POLITECNICO
MILANO 1863

DREAM - Data-dRiven PrEdictive FArMing in Telangana

Careddu Gianmario
La Greca Michele Carlo
Zoccheddu Sara

Prof. Elisabetta Di Nitto

Requirement Analysis and Specification Document

Version 1.2

5th February 2022

Deliverable: RASD

Title: Requirement Analysis and Specification Document

Authors: Careddu Gianmario
La Greca Michele Carlo
Zoccheddu Sara

Version: 1.2

Date: 5th February 2022

Download page: https://github.com/gccianmario/Zoccheddu-LaGreca-Careddu

---

TABLE OF CONTENTS

1. INTRODUCTION **5**
1.1. Purpose 5
1.2. Goals 5
1.3. Definitions, Acronyms, Abbreviations 5
1.3.1. Definitions 5
1.3.2. Acronyms 7
1.3.3. Abbreviations 7
1.4. Scope 8
1.4.1. World Phenomena 8
1.4.2. Shared Phenomena 8
1.5. Revision history 12
1.6. Document Structure 12

2. OVERALL DESCRIPTION **14**
2.1. Product perspective 14
2.1.1. Class diagrams 14
2.1.2. Statechart 16
2.2. User characteristics 19
2.3. Product functions 20
2.4. Assumptions, dependencies, constraints 24
2.4.1. Domain assumptions 24

3. SPECIFIC REQUIREMENTS **26**
3.1. External Interface Requirements 26
3.1.1. Software Interfaces 26
3.2. Functional Requirements 26
3.2.1. Farmer Scenarios 26
3.2.1.1. Use case diagrams 28
3.2.1.2. Use case tables and activity diagrams 30
3.2.2 Agronomist Scenarios 46
3.2.2.1. Use case diagrams 47
3.2.2.2. Use case tables and activity diagrams 49
3.2.3 Policy Maker Scenarios 65
3.2.3.1. Use cases diagrams 66
3.2.3.2. Use cases tables and activity diagrams 66
3.2.4 Government admin Scenarios 75
3.2.4.1. Use cases diagrams 75
3.2.4.2. Use cases tables and activity diagrams 75
3.2.5. Requirements 79
3.3. Performance Requirements 86
3.4. Design Constraints 86
3.4.1. Standards compliance 86

3.5. Software System Attributes 86
3.5.1. Reliability 86
3.5.2. Availability 86
3.5.3. Security 86
3.5.4. Maintainability 87
3.5.5. Portability 87
3.5.6. Usability 87

4. FORMAL ANALYSIS USING ALLOY 88
4.1. Visit part 88
4.2. Forum part 90

5. EFFORT SPENT 93

6. REFERENCES 94

1. INTRODUCTION

1.1. Purpose

DREAM (Data-dRiven prEdictive fArMing) is an elaboration of an ongoing initiative promoted by Telangana's government which is the 11th largest state in India, with about 35 million residents living in 33 districts.
Telangana's economy, as the Indian economy, is mainly based on agriculture for the majority conducted by smallholder farmers with less than 2 hectares of farmland.
More than a fifth of the smallholder farm households are below poverty and in the following years the situation is going to be worse due to climate change. A 4%-26% loss in net farm income is expected, together with an increase in the food demand as a consequence of the forecasted population growth.
What Telangana wants is to prevent hitches on their food supply chain that are likely to occur during critical situations (COVID-19 pandemic is a clear example) by adopting data-driven policy making and optimizing as much as possible the productivity of their fragmented agricultural production.
In order to achieve its goal the Telangana's government needs to assemble as much meaningful data as possible and to be ready to help farmers in distress with ad-hoc initiatives, possibly making farmers aware about forthcoming issues and teaching them how to tackle them [2][4].

1.2. Goals

DREAM is meant to satisfy the needs of farmers, agronomists and policy makers of Telangana, reaching the following 5 goals:

G1: Sharing of knowledge among farmers

G2: Tracking farmers' production

G3: Allowing farmers to get help

G4: Supporting agronomist work

G5: Providing data-driven decision tools to agronomists and policy makers

1.3. Definitions, Acronyms, Abbreviations

1.3.1. Definitions

District: portion of Telangana equivalent to the political district area. There are 33 districts in total.

Area: sub portion of the district. They are decided by the government stakeholders outside the system.

Help request to farmers: text request sent by a farmer to one or possibly many neighbor farmers.

Help request to agronomist: text request sent by a farmer to the agronomist who is responsible for the farmer's area, it is classified using a category.

Help request messages: text messages associated with a specific help request that are exchanged between the receiver and the sender.

Tip request: request sent by a policy maker to a farmer, for asking to write a tip.

Tip request messages: text messages associated with a specific tip request from a policy maker to a farmer.

Post: content that can be published in the forum. It can be of the following types:

- Tip: useful advice. It has no answers.
- Question: question that can be answered by other forum users. The answer can be voted.

Vote: either a like or a dislike received by a tip or an answer.

Score: any votable item in the forum has a score obtained by:
(number of likes) - (number of dislikes)

Star tips: forum's tips highlighted by the policy maker, they are displayed in the relevant information section of farmers in the same area of the tip.

Visit: event of an agronomist visiting a farmer. It has a creation date (timestamp), a proposed date, a time length and a status (scheduled, done, finalized).

Visit plan: it is a calendar where all the visits that an agronomist has to do are saved.

Visit messages: text messages associated with a specific visit that allow the agronomist and the farmer to have a direct contact point in case of need (i.e. the farmer wants to postpone the visit).

Visit-report: predefined questionnaire where the agronomist assesses the farmer's work and the observed situation. It is divided in 3 sections:

- *Basic*: mandatory part of the report containing the farmer's assessment;
- *Problem*: optional part to describe issues that the farmer is facing or has faced;
- *Weather*: optional part used to report adverse weather conditions or events.

Harvest-report: predefined questionnaire where farmers add all the information about the harvest of crops. It is divided in 3 sections:

- *Basic*: mandatory part of the report containing quantitative information and description of the harvest.
- *Problem*: optional part to describe issues related to that specific harvest.
- *Weather*: optional part used to report weather issues related to that specific harvest

Super agronomist: agronomist who has decided to be available to perform visits to farmers in the whole district.

Metric: it is a numerical value computed by the system for every farmer. It is based on features extracted from specific data referring to a specific period (i.e. a metric can be computed every 2 months based on the farmer production and the number of HRs sent). There can be many metrics, and each metric is computed for every farmer.

Key performance indicator: it is a numerical value computed by the system combining one or possibly multiple metrics associated with a farmer. Their objective is to be a performance indicator. High values of the KPI correspond to a good performance, and vice versa.

Well-performing farmer: a farmer with KPI values higher than average.

Under-performing farmer: a farmer with KPI values below than average**.**

Authorization code: alpha-numeric string used by users to register to DREAM. Regarding agronomists and policy makers, this code is given to them by a government admin outside the system. Regarding the farmer, this code corresponds to the VAT number. All the authorization codes are associated with an area.

Crop Category: each crop is categorized in one of the nine categories:
Cereals, Seeds, Pulses, Oil Seeds, Fruits, Vegetables, Spices, Fodder Crops, Commercial crops [1].

1.3.2. Acronyms

HR: Help Request
TR: Tip Request
FAQ: Frequently Asked Questions
KPI: Key Performance Indicator
VAT: Value Added Tax

1.3.3. Abbreviations

Auth code: authorization code
HR-message: help request message

TR-message: tip request message

1.4. Scope

In the forthcoming section, all the world and shared phenomena are listed [6].

1.4.1. World Phenomena

- A farmer is facing some troubles about adverse weather conditions.
- A farmer is facing generic troubles not directly related to weather.
- A farmer collects the crops.
- An agronomist carried out a visit.
- An agronomist cannot perform a visit.
- A farmer cannot receive a visit.
- A farmer is improving the production.
- A farmer is reducing the production.
- A policy maker implements a policy.

1.4.2. Shared Phenomena

The following table describes the world, shared and the machine phenomena, including the reference to which part controls the phenomena.
The phenomena have been divided in 5 categories for ease of reading: access, forum, HR, visit, data.

Phenomenon	**Who controls it**
Access	
A farmer signs up	World
An agronomist signs up	World
A policy maker signs up	World
A farmer logs in	World
An agronomist logs in	World
A policy maker logs in	World
A government admin logs in	World
Forum	
A farmer writes a post (tip or question)	World

An agronomist writes a tip	World
A farmer answers a question	World
An agronomist answers a question	World
A farmer votes an answer	World
An agronomist votes an answer	World
A farmer votes a tip	World
An agronomist votes a tip	World
A policy maker highlights a tips (promotes it to star tip)	World
A policy maker requests a tip from a well-performing farmer	World
A policy maker reviews a tip proposed by a well-performing farmer.	World
A government admin deletes a post.	World
A government admin deletes an answer.	World
A government admin deletes a farmer's account	World
A government admin generates a new authorization code for an agronomist's account	World
A government admin generates a new authorization code for a policy maker's account	World
A government admin manages the list of VATs valid as authorization code for farmers' accounts	World
The system shows meteorological forecasts to agronomists	Machine
The system shows meteorological forecasts to farmers	Machine
HR	
A farmer sends a HR to neighbor farmers	World
A farmer accepts a HR from another farmer	World

A farmer rejects a HR from another farmer	World
A farmer sends a HR to the area agronomist	World
An agronomist sends a HR-message	World
A farmer sends a HR-message	World
An agronomist marks a HR-message as FAQ	World
The system notifies receivers of a HR	Machine
The system notifies receivers of HR-messages	Machine
Visit	
An agronomist schedules a new visit	World
An agronomist reschedules an already planned visit	World
An agronomist marks a scheduled visit visit as done	World
An agronomist fills the visit-report	World
A farmer asks for changes of the scheduled visit	World
An agronomist sets on or off their super-agronomist state	World
An agronomist delegates a visit to a super-agronomist	World
A super agronomist accepts the delegation of a visit	World
A super agronomist declines the delegation of a visit	World
The system provides a visit-report to an agronomist	Machine
The system shows the visit plan to the agronomist	Machine
The system notifies the farmer about a	Machine

new visit insertion	
The system notifies the farmer about a visit modification	Machine
The system notifies the agronomist the presence of unfilled reports	Machine
The system notifies the agronomist the presence of visit to mark as done	Machine
The system notifies the agronomist the impossibility to delegate a super agronomist for a visit	Machine
The system notifies the agronomist that a visit has been delegated to a super agronomist	Machine
The system provides to agronomists an automatic generated year visit plan	Machine
The system updates the agronomists visit plan after a new farmer sign up in their area	Machine
Data	
An agronomist fills the visit-report.	World
A farmer fills the harvest-report	World
A government admin deploys a metric	World
A government admin undeploys a metric	World
A government admin deploys a KPI	World
A government admin undeploys a KPI	World
The system queries the government API	Machine
The system queries the weather forecast API	Machine
The system presents latest computed KPIs to policy makers	Machine
The system presents latest computed KPIs to agronomists	Machine

1.5. Revision history

- 23 December 2021, version 1.0
  - First Implementation
- 08 January 2022, version 1.1
  - Some small clarifications about the deployment and undeployment of KPIs
  - Adjusted consistency of some requirements
  - Adjusted the process of retrieving the area of users from auth codes
  - Adjusted the procedure of sign up of users
  - Corrected a transcription error in product function 25
  - Added crop categories definition
  - Adjusted dimensions of activity diagrams images
  - Updated name of product function 16
- 05 February 2022, version 1.2
  - User characteristic section moved from 2.3 to 2.2
  - Slight modification to the second farmer scenario

1.6. Document Structure

- **Chapter 1** gives an introduction about the purpose of the document and the goals, with its corresponding specifications such as the definitions and acronyms. Also the scope of the project is described, as well as the world and shared phenomena. Finally, the revision history of the document and the references are also available in this chapter.

- **Chapter 2** contains the overall description of the project. In the product perspective are included the statecharts of the major functions of the application and the model description through a class diagram, to better describe what the project is meant to be. Product functions are also described, to better understand the functionalities of the system. In the user characteristics section are explained the types of actors that can use the application. Finally, the domain assumptions are included.

- **Chapter 3** represents the body of the document. It contains the interface requirements, which are hardware interfaces and software interfaces. Then, it lists some scenarios to show how the system acts in real world situations, followed by the description of the functional requirements, using use cases and sequence diagrams. All the requirements necessary in order to reach the goals are given, linked with the related domain assumptions. Lastly, the non-functional requirements are defined through performance requirements, design constraints and software system attributes.

- **Chapter 4** includes the formal analysis using alloy code, together with some instances of the built model and the explanation of the performed analysis.
- **Chapter 5** shows the effort spent by each member of the group.
- **Chapter 6** includes the reference documents.

2. OVERALL DESCRIPTION

2.1. Product perspective

DREAM wants to provide the Telangana government a dynamic tool to improve and stabilize their agriculture production, supporting their farmers and agronomists in an active way, but also combining all the data generated by the farmers and agronomists with other data coming from external data sources, to extract valuable data that can allow policy makers to make more informed policies.

By means of a forum, DREAM wants to help farmers to increase their productivity, allowing collaboration among them and the spread of best practices and knowledge in general. This is useful because there are many smallholder farmers who may work in similar conditions, and therefore they could benefit from techniques used by neighboring farmers.

Every farmer using DREAM should be able to report every crop harvest and provide quantitative information about it, specifying if they had problems regarding that specific harvest, such as weather related problems or any generic problem that led to a decrease in productivity.

The farmers using DREAM should be able to ask agronomists or neighbor farmers for help using direct help requests. This type of contact should help them in case of issues that need a quick response: it may be given not only by agronomists, but also by farmers that have faced the same issue. If agronomists receive lots of similar requests they have to be able to publish the more relevant resolutions in order to reduce not necessary work and speed up the resolution process.
The system aims to help agronomists in planning their job by providing an easy and handy way to schedule their visits to farmers, dealing also with extra visits for those farmers who need more help and should be visited more often, but also allowing the farmers to ask for a change of the visit date.

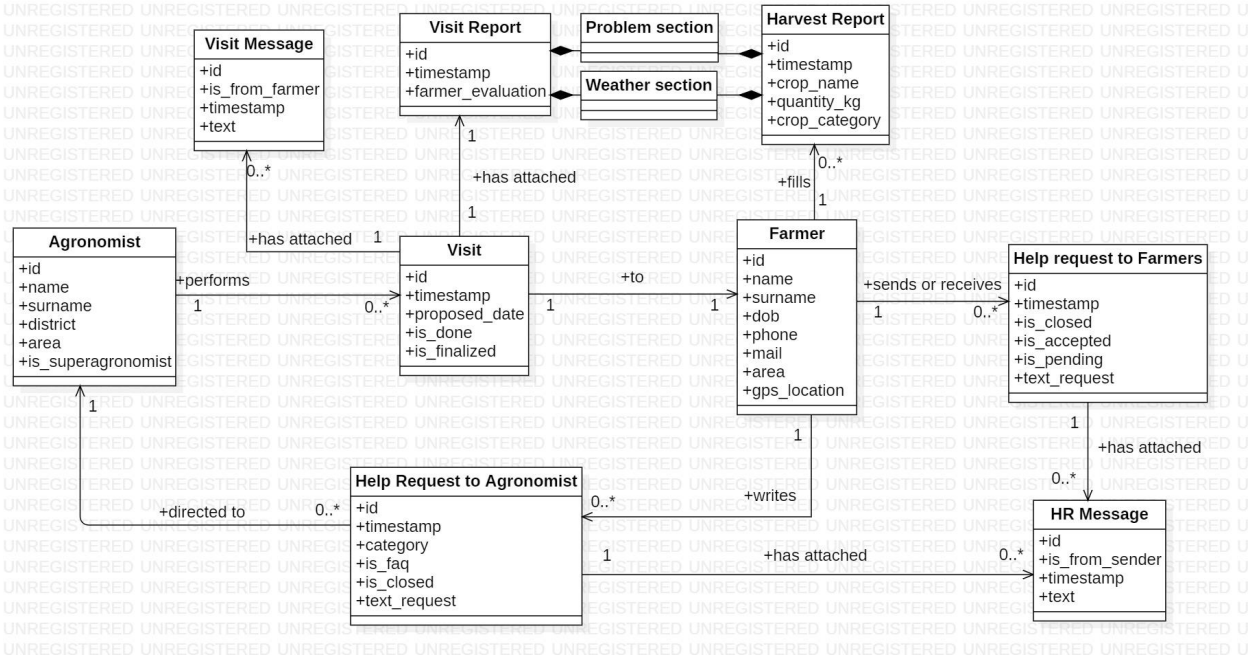
DREAM has to combine data coming from its internal traffic and from other external resources (weather forecast services, sensors deployed on the territory and irrigation systems) and provide to agronomists and policy makers KPIs (key performance indicators): using them, agronomists should be able to spend their work time more productively visiting more farmers that are underperforming, and policy makers can exploit them to understand which policy is better to pursue, and monitor the effects of previously adopted ones, together with the impact of the agronomists' work.

2.1.1. Class diagrams

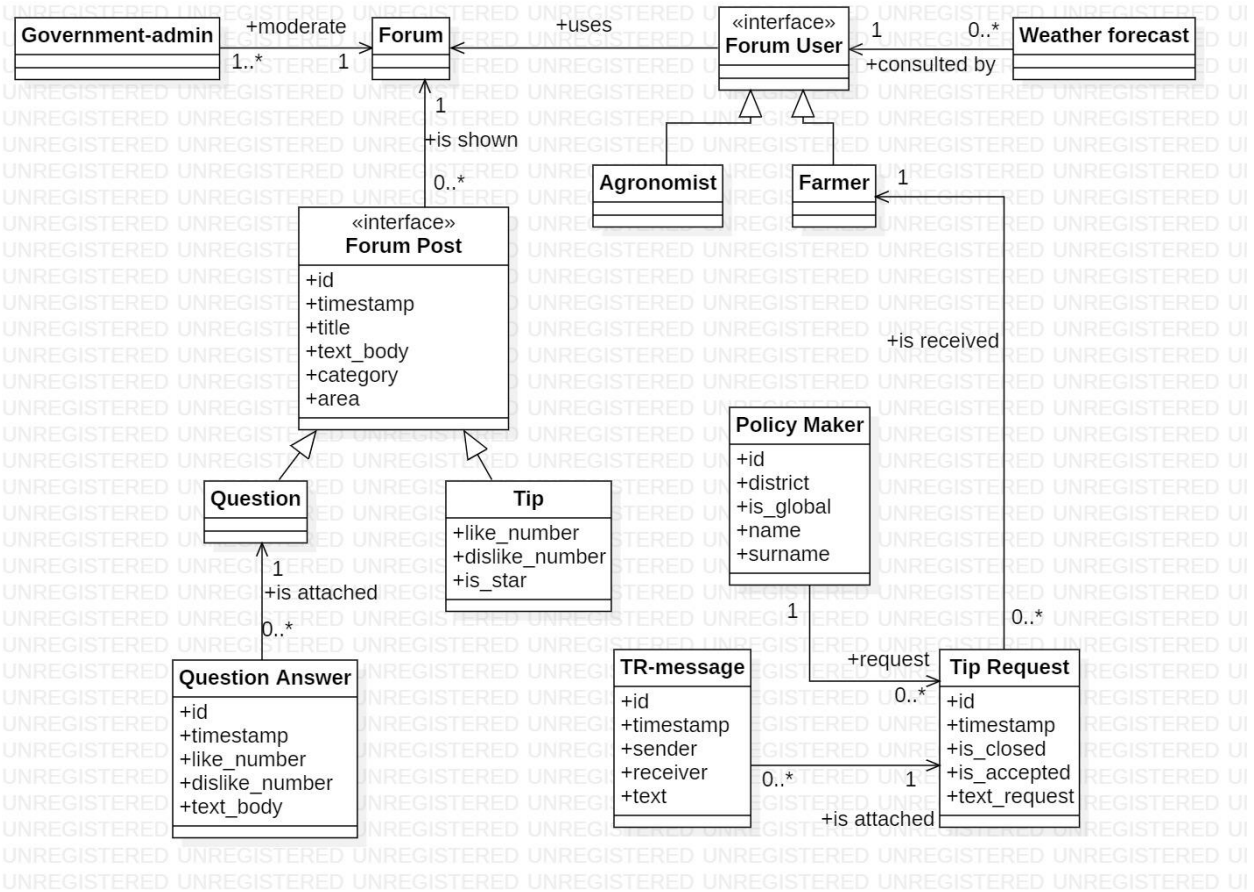
The class diagram has been subdivided in 3 parts for the sake of clarity, and the full description of classes is presented only in the first appearance.

Visits and HR view:

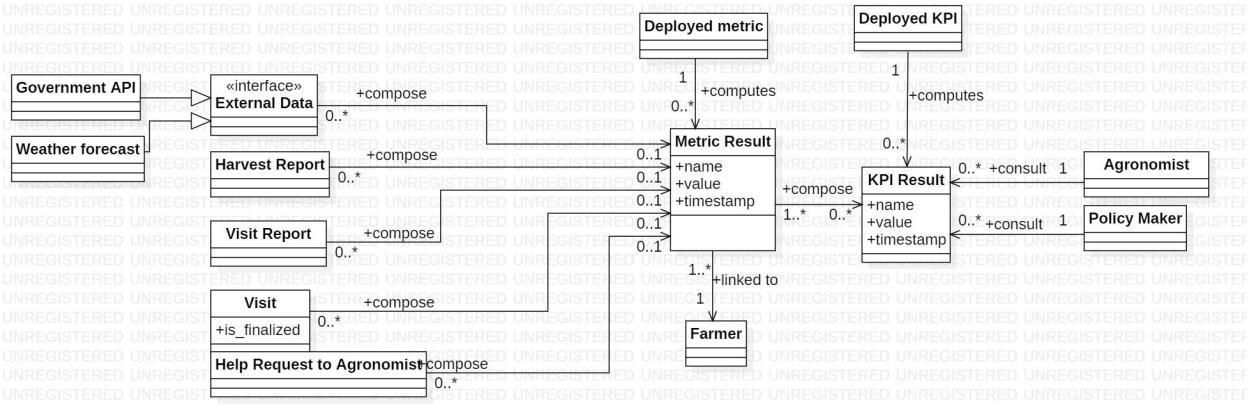
This class diagram is describing one single area: the farmer and the agronomist in the diagram belong to the same area.



Forum view:



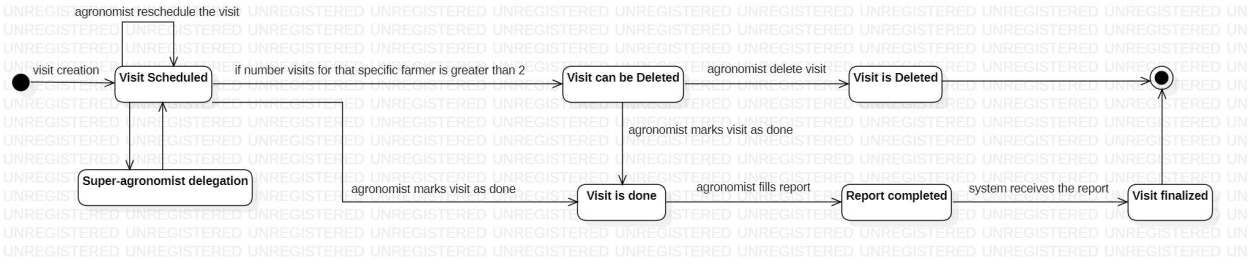
Data management view:



2.1.2. Statechart

- **Visit**
  When a visit is created either by the system or manually by the agronomist it has the state “scheduled”.
  A visit can always be modified (date and duration) by the agronomist of reference as long as its state is scheduled.

An agronomist can request the system to delegate a visit to a super-agronomist: if the delegation procedure ends positively then the visit is assigned to another agronomist (the super-agronomist), otherwise it is reassigned to the same agronomist.
If the total number of scheduled visits is greater than 2 for a certain farmer then a scheduled visit can be deleted if needed otherwise the deletion is prevented by the system.
The termination of the physical visit can be confirmed in the system by the agronomist, and after the confirmation a visit-report has to be filled.
Finally, after the fulfillment of the report, the system marks the visit as finalized.
Before the agronomist confirmation agronomist and farmer can exchange visit messages.
The completion of the visit report is mandatory to close the visit lifecycle.
The system constantly and severely notifies the agronomist if an available visit report has not been filled or a visit carried out in the past days has not been marked as done yet.



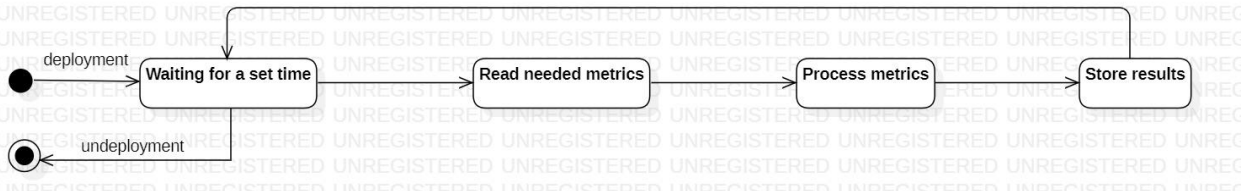
- **HR to farmers**
  When a farmer sends a new HR to other farmers the request is sent to all farmers within 10 km and it is in “request pending” state.
  When farmers receive a HR they can read what the HR is about, deciding to accept or reject it.
  If the HR is accepted, then the receiver and sender can chat exchanging HR-messages.
  In any moment after the chat has been opened both sender and receiver can decide to stop the exchange of HR-messages. The HR ends in “chat closed” state, the history of the chat will remain to both receiver and sender.

- **HR to agronomist**
  When a farmer sends a new HR to an agronomist, the request is sent to the agronomist responsible for the area where the farmer is located.
  The agronomist can only answer to a new HR, and after the agronomist's first answer they both can chat exchanging HR-messages until the agronomist decides that the problem is resolved.
  After the chat closure the system allows the agronomist to mark and publish that specific chat as FAQ.

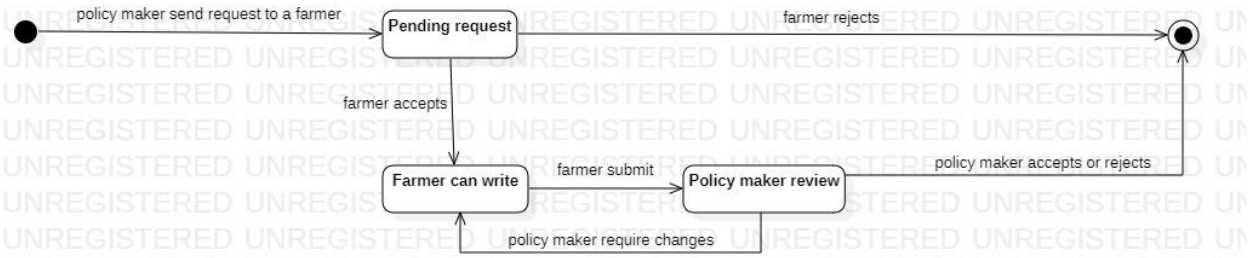
- **Metrics and KPIs**
  The possibility to deploy and undeploy metrics and KPIs is a functionality of the system available for the government admin.
  It is important to notice that the metric and KPIs cannot be deleted by the system, their definition is embedded in the system, the government admin can decide which of them are available to users employing them, and vice versa deploying them.
  After the deployment for both metrics and KPIs the system periodically will go through a data collection and data processing phase, storing the results and then starting again the routine.
  The results stored are respectively what in the class diagram (2.2.1 data management view) are called "metric results" and "KPI results".
  After the state "stored results" of each iteration, the system provides agronomists and policy makers the new computed KPIs using the latest available metrics in turn computed processing the latest data available.

  Metric statechart

  KPI statechart



- **Tip-request**
  The policy makers can request farmers to add new tips inside the forum.
  They can send the request to every farmer, but the feature is aimed for well-performing farmers.
  When the policy maker sends the tip request to a farmer it starts in the “pending request” state, then the farmer can decide either to accept and submit the draft tip or to reject.
  After the farmer’s submission the policy maker can review what the farmer wrote and ask for changes.
  The policy maker during the review can decide to accept and publish the tip or to reject it without any possibility for the farmer to modify it again.



2.2. User characteristics

The actors involved in the system are:

- **Farmer**
  It is a user that can access a section with relevant information (*see product function 13*) according to their area and their type of products. This kind of user can use the harvest report section to insert data about its production, and any relevant problem related to it. Through a forum, the farmer can also post tips and questions, answer questions, give evaluations to answers and tips. The farmer can send HR (Help Request) to other farmers and agronomists, and can accept or decline an HR coming from another farmer. Finally, the farmer can contact an agronomist to modify the proposed date of a visit.

- **Agronomist**
  It is a user that can access a section with relevant information about the weather and the performances of the farmers in that area. They can receive help requests from farmers and answer them, possibly posting them as FAQs. They can post tips in the forum, which will be marked with their area and the selected category. They can visualize their default visit plan and update time and day of the visits. They can add visits and delete them if the mandatory two visits to each farmer are either done or scheduled. They can interact with the farmer through a direct chat (when requested by the farmer, through a visit-message) to update the time and date of a visit. They can go to a visit, mark the visit as done and

then fill in a visit report. They receive notifications about visits scheduled in the past days not yet marked as done, and about reports of past done visits not yet finalized. They can mark themselves as super agronomists anytime in the system.

- **Super agronomist**
  It is an agronomist that can cover up the visits that the agronomists are not able to cover in a specific district. They can accept or refuse a delegation request for a visit.

- **Policy maker**
  It is a user which is in charge of one single district. They want to identify those farmers who are performing well, especially when they demonstrate to be able to face adverse weather conditions, and monitor whether the agronomist's work is having some effect on the farmers productivity: they can do this by visualizing data regarding the performance of farmers and agronomists using the available KPIs results. They can request the well-performing farmers to add tips in the forum. They can also highlight tips (promote to star tip) of the district they are responsible for. There is also one policy maker account in the system that can manage all districts, so in total there are 33 policy maker accounts associated with districts and 1 associated to all the 33 districts.

- **Government admin**
  It is a user whose function is the management of all the users in the system. In DREAM there are two government admins. They are responsible for the creation of the authorization codes for farmers, agronomists and policy makers profiles. For the farmers, they decide which VATs are valid for signing up, while for the other two they generate the authorization code using the system but deliver it outside the system.They are also the forum moderators, they have the right to delete posts or answer inside the forum if they are not acceptable in terms of quality. They have full control over the other actors, they can delete a farmer account and reset a policy maker or agronomist account by generating a new authorization code, allowing a new sign up.

