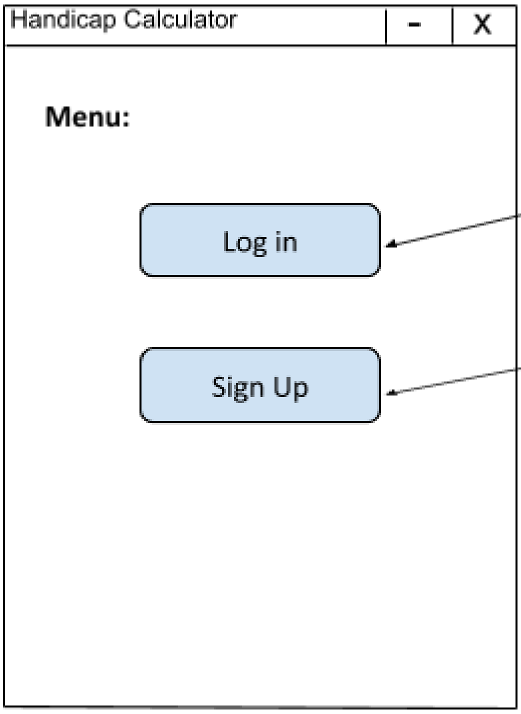
CRITERION B

GUI Design’s

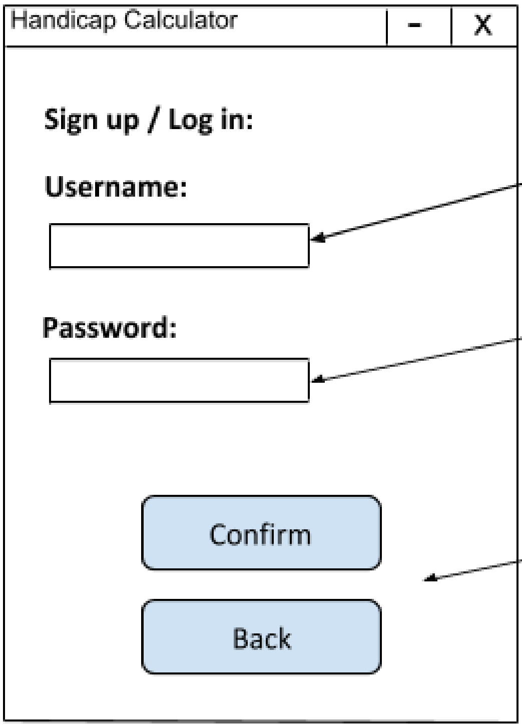
Start of the program - Menu GUI



This is a JButton that will take the user to a Log in GUI like the GUI below

This is a JButton that will take the user to a scorecard GUI so they can obtain a handicap before signing up.

Log in / Sign up GUI

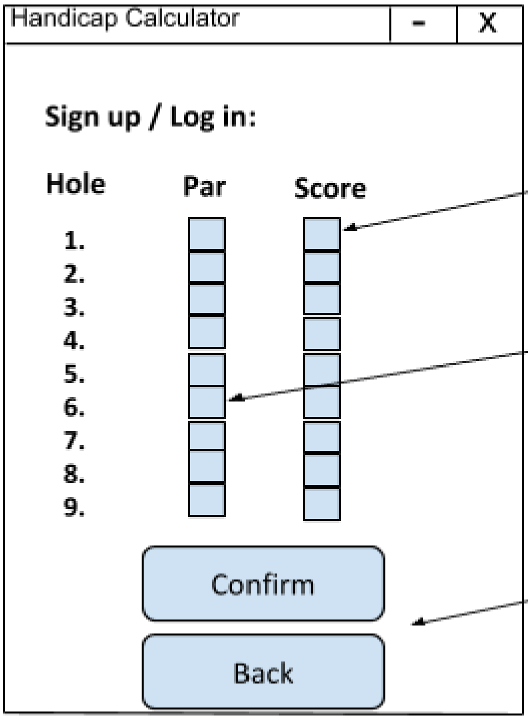


This is a JTextField where the user will enter their current username or a new username depending on whether they are signing up or login in

This is a JTextField where the user will enter their current password or a new password depending on whether they are signing up or login in

These are JButtons. The Confirm button will either save the users username and password or check the credentials against a file (see pseudocode) depending on whether they are signing up or login in. The back button takes the user back to the Menu GUI.

