OfCourses Requirements Specification

April 17, 2022

Table of Contents

1 Executive 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 of our 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/tree/main/Diagrams

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 course's and homework statistics

Lecturer/Instructor:

- Add/edit/delete courses
- Assign/edit/delete homework

- Check course's and homework statistics
- 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

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, Javascript, 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

- ○ Latency from the website will be less than 0.1s.
- ○ The server should be able to handle around 2000 users using it simultaneously.
- ○ An average traffic would be around 100 students/hour.
- ○ 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 accessable 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
They 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 will 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 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.

3.2.3.2 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.

3.2.3.3 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 have 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

1. Scenario title: <u>Successful user log-in</u>
   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: <u>User forgets his password</u>
   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: <u>View available courses</u>
   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: <u>Take a course</u>
   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 homepage.

5. Scenario title: <u>Cancel Course</u>
   a. For different reasons, the student wants to be enrolled 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: <u>Check grades</u>
   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: <u>View Taken Courses</u>
   a. The user wants to be informed about the courses that he is attended 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: <u>Edit credentials</u>
   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: <u>Add course</u>
   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 in the system.
   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: <u>Edit course</u>
    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: <u>Delete course</u>
    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: <u>Check course's Statistics</u>.
    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: <u>Assign HW</u>
    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: <u>Edit HW</u>
    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: <u>Delete HW</u>
    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: <u>Check HW's statistics</u>
    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: <u>Add instructor</u>
    a. After logging in, the administrator is redirected to his dashboard.
    b. The button "Add instructor" provides the administrator the opportunity to add a new instructor in the system.
    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: <u>Edit instructor</u>
    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: <u>Delete instructor</u>
    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: <u>Validate course</u>
    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.

4.2   Use Cases Diagram

4.2.1 General Use Case
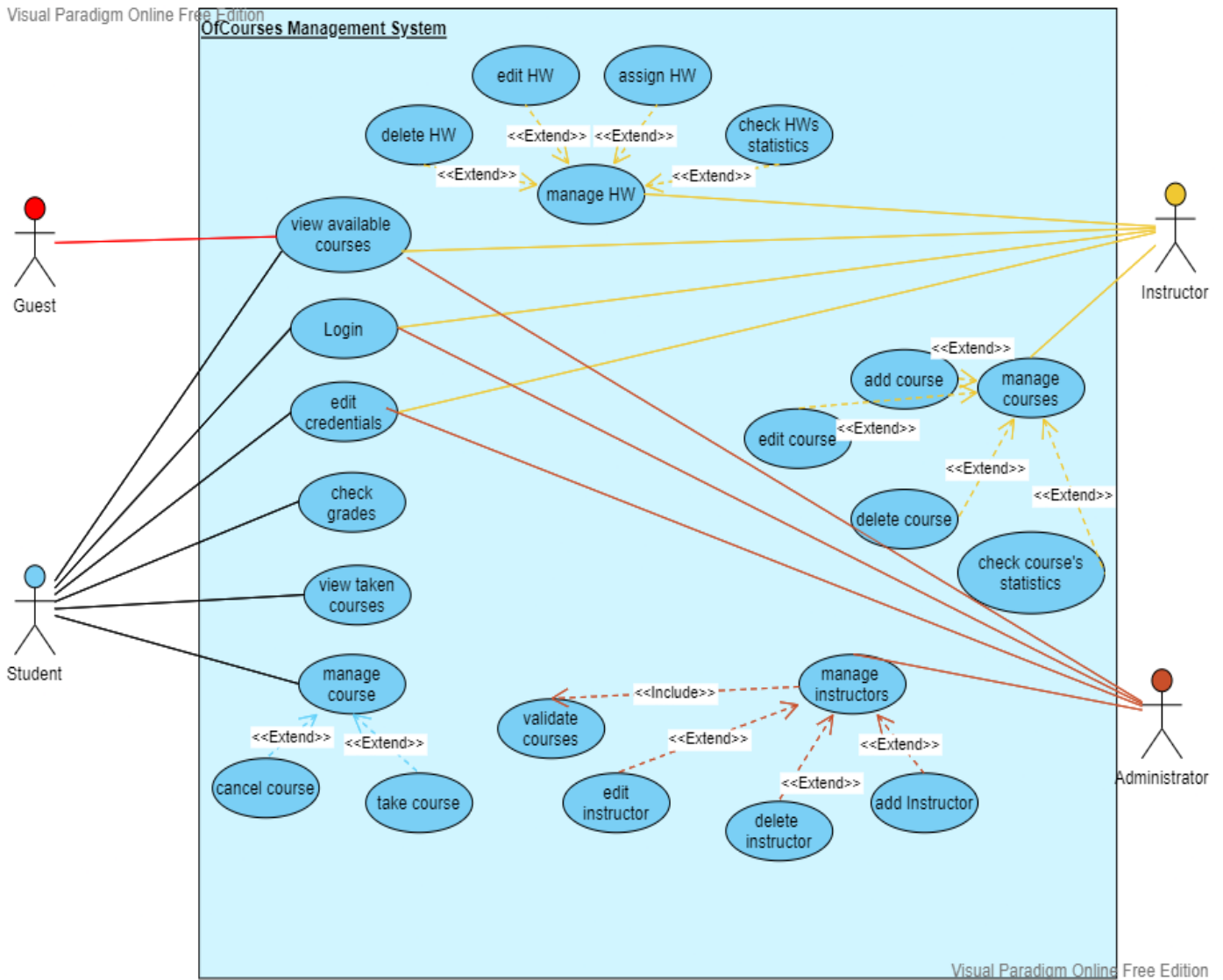
4.2.2 Guest Use Case
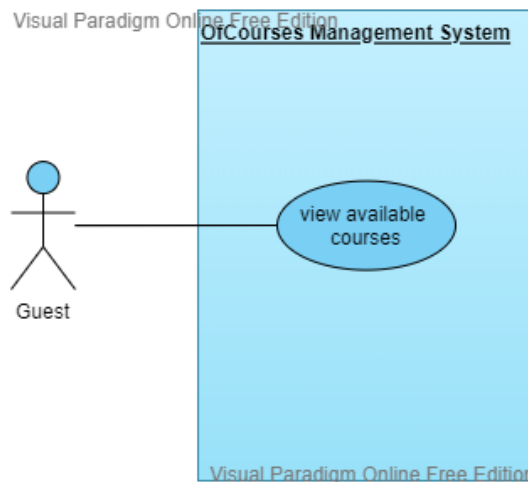
4.2.3 Student Use Case

