



RASD

Requirement Analysis and Specification Document

Alessandro Cecchetto (10865806)
Mattia Siriani (10571322)
Matteo Visotto (10608623)

Computer Science and Engineering
Software Engineering 2 course @ Politecnico di Milano

v1.0 - 23/12/2021



Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 4 |
| 1.1 | Purposes | 4 |
| 1.2 | Scope | 4 |
| 1.2.1 | Phenomena | 5 |
| 1.3 | Goals | 6 |
| 1.4 | Definition, acronyms, abbreviations | 6 |
| 1.4.1 | Definition | 6 |
| 1.4.2 | Acronyms | 8 |
| 1.4.3 | Abbreviations | 8 |
| 1.5 | Revision history | 8 |
| 1.6 | Reference documents | 9 |
| 1.7 | Document structure | 9 |
| 2 | Overall Description | 10 |
| 2.1 | Perspective | 10 |
| 2.1.1 | User Interfaces | 10 |
| 2.1.2 | Software Interfaces | 10 |
| 2.1.3 | Hardware Constraints | 11 |
| 2.2 | Product Function | 11 |
| 2.2.1 | Sign Up | 11 |
| 2.2.2 | Deviance | 11 |
| 2.2.3 | Notification | 12 |
| 2.3 | Actors | 13 |
| 2.3.1 | Visitor | 13 |
| 2.3.2 | User | 13 |
| 2.3.3 | Policy maker | 13 |
| 2.3.4 | Administrator | 13 |
| 2.4 | Assumptions, Dependencies, Constraints | 13 |
| 2.4.1 | Assumptions | 13 |
| 2.4.2 | Dependencies | 14 |
| 3 | Specific Requirements | 15 |
| 3.1 | Interface Requirements | 15 |
| 3.1.1 | User interfaces | 15 |
| 3.1.2 | Policy maker interfaces | 15 |
| 3.1.3 | Administrator interfaces | 16 |
| 3.1.4 | Forum interfaces | 17 |
| 3.1.5 | Homepage interface | 20 |
| 3.2 | Functional Requirements | 21 |
| 3.2.1 | Visitor scenarios | 21 |
| 3.2.2 | User scenarios | 26 |
| 3.2.3 | Policy maker scenarios | 35 |

| | | |
|----------|--------------------------------------|-----------|
| 3.2.4 | Administrator scenarios | 53 |
| 3.2.5 | Requirements | 61 |
| 3.2.6 | Traceability Matrix | 62 |
| 3.3 | Performance Requirements | 63 |
| 3.4 | Design constraints | 63 |
| 3.4.1 | Standards compliance | 63 |
| 3.4.2 | Hardware limitations | 63 |
| 3.4.3 | Privacy policies | 63 |
| 3.5 | Software system attributes | 63 |
| 3.5.1 | Easy usability | 63 |
| 3.5.2 | Reliability | 64 |
| 3.5.3 | Availability | 64 |
| 3.5.4 | Security | 64 |
| 3.5.5 | Modularity | 64 |
| 3.5.6 | Maintainability | 64 |
| 4 | Formal Analysis with Alloy | 65 |
| 4.1 | Alloy code | 65 |
| 4.2 | Worlds | 73 |
| 5 | Effort Spent | 76 |

Information

This project has been built following the indications of the stakeholders, referring to what is published in the official repository (<https://github.com/UNDP-india/Data4Policy>), taking into consideration, for some aspects, also the specification provided by the teachers in charge of the course.

1 Introduction

In the last years climate change has become an increasingly worrying topic because of its impacts on Earth's weather pattern. Extreme weather events are becoming more and more frequent: WMO (World Meteorological Organization) on its "State of Global Climate 2021 WMO Provisional report" pointed out that extreme events have increased in the last fifty years by five times, causing more than two million deaths and economic loss for around 3640 billions dollars. This is leading to the alternation of drought and floods more often and represents a serious threat to the agriculture sector. It is estimated that climate change will produce a 4%-26% loss in net farm income towards the end of the century.

Developing countries and the one with the higher population density like India are the most threatened: 58% of rural households depend on agriculture as the principal means of livelihood and the population growth leads to higher and higher food demand.

In order to better understand the relationship between agriculture and natural phenomena, different stakeholders need to obtain, store and analyze large amounts of data to find models that can help them in the development of the agricultural sector. This is why technology has to come to the rescue and understand how to do everything possible to avoid food shortages.

1.1 Purposes

The goal of Telengana's government is to design, develop and demonstrate anticipatory governance models for agriculture systems using digital public goods and community-centric approaches to strengthen data-driven policy making. DREAM is a digital platform that has the capability to visualize and analyze high resolution geospatial data, giving the opportunity to see the over time's changes that have happened to the agriculture ecosystem. The data reprocessed can help Policy makers to identify areas in which farms are working exceptionally well and to document the good practices in the forum provided, in order to create a knowledge network that can help other farms to do better.

1.2 Scope

To reach the goals required, Dream platform shall allow to aggregate and elaborate public data coming from different sources, alongside the need to create a network of knowledge.

It is fundamental that the platform is able to manage data of different types and it also needs to have the ability to elaborate them, using a proper algorithm called Deviance, in order to extract a ranking list according to predefined parameters. The platform shall also make the data collected publicly and freely accessible through the APIs or through a direct download.

The system shall have a section where registered users can communicate each other

and share their knowledge. Those tips shall be visible from every Visitor of the platform.

1.2.1 Phenomena

According to the paper "The World and the Machine" by M. Jackson and P. Zave, we can identify the application domains. The following table (Tab. 1) describes the world, shared and the machine phenomena, including the reference to which part controls the phenomena.

Table 1 Phenomena table

| Phenomena | Who controls it? | Is it shared? |
|--|------------------|---------------|
| Policy maker registration | M | Y |
| Policy maker login | W | Y |
| User registration | M | Y |
| User login | W | Y |
| Administrator login | W | Y |
| Check username, mail and password | M | N |
| User/Visitor select a filter | W | Y |
| Deviance computation | M | Y |
| Policy maker creates a new discussion on the forum | W | Y |
| Policy maker deletes a discussion from the forum | W | Y |
| Policy maker publishes a post on the forum | W | Y |
| Policy maker modifies a post on the forum | W | Y |
| Policy maker deletes a post from the forum | W | Y |
| Policy maker approves a post that is pending | W | Y |
| Policy maker declines a post that is pending | W | Y |
| Policy maker recalculate the Deviance | W | Y |
| User publish a post on the forum | W | Y |
| User modify its own post on the forum | W | Y |
| User deletes its own post from the forum | W | Y |
| Send notification by email | M | Y |
| Download of data with different formats | W | Y |
| Confirm registration | W | Y |
| Add a new data source | W | Y |
| Delete a data source | W | Y |
| Modify a data source | W | Y |

1.3 Goals

The main objectives of our system are the following:

- G1: Allow the system to calculate a Deviance using public data**

This is the main feature of the system. It uses different type of data (e.g. weather, production, soil ...) retrieved from multiple public data-sets, in order to compute a ranking divided in areas to find out conditions and factors that can be related to a good or bad agricultural production. This ranking is then controlled by Policy makers from which they can produce a report if necessary.

- G2: Allow the platform Administrator to decide which public data should be used in the Deviance computation**

Platform Administrators should decide which public data-set should be used in the system by adding/deleting them from their personal area.

- G3: Allow people to interact and build a knowledge network**

Policy maker's reports should be a good starting point to explain farmers the best techniques used in farming. An other fundamental aspect is the knowledge of the single farmer; for that reason sharing opinions is important as well as data. The system should provide a dedicated platform, administrated by the Policy makers, where registered users can write their experience. For that reason, the system should also let people to register freely.

- G4: Allow registered users to receive notification about activities of their interests in the knowledge network**

The system should let Users to receive notifications for all the discussion they have interacted with in order to let them up-to-date.

- G5: Allow Policy makers to release publicly their reports based on the Deviance result**

The system should let Policy makers to create new discussions in which they could publish reports. Discussions should be publicly visible and commentable only by registered users.

- G6: Allow Visitors to access data and documents publicly**

The system should be freely accessible by anyone for reading all contents available.

1.4 Definition, acronyms, abbreviations

1.4.1 Definition

- Visitors:** whoever is interested in retrieving the data collected by the project Dream. They can belong to any age and part of the world. Their purpose is to visit the site in order to retrieve some information regarding agricultural topic.

- **User:** it is a visitor who registers and then can interact with the forum.
- **Policy maker:** people who are involved in making policies and policy decisions, in this case regarding the agricultural development of some areas of India (relying on a ranking calculated by an internal service).
- **Administrator:** it is whoever administers the site. It can adds, deletes and modifies data sources.
- **Registered user:** User, Policy maker or Administrator are considered as Registered user when a distinction is not necessary.
- **Data format:** formats through which it is possible to retrieve data.
- **Notification:** it's an alert regarding a certain event that has occurred. This alert can be a new publication of a document in the forum, a new message written in the forum, etc.... This notification will be sent by email.
- **Actor:** User, Policy maker, Visitor or Administrator.
- **Identity Provider:** it is a system entity that creates, maintains, and manages identity information and also provides authentication services in order to relying applications within a federation or distributed network.
- **Service Provider:** it is a system entity that receives and accepts authentication assertions from an identity provider.
- **Farmer:** a person who owns or manages a farm. He/she can be a User or a Visitor depending whether it has an account or not.
- **Published:** it is a post that has already been approved by a Policy maker.
- **Policy maker ID:** it is a unique ID, assigned by third parties, associated to a specific Policy maker.
- **Data source:** an external service which provides data to the system.
- **Forum:** a website that provides an online exchange of information between people about a particular topic. It's the place where the Policy maker can start a new discussion (e.g. publishing a document about the trend of crops of a specific zone) and then every User can take part in this discussion.
- **Topic:** it is the top-level category in the forum post organization.
- **Discussion:** it is a place in the forum, created by a Policy maker, where the Users and the Policy makers can communicate. It represents the lowest categorization in the forum.
- **Reply:** an answer to a forum discussion by a registered user.

- **Post:** a reply. It is used when a specific context is not necessary.
- **API:** software interface that provides procedures to simplify software development and innovation by enabling applications to exchange data and functionality easily and securely.
- **Home Page:** it is the initial page of a site.
- **Deviance:** it is the algorithm that shows the ranking of the different zones, according to certain parameters.
- **Data Aggregator:** it is the component of the system that is responsible of receiving and analyzing data coming from external services.
- **Area:** it is a physical district of India.
- **Moderator:** person who manages the forum and decides which post should be published.

1.4.2 Acronyms

- **API:** Application Programming Interface, see definition above
- **IdP:** Identity Provider, see definition above
- **SP:** Service Provider, see definition above
- **UML:** Unified Modeling Language
- **BPMN:** Business Process Model and Notation

1.4.3 Abbreviations

- **ID:** identifier. It's a general unique sequence of numbers or letters in order to unambiguously identify an entity.
- **Gn:** goal number n.
- **Dn:** domain assumption number n.
- **Rn:** requirement number n.

1.5 Revision history

- v.1.0 - 23/12/2021 - Initial version

1.6 Reference documents

- Specification document: "R&DD Assignment A.Y. 2021-2021"
- Alloy official documentation: <https://alloytools.org/documentation.html>
- Paper: "Jackson and Zave: the world and the machine"
- Business Process Modeling Notation (BPMN), C. Cappiello, P. Plebani, M. Vitali
- Unified Modeling Language (UML) official specification: <https://www.omg.org/spec/UML/>
- State of Global Climate 2021 WMO Provisional report:
https://library.wmo.int/doc_num.php?explnum_id=10859
- Official Data4Policy stakeholder's project repository <https://github.com/UNDP-india/Data4Policy>

1.7 Document structure

- **Section 1: Introduction**

This section offers a brief description of the problem and required functionalities, also providing definition and acronyms that can be found in this document.

It also provides the revision history and the main structure of the document itself.

- **Section 2: Overall Description**

This section offers a summary description of the organization of the system, software constraints and the interfaces needed to get it work. It also contains a description of the features offered and the actor who use it.

- **Section 3: Specific Requirements** This section contains the description of functional requirements using some scenarios, use cases and diagram.

- **Section 4: Formal Analysis with Alloy** The last section which includes an analysis of the consistency of the model using Alloy language.

2 Overall Description

2.1 Perspective

Dream is a public domain, data aggregation platform and a knowledge network that provides all the functionalities described in section *Product Functions* (2.2).

It exploits an IdP to supply authentication and authorization in all its components, that can be easily integrated with standard external services and it also takes advantage of API interfaces to retrieve data needed for the elaboration.

2.1.1 User Interfaces

The system should interface with different actors through devices which must be connected to Internet. The interfaces are provided via web application, which is responsive and can be used from different type of devices (e.g. desktop and tablet).

2.1.2 Software Interfaces

The system allows the Administrator to dynamically add external APIs in order to retrieve data needed for the elaboration.

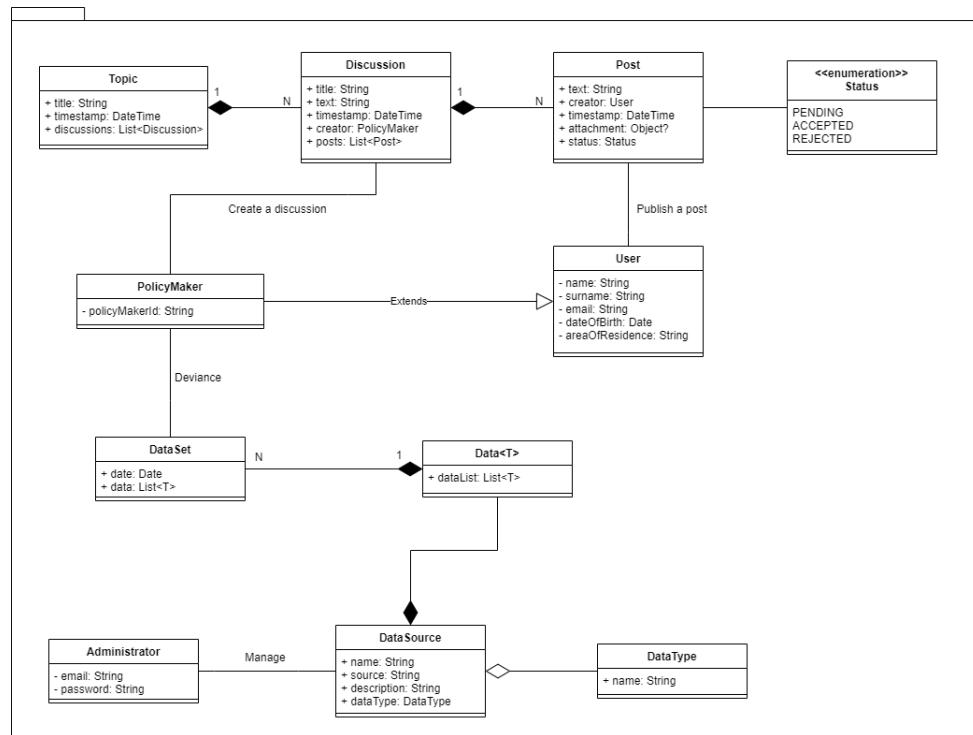


Figure 1: Class diagram

2.1.3 Hardware Constraints

Every person that wants to access the Dream service needs to have a device that features a browser and with an internet connection. Moreover the system is composed by a data-aggregator, on server side, that should have a stable and high speed internet connection (e.g. optic fiber).

2.2 Product Function

2.2.1 Sign Up

This functionality allows a Visitor to register in order to write in the Dream's forum. The first step is to insert all the requested credentials. If they are valid the system will send an email to the inserted address, asking the User to click on a link, in order to validate its account. At the end of the process the new User is taken to the login page.

