POLITECNICO
MILANO 1863

DRAM

RASD

Requirement Analysis and Specification Document

AA 2021-2022

Version: 1.0

Date: 21 December 2021

1 Introduction....4
1.1 Problem and purpose....4
1.1.1 Description of a given problem....4
1.1.2 Purpose....4
1.2 Scope....4
1.2.2 World phenomena....5
1.2.2 Shared phenomena....5
1.3 Definitions and abbreviations....6
1.3.1 Definitions....6
1.3.2 Abbreviations....6
1.3.3 Goals....7
1.4 revision history....7
1.5 Reference document....7
1.6 Document structure....7
2 Overall description....8
2.1 Production perspective....8
2.1.1 Scenarios....8
2.1.2 Class Diagram....10
2.2 Production functions....10
2.3 User characteristics....11
2.4 Assumptions, dependencies, and constraints....11
3 Specific requirements....12
3.1 External requirement....12
3.1.1 Hardware....12
3.1.2 Software....12
3.2 Functional requirements....12
3.2.1 List of Requirements....12
3.2.2 Mapping on goal....13
3.2.3 Use case diagrams....13
3.2.4 Use cases....15
3.2.5 Sequence diagram....20
3.2.6 Activity diagrams....25
3.2.7 Mapping on requirements....26
3.3 Performance requirements....27
3.4 Design constraints....27
3.5 Software system attributes....28
3.5.1 Reliability....28
3.5.2 Availability....28
3.5.3 Security....28
3.5.4 Maintainability....28
3.5.5 Portability....28
4 Formal analysis using Alloy....29
4.1 Code....29

4.2 Result....33
4.3 Generated instances....33
5 Effort spend....35
6 References....36

1 Introduction

1.1 Problem and purpose

1.1.1 Description of a given problem

The pandemic of the COVID-19 has had a huge impact on the world. Because the COVID-19 is spread in population movements, governments in various countries have generally adopted restrictions on travel, restrictions on population movements, and cargo transportation to minimize the spread of the virus.
However, using these methods will cause food supply problems in some marginalized communities

1.1.2 Purpose

Establish a digital system to collect various data related to agricultural products, such as types of agricultural products, water volume, soil moisture, and other data. Through analysis, and the results of the analysis are sent to farmers, agronomists. Cooperate with various departments to deal with food supply shocks and challenges.

1.2 Scope

To enable the system to collect, analyze data, feedback data, and implement data-driven policy formulation methods. Need to involve multiple stakeholders, the policy Manufacturers, farmers, and agronomists.

For policy manufacturers, they need to use the data feedback to them by the system to formulate corresponding policies and modify existing policies to adapt to the current food supply situation.

For farmers, they need to upload the type and quantity of their products to the system. The system will give them information about weather forecasts, fertilizer types, and other personalized suggestions to help farmers increase yields.
For agronomists, mainly help/guide farmers. The agronomist will work out a corresponding guidance plan based on the data. At the same time, they will interact with farmers in the discussion forums.

1.2.2 World phenomena

World phenomena	Description
Wp1	The Farmer enters the farm
Wp2	Agronomist enters the farm to guide
Wp3	The Farmer leave the farm
Wp4	Agronomists leave the farm
Wp5	policymakers make policy

1.2.2 Shared phenomena

Shared phenomena	Description	Control
Sp1	The Farmers entered the system	World control
Sp2	The Farmers choose the type of food they grow	World control
Sp3	The Farmers write about the amount of food they grow	World control
Sp4	The Farmers receive notification from the system-about weather forecast	Machine control
Sp5	The Farmers receive notification from the system about the types of fertilizers	Machine control
Sp6	The Farmers receive notification from the system about the personalized advice	Machine control
Sp7	The Farmers ask questions on the forum	World control
Sp8	The Agronomist answering questions on the forum	World control
Sp9	The agronomist receives data from the system	Machine control
Sp10	The agronomist formulates a corresponding guidance plan based on the data	World control

	fed back by the system	
Sp11	The policymakers receive data from the system	Machine control
Sp12	The policymakers formulate corresponding policies based on system data	World control
Sp13	The soil moisture sensor automatically uploads data to the system	Machine control

1.3 Definitions and abbreviations

1.3.1 Definitions

Definition	Description
Interval time-about sensor	Set how often the sensor uploads data
Weather forest	Get information through IT provider and get weather forecast
Personalized suggestions,	Personalized suggestions, the agronomist draws corresponding suggestions based on the analysis of statistical data and sends them to the farmers
Interval time -about the weather forecast	Obtain the latest weather forecast data from IT suppliers regularly
Visualization	Convert data into tables or pictures to view
Release time	The time the farmer/agronomist posted the message on the forum
discussion forums	Provide a platform for interactive communication between agronomists and farmers

1.3.2 Abbreviations

Abbreviation	Description
WP	World phenomena
SP	Shared phenomena
G	Goal
D	Domain assumption
R	Requirement
IT	Information Technology
RASD	Requirement analysis and specification document

1.3.3 Goals

Goals	Description
G1	The system combines the information is provided by other IT providers, and the data generated by senor
G2	The system allows farmers to manage the data about their products
G3	The system allows policymakers to monitor the performance of all farmers
G4	The system allows farmers to discuss in the forums
G5	The system allows users to manage their profile
G6	The system allows users to communicate by sending notifications(i.e., farmers send requests for problem solutions to agronomists or other farmers)

1.4 revision history

Version	Date	Description
1.0	22/12/2021	First version

1.5 Reference document

- Specification document: Assignment RDD AY 2021-2022
- IEEE/ISO/IEC 29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering - Life cycle processes - Requirements engineering
- Course slides.

1.6 Document structure

1. Section1

About introducing the purpose and scope of this system, analyzing the world model and sharing model. At the same time, explain the terms defined in the system. For the abbreviations used, write the full name.

2. Section2

Describe the whole system from the product perspective, product functions. Drawing the class diagram and state chart. Describe the user characteristics, to show the different types of the user. And define some domain assumptions, Assume the problems that may arise in the operation of the system.

3. Section3

It is about the specific requirements. From the external interface requirements, it has two aspects. One is hardware interfaces, and another is software interfaces. For the function requirement, we use a table to describe all the possible functions of the system. And drawing some diagrams to show more details about the data transfer and the user's cases. With the functional requirement, to design the sequence diagram and activity diagram. Finally, it also adds some constraints to this system to make the system more stable.

4. section4

Using alloy language to reach the specific constraints which are not possible to show in the class diagram

5. Section5

Record the time required for each person to complete the module

6. Section6

Reference

2 Overall description

2.1 Production perspective

2.1.1 Scenarios

1. Telengana's policymakers check the performance of all farmers

There are two main tasks that a policymaker should carry out. The first one is the check the performance of the farmers, another one is to understand the suggestions that are provided by agronomists are helpful or not.

A policymaker Steve login into the system to check the performance of farmers who plant crops in the area that is administered by policymaker Steve. For the farmers who perform well, policymaker Steve will offer special incentives and ask them to provide useful practice to other farmers. By contrast, for the farmers who have a particularly bad performance, policymaker Steve will identify these farmers.

Also, policymaker Steve needs to understand whether the suggestions are provided by agronomists result in positive effects or not through checking the agronomists' suggestions and the yield of farmers that the suggestion towards.

