Introduction

This will be a Gym Membership application with JavaFX (GUI application) where staff can manage the gym member. The staff will be allowed to add, delete, edit, search, perform payments and all sort of input validation will be included throughout the application, there will also be created a receipt every time a transaction is successful.

This application will use text files as database where the member’s details will be stored in, added, edited or delete from.

Some CSS may be expected as well in an extern file in a way to make the window more attractive and better looking.

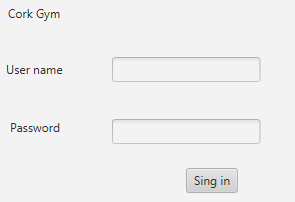
Main challenge will be to apply new features of java (JavaFX) recently learn this semester.

Ruben Rodrigues

R00144165

**Scenario 1**

As a fundamental part of any system there will be some security in this case will be a common login page where a user name and password will be needed to input and validation so then they can assess the information of the gym members.



**Scenario 2**

This GUI primary interface will be a table where the user can add, delete, edit and view all the gym members.

Table

* Any number of members can be added.
* Can be sorted by ID, Name, Address, Membership, Balance.
* Vertical and Horizontal bars are automatically created if the table is too long for the interface.

Add Member - Takes the user to another interface where a member can be added after the user input information is accepted with no validation errors

Delete Member - This button deletes a member after choosing a row and shows the user a confirmation alert message.

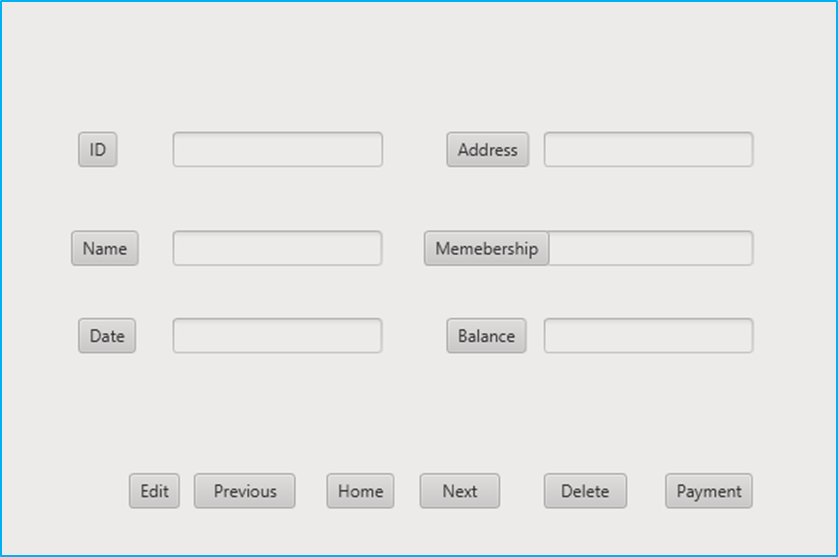
Search Member – This button will move the user into another interface where a prompt message comes up to enter the ID of the desired member to search.

Log out – This button will take you back to the log in interface.

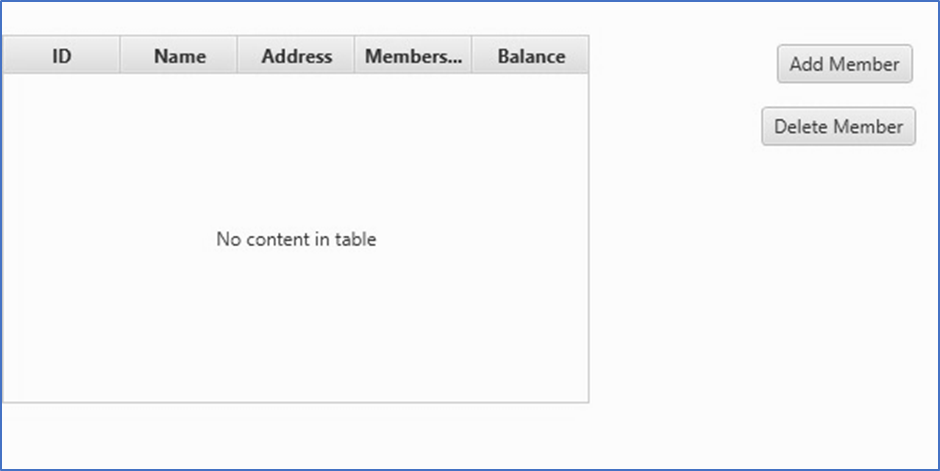
****

**Scenario 3**

There are also different ways to delete a member such as searching by ID where it would lead you to an individual view of the member with the option if button Delete.

****

Also, it could also be delete by simple selecting the member from the table and delete using the button Delete form this interface.

****

**Scenario 4**

This is the Add Member interface where the user can add gym members.

ID – Auto-Increment no need for input form user also in case there is a gap between number when deleting members when adding a new one goes to that id filling the gap.

