

OfCourses

Requirements Specification

Final Version

June 17, 2022



Group Members:

Dhimitri Sinjari

Edison Ponari

Frenki Ruçaj

Klea Kota

Megi Dëma

Zenel Hila

Table of Contents

Table of Contents

| | | |
|---------------|--|------------|
| 1. | <i>Execution Summary</i> | 4 |
| 1.1 | Project Overview | 4 |
| 1.2 | Purpose and Scope of This Specification | 4 |
| 2. | <i>Product/Service Description</i> | 5 |
| 2.1 | Product Context | 5 |
| 2.2 | User Characteristics | 5 |
| 2.3 | Assumptions | 6 |
| 2.4 | Constraints | 7 |
| 2.5 | Dependencies | 7 |
| 3. | <i>Requirements</i> | 8 |
| 3.1 | Functional Requirements | 8 |
| 3.2 | Non-Functional Requirements | 9 |
| 3.2.1 | Product Requirements | 9 |
| 3.2.2 | Organizational Requirements | 12 |
| 3.2.3 | External Requirements | 12 |
| 3.3 | Domain Requirements | 14 |
| 4. | <i>Software Design</i> | 14 |
| 4.1 | User Scenarios/Use Cases | 14 |
| 4.1.1 | User Scenarios | 14 |
| 4.1.2 | Use Cases Diagrams | 18 |
| 4.1.3 | Use Cases Tables (Extended) | 20 |
| 4.2 | Diagrams | 51 |
| 4.2.1 | Activity Diagrams | 51 |
| 4.2.2 | State Diagrams | 65 |
| 4.2.3 | Swimlane Diagrams | 70 |
| 4.2.4 | Sequence Diagrams | 86 |
| 4.2.5 | Collaboration Diagrams | 99 |
| 4.2.6 | Class Diagram | 111 |
| 4.2.7 | Object Diagram | 112 |
| 4.2.8 | Entity Relation Diagram and Relational Schema | 113 |
| 4.2.9 | Data Flow Diagrams | 114 |
| 4.2.10 | Component Diagram | 120 |
| 4.2.11 | Deployment Diagram | 121 |
| 5. | <i>Implementation</i> | 122 |
| 5.1 | Technology Used | 122 |
| 5.1.1 | Frontend (Client-side development) | 122 |

| | | |
|-----------|---|------------|
| 5.1.2 | Backend (Server-side development) | 122 |
| 5.1.3 | Database | 122 |
| 5.1.4 | MVC Architecture | 122 |
| 5.2 | Installation Manual | 123 |
| 5.3 | Configuration Manual | 123 |
| 5.4 | User Manual | 123 |
| 5.5 | Software Screenshots | 124 |
| 6. | <i>Project Planning</i> | 134 |
| 6.1 | Activity Table | 134 |
| 6.2 | Network Notation | 135 |
| 6.3 | Gantt Chart | 135 |
| | <i>Appendix A. Sketches</i> | 136 |
| | <i>Appendix B. Prototype</i> | 144 |

1. Execution Summary

1.1 Project Overview

This is a Software Engineering Project created by 6 Epoka University students. It's meant to be a web application used for creating a platform of online courses. These courses will contain sections of lecture slides, lecture videos, and assignments.

1.2 Purpose and Scope of This Specification

The main goal is to create a web application for online courses where students and professors can collaborate. Lecturers can post courses that will be reviewed by quality assurance authorities honor website, which will be then shared with students according to tags provided by the professor. And on completion, students shall receive certifications provided by professors.

In scope:

- This project is responsible for providing a platform where lecturers carry out online courses and interact with an audience of students who will follow and engage with their programs.
- The platform will have a ready format prepared to perform all the necessary course administration aspects like adding, editing, or deleting content; grading assignments; viewing statistics, etc.
- Continuous software maintenance shall be provided.
- There will also be website security protection.

Out of Scope:

- This project is not responsible for the creation of regular course content. It shall be within the responsibility of the lecturers or instructors.
- There may be modifications or additions to website features along the way.

2. Product/Service Description

2.1 Product Context

To better illustrate this section of specifications, we can refer to the diagrams made available by our team in the project's GitHub repository: <https://github.com/eponari/OfCourses/>.

2.2 User Characteristics

Admin:

- Validate courses
- Manage courses, instructors, and students
- View course's and instructor's rating
- Edit credentials
- Log in
- View about/contact
- View available courses
- View most preferred courses
- Delete/add/edit course
- Check courses and homework statistics

Lecturer/Instructor:

- Add/edit/delete courses
- Assign/edit/delete homework
- Check courses and homework statistics
- View available courses
- View most preferred courses
- View about/contact
- Log in
- Edit credentials
- Check and print grades

Student:

- View available courses
- View most preferred courses
- View about/contact
- Log in
- Edit credentials
- Check and print grades
- View course's and instructor's rating
- Cancel course
- View taken course
- Give instructor's and course's rating
- Take course

Guest:

- View available courses
- View most preferred courses
- View about/contact

2.3 Assumptions

- Our assumption is that it's feasible to deliver this project using the available technologies.
- The staff/team members obtain the technical skills to follow through with the concept.
- It is assumed that the lecturers who will be using the page, will have technical expertise in the area of study and provide the necessary materials for their courses.
- The students shall have access to all the needed resources in order to finish assignments and complete courses successfully.

2.4 Constraints

- Deadline: End of current semester.
- Technologies (programming languages and framework): PHP, HTML, CSS, Bootstrap, MySql.
- The content available in the course pages should be provided by the lecturers themselves.
- All professors must have a certificate/license and professional background on the field in order to fulfill the criteria for the application.

2.5 Dependencies

- This product will require regular updating of course content in order to execute all the required features.
- The requirement for the deadline feature should be completed before implementing a way to track the time of submission.
- The product should carry out the feature to view submissions without needing to download them before a method to unzip files automatically in the app can be built.

3. Requirements

3.1 Functional Requirements

| Req# | Requirement statement | Priority | Date Reviewed | SME Reviewed / Approved |
|-------|---|----------|---------------|-------------------------|
| FR-01 | The website should have a deadline feature to easily time assignments as well as a tracking system for when the assignments get turned in. | 1 | 10/04/2022 | Ari Gjerazi |
| FR-02 | The website should also offer a way to see how many times the student has modified the assignment after turning it in. | 2 | 10/04/2022 | Ari Gjerazi |
| FR-03 | The website should provide a method to view or unzip zipped files automatically in the application. | 3 | 10/04/2022 | Ari Gjerazi |
| FR-04 | The website should give the lecturer the ability to view a submission without needing to download it first. | 2 | 10/04/2022 | Ari Gjerazi |
| FR-05 | The website should have the content of the current week pinned at the top of the course page. | 2 | 10/04/2022 | Ari Gjerazi |
| FR-06 | The website should split the content into categories or have it organized in a menu for submissions/assignments, announcements, materials/slides/books/lecture notes, grade list and so on. | 1 | 10/04/2022 | Ari Gjerazi |
| FR-07 | The website should have a notification pop-up in the event that there is a new or unchecked post. | 2 | 10/04/2022 | Ari Gjerazi |
| FR-08 | The website should offer a plagiarism checking tool, such as StudyMoose, in order to compare submitted documents for similarities. | 3 | 14/04/2022 | Igli Hakrama |
| FR-09 | The website should arrange the content in chronological order week by week from top to bottom of the course page. | 2 | 14/04/2022 | Igli Hakrama |

3.2 Non-Functional Requirements

3.2.1 Product Requirements

3.2.1.1 *User Interface Requirements*

General UI requirements:

- The user interface should be flexible for different screen sizes since the product will be aimed to be used as a web application.
- The fonts used will only be of sans-serif font family to ensure professionalism.
- Each page will have a navigation bar with tabs for each type of user.
- The general first page will show general courses for given tags, about the page and a login/signup page for users.
- The login and signup pages will be minimalistic. The signup page will require the user to fill their name, email and password. The login page will require the user to fill their email and password.

User-specific UI requirements:

- Student:
 - The dashboard will contain “My courses”, “Ask the professor” and “My Profile” tabs.
 - “My courses” page will contain a list of courses the student has registered in. Using colors, the courses will be divided into failed(red), ongoing(blue) and finished successfully(green).
 - Each course when opened will enlist material of the course separated in weeks. The material can be documents or assignment. Assignments will also be color coded like the courses to show failed assignments, ongoing assignments and finished assignments.
 - “Ask the professor” page will contain questions and answers that you can do to a professor for a specific course.
 - “My profile” page will allow the user to change their name, email and password.
- Professor:
 - The dashboard will contain “Manage courses”, “Ask the professor” and “My Profile” tabs.
 - “Manage courses” page will contain a list of courses the professor has created. Using colors, the courses will be divided into rejected by administrators(red), ongoing(blue) and finished successfully(green).
 - Each course when opened will enlist material of the course separated in weeks. The material can be documents or assignment. Assignments will also be color coded like the courses to show ongoing assignments and finished assignments. The professor can change the number of weeks for the course and the layout will change.
 - “Ask the professor” page will contain questions from students to be answered by the professor.
 - “My profile” page will allow the user to change their name, email and password.
- Administrator:
 - The dashboard will contain “Validate courses” and “Users” tabs.
 - “Validate courses” will contain courses that have been created by professors. Administrator can reject, accept or delete a course depending on its content.
 - “Users” table will contain all users that aren’t administrators and the administrator can delete an account from them.

3.2.1.2 Usability

The product should be easy to learn and work with. The focus should be to make it easy to understand even for non-tech professors. A manual should be created to be handed to all types of users so they can learn the major functionalities of the web application.

Errors should be handled with friendly warnings for the end users to make the user experience satisfactory. On the other hand, risky operations such as updates or deletion of records should be handled with explanatory prompts for the users so they know the impact they are making.

3.2.1.3 Efficiency

3.2.1.3.1 Performance Requirements

- o Latency from the website will be less than 0.1s.
- o The server should be able to handle around 2000 users using it simultaneously.
- o An average traffic would be around 100 students/hour.
- o System should support a number of transactions of 10/sec.

3.2.1.3.2 Space Requirements

Assuming that each course will have 10 assignments for 100 students that will upload a file of 10kB and around 120MB of materials, each course will need around 130MB. Assuming that on average 50 courses are going to be posted each year, 6.5GB is going to be needed for each year. Since operations on this system aren't going to have a complex nature, a 1TB drive would be a good storage device for 15 years.

3.2.1.4 Dependability

Availability

- The system will be running 24/7 since the material should be accessible for the students and the professors at any time.
- The product will have a tolerance of 30 minutes of down time a week, typically in weekends when traffic on the website is reduced.
- It is intended to be used in any geographical area in Europe.

Reliability

The system should be reliable during school hours for professors to access their materials and during late hours since most deadlines are scheduled for 23:59. Downtime should be scheduled for early hours of the day, 0:00AM- 5:00AM since the traffic would be the lowest.

Monitoring

Logs will be kept for user access and user changes so any change on the system can be tracked back to the users. CPU usage and RAM usage of the server should always be kept under 80% and the space usage should always be kept under 90%.

Maintenance

Maintenance should be scheduled during school spring/summer break since students will not be using it during that time.

Integrity

The integrity of the system will always be kept using logs for any change on the system and using HTTPS and TLS the security of the system will be secure. Usability of the system will be kept on check time after time so the integrity of the system will be safe.

3.2.1.5 Security

- Encryption: Passwords saved in the database will be hashed using `password_hash()` function in PHP. This will make sure even administrators don't have access to password of users.
- SQL Injection: Using PDO library, SQL statements are safe from SQL injection.
- Cross-Side Scripting: Input from user will be put through `htmlentities()` to protect the site from Cross-Side Scripting.
- HTTPS and TLS protocols: Communication channels of our product will use HTTPS and TLS for data encryption and data safety.

3.2.2 Organizational Requirements

3.2.2.1 *Environmental Requirements*

The product shall be able to operate on any device, such as laptop/PC or mobile, so long as Internet Connection is provided. The website can be easily accessed at any time and space.

Provided that you are still logged into your personal account, you can be alerted of any new posts or important information through the notification pop-up on your screen.

3.2.2.2 *Operational Requirements*

This software is a web application which will provide a platform available to all different types of users. Both professors and students will benefit from the straightforward and efficient system that is our online courses website. Students shall easily interact with the lecturer's material/content, as well as approach them with direct questions about the lessons and such. The instructors will have a ready format to comfortably grade and evaluate students' performance. Lastly, admins will be able to manage the data that is accessible to them, like courses and user information of students and lecturers.

3.2.2.3 *Development Requirements*

The technologies that are going to be used in this project are as follows:

- Front-End – HTML, CSS, JavaScript, Bootstrap
- SQL language – MySQL
- Back-End – PHP

3.2.3 External Requirements

3.2.3.1 *Regulatory Requirements*

Our website will be sure to provide a privacy policy that clearly states how the data of our customers and guests is handled and processed. It will explicitly describe whether certain information is kept confidential or shared with a third party.

All content of the web page is protected by copyright, upon creation. This law includes any “material that is perceptible to the users of a particular website,” so it pertains to text, images, music, audio, and video. We will also make sure to add copyright notices and symbols when necessary.

3.2.3.2 Ethical Requirements

Users must agree to sharing a certain amount of their personal data within the system. Instructors should agree to sharing their personal information about:

- Full name.
- Email address.
- Qualifications – background in area of expertise, i.e. diplomas, certificates etc. Their data will be accessed by admins and students for educational purposes only.

Students should agree to sharing their personal information about:

- Full name.
- Email address.
- Preferences in subjects they're interested in, through tags that they select after signing up.
- Previous studies in similar modules that they want to pursue further.

Their data will be accessed by admins and instructors for educational purposes only.

3.2.3.3 Legislative Requirements

Our software will comply with Albanian laws, standards and regulations imposed by law enforcement agencies of the Albanian government. Our data management policy will ensure that the organization data and information assets are managed consistently and used properly, through a set of documented guidelines.

This includes: continually improving end-user awareness of proper data management; maintaining physical security of data infrastructure; implementing appropriate data access; providing data cataloging technologies to users.

3.2.3.3.1 Accounting Requirements

The application will not require the users to make any monetary transactions within the system. The platform will be absolutely free of charge for all kinds of end-users.

3.2.3.3.2 Security Requirements

The system will provide an appropriate security level for users through encryption. The passwords saved in the database will be hashed in PHP, which will exclude even admins in accessing them. Other security methods will be provided with SQL Injection, Cross-Side Scripting and HTTPS and TLS protocols.

3.3 Domain Requirements

The domain requirements for our software will be considering the framework of an educational course system. The prime objective of the application is to provide a platform where professors and students are able to interact freely. Each type of user has a set of actions they can perform on the site depending on their profiles or account characteristics. There are four main kinds of users that can access the web page, including: the administrator, instructor, student and guest.

4. Software Design

4.1 User Scenarios/Use Cases

4.1.1 User Scenarios

Guest scenarios

1. Scenario title: *Successful user log-in*

- a. The user opens OfCourses's homepage in order to get logged in to the system.
- b. After clicking the Login button, he is redirected to the login page.
- c. At the login page, he is prompted to enter his username or email and his password. After entering his credentials correctly, he clicks the Login button.
- d. The Login is successful and the user now is logged in to the system and ready to use it.

2. Scenario title: *User forgets his password*

- a. The user opens OfCourses's homepage in order to get logged in to the system.
- b. After clicking the Login button, he is redirected to the login page.
- c. At the login page, he is prompted to enter his username or email and his password. After entering his credentials, he will not be accepted by the system. An error message occurs "Invalid email/username or password".
- d. The user rechecks his email/username and password. If he gets the same error message and his email/username are correct, he might have forgotten the password.
- e. At the bottom there is an option called "Forgot password". That hyperlink is clicked by the user and he is redirected to the "Reset password" page.
- f. At "Forgot Password" page he is prompted to enter his email/username, and after typing it he submits it.
- g. The system automatically sends an email to the user with an automatically generated password.

3. Scenario title: *View available courses*

- a. After visiting the OfCourses, the user can see the homepage. There is an option for any user to check the available courses that are being taught at the moment, or will be taught in a short period of time
- b. The user clicks "View available Courses" and all the courses will be showed.

Student Scenarios

4. Scenario Title: *Take a course*
 - a. When a student enters the OfCourses system, he can choose to enroll in a certain course. In the dashboard he can choose the tags that he is focused on or he likes to study.
 - b. According to these tags, to his right, a certain amount of courses will be showed and the student can choose to enroll in whatever course he wants (if he meets some requirements).
 - c. In these courses showed up, the user can click on two buttons. “More information” button redirects the student to the information about that course page, where he can be informed about the syllabus, HWs or projects that course might contain.
 - d. The “Register” button enrolls the student to that course, and the redirects him to that course home page.
5. Scenario title: *Cancel Course*
 - a. For different reasons, the student wants to be unenrolled from a certain course.
 - b. At the course page, the student can click the button “Unenroll from this course”.
 - c. He will get a verification window that makes sure that the student wants to be unenrolled.
 - d. If the user clicks “Yes, unenroll me” button, he will be unenrolled and he will not be assigned more HWs, projects, or receive more notifications from that course.
 - e. If the user clicks “Cancel” button, he will not be unenrolled and still be part of that course.
6. Scenario title: *Check grades*
 - a. The user wants to be informed about his performance in a certain course.
 - b. At the course page, the student can click the “Check grades” button.
 - c. Different sections with grades will be provided, with projects, HWs or even Attendance.
 - d. The students can print those grades by clicking “Print PDF” button.
7. Scenario title: *View Taken Courses*
 - a. The user wants to be informed about the courses that he is attending right now.
 - b. At the student’s dashboard, “View Taken Courses” button will help the student by showing him all the courses that he is enrolled.
 - c. For each course, there will be two buttons. One button “Unenroll me from this course” will unenroll the student from that certain course.
 - d. “More information” will provide information about the course and the student’s grades.
8. Scenario title: *Edit credentials*
 - a. For some reason, the student wants to edit some information about him.
 - b. In his dashboard, at the “View profile” button he can see his personal information and courses he is enrolled.
 - c. Another button “Edit” will provide the student the opportunity to edit his personal information.

Instructor Scenarios

9. Scenario title: *Add course*
 - a. After logging in, the instructor is redirected to his dashboard.
 - b. The button “Add course” provides the instructor the opportunity to upload a course.
 - c. The above step will ONLY happen if the Administrator validates the instructor’s course.
 - d. The course cannot be uploaded without the Administrator permission.
10. Scenario title: *Edit course*
 - a. After the instructor is logged in, he can go to one of his courses he is teaching.
 - b. In the course page, the button “Edit” provides the opportunity to edit some information or some lessons about that course
 - c. After making changes, a window will pop up.
 - d. At the window, the button “Confirm” will make the changes that the instructor made.
 - e. The button “Cancel” will delete all the changes and restore everything to normal.
11. Scenario title: *Delete course*
 - a. For some certain reasons, the instructor wants to delete the course.
 - b. At the course page, if the button “Delete” is clicked the course will be deleted.
 - c. Before it deletes, a window will pop up with a confirmation.
 - d. If the button “Delete” is clicked, the course will be deleted.
 - e. If the button “Cancel” is clicked, the course will not be deleted.
 - f. The instructor can NOT delete a course without the Administrator confirmation.
12. Scenario title: *Check course’s Statistics*
 - a. After entering the course page, the instructor can take a look at how the course is progressing
 - b. After clicking “Check statistics” button, a page with all information about the students and course will be showed.
 - c. The instructor can check the attendance list by clicking “Attendance”.
 - d. The instructor can check the grades for HWs or projects by clicking “Grades”.
 - e. The instructor can check students enrolled by clicking “Students”.
13. Scenario title: *Assign HW*
 - a. The instructor can assign HWs to the students enrolled.
 - b. After entering the course page, at the “Manage HW”, the instructor can click “Assign” button to assign a certain HW to the students.
 - c. He will be able to drag and drop or upload the file to the system.
 - d. A confirmation window will pop up to confirm.
14. Scenario title: *Edit HW*
 - a. The instructor can edit a certain HW for different reasons.
 - b. At the HW page, he can click the “Edit” button.
 - c. He can edit certain words or upload a whole different document if necessary.
 - d. A confirmation window will pop up to confirm the action.

15. Scenario title: *Delete HW*

- a. The instructor can delete a HW for different reasons that may vary.
- b. At the HW page, an icon like a Rubbish Bin will delete the HW from the system.
- c. The students will not be required to do it, but if anyone wants to improve, he is free to work on it.
- d. A confirmation window will pop up to confirm.

16. Scenario title: *Check HW's statistics*

- a. After entering the course page, the instructor can take a look at how the course is progressing
- b. After clicking “Check statistics” button, a page with all information about the students and course will be showed.
- c. The “HW” button will redirect him to the HW page.
- d. The button “Check Statistics” will provide him information about the HWs.
- e. He can see all the submitted HWs by clicking “Submitted” button.
- f. He can see all the graded HWs by clicking “Graded” button.

Administrator scenarios

17. Scenario title: *Add instructor*

- a. After logging in, the administrator is redirected to his dashboard.
- b. The button “Add instructor” adds a new instructor.
- c. The administrator will fill the form with his personal information and career achievements.
- d. The button “Add” will add the new instructor.
- e. A window will pop up for the confirmation.

18. Scenario title: *Edit instructor*

- a. After the administrator is logged in, he can go to one of his instructors that is teaching.
- b. In the instructor page, the button “Edit” provides the opportunity to edit some information or some achievements about that instructor
- c. After making changes, a window will pop up.
- d. At the window, the button “Confirm” will make the changes that the administrator made.
- e. The button “Cancel” will delete all the changes and restore everything to normal.

19. Scenario title: *Delete instructor*

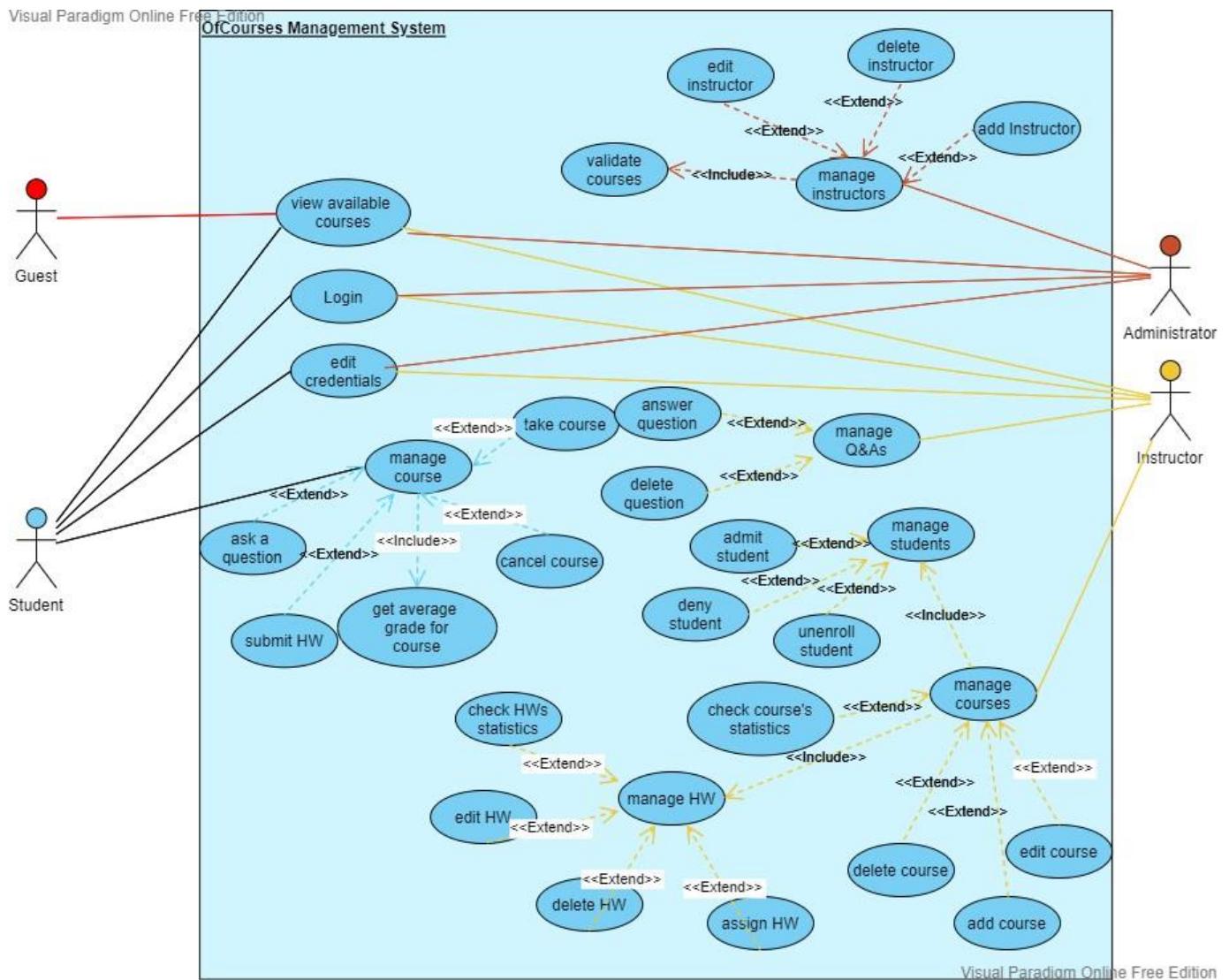
- a. For some certain reasons, the administrator wants to delete the instructor.
- b. At the instructor page, if the button “Delete” is clicked the instructor will be deleted.
- c. Before it deletes, a window will pop up with a confirmation.
- d. If the button “Delete” is clicked, the instructor will be deleted.
- e. If the button “Cancel” is clicked, the instructor will not be deleted.

20. Scenario title: *Validate course*

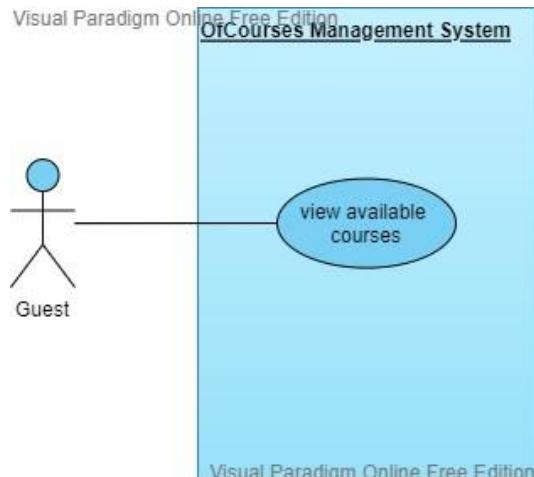
- a. The administrator must control what courses are taught in the system.
- b. When an instructor wants to upload a new course, he must provide the reasons first to the administrator why should it be taught and what things are going to be taught.
- c. The administrator checks the course and if he thinks that is fulfills the requirements, he validates it and allows the instructor to upload it.
- d. If the course does not meet the requirements, the instructor cannot upload it.

4.1.2 Use Cases Diagrams

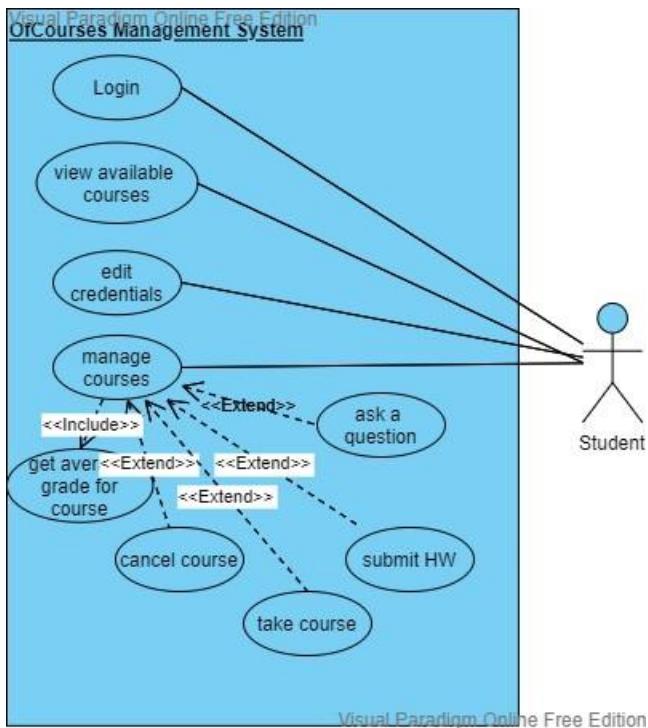
General Use Case



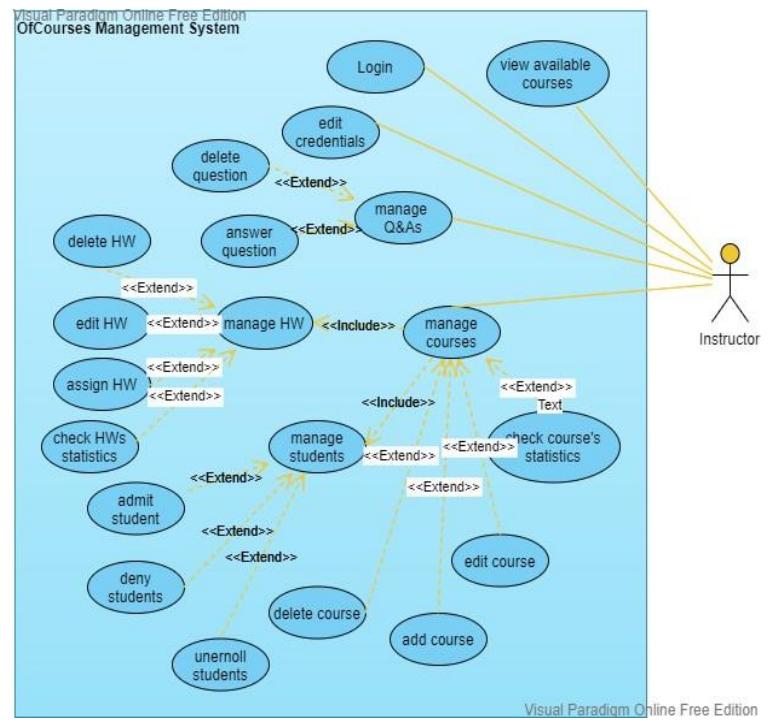
Guest Use Case



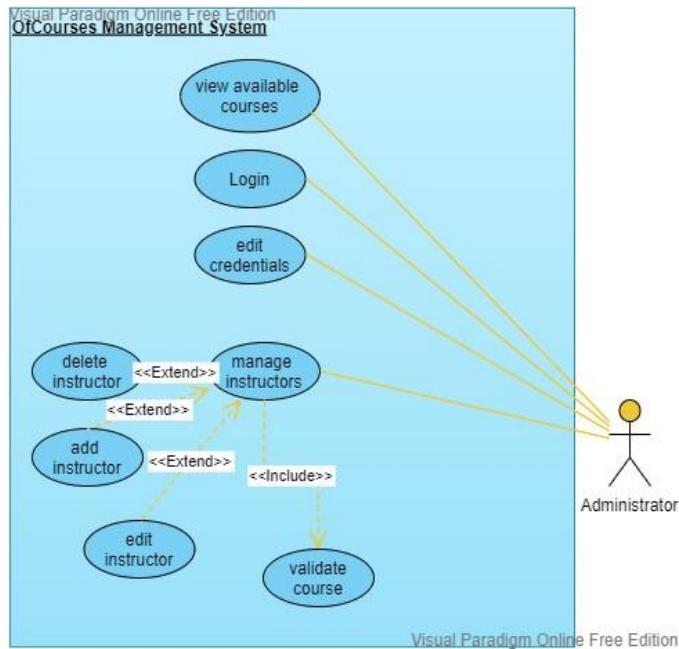
Student Use Case



Instructor Use Case



Administrator Use Case



4.1.3 Use Cases Tables (Extended)

| UC_1.1 | Login |
|--|---|
| Summary | - Each user will be able to login into the system by entering their respective credentials. |
| Dependency | - |
| Actors | - Student, Instructor, Administrator |
| Preconditions | - The user must have a valid account, where the Students can easily sign up, while the Instructors will be created by the Administrator, which will be created in advance, |
| Description of the Main Sequence | - Each user will be able to log in by entering their credentials if they already have a valid account. |
| Description of the Alternate Sequence | - In the other case, where a user does not have a valid account, he (student) can signup by entering some required credentials. - The Administrator will be a default account. - Instructors will be added by the Administrator. |
| Non-functional requirements | - This is a crucial part of the project because this is where the 'hackers' will try to break through, by using SQL injection. It is made sure that this will be avoided. Also, the password will be hashed and cannot be seen by any person. |
| Postconditions | - The user will be directed to his dashboard. |

| | |
|--|--|
| UC_1.2 | Edit Credentials |
| Summary | - Each user will have the opportunity to edit their information details. |
| Dependency | - |
| Actors | - Students, Instructor, Administrator |
| Preconditions | <ul style="list-style-type: none"> - User must have an account - User must be logged in |
| Description of the Main Sequence | - Each user will have the opportunity to change or edit their profile including name, birthday, username, password, and photo (optional) according to the given rules. |
| Description of the Alternate Sequence | - If the newly changed credential does not meet the requirements, then it will not be saved and it will remain the same as before |
| Non-functional requirements | - The password will be hashed and cannot be seen by anyone, except the user. A hash library is necessary. |
| Postconditions | - The information for the user will be updated, |

| | |
|--|---|
| UC_1.3 | View Available Courses |
| Summary | - The user can see which courses will be available for him to take |
| Dependency | - |
| Actors | - Guest, Student, Instructor, Administrator |
| Preconditions | - The user must have established an internet connection - User must have visited OfCourses |
| Description of the Main Sequence | - The user can check the courses that are offered at the moment, and maybe take on if one of these courses fits him |
| Description of the Alternate Sequence | - There might be no available courses at the moment and all the user will see, will be empty, or finished courses. - User might be logged out (Guest), and the only thing he can do is watch |
| Non-functional requirements | - To meet the Usability Requirements, the user will be able to use the product everywhere: PC, phone, laptop, etc. |
| Postconditions | - If one course is selected, then the user is directed to that course page. |

| | |
|--|--|
| UC_2.1 | Manage Courses |
| Summary | - Users can take action on courses they are involved with. |
| Dependency | - |
| Actors | - Students |
| Preconditions | - Internet connection - User must have at least started a course. |
| Description of the Main Sequence | - The user can either get information such as course descriptions or add and drop courses. |
| Description of the Alternate Sequence | - The user is not logged in. - The user is not enrolled in that course. - The course is not available at the moment |
| Non-functional requirements | -The user's personal information will be protected and it will not be visible to the others, except the user, instructor, and administrator. |
| Postconditions | - The user will have the ability to interact with the course. |

| | |
|--|--|
| UC_2.1.1 | Cancel Course |
| Summary | - Users can select a course and choose to drop from it. |
| Dependency | - Manage Courses |
| Actors | - Students |
| Preconditions | <ul style="list-style-type: none"> - Instructors must have added a course. - Students have to be enrolled in that course. |
| Description of the Main Sequence | -The user can unenroll himself from a certain course for different reasons that may vary. |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The user cannot unenroll himself from the course because the add/drop period has ended(depending on the course). - The user has reached his drop courses limit. |
| Non-functional requirements | - |
| Postconditions | - The user will be unenrolled from that course and will not be asked or charged. |

| | |
|--|---|
| UC_2.1.2 | Take Course |
| Summary | - Students can participate in a certain course. |
| Dependency | - Manage Courses |
| Actors | - Students |
| Preconditions | <ul style="list-style-type: none"> - Instructors must have added a course. - Students must be first logged in. |
| Description of the Main Sequence | - Students can select a specific course and choose to enroll in it. |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - Students cannot be enrolled in a certain course because it has reached its quota. - The course is not available at the moment. |
| Non-functional requirements | - |
| Postconditions | - The student is directed to the course page. |

| | |
|--|--|
| UC_2.1.3 | Get an average grade for a course |
| Summary | - Students can get/view a grade for a course that they are enrolled in. |
| Dependency | - Manage Courses |
| Actors | - Students |
| Preconditions | - Student users must be enrolled in at least one course. |
| Description of the Main Sequence | <ul style="list-style-type: none"> - Students can get a grade as a feedback for a certain HW - Students can get an average grade at the end of the course - Students can check grading information such as individual grades for exams, projects, etc., or average grades and other grading properties. |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The student has not been enrolled in any course before. |
| Non-functional requirements | <ul style="list-style-type: none"> - The student's grades will be invisible to others, except himself, the Instructor, and the Administrator. |
| Postconditions | <ul style="list-style-type: none"> - The user is directed to the page with grade information |

| | |
|--|--|
| UC_2.1.4 | Submit HW |
| Summary | - Students can submit a certain HW |
| Dependency | - Manage Courses |
| Actors | - Students |
| Preconditions | <ul style="list-style-type: none"> - Students must be enrolled in at least one course. - The instructor must have assigned an HW |
| Description of the Main Sequence | <ul style="list-style-type: none"> - The instructor assigns an HW - Students after solving it, submit it |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The student is not enrolled in any course yet. - The student has not participated in exams or has not submitted HWs. - There were no exams or Hws at the moment. |
| Non-functional requirements | <ul style="list-style-type: none"> - |
| Postconditions | <ul style="list-style-type: none"> - Users submit their HW and redirect to the dashboard |

| | |
|--|--|
| UC_2.1.5 | Ask a question |
| Summary | - Students can ask certain questions to the instructors |
| Dependency | - Manage Course |
| Actors | - Students |
| Preconditions | <ul style="list-style-type: none"> - Students must be enrolled in at least one course. - Student must have participated in exams or submitted his HWs. |
| Description of the Main Sequence | <ul style="list-style-type: none"> - Students may have questions - Students write their questions to this forum - Instructor answers the question |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The student is not enrolled in any course yet. - The question is a redundant one or a non-related topic |
| Non-functional requirements | <ul style="list-style-type: none"> - The student's questions will be invisible to others(optional) |
| Postconditions | <ul style="list-style-type: none"> - Users are directed to their dashboard |

| | |
|--|---|
| UC_3.1 | Manage Q&As |
| Summary | - Instructors can answer or delete questions from certain students |
| Dependency | - |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - The user must be logged in as an instructor and have started a course. - The instructor must have at least one course ongoing. |
| Description of the Main Sequence | <ul style="list-style-type: none"> - Instructors can answer a certain question - Instructors can delete a certain question |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - There are no questions to be answered |
| Non-functional requirements | - |
| Postconditions | - The instructor is directed to the “Manage Q&As” page |

| | |
|--|--|
| UC_3.1.1 | Answer question |
| Summary | - Instructors can answer questions from students. |
| Dependency | - Manage Q&As |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - The User must be logged in as an instructor. - The Instructor must have started a course. |
| Description of the Main Sequence | <ul style="list-style-type: none"> - A question is asked by a student - The instructor answers the student's question |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The instructor has not started a course. - There are no questions asked. |
| Non-functional requirements | <ul style="list-style-type: none"> - |
| Postconditions | - The instructor is directed to the "Manage Q&As" page |

| | |
|--|--|
| UC_3.1.2 | Delete question |
| Summary | - Instructors can delete questions from students. |
| Dependency | - Manage Q&As |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - The User must be logged in as an instructor. - The Instructor must have started a course. |
| Description of the Main Sequence | <ul style="list-style-type: none"> - A question is asked by a student - The question is inappropriate - The instructor deletes the question |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The instructor has not started a course. - There are no questions asked. |
| Non-functional requirements | <ul style="list-style-type: none"> - |
| Postconditions | <ul style="list-style-type: none"> - The instructor is directed to the “Manage Q&As” page |

| | |
|--|--|
| UC_3.2 | Manage Courses |
| Summary | - The Instructor can manage a course by adding, editing, and deleting. |
| Dependency | - |
| Actors | - Instructor |
| Preconditions | - The instructor must have the course validated by the Administrator. |
| Description of the Main Sequence | - The instructor will have to pass the Administrator test for the course - The instructor can add the course, edit it, or delete it. |
| Description of the Alternate Sequence | - The instructor does not have the validation from the Administrator and he cannot add the course. |
| Non-functional requirements | - |
| Postconditions | - The course is shown in the student's dashboard for selection or selected courses, and the instructor's dashboard as a course that needs management |

| | |
|--|---|
| UC_3.2.1 | Manage Students |
| Summary | - The Instructor will be able to manage the students enrolled in his course. |
| Dependency | - Manage Courses |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - The User must be logged in as an instructor. - The Instructor must have started a course. - The Instructor must have at least one student enrolled. |
| Description of the Main Sequence | <ul style="list-style-type: none"> - The Instructor can admit new students. - The instructor can deny new students. - The instructor can unenroll students. |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - There are no students to manage. |
| Non-functional requirements | <ul style="list-style-type: none"> - |
| Postconditions | - The instructor is redirected to the “Manage Students” page. |

| | |
|--|--|
| UC_3.2.1.1 | Admit Students |
| Summary | - The Instructor will be able to admit the students who want to enroll in his course. |
| Dependency | - Manage Students |
| Actors | - Instructor |
| Preconditions | - The User must be logged in as an instructor. - The Instructor must have started a course. |
| Description of the Main Sequence | - The student enrolls for a course - The instructor of that course will admit the student |
| Description of the Alternate Sequence | - There is no course to enroll in. |
| Non-functional requirements | - |
| Postconditions | - The instructor is redirected to the “Manage Students” page. |

| | |
|--|---|
| UC_3.2.1.2 | Deny Students |
| Summary | - The Instructor will be able to deny the students who want to enroll in his course. |
| Dependency | - Manage Students |
| Actors | - Instructor |
| Preconditions | - The User must be logged in as an instructor. - The Instructor must have started a course. |
| Description of the Main Sequence | - The student enrolls for a course - The instructor of that course will deny the student for different reasons |
| Description of the Alternate Sequence | - There is no course to enroll in. |
| Non-functional requirements | - |
| Postconditions | - The instructor is redirected to the “Manage Students” page. |

| | |
|--|--|
| UC_3.2.1.3 | Unenroll Students |
| Summary | - The Instructor will be able to unenroll the students from his course. |
| Dependency | - Manage Students |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - The User must be logged in as an instructor. - The Instructor must have started a course. - There must be students enrolled in the course |
| Description of the Main Sequence | <ul style="list-style-type: none"> - The student enrolls for a course - The instructor of that course will admit the student - The instructor unenrolls the student for different reasons |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - There is no student to unenroll. |
| Non-functional requirements | <ul style="list-style-type: none"> - |
| Postconditions | - The instructor is redirected to the “Manage Students” page. |

| | |
|--|--|
| UC_3.2.2 | Add Course |
| Summary | - The Instructor can add a course. |
| Dependency | - Manage courses |
| Actors | - Instructors |
| Preconditions | - Instructor must have the course validated by the Administrator |
| Description of the Main Sequence | <ul style="list-style-type: none"> - Instructor will have to pass the Administrator test for the course - Instructor can add the course |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - Instructor does not have the validation from the Administrator and he cannot add the course. - The course is already added. |
| Non-functional requirements | <ul style="list-style-type: none"> - |
| Postconditions | - The course is shown in the student's dashboard for selection or selected courses, and on the instructor's dashboard as a course that needs management. |

| | |
|--|---|
| UC_3.2.3 | Edit Course |
| Summary | - The Instructor can edit a course |
| Dependency | - Manage courses |
| Actors | - Instructors |
| Preconditions | <ul style="list-style-type: none"> - Instructor must have the course validated by the Administrator - Course must be available on OfCourses |
| Description of the Main Sequence | <ul style="list-style-type: none"> - Instructor will have to pass the Administrator test for the course - The course must be available on OfCourses - Instructor will be able to edit information, HWs, etc. in the course |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - Instructor does not have the validation from the Administrator and he cannot add the course. - There is no added course to edit. |
| Non-functional requirements | <ul style="list-style-type: none"> - |
| Postconditions | <ul style="list-style-type: none"> - Course is shown in the student's dashboard for selection, or selected courses and instructor's dashboard as a course that needs management |

| | |
|--|--|
| UC_3.2.4 | Delete Course |
| Summary | - The instructor can delete the course from OfCourses |
| Dependency | - Manage Courses |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - Instructor must have the course validated by the Administrator - Instructor must have the courses posted |
| Description of the Main Sequence | <ul style="list-style-type: none"> - For different reasons, the instructor can delete the course from OfCourses, and all the enrolled students are free and not charged for that course |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The course is not deleted, but the students still will not be charged and assigned anything for that course - The Administrator will delete it manually from the database |
| Non-functional requirements | <ul style="list-style-type: none"> - |
| Postconditions | <ul style="list-style-type: none"> - The course is deleted from the Student's and Administrator's dashboard |

| | |
|--|---|
| UC_3.2.5 | Check Course's Statistics |
| Summary | - The Instructor can check how the students performed in this course. |
| Dependency | - Manage Courses |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - The Course must be validated by the Administrator. - The Course is added in OfCourses. - The Course must have students enrolled. |
| Description of the Main Sequence | <ul style="list-style-type: none"> - The Instructor can take a look at the students' statistics. - The Instructor can take a look at attendance. - The Instructor can see how the students performed in exams and HWs. |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The course has no students enrolled. - There are no exams, HWs, or attended lessons. |
| Non-functional requirements | <ul style="list-style-type: none"> - |
| Postconditions | - The Instructor is informed about his course progress. |

| | |
|--|--|
| UC_3.2.6 | Manage HWs |
| Summary | - The Instructor can do certain actions regarding homework. |
| Dependency | - Manage HWs |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - The user must be logged in as an instructor and have started a course. - The instructor must have at least one course ongoing |
| Description of the Main Sequence | - Instructors can assign, edit, delete homework or check homework statistics for each student. |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - There are no HWs to manage because they are not assigned. - The instructor has not started the course yet(or not at all). |
| Non-functional requirements | - |
| Postconditions | - The instructor is directed to the “Manage homework” page |

| | |
|--|--|
| UC_3.2.6.1 | Assign HWs |
| Summary | - Instructors can assign homework to students with an (optional) deadline. |
| Dependency | - Manage HW |
| Actors | - Instructor |
| Preconditions | - The User must be logged in as an instructor. - The Instructor must have started a course. |
| Description of the Main Sequence | - The instructor can assign homework to students with a deadline, and respond to their questions in the comments if there are any. |
| Description of the Alternate Sequence | - The instructor has not started a course. - The course of the Instructor has not been validated by the Administrator. |
| Non-functional requirements | - |
| Postconditions | - The instructor's page will be updated and students will have a new assignment. |

| | |
|--|---|
| UC_3.2.6.2 | Delete HWs |
| Summary | - Instructors can delete a certain HW for various reasons. |
| Dependency | - Manage HWs |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - The User must be logged in as an instructor. - The Instructor must have started a course. - The Instructor must have assigned an HW. |
| Description of the Main Sequence | <ul style="list-style-type: none"> - The Instructor can select an HW, and for different reasons delete it. - The Student will not be asked to submit that assignment. - The Student will not be graded for that assignment |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The instructor has not started a course. - The course of the Instructor has not been validated by the Administrator. - The HW deadline has been passed. |
| Non-functional requirements | - |
| Postconditions | - The instructor's page will be updated and students will have one less assignment |

| | |
|--|--|
| UC_3.2.6.3 | Edit HWs |
| Summary | - Instructors can make changes regarding homework. |
| Dependency | - Manage HWs |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - The User must be logged in as an instructor. - The Instructor must have started a course. - The Instructor must have assigned an HW |
| Description of the Main Sequence | - Instructors can select an HW, then make certain changes such as changing the HW description, adding or deleting files, or changing the HW deadline. |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The instructor has not started a course. - The course of the Instructor has not been validated by the Administrator. - The HW deadline has been passed |
| Non-functional requirements | - |
| Postconditions | - The instructor's page will be updated and students will have an updated assignment |

| | |
|--|---|
| UC_3.2.6 | Check HWs statistics |
| Summary | - The Instructor will be able to look at the statistics from the HWs. |
| Dependency | - Manage HWs |
| Actors | - Instructor |
| Preconditions | <ul style="list-style-type: none"> - The User must be logged in as an instructor. - The Instructor must have started a course. - The Instructor must have assigned an HW. - The Students must have submitted HW |
| Description of the Main Sequence | <ul style="list-style-type: none"> - The Instructor can see how many of the students have submitted the HW. - The Instructor can see the average grade of the class. - The Instructor can see each student's grade |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - There are no HWs to be graded. - There are no submitted HWs. |
| Non-functional requirements | - |
| Postconditions | - The instructor can see an overview of the HWs |

| | |
|--|--|
| UC_4.1 | Manage Instructors |
| Summary | - The Administrator can manage the Instructors(add, edit, delete). |
| Dependency | - |
| Actors | - Administrator |
| Preconditions | - Must be logged in as an Administrator. |
| Description of the Main Sequence | <ul style="list-style-type: none"> - The Administrator can add new instructors to the course manually. - The Administrator can edit an instructor's personal information or Course mistakes. - The Administrator can delete an Instructor. - The Administrator can apply disciplines to an Instructor. |
| Description of the Alternate Sequence | - |
| Non-functional requirements | - |
| Postconditions | - The Instructors and their Courses will be modified according to the changes that will be made. |

| | |
|--|---|
| UC_4.1.1 | Validate Courses |
| Summary | - The Administrator decides if this course fulfills the requirements. |
| Dependency | - Manage Instructors |
| Actors | - Administrator |
| Preconditions | <ul style="list-style-type: none"> - Must be logged in as an Administrator. - There must be a given course to be examined |
| Description of the Main Sequence | <ul style="list-style-type: none"> - The Administrator reviews the course that the Instructor provides. - If the course meets the requirements, then the Administrator validates it. |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - The course does not meet the given requirements, and feedback is given to the Instructor. - The Instructor works on the course, or he has the choice to not teach it at all. |
| Non-functional requirements | - |
| Postconditions | - The new course will be available for the Students. |

| | |
|--|---|
| UC_4.1.2 | Edit Instructor |
| Summary | - The Administrator can edit an Instructor. |
| Dependency | - Manage Instructors |
| Actors | - Administrator |
| Preconditions | <ul style="list-style-type: none"> - Must be logged in as Administrator. - Must have at least one Instructor. |
| Description of the Main Sequence | <ul style="list-style-type: none"> - The Administrator can edit personal information about the Instructor. - The Administrator can edit course validation for the Instructor. |
| Description of the Alternate Sequence | <ul style="list-style-type: none"> - There are no Instructors to be edited. |
| Non-functional requirements | <ul style="list-style-type: none"> - |
| Postconditions | - The information about the Instructor is updated |

