CIS-11 Project Documentation Template

Remix
Victor Quezada, Destiny Thap, Saul Resendiz
Test Score Calculator
May 19, 2024

Advisor: Kasey Nguyen, PhD

Part I – Application Overview

Objectives

Why are we doing this?

To elicit the objectives, ask the business expert, the development manager, and the project sponsor the following questions:

- What business objectives of the company will this project help achieve? Possible objectives might be reducing costs, improving customer service, simplifying the workflow, replacing obsolete technology, piloting a new technology, and many others. Also, make sure you understand exactly how the proposed project will help accomplish the stated objective.

The project can reduce the time and resources needed for manual work by automating the manual stuff onto a calculation-based test score, lowering the costs of manual calculation. Customers may receive quick and accurate test results from the calculator which will improve the entire experience and happiness. Increased client retention and loyalty may result from this. By simplifying the process of computing exam results and eliminating the time-consuming nature of manual work, the automatic calculator can free up time.

- Why are we doing this project now? What will happen if we do it later? What if we do not do it at all?

The manual method we currently use to calculate tests takes lots of time, is prone to errors, and uses a lot of resources just to do that. This procedure can be automated to cut down on labor and lower the possibility of mistakes. We must remain competitive by offering an innovative and effective method of accessing exam results since more and more students need online services. Delaying the project would mean falling behind our competitors and in general just losing business.

- Who will benefit from this project? Do the people who will benefit from it consider it the most important improvement that can be made at this time? Should we be doing a different project instead?

Students/clients will benefit from faster and more accurate test scores. It is a worthwhile improvement as it establishes a drastic increase in efficiency and accuracy for student test results, which is a very important aspect in most students' lives and will help them immensely in viewing their test performances. I wouldn't say it's a major issue to our society but it does make it time efficient for them.

Business Process

In the future, this application can become a more sophisticated tool for calculating and reporting scores based on weighted grades, different categories, and percentage contributions to the final grade. (i.e. quizzes are worth 15% while exams are 85%) To provide a more accurate reflection of student performance, we can implement enhancements such as updating the program (user interface, variable, count, subroutines, etc.), improving grading accuracy, and changing the user's number of input/scores to be their choice. This allows for an even more broad range of flexibility and tests overall performance at the fingertips for students from across the globe to access. This application will serve as a detrimental tool for many students to stay on top of their academic success and be able to track and calculate their test performances on a broad range of spectrums.

User Roles and Responsibilities

Victor: Give an overview of what's going to be done and by when and also help with code and documentation. Weekly meeting with team members to discuss what work they have done.

Destiny: Help with flowchart and documentation, discuss the process of each step that is going to be done, and work on a rough draft breaking down the codes.

Saul: Help with code functionality and proper readability. Help with code logic and debugging or with any issues when necessary.

Production Rollout Considerations

Terminology

- **Grade: represents the value of the student's performance on a test, a specific test that ranges the score from 0-100**
- **Student: the individual taking the test**
- **Avg_grade: the sum of 5 test scores from a student or class that is divided by n time of test score inputted**
- **Min_grade: the lowest score and max grade is the highest score**
- **Test Score data: the data that is stored in the file**

Part II – Functional Requirements

Statement of Functionality

- **Input function**
  - **Prompt user to input five test scores**
  - **The program will output a message asking the user to enter five test scores via the keyboard**
- **Minimum score calculation**
  - **Determine the minimum score of the five test scores**
  - **The program will compare scores and find the smallest value. Store**
- **Maximum score calculation**
  - **Determine the maximum score of the five test scores**
  - **The program will compare scores and find the largest value. Store**
- **Average score calculation**
  - **Compute the average score**
  - **The program will add all entered test scores and then divide by five (the total number of scores)**
- **Determination of letter grade**
  - **Assign letter grades to each test score based on the predetermined ranges**
  - **Conditional branching**
- **Output function**
  - **Display results to the console**
  - **The program will display the above information to the console (minimum score, maximum score, average score, letter grade corresponding to the test scores**

