
Software Requirements Specification

for

Online Course Enrollment System (OCES)

Version 1.0 approved

Prepared by Cortilious Spencer

The University of Arizona Global Campus

11/10/2025

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Project Scope	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Features	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment	3
2.5 Design and Implementation Constraints	3
2.6 User Documentation	3
2.7 Assumptions and Dependencies	4
3. System Features	4
3.1 System Feature 1	4
3.2 System Feature 2 (and so on)	Error! Bookmark not defined.
4. External Interface Requirements	6
4.1 User Interfaces	6
4.2 Hardware Interfaces	6
4.3 Software Interfaces	6
4.4 Communications Interfaces	6
5. Other Nonfunctional Requirements	7
5.1 Performance Requirements	7
5.2 Safety Requirements	7
5.3 Security Requirements	7
5.4 Software Quality Attributes	7
6. Other Requirements	7
Appendix A: Glossary.....	7
Appendix B: Analysis Models	8
Appendix C: Issues List	8

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document specifies the requirements for the Online Course Enrollment System (OCES). The OCES enables students to register, enroll in courses, and manage their profiles securely through a web-based interface. This document provides complete functional and non-functional needs to guide design and development.

1.2 Document Conventions

All constraints are described sequentially (e.g., REQ-1). Urgency levels are defined as High, Medium, or Low. Mandatory requirements are categorized as High.

1.3 Intended Audience and Reading Suggestions

There are four main audiences for this document: developers, testers, project managers, and academic administrators. A reader's first step should be to read Section 1 (Introduction) for an overview, then move to Section 2 - Overall Description and Section 3 - System Features for detailed technical information.

1.4 Project Scope

The Online Course Enrollment System (OCES) targets to provide a secure and efficient platform for students to register, enroll in, and manage their course activities. It includes automated notifications, waitlist functionality, and robust data management while maintaining compliance with FERPA privacy standards.

The Online Course Enrollment System (OCES) will automate the student course registration process for three academic semesters: spring, summer, and fall. The system ensures data accuracy, improves efficiency, and provides real-time updates for both students and administrators.

1.5 References

GeeksforGeeks. (2025). Software Requirement Specification (SRS) – Overview.

<https://www.geeksforgeeks.org/software-requirement-specification-srs/>

IEEE. (1998). IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications. IEEE Computer Society.

IEEE Computer Society. (2024). *IEEE Recommended Practice for Software Requirements Specifications* (IEEE 830-2024). IEEE Press

Sommerville, I. (2020). Software Engineering (10th ed.). Pearson Education.

TechTarget. (2023). Software requirement specifications: *Best practices for writing SRS documents*.

<https://www.techtarget.com>

2. Overall Description

2.1 Product Perspective

The OCES is a standalone web application using a client-server architecture. It integrates with a secure relational database for user and course data. The application is compatible with standard web browsers.

2.2 Product Features

- Secure user registration and authentication
- Profile creation and editing
- Course listing by semester
- Enrollment and waitlist management
- Automated notifications for open seats.

2.3 User Classes and Characteristics

- **Students:** Register, enroll, and manage courses.
- **Administrators:** Manage course data, monitor enrollment, and oversee waiting lists.

2.4 Operating Environment

Compatible with modern browsers (Chrome, Edge, Firefox) on Windows, macOS, and mobile devices. The server environment utilizes HTTPS with a MySQL or PostgreSQL database.

2.5 Design and Implementation Constraints

The OCES must comply with FERPA, use HTTPS for all communications, and support at least 1,000 concurrent users. It will leverage modern frameworks, such as Python/Django or Node.js, for backend development, and MySQL or PostgreSQL for database management.

- Must support at least 100 concurrent users.
- Must use HTTPS for all transactions.
- Must comply with UAGC data security standards.

2.6 User Documentation

The system features an online help section and in-app tutorials.

2.7 Assumptions and Dependencies

- A reliable internet connection is required.
- Course data provided by the registrar's database.
- System hosted on a stable web server.

3. System Features

This section describes the major functional capabilities of the Online Course Enrollment System (OCES). Each feature includes its description, priority, interaction flow, and functional requirements. The features are organized according to the main services provided by the system, including registration, login, course browsing, enrollment, waitlisting, and cancellation.

3.1 User Registration

3.1.1 Description and Priority

This feature enables new users to create an account within the OCES system. During registration, the user must provide a unique ID, password, and required profile information.

Priority: High

3.1.2 Stimulus/Response Sequences Stimulus:

1. User selects “Register” on the login page.
2. User enters required profile data and chooses a unique ID and password.

3. User submits registration form.

Response:

1. System validates all required fields.
2. System checks whether the chosen ID already exists.
3. If valid, the system creates the account and stores encrypted credentials.
4. System displays confirmation message and prompts user to log in.

3.1.3 Functional Requirements:

- REQ-1: The system shall allow new users to create an account by entering a unique User ID, password (minimum 8 characters), first name, last name, and email address. The system shall reject registration attempts when required fields are missing.
- REQ-2: The system shall prevent duplicate User IDs by performing a uniqueness check against all existing User ID values in the database before account creation.
- REQ-3: The system shall hash and salt all passwords using an industry-standard hashing algorithm (e.g., bcrypt with a work factor of 10 or higher) before storing them in the database.
- REQ-4: The system shall provide a password recovery feature that sends a reset link to the user's verified email address within 60 seconds of the request.

3.2 Course Enrollment Management

Priority: High

Description: Allows students to enroll in and withdraw from courses.

Functional Requirements:

- REQ-5: The system shall display a list of available courses for the selected semester within 3 seconds of the user selecting "Spring," "Summer," or "Fall."

- REQ-6: The system shall enforce maximum enrollment capacity by preventing enrollment when the number of enrolled students equals the course's defined capacity.
- REQ-7: If a course is full, the system shall automatically place the student on the waitlist in the order the requests are received and display the student's waitlist position.
- REQ-8: When a seat becomes available, the system shall notify the first waitlisted student via email within 30 seconds and provide a 24-hour window for the student to confirm enrollment

3.3 Administrator Functions

Priority: Medium

Description: Enables course management and reporting for administrators.

Functional Requirements:

- REQ-9: The system shall allow administrators to add, edit, and delete course details including course name, instructor, semester, credits, and maximum enrollment using a secure admin interface.
- REQ-10: The system shall generate real-time enrollment and waitlist reports downloadable in CSV format. Reports shall be generated within 5 seconds for datasets up to 5,000 records.

4. External Interface Requirements

4.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

4.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the

nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

4.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

4.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: Issues List

<This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>