Tunisian Republic Ministry of Higher Education and Scientific Research

University of Carthage

Higher Institute of Information Technologies and Communication

Report
Of
Requirements specification
“LearnUp”

Subject:

develop educational website

Prepared by:

Adem Abassi, Adem Bouadila &

Oussama Meddeb

College year 2023-2024

Summary:

1.Introduction:

1.1. Description:

"Learn Up" is an educational app within the online learning domain, designed to help students overcome time constraints and understand difficult subjects. It offers a user-friendly

interface for searching courses, sharing materials, and engaging in discussions, with a tagging system for efficient navigation.

1.2. Objective:

This report presents the "Learn Up" project, an educational platform designed to help students overcome common challenges such as time constraints and difficulties understanding certain subjects. The application aims to be a supportive companion for those pressed for time or in need of additional clarity, especially as exams approach. With its user-friendly interface, the application offers a seamless experience, allowing users to easily search for specific courses, share study materials, and engage in discussions to deepen their understanding. The tagging system ensures efficient navigation, making it easy to locate relevant subjects, resumes, or inquiries.

2.Functional Requirements Specification:

2.1. Requirements:

The project aims to create a study website that meets the following needs:

- Manage account
- Document management
- Voting
- Authentication
- Discussion
-Verifying documents

2.2. Actors Identification:

An actor is the role assumed by a user interacting with the developed system:

User: Any person accessing the LearnUp platform has the capability to browse courses, exams, and engage in chat. Additionally, users can add, download, vote on courses and exams, or participate in discussions with others.

The admin: This is the supervisor who possesses all control permissions to manage users and verify and delete their courses, exams and messages.

3.Non-Functional Requirements Specification:

Non-functional requirements are like quality checks for your app's features. They're super important because they impact how well the app works for users, even though they're not about the specific things the app does. Here's what you need to think about:

- Reliability: The app should work smoothly without any hiccups, making sure users are happy with how it performs.

- Error Handling: If something isn't clear or goes wrong, the app should explain it clearly to users so they can understand and keep using it.

- User-Friendly Design: The app should be easy and pleasant to use, so users don't have to struggle with finding things or understanding what they see.

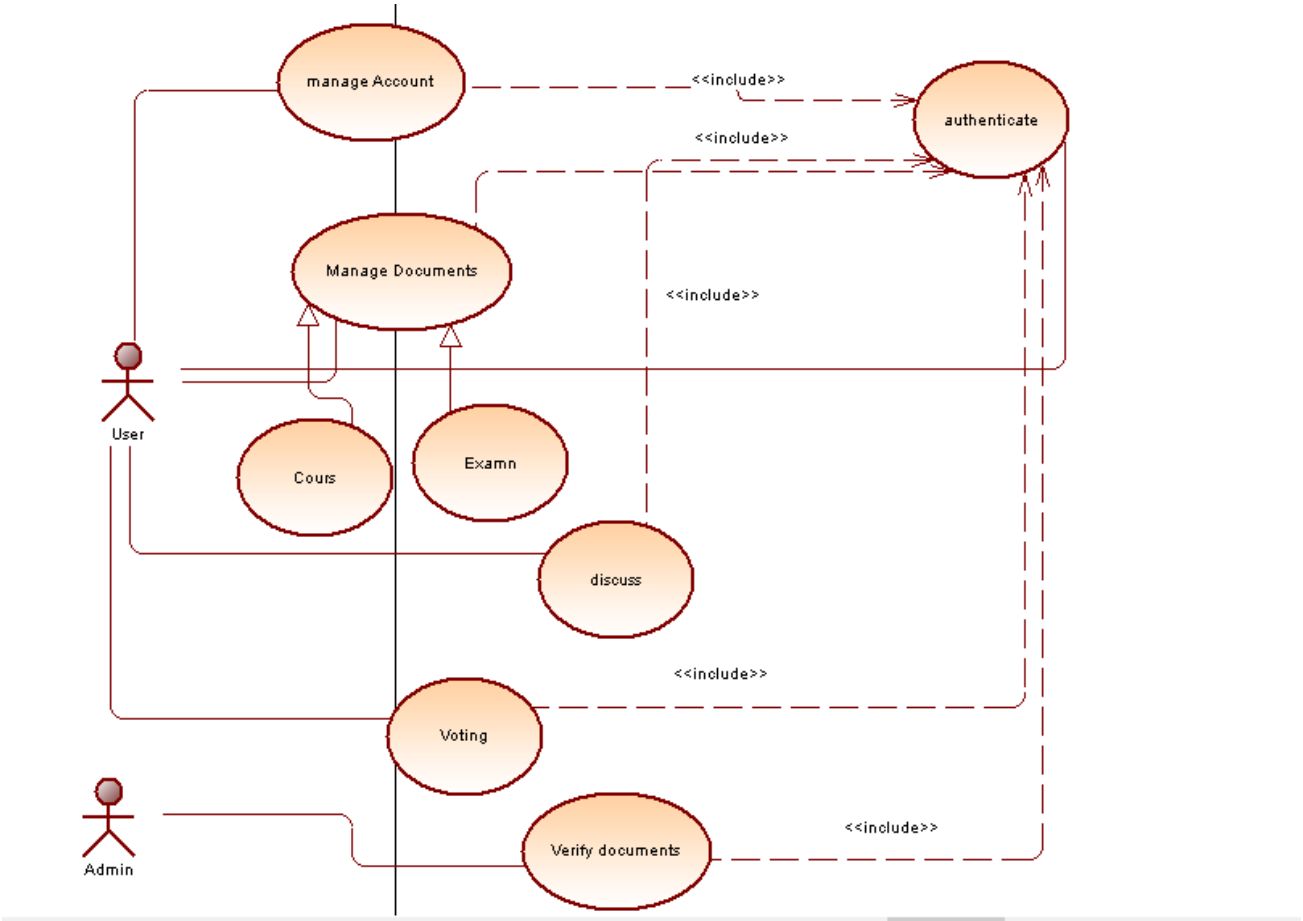
- Security: It's crucial to keep user’s personal info safe, so the app needs to be designed to protect their data.

- Maintenance and Reusability: Make sure the app is built in a way that makes it easy to update and use parts of it again in other projects.

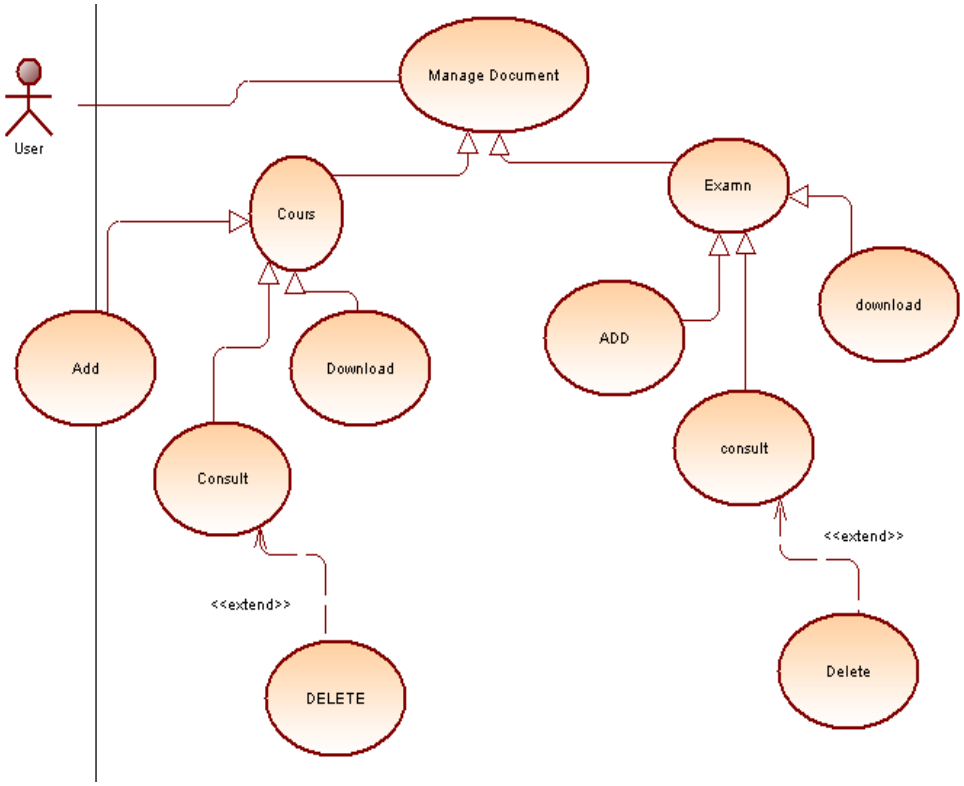
4.Conception UML:

4.1User case diagrams:

1.Global use case diagram:



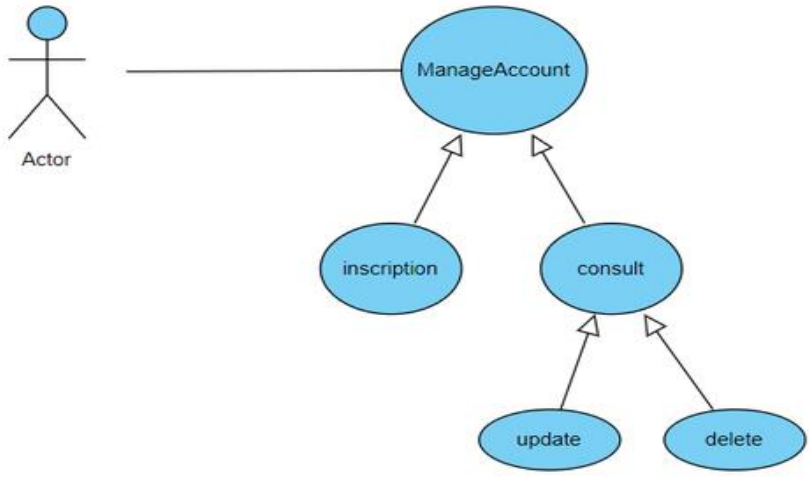
2.Manage document diagram:



The following table elaborates on this user story with textual description:

	As a user, I can manage documents
Actors	User
Pre-Condition	The user must be connected
Post-Conditions	Document managed
Main scenario	The system will give the user the ability to have limited control and interactions with the presented documents in the website (courses/exams)
Extensions	• Add • Download • Consult • Delete

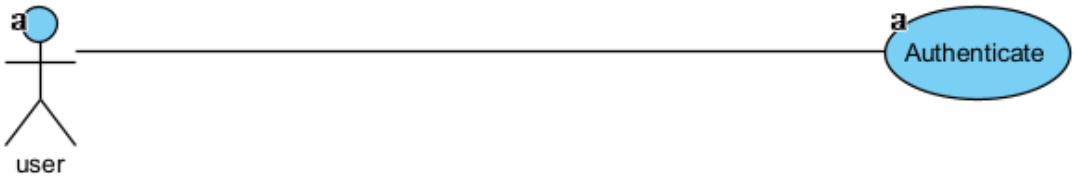
3.Manage account diagram:



The table below provides a detailed description of this user story in text.:

Use case scenario	As a user, I can manage my account
Actors	User
Pre-Condition	-The user must be inscripted
Post-Conditions	Account managed
Main scenario	The system will show options to the user to have full control over his personal account and information
Extensions	• Inscription • Consult account • Update account • Delete account

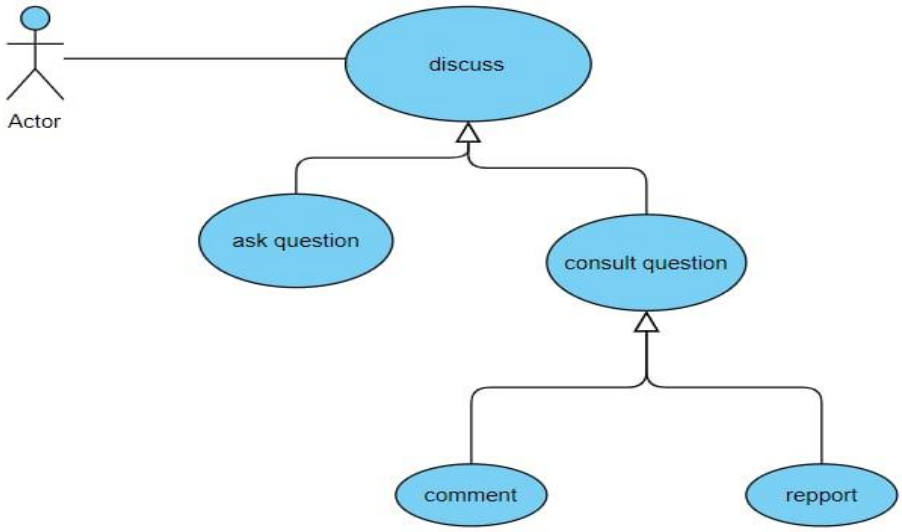
4.Authentification:



The table below provides a detailed description of this user story in text.:

Use case scenario	as a user, I can be authticated
Actors	User
Pre-Condition	User must be registered
Post-Conditions	User authticated
Main scenario	The system will allow the user to access and use the platform.

