**Design Documentation**

**By**

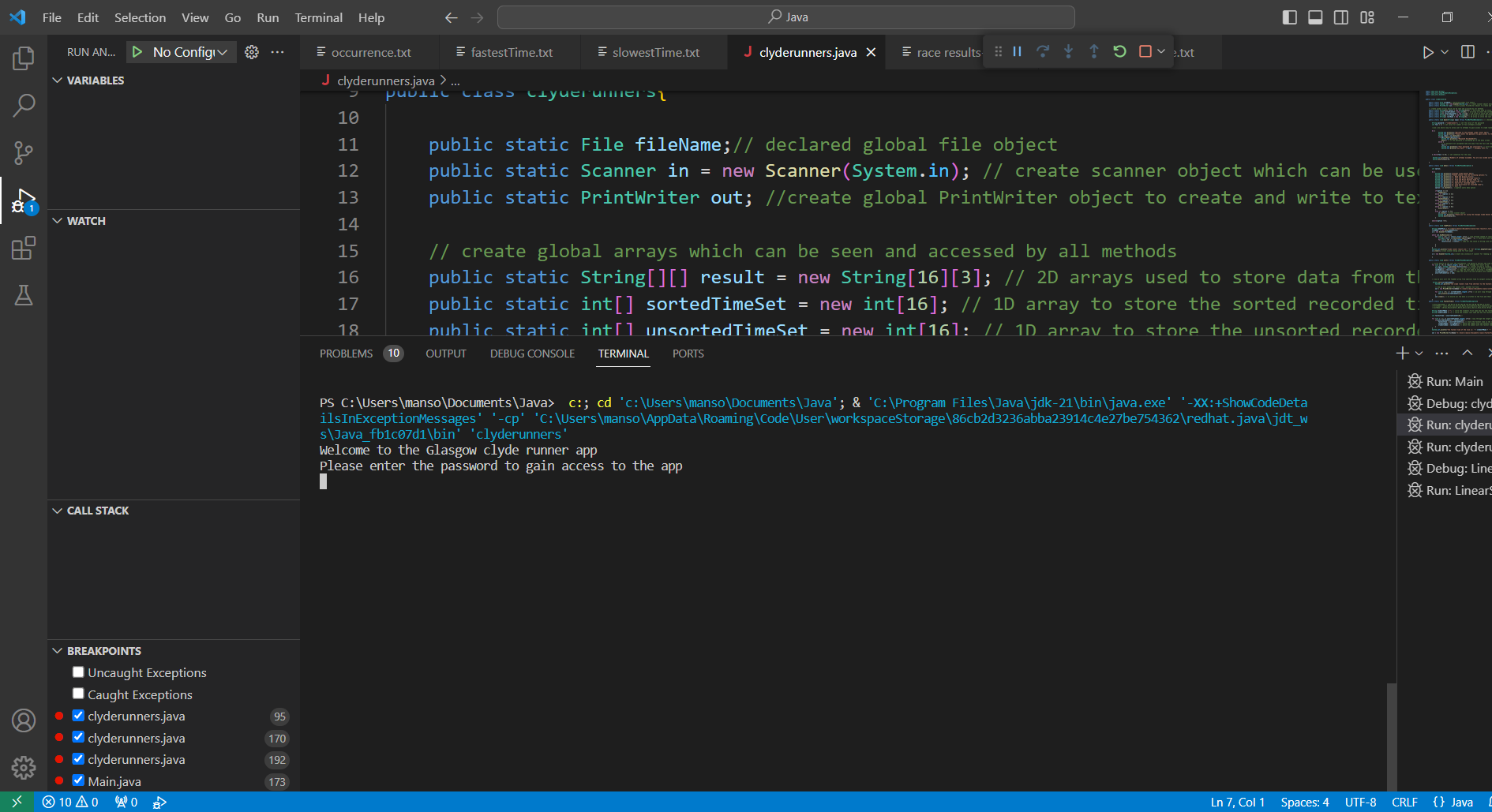
**Mansoor Munawar**

# **Overview**

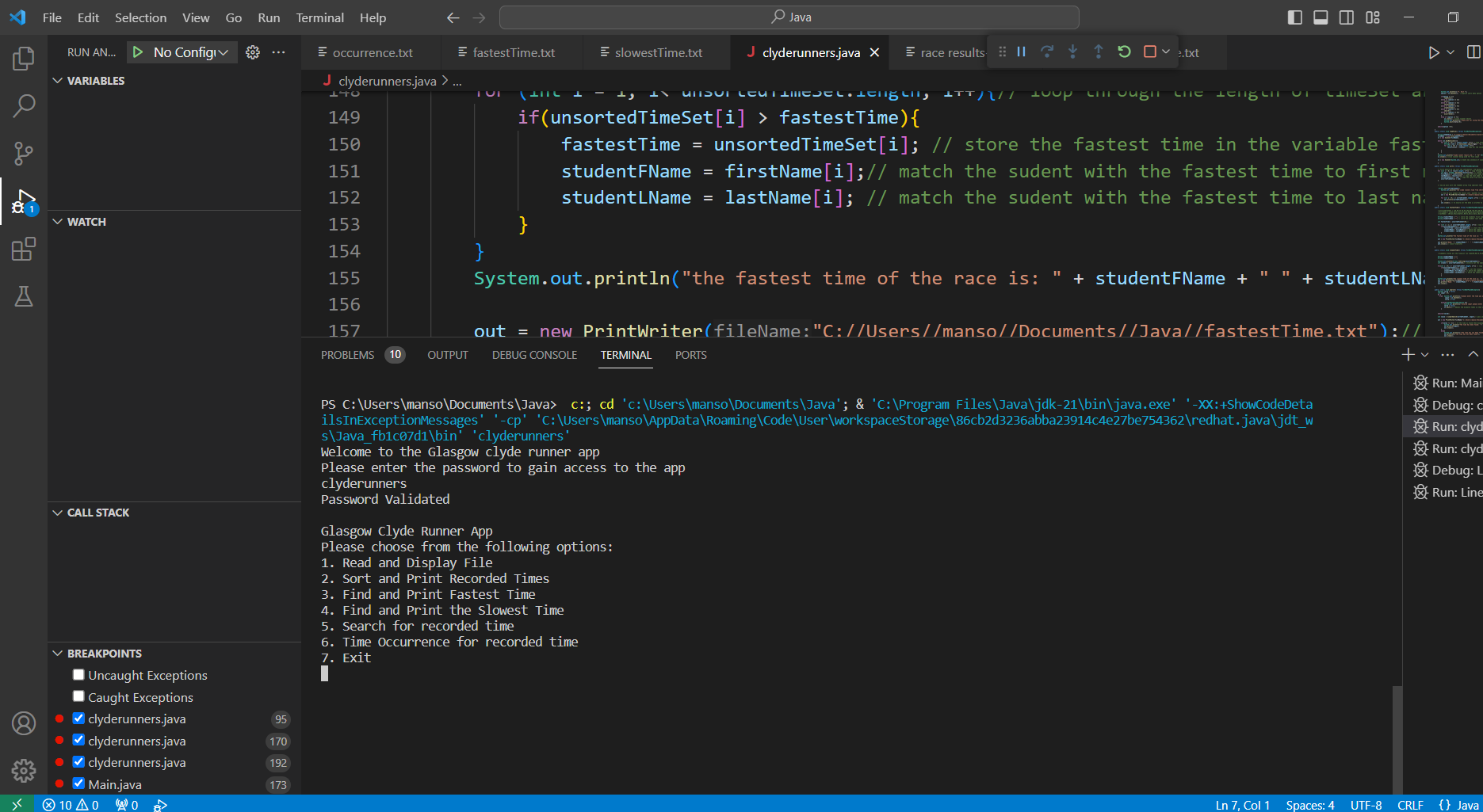
I have been asked to complete design documentation for the Glasgow Clyde Runners App which will include user interface design. The user interface design will include screenshot of the console terminal window that shows the login page and main menu interface from the perspective of the user. I will also include the data dictatory of all the different types of variables, their names and how they will be used within the program. I also need to include a structure chart that shows the program functions from the highest level to the lowest. Finally, I will write an algorithm using pseudocode to breakdown the program using non syntax language

