



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

DREAM - Data-dRiven PrEdictive FArMing in Telengana

RASD
SOFTWARE ENGINEERING 2

TEAM MEMBERS:

Simone Brunello - 10831750
Nicholas Nicolis - 10867841

Academic Year: 2021-22

Contents

Contents	i
1 INTRODUCTION	1
1.1 Purpose	1
1.2 Scope	1
1.2.1 World phenomena	2
1.2.2 Shared phenomena	2
1.2.3 Goals	4
1.2.4 Definitions	5
1.2.5 Acronyms	5
1.2.6 Abbreviations	6
1.3 Document Structure	6
1.3.1 Section 1	6
1.3.2 Section 2	6
1.3.3 Section 3	6
1.3.4 Section 4	7
1.3.5 Section 5	7
1.3.6 Section 6	7
2 OVERALL DESCRIPTION	8
2.1 Product perspective	8
2.1.1 Scenarios	8
2.1.2 Static Information Model	11
2.1.3 Dynamic Class Behaviour Models	13
2.2 Product functions	18
2.3 User characteristics	19
2.4 Assumptions	20
3 SPECIFIC REQUIREMENTS	21

3.1	External Interface Requirements	21
3.1.1	User Interfaces	21
3.1.2	Software Interfaces	22
3.1.3	Communication Interfaces	23
3.2	Functional Requirements	23
3.2.1	List Of Requirements	23
3.2.2	Mapping Requirements on Goals	25
3.2.3	Mapping Domain Assumptions on Goals	27
3.2.4	Mapping explicit table	27
3.2.5	Use Case Diagrams	40
3.2.6	Sequence Diagrams	54
3.3	Performance Requirements	63
3.4	Design Constraints	63
3.4.1	Standards compliance	63
3.4.2	Hardware limitations	64
3.5	Software System Attributes	64
3.5.1	Reliability	64
3.5.2	Availability	64
3.5.3	Security	64
3.5.4	Maintainability	65
3.5.5	Portability	65
4	FORMAL ANALYSIS USING ALLOY	66
4.1	Alloy Model	66
4.1.1	Results	66
4.1.2	Graphic Model	66
4.2	Code	68
5	EFFORT SPENT	71
6	REFERENCES	73
6.1	Reference Documents	73
6.2	Reference Sites	73
7	CORRECTIONS	74

1 | INTRODUCTION

1.1. Purpose

The Indian's population counts 1,8 bilions people and their main source of income comes from the agriculture sector. More precisely 80% of farmers own less than 2 hectares of cultivable land. Nowadays, as a consequence, an important part of the population is already below the poverty threshold, and if nothing will change during the next decades this condition will become even worst.

In addition to this, the productivity will decrease due to many factors: the increasing demand of food caused by the growth of the population and the aggravation of climate conditions as a consequence of irresponsible behaviour towards the environment.

An improvement in the communication and data management system could potentially avoid and prevent production problems and lack of food for the poorest part of the population.

Our main goal is to help both farmer and Telengana's policy makers to improve their communication with the focus in quality and quantity of information. Farmers need to receive more technical and environmental advice in order to develop their working skills and consequently their productions. On the other hand policy makers should have enough data and feedbacks about farmer's performances in order to improve their economical policies and the country general welfare.

1.2. Scope

In order to manage more efficiently the communication between farmers and policy makers our DREAM application will provide an easy way to access the system which will make available to all different users a dedicate set of tools and information.

Farmers will be able to monitor weather conditions, crops and fertilized suggestions. They will have the possibility to send direct requests to expert or other farmers in order to receive advice. The ease of communicating their production data and problems will be

a key point.

Telengana's policy makers will be able to monitor farmers performances and decide if current policies are providing good results. They will also be supported in the visualization of critical situations in order to intervene in advance.

1.2.1. World phenomena

World phenomena	Description
WP01	Weather changes
WP02	A rare climatic event occurs
WP03	Farmer cultivates his land
WP04	Farmer harvests
WP05	A new law concerning agriculture is published

Table 1.1: Table of World phenomena

1.2.2. Shared phenomena

Shared Phenomena	Description	Control
SP1	The farmer checks the weather map of the area where he belongs.	world controlled
SP2	The farmer reads a news about an important climatic event.	machine controlled
SP3	The farmer sends an ticket request.	world controlled
SP5	The farmer gets an answer to a ticket he sent.	machine controlled
SP6	The farmer reads information in the archive of the software.	world controlled
SP7	The farmer writes in the forum.	world controlled.
SP8	The farmer is notified of new replies of a forum's thread.	machine controlled
SP9	The farmer reads comments in a forum's thread.	world controlled

SP10	The farmer is able to send a report about his/her production.	world controlled
SP11	The farmer is notified if the policy maker answers to his/her report.	machine controlled
SP12	The farmer is able to read the policy maker report about his/her production.	world controlled
SP13	The policy maker is able to analyze the weather map of his/her working area.	world controlled
SP14	The policy maker gets notification about the main climatic events.	machine controlled.
SP15	The policy maker is able to read the news about the main climatic events.	world controlled
SP16	The policy maker is able to watch and analyze the statistics.	world controlled
SP17	The policy maker is able to access to the contact list.	world controlled
SP18	The policy maker receives notifications concerning new tickets addressed to him/her.	machine controlled
SP19	The policy maker is able to answer to the tickets.	world controlled
SP20	The policy maker gets a notification about a farmer's report.	machine controlled

Table 1.2: Table of Shared Phenomena

1.2.3. Goals

Goals	Description
G1	Allow farmers to easily check weather condition.
G2	Allow farmers to have technical and personalized advices from other farmers or experts of the field.
G3	Allow farmer to ask for help to the government.
G4	Improve the communication between farmers through a forum.
G5	Improve data communication between farmers and government.
G6	Allow the government to improve the analysis and the sharing of important data concerning agriculture.
G7	Allow the government to have specific data about farmers.
G8	Allow the policy makers to easily recognise critical and virtuous situations.

Table 1.3: Table of Goals

1.2.4. Definitions

Definition	Description
Farmer	A farmer registered in the system.
Policy Maker	An authorized user who works for the government.
Farmer Report	Document containing information about production, expenses and incomes of farmers.
Policy Maker Report	Document containing an economical/production analysis towards a specific farmer. It includes also management suggestions about what they could do to improve or to maintain the situation.
Archive	A collection of technical information about plants, fertilizers, practical tools and techniques.
Forum	An application's section where users can hold conversations in the form of posted messages.
Ticket	A special message which farmers can use in order to directly contact policy makers, usually to ask help.
Mockup	A realization for illustrative or merely display purposes of the application's UI.

Table 1.4: Table of Definitions

1.2.5. Acronyms

Acronyms	Description
GPS	Global Positioning System
UML	Unified Modeling Language
WP	World Phenomena
SP	Shared Phenomena
G	Goal
D	Domain Assumption
R	Requirement
UI	User Interface
API	Application Programming Interface

Table 1.5: Table of Acronyms

1.2.6. Abbreviations

Abbreviations	Description
alt	alternative
opt	optional

Table 1.6: Table of Abbreviations

1.3. Document Structure

1.3.1. Section 1

Introduction about the purpose and scope of the system. Discussion of the main world and shared phenomena concerning our application's domain and goals. Furthermore we sum up all the definitions and abbreviations in order to have a better comprehension of the following chapters.

1.3.2. Section 2

Descriptions of different scenarios which illustrate multiple interactions that the application could face, then there is a structural description of the system represented by the presence of various graphs such as class diagrams, statechart, sequence diagrams and activity diagrams with all of their main characteristics. After this in the product functions section there are multiple descriptions of all the possible functionalities which are present inside the application. In the last part there is a list of the domain assumptions and the characteristics of the users who will exploit the application.

1.3.3. Section 3

This is the main part of the document, at the beginning some general mockups are showed and there is a description of the software and communication interfaces. After this it's presented the list of the requirements along with their description and some tables which map goals and domain assumptions with the respective requirements. There is also a part with all the use case diagrams followed by their descriptions and the corresponding sequence diagrams. At the end a list of the performance requirements and design constraints of the system are presented.

1.3.4. Section 4

This section contains a formal description of some important parts of the system using the Alloy language. Some charts and the code close this section.

1.3.5. Section 5

This section is created to show how much time every student spent in the multiple parts of the documents.

1.3.6. Section 6

This sections is made in order to point out all the references and tools used during the creation of this document.

2 | OVERALL DESCRIPTION

2.1. Product perspective

2.1.1. Scenarios

Registration Yamir is a farmer in the province of Hyderabad, he saw in the newspaper the new project DREAM proposed by the government to help the agriculture's economy of the country. So he decided to register to the program. After he opens the app the system asks him to insert various information about his personal status, his farmlands and how many collaborators he has. Yamir fills all the text boxes and clicks confirm. Then he receives a confirmation email from DREAM in order to conclude the registration. The system shows to Yamir that the registration has been completed.

Checking the weather Anirudh is a farmer who lives in the outskirt of Warangal, he has bought a new piece of land where he would plant a new sort of vegetable. In order to check if this is the right time to plant the seeds he needs to know as soon as possible if there will be a rainy or a sunny period in the next weeks. For this purpose Anirudh uses the application DREAM to check the weather conditions in the next period, the DREAM weather-forecast allows the farmer to check lot of data such as the humidity in the air, the amount of water that will fall down and the probability of rain in a certain day. This is extremely helpful in order to understand if this is the right time to plant a new type of vegetable or not, because the rain water is vital for the beginning of its life-cycle.

Looking for plants information Shyla have planned a visit to the city market for tomorrow in order to buy some new plants and seeds for her land. She has already some ideas about the plants she wanna cultivate, but she would like to know something more. Since she is already registered in the DREAM platform she logs in and navigates to the archive area. She searchs for "Apple Gourd" in the archive and opens its technical sheet. Shyla sees that "Apple Gourd" is a really nice plant for the humidity and type of soils of her land. Shyla is now more informed and sure of her choice.

Sending a ticket Ravi has a problem inside a piece of his lands because during the monsoons season the river has overflowed and flooded a good amount of fields. This

particular part of his territory is very fruitful, he's always able to obtain a good amount of products from it, but this year he doesn't know how to come out from this terrible situation. Furthermore, he is not sure if he will be able to pay the providers without the incomes produced by this piece of land. Ravi tries to resolve this situation by sending a ticket to his referenced policy maker, thanks to the dedicated part in the DREAM application. In this help request he writes about his problem and in few days an expert will answer to him by explaining a possible solution to balance the income and to recover the piece of land which is under water.

Checking for news Ranjeet usually wakes up at 6:00 A.M., the first thing he checks is the news part on the DREAM application in order to understand how to organize his daily work. He discovers that in the afternoon of that day with a high probability there will be a powerful storm. Thanks to the notifications he is able to close all the greenhouses and to ask for help to his colleagues in order to make his cultivation's environment safe and ready to prevent huge damages.

Filling a report Anish, at the end of the harvesting season, has to share the results of his productions with the government and the experts in order to understand if he's doing a good job with his farms or not. His aim is to obtain extra grants because he spent lot of time and efforts in the cultivation of his fields. The application allows him to complete a pre-compiled report where Anish can insert all of the needed data which will be sent to the policy maker. If the results of his productivity will satisfy certain criteria he will obtain more money from the state. On the other hands in case of really negative evaluation he could obtain monetary aids and more experts' help.

Replying to a thread in the forum Naresh is an expert and productive farmer in the province of Adilabad and he likes to share his knowledge in order to help other farmers. As every evening after dinner he logs into the DREAM application and gives a look into the forum section. He notices some new messages in a thread he was really expert about. After opening it he reads all the messages and then he clicks in the "send comment" button in order to give his opinion. After having wrote the message he confirms to submit it and he return in the forum's home looking for some other open threads.

Receiving response of economical help Kushagra is eating his lunch when he receives a notification from the DREAM application. Since he was waiting an answer for an important request he lunches the app and, since he is already logged in, the ticket's page is showed. He sees the reply he was waiting for and he click to read all the answer. in the message he is asked to send other specific information in order to proceed with the request of economic refreshment. He click to the reply button and writes down all the information. Now he presses the forward button and then he logs out from the

application.

Looking for statistics Ajar has to share information to government about the farmers in his monitored area. He logs into the DREAM system and navigates to the analysis page. Here he can visualize all the important data of the area through different charts, grouping and ordering them how he prefers. Once he has arranged the data in a meaningful way he exports them. In the next meeting he will use them to sum up the situation about his controlled zone.

Replying to tickets Bhavin is working in his office when he receives a notification through the DREAM application. A new ticket from Ravi, regarding an hydrological problem. Ravi clicks on the notification and the app opens on the ticket's management page. Bhavin now can open the unread ticket and read the entire message. After that he clicks on the reply button and writes down some advice and useful information for Ravi's problem.

Organize an inspection Akanksha is one of the policy makers of Siddipet and he's scheduling his next inspection week. In order to fix all the appointments he needs to call Dayanand and Kamalkant, two big farm owners. After he has logged in the DREAM application he navigate to the contacts list. Here he can insert the names and click the search buttons. Two contacts are now showed in the list and he can click on each one to see more information. After clicking on Kamalkant he see his phone number and he is ready to call. Once he has finished the call he can do the same for Dayanand.

2.1.2. Static Information Model

The diagram represented below provides a static information model of the application domain. It is the structure of the system, but it only contains few attributes and it doesn't include every class which will be used for the DREAM implementation.

The main aspects highlighted by the diagram are:

1. The application considers two types of users: "Farmer" and "Policy Maker". Since they share some attributes and functionalities, they are both a specialization of the main class "User".
2. There must be the presence of a map which includes all the zones, the farms and the weather. Map-zone and zone-farm are linked through a aggregation relationship because a "Map" is composed by multiple "Zone" and every "Zone" is composed by multiple "Farm".
3. The "Archive" class is created by an aggregation of all the other classes: "Plant", "Fertilizer", "Tool" and "Technique". The "Archive" is unique.
4. The "Forum" class is created as a list of "Thread" which contains a set of "Message". "Forum" is unique.
5. "Report" and "Ticket" classes represent the two kind of messages that "Policy-Maker" and "Farmer" can use to communicate inside the application.
6. "News" is a class that is called by everyone in the system in order to communicate the information to all the "User".
7. "Statistic" and "Contacts" instead are classes that will be used to implement "PolicyMaker"'s functionalities.

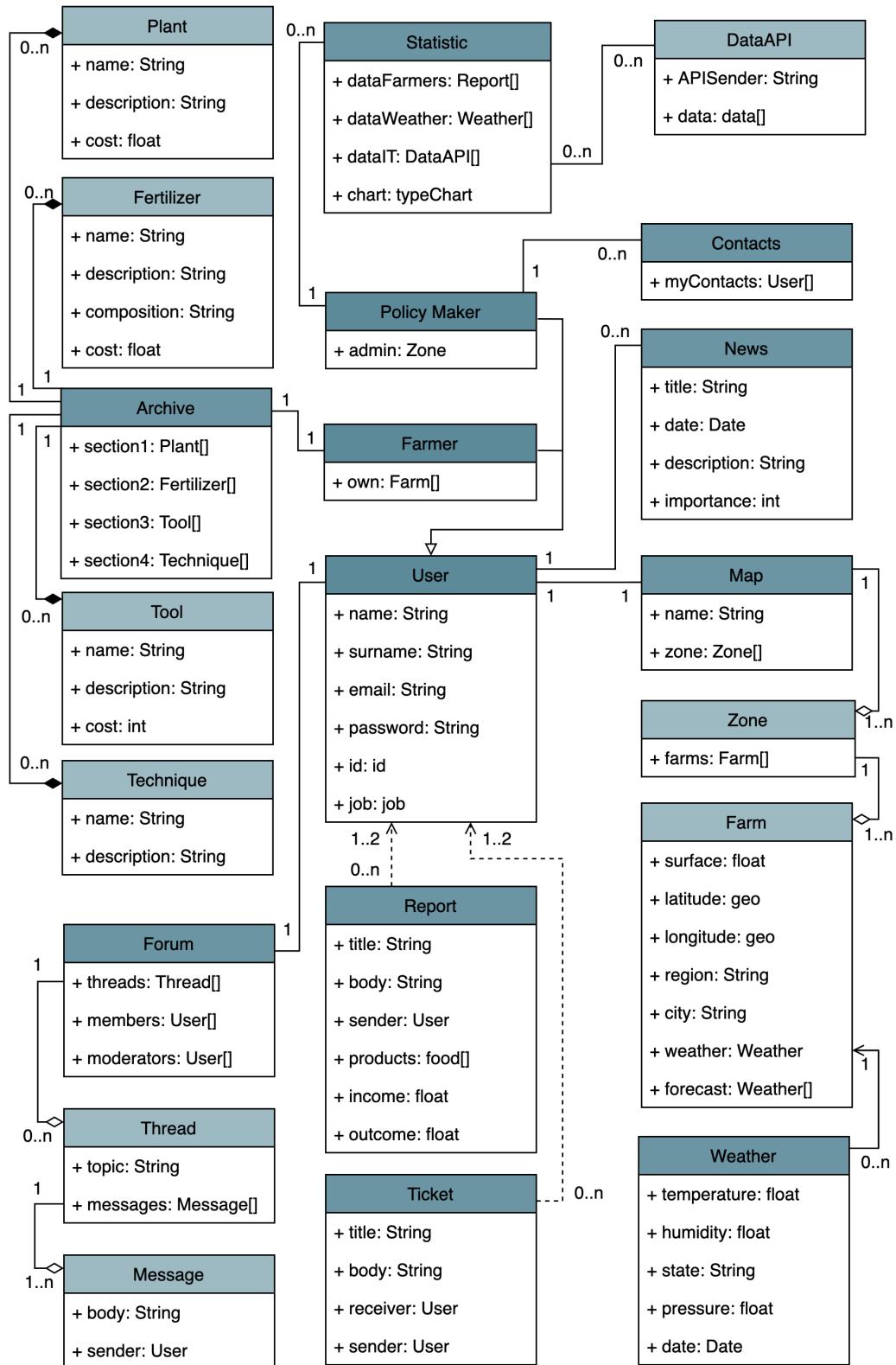


Figure 2.1: class diagram

2.1.3. Dynamic Class Behaviour Models

The state diagrams listed below show some particular behaviours of the application: how these are modeled and evolve through time. While the activity diagrams explain the sequence of actions that a user has to do in order to complete that specific activity.

