Software Requirements Specification

for

Course Enrollment System

Version 1.0 approved

Prepared by Jacob Colburn

The University of Arizona Global Campus

29JUL24

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 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Jacob Colburn | 29JUL24 | Initial Draft | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this SRS document is to document the requirements for the requested enrollment management system. This document will outline the systems interface, performance, constraints and functionality to ensure compliance with the requirements of the stakeholders and customers.

## Document Conventions

Every function has a high priority because they are all requirements of the system.

## Intended Audience and Reading Suggestions

*This document is intended for :*

* *Developers*
* *Users*
* *Project managers*
* *Testers*
* System Architects

The rest of the document cover the software system in more detail explaining the functionality, constraints and performance characteristics. Reading this document in order should fully detail the requirements of the course management system.

## Product Scope

The course management system will help make enrollment an easier process allowing users to register for accounts, create their own profiles, enroll in courses, and manage their courses and enrollment. Logins will be through a ID and password form, and the profiles will include name, phone, email, student ID. A waitlist will be implemented for users applying to courses at max capacity. Courses will be scheduled for three semesters a year (spring/summer/fall).

## References

Berkeley, (2024). “Database Systems”. CS186 Berkeley. <https://cs186berkeley.net/notes/note7/>

IEEE STD 830-1998, “IEEE Recommended Practice for Software Requirement Specifications”. doi: 10.1109/IEEESTD.1998.88286

Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.). Jones & Bartlett Learning. [Course Text](https://platform.virdocs.com/read/2348054/11/#/4/2[ch06]/6/4[ch06-sect1-002]/10/4/2)

# Overall Description

## Product Perspective

The course management system will be a standalone system that will provide the necessary functionality to the users and administrators of the system.Functionality will include user registration, course management, profile management. Allowing users to efficiently create an account, enroll, drop, or get on a wait list for courses, and set up a profile. The systems backend will be a SQL database that stores all necessary user and course information.

## Product Functions

* User Registration
* User Authentication
* Profile Management
* Course listing per semester (Spring/Summer/Fall)
* Course Enrollment
* Course Withdrawal
* Course Waitlist

## User Classes and Characteristics

Students – Users who are students at the university enrolling In courses.

Administrators – Users with privileges to manage courses or students at the university.

## Operating Environment

The system will be website based and easily accessible on all popular modern web browsers such as Firefox, chrome, edge, explorer, and safari on Windows, Linux, and MacOS operating systems.

## Design and Implementation Constraints

* All languages must be available to accommodate the diversity of users but English will be primary.
* System must ensure each user has a unique ID
* Profiles must include key identifiable information such as name, phone, and email address
* Courses should have a maximum number of allowed enrollments that differentiate.
* System must handle concurrent users accessing

## User Documentation

* User Manual
* Online Tutorial
* Online Help Desk

## Assumptions and Dependencies

*Assumptions:*

* *Users will have internet access*
* *Users will have a basic understanding of computers.*
* *Users will understand how course enrollment works and the flow of course scheduling*

*Dependancies:*

* *System may utilize hashing for passwords saved to the database*

# External Interface Requirements

## User Interfaces

* The system will provide a web based interface for all the users interactions.
* The layout will be simple with a standard square login box for user login and authentication. Once logged in it will open to the users profile.
* There will be a nav bar located on the left side of the screen with buttons leading to course management and profile management pages.
* All fonts will be Arial
* All screen resolutions below 1440p will be supported
* The system will be compliant to section 508 following the compliance checklist. (BetaBreakers, 2024)

## Hardware Interfaces

This is strictly a software system. All hardware must be owned by the user.

## Software Interfaces

The system will for now interface with a backend SQL database to store any necessary user and course information such as userID, courseID, fname, lname, semester, etc.

## Communications Interfaces

* The system will utilize HTTP/HTTPS for client server communication
* The stem will communicate with the users email for registration

# System Use Cases

The overall use case diagram should be here.

