Software Requirements Specification

for

Class Registration System

Version 1.0 approved

Prepared by Angelica Topete

University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Joseph Rangitsch

July 24, 2023

Table of Contents

Table of Contents ii

Revision History iii

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 1

2.1 Product Perspective 1

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

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 5

3.1 User Interfaces 5

3.2 Hardware Interfaces 6

3.3 Software Interfaces 6

3.4 Communications Interfaces 7

4. System Features 7

4.1 User Registration 7

4.2 Profile Management 8

4.3 Course Listing 8

4.4 Enrollment Process 8

4.5 Enrollment Deletion 9

4.6 Wait List Management 9

4.7 User Support 10

4.8 Administrative Tools 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

6. Test Plan 11

6.1 Introduction 11

6.2 Scope 12

6.3 Quality Objective 12

6.4 Roles and Responsibility 12

7. Test Methodology 13

7.1 Overview 13

7.2 Test Levels 13

7.3 Bug Triage 13

7.4 Suspension Criteria and Resumption Requirements 14

7.5 Testing Completeness 14

8. Test Deliverables 15

8.1 Test Processes 15

8.2 Test Levels 13

9. Resource and Envoirnment Needs 19

9.1 Testing Tools 19

9.2 Test Enviornment 20

9.3 Test Reporting Requirements 20

Appendix A: Glossary 21

Appendix B: Analysis Models 23

Revision History

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

# Introduction

## Purpose

The purpose of this document is to present the proposed platform for a course registration system, with the goal of simplifying the process for students. This document offers a comprehensive overview of the software system's architecture and the design of its individual components. Ultimately, this document aims to provide readers with a clear understanding of the system's inner workings, enabling them to design, maintain or modify the class registration system effectively.

## Document Conventions

This document is organized using a consistent and clear section heading approach. Conventional formatting is used for headings, subheadings, text, tables, and diagrams. Bullet points are used for clarity when presenting requirements and features. Diagrams and visuals have been incorporated to represent system architecture, data flow, use case scenarios, and user interface designs.

## Intended Audience and Reading Suggestions

This document is targeted toward individuals closely involved in the course registration project, including the development team, system analysts, clients, and project stakeholders. This project is a prototype for the course registration system and is limited to use within the organization's premises. It provides value to the systems team and its users.

## Product Scope

The proposed course registration system aims to provide an online platform for students to conveniently register and modify their class enrollments. It enables users to create accounts, modify profile information, and access online course offerings anytime. The system's primary goal is to serve as an alternative to the traditional, in-person class enrollment process. By implementing the system, students can benefit from simplified registration and profile management, improved accessibility to course options, and enhanced efficiency. In addition, the organization benefits from implementing an automated system that improves employee workload.

# Overall Description

## Product Perspective

The class registration system is a new self-contained product that includes user registration, user profile, login functionality, semester-based course offerings, course enrollment limits, waiting list, and enrollment cancellations. Its major features include a database, user interface, and external interface. These features collectively ensure a smooth and efficient class registration process for students.

A diagram of components for the class enrollment system

Description automatically generated

Figure 1.1 A diagram of components for the class enrollment system**.**

## Product Functions

A diagram of a course

Description automatically generated with medium confidence

Figure 1.2 A diagram of the classes and attributes of the system**.**

## User Classes and Characteristics

Users should be able to:

* Create an account
* Log in to the system
* View/update profile info
* Enroll in courses
* Cancel course enrollments
* Add themselves to the waiting list if a course is full.

System administrators should be able to:

* Manage user accounts
* Manage course information
* Set maximum enrollment limits for courses
* Process waiting lists
* Handle system configuration/maintenance.

Faculty should be able to:

* Create/manage courses
* Set course schedules
* View course enrollment details

## Operating Environment

The class enrollment system should be able to run on standard platforms like desktop computers, laptops, and servers. The software should also be compatible with mobile devices such as smartphones and tablets.

Operating System

* The software should be designed to be cross-platform compatible so that the system can be accessed on common operating systems such as Windows, macOS, Linux, etc.

Web Browsers

* The software should be compatible with popular web browsers that students, faculty, and administrators commonly use, such as Google, Chrome, Mozilla Firefox, Microsoft Edge, Safari, etc.

Database Management System (DBMS)

* The software should be designed to work with MySQL 8.0.

Web Server

* The software should be hosted on a web server software such as Apache HTTP Server 2.4.

