APPENDICES

Appendix 1 - Transcript of the first conversation with the client (whilst playing golf)

Me: Hi Callum, would you be interested in me designing a computer program for you

Client: Yeah, maybe, what type of program will it be.

Me: The program will be made using java but the program will depend on what the client needs

Client: Yeah that sounds good.

Me: Okay, what would you like the program to do?

Client: A golf program would be useful because when I play with my friends they don't always have a handicap and this means that it's hard to keep score and compete with each other.

Me: That's a good idea, is there any specific features that you would like from the program

Client: I'm not sure … a login system would be good because then you would be able to keep track of each other's handicaps.

Me: Great, is there any other features you would like

Client: It would be good if the Handicap could be calculated using a proper system.

Me: Okay I'll use the CONGU guidelines for the program. Is that okay?

Client: Yes that sounds good

Me: Would you like anything else, like how would you like it to look or would you like a menu?

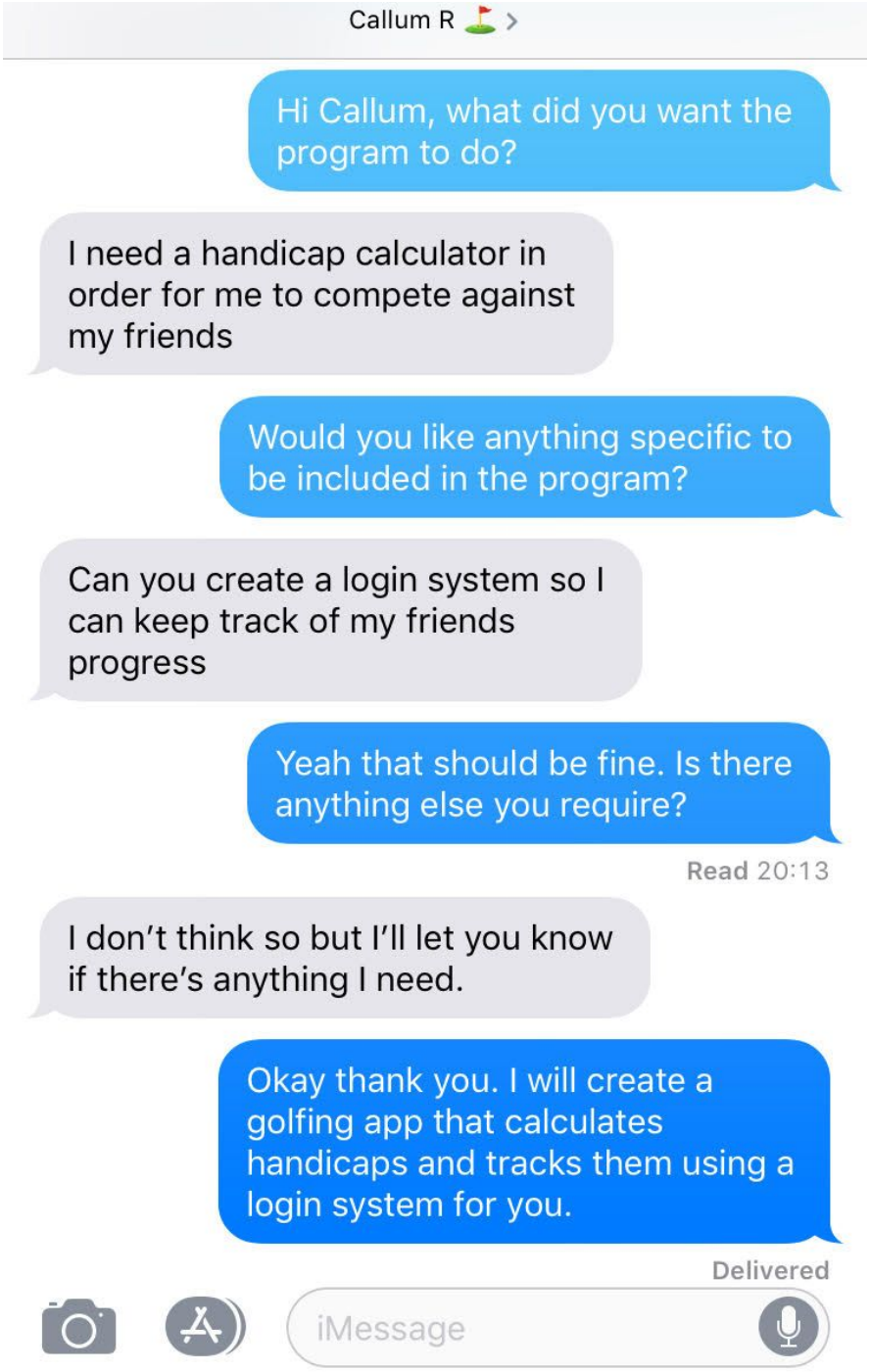
Client: It would be good if the program starts up with a menu so that I can choose whether I want to log in or sign up. Also, I would like a page where I would be able to see all my details, I think that is important.

Me: Okay that sounds good, anything else?

Client: No, I think that sounds good for me, I'll let you know if I think of anything else.

Me: Okay cool, thanks

Appendix 2 - Confirmation with client



Appendix 3 - 2nd meeting with the client (Transcript)

Me: Now that you have seen the product is there any changes you would make or like to see?

Client: Yeah, some additional information would be useful in parts as sometimes the program is a little confusing but overall, I like it.

Me: That's good to hear, do you have any other thoughts?

Client: Yeah, I see that the program requires details of age and gender. Would you be able to do something with this?

Me: Yeah, I could add some specialisation so that the user gets something different. For example, I could give boys a blue screen when they log in.

Client: That sounds good, I think that would be interesting for other users.

Me: Okay cool, is that everything?

Client: Yeah the rest of it seems good to me, it has all the features that I asked for

Appendix 4 - Final meeting with the client (Transcript)

Me: Now that you have seen the finished product what do you think?

Client: Overall, I am very happy because all of my needs have been met. I have a working program that is able to calculate handicaps and keep track of users.

Me: That's good, what are your favourite features?

Client: I really like the login feature because that was something I specifically requested and was very important to me. In addition to this, it calculated handicaps using a proper system and this was also something I specifically requested.

Me: Yeah that's good was there any other features you liked?

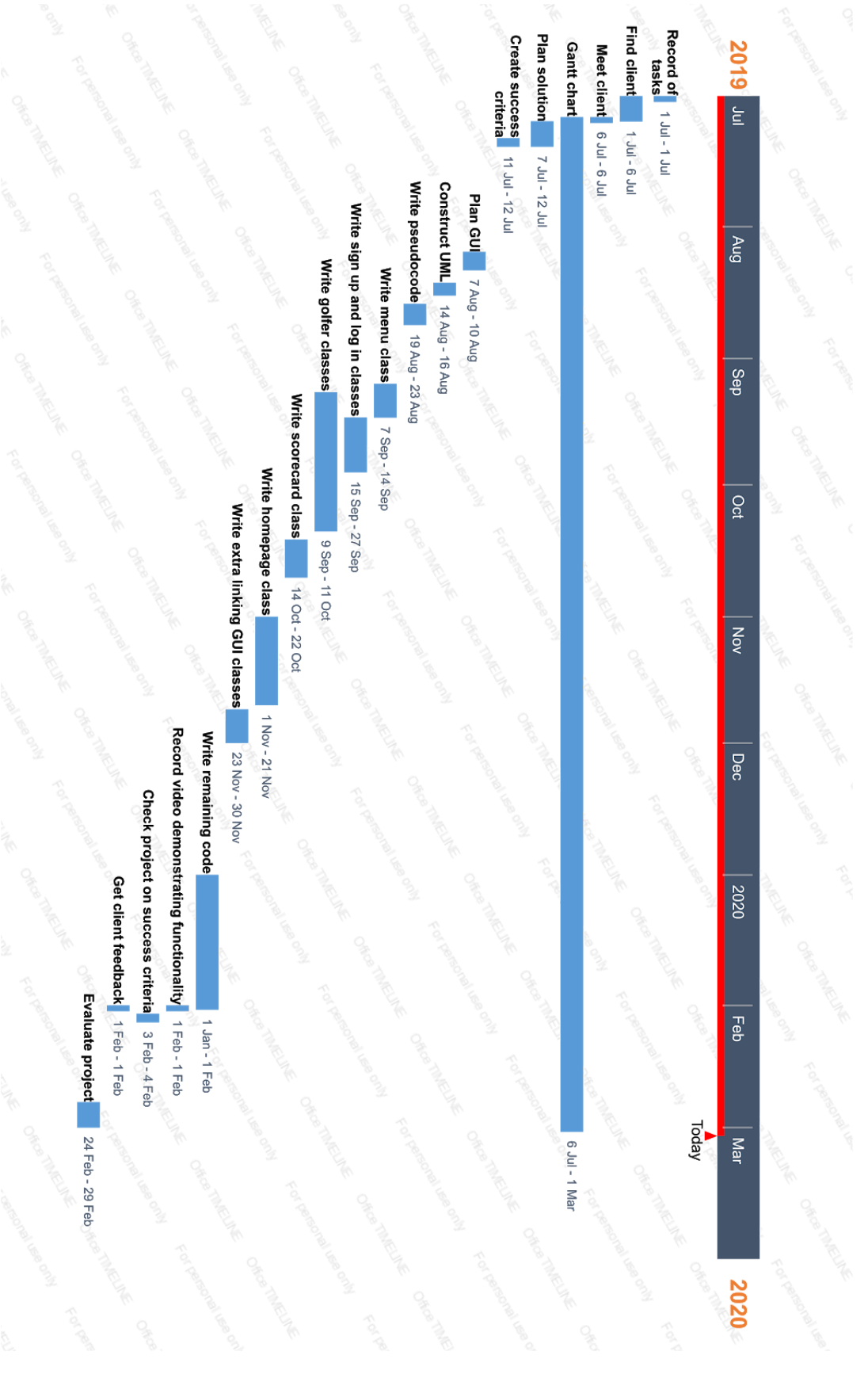
Client: Yes there are many features I could list. I like the flexibility and the flow of the program - you can go back whenever you like. Also, you have made the changes I have suggested in our last meeting.

Me: Okay cool. Is there anything that could be improved?

Client: Not much, I am very pleased with the product however I would like some more specialisation and for the program to run on a phone if that would be possible.

Me: Okay thank you for your feedback and collaboration

Appendix 5 - Gantt Chart



Appendix 6 - Completed test plan

Test No.	Test	Reason	Expected outcome	Actual outcome	Success
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	A GUI appears that asks the user if they want to sign up, login or if they want more information	✔
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	The user is not directly taken to the scorecard instead they enter their name, age and gender before they get to the scorecard	✔
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	Their is a back or cancel button on the GUI's where the user can choose to go back a step or cancel a process	✔
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		✔
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	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	When the user enters a letter a JOptionPane appears that tells the user not to leave a box empty or not to enter a letter.	✔

	sure all the inputs are numbers				
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	There are combo boxes with 3 options for the par of each hole on the course	✔
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	All details including gender, age, name, score and par are all saved to the golfer object	✔
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	The Signp GUI has a field for username and password and the password field blanks out the letters as they are entered.	✔
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	As said previously all GUI's have an option to go back or cancel a process	✔
10	The Sign Up GUI has a confirm JButton that checks the username and password for length and if they	There needs to be security with the username and password as it is the authentication method they	The user is asked to choose a username or password that is longer than 4 characters	There is a length check for the username and password. The username and password are also checked for spaces	✔

	are not >4 characters then there is an error	will use to access their private account		and commas	
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	The details of the user are saved to the golfer object	✔
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	All the users details are copied from the golfer object into a file separated by commas	✔
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	The text file is not overwritten and a new user's details are saved on a new line.	✔
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	The program goes from the sign up to the login when the user creates a valid account	✔
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	The user is able to login using the details they created	✔
16	The login GUI contains an 'okay' JButton that checks the user's credentials against the text file and the user is taken to	The user should only be able to access their homepage using the correct information as their details are	When the okay button is clicked, the file will be read. If the details entered match the file the user will be	When the okay button is clicked, the file is read from and if the details entered match the file the user is taken to	✔

	a Homepage GUI if the credentials match	private and only for them to use or edit.	taken to Homepage.	Homepage.	
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	The user is presented with a homepage where they can change their details and their current details are displayed on the page.	✔
18	If the user chooses 'change handicap' they will be taken to a change handicap GUI where there will be 2 JTextFields (for score and par of the course they have played) 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	The user is taken to a GUI that allows them to enter their score and par. However they are sent to a confirmation page where they can confirm the change.	✔
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	The score is entered and a new handicap is calculated by recursion. The recursion includes parameter int shotsUnder as a base case which decreases to 0.	✔
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	When the user tries to change details they are required to enter their username and password	✔
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	Text file will be temporarily stored in an ArrayList so that users details can be edited and then reprinted to the file	Text file is temporarily stored in an ArrayList so that users details are edited and then reprinted to the text file.	✔

		access whereas other lists step through one by one.			

