Introduction

🏆 A1.1 Learning Activity

Requirements Modeling through UML

📘 Instructions

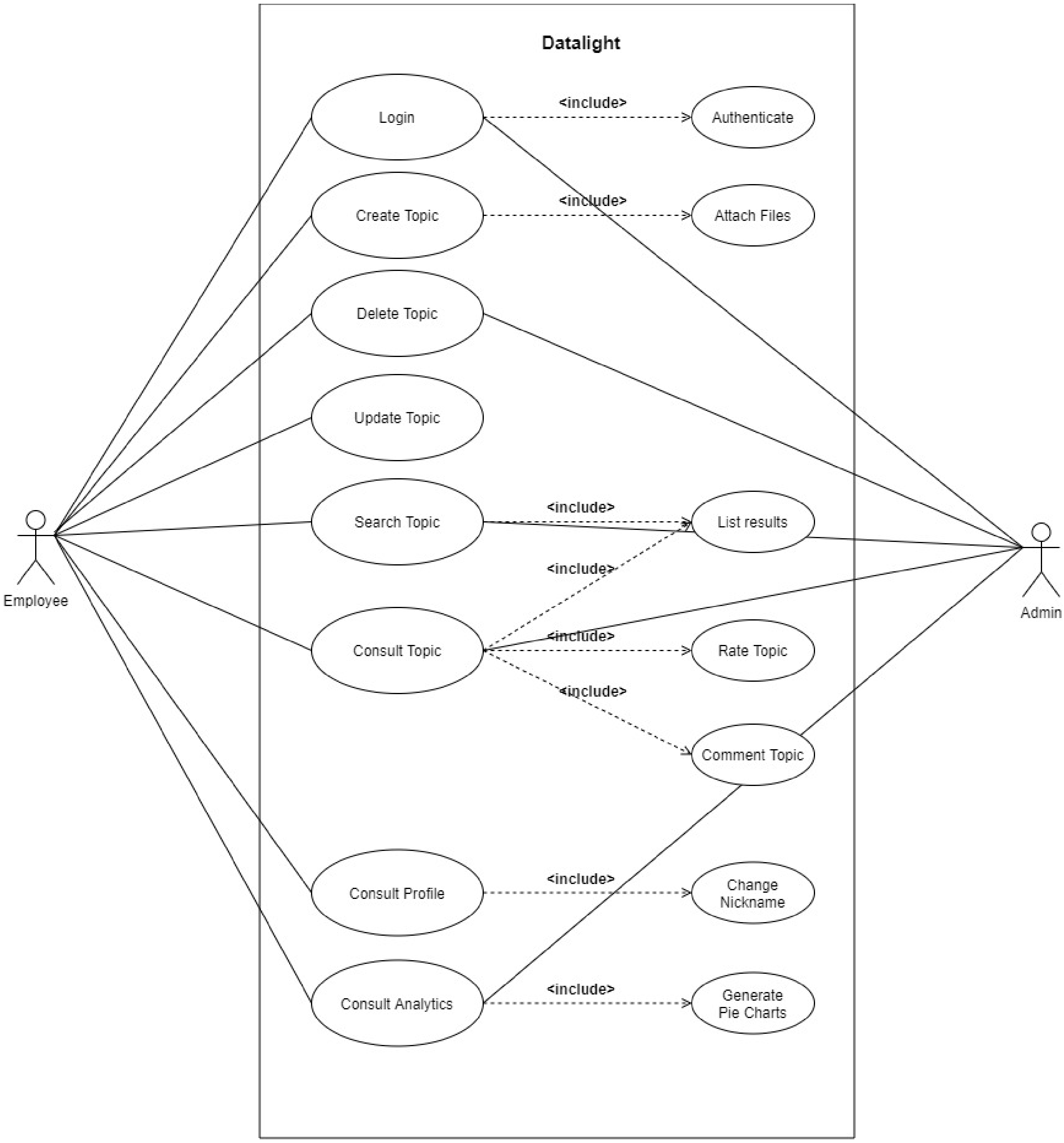
- Based on the provided document by the teacher and other sources, develop the indicated diagrams in the development for the case study that has been reviewed during the subject.
- Use the same tools from past challenges, such as drawio, or others that allow you to create the diagrams or images that can illustrate the concept.
- The team will explain what each type of diagram represents, what elements are part of it, as well as its interpretation within the case study.
- Every activity or challenge must be realized using **Markdown style with .md extension** and VSCode development environment, or another platform can be used, such as **Notion**, it has to be as a **single page** document, which means if the document has images, links to another external document it must be accessed from tags and links, and must be named using the nomenclature **A1.1_ActivityName_Student'sName.pdf.**
- As a requirement the .md file must contain a tag of the link to the repository of your Github document, for example a **Link to my GitHub** and after concluding the assignment it must be submitted to Github.
- From the **.md** file export a **.pdf** that must be uploaded to classroom in the corresponding section, serving as evidence of turned in, since it being the **official** platform, here you will receive the activity's result.
- Considering that the .pdf file was obtained from the .md file, both must be identical.
- Your repository, besides containing a **readme**.md inside the root directory containing student's information, team, subject, career, teacher's information, and also a logo or pictures, must contain a section of contents or index, which are **links to your md documents**, try not to use text to indicate internal or external links.
- It's suggested to use a structure similar to the one indicated below, however, you can use any other structure that can help you organize your repository.

```
| readme.md
| | blog
| | | Cx.1_ActivityName.md
| | | Ax.1_ActivityName.md
| | diagrams
| | docs
| | html
| | img
| | pdf
```

