POLITECNICO
MILANO 1863

Software Engineering II

A.Y. 2021/22

DREAM

Data-dRiven prEdictive fArMing

Requirements Analysis and Specification Document

Authors:
Mauro Famà
Giacomo Lombardo

December 23, 2021

Contents

1 Introduction

As world's population is increasing at steady pace, new challenges arise with its growth. Accordingly to a recent UN estimate, by 2050 globally there will be almost 10 billion people, and food demand is expected to increase between 59% to 98%. Furthermore, climate change is causing problems to the agriculture that are getting bigger every year: its effect is predicted to result in a 4%-26% loss by the end of the century in the Indian agricuture sector.
To tackle the situation, Telangana's state, which is the 11th biggest state in India, has decided to develop a platform to implement anticipatory, data-driven models to strengthen the policies in farming, with the ultimate goal of increasing the output of the agriculture sector.
This is the idea behind DREAM (Data-dRiven prEdictive fArMing), a platform where Telangana's Policy Makers, farmers and agronomists cooperate in order to enhance agriculture performances and facilitate the management of help requests, visits to farms and general communications betweens the actors in this scenario.

1.1 Purpose

The purpose of this document is to present a detailed description of the DREAM (Data-dRiven PrEdictive FArMing) platform. It will explain the purpose and features of the software, the interfaces of the software, what the software will do and the constraints under which it must operate. This document is intended for software users and also potential developers. The description will be offered by using models, providing scenarios and their relative use cases, analyzing most important functional and non-functional requirements.

1.2 Scope

The DREAM platform is intended for use by Telengana's policy makers, farmers and agronomists.
DREAM allows Telangana's policy makers, through the use of data provided by farmers regarding goods production, to extrapolate significative information and obtain a general view of the food production of the entire region. In this way, they will be given a platform to model and monitor the region's food system, facilitating the study and the realization of an adequate and well-performing long-term production strategy. The platform will provide policy makers the possibility to easily monitor farmers' performance and manage agronomists' interventions whenever and wherever needed. Furthermore, they will be able to identify those farmers whose production has been particularly good or bad, in order to reward the best performing farmers with incentives and to provide help to the farmers in need.
DREAM's goal is also to provide useful resources and tools to local farmers, who will be given a platform to facilitate the farmer-to-farmer and farmer-to-policy maker communication. Via DREAM, farmers will be able to participate to

forum discussions, easily create help requests and obtain sensitive information such as accurate weather forecasts and personalized suggestions regarding goods and agriculture routines.
For what concerns the local agronomists, DREAM facilitates the daily visit controls scheduling and regulates it accordingly to farmer needs, increasing the visits whether a farmer performance is significantly lower than the average.
In this document, the agronomist perspective won't be analyzed in detail, but it will be considered only marginally and in those cases where there is a significant interaction between agronomists and the other actors.

1.2.1 Goals

G1 Allow Telangana's Policy Makers to monitor local agriculture

G2 Allow farmers to receive assistance or rewards based on their performance

G3 Allow farmers to request and receive help

G4 Create a farmer's network to promote collaboration and mutual aid

G5 Improve farmers' performance and productivity

1.2.2 World and Machine Phenomena

World Phenomena	Description
WP1	Crop damage due to natural causes
WP2	Heavy meteorological conditions
WP3	Agriculture operations and routines
WP4	Agricuture product collection
WP5	Control visits to farmers

Shared Phenomena	Description	Controlled By
SP1	Insertion of crop collection data	W
SP2	Visualization of weather forecasts	M
SP3	Visualization of suggestions	M
SP4	Sending of notifications by TPMs	W
SP5	Receiving of notifications by farmers	M
SP6	Sending help requests	W
SP7	Contribution to a forum discussion	W
SP8	Visualization of crop collection data	M
SP9	Interview request	W
SP10	Agronomist visit request	W
SP11	Login/registration on the platform	W

1.3 Definitions, Acronyms, Abbreviations

- **DREAM**: Data-dRiven PrEdictive FArMing
- **G**: Goal
- **WP**: World Phenomena
- **SP**: Shared Phenomena
- **W**: World
- **M**: Machine
- **PM/TPM**: Policy Maker or Telangana Policy Maker
- **UI**: User Interface
- **API**: Application Programming Interface
- **DA**: Domain Assumption
- **R**: Requirements
- **UC**: Use Case
- **HTTPS**: HyperText Transfer Protocol over Secure Socket Layer
- **MQTT**: Message Queue Telemetry Transport
- **IoT**: Internet of Things
- **DBMS**: Database Management System

1.4 Revision history

- **Version 1.0** December 23, 2021
  First version of the complete RASD document

1.5 Softwares and tools used

- **Visual Studio Code** and **LaTeX** to model the document;
- **GitHub** for version control;
- **balsamiq** to realize the UI mockups;
- **Lucidchart** for diagrams;
- **Alloy Analyzer** for the Alloy analysis.

1.6 Document Structure

The document is presented as it follows:

1. **Introduction**: contains a brief overview of the problem, the proposed solution and the overall content of the document.

2. **Overall Description**: contains a general description of the system, focusing on functions, users and constraints.

3. **Specific Requirements**: contains a more specific description of the system requirements and describes in detail interfaces, use cases and system's attributes.

4. **Formal Analysis using Alloy**: contains an Alloy model representing the most critical aspects of the system.

5. **Effort Spent**: contains a brief recap of the time spent to complete this document.

2 Overall Description

2.1 Product Perspective

The system will be implemented from scratch, completely replacing the legacy system. The system will make use of some already existing functionalities, such as in weather forecasting where there is already an existing platform for collecting and visualizing data (https://www.tsdps.telangana.gov.in/aws.jsp).

2.1.1 Class diagram

The class diagram in 1 is a high-level representation of the system[3]. The main elements in the diagram are:

- **User**: a user of the DREAM platform. More specifically, a user can be:
  - **Farmer**
  - **Policy Maker**
- **Help Request**: represents a help request created by a farmer and taken in charge by a policy maker.
- **Forum**: represents the platform's forum, containing discussions grouped in threads.
- **Thread**: represents a thread of the forum.
- **Intervention**: represents an intervention in a forum discussion.
- **Terrain**: represents an area of property of a farmer destinated to agricultural use.
- **Sensor**: represents a generic sensor installed in a terrain to retrieve data such as irrigation or soil humidity.
- **Sensor Data**: high level astraction of the data retrieved by sensors and inserted in reports.
- **Product**: represents a generic agricultural product. Products can be subdivided in:
  - **Agricultural Products**: goods produced by farmers (e.g. vegetables).
  - **Support Products**: auxiliary products used in goods production (e.g. fertilizers).
- **Suggestion**: represents a suggestion related to a product.
- **Geographical Area**: represents a wide area containing multiple farmer's terrains. The Telangana territory is subdivided into areas to facilitate the workload division between policy makers.

- **Weather Forecasts**: represents a weather forecast for a specific date. Contains various informations such as weather conditions, sunrise and sunset times, precipitations, etc..
- **Report**: represents a report containing information about goods production, compiled by farmers at the end of the crop collecting phase.
- **Trimester**: represents a trimester of a year.

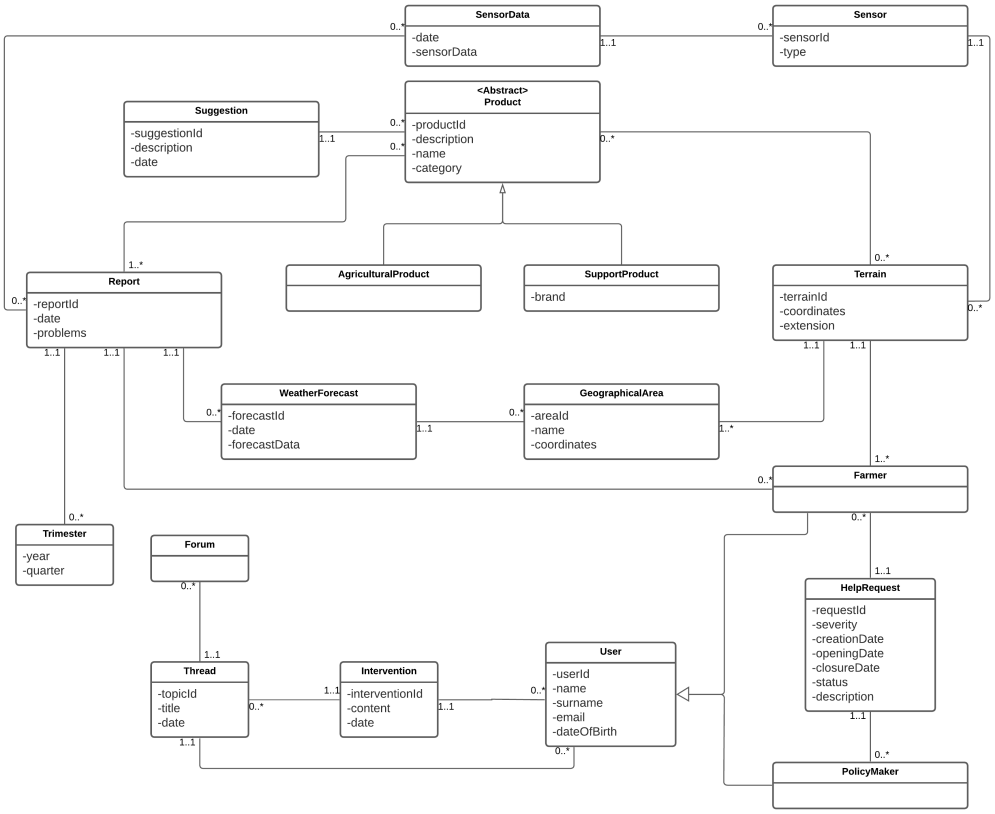

Figure 1: Class diagram

2.1.2 Activity and State diagrams

Activity diagrams describe main activities and their flow, highlighting the responsibilities of the various actors involved in the process. State diagrams describe the behaviour of the system while considering all possible states the objects can have when an event occurs. This analysis helps to clarify the most critical aspects of the system.

The activity diagram in Figure 2 represents the farmer registration pro-

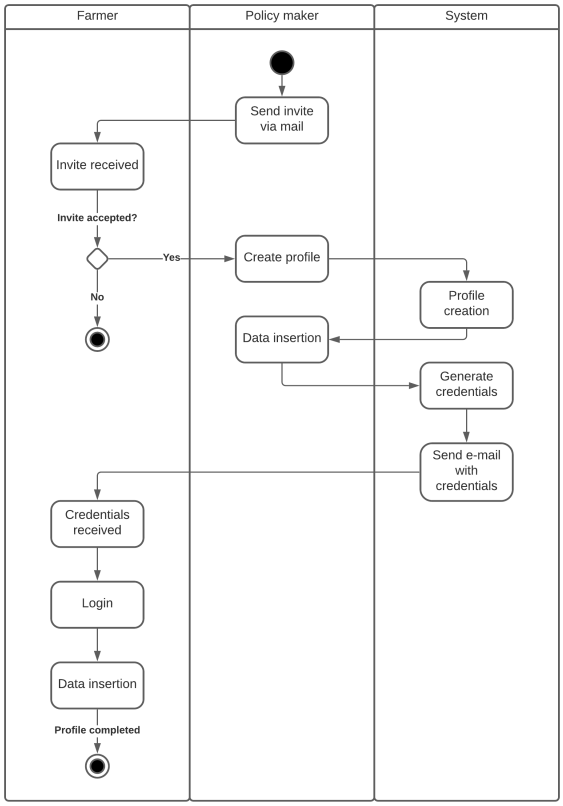

Figure 2: Activity diagram: Registration

cedure: after receiving an invite via e-mail, if the farmer wants to join the DREAM programme responds to the invitation, accepting it. The policy maker then proceeds to create the farmer's profile and to insert his data into the system. The system then generates the login credentials and an e-mail containing them and instructions to login is sent to the farmer. The farmer proceeds to login and to complete his profile, then the registration is complete.