4.2.4 Instructor Use Case

4.2.4 Administrator Use Case

4.3   Use Cases Table (Extended)

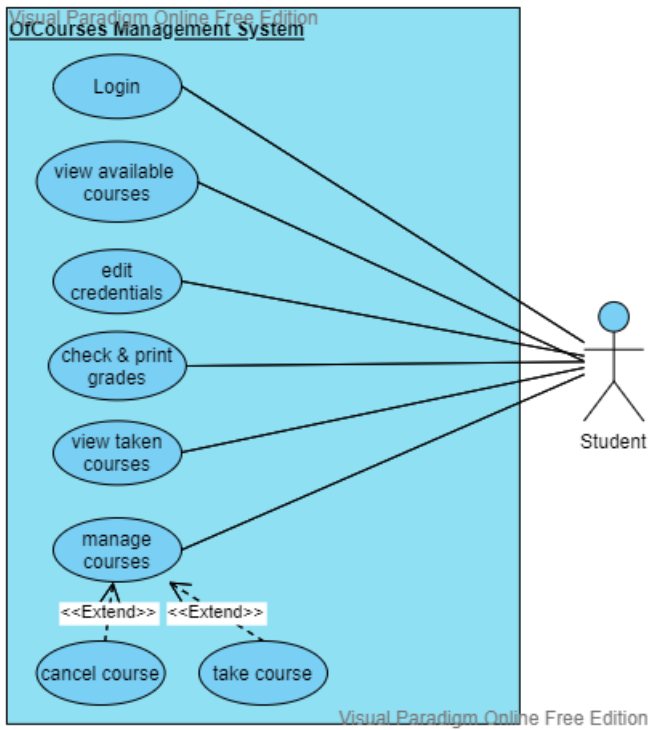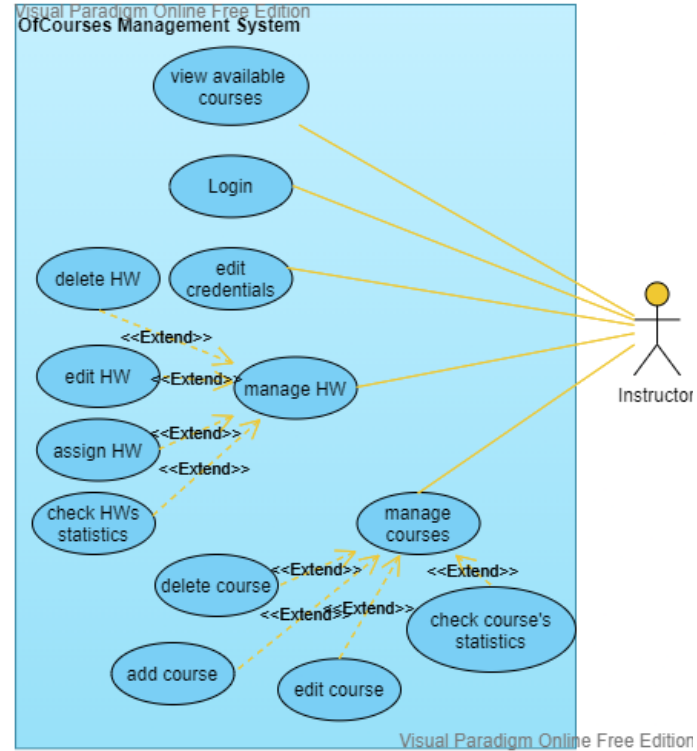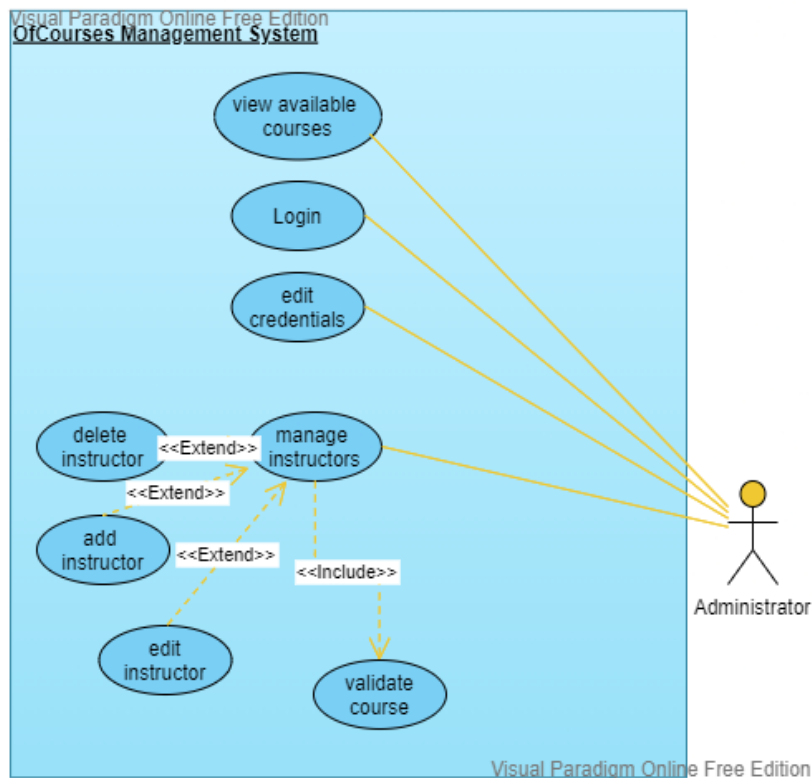| UC_1.1	Login
Summary	- Each user will be able to login into the system by entering their respective credentials.
Dependency	-
Actors	- Student, Instructor, Administrator
Preconditions	- The user must have a valid account, where the Students can easily sign up, while the Instructors will be created by the Administrator, which will be created in advance,
Description of the Main Sequence	- Each user will be able to login 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	- 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, 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 description 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	- 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	- 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	- 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	- 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_2.2	**View Taken Courses**
Summary	- Students can view courses that they are enrolled in and make changes if they want.
Dependency	-
Actors	- Students
Preconditions	- Student users must be enrolled in at least one course.
Description of the Main Sequence	- Students can observe course information for courses they are participating in at the moment or have previously participated in.
Description of the Alternate Sequence	- The student has not been enrolled in any course before.
Non-functional requirements	-
Postconditions	- The user is directed to the page with course information.

