Education Portal

CST438-40_SP24 - #3

Christopher Varela, Nathan Simpson, Oscar Ochoa, Severin Light

Software Requirements Specification Document

Version(1.0) **Date: 04/02/2024**

Table of Contents

Section 1.0 Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Service Oriented Architecture. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2. Scope of Project

This software system will be a Web-Based Platform for both students and instructors of a school system. This system is designed to assist instructors and students and provide an easy to navigate online resource for checking and maintaining school records.

For instructors, this system is designed to allow the creation, removal, and maintaining of assignments and grades for their classes. For students, the system allows them to view transcripts, grades, courses, and assignments as well as the ability to drop and add courses. The software will allow both students and instructors to easily access the necessary information for their specific academic needs. The system contains a relational database containing information for users, enrollments, grades, assignments, sections, courses, and terms.

1.3. Glossary

Term	Definition
Database	Collection of all the information monitored by the system.
Field	A cell within a form.
Table	A collection of fields in a database used to track specific information.
User Table	A table used to track the various users, both student and instructor.
Instructor User	Person responsible for administering and maintaining grades and assignments.
Student User	Person who can view transcripts, grades, courses, and assignments, as well as add/drop courses.
Enrollment Table	A table used to track a Student User's academic status for a specific class.
Course Table	A table used to track an academic class. (Unique)
Section Table	A table used to track various information for a specific course, such as the meeting location, times, etc. (Unique although 2 Sections can have the same Course)
Term Table	A table used to track a specific semester and year, along with the add, drop, start, and end dates/deadlines. (Unique)
Assignment Table	A table used to track the title, due date, and section of a specific assignment.
Grade Table	A table used to track the score of an assignment and to link the assignment to a specific enrollment.
Stakeholder	Any person with an interest in the project who is not a developer.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.

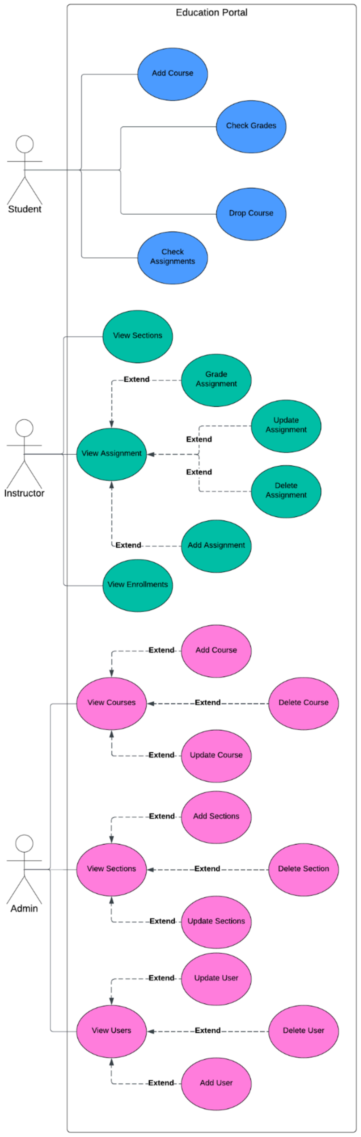
1.4. Overview of Document

In the System Overview section, we have provided UML diagrams for each user and their respective use case. It outlines the non-formal prerequisites and serves as a backdrop for the technical requirements specification outlined in the subsequent chapter. The Functional Requirements section is tailored predominantly for developers, delineating the technical intricacies of the product's functionality in precise terms.

Both sections of the document provide a comprehensive description of the software product, albeit targeting distinct audiences and employing varying linguistic approaches.

