS**: Database Management System
- **DREAM**: *Data-driven predictive farming* project
- **GPS**: Global Positioning System

- **IT**: Information Technology
- **MTTF**: Mean Time To Failure
- **MTTR**: Mean Time To Repair
- **PDPA**: Personal Data Protection Act
- **RASD**: Requirement Analysis and Specification Document
- **SSL**: Secure Sockets Layer
- **UML**: Unified Modeling Language

1.4 Revision history

1.5 Reference documents

- Specification document: "Assignment RDD AY 2021-2022"
- Alloy documentation: https://alloytools.org/documentation.html
- UML documentation: https://www.uml-diagrams.org/
- BPMN documentation: https://www.bpmn.org/
- Paper: "The World and the Machine" by M. Jackson and P. Zave

1.6 Document structure

- **Section 1** gives an introduction about the problem to tackle and about which functionalities will be implemented in the final product in order to comply to the users' needs.
- **Section 2** contains the overall description of the whole project, presenting it in a more formal way through class diagrams which will show the main flows and extepcted interactions between the user and the system. Furthermore, there will be presented: the so-called *actors* divided into the different types of user that will populate the application; the expected functionalities and the domain assumptions taken in consideration throughout the whole project, from the specification phase to the actual developing phase. To exemplify all the above, typical scenarios will be provided.

- **Section 3** delves deeply into the topics presented in *Section 2*, in order to be more clear in setting the bounds between what the application must implement and what not. It will show functional and non-functional requirements. The former will be presented through some use-cases as meaningful examples; while the latter will be disclosed by analysing performance, design choices and software system features that the project shall have.

- **Section 4** presents the Alloy code briefly explainig the purpose of it in modeling the given problem.

2 Overall Description

2.1 Product perspective

DREAM is a system that offers the functionalities described in the *Product Functions (2.2)* section. It manages to use some external interfaces, for more details refers to the *Software Interfaces (3.1.3)* section. In the following subsections we can see a list of the most common scenarios *(2.1.1)* and an high-level class diagram *(2.1.2)* that shows the basic structure of the system.

2.1.1 Scenarios

1. Aanya has just passed her test to become a policy maker and wants to consult the app. To do so she decides to download and install DREAM in order to visualize the data concerning her job.
    - She opens the app and click on the "Sign Up" button
    - She inserts all her data in order to fill every mandatory field and presses the "Confirm" button
    - The system verifies that the e-mail isn't already present in the database. Since there isn't an entry for her e-mail yet, it allows the registration and shows a confirmation message to Aanya's mailbox
    - She checks her mailbox in order to verify that the procedure ended successfully
2. Apu is an expert policy maker, he wants to know what are the latest data regarding soil humidity and water irrigation.
    - He takes his phone and logs in DREAM app
    - He presses the "Visualize Data" button
    - The system retrieves data from a database and shows him the soil moisture and the consumed liters of water by the irrigation system thorughtout graphs that computes the average quantities across the Country in a span of one year
    - He wishes to have more detailed information about the soil humidity and the water irrigation in Adilabad district since he read an alarming report about that region. He choses it from the menu and sets the time span at two weeks

- The system gives him new graphs reporting only data about soil humidity and water used for irrigation retrieved by Adilabad's sensors in the last two weeks
- He analyzes them and decides that he will talk about it later with a colleague, then he returns to the home page

3. Pajeet is one of the main policy makers in the state of Telangana. He wants to distribute special incentives to the best farmers and to do this he decides to open the app and visualize the data.
    - He opens the app on his device. Since he's already registered, he logs in
    - The system shows the homepage for a policy maker user
    - He visualizes the ranking of the well-performing farmers
    - He clicks on the name of a farmer
    - Now the system shows the profile of the selected farmer
    - Here the policy maker can find useful contact information such as phone number (if it has been given by the farmer) and email
    - He takes note of the desirable contact information and closes the app
4. Chandran is an old farmer that has been cultivating mais for all his life. He has recently decided to move out from his previous field in order to give it to his son, Aakesh. Aakesh has already his own land near his dad's one, so he wants to add the new gifted area to his DREAM account. Furthermore, he found out that wheat is more productive and more bug-resistant than mais, so he also wants to change the crop type for both the lands from mais to wheat.
    - He logs in DREAM app
    - He presses the "Profile Edit" button in order to change his area and crop type
    - He changes the two value by inserting a new area and a new crop type
    - He confirms by pressing the "Confirm" button

5. Sahil is a farmer that would like to share some best-practices about a particular product that he has cultivated during the last year. He decides to open a new discussion on the forum.

    - He opens the app on his device. Since he's already registered, he logs in
    - The system shows the homepage for a farmer user
    - He clicks on the "Forum" button and access the forum section
    - Now he clicks on the "Create new topic" button where he can select the name of the discussion
    - He posts a new message in the discussion
    - Finally some time later he checks if someone has joined his topic and replied to him

6. Damayanti is a farmer that would like to improve the productivity of his farm. To do so he decides to consult the section regarding the suggestions on specific crops to plant or specific fertilizers to use.

    - He opens the app on his device, if he is already registered he logs in
    - The system shows the homepage for a farmer user
    - He clicks on the "News" button and access the news section
    - Here he visualizes suggestions based on his type of production and on his location
    - If he is not satisfied, then he clicks on the "Request for help" button to send a message
    - Here the app suggests him the agronomist responsible of the area and opens a text section where he can type his requests for suggestions
    - Otherwise, he takes note from the article he read
    - He closes the app

7. Chakrika is a young farmer and decided to insert the data regarding last week production.

    - She takes her computer and logs in DREAM

- The system shows her the farmers home page
- She presses the "Insert production data" button
- She inserts the total quantity of rice harvested last week
- She checks the inserted data and finally press the "Confirm" button
- The system sends her back to the home page

8. Shaleena is an agronomist responsible of Mahbubnagar, one of the biggest region in Telangana. She wants to compose the daily plan to decides which farms to visit today and for this reason she decides to open the app.

    - She opens the app on his device, if she is already registered she logs in
    - The system shows the homepage for an agronomist user
    - She clicks on the "Daily plan" button and access the new section
    - She adds the farms she wants to visit to the daily plan by choosing them from the list of available ones
    - Now the system shows the addresses and the date of the last visit of the selected farms
    - She takes note of the addresses and close the app

9. Chaitanya is an agronomist that after a hard day of work wants to update the daily plan on the app. At the end of the day she realizes that she has visited three farms instead of the two indicated in the daily plan.

    - She opens the app on his device, if she is already registered she logs in
    - The system shows the homepage for an agronomist user
    - She clicks on the "Daily plan" button and access the new section
    - Here she can visualize the daily plan
    - She can click on the "Add" or on the "Remove" button

- In case of "Add" the system would redirect her to the list of available ones, in case of "Remove" she can delete a farm from the list simply by clicking on the cross at the right
- She clicks on the "Confirm" button to validate the daily plan
- The system returns to the homepage
- Finally she closes the app

10. Pradeep is an agronomist who wants to take care about help requests before going to work.
    - He opens the app on his device, if he is already registered he logs in
    - The system shows the homepage for an agronomist user
    - He clicks on the "Mail box" button and access the new section
    - Here he visualizes a list of help requests
    - By clicking on a help request he can visualize the request and answer it
    - He does this for every help request until they are finished
    - Finally he closes the app and begins his working day
11. Sonya is an agronomist that want to check who are the best farmers of her responsibility area.
    - She take her IT device and log in the system
    - She can already see who are the top farmers of the area
    - She press the widget and look deeply the ranking
    - She press the "Home" button to return to home page

2.1.2 Class diagram

A UML class diagram describes the main entities in the system. Three types of actors can access the system: farmers, policy makers and agronomists. During the sign-up, they all create a user account with basic information such as username, password and e-mail. At this point, users are requested to choose what type of account to create. It is crucial that phisycal users

register themselves accordingly to their job to ensure that the system works properly. Furthermore, this choice affects the next steps in the registration phase; for example, to complete the registration, farmers need to specify the position of their farm and what type of crop they grow, whereas agronomists must declare their area of responsibility.

The system provides several functionalities tailored for the type of user, one of these is the forum for the farmers. When the farmers select the forum section, they are presented with the topics that are already active. For each topic, the discussion that has evolved through time can be seen: it's composed by the original post and its replies from other users. From here, the user can join an already ongoing discussion or create a new one generating a new topic and post the first message for it.

Another functionality is the *daily plan* feature for the agronomists. A daily plan is a list of farms that an agronomist has decided to visit on a particular day of the year, it has a date and a status flag that indicates whether the plan has already been confirmed or not[1]. For the planner, agronomists have access to the list of farmers currently in their area of responsibility and registered to the app.

Finally, there are various types of data that can be visualized and analyzed by the users depending on the type of their account. There are four types of data:

- **Soil humidity sensors data**
  Acquired by the sensors distributed all over the territory, they report values that represent the soil moisture level of the land they cover.

- **Weather forecast data**
  Meteorological short-term and long-term forecast acquired by Telangana's government and displayed on their official website. They are comprehensive of temperature, rainfall probability and quantity, direction and strength of the wind and air humidity.

- **Water irrigation data**
  Acquired by the water irrigation system installed in the farms, the only significant value is the amount of water used by the system during the day.

[1]More technical details can be found in the DD document.

- **Production data**
  Provided by the farmers, this type of data is displayed as a table that maps the type of cultivated product to the sown amount and the respective harvested quantity.
- **Report**
  Reports are written by Agronomists whenever they lead new initiatives aiming to better the results of cultivation in their area. The reports include a detailed explanation of the content of the initiative and the feedback they've been collecting since their attuation.

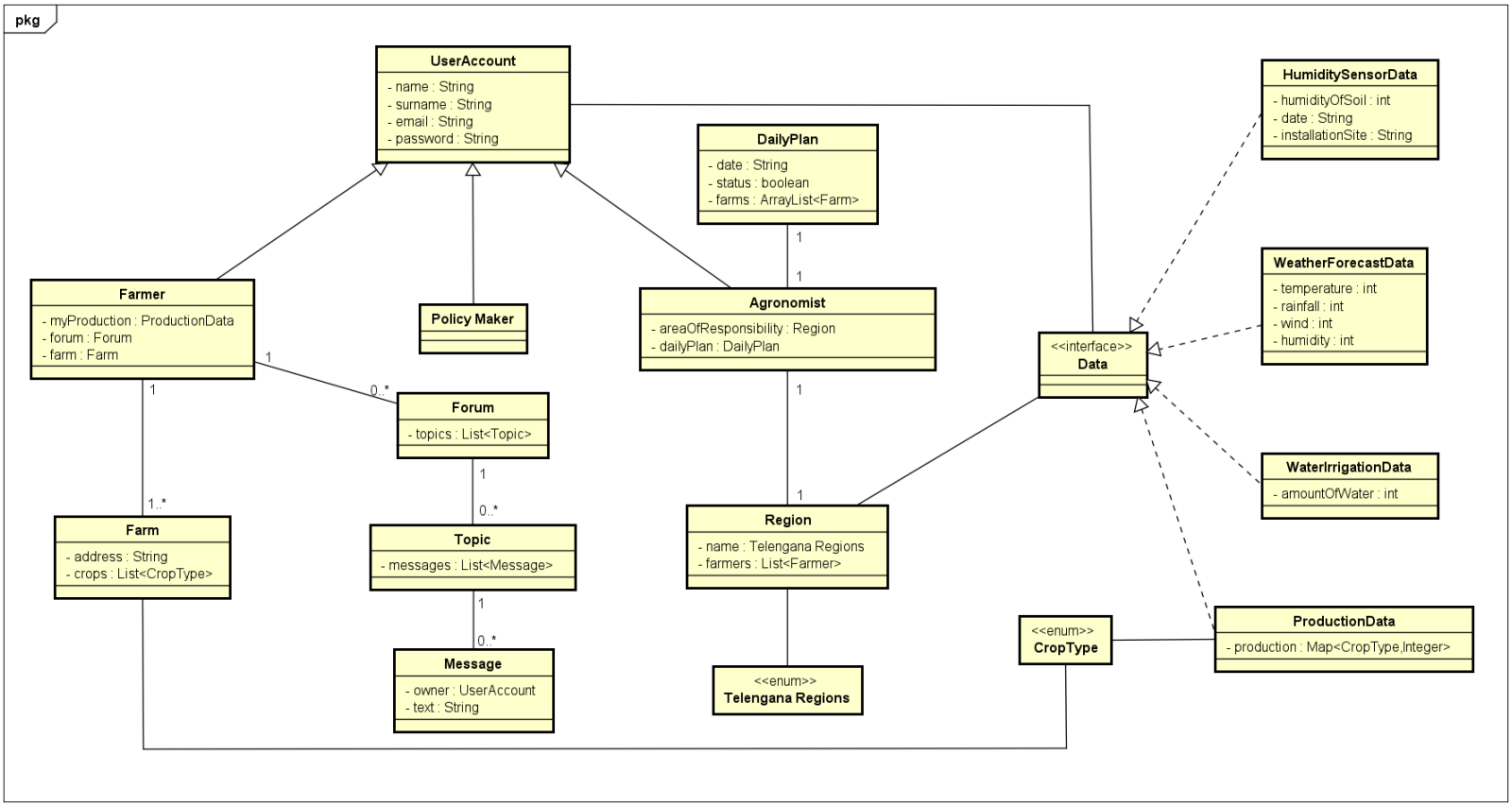

Figure 1: High-level UML

2.2 Product functions

2.2.1 Sign-up and shared functions

- **Sign-up:** let the user sign-up thorugh an email and a password, creating a profile tailored for the user's job. Specify the area in which they live and what type of cultivation they manage[2]. The adding of a farmland is done by inserting its coordinates (raw coordinates or by selecting a spot on the map) into the app, since it's difficult to find a location through an address in rural areas.

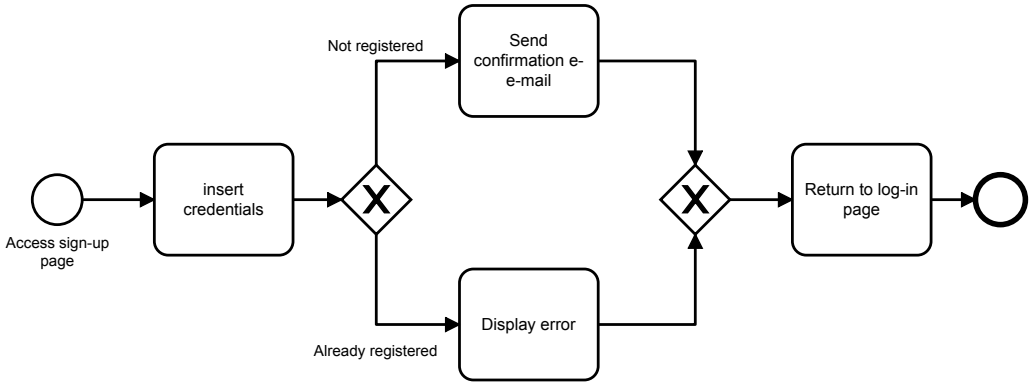

Figure 2: Sign Up BPMN

2.2.2 Policy makers functions

- **Visualize relevant data and initiatives:** let the policy makers know a variety of data like the performances of the farmers by grouping them in a rank to know who are the farmers that are performing well and who are the worst one mainly based upon the production data provided by them. Policy makers can also visualize the steering iniziatives presented by the agronomists' reports that are collected into a specific subsection of their application.

[2]The positioning of a farm is tackled more in depth in the DD document.

2.2.3 Farmers functions

- **Getting information about the agricolture field:** allow farmers to visualize news relevant to their business interests such as: wheather forecast for the day or for the next few days, articles posted on major online farming magazines that are related to their own crop type and specific fertilizers for their needs.
- **Agronomists Interactions:** allow farmers to send messages to the agronomist.
- **Building a useful community on the forum:** allow farmers to create a new topic or reply to a message in the dedicated forum.

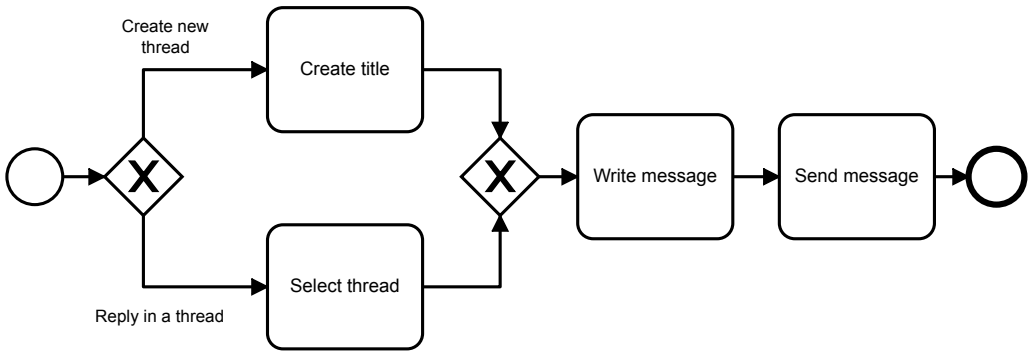

Figure 3: Forum BPMN

2.2.4 Agronomists functions

- **Area functions:** allow the agronomists to insert their responsibility area and visualize related data like wheather forecast or the rank of the farmers in that region.
- **Manage farmers requests:** agronomists reply to farmers' requests for help or suggestion.

- **Manage daily plan:** allow the agronomist to make their daily plan by registring the scheduled visits to the farmers. At the end of the day an agronomist must confirm if they were able to complete the plan or specify the deviations occurred during the day. Updating the daily plan because of a deviation means either adding a farm that has been visited during that day that was not originally in the plan, or removing one from the schedule if the agronomist has not been able to pay a visit that same day.

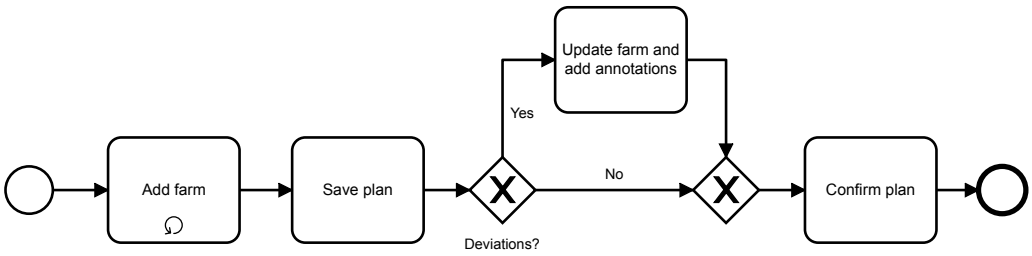

Figure 4: Daily plan BPMN

2.3 User characteristics

The application has been thought for the three different user categories that follows:

- **Policy makers** are government's employees that are in charge of analysing the general agricolture trends among all the districts in Telangana, then promote based state-wide policies to better the whole food system. Their main goal is not only to secure the current provision, but also to identify now the best practices that will lead to a flourishing food production in the future. By doing so, the plan is to grow more resilient and profitable crops and prepare the lands to face future menaces, for instance the climate that is getting more hostile or the foreseen increment of the food demand. As a consequence, they want to be notified about the best performing farmers in order to contact them and get more insight from them about their procedures, with

the aim to acquire best practises to be shared and applied on a larger scale. At the same time, they need to know who, on the other hand, is performing particularly badly, so that they can be given the help they need to better their results, since obtaining the foresaid goal requires the structure to run smoothly in all its parts. Policy makers also need a feedback system that let them be aware of the true impact *a posteriori* of the initiatives carried out by the agronomists in collaboration with the knowledge and practice of the best farmers.

- **Farmers** are interested in functionalities that will help with their day-to-day life at work, so they would like to receive in one place all the information about the weather to plan before hand the work day and useful data, like suggestions and news about the specific crop they cultivate, if some crop's illness is spreading in their area and how to treat it, which fertilizers boost the plant's production, etc. Moreover, they should insert data about their own production and ask for help to a regional agronomist through the app if it's needed. Being part of a larger community of people that share the same purpose (such as being more productive) brings more knowledge in general, so it's easier for the farmers to get in touch with their collegues that grow the same crops and might have faced the same challenges they do through the in-app forum. The feature allows them to enlarge their pool of acquaintances and brings them together online, even though they might be kilometers away from each other.

