

**Tunisian Republic Ministry of Higher Education and Scientific Research**

**University of Carthage**

**Higher Institute of Information Technologies and Communication**

**Report**

**Of**

**Mini Project**

**Subject:**

**develop educational website**

**Prepared by:**

**Adem Abassi, Adem Bouadila &**

**Oussama Meddeb**

**College year 2023-2024**

**Summary**

[**General introduction** 4](#_Toc164729308)

[**Road map introduction:** 5](#_Toc164729309)

[***Chapter 1: Requirements specification*** 5](#_Toc164729310)

[**1.Introduction:** 5](#_Toc164729311)

[**2.Identification of functional requirement:** 5](#_Toc164729312)

[**3.Identification of non-functional requirement:** 5](#_Toc164729313)

[**4.Actors Identification:** 6](#_Toc164729314)

[**5.Global use case diagram:** 7](#_Toc164729315)

[**6.Product backlog** 7](#_Toc164729316)

[**7.Work environment** 8](#_Toc164729317)

[**7.1. Project management methodology:** 8](#_Toc164729318)

[**7.1. Technological choices:** 9](#_Toc164729319)

[**8.conclusion:** 12](#_Toc164729320)

[***Chapter 2: Sprint 0*** 12](#_Toc164729321)

[**1.Introduction:** 12](#_Toc164729322)

[**2.Identification of sprint0 backlog:** 12](#_Toc164729324)

[**3.Refinement of sprint0:** 12](#_Toc164729325)

[**3.1. Refinement the administrator story “document management”:** 13](#_Toc164729326)

[**3.2. Refinement the administrator story “manage account”:** 14](#_Toc164729327)

[**3.2. Refinement the administrator story “authentification”:** 15](#_Toc164729328)

[**4.Design of sprint 0:** 15](#_Toc164729329)

[**Class diagram:** 15](#_Toc164729330)

[**Sequance diagram:** 16](#_Toc164729331)

[**4.1. Manage documents:** 16](#_Toc164729332)

[**4.2. Manage account:** 23](#_Toc164729333)

[**5. implementation of spring 0:** 27](#_Toc164729334)

[**6.Conclusion:** 31](#_Toc164729335)

[***Chapter 3: Sprint 1*** 32](#_Toc164729336)

[**1.Introduction:** 32](#_Toc164729337)

[**2.Identification of sprint 1 backlog:** 32](#_Toc164729338)

[**3.Refinement of sprint 1:** 32](#_Toc164729339)

[**3.1. Refinement the administrator story “discuss”:** 33](#_Toc164729340)

[**3.1. Refinement the administrator story “voting”:** 34](#_Toc164729341)

[**3.1. Refinement the administrator story “admin verify”:** 34](#_Toc164729342)

[**4.Design of sprint 1:** 35](#_Toc164729343)

[**4.1. Discuss:** 35](#_Toc164729344)

[**4.2. vote:** 36](#_Toc164729345)

[**4.3. verify documents:** 37](#_Toc164729346)

[**5. implementation of sprint 1:** 38](#_Toc164729347)

[**6.Conclusion:** 40](#_Toc164729348)

[**General conclusion** 40](#_Toc164729349)

# **General introduction**

Understanding the common challenges faced by students, such as time constraints and difficulties grasping certain subjects, we acknowledge the hurdles many encounters on their academic journeys. Navigating through tight schedules or wrestling with complex concepts can indeed pose obstacles to effective learning. In response to these challenges, we proudly introduce our purpose-driven educational platform.

Our app is designed to be a supportive companion for those who find themselves pressed for time or in need of additional clarity, especially as exams draw near. With its user-friendly interface, the app offers a seamless experience, allowing users to effortlessly 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.

# **Road map introduction:**

for the development stage of our project, we adopted Scrum as a working plan and by that we splitted the project into the following three chapters:

• The first chapter under the name of” Requirements specification” showcases the requirement specification.

• The second chapter under the name of “sprint 0”.

• The third chapter under the name of “sprint 1”:

# ***Chapter 1: Requirements specification***

## **1.Introduction:**

Studying the project is interesting because it helps describe the company's environment and state our project's main goals. First, we'll talk about the host organization. Then, we'll explain the project, its challenges, and the proposed solution. Finally, we'll explain how we'll work to complete our project.

## **2.Identification of functional requirement:**

Our project aims to create a studying website that meets the following needs:

+Register

+Manage documents

+voting

+authenticate

+community chat

## **3.Identification of non-functional requirement:**

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.

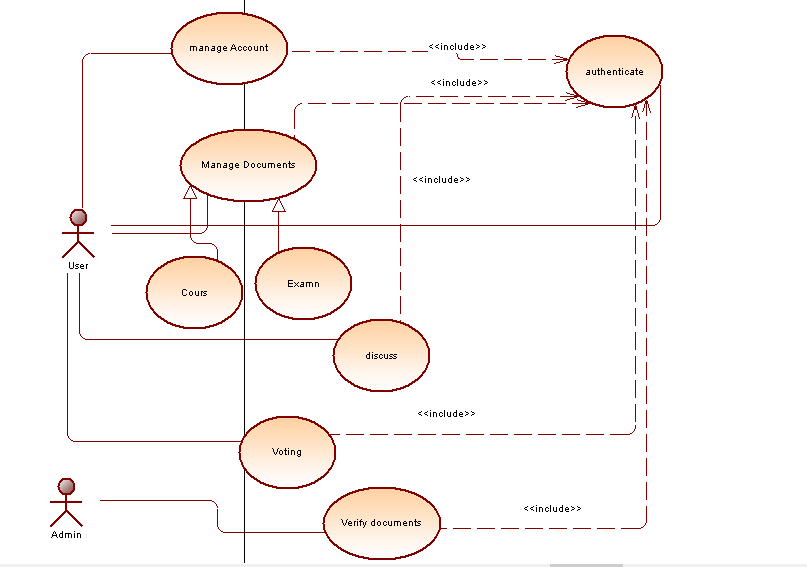
## **4.Actors Identification:**

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 their courses and exams.

## **5.Global use case diagram:**



## **6.Product backlog:**

|  |  |  |  |
| --- | --- | --- | --- |
| Backlog of product | Priority | Estimation | Planning |
| As a user I can register. | 1 | High | Sprint 0 |
| As a user, I can authenticate myself. | 1 | High | Sprint 0 |
| As a user I can manage documents. | 1 | High | Sprint 0 |
| As a user I can discuss. | 2 | Average | Sprint 1 |
| As a user I can voting | 2 | Average | Sprint 1 |
| As an admin I can verify documents | 2 | Average | Sprint 1 |

## **7.Work environment**

## **7.1. Project management methodology:**

For this project, we have chosen to work with the Agile methodology:

**Scrum Work Methodology:**

Scrum is a methodology used for developing and maintaining complex products. It helps teams tackle intricate and evolving issues, delivering products of the highest possible value in a creative and productive manner. Scrum employs an incremental approach to enhance predictability and manage risk.

**Sprint :**

Sprint planning involves identifying the priority points that the team believes can be accomplished during the sprint.

The sprint review takes place at the end of each sprint, where the development team presents the completed features.

The produced increment is potentially deliverable, and the setup for the next sprint can be anticipated.

## **7.1. Technological choices:**

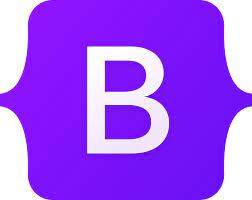
**Angular:**

Angular is a framework created by Google for building dynamic web applications. It is based on TypeScript, which is a superset of JavaScript. Angular offers features like two-way data binding, dependency injection, and a component-based architecture, making it easier to develop and maintain complex web applications.



**Bootstrap:**

Bootstrap is a popular open-source front-end framework that facilitates the development of responsive and mobile-first websites. It includes a collection of pre-designed HTML, CSS, and JavaScript components, making it easier for developers to create a visually appealing and consistent user interface



**php :**

PHP is a widely used open-source server-side scripting language that is especially suited for web development and can be embedded into HTML. It is commonly used to create dynamic web pages or applications. PHP code is executed on the server, generating HTML which is then sent to the client's web browser.PHP can also be used for command-line scripting and writing desktop applications.



**MySQL :**

MySQL is an open-source relational database management system (RDBMS) that allows users to interact with databases either directly by using SQL, or more often with other programs to implement applications that require relational database capabilities.



**VisualStudio :**

Visual Studio is an integrated development environment (IDE) created by Microsoft.It is used to develop computer programs, websites, web apps, and mobile apps. Visual Studio provides a suite of tools for code editing, debugging, testing, and deployment, all integrated into a single development environment. It supports various programming languages, including C#, C++, Visual Basic, JavaScript, and Python, among others. Visual Studio is widely used by developers and organizations for building a wide range of software applications.



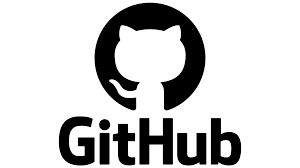
**xampp :**

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.



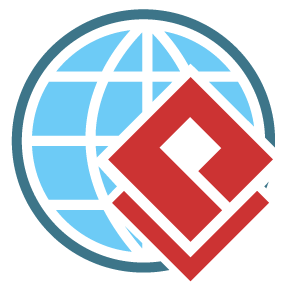
**Github :**

GitHub is a web-based platform used for version control and collaboration on software development projects. It allows developers to host and share code, track changes to code files, and collaborate with others on coding projects. GitHub uses Git, a distributed version control system, to manage and track changes to files. It provides features such as issue tracking, pull requests, and project management tools, making it a popular choice for managing software development projects, both for individuals and teams.



**Visual paradigm:**

Visual Paradigm is a software development platform used for modeling and designing software applications. It supports various types of diagrams like UML, BPMN, and ERD, helping users visualize and communicate complex software designs.



## **8.conclusion:**

In this chapter, we covered the importance of requirements specification for our software under the name of “LearnUp”. We discussed identifying functional and non-functional requirements, determining actors, and creating a global use case diagram. We also touched on the product backlog, which lists all necessary features, and the work environment, including project management methodology and technological choices.

# ***Chapter 2: Sprint 0***

## **1.Introduction:**

## In this chapter, we'll discuss the first version of our project “LearnUp”. We'll outline the most important use cases and explain how we refined, designed, and implemented them in this sprint.

