Final Software Project

Keyona Teamer

University of Arizona Global Campus

Capstone for Computer Software Technology: CST499

Dr. Charmelia Butler

08/31/2023

Table of Contents

Table of Contents 2

1. Software Requirements Specification 3

1.1 Purpose 3

1.2 Document Convention 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Product Scope 3

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 6

2.7 Assumptions and Dependencies 6

3. External Interface Requirements 6

3.1 User Interfaces 6

3.2 Hardware Interfaces 7

3.3 Software Interfaces 7

3.4 Communications Interfaces 7

4. System Features 7

4.1 Account Creation 5

4.2 User Login 6

4.3 Course Searchability 7

4.4 Course Registration 8

4.5 Course Withdrawal 8

4.6 Course Waitlist 9

5. Other Nonfunctional Requirements 11

5.1 Performance Requirements 11

5.2 Safety Requirements 12

5.3 Security Requirements 12

5.4 Software Quality Attributes 12

5.5 Business Rules 12

6. Testing Levels and UML Models 13

6.1 Introduction 13

6.2 Component Testing 13

6.3 Integration Testing 14

6.4 System Testing 14

6.5 Acceptance Testing 15

7. Landing, Login, and Enrollment Pages 16

8. MySQL Database and Class Registration 18

9. PHP Code 19

6. References 24

**Software Requirements Specifications**

# Introduction

## Purpose

The product for this Software Requirements Specification (SRS) document details the required elements for the Student Enrollment System. The system will ensure students can organize and control the courses they attend using a self-enrollment system. This is version 1.0 of the Student Enrollment System. The SRS will include the entirety of the initial system development. Section four of the SRS will describe essential elements of the Student Enrollment System and section 5 will identify the non-functional requirements.

## Document Conventions

The SRS includes five sections. There are headers for each section to introduce the section’s topic, provide descriptions, interface requirements, system features, and other non-functional requirements. The subtitles will offer detailed information about the section.

## Intended Audience and Reading Suggestions

This SRS document is intended for anyone that will assist in managing and building the system like, developers, architects, project managers, marketing staff, users, testers, and documentation writers and any other stakeholder who has an interest in the system. It is strongly recommended that all suggested parties read this documentation in its entirety to obtain an understanding about the system requirements and functionality.

## Product Scope

The objective of the Student Enrollment System is to ensure the user can view, schedule, and cancel courses for up to three semesters (spring/summer/fall). The user will have the ability to register and create a profile. Therefore, having the ability to login to the system with a unique ID and password, view their schedule, search for courses, register for a course, drop a course, and get waitlisted when a course is full. The user can also cancel enrollment in a class. This system addresses the business need for a product that allows users the ability to access courses, withdraw, enroll, and get waitlisted.

# Overall Description

## Product Perspective

The Student Enrollment System is a new, self-contained product.

## Product Functions

* User Registration Form that includes account and profile information.
* Each user has a user ID and password.
* No two users can have the same user ID.
* Users can login after registration using the user ID and password.
* The online courses will include three semesters (spring, summer, fall).
* There are limitations to the number of users that can enroll in a course.
* A waitlist will be available when a course is full.
* A user can cancel an enrollment.
* The system will select a user from the waitlist to fill the cancelled seat.



## User Classes and Characteristics

The user classes for the Student Enrollment System include students, faculty, administrators, and registrar’s Office. The student class needs the ability to register, login, view courses, search for courses, add and delete courses, and get waitlisted for a course. The faculty class needs access to manipulate the courses by adding and deleting a class. The administrator will need access to all the product functions to assist with login issues and assist with course issues using the student ID. The main class for the system is the student ID. The system will also need an ID for the courses with descriptions for each course and the ability to manipulate a course by adding or deleting enrollment.

## Operating Environment

The system will need to be accessible via a laptop, desktop, and any mobile device. Therefore, it will need to function across different platforms. In addition to being responsive to different devices it will need to function on all major web browsers (i.e., Chrome, Firefox, Microsoft Edge, Bing, etc..). The operating system (OS) utilization will include Mac and Windows.

## Design and Implementation Constraints

The user passwords for the accounts must be secure, therefore they need to be encrypted. The system will use one database to store the passwords XAMPP. Timing constraints for this system is eight months to develop and deploy. According to the U.S. Department of Education (n.d.) said the website must ensure student information is properly secured to adhere to the Faily Educational Rights and Privacy Act (FERPA). Additionally, the font and color on the website must be viewable for those with low vision. ADA (2007) suggests website developers avoid dictating the font color and size and instead allow the font and color set in the users’ web browsers and operating systems to make the determination.

## User Documentation

The website will include troubleshooting suggestions for class enrollment and course selection. Additionally, there will be a developing frequently asked questions (FAQ) document that will be maintained by the university for the website.