Scorecard GUI

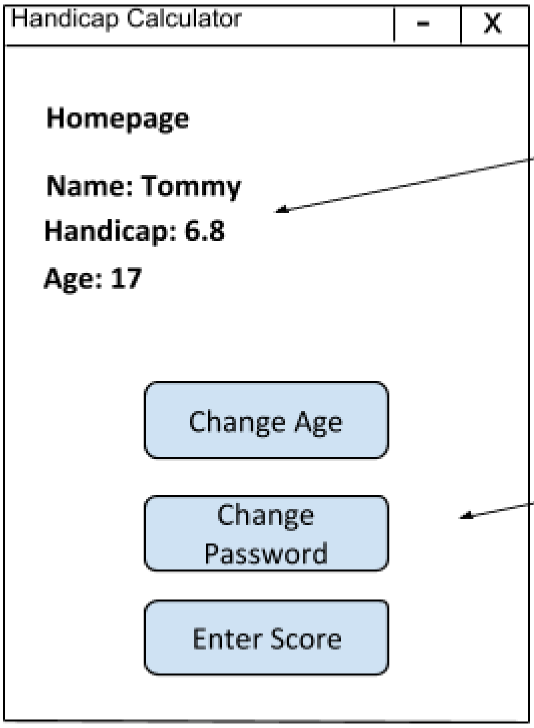


This is a JTextField where the user will be able to enter their score.

This is a JComboBox where the user will have one of 3 options for the par of a hole.

These are JButtons. The Confirm button will take the scores and calculate a handicap (see pseudocode) then it will take the user to the Sign Up GUI. The Back button goes back to the menu.

