

---

# **Software Requirements Specification**

**for**

# **Student Online Course Enrollment**

**Version 2.0**

**Prepared by Jason Simms**

**University of Arizona Global Campus**

**CST499: Capstone for Computer Software Technology**

**Dr. Amr Elchouemi**

**Monday, January 24, 2022**

# Table of Contents

<b>Table of Contents .....</b>	<b>ii</b>
<b>Revision History .....</b>	<b>ii</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose .....	1
1.2 Document Conventions .....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Project Scope .....	2
<b>2. Overall Description .....</b>	<b>2</b>
2.1 Product Perspective .....	2
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 .....	3
2.7 Assumptions and Dependencies .....	3
<b>3. Software Requirements Specifications.....</b>	<b>4</b>
3.1 Students can create an account and profile in the system.....	4
3.2 Student IDs should be unique and associated with a password.....	5
3.3 Profiles must include required information .....	5
3.4 Students can log into system with ID and password .....	6
3.5 Students can view courses by semester .....	7
3.6 Students can register for selected courses .....	8
3.7 Courses should have a maximum student count.....	8
3.8 Students can add themselves to waitlist if course is full .....	9
3.9 Students can cancel enrollment in courses .....	10
<b>4. UML Models and Testing Levels.....</b>	<b>11</b>
4.1 Diagrams.....	11
4.2 Testing Introduction .....	16
4.3 Component Testing .....	17
4.4 Integration Testing .....	17
4.5 System Testing.....	18
4.6 Acceptance Testing .....	18
<b>Appendix A: Glossary.....</b>	<b>19</b>
<b>Appendix B: System Figures.....</b>	<b>20</b>
<b>Appendix C: To Be Determined List (Pending).....</b>	<b>26</b>
<b>Appendix D: References .....</b>	<b>26</b>

## Revision History

Name	Date	Reason For Changes	Version
Jason Simms	12/13/2021	Initial Version	1.0
Jason Simms	12/20/2021	Added UML Models and Testing Levels	2.0

# **1. Introduction**

## **1.1 Purpose**

The purpose of this SRS is to present a detailed description and layout of the system. The document will explain the features, interfaces, capabilities, and constraints of the system. The document is intended for stakeholders and developers and will be presented to the University for approval.

The purpose of this project is to build an online enrollment system (OES) for students. This system will provide the following features:

- New user registration that includes account and profile creation.
- Each new user should have a unique ID associated with a password. The system should guard against two users using the same ID for registration.
- Profiles must include some key information about the applicant including name, phone, email, and any other information you may see necessary.
- Post registration, users can login to the system at any time using the ID and the password created during the registration process.
- Online courses run through three semesters per year (spring/summer/fall), and students can list the courses that will be offered during any semester, as not all courses will be offered in every semester.
- Each course should have a maximum number of enrollments that may be different depending on the course.
- If a user wants to enroll into a course and the course is full, the student can add themselves onto a waiting list.
- A user can also cancel the enrollment from any course that they are enrolled in, and the system should inform the first in the waiting list (if any) that they can enroll into the class.

## **1.2 Document Conventions**

All section and sub-section headings can be found in bold text within the document.

## **1.3 Intended Audience and Reading Suggestions**

This document has been created for the development team, designers, project managers, and testers. This group should be very familiar with this document. The developers and designers need to understand the requirements, purpose of the project, and functional and non-functional requirements to ensure a cohesive working build of the student enrollment system. The project managers need to understand each piece to effectively lead the work and produce a final shippable product. The testers need to understand the website and its functionality and features to develop robust test cases that provide solid feedback to the developers. This document is also available for review for the stakeholders and external companies providing hardware and functionality.

## **1.4 Project Scope**

The scope of this project includes the development of the student OES for the University. It also includes the development and maintenance of a student and course database and API setup for third-party applications.

# **2. Overall Description**

## **2.1 Product Perspective**

The OES will be the first system of its kind to be launched at the University. This system will be beneficial for both students and faculty. The OES will provide students with the ability to view detailed class information and register online without the need to formally register in person at the University. Faculty will have the ability to create course registrations online and view detailed student registration information. This system will reduce administrative burden, create a more efficient registration process, and provide information more readily to students and faculty. Majority of the student information for the system will come from the University's Student Information Database (SIDB) while majority of the course information will come from the University's Course Information Database (CIDB). The external interfaces will include a course screen, registration screen, student profile screen, and user preferences screen. The OES will be portable and available 24 hours a day.

## **2.2 Product Functions**

For students, the OES will require a login process. It will then fetch course information, student information, and the ability to register and remove themselves from courses. The OES will also generate reporting on courses information and student information.

## **2.3 User Classes and Characteristics**

The users of the OES will include students, faculty including professors, and network administrators. Users are expected to have basic knowledge and experience with web browsers. The graphical user interface (GUI) and detailed system help file will also assist users with the OES and its functionality.

### **2.3.1 Students:**

Students will be most frequent user of the system and will use it to view and register for courses. They are not expected to have a high-level of technical expertise, so a simple and easy to navigate interface is important.

### **2.3.2 Faculty:**

Administrative personnel will load and update courses based on semester schedules. Instructors will use the system to view student and class size information for courses they are responsible for. The faculty is not expected to have a high-level of technical expertise, so a simple easy to navigate interface is important.

### **2.3.3 Database Administrator (DBA):**

The database administrators will maintain and manage the student and course databases (SIDB and CIDB). They should have an appropriate college degree, at least 7 years of experience, and an advanced understanding of database management and design.

### **2.3.4 Network and System Administrator:**

The network and system administrators will maintain and manage the system and implement policies and procedures to support the concepts of confidentiality, integrity, availability, nonrepudiation, and authentication (CIANA) . They should have an appropriate college degree, at least 10 years of experience, and an advanced understanding of systems, security policies, incident management, and recovery strategies.

## **2.4 Operating Environment**

The OES will have cross-platform functionality and work on all major browsers. In addition, it will be mobile friendly for student access.

## **2.5 Design and Implementation Constraints**

The OES is dependent on information from the SIDB and CIDB. This means that all users must ensure that their computers are connected to the University's network. Students will only be allowed to view their information and not the information of other students. This system only addresses the need to register for the course online but does not include the ability to attend the course online. If that is a desired feature of the system, it will need to be addressed in a future version. The system will be available in English, French, and Spanish. Additional languages will need to be addressed in future versions. The system will also follow regulatory requirements and include ADA compliance.