## Assumptions and Dependencies

The cost of the project can become overrun if the budget is not set and honored (Tsui et al., 2018). The budget for the Student Enrollment System will be $200,000. The development team will be available for eight months to create the website.

# External Interface Requirements

## User Interfaces

The layout of the website must include consistent headers, footers, navigation bars, and icons. The W3schools (n.d.) explained that a developer can use the Font Awesome icons by adding the following in the <head> section: <link rel ="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">.

The webpage will also include indicators that will guide them to the previous or next page while maintaining the session for which the user is currently logged. The color pattern will include properly contrasting colors for readability. The webpages will include a home button that is easily visible to the user to allow for easy navigation to the home page. Error message will clearly identify the issue and provide guidance about how to current the error.

## Hardware Interfaces

The devices that will be able to use the website include all desktops, laptops, tablets, smart phones, and any additional mobile devices. To encrypt the connections and hide user history, and other personal metadata the website will need to use Hypertext Transfer Protocol Secure (HTTPS) (Loshin & Loshin, 2003). The data shared on the website will be secure using HTTPS.

## Software Interfaces

The website will include a connection to the database that houses all user access information. Additionally, the website will connect to the most popular operating systems (Windows, Mac, Linux). The website will also allow users to access their school email using the Simple Mail Transfer Protocol (SMTP).

## Communications Interfaces

The users will have access to communications from the website using their school email address. For example, when a user withdraws from a course, or a notification is received by the user with a seat opens up in a class where the user is currently on a waiting list.

# System Features

## Account Creation

* Priority: High
* Description: The user will create an account by clicking the registration button which includes part of the website functionality. Access to the website is not possible without registration.
* Benefit: 9

**Stimulus/Response Sequences**

* The home page will prompt users to log in or register.
* User selects register if a first-time user. User selects login if registration is complete.
* User is a first-time user, complete the registration form.
* User will submit the form.
* Users will receive confirmation of account set-up or an error message indicating what need to be corrected.
* User selects log-in if they are returning.
* User enters credentials (user ID and password).
* User is directed to the home page.

**Functional Requirements**

* REQ-1: The website presents two options on the landing page, register and login.
* REQ-2: The website displays the registration form when user selects “register.”
* REQ-3: The information from the form is confirmed, sanitized, treated before it is sent to the database once the submit button is selected.
* REG-4: Error messages will appear if a field is empty or if inputs for user ID have been used already.
* REG-5: If no errors or duplications of user ID, a successful registration message will display.

## User Login

* Priority: High
* Description: Users who have registered can login using the user ID and password created during registration.
* Benefit: 9

**Stimulus/Response Sequences**

* The landing page prompts the user to login or register.
* User selects the login button.
* Enter login credentials (user ID and password).
* The submit button is selected.
* The system displays a message confirming a successful login or an error message indicating credential errors.

**Functional Requirements**

* REQ-1: The landing page will show a login and register button.
* REQ-2: Selecting the login button will present the user ID and password.
* REQ-3: The data is verified, sanitized, processed, and compared to the data in the

database.

* REQ-4: The website will display a success message upon confirmation of the user

credentials.

## Course Searchability

* Priority: High
* Description: The search feature will allow users to view courses.
* Benefit: 8

**Stimulus/Response Sequences**

* The option titled “course registration”
* The option to select a course appears in the dropdown.

**Functional Requirements**

* REQ-1: The option titled “course registration” link
* REQ-2: The selected course will display available courses.

## Course Registration

* Priority: High
* Description: The registration option will be viewable.
* Benefit: 9

**Stimulus/Response Sequences**

* Select the profile link to view the course schedule.
* If the course is full, the user will be placed on the waitlist.

**Functional Requirements**

* REQ-1: The course registration link will appear on the navigation bar.

## Course Withdrawal

* Priority: High
* Description: The withdrawal option allows users to cancel registration, this is part of the website’s basic functionality. The ability to manage a schedule ensures a user can withdraw from a course if needed.
* Benefit: 9

**Stimulus/Response Sequences**

* Select course registration from the navigation bar.
* Schedule appears.
* User selects a course to drop
* select the submit option.
* Redirect to profile page

**Functional Requirements**

* REQ-1: The course scheduling screen is accessible from the navigation bar.
* REQ-2: The course scheduling screen will appear after selecting the link.
* REQ-3: Submit button will appear after selecting a course.

## Waitlisted

* Priority: Medium
* Description: Notification of course availability displays.
* Benefit: 7

**Stimulus/Response Sequences**

* Account login
* Message displays indicating waitlisted.

**Functional Requirements**

* REQ-1: User login
* REQ-2: Notification appears at the top of the page.