2.3. Product functions

1. **Signing up**
   This functionality is available to every farmer, agronomist and policy maker. Each of them, to register to DREAM, has to use an authorization code. They can insert it in the registration phase (the definition of the authorization code for all the actors is given in section 1.3.1). The system retrieves automatically the area related to each user from the auth code inserted in the sign up process.

2. **Logging in**

This functionality is available to every farmer, agronomist, policy maker and government admin. Each of them, by providing valid credentials, has to be able to access their account on the system.

3. **Filling in the harvest-report**
   This functionality is available to every farmer. A farmer can insert the data inside the system using this functionality. They can add the crop name, the quantity (in KGs) of the production, the category of the crop, and any problem faced with this production. The harvest report will have a date and time, setted automatically by the system (date and time of submission).

4. **Sending a HR to farmers**
   This functionality is available to every farmer. A farmer can send a HR to all the farmers whose distance from them is less than 10 km.

5. **Sending a HR to agronomist**
   This functionality is available to every farmer. A farmer can send a HR to the agronomist of his area.

6. **Receiving a HR by a farmer**
   This functionality is available to every farmer. The farmers, once they receive a HR from a farmer, can decide to accept it or not. If yes, HR-messages between them can be exchanged.

7. **Receiving a HR by an agronomist**
   This functionality is available to every agronomist**.** Agronomists can receive HRs from farmers of their area. After an agronomist answers a HR, HR-messages can be exchanged between them.

8. **Using HR-messages**
   This functionality is available to every farmer and agronomist. Once a farmer accepts a HR, and once an agronomist answers for the first time to a HR, HR-messages can be exchanged between the receiver and the sender. They are text associated with the name of the sender as a common online chat. HR-messages can be exchanged as long as the HR is not closed.

9. **Posting a FAQ**
   This functionality is available to every agronomist. After a HR between the agronomist and a farmer has been closed, the agronomist can decide to flag as FAQ a HR: the title of this FAQ will be the same as the HR, while the body of the FAQ will be all the HR-messages exchanged, they are classified using the HR category.

10. **Posting a tip**
    This functionality is available to every farmer and agronomist. They can insert a tip. Each tip has information about the area of the farmer or agronomist who posted it and the category.

11. **Posting a question**

This functionality is available to every farmer. A farmer can post a question. It has information about the area of the farmer who posted it and the category.

12. **Posting an answer to a question**
    This functionality is available to every farmer and agronomist. They can post answers to existing questions. It has information about the farmer or agronomist who posted it.

13. **Voting inside the forum**
    This functionality is available to every farmer and agronomist. They can vote (like or dislike) for a specific tip or answer as much time as they want but only the last is counted. The positive vote consists of +1, and the negative vote consists of -1 on the total score of the tip or answer. The votes are anonymous.

14. **Checking relevant information**
    This functionality is available to every farmer. Farmers can access relevant information of two types: weather information and tips. Each farmer can see the most two relevant tips for each type of product that they inserted in their harvest reports already submitted. In case of no submission of any harvest reports, farmers see the most relevant tips in their area. All the star tips of the farmer area are visualized in this section. Long and short time meteorological forecasts about the area of the farmer will also be visualized.

15. **Initialization of a visit plan**
    This functionality is available to every agronomist. The visit plan of an agronomist is created 15 days before the beginning of the year, with two default visits for every farmer in the agronomist's area, since for each farmer there must be at least two visits per year.

16. **Giving a contact point for a visit**
    This functionality is available to every farmer. Once a visit is inserted in an agronomist’s visit plan, or something about it is updated, the farmer who will receive that visit is notified by receiving a notification. From the notification the farmer can decide to use a chat to exchange visit messages and ask the agronomist to update the date or time of that visit. The farmer can also decide to not use the chat and keep the information of the visit unchanged.

17. **Adding a visit**
    This functionality is available to every agronomist. An agronomist, by monitoring the performances of the farmer in his area, or after having received a HR by some farmers, can schedule some new visits to investigate what are the reasons for those events. These additional visits will be inserted in their visit plan.

18. **Changing a visit state**
    This functionality is available to every agronomist. While the state of a visit is scheduled, it is possible to update its information. Moreover, if a visit is scheduled a farmer can use the visit messages to ask for an update of this visit. After the agronomists have done a visit, they are supposed to change the state of that visit from scheduled to done. If a visit is done, it is not possible anymore to update it or ask for an update by the farmer. If the agronomists do a visit, not changing the state of the visit to done, the days after they will receive a notification, reminding them to change the state to done.

19. **Filling the visit-report**
    This functionality is available to every agronomist. After the agronomist has made a visit, and therefore the state of that visit is done, the agronomist has to fill and confirm the visit-report. The agronomist can fill the basic section and the issue section, to report about some issues faced by the farmer. They can also fill the weather section, reporting if the farmer is facing some adverse weather conditions. After having filled the report, the agronomist submits it, confirms the report and as a result the visit state becomes finalized.

20. **Giving availability to be a super agronomist**
    This functionality is available to every agronomist. An agronomist can decide to give the availability to work in the whole district, becoming a super agronomist, and helping other agronomists in the district with their visits. A super agronomist can decide to quit and go back to being a simple agronomist.

21. **Delegation of a visit to a super agronomist**
    This functionality is available to every agronomist. An agronomist, having a huge amount of visits to handle, can decide to delegate some of them to a super-agronomist: a visit will be given to the closest super agronomist with respect to the farmer that necessitates that visit. A super agronomist can refuse a visit, if so the visit will be given to the second most close super agronomist, and so on. If none of the super-agronomists available in all the districts accept that visit, the initial agronomist will handle it.

22. **Monitoring of farmers performance**
    This functionality is available to policy makers and agronomists. Policy makers, looking at the KPIs of farmers in their district, and agronomists, looking at KPIs of farmers in their zone, can figure out how the farmers are performing and understand which ones are performing well or bad.

23. **Requesting a tip**
    This functionality is available to policy makers. When policy makers consult KPIs and see well performing farmers they can request them to write a tip for the forum in order to share their knowledge. After the farmer's response they can eventually accept, abort the request or to send it back for modification to the farmer, each tip request is associated with a chat where farmer and policy maker can communicate using tip request messages.

24. **Promoting a tip to star tip**
    This functionality is available to every policy maker. Policy makers can promote tips (belonging to their district) to star tips increasing their relevance in the forum .

25. **Monitoring of agronomists performance**
    This functionality is available to policy makers and agronomists. In the same way as described in *product function 22,* they can consult KPIs of farmers that point out effects of the productivity after interactions with agronomists.

26. **Managing authorization codes**

This functionality is available to every government admin. Government admin can add in the system the list of VATs that correspond to farmers who are allowed to sign up in the system. They can also generate the auth code for the static accounts of agronomists and policy makers. These auth codes allow a new sign up and the update of personal data, settings and credentials associated to these static accounts.

27. **Resetting a policy maker or agronomist's account**
    This functionality is available for every government admin. The static accounts of policy makers and agronomists cannot be deleted by the system. However the government admin can decide to regenerate the auth code for one of those profiles, preventing the log in with the current credentials, but allowing a new sign up in the same account by using the new generated auth code.

28. **Deleting a farmer account**
    This functionality is available to every government admin. Farmer accounts can be deleted by the government admin: the visits associated with them are automatically marked as finalized and every HR where the farmer is taking part is closed, KPIs and metrics associated with the deleted farmer are not calculated anymore, while all their interactions in the forum are kept.

29. **Creating visits for a new farmer account**
    This functionality is available to every agronomist and farmer. When new farmers successfully sign-up, two visits are automatically added to the visit plan of the agronomist in their zone.

2.4. Assumptions, dependencies, constraints

2.4.1. Domain assumptions

D1: Data about humidity of soil and irrigation system is made available by an external system (government API).

D2: Weather information is obtained through an API provided by an external meteorological service (weather forecast API).

D3: There is an analysis team external to the system that defines metrics and KPIs.

D4: The KPIs and metrics definition is meaningful.

D5: The areas are defined by the government stakeholders such that the workload can be managed by one agronomist, in the worst case with the help of super-agronomists.

D6: An agronomist will always have the time to deal with all the HRs sent by farmers of their area.

D7: There is the availability of one agronomist for each area.

D8: The majority of the farmers of Telangana use DREAM and the application is uniformly distributed on the territory.

D9: For each district, there is only one policy maker referring to it. In addition, there is another policy maker referring to all districts.

D10: Farmers have the availability of a device with internet access, and the GPS location at least in the registration phase.

D11: Each farmer has a VAT number.

D12: The government will instruct the role of government admins to well trained and trusted people that will act according to what the goals of the system are.

3. SPECIFIC REQUIREMENTS

3.1. External Interface Requirements

3.1.1. Software Interfaces

The system uses external data coming from 2 APIs: the government API and weather forecast API.
The development team is free to choose which weather forecast API to use among all the possible available, but it is required that this API provides both short and long term predictions updated at least 6 times every 24 hours.
The government API has to provide data about soil humidity and irrigation systems at the end of every day the development team has to take care that the system acquire the data from the API in no more than 30 minutes from when it is available

3.2. Functional Requirements

3.2.1. Farmer Scenarios

- **Scenario 1: Sign-up**

  Bob is a farmer who lives in Adilabad, one the 33 districts of Telangana. Recently, the government let every farmer know that there is a new web application, called DREAM, that supports their farming activity. Many of his friends already use DREAM, so he decides to sign up. Since it is the first time that he uses the application, he is required to sign-up as a farmer. He enters his first name, last name, his VAT number as auth code, in the fields of the sign-up page. The auth code is accepted: now Bob can set his new credentials, which he will use to access from the log-in page, the next time he will open the DREAM web application.

- **Scenario 2**: **Filling-in the harvest report**

  Carl lives in the district of Medak and is a hard-working farmer with a strong sense of duty. To improve his farming activity, he started to use DREAM. In order to get the most benefits from this web application, he needs to keep track of his production by filling in a harvest report directly inside the application. This time, he has to register his last production of potatoes. He logs in into DREAM, and goes to the “Fill in the harvest report” functionality. Here, he adds crop name (“potatoes”), quantity in kilograms, category of the crop (i.e., “horticulture crops”) and any problem he faced, for example the heavy rain he had to deal with, a few days ago. The report is now completed and inserted into the system.

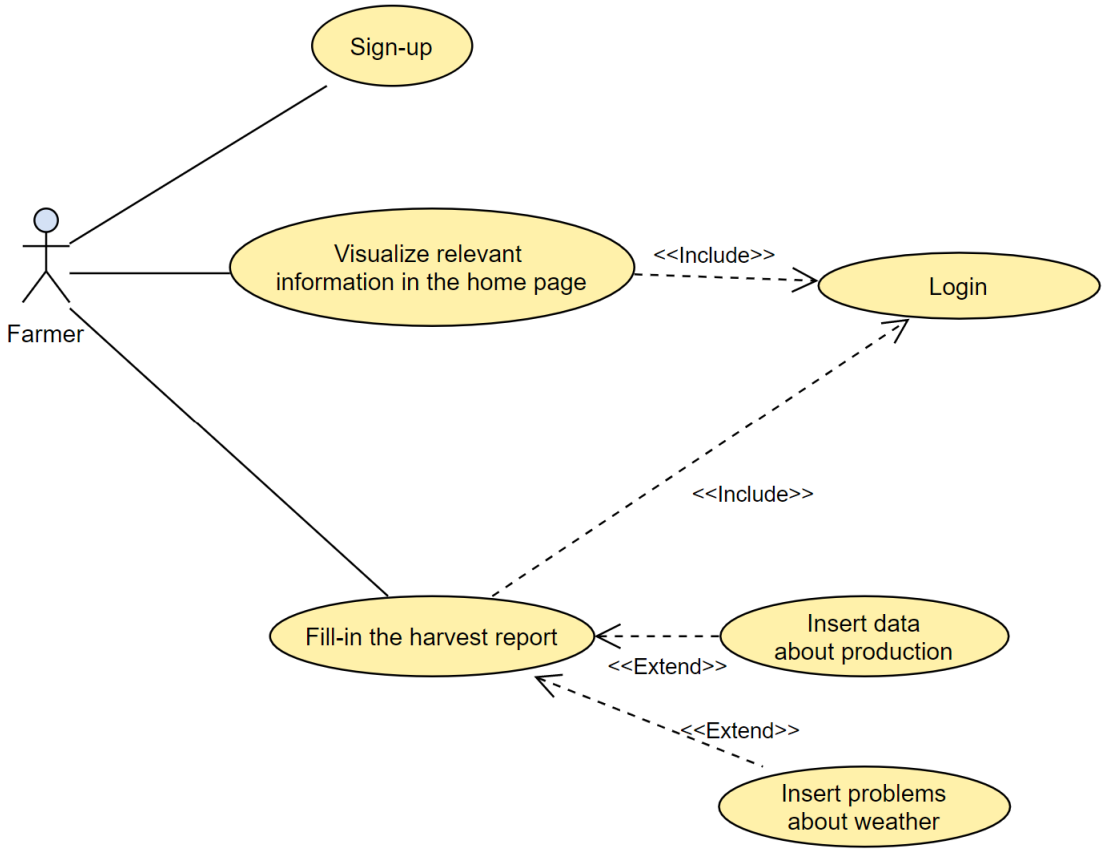
- **Scenario 3: Visualize relevant information and navigate the forum**

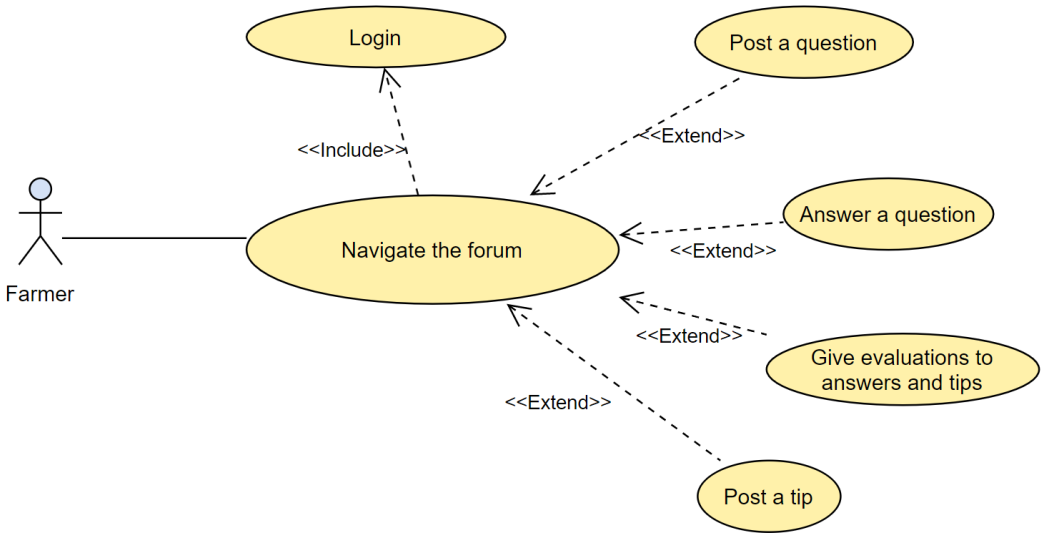
Alice is a farmer that has been using the DREAM app for a couple of weeks. As almost every day, she turns on her laptop, opens DREAM and logs-in into her farmer account. She takes a quick look at the relevant information section, where she generally can find updated, local weather information and useful tips related to her area and to the crops she produces: in fact, the system learnt from her past harvest report what she is most interested in. Later, she decides to spend some time navigating the forum. She wants to check if somebody answered her last question. She finds out that five farmers in her area answered. Although she hoped that at least an agronomist would respond, she notices that one of the answers has been largely positively voted, therefore she reads it carefully and really appreciates it. Alice strongly believes in collaborating with other farmers, therefore before logging-out she writes in the forum a tip, hoping that someone will find it useful.

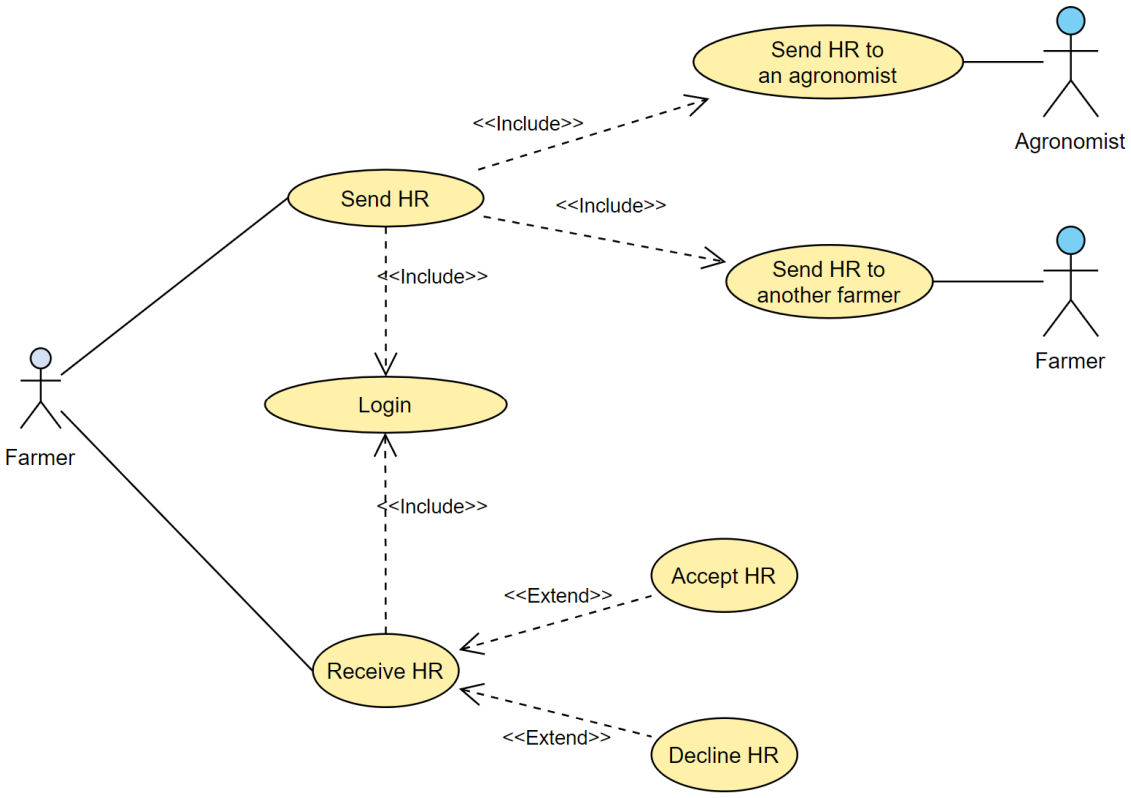
- **Scenario 4: Sending a HR to the agronomist**

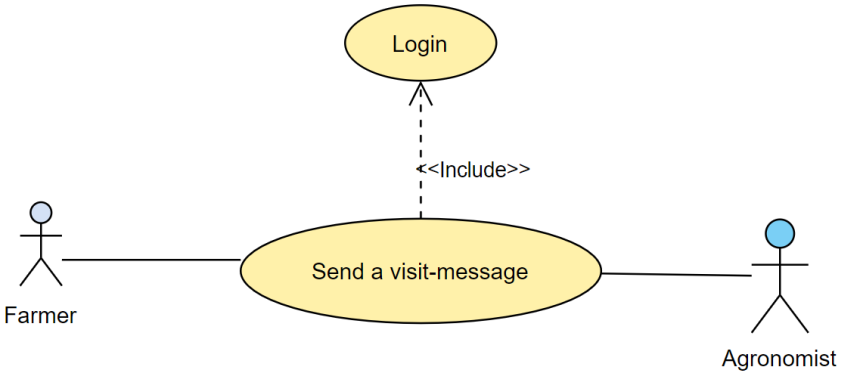
  The young farmer David is trying to protect his crops from locusts, but he is new in farming, and he doesn't have much experience. His last crop was assaulted by locusts, and much was destroyed. He decides to use the Help Request (HR) functionality available in the DREAM web application. Therefore, he opens the application and logs-in into his farmer account, then he sends the HR to the agronomist of his area and waits for a response. After a while, he gets a notification on his device: the agronomist answered with a useful suggestion about a repellent. The HR threads can now be terminated.

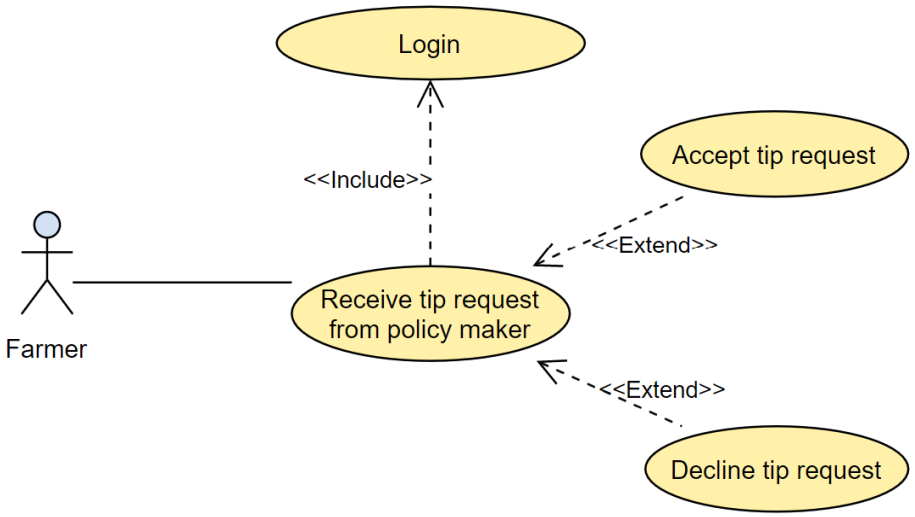
3.2.1.1. Use case diagrams





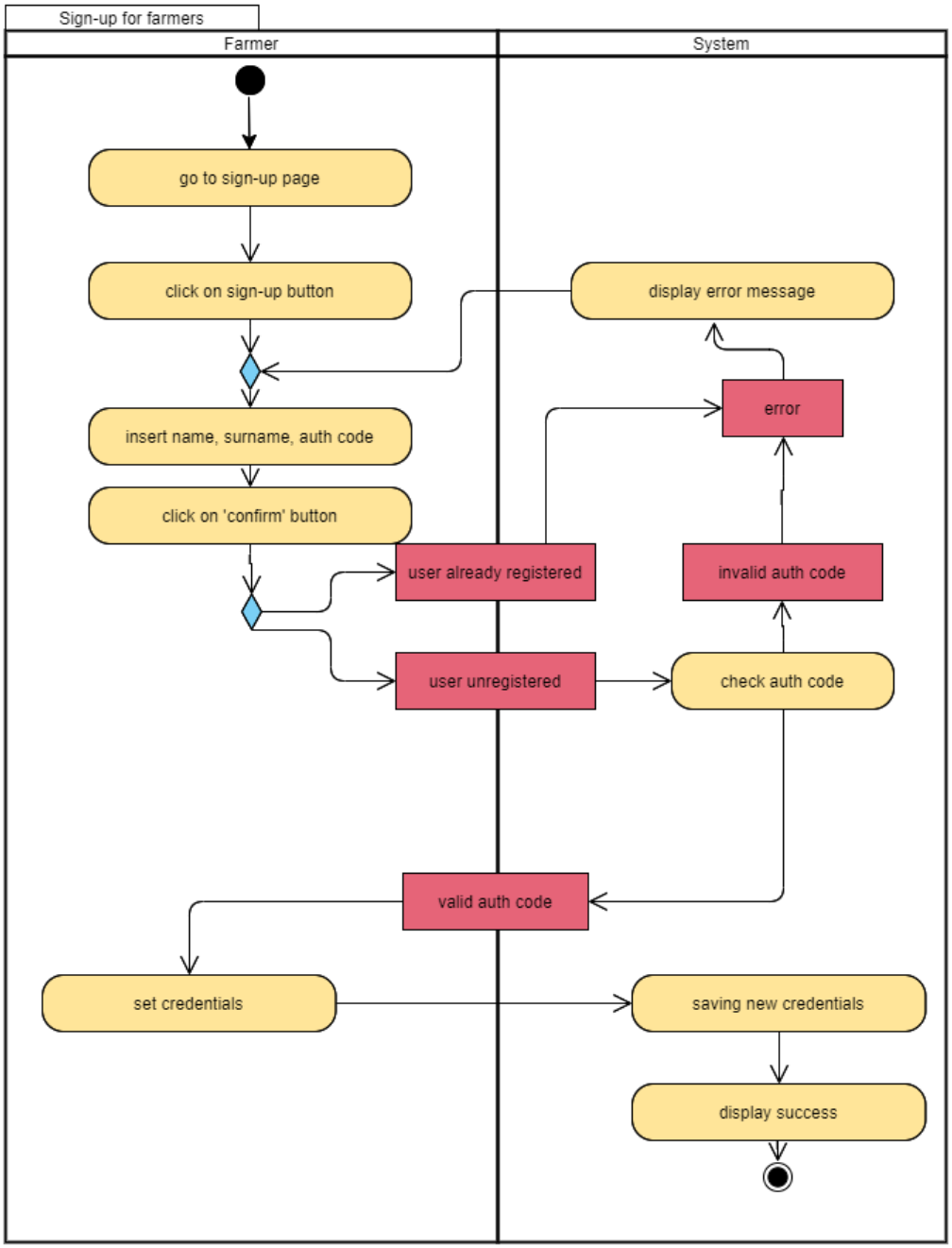






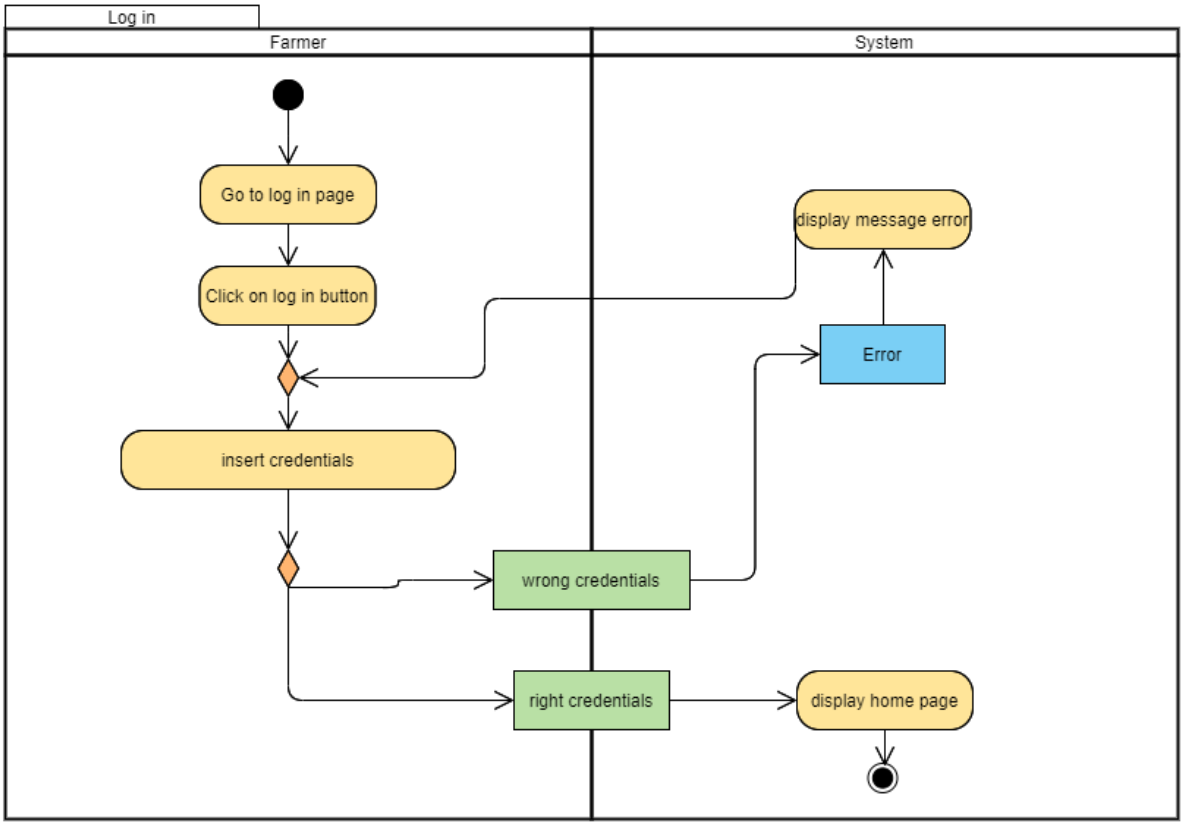
3.2.1.2. Use case tables and activity diagrams

Name	Sign-up Farmer
Actor	Farmer
Entry conditions	The farmer opened the DREAM web application. The farmer has an authorization code.
Event flow	1. The application shows the Sign-up page. 2. The farmer clicks on “Sign-up”. 3. The application displays the following mandatory fields: full name and last name, authorization code. 4. The system, using a location API, obtains the location of the farmer. 5. The farmer fills-in the mandatory fields. 6. The farmer clicks on the “Confirm” button. 7. The app allows the farmer to set the credentials. 8. The farmer writes the credentials. 9. The farmer click on ‘confirm’
Exit condition	The farmer registration is successfully completed and the farmer’s data is saved into the system. The app shows the screen with the confirmation “You have successfully registered!”
Exceptions	1. The farmer inserts a wrong authorization code. The application displays an error message and the farmer is invited to try again. 2. The farmer is already registered. The application shows the message “You have already registered! Log-in with your credentials”. The farmer is invited to try again.