Forum - farmer

This state diagram represents how a farmer can interact with the application's forum. At the beginning the system just waits for a command more precisely the "Forum" button. After this, there will be the Forum home page where the farmer can search for a particular thread, then there are 2 branches which explain the possibility to find or not this thread. If the thread has been found, it can be read, replied or closed (if the farmer has the ownership of the thread). In the negative case the user is brought back to the forum page.

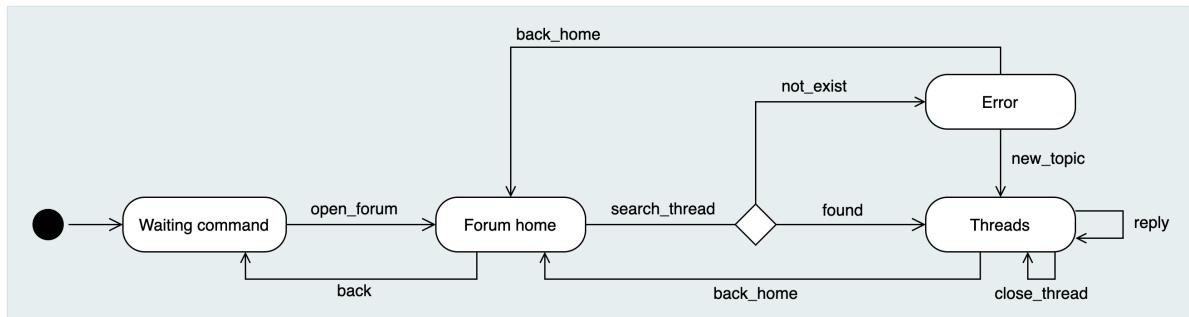


Figure 2.2: Forum - farmer | statechart

Ticket System - farmer

This chart represents the ticket system, from the point of view of a farmer. A farmer has the possibility to create a new ticket or to check if one of the old tickets sent before has received an answer. If not, he/she has the possibility to come back to the waiting state.

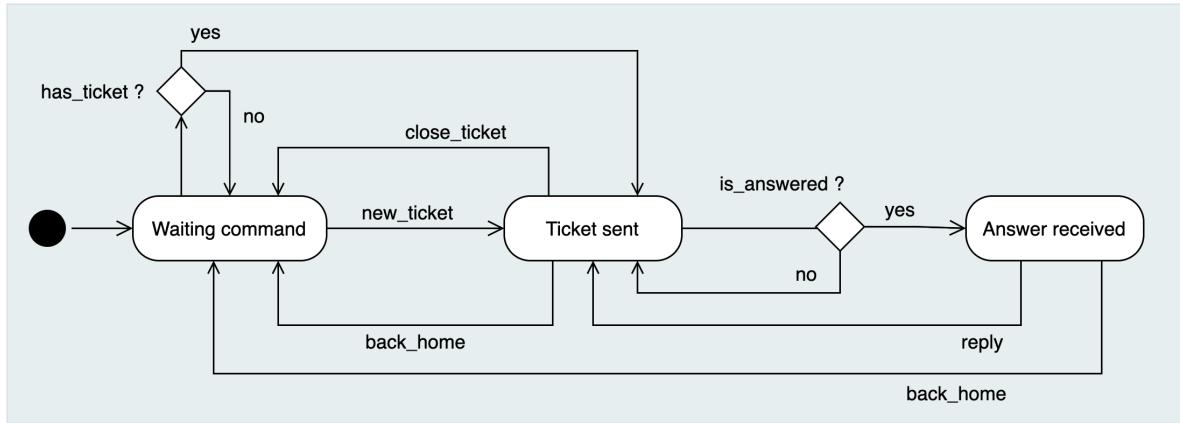


Figure 2.3: Ticket System - farmer | statechart

Report System - farmer

In the last state diagram it's showed the situation where a farmer sends a report to the policy maker who is the responsible of his/her area. As always the system wait for a command by the user. He/she can check the answers of previous reports or he/she can compile a new one with all the data and then send it to the specific policy maker. As we can see tickets and reports have some similar behaviour, the main difference indeed is in the meaning and in their using situations.

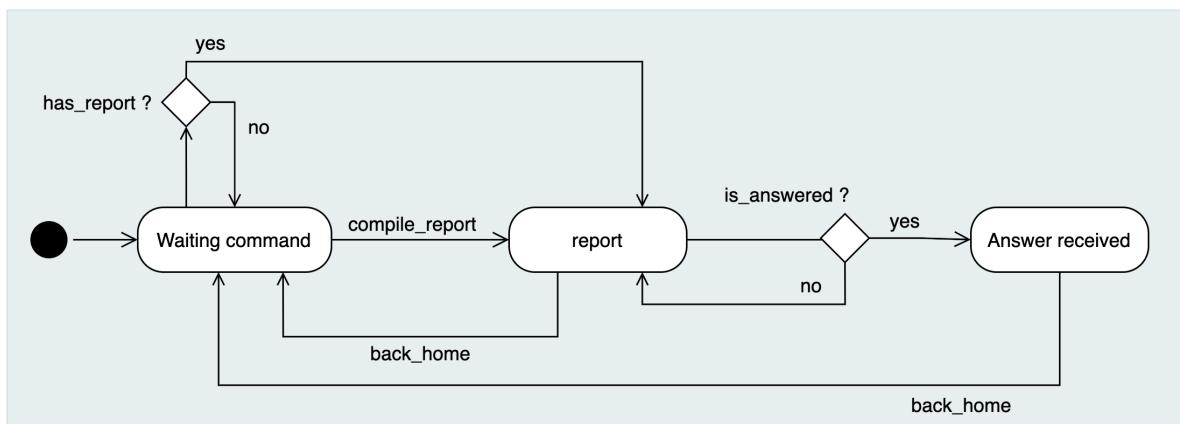


Figure 2.4: Report System - farmer | statechart

Weather forecast

The first activity diagram represents the situation where a user check the weather map of his/her zone. He/She has the possibility to look for a specific range of time, then the

system will shows the forecast of the chosen area and time. After that, the user can request an other forecast or end the activity.

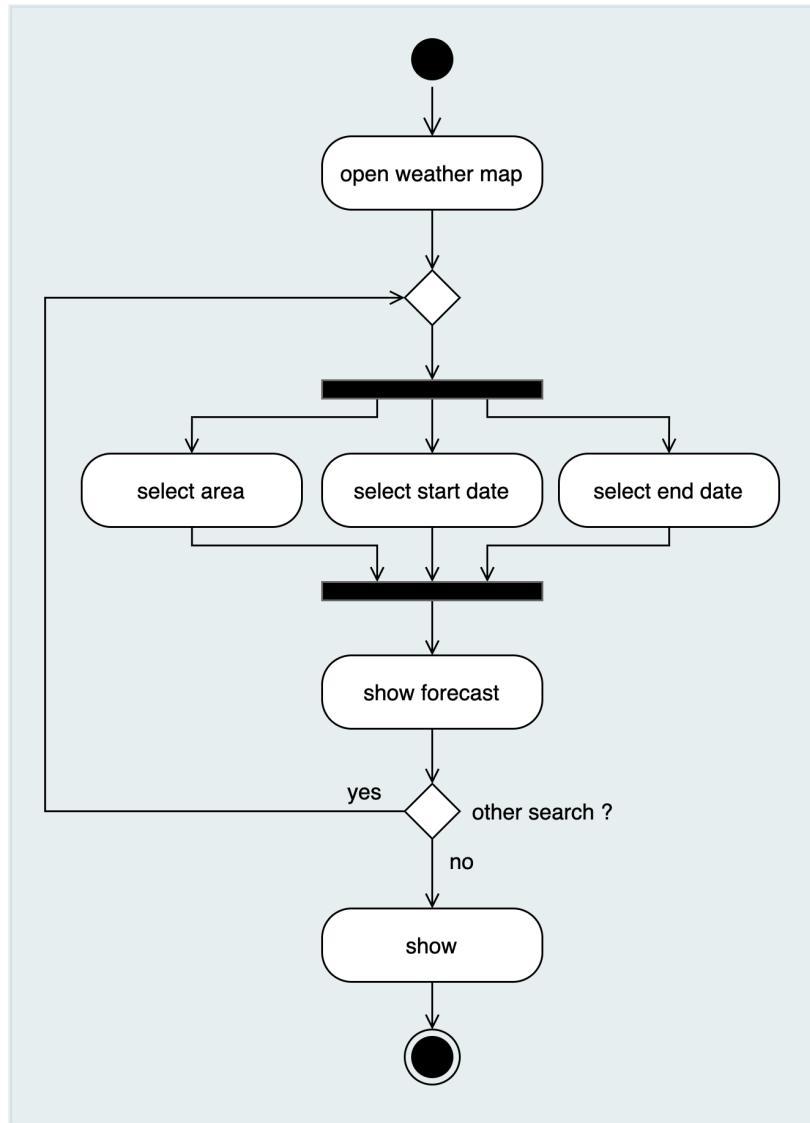


Figure 2.5: Weather forecast | activity diagram

Analysis of data - policy maker

Policy Maker is assigner just to a single area, so he can't chose which area to display data, but he can chose if he wanna consider all data or just a farmer data (providing the farmer id). After this he/she can decide which data to consider and how to visualize and organize them in order to better understand.

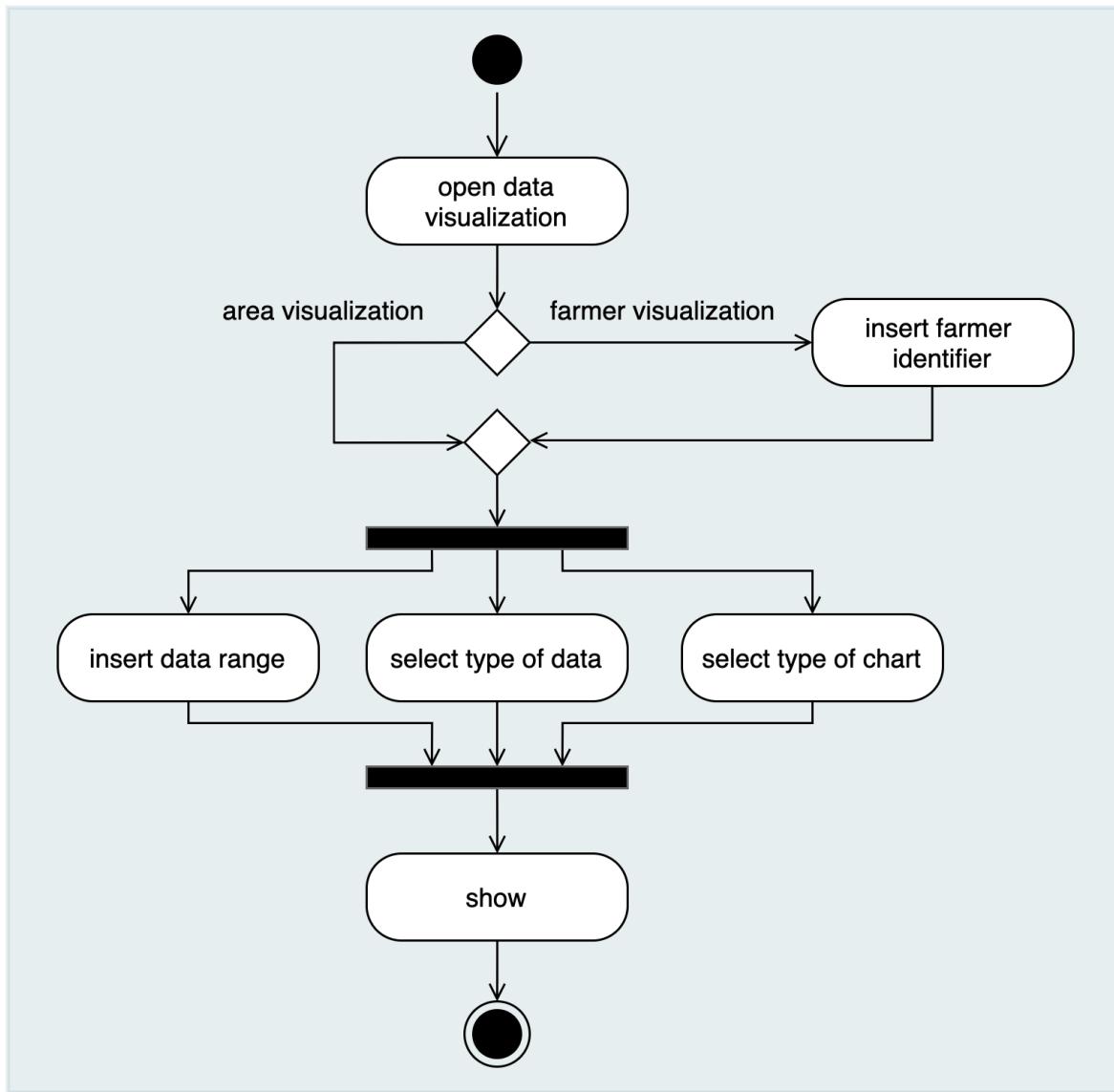


Figure 2.6: Analysis of data - policy maker | activity diagram

Report system - farmer

In the last activity diagram there is a more detailed explanation of the report system (always farmers side): here every step is explained. At the beginning the farmer has to insert the id of a farm. Secondly he/she has to insert all the mandatory data needed for the future analysis as in DA if she/he doesn't own all the data, he/she will contact an external source in order to have them. At the end he/she sends the completed report to the specific policy maker. If this farmer owns more than one property he/she can repeat this process until all the properties are covered.

