

# ***OfCourses Requirements Specification***

**April 17, 2022**

## **Table of Contents**

<b>OFCOURSES REQUIREMENTS SPECIFICATION</b>	<b>1</b>
<b>APRIL 17, 2022</b>	<b>1</b>
<b>1. EXECUTIVE SUMMARY</b>	<b>3</b>
1.1 PROJECT OVERVIEW	3
1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION	3
<b>2. PRODUCT/SERVICE DESCRIPTION</b>	<b>3</b>
2.1 PRODUCT CONTEXT	3
2.2 USER CHARACTERISTICS	3
2.3 ASSUMPTIONS	3
2.4 CONSTRAINTS	3
2.5 DEPENDENCIES	4
<b>3. REQUIREMENTS</b>	<b>4</b>
3.1 FUNCTIONAL REQUIREMENTS	5
3.2 NON-FUNCTIONAL REQUIREMENTS	5
3.2.1 <i>Product Requirements</i>	5
3.2.1.1 <b>User Interface Requirements</b>	6
3.2.1.2 <b>Usability</b>	6
3.2.1.3 <b>Efficiency</b>	6
3.2.1.3.1 Performance Requirements	6
3.2.1.3.2 Space Requirements	6
3.2.1.4 <b>Dependability</b>	6
3.2.1.5 <b>Security</b>	7
3.2.2 <i>Organizational Requirements</i>	7
3.2.2.1 <b>Environmental Requirements</b>	7
3.2.2.2 <b>Operational Requirements</b>	7
3.2.2.3 <b>Development Requirements</b>	7
3.2.3 <i>External Requirements</i>	7
3.2.3.1 <b>Regulatory Requirements</b>	7
3.2.3.2 <b>Ethical Requirements</b>	7
3.2.3.3 <b>Legislative Requirements</b>	7
3.2.3.3.1 Accounting Requirements	7
3.2.3.3.2 Security Requirements	7
3.3 DOMAIN REQUIREMENTS	7
<b>4. USER SCENARIOS/USE CASES</b>	<b>7</b>

# 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

## *OfCourses Requirements Specification*

- 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.

### *OfCourses Requirements Specification*

- 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*

<b>Req#</b>	<b>Requirement statement</b>	<b>Priority</b>	<b>Date Reviewed</b>	<b>SME Reviewed / Approved</b>
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 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

## *OfCourses Requirements Specification*

### **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.

## *OfCourses Requirements Specification*

### **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.