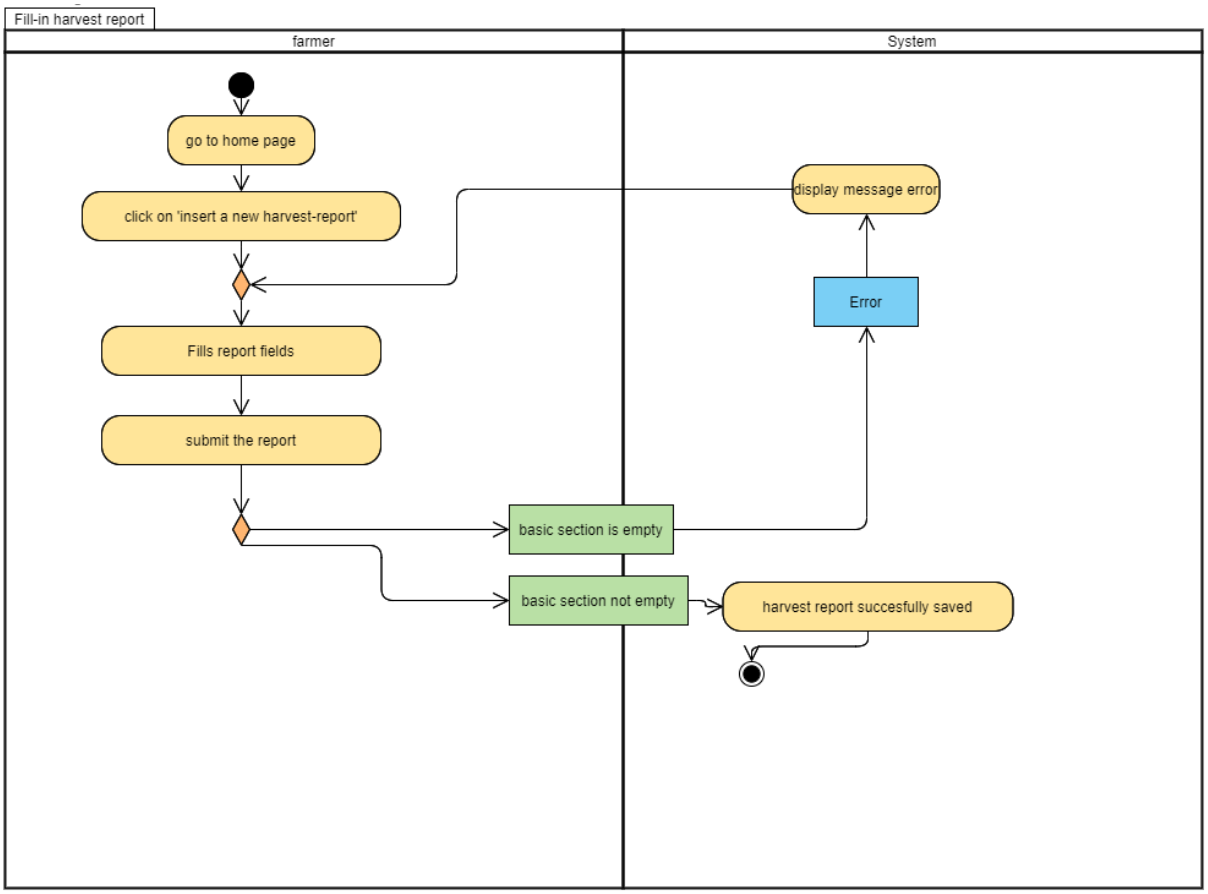
Name	Log-in farmer
Actor	Farmer
Entry conditions	The farmer opened the DREAM web application. The farmer is registered into the system.
Event flow	1. The system displays the log-in page. 2. The farmer inserts the credentials into the mandatory fields.

	3. The farmer clicks on the “Log-in” button. 4. The system checks for the validity of the credentials. 5. The system shows the home page of the application
Exit condition	The farmer is successfully logged-in.
Exceptions	The farmer inserts the wrong credentials. The system displays an error message and invites the farmer to try again.



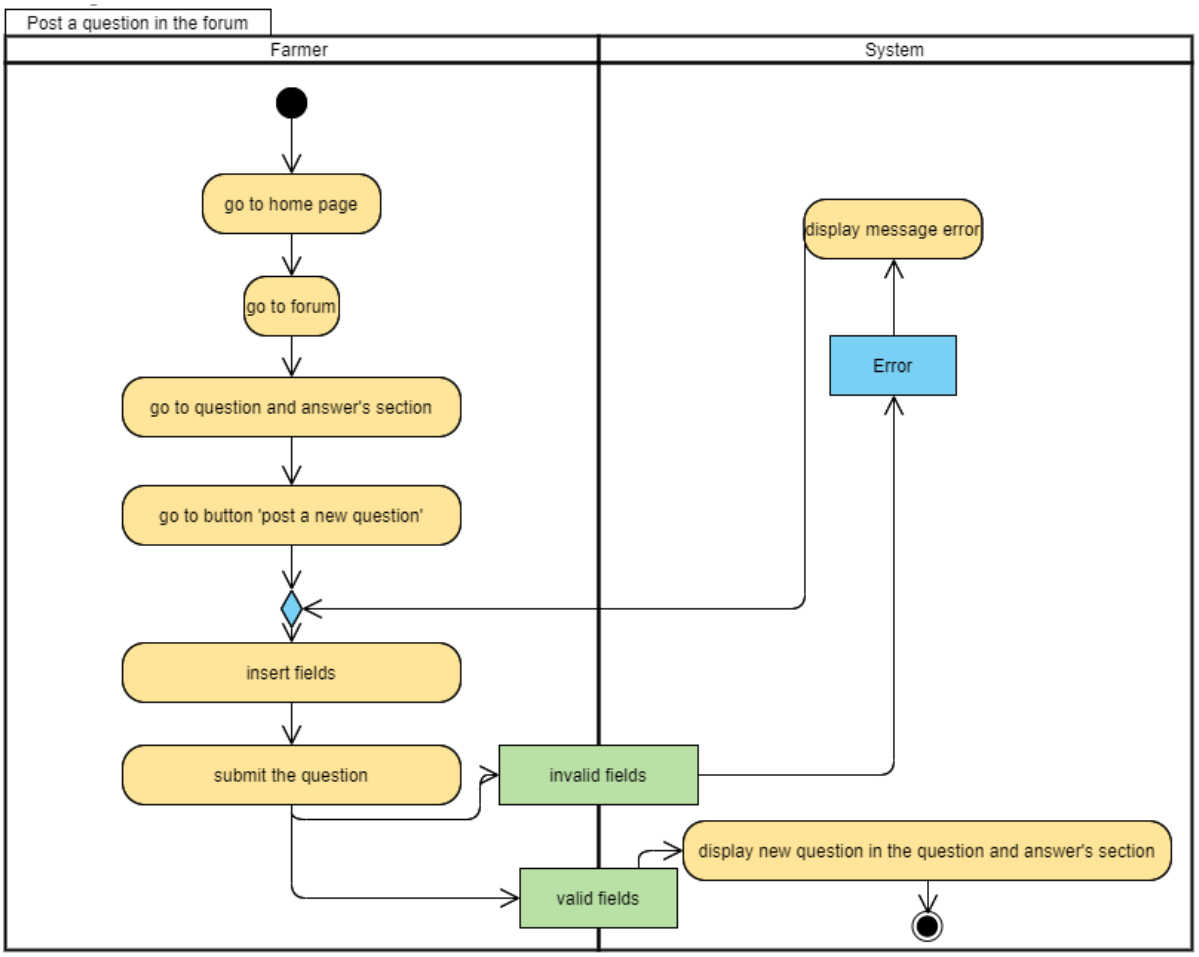
Name	Fill-in the harvest-report
Actor	Farmer
Entry conditions	The farmer is logged-in.
Event flow	1. From the home-page, the farmer presses on “Fill-in new harvest report” 2. The application displays a page with: a. the name of the farmer, the area and the date, all automatically compiled; b. the basic section (mandatory) : “Crop name”,“Quantity” (in kg), “Crop category”; c. the problem section (optional); d. the weather section (optional);

	3. The farmer fills in the mandatory fields and briefly describes any problem faced in the optional field. 4. The farmer clicks on the “Send harvest-report” button.
Exit condition	The system correctly receives the filled-in harvest-report and shows the message “Harvest-report received!”.
Exceptions	The farmer doesn’t fill in one or more mandatory fields. The system displays an error message, telling which field(s) are missing.



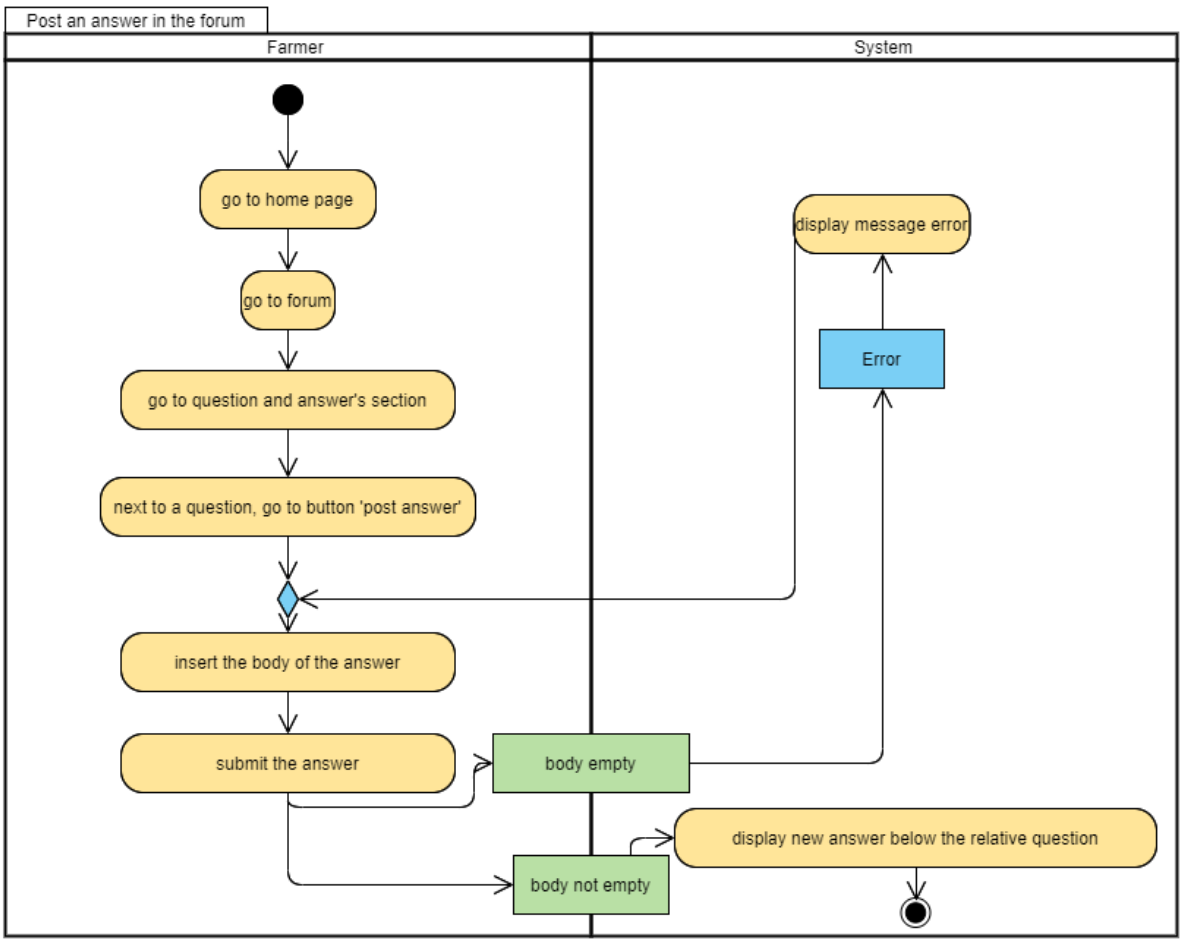
Name	Post a question in the forum
Actor	Farmer
Entry conditions	The farmer is logged-in.
Event flow	1. From the home-page, the farmer presses on the “Forum” button 2. The system displays a page divided in two sections: “Questions & Answers” and “Tips”. 3. In the “Questions & Answers” section, the system

	displays the posted questions, from the most recent to the less recent, and the related answers, ordered according to the score. At the top of the section, a button “Post a new question” is displayed. 4. The farmer presses on the “Post a new question” button. 5. The system shows a field “Category”, a blank field, an auto-filled field with the area of the farmer, and a “Submit” button. 6. The farmer selects the category of the question and writes in the field the question. 7. The farmer clicks on the “Submit” button.
Exit condition	The system displays the new question on the top the page
Exceptions	The farmer clicks on the “Submit” without selecting the category and/or filling in the blank field. The system displays an error message. The system invites the farmer to try again.

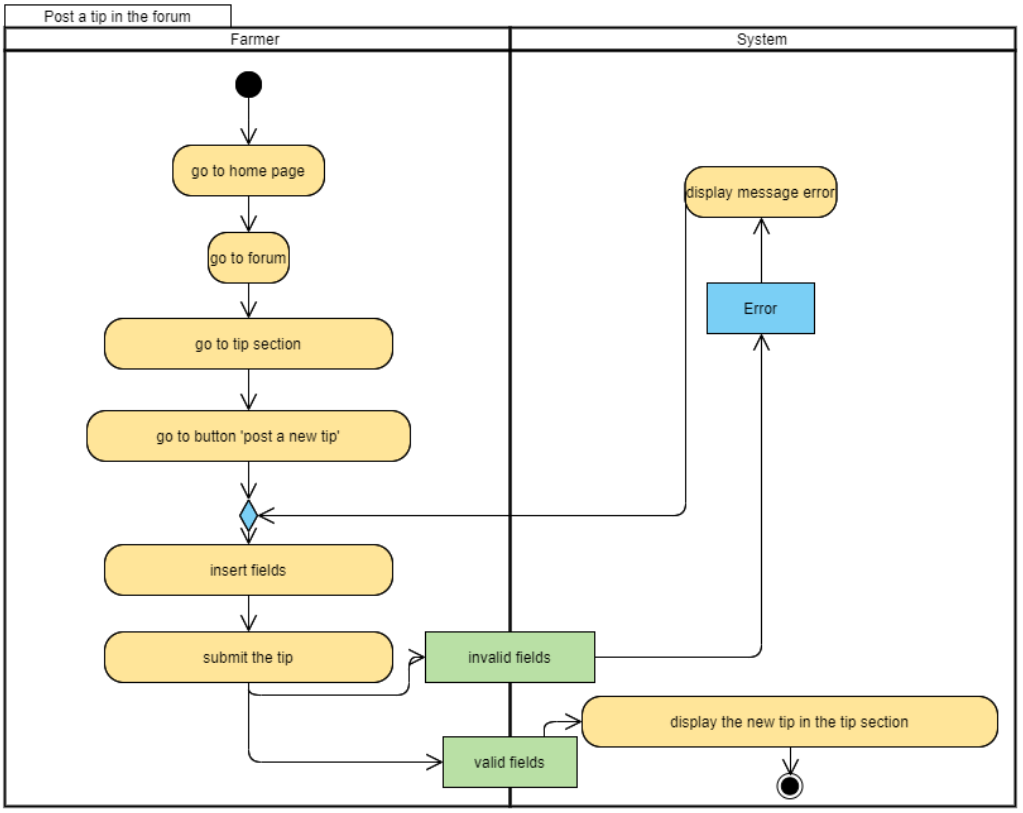


Name	Post an answer in the forum
Actor	Farmer

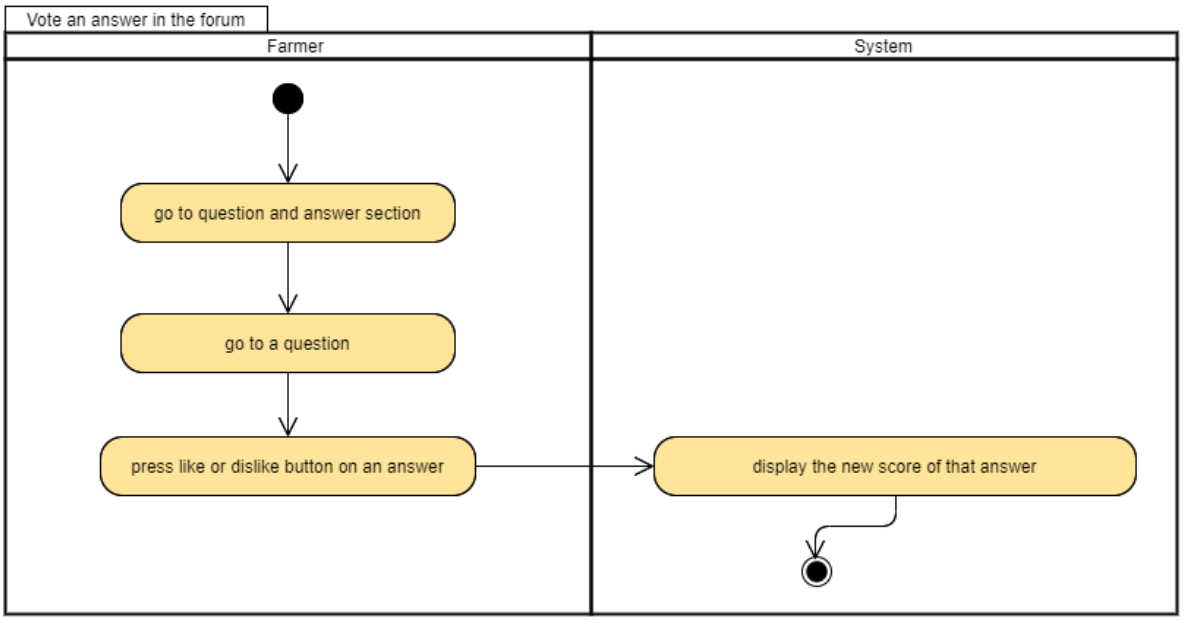
Entry conditions	The farmer is logged-in.
Event flow	1. From the home-page, the farmer presses on the “Forum” button. 2. The system displays a page divided in two sections: “Questions & Answers” and “Tips”. 3. In the “Questions & Answers” section, the system displays the posted questions. Next to each question, the system shows a button “Post an answer”. 4. The farmer presses on that button. 5. The system shows a blank field, an auto-filled field with the area of the farmer and a “Submit” button. 6. The farmer writes in the field the answer. 7. The farmer clicks on the “Submit” button.
Exit condition	The system displays the new answer at the right position (number of likes = 0) of the answers list, where answers are ordered from the most voted to the least voted,
Exceptions	The farmer clicks on the “Submit” without writing anything. The system displays an error message. The system invites the farmer to try again.



Name	Post a tip
Actor	Farmer
Entry conditions	The farmer is logged-in.
Event flow	1. From the home-page, the farmer presses on the “Forum” button 2. The system displays a page divided in two sections: “Questions & Answers” and “Tips”. 3. In the “Tips” section, the system displays the posted tips, ordered from the most recent to the least recent. At the top of the section, a button “Post a new tip” is displayed. 4. The farmer presses on that button. 5. The system shows a field “Category”, a blank field, an auto-filled field with the area of the farmer and a “Submit” button. 6. The farmer selects the category and writes in the field the tip they want to post. 7. The farmer clicks on the “Submit” button.
Exit condition	The system displays the new tip at the top of the tip section.
Exceptions	The farmer clicks on the “Submit” without selecting the category and/or without filling in the blank field. The system displays an error message. The system invites the farmer to try again.

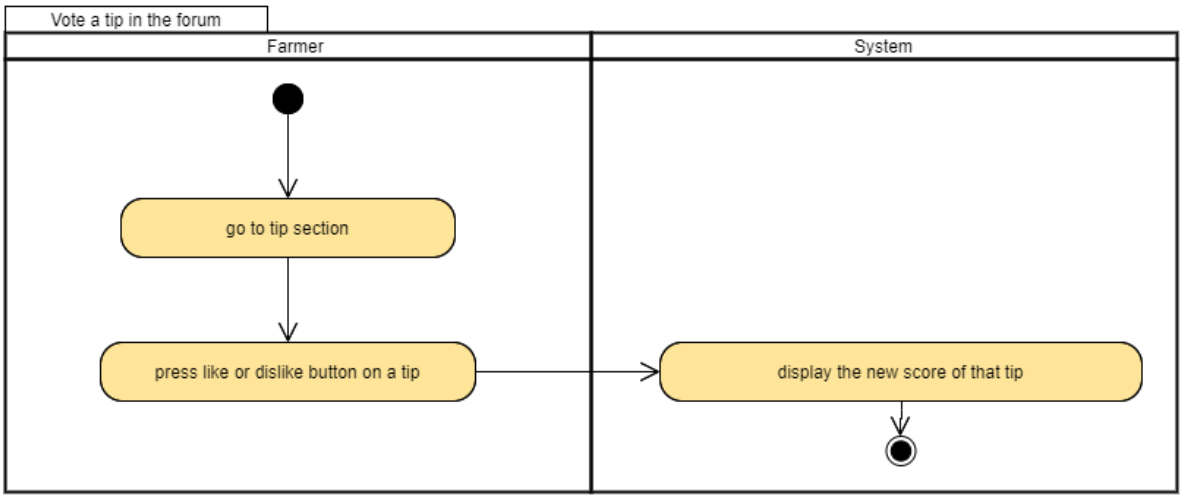


Name	Give an evaluation to an answer
Actor	Farmer
Entry conditions	The farmer is logged-in. The farmer entered the forum.
Event flow	1. In the “Questions & Answers” section, the system displays the posted questions and relative answers, ordered according to the score. Next to each answer, a like symbol and a dislike symbol are displayed together with the last vote expressed, if any. 2. The farmer chooses an answer and presses on the like symbol or on the dislike symbol.
Exit condition	The system shows an update score of that answer according to the action of the farmer.
Exceptions	None

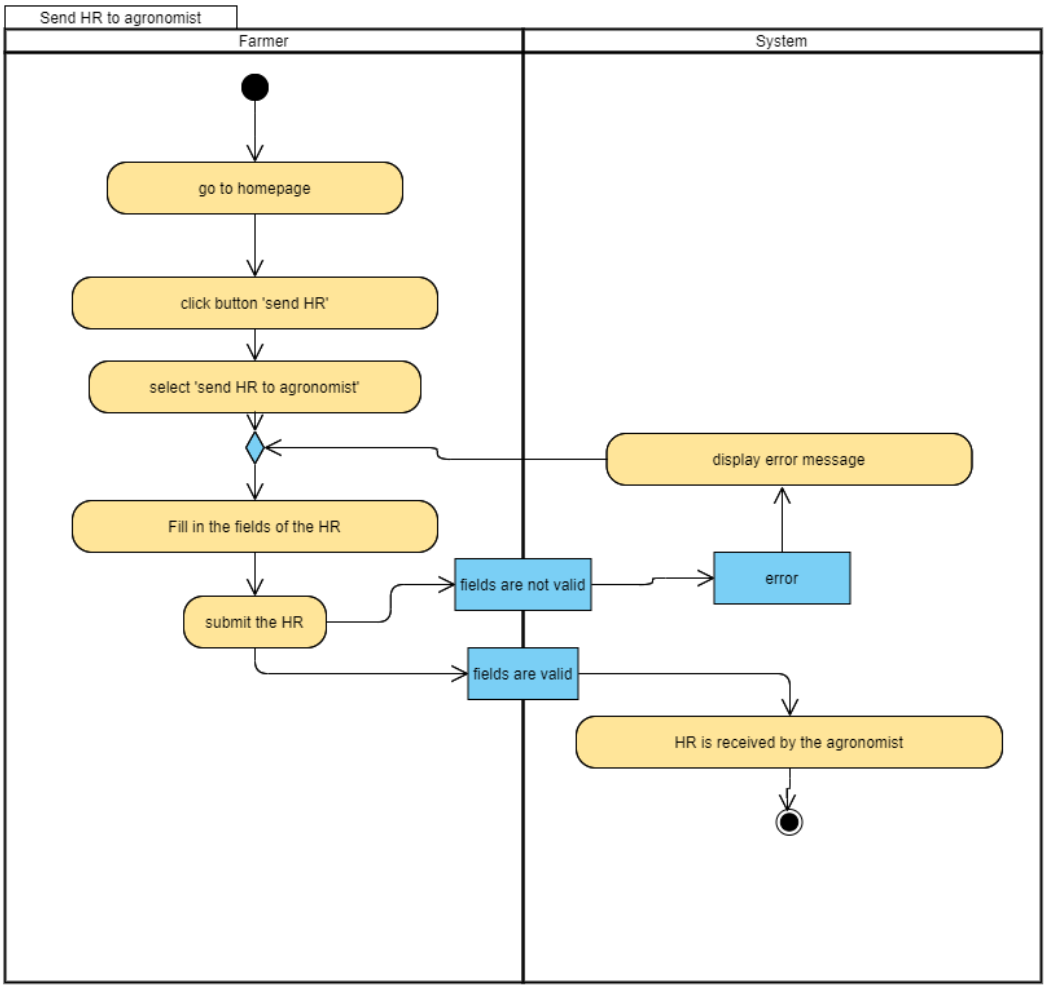


Name	Give an evaluation to a tip
Actor	Farmer
Entry conditions	The farmer is logged-in. The farmer entered the forum.
Event flow	1. In the “Tips” section, the system displays the posted tips, ordered according to the date (from the most recent to least recent). Next to each tip, a like symbol and a dislike symbol are displayed, together with the last vote expressed, if any. 2. The farmer presses on the like symbol or on the dislike

	symbol.
Exit condition	The system shows the updated score of that tip according to the action of the farmer.
Exceptions	None

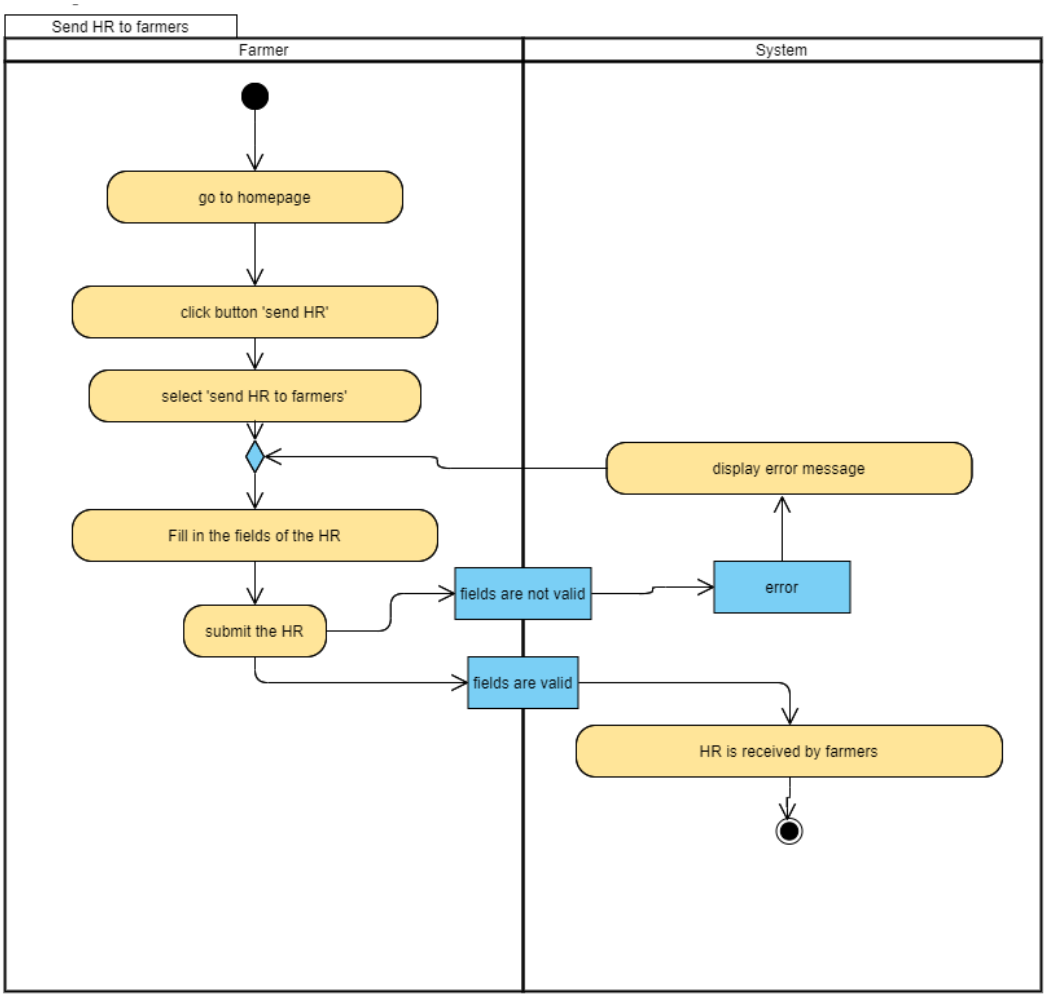


Name	Send an HR to an agronomist
Actor	Farmer
Entry conditions	The Farmer is logged-in.
Event flow	1. From the home-page, the farmer presses on the “Send HR” button 2. The system displays two buttons: “Send HR to agronomist” and “Send HR to farmers”. 3. The farmer chooses the button “Send HR to agronomist”. 4. The system shows: a. an autocomplete field called “Agronomist” containing the name of the agronomist in charge of the farmer’s area; b. a field “Category” with a dropdown menu; c. a blankfield. 5. The farmer chooses the right category from the menu and fills in the blank field with the request. 6. The farmer clicks-on the “Submit” button.
Exit condition	The system forwards it to the agronomist of the farmer’s area.
Exceptions	1. The farmer clicks-on the “Submit” button without selecting the Category. The system displays an error message. 2. The farmer clicks-on the “Submit” without writing anything. The system displays an error message.



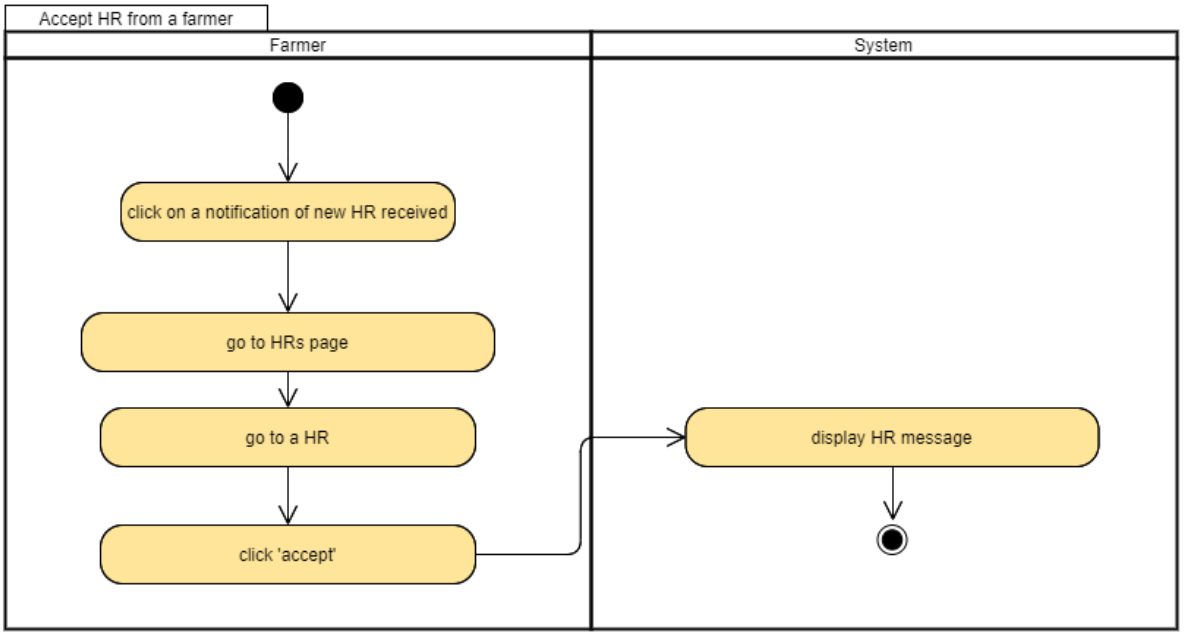
Name	Send an HR to farmers
Actor	Farmer
Entry conditions	The farmer is logged-in.
Event flow	1. From the home-page, the farmer presses on the “Send HR” button. 2. The system displays two buttons: “Send HR to agronomist” and “Send HR to farmers”. 3. The farmer chooses the button “Send HR to farmers”. 4. The system shows: a. an autocomplete field called “Farmers” containing the names of the farmers whose distance is less than 10km from the sender; b. a blank field. 5. The farmer fills-in the blank field with the request. 6. The farmer clicks-on the “Submit” button.
Exit condition	The system forwards it to the farmers within 10km. The system sets the HRs to the “pending state”.
Exceptions	1. The farmer clicks on the “Submit” without writing

	anything. The system displays an error message. 2. There are no farmers within 10km. In this case the system displays the message “No farmers close to you. You cannot send a HR to any farmer.”