Integration with Student Information System (SIS)

* The software should integrate with the institution's existing Student Information System to exchange courses, students, and enrollment data.

Email Service:

* The software should interact with SMTP to send automated notifications and communications to users.

Security Software

* Software should be compatible with antivirus and firewall software installed on the operating system to ensure system security.

## Design and Implementation Constraints

Time Constraint

* The institution must implement a new class registration system before the upcoming year.
  + The project team has a limited time frame to design, develop, and deploy the system, ensuring it is fully functional and tested before the registration period begins.

Resource Constraints

* The institution has allocated a specific budget for developing its class registration system.
  + The project needs to work with the assigned budget to obtain necessary hardware and software licenses and hire additional development resources as needed.

Security and Privacy Constraints

* The institution must comply with data protection regulations when implementing the system.
  + The team must ensure that student personal information, such as names, addresses, and social security numbers, is securely stored and transmitted using encryption protocols to protect against unauthorized access or data breaches.

Integration Constraint

* The institution plans to integrate the class registration system with the existing Student Information System (SIS) to streamline enrollment.
  + The team must ensure a seamless interaction between the two systems that enable real-time exchange of student information, courses, and enrollment data.

## User Documentation

The user documentation components (included in web application) that will be delivered along with the software include the following,

Online Support

* Users can access relevant help topics from the system, assisting specific features, options, and procedures.

FAQ (Frequently Asked Questions) Forum

* The FAQ helps users quickly find solutions to their queries and address potential concerns or doubts while using the system.

## Assumptions and Dependencies

Availability and Reliability of Internet Connectivity

* This project assumes that the users have a reliable internet connection to access the online class registration system.
  + Any issues with internet connectivity, such as outages or slow connections, can affect the usability and functionality of the system.

Integration with Student Information System (SIS)

* This system depends on integration with the institutions existing SIS. This dependency assumes the availability of interfaces and APIs provided by the SIS.
  + Any changes or limitations in the SIS interface, version, or compatibility can impact the ability of the system to communicate overall, negatively affecting the class registration system.

# External Interface Requirements

## User Interfaces

User Interface Components

* Account Creation and Login
  + This interface allows new users to create an account by providing necessary information such as username, password, and profile details.
    - Existing users can log in using their credentials to access the system
* Course Search and Selection
  + Users can search for available courses based on semester, department, or keyword criteria.
    - Users can view course details (description, instructor info, etc.)
* Enrollment and Waiting List
  + This interface allows users to enroll in courses if available.
    - Users can add themselves to the waiting list if a course is full.
    - Users can also cancel their enrollment if needed
* Profile Management
  + Users can view their personal information, such as contact details, email, addresses, etc.
    - Users cannot edit their names, socials, or birthday for security purposes.

Graphical User Interface (GUI) Standards

* Screen Layout
  + The GUI should follow a consistent layout across each page, ensuring easy navigation and excellent user experience.
    - Design principles such as proper alignment, spacing, and visual hierarchy should be applied.
* Standards Button and Functions
  + Common buttons and functions such as "Submit," "Cancel," "Save," and "Help" should be placed and labeled across the interface for familiarity and ease of use.
* Error Message Display
  + A standard error message display should be implemented to guide users when errors or validation issues occur.

## Hardware Interfaces

Supported Device Types

* Desktop computers
  + Compatible with desktop computers running common operating systems (such as Windows, macOS, or Linux).
* Laptop and Tablets
  + Ensures a responsive and user-friendly interface across different screen sizes.
* Mobile Devices
  + Consideration should be given to supporting mobile devices.

Data and Control Interactions

* Input devices
  + It should support standard input devices such as keyboards, mice, touchscreens, and stylus pens for user interaction.
* Output Devices
  + It should be capable of displaying information and interfaces through monitors, screens, or other visual output devices.
* Data Exchange
  + Software must interact with hardware components to exchange data (user input, system responses, and database updates).
* User Control
  + The user interface should enable users to control the system through input devices, triggering actions such as button clicks, text entry, and option selection.

Communication Protocols

* Network Connectivity
  + The software should use standard communication protocols such as HTTPS to facilitate communication between the client-side interface and the server-side components.

Physical Hardware Requirements

* Storage
  + The system may require network storage to store user data, configurations, and other relevant data.
* Networking Capabilities
  + The hardware should have network connectivity such as ethernet or Wi-Fi to enable communication between the software and external databases.

