Even Semester (2023)

BINUS UNIVERSITY

BINUS INTERNATIONAL

Object-Oriented Programming Final Project

(Individual Work)

Student Information:

Surname: Pandiora **Given Name:** Felise Amore **Student ID:** 2602174453

Course Code **:** COMP6699001 **Course Name** **:** Object-Oriented Programming

Class **:** L2BC **Lecturer** **:** Jude Joseph Lamug Martinez, MCS

Type of Assignment **:** Final Project Report

Submission Pattern:

Due Date **:** 16 June 2023 **Submission Date** **:** 16 June 2023

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/ Cheating

BINUS International seriously regards all forms of plagiarism, cheating, and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity, and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept, and consent to BINUS International's terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student **:** Felise Amore Pandiora

Table Of Contents

DESCRIPTION

1. Introduction

This program is a simple gym management system built using Java and Java Swing, a GUI to make a better user interface. When the program is run, the user will be brought to the login page, and since the program is from the admin point of view, the user will have to log in as an admin. Then they will be brought to the main page with six interactive menus for the users to add new members, update and delete members, view the list of members, check or verify the payment, log out, and exit the program.

This program also made use of the MySQL database to store the members' data. There are two tables used for the database system. One is for the member's personal information and the other one is for the member's payment record. Here is the usage of the database for each page:

New member page	To store all new members' information
Update and delete page	To update the previous data in the database and delete members' data from the database.
List of members page	To display all current members
Payment page	To track members' payment record

2. Purpose of Program

This program is made to complete the Object-oriented Programming final project. For this final project, students are expected to make a program using the knowledge and skills they have learned in Java over the semester. One of the must-include topics is inheritance, interface, polymorphism, and either database or file system. Of course, students are not limited to only using the topics taught in the syllabus, on the other hand, we are encouraged to use topics outside of it.

The main purpose of this gym management system is to help gym owners or admins to simplify the process of administration and operation of the gym center. It enables the admin to efficiently do registration and payments.

