**Software Project – Course Enrollment System**

David Ashby

The University of Arizona Global Campus

CST 499: Capstone for Computer Software Technology

Instructor Amr Elchouemi

September 26, 2022

Table of Contents

Table of Contents 2

Revision History 3

Introduction 3

Purpose 3

Document Conventions 4

Intended Audience and Reading Suggestions 4

Product Scope 4

Overall Description 5

Product Perspective 5

Product Functions 5

User Classes and Characteristics 5

Operating Environment 6

Design and Implementation Constraints 6

User Documentation 6

Assumptions and Dependencies 6

External Interface Requirements 7

User Interfaces 7

Hardware Interfaces 7

Software Interfaces 7

Communications Interfaces 8

System Features 8

New User Registration 8

Login Feature for New and Returning Student 9

Search Feature 10

View Courses(s) Feature 11

Register for a Course Feature 12

Drop a Course Feature 13

Notification of Course Availability Feature 14

Other Nonfunctional Requirements 15

Performance Requirements 15

Safety Requirements 16

Security Requirements 16

Software Quality Attributes 16

Business Rules 16

Testing Levels and UML Models 17

Component Testing 17

Integration Testing 18

System Testing 18

Acceptance Testing 19

Class Diagram 20

Sequence Diagram 21

Activity Diagram 22

State Diagram 23

Use Case Diagram 24

Test Management 26

Test Teams 26

Tasks and Qualifications 26

Exit Criteria 27

Estimated Cost Effort 27

Incident Reporting 27

Defect Classification 28

**Configuration Management** 29

**Testing Tools** 29

**Appendix A: Glossary** 33

Appendix B: Analysis Models 33

Appendix C: To Be Determined List 33

GitHub 34

**References** 34

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| David Ashby | 8-29-2022 | Creation of document for Week 1. | 0.1 |
| David Ashby | 9-20-2022 | Added Test Plan | 0.2 |
| David Ashby | 9-26-2022 | Completed Final Paper and Project | 1.0 |

# Introduction

## Purpose

This software requirements specification (SRS) outlines the requirements for the Course Enrollment System. The purpose of the system is to allow students the ability register their profile, view online courses that are offered per semester, select courses that are available, be added to a waiting list, or cancel a course. The scope of the SRS document is to provide the context of the Course Enrollment System, along with the initial build, functionality based on requirements, including non-functional requirements.

## Document Conventions

This SRS is delivered in five sections: 1. Introduction, 2. Overall Description 3. External Interface Requirements 4. System Features 5. Other Nonfunctional Requirements. Within each section will be sub-headers to provide additional detail.

## Intended Audience and Reading Suggestions

This document is intended for, developers, project managers, marketing staff, users, testers, documentation writers, and other stakeholders involved with the project. To gain a better understanding of the Course Enrollment System, it is suggested that everyone should read the document in its entirety to understand the full scope of the project.

## Product Scope

The purpose of Course Enrollment System is to allow students the ability to mange their account and course selection for the given semester. This will include students the ability to create and manage their account and profile, they will be able to login into the system, search for courses that are offered during any semester, enroll in a course, or have the option to be put onto a waiting list if the course is full, and the option to drop a course they are enrolled in. The Course Enrollment System aligns with the scope of the business to create a user-friendly enrollment system, that will encourage students to no long come in person to enroll, increase student experience, and reduce cost and time from the current process.

# Overall Description

## Product Perspective

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

## Product Functions

Students should be able to log into the Course Enrollment System or register, search for courses, view available courses for any semester, see the maximum number of students allowed per course, student can add or drop courses, student will have the ability to add themselves to a waiting list, student will be notified when the class is available, and student can view or manage their courses.



## User Classes and Characteristics

User classes for the Course Enrollment System will be students, registration faculty, and administrators. The student class will be for the students that use the system will be able to register/login, view, add, or drop classes, be added to waiting list, and manager their profile information, including course selections; students will be the primary user class. The registration faculty class will be for adding, updating, or removing courses. The administrator class will be for helping students or registration faculty with account or course issues, including the ability to add, update, or delete customers, registration faculty staff, or courses.