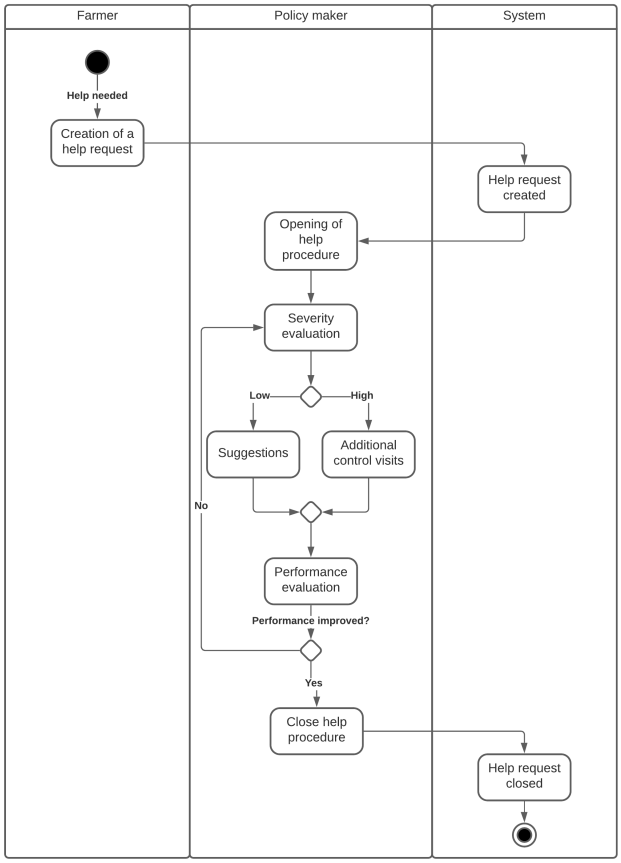

Figure 3: Activity diagram: Creation and handling of a help request

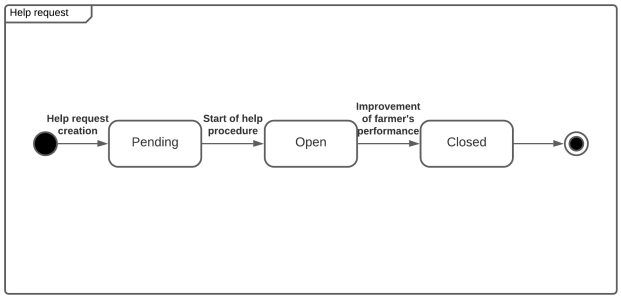

Figure 4: State diagram: Help request object

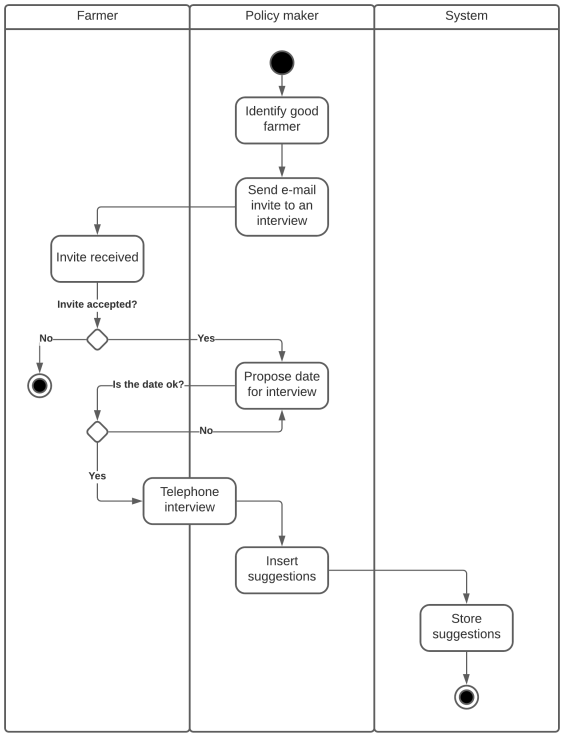

Figure 5: Activity diagram: Interview of a good-performing farmer

The activity diagram in Figure 3 represents the creation and handling of a help request. When a farmer needs help, creates a help request via the webapp. The system stores the request and when the policy maker visualizes it, he takes charge of the request. He then proceeds to evaluate the severity of the issue and to take the necessary actions. After that, the farmer's performance is monitored to verify whether the issue has been solved or not. If the performance improves, the help request is closed, if not, the request's severity is increased and new actions are taken.
The state diagram in Figure 4 represents the states of an instance of a help request object. More specifically, a help request is *Pending* when created by the farmer. When a Policy Maker takes charge of the request, it becomes *Open* and finally, when the problem is solved and the farmer's performances are back to normal, the request becomes *Closed*.

The state diagram in Figure 5 represents the interview of a good-performing farmer procedure: after identifying a good-performing farmer, the policy maker proceeds to send an e-mail to him/her, requesting if he/she is open to an interview. If the farmer agrees, a date is chosen and an interview is scheduled

for that date. The interview consists in a phone call where the farmer is asked about his methods and techniques. Afterwards, the policy maker inserts into the system suggestions for other farmers based on the information obtained in the interview.

2.2 Product functions

This section will summarize the main functionalities of the system.

2.2.1 Farmers registration

The procedure of registrating farmers into the system will be supervised by policy makers. Local farmers will not be able to register on their own but, if they intend to participate to the DREAM project, they will request and be provided with the credentials to login into the system. Policy makers will have to register the farmer's anagrafical informations and his possessions according to the local registries. Along with the registration in the system, farmers will be provided with sensors to facilitate the data gathering process: these sensors will be installed by an expert and they will automatically collect additional information about irrigation that will be included in periodical reports.

2.2.2 Agriculture monitoring

The system will provide Telangana Policy Makers a platform to monitor and manage the local agriculture system. Policy makers will be able to visualize relevant data about goods produced by farmers and will be able to evaluate their performance based on various parameters. The data will be both inserted manually by farmers (e.g. data regarding the quantity of goods produced) and automatically collected from sensors (e.g. data regarding weather conditions, soil humidity and irrigation). The system will automatically process the data to provide policy makers a human-friendly interface where relevant information will be displayed. The UI will display to TPMs both information obtained through data aggregation regarding the overall state of local's agriculture and individual information about farmers performance. In this way, TPMs will be able to identify farmers whose performance has been particularly good or bad and act on them.

2.2.3 Farmer's assistance

The system will provide Telangana's local farmers an easy-to-use platform to obtain relevant information and to favor the interaction with TPMs and other farmers. Farmers will be able to visualize accurate weather forecasts and personalized suggestions regarding the whole goods production process. They will also be able to participate in forum discussions with other farmers, favoring collaboration and exchange of knowledge useful to the collectivity. DREAM will also provide to farmers an easy way to create help requests and send them to TPMs, in order to facilitate and speed-up the support process and favour interventions where needed. Through DREAM, farmers' performance will be monitored by policy makers, and they will be provided assistance if needed. Also, high-performing farmers will be rewarded with incentives such as prizes, insurances and invitations to excellence programmes.

2.2.4 Goods production data insertion and analysis

In order to monitor and visualize information about the local agricuture system, local farmers participating to the DREAM project will be requested to provide the system data regarding their production. Goods production data will then be analyzed and will provide TPMs useful informations. The collected data will regard:

- Quantity of goods produced;
- Quantity of products used during the process (e.g. fertilizers);
- Quantity of water used in irrigation;
- Weather conditions;
- Soil humidity

The system will then process the collected data, calculating parameters used to evaluate farmers' performance. These parameters include:

- Quantities produced based on hectares used;
- Seeds quantity;
- Amount of water used based on weather conditions;
- Resilience to bad weather;
- Ambiental impact of used products;
- Crop rotation;

2.2.5 Scenarios

- *Registration to the DREAM platform*
  Santosh is a Telangana's local farmer. He receives a letter from the Ministry of Agriculture, Food and Forestry of Telangana containing information about the DREAM project. Santosh thinks that he could benefit from the participation to the programme, so he follows the instructions of the letter and sends a application to participate to the DREAM project via e-mail to the Ministry. A few days letter his application is accepted and a policy maker sends him via e-mail further instructions: he will be provided with credentials to login into the system and an operator will visit his farm to install sensors to monitor irrigation. In the meantime the policy maker inserts into the DREAM database the information about Santosh. When Santosh receives the login credentials, he is ready to access to the system and start participating to the project.

- *Insertion of crop collection data into the system.*
  Rajesh is an experienced farmer who owns some land in the Telangana region. He was very excited to access the agriculture development program offered by the government, and made sure to get access to the DREAM platform as quickly as possible.
  For Rajesh, harvest time has finally arrived, his rice fields have yielded well, and after several days dedicated to harvesting the entire crop, it is time to report the results to the Policy Maker assigned to his area. Rajesh takes himself to the platform's web application, logs in, and, after consulting some data provided in his dashboard, gets ready to fill in the report form. First of all he selects what kind of harvest he has collected from the drop-down menu, then he enters the quantities of interest required (inlcuding auxiliary products), writes a complete and detailed description inserting also some inconveniences he has faced. After reviewing the data entered, he clicks on the buttons that retrieve the data from the sensors in the period of interest, waits about a minute before the procedure is completed and is ready to submit his report.

- *Awarding of the best performing farmer*
  Guepequesh works as PM at the Ministry of Agriculture, Food and Forestry of Telengana. Like every end of the quarter, it's time to designate who the top performing farmers of the period were. Guepequesh logs in with his credentials to the DREAM portal and goes to his area of expertise, the dashboard shows the ranking of farmers sorted according to the synthetic indices of interest established by the Committee of Policy Makers of Telengana, updated the same day. Guepequesh selects the top 10 farmers shown in the ranking, sends them an email alert stating that they have been selected as the top performing farmers of the quarter, specifying their position in the ranking. The notice contains instructions on how to claim the prizes they have won, and also indicates the dates available for an interview in which the PM will ask the farmer to provide information about the best practices he uses to grow his crops. After notifying all winners, Guepequesh will notify the system to proceed with the closing of the quarter. In the following days, following the interviews, the PM will enter all data related to the best practices followed by the farmer in the system, and they will be forwarded to all farmers in the area who grow the same type of product.

