Documento a cura di Francesco Mazzola e Alessio Ferrara.   
Anno Accademico 2021-2022, Politecnico di Milano, MI

**RASD 2021-2022**

**1 INTRODUCTION**

**1.1 Purpose**

The goal of Telangana's government is to design, develop and demonstrate anticipatory governance models for food systems using digital public goods and community-centric approaches to strengthen data-driven policy making in the state. This will require the involvement of multiple stakeholders, from normal citizens to policy makers, farmers, market analysts, agronomists, etc.  
The purpose is to create an application that aims to make the farmers learn new techniques and improve their production through the help of other farmers and agronomists.  
  
This document contains an analysis of the system starting from a general description of the main goals, functionalities and scenarios and then going more into details describing the functional and non-functionals requirements that the system should fulfill, the use cases and their relative sequence diagrams that show the main interactions with the system.

**1.1.1 Goals**

|  |  |
| --- | --- |
| **Goals** | **Description** |
| **G1** | **Allow farmers to improve their productivity via personalized informations.** |
| **G2** | **Provide the Farmers the help and suggestions they may need during production.** |
| **G3** | **Allow policy maker to help farmers improve and let them decide if the carried-on initiatives are working.** |

**1.2 Scope**

DREAMS allows the farmer to insert their production data directly from their device. Moreover, they can view their personalized data based on their type of production and location and interact with other farmers in discussion forums.

The system also allows the Policy Maker to check the progress made by the farmers that are registered on DREAMS to help the farmers increase their production.

Furthermore, whether the Policy Maker evaluates a farmer as performing particularly badly he can pair said farmer with agronomist in order to carry a Steering Initiative aiming at improving the farmer’s performance.

**1.2.1 World Phenomena**

|  |  |
| --- | --- |
| **World Phenomena** | **Description** |
| **WP1** | **The user accesses the application web** |
| **WP2** | **The deployed sensors measure some data** |

**1.2.2 Shared Phenomena**

|  |  |  |
| --- | --- | --- |
| **Shared Phenomena** | **Description** | **Control** |
| **SP1** | **The user selects the option to log-in** | **WC** |
| **SP2** | **The user selects the option to log-out** | **WC** |
| **SP3** | **The farmer selects the option to insert production data** | **WC** |
| **SP4** | **The farmer selects the type of his production** | **WC** |
| **SP5** | **The farmer inserts the production data** | **WC** |
| **SP6** | **The farmer inserts the problem/s he faced during production** | **WC** |
| **SP7** | **The farmer selects the option to add a comment to a discussion** | **WC** |
| **SP8** | **The farmer selects the option to create a forum discussion** | **WC** |
| **SP9** | **The system shows the farmer all his personalized data** | **MC** |
| **SP10** | **The sensors collect the data that the system elaborates** | **WC** |
| **SP11** | **The policy maker selects the option to view farmer’s data** | **WC** |
| **SP12** | **The system shows the policy maker the data inherent to a farmer** | **MC** |
| **SP13** | **The policy maker selects the option to mark a farmer as a “Good farmer”** | **WC** |
| **SP14** | **The farmer receives a notification informing him of being identified as a “Good farmer”** | **MC** |
| **SP15** | **The policy maker selects the option to view all the farmers that are performing particularly badly** | **WC** |
| **SP16** | **The policy maker selects the option to view all ongoing steering initiatives** | **WC** |
| **SP17** | **The system shows the policy maker all the informations about the steering initiative production** | **MC** |
| **SP18** | **The system collects the data from the deployed sensors** | **WC** |

**1.3 Definitions and abbreviations**

**1.3.1 Definitions**

|  |  |
| --- | --- |
| **Definition** | **Description** |
| **Type of Production** | **The food that the farmer is producing** |
| **Good Farmer** | **A farmer that is performing particularly well** |
| **Steering initiative** | **A set of instructions that are provided by an Agronomist that is assigned to a farmer that is performing badly** |
| **Agronomist** | **A person that provides useful practices to those farmers that are having problems** |
| **Discussion Forum** | **A post in which a farmer asks for help or provides information on some problems to other farmers** |

**1.3.2 Abbreviations**

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| **WP** | **World Phenomena** |
| **SP** | **Shared Phenomena** |
| **WC** | **World Controlled** |
| **MC** | **Machine Controlled** |
| **G** | **Goals** |
| **D** | **Domain Assumptions** |
| **R** | **Requirements** |
| **RASD** | **Requirement Analysis and Specification Document** |
| **DD** | **Design Document** |

**1.4 Revision history**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Description** |
| **1.0** | **19/11/2020** | **The first version of the RASD** |
| **1.1** | **02/12/2020** | **List of changes:**   * **Added Alloy modulation and results.** * **Added sequence diagrams** |
| **1.2** | **18/12/2020** | **List of changes:**   * **Fixed some use cases diagram** * **Fixed some use cases** |
| **1.3** | **20/12/2020** | **Final review before delivery.** |

**2 OVERALL DESCRIPTION**

**2.1 PRODUCT PERSPECTIVE**

**2.1.1 Scenarios**  
  
*1 Marking a Farmer*  
John is a Telangana's Policy Maker and wants to have an overview on how a particular farmer is doing. After the log-in in the DREAMS Application he selects the option to have displayed a list of farmers with their production information. After some reasoning John, based on the data that the farmer and some sensors have provided, decides to mark a specific farmer as a “Good Farmer”.

*2 Analyse Results of Steering Initiatives*

Frank is a Telangana's Policy Maker that aims to analyse the production of a farmer after the tutoring of an agronomist. After the log-in in the DRAEMS Application Frank choses the option to have displayed a list of Steering Initiatives pursued on some Farmers.  
Frank selects a Steering Initiative and gets displayed the evolution of the production results during the timespan starting from two months before the help received by an Agronomist and ending three months later. Frank, once he has analysed the data can decide that the Steering Initiative carried by the Agronomist and the Farmer hasn’t produced significant results.