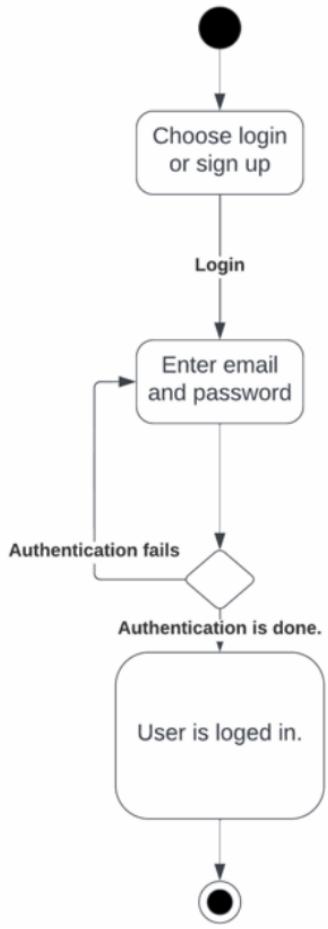
| | |
|--|--|
| UC_4.1.3 | Add instructor |
| Summary | - The Administrator can add an Instructor. |
| Dependency | - Manage Instructors |
| Actors | - Administrators |
| Preconditions | - The user must be logged in as an Administrator. |
| Description of the Main Sequence | - The Administrator can add a new Instructor. |
| Description of the Alternate Sequence | - The new Instructor does not meet the requirements. |
| Non-functional requirements | - |
| Postconditions | - A new Instructor is added |

| | |
|--|--|
| UC_4.1.4 | Delete Instructor |
| Summary | - The Administrator can delete an Instructor. |
| Dependency | - Manage Instructors |
| Actors | - Administrator |
| Preconditions | <ul style="list-style-type: none"> - The user must be logged in as an Administrator. - It must be at least one Instructor. |
| Description of the Main Sequence | - The Administrator can delete an Instructor for certain reasons |
| Description of the Alternate Sequence | - The Instructor cannot be deleted, and the Administrator must do that manually |
| Non-functional requirements | - |
| Postconditions | - The Instructor is deleted from OfCourses. |

