

# Software Requirement Specifications for Code Grader

**Version 1.0**

Prepared by  
Thomas Kwashnak, Isaac Crawford, and Sadjell Mamon

Quinnipiac University

December 6, 2022

## Revision History

Date	Version	Description	Author
11/6/2022	1.0.0	Initial Document	Thomas Kwashnak, Sadjell Mamon, Isaac Crawford

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Intended Audience and Reading Suggestion . . . . .	4
1.3	References . . . . .	4
<b>2</b>	<b>Overall Description</b>	<b>4</b>
2.1	Product Perspective . . . . .	4
2.2	Product Functions . . . . .	4
2.3	User Classes and Characteristics . . . . .	4
2.4	Operating Environment . . . . .	5
2.5	Design and Implementation Constraints . . . . .	5
2.6	Assumptions and Dependencies . . . . .	5
<b>3</b>	<b>Specific Requirements</b>	<b>6</b>
3.1	Interfaces . . . . .	6
3.1.1	Login Page . . . . .	6
3.1.2	Instructor Home . . . . .	6
3.1.3	Admin Home . . . . .	7
3.1.4	Student Home . . . . .	7
3.1.5	Instructor Course View . . . . .	8
3.1.6	Student Grades View . . . . .	8
3.1.7	Create Assignment . . . . .	9
3.1.8	View Assignment . . . . .	9
3.1.9	Instructor View Submission . . . . .	10
3.1.10	Create Account . . . . .	10
3.1.11	Create Course . . . . .	11
3.2	Use Case Diagram . . . . .	12
3.3	System Features . . . . .	13
3.3.1	Login . . . . .	13
3.3.2	Create Instructor Account . . . . .	13
3.3.3	Create Course . . . . .	14
3.3.4	Create Student Account . . . . .	14
3.3.5	Upload Test Case . . . . .	15
3.3.6	Create and Update Assignments . . . . .	15
3.3.7	Submit Assignments . . . . .	16
3.3.8	View Student Grades . . . . .	17
3.3.9	Download Student Grades . . . . .	17
3.3.10	Provide Submission Feedback . . . . .	18
3.3.11	Delete Instructor Account . . . . .	18
3.3.12	Delete Course . . . . .	19
3.3.13	Delete Student Account . . . . .	19
3.3.14	Edit Instructor Account . . . . .	20
3.3.15	Edit Course . . . . .	20
3.3.16	Edit Student Account . . . . .	21
3.3.17	View Test Case Output . . . . .	21
3.4	Nonfunctional Requirements . . . . .	22
3.4.1	Performance Requirements . . . . .	22
3.4.2	Safety Requirements . . . . .	22
3.4.3	Software Quality Requirements . . . . .	22
3.4.4	Security Requirements . . . . .	22



# 1 Introduction

## 1.1 Purpose

The purpose of this document is to outline the requirements for the development of the Code Grader System. This document will provide information about the functionality the Code Grader system will have. This project will be completed by Spring 2023

This project is designed to help professors grade assignments faster and more efficiently. Students will be able to join classes and submit their assignments; while teachers will be able to add students to their classes and see their submissions.

## 1.2 Intended Audience and Reading Suggestion

This document is designed for project managers and developers to help them understand the clients' requirements. It contains a project's description, requirements, and use cases. The reader should start with the description and end with the requirements.

## 1.3 References

Code Grader Vision Document v1.0.0

# 2 Overall Description

## 2.1 Product Perspective

Quinnipiac University desires to improve their grading system. This new product should help professors grade the submitted assignments quickly and efficiently. By creating a system for assignment submission and grading, this product should allow students to automatically receive a grade after submitting their assignments.

## 2.2 Product Functions

- Allow all users to create an account and log in.
- Allow all professors to create a class.
- Allow all professors to invite students to a class.
- Allow all students to join a class.
- Allow all professors to post assignments
- Allow all students to view/submit assignments.
- Allow all users to see grades of their submitted/posted assignment.