Appendix 7 - Code

```
public class Launcher {


    // The first method to be run when the code starts
    public static void main(String[] args){


        //This is where the program is launched - First GUI is called
        new Menu("Handicap Helper", 210, 350);


    }
}


public class Golfer {
// Attributes to the golfer
    private String username;
    private String password;
    private String name;
    private double handicap;
    private int age;
    private int adScores1;
    private int adScores2;
    private int totalScores;
    private int totalPar;
    private int par1;
    private int par2;
    private String gender;
```

```
//4 constructors methods for the golfer
    public Golfer(String name, int age, String username, String password, double handicap) {
        this.name = name;
        this.age = age;
        this.username = username;
        this.password = password;
        this.handicap = handicap;
    }

    public Golfer(String name, int age, String username, String password, double handicap, String gender) {
        this.name = name;
        this.age = age;
        this.username = username;
        this.password = password;
        this.handicap = handicap;
        this.gender = gender;
    }

    public Golfer(String name, int age) {
        this.name = name;
        this.age = age;

    }
    public Golfer(String gender){
        if(gender.equals("Rather not say")){
            this.gender = "N/A";
        }else if(gender.equals("Other")){
            this.gender = "Unspecified";
```

```
        }else {
            this.gender = gender;
        }
    }

//Getters and setter(mutator) methods that means the object can accessed from outside
    void setName(String name){
        this.name = name;
    }
    void setAdScores1(int adScores1) { this.adScores1 = adScores1; }
    void setAdScores2(int adScores2) { this.adScores2 = adScores2; }
    void setPar1(int par1) { this.par1 = par1; }
    void setPar2(int par2) { this.par2 = par2; }
    void setHandicap(double handicap) { this.handicap = handicap; }
    void setHandicap() { this.handicap = -((this.par1 + this.par2) - (this.adScores1 + this.adScores2)); }
    void setUsername(String username) { this.username = username; }
    void setPassword(String password) { this.password = password; }
    void setAge(int age) { this.age = age; }

    public int getAge() { return this.age; }
    public String getGender() { return this.gender; }
    public String getName() {
        return this.name;
    }
    public double getHandicap() { return this.handicap; }
    public String getUsername() {
        return this.username;
    }
    public String getPassword() {
```

```
        return this.password;
    }
    public int getTotalScores() {
        return adScores1 + adScores2;
    }
    public int getTotalPar() {
        return par1 + par2;
    }

}

public class BoyGolfer extends Golfer{
//class inherited from Golfer
//additional attributes
    private String info = "This is a junior account designed";
    private String info1 = "for boys";
    private int red = 0;
    private int green = 204;
    private int blue = 255;

//Constructor
    public BoyGolfer(String name, int age, String username, String password, double handicap, String gender){
        super(name, age, username, password, handicap, gender);
    }
//getters
     public String getInfo(){return info;}
     public String getInfo1(){return info1;}
     public int getRed(){return red;}
     public int getGreen(){return green;}
```

```
    public int getBlue(){return blue;}
}
public class GirlGolfer extends Golfer{
//inherited from golfer
//extra attributes
   private String info = "This is a junior account designed";
   private String info1 = "for girls";

//constructor
   public GirlGolfer(String name, int age, String username, String password, double handicap, String gender){
      super(name, age, username, password, handicap, gender);
   }

//getters
   public String getInfo(){return info;}
   public String getInfo1(){return info1;}

}

import javax.swing.*;
      import java.awt.*;
      import java.io.*;
      import java.util.Scanner;
      import java.util.ArrayList;
      import java.util.List;
//imported java classes
public class ChangeAge {

   public JFrame frame;
```

```
    //Allows us to draw graphics to it
    public JPanel panel;

//attributes
    private String title;
    private int width;
    private int height;
    private Golfer golfer;
    private BoyGolfer golfer1;
    private GirlGolfer golfer2;

//Constructors
    ChangeAge(String title, int width, int height, Golfer golfer) {
        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer = golfer;

        //Calling the method to initialise the JFrame
        createDisplay();
        addToDisplay();
        buttonFunction();
        cancel();
    }

    ChangeAge(String title, int width, int height, BoyGolfer golfer) {
        this.title = title;
        this.width = width;
        this.height = height;
```

```
        this.golfer1 = golfer;

        //Calling the method to initialise the JFrame
        createDisplay();
        addToDisplay();
        buttonFunction1();
        cancel1();
    }

    ChangeAge(String title, int width, int height, GirlGolfer golfer) {
        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer2 = golfer;

        //Calling the method to initialise the JFrame
        createDisplay();
        addToDisplay();
        buttonFunction2();
        cancel2();
    }

//additional attributes
    private JLabel confirm = new JLabel("Confirm your previous login details");
    private JLabel user = new JLabel("Username:");
    private JLabel pass = new JLabel("Password:");
    private JLabel newAge = new JLabel("Enter your age");
    private JButton cancel = new JButton("Cancel");
    private JButton change = new JButton("Change");
```

```
private JTextField newAgeBox = new JTextField();
private JTextField oldUsername = new JTextField();
private JPasswordField oldPassword = new JPasswordField();



private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}

private void addToDisplay() {
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    panel.add(confirm);
```

```
    panel.add(user);
    panel.add(oldUsername);
    panel.add(pass);
    panel.add(oldPassword);
    panel.add(newAge);
    panel.add(newAgeBox);
    panel.add(new JPanel());
    panel.add(change);
    panel.add(cancel);
    frame.add(panel);
    frame.pack();
}

private void buttonFunction() {
    change.addActionListener(e -> {
        String newAge = newAgeBox.getText();
        boolean integer = true;
        char[] chars = newAge.toCharArray();
        for (int i = 0; i < chars.length; i++) {
            if ((int) chars[i] < 48 || (int) chars[i] > 57) {
                integer = false;
            }
        }
        String user = oldUsername.getText();
        String pass = String.valueOf(oldPassword.getPassword());

        if (integer) {
            List<String[]> users = new ArrayList<>();
            File fileName = new File("users.txt");
```

```
            String line;
            String[] newArray = {golfer.getUsername(), golfer.getPassword(), golfer.getName(),
Integer.toString(golfer.getAge()), Double.toString(golfer.getHandicap()), golfer.getGender()};

            if (golfer.getUsername().equals(user) && golfer.getPassword().equals(pass)) {

               try (Scanner scanner = new Scanner(new FileReader(fileName))) {
                  while (scanner.hasNextLine()) {
                     line = scanner.nextLine();
                     String[] lines = line.split(", ", 6);
                     users.add(lines);
                     if (lines[0].equals(golfer.getUsername()) && lines[1].equals(golfer.getPassword())) {
                        golfer.setAge(Integer.parseInt(newAge));
                        newArray[0] = golfer.getUsername();
                        newArray[1] = golfer.getPassword();
                        newArray[2] = golfer.getName();
                        newArray[3] = Integer.toString(golfer.getAge());
                        newArray[4] = Double.toString(golfer.getHandicap());
                        newArray[5] = golfer.getGender();
                     }
                  }

                  FileWriter clearFile = new FileWriter(fileName, false);
                  PrintWriter clearPrint = new PrintWriter(clearFile);
                  clearPrint.flush();
                  clearPrint.close();

                  for (int i = 0; i < users.size(); i++) {
```

```
            if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer.getPassword())) {
                users.set(i, newArray);

                FileWriter fileWriter = new FileWriter(fileName, true);
                PrintWriter printWriter = new PrintWriter(fileWriter);
                String toFile = String.join(", ", newArray);
                printWriter.println(toFile);
                printWriter.close();

            } else {
                FileWriter fileWriter = new FileWriter(fileName, true);
                PrintWriter printWriter = new PrintWriter(fileWriter);
                String toFile2 = String.join(", ", users.get(i));
                printWriter.println(toFile2);
                printWriter.close();
            }
        }
        new Homepage("Handicap Helper", 210, 350, golfer);
        frame.dispose();

      } catch (Exception e1) {
        new ChangeUsername("Handicap Helper", 210, 350, golfer);
      }
    } else {
      JOptionPane.showMessageDialog(frame, "Change Age failed! Try Again!");
    }
} else {
    JOptionPane.showMessageDialog(frame, "New Age must be a number");
}
```

```
        });
    }
    private void buttonFunction1() {
        change.addActionListener(e -> {
            String newAge = newAgeBox.getText();
            boolean integer = true;
            char[] chars = newAge.toCharArray();
            for (int i = 0; i < chars.length; i++) {
                if ((int) chars[i] < 48 || (int) chars[i] > 57) {
                    integer = false;
                }
            }
            String user = oldUsername.getText();
            String pass = String.valueOf(oldPassword.getPassword());

            if (integer) {
                List<String[]> users = new ArrayList<>();
                File fileName = new File("users.txt");
                String line;
                String[] newArray = {golfer1.getUsername(), golfer1.getPassword(), golfer1.getName(),
Integer.toString(golfer1.getAge()), Double.toString(golfer1.getHandicap()), golfer1.getGender()};

                if (golfer1.getUsername().equals(user) && golfer1.getPassword().equals(pass)) {

                    try (Scanner scanner = new Scanner(new FileReader(fileName))) {
                        while (scanner.hasNextLine()) {
                            line = scanner.nextLine();
                            String[] lines = line.split(", ", 6);
                            users.add(lines);
```

```
    if (lines[0].equals(golfer1.getUsername()) && lines[1].equals(golfer1.getPassword())) {
        golfer1.setAge(Integer.parseInt(newAge));
        newArray[0] = golfer1.getUsername();
        newArray[1] = golfer1.getPassword();
        newArray[2] = golfer1.getName();
        newArray[3] = Integer.toString(golfer1.getAge());
        newArray[4] = Double.toString(golfer1.getHandicap());
        newArray[5] = golfer1.getGender();
    }
}



FileWriter clearFile = new FileWriter(fileName, false);
PrintWriter clearPrint = new PrintWriter(clearFile);
clearPrint.flush();
clearPrint.close();

for (int i = 0; i < users.size(); i++) {
    if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer1.getPassword())) {
        users.set(i, newArray);

        FileWriter fileWriter = new FileWriter(fileName, true);
        PrintWriter printWriter = new PrintWriter(fileWriter);
        String toFile = String.join(", ", newArray);
        printWriter.println(toFile);
        printWriter.close();

    } else {
        FileWriter fileWriter = new FileWriter(fileName, true);
```

```
                    PrintWriter printWriter = new PrintWriter(fileWriter);
                    String toFile2 = String.join(", ", users.get(i));
                    printWriter.println(toFile2);
                    printWriter.close();
                }
            }
            new Homepage("Handicap Helper", 210, 350, golfer1);
            frame.dispose();

        } catch (Exception e1) {
            new ChangeUsername("Handicap Helper", 210, 350, golfer1);
        }
      } else {
        JOptionPane.showMessageDialog(frame, "Change Age failed! Try Again!");
      }
    } else {
      JOptionPane.showMessageDialog(frame, "New Age must be a number");
    }
  });
}
private void buttonFunction2() {
  change.addActionListener(e -> {
    String newAge = newAgeBox.getText();
    boolean integer = true;
    char[] chars = newAge.toCharArray();
    for (int i = 0; i < chars.length; i++) {
      if ((int) chars[i] < 48 || (int) chars[i] > 57) {
        integer = false;
      }
```

```
        }
        String user = oldUsername.getText();
        String pass = String.valueOf(oldPassword.getPassword());

        if (integer) {
            List<String[]> users = new ArrayList<>();
            File fileName = new File("users.txt");
            String line;
            String[] newArray = {golfer2.getUsername(), golfer2.getPassword(), golfer2.getName(),
Integer.toString(golfer2.getAge()), Double.toString(golfer2.getHandicap()), golfer2.getGender()};

            if (golfer2.getUsername().equals(user) && golfer2.getPassword().equals(pass)) {

                try (Scanner scanner = new Scanner(new FileReader(fileName))) {
                    while (scanner.hasNextLine()) {
                        line = scanner.nextLine();
                        String[] lines = line.split(", ", 6);
                        users.add(lines);
                        if (lines[0].equals(golfer2.getUsername()) && lines[1].equals(golfer2.getPassword())) {
                            golfer2.setAge(Integer.parseInt(newAge));
                            newArray[0] = golfer2.getUsername();
                            newArray[1] = golfer2.getPassword();
                            newArray[2] = golfer2.getName();
                            newArray[3] = Integer.toString(golfer2.getAge());
                            newArray[4] = Double.toString(golfer2.getHandicap());
                            newArray[5] = golfer2.getGender();
                        }
                    }
```

```
    FileWriter clearFile = new FileWriter(fileName, false);
    PrintWriter clearPrint = new PrintWriter(clearFile);
    clearPrint.flush();
    clearPrint.close();

    for (int i = 0; i < users.size(); i++) {
       if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer2.getPassword())) {
          users.set(i, newArray);

          FileWriter fileWriter = new FileWriter(fileName, true);
          PrintWriter printWriter = new PrintWriter(fileWriter);
          String toFile = String.join(", ", newArray);
          printWriter.println(toFile);
          printWriter.close();

       } else {
          FileWriter fileWriter = new FileWriter(fileName, true);
          PrintWriter printWriter = new PrintWriter(fileWriter);
          String toFile2 = String.join(", ", users.get(i));
          printWriter.println(toFile2);
          printWriter.close();
       }
    }
    new Homepage("Handicap Helper", 210, 350, golfer2);
    frame.dispose();

} catch (Exception e1) {
    new ChangeUsername("Handicap Helper", 210, 350, golfer2);
```

```
                }
            } else {
                JOptionPane.showMessageDialog(frame, "Change Age failed! Try Again!");
            }
        } else {
            JOptionPane.showMessageDialog(frame, "New Age must be a number");
        }
    });
}

private void cancel() {
    cancel.addActionListener(e -> {
        new Homepage("Handicap Helper", 210, 350, golfer);
        frame.dispose();
    });
}

private void cancel1() {
    cancel.addActionListener(e -> {
        new Homepage("Handicap Helper", 210, 350, golfer1);
        frame.dispose();
    });
}
private void cancel2() {
    cancel.addActionListener(e -> {
        new Homepage("Handicap Helper", 210, 350, golfer2);
        frame.dispose();
    });
}
```

```
}
import javax.swing.*;
import java.awt.*;
import java.text.DecimalFormat;



public class ChangeHandicap {

    public JFrame frame;
    //Allows us to draw graphics to it
    public JPanel panel;



    private String title;
    private int width;
    private int height;
    private Golfer golfer;
    private BoyGolfer golfer1;
    private GirlGolfer golfer2;



    ChangeHandicap(String title, int width, int height, Golfer golfer){
        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer = golfer;

        //Calling the method to initialise the JFrame
        createDisplay();
```

```
    addToDisplay();
    buttonFunction();
    cancel();
}

ChangeHandicap(String title, int width, int height, BoyGolfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer1 = golfer;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction1();
    cancel1();
}

ChangeHandicap(String title, int width, int height, GirlGolfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer2 = golfer;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction2();
