Final Software Project

Austin Blair

CST499: Capstone for Computer Software Technology

Joseph Rangitsch

3-3-25

Software Requirements Specification (SRS)

Software Requirements Specification

for

Online Student Enrollment Website

Version 1.0 approved

Prepared by Austin Blair

UAGC

Table of Contents

Table of Contents ii
Revision History ii
1. Introduction 1
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Intended Audience and Reading Suggestions 1
1.4 Project Scope 2
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Features 3
2.3 User Classes and Characteristics 3
2.4 Operating Environment 4
2.5 Design and Implementation Constraints 4
2.6 User Documentation 4
2.7 Assumptions and Dependencies 5
3. System Features 5
3.1 New User Account Registration 5
3.2 User Login 6
3.3 Course Search 7
3.4 Register for a Course 8
3.5 View Course Schedule 9
3.6 Drop Course 10
3.7 User Logout 11
3.8 Waitlist Notification System 12
4. External Interface Requirements 13
4.1 User Interfaces 13
4.2 Hardware Interfaces 13
4.3 Software Interfaces 13
4.4 Communications Interfaces 14
5. Other Nonfunctional Requirements 14
5.1 Performance Requirements 14
5.2 Safety Requirements 15
5.3 Security Requirements 15
5.4 Software Quality Attributes 15
Appendix A: Glossary 16
References 17

The Software Requirements Specification document provides a clear outline of the goals of the Online Course Enrollment website and how it will function. The SRS specifies the details of the software build including System Features, External Interface Requirements, and Nonfunctional requirements such as safety, performance, security and other quality attributes.

Software Requirements Specification (SRS)

One of the key sections of the SRS is System Features. This section explains the different functions that are required by the system to satisfy end-user requirements.

System Features:

- New User Account Registration
- User Login
- Course Search
- Register for a Course
- View Course Schedule
- Drop Course
- User Logout
- Waitlist Notification System

3.3 Course Search

3.3.1 Description and Priority

- ◊ Description: Users should have the ability to search for available courses. Course searches shall be filtered by semester (Spring, Summer, Fall). This will ensure that only available courses are displayed.
- ◊ Priority: High
- ◊ Benefit: High

3.3.2 Stimulus/Response Sequences

- ◊ User selects “Course Search” tab from the “Course Registration Page”
- ◊ User selects the "Semester” filter option for either Spring, Summer, or Fall.

3.4 Register for a Course

3.4.1 Description and Priority

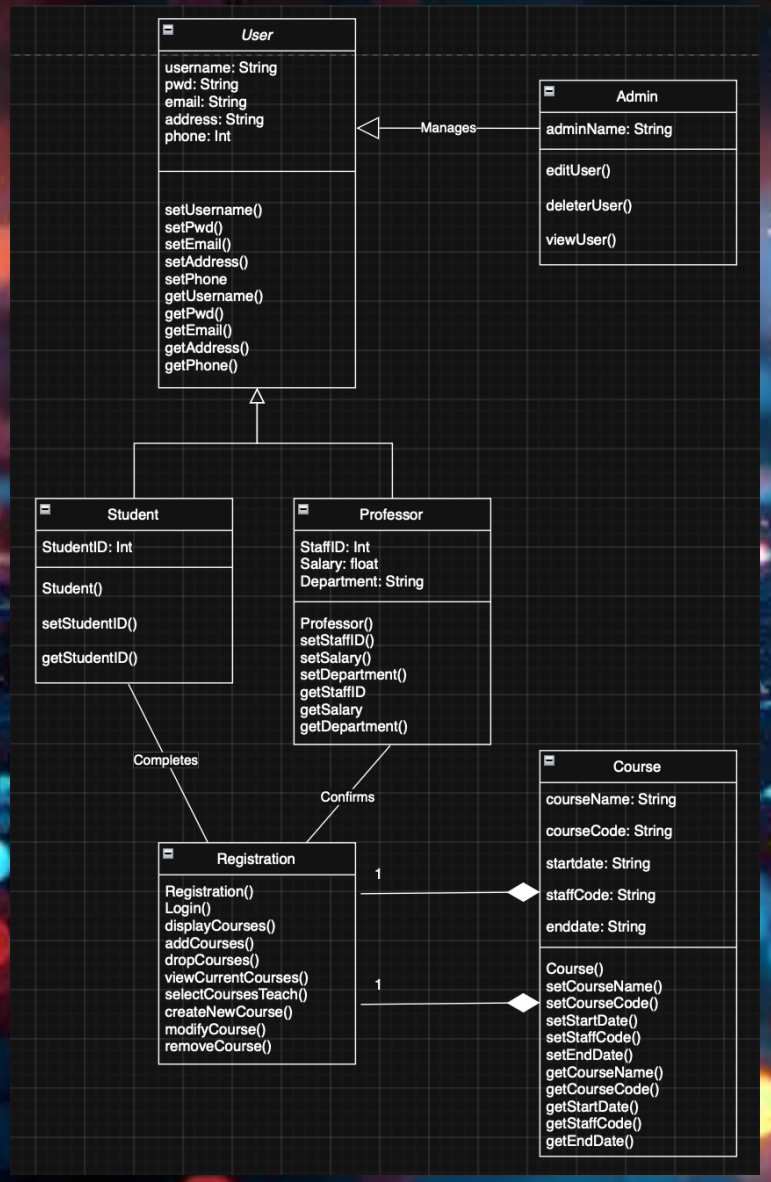
- ◊ Description: Users should have the ability to register for available courses. This function is dependent on a current user account.
- ◊ Priority: High
- ◊ Benefit: High

3.4.2 Stimulus/Response Sequences

- ◊ Once logged in, the user selects the “Register for Course” tab from the “Course Registration” home page.
- ◊ User selects the "Semester” filter option for either Spring, Summer, or Fall.
- ◊ User selects the “Submit” once the chosen semester is selected.
- ◊ User is provided with available courses for the specified semester.
- ◊ User selects which course that is desired for enrollment and selects submit.
- ◊ If the course is available, the user receives a successful registration message.
- ◊ If the course is full, the user will be notified, and presented with the option of being placed on a waiting list
- ◊ If the waiting list option is selected, the user is redirected to the “Course Registration” home page to enable further class searches.

