SRS Document Development and GitHub

Jamain Hughes

The University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Professor Amjad Alkilani

December 18, 2023

Software Requirements Specification

for

Student Registration System

Version 1.0 approved

Prepared by Jamain Hughes

CST 499 UAGC

December 17, 2023

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

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3. External Interface Requirements 4

3.1 User Interfaces 4

3.2 Hardware Interfaces 5

3.3 Software Interfaces 5

3.4 Communications Interfaces 5

4. System Features 6

4.1 Account Creation…………………………………………………………………………………..6

4.2 User Login………………………………………………………………………………………..7

4.3 Serach for Available Courses……………………………………………………………………..8

4.1 View Course Schedule……………………………………………………………………………..9

4.1 Register Course...………………………………………………………………………………..10

4.1 Drop Course……………………………………………………………………………………..11

4.1 Waitlist Course Notification……...……………………………………………………………..12

4.1 User Logout……………………………………………………………………………………..13

5. Other Nonfunctional Requirements 14

5.1 Performance Requirements 14

5.2 Safety Requirements 14

5.3 Security Requirements 14

5.4 Software Quality Attributes 15

5.5 Business Rules 15

6. Other Requirements 15

Appendix A: Glossary 15

Appendix B: References 16

Appendix C: GitHub……………………………………………………………………………17

# Introduction

## Purpose

The purpose of this Software Requirement Specification (SRS) document is to outline the functional and non-functional requirements for the development of an student registration system that provides a way for students to enroll into courses for an online institute.

## Document Conventions

This document will have five sections with the following headers: Introduction, overall description, external, Interface requirements, system features, and other non-functional requirements. Each section will have sub-headers to provide details of each section.

## Intended Audience and Reading Suggestions

This SRS document is designed for the development team, project managers, quality assurance team, system designers, and stakeholders involved in the design and implementation of the Student Registration System. This document is to aid in the understanding of the purpose and requirements of the registration system. Due to this reason all members are encouraged to read this document.

## Product Scope

This project is essentially a student registration a student registration system. The overall goal of this is to allow students to easily manage their course schedules. Students will be able to register as a new student and create their account and profiles. Students will also be able to enroll themselves onto courses if they are available during the respected semester. If the desired course is not available, students can add themselves onto a waiting list. Students will also be able to cancel their enrollment from any course that they are enrolled in, and the system should inform the first in the waiting list that they can enroll into the class. This project aligns with the business objectives to hold onto students, improve the student’s online college experience, and reducing costs while increasing the speed of the college registration process compared to manually registering students.

# Overall Description

## Product Perspective

The student registration system build is a new self-contained project.

## Product Functions

This system should allow users to do the following actions:

* Create and register a new account.
* Log into and out of their account.
* Search for available courses.
* View their course schedule.
* Register for a course or be added to a waitlist if the course is unavailable.
* Drop courses from their schedule
* Added to a waitlist if course is unavailable.
* Be informed once a course is available (if on waiting list).

## User Classes and Characteristics

The software system will have three user classes and one course class. These classes will be known as the administrator, students, and the Registrar Staff class. The student class shall be able to create an account, log into or out of their account, view course schedules, search for courses that are available, and register for or drop a course. The registrar staff class shall be able to remove, update, or add courses that are available. The administrator class shall be able to assist the users from the student class with any technical difficulties they are having. This may include help with logging into accounts and resetting login information. The course class shall have the attributes and methods used to manipulate courses. This manipulation will consist of creating and deleting course enrollments.

## Operating Environment

The student registration system should be available across all platforms. All major web browsers should be able to use the system including web browsers on mobile devices.

## Design and Implementation Constraints

The project must by the policies of ADA web accessibility (ADA, 2007) and FERPA (Family Education Rights and Privacy Act). The project should be built within 3 months. The system will utilize relational databases through MySQL. The system will also utilize four primary programming languages PHP, HTML5, CSS, and JavaScript. Users will be trusting the application with their sensitive information so the application should support data encryption. After the deployment of the software system, the educational institute will be responsible for maintain the system so, this project need to have the attributes of maintainability, modifiability, and testability.

## User Documentation

The project needs to include documentation to aid users needing guidance or support using the system. The system will have pop up windows showing users what to do during the registration process.

## Assumptions and Dependencies

The student registration system is projected to take three months. The budget for the system is $200,000. This project will facilitate the use of the Scrum framework which will require seven staff personnel and three more people if needed. Due to the various given from the planning stage of the software product, there is no need for a large team. The team will consist of a project manager, software architects, software developers, software designers, software testers, and database administrators.