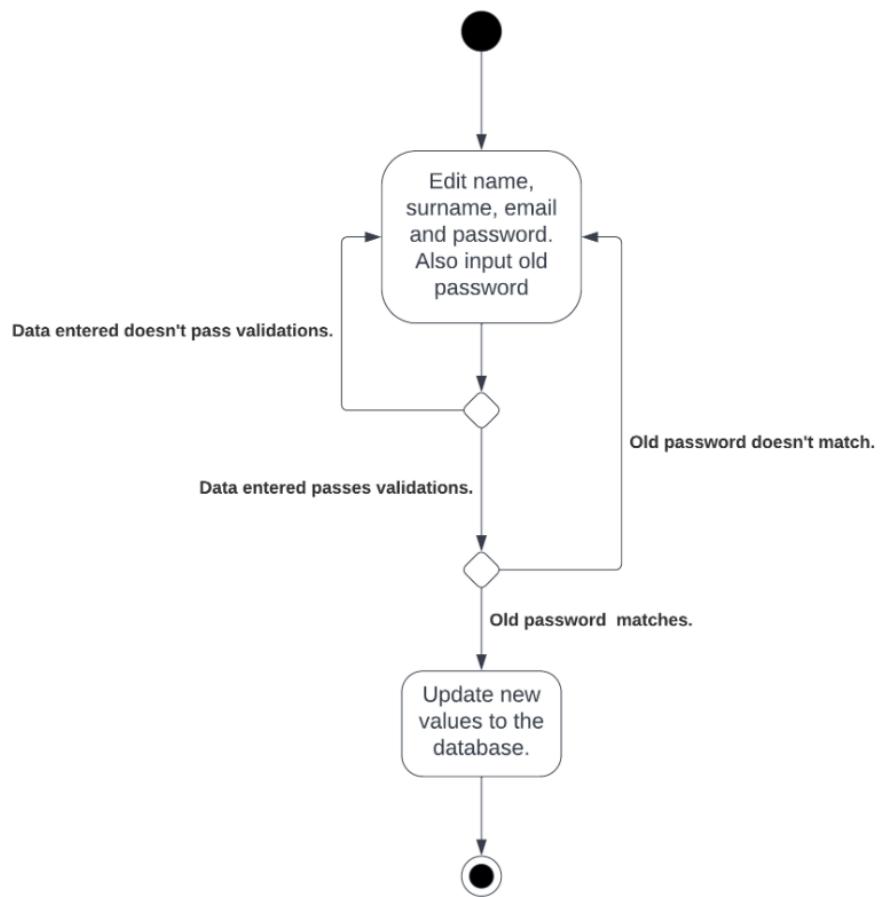
4.2 Diagrams

4.2.1 Activity Diagrams

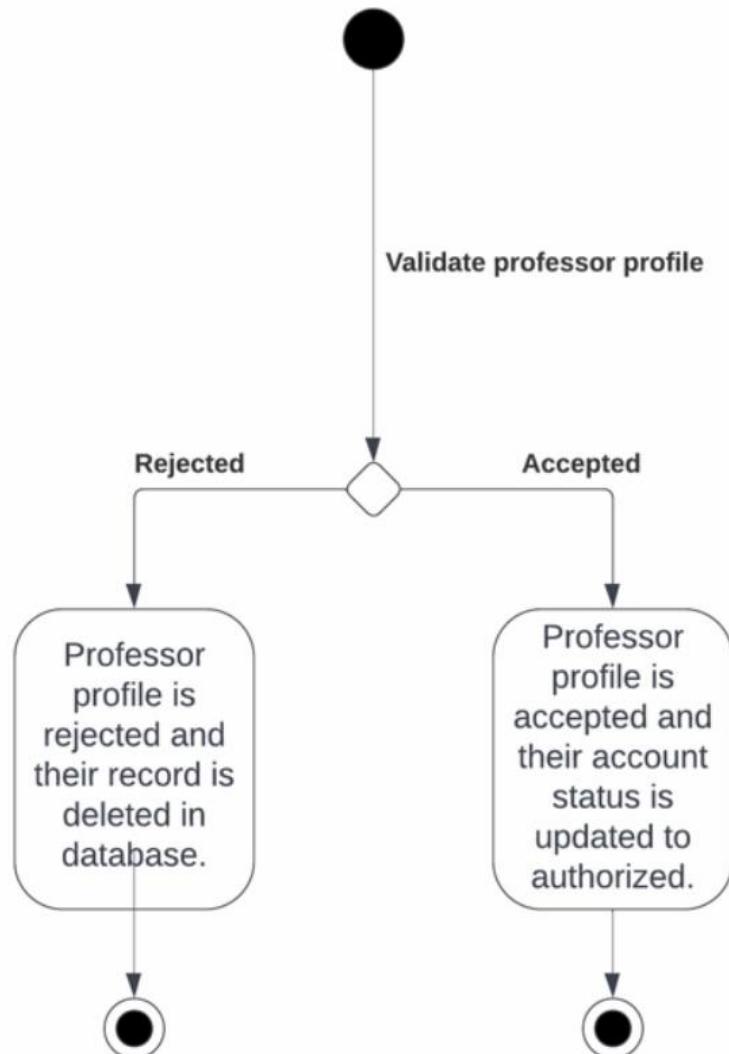
AC_1.1 Login



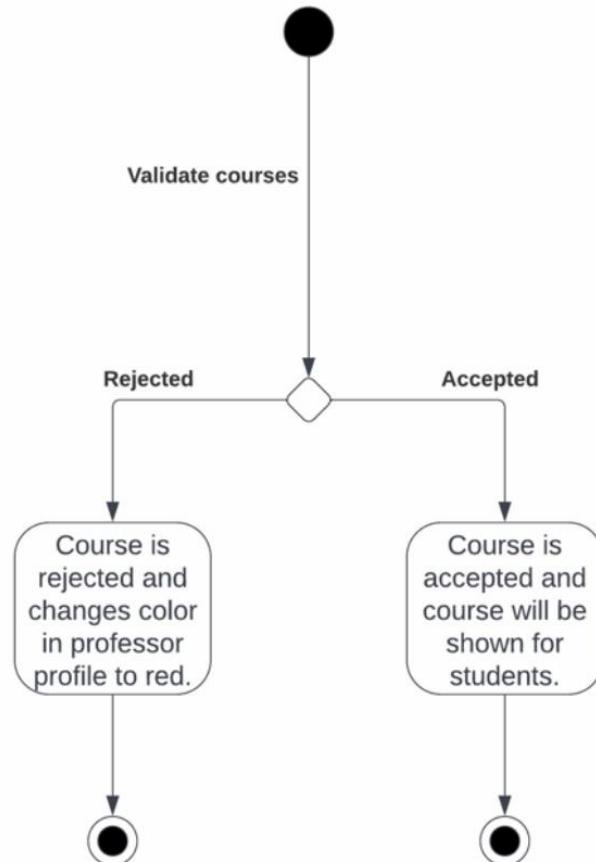
AC_1.2 Edit user profile



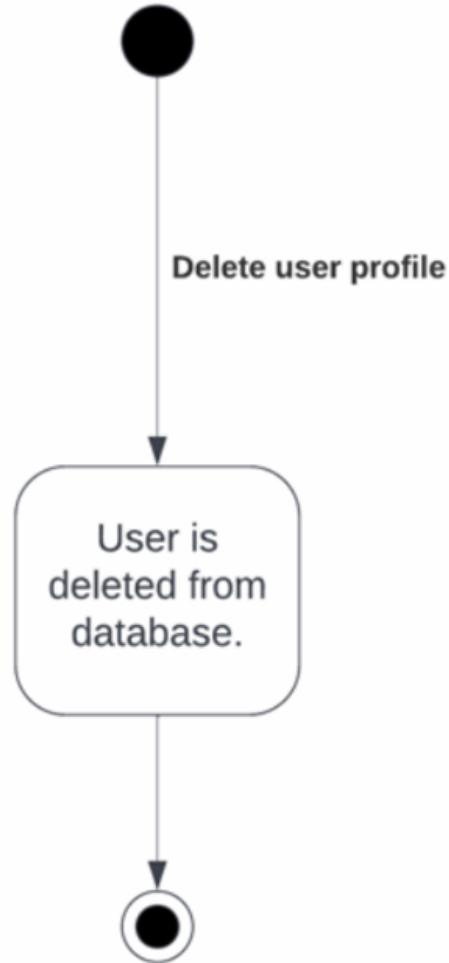
AC_1.3 Validate professor profile



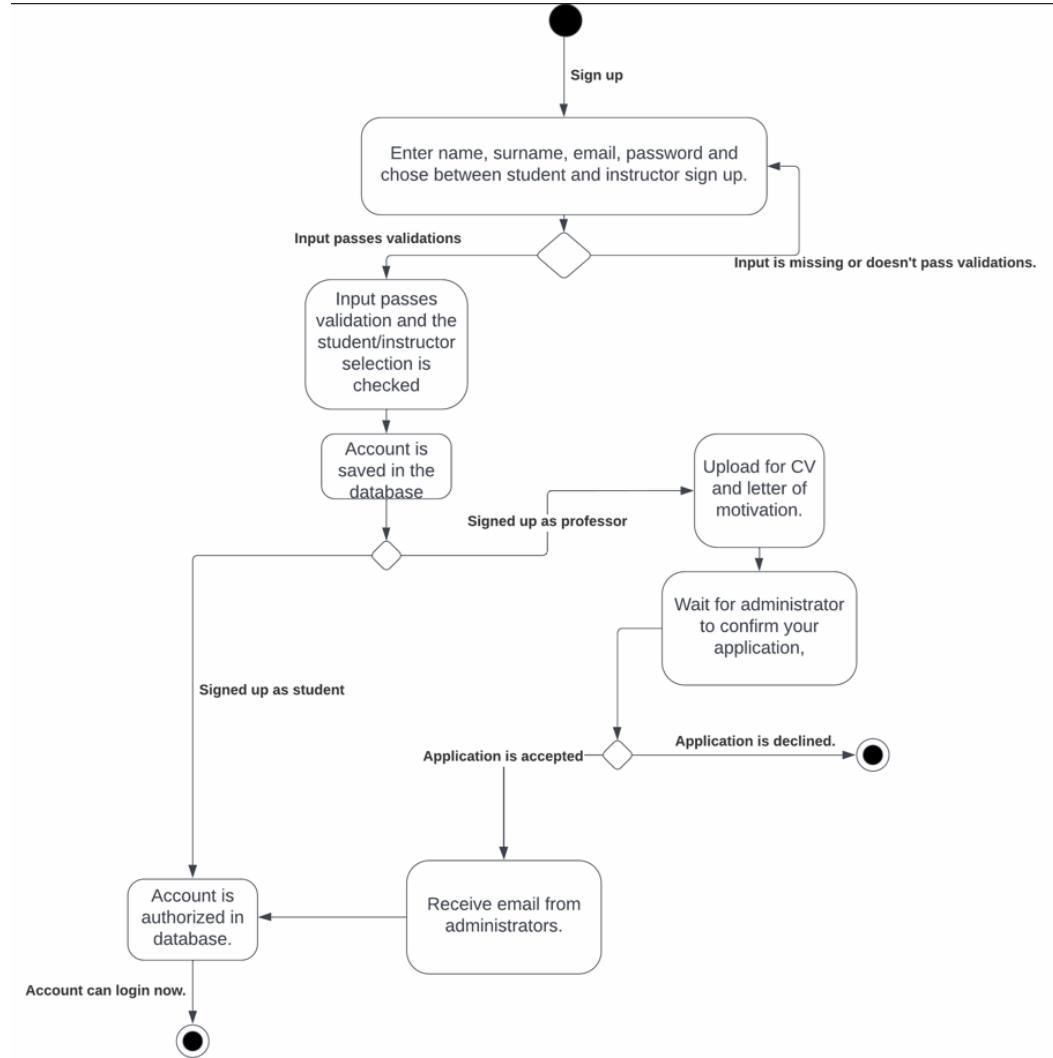
AC_1.4 Validate Courses



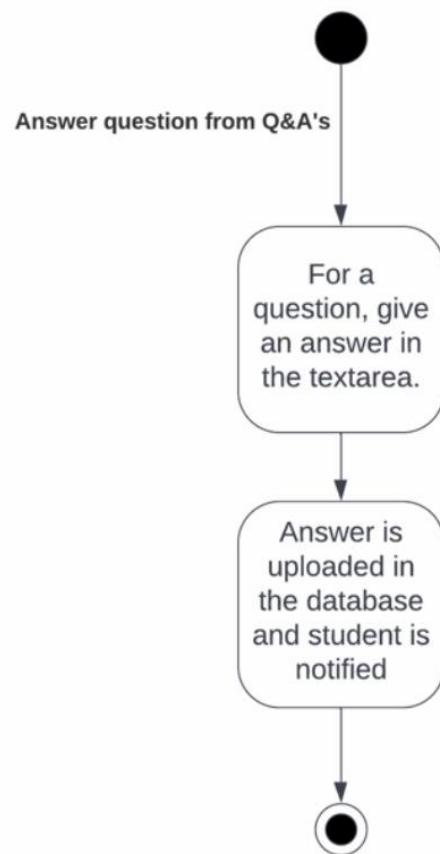
AC_1.5 Delete User Profile



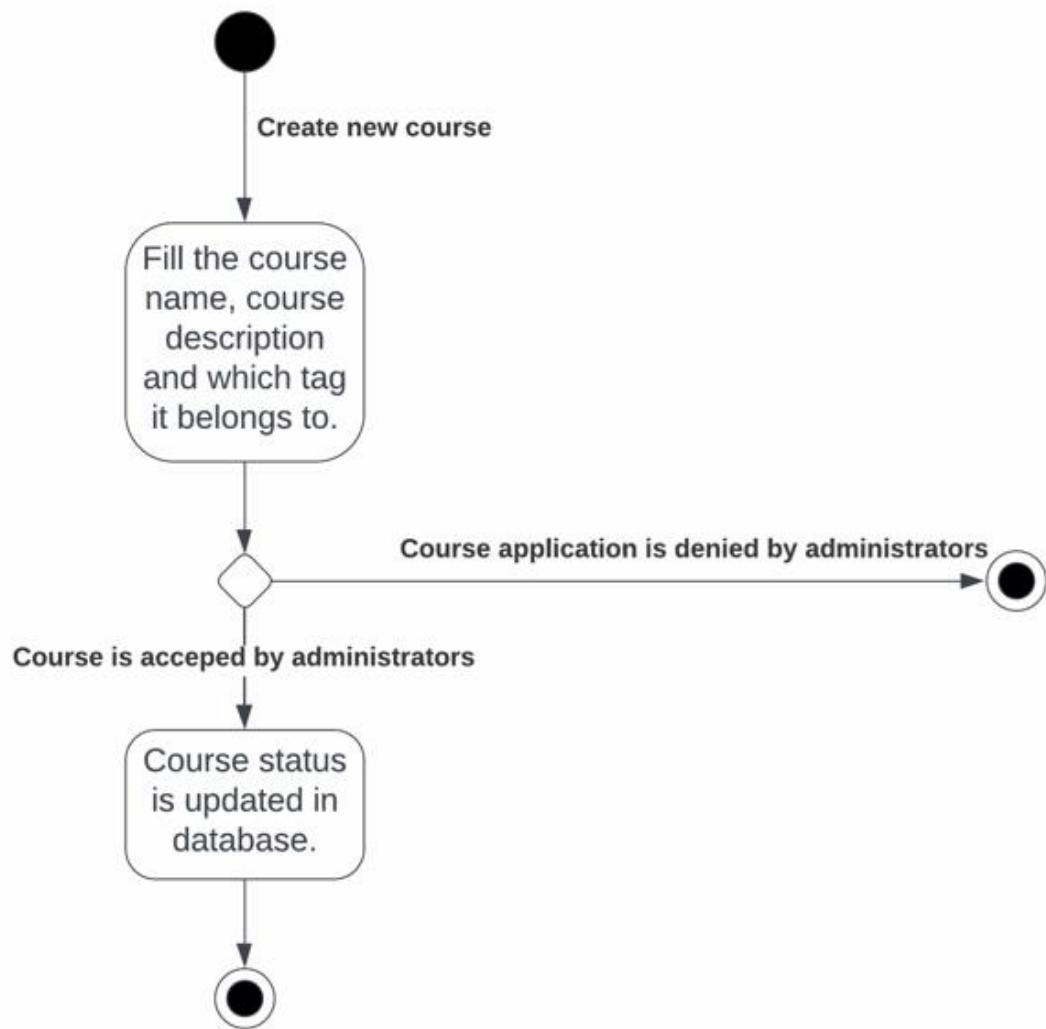
AC_1.6 Sign up



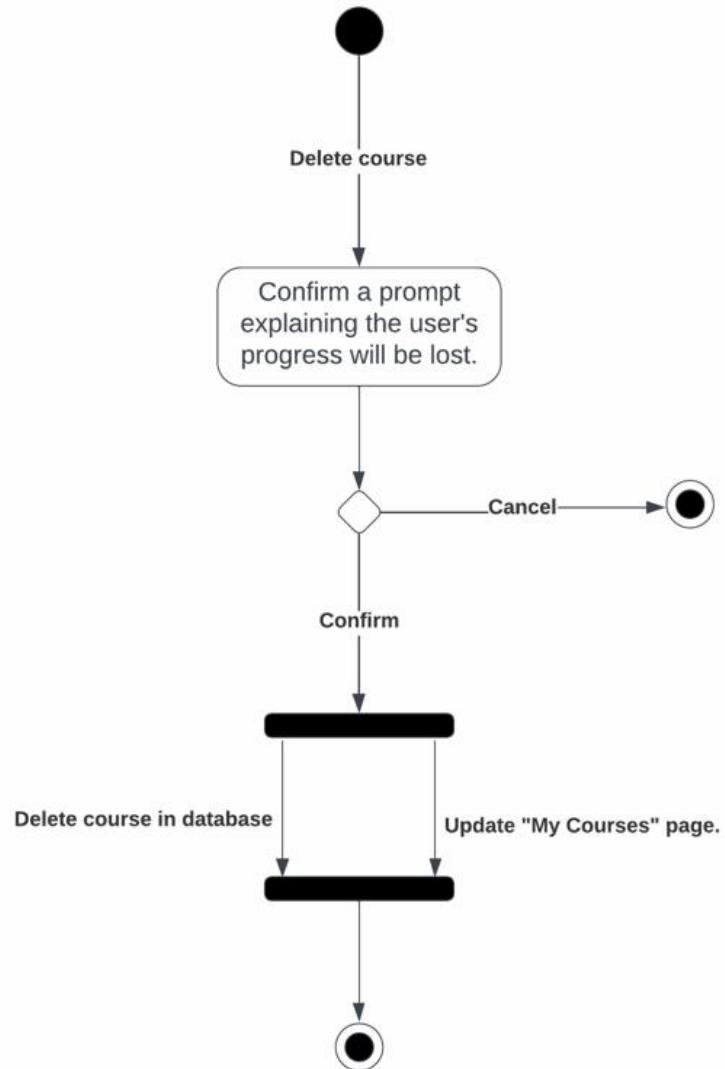
AC_1.7 Answer questions



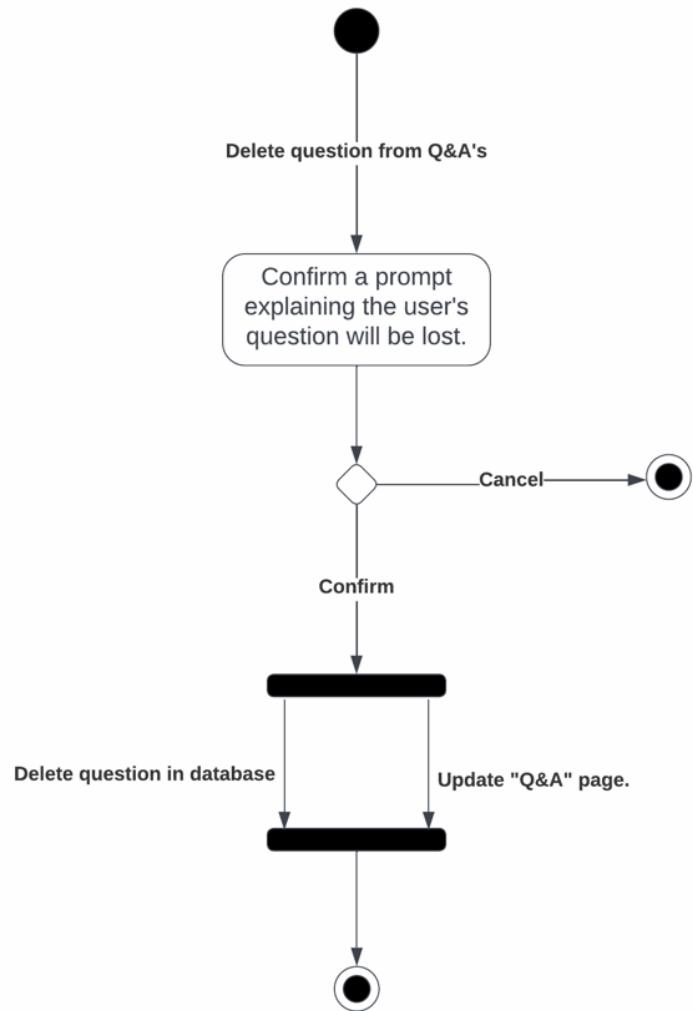
AC_1.8 Create Course



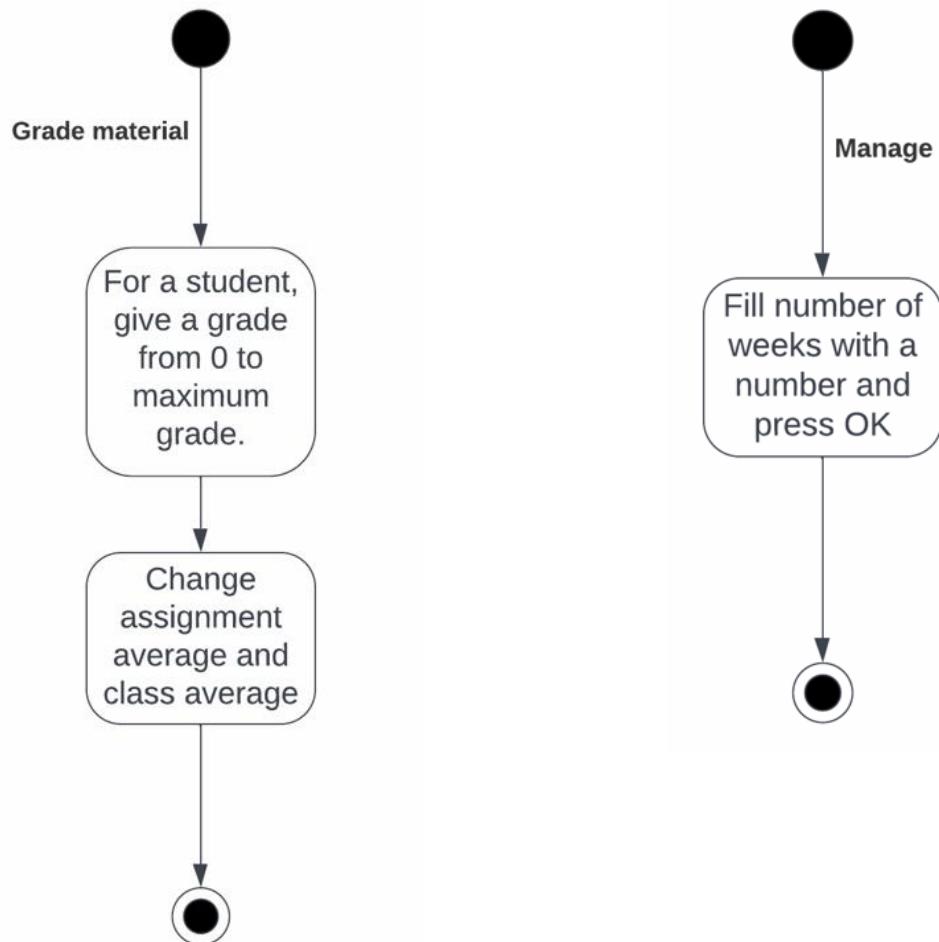
AC_1.9 Delete Course



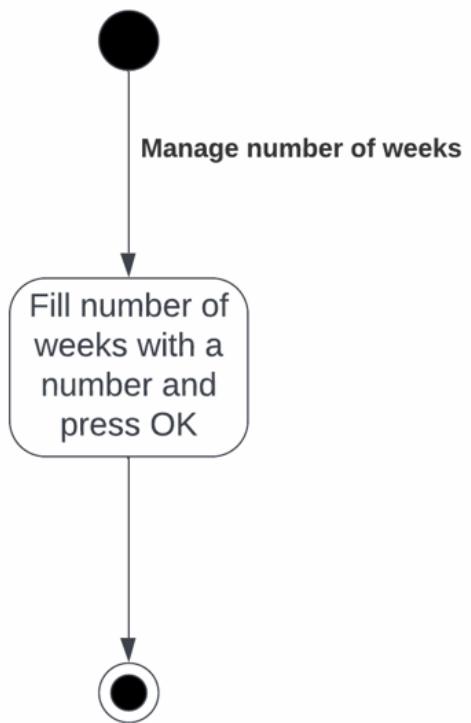
AC_1.10 Delete Question



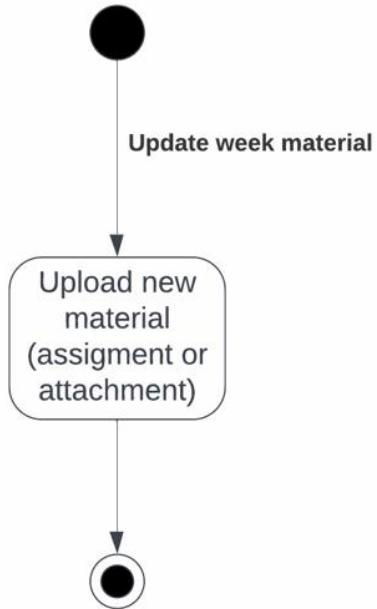
AC_1.11 Grade Assignment



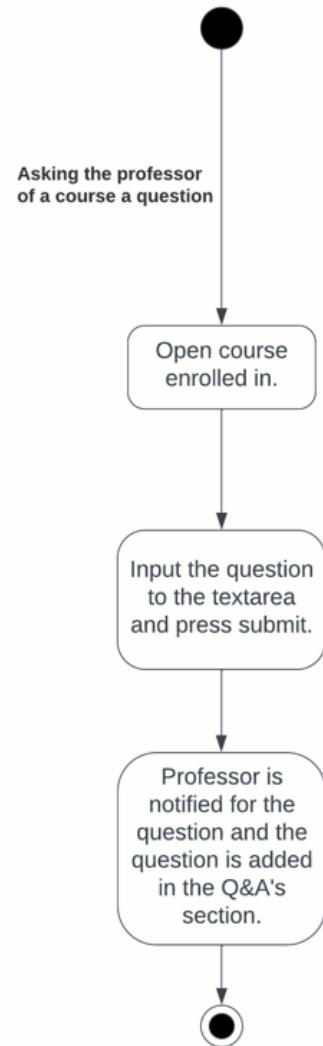
AC_1.12 Manage Number of Weeks



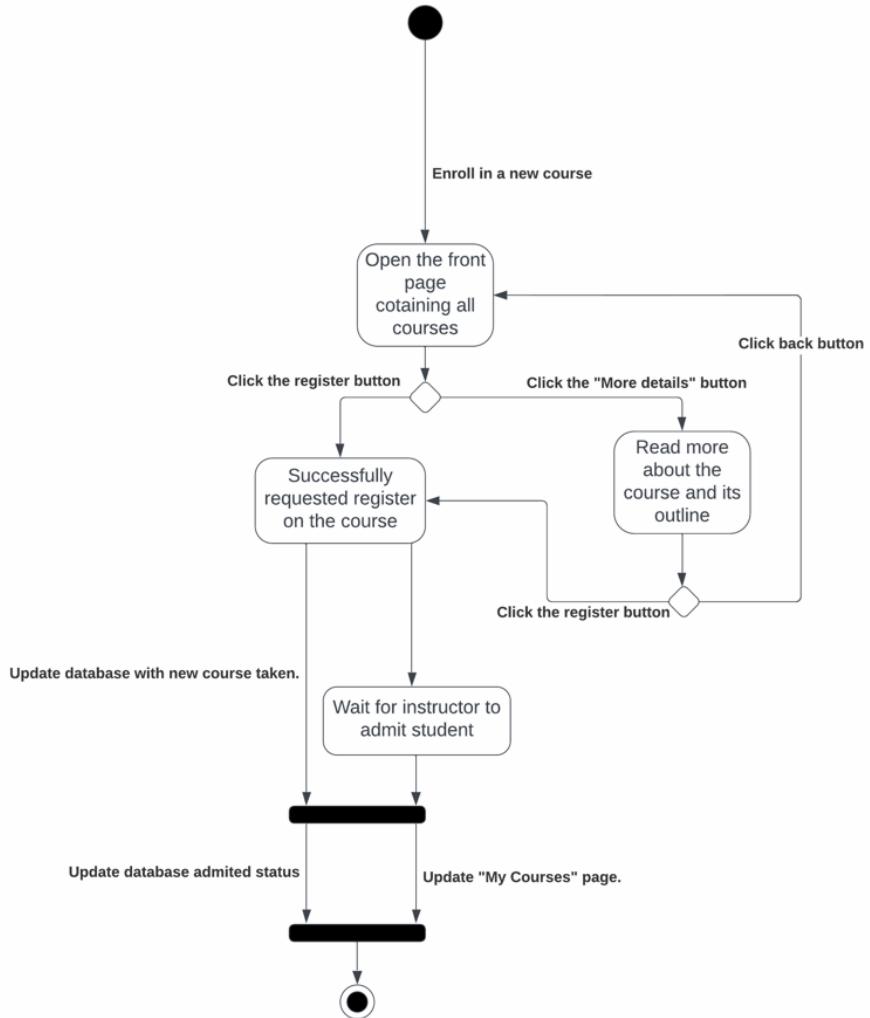
Ac_1.13 Update Week Material



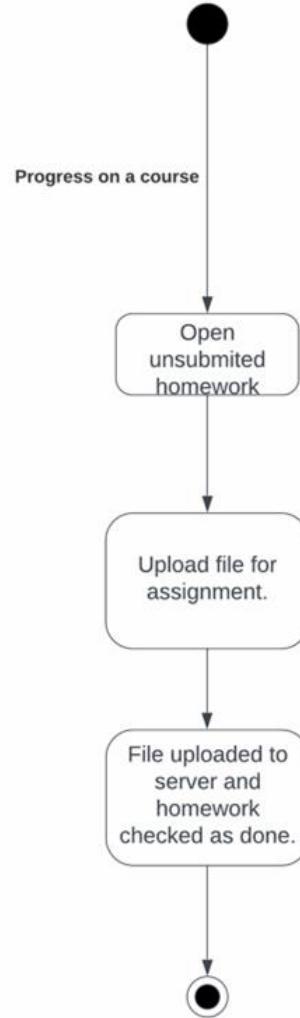
AC_1.14 Ask a Question



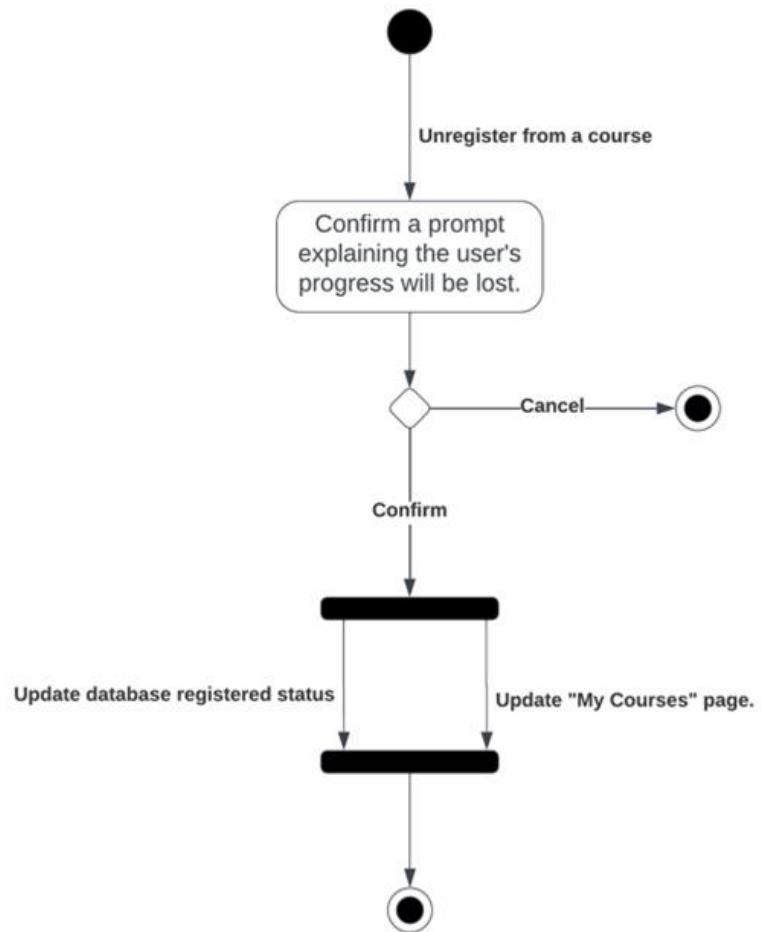
AC_1.15 Enroll in new course



AC_1.16 Submit homework

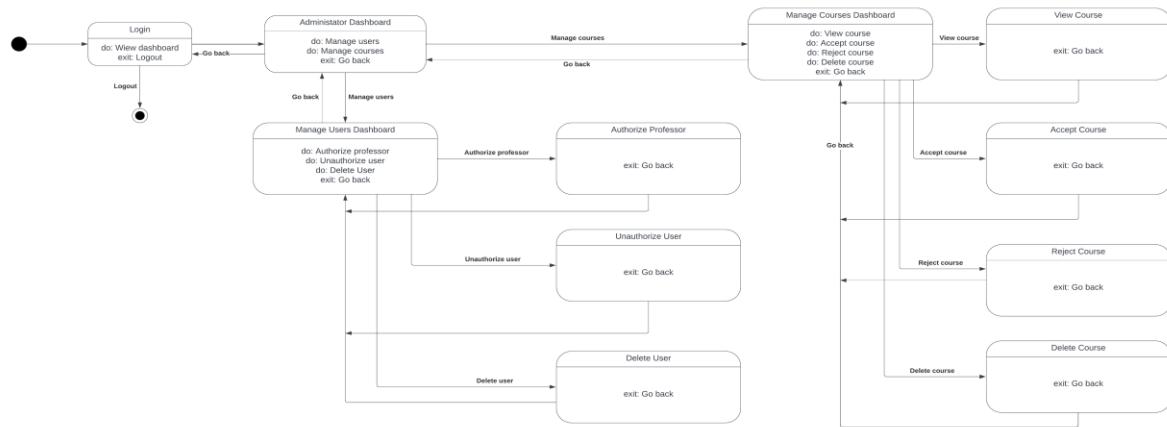


AC_1.17 Unregister from a course

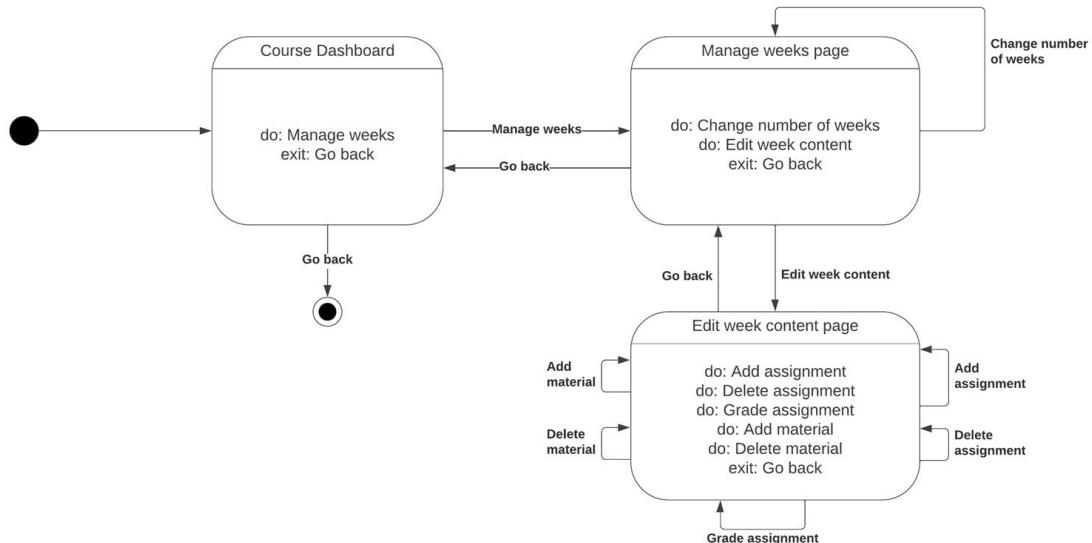


4.2.2 State Diagrams

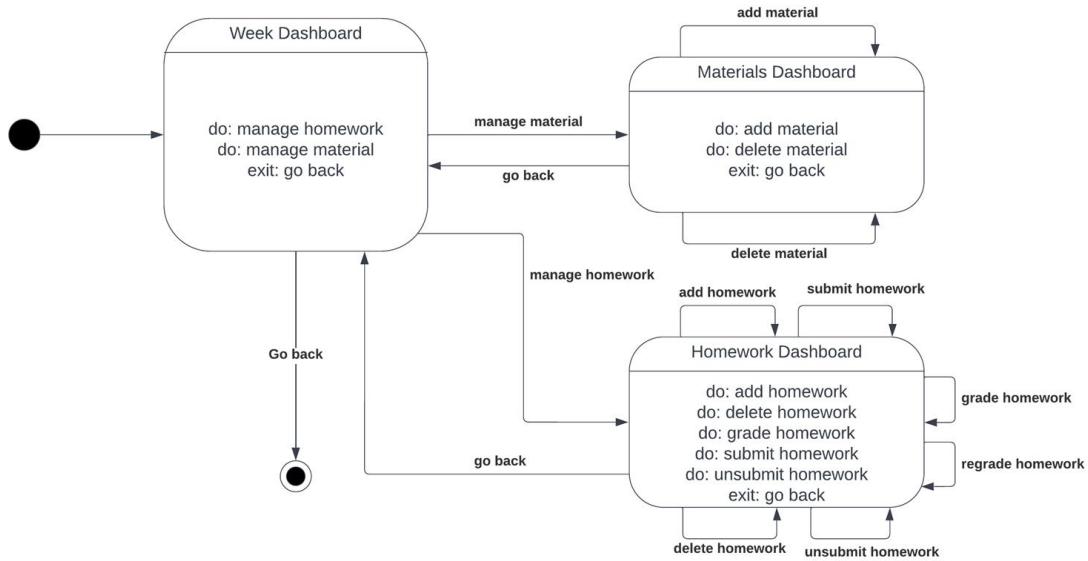
ST_1.1 Administrator



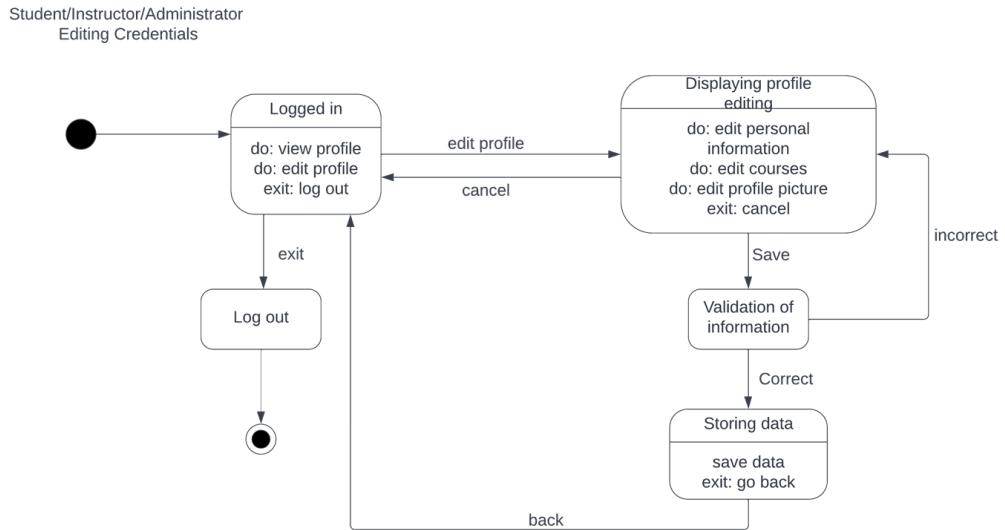
ST_1.2 Course



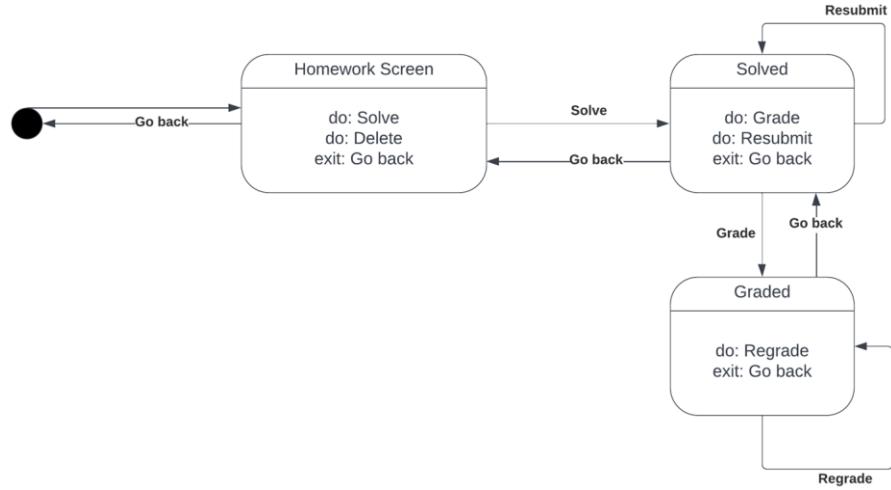
ST_1.3 Course Week



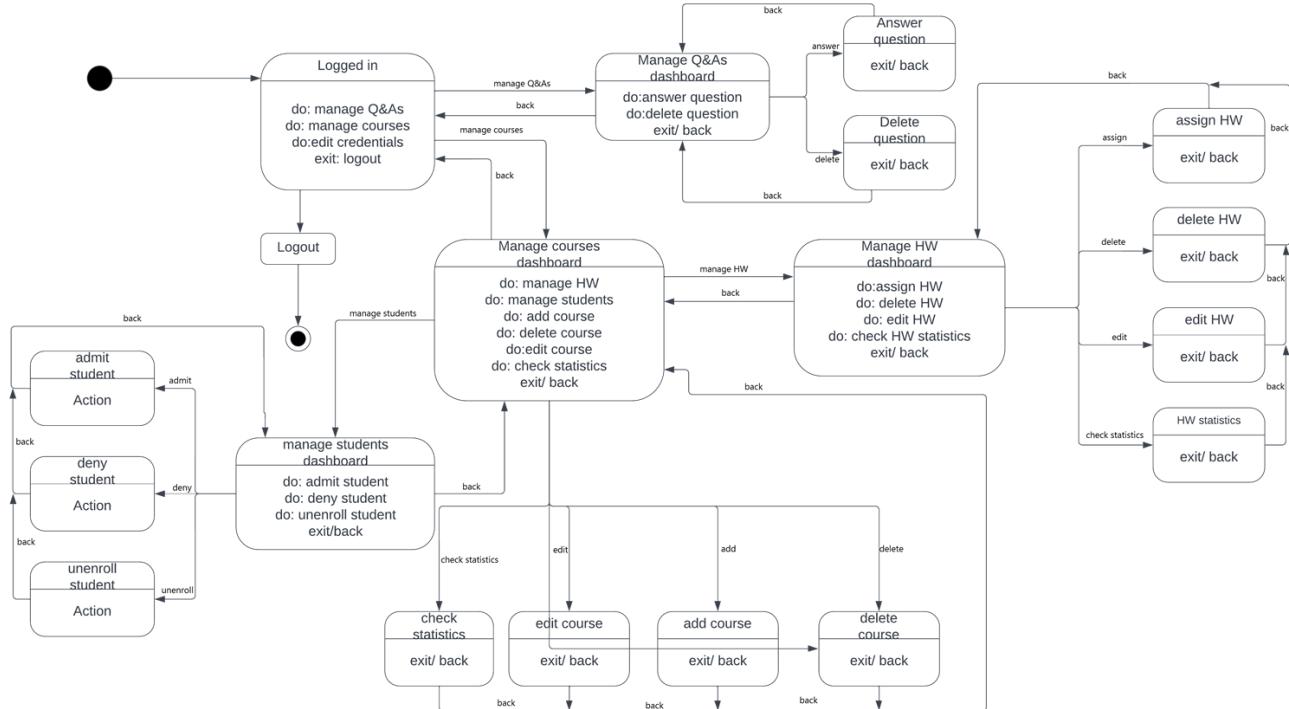
ST_1.4 Edit Credentials



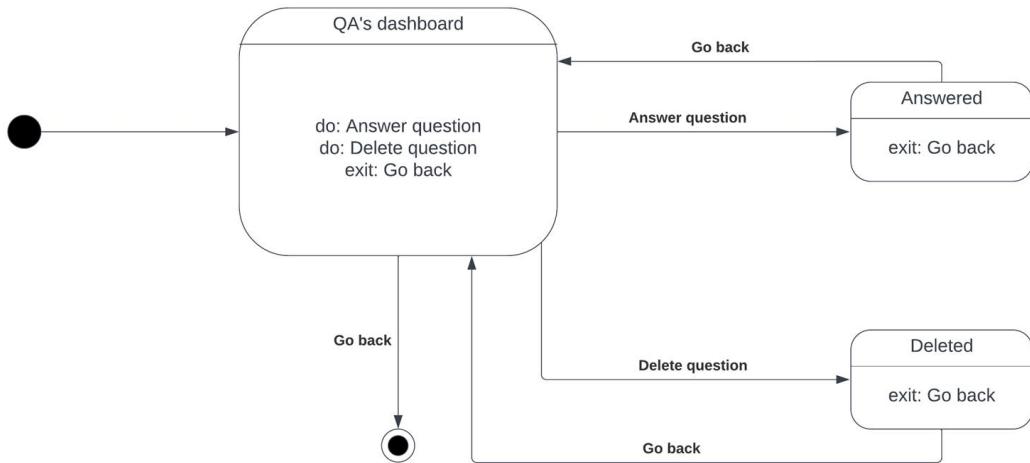
ST_1.5 Homework



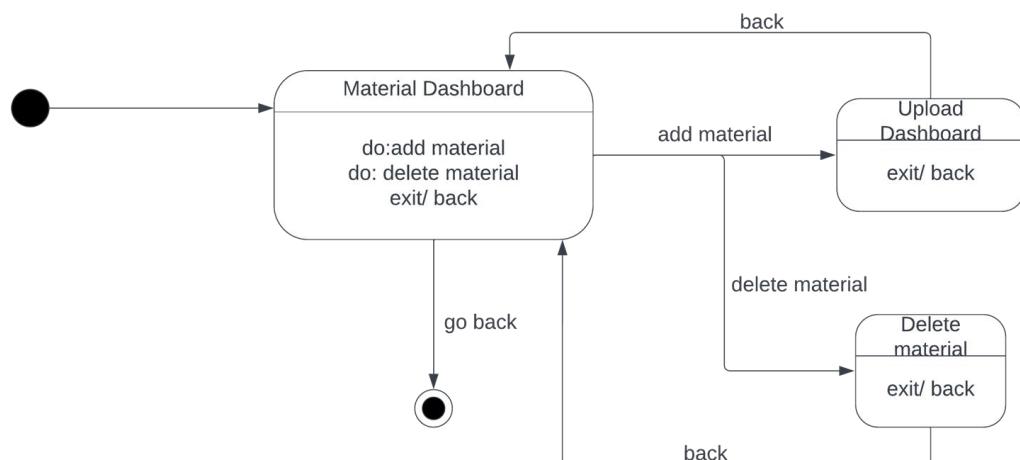
ST_1.6 Instructor



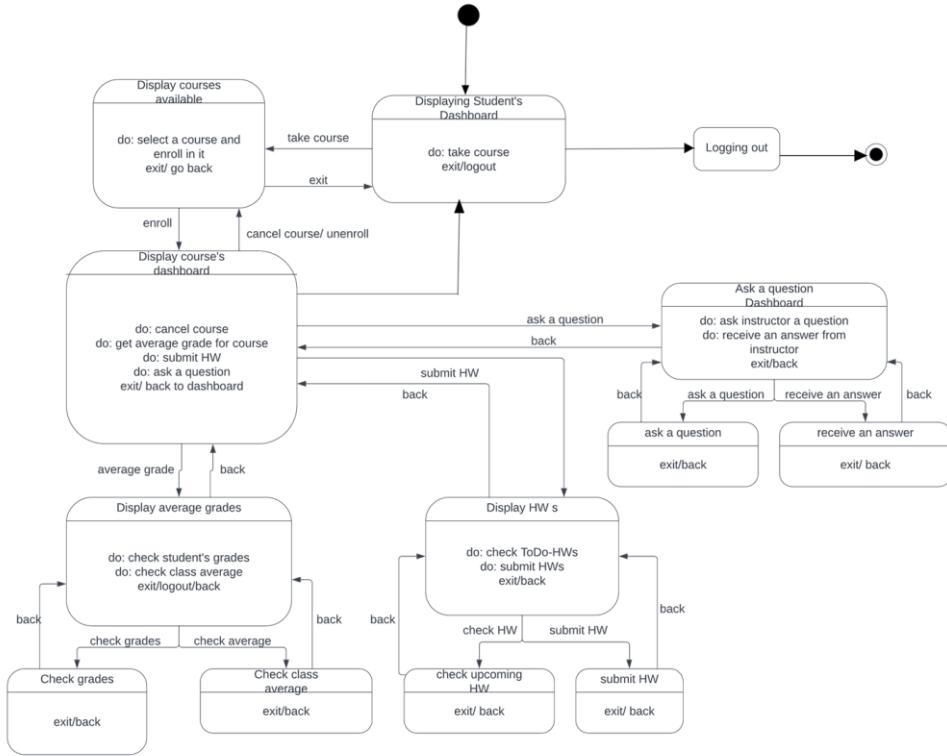
ST_1.7 Q&A