2. Farmers check the data related to them

There are several points of information farmers need to be checked in their daily workday, and all these points of information are based on individual farmers' location and the type of crops.

A farmer logins into to system to check the data. First is the weather forecast, since the weather plays a fundamental role in the production of crops. The next one is personalized suggestions concerning their plan to plant the specific crop and the fertilizer to use.

3. Farmers upload the data related to production

Farmers upload the data relevant to the job he was done. Such as the amount of water he sprinkled, the fertilizers he used, and the time he acted, the weight of harvest.

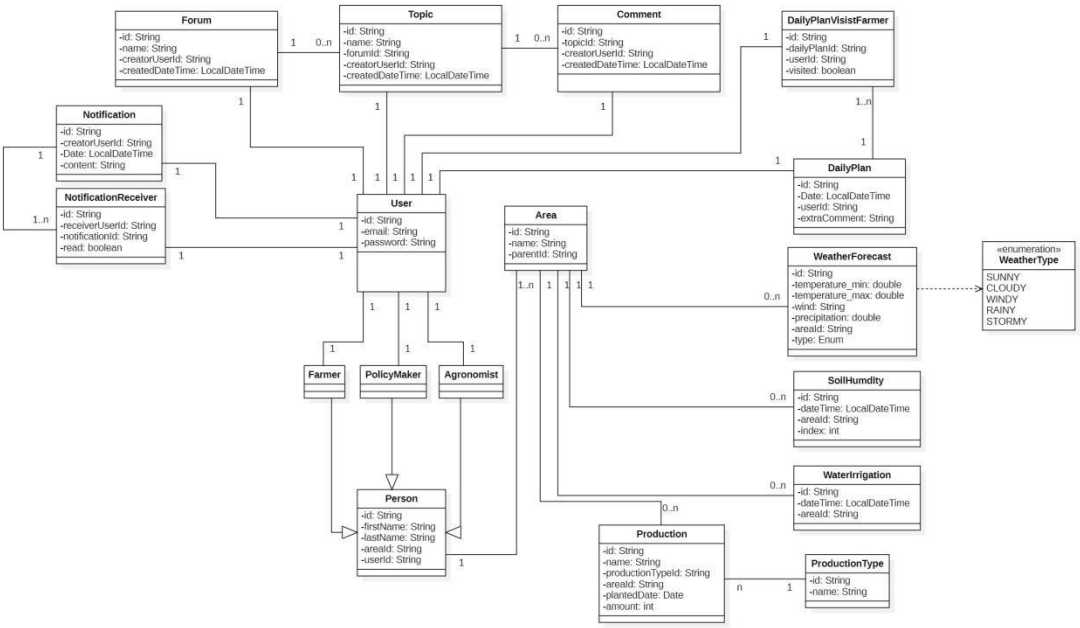
4. Farmers report any issue they encountered

Farmers may have some problems they can't solve by themselves. They report these problems to the specific agronomist who is responsible for the area that the farmers locate. Or they can publish the topic in the forums to discuss with other farmers.

5. Farmers create forums with other farmers

Farmers can create forums to discuss with other farmers. They can browse the forums if they have access to the internet, also, they can comment on the topic or other comments to discuss the problem they have.

2.1.2 Class Diagram



Note:

- The main purpose to show weather class and weather type enumeration in the diagram is for including all the information that the system needs. However, from my perspective, there are many weather APIs on the internet, we can use these APIs to get the weather forecast instead of saving weather forecast information on our server.
- The tables of policymaker, farmer, agronomist, and user information are divided into 4 tables instead of putting together into one table. From my perspective, each role's information is saved separately can lead to several benefits. The first one is that the code of information management and system support code can be divided into business and system modules. The second one is that it is easier to modify the table structure when policymaker, farmer, agronomist has their field that has to be added.

2.2 Production functions

The system has four main functions

- **Data visualization**: policymakers, farmers, and agronomists need to see the data they are concerned with. For policymakers, they need to find out the farmers who perform well or badly by checking those farmers that data they inserted. For farmers, they need to check the data related to them, such as the weather forecast. Moreover, based on their location and type of crops, showing individualized suggestions concerning specific crops they plant and specific fertilizers to use. For agronomists, they need to check their daily plan to visit the farmers they need to.
- **Data management**: the system ought to manage the data that users generate while

they are using it. For farmers, they need to insert the data about their product and any problem they face, also, they interact with other farmers or agronomists, they need to insert the area they are responsible for, update the daily plan and receive and replay the request from farmers. These data should be managed by the system. What's more, some data does not generate by users, but plays fundamental roles in the system, like the weather forecast, water consumption info from the water irrigation system, the humidity of the soil, which also should be managed by the system.

- **Forum**: it is the area for farmers to discuss the problems to face.
- **Notification**: it is the function for users to send notifications. For policymakers, they can send notifications to reward farmers who perform well or publish policies.

2.3 User characteristics

- **Policymaker**

A person who monitors the performance of farmers and agronomists in the area they are ruling

- **Farmer**

A person who responsible for crop production

- **Agronomist**

A person who takes care of the production situation of farmers in the area he/her responsible for, and gives suggestions base on the individual situation.

2.4 Assumptions, dependencies, and constraints

Domain Assumption	Description
D1	The water irrigation system has access to the internet
D2	The soil humidity sensor has access to the internet
D3	The Soil humidity sensor can deal with the task of data transmission
D4	A policymaker can rule at least one area
D5	An agronomist can monitor at least one
D6	Every farmer has at least one field
D7	Every unregistered user has an email
D8	Every email is only can be used by one user

3 Specific requirements

3.1 External requirement

3.1.1 Hardware

The system is a management and analysis system. Need to have relevant hardware to collect information. Such as soil moisture sensors and water irrigation sensors, these devices are transmitted to the system through the network. In addition, there are handheld terminals, such as mobile apps. Farmers use the terminal to select the type and quantity of plants currently planted. After being aggregated into the system for analysis, the results are returned to the farmer's mobile app to view the required data. For policymakers, because they need to view a lot of data and must consider various factors, they also need computers and other hardware devices to display the data they need through HTML pages

3.1.2 Software

The system will connect with the IT provider through the interface to obtain weather forecasts and other data. It also obtains soil moisture and other water volume information through the sensor interface. To provide different information to different types of users, we need to use an external API to send the specified information to the specified user. For farmers, they need weather forecasts, personalized recommendations, fertilizer types, etc. to increase yields. For policymakers, it is necessary to obtain questions about whether farmers are performing well, whether the guidance of agronomists can achieve significant results, etc., and formulate corresponding policies based on this information.