# Other Nonfunctional Requirements

## Performance Requirements

Account registration must take less than 30 seconds. The course searchability should take no more than two seconds to produce results. Submitting or withdrawing from a course must occur in less than three seconds. The website must perform properly and within the requirements. If not, it will result in enrollment delays in the registrar’s office.

## Safety Requirements

Personal information must be protected by encrypting the user password. The system must follow FERPA and ADA guidelines. Finally, a session must disconnect within five minutes if there is no activity.

## Security Requirements

Login credentials must be verified before a user can gain access to any course registration information.

## Software Quality Attributes

In addition to the Student Enrollment System meeting ADA requirements, it must also have simplistic navigation and usability. The reliability of the website will be demonstrated by ensuring no interruptions occur beyond the requirements of performance. Site development should be built so that enhancements and add-ons can occur in the future. Additionally, the university will maintain the system, which must be testable.

## Business Rules

Users who have not registered cannot have access to the website to add or withdraw from courses. The administrators must be able to reset user credentials if they get locked out of their account. Also, the administrators must have the ability to pull reports about students using the user ID that shows enrollment and withdrawal information for each course.

# Testing Levels and UML Models

## Introduction

The software development life cycle (SLDC) testing phase is imperative to producing a quality product for the stakeholders. Testing, when conducted properly and effectively, can have many benefits. Testing can help reduce costs post-development, help with maintainability, discover defects and errors earlier, and present a product to the stakeholders that meet their requirements. Several kinds of testing are seen in the SDLC to ensure the system performs as expected. This paper will discuss the details associated with the levels of testing, such as component, integration, system, and acceptance testing.

**6.2 Component Testing**

Component testing can take place throughout the SDLC. Unit testing is another name for components testing, which means the tester takes each part of the system and tests it independently from the other components of the system. This testing phase help save time in finding bugs at the early development stages. The goal of component testing is to check the input and output performance of the unit. It is vital to test all classes and functions. The phases of component testing for a student enrollment system can be determined using a use case diagram, see Figure 1.

**Figure 1** Use Case Diagram

A diagram of a diagram

Description automatically generated

Note: [This is a Use Case Diagram created with Draw.io. Click here for the details](https://drive.google.com/file/d/1M4E7vy067e5XkGEIQ64WOpWZRga3MkFL/view?usp=sharing)

**6.3 Integration Testing**

Ali et al. (2018) state that integration testing will locate bugs and errors in the interaction between integrated components. Once the initial testing phase is completed, component testing, the next step is to test the units in collaboration, which is integration testing. For example, Figure 2 illustrates how the units will function together. In this case, the sequential events that should occur in the Student Enrollment System are shown in the Class Diagram. While the units can pass the component testing, the control and data flow may have issues that can be checked during integration.

**Figure 2** Class Diagram

A screenshot of a computer program

Description automatically generated

Note: [This is a Class Diagram created with Draw.io. Click here for the details](https://drive.google.com/file/d/1M4E7vy067e5XkGEIQ64WOpWZRga3MkFL/view?usp=sharing)

**6.4 System Testing**

The next step in the sequence of testing is system testing. System testing confirms that the system is doing what the stakeholders expect, such as usability, performance, security, and compatibility (Suffian et al., 2016). For example, Figure 3 shows an activity diagram highlighting actions that should occur during each requirement. The test is designed to find defects in the integrated units and the entire system (Suffian et al., 2016). The entire system is tested to see if it is ready to be tested by the end-user or stakeholders; this is also an excellent time to observe the system’s behavior, constraints imposed by conditions, and sequential and concurrent activities. Additionally, Figure 4 will clearly show the sequential steps of the system.

**Figure 3** Student Enrollment System Activity Diagram

A diagram of a program

Description automatically generated

Note: [The image above is a UML image of the Activity Diagram for a Student Enrollment System](https://drive.google.com/file/d/1M4E7vy067e5XkGEIQ64WOpWZRga3MkFL/view?usp=sharing)

**Figure 4** Sequence Diagram

A diagram with text and words

Description automatically generated with medium confidence

Note: [The image above is a UML image of the Sequence Diagram for a Student Enrollment System](https://drive.google.com/file/d/1M4E7vy067e5XkGEIQ64WOpWZRga3MkFL/view?usp=sharing)

**6.5** Acceptance Testing

The final testing phase is acceptance testing. This stage of testing seeks the approval and feedback of the end users. This is an excellent opportunity for all actors on the system to test the functionality, such as the Registrar’s Office, Administrators, and students. Using different devices and operating systems is vital during this phase. Figure 3 illustrates the activities the students must test to ensure all functionality performs as expected.