- *Farmers request for help and suggestions by other farmers*
  Vinay is a young farmer from Telengana. He has recently decided to strike out on his own in the world of agriculture and knows that he will face many challenges. During a normal day's work, Vinay notices that even several hours after irrigation, his soil remains very wet and even puddles form. He had never encountered this problem before and decides to rely on the farmers' forum provided by the DREAM platform to which he is subscribed. Vinay logs in and goes to the "Problems and Help Requests" section of the forum, creates a new thread and prepares to

fill in the required fields for the submission of the request. The young farmer enters a short title, his working area, the type of shoot with which he encountered the problem and an accurate description of the problem he had. All farmers receive a notification on the platform that a new application has been entered, the notification includes the title and type of sprout so that a farmer can immediately determine if he or she considers themselves competent in the field or not. In particular, an experienced farmer named Amrit pauses to read the full description of the problem and enters a very accurate response to the problem. All of the farmers who read the response enter positive feedback, and Vinay, the next day, feels confident in following Amrit's advice, given the many positive feedbacks.

- *Identify those farmers who need to be helped*
  Om Ashgabash is a Telangana Policy maker. At the end of the quarter, he checks as usual the performances of farmers. He notices that Vidit Parkamush, a local farmer of his area of competence, has produced goods for a significantly lower amount than the other farmers over the same periods. He then decides that Vidit might need some help in order to perform better in the next period. To help him, first of all he schedules a call with Vidit to understand his situation and evaluate which actions might be needed. Once understood the situation, Om decides first of all to provide Vidit some useful suggestions related to his goods production. Then, he adds to the local control visits schedule additional visits to Vidit's farm in order to keep under better control his situation for the next months. Vidit is also added to the monitoring list, where policy makers can easily check the performances of farmers in difficulty.

2.3 User Characteristics

In this section the different types of users of the system will be analyzed. DREAM has three categories of users:

- **Farmers**
  Telangana Farmers will mainly use the platform to visualize weather forecasts, insert data about goods production, create help requests and intervene in forum discussions. It is safe to assume that most of them will not have much familiarity with this kind of procedures and with technological products in general. This requires a UI as simple and intuitive as possible, providing them all the desired functionalities without requiring much experience with the system.

- **Policy Makers**
  Telangana Policy Makers will use the platform to monitor and manage the local agriculture. They will perform more complex tasks than farmers, such as visualizing data about performance, interact with farmers and manage help requests and suggestions and agronomists' schedules. We can assume that Policy Makers will have at least some experience with

the use of management software, and they will be provided more complex functionalities that will require some experience and training with the platform.

- **DREAM system admins**
  The DREAM platform will also require administration operations from qualified operators. They will have a robust knowledge of the platform since they will perform system's maintenance and update. They will be designed with this role before the deployment of the system and it won't be possible to register to the platform with this role.

2.4 Constraints

This section contains general observations about boundaries that will limit the system options.

2.4.1 Regulatory policies

The email addresses provided for registration will not be used for commercial purposes or given to third parties.
The phone interviews will be recorded: permission to record the interview will be asked to the farmer.

2.4.2 Hardware limitations

The system will have support a broad list of devices since the end users of the product will be likely to have basic computers with low performances.

2.4.3 Interfaces to other applications

DREAM will implement a pre-existing weather forecasting system, which will be integrated via APIs. Failures in the system above will not affect the overall functioning of DREAM but will result in wrong forecasts or in the unavailability of the feature.

2.5 Domain assumptions

Domain assumptions define the world in which DREAM works.

DA1 Each policy maker has one and one only area to monitor and each area is monitored by one and one only policy maker.

DA2 Terrains cannot be shared: each terrain is associate with one and only farmer.

DA3 Farmers have a device able to access the platform.

DA4 Farmers always insert correct data in the system.

DA5 Farmers participating to the programme actively use the platform.

DA6 Farmers' possessions are already registered at the local Ministry.

DA7 Policy makers' decisions are solely data-driven and are not affected by personal opinions.

DA8 Humidity soil sensors are already installed and cover the entire geographical area.

DA9 Each farmer's terrain humidity data is provided by one and only sensor, associated accordingly to geographical distance from the center of the terrain.

DA10 Data provided by humidity soil sensors is always correct.

DA11 Each farmer has an automated irrigation system with a control unit.

DA12 Each irrigation control unit supports the sensors provided by DREAM.

DA13 Data provided by the irrigation sensors is always correct.

DA14 The units of measurement used will always be consistent.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

