Introduction

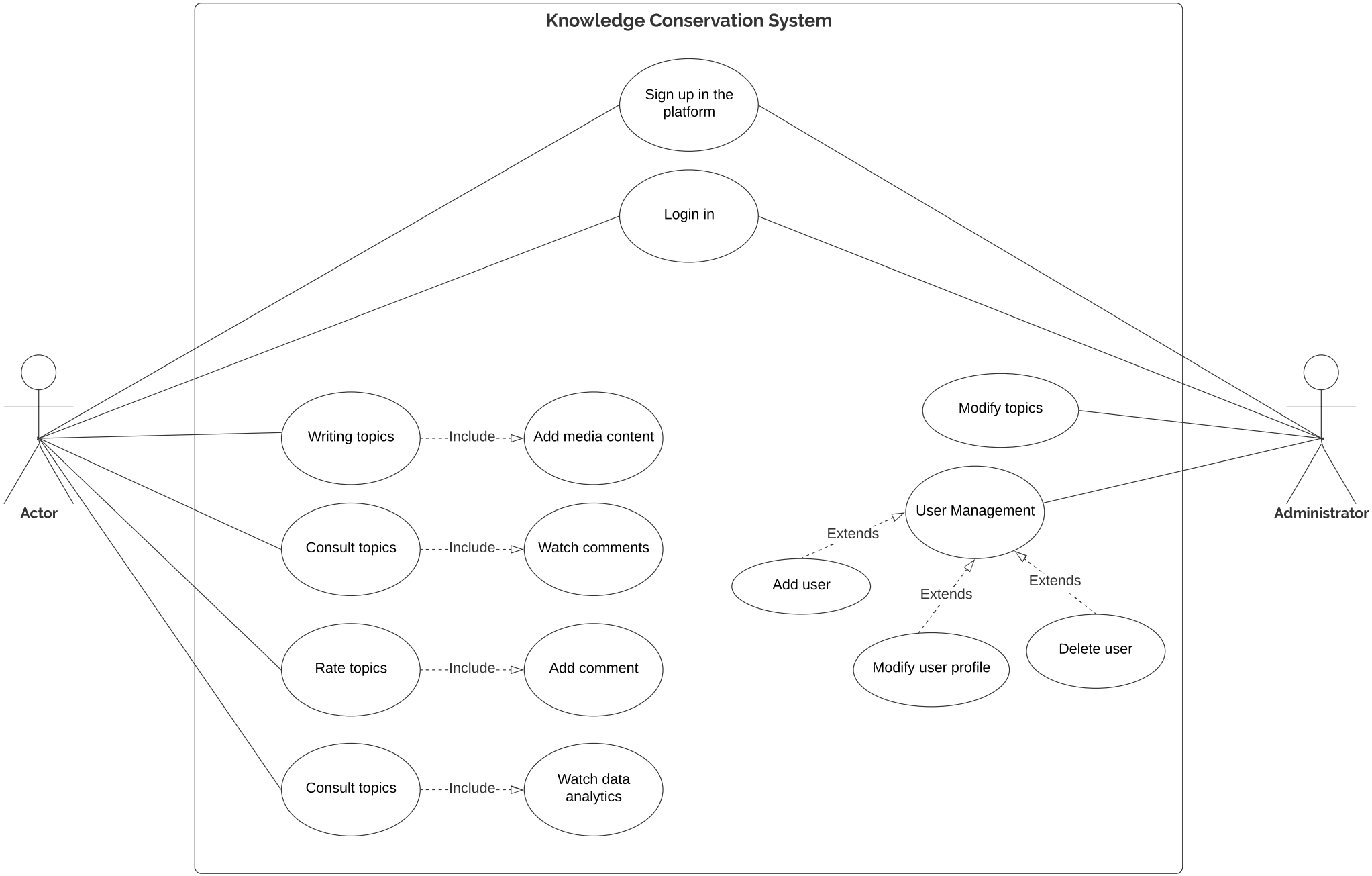
🏆 A1.1 Learning Activity

