Design Document

PROGRAMMING GROUP PROJECT

PROGRAMMING 1 COMP2603

VLADIMIR BREABIN C21345416
RUAN MULLIGAN C21331716
EMIL SOBOCINSKI C21318073

TU821/2 GROUP B

Table Of Contents

High Level Description:

Goals:

The goal of this project is to create a program for entering, calculating, and printing the scorecards. The requirements as we understand them are:

- Allow players to create, enter the information, calculate the results, and print out their scorecards.
- Allow the club management to view a player's information & scorecards and create & delete users.

Additionally, we saw fit to implement a security system that required users to enter a username and password to access either their player accounts or the admin account.

Player Functionality:

Login:

The Player will be required to enter their unique username and password to access the functions below. This is to prevent other players, either intentionally or unintentionally, viewing or creating scorecards or the players' information.

Create Scorecard & Information:

We decided to combine these two requirements into one function as the only time a player would be interested in creating a scorecard was when they had information to enter. The information we ask the user to enter is the date the game took place and how many strokes they had on each hole. This will then be entered into a text document that can be read later. The reason why the date is asked to be entered is to differentiate it from other scorecards the player might like to enter.

Calculate & Printout Results:

We combined these two requirements as we decided instead of storing the full scorecard on the text file, we would only store the strokes on each hole and then, before printing the results, calculate the score. The results (the hole number, difficulty & par and the players strokes, free strokes, and total score at the end.) are then printed out in a manner the user can understand.

Club Management Functionality:

Login:

The management and/or administrator will be required to enter their password to access the functions below. This is to prevent other players, either intentionally or unintentionally, deleting players, creating players or view the information or scorecards of other players.

View Player Information & Scorecards:

The administration will be able to enter a player's username to view their full name, date of birth, handicap, and password. The administrator will then be able to enter the date of a game that the player entered to see their scorecard

Create Users:

Before any of the players will be able to store their game information in a file, the administration will have to create a unique profile for the player. In the profile there will be the user's username and password along with their full name, date of birth and handicap. Once created, the user will be able to login and add game data to their profile.

Delete Users:

The administration will be able to delete individual players and any information relating to them providing there is such a file for that user.name and they will be able to delete their profile, preventing them from logging in.

Algorithms Used:

UserMenu():

The user menu works by getting input from the user and returning a specific void function depending on the user's choice. For example:

1. Creating a new scorecard will run the writeScorecard() function.
2. Print Scorecards will run the readScorecard() function.
3. View Profile will run the Readprofile() function.
4. Logout and return to menu will run the Menu().
5. Exit & Close will close the program and print out a message.

login():

The login function uses the entered username to open the users text file and compare their entered username & password with that on file and if matching will let them proceed to the function UserMenu(). If either the username or password does not exist or match it will print off an error and ask for a new input.

Menu():

The menu function gets a user's input of either logging in as a user or as an admin, the user option will run the **login()** function while the admin option will run the **loginAdmin()** function.

adminOptions():

The adminOptions() function lets the administrator either create a new user, view user details, scorecards, remove an existing user, logout and return to menu & exit.
The selected option runs one of the following functions respectively: **getDetails() , readProfile(), removeUser(), Menu(),** and the exit option exits the program entirely.

loginAdmin():

The loginAdmin() function compares the entered password attempt with the hard coded password of “Admin” if the password is correct, it prints a welcome screen and runs the **adminOptions()** function.

removeUser():

The removeUser() function uses the entered username to delete the appropriate text file containing the user's information.

getDetails():

The getDetails() function is the function used to gather user details during registration it requests the user to input the following information: username, password, name, surname, date of birth & handicap level. Once all those have been collected it runs the **writeProfile()** function which then stores that data to a text file.

writeProfile():

The writeProfile() function uses the collected information from the function **getDetails()** and then stores it to a text file of the same name as the username.

readProfile():

The readProfile() function checks if the current user is the admin and if so, asks which user they would like to view. If it is a player, it retrieves their information from their text file.

writeScorecard():

The writeScorecard() function requests the user to input a date that the game was stored and then the strokes for each hole, once done it stores all the strokes in a text file that is called the username and date.

readScorecard():

The readScorcard() function retrieves the scorecard of the player depending on the date that the user entered. If no scorecard exists, the program returns to the main menu and prints out an error. If the scorecard exists it will use the handicap stored to calculate the results of that game and print them out. E.g., points, free strokes, net strokes, hole difficulty, par & hole number.