## **2.Identification of sprint0 backlog:**

Below is the list of tasks that need to be completed during sprint 0 for the first release. This backlog outlines the specific items that must be addressed to ensure a successful start to the project:

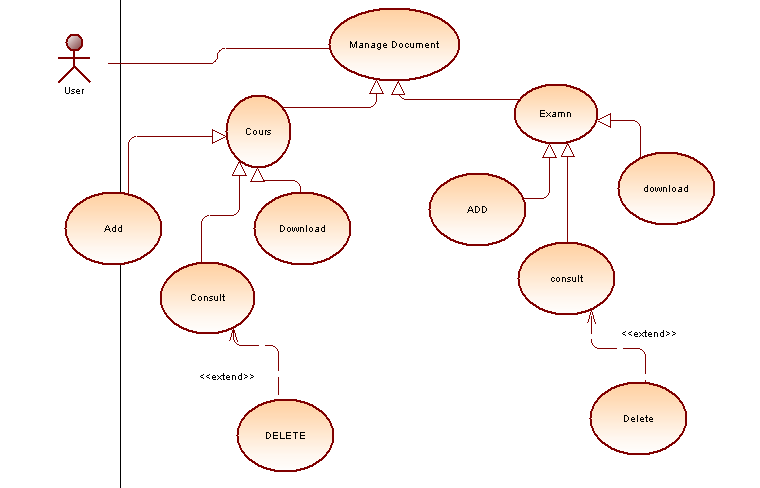
|  |  |  |  |
| --- | --- | --- | --- |
| **Backlog of the product** | **priorities** | **estimate** | **planning** |
| As a user, I can manage documents | 1 | high | Sprint 0 |
| As a user, I can manage my account | 1 | high | Sprint 0 |
| As a user, I can be authenticated | 1 | high | Sprint 0 |

## **3.Refinement of sprint0:**

In this part, we examine the various scenarios for using the first sprint.

## **3.1. Refinement the administrator story “document management”:**

The following figure showcases the use case.

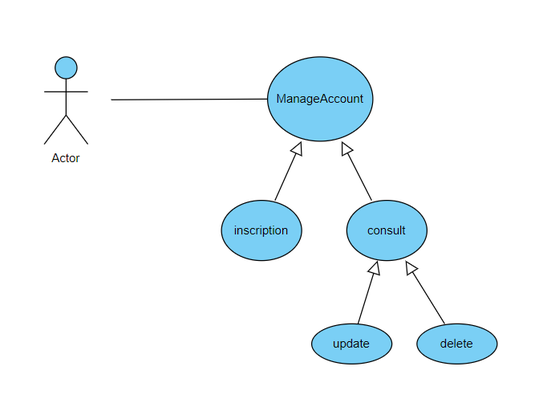


The following table elaborates on this user story with textual description.

|  |  |
| --- | --- |
|  | 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 |

## **3.2. Refinement the administrator story “manage account”:**

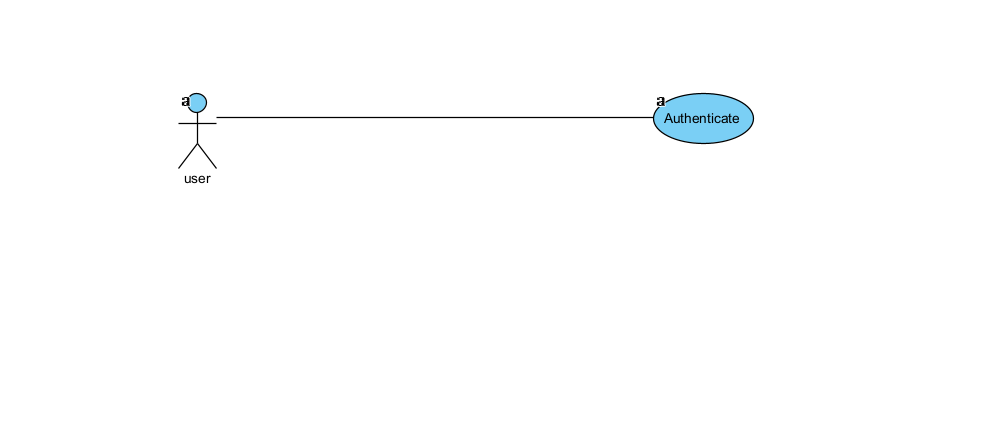
The following figure showcases the use case:

****

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 insciripted |
| **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 |

## **3.2. Refinement the administrator story “authentification”:**



|  |  |
| --- | --- |
| **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. |

## **4.Design of sprint 0:**

We will showcase the class and sequence diagrams for the various use-case scenarios we implemented in the final section of their refinements.

## **Class diagram:**

A class diagram is a visual representation used in software engineering to show the structure and relationships of classes in a system. It helps in understanding how classes are related, including their attributes and methods. This diagram is part of the Unified Modelling Language (UML) and is commonly used to design and analyse object-oriented systems.

## **Sequance diagram:**

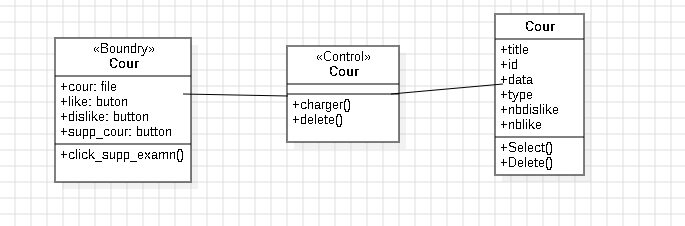
A sequence diagram in the Unified Modelling Language (UML) depicts how objects or components interact by showing the sequence of messages they exchange over time. It visually represents interactions between actors, objects, or components in a specific scenario or use case. These diagrams are useful for understanding how control and communication flow within a system and are frequently employed in software development to model system behaviour during runtime.

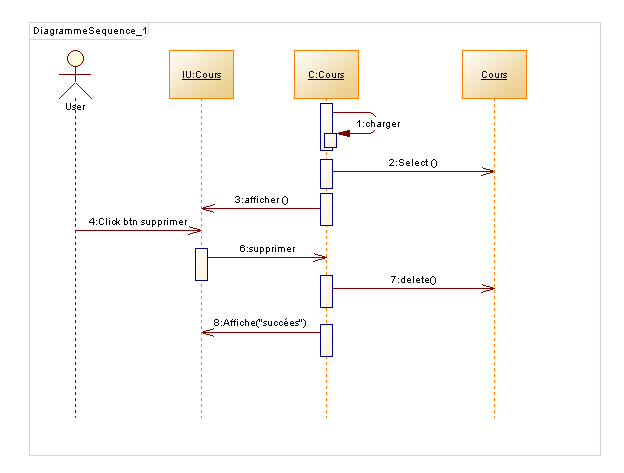
## **4.1. Manage documents:**

The next diagrams below represent the class and sequence diagrams related to the "manage documents" use case.

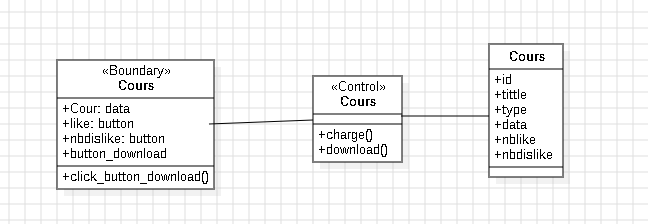
1.course:

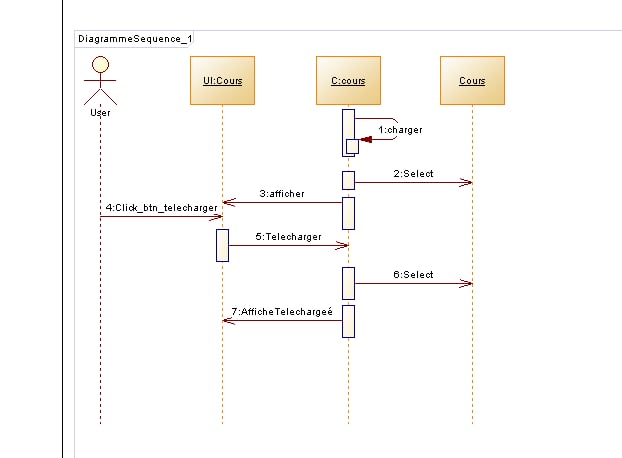
The class and sequence diagrams of “delete course”:



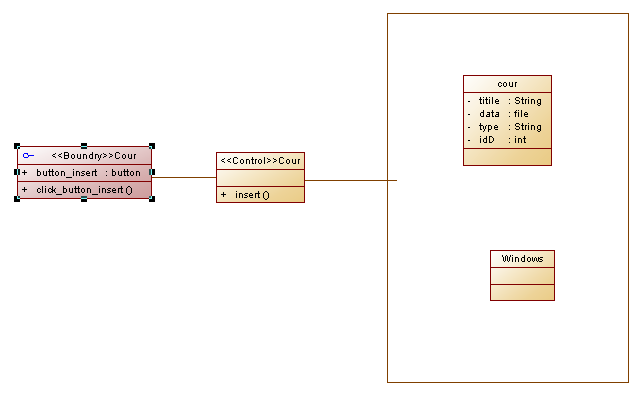


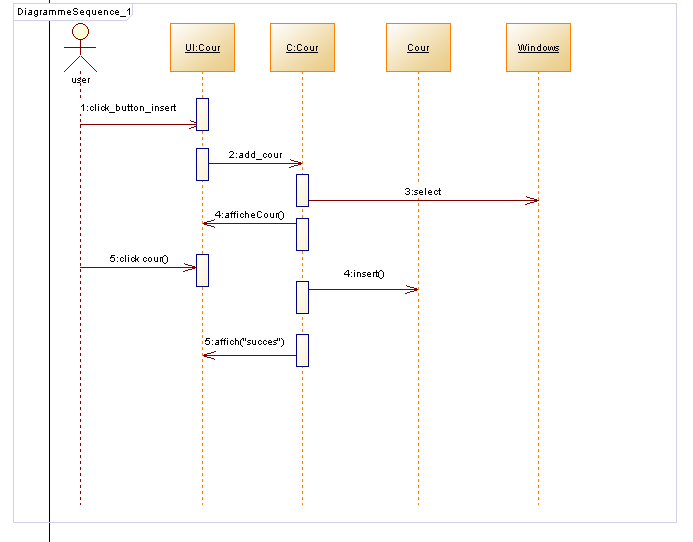
The class and sequence diagrams of “download course”:



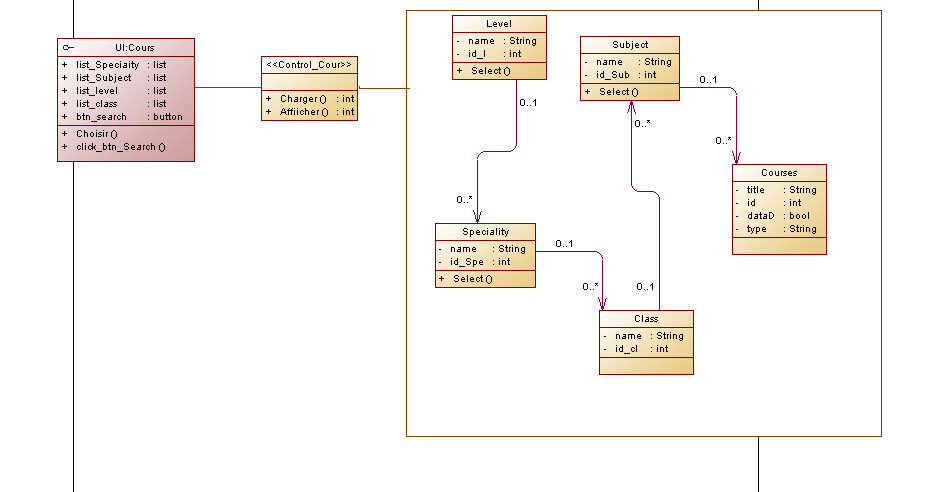


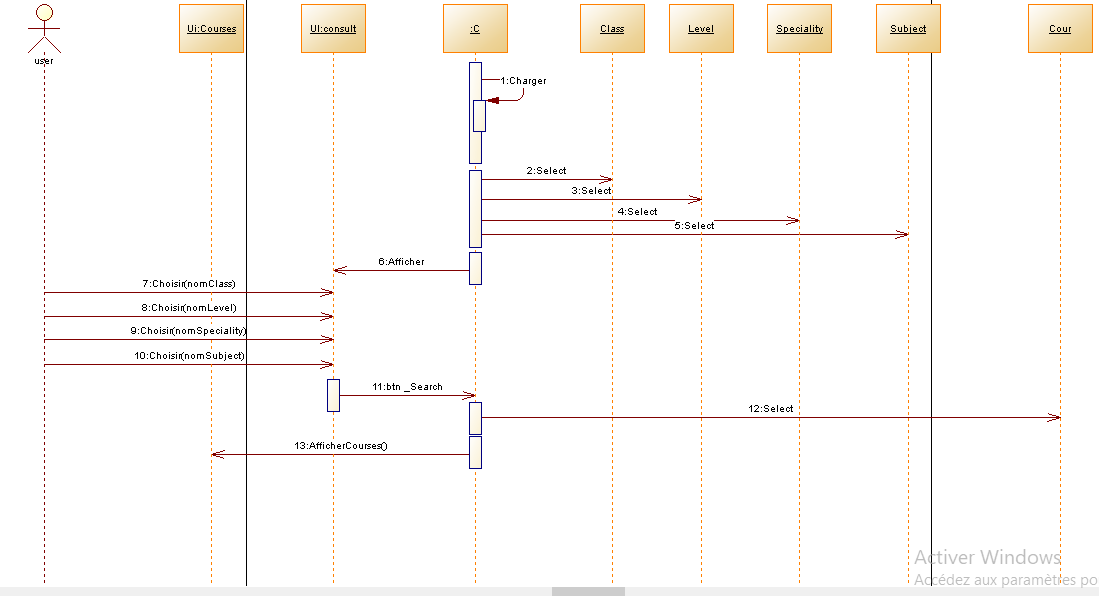
The class and sequence diagrams of “add course”:





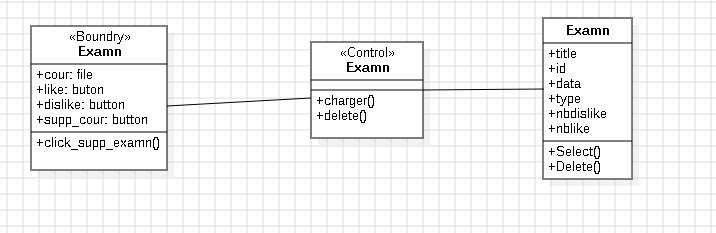
The class and sequence diagrams of “consult course”:

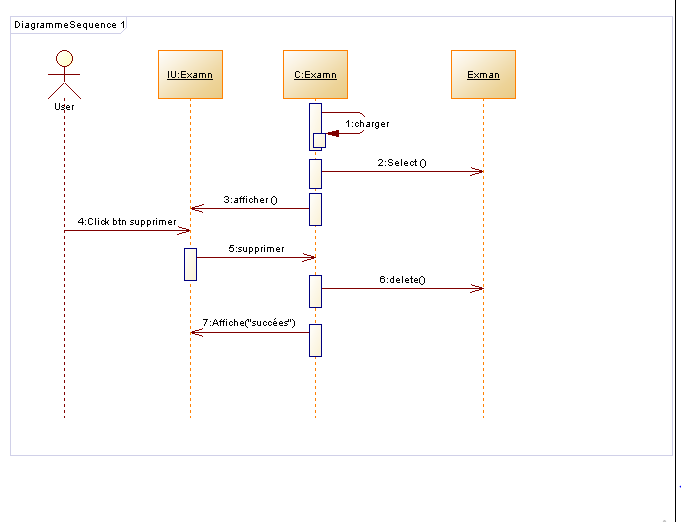




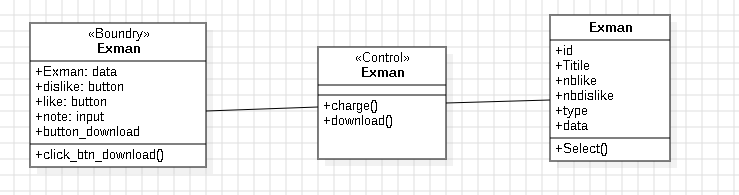
2.Exam:

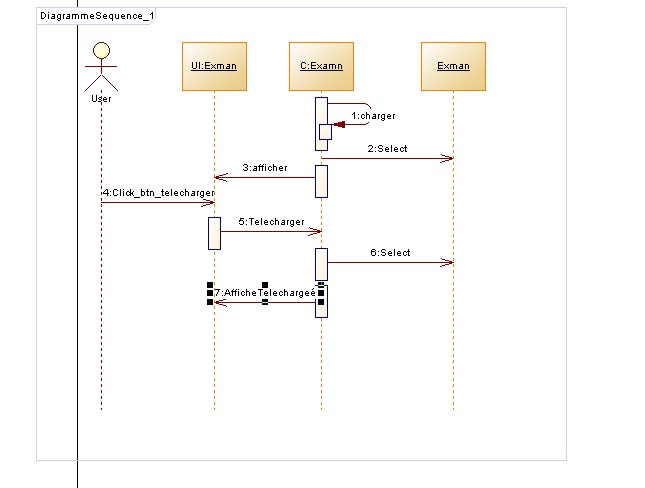
The class and sequence diagrams of “delete exam”:



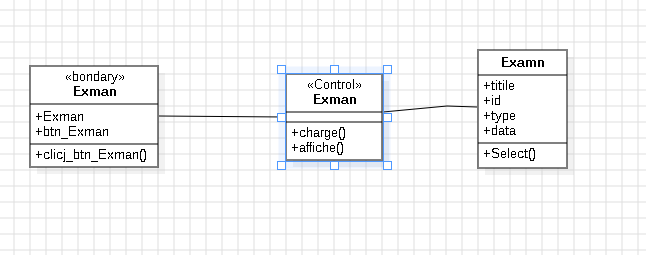


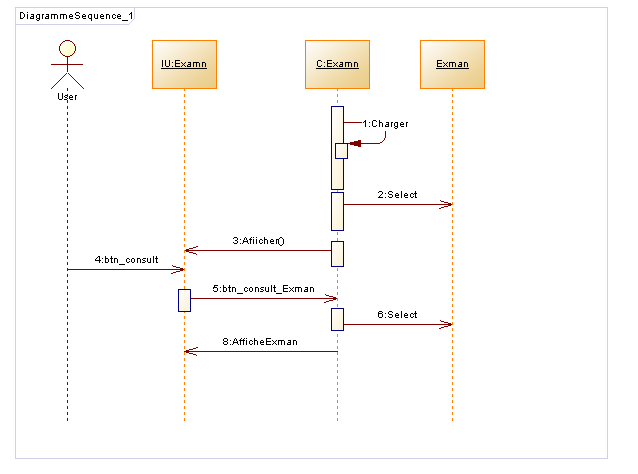
The class and sequence diagrams of “download exam”:



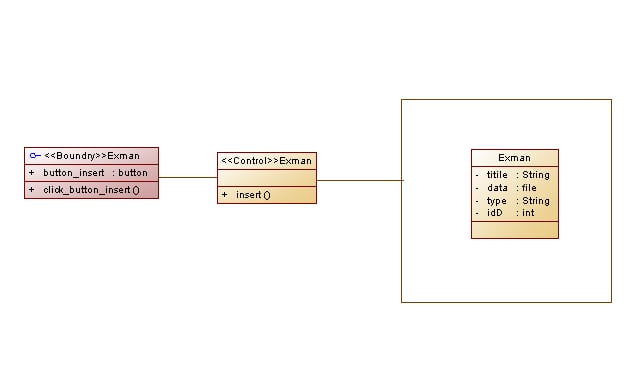


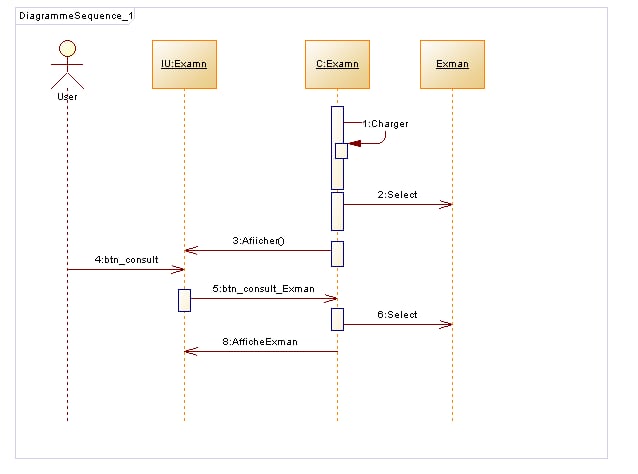
The class and sequence diagrams of “consult exam”:





The class and sequence diagrams of “add exam”:

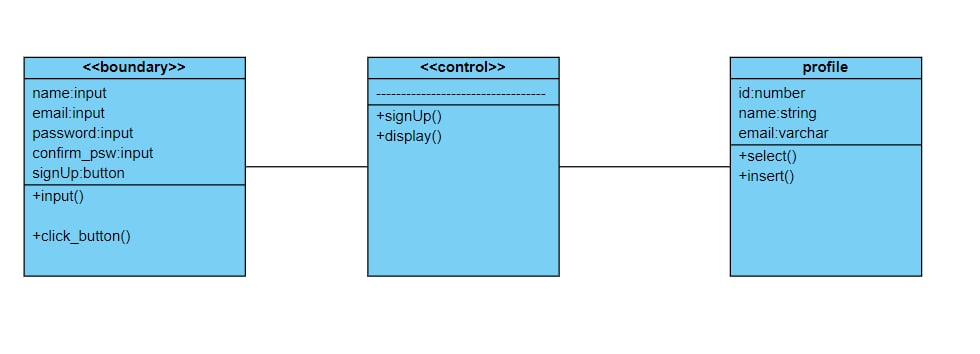


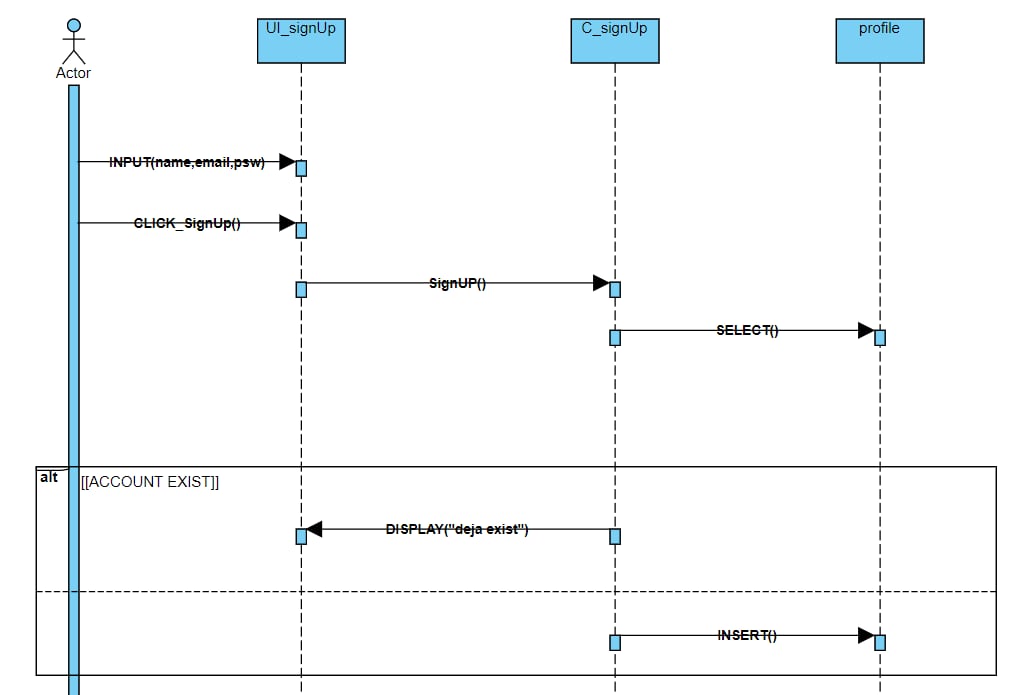


## **4.2. Manage account:**

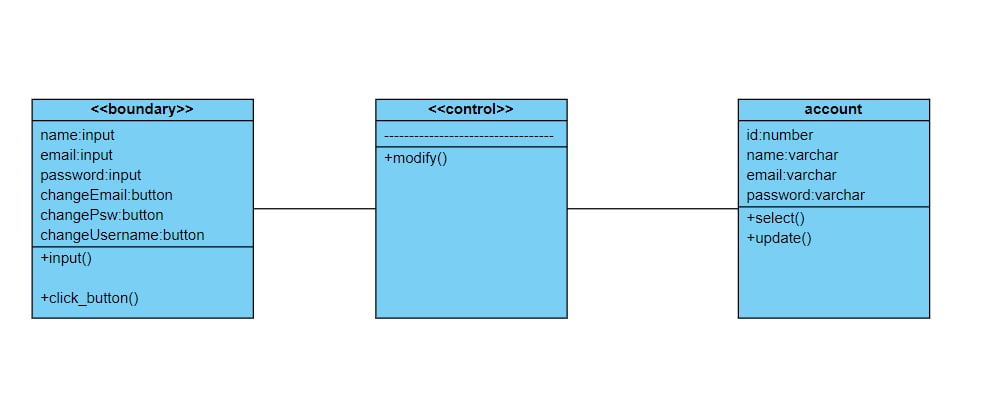
The next diagrams below represent the class and sequance diagram related to the "manage account" use case:

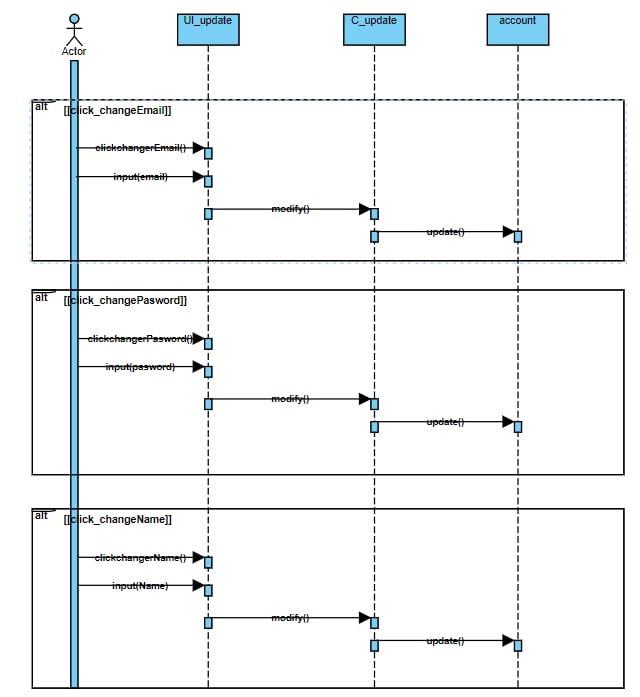
1. The class and sequence diagrams of “Inscription”:



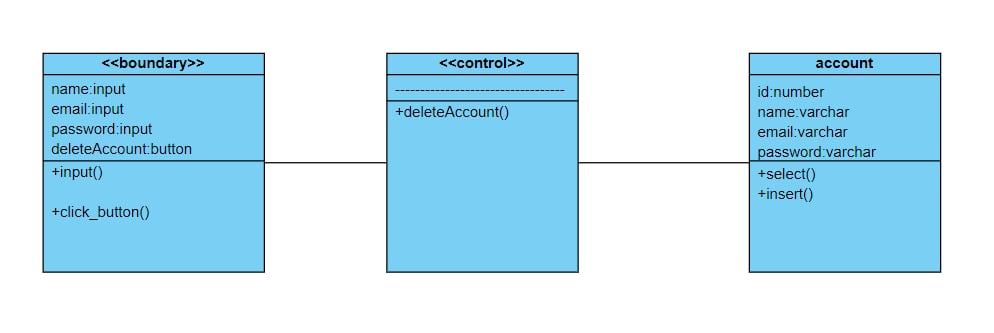


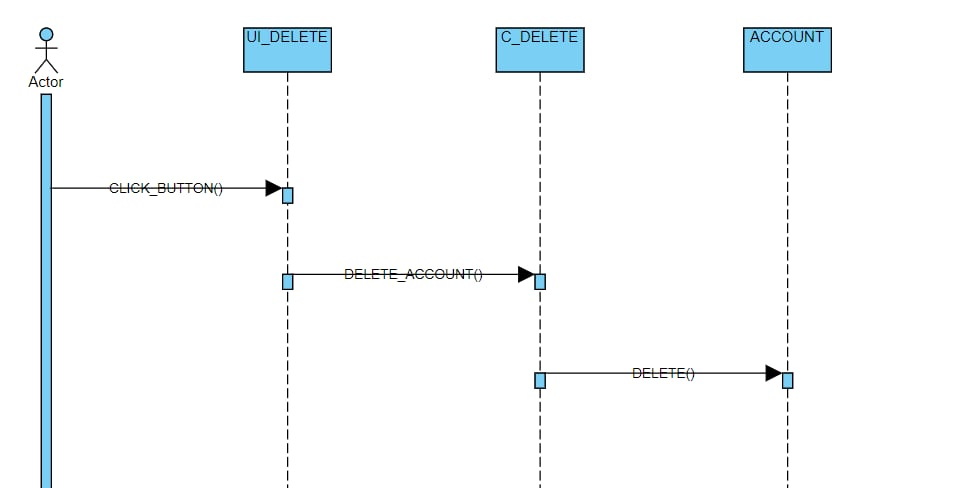
2. The class and sequence diagrams of “update account”:



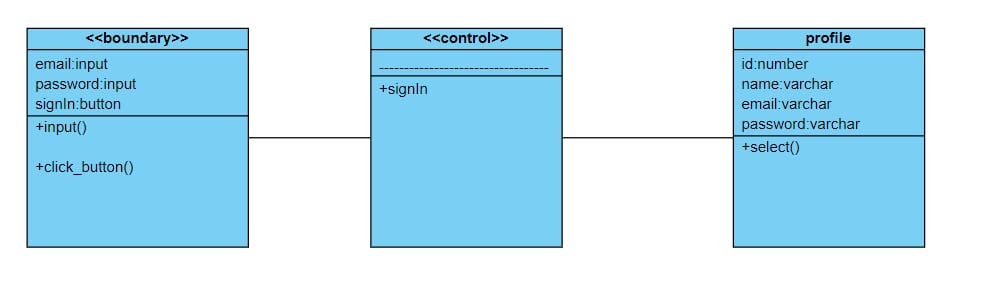


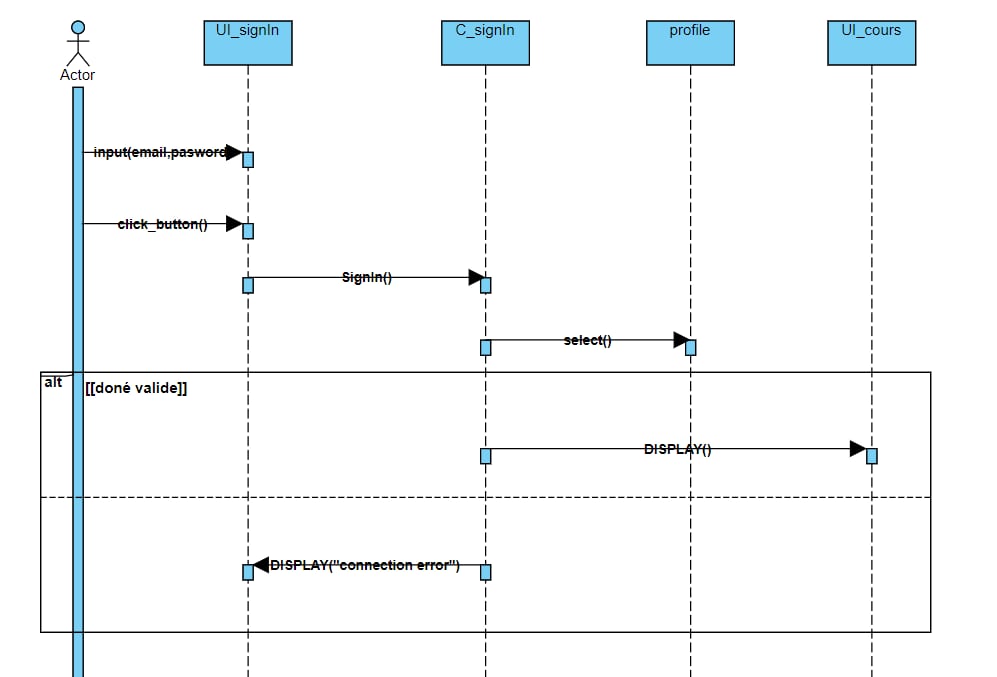
3. The class and sequence diagrams of “delete account”:





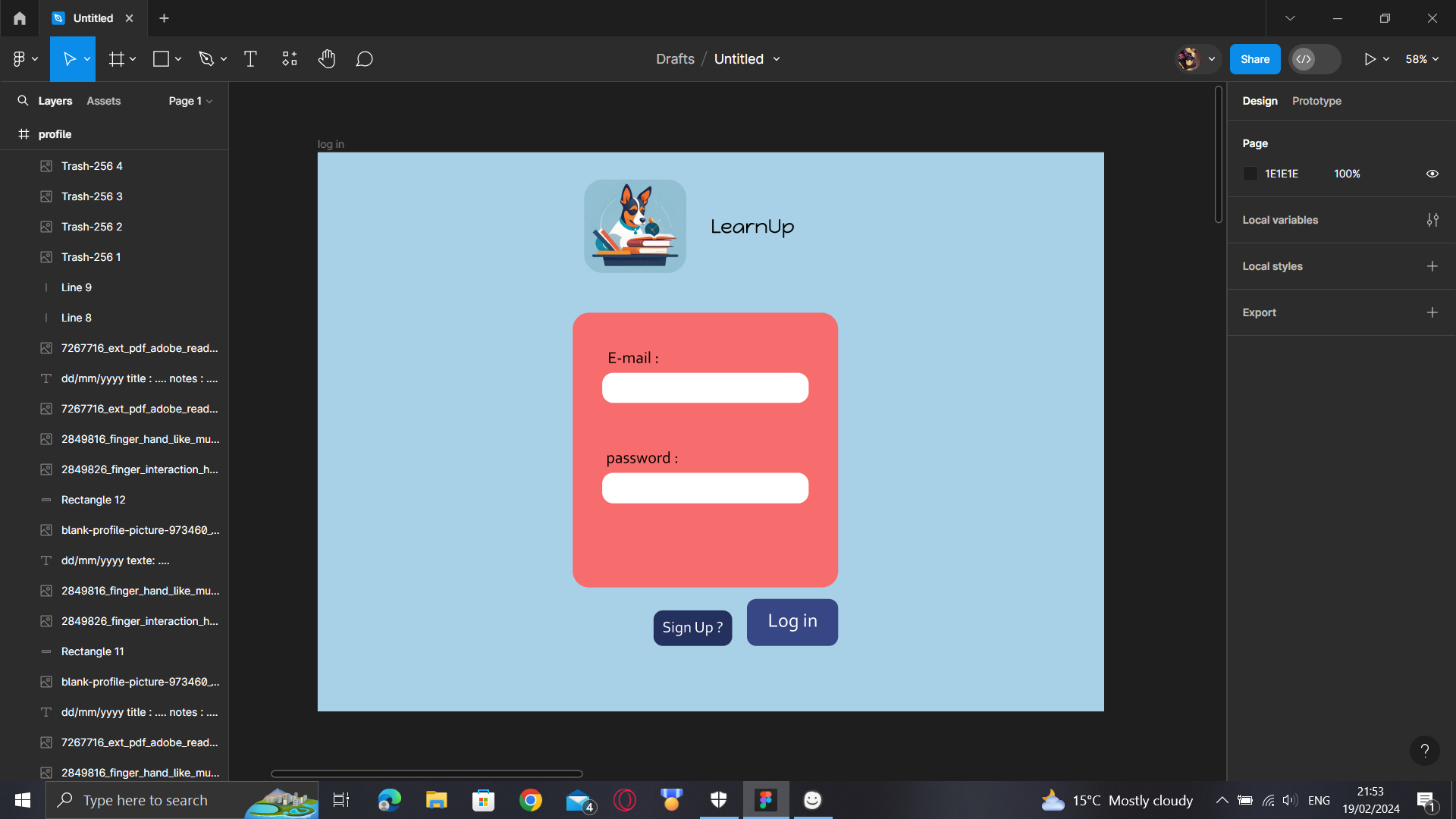
The diagrams below represent the class and sequance diagram related to the "authenticate" use case:



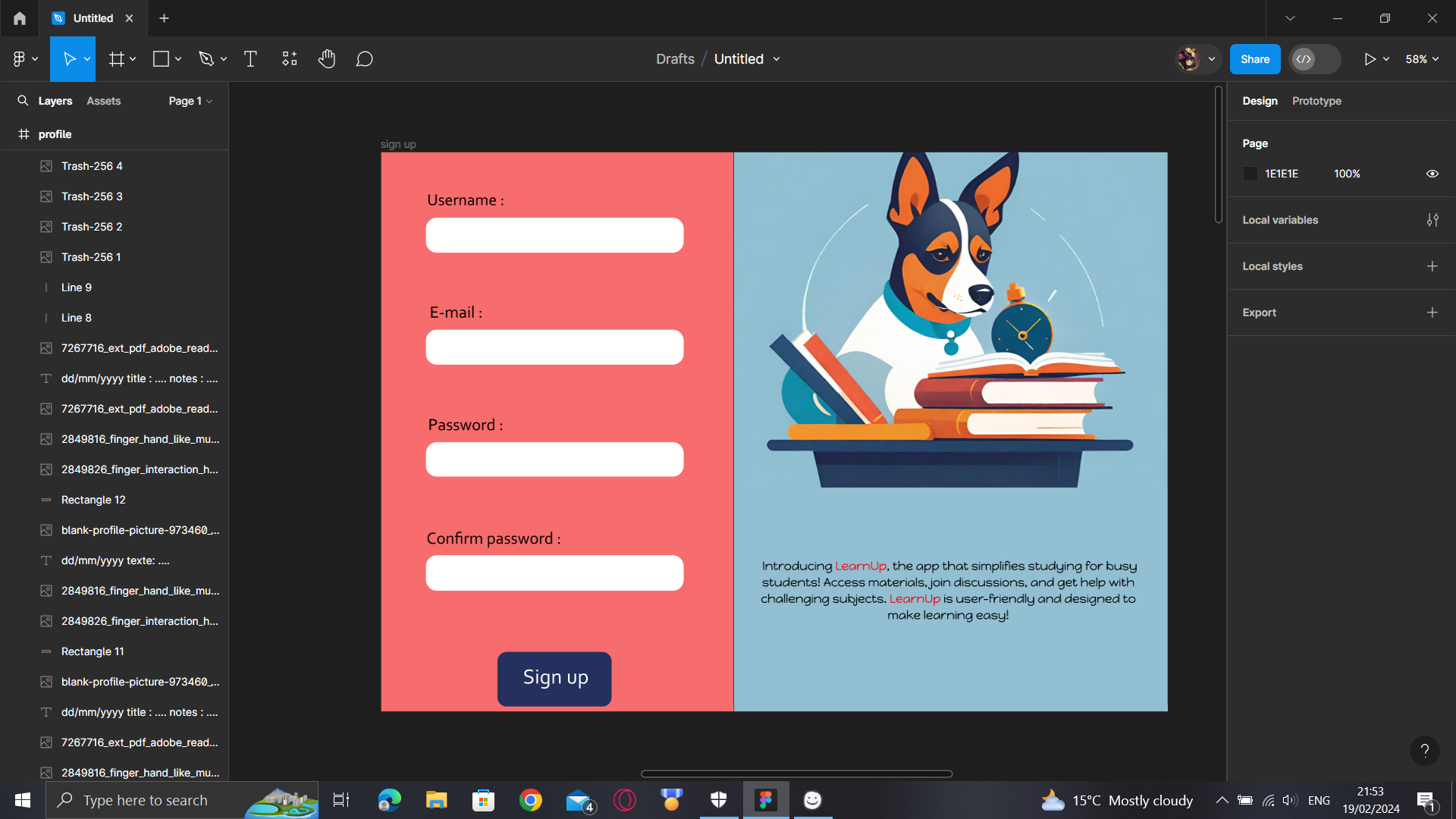


## **5. implementation of spring 0:**

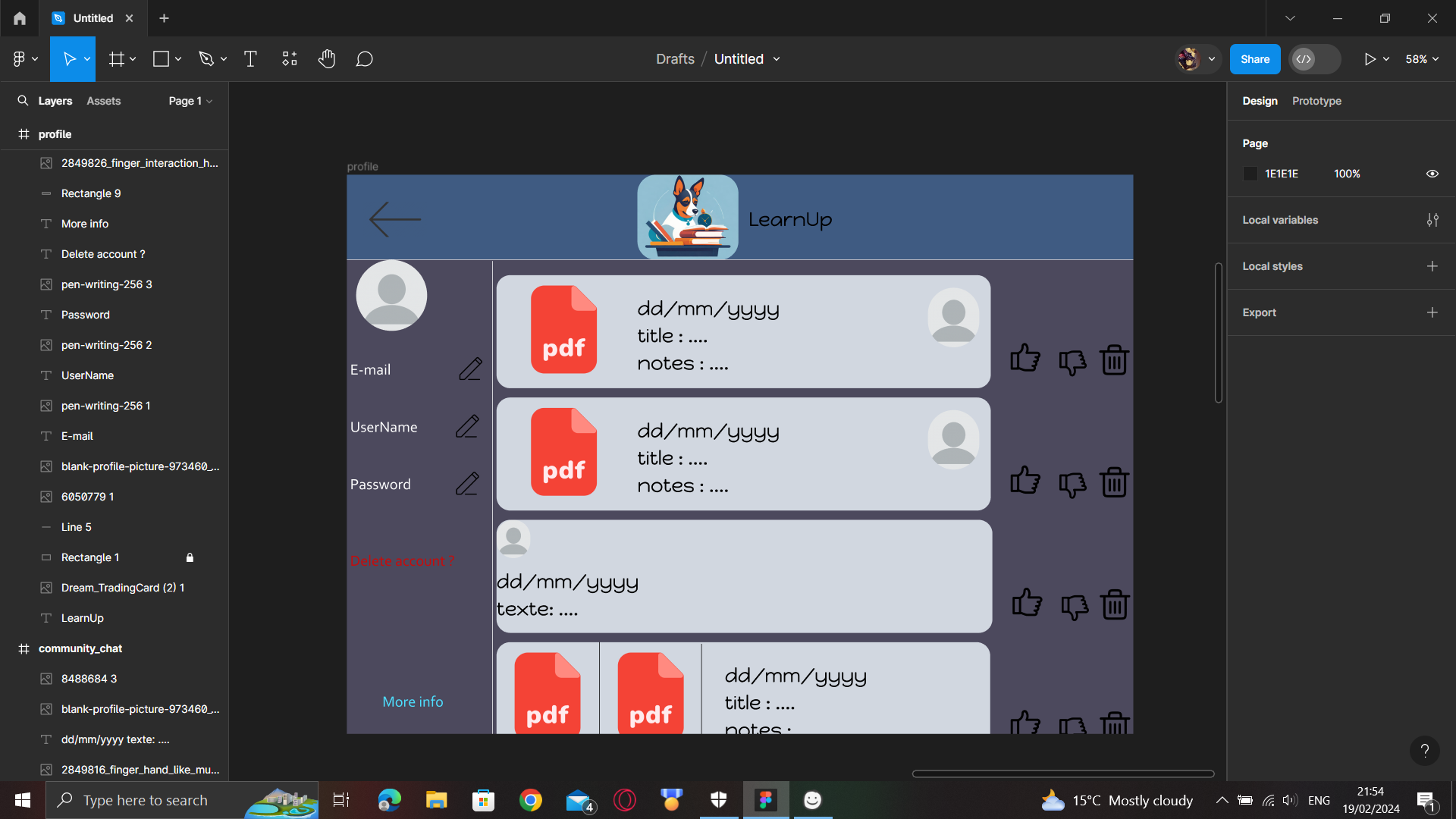
Realization of the user story “authenticate”