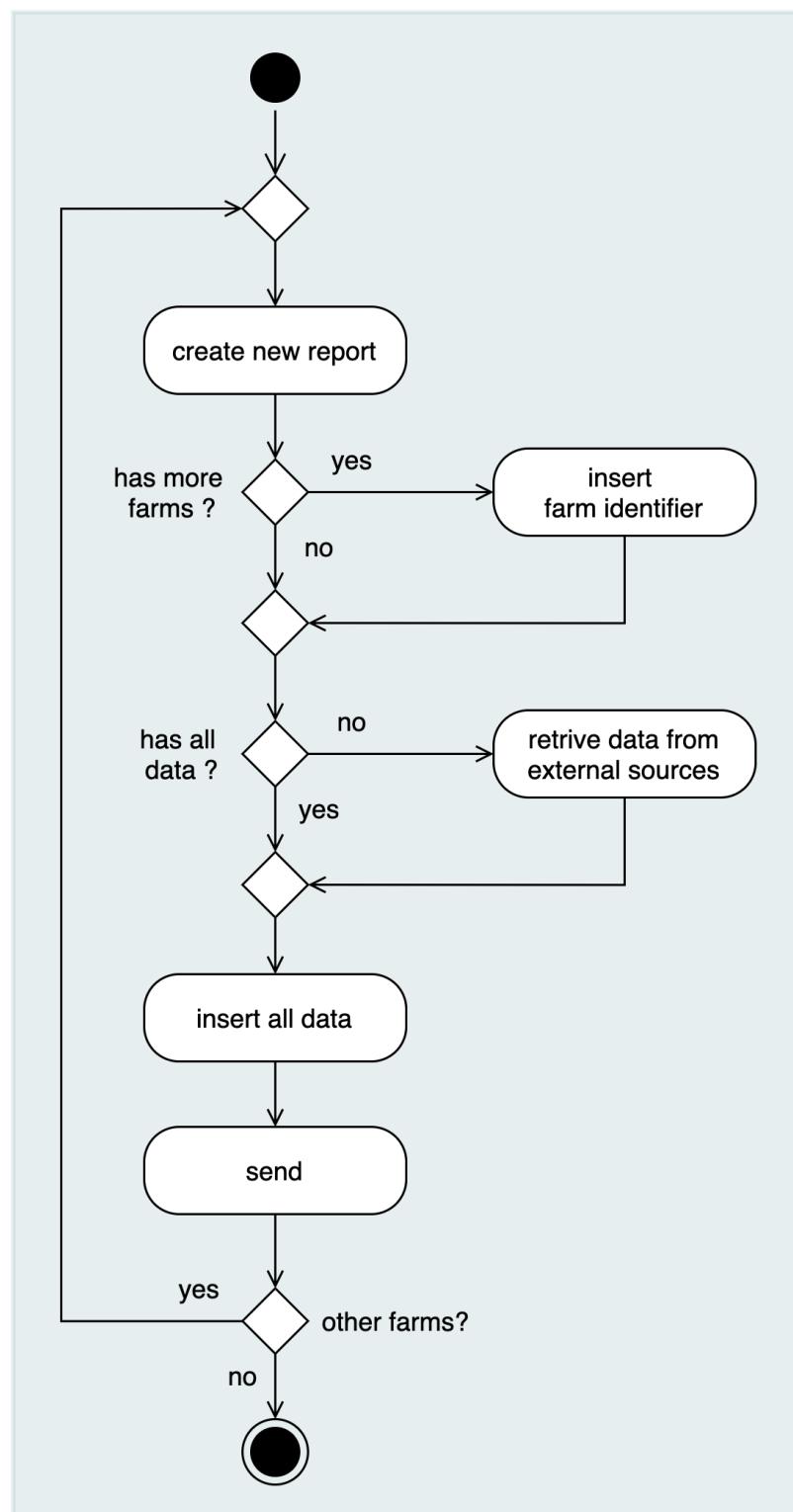


Figure 2.7: Report system - farmer | activity diagram

2.2. Product functions

The system gives the farmers and policy makers the opportunity to exploit multiple main functionalities.

Functionalities available for both:

- **Weather forecast:** A user has the possibility to see the weather forecast of the place where he/she is located. In addition to this he/she is able to analyze other important data such as the amount of water that could fall down, the percentage of humidity in the air or the probability that there will be a thunderstorm.
- **Reading news:** A User is able to stay updated on the latest news thanks to the dedicated part on the application. The information gathered in this section concern all the possible facts linked to politics, agriculture and climate changes.

Functionalities available for the farmers:

- **Archive:** The farmer can improve his knowledge about the plant and fertilizers he has in his farmland thanks to the large pool of information inside the plants archive in the application. He/She has the possibility to search the plants by their name and then obtain the important details in order to cultivate them.
- **Sending Tickets:** The farmer can send a ticket in the appropriate part inside the application where he/she can ask everything about one or more problems he/she faces during his/her job and an expert or a member of the government will analyse the request and propose a solution.
- **Using Forum:** The farmer can discuss a problem or propose his/her opinion with the other farmers and experts by writing it in the forum that is available on the application. Here all the comments or discussions are public and they can be read by everyone.
- **Sending reports:** The farmer can share with the policy maker and the experts his/her productivity data by completing a pre-compiled report which is available inside the appropriate part in the application.

Functionalities available for the policy makers:

- **Read and answer to tickets:** A policy maker can read, analyze and answer to all the tickets in the compartment present inside the software. He/She is also able to know from who the tickets come from, in this way he/she can try to resolve the problem in a more direct way.

- **Read and answer to reports:** A policy maker can read, analyze and answer to reports in the compartment present inside the software. He/She is also able to know from who the reports come from, in this way he/she can understand which farmer should be awarded and which is in difficulty.
- **Analyze statistics:** The policy maker has the possibility to watch and analyze all the statistics which come from the analysis of all the report sent by the farmer. Inside this area there will also be the possibility to watch information about the weather forecast, the humidity in the air and other important data linked with the climate.
- **Address book:** The policy maker has the possibility to access to all of his/her contacts in the special area inside the application. In this way he/she can communicate with all kind of people from the experts of the different sectors to the farmers.
- **Read Forum:** A policy maker is able to enter in the forum in order to read all the conversations and topics made by farmers concerning their various problems.

2.3. User characteristics

The system can be exploited by the following actors:

- **Farmer:** A person which owns a farm. He/she can be helped from the government, policy makers and experts and could help other colleagues if he/she wishes.
- **Policy maker:** A person who is designed to analyse and monitor a specific zone composed of different farms. He/she becomes a link between farmers and government.

2.4. Assumptions

Domain Assumptions	Description
D1	A farmer can exploit the application functions only if he/she is registered and the account is unique.
D2	A policy maker can exploit the application functions only if he/she is registered and the account is unique.
D3	Every farmer and policy maker give the authorization to use their geographical data for the internal system processes.
D4	Every farmer give the authorization to use their land registry data for the internal system processes.
D5	The registered farmers are the owners of the farm.
D6	Every farm is delimited, land registered from government and has a unique identifier.
D7	Before sending economical helps or bonuses to farmers there is an external check by a government financial organ.
D8	The weather data extracted from the external API are reliable.
D9	Every farmer own a generic mobile device in order to use the application, otherwise the government will provide one.
D10	A farmer owns all the information to fulfill the report or he/she obtains them from an external provider.
D11	Farmers send the report with all of their data every 6 months.
D12	The farmers' forum is moderated internally by them.

Table 2.1: Table Of Domain Assumptions

3 | SPECIFIC REQUIREMENTS

3.1. External Interface Requirements

3.1.1. User Interfaces

The following mockups are presented here just to show an idea of the application that will be in use by the farmers. By figure 3.2 we can see the functionalities that will be available by the farmer: check weather map and forecast, check news, search in the archive, open the forum, open the ticket list and open the report list. All this functionalities will be better described in section 3.2.5.

Policy maker interface is not here displayed for the sake of brevity, but the general layout will be similar to the farmer one, with addition of more functionalities as analyse data and check farmers' informations. For further details about policy maker functionalities check again section 3.2.5.

For a better/faster user experience, mainly meant for the policy maker workflow it will be provided also a web app interface. Also this one will follow the same general layout here displayed. The complete list of mockups (app and web app) will be available in the design document.

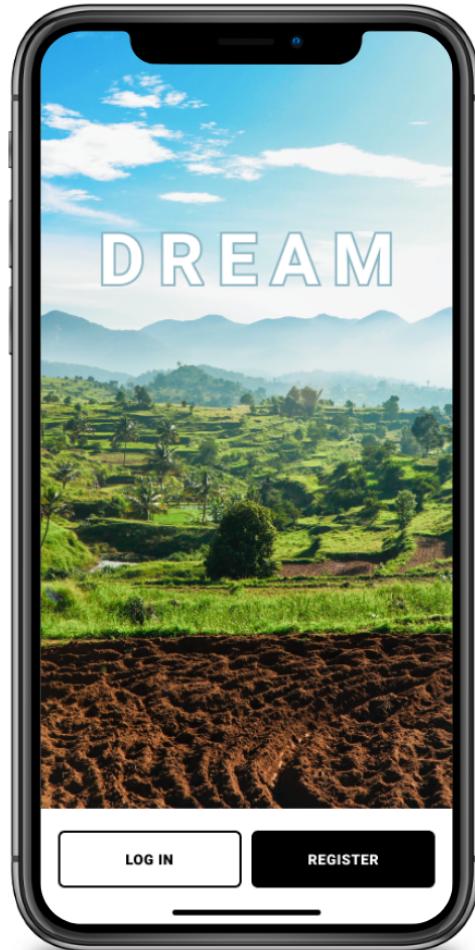


Figure 3.1: Mockup: logged out

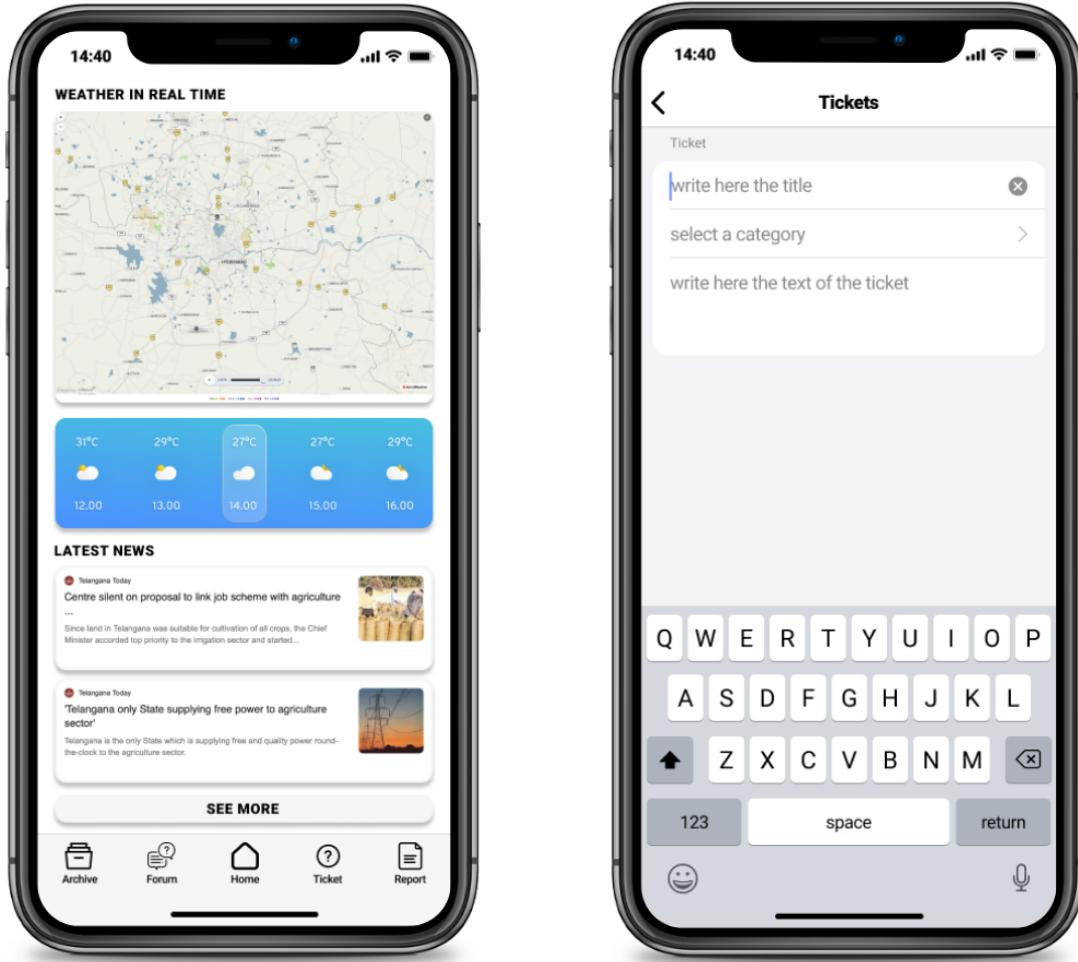


Figure 3.2: Mockup: on the left farmer's home, on the right farmer's new ticket

3.1.2. Software Interfaces

- **geo-localization:** The system needs to interface with an external geo-localization API in order to understand the position of the farmer's land, to filter the information giving relevance based on the geographical position.
- **land registry:** In addition to this interface, the application uses an API to obtain land registry data useful for the internal system behaviour.
- **weather:** In order to give weather information and forecasting the application exploits an external weather API, this functionality must be reliable with the aim to guarantee the correct system behaviour. It provides data concerned the temperature, the humidity in the air, the wind speed, the probability of precipitations and its quantity.
- **graphic:** An external chart API is exploited with the purpose to offer to the policy

makers multiple facets and analysis of the data.

- **archive:** The system also requires and API to support the internal archive, this allows to always have up to date data about plants and fertilizer.
- **forum:** The application uses an external forum's framework in order to manage all the threads.
- **news:** In order to keep our news section reliable and up to date the application interfaces with an external news API which extracts only the relevant information about politics and agriculture.

3.1.3. Communication Interfaces

In order to maintain the correct behaviour of the application and up to date information the system requires a stable internet connection.

3.2. Functional Requirements

3.2.1. List Of Requirements

Requirements	Description
R1	The system shall allow an unregistered farmers to register.
R2	The system shall allow an unregistered policy makers to register.
R3	The system shall allow a registered farmers to unregister.
R4	The system shall allow a registered policy maker to unregister.
R5	The system shall send a confirmation email to a farmer who finishes the registration process in order to confirm the entire procedure.
R6	The system shall send a confirmation email to a policy maker who finishes the registration process in order to confirm the entire procedure.
R7	The system shall allow the farmer to check the weather conditions of the entire week.
R8	The system shall allow the farmer to visualize the weather characteristics like the humidity , the temperature and the probability of precipitations in the current day.

R9	The system shall allow the farmer to visualize the general weather map of Telengana.
R10	The system shall allow the farmer to visualize the weather map of his farm.
R11	The system shall allow the farmer to visualize the news.
R12	The system shall allow the farmer to visualize the archive.
R13	The system shall allow the farmer to search inside the archive.
R14	The system shall allow the farmer to visualize the results of the research in the archive.
R15	The system shall allow the farmer to send a ticket request.
R16	The system shall allow the farmer to get notified by an answer of his/her ticket.
R17	The system shall allow the farmer to compile a report.
R18	The system shall allow the farmer to send a report to a policy maker.
R19	The system shall allow the farmer to get notified for a report answer.
R20	The system shall allow the farmer to visualize the forum.
R21	The system shall allow the farmer to open a thread in the forum.
R22	The system shall allow the farmer to read a thread in the forum.
R23	The system shall allow the farmer to create a new thread in the forum.
R24	The system shall allow the farmer to answer to a thread in the forum.
R25	The system shall allow the farmer to delete a thread in the forum.
R26	The system shall allow the farmer to read a thread in the forum.
R27	The system shall allow the policy maker to check the weather conditions of the entire week.
R28	The system shall allow the policy maker to visualize the weather characteristics like the humidity , the temperature and the probability of precipitations in the current day.
R29	The system shall allow the policy maker to visualize the general weather map of Telengana.
R30	The system shall allow the policy maker to visualize the weather of the area he/she checking.

R31	The system shall allow the policy maker to visualize the news.
R32	The system shall allow the policy maker to get notified when he/she gets a ticket.
R33	The system shall allow the policy maker to answer to a ticket request.
R34	The system shall allow the policy maker to get notified when he/she gets a report from a farmer.
R35	The system shall allow the policy maker to write an answer to a report.
R36	The system shall allow the policy maker to visualize the forum.
R37	The system shall allow the policy maker to read a thread in the forum.
R38*	The system shall allow the policy maker to read a thread in the forum.
R39	The system shall allow the policy maker to visualize data.
R40	The system shall allow the policy maker to aggregate and disaggregate data.
R41	The system shall allow the policy maker to change the visualization of data.
R42	The system shall allow the policy maker to visualize contacts.

Table 3.1: Table Of Requirements

3.2.2. Mapping Requirements on Goals

R/G	G1	G2	G3	G4	G5	G6	G7	G8
R1	X	X	X	X	X	X	X	X
R2			X		X	X	X	X
R3								
R4								
R5	X	X	X	X	X	X	X	X
R6			X		X	X	X	X
R7	X	X						
R8	X	X						

R9	X							
R10	X	X						
R11		X			X			
R12		X						
R13		X						
R14		X						
R15			X		X	X	X	X
R16			X		X			
R17			X		X	X	X	X
R18			X		X	X	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				
R27						X		
R28						X		
R29						X		
R30						X		
R31					X	X		
R32			X		X	X	X	X
R33			X		X	X	X	X
R34			X		X	X	X	X
R35			X		X	X	X	X
R36					X	X		
R37					X	X		
R38					X	X		
R39			X		X	X	X	X
R40			X		X	X	X	X

R41			X		X	X	X	X
R42			X		X		X	

Table 3.2: Mapping Requirements on Goals Table

3.2.3. Mapping Domain Assumptions on Goals

D/G	G1	G2	G3	G4	G5	G6	G7	G8
D1	X	X						
D2			X		X	X	X	X
D3	X	X	X		X	X	X	X
D4	X	X	X		X	X	X	X
D5			X		X	X	X	X
D6		X	X		X	X	X	X
D7			X		X	X	X	X
D8	X	X				X	X	
D9	X	X	X	X	X	X	X	X
D10			X		X	X	X	X
D11			X		X	X	X	X
D12		X		X				

Table 3.3: Mapping Domain Assumptions on Goals Table

3.2.4. Mapping explicit table

G1	Allow farmers to easily check weather condition
R1	The system shall allow an unregistered farmers to register.
R5	The system shall send a confirmation email to a farmer who finishes the registration process in order to confirm the entire procedure.
R7	The system shall allow the farmer to check the weather conditions of the entire week

R8	The system shall allow the farmer to visualize the weather characteristics like the humidity , the temperature and the probability of precipitations in the current day.
R9	The system shall allow the farmer to visualize the general weather map of Telengana.
R10	The system shall allow the farmer to visualize the weather map of his/her farm.
D1	A farmer can exploit the application functions only if he/she is registered and the account is unique.
D3	Every farmer and policy maker give the authorization to use their geographical data for the internal system processes.
D4	Every farmer give the authorization to use their land registry data for the internal system processes.
D8	The weather data extracted from the external API are reliable.
D9	Every farmer own a generic mobile device in order to use the application, otherwise the government will provide one.