# External Interface Requirements

## User Interfaces

The student registration website should have a home page that displays the name of the school with an image of the primary building to where the institute resides. The website should be themed with the school colors which is maroon and yellow. The hex code for the maroon color is #800000 and the Canadian yellow hex number is #FDDA0D. The layout of every page should be consistent with the same header, footer and navigation bar on each page. There should be a home button available on the navbar on every single page. Session data should last until the user exits the website. All forms should provide clear instructions for correcting incorrect data inputs from users.

## Hardware Interfaces

The student registration system should be accessible from every major device like cellphone, tablets, PC’s, and Macs both laptops and desktops. Data shall be transferred using Transmission Control Protocol (TCP/IP). The website will be loaded using Hypertext Transfer Protocol (HTTP). PHP will be used to as the back end scripting language.

## Software Interfaces

The project should work on all of the major web browsers such as Chrome, Opera, Vivaldi, Firefox, Microsoft Edge, Epic, Safari, Brave, Internet Explorer, and Blisk. The project should also work on all operating systems such as Microsoft, Mac, Linx, Android, and IOS systems. As users use their email address to register their account, they will be able to receive email notification to the corresponding email via PHP.

## Communications Interfaces

The project will send notifications via email using HTTP. This email should notify students once a requested course becomes available.

# System Features

## Account Creation

4.1.1 Description and Priority

* Priority: High
* Description: Users shall create an account. Without an account, the students won’t be able to enroll into a class.
* Benefit:9

4.1.2 Stimulus/Response Sequences

* Users is directed to a login or registration form on the home page.
* User selects register to create new account.
* User enters their information into data form.
* User submits data.
* User receives either confirmation or error notification.

4.1.3 Functional Requirements

* + - REQ-1: Website should have clickable login and clickable register button on home page.
    - REQ-2: Website should direct user to a registration form upon clicking the register button.
    - REQ-3: Website should direct user to their profile page upon logging in.
    - REQ-4: Website shows an error massage if registration data is not correct.
    - REQ-5: User will be given a success message when account has been created successfully.

## User Login (For existing accounts)

4.2.1 Description and Priority

* Priority: High
* Description: This is one of the basic functionalities of the website. After a user has created their account, they need to have continuous access to that account so they can enroll or drop classes..
* 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 submits form data.
* User receives confirmation of a successful login.
* User is directed to their profile after successfully logging in.

4.2.3 Functional Requirements

* + - REQ-1:Website should have a login and a register button.
    - REQ-2: Website should display a login form upon clicking login.
    - REQ-3: The data from the form is validated, sanitized, processed, before being sent to the database upon entering the submit button.
    - REQ-4: Website provides a message informing user if their character input is correct.
    - REQ-5: The user credentials must match with the data within the database.
    - REQ-6: Website gives user a success massage after successfully login in followed by a redirection to the user’s profile page.

## Search for available courses

4.3.1 Description and Priority

* Priority: High
* Description: This is one of the essential functions of the website. Users can not enroll into courses without being able to search for available courses.
* Benefit:7

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 shall have a “Register for Courses” link on the schedule page.
    - REQ-2: Website should have a semester drop down menu.
    - REQ-3: Website should display the search results.
    - REQ-4: Website should keep search results updated.

## View Course Schedule

4.4.1 Description and Priority

* Priority: High
* Description: This is an essential functionality of the website. Users should be able to view their course schedules.
* Benefit:8

4.4.2 Stimulus/Response Sequences

* User selects the “view course schedule” from the navigation bar.
* Website takes user to the “view course schedule” page.

4.4.3 Functional Requirements

* + - REQ-1: Website shall have a “view course schedule” button on the navigation bar.
    - REQ-2: User is redirected to the “view course schedule” button.

## Register Course

4.5.1 Description and Priority

* Priority: High
* Description: This is the basic functionality of the website. Users should be able to register for courses in order to participate in their program.
* 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 the submit button.
* User is able to see the search result list.
* User is able to select a register checkmark box for desired courses in the list.
* User selects the submit button.
* Students will receive confirmation that they have successfully submitted registered for their course if it is not full.
* When the user confirms the confirmation message, they are redirected to the “View Course Schedule” page.