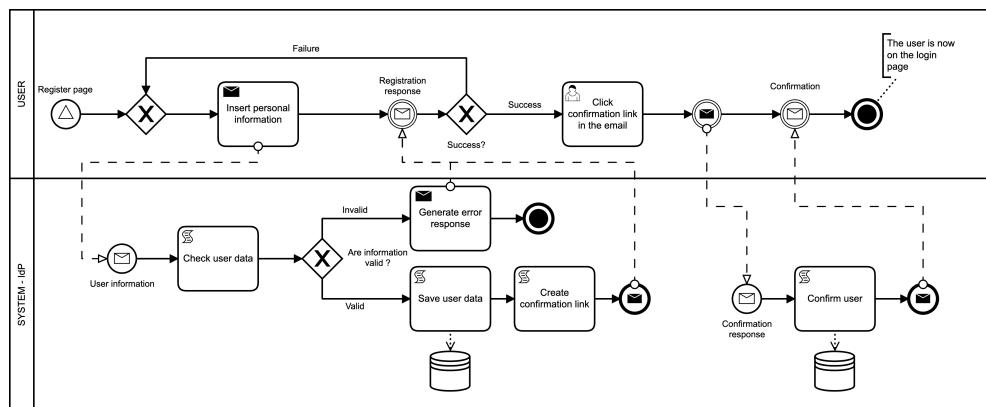


Figure 2: Sign Up BPMN

2.2.2 Deviance

This functionality is accessible only to the Policy makers. By default the Deviance is going to be calculated by the system itself, according to predefined parameters. The Policy maker has the capability to make the system recalculate the Deviance, modifying the considered parameters. To do so he should go to his reserved area and click on the Deviance button, after that, he can select the different parameters from a list and then recalculates the Deviance.

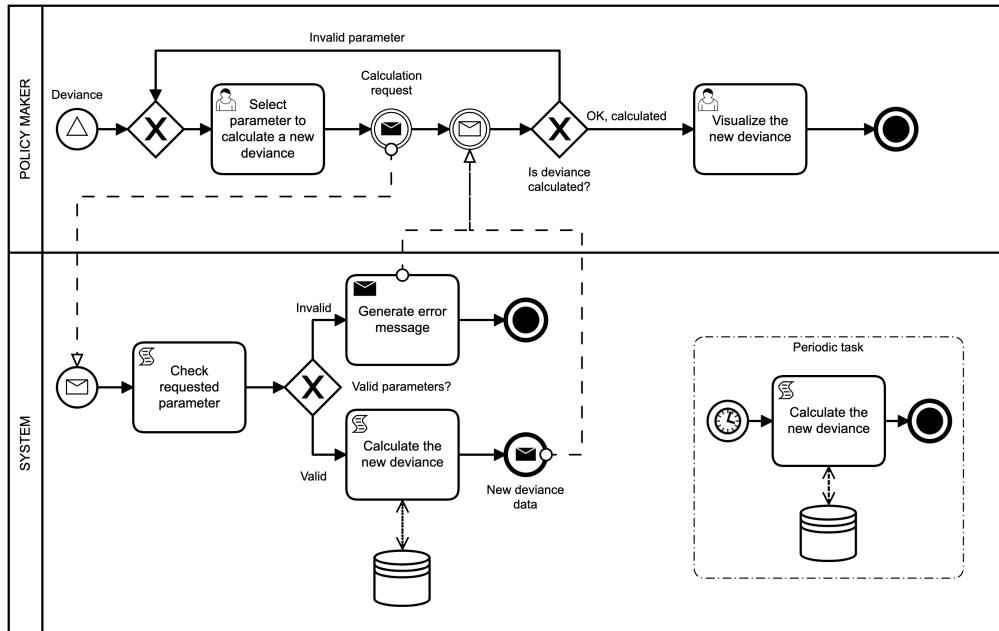


Figure 3: Deviance BPMN

2.2.3 Notification

This functionality lets the system send a notification to a User when his post gets accepted or rejected by a Moderator. Also, a notification is sent when a new post is published in a discussion in which an User or Policy maker has interacted with.

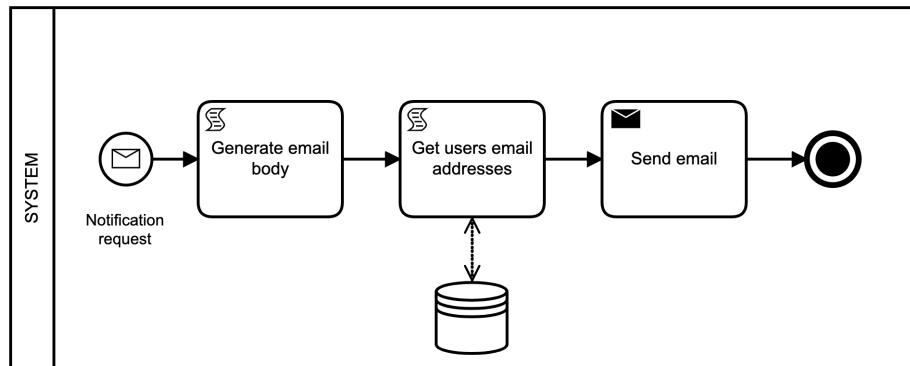


Figure 4: Notification BPMN

2.3 Actors

2.3.1 Visitor

Whoever has accessed the site, without having an account. He can freely access the data, filter them, download them and he can also visit the forum and read all the discussions in it. The only limitation he has is that he can't write in the forum.

2.3.2 User

A Visitor that has registered to the site and is logged in. In addition to all the functions of the Visitor he can also write replies in the forum.

2.3.3 Policy maker

A person who is in charge of building up the knowledge network, according to a ranking list. He can register to the site only if he owns a unique identifier (the Policy maker ID), that associates his real identity to the digital one. He can retrieve the ranking list (and also “forge” another one) in his private area. In addition every Policy maker has the role of moderator in the forum.

2.3.4 Administrator

The administrator of the Dream's site. He is the one who manages the data sources.

2.4 Assumptions, Dependencies, Constraints

2.4.1 Assumptions

- **D1:** The registration of a User in the IdP doesn't imply the User registration in the SP.
- **D2:** The roles used by SPs are already correctly mapped by the IdP.
- **D3:** A new discussion can be created only by a Policy maker.
- **D4:** Users can modify only their posts.
- **D5:** Policy makers are forum moderators.
- **D6:** Policy makers can modify all the posts.
- **D7:** The administration role could be given only by another Administrator.
- **D8:** The Policy maker ID is given by third parties.
- **D9:** The Policy maker ID is also used as personal API KEY.

2.4.2 Dependencies

Administrator's accounts are the only ones who can manage the system. They can be created only from another Administrator, so an internal service is sufficient to manage this group of accounts.

Different types of User will access the forum, with different permissions and roles. For this reason it will be a good choice letting Users register in other identity providers to login using their already existing account so, an external authentication system is required. Meanwhile, Policy makers are internal user too but they can also interact with the forum. For that reason, it is also needed that the system act like an Identity Provider and, at this point, Users can register using the same IdP that Policy maker relies on.

Shibboleth is the most used service in academic field. It is structured to base the authentication process using a LDAP directory. For this reason, the user data needs to be saved in OpenLDAP. LDAP presents specific application to manage directory content and it relies on its own internal administrator.

This technology has to be considered as an external component in the project due to the fact that could also be used as a service in the cloud.

3 Specific Requirements

3.1 Interface Requirements

3.1.1 User interfaces

In Figure 5 we can see the registration page. If a Visitor doesn't already have an account he can decide to create a new one, by inserting all the required data (name, surname, birthday, area of residence, email and password). Policy makers who wants to register, need to click on the "Policy maker" tab and after clicking on it, a new field will appear asking to insert the Policy maker ID.

The figure displays two side-by-side registration forms, labeled (a) and (b).

(a) User: This form is for regular users. It features tabs at the top labeled "USER" and "POLICY MAKER". The "USER" tab is currently selected, indicated by a blue underline. The form contains fields for Name, Surname, Birthday, Area of residence (with a dropdown arrow), Email, and Password. At the bottom is a blue "SIGN UP" button, and below it are links for "Already have an account?" and "Sign in here".

(b) Policy maker: This form is for policy makers. It also has "USER" and "POLICY MAKER" tabs at the top, with "POLICY MAKER" selected (blue underline). In addition to the fields in the user version, it includes a "Policy Maker ID" field. The layout is identical to the user version, with a blue "SIGN UP" button at the bottom and links for "Already have an account?" and "Sign in here".

Figure 5: Register pages

3.1.2 Policy maker interfaces

The Policy maker reserved area interface is composed by two different columns: on the left there is the menu with a list of the different section. Once selected the section of interest, the right part of the page renders the corresponding view. In Figure 6 we can see the "Deviance" section: a ranking list of areas is represented alongside the checkbox list in which the Policy maker can choose the parameters that will be taken into account when a new Deviance is recalculated.

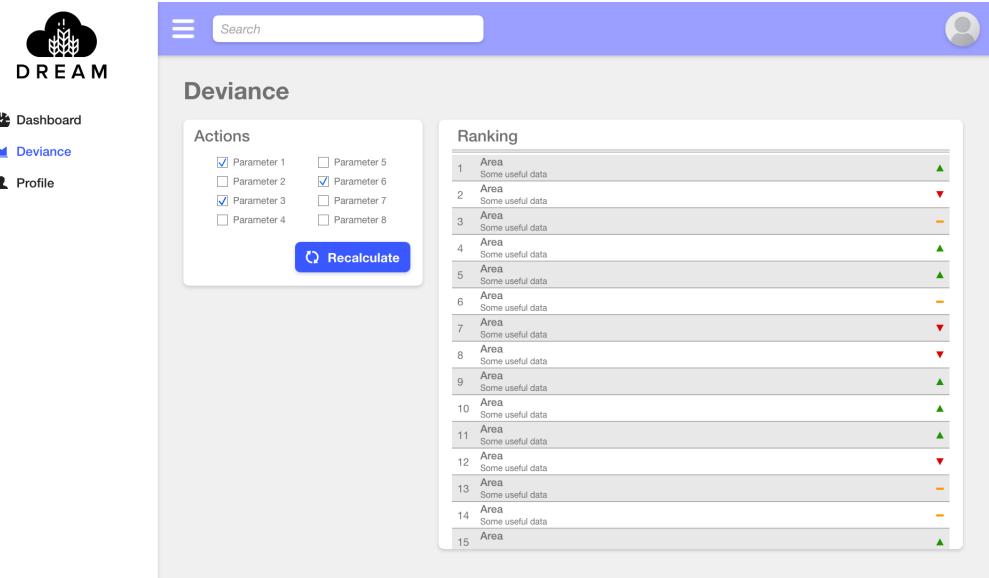


Figure 6: Deviance page

3.1.3 Administrator interfaces

The Administrator reserved area (Figure 7) interface is composed by two different columns: on the left there is the menu with the list of the different sections (similar to the Policy makers' one). In this case the "Data Sources" page is selected and a table with all the active sources in the system is visible. For each row two buttons are available: the first one is a gear and it is used to display the configuration of the corresponding data source, while the second one, is a red trash bin and if it's clicked it lets the Administrator to delete the corresponding data source. Finally on the top right part of the internal view there is a button that lets the Administrator to add a new data source.

| # | Data source | Actions |
|---|---|---------|
| 1 | Open data 1 https://api.opendata1.org/api/get_data | |

Figure 7: Data sources

3.1.4 Forum interfaces

The Forum’s home page is like in Figure 8 and it’s divided in three columns. The first one represents the menu, in which it’s presents the “Home” tab (containing a list of all the topics) and the “Explore topics” tab (containing random discussion belonging to different topics). Since in the figure it’s selected the “Home” tab, we can see in the central column a list of all the topics present in the forum. Finally, in the third column are present two boxes: the first one contains a list of the most active User and the second one holds different quick links. Clicking a topic, we will be redirected to a new page containing a list of all the discussion contained in the selected topic (Figure 9) and clicking on a discussion we will be redirected to the selected one, from which we can also see the complete text of the discussion and all its replies (Figure 10).

The screenshot shows the DREAM forum homepage. At the top, there is a header with the DREAM logo, a search bar, and 'Sign Up' and 'Sign In' buttons. Below the header is a menu bar with 'Home' selected. The main content area displays five topic cards, each with a folder icon, a title 'Topic title', and a long placeholder text block. To the right of the topics is a sidebar titled 'Top User' showing five user entries with names like 'Username' and scores like '120'. At the bottom right of the sidebar are links for 'Help', 'About', 'Contact', 'Information', 'Topics', and 'Top topics'.

Figure 8: Forum homepage (without login)

This screenshot shows the same forum homepage as Figure 8, but with a user logged in. The top right corner shows a profile picture and the text 'Hi, Username Policy Maker'. The menu bar now includes 'My discussions', 'My replies', and 'Profile'. The sidebar on the right also includes a 'New discussion' button. The topics displayed are identical to Figure 8, and the 'Top User' sidebar remains the same.

Figure 9: Discussions inside a topic

The screenshot displays a user interface for a platform named "DREAM". At the top, there is a navigation bar with a search bar and a user profile icon labeled "Hi, Username Policy Maker". On the left, a sidebar titled "Menu" includes links for "Home", "Explore topics", "My discussions", "My replies", and "Profile". The main content area shows a discussion titled "Discussion title" by "Name Surname" (Policy Maker, Member since: 2021). The discussion content is placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. In mollis venenatis quam nec feugiat. Sed scelerisque pretium orci eget sodales. Fusce ullamcorper nec odio sed congue. Suspendisse aliquet finibus lectus a ullamcorper. Cras eu pretium ligula, vehicula laculis ex. Integer molestie risus arcu, bibendum rutrum arcu malesuada eu. Vivamus eu odio eu urna sodales tempus sit amet eget libero. Vestibulum dapibus sapien ut mi rhoncus, ac elementum turpis aliquet. Sed at pharetra ipsum. Aenean arcu nibh, faucibus eget felis vel, suscipit pulvinar mi. Phasellus id placerat tortor. Aliquam bibendum ihsa ligula. Cum nibh sollicitudin leo a eleifend incidentum. Nulla blandit, elit ut condimentum convallis, augue magna molestie libero, et placerat libero mauris sed orci. Curabitur fermentum volutpat odio quis varius. Duis at ipsum vel tortor semper pulvinar ut metus. Maecenas ac turpis pharetra, gravida felis tristique, lacinia dolor. Mauris lacreet eu ex eu vestibulum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilla curae; Sed tristique justo vitae massa porttitor, pharetra porttitor libero pellentesque.". Below the discussion are two reply buttons: "Report 20211223 v1.0" and "Data 20211223 v1.0". To the right of the discussion is a "Reply" button. Further down, there are two more reply sections with similar placeholder text and user profiles. On the far right, there is a sidebar with a "New discussion" button and a "Top User" section listing five users (Username, You) with their scores (120). Below the top user section are links for "Help", "About", "Contact", "Information", "Topics", and "Top topics".

Figure 10: Content of a discussion

3.1.5 Homepage interface

We can also see in Figure 11 the home page of the site, where the Visitor could access the data or Sign up/Sign in. In this page, if a Policy maker is already logged in, it's present a button from which the Policy maker can reach his Reserved Area.



Figure 11: DREAM Homepage

3.2 Functional Requirements

3.2.1 Visitor scenarios

Scenario 1

Prabaker, a farmer from Telangana, wants to know how is the trend of agriculture in his country and for this reason he wants to have access to data about this. A friend of him, Sundar, suggested him to visit Dream's web page, a project by the United Nations of India, giving him the link. Prabaker opens the link and connects to the site's home page, where he can access different data set. Once he clicks on the button "Access the data" he notices that there's the possibility to select different filters to see information coming from the various districts of Telangana. He decides to select the filter that shows how much rain has fallen in the last 24 hours in the zone of his land holding because in the monsoon season is important to keep track of it in order to not waste the crop. The page is rendered and he can see the updated data on screen.

Scenario 2

Matteo, a researcher, needs to analyze soil quality data of the Telengana region about last month for his study in resource efficiency for crops in developing countries. He learns about project Dream that allows free access to all the resources he needs for his study. So Matteo connects to the Dream home page and then he selects the button to access the data. After doing that a new page is rendered and selecting a data set and scrolling down on it, he notices the possibility to download the data in different formats. He selects the one he suits the most and clicks on the button download.

