**Testing plan:**

**Project Name: Glasgow Runners Club Program**

**Testing Methods Explained**

**White box testing:**

White box testing is a method of testing where I, the programmer has working knowledge of the code and so can design a plan in which I know what the outcomes should and should not be. White box testing in the test method used for this program and I will be the one performing the tests.

**Unit testing:**

Unit testing is the method of testing each of the individual methods, functions, and algorithms individually so that all the code can be validated as working correctly before beginning integration testing. I will run these tests before I integrate the supplied password code into the larger system.

**Integration testing:**

Integration testing is the method of testing your larger combined code as a group. I will run these tests after merging the password and choice menu parts of my code.

**System testing:**

System testing is the method of testing the full working system after all code has been combined. Using system testing I will run through the entirety of the code, making sure it has all required functionality as per design specifications.

**Black box testing:**

Black box testing is the method of testing in which the tester has no prior knowledge of the inner workings of the code. For this program, our tutor will be taking the place of the black box tester.

**Acceptance (functional requirement by the Client):**

Acceptance testing is the method of testing performed once the development has been completed and is used to validate the program meets all functional requirements.

**Functional requirements:**

The program must have a login page with a password for admins.

The program will need to take inputs from the user

The program must have a menu interface.

The program must display users choices once selected

The program must be able to read a text file

The program must be able to create a new text file

The program must be able to write to a text file

The program must read and display the information from a text file

The program must sort the information from a text file and print recorded times

The program must find and print the fastest time based on information from a text file

The program must find and print the slowest time based on information from a text file

The program must take user inputs and using that, search for a time in a given race and the number of times this was achieved and print the results.

The program must take user inputs and use that to count occurrences based on information from a text file

The program must be able to exit the program

As client has paid for previous work to be done, they have supplied some worked on code to be integrated into our trial software.

**Usability (non-functional requirements by the client)**

Similarly, usability testing is the method of testing the programs non-functional requirements. For example, making sure the language is correct and that the club’s name is displayed, how lightweight the program runs and its ease of use.

**Non-functional requirements:**

Client would like to have club name displayed in the program

Client would like the software to be lightweight and robust

Client would like the software to be easy to use

**Unit tests:**

**Main Function**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| Main function | Valid password | Password = clyderunners | Valid – call display\_menu |  |  |
| Main function | Invalid password | Password = glasgow123 | Invalid - Username or password incorrect – “number” attempts remining |  |  |
| Main function | No input | Password= “ ” | Invalid Username or password incorrect – “number” attempts remining |  |  |

**Get\_userdetails:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| Get\_userdetails | Valid password input | Password = clyderunners | Valid – call display\_menu |  |  |
| Get\_userdetails | Invalid password input | Password = glasgow123 | Invalid - Password incorrect – “number” attempts remining |  |  |
| Get\_userdetails | No pasword input | Password= “ ” | Invalid - Password incorrect – “number” attempts remining |  |  |
| Get\_userdetails | Valid username input | Username= admin123$ | Valid input |  |  |
| Get\_userdetails | username input limit | Username = GlasgowClyde200000 | Valid input |  |  |
| Get\_userdetails | No username input | Username = “ ” | Username = “ ” |  |  |

**Get\_Choice**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| Get\_choice | Valid input | 1 | You have selected choice 1 “choice name” |  |  |
| Get\_choice | Valid input | 7 | You have selected choice 7 “choice name” |  |  |
| Get\_choice | Invalid input | 0 | Display menu. To make a selection type, type a number from 1 to 7 |  |  |
| Get\_choice | Invalid input | 8 | Display menu. To make a selection type, type a number from 1 to 7 |  |  |

**Act\_On\_Choice**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| Act\_on\_choice | Validate switch statement | Case 1 | 1. Read and display file |  |  |
| Act\_on\_choice | Validate switch statement | Case 2 | 2. Sort and print recorded times |  |  |
| Act\_on\_choice | Validate switch statement | Case 3 | 3. find and print the fastest time |  |  |
| Act\_on\_choice | Validate switch statement | Case 4 | 4. find and print the slowest time |  |  |
| Act\_on\_choice | Validate switch statement | Case 5 | 5. search |  |  |
| Act\_on\_choice | Validate switch statement | Case 6 | 6. time occurrences |  |  |
| Act\_on\_choice | Validate switch statement | Case 7 | 7. exit program |  |  |

**ReadFromFile:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| ReadFromFile | Valid reading | Race results.text = John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 | Print - John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 |  |  |
| ReadFromFile | Valid reading | No file passed | Empty array |  |  |
|  |  |  |  |  |  |

