

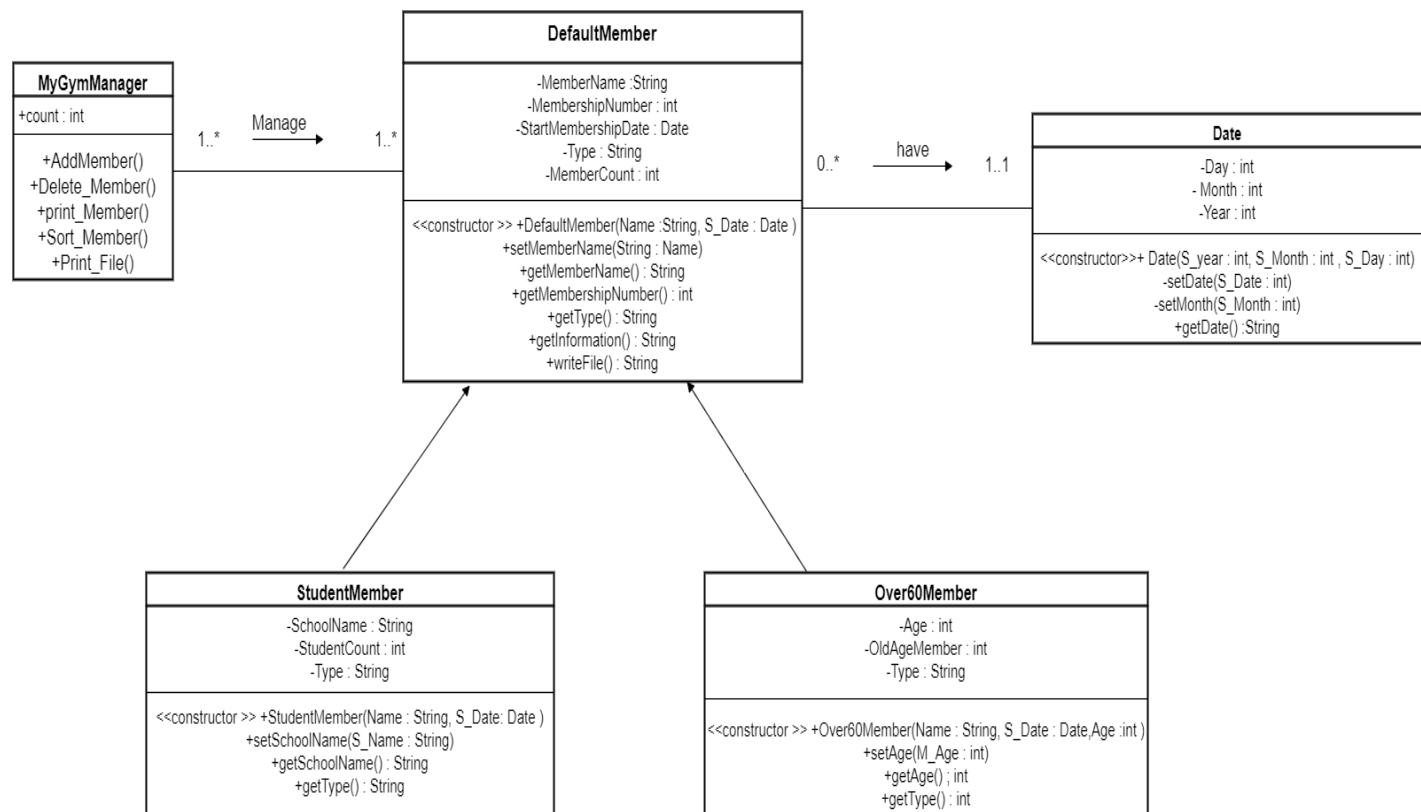
Report

Name = J.Thuwarakan

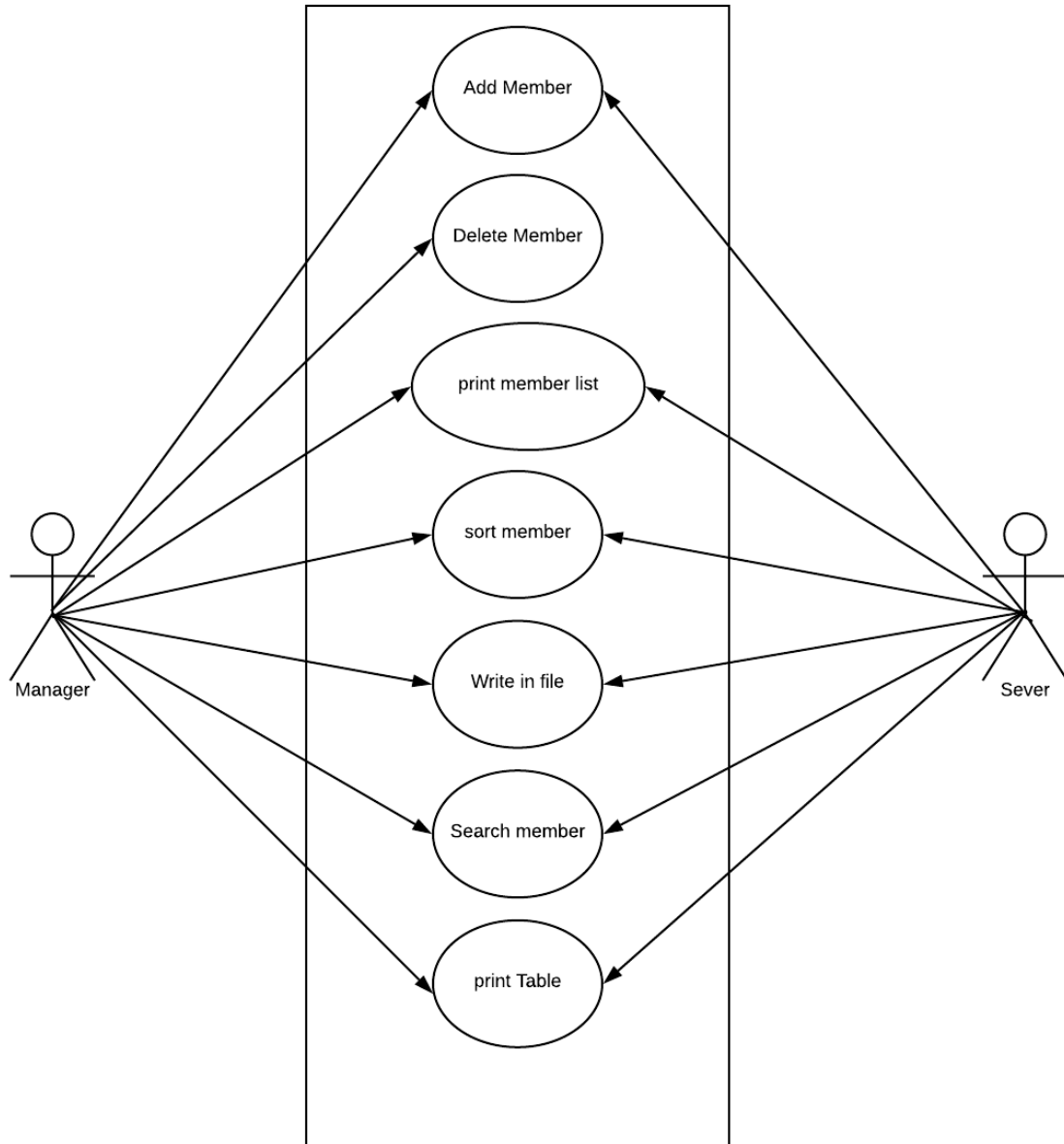
ID = 2019795

UoW ID = 1790265

Class Diagram



Use case diagram



Test case

TEST PLANE FOR JAVA COURSE WORK

Student name: J. Thuwarakan

student ID: 2019795

Test No	Test Inputs and instruction	Expected Result	Actual Result (State if not attempted)	Pass/Fail
1	Enter the method Number: 1 for select add member method	Display "Add Method" and say how many more members can be add	Display "Add Method" and say how many more members can be add	pass
2	Enter the method Number: 2 For select Delete method (if you add a member before)	Display "Delete Method" and ask deleting member id	Display "Delete Method" and ask deleting member id	pass
3	Enter the method Number: 2 For select Delete method (if you didn't add a member before)	Display "Your total member count is 0 First add some members"	Display "Your total member count is 0 First add some members"	Pass
4	Enter the method Number: 3 For select Print member method (if you add a member before)	Display a Member List	Display a Member List	pass
5	Enter the method Number: 3 For select Print member method (if you didn't add a member before)	Display "Your total member count is 0 First add some members"	Display "Your total member count is 0 First add some members"	pass
6	Enter the method Number: 4 For select Sort member method (if you add a member before)	Display a List of Sorted Member	Display a Sorted List of Member	pass
7	Enter the method Number: 4 For select Sort member method (if you didn't add a member before)	Display "Your total member count is 0 First add some members"	Display "Your total member count is 0 First add some members"	pass