Homepage GUI



Name: Tommy

Handicap: 6.8

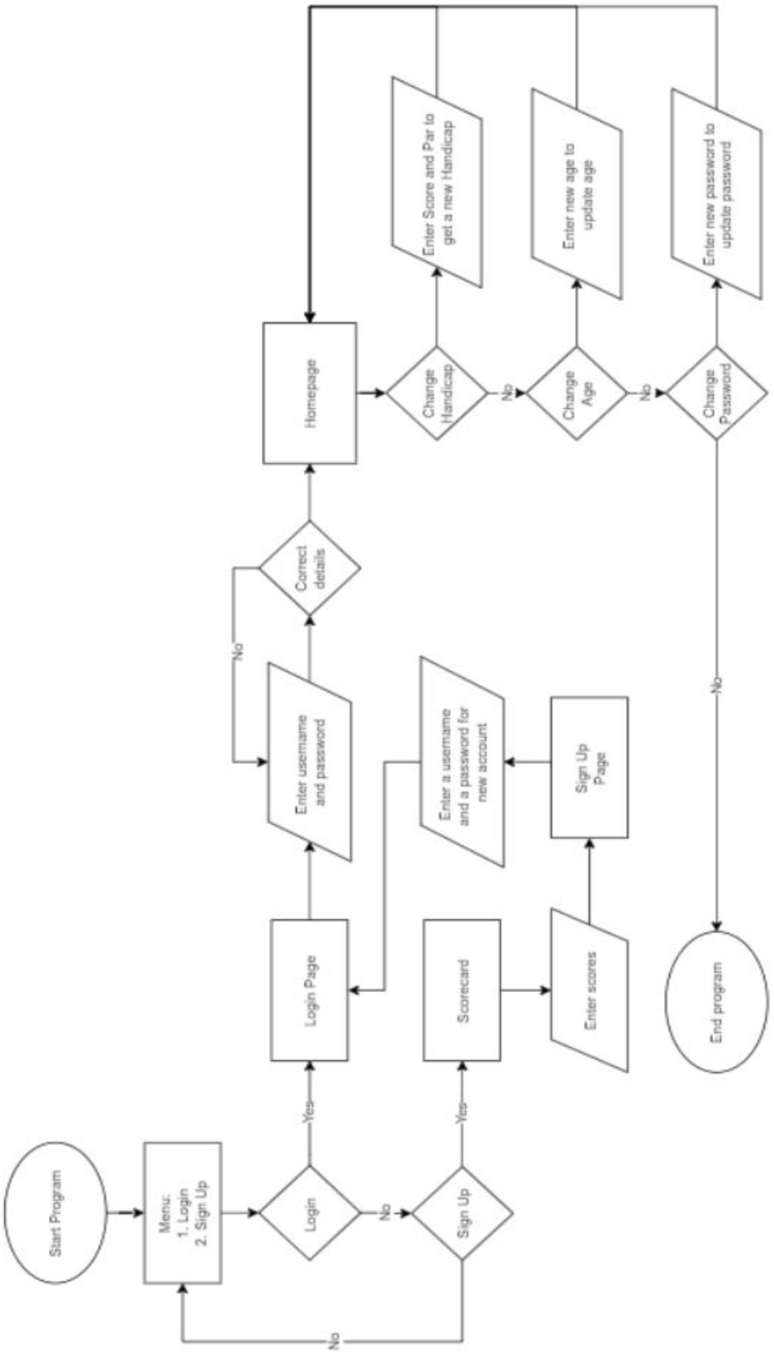
Age: 17

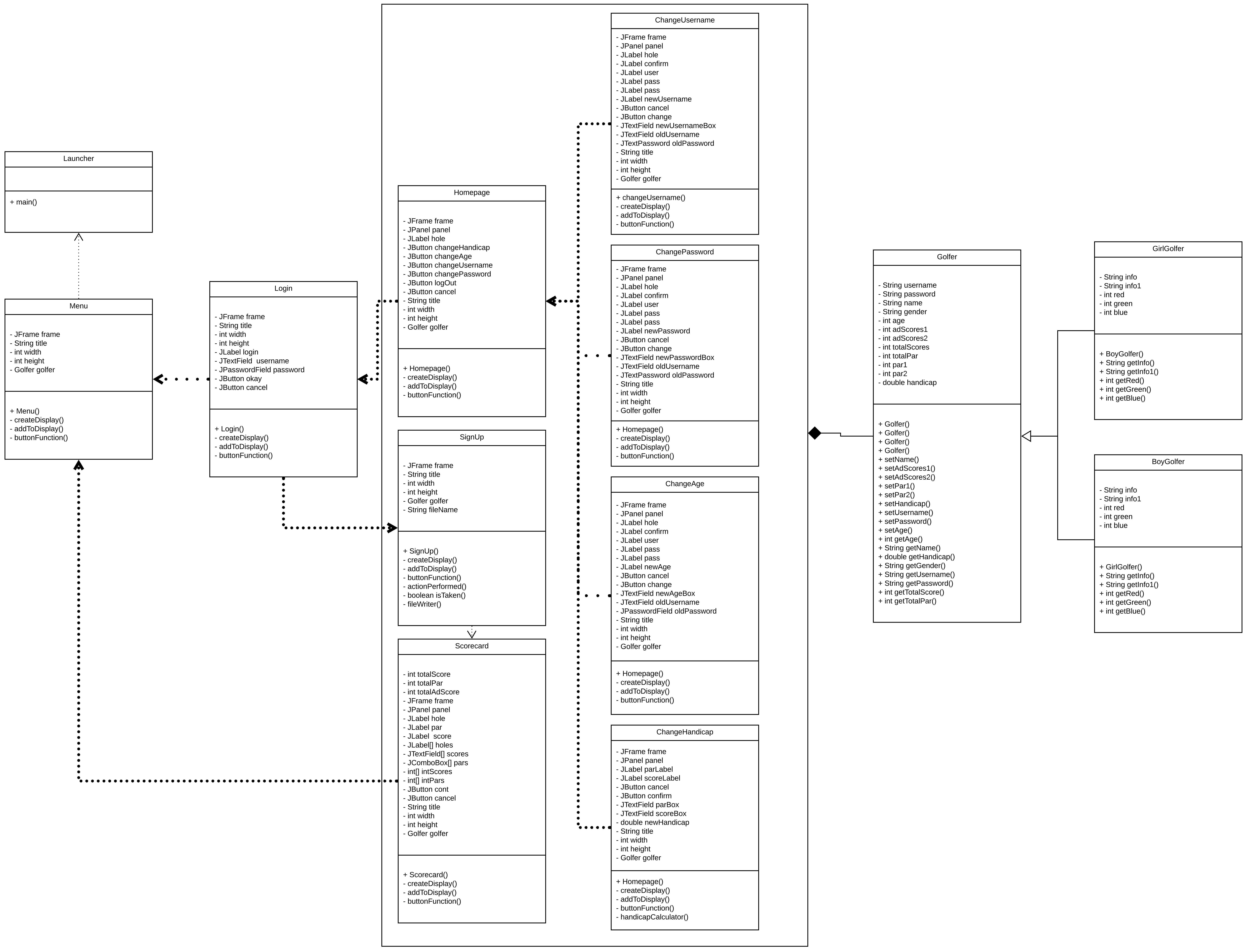
These are JLabels that present the user with their details.