    cancel2();
```

```
}

private JLabel parLabel = new JLabel("Par:");
private JLabel scoreLabel = new JLabel("Score:");
private JButton cancel = new JButton("Cancel");
private JButton confirm = new JButton("Confirm");
private JTextField parBox = new JTextField();
private JTextField scoreBox = new JTextField();
double newHandicap = 0.0;



private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}
private void addToDisplay(){
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
```

```
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    panel.add(parLabel);
    panel.add(parBox);
    panel.add(new JPanel());
    panel.add(scoreLabel);
    panel.add(scoreBox);
    panel.add(new JPanel());
    panel.add(confirm);
    panel.add(cancel);
    frame.add(panel);
    frame.pack();
}

private void buttonFunction(){
    confirm.addActionListener(e -> {
        String score = scoreBox.getText();
        String par = parBox.getText();
        boolean parIsInteger = true;
        boolean scoreIsInteger = true;
        char[] scoreChars = score.toCharArray();
        char[] parChars = par.toCharArray();
        DecimalFormat df2 = new DecimalFormat("#.##");

        for(int i = 0; i < scoreChars.length; i++) {
            if ((int)scoreChars[i] < 48 || (int)scoreChars[i] > 57 ) {
                scoreIsInteger = false;
```

```
            }
        }
        for(int i = 0; i < parChars.length; i++) {
            if ((int)parChars[i] < 48 || (int)parChars[i] > 57 ) {
                parIsInteger = false;
            }
        }

        if(scoreIsInteger && parIsInteger) {
            String toPar;
            int shotsUnder = 0;
            int netScore = (int)((Integer.parseInt(score) - Math.round(golfer.getHandicap())) -
(Integer.parseInt(par)));

            if(Integer.parseInt(score) > 120){
                toPar = "tooHigh";
            }else if(Integer.parseInt(score) < 50){
                toPar = "tooLow";
            }else if(Integer.parseInt(par) < 54){
                toPar = "parLow";
            }else if(Integer.parseInt(par) > 90 ){
                toPar = "parHigh";
            }else if (netScore < 0) {
                 toPar = "under";
                 shotsUnder = -(netScore);
            }else if(netScore > 0){
                    toPar = "over";
            }else{
                toPar = "level";
```

```
            }
             boolean[] cat = {false, false, false, false, false, false};
            if((int) golfer.getHandicap() > 5.4 && (int) Math.round(golfer.getHandicap()) < 11.5){
               cat[2] = true;
            }else if((int) Math.round(golfer.getHandicap()) > 11.4 && (int) Math.round(golfer.getHandicap()) <
20.5){
               cat[3] = true;
            }else if((int) Math.round(golfer.getHandicap()) > 20.4 && (int) Math.round(golfer.getHandicap()) <
27.5){
               cat[4] = true;
            }else if((int) Math.round(golfer.getHandicap()) > 27.4){
               cat[5] = true;
            }else{
               cat[1] = true;
            }
            double currentHandicap = golfer.getHandicap();

            if(toPar.equals("under")){
                  handicapCalculator(currentHandicap, shotsUnder);
                  newHandicap = Double.parseDouble(df2.format(newHandicap));
                  new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                  frame.dispose();
            }else if(toPar.equals("over")){
               if(cat[1] && netScore > 1 ){
                  newHandicap = currentHandicap + 0.1;
                  newHandicap = Double.parseDouble(df2.format(newHandicap));
                  new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
```

```
                frame.dispose();
            }else if(cat[2] && netScore > 2 ){
                newHandicap = currentHandicap + 0.1;
                newHandicap = Double.parseDouble(df2.format(newHandicap));
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }else if(cat[3] && netScore > 3 ){
                newHandicap = currentHandicap + 0.1;
                newHandicap = Double.parseDouble(df2.format(newHandicap));
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }else if(cat[4] && netScore > 4 ){
                newHandicap = currentHandicap + 0.1;
                newHandicap = Double.parseDouble(df2.format(newHandicap));
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }else if(cat[5] && netScore > 5 ){
                newHandicap = currentHandicap + 0.1;
                newHandicap = Double.parseDouble(df2.format(newHandicap));
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }
        }else if(toPar.equals("tooHigh")){
            JOptionPane.showMessageDialog(frame, "The score you have entered exceeds the maximum
limit");
```

```
                new ChangeHandicap("Handicap Helper", 210, 350, golfer);
                frame.dispose();
            }else if(toPar.equals("tooLow")){
                JOptionPane.showMessageDialog(frame, "The score you have entered exceeds the minimum
limit");
                new ChangeHandicap("Handicap Helper", 210, 350, golfer);
                frame.dispose();
            }else if(toPar.equals("parLow")){
                JOptionPane.showMessageDialog(frame, "The course par you have entered exceeds the minimum
limit");
                new ChangeHandicap("Handicap Helper", 210, 350, golfer);
                frame.dispose();
            }else if(toPar.equals("parHigh")){
                JOptionPane.showMessageDialog(frame, "The score you have entered exceeds the maximum
limit");
                new ChangeHandicap("Handicap Helper", 210, 350, golfer);
                frame.dispose();
            }else{
                newHandicap = currentHandicap;
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }



        }else{
            JOptionPane.showMessageDialog(frame, "New Age must be a number");
        }
    });
```

```
    }
    private void buttonFunction1(){
        confirm.addActionListener(e -> {
            String score = scoreBox.getText();
            String par = parBox.getText();
            boolean parIsInteger = true;
            boolean scoreIsInteger = true;
            char[] scoreChars = score.toCharArray();
            char[] parChars = par.toCharArray();
            DecimalFormat df2 = new DecimalFormat("#.##");

            for(int i = 0; i < scoreChars.length; i++) {
                if ((int)scoreChars[i] < 48 || (int)scoreChars[i] > 57 ) {
                    scoreIsInteger = false;
                }
            }
            for(int i = 0; i < parChars.length; i++) {
                if ((int)parChars[i] < 48 || (int)parChars[i] > 57 ) {
                    parIsInteger = false;
                }
            }

            if(scoreIsInteger && parIsInteger) {
                String toPar;
                int shotsUnder = 0;
                int netScore = (int)((Integer.parseInt(score) - Math.round(golfer1.getHandicap())) -
(Integer.parseInt(par)));

                if(Integer.parseInt(score) > 120){
```

```
            toPar = "tooHigh";
        }else if(Integer.parseInt(score) < 50){
            toPar = "tooLow";
        }else if(Integer.parseInt(par) < 54){
            toPar = "parLow";
        }else if(Integer.parseInt(par) > 90 ){
            toPar = "parHigh";
        }else if (netScore < 0) {
            toPar = "under";
            shotsUnder = -(netScore);
        }else if(netScore > 0){
            toPar = "over";
        }else{
            toPar = "level";
        }
        boolean[] cat = {false, false, false, false, false, false};
        if((int) golfer1.getHandicap() > 5.4 && (int) Math.round(golfer1.getHandicap()) < 11.5){
            cat[2] = true;
        }else if((int) Math.round(golfer1.getHandicap()) > 11.4 && (int) Math.round(golfer1.getHandicap())
< 20.5){
            cat[3] = true;
        }else if((int) Math.round(golfer1.getHandicap()) > 20.4 && (int) Math.round(golfer1.getHandicap())
< 27.5){
            cat[4] = true;
        }else if((int) Math.round(golfer1.getHandicap()) > 27.4){
            cat[5] = true;
        }else{
            cat[1] = true;
        }
```

```
            double currentHandicap = golfer1.getHandicap();


            if(toPar.equals("under")){
               handicapCalculator(currentHandicap, shotsUnder);
               newHandicap = Double.parseDouble(df2.format(newHandicap));
               new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer1, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
               frame.dispose();
            }else if(toPar.equals("over")){
               if(cat[1] && netScore > 1 ){
                  newHandicap = currentHandicap + 0.1;
                  newHandicap = Double.parseDouble(df2.format(newHandicap));
                  new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer1, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                  frame.dispose();
               }else if(cat[2] && netScore > 2 ){
                  newHandicap = currentHandicap + 0.1;
                  newHandicap = Double.parseDouble(df2.format(newHandicap));
                  new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer1, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                  frame.dispose();
               }else if(cat[3] && netScore > 3 ){
                  newHandicap = currentHandicap + 0.1;
                  newHandicap = Double.parseDouble(df2.format(newHandicap));
                  new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer1, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                  frame.dispose();
               }else if(cat[4] && netScore > 4 ){
                  newHandicap = currentHandicap + 0.1;
```

```
                    newHandicap = Double.parseDouble(df2.format(newHandicap));
                    new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer1, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                    frame.dispose();
                }else if(cat[5] && netScore > 5 ){
                    newHandicap = currentHandicap + 0.1;
                    newHandicap = Double.parseDouble(df2.format(newHandicap));
                    new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer1, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                    frame.dispose();
                }
            }else if(toPar.equals("tooHigh")){
                JOptionPane.showMessageDialog(frame, "The score you have entered exceeds the maximum
limit");
                new ChangeHandicap("Handicap Helper", 210, 350, golfer1);
                frame.dispose();
            }else if(toPar.equals("tooLow")){
                JOptionPane.showMessageDialog(frame, "The score you have entered exceeds the minimum
limit");
                new ChangeHandicap("Handicap Helper", 210, 350, golfer1);
                frame.dispose();
            }else if(toPar.equals("parLow")){
                JOptionPane.showMessageDialog(frame, "The course par you have entered exceeds the minimum
limit");
                new ChangeHandicap("Handicap Helper", 210, 350, golfer1);
                frame.dispose();
            }else if(toPar.equals("parHigh")){
                JOptionPane.showMessageDialog(frame, "The score you have entered exceeds the maximum
limit");
```

```
                new ChangeHandicap("Handicap Helper", 210, 350, golfer1);
                frame.dispose();
            }else{
                newHandicap = currentHandicap;
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer1, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }



        }else{
            JOptionPane.showMessageDialog(frame, "New Age must be a number");
        }
    });
  }
  private void buttonFunction2(){
    confirm.addActionListener(e -> {
        String score = scoreBox.getText();
        String par = parBox.getText();
        boolean parIsInteger = true;
        boolean scoreIsInteger = true;
        char[] scoreChars = score.toCharArray();
        char[] parChars = par.toCharArray();
        DecimalFormat df2 = new DecimalFormat("#.##");

        for(int i = 0; i < scoreChars.length; i++) {
            if ((int)scoreChars[i] < 48 || (int)scoreChars[i] > 57 ) {
                scoreIsInteger = false;
            }
```

```
        }
        for(int i = 0; i < parChars.length; i++) {
            if ((int)parChars[i] < 48 || (int)parChars[i] > 57 ) {
                parIsInteger = false;
            }
        }

        if(scoreIsInteger && parIsInteger) {
            String toPar;
            int shotsUnder = 0;
            int netScore = (int)((Integer.parseInt(score) - Math.round(golfer2.getHandicap())) -
(Integer.parseInt(par)));

            if(Integer.parseInt(score) > 120){
                toPar = "tooHigh";
            }else if(Integer.parseInt(score) < 50){
                toPar = "tooLow";
            }else if(Integer.parseInt(par) < 54){
                toPar = "parLow";
            }else if(Integer.parseInt(par) > 90 ){
                toPar = "parHigh";
            }else if (netScore < 0) {
                toPar = "under";
                shotsUnder = -(netScore);
            }else if(netScore > 0){
                toPar = "over";
            }else{
                toPar = "level";
            }
```

```
            boolean[] cat = {false, false, false, false, false, false};
            if((int) golfer2.getHandicap() > 5.4 && (int) Math.round(golfer2.getHandicap()) < 11.5){
               cat[2] = true;
            }else if((int) Math.round(golfer2.getHandicap()) > 11.4 && (int) Math.round(golfer2.getHandicap())
< 20.5){
               cat[3] = true;
            }else if((int) Math.round(golfer2.getHandicap()) > 20.4 && (int) Math.round(golfer2.getHandicap())
< 27.5){
               cat[4] = true;
            }else if((int) Math.round(golfer2.getHandicap()) > 27.4){
               cat[5] = true;
            }else{
               cat[1] = true;
            }
            double currentHandicap = golfer2.getHandicap();

            if(toPar.equals("under")){
               handicapCalculator(currentHandicap, shotsUnder);
               newHandicap = Double.parseDouble(df2.format(newHandicap));
               new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer2, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
               frame.dispose();
            }else if(toPar.equals("over")){
               if(cat[1] && netScore > 1 ){
                  newHandicap = currentHandicap + 0.1;
                  newHandicap = Double.parseDouble(df2.format(newHandicap));
                  new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer2, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                  frame.dispose();
```

```
            }else if(cat[2] && netScore > 2 ){
                newHandicap = currentHandicap + 0.1;
                newHandicap = Double.parseDouble(df2.format(newHandicap));
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer2, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }else if(cat[3] && netScore > 3 ){
                newHandicap = currentHandicap + 0.1;
                newHandicap = Double.parseDouble(df2.format(newHandicap));
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer2, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }else if(cat[4] && netScore > 4 ){
                newHandicap = currentHandicap + 0.1;
                newHandicap = Double.parseDouble(df2.format(newHandicap));
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer2, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }else if(cat[5] && netScore > 5 ){
                newHandicap = currentHandicap + 0.1;
                newHandicap = Double.parseDouble(df2.format(newHandicap));
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer2, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }
        }else if(toPar.equals("tooHigh")){
            JOptionPane.showMessageDialog(frame, "The score you have entered exceeds the maximum
limit");
            new ChangeHandicap("Handicap Helper", 210, 350, golfer2);
```

```
                frame.dispose();
            }else if(toPar.equals("tooLow")){
                JOptionPane.showMessageDialog(frame, "The score you have entered exceeds the minimum
limit");
                new ChangeHandicap("Handicap Helper", 210, 350, golfer2);
                frame.dispose();
            }else if(toPar.equals("parLow")){
                JOptionPane.showMessageDialog(frame, "The course par you have entered exceeds the minimum
limit");
                new ChangeHandicap("Handicap Helper", 210, 350, golfer2);
                frame.dispose();
            }else if(toPar.equals("parHigh")){
                JOptionPane.showMessageDialog(frame, "The score you have entered exceeds the maximum
limit");
                new ChangeHandicap("Handicap Helper", 210, 350, golfer2);
                frame.dispose();
            }else{
                newHandicap = currentHandicap;
                new ConfirmNewHandicap("Handicap Helper", 210, 350, golfer2, newHandicap,
Integer.parseInt(score), Integer.parseInt(par));
                frame.dispose();
            }



        }else{
            JOptionPane.showMessageDialog(frame, "New Age must be a number");