DESIGN

1. Activity Diagram
   - Login Page

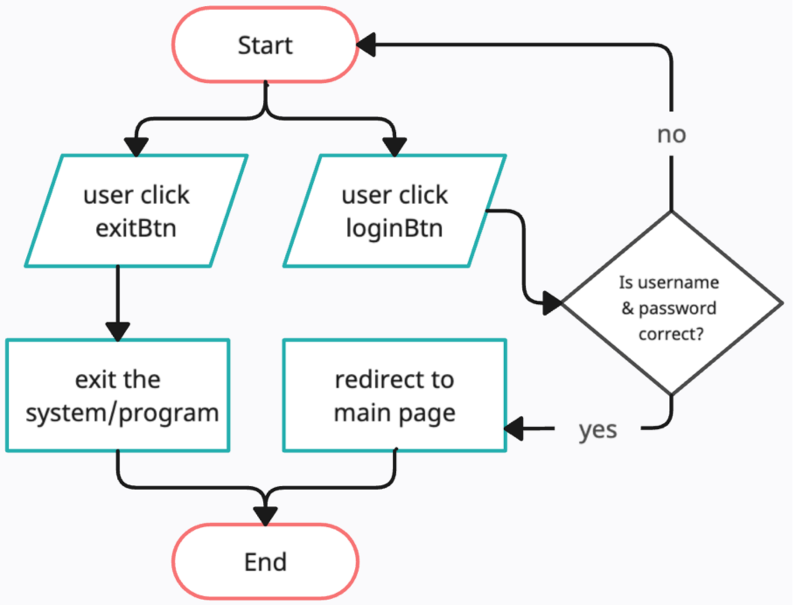


Fig 1.1 Activity Diagram for the login page

   - Main Page

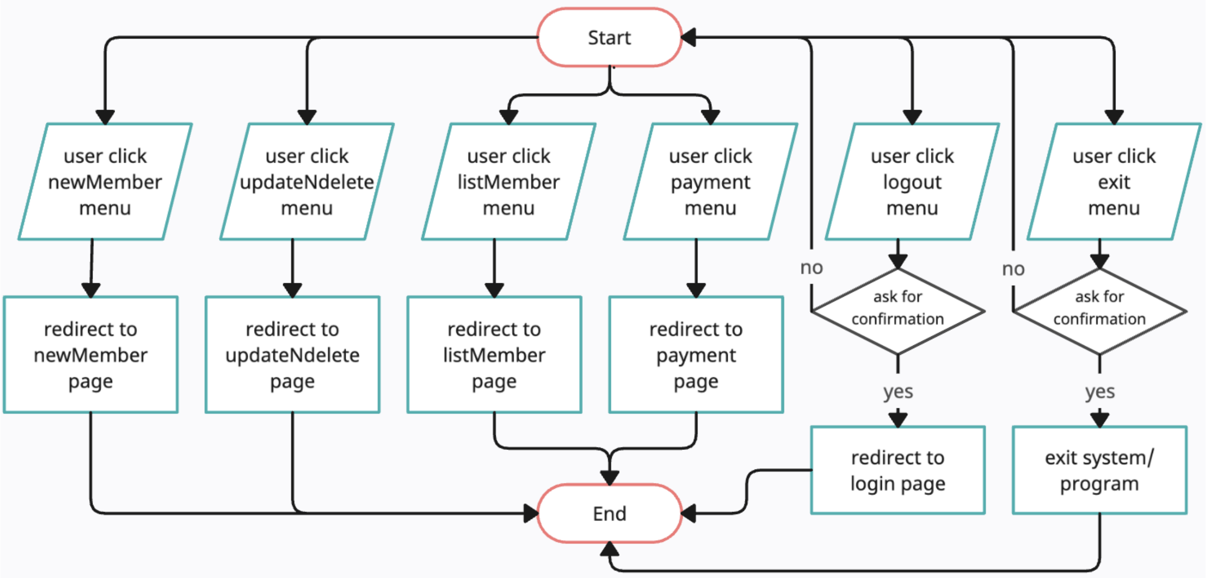


fig 1.2 Activity Diagram for Reset Button in Main Page

- Add New Member Page

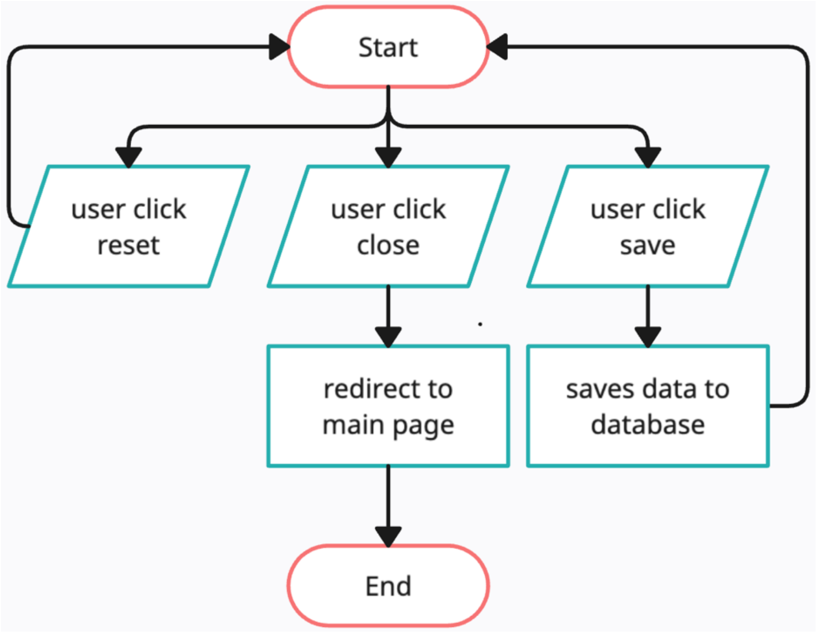


fig 1.3 Activity Diagram forvAdd New Member Page

- Update and Delete Member Page

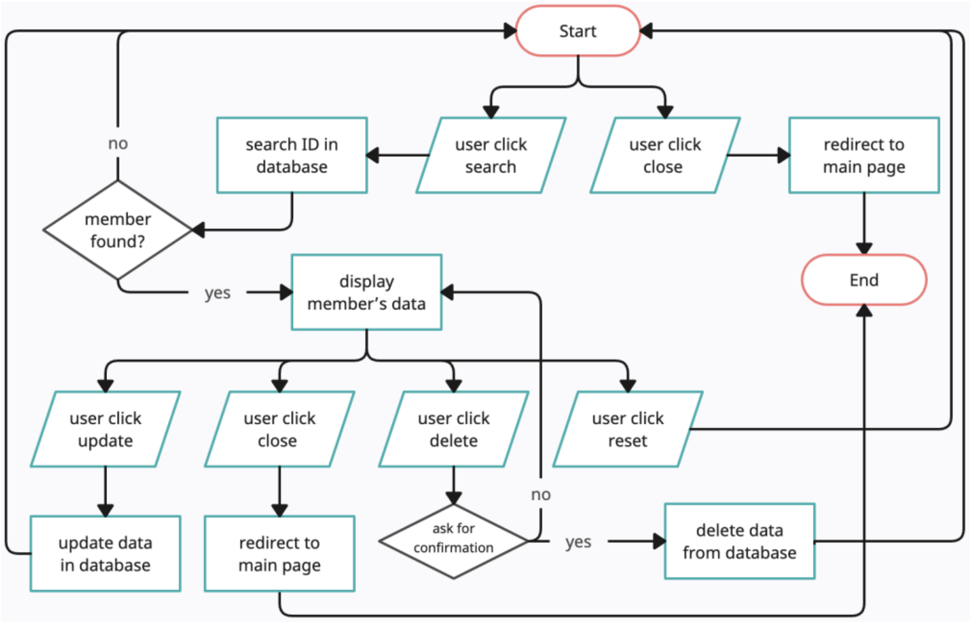


fig 1.4 Activity Diagram for Update and Delete Member Page

- List of Members Page

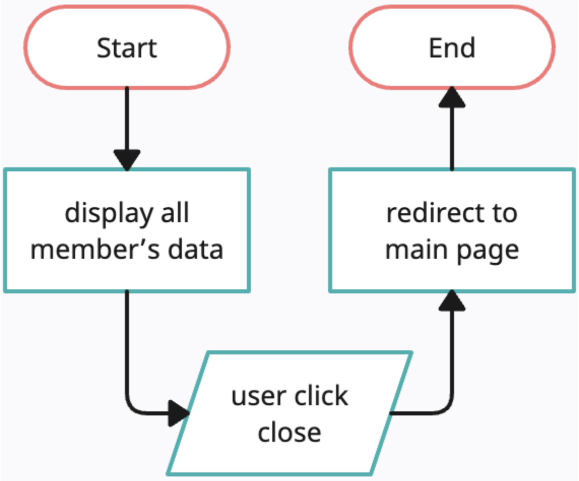


fig 1.5 Activity Diagram for List of Members Page

- Payment Page

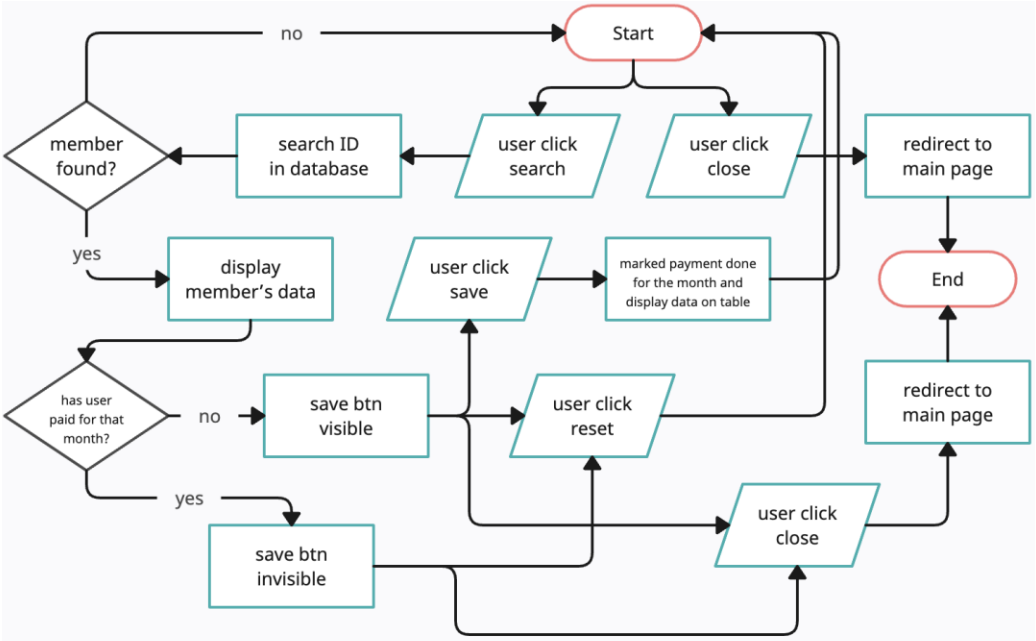


fig 1.6 Activity Diagram for Payment Page

2. Class Diagram

<<Interface>> **Interface**
+ goBack(): void

fig 2.1 Interface Diagram for Interface

newMember **(extends javax.swing.JFrame)** **(implements Interface)**
- ageData: JTextField - ageLbl: JLabel - amountData: JTextField - amountLbl: JLabel - closeNM: JButton - emailData: JTextField - emailLbl: JLabel - fatherData: JTextField - fatherLbl: JLabel - genderData: JComcoBox<String> - genderLbl: JLabel - idDataLbl: JLabel - idLbl: JLabel - idNumData: JTextField - idNumLbl: JLabel - mobileData: JTextField - mobileLbl: JLabel - motherData: JTextField - motherLbl: JLabel - nameData: JTextField - nameLbl: JLabel - newMemberLbl: JLabel - panel: JPanel - resetBtn: JButton - saveBtn: JButton - timeData: JComcoBox<String> - timeLbl: JLabel
+ goBack(): void + newMember() - initComponents(): void - closeNMActionPerformed (ActionEvent evt): void - saveBtnActionPerformed (ActionEvent evt): void - resetBtnActionPerformed (ActionEvent evt): void

fig 2.2 Class Diagram for newMember

main **(extends javax.swing.JFrame)**
- exit: JMenu - listMember: JMenu - logout: JMenu - menuBar: JMenuBar - newMember: JMenu - panel: JPanel - payment: JMenu - updateNdelete: JMenu - welcomeLbl: JLabel
+ main() - initComponents(): void - logoutMouseClicked (MouseEvent evt): void - exitMouseClicked (MouseEvent evt): void - newMemberMouseClicked (MouseEvent evt): void - updateNdeleteMouseClicked (MouseEvent evt): void - listMemberMouseClicked (MouseEvent evt): void - paymentMouseClicked (MouseEvent evt): void

fig 2.3 Class Diagram for main

login **(extends javax.swing.JFrame)** **(implements Interface)**
- exitBtn: JButton - loginBg: JLabel - loginBtn: JButton - loginLbl: JLabel - password: PasswordField - showPass: JCheckBox - username: JTextField - warningLbl: JLabel