Use case diagram

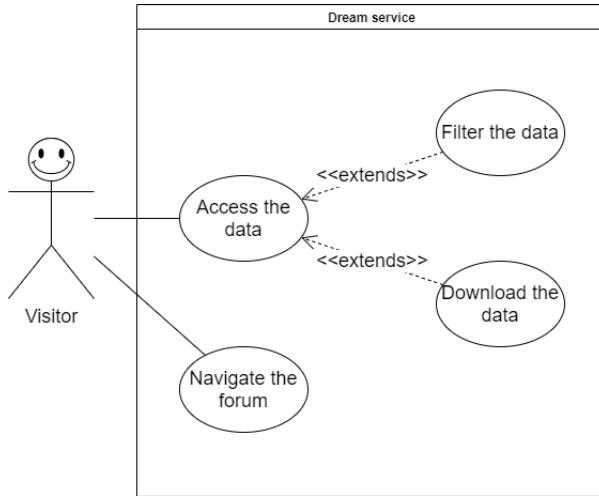


Figure 12: Use case diagram

Use case tables

Table 2 Filter data

| | |
|------------------------|---------------------------------------|
| ID | 1 |
| Name | Filter data |
| Actor | Visitor |
| Entry condition | The Visitor is in the Dream home page |

| | |
|-----------------------|--|
| Events flow | <ul style="list-style-type: none"> The Visitor clicks on the "Access the data" button The system displays the available data. The Visitor searches for the data set of interest The Visitor selects the “Filter” button on the data set he wants to filter The System pops up a list of different filters that can be applied to do a more precise search The Visitor select all the filters he wants to apply The Visitor clicks on the “Search” button The System search applying the selected filters |
| Exit condition | The new search is displayed |

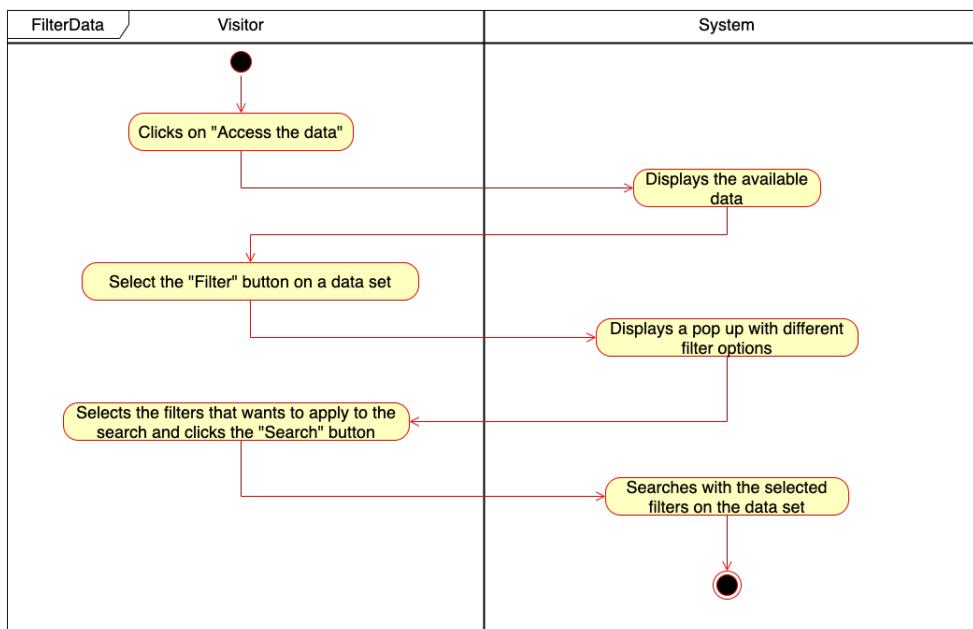


Figure 13: Filter data

Table 3 Download data set

| | |
|------------------------|---|
| ID | 2 |
| Name | Download data set |
| Actor | Visitor |
| Entry condition | The Visitor is in the Dream home page |
| Events flow | <ul style="list-style-type: none">• The Visitor clicks on the "Access the data" button• The system displays the available data.• The Visitor search for the data set of interest• The Visitor select the “Download” button on the data set he wants to download• The System pops up a data panoramic showing the different formats in which the data can be downloaded• The Visitor select the data format he prefers• The Visitor clicks on the “Download” button• The System starts the download |
| Exit condition | The data set is downloaded |
| Output | Data set |

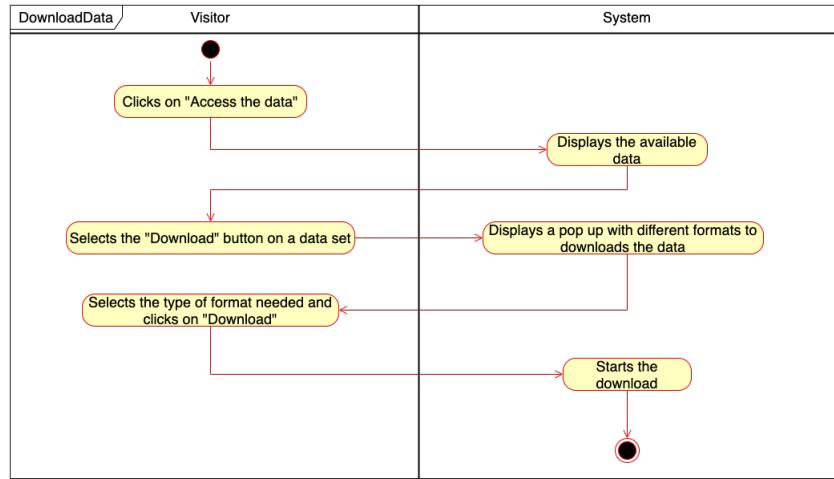


Figure 14: Download data set

Table 4 Navigate the forum and download a document

| | |
|------------------------|---|
| ID | 3 |
| Name | Navigate the forum and download a document |
| Actor | Visitor |
| Entry condition | The Visitor is in the Dream home page |
| Events flow | <ul style="list-style-type: none"> • The Visitor click on "Open the forum" button • The Visitor is redirected in the forum home page where all the topics are visible • The Visitor click on the topic of his interest • The System displays all the discussion related to the chosen topic • The Visitor select the discussion he is interested in • The system displays the discussion published by the Policy maker with all the replies from different users. • The Visitor click and start the download of the attachment contained in the Policy maker's relation. |
| Exit condition | The visitor download the document |
| Output | Attachment |

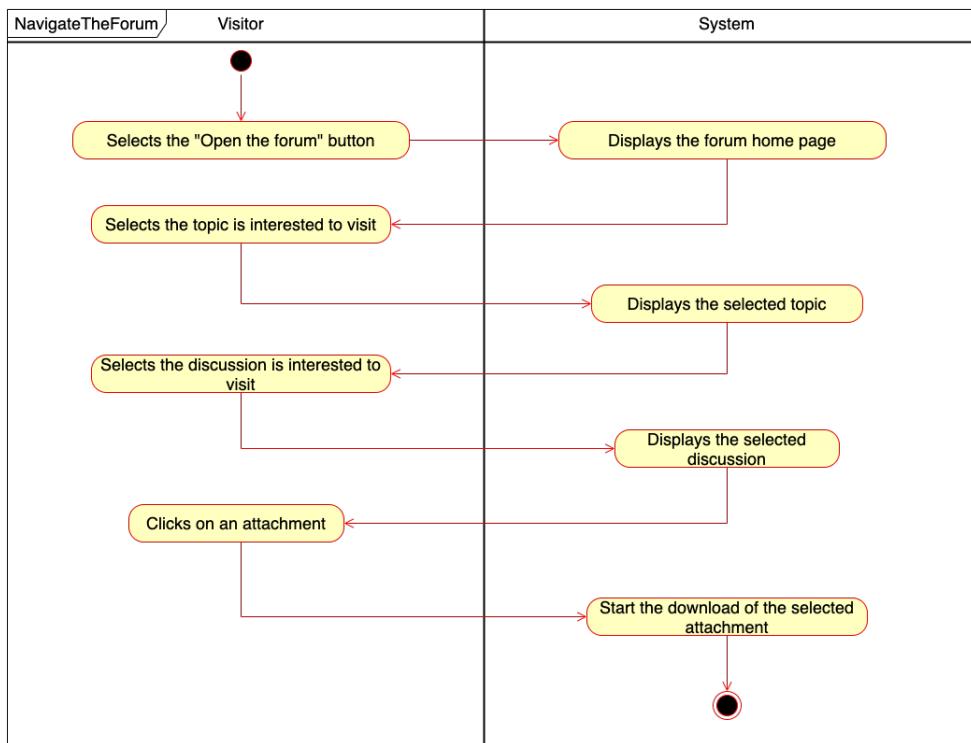


Figure 15: Navigates the forum and download a document

3.2.2 User scenarios

Scenario 3

Jishnu, a Telengana's farmer, finds out about Dream forum and wants to read some posts. He clicks on the link found using a browser and selects his region, finding out the last review document published by a Policy maker. Once he has read the discussion he decides to make his contribution and shares the agricultural techniques he uses. Jishnu hasn't got an account yet, but he needs to be registered to post on the forum. For this purpose, he clicks on the "Sign up" button on the top right and he inserts his data in the following page, including the region in which he lives, his personal email address and eventually he chooses a safe password to protect his new account. Once he clicks on the confirmation button, the system tells him to check the email he has used to register to conclude the registration process. He opens his mail looking for a mail from Dream containing a confirmation link. When he clicks on it a browser page pops up containing a check message which confirms that the registration ended correctly. Now he can login from the home page, go to the page he was visiting and add his comment.

Scenario 4

Basant is an active User of the forum of the site Dream. Unluckily, he discovers that the last post he published on a discussion contains a file that is not the file he wants to publish. So he returns to that post, clicks on the modify button and replaces the file with the correct one. After doing this procedure he clicks the confirmation button and then the changes to his post go to the pending approval list of the system.

Use case diagram

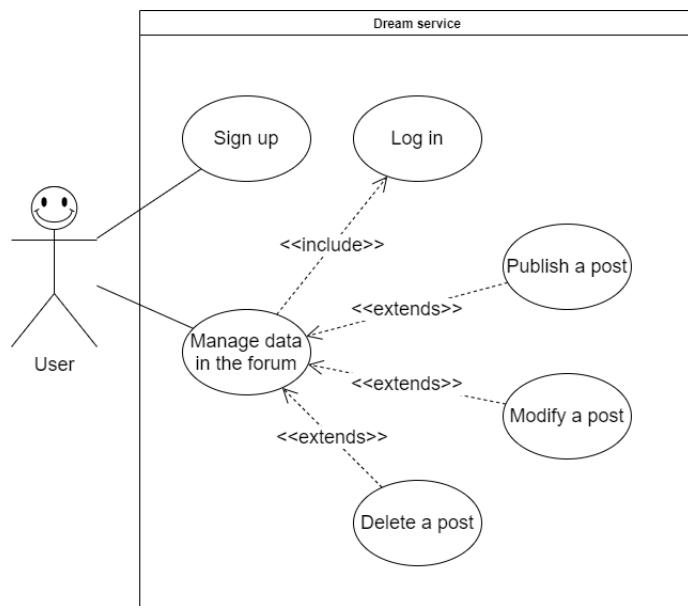


Figure 16: Use case diagram

Use case tables

Table 5 Sign Up User

| | |
|------------------------|--|
| ID | 4 |
| Name | Sign up User |
| Actor | User |
| Entry condition | User has reached the site |
| Input | Personal data, email and password to use for the registration |
| Events flow | <ul style="list-style-type: none"> • The site displays the “Sign up” button on the top right of the screen • User clicks on “Sign up” • The site displays a new page containing blank fields where user has to insert his data: name, surname, date of birth, area of residence, email and password • User inserts the mandatory data and accepts the “Terms of Service” • User confirms by clicking the confirmation button • The page shows a message inviting User to visit his email address in order to conclude the registration • User opens his inbox, find the email from Dream and clicks on confirmation link • The site displays a confirmation message of successful registration |
| Exit condition | User registration has been successful: the inserted data are stored in the database of the system. Now User can login using his credentials and post in the forum |
| Output | Registration data are stored in the database of Dream site. |

| Exception | <ul style="list-style-type: none"> User inserts non valid data(e.g. wrong email format or nonexistent area). The application displays an error message telling the User that he must check the data inserted and correct the invalid ones User inserts an e-mail which is already stored in the database. So, after user inserts his data and clicks on confirm, the application displays an error message telling User that he's already registered to the service and invites him to login with that email |
|-----------|--|
|-----------|--|

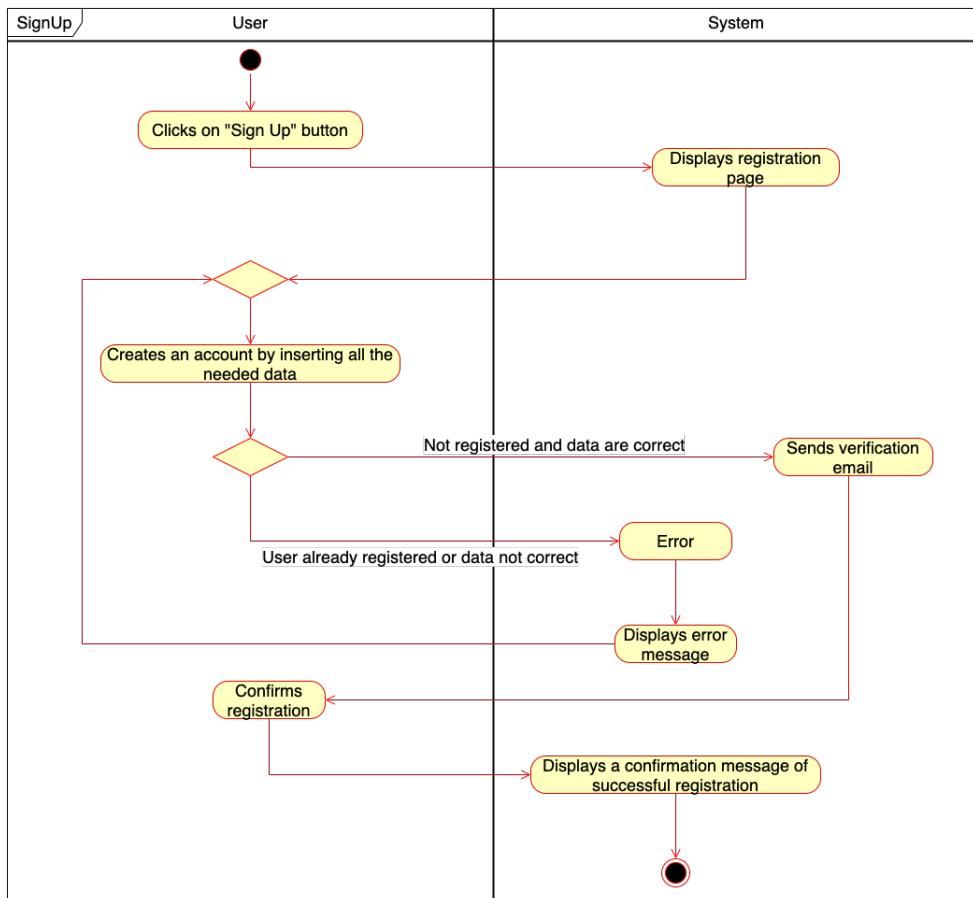


Figure 17: Sign Up User

Table 6 Login User

| | |
|------------------------|---|
| ID | 5 |
| Name | Login User |
| Actor | User |
| Entry condition | <ul style="list-style-type: none">• User has reached the site• User is already registered to the platform |
| Input | User email and password associated to a successful registration |
| Events flow | <ul style="list-style-type: none">• The site displays the “Sign In” button on the top right of the screen• User clicks on “Sign In”• The system displays the login page• User fills the username (email) and password fields using the credential inserted during the registration• System checks the validity of the credentials inserted• The system displays the precedent page or, if unavailable, the home page of the site |
| Exit condition | User is logged in |
| Exceptions | User inserts wrong credentials and clicks on “login” button. The system shows an error message inviting the User to check the credentials before trying again to login |

