# *CIS-11 Project Documentation Template*

**Team Cooked**
**Lindzy Tyler Marie So**
**Isaac Hernandez**
**Nour AlDeen Hajjar**
**Test Calculator**
**June 6th 2024**

**Advisor: Kasey Nguyen, PhD**

# Part I – Application Overview

*The test score calculator is designed to calculate 5 test scores input by the user. To increase the efficiency of educators and students in calculating their minimum and maximum test scores, average grade score, and their letter grade.*

## Objectives

*This test calculator's objective is to increase time efficiency of educators across the country by introducing technology and algorithms into calculating test scores. In addition, the evaluation of grades helps upper management scope the curriculum's weaknesses and strengths.*

### Why are we doing this?

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

- **Our goal as a company to facilitate an efficient workspace in the education field as we want to guarantee enough time is being spend to the children. Our main goals are to increase the efficiency of educators as our product will give them immediate feedback and aid in the collection of data and the analysis of it.**
- **As technology begins to integrate itself into each part of the world we saw this as an opportunity to help the education system as this is what builds our future workers. To allow the teachers to use more time working on curriculum over the time used on grading and calculating averages.**
- **Educators across the country will be able to access this test score calculator and will see a change in their productivity. Seeing an increase of time efficiency and their ability to set aside more time in planning a curriculum best shaped for the students.**

## Business Process

*Without our test calculator educators are only able to calculate the minimum, maximum and average test scores manually taking a lot of time away from their planning days. With our Test Score Calculator educators are able to input five test scores and can receive immediate feedback*

## User Roles and Responsibilities

*For our Test Calculator we have different users in mind the educators, the students, and the administration each holding their own separate responsibilities with using this product.*

*For the educators their responsibilities are to input the five test scores into the calculator and analyze the given data changing or forming the curriculum as they go best suited for the students. This product could be used daily or throughout the week, when necessary, by the educator increasing efficiency as less time will need to be allocated for grading.*

*For the students, their responsibility is to input the five test scores and analyze their given date to shape their study habits to attain or maintain their desired score. This product can be always used by the student to increase their knowledge of their score and to help steer them in the right path for their studying habits.*

*For administration, their responsibility is to take the data received from the Test Score Calculator and see what as a school can they gain insight of students' performance. Allowing them to understand what they can do as a school to maximize their students' time and education.*

## Production Rollout Considerations

*In our rollout we work to create product that caters to the users and will use any feedback to form an algorithm that will increase their efficiency. Our first step to the product rollout will be the initiation phases which is where we will discuss our man objectives of the product and the roles it will have in the field of launch which is education. Next, we will roll into our design phase creating the code to function in the ways we present and fixing any errors. Then we will work through the pilot phase, where will test the program with a random group surveying them and improving the product in relations to their feedback. Finally once all the changes are finalized we will proceed with the last phase the rollout phase, launching the product to the public.*

## Terminology

*Input validation: the process of ensuring the input of the user is valid in order to properly calculate the scores of the students.*

*User interface: the portion of the code that is visible to the user. The interface the users will interact with during their use of the code.*

## Part II – Functional Requirements

1. The software will provide a user-friendly interface to input five test scores.

2. From the user input the minimum, maximum, and average values will be displayed.

3. The software will ensure data accuracy through validation checks and error prevention mechanisms during data entry and processing.

4. The software will assign letter grades using the following scales:

   0-59 = F<br>
   60-69 = D<br>
   70-79 = C<br>
   80-89 = B<br>
   90-100 = A

5. The software will display the calculated results promptly after receiving user input.

## Statement of Functionality

*The user will input exactly five test scores. The system will validate inputs to ensure they are within the range of 0-100. The system will not accept more than five scores per entry. The system will calculate minimum, maximum, and average scores from the provided inputs. The system will compute the corresponding letter grade based on each score entry. The system will not calculate statistical measures beyond, minimum, maximum, and average values. The system will display the result immediately upon calculation on the console. The display will include minimum, maximum, average values, and letter grades for each test score. The system will not correct input errors.*

*For educators, administrative staff, and IT support*

- *Can input and view results of test scores.*
- *Cannot alter scores once they have been entered.*
- *Can access analytics for classes.*

*The functionalities listed here define precisely what the Test Score Calculator program will deliver clarity and provide developers and customers with a definite guide to follow regarding the system's capabilities and limitations.*

.