        }
      });
   }
```

```
private void cancel(){
    cancel.addActionListener(e -> {
        new Homepage("Handicap Helper", 210, 350, golfer);
        frame.dispose();
    });
}
private void cancel1(){
    cancel.addActionListener(e -> {
        new Homepage("Handicap Helper", 210, 350, golfer1);
        frame.dispose();
    });
}
private void cancel2(){
    cancel.addActionListener(e -> {
        new Homepage("Handicap Helper", 210, 350, golfer2);
        frame.dispose();
    });
}
private void handicapCalculator(double handicap, int shotsUnder) {
    if ((int) Math.round(handicap) < 6 && shotsUnder > 0) {
        handicap = handicap - 0.1;
        shotsUnder--;
        handicapCalculator(handicap, shotsUnder);
    } else if ((int) Math.round(handicap) > 5 && (int) Math.round(handicap) < 13 && shotsUnder > 0) {
        handicap = handicap - 0.2;
        shotsUnder--;
        handicapCalculator(handicap, shotsUnder);
    } else if ((int) Math.round(handicap) > 12 && (int) Math.round(handicap) < 21 && shotsUnder > 0) {
        handicap = handicap - 0.3;
```

```
            shotsUnder--;
            handicapCalculator(handicap, shotsUnder);
        } else if ((int) Math.round(handicap) > 20 && (int) Math.round(handicap) < 29 && shotsUnder > 0) {
            handicap = handicap - 0.4;
            shotsUnder--;
            handicapCalculator(handicap, shotsUnder);
        } else if ((int) Math.round(handicap) > 28 && shotsUnder > 0) {
            handicap = handicap - 0.5;
            shotsUnder--;
            handicapCalculator(handicap, shotsUnder);
        }else {
            newHandicap = handicap;

        }
    }
}
import javax.swing.*;
import java.awt.*;
import java.io.*;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;

public class ChangePassword {

    public JFrame frame;
    //Allows us to draw graphics to it
    public JPanel panel;
    private String title;
```

```
 private int width;
 private int height;
 private Golfer golfer;
 private BoyGolfer golfer1;
 private GirlGolfer golfer2;


ChangePassword(String title, int width, int height, Golfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer = golfer;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction();
 }


 ChangePassword(String title, int width, int height, BoyGolfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer1 = golfer;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
```

```
    buttonFunction1();
}

ChangePassword(String title, int width, int height, GirlGolfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer2 = golfer;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction2();
}

private JLabel confirm = new JLabel("Confirm your previous login details");
private JLabel user = new JLabel("Username:");
private JLabel pass = new JLabel("Password:");
private JLabel newPassword = new JLabel("Enter the password you would like");
private JButton cancel = new JButton("Cancel");
private JButton change = new JButton("Change");
private JPasswordField newPasswordBox = new JPasswordField();
private JTextField oldUsername = new JTextField();
private JPasswordField oldPassword = new JPasswordField();


private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
```

```
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}
private void addToDisplay(){
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    panel.add(confirm);
    panel.add(user);
    panel.add(oldUsername);
    panel.add(pass);
    panel.add(oldPassword);
    panel.add(newPassword);
    panel.add(newPasswordBox);
    panel.add(new JPanel());
    panel.add(change);
    panel.add(cancel);
```

```
        frame.add(panel);
        frame.pack();
    }

    private void buttonFunction(){
        change.addActionListener(e -> {
            String user = oldUsername.getText();
            String pass = String.valueOf(oldPassword.getPassword());
            String newPass = String.valueOf(newPasswordBox.getPassword());
            List<String[]> users = new ArrayList<>();
            File fileName = new File("users.txt");
            String line;
            String[] newArray = {golfer.getUsername(), golfer.getPassword(), golfer.getName(),
Integer.toString(golfer.getAge()), Double.toString(golfer.getHandicap()), golfer.getGender()};

            if (golfer.getUsername().equals(user) && golfer.getPassword().equals(pass)) {

                try (Scanner scanner = new Scanner(new FileReader(fileName))) {
                    while (scanner.hasNextLine()) {
                        line = scanner.nextLine();
                        String[] lines = line.split(", ", 6);
                        users.add(lines);
                        if (lines[0].equals(golfer.getUsername()) && lines[1].equals(golfer.getPassword())) {
                            golfer.setPassword(newPass);
                            newArray[0] = golfer.getUsername();
                            newArray[1] = golfer.getPassword();
                            newArray[2] = golfer.getName();
                            newArray[3] = Integer.toString(golfer.getAge());
                            newArray[4] = Double.toString(golfer.getHandicap());
```

```
        newArray[5] = golfer.getGender();
    }
}

FileWriter clearFile = new FileWriter(fileName, false);
PrintWriter clearPrint = new PrintWriter(clearFile);
clearPrint.flush();
clearPrint.close();

for (int i = 0; i < users.size(); i++) {
    if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer.getPassword())) {
        users.set(i, newArray);

        FileWriter fileWriter = new FileWriter(fileName, true);
        PrintWriter printWriter = new PrintWriter(fileWriter);
        String toFile = String.join(", ", newArray);
        printWriter.println(toFile);
        printWriter.close();

    } else {
        FileWriter fileWriter = new FileWriter(fileName, true);
        PrintWriter printWriter = new PrintWriter(fileWriter);
        String toFile2 = String.join(", ", users.get(i));
        printWriter.println(toFile2);
        printWriter.close();
    }
}
new Homepage("Handicap Helper", 210, 350, golfer);
frame.dispose();
```

```
            } catch (Exception e1) {
                new ChangeUsername("Handicap Helper", 210, 350, golfer);
            }
        } else {
            JOptionPane.showMessageDialog(frame, "Change Password failed! Try Again!");
            new ChangeUsername("Handicap Helper", 210, 350, golfer);
        }

    });
    cancel.addActionListener(e -> {
        new Homepage("Handicap Helper", 210, 350, golfer);
        frame.dispose();
    });
}
  private void buttonFunction1(){
    change.addActionListener(e -> {
        String user = oldUsername.getText();
        String pass = String.valueOf(oldPassword.getPassword());
        String newPass = String.valueOf(newPasswordBox.getPassword());
        List<String[]> users = new ArrayList<>();
        File fileName = new File("users.txt");
        String line;
        String[] newArray = {golfer1.getUsername(), golfer1.getPassword(), golfer1.getName(),
Integer.toString(golfer1.getAge()), Double.toString(golfer1.getHandicap()), golfer1.getGender()};

        if (golfer1.getUsername().equals(user) && golfer1.getPassword().equals(pass)) {

            try (Scanner scanner = new Scanner(new FileReader(fileName))) {
```

```
while (scanner.hasNextLine()) {
    line = scanner.nextLine();
    String[] lines = line.split(", ", 6);
    users.add(lines);
    if (lines[0].equals(golfer1.getUsername()) && lines[1].equals(golfer1.getPassword())) {
        golfer1.setPassword(newPass);
        newArray[0] = golfer1.getUsername();
        newArray[1] = golfer1.getPassword();
        newArray[2] = golfer1.getName();
        newArray[3] = Integer.toString(golfer1.getAge());
        newArray[4] = Double.toString(golfer1.getHandicap());
        newArray[5] = golfer1.getGender();
    }
}

FileWriter clearFile = new FileWriter(fileName, false);
PrintWriter clearPrint = new PrintWriter(clearFile);
clearPrint.flush();
clearPrint.close();

for (int i = 0; i < users.size(); i++) {
    if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer1.getPassword())) {
        users.set(i, newArray);

        FileWriter fileWriter = new FileWriter(fileName, true);
        PrintWriter printWriter = new PrintWriter(fileWriter);
        String toFile = String.join(", ", newArray);
        printWriter.println(toFile);
        printWriter.close();
```

```
                } else {
                    FileWriter fileWriter = new FileWriter(fileName, true);
                    PrintWriter printWriter = new PrintWriter(fileWriter);
                    String toFile2 = String.join(", ", users.get(i));
                    printWriter.println(toFile2);
                    printWriter.close();
                }
            }
            new Homepage("Handicap Helper", 210, 350, golfer1);
            frame.dispose();

        } catch (Exception e1) {
            new ChangeUsername("Handicap Helper", 210, 350, golfer1);
        }
    } else {
        JOptionPane.showMessageDialog(frame, "Change Password failed! Try Again!");
        new ChangeUsername("Handicap Helper", 210, 350, golfer1);
    }

  });
  cancel.addActionListener(e -> {
    new Homepage("Handicap Helper", 210, 350, golfer1);
    frame.dispose();
  });
}
private void buttonFunction2(){
  change.addActionListener(e -> {
    String user = oldUsername.getText();
```

```
        String pass = String.valueOf(oldPassword.getPassword());
        String newPass = String.valueOf(newPasswordBox.getPassword());
        List<String[]> users = new ArrayList<>();
        File fileName = new File("users.txt");
        String line;
        String[] newArray = {golfer2.getUsername(), golfer2.getPassword(), golfer2.getName(),
Integer.toString(golfer2.getAge()), Double.toString(golfer2.getHandicap()), golfer2.getGender()};

        if (golfer2.getUsername().equals(user) && golfer2.getPassword().equals(pass)) {

            try (Scanner scanner = new Scanner(new FileReader(fileName))) {
                while (scanner.hasNextLine()) {
                    line = scanner.nextLine();
                    String[] lines = line.split(", ", 6);
                    users.add(lines);
                    if (lines[0].equals(golfer2.getUsername()) && lines[1].equals(golfer2.getPassword())) {
                        golfer2.setPassword(newPass);
                        newArray[0] = golfer2.getUsername();
                        newArray[1] = golfer2.getPassword();
                        newArray[2] = golfer2.getName();
                        newArray[3] = Integer.toString(golfer2.getAge());
                        newArray[4] = Double.toString(golfer2.getHandicap());
                        newArray[5] = golfer2.getGender();
                    }
                }

                FileWriter clearFile = new FileWriter(fileName, false);
                PrintWriter clearPrint = new PrintWriter(clearFile);
                clearPrint.flush();
```

```
            clearPrint.close();

            for (int i = 0; i < users.size(); i++) {
                if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer2.getPassword())) {
                    users.set(i, newArray);

                    FileWriter fileWriter = new FileWriter(fileName, true);
                    PrintWriter printWriter = new PrintWriter(fileWriter);
                    String toFile = String.join(", ", newArray);
                    printWriter.println(toFile);
                    printWriter.close();

                } else {
                    FileWriter fileWriter = new FileWriter(fileName, true);
                    PrintWriter printWriter = new PrintWriter(fileWriter);
                    String toFile2 = String.join(", ", users.get(i));
                    printWriter.println(toFile2);
                    printWriter.close();
                }
            }
            new Homepage("Handicap Helper", 210, 350, golfer2);
            frame.dispose();

        } catch (Exception e1) {
            new ChangeUsername("Handicap Helper", 210, 350, golfer2);
        }
    } else {
        JOptionPane.showMessageDialog(frame, "Change Password failed! Try Again!");
        new ChangeUsername("Handicap Helper", 210, 350, golfer2);
```

```
            }


        });
        cancel.addActionListener(e -> {
            new Homepage("Handicap Helper", 210, 350, golfer2);
            frame.dispose();
        });
    }
}
import javax.swing.*;
import java.awt.*;
import java.io.*;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;

public class ChangeUsername {

    public JFrame frame;
    public JPanel panel;

    private String title;
    private int width;
    private int height;
    private Golfer golfer;
    private BoyGolfer golfer1;
    private GirlGolfer golfer2;

    ChangeUsername(String title, int width, int height, BoyGolfer golfer){
```

```
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer1 = golfer;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction1();
}

ChangeUsername(String title, int width, int height, GirlGolfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer2 = golfer;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction2();
}

ChangeUsername(String title, int width, int height, Golfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer = golfer;
```

```
    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction();
}



private JLabel confirm = new JLabel("Confirm your previous login details");
private JLabel user = new JLabel("Username:");
private JLabel pass = new JLabel("Password:");
private JLabel newUsername = new JLabel("Enter the username you would like");
private JButton cancel = new JButton("Cancel");
private JButton change = new JButton("Change");
private JTextField newUsernameBox = new JTextField();
private JTextField oldUsername = new JTextField();
private JPasswordField oldPassword = new JPasswordField();



private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
```

```
    //Making sure the window is actually visible
    frame.setVisible(true);
}
    private void addToDisplay(){
        panel = new JPanel();
        //set the size of the JPanel (= to size of JFrame)
        panel.setPreferredSize(new Dimension(width, height));
        //Making sure you can't change the size of the JPanel
        panel.setMaximumSize(new Dimension(width, height));
        panel.setMinimumSize(new Dimension(width, height));
        //Setting the layout for the JPanel
        panel.setLayout(new GridLayout(12, 5));
        panel.add(confirm);
        panel.add(user);
        panel.add(oldUsername);
        panel.add(pass);
        panel.add(oldPassword);
        panel.add(newUsername);
        panel.add(newUsernameBox);
        panel.add(new JPanel());
        panel.add(change);
        panel.add(cancel);
        frame.add(panel);
        frame.pack();
    }

    private void buttonFunction(){
        change.addActionListener(e -> {
            String user = oldUsername.getText();
```

```
            String pass = String.valueOf(oldPassword.getPassword());
            String newUser = newUsernameBox.getText();
            List<String[]> users = new ArrayList<>();
            File fileName = new File("users.txt");
            String line;
            String[] newArray = {golfer.getUsername(), golfer.getPassword(), golfer.getName(),
Integer.toString(golfer.getAge()), Double.toString(golfer.getHandicap()), golfer.getGender()};
            boolean userTaken = false;
            if (golfer.getUsername().equals(user) && golfer.getPassword().equals(pass)) {

               try (Scanner scanner = new Scanner(new FileReader(fileName))) {
                  while (scanner.hasNextLine()) {
                     line = scanner.nextLine();
                     String[] lines = line.split(", ", 6);
                     users.add(lines);
                     if (newUser.equals(lines[0])) {
                        userTaken = true;
                     }
                     if (lines[0].equals(golfer.getUsername()) && lines[1].equals(golfer.getPassword()) &&
!userTaken) {
                        golfer.setUsername(newUser);
                        newArray[0] = golfer.getUsername();
                        newArray[1] = golfer.getPassword();
                        newArray[2] = golfer.getName();
                        newArray[3] = Integer.toString(golfer.getAge());
                        newArray[4] = Double.toString(golfer.getHandicap());
                        newArray[5] = golfer.getGender();
                     }
                  }
```

```
            if (userTaken) {
                JOptionPane.showMessageDialog(frame, "This username is taken. Please try another");
                new ChangeUsername("Handicap Helper", 210, 350, golfer);
                frame.dispose();
            }
            FileWriter clearFile = new FileWriter(fileName, false);
            PrintWriter clearPrint = new PrintWriter(clearFile);
            clearPrint.flush();
            clearPrint.close();

            for (int i = 0; i < users.size(); i++) {
                if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer.getPassword())&& !userTaken)
{
                    users.set(i, newArray);

                    FileWriter fileWriter = new FileWriter(fileName, true);