Name – Input from user

Address- Input from user

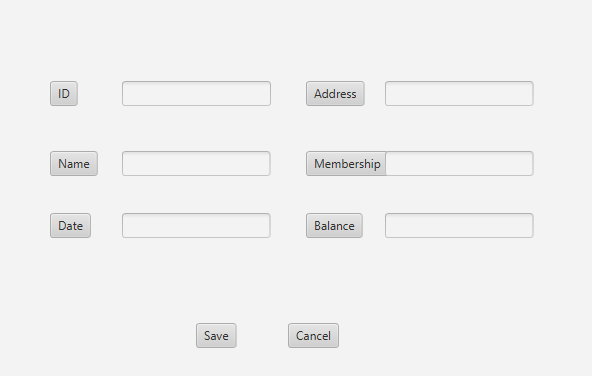
Date – Automatically get the date of when the member is added

Membership – User choose from a drop box different types of memberships

Balance – Automatically changes depending on the type of membership

Save Button – Added the information to the database and saves it also performs different sorts of validation from the user input and confirming with an alert message.

Cancel Button– Returns to the main interface of this GUI not saving any data



**Scenario 5**

This is the search interface where a prompt message come up and a text box to insert the id number which is the attribute used to search for members.

After will show the same view as before an individual view of the member searched with some additional functionalities such as:

Edit Button – Allows changes in the member details.

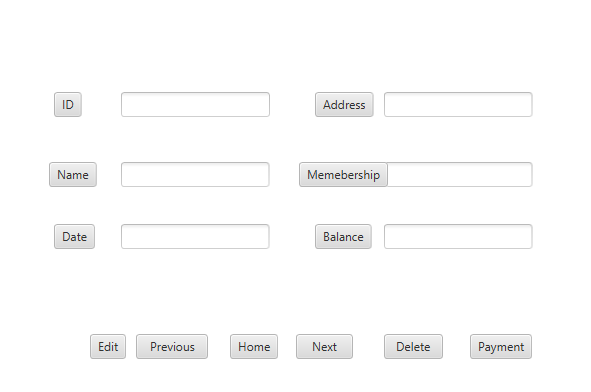
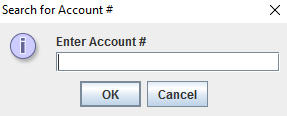
Previous Button – Goes to the previous ID member.

Next Button – Goes to the next Id member

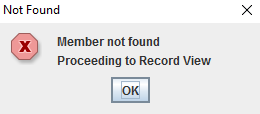
Delete Button – Deletes the member permanently.

Payment Button – Goes to another interface for where member card details will be need for input and validation.

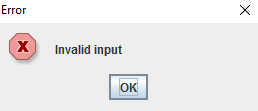
Home Button – Goes back to the list of member’s interfaces.



If and inexistent ID is entered an error message of the following type will show.

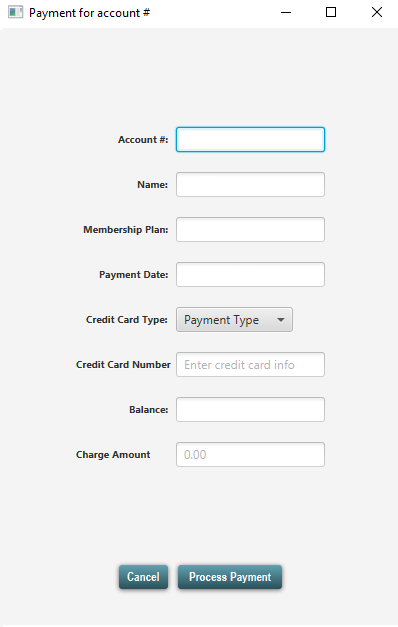


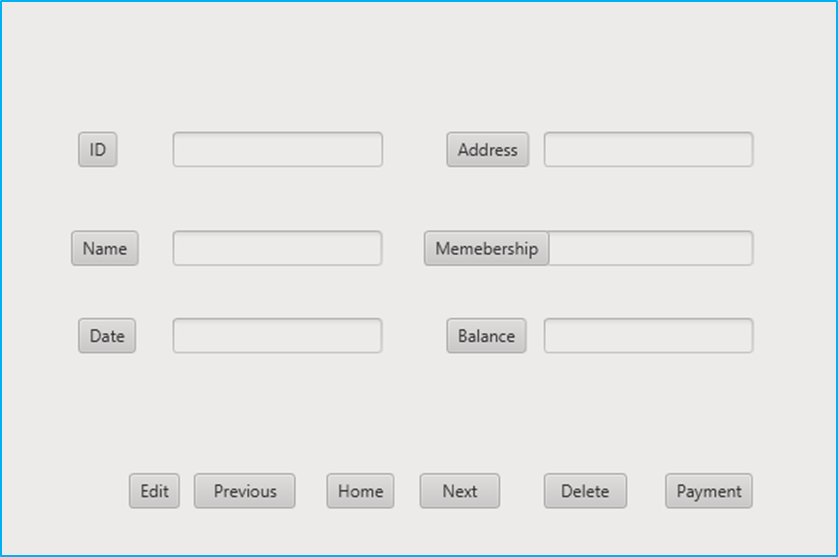
If and invalid input (e.g. a name) is entered an error message of the following type will show.



**Scenario 6**

This is the payment interface where the user will be needed to insert details about the member’s card for payment.





**Scenario 7**

This is the edit interface which is very similar to the Add Member interface, actually the only difference is that this one will have already values in the field to be edited.