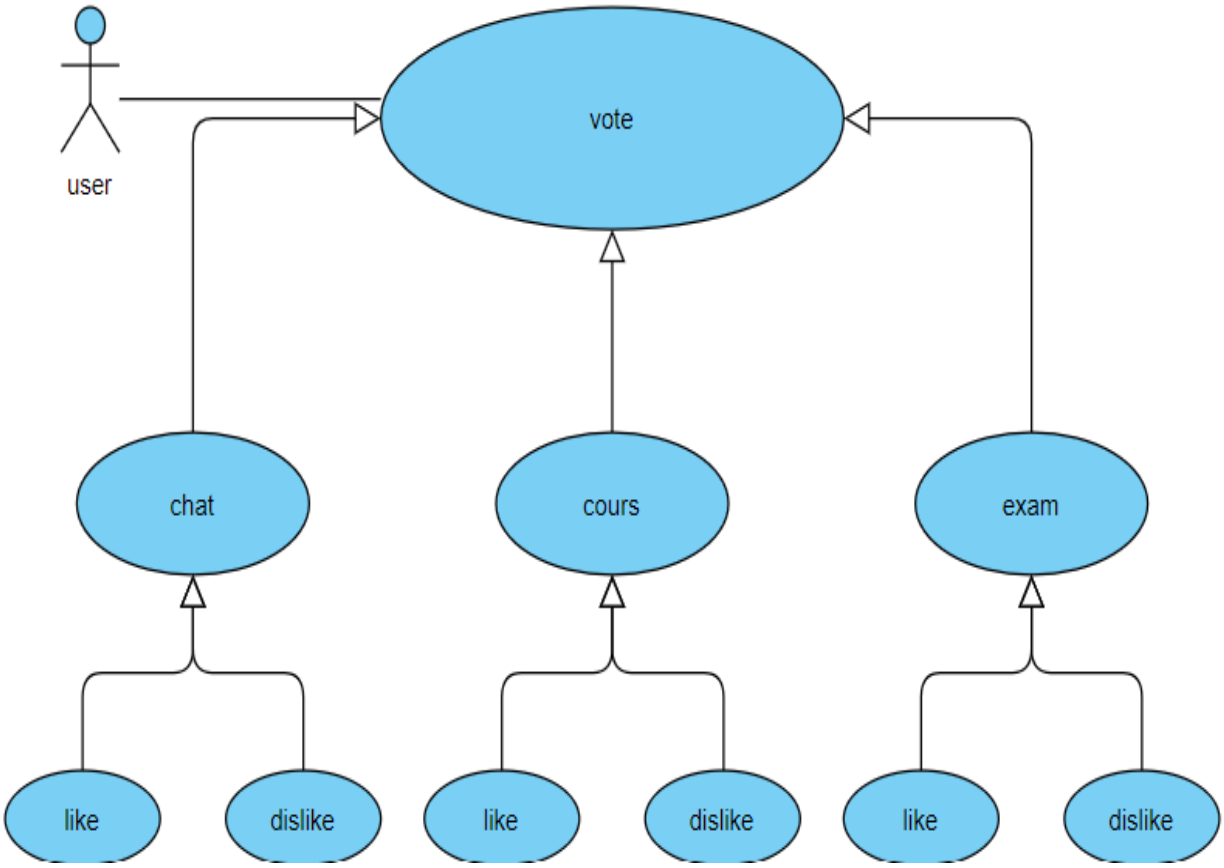
5.Discuss diagram:



The table below provides a detailed description of this user story in text.:

Use case scenario	As a user, I can discuss on the chat
Actors	User
Pre-Condition	-The user must be inscripted
Main scenario	The system will post report, or delete a post in the community chat
Extensions	• Ask question • Consult question • comment • repport

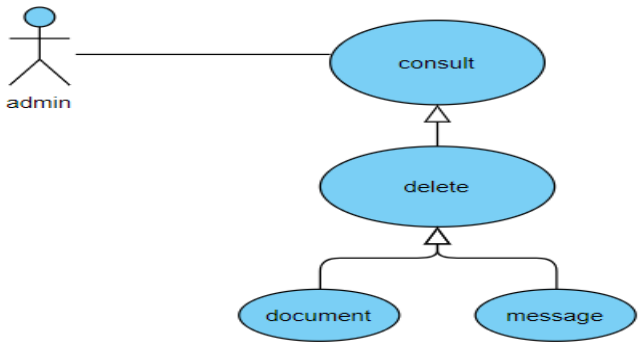
6.Voting diagram:



The table below provides a detailed description of this user story in text.:

Use case scenario	As a user, I vote by like or dislike
Actors	User
Pre-Condition	-The user must be inscripted
Post-Conditions	Vote sent
Main scenario	The system will add a like or dislike on a post
Extensions	• Chat • Course • Like • Dislike

7.Admin verify diagram:



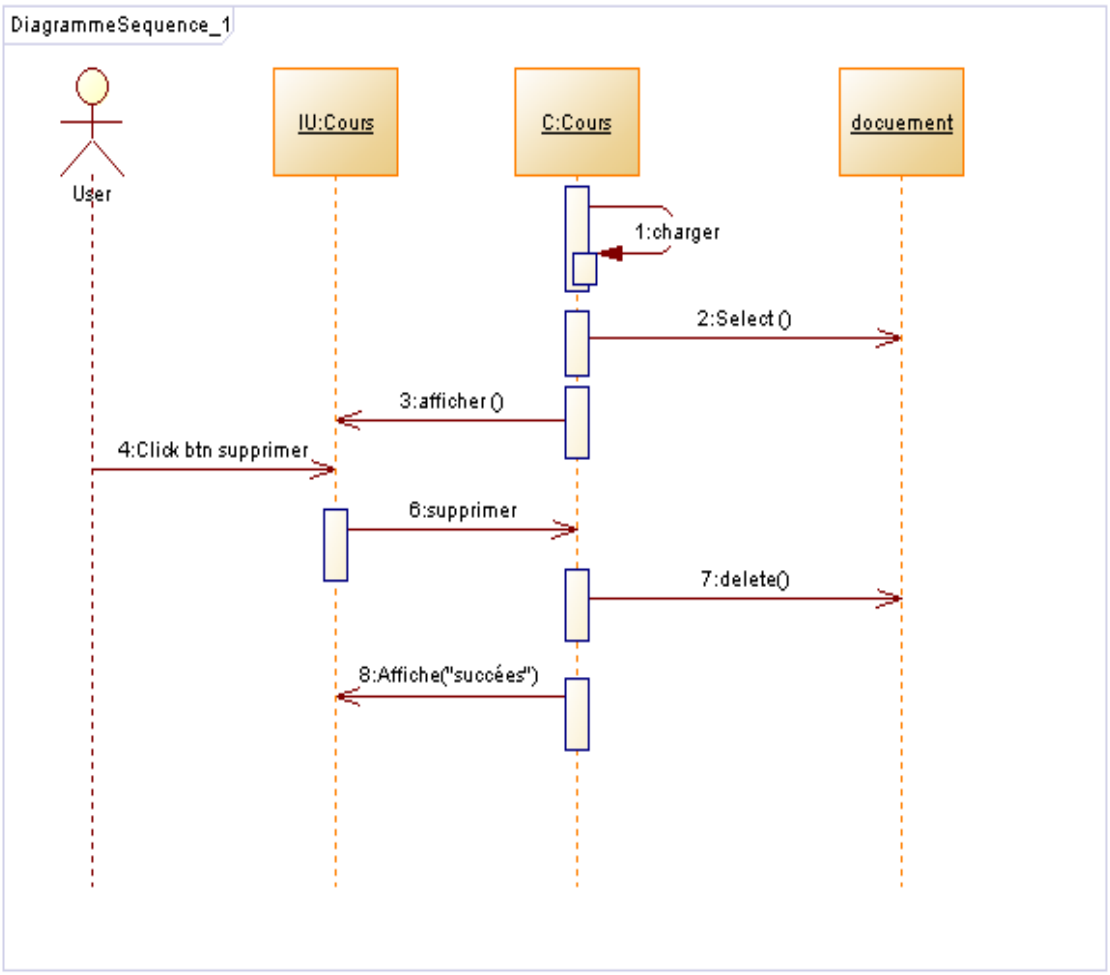
The table below provides a detailed description of this user story in text.:

Use case scenario	As a admin, I can consult documents and massages
Actors	Admin
Pre-Condition	-The post must be reported
Post-Conditions	Post deleted (optional)
Main Conditions	The system will delete the reported post

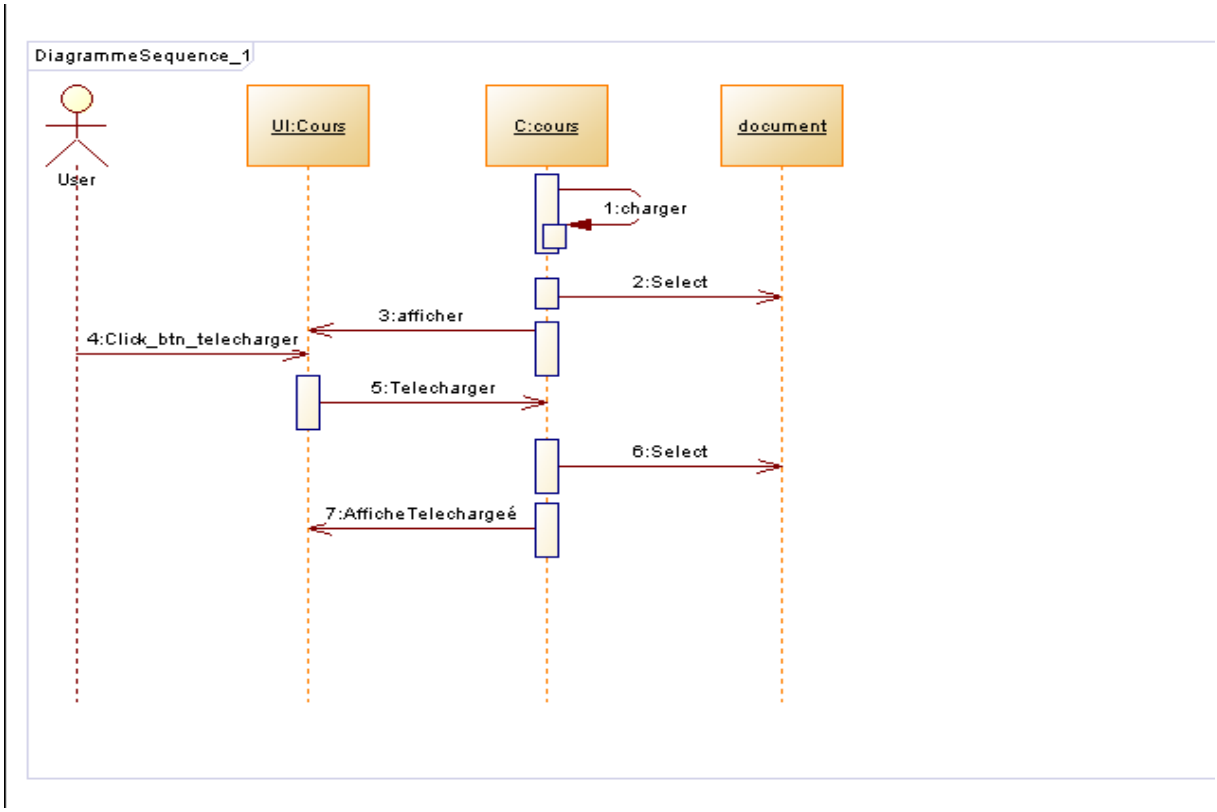
Extensions	• delete • document • message

