CIS-11 Project Documentation Template

Team Name Team Something
Team Members Sadia Plimley & Hugo Ngo
Project Name Test Score Calculator
Date May 25, 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.

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?**

The company holds the following business objectives: to provide convenience and transparency to its customers. This project will help achieve those objectives by giving customers full transparency of how they score in relation to others and gives them the convenience of seeing their letter grade without having to calculate it themselves. We are doing this project now as finals season is rapidly approaching. High school students need to know their grades and how they do on their previous tests in relation to others to see how they should be studying for their finals and to eventually understand what they earned on their final. If we do it later, it will be only half as helpful, as only college students will be taking exams during summertime. If we do not do it at all, students everywhere will be sacrificing their energy they could be using on schoolwork to calculate their test grades. Students will be benefitting greatly from this project. Out of all three choices, students consider this to be the project of utmost importance and a much greater choice over the other two.

Business Process

Currently, customers must calculate their own test grades as teachers have widely abandoned putting letter grades on their exams. There is also no service for finding the averages of one's test scores, leading many students to be unaware of their overall work.

In the system we are developing, students will enter test scores (which may all belong to them or include scores from their friends) and see how the scores match up with each other and their grade overall.

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.

User Roles and Responsibilities

Student (to learn):

- Finding lowest score helps student learn what test they must study the hardest on OR helps students learn who in their group needs the most help
- Finding highest score helps student know which content they know best OR shows them who can help them the best
- Student can use the Letter Grade function to see how they did overall
- Student can see their general average score (to see how they do in a class overall) OR see the average score among their friends/peers.

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.

Teacher (to teach):

- If teacher decides to include Letter Grades on exams, they can use the Letter Grade function.
- Can find class average and know how well they taught their content
- Can find the student with the weakest score and reach out and help them
- Can find student with strongest score and ask them to help their peers

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

---

The strategy for production rollout is to break the program into smaller pieces/functions (Letter Grade, Average, Lowest Score, and Highest Score) and have the company members work together on building the code for each task, polishing it together before moving onto the next function.

System data will be populated through utilizing past test scores of the project engineers.

for rollout and the discussion of the expected data and transaction volume.

```
Function: a program that
achieves one task and
can be used multiple
times in a larger
program

Finals: A large
test/project that
represents a student's
overall knowledge from
an entire year of class
and significantly
influences a student's
grade.
```

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.

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.

The features of the test grade calculator application are finding the average of five test scores, finding the lowest of five test scores, finding the highest of said test scores, and returning the letter grade of each test score to the user. All five test scores will be inputted by the user.

Scope

Phase I: pseudocode (written in javascript) will represent all functions. All functionality will be delivered-but in a separate language.

Phase II: Pseudocode will be converted into Assembly language roughly. Some functionality will be available, but there will be bugs.

Phase III: Project is fully functional and provides users the ability to insert five test scores and see the highest, lowest, the average, and the Letter grade of each.

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.

The application should return the highest score, the lowest score, the average, and the Letter Grades within ten seconds of inputting the five test scores.

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.

The input of five test scores must be obvious and must be understood by the users within five seconds of running the application.

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
05/20	Quicker program	Sadia	Application stalls	Medium	Not Yet

Part III – Appendices

```
The application may solve
the following problem:

A student wishes to study
for their comprehensive
end-of-year exam but does
not know which subject to
start with as it has been
a long time since they've
taken all of their tests.
They find their test
scores for every test but
does not know which score
is the lowest.
```

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.

PSEUDO CODE ON NEXT PAGE

```
// collect grades
// Create empty lists grades and letterGrades
// Create a loop that takes in user responses for five grades
// Add all grades to a list
// Use four subroutines to find the min, max, average, and letter grades

Var grades = [];
Var grade = 0;
Var letterGrades = [];
Var letter = "";
For (var i = 0; i < 5; i++) {
   Grade = 0;
   Grade = prompt("Insert a grade.");
   Grades.push(grade);
}
FindMin(grades);
FindMax(grades);
FindAvg(grades);
LetterGrade(grades);

Function FindMin(grades) {
  Var min = grades(0);
  Var B = 1;
  For (var I=0;I<5;I++) {
   If (grades(A) > grades(B)) {
     Min = grades(B);
   }
   B++;
  }
  Return(min);
}

Function FindMax(grades) {
  Var max = grades(0);
  Var B = 1;
  For (var I=0;I<5;I++) {
   If (max < grades(B)) {
     Max = grades(B);
   }
   B++;
  }
  Return(max);
}

Function FindAvg(grades) {
  Var sum = 0;
  Var x = 0;
  For (var i=0;i<5;i++) {
   sum = sum + grades(x);
   x++;
  }
  Return sum / 5;
}

Function LetterGrade(grades) {
  Var x = 0;
  For (var I=0;I<5;I++) {
    Letter = "";
    If (grades(x) <= 59) {
     Letter = "F";
    } Else if (grades(x) <= 69) {
     Letter = "D";
    } Else if (grades(x) <= 79) {
     Letter = "C";
    } Else if (grades(x) <= 89) {
     Letter = "B";
    } Else if (grades(x) > 89) {
     Letter = "A";
    }
   LetterGrades.push(Letter);
    X++;
  }
  Return LetterGrades;
}
```