## 2.3 User Classes and Characteristics

Class	Description
Admin	The admin will use the system to create and manage instructor accounts
Professors	The professors will use the system to create classes, add students to the classes, post assignments, change the grades, and leave comments.
Students	The students will use the system to view/submit assignments, and access grades and comments

## **2.4 Operating Environment**

The system will be hosted on a website that will be accessible by all common web browsers (Google Chrome, Safari, Firefox, etc.). The system will require an internet connection to be accessible. No additional software components or applications will be required to use the system

## **2.5 Design and Implementation Constraints**

The database must be able to handle up to 5000 requests at a time. The web application must be usable on mobile devices. The client is responsible for maintaining the application after completion

## **2.6 Assumptions and Dependencies**

Professors can have multiple classes. Student accounts are created on a per-class basis

## 3 Specific Requirements

### 3.1 Interfaces

#### 3.1.1 Login Page

### Log in to CodeGrader

[Log In](#)

#### 3.1.2 Instructor Home

### Instructor Home

[Log out](#)[SER 101](#)[SER 102](#)[SER 103](#)[Create Course](#)

### 3.1.3 Admin Home

Admin Home

Log out

Instructors

Professor Hoffman	Edit	Delete
Professor Alex	Edit	Delete
Professor Dean	Edit	Delete
Professor Seventy-Four	Edit	Delete

Add New Instructor

### 3.1.4 Student Home

SER 101

Log out

Assignment 1

Closed

Assignment 2

Open

Assignment 3

Open

Assignment 4

Open

Assignment 5

Open



### 3.1.5 Instructor Course View

[Back](#)[Log out](#)

SER 101

Assignment 1Edit

Assignment 2Edit

Assignment 3Edit

Assignment 4Edit

Add Student to Course

View Students

### 3.1.6 Student Grades View

[Back](#)[Log out](#)

Student Name	Assignment 1	Assignment 2	Assignment 3	Total Grade
Billy	0/100	50/100	30/100	80/300
Bob	90/100	95/100	80/100	??/100
Bo	0/100	50/100	30/100	£600
Meep	NA	NA	NA	0/300
Salem	40/100	NA	NA	40/300

[Add Student](#)[Download Grades](#)

### 3.1.7 Create Assignment

[Back](#)[Log out](#)

Create Assignment

Assignment Name

Description

Test Case A

Test Case B

Test Case C

Test Case D

Upload Test Case

Post Assignment

### 3.1.8 View Assignment

[Back](#)[Log out](#)

Assignment 3 - LOREM IPSUM

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

code.py

Upload Code

Submit

Submission Results

Test Case 1: Pass

Test Case 2: Fail

Don't forget to indent your code!

ACTUAL OUTPUT HERE

EXPECTED OUTPUT HERE

9

### 3.1.9 Instructor View Submission

[Back](#)[Log out](#)

### Submission for Assignment X

70%

```
//filler submission for Code Assignment

public static void main(String args[]) {
    while(true) {
        while(true) {
            while(true) {
                System.out.println("oops");
            }
        }
    }
}
```

Test Case 1: Pass

Test Case 2: Fail

Test Case 3: Pass

Test Case 4: Pass

Add comments here...

Post Comment

### 3.1.10 Create Account

[Back](#)[Log out](#)

