

Umrabulo Session #03 – Gym Frame

Author: Vuyisile Memani

Module: AOP216D/AOR216D

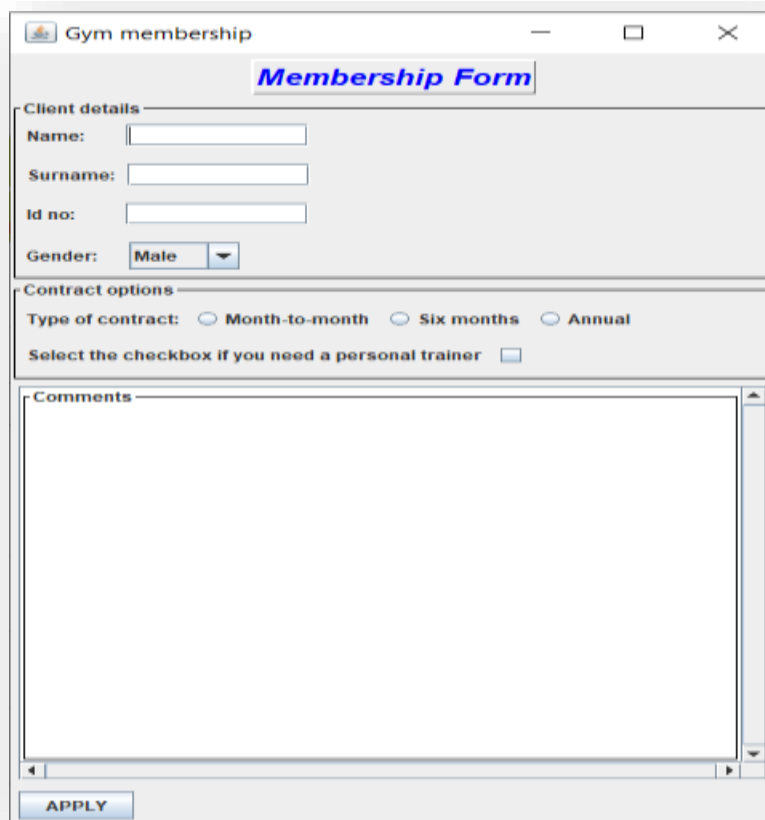
Date: 15 July 2023

Introduction

In this Mrabulo Session we are going to create a greetings frame (form) using the **JFrame** class. The tutorial will introduce the student to more Swing components such as **JComboBox**, **JRadioButton**, **JCheckBox**, and **JScrollPane** for a scrollable text area. The developed solution will be stored on **GitHub**. The tutorial will conclude with a **DIY** (Do It Yourself) exercise.

Problem statement

Create a membership frame and display it. The output should resemble the following frame:



The screenshot shows a Java Swing window titled "Gym membership" with standard window controls (minimize, maximize, close). Inside the window is a form titled "Membership Form" in blue text. The form is divided into three sections: "Client details" with input fields for Name, Surname, and Id no, and a dropdown menu for Gender (currently showing "Male"); "Contract options" with three radio buttons for "Month-to-month", "Six months", and "Annual", and a checkbox for "Select the checkbox if you need a personal trainer"; and "Comments" which is a large text area with a scrollbar. At the bottom of the window is an "APPLY" button.

Solution

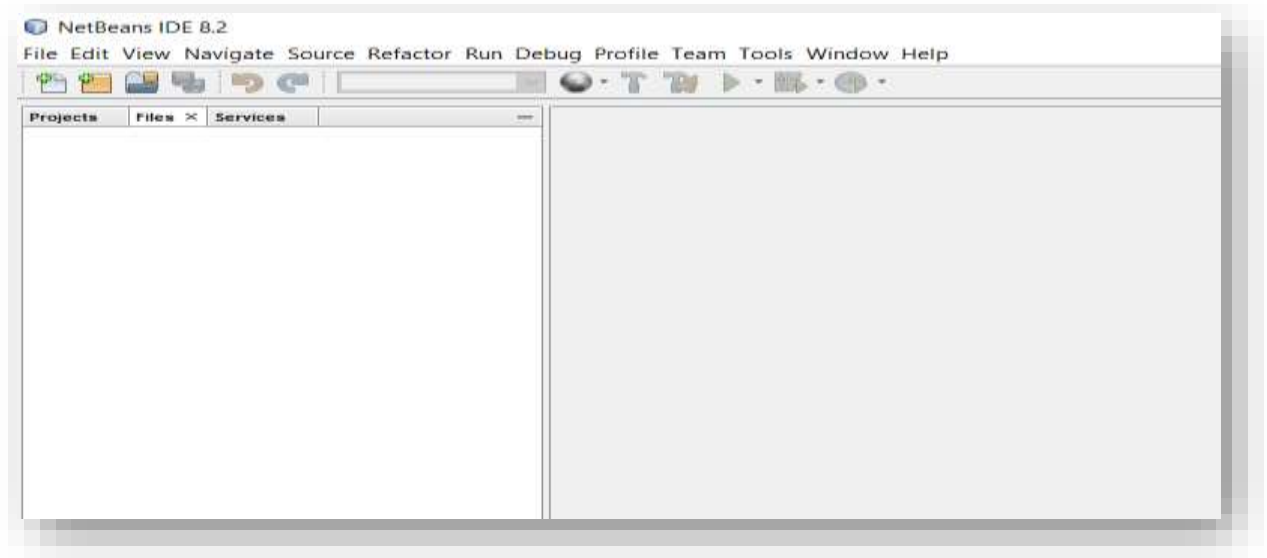
The solution is divided into three parts, namely A, B and C. In part A we create the frame, in B we test it, and in C we store our source code on GitHub.

Part A: Create the frame

In this section we create the actual frame in a backend class.

Step 1

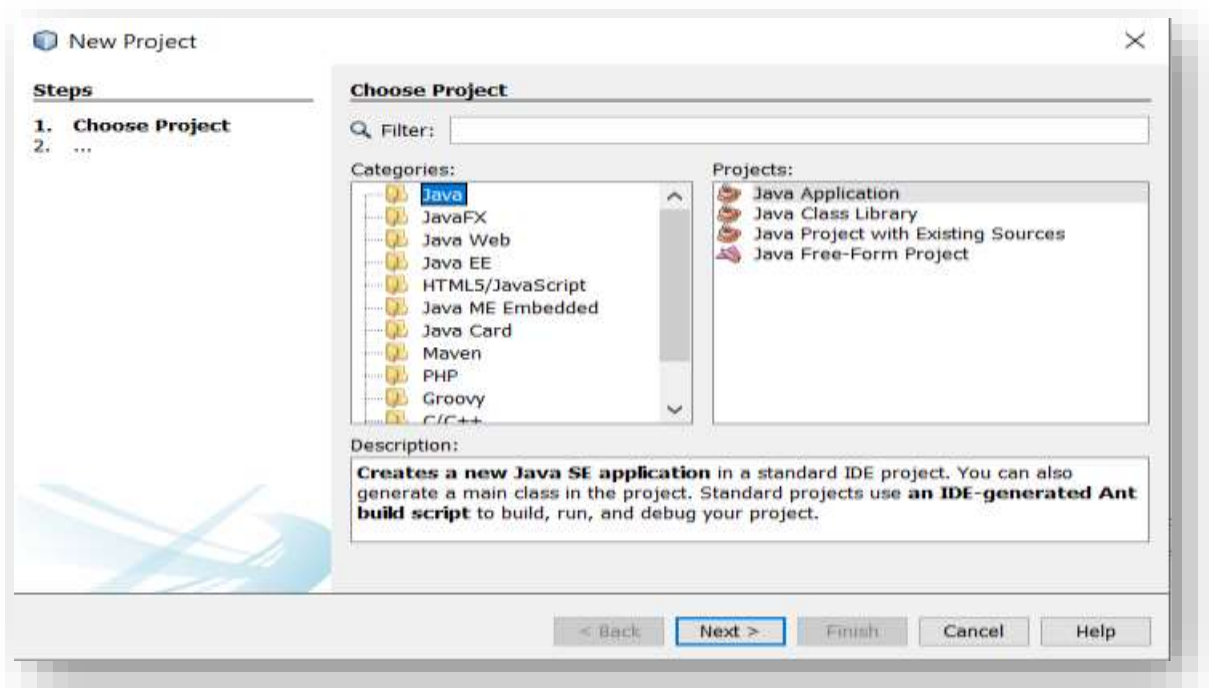
Launch NetBeans.



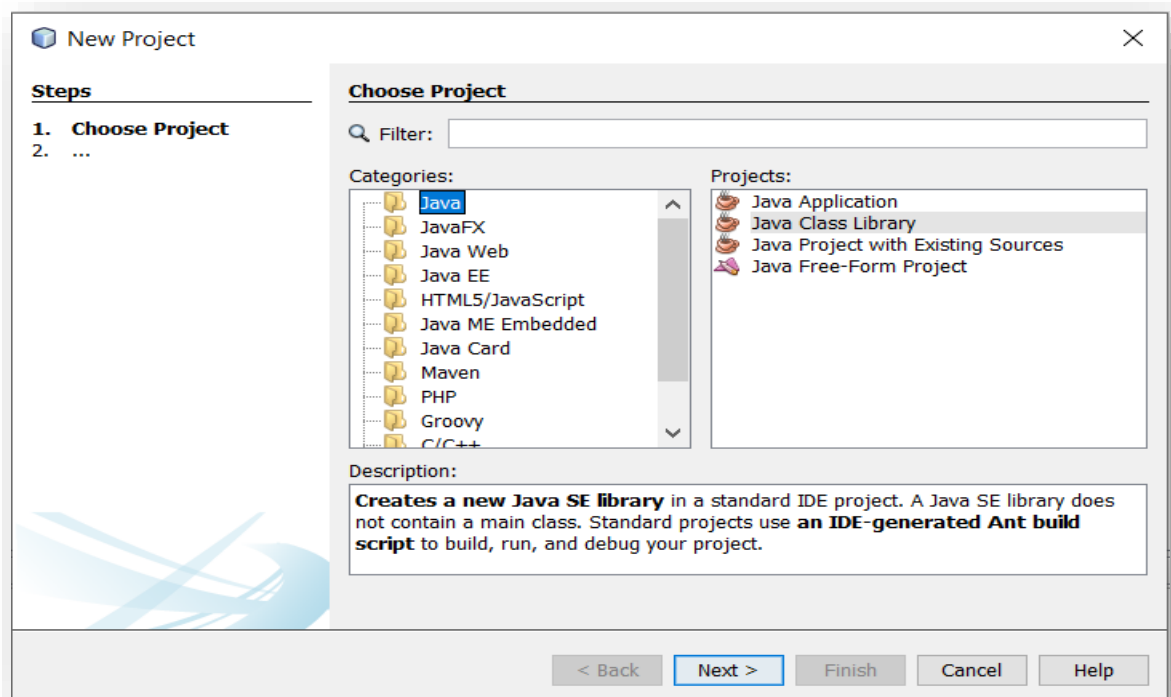
Step 2

Create a Java Project. Perform the following tasks:

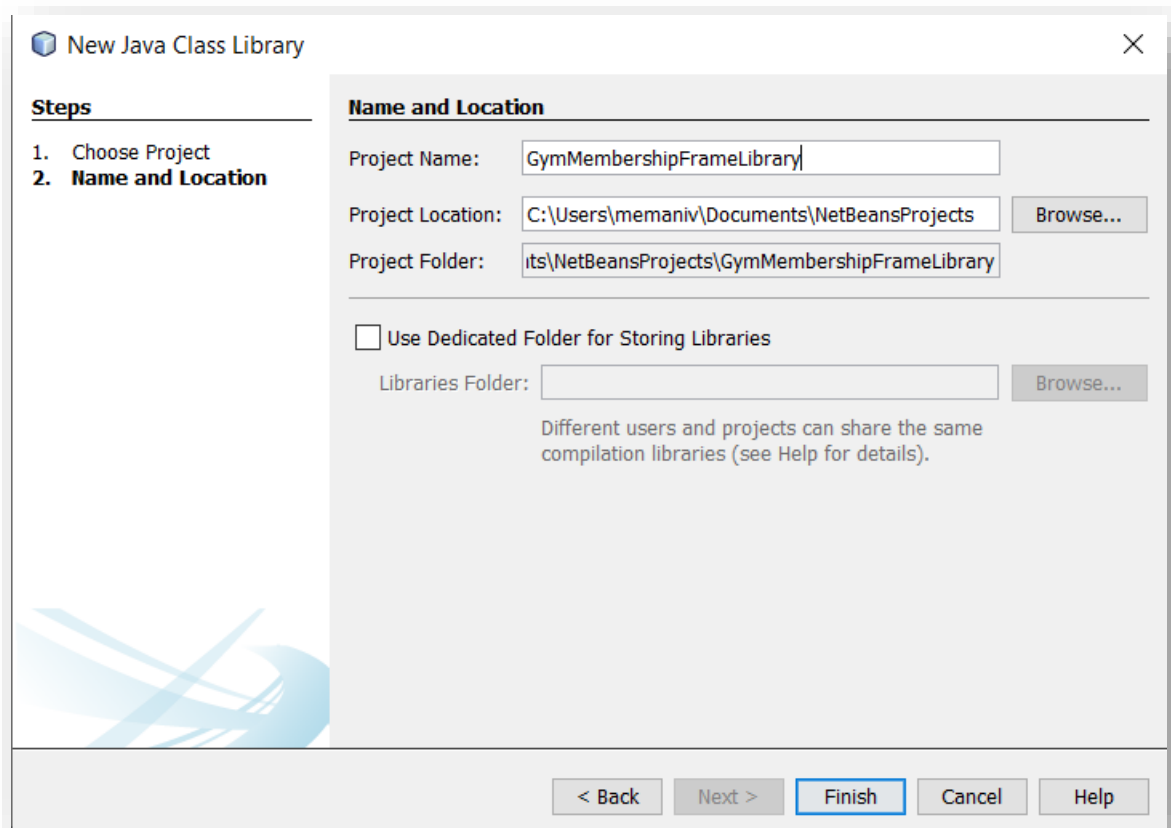
- Click on **File | New Project**



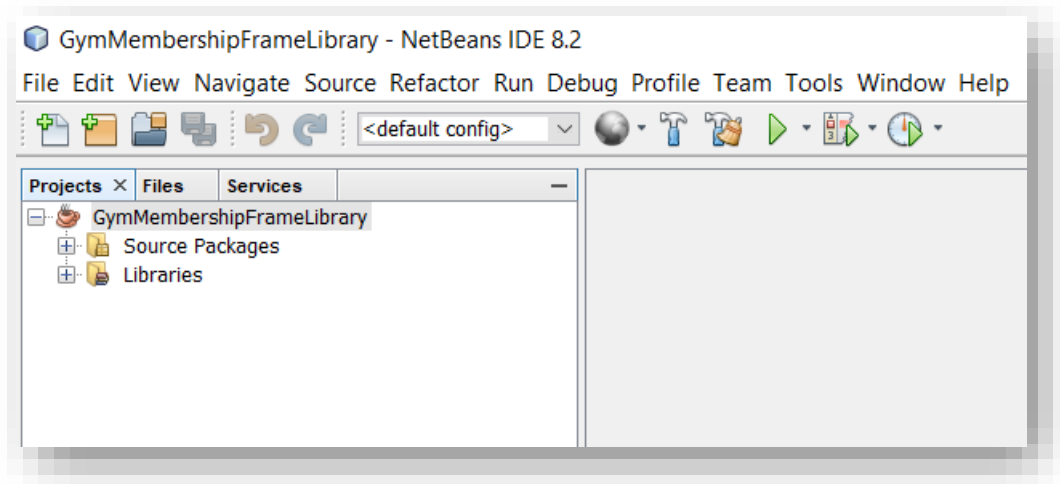
- Select **Java** under **Categories** and **Java Class Library** under **Projects**.



- Click **Next**. Name the project **GymMembershipFrameLibrary**.



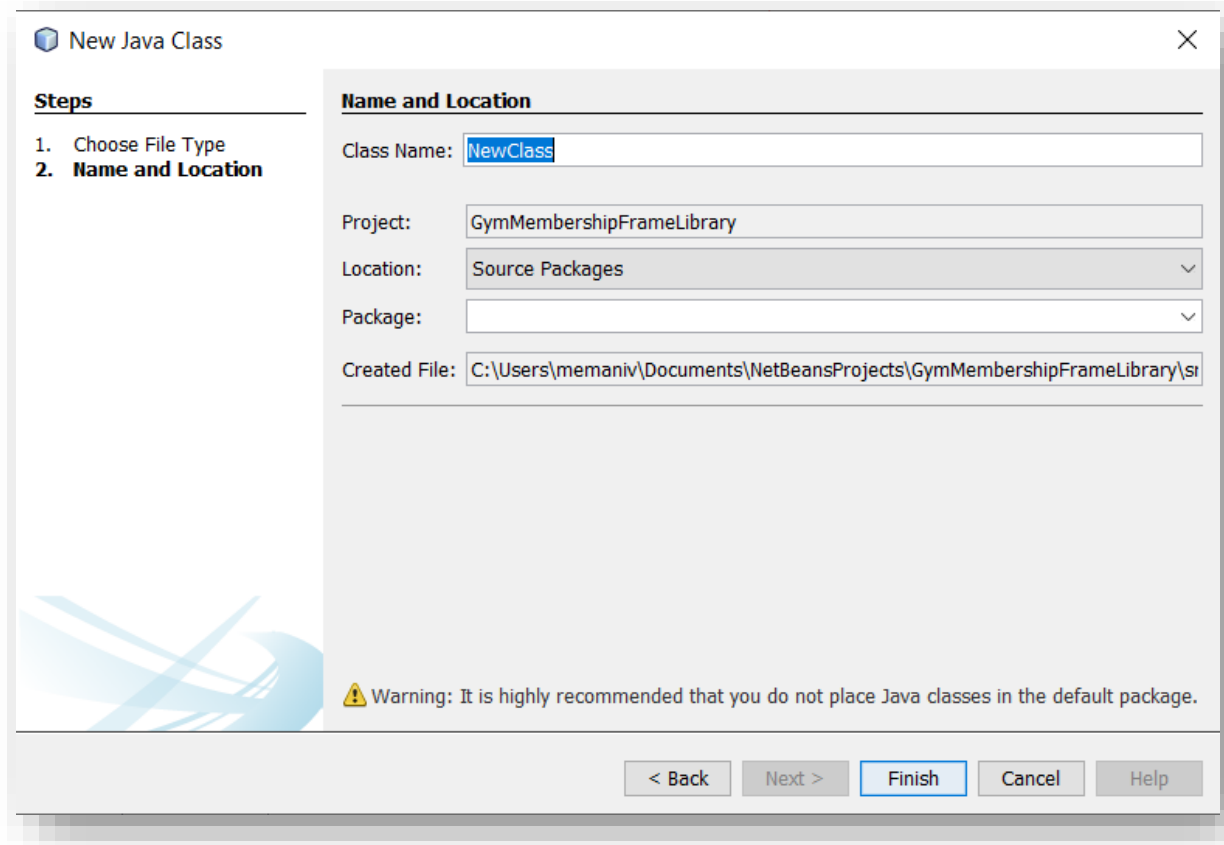
- Click **Finish**.



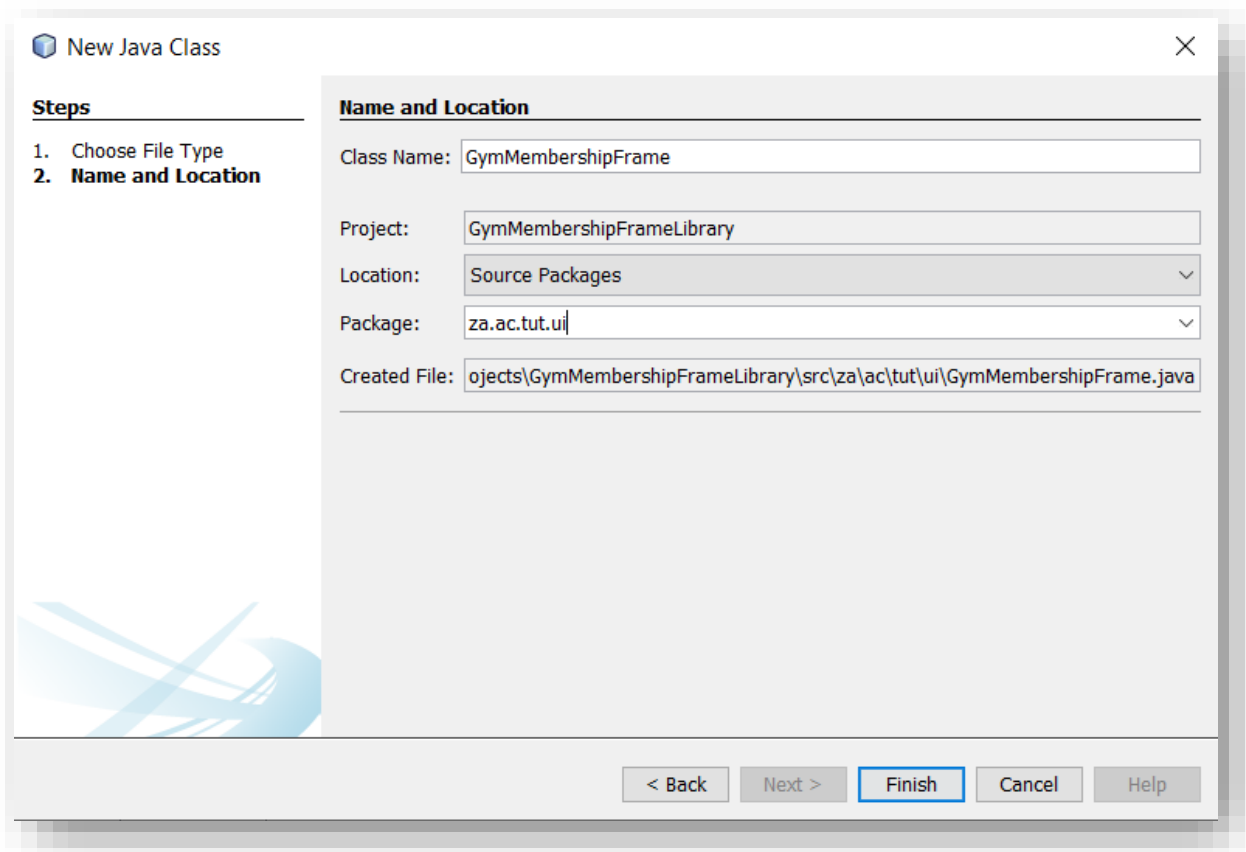
Step 3

Create the frame. Perform the following tasks:

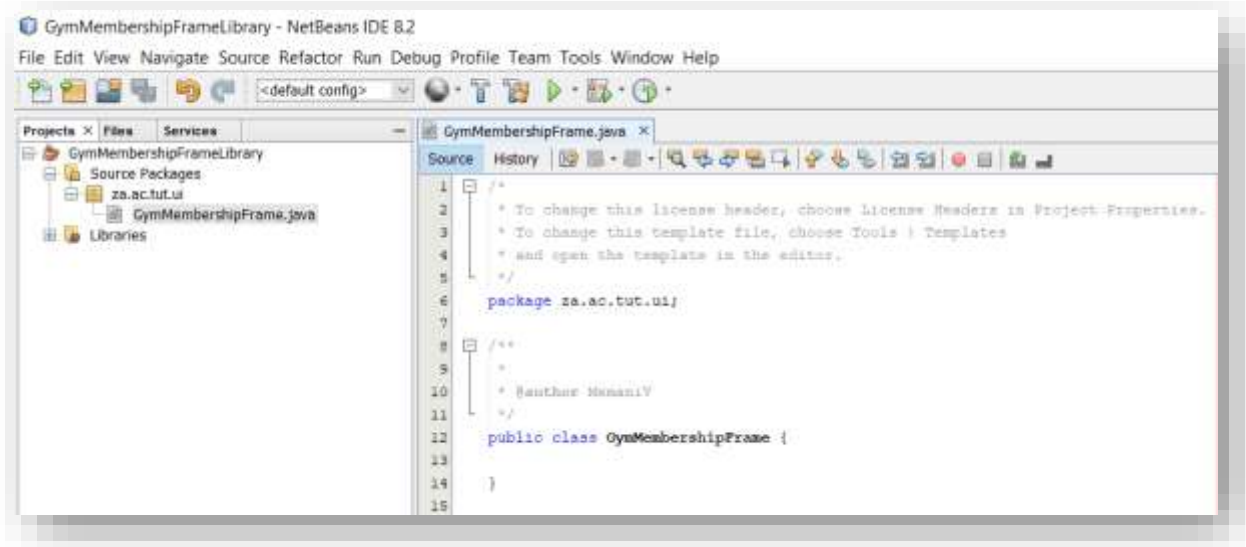
- Right click on the project and select **New | Java class**



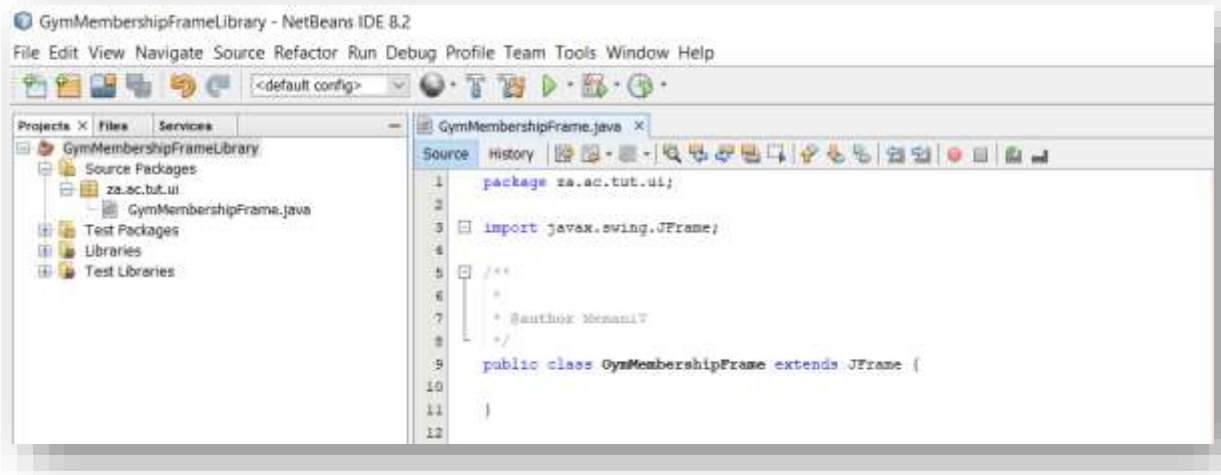
- Name the class as **GymMembershipFrame** and package it under **za.ac.tut.ui**



- Click **Finish**.



- Make the **GymMembershipFrame** class a frame by extending the **JFrame**.



- Create the frame.

```

1  package za.ac.tut.ui;
2
3  import java.awt.BorderLayout;
4  import java.awt.Color;
5  import java.awt.FlowLayout;
6  import java.awt.Font;
7  import java.awt.GridLayout;
8  import javax.swing.ButtonGroup;
9  import javax.swing.JButton;
10 import javax.swing.JCheckBox;
11 import javax.swing.JComboBox;
12 import javax.swing.JFrame;
13 import javax.swing.JLabel;
14 import javax.swing.JPanel;
15 import javax.swing.JRadioButton;
16 import javax.swing.JScrollPane;
17 import javax.swing.JTextArea;
18 import javax.swing.JTextField;
19 import javax.swing.border.LineBorder;
20 import javax.swing.border.SoftBevelBorder;
21 import javax.swing.border.TitledBorder;
22
23 /**
24  *
25  * @author MemaniV
26  */
27 public class GymMembershipFrame extends JFrame {
28     //panels
29     private JPanel headingPnl;
30     private JPanel clientPnl;
31     private JPanel namePnl;
32     private JPanel surnamePnl;
33     private JPanel idNoPnl;
34     private JPanel genderPnl;
35     private JPanel contractsPnl;
36     private JPanel personalTrainerOptionPnl;
37     private JPanel membershipPnl;

```

```

38     private JPanel commentsPnl;
39     private JPanel btnsPnl;
40     private JPanel headingClientCombinedPnl;
41     private JPanel membershipCommentsCombinedPnl;
42     private JPanel mainPnl;
43
44     //labels
45     private JLabel headingLbl;
46     private JLabel nameLbl;
47     private JLabel surnameLbl;
48     private JLabel idNoLbl;
49     private JLabel genderLbl;
50     private JLabel personalTrainerLbl;
51     private JLabel contractTypeLbl;
52
53     //textfields
54     private JTextField nameTxtFld;
55     private JTextField surnameTxtFld;
56     private JTextField idNoTxtFld;
57
58     //combobox
59     private JComboBox genderComboBox;
60
61     //radio buttons
62     private JRadioButton monthToMonthRadBtn;
63     private JRadioButton sixMonthsRadBtn;
64     private JRadioButton annualRadBtn;
65
66     //checkbox
67     private JCheckBox personalTrainerChkBx;
68
69     //buttongroup
70     private ButtonGroup btnGrp;
71
72     //textarea
73     private JTextArea commentsArea;

```

```

75 //scrollpane
76 private JScrollPane scrollableTextArea;
77
78 //private button
79 private JButton applyBtn;
80
81 //construct the gui
82 public GymMembershipFrame(){
83     setTitle("Gym membership");
84     setSize(500, 550);
85
86     //create panels
87     headingPnl = new JPanel(new FlowLayout(FlowLayout.CENTER));
88
89     clientPnl = new JPanel(new GridLayout(4,1,1,1));
90     clientPnl.setBorder(new TitledBorder(new LineBorder(Color.BLACK, 1), "Client details"));
91
92     namePnl = new JPanel(new FlowLayout(FlowLayout.LEFT));
93     surnamePnl = new JPanel(new FlowLayout(FlowLayout.LEFT));
94     idNoPnl = new JPanel(new FlowLayout(FlowLayout.LEFT));
95     genderPnl = new JPanel(new FlowLayout(FlowLayout.LEFT));
96
97     contractsPnl = new JPanel(new FlowLayout(FlowLayout.LEFT));
98     personalTrainerOptionPnl = new JPanel(new FlowLayout(FlowLayout.LEFT));
99
100     membershipPnl = new JPanel(new GridLayout(2,1,1,1));
101     membershipPnl.setBorder(new TitledBorder(new LineBorder(Color.BLACK, 1), "Contract options"));
102
103     commentsPnl = new JPanel(new FlowLayout(FlowLayout.LEFT));
104
105     btnsPnl = new JPanel(new FlowLayout(FlowLayout.LEFT));
106     headingClientCombinedPnl = new JPanel(new BorderLayout());
107     membershipCommentsCombinedPnl = new JPanel(new BorderLayout());
108     mainPnl = new JPanel(new BorderLayout());

```



```

110 //create labels
111 headingLbl = new JLabel("Membership Form");
112 headingLbl.setFont(new Font(Font.SANS_SERIF, Font.ITALIC + Font.BOLD, 20));
113 headingLbl.setForeground(Color.BLUE);
114 headingLbl.setBorder(new SoftBevelBorder(SoftBevelBorder.RAISED));
115
116 nameLbl = new JLabel("Name: ");
117 surnameLbl = new JLabel("Surname: ");
118 idNoLbl = new JLabel("Id no: ");
119 genderLbl = new JLabel("Gender: ");
120 contractTypeLbl = new JLabel("Type of contract: ");
121 personalTrainerLbl = new JLabel("Select the checkbox if you need a personal trainer ");
122
123 //create textfields
124 nameTxtFld = new JTextField(10);
125 surnameTxtFld = new JTextField(10);
126 idNoTxtFld = new JTextField(10);
127
128 //create combobox
129 genderComboBox = new JComboBox();
130 genderComboBox.addItem("Male");
131 genderComboBox.addItem("Female");
132
133 //create radio buttons
134 monthToMonthRadBtn = new JRadioButton("Month-to-month");
135 sixMonthsRadBtn = new JRadioButton("Six months");
136 annualRadBtn = new JRadioButton("Annual");
137
138 //create cscheck box
139 personalTrainerChkBx = new JCheckBox();
140
141 //create button group
142 btnGrp = new ButtonGroup();
143 btnGrp.add(monthToMonthRadBtn);
144 btnGrp.add(sixMonthsRadBtn);
145 btnGrp.add(annualRadBtn);

```

```

147 //create text area
148 commentsArea = new JTextArea(20,40);
149 commentsArea.setBorder(new TitledBorder(new LineBorder(Color.BLACK, 1), "Comments"));
150
151 scrollableTxtArea = new JScrollPane(commentsArea, JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
152                                     JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
153
154 //create button
155 applyBtn = new JButton("APPLY");
156
157 //add components to panels
158 headingPnl.add(headingLbl); //--> first collective panel
159
160 namePnl.add(nameLbl);
161 namePnl.add(nameTxtFld);
162
163 surnamePnl.add(surnameLbl);
164 surnamePnl.add(surnameTxtFld);
165
166 idNoPnl.add(idNoLbl);
167 idNoPnl.add(idNoTxtFld);
168
169 genderPnl.add(genderLbl);
170 genderPnl.add(genderComboBox);
171
172 clientPnl.add(namePnl); //---> second collective panel
173 clientPnl.add(surnamePnl);
174 clientPnl.add(idNoPnl);
175 clientPnl.add(genderPnl);
176
177 headingClientCombinedPnl.add(headingPnl, BorderLayout.NORTH);
178 headingClientCombinedPnl.add(clientPnl, BorderLayout.CENTER);
179
180 contractsPnl.add(contractTypeLbl);
181 contractsPnl.add(monthToMonthRadBtn); //---> third collective panel
182 contractsPnl.add(sixMonthsRadBtn);
183 contractsPnl.add(annualRadBtn);
184

```

```

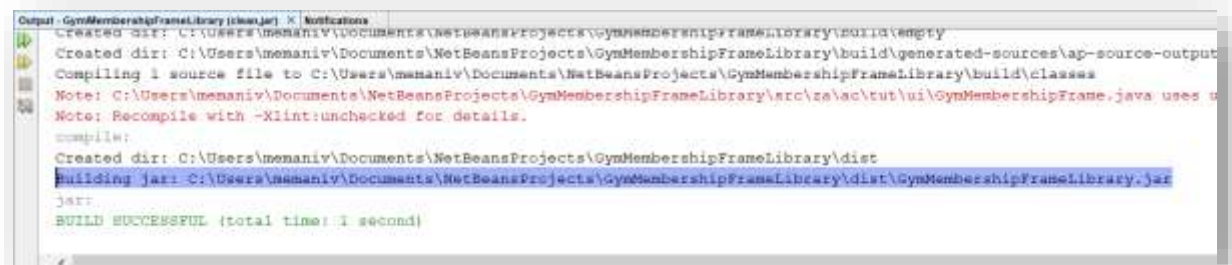
185     personalTrainerOptionPnl.add(personalTrainerLbl);
186     personalTrainerOptionPnl.add(personalTrainerChkBx);
187
188     membershipPnl.add(contractsPnl);
189     membershipPnl.add(personalTrainerOptionPnl);
190
191     commentsPnl.add(scrollableTxtArea); //---> fourth collective panel
192
193     membershipCommentsCombinedPnl.add(membershipPnl, BorderLayout.NORTH);
194     membershipCommentsCombinedPnl.add(commentsPnl, BorderLayout.CENTER);
195
196     btnsPnl.add(applyBtn); //---> fifth collective panel
197
198     mainPnl.add(headingClientCombinedPnl, BorderLayout.NORTH);
199     mainPnl.add(membershipCommentsCombinedPnl, BorderLayout.CENTER);
200     mainPnl.add(btnsPnl, BorderLayout.SOUTH);
201
202     add(mainPnl);
203
204     pack();
205     setVisible(true);
206 }
207
208 }
209

```

Step 4

Compile the project. Perform the following tasks:

- Right click on the project and select **Clean and Build**.