## **2.6 User Documentation**

The OES will include an online user manual and detailed help file. This will assist with information on how to create a profile, search for courses, and register for courses. It will also include a chat feature for any additional support needs.

## **2.7 Assumptions and Dependencies**

Since the OES is available at all times it is assumed that it will have access to the SIDB and CIDB in real time. It is also assumed that the system will be able to distinguish the differences between a student profile and a faculty profile, as each will have different characteristics.

### 3. Software Requirements Specifications

#### 3.1 Students can create an account and profile in the system

Students can access the OES and create an account and profile in the system. This account will be used to access the University's network and register for courses during each semester and view courses that the student is currently registered for.

3.1.1 **Priority:** This requirement is a high priority.

##### 3.1.2 Flow of Events:

*3.1.2.1 Student clicks on 'Student Login' on university main page.*

*3.1.2.2 Student clicks on 'Create Profile' on login page.*

*3.1.2.3 Student selects the create new profile option.*

*3.1.2.4 Student enters data to create new profile. Data will include first name, last name, address, phone, email, last 4 of SSN, and password.*

*3.1.2.5 Student clicks on 'Submit' button.*

*3.1.2.6 System confirms that new profile has been created and assigns student ID.*

##### 3.1.3 Functional Requirements:

*3.1.3.1 University main page on website should include a student login icon at the top.*

*3.1.3.2 System should display registration form when student clicks on the 'Create Profile' button.*

*3.1.3.3 System should authenticate user information.*

*3.1.3.4 System should include success or error message to student.*

##### 3.1.4 Non-Functional Requirements:

*3.1.4.1 System should authenticate student through email or text message.*

*3.1.4.2 System should send confirmation email to student within 10 minutes of profile creation.*

*3.1.4.3 Creation of profile should be supported on desktop and IOS/Android.*

### 3.2 Student IDs should be unique and associated with a password

Upon creation of a profile the system should assign an ID to the student. Student IDs should be unique and will be a combination of first and last name initials and randomly assigned number. A password must be present for the system to assign a student ID.

3.2.1 **Priority:** This requirement is a high priority.

#### 3.2.2 Functional Requirements:

*3.2.2.1 System should assign and display student ID with the profile creation success message.*

#### 3.2.3 Non-Functional Requirements:

*3.2.3.1 System should create student ID with first and last initials and random number based on database key creation.*

*3.2.3.2 System should ensure that generated ID is unique and does not match any current ID in the SIDB.*

*3.2.3.3 System should perform checks to ensure that password is present prior to assigning student ID.*

### 3.3 Profiles must include required information

Students will be required to provide a set of information in order to create a profile. The system must validate that the information has been provided and meets the expected requirements. The required fields include first name, last name, address, phone, email, last 4 of SSN, and password.

3.3.1 **Priority:** This requirement is a high priority.

#### 3.3.2 Functional Requirements:

*3.3.2.1 System should provide fields for students to complete for required information.*

*3.3.2.2 System should display an error message if a required field is not completed, and the student tries to create the profile.*

### 3.3.3 Non-Functional Requirements:

*3.3.3.1 System should validate that all required fields have been completed and produce error message if data field is incomplete within 3 seconds of student clicking the submit button.*

## 3.4 Students can log into system with ID and password

If a student has created a profile, they should have the ability to log into the registration system with their assigned student ID and password.

**3.4.1 Priority:** This requirement is a high priority.

### 3.4.2 Flow of Events:

*3.4.2.1 Student clicks on 'Student Login' on university main page.*

*3.4.2.2 Student enters student ID and password on login page.*

*3.4.2.3 Student is directed to registration system main page.*

### 3.4.3 Functional Requirements:

*3.4.3.1 University main page on website should include a student login icon at the top.*

*3.4.3.2 System should display fields for student ID and password and login button.*

*3.4.3.3 System should include a forgot password link to reset password if forgotten by student.*

*3.4.3.4 System should include success or error message to student login attempt.*

### 3.4.4 Non-Functional Requirements:

*3.4.4.1 System should authenticate student based on information in the SIDB in under 5 seconds of hitting submit button.*

*3.4.4.2 System login form should be supported on desktop and IOS/Android.*

*3.4.4.3 System should validate password submitted by user.*

*3.4.4.4 System should perform multi-factor authentication if student forgets password and needs to reset.*

*3.4.4.5 System should update SIDB with new password if reset by student.*



### 3.5 Students can view courses by semester

The university offers courses three semesters per year (spring/summer/fall) and students should be able to view courses available by semester.

3.5.1 **Priority:** This requirement is a medium priority.

#### 3.5.2 Flow of Events:

*3.5.2.1 Student logs in to registration system.*

*3.5.2.2 Student clicks on link to view courses.*

*3.5.2.3 Student clicks on semester drop down field and selects a semester.*

*3.5.2.4 System displays list of courses available during chosen semester.*

#### 3.5.3 Functional Requirements:

*3.5.3.1 Registration system main page should include a link to the course catalog.*

*3.5.3.2 System should provide a drop down field with each semester as an option.*

*3.5.3.3 System should display all courses by semester in a tabular format.*

*3.5.3.4 Course table should provide basic information to student. Data to include course name, course ID, brief description, number of available seats, remaining seats available, and number of students on waitlist.*

*3.5.3.5 Student should be able to click on course to view more detailed information on course and availability.*

#### 3.5.4 Non-Functional Requirements:

*3.5.4.1 System should display available courses by selected semester based on information in the CIDB.*

*3.5.4.2 System should display list of courses within 5 seconds of choosing the semester from the drop down.*

*3.5.4.3 System should display number of enrolled students and waitlisted students based on Student Course Information Database (SCIDB).*

### 3.6 Students can register for selected courses

When viewing courses, students can register on the main course catalog page or on the more detailed course page. Course registration by a student creates an entry in the SCIDB.

3.6.1 **Priority:** This requirement is a high priority.

#### 3.6.2 Flow of Events:

*3.6.2.1 Student views course catalog.*

*3.6.2.2 Student clicks on the 'Register for Course' button.*

*3.6.2.3 System displays successful registration message.*

#### 3.6.3 Functional Requirements:

*3.6.3.1 Course catalog should include a register icon button on the main page. Registration button should also be available on the course detail page.*

*3.6.3.2 System should provide a registration successful message once student registers for course.*

*3.6.3.3 System should provide a list of courses student is registered for on profile page.*

#### 3.6.4 Non-Functional Requirements:

*3.6.4.1 System should validate student registration based on availability in the SCIDB and update course availability totals.*

*3.6.4.2 System should grey out registration button if course is full.*

*3.6.4.3 System should send email to student with registration and course information within 10 minutes of registering for course.*

### 3.7 Courses should have a maximum student count

Courses at the university have a maximum seat count. Students should only be allowed to register for a course if space is available. If course is full then students will need to be added to a waitlist queue. All courses in the system should have a maximum seat count assigned to them by faculty.

