**SRS Document Development and GitHub**

Benjamin Brown

CST499: Capstone for Computer Software Technology Professor Joseph Rangitsch

March 17, 2025

***Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

**Software Requirements**

**Specification**

**for**

**<Project>**

**Version 1.0 approved**

**Prepared by Benjamin Brown**

**UAGC**

**03/17/2025**

***Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.***

# Table of Contents

[Table of Contents iii](#_TOC_250048)

[Revision History iv](#_TOC_250047)

1. [Introduction 1](#_TOC_250046)
   1. [Purpose 1](#_TOC_250045)
   2. [Document Conventions 1](#_TOC_250044)
   3. [Intended Audience and Reading Suggestions 1](#_TOC_250043)
   4. [Project Scope 1](#_TOC_250042)
2. [Overall Description 2](#_TOC_250041)
   1. [Product Perspective 2](#_TOC_250040)
   2. [Product Features 2](#_TOC_250039)
   3. [User Classes and Characteristics 2](#_TOC_250038)
   4. [Operating Environment 2](#_TOC_250037)
   5. [Design and Implementation Constraints 3](#_TOC_250036)
   6. [User Documentation 3](#_TOC_250035)
   7. [Assumptions and Dependencies 3](#_TOC_250034)
3. [System Features 3](#_TOC_250033)
   1. [User register 3](#_TOC_250032)
      1. [Description and Priority 3](#_TOC_250031)
      2. [Stimulus/Response Sequences 3](#_TOC_250030)
   2. 3.1.3 Functional Requirements 4
   3. [User Log in 4](#_TOC_250029)
      1. [Overview and Priority 4](#_TOC_250028)
      2. [User Interaction Flow 4](#_TOC_250027)
      3. [Functional Requirements 4](#_TOC_250026)
   4. [User Logout 4](#_TOC_250025)
      1. [Feature Overview and Priority 4](#_TOC_250024)
      2. [User Interaction Flow 5](#_TOC_250023)
      3. [System Requirements 5](#_TOC_250022)
   5. [Waitlist Notification 5](#_TOC_250021)
      1. [Feature Overview and Priority 5](#_TOC_250020)
      2. [User Interaction Flow 5](#_TOC_250019)
      3. [System Requirements 5](#_TOC_250018)
   6. [Course Registration Process 5](#_TOC_250017)
      1. [User Interaction Flow 5](#_TOC_250016)
      2. [System Requirements 6](#_TOC_250015)
   7. [Course Dropping Process 6](#_TOC_250014)
      1. [Feature Overview and Priority 6](#_TOC_250013)
      2. [User Interaction 6](#_TOC_250012)
      3. [System Requirements 6](#_TOC_250011)
4. [External Interface Requirements 7](#_TOC_250010)
   1. [User Interfaces 7](#_TOC_250009)
   2. [Hardware Interfaces 7](#_TOC_250008)
   3. [Software Interfaces 7](#_TOC_250007)
   4. [Communications Interfaces 7](#_TOC_250006)
5. [Other Nonfunctional Requirements 7](#_TOC_250005)
   1. [Performance Requirements 7](#_TOC_250004)
   2. [Safety Requirements 8](#_TOC_250003)
   3. [Security Requirements 8](#_TOC_250002)
   4. [Software Quality Attributes 8](#_TOC_250001)

[Appendix A: Glossary 8](#_TOC_250000)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Benjamin Brown | 2/17/2025 | Creation date | 0.1 |
| Benjamin Brown | 3/17/2025 | Requirement revisions | 1.1 |

# Introduction

## Purpose

This Software Requirements Specification (SRS) document is designed to outline the requirements for the development of a student course enrollment system. With the new system, students can manage their own course registrations, reducing administrative workload and improving their overall experience. For registration, the system will provide essential features such as enrollment, course search, schedule management, and waitlist capabilities.

## Document Conventions

In the SRS, five sections cover different aspects of the system's development:

1. **Introduction**
2. **General Overview**
3. **External Interface Requirements**
4. **System Features**
5. **Non-Functional Requirements**

There are further subsections in each of these sections that describe the objectives of the system, the functionality, as well as the design, in greater detail.

## Intended Audience and Reading Suggestions

The purpose of this document is to provide guidance to the entire project team, including the developers, testers, and project managers. The document can also be used by stakeholders such as system administrators, users, and everyone involved in implementing and deploying the course enrollment system. This document should be reviewed by all team members for an understanding of the system's scope and requirements.

## Project Scope

Using the student course enrollment system, students can efficiently manage their academic schedules. By automating course registration, the system will enable functions like course search, enrollment, and schedule management. This will reduce errors, make the process more transparent, and make it less time-consuming. As well as increasing awareness about course availability, the system also ensures a smooth registration experience to boost student retention.

# Overall Description

## Product Perspective

The system is an online platform created to simplify the enrollment process for courses. Initially, it will be an independent system, but eventually it will be integrated with existing school infrastructure. Students can register and manage their schedules from anywhere using the web-based system, which is compatible with major browsers and mobile devices.