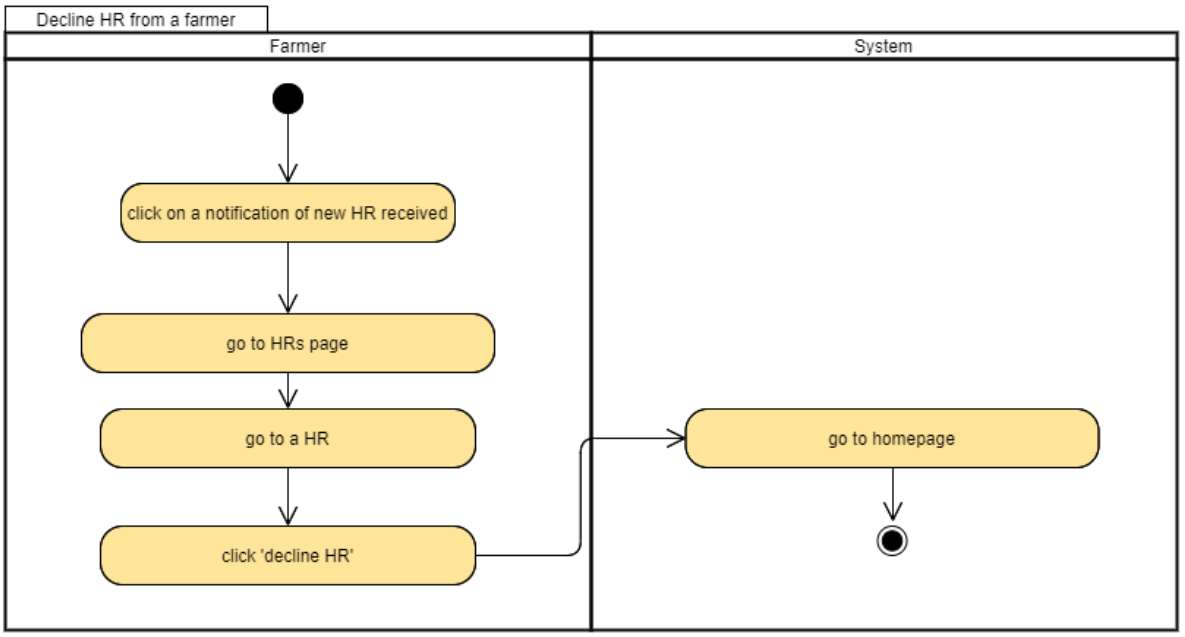


Name	Accept an HR from a farmer
Actor	Farmer
Entry conditions	The farmer is logged-in. The farmer receives a notification.
Event flow	1. From his device, the farmer clicks-on the notification about a new HR received. 2. The system redirects the Farmer directly to a page showing the HR he received, that includes the name of the sender and the text request. 3. Next to a HR, the system displays two buttons: “Accept HR” and “Decline HR”. 4. The farmer chooses the button “Accept HR”.
Exit condition	The system creates and displays the HR-message.

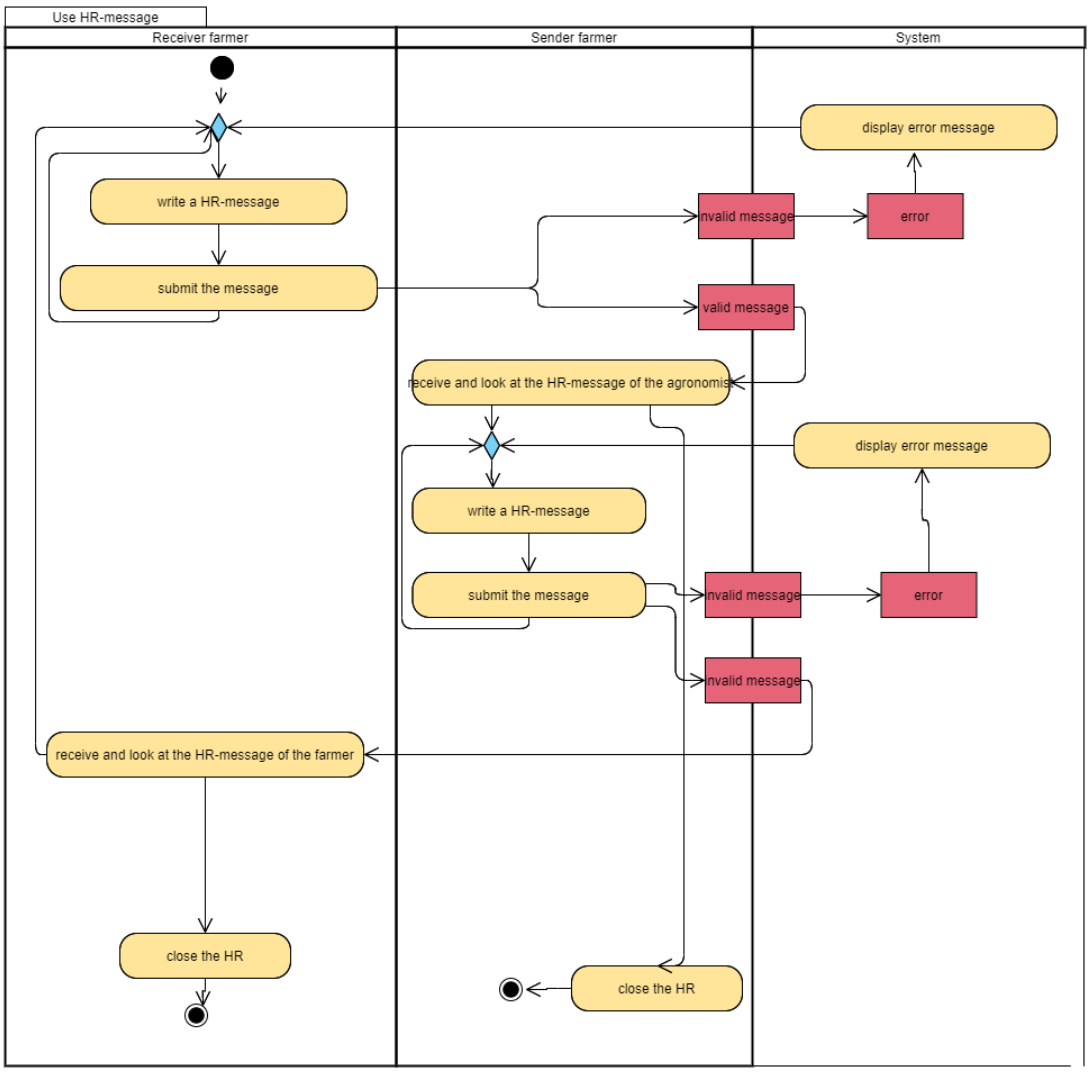
Exceptions	None



Name	Decline an HR from a Farmer
Actor	Farmer
Entry conditions	The farmer is logged-in The farmer receives a HR notification.
Event flow	1. From his device, the farmer clicks-on the notification about a new HR received. 2. The system redirects the farmer directly to a page showing the HR he received, that includes the name of the sender and the text request. 3. Next to a HR, the system displays two buttons: “Accept HR” and “Decline HR”. 4. The farmer chooses the button “Decline HR”.
Exit condition	The system redirects the farmer to the home page of the application.
Exceptions	None

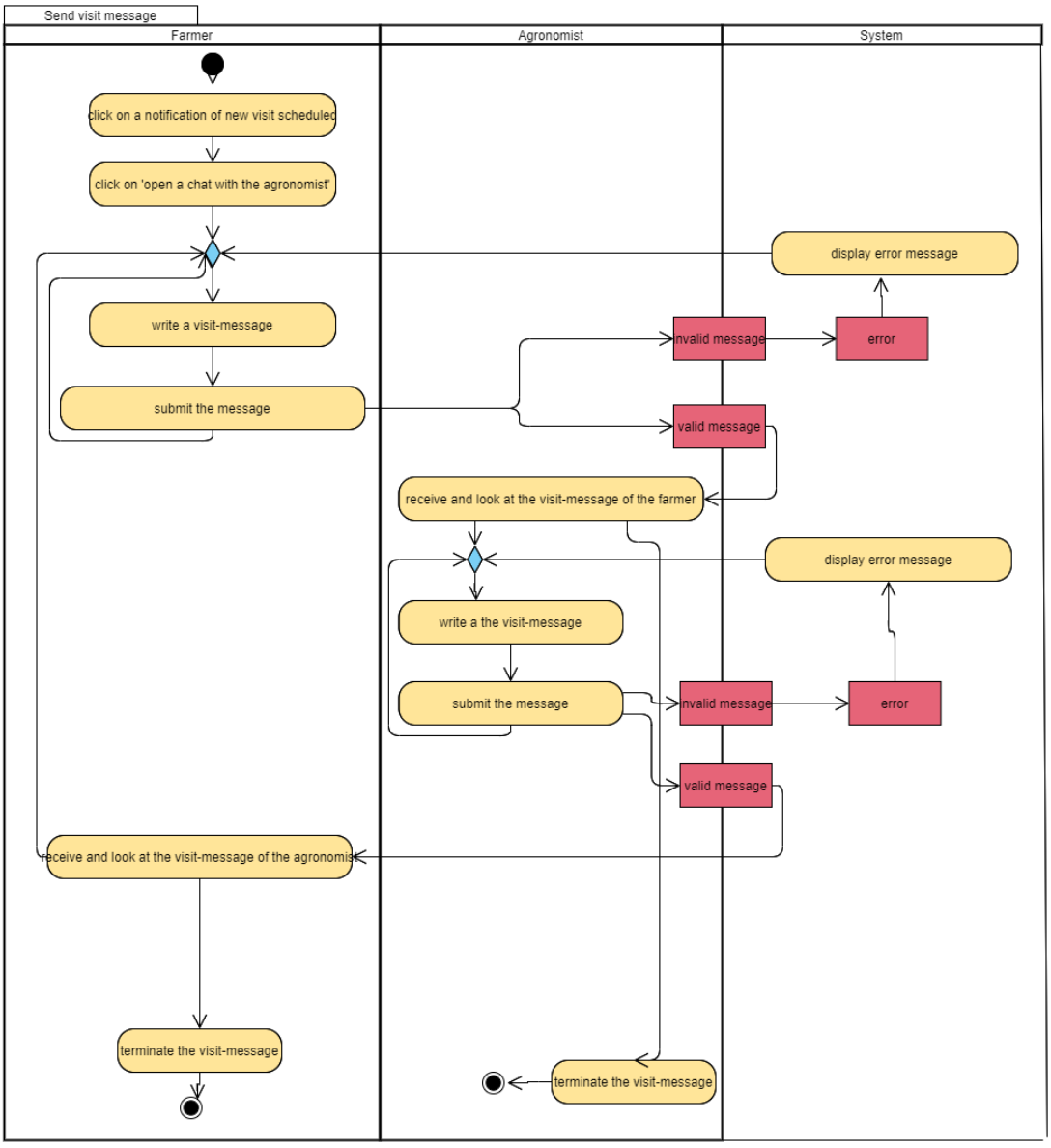


Name	Use HR-messages by farmers
Actor	Farmer
Entry conditions	A farmer (receiver of the HR) accepted a HR from another farmer (sender of the HR). Farmers are logged in.
Event flow	5. The receiver farmer writes one or possibly many HR-messages and submits it. 6. The sender farmer looks at the HR-message(s) received by the receiver farmer. 7. The sender farmer writes one or possibly many HR-messages and submits it. 8. The receiver farmer looks at the HR-message(s) received by the sender farmer. 9. The receiver farmer writes one or possibly many HR-messages and submits it. 10. Go to point 2.
Exit condition	One of the two farmers closes the HR.
Exceptions	1. A farmer clicks on the “Submit” button without writing a HR-message. The system displays an error message.

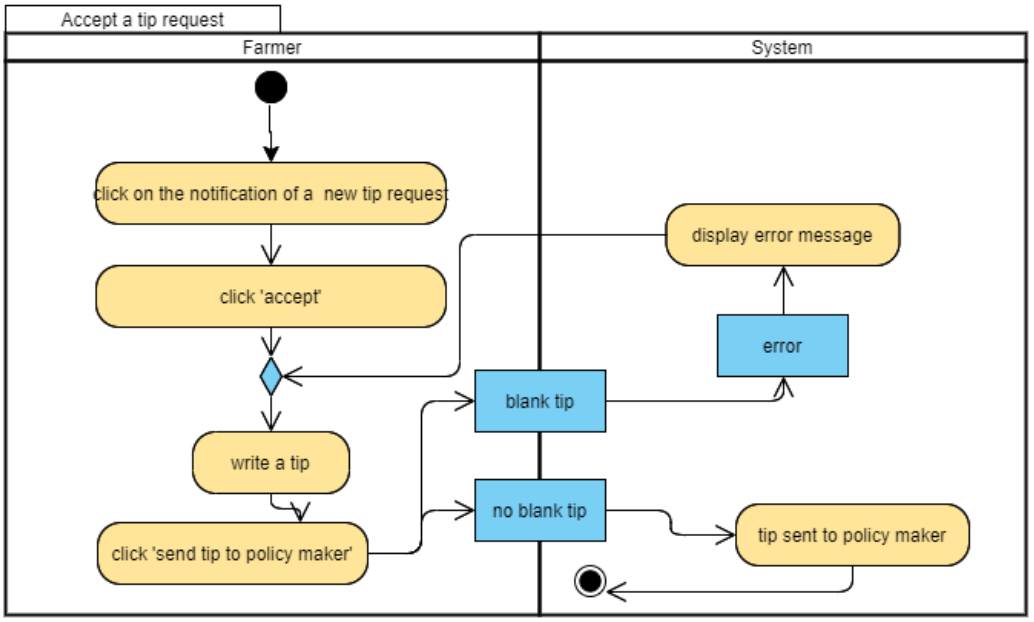


Name	Send a visit-message to modify an upcoming visit
Actor	Farmer Agronomist
Entry conditions	The farmer has received a notification about a new scheduled visit from the agronomist. The farmer wants to change the date of the visit. The farmer is logged in. The agronomist is logged in.
Event flow	1. The farmer clicks-on the notification about a new visit scheduled. 2. The system redirects the farmer directly to a page showing the button “Open a chat with the agronomist”. 3. The farmer clicks-on that button. 4. The farmer writes one or possibly many visit-messages and submits it. 5. The agronomist looks at the visit-message(s) received by the farmer. 6. The agronomist writes one or possibly many

	visit-messages and submits it. 7. The farmer looks at the visit-message(s) received by the agronomist. 8. The farmer writes one or possibly many visit-messages and submits it. 9. Go to point 5.
Exit condition	One between the farmer or the agronomist terminates the visit-messages.
Exceptions	The farmer or the agronomist clicks-on the “Submit” button without writing anything. The system displays an error message. The farmer/agronomist is invited to try again.

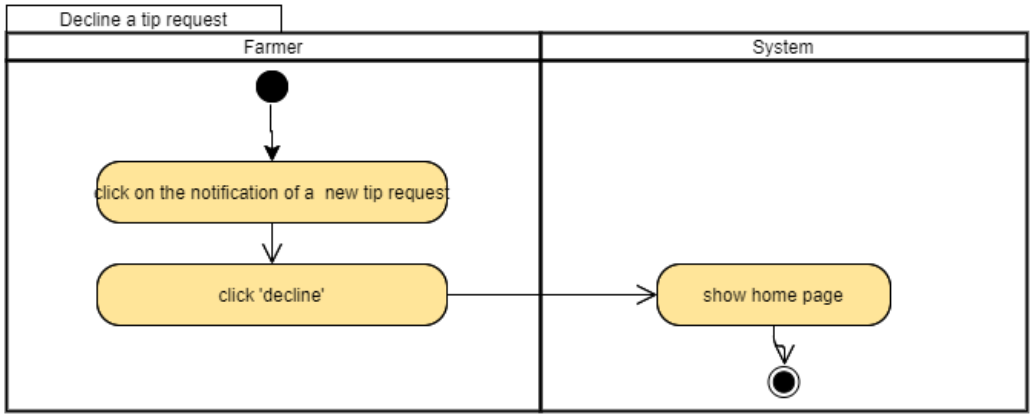


Name	Accept a tip request
Actor	Farmer
Entry conditions	The farmer is logged-in. The farmer received a tip request from the policy maker.
Event flow	1. The farmer clicks-on the notification about a tip request. 2. The system displays the request to write a tip and two buttons “Accept” and “Decline”. 3. The farmer clicks on the button “Accept”. 4. The system displays a message “Write your tip” and a blank field. 5. The farmer writes the tip in the blank field. 6. The farmer clicks on the button “Send tip to policy maker”.
Exit condition	The system sends the tip to the Policy maker.
Exceptions	The Farmer or the agronomist clicks-on the “Submit” button without writing anything. The system displays an error message. The farmer/agronomist is invited to try again.



Name	Decline a tip request
Actor	Farmer
Entry conditions	The farmer is logged-in The farmer received a tip request from the policy maker of his district.
Event flow	1. The farmer clicks-on the notification about a tip request. 2. The system displays the request to write a tip and two buttons “Accept” and “Decline”.

	3. The farmer clicks on the button “Decline”.
Exit condition	The system displays the home page.
Exceptions	None



3.2.2 Agronomist Scenarios

- **Scenario 1: Sign up and visualize relevant information**

  Jenny is the agronomist in charge of an area in the district of Jangaon. She has just become aware of a brand-new web application called DREAM that could help her in her job. The government told her to sign up through the agronomist sign up page with the authorization code that they gave her. In fact, her profile was already set-up by the government and it just needs to be activated. So, after opening the DREAM web application, Jenny inserts first and last name, and the authorization code. Now Jenny’s profile is active and she can set her new credentials, which she will use to login the next time she opens the web application. Once she has done so, the application displays the home page, with relevant information about the local weather and the performance of the farmers that work in that area.

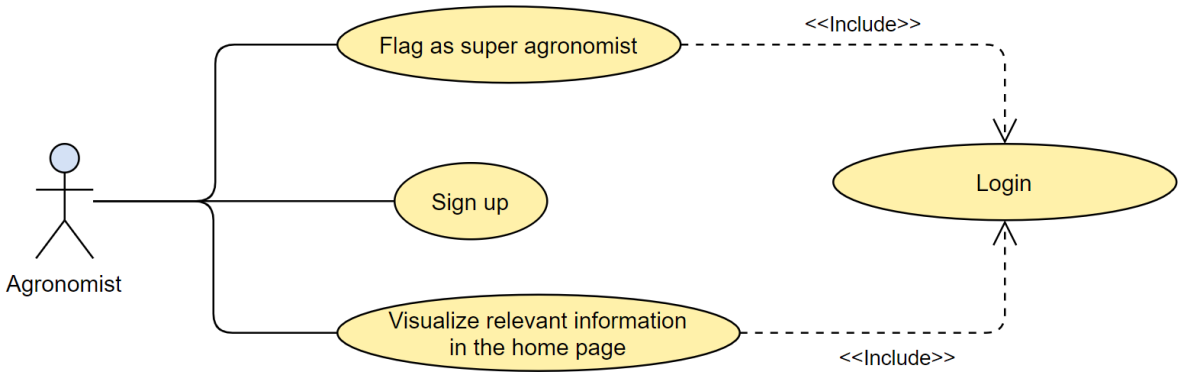
- **Scenario 2: Fill in the visit report**

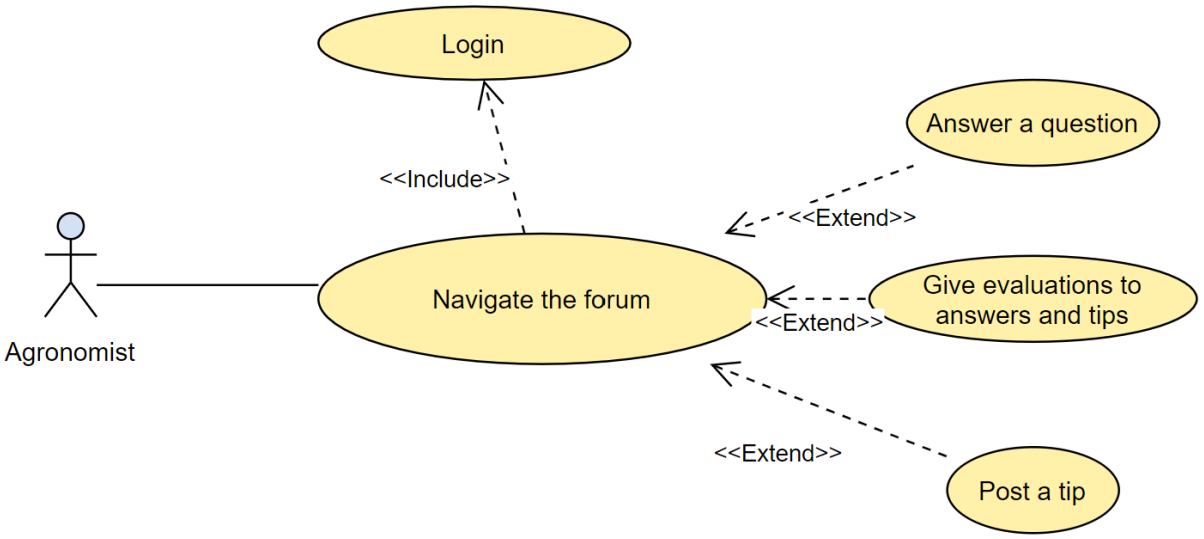
  The agronomist Najee has just visited one of the farmers of his area. He picks up his smartphone, goes into the browser, logs in into DREAM and, from the visit plan section, marks the visit he has just attended as “done”. Then, he goes home, turns on his laptop and, after the usual log-in into DREAM, he starts to fill-in the visit report. He starts from the basic section, which is mandatory and contains the farmer’s assessment. Then he goes on with the optional problem section: he remembers the farmer told him about the lack of water in the irrigation system for three full days, and he notes it down. Finally, the last section is about the weather: nothing relevant happened this time, therefore he leaves it blank. The report is now completed and Najee can submit it.

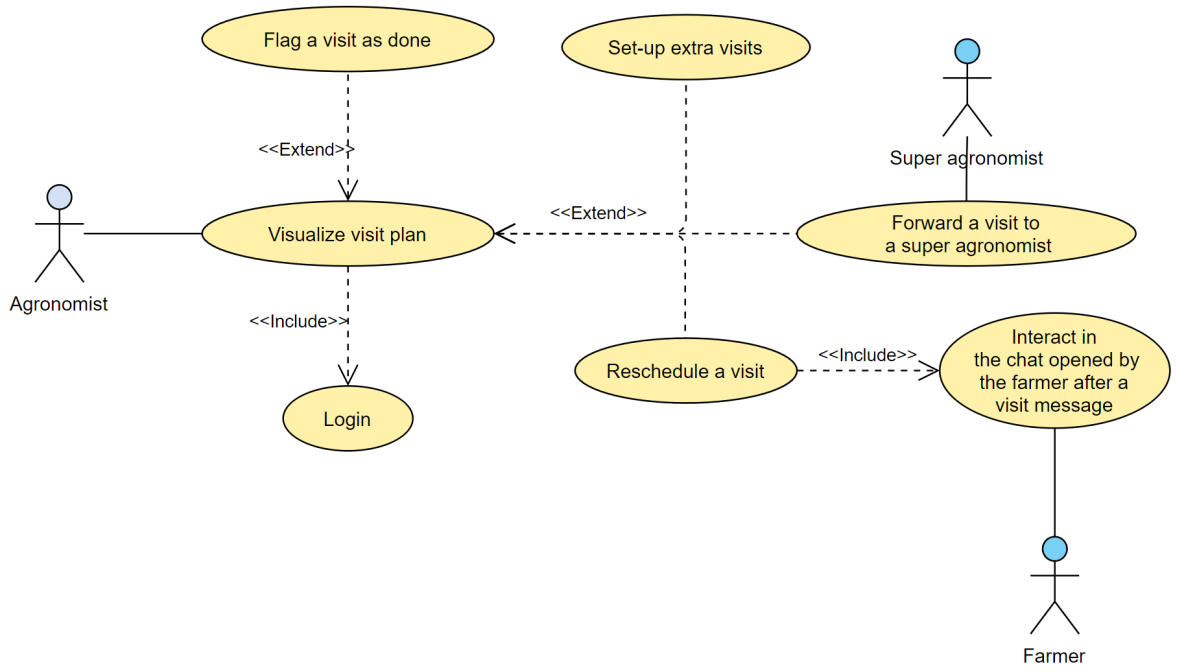
- **Scenario 3: Dealing with super-agronomist delegation**

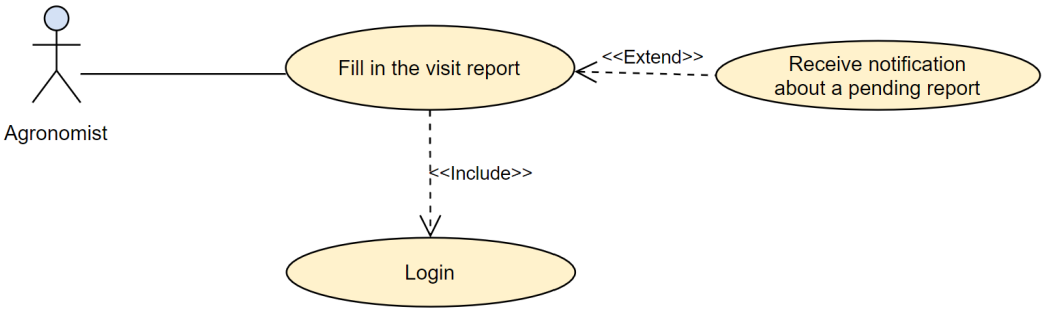
  Mark is an agronomist who decided to help his colleagues in the district: in fact, some time ago he logged into the DREAM web application and, in the settings section, he flagged himself as “super-agronomist”. Right this morning, he received a notification about a delegation request for a visit from another agronomist. However, he really couldn’t make the visit and was forced to refuse it, through the application. Now, he has just received another delegation request: this time, he accepts it and the visit now appears in his visit plan.