Section 2.1 System Overview

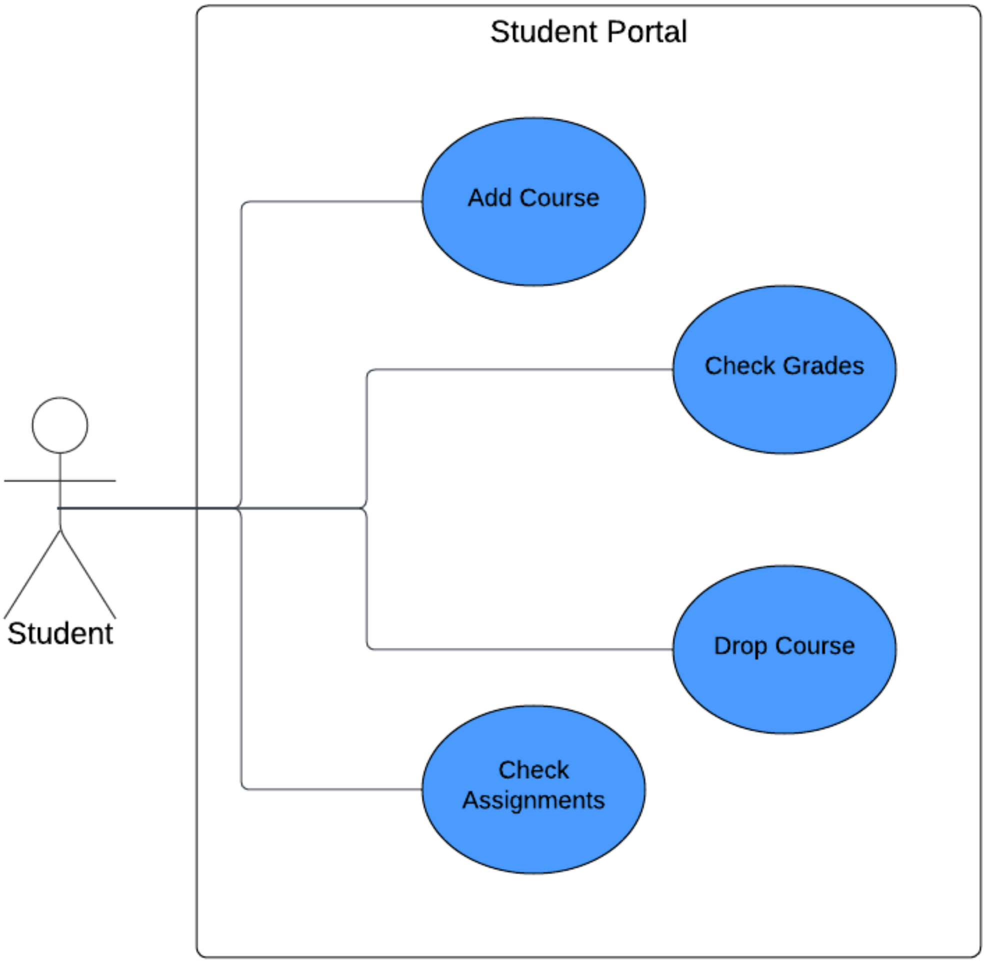
2.1.1 Full Educational Portal UML Use Case Diagram



The above outlines the use cases based on the type of user: student, instructor, or administrator. Each of these user types will be further elaborated upon in their respective categories below.