## Software Interfaces

The connection between the system and the software components includes the database, the operating system, tools/libraries, and services/communication.

Database

* Database management system
  + MySQL 8.0
* Data Items
  + Flowing in: Registration details, course info, enrollment requests, and profile updates.
  + Flowing out: Enrollment confirmations, waitlist notifications, and user information.

Operating System

* Ensures compatibility with multiple operating systems
  + Windows, macOS, Linux.

Tool and Libraries

* Web development framework
  + Django
* Front-end Technology
  + HTML, CSS, and Javascript

Integrated Commerical Components

* Email Service Provider
  + SMTP server

## Communications Interfaces

Email Communication

* The system should support standard email communication to support notifications, account verification emails, enrollment confirmation, and other relevant information to users.
  + - Communication Standard: Simple Mail Transfer Protocol (SMTP)

Web Browser Communication

* The system should be designed and developed to work with major web browsers.
  + - Communication Standard: Hyper Transfer Protocol (HTTP)

Network Server Communication

* The system should communicate with network servers to handle requests, process, and retrieve data.
  + - Communication Protocol: Transmission Control Protocol/Internet Protocol).

Communication Security and Encryption

* The system should implement security measures to protect communication channels and sensitive data.
  + - Secure protocols: HTTPS
    - Encryption Standards: Strong encryption algorithms and standards (AES).

# System Features

## User Registration

4.1.1 Description and Priority

* Users can create new accounts to access the class registration system.
  + - Priority: High

4.1.2 Stimulus/Response Sequences

* Stimulus: The user selects the "Register" option
* Response: The system prompts users to enter their information and create a new account.

4.1.3 Functional Requirements

REQ-1: Users must provide a unique email address during registration.

REQ-2: The system should validate the email address format to ensure correctness.

REQ-3: The system should generate a unique user ID and Password for the account.

REQ-4: User information should be securely stored and protected.

## Profile Management

4.2.1 Description and Priority

* Users can manage their personal information and preferences.
  + - Priority: Medium

4.2.2 Stimulus/Response Sequences

* Stimulus: The user selects the "Profile" option
* Response: The system displays the user's profile information for viewing or editing.

4.2.3 Functional Requirements

REQ-1: Users can update their contact details, such as phone number, email, and address.

REQ-2: The system should enforce data validation rules to ensure accurate and complete information.

REQ-3: Users can receive notifications via email or other communication channels.

REQ-4: The system should provide an option to reset the user's password if needed.

## Course Listing

4.3.1 Description and Priority

* Users can browse and view available courses for enrollment.
  + - Priority: High

4.3.2 Stimulus/Response Sequences

* Stimulus: The user selects the "Course Listing" option
* Response: The system displays a list of available courses for the selected semester.

4.3.3 Functional Requirements

REQ-1: The system should retrieve and display course information, including course, names, descriptions, and other details.

REQ-2: Users can search and filter courses based on criteria such as department, course code, or instructor name.

REQ-3: The system should provide a scrolling functionality for easy navigation through the course list.

## Enrollment Process

4.4.1 Description and Priority

* Users can enroll in courses of their choice
  + - Priority: High

4.4.2 Stimulus/Response Sequences

* Stimulus: The user selects a course and chooses the "Enroll" option
* Response: The system processes the enrollment request and confirms the enrollment.

4.4.3 Functional Requirements

REQ-1: The system should check the course availability and verify that the user meets the criteria before allowing enrollment.

REQ-2: Users should receive immediate feedback on enrollment status (successful or unsuccessful).

REQ-3: The system should update the user's course schedule and account for the enrolled course.

REQ-4: The system should handle and prevent a user's double enrollment in the same course.

## Enrollment Deletion

4.5.1 Description and Priority

* Users can delete their course enrollment after initial registration.
  + - Priority: Medium

4.5.2 Stimulus/Response Sequences

* Stimulus: The user selects the "delete " option
* Response: The system allows users to register or delete courses from their enrollment.

4.5.3 Functional Requirements

REQ-1: Users can view their current course schedule and delete courses.

REQ-2: The system should check the availability of the requested course and allow modification if possible.

REQ-3: The system should update the user's course schedule and reflect the changes in real-time.

REQ-4: Users should receive confirmation of successful modifications.

## Waiting List Management

4.6.1 Description and Priority

* Users can add themselves to a waiting list if a course is full.
  + - Priority: Medium

4.6.2 Stimulus/Response Sequences

* Stimulus: The user selects the "Enroll in waiting list" option or a full course.
* Response: The system adds users to the waiting list and provides relevant information.