8	Enter the method Number: 5 For select Write file method (if you didn't add a member before)	Display "Your total member count is 0 First add some members"	Display "Your total member count is 0 First add some members"	pass
9	Enter the method Number: 5 For select Write file method (if you add a member before)	Save the member details in file	Save the member details in file	pass
10	Enter the method Number: 6 For select Open UI method (if you add a member before)	Open the GUI	Open the GUI	pass
11	Enter an integer value above 6 or below 1 in the method number Ex: Enter the method Number: 8 Enter the method Number: -1	Display "You Enter a wrong number"	Display "You Enter a wrong number"	pass
12	Enter a string or a double value Ex: Enter method Number: thuwa Enter method Number: 1.1	Display "The System only accept 'Integer' Value"	Display "The System only accept 'Integer' Value"	pass
13	First select adding member method and select Number: 1 for adding Default member	Display the adding Default member method	Display the adding Default member method	pass
14	First select adding member method and select Number: 2 for adding Student member	Display the adding Student member method	Display the adding Student member method	Pass
15	First select adding member method and select Number: 3 for adding Over60 member	Display the adding Over60 member method	Display the adding Over60 member method	Pass
16	First select adding member method and Enter an integer value above 3 or below 1 in the method number Ex: Enter the method Number: 4 Enter the method Number: -1	Display" Wrong Number Please check your input"	Display" Wrong Number Please check your input"	pass

17	First Select adding Member method then Enter a string or a double value Ex: Enter method Number: thuwa Enter method Number: 1.1	Display "Invalid input"	Display "Invalid input"	pass
18	First add a default Member then go to delete member part and delete that member	System delete that member and display "you successfully deleted the Default Member"	System delete that member and display "you successfully deleted the Default Member"	pass
19	First add a Student Member then go to delete member part and delete that member	System delete that member and display "you successfully deleted the Student Member"	System delete that member and display "you successfully deleted the Student Member"	pass
20	First add an Over60 Member then go to delete member part and delete that member	System delete that member and display "you successfully deleted the Over60 Student Member"	System delete that member and display "you successfully deleted the Over60 Member"	pass
21	First add a member then go to the Delete member part and Enter an ID Number above 1000 and a wrong ID Number	Display "You input is not match with the values"	Display "You input is not match with the values"	pass
22	First add a member then go to the Delete member part and Enter an ID Number below 1000	Display "Invalid Membership Number"	Display "Invalid Membership Number"	pass
23	Add three members with the following names thuwarakan, mathuravan and jeyakumar after that select the sort member method	System first display the details of jeyakumar then print the details of mathuravan finally show the details of thuwarakan	System first display the details of jeyakumar then print the details of mathuravan finally show the details of thuwarakan	pass

24	First add some members then Enter 6 to open the GUI	System display a tab pane and one of the tab pane tab show a table of members	System display a tab pane and one of the tab pane tab show a table of members	pass
25	First add some members then Enter 6 to open the GUI and go to search pane and Enter a member ID number	Display the details of that member	Display the details of that member	pass
26	First add some members then Enter 6 to open the GUI and go to search pane and Enter a String value in the search bar	Display ""System only accept Integer number"	Display ""System only accept Integer number"	pass
27	First add some members then Enter 6 to open the GUI and go to search pane and Enter a unmatching number value in the search bar	Display "Invalid input"	Display "Invalid input"	pass