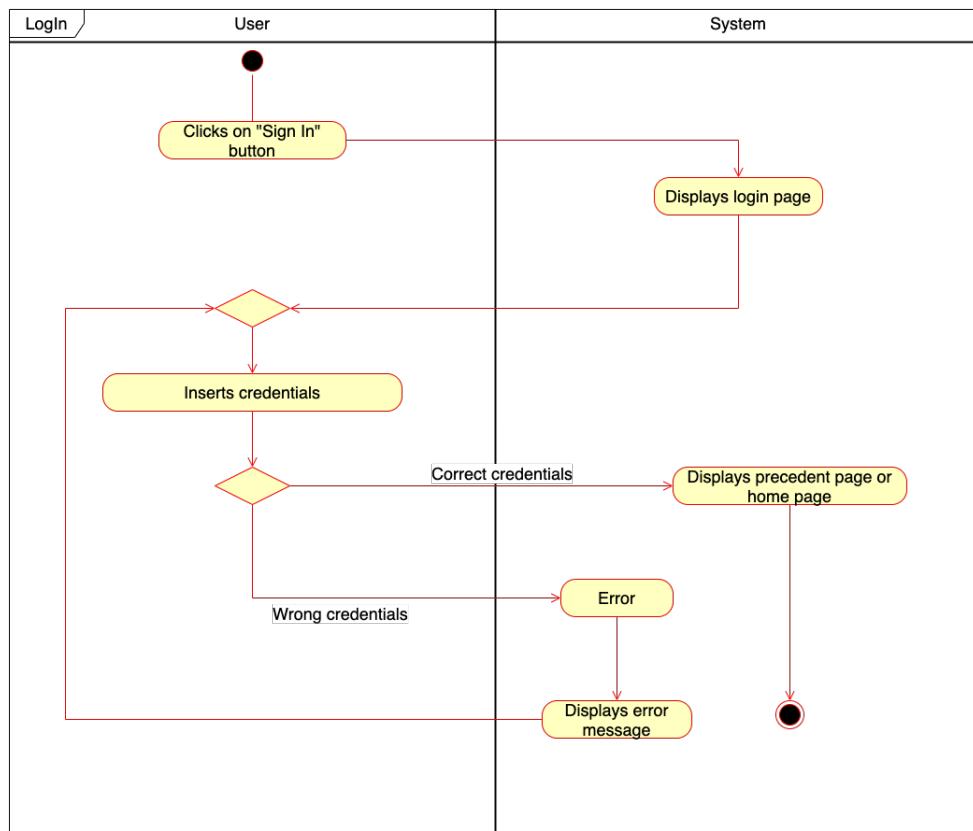


Figure 18: Login User

Table 7 Publish a post by User

| | |
|------------------------|--|
| ID | 6 |
| Name | Publish a post by User |
| Actor | User |
| Entry condition | <ul style="list-style-type: none"> • The User is already registered in the system • The User is already logged in the system • The User is in the forum home page |
| Input | Post |

| | |
|-----------------------|---|
| Events flow | <ul style="list-style-type: none"> The User selects the section of his interest The User selects the discussion about the argument of interest The User inserts a new answer in the form The User confirms the answer by clicking the confirmation button |
| Exit condition | The answer is in pending approval |
| Exceptions | The User tries to insert invalid contents (invalid characters, invalid file format, etc...) |

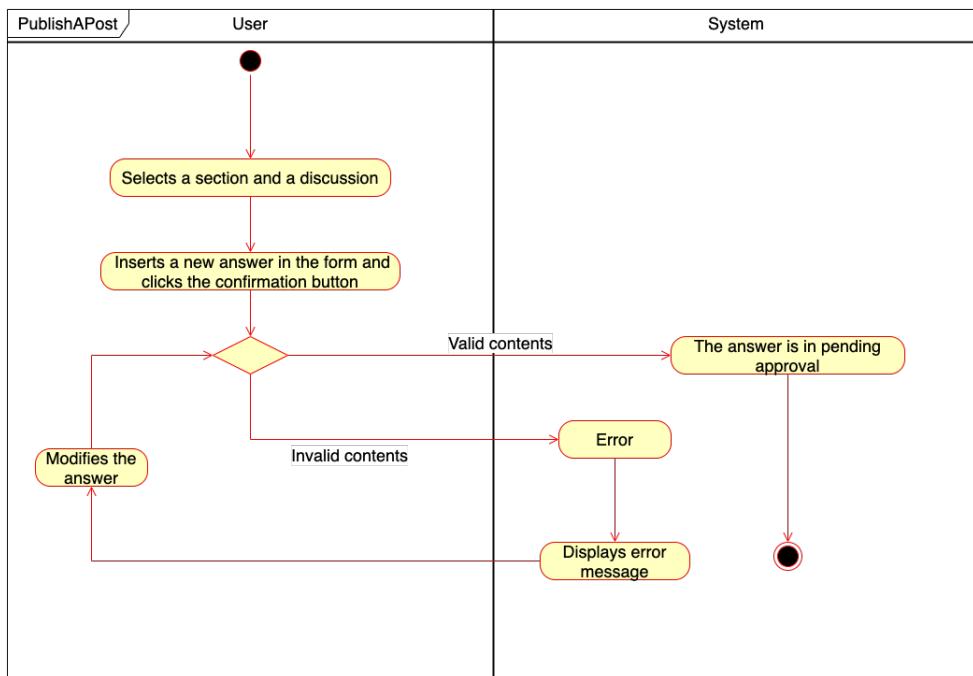


Figure 19: Publish a post by User

Table 8 Modify a post by User

| | |
|-----------------------------|--|
| ID | 7 |
| Name | Modify a post by User |
| Actor | User |
| Entry condition | <ul style="list-style-type: none"> • The User is already registered in the system • The User is already logged in the system • The User is in the discussion page where the post is present • The User has already published the post |
| Input | Post |
| Events flow | <ul style="list-style-type: none"> • The User press on the “Modify” button on the selected post • The System returns to the User the edit page • The User introduces the changes in the post • The User confirms the changes by clicking the confirmation button |
| Exit condition | The changes to the post are in pending approval |
| Exceptions | The User tries to insert invalid contents (invalid characters, invalid file format, etc...) |
| Special requirements | The User can modify only post written by his account |

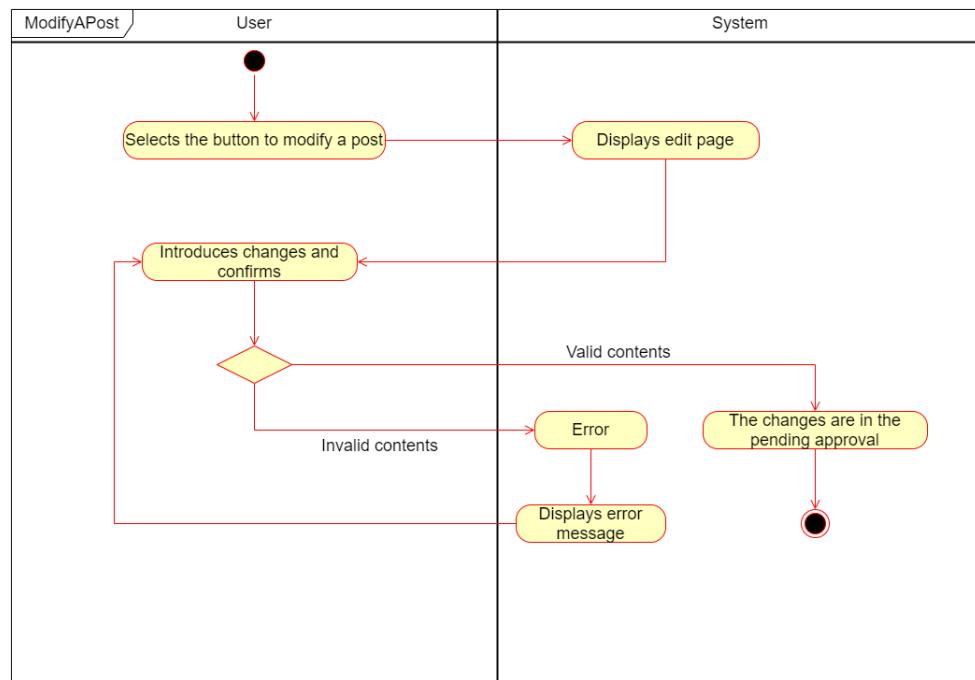


Figure 20: Modify a post by User

Table 9 Delete a post by User

| | |
|------------------------|---|
| ID | 8 |
| Name | Delete post by User |
| Actor | User |
| Entry condition | <ul style="list-style-type: none"> The User is already registered in the system The User is already logged in the system The User is in the discussion page where the post is present The User has already published the post |
| Input | Post |

| | |
|-----------------------------|--|
| Events flow | <ul style="list-style-type: none"> The User press on the “Delete” button on the selected post The System returns to the User a confirmation message The User confirms the operation by clicking the confirmation button |
| Exit condition | The post is deleted from the forum |
| Exception | The User doesn't confirm the operation |
| Special requirements | The User can delete only post written by his account |

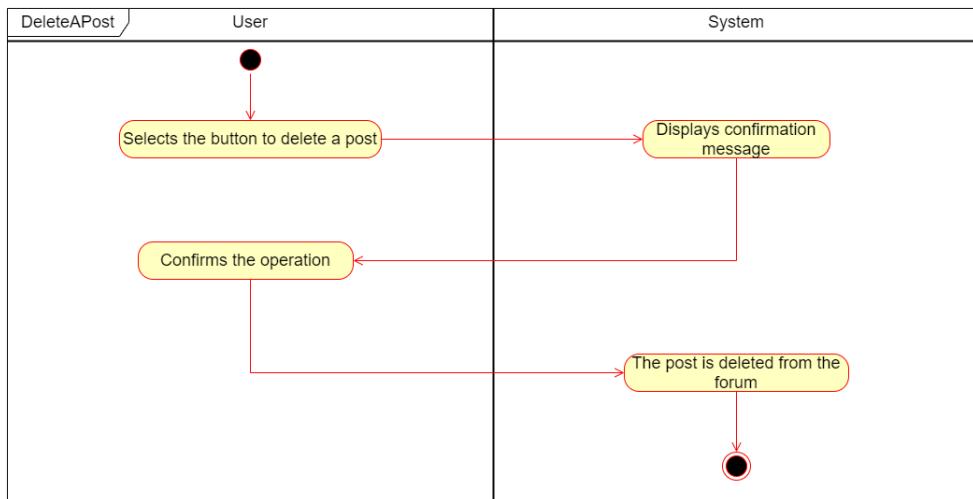


Figure 21: Delete a post by User

3.2.3 Policy maker scenarios

Scenario 5

Gurleen, a Telengana's Policy maker, needs to visualize data about agricultural harvest quality. Gurleen connects at Dream's site and lands at the home page. She clicks a button in the top right of the screen with the words “Sign In”. Gurleen is already registered to the service, so she authenticates digitizing her email and password and accesses to her reserved area. From this screen she accesses the Deviance section and from that point she can visualize a ranking list, established by Dream project's service, of the different areas. At this point, she identifies the

areas with the best score and publishes a document, publicly visible, where she asks to the farmers to present a report about their cultivation techniques, that will be published in the forum.

Scenario 6

Digamber, another Policy maker from the same area of Gurleen, checks his email and notice a notification from Dream about a new publication in a discussion that he has created. Clicking on the link from the email he opens the correspondent forum discussion. Once he has read Gurleen's replies he notice that he can't answer because he isn't logged in. So he clicks the "Sign In" button in the top right and he fills the fields with his username and password to authenticate. In this way he can write his replies and publish it in the discussion.

Scenario 7

Bipin is a Telengana's Policy maker. His boss asked him to check the efficiency of agricultural production in the district of Medak. Bipin goes to the Dream site home page and since is the first time he's assigned this job he has firstly to register to the portal. He clicks on the "Registration" button on the top right and inserts his data in the following page. He's asked for his personal code (Policy maker ID) which identifies him as Policy maker and eventually he confirms data insertion. A mail is send to the email address used during the registration with a confirmation link that once clicked consent to activate the new account. Bipin clicks on it and land on a confirmation page.

Scenario 8

Akash is a Policy maker of Telengana and needs to manage the forum of the Dream's site. Akash then connects to the home page of the Dream's site and since he is already logged in, he can go directly on the forum page, by clicking a button present on the home page. When he reaches the forum, Akash clicks on the "Moderator area", then he selects the pending list option and from that moment he can see a pending list with all the posts that required approval before being published in the forum. Akash decides to open the first post, examine it and then he approves it to be published; after the confirmation by Akash, the system automatically publishes the post in the forum and sends a notification to the author of the post.

Scenario 9

Parag is a Policy maker already registered in Dream's site. Today he needs to recalculate the Deviance according to certain parameters, told by his boss. So since he is already logged in, he can go directly to his reserved area, by clicking a button present on the home page. After doing that he selects the Deviance section and from that moment he can see the actual ranking list, calculated by default by the

system, and a list of parameters that he can select in order to recalculate a custom Deviance. He selects on three different parameters and then clicks on recalculate. The system recalculates the Deviance and then reloads the contents, showing the new custom ranking list.

Use case diagram

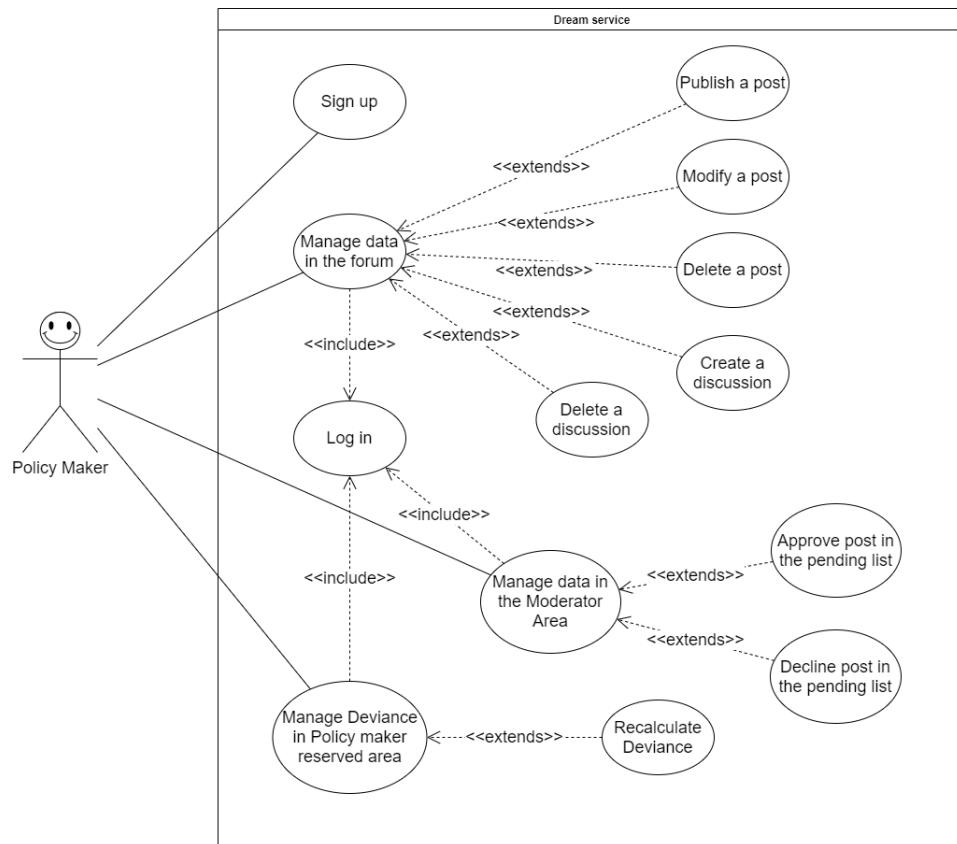


Figure 22: Use case diagram

Use case tables

Table 10 Sign Up Policy maker

| | |
|------------------------|--|
| ID | 9 |
| Name | Sign up Policy maker |
| Actor | Policy maker |
| Entry condition | Policy maker has reached the site |
| Input | Personal data, Policy maker ID, email and password to use for the registration |
| Events flow | <ul style="list-style-type: none"> • The site displays the “Sign Up” button on the top right of the screen • Policy maker clicks on “Sign Up” • The site displays a new page containing blank fields where user has to insert his data: name, surname, Policy maker ID, date of birth, area of residence, email and password • Policy maker inserts the mandatory data and the Policy maker ID • Policy maker confirms by clicking the confirmation button • The page shows a message inviting Policy maker to visit his email address in order to conclude the registration • Policy maker opens his inbox, find the email from Dream and clicks on confirmation link • The site displays a confirmation message of successful registration |
| Exit condition | Policy maker registration has been successful: the inserted data are stored in the database of the system. Now Policy maker can login using his credentials and post in the forum |
| Output | Registration data are stored in the database of Dream site. |

| | |
|-------------------|---|
| Exceptions | <ul style="list-style-type: none"> • Policy maker inserts non valid data (wrong date format or nonexistent area). The application displays an error message telling the Policy maker that he must check the data inserted and correct the invalid ones. • Policy maker inserts an email which is already stored in the database. So, after user inserts his data and clicks on confirm, the application displays an error message telling User that he's already registered to the service and invites him to login with that email • Policy maker inserts a non valid ID • Policy maker inserts data non corresponding to the ID |
|-------------------|---|