3.2 Functional requirements

3.2.1 List of Requirements

Requirement	Description
R1	The system allows users to register by using their email
R2	The system allows user login by using the correct account and password
R3	For the first time that the user's login the system, the system will require them to insert the information that the system needs (i.e., first name, last name, area they are responsible for, etc.)
R4	The system allows users to modify their information
R5	The system provides APIs to receive the data from the water irrigation system and save it in local server
R6	The system provides APIs to receive the data from soil moisture of the

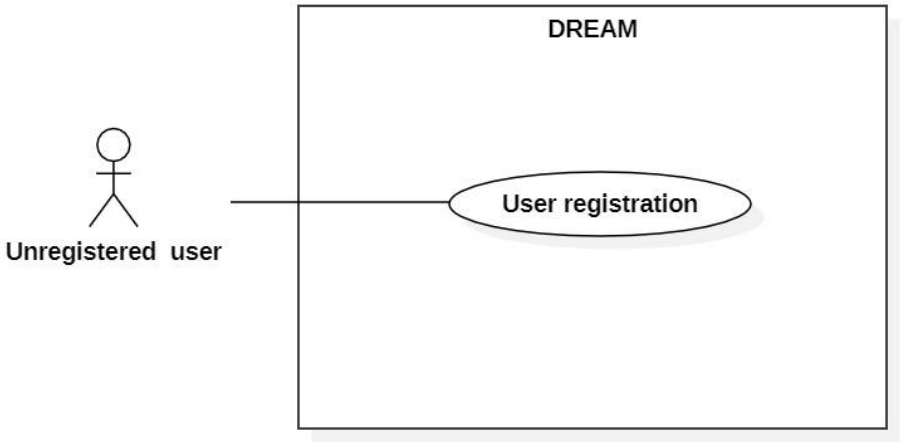
R7	The system allows policymakers to check the statistic of farmers
R8	The system allows policymakers to check the statistic of agronomists
R9	The system allows policymakers to send a notification to farmers telling them that they are rewarded because of outstanding performance
R10	The system allows policymakers to publish policies by sending notifications
R11	The system allows policymakers to check the notifications they sent in previous time
R12	The system allows policymakers to withdraw the notifications if the receivers have not read it
R13	The system allows farmer users to check the weather forecast
R14	The system allows farmer users to check the personalized suggestions which are sent by notification
R15	The system allows farmer users to insert the data about the production
R16	The system allows farmer users to modify the data they inserted before
R17	The system allows farmer users to remove the data they inserted before
R18	The system allows farmer users to check the data about the production they inserted before
R19	The system allows farmer users to check the water irrigation data
R20	The system allows farmer users to check the soil humidity data
R21	The system allows farmer users to report the problems they face
R22	The system allows farmer users to send requests by notifications for asking the help from agronomists or other farmers
R23	The system allows farmer users to receive notifications from agronomists or other farmers reply to their requests
R24	The system allows farmer users to create forums
R25	After farmer users create forums, they can create topics (i.e., problems they encountered)
R26	After farmer users create a topic, all the users can comment below the topic

3.2.2 Mapping on goal

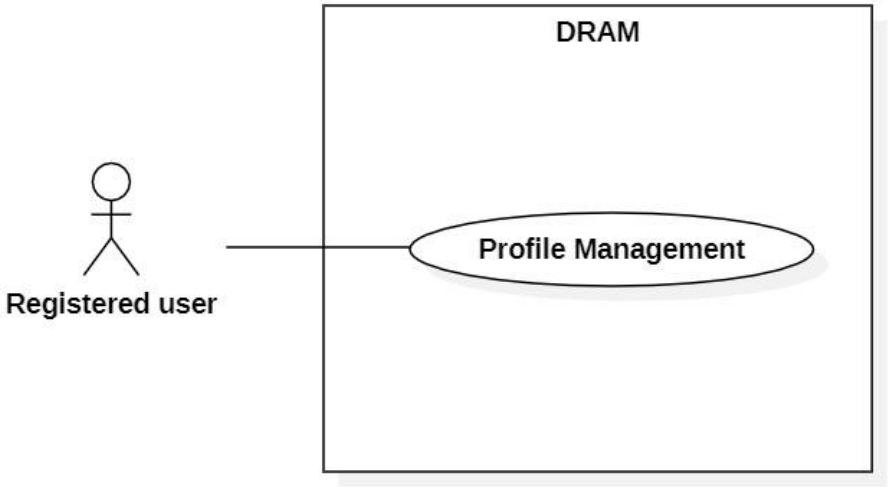
Goal	Domain assumption	Requirement
G1	D1, D2, D3	R5, R6
G2	D6	R13, R14, R15, R16, R17, R18, R19, R20,
G3	D4	R7, R8
G4	D7	R24, R25, R26
G5	D7	R3, R4
G6	D8	R10, R11, R12,R22, R23

3.2.3 Use case diagrams

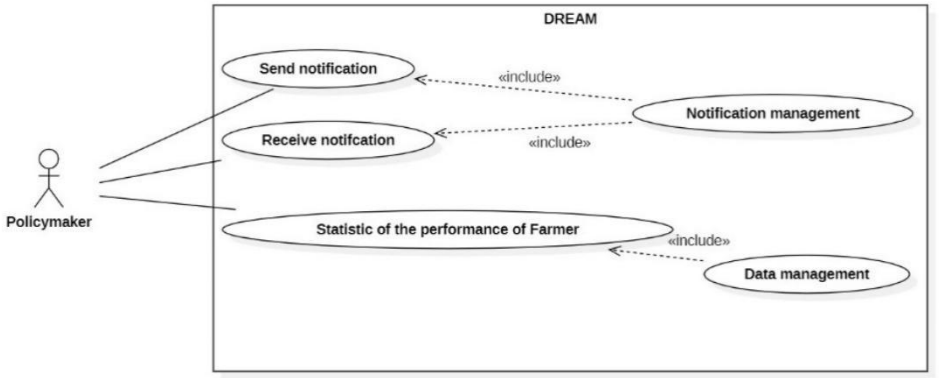
- **Unregistered user**



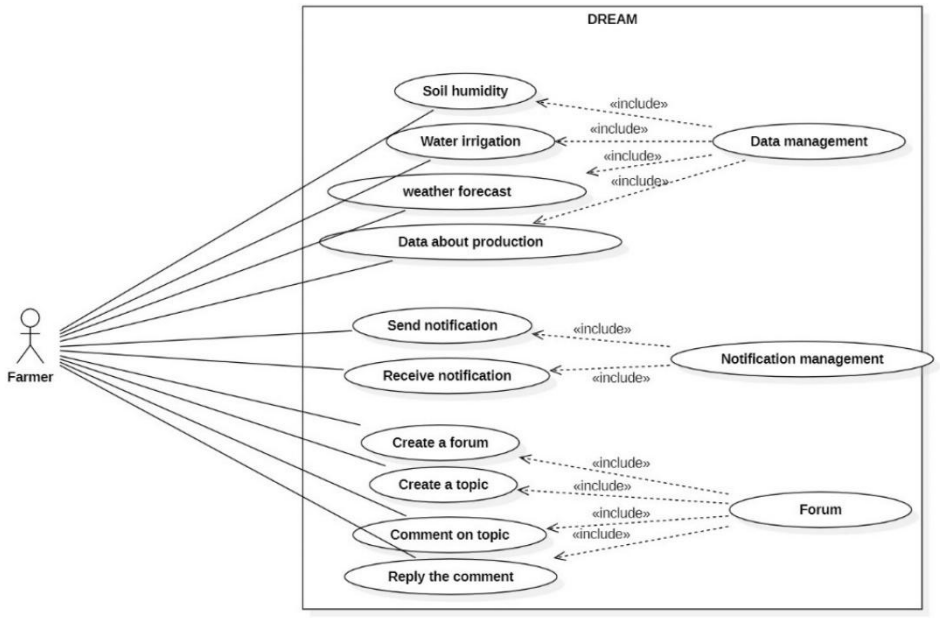
- **Registered user**



- **Policymaker**



- **Farmer**



3.2.4 Use cases

3.2.4.1 User registration

Name	User registration
Actors	Unregistered users
Entry conditions	● The internet is connected
Event flow	1. User clicks the “Register” button 2. User inserts email and password 3. User enters password again for password double-check 4. User clicks “Submit” button 5. The web page shows “Submit successfully” 6. The web page goes forward to the login page
Exit conditions	● The internet is connected
Exceptions	● The internet is not connected ● The email is used ● The password double-check is not passed