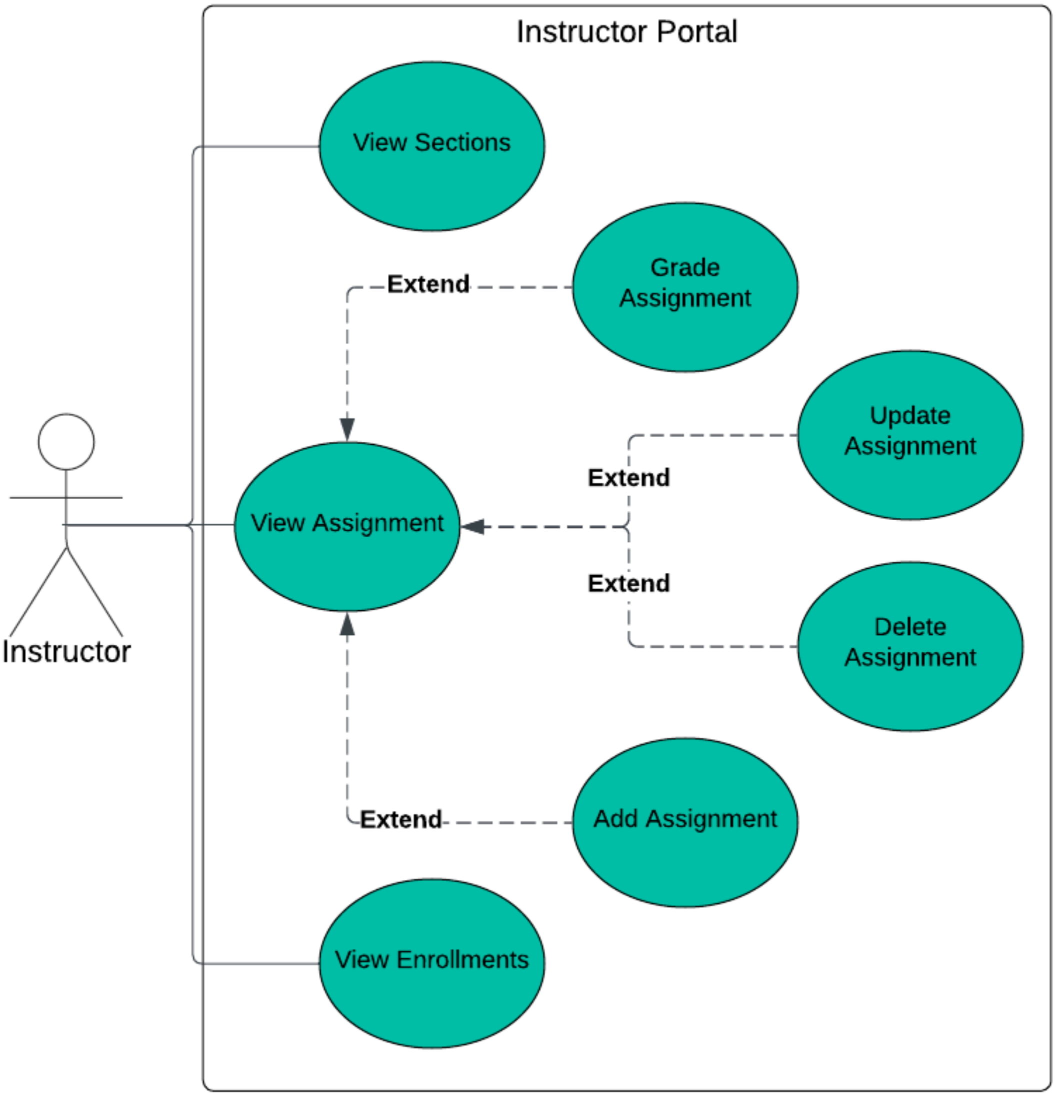
2.1.2 Student

If the user is a student they shall have access to view their grades/transcript. They shall also have the ability to add or drop courses as long as it falls within the drop deadline. The student shall also be able to check their assignments and view if they have been graded.



