CIS-11 Project Documentation Template

**Team Name**

**James Destefano, Angel Flores**

**Test Score Calculator**

**05/25/25**

**Advisor: Kasey Nguyen, PhD**

# Part I – Application Overview

### The purpose of this project is to be able to calculate different test scores. This calculator will show the maximum, minimum, average, and give a letter grade equivalence. This fits school districts as well as online quizzes and test programs.

## Objectives

### This project should be able to calculate the minimum, maximum, and average grade.

### 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.

#### Why are we doing this?

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

### This project will help School districts and online schools to have a fluid and easier way to calculate scores rather than doing everything by hand. The students will receive there results faster, simplifying confusion in the classroom.

### The reason we are doing this project now is because we see the importance in having communication in a classroom between students and teachers.

### This is something students would like to be prioritized because they want to now their grade as soon as possible.

## Business Process

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

### A existing system to calculate scores already exist for big districts at a higher price, or even sometimes Professors/Teachers will do the calculations by hand per student which takes too long. Our application would help small districts and/or smaller classrooms that do not need a big and costly application. The business process will change because instead of having buy/ rent a app for a big classroom, this will fit a small classroom's needs.

## 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.

## 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.

## 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!

# Part II – Functional Requirements

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

### This program will include several parts. These include obtaining five test scores with a range from 0 to 100 as inputs from the user, and finding which was the lowest score, highest score, and the average score. The program should also display these scores as a letter grade.

### To find which score is the maximum, we will hold the first test score in a register and compare this test score with each other test score swapping it for any test score that is higher.

## 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.

## 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.

## 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.

## 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.

# 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.