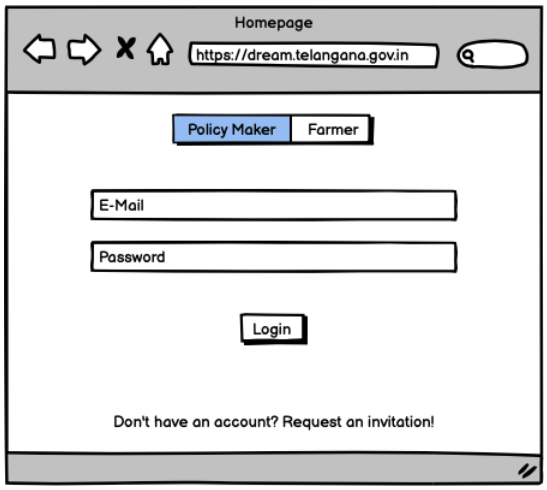

Figure 6: Login page

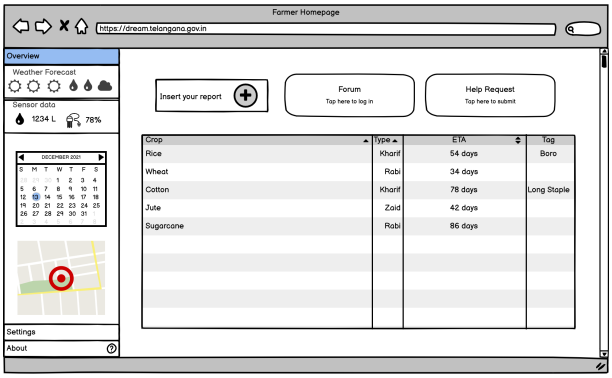

Figure 7: Homepage of farmer's web application

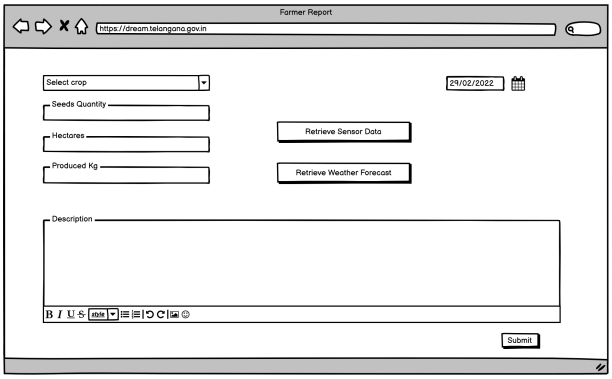

Figure 8: Farmer report form page

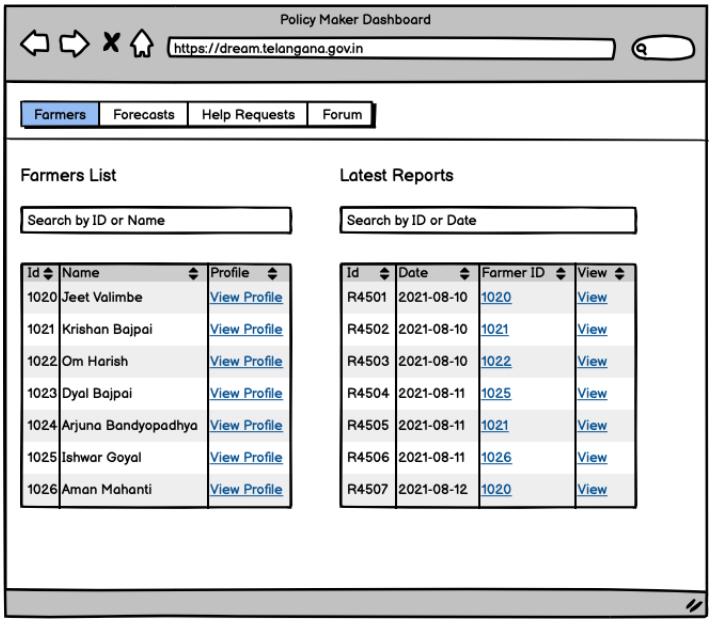

Figure 9: PM Dashboard - Farmers page

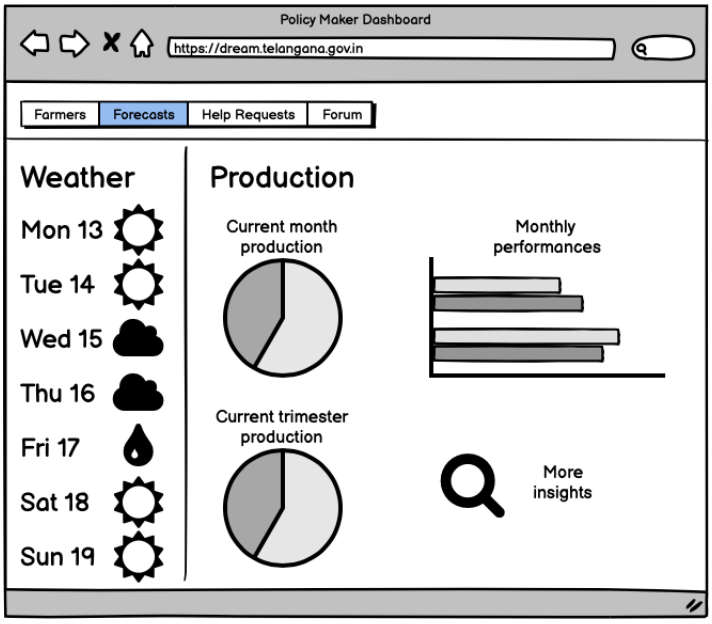

Figure 10: PM Dashboard - Forecasts page

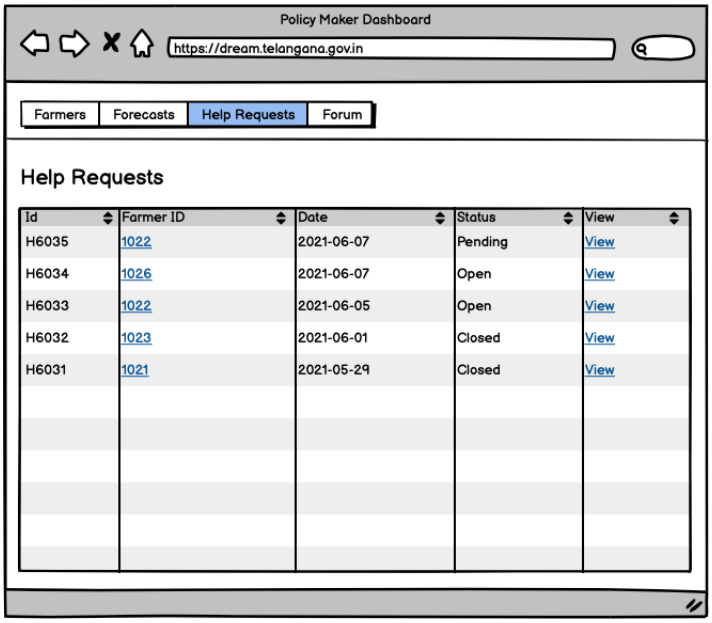

Figure 11: PM Dashboard - Help Requests page

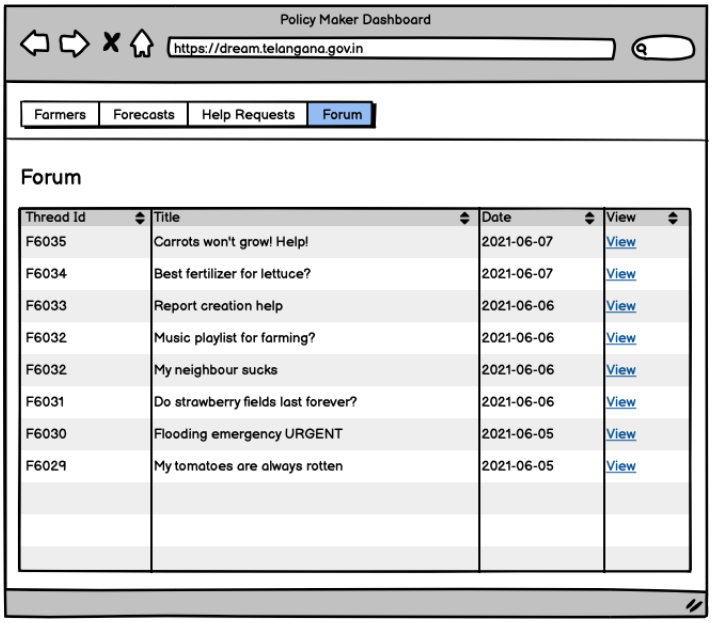

Figure 12: PM Dashboard - Forum page

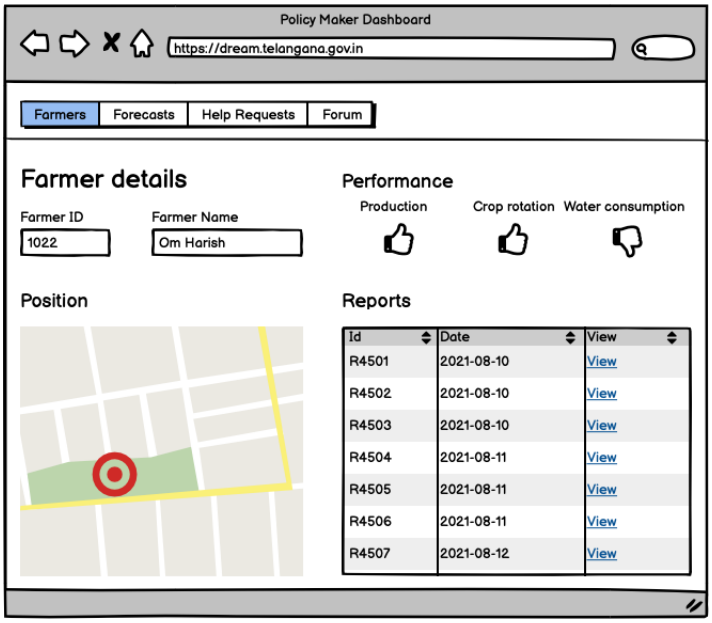

Figure 13: PM Dashboard - Farmer detail page

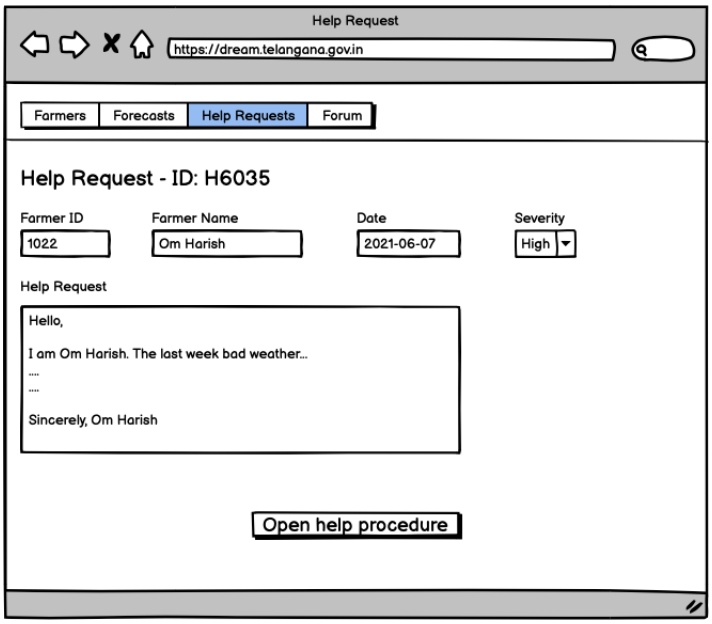

Figure 14: PM Dashboard - Help request detail page

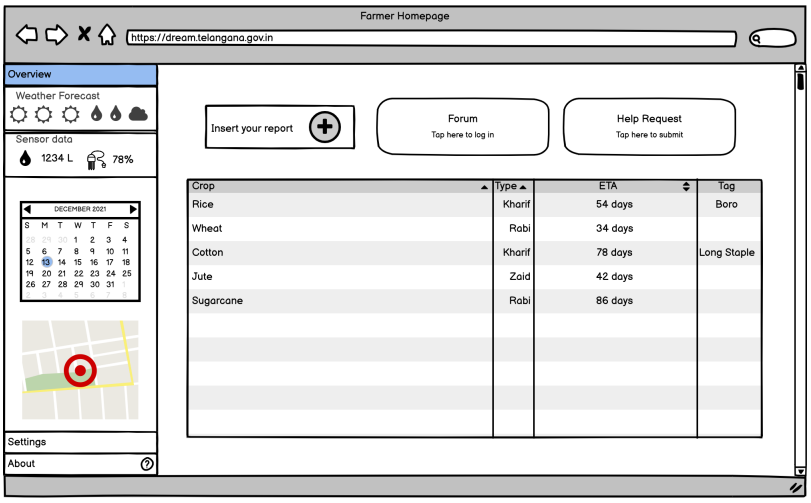

Figure 15: Farmer Dashboard - Homepage

3.1.2 Hardware Interfaces

Different types of hardware interfaces will be required to ensure that the entire infrastructure functions properly.
Farmers and PMs will need to have computers connected to the internet and capable of supporting DREAM's dedicated web app.
All sensors will have to use IoT technologies[2], in fact they will necessarily have to be connected to a Wi-Fi module and programmed to create data and send it to the DREAM server.

3.1.3 Software Interfaces

The system will use the external API of the Telengana governmental weather service to collect and transmit data of interest to farmers, depending on their area.
The system doesn't provide any API to external application because of the privacy of the farmers.

3.1.4 Communication Interfaces

All the communications between end-devices and the central system are made via internet using the HTTPS protocol. The communications between clients and sensors are made via a publish-subscribe communication protocol, namely MQTT.

3.2 Functional Requirements

3.2.1 Requirements

R1 The system shall allow system admins to register policy maker.

R2 The system shall allow policy makers to login.

R3 The system shall allow policy makers to register farmers.

R4 The system shall provide farmers credentials to log in.

R5 The system shall allow farmers to complete their profile information.

R6 The system shall allow farmers to log into the platform.

R7 The system shall allow farmers to fill reports with data about goods production.

R8 The system shall allow farmers to fill reports with data about auxiliary products used for goods production.

R9 The system shall allow farmers to fill reports with problems faced during goods production.

R10 The system shall retrieve data from sensors and attach it to reports.

R11 The system shall retrieve data from weather stations and attach it to reports.

R12 The system shall allow farmers to visualize weather forecasts.

R13 The system shall allow farmers to visualize data retrieved from humidity and irrigation sensors.

R14 The system shall allow farmers to create and contribute to forum discussions.

R15 The system shall allow farmers to create help requests.

R16 The system shall allow farmers to visualize personalized suggestions about goods and techniques.

R17 The system shall allow PMs to monitor farmers' performances.

R18 The system shall provide PMs with parameters to evaluate farmers' performances.

R19 The system shall allow PMs to compare and aggregate farmers' performances.

R20 The system shall allow PMs to filter farmers based on their information.

R21 The system shall allow PMs to filter farmers based on their performance.

R22 The system shall allow PMs to sort farmers based on their information.

R23 The system shall allow PMs to sort farmers based on their performance.

R24 The system shall allow PMs to reward best performing farmers.

R25 The system shall allow PMs to visualize weather forecasts.

R26 The system shall allow PMs to visualize help requests.

R27 The system shall allow PMs to take charge of help requests.

R28 The system shall allow PMs to visualize farmers who have requested help and their performances.

R29 The system shall allow PMs to close help requests.

R30 The system shall allow PMs to file reports about farmers interviews.

R31 The system shall allow PMs to re-schedule agronommist's control visit plans.

R32 The system shall allow PMs to create and contribute to forum discussions.

R33 The system shall allow PMs to create suggestions.

3.2.2 Goal Mapping on Requirements

G1 Allow Telangana's Policy Makers to monitor local agriculture

R1 The system shall allow system admins to register policy maker.

R2 The system shall allow policy makers to login.

R17 The system shall allow PMs to monitor farmers' performances.

R18 The system shall provide PMs with parameters to evaluate farmers' performances.

R19 The system shall allow PMs to compare and aggregate farmers' performances.

R20 The system shall allow PMs to filter farmers based on their information.

R21 The system shall allow PMs to filter farmers based on their performance.

R22 The system shall allow PMs to sort farmers based on their information.

R23 The system shall allow PMs to sort farmers based on their performance.

R25 The system shall allow PMs to visualize weather forecasts.

DA1 Each policy maker has one and one only area to monitor.

DA2 Terrains cannot be shared: each terrain is associate with one and only farmer

DA5 Farmers participating to the programme actively use the platform.

DA6 Farmers' possessions are already registered at the local Ministry.

DA8 Humidity soil sensors are already installed and cover the entire geographical area.

DA9 Each farmer's terrain humidity data is provided by one and only sensor, associated accordingly to geographical distance from the center of the terrain.

DA10 Data provided by humidity soil sensors is always correct.

DA11 Each farmer has an automated irrigation system with a control unit.

DA12 Each irrigation control unit supports the sensors provided by DREAM.

DA13 Data provided by the irrigation sensors is always correct.

DA14 The units of measurement used will always be consistent.

G2 Allow farmers to receive assistance or rewards based on their performance

R3 The system shall allow policy makers to register farmers.

R4 The system shall provide farmers with credentials to log in.

R5 The system shall allow farmers to complete their profile information.

R6 The system shall allow farmers to log into the platform.

R7 The system shall allow farmers to fill reports with data about goods production.

R8 The system shall allow farmers to fill reports with data about auxiliary products used for goods production.

