Software Requirements Specification

for

Online User and Course Registration

Version 1.0 approved

Prepared by Anders Berg

Capstone for Computer Software Technology -CST 499 - Arizona Global Campus

November 11, 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

To describe the requirements for the development on an online solution for user and course registration. This document will outline the requirements for the design and construction of this solution for user and course registration intended for learning institutions.

1.2 Document Conventions

For this documentation, there is no differentiation between requirement hierarchy. Since this is a small project, each requirement is treated equally and individually.

1.3 Intended Audience and Reading Suggestions

The intended audience for this report is developers, project managers, marketing staff, users, testers, and documentation writers that will be involved in the construction of this project. This report is organized by describing the overall description of the product, the external interface requirements, system features and other non-functional requirements. Describe what the rest of this SRS contains and how it is organized. It is intended that this document be read in order from front to back to fully understand the product scope.

1.4 Product Scope

The scope of this development is to create a system for user and course registration for learning institutions. The system will need to be able to register users that are new and provide the ability for them to create an account. In addition, users need to be assigned a unique id and password. User ID's need to be unique and not be duplicated. User profiles need to include name, phone, and email. Once a user completes a registration, they need to be able to login with their unique id and password that they created. Users will need to be able to register for course in spring/summer/fall semesters. Courses will need to be listed out and available for registration per semester. Course need to have a cap for max registration and if it is full users will be waitlisted. Finally, users need to be able to cancel their registration and automatically allow for waitlisted users to register.

1.5 References

American Society for Quality. (2021). *What is Software Quality?* ASQ. https://asq.org/quality-resources/software-quality.

Pressman, R. S. (1992). *Software engineering: A practitioner's approach* (8th ed.). Nueva York: McGraw-Hill.

Sommerville, I. (2018). *Software Engineering*. Hallbergmoos/Germany: Pearson.

2. Overall Description

2.1 Product Perspective

The development of this product will be from ground up as a new system. This will be a self-contained system that does not have any internal dependencies that need to be considered. For the development of this system, all external technologies and libraries can be considered for development. The best technologies that fit the product scope will be selected to implement the desired functionality of the ecommerce website.

2.2 Product Functions

Functional

- Able to register new users
- Users need to be assigned a unique id and password that they can login with
- User profiles need to include name, phone, and email
- Users will need to be able to register for course in spring/summer/fall semesters
- Users able to register for course based on semester
- Course have a cap for max registration and allow users to be waitlisted
- Allow users to cancel course registration and automatically allow for waitlisted users to register

Non-functional

- Website availability need to be 24 hours a day.
- Website needs to be authenticated with an SSL certificate
- Build using modern web languages like HTML, CSS, JavaScript and PHP.
- Website that supports all devices including mobile, tablet and Traditional PC's
- Database design will be constructed with a user and transaction table
- Website needs to be handled a variety of traffic loads, including high traffic during peak hours or during promotional phases.
- Built on AWS cloud architecture

2.3 User Classes and Characteristics

Classes used in system design will be user, student, admin, interface, catalog, and course. User will be a parent class that will pass methods and properties to student and admin. Student and admin classes will be made up of ID's and personal information. Admins and students will be able view their profile and registration history. An interface will allow admins to login into the system, update accounts and registration. Catalog will be constructed with catalog id, semester id, course id and max number of registrants. Course will have id that are related to catalog.

2.4 Operating Environment

The plan is for this website to be hosted using AWS and cloud architecture. This will the project to implement multiple servers and provide resources to users through an interface. Ultimately allowing for faster performance, due to multiple servers balancing the workload.

2.5 Design and Implementation Constraints

Develop the solution using any modern web languages that can handle both server-side and client-side programming. Server-side should focus on deploying information quickly, without taxing the user with excess data on their mobile device. The client-side interface needs to be developed with mobile presentation first.

2.6 User Documentation

No documentation yet.

2.7 Assumptions and Dependencies

Dependent on cloud architecture, and existing API's used for development.

3. External Interface Requirements

3.1 User Interfaces

User Interface will be designed with mobile in mind. The design will minimize any overhead features and only present want is need for the user to execute the core functionality of user and course registration.

3.2 Hardware Interfaces

Hardware Interface will be within AWS or cloud computing platform.

3.3 Software Interfaces

Software Interface will be within AWS or cloud computing platform.

3.4 Communications Interfaces

Communication Interface will be within AWS or cloud computing platform.

4. System Features

4.1 Able to register new users

4.1.1 Description and Priority

Presented in the user interface within an actionable sequence that allow users to register for the system. Priority: High

4.1.2 Stimulus/Response Sequences

User clicks on and fills out form that allows them to register for the system

4.1.3 Functional Requirements

- Checks against unique ID of it is not available
- Checks password to ensure that it meets strength requirements
- Check that all form values are submitted

4.2 Users will need to be able to register for course in spring/summer/fall semesters

4.1.1 Description and Priority

Presented in the user interface within an actionable sequence that allow user to review and register for course. Priority: Medium

4.1.2 Stimulus/Response Sequences

User clicks on and fills out form that allows them to register for courses

.

4.1.3 Functional Requirements

- Check that all form values are submitted
- Max registration for course
- Allows user to be waitlisted for course that are full

4.3 Allow users to cancel course registration/register waitlisted users

4.1.1 Description and Priority

Presented in the user interface within an actionable sequence that allow user to cancel registration Priority: Medium

4.1.2 Stimulus/Response Sequences

User clicks on course they have registered for and remove it from their queue

4.1.3 Functional Requirements

- Updates the registration count
- Allows waitlisted user to register for course
- Updates waitlist queue

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Transactions when submitted need to be completed within 10 seconds to prevent loss of data. Requests for items and actions from the user need to be completed withing 3 to 5 seconds. The system needs to be available 24/7 to avoid loss of sales and be operable in peak campaign hours.

5.2 Safety Requirements

5.3 Security Requirements

Transaction need to be secure an encrypted to prevent data and information loss.

5.4 Software Quality Attributes

Our system is that it needs to be available 24/7 and ensure that the user and course platform is continually available. Our goal will to be to design this system to be highly available. Availability will be measured by how reliable this system and its core functionality are. We will determine availability by measuring the time it is available and the success rate for executing functionality. To design a system to be highly available we will make sure that it is designed to prevent faults and be resilient to errors. Another factor will be designing the system with redundancy, particularly in the cloud data architecture, so that internal and external information will be duplicated and be able to be retrieved 24/7. Since this is an external quality attribute it will be tested and validated during alpha and beta user tests of the system.

Maintainability is the level of effort that is required to maintain a software system. Changes will be inevitable during the life of this online business ecommerce platform, but the response to changes will be crucial for the viability of this system being continually available. To do this, our enterprise needs to be architected to be flexible and for dependencies to be reduced in the domain.

5.5 Business Rules

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: To Be Determined List