4.6.3 Functional Requirements

REQ-1: The system should track and maintain a waiting list for each course.

REQ-2: Users on the waiting list should be notified if a spot becomes available.

REQ-3: The system should automatically enroll the first user on the waiting list when a spot opens.

REQ-4: Users should have the option to remove themselves from the waiting list.

## User Support

4.7.1 Description and Priority

* Users can access support resources and seek assistance when needed.
  + - Priority: Low

4.7.2 Stimulus/Response Sequences

* Stimulus: The user selects the "Help/Support" link in the notification bar
* Response: The system provides access to user documentation, including FAQ's and online support.

4.7.3 Functional Requirements

REQ-1: The system should include user documentation, such as tutorials, or online help, to assist users in navigating and using the system.

REQ-2: Users should be able to access a FAQ section that addresses common issues.

REQ-3: The system should provide a support contact for users to seek assistance for specific issues or inquiries.

## Administrative Tools

4.5.1 Description and Priority

* Authorized administrators can access tools to manage system settings and user accounts.
  + - Priority: High

4.5.2 Stimulus/Response Sequences

* Stimulus: The administrator logs in and accesses the administrative panel.
* Response: The system provides access to administrative tools and functionalities.

4.5.3 Functional Requirements

REQ-1: Administrators should have secure login credentials and appropriate user roles and permissions.

REQ-2: The system should allow administrators to manage system configurations, such as semester dates, course offerings, and enrollment limits.

REQ-3: Administrators should be able to view and manage user accounts, including account activation, password resets, and account deletion.

REQ-4: The system should provide reporting capabilities to administrators, such as statistics and user activity logs.

# Other Nonfunctional Requirements

## Performance Requirements

The system should be able to handle a large number of concurrent users during peak registration periods with an average response time of under 3 seconds.

## Security Requirements

User passwords should be stored using strong encryption algorithms to prevent unauthorized access to user accounts. The system should comply with industry standards security protocols, such as HTTPS, to ensure secure communication between the system and users.

## Software Quality Attributes

Usability

* Ease of use
* Intuitive user interfaces
* User-friendly (clear navigation, simple registration process, etc.)

Reliability

* Availability
* Stability
* Accurate data handling

Scalability

* Handle a large number of users

Security

* Protect user data
* Ensures Confidentiality
* Prevents Unauthorized Access
  + Strong security measures

Interoperability

* Exchanges data with external systems.

Testability

* Testable units
  + Measured by ease and efficiency of test case creation, execution, and coverage.

## Business Rules

Enrollment Limit Rule

* Each course has a maximum enrollment limit.
  + The system should enforce this rule to prevent over-enrollment and maintain an optimal student-to-teacher ratio.

Waitlist Rule

* If a course reaches its enrollment limit and a student attempts to register, the system should automatically add the student to a waitlist.
  + When a seat opens due to a cancellation or increase in enrollment capacity, the system should inform the first student on the waitlist, allowing them to enroll in the course.

# Testing Plan

## Introduction

The testing process for the course registration system project aims to ensure the software’s reliability, functionality, usability performance, and security (Spillner et al., 2014). It will follow a comprehensive apporach that includes both manual and automated testing methodologies. The testing process will follow an interative and agile apporach, aligned with the projects development methodology. The primary objectives of testing include verifying that the application adheres to the specified requriments, meets quality specifications defined by the client, and identifies/addresses any bugs or issues before the system is deployed.

## Scope

### *In Scope*

In scope for the testing process are:

* New user registration that includes account and profile creation.
* Unique ID and password creation for each user during registration.
* Guarding against two users using the same ID for registration.
* Profile information for each user, including name, phone, email, and other necessary information.
* User login using the created ID and password.
* Online courses offerings in three semesters: Spring, Summer, and Fall.
* Course enrollment with a max number of enrollments for each course.
* Waiting list functionality for full courses.
* Course cancellation and notifying the first user on the waiting list.

### *Out of Scope*

* Detailed testing of specific hardware or infrastructure components not directly related to the system.

Testing of unrelated third-party software or services not essential for the functioning of the course registration system

## Quality Objective

The primary quality objectives of the testing proess are:

* Ensure the course registration system meets all specified functional and non-functional requirements outlined in the SRS document.
* Validate that the application performs reliably and meets the quality specifications defined by the client
* Identify and report any defects or issues to the development team for a timely resolution.