Scope

Phase 1: Gathering requirements and analysis
- **Define requirements for the program (inputs, desired outputs, etc.)**
- **Functionality requirements**

Phase 2: Design
- **Design structure and flow of the program**
- **Pseudocode or flowchart to get a detailed bigger picture**

Phase 3: Implementation
- **Write the code based on the design**

Phase 4: Testing
- **Input validation, correct functions**
- **All specific requirements met**
- **Verify accuracy**

Phase 5: Launch
- **Prepare application for launch**

Phase 6: Maintenance and fixes
- **Based on user feedback and/or business requirements, provide ongoing support, bug fixes, and implement any possible enhancements**

Performance

Our calculation time will be calculated in a matter of seconds. The way this works is that the application has a user interface that when the user inputs 5 of their test scores it will show the minimum, maximum, average, and letter grade based on their 5 test scores. Every time the user puts their test score it is stored and incremented by 1. If there is any score that is below the number 5, it should say "Try again with 5 numbers." This implements input validation for our application and prevents errors such as if someone just decides to input a number lower than 5. Our application will be able to perform efficient test score calculations within seconds and the process should be very easy and efficient, as long as the user using the application knows what test scores to input. After they input their designated test scores and run the program, they should receive the stats regarding their scores.

Usability

The UI is pretty readable and easy to navigate. The response and the error time are fast I would say. Even with simple stuff, it will have some sort of label saying this is what this does.

Documenting Requests for Enhancements

There does come a time when the requirements for the initial release of your application are frozen. Usually, it happens after the system acceptance test which is the last chance for the users to lobby for some changes to be introduced in the upcoming release.

Currently, you need to begin maintaining the list of requested enhancements. Below is a template for tracking requests for enhancements.

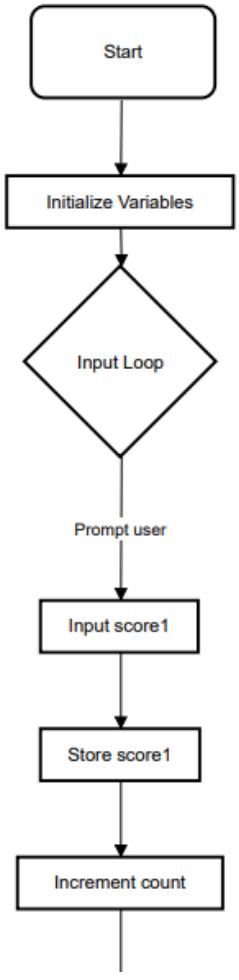
Date	Enhancement	Requested by	Notes	Priority	Release No/ Status
5-18-24	Add a functionality that will display max, min, and avg of 5 test scores and show the corresponding grading score		Grading scale should be followed from A, B, C, D, F	High	
5-19-24	Implement ASCII conversion operations to convert numerical grades to their respective letter grade		Needed to understand the better overview of the grading scale	High	