UML Models

Class

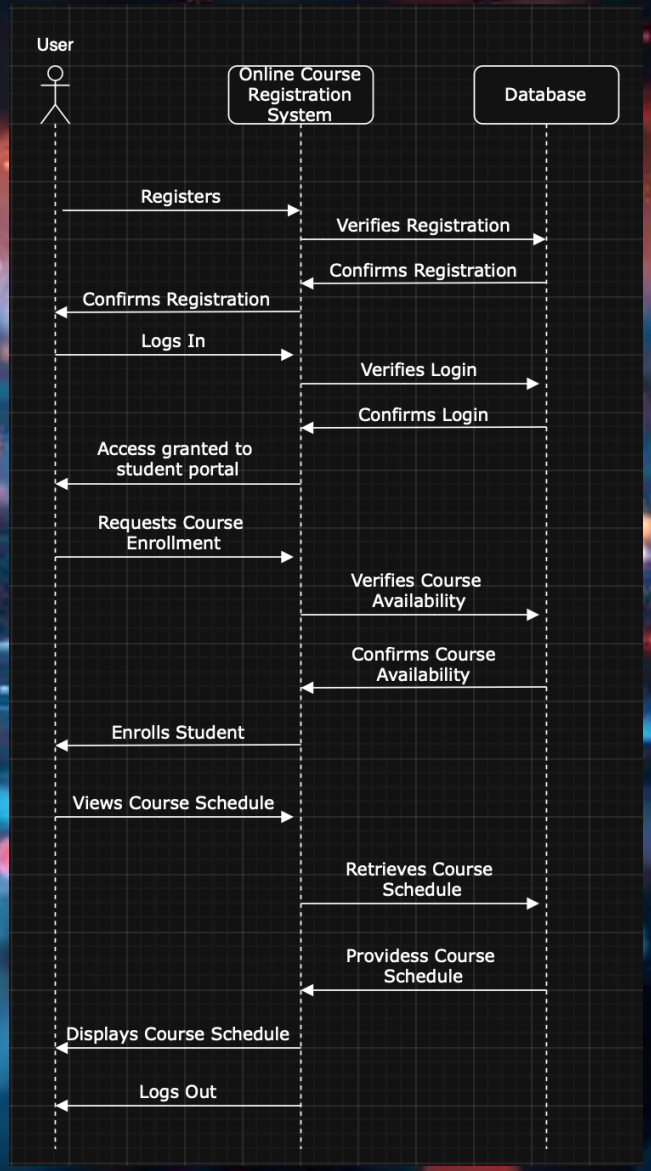


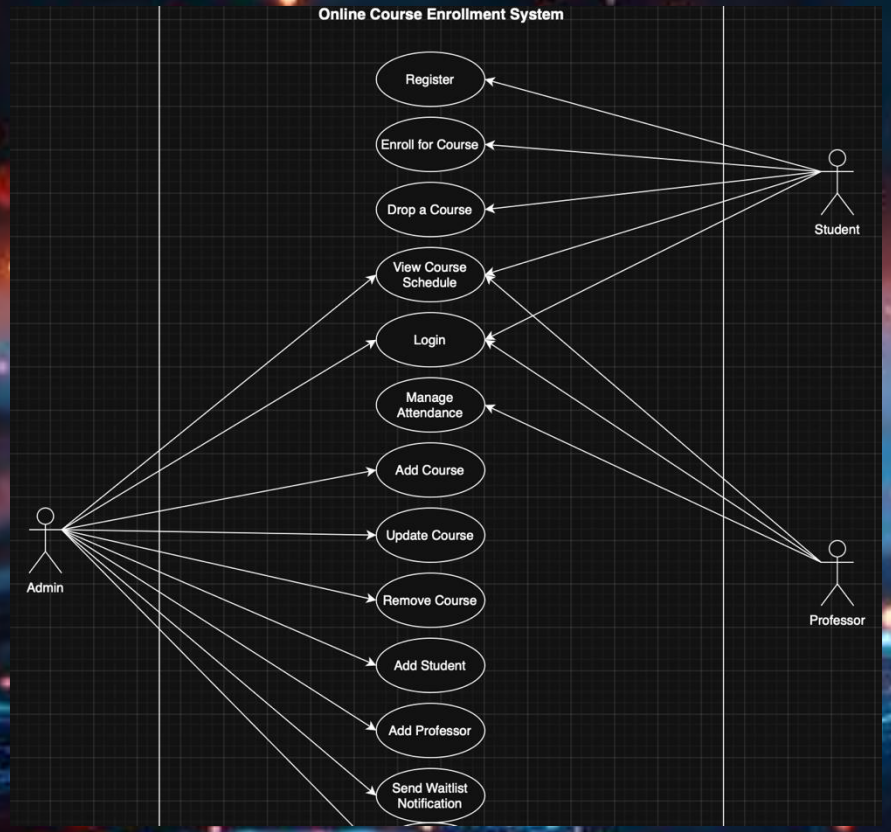
Unified Modeling Language (UML) models are a general-purpose modeling language that defines a standard way to visualize a software system's design, behavior, and structure.