+ goBack(): void + login() - initComponents(): void - exitBtnActionPerformed (ActionEvent evt): void - loginBtnActionPerformed (ActionEvent evt): void - usernameFocusGained (FocusEvent evt): void - usernameFocusLost (FocusEvent evt): void - passwordFocusGained (FocusEvent evt): void - passwordFocusLost (FocusEvent evt): void - showPassActionPerformed (JActionEvent evt): void

fig 2.4 Class Diagram for login

	updateNdeleteMember **(extends javax.swing.JFrame)** **(implements Interface)**
-	ageData: JTextField
-	ageLbl: JLabel
-	amountData: JTextField
-	amountLbl: JLabel
-	closeBtn: JButton
-	deleteBtn: JButton
-	emailData: JTextField
-	emailLbl: JLabel
-	fatherData: JTextField
-	fatherLbl: JLabel
-	genderData: JTextField
-	genderLbl: JLabel
-	idData: JTextField
-	idLbl: JLabel
-	idNumData: JTextField
-	idNumLbl: JLabel
-	mobileNumData: JTextField
-	mobileNumLbl: JLabel
-	motherData: JTextField
-	motherLbl: JLabel
-	nameData: JTextField
-	nameLbl: JLabel
-	searchBtn: JButton
-	resetBtn: JButton
-	timeData: JTextField
-	timeLbl: JLabel
-	updateBtn: JButton
-	updateNdeleteMemberLbl: JLabel
-	panel: JPanel
+	goBack(): void
+	updateNdeleteMember()
-	initComponents(): void
-	closeBtnActionPerformed (ActionEvent evt): void
-	resetBtnActionPerformed (ActionEvent evt): void
-	searchBtnActionPerformed (ActionEvent evt): void
-	updateBtnActionPerformed (ActionEvent evt): void
-	deleteBtnActionPerformed (ActionEvent evt): void

fig 2.5 Class Diagram for updateNdeleteMember

payment
(extends javax.swing.JFrame)
(implements Interface)

```
-   amountData: JTextfield
-   amountLbl: JLabel
-   closeBtn: JButton
-   dateData: JLabel
-   dateLbl: JLabel
-   emailData: JTextfield
-   emailLbl: JLabel
-   idData: JTextfield
-   idLbl: JLabel
-   mobileNumData: JTextfield
-   mobileNumLbl: JLabel
-   nameData: JTextfield
-   nameLbl: JLabel
-   panel: JPanel
-   resetBtn: JButton
-   saveBtn: JButton
-   paymentLbl: JLabel
-   scrollPane: JScrollPane
-   searchBtn: JButton
-   table: JTable
```

```
+   goBack(): void
+   payment()
+   tableDetails(): void
+   date(): void
-   initComponents(): void
-   searchBtnActionPerformed (ActionEvent evt): void
-   closeBtnActionPerformed (ActionEvent evt): void
-   resetBtnActionPerformed (ActionEvent evt): void
-   saveBtnActionPerformed (ActionEvent evt): void
```

fig 2.6 Class Diagram for payment

listMember **(extends javax.swing.JFrame)** **(implements Interface)**
- closeBtn: JButton - listMemberLbl: JLabel - panel: JPanel - scrollPane: JScrollPane - table: JTable
+ goBack(): void + listMember() - initComponents(): void - closeBtnActionPerformed (ActionEvent evt): void

fig 2.7 Class Diagram for listMember

IMPLEMENTATION

1. Method used in Program

login.java

→ This file is for the login page, it extends the javax.swing.JFrame class and implements the Interface interface.