G2	Allow farmers to have technical and personalized advices from other farmers or experts of the field.
R1	The system shall allow an unregistered farmers to register.
R5	The system shall send a confirmation email to a farmer who finishes the registration process in order to confirm the entire procedure.
R7	The system shall allow the farmer to check the weather conditions of the entire week.
R8	The system shall allow the farmer to visualize the weather characteristics like the humidity , the temperature and the probability of precipitations in the current day.

R10	The system shall allow the farmer to visualize the weather map of his/her farm.
R11	The system shall allow the farmer to visualize the news.
R12	The system shall allow the farmer to visualize the archive.
R13	The system shall allow the farmer to search inside the archive.
R14	The system shall allow the farmer to visualize the results of the research in the archive.
R20	The system shall allow the farmer to visualize the forum.
R21	The system shall allow the farmer to open a thread in the forum.
R22	The system shall allow the farmer to read a thread in the forum.
R23	The system shall allow the farmer to create a new thread in the forum.
R24	The system shall allow the farmer to answer to a thread in the forum.
R25	The system shall allow the farmer to delete a thread in the forum.
R26	The system shall allow the farmer to read a thread in the forum.
D1	A farmer can exploit the application functions only if he/she is registered and the account is unique.
D3	Every farmer and policy maker give the authorization to use their geographical data for the internal system processes.
D4	Every farmer give the authorization to use their land registry data for the internal system processes.
D6	Every farm is delimited, land registered from government and has a unique identifier.
D8	The weather data extracted from the external API are reliable.

D9	Every farmer own a generic mobile device in order to use the application, otherwise the government will provide one.
D12	The farmers' forum is moderated internally by them.

G3 Allow farmer to ask for help to the government.	
R1	The system shall allow an unregistered farmers to register.
R2	The system shall allow an unregistered policy makers to register.
R5	The system shall send a confirmation email to a farmer who finishes the registration process in order to confirm the entire procedure.
R6	The system shall send a confirmation email to a policy maker who finishes the registration process in order to confirm the en-tire. procedure.
R15	The system shall allow the farmer to send a ticket request.
R16	The system shall allow the farmer to get notified by an answer of his/her ticket.
R17	The system shall allow the farmer to compile a report.
R18	The system shall allow the farmer to send a report to a policymaker.
R19	The system shall allow the farmer to get notified for a report answer.
R32	The system shall allow the policy maker to get notified when he/she gets a ticket.
R33	The system shall allow the policy maker to answer to a ticket request.
R34	The system shall allow the policy maker to get notified when he/she gets a report from a farmer.
R35	The system shall allow the policy maker to write an answer to a report.
R39	The system shall allow the policy maker to visualize data.

R40	The system shall allow the policy maker to aggregate and disaggregate data.
R41	The system shall allow the policy maker to change the visualization of data.
R42	The system shall allow the policy maker to visualize contacts.
D2	A policy maker can exploit the application functions only if he/she is registered and the account is unique.
D3	Every farmer and policy maker give the authorization to use their geographical data for the internal system processes.
D4	Every farmer give the authorization to use their land registry data for the internal system processes.
D5	The registered farmers are the owners of the farm.
D6	Every farm is delimited, land registered from government and has a unique identifier.
D7	Before sending economical helps or bonuses to farmers there is an external check by a government financial organ.
D9	Every farmer own a generic mobile device in order to use the application, otherwise the government will provide one.
D10	A farmer owns all the information to fulfill the report or he/she obtains them from an external provider.
D11	Farmers send the report with all of their data every 6 months.

G4	Improve the communication between farmers through a forum.
R1	The system shall allow an unregistered farmers to register.
R5	The system shall send a confirmation email to a farmer who finishes the registration process in order to confirm the entire procedure.

R20	The system shall allow the farmer to visualize the forum.
R21	The system shall allow the farmer to open a thread in the forum.
R22	The system shall allow the farmer to read a thread in the forum.
R23	The system shall allow the farmer to create a new thread in the forum.
R24	The system shall allow the farmer to answer to a thread in the forum.
R25	The system shall allow the farmer to delete a thread in the forum.
R26	The system shall allow the farmer to read a thread in the forum.
D9	Every farmer own a generic mobile device in order to use the application, otherwise the government will provide one.
D12	The farmers' forum is moderated internally by them.

G5 Improve data communication between farmers and government.	
R1	The system shall allow an unregistered farmers to register.
R2	The system shall allow an unregistered policy makers to register.
R5	The system shall send a confirmation email to a farmer who finishes the registration process in order to confirm the entire procedure.
R6	The system shall send a confirmation email to a policy maker who finishes the registration process in order to confirm the en-tire. procedure.
R11	The system shall allow the farmer to visualize the news.
R15	The system shall allow the farmer to send a ticket request.
R16	The system shall allow the farmer to get notified by an answer of his/her ticket.

R17	The system shall allow the farmer to compile a report.
R18	The system shall allow the farmer to send a report to a policymaker.
R19	The system shall allow the farmer to get notified for a report answer.
R31	The system shall allow the policy maker to visualize the news.
R32	The system shall allow the policy maker to get notified when he/she gets a ticket.
R33	The system shall allow the policy maker to answer to a ticket request.
R34	The system shall allow the policy maker to get notified when he/she gets a report from a farmer.
R35	The system shall allow the policy maker to write an answer to a report.
R36	The system shall allow the policy maker to visualize the forum.
R37	The system shall allow the policy maker to read a thread in the forum.
R38	The system shall allow the policy maker to read a thread in the forum.
R39	The system shall allow the policy maker to visualize data.
R40	The system shall allow the policy maker to aggregate and disaggregate data.
R41	The system shall allow the policy maker to change the visualization of data.
R42	The system shall allow the policy maker to visualize contacts.
D2	A policy maker can exploit the application functions only if he/she is registered and the account is unique.
D3	Every farmer and policy maker give the authorization to use their geographical data for the internal system processes.

D4	Every farmer give the authorization to use their land registry data for the internal system processes.
D5	The registered farmers are the owners of the farm.
D6	Every farm is delimited, land registered from government and has a unique identifier.
D7	Before sending economical helps or bonuses to farmers there is an external check by a government financial organ.
D9	Every farmer own a generic mobile device in order to use the application, otherwise the government will provide one.
D10	A farmer owns all the information to fulfill the report or he/she obtains them from an external provider.
D11	Farmers send the report with all of their data every 6 months.

G6	Allow the government to improve the analysis and the sharing of important data concerning agriculture.
R1	The system shall allow an unregistered farmers to register.
R2	The system shall allow an unregistered policy makers to register.
R5	The system shall send a confirmation email to a farmer who finishes the registration process in order to confirm the entire procedure.
R6	The system shall send a confirmation email to a policy maker who finishes the registration process in order to confirm the en-tire. procedure.
R15	The system shall allow the farmer to send a ticket request.
R17	The system shall allow the farmer to compile a report.
R18	The system shall allow the farmer to send a report to a policymaker.
R27	The system shall allow the policy maker to check the weather conditions of the entire week.

R28	The system shall allow the policy maker to visualize the weather characteristics like the humidity , the temperature and the probability of precipitations in the current day.
R29	The system shall allow the policy maker to visualize the general weather map of Telengana.
R30	The system shall allow the policy maker to visualize the weather of the area he/she checking.
R31	The system shall allow the policy maker to visualize the news.
R32	The system shall allow the policy maker to get notified when he/she gets a ticket.
R33	The system shall allow the policy maker to answer to a ticket request.
R34	The system shall allow the policy maker to get notified when he/she gets a report from a farmer.
R35	The system shall allow the policy maker to write an answer to a report.
R36	The system shall allow the policy maker to visualize the forum.
R37	The system shall allow the policy maker to read a thread in the forum.
R38	The system shall allow the policy maker to read a thread in the forum.
R39	The system shall allow the policy maker to visualize data.
R40	The system shall allow the policy maker to aggregate and disaggregate data.
R41	The system shall allow the policy maker to change the visualization of data.
D2	A policy maker can exploit the application functions only if he/she is registered and the account is unique.

D3	Every farmer and policy maker give the authorization to use their geographical data for the internal system processes.
D4	Every farmer give the authorization to use their land registry data for the internal system processes.
D5	The registered farmers are the owners of the farm.
D6	Every farm is delimited, land registered from government and has a unique identifier.
D7	Before sending economical helps or bonuses to farmers there is an external check by a government financial organ.
D8	The weather data extracted from the external API are reliable.
D9	Every farmer own a generic mobile device in order to use the application, otherwise the government will provide one.
D10	A farmer owns all the information to fulfill the report or he/she obtains them from an external provider.
D11	Farmers send the report with all of their data every 6 months.

G7 Allow the government to have specific data about farmers.	
R1	The system shall allow an unregistered farmers to register.
R2	The system shall allow an unregistered policy makers to register.
R5	The system shall send a confirmation email to a farmer who finishes the registration process in order to confirm the entire procedure.
R6	The system shall send a confirmation email to a policy maker who finishes the registration process in order to confirm the en-tire. procedure.
R15	The system shall allow the farmer to send a ticket request.
R17	The system shall allow the farmer to compile a report.
R18	The system shall allow the farmer to send a report to a policymaker.

R32	The system shall allow the policy maker to get notified when he/she gets a ticket.
R33	The system shall allow the policy maker to answer to a ticket request.
R34	The system shall allow the policy maker to get notified when he/she gets a report from a farmer.
R35	The system shall allow the policy maker to write an answer to a report.
R39	The system shall allow the policy maker to visualize data.
R40	The system shall allow the policy maker to aggregate and disaggregate data.
R41	The system shall allow the policy maker to change the visualization of data.
R42	The system shall allow the policy maker to visualize contacts.
D2	A policy maker can exploit the application functions only if he/she is registered and the account is unique.
D3	Every farmer and policy maker give the authorization to use their geographical data for the internal system processes.
D4	Every farmer give the authorization to use their land registry data for the internal system processes.
D5	The registered farmers are the owners of the farm.
D6	Every farm is delimited, land registered from government and has a unique identifier.
D7	Before sending economical helps or bonuses to farmers there is an external check by a government financial organ.
D8	The weather data extracted from the external API are reliable.

D9	Every farmer own a generic mobile device in order to use the application, otherwise the government will provide one.
D10	A farmer owns all the information to fulfill the report or he/she obtains them from an external provider.
D11	Farmers send the report with all of their data every 6 months.

G8	Allow the policy makers to easily recognise critical and virtuous situations.
R1	The system shall allow an unregistered farmers to register.
R2	The system shall allow an unregistered policy makers to register.
R5	The system shall send a confirmation email to a farmer who finishes the registration process in order to confirm the entire procedure.
R6	The system shall send a confirmation email to a policy maker who finishes the registration process in order to confirm the en-tire. procedure.
R15	The system shall allow the farmer to send a ticket request.
R17	The system shall allow the farmer to compile a report.
R18	The system shall allow the farmer to send a report to a policymaker.
R32	The system shall allow the policy maker to get notified when he/she gets a ticket.
R33	The system shall allow the policy maker to answer to a ticket request.
R34	The system shall allow the policy maker to get notified when he/she gets a report from a farmer.
R35	The system shall allow the policy maker to write an answer to a report.
R39	The system shall allow the policy maker to visualize data.

R40	The system shall allow the policy maker to aggregate and disaggregate data.
R41	The system shall allow the policy maker to change the visualization of data.
D2	A policy maker can exploit the application functions only if he/she is registered and the account is unique.
D3	Every farmer and policy maker give the authorization to use their geographical data for the internal system processes.
D4	Every farmer give the authorization to use their land registry data for the internal system processes.
D5	The registered farmers are the owners of the farm.
D6	Every farm is delimited, land registered from government and has a unique identifier.
D7	Before sending economical helps or bonuses to farmers there is an external check by a government financial organ.
D9	Every farmer own a generic mobile device in order to use the application, otherwise the government will provide one.
D10	A farmer owns all the information to fulfill the report or he/she obtains them from an external provider.
D11	Farmers send the report with all of their data every 6 months.

3.2.5. Use Case Diagrams

Unregistered Farmer

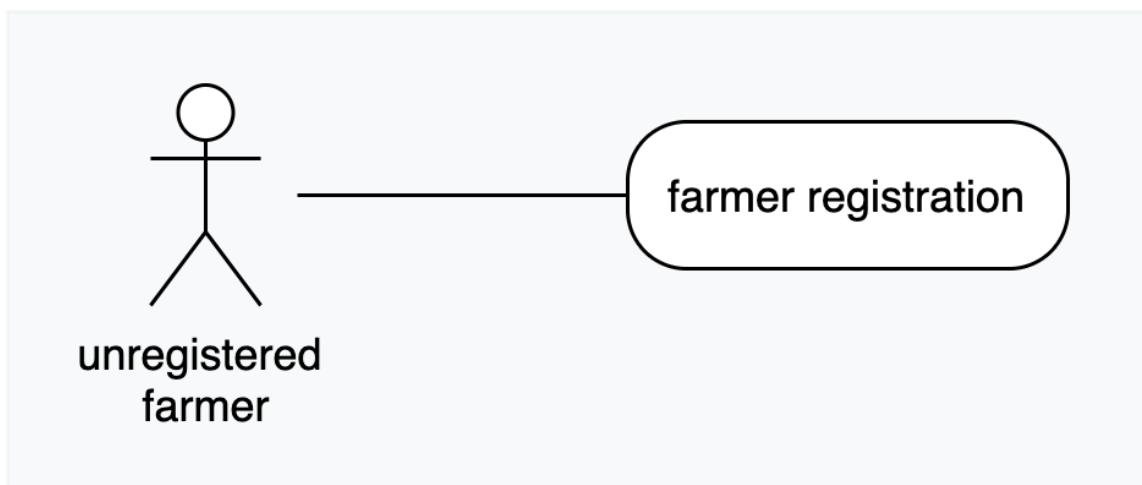


Figure 3.3: Unregistered Farmer | use cases

1	Registration to Dream as Farmer
Actors	Unregistered Farmer
Entry Condition	/
Event Flow	<ol style="list-style-type: none"> 1. The farmer opens the DREAM application. 2. The farmer presses on the "Registration" button. 3. The farmer fills in every mandatory fields on himself/herself. 4. The farmer presses on the "Confirmation" button. 5. The system stores the information about the farmer.
Exit Condition	Farmer is registered in the system. The system saves all of his/her information.
Exception	<ol style="list-style-type: none"> 1. Farmer uses an email or a username already present in the system. 2. Farmer inserts invalid information. <p>The exception is notified to the user and he's returned to the step where the error occurred: 3.</p>
Special Requirements	/