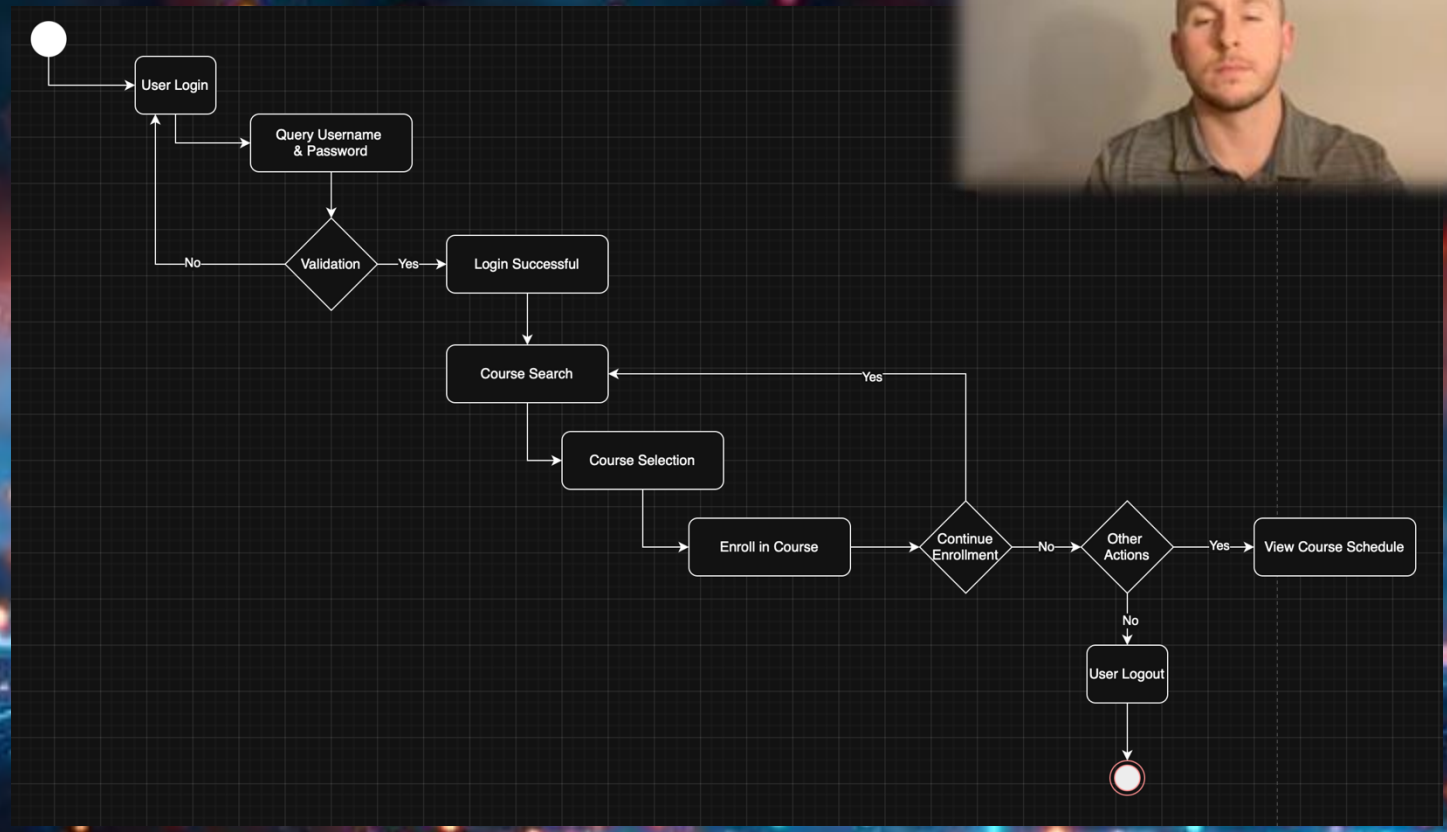
The **Class** diagram depicts the static structure of the system by displaying the system's classes, methods, and attributes.

The **Sequence** diagram displays how and in what order the objects in the Online Course Enrollment System function.

