

# Vision Document 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
October 5th, 2022	1.0	Initial Submission	Thomas Kwashnak, Sadjell Mamon, Isaac Crawford

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Goal . . . . .	3
1.3	Scope . . . . .	3
1.4	Definitions, Acronyms, and Abbreviations . . . . .	3
1.5	References . . . . .	3
<b>2</b>	<b>Positioning</b>	<b>3</b>
2.1	Problem Statement . . . . .	3
<b>3</b>	<b>Stakeholder and User Descriptions</b>	<b>4</b>
3.1	Stakeholder and User Summary . . . . .	4
3.1.1	Developer . . . . .	4
3.1.2	Instructor . . . . .	4
3.1.3	Student . . . . .	5
3.1.4	Administrator . . . . .	5
3.1.5	Advisor . . . . .	5
3.2	Stakeholder and User Environment . . . . .	6
<b>4</b>	<b>Product Overview</b>	<b>6</b>
4.1	Summary of System Capabilities . . . . .	6
4.2	Assumptions . . . . .	6
4.3	Dependencies . . . . .	6
<b>5</b>	<b>Product Features</b>	<b>6</b>
5.1	Instructor Account Creation . . . . .	6
5.2	Student Account Creation . . . . .	6
5.3	Course Creation . . . . .	6
5.4	Assignment Creation . . . . .	6
5.5	Download Student Information . . . . .	7
5.6	Assignment Submission . . . . .	7
5.7	Assignment Grading . . . . .	7
5.8	Web Application . . . . .	7
5.9	Score History . . . . .	7
5.10	Submission Feedback . . . . .	7
5.11	Instructor Course View . . . . .	7
5.12	Instructor Assignment View . . . . .	7
5.13	Student Assignment View . . . . .	7
<b>6</b>	<b>Constraints</b>	<b>7</b>
6.1	Portability . . . . .	7
6.2	Security . . . . .	7
6.3	Responsiveness . . . . .	7
6.4	Capacity . . . . .	8
6.5	Compatibility . . . . .	8

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to provide a list of features gathered from all stakeholders involved in this discussion. This document provides an overview of the project. This document also discusses a proposed solution to how the Code Grader system fulfils all listed requirements.

## 1.2 Goal

The main goal of the Code Grader system is to simplify the grading of coding-based assignments. This both removes the need to manually grade individual student's submissions, while providing the student with near-immediate feedback for the student to learn from.

## 1.3 Scope

This document is being written for the further development of the Code Grader

## 1.4 Definitions, Acronyms, and Abbreviations

**Instructor** The professor or teacher running a course. The instructor is the user that assigns coding assignments and manages the course

**Administrator** Users that can manage other accounts and courses open

**Student** A user that is currently enrolled in a course led by an instructor. Students are the users who submit their answers to the homework assignments.

**Linux** An open-source unix-like operating system based on the linux kernel. Often used for servers due to its reliability and extensive command line tools.

## 1.5 References

None

# 2 Positioning

## 2.1 Problem Statement

<b>The problem of</b>	having to manually grade programming assignments
<b>Affects</b>	professors and students
<b>The impact of which is</b>	more time consumed for the professor and a longer wait for the student to receive feedback on their assignment.
<b>A successful solution would be</b>	automatically giving feedback to the student and allowing them multiple chances to fix their work, as well as grading the submitted code for the professor

### 3 Stakeholder and User Descriptions

#### 3.1 Stakeholder and User Summary

##### 3.1.1 Developer

<b>Description</b>	Thomas Kwashnak, Sadjell Mamon, and Isaac Crawford
<b>Type</b>	Developer
<b>Responsibilities</b>	<ul style="list-style-type: none"><li>• Create an application which serves as a replacement for Mimir</li><li>• Ensure all the requirements are properly implemented</li></ul>
<b>Success Criteria</b>	<ul style="list-style-type: none"><li>• The system possesses all important requirements</li><li>• The system has a low probability of crashing</li><li>• The system automatically grades programming assignments after submissions</li></ul>
<b>Involvement</b>	<ul style="list-style-type: none"><li>• Application Designing</li><li>• Development</li><li>• Software Testing</li></ul>
<b>Deliverables Required</b>	<ul style="list-style-type: none"><li>• System that fulfils all the specified requirements based off of user needs</li></ul>
<b>Comments / Issues</b>	None

##### 3.1.2 Instructor

<b>Description</b>	Quinnipiac's Software Engineering Instructors using the system for a course
<b>Type</b>	User
<b>Responsibilities</b>	<ul style="list-style-type: none"><li>• Create courses</li><li>• Create student accounts for courses</li><li>• Post assignments</li></ul>
<b>Success Criteria</b>	<ul style="list-style-type: none"><li>• Instructor can create a course</li><li>• Instructor can create student accounts for their courses</li><li>• Instructor can post assignments</li><li>• Instructor can export grades</li></ul>
<b>Involvement</b>	<ul style="list-style-type: none"><li>• Uses the final product to instruct courses and assign coding assignments to students</li></ul>
<b>Deliverables Required</b>	None
<b>Comments / Issues</b>	None

