POLITECNICO
MILANO 1863

Dipartimento di Elettronica, Informazione e Bioingegneria

Software Engineering II Project RASD

DREAM

Authors:
Farimah Anvari: 10772192
Sajedeh Firouzizadeh: 10771435

Supervisors:
Prof. Elisabetta Di Nitto
Prof. Matteo Rossi

December 2021

Contents

1 Introduction

1.1 Purpose

This document focuses on Requirements Analysis and Specification Document (RASD) and contains the description of the main goals, the domain and its representation through some models, the analysis of the scenario with the uses cases that describe them, the list of the most important requirements and specifications that characterize the development of the software described below.

It also includes the research about the interfaces, functional and non-functional requirements and the attributes that distinguish the quality of the system.

This document has the purpose to guide the developer in the realization of the software called DREAM, Data-dRiven PrEdictive FArMing in Telengana.

Finally, to understand better the development of the document, it contains the history that describes how it is made, with the references used and the description of its structure.

1.2 Scope

Here a review of which is the scope of the application is made referring to what has been stated in the RASD document. The goal of this app is to design, develop and demonstrate anticipatory governance models for food systems using digital public goods and community-centric approaches to strengthen data-driven policy making in the state of Telengana.

Basic service Farmers can visualize data relevant to them and also allow farmers to follow the progress of their lands and the details about the irrigation and products based on the information that the application gave them. Also, they can insert the data about their lands and also the updates about their products.

Advance Function1 Second functionality point out about the farmers that can create discussions with other farmers. Also, they can insert any problem that they are faced.

1.2.1 World Phenomena

WP1	Farmers want to insert data
WP2	Farmers want to start discussions
WP3	Sensors sends data
WP4	Water irrigation system sends the data
WP5	Farmers update data about their lands
WP6	Policy Maker choose the best farmer
WP5	Policy Maker choose the worst farmer

1.2.2 Shared phenomena

SP1	Receive a notification about suggestions for the land
SP2	Receive a notification from policy maker
SP4	Farmers choose which add products to the app
SP5	Farmers add their lands to the app
SP6	Receive information related to forum
SP7	Suggestions for farmers to improve their performance

1.2.3 Goals

G1	Allow farmers to see the details of their farm lands
G2	Allow policy makers to see the details of different lands
G3	Allow policy makers to identify farmers who are performing well
G4	Allow policy makers to identify farmers who need help & performing particularly badly
G5	Allow farmers to create new forum
G6	Allow farmers to join in discussions
G7	Allow farmers to send message in discussions
G8	Allow farmers to leave in discussions
G9	Allow farmers to create new forum
G10	Allow farmers to ask problems
G11	Allow farmers to answer to problems
G12	Show farmers some personal suggestions

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

Policy Maker	A person who observe farmers progress
Farmers	A person who has a farm land on Telengana
Water-Irrigation System	An organization who observe the amount of used water
Sensor	A device that sense a physical phenomena

1.3.2 Acronyms

RASD	Requirement Analysis and Specification Document
GPS	Global Positioning System
app	Application
API	Application Programming Interface
DDoS	Distributed Denial of Service

1.3.3 Abbreviations

WPn	World Phenomenon number n
SPn	Shared Phenomenon number n
Gn	Goal number n
BS	Basic Service of DREAM
AF1	Advance Function 1 of DREAM
Rn	Requirement number n
Dn	Domain assumption number n
Cn	Constraint number n

1.4 Revision history

Date	Modifications
05/12/2020	First document
12/12/2020	• Adding use cases • Update Class Diagram.
22/12/2020	Update Alloy

1.5 Reference Documents

- Specification Document: "R&DD Assignment A.Y. 2021-2022.pdf"
- Slides of the lectures.

1.6 Document Structure

This document is divided in six sections.

- **Chapter 1** describes the purpose of this document and contains the description of the given problem we want to solve with our application. We state the goals of DREAM and we describe the phenomena related to the "world" where it will be used and the ones related to our system.
- **Chapter 2** is about presenting the product perspective, including details on how we abstracted the problem using a class diagram. We describe the main functions of the application using also some state diagrams. We resent the needs of the potential users of the application. Finally we state the domain assumptions and the dependencies.
- **Chapter 3** contains the external interface requirements, including: user interfaces, hardware interfaces, software interfaces and communication interfaces. We define the functional requirements and the use cases. We use class diagrams and sequence diagrams to describe better the use cases and the interaction between different parts of the system. Lastly we include the performance requirements and the software system attributes.
- **Chapter 4** contains a model written using the Alloy language in order to describe formally the application
- **Chapter 5** contains the tables where we reported for each group member the hour spent working on the project
- **Chapter 6** include the reference documents.

2 Overall Description

2.1 Product perspective

2.1.1 UML Description

The UML below describes at a high level the model of the system to be developed. It considers the basic service together with the advanced function 1 which previously specified. The UML does not include every class that will be necessary to define the complete architecture of the system. DREAM offers different functions beyond the basic service. Here we can identify the main aspects related to DREAM:

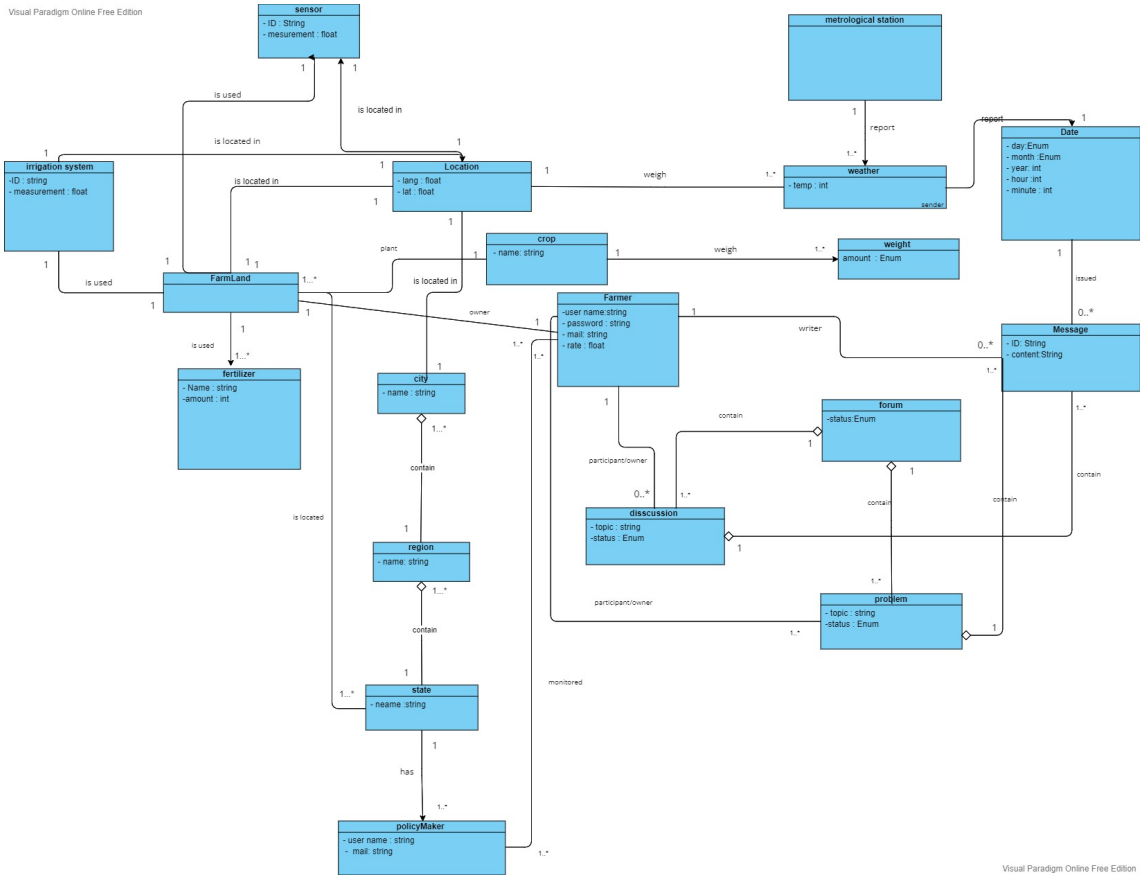
Each farmland is shown by a specific point on the map and the policymaker by selecting each point can see the detail of farmland location, Progress status of each farmer, type of production, humidity, and used water. also, farmers can see their own status and insert faced problems in the system and production information.

After login farmers should Insert the location of their farmland and based on the type of production and weather forecast of these locations can suggest what crop to plant and which fertilizer is better to use also they can insert production detail during the growth of crops and any related problem they face and by a related expert, system find the best solution and help them to keep going.

Sensors that are located in the depth of land with defined distance, periodically send messages containing soil humidity and send it to the network access point to gather and analyze and give specific alerts to necessity role.

An irrigation system that is located all over the land with defined distance, automatically measuring the used water sends to the network access point to gather and analysis and give specific alert to necessity role.

policymaker assesses the performance of the farmer and determines the best and worst ones with collected environmental data, such as the used amount of water that measure by irrigation system or humidity of soil that obtained from a sensor located in each farmland. Also, consider the type of production and location of land or even farmer faced a problem.



2.2 state Charts

Now we are going to analyze some critical aspects of the application, modeling their behaviors and showing the evolution over time of their states through appropriate state diagrams, which are reported below.

The state diagram mainly reports the changes of state in class report

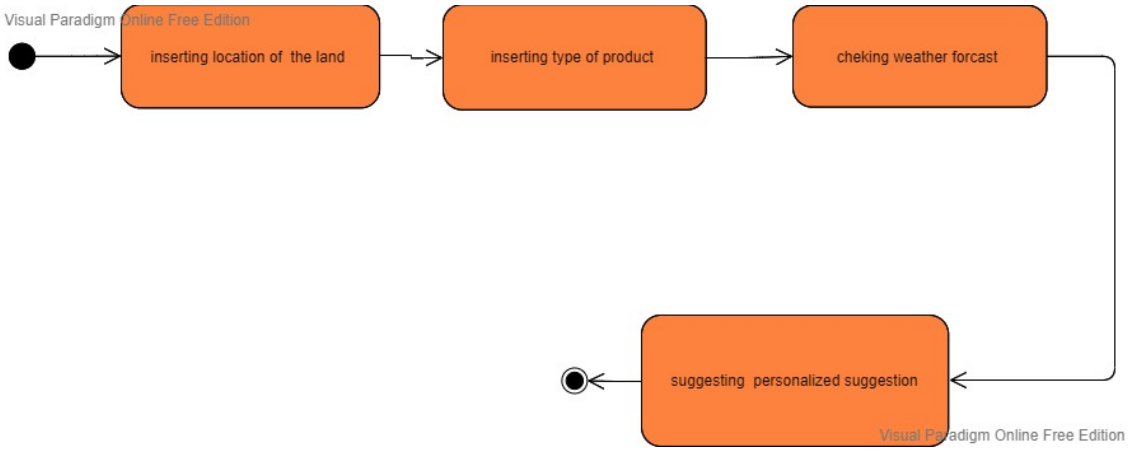


Figure 1: State Diagram 1 - inserting of initial information to give personalized suggestion