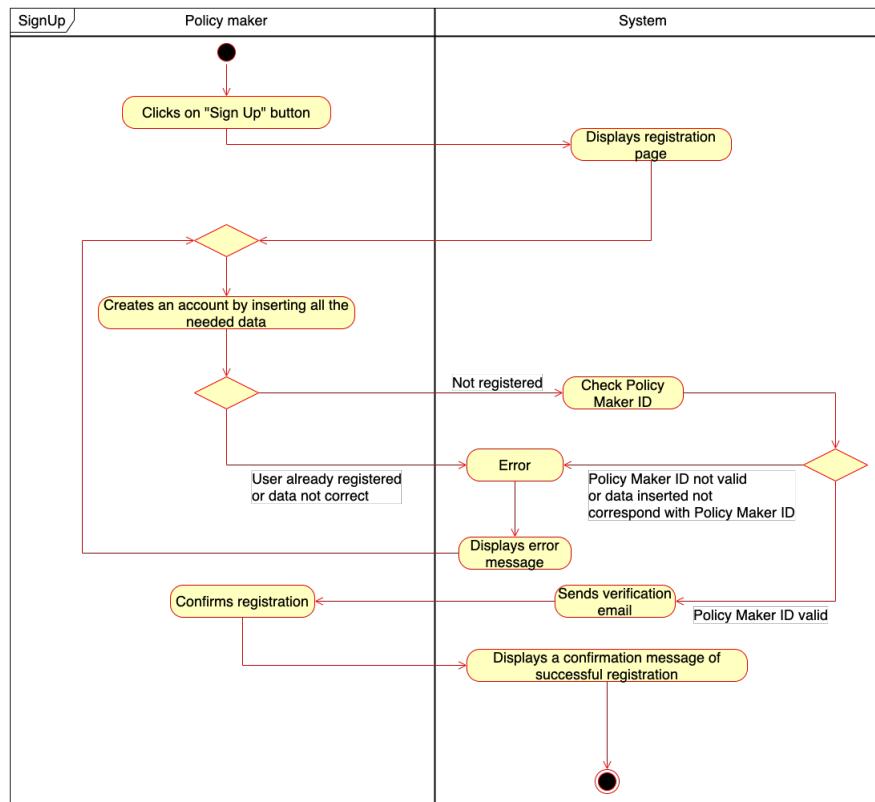


Figure 23: Sign Up Policy maker

Table 11 Login Policy maker

| | |
|------------------------|---|
| ID | 10 |
| Name | Login Policy maker |
| Actor | Policy maker |
| Entry condition | <ul style="list-style-type: none">• Policy maker has reached the site• Policy maker is already registered to the platform |
| Input | Policy maker email and password associated to a successful registration |
| Events flow | <ul style="list-style-type: none">• The site displays the “Sign in” button on the top right of the screen• Policy maker clicks on “Sign In”• The system displays the login page• Policy maker fills the username (email) and password fields using the credential inserted during the registration• System checks the validity of the credentials inserted• The system displays the precedent page or, if unavailable, the home page of the Policy maker dashboard |
| Exit condition | Policy maker is logged in |
| Exceptions | Policy maker inserts wrong credentials and clicks on login button. The system shows an error message inviting the Policy maker to check the credentials before trying again to login |

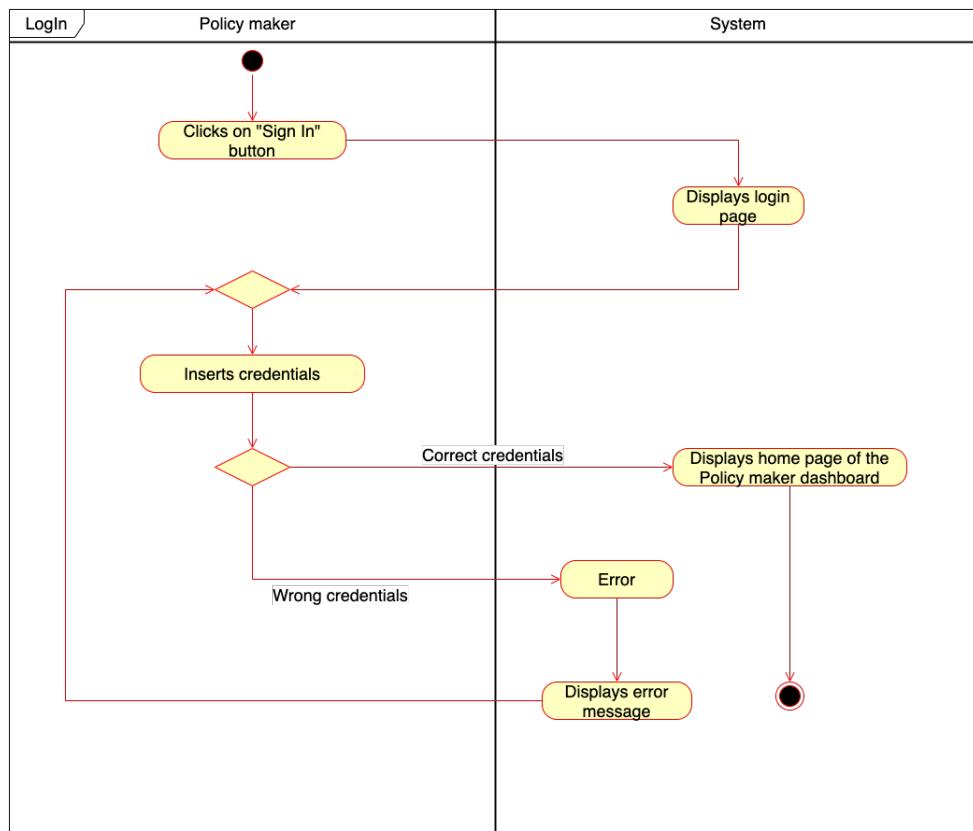


Figure 24: Login Policy maker

Table 12 Publish a post by Policy maker

| | |
|------------------------|--|
| ID | 11 |
| Name | Publish a post by Policy maker |
| Actor | Policy maker |
| Input | Post |
| Entry condition | <ul style="list-style-type: none"> • The Policy maker is already registered in the system • The Policy maker is already logged in the system • The Policy maker is in the forum home page |

| | |
|-----------------------|---|
| Events flow | <ul style="list-style-type: none"> The Policy maker selects the section of his interest The Policy maker selects the discussion about the argument of interest The Policy maker inserts a new answer in the form The Policy maker confirms the answer by clicking the confirmation button |
| Exit condition | The answer is published |
| Exceptions | The Policy maker tries to insert non valid contents (invalid characters, invalid file format, etc...) |

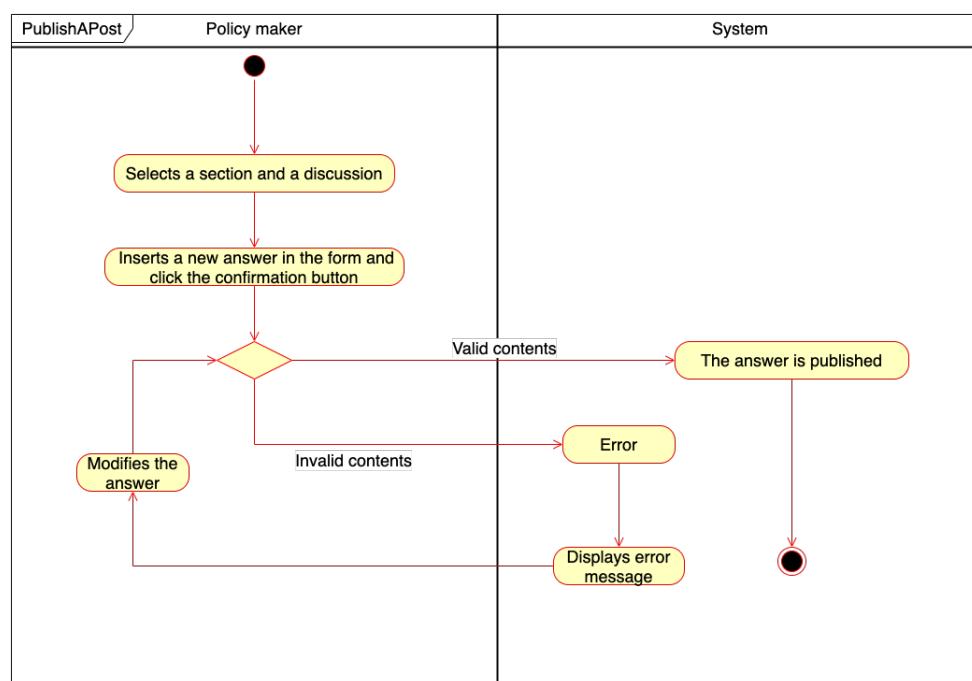


Figure 25: Publish a post by Policy maker

Table 13 Modify a post by Policy maker

| | |
|-----------------------------|--|
| ID | 12 |
| Name | Modify a post by Policy maker |
| Actor | Policy maker |
| Entry condition | <ul style="list-style-type: none"> • The Policy maker is already registered in the system • The Policy maker is already logged in the system • The Policy maker is in the discussion page where the post is present |
| Input | Post |
| Events flow | <ul style="list-style-type: none"> • The Policy maker press on the “Modify” button on the selected post • The System returns to the Policy maker the edit page • The Policy maker introduces the changes in the post • The Policy maker confirms the changes by clicking the confirmation button |
| Exit condition | The changes are introduced in the post |
| Exception | The Policy maker tries to insert invalid contents (invalid characters, invalid file format, etc...) |
| Special requirements | The Policy maker can modify every post |

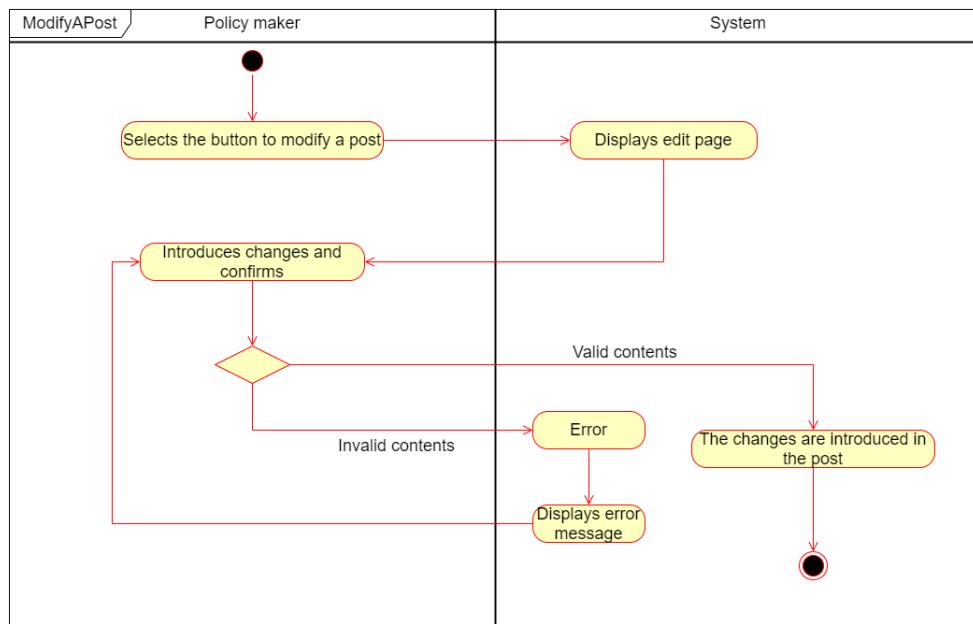


Figure 26: Modify a post by Policy maker

Table 14 Delete a post by Policy maker

| | |
|------------------------|---|
| ID | 13 |
| Name | Delete a post by Policy maker |
| Actor | Policy maker |
| Entry condition | <ul style="list-style-type: none"> • The Policy maker is already registered in the system • The Policy maker is already logged in the system • The post is already published • The Policy maker is in the discussion page where the post is present |
| Input | Post |

| | |
|-----------------------------|--|
| Events flow | <ul style="list-style-type: none"> The Policy maker press on the “Delete” button on the selected post The System returns to the Policy maker a confirmation message The Policy maker confirms the operation by clicking the confirmation button |
| Exit condition | The post is deleted from the forum |
| Exception | The Policy maker doesn't confirm the operation |
| Special requirements | The Policy maker can delete every post |

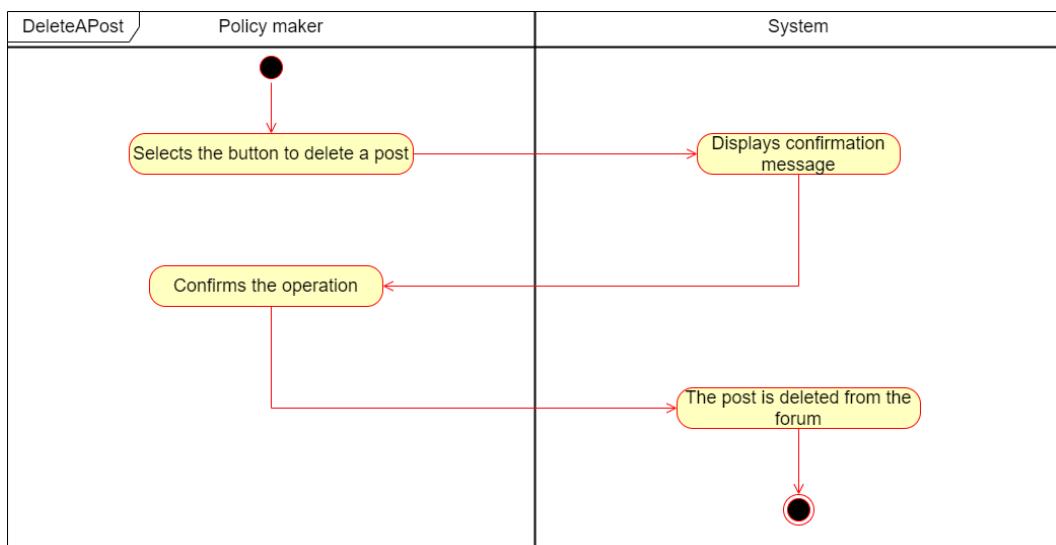


Figure 27: Delete a post by Policy maker

Table 15 Create a new discussion

| | |
|------------------------|--|
| ID | 14 |
| Name | Create a new discussion |
| Actor | Policy maker |
| Entry condition | <ul style="list-style-type: none">• The Policy maker is already registered in the system• The Policy maker is already logged in the system• The Policy maker is in the forum home page |
| Events flow | <ul style="list-style-type: none">• The Policy maker selects the topic of his interest• The Policy maker click on the “New discussion” button• The Policy maker writes/uploads the report• The Policy maker confirms the operation by clicking on the confirmation button |
| Exit condition | The report is saved and published in the system |
| Exception | The Policy maker tries to insert invalid contents |