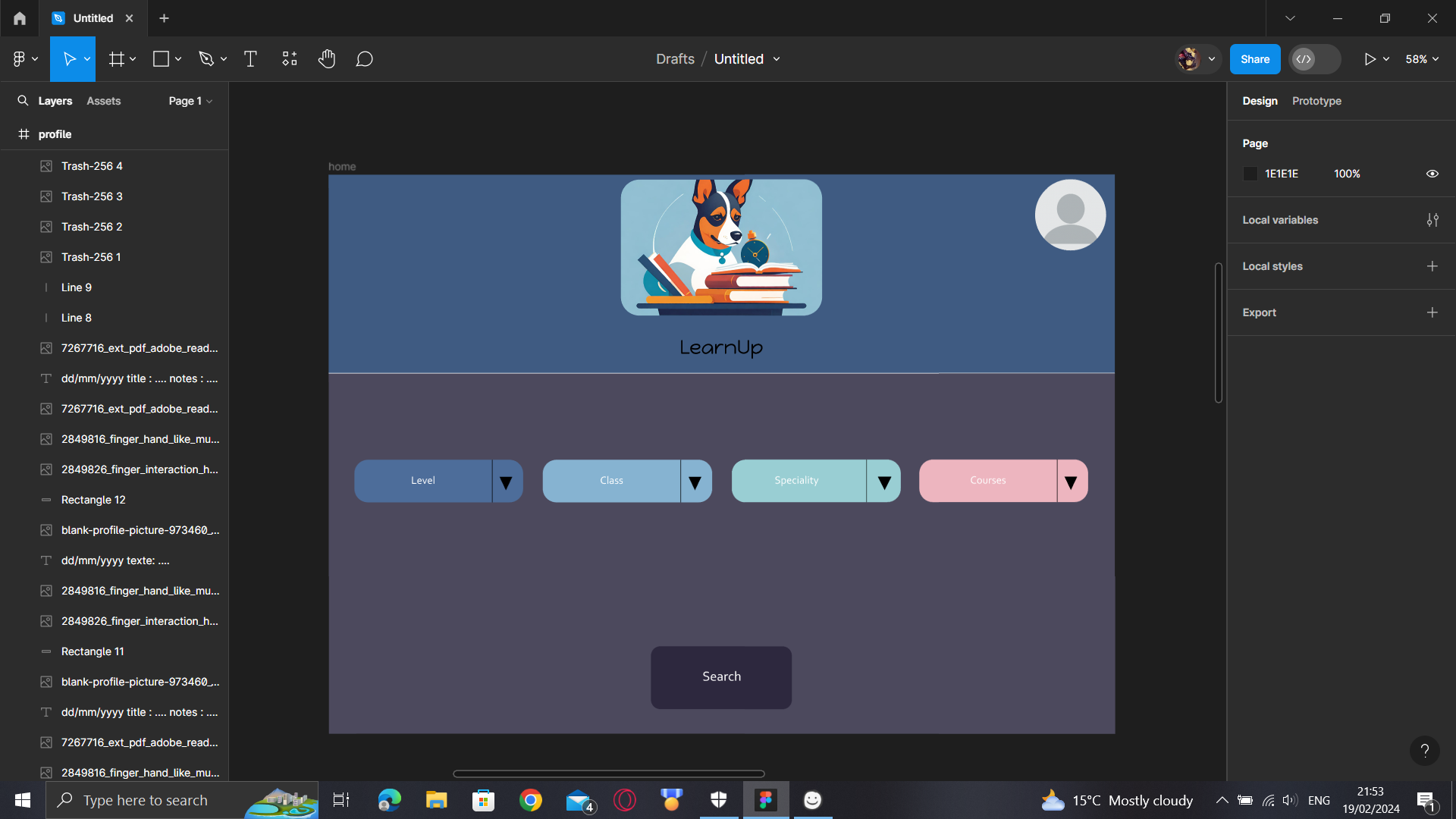
Realization of the user case “inscription”



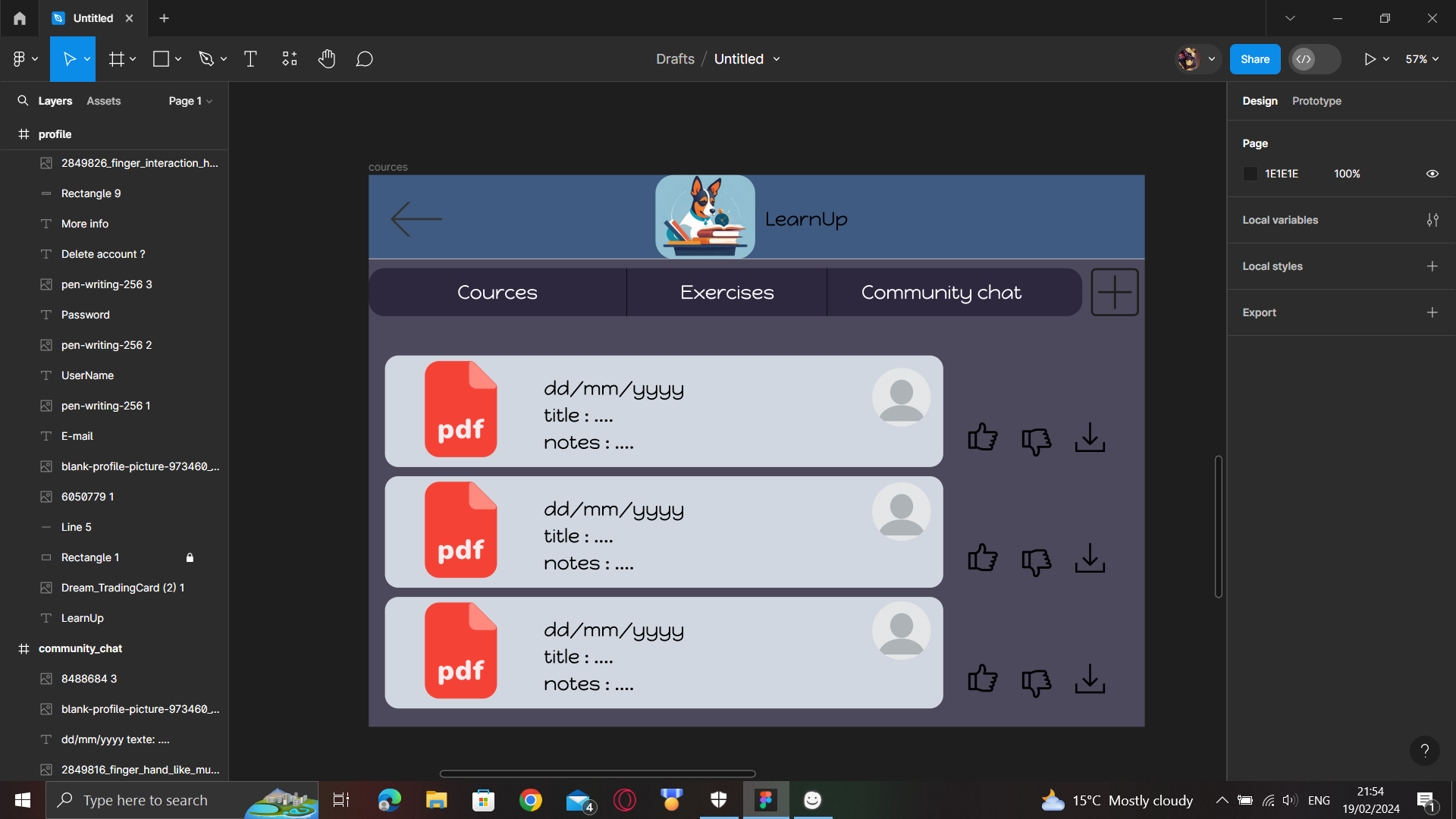
Realization of the user story “manage account”