In the first diagram (figure1),we model the inserting of initial information by Farmer showing the first step after login to give personalized suggestion

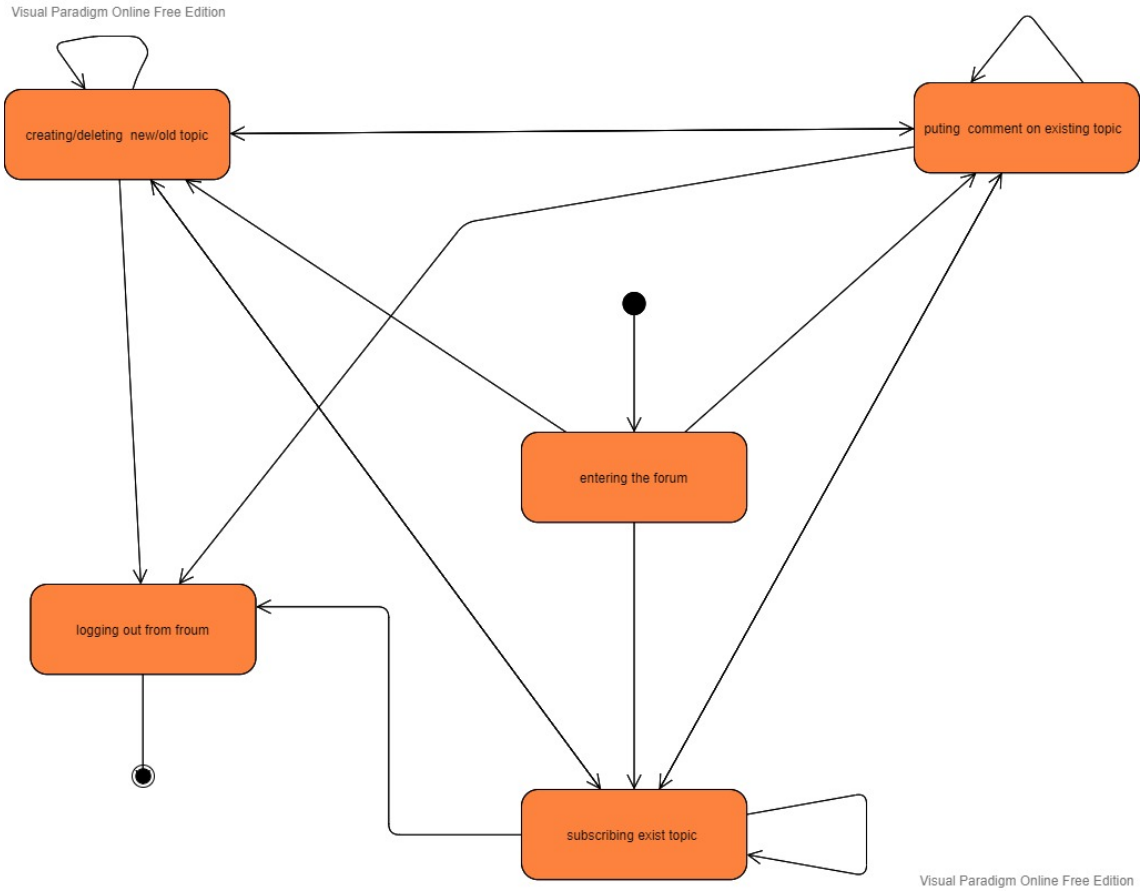


Figure 2: State Diagram 2 - communicating in forum with other farmer

in the second diagram (figure2), we model how the farmer can communicate with another one, they can create or delete the topic, but comment on the topic, and subscribing to existing topics

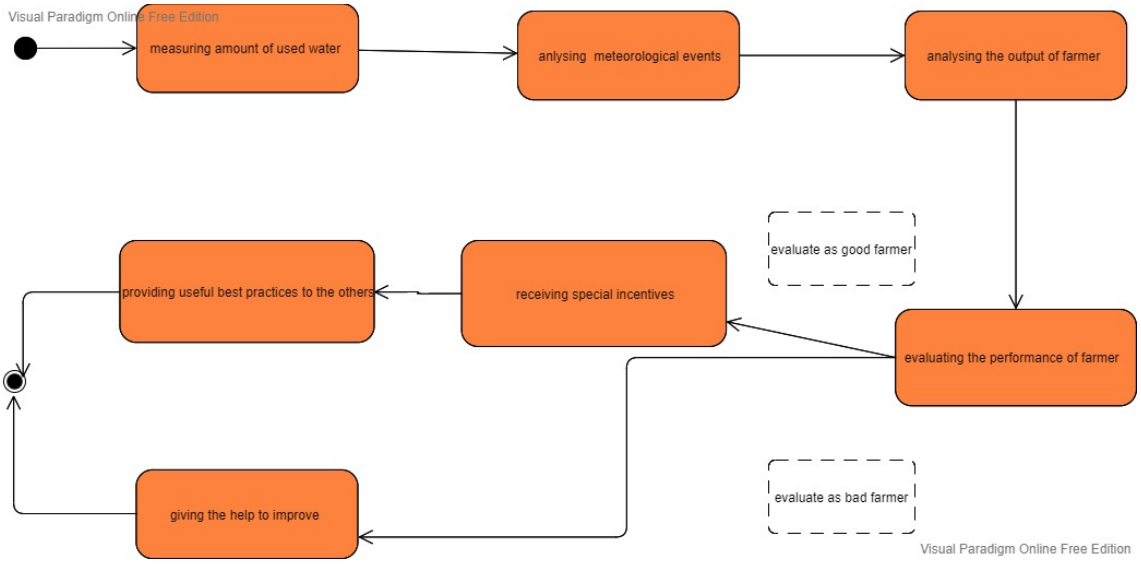


Figure 3: State Diagram 3 - evaluate the farmer by policymaker

in the third diagram (figure3), we model how the farmers are evaluated by policymakers by using analyzing environmental ones such as meteorological events or used water to identify good farmers to receive incentives or who need to be helped as they are performing particularly badly.

2.3 product function

Here we described some functions of the software. Be aware that some products functions are described also in other parts of this documents, particularly in the goal parts and in the parts of the functional requirements

give the personalized suggestion

DREAM, According to collected data such as location, type of production that farmers inserted in the initial steps, also by using the Weather forecast data collected from collected from meteorological stations and by applying the machine learning algorithm, give suggestions to use what crop to plant or specific fertilizers to use.

evaluate the outcome of the agriculturalists

the service helps policymakers access the performance of the farmer and determines the best and worst ones with collected environmental data, such as the used amount of water measured by irrigation system or humidity of soil obtained from a sensor located in each farmland. Also, consider the type of production and location of land or even what problem the farmer is faced with. The best farmers are encouraged and their opinion is used to improve the performance of others and help those who have not performed well.

communicate in the forum

DREAM has a function that lets farmers have communication in the forum and exchange their ideas, comments, problems with other farmers. we should consider that all of these members should be authenticated in our system, consider that farmers can see topics subscribing to them or create their own topic and they can put comments, also they have the ability to create new topics or delete an existing topics which have permissions.

2.4 user characteristics

the actor of the application are the following :

1. **farmer:** who has at least one farmland to produce crop and system help them to improve their capability. a farmer by inserting the type of production, location, Details at work, faced problem has interaction with the system and based on all of this information system besides of environmental factor improve their outcome.

2. **Policy Maker:** assess the performance of the farmer and determine the best and worst one with collected environmental data, such as the used amount of water that measure by irrigation system or humidity of soil that obtained from a sensor that located in each farmland

2.5 Assumption,dependencies and constraints

2.5.1 Domain assumption

D1	every farmer has at least one land
D2	every land has sensors to measure the humidity
D3	the internet connection work properly
D4	farmers insert all of the information about production correctly
D5	farmer insert all of the information correctly
D6	weather forecast ,base on the location is accurate
D7	amount of used water report correctly by irrigation system
D8	police maker asses the former based on true information
D9	just authenticated farmer can send message to forum
D10	amount of reported humidity is correct
D11	farmers insert problems about farm land,production
D12	in specific location ,there is just one farm land
D13	each user has a smartphone
D14	each smartphone has a GPS sensor integrated
D15	GPS is active while app is running
D16	GPS provide the exact location with an error of 5 meters at most

3 Specific Requirements

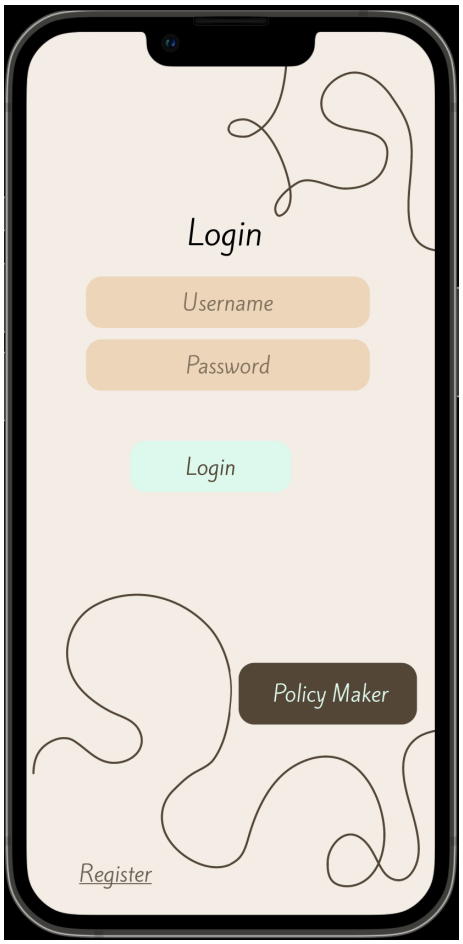
3.1 External Interface Requirements

3.1.1 User Interfaces

This application that we have should be able to receive information about each land and then track the progress of the farmers. We assume that each farmer signs up to our application and then inserts the data of the land, also we receive the forecasts of the meteorological data of Telangana. In these applications, farmers can create discussions and also ask about their problems. Policymakers can observe the farmland and choose the best and worst. There are notifications for farmers about the humidity, the amount of water that the land used, the forecast for weather in the app that helps them to improve their performance. You can see the mock up of our application in the following:

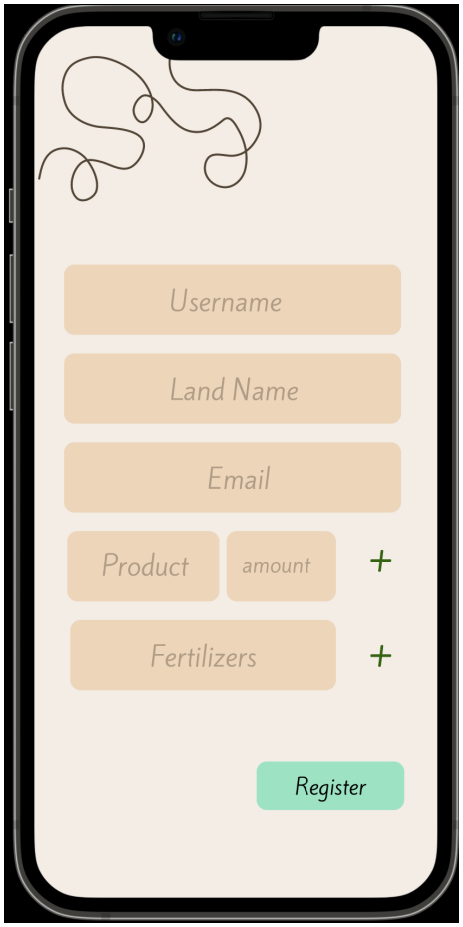
Figure 4: App Icon

(a) Opening page

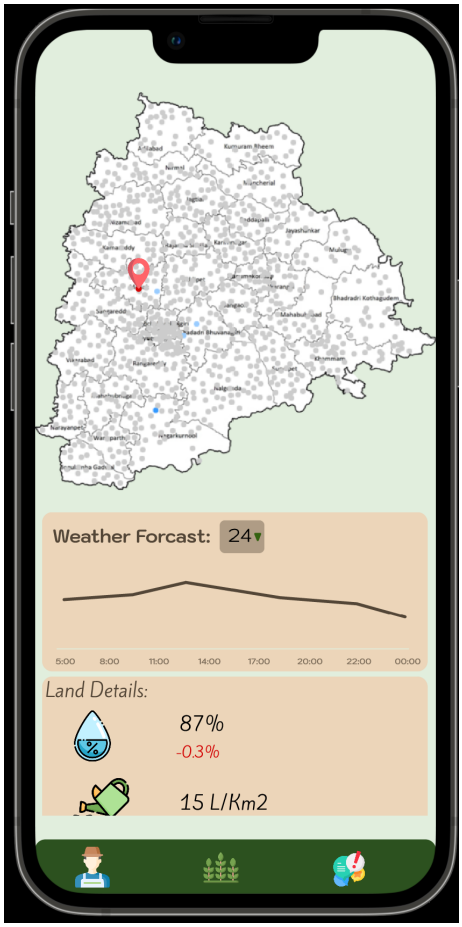


(b) sign in Farmers

(c) sign in policy makers



(d) sign up Farmers



(a) Farmers Main page

(b) Policy-Makers Main page

Figure 6: Mockup

3.1.2 Hardware Interfaces

In this system, farmers need to have a smartphone with GPS or a desktop computer that they can, register and add their land to the application and track the changes and also their suggestions. Also, their smartphone should contain a touch screen or keyboard which helps them to start a discussion in the application, ask their questions and share their experiences. The policymaker should also have a smartphone or a desktop computer that login into their account and observe the farmer's progress in the system. As we just have a mobile application for this program yet, they just need a smartphone.

3.1.3 Software Interfaces

Our system contains the following interfaces:
Telangana's map, We'll use some API that allows us to

- Locate the farmlands
- Getting information from water irrigation systems
- Getting data related to each land from sensors
- Getting meteorological information from Telangana's government

The application should receive information about each land and update the data of each profile in real-time.
Also, we need push notification services when:

- some urgent information change in the servers
- they receive a message from policymakers
- someone sends a message in the discussion that they start or join.

.

3.1.4 Communication Interfaces

The devices connect to DREAM via an internet connection.

3.2 Functional Requirements

3.2.1 List of Requirements

R1	Farmers certified with an authentication
R2	Policy-Makers certified with an authentication
R3	Policy-Makers should register to the application with extra mandatory fields
R4	Only Policy-Makers can observe all the farm land
R5	Only Policy-Makers can best and worst farmers
R6	Policy-Makers can choose farmers who received special incentives
R7	Policy-Makers must identify those farmers who need to be helped
R8	Farmers can be asked to provide useful best practices to the others
R9	Farmers must accept the GPS location for the application
R10	Farmers must insert their product to the system
R11	Farmers can visualize data relevant to them
R12	Farmers must insert the fertilizers that they are using to the system
R13	Farmers could see the list of forecast related to their location
R14	Farmers can insert any problem they face
R15	Farmers can start a discussion
R16	Farmers can request for help and suggestions by other farmers
R17	Farmers can join to an existing discussion
R18	Farmers can terminate their own discussion
R19	The system should receive the data related to meteorological short-term and long-term forecasts from Telengana's governments
R20	The system should suggest fertilizer based on the farmers locations and products
R21	The system should receive the data from sensors periodically
R22	The system should receive data from water irrigation system periodically
R23	The system should send notifications about the farm land to the farmers
R24	The system sends a notification to farmers if they receive new message
R25	The system sends a notification to farmers if the humidity if water decrease
R26	The system sends a notification to farmers if policymakers choose them as worst or best farmers

3.2.2 Mapping

Goal	Domain assumption	Requirements
G1	D1,D4,D5,D13	R1,R9,R10, R12,R13,R19,R21,R22
G2	D4,D5	R2,R3,R4, R10,R21,R22
G3	D1,D4,D5,D7,D8,D10	R2,R3,R4,R6,R10,R21,R22,R26
G4	D1,D4,D5,D7,D8	R2,R3,R4,R7,R10,R21,R22,R26
G5	D1,D2,D11	R1,R15,R16,R17, R24
G6	D1,D2,D11	R1,R17, R18
G7	D1,D9,D11,D13	R1,R15,R17,R24
G8	D1, D9	R1,R15,R17,R18
G9	D1,D2,D11	R1,R16,R18
G10	D9,D11	R1,R14
G11	D1,D9,D11,D13	R1,R9,R17,R18, R24
G12	D1,D4,D13	R8, R10,R23,R25

G1	Allow farmers to see the details of their farm lands
D1	every farmer has at least one land
D4	farmers insert all of the information about production correctly
D5	farmer insert all of the information correctly
D13	each user has a smartphone
R1	Farmers certified with an authentication
R9	Farmers must accept the GPS location for the application
R10	Farmers must insert their product to the system
R12	Farmers must insert the fertilizers that they are using to the system
R13	Farmers could see the list of forecast related to their location
R19	The system should receive the data related to meteorological short-term and long-term forecasts from Telengana's governments
R21	The system should suggest fertilizer based on the farmers locations and products
R22	The system should receive data from water irrigation system periodically
G2	Allow policy makers to see the details of different lands
D4	farmers insert all of the information about production correctly
D5	farmer insert all of the information correctly
R2	Policy-Makers certified with an authentication
R3	Policy-Makers should register to the application with extra mandatory fields
R4	Only Policy-Makers can observe all the farm land

R10	Farmers must insert their product to the system
R21	The system should receive the data from sensors periodically
R22	The system should receive data from water irrigation system periodically
G3	Allow policy makers to identify farmers who are performing well
D1	every farmer has at least one land
D4	farmers insert all of the information about production correctly
D5	farmer insert all of the information correctly
D7	amount of used water report correctly by irrigation system
D8	policy maker asses the former based on true information
D10	amount of reported humidity is correctly
R2	Policy-Makers certified with an authentication
R3	Policy-Makers should register to the application with extra mandatory fields
R4	Only Policy-Makers can observe all the farm land
R6	policy-Makers can choose farmers who received special incentives
R10	Farmers must insert their product to the system
R21	The system should receive the data from sensors periodically
R22	The system should receive data from water irrigation system periodically
R26	The system sends a notification to farmers if policymakers choose them as worst or best farmers
G4	Allow policy makers to identify farmers who need help
D1	every farmer has at least one land
D4	farmers insert all of the information about production correctly
D5	farmer insert all of the information correctly
D7	amount of used water report correctly by irrigation system
D8	policy maker asses the former based on true information
R2	Policy-Makers certified with an authentication
R3	Policy-Makers should register to the application with extra mandatory fields
R4	Only Policy-Makers can observe all the farm land
R7	Policy-Makers must identify those farmers who need to be helped
R10	Farmers must insert their product to the system
R21	The system should receive the data from sensors periodically
R22	The system should receive data from water irrigation system periodically
R26	The system sends a notification to farmers if policymakers choose them as worst or best farmers
G5	Allow farmers to create new forum

D1	The internet works properly
D2	Its better for users to have smartphones
D11	farmers insert problems about farm land,production
R1	Users certified with an authentication
R15	Farmers can start a discussion
R16	Farmers can request for help and suggestions by other farmers
R17	Farmers can join to an existing discussion
R24	The system sends a notification to farmers if they receive new message
G6	Allow farmers to join in discussions
D1	The internet works properly
D2	Its better for users to have smartphones
D11	farmers insert problems about farm land,production
R1	Farmers certified with an authentication
R17	Farmers can join to an existing discussion
R18	Farmers can terminate their own discussion
G7	Allow farmers to send message in discussions
D1	The internet works properly
D9	just authenticated farmer can send message to forum
D11	farmers insert problems about farm land,production
D13	each user has a smartphone
R1	Farmers certified with an authentication
R15	Farmers can start a discussion
R17	Farmers can join to an existing discussion
R24	The system sends a notification to farmers if they receive new message
G8	Allow farmers to create new forum
D1	The internet works properly
D9	just authenticated farmer can send message to forum
R1	Farmers certified with an authentication
R15	Farmers can start a discussion
R17	Farmers can join to an existing discussion
R18	Farmers can terminate their own discussion
G9	Allow farmers to leave in discussions
D1	The internet works properly
D2	Its better for users to have smartphones

D11	farmers insert problems about farm land,production
D11	farmers insert problems about farm land,production
R1	Farmers certified with an authentication
R16	Farmers can leave a discussion
R18	Farmers can terminate their own discussion
G10	Allow farmers to ask problems
D9	just authenticated farmer can send message to forum
D11	farmers insert problems about farm land,production
R1	Farmers certified with an authentication
R14	Farmers can insert any problem they face
G11	Allow farmers to answer to problems
D1	The internet works properly
D9	just authenticated farmer can send message to forum
D11	farmers insert problems about farm land,production
D13	each user has a smartphone
R1	Farmers certified with an authentication
R15	Farmers can start a discussion
R17	Farmers can join to an existing discussion
R18	Farmers can terminate their own discussion
R24	The system sends a notification to farmers if they receive new message
G12	Show farmers some personal suggestion
D1	every farmer has at least one land
D4	farmers insert all of the information about production correctly
D13	each user has a smartphone
R8	Farmers can be asked to provide useful best practices to the others
R10	Farmers must insert their product to the system
R23	The system should send notifications about the farm land to the farmers
R25	The system sends a notification to farmers if the humidity if water decrease

3.2.3 Use Cases

Use Cases Description

Name	Sign up
Actor	farmer
Entry conditions	Use the user has installed the application on her/his device
Event flow	1. Click on "Sign Up" button" 2. Fill all the mandatory fields and provide the necessary information 3. Click on "Confirm" button 4.the system saves the data
Exit conditions	The user successfully registered and now he's able to use the application.
Exceptions	1.the user is already signed up 2.the user didn't fill all of the mandatory fields with valid data 3.The username is already taken 4.the e-mail is already registered 5.All the exception are handled by notifying the user and taking him back to the sign up activity

Name	log in
Actor	farmer
Entry conditions	The user is previously successfully signed up and has the application installed in his/her device
Event flow	1. The user opens the application on his/device 2. He enters his credentials in the "Username" and "Password" fields of the home page of "DREAM" 3. The user clicks on the "Log in" button 4. The user is successfully logged in his/her "DREAM" the system automatically redirects him/her to the home page
Exit conditions	The user is successfully redirected to the home page
Exceptions	1. The user enters an invalid Username 2. The user enters invalid Password 3. All the exceptions are handled by notifying the user and taking him/her back to the login activity