Sequence







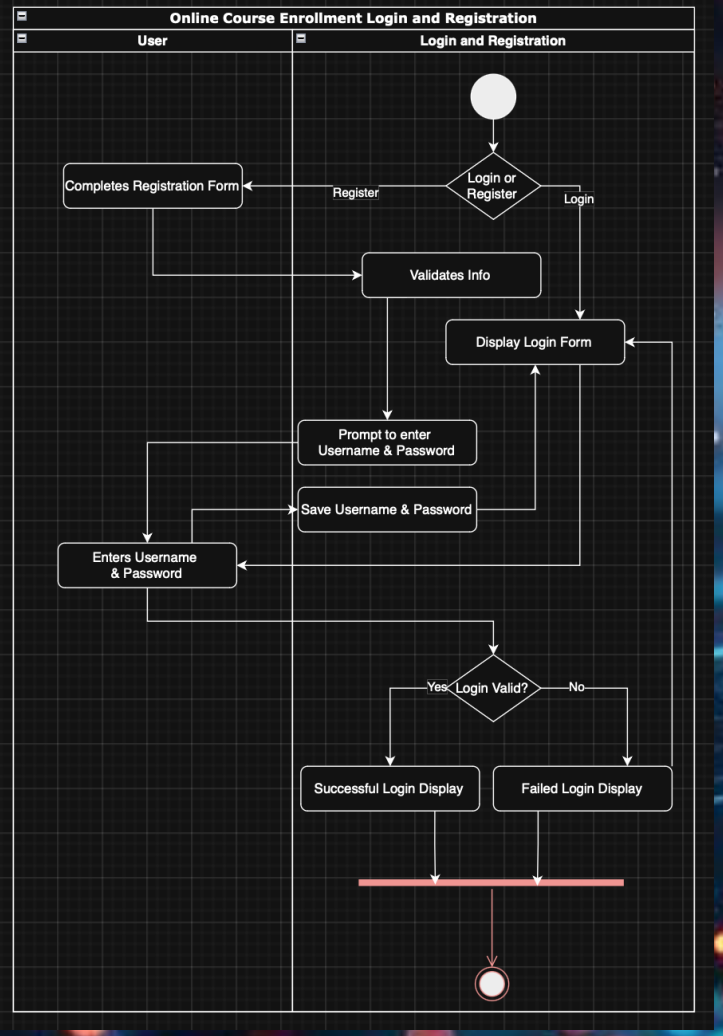
UML Models

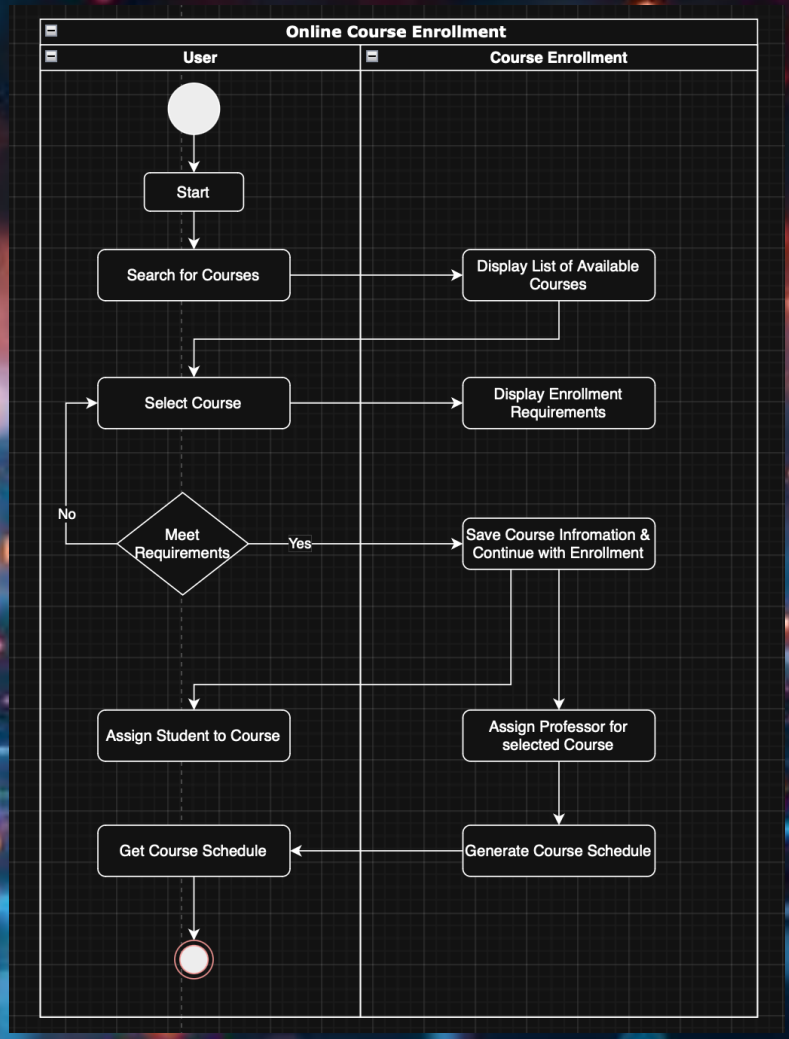
The Use-Case diagram, as seen in the Figure above, illustrates the system's functionality by showing the different actions permitted by each user of the system .

The State Machine diagram, as seen above, represents the condition of the Online Course Enrollment system at fixed instances in time.

UML Models

Activity diagrams represent the flow of control in the Online Course Enrollment system.