## Roles and Responsibilities

The following team members have specific roles and responsibilities in the testing process:

* QA Analyst: Responsible for designing test cases, executing test, and documenting results.
* Test Manager: Oversees the testing process, coordinates resources, and ensures the objectives are met.
* Configuration Manager: Manages the test environment and configurations for testing.
* Automated Test Engineer: Responsible for designing the test procedures that utilize test tools.
* Developers: Collaborate with the testing team to address and resolve reported defects.
* Installation Team: Responsible for deploying the application in the testing environment.
* End user: Responsible for deploying the application in real life environment.

# Test Methodology

## Overview

**Test Methodology:** V Model

* The V Model aligns development phases with corresponding testing phases, emphasizing early testing involvement.
* Testing and development phases, including component test, integration test, system test, and acceptance test, will be conducted in parallel.
* Black box testing and white box testing will be employed to test different aspects of the e-commerce website.
* The approach aims to catch defects early, reducing the cost of fixing issues in later stages (Spillner et al., 2014).
* The V Model's systematic approach will contribute to the overall quality and reliability of the software product (Spillner et al., 2014).

## Test Levels

* 1st Task: *Component testing*Verifies if the component design fulfills the specification.
* 2nd Task: *Integration testing*Verifies whether the structural units and subsystems interact properly (GeeksforGeeks,2022).
* 3rd Task: *System Testing*

Checks if the integrated software meets the specified requirements "from the perspective of the customer and future user" (Spillner et al., 2014, P.58).

* 4th Task: *Acceptance Testing*Verifies whether the system meets the user's requirements and see if the system is ready for real-world use (GeeksforGeeks, 2022).

## Bug Triage

The goal of the triage is to

* To define the type of resolution for each bug
* To prioritize bugs and determine a schedule for all “To Be Fixed Bugs’.

### Integrity Level Schema

The functionalities of the course registration system are identified using three levels:

**High risk:** Errors in these functionalities have an elevated possibility of causing problems.

* Site Access
* Profile Registration Process
* Login Process
* Database Update

**Medium risk:** Errors in these functions have a standard possibility of causing problems.

* Navigation
* Automatic Redirection

**Low risk:** Errors in these functions have a low possibility of causing problems.

Inconsistency in page design

## Suspense Criteria and Resumption Requirements

**Suspension criteria**: used to suspend all or part of the testing procedure.

* Critical system outage
* Blocking effect
* Resource Constraint
* Change in requirement.

**Resumption criteria**: determines when testing can resume after it has been suspended.

* Defect resolution
* Availability of resources
* System stability
* Review of changes

## Test Completeness

The testing process will be considered complete when the following criteria are met.

**Projects Defined Exit Criteria:**

* Test case percentage  
  ▪ 100% of tests must be executed  
  ▪ Pass percentage should be at least 90%
* Test budget  
  ▪ The test budget must be depleted.
* Code Coverage  
  ▪ 95% of code coverage will be performed  
  ▪ 95% of the code will be errorless (no critical defects)
* Closed/fixed defects  
  ▪ All report defects will be fixed or closed  
  ▪ Test manager must verify all closure reports.
* Project Risk  
  ▪ Testing in high-risk areas must be complete.

# Test Deliverables

## Test Processes

***Activity: Concept***

**Test task:**  
Perform research of other another’s schools online course enrollment systems to see what features are most vital in modern registration systems and determine what would make the system stand out. •

*Methods:*

* Comparison testing
* Outcome-Driven Innovation
* Customer-Centric Strategy

Inputs:

* Define other participants.
* Discover the user’s needs.
* Map the customer journey.

Outputs:

* Detection of essential functionalities that the system will require.
* Reveal different layouts and styles that appeal most to users.
* Uncover what is the target users’ needs.
* Log test

Resources:

* Desktop Computer
* Network access

Risks:

* Different testers may have a different solution to this task, so the results may vary.
* Risks of not completing the concept activity include not understanding features of the system that may be vital and not understanding the clients needs.

Roles and responsibilities:

• Development team

o Study popular student course registration websites on the internet.

o Outline the most critical features.

***Activity:* Requirements**

**Test task:**

* Identify the projects requirements.
* Validate that they are correct and complete.

Methods:

* Static Testing
* Inspection

Inputs:

* Planning- Selecting a review team and meeting schedule.
* Preparation- Moderator collects all the information for the inspection meeting.
* Meeting-The moderator conducts the review meeting.