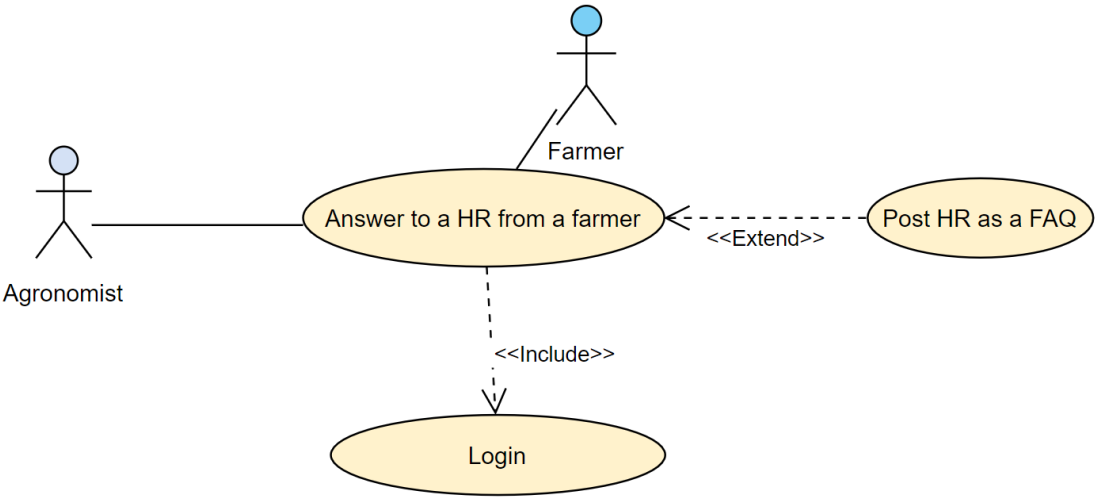
3.2.2.1. Use case diagrams

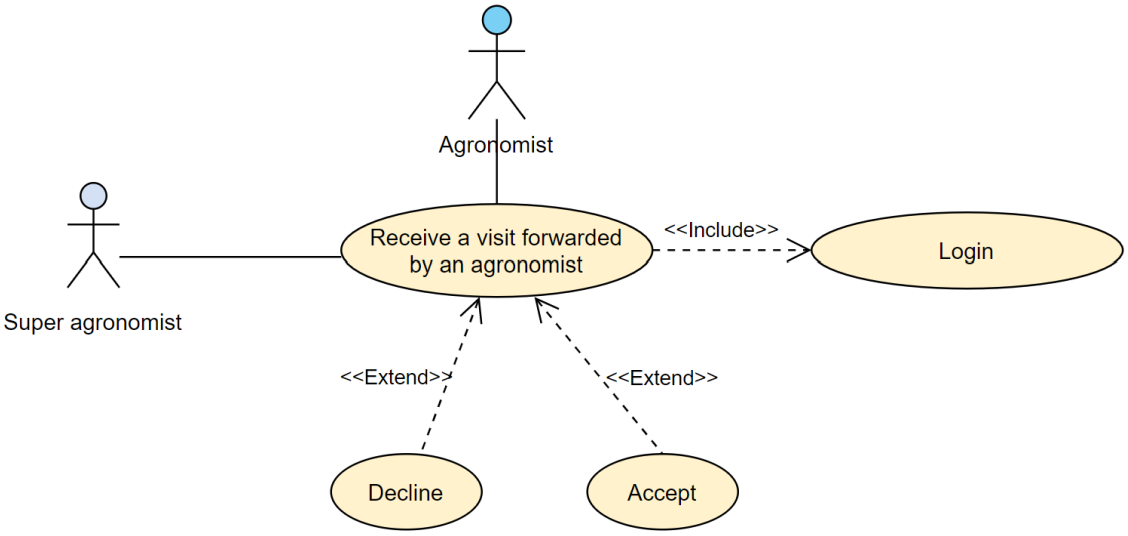






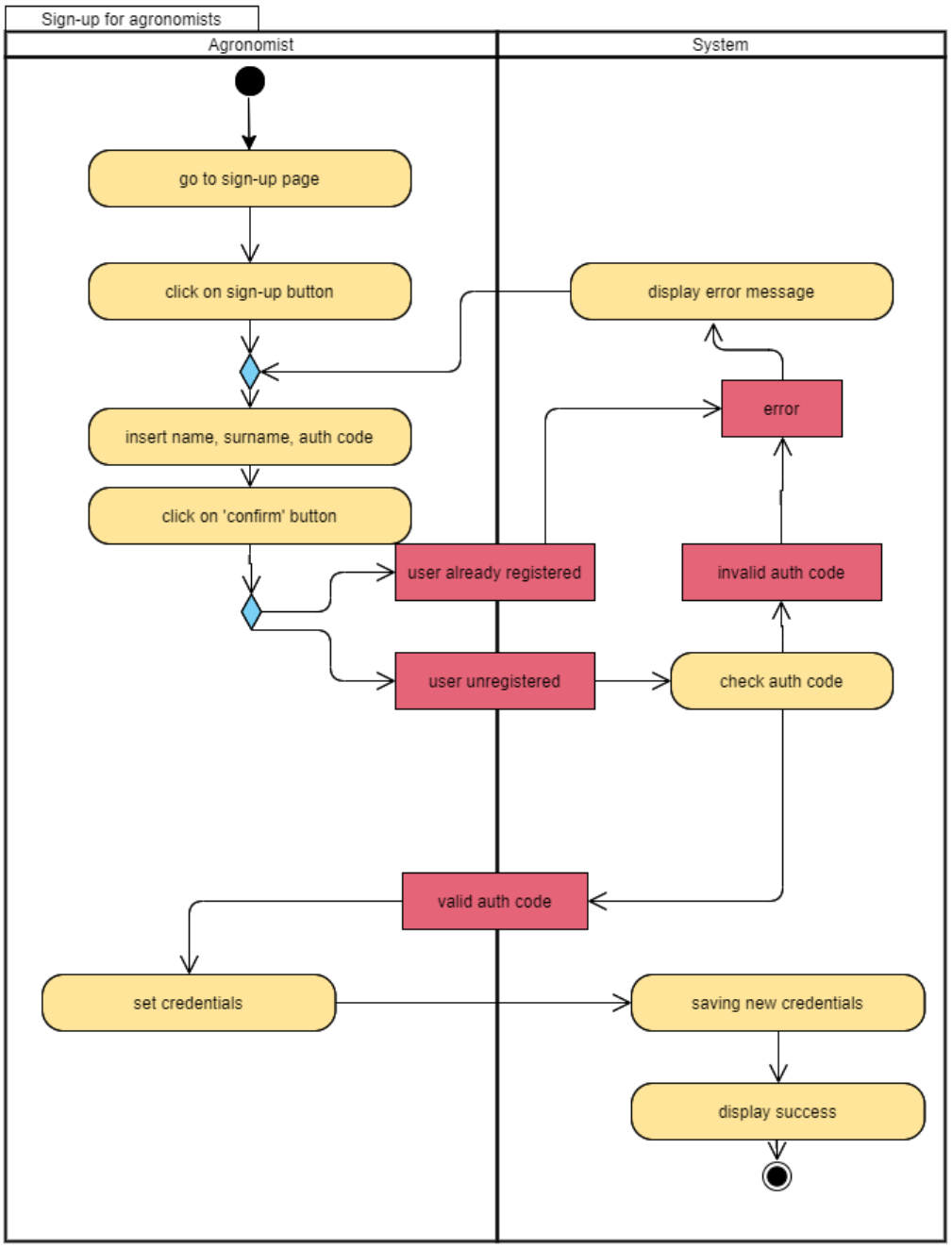






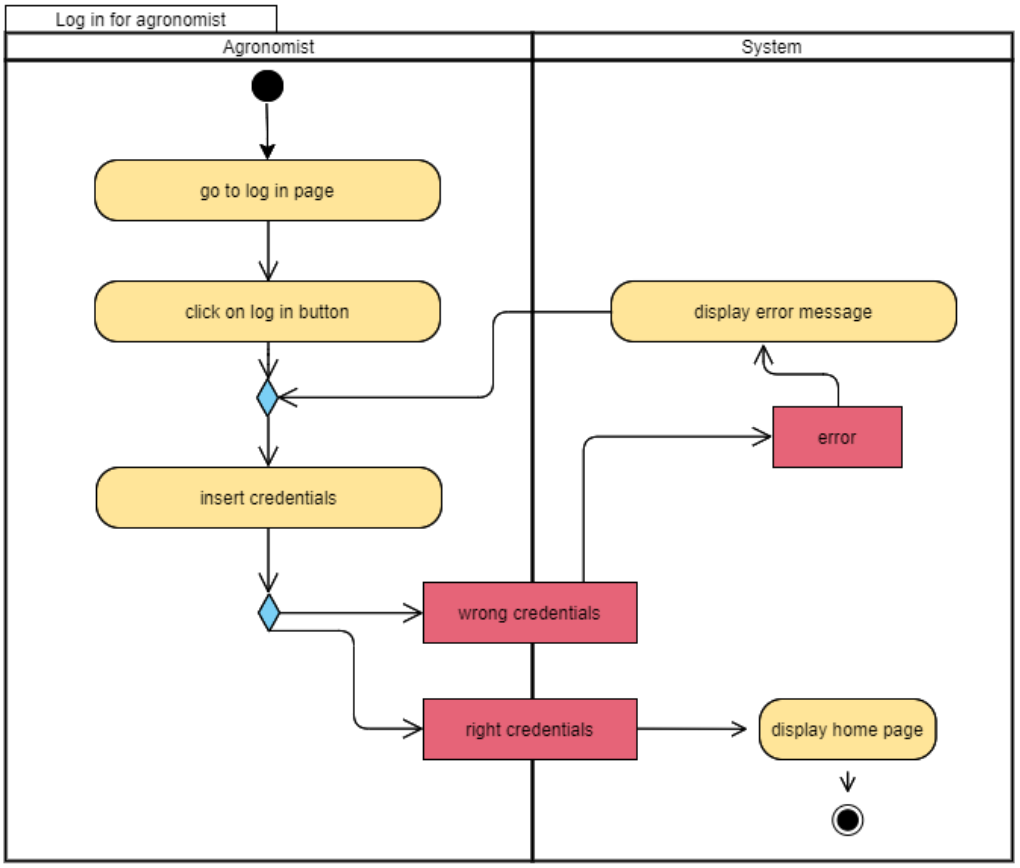
3.2.2.2. Use case tables and activity diagrams

Name	Sign-up agronomist
Actor	Agronomist
Entry conditions	The agronomist opened the DREAM web application. The agronomist has an authorization code.
Event flow	1. The application shows the Sign-up page 2. The agronomist clicks on “Sign-up” 3. The application displays the following mandatory fields: first name and last name, authorization code. 4. The agronomist fills-in the mandatory fields. 5. The agronomist clicks on the “Confirm” button. 6. The app allows the agronomist to set the credentials. 7. The agronomist writes the credentials. 8. The agronomist clicks on ‘confirm’.
Exit condition	The agronomist registration is successfully completed and the agronomist’s data are saved into the system. The app shows the screen with the confirmation “You have successfully registered!”
Exceptions	1. The agronomist inserts a wrong authorization code. The application displays an error message and the agronomist is invited to try again. 2. The agronomist is already registered. The application shows the message “You have already registered! Log-in with your credentials”. The agronomist is invited to try again.



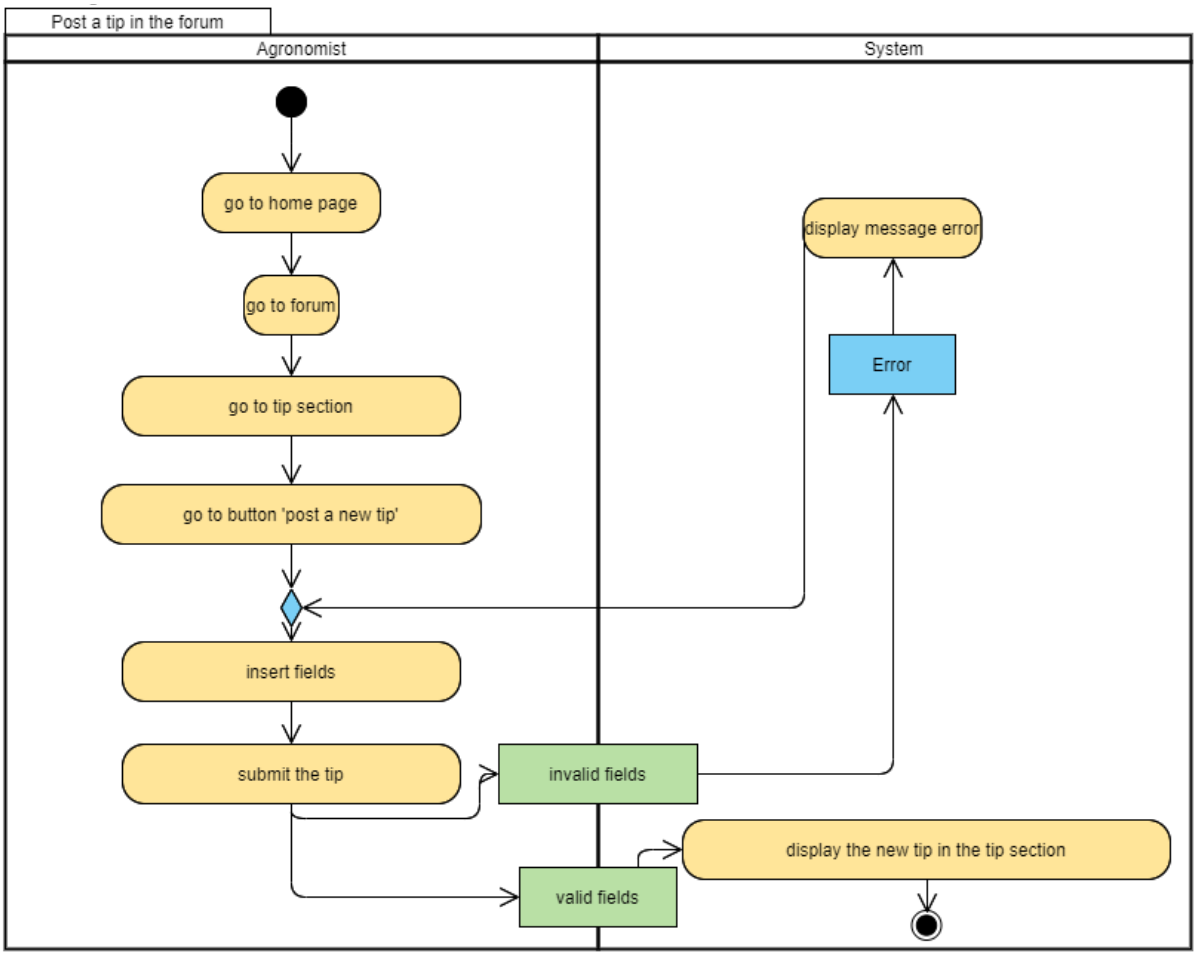
Name	Log-in agronomist
Actor	Agronomist
Entry conditions	The agronomist opened the DREAM web application. The agronomist is registered into the system.
Event flow	1. The system displays the log-in page. 2. The agronomist inserts the credentials into the mandatory fields. 3. The agronomist clicks on the “Log-in” button. 4. The system checks for the validity of the credentials.

	5. The system shows the home page of the application
Exit condition	The agronomist is successfully logged-in.
Exceptions	1. The agronomist inserts the wrong credentials. The system displays an error message and invites the agronomist to try again.



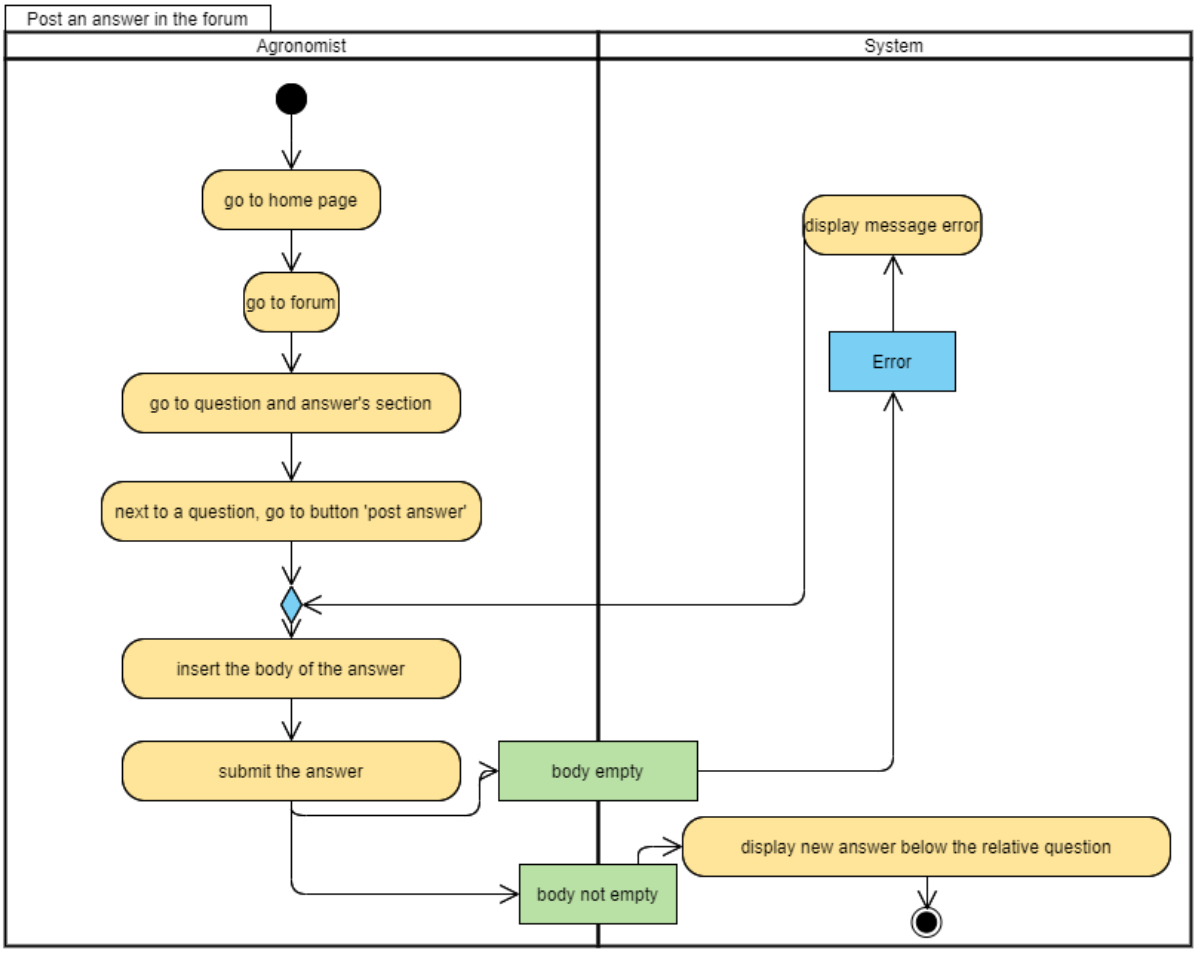
Name	Post a tip
Actor	Agronomist
Entry conditions	The agronomist is logged-in.
Event flow	1. From the home-page, the agronomist presses on the “Forum” button 2. The system displays a page divided in two sections: “Questions & Answers” and “Tips”. 3. In the “Tips” section, the system displays the posted tips, ordered from the most recent to the least recent. At the top of the section, a button “Post a new tip” is displayed. 4. The agronomist presses on that button. 5. The system shows a field “Category”, a blank field, an auto-filled field with the area of the agronomist and a “Submit” button. 6. The Agronomist writes in the field the answer they want

	to post. 7. The Agronomist clicks on the “Submit” button.
Exit condition	The system displays the new answer at the top of the tip section.
Exceptions	The Agronomist clicks on the “Submit” without selecting the category or filling in the blank field. The system displays an error message. The system invites the agronomist to try again.



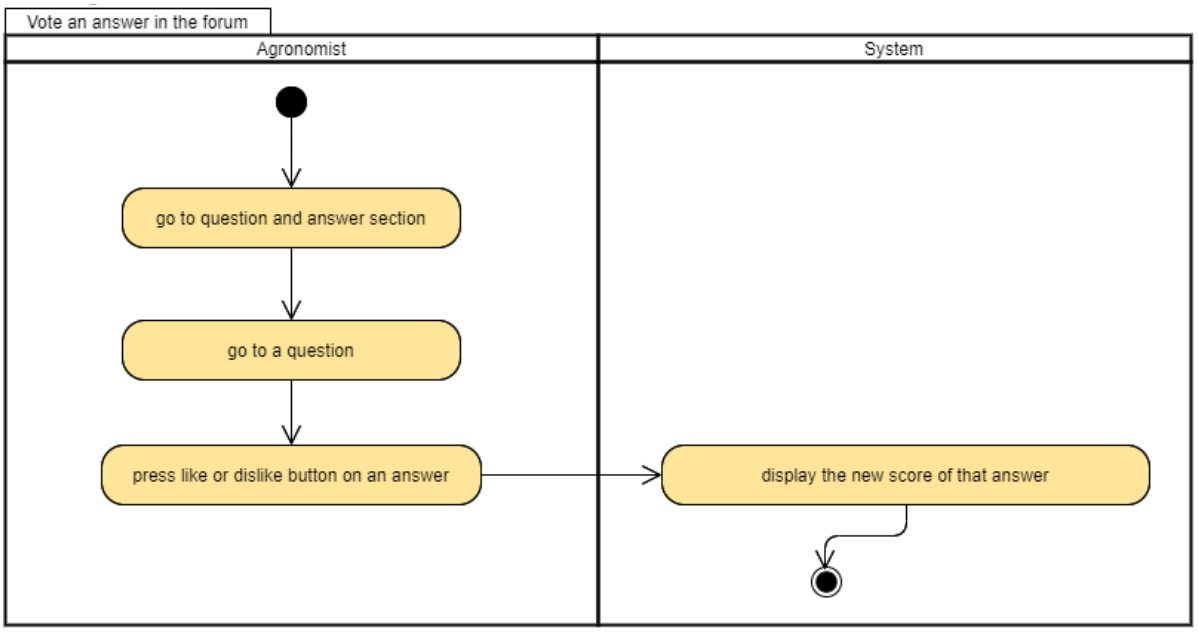
Name	Post an answer in the forum
Actor	Agronomist
Entry conditions	The agronomist is logged-in.
Event flow	1. From the home-page, the agronomist presses on the “Forum” button 2. The system displays a page divided in two sections: “Questions & Answers” and “Tips”. 3. In the “Questions & Answers” section, the system

	displays the posted questions. Next to each question, the system shows a button “Post an answer”. 4. The agronomist presses on that button. 5. The system shows a blank field, an auto-filled field with the area of the Agronomist and a “Submit” button. 6. The agronomist writes in the field the answer they want to post. 7. The agronomist clicks on the “Submit” button.
Exit condition	The system displays the new answer at the bottom of the answers list.
Exceptions	The Agronomist clicks on the “Submit” without writing anything. The system displays an error message. The system invites the agronomist to try again.

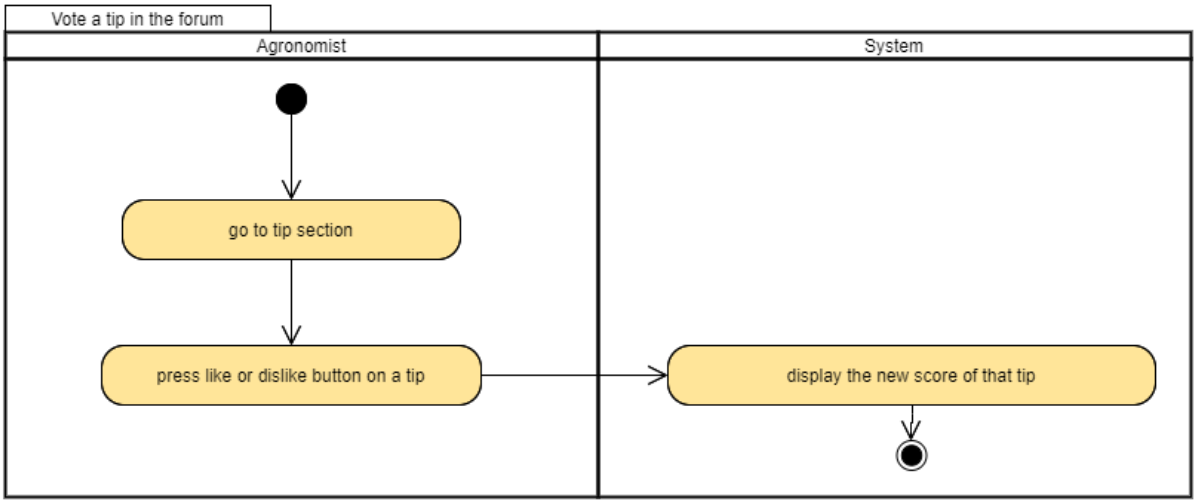


Name	Give an evaluation to an answer
Actor	Agronomist
Entry conditions	The agronomist is logged-in. The agronomist entered the forum.
Event flow	1. In the “Questions & Answers” section, the system

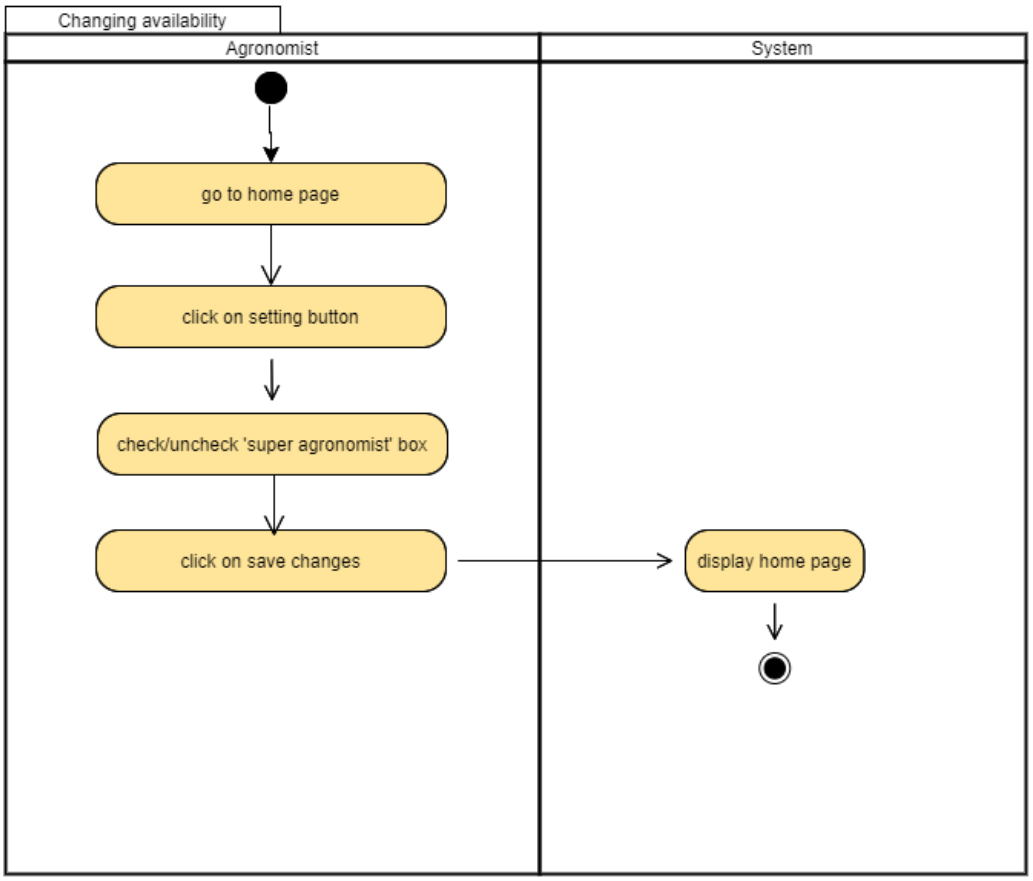
	displays the posted questions and relative answers, ordered according to the score. Next to each answer, a like symbol and a dislike symbol are displayed together with the last vote expressed, if any. 2. The agronomist chooses an answer and presses on the like symbol or on the dislike symbol.
Exit condition	The system shows an update score of that answer according to the action of the agronomist.
Exceptions	None



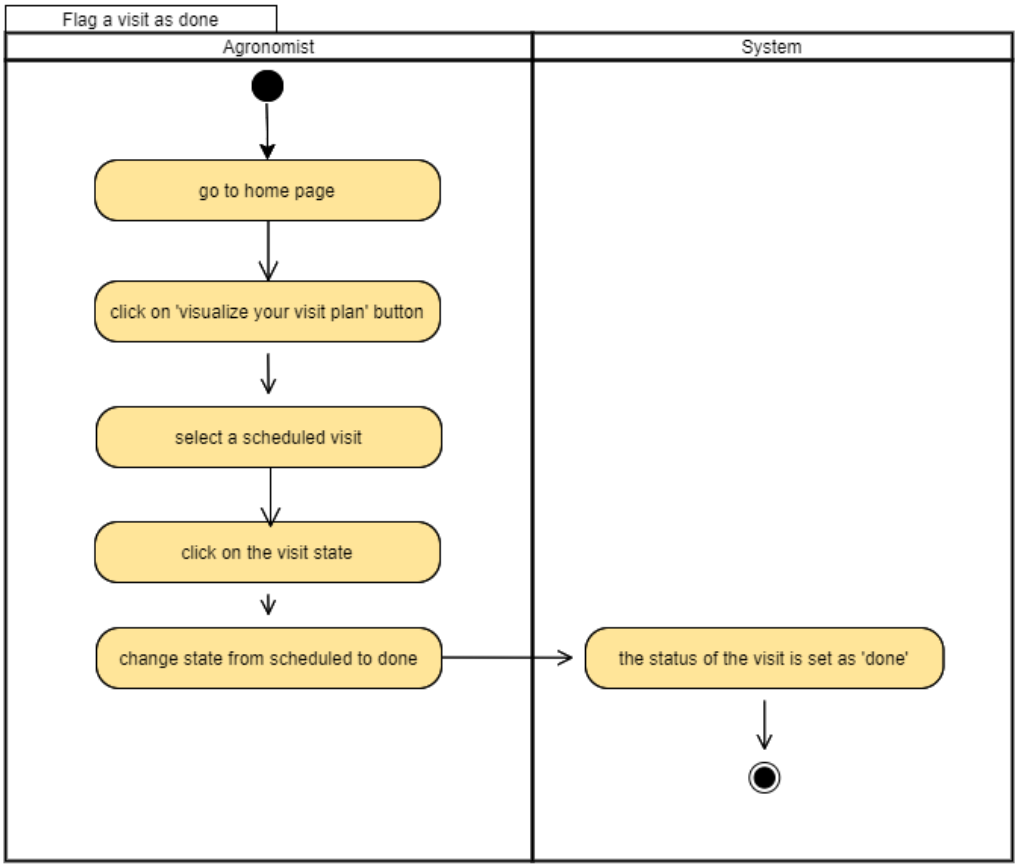
Name	Give an evaluation to a tip
Actor	Agronomist
Entry conditions	The agronomist is logged-in. The agronomist entered the Forum.
Event flow	1. In the “Tips” section, the system displays the posted tips, ordered according to the date (from the most recent to least recent). Next to each tip, a like symbol and a dislike symbol are displayed, together with the last vote expressed, if any. 2. The agronomist presses on the like symbol or on the dislike symbol.
Exit condition	The system shows the updated score of that tip according to the action of the agronomist.
Exceptions	None



Name	Changing availability as super agronomist
Actor	Agronomist
Entry conditions	The agronomist is logged in.
Event flow	1. In the homepage, the system displays a button “Settings”. 2. The agronomist clicks on the button. 3. The system displays several settings, one of these is “Super agronomist” and a check box next to it. 4. The agronomist (un)checks that box. 5. At the bottom of the page, there is a button “Save changes”. 6. The agronomist presses on that button. 7. The system redirects the agronomist to the home page.
Exit condition	The profile of the agronomist is now updated as super agronomist profile or brought back to a standard agronomist profile.
Exceptions	None