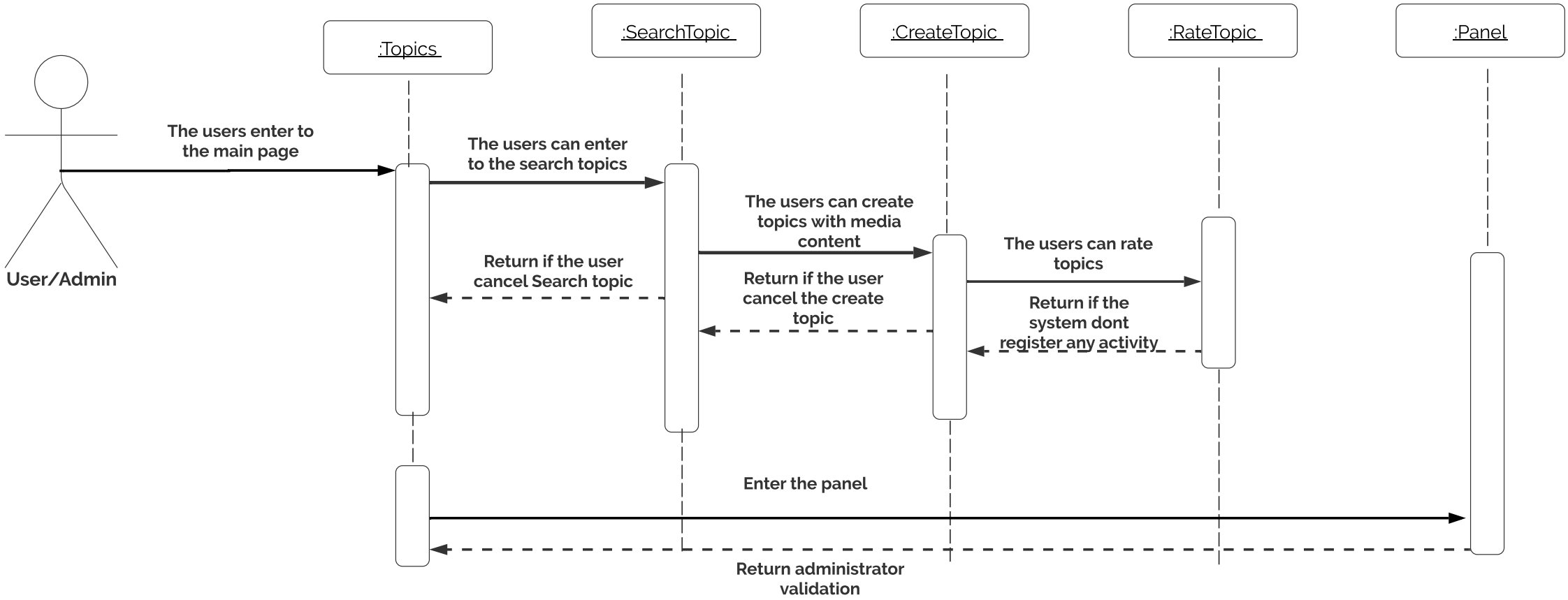
Modeling requirements through UML diagrams.