## Operating Environment

The Course Enrollment System needs to have cross-platform functionality, including mobile or tablet support, this includes functioning on various operating systems such as Mac, Windows, iOS, or Android. It should operate on all major browsers which can include Chrome, Edge, Bing, Safari, and Internet Explorer (until IE is decommissioned).

## Design and Implementation Constraints

With the Course Enrollment System being a new design there needs to be a relational database and encryption for all user accounts. The organizations IT department will maintain the Course Enrollment System and utilize their programming standards, such as modifiability, maintainability, and testability.

## User Documentation

Since this is a new Course Enrollment System, Students will be offered a “Frequently Asked Questions” (FAQs) to troubleshoot or speak to an agent by either calling tech support, email or chat bot when available.

## Assumptions and Dependencies

Constraints for the project include a budget of $150,000, a timeline of 5 weeks, and the project team, which would include, scrum master, product owner, developers, testers, software architects, quality assurance and user acceptance testers. Following the scrum guide, the project team will be no more than 8.

# External Interface Requirements

## User Interfaces

The Course Enrollment System will have the same layout throughout each page. Meaning, same header, footer, and search bar location. Also, the accessibility needs to follow the Web Content Accessibility Guidelines (WCAG). Again, the system needs to be mobile/tablet friendly, the student’s profile and home page button should be visible. The search process needs to be easy and smooth, allowing students to easily add or drop courses, be put onto a waitlist, and update their profile as needed. The password system needs to be stored securely.

## Hardware Interfaces

The Course Enrollment System will be accessible on Microsoft and Apple products, smartphones, tablets. It will utilize HTTP and TCP for making and transferring resources from the web server to the client. The system will support fingerprint login when appropriate and screen readers.

## Software Interfaces

The Course Enrollment System will be compatible with numerous operating systems which include Microsoft, Mac Android, iOS. An API will communicate with the database to store and retrieve information all user and course data. Also, when appropriate the site will allow for single sign-on or allow for signing in with Google or Facebook.

## Communications Interfaces

Users will have the ability to receive emails for registration completion, invalid login attempts, password reset, confirmation of changes, confirmation of course adds or drops, and email when courses are available based on their spot in the waitlist. They can also opt in to receive emails or SMS notifications for newsletters from the school. All personal identifiable information (PII) will be safe and secure, along with password encryptions.

# System Features

## New User Registration

4.1.1 Description and Priority

Priority - High

Description - Users will have the ability to register with the Course Enrollment System to select courses. A user cannot view courses or make changes until they are registered. This is one of the basic functionalities of the website.

4.1.2 Stimulus/Response Sequences

* A user will see a login or register options on the website
* A new user will click the register button
* The user will provide their information to create their account
* The user will click Submit
* User will receive a message stating: “A request has been sent to your email to finish the registration. Once completed please come back to this page.”
* User gets email finishes last component to register.
* User goes back to page and the page automatically refreshes, logging them into the website.

4.1.3 Functional Requirements

REQ-1: User should see two buttons on the website: Login and Register

REQ-2: User should see register form when “Register” is selected

REQ-3: User will receive message for any incomplete or missed information not completed

REQ-4: Data is sent and stored safely to the database when user hits submit

REQ-5: Email is triggered for user to complete registration

REQ-6: Website will display message: “A request has been sent to your email to finish the registration. Once completed please come back to this page.”

REQ-7: Website sends automatic email to user account

REQ-8: Once user confirms account, page is auto-refreshed and automatically signs the user into the website

## Login Feature for New and Returning Student

4.2.1 Description and Priority

Priority - High

Description – New and returning users can login to the website. A user who logs in can now select courses offered by the University. A user cannot manage their account until they login.

4.2.2 Stimulus/Response Sequences

* A user will see a login or register options on the website
* A user will click the login button
* The user will provide their login information
* The user will click Login
* Upon successful completion of logging in, the user will be directed to the home page with their username visible.