## Product Features

There will be several key functionalities included in the system, such as course search, enrollment, schedule management, and waitlisting. Account creation and management, searching for available courses, enrolling in them, and viewing current schedules will all be possible for students. In the event that a course is full, students have the option of joining a waitlist. The system will notify students when a spot becomes available, automating the process and minimizing manual intervention.

## User Classes and Characteristics

Three main user types will engage with the system:

* **Students**: Set up an account, sign in, search for and enroll in courses, and manage schedules.
* **Administrators**: Provide account management, login support, and system monitoring.
* **Registration Staff**: Make sure course data is updated and added or removed from available offerings.

To ensure security and functionality, each role will have specific access rights and permissions.

## Operating Environment

To ensure that all students have access to the system, the system should support multiple platforms, from mobile devices to desktops. Cross-browser compatibility and accessibility features, such as screen reader support, are essential. Information about users will be protected with encryption to ensure data security. It should also be scalable in the future, allowing it to accommodate more students and features as they become available.

## Design and Implementation Constraints

## User Documentation

For individuals seeking help or instructions, comprehensive user manuals should be provided. In addition, a virtual assistant powered by artificial intelligence will be integrated, so that users can easily obtain support whenever they need it.

## Assumptions and Dependencies

This project has a six-month timeline, a $100,000 budget, and a development team of eight at most. In order to accommodate future requirements, the system must be scalable and secure. This system can be integrated with minimal changes to the institution's current infrastructure.

# System Features

## User register

### Description and Priority

* + - * **Priority**: High
      * **Description**: For accessing core features like course enrollment and management, users must create an account. All new users must create an account. For users to engage with the platform, this feature serves as a foundation.
      * **Benefit**: 9

### Stimulus/Response Sequences

* Upon arriving at the homepage, the user is prompted to log in or create an account.
* Clicking "Register" initiates the registration process.
* The user completes the registration form with necessary information.
* A success message is displayed upon submission of the data or specific instructions are presented to correct any errors.
  1. **Functional Requirements**
* **REQ-1**: There should be two distinct buttons on the homepage: one for logging in and one for registering.
* **REQ-2**: Upon clicking the "Register" button, the user should be redirected to a registration form.
* **REQ-3**: Before storing input data in the database, the registration form should validate and sanitize it.
* **REQ-4**: When the user's input is invalid or incomplete, the system must display clear, actionable error messages.
* **REQ-5**: After successful registration, the system must notify the user that their account has been created with a confirmation message.

## User Log in

### Overview and Priority

* + - * **Priority**: High
      * **Overview**: The user must be able to access his or her account by logging in. Logging in is crucial for accessing the site's core features, including viewing and managing course schedules.
      * **Benefit**: 9

### User Interaction Flow

* + - * Logging in or creating an account is prompted upon visiting the homepage.
      * The user selects the log in option.
      * The user provides log in credentials.
      * The user receives a message confirming successful login or clear instructions on how to correct any input errors.

### Functional Requirements

* + - * **REQ-1**: The homepage should feature buttons for both login and registration.
      * **REQ-2**: A login form appears after the user selects the "login" button.
      * **REQ-3**: In order for the data to be sent to the database when the form is submitted, the data must be validated, sanitized, and processed by the system.
      * **REQ-4**: Invalid input should be communicated clearly to the user so that they can correct the information if necessary.
      * **REQ-5**: In order to ensure accuracy, the credentials of the user must be cross-checked against the database.
      * **REQ-6**: Upon successful login, the system should show a confirmation message.

## User Logout

### Feature Overview and Priority

* + - * **Priority**: High
      * **Overview**: Users must be able to log out of the platform when not using it, ensuring the security of their accounts. Data protection is a core function.
      * **Benefit**: 9

### User Interaction Flow

* + - * Selecting "Logout" from the navigation bar allows the user to log out.
      * A user's session is terminated, and they are directed to the homepage.

### System Requirements

* + - * **REQ-1**: It should be possible to log out from the navigation bar.
      * **REQ-2**: By clicking this button, the user will be logged out and their session will be terminated.
      * **REQ-3**: Following logging out, the user should be directed to the homepage.

## Waitlist Notification

### Feature Overview and Priority

* + - * **Priority**: Medium
      * **Overview**: In the event that a course from their waitlist becomes available, the user should be notified.
      * **Benefit**: 7

### User Interaction Flow

* + - * Using the system, the user logs in.
      * Whenever a course becomes available, a notification appears on the page.
      * Upon receiving the notification, the user acknowledges it.

### System Requirements

* + - * **REQ-1**: Notifications can only be received by logged-in users.
      * **REQ-2**: The system will display a notification at the top of the page if a user is moved from a waitlist into a registered course once logged in.