*3 Getting Personalized Informations*

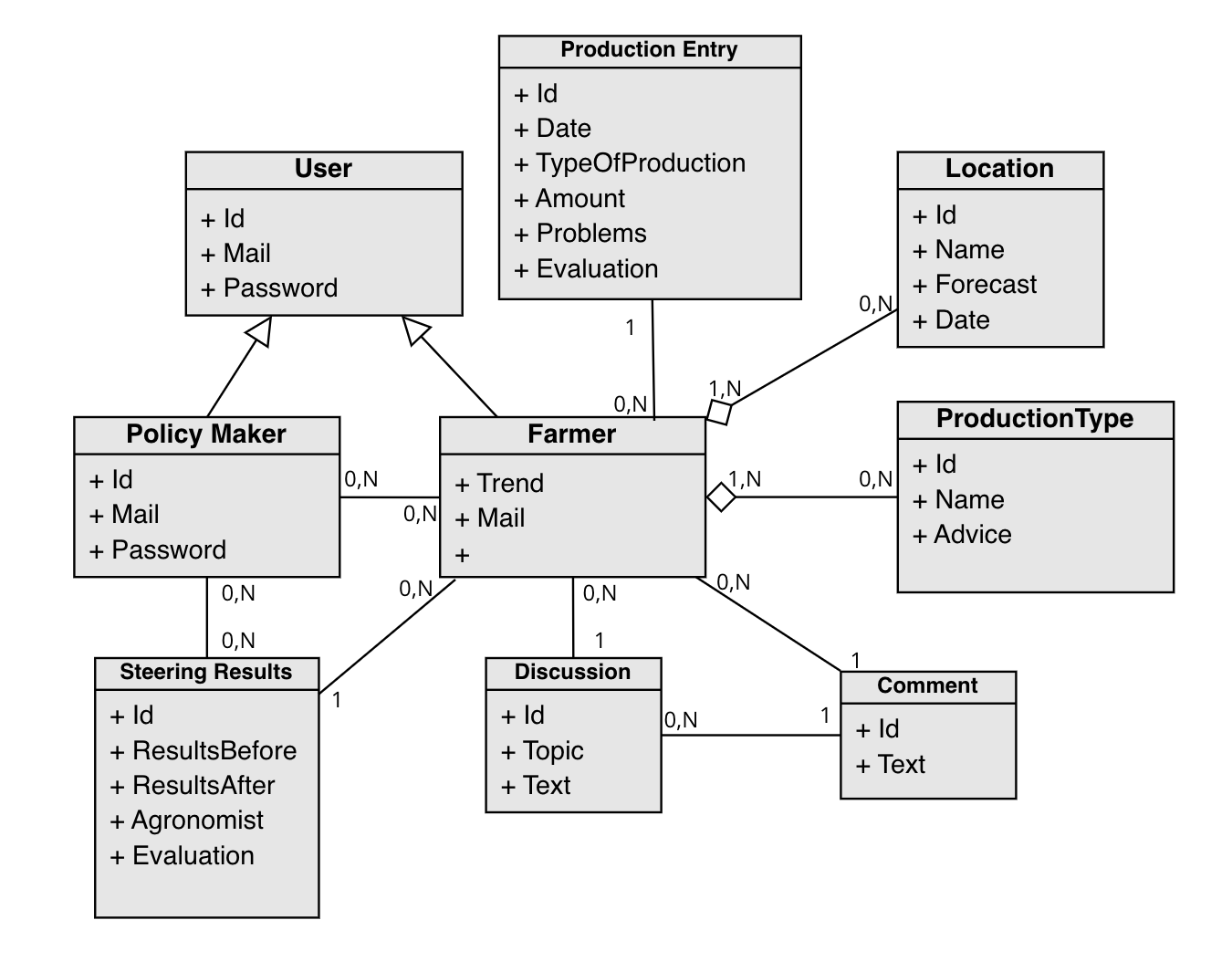
Alex is an Indian Farmer in the Telangana State and is registered into the DREAMS Application. Today Alex has decided to log into the Application to get an overview of the weather forecast and all the data that can be useful before farming.  
Alex after the log-in into the DREAMS selects the option to show the personalized suggestions and informations. The application shows him the data based on his location and type of production, so Alex writes it in a notebook to inform all his co-workers.

*4 Insert in the System*  
  
Rajesh is a Farmer whose farm is in the Telangana State. After the end of a workday, he decides to put all the information about the production into the DREAMS application. So, Rajesh logs into the application and selects the option to insert the production results, inserting that he has produced 50Kgs of rice and 40Kgs of bananas stating that he and his co-workers had some problems with the rice since some of it was unusable and went bad due to the excessive humidity of soil.

*5 Requesting for help and suggestions*  
  
Shanti, a farmer from the Telangana District working for the family’s farm has encountered some problems producing cotton. Since she’s registered into the DREAMS application she decides to log in and select the option to request for some help by creating a new forum.  
She writes her post asking for help on how to improve the production of cotton and giving some information about her farm to give more details on the problems she encountered in order to get more specific responses from other farmers or agronomists.

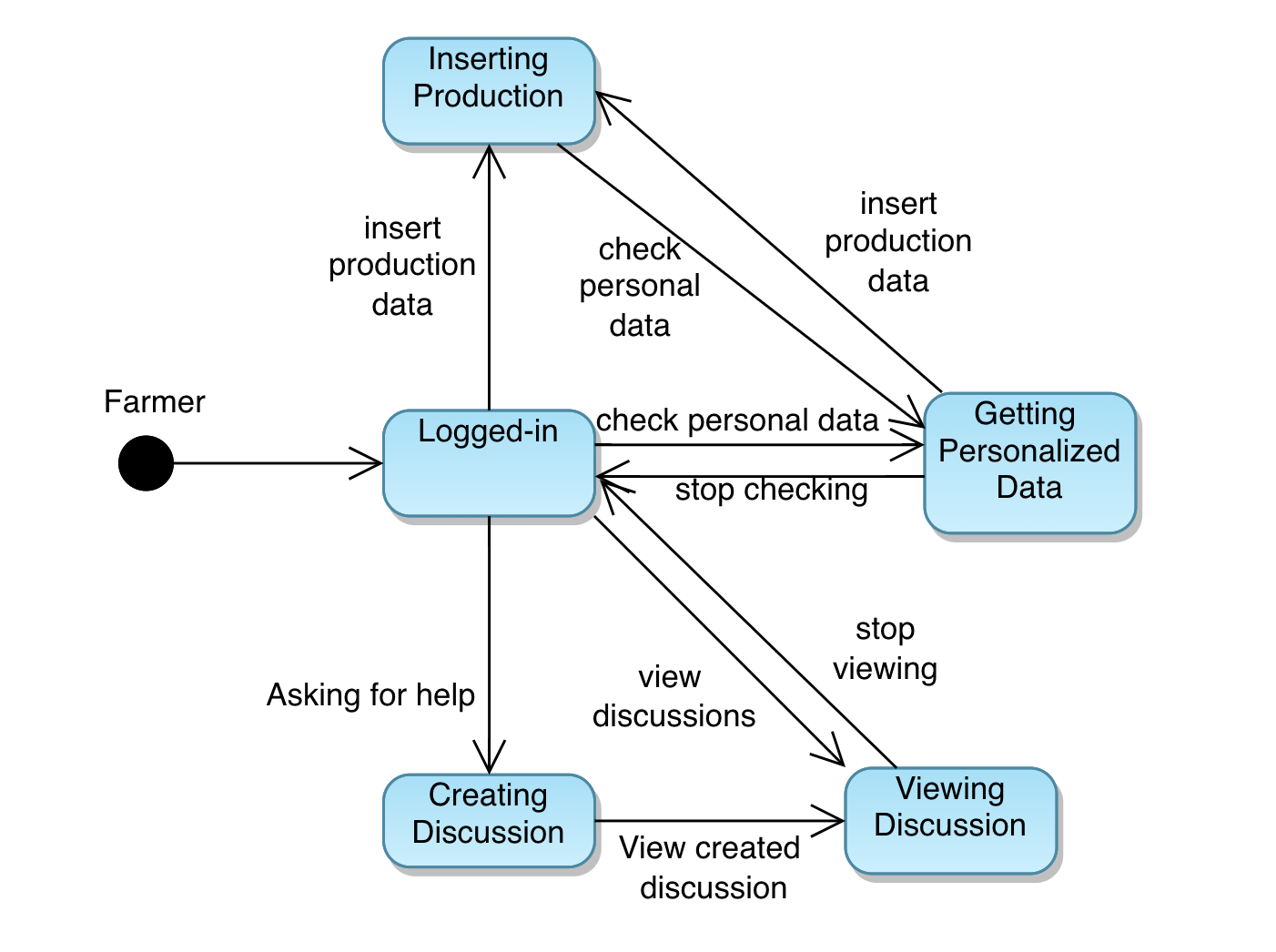
*6 Commenting in a discussion forum*  
  
Joel is a farmer from the Telangana state in India that is very good at his job.  
He’s currently at home after a productive workday and he decides to log-in into the DREAMS application to check if there are any help requests from other farmers.  
After finding a forum whose topic is inherent to Joel expertise, he opens he inserts a new comment answering the farmer who’s created the form.

