**Week 1 Assignment: SRS Document Development and GitHub**

Shaun Hoadley

CST499: Capstone for Computer Software Technology

Professor Amr Elchouemi

January 13, 2022

Software Requirements Specification

for

Course Enrollment System

Version 1.0 approved

Prepared by Shaun Hoadley

CST499 UAGC

January 13, 2022

Table of Contents

Table of Contents iii

Revision History iii

1. Introduction 4

1.1 Purpose 4

1.2 Document Conventions 4

1.3 Intended Audience and Reading Suggestions 4

1.4 Product Scope 4

2. Overall Description 4

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Classes and Characteristics 5

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

3. External Interface Requirements 6

3.1 User Interfaces 6

3.2 Hardware Interfaces 6

3.3 Software Interfaces 6

3.4 Communications Interfaces 6

4. System Features 6

4.1 New User Account Creation 6

4.2 User Login (for existing accounts) 7

4.3 Search for Available Courses 7

4.4 View Course Schedule 8

4.5 Register a Course 8

4.6 Drop a Course 9

4.7 Waitlist Course Available Notification 9

4.8 User Logout 10

5. Other Nonfunctional Requirements 10

5.1 Performance Requirements 10

5.2 Safety Requirements 11

5.3 Security Requirements 11

5.4 Software Quality Attributes 11

5.5 Business Rules 11

Appendix A: Glossary 11

Appendix B: References 12

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this SRS (Software Requirements Specification) document is to provide an outline of the student course enrollment system being developed for the CST499 final project. Through the use of a self-service website, students should be allowed the ability to manage their course schedules. The scope of this SRS document covers the initial build of the student course enrollment system in its entirety, including the site’s basic features and the non-functional requirements described further down in this document.

## Document Conventions

There will be five sections in this document with the following headers: Introduction, Overall Description, External Interface Requirements, System Features, and Other Non-Functional Requirements. Each section has sub-headers to provide the detail of the specifics for those sections.

## Intended Audience and Reading Suggestions

All audience members are encouraged to read this SRS document entirely to gain an understanding of the purpose and requirements of the student course enrollment system. The audience for which this document is intended includes project managers, software architects, software designers, software developers, UI/UX (user interface / user experience) designers, software testers, and all other stakeholders involved with this project.

## Product Scope

The goal of this project is to allow students to quickly and easily manage their course schedules. Students should be able to create and register an account, log into and out of their account, view their current schedule, search available courses, register for a course or be added to the waitlist of full classes, and drop courses from their schedule. This project aligns with the business objectives to retain students, improve the student experience, increase student enrollment, and reducing costs and overhead of the manual course registration process currently being used.

# Overall Description

## Product Perspective

The student course enrollment system build is a new and self-contained project.

## Product Functions

Users should have the ability to create and register a new account, log into and out of their account, view their course schedule, search for available courses, register for a course or be added to a waitlist if the course is full, and drop courses from their schedule. Courses have a maximum number of students allowed to be enrolled and the website should reflect that maximum. If a student registers for a course that is full, they should be added to a waitlist. Furthermore, if a course has an opening become available, the site should notify the student of the change.

## User Classes and Characteristics

There will be three user classes and a course class. The user classes are student, administrator, and Registrar staff. For the student class, it should be able to create an account, log into or out of the account, view course schedules, search for courses that are available, and register for or drop a course. The Registrar staff class needs to be able to add, update, or remove the courses that are available. The administrator class needs access to all of the functions. Administrators need to be able to assist other users with logging into their accounts or reset their login information. Additionally, administrators should be able to look up student schedules by using their student ID number. The course class will contain the attributes and methods used to manipulate courses, such as creating and deleting course enrollments.

## Operating Environment

The student course enrollment system should be developed with cross-platform support so it may be used with all major web browsers and be responsive on mobile devices.

## Design and Implementation Constraints

After deployment, the school will be maintaining the system so it needs to support maintainability, modifiability, and testability. The timeframe for the project build is six months or less. The system must utilize relational databases and support encryption for sensitive data and passwords. The project must abide by ADA web accessibility (ADA, 2007) and FERPA (Family Education Rights and Privacy Act) guidelines to comply with school policies.

## User Documentation

The project needs to include documentation to aid users needing guidance or support using the system. Furthermore, the system will use an automated chatbot for users to request assistance as needed.

## Assumptions and Dependencies

The budget for the student course enrollment system project is $100,000. The deadline for completion is six months. The project team must consist of no more than eight people including project managers, software architects, software developers, software designers, software testers, and database administrators.

# External Interface Requirements

## User Interfaces

The student course enrollment system website should contain the same header, footer, and navigation bar on each page. The layout of each page should be more or less consistent. The home button on the navbar should be clearly visible on every page. Session information should not be lost when moving through pages and returning to previous pages. Users should be able to tab through pages and all images should contain alt tags. All forms should provide clear instructions for correcting incorrect data inputs from users. The selection of colors and fonts for the website should be done with readability, useability, and visual appeal in mind.