Unregistered Policy Maker

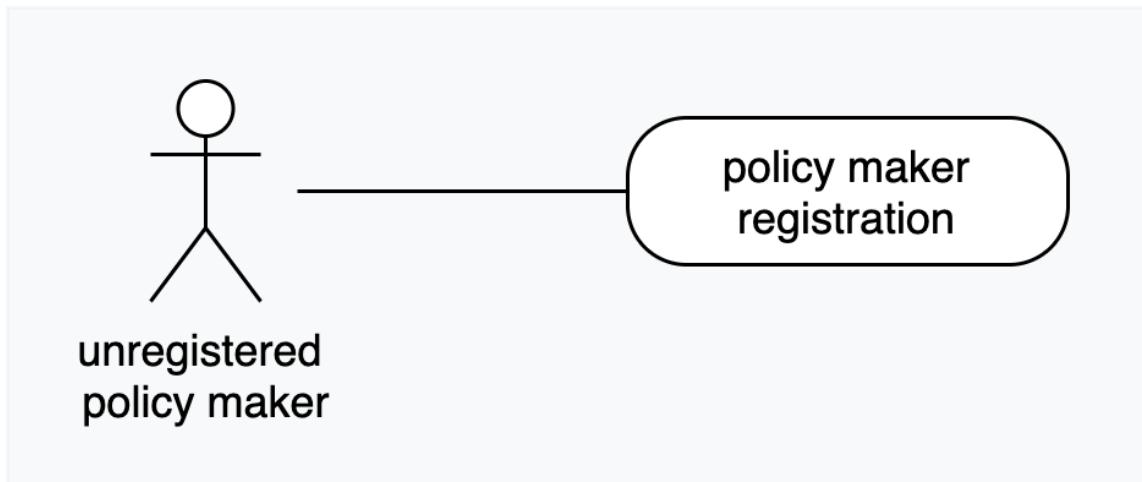


Figure 3.4: Unregistered Policy Maker | use cases

2 Registration to Dream as Policy Maker	
Actors	Unregistered Policy Maker
Entry Condition	/
Event Flow	<ol style="list-style-type: none"> 1. The policy maker opens the DREAM application. 2. The policy maker presses on the "Registration" button. 3. The policy fills in every mandatory fields on himself/herself. 4. The policy presses on the "Confirmation" button. 5. The system stores the information about the policy maker.
Exit Condition	Policy maker is registered in the system. The system saves all of his/her information.
Exception	<ol style="list-style-type: none"> 1. Policy maker uses an email or a username already present in the system. 2. Policy maker inserts invalid information. <p>The exception is notified to the user and he's returned to the step where the error occurred: 3.</p>
Special Requirements	/

General User

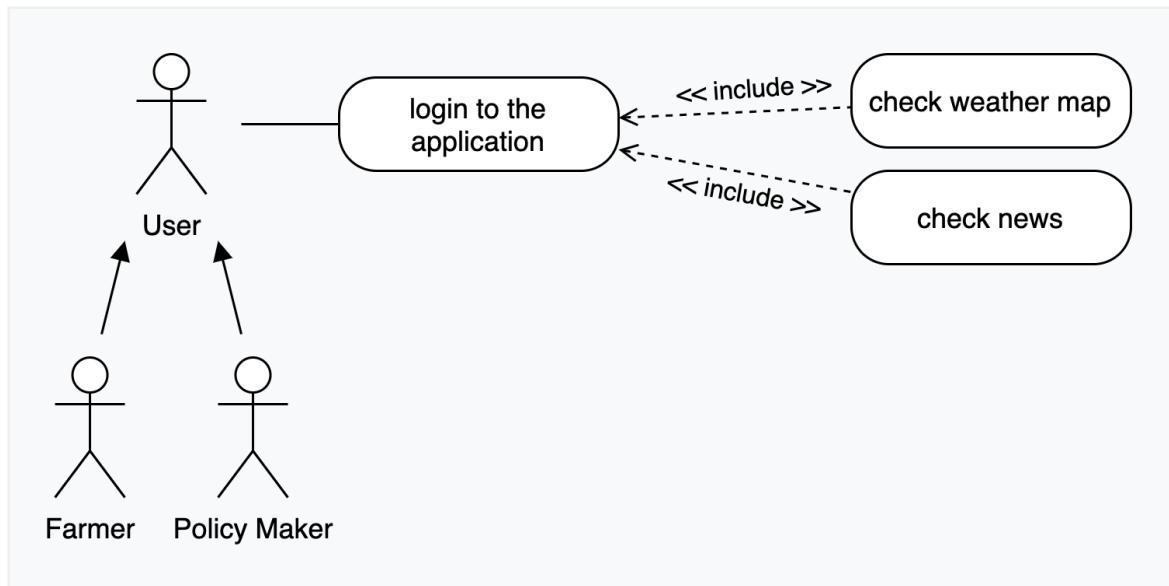


Figure 3.5: General User | use cases

3	Login into the application
Actors	Farmer, Policy Maker
Entry Condition	Policy maker/Farmer must be registered
Event Flow	<ol style="list-style-type: none"> 1. The farmer/policy maker opens the DREAM application. 2. The farmer/policy maker presses on the "Login" button. 3. The farmer/policy maker inserts his/her username and password. 4. The farmer/policy presses the "Confirmation" button.
Exit Condition	The farmer/policy maker is logged in.
Exception	<ol style="list-style-type: none"> 1. Farmer/Policy maker insert wrong username or password. The exception is notified to the user and he's returned to step 3.
Special Requirements	/

4	Check weather map
Actors	Farmer, Policy maker
Entry Condition	Policy maker/Farmer must be logged in the system
Event Flow	<ol style="list-style-type: none"> 1. The farmer/policy selects an area. 2. The farmer/policy selects a start and an end date. 3. The farmer/policy clicks "show weather forecast".
Exit Condition	Policy maker/Farmer has visualized the map.
Exception	<ol style="list-style-type: none"> 1. Farmer/Policy maker inserts a date which forecast is not available. 2. Farmer/Policy maker inserts a date which is too far from the current date either in the past or in the future. <p>The exceptions are notified to the user and he's returned to step 2.</p>
Special Requirements	/

5	Check news
Actors	Policy maker, Farmer
Entry Condition	Policy maker/Farmer must be logged in the system.
Event Flow	<ol style="list-style-type: none"> 1. The farmer/policy presses the "see more News" button in the application. 2. The farmer/policy opens the news he/she wants to be informed about. 3. The farmer/policy reads the news.
Exit Condition	The farmer/policy maker have read the news.
Exception	/
Special Requirements	/

Farmer

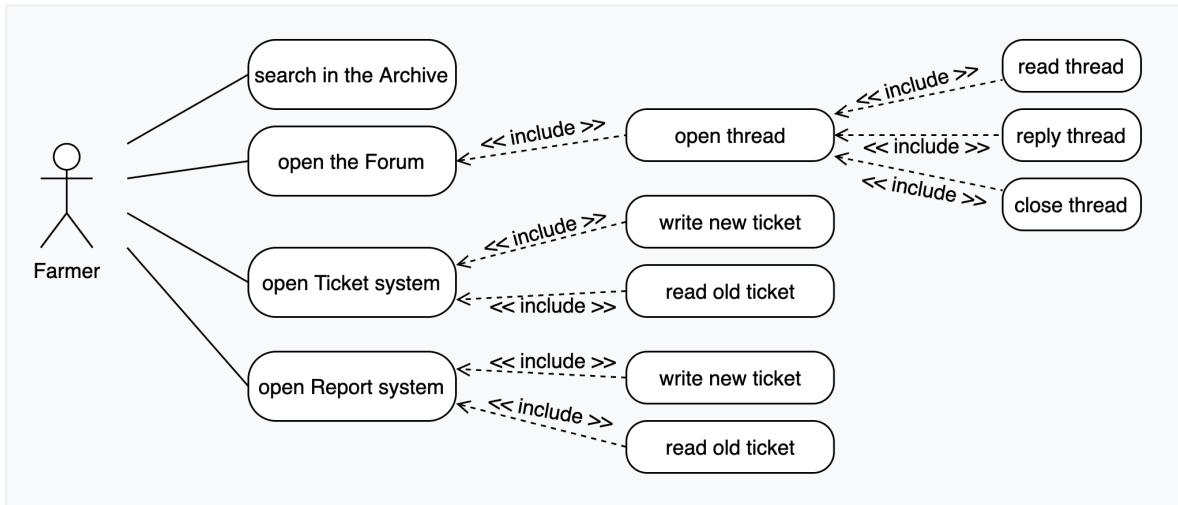


Figure 3.6: Farmer | use cases

6	Search in the Archive
Actors	Farmer
Entry Condition	Farmer must be logged in the system.
Event Flow	<ol style="list-style-type: none"> 1. The farmer presses the "Archive" button in the application. 2. The farmer writes in the appropriate input box what he/she wants to search inside the archive. 3. The farmer analyzes the results he/she obtains after the research. 4. The farmer clicks on the result he/she wants to be informed about. 5. The farmer reads the information about what he/she searched inside the archive.
Exit Condition	The farmer finds the result he/she wanted and read its data.
Exception	<ol style="list-style-type: none"> 1. Farmer search for something which is not present inside the archive. 2. Farmer misspells the name of the object he/she is looking for. <p>The exception is notified to the user and he's returned to step 2.</p>
Special Requirements	/

7	Open the Forum
Actors	Farmer
Entry Condition	Farmer must be logged in the system.
Event Flow	<ol style="list-style-type: none"> 1. The farmer presses the "Forum" button in the application. 2. The farmer searches a thread (if it doesn't exists): <ol style="list-style-type: none"> (a) The farmer can creates the thread 3. The farmer searchs a thread (if it exists): <ol style="list-style-type: none"> (a) The farmer reads the thread. (b) The farmer can reply to the thread. (c) The farmer can close the thread if he/she has created it.
Exit Condition	The farmer's operation succeeded.
Exception	<ol style="list-style-type: none"> 1. Farmer tries to delete a thread which is not been created by him/her. <p>The exception is notified to the user and he's returned to step 3.</p>
Special Requirements	/

8		Open Ticket System
Actors	Farmer	
Entry Condition	Farmer must be logged in the system.	
Event Flow	<ol style="list-style-type: none"> 1. The farmer presses the "Ticket" button in the application. 2. (alt 1) The farmer writes a new Report: <ol style="list-style-type: none"> (a) The farmer inserts a title for the ticket. (b) The farmer chooses a category for the ticket. (c) The farmer writes the body of the ticket. (d) The farmer presses the "Send" button. 3. (alt 2) The farmer open an old Report (if it exists): <ol style="list-style-type: none"> (a) The farmer selects the old ticket. (b) The farmer reads the old ticket. (c) The farmer reads the answer (if it's available). 	
Exit Condition	<ol style="list-style-type: none"> 1. (alt 1) The ticket has been correctly sent. 2. (alt 2) The ticket has been displayed correctly. 	
Exception	<ol style="list-style-type: none"> 1. (alt 1) Farmer doesn't compile all the mandatory fields of the ticket. The exception is notified to the user and he/she's returned to step 2. 	
Special Requirements	/	

9	Open Report System
Actors	Farmer
Entry Condition	Farmer must be logged in the system.
Event Flow	<ol style="list-style-type: none"> 1. The farmer presses the "Report" button in the application. 2. (alt 1) The farmer writes a new Report: <ol style="list-style-type: none"> (a) The farmer selects the id of one of his/her farms. (b) The farmer fills all the mandatory fields of the report. (c) The farmer presses the "Send" button. 3. (alt 2) The farmer opens an old Report (if it's exist): <ol style="list-style-type: none"> (a) The farmer selects the id of one of his/her farms. (b) The farmer reads the old report. (c) The farmer reads the answer (if it's available).
Exit Condition	<ol style="list-style-type: none"> 1. (alt 1) The report has been correctly sent. 2. (alt 2) The report has been displayed correctly.
Exception	<ol style="list-style-type: none"> 1. (alt 1) Farmer doesn't compile all the mandatory fields of the report. The exception is notified to the user and he/she's returned to step 2.
Special Requirements	/

Policy Maker

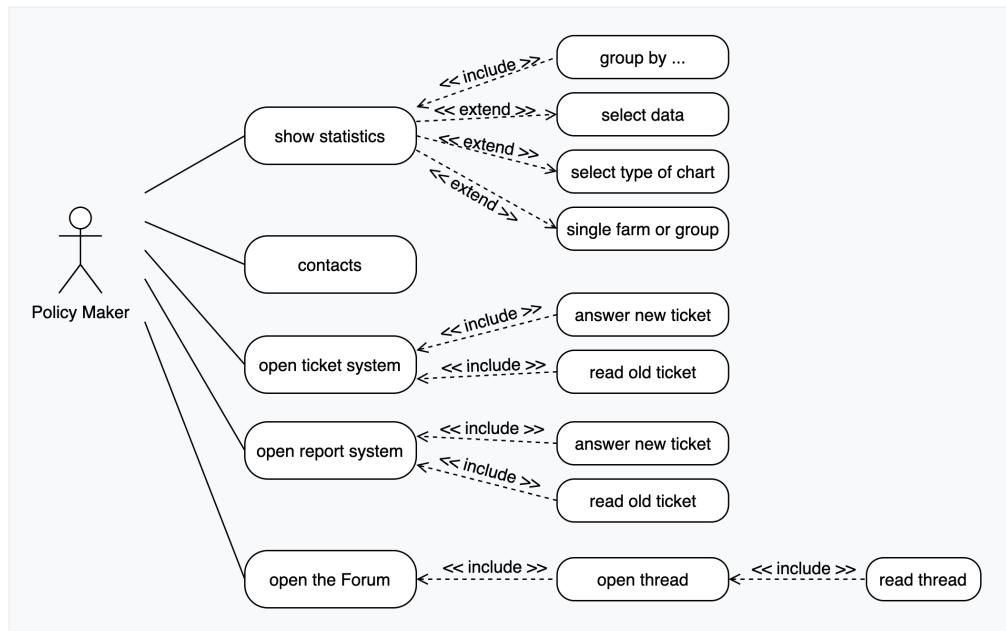


Figure 3.7: Policy Maker | use cases

10	Show statistics
Actors	Policy maker
Entry Condition	Policy must be logged in the system.
Event Flow	<ol style="list-style-type: none"> 1. The policy maker presses the "Analysis" button 2. (opt - otherwise all his/her competence area is selected): <ol style="list-style-type: none"> (a) The policy maker selects farmer identifier. 3. The policy maker inserts a start and an end date. 4. The policy maker selects the type of data to display. 5. The policy maker selects the type of chart to display. 6. The policy maker presses the "Show" button.
Exit Condition	The policy maker correctly visualizes the data he wanted to see.
Exception	<ol style="list-style-type: none"> 1. Policy maker doesn't find data about a specific farmer. 2. Policy maker doesn't find data about a specific date range. <p>The exception is notified to the user and he's returned to step 2.</p>
Special Requirements	/

11	Contacts
Actors	Policy maker
Entry Condition	Policy must be logged in the system.
Event Flow	<ol style="list-style-type: none"> 1. The policy maker presses the "Contacts" button. 2. The policy maker visualizes all of his/her contacts. 3. The policy maker press on the name of one of his/her contacts. 4. The policy maker visualizes name, surname, telephone number, email and the list of the properties of the contact.
Exit Condition	The policy maker correctly visualizes the contact and his/her data.
Exception	/
Special Requirements	/

12	Open Ticket System
Actors	Policy maker
Entry Condition	Policy must be logged in the system.
Event Flow	<p>1. The policy maker presses the "Ticket" button.</p> <p>2. (alt 1) The policy maker visualizes the new tickets which have been sent by farmers to him/her (if they exist):</p> <ul style="list-style-type: none"> (a) The policy maker presses on a ticket in order to answer to it. (b) The policy makes writes the answer to a specific ticket. (c) The policy maker presses the "send" button. <p>3. (alt 2) The policy maker visualizes the old, already answered, tickets which have been sent by farmers to him/her.</p>
Exit Condition	<p>1. (alt 1) The answer to a ticket has been sent correctly.</p> <p>2. (alt 2) The old ticket is correctly visualized.</p>
Exception	<p>1. (alt 1) Policy maker doesn't fills the body of the answer of the ticket. The exception is notified to the user and he/she's returned to step 2.b.</p>
Special Requirements	/

13		Open Report System
Actors	Policy maker	
Entry Condition	Policy maker must be logged in the system.	
Event Flow	<p>1. The policy maker presses the "Report" button</p> <p>2. (alt 1) The policy maker visualizes the new reports which have been sent by farmers to him/her (if they exist):</p> <ul style="list-style-type: none"> (a) The policy maker presses on a report in order to view all the data. (b) The policy maker presses the "Reply" button (c) The policy maker writes the answer in order to provide the analysis of the report sent by farmer. (d) The policy maker presses the "Send" button. <p>3. (alt 2) The policy maker visualizes the old, already answered, reports which are sent by farmers to him/her (if they exist).</p>	
Exit Condition	<p>1. (alt 1) The answer to a report has been sent correctly.</p> <p>2. (alt 2) The old report is correctly visualized.</p>	
Exception	<p>1. (alt 1) Policy maker doesn't fills the body of the answer of the report. The exception is notified to the user and he/she's returned to step 2.c.</p>	
Special Requirements	/	

14	Open the Forum
Actors	Policy maker
Entry Condition	Policy must be logged in the system.
Event Flow	<ol style="list-style-type: none"> 1. The Policy maker presses the "Forum" button in the application. 2. The policy maker searchs a thread. 3. The policy maker reads the thread (if it exists).
Exit Condition	The policy maker complete correclty the read operation.
Exception	<ol style="list-style-type: none"> 1. Policy maker searches a thread which does not exist. The exception is notified to the user and he's returned to step 2.
Special Requirements	/

3.2.6. Sequence Diagrams

In this subsection some sequence diagrams are presented in order to better understand how the steps presented in the use cases are supposed to be implemented in the application. In this part the system is presented as two separate components, DREAM APP and DREAM, to have a lower and more technical description. It's possible to consider DREAM APP as a static interface of the application, while DREAM represents more the business logic.