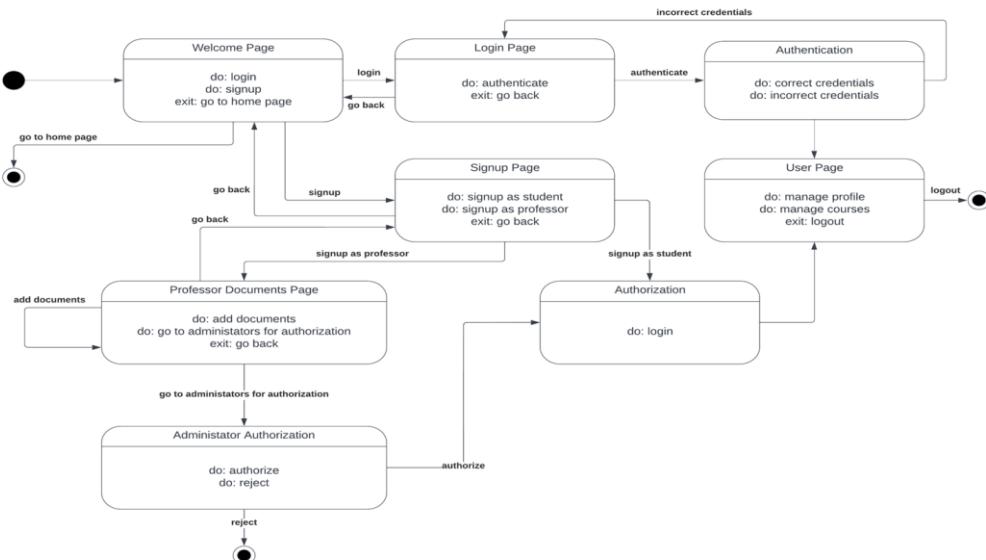
ST_1.8 Materials



ST_1.9 Student Manage Courses

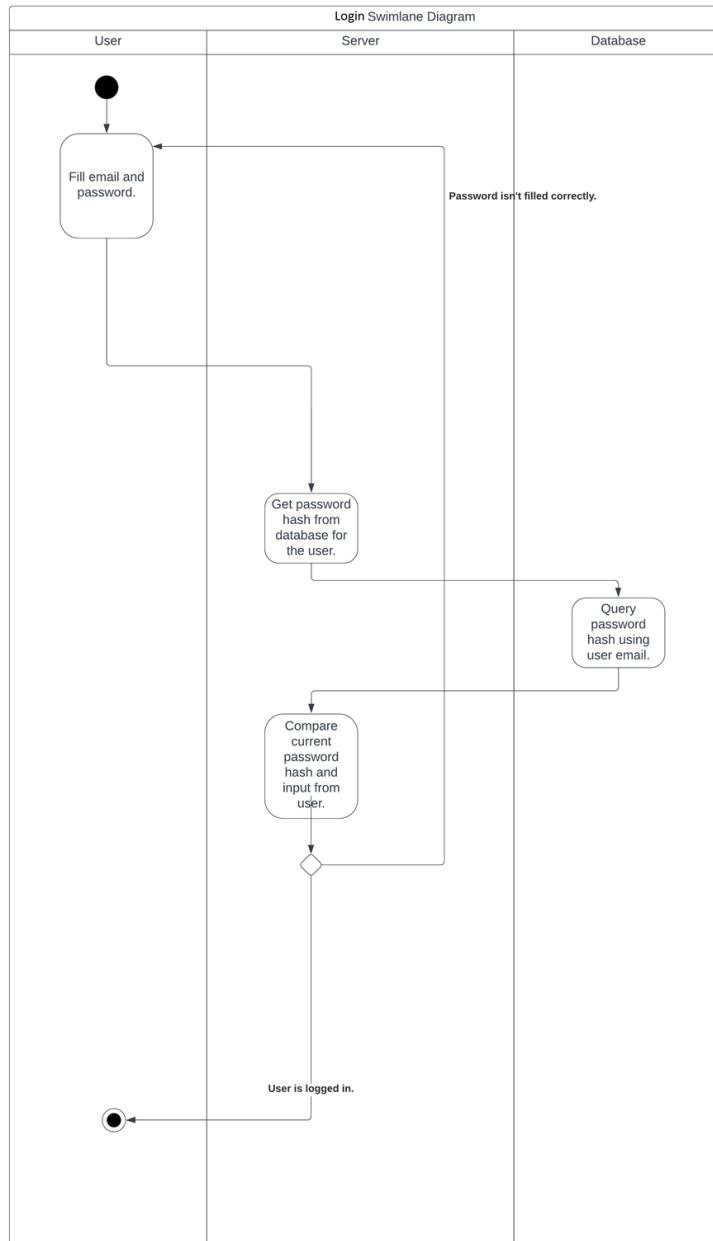


ST_1.10 User Account

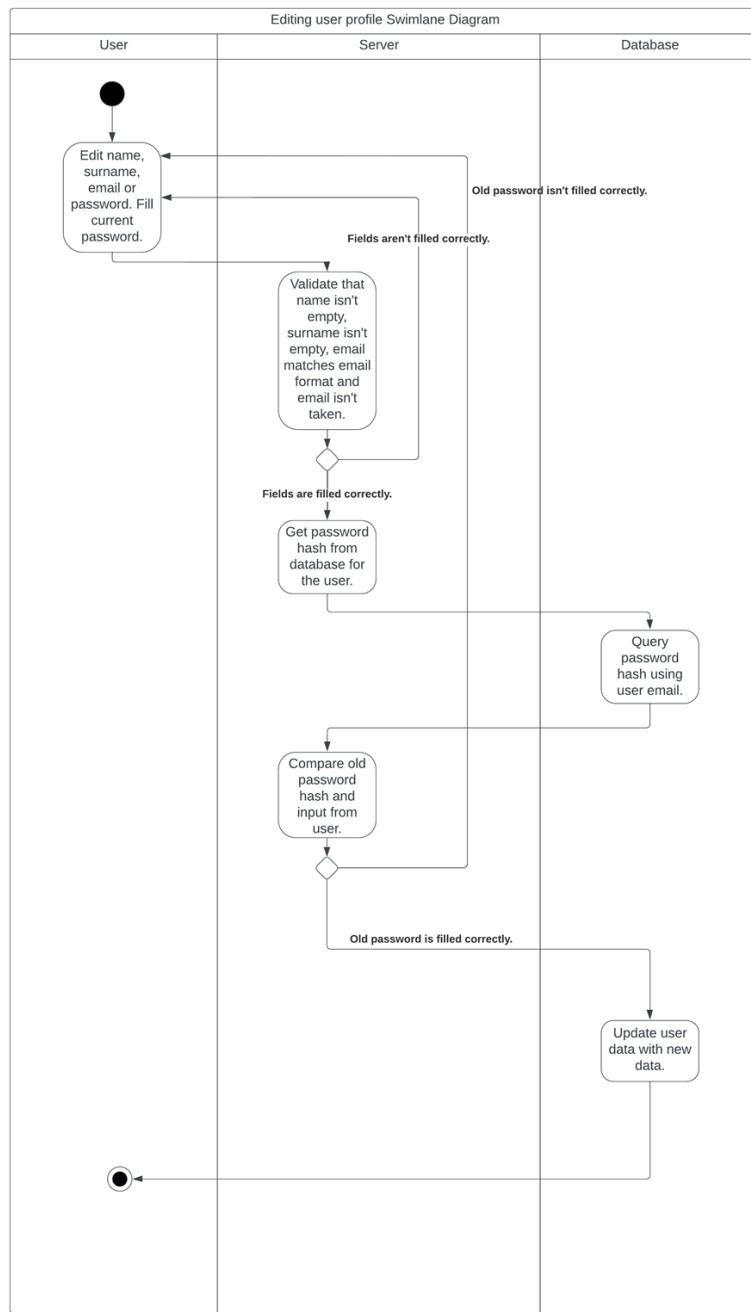


4.2.3 Swimlane Diagrams

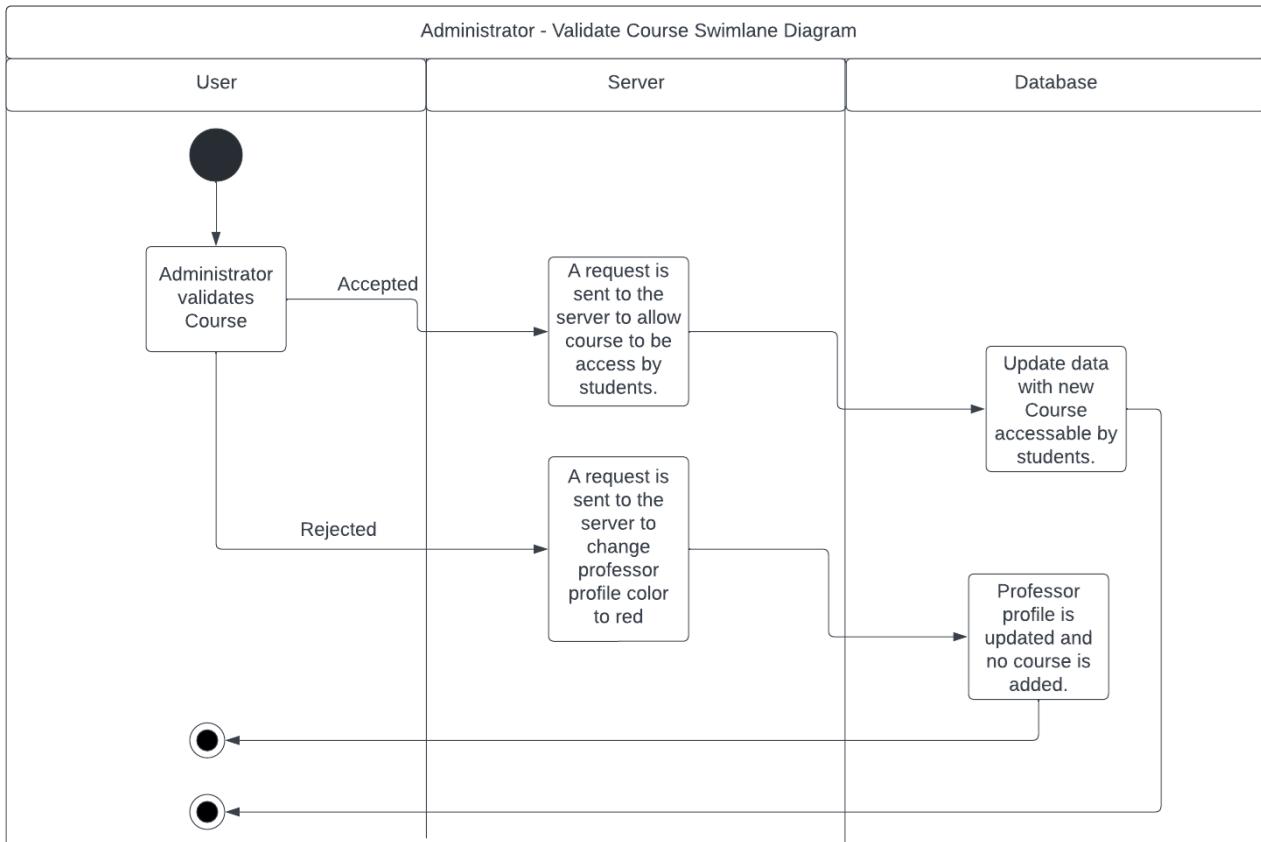
SW_1.1 Login



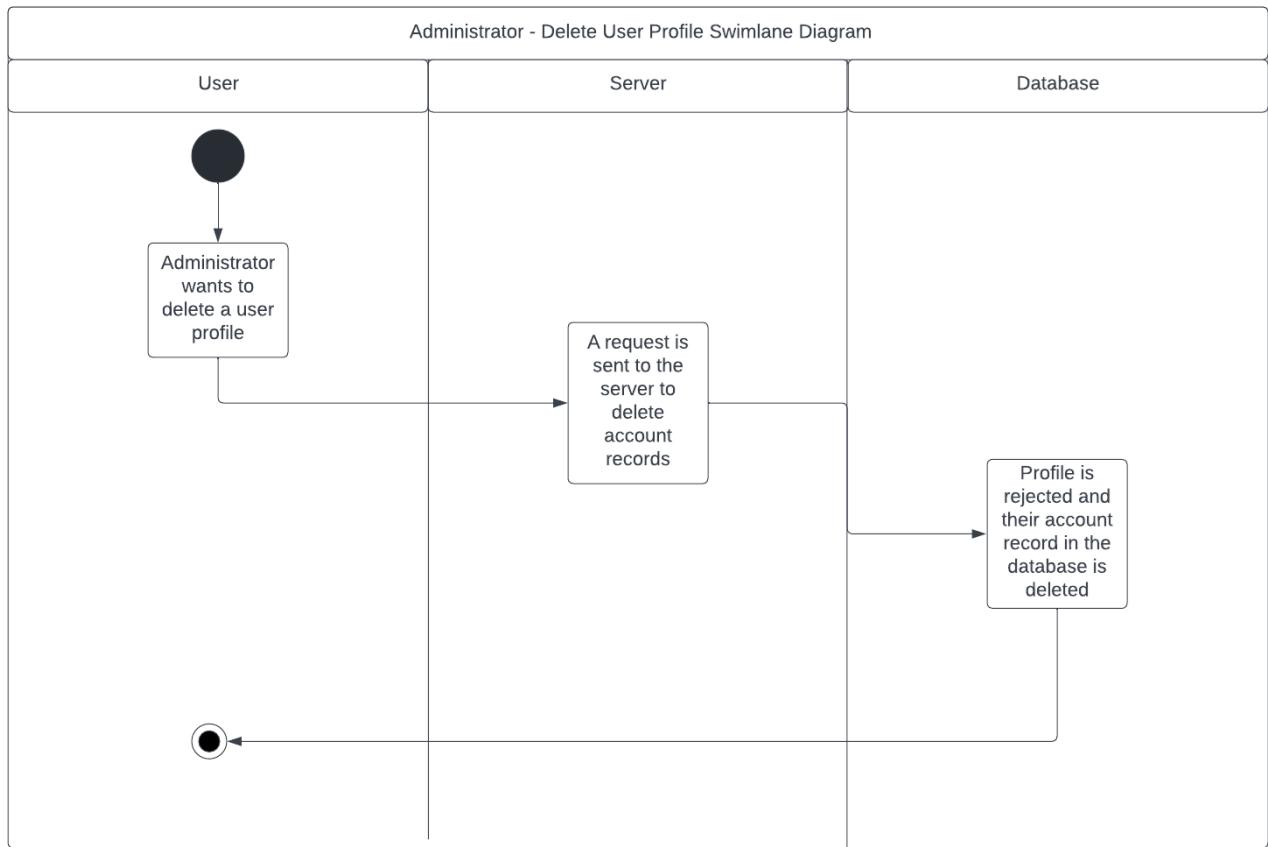
SW_1.2 Edit User Profile



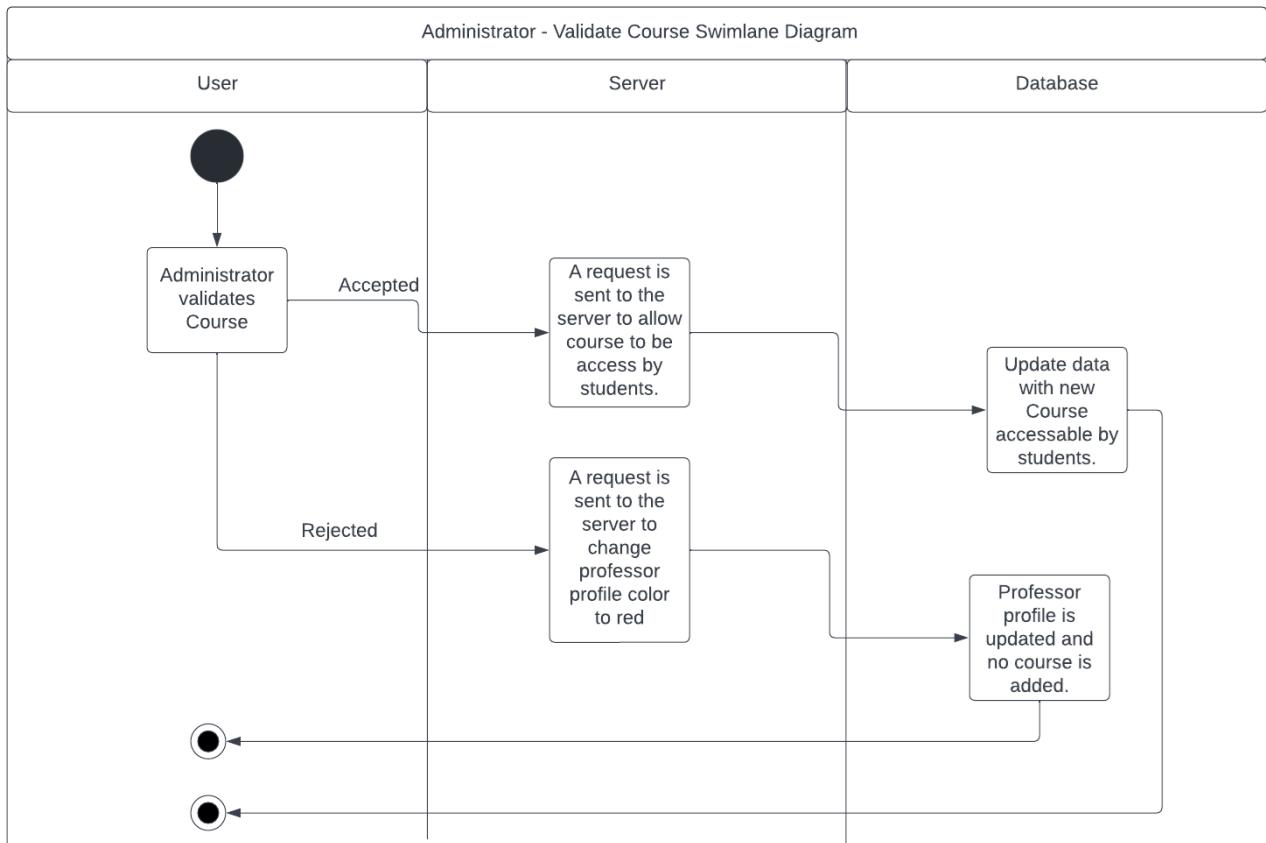
SW_1.3 Validate Professor Profile



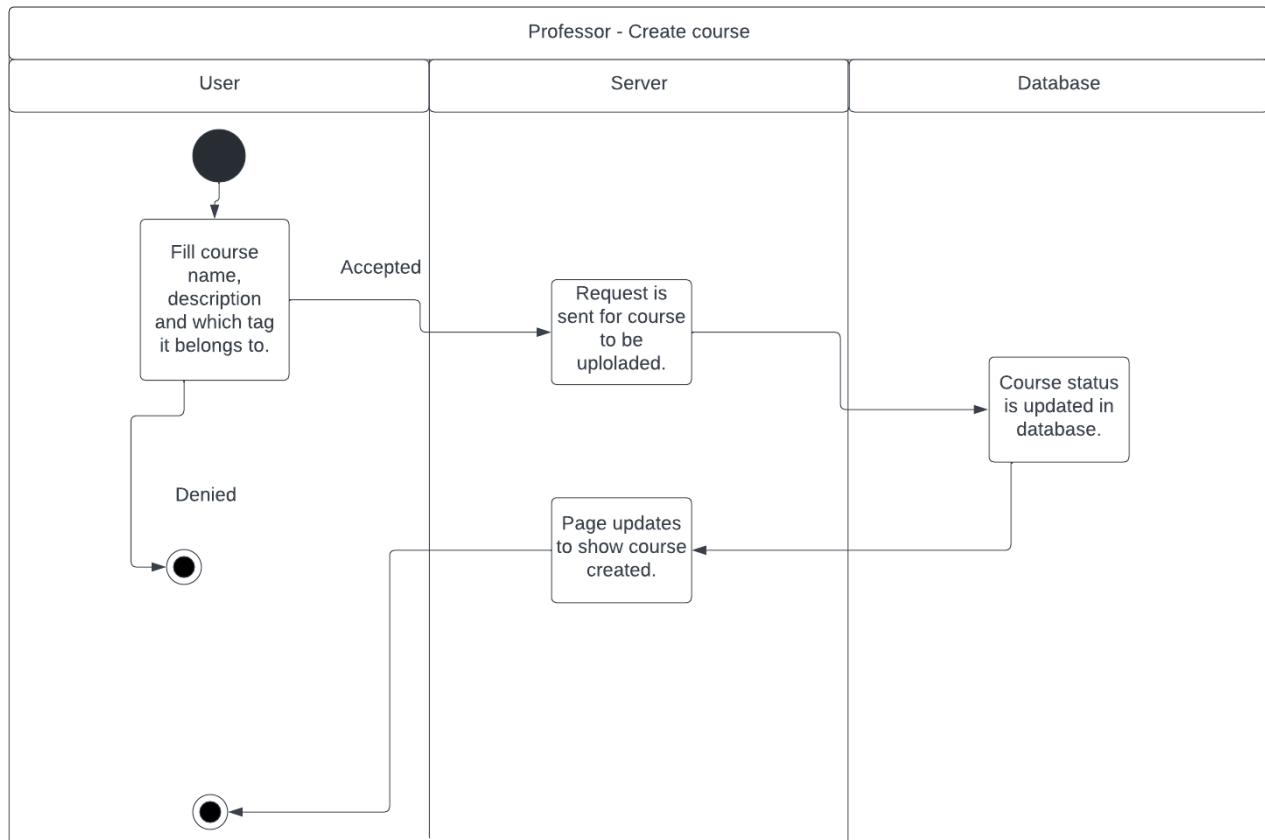
SW_1.4 Delete User Profile



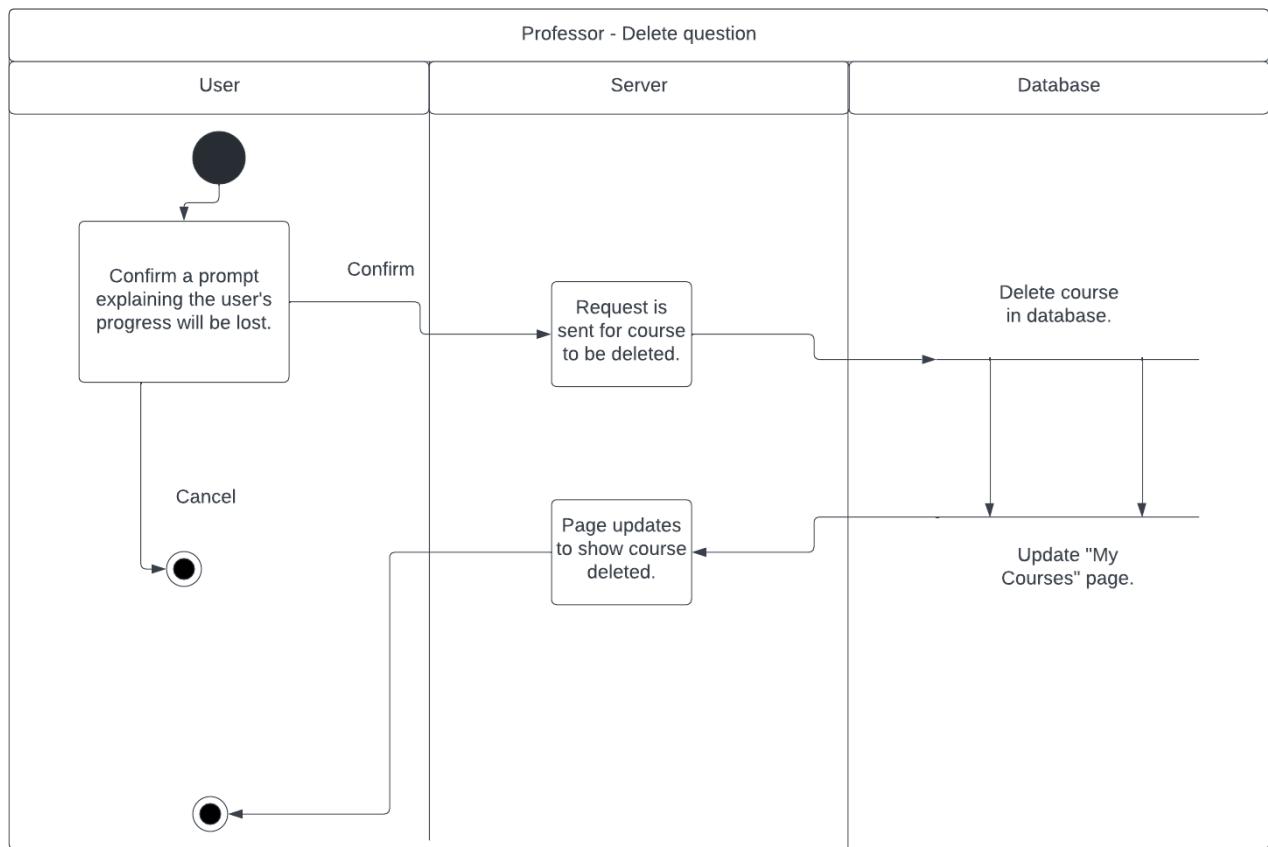
SW_1.5 Validate Course



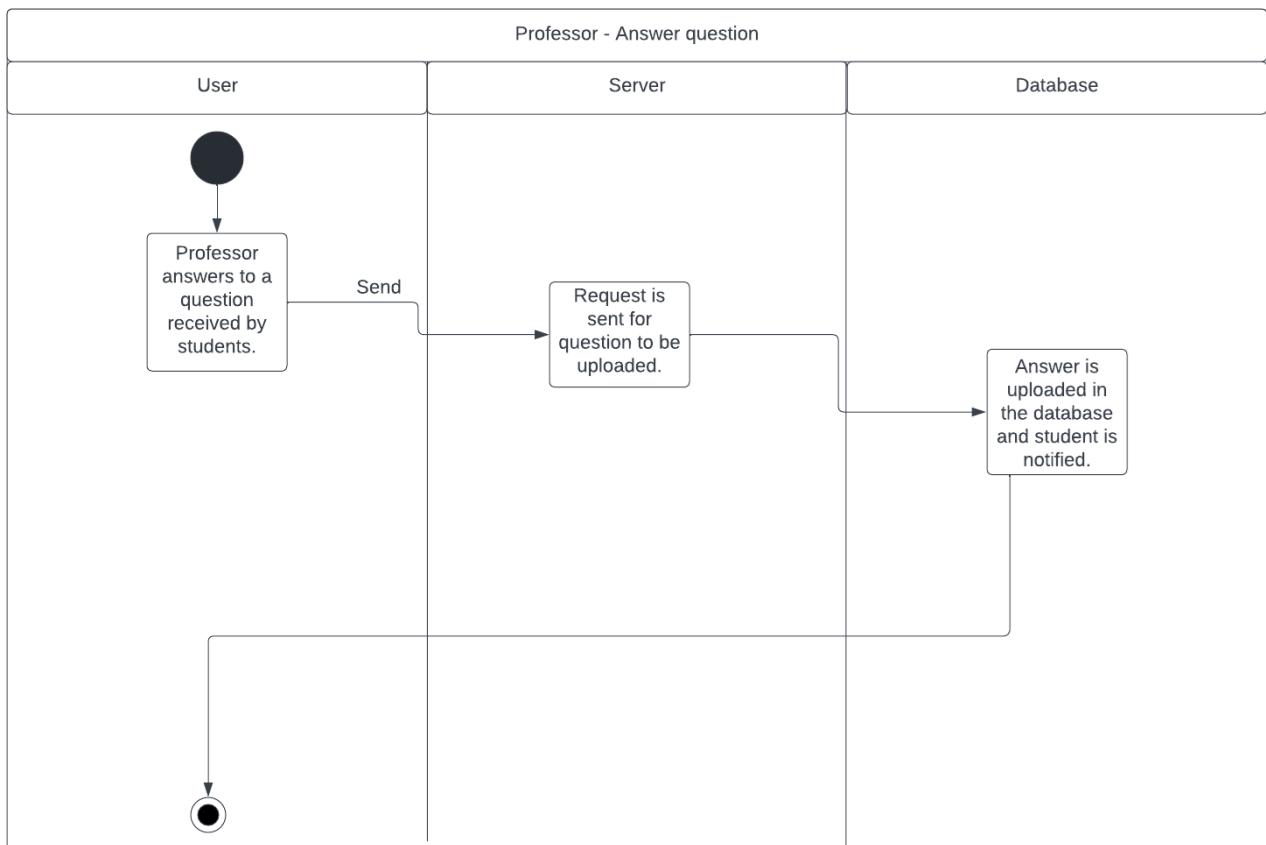
SW_1.6 Create Course



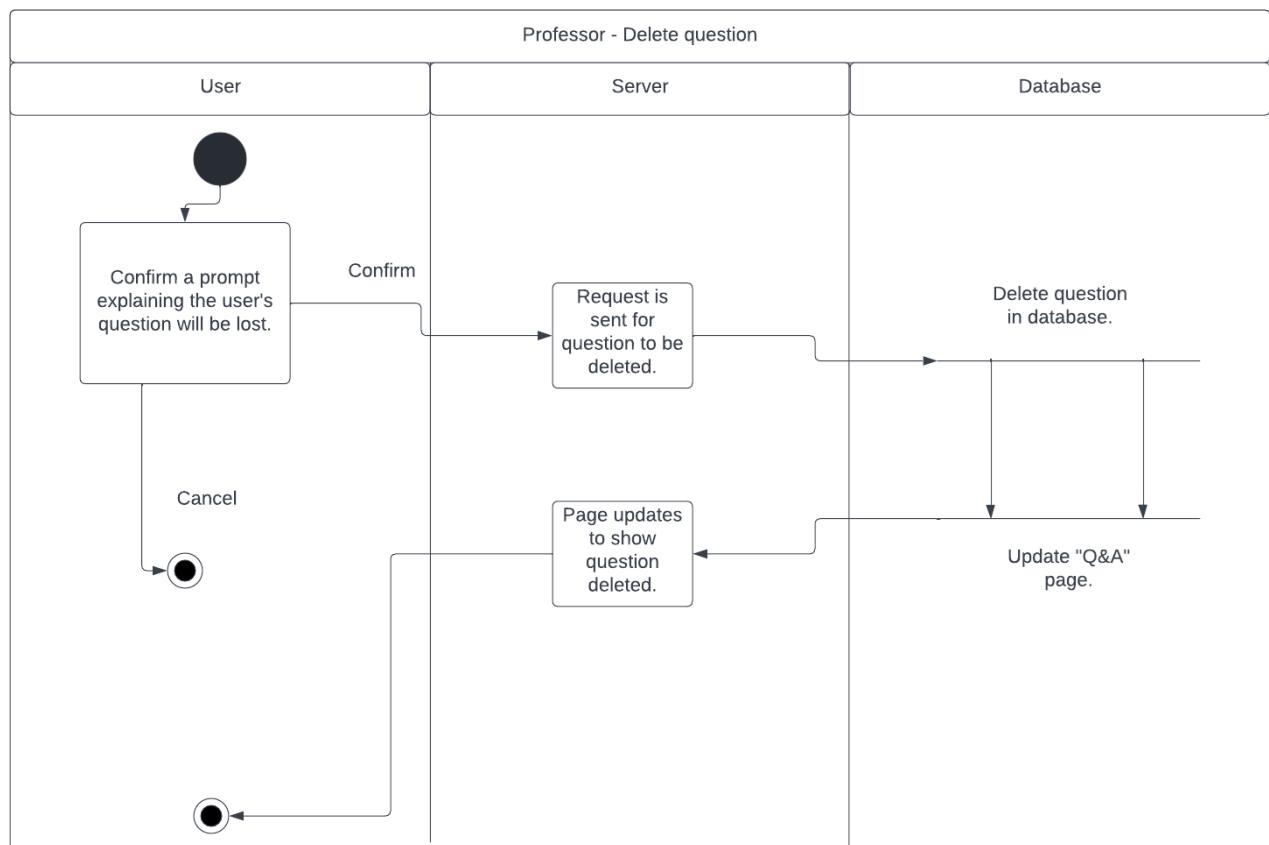
SW_1.7 Delete Course



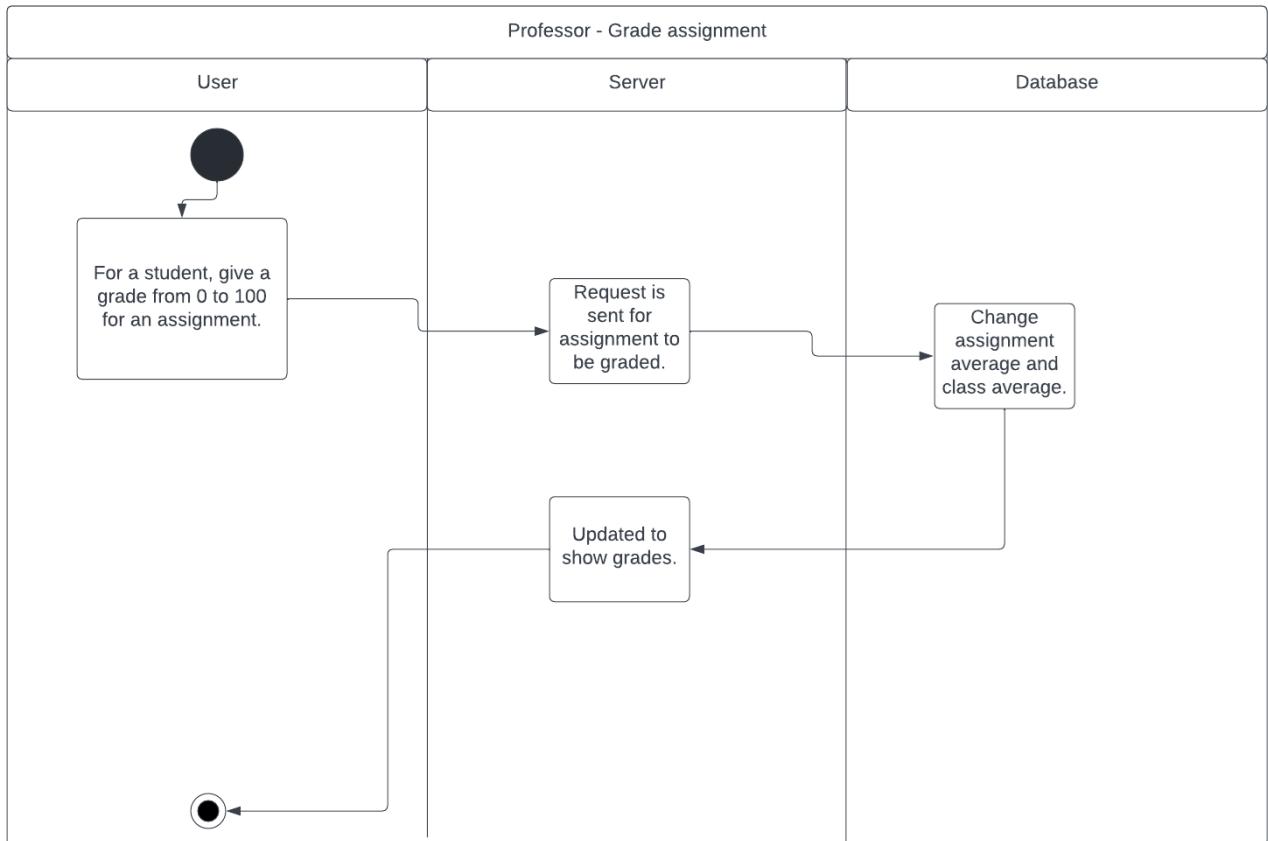
SW_1.8 Answer Question



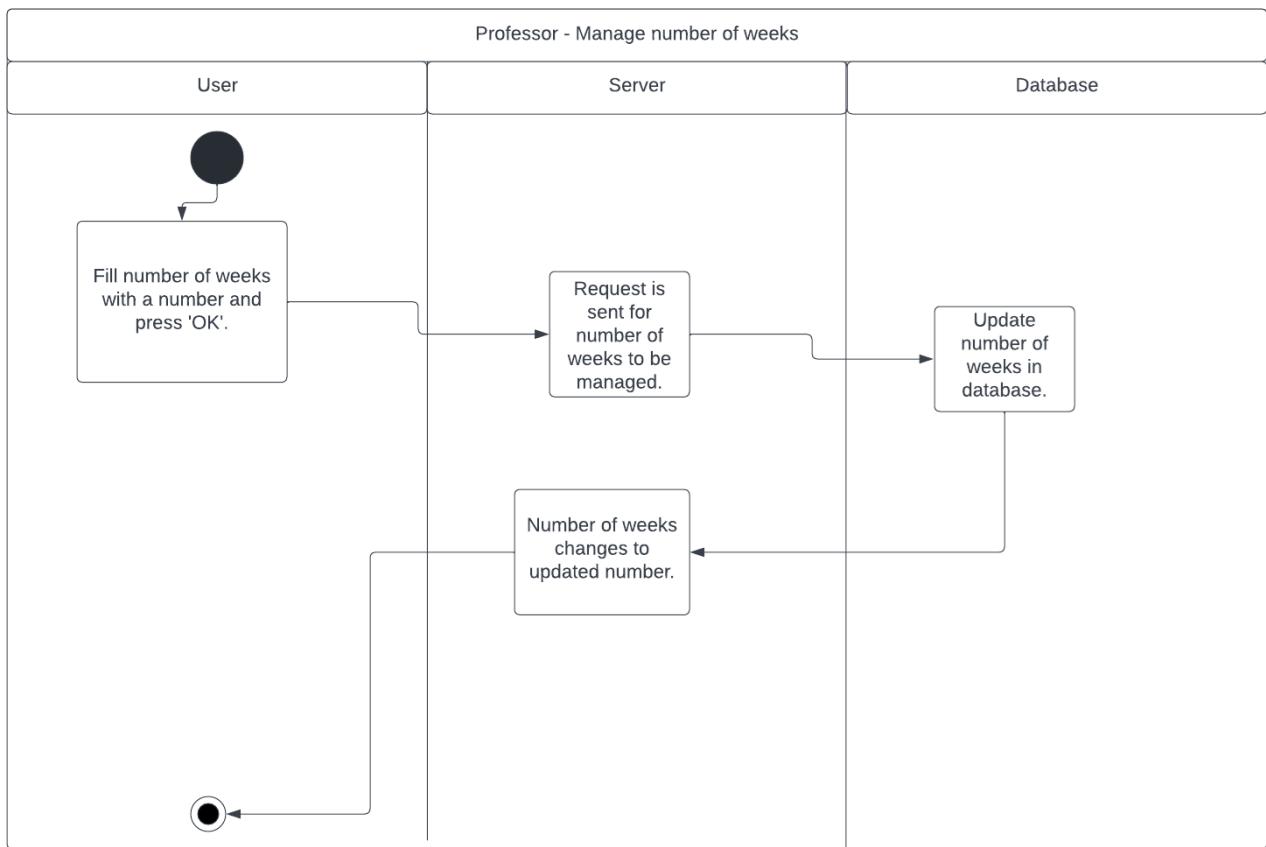
SW_1.9 Delete Question



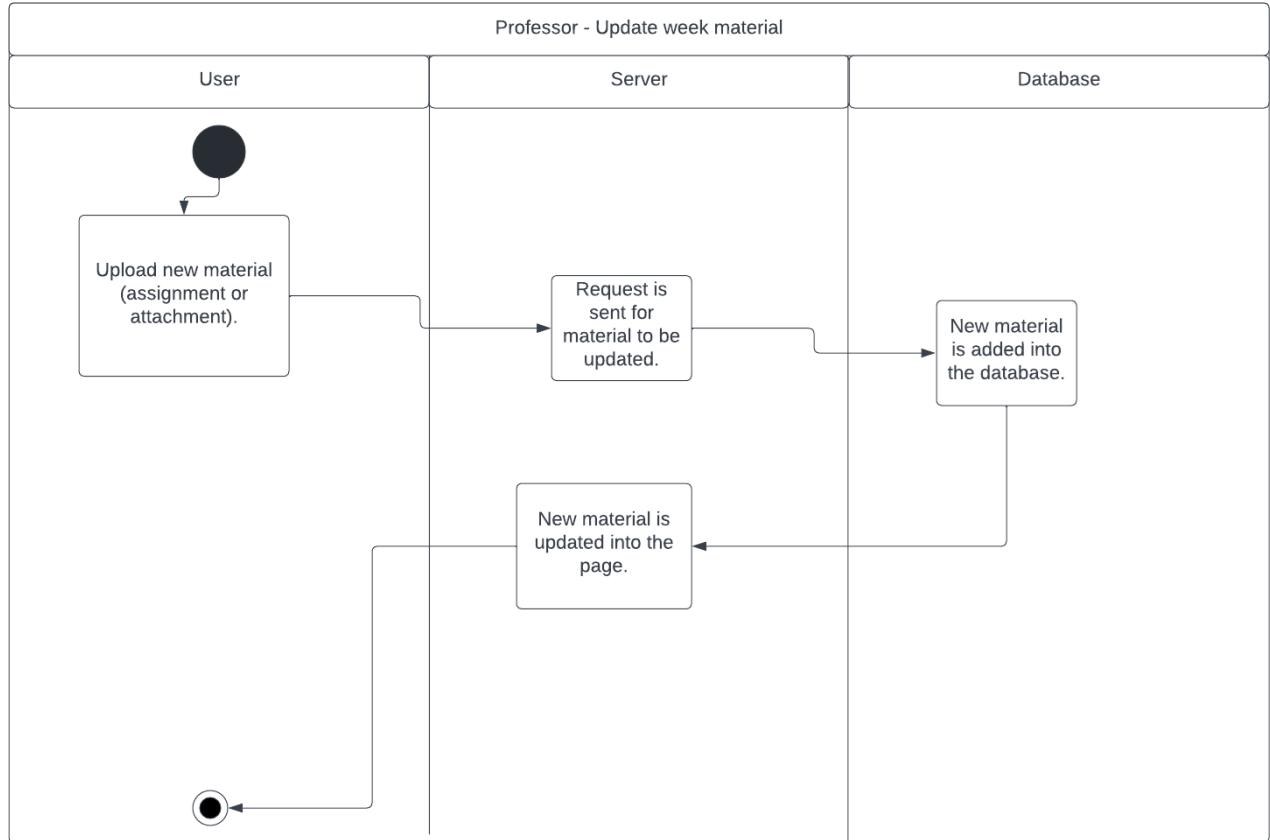
SW_1.10 Grade Assignment



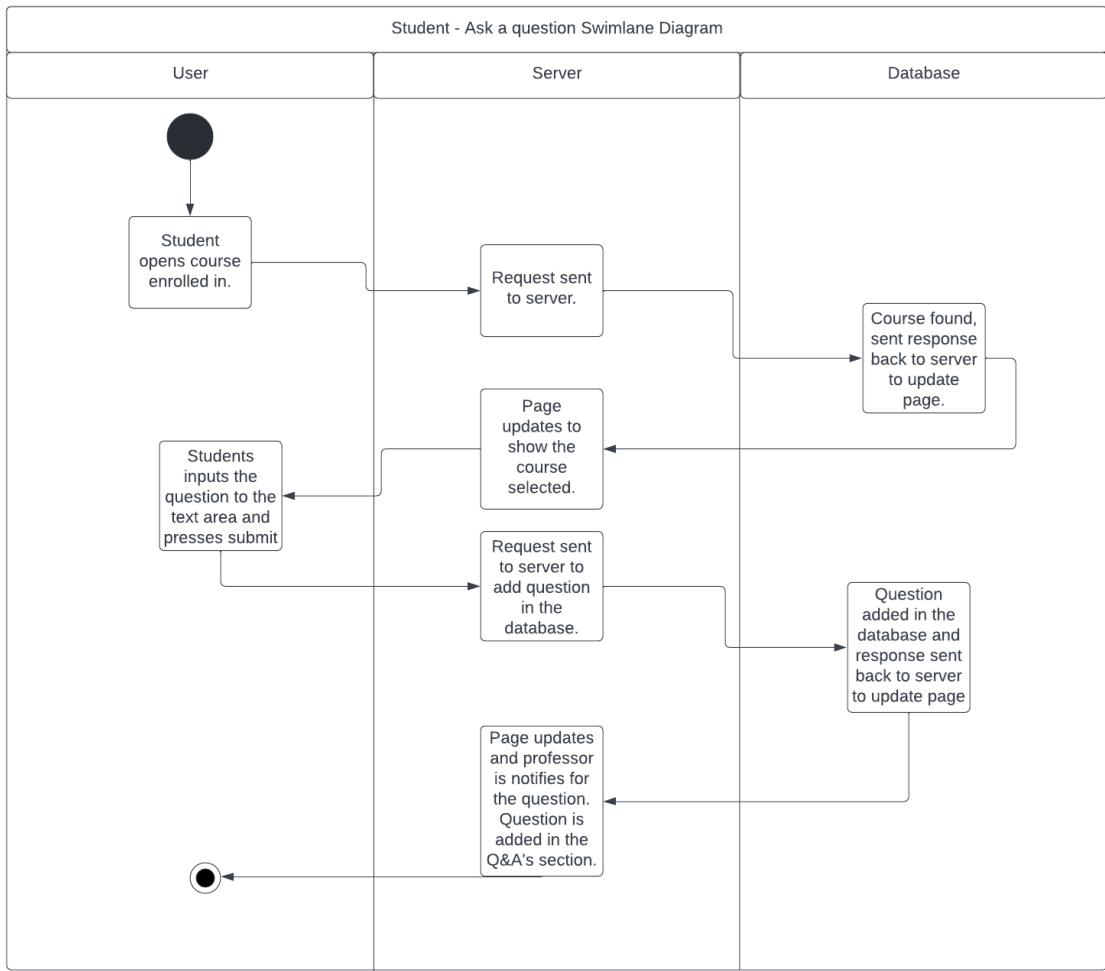
SW_1.11 Manage Number of Weeks



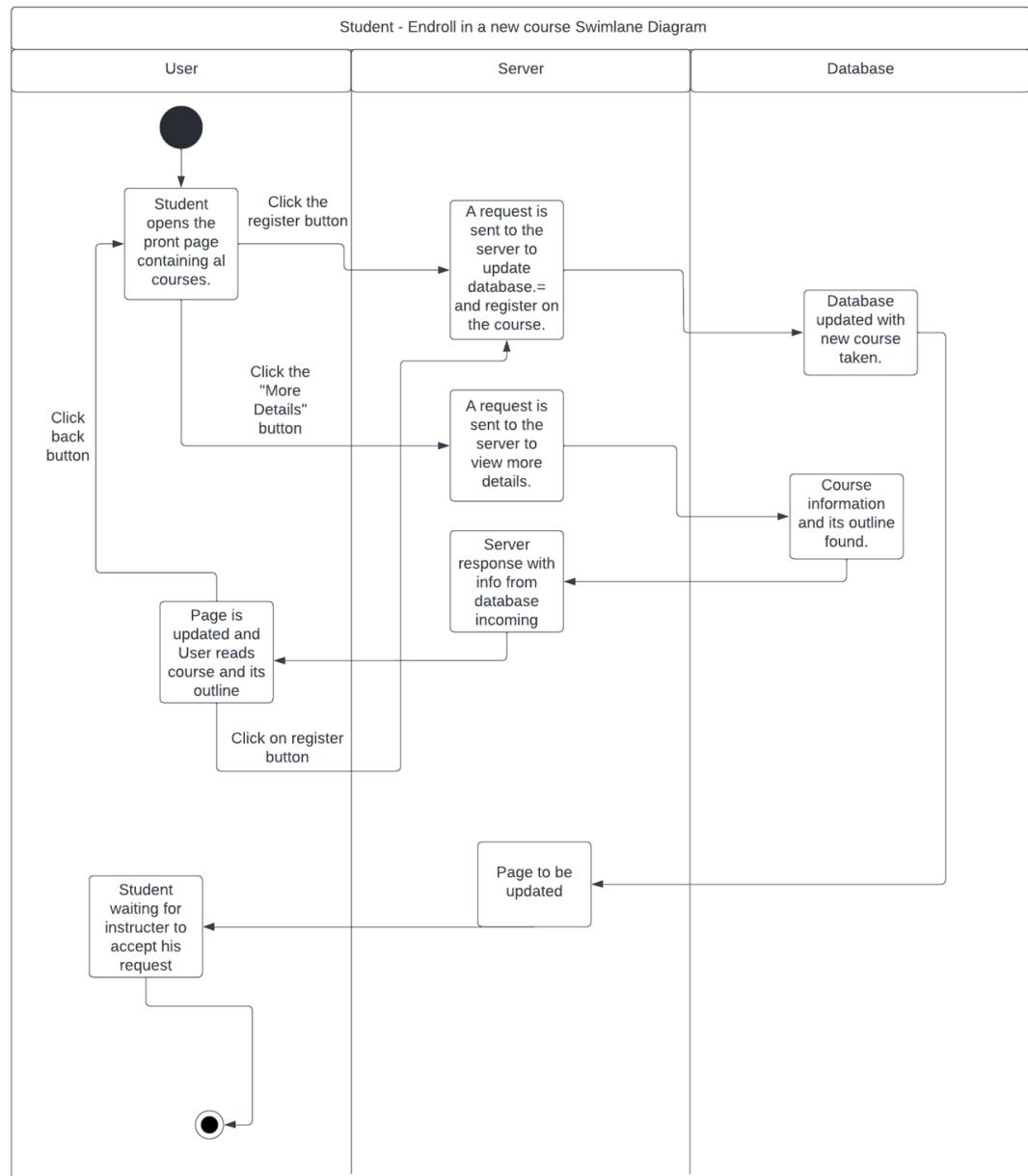
SW_1.12 Update Week Material



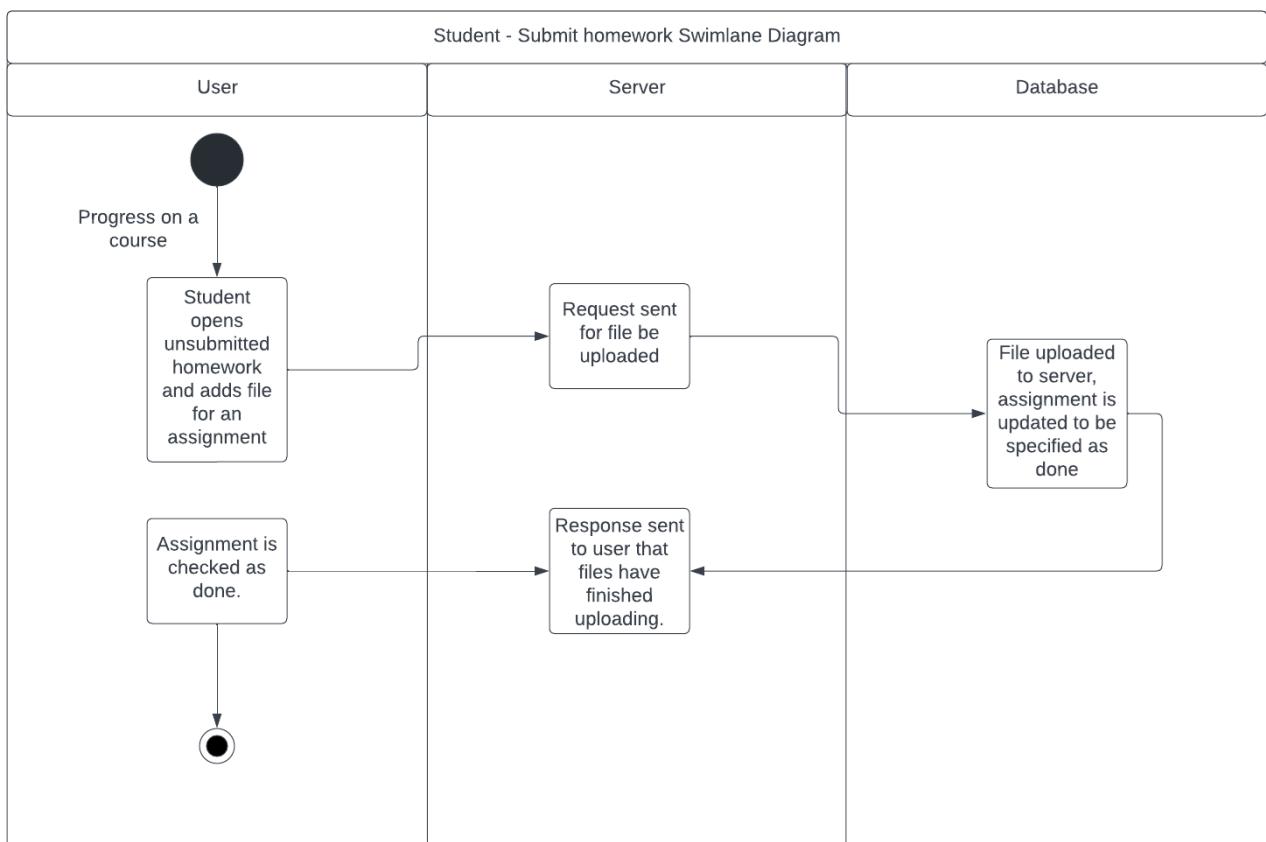
SW_1.13 Ask a Question



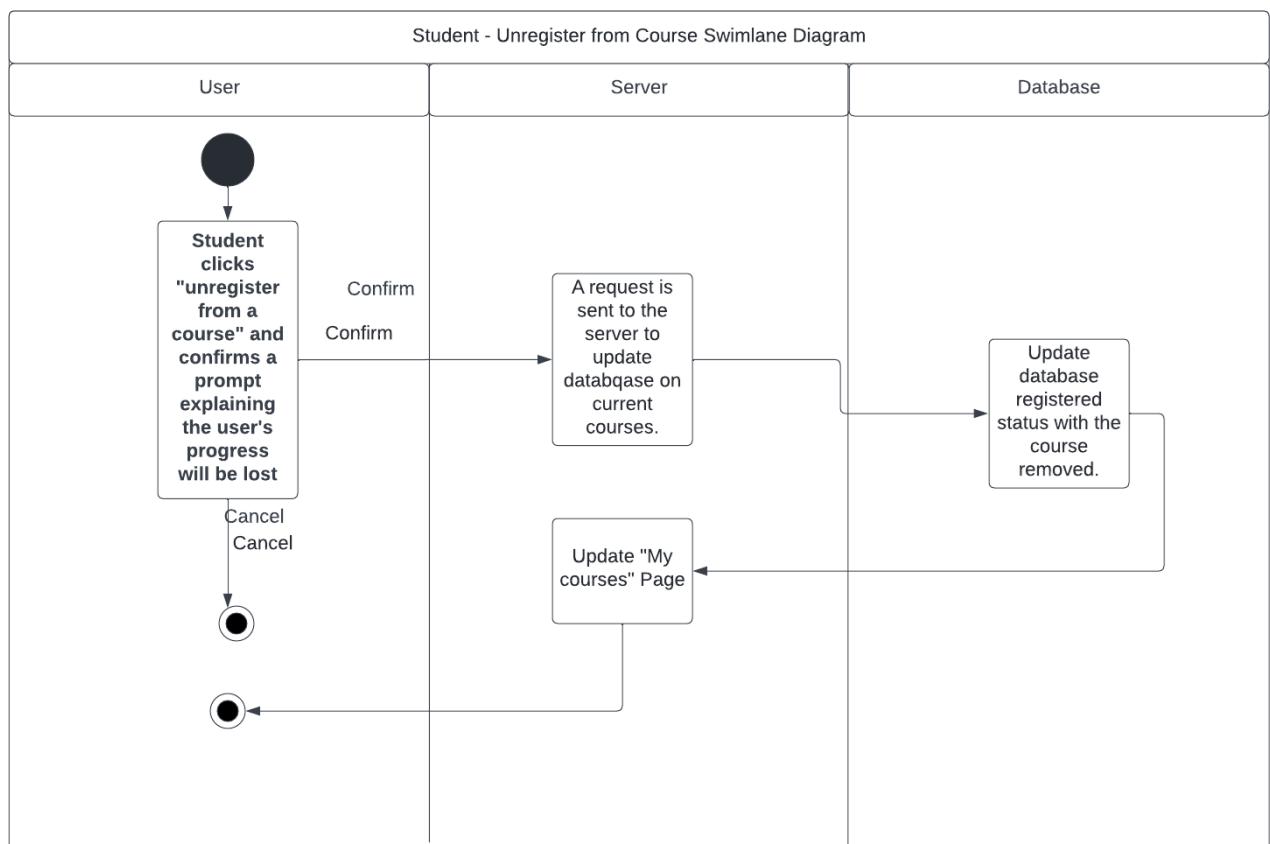
SW_1.14 Enroll in a new Course



SW_1.15 Submit Homework

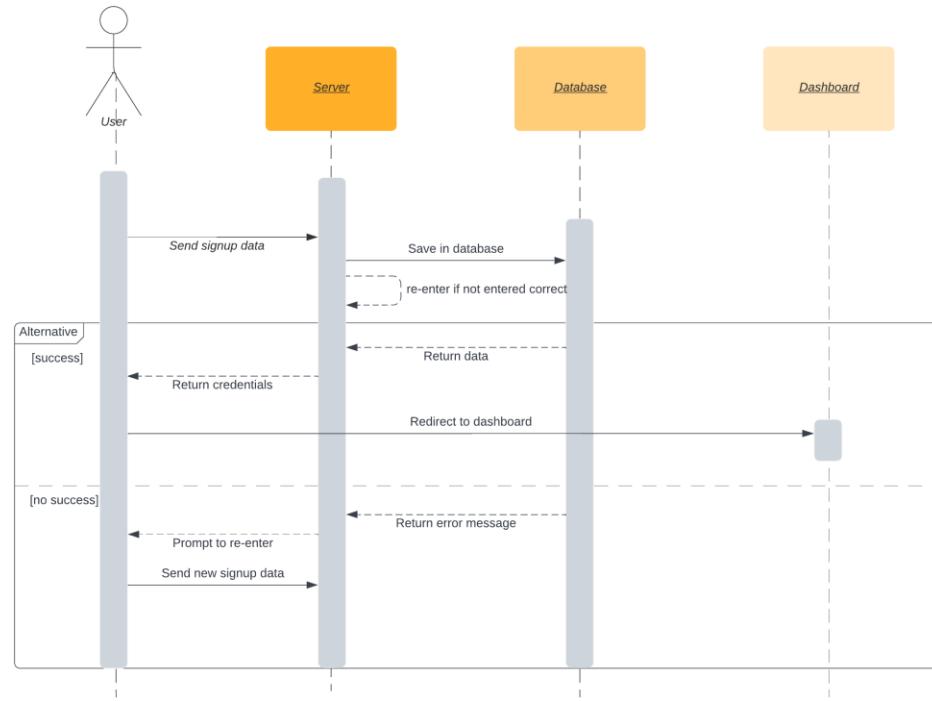


SW_1.16 Unregister from Course

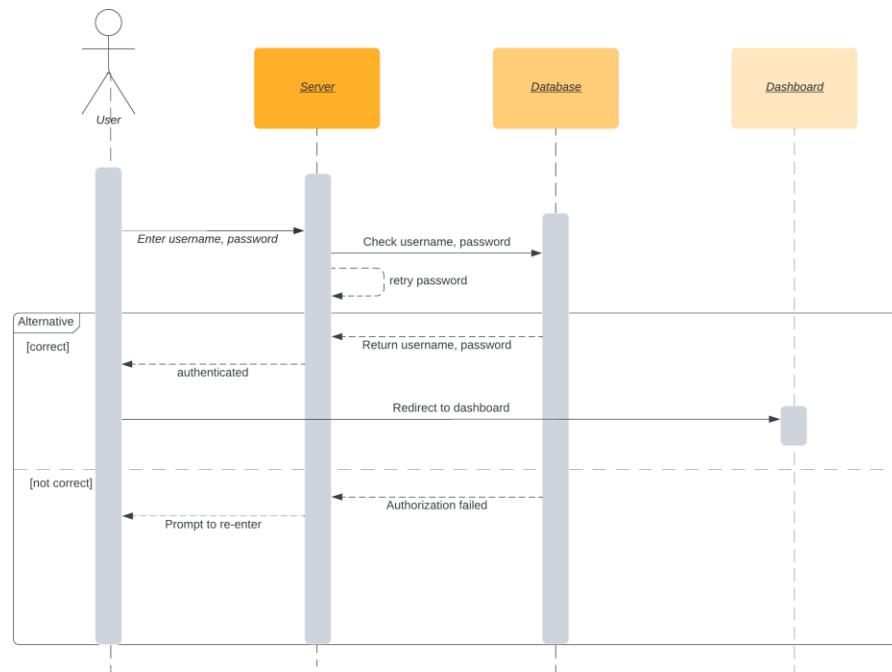


4.2.4 Sequence Diagrams

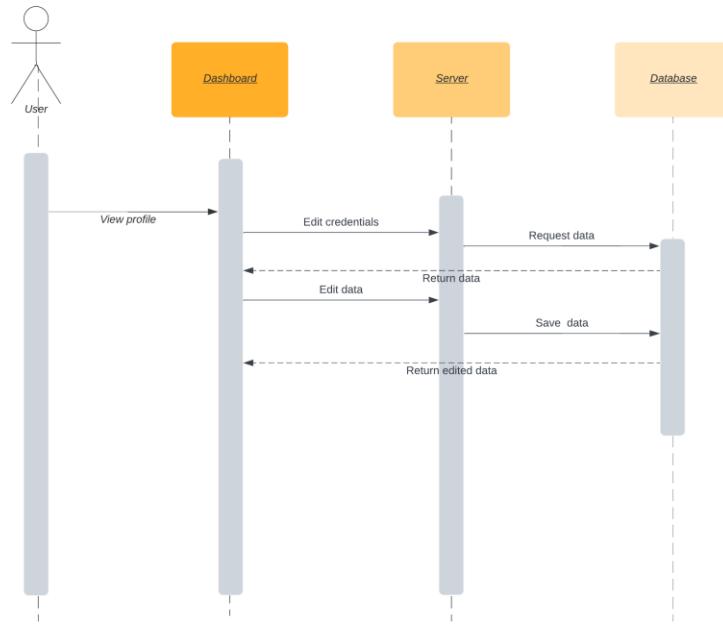
SD_1.1 signup



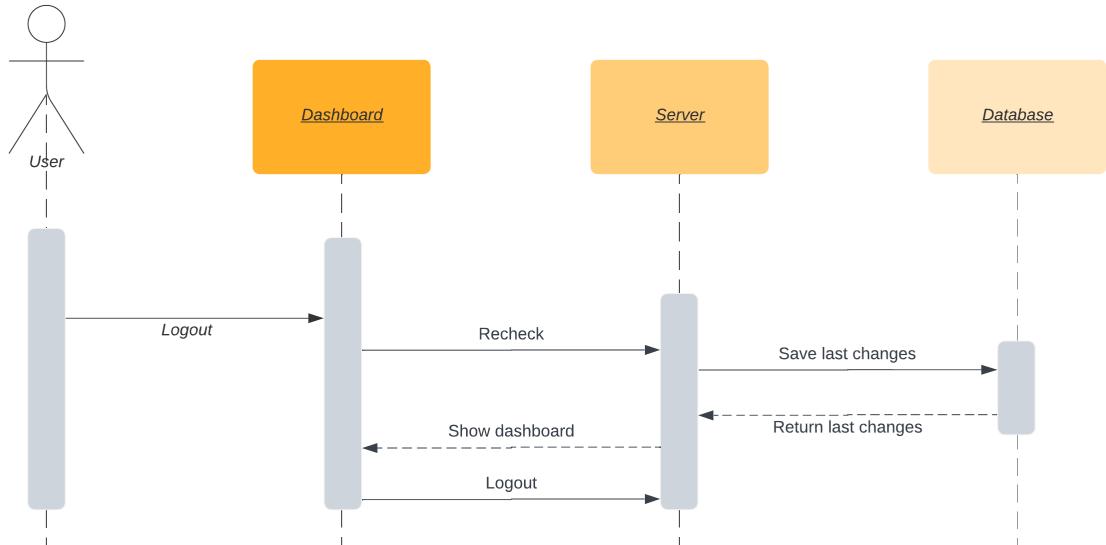
SD_1.2 login



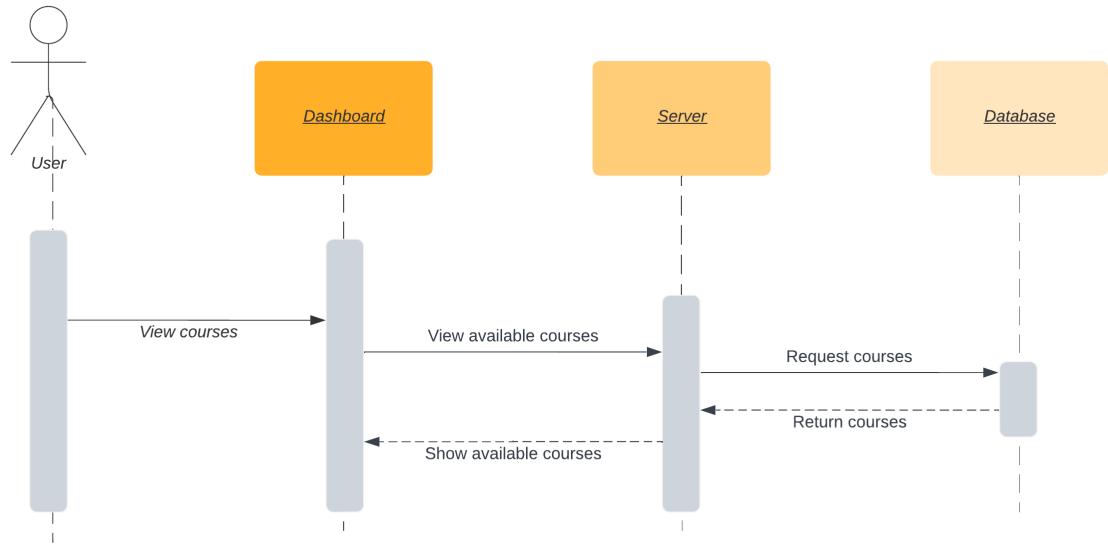
SD_1.3 Edit credentials



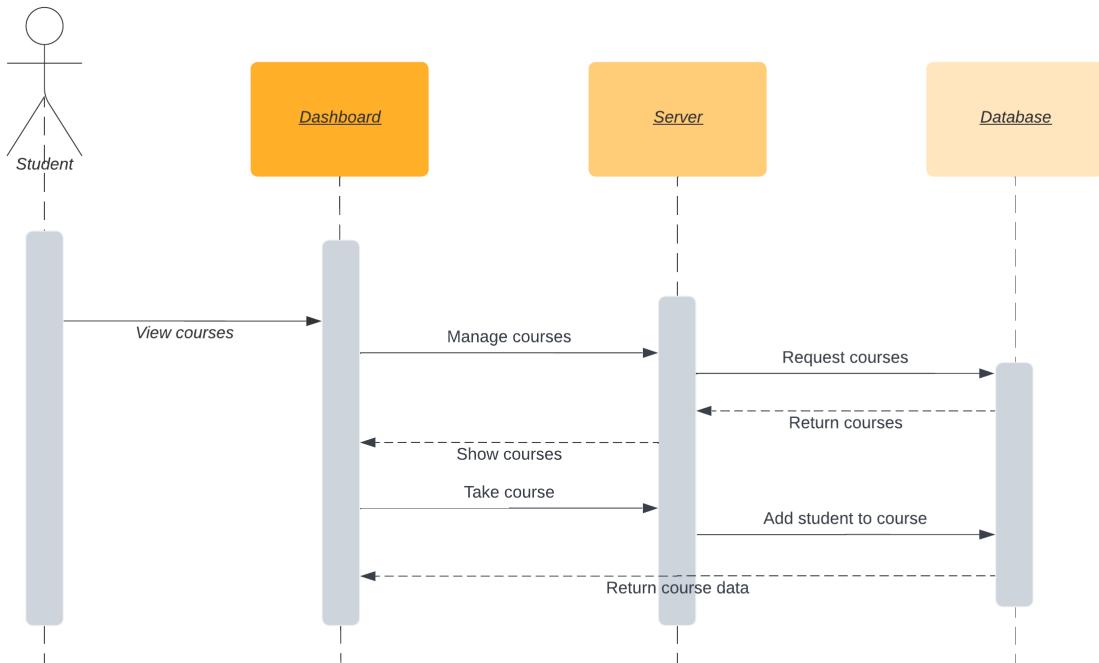
SD_1.4 logout



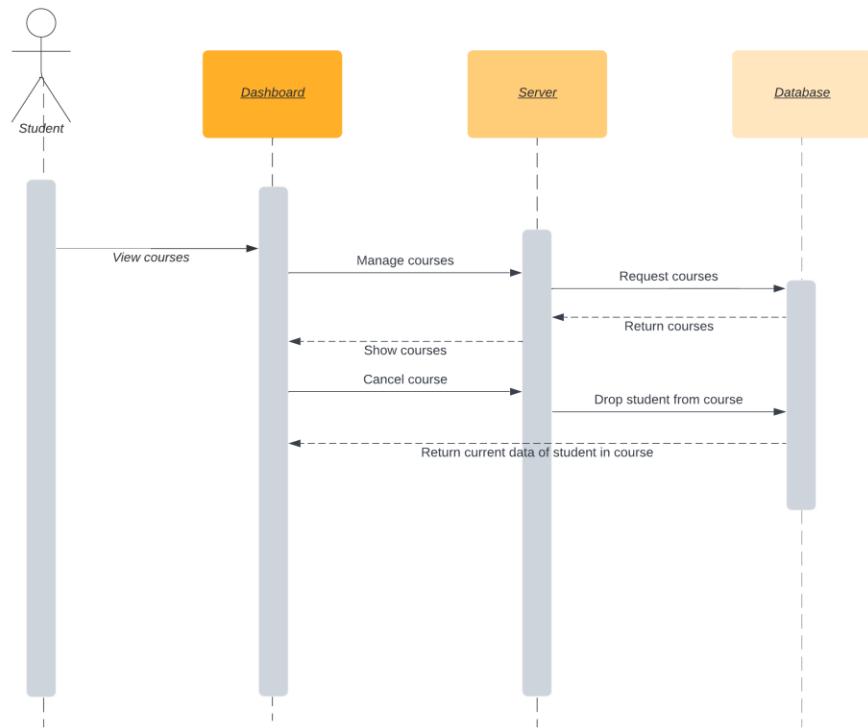
SD_1.5 View available courses



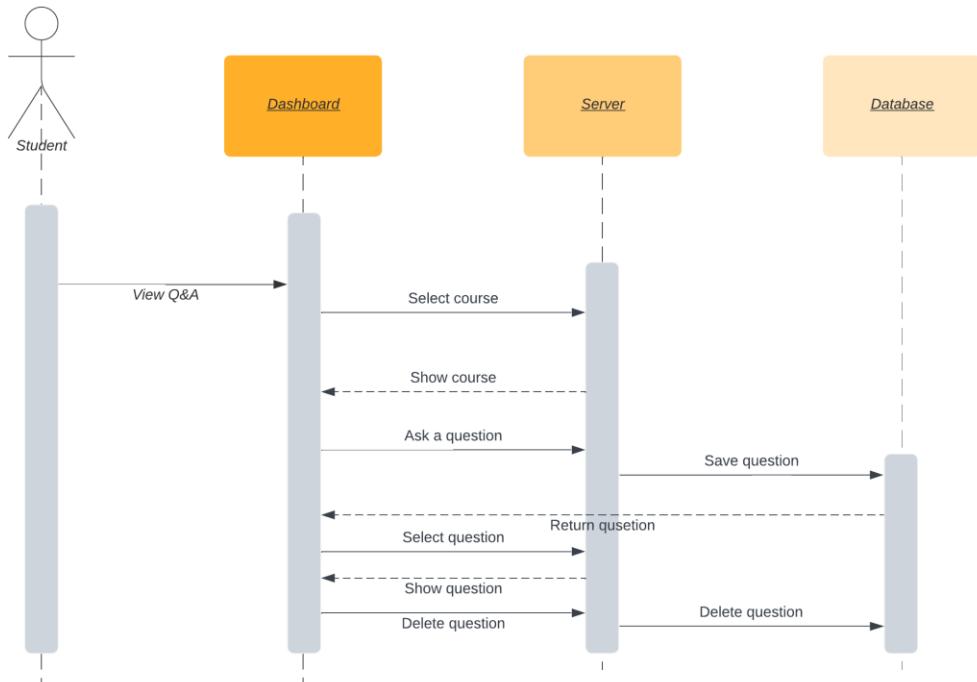