These are JButtons. The Change Age button takes the user to a page where they can enter their new age. Similarly the Change Age button takes the user to a page where they can enter their new password. The Enter Score button takes the user to a page where they can enter the par of the course and their score, the new handicap can then be calculated recursively (see pseudocode).

Data Structure	Explanation	Use in code
ArrayList of Arrays	ArrayList of arrays has indexing whilst still being dynamic and being able to change size therefore it is perfect for the scenario.	When the text changing username, password, age or handicap, the array is used to temporarily store user details

Flowchart




Launcher
+ main()
Menu
- JFrame frame
- String title
- int width
- int height
- Golfer golfer
+ Menu()
- createDisplay()
- addToDisplay()
- buttonFunction()
Login
- JFrame frame
- String title
- int width
- int height
- JLabel login
- JTextField username
- JPasswordField password
- JButton okay
- JButton cancel
+ Login()
- createDisplay()
- addToDisplay()
- buttonFunction()
Homepage
- JFrame frame
- JPanel panel
- JLabel hole
- JButton changeHandicap
- JButton changeAge
- JButton changeUsername
- JButton changePassword
- JButton logOut
- JButton cancel
- String title
- int width
- int height
- Golfer golfer
+ Homepage()
- createDisplay()
- addToDisplay()
- buttonFunction()
SignUp
- JFrame frame
- String title
- int width
- int height
- Golfer golfer
- String fileName
+ SignUp()
- createDisplay()
- addToDisplay()
- buttonFunction()
- actionPerformed()
- boolean isTaken()
- fileWriter()
Scorecard
- int totalScore
- int totalPar
- int totalAdScore
- JFrame frame
- JPanel panel
- JLabel hole
- JLabel par
- JLabel score
- JLabel[] holes
- JTextField[] scores
- JComboBox[] pars
- int[] intScores
- int[] intPars
- JButton cont
- JButton cancel
- String title
- int width
- int height
- Golfer golfer
+ Scorecard()
- createDisplay()
- addToDisplay()
- buttonFunction()
ChangeUsername
- JFrame frame
- JPanel panel
- JLabel hole
- JLabel confirm
- JLabel user
- JLabel pass
- JLabel pass
- JLabel newUsername
- JButton cancel
- JButton change
- JTextField newUsernameBox
- JTextField oldUsername
- JTextPassword oldPassword
- String title
- int width
- int height
- Golfer golfer
+ changeUsername()
- createDisplay()
- addToDisplay()
- buttonFunction()
ChangePassword
- JFrame frame
- JPanel panel
- JLabel hole
- JLabel confirm
- JLabel user
- JLabel pass
- JLabel pass
- JLabel newPassword
- JButton cancel
- JButton change
- JTextField newPasswordBox
- JTextField oldUsername
- JTextPassword oldPassword
- String title
- int width
- int height
- Golfer golfer
+ Homepage()
- createDisplay()
- addToDisplay()
- buttonFunction()
ChangeAge
- JFrame frame
- JPanel panel
- JLabel hole
- JLabel confirm
- JLabel user
- JLabel pass
- JLabel pass
- JLabel newAge
- JButton cancel
- JButton change
- JTextField newAgeBox
- JTextField oldUsername
- JPasswordField oldPassword
- String title
- int width
- int height
- Golfer golfer
+ Homepage()
- createDisplay()
- addToDisplay()
- buttonFunction()
ChangeHandicap
- JFrame frame
- JPanel panel
- JLabel parLabel
- JLabel scoreLabel
- JButton cancel
- JButton confirm
- JTextField parBox
- JTextField scoreBox
- double newHandicap
- String title
- int width
- int height
- Golfer golfer
+ Homepage()
- createDisplay()
- addToDisplay()
- buttonFunction()
- handicapCalculator()
Golfer
- String username
- String password
- String name
- String gender
- int age
- int adScores1
- int adScores2
- int totalScores
- int totalPar
- int par1
- int par2
- double handicap
+ Golfer()
+ Golfer()
+ Golfer()
+ Golfer()
+ setName()
+ setAdScores1()
+ setAdScores2()
+ setPar1()
+ setPar2()
+ setHandicap()
+ setUsername()
+ setPassword()
+ setAge()
+ int getAge()
+ String getName()
+ double getHandicap()
+ String getGender()
+ String getUsername()
+ String getPassword()
+ int getTotalScore()
+ int getTotalPar()
GirlGolfer
- String info
- String info1
- int red
- int green
- int blue
+ BoyGolfer()
+ String getInfo()
+ String getInfo1()
+ int getRed()
+ int getGreen()
+ int getBlue()
BoyGolfer
- String info
- String info1
- int red
- int green
- int blue
+ GirlGolfer()
+ String getInfo()
+ String getInfo1()
+ int getRed()
+ int getGreen()
+ int getBlue()