Anyway in the Design Document there will be presented a more complete description of the system and of its components.

Login Farmer

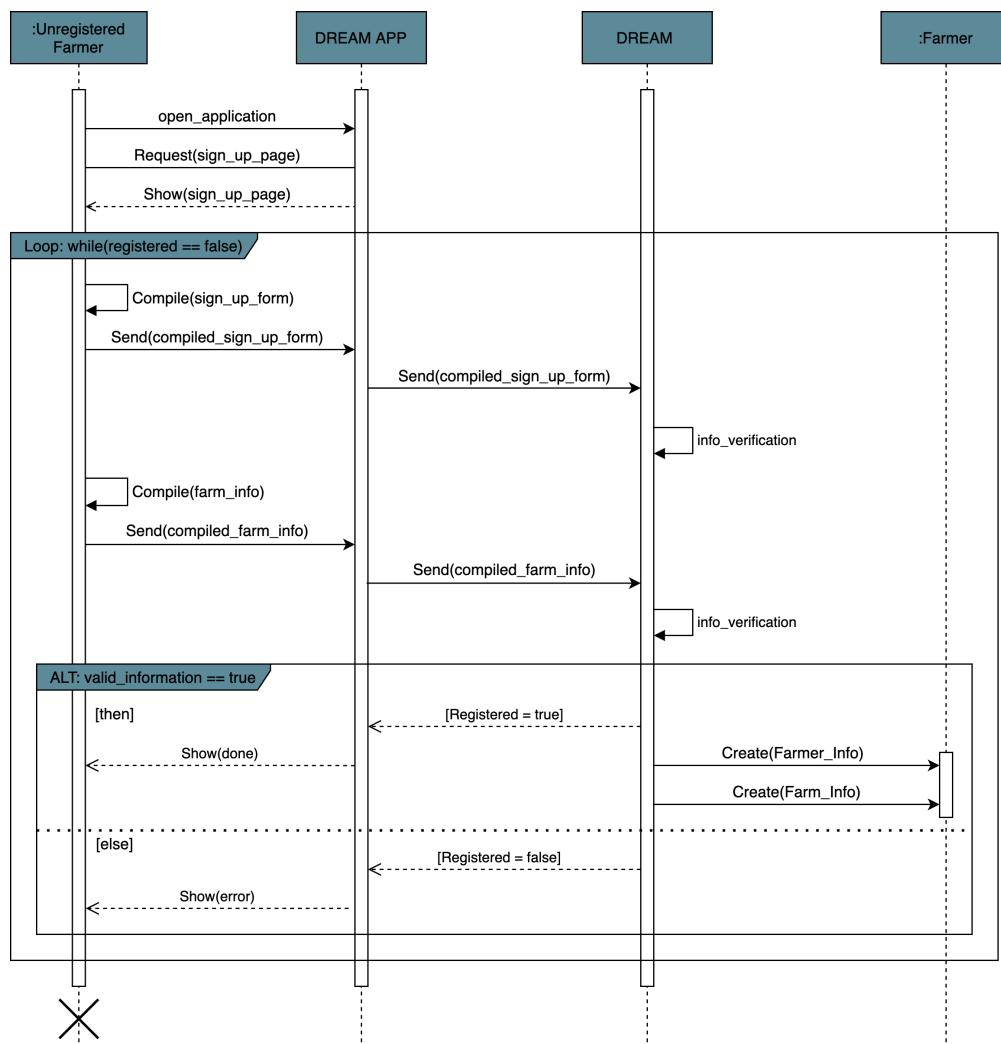


Figure 3.8: Login Farmer

Policy Maker Registration

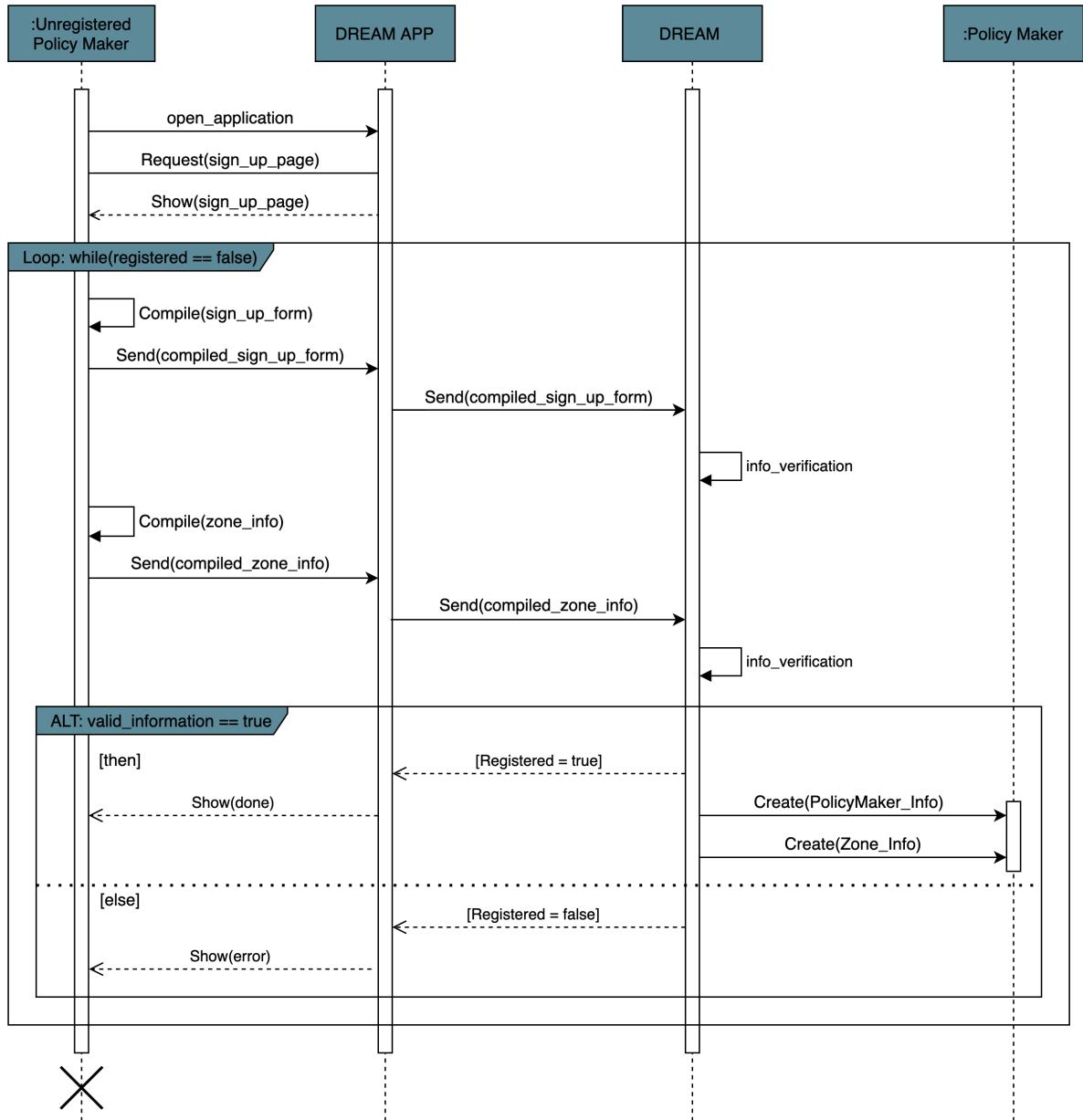


Figure 3.9: Policy Maker Registration

Login To The Application

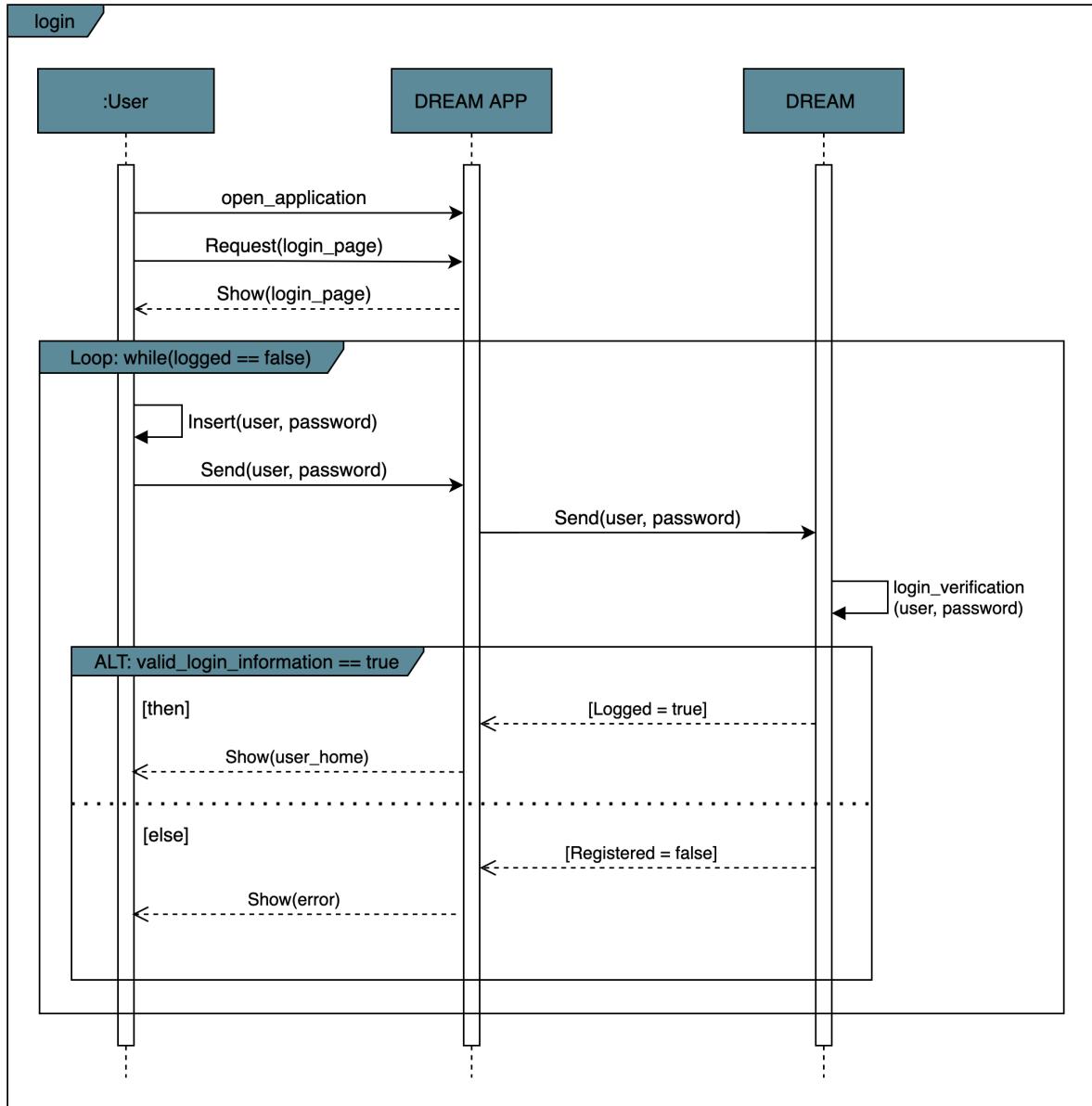


Figure 3.10: Login To The Application

User general functionalities

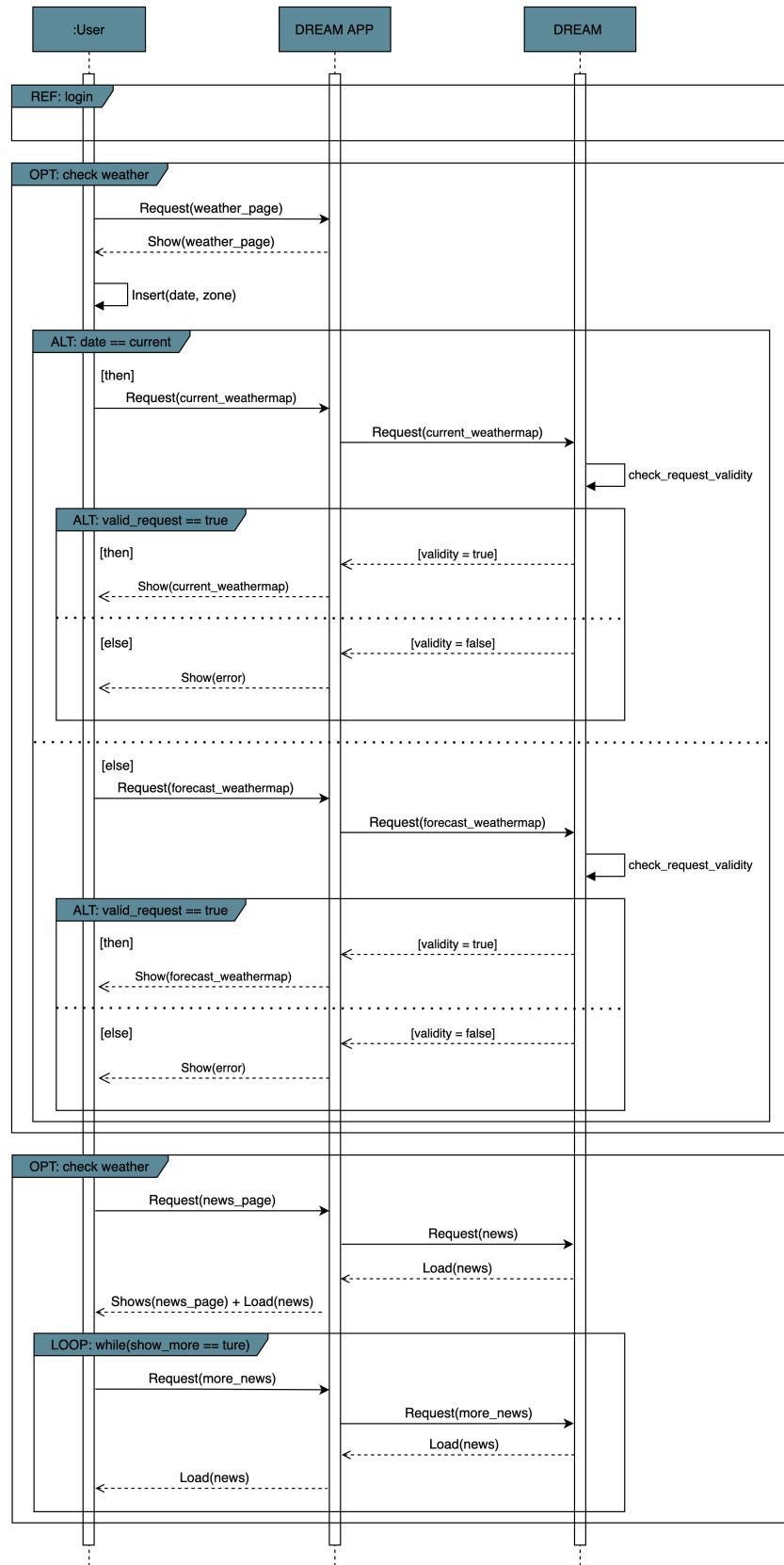


Figure 3.11: User general functionalities

Farmer Utilities

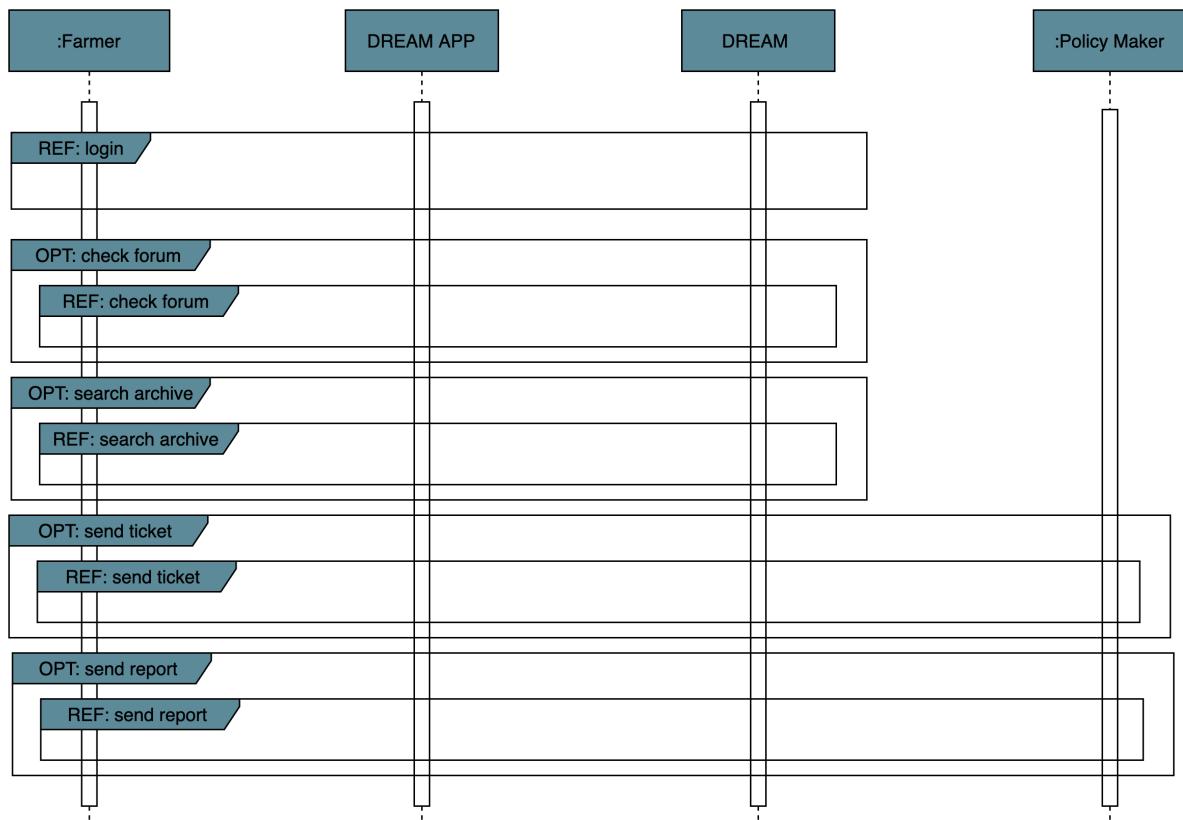


Figure 3.12: User general functionalities

Policy Maker Utilities

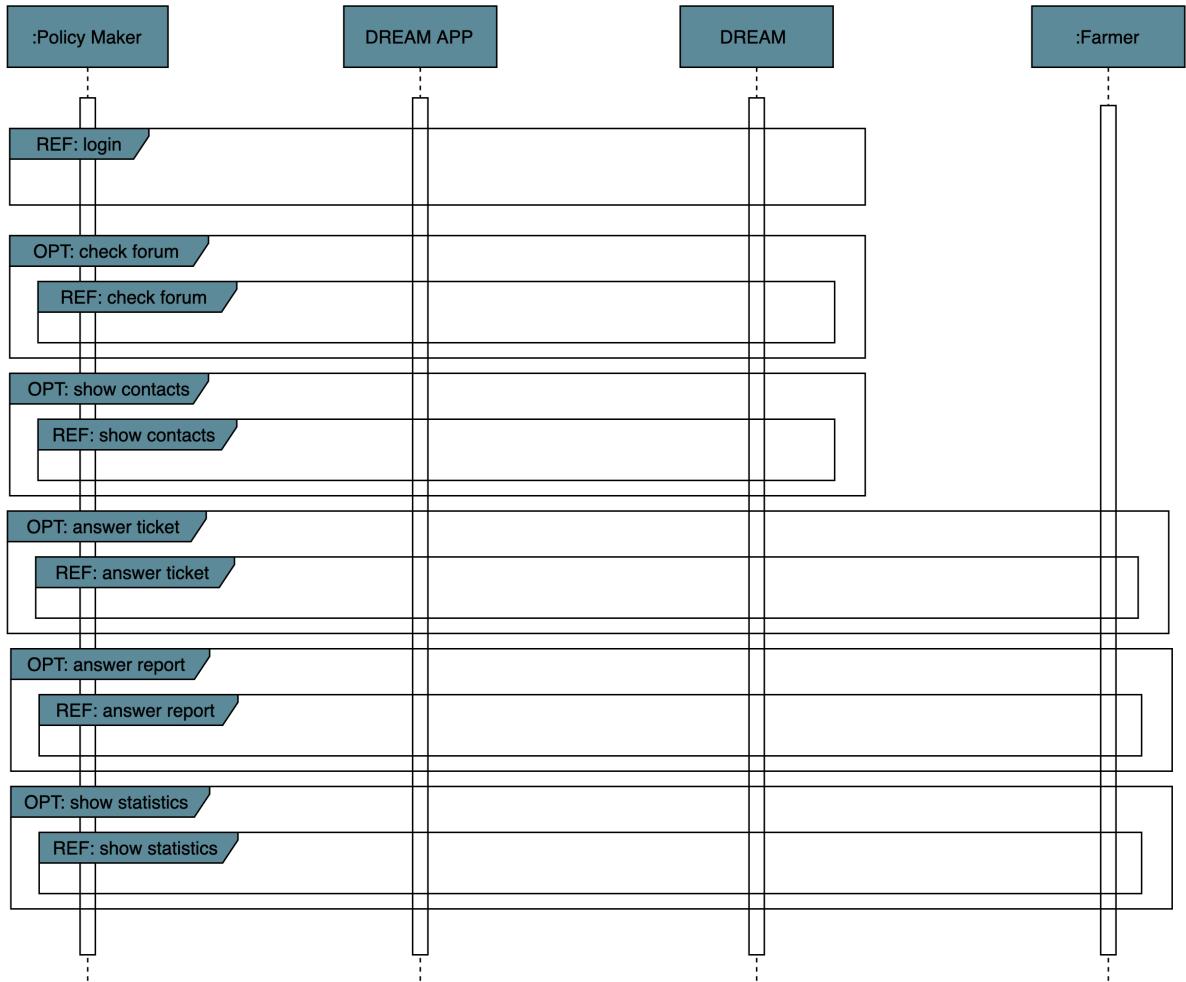


Figure 3.13: Policy Maker Utilities

In the sequence diagram, for the sake of brevity, on the checking process (from the DREAM "object") the validity of the request will be omitted. This functionalities will be the same as the one described in previous sequence diagrams.

The references REF parts of the previous diagrams are here showed. Just few of them are explicitly represented, in order to point out the most relevant behaviour and to avoid the repetitions of same structures.