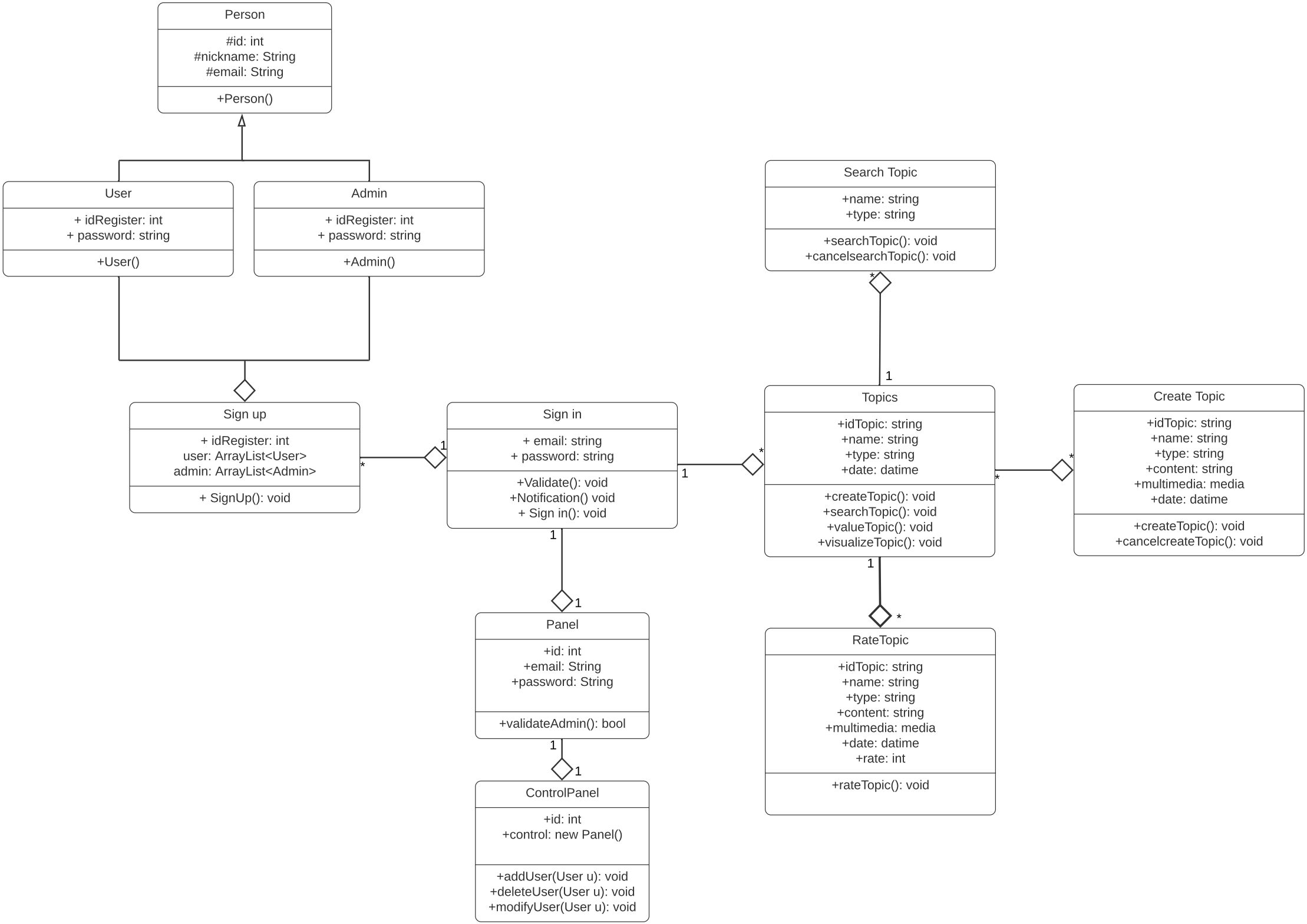
✏️ Development

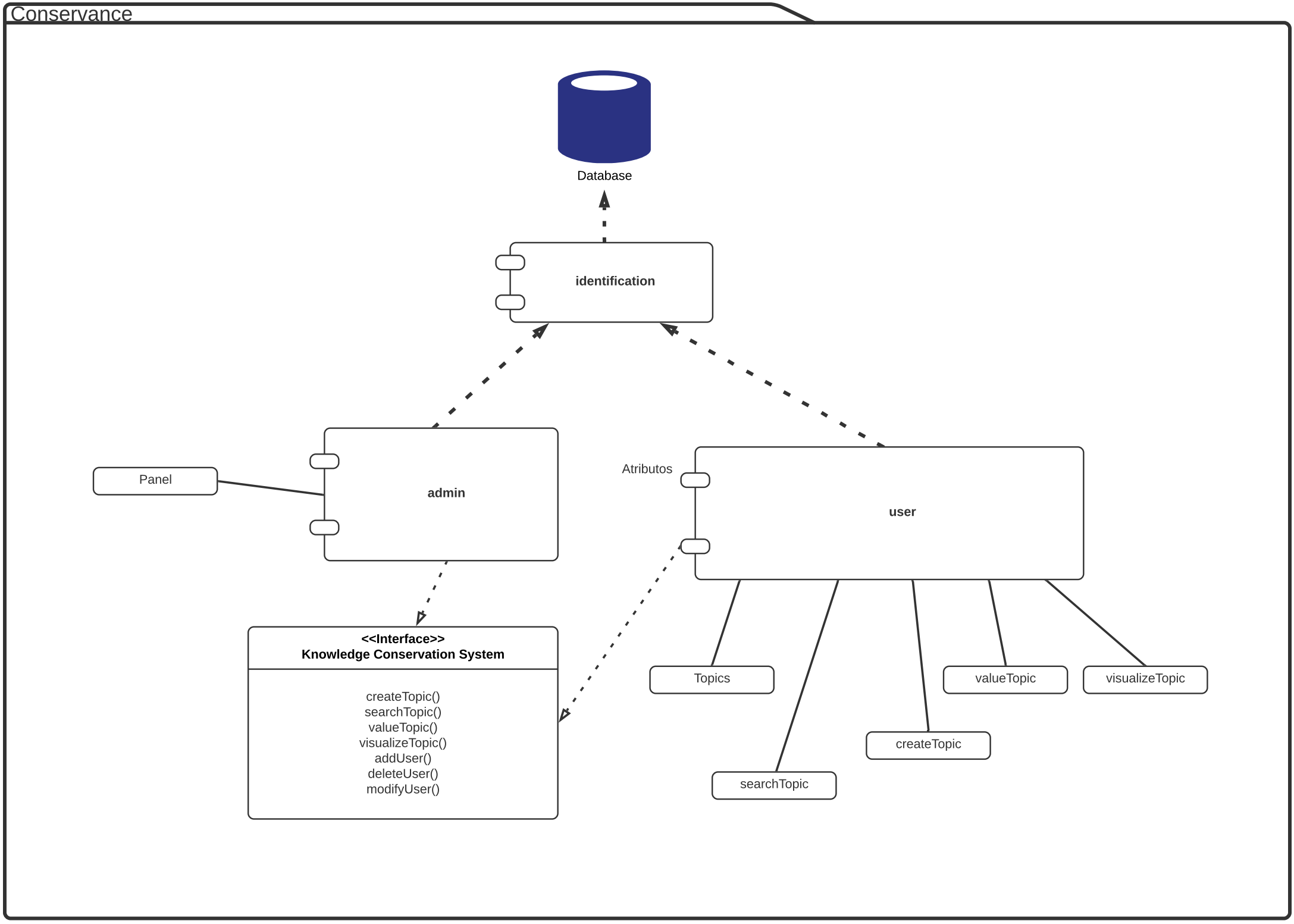
1. Based on each of the user and system requirements set out above for the case study, draft the requested modeling diagrams:

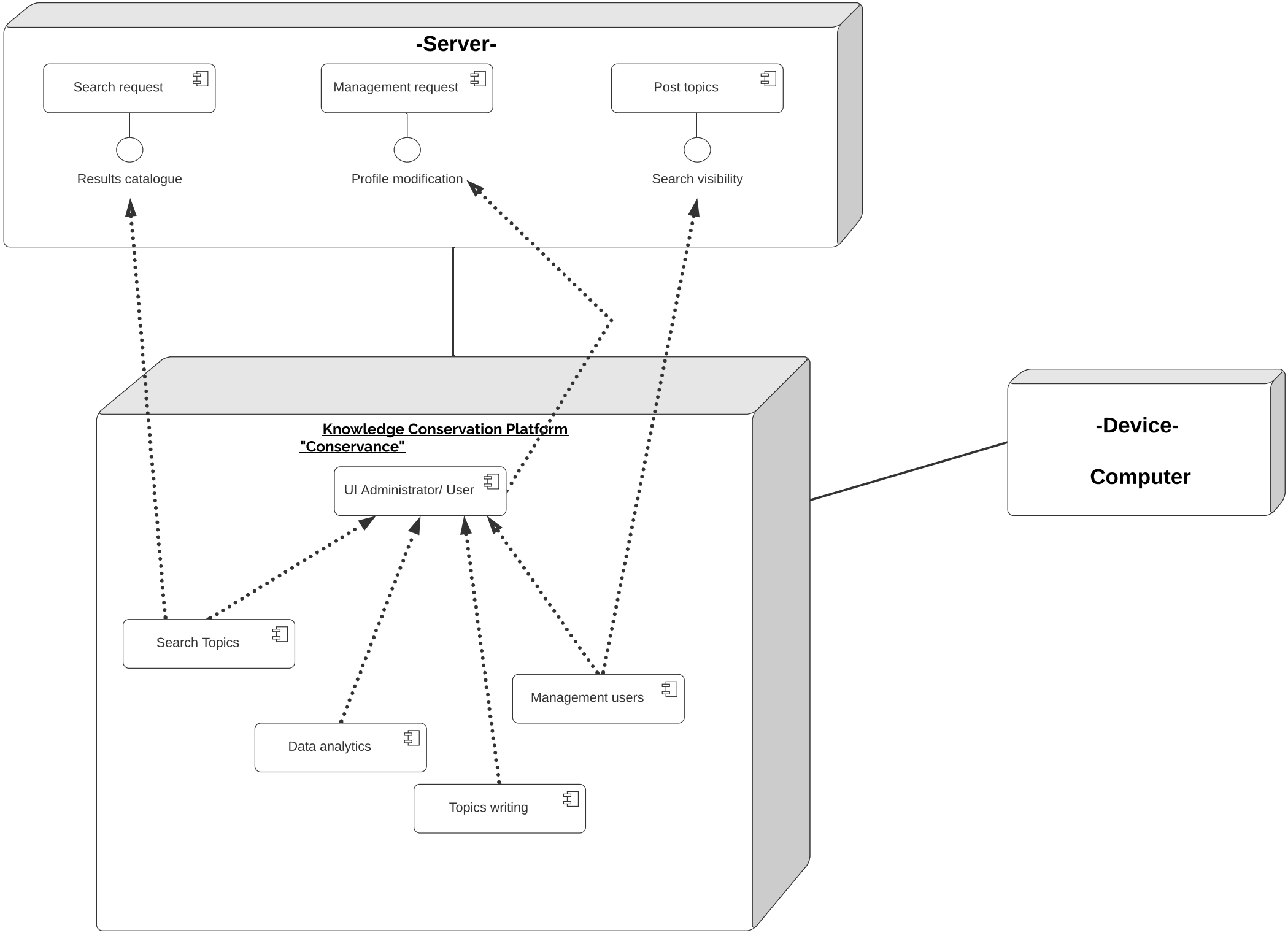
- A diagram that contains the relationships between the different uses cases, only for the scenario in which the users interact with the system making use of the main or central functions according to each case study. (Include at least 5 elements of the diagram)
- The sequence diagram that explains the communication between the different GUIs for the use cases of the previous point, considering that the user has already successfully entered the system. (Include at least 5 elements of the diagram)
- The class diagram for the objects to be instantiated, for the points mentioned above. (Include at least 5 elements of the diagram)
- A package diagram containing the component diagrams and the relationships between the database, user interfaces, controllers, or middleware. (Include at least 3 elements of the diagram)
- The distribution diagrams representing the physical structure of the system such as physical infrastructure, networks, storage and web services, firewall, mobile devices or any other physical resource that will be part of the system. (Include at least 3 elements of the diagram)