4.2.3 Functional Requirements

REQ-1: User should see two buttons on the website: Login and Register

REQ-2: User should see login form when “Login” is selected

REQ-3: User will receive message for invalid username/password combination

REQ-4: Data is sent and retrieved from database when user hits submit

REQ-5: User is redirected to home page with their username displayed on the page.

## Search Feature

4.3.1 Description and Priority

Priority - High

Description – User can search for courses and see course availability. A user cannot register for a course without being able to search for courses.

4.3.2 Stimulus/Response Sequences

* A user will type their requested course into the search bar
* User is provided with search suggestions along with category-wise suggestions
* User will have the ability to select course via a drop-down menu
* User will select a search suggestion
* User will see courses based on the search results

4.3.3 Functional Requirements

REQ-1: Website must have a search bar

REQ-2: Website must provide search suggestions to user

REQ-3: Website must display results when user selects one of the suggestions

REQ-4: Website must have a drop-down menu for course selection

## View Courses(s) Feature

4.4.1 Description and Priority

Priority - High

User can view their selected courses and be provided with the ability to drop courses as needed. A user cannot drop a course without being able to view their courses.

4.4.2 Stimulus/Response Sequences

* User will have the ability to view their courses
* User can drop courses by clicking “Drop”
* User will be prompted for confirmation to drop course
* User will click “Yes” or “No” to drop course

4.4.3 Functional Requirements

REQ-1: Users can view their courses by selecting the “See Course Schedule” button

REQ-2: Users can drop courses by selecting the “Drop” button

REQ-3: User can select the “Yes” or “No” button when dropping a course

## Register for a Course Feature

4.5.1 Description and Priority

Priority - High

Description - When user has registered and logged into their account they can begin registering for courses. A user needs to be logged in and have an account with the website to select course(s).

4.5.2 Stimulus/Response Sequences

* User clicks on “Register for Courses” link from Home page
* User can select the semester they wish to view
* User is provided with a list of courses
* User selects courses they wish to attend by clicking the checkbox next to the course
* User clicks “Register” when ready
* User receives confirmation and email confirmation message(s)
* User will be redirected to their personal registration page

4.5.3 Functional Requirements

REQ-1: Website will have link to “Register for Courses” on the Home page

REQ-2: Website will provide dropdown for user to select the semester in question

REQ-3: Website will provide checkboxes next to the courses the user wishes to attend

REQ-4: Website will provide “Register” button

REQ-5: Website will feed data selected to database

REQ-6: Website will display confirmation and email confirmation to user

REQ-7: Website will automatically send user to their personal registration page

## Drop a Course Feature

4.6.1 Description and Priority

Priority - High

Description – User can drop course(s). A user needs to be logged in and have an account with the website to drop course(s).

4.6.2 Stimulus/Response Sequences

* User clicks on “See Course Schedule” link from Home page
* User can view their course schedule
* User can select “Drop Course” next to course(s) in question
* User is prompted for confirmation
* User will click “Yes” or “No” to drop course
* User receives confirmation and email confirmation message(s)
* User will see updated course(s) list.

4.6.3 Functional Requirements

REQ-1: Website will have link to “Register for Courses” on the Home page

REQ-2: Website will provide checkboxes next to the courses the user wishes to attend

REQ-3: Website will provide “Submit” button

REQ-4: Website prompt user for confirmation

REQ-5: Website will select “Yes” or “No”

REQ-6: Website will display confirmation and email confirmation to user

REQ-7: Website will automatically page with updated courses

## Notification of Course Availability Feature

4.7.1 Description and Priority

Priority - High

Description – User will receive notification when a course is available from their waitlist.

4.7.2 Stimulus/Response Sequences

* User received email notification stating selected course(s) are available
* User logs into their account
* User see notification of course(s) available
* User selects course(s) they wish to add
* User is redirected to their updated course schedule

4.7.3 Functional Requirements

REQ-1: Website will send email to user when course is available

REQ-2: Website will allow user to login into their account

REQ-3: Website will display courses that are available per their waitlist

REQ-4: Website will allow user to select courses they want to register for

REQ-5: Website will redirect user to their updated course schedule

# Other Nonfunctional Requirements

## Performance Requirements

New students should be able to complete their registration in under one minute. When searching for courses, users should experience Near Real Time (NRT) results should appear within 5 seconds. Once course selection has been submitted, users should receive confirmation under 4 seconds and email under one minute. Speed is our top priority because students cannot afford lag times when registering for courses that have a student occupancy limit.