✏️ Development

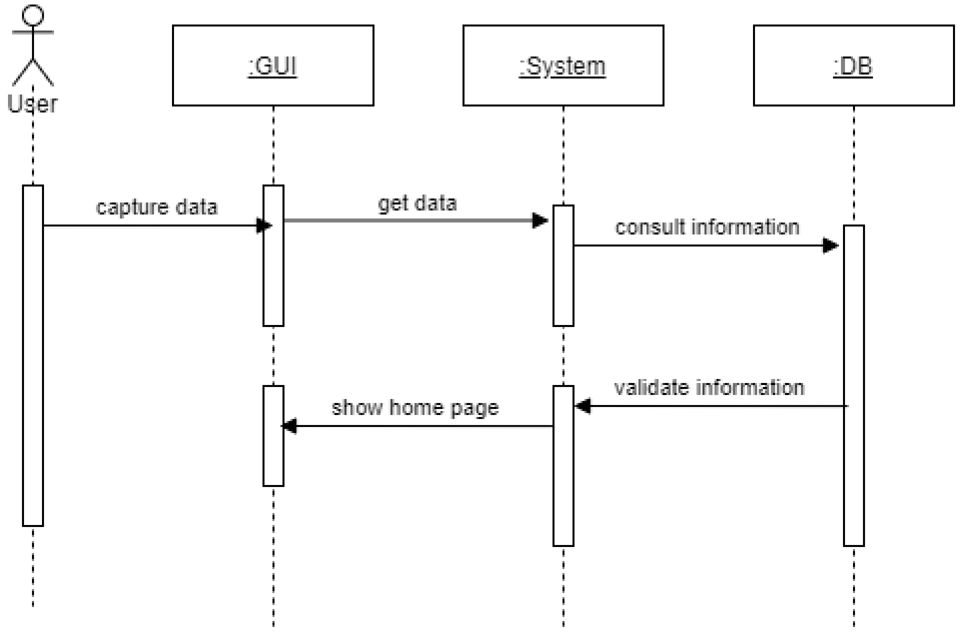
1. Based on each one of the previously established user and system requirements for the case study, draw up the requested modeling diagrams:

- A diagram containing the relationships between the different use cases, only for the scenario in which the user interacts with the system, using the main or central functions according to each case study. (Include at least 5 elements of the diagram)
  - The use case diagram uses actors, the class and the use cases related to the study case # 2 "knowledge conservation" in which the use cases corresponding to the primary activities of the system are placed and also the things that must to include