### Create New Account

Name

Username

Password

Re-enter Password

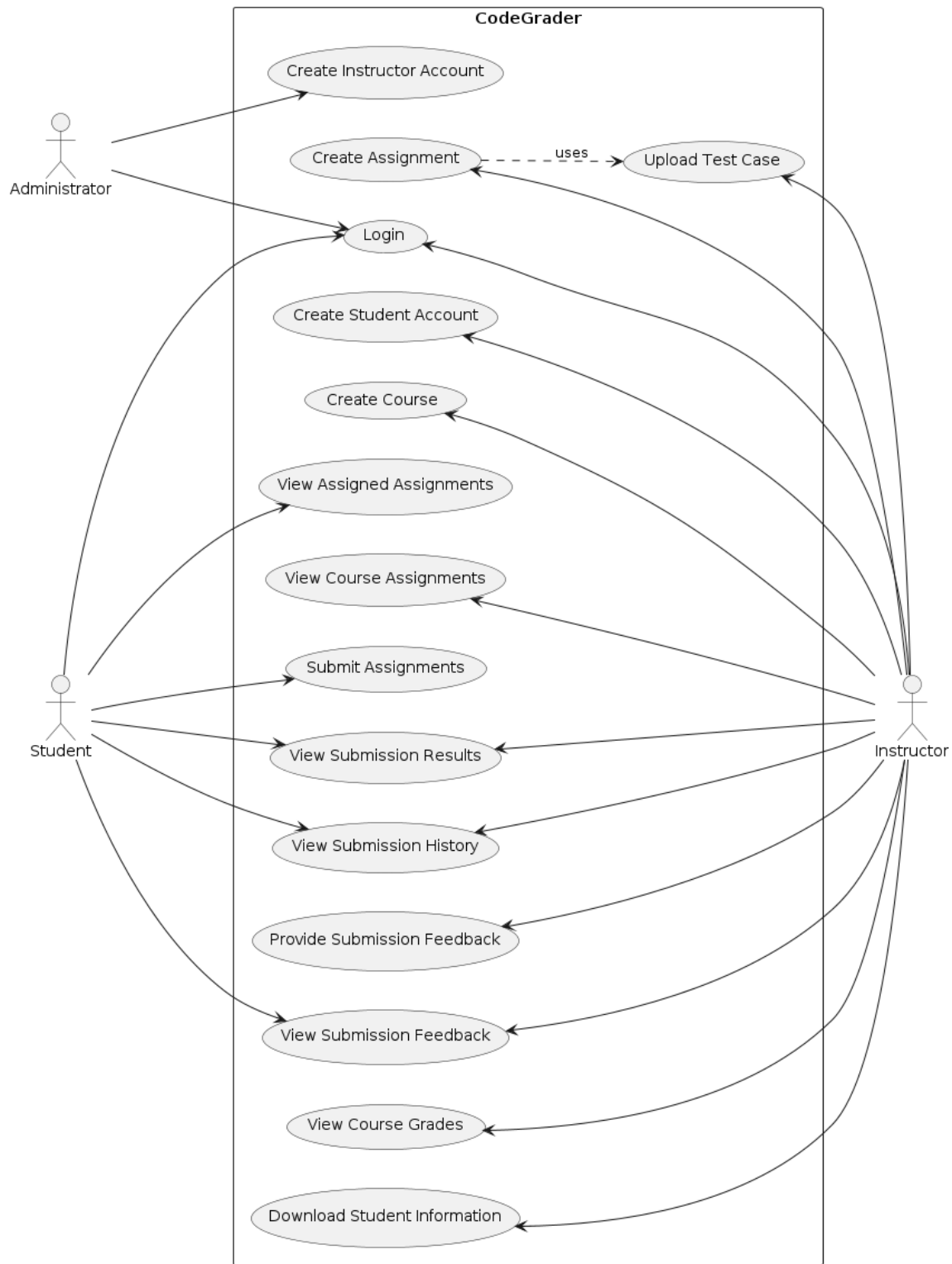
Create Account

### 3.1.11 Create Course

[Back](#)[Log out](#)

#### Create New Course

### 3.2 Use Case Diagram



### 3.3 System Features

#### 3.3.1 Login

**Description** An admin, instructor, or student signs into the application using their username and password

##### Functional Requirements and Use Cases

- **REQ-1** The system should allow an admin to login using their username and password
- **REQ-2** The system should allow an instructor to login using their username and password
- **REQ-3** The system should allow a student to login using their username and password