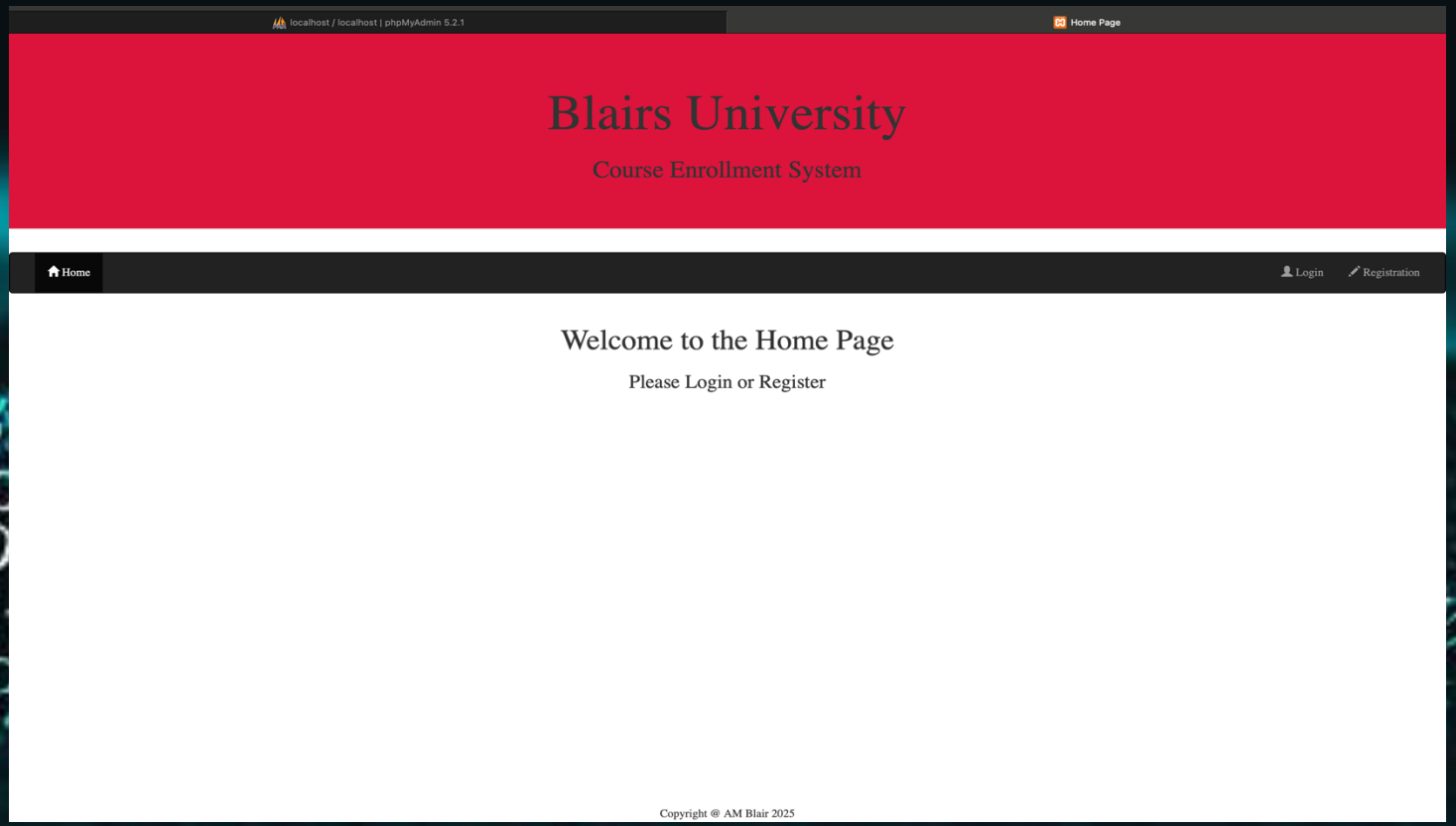


The diagram above represents the flow of control for user Login & Registration

The diagram on the right represents the flow of control for users enrolling in courses.

Landing Page

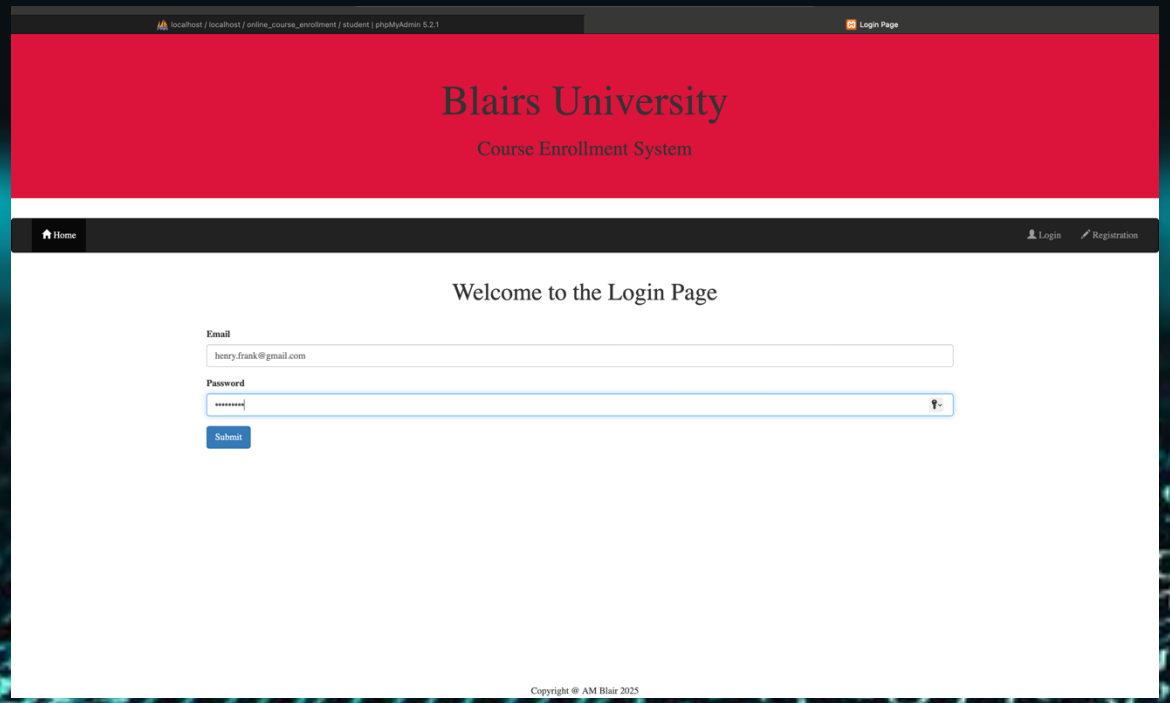
The Landing page is the initial page users are presented with when accessing the Online Course Enrollment website. The goal of this page was to create a uniform and user-friendly display, with ease of navigability for the available functions.