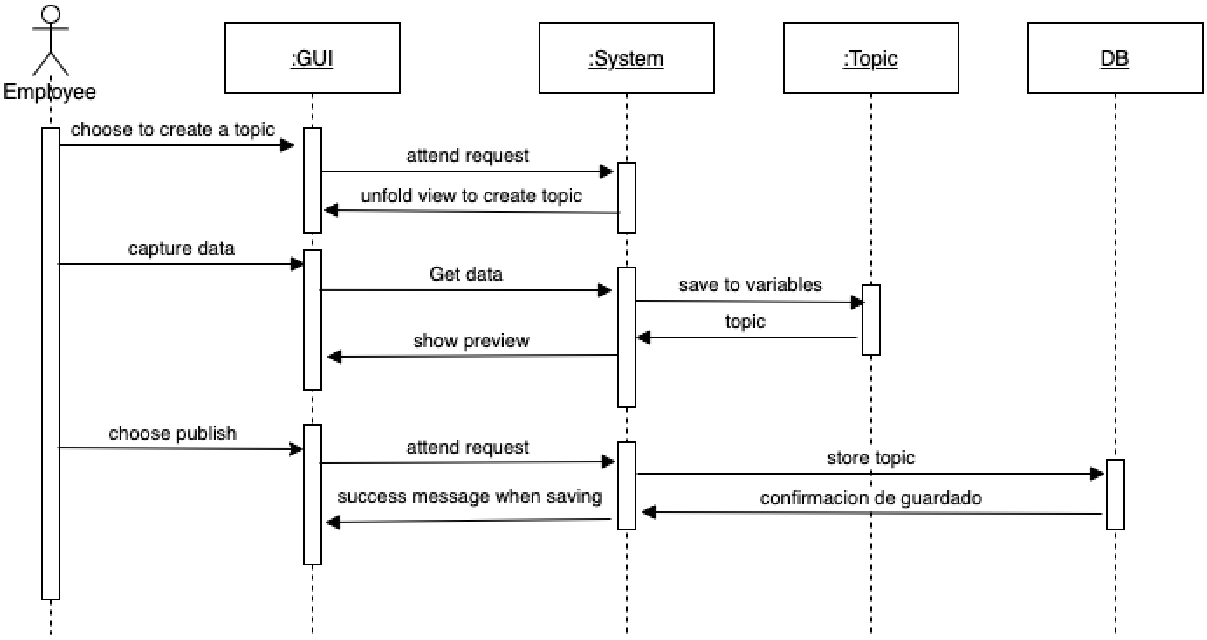


- The sequence diagram that explains the communication between the different GUIs for the use cases of the previous point, considering that the user has successfully accessed the system. (Include at least 5 elements from the diagram)
  - Sequence diagrams are used to describe the passage of activities through the system interfaces

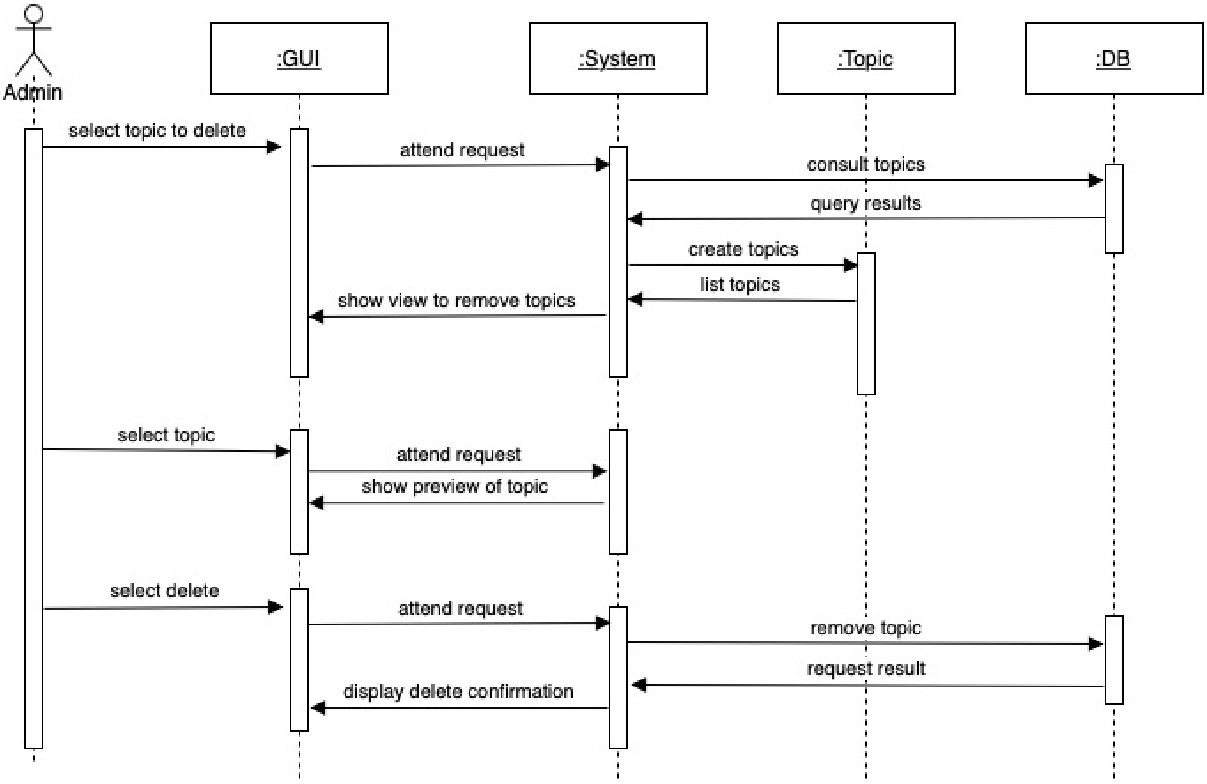
- Login: describes the employee's process for accessing the platform through the system, the interface and the DB