Name	insert the initial information
Actor	farmer
Entry conditions	The user has already logged in
Event flow	1. The farmer select the farm land location 2. He enters his credentials in the “Username” ,“land name” ,"email" ,"product","amount"and"fertilizer" 3. the user click on the register button 4. The user is successfully logged in his/her “DREAM” the system automatically redirects him/her to the home page
Exit conditions	The user is successfully redirected to the home page
Exceptions	1. The user enters an empty Username field 2. The user enters an empty land name field 3. The user enters an empty Email field 4. the user enter empty product & amount field

Name	analyse the initial conditions
Actor	farmer
Entry conditions	The farmer insert completely initial information and report
Event flow	1.The system get initial information 2. It get weather forecast based on GPS and land location 3. system analyze the information
Exit conditions	make personalized suggestion
Exceptions	1. the forecast data is not accurate or invalid 2. input data has conflict 3. All the exceptions are handled by notifying the user and taking him/her back to the insert information page

Name	make personalized suggestion
Actor	farmer
Entry conditions	analysing of initial information has completed
Event flow	1. get the analysing of initial information 2. apply the ML/AL algorithm find best crop and fertilizer 3. show the result to farmer
Exit conditions	The user successfully registered and now he's able to use the application.
Exceptions	1.the user is already signed up 2.the user didn't fill all of the mandatory fields with valid data 3.The username is already taken 4.the e-mail is already registered 5.All the exception are handled by notifying the user and taking him back to the sign up activity

Name	Measure used water
Actor	Irrigation system
Entry conditions	Install irrigation system on the land
Event flow	1.The irrigation system measure used water in specific frequency-time
Exit conditions	Send measured data to access point
Exceptions	1. Measured used water is not possible 2. Irrigation system doesn't work correctly 3. All the exceptions are handled by sent notification to access point

Name	Measure soil humidity
Actor	Sensor
Entry conditions	Install sensor with the specific distance in depth of soil
Event flow	1.the installed sensors sense humidity
Exit conditions	Send measured data to access point
Exceptions	1. Measured used soil humidity is not possible 2. Sensors doesn't work correctly 3. All the exceptions are handled by sent notification to access point

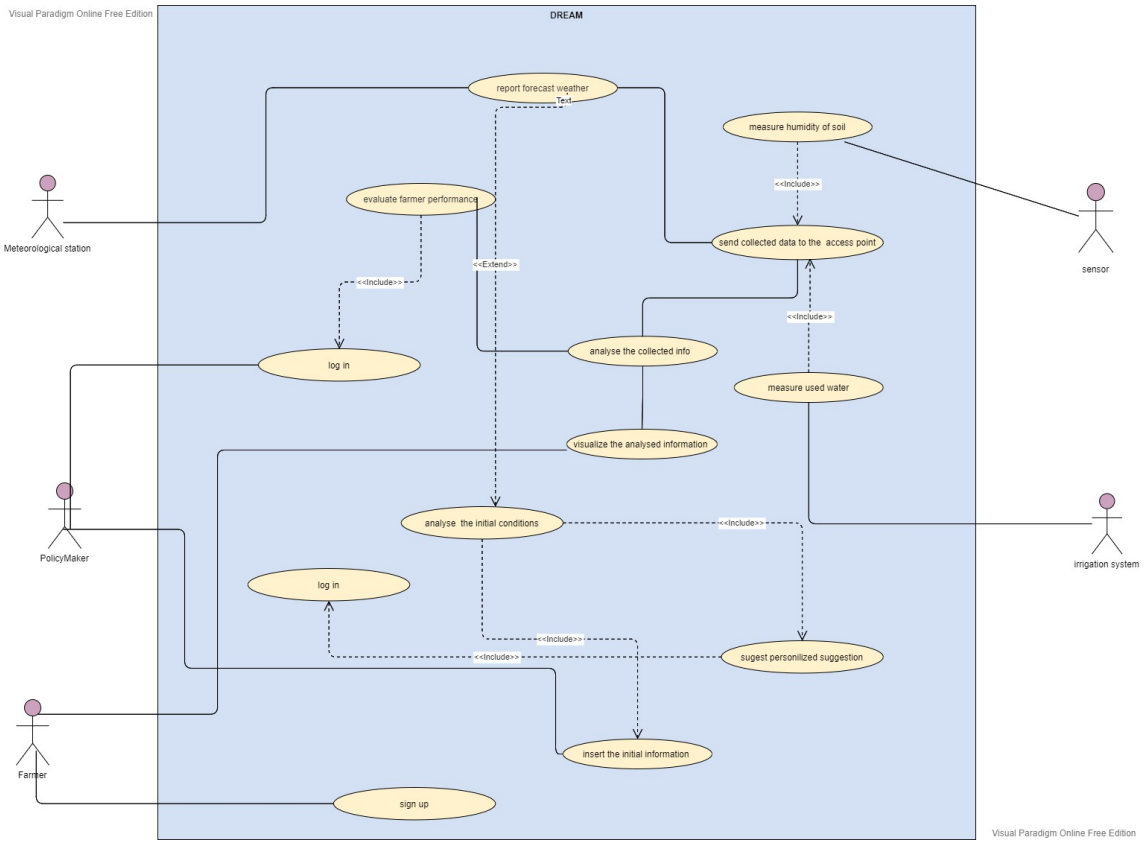
Name	send collected data to the access point
Actor	system
Entry conditions	measure soil humidity OR measure used water OR report forecast weather
Event flow	1. get the measured soil humidity from sensors OR get measured used water from irrigation system OR get weather forecast from meteorological station 2. collect information and prepare it to analyse
Exit conditions	analyse the collected info
Exceptions	1.The network doesn't work correctly 2.Sensors doesn't work correctly 3.Irrigation systems doesn't work correctly 4.Weather forecast doesn't report information correctly 5.All the exceptions are handled by sent notification to access point

Name	report forecast weather t
Actor	Meteorological station
Entry conditions	Meteorological station connects to the internet and GPS
Event flow	1.connect to the meteorological satellite 2. get the weather forecast 3.get the location information by GPS
Exit conditions	send collected data to the access point
Exceptions	1.The network doesn't work correctly 2.GPS doesn't work correctly 3. Meteorological station devices work correctly 5.All the exceptions are handled by sent notification to access point

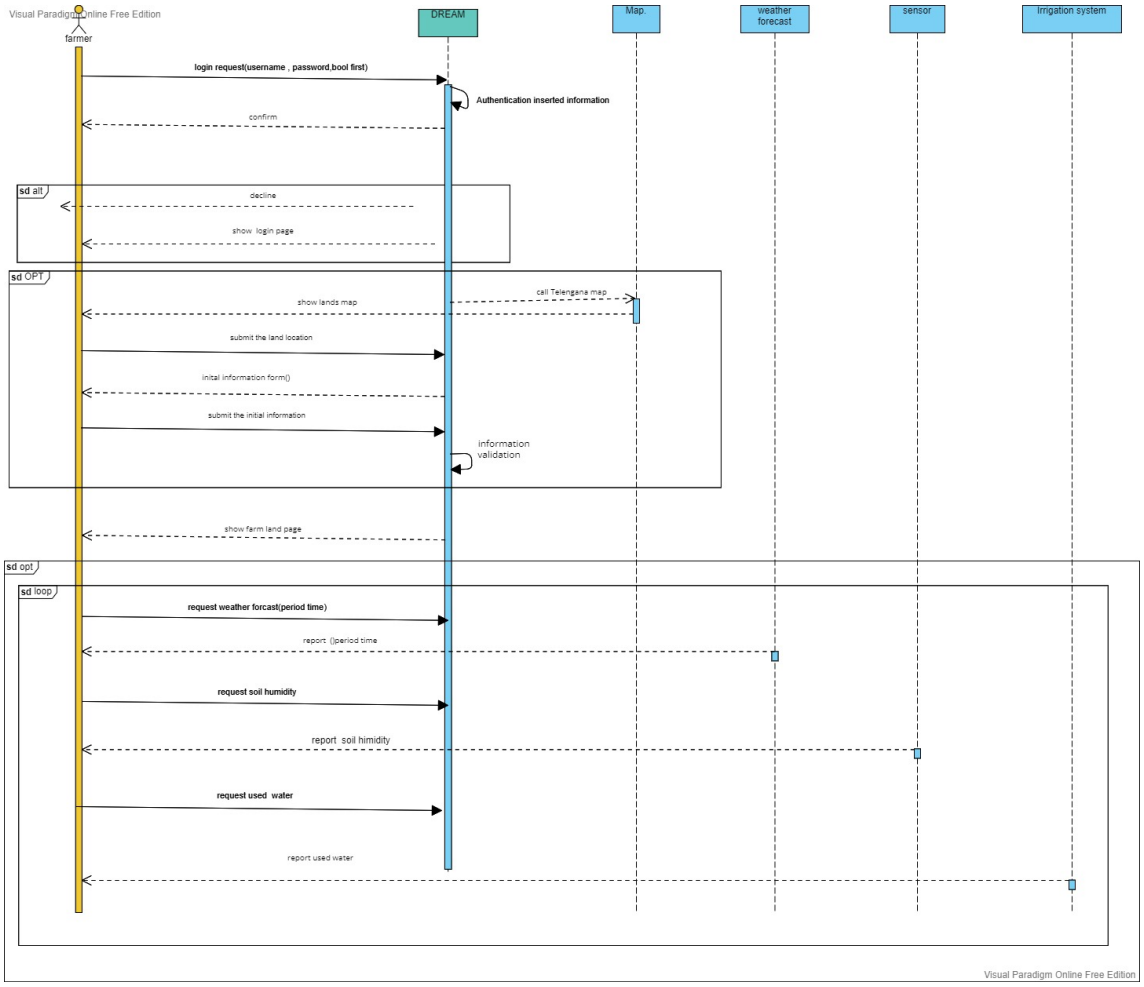
Name	evaluate farmer performance
Actor	policy maker
Entry conditions	analyse the collected info
Event flow	1.get analysing collected environment data 2. identify good and bad farmer based on analysed data
Exit conditions	after identifying the high and low performance farmer
Exceptions	1.analysed data is not accrued 2.analysed data has a shortage information 3.analysed data is not reliable and semantic meaning 5.All the exceptions are handled by sent notification to access point