                    PrintWriter printWriter = new PrintWriter(fileWriter);
                    String toFile = String.join(", ", newArray);
                    printWriter.println(toFile);
                    printWriter.close();

                } else {
                    FileWriter fileWriter = new FileWriter(fileName, true);
                    PrintWriter printWriter = new PrintWriter(fileWriter);
                    String toFile2 = String.join(", ", users.get(i));
                    printWriter.println(toFile2);
                    printWriter.close();
                }
            }
```

```
                if(!userTaken) {
                    new Homepage("Handicap Helper", 210, 350, golfer);
                    frame.dispose();
                }
            } catch (Exception e1) {
                new ChangeUsername("Handicap Helper", 210, 350, golfer);
                frame.dispose();
            }
        } else {
            JOptionPane.showMessageDialog(frame, "Change Username failed! Try Again!");
            new ChangeUsername("Handicap Helper", 210, 350, golfer);
            frame.dispose();
        }

    });
    cancel.addActionListener(e -> {
        new Homepage("Handicap Helper", 210, 350, golfer);
        frame.dispose();
    });
  }

private void buttonFunction1(){
   change.addActionListener(e -> {
      String user = oldUsername.getText();
      String pass = String.valueOf(oldPassword.getPassword());
      String newUser = newUsernameBox.getText();
      List<String[]> users = new ArrayList<>();
      File fileName = new File("users.txt");
      String line;
```

```
        String[] newArray = {golfer1.getUsername(), golfer1.getPassword(), golfer1.getName(),
Integer.toString(golfer1.getAge()), Double.toString(golfer1.getHandicap()), golfer1.getGender()};
        boolean userTaken = false;
        if (golfer1.getUsername().equals(user) && golfer1.getPassword().equals(pass)) {

           try (Scanner scanner = new Scanner(new FileReader(fileName))) {
              while (scanner.hasNextLine()) {
                 line = scanner.nextLine();
                 String[] lines = line.split(", ", 6);
                 users.add(lines);
                 if (newUser.equals(lines[0])) {
                    userTaken = true;
                 }
                 if (lines[0].equals(golfer1.getUsername()) && lines[1].equals(golfer1.getPassword()) &&
!userTaken) {
                    golfer1.setUsername(newUser);
                    newArray[0] = golfer1.getUsername();
                    newArray[1] = golfer1.getPassword();
                    newArray[2] = golfer1.getName();
                    newArray[3] = Integer.toString(golfer1.getAge());
                    newArray[4] = Double.toString(golfer1.getHandicap());
                    newArray[5] = golfer1.getGender();
                 }
              }
              if (userTaken) {
                 JOptionPane.showMessageDialog(frame, "This username is taken. Please try another");
                 new ChangeUsername("Handicap Helper", 210, 350, golfer1);
                 frame.dispose();
              }
```

```
            FileWriter clearFile = new FileWriter(fileName, false);
            PrintWriter clearPrint = new PrintWriter(clearFile);
            clearPrint.flush();
            clearPrint.close();

            for (int i = 0; i < users.size(); i++) {
               if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer1.getPassword())&& !userTaken)
{
                  users.set(i, newArray);

                  FileWriter fileWriter = new FileWriter(fileName, true);
                  PrintWriter printWriter = new PrintWriter(fileWriter);
                  String toFile = String.join(", ", newArray);
                  printWriter.println(toFile);
                  printWriter.close();

               } else {
                  FileWriter fileWriter = new FileWriter(fileName, true);
                  PrintWriter printWriter = new PrintWriter(fileWriter);
                  String toFile2 = String.join(", ", users.get(i));
                  printWriter.println(toFile2);
                  printWriter.close();
               }
            }
            if(!userTaken) {
               new Homepage("Handicap Helper", 210, 350, golfer1);
               frame.dispose();
            }
         } catch (Exception e1) {
```

```
                new ChangeUsername("Handicap Helper", 210, 350, golfer1);
                frame.dispose();
            }
        } else {
            JOptionPane.showMessageDialog(frame, "Change Username failed! Try Again!");
            new ChangeUsername("Handicap Helper", 210, 350, golfer1);
            frame.dispose();
        }
    });
    cancel.addActionListener(e -> {
        new Homepage("Handicap Helper", 210, 350, golfer1);
        frame.dispose();
    });
}
  private void buttonFunction2(){
    change.addActionListener(e -> {
        String user = oldUsername.getText();
        String pass = String.valueOf(oldPassword.getPassword());
        String newUser = newUsernameBox.getText();
        List<String[]> users = new ArrayList<>();
        File fileName = new File("users.txt");
        String line;
        String[] newArray = {golfer2.getUsername(), golfer2.getPassword(), golfer2.getName(),
Integer.toString(golfer2.getAge()), Double.toString(golfer2.getHandicap()), golfer2.getGender()};
        boolean userTaken = false;
        if (golfer2.getUsername().equals(user) && golfer2.getPassword().equals(pass)) {

            try (Scanner scanner = new Scanner(new FileReader(fileName))) {
                while (scanner.hasNextLine()) {
```

```
                line = scanner.nextLine();
                String[] lines = line.split(", ", 6);
                users.add(lines);
                if (newUser.equals(lines[0])) {
                    userTaken = true;
                }
                if (lines[0].equals(golfer2.getUsername()) && lines[1].equals(golfer2.getPassword()) &&
!userTaken) {
                    golfer2.setUsername(newUser);
                    newArray[0] = golfer2.getUsername();
                    newArray[1] = golfer2.getPassword();
                    newArray[2] = golfer2.getName();
                    newArray[3] = Integer.toString(golfer2.getAge());
                    newArray[4] = Double.toString(golfer2.getHandicap());
                    newArray[5] = golfer2.getGender();
                }
            }
            if (userTaken) {
                JOptionPane.showMessageDialog(frame, "This username is taken. Please try another");
                new ChangeUsername("Handicap Helper", 210, 350, golfer2);
                frame.dispose();
            }

            FileWriter clearFile = new FileWriter(fileName, false);
            PrintWriter clearPrint = new PrintWriter(clearFile);
            clearPrint.flush();
            clearPrint.close();

            for (int i = 0; i < users.size(); i++) {
```

```
            if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer2.getPassword())&&!userTaken) {
                users.set(i, newArray);

                FileWriter fileWriter = new FileWriter(fileName, true);
                PrintWriter printWriter = new PrintWriter(fileWriter);
                String toFile = String.join(", ", newArray);
                printWriter.println(toFile);
                printWriter.close();

            } else {
                FileWriter fileWriter = new FileWriter(fileName, true);
                PrintWriter printWriter = new PrintWriter(fileWriter);
                String toFile2 = String.join(", ", users.get(i));
                printWriter.println(toFile2);
                printWriter.close();
            }
        }
        if(!userTaken) {
            new Homepage("Handicap Helper", 210, 350, golfer2);
            frame.dispose();
        }
    } catch (Exception e1) {
        new ChangeUsername("Handicap Helper", 210, 350, golfer2);
        frame.dispose();
    }
} else {
    JOptionPane.showMessageDialog(frame, "Change Username failed! Try Again!");
    new ChangeUsername("Handicap Helper", 210, 350, golfer2);
    frame.dispose();
```

```
            }


        });
        cancel.addActionListener(e -> {
            new Homepage("Handicap Helper", 210, 350, golfer2);
            frame.dispose();
        });
    }
  }
import javax.swing.*;
import java.awt.*;

class Confirm {

    //Creating a window (so that we can see the game)
    private JFrame frame;

    private JButton confirm = new JButton("Confirm");
private JButton back = new JButton("Back");

    private String title;
    private int width;
    private int height;
    private int totalScore;
    private int totalPar;
    private int totalAdScore;
    Golfer golfer;

    Confirm(String title, int width, int height, int totalAdScore, int totalPar, int totalScore, Golfer golfer) {
```

```
    this.title = title;
    this.width = width;
    this.height = height;
    this.totalPar = totalPar;
    this.totalScore = totalScore;
    this.totalAdScore = totalAdScore;
    this.golfer = golfer;
    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction();
}

//Initialising the JFrame and the canvas in a new method
private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}
private void addToDisplay(){
    JPanel panel;
```

```
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    panel.add(new JLabel("Par: " + totalPar));
    panel.add(new JLabel("Score: " + totalScore));
    panel.add(new JLabel("Adjusted Score: " + totalAdScore));
    //Adding the menu label to the canvas
    panel.add(confirm);
    panel.add(back);
    frame.add(panel);
    frame.pack();
}

private void buttonFunction(){
    confirm.addActionListener(e ->  {
    new Scorecard2("Handicap Helper", 210, 350, golfer);
    frame.dispose();
    golfer.setAdScores1(totalAdScore);
    golfer.setPar1(totalPar);
});
    back.addActionListener(e ->  {
        new Scorecard1("Handicap Helper", 210, 350, golfer);
        frame.dispose();
    });
```

```
    }
}

import javax.swing.*;
import java.awt.*;

class Confirm2 {

    //Creating a window (so that we can see the game)
    private JFrame frame;

    private JButton confirm = new JButton("Confirm");

    private JButton back = new JButton ("Back");


    private String title;
    private int width;
    private int height;
    private int totalScore;
    private int totalPar;
    private int totalAdScore;
    Golfer golfer;

    Confirm2(String title, int width, int height, int totalAdScore, int totalPar, int totalScore, Golfer golfer) {
        this.title = title;
        this.width = width;
        this.height = height;
        this.totalPar = totalPar;
```

```
    this.totalScore = totalScore;
    this.totalAdScore = totalAdScore;
    this.golfer = golfer;
    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction();
}

//Initialising the JFrame and the canvas in a new method
private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}

private void addToDisplay(){
    JPanel panel;
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
```

```
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    panel.add(new JLabel("Par: " + totalPar));
    panel.add(new JLabel("Score: " + totalScore));
    panel.add(new JLabel("Adjusted Score: " + totalAdScore));
    //Adding the menu label to the canvas
    panel.add(confirm);
    panel.add(back);
    frame.add(panel);
    frame.pack();
}

private void buttonFunction(){
    confirm.addActionListener(e ->  {
        golfer.setAdScores2(totalAdScore);
        golfer.setPar2(totalPar);
        golfer.setHandicap();
        //new SignUp
        new Confirm3("Handicap Helper", 210, 350, golfer);
        frame.dispose();

    });
    back.addActionListener(e ->  {
        new Scorecard2("Handicap Helper", 210, 350, golfer);
        frame.dispose();
    });
```

```
    }
}

import javax.swing.*;
import java.awt.*;

class Confirm3 {

    //Creating a window (so that we can see the game)
    private JFrame frame;

    private JButton confirm = new JButton("Confirm");

    private JButton back = new JButton ("Back");


    private String title;
    private int width;
    private int height;
    Golfer golfer;

    Confirm3(String title, int width, int height, Golfer golfer) {
        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer = golfer;
        //Calling the method to initialise the JFrame
        createDisplay();
        addToDisplay();
```

```
    buttonFunction();
}

//Initialising the JFrame and the canvas in a new method
private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}

private void addToDisplay(){
    JPanel panel;
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    panel.add(new JLabel("Par: " + golfer.getTotalPar()));
```

```
        panel.add(new JLabel("Adjusted Score: " + golfer.getTotalScores()));
        //Adding the menu label to the canvas
        panel.add(confirm);
        panel.add(back);
        frame.add(panel);
        frame.pack();
    }

    private void buttonFunction(){
        confirm.addActionListener(e ->  {
            new SignUp("Handicap Helper", 210, 350,golfer);
            frame.dispose();

        });
        back.addActionListener(e ->  {
            new Scorecard1("Handicap Helper", 210, 350, golfer);
            frame.dispose();
        });
    }
}

import javax.swing.*;
import java.awt.*;
import java.io.*;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.List;

public class ConfirmNewHandicap {
```

```
    public JFrame frame;
    //Allows us to draw graphics to it
    public JPanel panel;



    private String title;
    private int width;
    private int height;
    private Golfer golfer;
    private BoyGolfer golfer1;
    private GirlGolfer golfer2;
    private double newHandicap;
    private int score;
    private int par;



    ConfirmNewHandicap(String title, int width, int height, Golfer golfer, double newHandicap, int score, int par)
{
        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer = golfer;
        this.newHandicap = newHandicap;
        this.score = score;
        this.par = par;



        //Calling the method to initialise the JFrame
```

```
        createDisplay();
        addToDisplay();
        buttonFunction();
    }

    ConfirmNewHandicap(String title, int width, int height, BoyGolfer golfer, double newHandicap, int score, int
par) {
        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer1 = golfer;
        this.newHandicap = newHandicap;
        this.score = score;
        this.par = par;



        //Calling the method to initialise the JFrame
        createDisplay();
        addToDisplay();
        buttonFunction1();
    }

    ConfirmNewHandicap(String title, int width, int height, GirlGolfer golfer, double newHandicap, int score, int
par) {
        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer2 = golfer;
        this.newHandicap = newHandicap;
```

```
    this.score = score;
    this.par = par;



    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction2();
}



private JButton cancel = new JButton("Cancel");
private JButton change = new JButton("Confirm");
private JTextField oldUsername = new JTextField();
private JPasswordField oldPassword = new JPasswordField();



private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
```

```
}

private void addToDisplay() {
    String actualHandicap;
    if (newHandicap > 0.4) {
        actualHandicap = Double.toString(newHandicap);
    } else if (newHandicap < 0 && newHandicap > -0.5) {
        actualHandicap = ("Scratch (" + ("+" + (-newHandicap)) + ")");
    } else if (newHandicap < 0.5 && newHandicap >= 0) {
        actualHandicap = ("Scratch (" + newHandicap + ")");
    } else {
        actualHandicap = ("+" + (-newHandicap));
    }
    JLabel areYouSure1 = new JLabel("Are you sure you want to confirm ");
    JLabel areYouSure2 = new JLabel("this score.");
    JLabel scoreLabel = new JLabel("You shot a " + score + " on a par " + par);
    JLabel yourNewHandicap = new JLabel("Your new Handicap: " + actualHandicap);

    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    panel.add(areYouSure1);
    panel.add(areYouSure2);
    panel.add(scoreLabel);
```

```
        panel.add(yourNewHandicap);
        panel.add(new JLabel("Enter your details to confirm"));
        panel.add(new JLabel("Username:"));
        panel.add(oldUsername);
        panel.add(new JLabel("Password:"));
        panel.add(oldPassword);
        panel.add(change);
        panel.add(cancel);
        frame.add(panel);
        frame.pack();
    }

    private void buttonFunction() {
        change.addActionListener(e -> {

            String user = oldUsername.getText();
            String pass = String.valueOf(oldPassword.getPassword());



            List<String[]> users = new ArrayList<>();
            File fileName = new File("users.txt");
            String line;
            String[] newArray = {golfer.getUsername(), golfer.getPassword(), golfer.getName(),
Integer.toString(golfer.getAge()), Double.toString(golfer.getHandicap()), golfer.getGender()};

            if (golfer.getUsername().equals(user) && golfer.getPassword().equals(pass)) {

                try (Scanner scanner = new Scanner(new FileReader(fileName))) {
                    while (scanner.hasNextLine()) {