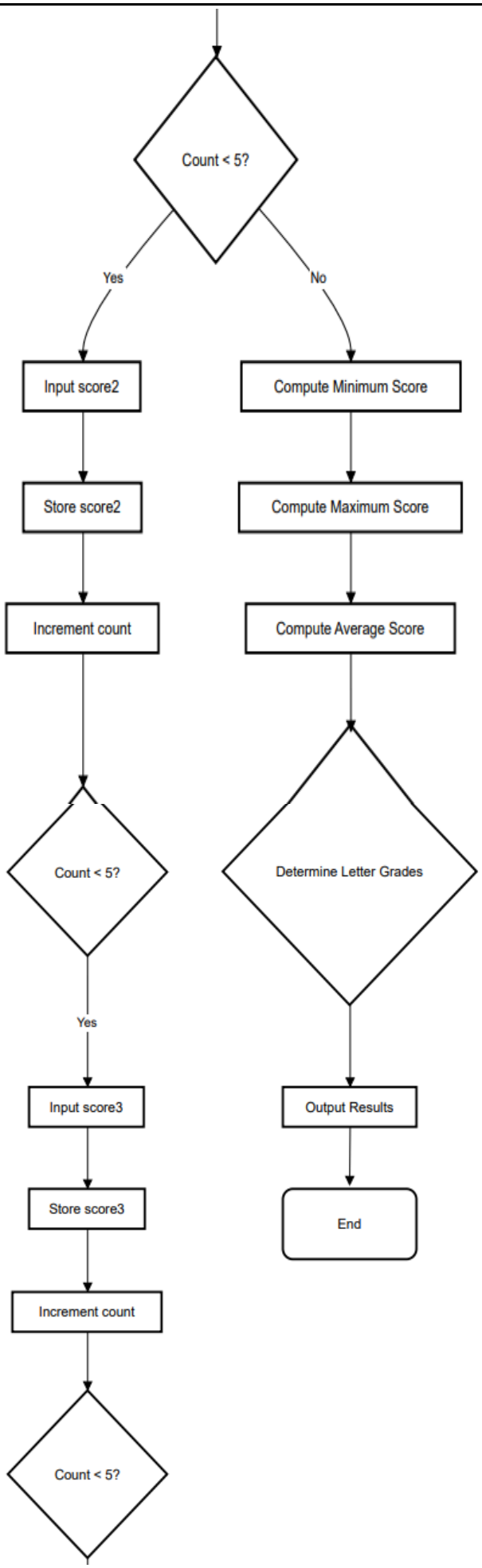
Part III – Appendices

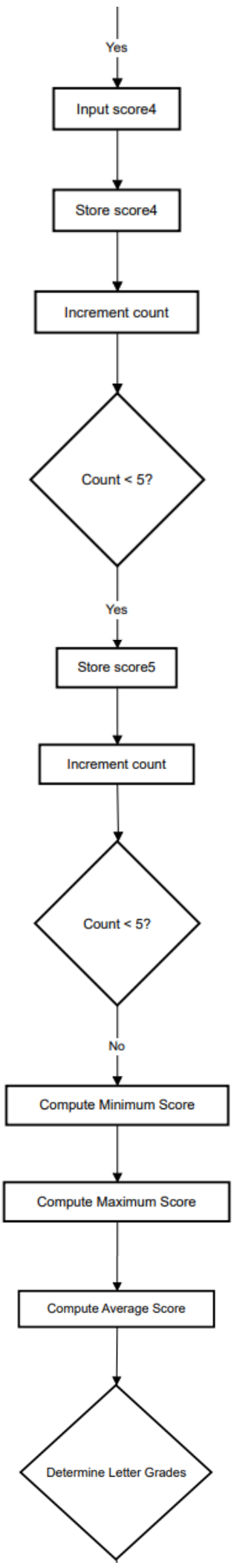
This application being a test score calculator, could be useful in a few ways. It is a simple application that makes calculating even easier. For example, if a student wanted to predict his/her average to pass a class, simply plug in five scores. They could input five existing scores or include predicted one/s to find out what their average would be. Students may incorporate the usage of this application to receive an accurate clarity of their test performance(s) in a quick and easy time efficient manner.

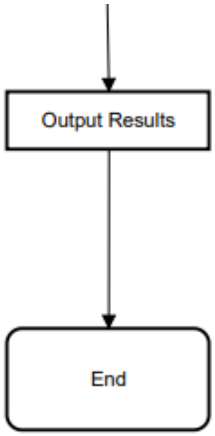
Flowchart

Include branching, iteration, subroutines/functions in flowchart or pseudocode.




Count < 5?
Yes
No
Input score2
Store score2
Increment count
Count < 5?
Yes
Input score3
Store score3
Increment count
Count < 5?
Compute Minimum Score
Compute Maximum Score
Compute Average Score
Determine Letter Grades
Output Results
End


Yes
Input score4
Store score4
Increment count
Count < 5?
Yes
Store score5
Increment count
Count < 5?
No
Compute Minimum Score
Compute Maximum Score
Compute Average Score
Determine Letter Grades


Output Results
End