# **User Interface Design**

# The login design:



The menu interface:

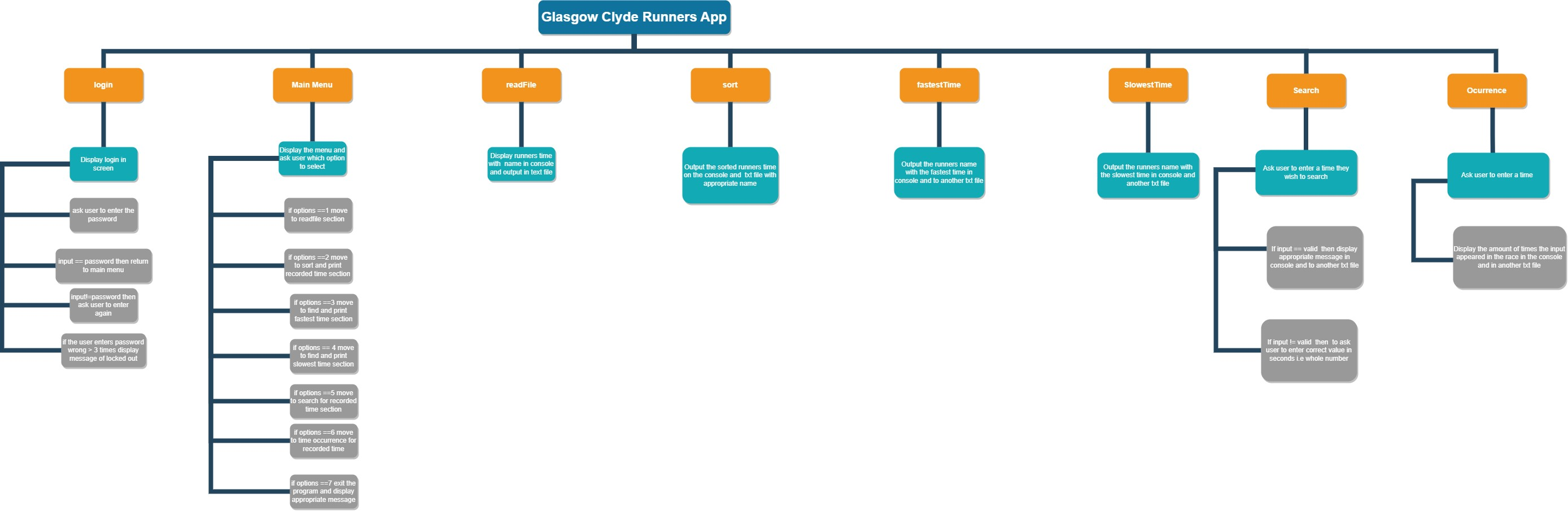


# **Data Dictionary**

|  |  |  |
| --- | --- | --- |
| **Name (Identifier)** | **Data Type** | **Description** |
| fileName | File | To be able to store the external text file |
| in | Scanner | To be able to take the user input, output and read file |
| out | Printwriter | To be able to output the information into another text file |
| result | String 2D array | To store the runners run time as well as the first and last name |
| sortedTimeSet | Int array | To extract the run time to 1d array in order to sort them from smallest to fastest |
| unsortedTimeSet | Int array | This will keep track of the runners time in the same order so that I can match runners time with their full name |
| firstName | String array | To store the first name of all the runners in the array so that I can match the fastest and slowest time with the correct name |
| lastName | String array | To store the last name of all the runners in the array so that I can match the fastest and slowest time with the correct name |
| password | String | To hold the password for the login and check to see if it matches with the user input |
| fail | int | To allow number of attempts by the user to get the past the login menu before they are locked out |
| login | String | To store the user input from the scanner class and later check to see if it matches the password |
| options | int | To match the user input with which functionality of the app they would like to interact with I.e display the info, sort etc |
| nameOfFiles | String | To store the location of the race txt file in order for it to be used later. |
| line | String | To capture each line from the text file and use it later to store in the 2D array. |
| time | Int | To capture each time of the runners and then transfer it into the 1D array |
| studentFName | String | To match runners first name with fastest and slowest time |
| studentLName | String | To match runners last name with the fastest and slowest time |
| Valid | boolean | To check whether the input by user matches with corresponding data type and if not ask the user to enter again |
| value | int | Returns the value of the linear search and whether it found the run time or not |
| count | int | Keeps track the number of times a finalist run time is recorded |