SD_2.1 Manage courses – take



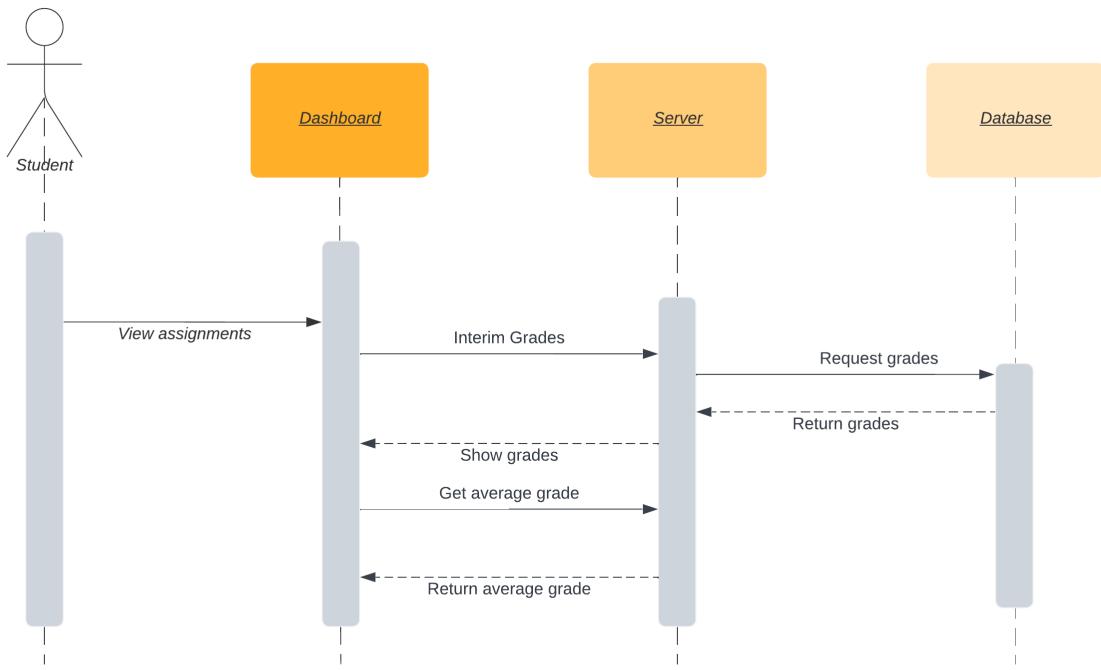
SD_2.1 Manage courses – cancel



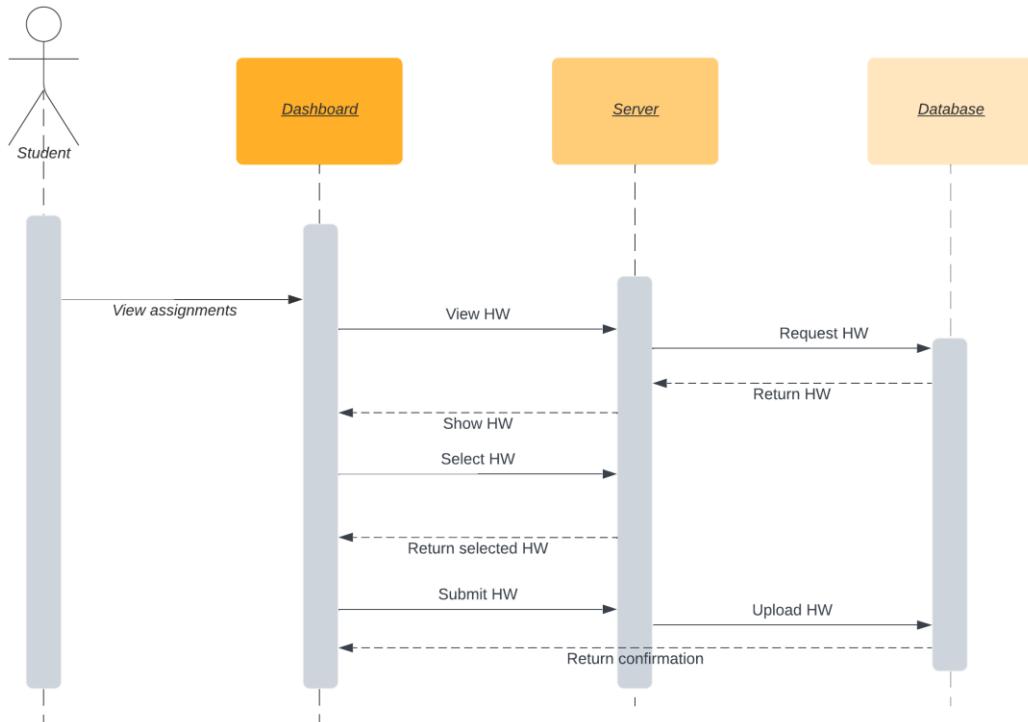
SD_2.2 Ask/Delete questions



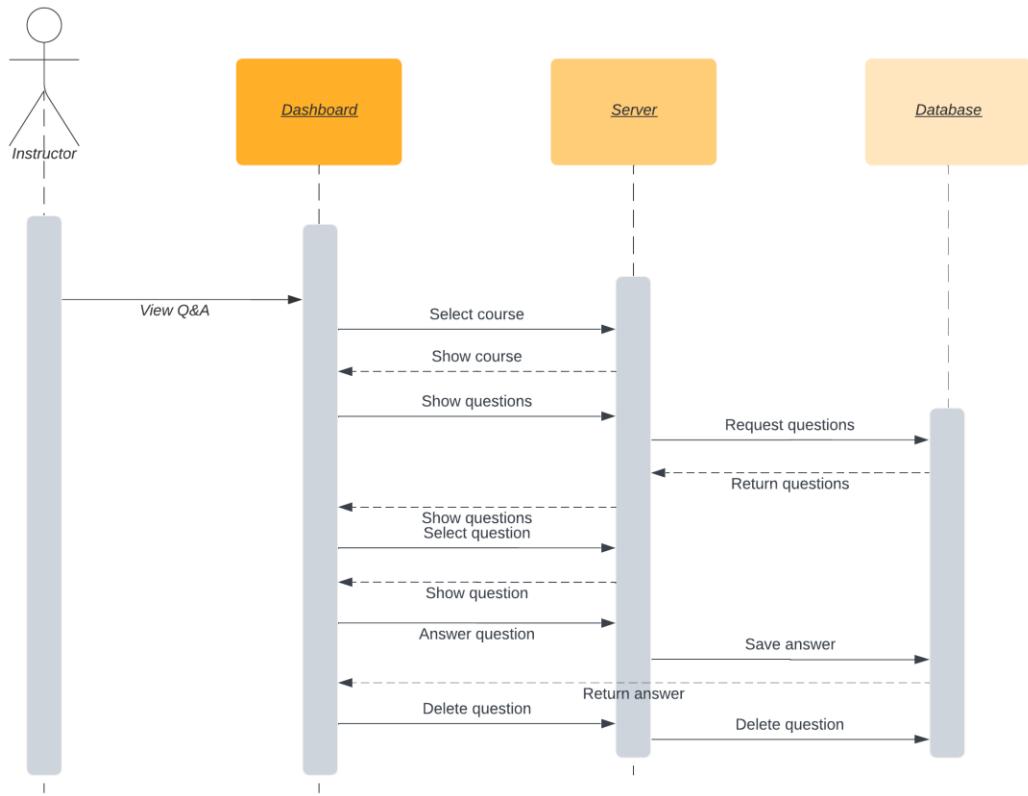
SD_2.3 Get average grade



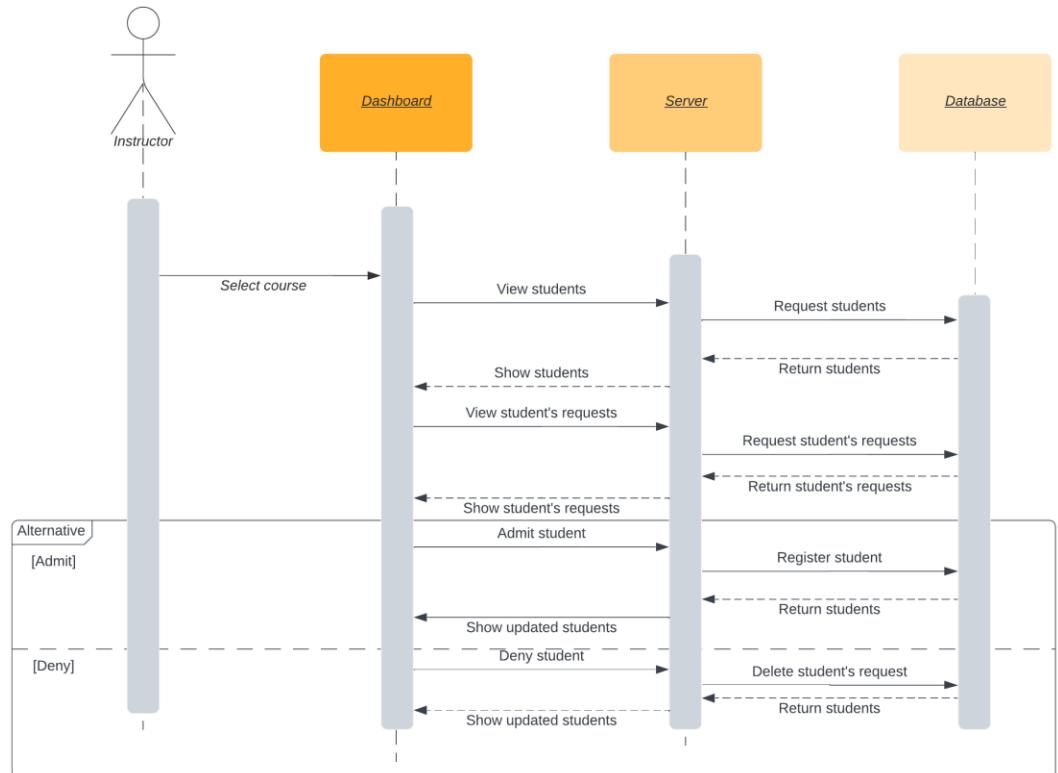
SD_2.4 Submit HW



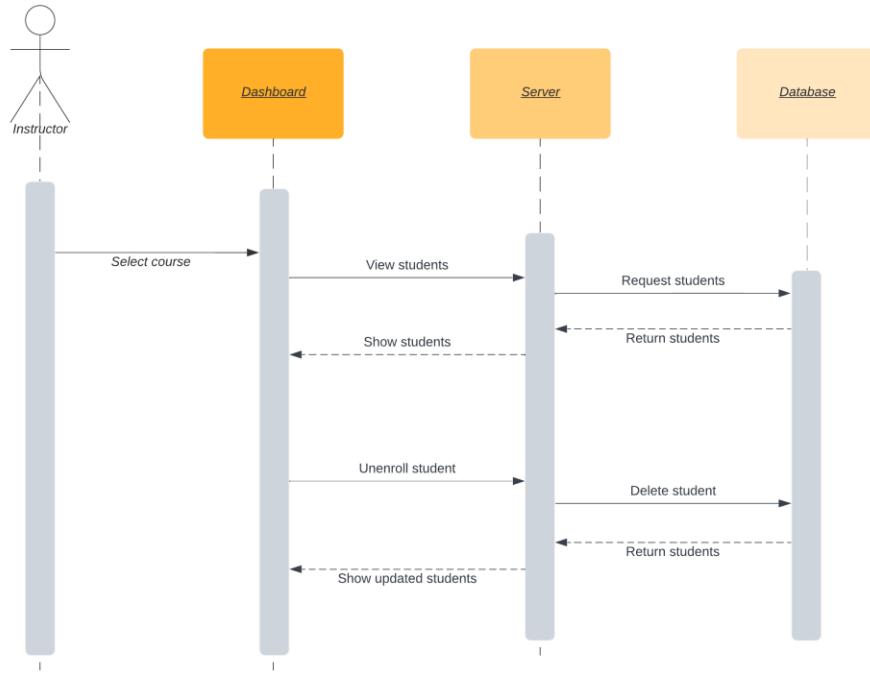
SD_3.1 Manage Q&A – Answer/Delete



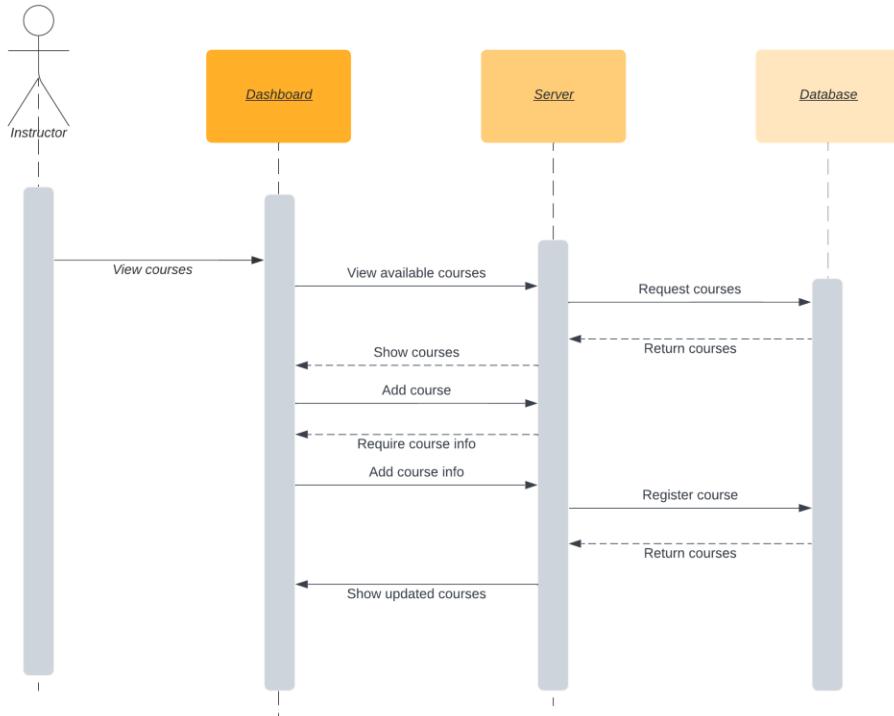
SD_3.2 Manage Students – Admit/Deny



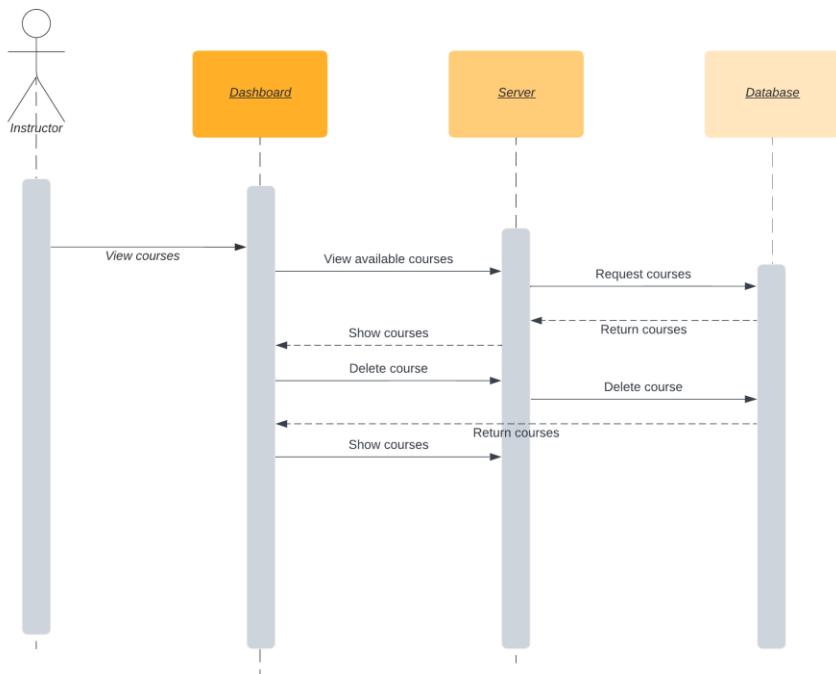
SD_3.2 Manage students – Unenroll



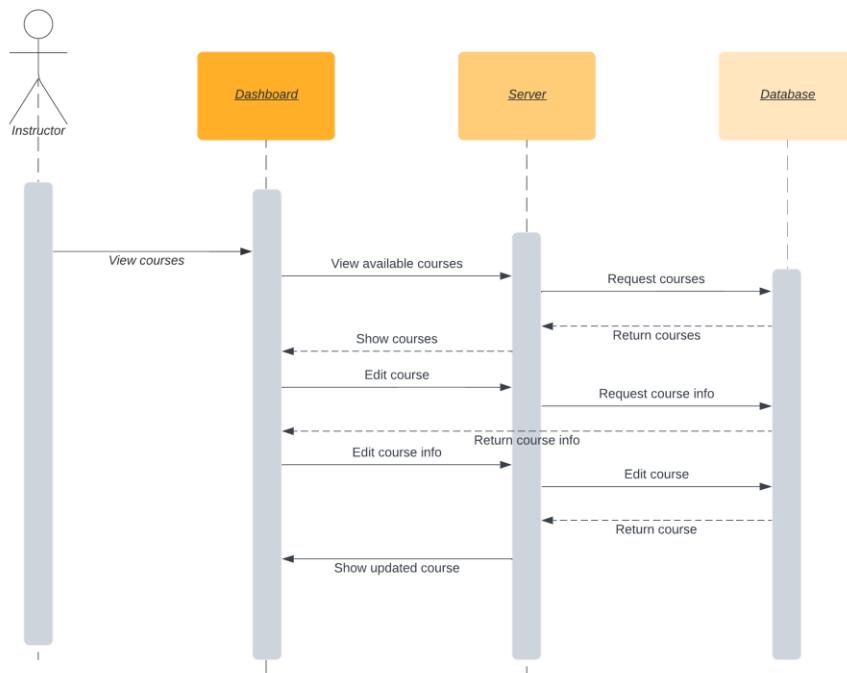
SD_3.3 Manage courses – Add



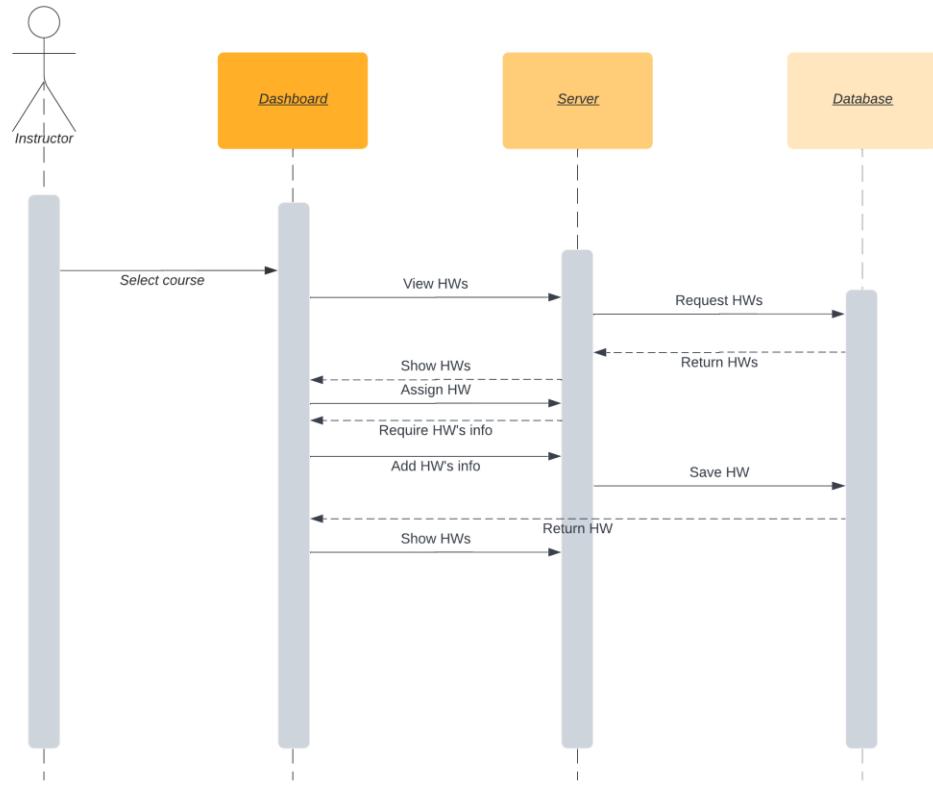
SD_3.3 Manage courses – Delete



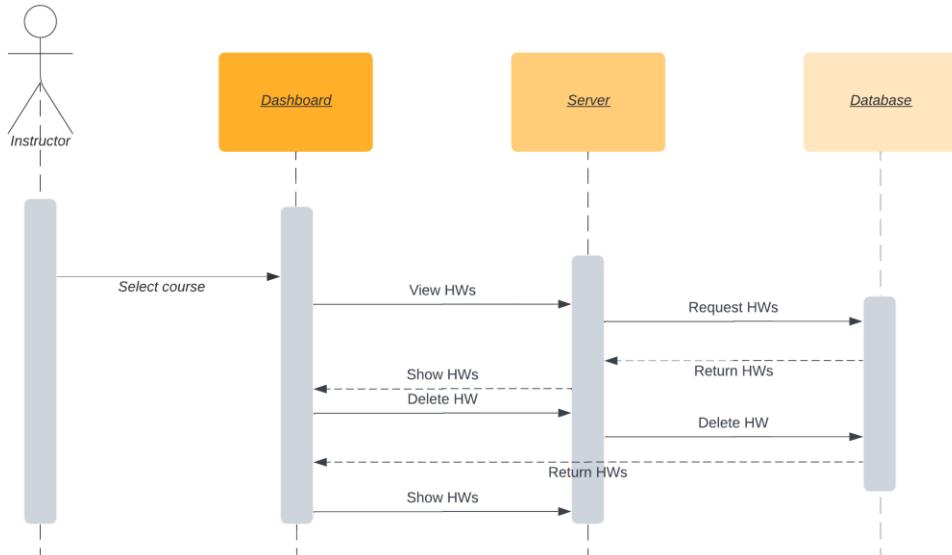
SD_3.3 Manage courses - Edit



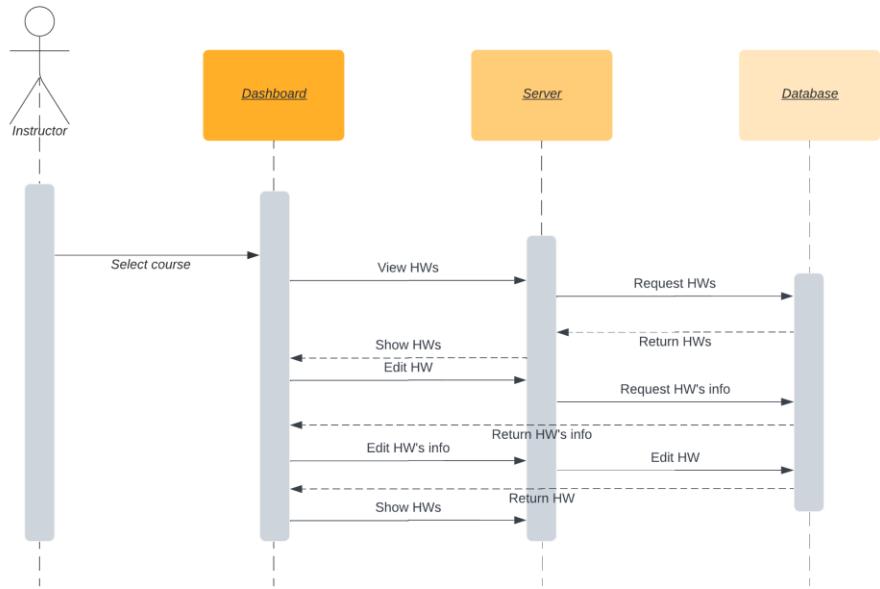
SD_3.4 Manage HW – Assign



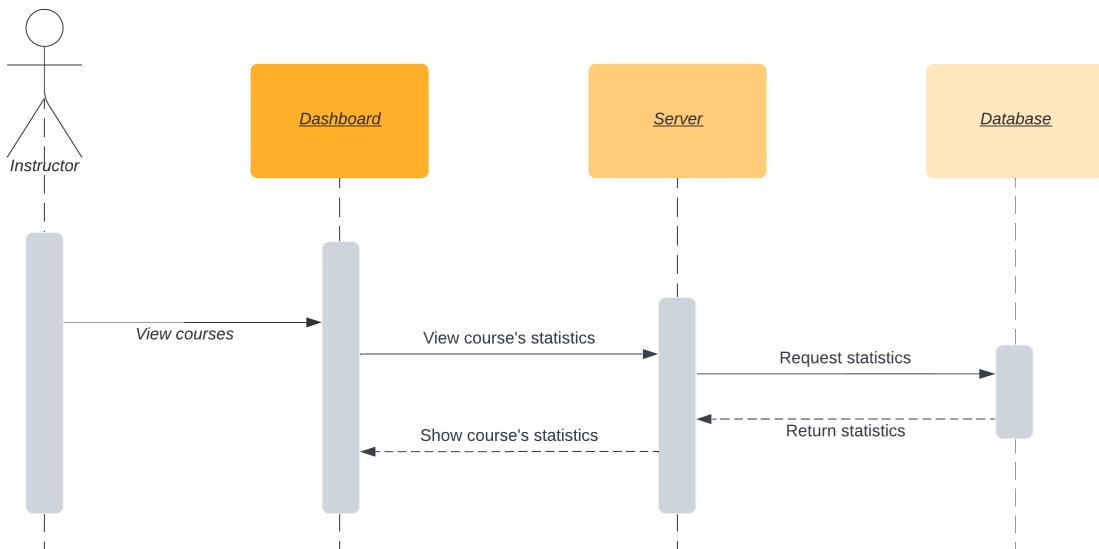
SD_3.4 Manage HW - Delete



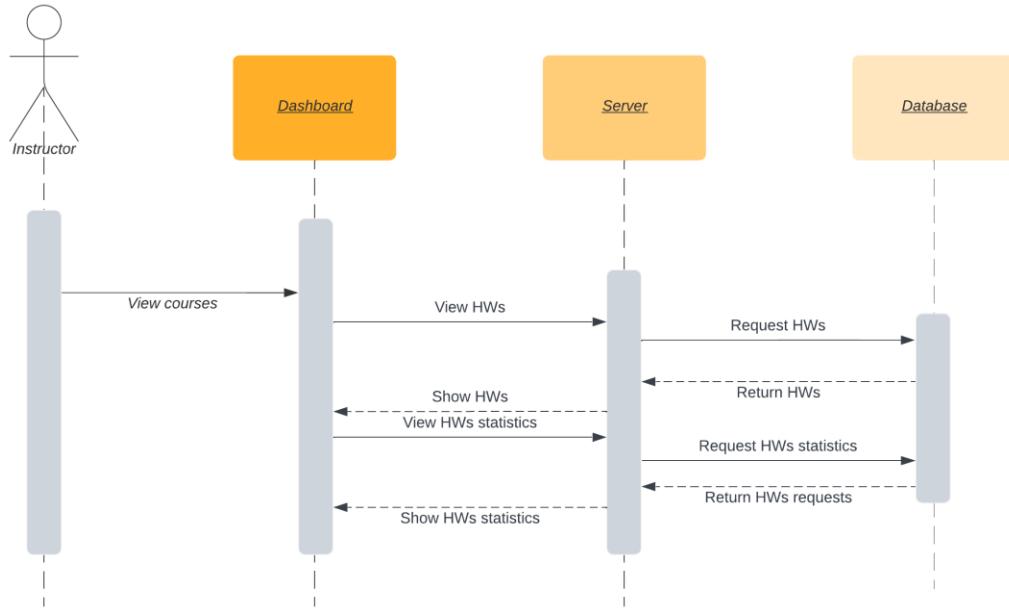
SD_3.4 Manage HW – Edit



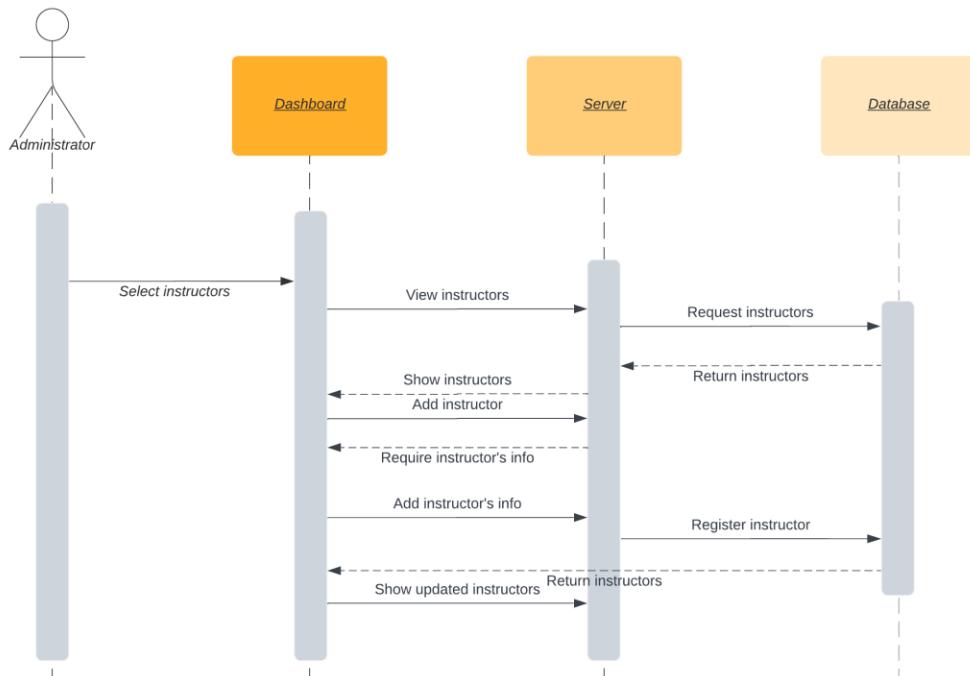
SD_3.5 Check course's statistics



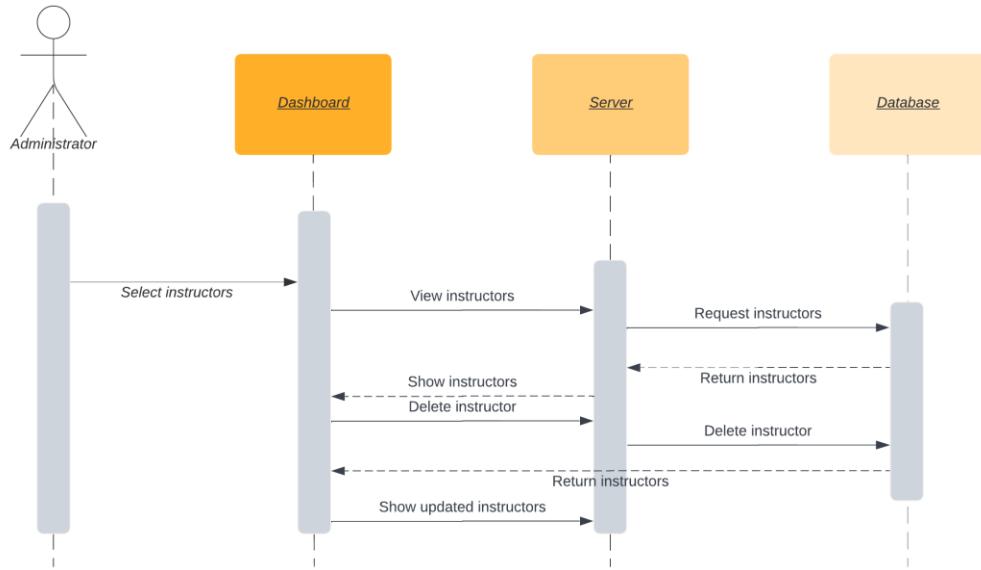
SD_3.6 Check HWs' statistics



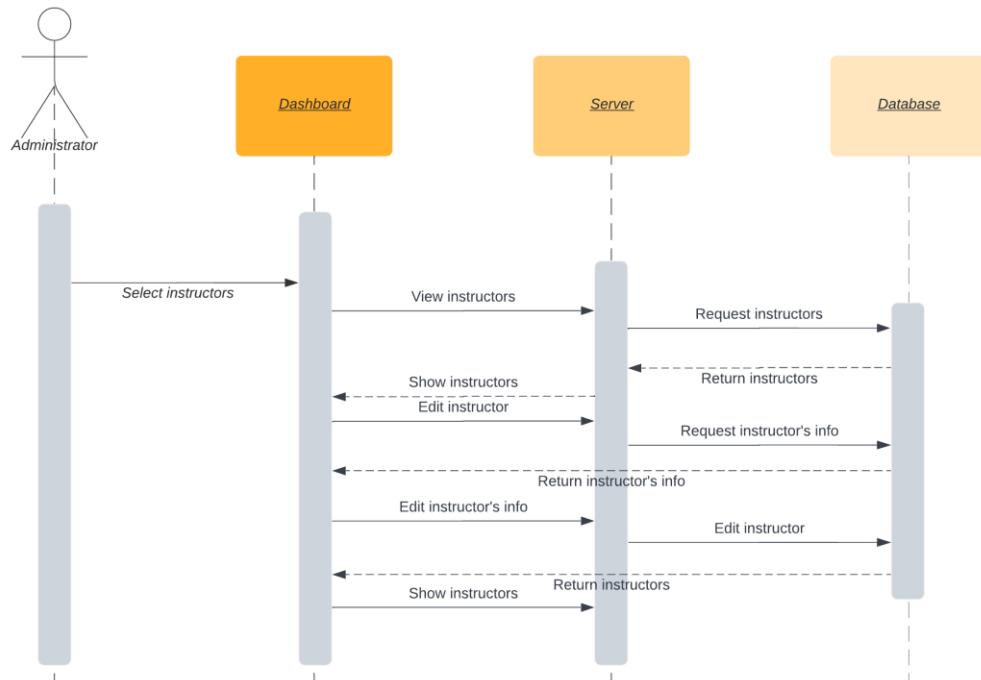
SD_4.1 Manage instructors – Add



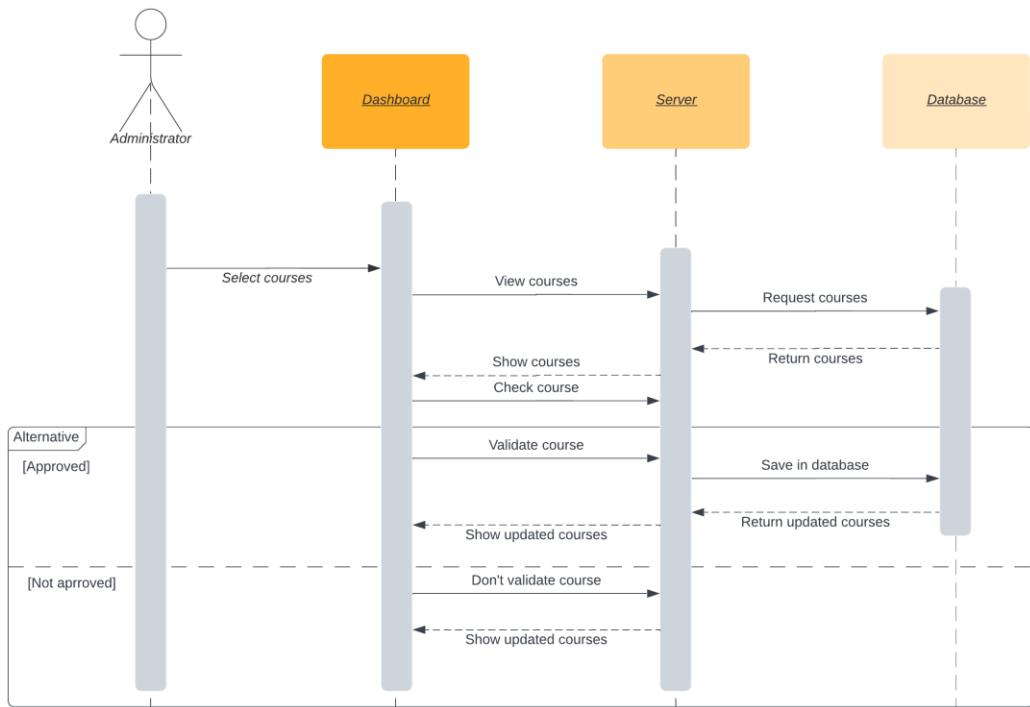
SD_4.1 Manage instructors – Delete



SD_4.1 Manage instructors – Edit

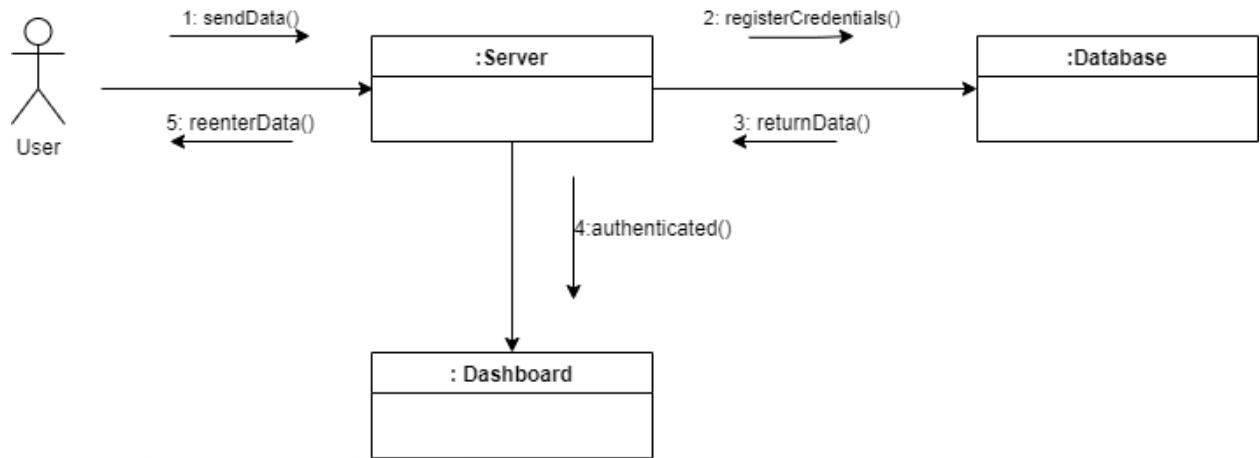


SD_4.2 Validate Course

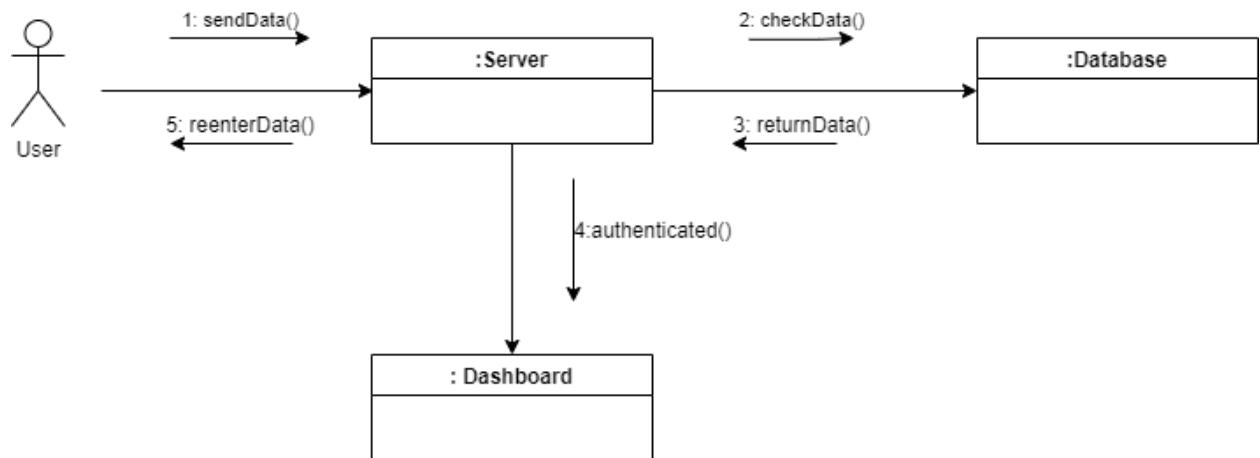


4.2.5 Collaboration Diagrams

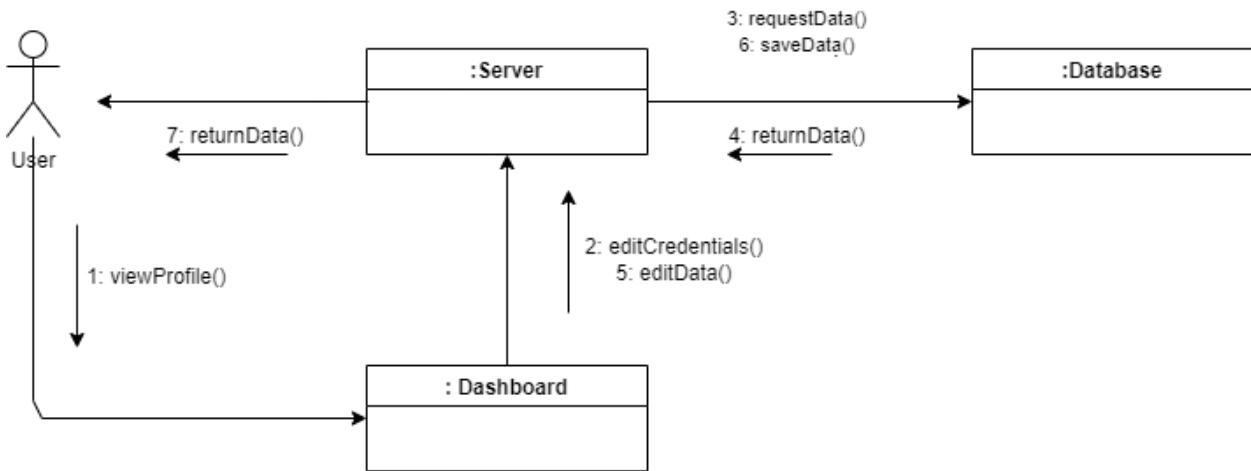
COLLAB_1.1 Signup



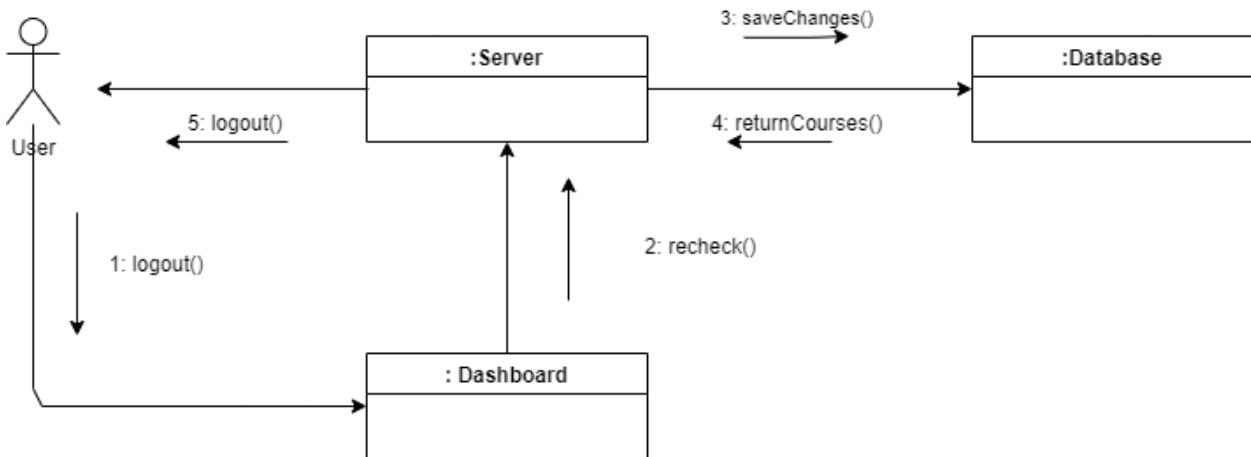
COLLAB_1.2 Login



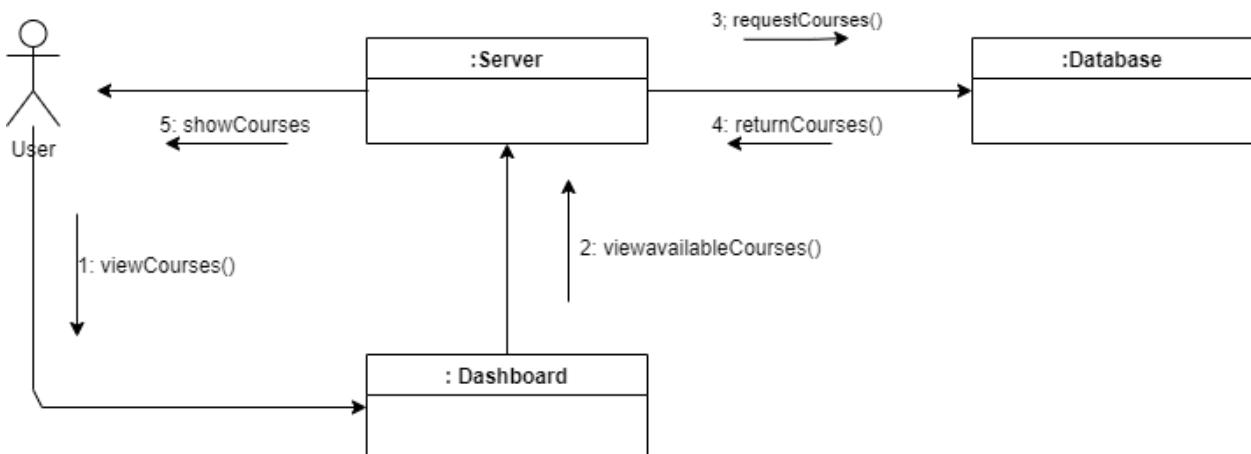
COLLAB_1.3 Edit Credentials



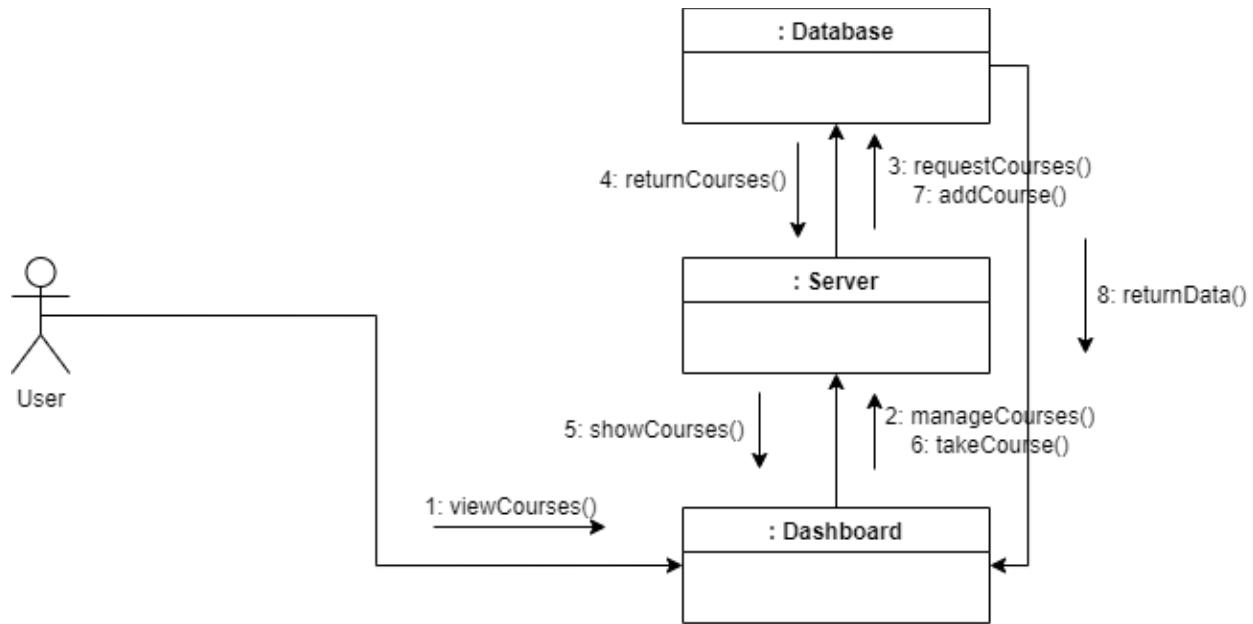
COLLAB_1.4 Logout