- public void goback():
  - this method is implemented from the Interface interface
  - make the main page visible and dispose the current page
- public login():
  - call the initComponents method
  - set the warning label (Incorrect Username or Password) invisible when first starting the page
- private void initComponents():
  - initialize all components
- private void exitBtnActionPerformed(java.awt.event.ActionEvent evt):
  - make pop-up message appear to ask for comfirmation to exit system when the exit button is clicked
- private void loginBtnActionPerformed(java.awt.event.ActionEvent evt):
  - if the username and password are correct, redirect user to the main page
  - if the username and password are incorrect, make the warning label appear to let the user knows
- private void usernameFocusGained(java.awt.event.FocusEvent evt):
  - set the warning label to invisible
  - if the username textfield is “Enter username” when clicked, make the textfield to nothing or no text
- private void usernameFocusLost(java.awt.event.FocusEvent evt):
  - set the warning label to invisible
  - if the username textfield is nothing or no text, make the textfield to “Enter username”
- private void passwordFocusGained(java.awt.event.FocusEvent evt):
  - set the warning label to invisible
  - if the password textfield is “Enter password” when clicked, make the textfield to nothing or no text
- private void passwordFocusLost(java.awt.event.FocusEvent evt):
  - set the warning label to invisible
  - if the password textfield is nothing or no text, make the textfield to “Enter password”

- private void showPassActionPerformed(java.awt.event.ActionEvent evt):
    - if the show password checkbox is ticked, whatever written in the password textfield will appear as it is
    - if the show password checkbox is not ticked, whatever written in the password textfield will appear as *

main.java

→ This file is for the main page, it extends the javax.swing.JFrame class but does not implement anything.

- public main():
    - call the initComponents method
- private void initComponents():
    - initialize all components
- private void logoutMouseClicked(java.awt.event.MouseEvent evt):
    - make pop-up message appear to ask for comfirmation to logout when the lougout menu is clicked
- private void exitMouseClicked(java.awt.event.MouseEvent evt):
    - make pop-up message appear to ask for comfirmation to exit system when the exit menu is clicked
- private void newMemberMouseClicked(java.awt.event.MouseEvent evt):
    - make the page to add new member visible and dispose the current page
- private void updateNdeleteMouseClicked(java.awt.event.MouseEvent evt):
    - make the page to update and delete members visible and dispose the current page
- private void listMemberMouseClicked(java.awt.event.MouseEvent evt):
    - make the page to view all members visible and dispose the current page
- private void paymentMouseClicked(java.awt.event.MouseEvent evt):
    - make the page for payment visible and dispose the current page

newMember.java

→ This file is for the page to add new members, it extends javax.swing.JFrame class and implements the Interface interface.

- public void goback():
    - this method is implemented from the Interface interface
    - make the main page visible and dispose the current page
- private void initComponents():
    - initialize all components
- private void closeNMActionPerformed(java.awt.event.ActionEvent evt):
    - call the goback method

- public newMember():
    - call the initComponents method
    - use the try catch block, the try is to:
    - make an id counter
    - establish a connection to the database
    - create a statement object to send SQL statements to the database
    - retrieve the max id from the member table in the database
    - while the next row is available, increase the id count by one
    - the catch block is to display what went wrong (error)
- private void saveBtnActionPerformed(java.awt.event.ActionEvent evt):
    - save each data to a variable
    - use the try catch block, the try is to:
    - create a connection and prepare a parameterized SQL statement to insert data into the member table in database
    - assign values to the parameters
    - insert data to the database and display message to let the users know
    - reset the page
    - the catch block is to display what went wrong (error)
- private void resetBtnActionPerformed(java.awt.event.ActionEvent evt):
    - make the newMember page visible and dispose the current page

updateNdeleteMember.java

→ This file is for the page to update and delete members, it extends javax.swing.JFrame class and implements the Interface interface.

- public void goback():
    - this method is implemented from the Interface interface
    - make the main page visible and dispose the current page
- public updateNdeleteMember():
    - call the initComponents method
- private void initComponents():
    - initiize all components
- private void closeBtnActionPerformed(java.awt.event.ActionEvent evt):
    - call the goback method
- private void resetBtnActionPerformed(java.awt.event.ActionEvent evt):
    - make the updateNdeleteMember page visible ad dispose the current page

- private void searchBtnActionPerformed(java.awt.event.ActionEvent evt):
    - make a counter to check the id exist or not
    - get the text value of id input by users
    - use the try catch block, the try is to:
    - create a connection to the database and create a statement to execute a select query to retrieve member data based on the given id
    - while the next row is available, set the counter to indicate that the id exist
    - make the idData uneditable
    - display the data retrieved from database and set them editable except for gender and gym time
    - if the counter is 0, display a message to let the users know that id does not exist
    - the catch block is to display what went wrong (error)
- private void updateBtnActionPerformed(java.awt.event.ActionEvent evt):
    - retrieve all the new data inputted by user
    - use the the try catch block, the try is to:
    - create a connection to the database and prepare an update statement to modify member data based on the given id
    - assigning new values to the statement parameters
    - update the new data to database and display a message to let users know
    - reset the page
    - the catch block is to display what went wrong (error)
- private void deleteBtnActionPerformed(java.awt.event.ActionEvent evt):
    - ask user for confirmation to delete member
    - if user chooses to yes, get the id value that the user inputted
    - use a try catch block, the try is to:
    - create a connection to the database and make a statement to execute the delete query to remove member from the member table inn database based on the given id
    - show message to let users know
    - reset the page
    - the catch block is to display what went wrong (error)

listMember.java

→ This file is to view all members' data, it extends javax.swing.JFrame class and implements the Interface interface.

- public void goback():
    - this method is implemented from the Interface interface
    - make the main page visible and dispose the current page

- public listMember():
    - call the initComponents method
    - get the default table model from JTable
    - use a try catch block, the try is to:
    - create a connection to the database and make a statement to execute the select query to retrieve user data from the member data in the database
    - while the next row is available, add data to the table row by row
    - the catch block is to display what went wrong (error)
- private void initComponents():
    - initialize all components
- private void closeBtnActionPerformed(java.awt.event.ActionEvent evt):
    - call the goBack method

payment.java

→ This file is for the payment page, it extends javax.swing.JFrame class and implements the Interface interface.