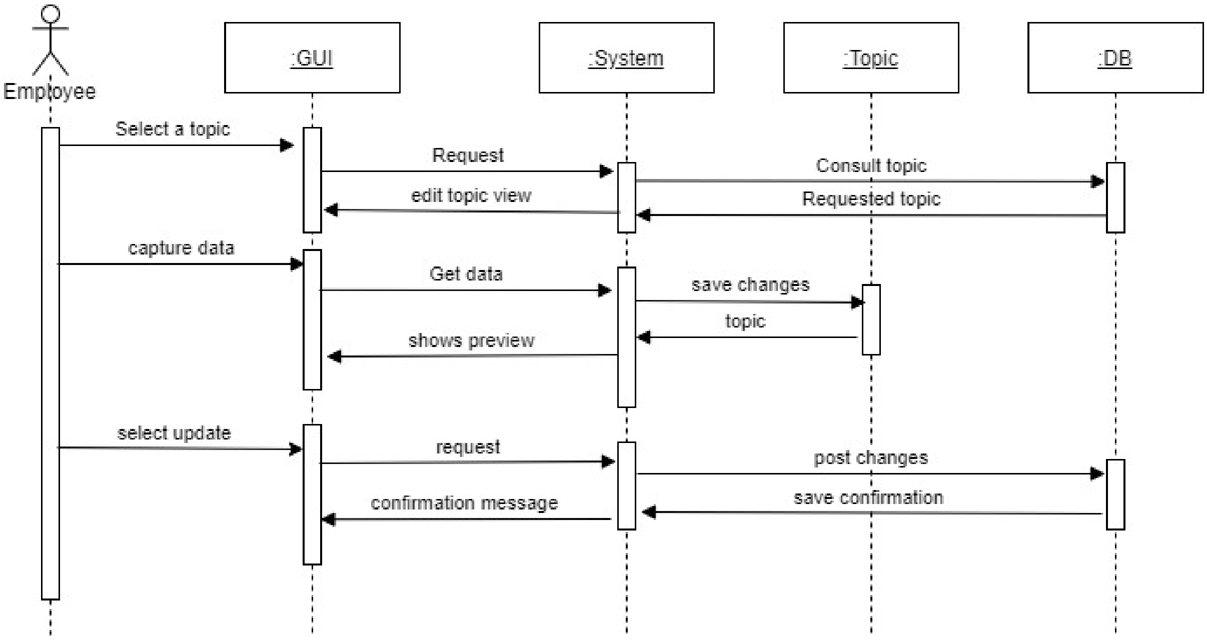
- Create Topic: Describes the employee's process for creating a topic and how he requests it from the system, creates a topic and saves it in the DB



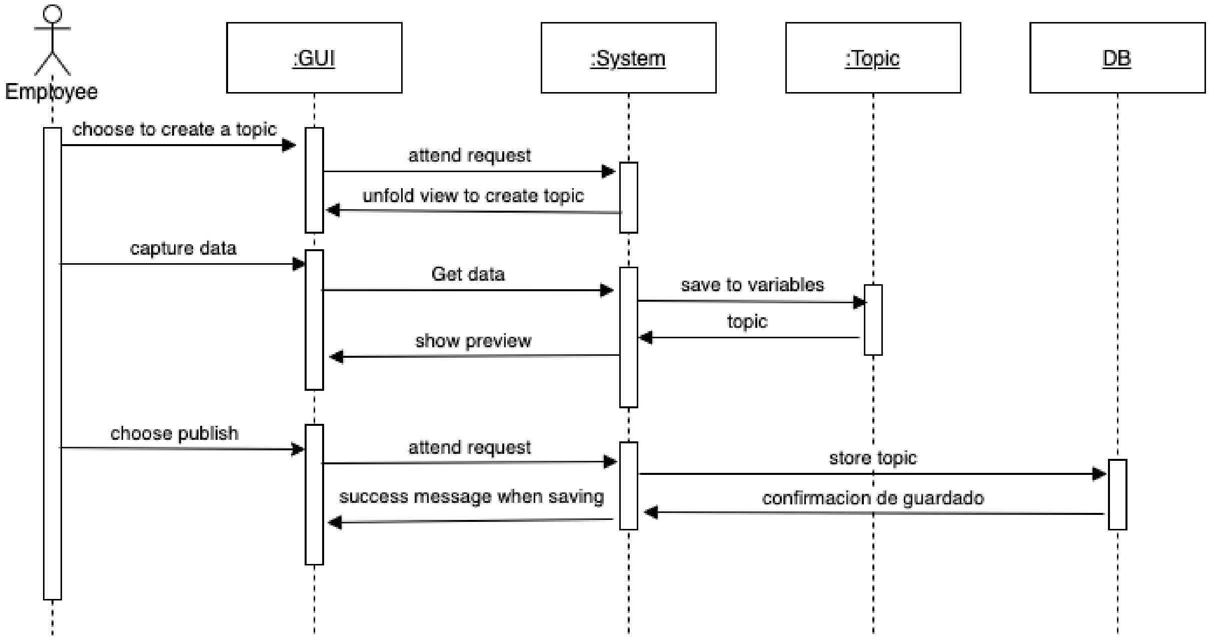
- Delete Topic: Describes the employee's process to delete a topic and how it asks the system for the stored topic, removes it and notifies the interface



