Exam Reviewer Project

2024 - 2025 CMSC 202 Group 2

Software Requirements Specification

Version 1.0

1. Introduction

1.1. Purpose

This document aims to provide a structure and a comprehensive overview of the system to be developed. We shall describe in detail the system's expected functionalities, target users, intended use-cases, program flow, and limitations. The target audience of this document are the developers, approvers, and end-users.

1.2. Scope

The system will be a text-based implementation of an exam reviewer using mock exams as the primary method for review. A pre-compiled list of questions included in the package will be the basis of the exams presented during the review and the condensed development timeline necessitated restriction to just "True/False" and "Multiple Choice" types of questions for now.

The system will have the functionality to show performance metrics after the review. There is also the ability for the user to save current results - including answers and scores - and view historical review performance at any given time.

Additionally, the user will also be provided with several customization options such as topic filters and time limits for a more focused review.

1.3. Glossary

Term	**Definition**
Text-based program	A computer program whose primary interface is through text input through a terminal
Question bank	A programmatically-accessed text file that contains the list of questions
Python	An interpreted high-level programming language

1.4. References

1. Software Requirements Specifications Sample, https://www.cse.msu.edu/~cse435/Handouts/SRSExample-webapp.doc
2. What is Python?, https://www.python.org/doc/essays/blurb/
3. Software Requirements Specification, https://snappify.com/blog/software-requirements-specification-sample

2. Overall Description

2.1. Product Perspective

The system is a requirement for the completion of the CMSC 202 course. This allows the students to showcase their understanding of the different programming concepts and their knowledge in using Python as a programming language.

The system to be developed is an exam reviewer system that essentially mimics an actual exam. It will show students questions from a pre-compiled list and will wait for their input providing immediate feedback on correct answers and accuracy. The students' scores on the mock tests can then be saved and analyzed to help them focus on areas for improvement.

2.2. Product Functions

The system is expected to have the following as its baseline capabilities:

1. Ability to store questions from multiple topics.
2. Ability to modify the question bank.
3. Provide immediate feedback on student response to each question.
4. Customization on the exams such as topic filtering and question-type filtering.
5. Customization of exam structure such as adding time limits and other gamification concepts.
6. Ability to save current results and provide historical data on student’s scores from previous exams.

2.3. User Characteristics

The target user is someone who has a need for an automated exam reviewer system. He is also expected to have some baseline knowledge on operating a computer and be comfortable enough in using a text-based user interface.

2.4. Design

The system will primarily utilize the Python programming language for its user interface and data operations. External files dependencies will utilize either .csv or .txt file types.

2.5. Constraints

The following are the currently identified constraints of the system:

1. Question type limited to just True/False or Multiple Choice types only
2. Data storage for the question bank is through text files only
3. Text-based user interface

2.6. User Flow

The expected user journey is below:

1. User runs Python Program
2. User enters exam time limit
3. User enters desired number of questions
4. User selects question type
5. User selects topics
6. Program picks a question from the question bank and display to user
7. User inputs their answer
8. Program provides feedback to user on correctness of answer
9. Repeat 7 and 8 until the exam is done
10. Program displays results of exam
11. Program asks user if they want to save their answers and results
12. User inputs answer to 11
13. Program asks user if they want to do another exam or exit

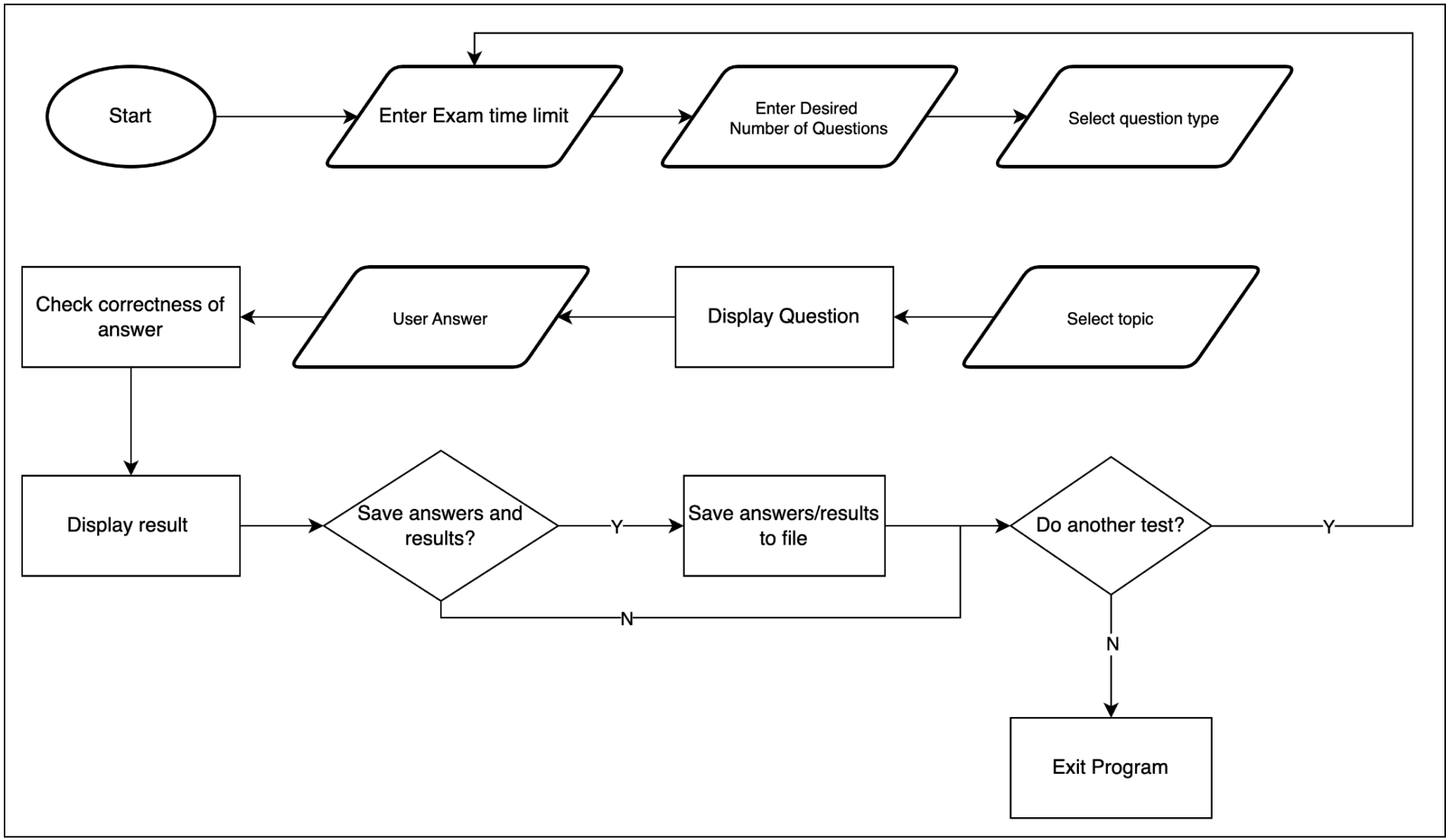

Figure 1. User Journey for the Exam Reviewer Program