## Hardware Interfaces

The student course enrollment system should be accessible from all major devices such as phones, tablets, PCs and Macs. The project should support screen readers. HTTP and PHP will be used to handle requests between the client and web server and TCP for resource transfers.

## Software Interfaces

The project should work consistently on Mac, Microsoft, Linux, Android, and IOS operating systems. Users should be able to sign on using their school provided email accounts. The project should contain an API to work with the relational databases for storing and retrieving user and course data.

## Communications Interfaces

All PII (personally identifiable information) and passwords should be secure and encrypted. Users should receive notifications to confirm a course has been added or dropped from their schedule as well as when an opening becomes available from waitlists they are on.

# System Features

## New User Account Creation

4.1.1 Description and Priority

* Priority: High
* Description: Users should be able to create an account. A user cannot view or manage course schedules without an account. This is a part of the basic functionality of the website.
* Benefit: 9

4.1.2 Stimulus/Response Sequences

* User prompted to login or register on the home page.
* User selects register (create new account).
* User enters form data.
* User selects submit form data.
* User receives confirmation of success or clear details on correcting errors.

4.1.3 Functional Requirements

* REQ-1: Website should have two buttons on home page, one for login and one for register.
* REQ-2: Website should open the registration form page when user clicks on “register”.
* REQ-3: The data from the form is validated, sanitized, processed, and sent to the database when a user hits “submit”.
* REQ-4: Website provides informative messages if input does not meet requirements.
* REQ-5: User is given a success message when account has been successfully created.

## User Login (for existing accounts)

4.2.1 Description and Priority

* Priority: High
* Description: Existing users should be able to login to their account. A user cannot view or manage course schedules without an account. This is a part of the basic functionality of the website.
* Benefit: 9

4.2.2 Stimulus/Response Sequences

* User prompted to login or register on the home page.
* User selects login.
* User enters form data.
* User selects submit form data.
* User receives confirmation of success or clear details on correcting errors.

4.2.3 Functional Requirements

* REQ-1: Website should have two buttons on home page, one for login and one for register.
* REQ-2: Website should open the login form page when user clicks on “login”.
* REQ-3: The data from the form is validated, sanitized, processed, and sent to the database when a user hits “submit”.
* REQ-4: Website provides informative messages if input does not meet requirements.
* REQ-5: User credentials verified against database.
* REQ-6: User is given a success message when account has been successfully created.

## Search for Available Courses

4.3.1 Description and Priority

* Priority: High
* Description: Users should be able to search for available courses. A user cannot a course without finding it through a search. This is a part of the basic functionality of the website.
* Benefit: 8

4.3.2 Stimulus/Response Sequences

* User selects “Register for Courses” link on the “View Course Schedule” page.
* User selects semester from drop down menu.
* User selects Submit.
* User is presented with search result list.

4.3.3 Functional Requirements

* REQ-1: Website should have a “Register for Courses” link on the “View Course Schedule” page.
* REQ-2: Website should present user a drop down menu with semester options.
* REQ-3: Website should return search results when user hits submit.
* REQ-4: Website should update search results when new drop down item is selected.

## View Course Schedule

4.4.1 Description and Priority

* Priority: High
* Description: Users should be able to view their course schedule. A user cannot view or manage course schedules without being logged into an account. This is a part of the basic functionality of the website.
* Benefit: 9

4.4.2 Stimulus/Response Sequences

* User selects “View Course Schedule” from the navigation bar.
* Website takes user to the “View Course Schedule” page.

4.4.3 Functional Requirements

* REQ-1: Website should have a “View Course Schedule” button on the navigation bar.
* REQ-2: When “View Course Schedule” button is selected, user is redirected to the “View Course Schedule” page.

## Register a Course

4.5.1 Description and Priority

* Priority: High
* Description: Users should be able to register for courses. A user cannot view or manage course schedules without being logged into an account. This is a part of the basic functionality of the website.
* Benefit: 9

4.5.2 Stimulus/Response Sequences

* User selects “Register a Course” link on the “View Course Schedule” page.
* User selects semester from drop down menu.
* User selects Submit.
* User is presented with search result list.
* User selects the register checkmark box for all desired courses in the list.
* User selects the submit button.
* If the course is not full, user receives confirmation that they have successfully
* registered for the course.
* If the course is full, user receives notification that they have been added to the waitlist for the course.
* When the user confirms the confirmation message, they are redirected to the “View Course Schedule” page.