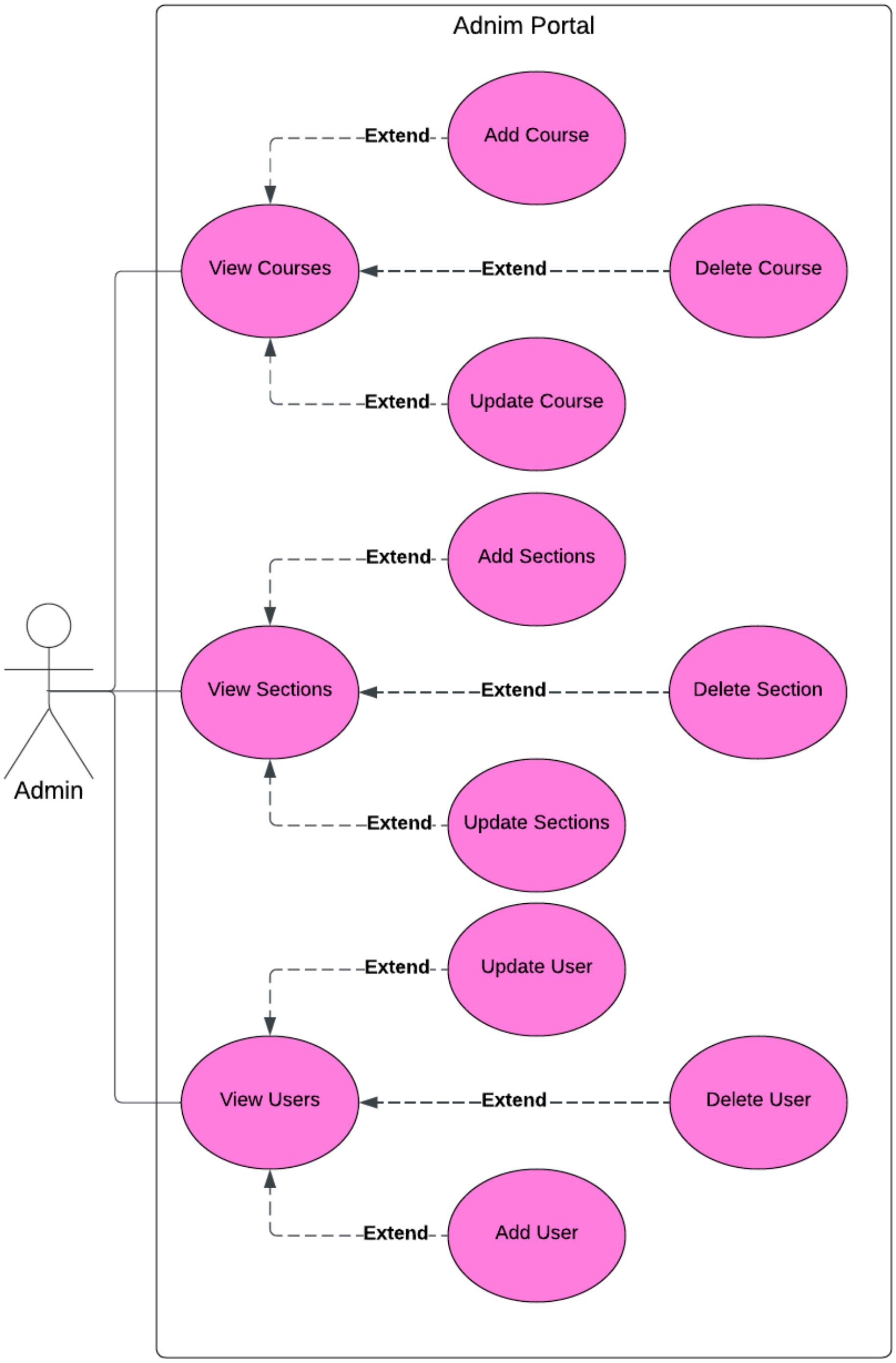
2.1.3 Instructor

The instructors shall be able to view the sections, assignments, and enrollments as long as they are the instructors who are in charge of such. They shall be able to grade, update, delete, and assign assignments from the portal.



2.1.4 Admin

If the user is an administrator they shall have the ability to view courses, sections, and users (ie students, instructors). Admin shall also be able to add, delete, and update courses as well as sections, and users.



Section 3.2 Functional Requirements

3.2.1 Admin: Manage User

Use Case Name	Manage User
XRef	Section 2.1.3 Admin
Trigger	The Admin accesses the Users page
Precondition	The user is logged in as an Administrator
Basic Path	1. The Admin navigates to the Users page. 2. The Admin selects the Edit button for a specified user, opening an edit window with the user’s details. 3. The Admin changes the desired details of the user. 4. The Admin selects the Save button, saving the entry with the changes, closing the edit window and displaying the list of users once again. 5. A response message displays if the save was successful or if there was an error.
Alternative Paths	In step 1, if the Admin wishes to remove a user. 2. The Admin selects the Delete button for a specified user, a dialog box appears asking the Admin to confirm the deletion. 3. The Admin selects the Yes button closing the edit window and displaying the list of users once again. 4. A response message displays if the save was successful or if there was an error.
Postcondition	The user has their details edited or is deleted.
Exception Paths	The Admin may quit anytime before selecting the Save or Delete buttons.
Other	None.

3.2.2 Admin: Add User

Use Case Name	Add User
XRef	Section 2.1.3 Admin
Trigger	The Admin accesses the Users page
Precondition	The user is logged in as an Administrator
Basic Path	1. The Admin navigates to the Users page. 2. The Admin selects the Add User button, opening a window with empty fields for the user information. 3. The Admin adds the desired details of the user. 4. The Admin selects the Save button, saving the entry, closing the edit window, and displaying the list of users once again. 5. A response message displays if the save was successful or if there was an error.
Alternative Paths	None
Postcondition	The user has been added to the database.
Exception Paths	The Admin may quit anytime before selecting the Save button.
Other	None.