**2.1.2 Class Diagram**

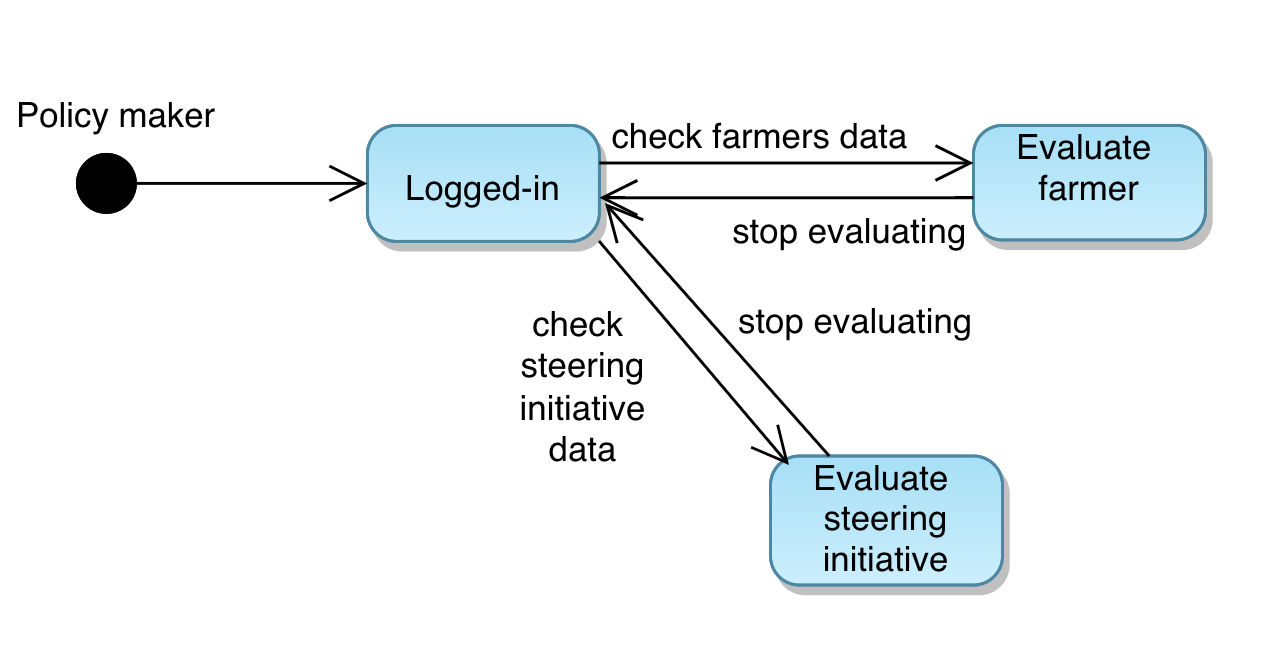


**2.1.3 Statechart Diagram**

**Farmer Statechart Diagram**



**Policy Maker Statechart Diagram**



**2.2 PRODUCT FUNCTIONS**

The system provides the users the following functionalities:

* **Collect production data:** The farmer can keep track of his productions inserting into the system all the informations concerning his productions, the type of productions, the amount and any problems he/she may have encountered.
* **Improve farmer production:** The farmer can request help by creating discussion forms asking for solutions to problems he may have encountered and getting some feedback from other farmers/ agronomists. In order to improve their production, the system also provides to the farmer a set of informations and suggestions based on his/her location and production type. Furthermore, a policy maker can pair a particularly bad performing farmer with an agronomist in order to carry a Steering Initiative aiming at improving the farmer’s production.

**2.3 USER CHARACTERISTICS**

The system involves the following actors:

* **Unregistered user**  
  A person that still need to login into the customer application to be able to   
  be identified.
* **Farmer**  
  A farmer that can insert information about their production and ask for help to other users.
* **Policy Maker**  
  A person that has been granted the ability to overview the Farmers data into the application and manage them.

**2.4 ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS**

**2.4.1 Domain Assumptions**

|  |  |
| --- | --- |
| **Domain Assumption** | **Description** |
| **D1** | **The location of the user corresponds to the location of his farm.** |
| **D2** | **The data of production declared by the farmer is correct.** |
| **D3** | **The information that the sensor provide is correct.** |
| **D4** | **The policy maker evaluates correctly the farmer.** |
| **D5** | **The farmer follows the steering initiatives properly.** |
| **D6** | **The reports about the steering initiatives results are correct.** |
| **D7** | **The farmer asks for help only when he needs it.** |

**3 Specific Requirements**

**3.1 External Interfaces Requirements**

**3.1.1 Software Interfaces**  
The System is deployed on an application that can be accessed via webpage, in order to retrieve all the data that the user needs.  
Some data may be obtained thanks to the use of some sensors that are deployed in the Telangana Region. The information that the above sensors provide are useful to our application but the hardware implementation of them is not of our concern.

**3.2 Functional Requirements**

**3.2.1 List of Requirements**

|  |  |
| --- | --- |
| **Requirements** |  |
| **R1** | **The system should allow the unregistered user to log-in.** |
| **R2** | **The system shall allow the farmer to view his personalized data.** |
| **R3** | **The system shall allow the farmer to edit his personal information.** |
| **R4** | **The system shall allow the farmer to edit his location.** |
| **R5** | **The system shall allow the farmer to edit his type of production.** |
| **R6** | **The system shall allow the farmer to insert data about his production and any problems he had.** |
| **R7** | **During the insertion of the production data the system shall allow the user to select the production type.** |
| **R8** | **The system shall allow the farmer to request for help and suggestions to other farmers.** |
| **R9** | **The system shall allow the farmer to create discussion forums.** |
| **R10** | **The system shall allow the farmer to view discussion opened by other farmers.** |
| **R11** | **The system shall allow the user to comment the opened discussion.** |
| **R12** | **The system should be able to create a report on the data before and after the Steering Initiative.** |
| **R13** | **The system should display to the Policy Maker all the farmers.** |
| **R14** | **The system should display to the Policy Maker the informations about a specific farmer.** |
| **R15** | **The system should let the Policy Maker decide whether the Farmer is performing well or not.** |
| **R16** | **The system shall allow the Policy Maker to see the farmers that are performing particularly badly.** |
| **R17** | **The system shall display the Policy Maker all the Steering Initiatives.** |
| **R18** | **The system shall display the data concerning a specific Steering Initiative.** |
| **R19** | **The system shall allow the Farmer to delete a discussion.** |
| **R20** | **The system shall allow the Farmer to edit a discussion.** |
| **R21** | **The system shall allow the Farmer to delete a comment.** |

**3.2.2 Mapping of Goals and Domain Assumptions**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **G1** | **G2** | **G3** |
| **D1** |  |  |  |
| **D2** |  |  |  |
| **D3** |  |  |  |
| **D4** |  |  |  |
| **D5** |  |  |  |
| **D6** |  |  |  |
| **D7** |  |  |  |

**3.2.3 Mapping of Goals and Domain Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **G1** | **G2** | **G3** |
| **R1** |  |  |  |
| **R2** |  |  |  |
| **R3** |  |  |  |
| **R4** |  |  |  |
| **R5** |  |  |  |
| **R6** |  |  |  |
| **R7** |  |  |  |
| **R8** |  |  |  |
| **R9** |  |  |  |
| **R10** |  |  |  |
| **R11** |  |  |  |
| **R12** |  |  |  |
| **R13** |  |  |  |
| **R14** |  |  |  |
| **R15** |  |  |  |
| **R16** |  |  |  |
| **R17** |  |  |  |
| **R18** |  |  |  |
| **R19** |  |  |  |
| **R20** |  |  |  |
| **R21** |  |  |  |

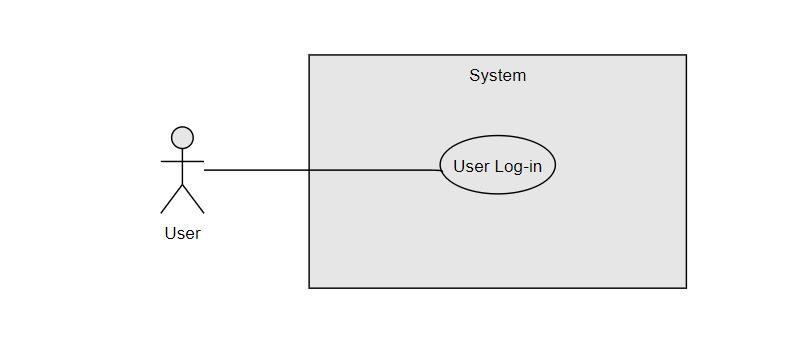
-To archive G1 we need to let the farmer log-in and let him modify his informations in order to update the personalized data according to his most updated informations such as location or type of production.