- **Agronomists** are the experts in the agricolture field, so the main function needed for them is the possibility of helping out the farmers that reach out to them. Each agronomist is in charge of a specific geographical area in Telangana, in order to be efficiently present on the territory in a fair and useful way according to the actual helping demand. In fact, they visit each farm spread among their area at least twice a year. That said, agronomists would like some functionalities that help them planning out their trips on the field in an simple yet flexible way. Furthermore, they would like to be notified of the farms' performace, especially the ones scoring poor results in order to plan their visits more often for those, depending on the problem their facing. Nonetheless, in order to make a complete and exhaustive report about the area productivity for the central government, they are also

interested into acknowledging the top performing farms.

2.4 Assumptions, dependencies and constraints

- D1: Agronomists actually stick to the scheduled daily plan
- D2: Agronomists correct their schedule at the end of the day if deviations occur
- D3: Each user registers only once, according to their job, and always feeds correct information to the app
- D4: The internet connection works properly when the application is being used[3]
- D5: The sensors measuring humidity level of the soil work properly, with at most an error of 1%
- D6: Weather forecasts are accurate up to a 80% for the short-term and up to 60% on the long-term
- D7: Water irrigation system send the information accurately, with an error of at most 1%
- D8: Agronomists always reply to farmers requests of help
- D9: Farmers periodically insert data about their crop status and resulting production
- D10: Data inserted by the Farmers sticks to the reality of the produced quantity.
- D11: Agronomists periodically report their initiatives to the central government.
- D12: Users own an IT device to connect to the application
- D13: Devices external to the system (e.g. humidity sensors, water irrigation systems, weather stations) are active and working properly for 99.9% of the year
- D14: Each Telangana's district has one and only one Agronomist[3]

[3]This constraint is purposely stronger than what it could be in order to simplify a

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The system should interface with users through devices which must be connected to the Internet. Everyone that needs to use this service would connect to it through a web broswer from an existing domain, for example: dream.com. It must be easy to use as it will have to be used from different kind of people, sometimes with a not great affinity with technologies.

3.1.2 Hardware Interfaces

The system "as it is" does not provide specific hardware equipment in order to access to web site. To use all functionalities it's required to have an IT devices connected to the internet, eventually a user can decide to use GPS to insert his own location and only proper devices with GPS equipment can do this.

Soil data, obtained from sensors, are collect by someone else and will be added by the policy makers, also this not requires any special kind of hardware.

3.1.3 Software Interfaces

In order to use the system, web broswer is required to run the application.

The system will also take advantage of some interfaces in order to accomplish its functionalities.

Firstly, an external API is necessary to visualize on maps a farm's position or to let farmers locate their farm in registration phase.

Then, another API service will be the one of the DBMS system which will be adopted in order to query the internal database.

Finally, an API will be use to access the external database containing data regarding the sensors data and the other external systems data that the application needs.

---

future implementation of the prototype (e.g. it won't allow local storaging of data while the device is not connected to internet, just one agronomist per district, etc.). It could be relaxed in a further release of the application.

3.1.4 Communication Interfaces

Internet connection is mandatory in order to access to every formation that the system will display. Users that want full functionalities have to have a GPS system on their device in order to guarantee a certain level of precision in the localization phase.

3.2 Functional Requirements

3.2.1 List of Requirements

ID	Requirement
R1	The system must allow only registered and logged-in users to use the app.
R2	The system shall allow users to be identified by an email of their choosing.
R3	The system must suggest to the users the area in which they are during the registration phase if the IT device is equipped with GPS technology. In any case, the final decision will be up to the registering user.
R4	The system must inform the user to try later if they are experimenting connectivity issues .
R5	The system must allow Policy Makers to see up-to-date statistic data, accordingly to the database, about water irrigation systems and soil humidity sensors.
R6	The system must allow Policy Makers to retrieve all agronomists' steering initiative reports uploaded to the database.
R7	The system must give Policy Makers up-to-date ranking of the best and worst performing farmers.
R8	The system must use a fair and scientific score to rank the Farmers in order to represent the real situation.
R9	The system must let Farmers insert their production information every day.

R10	The system must provide the contact of the agronomist appointed to the area of the Farmer requesting professional help.
R11	The system must allow every user registered as Farmer to access the forum, to create new discussions and to post replies to already existing ones.
R12	The system must notify Agronomists of unresolved requests of help from Farmers.
R13	The system must allow the Agronomists to add a new schedule for the day each day.
R14	The system must suggest to the Agronomists which farms to visit while planning the Daily Plan based upon the last visit day following a FIFO policy. Exceptions to the FIFO policy are the farms underperforming, which shall have higher priority.
R15	The system must suggest to the Agronomists only the farms among the ones in their competence area.
R16	The system must allow the Agronomists to update their schedule during the day and to confirm the execution before the end of the day.
R17	The system must register the already uploaded schedule as definitive if the user doesn't confirm the daily plan for day x before 23:59 of the day x.
R18	The system must give Agronomists up-to-date ranking of the best and worst performing farmers in their responsibility area.
R19	The system must show weather forecasts relevant to the area concerning the Farmer or the Agronomist.
R20	The system must present news concerning crop only if relevant to the Farmer's own crop type.

3.2.2 Mapping

G1	Reach out to farmers in need of help
D3	Each user registers only once, according to their job, and always feeds correct information to the app
D4	The internet connection works properly when the application is being used
D8	Agronomists always reply to farmers requests of help
D12	Users own an IT device to connect to the application
D14	Each Telangana's district has one and only one Agronomist
R1	The system must allow only registered and logged-in users to use the app.
R2	The system shall allow users to be identified by an email of their choosing
R4	The system must inform the user to try later if they are experimenting connectivity issues .
R7	The system must give Policy Makers up-to-date ranking of the best and worst performing farmers.
R10	The system must provide the contact of the agronomist appointed to the area of the Farmer requesting professional help.
R12	The system must notify Agronomists of unresolved requests of help from Farmers
R14	The system must suggest to the Agronomists which farms to visit (among the ones in their area of competence) while planning the Daly Plan based upon the last visit day following a FIFO policy. Exceptions to the FIFO policy are the farms under-perfoming, which shall have higher priority.
R15	The system must suggest to the Agronomists only the farms among the ones in their competence area.
R18	The system must give Agronomists up-to-date ranking of the best and worst performing farmers in their responsibility area

G2	Identification of well-performing and under-performing farmers
D3	Each user registers only once, according to their job, and always feeds correct information to the app
D4	The internet connection works properly when the application is being used
D9	Farmers periodically insert data about their crop status and resulting production
D10	Data inserted by the Farmers sticks to the reality of the produced quantity.
D12	Users own an IT device to connect to the application
R1	The system must allow only registered and logged-in users to use the app.
R2	The system shall allow users to be identified by an email of their choosing
R4	The system must inform the user to try later if they are experimenting connectivity issues .
R7	The system must give Policy Makers up-to-date ranking of the best and worst performing farmers.
R8	The system must use a fair and scientific score to rank the Farmers in order to represent the real situation.
R9	The system must let Farmers insert their production information every day.
R18	The system must give Agronomists up-to-date ranking of the best and worst performing farmers in their responsability area

G3	Visualize the results of steering initiatives
D3	Each user registers only once, according to their job, and always feeds correct information to the app.
D4	The internet connection works properly when the application is being used
D5	The sensors measuring humidity level of the soil work properly
D7	Water irrigation system sends the information accurately, with an error of at most 1%
D9	Farmers periodically insert data about their crop status and resulting production
D10	Data inserted by the Farmers sticks to the reality of the produced quantity
D11	Agronomists periodically report their initiatives to the central government.
D12	Users own an IT device to connect to the application
D13	Devices external to the system (e.g. humidity sensors, water irrigation systems, weather stations) are active and working properly for 99.9% of the year
D14	Each Telangana's district has one and only one Agronomist
R1	The system must allow only registered and logged-in users to use the app.
R2	The system shall allow users to be identified by an email of their choosing
R4	The system must inform the user to try later if they are experimenting connectivity issues .
R6	The system must allow Policy Makers to retrieve all agronomists' steering initiative reports uploaded to the database.

G4	Visualize relevant data for the business
D3	Each user registers only once, according to their job, and always feeds correct information to the app.
D4	The internet connection works properly when the application is being used
D6	Weather forecasts are accurate up to a 80% for the short-term and up to 60% on the long-term.
D12	Users own an IT device to connect to the application
R1	The system must allow only registered and logged-in users to use the app.
R2	The system shall allow users to be identified by an email of their choosing
R3	The system must suggest to the users the area in which they are during the registration phase if the IT device is equipped with GPS technology. In any case, the final decision will be up to the registering user
R4	The system must inform the user to try later if they are experimenting connectivity issues .
R19	The system must show weather forecasts relevant to the area concerning the Farmer or the Agronomists.
R20	The system must present news concerning crop only if relevant to the Farmer's own crop type.

G5	Keep track of the production
D3	Each user registers only once, according to their job, and always feeds correct information to the app
D4	The internet connection works properly when the application is being used
D9	Farmers periodically insert data about their crop status and resulting production.
D10	Data inserted by the Farmers sticks to the reality of the produced quantity
D12	Users own an IT device to connect to the application
R1	The system must allow only registered and logged-in users to use the app
R4	The system must inform the user to try later if they are experimenting connectivity issues .
R9	The system must let Farmers insert their production information every day.

G6	Request for help/suggestion
D3	Each user registers only once, according to their job, and always feeds correct information to the app.
D4	The internet connection works properly when the application is being used.
D8	Agronomists always reply to farmers' requests of help.
D12	Users own an IT device to connect to the application
D14	Each Telangana's district has one and only one Agronomist
R1	The system must allow only registered and logged-in users to use the app.
R4	The system must inform the user to try later if they are experimenting connectivity issues .
R10	The system must provide the contact of the agronomist appointed to the area of the Farmer requesting professional help.
R12	The system must notify Agronomists of unresolved requests for help from Farmers.
R20	The system must present news concerning crop only if relevant to the Farmer's own crop type

G7	Create and participate in discussions concerning the agricolture field
D3	Each user registers only once, according to their job, and always feeds correct information to the app.
D4	The internet connection works properly when the application is being used.
D12	Users own an IT device to connect to the application
R1	The system must allow only registered and logged-in users to use the app.
R4	The system must inform the user to try later if they are experimenting connectivity issues .
R11	The system must allow every user registered as Farmer to access the forum, to create new discussions and to post replies to already existing ones.

G8	Receive requests for help/suggestion all in one place
D3	Each user registers only once, according to their job, and always feeds correct information to the app
D4	The internet connection works properly when the application is being used.
D12	Users own an IT device to connect to the application
D14	Each Telangana's district has one and only one Agronomist
R1	The system must allow only registered and logged-in users to use the app.
R4	The system must inform the user to try later if they are experimenting connectivity issues .
R10	The system must provide the contact of the agronomist appointed to the area of the Farmer requesting professional help.
R12	The system must notify Agronomists of unresolved requests for help from the Farmers.

G9	Visualize area statistics
D3	Each user registers only once, according to their job, and always feeds correct information to the app
D4	The internet connection works properly when the application is being used.
D9	Farmers periodically insert data about their crop status and resulting production.
D10	Data inserted by the Farmers sticks to the reality of the produced quantity
D12	Users own an IT device to connect to the application
R1	The system must allow only registered and logged-in users to use the app.
R4	The system must inform the user to try later if they are experimenting connectivity issues .
R8	The system must use a fair and scientific score to rank the Farmers in order to represent the real situation.
R18	The system must give Agronomists up-to-date ranking of the best and worst performing farmers in their responsibility area.
R19	The system must show weather forecasts relevant to the area concerning the Farmer or the Agronomist.

G10	Easy daily planning procedure
D1	Agronomists actually stick to the scheduled daily plan.
D2	Agronomists correct their schedule at the end of the day if deviations occur.
D3	Each user registers only once, according to their job, and always feeds correct information to the app
D4	The internet connection works properly when the application is being used.
R1	The system must allow only registered and logged-in users to use the app.
R4	If users are experimenting connectivity issues,the system must inform them to try later.
R13	The system must allow the Agronomists to add a new schedule for the day each day.
R14	The system must suggest to the Agronomists which farms to visit while planning the Daily Plan based upon the last visit day following a FIFO policy. Exceptions to the FIFO policy are the farms underperforming, which shall have higher priority.
R15	The system must suggest to the Agronomists only the farms among the ones in their competence area.
R16	The system must allow the Agronomists to update their schedule during he day and to confirm the execution before the end of the day.
R17	The system must register the already uploaded schedule as definitive if the user doesn't confirm the daily plan for day x before 23:59 of the day x.