- public void goBack():
    - this method is implemented from the Interface interface
    - make the main page visible and dispose the current page
- public void tableDetails():
    - get the default table model from JTable
    - clear all data from the table when the method is called
    - get the value of the id inputted by user
    - use a try catch block, the try is to:
    - create a connection to the database and make a statement to execute the select query to retrieve user data from the payment table based on the id
    - while the next row is available, add data to the table row by row
    - the catch block is to display what went wrong (error)
- public void date():
    - create a simpleDateFormat object with the MMM-YYY pattern
    - create a date object to represent the current date
    - set the format when retrieving date
    - assigning value to the dateData component
- public payment():
    - call the initComponents method
    - call the date method
- private void initComponents():
    - initialize all components

- private void searchBtnActionPerformed(java.awt.event.ActionEvent evt):
    - call the tableDetails method
    - make a counter to check the id exist or not
    - get the text value of id and month
    - use a try catch block, the try is to:
    - create a connection to the database and make a statement to execute the select query to retrieve user data from the member table based on the id
    - while the next row is available, set the counter to 1 to indicate that the id exist and make the id uneditable
    - retrieve the data from the database and display them
    - if the counter is 0, show a message to let the users know that the id does not exist
    - executes a database query that selects record from the payment and member table using an inner join based on month and id for payment table and id for member table
    - while the next row is available, hide the save button and let the users know that the payment is done for the month
    - the catch block is to display what went wrong (error)
- private void closeBtnActionPerformed(java.awt.event.ActionEvent evt):
    - call the goBack method
- private void resetBtnActionPerformed(java.awt.event.ActionEvent evt):
    - make the payment page visible and dispose the current page
- private void saveBtnActionPerformed(java.awt.event.ActionEvent evt):
    - get the text value id, date, and amount and store each of them in a variable
    - use the try catch block, the try is to:
    - create a connection and prepare an insert statement to insert payment data
    - assigning values to the parameters of the prepared statement
    - update the value to the database
    - call the tableDetails method
    - show a message to let the users know that data has been updated
    - reset the page
    - the catch block is to display what went wrong (error)

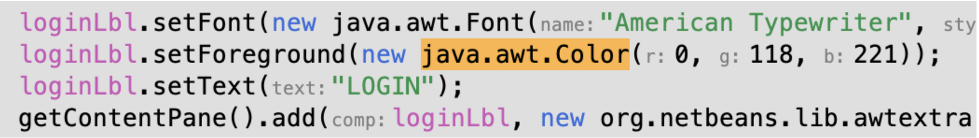
<u>Driver.java</u>

→ This file is used to run or start the program. It only contains the command to call the login page.

2. Program Dependencies

- java.awt.Color;

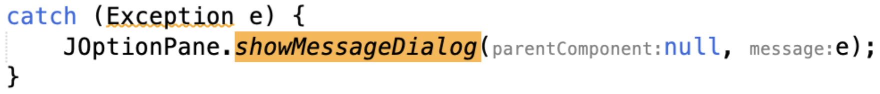
  → to set the components color



```
loginLbl.setFont(new java.awt.Font(name: "American Typewriter", sty
loginLbl.setForeground(new java.awt.Color(r: 0, g: 118, b: 221));
loginLbl.setText(text: "LOGIN");
getContentPane().add(comp: loginLbl, new org.netbeans.lib.awtextra
```

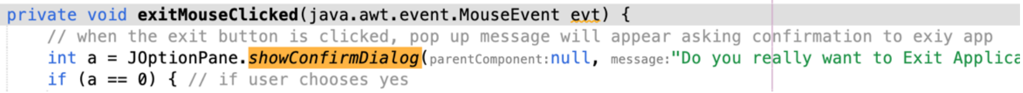
- javax.swing.JOptionPane;

  → showMessageDialog()



```
catch (Exception e) {
    JOptionPane.showMessageDialog(parentComponent:null, message:e);
}
```

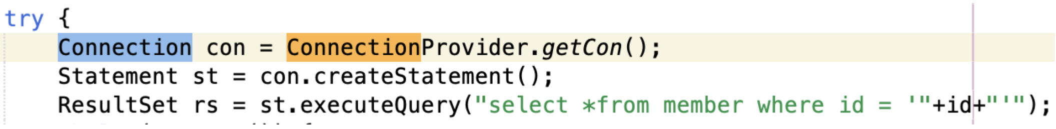
  → showConfirmDialog()



```
private void exitMouseClicked(java.awt.event.MouseEvent evt) {
    // when the exit button is clicked, pop up message will appear asking confirmation to exiy app
    int a = JOptionPane.showConfirmDialog(parentComponent:null, message:"Do you really want to Exit Applica
    if (a == 0) { // if user chooses yes
```

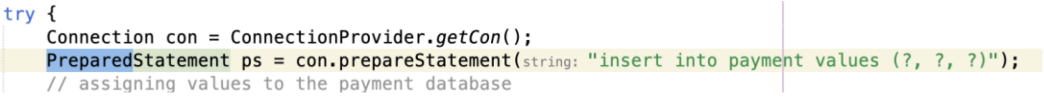
- java.sql.*;

  → Connection, Statement, ResultSet



```
try {
    Connection con = ConnectionProvider.getCon();
    Statement st = con.createStatement();
    ResultSet rs = st.executeQuery("select *from member where id = '"+id+"'");
```

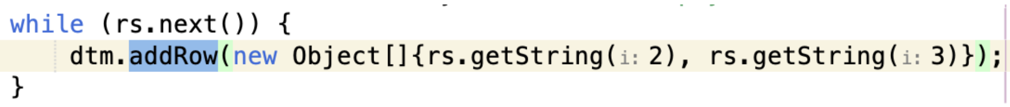
  → PreparedStatement



```
try {
    Connection con = ConnectionProvider.getCon();
    PreparedStatement ps = con.prepareStatement(string: "insert into payment values (?, ?, ?)");
    // assigning values to the payment database
```