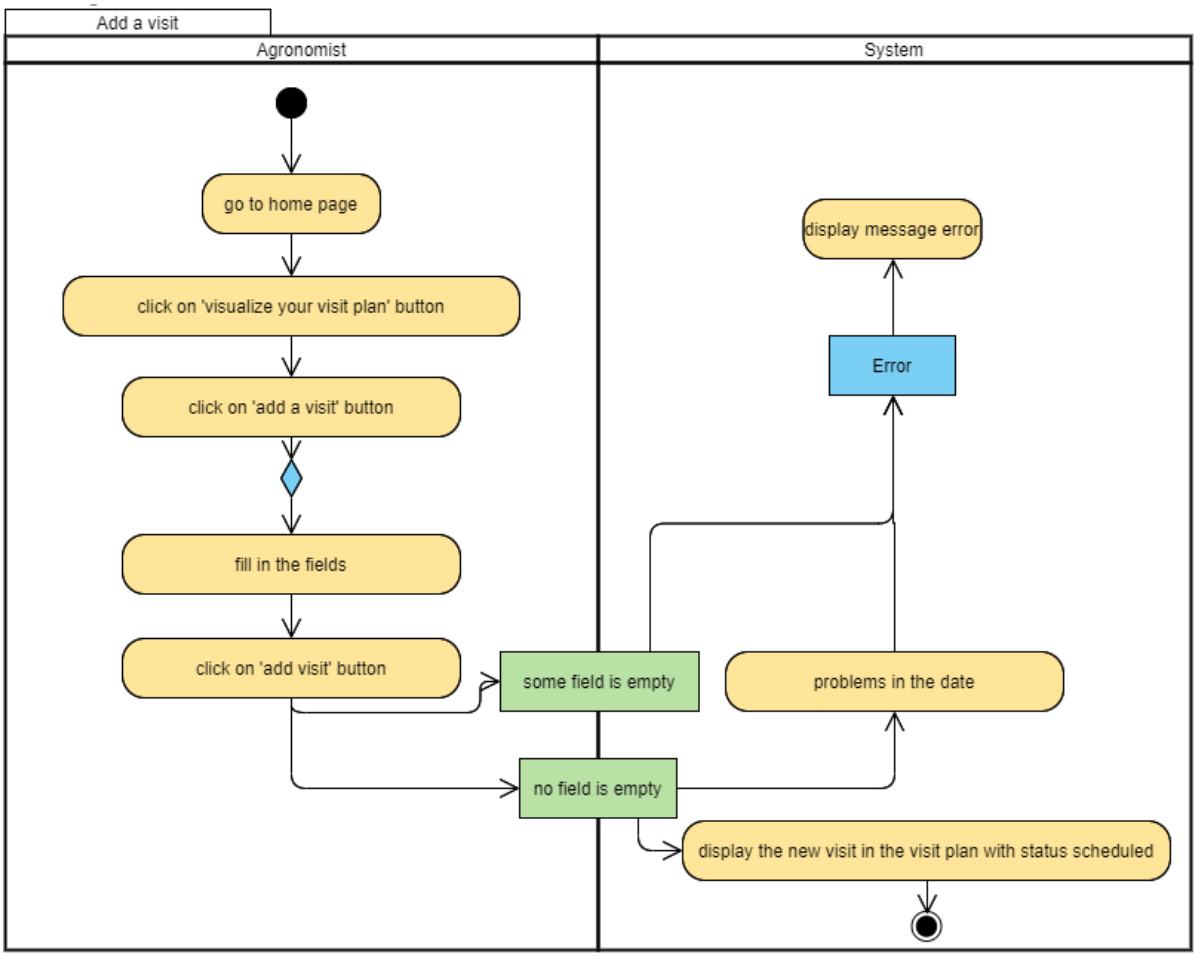


Name	Flag a visit as done
Actor	Agronomist
Entry conditions	The agronomist has visited a farmer. The agronomist is logged in.
Event flow	1. In the home page, the system displays a “Visualize your visit plan” button. 2. The agronomist presses on that button. 3. The system displays a calendar: for each day, a list of visits of that day is displayed, including the status of each visit. 4. The agronomist locates a done visit: the status of the visit is currently set to “scheduled”. 5. The agronomist clicks-on the status of the visit. 6. The agronomist changes the status from scheduled to done.
Exit condition	The system displays the status “done” next to the visit at issue. The button “Fill in visit report” is now displayed next to the visit.
Exceptions	None



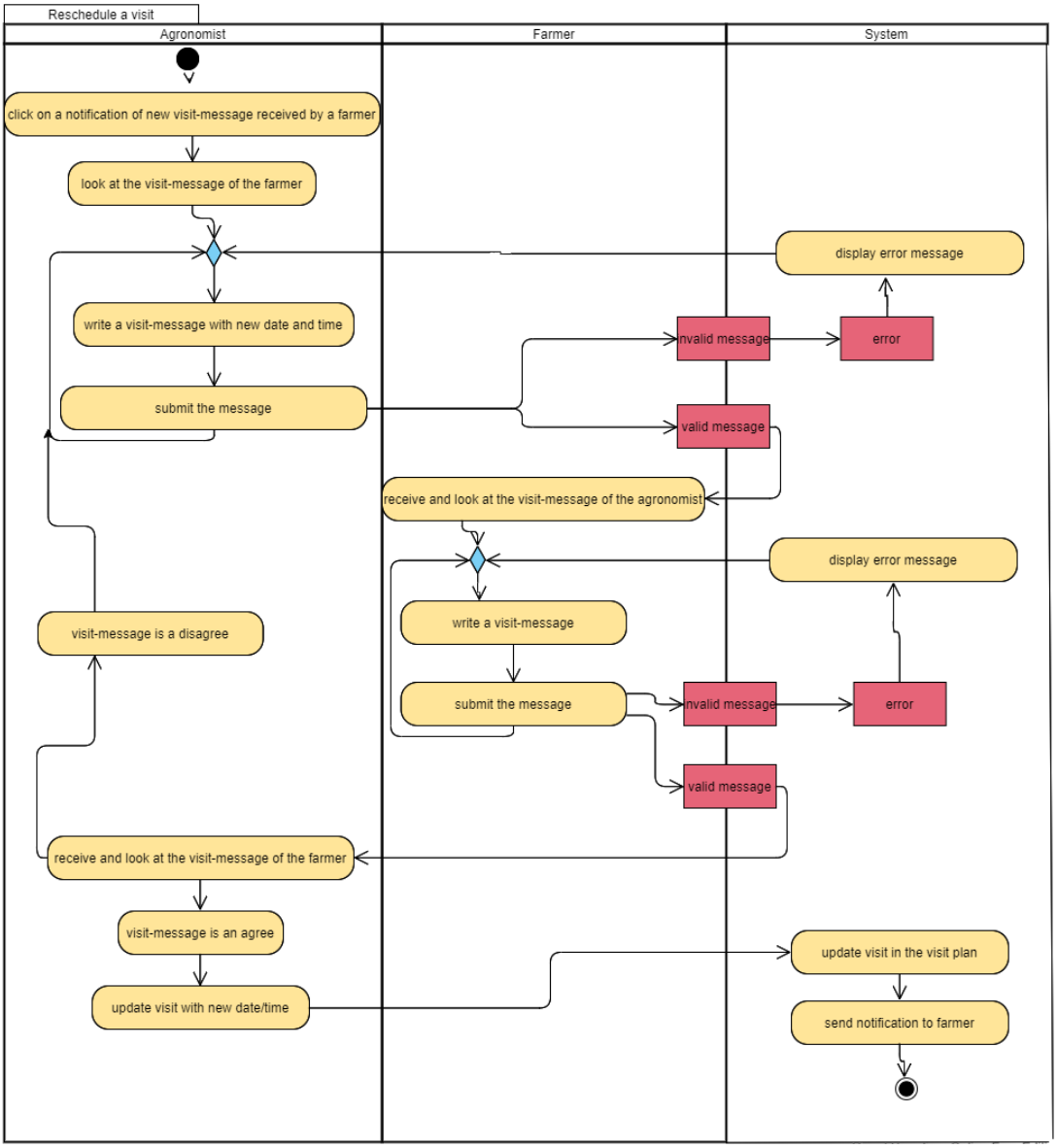
Name	Set-up extra visits
Actor	Agronomist
Entry conditions	The agronomist is logged in.
Event flow	1. In the home page, the system displays a “Visualize your visit plan” button. 2. The agronomist presses on that button. 3. The system displays a calendar: for each day, a list of the visits of that day is displayed, including the status of each visit. At the top of the page there is a button “Add a visit”. 4. The agronomist clicks on that button. 5. The systems displays a new page with the following mandatory fields: a. “Farmer”: here the agronomist will write the name of the farmer they intend to visit; b. “Date”; c. “Time”; d. “Duration”; 6. At the end of the page the button “Add visit” is displayed. 7. The agronomist fills in all the fields. 8. The agronomist presses on “Add visit”. 9. The visit is added in the visit plan according to the date and time, and its status is set to “scheduled”.

Exit condition	The system sends a notification to the Farmer about the new scheduled visit.
Exceptions	1. The agronomist clicks on “Add visit” without filling in one or more fields. The system displays an error message. 2. The agronomist sets the date and/or the time overlapped to another visit already scheduled, considering the duration of both visits and the distance automatically calculated between the two end points. The system displays an error message and invites the agronomist to modify date and/or time. 3. The agronomist sets as the date and the time of the new visit a date before the current date. The system displays an error message and invites the agronomist to modify date and/or time.



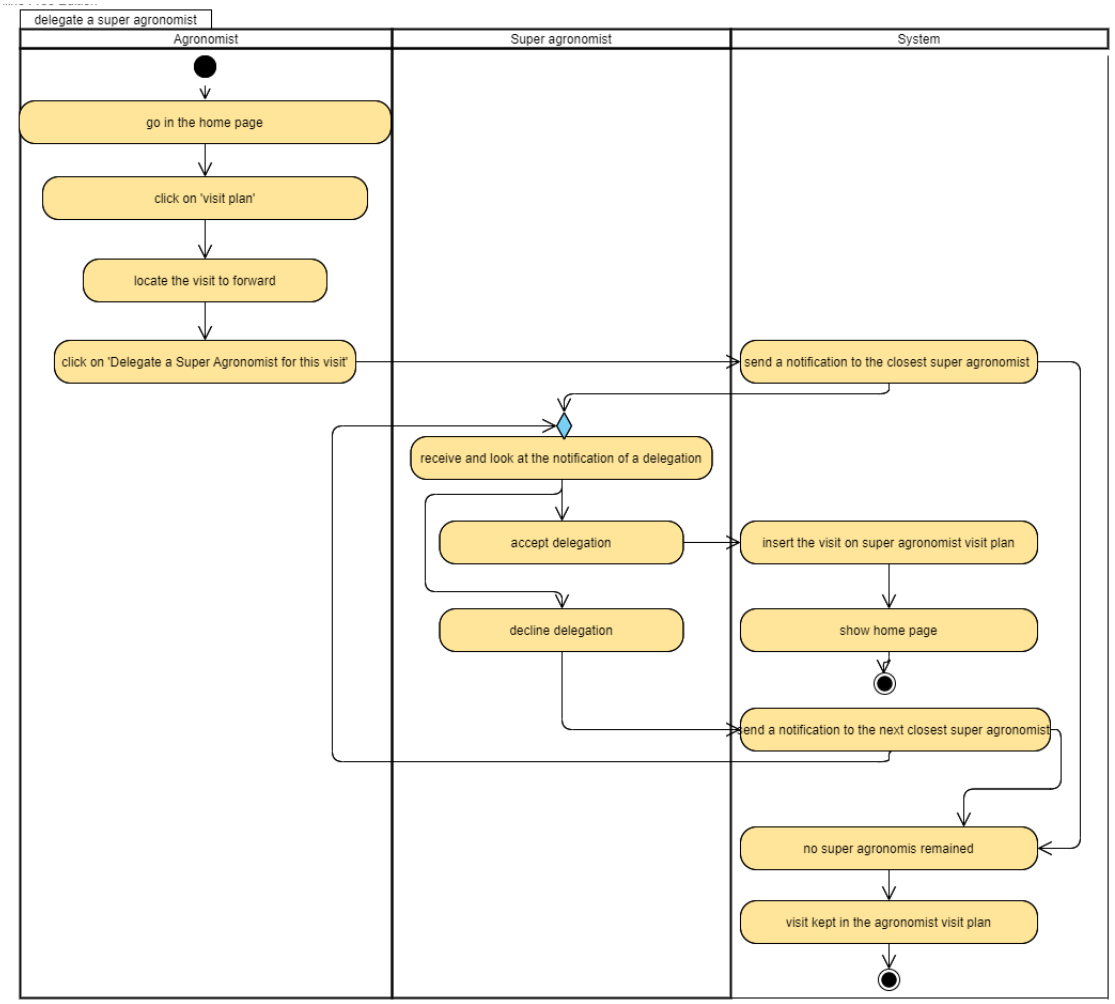
Name	Reschedule a visit after a visit-message
Actor	Agronomist Farmer
Entry conditions	The agronomist received a visit-message from a farmer that

	asks to modify a scheduled visit. The agronomist is logged in. The farmer is logged-in.
Event flow	1. The system shows the notification about a new visit message. 2. The agronomist clicks on the notification. 3. The system shows the visit message from the farmer and, at the side of the page, a button “Update visit”. 4. The agronomist reads the visit-message of the farmer. 5. The agronomist writes a visit-message with another date and time for the visit. 6. The farmer responds with a visit-message where they agree or disagree. 7. If the farmer agrees to the new date, the agronomist clicks on the “Update visit” button. 8. The system displays the old scheduled date and a field “New date”. 9. In the “New date” field, the agronomist inserts the data as agreed with the farmer. 10. The agronomist clicks on the “Confirm” button.
Exit condition	The visit date and time are updated and the visit is moved in the visit plan in the right position. The farmer receives a notification of confirmation about the updating. The visit-message between farmer and agronomist is successfully closed.
Exceptions	1. The farmer disagrees with the proposed new date. In this case the agronomist proposes another date and the exchange of visit-messages keeps going until the farmer and the agronomist reach an agreement. 2. The agronomist sends an empty message. The system displays an error message, and asks the agronomist to refill the message.



Name	Delegation of a visit to a super agronomist
Actor	Agronomist Super agronomist
Entry conditions	The agronomist realizes that they are not able to make a visit which is scheduled in their visit plan. The agronomist is logged in. The super agronomist is logged in.
Event flow	1. From the home page, the agronomist clicks on the “Visit plan” section. 2. The agronomist locates the visit they want to delegate. 3. The agronomist clicks on the visit. 4. The system displays an “Option” page. One of the the option is “Delegate a super agronomist for this visit” 5. The agronomist clicks on that option.

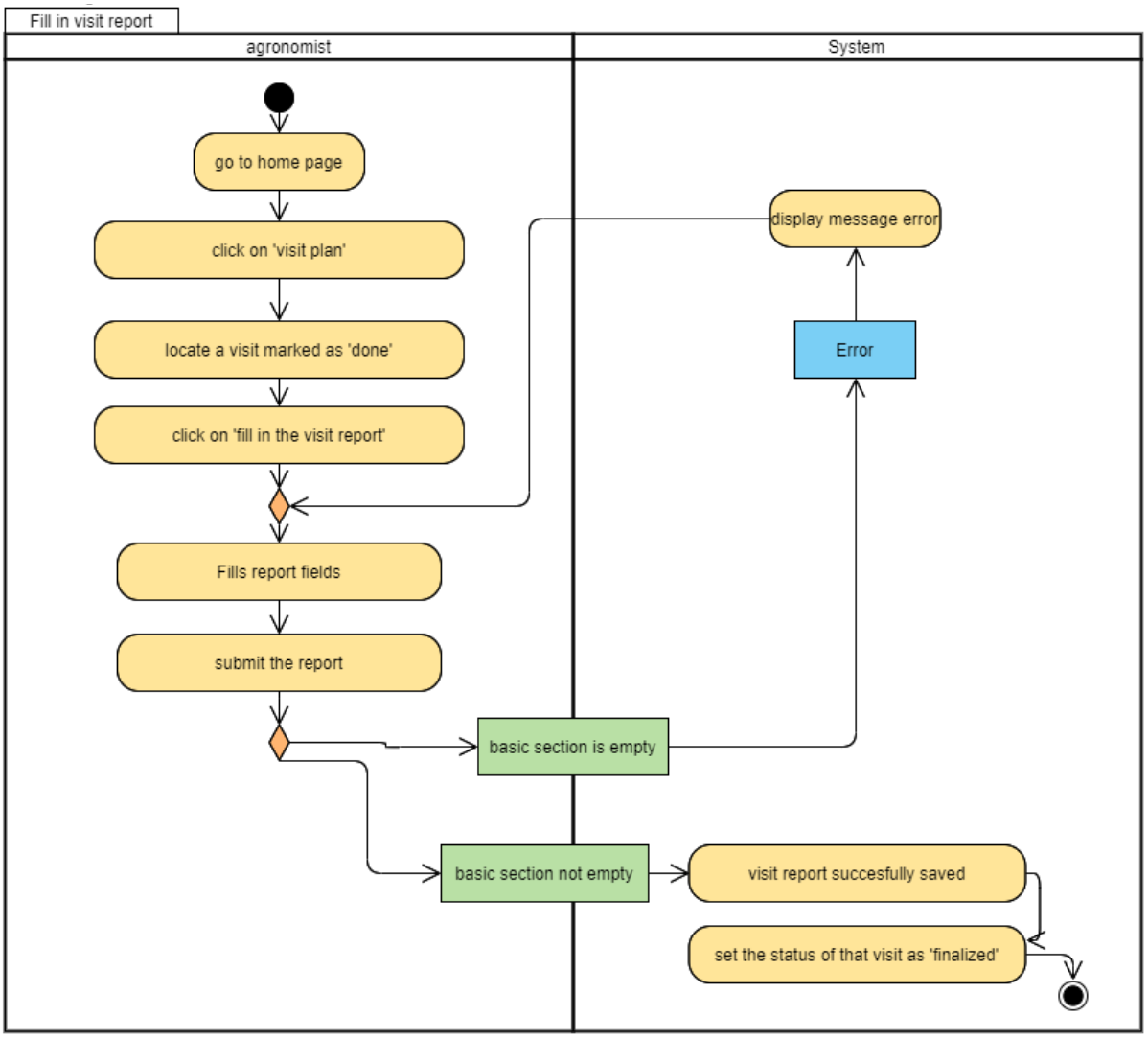
	6. The system sends a notification to the closest super agronomist wrt the farmer relative to the visit. 7. The super agronomist receives the notification. 8. The super agronomist clicks on the notification. 9. The system displays a page with the details of the forwarded visit and two buttons “Accept” and “Decline”. 10. The super agronomist clicks on “Accept”.
Exit condition	The visit is deleted from the agronomist’s visit plan. The visit is added to the super agronomist’s visit plan.
Exceptions	1. The super agronomist clicks on “Decline”. In this case, the visit is forwarded to the second closest super agronomist, according to the agronomist position, if any. 2. The forwarded visit is not accepted by any of the super agronomists in the district. In this case, the agronomists received a notification: “No super agronomist available in your district. You cannot forward this visit”, and the visit is not removed from the agronomist’s visit plan.



Name	Fill in the visit report
Actor	Agronomist
Entry conditions	The agronomist has made a visit. The agronomist updated the status of that visit to “done”. The agronomist is logged in.

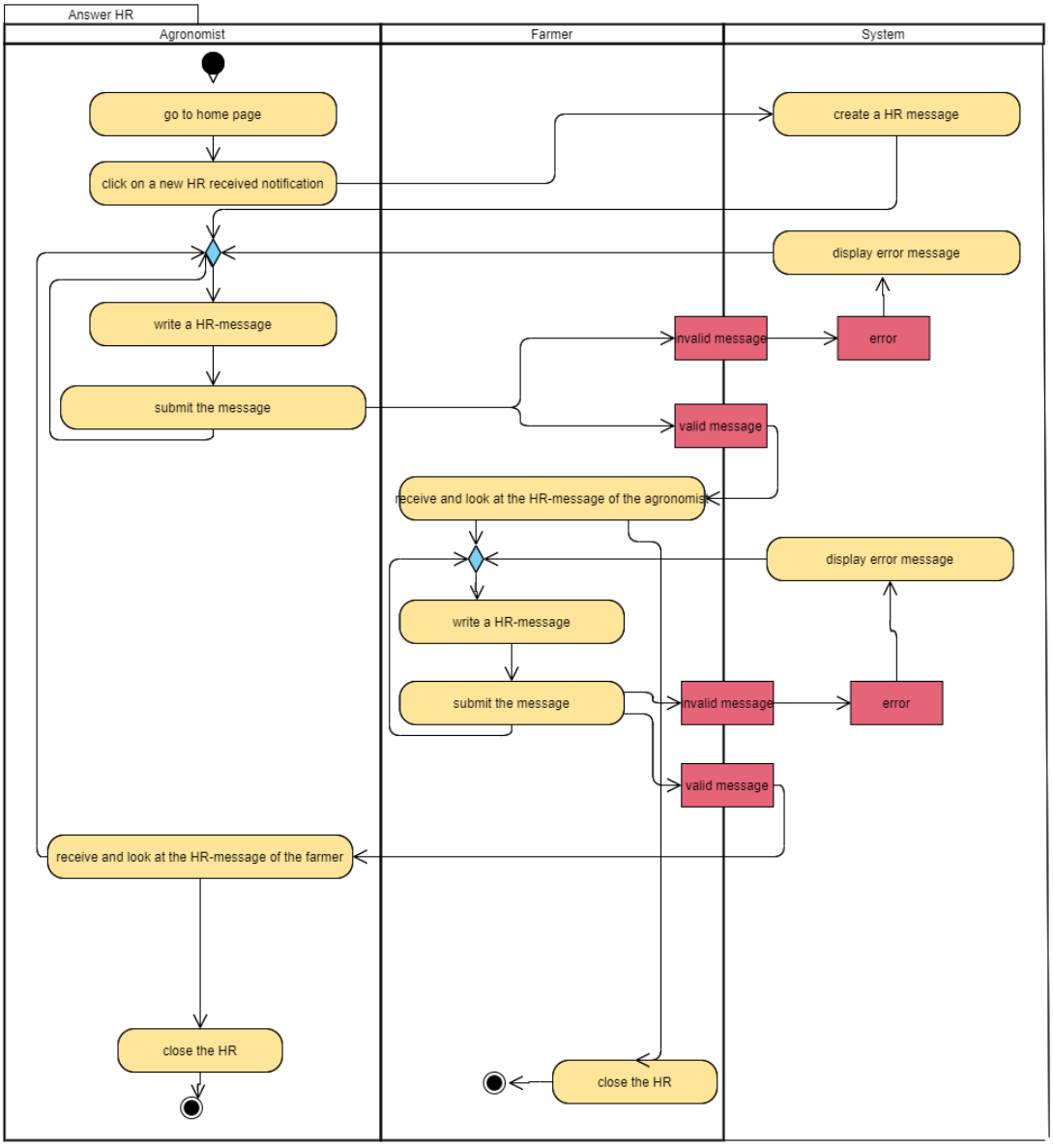
Event flow	1. From the home page, the agronomist clicks on “Visit plan”.
	2. The system shows the calendar with the visits.
	3. The agronomist locates the visit.
	4. The system displays the status “done” and a button “Fill in the visit report”.
	5. The agronomist clicks on that button.
	6. The system displays a page with the title “Visit report” and three sections:
	  a. basic section (mandatory), where to write the farmer’s assessment;
	  b. issue section (optional), in case of problems faced by the farmer;
	  c. weather section (optional), specific for weather-related issues.
	7. The agronomist fills in the basic section and, if necessary, the other two fields.
	8. At the bottom of the page, the system displays the “Submit” button.
	9. The agronomist presses on that button.

Exit condition	The report information is successfully stored into the system. The status of the visit is updated to “finalized”.
Exceptions	The Agronomist clicks on “Submit” without filling in the basic section. In this case, the system displays an error message.



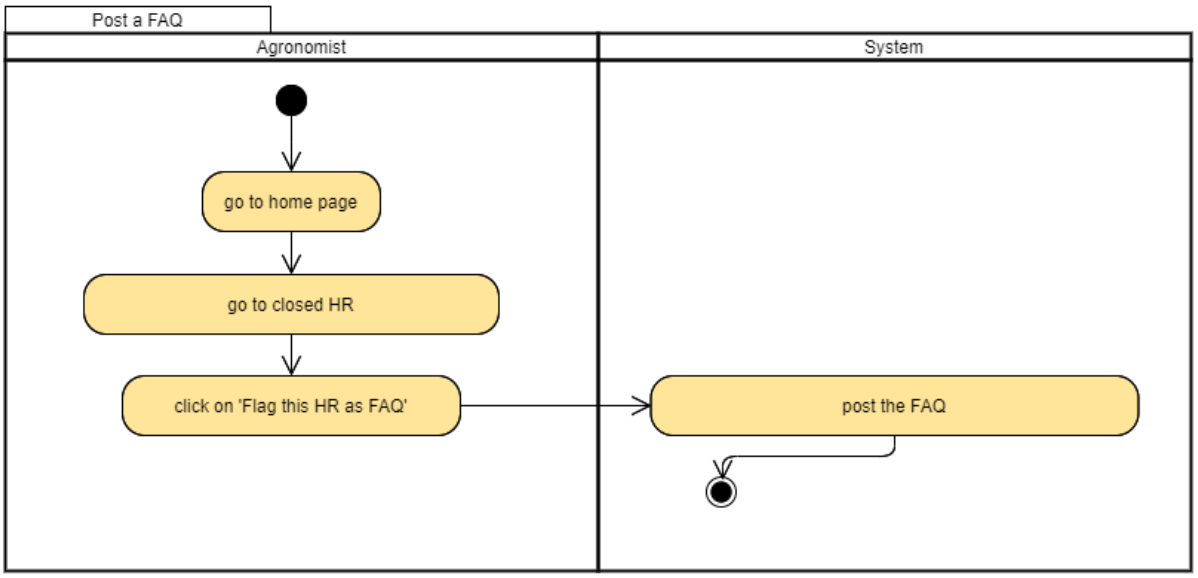
Name	Answer to a HR
Actor	Farmer Agronomist
Entry conditions	The agronomist received a notification about an HR from a farmer in their area. The farmer is logged in. The agronomist is logged-in
Event flow	1. From the home page, the agronomist clicks on the notification about the new HR received. 2. The agronomist writes one or possibly many HR-messages and submits it. 3. The farmer looks at the HR-message(s) received by the agronomist. 4. The farmer writes one or possibly many HR-messages and submits it. 5. The agronomist looks at the HR-message(s) received by the farmer. 6. The agronomist writes one or possibly many

	HR-messages and submits it. 7. Go to point 3.
Exit condition	One between farmer and agronomist closes the HR.
Exceptions	1. An agronomist clicks-on the “Submit” button without writing any HR-message. The system displays an error message. The agronomist is invited to try again. 2. A farmer clicks-on the “Submit” button without writing any HR-message. The system displays an error message. The farmer is invited to try again.



Name	Post a FAQ
Actor	Agronomist

Entry conditions	The agronomist and a farmer successfully resolved an exchange of HR-messages (the HR from the farmer has been answered and the HR is terminated). The agronomist is logged-in
Event flow	1. From the home page, the agronomist clicks on a closed HR. 2. The system shows to the agronomist the option “Flag this HR as FAQ”. 3. The agronomist selects that option.
Exit condition	The system posts the FAQ.
Exceptions	None



3.2.3 Policy Maker Scenarios

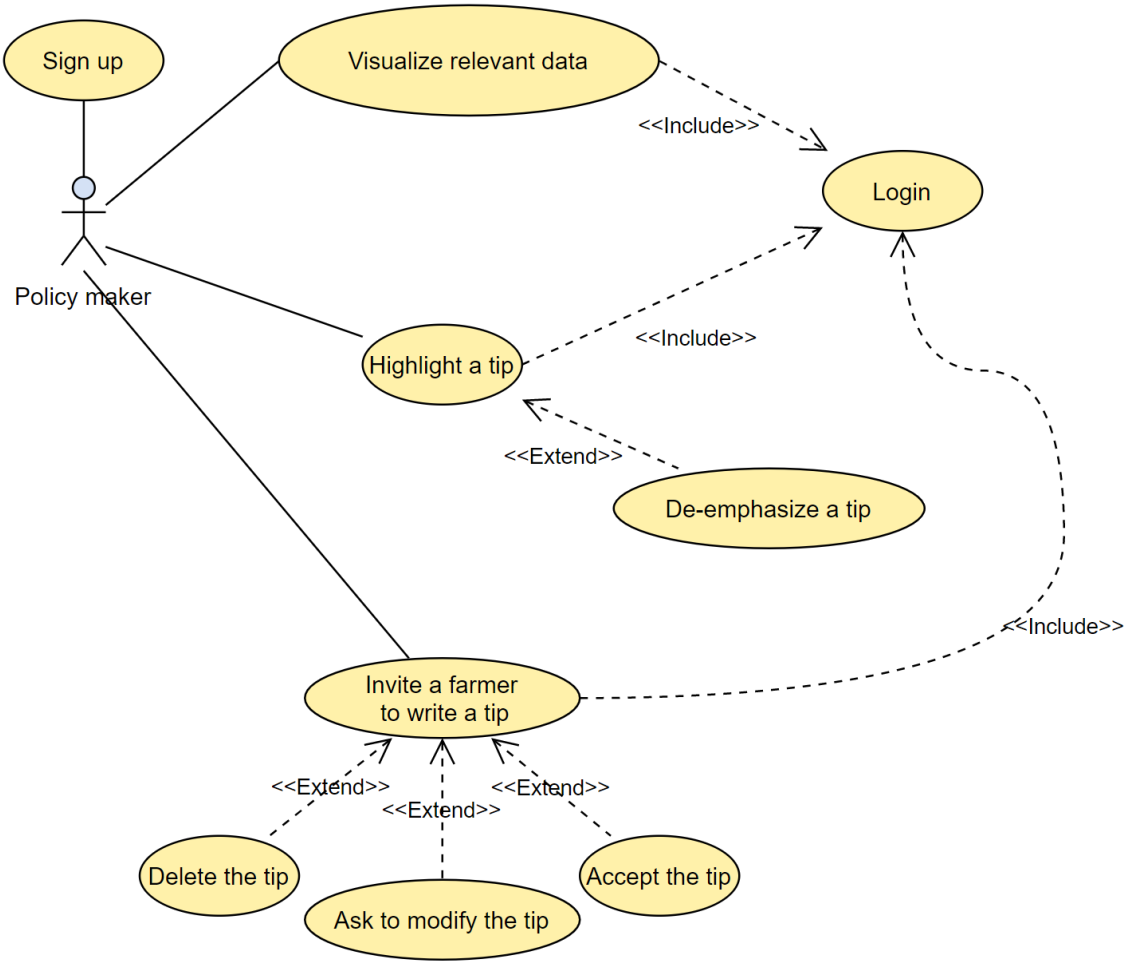
- **Scenario 1: Identify top-performing farmers and send a tip request**

  John is one of the policy makers of the Telangana state. He is in charge of the districts of Kamareddy. He wants to identify the farmers that are performing well in his district. In order to do so, he log-in with his credentials. In the home page he can take a look at the KPI of the farmers. He notices that the farmer Gabriel is having the best results in all the districts. Therefore, John invites him to post a tip through a tip request. Gabriel agrees and writes a helpful tip for dealing with heavy rain. The policy maker is satisfied: he approves Gabriel’s tip and makes it visible to the whole district (even though he could have decided to highlight it only in some specific areas). Now the farmer Gabriel will be rewarded for his cooperation.