3.2.3 Use Cases

3.2.3.1 Use Cases Diagram

- **Policy Makers**

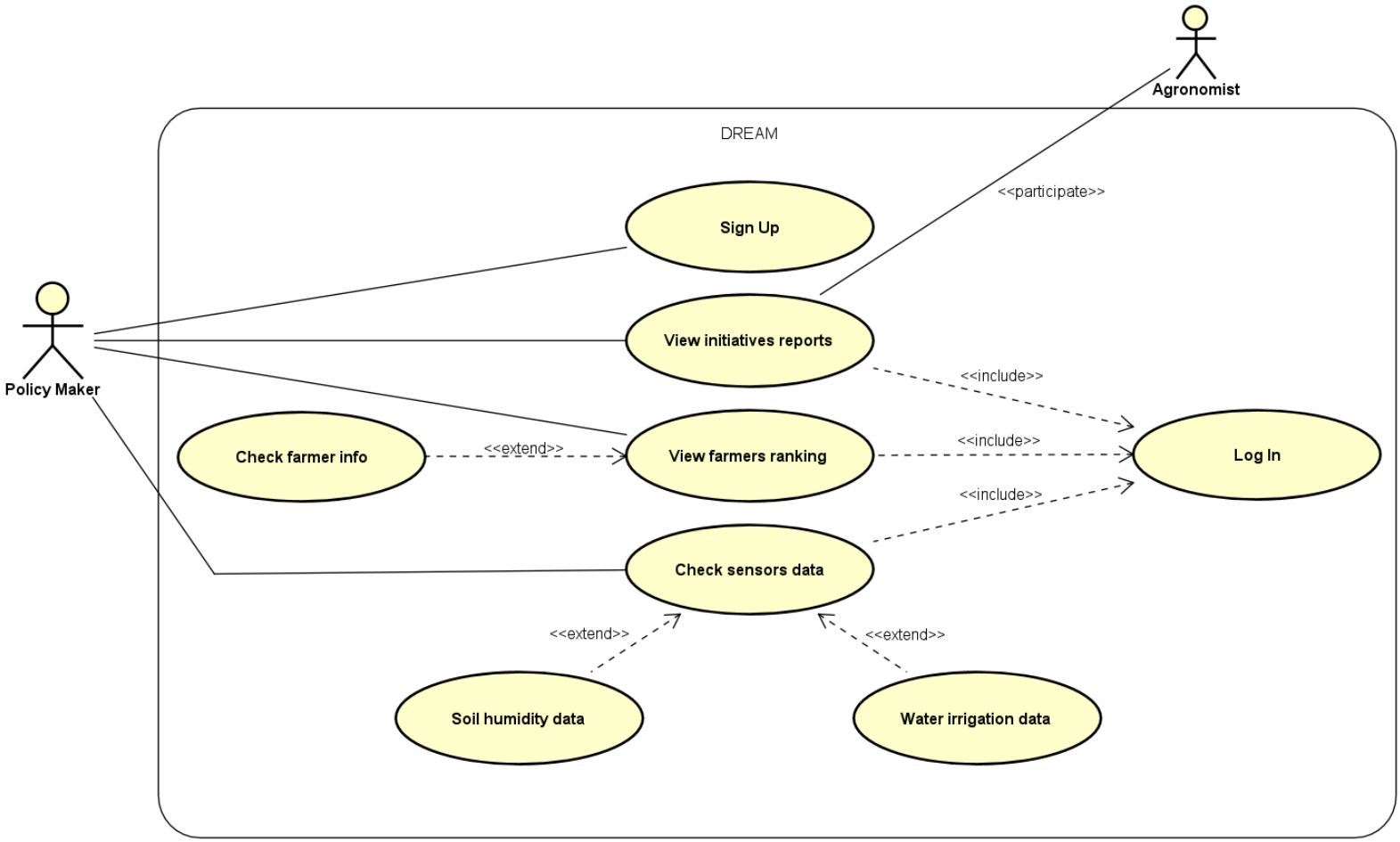

Figure 5: Policy Maker - Use Case Diagram

- **Farmers**

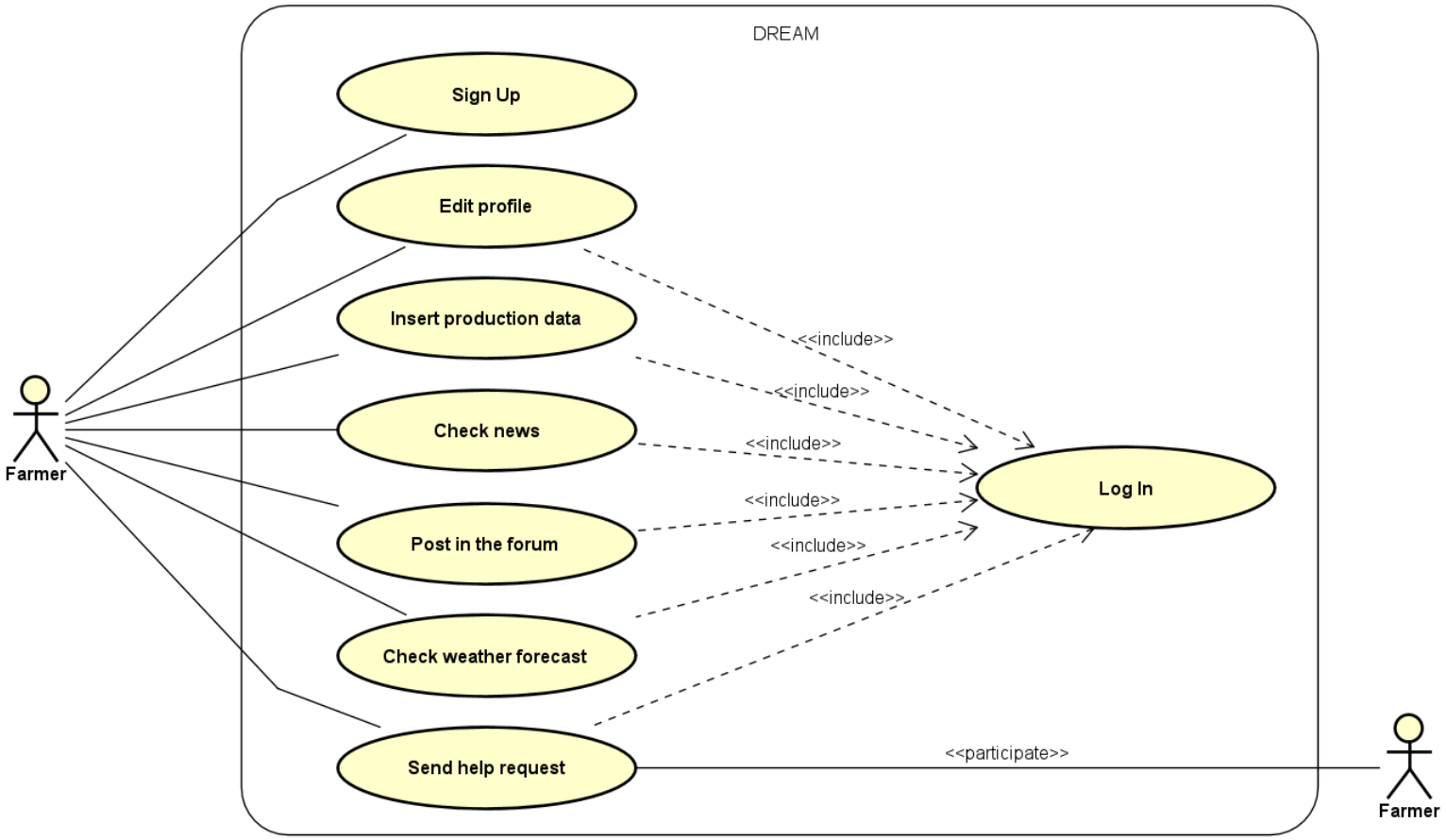

Figure 6: Farmer - Use Case Diagram

- **Agronomists**

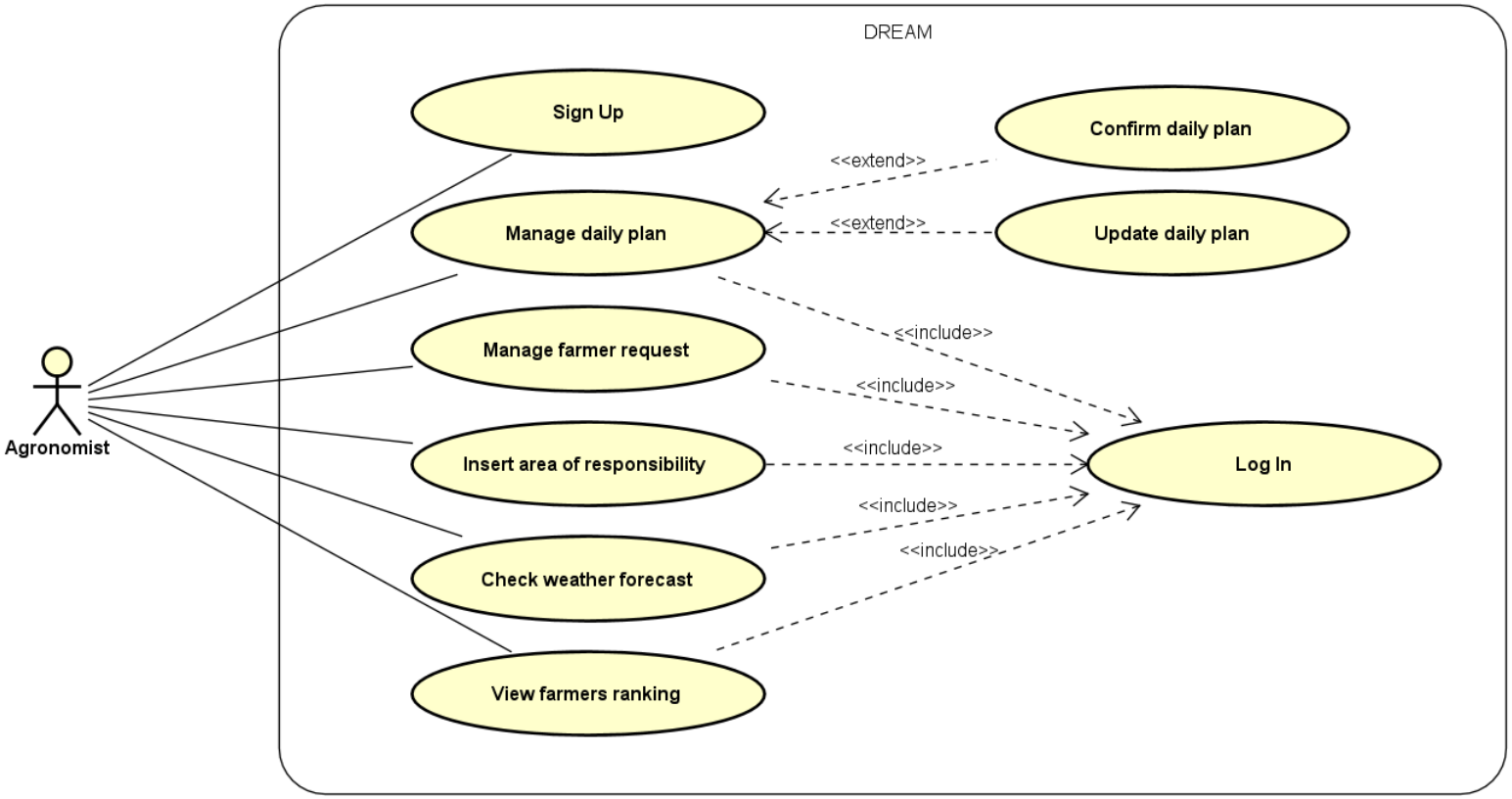

Figure 7: Agronomist - Use Case Diagram

3.2.3.2 Use Cases Description

- **Shared Use Cases**

Sign Up

Use Case	Sign Up
Actor	Policy Maker, Farmer, Agronomist
Entry condition	User wants to register in the system
Flow of events	1. User opens DREAM app 2. User presses the sign up button 3. User selects its username, password and job role 4. User presses the confirm button 5. System shows a confirmation message to the user 6. An email is sent to the user
Exit condition	User data are saved into the system and registration ends successfully
Exceptions	1. If the user insert an already taken username 2. If the user press the cancel button 3. Internet connection isn't working An error message is shown and the flow of events starts again from point 3

Log In

Use Case	Log In
Actor	Policy Maker, Farmer, Agronomist
Entry condition	User wants to log in the system
Flow of events	1. User opens DREAM app 2. User puts its own username and password 3. System correctly log in the user and redirects it the home page
Exit condition	The user correctly log in the system
Exceptions	1. If the user inserts a wrong username and password 2. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

- **Policy Makers**

View Initiatives Reports

Use Case	View Initiatives Reports
Actor	Policy Maker
Entry condition	User wants to check the Steering Initiative proposed by agronomists
Flow of events	1. User presses the "View Initiatives Reports" button 2. System retrieves the information 3. System sends to the user the steering initiative insert by the agronomist 4. System displays the data 5. User presses the back button and return to home page
Exit condition	User presses the back button
Exceptions	1. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

Check Soil Humidity Data

Use Case	Check Soil Humidity Data
Actor	Policy Maker
Entry condition	User wants to check the soil humidity data
Flow of events	1. User presses the "Check Sensors Data" button 2. User presess the soil humidity data button 3. System retrieves soil humidity data and send them back to the user 4. System displays the data 5. User presses the home button and return to home page
Exit condition	User presses the "Home" button
Exceptions	1. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

Check Water Irrigation Data

Use Case	Check Water Irrigation Data
Actor	Policy Maker
Entry condition	User wants to check the water irrigation data
Flow of events	1. User presses the "Check Sensors Data" button 2. User presses the water irrigation button 3. System retrieves water irrigation data and send them back to the user 4. System displays the data 5. User presses the home button and return to home page
Exit condition	User presses the "Home" button
Exceptions	1. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

View Farmers Ranking

Use Case	View Farmers Ranking
Actor	Policy Maker
Entry condition	User wants to see the farmers ranking
Flow of events	1. User presses the best or the farmers rank widget 2. System retrieves data on farmers ranking 3. System shows ranking data 4. User presses back button and return to home page
Exit condition	User press the "Back" button
Exceptions	1. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

View Specific Farmer Information (View Farmers Ranking)

Use Case	View Specific Farmer information (View Farmers Ranking)
Actor	Policy Maker
Entry condition	User wants to see the specific information of a farmer
Flow of events	1. First three events are the same of the "View Farmers Ranking" use case 2. User presses on a farmer name 3. System retrieves farmer data 4. System displays farmer data 5. User presses back button to return on the rank page
Exit condition	User presses "Back" button
Exceptions	

- **Farmers**

Profile Edit

Use Case	Profile Edit
Actor	Farmers
Entry condition	User wants to modify or update its profile
Flow of events	1. User presses the "Setting" button on its home page 2. User modifies the interest data (username, password, crop type or area) 3. User presses "Confirm" button 4. System applies the change 5. User receives a confirmation message 6. User is redirect to hom epage
Exit condition	User receives a confirmation message
Exceptions	1. User choose an already taken username 2. User choose the old password An error message is shown and the flow of events starts again from point 2

Insert Production Data

Use Case	Insert Production Data
Actor	Farmers
Entry condition	User wants to insert the production data
Flow of events	1. User presses the "Insert Production Data" button 2. User fills the interested fields 3. User press "Confirm" button 4. System collect analiyze and store the data 5. System shows a confirmation message to the user and redirects it to the home page
Exit condition	User receives a confirmation message
Exceptions	1. User insert data in a wrong manner 2. User press "Back" button An error message is shown and the flow of events starts again from point 2

Check News

Use Case	Check News
Actor	Farmers
Entry condition	User wants to check the news
Flow of events	1. User press the news box 2. The system retrieve data regarding news 3. System displays the news 4. User presses the back button
Exit condition	User presses "Back" button
Exceptions	1. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

Post in the Forum

Use Case	Post in the Forum
Actor	Farmers
Entry condition	User wants to post in the forum
Flow of events	1. User presses "Forum" button 2. User chooses a thread to reply in or the creation of new thread 3. User writes the text 4. User presses the "Post" button 5. System shows a confirmation message and redirects the user to the "Forum" page 6. User presses "Back" button and returns to home page
Exit condition	User presses "Back" button
Exceptions	1. User exceed the maximum number of character 2. Internet connection isn't working An error message is shown and the flow of events starts again from point 2

Check Weather Forecast

Use Case	Check Weather Forecast
Actor	Farmers
Entry condition	User wants to check weather forecast
Flow of events	1. User presses on the weather forecast widget 2. System retirves weather data 3. System displays data 4. User presses the home button and returns to home page
Exit condition	User presses "Home" button
Exceptions	1. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

Send an Help Request

Use Case	Send an Help Request
Actor	Farmers
Entry condition	User needs the agronomist help
Flow of events	1. User presses "Contact Agronomist" button 2. User chooses the object of its help request 3. User writes the message 4. User sends the request by pressing "Send request" button 5. System shows a confirmation message and redirects the user to the home page
Exit condition	The system shows a confirmation message to the user
Exceptions	1. User doesn't fill every mandatory boxes 2. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

- **Agronomist**

Confirm Daily Plan

Use Case	Confirm Daily Plan
Actor	Agronomist
Entry condition	User wants to confirm its daily plan
Flow of events	1. User presses the daily "Daily Plan" button 2. User chooses the farmers that he want to visit during the day 3. User checks if the farmers has at least two visit 4. User confirms the plan with the "Confirm" button 5. System displays a confirmation button and redirects the user to the home page
Exit condition	System displays a confirmation message
Exceptions	1. At the end of the year a farmer has less than two visit 2. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

Update Daily Plan

Use Case	Update Daily Plan
Actor	Agronomist
Entry condition	User wants to update its daily plan
Flow of events	1. User presses the "Daily Plan" button 2. User presses the "Update Daily Plan" button 3. User writes its deviations 4. User add or remove farmers according the daily plan 5. User confirms the plan with the "Confirm" button 6. System diplays a confirmation message and redirects the user to the home page
Exit condition	System displays a confirmation message
Exceptions	1. User doesn't fill every mandatory fields 2. Internet connection isn't working An error message is shown and the flow of events starts again from point 3

Manage Farmers Request

Use Case	Manage Farmers Request
Actor	Agronomist
Entry condition	User wants to reply to a farmer help request
Flow of events	1. System displays the notification of an incoming message 2. User presses the inbox icon 3. User inserts the object 4. User replys the request 5. User sends the message by pressing the "Send" button 6. System displays a confirmation message and redirects the user to the inbox page
Exit condition	System displays the user a confirmation message
Exceptions	1. Object field is empty 2. Message field is empty 3. User press the "Cancel" button 4. Internet connection isn't working An error message is shown and the flow of events starts again from point 2

Insert responsibility Area

Use Case	Insert responsibility Area
Actor	Agronomist
Entry condition	User wants to update its own coverage area
Flow of events	1. User presses the settings button 2. User selects the new responsibility area 3. User presses the "Confirm" button 4. System displays a confirmation message and redirects the user to the home page
Exit condition	System displays a confirmation message
Exceptions	1. User press the "Cancel" button 2. User select an already taken area 3. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

Check Weather Forecast

Use Case	Check Weather Forecast
Actor	Agronomist
Entry condition	User wants to check the weather forecast in its responasbility area
Flow of events	1. User presses the wheather widget 2. System retrieves and display data 3. System displays the data 4. User presses the home button to return to the home page
Exit condition	User presses "Home" button
Exceptions	1. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

View Farmers Area Ranking

Use Case	View Farmers Area Ranking
Actor	Agronomist
Entry condition	User wants to see the ranking of farmers in the area
Flow of events	1. User presses the ranking widget 2. System retrieves and displays the data 3. User presses the back button and returns to the home page
Exit condition	User presses "Home" button
Exceptions	1. Internet connection isn't working An error message is shown and the flow of events starts again from point 1

- **Use Cases to requirement mapping**

1)Sign Up	R3) If the IT device is equipped with GPS technology, then the system must suggest the users the area in which they are during the registration phase. In any case, the final decision will be up to the registering user. R2) The system shall allow users to be identified by a username of their choosing.
2)Log In	R18) The system shall allow users to be identified by a username of their choosing.
3)View Initiative Reports	R1) The system must allow only registered and logged-in users to use the app. R6) The system must allow Policy Makers to retrieve all agronomists' steering initiative rep orts uploaded to the database.
4)Check Soil Humidity Data	R1) The system must allow only registered and logged-in users to use the app. R5) The system must allow Policy Makers to see up-to-date statistic data, accordingly to the database, about water irrigation systems and soil humidity sensors.
5)Check Water Irrigation Data	R1) The system must allow only registered and logged-in users to use the app. R7) The system must allow Policy Makers to see up-to-date statistic data, accordingly to the database, about water irrigation systems and soil humidity sensors.
6)View Farmers Ranking	R1) The system must allow only registered and logged-in users to use the app. R7) The system must give Policy Makers up-to-date ranking of the best and worst performing farmers. R8) The system must use a fair and scientific score to rank the Farmers in order to represent the real situation.

7)View Farmers Ranking	R1) The system must allow only registered and logged-in users to use the app. R7) The system must give Policy Makers up-to-date ranking of the best and worst performing farmers.
8)Profile Edit	R1) The system must allow only registered and logged-in users to use the app.
9)Insert Production Data	R1) The system must allow only registered and logged-in users to use the app. R9) The system must let Farmers insert their production information every day. R4) The system must inform the user to try later if they are experimenting connectivity issues .
10)Check news	R1) The system must allow only registered and logged-in users to use the app. R20) The system must present news concerning crop only if relevant to the Farmer's own crop type.
11)Post in the Forum	R1) The system must allow only registered and logged-in users to use the app. R4) The system must inform the user to try later if they are experimenting connectivity issues . R11) The system must allow every user registered as Farmer to access the forum, to create new discussions and to post replies to already existing ones.
12)Check Weather Forecast	R1) The system must allow only registered and logged-in users to use the app. R19) The system must show weather forecasts relevant to the area concerning the Farmer or the Agronomist.
13)Send an Help Request	R1) The system must allow only registered and logged-in users to use the app. R10) The system must provide the contact of the agronomist appointed to the area of the Farmer requesting professional help. R4) The system must inform the user to try later if they are experimenting connectivity issues .

14)Confirm Daily Plan	R1) The system must allow only registered and logged-in users to use the app. R14) The system must suggest to the Agronomists which farms to visit (among the ones in their area of competence) while planning the Daily Plan based up on the last visit day following a FIFO policy. Exceptions to the FIFO policy are the farms underperforming, which shall have higher priority. R15) The system must suggest to the Agronomists only the farms among the ones in their competence area. R13) The system must allow the Agronomists to add a new schedule for the day each day. R16) The system must allow the Agronomists to update their schedule during the day and to confirm the execution before the end of the day. R17) The system must register the already uploaded schedule as definitive if the user doesn't confirm the daily plan for day x before 23:59 of the day x.
15)Update Daily Plan	R1) The system must allow only registered and logged-in users to use the app. R13) The system must allow the Agronomists to add a new schedule for the day each day. R16) The system must allow the Agronomists to update their schedule during the day and to confirm the execution before the end of the day. R17) The system must register the already uploaded schedule as definitive if the user doesn't confirm the daily plan for day x before 23:59 of the day x.
16)Manage Farmers Request	R1) The system must allow only registered and logged-in users to use the app. R12) The system must notify Agronomists of unresolved requests of help from Farmers.

17)Insert Responsibility Area	R1) The system must allow only registered and logged-in users to use the app. R2) The system must suggest to the users the area in which they are during the registration phase if the IT device is equipped with GPS technology. In any case, the final decision will be up to the registering user
18)Check Wheather Forecasat	R1) The system must allow only registered and logged-in users to use the app. R19) The system must show weather forecasts relevant to the area concerning the Farmer or the Agronomist.
19)View Farmers Area Ranking	R1) The system must allow only registered and logged-in users to use the app. R8) The system must use a fair and scientific score to rank the Farmers in order to represent the real situation. R18) The system must give Agronomists up-to-date ranking of the best and worst performing farmers in their responsibility area.