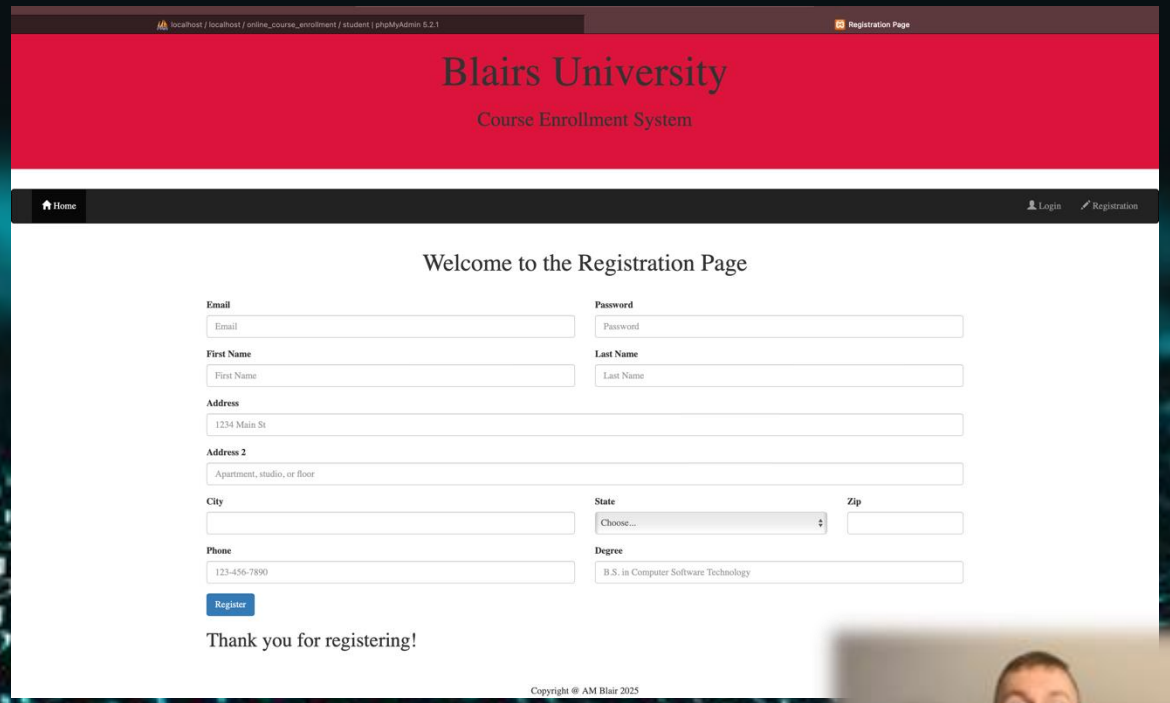


Login | Registration

The Login page, on bottom left, allows registered users to login and view profile details, view/alter their course schedule, and register for courses.

The Registration page, below, enables users who do not have an account to create one. As you can see in the snippet below, once an account is created the message “Thank you for registering!” is displayed signifying that the data was saved into the database.