Flowchart & Class diagram

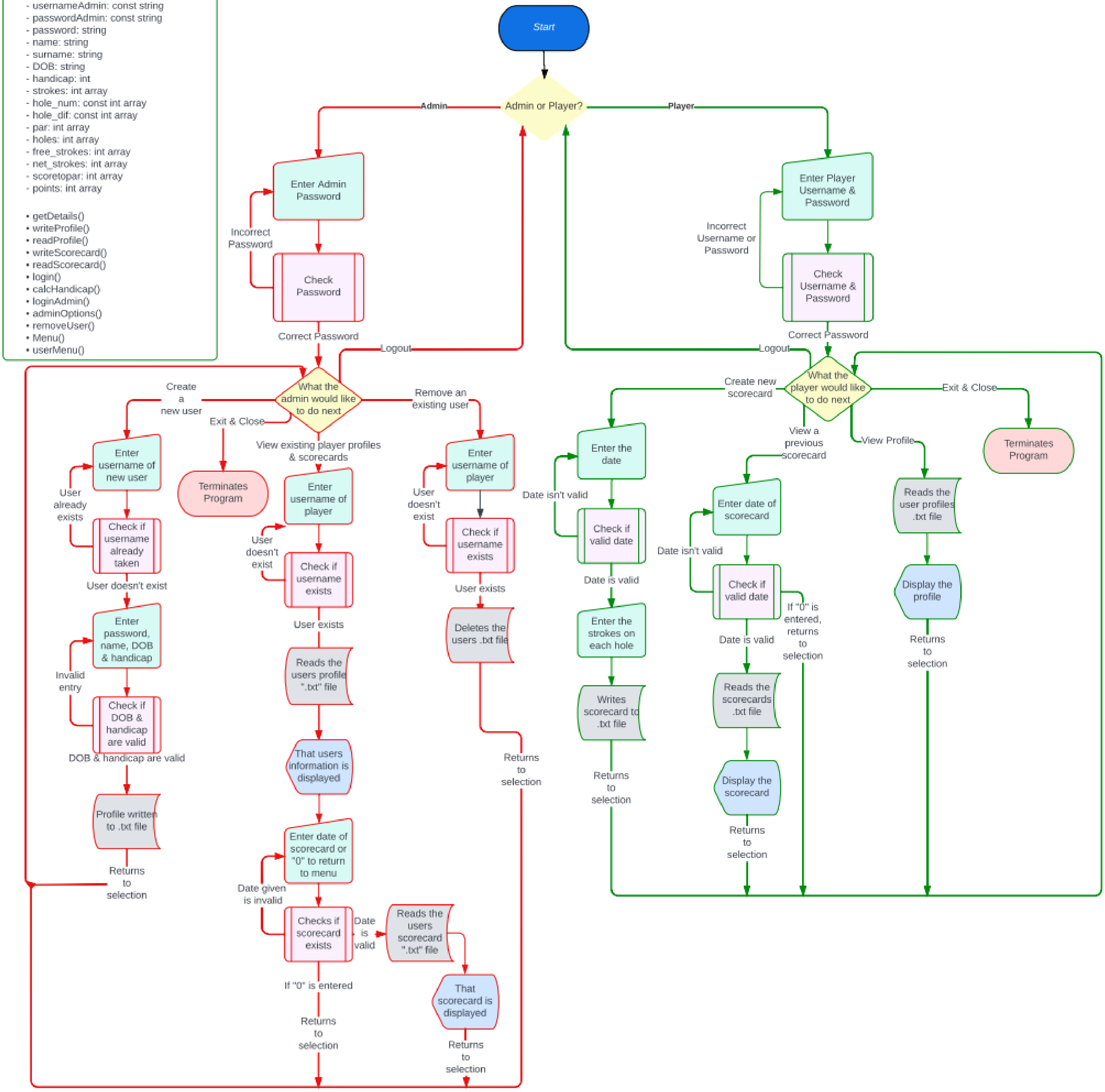
Golf Project Flow Chart & UML Diagram

Ruán Mulligan | Emil Sobocinski | Vladimir Breabin | December 2022 | COMP2603

NewUser

- exit3: int
- userNameAttempt: string
- passwordAttempt: string
- username: string
- usernameAdmin: const string
- passwordAdmin: const string
- password: string
- name: string
- surname: string
- DOB: string
- handicap: int
- strokes: int array
- hole_num: const int array
- hole_dif: const int array
- par: int array
- holes: int array
- free_strokes: int array
- net_strokes: int array
- scoretopar: int array
- points: int array

• getDetails()
• writeProfile()
• readProfile()
• writeScorecard()
• readScorecard()
• login()
• calcHandicap()
• loginAdmin()
• adminOptions()
• removeUser()
• Menu()
• userMenu()


Start
Admin or Player?
Admin
Player
Enter Admin Password
Incorrect Password
Check Password
Correct Password
Logout
Enter Player Username & Password
Incorrect Username or Password
Check Username & Password
Correct Password
Logout
What the admin would like to do next
Create a new user
Exit & Close
Remove an existing user
View existing player profiles & scorecards
Enter username of new user
User already exists
Check if username already taken
User doesn't exist
Enter password, name, DOB & handicap
Invalid entry
Check if DOB & handicap are valid
DOB & handicap are valid
Profile written to .txt file
Returns to selection
Terminates Program
Enter username of player
User doesn't exist
Check if username exists
User exists
Reads the users profile ".txt" file
That users information is displayed
Enter date of scorecard or "0" to return to menu
Date given is invalid
Checks if scorecard exists
Date is valid
Reads the users scorecard ".txt" file
That scorecard is displayed
Returns to selection
If "0" is entered
Returns to selection
Enter username of player
User doesn't exist
Check if username exists
User exists
Deletes the users .txt file
Returns to selection
What the player would like to do next
Create new scorecard
Exit & Close
View a previous scorecard
View Profile
Enter the date
Date isn't valid
Check if valid date
Date is valid
Enter the strokes on each hole
Writes scorecard to .txt file
Returns to selection
Enter date of scorecard
Date isn't valid
Check if valid date
If "0" is entered, returns to selection
Date is valid
Reads the scorecards .txt file
Display the scorecard
Returns to selection
Reads the user profiles .txt file
Display the profile
Returns to selection
Terminates Program