3.2.4 Sequence Diagrams

- **Shared Sequence Diagrams**

Sign Up

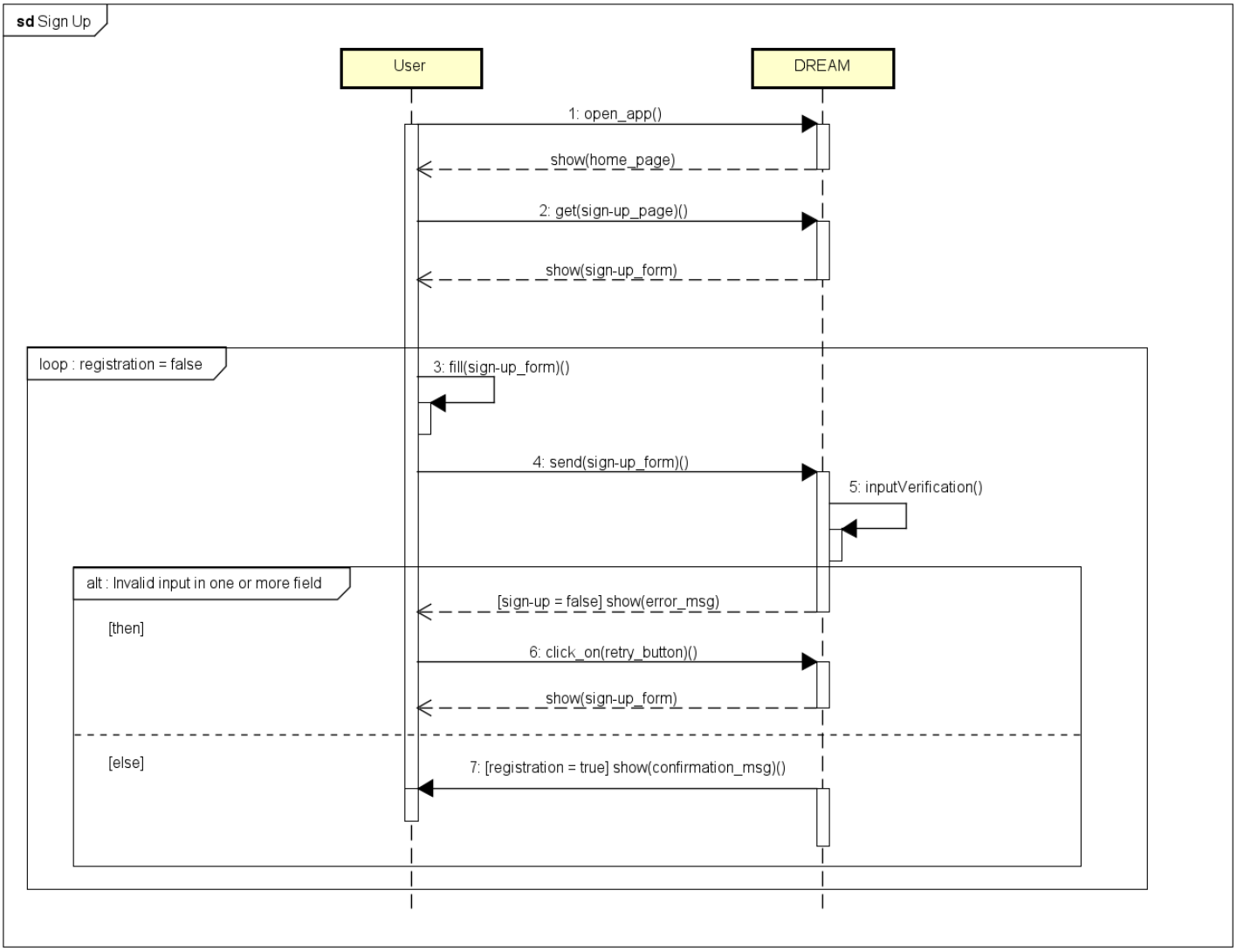

Figure 8: Shared Sequence Diagram - Sign Up Sequence Diagram

Log In

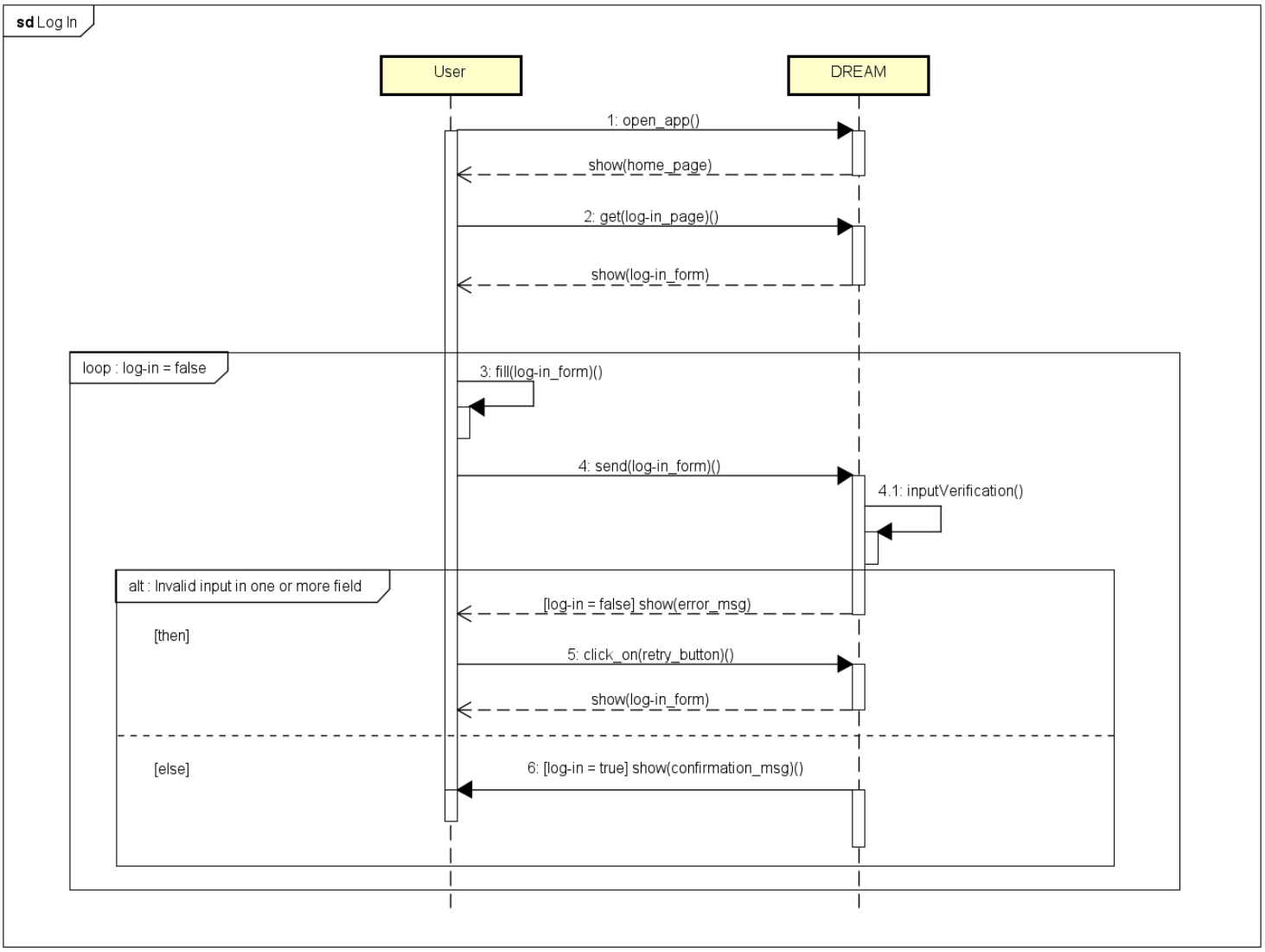

Figure 9: Shared Sequence Diagram - Log In Sequence Diagram

- **PolicyMakers**

  View Initiative Report Sequence

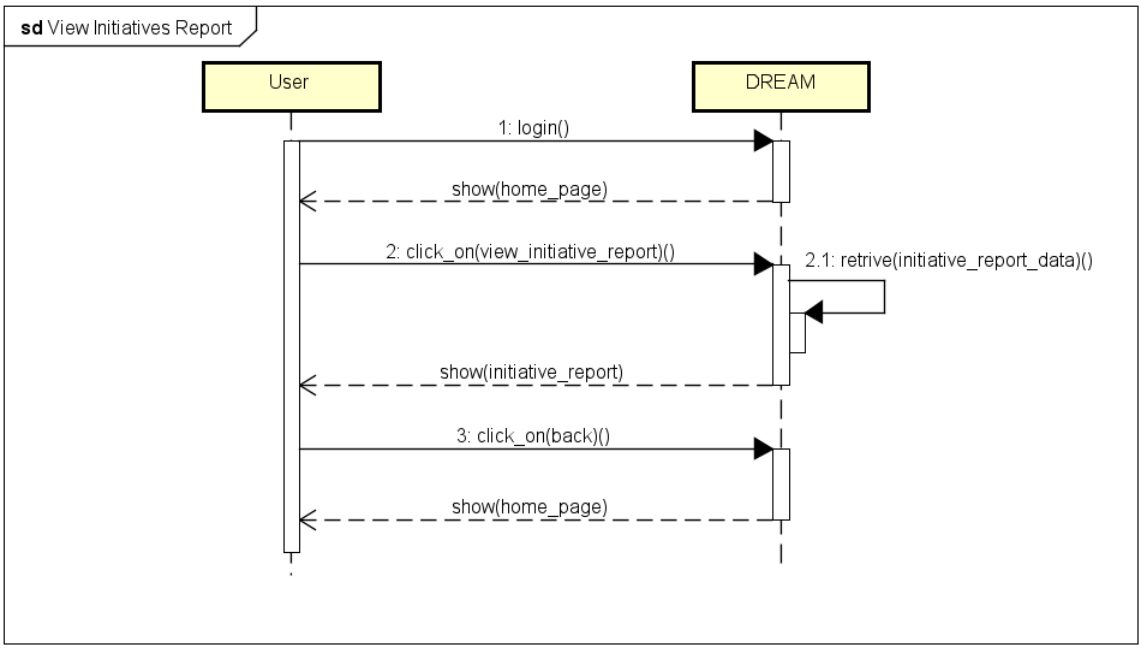

Figure 10: Policy Makers - View Initiative Report Sequence Diagram

Check Soil Humidity Data

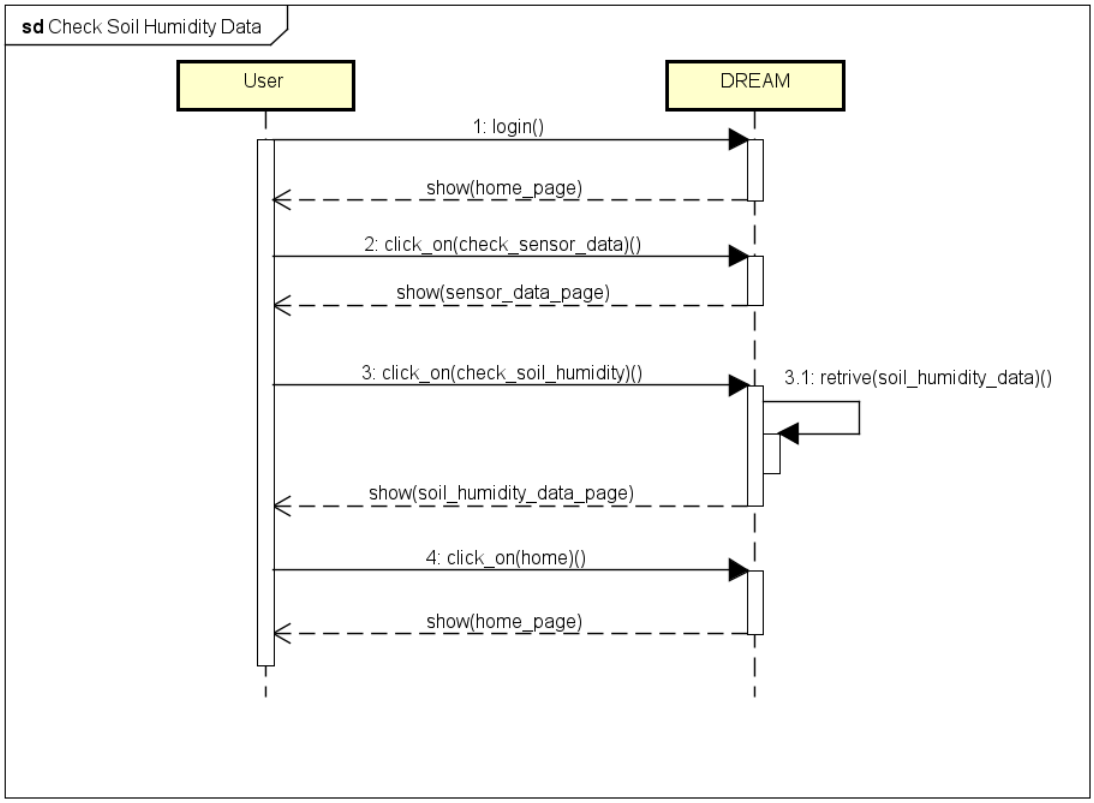

Figure 11: Policy Makers - Check Soil Humidity Data Sequence Diagram

Check Water Irrigation Data

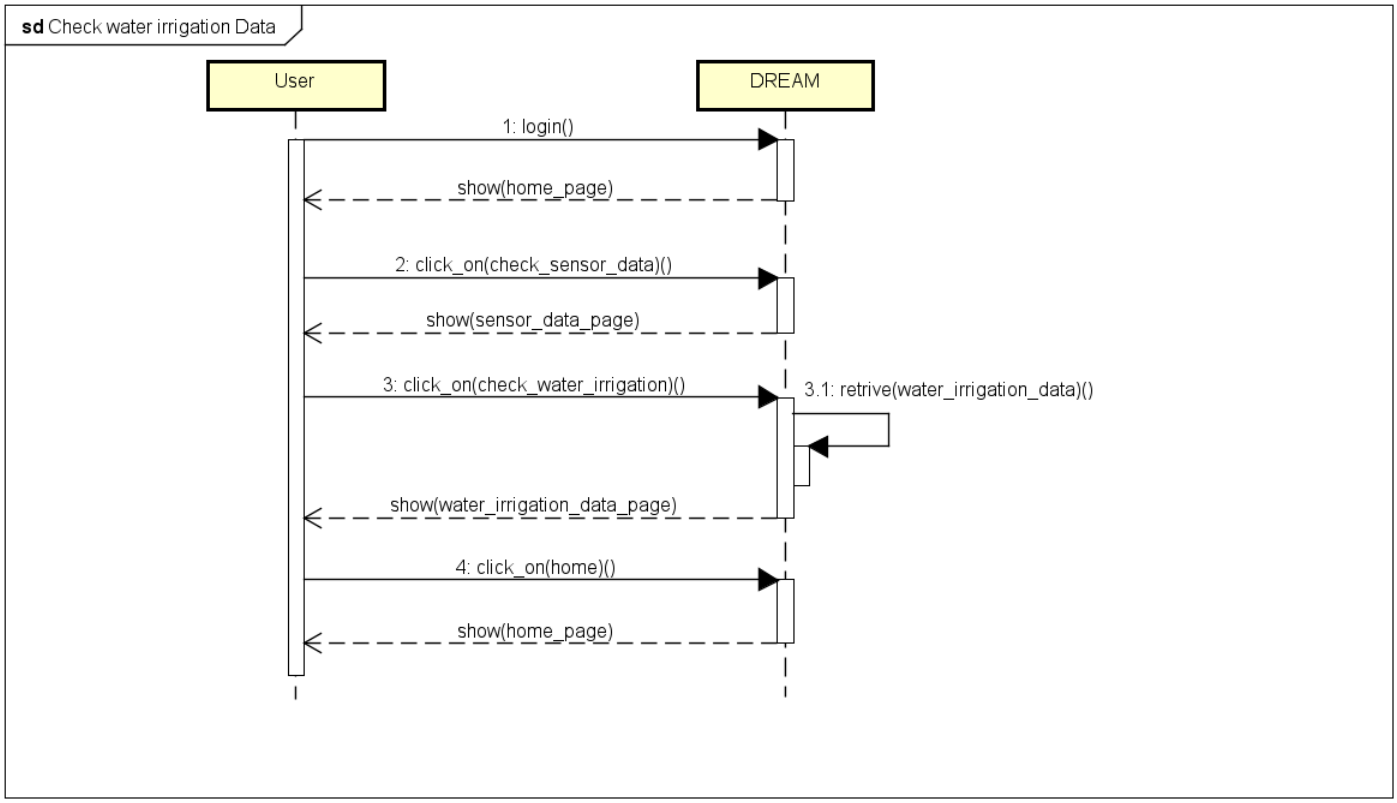

Figure 12: Policy Makers - Check Water Irrigation Data Sequence Diagram

View Farmers Ranking

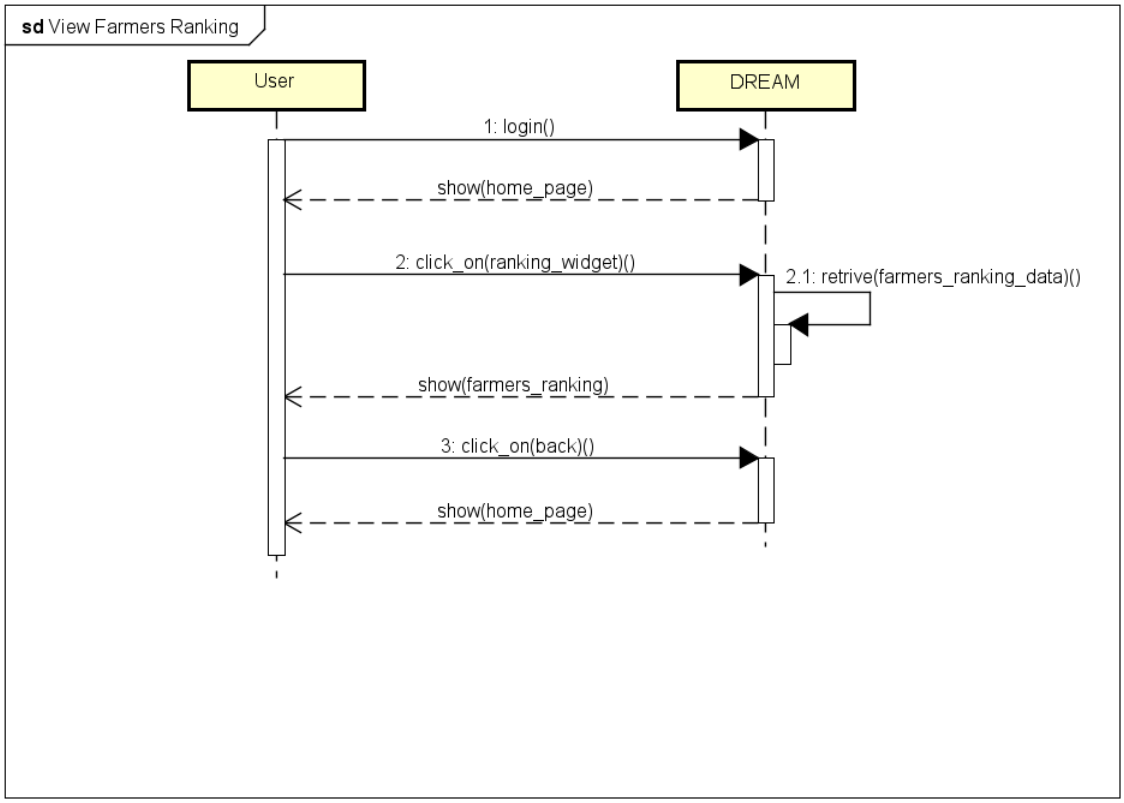

Figure 13: Policy Makers - View Farmers Ranking Sequence Diagram

View Specific Farmers information

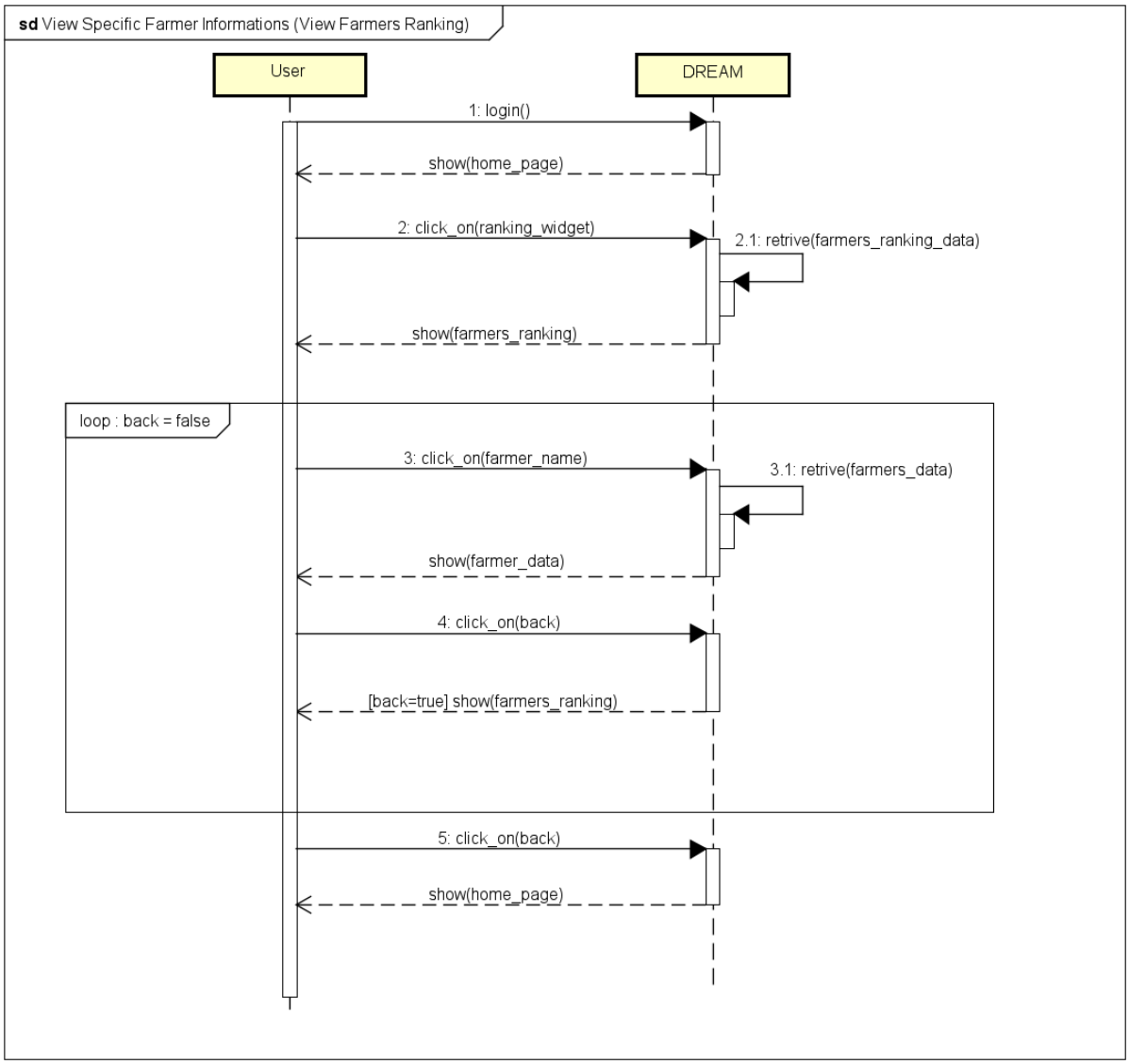

Figure 14: Policy Makers - View Specific Farmers information Sequence Diagram

- **Farmers**

Profile Edit

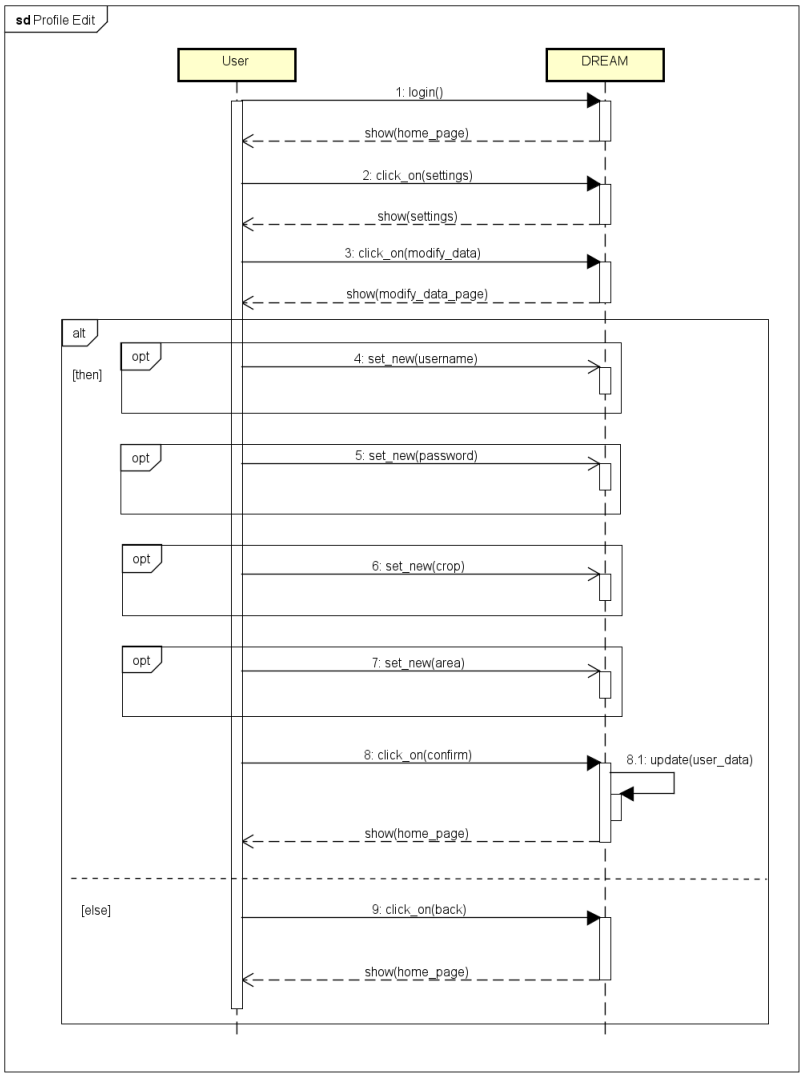

Figure 15: Farmers - Profile Edit Sequence Diagram

Insert Production Data

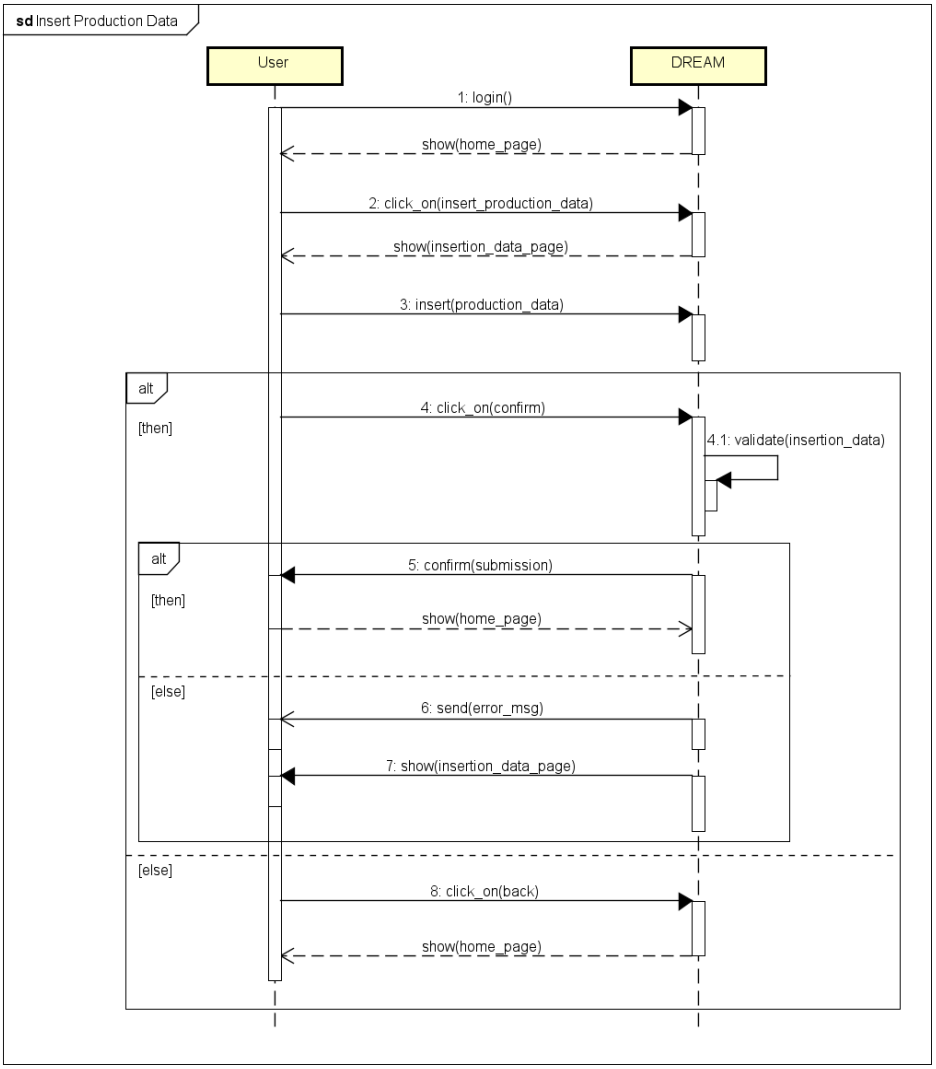

Figure 16: Farmers - Insert Production Data Sequence Diagram

Check News

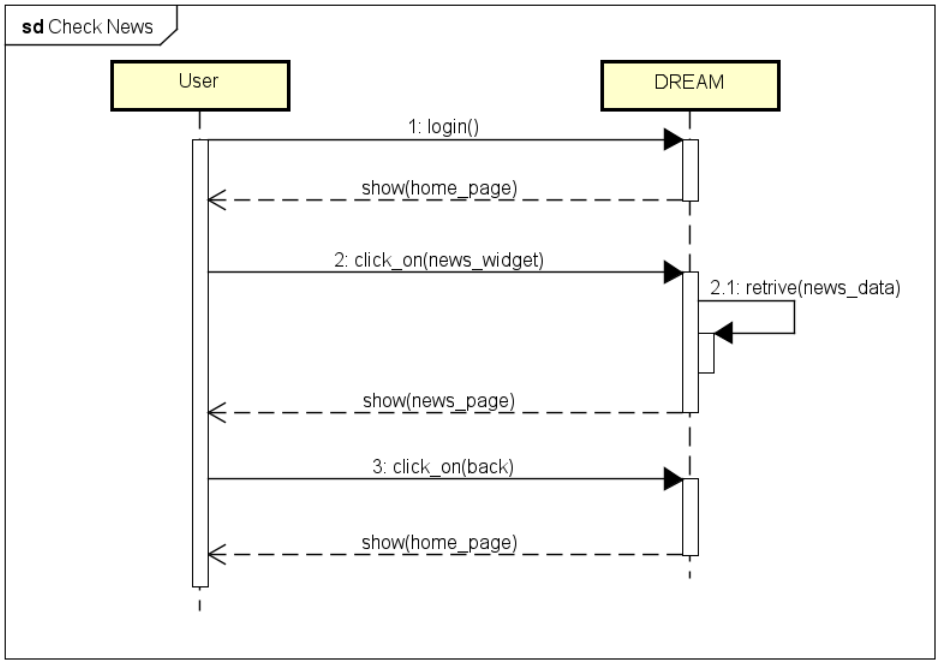

Figure 17: Farmers - Check News Sequence Diagram

Post in the Forum

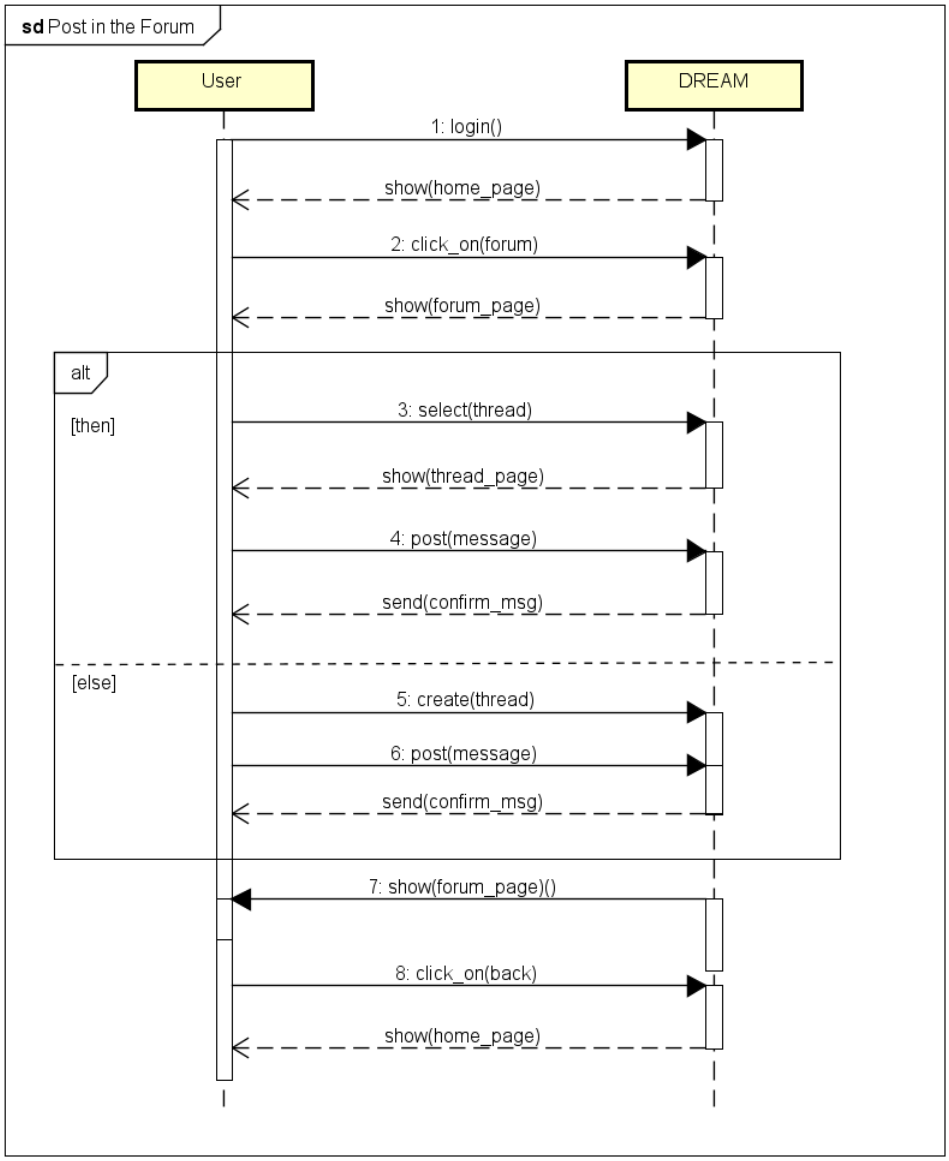

Figure 18: Farmers - Post in the Forum Sequence Diagram

Check Weather Forecast