```

```
    line = scanner.nextLine();
    String[] lines = line.split(", ", 6);
    users.add(lines);
    if (lines[0].equals(golfer.getUsername()) && lines[1].equals(golfer.getPassword())) {
        golfer.setHandicap(newHandicap);
        newArray[0] = golfer.getUsername();
        newArray[1] = golfer.getPassword();
        newArray[2] = golfer.getName();
        newArray[3] = Integer.toString(golfer.getAge());
        newArray[4] = Double.toString(golfer.getHandicap());
        newArray[5] = golfer.getGender();
    }
}

FileWriter clearFile = new FileWriter(fileName, false);
PrintWriter clearPrint = new PrintWriter(clearFile);
clearPrint.flush();
clearPrint.close();

for (int i = 0; i < users.size(); i++) {
    if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer.getPassword())) {
        users.set(i, newArray);

        FileWriter fileWriter = new FileWriter(fileName, true);
        PrintWriter printWriter = new PrintWriter(fileWriter);
        String toFile = String.join(", ", newArray);
        printWriter.println(toFile);
        printWriter.close();
```

```
                } else {
                    FileWriter fileWriter = new FileWriter(fileName, true);
                    PrintWriter printWriter = new PrintWriter(fileWriter);
                    String toFile2 = String.join(", ", users.get(i));
                    printWriter.println(toFile2);
                    printWriter.close();
                }
            }
            new Homepage("Handicap Helper", 210, 350, golfer);
            frame.dispose();

        } catch (Exception e1) {
            new ChangeUsername("Handicap Helper", 210, 350, golfer);
        }
    } else {
        JOptionPane.showMessageDialog(frame, "Change Handicap failed! Try Again!");
    }
});

cancel.addActionListener(e -> {
    new Homepage("Handicap Helper", 210, 350, golfer);
    frame.dispose();
});
}

private void buttonFunction1() {
    change.addActionListener(e -> {

        String user = oldUsername.getText();
```

```
        String pass = String.valueOf(oldPassword.getPassword());



        List<String[]> users = new ArrayList<>();
        File fileName = new File("users.txt");
        String line;
        String[] newArray = {golfer1.getUsername(), golfer1.getPassword(), golfer1.getName(),
Integer.toString(golfer1.getAge()), Double.toString(golfer1.getHandicap()), golfer1.getGender()};

        if (golfer1.getUsername().equals(user) && golfer1.getPassword().equals(pass)) {

            try (Scanner scanner = new Scanner(new FileReader(fileName))) {
                while (scanner.hasNextLine()) {
                    line = scanner.nextLine();
                    String[] lines = line.split(", ", 6);
                    users.add(lines);
                    if (lines[0].equals(golfer1.getUsername()) && lines[1].equals(golfer1.getPassword())) {
                        golfer1.setHandicap(newHandicap);
                        newArray[0] = golfer1.getUsername();
                        newArray[1] = golfer1.getPassword();
                        newArray[2] = golfer1.getName();
                        newArray[3] = Integer.toString(golfer1.getAge());
                        newArray[4] = Double.toString(golfer1.getHandicap());
                        newArray[5] = golfer1.getGender();
                    }
                }

                FileWriter clearFile = new FileWriter(fileName, false);
                PrintWriter clearPrint = new PrintWriter(clearFile);
```

```
            clearPrint.flush();
            clearPrint.close();

            for (int i = 0; i < users.size(); i++) {
                if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer1.getPassword())) {
                    users.set(i, newArray);

                    FileWriter fileWriter = new FileWriter(fileName, true);
                    PrintWriter printWriter = new PrintWriter(fileWriter);
                    String toFile = String.join(", ", newArray);
                    printWriter.println(toFile);
                    printWriter.close();

                } else {
                    FileWriter fileWriter = new FileWriter(fileName, true);
                    PrintWriter printWriter = new PrintWriter(fileWriter);
                    String toFile2 = String.join(", ", users.get(i));
                    printWriter.println(toFile2);
                    printWriter.close();
                }
            }
            new Homepage("Handicap Helper", 210, 350, golfer1);
            frame.dispose();

        } catch (Exception e1) {
            new ChangeUsername("Handicap Helper", 210, 350, golfer1);
        }
    } else {
        JOptionPane.showMessageDialog(frame, "Change Handicap failed! Try Again!");
```

```
        }
    });

    cancel.addActionListener(e -> {
        new Homepage("Handicap Helper", 210, 350, golfer1);
        frame.dispose();
    });
}

  private void buttonFunction2(){
      change.addActionListener(e -> {

          String user = oldUsername.getText();
          String pass = String.valueOf(oldPassword.getPassword());



          List<String[]> users = new ArrayList<>();
          File fileName = new File("users.txt");
          String line;
          String[] newArray = {golfer2.getUsername(), golfer2.getPassword(), golfer2.getName(),
Integer.toString(golfer2.getAge()), Double.toString(golfer2.getHandicap()), golfer2.getGender()};

          if (golfer2.getUsername().equals(user) && golfer2.getPassword().equals(pass)) {

              try (Scanner scanner = new Scanner(new FileReader(fileName))) {
                  while (scanner.hasNextLine()) {
                      line = scanner.nextLine();
                      String[] lines = line.split(", ", 6);
                      users.add(lines);
```

```
    if (lines[0].equals(golfer2.getUsername()) && lines[1].equals(golfer2.getPassword())) {
        golfer2.setHandicap(newHandicap);
        newArray[0] = golfer2.getUsername();
        newArray[1] = golfer2.getPassword();
        newArray[2] = golfer2.getName();
        newArray[3] = Integer.toString(golfer2.getAge());
        newArray[4] = Double.toString(golfer2.getHandicap());
        newArray[5] = golfer2.getGender();
    }
}

FileWriter clearFile = new FileWriter(fileName, false);
PrintWriter clearPrint = new PrintWriter(clearFile);
clearPrint.flush();
clearPrint.close();

for (int i = 0; i < users.size(); i++) {
    if (users.get(i)[0].equals(user) && users.get(i)[1].equals(golfer2.getPassword())) {
        users.set(i, newArray);

        FileWriter fileWriter = new FileWriter(fileName, true);
        PrintWriter printWriter = new PrintWriter(fileWriter);
        String toFile = String.join(", ", newArray);
        printWriter.println(toFile);
        printWriter.close();

    } else {
        FileWriter fileWriter = new FileWriter(fileName, true);
        PrintWriter printWriter = new PrintWriter(fileWriter);
```

```
                    String toFile2 = String.join(", ", users.get(i));
                    printWriter.println(toFile2);
                    printWriter.close();
                }
            }
            new Homepage("Handicap Helper", 210, 350, golfer2);
            frame.dispose();

        } catch (Exception e1) {
            new ChangeUsername("Handicap Helper", 210, 350, golfer2);
        }
      } else {
        JOptionPane.showMessageDialog(frame, "Change Handicap failed! Try Again!");
      }
    });

    cancel.addActionListener(e -> {
      new Homepage("Handicap Helper", 210, 350, golfer2);
      frame.dispose();
    });
  }
   }

import javax.swing.*;
import java.awt.*;


class Gender{
```

```
private JFrame frame;
//Allows us to draw graphics to it

private String title;
private int width;
private int height;

Gender(String title, int width, int height){

    this.title = title;
    this.width = width;
    this.height = height;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction();

}

//Label is just writing ("on the canvas")
private String[] genderArray = {"Male" , "Female", "Other", "Rather not say" };
private JLabel genderLabel = new JLabel("Please specify your gender");
private JComboBox gender = new JComboBox(genderArray);
private JButton okay = new JButton("Okay");
private JButton cancel = new JButton("Cancel");


//Initialising the JFrame and the canvas in a new method
```

```
private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}
private void addToDisplay(){
    JPanel panel;
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    //Adding the menu label to the canvas
    panel.add(genderLabel);
    panel.add(new JPanel());
    panel.add(gender);
    panel.add(new JPanel());
    panel.add(new JPanel());
```

```
        panel.add(new JPanel());
        panel.add(okay);
        panel.add(cancel);
        //adding the canvas to the JFrame and making sure its visible
        frame.add(panel);
        frame.pack();
    }

    private void buttonFunction(){
        okay.addActionListener(e -> {
            String userGender = (String)gender.getSelectedItem();
            Golfer golfer = new Golfer(userGender);
            new Name("Handicap Helper", 210, 350, golfer);
            frame.dispose();
        });
        cancel.addActionListener(e -> {
            new Menu("Handicap Helper", 210, 350);
            frame.dispose();
        });
    }
}

import javax.swing.*;
import java.awt.*;


public class Homepage {

    public JFrame frame;
```

```
//Allows us to draw graphics to it
public JPanel panel;



private String title;
private int width;
private int height;
private Golfer golfer;
private BoyGolfer golfer1;
private GirlGolfer golfer2;
private String handicap;
private int age;
private String name;
private String gender;
private JPanel space = new JPanel();

Homepage(String title, int width, int height, Golfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer = golfer;
    if (golfer.getHandicap() > 0.4) {
        this.handicap = Double.toString(golfer.getHandicap());
    }else if (golfer.getHandicap() < 0 && golfer.getHandicap() > -0.5){
        this.handicap = ("Scratch (" + ("+" + (-golfer.getHandicap())) + ")");
    }else if (golfer.getHandicap() < 0.5 && golfer.getHandicap() >= 0){
        this.handicap = ("Scratch (" + golfer.getHandicap()+")");
    }else{
        this.handicap = ("+" + (-golfer.getHandicap()));
```

```
    }
    this.age = golfer.getAge();
    this.name = golfer.getName();
    this.gender = golfer.getGender();
    //Calling the method to initialise the JFrame
    createDisplay();
    panel = new JPanel();
    addToDisplay();
    buttonFunction();
}
Homepage(String title, int width, int height, BoyGolfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer1 = golfer;
    if (golfer.getHandicap() > 0.4) {
        this.handicap = Double.toString(golfer.getHandicap());
    }else if (golfer.getHandicap() < 0 && golfer.getHandicap() > -0.5){
        this.handicap = ("Scratch (" + ("+" + (-golfer.getHandicap())) + ")");
    }else if (golfer.getHandicap() < 0.5 && golfer.getHandicap() >= 0){
        this.handicap = ("Scratch (" + golfer.getHandicap()+")");
    }else{
        this.handicap = ("+" + (-golfer.getHandicap()));
    }
    this.age = golfer.getAge();
    this.name = golfer.getName();
    this.gender = golfer.getGender();
    //Calling the method to initialise the JFrame
    createDisplay();
```

```
    space.setBackground(new Color(golfer1.getRed(), golfer1.getGreen(), golfer1.getBlue()));
    panel = new JPanel();
    panel.add(new JLabel(golfer.getInfo()));
    panel.add(new JLabel(golfer.getInfo1()));
    addToDisplay();
    panel.setBackground(new Color(golfer1.getRed(), golfer1.getGreen(), golfer1.getBlue()));
    buttonFunction1();
}

Homepage(String title, int width, int height, GirlGolfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer2 = golfer;
    if (golfer.getHandicap() > 0.4) {
        this.handicap = Double.toString(golfer.getHandicap());
    }else if (golfer.getHandicap() < 0 && golfer.getHandicap() > -0.5){
        this.handicap = ("Scratch (" + ("+" + (-golfer.getHandicap())) + ")");
    }else if (golfer.getHandicap() < 0.5 && golfer.getHandicap() >= 0){
        this.handicap = ("Scratch (" + golfer.getHandicap()+")");
    }else{
        this.handicap = ("+" + (-golfer.getHandicap()));
    }
    this.age = golfer.getAge();
    this.name = golfer.getName();
    this.gender = golfer.getGender();
    //Calling the method to initialise the JFrame
    createDisplay();
    space.setBackground(Color.pink);
```

```
    panel = new JPanel();
    panel.add(new JLabel(golfer.getInfo()));
    panel.add(new JLabel(golfer.getInfo1()));
    addToDisplay();
    panel.setBackground(Color.pink);
    buttonFunction2();
}
private JButton changeHandicap = new JButton("Change Handicap");
private JButton changeAge = new JButton("Change Age");
private JButton changeUsername = new JButton("Change Username");
private JButton changePassword = new JButton("Change Password");
private JButton logOut = new JButton("Log out");

public void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}
    private void addToDisplay() {

        //set the size of the JPanel (= to size of JFrame)
```

```
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    panel.add(new JLabel("Name: " + name));
    panel.add(new JLabel("Gender: " + gender));
    panel.add(new JLabel("Age: " + age));
    panel.add(new JLabel("Handicap: " + handicap));
    panel.add(space);
    panel.add(changeUsername);
    panel.add(changePassword);
    panel.add(changeAge);
    panel.add(changeHandicap);
    panel.add(logOut);
    frame.add(panel);
    frame.pack();
}

private void buttonFunction(){

    changeAge.addActionListener(e -> {
        new ChangeAge("Handicap Helper", 210, 350, golfer);
        frame.dispose();
    });

    changeHandicap.addActionListener(e -> {
        new ChangeHandicap("Handicap Helper", 210, 350, golfer);
```

```
            frame.dispose();
        });
        changeUsername.addActionListener(e -> {
            new ChangeUsername("Handicap Helper", 210, 350, golfer);
            frame.dispose();
        });
        changePassword.addActionListener(e -> {
            new ChangePassword("Handicap Helper", 210, 350, golfer);
            frame.dispose();
        });
        logOut.addActionListener(e -> {
            new Menu("Handicap Helper", 210, 350);
            frame.dispose();
        });
    }
private void buttonFunction1(){

    changeAge.addActionListener(e -> {
        new ChangeAge("Handicap Helper", 210, 350, golfer1);
        frame.dispose();
    });

    changeHandicap.addActionListener(e -> {
        new ChangeHandicap("Handicap Helper", 210, 350, golfer1);
        frame.dispose();
    });
    changeUsername.addActionListener(e -> {
        new ChangeUsername("Handicap Helper", 210, 350, golfer1);
        frame.dispose();
```

```
    });
    changePassword.addActionListener(e -> {
        new ChangePassword("Handicap Helper", 210, 350, golfer1);
        frame.dispose();
    });
    logOut.addActionListener(e -> {
        new Menu("Handicap Helper", 210, 350);
        frame.dispose();
    });
}
private void buttonFunction2(){

    changeAge.addActionListener(e -> {
        new ChangeAge("Handicap Helper", 210, 350, golfer2);
        frame.dispose();
    });

    changeHandicap.addActionListener(e -> {
        new ChangeHandicap("Handicap Helper", 210, 350, golfer2);
        frame.dispose();
    });
    changeUsername.addActionListener(e -> {
        new ChangeUsername("Handicap Helper", 210, 350, golfer2);
        frame.dispose();
    });
    changePassword.addActionListener(e -> {
        new ChangePassword("Handicap Helper", 210, 350, golfer2);
        frame.dispose();
    });
```

```
        logOut.addActionListener(e -> {
            new Menu("Handicap Helper", 210, 350);
            frame.dispose();
        });
    }
    }

import javax.swing.*;
import java.awt.*;

public class Information1{

    private JFrame frame;

    private String title;
    private int width;
    private int height;
    Golfer golfer;

    public Information1(String title, int width, int height){
        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer = golfer;

        //Calling the method to initialise the JFrame
        createDisplay();
        addToDisplay();
    }
```

```
//Label is just writing ("on the canvas")
private JLabel information = new JLabel("Information:");
private JLabel infoLn1 = new JLabel("This program is designed to use");
private JLabel infoLn2 = new JLabel("between family and friends so that") ;
private JLabel infoLn3 = new JLabel("you can compete fairly without the");
private JLabel infoLn4 = new JLabel("need to join a golf club. By signing");
private JLabel infoLn5 = new JLabel("up you will create an account that");
private JLabel infoLn6 = new JLabel("you can access at any time providing");
private JLabel infoLn7 = new JLabel("you with a handicap based on the");
private JLabel infoLn8 = new JLabel("CONGU handicap system.");
private JButton back = new JButton("back");
//Initialising the JFrame and the canvas in a new method

private void createDisplay() {

    frame = new JFrame(title);
    frame.setSize(width, height);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}

private void addToDisplay(){
    JPanel panel;
    panel = new JPanel();