3.2.4.2 Modify the location of the field

Name	Modify the location of the field
Actors	Registered users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “My profile” button 2. User clicks the “Modify” button 3. The web page shows the form of the user’s profile to modify 4. User modifies his/her location of the field 5. User clicks the “Submit” button

	6. The web page shows “Submit successfully”
Exit conditions	● The location information and the other information is not null ● The internet is connected
Exceptions	● The internet is not connected ● The information is null ● The location is used by other same type users

3.2.4.3 Check the statistic of performance of the farmer

Name	Check the statistic of performance of the farmer
Actors	Policymaker users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Data management” button 2. The web page shows the statistic of performance of the farmers based on the profile location
Exit conditions	There is no exit condition
Exceptions	● The internet is not connected

3.2.4.4 Send a notification

Name	Send a notification
Actors	Registered users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Send notification” button 2. The web page shows the form of the notification, the user fills the content (i.e., message, receivers, etc.) 3. User clicks the “submit” button 6. The web page shows “Submit successfully”
Exit conditions	The internet is connected
Exceptions	● The internet is not connected ● The content of the notification is blank

3.2.4.5 Check a notification

Name	Check a notification
Actors	Registered users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Check notification” button 2. The web page shows lists of their notification 3. User can click one notification to see the detail
Exit conditions	There is no exit condition
Exceptions	● The internet is not connected

3.2.4.6 Check the weather forecast

Name	Check the weather forecast
Actors	Farmer users
Entry conditions	● The user is already logged in to the system ● The user is already assigned his/her location of the profile

	● The internet is connected
Event flow	1. User clicks the “Weather forecast” button 2. The web page shows the weather forecast for the next 7 days based on the profile location 3. User can choose the specific day the check the detail of that day
Exit conditions	There is no exit condition
Exceptions	● The internet is not connected

3.2.4.7 Check the water irrigation data

Name	Check the water irrigation data
Actors	Farmer users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Data management” button 2. The web page shows the sub-options of “Data management” 3. User clicks the “Water irrigation” button 4. The web page shows the water irrigation of the farmer user’s area
Exit conditions	There is no exit condition
Exceptions	● The internet is not connected

3.2.4.8 Check the soil humidity data

Name	Check the soil humidity data
Actors	Farmer users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Data management” button 2. The web page shows the sub-options of “Data management” 3. User clicks the “Soil humidity” button 4. The web page shows the soil humidity of the farmer user’s area
Exit conditions	There is no exit condition
Exceptions	● The internet is not connected

3.2.4.9 Insert the data about the production

Name	Insert the data about the production
Actors	Farmer users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Data management” button 2. The web page shows the sub-options of “Data management” 3. The user chooses the “Production” button and clicks the “Add” button 4. User fills in the information about the production (i.e., date, production name, production type, etc.) 5. User clicks the “submit” button 6. The web page shows “Submit successfully”
Exit conditions	● The production information is not blank ● The internet is connected
Exceptions	● The internet is not connected ● The production information is blank

3.2.4.10 Modify the data about the production

Name	Insert the data about the production
Actors	Farmer users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Data management” button 2. The web page shows the sub-options of “Data management” 3. The user chooses the “Production” button and clicks the “Modify” button 4. User modifies the information about the production 5. User clicks the “submit” button 6. The web page shows “Submit successfully”
Exit conditions	● The production information is not blank ● The internet is connected
Exceptions	● The internet is not connected ● The production information is blank

3.2.4.11 Search the data about production

Name	Search the weather forecast
Actors	Farmer users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Data management” button 2. The web page shows the sub-options of “Data management” 3. The user chooses the “Production” button, and clicks the “Search” 3. User chooses the specific day or day range he/she wants to search 4. The web page shows the data
Exit conditions	There is no exit condition
Exceptions	● There is no data for the date range that the user selects ● The internet is not connected

3.2.4.12 Create a forum

Name	Create a forum
Actors	Farmer users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Forum” button 2. The web page shows the page of a forum 3. User clicks the “Create a forum” button 4. The web page shows the form for the user to fill to create a forum 5. User fills in the information (i.e., name, description, etc.) 6. User clicks the “Submit” button 7. The web page shows the “A forum is created successfully”
Exit conditions	● The internet is connected ● The information of form is not blank ● The forum name does not exist
Exceptions	● The internet is not connected ● The information of form is blank ● The forum name existed

3.2.4.13 Create a topic

Name	Create a topic
Actors	Farmer users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Forum” button 2. The web page shows the page of a forum 3. User clicks one specific forum which the user wants to create a topic under this forum 4. The web page shows the history topic of this forum 5. User clicks the “Create a topic” 6. The web page shows the form for the user to complete to create a topic 7. User completes the form. 6. User clicks the “Submit” button 7. The web page shows the “A topic is created successfully”
Exit conditions	● The internet is connected
Exceptions	● The internet is not connected ● The information of form is blank

3.2.4.14 Comment on a topic

Name	Comment on a topic
Actors	Farmer users
Entry conditions	● The user is already logged in to the system ● The internet is connected
Event flow	1. User clicks the “Forum” button 2. The web page shows the page of a forum 3. User clicks one specific forum 4. The web page shows the history topic of this forum 5. User clicks the topic that they want to comment 6. The web page shows the history comments of the topic 7. User clicks the “Comment” button 8. The web page shows a form for the user to make a comment 9. User clicks the “Submit” button 10. The web page shows the “submit successfully” 11. The web page refresh automatically, the user can see the comment he/she made
Exit conditions	● The internet is connected ● The information of form is not blank
Exceptions	● The internet is not connected ● The information of form is blank

3.2.4.15 Save water irrigation data

Name	Save water irrigation data
Actors	Water irrigation sensor
Entry conditions	● The internet of the server is connected ● The internet of water irrigation sensor is connected
Event flow	1. The water irrigation sensors push the data to the server through message sequence (i.e., Kafka) 2. The system receives the data, checks the data, and saves it on the

	server
Exit conditions	● The internet is connected
Exceptions	● The internet is not connected