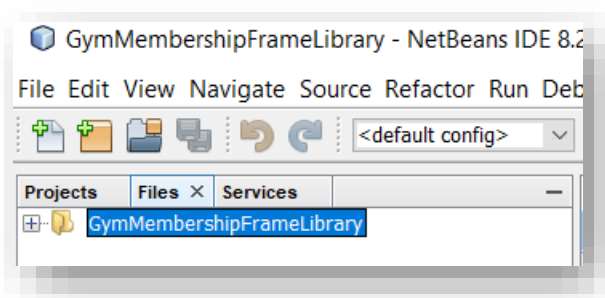


```

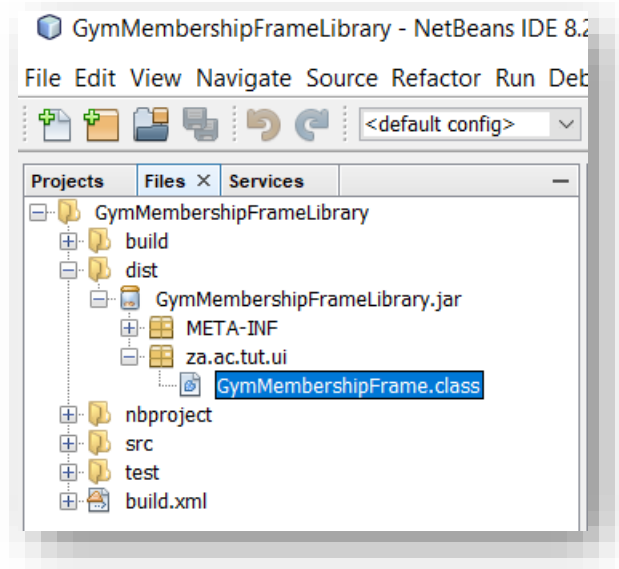
Output - GymMembershipFrameLibrary (clean.jar) X Notifications
Created dir: C:\Users\memaniv\Documents\NetBeansProjects\GymMembershipFrameLibrary\build\empty
Created dir: C:\Users\memaniv\Documents\NetBeansProjects\GymMembershipFrameLibrary\build\generated-sources\ap-source-output
Compiling 1 source file to C:\Users\memaniv\Documents\NetBeansProjects\GymMembershipFrameLibrary\build\classes
Note: C:\Users\memaniv\Documents\NetBeansProjects\GymMembershipFrameLibrary\src\za\ac\tut\ui\GymMembershipFrame.java uses s
Note: Recompile with -Xlint:unchecked for details.
compile:
Created dir: C:\Users\memaniv\Documents\NetBeansProjects\GymMembershipFrameLibrary\dist
Building jar: C:\Users\memaniv\Documents\NetBeansProjects\GymMembershipFrameLibrary\dist\GymMembershipFrameLibrary.jar
JART
BUILD SUCCESSFUL (total time: 1 second)

```

- Select the **Files** tab.



- Expand the folder. Under the **dist** folder you have the **GymMembershipFrameLibrary.jar** file that contains the **GymMembershipFrame.class** file.



Part B: Test the frame in an application

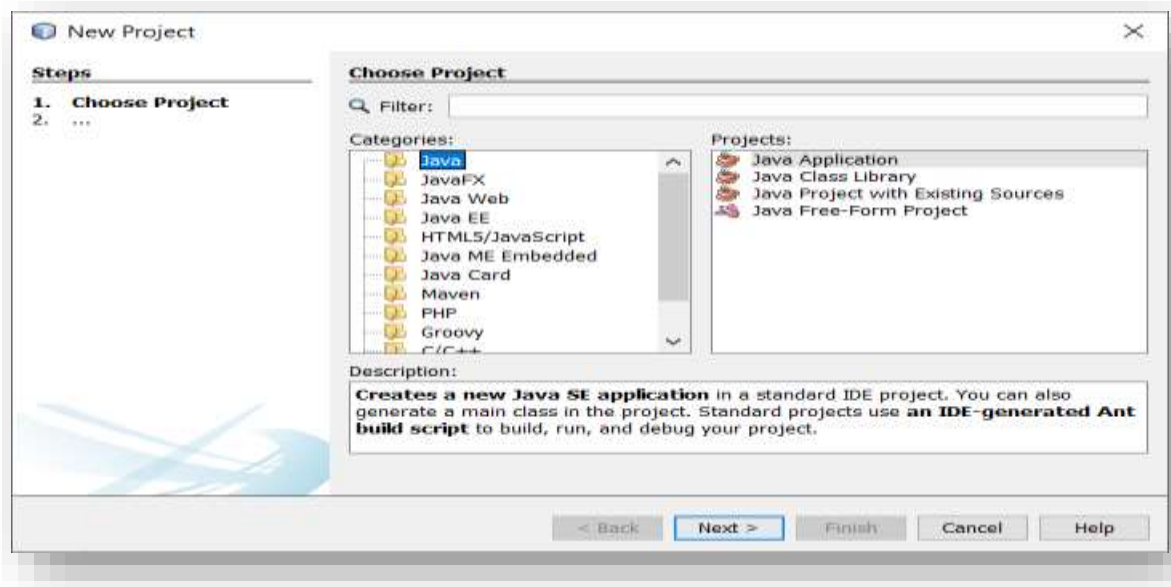
In this section we create an application that executes the GUI class. To accomplish the task, we will do the following:

- Create a frontend class.
- Include the frame's library in the current project.
- Import the frame class.
- Instantiate the frame.
- Run the program.

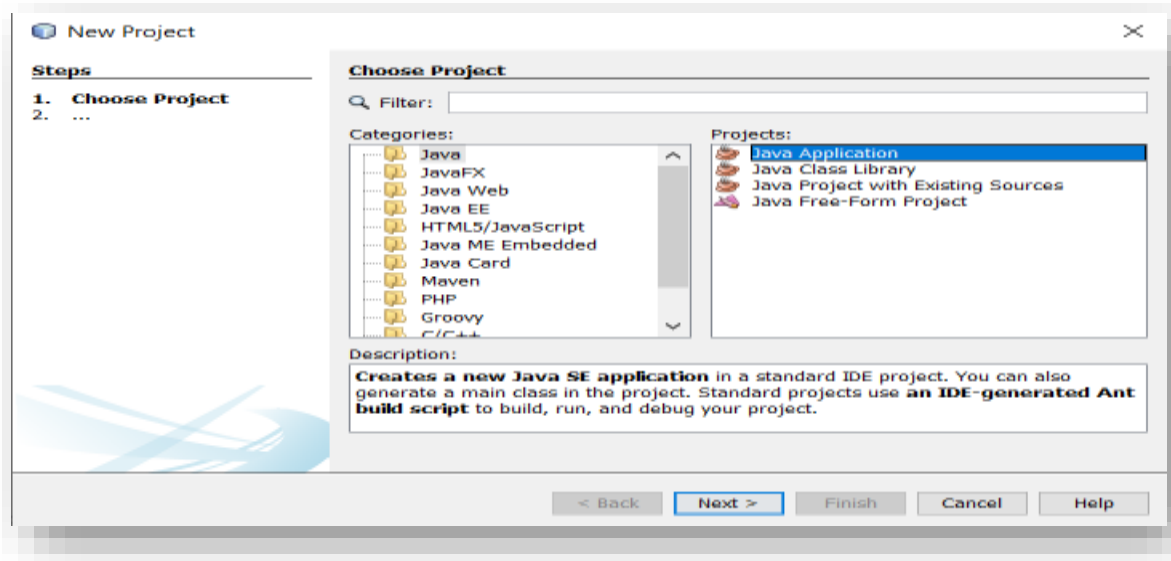
Step 1

Create a Java project application. To do this, perform the following tasks:

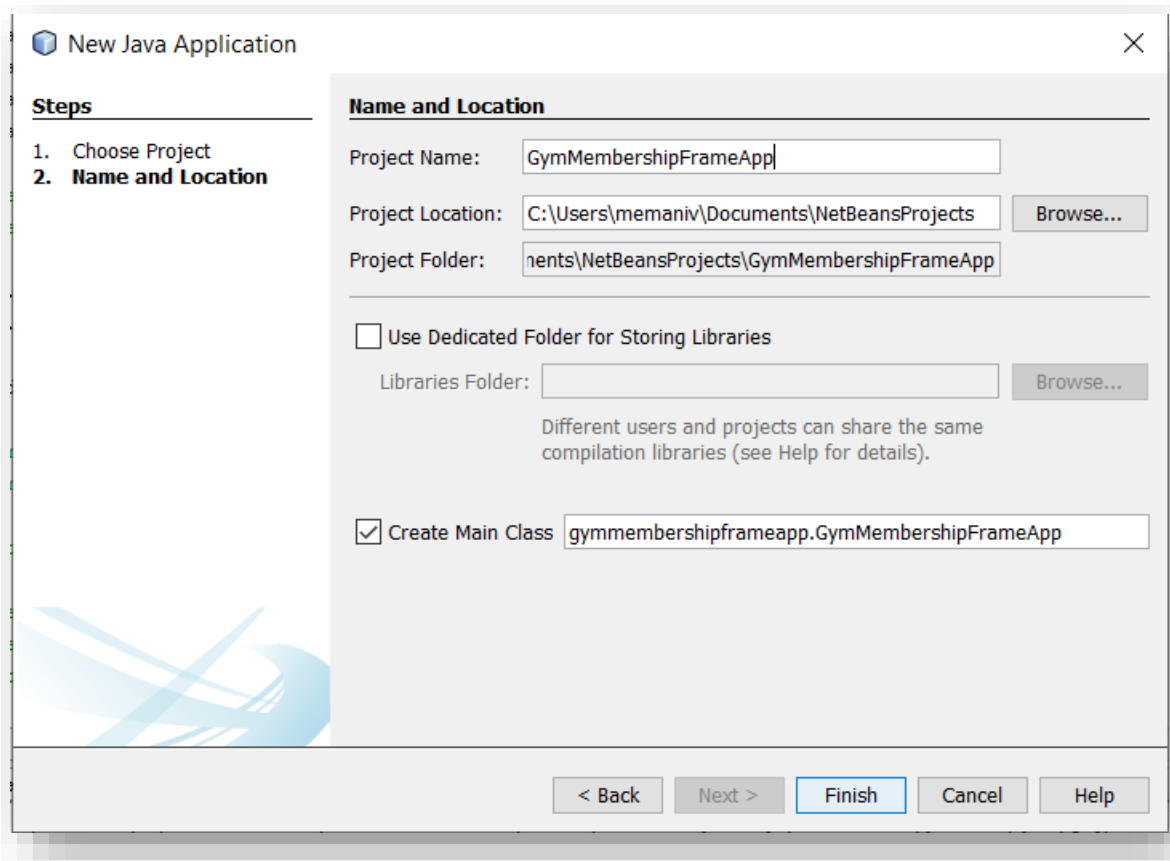
- Click on **File | New Project**.