```

```
        //set the size of the JPanel (= to size of JFrame)
        panel.setPreferredSize(new Dimension(width, height));
        //Making sure you can't change the size of the JPanel
        panel.setMaximumSize(new Dimension(width, height));
        panel.setMinimumSize(new Dimension(width, height));
        //Setting the layout for the JPanel
        panel.setLayout(new GridLayout(12, 5));
        panel.add(information);
        panel.add(new JPanel());
        panel.add(infoLn1);
        panel.add(infoLn2);
        panel.add(infoLn3);
        panel.add(infoLn4);
        panel.add(infoLn5);
        panel.add(infoLn6);
        panel.add(infoLn7);
        panel.add(infoLn8);
        panel.add(back);
        back.addActionListener(e -> {
            new Menu("Handicap Helper", 210, 350);
            frame.dispose();
        });
        //adding the canvas to the JFrame and making sure its visible
        frame.add(panel);
        frame.pack();
    }
}

import javax.swing.*;
```

```
import java.awt.*;

public class Information2{

    private JFrame frame;

    private String title;
    private int width;
    private int height;
    Golfer golfer;

    public Information2(String title, int width, int height, Golfer golfer){
        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer = golfer;

        //Calling the method to initialise the JFrame
        createDisplay();
        addToDisplay();
    }


    //Label is just writing ("on the canvas")
    private JLabel information = new JLabel("Information:");
    private JLabel infoLn1 = new JLabel("By filling in a scorecard you will");
    private JLabel infoLn2 = new JLabel("obtain a handicap based on the score") ;
    private JLabel infoLn3 = new JLabel("that you have entered. It is advised");
    private JLabel infoLn4 = new JLabel("that you fill in the scorecard");
```

```
private JLabel infoLn5 = new JLabel("based on an average round to get");
private JLabel infoLn6 = new JLabel("an accurate handicap. Once you");
private JLabel infoLn7 = new JLabel("have filled in the scorecard you");
private JLabel infoLn8 = new JLabel("can create an account to save it.");
private JButton back = new JButton("Back");
//Initialising the JFrame and the canvas in a new method

private void createDisplay() {

    frame = new JFrame(title);
    frame.setSize(width, height);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}

private void addToDisplay(){
    JPanel panel;
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    panel.add(information);
```

```
        panel.add(new JPanel());
        panel.add(infoLn1);
        panel.add(infoLn2);
        panel.add(infoLn3);
        panel.add(infoLn4);
        panel.add(infoLn5);
        panel.add(infoLn6);
        panel.add(infoLn7);
        panel.add(infoLn8);
        panel.add(back);
        back.addActionListener(e -> {
            new StartSignUp("Handicap Helper", 210, 350, golfer);
            frame.dispose();
        });
        //adding the canvas to the JFrame and making sure its visible
        frame.add(panel);
        frame.pack();
    }

}
import javax.swing.*;
import java.awt.*;
import java.io.File;
import java.io.FileReader;
import java.util.Scanner;



class Login {
```

```
private JFrame frame;
//Allows us to draw graphics to it

private String title;
private int width;
private int height;

Login(String title, int width, int height) {

    this.title = title;
    this.width = width;
    this.height = height;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction();

}

//Label is just writing ("on the canvas")
private JLabel login = new JLabel("Login");
private JTextField username = new JTextField();
private JPasswordField password = new JPasswordField();
private JButton okay = new JButton("Okay");
private JButton cancel = new JButton("Cancel");

//Initialising the JFrame and the canvas in a new method
```

```
private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}

private void addToDisplay() {
    JPanel panel;
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    //Adding the menu label to the canvas
    panel.add(login);
    panel.add(new JPanel());
    panel.add(new JLabel("Username:"));
    panel.add(username);
```

```
    panel.add(new JLabel("Password"));
    panel.add(password);
    panel.add(new JPanel());
    panel.add(okay);
    panel.add(cancel);
    frame.add(panel);
    frame.pack();
}

private void buttonFunction() {
    okay.addActionListener(e -> {
        String user = username.getText();
        String pass = String.valueOf(password.getPassword());

        File fileName = new File("users.txt");
        String line;
        boolean login = false;
        String username1 = "";
        String password1 = "";
        String name1 = "";
        int age1 = 0;
        double handicap1 = 0.0;
        String gender1 = "Male";

        try (Scanner scanner = new Scanner(new FileReader(fileName))) {
            while (scanner.hasNextLine() && !login) {
                line = scanner.nextLine();
                String[] lines;
                lines = line.split(", ", 6);
```

```
        username1 = lines[0];
        password1 = lines[1];
        name1 = lines[2];
        age1 = Integer.parseInt(lines[3]);
        handicap1 = Double.parseDouble(lines[4]);
        gender1 = lines[5];
        if (user.equals(username1) && pass.equals(password1)) {
            login = true;
        }
    }

} catch (Exception e1) {
    login = false;
}

String group;
if(age1 < 18 && gender1.equals("Male")){
    group = "boy";
}else if (age1 < 18 && gender1.equals("Female")){
    group = "girl";
}else if (age1 > 18 && gender1.equals("Female")){
    group = "female";
}else{
    group = "golfer";
}

if (login) {
    switch(group) {
        case "golfer":
```

```
                Golfer golfer1 = new Golfer(name1, age1, username1, password1, handicap1, gender1);
                new Homepage("Handicap Helper", 210, 350, golfer1);
                frame.dispose();
                break;

            case "boy":
                BoyGolfer golfer2 = new BoyGolfer(name1, age1, username1, password1, handicap1, gender1);
                new Homepage("Handicap Helper", 210, 350, golfer2);
                frame.dispose();
                break;

            case "girl":
                GirlGolfer golfer3 = new GirlGolfer(name1, age1, username1, password1, handicap1, gender1);
                new Homepage("Handicap Helper", 210, 350, golfer3);
                frame.dispose();
                break;

        }

    } else {
        JOptionPane.showMessageDialog(frame, "          Error!\nThere was a problem signing in\n make
sure your username and \n      password are correct.");
    }

});

cancel.addActionListener(e -> {
    new Menu("Handicap Helper", 210, 350);
    frame.dispose();
```

```
        });

        //adding the canvas to the JFrame and making sure its visible

    }

}
import javax.swing.*;
import java.awt.*;

public class Menu{

    private JFrame frame;

    private String title;
    private int width;
    private int height;
    Golfer golfer;

public Menu(String title, int width, int height){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer = golfer;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction();
```

```
}


//Label is just writing ("on the canvas")
private JLabel menu = new JLabel("Menu");
//Buttons will complete a task/action
private JButton login = new JButton("Login");
private JButton signUp = new JButton("Sign Up");
private JButton information = new JButton("Information");

   //Initialising the JFrame and the canvas in a new method

   private void createDisplay() {

      frame = new JFrame(title);
      frame.setSize(width, height);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      //Setting the location of where the window appears on the screen
      frame.setLocationRelativeTo(null);
      //Making sure the window is actually visible
      frame.setVisible(true);
   }

   private void addToDisplay(){
      JPanel panel;
      panel = new JPanel();
      //set the size of the JPanel (= to size of JFrame)
      panel.setPreferredSize(new Dimension(width, height));
      //Making sure you can't change the size of the JPanel
```

```
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    panel.add(menu);
    panel.add(new JPanel());
    panel.add(login);
    panel.add(new JPanel());
    panel.add(signUp);
    panel.add(new JPanel());
    panel.add(information);
    //adding the canvas to the JFrame and making sure its visible
    frame.add(panel);
    frame.pack();
}

private void buttonFunction(){
    //action listener for when a button is pressed
    login.addActionListener(e -> {
    new Login("Handicap Helper", 210, 350);
    frame.dispose();
});

    signUp.addActionListener(e -> {
        new Gender("Handicap Helper", 210, 350);
        frame.dispose();
    });

    information.addActionListener(e -> {
```

```
                new Information1("Handicap Helper", 210, 350);
                frame.dispose();
            });
        }
    }
import javax.swing.*;
import java.awt.*;



class Name{

    private JFrame frame;
    //Allows us to draw graphics to it

    private String title;
    private int width;
    private int height;
    private Golfer golfer;

    Name(String title, int width, int height, Golfer golfer){

        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer = golfer;

        //Calling the method to initialise the JFrame
        createDisplay();
        addToDisplay();
```

```
    buttonFunction();


}


//Label is just writing ("on the canvas")
private JLabel login = new JLabel("Personal Details");
private JTextField name = new JTextField();
private JTextField age = new JTextField();
private JButton okay = new JButton("Okay");
private JButton cancel = new JButton("Cancel");




//Initialising the JFrame and the canvas in a new method
private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}
private void addToDisplay(){
```

```
    JPanel panel;
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    //Adding the menu label to the canvas
    panel.add(login);
    panel.add(new JPanel());
    panel.add(new JLabel("What is your name:"));
    panel.add(name);
    panel.add(new JLabel("How old are you"));
    panel.add(age);
    panel.add(new JPanel());
    panel.add(okay);
    panel.add(cancel);
    //adding the canvas to the JFrame and making sure its visible
    frame.add(panel);
    frame.pack();
}

private void buttonFunction(){
    okay.addActionListener(e -> {

        try {
            boolean error = false;
```

```
            char[] chars = name.getText().toCharArray();
            for(int x = 0 ; x < chars.length; x ++){
               if((int)chars[x] == 32 || (int)chars[x] < 64 || (int)chars[x] > 123){
                  error = true;
               }
            }
            if(!error) {
               String name1 = name.getText();
               int age1 = Integer.parseInt(age.getText());
               golfer.setAge(age1);
               golfer.setName(name1);
               new StartSignUp("Handicap Helper", 210, 350, golfer);
               frame.dispose();
            }else if(error){
               JOptionPane.showMessageDialog(frame, "Error. Name must include letters only");
            }

         }catch(Exception e1){
            JOptionPane.showMessageDialog(frame, "Error. You must enter an number for your age");
         }

      });
      cancel.addActionListener(e -> {
         new Menu("Handicap Helper", 210, 350);
         frame.dispose();
      });
   }
}
ort javax.swing.*;
```

```
import java.awt.*;
import java.util.stream.IntStream;



class Scorecard1 extends JFrame{


    private int totalPar;
    private int totalScore;
    private int totalAdScore;
    //Creating a window (so that we can see the game)
    private JFrame frame;
    //Allows us to draw graphics to it
    private JPanel panel;
    //Label is just writing ("on the canvas")
    private JLabel hole = new JLabel ("Hole");
    private JLabel par = new JLabel("Par");
    private JLabel score = new JLabel("Score");
    private JLabel hole1 = new JLabel("1");
    private JLabel hole2 = new JLabel("2");
    private JLabel hole3 = new JLabel("3");
    private JLabel hole4 = new JLabel("4");
    private JLabel hole5 = new JLabel("5");
    private JLabel hole6 = new JLabel("6");
    private JLabel hole7 = new JLabel("7");
    private JLabel hole8 = new JLabel("8");
    private JLabel hole9 = new JLabel("9");
    private String[] parArray = {"4" , "3", "5"};
    private JTextField[] scores = new JTextField[9];
    private JComboBox[] pars = new JComboBox[9];
```

```
private int[] intScores = new int[9];
private int[] intPars = new int[9];
//Buttons will complete a task/action
private JButton cont = new JButton("Continue");
private JButton cancel = new JButton("Cancel");

private String title;
private int width;
private int height;
Golfer golfer;

Scorecard1(String title, int width, int height, Golfer golfer){
    this.title = title;
    this.width = width;
    this.height = height;
    this.golfer = golfer;

    //Calling the method to initialise the JFrame
    createDisplay();
    addToDisplay();
    buttonFunction();
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    scores[0].setHorizontalAlignment(SwingConstants.CENTER);
    scores[0].setPreferredSize(new Dimension(25,25));
    scores[1].setHorizontalAlignment(SwingConstants.CENTER);
    scores[1].setPreferredSize(new Dimension(25,25));
    scores[2].setHorizontalAlignment(SwingConstants.CENTER);
    scores[2].setPreferredSize(new Dimension(25,25));
    scores[3].setHorizontalAlignment(SwingConstants.CENTER);
```

```
scores[3].setPreferredSize(new Dimension(25,25));
scores[4].setHorizontalAlignment(SwingConstants.CENTER);
scores[4].setPreferredSize(new Dimension(25,25));
scores[5].setHorizontalAlignment(SwingConstants.CENTER);
scores[5].setPreferredSize(new Dimension(25,25));
scores[6].setHorizontalAlignment(SwingConstants.CENTER);
scores[6].setPreferredSize(new Dimension(25,25));
scores[7].setHorizontalAlignment(SwingConstants.CENTER);
scores[7].setPreferredSize(new Dimension(25,25));
scores[8].setHorizontalAlignment(SwingConstants.CENTER);
scores[8].setPreferredSize(new Dimension(25,25));

pars[0].setPreferredSize(new Dimension(50,25));

pars[1].setPreferredSize(new Dimension(50,25));

pars[2].setPreferredSize(new Dimension(50,25));

pars[3].setPreferredSize(new Dimension(50,25));

pars[4].setPreferredSize(new Dimension(50,25));

pars[5].setPreferredSize(new Dimension(50,25));

pars[6].setPreferredSize(new Dimension(50,25));

pars[7].setPreferredSize(new Dimension(50,25));

pars[8].setPreferredSize(new Dimension(50,25));
```

```
}

//Initialising the JFrame and the canvas in a new method
private void createDisplay() {
    JScrollPane scrollPane = new JScrollPane(panel);
    scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}
private void addToDisplay(){
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new FlowLayout(FlowLayout.LEADING));
```

```
for (int y = 0; y < 9; y++) {
    scores[y] = new JTextField();
    pars[y] = new JComboBox(parArray);
}
//Adding the menu label to the canvas
panel.add(new JLabel("Front 9"));
panel.add(new JLabel("                                    "));
panel.add(hole);
panel.add(new JLabel("     "));
panel.add(par);
panel.add(new JLabel("           "));
panel.add(score);
panel.add(new JLabel("           "));
panel.add(hole1);
panel.add(new JLabel("          "));
panel.add(pars[0]);
panel.add(new JLabel("     "));
panel.add(scores[0]);
panel.add(new JLabel("           "));
panel.add(hole2);
panel.add(new JLabel("          "));
panel.add(pars[1]);
panel.add(new JLabel("     "));
panel.add(scores[1]);
panel.add(new JLabel("             "));
panel.add(hole3);
panel.add(new JLabel("          "));
panel.add(pars[2]);
panel.add(new JLabel("     "));
```

```
panel.add(scores[2]);
panel.add(new JLabel("              "));
panel.add(hole4);
panel.add(new JLabel("          "));
panel.add(pars[3]);
panel.add(new JLabel("     "));
panel.add(scores[3]);
panel.add(new JLabel("              "));
panel.add(hole5);
panel.add(new JLabel("          "));
panel.add(pars[4]);
panel.add(new JLabel("     "));
panel.add(scores[4]);
panel.add(new JLabel("               "));
panel.add(hole6);
panel.add(new JLabel("          "));
panel.add(pars[5]);
panel.add(new JLabel("     "));
panel.add(scores[5]);
panel.add(new JLabel("               "));
panel.add(hole7);
panel.add(new JLabel("          "));
panel.add(pars[6]);
panel.add(new JLabel("     "));
panel.add(scores[6]);
panel.add(new JLabel("             "));
panel.add(hole8);
panel.add(new JLabel("          "));
panel.add(pars[7]);
```

```
    panel.add(new JLabel("     "));
    panel.add(scores[7]);
    panel.add(new JLabel("              "));
    panel.add(hole9);
    panel.add(new JLabel("           "));
    panel.add(pars[8]);
    panel.add(new JLabel("     "));
    panel.add(scores[8]);
    panel.add(new JLabel("               "));
    panel.add(cont);
    panel.add(cancel);
    //adding the canvas to the JFrame and making sure its visible
    frame.add(panel);
    frame.pack();

}

private void buttonFunction(){
    cont.addActionListener(e ->  {
        boolean error = false;