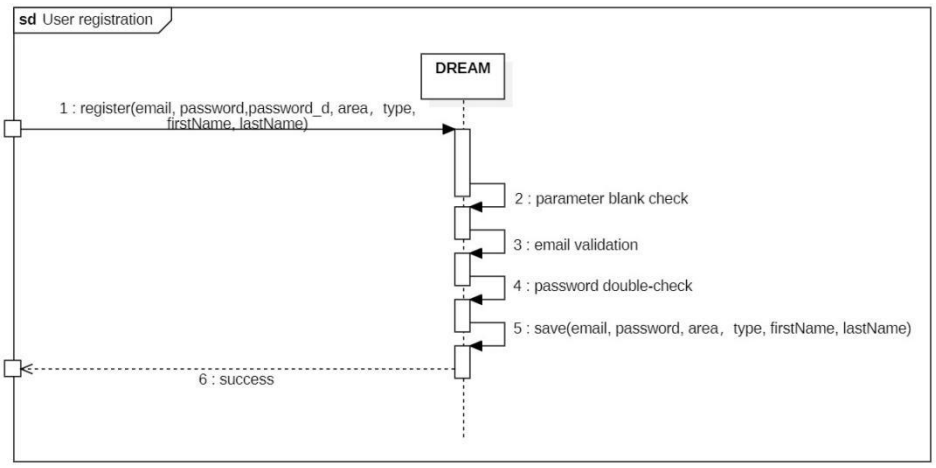
3.2.4.17 Save soil humidity data

Name	Save soil humidity data
Actors	Water irrigation sensor
Entry conditions	● The internet of the server is connected ● The internet of water irrigation sensor is connected
Event flow	1. The soil humidity sensors push the data to the server through message sequence (i.e., Kafka) 2. The system receives the data, checks the data, and saves it on the server
Exit conditions	● The internet is connected
Exceptions	● The internet is not connected

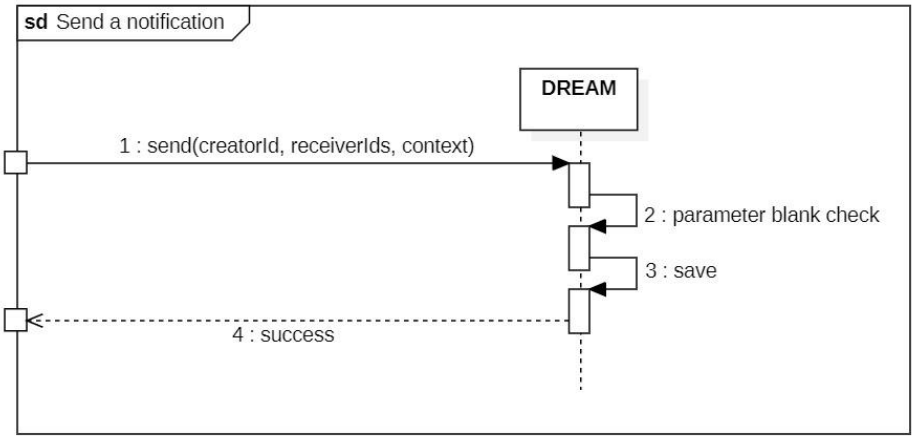
3.2.5 Sequence diagram

For all the sequence diagrams we aim to show the interactions between an actor and the system. For this reason, we are only considering the main “success” flow to keep them as readable as possible.

