Test Scores

**Team CIS Maestros**

**Jaspreet Singh, Alejandro Valenzuela, Kihwan Ryu**

**Case B: Test Score Calculator**

**May 25th, 2025**

**Advisor: Kasey Nguyen, PhD**

# Part I – Application Overview

### This part of the requirements document serves to present the “big picture” of the application. Here you lay out the objectives of the application, how it fits into the business process of the company, and how it relates to other software systems.

### The sections listed below should be included in this part of the requirements document.

## Objectives

### In this section you state the commonly accepted objectives of the project.

### You must determine the business objectives of the project early on; without clear objectives your project has little chance of succeeding anyway so it does not make sense to move on until the objectives are agreed upon.

### Create a flawless and efficient calculator with test results or scores being the primary target in mind.

### Ensure calculator is capable of handling and receiving up to 5 values as input.

### Exhibit the maximum, minimum, average, and letter grade associated with or of the inputted data set.

### Utilize subroutines, branching, and other LC-3 functions in abundance but with precision and appropriately.

#### 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 the 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.**
* **Why are we doing this project now? What will happen if we do it later? What if we do not do it at all?**
* **Who will benefit from this project? Do the people who will benefit from it consider it the most important improvement that can possibly be made at this time? Should we be doing a different project instead?**

### Makes the life of educators that grade exams, tests, and other assessments much easier.

### Simplifies the process for students to view their overall performance on major assessments.

### Helps inform students, educators, and parents on what subjects need improvement or more focusing on.

### Delaying or canceling this project altogether will only impede the academic progress of students and unnecessarily complicate the tasks of a teacher.

## Business Process

### In this section you describe the business process and how your application will be used in this context.

### In some cases you will need two sections of this sort – one describing the existing business process using existing systems and the other describing the future business process using the system you are developing. This happens any time the business process changes once your system is introduced. When this is the case, the purpose of describing the existing business process is to have a basis of reference to explain the new business process.

### If the business process won’t change when your application is introduced, you should be able to describe it in a single section. However, in this case be sure you understand and communicate to others what value your application brings to the customers. This question should be answered in the Objectives section.

### Currently, teachers have to insert test scores and then either manually or separately calculate the minimum, maximum, averages, and the letter grades that belong to the inputted test scores.

### Moreover, as of right now, students cannot view their test scores in a cumulative way like our project displays, and it’s a hassle for them to see how their academic progression is coming along.

### In comparison, once our project is introduced and fully implemented, 5 test scores only have to be inputted, and all the things mentioned are showcased immediately and efficiently, easing the lives of both students and teachers.

## User Roles and Responsibilities

### In this section you describe who the users are and how the system fits into what they do.

### You need to list all users for your system in terms of user roles. Typically each individual performs multiple roles in the course of his work since his job involves meeting multiple business objectives. A user role is related to meeting a specific business objective. When gathering requirements it is most useful to consider roles since you will want to focus only on those business objectives that are relevant to your application.

### For each role you need to list the tasks that involve the use of your system (directly or indirectly). You also need to describe the relationships among the tasks for each individual user role and the hand-offs from one role to another. This is usually represented as a workflow diagram.

### Consider time in describing tasks and their relationships – different sets of tasks may be performed at different times (daily, monthly, etc.) and several workflow diagrams may be needed.

### Once you have written the Objectives, Business Process, and the User Roles and Responsibilities sections, give them to the business expert to read. If you and he agree on what’s written, congratulations! You are well on your way to understanding what needs to be done.

### Jaspreet Singh (member of the CIS Maestros):

### Complete project documentation part 1 ‘Application Overview’.

### Assist with Pseudocode.

### Project Coding.

### GitHub Creation.

### Alejandro Valenzuela (member of the CIS Maestros):

### Complete project documentation part 2 ‘Functional Requirements’.

### Assist with Pseudocode.

### Project Coding.

### Kihwan Ryu (member of the CIS Maestros):

### Assist with Pseudocode.

### Project Coding.

## Production Rollout Considerations

### In this section you describe the strategy for production rollout.

### In addition, either this section, or an appendix in the requirements document, or a separate document should include the discussion of populating the system data for rollout and the discussion of the expected data and transaction volume.

### Advertise to schools as a whole, but especially educators teaching classes that require more major assessments (up to 5 values acceptable).

### Promote to all students, but especially those that are evidently struggling but do not seem to know their strengths and weaknesses (minimums and maximums).