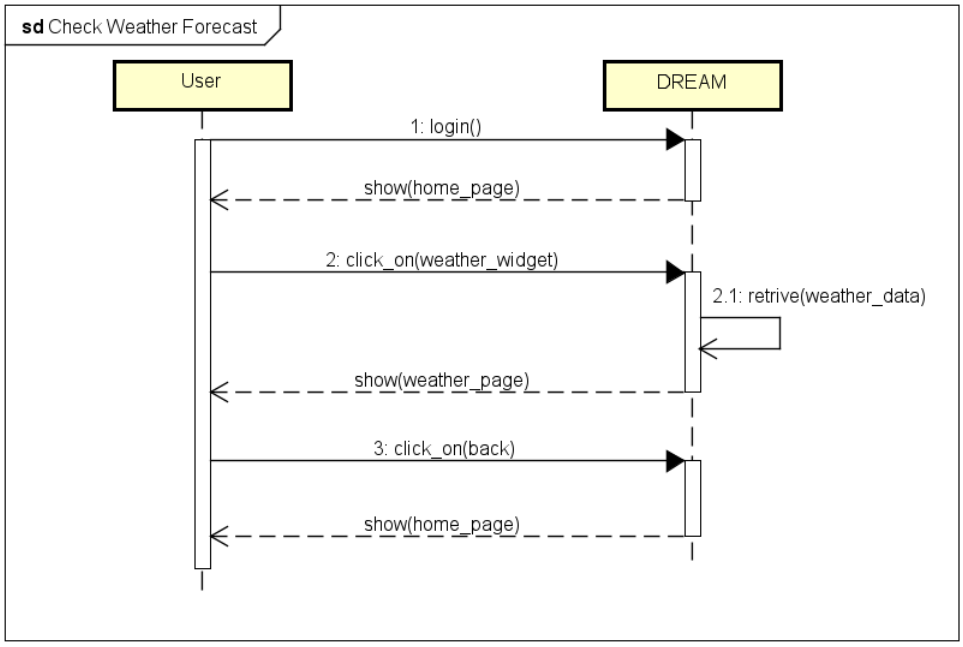

Figure 19: Farmers - Check Weather Forecast Sequence Diagram

Send Help Request

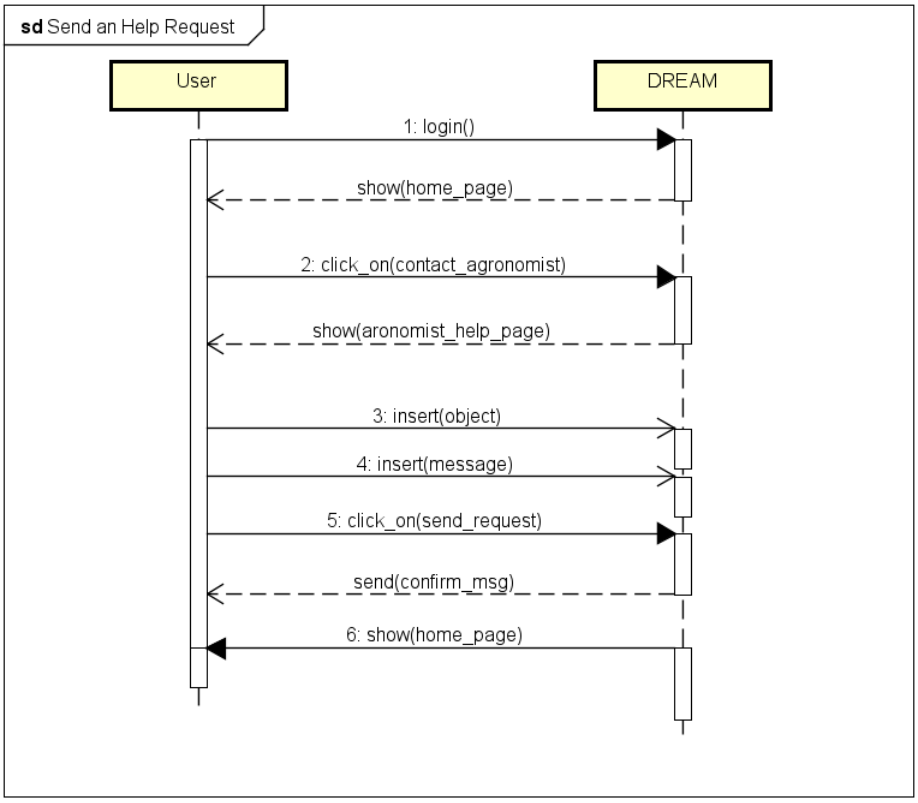

Figure 20: Farmers - Send Help Request Sequence Diagram

- **Agronomists**

  Confirm Daily Plan

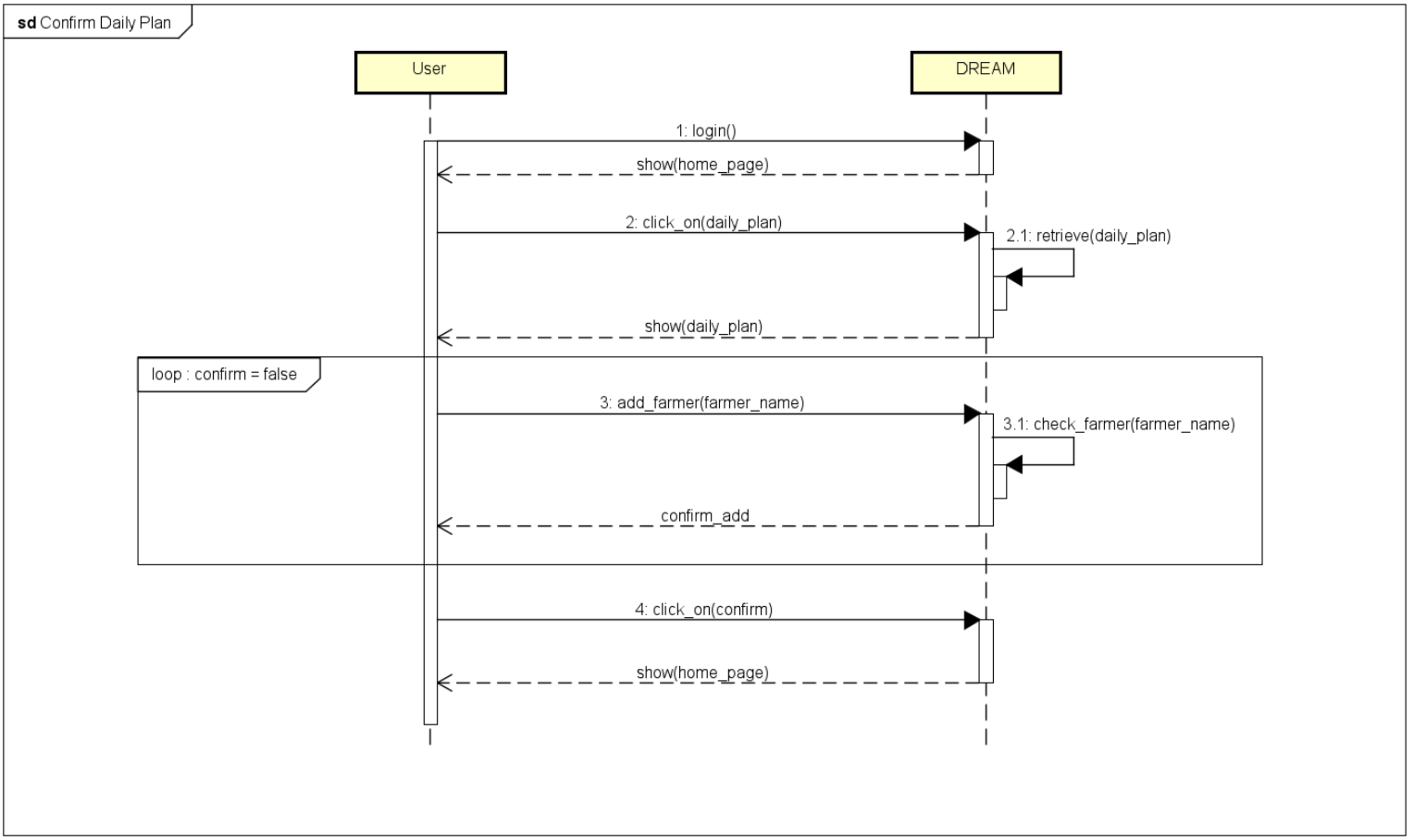

Figure 21: Agronomists - Confirm Daily Plan Sequence Diagram

Update Daily Plan

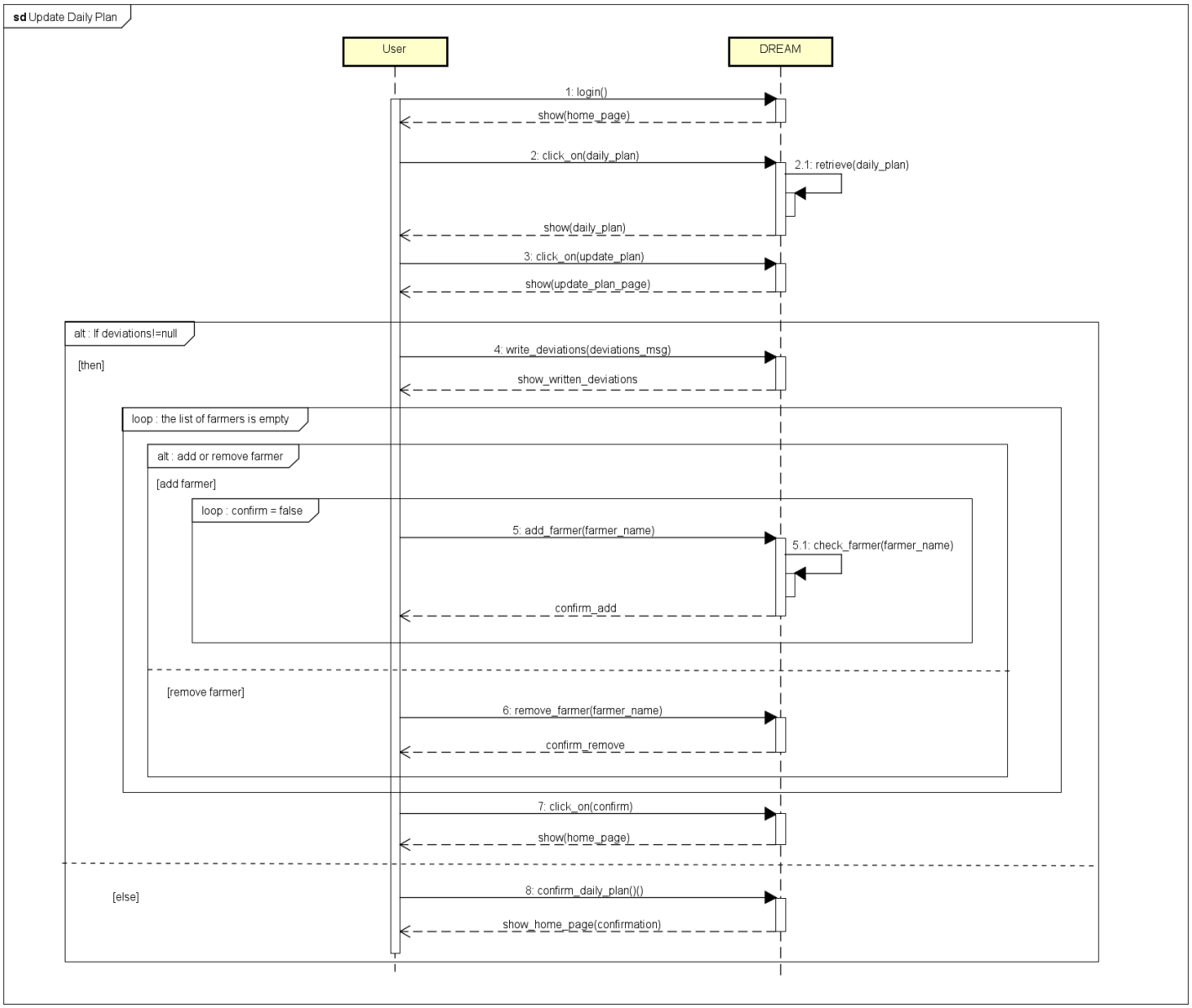

Figure 22: Agronomists - Update Daily Plan Sequence Diagram

Manage Farmers Request

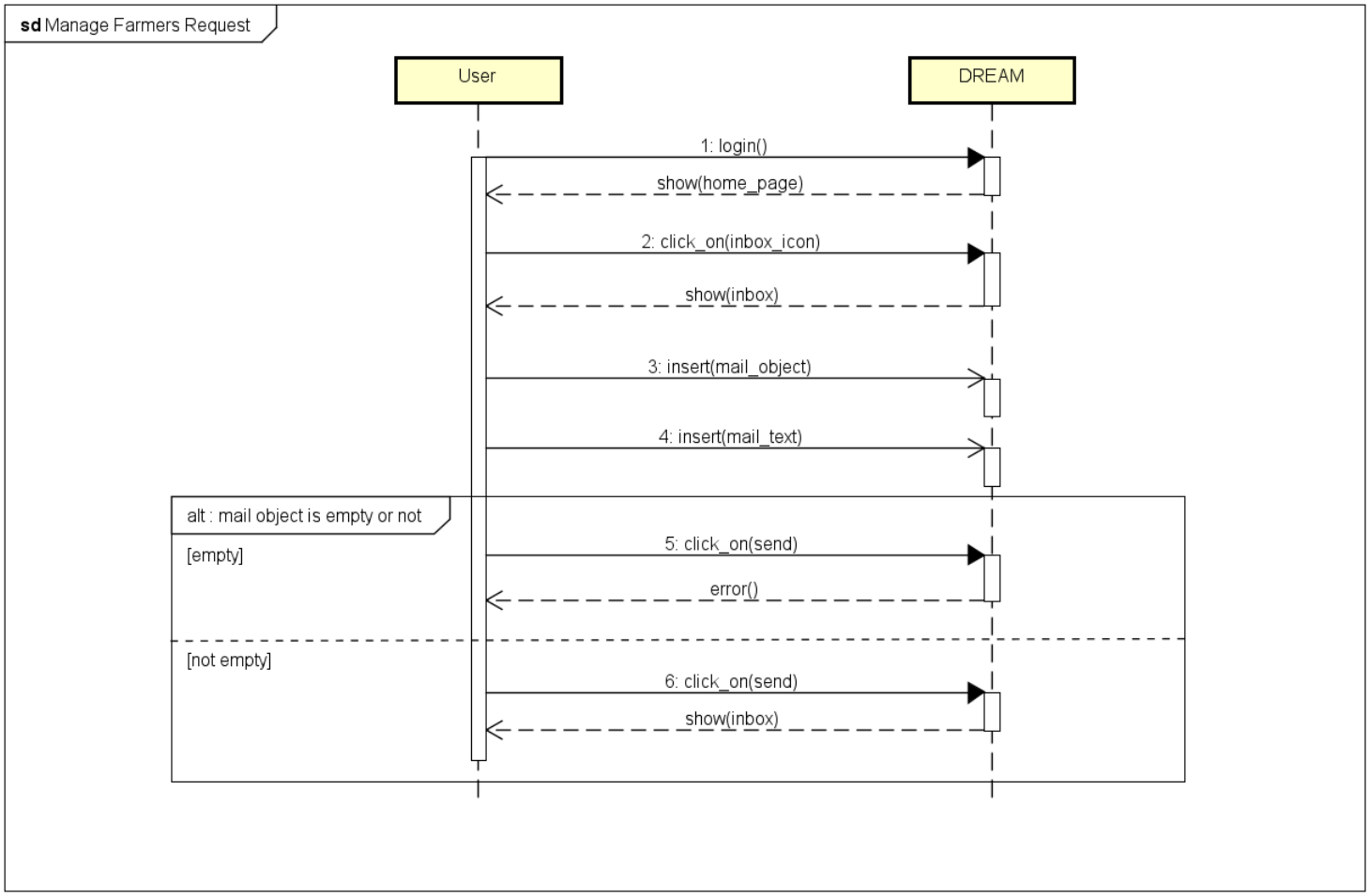

Figure 23: Agronomists - Manage Farmers Request Sequence Diagram

Insert Responsibility Area

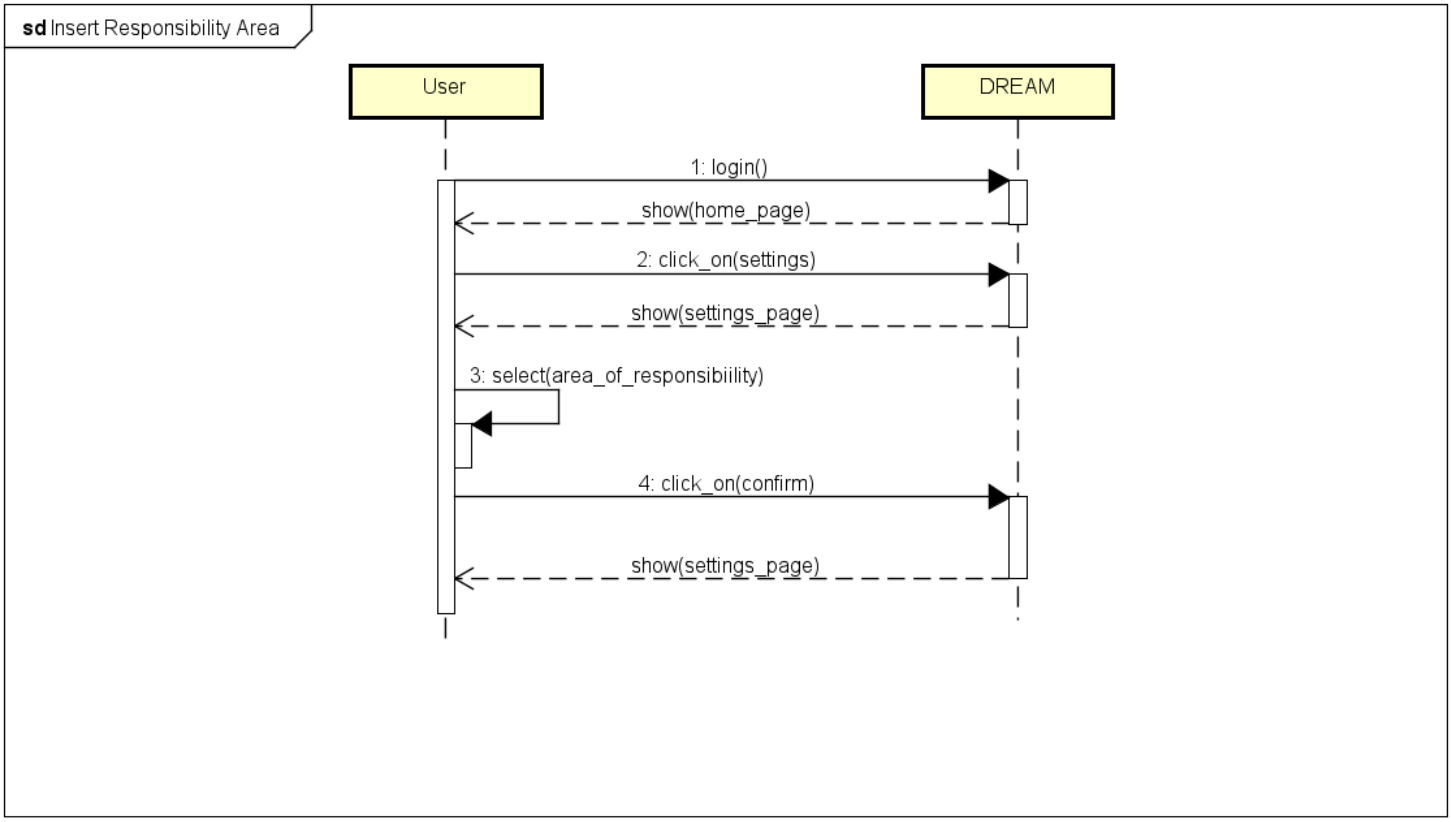

Figure 24: Agronomists - Insert responsibility Area Sequence Diagram

Check Weather Forecast

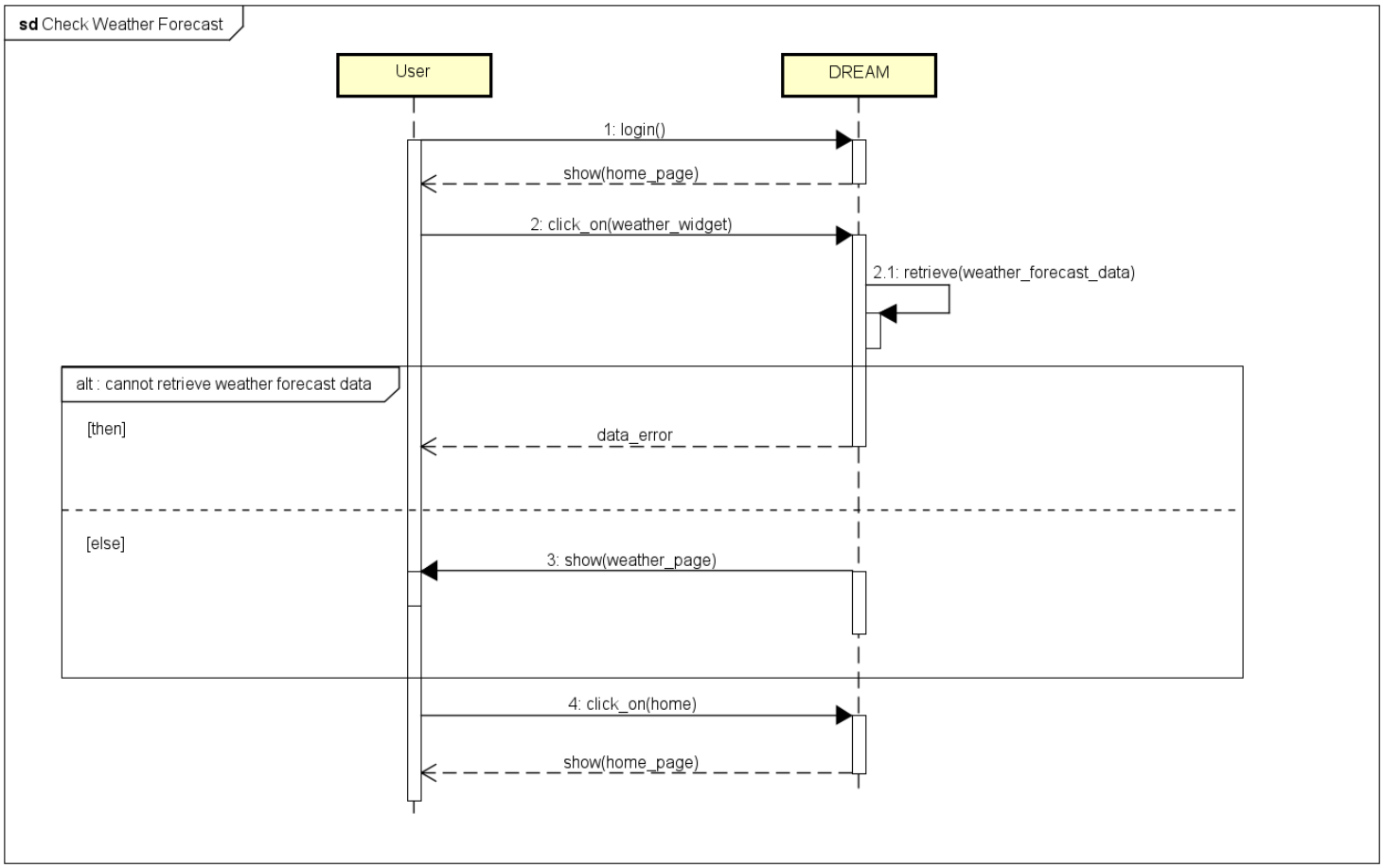

Figure 25: Agronomists - Check Weather Forecast Sequence Diagram

View Farmers Area Ranking

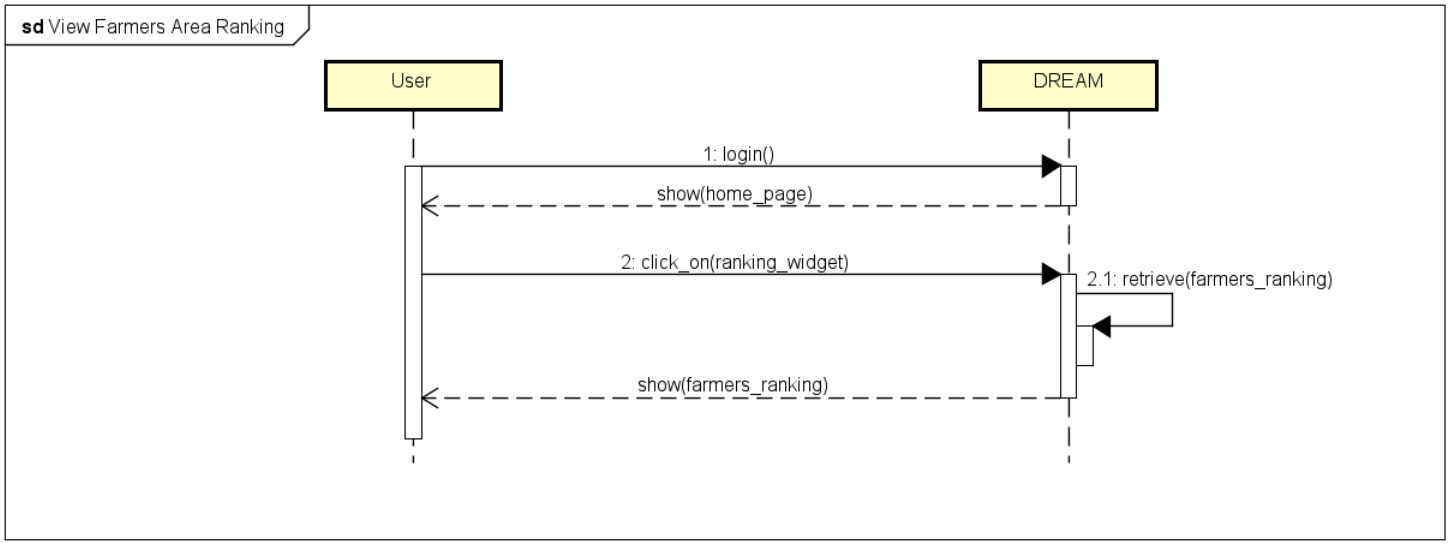

Figure 26: Agronomists - View Farmers Area Ranking Sequence Diagram

3.3 Performance Requirements

The majority of the user base will be represented by the farmers that cover the 58% [1] of the entire population of Telengana, while the number of policy-maker and agronomist users is irrelevant. According to Unique Identification Aadhar India, Telangana population in 2021 is estimated to be 39.9 million[2]. This means that the system is expected to be widely adopted with the registration of millions of users just in the first year. Also the average workload of the system is expected to be high, the goal is to guarantee the simultaneous connection of at least 2 millions of individuals. A mid-term goal could be to improve the scalability of the system in order to guarantee an efficient service to all users. Finally the system should have a good response time, less than 3 seconds is reasonable.

3.4 Design Constraints

3.4.1 Standards compliance

DREAM's user data must be treated respecting the law according to the PDPA[3].

3.4.2 Hardware limitations

The software is designed to enhance the portability, in fact it is enough to have a device with a stable internet connection. In order to use the application correctly it is required to have a device equipped with GPS.

3.5 Software System Attributes

3.5.1 Reliability

To ensure data consistency the system must have a fully backup infrastructure, this allows to recover from general failures in the main system. The four components shown in the *Figure(28)* operate in parallel, this guarantees high reliability and the correct functioning of the overall system in the case of a single component failure. Finally all the components of the system must be characterized by the highest MTTF and by the lowest MTTR possible.

3.5.2 Availability

The system should be up almost 24/7 in order to allow users to use the application's services at any time. The peak of use is expected during the working hours, it means that it is better to do all the maintainance activities during the night hours.