<b>Use Case Name</b>	Login
<b>Brief Description</b>	The user logs into their account using their username and password
<b>Actor(s)</b>	User
<b>Pre-conditions</b>	None
<b>Basic Flow</b>	<ol style="list-style-type: none"><li><b>1</b> The user opens up the web application</li><li><b>2</b> The system prompts the user to sign in using a username and password</li><li><b>3</b> The user provides their account's username and password, and submits</li><li><b>4</b> The system displays the user's home page</li></ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"><li><b>3A</b> The user inputs a username that does not exist</li><li><b>3A1</b> The system displays an error stating that either the username or password is incorrect, and asks the user to try signing in again</li><li><b>3B</b> The user inputs a password that does not match the user's actual password</li><li><b>3B1</b> The system displays an error stating that either the username or password is incorrect, and asks the user to try signing in again</li></ol>

#### 3.3.2 Create Instructor Account

**Description** An admin creates an instructor account, setting it's username and password

##### Functional Requirements and Use Cases

- **REQ-4** The system should allow an admin to create an instructor account with a username and password

<b>Use Case Name</b>	Create Instructor Account
<b>Brief Description</b>	The admin creates an instructor account, and specifies the username and password
<b>Actor(s)</b>	Admin
<b>Pre-conditions</b>	The admin is logged in, and is on the admin homepage
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li><b>1</b> The admin chooses to create a new instructor account</li> <li><b>2</b> The system prompts the admin to specify the instructor name, username, and password</li> <li><b>3</b> The admin fills in the name, username and password, and submits</li> <li><b>4</b> The system creates the account and returns the admin to the admin homepage</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li><b>3A</b> The user inputs a username that already exists</li> <li><b>3A1</b> The system displays an error asking the admin to choose another username</li> </ol>

### 3.3.3 Create Course

**Description** The instructor creates a course with a name and description

#### Functional Requirements and Use Cases

- **REQ-5** The system should allow an instructor to create a new course with a name and description

<b>Use Case Name</b>	Create Course
<b>Brief Description</b>	The instructor creates a course
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The instructor is logged in, and is currently on the instructor homepage
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li><b>1</b> The instructor chooses to create a new course</li> <li><b>2</b> The system prompts the instructor to fill in the new course name</li> <li><b>3</b> The instructor fills in the details and submits</li> <li><b>4</b> The system creates the course, and displays the course's page</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li><b>3A</b> The instructor already has a course with the given course-name</li> <li><b>3A1</b> The system displays an error asking the instructor to choose another course name</li> </ol>

### 3.3.4 Create Student Account

**Description** The instructor creates a student account for a course

#### Functional Requirements and Use Cases

- **REQ-6** The system should allow an instructor to create a student account for a course with a username and a password

<b>Use Case Name</b>	Create Student Account
<b>Brief Description</b>	The instructor creates a student account for a course
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The instructor is logged in, and is at the home page of one of their courses
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1 The instructor chooses to view the student grades list</li> <li>2 The system shows the user a list of all currently enrolled students, and the option to create a new account</li> <li>3 The instructor chooses to create a new account</li> <li>4 The system prompts the instructor to input the new account's name, username, and password</li> <li>5 The user inputs the relevant information and submits</li> <li>6 The system creates the user, and then shows an updated list of all students currently enrolled in that course.</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li>5A The instructor inputs a username that already has an account</li> <li>5A1 The system asks the instructor to input a different username because the username is currently in use.</li> </ol>

### 3.3.5 Upload Test Case

**Description** An instructor uploads files required to create a test case for an assignment

#### Functional Requirements and Use Cases

- **REQ-7** The system should allow an instructor to upload an input and expected output file as a test case.