3.2.3 Admin: Manage Course

Use Case Name	Manage Course
XRef	Section 2.1.3 Admin
Trigger	The Admin accesses the Courses page
Precondition	The user is logged in as an Administrator
Basic Path	1. The Admin navigates to the Courses page. 2. The Admin selects the Edit button for a specified course, opening an edit window with the course's details. 3. The Admin changes the desired details of the course. 4. The Admin selects the Save button, saving the entry with the changes, closing the edit window and displaying the list of courses once again. 5. A response message displays if the save was successful or if there was an error.
Alternative Paths	In step 1, if the Admin wishes to remove a course. 5. The Admin selects the Delete button for a specified course, a dialog box appears asking the Admin to confirm the deletion. 6. The Admin selects the Yes button closing the edit window and displaying the list of courses once again. 7. A response message displays if the save was successful or if there was an error.
Postcondition	The course has its details edited or is deleted.
Exception Paths	The Admin may quit anytime before selecting the Save or Delete buttons.
Other	None.

3.2.4 Admin: Add Course

Use Case Name	Add Course
XRef	Section 2.1.3 Admin
Trigger	The Admin accesses the Courses page
Precondition	The user is logged in as an Administrator
Basic Path	1. The Admin navigates to the Courses page. 2. The Admin selects the Add Course button, opening a window with empty fields for the course information. 3. The Admin adds the desired details of the course. 4. The Admin selects the Save button, saving the entry, closing the edit window, and displaying the list of courses once again. 5. A response message displays if the save was successful or if there was an error.
Alternative Paths	None
Postcondition	The course has been added to the database.
Exception Paths	The Admin may quit anytime before selecting the Save button.
Other	None.

3.2.5 Admin: Manage Section

Use Case Name	Manage Section
XRef	Section 2.1.3 Admin
Trigger	The Admin accesses the Sections page
Precondition	The user is logged in as an Administrator
Basic Path	1. The Admin navigates to the Sections page. 2. The Admin enters in the Course Prefix, Year, and Semester, then selects the Search for Sections button. 3. The Admin selects the Edit button for a specified section, opening an edit window with the section's details. 4. The Admin changes the desired details of the section. 5. The Admin selects the Save button, saving the entry with the changes. 6. A response message displays if the save was successful or if there was an error. 7. The Admin selects the Close button to once again view the sections.
Alternative Paths	In step 2, if the Admin wishes to remove a section. 3. The Admin selects the Delete button for a specified section, a dialog box appears asking the Admin to confirm the deletion. 4. The Admin selects the Yes button closing the edit window and displaying the list of sections once again. 5. A response message displays if the save was successful or if there was an error.
Postcondition	The section has its details edited or is deleted.
Exception Paths	The Admin may quit anytime before selecting the Save or Delete buttons.
Other	Upon editing a section, the Admin is assigning an instructor to the section by updating the Instructor Email field with the desired instructor's email.

3.2.6 Admin: Add Section

Use Case Name	Add Section
XRef	Section 2.1.3 Admin
Trigger	The Admin accesses the Sections page
Precondition	The user is logged in as an Administrator
Basic Path	1. The Admin navigates to the Sections page. 2. The Admin enters in the Course Prefix, Year, and Semester, then selects the Search for Sections button. 3. The Admin selects the Add Section button, opening a window with empty fields for the section information. 4. The Admin adds the desired details of the section. 5. The Admin selects the Save button, saving the entry, closing the edit window, and displaying the list of sections once again. 6. A response message displays if the save was successful or if there was an error.
Alternative Paths	None
Postcondition	The section has been added to the database.
Exception Paths	The Admin may quit anytime before selecting the Save button.
Other	Upon adding a section, the Admin is assigning an instructor to the section by inputting into the Instructor Email field with the desired instructor's email.