- javax.swing.table.DefaultTableModel;

  → addRow



```
while (rs.next()) {
    dtm.addRow(new Object[]{rs.getString(i: 2), rs.getString(i: 3)});
}
```

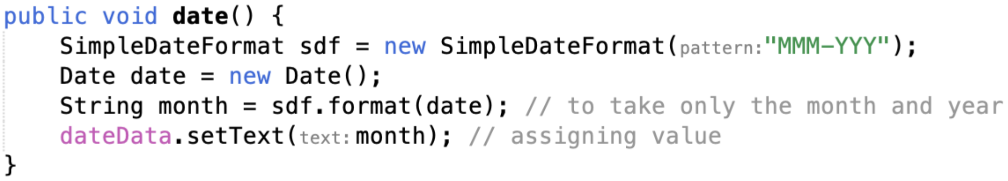
- java.util.Date;

  → create new date

  → retrieving date

- java.text.SimpleDateFormat;

  → formatting date with the pattern "MMM-YYY"



```
public void date() {
    SimpleDateFormat sdf = new SimpleDateFormat(pattern:"MMM-YYY");
    Date date = new Date();
    String month = sdf.format(date); // to take only the month and year
    dateData.setText(text: month); // assigning value
}
```

REFLECTION

During the process of making this gym management system, I met a lot of difficulties which I needed external help to solve. Firstly, when I have no idea how to make a certain thing, I would either find the solution on google or youtube videos. This is actually not hard as nowadays the internet has almost everything covered, however, sometimes a lot of time is needed to find the solution. Secondly, along the way of coding, not a few errors occurred. Usually for codes, I can find the solution in stackoverflow website.

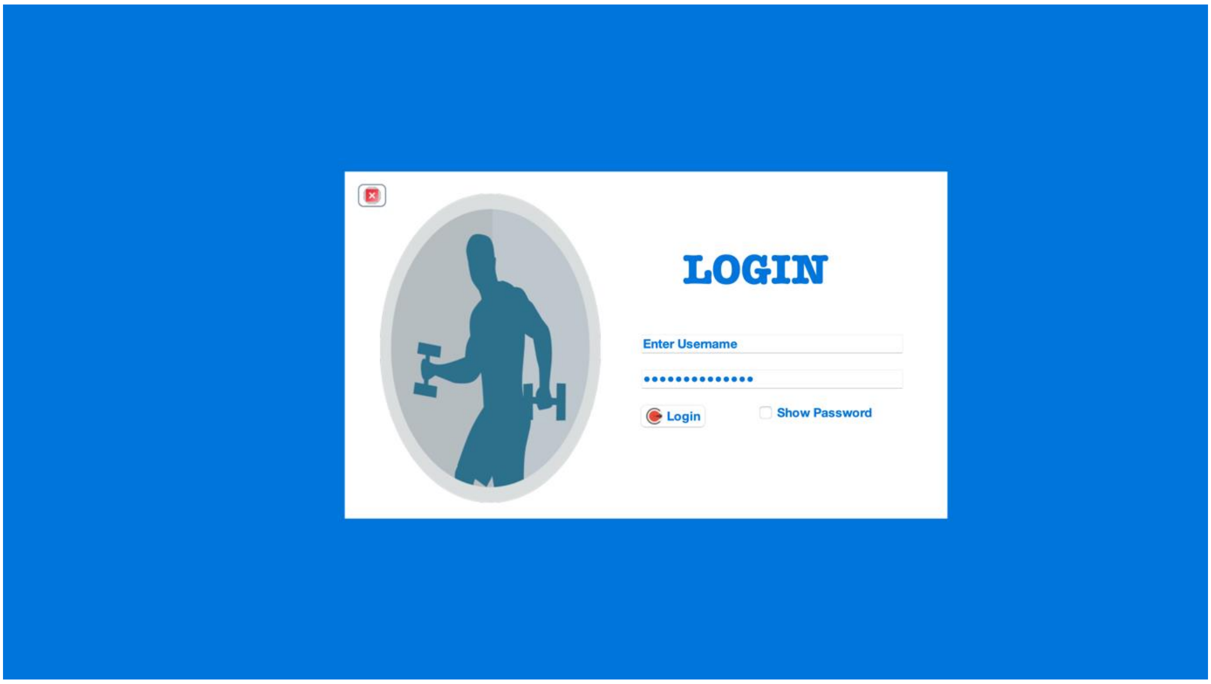
What I learnt through making this system is to not panic easily as some problems actually have really simple solutions but are made hard by our own overthinking. And I found that I am the person who needs some time to finish making something because my inspiration cannot come in a short amount of time.

Regarding this gym management system, the only trouble I still cannot solve is to use javaFx GUI. Personally, I think it is so hard to set it up, I have follow many youtube tutorials in setting it up, but mine still cannot run as it still display some error which I am really confused with. Thus, after trying for a week, I finally decided to give it up and just use java swing instead. However, I will continue to try javaFx but just not for this final project, apparently.