## Safety Requirements

Safety is another top priority; therefore, all personal identifiable information (PII) should be protected along with password encryptions. For users who become inactive during a session lasting 10 minutes, it should time out or the user should be prompted to advise if they wish to continue their current session.

## Security Requirements

PII should be protected and secured. When users call or email support there will need to be a verification process to ensure the correct user is being identified.

## Software Quality Attributes

The website needs to have great usability to allow for easy navigation and guaranteeing user satisfaction and repeat users. The website should be reliable, as stated above, the website needs to be available 24/7 outside of normal maintenance routines. The customer will be responsible for the addition of new features; therefore, the product needs to be maintainable and testable.

## Business Rules

If the user does not have an account, then they cannot access the website to search, add or drop courses, or maintain their account. The registration faculty need the ability to maintain courses meaning add or delete when needed. Administrators need to be able to assist with new and repeat users to help in a variety of activities that can include password resets, account issues, or course issues.

# Testing Levels and UML Models

## Component Testing

The first type of testing is Component testing (Module testing, Unit testing, or Program testing) is a software testing type. Typically, the software comprises several components, and testing is performed on each without involving other components; think of this as a bubble. As test scenarios are created, they will be broken into high-level test cases to low-level detailed test cases with Prerequisites. Component testing is performed by testers and is usually after the unit testing, which could explain why they sometimes share the same name (Hamilton, 2022, para 2). Component testing is performed without isolating other components, called Component Testing in Large. Still, if it is done with the isolation of another component, then it is referred to as Component Testing in Small. This is the first level of testing on the V-model development process and ties together with Component Specification. For the Course Enrollment System, testing will be for each functionality. So, the test will range from testing the registration function to ensuring a student can successfully sign up for the site. Another test could be testing a student adding a course to ensure the correct course, credits, and start/end date are displayed to the student. The Components will have several classes, and each must be tested thoroughly.

## Integration Testing

The next type of testing isIntegration testing, the second level of software testing, and units or individual software components are tested in a group. Integration testing aims to locate defects when interacting between integrated components or units. It meticulously picks module by module as a proper sequence is followed so that no scenarios are missed. Not all tests can perform perfectly or as expected; therefore, a set of positive and negative testing is performed; positive testing reacts as expected, while negative testing is when the unexpected happens. When negative testing happens, an issue or bug within the system must be reported to the developers (JavaTpoint, n.d., para 5). Lastly, there are strategies for testing, which can be top-down integration and starting from the top and working down the components; next is bottom-up integration, which begins with essential system components that do not call other components except for functions of the operating system, ad hoc integration is another test, that test components in the order in which they are finished, and finally is backbone integration, where a skeleton is built. Any component can be tested as integrated (Spillner et al., 2014, p. 57). Integration testing is the next step up the V-model process after the component and is aligned to the technical system design. The Course Enrollment System will undergo integration testing to ensure all faults and bugs have been found. One note to remember is even if everything passes component testing, it could fail during integration testing because the code is not flowing together as a fundamental component.

## System Testing

System testing is exactly what the name implies, testing the system, which is pushing the modules or components that are integrated to ensure the system works as expected or not. With system testing, the development team can deliver a high-quality product. Along with testing the system, the documentation created to support the system is testing, ranging from system manuals, user manuals, training manuals, etc. Testing configuration can also occur to test for performance and load testing. As companies depend on databases holding large amounts of data, it is becoming increasingly important for companies to test data quality in those systems. The system must be tested in multiple environments to ensure stability (Software Testing Help, 2022, para 3). System testing is the third testing in the V-model after Integration testing and connects to Functional system design. The Course Enrollment System will perform system testing to ensure all functional and non-functional requirements have been met; this environment will mirror a production-level product and be tested from a user perspective, yielding the best results.