### Scope

*During the rollout phase all functionalities are going to be available to the public. Educators and students are going to be able to calculate the minimum, maximum, and average scores of five tests they input into the computer. In addition, with the average score the program will return the corresponding grade letter.*

### Performance

*The Test Score Calculator will take the five test scores and will print the maximum, minimum, and average scores with the corresponding grade letter. Ensuring that the user input is valid and responding with a message that alerts them if not. The program can calculate the scores within a second of inputting all five grades.*

### Usability

*The program will be friendly to all users, ensuring all the instructions for what to input are clear to the user. Immediate feedback will be presented to the user labeling each of the scores, so it is clear the user which is the min, max, and average score. Users of all ages will use this product with ease.*

## Documenting Requests for Enhancements

When our initial release requirements are frozen, users will be allocated time to request any enhancements to the product.

| Date | Enhancement | Requested by | Notes | Priority | Release No/ Status |
|---|---|---|---|---|---|
| **05-10-2024** | **Include median score calculation** | **Team Member** | **Adds median calculation to the statistics** | **High** | **Planned for release 1.0** |
| **05-12-2024** | **Improve input validation UI** | **User Feedback** | **Enhance US for better input validation experience** | **Low** | **Under Review by Management** |
| **05-29-2024** | **Add Support for more than 5 scores** | **Beta Testing Team** | **Allow input of up to 10 scores for broader statistics** | **High** | **Planned for release on 1.1** |

## Part III – Appendices

1. .ORIG x3000
2. Load the variables for the loop count of 5 and the base address of the stack
3. Go into label START
    a. Load the address of the prompt into R0
        i. Printing “Enter the first digit of the test score: “
        ii. Get input and display onto screen
        iii. ASCII convert
    b. Load the address of the second prompt
        i. Printing “Enter the second digit of the test score: “
        ii. Get input and display onto screen
        iii. ASCII convert
    c. JSR to MULT
        i. The first digit will be multiplied by ten so we can add the second digit to get the proper test score
    d. Once return to calling program the resulting number from mult will be added to the second digit
    e. Push the resulting test score onto the stack
    f. START will reloop five times
4. Once the start loop is done the program will BRz to the COMPARE SUBROUTINE
    a. Compare will JSR POP to pop the value off the most recent test score out and will store into the address of MINVAL and MAXVAL in order to compare the rest
    b. Then it will go into the COMPLOOP
        i. This will pop the score off the stack misusing the most recent popped score with the current min and max val (if statement)
        ii. If input – min is negative the program will BRn to SETMIN and store the current score in the address of MINVAL
        iii. If input – max is negative the program will BRp to SETMAX and will store the current score in the address of MAXVAL
    c. Once it loops through COMPLOOP 4 times the program will BRnz to CALCAVG
5. CALCAVG
    a. Will take the total of all the scores and will jump to subroutine DIVISION
        i. Where the number will be divided by five and the quotient will be stored to the address of AVERAGE
    b. Then the program will JSR to DISPLAY
6. DISPLAY
    a. LEA R0, MAXMSG and display
        i. Printing “MAX SCORE: “
    b. From here the program will load the value of MAXVAL and will jump to the DISPLAYDIV subroutine
        i. Where the number will be divided by ten
        ii. The program will take in account of the quotient and the remainder
    c. The quotient and remainder will be ASCII converted and the program will output the quotient and then the remainder in order to print the test score properly
    d. This same method will repeat for printing the “MIN SCORE: “ and its value, the “AVERAGE SCORE: “ and its value
    e. Once the subroutine reaches to the end it will JSR to the GRADE subroutine
7. GRADE
    a. Grade will take the average score and load it to a register

b. Print the “GRADE: “ string by loading the prompt into the register and outputting onto the console

c. From there the average score will be subtracting by decimal 59

  i. If negative or zero it will BRnz to FGRADE where the grade F will be outputted to console

d. Resulting number will be minused by 10 again

  i. If negative or zero it will BRnz to DGRADE where the grade D will be outputted to console

e. Resulting number will be minused by 10 again

  i. If negative or zero it will BRnz to CGRADE where the grade C will be outputted to console

f. Resulting number will be minused by 10 again

  i. If negative or zero it will BRnz to BGRADE where the grade B will be outputted to console

g. Resulting number will be minused by 10 again

  i. If negative or zero it will BRnz to AGRADE where the grade A will be outputted to console

h. Once done the program will halt

8. Load all the stringz and values
9. .END