4.5.3 Functional Requirements

* + - REQ-1: Website shall have a “Register for Courses” link on the “View Course Schedule” page.
    - REQ-2: Website shall a drop-down menu with semester options to users.
    - REQ-3: Website shall return search results when user selects submit.
    - REQ-4: Website collects data from the table when the “Submit” button is selected.
    - REQ-5: Website should have submitted button available for users.
    - REQ-6: Website shall collect data from the table when the “Submit” button is selected.
    - REQ-7: Website shall display if the user have successfully registered or if the user is added to the waitlist.
    - REQ-8: Website shall redirect user to the “View Course Schedule” page after confirmation.

## Drop Course

4.6.1 Description and Priority

* Priority: High
* Description: This is the basic functionality of the website. Users should be able to drop a course.
* Benefit:9

4.6.2 Stimulus/Response Sequences

* User selects the “View Course Schedule" in the navigation bar.
* Users view their course schedule.
* User selects the drop checkbox beside their course.
* User selects the “Submit” button.
* Users receive confirmation of their dropped class.
* After confirmation, the user is redirected to the “view course schedule” page.

4.6.3 Functional Requirements

* + - REQ-1: Website shall have a “View Course Schedule” button on the navigation bar.
    - REQ-2: Upon selection, user is redirected to the “View Course Schedule” page.
    - REQ-3: Website shall have checkmark boxes beside each course under the “Drop” header.
    - REQ-4: Website shall have a “Submit” button.
    - REQ-5: Website should collect data from the table user selects the “Submit” button.
    - REQ-6: User shall see a confirmation of successfully dropped class.
    - REQ-7: Website shall redirect user to the “View Course Schedule" page upon successful completion.

## Waitlist Course Availability Notification

4.7.1 Description and Priority

* Priority: Medium
* Description: User should receive a notification via email once a course becomes available from the waitlist.
* Benefit:7

4.7.2 Stimulus/Response Sequences

* User log into system.
* Website shall present a notification at the top of the page once the 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 system.
    - REQ-2: Website shall present a notification at the top of the page once the user is logged in notifying them that they have been registered for a course that they were put on the waiting list for.
    - REQ-3: Website should send email to user notifying the user that they are have been added to a course.

## User Logout

4.8.1 Description and Priority

* Priority: High
* Description: This is a basic functionality of the website. Users should be able to logout from there account once they are done using it in order to promote security.
* Benefit:9

4.8.2 Stimulus/Response Sequences

* User shall be able to select “Logout” from the navigation bar.
* User logged out and session terminated from the student registration system.
* Website redirect user from home page.

4.8.3 Functional Requirements

* + - REQ-1: Website shall have a “Logout” button on the navigation bar.
    - REQ-2: Website shall log a user out and terminate the session upon clicking the “Logout: button.
    - REQ-3: Website shall redirect user to home page after log out.

# Other Nonfunctional Requirements

## Performance Requirements

Time is an important aspect of software applications. End users should have the best experience possible. When a software system is too slow, end users will seek other software systems that offer the same services but with quicker reactions. Because of this fact, this software system should allow new end users to create an account in under sixty seconds. Search results should also take no more than three seconds to come back to end users. Confirmation of registering or dropping a class should take three seconds.

## Safety Requirements

This system shall follow FERPA guidelines. All PII (Personal Identity Information) should be protected which involves encrypting passwords. Session data should automatically time out after 5 minutes of inactivity.

## Security Requirements

All users PII data should be secured. All users’ credentials should be verified before gaining access to account information and passwords must be encrypted.

## Software Quality Attributes

The student registration system should follow the web accessibility guidelines outlined by ADA (n.d.). The website should promote usability and reliability because users will get deter from the website and seek services elsewhere due to continuous inconveniences. The site should be scalable to accommodate an increasing number of end users. This site should be modifiable testable, and maintainable to allow maintenance, testing, and additions to features.

## Business Rules

Users should not be able 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 students ID number. 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

IP – Internet Protocol

PII – Personally Identifiable Information

TCP – Transfer Control Protocol

References

Tsui, F., Karam, O., & Bernal, B. (2018). [Essentials of software engineering](https://uagc.instructure.com/courses/126521/modules/items/6439323) (4th ed.). Jones & Bartlett

Learning.

GitHub Account:

I already had a GitHub account before beginning this course. Instead of creating a new account, I decided to use the account that I already have. Below are attachments of screenshots of my account and a new repository that I made for this assignment called Student-Registration-System. The URL for this repository is <https://github.com/Jamain31/Student-Registration-System.git>