3.7.1 **Priority:** This requirement is a high priority.

### 3.7.2 Non-Functional Requirements:

*3.7.2.1 Courses in the CIDB should have a designated maximum student count associated with them.*

*3.7.2.2 System should calculate number of registered users in SCIDB and compare against maximum student count.*

*3.7.2.3 System should disallow registration to course once maximum student count has been met.*

*3.7.2.4 System should open up waitlist once maximum student count has been met. Waitlisted students should be added in a first in first out order.*

## 3.8 Students can add themselves to waitlist if course is full

When a course is full students should be able to add themselves to a waitlist for the course. If the course becomes available, the system should notify students based on first in first out methodology that they can now enroll in the course.

**3.8.1 Priority:** This requirement is a medium priority.

### 3.8.2 Flow of Events:

*3.8.2.1 Student views course catalog.*

*3.8.2.2 Student notices that course is full and clicks on 'Add to Waitlist' button.*

*3.8.2.3 System displays successful waitlist message with queue number.*

### 3.8.3 Functional Requirements:

*3.8.3.1 Course catalog should include waitlist button when course is full.*

*3.8.3.2 System should provide a waitlist addition successful message once student is added to the waitlist for course.*

### 3.8.4 Non-Functional Requirements:

*3.8.4.1 System should add user to bottom of waitlist queue and inform student their number in the queue.*

*3.8.4.2 System should send email to student with waitlist and course information within 10 minutes of being added to waitlist for course.*

*3.8.4.3 System should inform first in line student in waitlist that course is available for registration if a student drops or cancels course.*

*3.8.4.4 If first in line waitlist student does not register for available course within 48 hours the system should remove them waitlist queue and send email to next student in line in waitlist. This pattern should continue until a waitlist student has been added to the course or no more students exist in the waitlist.*

## 3.9 Students can cancel enrollment in courses

Students should have the option to cancel enrollment in a course or remove themselves from a course waitlist. If a student cancels their enrollment in a course that has a waitlist the first person in the waitlist should be invited to register for the course.

**3.9.1 Priority:** This requirement is a high priority.

### 3.9.2 Flow of Events:

*3.9.2.1 Student views their list of registered and waitlisted courses.*

*3.9.2.2 Student clicks on 'Remove from Course' button.*

*3.9.2.3 System displays successful removal message.*

### 3.9.3 Functional Requirements:

*3.9.3.1 Student course list should include removal button to pull themselves from courses or waitlists.*

*3.9.3.2 System should provide a course or waitlist removal successful message once student is removed from course or waitlist.*

### 3.9.4 Non-Functional Requirements:

*3.9.4.1 Student course list should update within 5 minutes of them cancelling their course registration.*

*3.9.4.2 System should recalculate course capacity when student removes themselves from course.*

*3.9.4.3 System should send email to student with cancellation verification information.*

*3.9.4.4 System should inform first in line student in waitlist that course is available for registration if a student drops or cancels course.*

## 4. UML Models and Testing Levels

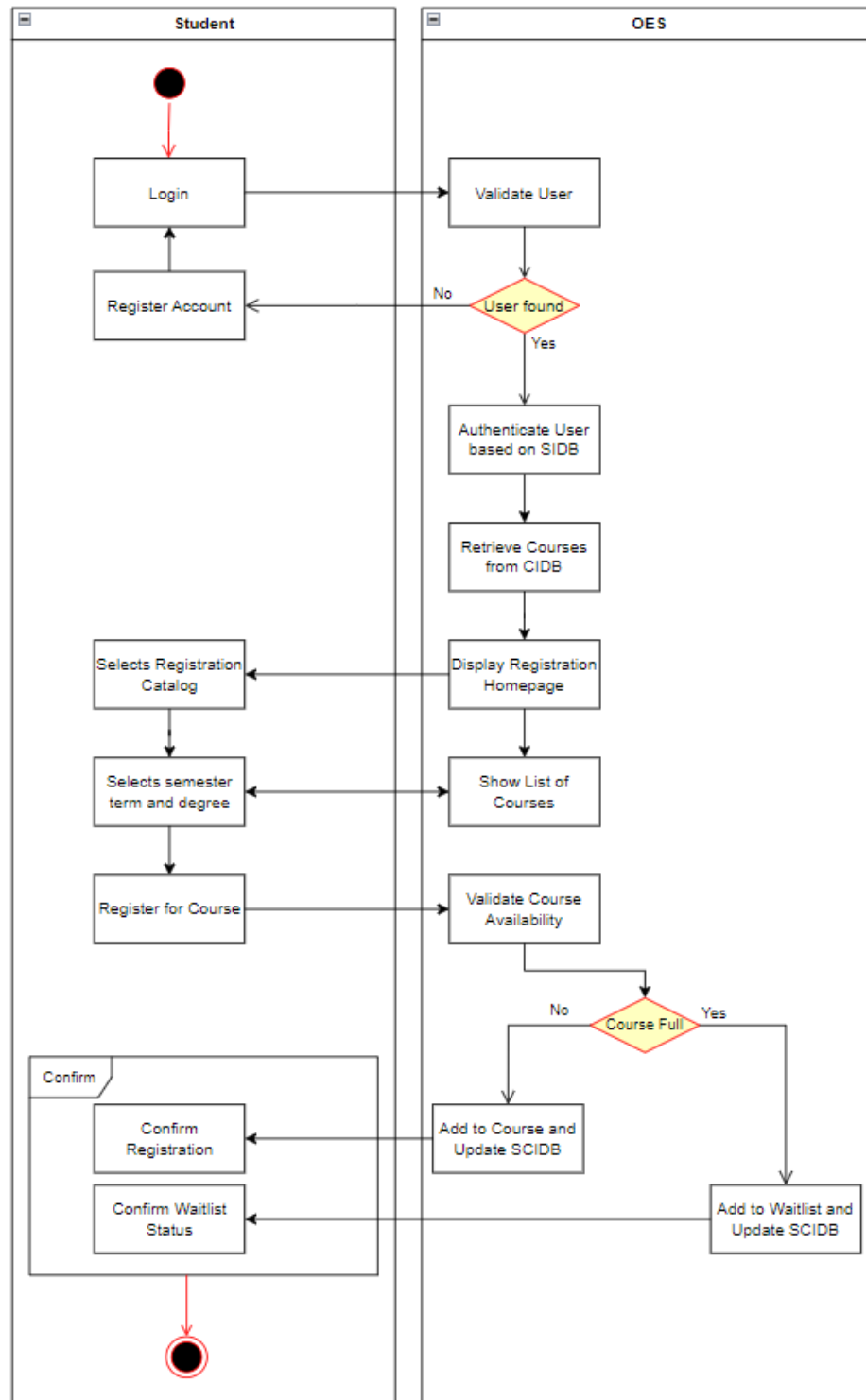
### 4.1 Diagrams

System models provide information and different perspectives of the system. These perspectives will help the team better understand the system's functionality and create consistent communication between the team and end users.