**Figure 3** Student Enrollment System Activity Diagram

A diagram of a program

Description automatically generated

Note: [The image above is a UML image of the Activity Diagram for a Student Enrollment System](https://drive.google.com/file/d/1M4E7vy067e5XkGEIQ64WOpWZRga3MkFL/view?usp=sharing)

# Landing, Login, Enrollment Pages

Landing Page

**A screenshot of a computer

Description automatically generated**

Login

**A screenshot of a login page

Description automatically generated**

Enrollment

**A screenshot of a computer registration page

Description automatically generated**

Admin Logged-In Student Logged-In

**A screen shot of a computer

Description automatically generated**  **A screen shot of a computer

Description automatically generated**

**Admin Adds Courses Student Selects Classes**

**A screenshot of a computer

Description automatically generated**  **A screenshot of a computer

Description automatically generated**

**Student Profile of Enrolled Courses Waitlist Display**

**A screenshot of a profile page

Description automatically generated**  **A screen shot of a computer

Description automatically generated**

**Dropping a Class Profile Showing Dropped Course**

**A screenshot of a computer

Description automatically generated A screenshot of a profile page

Description automatically generated**

# MySQL Database and Class Registration

Tables

A screenshot of a computer

Description automatically generated

User Table Users

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

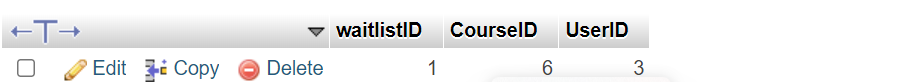
Course Table Courses Added

**A screenshot of a computer

Description automatically generated** **A screenshot of a computer

Description automatically generated**

Database Showing Waitlist Data Showing Course Enrollment

**A screenshot of a computer

Description automatically generated**

# PHP Code

Index

A screenshot of a computer program

Description automatically generated

Landing Page

**A screenshot of a computer

Description automatically generatedA computer screen shot of a computer code

Description automatically generated**

Source Code for Login Page

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

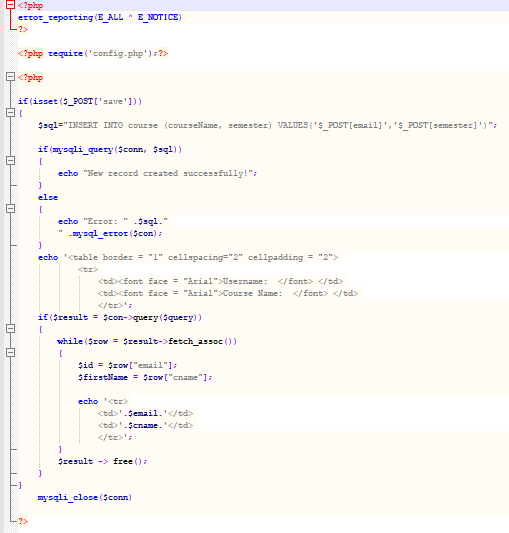
Source Code for Registration Page (pt1) Source Code for Registration Page (pt.2)

A screenshot of a computer

Description automatically generated A screenshot of a computer program

Description automatically generated

**Course Enroller Course Registration (Pt.1)**

****  **A screenshot of a computer program

Description automatically generated**

**Course Registration (Pt. 2) Course Management (Pt.1)**

**A screenshot of a computer program

Description automatically generated A screenshot of a computer

Description automatically generated**

**Course Management (Pt. 2) Course Management (Pt. 3) Course Management (Pt. 4)**

**A screenshot of a computer program

Description automatically generated**  **A screenshot of a computer program

Description automatically generated A white screen with black text

Description automatically generated**

**Course Management (Pt. 5)**

****

**Master**

**A screenshot of a computer

Description automatically generated**

SQL Code for Tables

Text

Description automatically generated with low confidence

# References

ADA. (2007). Website Accessibility Under Title II of the ADA. <https://www.ada.gov/pcatoolkit/chap5toolkit.htm>.

Ali, S., Imran, M., Hafeez, Y., Abbasi, T. R., Haider, W., & Salam, A. (2018). Improving Component-Based Software Integration Testing Using Data Mining Technique. 2018 12th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), Mathematics, Actuarial Science, Computer Science and Statistics (MACS), 2018 12th International Conference On, 1–6. https://doi-org.proxylibrary.ashford.edu/10.1109/MACS.2018.8628368

Loshin, P., & Loshin, P. (2003). Tcp/ip clearly explained. Elsevier Science & Technology.

Suffian, M. D., Rizal, F., Loo, F. A., Aman, N. F., & Bajuri, N. (2016). Software capability rating using system testing scores. 2016 IEEE Conference on Open Systems (ICOS), Open Systems (ICOS), 2016 IEEE Conference On, 105–110. https://doiorg.proxylibrary.ashford.edu/10.1109/ICOS.2016.7881997

Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.). Jones & Bartlett Learning.