UC_2.2	**Check Grades**
Summary	- Students can check their course grades.
Dependency	-
Actors	- Students, Instructor, Administrator
Preconditions	- Students must be enrolled in at least one course. - Student must have participated in exams or submitted his HWs.
Description of the Main Sequence	- 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	- 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	- The student's personal grades will be invisible to others, except himself, the Instructor, and the Administrator.
Postconditions	- Users are directed to the page with their grades and courses.

UC_3.1	Manage HW
Summary	- Instructors can do certain actions regarding homework.
Dependency	-
Actors	- Instructor
Preconditions	- 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	- 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_3.1.1	Assign HW
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 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_3.1.2	**Edit HW**
Summary	- Instructors can make changes regarding homework.
Dependency	- Manage HW
Actors	- Instructor
Preconditions	- 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	- 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_3.1.3	**Delete HW**
Summary	- Instructors can delete a certain HW for various reasons..
Dependency	- Manage HW
Actors	- Instructor
Preconditions	- 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	- 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	- 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_3.1.4	Check HW's Statistics
Summary	- The Instructor will be able to look at the statistics from the HWs.
Dependency	- Manage HW
Actors	- Instructor
Preconditions	- 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	- 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	- 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_3.2	Manage Courses
Summary	- The Instructor is able to 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 is able to 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 in the instructor's dashboard as a course that needs management

UC_3.2.1	Add Course
Summary	- The Instructor is able to add a course.
Dependency	- Manage courses
Actors	- Instructors
Preconditions	- Instructor must have the course validated by the Administrator
Description of the Main Sequence	- Instructor will have to pass the Administrator test for the course - Instructor is able to add the course
Description of the Alternate Sequence	- Instructor does not have the validation from the Administrator and he cannot add the course. - The course is already added.
Non-functional requirements	-
Postconditions	- The course is shown in the student's dashboard for selection, or selected courses and in the instructor's dashboard as a course that needs management.

UC_3.2.2	Edit Course
Summary	- The Instructor is able to edit a course
Dependency	- Manage courses
Actors	- Instructors
Preconditions	- Instructor must have the course validated by the Administrator - Course must be available on OfCourses
Description of the Main Sequence	- 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	- 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	-
Postconditions	- Course is shown in student's dashboard for selection, or selected courses and in instructor's dashboard as a course that needs management

UC_3.2.3	Delete Course
Summary	- The instructor is able to delete the course from OfCourses
Dependency	- Manage Courses
Actors	- Instructor
Preconditions	- Instructor must have the course validated by the Administrator - Instructor must have the courses posted
Description of the Main Sequence	- For different reasons, the instructor is able to delete the course from OfCourses, and all the enrolled students are free and not charged for that course
Description of the Alternate Sequence	- 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	-
Postconditions	- The course is deleted from the Student's and Administrator's dashboard

UC_3.2.4	Check Course's Statistics
Summary	- The Instructor can check how the students performed at this course.
Dependency	- Manage Courses
Actors	- Instructor
Preconditions	- 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	- 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	- The course has no students enrolled. - There are no exams, HWs, or attended lessons.
Non-functional requirements	-
Postconditions	- The Instructor is informed about his course progress.

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	- 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	- Must be logged in as an Administrator. - There must be a given course to be examined
Description of the Main Sequence	- 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	- 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 is able to edit an Instructor.
Dependency	- Manage Instructors
Actors	- Administrator
Preconditions	- Must be logged in as Administrator. - Must have at least one Instructor.
Description of the Main Sequence	- The Administrator can edit personal information about the Instructor. - The Administrator can edit course validation for the Instructor.
Description of the Alternate Sequence	- There are no Instructors to be edited.
Non-functional requirements	-
Postconditions	- The information about the Instructor is updated

UC_4.1.3	**Add instructor**
Summary	- The Administrator is able to 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 is able to delete an Instructor.
Dependency	- Manage Instructors
Actors	- Administrator
Preconditions	- 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.