R9 The system shall allow farmers to fill reports with problems faced during goods production.

R10 The system shall retrieve data from sensors and attach it to reports.

R11 The system shall retrieve data from weather stations and attach it to reports.

R24 The system shall allow PMs to reward best performing farmers.

R30 The system shall allow PMs to file reports about farmers interviews.

R31 The system shall allow PMs to re-schedule agronommist's control visit plans.

DA3 Farmers have a device able to access the platform

DA4 Farmers always insert correct data in the system.

DA5 Farmers participating to the programme actively use the platform.

DA6 Farmers' possessions are already registered at the local Ministry.

DA7 Policy makers' decisions are solely data-driven and are not affected by personal opinions.

DA8 Humidity soil sensors are already installed and cover the entire geographical area.

DA9 Each farmer's terrain humidity data is provided by one and only sensor, associated accordingly to geographical distance from the center of the terrain.

DA10 Data provided by humidity soil sensors is always correct.

DA11 Each farmer has an automated irrigation system with a control unit.

DA12 Each irrigation control unit supports the sensors provided by DREAM.

DA13 Data provided by the irrigation sensors is always correct.

DA14 The units of measurement used will always be consistent.

G3 Allow farmers to request and receive help

R3 The system shall allow policy makers to register farmers.

R4 The system shall provide farmers with credentials to log in.

R5 The system shall allow farmers to complete their profile information.

R6 The system shall allow farmers to log into the platform.

R15 The system shall allow farmers to create help requests.

R16 The system shall allow farmers to visualize personalized suggestions about goods and techniques.

R26 The system shall allow PMs to visualize help requests.

R27 The system shall allow PMs to take charge of help requests.

R28 The system shall allow PMs to visualize farmers who have requested help and their performances.

R29 The system shall allow PMs to close help requests.

R33 The system shall allow PMs to create suggestions.

DA3 Farmers have a device able to access the platform

DA4 Farmers always insert correct data in the system.

DA6 Farmers' possessions are already registered at the local Ministry.

DA7 Policy makers' decisions are solely data-driven and are not affected by personal opinions.

DA14 The units of measurement used will always be consistent.

G4 Create a farmer's network to promote collaboration and mutual aid

R3 The system shall allow policy makers to register farmers.

R4 The system shall provide farmers with credentials to log in.

R5 The system shall allow farmers to complete their profile information.

R6 The system shall allow farmers to log into the platform.

R14 The system shall allow farmers to create and contribute to forum discussions.

R32 The system shall allow PMs to create and contribute to forum discussions.

R15 The system shall allow farmers to create help requests.

R26 The system shall allow PMs to visualize help requests.

R27 The system shall allow PMs to take charge of help requests.

R28 The system shall allow PMs to visualize farmers who have requested help and their performances.

R29 The system shall allow PMs to close help requests.

DA3 Farmers have a device able to access the platform

DA4 Farmers always insert correct data in the system.

DA6 Farmers' possessions are already registered at the local Ministry.

DA7 Policy makers' decisions are solely data-driven and are not affected by personal opinions.

DA14 The units of measurement used will always be consistent.

G5 Improve farmer's productivity

R3 The system shall allow policy makers to register farmers.

R4 The system shall provide farmers with credentials to log in.

R5 The system shall allow farmers to complete their profile information.

R6 The system shall allow farmers to log into the platform.

R12 The system shall allow farmers to visualize weather forecasts.

R13 The system shall allow farmers to visualize data retrieved from humidity and irrigation sensors.

R15 The system shall allow farmers to create help requests.

R16 The system shall allow farmers to visualize personalized suggestions about goods and techniques.

R33 The system shall allow PMs to create suggestions.

DA3 Farmers have a device able to access the platform

DA4 Farmers always insert correct data in the system.

DA5 Farmers participating to the programme actively use the platform.

DA6 Farmers' possessions are already registered at the local Ministry.

DA7 Policy makers' decisions are solely data-driven and are not affected by personal opinions.

DA8 Humidity soil sensors are already installed and cover the entire geographical area.

DA9 Each farmer's terrain humidity data is provided by one and only sensor, associated accordingly to geographical distance from the center of the terrain.

DA10 Data provided by humidity soil sensors is always correct.

DA11 Each farmer has an automated irrigation system with a control unit.

DA12 Each irrigation control unit supports the sensors provided by DREAM.

DA13 Data provided by the irrigation sensors is always correct.

DA14 The units of measurement used will always be consistent.

3.2.3 Use Cases

UC1 Policy maker registration

Name	**Policy maker registration**
Actors	System admin, Policy Maker
Entry conditions	• The platform is running • The system admin has logged into the system
Flow of events	(a) The system admin creates a new policy maker profile (b) The system admin inserts the policy maker data in the newly created profile (c) The system automatically generates log in credentials for the policy maker (d) The system automatically sends an e-mail to the address associated to the profile containing credentials and instructions to log in
Exit conditions	The policy maker receives his credentials
Exceptions	• If the policy maker e-mail address is already in use, the process is aborted.

Table 1: *Policy maker registration* use case description

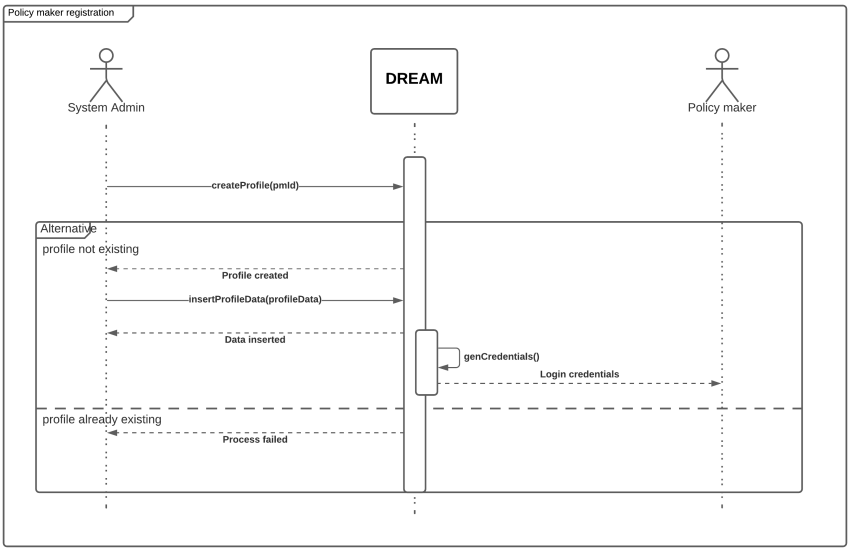

Figure 16: Policy maker registration

UC2 Farmer Registration

Name	**Farmer Registration**
Actors	Farmer, Policy Maker
Entry conditions	• The platform is running • The policy maker has logged into the system
Flow of events	(a) The policy maker sends an invite via e-mail to the farmer (b) The farmer accepts the invite (c) The policy maker creates a new profile and inserts the farmer data (d) The system generates the credentials for the farmer login (e) The farmer receives the credentials via e-mail (f) The farmer logs into the platform and completes the registration (g) The system saves the farmer data
Exit conditions	The farmer is registered in the system
Exceptions	• If the farmer does not complete the registration in time, the process is aborted. The farmer can request to repeat the registration by sending an e-mail.

Table 2: *Farmer Registration* use case description

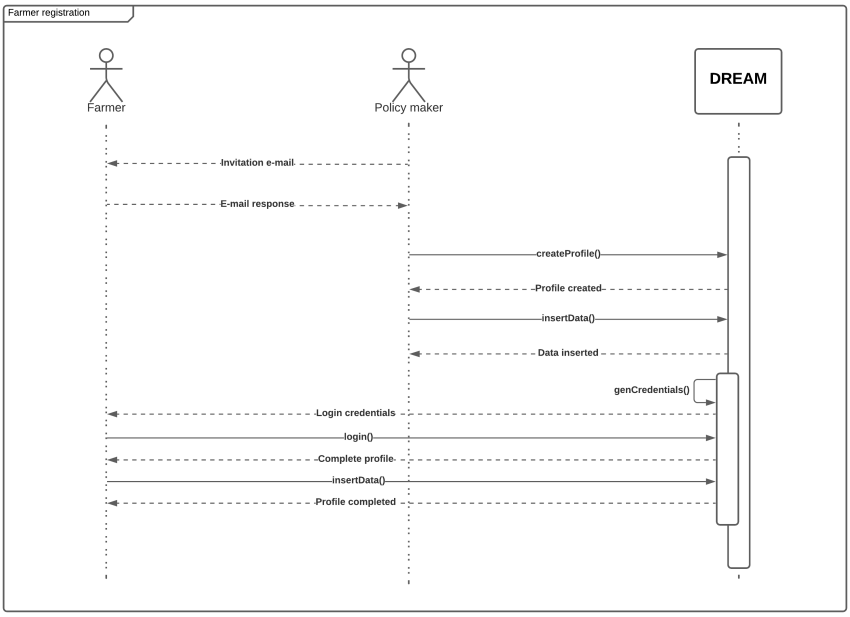

Figure 17: Farmer Registration

UC3 User login

Name	**User login**
Actors	Farmer, Policy Maker
Entry conditions	The platform is running
Flow of events	(a) The user goes to the login page (b) The user inserts the login credentials (c) The user submits the form
Exit conditions	The user has successfully logged into the platform
Exceptions	• If the login credentials are incorrect, the system notifies the user and the process is aborted.

Table 3: *User login* use case description

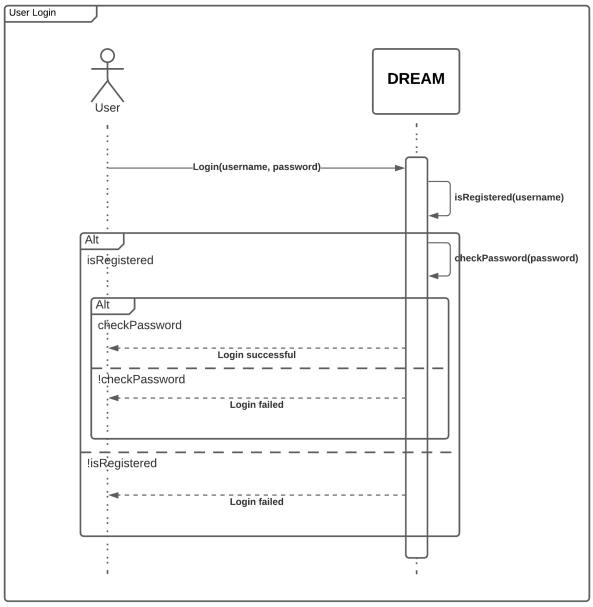

Figure 18: User login

UC4 Insertion of report

Name	**Insertion of report**
Actors	Farmer
Entry conditions	• The platform is running • The farmer has logged into the system • The farmer has collected his crop
Flow of events	(a) The farmer inserts the type and quantity of goods produced (b) The farmer inserts the type and quantity of support products used (c) The farmer application retrieves data from sensors and inserts it into the report (d) The farmer sends the report (e) The system checks the report (f) The system stores the report
Exit conditions	The report is stored in the system
Exceptions	• If the report contains wrong information, the process is aborted.