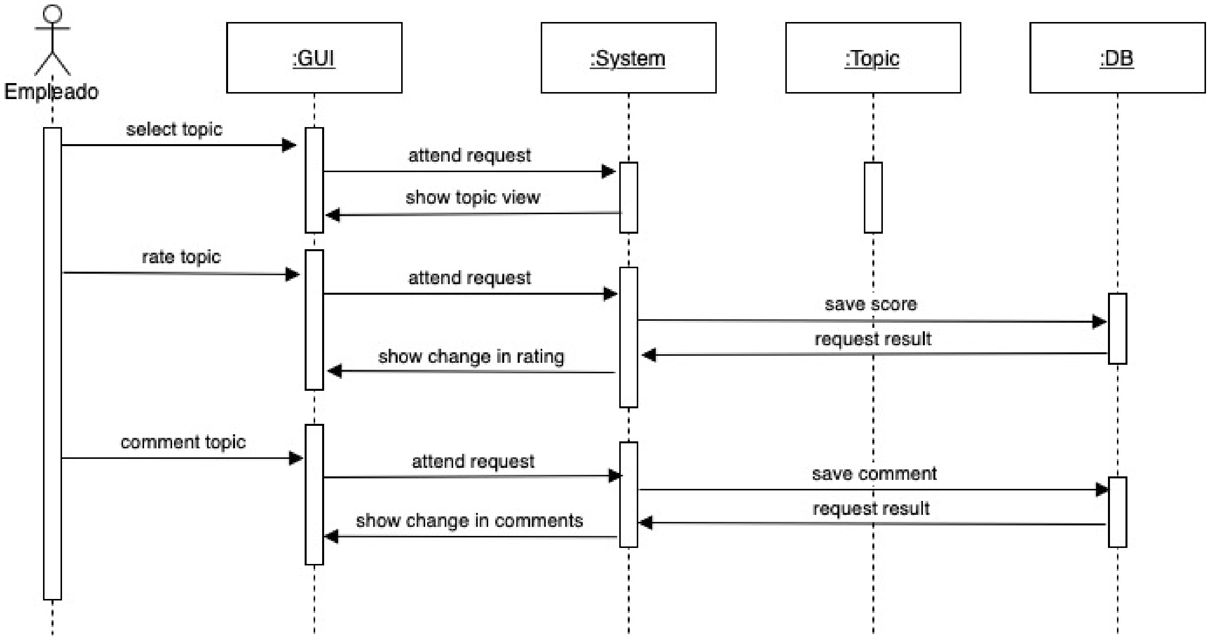
- Update Topic: Describes the employee's process to update a topic and how it asks the system for the topic stored, shows the interfaces to edit it, captures the data and saves and publishes the changes and notifies the interface



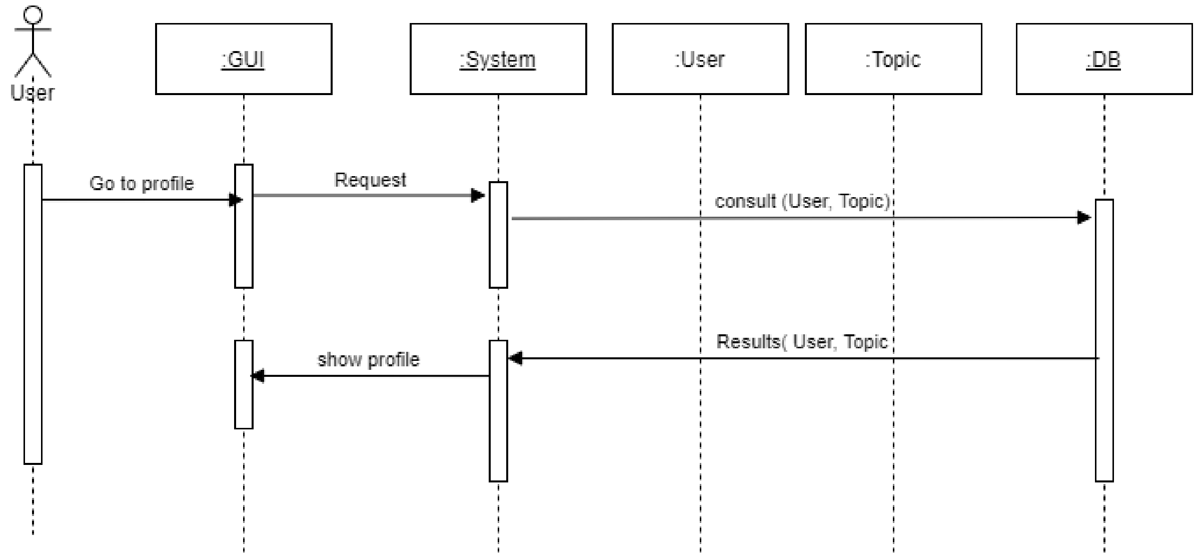
- Search Topic: Describes the employee's process to search for a topic and how it asks the system for the topic stored in the DB, consults the topics and returns a list to the interface of the related results.