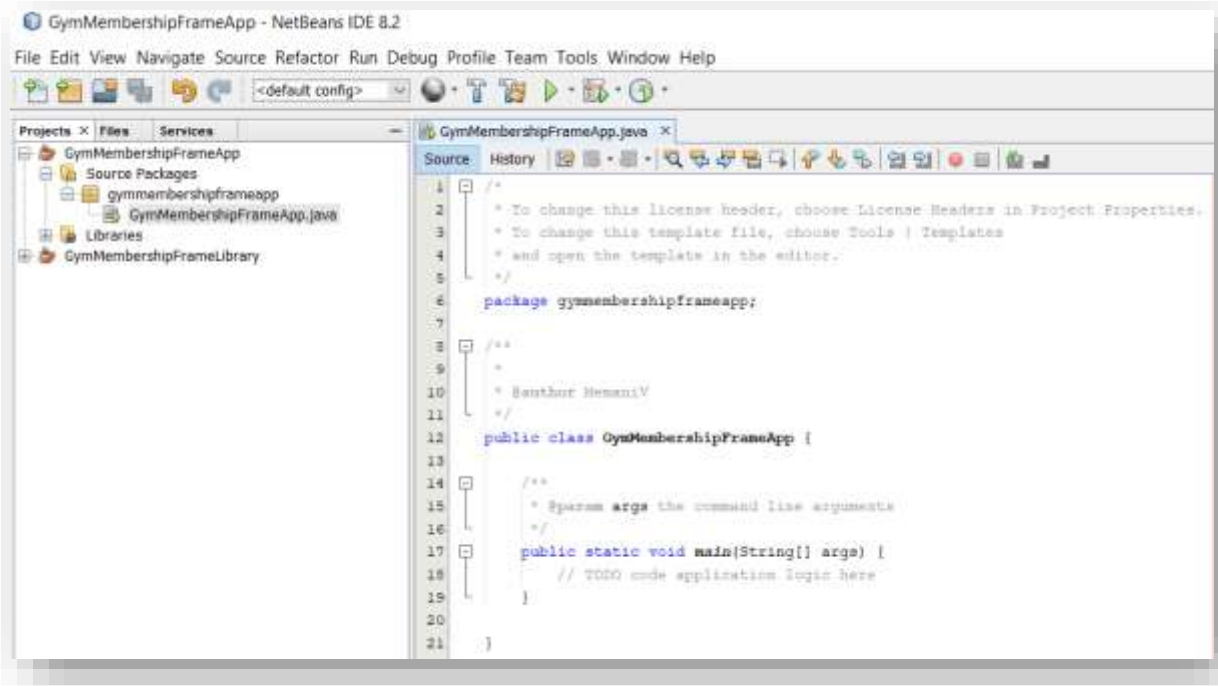
- Select **Java** under **Categories** and **Java Application** under **Projects**.



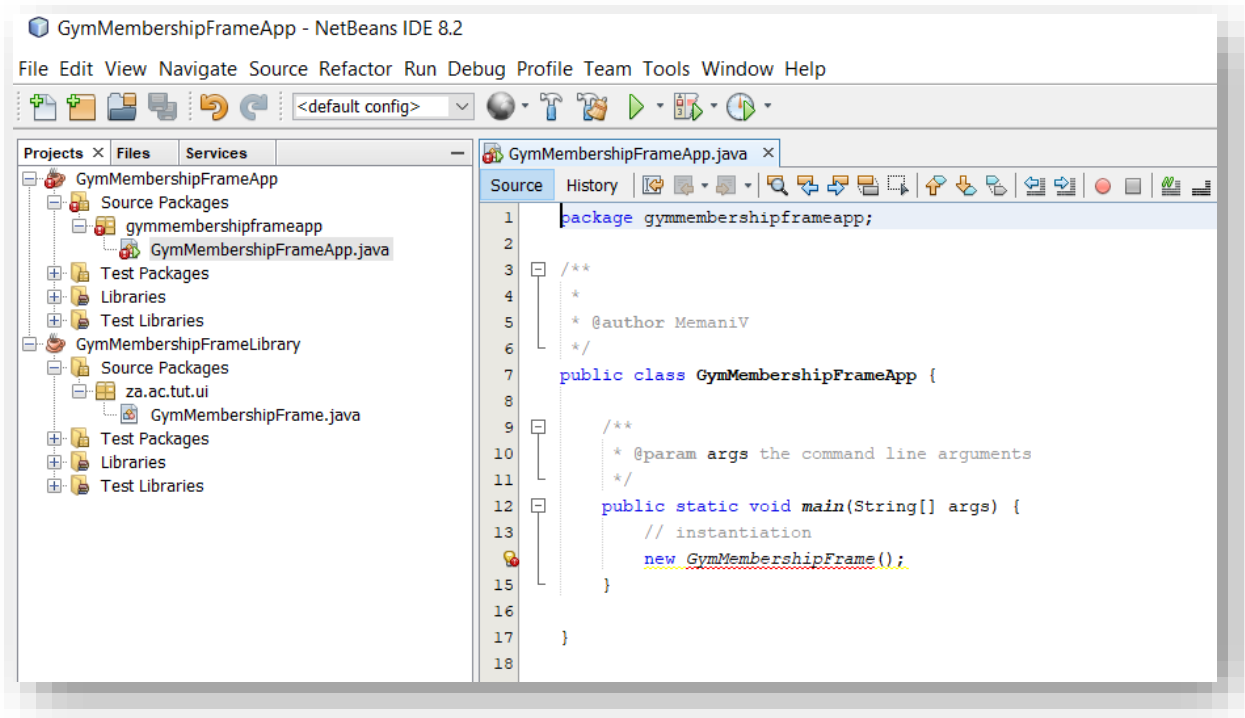
- Click **Next**. Name the project as **GymMembershipFrameApp**.



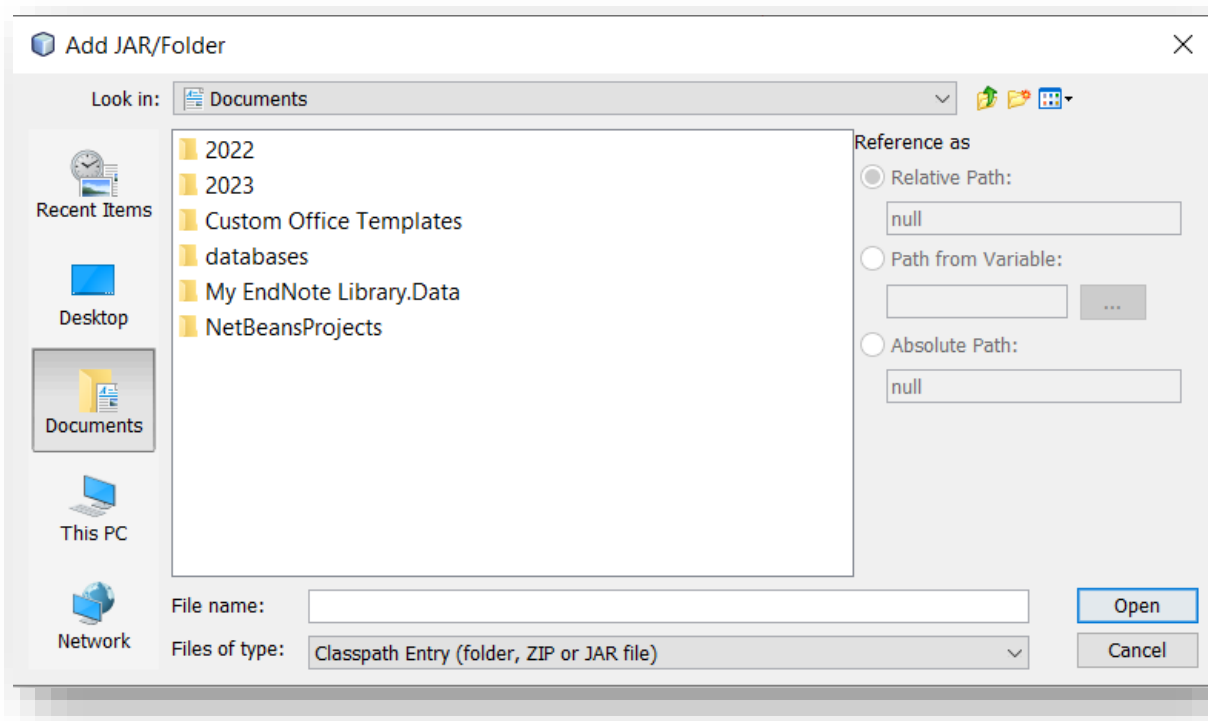
- Click **Finish**.



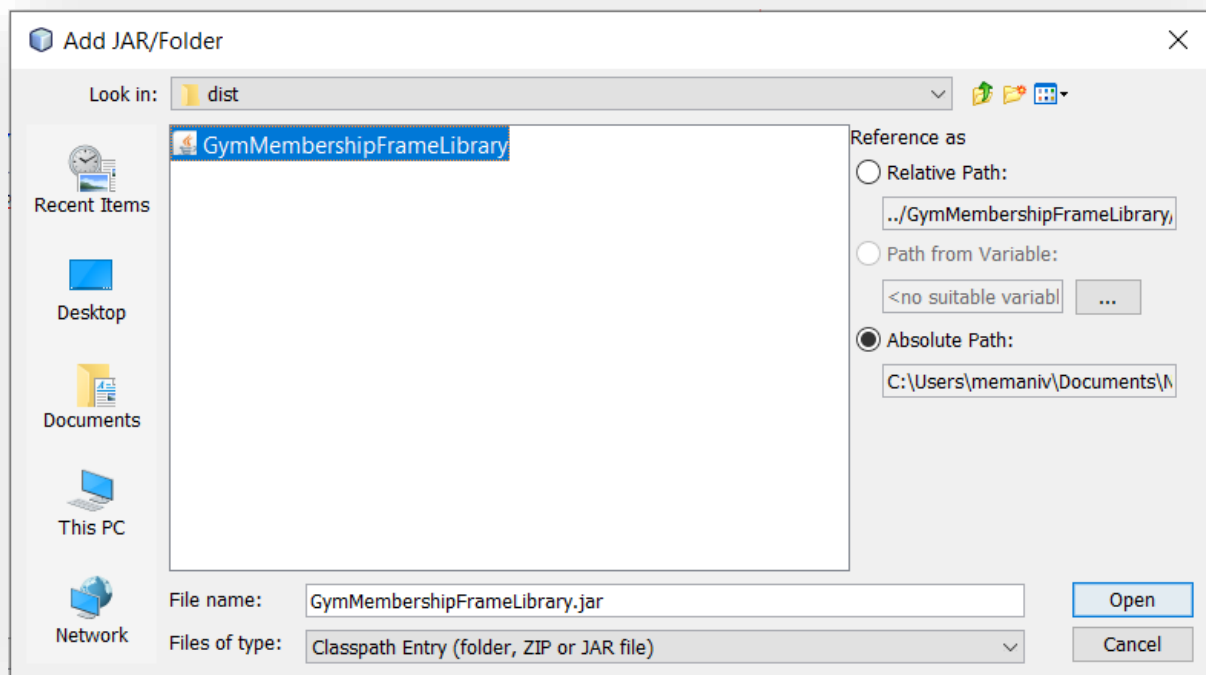
- Inside the main method, instantiate **GymMembershipFrame**.