Table 4: *Insertion of report* use case description

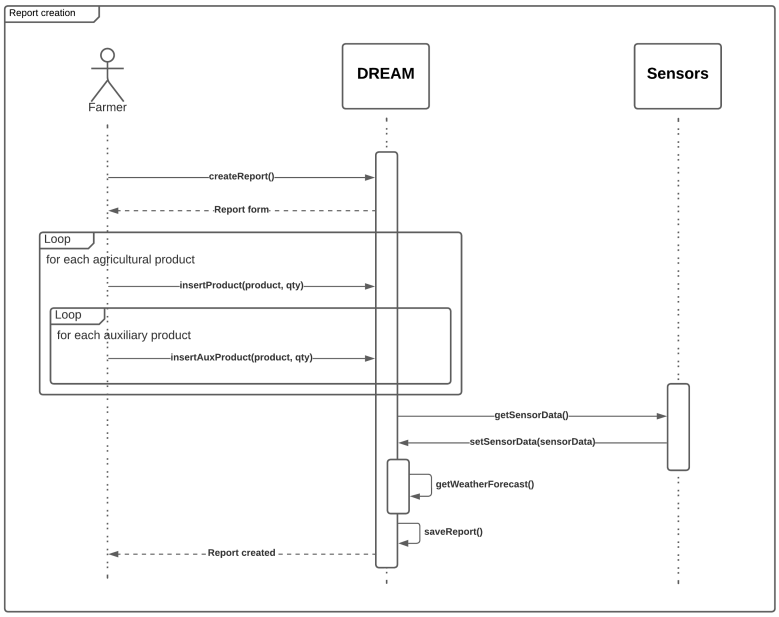

Figure 19: Insertion of report

UC5 Creation and handling of a help request

Name	**Creation and handling of a help request**
Actors	Farmer, Policy Maker
Entry conditions	• The platform is running • The farmer has logged into the system • The policy maker has logged into the system • The farmer needs help
Flow of events	(a) The farmer compiles the help request form (b) The help request is created with *pending* status (c) The policy maker takes charge of the help request (d) The help request status becomes *open* (e) The policy maker evaluates the severity of the help request (f) The policy maker takes the necessary actions based on the severity (g) The policy maker monitors the farmer's performances in the next period (h) If the farmer's performances do not improve, the severity of the request is increased and new actions are taken (*back to (f)*) (i) If the farmer's performances improve, the help request is closed
Exit conditions	The help request status becomes *closed*
Exceptions	• The farmer can close the help request by itself during the process if authority help is no more necessary

Table 5: *Creation and handling of a help request* use case description

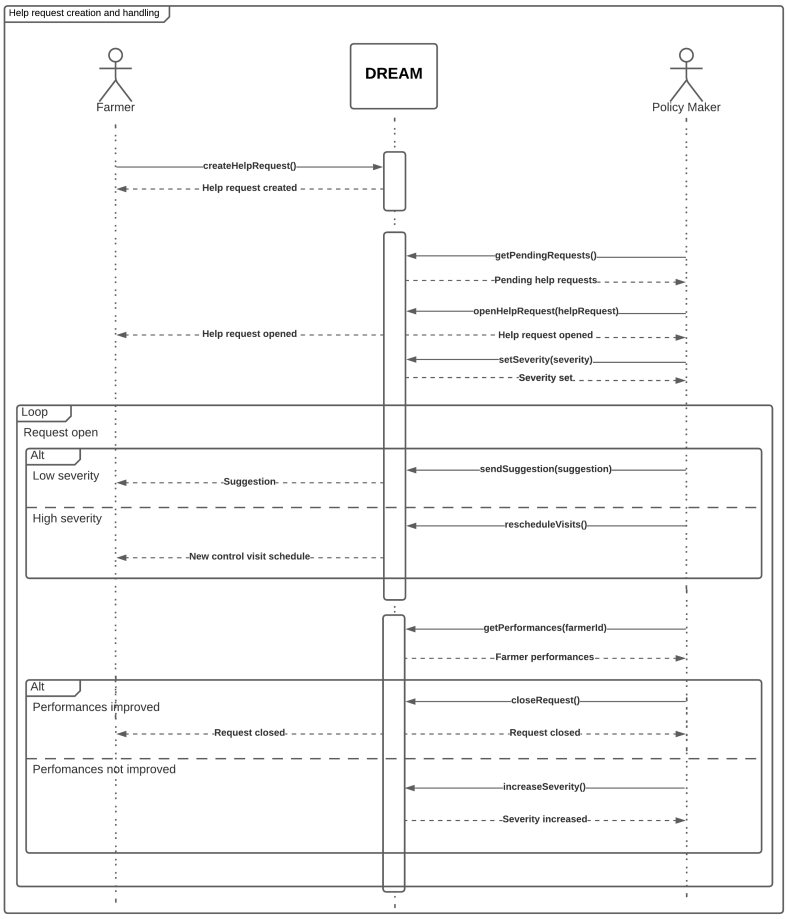

Figure 20: Creation and handling of a help request

UC6 Interview of a good-performing farmer

Name	**Interview of a good-performing farmer**
Actors	Farmer, Policy Maker
Entry conditions	• The platform is running • The policy maker has logged into the system • The policy maker identifies a good-performing farmer
Flow of events	(a) The policy maker sends a request for an interview to the selected farmer (b) The farmer responds to the request (c) If the response is negative, the process is aborted (d) If the response is positive, the policy maker and the farmer agree on a date to do the interview (e) On the decided date, an interview via phone is held (f) Once the interview is ended, the policy maker inserts into the system the relevant information obtained (e.g. suggestions to other farmers)
Exit conditions	The interview information is inserted into the system
Exceptions	• If the policy maker and the farmer cannot agree on a date, the process is aborted

Table 6: *Interview of a good-performing farmer* use case description

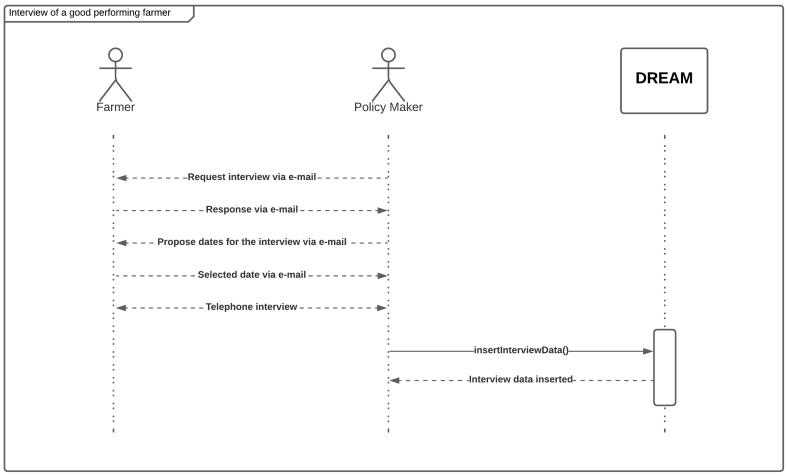

Figure 21: Interview of a good-performing farmer

UC7 Rewarding of good-performing farmers

Name	**Rewarding of good-performing farmers**
Actors	Farmer, Policy Maker
Entry conditions	• The platform is running • The policy maker has logged into the system • The trimester has ended
Flow of events	(a) The policy maker visualizes the list of the top 10 performing farmers of the trimester (b) The PM sends to each of them a mail notificating that their performances were excellent and containing instructions on how to claim their rewards (c) For each farmer, an interview is scheduled (if possible)
Exit conditions	Every interview is scheduled

Table 7: *Rewarding of good-performing farmers* use case description

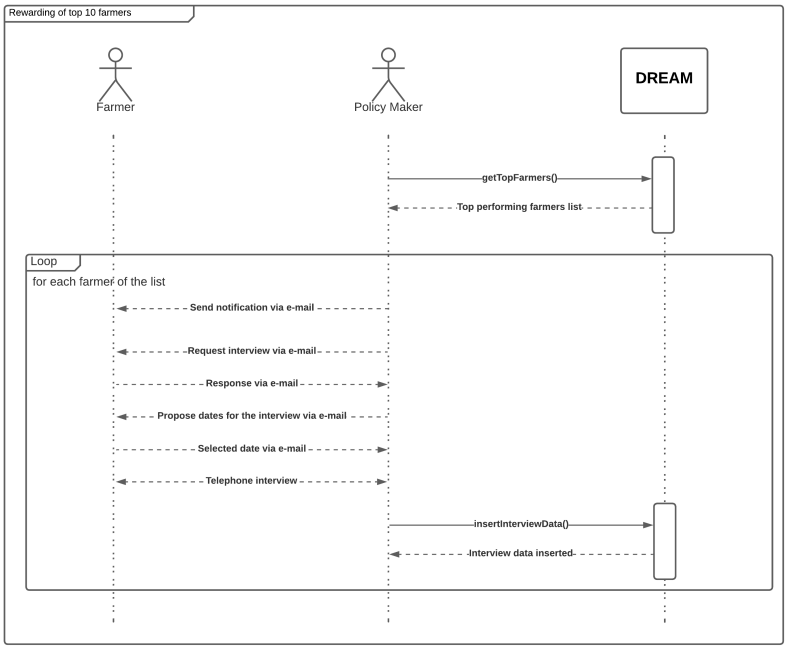

Figure 22: Rewarding of good-performing farmers

UC8 Forum discussion

Name	**Forum discussion**
Actors	User
Entry conditions	• The platform is running • The user has logged into the platform
Flow of events	(a) The user creates a new forum discussion about a specific topic of its interest (b) The system publishes the thread on the farmer's forum (c) Another user notices the new thread in the discussions list (d) The interested user submits his answer (e) The user who has created the discussion checks his discussion and reads the answer submitted (f) The user has solved his problem and closes the discussion
Exit conditions	The user closes the discussion after reading the answer
Exceptions	• The user finds a solution to his problem by himself and closes the discussion

Table 8: *Creation of a forum discussion* use case description

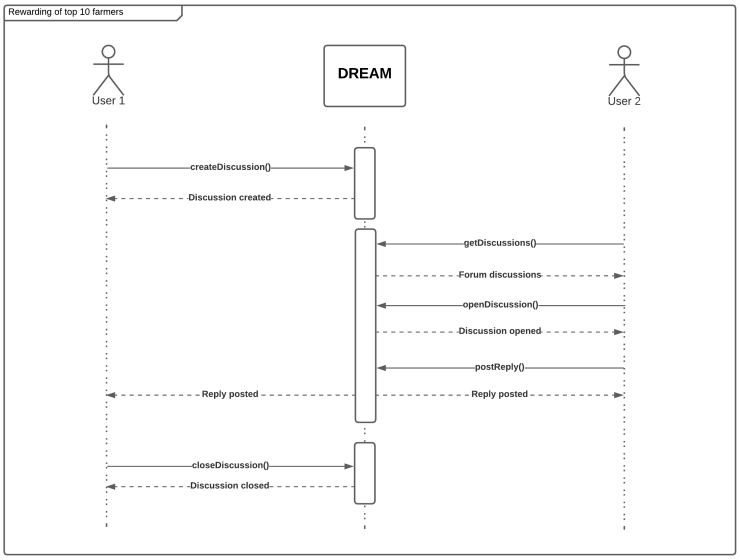

Figure 23: Forum discussion

3.2.4 Traceability matrix

	G1	G2	G3	G4	G5	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
R1	X					X							
R2	X							X		X	X	X	
R3		X	X	X	X		X						
R4		X	X	X	X		X						
R5		X	X	X	X		X						
R6		X	X	X	X			X	X	X	X	X	X
R7		X							X				
R8		X							X				
R9		X							X				
R10		X							X				
R11		X							X				
R12					X					X			
R13					X					X			
R14				X									X
R15			X	X	X					X			
R16			X		X					X			
R17	X											X	
R18	X											X	
R19	X											X	
R20	X											X	
R21	X											X	
R22	X											X	
R23	X											X	
R24		X										X	
R25	X									X			
R26			X	X						X			
R27			X	X						X			
R28			X	X						X			
R29			X	X						X			
R30		X									X	X	
R31		X									X	X	
R32				X									X
R33			X		X					X			

Table 9: Traceability matrix

3.3 Performance Requirements

Since the system does not need immediate responsiveness, the latency between components can also be in the order of seconds (e.g. 1 to 20 sec).
The amount of data produced by IoT sensors is significantly high, the system is expected to be able to process and classify all the data, thus having high storage capacities.