<b>Use Case Name</b>	Upload Test Case
<b>Brief Description</b>	The instructor uploads a test case as two files including the input and output
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The instructor is logged in, and is currently editing or creating an assignment
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1 The instructor chooses to upload a test case</li> <li>2 The system prompts the instructor to upload two documents, one labelled 'Input' and the other labelled 'Output'</li> <li>3 The instructor uploads the input and output files and submits</li> <li>4 The system creates the test case, attaches it to the assignment, and returns the instructor to the assignment editor</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li>3A The instructor does not upload an input file</li> <li>3A1 The system prints an error and asks the instructor to upload an input file</li> <li>3B The instructor does not upload an output file</li> <li>3B1 The system prints an error and asks the instructor to upload an output file</li> </ol>

### 3.3.6 Create and Update Assignments

**Description** An instructor is able to create and update assignments within a course



### Functional Requirements and Use Cases

- **REQ-8** The system allows an instructor to create an assignment with a name, description, and test cases
- **REQ-9** The system allows an instructor to edit any assignment in one of their courses

<b>Use Case Name</b>	Create Assignment
<b>Brief Description</b>	The instructor creates an assignment in one of their courses
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The instructor is logged in
<b>Basic Flow</b>	<ol style="list-style-type: none"><li>1 The instructor chooses a course</li><li>2 The system displays a list of assignments in that course</li><li>3 The instructor chooses to create an assignment</li><li>4 The system prompts the instructor to input the assignment name, description, and upload any test cases</li><li>5 The instructor inputs the proper data, and submits the assignment</li><li>6 The system displays an updated assignment list with the new assignment</li></ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"><li>3A The instructor does not upload an input file</li><li>3A1 The system prints an error and asks the instructor to upload an input file</li><li>3B The instructor does not upload an output file</li><li>3B1 The system prints an error and asks the instructor to upload an output file</li></ol>

<b>Use Case Name</b>	Update Assignment
<b>Brief Description</b>	The instructor creates an assignment in one of their courses
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The instructor is logged in
<b>Basic Flow</b>	<ol style="list-style-type: none"><li>1 The instructor chooses a course</li><li>2 The system displays a list of assignments in that course</li><li>3 The instructor chooses to edit one of the assignments in the course</li><li>4 The system displays an editable screen for the assignment, allowing the instructor to edit the name, description, and upload new test cases</li><li>5 The user updates the assignment and submits</li><li>6 The system updates the assignment and displays an updated assignment list</li></ol>
<b>Alternate Flows</b>	None

### 3.3.7 Submit Assignments

**Description** A student completes and submits an assignment for grading

### Functional Requirements and Use Cases

- **REQ-10** The system should allow a student to upload and submit their code for an assignment

<b>Use Case Name</b>	Submit Assignment
<b>Brief Description</b>	The student uploads their code and submits an assignment
<b>Actor(s)</b>	Student
<b>Pre-conditions</b>	The Student is Logged In
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1 The student chooses an assignment to complete</li> <li>2 The system displays details for that assignment, and gives the student the ability to upload code and submit</li> <li>3 The student uploads their code and submits</li> <li>4 The system uploads the code and submits the assignment</li> <li>5 The system grades the assignment, and displays the grading results to the user</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li>3A The student does not upload any file and submits</li> <li>3A1 The system prints an error requesting that the user uploads a code file to submit</li> </ol>

### 3.3.8 View Student Grades

**Description** The instructor views the submission grades of students in a course

#### Functional Requirements and Use Cases

- **REQ-11** The system should allow an instructor to view a list of grades in a course

<b>Use Case Name</b>	View Course Grades
<b>Brief Description</b>	The instructor navigates and views a list of course grades
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The Instructor is Logged In
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1 The instructor chooses a course</li> <li>2 The system displays the list of assignments for the course</li> <li>3 The instructor chooses to view course grades</li> <li>4 The system displays a table of students and their grades for each assignment</li> </ol>
<b>Alternate Flows</b>	None

### 3.3.9 Download Student Grades

**Description** The instructor downloads student grades as a file

#### Functional Requirements and Use Cases

- **REQ-12** The system should allow an instructor to download course grades as a file

<b>Use Case Name</b>	Download Course Grades
<b>Brief Description</b>	The instructor downloads student grades for a particular course
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The Instructor is Logged In and is viewing grades for one of their courses
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1 The instructor chooses to download the course grades</li> <li>2 The system asks the instructor what file format they want the grades in</li> <li>3 The instructor selects a file format</li> <li>4 The system compiles the grades into a file of the provided format, and sends it to the instructor</li> </ol>
<b>Alternate Flows</b>	None