-To archive G2 we need to let the farmer log-in and let him add his productions and let him point out any problems he had. The farmer also needs to interact with the other farmers in order to learn new techniques, hence he needs to be able to view, create and comment a discussion.

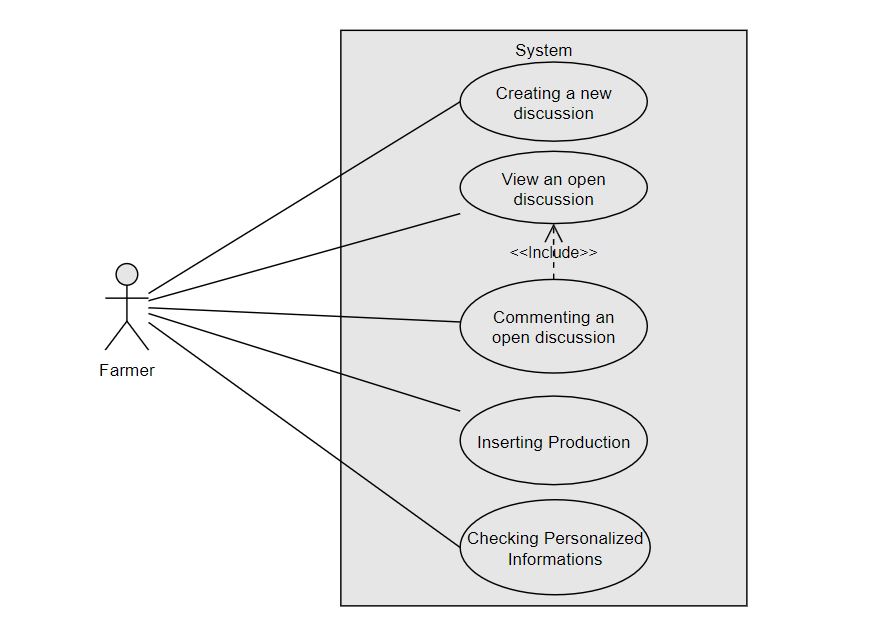
-To archive G3 we need to let the policy maker log-in and evaluate all the informations about the farmers in order to help them improve their production or let them inform other farmers on the best techniques that they are using if performing well.

**3.2.4 Use case Diagrams**

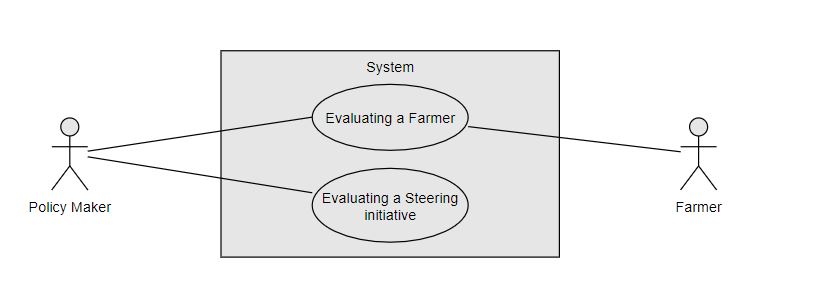
* **User**



* **Farmer**



* **Policy Maker**



The farmer has been added to the Policy Maker Use Case Diagram in order to emphasize his passivness in the “Evaluating Farmer” Use case.

**3.2.5 Use cases**

|  |  |
| --- | --- |
| **Name** | **User log-in** |
| **Actors** | **User** |
| **Entry Conditions** | **The user, from one of the system pages, has selected the option to log-in in our system** |
| **Flow of events** | 1. **The system shows the user the log-in page.** 2. **The user inserts his credentials.** 3. **The user selects the confirm option.** 4. **The system checks the credentials (username and password) submitted by the user.** |
| **Exit Conditions** | **The user is successfully logged in either as a farmer or a policy maker. The system displays the user the last page visited before the log-in process.** |
| **Exceptions** | **5. If the system doesn’t find any matching users, it displays an error message and it will go back to point 1** |

|  |  |
| --- | --- |
| **Name** | **Inserting Production** |
| **Actors** | **Farmer** |
| **Entry Conditions** | **The farmer selects the option to add a new production** |
| **Flow of events** | 1. **The system shows the farmer the form to fill.** 2. **The farmer inserts the amount information.** 3. **The system shows the farmer one of the available production types that the farmer has declared.** 4. **The farmer selects one of the available production types.**   **4.1. The farmer can insert any problems he faced during production.**   1. **The Farmer selects the option to submit all the informations.** 2. **The system shows the farmer the Personal Data page.** |
| **Exit Conditions** | **The production has been inserted successfully in the system.** |

|  |  |
| --- | --- |
| **Name** | **Checking personalized information** |
| **Actors** | **Farmer** |
| **Entry Conditions** | **The farmer selects the option to check personalized informations** |
| **Flow of events** | 1. **The system shows the farmer his personalized informations based on his location and production type such as the: weather forecast,**  **suggestions concerning specific crops to plant or specific fertilizers to use.**   **1.1. The farmer can select a different time period in order to view the weather forecast for that timespan.** |
| **Exit Conditions** | **The farmer can correctly visualize all the personalized informations and suggestions.** |

|  |  |
| --- | --- |
| **Name** | **Creating a new discussion** |
| **Actors** | **Farmer** |
| **Entry Conditions** | **The farmer selects the option to create a new discussion.** |
| **Flow of events** | 1. **The system shows the farmer the form to create a new discussion.** 2. **The farmer fills the form with all the needed informations to create the discussion.** 3. **The farmer selects the option to create discussion.** 4. **The system asks the farmer to confirm that the inserted data is correct.** 5. **The farmer selects the confirm option.** 6. **The system shows the farmer the discussion created.** |
| **Exit Conditions** | **The discussion has been correctly created and inserted in the system.** |

|  |  |
| --- | --- |
| **Name** | **View an open discussion** |
| **Actors** | **Farmer** |
| **Entry Conditions** | **The farmer selects the option to view opened discussions.** |
| **Flow of events** | 1. **The system shows the farmer a list of all opened discussions by other farmers.** 2. **The farmer selects a discussion that he is interested into.** 3. **The system shows the farmer the discussion with all the comments from the other users.** |
| **Exit Conditions** | **The farmer can successfully view the desired discussion and interact with other users.** |

|  |  |
| --- | --- |
| **Name** | **Commenting an open discussion** |
| **Actors** | **Farmer** |
| **Entry Conditions** | **The farmer is viewing an open discussion and selects the option to comment** |
| **Flow of events** | 1. **The system shows the farmer the form to fill in order to create the comment.** 2. **The farmer fills the form with the desired comment.** 3. **The farmer selects the option to post the comment.** 4. **The system shows the user the updated discussion with the newly created comment.** |
| **Exit Conditions** | **The comment is correctly posted in the system and it is shown in the discussion.** |