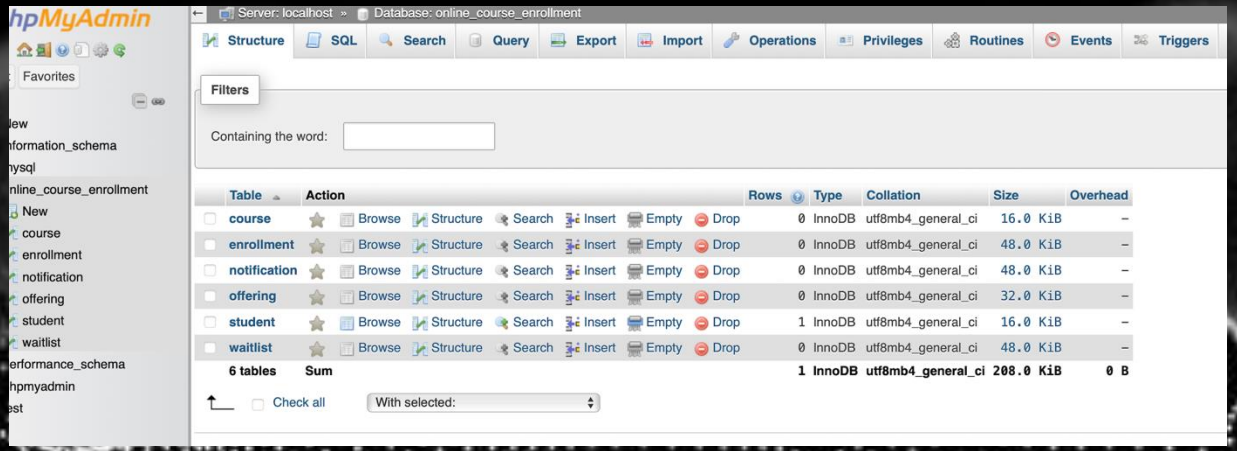


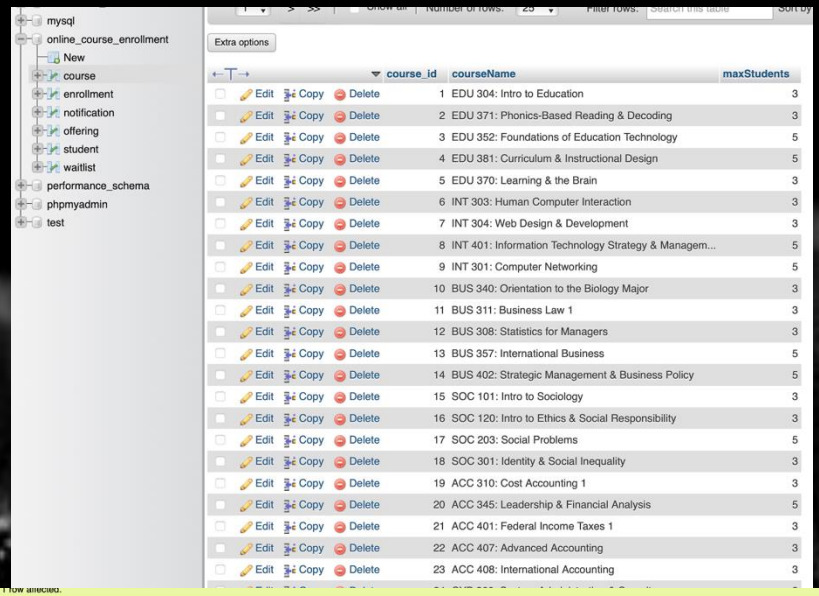


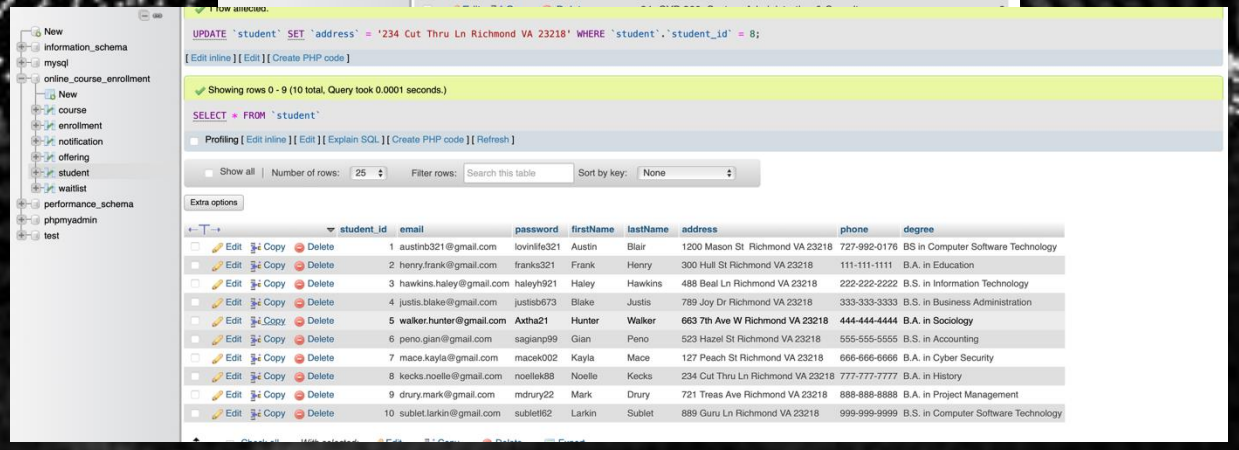
MySQL Database

The first image below is a snippet of the offering table. This table incorporates the entirety of the available courses offered on the Online Course Enrollment Website. The courses were inserted into the database using the populateData.sql file.

The second image below is the users table. This table displays the different users who have created accounts using the registration page. The users table is accessed each time a user logs in, permitting access to their profile page and website functions such as viewing/altering their course schedule or searching and enrolling for additional courses.





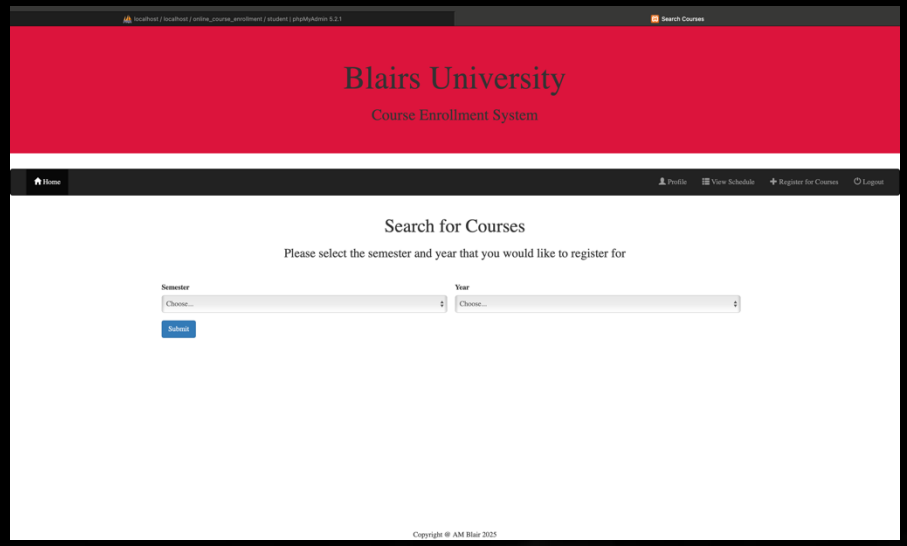


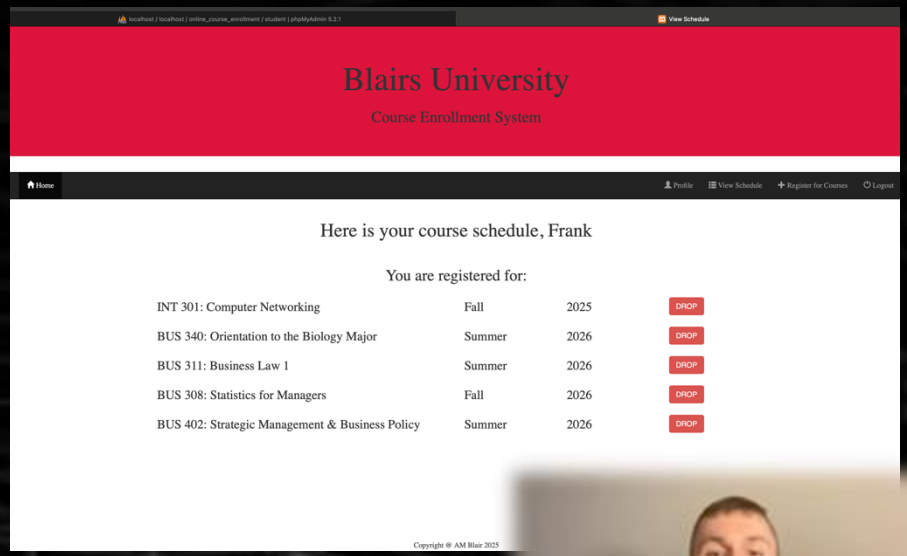
The database named online_course_enrollment consists of six different tables as seen in the above image and acts as the backend of the website.

Class Registration

To allow users to search and enroll for courses, the **Register for Courses** tab in the navigation bar directs users to the page on the top right. This page allows users to filter available courses by semester and year. Once selected, available courses are presented, and the user may either select the course(s) they wish to enroll in or initiate another search.