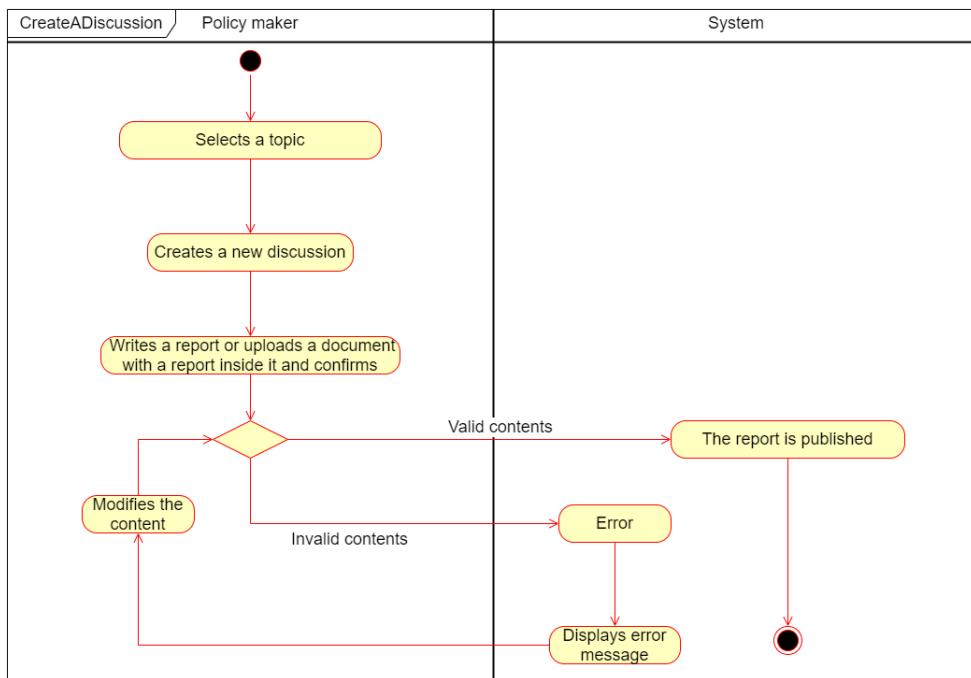


Figure 28: Create a new discussion

Table 16 Delete a discussion

| | |
|------------------------|--|
| ID | 15 |
| Name | Delete a discussion |
| Actor | Policy maker |
| Entry condition | <ul style="list-style-type: none"> The Policy maker is already registered in the system The Policy maker is already logged in the system The discussion is already existing The Policy maker is in the topic where the discussion is present |
| Input | Discussion |

| | |
|-----------------------------|--|
| Events flow | <ul style="list-style-type: none"> The Policy maker press on the “Delete” button on the selected discussion The System returns to the Policy maker a confirmation message The Policy maker confirms the operation by clicking the confirmation button |
| Exit condition | The discussion and all the posts inside of it are deleted from the forum |
| Exception | The Policy maker doesn't confirm the operation |
| Special requirements | The Policy maker can delete every discussion |

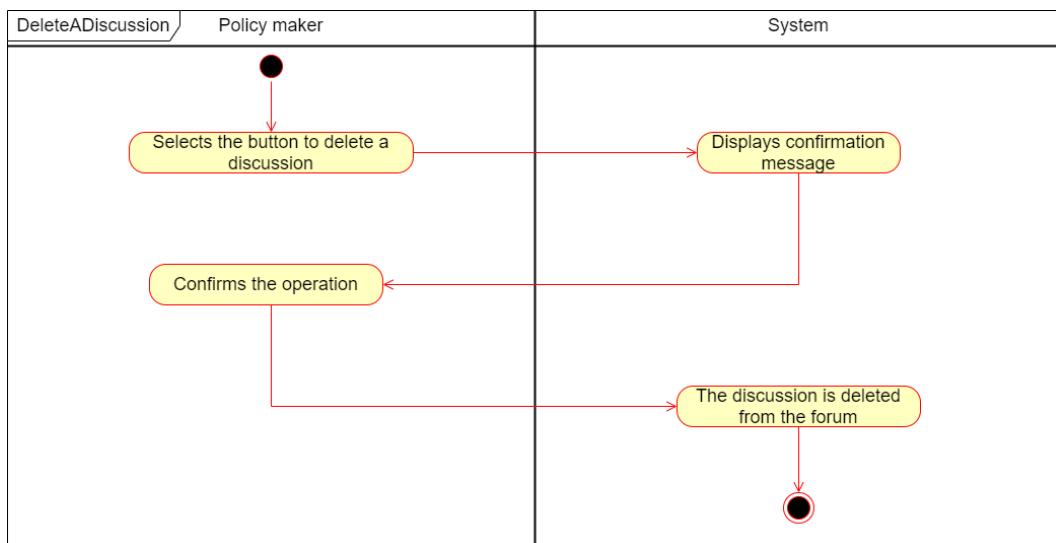


Figure 29: Delete a discussion

Table 17 Confirm pending post

| | |
|-----------------------------|---|
| ID | 16 |
| Name | Confirm pending post |
| Actor | Policy maker |
| Entry condition | <ul style="list-style-type: none"> • The Policy maker is already registered in the system • The Policy maker is already logged in the system • The Policy maker is in the forum home page |
| Events flow | <ul style="list-style-type: none"> • The Policy maker selects the “Moderator Area” button • The system displays the Moderator Area • The Policy maker selects the “pending list” option • The system displays the pending post list • The Policy maker selects a pending post request • The system displays the selected post • The Policy maker reviews the post • The Policy maker approves the post by clicking on the confirmation button |
| Exit condition | <ul style="list-style-type: none"> • The post is removed from the pending list and published in the forum • A notification is sent to the Author of the post |
| Special Requirements | The pending post must be accepted no later than 30 days after it is sent by User |

Table 18 Decline pending post

| | |
|--------------|----------------------|
| ID | 17 |
| Name | Decline pending post |
| Actor | Policy maker |

| | |
|------------------------|---|
| Entry condition | <ul style="list-style-type: none"> • The Policy maker is already registered in the system • The Policy maker is already logged in the system • The Policy maker is in the forum home page |
| Events flow | <ul style="list-style-type: none"> • The Policy maker selects the “Moderator Area” button • The system displays the Moderator Area • The Policy maker selects the “pending list” option • The system displays the pending post list • The Policy maker selects a pending post request • The Policy maker reviews the post • The Policy maker declines the approval of the post by clicking on the decline button |
| Exit condition | <ul style="list-style-type: none"> • The pending post request is deleted and removed from the pending post list • A notification is sent to the Author of the post |

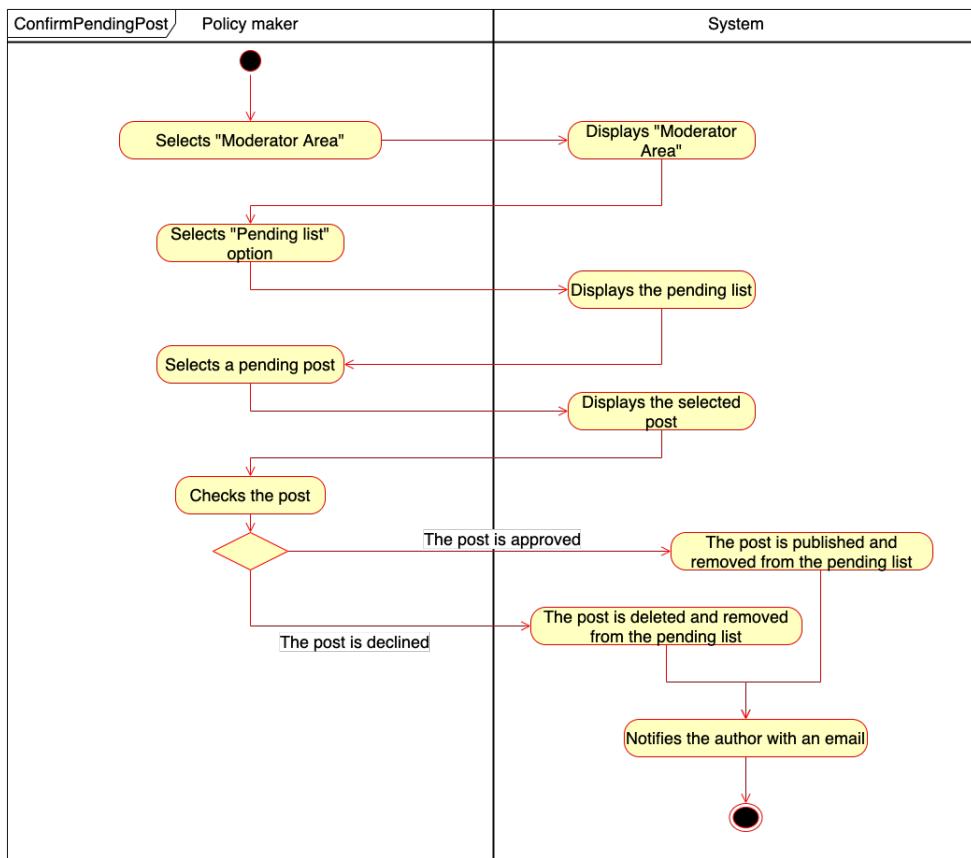


Figure 30: Confirm or decline a pending post

Table 19 Recalculate new deviance

| | |
|------------------------|--|
| ID | 18 |
| Name | Recalculate new Deviance |
| Actor | Policy maker |
| Entry condition | <ul style="list-style-type: none"> • The Policy maker is already registered in the system • The Policy maker is already logged in the system • The Policy maker is in the home page |
| Input | Parameters and data |

| | |
|-----------------------|---|
| Events flow | <ul style="list-style-type: none"> • The Policy maker selects the “Reserved Area” button • The System displays the Reserved Area • The Policy maker selects the “Deviance” option • The System displays the actual Deviance, calculated by default by the system, and a list of parameters that he can select in order to recalculate a custom Deviance • The Policy maker selects different parameters • The Policy maker clicks on Recalculate • The System recalculate the new Deviance • The System displays the new ranking list |
| Exit condition | The Deviance is recalculated and it's displayed the new ranking list |
| Exception | The Policy maker didn't selects any parameter |

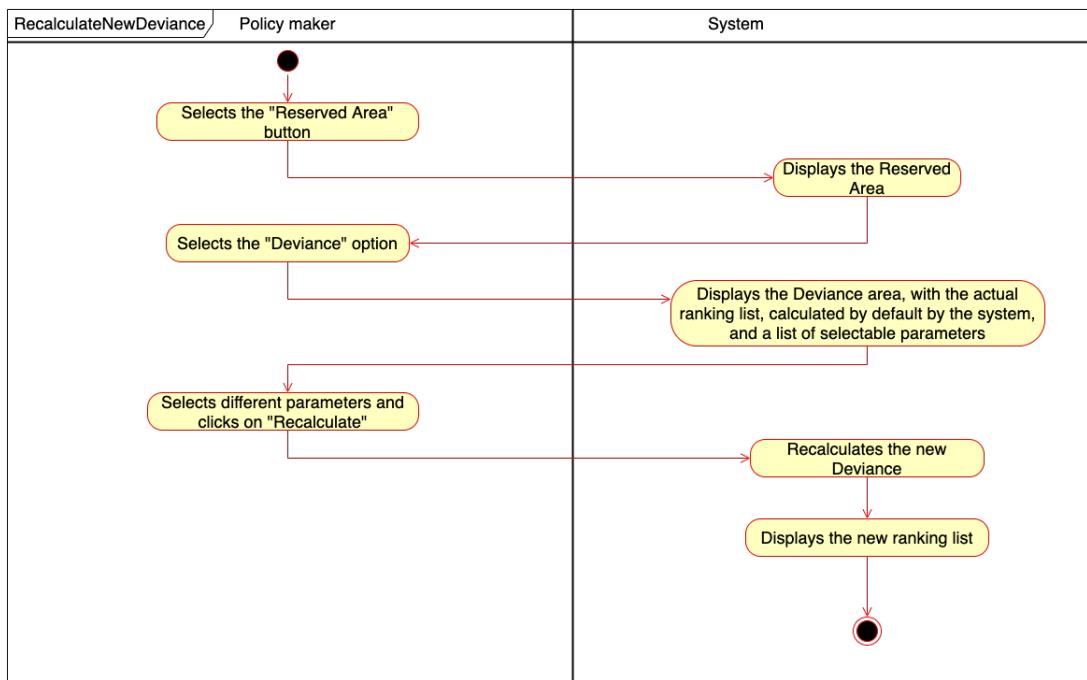


Figure 31: Recalculate new deviance

3.2.4 Administrator scenarios

Scenario 10

Alessandro is an administrator of the Dream system and wants to add a new data source. He connects to the administration site and login using his administrator credentials. On the administration screen he selects the “Manage data sources” section. He clicks on the button to add a new data source and inserts the source link from where the data will be recovered and clicks on the import button.

Use case diagram

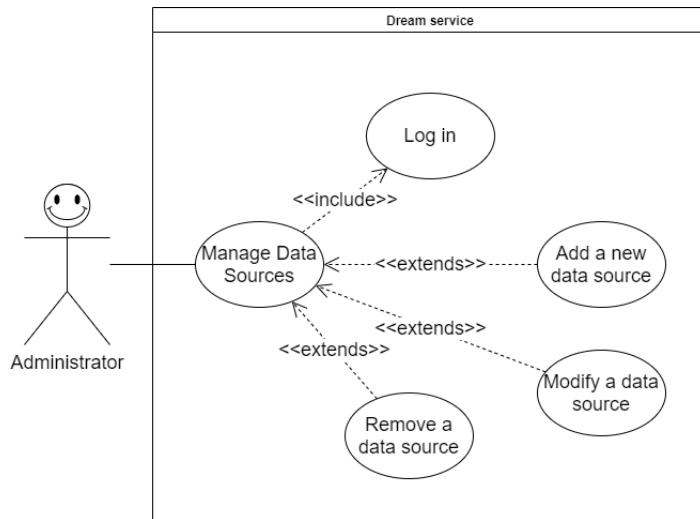


Figure 32: Use case diagram

Use case tables

Table 20 Login Administrator

| | |
|------------------------|---|
| ID | 19 |
| Name | Login Administrator |
| Actor | Administrator |
| Entry condition | <ul style="list-style-type: none"> Administrator has reached the administration site |
| Input | Administrator's email and password |
| Events flow | <ul style="list-style-type: none"> The system displays the login page Administrator fills the username (email) and password fields using the credential System checks the validity of the credentials inserted The system displays the precedent page or, if unavailable, the home page of the administration dashboard |

| | |
|-----------------------|--|
| Exit condition | Administrator is logged in |
| Exceptions | Administrator inserts wrong credentials and clicks on “login” button. The system shows an error message inviting the Administrator to check the credentials before trying again to login |