3.2.7 Instructor: Manage Assignment

Use Case Name	Manage Assignment
XRef	Section 2.1.2 Instructor
Trigger	The Instructor accesses the Sections page
Precondition	The user is logged in as an Instructor
Basic Path	1. The Instructor enters the Year and Semester into the search fields and selects the Show Sections button. 2. The Instructor selects the View Assignments button from the desired section listing. 3. The Instructor selects the Edit button on the desired assignment. 4. An edit window appears and the Instructor changes the desired information. 5. The Instructor selects the Save button to commit the changes. 6. A response message displays if the save was successful or if there was an error. 7. The Instructor selects the Close button to once again view the assignments.
Alternative Paths	In step 2, if the Instructor wishes to remove an assignment. 3. The Instructor selects the Delete button for a specified assignment, a dialog box appears asking the Admin to confirm the deletion. 4. The Instructor selects the Yes button closing the edit window and displaying the list of assignments once again. 5. A response message displays if the save was successful or if there was an error.
Postcondition	The assignment has its details edited or is deleted.
Exception Paths	The Instructor may quit anytime before selecting the Save or Delete buttons.
Other	None.

3.2.8 Instructor: Add Assignment

Use Case Name	Add Assignment
XRef	Section 2.1.2 Instructor
Trigger	The Instructor accesses the Sections page
Precondition	The user is logged in as an Instructor
Basic Path	1. The Instructor enters the Year and Semester into the search fields and selects the Show Sections button. 2. The Instructor selects the View Assignments button from the desired section listing. 3. The Instructor selects the Add Assignment button below the listings, opening a window to enter assignment information. 4. The Instructor inputs the desired details for the assignment. 5. The Instructor selects the Save button and a response message is displayed in the window if the assignment was added or if there was an error. 6. The Instructor selects the Close button to once again view the assignment listings.
Alternative Paths	None
Postcondition	The assignment has been added to the database.
Exception Paths	The Instructor may quit anytime before selecting the Save button.
Other	None.

3.2.9 Instructor: Grade Assignment

Use Case Name	Grade Assignment
XRef	Section 2.1.2 Instructor
Trigger	The Instructor accesses the Sections page
Precondition	The user is logged in as an Instructor
Basic Path	1. The Instructor enters the Year and Semester into the search fields and selects the Show Sections button. 2. The Instructor selects the View Assignments button from the desired section listing. 3. The Instructor selects the View Grades button next to the assignment, navigating to the assignment page and showing a list of students enrolled in the section. 4. The Instructor inputs the score for the desired student's assignment. 5. The Instructor selects the Save button and a response message is displayed at the top if the score was saved or if there was an error.
Alternative Paths	None
Postcondition	The assignment has been updated with the score in the database.
Exception Paths	The Instructor may quit anytime before selecting the Save button.
Other	None.

3.2.10 Instructor: Final Grade

Use Case Name	Final Grade
XRef	Section 2.1.2 Instructor
Trigger	The Instructor accesses the Sections page
Precondition	The user is logged in as an Instructor
Basic Path	1. The Instructor enters the Year and Semester into the search fields and selects the Show Sections button. 2. The Instructor selects the View Enrollments button from the desired section listing. 3. The Instructor enters the Final Grade in the desired student's enrollment. 4. The Instructor selects the Save Grade button and a response message is displayed at the top if the final grade was saved or if there was an error.
Alternative Paths	None
Postcondition	The enrollment has been updated with the final grade in the database.
Exception Paths	The Instructor may quit anytime before selecting the Save button.
Other	None.

3.2.11 Student: Course Enroll

Use Case Name	Course Enroll
XRef	Section 2.1.1 Student
Trigger	The Student accesses the add course page
Precondition	The user is logged in as a Student
Basic Path	1. The Student navigates to the add course page. 2. The Student selects a section from the drop-down menu. 3. The Student clicks the “Enroll” button. 4. A response message is displayed notifying the Student that they have been successfully enrolled.
Alternative Paths	None.
Postcondition	The student is enrolled in the section and is added to the database.
Exception Paths	The Student may exit before clicking the enroll button.
Other	None.

3.2.12 Student: Drop a Course

Use Case Name	Drop a Course
XRef	Section 2.1.1 Student
Trigger	The Student accesses the schedule page
Precondition	The user is logged in as a Student
Basic Path	1. The Student navigates to the schedule page 2. The Student enters a valid Year and Semester into the text boxes. 3. The Student clicks the "View Schedule" button. 4. A list of all the Courses the Student is currently enrolled in is displayed including their ID, Section, Year, and Semester. 5. The student may click the "DROP" button to drop a specific course. 6. The course will drop, and the page will refresh showing the Student that they are no longer enrolled.
Alternative Paths	None.
Postcondition	The student is dropped from the section.
Exception Paths	The Student may exit before clicking the drop button
Other	None.