### 3.3.10 Provide Submission Feedback

**Description** The instructor provides feedback to the submission of a particular student for a particular assignment

#### Functional Requirements and Use Cases

- **REQ-13** The system should allow the instructor to add feedback to a student's assignment submissions

<b>Use Case Name</b>	Add Submission Feedback
<b>Brief Description</b>	The instructor provides feedback to a submission
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The Instructor is Logged In and is currently viewing the list of assignments for a course
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1 The instructor chooses one of the assignment grades for a student</li> <li>2 The system displays that submission details, including the submitted code and the test case results. The system also displays an option to add feedback as a comment</li> <li>3 The instructor chooses to add feedback</li> <li>4 The system prompts the user to input the feedback</li> <li>5 The instructor inputs the feedback and submits</li> <li>6 The system adds the feedback to the submission and displays the submission</li> </ol>
<b>Alternate Flows</b>	None

### 3.3.11 Delete Instructor Account

**Description** The admin deletes an instructor account

#### Functional Requirement and Use Cases

- **REQ-14** The system should allow an admin to delete an instructor account

<b>Use Case Name</b>	Delete Instructor Account
<b>Brief Description</b>	The admin deletes an instructor account
<b>Actor(s)</b>	Admin
<b>Pre-conditions</b>	The admin is logged in and sees a list of instructors
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li><b>1</b> The admin chooses to delete an instructor from the list</li> <li><b>2</b> The system asks the admin if they are sure they would like to delete that account, all courses under that account, and all students in those courses</li> <li><b>3</b> The admin chooses to confirm and delete the account</li> <li><b>4</b> The system deletes the account, all courses under the account, and all students under those courses. The system then displays the updated list of instructors to the admin.</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li><b>3A</b> The admin chooses to cancel</li> <li><b>3A1</b> The system returns to the admin home page</li> </ol>

### 3.3.12 Delete Course

**Description** An instructor deletes one of their courses

#### Functional Requirements and Use Cases

- **REQ-15** The system should allow the instructor to delete one of their courses

<b>Use Case Name</b>	Delete Course
<b>Brief Description</b>	An instructor deletes a course
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The instructor is currently logged in
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li><b>1</b> The instructor chooses to delete a course from their list of courses</li> <li><b>2</b> The system asks the instructor if they really wish to delete that course</li> <li><b>3</b> The instructor confirms they wish to delete the course</li> <li><b>4</b> The system deletes the course, and all enrolled student accounts</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li><b>3A</b> The instructor chooses to cancel</li> <li><b>3A1</b> The system displays an unchanged course list</li> </ol>

### 3.3.13 Delete Student Account

**Description** An instructor deletes a student account from their course.

#### Functional Requirements and Use Cases

- **REQ-16** The system should allow the instructor to delete a student account in one of their courses.

<b>Use Case Name</b>	Delete Student Account
<b>Brief Description</b>	An instructor deletes a student account from one of their courses
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The instructor is currently logged in, and has navigated to the student list in one of their courses
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li><b>1</b> The instructor chooses to delete one of the students</li> <li><b>2</b> The system asks the instructor if they really wish to delete that student account</li> <li><b>3</b> The instructor confirms they wish to delete that student account</li> <li><b>4</b> The system deletes the student account, and updates the student list</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li><b>3A</b> The instructor chooses to cancel</li> <li><b>3A1</b> The system displays an unchanged student list</li> </ol>

### 3.3.14 Edit Instructor Account

**Description** The admin edits the information for an instructor account

#### Functional Requirement and Use Cases

- **REQ-17** The system should allow an admin to edit an instructor account's name, username, and password