3.4 Software System Attributes

3.4.1 Availability

The target is to achieve an improved level of resilience with semi-automated recovery and to be taken offline for maintenance in agreed windows. System relies on people that use it during the working hours, moreover, it is not needed to provide very high availability because the system is not strictly related with emergency situations. According to these assumptions, the system should provide an Availability of 99.5%.

Availability %	Downtime per year	Downtime per month
99.5%	1.83 days	3.6 hours

3.4.2 Reliability

The operations performed must be executed as intended as frequently as possible, in fact the probability that the data displayed is correct must be very high, so as not to compromise the evaluations performed by the actors in the system. To meet these requirements, the reliability of the system must not be less than 99.97%.

Reliability %	Downtime per year	Downtime per month
99.97%	2.63 hours	13.14 minutes

3.4.3 Security

The data entered into the platform needs to be protected as it contains confidential information about telengana farmers. A secure communication channel and encryption of messages will be necessary.
Operations will always need to be authorized with an authentication mechanism, specifically the PM login mechanism must be 2FA (two-factor authentication).

3.4.4 Maintainability

The infrastructure needs thorough documentation of the code base to ensure that the underlying logic can be quickly understood by maintenance personnel. Through the use of design patterns and best practices in the implementation, it will be easier to ensure future additions.

3.4.5 Portability

The client application, including the front-end and user interface, must be supported by Windows and Mac operating systems.
A mobile version of the client application is not planned.

3.5 Design Constraints

3.5.1 Standards compliance

In order to effectively store data into the DBMS and to easily retrieve significant information, the reports created by farmers will have to follow a standard model. Particular attention is required when inserting the types of goods and auxiliary products in the report: these will have to be the same across different reports in order to facilitate querying and data aggregation. In order to do this, the report form will guide farmers to insert correct data and avoid misalignments with the information stored in the DBMS: the ultimate goal is to prevent errors such as typos when compiling the report form and to avoid the insertion of data which is incoherent with the other reports.

4 Formal analysis using Alloy

Alloy [1] is an open source language and analyzer for software modeling. In the following section is presented a formal model of the problem specification, showing how constraints and properties will be satisfied. This section also provides some possible worlds to clarify some critical aspects of the problem. In the Alloy representation of the problem, some aspects have not been included for the sake of simplicity and to focus on the main aspects to analyze. For example, there is no representation of the platform's forum, since it represents a very basic functionalities with few constraints and no significant properties. Particular focus is instead put into the representation of the geographical aspects of the problem specification.

4.1 Alloy Model

```
1  ---------------------------------
2  -- Signatures
3  ---------------------------------
4  sig StringVar {}
5  sig Date {}
6  sig Time {}
7  sig DateTime {}
8  sig Coordinates {}
9  sig GeographicalExtension {}
10 sig ForecastData {}
11 sig SensorType {}
12 sig SensorDataPayload {}
13
```

```
14 sig UserID {}
15 sig Email {}
16 sig Password {}
17 sig Username {}
18
19 sig ProductID {}
20
21 sig SuggestionID {}
22
23 sig HelpRequestID {}
24 abstract sig HReqStatus {}
25 sig PENDING extends HReqStatus {}
26 sig OPEN extends HReqStatus {}
27 sig CLOSED extends HReqStatus {}
28 abstract sig HReqSeverity {}
29 sig LOW extends HReqSeverity {}
30 sig MEDIUM extends HReqSeverity {}
31 sig HIGH extends HReqSeverity {}
32
33 sig TerrainID {}
34
35 sig AreaID {}
36
37 sig ForecastID {}
38
39 sig ReportID {}
40
41 sig SensorID {}
42
43 abstract sig Product {
44   productId: ProductID,
45 }
46 sig AgrProduct extends Product {}
47 sig AuxProduct extends Product {}
48
49 abstract sig User {
50   userID: UserID,
51   email: one Email,
52   username: Username,
53   password: Password,
54 }
55
56 sig Farmer extends User {}
57 sig PolicyMaker extends User {} {
58   one this.~supervisor // each policy maker has one and one only
       area to monitor
59 }
60
61 sig Area {
62   areaID: AreaID,
63   name: StringVar,
64   supervisor: one PolicyMaker,  // each area is monitored by one
       and one only policy maker
65   coordinates: Coordinates
66 }
67
68 sig Terrain {
```

```
  terrainID: TerrainID,
  area: Area,
  owner: one Farmer, // each terrain is associated to one and only
    farmer
  coordinates: Coordinates,
  extension: GeographicalExtension,
}

sig Sensor {
  sensorID: SensorID,
  terrain: Terrain,
  type: SensorType
}

sig SensorData {
  sensor: Sensor,
  date: one Date,
  payload: SensorDataPayload
}

sig WeatherForecast {
  forecastID: ForecastID,
  area: Area,
  date: one Date,
  forecastData: one ForecastData
}

sig Production {
  product: AgrProduct,
  quantity: Int,
  auxiliaryProducts: some AuxProduct
}

sig Report {
  reportID: ReportID,
  date: one Date,
  creator: one Farmer,
  problems: set StringVar,
  productions: some Production,
  forecast: set WeatherForecast,
  sensorData: set SensorData
}

sig HelpRequest {
  helpRequestID: HelpRequestID,
  creator: one Farmer,
  managedBy: one PolicyMaker,
  severity: HReqSeverity,
  status: HReqStatus,
  creationDate: one Date,
  openingDate: one Date,
  closureDate: one Date
}

sig Suggestion {
  suggestionId: SuggestionID,
  product: Product,
```

```
  date: one Date
}

---------------------------------
-- Facts
---------------------------------

fact uniqueUserIDs {
  no disj u1, u2: User | u1.userID = u2.userID
}

fact noUserIDWithoutUser {
  all uid: UserID | one u: User |  u.userID = uid
}

fact uniqueProductIDs {
  no disj p1, p2: Product | p1.productId = p2.productId
}

fact noProductIDWithoutProduct {
  all pid: ProductID | one p: Product |  p.productId = pid
}

fact uniqueSensorIDs {
  no disj s1, s2: Sensor | s1.sensorID = s2.sensorID
}

fact noSensorIDWithoutSensor {
  all sid: SensorID | one s: Sensor |  s.sensorID = sid
}

fact uniqueSuggestionIDs {
  no disj s1, s2: Suggestion | s1.suggestionId = s2.suggestionId
}

fact noSuggIDWithoutSuggestion {
  all sid: SuggestionID | one s: Suggestion |  s.suggestionId = sid
}

fact uniqueTerrainIDs {
  no disj t1, t2: Terrain | t1.terrainID = t2.terrainID
}

fact noTerrainIDWithoutTerrain {
  all tid: TerrainID | one t: Terrain |  t.terrainID = tid
}

fact noExtWithoutTerrain {
  all ge: GeographicalExtension | one t: Terrain | t.extension = ge
}

fact uniqueAreaIDs {
  no disj a1, a2: Area | a1.areaID = a2.areaID
}

fact noAreaIDwithoutArea {
  all aid: AreaID | one a: Area |  a.areaID = aid
```

```
}

fact uniqueHelpRequestIDs {
  no disj h1, h2: HelpRequest | h1.helpRequestID = h2.helpRequestID
}

fact noHReqIDWithoutHReq {
  all hid: HelpRequestID | one h: HelpRequest |  h.helpRequestID =
    hid
}

fact noHreqStatusWithoutHReq{
  all hS: HReqStatus | one h: HelpRequest |  h.status = hS
}

fact noSeverityWithoutHReq {
  all hS: HReqSeverity | one h: HelpRequest | h.severity = hS
}

fact uniqueReportIDs {
  no disj r1, r2: Report | r1.reportID = r2.reportID
}

fact noReportIDWithoutReport {
  all rid: ReportID | one r: Report |  r.reportID = rid
}

fact noNegativeQuantity {
  all p: Production | p.quantity > 0
}

fact uniqueUserEmail {
  no disj u1, u2: User | u1.email = u2.email
}

fact noEmailWithoutUser {
  all e: Email | one u: User |  u.email = e
}

fact uniqueUsernames {
  no disj u1, u2: User | u1.username = u2.username
}

fact noUsernameWithoutUser {
  all uname: Username | one u: User | u.username = uname
}

fact noPasswordWithoutUser {
  all psw: Password | one u: User | u.password = psw
}

fact noFarmerWithoutTerrain {
  all f: Farmer | one t: Terrain | t.owner = f
}

fact noFIDWithoutForecast {
  all fid: ForecastID | one f: WeatherForecast | f.forecastID = fid
```

```
}

fact noFDataWithoutForecast {
  all fd: ForecastData | one f: WeatherForecast | f.forecastData = fd
}

fact noDuplicateForecast { //no weather forecasts for the same area and for the same date
  no disj wf1, wf2: WeatherForecast | (wf1.area = wf2.area) and (wf1.date = wf2.date)
}

fact noStypeWithoutSensor {
  all stype: SensorType | one s: Sensor | s.type = stype
}

fact noPayloadWithoutSData {
  all sdp: SensorDataPayload | one sd: SensorData | sd.payload = sdp
}

fact noProductionWithoutReport {
  all p: Production | one r: Report | p in r.productions
}

fact noProductWithoutProduction {
  all p: Product | one prod: Production | ((prod.product = p) or (p in prod.auxiliaryProducts))
}

fact noTerrainsWithSameCoordinates {
  no disj t1, t2: Terrain | t1.coordinates = t2.coordinates
}

fact noAreaWithSameCoordinates {
  no disj a1, a2: Area | a1.coordinates = a2.coordinates
}

fact noTerrainWithAreaCoordinates {
  no disj t: Terrain, a: Area | t.coordinates = a.coordinates
}

fact noStandaloneDate {
  all d: Date | (some h: HelpRequest | h.creationDate = d) or
   (some h: HelpRequest | h.openingDate = d) or
   (some h: HelpRequest | h.closureDate = d) or
   (some f: WeatherForecast | f.date = d) or
   (some s: SensorData | s.date = d) or
   (some sug: Suggestion | sug.date = d) or
   (some r: Report | r.date = d)
}

fact sensedHelpRequest { // an help request can be created and opened the same day, but not closed
  no hp: HelpRequest | hp.creationDate = hp.closureDate or hp.openingDate = hp.closureDate
```

```
}

---------------------------------
-- Predicates
---------------------------------

pred world1 { //focus on sensors
  #Sensor = 3
  #Terrain = 2
  #WeatherForecast = 2
  #Report = 0
  #HelpRequest = 0
}
run world1 for 5

pred world2 { //focus on reports
  #PolicyMaker = 1
  #Farmer = 1
  #Report = 1
  #Production = 2
  #ForecastData = 1
  #Suggestion = 0
  #HelpRequest = 0
  #SensorData = 1
}
run world2 for 5

pred world3 { //focus on help requests
  #HelpRequest = 2
  #Date = 5
  #Report = 0
  #Suggestion = 0
  #WeatherForecast = 0
  #SensorData = 0
}
run world3 for 5
```

4.1.1 First World

The first world, represented in Figure 24, focuses on the representation of terrains and geographical areas, including related sensors and weather forecasts. Each sensor is associated to one terrain and each forecast to an area, while each terrain is associated with a farmer and each area with a supervisor.