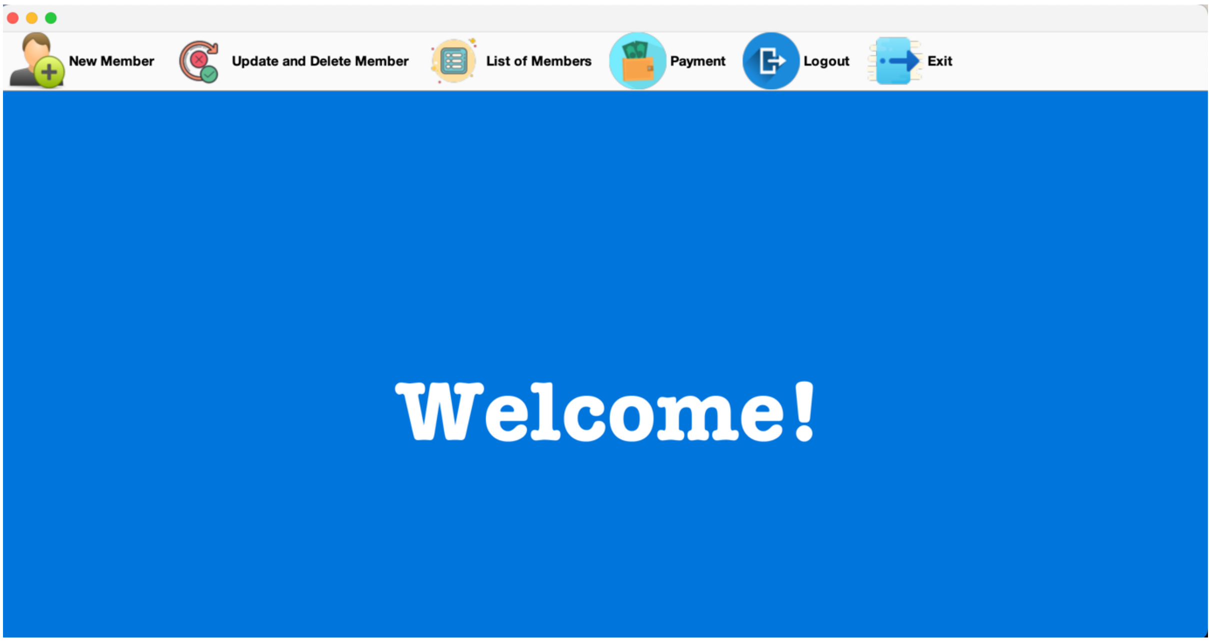
In my opinion, this gym management system can still be improved in some way. For now, some of the improvements I can think of are; a feature to make the user able to make membership and a feature for the user to choose what type of gym classes they want to take.