Code

Main class code

```
public static void main(String[] args) throws IOException {
    MyGymManager manager_1 = new MyGymManager();

    while (true) {
        System.out.println("\n===== Welcome
=====");
        System.out.println("\nYou can select which method you Want\nby
Entering Number\n ");
        System.out.println("1 = Adding members (this method will help you to
add members.)");
        System.out.println("2 = Deleting members (this method will help you
to Delete members by there MembershipNumber)");
        System.out.println("3 = Print member details (this method will print
a list of members with details.)");
        System.out.println("4 = Sort member (this method will help you to
sort members by there name.)");
        System.out.println("5 = Write File (this method will help you to
Write and Save member details in file.)");
        System.out.println("6 = Open UI (this method will help you to Search
members and display members list.)");
        try {
            Scanner input = new Scanner(System.in);
            System.out.print("\nEnter the method Number : ");
            int method_no = input.nextInt();

            switch (method_no) {
                case 1:
                    manager_1.Add_Member();
                    break;
                    //case 1 is finish
                case 2:
                    manager_1.Delete_Member();
                    break;
                    // case 2 is finish
                case 3:
                    manager_1.print_Member();
                    break;
                    // case 3 is finish
                case 4:
                    manager_1.Sort_Member();
                    break;
                    // case 4 is finish
                case 5:
                    manager_1.Print_File();
                    break;
                    //case 5 is finish
                case 6:
                    launch(args);
                    break;
                    //case 6 is finish
                default:
            }
        }
    }
}
```

```

        System.out.println("\nYou Enter a wrong number ");

    }

    } catch (RuntimeException e) {
        System.out.println("\nThe System only accept 'Integer' Value");
    }
}

```

Default Member class

```

public class DefaultMember {
    private String MemberName;
    private static int MemberCount;
    private int MembershipNumber;
    private Date StartMembershipDate;
    private static String Type;

    public DefaultMember(String Name, Date S_Date ) {
        super();
        setMemberName(Name);
        StartMembershipDate = S_Date;
        MemberCount=MemberCount+1;
        MembershipNumber = 1000 + MemberCount;
    }

    public void setMemberName(String Name) {
        MemberName = Name;
    }

    public String getMemberName(){
        return MemberName;
    }

    public int getMembershipNumber(){
        return MembershipNumber;
    }

    public String getType(){
        Type = "Default Member";
        return Type;
    }

    public String getInformation(){
        String info = String.format("%6s %18s %15s %18s",MembershipNumber,
MemberName, StartMembershipDate.getDate(), getType());
        return info;
    }

    public String writeFile(){
        return "Name = "+MemberName+"\nMembershipNumber =
"+MembershipNumber+"\nStart Date = "+StartMembershipDate.getDate()
        +"\nType = "+getType();
    }
}

```



```
    }  
}
```

Student Class code

```
public class StudentMember extends DefaultMember {  
    private String SchoolName;  
    private static int StudentCount;  
    private static String Type;  
  
    public StudentMember(String Name ,Date S_Date) {  
        super(Name, S_Date);  
        StudentCount = StudentCount +1;  
    }  
  
    public void setSchoolName(String S_Name){  
        SchoolName = S_Name;  
    }  
  
    public String getSchoolName(){  
        return SchoolName;  
    }  
  
    public String getType(){  
        Type = "Student Member";  
        return Type;  
    }  
}
```

over60 member

```
public class Over60Member extends DefaultMember {  
    private int Age;  
    private static int OldAgeMember;  
    private static String Type;  
  
    public Over60Member(String Name, Date S_Date, int M_Age) {  
        super(Name, S_Date);  
        setAge(M_Age);  
        OldAgeMember += 1;  
    }  
  
    public void setAge(int M_Age){  
        if(M_Age > 60 && M_Age <100){  
            Age = M_Age;  
        }  
        else {  
            System.out.print("Age is wrong");  
        }  
    }  
}
```

```

    public int getAge() {
        return Age;
    }

    public String getType() {
        Type = "Over60 Member";
        return Type;
    }
}

```

Date class code

```

public class Date {
    private int Day;
    private int Month;
    private int Year;

    public Date(int S_year, int S_Month, int S_Day) {
        setYear(S_year);
        setMonth(S_Month);
        setDay(S_Day);
    }

    private void setYear(int S_Year) {
        if(S_Year>2000 && S_Year<2030 ) {
            Year = S_Year;
        }
        else{
            System.out.println("\nYou enter a wrong year");
        }
    }

    private void setMonth(int S_Month) {
        if (S_Month > 0 && S_Month <= 12) {
            Month = S_Month;
        }
        else {
            System.out.println("\nInvalid Month");
        }
    }

    private void setDay(int S_Day) {
        int DaysinMonth[] = {31,29,31,30,31,30,31,31,30,31,30,31};
        if(S_Day <= DaysinMonth[Month-1]) {
            if(Month==2 && S_Day == 29) {
                if(Year%4 == 0) {
                    Day = S_Day;
                }
                else{
                    System.out.println("\n"+Year+" is not a leap year");
                }
            }
            else{