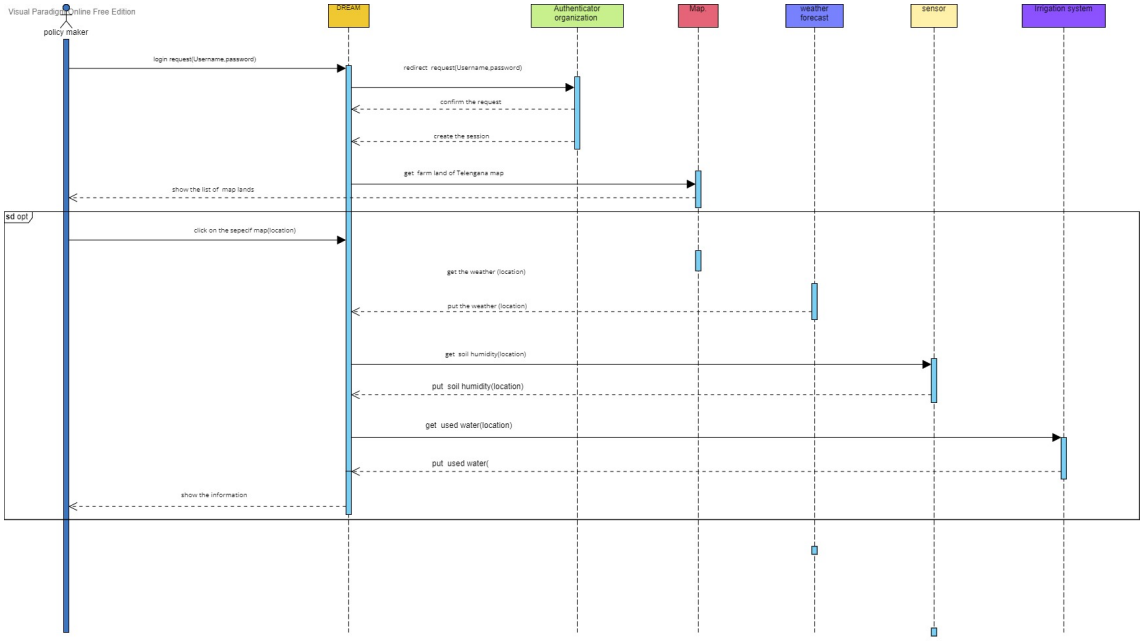
Name	visualize the analysed information
Actor	farmer
Entry conditions	analyse the collected info
Event flow	1.get analysed data 2. apply the specific algorithm to visualized the data
Exit conditions	result is ready to show to farmer
Exceptions	1. analyzed data doesn't have enough quality to analyzed that can be visual 2.data doesn't have semantic meaning

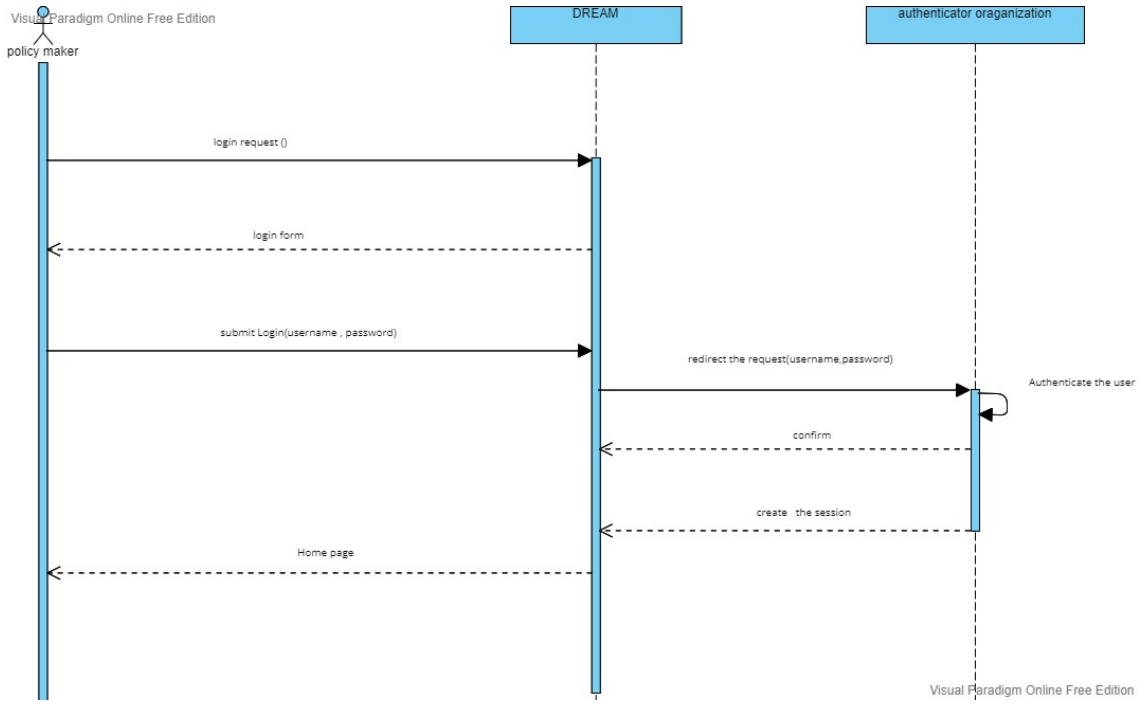
Use Cases Diagram

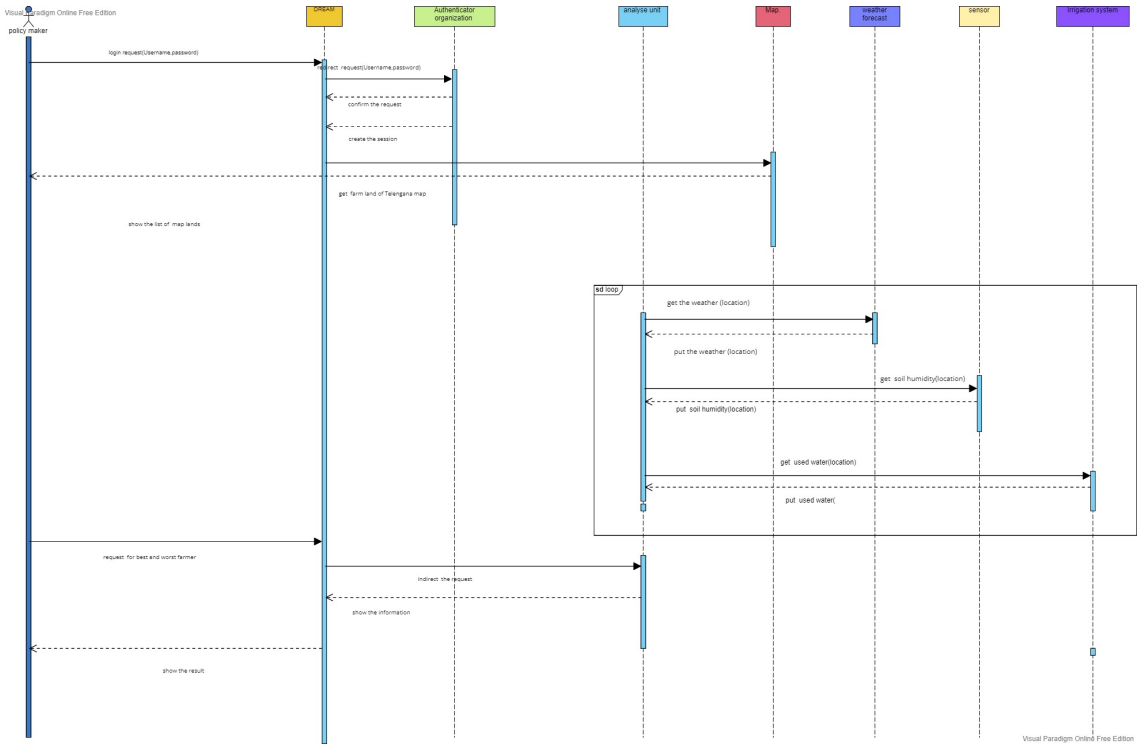


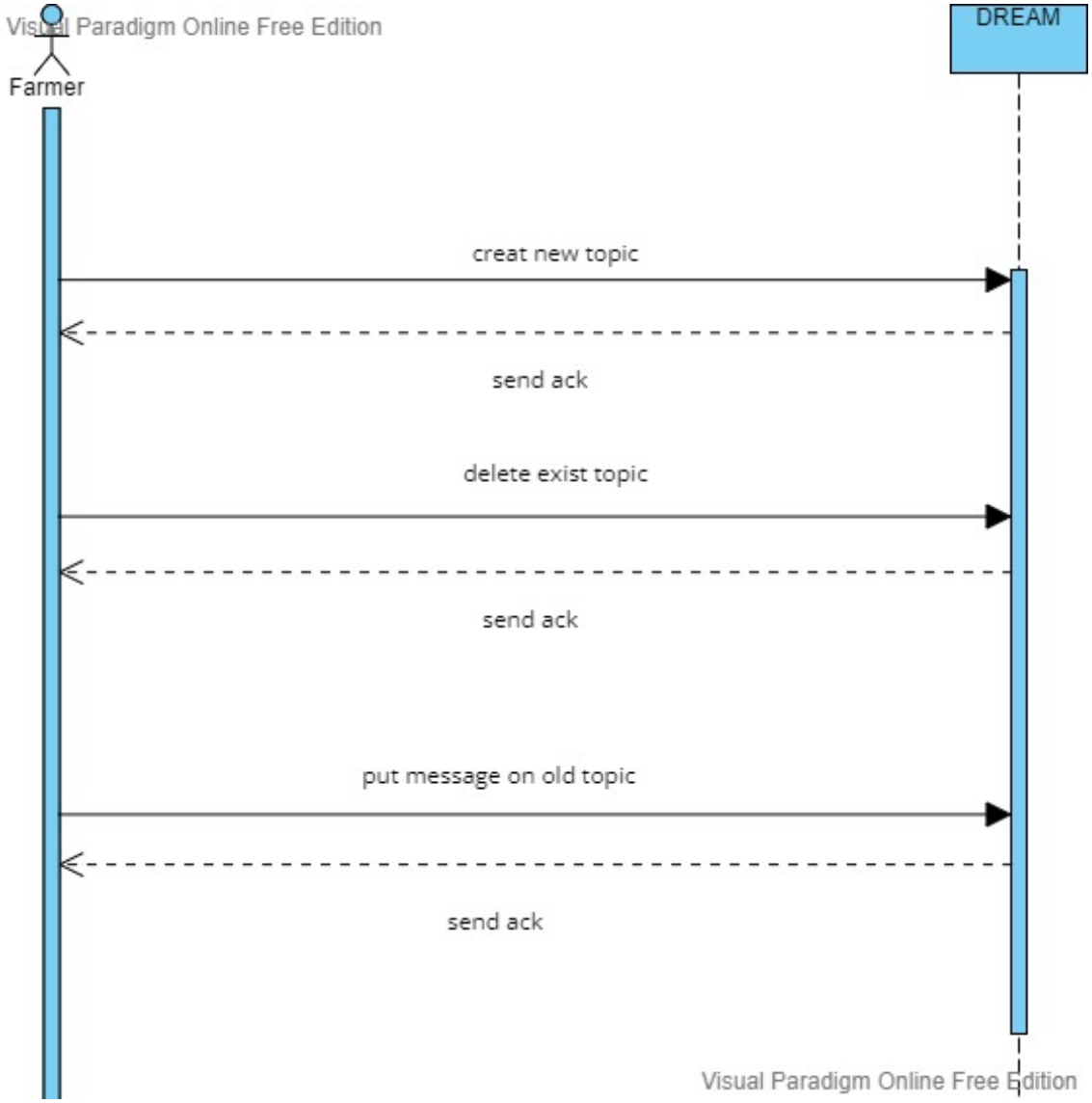
3.2.4 Sequence diagram








Visual Paradigm Online Free Edition
policy maker
DREAM
Authenticator organization
analyse unit
map
weather forecast
sensor
irrigation system
login request(Username,password)
redirect request(Username,password)
confirm the request
create the session
get farm land of Telengana map
show the list of map lands
sd loop
get the weather (location)
put the weather (location)
get soil humidity(location)
put soil humidity(location)
get used water(location)
put used water(
request for best and worst farmer
redirect the request
show the information
show the result
Visual Paradigm Online Free Edition


Visual Paradigm Online Free Edition
Farmer
DREAM
creat new topic
send ack
delete exist topic
send ack
put message on old topic
send ack
Visual Paradigm Online Free Edition

3.2.5 Scenarios

General
Nila is a farmer and as she was a teenager she start working on the land with her father. She starts working every day from 4:00 Am until 7:00 Pm on the land and check all the products qualities, the fertilizers, and the harvest time. So, she downloads this application, inserted the data of her land and the products, the beginning and ending time that she should implement or harvest the products and by using this app she can follow the progress of her land and see if there are some mismatches on the data that she got from the application and the products that she plants.