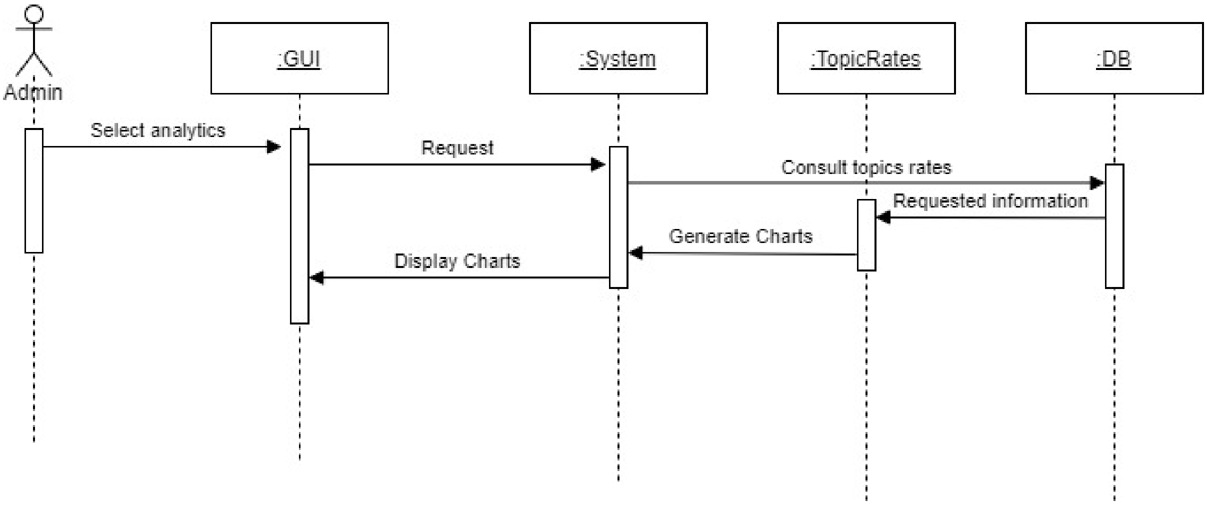
- Consult Topic: Describes the employee's process to access a topic and how it asks the system for the topic stored in the DB, consults the topic in the interface and how it is able to comment and rate a topic



- Consult Profile:Describes the employee's process to access their profile and how the system asks for the topics created by the user, and their own information and shows it in the interface.



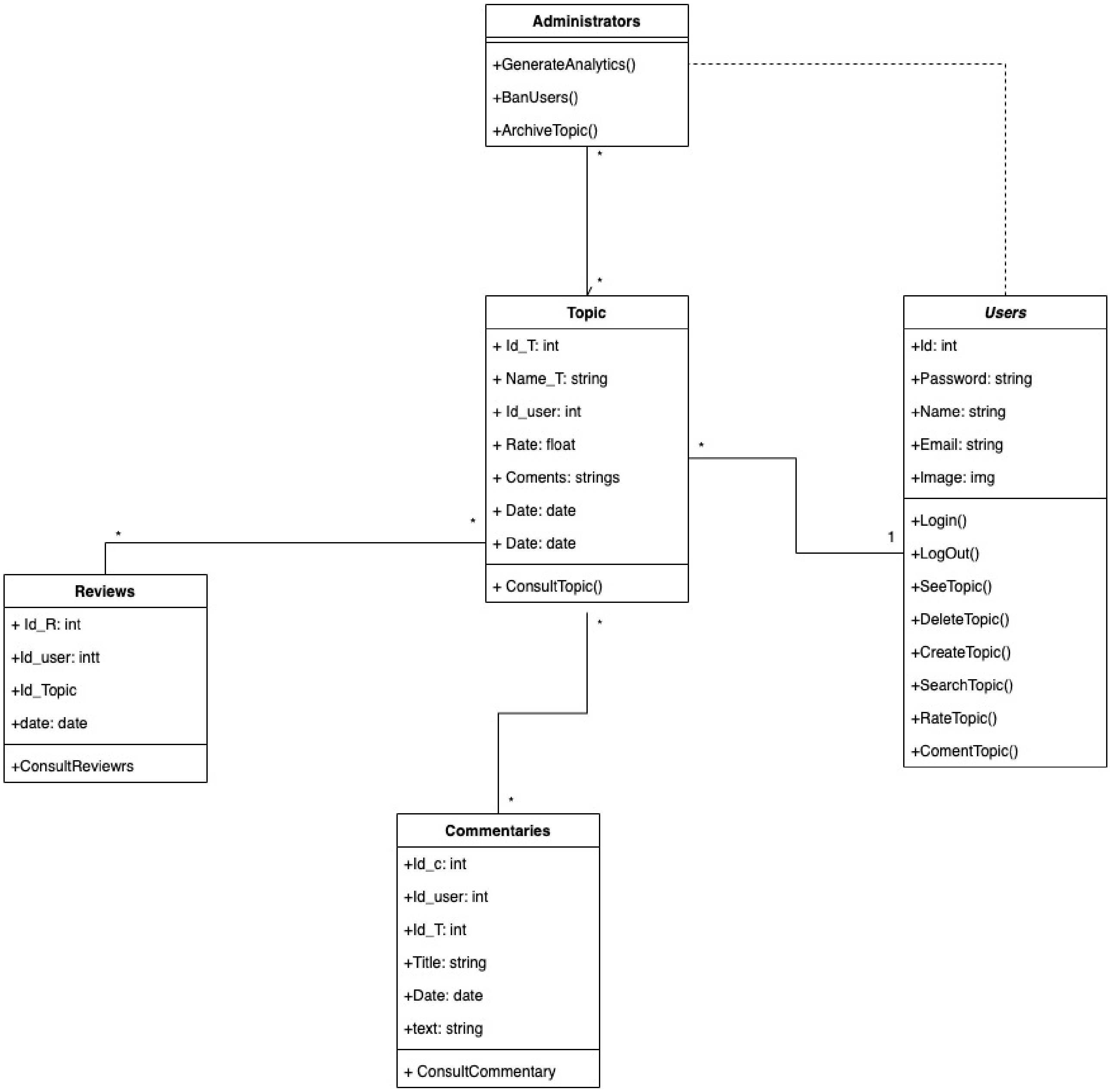
- Consult Analytics: Describes the administrator's process to access the analytics and how the system accesses the ratings of the topics stored in the DB and based on these generates a graph to display in the UI