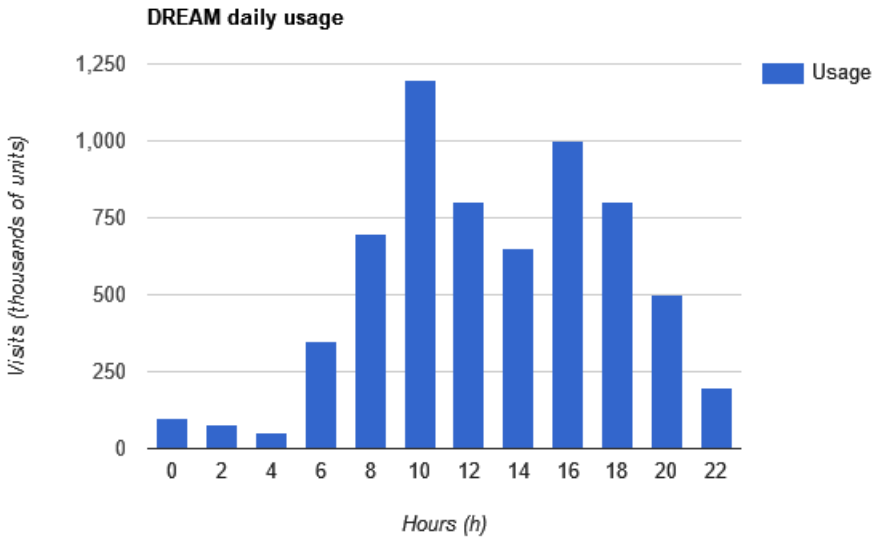

Figure 27: DREAM daily usage

Component	Availability	Downtime
Main System	99.999% (5-nines)	5 minutes/year
Government Website	99.9% (3-nines)	8.76 hours/year
Water Irrigation System	99.9% (3-nines)	8.76 hours/year
Humidity Sensors	99% (2-nines)	3.65 days/year
GPS	99.999% (5-nines)	5 minutes/year
Entire System	99.998%	10 minutes/year

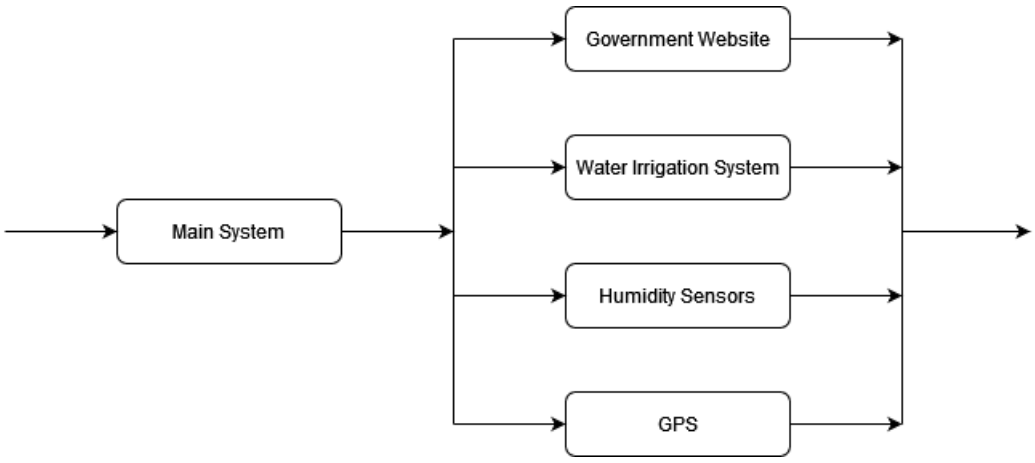

Figure 28: DREAM combined availability

3.5.3 Security

All the data stored by the system must be encrypted with a standard algorithm like the AES (Advanced Encryption Standard). The communication passes through a secure channel powered by the SSL protocol. All data should be stored in compliance with PDPA's regulation.

3.5.4 Maintainability

The maintainability is guaranteed by high standard of code, the use of design patterns will enhance this point and guarantee high reusability. The code must be extensively tested. The system must be extendible in order to support future updates. User's application will be regularly updated.

3.5.5 Portability

The system is designed to work on different operating systems like Windows, MacOS and Linux. All main browsers are supported.

4 Formal Analysis using Alloy

4.1 Formal Analysis Purpose

The following analysis aims to formally prove the correctness of the system model by exploiting Alloy verification tool. To achieve so, we test the model by checking if some of the previous defined goals are met.

Specifically, we stress:

- **(G3) Visualize the result of steering initiatives**: it is modelled as an assertion that checks the ability of the system to retrieve all the reports written by the agronomists and present them to each policy maker.
- **(G4) Visualize relevant data for the farmer business**: the goal is checked by asserting that if and only if there's a news concerning some crop type x or a certain area y, then the system will present it to all the farmers growing crop type x or working in area y.
- **(G5) Keep track of the production**: thanks to the predicate, we can formally illustrate the scenario in which a farmer two entries of production are succesfully registered in his/her history production in app (refer to fig. 31).
- **(G6) Request for help/suggestions**: the aim here is to make sure that every farmer that files an help request, there will always be an agronomist (in particular, the one related to the same district where the farm is) that will reply. We show this through an assertion and a predicate that illustarte an instance of this messages' exchange (see fig. 29). Furthermore, we display that some news that are deemed as interesting for a farmer, are actually presented to him/her (refer to fig. 30).
- **(G7) Create and participate in discussions concerning the agricolture field**: through two assertions, we make sure that every farmer can create a new topic in the Forum as long as he/she belong to the system and can also join discussions already started by anyone else. Moreover, we make sure thanks to another assertion that each farmer posting something in a dicussion gets automatically linked to the same

forum topic. Finally, we show an example of the dynamics that could happen in a forum topic, as shown in fig. 33.

- **(G8) Receive requests of help all in one place**: we check that all the help requests filed by farmers end up in exactly one agronomist's inbox, that is the agronomist whose area of competence matches with the region to which the farmer belongs.
- **(G10) Easy daily planning procedure**: we show an instance of an agronomist's daily plan and we make sure that every farm planned to visit is in the same region of competence of the agronomist.

Plus, we runned some empty predicates `show` to generate some more general instances of the app system.

4.2 Alloy Code

```
one sig AppSystem{
users: some User,
externalData: some DataProvider,
forum: one Forum,
farmerRanking: one FarmerRanking
}{
        all r: FarmerRanking | r = farmerRanking
}

sig string{}
sig Email{}

abstract sig User{
userDevice : some SmartDevice,
email: one Email,
password: one string,
name: one string,
surname: one string,
}{
#userDevice > 0
}

sig PolicyMaker extends User{
reports: set Report,
statisticData: set DataProvider
}

sig Farmer extends User{
region: one District,
userFarm : one Farm,
productionData: disj set ProductionData,
helpRequests: disj set HelpRequest,
```

```
helpReplies: disj set HelpReply,
forumDiscussions: set ForumTopic,
relevantNews: disj set News,
weatherForecast: disj set WeatherForecast
}{
#helpRequests ≥ #helpReplies
}

sig Agronomist extends User{
district: one District,
dailyPlan : disj set DailyPlan,
dailyVisits: disj set Visit,
helpRequests: disj set HelpRequest,
helpReplies: disj set HelpReply,
reports: disj set Report
}{
#helpRequests ≥ #helpReplies
}

sig Farm{
owner: one Farmer,
position : one Location,
cropType: some Crop
}

abstract sig DataProvider{
dateOfMeasure: one Date,
location: one Location
}
sig SoilSensor extends DataProvider{}
sig IrrigationSystem extends DataProvider{}
sig WeatherForecast extends DataProvider{}

sig ProductionData{
farm: one Farm,
sownQty: one Int,
harvestedQty: one Int,
cropType: one Crop,
date: one Date
}{
harvestedQty ≥ 0
sownQty > 0
}

sig Crop{}

//set of farms that the Agronomist plan to visit for the date visitDate
sig DailyPlan{
farmsToVisit : disj some Farm,
visitDate : one Date,
insertionDate : one Date
}{
}

sig Report{
author: one Agronomist
}{
```

```
all a:Agronomist | this in a.reports iff author=a
}

//visits actually done by an Agronomist: can be the same foreseen by the
    ↪ dailyPlan or not
//because an Agronomist can make deviations on his/her dailyPlan (new farms,
    ↪  skipping farm, etc.)
sig Visit{
date: one Date,
farm: one Farm
}

one sig Forum{
topics: set ForumTopic
}
sig ForumTopic{
name: one string,
creator: one Farmer,
posts: disj some Post
}
sig Post{
user: one Farmer,
date: one Date
}

sig News{
cropType: set Crop,
area: set District
}

abstract sig Message{}
sig HelpRequest extends Message{
sender: one Farmer,
receiver: one Agronomist
}
sig HelpReply extends Message{
sender: one Agronomist,
receiver: one Farmer
}

one sig FarmerRanking{
entries: set RankingEntry
}
sig RankingEntry{
user: one Farmer,
score: one Float,
rank: one Int
}{
rank > 0
score.leftPart ≥ 0
}

//Geographical location (coordinates)
sig Location{
region: one District
}
//Political location
```

```
sig District{
manager: one Agronomist
}

sig Date{}

sig SmartDevice{
localizationActive : lone GPS
}
sig GPS{}

sig Float{
    leftPart: one Int,
    rightPart: one Int
}{
rightPart ≥ 0
}

//-------UTILITIES-----
fun isGreater[f1,f2: Float] : lone Float{
        {f: Float | (f=f1 iff (f1.leftPart > f2.leftPart
                                or (f1.leftPart=f2.leftPart and f1.rightPart
                                    ↪  > f2.rightPart)))
                        or (f=f2 iff (f1.leftPart < f2.leftPart
                                or (f1.leftPart = f2.leftPart and f1.
                                    ↪ rightPart < f2.rightPart)))}
}

//-------FACTS-------

//Since email will be the username to access the profile, email must be
    ↪ unique
fact emailUniqueness{
        no disj u1,u2: User | u1.email=u2.email
}

//A device cannot be owned by different users
fact deviceUniqueness{
        all disj u1,u2: User | all d: SmartDevice | d in u1.userDevice
            ↪ implies d not in u2.userDevice
}

fact allUserInAppSys{
        all u: User, sys: AppSystem | u in sys.users
}

fact oneAgronomistOneDistrict{
        no disj d1,d2: District | d1.manager=d2.manager
}

fact oneFarmOneFarmer{
        no disj f1, f2: Farmer | f1.userFarm=f2.userFarm
}

fact ownerFarm1{
         all farm: Farm | no disj f1, f2: Farmer |
                farm.owner = f1 and f2.userFarm = farm
```

```
}

fact ownerFarm2{
        all u: Farmer | no disj f1,f2: Farm |
                f1.owner = u and f2.owner = u
}

fact ownerOfProductionData{
        all p: ProductionData | one f: Farm, u: Farmer |
                p.farm = f and f.owner = u iff p in u.productionData
}

fact senderOfHelpRequestIsTheFarmer{
        all m: Message | one f: Farmer | (m in f.helpRequests implies m.
            ↪ sender=f)
                        and (m in f.helpReplies implies m.receiver=f)
}

fact senderOfHelpReplyIsAgronomist{
        all m: Message | one a: Agronomist | (m in a.helpRequests implies m.
            ↪ receiver=a)
                        and (m in a.helpReplies implies m.sender=a)
}

fact farmerSendMessageToAreaAgronomist{
        all f: Farmer, m: Message |  (m in f.helpRequests)
                        implies (m.sender).region = (m.receiver).district

}

fact farmerReceiveMessageFromAreaAgronomist{
        all f: Farmer, m: Message | one a: Agronomist | (m in f.helpReplies)
                iff ((m.sender).district = (m.receiver).region
                        and m in a.helpReplies)
}

fact{
        all m: HelpReply | one a: Agronomist | m.sender=a iff m in a.
            ↪ helpReplies
}

fact{
        all m: HelpRequest | one f: Farmer | m.sender=f iff m in f.
            ↪ helpRequests
}


fact agronomistMustRespond{
        all m: HelpRequest,  a: Agronomist | one f: Farmer, m1: HelpReply |
                (m1 in (a.helpReplies & f.helpReplies)
                and m1.sender = m.receiver and m1.receiver = m.sender)
                implies m in a.helpRequests
}

fact planToVisitOnlyInTheArea{
        all a: Agronomist | no dP: DailyPlan, f: Farm | a.dailyPlan=dP
                and (f in dP.farmsToVisit)
```

```
                and f.region≠a.district
}

fact allVisitedFarmsAreInAgroDistrict{
        all v: Visit | one a: Agronomist | v in a.dailyVisits
                and v.farm.position.region = a.district
}

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

//An Agronomist cannot plan two different daily plan for the same day
fact noDifferentDailyPlansWithSameVisitDate{
        all a: Agronomist | no disj dp1, dp2: DailyPlan |
                (dp1 + dp2) in a.dailyPlan
                        and dp1.visitDate = dp2.visitDate
}


fact allVisitOfAgroAreInSameArea{
        no disj v1,v2: Visit | all a: Agronomist | v1 in a.dailyVisits
                and v2 in a.dailyVisits
                and v1.farm.location.region≠v2.farm.location.region
                and v1.farm.location.region≠a.district
}

//All the farms planned by an Agronomist in his/her daily plan must be
//in the area of competence of Agronomist
fact allPlannedFarmInAgroArea{
        all f: Farm | all dP: DailyPlan, a: Agronomist | dP in a.dailyPlan
                and f in dP.farmsToVisit
                implies f.position.region = a.district
}

//An agronomist cannot visit the same farm more than once a day
fact noMoreThanOneVisitToTheSameFarmADay{
        all a: Agronomist, disj v1, v2: Visit | no f: Farm | v1 in a.
            ↪ dailyVisits
                and v2 in a.dailyVisits
                and f=v1.farm and f=v2.farm
}

//All the farms planned in a dailyPlan must be in the same Telangana's
    ↪ district
fact allFarmsInDpSameArea{
        no disj f1, f2: Farm | all dP: DailyPlan | f1 in dP.farmsToVisit
                and f2 in dP.farmsToVisit
                and f1.region≠f2.region
}

//production cropType is the same of the farm
fact prodCropTypeSameAsFarm{
        all f: Farmer | all prodData: ProductionData | f.productionData=
            ↪ prodData
                implies prodData.cropType in f.userFarm.cropType
}
```

```
//a news must have at least one of the two: a concerned cropType or a
    ↪ concerned area
fact newsType{
        all n: News | #n.cropType > 0 or #n.area > 0
}

//all relevant news to the farmer (same CropType or same Area) must be shown
    ↪  to him/her
fact ifNewsRelevantThenShow{
        all n: News, f: Farmer |
                ((f.userFarm.cropType & n.cropType)≠ none
                        or f.userFarm.position.region in n.area)
                iff n in f.relevantNews
}

//Ranking (precedences on harvested/sown)
fact eachFarmerInRanking{
        all r: FarmerRanking | all f: Farmer | f in r.entries.user
}

fact ranking{
        all disj r: FarmerRanking, e1,e2: RankingEntry |
                let max = isGreater[e1.score, e2.score] |
                        e1 in r.entries and e2 in r.entries
                        and ((e1.score = max iff e1.rank < e2.rank)
                        or (e2.score = max iff e1.rank > e2.rank))
}

fact oneRankOneFarmer{
        no disj r1, r2: RankingEntry | r1.user = r2.user
}


//Topics in Forum and discussions
fact allForumTopicInForum{
        all f: Forum, t: ForumTopic | t in f.topics
}

fact allPostsInForum{
        all p: Post | one ft: ForumTopic | p in ft.posts
}

//If a farmer is posting on a discussion, then he/she is joining it
fact ifFarmerPostThenJoinDiscussion{
        all u: Farmer, t: ForumTopic | some p: Post |
                (p.user=u and p in t.posts) iff t in u.forumDiscussions
}

fact{
        all u: Farmer | no disj t1, t2: ForumTopic, p1,p2: Post |
                p1.user = u and p2.user = u and p1 in t1.posts
                and p2 in t2.posts
                and t1 in u.forumDiscussions
                and t2 not in u.forumDiscussions
}
```

```
fact{
        all p: Post | no t: ForumTopic | p in t.posts
                and t not in p.user.forumDiscussions
}


fact topicCreatorsAlsoJoinDiscussion{
        all t: ForumTopic |   t in t.creator.forumDiscussions
}

fact forumTopicNameUniqueness{
        no disj t1,t2: ForumTopic | t1.name = t2.name
}

fact allSteeringInitiativesAreReceivedByPC{
        all pc: PolicyMaker | all r: Report | r in pc.reports
}

//External data facts
//Interaction between external data and the application
fact allExternalDataInSys{
        all d: DataProvider | one s: AppSystem | d in s.externalData
}
fact weatherRelevantToFarmer{
        all f: Farmer | one w: WeatherForecast |
                w in f.weatherForecast iff w.location.region = f.region
}

fact locationUniquenessForSoilData{
        no disj d1, d2: SoilSensor | d1.location = d2.location
                and d1.dateOfMeasure = d2.dateOfMeasure
}

fact locationUniquenessForWaterData{
        no disj d1, d2: IrrigationSystem | d1.location = d2.location
                and d1.dateOfMeasure = d2.dateOfMeasure
}

fact locationUniquenessForWeatherForecast{
        no disj d1,d2: WeatherForecast | d1.location = d2.location
                and d1.dateOfMeasure = d2.dateOfMeasure
}

fact statisticDataForPc{
        all d: DataProvider | all pc: PolicyMaker | d in pc.statisticData
}

fact weatherRelevantForFarmer{
        all u: Farmer, w: WeatherForecast |
                w in u.weatherForecast iff w.location = u.userFarm.position
}

fact locationUniquenessOfFarms{
        no disj f1, f2: Farm | f1.position = f2.position
}

fact justOneAgronomistRelatedToDistrict{
```

```
        all d: District | no disj a1,a2: Agronomist |
                a1.district = d and a2.district = d
}


//---------ASSERTIONS & PREDICATES--------

//G3. Visualize the results of steering initiatives
assert allSteeringInitiativesAreReceivedByPC{
        all r: Report, pc: PolicyMaker | #PolicyMaker > 0
                implies r in pc.reports
}

//G4. Visualize relevant data for the farmer business
assert relevantNewsForFarmer{
        all f: Farmer, n: News | n in f.relevantNews
                iff ((f.userFarm.cropType & n.cropType) ≠ none
                        or (f.userFarm.position.region in n.area))
}

//G5. Keep track of the production
pred farmerProdEntryInsertion[f: Farmer, p1,p2: ProductionData]{
        p1.cropType in f.userFarm.cropType
        and p2.cropType in f.userFarm.cropType
        and p1.farm = f.userFarm and p2.farm = f.userFarm
        and (p1.sownQty ≥ 0 or p1.harvestedQty ≥ 0)
        //p2 > 0 to show at least one interesting production entry
        and (p2.harvestedQty > 0 or p2.sownQty > 0)
        and p1≠p2 and p1.date≠p2.date
        and (p1 + p2) in f.productionData
}

//G6. Request for help/suggestions
pred requestHelpAndGetResponseP[f: Farmer, hm: HelpRequest, a: Agronomist,
    ↪ hp: HelpReply]{
        (hm in f.helpRequests) implies
                (hm in a.helpRequests
                and hp in (f.helpReplies & a.helpReplies)
                and hp.sender = a and hp.sender = hm.receiver
                and hp.receiver = f and hp.receiver = hm.sender
                and a.district=f.region)
}

assert requestHelpAndGetResponse{
        all f:Farmer, hm: HelpRequest | one a: Agronomist, hp: HelpReply |
                hm in f.helpRequests iff
                        (hm in a.helpRequests
                        and hp in (f.helpReplies & a.helpReplies)
                        and hp.sender = a and hp.sender = hm.receiver
                        and hp.receiver = f and hp.receiver = hm.sender
                        and a.district=f.region)
}

pred interestingNews[n1,n2: News, u: Farmer]{
        n1≠n2 and u.userFarm.cropType in n1.cropType
        and u.userFarm.position.region in n2.area
}
```

```
//G7. Create and participate in forum discussions
assert createADiscussion{
        all sys: AppSystem | all t: ForumTopic | t in sys.forum.topics
                implies (t.creator in sys.users)
}

assert joinADiscussion{
        all sys: AppSystem | all t: ForumTopic, p: Post |
                (t in sys.forum.topics and p in t.posts)
                implies (p.user in sys.users)
}

pred ForumDynamics[p1, p2: Post, ft: ForumTopic]{
        (p1 + p2) in ft.posts and p1.user ≠ p2.user
        and p1.user.forumDiscussions = ft
        and p2.user.forumDiscussions = ft
}

assert farmersLinkedToTopicIfPosted{
        all u: Farmer, t: ForumTopic | some p: Post | (p.user=u and p in t.
            ↪ posts)
                iff t in u.forumDiscussions
}

//G8. Receive Requests of help all in one place
assert eachHelpRequestDoneIsInAgronomistInbox{
        all hr: HelpRequest | one a: Agronomist | hr in a.helpRequests
                and hr.sender.region=a.district
}

//G10. Easy daily planning procedure
pred insertDailyPlans[a: Agronomist, dp1,dp2: DailyPlan]{
        (dp1 + dp2) in a.dailyPlan and dp1≠dp2
        and #dp1.farmsToVisit > 0
        and #dp2.farmsToVisit > 0
}

assert allFarmsInDailyPlanAreVisitableByAgro{
        all a: Agronomist | all f: Farm, dp: DailyPlan |
                (dp in a.dailyPlan and f in dp.farmsToVisit)
                implies f.position.region = a.district
}

pred show {}


//-----------RUN & CHECK -------

check allSteeringInitiativesAreReceivedByPC for 10
check relevantNewsForFarmer for 10
check createADiscussion for 10
check joinADiscussion for 10
check requestHelpAndGetResponse for 5
check eachHelpRequestDoneIsInAgronomistInbox for 10
check allFarmsInDailyPlanAreVisitableByAgro for 10
check farmersLinkedToTopicIfPosted for 10
```

```
run requestHelpAndGetResponseP for 5
run farmerProdEntryInsertion for 5
run interestingNews for 3
run ForumDynamics for 5

run show{
#ForumTopic = 2
#Farmer > 3
#Post > 3
#Farmer.forumDiscussions > 1
} for 5

run show{
#PolicyMaker > 1
#Report > 1
#Agronomist > 0
}


run show{
#WeatherForecast > 1
#Farmer > 1
#PolicyMaker > 0
#SoilSensor > 1
#IrrigationSystem > 1
} for 6
```

Here we present some of the instances satifying the predicates that have been generated by the Alloy Analyser.