Pseudocode

Handicap calculator - scorecard

```
Textfield[9] scores
JComboBox[9] pars
Loop (i 0 To 9) {
    int  intScores = parseInt(scores[i])
    int  intPars = parseInt(par[i])
}
int  totalScore = sum of intScores[]
Loop (x 0 To 9) {
    if((intScore[x] - intPars[x])  >  1){
        intScore[x] = intPars[x]  +  2
    }
}
int  totalAdScore = sum of intScores[]
int  totalPar = sum of intPars[]
```

Handicap calculator - recursion

```
int newHandicap
handicapCalculator(double handicap, int shotsUnder) {
    if (rounded handicap  <  6  &  shotsUnder  >  0) {
        handicap = handicap - 0.1
        shotsUnder = shotsUnder - 1
        handicapCalculator(handicap, shotsUnder)
    } else  if (rounded handicap  >  5  &  shotsUnder  >  0) {
        handicap = handicap - 0.2
        shotsUnder = shotsUnder - 1
        handicapCalculator(handicap, shotsUnder)
    } else  if (rounded handicap  >  12  &  shotsUnder  >  0) {
        handicap = handicap - 0.3
        shotsUnder = shotsUnder - 1
        handicapCalculator(handicap, shotsUnder)
    } else  if (rounded handicap  >  20  &  shotsUnder  >  0) {
        handicap = handicap - 0.4
        shotsUnder = shotsUnder - 1
        handicapCalculator(handicap, shotsUnder)
    } else  if (rounded handicap  >  28  &  shotsUnder  >  0) {
        handicap = handicap - 0.5
        shotsUnder = shotsUnder - 1
        handicapCalculator(handicap, shotsUnder)
    } else {
        newHandicap = handicap
    }
```

Log in

```
JTextField username
JPasswordField password

String user  =  get text from username
String pass  =  get text from password

File filename = “users.txt”
String line
boolean login  =  false

String username1
String password1
String name1
int age1
double handicap1
String gender1

try (Read file)
    While (Another line in file  &  login = false)
        line  =  line read from file
        String[] lines
        lines  =  split line into 6 with delimiter “, ”
        username1 = lines[0]
        password1 = lines[1]
        name1 =  lines[2]
        age1  =  parseInt lines[3]
        handicap1  =  parseDouble lines[4]
        gender1  =  lines[5]
        if (user  =  username1  &  pass  =  password1){
            login  =  true
        }
} catch (exception){
     login  =  false
}
```

Test Table

Test No.	**Test**	**Reason**	**Expected outcome**	**Success Criteria Tested**
1	The program starts with a GUI giving the user an option to 'login' or 'sign up'	Need to give the user a choice of logging in or signing up	When the program is run a menu GUI will appear where the user is given an option to either login in or sign up. These options should appear via JButtons	1 2
2	If you choose the 'sign up' JButton the program takes you to a scorecard GUI so that you can enter your scores	The user must be taken to a page where they are able to begin the sign up process - this begins by entering scores to get a handicap	The Sign Up button takes user to the Sign Up GUI	2
3	There is a 'back' JButton on all of the GUI's that takes the user back to the start GUI or previous page	The user should have the option to go back if they do not want to continue with their sign up or login process.	The back button takes the user to the start GUI	3
4	The scorecard GUI tells the user they need to fill out the scorecard to obtain their first handicap (this scorecard contains the first 9 holes)	Needs to be some information for the user so that they know what they need to do	Scorecard GUI gives the user information on what they have to do next	4
5	If one of the score boxes (JTextFields) on the scorecard GUI is filled with a character or left blank then an error message will appear via a JOptionPane asking to make sure all the inputs are numbers	The user can't enter letters in the score boxes because the score will not be able to be calculated. By checking if an integer has been entered this is avoided.	When a user enters a letter into one of the score boxes an error will appear in the form of a JOptionPane	4 (handicap won't be given if a score can not be calculated)
6	There are JComboBoxes for the par of each hole which contain the options 3, 4 or 5	Seeing as Par has a limited number of options a combo box is suitable for the user to choose the par of the hole	There will be combo boxes with 3 options for the par of each hole on the course	4