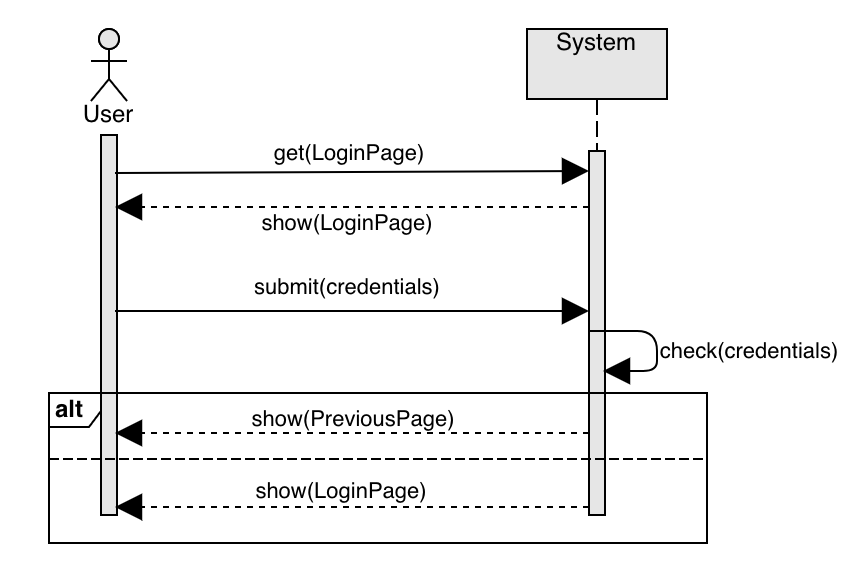
|  |  |
| --- | --- |
| **Name** | **Evaluating a farmer** |
| **Actors** | **Policy Maker, Farmer** |
| **Entry Conditions** | **The Policy Maker selects the option to view all farmers** |
| **Flow of events** | 1. **The system shows the policy maker the list of all farmers.** 2. **The policy maker selects the farmer that he wants to evaluate.** 3. **The system shows the policy maker the selected farmer’s data.** 4. **The policy maker can make an evaluation on the farmer’s productivity.** 5. **The policy maker evaluates the farmer as positive, he selects the option to mark him as a “Good Farmer” or “Bad Farmer”.** 6. **The system asks the policy maker to confirm his choice.** 7. **The policy maker selects the option to confirm.** 8. **The system notifies the farmer that he/her is now been evaluated by a policy maker as a “Good Farmer” or “Bad Farmer”.** 9. **The system shows the policy maker the list of all farmers as described in point 1.** |
| **Exit Conditions** | **The evaluation has been correctly inserted in the system.**  **Either the farmer has been evaluated correctly as a “Good Farmer” and will see a notification after the evaluation or the farmer has been evaluated correctly as a “Bad Farmer” and it’s been inserted in the list of bad farmers and will also see a notification after the evaluation.** |
| **Exceptions** | **If the policy maker doesn’t evaluate the farmer as a “Good Farmer” or “Bad Farmer” he can go back to the list of all farmers as described in point 1.** |

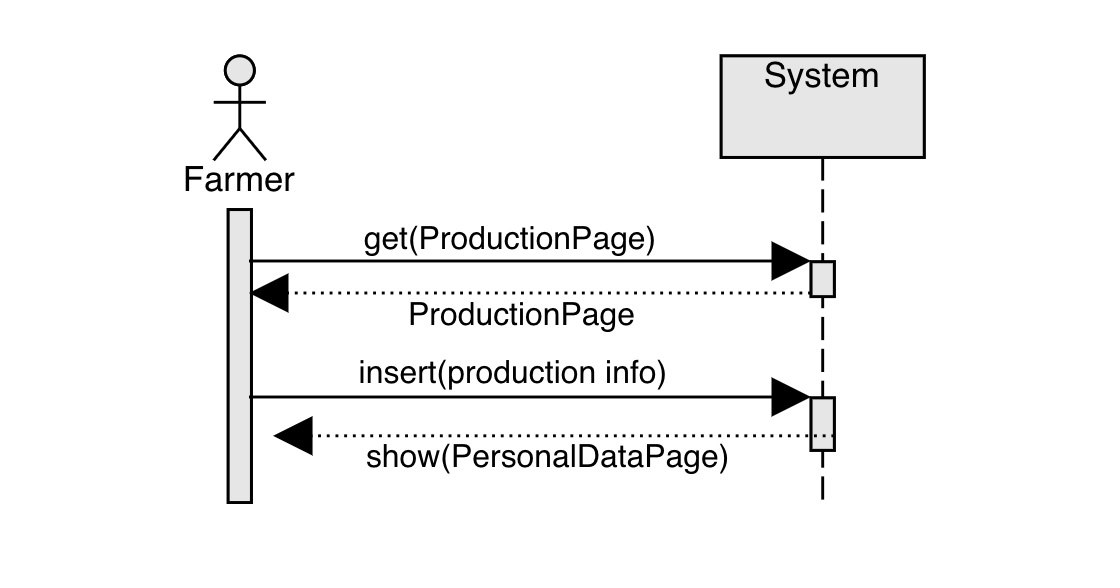
|  |  |
| --- | --- |
| **Name** | **Evaluating a steering initiative** |
| **Actors** | **Policy Maker** |
| **Entry Conditions** | **The Policy Maker selects the option to view all the concluded steering initiatives** |
| **Flow of events** | 1. **The system shows the policy maker the list of all steering initiatives that have ended.** 2. **The policy maker selects one steering initiative to evaluate in the list.** 3. **The system shows the policy maker the data useful to evaluate the steering initiative and all the informations about the steering initiative itself.** 4. **The policy maker evaluates the steering initiative as positive.** 5. **The policy maker selects the option to mark the steering initiative as positive** 6. **The system asks the policy maker to confirm his choice.** 7. **The policy maker selects the option to confirm.** 8. **The system shows the policy maker the list of all steering initiatives as described in point 1.** |
| **Exit Conditions** | **The evaluation has been correctly inserted in the system.**  **The policy maker has been evaluated correctly the steering initiative.** |
| **Exceptions** | **If the policy maker doesn’t evaluate the steering imitative as positive, he can go back to the list of all steering initiatives as described in point 1.** |