- A class diagram for the objects that will be instanced, for the previously mentioned aspects (Include at least 5 elements of the diagram) This diagram is responsible for describing what should be in the system that is being modeled, this has an OOP approach since these classes that are created have attributes and methods, in addition to that with this it can be understood how it is that these are related and as well understand how the system or application works.

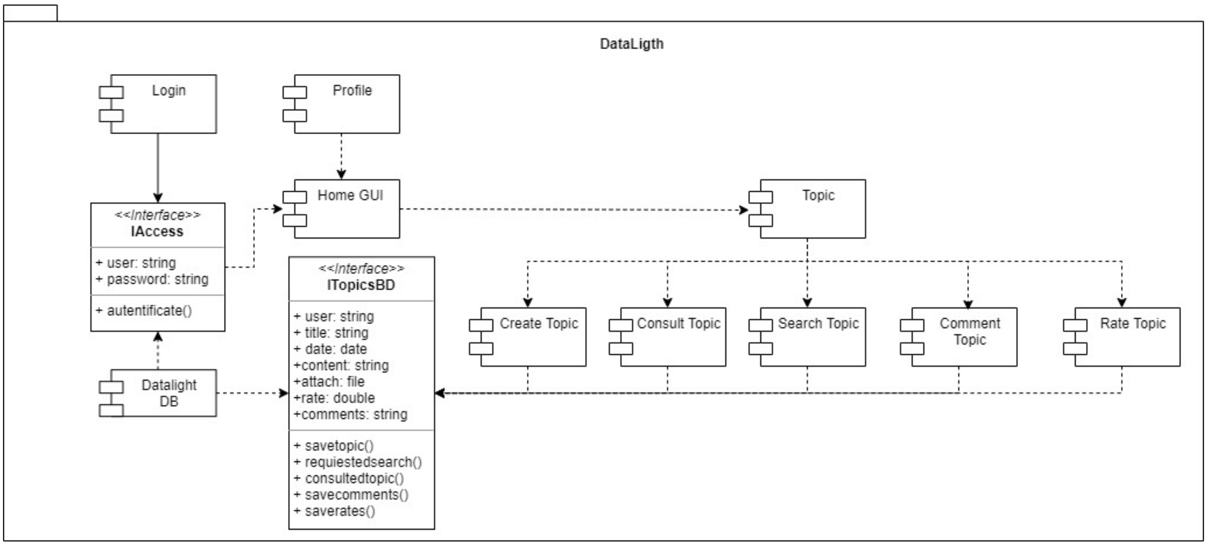
  ### The components of this diagram are:

  - Class: this is a rectangle composed of three parts, in the upper one is the name of the class, in the next part that is in the middle are the attributes, this section is used to describe qualities of the class, and finally at the bottom are the methods, this section includes a list of all the operations that the class can perform.
  - Association: It is a line which allows to establish the relationship between two classes.
  - Packages: It is a figure in the form of a folder that represents a set of classes.
  - Comments: They are a type of rectangle with a double in the upper right part, which allows to make a comment in case any attribute or method is not very clear.
  - Suspension points: They are three points that are placed under the attributes or methods so that it is known that there are more than those that were written.