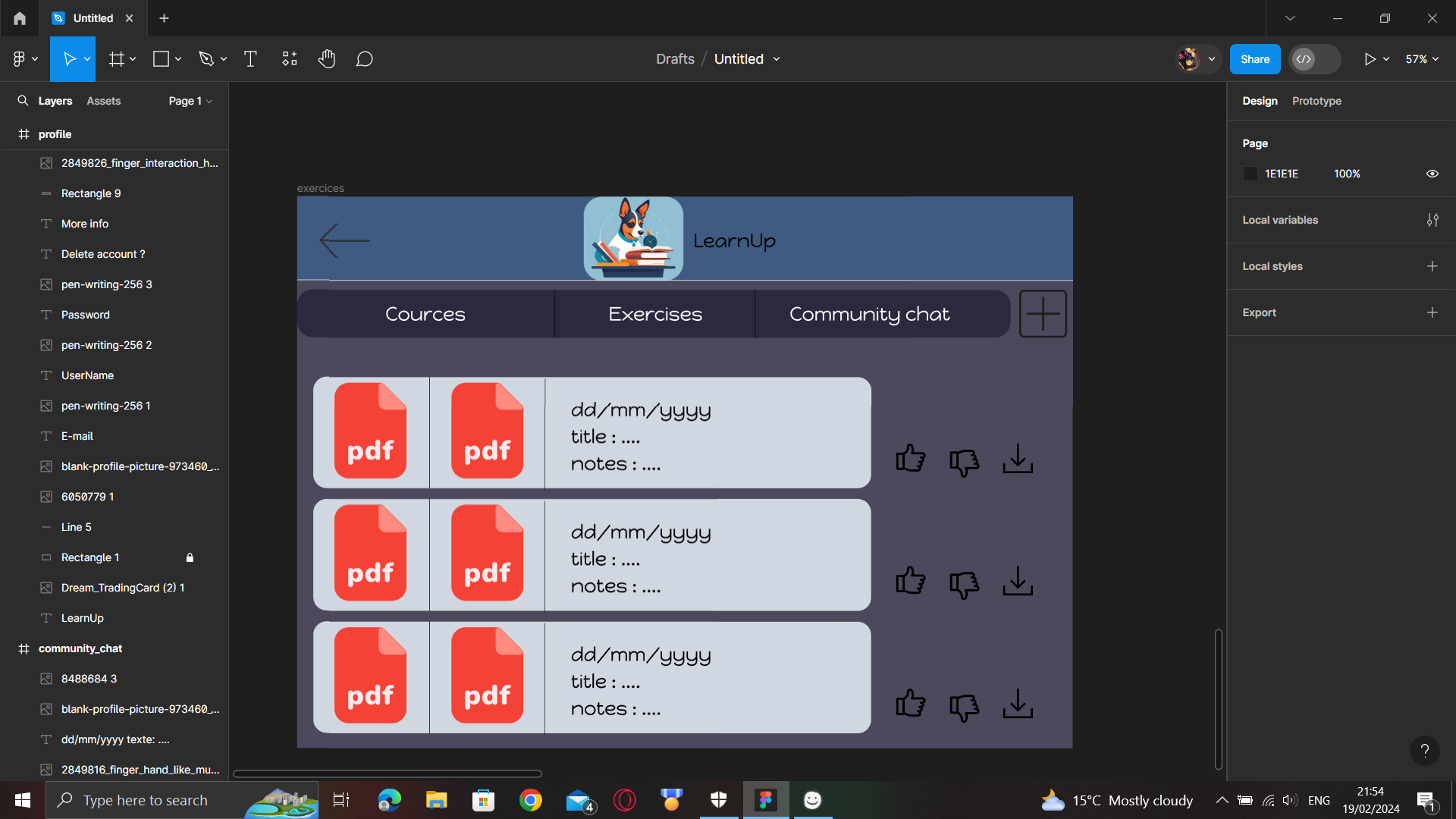


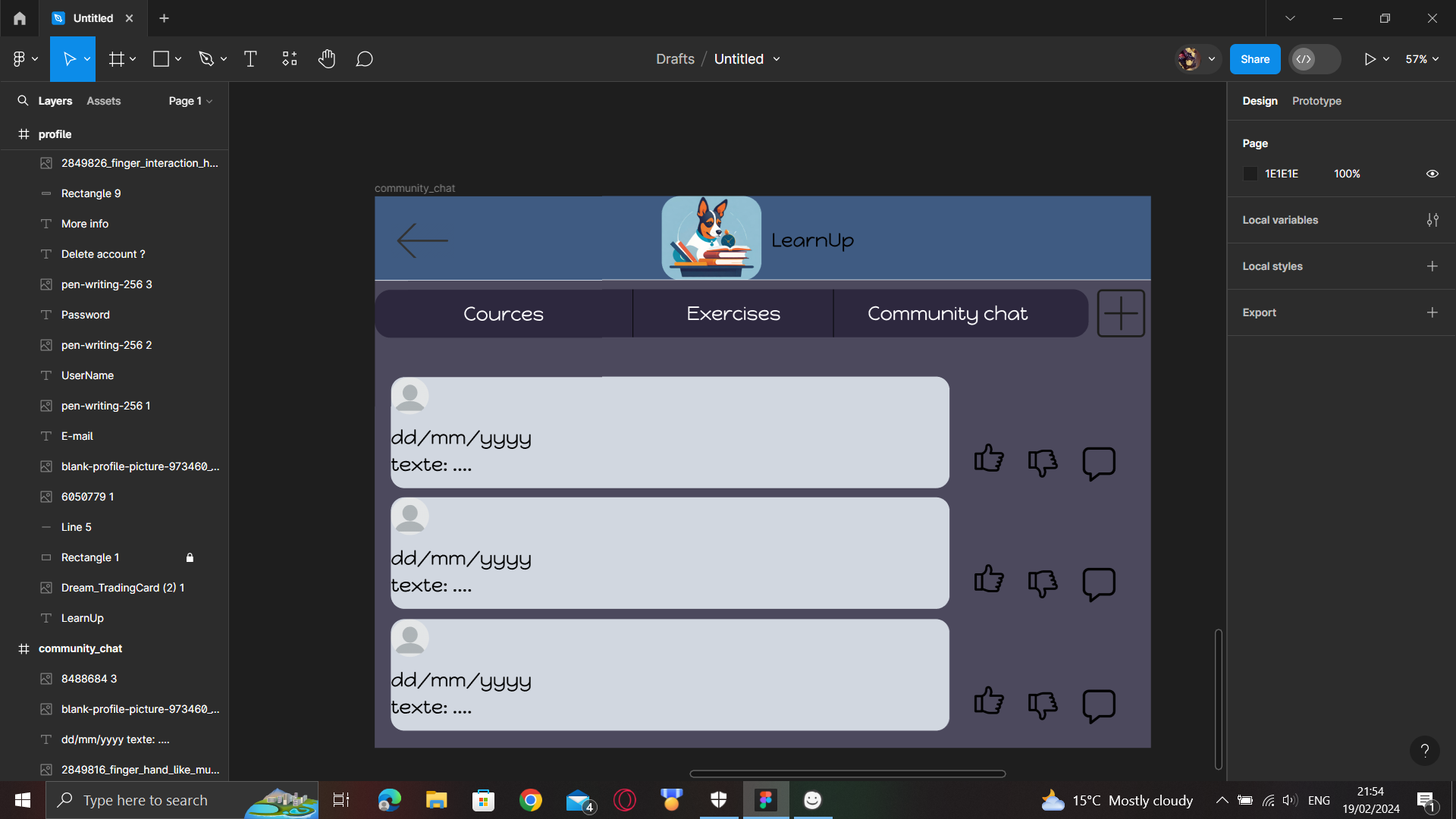
Realization of the “home page”



Realization of the pages “exercises”, “cources” and “community chat”







## **6.Conclusion:**

In this chapter, we finished working on the tasks planned for sprint 0. After that, we talked about different interfaces and looked closely at each one. This approach helps us understand not just how the release works but also how the interfaces are designed and how people interact with them.

# ***Chapter 3: Sprint 1***

## **1.Introduction:**

In the first part of Sprint 1 for our project "LearnUp", we will work on creating features that match our goals. We'll focus on building the basic parts of our product, like its main functions and design.

## **2.Identification of sprint 1 backlog:**

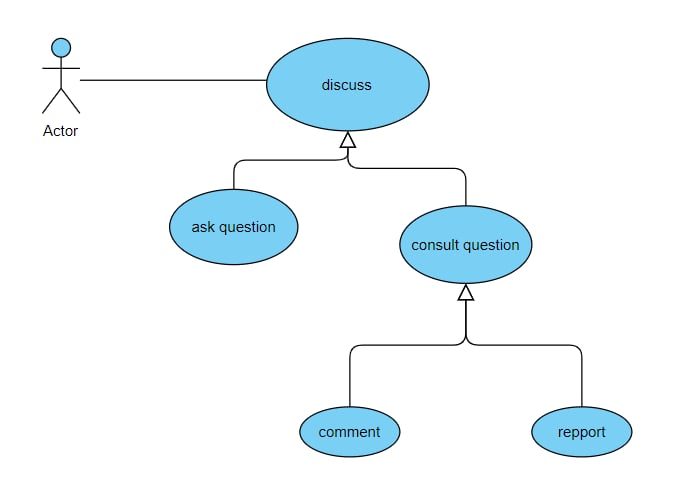
Here is the list of tasks that must be completed during sprint 1 for the initial release. This backlog details the specific items that need to be addressed to ensure a successful project start:

|  |  |  |  |
| --- | --- | --- | --- |
| As a user I can discuss. | 2 | Average | Sprint 1 |
| As a user I can voting | 2 | Average | Sprint 1 |
| As an admin I can verify documents | 2 | Average | Sprint 1 |

## **3.Refinement of sprint 1:**

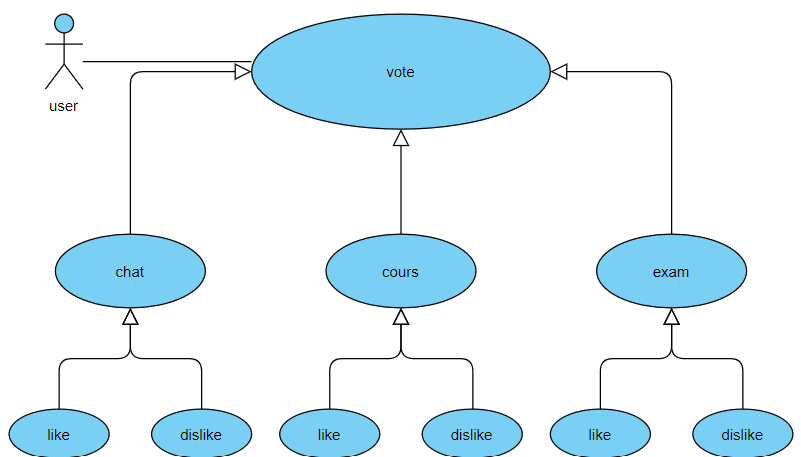
In this part, we examine the various scenarios for using the second sprint.

## **3.1. Refinement the administrator story “discuss”:**



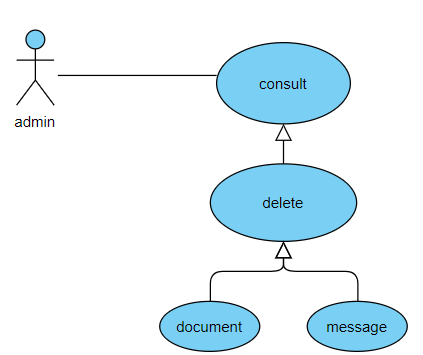
|  |  |
| --- | --- |
| **Use case scenario** | As a user, I can discuss on the chat |
| **Actors** | User |
| **Pre-Condition** | -The user must be insciripted |
| **Main scenario** | The system will post report, or delete a post in the community chat |
| **Extensions** | * Ask question * Consult question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

## **3.1. Refinement the administrator story “voting”:**



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

## **3.1. Refinement the administrator story “admin verify”:**



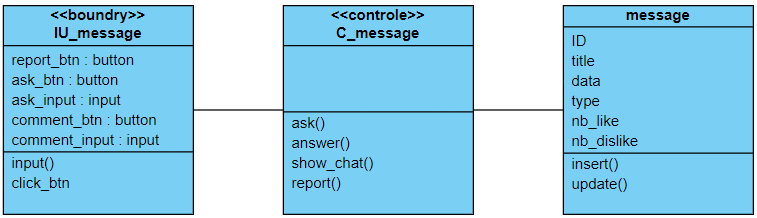
|  |  |
| --- | --- |
| **Use case scenario** | As a admin, I can consult documents and massages |
| **Actors** | Admin |
| **Pre-Condition** | -The post must be reported |
| **Post-Conditions** | Post deleted (optional) |
| **Main Conditions** | The system will delete the reported post |
| **Extensions**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

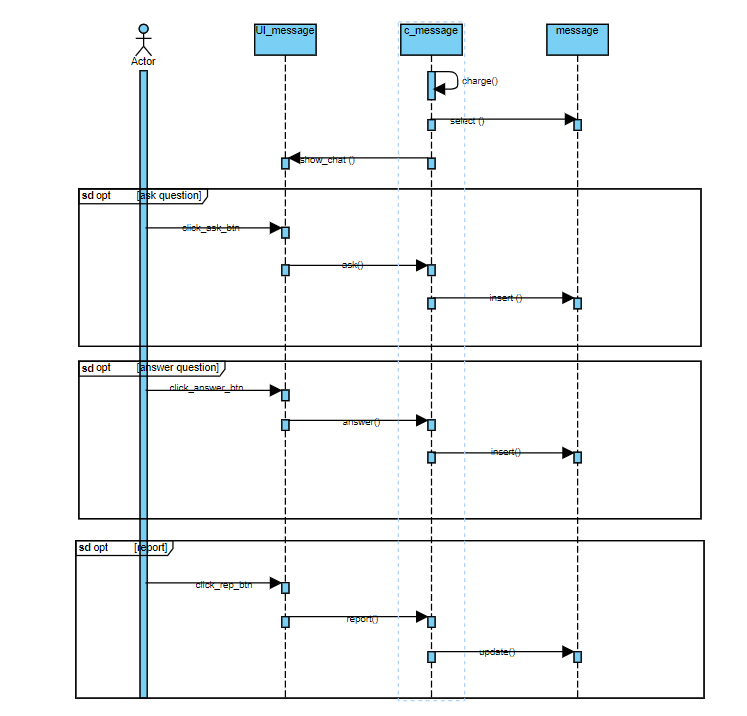
## **4.Design of sprint 1:**

We will showcase the class and sequence diagrams for the various use-case scenarios we implemented in the final section of their refinements.