#### 4.1.1 Activity Diagram

Context models provide environmental and system boundary information. System boundaries lay out what is and is not included in the system. They indicate when other systems may be used or where dependencies may occur with the system. The role of context models is to illustrate all systems within the environment. The context perspective is represented with the UML activity diagram. An activity diagram lays out which systems are used in business processes and the activities involved in those processes.

An activity diagram will be used to lay out the flow of activities with the university's OES. Figure 1 shows the systems and flow of activities from the student's initial login to their course registration.

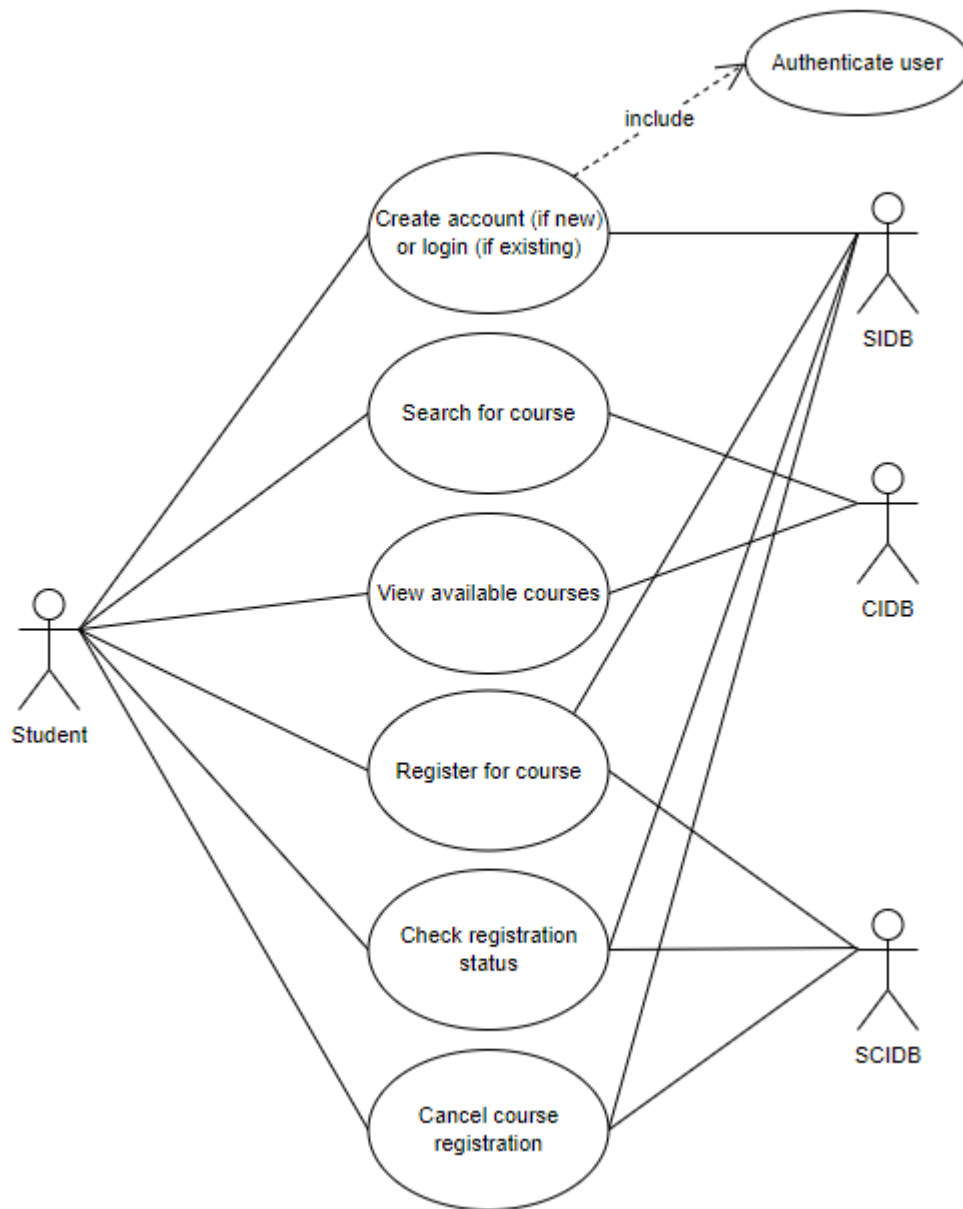
**Figure 1** – Activity Diagram

*Note.* Activity diagram of student login and course registration in OES depicts system workflows and actions.