Outputs:

* The review team points out and records any identified flaws.
* The author modifies the requirements/specification.
* Log test

Roles and Responsibilities:

* Moderator: Has the responsibility to carry out the inspection.
* Review Team: 3 to 4 people from the project's development team.
* Author: The author of the SRS document must modify the document as needed.

***Activity: Design***

**Test task:**  
• Review/validate the UML design models for the course enrollment system.

Methods:  
• Experience based testing techniques

Inputs/Outputs:

• Use case

* Validate that each role has access to the correct functionality.
* Deployment
  + Validate that the systems devices interact properly.
  + Ensure that no elements are missing.
* Sequence
  + Validate that the systems behaviors interact appropriately.
  + Ensure that no functionalities are missing.
* Activity Diagram
  + Validate that the systems behaviors interact appropriately.
  + Ensure that no functionalities are missing.

Roles and Responsibilities:

• Development team should conduct the review of the systems design

***Activity: Implementation***

**Test task**:  
• Verify the systems code.

Methods:  
• Static Testing Techniques

• Complier

• Data flow Analysis

* White box testing
* Error Guessing

Inputs:

* Statement coverage
* Branch Coverage
* Condition Coverage
* Path Coverage

Outputs

* Risk analysis
* 95% code coverage
* Log test

Roles and Responsibilities:

* Developer
  + Test each statement.

**3.1.2 Process**: Operation

***Activity: Interface***

**Test task**:  
• Ensure that each functionality of the websites interface interact properly.

Methods:

* Manual Testing
* Black box testing

Inputs:

* Access web application
* Access home page
* Click on the buttons in the navigational bar.

Outputs:

* Website should be accessible using the web browser.
* Login/registration should allow the user to input text.
* Website should direct the user to the requested pages.
* Each page must contain relevant information.
* Log test

Resources:

* Desktop Computer (with real-world operating system)
* Mobile phone
* Network access

Roles and responsibilities:

• End-user/client

* + - Test each functionality in the interface.

***Activity: User Registration***

**Test task:**

* Ensure the system reacts appropriately when registering a user.

Methods:

* Black box testing
* Equivalence class partitioning
* Boundary Value Analysis
* CAST Tools

Inputs

* The tool inputs a user's required information (name, email, password etc.)
* User already registered (check user info prior to registration).
* Error in password length

Outputs

* Website should register the user.
* Once registered the user should be directed to their profile page.
* Error message stating the user is already registered should appear.
* System should state that the user passwords characteristics are invalid.
* Log test

Resources:

* Desktop Computer
* Automated tool (Tool Cast)
* Network access

Roles and responsibilities

* Development team  
  o Run the test using the automated testing tool.
* Test Automator:  
  o Ensure the CAST tool is integrated with the proper conditions for the test.

***Activity: Login***

**Test task:**  
• Ensure the system reacts appropriately when a user logs in

Methods:

* Black box testing

o Equivalence class partitioning

o Boundary Value Analysis

* + CAST Tools

Inputs:

* + The tool inputs a user's required information (email, password) into the designated area.
  + Password Error
  + Renter password. If correct, log the user in. If not repeat the output

Outputs:

* Website should log the user into their profile.
* System should state the password was incorrect.
* If password is incorrect after three tries two step verification is enabled
* Log Test

Resources:

* Desktop Computer
* Automated tool (ToolCast)
* Network access

Roles and responsibilities:

* Development team  
  o Run the test using the automated testing tool.
* Test Automator:  
  o Ensure the CAST tool is integrated with the proper conditions for the test.

***Activity: Course Enrollment/Join Waiting List***

**Test task:**  
• Ensure the system reacts appropriately when enrolling to a course and ensure the system reacts appropriately when enrolling into the waiting list.

Methods:

* Black box testing

o Equivalence class partitioning

o Boundary Value Analysis

* CAST Tools

Inputs:

* The tool inputs all the users required information for enrollment (name, email, etc.) and hits submit.
* (Wait List) The tool clicks Join Waitlist.