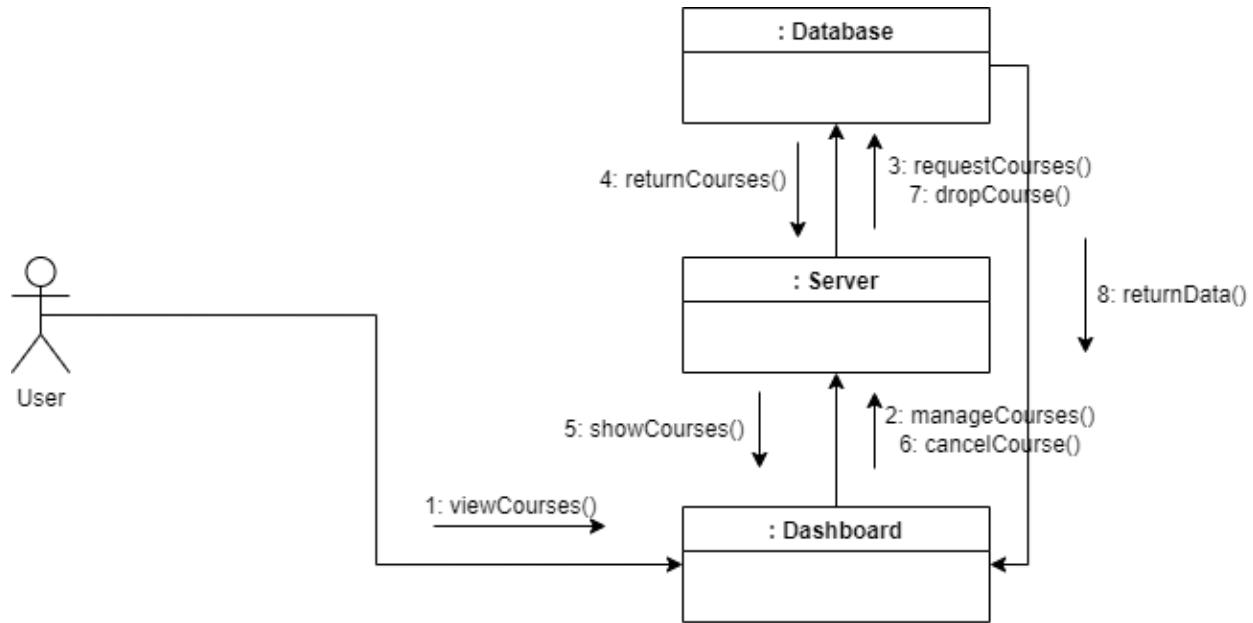
COLLAB_1.5 View available courses



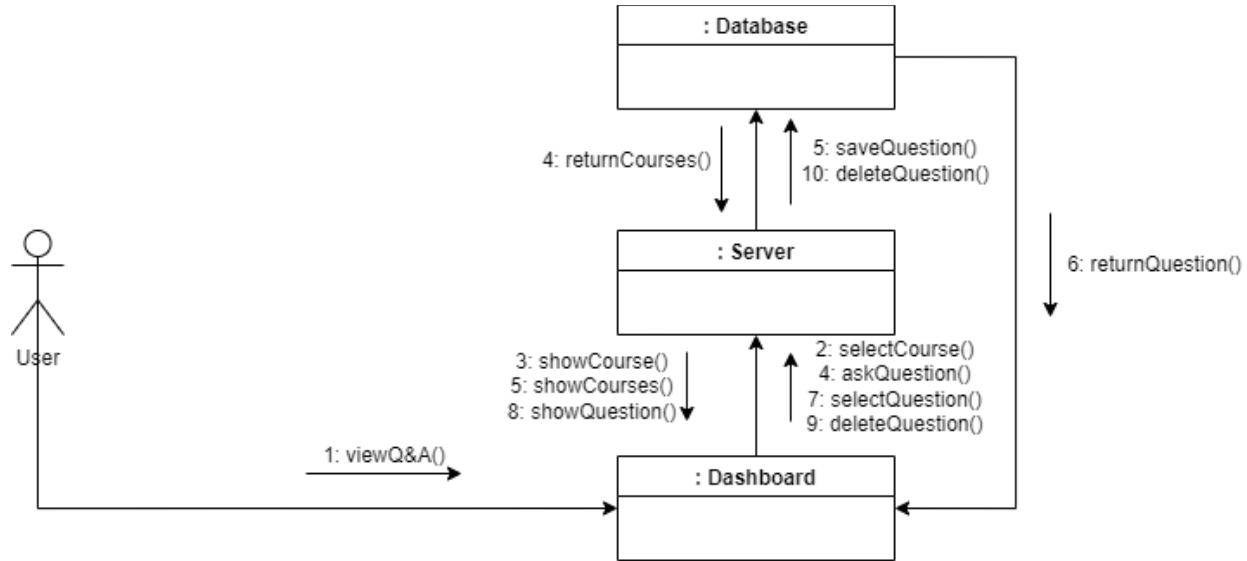
COLLAB_2.1 Manage courses – Take



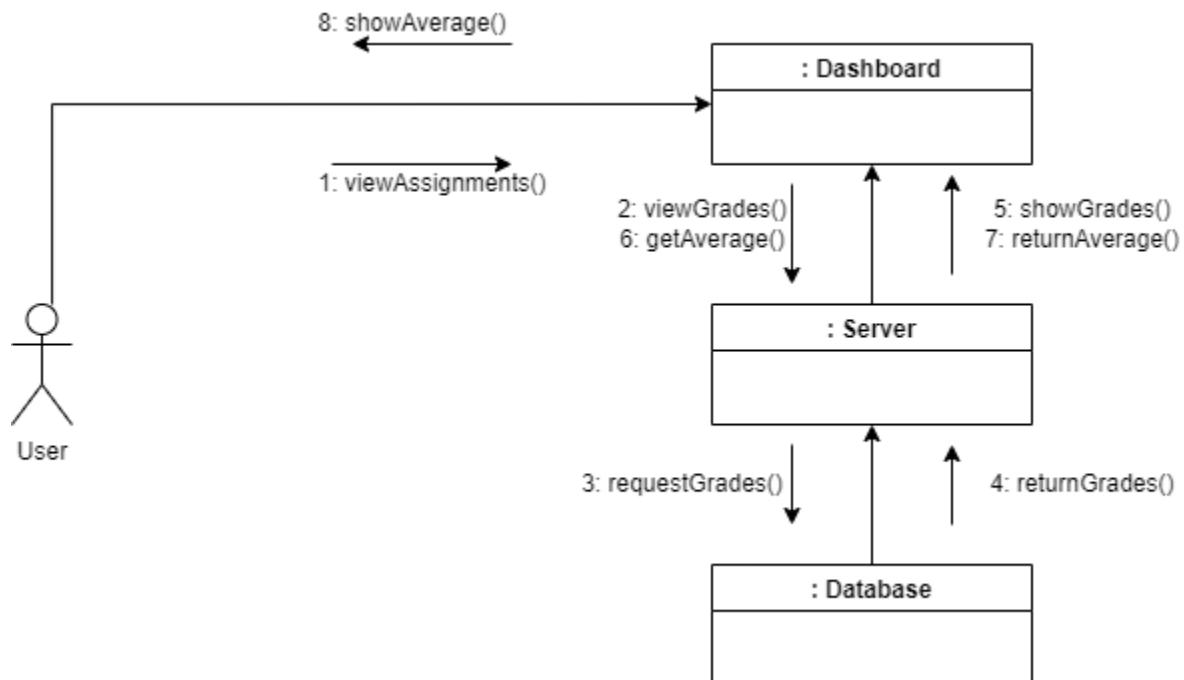
COLLAB_2.1 Manage courses – Cancel



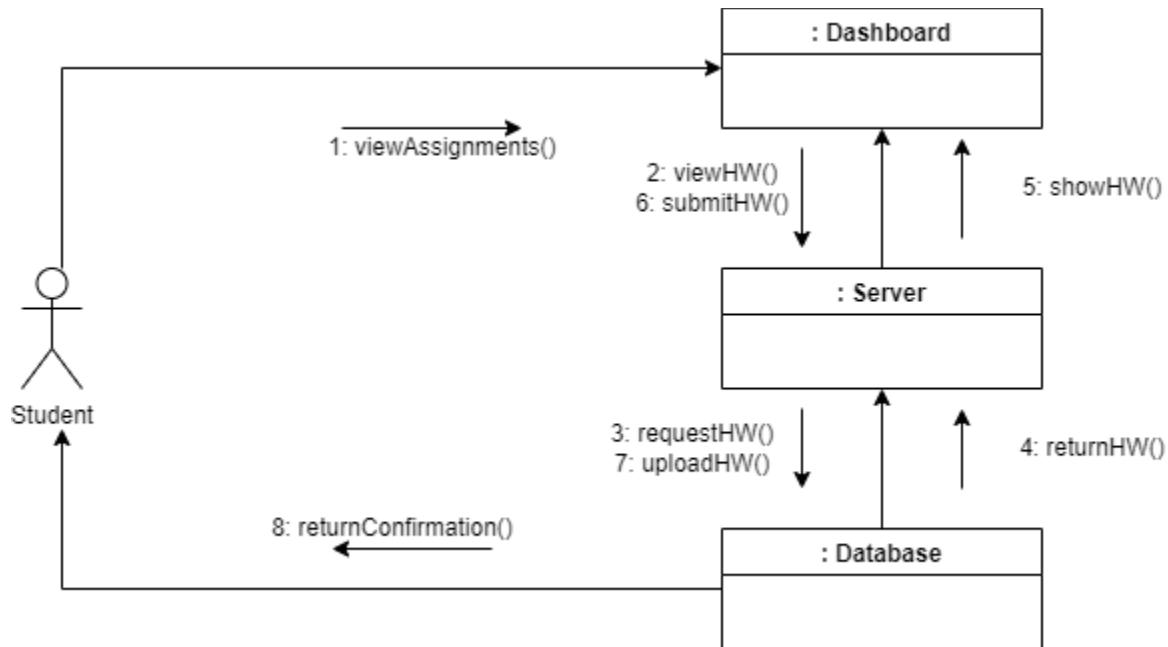
COLLAB_2.2 Ask/Delete question



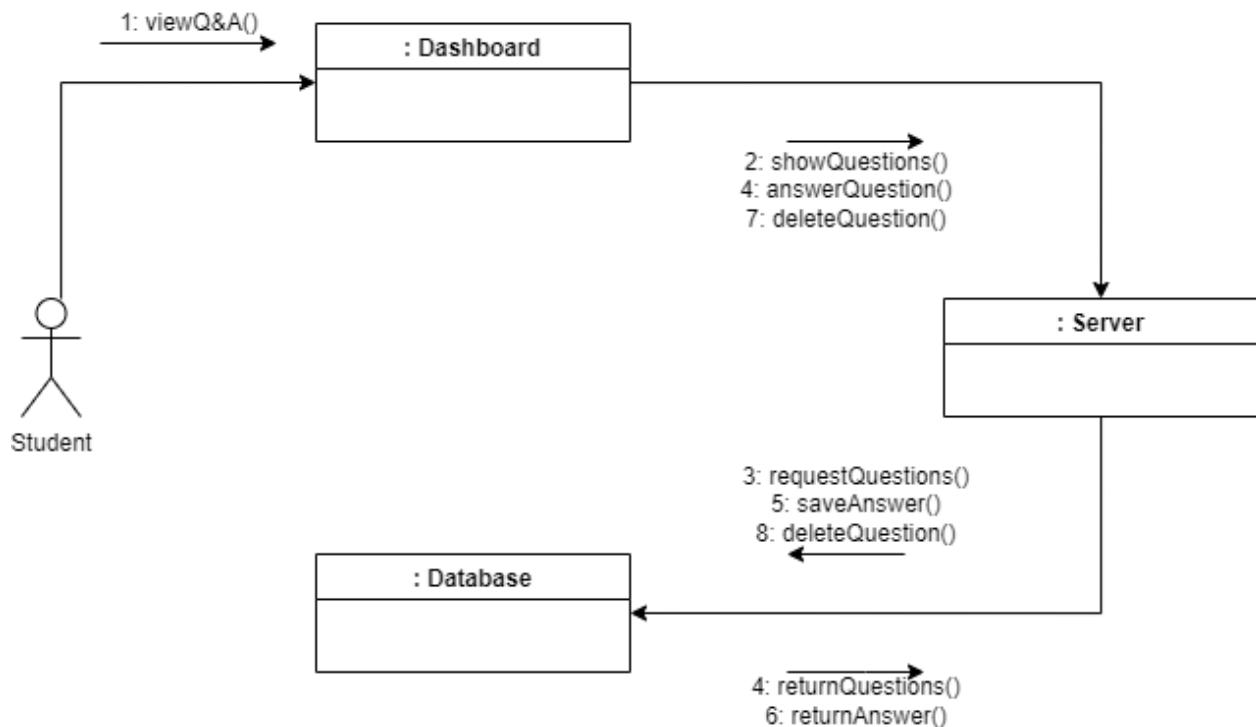
COLLAB_2.3 Get average grade



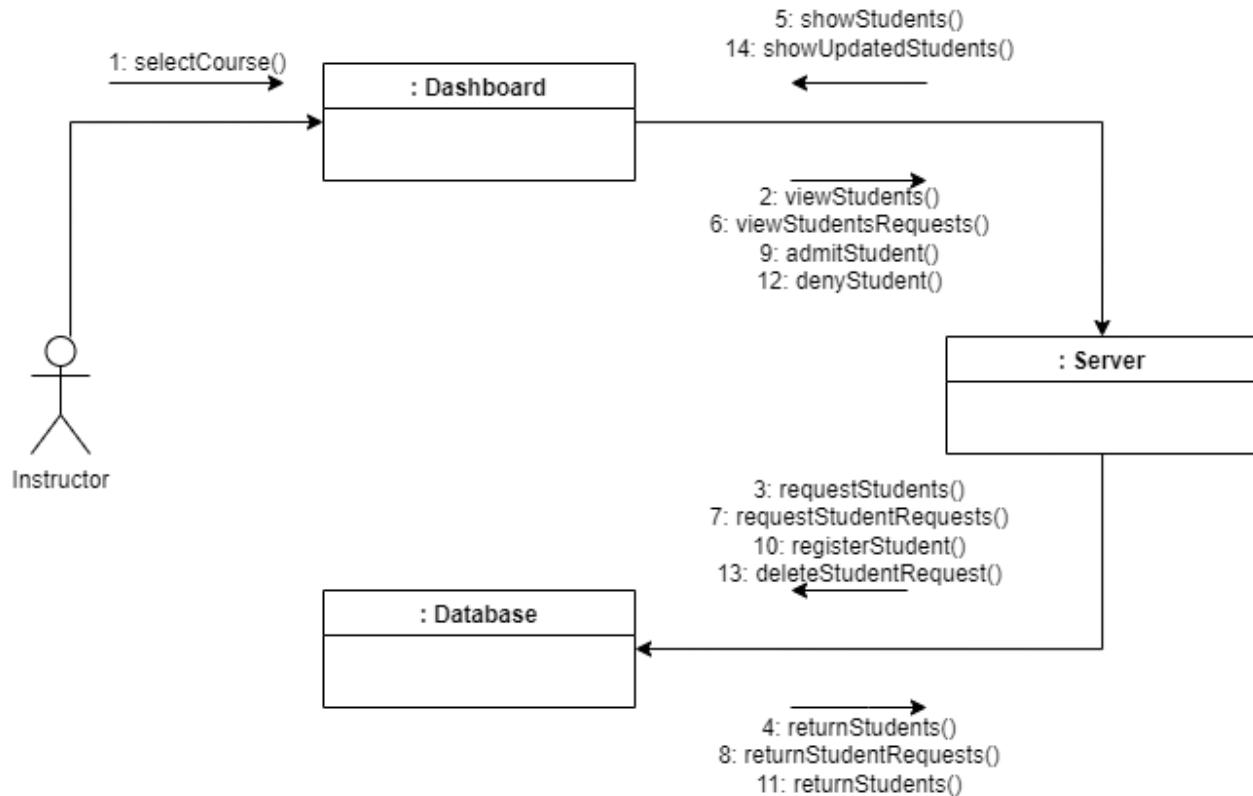
COLLAB_2.4 Submit HW



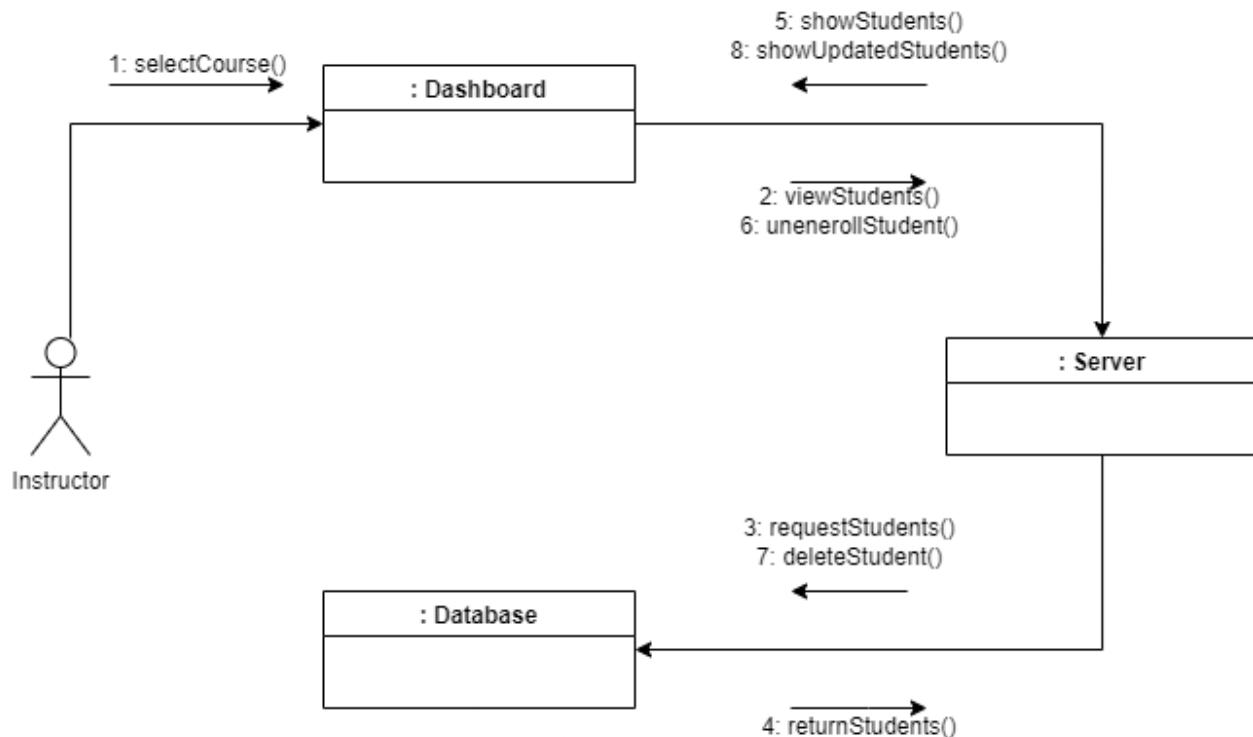
COLLAB_3.1 Manage Q&A



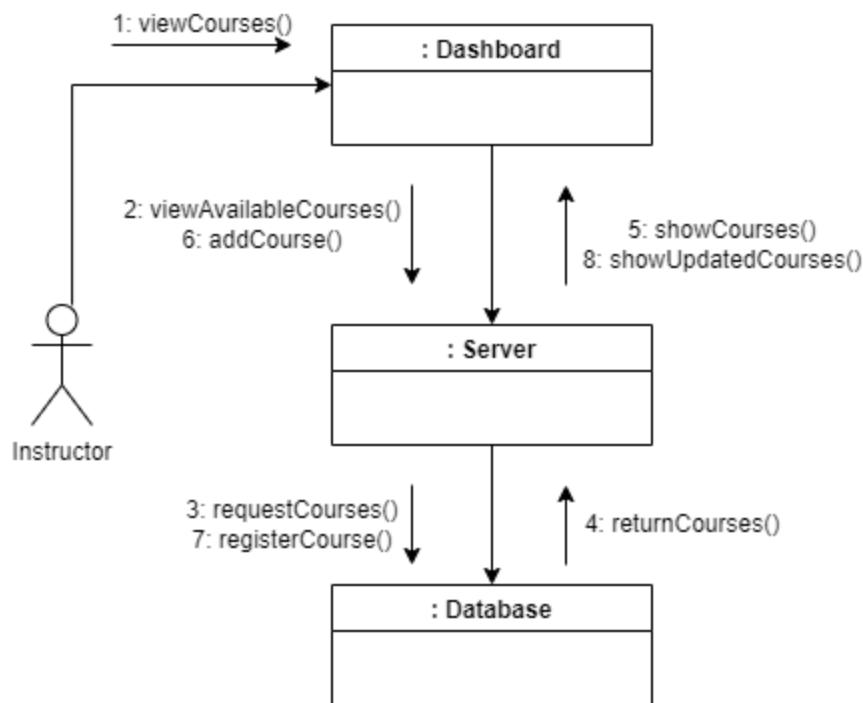
COLLAB_3.2 Manage students – Admit/Deny



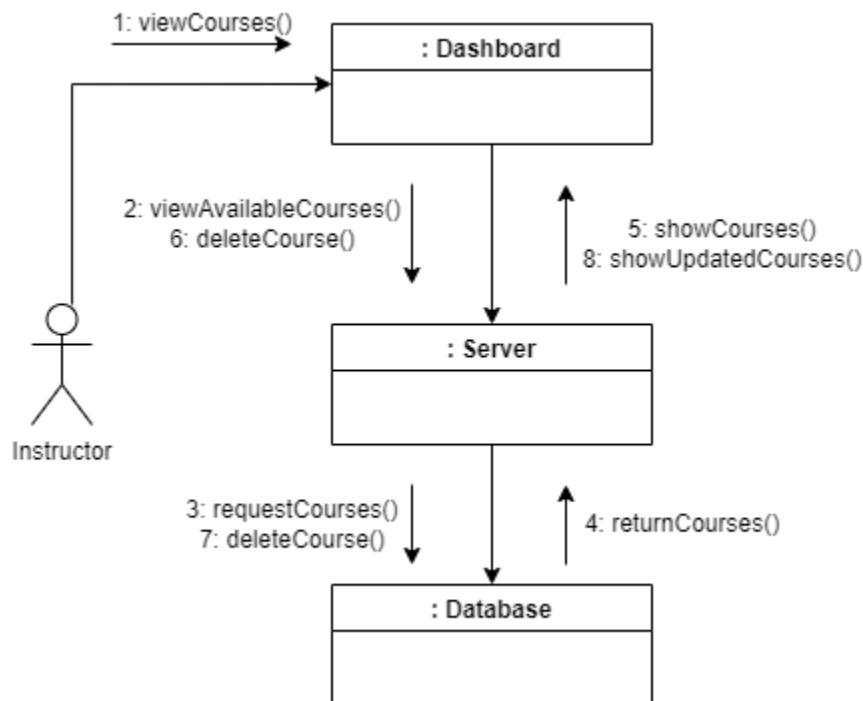
COLLAB_3.2 Manage students – Unenroll



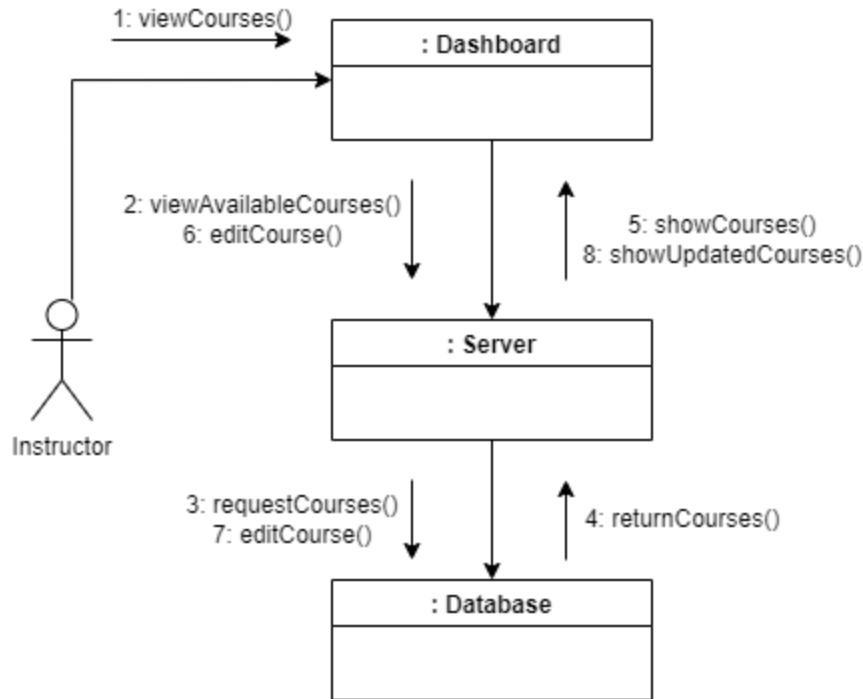
COLLAB_3.3 Manage courses – Add



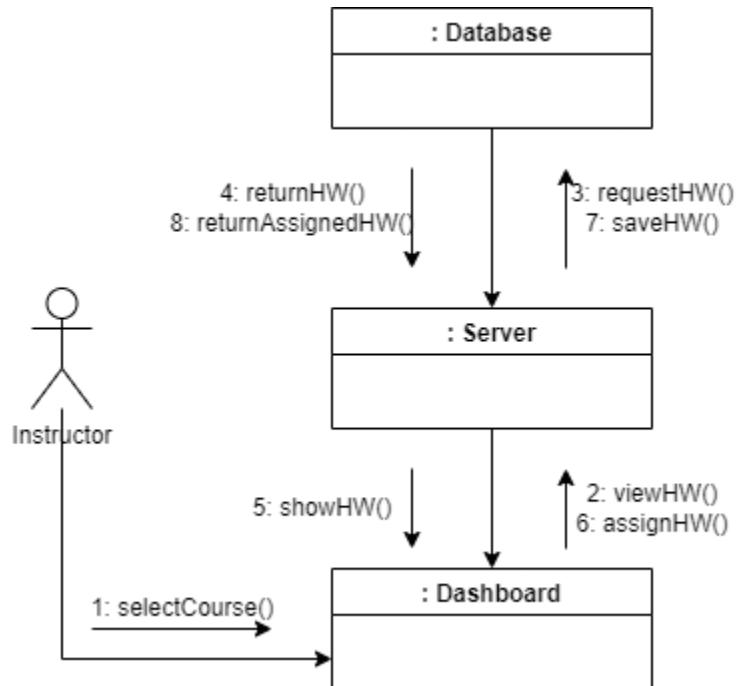
COLLAB_3.3 Manage courses – Delete



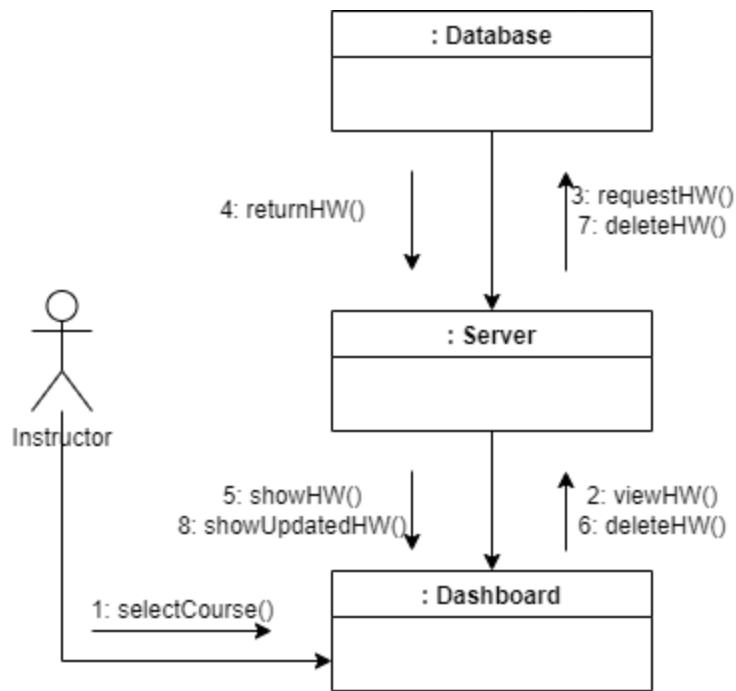
COLLAB_3.3 Manage courses – Edit



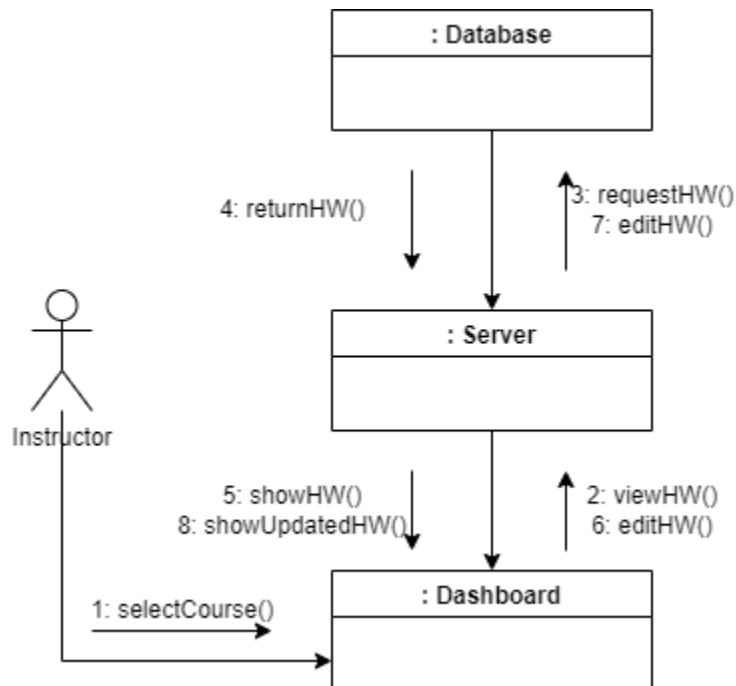
COLLAB_3.4 Manage HW – Assign



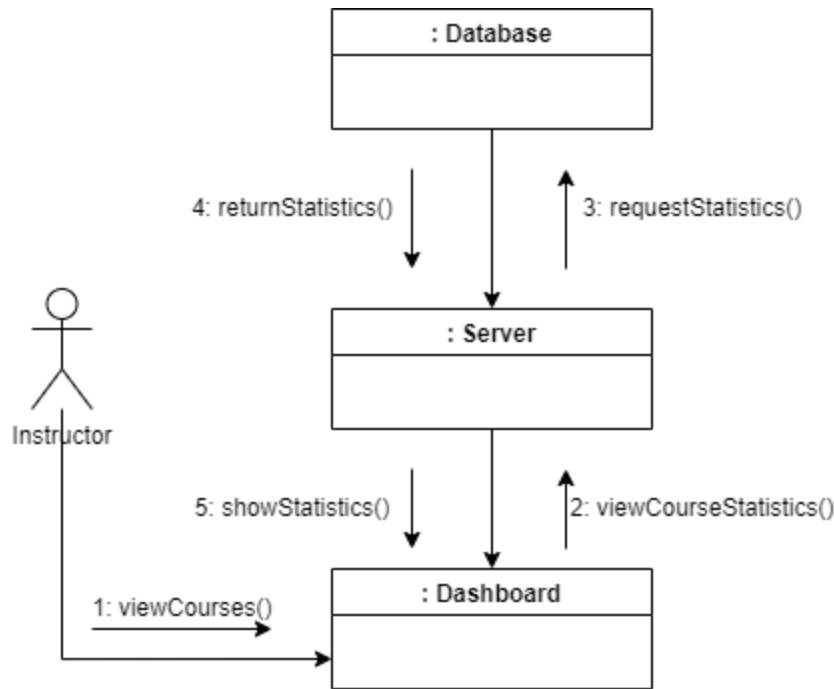
COLLAB_3.4 Manage HW – Delete



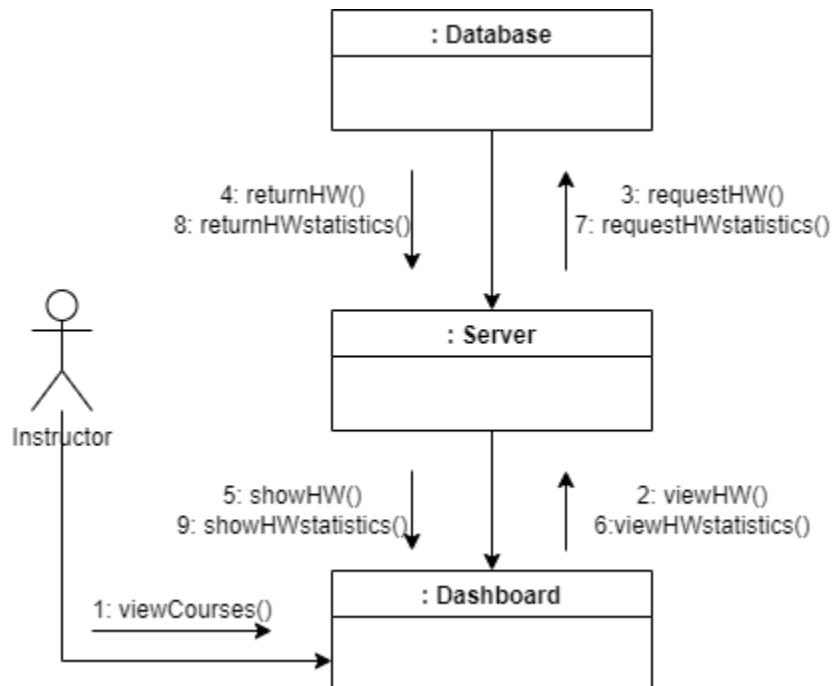
COLLAB_3.4 Manage HW – Edit



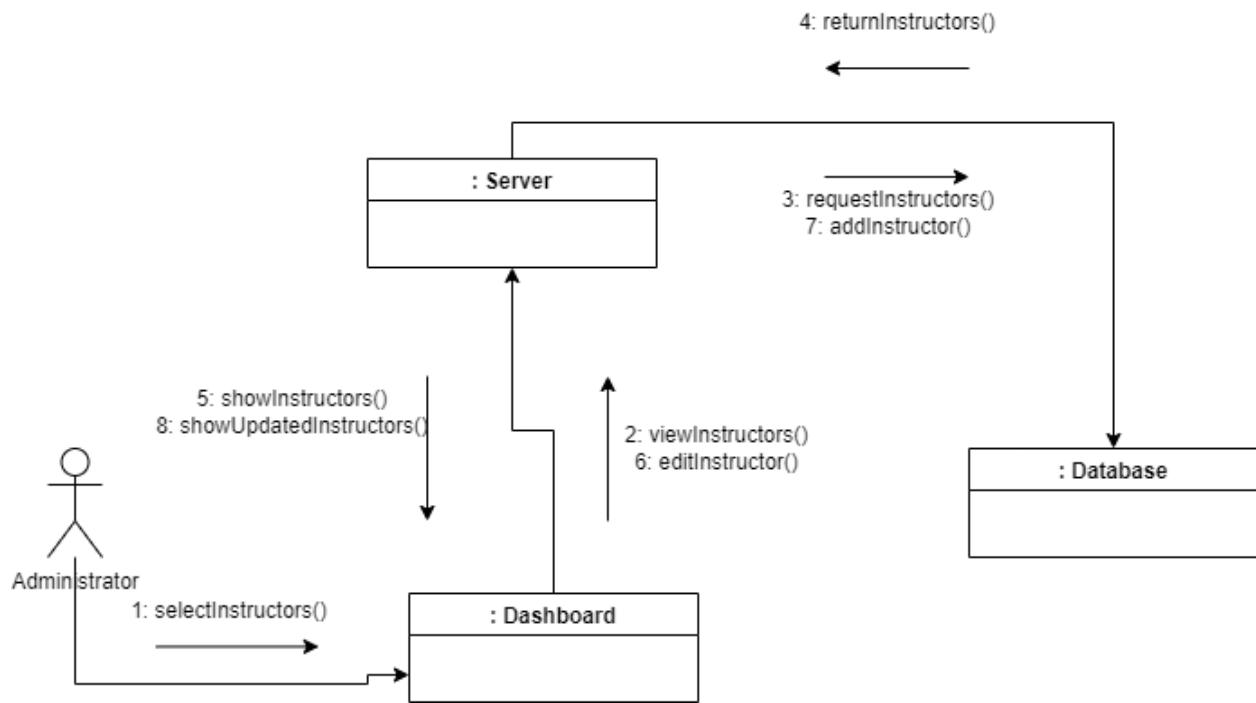
COLLAB_3.5 Check course's statistics



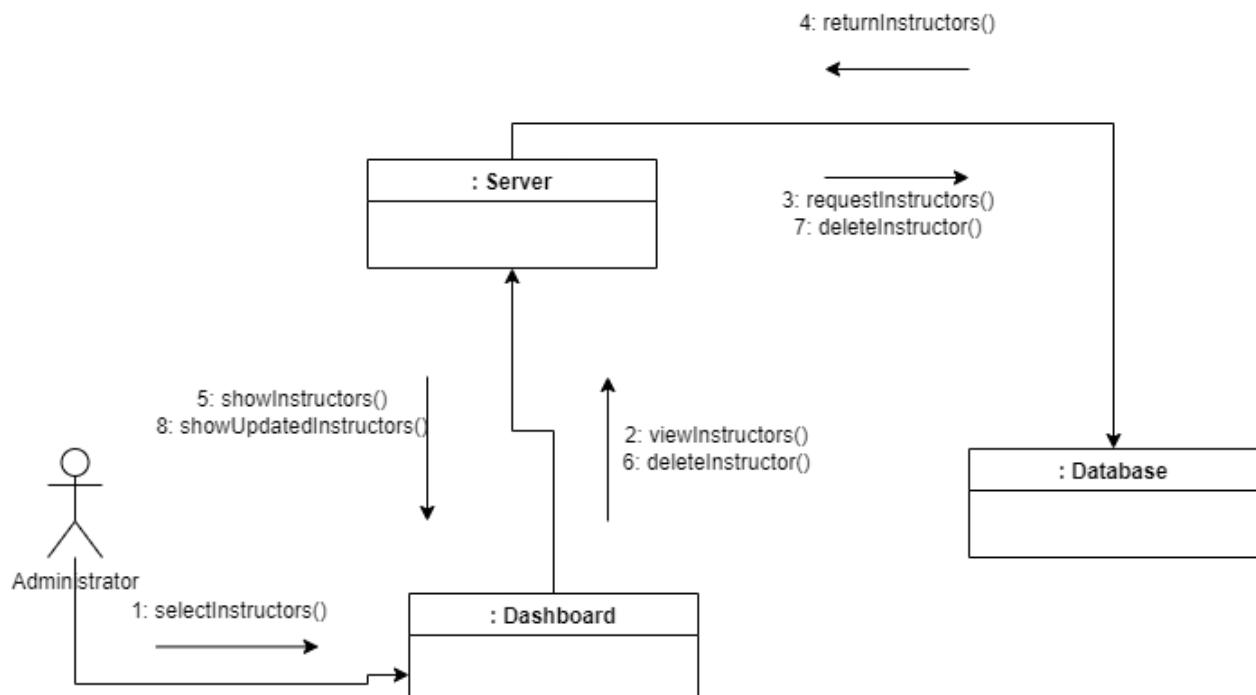
COLLAB_3.6 Check HW statistics



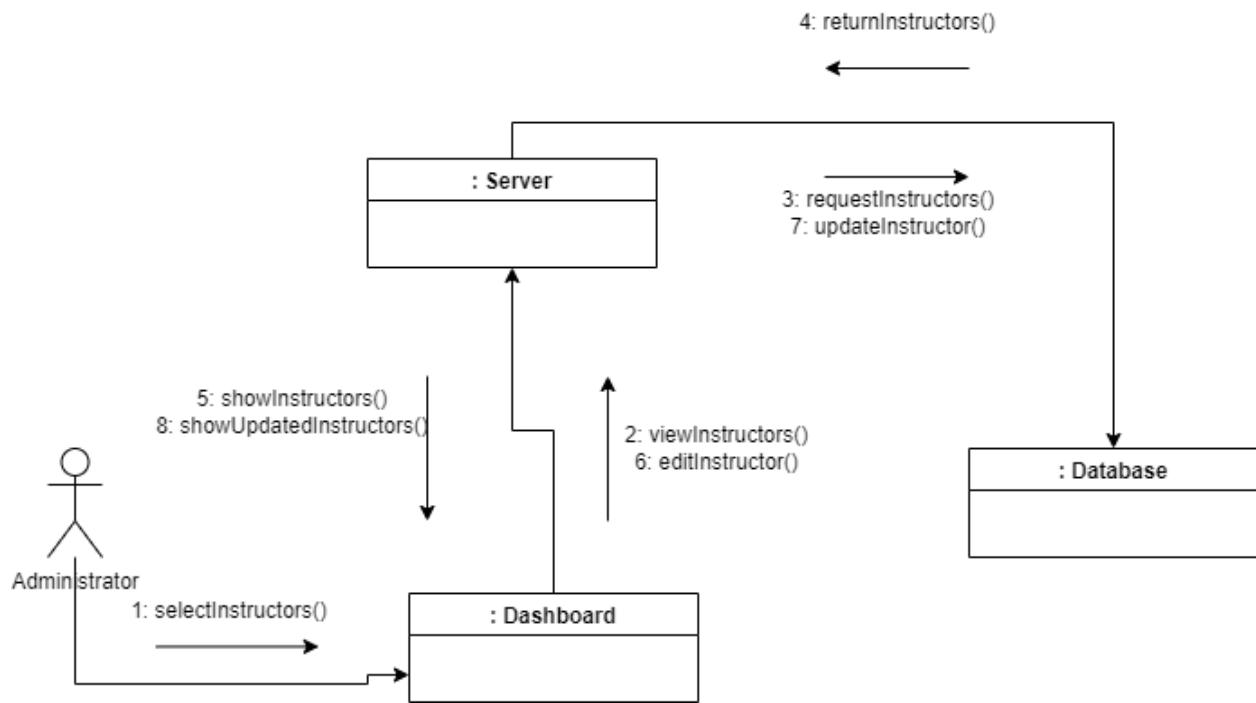
COLLAB_4.1 Manage instructors – Add



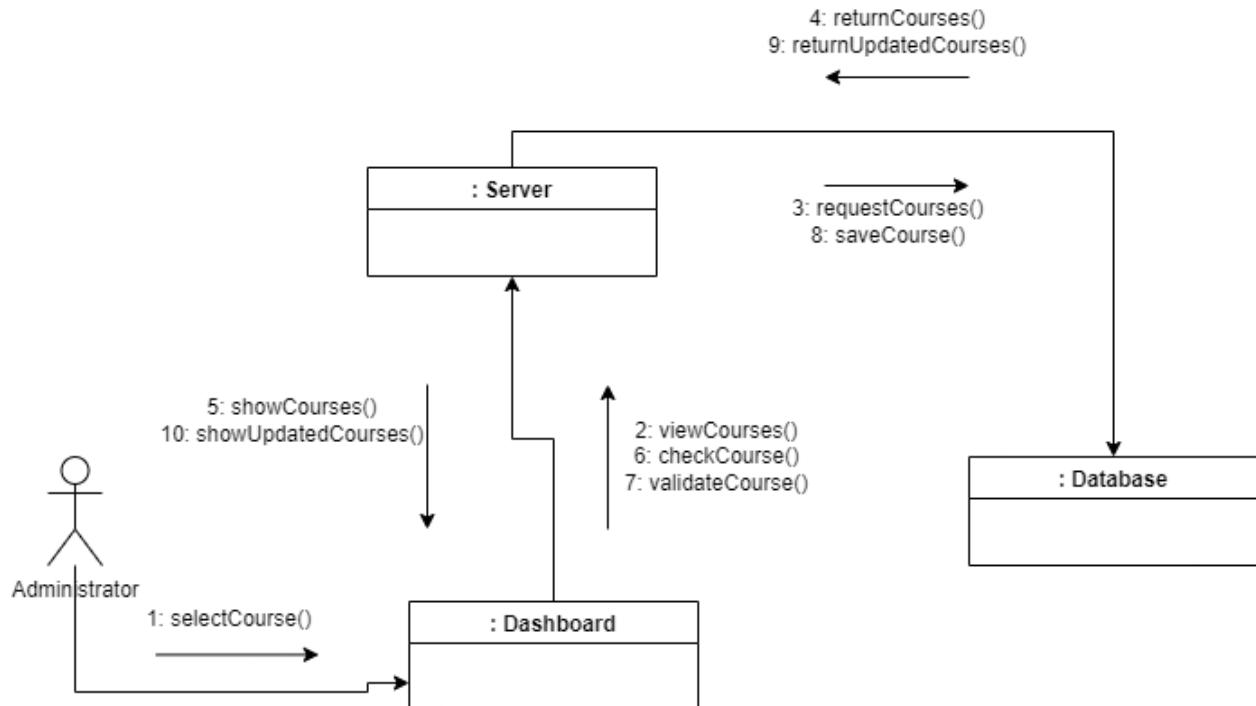
COLLAB_4.1 Manage instructors – Delete



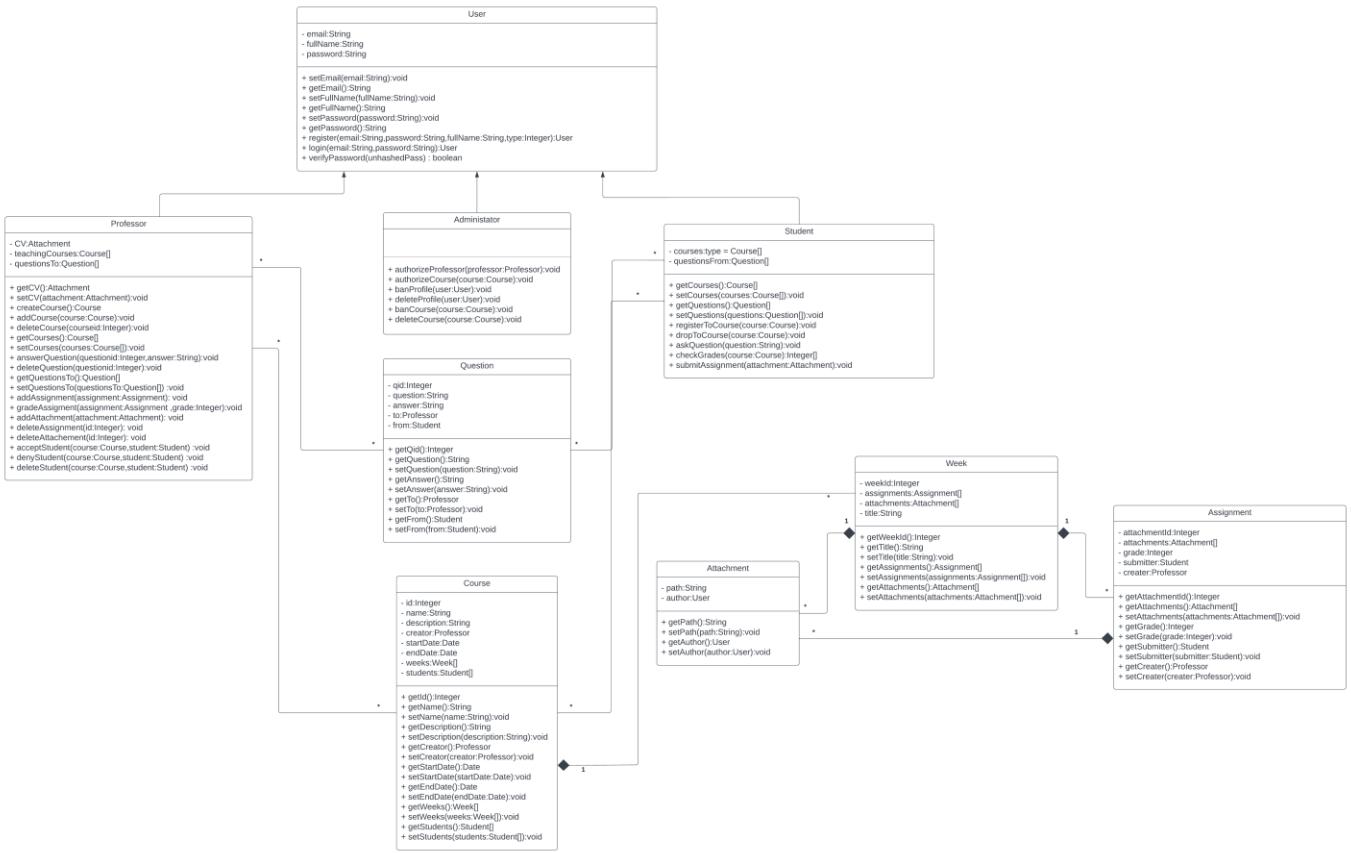
COLLAB_4.1 Manage instructors – Edit



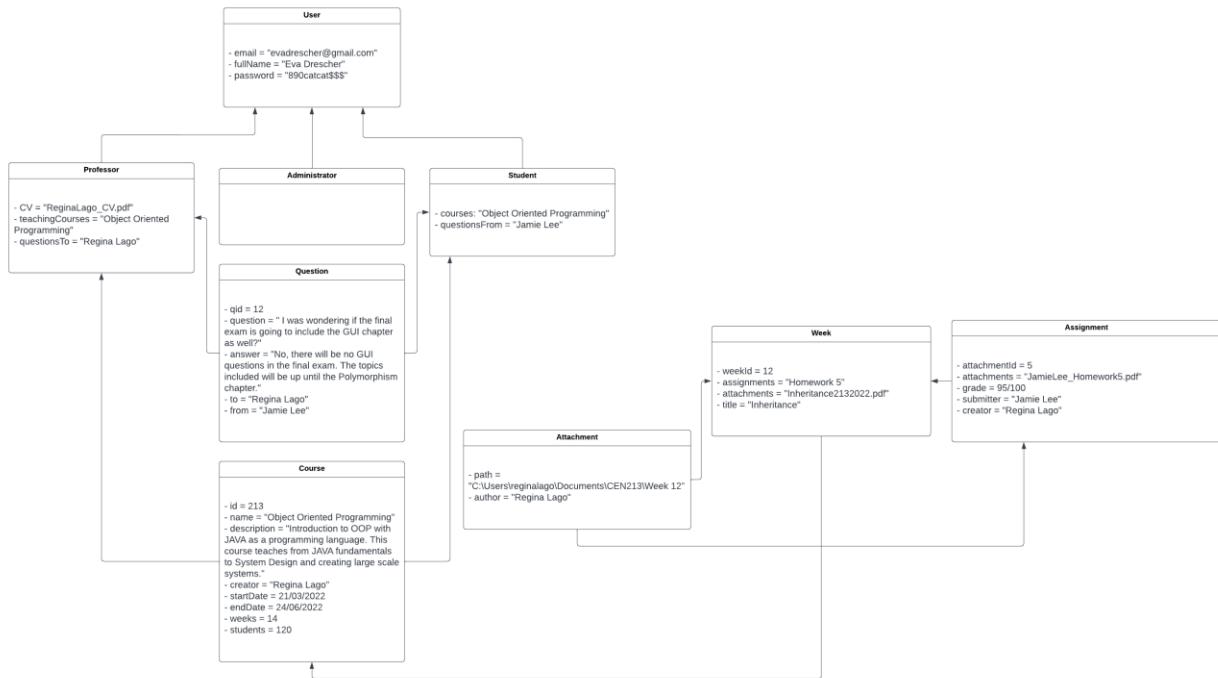
COLLAB_4.2 Validate course



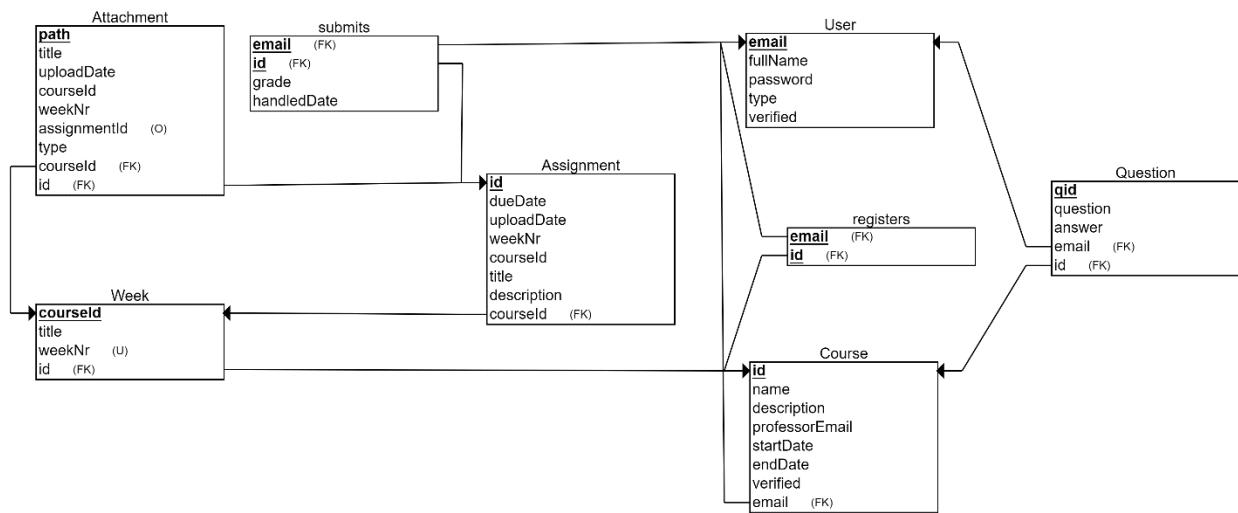
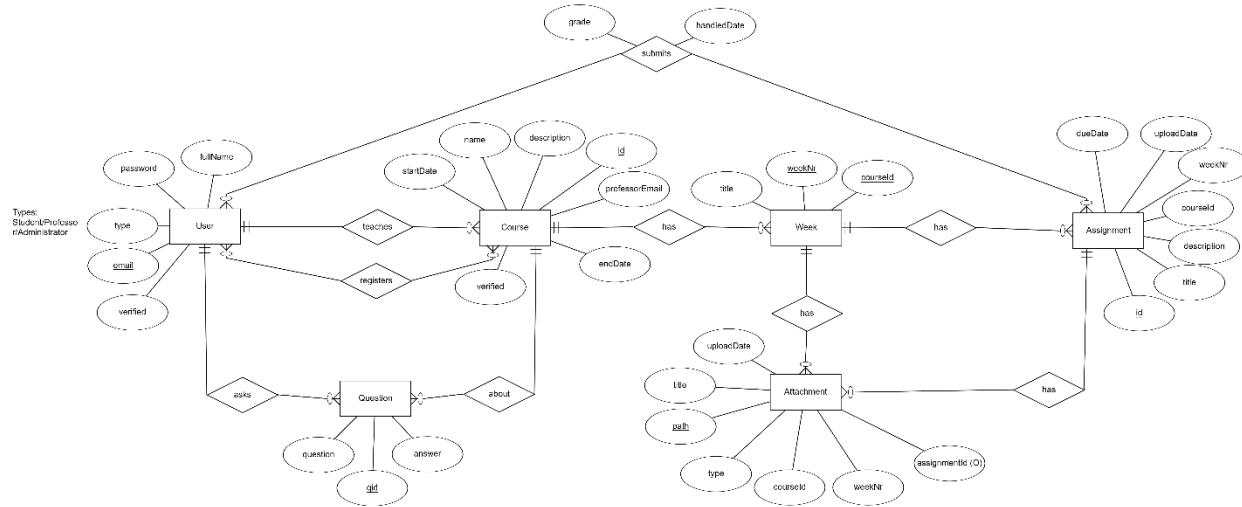
4.2.6 Class Diagram