**Print\_array**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| Print\_array | Valid reading | Race results.text = John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 | John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 |  |  |
| Print\_array | Valid reading | No file passed | Empty array |  |  |
|  |  |  |  |  |  |

**Sort\_Array**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| Sort\_Array | Valid sort | Unsorted array = 70, 90, 75, 70, 95, 103, 80, 110, 68, 120, 80, 140, 90, 72, 78 97 | 68, 70, 70, 72, 75, 78, 80, 80, 90, 90, 95, 97, 103, 110, 120, 140 |  |  |
| Sort\_Array | Valid sort | Sorted array = 68, 70, 70, 72, 75, 78, 80, 80, 90, 90, 95, 97, 103, 110, 120, 140 | 68, 70, 70, 72, 75, 78, 80, 80, 90, 90, 95, 97, 103, 110, 120, 140 |  |  |
| Sort\_Array | Valid sort | Empty array | Empty array |  |  |

**WriteToFile**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| WriteToFile | Valid write | Unsorted array = 70, 90, 75, 70, 95, 103, 80, 110, 68, 120, 80, 140, 90, 72, 78 97 | Unsorted array = 70, 90, 75, 70, 95, 103, 80, 110, 68, 120, 80, 140, 90, 72, 78 97 |  |  |
| WriteToFile | Valid write | Sorted array = 68, 70, 70, 72, 75, 78, 80, 80, 90, 90, 95, 97, 103, 110, 120, 140 | 68, 70, 70, 72, 75, 78, 80, 80, 90, 90, 95, 97, 103, 110, 120, 140 |  |  |
| WriteToFile | Valid write | Empty array | Empty array |  |  |

**Find\_Min**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| Find\_Min | Valid minimum | Unsorted array = 70, 90, 75, 70, 95, 103, 80, 110, 68, 120, 80, 140, 90, 72, 78 97 | 68 |  |  |
| Find\_Min | Valid minimum | Sorted array = 68, 70, 70, 72, 75, 78, 80, 80, 90, 90, 95, 97, 103, 110, 120, 140 | 68 |  |  |
| Find\_Min | No array passed | Empty array | Empty array |  |  |

**Find\_Max**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| Find\_Max | Valid maximum | Unsorted array = 70, 90, 75, 70, 95, 103, 80, 110, 68, 120, 80, 140, 90, 72, 78 97 | 140 |  |  |
| Find\_Max | Valid maximum | Sorted array = 68, 70, 70, 72, 75, 78, 80, 80, 90, 90, 95, 97, 103, 110, 120, 140 | 140 |  |  |
| Find\_Max | No array passed | Empty array | Empty array |  |  |

**Integration test**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| Password code and choice menu | Valid password | Password = clyderunners | Displays welcome message and choice menu |  |  |
| Password code and choice menu | Invalid password | Password = glasgow123 | Username or password incorrect – “number” attempts remining |  |  |
| Password code and choice menu | No input | Password= “ ” | Username or password incorrect – “number” attempts remining |  |  |

**System test:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit name | Reason | Test data | Expected result | Result | Comments |
| Read and display | Full system test | Password=clyderunners  Choice 1  Race results.text = John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 | John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 |  |  |
| Sort and print recorded times | Full system test | Password=clyderunners  Choice 2  Race results.text = John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 | 68, 70, 70, 72, 75, 78, 80, 80, 90, 90, 95, 97, 103, 110, 120, 140 |  |  |
| Find and print slowest time | Full system test | Password=clyderunners  Choice 3  Race results.text = John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 | 68 |  |  |
| Find and print the fastest time | Full system test | Password=clyderunners  Choice 4  Race results.text = John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 | 140 |  |  |
| search | Full system test | Password=clyderunners  Choice 5  User input = 68  Race results.text = John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 | Chantelle Oliver 68 |  |  |
| Count occurrences | Full system test | Password=clyderunners  Choice 6  User input = 80  Race results.text = John Brown 70  Peter Black 90  Anne Waters 75  William White 70  Betty Davis 95  Colin Davis 103  Natalie Wallis 80  Paul Blue 110  Chantelle Oliver 68  Gavin Brown 120  Elliot Ness 80  Al Capone 140  Richard Smith 90  Callum Dawson 72  Adam Stark 78  Pauline Cook 97 | 80 appears 2 times |  |  |
| Exit program | Full system test | Password=clyderunners  Choice 7 | \*Program closes\* |  |  |