# **Structure Chart**

I have also attached png image of the structure chart in the zip folder for a more detailed image.



# **Pseudocode**

Print Welcome statement + ask user for password

If input == password

Go to Menu screen

Else if input not equal to password

Ask user to enter again + number of attempts left

If attempts > 3

Print appropriate message and close the program

Display main menu screen

Ask user to select an option from given choice

Do the following:

if user input == 1

Go to read and display function

If user input == 2

Go to sort and print recorded time function

If user input == 3

Go to find and print fastest time function

If user input == 4

Go to find and print slowest time function

If user input == 5

Go to search time function

If user input == 6

Go to time occurrence function

If user input == 7

Print appropriate message and close program

Until user input == 7

**Menu function**

Save external file location to variable

Read the external file while there is line to read

for loop and go through the length of the file

Copy each line of the text file to an array variable

Copy the line to a record

Do it until there is no further line to read

Print the record to the user

Also write to text file and save it to location

**Sort function**

Use for loop to go through length of record

Copy each runner's time from record to an array

Copy each first name from record to a separate array

Copy the last name from record to separate array

Use built in java function to sort the runner's time fastest to slowest

Print the information to the user

Also print it to another separate text file and save it to a location

**Fastest time function**

Create a variable to hold first name of runner

Create a variable to hold the last name of runner

Store the first number in an unsorted run time array to a variable fastestTime

For loop through the length of unsorted array

Compare the value in fastestTime variable to the next value in the array

if fastestTime > than next array vale then set the fastesttime to the new value

Also set first name/last name to the location of the value thereby matching the name with the time

Repeat until checked with all the value of an array

Print the fastest time to the user with the first and last name of runner

Print the information to another text file with appropriate name

**Slowest time function**

Create a variable to hold first name of runner

Create a variable to hold the last name of runner

Store the first number in an unsorted run time array to a variable Slowest

For loop through the length of unsorted array

if slowest time < than next array vale then set the slowest time to the new value

Also set first name/last name to the location of the value thereby matching the name with the time

Repeat until checked with all the value of an array

Create a location for the text file

Print the slowest time to the user with the first and last name of runner

Print the information to another text file with appropriate name

**Search function**

Create a Boolean true/ false variable and set it too false

Create variable with the name input and set it = 0

Do the following:

Under the try exception

Print the message asking user to input time they wish to find

Use scanner object to take the user input

If the input == interger than set Boolean variable to true

Catch input miss match exception

Print message input is not valid

Clear the previous input

Do this until (input == integer)

Create a variable to hold return value of linear search function

If return value not equal to –1 than print message to user that value found

Else print message to user not found

Create a location for the text file

Output the information to the user with appropriate message

Output the information to another text file

**Linear search function**

For loop through the entire length of runner's time

Compare each value in the runner time array with the user input

If user input == runners time

Return input;

Else return negative 1

**Occurrence function**

Print message asking user to enter a time to see how many times it appeared in race

Create variable called input = user input by using scanner object

Create count variable

For loop to go through the entire array of runners time

Compare each value to the user input

If user input == runners time

Increase the count variable by 1

Create txt file location to store the information

Print the information to the txt file

Also print the information to user with appropriate message