3.2.13 Student: View Assignments and Grades

Use Case Name	View Assignments and Grades
XRef	Section 2.1.1 Student
Trigger	The Student accesses the assignments page
Precondition	The user is logged in as a Student
Basic Path	1. The Student navigates to the assignments page 2. The Student enters a valid Year and Semester into the text boxes. 3. The Student clicks the “Search for Assignments” button. 4. A list of all the student's assignments are displayed including their ID, title, due date, and score.
Alternative Paths	None.
Postcondition	None.
Exception Paths	The Student may exit before clicking the “Search for Assignments” button
Other	None.

3.2.14 Student: View Transcript

Use Case Name	View Transcript
XRef	Section 2.1.1 Student
Trigger	The Student accesses the transcript page
Precondition	The user is logged in as a Student
Basic Path	1. The Student navigates to the transcript page 2. A list of all the student's courses are displayed including their year, semester, course id, section id, credits, and final grade.
Alternative Paths	None.
Postcondition	None.
Exception Paths	None.
Other	None.

Section 3.3 Non-Functional Requirements

This section outlines the non-functional requirements that pertain to the overarching qualities of the Education Portal, rather than specific functionalities. These requirements ensure that the portal operates effectively and securely on a broad scale.

3.3.1 Performance

- **Load Time:** Web pages within the Education Portal shall load within 2 seconds under normal conditions, ensuring swift access to information for all users.
- **Concurrent Users:** The system must support up to 500 concurrent users without performance degradation, allowing multiple students and staff to interact with the portal simultaneously during peak usage times.
- **Database Queries:** Critical database queries, such as retrieving a student's grades or course list, should not exceed 1 second, providing a responsive experience for users.

3.3.2 Security

- **Data Encryption:** All sensitive data, including student records and personal information, will be encrypted using industry-standard protocols both in transit and at rest.
- **Authentication:** Users must authenticate through a secure login process with password complexity requirements to prevent unauthorized access.
- **Authorization:** The system will enforce role-based access controls, ensuring users can only interact with functionalities pertinent to their role (student, instructor, or administrator).

3.3.3 Reliability

- **Uptime:** The Education Portal shall maintain a 99.9% uptime, excluding scheduled maintenance periods, to ensure constant availability for users.
- **Data Integrity:** The system will include mechanisms to prevent data corruption and ensure the integrity of records, with transaction logs kept for a minimum of one year.
- **Backup and Recovery:** Regular backups of the system data will be conducted, with a recovery plan in place to restore services within 4 hours in the event of a critical system failure.

Section 3.4 Logical Database Diagram

This is the logical structure of the data to be stored in the Education Management System