- **Scenario 2: Highlight tips**

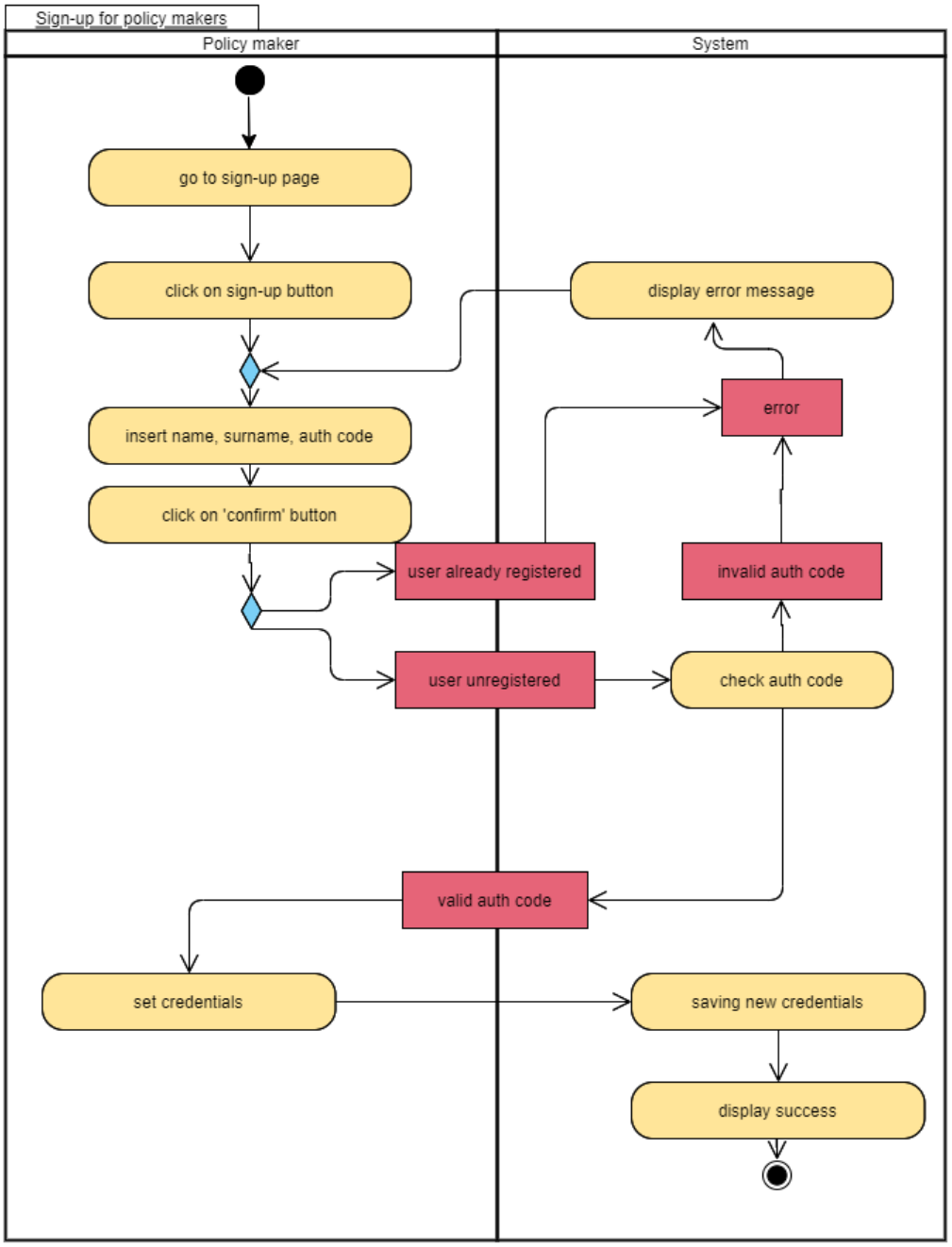
  The policy maker Sammy is taking a look at the tips posted in the forum of the DREAM application. In fact, she can visualize all the tips written by farmers from the district she is in charge of. She notices some very useful tips and decides to highlight them as “star tip”, so as to capture other farmers’ attention and be somehow useful to them. While scrolling, she notices that some already highlighted tips are a bit outdated, therefore she decides to “un-star” them.

3.2.3.1. Use cases diagrams



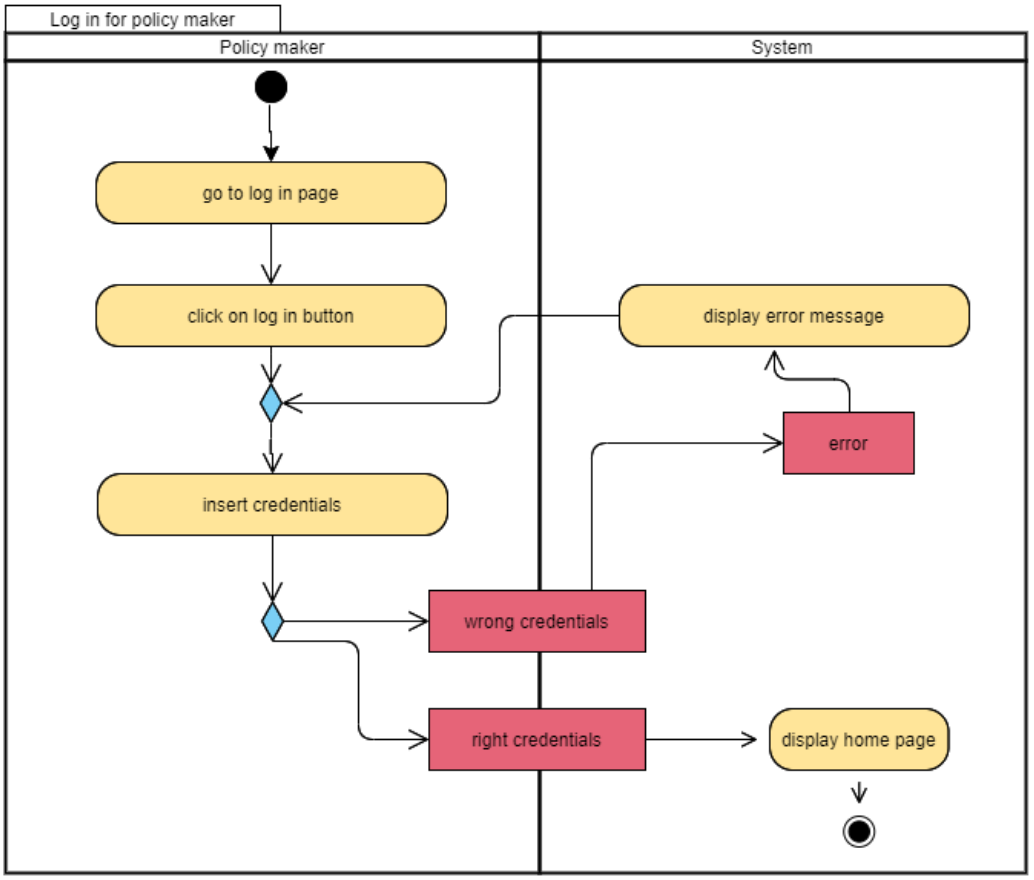
3.2.3.2. Use cases tables and activity diagrams

Name	Sign-up policy maker
Actor	Policy maker
Entry conditions	The policy maker opened the DREAM web application. The policy maker has an authorization code.
Event flow	1. The application shows the Sign-up page 2. The policy maker clicks on “Sign-up” 3. The application displays the following mandatory fields: first name and last name, authorization code. 4. The policy maker fills-in the mandatory fields. 5. The policy maker clicks on the “Confirm” button. 6. The app allows the policy maker to set the credentials. 7. The policy maker writes the credentials. 8. The policy maker clicks on ‘confirm’.
Exit condition	The policy maker registration is successfully completed and the policy maker’s data are saved into the system. The app shows the screen with the confirmation “You have successfully registered!”
Exceptions	1. The policy maker inserts a wrong authorization code. The application displays an error message and the policy maker is invited to try again. 2. The policy maker is already registered. The application shows the message “You have already registered! Log-in with your credentials”. The policy maker is invited to try again.



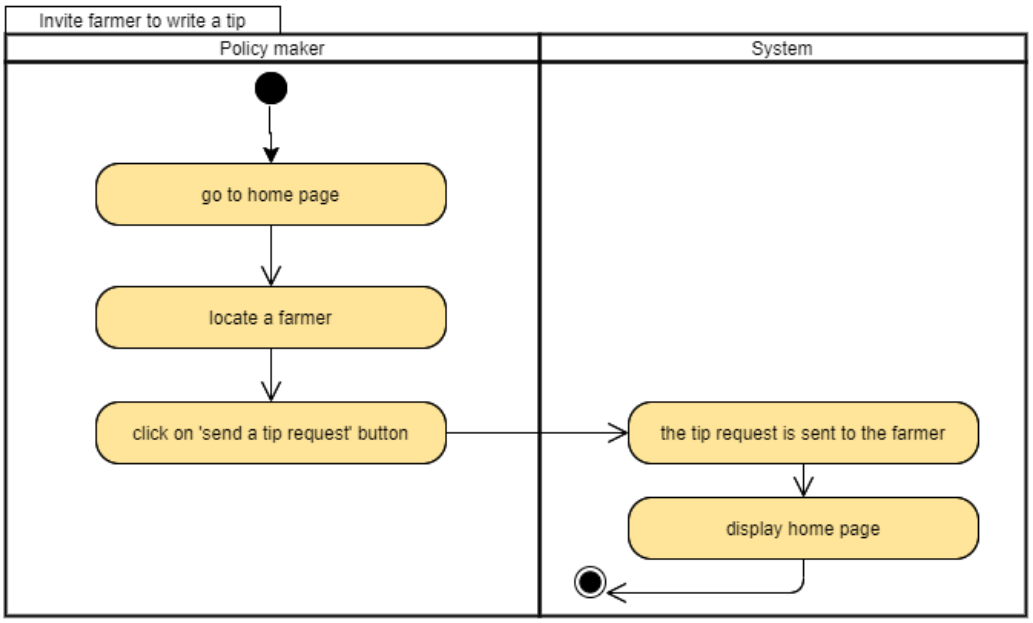
Name	Log-in policy maker
Actor	Policy maker
Entry conditions	The policy maker opened the DREAM web application. The policy maker is registered into the system.
Event flow	1. The system displays the log-in page. 2. The policy maker inserts their credentials into the mandatory fields. 3. The policy maker clicks on the “Log-in” button. 4. The system checks for the validity of the credentials.

	5. The system shows the home page of the application
Exit condition	The policy maker is successfully logged-in.
Exceptions	1. The policy maker inserts the wrong credentials. The system displays an error message and invites the policy maker to try again.

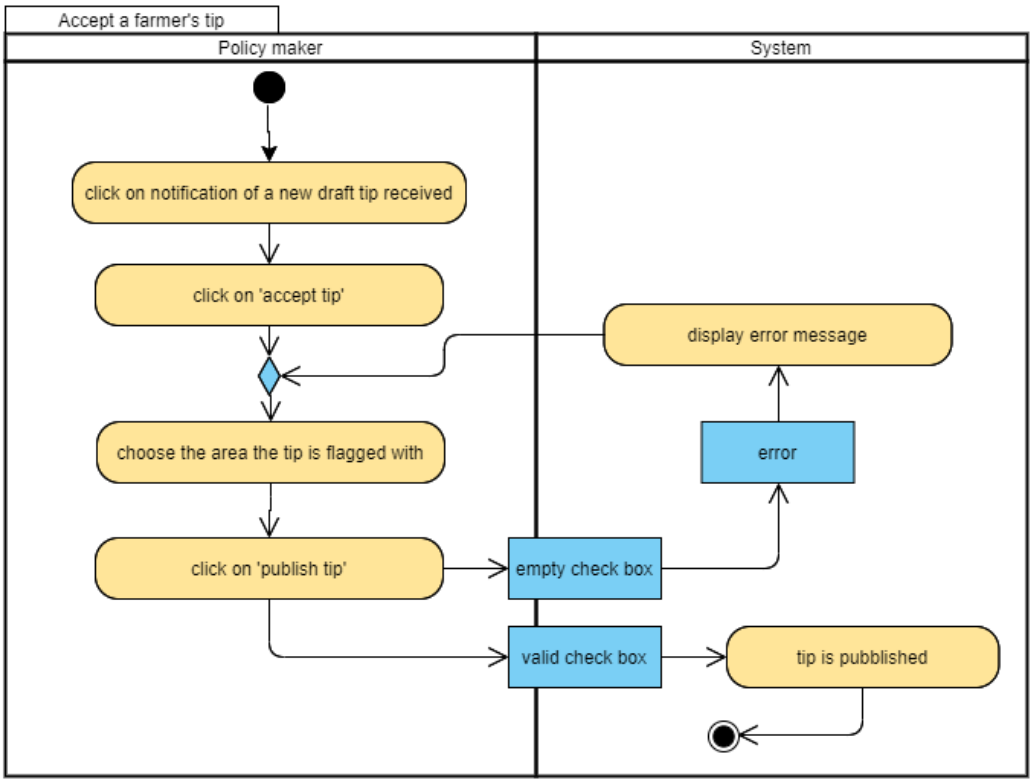


Name	Invite a farmer to write a tip
Actor	Policy Maker
Entry conditions	The Policy Maker is logged-in. The Policy Maker identified a farmer that is well performing by looking at the farmer's KPI. The Farmer is logged in.
Event flow	1. In the Policy Maker's home page, the system displays the data regarding the farmers' KPIs, ordered from the best performance to the poorest performance. 2. The Policy Maker clicks on the name of the farmer. 3. The system displays the button "Send a tip request to this farmer". 4. The Policy Maker clicks on that button.
Exit condition	The tip request is sent to the farmer. Display the homepage of

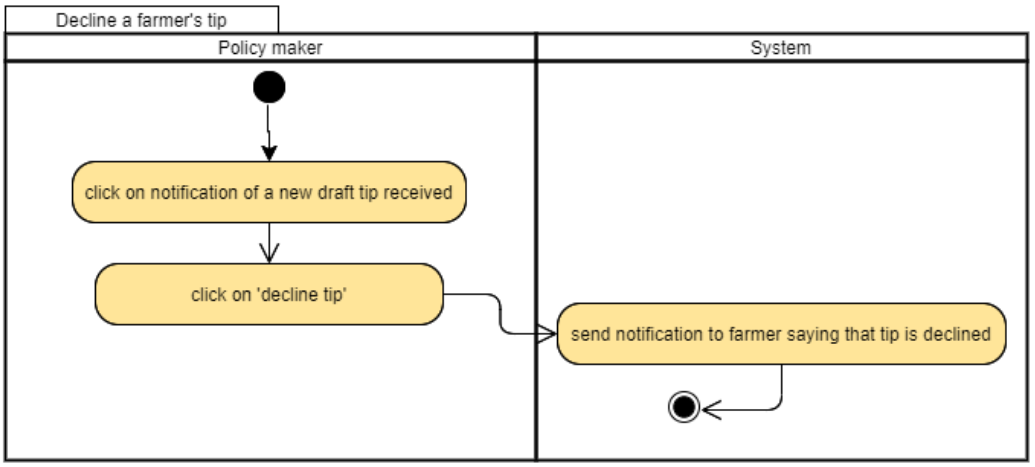
	the policy maker.
Exceptions	None



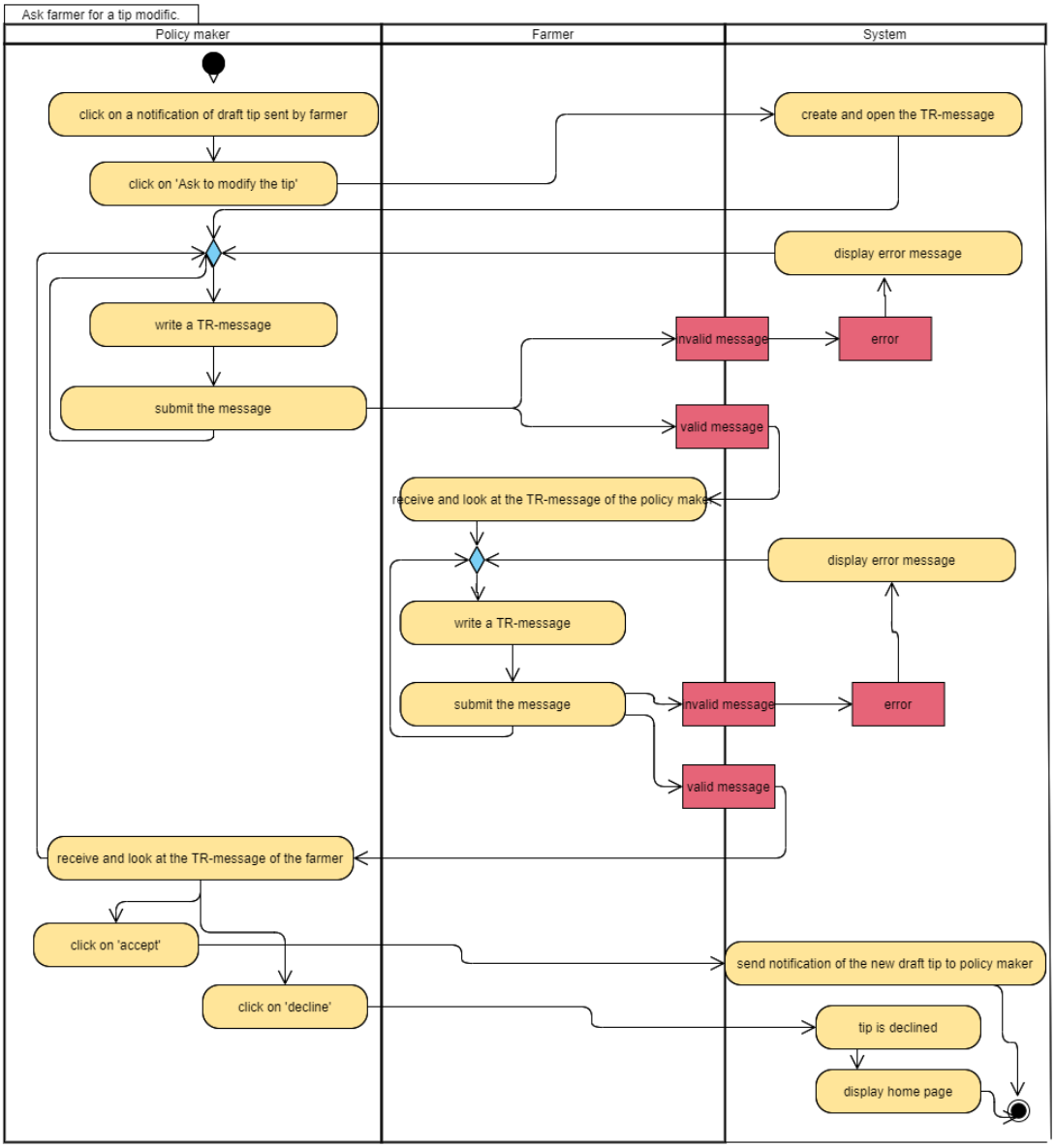
Name	Accept a Farmer's tip
Actor	Policy Maker
Entry conditions	The policy maker is logged-in. The policy maker requested a well-performing farmer to write a tip. The policy maker received a notification of a draft tip.
Event flow	1. The policy maker clicks on the notification. 2. The system displays to the policy maker the draft tip and three buttons "Accept tip", "Decline tip", "Ask to modify the tip". 3. The policy maker clicks on "Accept the tip". 4. The system displays the areas of the districts, each one with a checkbox, and asks to check the area in which the policy maker would like to publish the tip. 5. The policy maker checks the areas they want and clicks on the "Publish" button.
Exit condition	The tip is accepted and inserted into the forum.
Exceptions	The Policy Maker doesn't check any area and clicks on submit. The system displays an error message. The system invites the policy maker to try again.



Name	Decline a farmer's tip
Actor	Policy maker
Entry conditions	The policy maker is logged-in. The policy maker requested a well-performing farmer to write a tip. The policy maker received a notification of a draft tip.
Event flow	1. The policy maker clicks on the notification. 2. The system displays to the policy maker the draft tip and three buttons "Accept tip", "Decline tip", "Ask to modify the tip". 3. The policy maker clicks on "Decline the tip".
Exit condition	A notification "Your tip is rejected" is sent to the farmer.
Exceptions	None

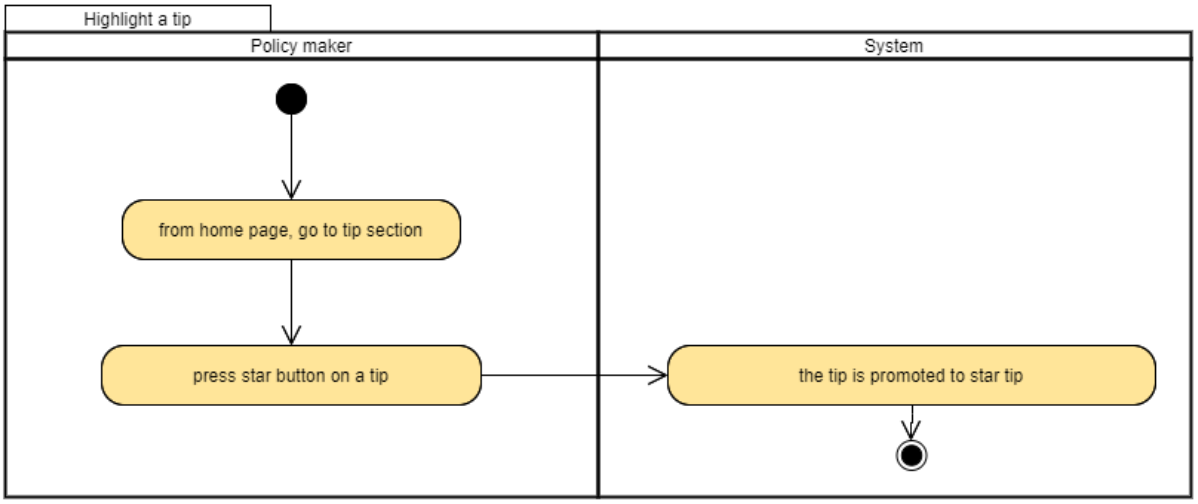


Name	Ask to modify a farmer's tip
Actor	Policy maker Farmer
Entry conditions	The policy maker is logged-in. The policy maker requested a well-performing farmer to write a tip. The farmer is logged in. The policy maker received a notification of a draft tip.
Event flow	1. The policy maker clicks on the notification. 2. The system displays to the policy maker the draft tip and three buttons "Accept tip", "Decline tip", "Ask to modify the tip". 3. The policy maker clicks on "Ask to modify the tip". 4. The policy maker writes one or many TR-messages and submits it. 5. The farmer looks at the TR-message(s) received by the policy maker. 6. The farmer writes one or possibly many TR-messages and submits it. 7. The policy maker looks at the TR-message(s) received by the farmer. 8. The policy maker writes one or many TR-messages and submits it. 9. Go to point 6.
Exit condition	1. The policy maker clicks-on "Accept" next to the farmer message: the policy maker receives a new notification of the updated draft tip. 2. The policy maker clicks-on "Decline" next to the farmer message: the tip is declined and the home page is shown.
Exceptions	One between policy maker and farmer click on 'Submit message' and the TR-message is empty. An error message is shown, the user is invited to try again.

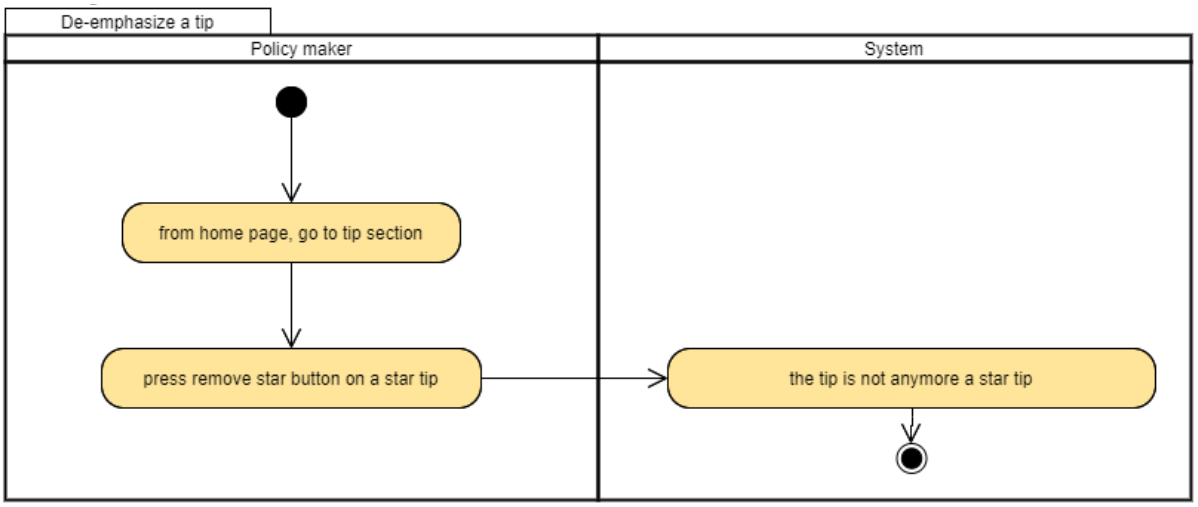


Name	Highlight a tip
Actor	Policy maker
Entry conditions	The policy maker is logged-in.
Event flow	1. From the home page, the policy make clicks on “Tips” button 2. The system displays a list of the tips belonging to the policy maker’s district. 3. Next to each tip which is not already highlighted there is a button “Star”. 4. The policy maker clicks on that button.
Exit condition	The tip became a “star” tip. The button next to it now displays “Remove star”.

Exceptions	None



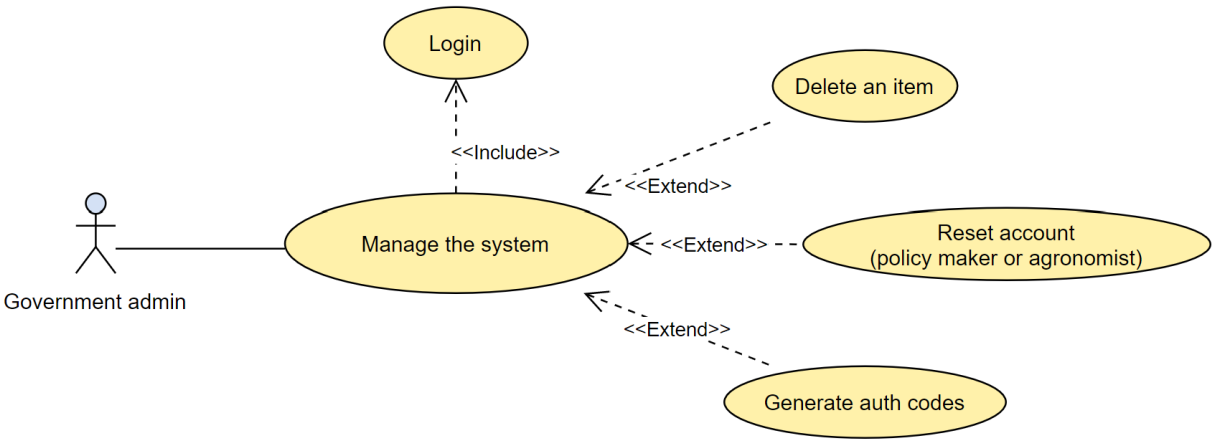
Name	De-emphasize a tip
Actor	Policy maker
Entry conditions	The policy maker is logged-in.
Event flow	5. From the home page, the policy make clicks on “Tips” button 6. The system displays a list of the tips belonging to the policy maker’s district. 7. Next to each tip which is already highlighted there is a button “Remove star”. 8. The policy maker clicks on that button.
Exit condition	The tip became a standard tip. The button next to it now displays “Star”.
Exceptions	None



3.2.4 Government admin Scenarios

Scenario: Moderating the forum.
Rob is one of the two government administrators of the state of Telangana. Since the application DREAM perfectly integrated into the Telangana farming system, he has been working as moderator in the DREAM forum. He reads questions, answers, tips and makes sure that the users of the forum are well behaving. Rob turns on his laptop and starts checking around. He reads about a tip posted 30 minutes ago: “Best movies of the last five years”, signed Farmer Zack. He asks himself why a farmer should write about movies in a farming forum, and he deletes the item.