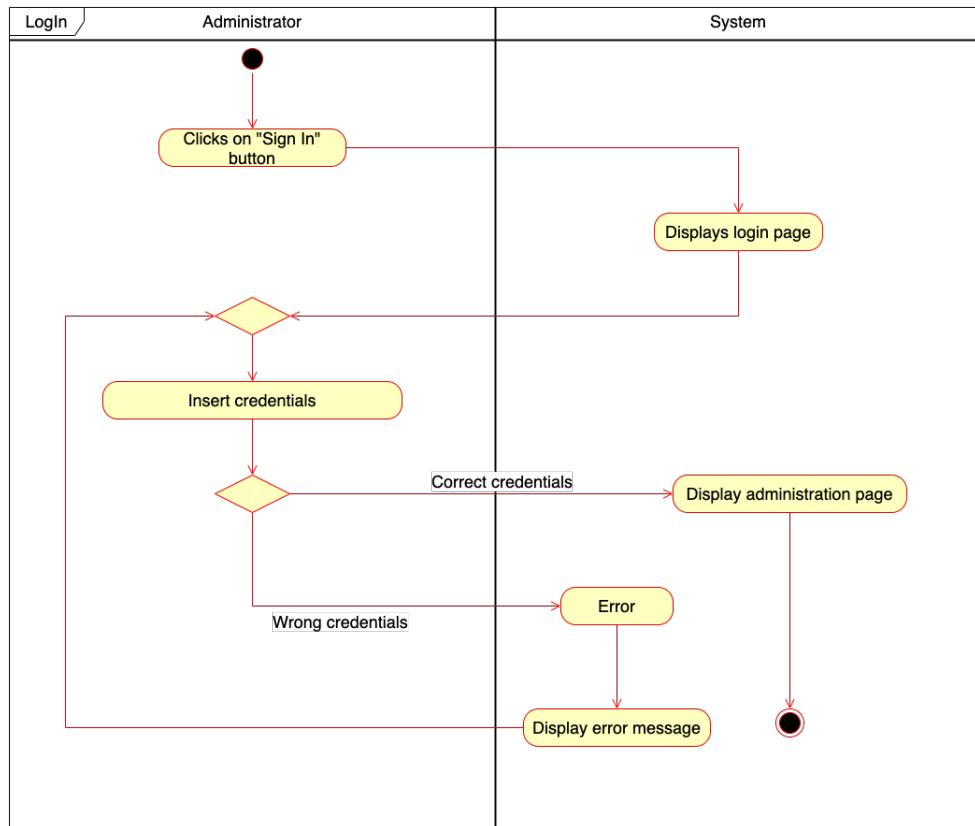


Figure 33: Login Administrator

Table 21 Add new data source

| | |
|-----------------------------|---|
| ID | 20 |
| Name | Add new data source |
| Actor | Administrator |
| Input | Data source to add |
| Entry condition | <ul style="list-style-type: none">• The Administrator is already registered in the system• The Administrator is already logged in the system• The Administrator is in the admin home page |
| Events flow | <ul style="list-style-type: none">• The Administrator selects the “Data sources” section• The System opens the “Data sources” page• The Administrator clicks on the button to add a new source• The System render the form to add a new source• The Administrator compile the form with the new data source’s origin• The Administrator confirms the operation |
| Exit condition | The new data source is available on the site |
| Exceptions | <ul style="list-style-type: none">• The Administrator tries to insert an unavailable source• The Administrator tries to insert a data source already present in the system |
| Special requirements | The Administrator needs to have the copyright permission to use the data source |

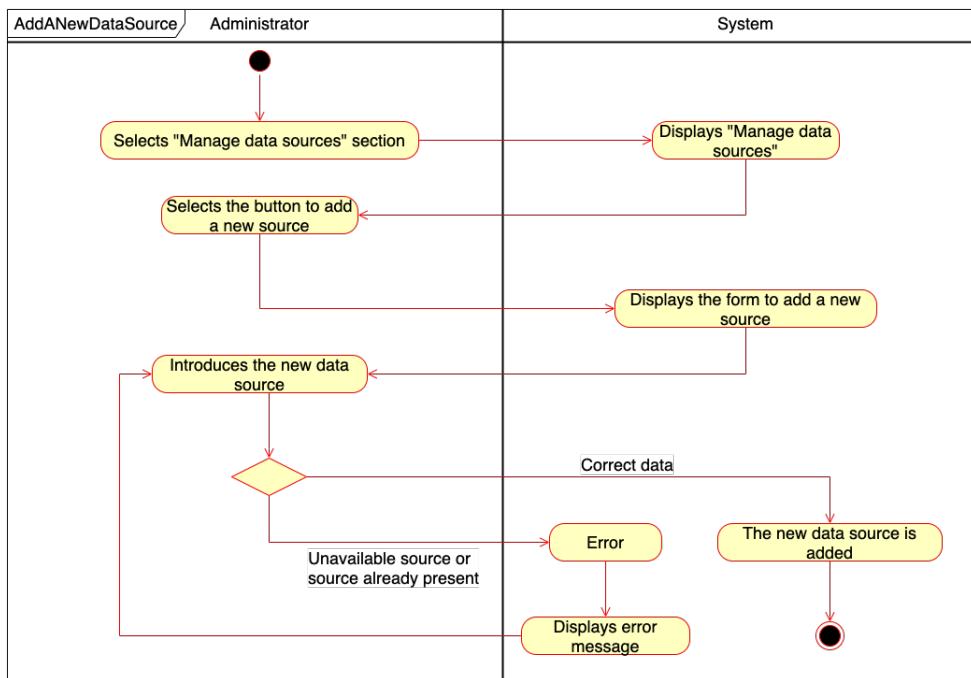


Figure 34: Add new data source

Table 22 Modify a data source

| | |
|------------------------|---|
| ID | 21 |
| Name | Modify a data source |
| Actor | Administrator |
| Input | Changes to a data source |
| Entry condition | <ul style="list-style-type: none"> • The Administrator is already registered in the system • The Administrator is already logged in the system • The Administrator is in the admin home page |

| | |
|-----------------------------|---|
| Events flow | <ul style="list-style-type: none"> • The Administrator selects the “Data sources” section • The System opens the “Data sources” page • The Administrator clicks on the “Modify” button associated to the data source he wants to modify • The System render a form, from which the administrator could modify some data • The Administrator modify some parameters of the selected data source • The Administrator confirms the operation |
| Exit condition | The data source is modified and is available on the site |
| Exceptions | <ul style="list-style-type: none"> • The Administrator doesn't confirm the operation • The changes results in equivalence with another existing data source |
| Special requirements | The Administrator needs to have the copyright permission to use the data source |

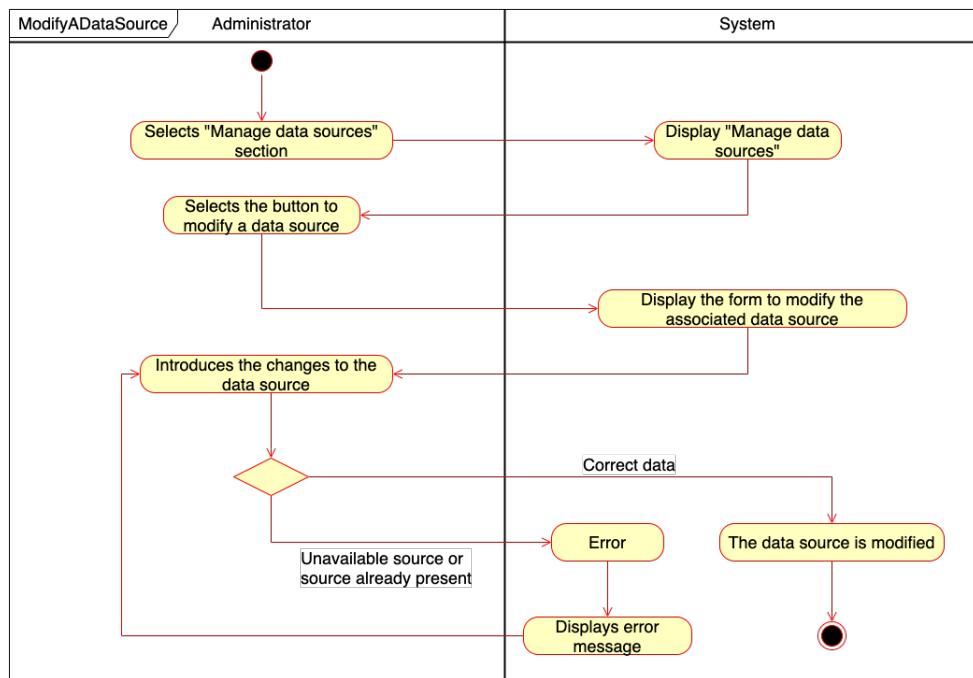


Figure 35: Modify a data source

Table 23 Remove a data source

| | |
|------------------------|---|
| ID | 22 |
| Name | Remove a data source |
| Actor | Administrator |
| Input | Data source to remove |
| Entry condition | <ul style="list-style-type: none"> • The Administrator is already registered in the system • The Administrator is already logged in the system • The Administrator is in the admin home page |

| | |
|-----------------------|--|
| Events flow | <ul style="list-style-type: none"> The Administrator selects the “Data sources” section The System opens the “Data sources” page The Administrator clicks on the button to remove a data source, associated to the data source he wants to remove The System render an alert asking for confirmation to remove the data source The Administrator confirms the operation |
| Exit condition | The data source is no more available in the system |
| Exceptions | The Administrator doesn't confirm the operation |

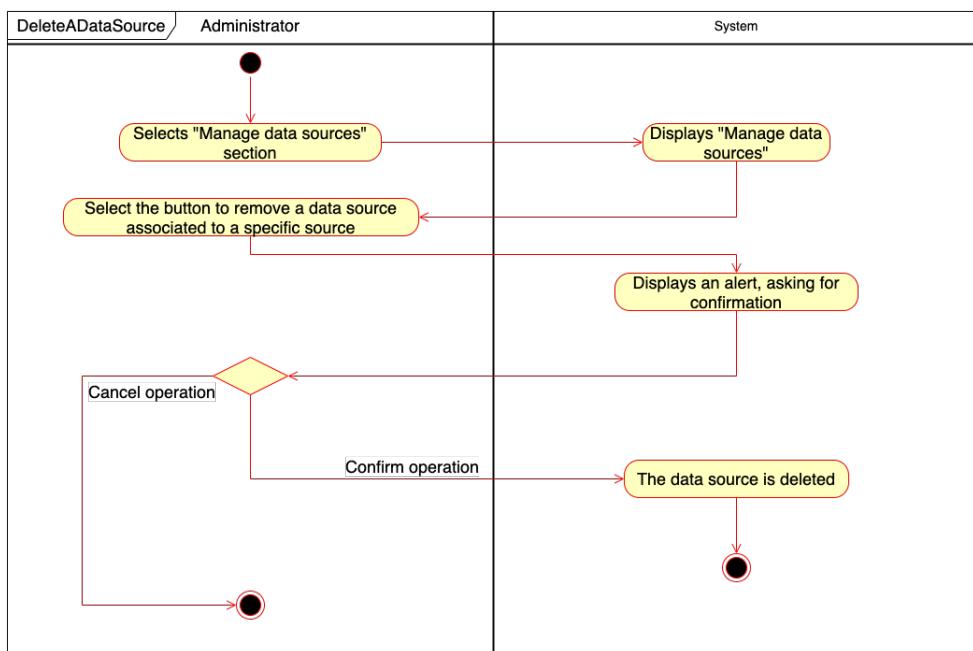


Figure 36: Remove a data source

3.2.5 Requirements

- **R1:** The system let a Visitor to register to the identity provider.
- **R2:** The system let identity provider's user to login in the correctly mapped role to associated to its account.
- **R3:** The system lets a registered user to add a reply of a discussion.
- **R4:** The system lets the Policy maker to modify a reply of a discussion.
- **R5:** The system lets the Policy maker to delete a reply of a discussion.
- **R6:** The system lets the Policy maker to see all the pending reply publication.
- **R7:** The system lets the Policy maker to accept a reply publication.
- **R8:** The system lets the Policy maker to decline a reply publication.
- **R9:** The system lets the Policy maker to create a new discussion.
- **R10:** The system lets the Policy maker to delete a discussion.
- **R11:** The system allows an User to modify the post written by him.
- **R12:** The system allows an User to delete the post written by him.
- **R13:** The system should sends a notification at all the participant to a discussion when a changes is made.
- **R14:** The system should sends a notification to the User when his reply is approved.
- **R15:** The system should sends a notification to the User when his reply is denied.
- **R16:** The system lets the Administrator to add a new data source.
- **R17:** The system lets the Administrator to remove a data source.
- **R18:** The system lets the Administrator to modify a data source.
- **R19:** The system lets the Administrator to see the active data sources.
- **R20:** The system lets the Policy maker to select a specific parameter when trying to recalculate the Deviance.
- **R21:** The system lets the Policy maker to recalculate the Deviance.
- **R22:** The system lets the Visitor to search for a subset of the data.
- **R23:** The system lets the Visitor to download a subset of the data.
- **R24:** The system lets the Visitor to navigate the forum.

3.2.6 Traceability Matrix

Table 24 Traceability Matrix

| Requirement | Use cases |
|-------------|---|
| R1 | Sign up with email |
| R2 | Login User Login Policy maker Login Administrator |
| R3 | Publish a post by User Publish a post by Policy maker |
| R4 | Modify post by Policy maker |
| R5 | Delete a post by Policy maker |
| R6 | Confirm pending post Decline pending post |
| R7 | Confirm pending post |
| R8 | Decline pending post |
| R9 | Create a new discussion with some content |
| R10 | Delete a discussion by Policy maker |
| R11 | Modify a post by User |
| R12 | Delete a post by User |
| R13 | Modify post by Policy maker Publish a post by Policy maker Confirm pending post |
| R14 | Confirm pending post |
| R15 | Decline pending post |
| R16 | Add new data source |
| R17 | Remove a data source |
| R18 | Modify a data source |
| R19 | Add new data source Remove a data source Modify a data source |
| R20 | Recalculate new Deviance |
| R21 | Recalculate new Deviance |
| R22 | Filter data |
| R23 | Download data |
| R24 | Visitor navigate the forum |

3.3 Performance Requirements

The system needs to have a response time less than a second, for chunk of one thousand record, in order to be able to provide data when needed, but also to be performing enough to load and analyze incoming data like a real-time system would do. The average workload of the system is expected to be very high, because it needs to manage a huge number of data. This last goal could be achieved by using a distributed system.

3.4 Design constraints

3.4.1 Standards compliance

The system adopts standardized and precise units of measure to quantify the different data obtained through the APIs:

- Temperature: Celsius degree ($^{\circ}C$)
- Humidity: $\frac{Kg}{m^3}$
- Rainfall: Millimeter (mm)
- Population density: $\frac{Inhabitants}{m^2}$

3.4.2 Hardware limitations

As specified in the “Hardware constraints” section (2.1.3), specific hardware requirements are not asked, except for the internet connection and the ability to run a browser.

3.4.3 Privacy policies

The system uses only data retrieved from public domain. Data from non public domain data sets are not collected by choice. There aren't sensible data that could be provided to third part applications, so User doesn't need to authorize the use of any data. Personal User's data are stored and encrypted by IdP and only the IdP's administrator have access to the database.

3.5 Software system attributes

3.5.1 Easy usability

The system should be easy to use in general but a particular attention must be taken in the following components:

- Forum: it should be designed to be minimal with easy to access functionality considering that it can be used by non-tech people.

- Public data access: APIs and their parameters should be correctly named to make their scope clearly understandable without the necessity of specific documentation (e.g. an endpoint to get weather data should be named 'get_weather?start_date&end_date').

3.5.2 Reliability

The system must prevent data loss both in the internal data set and in the incoming data.

The system must also be designed to be capable of supporting a huge number of access through the API's service and be able to manage and store a high number of Users.

3.5.3 Availability

The system must be available as much as possible with a minimum value of 99% with regards to the data manager component while lower values could be accepted for the forum (but possibly higher than 90%).

3.5.4 Security

Communication between parties are encrypted and goes on a secure channel using SSL protocol and authentication and authorization are performed through SAML2.0 protocol in order to provide a system that can easily integrate with institutional identity providers. Database operation are always authorized (for instance a Policy maker cannot modify Data sources).

3.5.5 Modularity

The system is designed in function of an high re-usability: a modular design permits to add functions in future with less effort.

3.5.6 Maintainability

The usage of a modular approach should facilitate further maintenance.