4.2. Sequence diagrams:

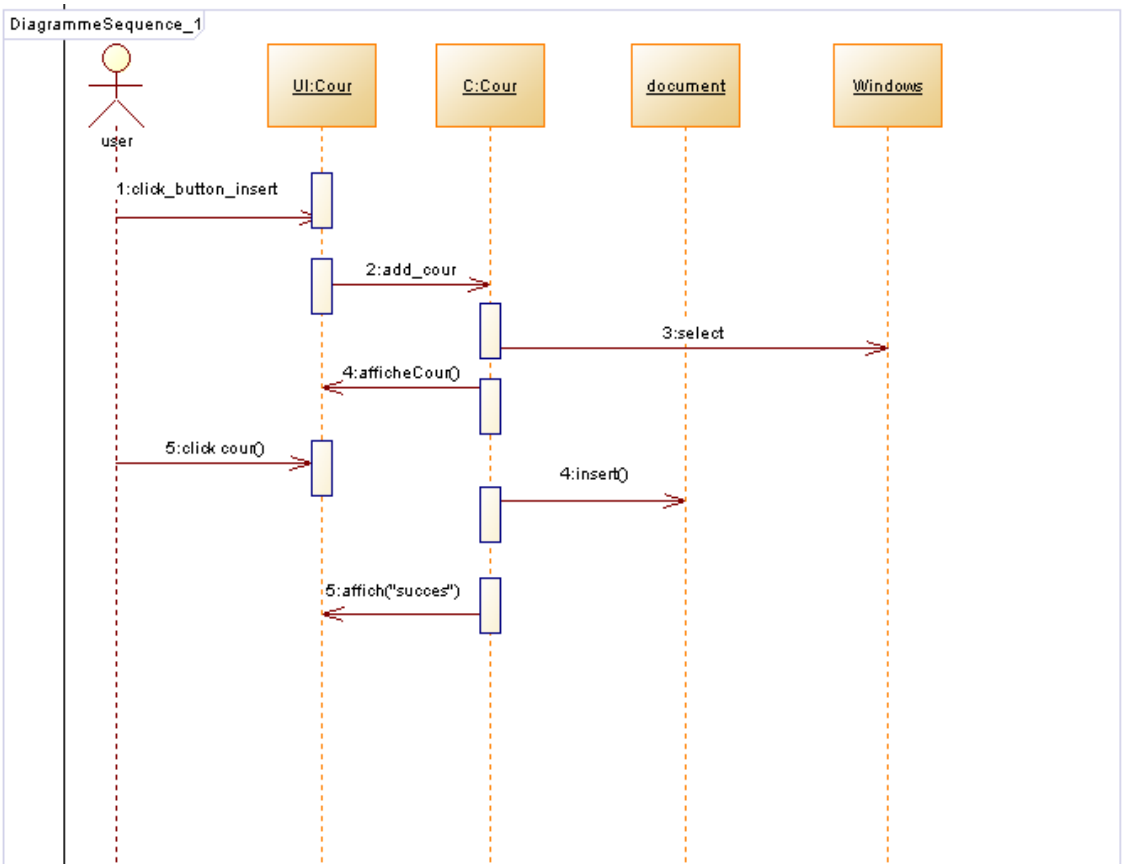
1.The sequence diagrams of “delete course”:



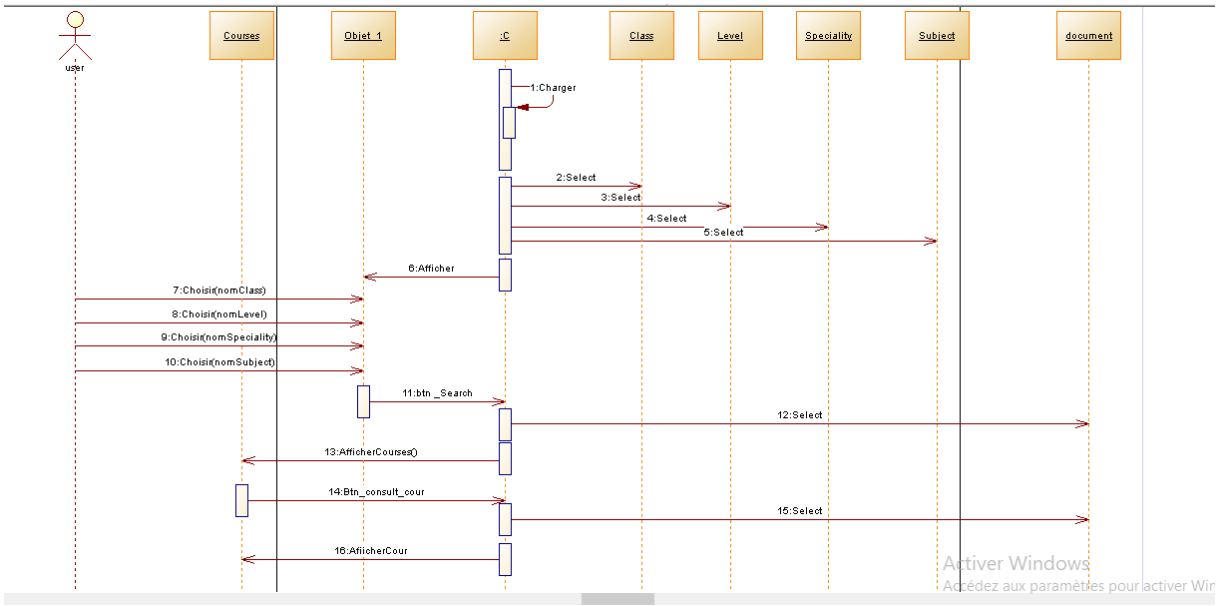
2.The sequence diagrams of “download course”:



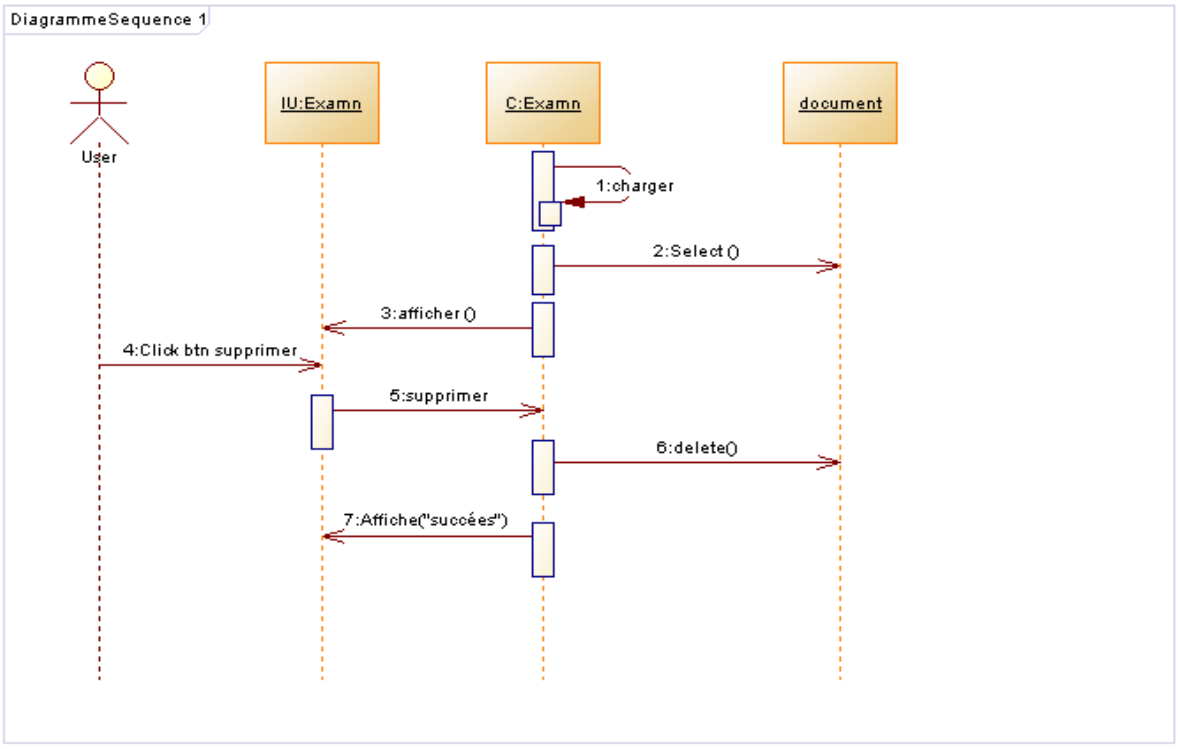
3.The sequence diagrams of “add course”:



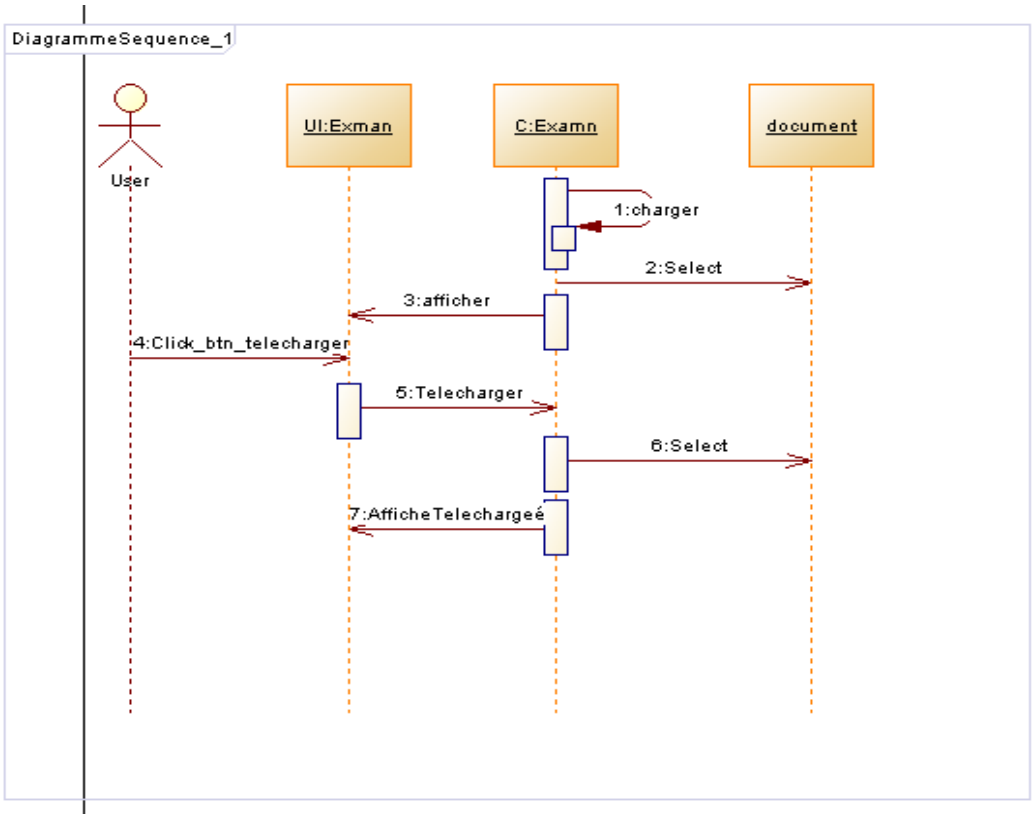
4.The class and sequence diagrams of “consult course”:



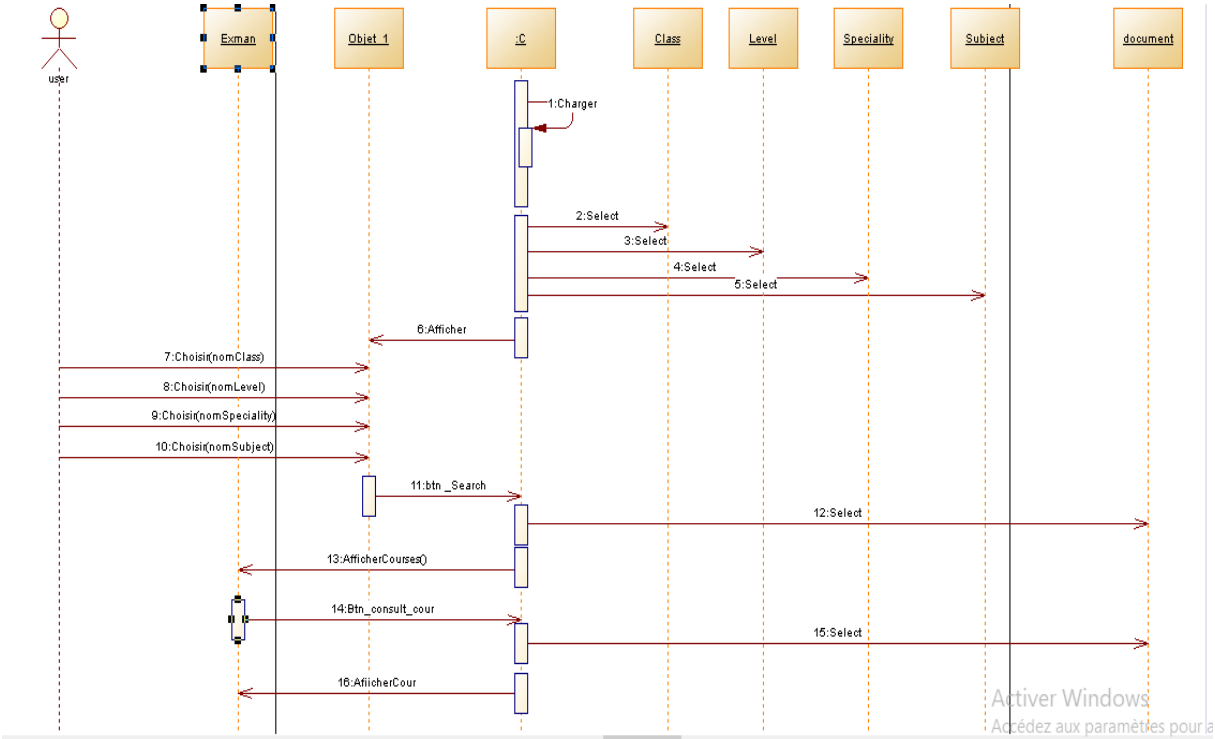
5.The sequence diagrams of “delete exam”:



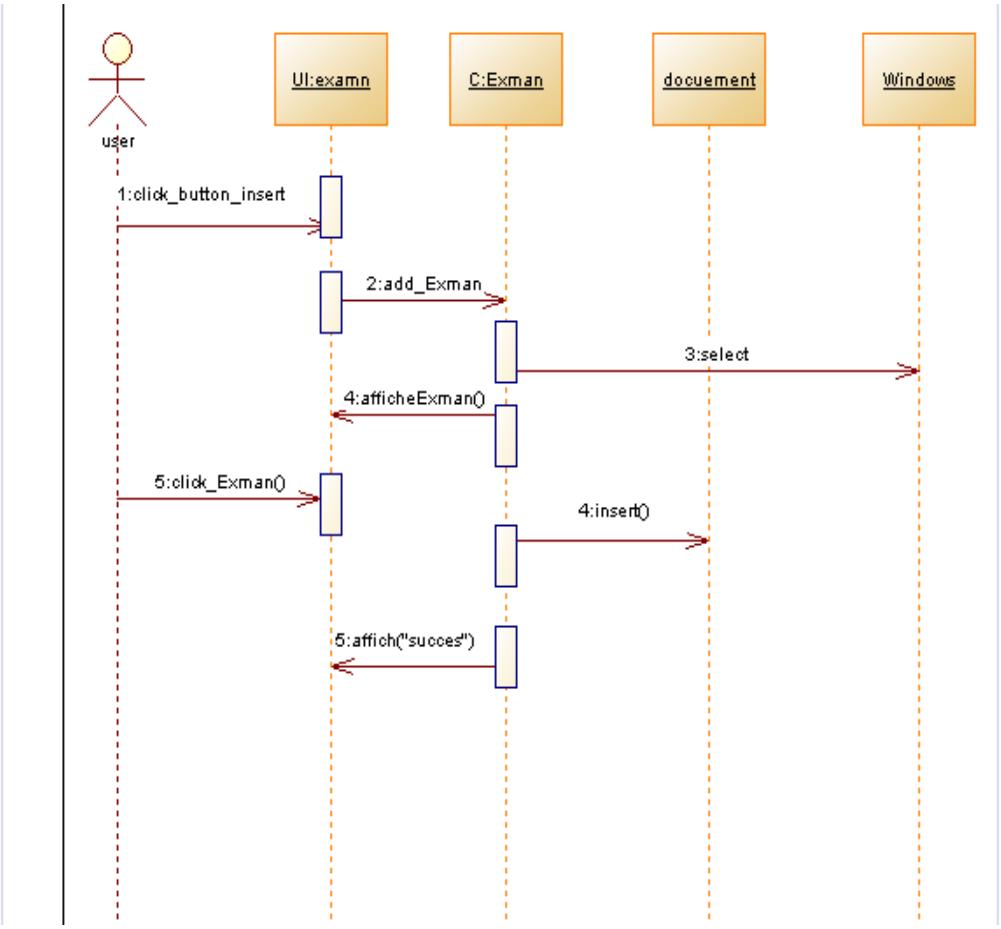
6.The sequence diagrams of “download exam”:



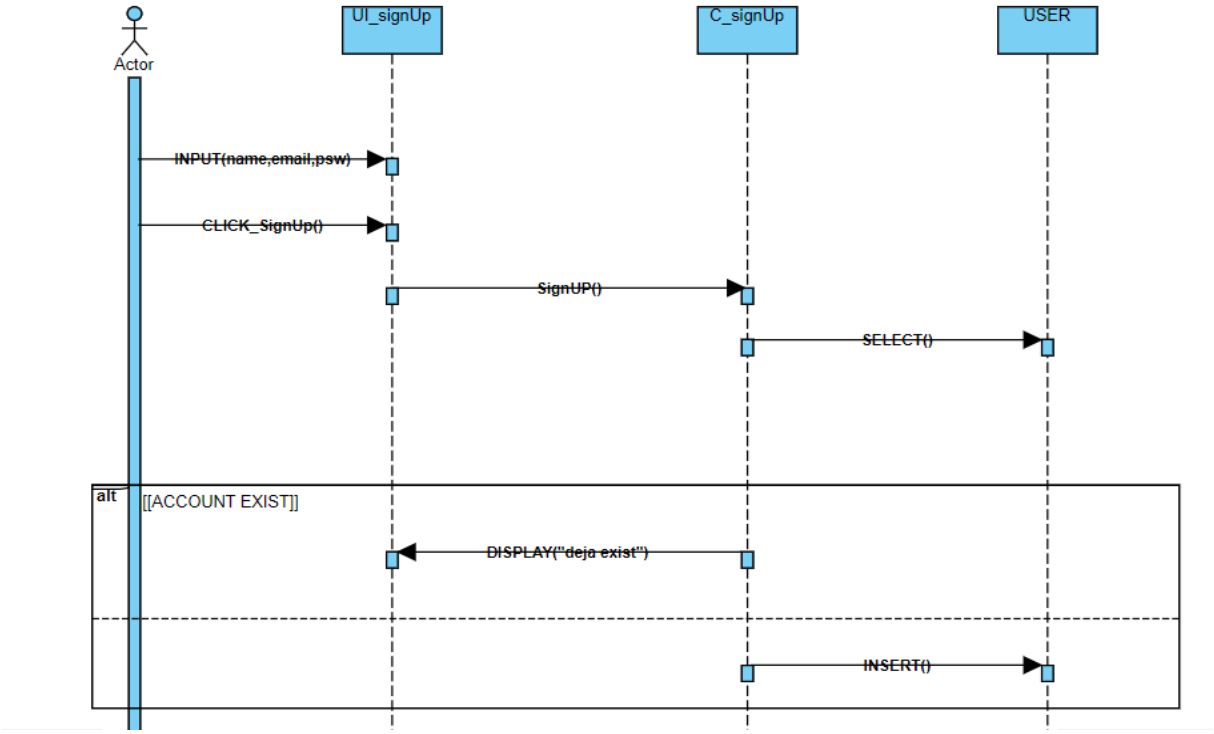
7.The sequence diagrams of “consult exam”:



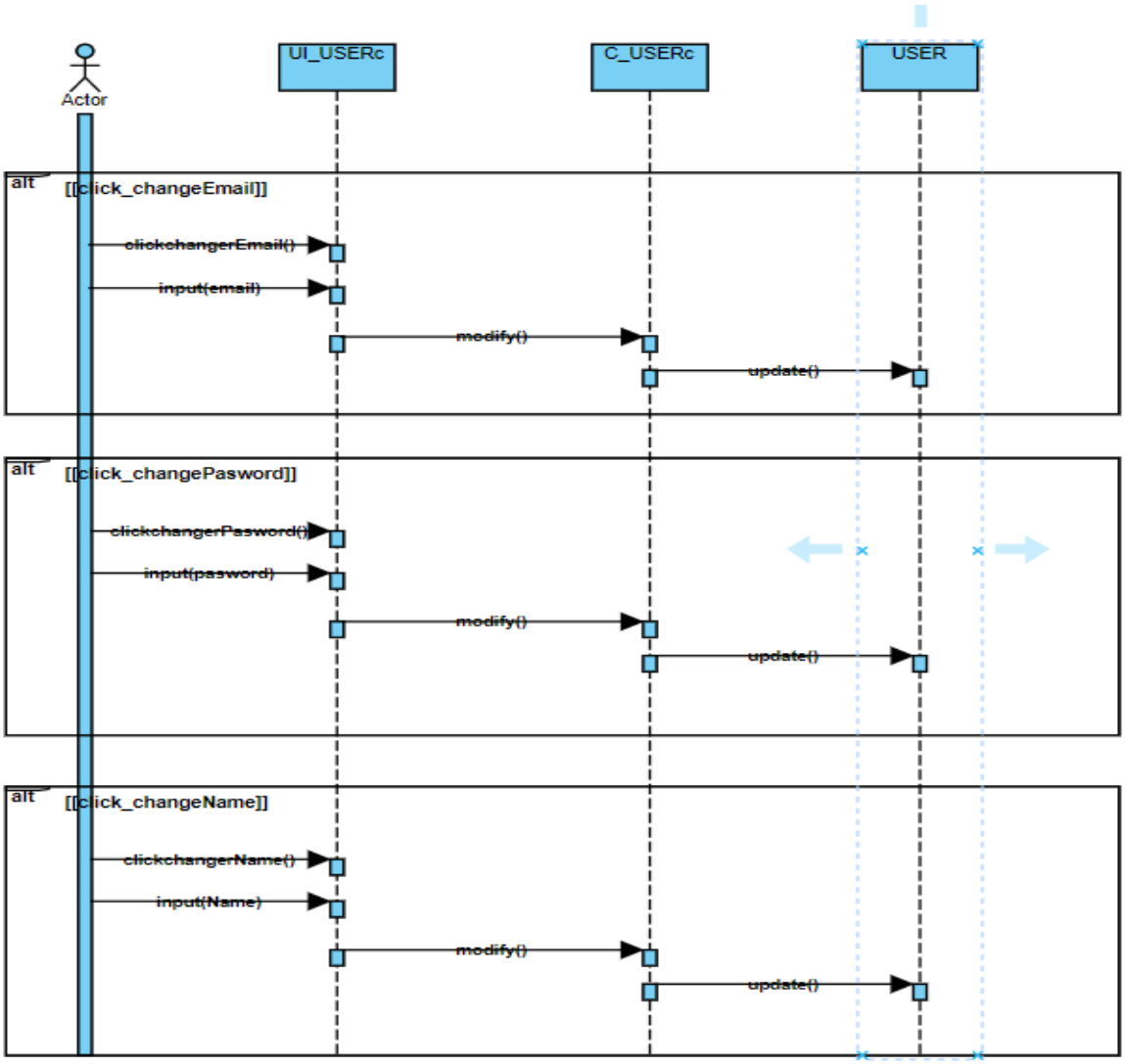
8.The class and sequence diagrams of “add exam”:



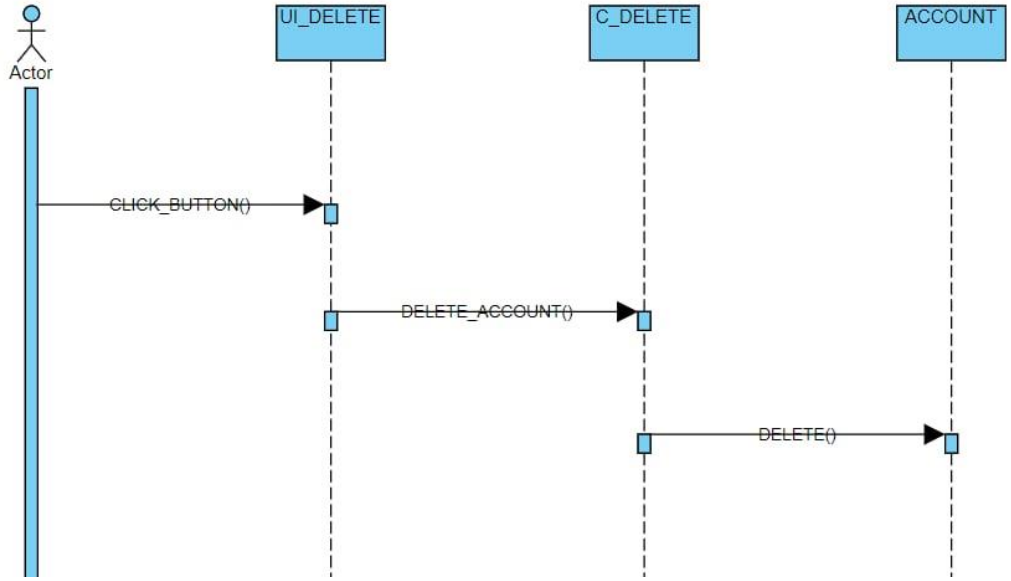
9. The sequence diagrams of “Inscription”:



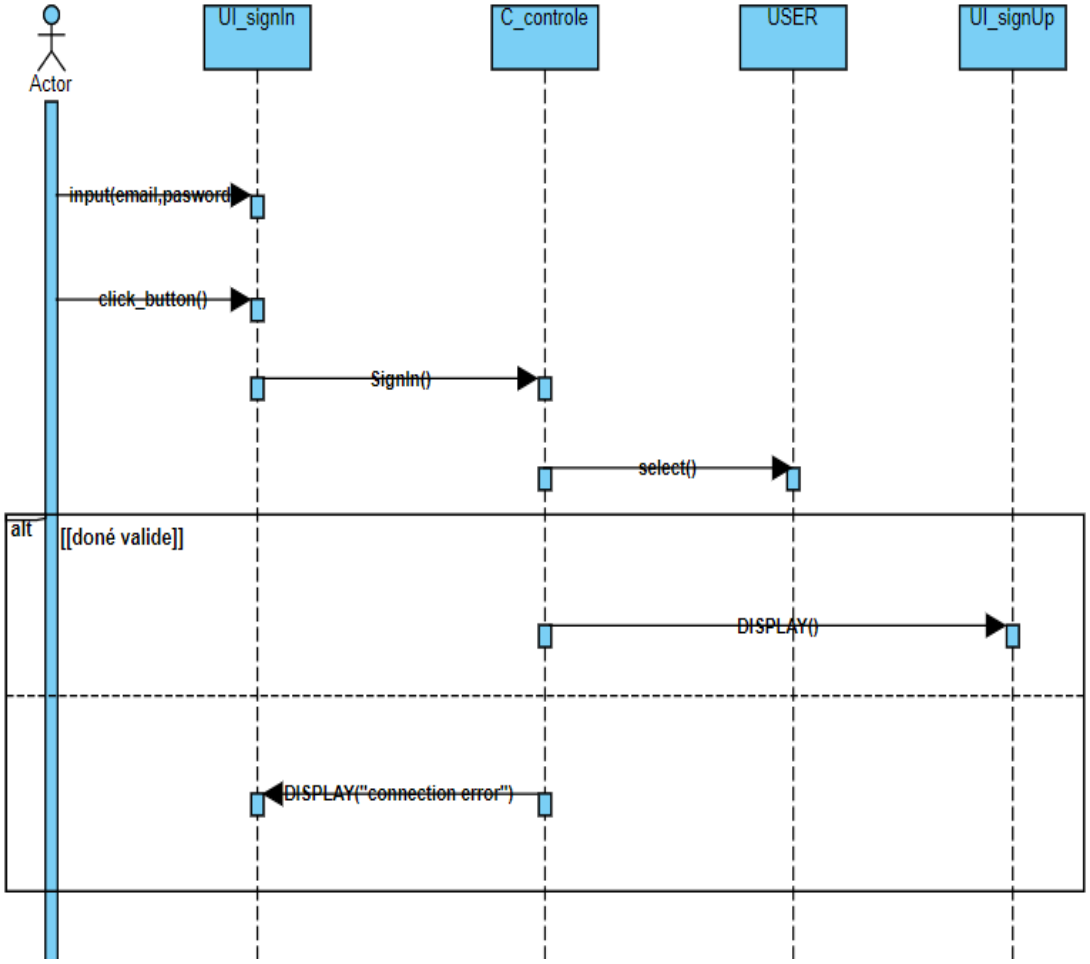
10. The sequence diagrams of “update account”:



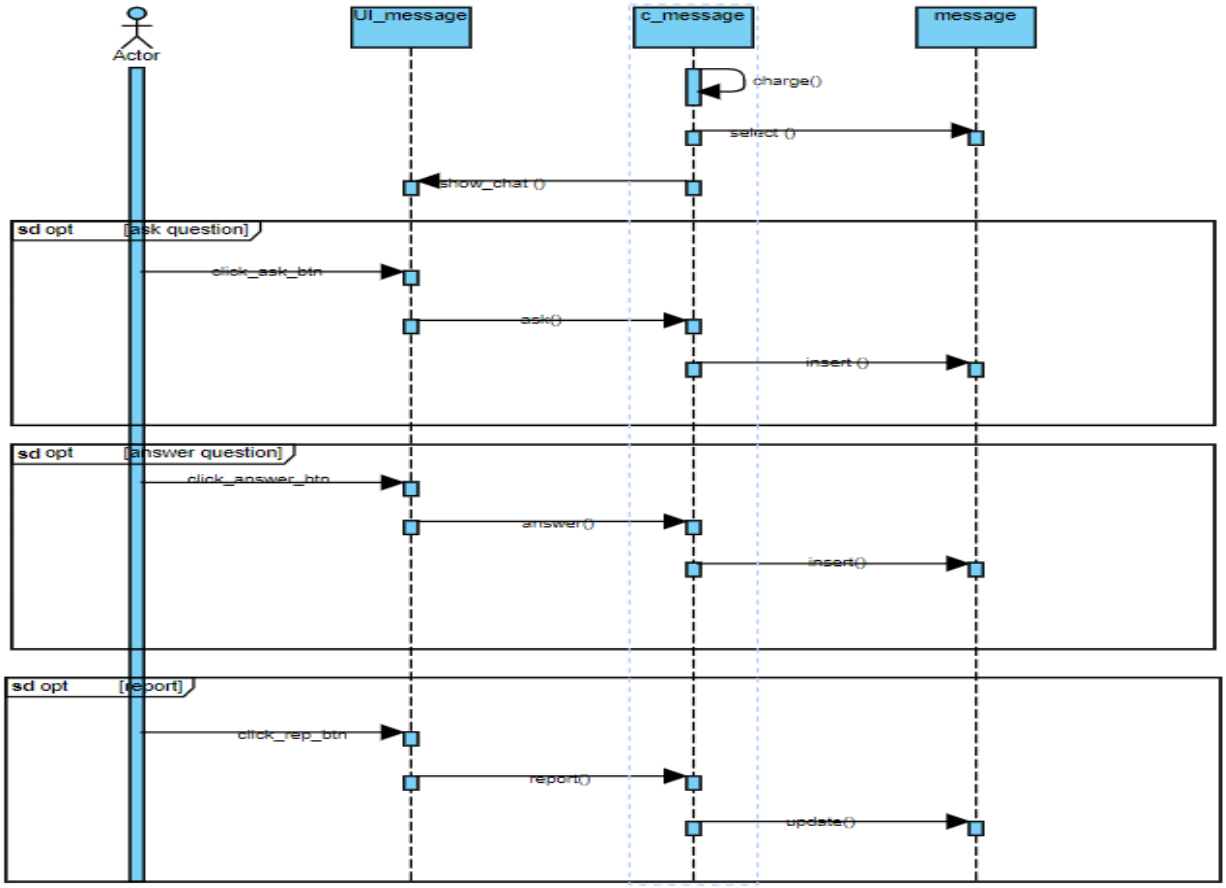
11. The sequence diagrams of “delete account”:



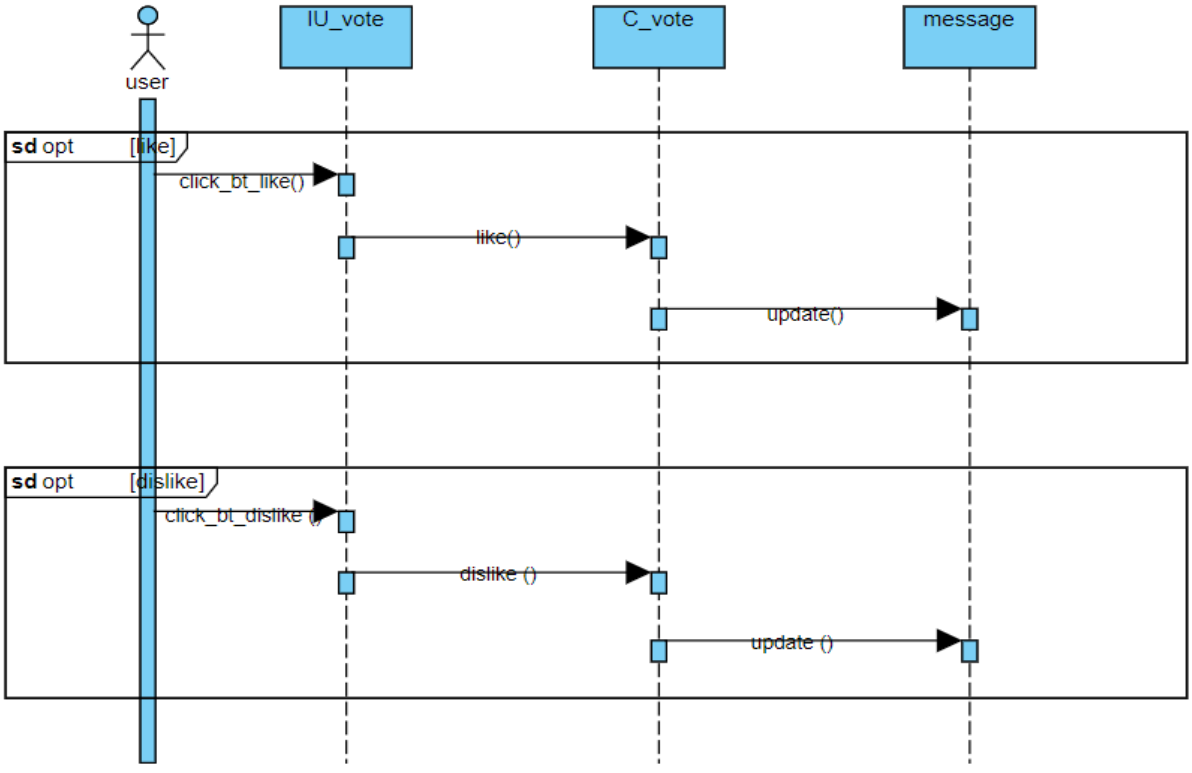
12.The sequence diagram of "authenticate":