Search Archive

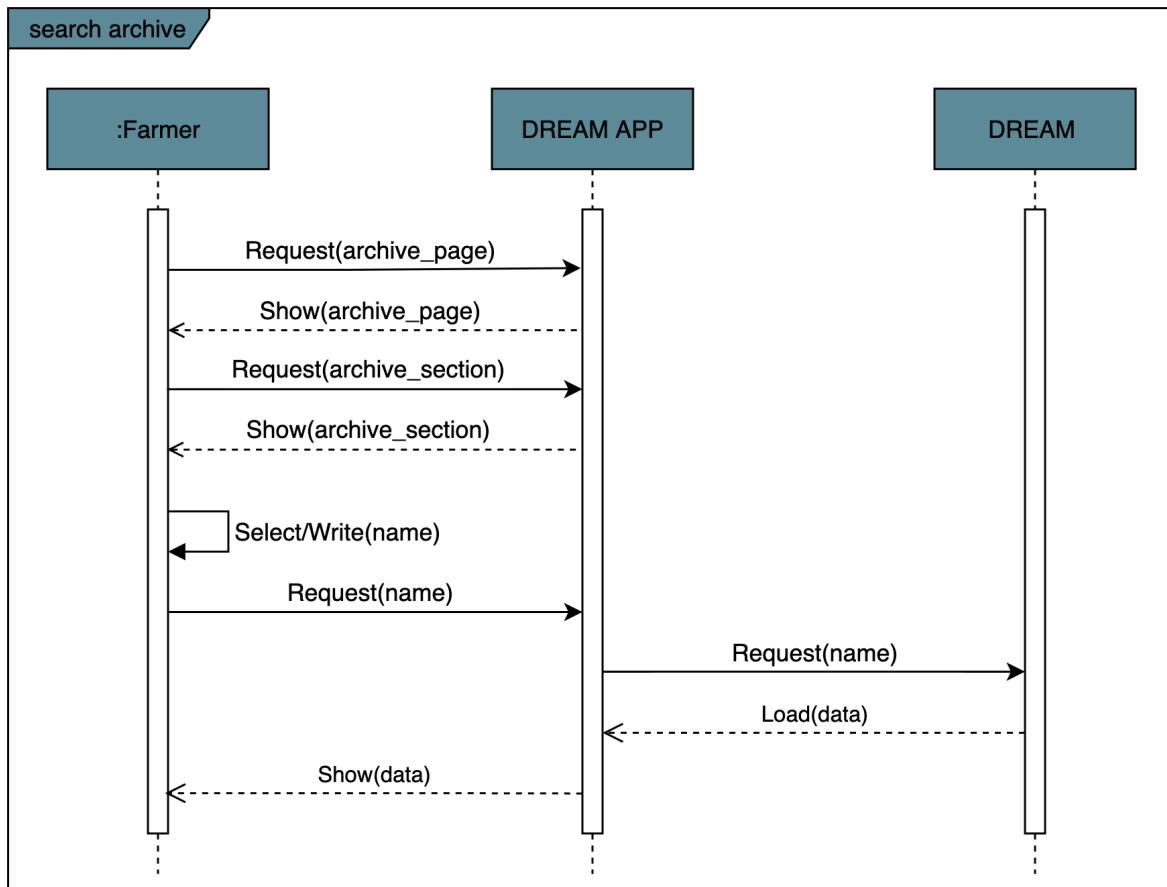


Figure 3.14: Search Archive

Send ticket

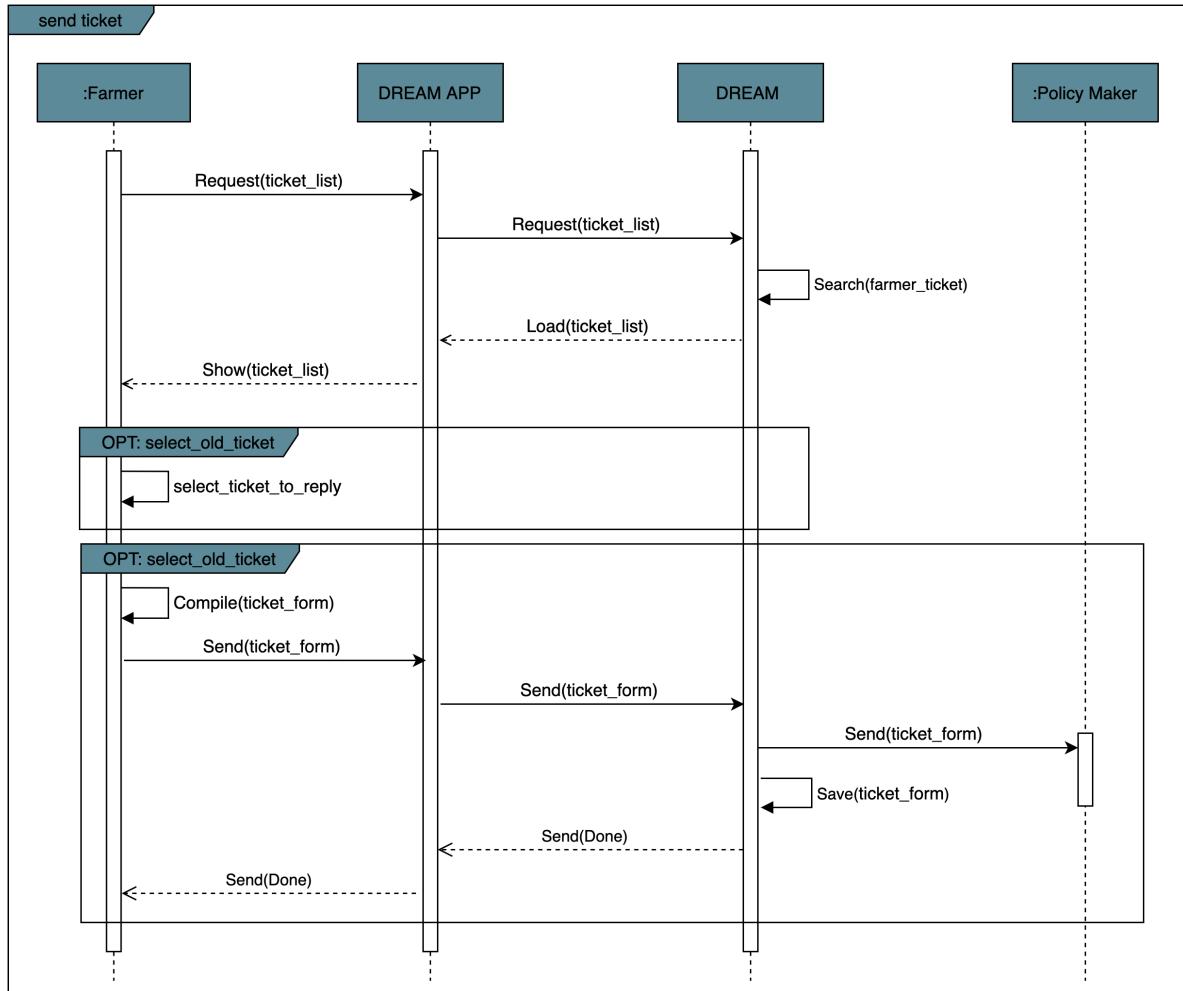


Figure 3.15: Send ticket

Answer ticket

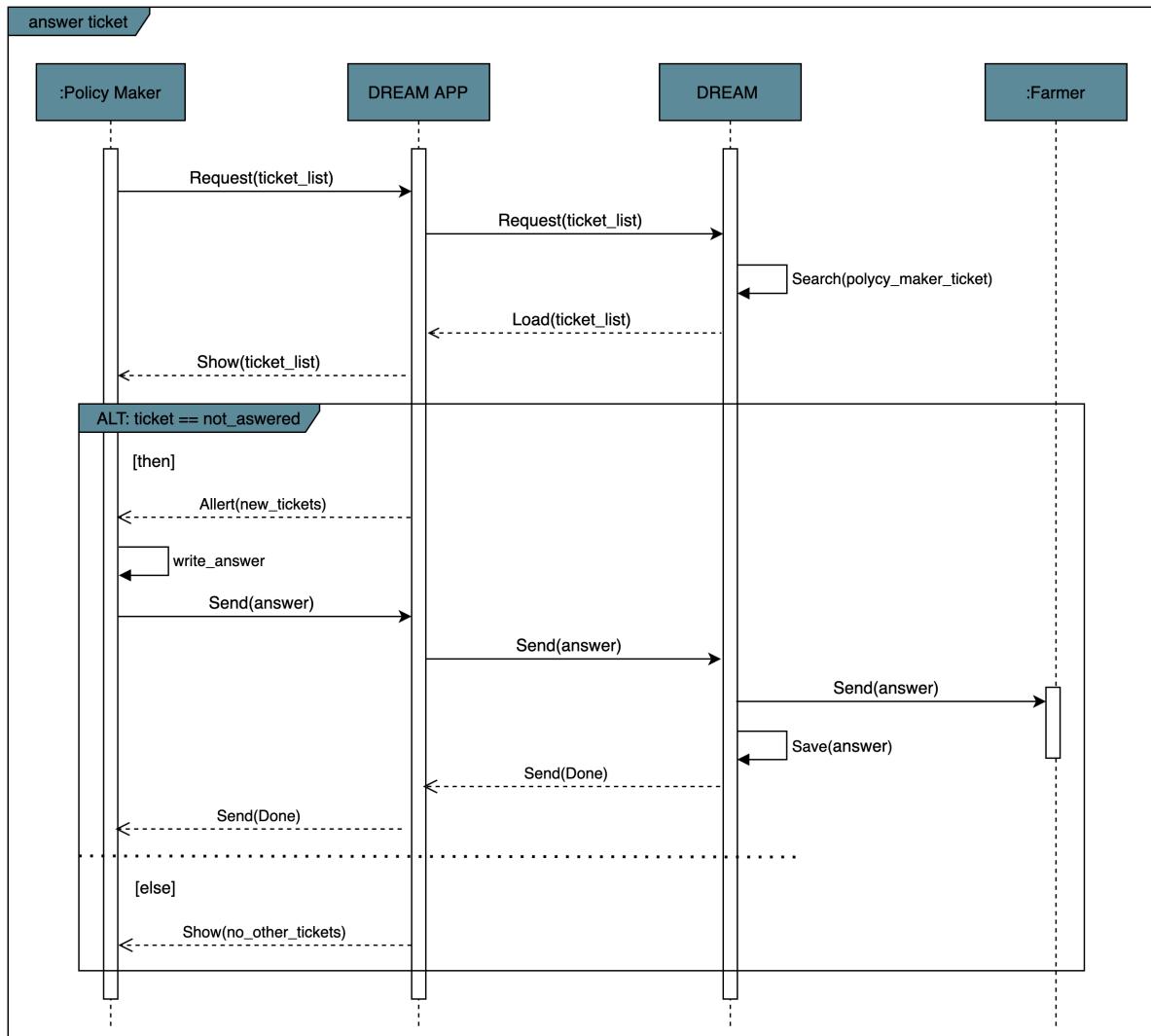


Figure 3.16: Answer ticket

Show analysis

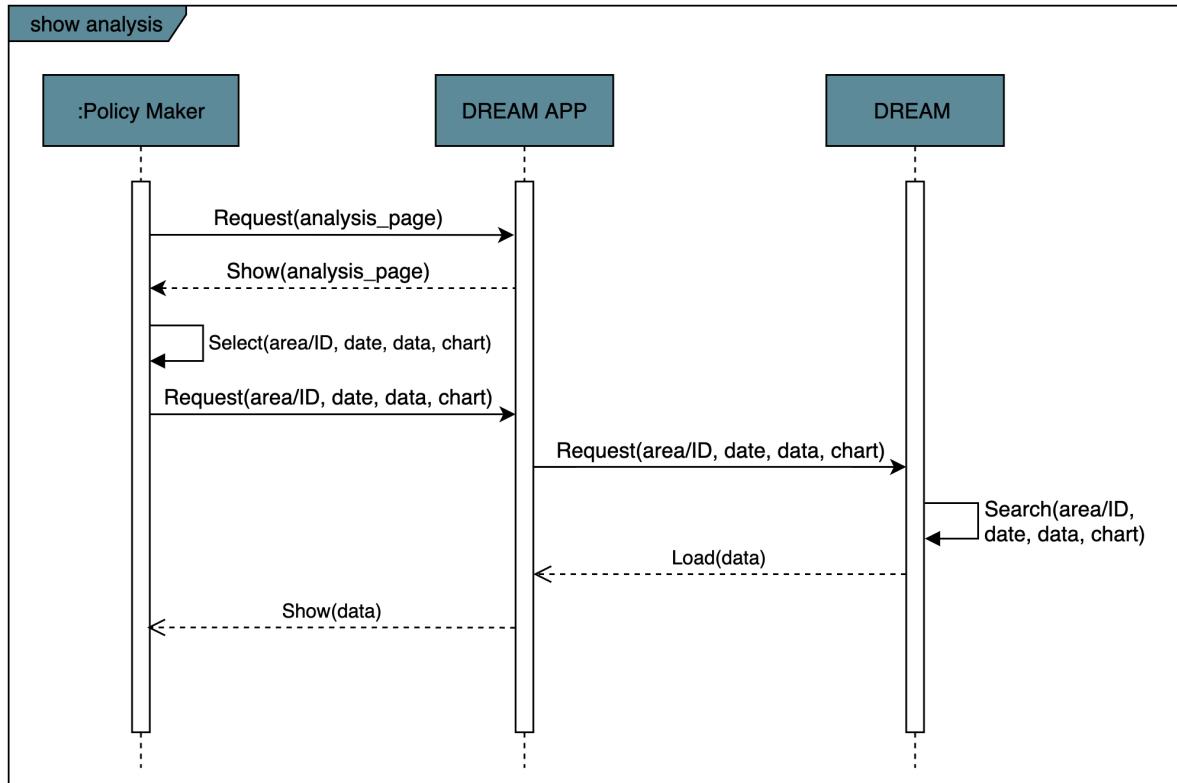


Figure 3.17: Show analysis

3.3. Performance Requirements

- The system should be available 99.5% of the time.
- The response time of a research in the archive should last a maximum of 4 seconds.
- The response time of a research of a news should last a maximum of 4 seconds.
- The load time of the weather map should last a maximum of 3 seconds.
- The refresh time of the weather map should last a maximum of 3 seconds.

3.4. Design Constraints

3.4.1. Standards compliance

- The system must require the farmer the permission to retrieve data regarding the position, economical situation, land properties and farm's organization.

- The system must manage the data retrieved from the farmers in respect with the privacy laws.
- The system must keep data anonymous if they are used for external analysis or public disclosure.

3.4.2. Hardware limitations

- **user:**

- Mobile network connection to access the service.
- GPS to request precise information.
- Enough storage space for the execution of the application.

- **server:**

- Stable network connection to keep the server up.
- Enough processing power in order to handle all the requests made simultaneously by users.
- Enough storage space for the execution of the application.

3.5. Software System Attributes

3.5.1. Reliability

Due to the presence of a great amount of sensible data the system provides duplicates of the components. In addition to this it should be safe implementing a storage system which implies the redundancy of data with the aim to avoid data losses.

3.5.2. Availability

There's should be copies of the different components of the system in order to have the possibility to keep the service up while there is the necessity to make maintenance operations.

3.5.3. Security

There are many important data give by the farmers like their landing data or they geographical position which must be protected from a potential attack directed to the

database. It is necessary a system of passwords which must be encrypted and in case of password recovery, this must never be sent in clear. To avoid sniffing and spoofing the communication of important data throughout the application must be done through the usage of some sort of encryption.

3.5.4. Maintainability

The system is developed by using the best practices and modalities of software engineering in order to maintain it and have the possibility to expand its functionalities.

3.5.5. Portability

The application is developed in order to be compatible with the majority of the versions of iOS and android for smartphone, it will be possible to use it in the future versions of these two and potentially it will be compatible with new emergent operating systems which could expand their presence in the country.

4 | FORMAL ANALYSIS USING ALLOY

In this chapter alloy analysis will be used in order to demonstrate the validity of the model, pointing out some peculiarities hard to catch through the previous UML representations.

4.1. Alloy Model

The following details will show the hierarchy that characterized our model, more precisely we use alloy in order to highlight the link between the areas and their composition and the respective supervisors.

4.1.1. Results

```
Executing "Run show for 30"
Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20 Mode=batch
417415 vars. 10170 primary vars. 756053 clauses. 1288ms.
Instance found. Predicate is consistent. 388ms.
```

Figure 4.1

4.1.2. Graphic Model

In the first figure it is represented the most general view of our model, in particular there are the links between the "Ticket", "Farmer" and policy maker which represent the communication between them. For simplicity in this case we assume that there's always the presence of a report between a farmer and a policy maker.

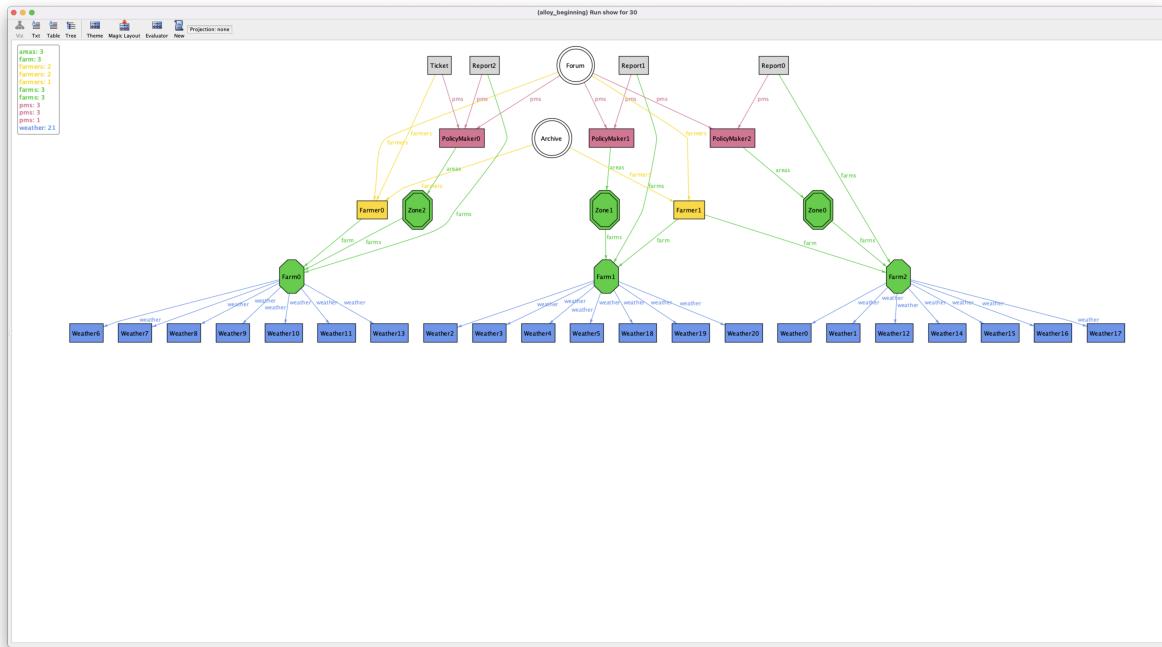


Figure 4.2

In the second figure there is the same representation as before but in this case there is also the possibility that a ticket doesn't obtain an answer from a policy maker.

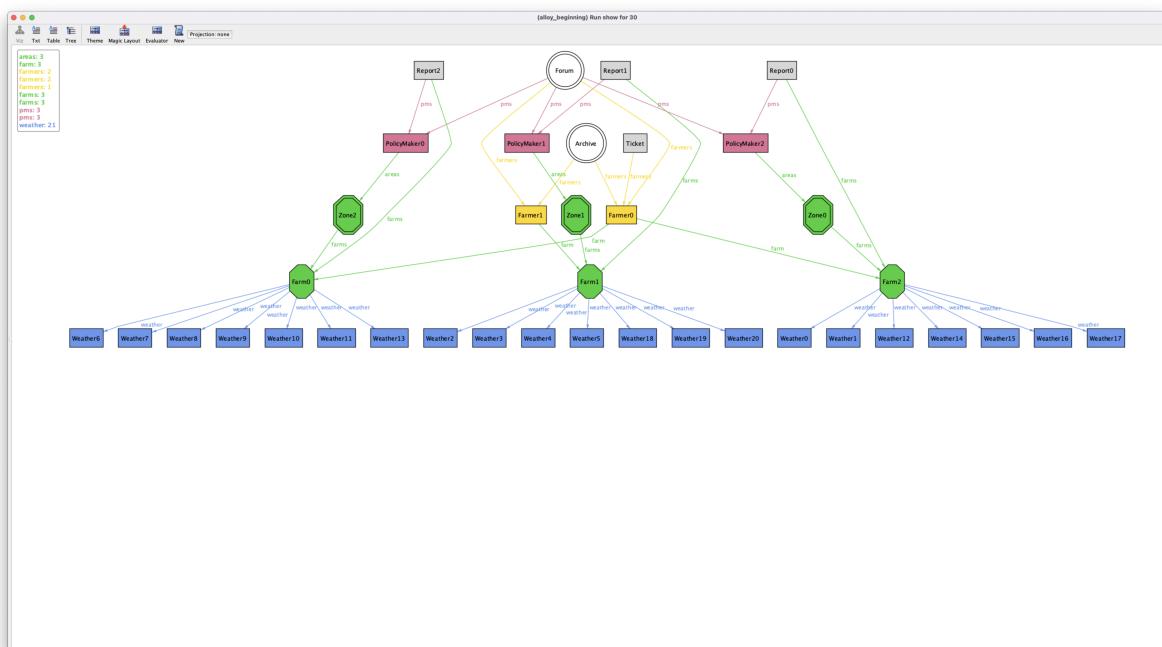


Figure 4.3

4.2. Code

```

//Represents a set of farm
sig Zone{
    farms: some Farm,
}

//Represents the policy maker
sig PolicyMaker{
    areas: one Zone,
}

//Represents the farmer
sig Farmer{
    farm: some Farm,
}

//Represent the farm
sig Farm{
    weather: set Weather
//Weather = 7 in order to represents the 7 days of the week
}{#weather = 7}

//Represents the weather of one day of the week
sig Weather{

}

//Represents the archive linked with all the farmers