Outputs:

* Website should direct the user to a webpage that states the enrollment to this course was successful.
* (Wait List) The website should direct the user to a webpage that states the enrollment to the wait list was successful.
* Log Test

Resources:

* Desktop Computer
* Automated tool (Tool Cast)
* Network access

Roles and responsibilities:

* Development team

O Ensure the CAST tool is integrated with the proper conditions for the test

o Run the test using the automated testing tool

***Activity: Deleting Course Registration/Waitlist Registration***

**Test task:**  
• Ensure the system reacts appropriately when a user deletes registration to a course or waitlist.

Methods:

* Black box testing  
  o Equivalence class partitioning

o Boundary Value Analysis

* CAST Tools

Inputs:

* The tool (after login) accesses the schedule page and clicks delete on a registered source.

Outputs:

* The website should redirect the user to a page that states the deletion was complete.
* The database should reflect the change.
* Log Test

Resources:

* Desktop Computer
* Automated tool (Tool Cast)
* Network access

Roles and responsibilities:

* Development team

o Run the test using the automated testing tool.

* Test Automator:

o Ensure the CAST tool is integrated with the proper conditions for the test

For the operation process, multiple tests for each category should be conducted. The test should include error values to ensure the system reacts appropriately under these conditions.

# Resource and Envoirnment Needs

## Testing Tools

**Required Testing Tools:**

* **Requirements Tracking Tool:** Jira
  + - Jira is a widely used issue and project tracking tool that enables teams to manage project requirements, user stories, and test cases effectively. It facilitates collaboration between development and testing teams, ensuring seamless communication and progress monitoring (Atlassian, n.d.).
* **Bug Tracking Tool:** Bugzilla
  + - Bugzilla is an open-source bug tracking system that provides a centralized platform for logging and tracking bugs or defects found during testing. It helps testers and developers prioritize and resolve issues efficiently (Bugzilla, n.d.).
* **Automation Tools:** Selenium WebDriver
  + - Selenium WebDriver is a popular open-source automation tool for web application testing. It allows testers to create automated test scripts in various programming languages (e.g., Java, Python, C#) to perform functional testing and regression testing (SeleniumHQ, 2021.)

## Test Environment

The test environment for the course registration system consists of the minimum hardware and software requirements necessary to conduct comprehensive testing. Ensuring that the test environment closely mimics the production environment is crucial to obtaining accurate test results.

**Minimum Hardware Requirements:**

Processor: Intel Core i5 or equivalent

RAM: 8 GB

Storage: 256 GB SSD

Display: 1366x768 resolution

**Software Requirements:**

* Operating System: Windows 8 and above (64-bit)
* Office Suite: Microsoft Office 2013 and above (for document-related tests)
* Email Client: Microsoft Exchange or compatible (for email-related tests)

**Additional Software:**

* Web Browsers: Latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari (for cross-browser compatibility testing)
* Database Management System: MySQL (for database-related tests)
* Testing Frameworks: Selenium WebDriver, JUnit (for automated testing)
* Test Management Tool: Jira, TestRail, or similar (for test case management and bug tracking)
* Version Control: Git (for collaborative development and version control)

**9.3 Test Reporting Requirements**

**Incident Reporting Process**

* Incident Reporting Database
* Test logs are evaluated and compared to the expected results.
  + - If an error is present, the incident is logged.
* The tester creates an incident report, including the problem, the tested.

software, the test environment, the defect class, and prioritization

(Spillner et al., 2014).

* The status of the incident is continuously monitored using the model.

seen in figure 3.1.

* Only the test manager can close a report.
* All incidents must be addressed as exit criteria.

Figure 3.1 Incident reporting

A diagram of a process

Description automatically generated

Appendix A: Glossary

Student Information System (SIS)

* A software application used by educational institutions to manage student-related data and streamline administrative tasks (Edwards, 2022).

Class diagram

* A visual representation that depicts the structure and relationships of classes, interfaces, and their attributes and methods within a system or software application (GeeksforGeeks, 2017).

State Diagram

* A state diagram is a visual representation that illustrates the various states and transitions of an object or system throughout its lifecycle (GeeksforGeeks, 2017).

Availability

* Relates to system failure and its consequences (University of New Brunswick, n.d). This occurs when the system no longer works as expected and affects the system users and the institution.

Cast Tools

* “Software applications that assist in testing software programs”(Techreviewer, 2022, Sec. 2).

Black Box Testing

* Test method in which testers have no knowledge of the inner workings (code) of the software (Spillner et al., 2014).

White Box Testing

* A testing method in which developers evaluate the inner structure, design, coding, functionality of the software (Spiller et al., 2014).

Equivelence Class Partitioning (ECP)

* Divides test data into partitions.