Once enrolled in courses, users can select the **View Schedule** tab on the navigation bar which directs them to the course schedule page. This page allows users to view and verify their schedule, with the option to drop courses as needed.





PHP Code

PHP is a server-side, open-source, scripting language design specifically for web development. In the development of the Online Course Enrollment website, PHP was integrated with MySQL to process user inputs to and from the database/backend. Additionally, it was used to generate content for the overall design through encapsulation of HTML code.

The segments of PHP code below incorporate the logic behind the sole purpose of the website: enroll in courses. Each PHP code file communicates with the database to enable users to complete a desired action. These include search for available courses, add courses based on availability, and view their course schedule. The logic which enables users to drop classes is encapsulated within the view schedule code.

viewSchedule.php

```php
<?php

    error_reporting(error_level: E_ALL ^ E_NOTICE);
    require_once('Connect.php');
    unset($_SESSION['dropOfferingId']);
    unset($_SESSION['droppedCourseName']);
    unset($_SESSION['droppedSemester']);
    unset($_SESSION['droppedYear']);
    unset($_SESSION['numStudentsEnrolled']);
    unset($_SESSION['maxStudents']);
    unset($_SESSION['numStudentsOnWaitlist']);
    unset($_SESSION['waitlistedStudentId']);
    unset($_SESSION['dateTimeAdded']);
    $myConnection = $newConnection->$connection;

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <title> View Schedule </title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-sacle=1">
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
    <link rel="stylesheet" href="index.css">
    <script src="https://ajax.googleleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
    <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>
<body>

<?php include 'master.php';?>

<div style='margin-bottom:60px' class="container text-center">
        <?php

            if(isset($_SESSION['username'])) {
                echo "<h1>Here is your course schedule, ".$_SESSION['username']."</h1>";
                echo "<br>";
                echo "<h2>You are registered for:</h2>";

                displayCourseSchedule(connection: $myConnection,studentId: $_SESSION['studentId']);
            }
            else {
                echo "<h1>Course Schedule Page</h1>";
                echo "<h3>Please login or register</h3>";
            };
```

View Course Schedule

addCourse.php

```php
<?php
    error_reporting(error_level: E_ALL ^ E_NOTICE);
    require_once('Connect.php');
    unset($_SESSION['selectedOfferingId']);
    unset($_SESSION['registered']);
    unset($_SESSION['waitlisted']);
    unset($_SESSION['numStudentsEnrolled']);
    unset($_SESSION['maxStudents']);
    require 'master.php';
    $myConnection = $newConnection->$connection;
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title> Add Course </title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-sacle=1">
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
    <link rel="stylesheet" href="index.css">
    <script src="https://ajax.googleleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
    <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>
<body>

    <div class="container text-center">
        <?php

            if($_SESSION['selectedYear'] == 2024 && $_SESSION['selectedSemester'] == 'Spring') {
                echo "<h1>Sorry, registration for ".$_SESSION['selectedSemester']." ".$_SESSION['selectedYear']." is
            } else {
                echo "<h1>Register for ".$_SESSION['selectedSemester']." ".$_SESSION['selectedYear']."</h1>";
                echo "<h3>Please select the Course that you would like to Register for</h3>";
            }

        ?>
        </div>
        <div style='margin-bottom:60px' class="container">
        <form class="padding-top" method="post" action="<?php echo htmlspecialchars(string: $_SERVER["PHP_SELF"]);?>"
        <div class="form-row">
        <div class="form-group col-md-12" id="no-padding-left">
                <label for="inputCourse">Course</label>
                <select id="inputCourse" class="form-control" name="course" required>
                    <option>Choose...</option>
                <?php

                    $availableCoursesArray = array();
                    $availableCoursesArray = getAvailableCourses(connection: $myConnection,year: $_SESSION['selectedY
                    foreach($availableCoursesArray as $data) {
```

Add Course

searchCourses.php

```php
<?php
    error_reporting(error_level: E_ALL ^ E_NOTICE);
    require_once('Connect.php');
    unset($_SESSION['selectedSemester']);
    unset($_SESSION['selectedYear']);
    unset($_SESSION['selectedCourse']);
    unset($_SESSION['selectedOfferingId']);
    require 'master.php';
    $myConnection = $newConnection->$connection;
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title> Search Courses </title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-sacle=1">
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
    <link rel="stylesheet" href="index.css">
    <script src="https://ajax.googleleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>
    <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container text-center">
        <h1>Search for Courses</h1>
        <h3>Please select the Semester and Year that you would like to register for<h3>
    </div>
    <div style='margin-bottom:60px' class="container">
        <form class="padding-top" method="post" action="<?php echo htmlspecialchars(string: $_SERVER["PHP_SELF"]
            <div class="form-row">
                <div class="form-group col-md-6" id="no-padding-left">
                    <label for="inputSemester">Semester</label>
                    <select id="inputSemester" class="form-control" name="semester" required>
                        <option>Choose...</option>
                        <?php
                            $semestersArray = array();
                            $semestersArray = getSemestersAvailable(connection: $myConnection);
                            foreach($semestersArray as $data) {
                                echo "<option>".$data['semester']."</option>";
                            }
                        ?>
                    </select>
                </div>
                <div class="form-group col-md-6" id="no-padding-left">
                    <label for="inputYear">Year</label>
                    <select id="inputYear" class="form-control" name="year" required>
                        <option>Choose...</option>
                        <?php
```

Search for Courses

THANK YOU

References

Connolly, R., & Hoar, R. (2022). *Fundamentals of web development* (3rd ed.). Pearson.

Spillner, A., Linz, T., & Schaefer, H. (2014). *Software testing foundations: A study guide for the certified tester exam* (4th ed.). Rocky Nook.

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

W3Schools. (n.d.). PHP Introduction. https://www.w3schools.com/php/php_intro.asp