4.5.3 Functional Requirements

* REQ-1: Website should have a “Register for Courses” link on the “View Course Schedule” page.
* REQ-2: Website should present user a drop down menu with semester options.
* REQ-3: Website returns search results when user selects submit.
* REQ-4: Website should offer checkboxes next to courses under the “Register” header.
* REQ-5: Website should have a “Submit” button.
* REQ-6: Website collects data from the table when “Submit” button is selected.
* REQ-7: Website should display a confirmation modal that user has either successfully registered, or have been added to the waitlist.
* REQ-8: Website should redirect user to “View Course Schedule” page after confirming the modal.

## Drop a Course

4.6.1 Description and Priority

* Priority: High
* Description: Users should be able to drop a course. A user cannot view or manage course schedules without being logged into an account. This is a part of the basic functionality of the website.
* Benefit: 9

4.6.2 Stimulus/Response Sequences

* User selects “View Course Schedule” on the navigation bar.
* User views their course schedule.
* User checks the “drop” checkbox next to all desired courses.
* User selects the “Submit” button.
* User receives confirmation that they have successfully dropped the course
* When the user confirms the confirmation message, they are redirected to the “View Course Schedule” page.

4.6.3 Functional Requirements

* REQ-1: Website should have a “View Course Schedule” button on the navigation bar.
* REQ-2: When “View Course Schedule” button is selected, user is redirected to the “View Course Schedule” page.
* REQ-3: Website should have checkmark boxes beside each course under the “Drop” header.
* REQ-4: Website should have a “Submit” button.
* REQ-5: Website should collect data from the table when the “Submit” button is selected.
* REQ-6: Website should display a confirmation modal that user successfully dropped courses.
* REQ-7: Website redirects user to “View Course Schedule” page after confirming the modal.

## Waitlist Course Available Notification

4.7.1 Description and Priority

* Priority: Medium
* Description: Users should be notified when a course becomes available from the waitlist.
* Benefit: 7

4.7.2 Stimulus/Response Sequences

* User logs into system.
* Website presents a notification at the top of the page when user has been moved from a waitlist and registered to the course.
* User confirms notification.

4.7.3 Functional Requirements

* REQ-1: User logs into the system.
* REQ-2: Website presents a notification at the top of the page once a user is logged in notifying them that they have been registered for a course that they were on the waitlist for

## User Logout

4.8.1 Description and Priority

* Priority: High
* Description: Users should be able to logout of an account. User data cannot be protected if they cannot log out of their account. This is a part of the basic functionality of the website.
* Benefit: 9

4.8.2 Stimulus/Response Sequences

* User selects “Logout” from the navigation bar.
* User logged out and session terminated from the student course enrollment system.
* Website redirects to home page.

4.8.3 Functional Requirements

* REQ-1: Website should have a “Logout” button on the navigation bar.
* REQ-2: Website should log a user out and terminate the session when the “Logout” button is selected.
* REQ-3: Website should redirect the user to the home page after logging out.

# Other Nonfunctional Requirements

## Performance Requirements

A new user should be able to create an account in under sixty seconds. Search results should take no more than three seconds to come back for user to view. Confirmation messages from registering or dropping courses should appear within three seconds. Speed is vital to giving users a good experience and alleiviating some of the overhead in the Registrar’s Office. Delays may detract users from returning to the site and causing more overhead in the Registrar’s Office.

## Safety Requirements

All PII should be protected, and passwords should be encrypted. The system needs to follow FERPA guidelines. Sessions should timeout after 5 minutes of inactivity, without user confirmation to continue the session.

## Security Requirements

All PII should be protected and secure. Credentials should be verified before a user is given access to account information. Passwords must be encrypted.

## Software Quality Attributes

The Student Course Enrollment System should follow web accessibility guidelines outlined by ADA (n.d.). Additionally, the site should be easy to navigate and use. As interruptions in service may deter users from returning, the website should be reliable. The site should be scalable to accommodate for increased enrollment. It is important that the site be modifiable, to allow for the addition of new features, and the product should be maintainable and testable, as maintenance will be the responsibility of the client.

## Business Rules

Users should not be allowed to view their schedule or register/drop courses without being logged in. Administrators should have the ability to assist users in resetting credentials. They should also be able to search for student information and schedules using the student ID number. Finally, administrators should be able to retrieve a list of all students, enrollment, dropped courses, and course availabilities, for reporting purposes.

Appendix A: Glossary

ADA- Americans with Disabilities Act of 1990

FERPA- Family Educational Rights and Privacy Act

HTTP- hypertext transfer protocol

PII- Personally Identifiable Information

SRS- Software Requirements Specification

TCP- transmission control protocol

UI- User Interface

UX- User Experience

**References**

ADA. (2007). Website Accessibility Under Title II of the ADA. Retrieved January 15, 2022 from https://www.ada.gov/pcatoolkit/chap5toolkit.htm

U.S. Department of Education. (n.d.). Family Educational Rights and Privacy Act (FERPA). Retrieved January 15, 2022 from <https://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html>

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