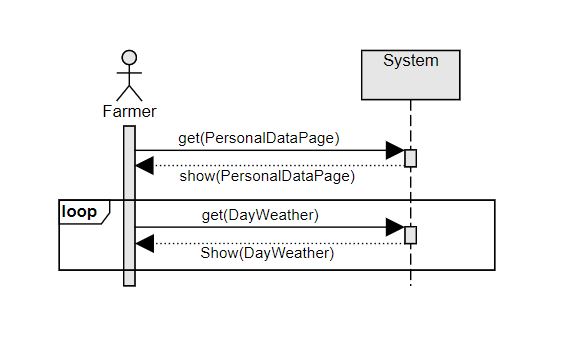
**3.2.6 Sequence Diagrams**

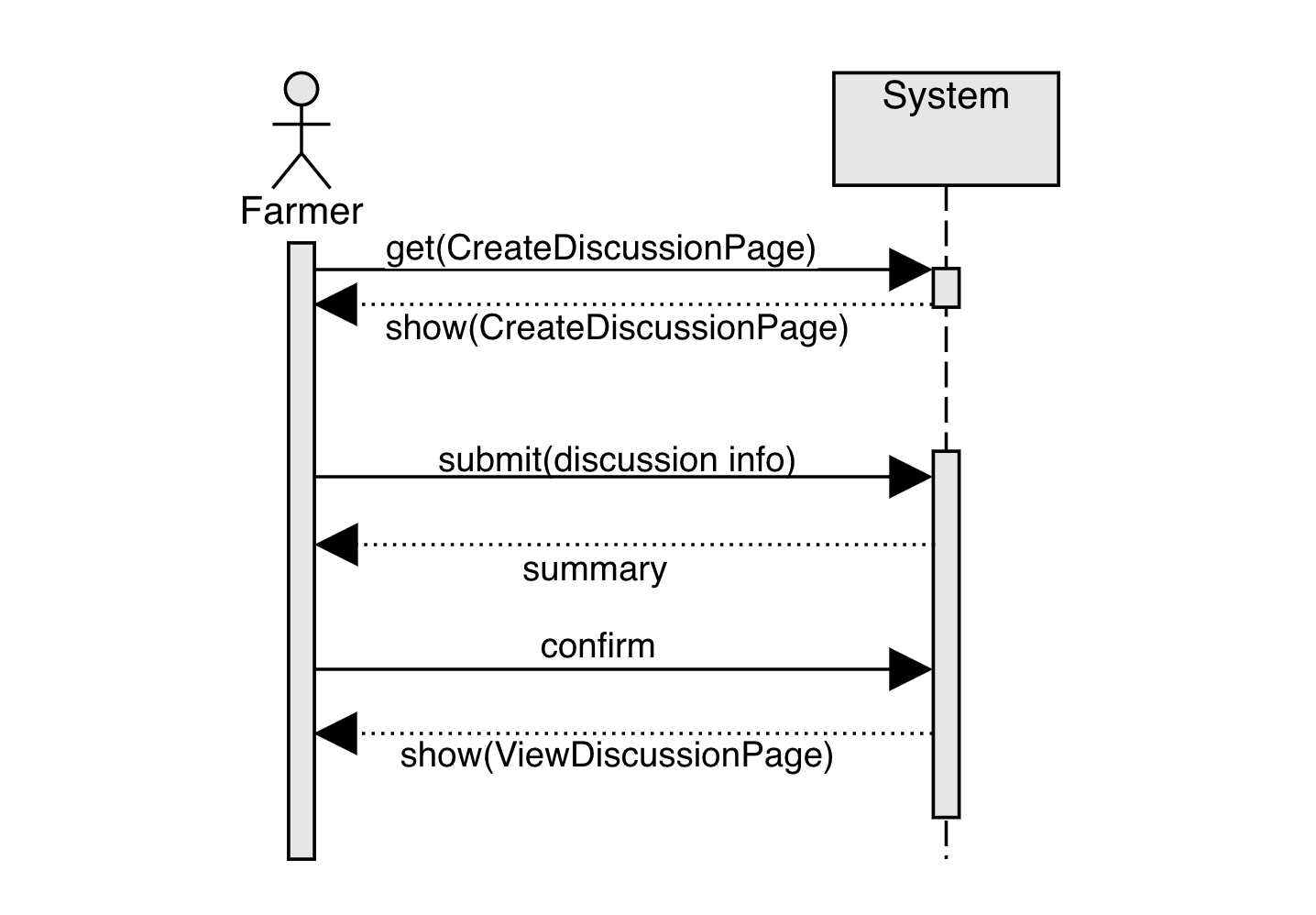
For all the sequence diagrams our aim describes how—and in what order—a group of objects works together, especially the interactions between actor and system. For this reason, we are only considering the main “success” flow in order to keep them as readable as possible

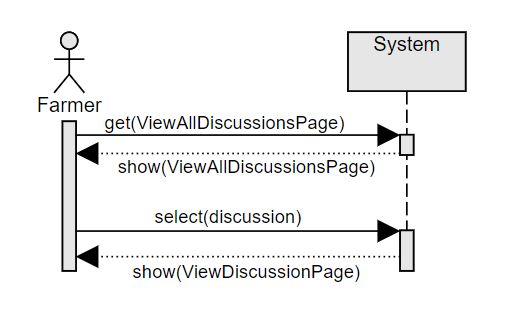
* **User Log-in**



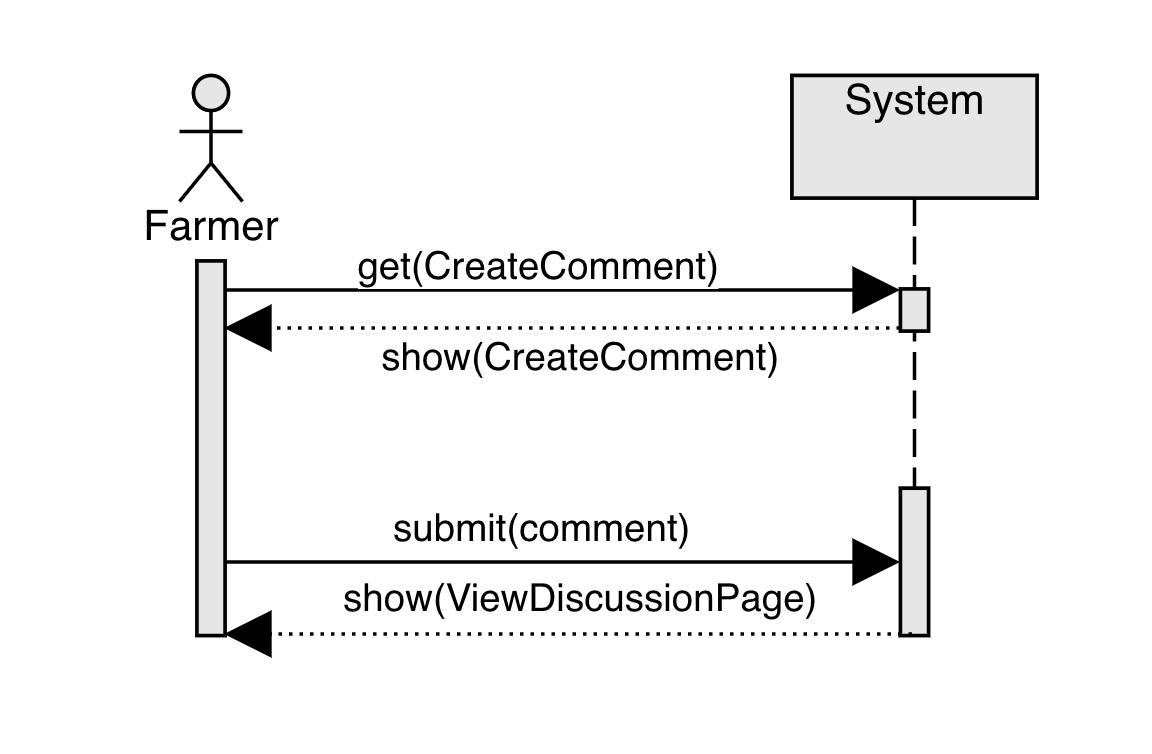
* **Inserting Production**
* **Checking personalized information**