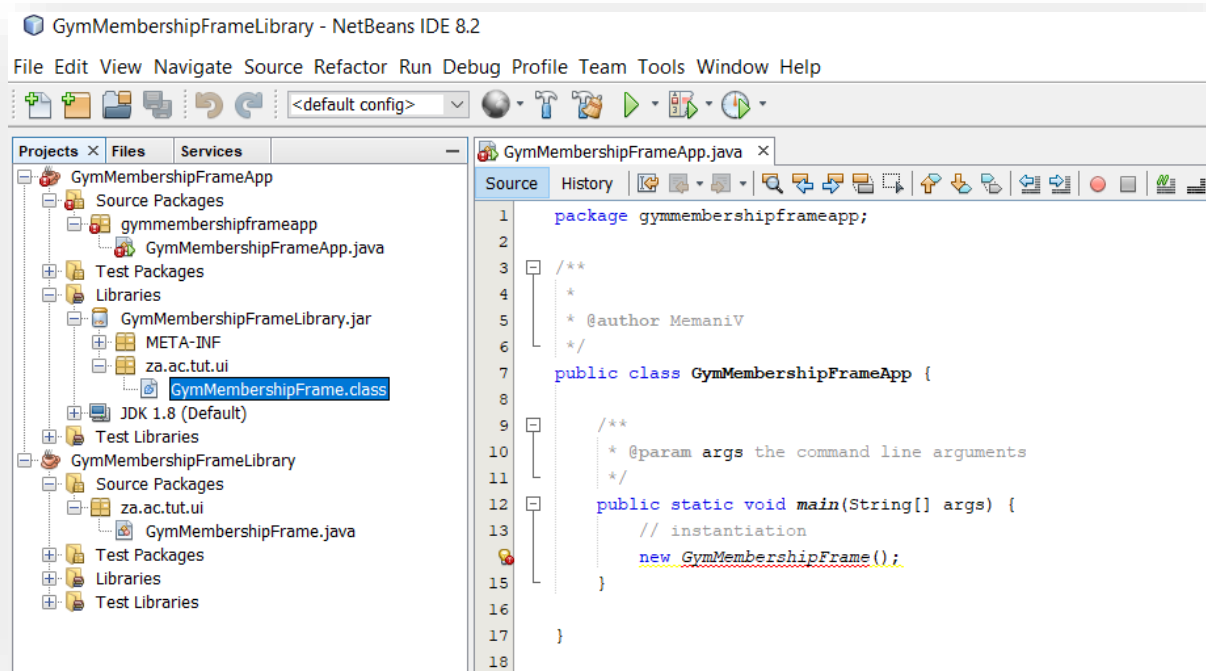
- There's an error. **GymMembershipFrame** is not visible inside the application. It is located in the jar file. Let's include the jar file library to the current project. Perform the following tasks:
 1. Right click on the **Libraries** folder and select **Add Jar/Folder**.



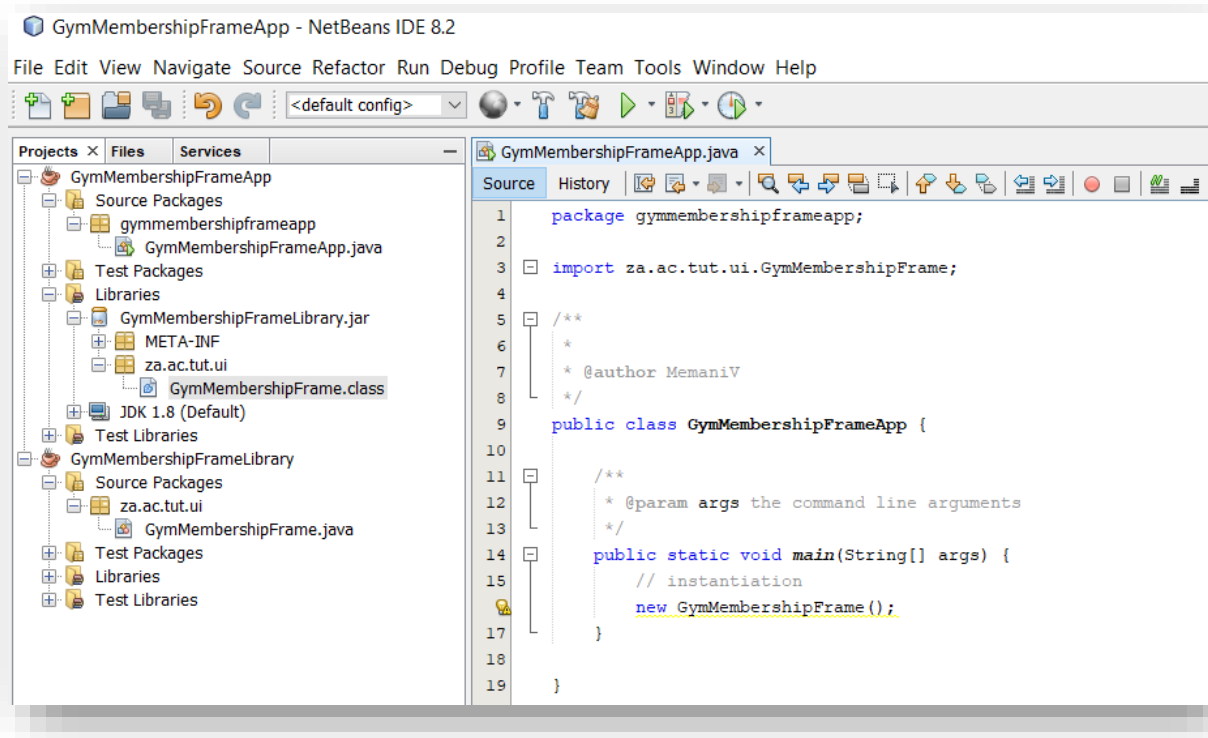
2. Navigate to the location of **GymMembershipFrameLibrary.jar** file.



3. Select the file and click on **Open**.

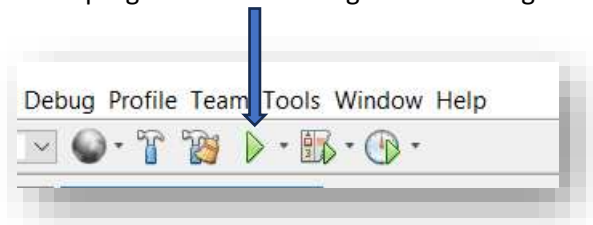


- Now the GymMembershipFrame library is included in the current project. Import the class, the error disappears.



Step 2

Run the program. Click on the green icon image.



Output:

The image shows a screenshot of a software window titled "Gym membership". Inside the window, there is a section titled "Membership Form" in blue text. The form is divided into three main sections:

- Client details:** This section contains four input fields: "Name:", "Surname:", "Id no:", and "Gender:". The "Gender:" field has a dropdown menu currently showing "Male".
- Contract options:** This section contains two lines of options. The first line is "Type of contract:" followed by three radio buttons labeled "Month-to-month", "Six months", and "Annual". The second line is "Select the checkbox if you need a personal trainer" followed by an unchecked checkbox.
- Comments:** This section is a large text area with a vertical scrollbar on the right side, currently empty.

At the bottom of the window, there is a button labeled "APPLY".

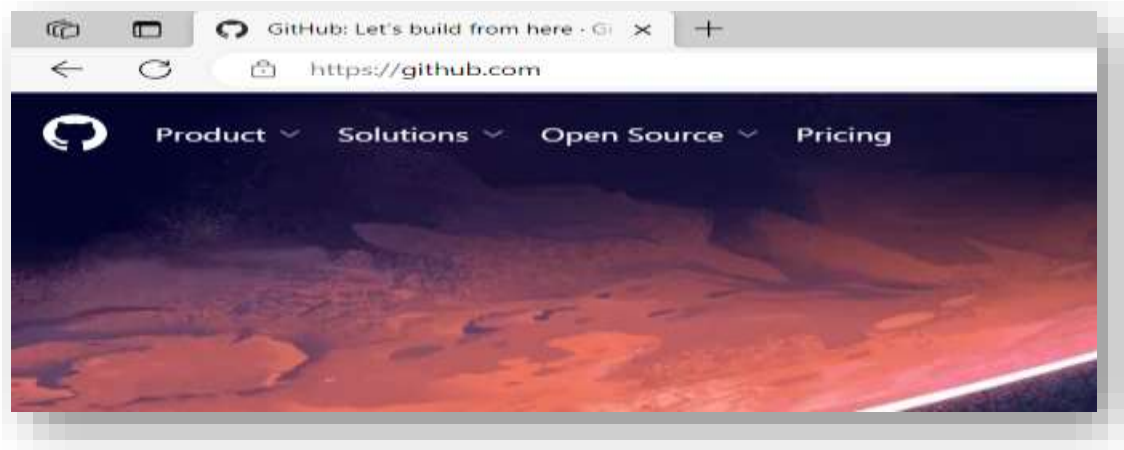
Part C: Store the source code on GitHub

In this section we are going to store the source code of **GymMembershipFrameLibrary.java** and **GymMembershipFrameApp.java** on GitHub. To accomplish this, we will perform the following:

- Get to the GitHub website.
- Create a repository for our project.
- Upload the two files (**GymMembershipFrameLibrary.java** and **GymMembershipFrameApp.java**) to the repository.
- Get a link of the repository. This we will share with anyone we want to access our repository.

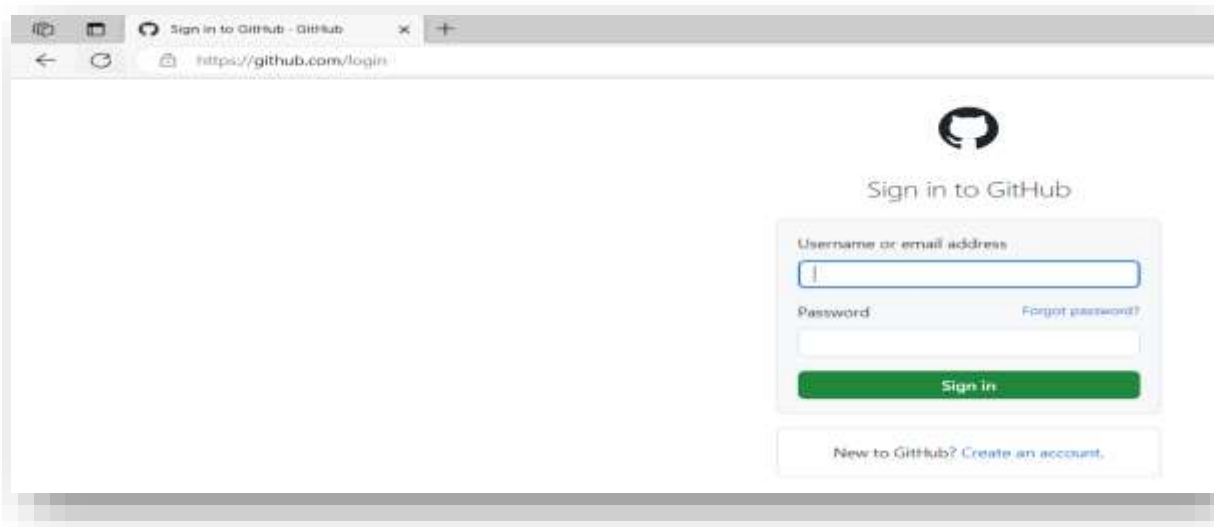
Step 1

Go to the GitHub website.

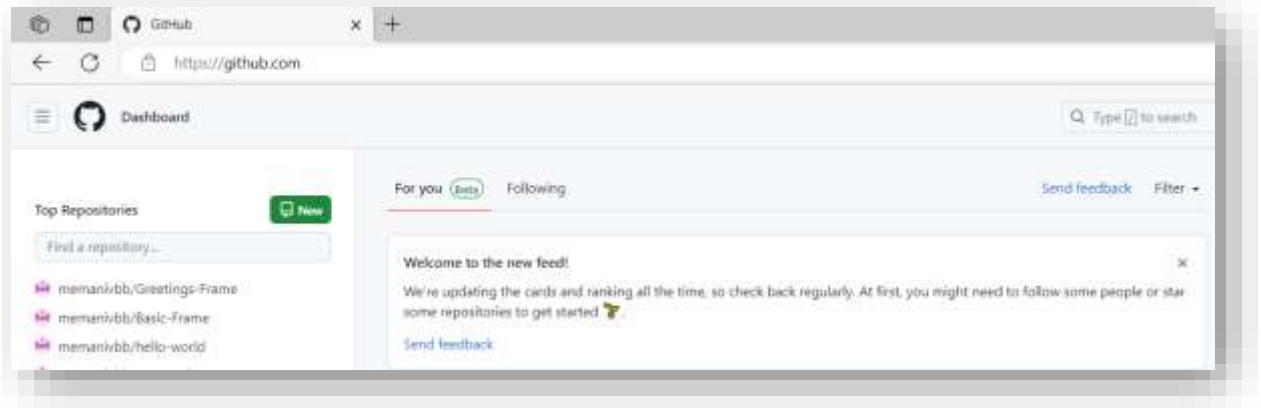


Step 2

On the far upper right of the page, click on **Sign in** if you have an account, or **Sign up** to create a new one.