Knowledge Conservation System
Sign up in the platform
Login in
Actor
Writing topics
Include
Add media content
Consult topics
Include
Watch comments
Rate topics
Include
Add comment
Consult topics
Include
Watch data analytics
Modify topics
User Management
Extends
Add user
Extends
Modify user profile
Extends
Delete user
Administrator


:Topics
:SearchTopic
:CreateTopic
:RateTopic
:Panel
User/Admin
The users enter to the main page
The users can enter to the search topics
The users can create topics with media content
The users can rate topics
Return if the user cancel Search topic
Return if the user cancel the create topic
Return if the system dont register any activity
Enter the panel
Return administrator validation


Person
#id: int
#nickname: String
#email: String
+Person()
User
+ idRegister: int
+ password: string
+User()
Admin
+ idRegister: int
+ password: string
+Admin()
Sign up
+ idRegister: int
user: ArrayList<User>
admin: ArrayList<Admin>
+ SignUp(): void
Sign in
+ email: string
+ password: string
+Validate(): void
+Notification() void
+ Sign in(): void
Panel
+id: int
+email: String
+password: String
+validateAdmin(): bool
ControlPanel
+id: int
+control: new Panel()
+addUser(User u): void
+deleteUser(User u): void
+modifyUser(User u): void
Search Topic
+name: string
+type: string
+searchTopic(): void
+cancelsearchTopic(): void
Topics
+idTopic: string
+name: string
+type: string
+date: datime
+createTopic(): void
+searchTopic(): void
+valueTopic(): void
+visualizeTopic(): void
Create Topic
+idTopic: string
+name: string
+type: string
+content: string
+multimedia: media
+date: datime
+createTopic(): void
+cancelcreateTopic(): void
RateTopic
+idTopic: string
+name: string
+type: string
+content: string
+multimedia: media
+date: datime
+rate: int
+rateTopic(): void


Conservance
Database
identification
Panel
admin
Atributos
user
<<Interface>>
Knowledge Conservation System
createTopic()
searchTopic()
valueTopic()
visualizeTopic()
addUser()
deleteUser()
modifyUser()
Topics
searchTopic
createTopic
valueTopic
visualizeTopic


-Server-
Search request
Management request
Post topics
Results catalogue
Profile modification
Search visibility
Knowledge Conservation Platform
"Conservance"
UI Administrator/ User
Search Topics
Management users
Data analytics
Topics writing
-Device-
Computer

Conclution by Edson:

The use of the diagrams and their implementation based on both a case study and a project are useful and necessary tools that give us an overview of how a project can be proposed or implemented and the course that it must follow to reach a final product. Through diagrams of classes, components, sequence, use cases, distribution, modules, sections or user interaction with the system itself can be identified.

Conclution written by Antonio:

It can be concluded that through the use of various diagrams , we can observe that many objects , components , classes , relations exist in our project , and in this way realize if we were missing things by adding or had things that remain , similarly we provides an overview of the project in the sense of how many things we have to take into account , that things should be planned. The help of the planning in the diagrams helps us to continue on a fixed path without deviating from the requirements that the client requested. It took a while to make the diagrams because of the decision to know what elements could correspond to the idea we wanted to convey , but I believe that the objective was achieved.

Conclution written by Joseph

Diagrams are an important complement to carrying out the sequence of a project. The diagram that caught my most attention was the class diagram because from that diagram you can build a functional software with all the classes, attributes and methods that are mentioned as well as the interfaces and abstract classes. In this diagram the relationships between classes are also carried out for the time of programming can be visualized and so that the developers can support themselves by observing and carrying out an analysis of it.

The diagram that became a little more complicated in my opinion was the sequence one since it is a diagram that had been worked Little and the idea is to build the sequential ideas with objects

Go to my GitHub