## Acceptance Testing

Acceptance testing, which is the testing for acceptability or testing the compliance of the system utilizing the business requirements to ensure the system is ready for release or not. Acceptance testing has many forms, but for simplicity, this paper will concentrate on UAT or User Acceptance Testing. This testing is usually performed by the end-user or a Subject Matter Expert (SME). End-users test the functionality of the system or application as if they were already utilizing the product. When end-users test as if they were using the product already, it enables issues not found in the previous testing to be found, and that is because users know how the system and how it should act depending on what the user is trying to achieve. Acceptance testing is the last testing within the V-group model and aligns with the requirements definition. Therefore, the Course Enrollment System will go through acceptance testing and be performed by the end users to ensure requirements have been met and a high-quality website has been produced.

## Class Diagram

Diagram

Description automatically generated

## Sequence Diagram

Diagram

Description automatically generated

## Activity Diagram

Diagram

Description automatically generated

## State Diagram

Diagram

Description automatically generated

## Use Case Diagram

Diagram

Description automatically generated

# Landing, Login, and Enrollment Pages

## PHP and XAMPP

PHP or Hypertext Preprocessor is a scripting language that is used for web development and can be utilized along with HTML. XAMPP is an open-source web server package, which X is cross-platform, A is Apache Server, M is MariaDB, P is PHP, and the last P is Perl. Once you have installed XAMPP you can begin creating php documents under the htdocs folder that is a part of XAMPP. We can utilize an application like Notepadd++ to create a php document, for example. I can create a script <?php echo “Hello World”?> and save it as Hello\_World.php (Simplilearn, 2021, para 14). Now, I need to open the control panel for XAMPP and connect to Apache which allows me to host a local web server on my machine. Once connect, I can open a webpage from Edge, Chrome, or another internet browser and type localhost/Hello\_World.php and I should see a new webpage open saying Hello World. From there I can continue enhancing my Hello\_World.php document or I can create new documents (web pages) and connect them altogether, I just need to remember to call out the php document I am wanting to switch to. This will go into more detail later in the paper.

## MySQL

Through XAMPP I connected to MySQL and created my tables in phpMyAdmin. First, I created a new schema called course\_enrollmentdb which will house all the tables. For this week’s assignment, I only created a generic student which is used during the registration process. My table includes an id that is automatically incremented for each new user that is created, from there it includes the username, password (which is hashed), email, firstName, lastName, address, and phone number. Once the user has submitted their information and it is added to the database, they will use their unique username and password to access the site. This is happened by querying against the database to verify the username exists and is correct. Next, it checks the password and hashed password are correct, if everything matches and is verified the user is taking to their profile page where they can see their information.

## Registration Page

The Registration Page will ask the user to provide a username, password, confirm password, email, first name, last name, address, and phone. The first step was to include the header information which is under the master.php, so I did a <?php require ‘master.php’; ?> after that I included a “wrapper” to keep the encloses the elements together in the HTML. Then I labeled each identifier with a type, name, and class, this helps me identify what each identifier is for and is used for the MySQL $\_POST part to get the information into the correct table based on the names provided. Once a user completes their registration and clicks ‘Submit’ the information is loaded into the MySQL table called ‘student’ which as an id that is the primary key and automatically incremented for each new user that gets added to the system. Then when the user logs into the system the id, username, and password are matched on the table to make sure the correct user information is being pulled into the system. To check how many users are in the database I can perform SQL \* FROM student to the everything in the table (Connolly & Hoar, 2018, p. 633).