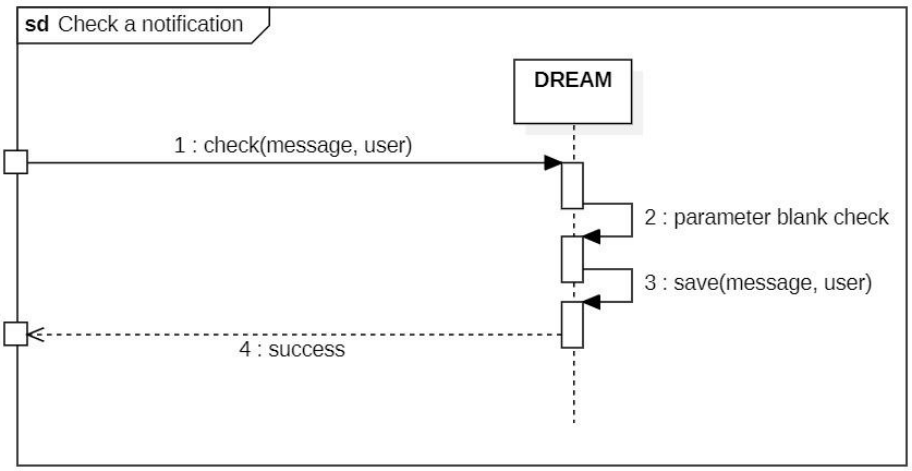
3.2.5.1 User registration



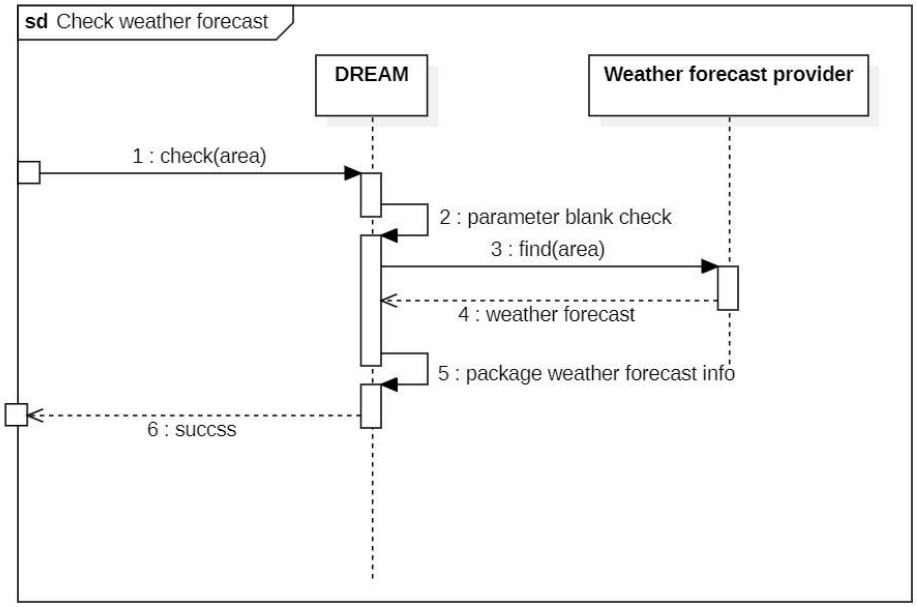
3.2.5.2 Send a notification



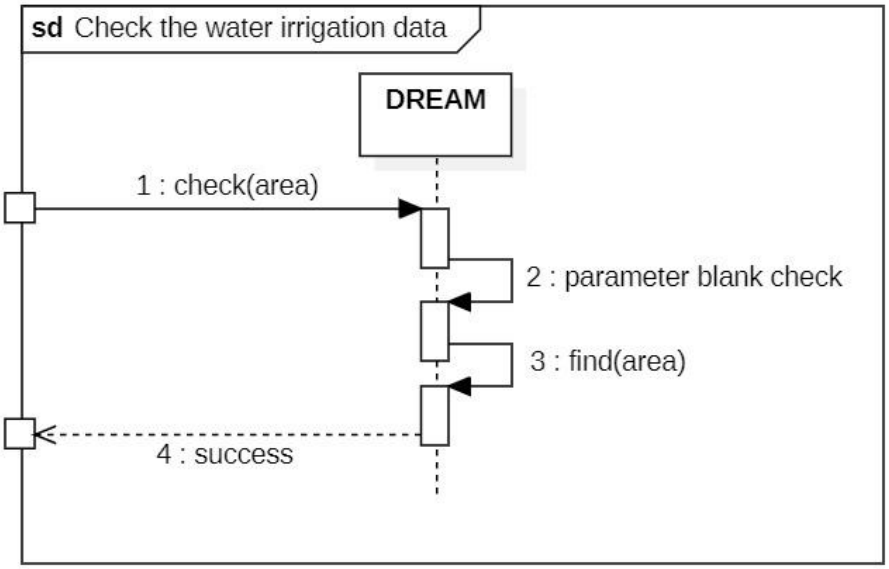
3.2.5.3 Check a notification



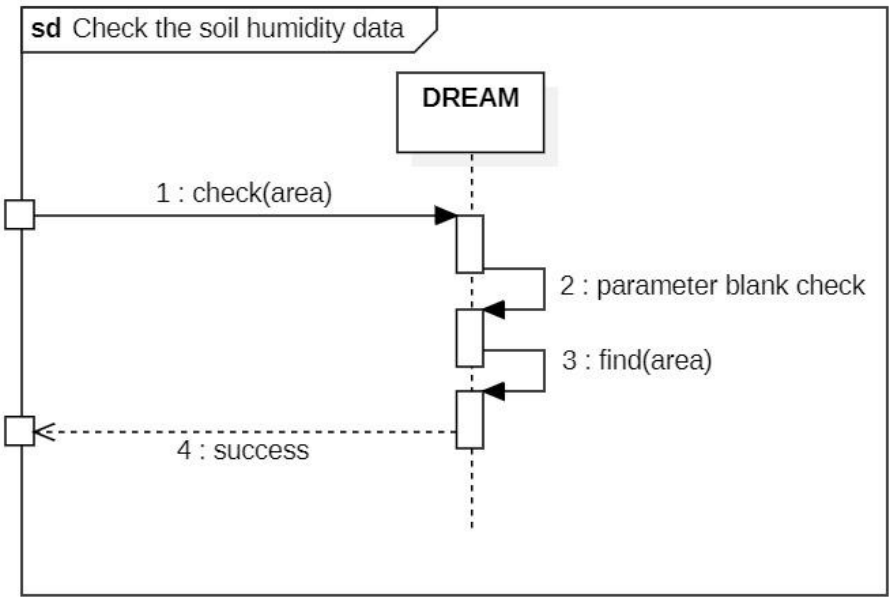
3.2.5.4 Check the weather forecast



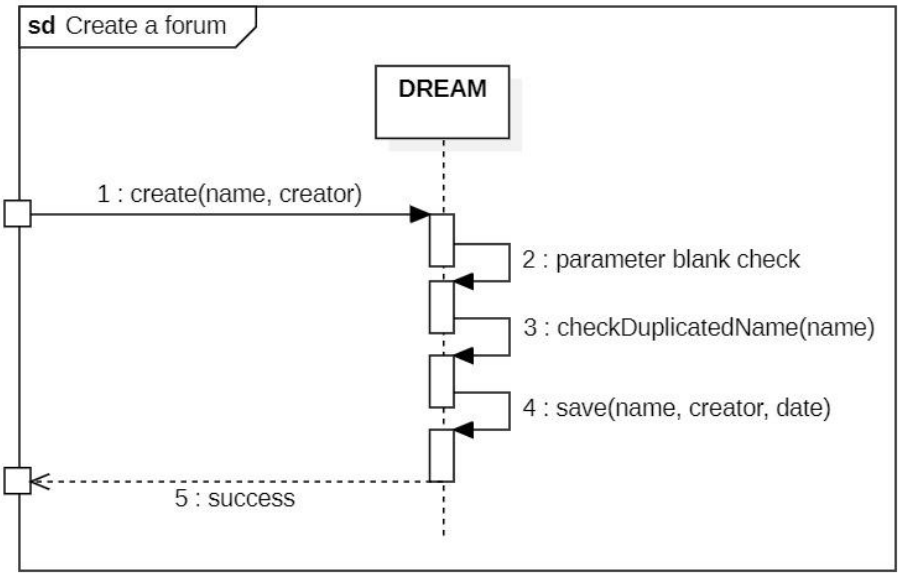
3.2.5.5 Check the water irrigation data



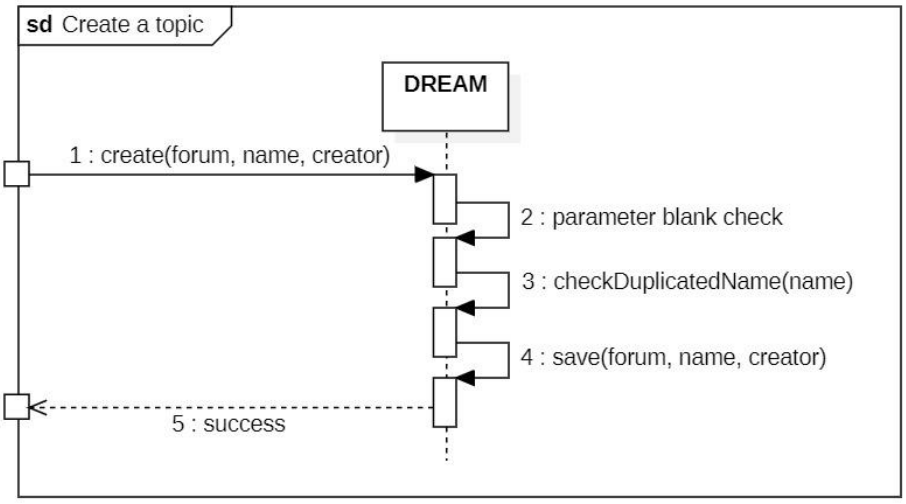
3.2.5.6 Check the soil humidity data



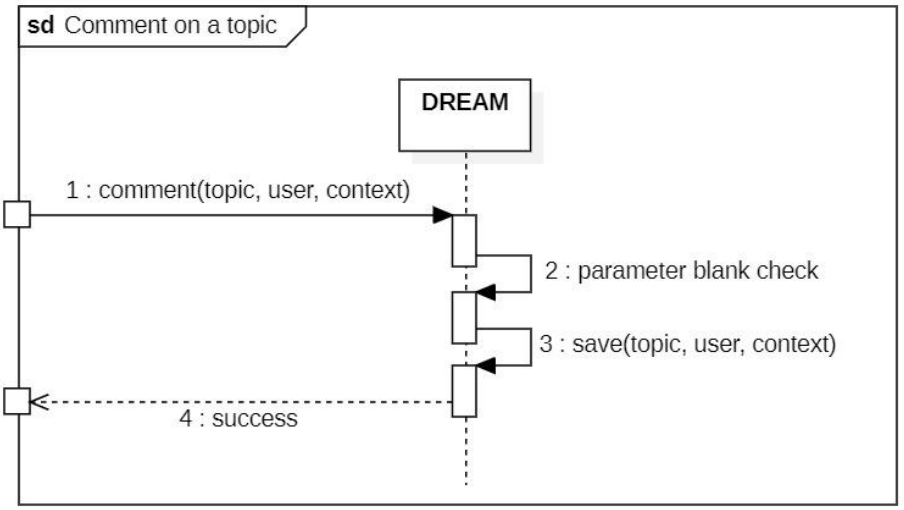
3.2.5.7 Create a forum



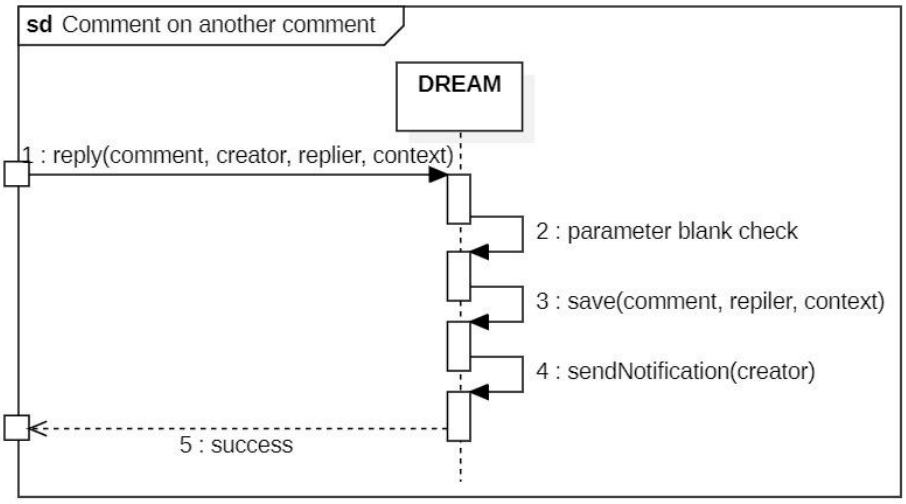
3.2.5.8 Create a topic



3.2.5.9 Comment on a topic

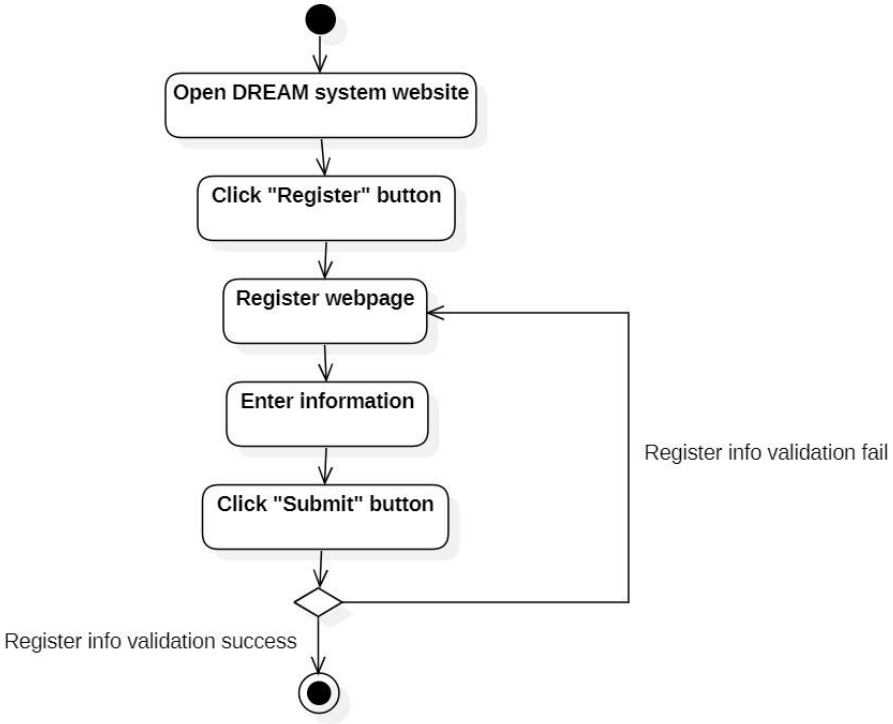


3.2.5.10 Comment on another comment

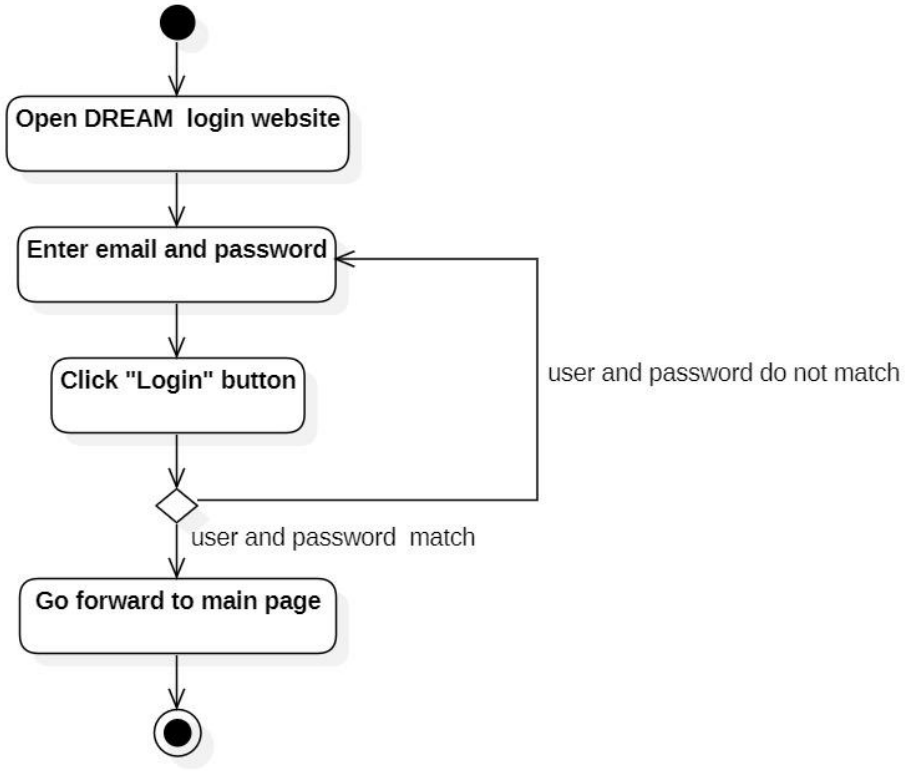


3.2.6 Activity diagrams

3.2.6.1 Unregistered users



3.2.6.2 Registered users



3.2.7 Mapping on requirements

Use case	Requirements
User registration	R1
Modify the location of the field	R3, R4
Save water irrigation data	R5
Save soil humidity data	R6
Check the statistic of performance	R7
Send a notification	R9, R10, R28
Check a notification	R11, R14, R23, R28
Check the weather forecast	R13
Check the water irrigation data	R19
Check the soil humidity data	R20

Insert the data about the production	R15
Search the data about the production	R18
Create a forum	R24
Create a topic	R21, R25
Comment on a topic	R26
Comment on another comment	R27

3.3 Performance requirements

- The system should be available 99% of the time
- In most cases, the system should respond to users in 3 seconds
- The system able to receive and process the data is sent by soil humidity sensors and water irrigation sensors

3.4 Design constraints

3.4.1 Standard compliance

- The users have access to the internet
- The sensors of water irrigation and soil humidity have access to the internet

3.4.2 Hardware limitations

- Soil humidity sensor
  - Wireless/wired internet access
  - Capable CPU and memory for sensor running
  - Stable power supply
  - Water-proof

- Water irrigation sensor
  - Wireless/wired internet access
  - Capable CPU and memory for sensor running
  - Stable power supply
  - Water-proof

- User device
  - Wireless/wired internet access
  - Capable CPU and memory for sensor running

- Server
  - Wire internet access
  - Capable CPU and memory for sensor running
  - Stable power supply

3.5 Software system attributes

3.5.1 Reliability

The system should be fault-tolerant, which means the system can handle the invalid or false type inputs. Moreover, when the system encounters the executing exceptions, the system can respond to the error messages and keep the system running.