#### **4.1.2 Use Case Diagram**

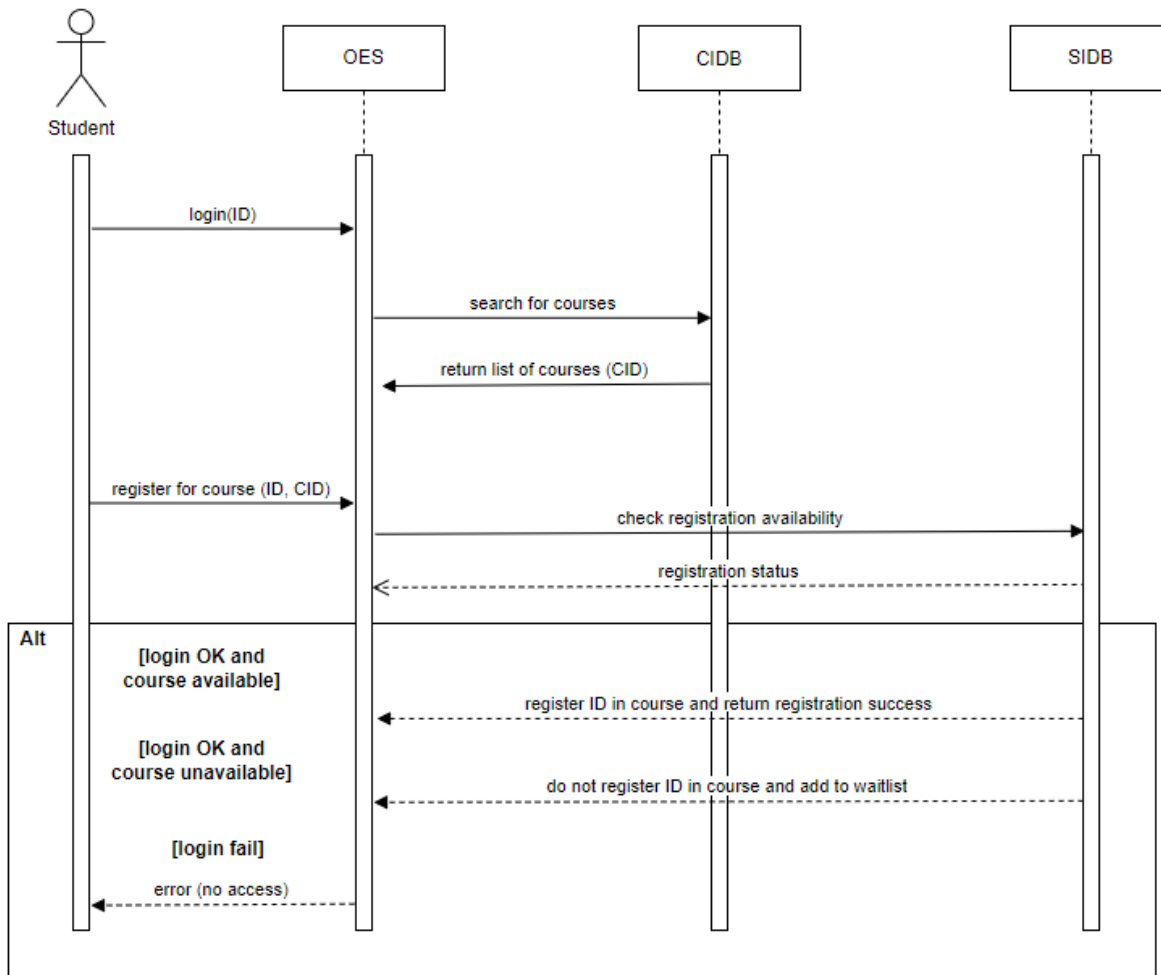
Interaction models represent the touchpoints with the user, between systems, and with components. User interaction generally involves inputs and outputs, and an interaction model helps highlight user requirements. Interaction between systems helps to identify any possible communication issues between the system being developed and others in the environment. Component interaction represents the system's performance and dependability requirements. Interaction modeling is generally presented through use case modeling for interactions between systems and users or sequence diagrams for component interactions.

Use case modeling not only maps out the interactions between the users and the system but also the expected behavior of the system within these interactions. A use case diagram for the university's OES can be seen in figure 2. Sequence diagrams lay out the interactions between the users and system components. A sequence diagram for the university's OES can be seen in figure 3.

**Figure 2** – Use Case Diagram

*Note.* Use case diagram depicts the system's users and their interactions with the university's OES



**Figure 3** – Sequence Diagram

*Note.* Sequence diagram of course search and registration in the university’s OES depicts system interactions and messages between objects.

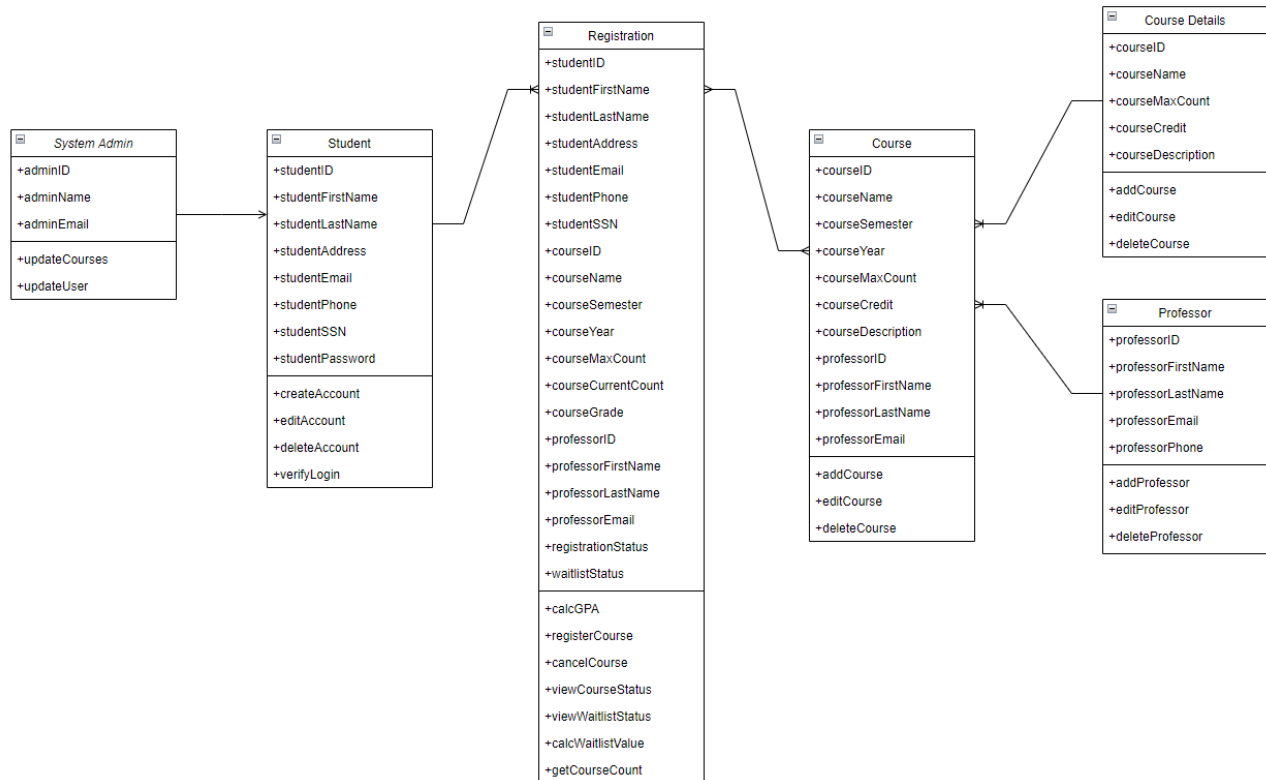
#### 4.1.3 Class Diagram

Structural models show the organization and relationships of system components. “Structural models may be static models, which show the organization of the system design, or dynamic models, which show the organization of the system when it is executing” (Sommerville, 2016). These views are used to show the overall system architecture. Class diagrams are used for

static structures to show the system classes, associations between them, attributes, and operations.