### Inform parents that are confused about why their children are struggling with certain subjects despite completing homework or other trivial assignments.

## Terminology

### In this section you define the business terms used in the requirements document.

### You should include this section even if at first it seems like a waste of everyone’s time. Once you show it to people you may be surprised to learn that not everyone understood the terms the same way after all!

### Minimum: the least or smallest amount or quantity attained.

### Maximum: the greatest or highest amount or quantity attained.

### Pseudocode: a detailed yet readable description of what a computer program or algorithm should do.

### Subroutines: a set of instructions designed to perform a frequently used operation within a program.

### Branching: the mechanism by which the program's execution can deviate from the sequential order of instructions.

### LC-3: an educational programming language primarily used to teach fundamental concepts of computer architecture and programming.

# Part II – Functional Requirements

### This part of the requirements document states in a detailed and precise manner what the application will do.

### Functional Requirements: This program will prompt the user to input 5 values from 50 - 100 (any input less than 50 will automatically default to an "F" grade). The program will then calculate the Maximum, Minimum, and overall Average values based on user input. Finally, this program will output these values for the user convenience.

## Statement of Functionality

### In this section you state **precisely** what the application will do.

### This part, more than anything else in the requirements document, spells out your contract with your customers. The application will **include all functions listed here and will not include any of the functions not listed**.

### In this section you must use as precise language as you can since the developers will use it to code the application. When reviewing this part with other people you should pay extreme attention to removing any possibility for ambiguous interpretation of any of the requirements.

### If your application has several distinct categories of users, you can list the requirements by user category. User categories may be defined in terms of their job title (clerk, manager, administrator), the frequency with which they will use the system (heavy or casual), the purpose for which they will use the system (operational decisions, long-term decisions), etc. If each category of users uses the system in its own way, structuring the requirements based on user category will make sense.

### If your application deals with several kinds of real-world objects, you can list the requirements by object. For example, for a reservation system a booking is an important object, and you may want to list all requirements pertaining to bookings in one sub-section.

### One of the most common approaches is to list the requirements by feature. For example, features of a word processing application are file management, formatting, editing, etc.

### This program will be able to take in at least 5 double digit user input(s) with each input valued from 0 - 100. This program will able to categorize these values to an associated letter grade depending on the grade range (0-59 = F, 60 - 69 = D, 70 - 79 = C, 80 - 89 = B, 90 - 100 = A). The program will also compute the Maximum, Minimum, and Average value of the user inputs, respectively.

## Scope

### In this section you state what functionality will be delivered and in which phase.

### You should include this section if your development consists of multiple phases. As an alternative to this section, you can note the planned project phase for each feature in the functionality statement section. Usually, it is better to include a separate scope section for easy reference and communication.

### Phase 1 - Prompt the user for input and store that input to associated memory locations or registers

### Phase 2 - Handle user input appropriately and assign letter grades to individual values

### Phase 3 - Begin computations to calculate the Maximum, Minimum, and Average values based on user input

### Phase 4 - Output Letter Grades and Program totals for Maximum, Minimum and Average values to user

## Performance

### In this section you describe any specific performance requirements.

### You should be very specific and use numeric measures of performance. Stating that the application should open files quickly is not a performance requirement since it is ambiguous and cannot be verified. Stating that opening a file should take less than 3 seconds for 90% of the files and less than 10 seconds for every file is a requirement.

### Instead of providing a special section on performance requirements, you may include the relevant information for each feature in the statement of functionality.

### Not Applicable (included in Statement of Functionality)

## Usability

### In this section you describe any specific usability requirements.

### You need to include this section only if there are any “overarching” usability goals and considerations. For example, the speed of navigation of the UI may be such a goal. As in the previous section, use numeric measures of usability whenever possible.

### Not Applicable

# 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** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Part III – Appendices

### Appendices are used to capture any information that does not fit naturally anywhere else in the requirements document yet is important. Here are some examples of appendices.

### Supporting and background information may be appropriate to include as an appendix – things like results of user surveys, examples of problems to be solved by the applications, etc. Some of the supporting information may be graphical – remember all those charts you drew trying to explain your document to others?

### Appendices can be used to address a specialized audience. For example, some information in the requirements document may be more important to the developers than to the users. Sometimes this information can be put into an appendix.

## Flow chart or pseudo-code.

Include branching, iteration, subroutines/functions in flow chart or pseudocode.