Ask for Assistant
Aadhya is a middleman farmer who is tired of a new kind of pest last month. After he sold the product of his last harvest and calculated his benefit, he figured out it was negative, and he didn't have any profit because his last products are attacked by the pests, and he had to pay lots of money to kill them and use more fertilizers to resuscitate the soil so he earn nothing. So, he downloads this application inserted the data of his land and by searching in the discussions he found some useful information from other farmers that had this problem and found a good solution to harness them. So he uses these devices and until now he gets a good result.

Be mechanized
Tom is a farmer who used traditional farming. His son is going to start working with him and he wants to improve the quality of the product and increase their performance. As their land is too wide they need some help, for example, more workers or using some new technologies. He decided to insert some sensors in their land to be more mechanized. These sensors can measure the humidity of the soil and also by using the DREAM application they can insert the data about their products and also track the humidity of their soil and product. Also by using the data that we have from the weather forecast and personalizing suggestions, they decrease the amount of water that they need in irrigation or the number of fertilizers that they need. By using this approach Krishna's son could increase their performance by about 10% in just 2 months after using them and he hopes that they can increase the performance by about 20

faster comments
Sarah is a policymaker in Telengana and she tracks the performance and quality of products for about 2 years. Now she figured out there was some aspect that she didn't consider these years because of the number of farmers and different features that she had to track. Also, most of the comments that she sent for the farmers were useless because she couldn't track their progress in real-time, and when she sent them because of changes that they had for example changing season most of them wasn't useful anymore. She didn't have any platform that shows the progress of a farmer or farmland for a period of time and she had to compare lots of paper things and it is obvious that she drop some points. After she started using DREAM she can observe the progress of farmers faster, send suggestions sooner so she got better results from her observation, and she saw the comments that she send for the farmers are more useful than before.

Best farmer
Krishna one of the policymakers should evaluate the result of the farmers, she wants to know who has the best and worst approaches according to the information registered in the system, she logs in the application with official mail after the authentication, she goes to the rank section and sees the result.

Commenting
Moila is an inexperienced farmer who wants to start farming on her land, she wants to use the experience of others to improve her knowledge so she decided to subscribe to the "how to get a crop from the land" and read the comments also she post her questions

3.3 Performance Requirements-Non-Functional Requirements

The system should be able to guarantee a safe and consistent connection. Based on the population of Telangana in the worst case, the application should handle about 360,000 users that is the number of Telangana's farmers. Also because most of the farmers work after the sunrise until the sunset so we must expect some burst in our requests.

3.4 Design Constraints - Non-Functional Requirements

3.4.1 Standards compliance

The code should follow the requirements contained in this document. Furthermore, its comments should be clear and focused.

3.4.2 Hardware limitations

The software application requires a mobile device able to capture the position. In alternative, the authorities, can use a computer to observe the progress of the farmers. Both devices can be able to send data to the software via internet connection.

3.4.3 Any other constraint

The user has the limitation of being the owner of more than one farmland simultaneously.

3.5 Software System Attributes - Non-Functional Requirements

3.5.1 Reliability and availability

The application provides a reliable service in which individual farmers can easily log in and see the data related to their lands. Furthermore, it warranties that the chain of custody of the information coming from the farmers, sensors, and other systems is never broken, and the information is never altered. This would provide a secure and reliable system.

The application must offer availability in the order of 99% granting with at least 20 hours in a day. The lack of service must be minimal. It's better to maintain system at nights and before the sunrise because farmers mostly do not use this application at night. Even at night, we could put some resources at sleep mode to reduce the cost of application.

3.5.2 Security

The users' location and personal data is a very important data and must be encrypted so this part must be more secure than other data stores in the application. Moreover, in case of password recovery this should never be sent in clear. The system must be behind a proxy servers (like cloudflare) to prevent the DDoS attacks

3.5.3 Maintainability

The application must keep a service log in order to fix bugs more easily. The app should be developed using a micro-service approach, so adding new functions shouldn't't require to change the previous code, and we can have different instance of services in different time of day. This application needs infrastructure like kubernetes to orchestrate the containers.

3.5.4 Portability

The user side must be available in different platform like Android, iOS and Web. The back-end side is better to be deployed with docker to easily deploy on any server.

3.5.5 Scalability

As Climate change continues to be a real and potent threat to the agriculture sector and food demand is expected to increase anywhere, so we will predict that so many farmers will join this app to track their progress, so the system must have a good infrastructure that could balance the loads and easily we must add new resources to it.

3.6 Additional Specifications

3.6.1 Mandatory Fields

At the moment of the registration, an authority will have to compile the following mandatory fields:

- Exact location of each farm land
- The products that each farmer has in his land
- Phone number
- Official Email for policymakers

4 Formal Analysis Using Alloy

4.1 Alloy Code

```
sig Float{}
sig Email, Username, Password, Id{}
sig Integer{}
sig Str{}
sig Fertilizer{}
sig Status{}
one sig Active extends Status {}
one sig Closed extends Status {}
sig City{}
sig Region{
        city: some City
}
sig State{
        reg: some Region
}
sig Date        {
        timeStamp: one Integer
}
sig Message{
        date: one Date,
        writer: one Farmer,
        content: one Str
}
sig Sensor{
        id: one Id,
        measur : one Float,
        location: one Location,
}

sig WaterIrr{
        id: one Id,
        measur: one Float,
        location: one Location,
}
```

```
sig Weather{
        date: one Float,
        temp: one Integer,
        location: one Location
}
sig Crop{
        name: Str,
        weight: Int
}
sig Location{
        state: one State,
        long: one Float,
        lat: one Float,
}
sig Discussion{
        topic: Str,
        participant: some Farmer,
        owner: one Farmer,
        status: Active,
        msg: some Message
}
sig Problem{
        topic: Str,
        participant: set Farmer,
        owner: one Farmer,
        status: Active,
        msg: set Message
}
sig Farmland{
        farmer: one Farmer,
        crops: set Crop,
        fertilizers: set Fertilizer,
        location: one Location,
        humidity: one Float,
        water: one Float,
        weather: one Weather
```

```
}
sig Farmer{
        email: one Email,
        username: one Username,
        password: one Password,
        problems: some Problem,
        discussions: some Discussion,
        land: Farmland,
        rate: Int
}{ rate >0 and rate <=5}
sig PolicyMaker{
        username: one Username,
        email: one Email,
        farmers: set Farmer,
        bestF: set Farmer,
        worstF: set Farmer
}{ worstF != bestF }
////////////////////////////////  FACT
  ////////////////////////////
fact disEmail{
  no disj f1,f2: Farmer | f1.email = f2.email
}
fact disEmailP{
  no disj p1,p2: PolicyMaker | p1.email = p2.
    email
}
fact disUsername{
  no disj f1,f2: Farmer | f1.username = f2.
    username
}
fact DistinctWaterIrrigationSystems{
        no disj w1, w2: WaterIrr | w1.location =
          w2.location
}
fact DistinctMessagesDiscussion{
        no disj d1, d2: Discussion | d1.msg = d2.
          msg
```

```
}
fact DistinctMessagesProblem{
        no disj p1, p2: Problem | p1.msg = p2.msg
}
//
fact MappingHumidity{
        all fl: Farmland, s:Sensor |  fl.humidity
           = s.measur iff fl.location = s.
          location
}
fact MappingWater{
        all fl: Farmland, w:WaterIrr |  fl.water
          = w.measur iff fl.location = w.location
}
fact disjointIdS{
        no disj s1, s2: Sensor| s1.id = s2.id
}
fact disjointIdW{
        no disj w1, w2: WaterIrr| w1.id = w2.id
}
//Assign each farmeland to exacly one farmer
fact OneOwnerForDisscussion{
        all d: Discussion | one f: Farmer | d.
          owner = f
}
// Only Farmers can participate in a problem or
  discussions iff they are in participant list
fact OnlyParticipantCanParticipateInDiscussion{
        all f: Farmer | all d: Discussion| d in f
          .discussions implies f in d.participant
}
fact OnlyParticipantCanParticipateInProblem{
        all f: Farmer | all p: Problem| p in f.
          problems implies f in p.participant
}
// Writer should be part of participants
fact MappingParticipantToWriters{
```

```
        all p: Problem | one f:Farmer | f in p.
          msg.writer iff f in p.participant
}
fact MappingParticipantToDiscussion{
        all d: Discussion | one f:Farmer | f in d
          .msg.writer iff f in d.participant
}
fact BestFarmerInTheList{
        all p: PolicyMaker | all f: Farmer| f in
          p.bestF iff f in p.farmers
}
fact distinctWorstandBest{
        all p: PolicyMaker | p.worstF != p.bestF
}

fact SelectWorstFarmer{
        all p: PolicyMaker | all f: Farmer | p.
          worstF = f iff f.rate < 2
}
fact SelectWorstFarmer{
        all p: PolicyMaker | all f: Farmer | p.
          bestF = f iff f.rate > 4
}
//////////////////////////    Pred
  ////////////////////////////////////

//Create Farm Land
pred InitialInformation(fl: Farmland, f: Farmer,
  c: Crop, fr: Fertilizer, l: Location, h: Float,
   w: Float, wt: Weather) {
        fl.farmer = f
        fl.crops = fl.crops +c
        fl.fertilizers = fl.fertilizers + fr
        fl.location = l
        fl.weather = wt
        fl.water = w
}
```

```
run InitialInformation

// Start Discussion
pred CreateForumD(d: Discussion, f: Farmer, t:
   Str, m: Message){
         d.owner = f
         d.participant = d.participant + f
         d.topic = t
         d.msg = m
}

run CreateForumD

//Ask Problem
pred CreateForumP(p: Problem, f: Farmer, t: Str,
   m: Message){
         p.owner = f
         p.participant = p.participant + f
         p.topic = t
         p.msg = m
}

run CreateForumP

// Join to existing Problem
pred JoinForumP(p: Problem, f: Farmer, fo: Farmer
   , t: Str, m: Message){
         p.owner = fo
         p.participant = p.participant + f
         p.topic = t
         p.msg = m
}
run JoinForumP

// Select Best and Worst Farmers by PolicyMakers
pred SelectBestFarmer(p: PolicyMaker, bf: Farmer,
    wf:Farmer, f: Farmer, u: Username, e:Email){
```

```
        p.username = u
        p.email = e
        p.bestF = bf
        p.worstF = wf
        p.farmers = f
}

run SelectBestFarmer
```