sig Archive{
    //every Forum has a set of Farmer
    farmers: some Farmer
}

sig Report{
    //a report obtained by a PolicyMaker
    pms: one PolicyMaker,
    //There is a report for every farm owned by a farmer
    farms: one Farm
}

//The forum is unique and can be seen by everyone
sig Forum{
    //a Forum has a set of PolicyMaker
    pms: some PolicyMaker,
    //a Forum has a set of Farmer
    farmers: some Farmer
}

//Multiple tickets can be sent by farmers to the respective policy makers
sig Ticket{
    //chiedere per il rapporto tra ticket , farmers s policymakers
    //a report compiled by a farmer
    farmers: one Farmer,
}

```

```

//a report obtained by a PolicyMaker
pms: lone PolicyMaker
}

//The ticket must be sent to a policy maker who is responsible for the area where the
//→ farmer has his/her farm
pred sameTicketFarmerPm[pm: PolicyMaker ,far: Farmer,ti: Ticket] {
    pm in ti.pms and far in ti.farmers
}
fact "same_ticket_per_pm_and_farmer"{
    all pm: PolicyMaker , f: Farmer , ti: Ticket | sameTicketFarmerPm[pm,f,ti] implies f.
    //→ farm = pm.areas.farms
}

//The report must be sent to a policy maker who is responsible for the area where the
//→ farmer has his/her farm
pred sameReportFarmerPm[pm: PolicyMaker ,far: Farm,re: Report] {
    pm in re.pms and far in re.farms
}
fact "same_report_per_pm_and_farmer"{
    all pm: PolicyMaker , f: Farm , re: Report | sameReportFarmerPm[pm,f,re] implies f =
    //→ pm.areas.farms
}

//For semplicity we assume that a policy maker can receive one report for each farm
fact "one_PolicyMaker_Per_Report"{
    all pm: PolicyMaker | one r: Report | pm in r.pms
}

//Every farm is assigned to a report which will be sent to the respective policy maker
fact "one_report_per_farm"{
    all far: Farm| one r: Report| far in r.farms
}

//Every farm is associated with 7 weather signature to indicate the 7 days of the week
fact "7_weathers_per_farm"{
    all w: Weather | one f: Farm | w in f.weather
}

//A farm can be owned by only one farmer
fact "one_farmer_per_farm"{
    all farms: Farm | one f: Farmer | farms in f.farm
}

//We assume that there aren't intersection between areas
fact "no_2_area_for_the_same_farm"{
    all farm: Farm | one a: Zone | farm in a.farms
}
//Every area is associated with one policy maker
fact "one_pm_per_area"{
    all area: Zone | one pm: PolicyMaker | area in pm.areas
}
//The forum is unique for every policy maker
fact "one_forum_per_pm"{
    all pm: PolicyMaker | one f: Forum| pm in f.pms
}

```

```
}

//The forum is unique for every farmer
fact "one_forum_per_farmer"{
    all far: Farmer | one f: Forum | far in f.farmers
}

//The archive is unique for every farmer
fact "oneArchive_per_farmer"{
    all far: Farmer | one a: Archive | far in a.farmers
}

pred show{
    #Forum = 1
    #Archive = 1
    #Farm = 3
    #Farmer = 2
}

run show for 30
```

5 | EFFORT SPENT

The time here reported is an estimation.

Student 1: Brunello Simone

Topic	Hours
General Reasoning	9
Purpose	2
Scope	4
Scenarios	3
Static Information Model	3
Dynamic Class behaviour Model	2
Product Functions	2
User characteristics	0.30
Domain Assumptions	2
Software Interfaces	1
Functional Requirements	6
Use Cases	5
Sequence Diagrams	2
Performance, Design and Attributes	1
Alloy	3
Document Organisation	1

Student 2: Nicolis Nicholas

Topic	Hours
General Reasoning	9
Purpose	2
Scope	4
Scenarios	2
Static Information Model	3
Dynamic Class behaviour Model	5
Product Functions	1
User characteristics	0.30
Domain Assumptions	1
User Interfaces	4
Software Interfaces	0.30
Functional Requirements	6
Use Cases	5
Sequence Diagrams	5
Performance, Design and Attributes	1
Alloy	1
Document Organisation	1

6 | REFERENCES

6.1. Reference Documents

- Specification document: Assignment RDD A.Y. 2021-2022
- Course slides
- wikipedia

6.2. Reference Sites

- <https://www.accuweather.com/>
- <https://www.diagram.net/>
- <https://www.canva.com/>
- <https://www.figma.com/>
- <https://www.overleaf.com/>
- <https://github.com/Angtrim/alloy-latex-highlighting>

7 | CORRECTIONS

- **Section 1.2:** The API acronym was added to the table
- **Section 1.2:** Splitted the table of 'Abbreviations' into tables of 'Acronyms' and 'Abbreviations'.
- **Section 3.2:** Change the title of the table of goal n° 6 because it was the same as the one in table of goal n° 5.
- **Section 3.2:** In the 'Mapping Requirements on Goals' and in the 'Mapping explicit table' tables in correspondence of goal n°6 requirements n°23 and n°24 need to be added. We specify here this error instead of modifying all the tables.
- **Section 3.2:** The requirements n° 38 is a repetition of the requirement n° 37, this error propagates also in the table of chapter 3.2.4.. We highlighted the error instead of changing all the tables in the document.