```

```

        Day = S_Day;
    }
}
else{

    System.out.println("\n Invalid Day");
}
}

public String getDate(){
    String Date = Day+"/"+Month+"/"+Year;
    return Date;
}
}

```

Gym manager interface code

```

public interface GymManager {

    public void Add_Member();
    public void Delete_Member();
    public void print_Member();
    public void Sort_Member();
    public void Print_File() throws IOException;
    public String Search_Member(int Number);

    ArrayList<DefaultMember> ListofMembers = new ArrayList<>();
    ArrayList<DefaultMember> ListofMembers_2 = new ArrayList<>();
    int count = 0;

}

```

My gym manager class code

Code for Add member method

```

public void Add_Member() {
    try {
        Scanner get =new Scanner(System.in);
        if (count <100) {
            count = ListofMembers.size();
            int total = 100 - count;
            System.out.println("\n" + total + "more members can be add ");
            System.out.println("----select member type for adding member----
\n");

            System.out.println("Enter 1 for Default member");
            System.out.println("Enter 2 for Student member");
            System.out.println("Enter 3 for Over60 member");

```

```

System.out.print("\nNumber : ");
int member_selector = get.nextInt();

if (member_selector == 1) {
    System.out.print("Enter member Name : ");
    get.nextLine();
    String obj_name = get.nextLine();
    System.out.println("\nEnter StartMembershipDate (all inputs
in numbers)");
    System.out.print("Enter Start Year : ");
    int year = get.nextInt();
    System.out.print("Enter Start Month : ");
    int month = get.nextInt();
    System.out.print("Enter Start Day : ");
    int day = get.nextInt();

    Date date_start = new Date(year, month, day);
    DefaultMember member = new
DefaultMember(obj_name.toLowerCase(), date_start);
    ListofMembers.add(member);
    ListofMembers_2.add(member);
    System.out.println("You Successfully added the member");

} else if (member_selector == 2) {
    System.out.print("Enter member Name : ");
    get.nextLine();
    String obj_name = get.nextLine();
    System.out.println("\nEnter StartMembershipDate (all inputs
in numbers)");
    System.out.print("Enter Start Year : ");
    int year = get.nextInt();
    System.out.print("Enter Start Month : ");
    int month = get.nextInt();
    System.out.print("Enter Start Day : ");
    int day = get.nextInt();
    System.out.print("Enter member school name :");
    get.nextLine();
    String obj_school = get.nextLine();

    Date date_start = new Date(year, month, day);
    StudentMember member = new
StudentMember(obj_name.toLowerCase(), date_start);
    member.setSchoolName(obj_school.toLowerCase());
    ListofMembers.add(member);
    ListofMembers_2.add(member);
    System.out.println("You Successfully added the member");

} else if (member_selector == 3) {
    System.out.print("Enter member Name : ");
    get.nextLine();
    String obj_name = get.nextLine();
    System.out.println("\nEnter StartMembershipDate (all inputs
in numbers)");
    System.out.print("Enter Start Year : ");
    int year = get.nextInt();
    System.out.print("Enter Start Month : ");

```

```

        int month = get.nextInt();
        System.out.print("Enter Start Day : ");
        int day = get.nextInt();
        System.out.print("Enter member Age(in numbers) :");
        int obj_age = get.nextInt();

        Date date_start = new Date(year, month, day);
        Over60Member member = new
Over60Member(obj_name.toLowerCase(), date_start, obj_age);
        ListofMembers.add(member);
        ListofMembers_2.add(member);
        System.out.println("You Successfully added the member");

    } else {
        System.out.println("Wrong Number Please check your input");
    }

    } else {
        System.out.println("system can store max 100 members");
    }
}
catch (RuntimeException e){
    System.out.println("\nInvalid input");
}
}
}

```

code for delete member method

```

try{
    if(ListofMembers.size() != 0) {
        Scanner get = new Scanner(System.in);
        System.out.print("\nEnter deleting member MembershipNumber : ");
        int number = get.nextInt();

        if (number > 1000) {
            for (int i = 0; i < ListofMembers.size(); i++) {
                int check = ListofMembers.get(i).getMembershipNumber();
                if (number == check) {
                    System.out.println("\nyou successfully deleted the "
+ ListofMembers.get(i).getType());
                    ListofMembers.remove(i);
                    ListofMembers_2.remove(i);
                    break;
                } else {
                    System.out.println("\nYou input is not match with the
values");
                }
            }
        } else {
            System.out.println("\nInvalid Membership Number");
        }
    }
    else{
        System.out.println("\nYour total member count is 0 \nFirst add