After successful login you will see something along the lines:



Step 3

Click on **New** to create a new repository.

A screenshot of the 'Create a new repository' form on GitHub. The form has a title 'Create a new repository' and a subtitle 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. Below this, it says 'Required fields are marked with an asterisk (*)'. The form fields include: 'Owner *' with a dropdown menu showing 'memanivbb', 'Repository name *' with an empty text box, 'Description (optional)' with an empty text box, 'Public' (selected) and 'Private' (unselected) radio buttons, 'Initialize this repository with:' section with 'Add a README file' (unselected) checkbox, 'Add .gitignore' section with a dropdown menu showing '.gitignore template: None', and 'Choose a license' section. There are also links for 'vigilant-disco ?' and 'Learn more about READMEs'.

Step 4


Fill in the form.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 memanivbb ▾

Repository name *


GymMembershipFrameAp

✔ GymMembershipFrameApp is available.


Great repository names are short and memorable. Need inspiration? How about [fuzzy-octo-guide](#) ?

Description (optional)

In this project we create a frame that can be used to solicit a gym client's information.

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

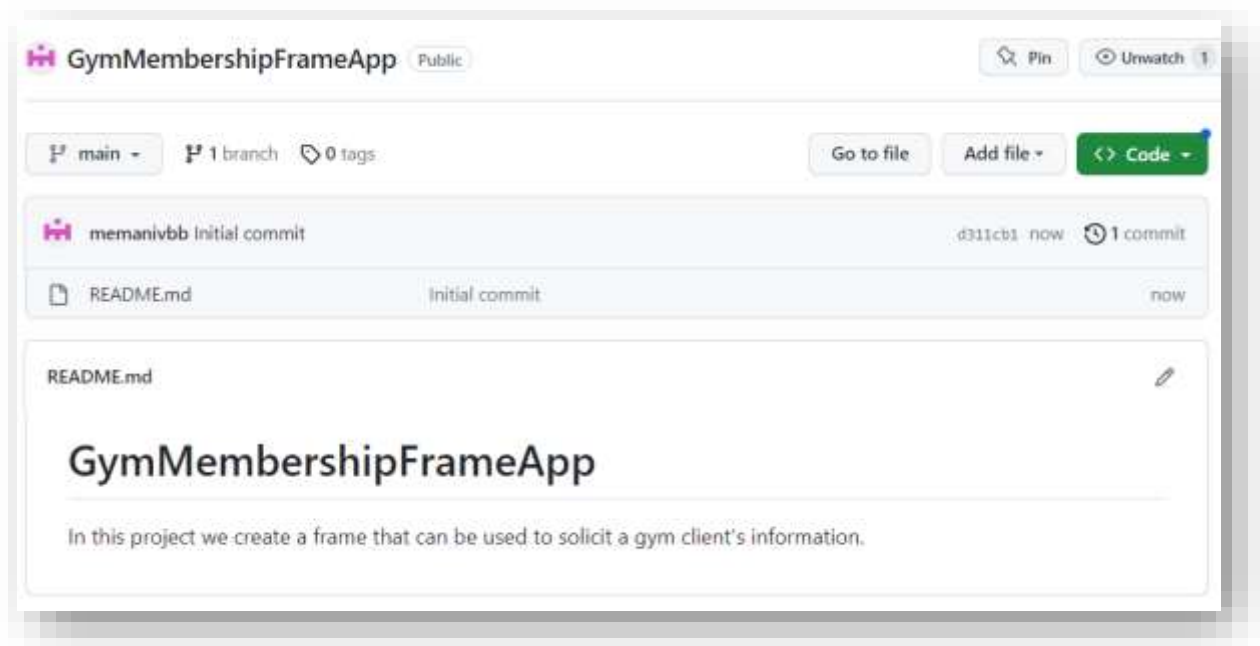
This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

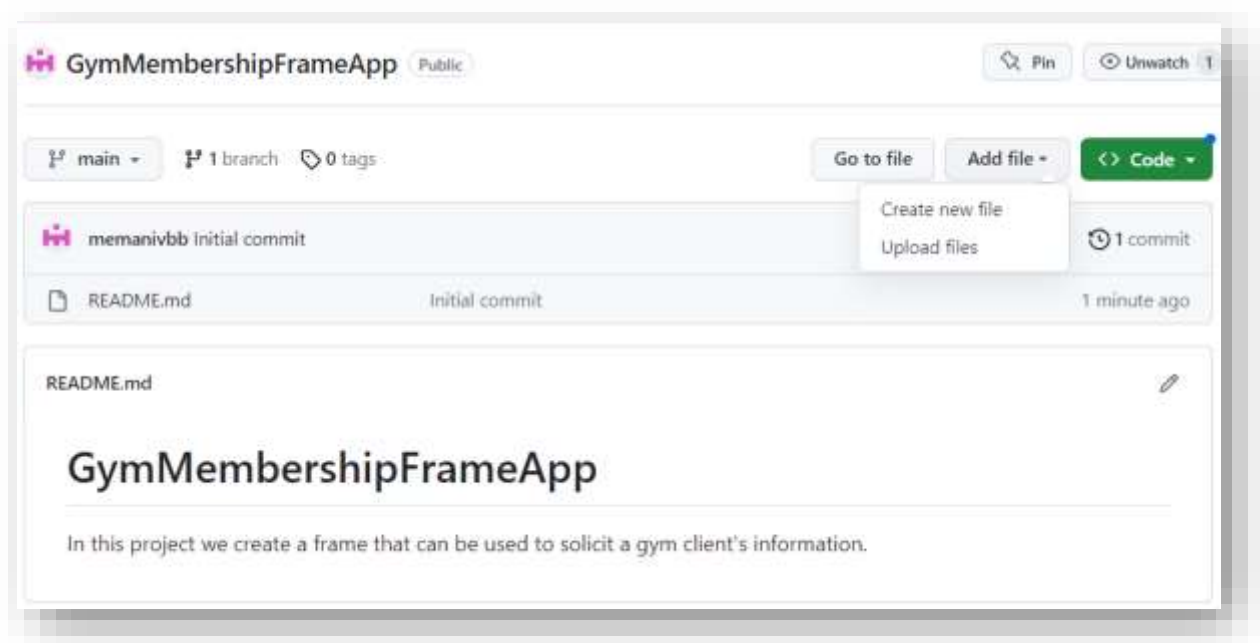
Step 5

Click on **Create repository**.



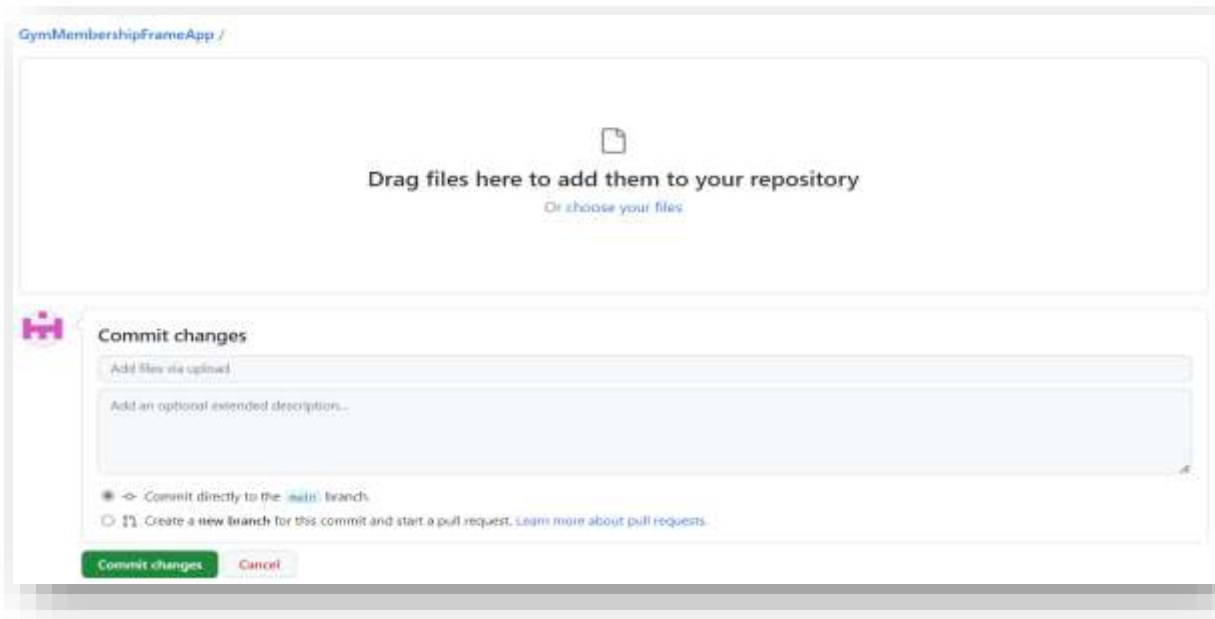
Step 6

Click on the **Add file** dropdown menu.



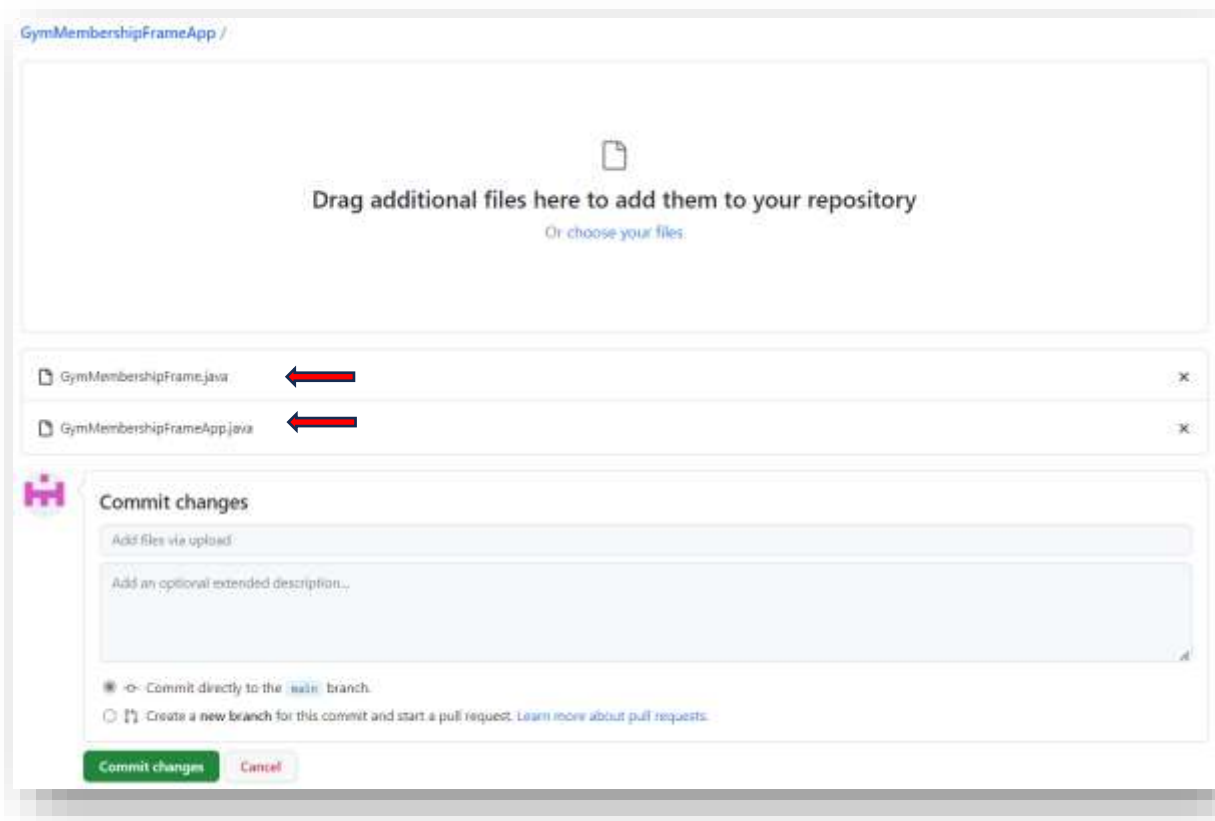
Step 7

Select the **Upload files** option.