## Screenshot of Home Page

Graphical user interface, text, application

Description automatically generated

## Screenshot of Registration Page

A picture containing table

Description automatically generated

## Screenshot of Login Page

Graphical user interface, application, website

Description automatically generated

## Screenshot of Login Page

Graphical user interface, text, application, email

Description automatically generated

## Screenshot of Reset Password Page

Graphical user interface, text, email

Description automatically generated

# Course Registration and MySQL Database

## MySQL Database

Using MySQL database, we first start with a database called course\_enrollmentdb which in the moment is empty. There are two options; first, I can create the tables using phpMyAdmin and manually create the tables or I can use a MySQL statement. I decided to use a CREATE TABLE sql statement to create all my tables at once. I assigned primary and foreign keys to associate the appropriate tables together. After that, I did a INSERT INTO specific tables I wanted data to be pre-populated into. This saved time and made it easier from the student perspective to pick courses. I decided not to pre-populate student data to ensure that the php and sql code performed correctly.

## Registration Functionalities

This week, I added the ability for a student to view their schedule, register for a specific year and semester, along with adding courses. The pages are very simplified using the master.php portion to obtain the header information along with the footer.php to maintain the bottom portion. After a student signs-in they can view their current schedule (if any). From there, they can register for a specific semester and year, click submit, and then select the courses available. Once they are finished, they can click view schedule to see what classes they are scheduled for and can drop classes they no longer want to be enrolled in.

For any classes that are full, the student will receive a message advising the course is full, but they have been added to the waitlist and will be notified once the class is open and they are automatically added to the course when it becomes available.

## Empty Database Screenshot

Graphical user interface, text, application

Description automatically generated

## Database and Tables Created Screenshot

Background pattern

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

## Course Table Empty Screenshot

Graphical user interface, text, application, email

Description automatically generated

## Course Table Data Screenshot

Graphical user interface, text, application, email

Description automatically generated

## Available Table Empty Screenshot

Graphical user interface, text, application, email

Description automatically generated

## Available Table Data Screenshot

Graphical user interface, application, email

Description automatically generated

## Waitlist Screenshot

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Notification Screenshot

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

# Test Management

## Test Teams

The development team is responsible for testing but should not test their code. Instead, developers need to test each other’s code. In addition, there needs to be a dedicated testing team and independent test specialists, which will perform performance tests, usability tests, security tests, or to show conformance to standards and regulatory rules (Spillner et al., 2014, p.170). The project manager must also ensure that testing standards and regulations are applied and require test logs from the testers.

## Tasks and Qualifications

**Test Manager** – Expert knowledge in test planning and control. Will write and coordinate the test plan, represent the testers, procure resources, and introduce test strategies.

**Test Designer** – Expert in test methods and specifications. Will review requirements, specifications, and models for testability to design test cases. It will also help in acquiring test data.

**Test Automator** – Must have programming experience, testing basics, and knowledge of testing tools and script languages. Will automate tests as needed, making use of tools provided.

**Test Administrator** – Will be used to setting up the testing environment along with the system administrator and network management.

**Tester** – This will range from developers and QA testers to executing tests and reporting failures to end users participating in User Acceptance Testing (UAT).

## Exit Criteria

Identifying straightforward test entry and exit criteria is essential to test planning. The exit criteria are needed to make sure testing is not stopped early. The teams will ensure the environments are ready, test tools are available, test objects are installed, and the data is prepared to be tested (Spillner et al., 2014, p. 184). 100% requirements coverage must be achieved; a defined defect count is reached with no blockers or known critical defects. The test coverage should be at a 95% achieved rate, and for the remaining 5%, they must be low-priority fails. Finally, testing documents are published, and the go/no-go meeting has been completed.

## Estimated Cost Effort

The test manager should estimate the testing efforts to be expected. Since this is a new system, we can consider this a large project; thus, the test manager must list all testing tasks and let either the test owner or experts estimate each job or use previous experience based on similar projects. Previous experience usually provides better results (Spillner et al. 2014, p. 184).