3.5.2 Availability

The system user the micro-service technology to avoid the whole system breaking. When one of the services is down, it can not lead to the whole system breaking down, because every service is running separately.

3.5.3 Security

The user information that users provide and generate should be protected from leaking. The password of the user is encrypted, and check every parameter is SQL Inject or not.

3.5.4 Maintainability

The system mainly uses open-source technology for development, which bring high maintainability to the system, cause there are many mature docs and mature development community.

3.5.5 Portability

The develop language is used in this system development is Java that is platform-careless, and the architecture of the system is web client-server, so it can be used on most platforms (i.e., Mac, Windows, Linux) if and only the device users use have corresponded compiler for Java and internet access.

4 Formal analysis using Alloy

4.1 Code

```
open util/integer

//SIGNATURES
sig Area{
}

sig User{
}

sig Farmer extends User{
farmerFields: some Area
}

sig Policymaker extends User{
pmakerFields: some Area
}

sig Agronomist extends User{
agroFields: some Area
}

sig Forum{
forumCreatorUser: one Farmer
}

sig Topic{
topicCreatorUser: one Farmer,
forum: one Forum
}

sig Comment{
topic: one Topic,
commCreatorUser: one User
}


sig Notification{
creatorUser: one User,
```

```
receivers: some User
}

sig ProductionType{}

sig Production{
produtionType: one ProductionType,
productionArea: one Area
}

sig WaterIrrigation{
waterIrrigationArea: one Area
}

sig SoilHumidity{
soilHumidityArea: one Area
}

enum WeatherType{
SUNNY,
CLOUDY,
WINDY,
RAINY,
STORMY
}

sig WeatherForecast{
weatherForecastArea: one Area,
weatherType: one WeatherType
}

sig DailyPlan{
agronomist: one Agronomist,
visitedUsers: some Farmer
}

//FACTS

// There is no interset between the set of fields of each farmer
fact noInterFarmerFields{
all f1: Farmer, f2: Farmer | (f1 != f2) implies (f1.farmerFields not in f2.farmerFields)
}

// There is no interset between the set of fields of each policymaker
```

```
fact noInterPmakerFields{
all p1: Policymaker, p2: Policymaker | (p1 != p2) implies (p1.pmakerFields not in p2.pmakerFields)
}

// There is no interset between the set of fields of each agronomist
fact noInterAgroFields{
all a1: Agronomist, a2: Agronomist | (a1 != a2) implies (a1.agroFields not in a2.agroFields)
}

// // A forum is created by a farmer
fact userForumAssociation{
all af: Farmer | one f: Forum | f.forumCreatorUser in af
}

// A topic is associated with a forum
fact topicForumAssociation{
all f: Forum | one t: Topic | t.forum in f
}

// A topic is associated with a farmer
fact topicUserAssociation{
all f: Farmer | one t: Topic | t.topicCreatorUser in f
}

// A comment is associated with a topic
fact commentTopicAssociation{
all c: Comment | one t: Topic  | t in c.topic
}

// A comment is associated with a user
fact commentUserAssociation{
all u: User | one c: Comment | c.commCreatorUser in u
}

// A notificaton is associated with a  creator user
fact notificationCreatorAssociation{
all u: User | one n: Notification | n.creatorUser in u
}

// A notification is associated with at least one receiver user
fact notificationReceiverAssociation{
all u: User | one n: Notification | n.receivers in u
}
```

```
// The notification creator is not in the set of receiver of notification
fact creatorNotReceiver{
all n: Notification | n.creatorUser not in n.receivers
}

// A production is associated with an area
fact productionAreaAssociation{
all a: Area | one p: Production | p.productionArea in a
}

// A production is associated with a production type
fact productionTypeAssociation{
all pt: ProductionType | one p: Production | p.produtionType in pt
}

// A row of water irrigation data is associated with an area
fact waterIrrigationAreaAssociation{
all a: Area | one w: WaterIrrigation | w.waterIrrigationArea in a
}

// A row of soil humidity data is associated with an area
fact soilHumidityAreaAssociation{
all a: Area | one s: SoilHumidity | s.soilHumidityArea in a
}

// A weather forecast is associated with an area
fact weatherForecastAreaAssociation{
all a: Area | one w: WeatherForecast | w.weatherForecastArea in a
}

// A daily plan is associated with an agronomist
fact dailyPlanAgronomistAssociation{
all a: Agronomist | one d: DailyPlan | d.agronomist in a
}

// A daily plan is associated with at least one farmer
fact dailyPlanFarmerAssociation{
all f: Farmer | one d: DailyPlan | d.visitedUsers in f
}

pred show {
#Farmer > 0
#Policymaker >0
#Agronomist >0
```

#Area = 6
#Forum = 3
#Topic = 3
#Comment = 7
#Notification >= 0
#ProductionType > 0
#Production > 0
#WaterIrrigation > 0
#SoilHumidity > 0
#WeatherForecast > 0
#DailyPlan > 0
}

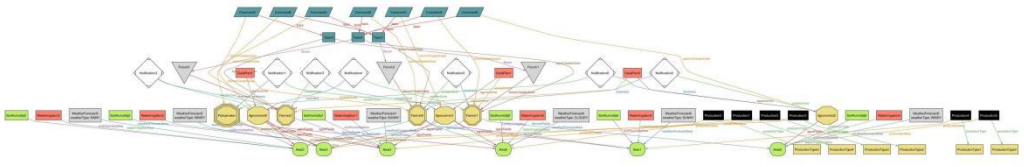
run show for 10

4.2 Result

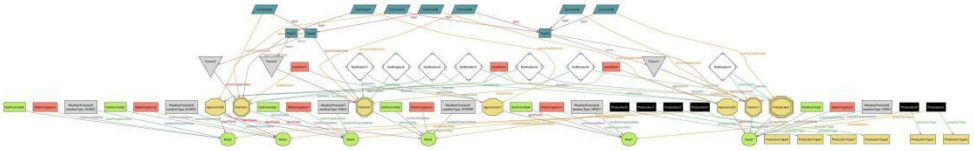
```
Executing "Run show for 10"
  Solver=sat4j Bitwidth=4 MaxSeq=7 SkolemDepth=1 Symmetry=20 Mode=batch
  27858 vars. 1900 primary vars. 65284 clauses. 132ms.
  Instance found. Predicate is consistent. 265ms.
```

4.3 Generated instances

- First run



- Second run



5 Effort spent

ZHENGLIANG MA(10824853)

Topic	Hours
General reasoning	10:00h
Purpose&Scope	1:30h
Class diagram	5:00h
Product functions	2:30h
Domain Assumptions	1:30h
Functional requirement	6:00h
Use cases & use cases diagrams	5:00h
Sequence diagrams	4:00h
Alloy	6:00h
Document organization	4:00h

BOREN JIANG(10825686)

Topic	Hours
General reasoning	10:00h
Purpose&Scope	2:30h
Class diagram	3:00h
Product functions	1:30h
Domain Assumptions	1:00h
Functional requirement	4:30h
Use cases & use cases diagrams	4:00h
Sequence diagrams	3:00h
Alloy	4:00h
Document organization	3:00h

6 References

All the diagrams have been made with StarUML
Alloy code was made on the specific tool Alloy