### 3.1.3 Student

<b>Description</b>	Student currently taking a course from an instructor using the system
<b>Type</b>	User
<b>Responsibilities</b>	<ul style="list-style-type: none"><li>• Submit code solutions to course assignments to be graded</li></ul>
<b>Success Criteria</b>	<ul style="list-style-type: none"><li>• Students can access courses their professors assigned them to</li><li>• Students can upload a solution to the assignments for their courses</li><li>• Students see graded feedback on their submitted solution after submitting and can submit again</li></ul>
<b>Involvement</b>	<ul style="list-style-type: none"><li>• Input files that will be used by the system</li></ul>
<b>Deliverables Required</b>	None
<b>Comments / Issues</b>	None

### 3.1.4 Administrator

<b>Description</b>	Prof. Blake
<b>Type</b>	Admin / Client
<b>Responsibilities</b>	<ul style="list-style-type: none"><li>• Provide feedback during system development</li><li>• Providing requirements</li><li>• Testing the system</li><li>• Create instructor accounts</li></ul>
<b>Success Criteria</b>	<ul style="list-style-type: none"><li>• Instructor accounts can be created</li></ul>
<b>Involvement</b>	<ul style="list-style-type: none"><li>• Ensure the product satisfies all the requirements</li><li>• Use the product to create instructor accounts</li></ul>
<b>Deliverables Required</b>	None
<b>Comments / Issues</b>	None

### 3.1.5 Advisor

<b>Description</b>	Prof. Elkhartboutly
<b>Type</b>	Project Advisor
<b>Responsibilities</b>	<ul style="list-style-type: none"><li>• Review and grade submitted documents</li></ul>
<b>Success Criteria</b>	<ul style="list-style-type: none"><li>• The application meets all the specified requirements correctly</li></ul>
<b>Involvement</b>	<ul style="list-style-type: none"><li>• Manages developers, provides feedback</li></ul>
<b>Deliverables Required</b>	None
<b>Comments / Issues</b>	None

### 3.2 Stakeholder and User Environment

The system is to be used both in and out of the classroom environment, to provide professors and students with flexibility in uploading and submitting assignments respectively.

## 4 Product Overview

### 4.1 Summary of System Capabilities

User Benefit	Supporting Feature
Near-Immediate Feedback	<ul style="list-style-type: none"><li>• Automatic testing of submissions against test cases</li><li>• Instructor can set test cases</li></ul>
Exportable Grades to a File	<ul style="list-style-type: none"><li>• Instructor can download grades in preferred file format</li></ul>
Ability to manage courses and students	<ul style="list-style-type: none"><li>• Instructor can create student accounts</li><li>• Instructor can create courses and assign specific students to those courses</li></ul>

### 4.2 Assumptions

It is assumed that the system used to host the server will run on a linux based operating system to provide sufficient access to command-line tools.

### 4.3 Dependencies

None

## 5 Product Features

### 5.1 Instructor Account Creation

The admin will be able to create instructor accounts.

### 5.2 Student Account Creation

The instructor will be able to create student accounts for a course.

### 5.3 Course Creation

The instructor will be able to create a course.

### 5.4 Assignment Creation

The instructor will be able to create an assignment for a course, specifying a due date, as well as upload test cases and starter code for that assignment. Instructor should be able to specify the language of the assignment.

## **5.5 Download Student Information**

The instructor will be able to access a list of students taking the course. The instructor should also be able to download a file of assignment submissions by name.

## **5.6 Assignment Submission**

The student should be able to submit assignments by uploading their solution code.

## **5.7 Assignment Grading**

Assignment submissions will need to be automatically graded against test cases by the system when they are uploaded.

## **5.8 Web Application**

The application needs to be accessed through a modern browser with an internet connection.

## **5.9 Score History**

A history of attempts, including the grades, needs to be accessible by both the student and the instructor.

## **5.10 Submission Feedback**

The instructor will be able to provide feedback on individual submissions

## **5.11 Instructor Course View**

Instructor can view a list of courses that they are instructing

## **5.12 Instructor Assignment View**

Instructor needs to be able to view a list of assignments for a particular course.

## **5.13 Student Assignment View**

The student needs to be able to view a list of assignments in a course

# **6 Constraints**

## **6.1 Portability**

The application interface needs to be usable on various screen sizes, including mobile screens.

## **6.2 Security**

The system must comply with FERPA law. Therefore, access to view students grades should be restricted to only those with permissions under FERPA. For the sake of the project, this means that a student's grades should only be accessible by the course instructor.

## **6.3 Responsiveness**

The system must be able to run and grade code submissions within 1 minute of uploading in order to provide quick feedback to the student.



## **6.4 Capacity**

The system should be able to handle at least 20 students accessing the system at once.

## **6.5 Compatibility**

The web application should be compatible with most modern web browsers.