## **4.1. Discuss:**

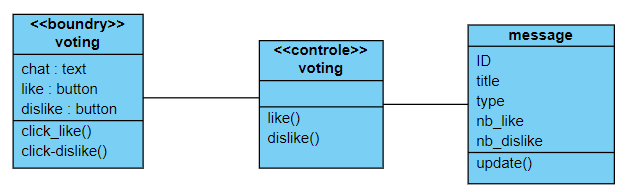
The next diagrams below represent the class and sequance diagrams related to the "discuss" use case.

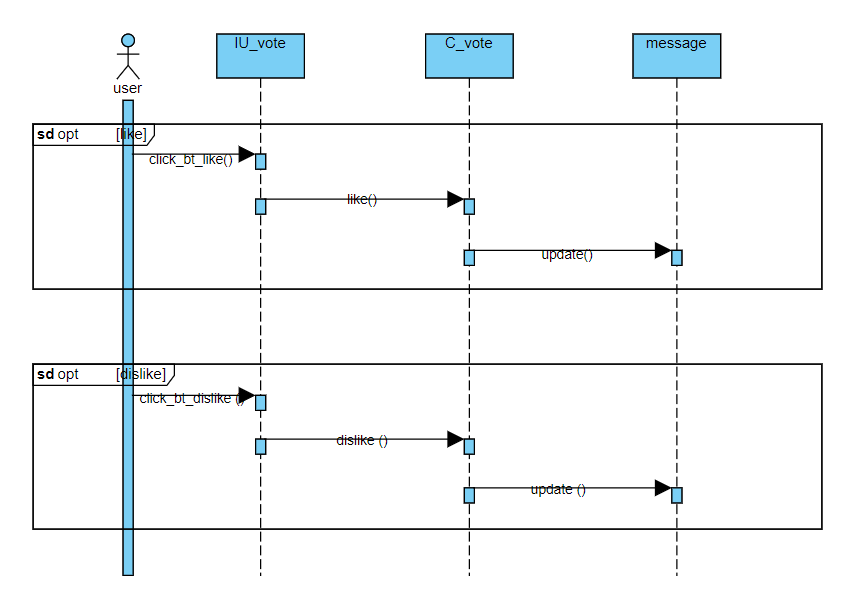




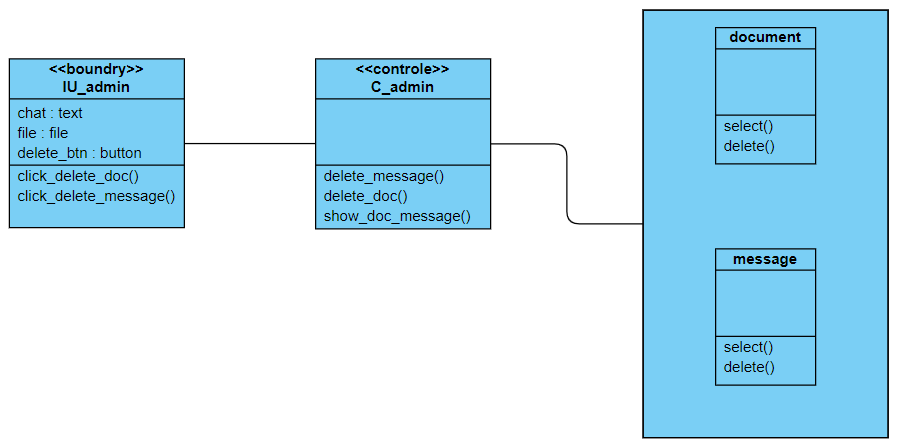
## **4.2. vote:**

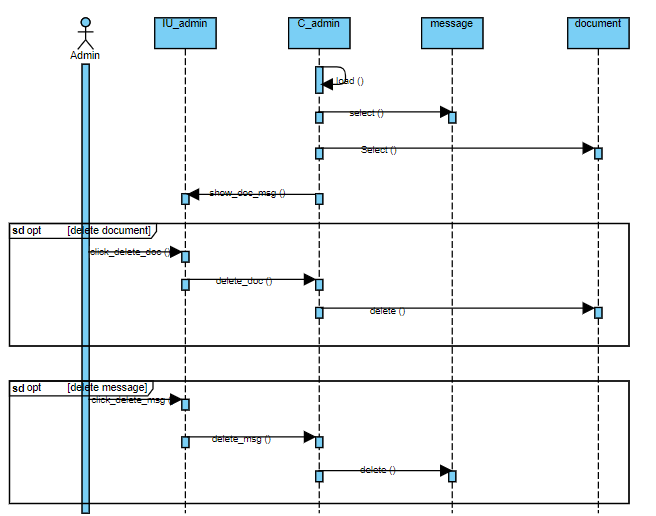
The next diagrams below represent the class and sequance diagrams related to the "vote" use case.





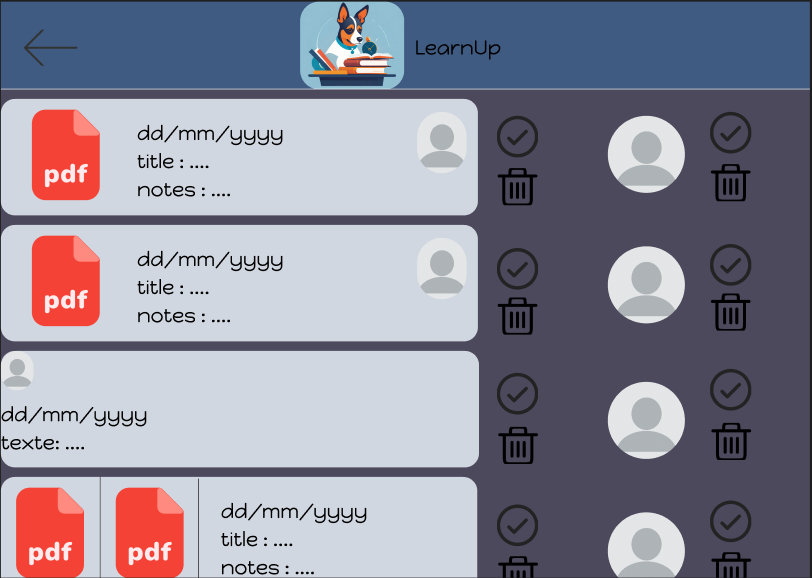
## **4.3. verify documents:**

The next diagrams below represent the class and sequance diagrams related to the "admin verify" use case.

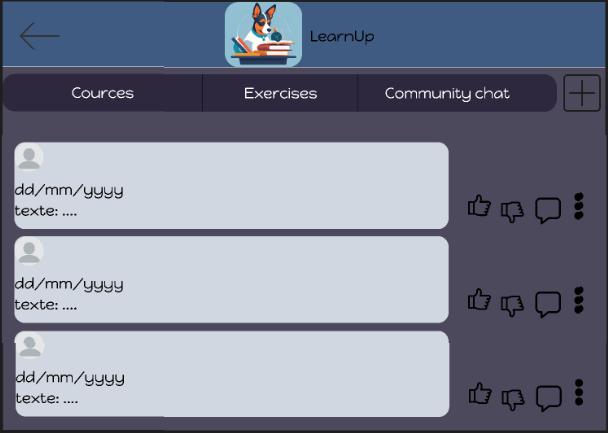


## **5. implementation of sprint 1:**

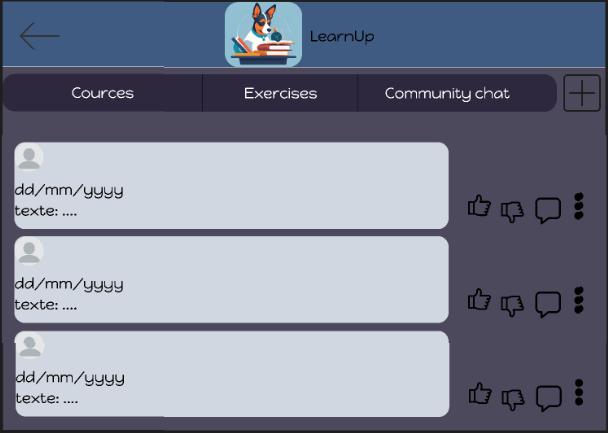
Admin UI:



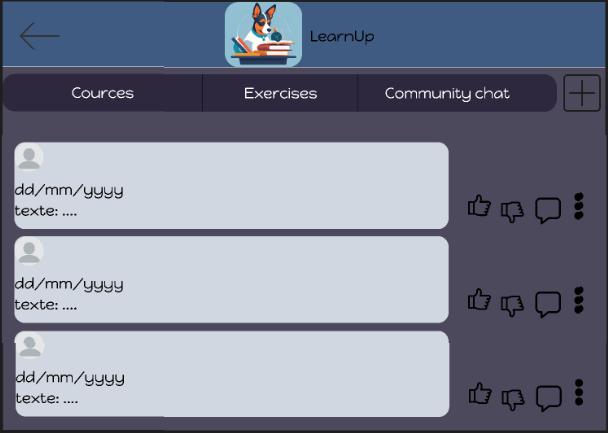
User message UI:



User voting UI:



User comment UI:



## **6.Conclusion:**

In this chapter, we completed the tasks scheduled for sprint 1. Following that, we delved into various interfaces, examining each one closely. This method enables us to comprehend not only the functionality of the release but also the design of the interfaces and how users interact with them.

# **General conclusion**

In conclusion, the completion of Sprint 0 and Sprint 1 has provided a strong foundation for our project. Sprint 0 helped us set up the project and plan our work, while Sprint 1 focused on building the core features. We also paid close attention to how users will interact with our project, which will guide our future improvements.

Moving forward, we are confident that the progress made in these sprints will drive our project's success. We have learned valuable lessons and achieved significant milestones.