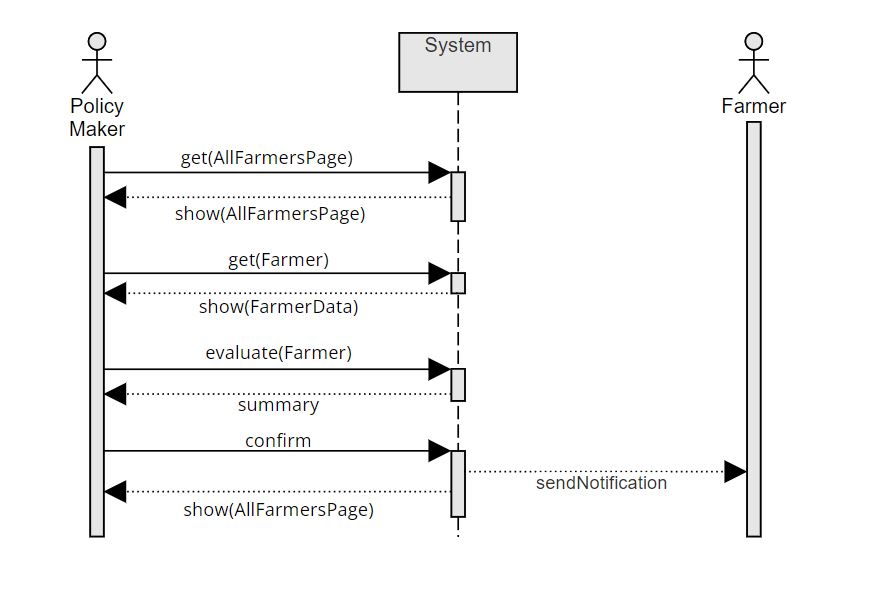
* **Creating a new discussion**
* **View an open discussion**



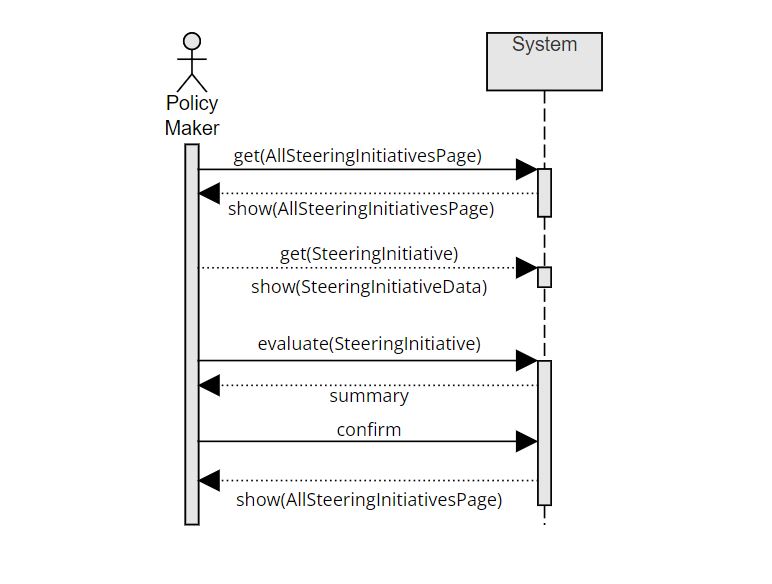
* **Commenting an open discussion**



* **Evaluating a farmer**



* **Evaluating a Steering initiative**



**3.3 PERFORMANCE REQUIREMENTS**

* The system should be available 99% of the time.

**3.4 DESING CONSTRAINTS**

**3.4.1 Standards Compliance**

* The system must ask the user permission to let the Policy Makers retrieve their data.
* The system must manage the data retrieved from the users in respect with the privacy laws.

**3.4.2 Hardware Limitations**

1. **Customer Device**  
   **-** Network connection
2. **Sensors**  
   **-** Network connection
3. **Server**  
   - Cabled network connection  
   - Sufficient processing power to handle the multiple requests in parallel

**3.5 SOFTWARE SYSTEM ATTRIBUTES**

**3.5.1 Reliability**

The system shall never crash or hang, other than as the result of an operating system error. The system shall provide graceful degradation in the face of network delays and failures, or when handling large amounts of data.

**3.5.2 Availability**

Possible maintenance procedures in the system should be performed in one of the duplicate components at the time, in order to ensure continuity to the service.

**3.5.3 Security**

The user information provided by the users should be treated as sensible and thus protected by possible attacks on the database. The transmission of data should be secured by encryption and decryption techniques.

**3.5.4 Maintainability**

All code shall be fully documented. Each function shall be commented with pre- and post-conditions. All program files shall include comments concerning authorship and date of last change. The code shall be modular to permit future modifications

**3.5.5 Portability**

The software shall be designed to run on at least one of the following four platforms:

• Microsoft Windows (Windows implementations shall be portable to all versions of Windows up to and including Windows XP)

• UNIX (Unix implementation shall be portable to any version of Unix that supports the user interface libraries used)

• Apple Macintosh OS (Apple Mac implementations shall be portable to all current versions of the MacOS)

• Linux (Linux implementations shall run on at least one version of Linux) No other specific portability requirements have been identified

Using one of the available browsers (Chrome, Edge, Safari etc...) on the above platforms.

**4 FORMAL ANALYSIS USING ALLOY**

**4.1 Code**

The main goal of our Alloy analysis is to validate the model, imposing some constraints that aren’t shown in the class diagram.  
Those constraints are:

* The farmer can only insert a production coherent with his production type

And are expressed by the following Alloy code:

open util/integer

//----SIGNATURES----

abstract sig User { }

sig Farmer extends User {

location: one Location,

typeP: set ProductionTypes ,

production: set Production,

discussion: set Discussion,

comment: set Comment

}

sig Policymaker extends User {

badgeNumber: one Int

}{badgeNumber >= 0}

sig Location {

weather: set Weather

}

sig Weather {

month: one Int,

day: one Int ,

hour: one Int

}{day > 0 and month > 0 and hour > 0}

sig Discussion {

commentDiscussion : set Comment}

sig Comment {}

//The most common productions in Telangana according to Professor Jayashankar, Professor at the Telangana state agricultural university

enum ProductionTypes {

Rice,

Corn,

Jowar,

Cotton,

Castor,

Groundnut,

Soya,

RedGram,

GreenGram,

BlackGram,

Sesame

}

sig Production{

kilograms: one Int,

type: one ProductionTypes,

} {kilograms > 0}

//FACTS

//A production is always associated with a farmer

fact farmerProduction {

all p: Production | one f: Farmer | p in f.production

}

//A production have a type in the farmer typeP

fact farmerTypeProd{

all p: Production| one f: Farmer| p in f.production &&

p.type in f.typeP

}

//A weather is always associated with a location

fact weatherLocation{

all w: Weather | one l: Location | w in l.weather

}

//Polimakers should have different bedge number

fact differenteBedge {

all p1: Policymaker , p2: Policymaker | p1.badgeNumber = p2.badgeNumber implies p1 = p2