        for(int i = 0; i < 9; i++){
            try {
                intScores[i] = Integer.parseInt(scores[i].getText());
                if(intScores[i] > 20){
                    error = true;
                }
            }catch(NumberFormatException e1){
                error = true;
                break;
```

```
    }
    try {
        intPars[i] = Integer.parseInt((String)pars[i].getSelectedItem());
    }catch(NumberFormatException e1){
        error = true;
        break;
    }
}
for (int x = 0; x < 9; x++) {
    if (scores[x].getText().isEmpty()) {
        error = true;
    }
}
    if (!error) {

        totalScore = IntStream.of(intScores).sum();
        for (int x = 0; x < 9; x++) {
            if ((intScores[x] - intPars[x]) > 1) {
                intScores[x] = intPars[x] + 2;
                scores[x].setText(Integer.toString(intScores[x]));
            }
        }
        totalAdScore = IntStream.of(intScores).sum();
        totalPar = IntStream.of(intPars).sum();
        new Confirm("Handicap Helper", 210, 350, totalAdScore, totalPar, totalScore, golfer);
        frame.dispose();
    } else {
```

```
                JOptionPane.showMessageDialog(frame, "Make sure scores boxes are filled with numbers less than
20");
                frame.dispose();
                new Scorecard1("Handicap Helper", 210, 350, golfer);
            }



        });
        cancel.addActionListener(e -> {
            new Menu("Handicap Helper", 210, 350);
            frame.dispose();
        });
    }
}
import javax.swing.*;
import java.awt.*;
import java.util.stream.IntStream;



class Scorecard2 extends JFrame {

    private int totalPar;
    private int totalScore;
    private int totalAdScore;
    //Creating a window (so that we can see the game)
    private JFrame frame;
    //Allows us to draw graphics to it
    private JPanel panel;
    //Label is just writing ("on the canvas")
```

```
private JLabel hole = new JLabel("Hole");
private JLabel par = new JLabel("Par");
private JLabel score = new JLabel("Score");
private JLabel hole1 = new JLabel("10");
private JLabel hole2 = new JLabel("11");
private JLabel hole3 = new JLabel("12");
private JLabel hole4 = new JLabel("13");
private JLabel hole5 = new JLabel("14");
private JLabel hole6 = new JLabel("15");
private JLabel hole7 = new JLabel("16");
private JLabel hole8 = new JLabel("17");
private JLabel hole9 = new JLabel("18");
private String[] parArray = {"4", "5", "3"};
private JTextField[] scores = new JTextField[9];
private JComboBox[] pars = new JComboBox[9];
private int[] intScores = new int[9];
private int[] intPars = new int[9];
//Buttons will complete a task/action
private JButton cont = new JButton("Continue");
private JButton cancel = new JButton("Cancel");

private String title;
private int width;
private int height;
private Golfer golfer;

Scorecard2(String title, int width, int height, Golfer golfer) {
    this.title = title;
    this.width = width;
```

```
this.height = height;
this.golfer = golfer;

//Calling the method to initialise the JFrame
createDisplay();
addToDisplay();
buttonFunction();
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
scores[0].setHorizontalAlignment(SwingConstants.CENTER);
scores[0].setPreferredSize(new Dimension(25, 25));
scores[1].setHorizontalAlignment(SwingConstants.CENTER);
scores[1].setPreferredSize(new Dimension(25, 25));
scores[2].setHorizontalAlignment(SwingConstants.CENTER);
scores[2].setPreferredSize(new Dimension(25, 25));
scores[3].setHorizontalAlignment(SwingConstants.CENTER);
scores[3].setPreferredSize(new Dimension(25, 25));
scores[4].setHorizontalAlignment(SwingConstants.CENTER);
scores[4].setPreferredSize(new Dimension(25, 25));
scores[5].setHorizontalAlignment(SwingConstants.CENTER);
scores[5].setPreferredSize(new Dimension(25, 25));
scores[6].setHorizontalAlignment(SwingConstants.CENTER);
scores[6].setPreferredSize(new Dimension(25, 25));
scores[7].setHorizontalAlignment(SwingConstants.CENTER);
scores[7].setPreferredSize(new Dimension(25, 25));
scores[8].setHorizontalAlignment(SwingConstants.CENTER);
scores[8].setPreferredSize(new Dimension(25, 25));
pars[0].setPreferredSize(new Dimension(50, 25));

pars[1].setPreferredSize(new Dimension(50, 25));
```

```
        pars[2].setPreferredSize(new Dimension(50, 25));

        pars[3].setPreferredSize(new Dimension(50, 25));

        pars[4].setPreferredSize(new Dimension(50, 25));

        pars[5].setPreferredSize(new Dimension(50, 25));

        pars[6].setPreferredSize(new Dimension(50, 25));

        pars[7].setPreferredSize(new Dimension(50, 25));

        pars[8].setPreferredSize(new Dimension(50, 25));

    }

    //Initialising the JFrame and the canvas in a new method
    private void createDisplay() {
        JScrollPane scrollPane = new JScrollPane(panel);
        scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);
        frame = new JFrame(title);
        frame.setSize(width, height);
        //very important line of code to make sure the window closes down properly
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //OPTIONAL EXTRAS
        //Making sure you cant resize the frame
        frame.setResizable(false);
        //Setting the location of where the window appears on the screen
```

```
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}

private void addToDisplay() {
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new FlowLayout(FlowLayout.LEADING));

    for (int y = 0; y < 9; y++) {
        scores[y] = new JTextField();
        pars[y] = new JComboBox(parArray);
    }
    //Adding the menu label to the canvas
    panel.add(new JLabel("Back 9"));
    panel.add(new JLabel("                                    "));
    panel.add(hole);
    panel.add(new JLabel("     "));
    panel.add(par);
    panel.add(new JLabel("           "));
    panel.add(score);
    panel.add(new JLabel("           "));
    panel.add(hole1);
```

```
panel.add(new JLabel("         "));
panel.add(pars[0]);
panel.add(new JLabel("     "));
panel.add(scores[0]);
panel.add(new JLabel("            "));
panel.add(hole2);
panel.add(new JLabel("         "));
panel.add(pars[1]);
panel.add(new JLabel("     "));
panel.add(scores[1]);
panel.add(new JLabel("              "));
panel.add(hole3);
panel.add(new JLabel("         "));
panel.add(pars[2]);
panel.add(new JLabel("     "));
panel.add(scores[2]);
panel.add(new JLabel("              "));
panel.add(hole4);
panel.add(new JLabel("         "));
panel.add(pars[3]);
panel.add(new JLabel("     "));
panel.add(scores[3]);
panel.add(new JLabel("              "));
panel.add(hole5);
panel.add(new JLabel("         "));
panel.add(pars[4]);
panel.add(new JLabel("     "));
panel.add(scores[4]);
panel.add(new JLabel("           "));
```

```
panel.add(hole6);
panel.add(new JLabel("         "));
panel.add(pars[5]);
panel.add(new JLabel("     "));
panel.add(scores[5]);
panel.add(new JLabel("          "));
panel.add(hole7);
panel.add(new JLabel("         "));
panel.add(pars[6]);
panel.add(new JLabel("     "));
panel.add(scores[6]);
panel.add(new JLabel("              "));
panel.add(hole8);
panel.add(new JLabel("         "));
panel.add(pars[7]);
panel.add(new JLabel("     "));
panel.add(scores[7]);
panel.add(new JLabel("              "));
panel.add(hole9);
panel.add(new JLabel("         "));
panel.add(pars[8]);
panel.add(new JLabel("     "));
panel.add(scores[8]);
panel.add(new JLabel("         "));
panel.add(cont);
panel.add(cancel);
//adding the canvas to the JFrame and making sure its visible
frame.add(panel);
frame.pack();
```

```
}

private void buttonFunction() {
    cancel.addActionListener(e -> {
        new Menu("Handicap Helper", 210, 350);
        frame.dispose();
    });
    cont.addActionListener(e -> {
        boolean error = false;
        for (int i = 0; i < 9; i++) {
            try {
                intScores[i] = Integer.parseInt(scores[i].getText());
                if(intScores[i] > 20){
                    error = true;
                }
            } catch (NumberFormatException e1) {
                error = true;
                break;
            }
            try {
                intPars[i] = Integer.parseInt((String) pars[i].getSelectedItem());
            } catch (NumberFormatException e1) {
                error = true;
                break;
            }
        }

            for (int x = 0; x < 9; x++) {
                if (scores[x].getText().isEmpty()) {
```

```
                error = true;
            }
        }
        if (!error) {

            totalScore = IntStream.of(intScores).sum();
            for (int x = 0; x < 9; x++) {
                if ((intScores[x] - intPars[x]) > 1) {
                    intScores[x] = intPars[x] + 2;
                    scores[x].setText(Integer.toString(intScores[x]));
                }
            }
            totalAdScore = IntStream.of(intScores).sum();
            totalPar = IntStream.of(intPars).sum();
            new Confirm2("Handicap Helper", 210, 350, totalAdScore, totalPar, totalScore, golfer);
            frame.dispose();
        }else {
            JOptionPane.showMessageDialog(frame, "Make sure scores boxes are filled with numbers less than
20");
            frame.dispose();
            new Scorecard1("Handicap Helper", 210, 350, golfer);

        }

    });
  }


}
```

```
import javax.swing.*;
import java.awt.*;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;

class SignUp {

    private JFrame frame;
    //Allows us to draw graphics to it



    private String title;
    private int width;
    private int height;
    private Golfer golfer;
    private String fileName = "users.txt";

    SignUp(String title, int width, int height, Golfer golfer) {

        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer = golfer;

        //Calling the method to initialise the JFrame
```

```
    createDisplay();
    addToDisplay();
    buttonFunction();
}

//Creating a window (so that we can see the game)
//Label is just writing ("on the canvas")
private JLabel signUp = new JLabel("Sign up");
//Buttons will complete a task/action
private JTextField username = new JTextField();
private JPasswordField password = new JPasswordField();
private JButton okay = new JButton("Okay");
private JButton cancel = new JButton("Cancel");
private String newUser;
private String newPass;

//Initialising the JFrame and the canvas in a new method

private void createDisplay() {
    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
```

```
    frame.setVisible(true);
}

private void buttonFunction() {
    okay.addActionListener(e -> {
        boolean userTaken = isTaken();
        actionPerformed(userTaken);
    });
}

private void actionPerformed(boolean userTaken) {
    boolean userHasNoComma = true;
    boolean passHasNoComma = true;
    boolean isLongEnough = true;

    char[] chars = newUser.toCharArray();
    for(int x = 0 ; x < chars.length; x ++){
        if((int)chars[x] == 32 || (int)chars[x] == 44){
            userHasNoComma = false;
        }
    }

    char[] chars2 = newPass.toCharArray();
    for(int x = 0 ; x < chars2.length; x ++){
        if((int)chars2[x] == 32 || (int)chars2[x] == 44){
            passHasNoComma = false;
        }
    }
    if (newUser.length() < 5 || newPass.length() < 5){
```

```
            isLongEnough = false;
        }
        if (userTaken) {
            JOptionPane.showMessageDialog(frame, "This username has already been taken \n            Please try
another");
            new SignUp("Handicap Helper", 210, 350, golfer);
            frame.dispose();
        }else if(!userHasNoComma){
            JOptionPane.showMessageDialog(frame,"You can not have spaces or commas in your username");
            new SignUp("Handicap Helper", 210, 350, golfer);
            frame.dispose();
        }else if(!passHasNoComma){
            JOptionPane.showMessageDialog(frame,"You can not have spaces or commas in your password");
            new SignUp("Handicap Helper", 210, 350, golfer);
            frame.dispose();
        }else if(!isLongEnough){
            JOptionPane.showMessageDialog(frame,"Your username and password must be longer than 4
characters");
            new SignUp("Handicap Helper", 210, 350, golfer);
            frame.dispose();
        }else{
            try {
                fileWriter();
            } catch (Exception e1) {
                JOptionPane.showMessageDialog(frame, "Error. Try Again");
            }
        new Login("Handicap Helper", 210, 350);
        frame.dispose();
    }
```

```
}
    private boolean isTaken(){
        newUser = username.getText();
        newPass = String.valueOf(password.getPassword());
        try (Scanner scanner = new Scanner(new FileReader(fileName))) {
            String line;
            while (scanner.hasNextLine()) {
                line = scanner.nextLine();
                String[] lines;
                lines = line.split(", ", 5);
                if (lines[0].equals(newUser)) {
                    return true;
                }
            }
        } catch (Exception e1) {
            JOptionPane.showMessageDialog(frame, "Trouble checking new username");
            new SignUp("Handicap Helper", 210, 350, golfer);
        }
        return false;
    }

    private void addToDisplay(){
        JPanel panel;
        panel = new JPanel();
        //set the size of the JPanel (= to size of JFrame)
        panel.setPreferredSize(new Dimension(width, height));
        //Making sure you can't change the size of the JPanel
        panel.setMaximumSize(new Dimension(width, height));
```

```
        panel.setMinimumSize(new Dimension(width, height));
        //Setting the layout for the JPanel
        panel.setLayout(new GridLayout(12, 5));
        //Adding the menu label to the canvas
        panel.add(signUp);
        panel.add(new JLabel("Username you would like:"));
        panel.add(username);
        panel.add(new JLabel("Password you would like:"));
        panel.add(password);
        panel.add(new JPanel());
        panel.add(okay);
        panel.add(cancel);
        cancel.addActionListener(e -> {
            new Menu("Handicap Helper", 210, 350);
            frame.dispose();
        });
        //adding the canvas to the JFrame and making sure its visible
        frame.add(panel);
        frame.pack();
    }
    private void fileWriter() throws IOException {
        FileWriter fileWriter = new FileWriter(fileName, true);
        PrintWriter printWriter = new PrintWriter(fileWriter);
        printWriter.println(newUser + ", " + newPass + ", " + golfer.getName() + ", " + golfer.getAge() + ", " +
golfer.getHandicap() + ", " + golfer.getGender());
        printWriter.close();
    }
}
import javax.swing.*;
```

```
import java.awt.*;

 class StartSignUp {

    private JFrame frame;
    //Allows us to draw graphics to it



    private String title;
    private int width;
    private int height;
    private Golfer golfer;

    StartSignUp(String title, int width, int height, Golfer golfer) {

        this.title = title;
        this.width = width;
        this.height = height;
        this.golfer = golfer;

        //Calling the method to initialise the JFrame
        createDisplay();
        addToDisplay();
        buttonFunction();
    }

     private JButton okay = new JButton("Okay");
     private JButton cancel = new JButton("Cancel");
     private JButton info = new JButton("Information");
```

```
//Initialising the JFrame and the canvas in a new method

private void createDisplay() {

    frame = new JFrame(title);
    frame.setSize(width, height);
    //very important line of code to make sure the window closes down properly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //OPTIONAL EXTRAS
    //Making sure you cant resize the frame
    frame.setResizable(false);
    //Setting the location of where the window appears on the screen
    frame.setLocationRelativeTo(null);
    //Making sure the window is actually visible
    frame.setVisible(true);
}

private void addToDisplay(){
    JPanel panel;
    panel = new JPanel();
    //set the size of the JPanel (= to size of JFrame)
    panel.setPreferredSize(new Dimension(width, height));
    //Making sure you can't change the size of the JPanel
    panel.setMaximumSize(new Dimension(width, height));
    panel.setMinimumSize(new Dimension(width, height));
    //Setting the layout for the JPanel
    panel.setLayout(new GridLayout(12, 5));
    //Adding the menu label to the canvas
    panel.add(new JLabel("To sign up you must first fill in a "));
```

```
        panel.add(new JLabel("scorecard to obtain a handicap"));
        panel.add(new JPanel());
        panel.add(new JPanel());
        panel.add(new JLabel("Click okay to continue"));
        panel.add(okay);
        panel.add(cancel);
        panel.add(info);
        frame.add(panel);
        frame.pack();
    }

    private void buttonFunction(){
        okay.addActionListener(e -> {
        frame.dispose();
        new Scorecard1("Handicap Helper", 210, 350, golfer);
        });
        cancel.addActionListener(e -> {
            frame.dispose();
            new Menu("Handicap Helper", 210, 350);
        });
        info.addActionListener(e -> {
            frame.dispose();
            new Information2("Handicap Helper", 210, 350, golfer);
        });
    }
}
```