4 Formal Analysis with Alloy

4.1 Alloy code

```
1 //Signatures
2 sig Name {}
3
4 sig Surname {}
5
6 sig Email {}
7
8 sig Date {}
9
10 sig AreaOfResidence {}
11
12 sig PolicyMakerId {}
13
14 sig User {
15     name: one Name,
16     surname: one Surname,
17     email: one Email,
18     dateOfBirth: one Date,
19     areaOfResidence: one AreaOfResidence,
20 }
21
22
23 sig PolicyMaker extends User {
24     policyMakerId: one PolicyMakerId,
25 }
26
27 sig Administrator {
28     email: one Email,
29     password: one Password,
30 }
31
32 sig Password {}
33 { //Each password is associated to an Administrator
34     all p : Password | (some a: Administrator | a.
35         → password = p)
36 }
37 sig Text {}
38
39 sig DateTime in Int {}
```

```

40
41 abstract sig Status {}
42
43 sig PENDING extends Status{}
44
45 sig ACCEPTED extends Status {}
46
47 sig REJECTED extends Status {}
48
49 sig Attachment {}
50
51 sig DiscussionTitle {}
52
53 sig SubTitle {}
54
55 sig TopicTitle {}
56
57 abstract sig Visibility {}
58
59 sig Visible extends Visibility {}
60
61 sig Invisible extends Visibility {}
62
63 sig Topic {
64     title: one TopicTitle,
65     timestamp: one DateTime,
66     discussions: some Discussion,
67 }
68
69 sig Discussion {
70     title: one DiscussionTitle,
71     subTitle: one SubTitle,
72     timestamp: one DateTime,
73     creator: one PolicyMaker,
74     posts: some Post,
75 }
76
77 sig Post {
78     text: one Text,
79     creator: one User,
80     timestamp: one DateTime,
81     attachment: lone Attachment,
82     status: one Status,
83     visibility: one Visibility,

```

```

84    }
85
86 sig DataName {}
87
88 sig Source {}
89
90 sig Description {}
91
92 sig DataType {
93     name: one DataName,
94 }
95
96 sig DataSource {
97     name: one DataName,
98     source: one Source,
99     description: lone Description,
100    dataType: one DataType,
101 }
102
103 -----
104 //Facts
105
106 fact { //A name exist only if it's present an User
107     all n: Name | one u: User | u.name = n
108 }
109
110 fact { //A surname exist only if it's present an User
111     all s: Surname | one u: User | u.surname = s
112 }
113
114 fact { //Each policyMakerId is unique
115     no disj p1, p2: PolicyMaker | p1.policyMakerId =
116         ↪ p2.policyMakerId
117 }
118
119 fact { //Each User has an unique email
120     no disj u1, u2 : User | u1.email = u2.email
121 }
122
123 fact { //Each Administrator has an unique email
124     no disj a1, a2 : Administrator | a1.email = a2.email
125 }
126 fact { //Administrators and Users have different emails

```

```

127         no disj a: Administrator, u: User | a.email = u.email
128     }
129
130 fact { //If a post exist, it must be PENDING, ACCEPTED or
131   ↳ REJECTED
132     all p: Post | p.status = PENDING or p.status =
133       ↳ ACCEPTED or p.status = REJECTED
134   }
135
136 fact { //Two Users can not be creators of the same Post
137   all p: Post | (no disj u1, u2: User | (u1 = p.creator
138     ↳ and u2 = p.creator and u1 != u2))
139   }
140
141
142 fact { //Two Policy makers can not be creators of the same
143   ↳ Discussion
144     all d: Discussion | (no disj p1, p2: PolicyMaker | (p1
145       ↳ = d.creator and p2 = d.creator and p1 != p2))
146   }
147
148
149 fact { //A Post always belong to one Discussion
150   all p: Post | one d: Discussion | p in d.posts
151   }
152
153
154 fact { //A Discussion always belong to one Topic
155   all d: Discussion | one t: Topic | d in t.discussions
156   }
157
158 fact { //A Discussion has always a title
159   all d: Discussion | one t: DiscussionTitle | d.title =
160     ↳ t
161   }
162
163 fact { //A Topic has always a title
164   all to: Topic | one ti: TopicTitle | to.title = ti
165   }
166
167
168 fact { //A Post can not exists without a creator
169   all p: Post | one u: User | u = p.creator
170   }
171
172
173 fact { //A Discussion can be created only by a Policy maker
174   all d: Discussion | one p: PolicyMaker | p = d.creator
175     ↳ and p.policyMakerId != none

```

```

164 }
165
166 fact { //There can not exist a Post without Text and without
167   → Attachment
168     no disj p: Post | p.text = none and p.attachment =
169       → none
170 }
171
172 fact { //A Post is not visible if it has been rejected or is
173   → still in the pending list
174     all p: Post | (p.status = PENDING or p.status =
175       → REJECTED) implies p.visibility = Invisible
176 }
177
178 fact { //A Post is visible if it has been accepted
179   all p: Post | p.status = ACCEPTED implies p.visibility =
180     → = Visible
181 }
182
183 fact { //An attachment exist only if it's present a Post
184   all a: Attachment | one p: Post | p.attachment = a
185 }
186
187 fact { //A text exist only if it's present a Post
188   all t: Text | one p: Post | p.text = t
189 }
190
191 fact { //A subTitle exist only if it's present a Discussion
192   all s: SubTitle | one d: Discussion | d.subTitle = s
193 }
194
195 fact { //A discussion has at least one ACCEPTED post
196   all d: Discussion | some p: Post | p.status = ACCEPTED
197     → and p in d.posts
198 }
199
200 fact { //Each post is unique
201   no disj p1, p2: Post | p1 = p2
202 }
203
204 fact { //Each discussion is unique
205   no disj d1, d2: Discussion | d1 = d2
206 }

```

```

202 fact { //Each topic is unique
203     no disj t1, t2: Topic | t1 = t2
204 }
205
206 fact { //A post of a Policy maker is always accepted
207     no p: Post | one pm: PolicyMaker | pm = p.creator and
208     ↳ p.status != ACCEPTED
209 }
210
211 fact { //Each topic has an unique title
212     no disj t1, t2 : Topic | t1.title = t2.title
213 }
214
215 fact { //Each discussion has an unique title
216     no disj d1, d2 : Discussion | d1.title = d2.title
217 }
218
219 fact { //A description exist only if it's present a DataSource
220     all d: Description | one ds: DataSource |
221     ↳ ds.description = d
222 }
223
224 fact { //A source exist only if it's present a DataSource
225     all s: Source | one ds: DataSource | ds.source = s
226 }
227
228 fact { //A Data source can not exist without a source
229     all d: DataSource | one s: Source | d.source = s
230 }
231
232 fact { //DataSource and DataType have different names
233     no disj ds: DataSource, dt: DataType | ds.name =
234     ↳ dt.name
235 }
236
237
238
239
240 -----
241 //Assertions
242

```

```

243 //G2: Allow the platform Administrator to decide which public
244   ↵ data should be used in the Deviance computation
245 assert dataSourceIsInserted {
246   all ds: DataSource | one dt: DataType, s: Source |
247     ↵ ds.source = s and ds.dataType = dt
248 }
249 check dataSourceIsInserted for 40
250
251 // G3: Allow people to interact and build a knowledge network
252 assert publishAPost {
253   no p: Post | all d: Discussion | some u: User | u =
254     ↵ p.creator and p not in d.posts
255 }
256 check publishAPost for 40
257
258 // G3: Allow people to interact and build a knowledge network
259 assert confirmAPost {
260   no p: Post | one d: Discussion | p.status = ACCEPTED
261     ↵ and p.visibility = Invisible and p in d.posts
262 }
263 check confirmAPost for 40
264
265 // G5: Allow Policy Makers to release publicly their reports
266   ↵ based on the Deviance result
267 assert createADiscussion {
268   all d: Discussion | some p: Post | one pm:
269     ↵ PolicyMaker, t: Topic | p in d.posts and d.creator
270     ↵ = pm and d in t.discussions and p.status =
271     ↵ ACCEPTED
272 }
273 check createADiscussion for 20
274
275 -----
276 //Predicates
277
278 pred dataAdministration {
279   # Administrator > 0
280   # DataSource > 3
281   # Topic = 0
282   # Discussion = 0
283   # Post = 0
284   # User = 0
285 }
```

```
279 run dataAdministration for 10
280
281
282 pred forum {
283     # Topic > 2
284     # Discussion > 3
285     # Post > 3
286     # DataSource = 0
287     # DataType = 0
288 }
289 run forum for 10
290
291 pred accounts {
292     # Administrator > 2
293     # PolicyMaker > 2
294     # User > 4
295     # DataSource = 0
296     # DataType = 0
297     # Topic = 0
298 }
299 run accounts for 10
```

4.2 Worlds

73

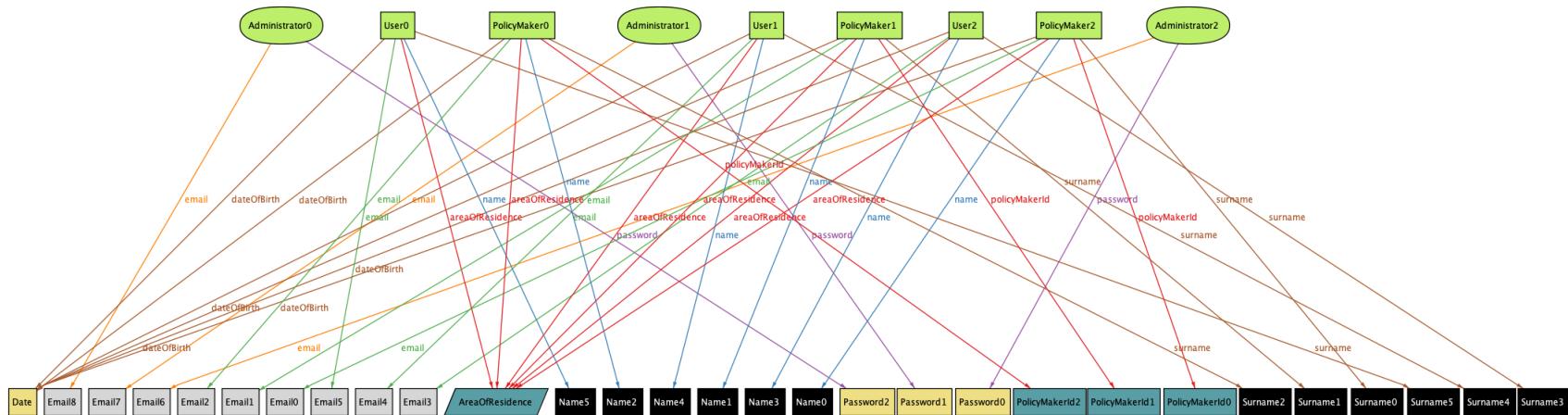


Figure 37: Output of "accounts" world

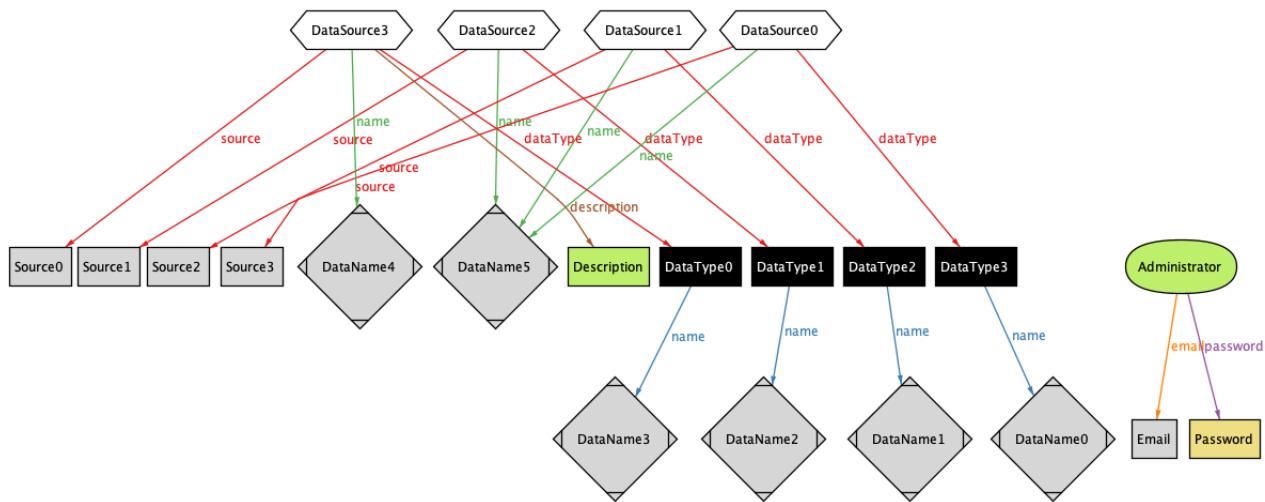


Figure 38: Output of "dataAdministrator" world

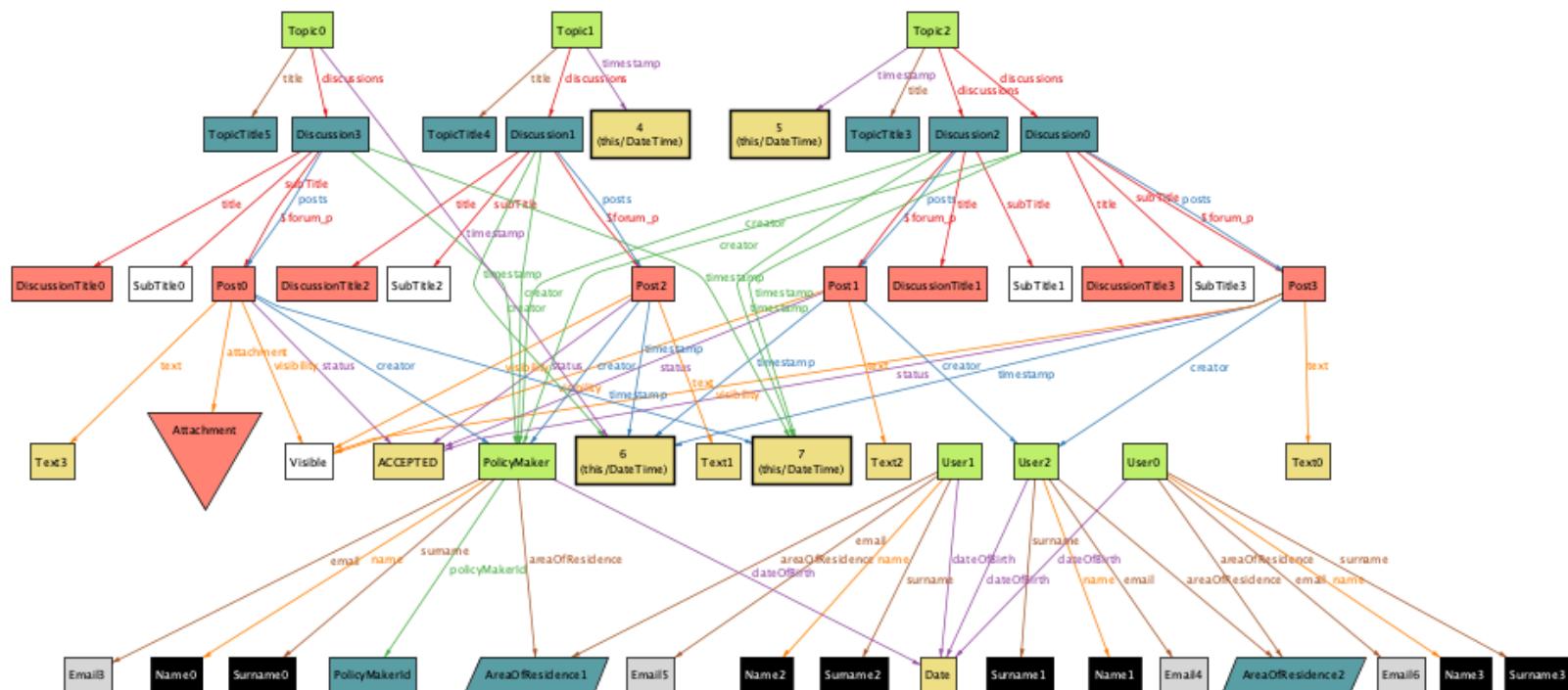


Figure 39: Output of "forum" world

5 Effort Spent

| | Alessandro Cecchetto | Mattia Siriani | Matteo Visotto |
|--------------|----------------------|----------------|----------------|
| Time for S.1 | 6h | 4h | 4h |
| Time for S.2 | 7h | 7h | 7h |
| Time for S.3 | 13h | 16h | 15h |
| Time for S.4 | 4h | 4h | 5h |
| Total | 30h | 31h | 31h |