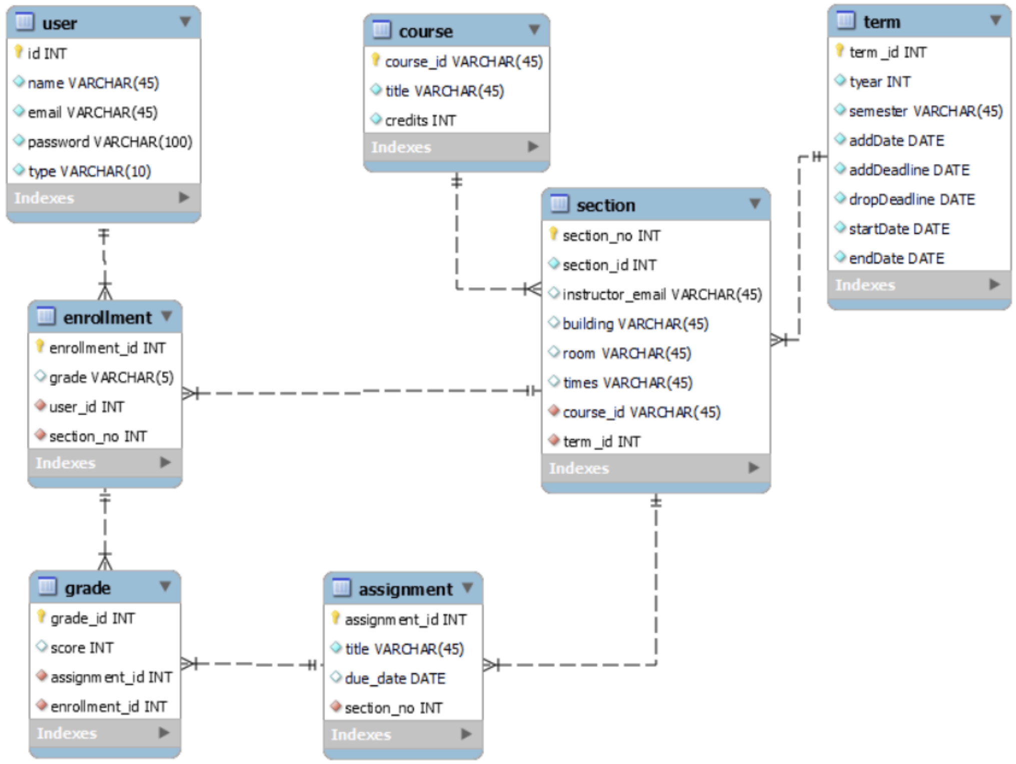


Figure 3.1 - Logical Structure of the Education Management System

The data descriptions of each of the data entities is as follows:

3.4.1 User Data Entity

Data Item	**Type**	**Description**	**Comment**
id	Integer	Unique identifier	
name	Text	Name of user	
email	Text	Email of user	
type	Text	Type of user	May be ‘STUDENT’, ‘INSTRUCTOR’, OR ‘ADMIN’
Enrollment	Pointer	Enrollments for user	May have several

3.4.2 Enrollment Data Entity

Data Item	**Type**	**Description**	**Comment**
enrollment_id	Integer	Unique identifier	
grade	Text	Grade for user	May be null until final grade is input
user_id	Integer	Pointer to user entity	Can only be point to one user
section_no	Integer	Pointer to section entity	Can only point to one section
grade	Pointer	Grades for enrollments	May have several

3.4.3 Grade Data Entity

Data Item	**Type**	**Description**	**Comment**
grade_id	Integer	Unique identifier	

score	Integer	Score for assignment	May be null
assignment_id	Integer	Pointer to Assignment	Can only point to one assignment
enrollment_id	Integer	Pointer to enrollment	Can only point to one enrollment

3.4.4 Assignment Data Entity

Data Item	**Type**	**Description**	**Comment**
assignment_id	Integer	Unique identifier	
title	Text	Title for assignment	
due_date	Date	Due date for assignment	May be null
section_no	Integer	Pointer to section	Can only point to one section
grade	Pointer	Grades for assignments	May have several

3.4.5 Section Entity

Data Item	**Type**	**Description**	**Comment**
section_no	Integer	Unique Identifier	
section_id	Integer		Not unique, there can be many sections with different times/terms
instructor_email	Text	Section instructor email	May be null
building	Text	Building building for section	May be null

room	Text	Room for section	May be null
times	Text	Time of class	May be null
course_id	Text	Pointer to course	Can only point to one course
term_id	Integer	Pointer to term	Can only point to one term
Enrollment	Pointer	Enrollment Entity	May have several
Assignment	Pointer	Assignment Entity	May have several

3.4.6 Course Data Entity

Data Item	**Type**	**Description**	**Comment**
course_id	Text	Course name	
title	Text	Title of the course	
credits	Integer	Amount of credits	
Section	Pointer	Section Entity	May have several

3.4.7 Term Data Entity

Data Item	**Type**	**Description**	**Comment**
term_id	Integer	Unique identifier	
tyear	Integer	Year of term	
semester	Text	Semester of Term	‘Spring’, ‘Summer’, ‘Fall’, ‘Winter’
addDate	Date	First date to add section	
addDeadline	Date	Deadline to enroll in	

		section	
dropDeadline	Date	Deadline to drop section	
startDate	Date	Start date of section	
endDate	Date	End date of section	
Section	Pointer	Section Entity	May have several