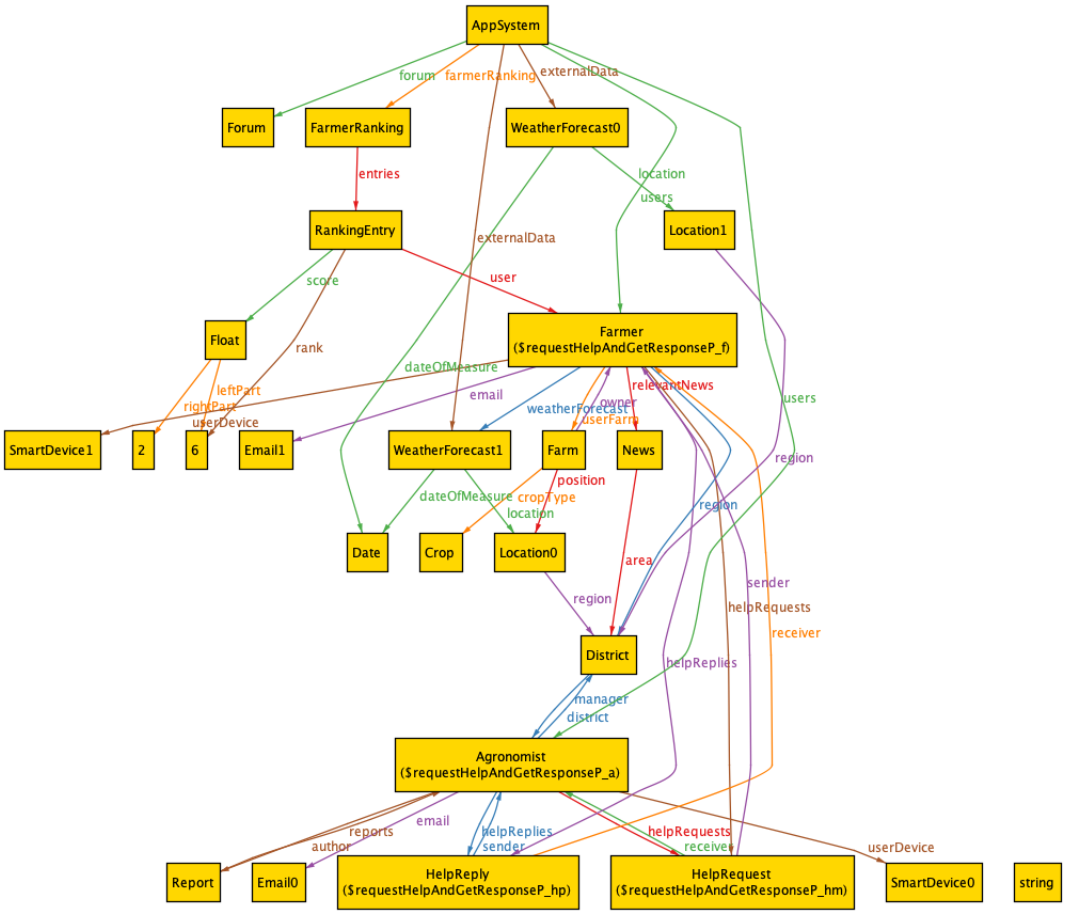

Figure 29: Predicate requestHelpAndGetResponseP

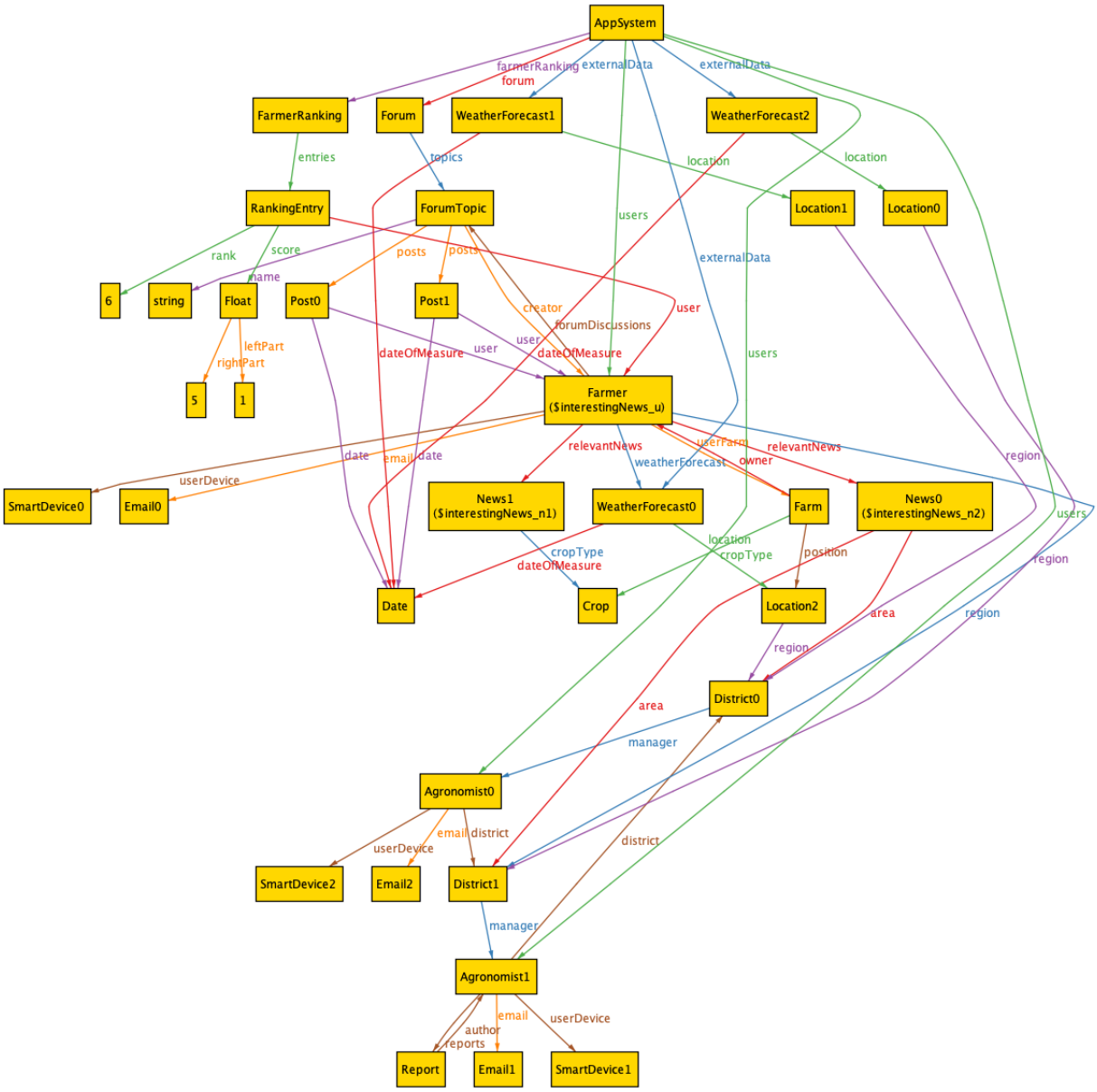

Figure 30: Predicate interestingNews

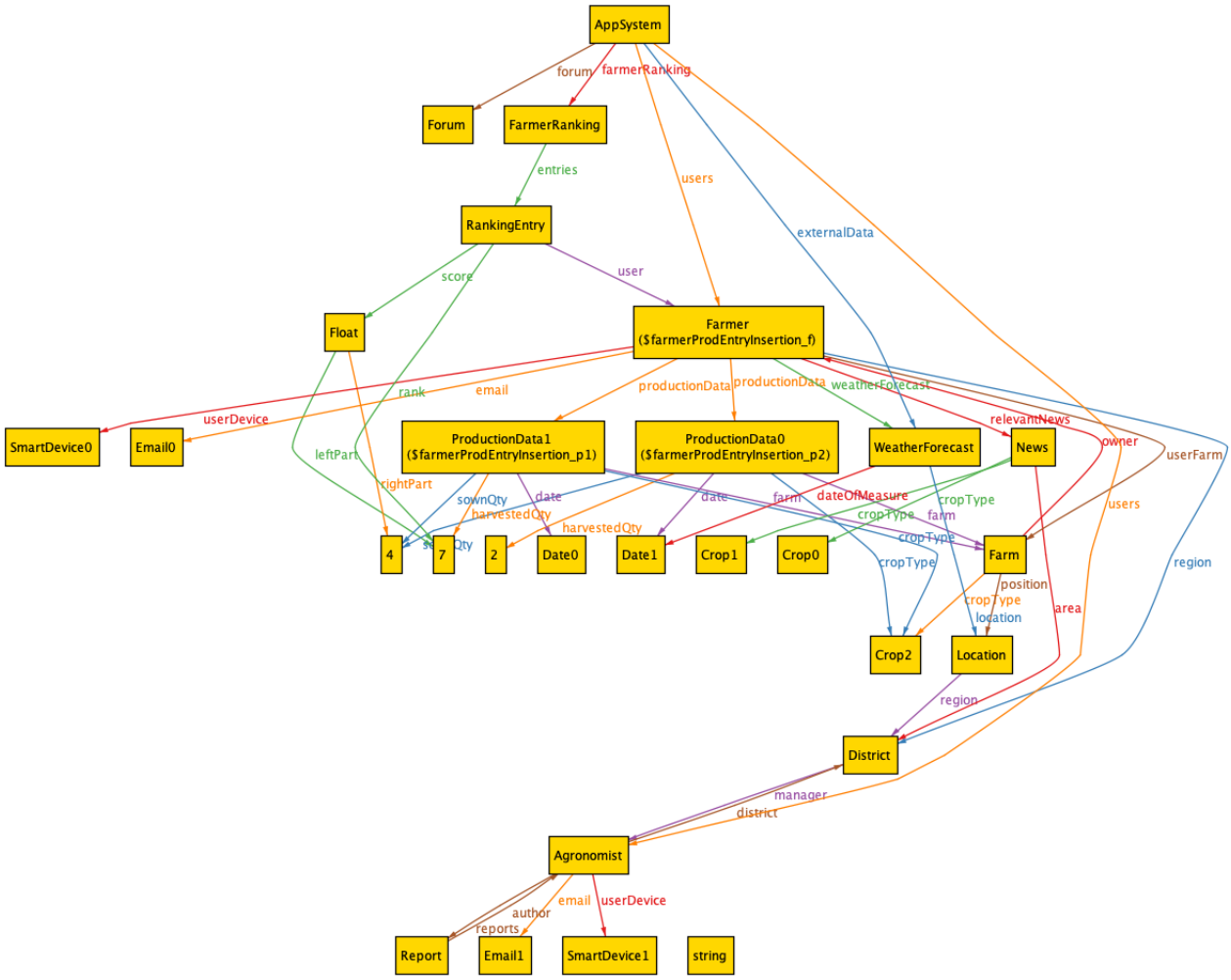

Figure 31: Predicate farmerProdEntryInsertion

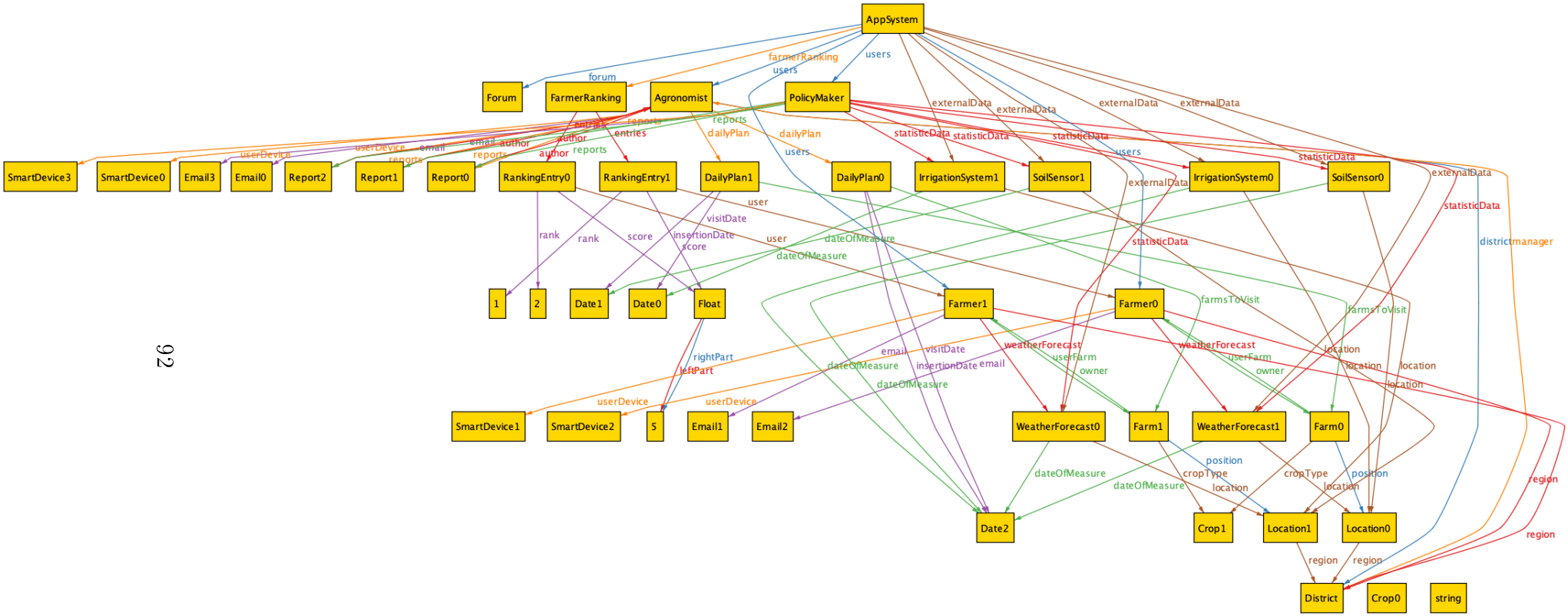

Figure 32: Predicate show focused on external statistic data

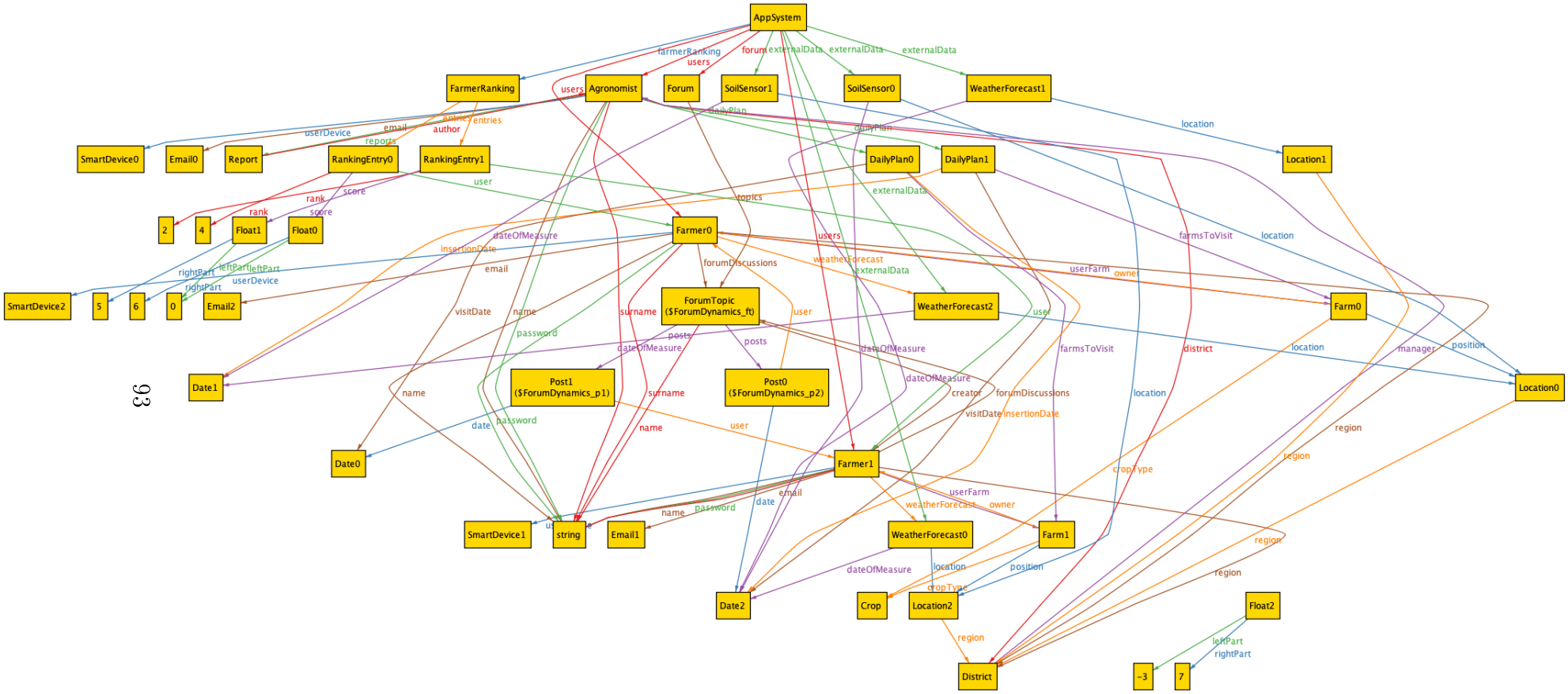

Figure 33: Predicate forumDynamics

5 Effort Spent

Student	Time for S.1	S.2	S.3	S.4
Ottavia Belotti	2.5h	5h	11h	10h
Alessio Braccini	3h	6h	15h	2h
Riccardo Izzo	3h	6h	15h	2h

6 References

References

[1] Software Engineering 2, *Requirement Engineering and Design Project: goal, schedule and rules*, A.Y. 2021-2022.

[2] `https://www.populationu.com/in/telangana-population`

[3] Indian Ministry of Electronincs and Informations Technology, `https://www.meity.gov.in/writereaddata/files/Personal_Data_Protection_Bill,2018.pdf`