## Course Registration Process

**Feature Overview and Priority**

* **Priority**: High
* **Overview**: Registration for courses must be possible via the platform. Managing a student's course load is a crucial feature.
* **Benefit**: 9

### User Interaction Flow

* + - * Users can register for courses by clicking on the "Register for Courses" link on the course schedule.
      * Drop-down menus allow users to select the semester they wish to enroll in.
      * When a user clicks “Submit” the system returns to a list of courses that are available.
      * Selecting courses to register for requires the user to check the checkboxes..
      * The user clicks "Submit" to confirm their selections.
      * If a course is available, the user receives confirmation of successful registration.
      * If a course is full, the user is notified that they’ve been placed on the waitlist.
      * When the user confirms the registration status message, they are redirected back to the schedule of courses.

### System Requirements

* + - * **REQ-1**: There should be a link on the course schedule page to "Register for Courses.".
      * **REQ-2**: Users should be able to select the semester for course registration from the drop-down menu.
      * **REQ-3**: As soon as the submission is made, the system should return a list of available courses for the semester selected.
      * **REQ-4**: Checkboxes must be provided for selecting courses.
      * **REQ-5**: For course registration to be completed, a "Submit" button must be available.
      * **REQ-6**: It is important that the system processes selected courses and displays the status (registered or waitlisted).
      * **REQ-7**: If you register or waitlist, you should see a confirmation message.
      * **REQ-8**: Users should be redirected to the course schedule page once the status has been confirmed.

## Course Dropping Process

### Feature Overview and Priority

* + - * **Priority**: High
      * **Overview**: Dropping courses from the schedule must be possible for users. It is crucial for managing course enrollments to have this feature.
      * **Benefit**: 9

### User Interaction

* + - * Using the navigation bar, the user selects "View Course Schedule".
      * The user views their course schedule.
      * Checkboxes are displayed next to the courses that the user wants to drop.
      * After selecting the courses, the user clicks "Submit".
      * After successfully dropping courses, the user receives a confirmation message.
      * Once the drop message is confirmed, the user will be redirected to the course schedule.

### System Requirements

* + - * **REQ-1**: On the navigation bar, there must be a button called "View Course Schedule".
      * **REQ-2**: By clicking this button, the user will be taken to their course schedule.
      * **REQ-3**: There should be a checkbox for dropping courses in the system.
      * **REQ-4**: To complete the drop action, there must be a "Submit" button.
      * **REQ-5**: Dropping a course should be processed by the system and a confirmation message should be displayed.
      * **REQ-6**: Upon confirmation of the drop, the user should be redirected to the course schedule.

# External Interface Requirements

## User Interfaces

An easy navigation experience will be provided by a simple yet intuitive user interface. In the layout, there will be a header, a footer, and clearly labeled action buttons. A mobile-responsive interface, alt texts for images, and accessible navigation will ensure inclusivity.

## Hardware Interfaces

A variety of devices will be able to access the system, including smartphones, tablets, and desktop computers. Mobile devices are commonly used by students, so the platform will be optimized for both mobile and desktop devices. In addition, the system will run on all major operating systems, including iOS, Android, Windows, and macOS.

## Software Interfaces

In order for the system to function seamlessly within the institution's environment, it must be integrated with the existing database. Data on students, courses, and registration will be stored in this database. APIs will also be used to connect the front-end interface with back-end databases, ensuring smooth communication and real-time data updates.

## Communications Interfaces

To protect student privacy, communications, especially those involving sensitive data, will be encrypted. Students will receive automatic notifications about enrollment status updates when important changes occur (such as course availability). To maintain clear communication channels, email and system alerts will be used.

# Other Nonfunctional Requirements

## Performance Requirements

It should take less than 60 seconds to create an account. Three seconds is the maximum time that should be taken for search and confirmation actions.

## Safety Requirements

It is important to encrypt personal data. Inactivity will result in a session timeout after five minutes.

## Security Requirements

Encrypting passwords and PII is a must. It is essential to verify user credentials before granting access to sensitive data. The Gramm-Leach-Bliley Act (GLBA) and General Data Protection Regulation (GDPR) must be followed in order to protect personal financial information. To safeguard consumers' sensitive data and maintain trust, financial institutions must implement privacy policies and practices..

## Software Quality Attributes

There should be ease of navigation and reliability in the system. Scalability and accessibility guidelines should be met so that it can handle increased enrollment.

# Appendix A: Glossary

PII- Personally Identifiable Information SRS- Software Requirements Specification TCP- transmission control protocol

UI- User Interface UX- User Experience

**References:**

*Federal Trade Commission. (n.d.). Gramm-Leach-Bliley Act. Federal Trade Commission.* [*https://www.ftc.gov/legal-library/browse/statutes/gramm-leach-bliley-act*](https://www.ftc.gov/legal-library/browse/statutes/gramm-leach-bliley-act)

*Wiegers, K. E. (1999). Software requirement specifications for [Template].* [*https://web.cs.dal.ca/~hawkey/3130/srs\_template-ieee.doc*](https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc)

Federal Trade Commission. (n.d.). *General Data Protection Regulation (GDPR)*. Federal Trade Commission. <https://www.ftc.gov/legal-library/browse/statutes/gramm-leach-bliley-act>