```

```

some members");
    }
}
catch(RuntimeException e){
    System.out.println("\nInvalid input");
}
}

```

code for print member list method

```

public void print_Member() {
    if (ListofMembers.size() != 0) {
        System.out.println("\n Member No      Member Name      date
Type\n");
        for (int i = 0; i < ListofMembers.size(); i++) {
            System.out.println(ListofMembers.get(i).getInformation());
        }
    } else {
        System.out.println("\nYour total member count is 0 \nFirst add
some members");
    }
}
}

```

code for sort member method

```

if(ListofMembers_2.size() != 0) {
    int length = ListofMembers_2.size();
    for(int i=0; i <length-1;i++) {
        for (int j = 0; j < length - (i + 1); j++) {
            if
(ListofMembers_2.get(j).getMemberName().compareTo(ListofMembers_2.get(j +
1).getMemberName()) > 0) {
                DefaultMember part =ListofMembers_2.get(j);
                ListofMembers_2.set(j,ListofMembers_2.get(j+1));
                ListofMembers_2.set(j+1,part);
            }
        }
    }
    System.out.println("\n Member No      Member Name      date
Type\n");
    for (int i = 0; i < ListofMembers_2.size(); i++) {
        System.out.println(ListofMembers_2.get(i).getInformation());
    }
}
else{
    System.out.println("Your total member count is 0 \nFirst add some
members");
}
}
}

```

code for write file method

```
public void Print_File() throws IOException {
    if(ListofMembers.size() != 0) {
        //File myfile = new File("C:\\Users\\thuwarakan\\Documents\\pp2-
file\\memberFile.txt");
        File myfile = new File("memberFile.txt");
        FileWriter wrt = null;
        PrintWriter prt = null;
        try {
            wrt = new FileWriter(myfile, true);
            prt = new PrintWriter(wrt, true);

            for(int i = 0; i < ListofMembers.size();i++){
                prt.println("\n"+ListofMembers.get(i).writeFile());
            }
        } catch (Exception e) {
            System.out.println("Something wrong");
        } finally {
            wrt.close();
            prt.close();
        }
    }
    else{
        System.out.println("Your total member is 0 \nFirst add some
members");
    }
}
```

code for search member method

```
public String Search_Member(int Number){
    String set = null;
    for(int i = 0; i < ListofMembers.size(); i++){
        if(Number == ListofMembers.get(i).getMembershipNumber()){
            set = ListofMembers.get(i).writeFile();
            break;
        }
        else{
            set = "Invalid Number";
        }
    }
    return set;
}
```

code for print table view method

```
public static ObservableList printTable() {
    ObservableList<DefaultMember> tableList =
FXCollections.observableArrayList();
    for(DefaultMember member : ListofMembers) {
        tableList.add(member);
    }
}
```

```

    }
    return tableList;
}

```

code for creating a UI

```

ManagerUI.setTitle("UI for Manager");
TabPane UIforManager = new TabPane();

Tab List_of_members = new Tab("Search Members");

TableView table = new TableView();

TableColumn<DefaultMember, Integer> column1 = new TableColumn<>("Member
Id");
column1.setMinWidth(100);
column1.setCellValueFactory(new
PropertyValueFactory<>("MembershipNumber"));

TableColumn<DefaultMember, String> column2 = new TableColumn<>("Name");
column2.setMinWidth(150);
column2.setCellValueFactory(new PropertyValueFactory<>("MemberName"));

TableColumn<DefaultMember, String> column3 = new TableColumn<>("Type");
column3.setMinWidth(150);
column3.setCellValueFactory(new PropertyValueFactory<>("Type"));

table.getColumns().addAll(column1, column2, column3);
table.setItems(printTable());

VBox vbox = new VBox(table);
Pane base_1 = new Pane(vbox);
List_of_members.setContent(base_1);
//-----
Tab Search_member = new Tab("Search Members");

Label label_1 = new Label();
label_1.setText("Welcome!\nYou can search members from here.\nBy members
membership number ");
label_1.setLayoutY(20);
label_1.setLayoutX(15);
label_1.setStyle("-fx-text-fill: black;-fx-font:bold 14px 'arialblack'");

TextField search = new TextField();
search.setLayoutX(45);
search.setLayoutY(100);
search.setStyle("-fx-font:bold 18px 'arialblack';-fx-background-
radius:22px");

Button get = new Button("Search");
get.setLayoutX(300);