<b>Use Case Name</b>	Edit Instructor Account
<b>Brief Description</b>	The admin edits an instructor account
<b>Actor(s)</b>	Admin
<b>Pre-conditions</b>	The admin is logged in and sees a list of instructors
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li><b>1</b> The admin chooses to edit an instructor account from the list</li> <li><b>2</b> The system gives the admin the options to change the instructor account's name, username, and password</li> <li><b>3</b> The admin updates the instructor's information and submits</li> <li><b>4</b> The system updates the instructor's account and displays an updated list of instructors</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li><b>3A</b> The admin changes the username to a username that is already in use</li> <li><b>3A1</b> The system prints an error stating that the username is in use, and asks the admin to choose a new username</li> </ol>

### 3.3.15 Edit Course

**Description** An instructor edits the information of one of their courses

#### Functional Requirements and Use Cases

- **REQ-18** The system should allow the instructor to edit the name of one of their courses

<b>Use Case Name</b>	Edit Course
<b>Brief Description</b>	An instructor edits a course
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The instructor is currently logged in
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li><b>1</b> The instructor chooses to edit a course in their list</li> <li><b>2</b> The system displays options to change the course name</li> <li><b>3</b> The instructor updates the course information and submits</li> <li><b>4</b> The system updates the course information and displays an updated course list</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li><b>3A</b> The instructor changes the name to a name that is already in use by the instructor</li> <li><b>3A1</b> The system displays an error saying that the instructor needs to choose a different course name</li> </ol>

### 3.3.16 Edit Student Account

**Description** An instructor edits a student account in their course.

#### Functional Requirements and Use Cases

- **REQ-19** The system should allow the instructor to delete a student account in one of their courses.

<b>Use Case Name</b>	Edit Student Account
<b>Brief Description</b>	An instructor edits a student account in one of their courses
<b>Actor(s)</b>	Instructor
<b>Pre-conditions</b>	The instructor is currently logged in, and has navigated to the student list in one of their courses
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li><b>1</b> The instructor chooses to edit one of their students</li> <li><b>2</b> The system displays options to change the students name, username, and password</li> <li><b>3</b> The instructor updates the information and submits</li> <li><b>4</b> The system updates the student account data and displays the updated student list</li> </ol>
<b>Alternate Flows</b>	<ol style="list-style-type: none"> <li><b>3A</b> The instructor changes the username to a different username that already exists</li> <li><b>3A1</b> The system displays an error and asks the instructor to choose a different username</li> </ol>

### 3.3.17 View Test Case Output

**Description** A student views the output compared to the expected output for a test case in one of their submissions.

#### Functional Requirements and Use Cases

- **REQ-20** The system should allow the student to view the actual output compared to the expected output for a test case.

<b>Use Case Name</b>	View Test Case Results
<b>Brief Description</b>	A student views the test case results for a submission
<b>Actor(s)</b>	Student
<b>Pre-conditions</b>	The student is signed in, and has submitted an assignment at least once
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li><b>1</b> The student opens an assignment that they have already submitted</li> <li><b>2</b> The system displays the assignment information, as well as the student's submission. The system displays a list of test cases and whether the student has passed or failed each one</li> <li><b>3</b> The student chooses one of the test cases</li> <li><b>4</b> The system displays the details of the test case, including the comparison between the actual result, and expected result</li> </ol>
<b>Alternate Flows</b>	None

### 3.4 Nonfunctional Requirements

#### 3.4.1 Performance Requirements

- The system should display a login screen within 2 seconds
- The system should grade a submission within 2 minutes

#### 3.4.2 Safety Requirements

- The system should not harm the user under any circumstances

#### 3.4.3 Software Quality Requirements

- The system must hold usernames and passwords in a secure database
- Upon login, the system should send a text message or email to authenticate the user

#### 3.4.4 Security Requirements

- To comply with FERPA, the system should restrict student information to only be accessible by either the student or instructor.

## A Glossary

**Test Case** A test case is an execution instance that tests submission code for expected results. Test cases consist of two parts: an input and output. The input is what is fed into the code, and the output is what the code should output. The actual output is compared with the expected output to pass or fail the test.