EVIDENCE OF A WORKING PROGRAM

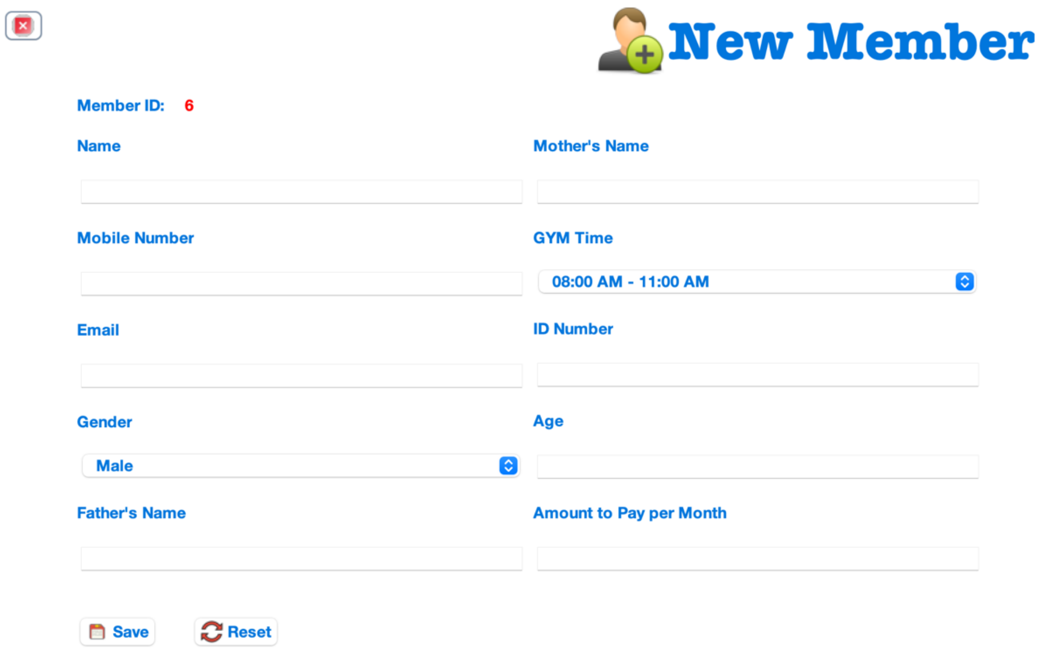
Login Page



Main Page



New Member Page

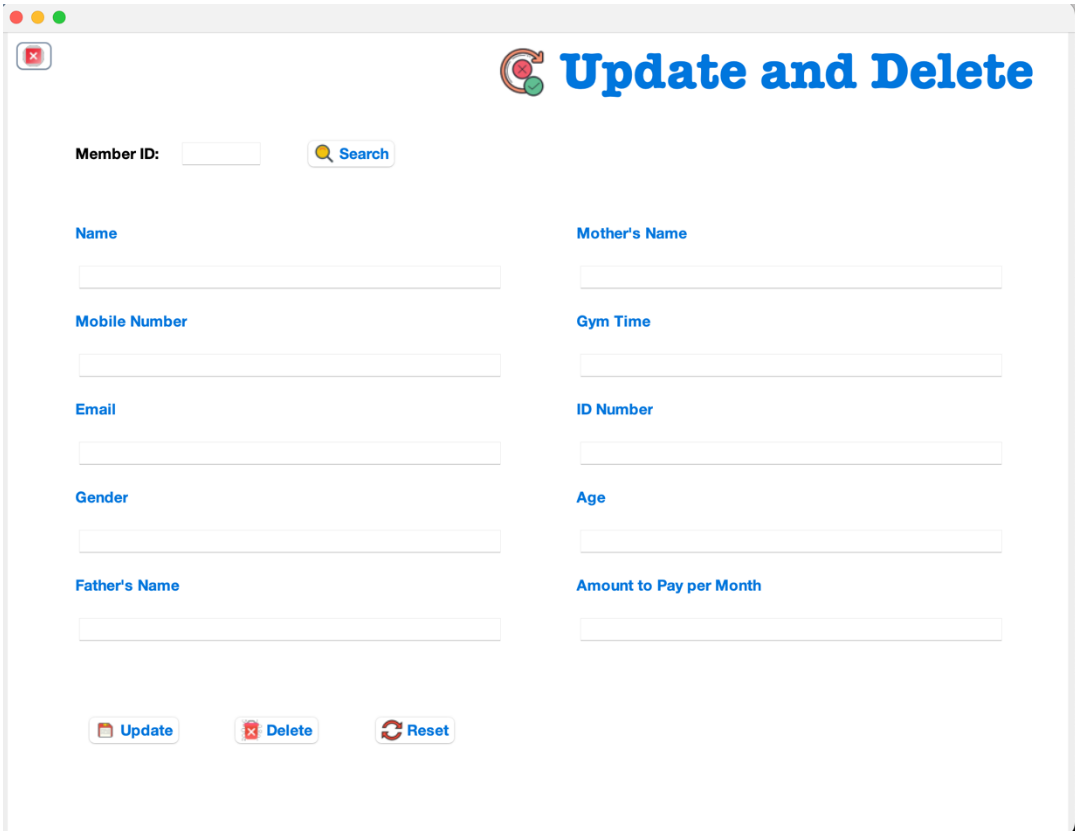


Member ID: 6

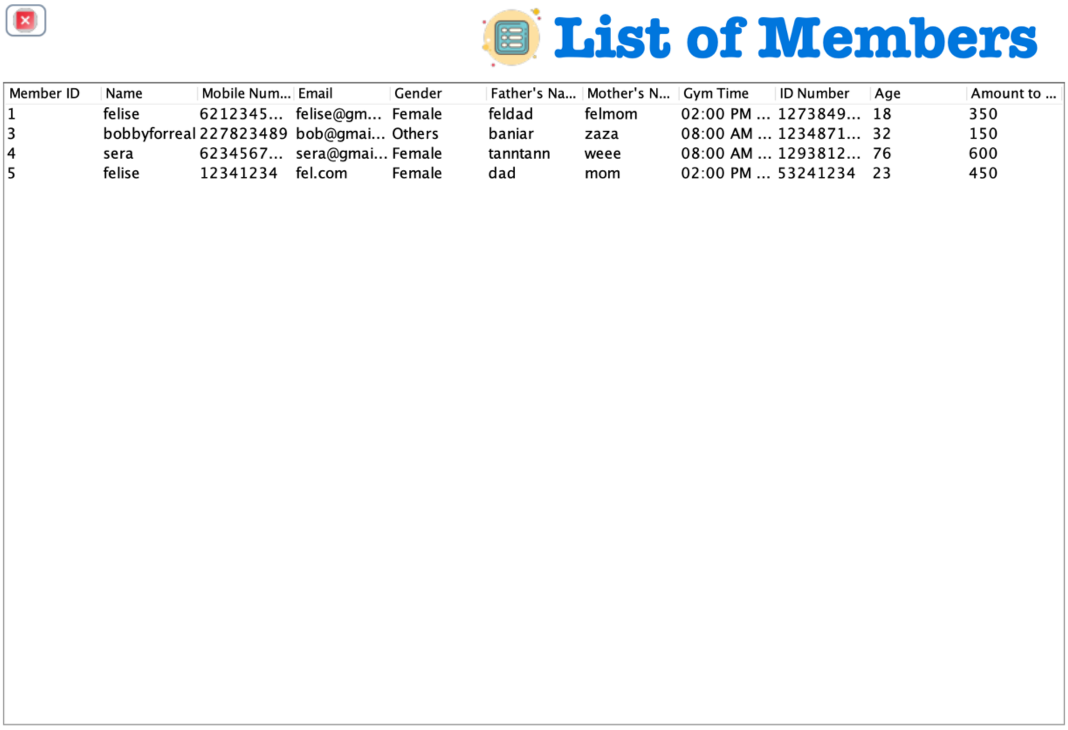
GYM Time: 08:00 AM - 11:00 AM

Gender: Male

Update and Delete Member Page

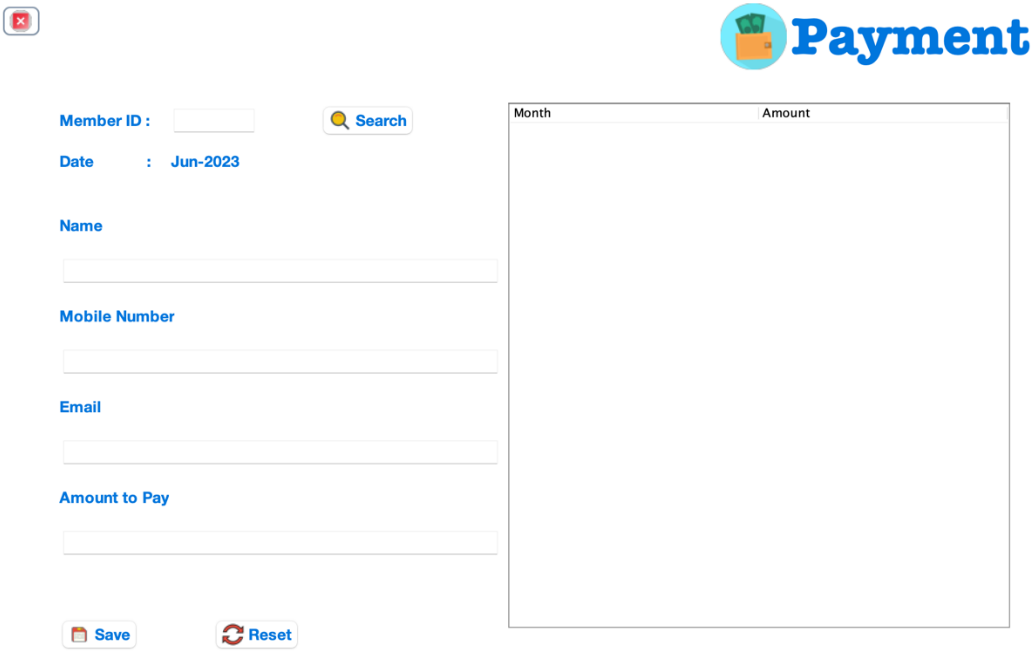


List of Members Page



Member ID	Name	Mobile Num...	Email	Gender	Father's Na...	Mother's N...	Gym Time	ID Number	Age	Amount to ...
1	felise	6212345...	felise@gm...	Female	feldad	felmom	02:00 PM ...	1273849...	18	350
3	bobbyforreal	227823489	bob@gmai...	Others	baniar	zaza	08:00 AM ...	1234871...	32	150
4	sera	6234567...	sera@gmai...	Female	tanntann	weee	08:00 AM ...	1293812...	76	600
5	felise	12341234	fel.com	Female	dad	mom	02:00 PM ...	53241234	23	450

Payment Page



Member ID :

Date : Jun-2023

Name

Mobile Number

Email

Amount to Pay

Month	Amount