Project Assessment Guidelines

**Scenario: Case Study**

Glasgow Clyde Runners Club currently hold finals for the following races:

* 400m
* 800m
* 1500m

To reach the finals each runner would have taken part in a series of heats held across several different local clubs. There are 8 runners competing in each race. There are currently 2 finals held in any given year.

Once the final races have been run Glasgow Clyde Runners Club must record the following information for each competitor

* First Name
* Second Name
* Time taken to run their race (recorded in seconds)

Although this has been good enough up until now the club has recently decided that they would also like to record and store information such as:

* Slowest recorded time
* Fastest recorded time
* Ordered list of times from slowest to fastest
* Search for a time in a particular race and the number of times this was achieved.

At present everything recorded is done using a text-based system which makes the possibility of achieving the additional recording very difficult. They would like to move everything over to a software system which would allow the Glasgow Clyde Club Administrators the option to carry out this additional reporting with ease.

They would like the application to be user friendly and include a menu interface that allows the user to make choices based on the actions they would like to take. They feel this would make it much easier for their non-technical staff to use the software.

They would also like the name of their club to be displayed in the program.

They would like the results to be printed to a text file that can be accessed by members of the Glasgow Clyde Runners Club administration team as well as a suitable message being output to the user when using the software.

They would like the software to be robust and able to run on the hardware available without having to upgrade. Therefore, they would like the software to be as lightweight as possible. However, they are willing to upgrade the hardware if it is deemed necessary but would like this to be discussed at a later date.

They would also like the application to be password protected so that only members of the administration team can access it.

They would like the development to begin on February 01st 2024 and completed by March 31st 2024. They have allocated a budget of £10,000 for the entire project.

Before developing an entirely new system they would like you to produce a trial program based on their last 2 recorded 400m final races. If this is successful, then they would look to expand this to include both the 800 and 1500m finalists.

**Additional Information**

As part of an earlier investigation and development process another developer did begin the development by creating a unit of code based on the password requirement mentioned above.

Unfortunately, the developer was unable to continue but did inform us that we could if we wished pass on this code to the next developers so that they may integrate this into any future development.

We will supply you with the source code for the password solution and ask that this be integrated into the proposed future development.

**Instructions:**

Based on the scenario above you must:

**Analysis and Design:**

Apply the Waterfall approach to the given scenario and focusing on the Analysis and Design phases, your task is to produce the following:

**Requirement Specification Report to include:**

* Overview of the project (What have you been asked to do)
* A list of Functional and Non-Functional Requirements for the whole system.
* Test Plan (How do plan to test your program i.e., black box / white box etc.)
* Project Plan (Main Tasks / Sub Tasks / Timescales / Milestones)

**Design:**

* User Interface Design using a suitable method i.e., Storyboard or Wireframe. In the case of a Console based application then your Login / Menu interface can be used.
* Data Dictionary for the new system listing all the data items, their format, and a brief description of their content.
* An overall design using Structure Chart. This should show the breakdown of the system into its lowest levels.
* Detailed design of the system using pseudocode which explains the steps in an English-Like format.

**Implementation:**

You must implement a software solution for the given scenario.

The implemented solution should follow the following structure as a minimum:

* **Login**

The user must login into the system using a validated password. For the purpose of the trial program the password will be: ‘clyderunners’.

As this code is being supplied by another developer you must first unit test it to ensure that it works before integrating it with the rest of the program as part of the integration testing. (The source code for login will be supplied to you by your assessor).

* **Menu**

The user will be presented with the following menu:

1. **Read and Display File** (the file will be the text file supplied to you by Glasgow Clyde Runners and will include the runner’s names and times for the 400m finalists) The data in the file should be stored using a 2D array and printed back to the user.
2. **Sort and Print Recorded Times**

Using a simple sort algorithm, you must sort the recorded times from slowest to fastest and output the result to anther text file using a suitable name etc. you should also output the result to the user.

1. **Find and Print Fastest Time**

Using the min algorithm, you must find the fastest recorded time and output the result to another file using a suitable name etc. you should also output the result to the user.

1. **Find and Print the Slowest Time**

Using the max algorithm, you must find the slowest recorded time and output the result to another file using a suitable name etc. you should also output the result to the user.

1. **Search**

Ask the user to enter a time (seconds) and using a linear search algorithm see if this time has been recorded for the 400m race. You must output the result to another file using a suitable name etc. you should also output the result to the user.

1. **Time Occurrence**

Ask the user to enter a time (seconds) and using the occurrences algorithm see how many times this time has been recorded for the 400m race and output the result to another file using a suitable name etc. You should also output the result to the user.

1. **Exit Program**

The program should exit, and the user should be given a suitable message.

**The above menu should be available to the user until they wish to exit the program.**

**You will then upload your solution to canvas in the following format:**

* **Zipped project folder (object code)**
* **Document with your full source code**

**Testing:**

**To satisfy the criteria needed you must produce a report which outlines the following:**

While you were implementing your program, provide evidence to show that you used the following test levels:

* unit testing (password code)
* integration testing (password code and menu etc.)
* system testing (entire program: Functional System Test)

You must complete a test log sheet which includes:

* Test Data (including normal, extreme, and exceptional test cases where applicable)

You must provide evidence that shows that you have implemented the following debugging techniques:

* Dry run (trace table for at least one algorithm)
* Walkthrough (this can be done when you demonstrate your program to your lecturer)
* Break Point (show screenshot of you using a break point to debug your program)

**Note: A Testing document outline will be available on canvas.**

**You will then upload your Testing Document to canvas.**