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.