4.2.7 Object Diagram

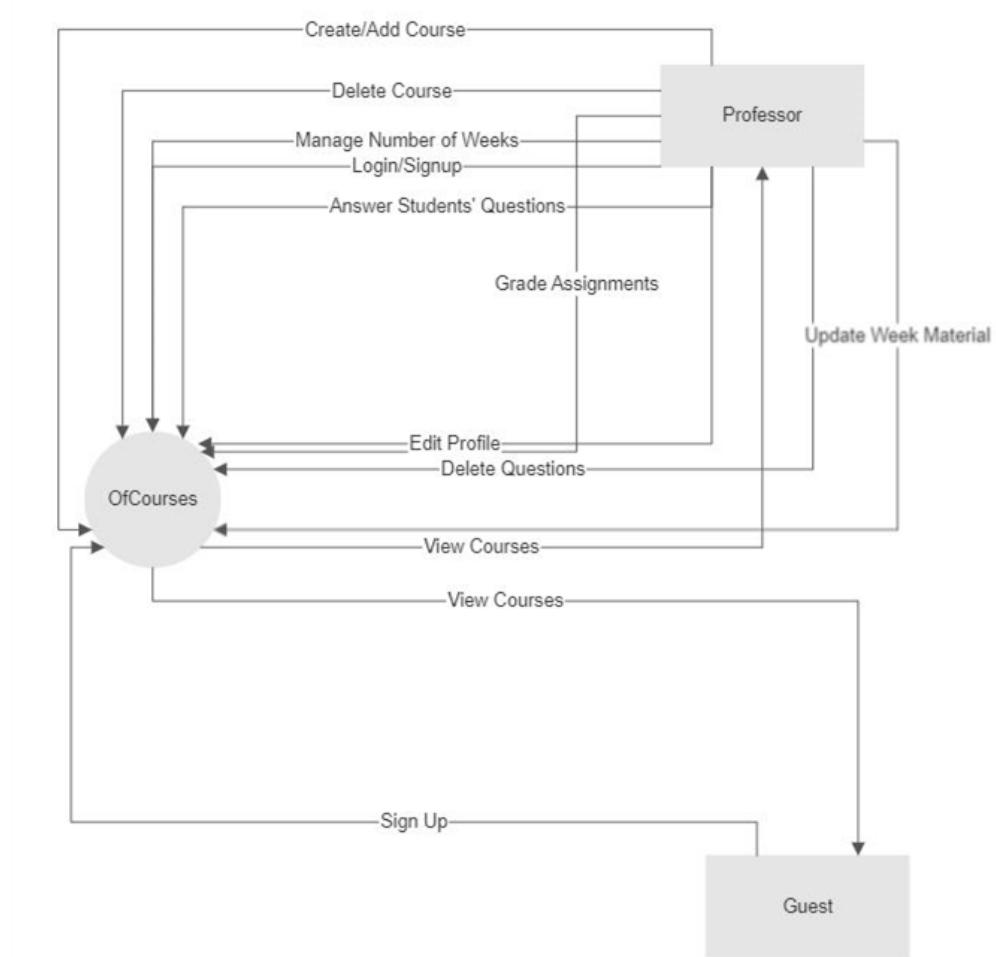


4.2.8 Entity Relation Diagram and Relational Schema

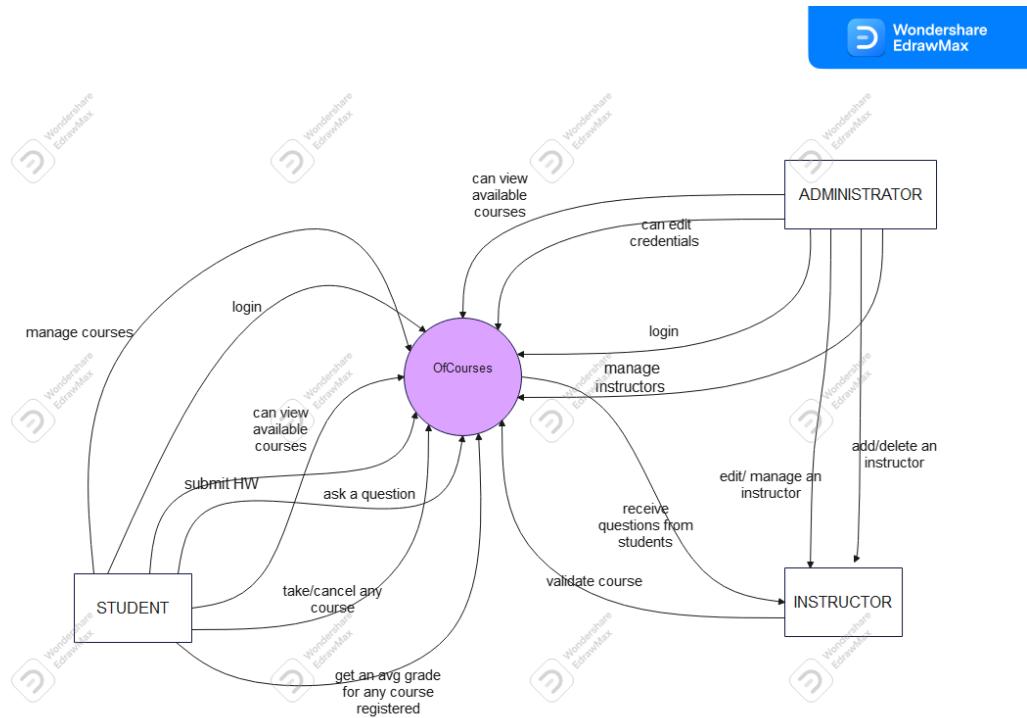


4.2.9 Data Flow Diagrams

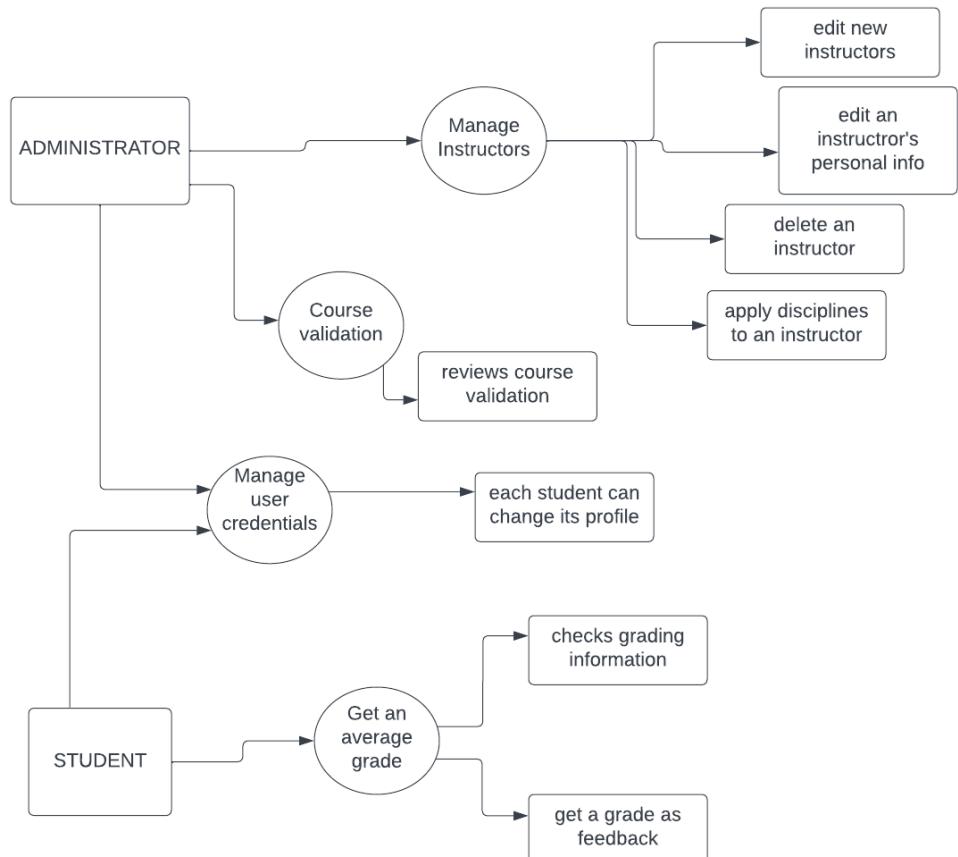
DFD_LVL0_Professor_Guest



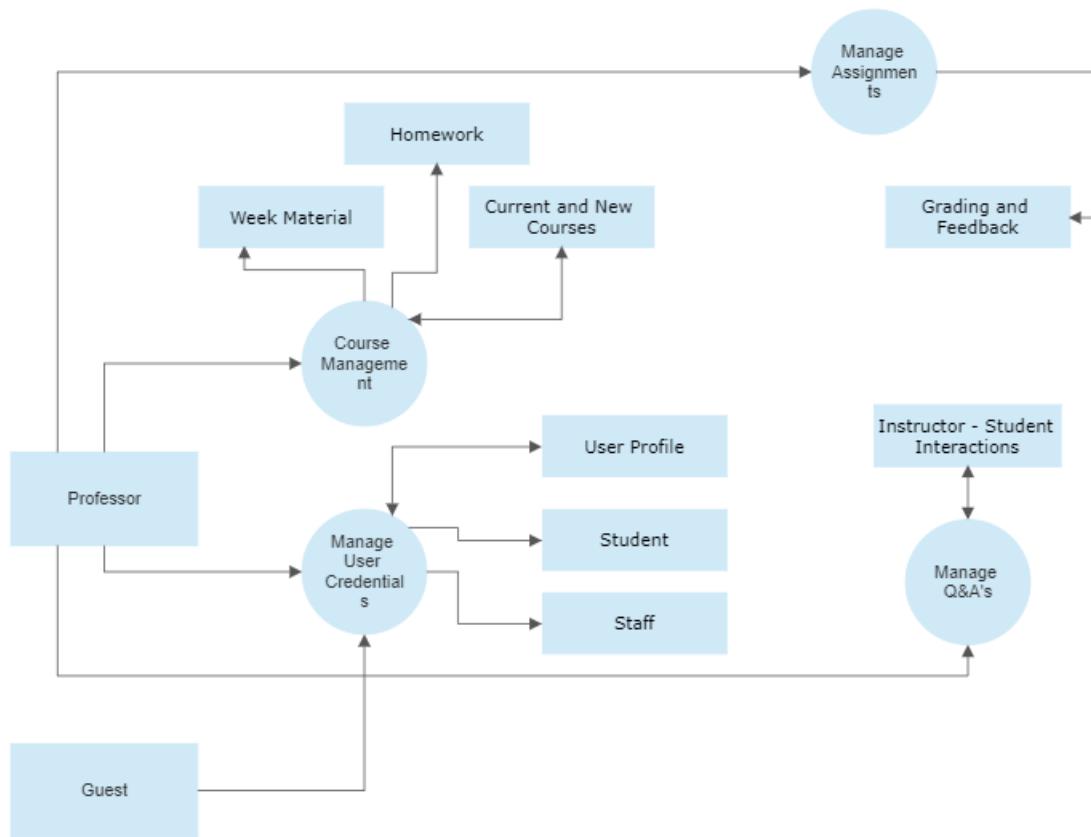
DFD_LVL_0



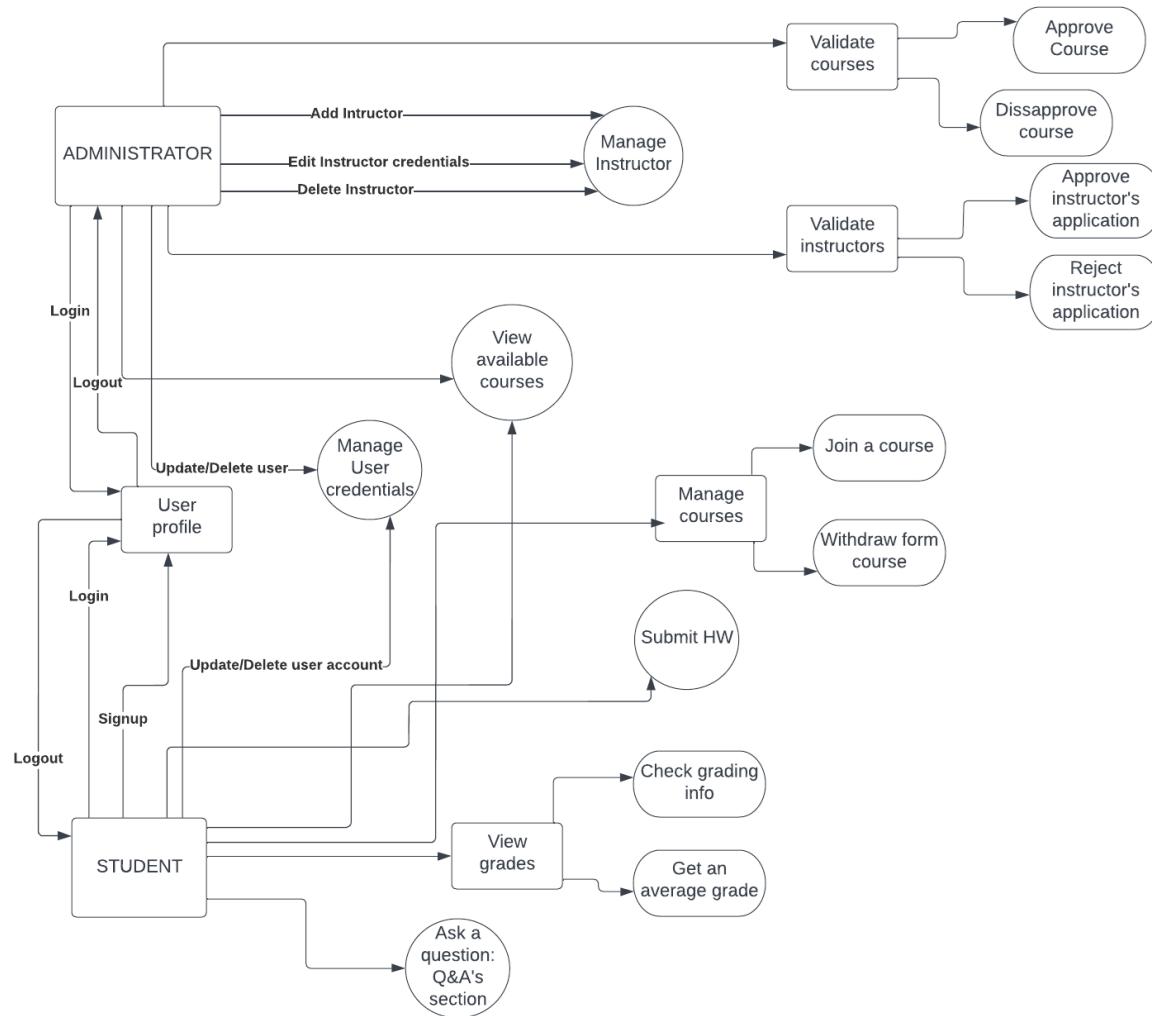
DFD_LVL1_Administrator_Student



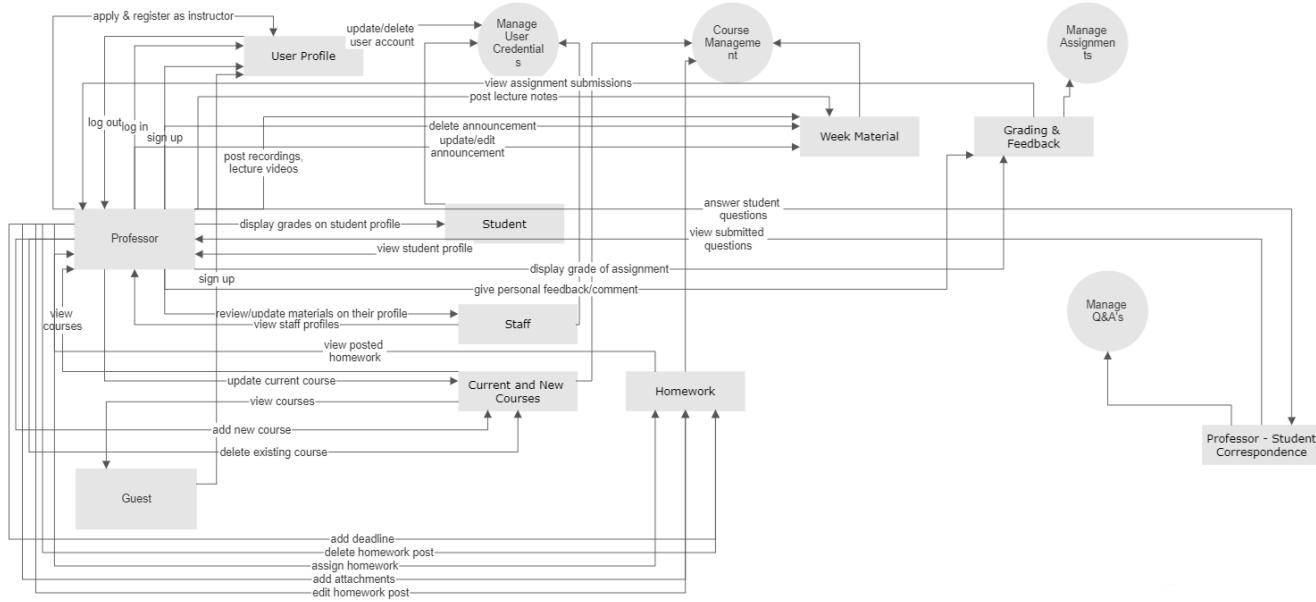
DFD_LVL1_Professor_Guest



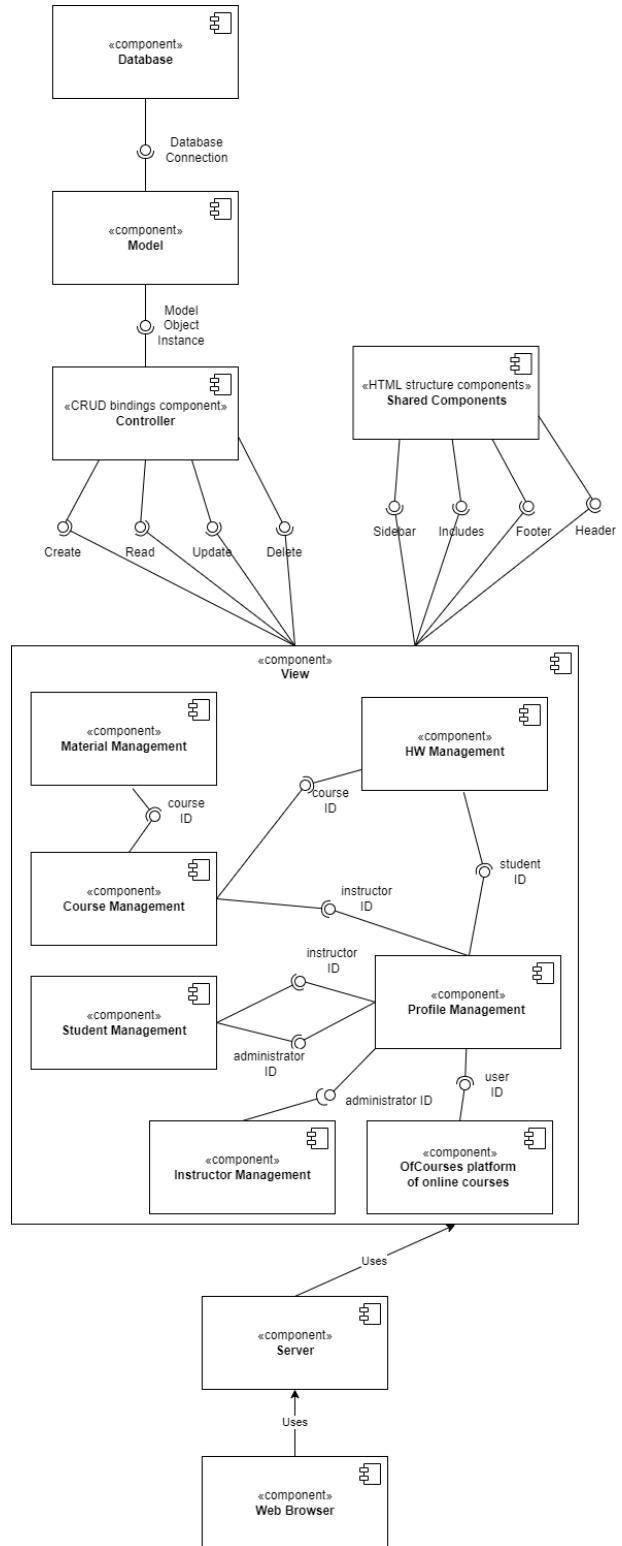
DFD_LVL2_Administrator_Student



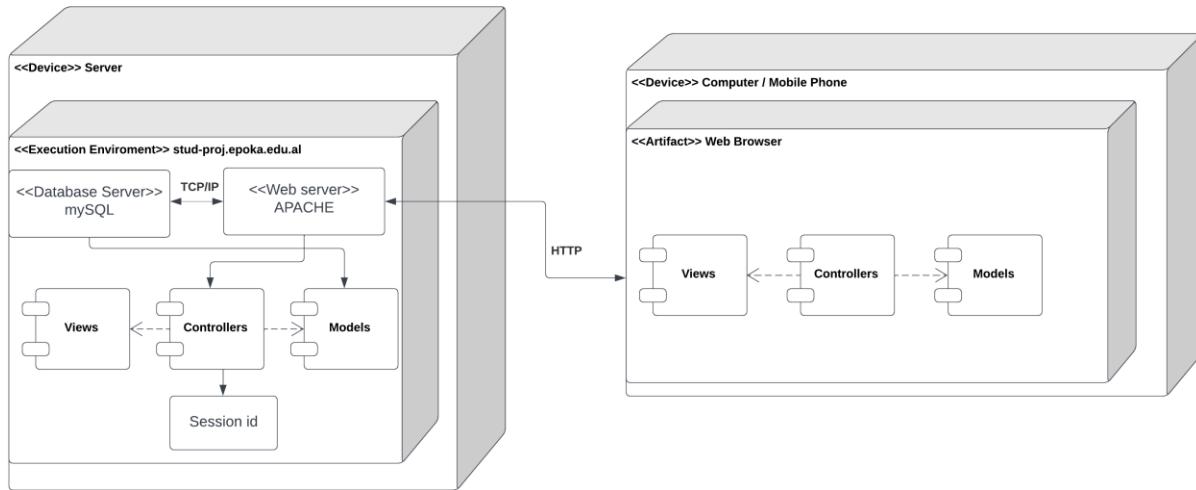
DFD_LVL2_Professor_Guest



4.2.10 Component Diagram



4.2.11 Deployment Diagram



5. Implementation

5.1 Technology Used

5.1.1 Frontend (Client-side development)

These scripting and stylesheet languages are used on the client side of this software:

- HTML5 – A markup language used for structuring and presenting content on the World Wide Web. It is used to create the structure of the front-end.
- CSS – A styling language, used for the designing the pages, layout, fonts, colors.

5.1.2 Backend (Server-side development)

This software's backend is written in vanilla PHP, a quick, versatile, and pragmatic general-purpose scripting language that is particularly well suited to web development. The most widely used PHP development environment is XAMPP. The Apache HTTP Server is utilized as part of XAMPP. Apache is a cross-platform HTTP web server that is used to distribute online content all over the world. Apache's remote server sends the user the requested files, pictures, and other documents. The PDO abstraction layer is used to gain database access from the backend. It has a uniform interface that allows you to access a variety of databases.

PDO advantages:

- Security (prepared statements achieve protection from SQL injections)
- Usability (many helper functions to automate routine operations)
- Reusability (unified API to access multitude of databases)

5.1.3 Database

MySQL, a relational database management system, to be more specific, the open-source MariaDB as part of XAMPP, a derivative of MySQL, is the database utilized by this program. It was the greatest choice for this type of project since it used standard structured query language, was quick, dependable, compliant on a variety of platforms, and was simple to work with.

5.1.4 MVC Architecture

Our approach to utilizing an MVC folder structure for this implementation is:

- **Model:** The model classes' primary role is to communicate with the database directly. The corresponding routines that prepare the query statements are built within each model. These routines return to the controllers the results of database operations. For example, a course object will retrieve the course's information from the database, manipulate it and update it back to database.
- **View:** Our application's front-end is made up of views. These are simple HTML files with user interface components that use CSS to provide an interactive experience. Different UI components in the view fire actions for the controllers to manage.
- **Controller:** To process all incoming requests, apply business logic, modify data using the Model, and render the output into the views, the controller functions as an interface between the Model and View Component.

5.2 Installation Manual

These are the steps you need to follow for the software to be used:

- Make sure to have XAMPP installed on your computer. If not, follow this link to download: [XAMPP](#)
- Fork the GitHub repository: [OfCourses](#), or you can download the zip file.
- The downloaded file should be copied/moved inside the htdocs folder under the XAMPP directory structure.
- Open XAMPP and start MySQL and Apache servers.
- [localhost/phpMyAdmin](#) – copy this link, and run the [SQL starting queries](#), to create the database and all the tables.

5.3 Configuration Manual

Firstly, some development variables need to be configured before starting using the software. These needs to be configured based on your working environment.

In the [db_connection.php](#) file, are some constants that need to be defined to acquire the connection:

```
$servername = 'localhost';  
$username = 'root';  
$password = "";  
$database = "ofcoursesdb";
```

5.4 User Manual

This system can be used by these types of users: Guests, Students, Instructors, Administrators. The respective credentials are listed below:

- Student – email: student@gmail.com password: student1
- Professor – email: professor@gmail.com password: professor1
- Administrator – email: admin@gmail.com password: admin1

5.5 Software Screenshots

5.5.1 Welcome page

Welcome to OfCourses!

"To know, is to know that you know nothing. That is the meaning of true knowledge." - Socrates.

5.5.2 View available courses

Code With Us!
Introduction to programming languages. This course teaches the fundamentals of the most common languages.

Web Design Development.
Learn how to use HTML, CSS, Javascript and PHP to build a website from scratch.

Get started with Data Analysis.
Dive into the implementation of Python, Java and SQL in the major data analysis methods.

Digital Media and Mobile Applications.
Learn about the different alternatives of development approaches when building mobile applications.

Introduction basics to Cyber Security.

5.5.3 Contact us

OfCourses

Log In Sign Up About Contact Us



[Talk to us on multiple platforms.](#)



@OfCourses



@OfCourses_official



@OfCourses_Page

5.5.4 Edit profile

OfCourses

My Courses My Account Log Out



Welcome to your account!



Name Surname

Full Name:

Email:

Password:

Submit Changes.

5.5.5 Signup

Signup Here

Name
student

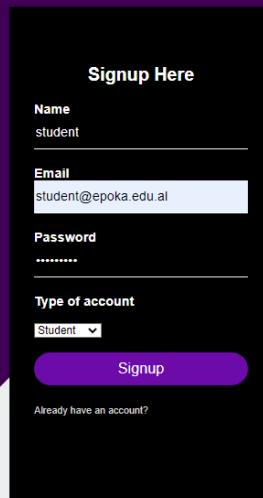
Email
student@epoka.edu.ai

Password
.....

Type of account
Student

Signup

Already have an account?



5.5.6 Login

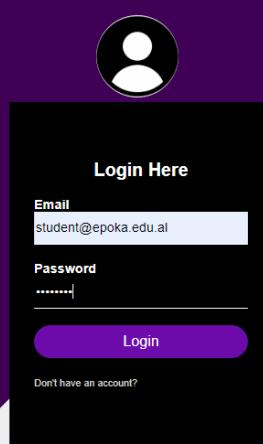
Login Here

Email
student@epoka.edu.ai

Password
.....

Login

Don't have an account?



5.5.7 Student welcoming page

OfCourses

My Courses My Account Log Out



Welcome, student!

5.5.8 Student enrolled courses

OfCourses

My Courses My Account Log Out



Browse your courses!

Parallel Programming

This is a course given at Epoka University. This is a course given at Ep...

Enroll Ongoing: 0 enrolled students

5.5.9 Student view course

OfCourses

Software Engineering

Number of enrolled students: 1
Number of weeks: 14

Course Description:
This Course introduces the concepts, tools and techniques of Software Engineering. It emphasizes the development of reliable and maintainable software via system requirements and specifications, software analysis and design methodologies including object-oriented design, implementation, integration and testing, software project management, life-cycle documentation. Students undertake a semester group mini-project as a major part of the course.

Week 1

Attachments: [Lecture 10](#) [Lecture 11](#) [Lecture 12](#)

Assignments: [Assignment 1](#)

5.5.10 Student view assignment

OfCourses

Assignment 1

Due date: 2022-06-17

Assignment Description:
You should submit Personal Submission Files in accordance with the template provided to you. Please remember to grade all members but NOT YOURSELF. The leader should provide an opinion on all members

Attachments:
[Assignment 10](#)

Solution file: Administato... r profiles.png

5.5.11 Student check grade

OfCourses

Assignment 1

Due date: 2022-06-17

Assignment Description:
You should submit Personal Submission Files in accordance with the template provided to you. Please remember to grade all members but NOT YOURSELF. The leader should provide an opinion on all members

Attachments:
[Assignment 10](#)

Your grade: 91

5.5.12

5.5.13 Student ask the professor

OfCourses

My Courses My Account Log Out

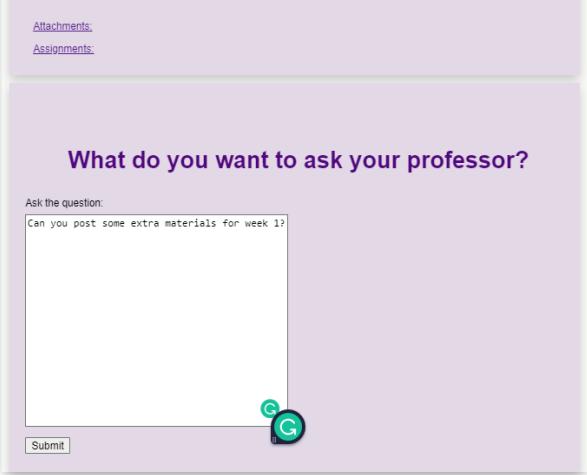
Attachments:
Assignments:

What do you want to ask your professor?

Ask the question:

Can you post some extra materials for week 1?

Submit G



5.5.14 Professor welcoming page

OfCourses

Manage Courses Ask the Professor My Profile Log Out



5.5.15 Professor manage courses

OfCourses

Manage Courses Ask the Professor My Profile Log Out

Add New Course

[Software Engineering](#)

This Course introduces the concepts, tools and techniques of Software Engineering. It emphasizes the development of reliable and maintainable software via system requirements and specifications, soft...

Delete Completed: 0 enrolled students



5.5.16 Professor add a course

The screenshot shows a user interface for adding a course. At the top, there is a navigation bar with the 'OfCourses' logo and four buttons: 'Manage Courses', 'Ask the Professor', 'My Profile', and 'Log Out'. Below the navigation bar, there is a large input field labeled 'Lecture 1'. Underneath the input field is a button labeled 'Choose Files' followed by the text '3 files'. At the bottom of the input field area are two buttons: 'Submit' and 'Add Attachment'.

5.5.17 Professor edit course

The screenshot shows a course management interface for 'Software Engineering'. At the top, there is a navigation bar with the 'OfCourses' logo and four buttons: 'Manage Courses', 'Ask the Professor', 'My Profile', and 'Log Out'. The main title of the course is 'Software Engineering'. Below the title, there are two pieces of information: 'Number of enrolled students: 0' and 'Number of weeks: 14'. Underneath this information is a section titled 'Course Description:' with a detailed description of the course. At the bottom of the page is a modal window titled 'Week 1' with an 'Okay' button. Inside the modal, there is a list of attachments: 'Assignment 1.0', 'Lecture 10', 'Lecture 11', and 'Lecture 12'. Below the attachments is a list of assignments: 'Assignment 1'. At the bottom of the modal are two buttons: 'Add assignment' and 'Add attachment'.

5.5.18 Professor add attachment

OfCourses

Manage Courses Ask the Professor My Profile Log Out

Software Engineering

Course Description:
maintainable software via system requirements and specifications, software analysis and design methodologies including object-oriented design, implementation, integration and testing, software project management, life-cycle documentation. Students undertake a semester group mini-project as a major part of the course.

Start Date: 31-Mar-2022

End Date: 17-Jun-2022

Number of weeks: 14

5.5.19 Professor add assignment

OfCourses

Manage Courses Ask the Professor My Profile Log Out

Assignment Title: Assignment 1

Due: 17-Jun-2022

Assignment Description:
You should submit Personal Submission Files in accordance with the template provided to you. Please remember to grade all members but NOT YOURSELF. The leader should provide an opinion on all members and their work, while other team members only need to speak of their own work: however they do have to put any comments on their grading

Class Diagram.png

5.5.20 Professor grade assignment

OfCourses

Manage Courses Ask the Professor My Profile Log Out

Assignment 1

Due date: 2022-06-17

Assignment Description:

You should submit Personal Submission Files in accordance with the template provided to you. Please remember to grade all members but NOT YOURSELF. The leader should provide an opinion on all members

Attachments:

[Assignment 10](#)

| Student | Submission | Grade | |
|---------|---|---------------------------------|-----------------------------------|
| student | ./Attachments/61655376434.png | <input type="text" value="91"/> | <input type="button" value="OK"/> |

5.5.21 Professor answer question

OfCourses

Manage Courses Ask the Professor My Profile Log Out

Q&A Forum: Ask the Professor!

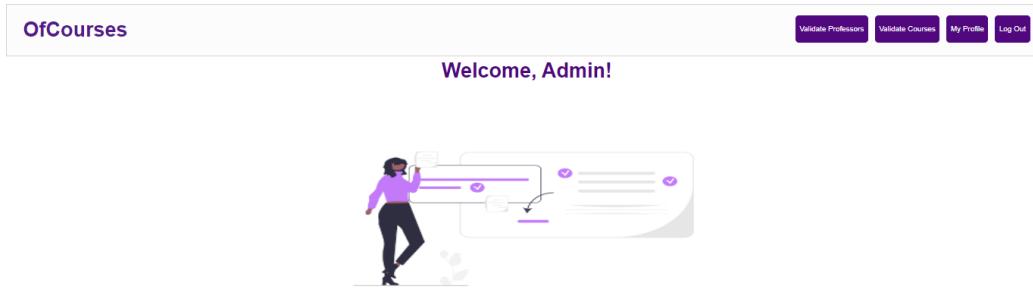
Question from student from Course
"Software Engineering":

Q: Can you post some extra materials for week 1?

Yes, of course.

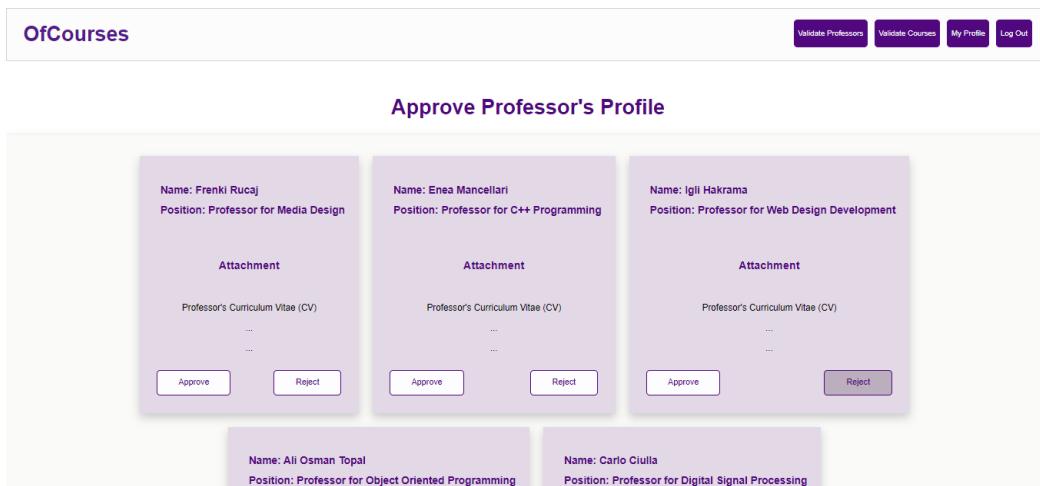


5.5.22 Administrator welcoming page



The screenshot shows the administrator's welcome page. At the top, there is a navigation bar with the 'OfCourses' logo and links for 'Validate Professors', 'Validate Courses', 'My Profile', and 'Log Out'. Below the navigation bar, the text 'Welcome, Admin!' is displayed. A decorative illustration of a person standing next to a large document or interface is visible.