Figure 4 shows the OES's system relationships.

**Figure 4 – Class Diagram**



*Note.* The class diagram depicts the classes and associated attributes for registration in the university's OES

## 4.2 Testing Introduction

Testing is an integral part of the development and programming activities in a project, as it allows the team to find bugs in the software and demonstrate that the software meets requirements. While validation and verification may be lumped together, they each serve different purposes. Validation is centered around meeting the customer's needs while verification is ensuring that the software meets its function and non-functional requirements. One model used to test throughout the software life cycle is the general V model. The left side of the V corresponds

with the software development life cycle (SDLC) and the right side of the V represents the software testing life cycle (STLC). The V model follows a waterfall model, and each phase starts after the previous one is complete. The V model starts on the left development side and contains requirements definition, functional system design, technical system design, component specification, and programming activities. Each of these activities is tied to a testing process that is in place to validate and verify the outcome. Once programming is complete the testing phases begin with component or unit testing, integration testing, system testing, and finally acceptance testing.

### **4.3 Component Testing**

Component testing focuses on the testing of component interfaces. This test phase is the validation of single software components at the code level. This is the lowest test level and generally the longest test phase. That being said component testing should eliminate the majority of issues and bugs.

The university's OES will have functional components that need to be tested including suggested course population, dynamic filtering, search results display, displaying course details, and confirming registration statuses. There are also system classes that will need to be tested including student, course, course details, web admin, registration, and professor.

### **4.4 Integration Testing**

Integration is the connection of components into larger structural units and subsystems. These are testing to ensure that all of the components work together as expected. "Thus, the goal of the integration test is to expose faults in the interfaces and in the interaction between integrated components" (*Spillner, Linz, & Schaefer, 2014*).

After completing component testing it will be important to begin testing how each of those components work together. For example, testing the course and registration classes when a

student registers for a course. Does the data flow successfully throughout the interface? Do the components work successfully with each other?

## **4.5 System Testing**

System testing focuses on the testing of component interactions. This involves testing system functionality, performance, and communication with external systems. This phase will uncover potential hardware and software compatibility issues from the functional system design phase. This phase is also used to validate user documentation such as system manuals or user manuals.

The team will need to test the system to ensure that it not only performs quickly but that it is also reliable, secure, and maintainable. The team will need to perform in a test environment that very closely mimics production and view each interaction and functionality from the user's perspective.

## **4.6 Acceptance Testing**

"Lastly, acceptance testing is the process of implementing all tests created during the initial requirements phase and should ensure that the system is functional in a live environment with actual data, ready for deployment" (*Airbrake, 2016*). This phase is all about validating that the system meets the requirements laid out in the software requirements phase. It involves testing in a user environment from the user's perspective but also focusing on non-functional requirements, like performance, to ensure it all works as expected.

In this phase, the user's will finally be able to test in a production environment to find any outstanding bugs or issues. The team will need to determine if the look, feel, and functionality meet all of the requirements. It will be important to run tests with external systems, such as databases. The team will also need to make sure that they are able to run all necessary reporting. Full testing from component to acceptance is absolutely critical to ensure a successful end product.

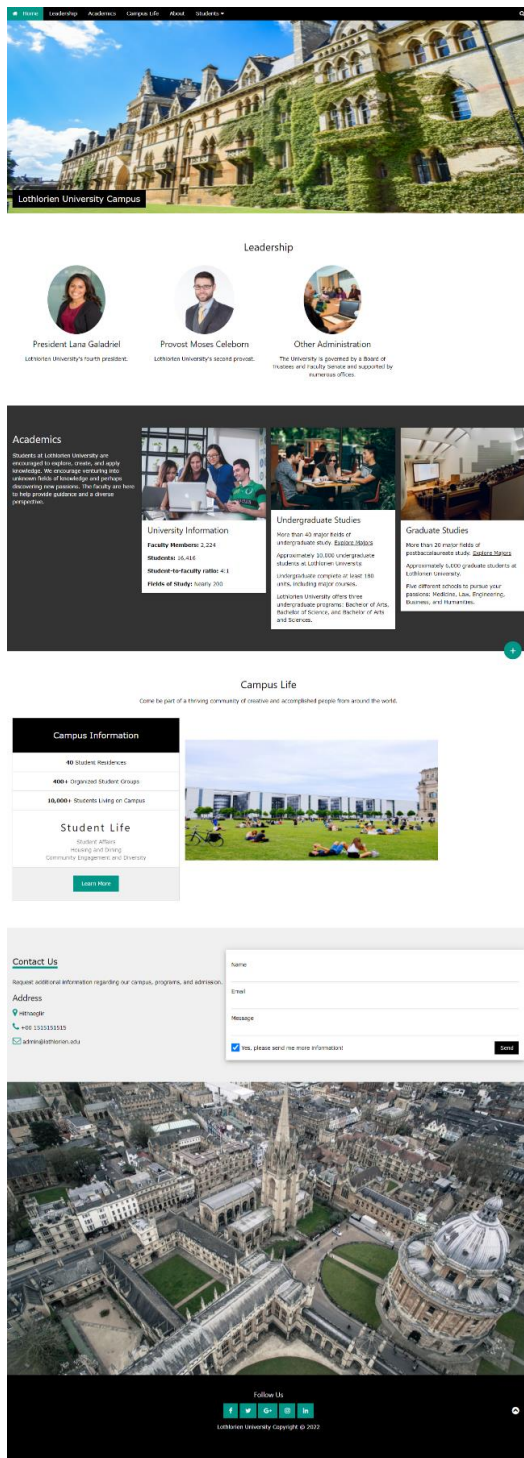
## Appendix A: Glossary

Term	Definition
Student	Person viewing and registering for courses at university.
Faculty	University personnel responsible for entering course information and managing student relations.
Course	Class subject to be taught by university and found in course catalog.
SRS	Software requirements specification
OES	Online enrollment system
SIDB	Student information database
CIDB	Course information database
SCIDB	Student course information database
ID	Identification

# Appendix B: System Figures

## Figure B1

### University Home Page



The screenshot displays the Lothlorien University Home Page. At the top, a navigation bar includes links for Home, Undergraduate, Academic, Campus Life, About, and Students. The main header features a large image of the Lothlorien University Campus with the text "Lothlorien University Campus" overlaid.