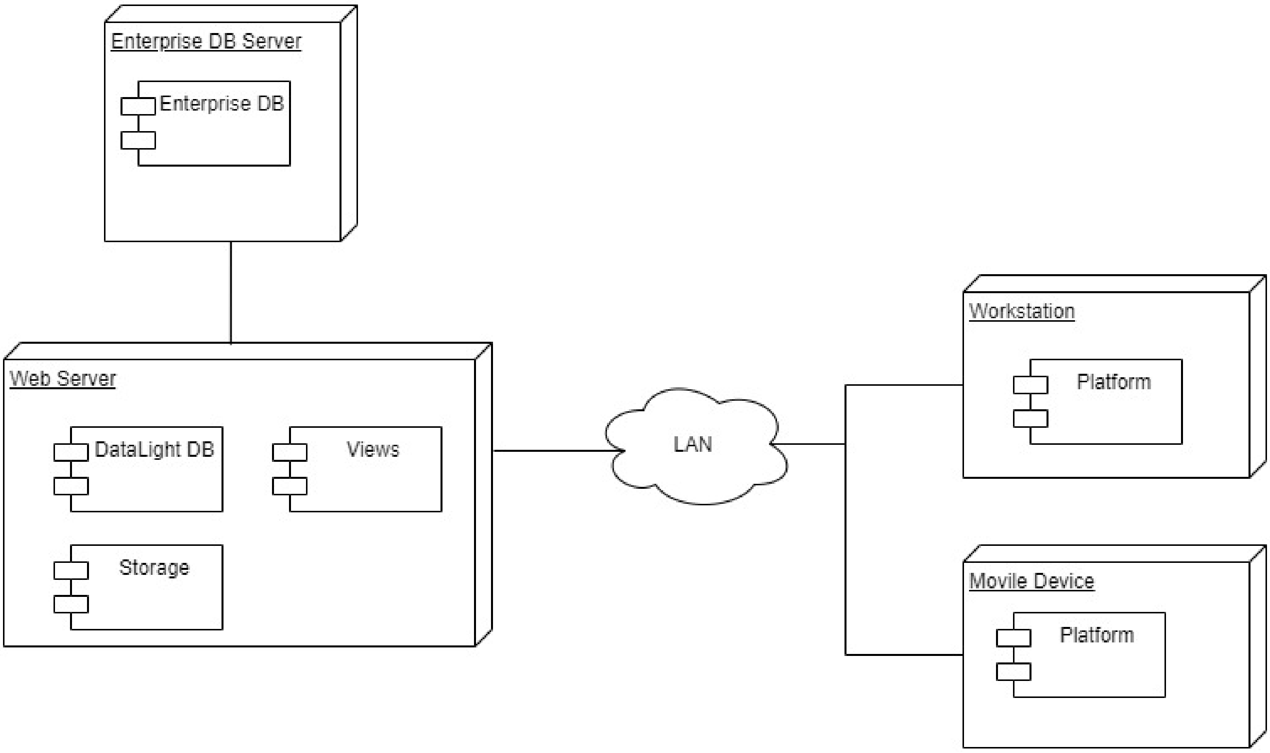


- A package diagram containing the component diagrams and the relationship between the database, user interfaces, controllers or intermediate layers - middleware (Include at least 3 elements of the diagram)
    - This diagram represents the objects and components that interact in the system, here the login is shown and this is related to an access interface that is connected to the database of the platform and how the employees and these to a profile and a topic, in the topics are all the possible actions with them and how they are connected to another topic interface and it is connected to

the DB



- The distribution diagrams representing the system's physical structure, such as physical infrastructure, networks, storage and web servers, firewall, mobile devices or any other physical resource that will be part of the system (include at least 3 elements of the diagram)
  - This diagram represents how the system is structured, see the different hardware and software devices and how they are connected to each other. Each device is represented by a node and each node has its own component that makes it up. It can be seen how different devices using the platform are connected through the network, to the web service where the views can be found and where files are stored. The Web service is connected to the company's database where it can access user and topic data.



Conclusions

Arredondo Bonilla Cesar

With this practice, it was possible to understand how these diagrams are correctly integrated into what was proposed in the case study, since we had to analyze the different diagrams that each one had and be able to select the one that was best structured and that if it complied with what is established in the case study in order to make the necessary corrections and use it. During the practice there were some difficulties, as was the case with the distribution diagram since this had a wrong idea of what it was like, and it had to be re-analyzed to establish what physical structure could exist within the company where it would be installed the solution that we are going to develop.

Chavez Vargas Javier

The UML diagrams of this activity teach us to better understand how the platform of the case study works, by doing these UML we realize how the process of each activity is and how it should be, what it is composed of, how it is built and who interacts In it, based on these, it is possible to analyze very well how to plan and develop the implementation platform well. When creating these diagrams it is a bit tiring since they can be several depending on the number of use cases you have, you also have to think carefully about what components each UML carries and how they are related, you have to pay close attention to how you place them and what wants it to represent because if you put it wrong it can represent something different from the one you want to pose

Mancilla Mora Moises

By developing the diagrams in the previous activities, the team members were able to analyze our diagrams and consolidate this information in order to develop better diagrams. By putting the activities together we were able to be clear about all the use cases that our platform would have, with that we elaborated the sequence diagrams where we saw how the processes would be in each use case. With the component diagram, all the functions of the platform were analyzed, how they would be structured and how they would interact with the database. It was re-employed how the structure of the system would be and how the different devices are connected. By developing these diagrams, it was possible to have a better idea of how our platform would work and thus be able to continue working on it.

Valerde Sanchez Alejandro

During software development it is important to document and capture the ideas behind the design and operation of the platform. With this practice, different types of models were used that capture the interaction and main components of our project. The teamwork was carried out with the help of the models that we previously made individually. These models were that of classes, sequence, components, and distribution. Each member contributed to the final delivery and we await comments and feedback.

💣 Rubric

Criteria	Description	Score
Instructions	Is each one of the points indicated in the instructions section fulfilled?	10
Development	Was each one of the points requested within the development of the activity answered?	60

Criteria	Description	Score
Demonstration	Is the student present during the explanation of the functionality of the activity?	20
Conclusions	Is a personal opinion of the activity included by each of the team members?	10

🏡 My Github