Vision Document
for
Code Grader

Version 1.0

Prepared by
Thomas Kwashnak, Isaac Crawford, and Sadjell Mamon

Quinnipiac University

October 7, 2022

Revision History

Date	Version	Description	Author
October 5th, 2022	1.0	Initial Submission	Thomas Kwashnak, Sadjell Mamon, Isaac Crawford

Contents

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

The problem of	having to manually grade programming assignments
Affects	professors and students
The impact of which is	more time consumed for the professor and a longer wait for the student to receive feedback on their assignment.
A successful solution would be	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

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

3.1.2 Instructor

Description	Quinnipiac's Software Engineering Instructors using the system for a course
Type	User
Responsibilities	• Create courses • Create student accounts for courses • Post assignments
Success Criteria	• Instructor can create a course • Instructor can create student accounts for their courses • Instructor can post assignments • Instructor can export grades
Involvement	• Uses the final product to instruct courses and assign coding assignments to students
Deliverables Required	None
Comments / Issues	None

3.1.3 Student

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

3.1.4 Administrator

Description	Prof. Blake
Type	Admin / Client
Responsibilities	• Provide feedback during system development • Providing requirements • Testing the system • Create instructor accounts
Success Criteria	• Instructor accounts can be created
Involvement	• Ensure the product satisfies all the requirements • Usese the product to create instructor accounts
Deliverables Required	None
Comments / Issues	None

3.1.5 Advisor

Description	Prof. Elkharboutly
Type	Project Advisor
Responsibilities	• Review and grade submitted documents
Success Criteria	• The application meets all the specified requirements correctly
Involvement	• Manages developers, provides feedback
Deliverables Required	None
Comments / Issues	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	• Automatic testing of submissions against test cases • Instructor can set test cases
Exportable Grades to a File	• Instructor can download grades in preferred file format
Ability to manage courses and students	• Instructor can create student accounts • Instructor can create courses and assign specific students to those courses

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 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 assignemnt 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.