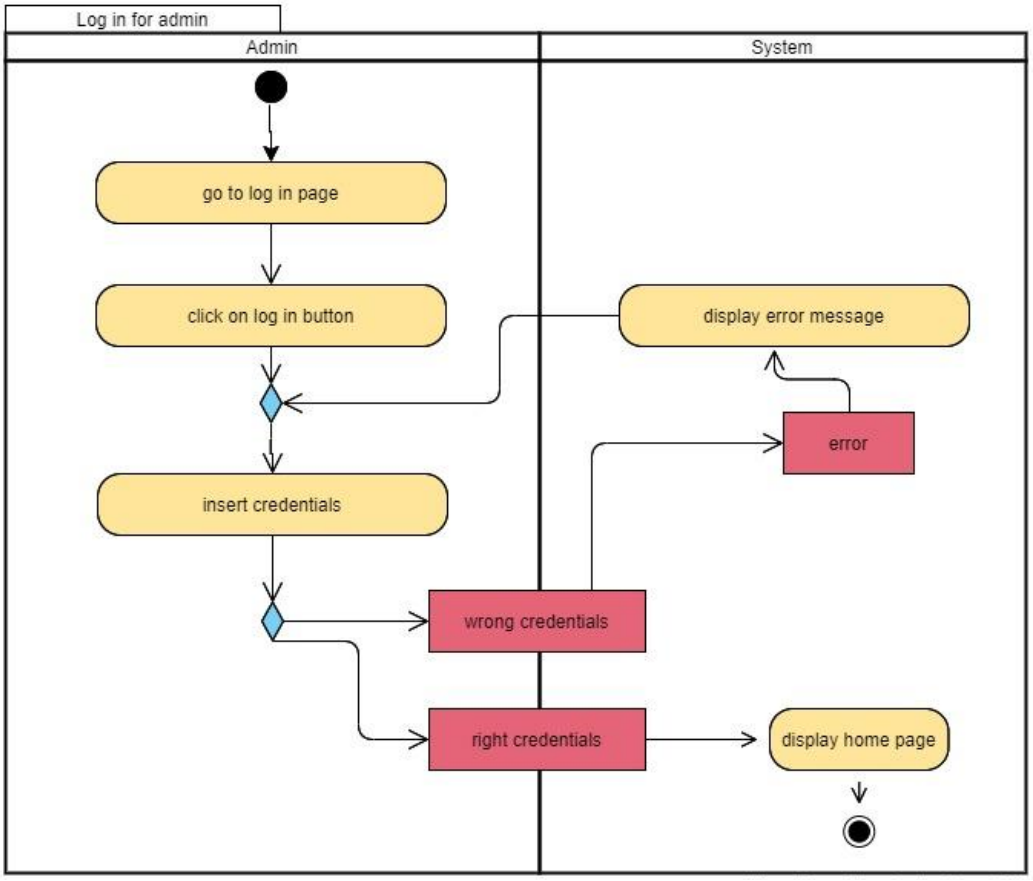
3.2.4.1. Use cases diagrams



3.2.4.2. Use cases tables and activity diagrams

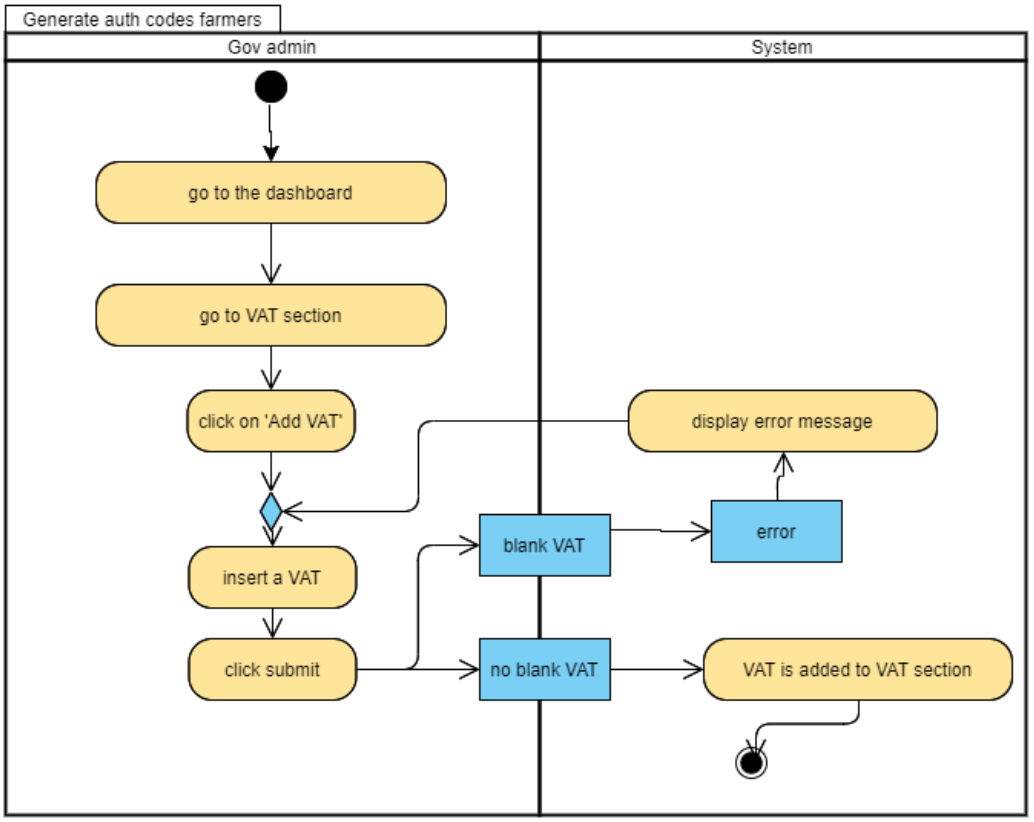
Name	Log in government admin
Actor	Government admin
Entry conditions	The government admin opened the DREAM web application. The government admin has their own credentials.
Event flow	1. The system displays the log-in page. 2. The government admin inserts the credentials into the mandatory fields. 3. The government admin clicks on the “Log-in” button. 4. The system checks for the validity of the credentials. 5. The system shows the home page of the application.

Exit condition	The government admin is successfully logged-in.
Exceptions	1. The government admin inserts the wrong credentials. The system displays an error message and invites the government admin to try again.



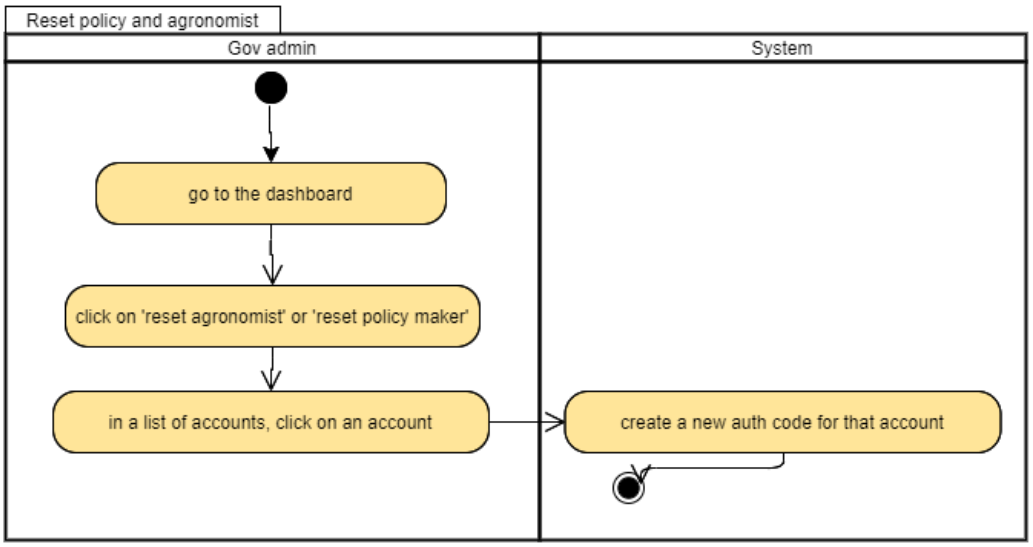
Name	Generate auth codes for farmers
Actor	Government admin
Entry conditions	The government admin is logged in.
Event flow	1. The government admin goes to the dashboard. 2. The system displays a section with the list of VATs belonging to the farmers that can already access DREAM, and a button “Add a VAT”. 3. The government admin clicks on the button 4. A blank field is displayed. 5. The government admin writes the VAT of the farmer they want to allow access to the app. 6. The government admin clicks submit.
Exit condition	The new VAT is added to the list. Now the farmer can use their

	VAT as auth code for signing up.
Exceptions	The government admin inserts a blank VAT. The system displays an error message and invites the government admin to try again.

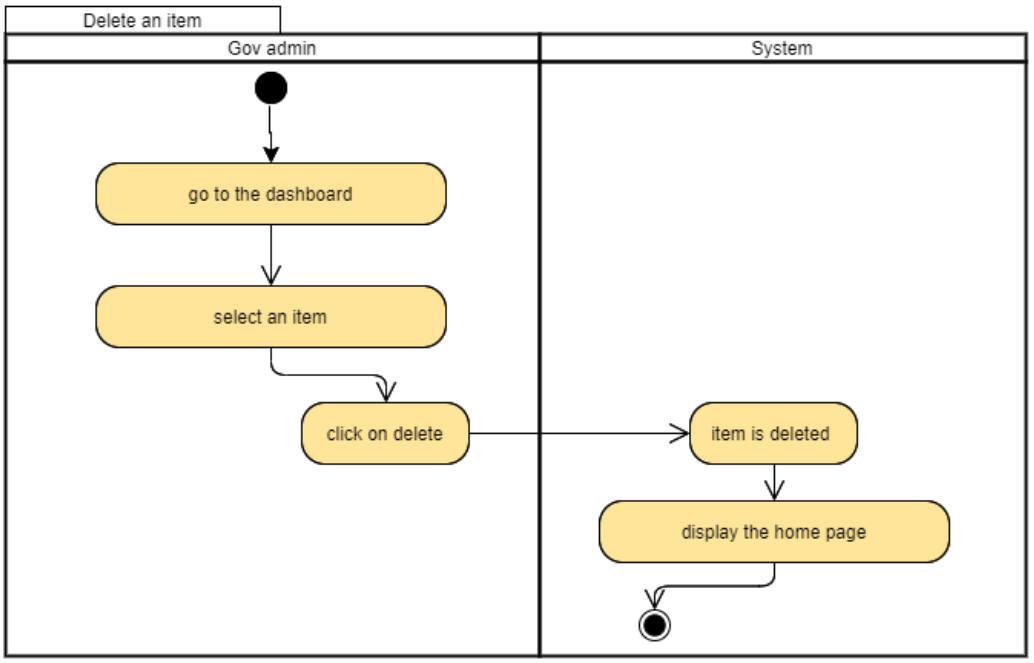


Name	Reset a policy maker or agronomist's account
Actor	Government admin
Entry conditions	A policy maker or agronomist account has to be reseted. The government admin is logged in.
Event flow	1. The government admin goes to the dashboard. 2. The system displays the buttons "Reset an agronomist account" and "Reset a policy maker account". 3. The government admin clicks on the right button. 4. The system displays a list of the policy makers' accounts or the agronomists' accounts, depending on the button they chose. 5. The government admin clicks on the account they want to reset. 6. The system automatically generates a new auth code for that profile and displays it.
Exit condition	The profile is resetted: all information is kept besides personal data, settings and credentials which will be set up at the next

	sign up.
Exceptions	None



Name	Delete an item
Actor	Government admin
Entry conditions	An item (farmer account or post) has to be deleted. The government admin is logged-in.
Event flow	1. The government admin goes to the dashboard. 2. The government admin clicks on an item (it could be a post or also a farmer account). 3. The system displays a “Delete” button. 4. The government admin clicks on that button.
Exit condition	The item is deleted and the system displays the home page.
Exceptions	None



3.2.5. Requirements

G1: Sharing of knowledge among farmers

D2: Weather information is obtained through an API provided by an external meteorological service (weather forecast API).

D8: The majority of the farmers of Telangana use DREAM and the application is uniformly distributed on the territory.

D9: For each district, there is only one policy maker referring to it. In addition, there is another policy maker referring to all districts.

D10: Farmers have the availability of a device with internet access, and the GPS location at least in the registration phase.

D11: Each farmer has a VAT number.

D12: The government will instruct the role of government admins to well trained and trusted people that will act according to what the goals of the system are.

R1: the system allows farmers to register using an authorization code (VAT).

R2: the system allows farmers to log in.

R3: the system allows agronomists to register using an authorization code.

R4: the system allows agronomists to log in.

R5: the system allows farmers and agronomists to publish tips in the forum.

R6: the system allows farmers to publish a question in the forum.

R7: the system allows farmers and agronomists to answer a question in the forum.

R8: the system allows farmers and agronomists to vote tips and answers in the forum.

R9: the system allows farmers to visualize relevant information regarding weather and tips.

R24: the system allows policy makers to register using an authorization code.

R25: the system allows policy makers to log in.

R26: the system allows policy makers to highlight a tip as a star tip.

R27: the system allows policy makers to send a tip request.

R28: the system allows policy makers and farmers to use tip request messages.

R29: the system allows policy makers to de-highlight a tip.

R31: the system allows government admin to log-in

R32: the system allows policy makers to visualize farmers KPIs.

R36: the system allows government admins to manage authorization codes.

R38: the system allows government admins to delete a farmer's account.

R39: the system allows government admins to delete posts or answers.

G2: Tracking farmers' production

D8: The majority of the farmers of Telangana use DREAM and the application is uniformly distributed on the territory.

D10: Farmers have the availability of a device with internet access, and the GPS location at least in the registration phase.

D11: Each farmer has a VAT number.

D12: The government will instruct the role of government admins to well trained and trusted people that will act according to what the goals of the system are.

R1: the system allows farmers to register using an authorization code (VAT).

R2: the system allows farmers to log in.

R10: the system allows farmers to fill in the harvest report.

R31: the system allows government admin to log-in

R36: the system allows government admins to manage authorization codes.

R38: the system allows government admins to delete a farmer's account.

G3: Allowing farmers to get helped

D1: Data about humidity of soil and irrigation system is made available by an external system (government API).

D2: Weather information is obtained through an API provided by an external meteorological service (weather forecast API).

D3: There is an analysis team external to the system that defines metrics and KPIs.

D4: The KPIs and metrics definition is meaningful.

D5: The areas are defined by the government stakeholders such that the workload can be managed by one agronomist, in the worst case with the help of super-agronomists.

D6: An agronomist will always have the time to deal with all the HRs sent by farmers of their area.

D7: There is the availability of one agronomist for each area.

D8: The majority of the farmers of Telangana use DREAM and the application is uniformly distributed on the territory.

D9: For each district, there is only one policy maker referring to it. In addition, there is another policy maker referring to all districts.

D10: Farmers have the availability of a device with internet access, and the GPS location at least in the registration phase.

D11: Each farmer has a VAT number.

D12: The government will instruct the role of government admins to well trained and trusted people that will act according to what the goals of the system are.

R1: the system allows farmers to register using an authorization code (VAT).

R2: the system allows farmers to log in.

R3: the system allows agronomists to register using an authorization code.

R4: the system allows agronomists to log in.

R11: the system allows farmers to send an HR to farmers.

R12: the system allows farmers to send an HR to his area's agronomist.

R13: the system allows farmers to accept or decline an HR coming from another farmer.

R14: the system allows farmers and agronomists to use HR messages.

R15: the system allows agronomists to add a new visit.

R16: the system allows agronomists to update an existing visit.

R17: the system allows farmers and agronomists to use visit messages.

R18: the system allows farmers and agronomists to terminate visit messages.

R19: the system allows agronomists to publish HR-messages as FAQ.

R20: the system allows agronomists to fill in and confirm a visit report.

R21: the system allows agronomists to change the status of a visit.

R22: the system allows agronomists to visualize farmers KPIs.

R23: the system allows agronomists to delegate a visit to super agronomists.

R30: the system adds the two mandatory visits to an agronomist's visit plan when a new farmer signs up.

R31: the system allows government admin to log-in

R32: the system allows policy makers to visualize farmers KPIs.

R33: the system allows agronomists to become a super agronomist.

R34: the system allows policy makers to visualize agronomist related KPIs.

R35: the system allows agronomists to visualize agronomist related KPIs.

R36: the system allows government admins to manage authorization codes.

R37: the system allows government admins to reset an agronomist or policy maker's account.

R38: the system allows government admins to delete a farmer's account.

R39: the system allows government admins to delete posts or answers.

G4: Supporting agronomist work

D1: Data about humidity of soil and irrigation system is made available by an external system (government API).

D2: Weather information is obtained through an API provided by an external meteorological service (weather forecast API).

D3: There is an analysis team external to the system that defines metrics and KPIs.

D4: The KPIs and metrics definition is meaningful.

D5: The areas are defined by the government stakeholders such that the workload can be managed by one agronomist, in the worst case with the help of super-agronomists.

D6: An agronomist will always have the time to deal with all the HRs sent by farmers of their area.

D7: There is the availability of one agronomist for each area.

D12: The government will instruct the role of government admins to well trained and trusted people that will act according to what the goals of the system are.

R1: the system allows farmers to register using an authorization code (VAT).

R2: the system allows farmers to log in.

R3: the system allows agronomists to register using an authorization code.

R4: the system allows agronomists to log in.

R15: the system allows agronomists to add a new visit.

R16: the system allows agronomists to update an existing visit.

R17: the system allows farmers and agronomists to use visit messages.

R18: the system allows farmers and agronomists to terminate visit messages.

R20: the system allows agronomists to fill in and confirm a visit report.

R21: the system allows agronomists to change the status of a visit.

R23: the system allows agronomists to delegate a visit to super agronomists.

R30: the system adds the two mandatory visits to an agronomist's visit plan when a new farmer signs up.

R31: the system allows government admin to log-in

R33: the system allows agronomists to become a super agronomist.

R35: the system allows agronomists to visualize agronomist related KPIs.

R36: the system allows government admins to manage authorization codes.

R37: the system allows government admins to reset an agronomist or policy maker's account.

G5: Providing data-driven decision tools to agronomists and policy makers

D1: Data about humidity of soil and irrigation system is made available by an external system (government API).

D2: Weather information is obtained through an API provided by an external meteorological service (weather forecast API).

D3: There is an analysis team external to the system that defines metrics and KPIs.

D4: The KPIs and metrics definition is meaningful.

D7: There is the availability of one agronomist for each area.

D8: The majority of the farmers of Telangana use DREAM and the application is uniformly distributed on the territory.

D9: For each district, there is only one policy maker referring to it. In addition, there is another policy maker referring to all districts.

D10: Farmers have the availability of a device with internet access, and the GPS location at least in the registration phase.

D11: Each farmer has a VAT number.

D12: The government will instruct the role of government admins to well trained and trusted people that will act according to what the goals of the system are.

R1: the system allows farmers to register using an authorization code (VAT).

R2: the system allows farmers to log in.

R3: the system allows agronomists to register using an authorization code.

R4: the system allows agronomists to log in.

R24: the system allows policy makers to register using an authorization code.

R25: the system allows policy makers to log in.

R22: the system allows agronomists to visualize farmers KPIs.

R26: the system allows policy makers to highlight a tip as a star tip.

R29: the system allows policy makers to de-highlight a tip.

R10: the system allows farmers to fill in the harvest report.

R31: the system allows government admin to log-in

R32: the system allows policy makers to visualize farmers KPIs

R34: the system allows policy makers to visualize agronomist related KPIs.

R35: the system allows agronomists to visualize agronomist related KPIs.

R36: the system allows government admins to manage authorization codes.

R37: the system allows government admins to reset an agronomist or policy maker’s account.

3.3. Performance Requirements

The system for achieving the goals has to be used by the majority of farmers in Telangana then it has to guarantee a number of operations in the order of millions concentrated during the day and a significant decrease of load during the night.

It is essential to have a short response time for all the chat features of the system, the maximum response time should be less than 1.5 seconds, for the other functionality higher response time could be tolerated however they don't have to exceed 5 seconds.

3.4. Design Constraints

3.4.1. Standards compliance

DREAM' contents have to be compliant with the standard WCAG 2.0 in order to guarantee accessibility to a wider range of people with disabilities, including blindness and low vision, deafness and hearing loss, learning disabilities, cognitive limitations, limited movement, speech disabilities, photosensitivity and combinations of these [3].

3.5. Software System Attributes

3.5.1. Reliability

The system has to be capable of detecting runtime problems that can prevent its functions and being able to let the users know it and preventing them to take new actions until the system is repaired, in order that whatever problem occur it is not masked to the clients, whenever they use the system they need the guarantee that all the actions performed on the system are not lost due to errors.

3.5.2. Availability

The system has to be "highly available", an availability between 99.95% and 99.99% is required since in the worst case of having all the unplanned downtime in one day higher than 5 hours can lead to problems in the communication between farmers and agronomists.

3.5.3. Security

The exchange of direct messages among users inside the system has to be encrypted to guarantee the protection of possible sensible information presents.
Also all the authentication procedures have to be encrypted.

The system has to guarantee the confidentiality of the data based on which user is using it.

3.5.4. Maintainability

The government admins are responsible for the maintenance of the system after its deployment they have the permission of doing everything that all the other users can do with the addition of the rights to delete anything from the forum and managing user profiles, deleting an account if needed or creating new ones, they are responsible in case an agronomist has to be assigned to another place or removed from the system same for the policy maker's accounts.

3.5.5. Portability

DREAM has to be utilized ideally by every farmer of Telangana, then it has to be supported by all the principal operating systems and on all the devices and it has to be compatible with all the principal browsers.
To use DREAM no installation is required in the clients' devices all the services are available browsing url address from the web browser.

3.5.6. Usability

The UX (User experience) should be as simple as possible, the number of actions that can be taken from any page should be reduced to essential and each action component has to be highlighted.
Since the system can be used on devices with any screen size the usability has to be high whatever devices is used

4. FORMAL ANALYSIS USING ALLOY

The formal analysis has been carried out for the forum and visit part separately, the used code is reported as some images taken by the code execution.
In the following the expression like “an instance of the model has not been found” will be used for simplicity to refer to simulations of the world with a sufficiently high number of allowed signatures (the number used is reported in the code) that did not result in any allowed world therefore not a formal proof but still a valuable result given the high number of signatures [5].

4.1. Visit part

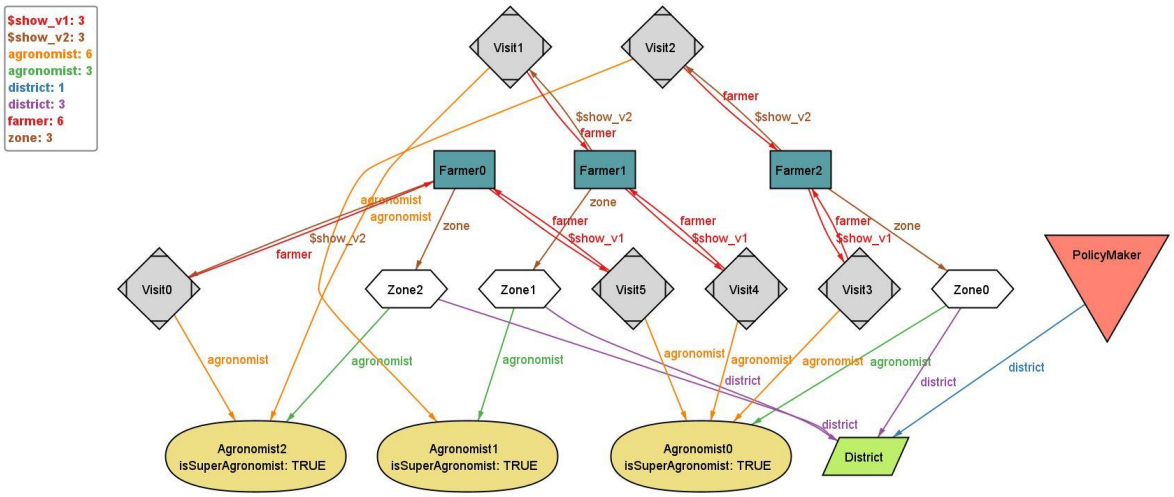
The model coherence has been proven using 5 assertions.
With the test2 and test1 no instance of the model can be found, test1 confirms that agronomists cannot perform visits outside their district in general and test1 that all the non-super agronomists perform visits only in their zone, the picture is completed with test3 that can find a counterexample of super agronomists performing visits outside their area combining all the first 2 tests we can say that only super agronomists can work outside their area but never going outside their district.
For farmers the test4 and test5 show that every farmer must receive at least 2 visits, since test4 find a counterexample to all the farmers not having 2 visits and for test5 ensure that for one or more farmers with just 1 or 0 visits no instance of the model can be found.

```
sig Farmer {
	zone: one Zone
}
sig Agronomist {
	isSuperAgronomist: one Bool
}{ // for all agronomist there is a correspondent zone of association
all agr : Agronomist | ( one zone : Zone | zone.agronomist = agr)
}
sig PolicyMaker {
	district: one District
}
sig District{
}{	 // for all districts there is at least 1 zone
	all d : District | ( some zone : Zone | zone.district = d)
	// for all districts there is at least 1 policy maker
	all d : District | ( one p : PolicyMaker | p.district = d)
}
sig Zone{
	district: one District,
	agronomist: one Agronomist
}{
	// for all zones there is at least one farmer
	all z: Zone | (some f: Farmer | f.zone = z)
}
sig Visit{
	farmer: one Farmer,
	agronomist: one Agronomist,
}
abstract sig Bool {}
one sig TRUE extends Bool {}
one sig FALSE extends Bool {}
```



```
//++++++++++++ FACTS ++++++++++++

fact{ //a visit can be carried out by an agronomist or by a superagronomist operating in the same district
 (all v:Visit | all z:Zone | z.agronomist = v.agronomist => z = v.farmer.zone ) or
(all v:Visit | all z:Zone | z.agronomist = v.agronomist =>( z.district = v.farmer.zone.district && z.agronomist.isSuperAgronomist =
TRUE) )
}

fact { //a unique agronomist is associated to each zone
        no disj z1 , z2 : Zone | z1.agronomist = z2 .agronomist
}

fact{ //a unique policy maker is associated to each district
        no disj p1 , p2 : PolicyMaker | p1.district = p2.district
}

fact{//each farmer have to have associated at least 2 visits
        all f:Farmer | some v1,v2 : Visit | v1 != v2 and ( v1.farmer = f and v1.farmer = v2.farmer)
}

//++++++++++++ ASSERTIONS ++++++++++++

assert test1{ // no agronomist outside their district ( for #instance > 50 enlarge memory size in options)
no v:Visit | some  z:Zone | z.agronomist = v.agronomist and ( z.district != v.farmer.zone.district)
}
assert test2{ // no NON SUPER agronomist outside their zone
no v:Visit | some  z:Zone | z.agronomist = v.agronomist and ( z != v.farmer.zone) and (z.agronomist.isSuperAgronomist = FALSE)
}
assert test3{ // here we look for a counterexample that an agronomist can work outside a zone if isSuperAgronomist = TRUE
no v:Visit | some  z:Zone | z.agronomist = v.agronomist and ( z != v.farmer.zone) and (z.agronomist.isSuperAgronomist = TRUE)
}
assert test4{ // no farmer has 2 visits, counter example expected
no f:Farmer | some v1,v2 : Visit | v1 != v2 and ( v1.farmer = f and v2.farmer = f)
}
assert test5{ // no farmer cannot be visited 0 or 1 times
(no f:Farmer | f not in Visit.farmer) or
(some f:Farmer | one v1: Visit | no v2 : Visit | v1 != v2 and ( v1.farmer = f and v2.farmer = f))
}

//++++++++++++ EXECUTION PARAMS ++++++++++++
pred show{
#Agronomist>= 3
#Farmer =3
#Visit = 6
}
run show for 40

//check test5 for 40
```

Example of a modelinstance where a super agronomist can perform visits on the district but different zones: Agronomist0 is responsible of Zone0 but visits (Visit 4 and 5) Farmer0 and Farmer1 not working in Zone1



4.2. Forum part

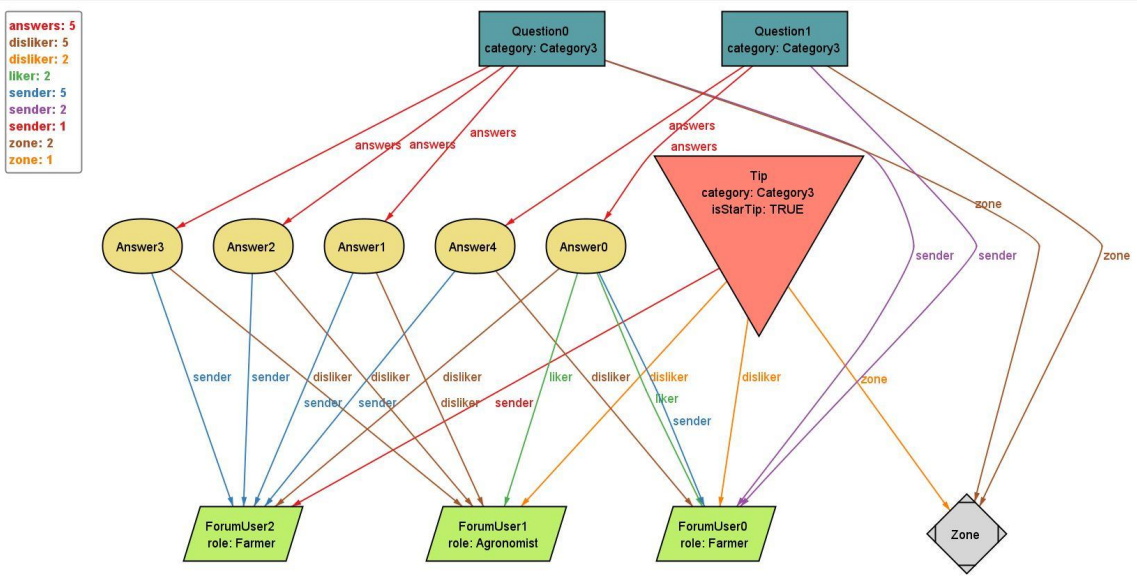
The model coherence has been proven using 4 assertions.
Test1 and test2 ensure that all questions have a farmer as sender and each of all the answers in the forum is associated with exactly one question.
The assertion test3 guarantees that the same forum user cannot like and dislike a tip at the same time, test4 does the same check but for answers.

```
enum Categories { //categories of forum posts
Category1,
Category2,
Category3
}
enum Roles { //roles of user in the forum
Farmer,
Agronomist
}
sig Zone{ // post are associated with a certain zone
}
sig ForumUser{
role : one Roles
}
sig Question{
sender: one ForumUser,
answers : set Answer,
category: one Categories,
zone: one Zone
}
sig Answer{
sender: one ForumUser,
liker: set ForumUser,
disliker: set ForumUser
}
sig Tip{
sender: one ForumUser,
liker: set ForumUser,
disliker: set ForumUser,
category: one Categories,
```



```
zone: one Zone,
isStarTip: one Bool
}

abstract sig Bool {}
one sig TRUE extends Bool {}
one sig FALSE extends Bool {}

//++++++++++++ FACTS ++++++++++++

fact{//one forum user can only like or dislike a likable item; notice user can like their own items
all a: Answer | all u: ForumUser | ((u in a.liker) => (u not in a.disliker)) and ((u in a.disliker) => (u not in a.liker))
all t: Tip | all u: ForumUser | ((u in t.liker) => (u not in t.disliker)) and ((u in t.disliker) => (u not in t.liker))
}

fact{//only farmers can ask questions
all q : Question | q.sender.role = Farmer
}

fact{// answers have to be associated with a question
        all a: Answer | one q: Question | a in q.answers
}


//++++++++++++ ASSERTIONS ++++++++++++

assert test1{ // all questions are from farmers
all q : Question | q.sender.role = Farmer
}
assert test2{ // all answers belong to exactly one question
all a : Answer | one q: Question | a in q.answers
}
assert test3{ // tip cannot be both liked and disliked by the same user
all t:Tip | no f: ForumUser | (f in t.liker) and (f in t.disliker)
}
assert test4{ // answers cannot be both liked and disliked by the same user
all a: Answer | no f: ForumUser | (f in a.liker) and (f in a.disliker)
}


//++++++++++++ EXECUTION PARAMS ++++++++++++
pred show{
#Zone <= 2
#ForumUser>= 3
#Question >= 2
#Answer >= 4
#Answer.liker > 1
#Answer.disliker > 1
}
run show for 50

//check test1 for 100
```


Example of a possible Instance with all the signatures



5. EFFORT SPENT

	S1	**S2**	**S3**	**Alloy**	**Meetings**
Careddu Gianmario	2h 30min	9h	4h	9h 30min	30h
La Greca Michele	7h 30min	12h 30min	15h	1h	30h
Zoccheddu Sara	2h30min	9h	16h30min	1h	30h

6. REFERENCES

1. **https://thefactfactor.com/facts/pure_science/biology/crops/2082/**
2. **https://en.wikipedia.org/wiki/Telangana**
3. **https://www.w3.org/TR/WCAG20/**
4. **Specification document: “A.Y. 2020-2021 Software Engineering 2 R&DD Assignment”**
5. **Alloy official documentation: https://alloytools.org/documentation.html**
6. **Paper: "Jackson and Zave: the world and the machine"**