Below the header, the "Leadership" section introduces three key figures: President Lana Galadriel (Lothlorien University's fourth president), Provost Moses Celeborn (Lothlorien University's second provost), and Other Administration (The University is governed by a Board of Trustees and Faculty Senate and supported by numerous offices).

The "Academics" section highlights the university's commitment to excellence in teaching, research, and scholarship. It includes three sub-sections:
 

- University Information:** Faculty Members: 2,224; Students: 10,000; Student-to-Faculty ratio: 4:1; Fields of Study: nearly 200.
- Undergraduate Studies:** More than 45 major fields of study in both Arts, Business, and Science; Approximately 10,000 undergraduate students at Lothlorien University; Undergraduate programs at least 180 years, including many classics.
- Graduate Studies:** More than 20 major fields of study in both Arts, Business, and Science; Approximately 6,000 graduate students at Lothlorien University; Five different schools to pursue your passion: Medicine, Law, Engineering, Business, and Humanities.

The "Campus Life" section invites visitors to join a thriving community of creative and accomplished people from around the world. It includes a "Campus Information" sidebar with statistics: 40 Student Residences, 400+ Organized Student Groups, and 10,000+ Students Living on Campus. The main content area features a large image of a student lounge and a "Student Life" section with links to Student Affairs, Housing and Dining, and Community Engagement and Diversity. A "Learn More" button is also present.

The "Contact Us" section provides a form for requesting additional information regarding campus, programs, and admission. The form includes fields for Name, Email, and Message, and a checkbox for "Yes, please send me more information." A "Send" button is located at the bottom right of the form.

At the bottom of the page, there is a large aerial image of the university campus. Below this image, a "Follow Us" section includes social media icons for Facebook, Twitter, YouTube, and LinkedIn. The footer text reads "Lothlorien University Copyright © 2022".

## Figure B2

### Student Portal Registration Page

Home Leadership Academics Campus Life About Students

Lothlorien University Campus

## Student Registration

Please enter your information to register for access to the Student portal.

First Name

Last Name

Address

Phone: 916-123-4567

Email Address

Last 4 SSN

Password

Register Reset

Already have access to the student portal? [Login here](#)

Lothlorien University Copyright © 2022

## Figure B2

### Student Portal Registration Page



## Figure B3

### Login Page

Home Leadership Academics Campus Life About Students

Lothlorien University Campus

### Student Portal Login

Please enter your school email and password to access the Lothlorien University Student Portal.

Email Address

Password

Login Reset


Don't have an account? [Register here](#)

Lothlorien University Copyright @ 2022



**Figure B4***Profile Page*

[Home](#) [Leadership](#) [Academics](#) [Campus Life](#) [Contact Us](#) [Students](#) [Q](#)

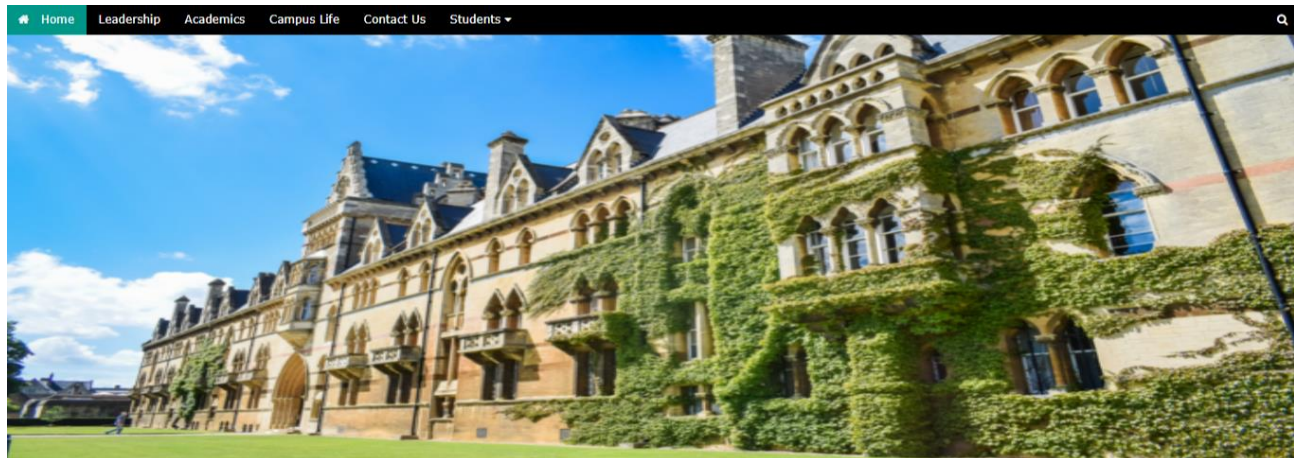


Hello Frodo. Here is the current information we have on record for you.