4.1.2 Second World

The second world, represented in Figure 25, extends the concepts represented in the first world by representing a report inserted by the farmer into the platform. A report is related to a terrain and contains information about productions, sensors and also forecasts for the considered time period.

4.1.3 Third World

In the third world, represented in Figure 26, the focus is on the representation of help requests created by a farmer. Each request has a severity, a status and three dates concerning the creation, opening and closure of the request and it is managed by a Policy Maker.

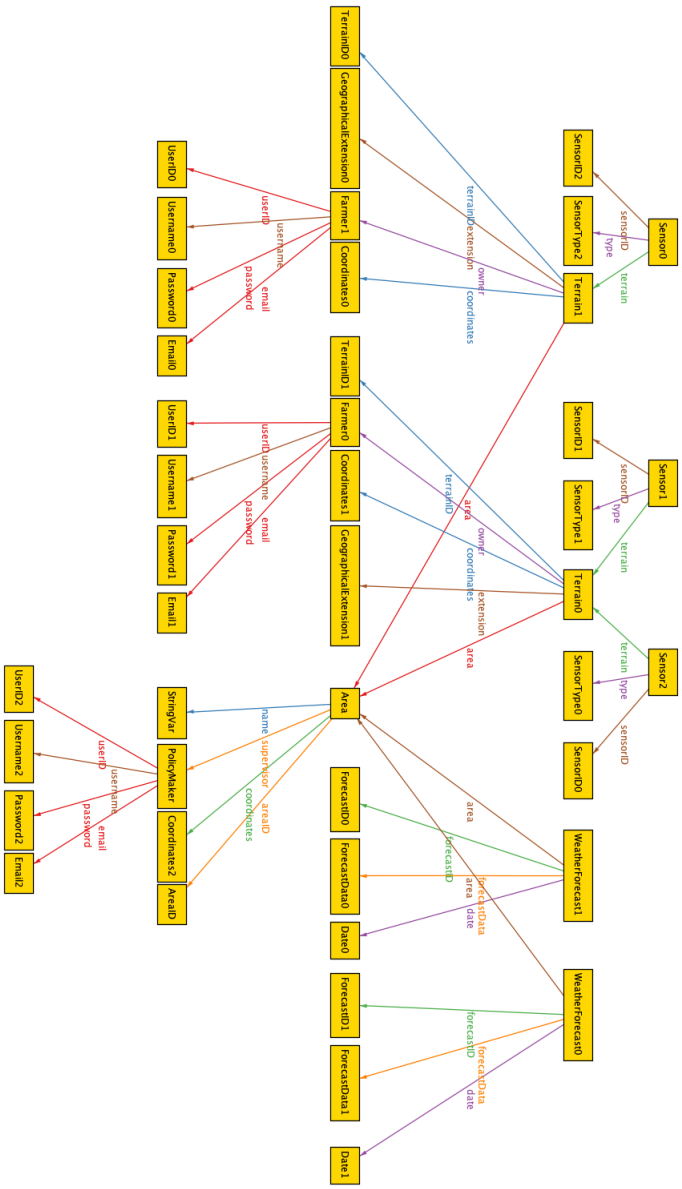

Figure 24: First World

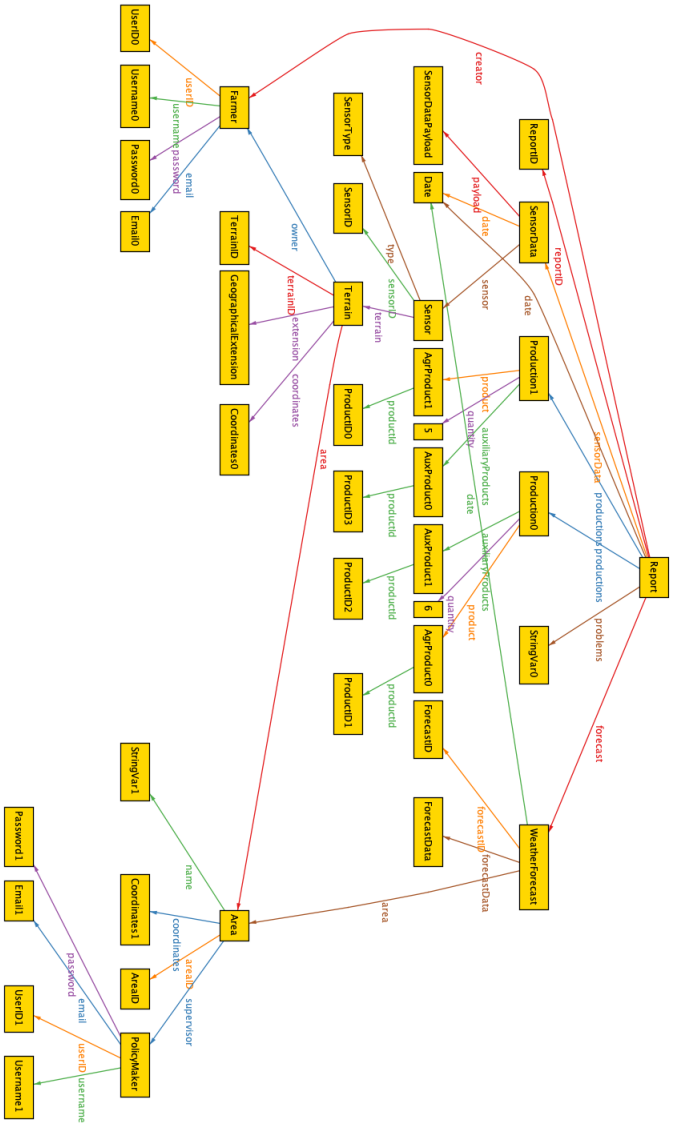

Figure 25: Second World

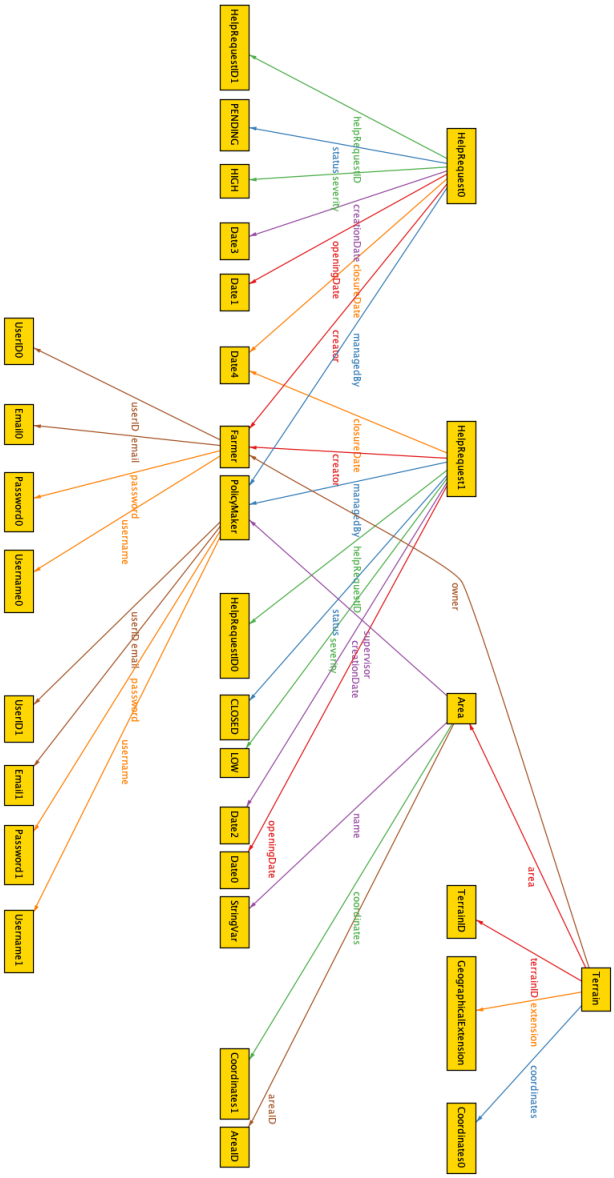

Figure 26: Third World

5 Effort Spent

This section keeps track of team and personal effort spent to produce the RASD and its sections. The activities that involved crucial decisions regarding problem specification and proposed solutions were performed in group, while the individual work mainly consists in activities that do not involve decision-making and team discussions.

5.1 Team Effort

Task	**Hours**
Initial briefing	4
Introduction	1
Class Diagram	2
State Diagram	2
Product Functions	1
Constraints & Domain Assumptions	1
External Interfaces	1
Functional Requirements	1
Use Cases	1
Alloy	2
Document Revisioning	2
Total	20

Table 10: Team effort

5.2 Mauro's Effort

Task	**Hours**
Product Functions	8
Constraints & Domain Assumptions	3
External Interfaces	2
User Interfaces	4
Functional Requirements	8
System Performance & Attributes	2
Alloy	2
Total	30

Table 11: Mauro's effort

5.3 Giacomo's Effort

Task	Hours
Introduction	2
Class Diagram	2
State Diagrams	4
User Interfaces	2
Use cases	8
Alloy	10
Document revisioning	2
Total	30

Table 12: Giacomo's effort

References

[1] alloytool.org. *Alloy Documentation*. Software Design Group. URL: http://alloytools.org/documentation.html.

[2] Nurulisma Ismail. *Smart irrigation system based on internet of things (IOT)*, 2019. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1339/1/012012.

[3] omg.org. *UML Documentation*. Standards Development Organization. URL: https://www.omg.org/spec/UML/.