4.2 Meta Model

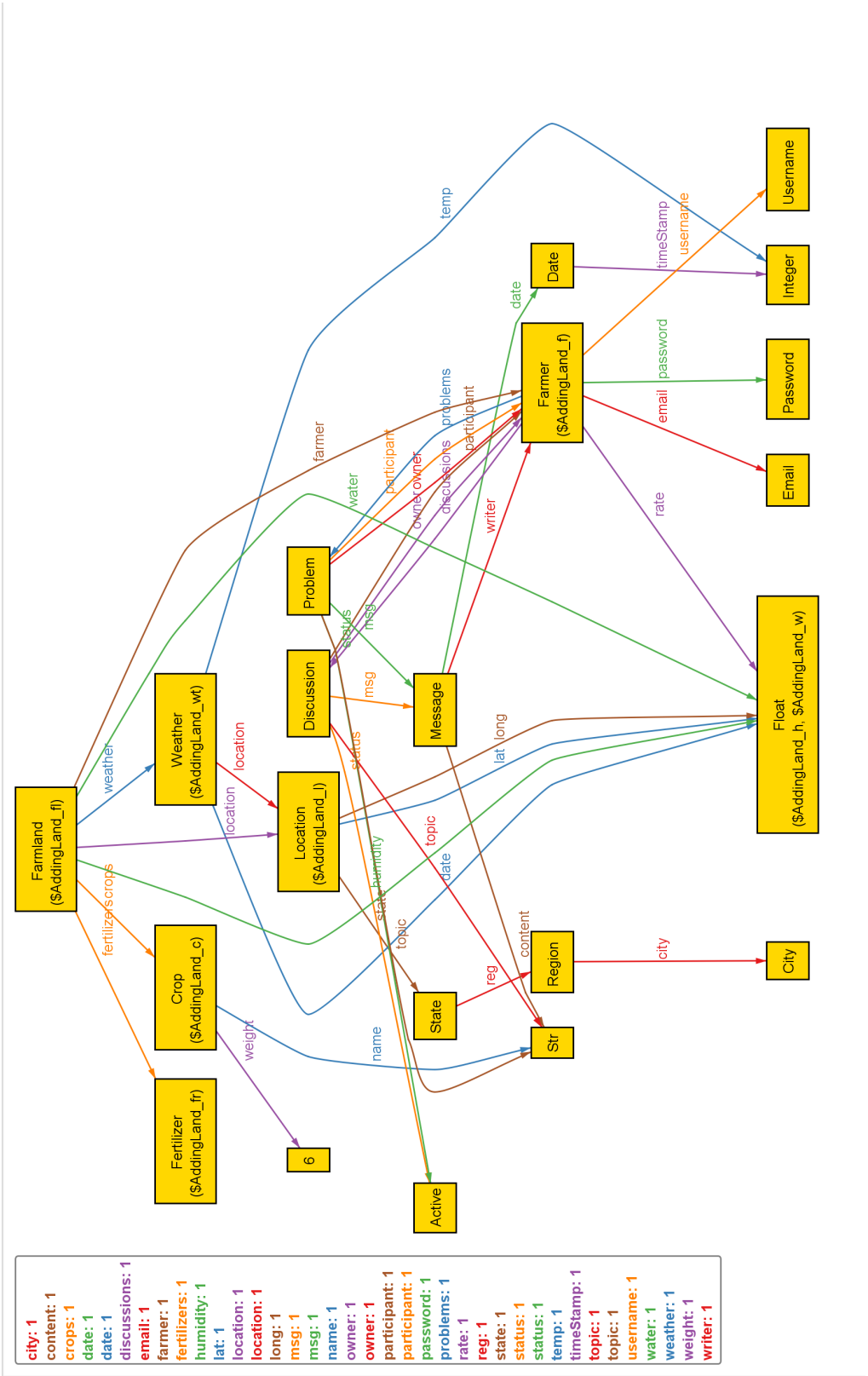


Figure 7: Insert Land

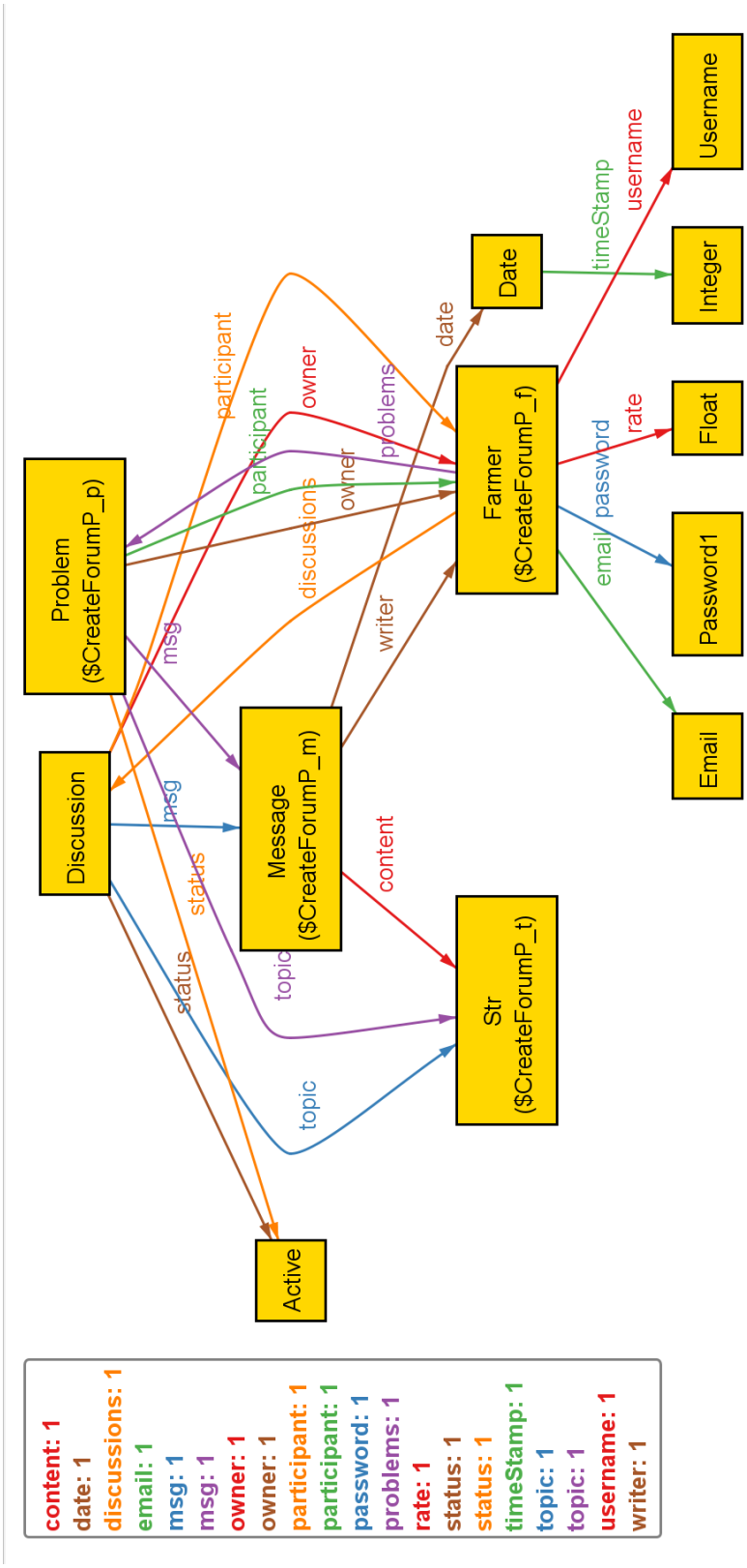


Figure 8: Create Problem

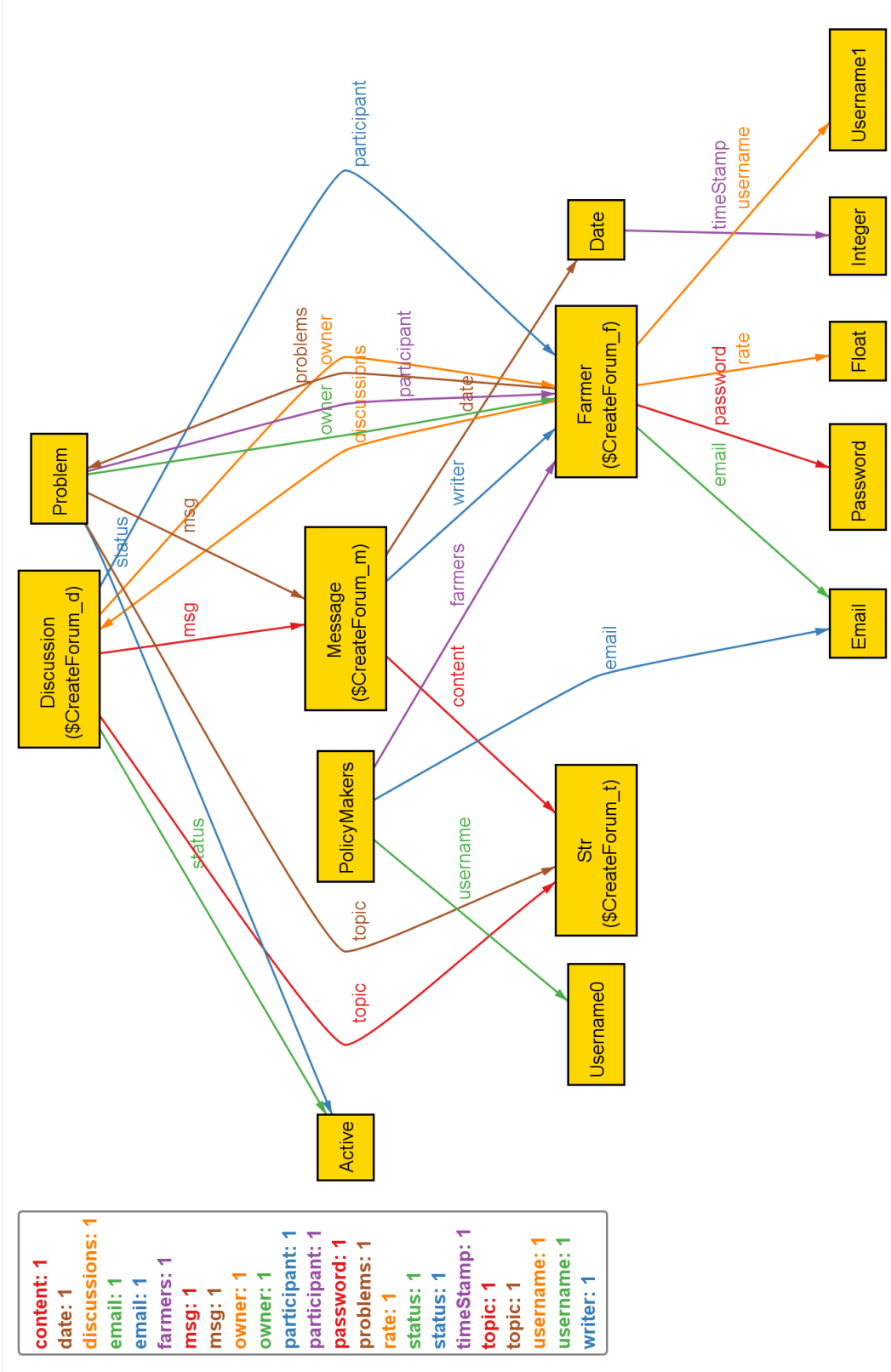


Figure 9: Create Discussion

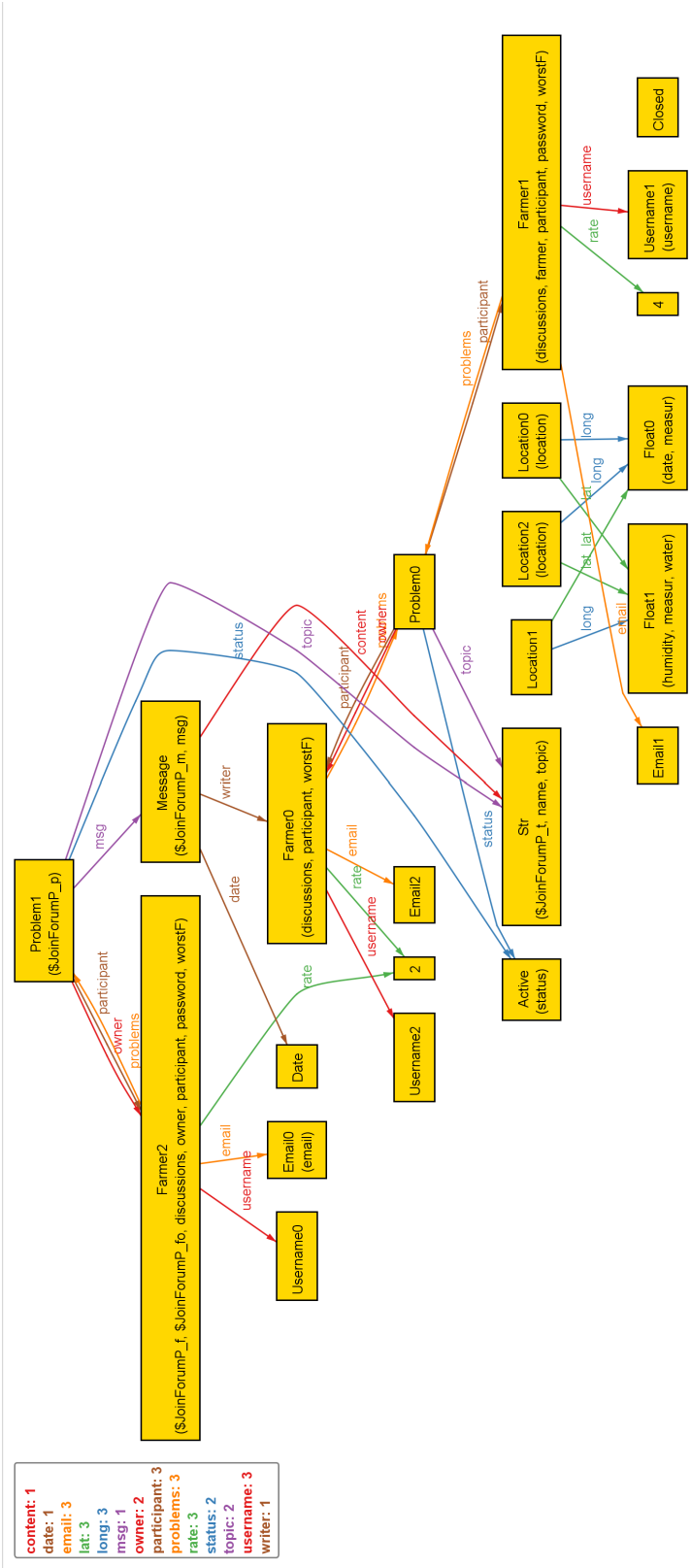


Figure 10: Join Forum

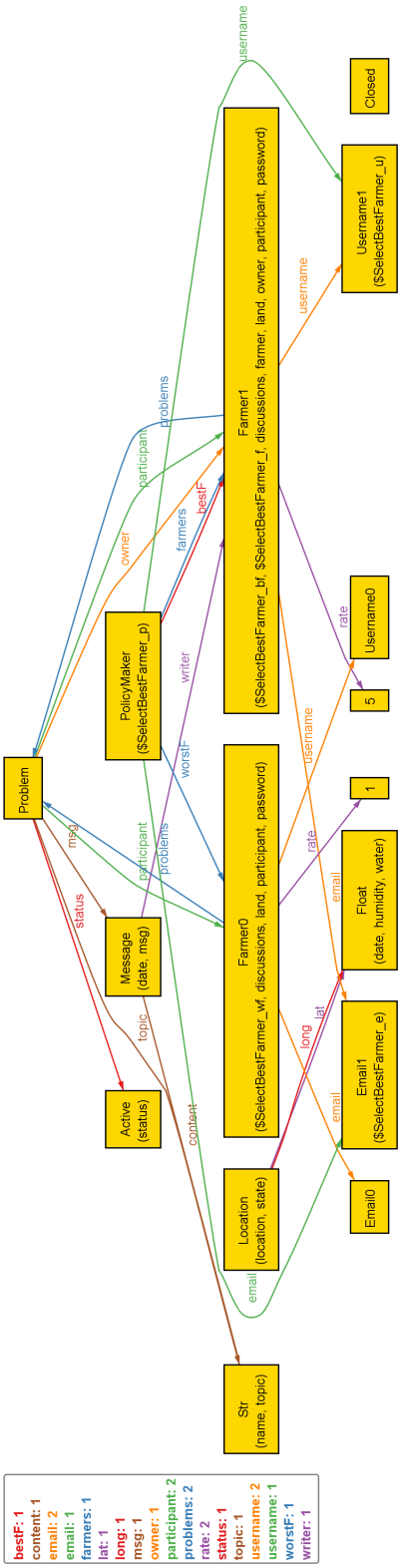


Figure 11: Select Best and Worst Farmers

4.3 Results of Predicates

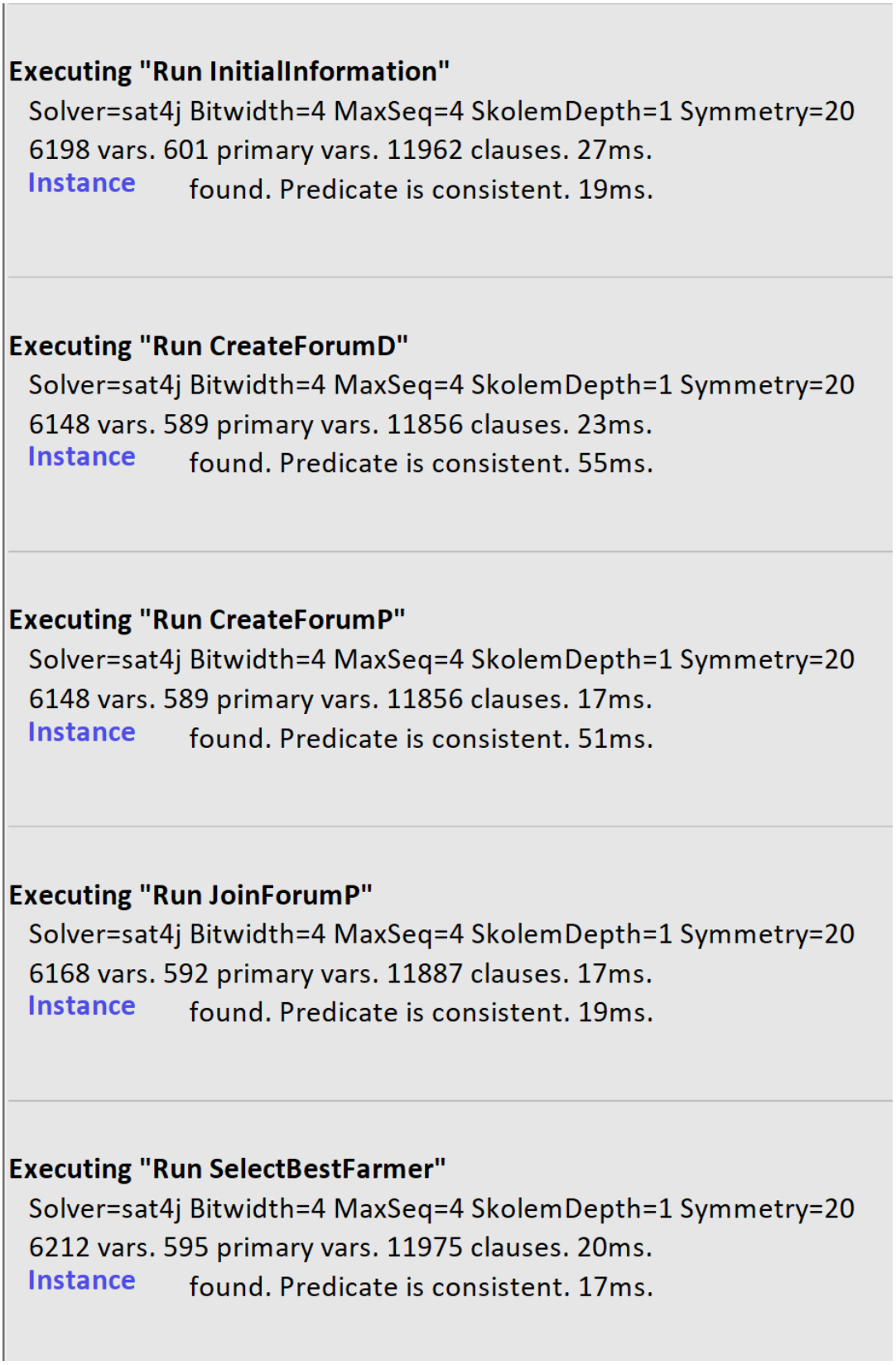


Figure 12: Executing Results

5 Effort spent

Farimah Anvari:

Topic	Hours (hh:mm)
First meeting	2:30
Discussion on first part	4:00
Goals	2:00
World & Shared phenomena	2:00
Discussion on UML Description & State charts	2:00
User characteristics	2:00
User Interface Design	12:00
List of Requirements	3:00
Mapping	4:00
Scenarios	1:00
Software System Attributes	3:00
Design Constraints NF Requirements	2:00
Alloy	15:00
Document Revision	2:00

Sajedeh Firouzizadeh:

Topic	Hours (hh:mm)
First meeting	1:30
Discussion on first part	3:00
UML Description & state charts	9:00
Product functions	5:00
Domain assumptions	2:00
mapping	4:00
Use cases	2:00
Sequence Diagrams	4:00
Scenarios	2:00
Alloy	5:00
Document Revision	2:00

6 References

[1] Alloy Code writes using Alloy Tool
[2] Diagrams are drawn by using Visual Paradigm **https://online.visual-paradigm.com/**