### Profile Information

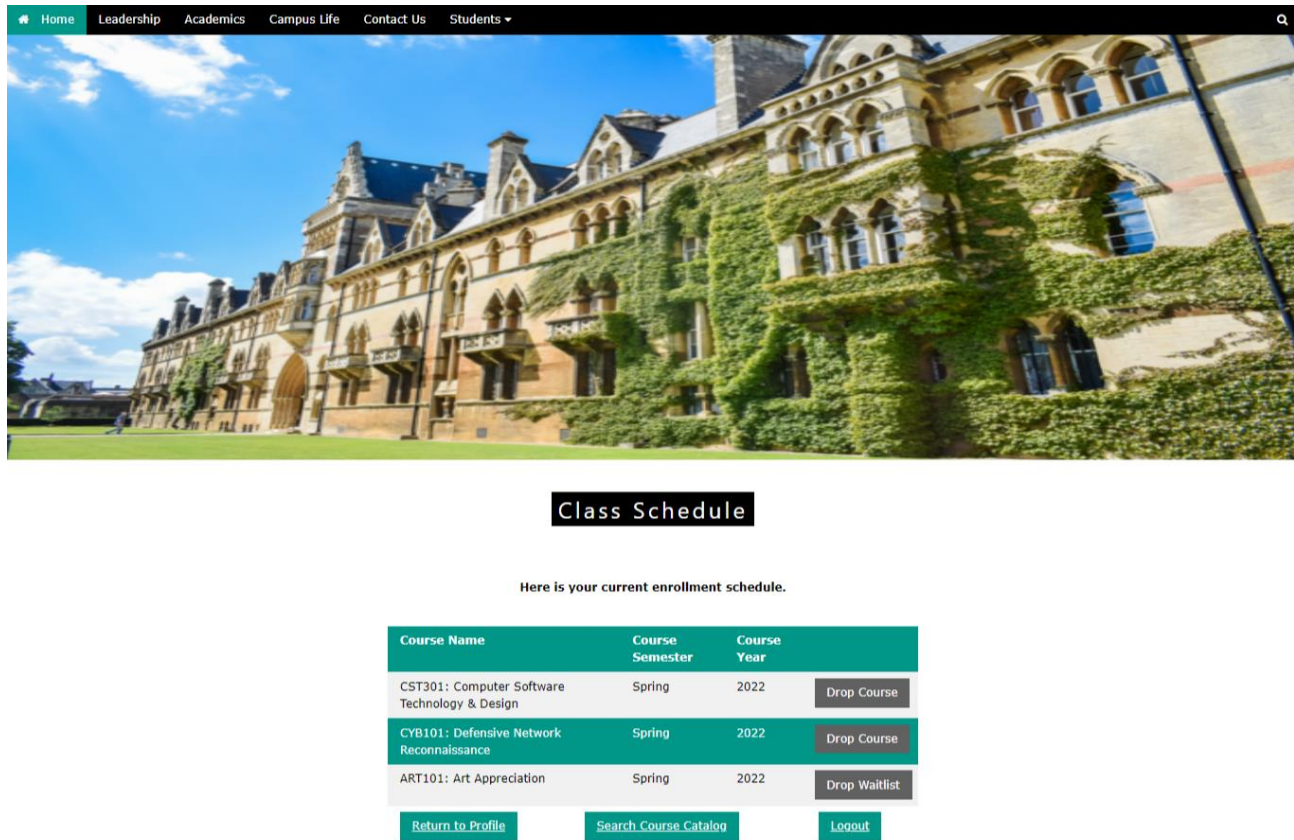
First Name	Last Name	Email	Address	Street Name	City	State	Zip	Phone
Frodo	Baggins	fbaggins@lothlorien.edu	1	Bagshot Row	Hobbiton	NZ	138933	202-869-4568

[View Enrolled Courses](#) [Search Course Catalog](#) [Logout](#)

**Figure B5***View Class Schedule***Class Schedule**

Here is your current enrollment schedule.

Course Name	Course Semester	Course Year	
CST301: Computer Software Technology & Design	Spring	2022	Drop Course
CYB101: Defensive Network Reconnaissance	Spring	2022	Drop Course
ART101: Art Appreciation	Spring	2022	Drop Waitlist
<a href="#">Return to Profile</a>	<a href="#">Search Course Catalog</a>	<a href="#">Logout</a>	

**Figure B6***View Course Catalog*

Home Leadership Academics Campus Life Contact Us Students

**Class Schedule**

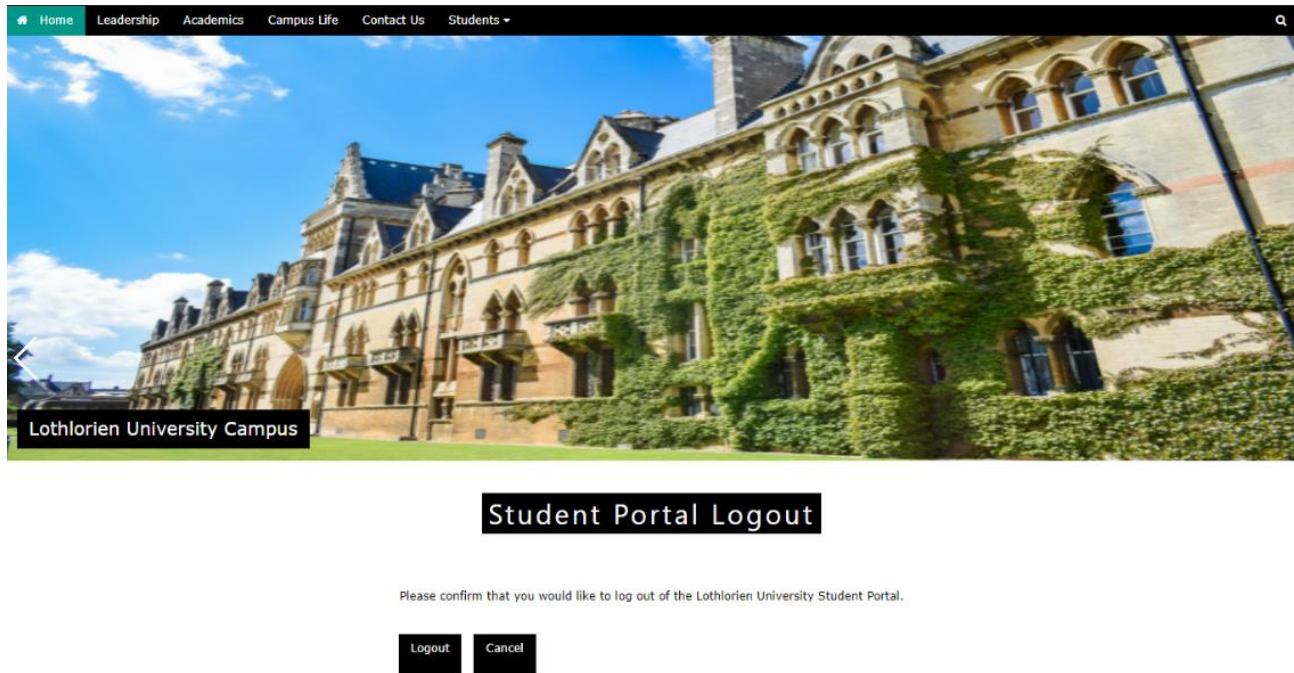
Here is your current enrollment schedule.

Course Name	Course Semester	Course Year	
CST301: Computer Software Technology & Design	Spring	2022	Drop Course
CYB101: Defensive Network Reconnaissance	Spring	2022	Drop Course
ART101: Art Appreciation	Spring	2022	Drop Waitlist

[Return to Profile](#) [Search Course Catalog](#) [Logout](#)

**Figure B7**

*Logout Page*



## Appendix C: To Be Determined List (Pending)

## Appendix D: References

Airbrake. (2016, December 26). V-Model: What Is It and How Do You Use It? Retrieved from

Airbrake: <https://airbrake.io/blog/sdlc/v-model>

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

Spillner, A., Linz, T., & Schaefer, H. (2014). Software Testing Foundations: A study guide for the certified tester exam (4th ed.). Santa Barbara: Rocky Nook Inc.