}

//Every comment is always associated witha one farmer

fact commentFarmer {

all c: Comment | one f: Farmer | c in f.comment

}

//Every comment is always associated with one discussion

fact commentDiscussion {

all c: Comment | one d: Discussion | c in d.commentDiscussion

}

//Every discussion is always associated with one farmer

fact discussionFarmer {

all d: Discussion | one f: Farmer | d in f.discussion

}

pred show{

#Farmer > 1

#Policymaker > 1

#Production = 3

#Location= 2

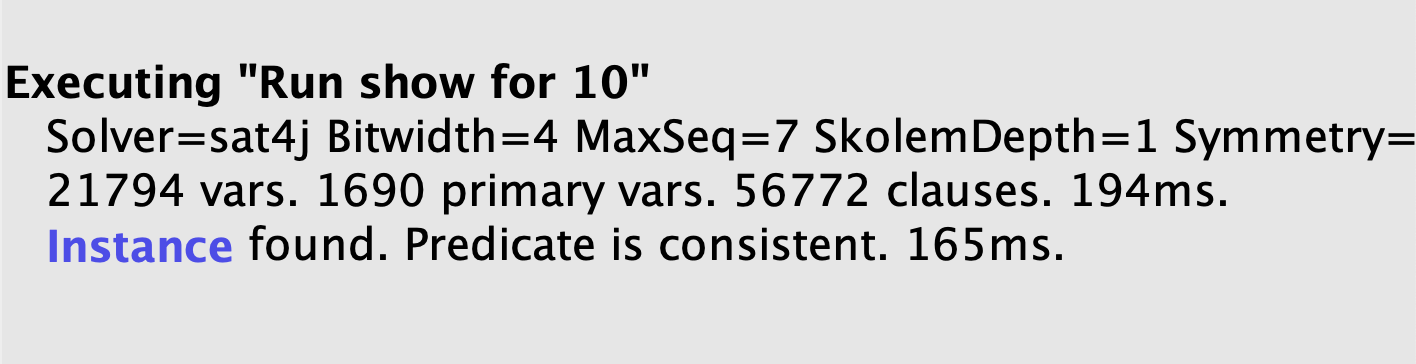
#Weather > 2

#Comment = 4

#Discussion= 4}

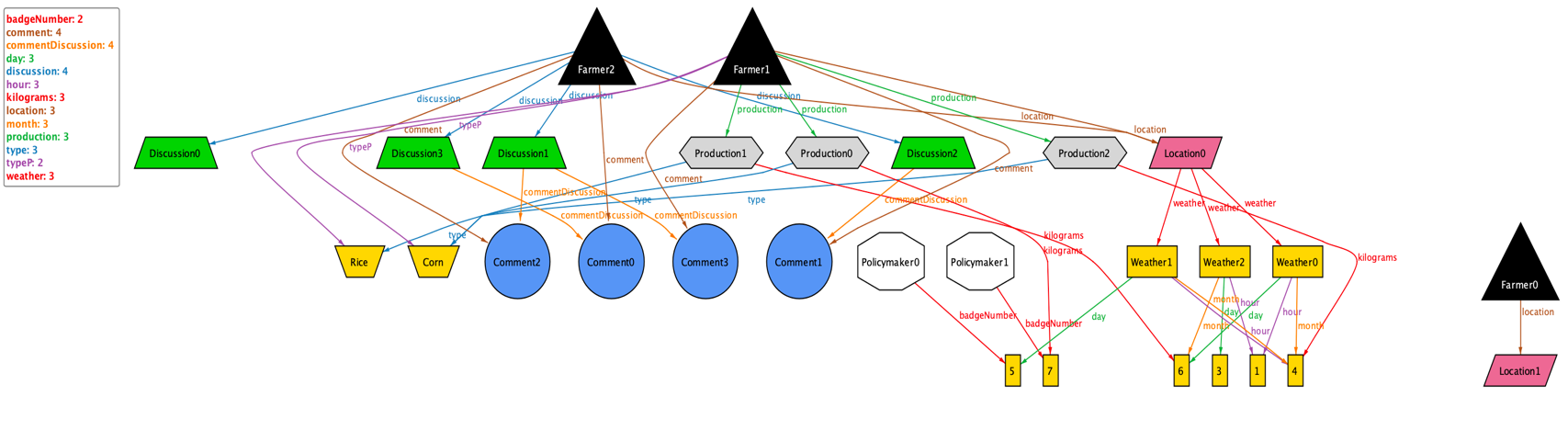
run show for 10

**4.2 Result**

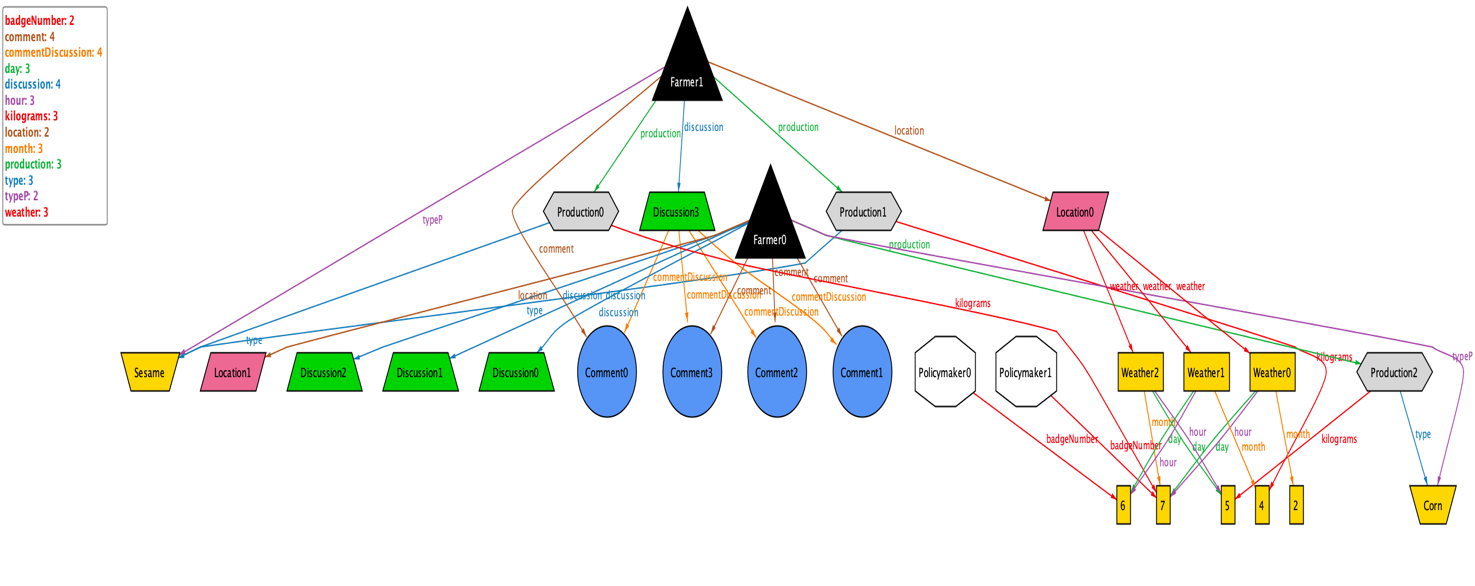


**4.3 Generated instances**

* **First run**



* **Second run**



**5 EFFORT SPENT (anonymously)**

**Student 1**

|  |  |
| --- | --- |
| **Topic** | **Hours** |
| **General Reasoning** | **5 h** |
| **Purpose & Scope** | **2 h** |
| **Class Diagram** | **1 h** |
| **Statechart** | **1 h** |
| **Product Functions** | **1 h** |
| **Domain Assumptions** | **2 h** |
| **Functional Requirements** | **4 h** |
| **Use Cases and Use Cases Diagrams** | **5 h** |
| **Sequence Diagrams** | **6 h** |
| **Alloy** | **4 h** |
| **Document Review** | **5 h** |

**Student 2**

|  |  |
| --- | --- |
| **Topic** | **Hours** |
| **General Reasoning** | **5 h** |
| **Purpose & Scope** | **2 h** |
| **Class Diagram** | **1 h** |
| **Statechart** | **1 h** |
| **Product Functions** | **1 h** |
| **Domain Assumptions** | **2 h** |
| **Functional Requirements** | **3 h** |
| **Use Cases and Use Cases Diagrams** | **5 h** |
| **Sequence Diagrams** | **4 h** |
| **Alloy** | **6 h** |
| **Document Review** | **5 h** |