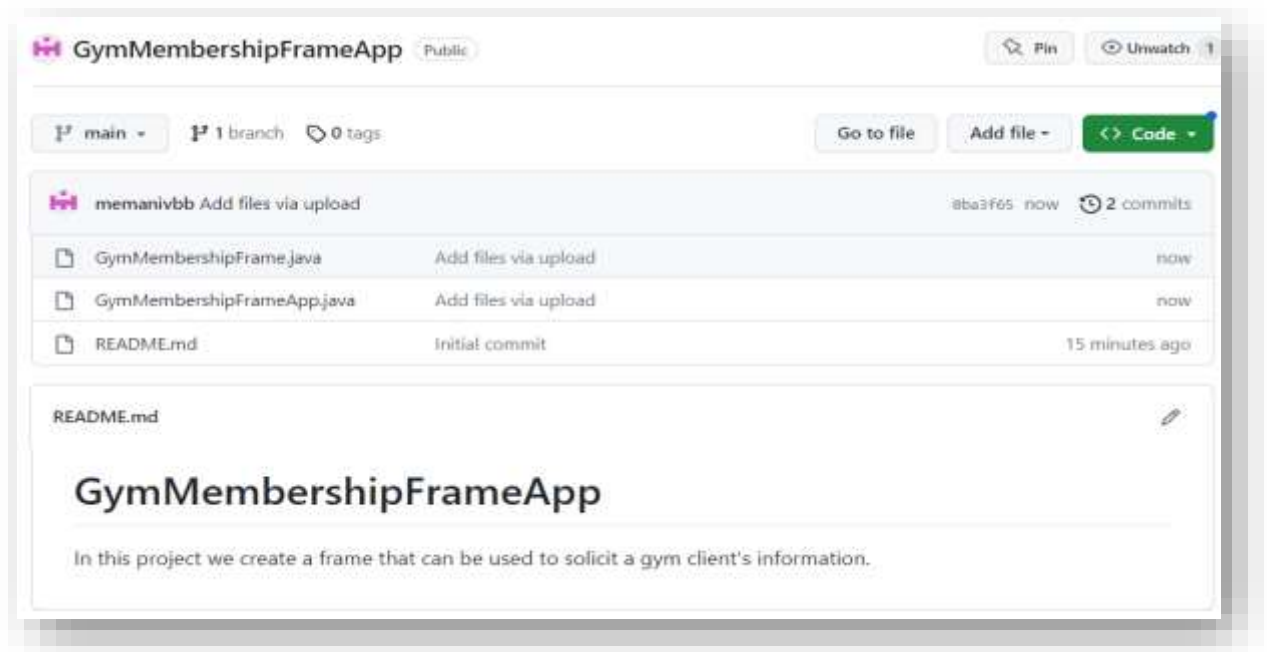
Step 8

Upload the files (**GreetingsGUI.java** and **GreetingsFrameApp.java**) by clicking on the link: **choose your files**.



Step 9

Click on the **Commit changes** button.



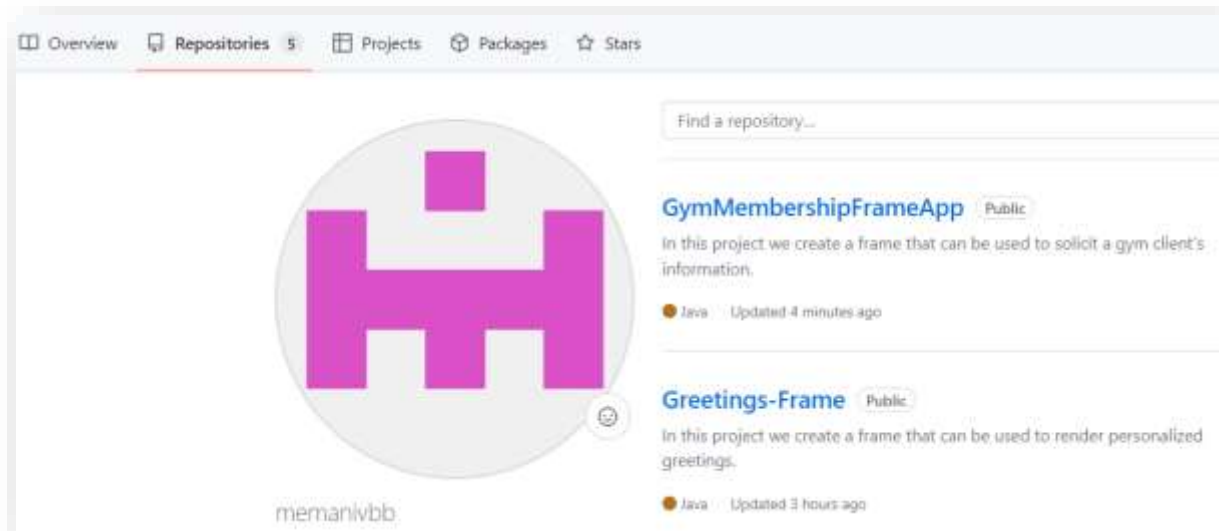
Step 10

Go back to the main page.



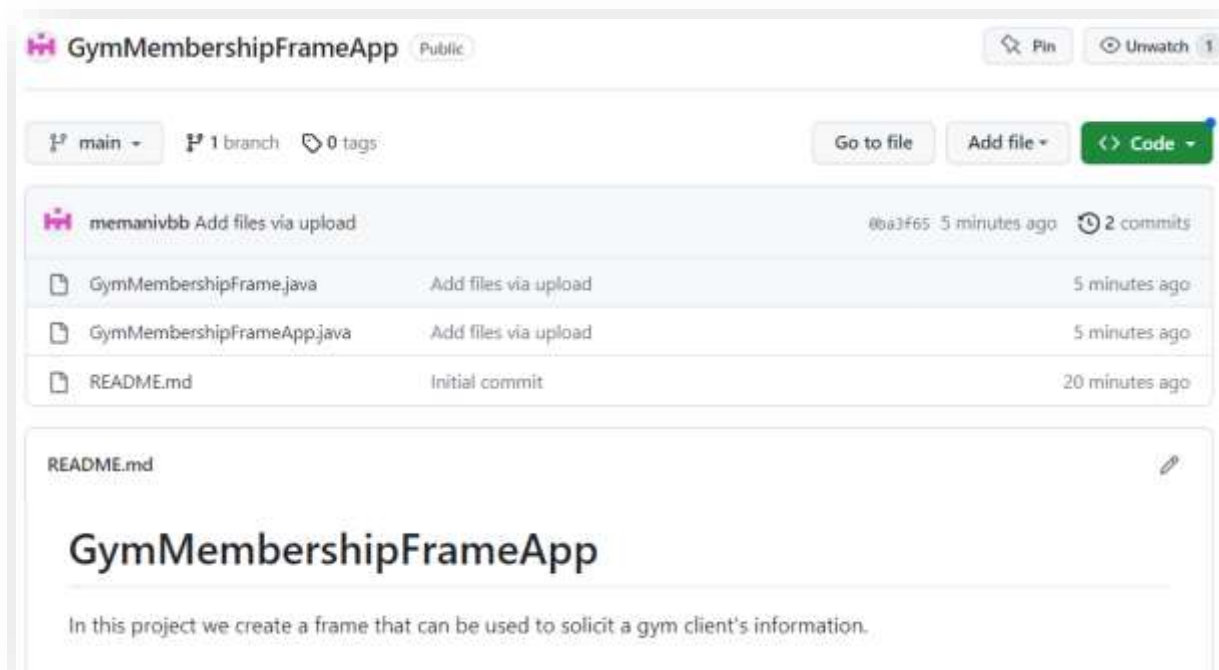
Step 11

Click on **Repositories**.



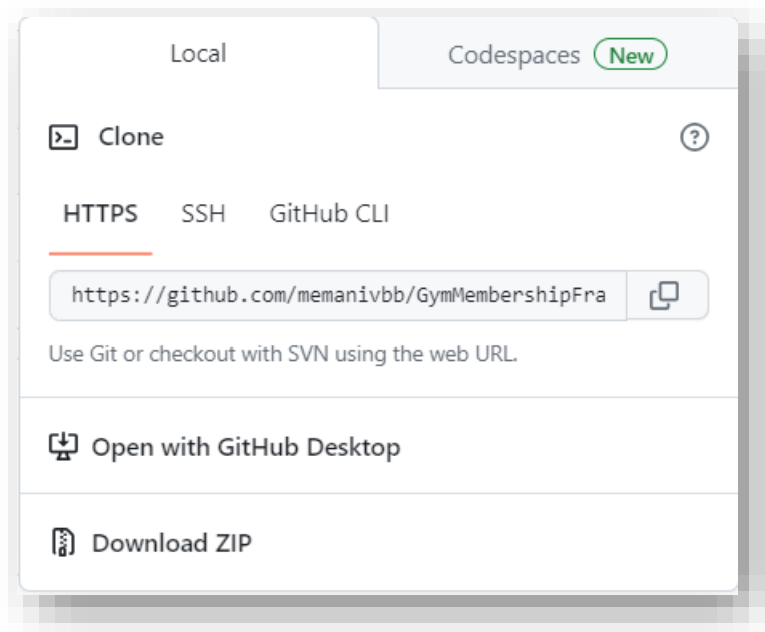
Step 12

Click on the **GymMembershipFrameApp** project link.



Step 13

Click on **Code** dropdown menu.



Step 14

Copy the repository's link by clicking on the corresponding icon . View the link (paste it here).

<https://github.com/memanivbb/GymMembershipFrameApp.git>

You can share this link with anyone you want to access your repository.

Conclusion

In this Mrabulo Session we managed to do three things, namely:

1. Created a frame in a backend class.
2. Tested the frame in a frontend class.
3. Uploaded our source code on GitHub.

Please do the DIY exercise below to test your knowledge. Thank you very much for having taken time to do the tutorial. Enjoy the rest of your day and God bless you.

DIY

Create an information requesting form similar to the own below.

Contact Information

Please enter your contact information

Do you have a South African Cell Phone Number? ---Please Select--- ▼ *

Work Telephone Number:

Home Telephone Number:

Email: *

Verify email: *

Residence Information

Please select whether you want to apply for residence

Do you want to apply for residence? --- Please select --- ▼ *

Disability Information

Please indicate whether you have any disabilities

Do you have a disability or impairment? ☐

The Cell Phone Number options are as follows:

Contact Information

Please enter your contact information

Do you have a South African Cell Phone Number? ---Please Select--- ▼ *

Work Telephone Number:

Home Telephone Number:

---Please Select---
Yes
No

The residence options are as follows:

Residence Information

Please select whether you want to apply for residence

Do you want to apply for residence? --- Please select --- ▼ *

--- Please select ---
Yes
No

Disability Information

Please indicate whether you have any disabilities