```



```

        get.setLayoutY(98);
        get.setStyle("-fx-background-color:darkslateblue;-fx-text-fill: white;-fx-background-radius:28px;" +
            "-fx-padding:8px;-fx-font:bold 16px 'arialblack'");

        Label label_2 = new Label();
        label_2.setLayoutY(175);
        label_2.setLayoutX(80);
        label_2.setStyle("-fx-text-fill: black;-fx-font:bold 14px 'arialblack'");

        Pane base_2 = new Pane();
        base_2.setStyle("-fx-background-color:lightblue;");
        base_2.getChildren().addAll(label_1, search, get, label_2);
        Search_member.setContent(base_2);

        get.setOnAction(new EventHandler<ActionEvent>() {

            UIManager.setTabClosingPolicy(TabPane.TabClosingPolicy.UNAVAILABLE);
            UIManager.getTabs().addAll(List_of_members, Search_member);
            ManagerUI.setScene(new Scene(UIManager, 400, 400));

            ManagerUI.show();

```

Code for search button

```

MyGymManager manage = new MyGymManager();
    try {
        int number = Integer.parseInt(search.getText());
        if (number > 1000) {
            label_2.setText(manage.Search_Member(number));
        } else {
            label_2.setText("Invalid Number");
        }
    } catch (RuntimeException e) {
        label_2.setText("System only accept Integer number");
    }
}

```

Unite testing codes:

```

Date class testing
public class DateTest {
    final static Date date_4 = new Date(2015,5,23);

    @Test
    public void getDate_testing() {
        String output = "23/5/2015";
        assertEquals(output, date_4.getDate());
    }
}

```

```

default member class testing
public class DefaultMemberTest {
    final static Date date_1 = new Date(2012,12,11);
    final static DefaultMember member_for_test = new
DefaultMember("Thuwarakan",date_1);

    @Test
    public void setMemberName_testing() {

        String new_name = "jeyakumar";
        member_for_test.setMemberName(new_name);
        assertEquals(new_name , member_for_test.getMemberName());

    }

    @Test
    public void getMemberName_testing() {
        assertEquals("Thuwarakan" , member_for_test.getMemberName());
    }

    @Test
    public void getMembershipNumber_testing() {
        assertEquals(1001, member_for_test.getMembershipNumber());
    }

    @Test
    public void getType_testing() {
        assertEquals("Default Member",member_for_test.getType());
    }

    @Test
    public void getInformation_testing() {
        String new_test = String.format("%6s %18s %15s
%18s","1001","Thuwarakan","11/12/2012","Default Member");
        assertEquals( new_test, member_for_test.getInformation() );

    }

    @Test
    public void writeFile_testing() {
        String test_file = "Name = "+"Thuwarakan"+"\\nMembershipNumber =
"+"1001"+"\\nStart Date = "+"11/12/2012"
        +"\\nType = "+"Default Member";
        assertEquals(test_file,member_for_test.writeFile());
    }

}

```

```

test for student member class
public class StudentMemberTest {
    final static Date date_2 = new Date(2012,12,11);
    final static StudentMember member_for_test_2= new
StudentMember("Mathuravan",date_2);

```

```

@Test
public void setSchoolName_testing() {
    String school_name = "Hindu college";
    member_for_test_2.setSchoolName(school_name);
    assertEquals(school_name, member_for_test_2.getSchoolName());
}

@Test
public void getSchoolName_testing() {
    assertEquals("Hindu college", member_for_test_2.getSchoolName());
}

@Test
public void getType() {
    assertEquals("Student Member", member_for_test_2.getType());
}
}

unit test for over 60member class

public class Over60MemberTest {
    final static Date date_3 = new Date(2012,12,11);
    final static Over60Member member_for_test_3= new
Over60Member("Mathuravan", date_3, 65);

    @Test
    public void setAge_testing() {
        int age = 67;
        member_for_test_3.setAge(age);
        int output = member_for_test_3.getAge();
        assertEquals(67, output);
        member_for_test_3.setAge(65);
    }

    @Test
    public void getAge_testing() {
        int output = member_for_test_3.getAge();
        assertEquals(65, output);
    }

    @Test
    public void getType_testing() {
        assertEquals("Over60 Member", member_for_test_3.getType());
    }
}

```

Link of Youtube video: <https://www.youtube.com/watch?v=jVRj1gps3bA>