Branch Coverage

* Tests ever outcome from statement or loops (Spillner et al., 2014).

Boundary Value Analysis

* Test the boundary values and invalid partitions (Spillner et al., 2014).

Statement Coverage

* Used to develop test cases. Requires running the source code to calculate the number of statements (Spillner et al., 2014).

Path Coverage

* Inspects all possible paths of execution (Spillner et al., 2014).

Error Guessing

* Testers guess the errors and faults in areas where formal testing will not work (Spillner et al., 2014).

Compiler

* Form of automated testing that identfies faults or defects in source code.

Data Flow Analysis

* A technique used to reveal defects by focusing on the path through the codes (Spillner et al., 2014).

Inspection

* Formal review stragety that follows a conventional process (Spillner et al., 2014).

Comparison Testing

* Comparing two or more models to the product.

Static Testing

* Testing a system without executing the actual code.

Appendix B: Analysis Models

Figure 1.1 Use Case Diagram

A diagram of a person's diagram

Description automatically generated

Note. The use case diagram for the course registration system serves as a visual representation of the system's various roles and interactions. This diagram provides a clear overview of the distinct roles involved and illustrates how these roles interact with the system.

**A diagram of a server

Description automatically generated**Figure 1.2 Deployment Diagram

Note. The deployment diagram for the course management system shows the system's architecture, illustrating the hardware and software components and their environment. This diagram visually represents how the website is deployed and demonstrates the connections and relationships between the elements.

**A diagram of a software application

Description automatically generated**Figure 1.3 Sequence Diagram

Note. The sequence diagram for the course registration system provides a visual representation of how objects interact within a specific use case. In this case, the deployment diagram models the flow of interaction between the system when a user accesses certain functions.

A diagram of a state diagram

Description automatically generatedFigure 1.4 State Diagram

Note. The state diagram for the course registration system is designed to illustrate the different states that the system can be in. In this particular diagram, we focus on the control flow within the system, specifically based on the user's input.

*A diagram of access granted

Description automatically generated*

Figure 1.5 (A) State Diagram for Password Verification

Note. The state diagram for the system's password verification feature illustrates a security feature that blacks account access after the 3rd consecutive failed password attempt. Access to the system is granted if the correct password is entered before the third attempt.

*A table with text on it

Description automatically generated*Figure 1.5 (B) State Transition Table

Note. The state transition table shows the transition states for the password verification system.

**A diagram of a model

Description automatically generated**Figure 1.6 Model Controller View Model

Note. The Model Controller View (MCV) model divides the application into three interconnected components the model, the view, and the controller.

A diagram of a course management system

Description automatically generatedFigure 1.7 Class Diagram

Note. The class diagram provides a comprehensive representation of the structure of the course registration system. It visualizes the different classes within the system, including their attributes, methods, and relationships.

**References**

Atlassian. (n.d.). Jira Software. <https://www.atlassian.com/software/jira>

Bugzilla. (n.d.). About Bugzilla. <https://www.bugzilla.org/about/>

Edwards, L. (2022, April 2). *Student Information Systems*. TechLearningMagazine. <https://www.techlearning.com/news/student-information-systems>

GeeksforGeeks. (2017). *Unified Modeling Language (UML) | State Diagrams - GeeksforGeeks*. GeeksforGeeks. <https://www.geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/>

GeeksforGeeks. (2022). *Software engineering: SDLC V-model.* GeeksforGeeks. https://www.geeksforgeeks.org/software-engineering-sdlc-v-model/

Spillner, A., Linz, T., & Schaefer, H. (2014). Software Testing Foundations: A Study Guide for the Certified Tester Exam. Rocky Nook.

SeleniumHQ. (2021). Selenium WebDriver. <https://www.selenium.dev/documentation/en/webdriver/>

Techreviewer. (2022). All You Need to Know About Computer Aided Software Testing (CAST). https://www.tutorialspoint.com/software\_testing\_dictionary/review.htm#:~:text =Importance%20of%20Review%20Process%3A%201%20Productivity%20of%20Dev

University of New Brunswick. (n.d). Understanding quality attributes. <https://www.cs.unb.ca/~wdu/cs6075w10/sa2.htm#:~:text=Quality%20Attribute%20Scenarios%20in%20Practice%201%20AVAILABILITY%20%C2%B7,...%207%20COMMUNICATING%20CONCEPTS%20USING%20GENERAL%20SCENARIOS%20>

‌