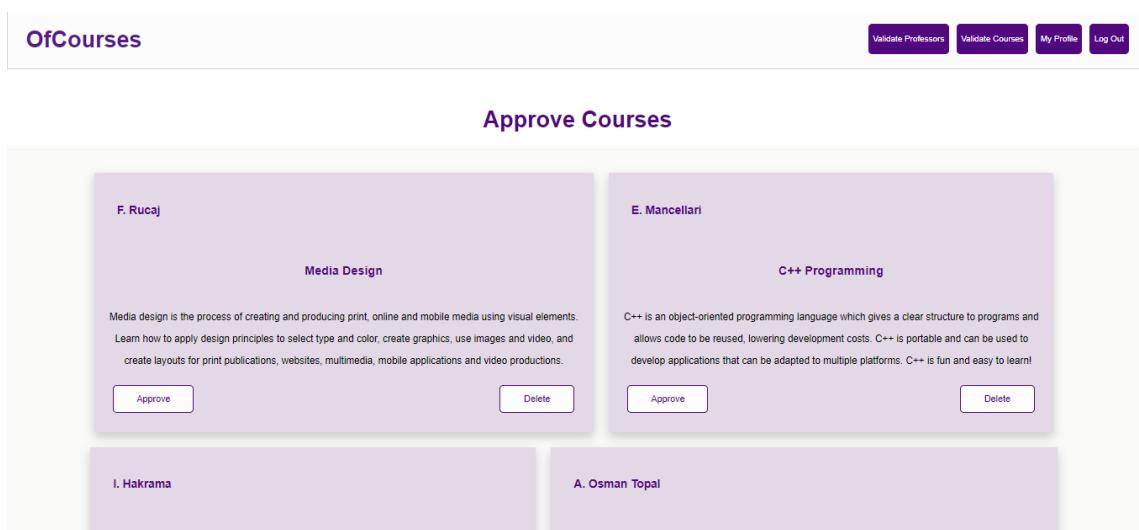
5.5.23 Administrator approve professor



The screenshot shows the 'Approve Professor's Profile' section. It displays three professor profiles with their names and positions, attachments (CVs), and approval/rejection buttons. Below this, there are two more profiles partially visible.

| Name | Position | Attachment | Action |
|-----------------|---|-----------------------------------|----------------|
| Frenki Rucaj | Professor for Media Design | Professor's Curriculum Vitae (CV) | Approve Reject |
| Enea Mancellari | Professor for C++ Programming | Professor's Curriculum Vitae (CV) | Approve Reject |
| Igli Hakrama | Professor for Web Design Development | Professor's Curriculum Vitae (CV) | Approve Reject |
| Ali Osman Topal | Professor for Object Oriented Programming | | |
| Carlo Ciulla | Professor for Digital Signal Processing | | |

5.5.24 Administrator approve course



The screenshot shows the 'Approve Courses' section. It displays four course profiles with their names and descriptions, along with approval/rejection buttons.

| Name | Description | Action |
|----------------|-----------------|----------------|
| F. Rucaj | Media Design | Approve Delete |
| E. Mancellari | C++ Programming | Approve Delete |
| I. Hakrama | | |
| A. Osman Topal | | |

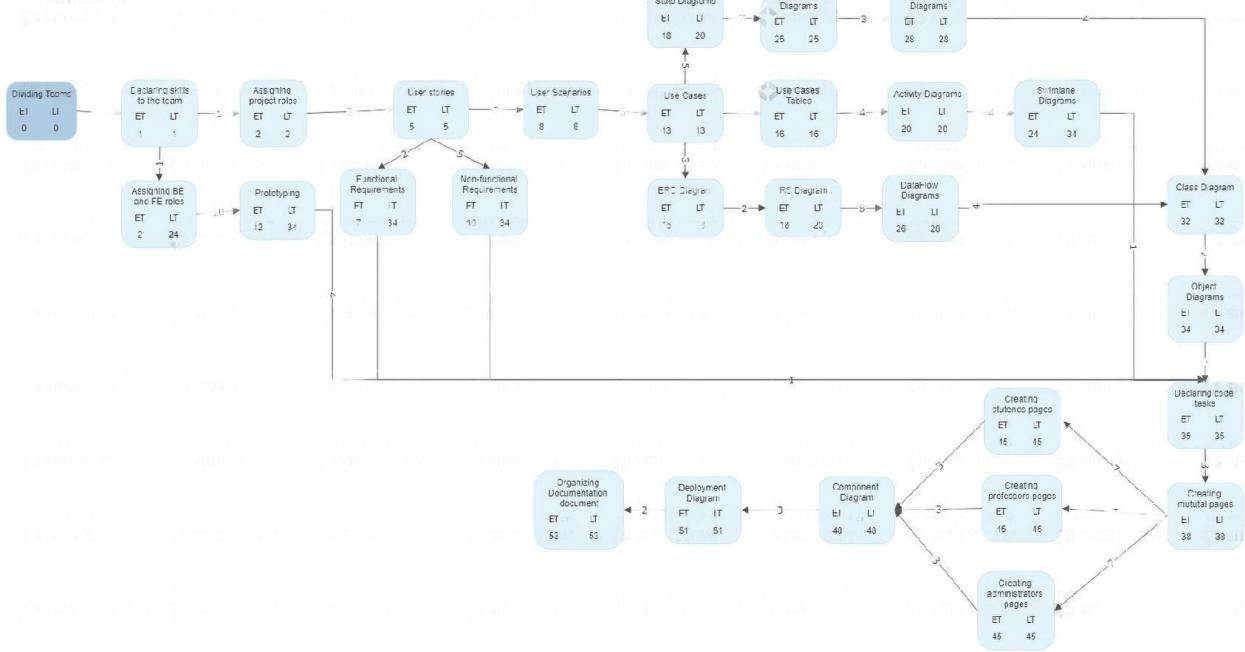
6. Project Planning

6.1 Activity Table

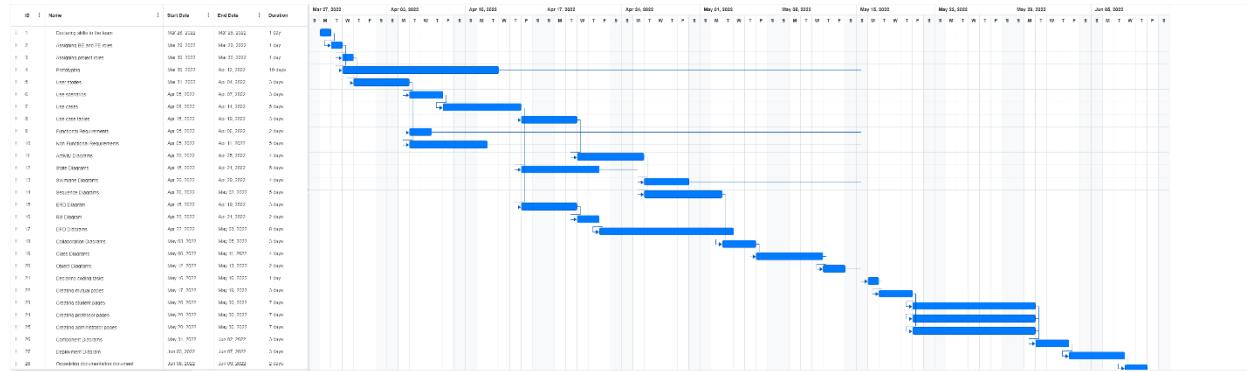
| Activity Number | Activity Name | Dependencies | Duration | Pessimist Time | Optimist Time |
|-----------------|------------------------------|--------------|----------|----------------|---------------|
| 1 | Declaring skills to the team | | 1 | 2 | 1 |
| 2 | Assigning BE and FE roles | 1 | 1 | 2 | 1 |
| 3 | Assigning project roles | 1 | 1 | 2 | 1 |
| 4 | Prototyping | 2 | 10 | 12 | 8 |
| 5 | User Stories | 3 | 3 | 4 | 2 |
| 6 | Use scenarios | 5 | 3 | 4 | 2 |
| 7 | Use cases | 6 | 5 | 7 | 4 |
| 8 | Use case tables | 7 | 3 | 4 | 2 |
| 9 | Functional Requirements | 5 | 2 | 3 | 1 |
| 10 | Non-Functional Requirements | 5 | 5 | 6 | 4 |
| 11 | Activity Diagrams | 8 | 4 | 5 | 3 |
| 12 | State Diagrams | 7 | 5 | 7 | 3 |
| 13 | Swimlane Diagrams | 11 | 4 | 5 | 3 |
| 14 | Sequence Diagrams | 11,12 | 5 | 6 | 4 |
| 15 | ERD Diagram | 7 | 3 | 4 | 2 |
| 16 | RS Diagram | 15 | 2 | 3 | 1 |
| 17 | DFD Diagrams | 16 | 8 | 10 | 6 |
| 18 | Collaboration Diagrams | 14 | 3 | 4 | 2 |
| 19 | Class Diagrams | 17,18 | 4 | 5 | 3 |
| 20 | Object Diagrams | 19 | 2 | 3 | 1 |
| 21 | Declaring coding tasks | 4,9,10,13,20 | 1 | 2 | 1 |
| 22 | Creating mutual pages | 21 | 3 | 4 | 2 |
| 23 | Creating student pages | 22 | 7 | 9 | 5 |
| 24 | Creating professor pages | 22 | 7 | 9 | 5 |
| 25 | Creating administrator pages | 22 | 7 | 9 | 5 |
| 26 | Component Diagram | 23,24,25 | 3 | 4 | 2 |
| 27 | Deployment Diagram | 26 | 3 | 4 | 1 |
| 28 | Organizing documentation | 27 | 2 | 3 | 1 |

6.2 Network Notation

Activity Network

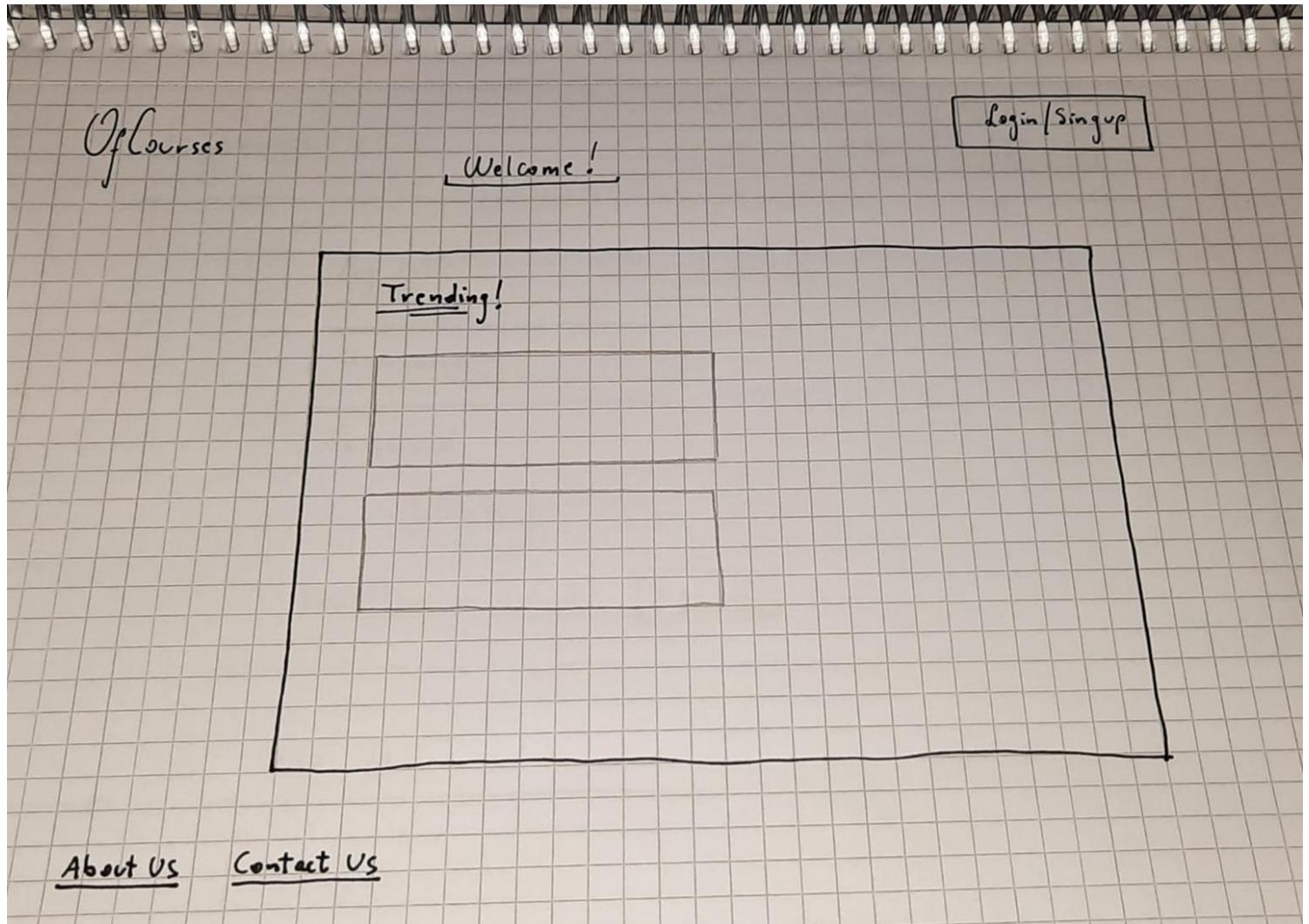


6.3 Gantt Chart



Appendix A. Sketches

- Homepage Sketch



- Login Signup Sketch

The sketch illustrates a user interface for a service named "OfCourses". It features a main login area at the top and a sign-up area below it, connected by a horizontal arrow pointing right.

Main Area:

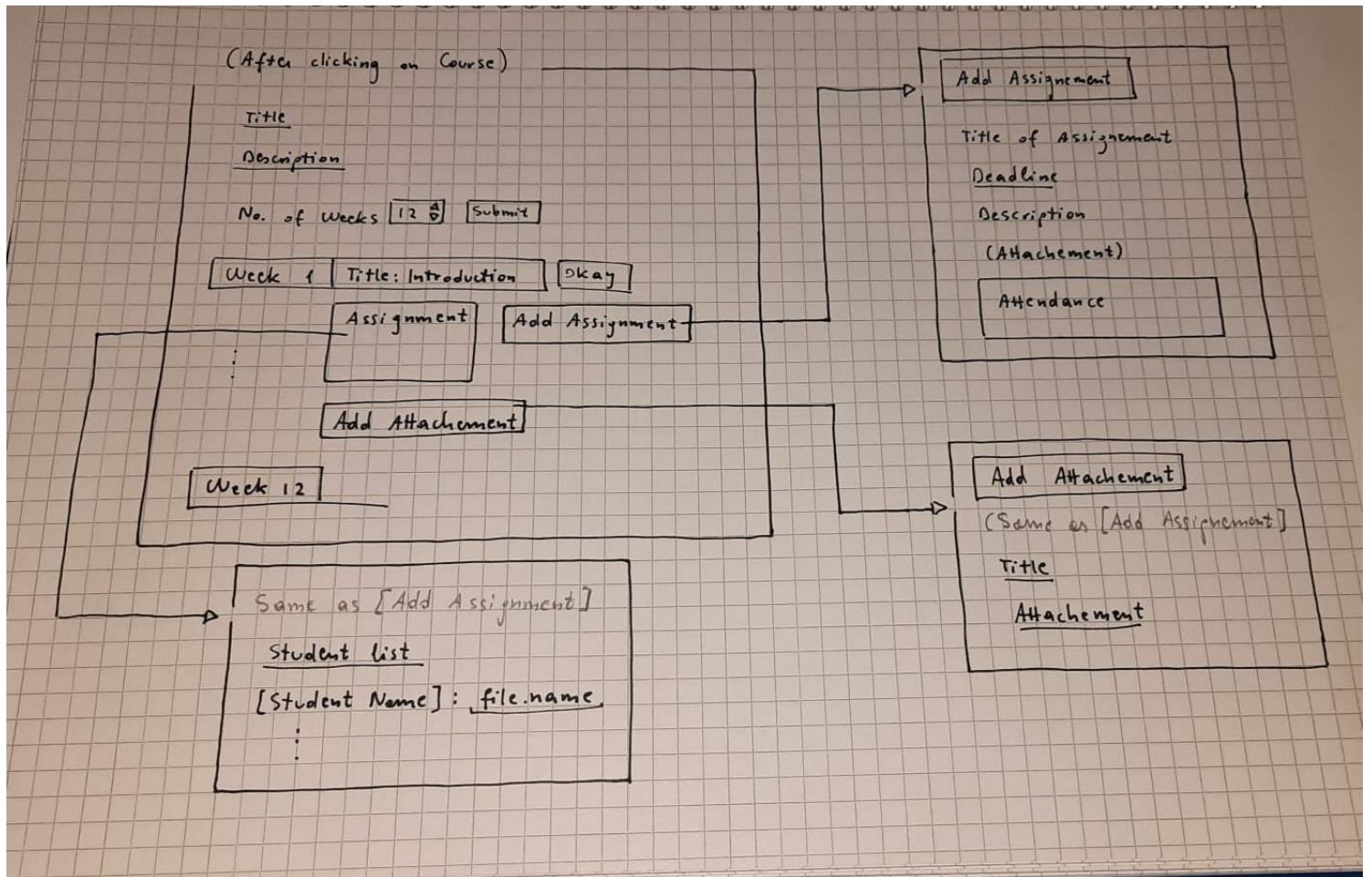
- Logo:** "OfCourses" (written vertically).
- Buttons:** "Log in" and "Sign up" (both enclosed in rectangular boxes).
- Text:** "Log in:"
- Inputs:** "Username" and "Password" (both enclosed in rectangular boxes).
- Buttons:** "Log In" (enclosed in a rectangular box).
- Text:** "Don't have an account? [Sign up](#)" (with "Sign up" in a rectangular box).

Sign Up Area:

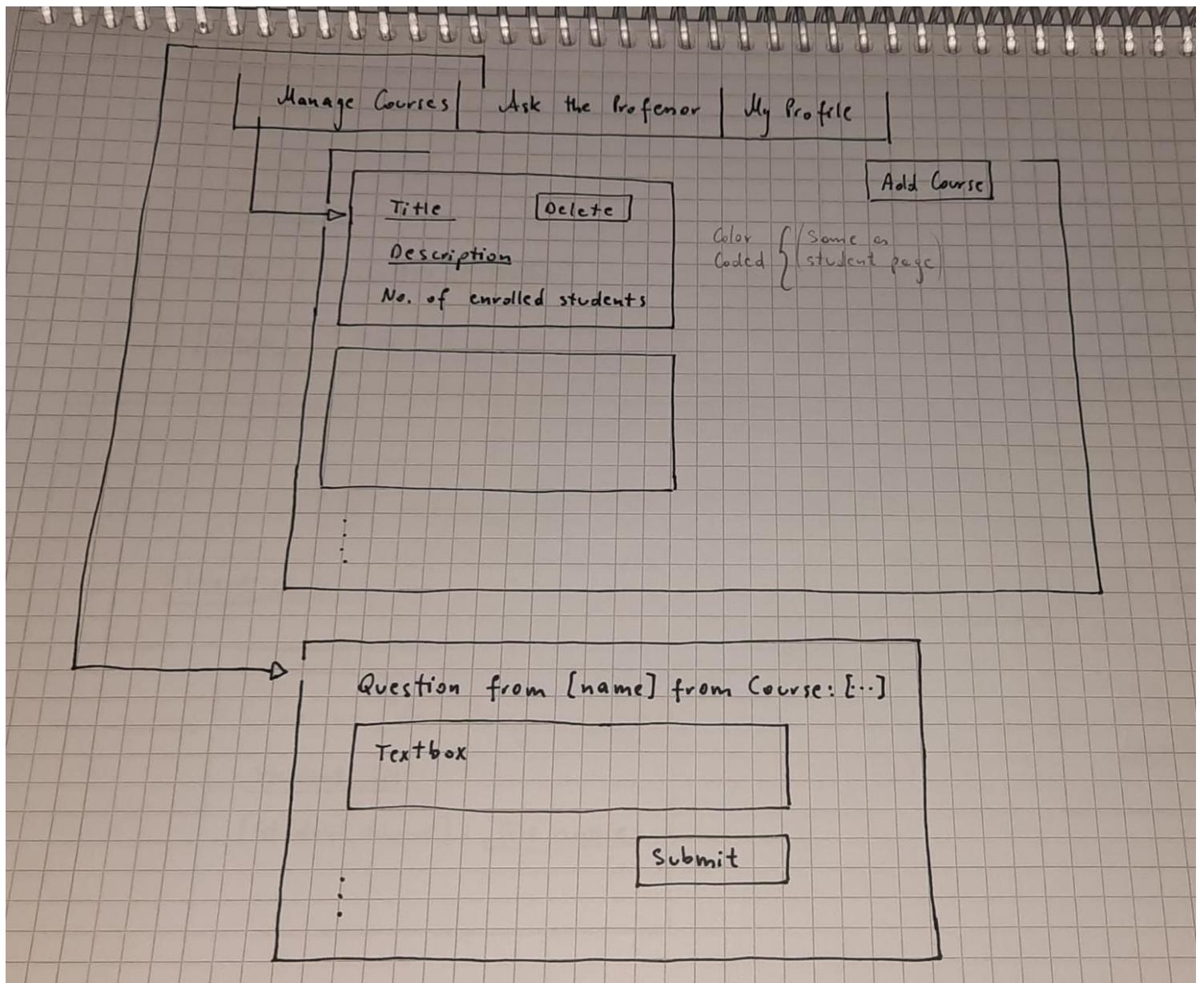
- Text:** "Sign Up:"
- Inputs:** "Name (or username)", "Email", and "Password" (all enclosed in rectangular boxes).
- Buttons:** "Sign Up" (enclosed in a rectangular box).

A horizontal arrow points from the "Sign Up:" text in the main area to the "Sign Up:" text in the sign-up section.

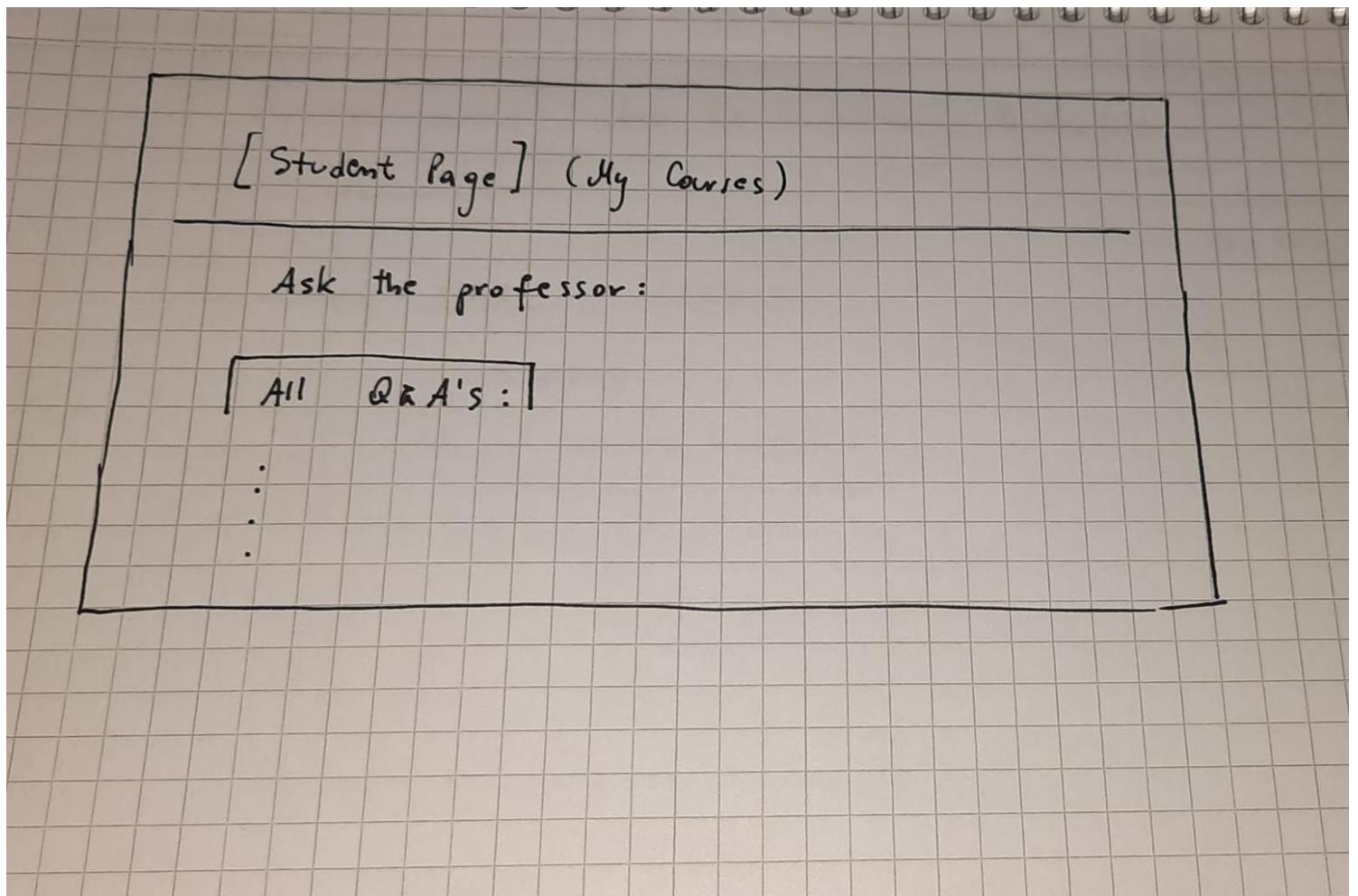
- Professor Course Page Sketch



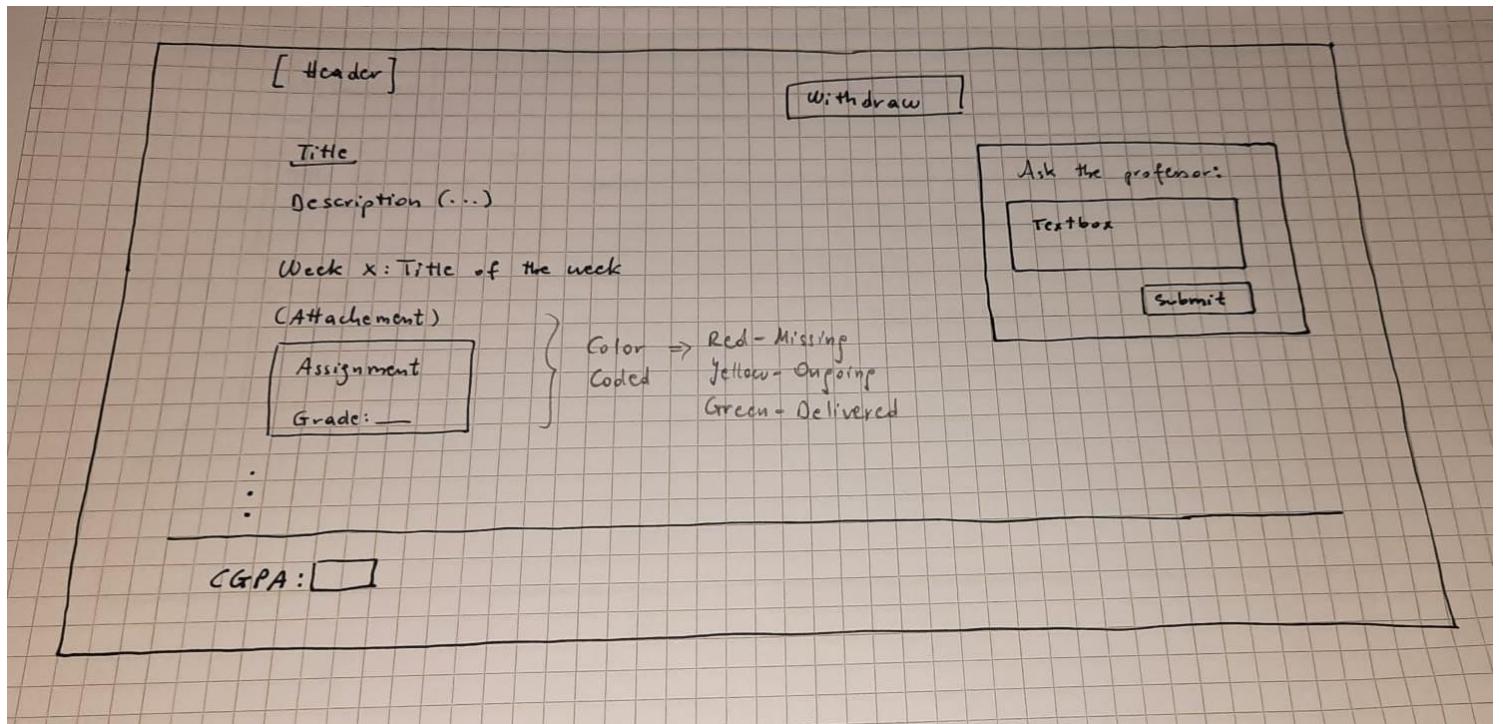
- Professor Manage Courses Sketch



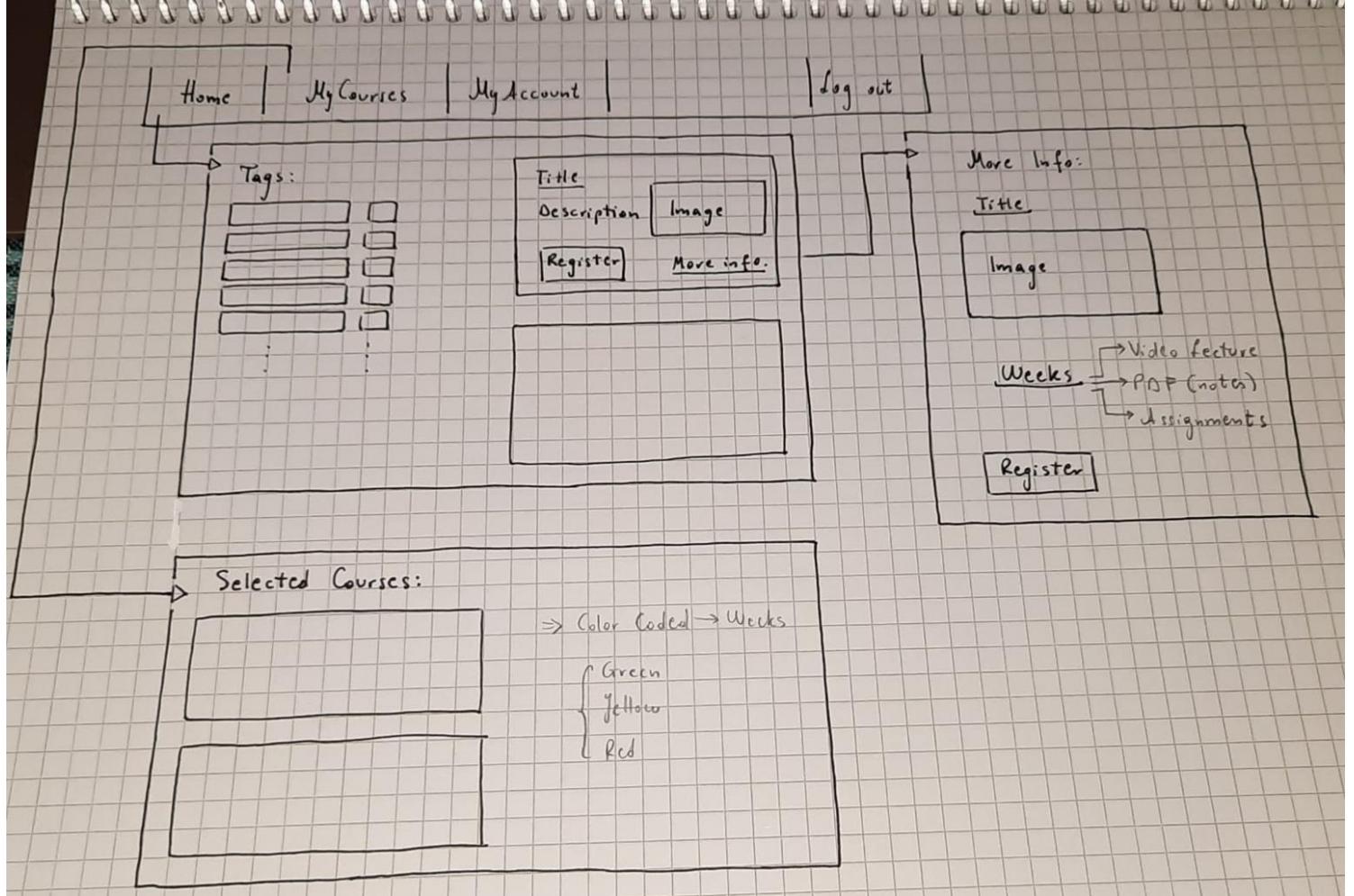
- Student Ask the Professor Sketch



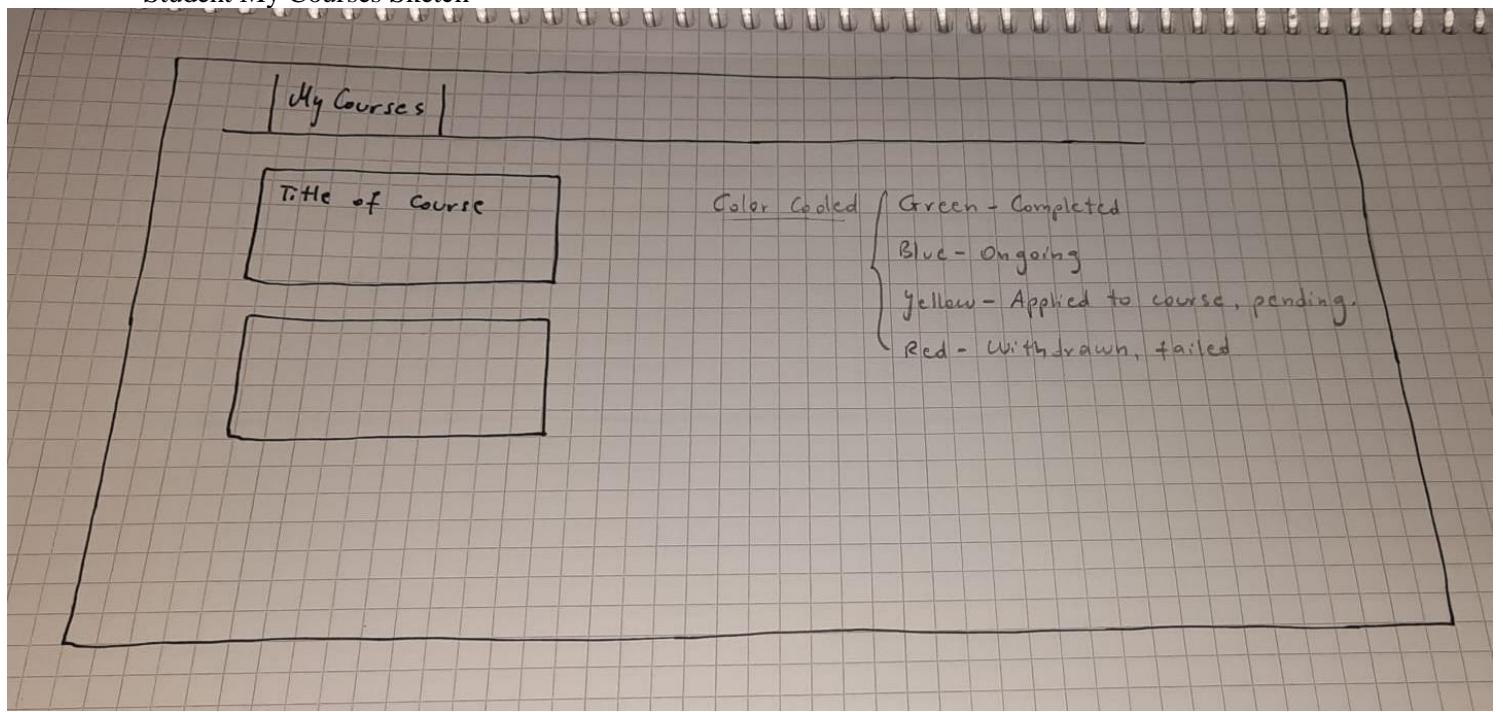
- Student Course Page Sketch



- Student Homepage Sketch



- Student My Courses Sketch



Appendix B. Prototype

- Assignment Page Prototype

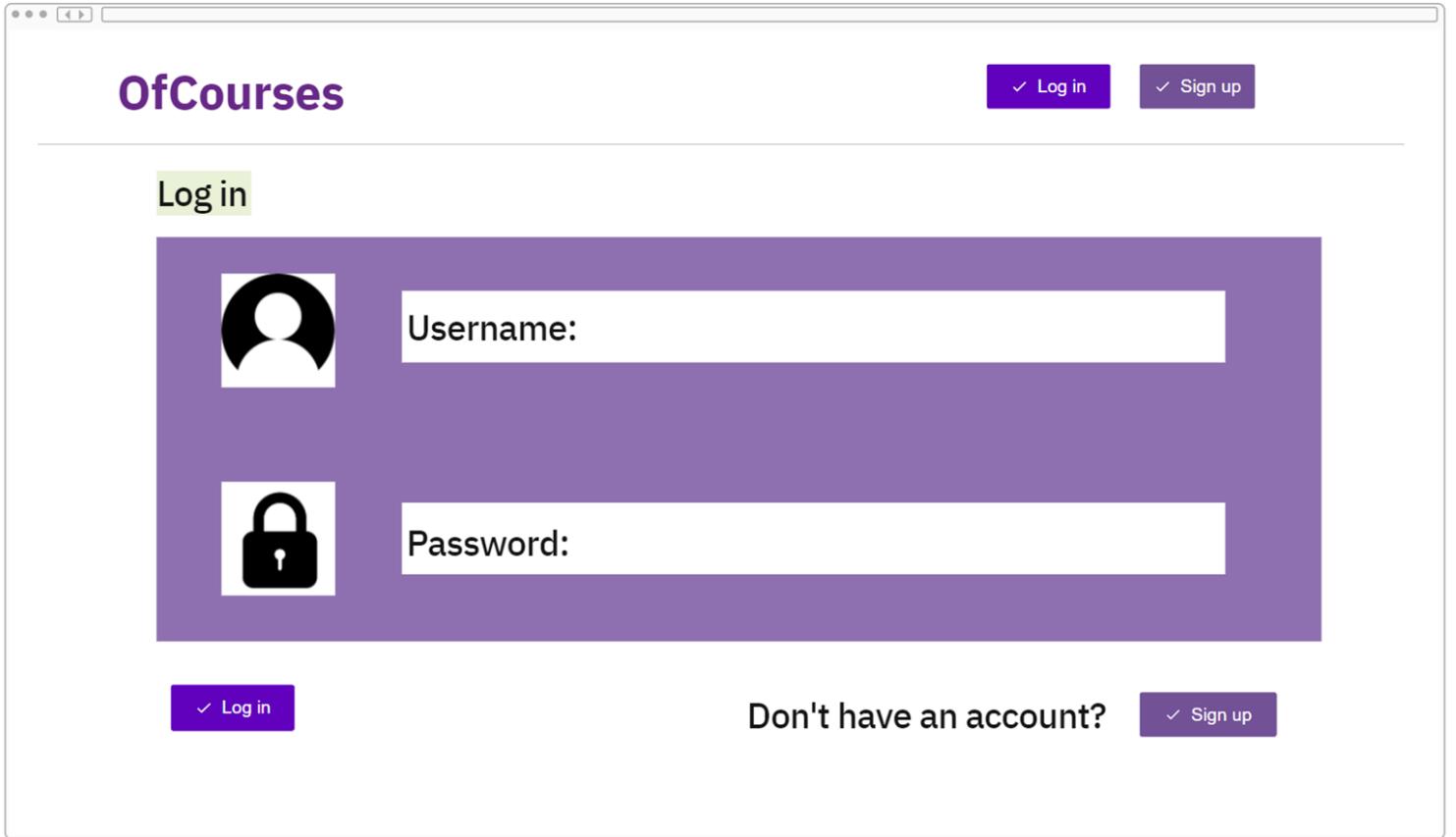
The screenshot shows a web-based assignment page. At the top, a header bar displays three small icons (back, forward, and search). Below the header, the title "After clicking assignment" is centered. In the top right corner, there is a large button with the text "+Add or create" and a red circular arrow icon.

The main content area is titled "Homework 1". To the left of the title, the author "Erind Bedalli" and the submission date "Monday 25th April 2022 9:05" are listed. To the right, the due date "Due May 1, 11:59 PM" is shown. Below the title, the point value "100 points" is displayed.

A bold instruction "SUBMISSION: You will submit the source files and an accompanying report (in Word or PDF format) where you are going to explain your work." is present.

Below the submission instructions, there is a file preview section. It shows a thumbnail of a document titled "CEN352 Preliminary Homew..." and a "PDF" link. To the right of this section, there is a "Attendance" field with two options: "Yes" (marked with a checkmark) and "No".

- Login Prototype



The image shows a login prototype for a website called "OfCourses". The interface is designed with a purple color scheme. At the top, there is a header bar with the "OfCourses" logo on the left and two buttons on the right: "✓ Log in" and "✓ Sign up". Below the header, the word "Log in" is displayed in a green button. The main form area has a purple background. It contains two input fields: one for "Username" with a user icon and one for "Password" with a lock icon. At the bottom of the form, there are three buttons: "✓ Log in" on the left, "Don't have an account?" in the center, and "✓ Sign up" on the right.

Log in

✓ Log in ✓ Sign up

OfCourses

✓ Log in ✓ Sign up

Log in

Username:

Password:

✓ Log in Don't have an account? ✓ Sign up

- Q&A's Professor Prototype

The screenshot shows a web-based application for managing courses and interacting with a professor. At the top, there is a navigation bar with four items: "Manage Courses", "Ask the professor" (which is underlined, indicating it is the active page), "My Profile", and "Log out". Below the navigation bar, the main content area has a heading "Q & As from students". Under this heading, there is a list item starting with a bullet point: "• Question from Kevin from Course: Artifical Intelligence:". Following this is a question from a student named Kevin, asking for help with Lecture 7 exercises. Below the question is a large, empty text input field labeled "Answer:" where a response would be typed. In the bottom right corner of this input field is a purple "Submit" button with a white checkmark icon.

Manage Courses Ask the professor My Profile Log out

Q & As from students

- Question from Kevin from Course: Artifical Intelligence:

Q: Hello professor, I had a question regarding Lecture 7. I have learned all the material until now, I still have problems solving the exercises of the assignment given. Can you please give me a hint on how to deal with this thing?

Answer:

✓ Submit

- Q&A's Student Prototype

The screenshot shows a web-based application interface. At the top, there is a navigation bar with four items: "Home", "My Courses" (which is the active tab, indicated by a purple background), "My Account", and a purple "Log out" button. Below the navigation bar, the main content area has a title "Q & As from professors". Under this title, there is a list item starting with a bullet point: "• Artificial Intelligence - Erind Bedalli 2 w". Following this, there is a question from a user named "Q" and an answer from a professor named "A". The question from "Q" is: "Hello professor, I had a question regarding Lecture 7. I have learned all the material until now, I still have problems solving the exercises of the assignment given. Can you please give me a hint on how to deal with this thing?". The answer from "A" is: "Hello Kevin, If you see the beginning of Lecture 8, you can solve the exercises pretty easily. For any other problem you may have, do not hesitate to ask me again. Best.". At the bottom left, there is a text input field with the placeholder "Ask your professor:" and a purple "Submit" button with a checkmark icon.

Home | My Courses | My Account | Log out

Q & As from professors

• Artificial Intelligence - Erind Bedalli 2 w

Q: Hello professor, I had a question regarding Lecture 7. I have learned all the material until now, I still have problems solving the exercises of the assignment given. Can you please give me a hint on how to deal with this thing?

A: Hello Kevin,

If you see the beginning of Lecture 8, you can solve the exercises pretty easily. For any other problem you may have, do not hesitate to ask me again.

Best.

Ask your professor:

✓ Submit

- Course Assignment Prototype

Assignment page

Homework 1

100 points Due May 1, 11:59 PM

Description - SUBMISSION: You will submit the source files and an accompanying report (in Word or PDF format) where you are going to explain your work.

| Turned in: | Student's name | File name |
|------------|----------------|--------------------------|
| | Adri Cara | Adri Cara HW.pdf |
| | Anxhela Baxhia | Anxhela Baxhia.pdf |
| | Anton Papa | Anton Papa.pdf |
| | Blerim Hasa | Blerim Hasa Homework.pdf |
| | Bashkim Dakoli | Bashkim Dakoli.pdf |
| | . | . |
| | . | . |
| | Zenel Hila | Zenel Hila HW.pdf |
| | Xhesika Feto | Xhesika Feto.pdf |

- Course Page Prototype

The screenshot shows a web-based course management system interface. At the top, there is a navigation bar with four items: "Manage Courses", "Ask the Professor", "My Profile", and a purple "✓ Log out" button. Below the navigation bar, the main content area displays a course entry for "Web Development". The course has 90 students. A "Delete" button is located to the right of the student count. The "Course Description" section contains a detailed overview of the course content, mentioning client-side and server-side development, various programming languages, and specific learning objectives. The "Course Objectives" section outlines the goals for students, focusing on creating interactive websites using HTML, CSS, and JavaScript.

Title : Students : 90
Web Development

Course Description :
Introduction to Client Side and Server Side. An overview of HTML, CSS, client-server architecture, protocols such as HTTP, tools such as web, application, database, FTP, secure shell server. Application development using scripting languages. An overview of web servers and their services, how to configure them. Knowledge on PHP Programming, PHP variables, Functions, Object-Oriented Techniques and database connectivity from PHP. Understanding other scripting languages such as CGI, JSP, and other server-side languages. Session management, multimedia, and graphics, application security, XML. Learning how to interactively use PHP with JavaScript and AJAX.

Course Objectives :
Teaching students how to create and publish a basic web page using HTML and its many tags; to describe limitations of creating interactivity including browser support and differences; understanding JavaScript variables, control structures, functions, arrays, and objects; utilize HTML, XHTML, CSS, XML, and JavaScript to develop an interactive website.

- Homescreen Prototype

The image shows a prototype of the OfCourses website's homescreen. At the top, there is a navigation bar with a star icon, the logo 'OfCourses' in purple, a search bar with placeholder text 'Type here', and buttons for 'Search', 'Log in', and 'Sign up'. Below the navigation, a 'Welcome!' message is displayed. A large section titled 'Trending courses!' lists two courses: '100 Days of Python.' and 'Relational Database for Beginners.'. Each course card includes a brief description, a 'Register here' button, and a small thumbnail image. To the right of the cards is a photograph of a laptop displaying a dashboard with various data visualizations. At the bottom left of the main content area is a 'About Us' button.

OfCourses

Type here

Search

Log in

Sign up

Welcome!

Trending courses!

100 Days of Python.

Practice learning one of the most effective and trending languages these days in the computer engineering world, Python.

[Register here](#)

Relational Database for Beginners.

A relational database is a collection of data items with pre-defined relationships between them.

[Register here](#)

About Us

- Manage Course Prototype

After clicking on the Course

Title : Artificial Intelligence

Description : ...

Number of weeks : 12

Week 1:

Title Introduction: ... Add Assignment :

Week 2:

Title Introduction: ... Add Assignment :

...

Week 12:

Title Introduction: ... Add Assignment :

- My Account Page Prototype

Home | My Courses | My Account | ✓ Log out

Name Surname

Username: Placeholder

Email: Placeholder

Password: Placeholder

Phone Nr:
(optional) Placeholder

Bio:
(tell us something
about you)

Placeholder

Share your status:

Working

Away

Focusing

✓ Edit Profile

- My Account Page Professor Prototype

The screenshot shows a web-based account management interface for a professor. At the top, there are four navigation links: "Manage Courses", "Ask the professor", "My Profile" (which is underlined and bolded, indicating it's the active page), and "✓ Log out". Below the navigation, there is a placeholder profile picture icon. The main content area is titled "Prof. Name Surname". It contains several input fields with placeholder text ("Placeholder") and a section for "Your qualifications and specialties" with checkboxes for "Web Development", "Object Oriented Programming" (which is checked), "Cyber Security" (which is checked), and an "Add another" option. There is also a section for "Share your status" with checkboxes for "Working", "Away" (which is checked), and "Focusing" (which is checked). A purple button at the bottom left says "✓ Edit Profile".

Manage Courses | Ask the professor | My Profile | ✓ Log out

Prof. Name Surname

Username: Placeholder

Email: Placeholder

Password: Placeholder

Phone Nr: (optional) Placeholder

Your qualifications and specialties:

Web Development

Object Oriented Programming

Cyber Security

Add another

Share your status:

Working

Away

Focusing

✓ Edit Profile

- My Courses Student Page Prototype

Home | My Courses | My Account | ✓ Log out

Selected Courses:

- Web Development
- Computer Networks
- Artificial Intelligence
- Numerical Analysis

Ask the professor section :

Which do you want to ask? ▾

- **Web Development - Weeks 14**
Status of the course: Completed 14/14 CGPA : 3.50 / 4.00
- **Artificial Intelligence - Weeks 12**
Status of the course: Ongoing 8/12 CGPA : / 4.00
- **Numerical Analysis - Weeks 10**
Status of the course: Pending
- **Computer Networks - Weeks 16**
Status of the course: Failed

- Signup Prototype

A wireframe prototype of a Signup form. The form is contained within a light gray rectangular frame. At the top center is a green button labeled "Sign up". Below it is a purple rectangular input field divided into four horizontal sections. The first section contains the label "Name Surname :". The second section contains the label "Email :". The third section contains the label "Password :". The fourth section contains the label "Confirm Password :". Below these four sections is a purple horizontal bar containing a white rectangular button with the text "✓ Sign up".

- Student Page Prototype

The screenshot shows a student page prototype with a navigation bar at the top. The navigation bar includes links for Home, My Courses, My Account, and Log out. Below the navigation bar, there is a section titled "TAGS:" containing a list of course categories. To the right, there are two course details sections: "Web Development" and "Computer and Information Technology".

TAGS:

- Engineering
- Architecture
- Business Administration
- Criminal Law
- Education

Web Development

Learn about front end development tools, how to build websites and how to use languages such as HTML, CSS, JAVASCRIPT and PHP.

[REGISTER](#) [More Information.](#)

Computer and Information Technology

Learn how to program your own code projects and applications along with the needed theoretical and technological approach.

[REGISTER](#) [More Information.](#)

- Upload Page Prototype

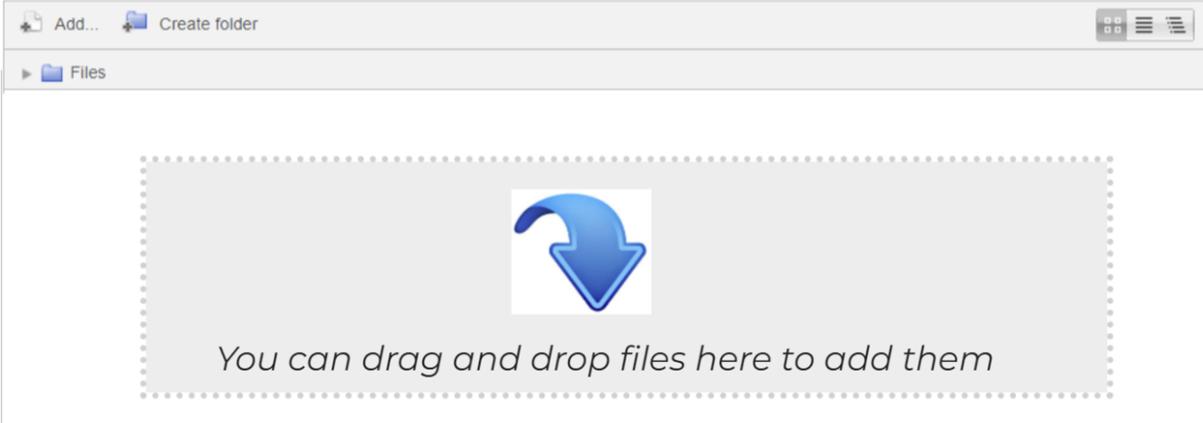
Add Attachment page

Homework 1

Due May 1, 11:59 PM

100 points

Please submit your assignment regarding the lab here.

A screenshot of a web-based file upload interface. At the top, there are buttons for 'Add...' and 'Create folder'. Below that, a folder icon labeled 'Files' is shown. In the center is a large, dotted rectangular area with a blue downward-pointing arrow icon inside it. Below this area, the text 'You can drag and drop files here to add them' is displayed.