The text description of each use case should follow.

## User Registration

1. **Objective** – Create User Account to register to the system.
2. **Priority** – High
3. **Source** – Terry Crew (End User)
4. **Actors** – Users/ Administrators
5. **Flow of Events** 
   1. **Basic Flow** – User will select sign up. User will then fill out the online form to register for profile. Profile will be created and user can login.
   2. **Alternative Flow(s)** – User already has an account setup so is redirected to the login page.
   3. **Exception Flow(s)** – Information is input incorrectly leading to error messages.
6. **Includes** – User Login
7. **Preconditions** – user must not already be registered.
8. **Post conditions** – User is registered
9. **Notes/Issues** - none

## Profile Creation

1. **Objective** – User creates a profile that includes information about them such as name, phone, email, age, sex, and address.
2. **Priority** – High
3. **Source** – Carl Gnome (marketing)
4. **Actors** – Users/Administrators
5. **Flow of Events** 
   1. **Basic Flow**
      1. User is presented with a profile creation form.
      2. User fills in the stated information.
      3. User selects finished.
      4. User is redirected to the profile page
   2. **Alternative Flow 1** – At step 5.1.2 if user inputs the wrong format of data and error will be thrown after selecting finished and the user will be instructed on what was wrong with his submission.

1. **Includes**
2. **Preconditions** – User is logged in
3. **Post conditions** – User is redirected to their newly created profile.
4. **Notes/Issues** - None

## Course Enrollment/Withdrawal

1. **Objective** – User can enroll or withdrawal from a list of available courses. If the course is full student can select to be placed in a waiting list.
2. **Priority** – High
3. **Source** – Eddy Van (Stakeholder)
4. **Actors** – Users/Administrators
5. **Flow of Events**
6. **Basic Flow**

6.1. User is presented with a list of current courses and a list of their currenlty enrolled courses along with the current number of open seats.

*6.2.* User selects a course

*6.3.* User selects enroll

*6.4.* User is enrolled in course if seating is available.

1. **Alternative Flow 1** – At step 6.2 if user is already enrolled in the course they will have the option to withdrawal instead.
2. **Alternative Flow 2** *–* At step 6.2 if the class is full, user can select to be placed on a waitlist for the course.
3. **Includes**
4. **Preconditions** – User is logged in
5. **Post conditions** – User is enrolled in a course, withdrawn from a course, or added to a waitlist in a course.
6. **Notes/Issues** - None

# Other Nonfunctional Requirements

## Performance Requirements

* The system should respond to user interactions within 2 seconds under normal load and conditions.
* System will have an uptime of at least 99%
* System will ensure high availability and not exceed 5 hours of downtime a month

## Safety Requirements

Being a software system, no physical harm is a risk.

## Security Requirements

* The system will use SHA-2 hash algorithm for password hashing to ensure no password is the same and to make it harder for attackers to gain access (Berkeley, 2024).
* The system will protect user information from unauthorized access

## Software Quality Attributes

* The systems priorities are high availability and reliability to ensure students can access their courses at any convenience.
* System must be maintainable and update for future requirements and necessary Security updates.
* The system will be prioritizing ease of use since not everyone is adept in computer knowledge.

# Other Requirements

None that I can think of at the moment.

# System Requirements Chart

Appendix A: Glossary

ID: unique identifier for users and admin class profiles

Profile: User specific page where information is stored and displayed

Wait List: A list of students waiting to enroll in a currently full course.

Appendix B: Analysis Models

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

**Don’t do any of these for CS421 SRS. You will create these models during the high level design deliverable.**

**Will do once instructed to create models.**

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

List here any open questions or things you know still need to be done to the SRS, but haven’t been addressed yet. (It’s okay to have things like that, especially in this CS421 project because we don’t have time to do everything.)

Im not currently sure if the system requirements chart was filled out correctly

I may also need to add more information to section 5 other nonfunctional requiremetns.