7	There is a continue JButton on scorecard GUI that takes the user to the actual sign up page and also saves scores to the Golfer object	The user must be able to go to the sign up page where they will be able to create their username and password. The button must also save credentials to the golfer object so that they can be written to the file when the user completes the sign up process	Scores and handicap that is calculated is saved to the golfer object	4
8	The Sign Up GUI has a JTextField and JPasswordField so the user can create a username and password	It is vital that the user has these fields so they can enter the username and password they would like. JPasswordField is important as it blanks out the characters that the user enters	The SignUp GUI will have a field for username and password and the password field will blank out letters	5
9	The Sign Up GUI gives them a choice to cancel the whole sign up process via a JButton	The user should have the opportunity to cancel signing up at any time	There is a cancel button that takes the user back to the start page	3
10	The Sign Up GUI has a confirm JButton that checks the username and password for length and if they are not >4 characters then there is an error	There needs to be security with the username and password as it is the authentication method they will use to access their private account	The user is asked to choose a username or password that is longer than 4 characters	5
11	If the username and password entered via the Sign Up GUI are valid then they are saved to the Golfer object	The username and password must be saved to the golfer object so that it can be saved to the file with the rest of the credentials	The details of the user will be saved to the golfer object	6
12	When the user signs up their details are all taken from the Golfer object and saved to a text file	The file is where all users are stored so it is important that all credentials from the user are written to the file	All the users details are copied from the golfer object into a file	6

13	The text file used to save the users details is not overwritten when a new user joins (each new user's details are stored a new line)	Saving a line to the file is adding a new user so lines must not be overwritten when saving to the file as it will delete a user.	The text file will not be overwritten and a new user's details are saved on a new line	6
14	Once the user has signed up using the Sign Up GUI the user is taken to the login GUI	Once the user is signed up they must use their new username and password to login and retrieve their details	The program will go from the sign up to the login if the user creates a valid account	5
15	The user is able to login to their account using the username and password created through the signup option	The user should be able to log in using the username and password made as this is how the will control and maintain their handicap	The user will be able to login using the details they created	5
16	The login GUI contains an 'okay' JButton that checks the user's credentials against the text file and the user is taken to a Homepage GUI if the credentials match	The user should only be able to access their homepage using the correct information as their details are private and only for them to use or edit.	When the okay button is clicked, the file will be read. If the details entered match the file the user will be taken to Homepage.	5 6
17	The Homepage GUI contains the user's credentials as well as the option to change their credentials	It is important that the homepage shows the user their credentials including the handicap as this will be the main page that they can show their friends	The user will be presented with a clear homepage that gives them options to change their details and their current details will be displayed	6 7
18	If the user chooses 'change handicap' they will be taken to a change handicap GUI with 2 JTextFields (for score and par of the course) and 2 JButton 'cancel' and 'confirm'	Changing handicap is the most important feature of the GUI as it is where the user will go to enter scores and update their handicap	The user will be taken to a GUI that allows them to enter a score and get a new handicap by confirming it	8

19	The program calculates a new handicap recursively	Calculating handicap requires a fraction to be taken off the handicap for every shot the user is under par therefore the best way to do this is recursion	When the score is entered this is taken and a new handicap will be calculated by recursion. Recursion will include parameter int shotsUnder as the base case	8
20	Any time the user tries to change a credential they are asked to confirm their username and password	This is for safety if the account has been left open nobody will be able to change details.	The user will be required to correctly enter their username and password to change details	6 7 8
21	Any time the user changes their credentials the text file is saved to an ArrayList of arrays and edited before printing the new credentials back to the file	Each user has an array of details and users can be edited so a dynamic data structure is needed. An ArrayList is best as it has direct access whereas other lists step through one by one.	Text file will be temporarily stored in an ArrayList so that users details can be edited and then reprinted to the file	7

REFER TO APPENDIX 6 FOR COMPLETED TEST TABLE AND ACTUAL OUTCOMES