## Incident Reporting

Incident reporting is a written description of an incident observed during testing. First, an incident is a variation or deviation observed in system behavior from what is expected. After an incident is found, a detailed report is prepared, logged, and tracked; there can be several incidents within the software, and therefore, reporting is needed to keep track of these. The information is also essential for developers to deliver solutions to the issue or help categorize which incidents take higher precedence over another. The report should include the following elements: summary, description, test data, severity, priority, steps used, and results (STC Admin, 2016). The following concept map provides a small example of Incident Reporting:

Diagram

Description automatically generated

## Defect Classification

They will need to be put into categories to capture and manage reported defects because not all defects are of high importance. Defect classification is broken into 5 classes: 1-Fatal, 2-Very Serious, 3-Serious, 4-Moderate, and 5-Mild. When creating a defect, the user's point of view should be considered when selecting a severity and determining how quickly the defect needs to be resolved. These are called fault priority and have four priority levels: 1-Immediate, 2-Next Release, 3-On Occasion, and 4-Open. The priority selection depends on when the fix is released into production (Spillner et al., 2014, p. 197). Labeling these classes and priorities allows the developers and testers to understand what needs to be fixed and tested first. Below is a concept map of Defect Classification.

**Diagram

Description automatically generated**

## **Configuration Management**

Configuration management will be important within the Course Enrollment System because if not done correctly, we can risk overwriting modifications, impeding integration activities, problem analysis, fault correction, and regression testing are difficult, and test/test evaluation are hindered. This means that version management, configuration identification, incident and change status control, and configuration audits must be implemented to ensure protection (Spillner et al. 2014, p. 201).

## **Testing Tools**

Introduction

Test tools are used for improving testing efficiency by adding automatic testing and reducing, if not removing, manual testing (as much as possible). It can be used for enabling tests; this means having tools to analyze and make execution possible to do that otherwise might be impossible to do manually. Test tools support improving testing reliability by automating manual tasks like comparing large amounts of data or simulating program behavior (Spillner et al. 2014, p. 205). The following are phases of testing tools that can be utilized in the Course Enrollment System, tools for management and control testing, test specification, static testing, dynamic testing, and non-functional testing.

Tools for Management and Control Testing

Tools for Management and Control will need the ability to provide easy documentation, prioritization, listing, and maintenance of test cases. This allows the tester to evaluate if, when, and how often a test case has been executed. There are advanced test management tools that support requirements-based testing. The tool also needs the ability to maintain incident management for test managers, testers, or developers to report incidents and managers to set prioritizations around those incidents. Configuration management is not a testing tool per se. Instead, it makes it possible to track different versions and builds of the software to be tested. Tool integration is becoming extremely important as it allows tools to work together to move requirements into test planning, test execution tools are then used and supplied with test scripts, management tools are also coupled within incident tools and help support the need for retesting, and finally, configuration management allows for every code change to be accounted for (Spillner et al., 2014, p 208). I would recommend Bitbucket along with Jira for testing management and control. It has a simple interface, is easy to use, has built-in requirements management, allows for easy tracking of incidents, defects, or any other issues that need to be captured, and has complete visibility for teams.

Tools for Test Specification

Tools for Test Specification, to have test cases be reproducible, they need to be specified. Database-based test data generators process database schemas and can produce test databases from the schemas. Code-based test data generators analyze the source code to create test data; however, only existing code can be tested, and any code missing that could contain faults will be unaccounted for; this is best suitable for regression or platform tests if a system is already existing or reliable. Interface-based test data generators are used to test the object’s interface and its parameter domains and use. These tools can be used for APIs or GUIs and are well suited for the automatic generation of negative tests because the specific target values are not necessary. It is whether errors are produced or not. Specification-based test data generators use a specification to derive test data and corresponding expected results (Spillner et al., 2014, p 210). Using a MySQL database table, I recommend the EMS Data Generator from SQL Manager.net. You can save and edit generated data in SQL script and preview the data generated. It supports data types like SET, ENUM, GEOMETRY types, etc., and offers various generated parameters for each field type.

Tools for Static Testing

Tools for Static Testing can check for defects in software applications without executing the code (Hamilton, 2022, para 1). These tools are valuable in testing use case requirements, which validate that all end-user actions are identified; functional requirement validation identifies all necessary elements while looking at database functionality, interface listings, hardware, software, and network requirements. For static testing, I would recommend using CAST as it is an automated tool with more than 50+ languages, the project size does not matter, and it has a user-friendly dashboard to measure quality and productivity.