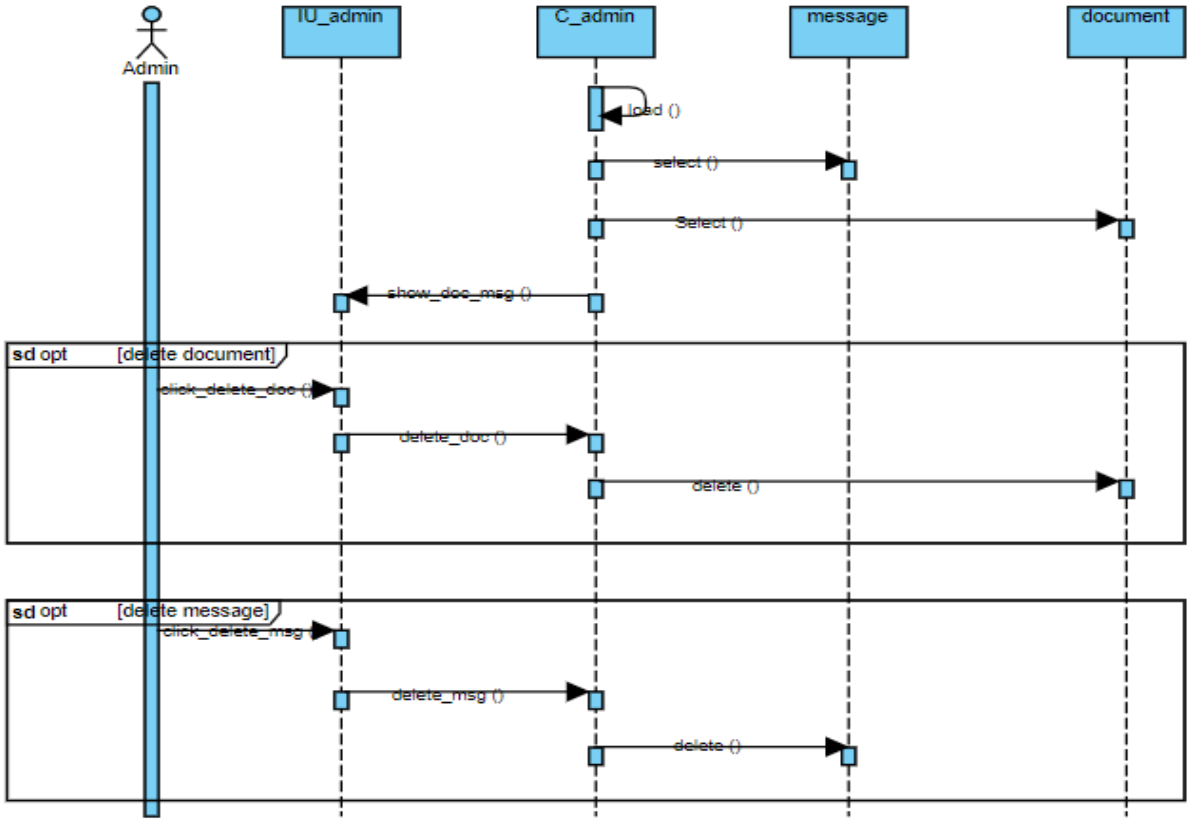
13The sequence diagrams of "discuss":



14.The sequence diagram of "vote":

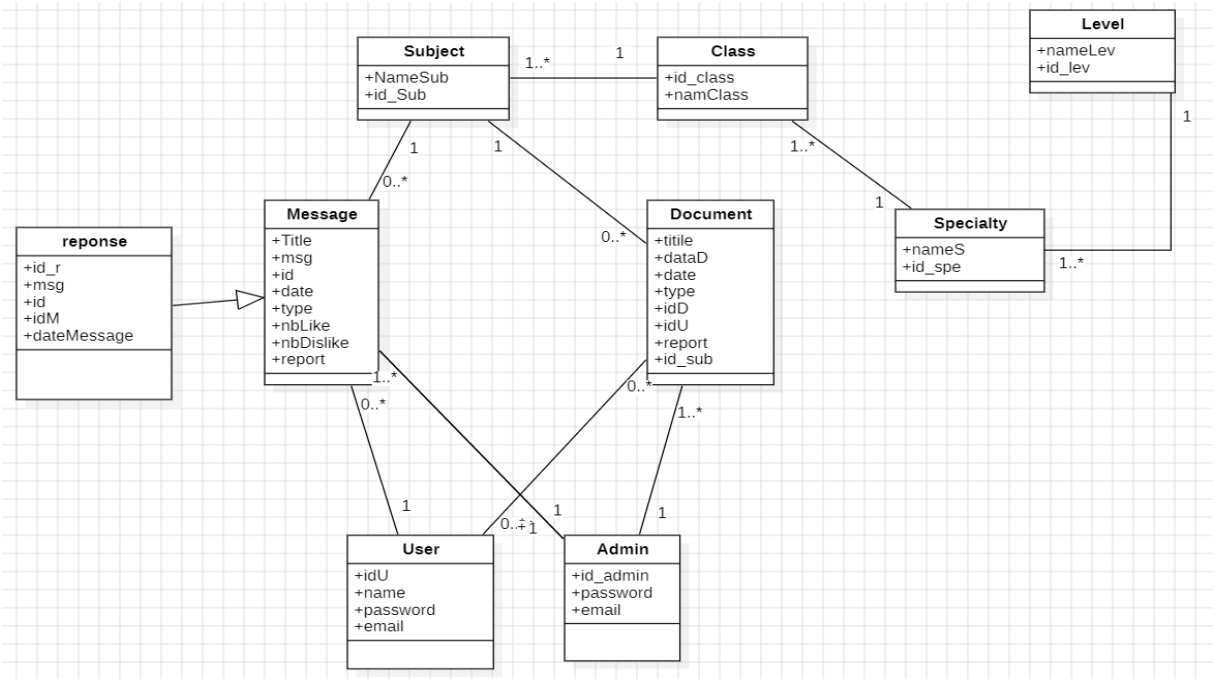


15.The sequence diagram of "admin verify:

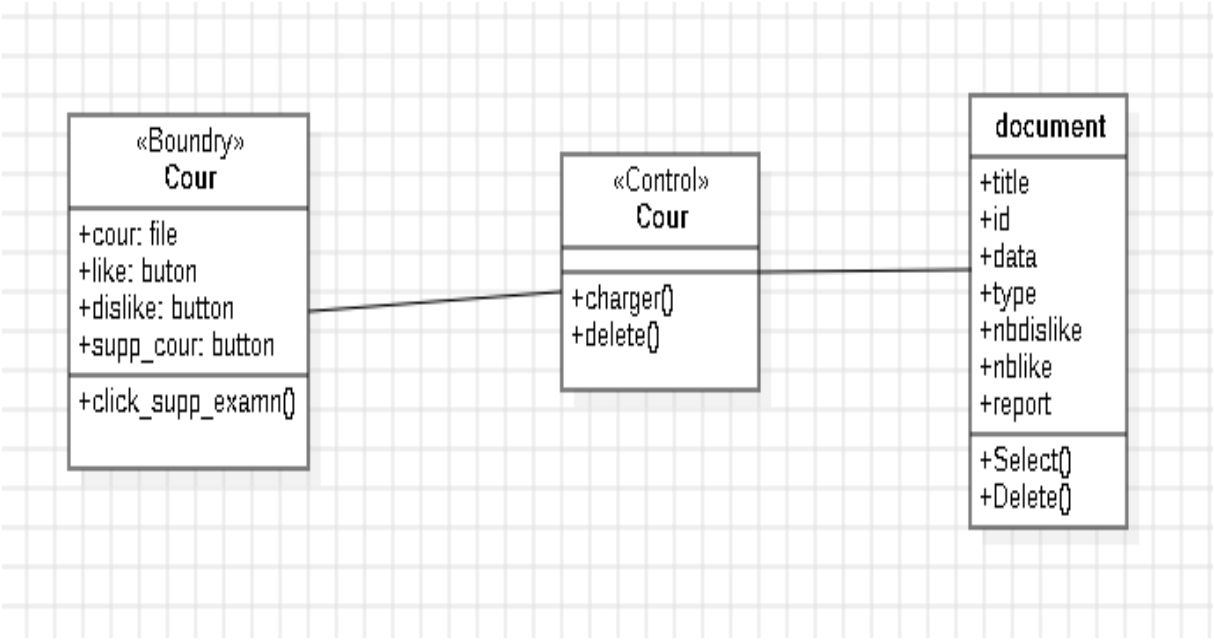


4.3. Classes diagrams:

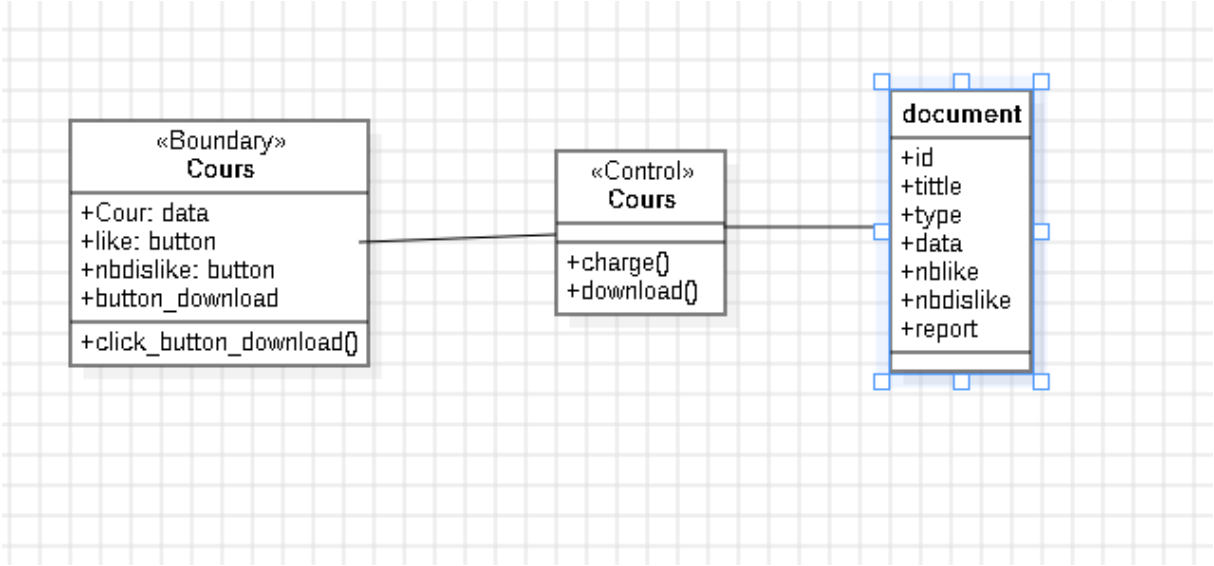
1.The global class diagrams:



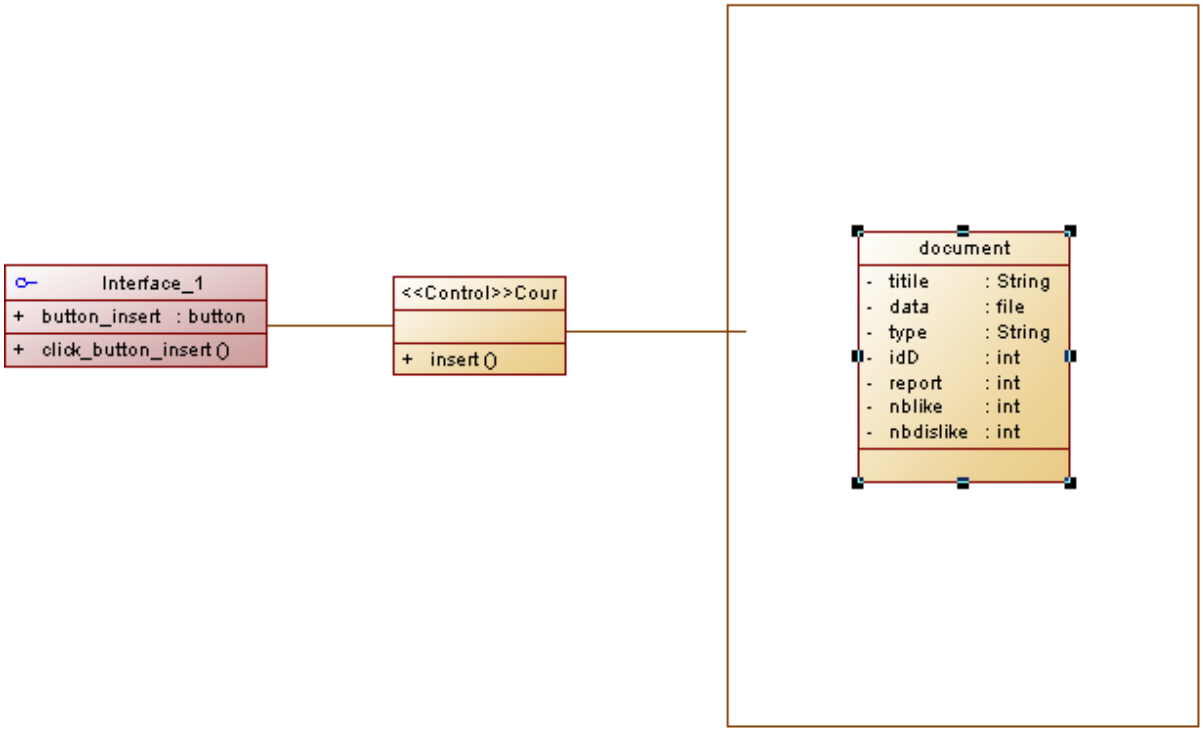
2.The class diagrams of “delete course”:



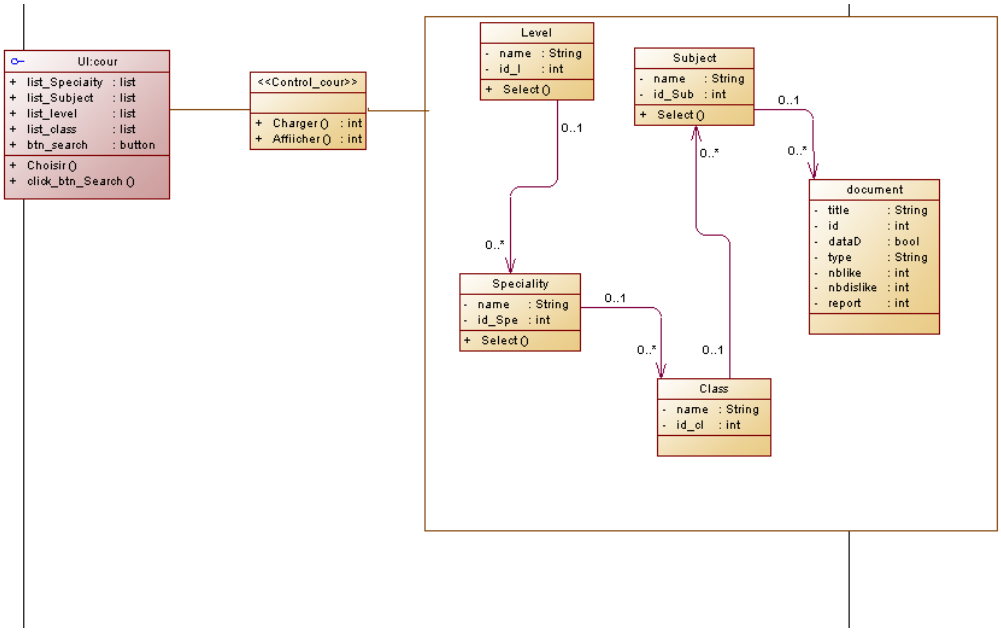
3. The class diagrams of “download course”:



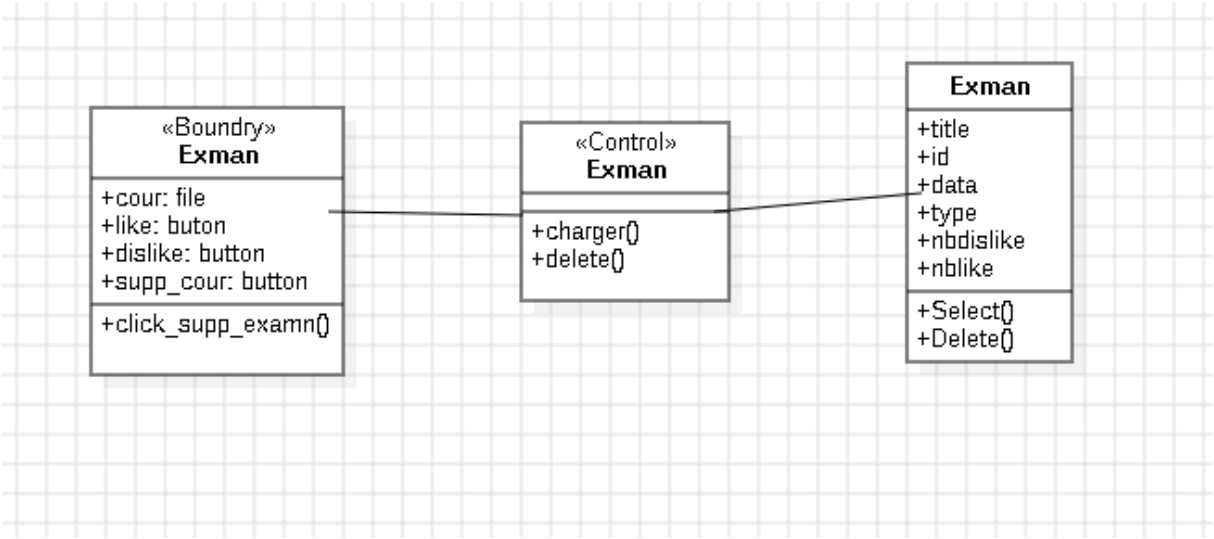
4. The class diagrams of “add course”:



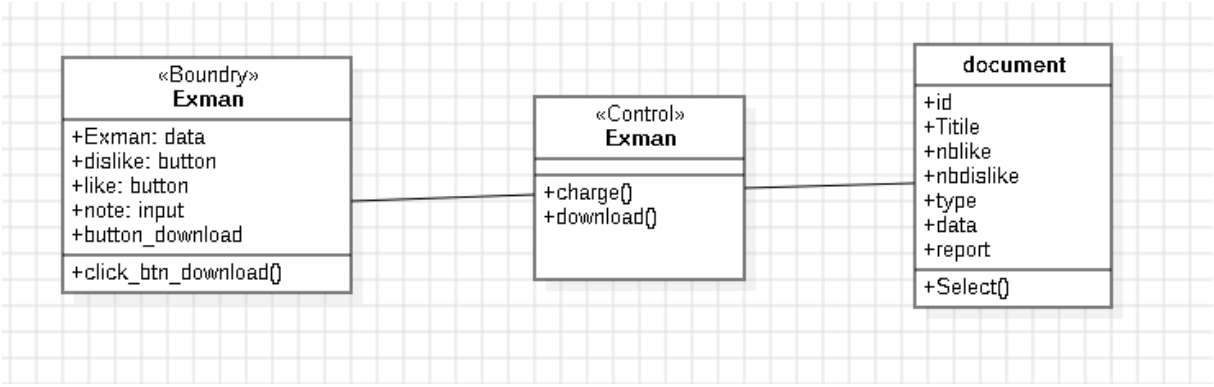
5.The class diagrams of “consult course”:



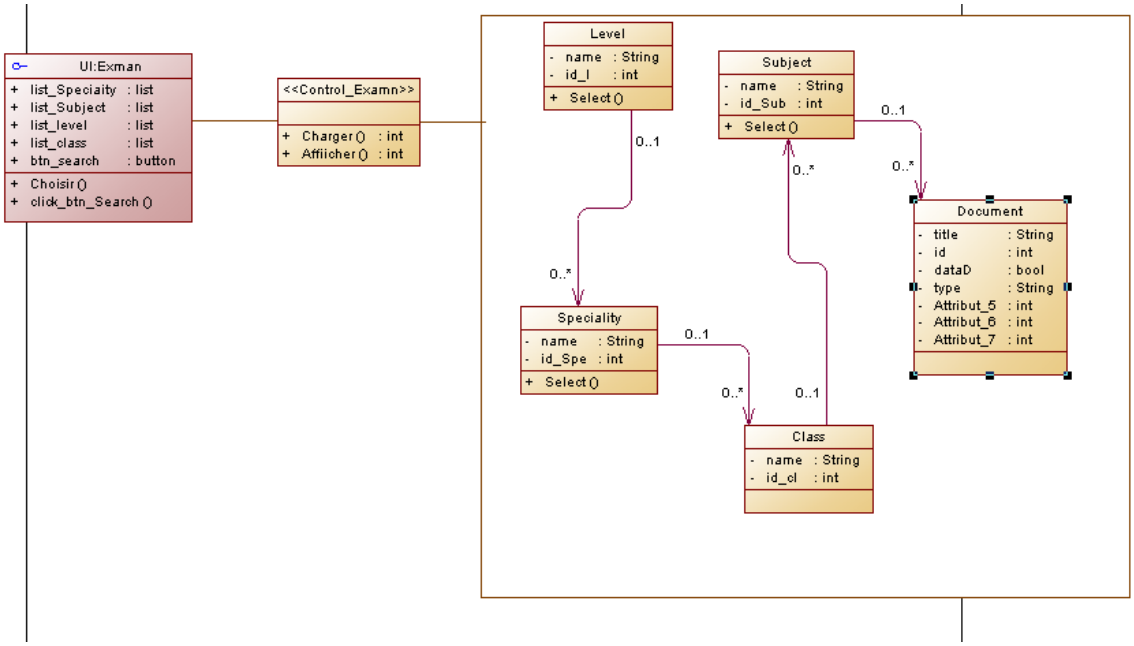
6. The class diagrams of of “delete exam”:



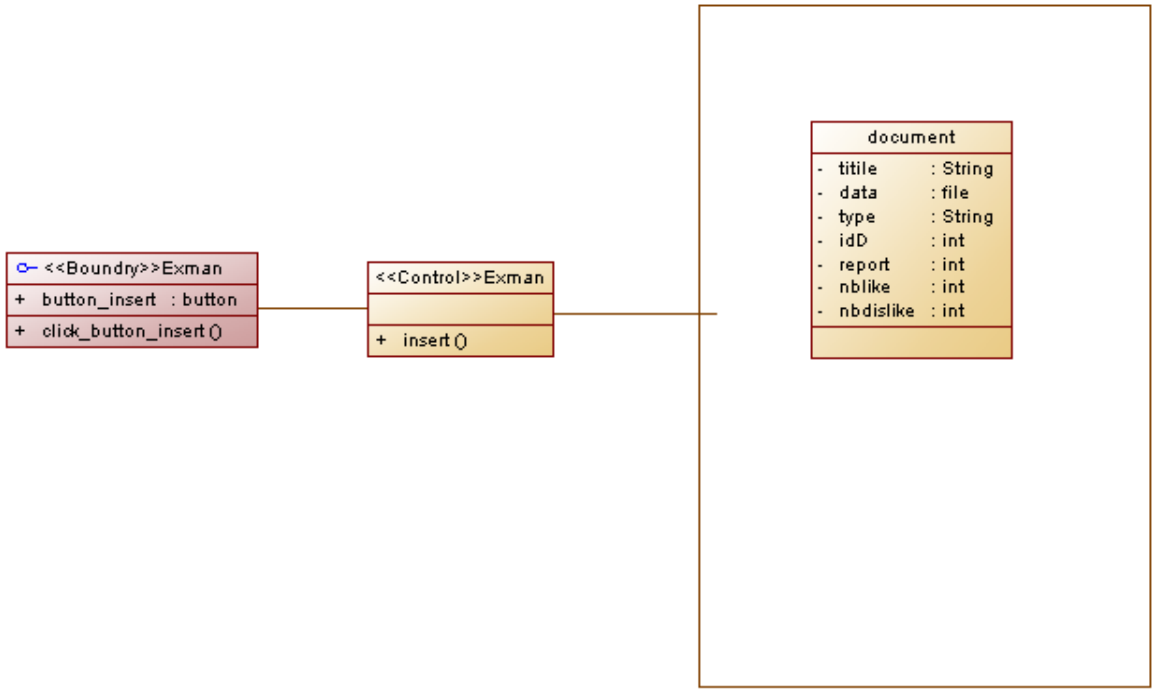
7. The class diagrams of “download exam”:



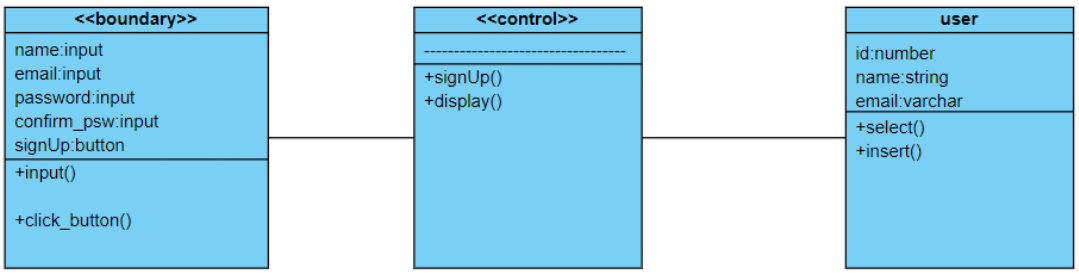
8. The class diagrams of “consult exam”:



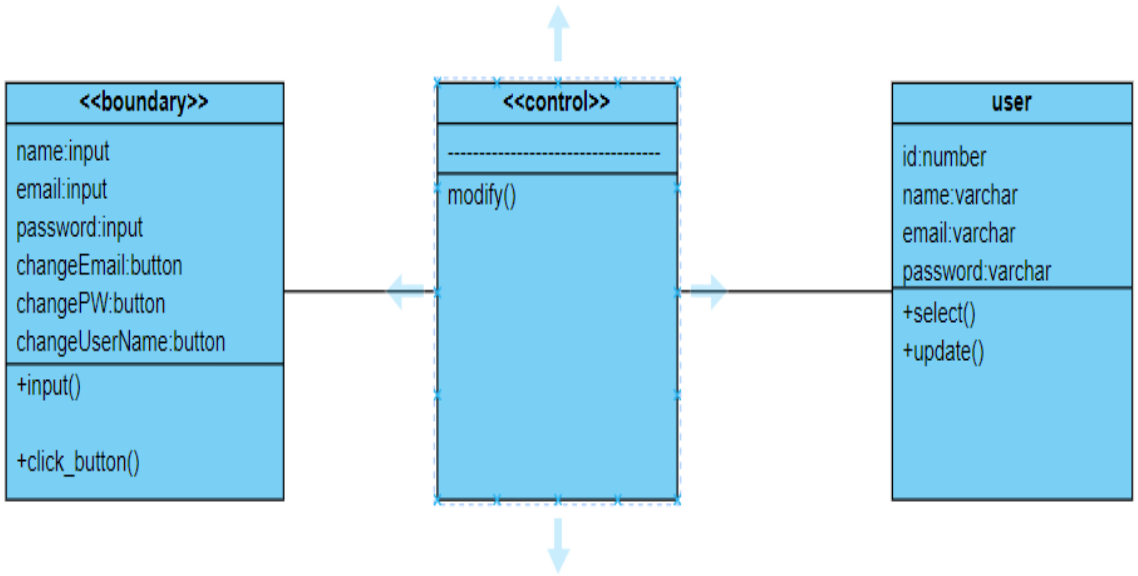
9.The class and sequence diagrams of “add exam”:



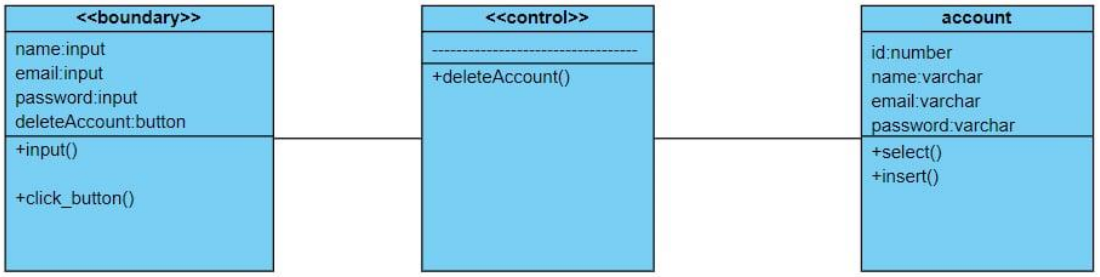
10. The class diagrams of “Inscription”:



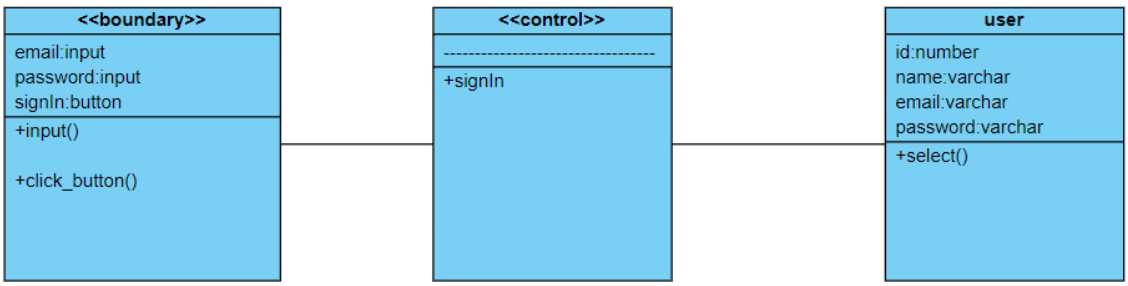
11. The class diagrams of “update account”:



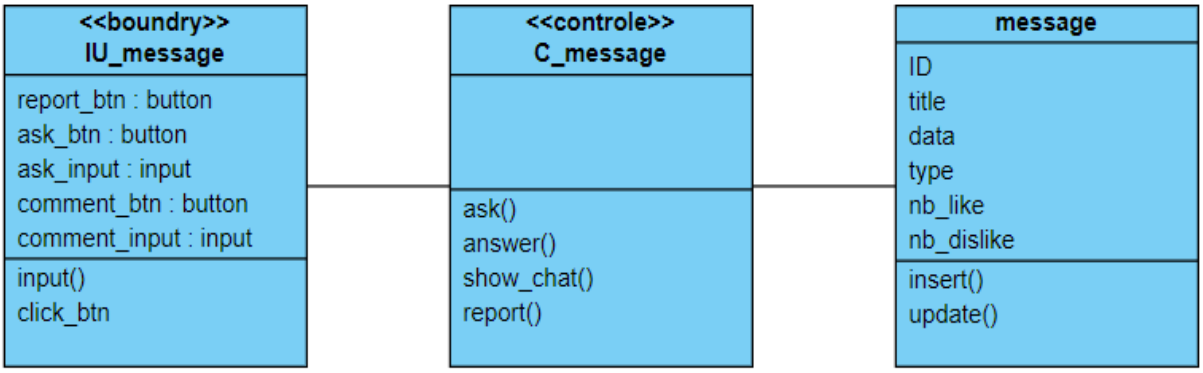
12. The class diagrams of “delete account”:



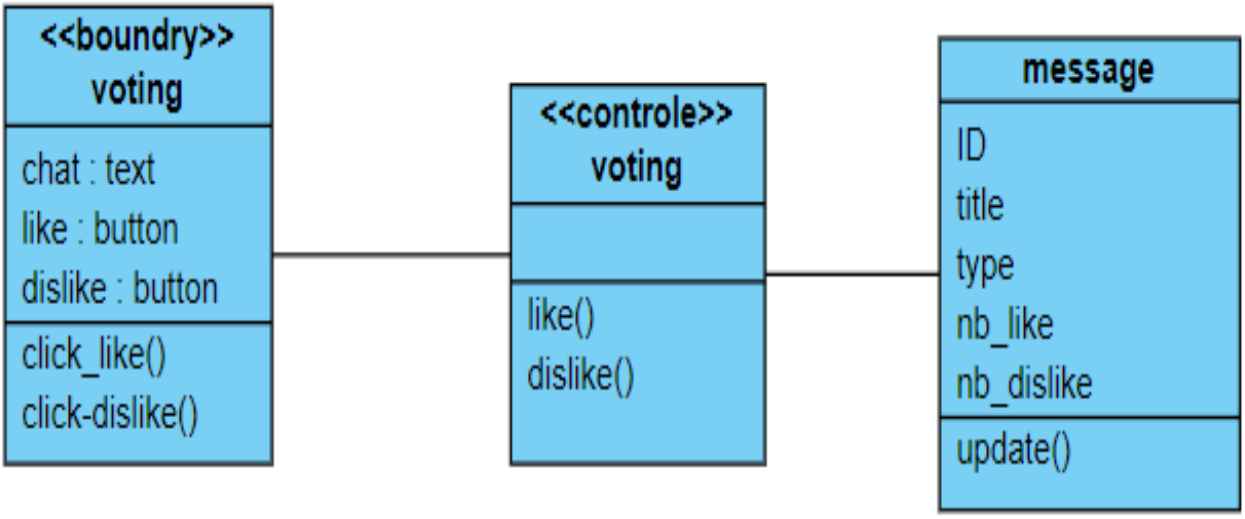
13. The class diagrams of "authenticate":



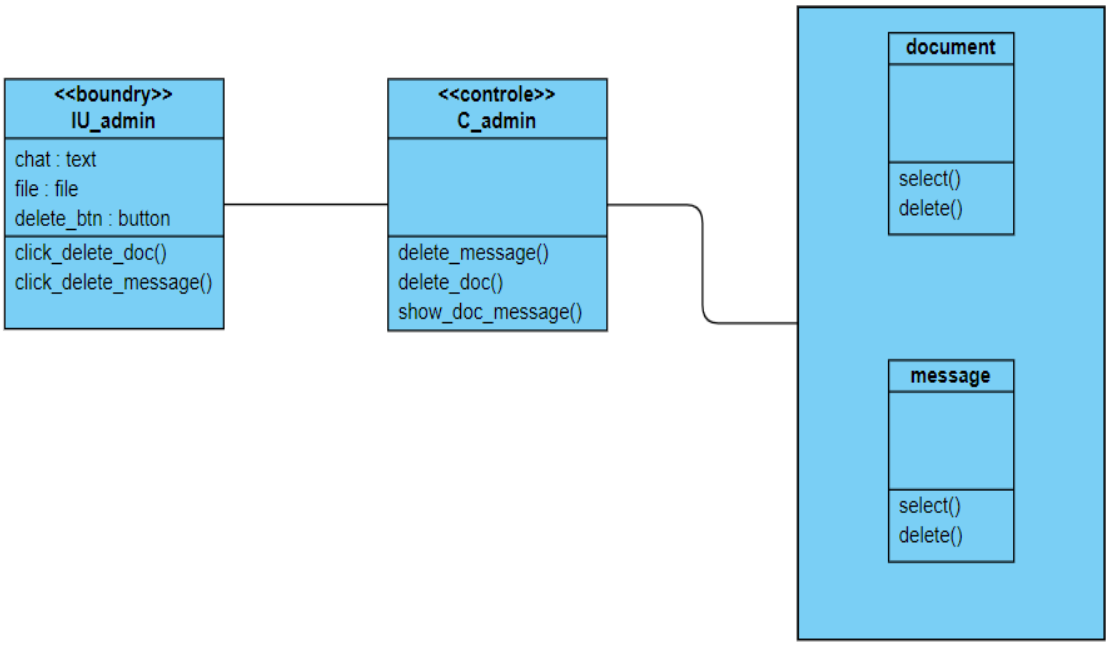
14. The class diagrams of "discuss"



15. The class diagrams of "vote".



16. The class diagrams of "admin verify"



5.Conclusion:

In this chapter, we introduced our project “LearnUp” alongside of covering the importance of requirements specification for our software under the name of "LearnUp". We discussed identifying functional and non-functional requirements including reliability, error Handling, user-Friendly Design, security and maintenance and reusability, determining actors, and creating user case diagram in addition of sequence diagram and classes diagram.