Tools for Dynamic Testing

Tools for Dynamic Testing can test the software with existing or current data. Test drivers provide the input data to a module-under-test (MUT) (GeeksforGeeks, 2020, para 8). Debuggers allow for part of line-by-line testing of a program; simulators can be used to test the application environments if the final system is not ready. Test robots can log all manual tester inputs and then create test scripts from this, which can allow the tester to automatically “play” the test again and again; however, the tester must think this process through because depending on the test being executed, the information may already exist in the system and will either produce an error or ignore the data and continue, which may or may not be appropriate. There are numerous tools for dynamic testing, but Daikon is an open-source tool that detects likely invariants of a program and works with popular programs such as C, C++, Java, Perl, and more. It is also easily extendable to other applications. Most IDEs or text editors come with some Debugger, which will be needed for the Course Enrollment System.

Tools for Nonfunctional Testing

Non-functional testing tests performance, security, usability, and compatibility; load and performance testing are used for parallel database queries, user transactions, or network traffic and can show volume, stress, and performance. Performance measures and records the response time behavior of the system being tested depending on the load size. Security testing tools look for vulnerabilities within the system, and virus scanners and firewalls can also fall into this category (Spillner et al., 2014, p. 217). Usability and compatibility ensure the application is ready for use and can be used on various platforms. Before and after any conversion or migration, the data must be checked to ensure the quality remains intact; this is data quality assessment. Apache JMeter, widely used for reliability and automation testing, focuses on web applications; as a bonus, it is free of cost and could be an excellent option to incorporate within the Course Enrollment System. It is user-friendly and has a good reputation for compatibility testing, measuring, and analyzing the performance of programs.

## **Appendix A: Glossary**

PII – Personally Identifiable Information

SRS – Software Requirements Specification

UAT – User Acceptance Testing

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

GitHub

![Graphical user interface

Description automatically generated]()

## **References**

Bernstein, C (n.d.). Personally Identifiable Information (PII). <https://www.techtarget.com/searchsecurity/definition/personally-identifiable-information-PII#:~:text=Personally%20identifiable%20information%20(PII)%20is,anonymous%20data%20is%20considered%20PII>.

Connolly, R., & Hoar, R. (2018). [Fundamentals of web development](https://ashford.redshelf.com/app/ecom/shelf/course-section/3829410) (2nd ed.).

GeeksforGeeks (2019, April 30). *Acceptance Testing | Software Testing.* <https://www.geeksforgeeks.org/acceptance-testing-software-testing/>

GeeksforGeeks. (2021, December 13). Software Requirement Specification (SRS) Format. <https://www.geeksforgeeks.org/software-requirement-specification-srs-format/>

Hamilton, T. (2022, February 26). *What is Component Testing? Techniques, Example Test Cases.* <https://www.guru99.com/component-testing.html>

Kumar, Jayant (2015, April). Apache Solr Search Patterns. <https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781783981847/1/ch01lvl1sec12/the-e-commerce-problem-statement>

JavaTpoint. (n.d.). *Integration testing*. <https://www.javatpoint.com/integration-testing>

Software Testing Help (2022, April 3). *What is System Testing – A Ultimate Beginner’s Guide.* <https://www.softwaretestinghelp.com/system-testing/>

Javatpoint. (n.d.) XAMPP Tutorial. <https://www.javatpoint.com/xampp>

Simplilearn. (2021, September 14). How to Run a PHP File Using XAMPP: A Step By Step Guide. <https://www.simplilearn.com/tutorials/php-tutorial/php-using-xampp>

Spillner, A., Linz, T., & Schaefer, H. (2014). [Software testing foundations: A study guide for the certified tester exam](https://ashford.instructure.com/courses/106524/modules/items/5408048)(4th ed.).

STC Admin. (2016, December 6). *What is Incident Report in Software Testing?* <https://www.softwaretestingclass.com/what-is-incident-report-in-software-testing/>

Tsui, F., Karam, O., & Bernal, B. (2018). [Essentials of software engineering](https://ashford.instructure.com/courses/106524/modules/items/5408048) (4th ed.)