PP02 Course work 02 Report

- ✓ First Name -: Sithira
- ✓ Surname -: Panditha
- ✓ IIT ID -: 20191106
- ✓ UoW ID -: w1790355

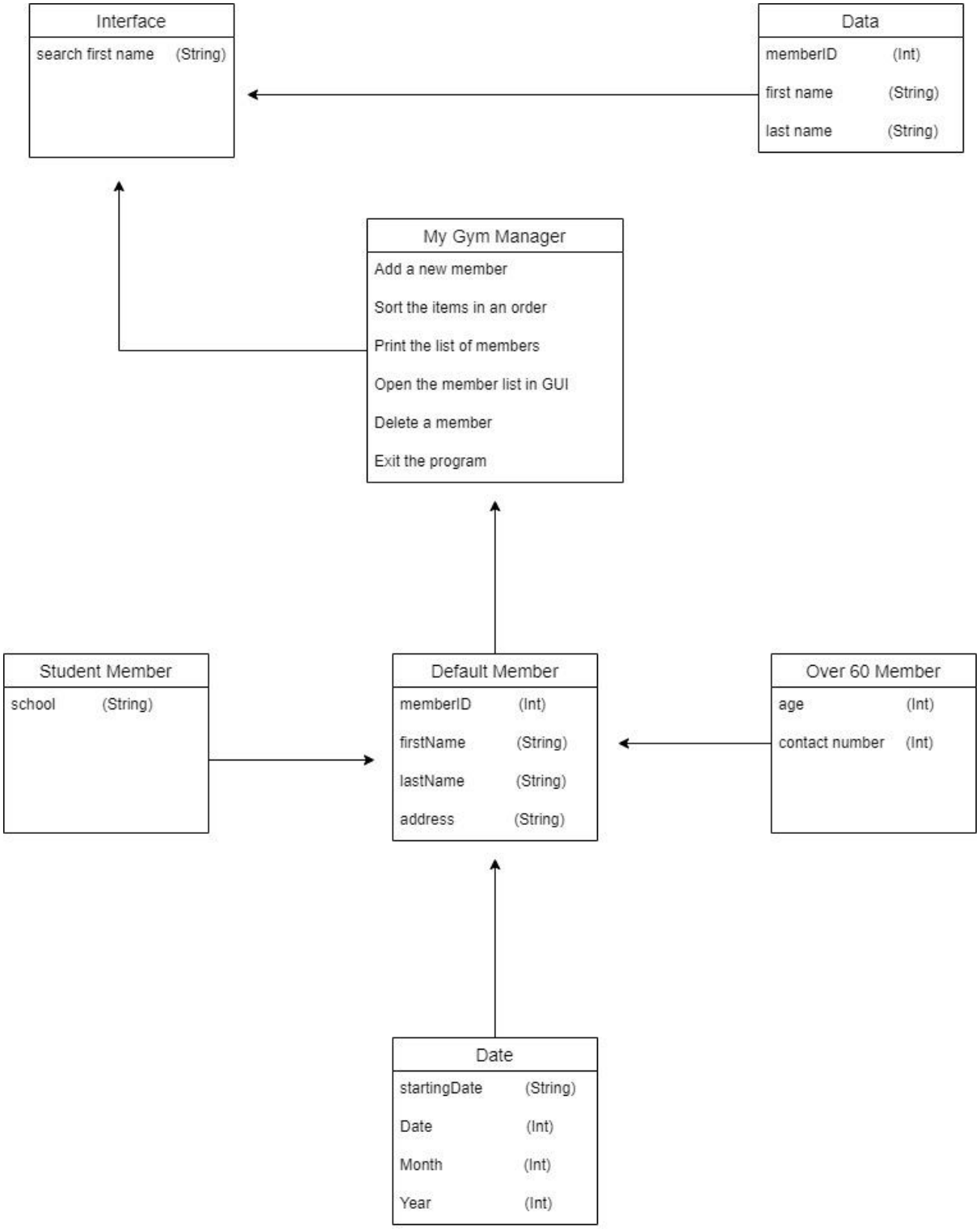
- ➢ Program demonstration link -:

https://youtu.be/24QzPkmc08w

- ➢ Brief explanation about the Program -:

- In this Course work, we have to design a Gym Management System. So, in the program the Admin of this program can do several things.
- One thing is, if there is a new member want to join the GYM, Admin can add his or her details to the program in 3 categories.
- All the member details are saved in the database related to this system.
- Admin able to save only 100 members details in this database. Otherwise it will be not able to save anymore details.
- Delete a member details, Sort in the items in an order, Print the list of the numbers are some other functions that admin can do by using this program.

- Class Diagram -:



➢ Code -:

- MyGymManager.java

```
import com.mongodb.BasicDBObject;
import com.mongodb.MongoClient;
import com.mongodb.MongoCommandException;
import com.mongodb.client.FindIterable;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import java.io.IOException;
import java.util.Scanner;

public class MyGymManager {


    //Connecting to MongoDB
    public static MongoClient mongoClient = new MongoClient("localhost",
27017);
    public static MongoDatabase database =
mongoClient.getDatabase("gymDatabase");

    public static void main(String[] args) throws IOException {

        while (true) {
            if (database.getCollection("gymMember") == null) {
                database.createCollection("gymMember");
            }

            MongoCollection<Document> input =
database.getCollection("gymMember");

            //Prompt the choices
            Scanner gym = new Scanner(System.in);

            System.out.println("********************Gym Management
System*******************");
            System.out.println();
            System.out.println("Enter number 01 to Add a new member");
            System.out.println("Enter number 02 to Sort the items in an
order");
            System.out.println("Enter number 03 to Print the list of
members");
            System.out.println("Enter number 04 to Open the member list in
GUI");
            System.out.println("Enter number 09 to Delete a member");
            System.out.println("Enter number 00 to Exit the program");
            System.out.println("");


            //user's choice validation
            int maximum = 0;

            int choice = gym.nextInt();
            if (choice == 1) {
```

```
                if (maximum >= 100) {
                    System.out.println();
                }
            }

            System.out.println("You entered number " + choice);
            System.out.println("------------------------------");

            if (choice == 1) {
                System.out.println("**********Add a new member**********");
                System.out.println("If you are Student Member please enter
01");
                System.out.println("If you are a Over60Member please enter
02");
                System.out.println("If you are a Default member please
enter 03");

                int select = gym.nextInt();

                if (select == 1) {
                    System.out.println("**********Student
Member**********");
                    StudentMember student_member = new StudentMember();
                    student_member.setMemberID();
                    student_member.setfstName();
                    student_member.setlstName();
                    student_member.setaddress();
                    student_member.setSchool();
                    Date date = new Date();
                    date.setStartingDate();
                    student_member.setDefaultdate(date.getStartingDate());

                    Document details = new Document();
                    details.put("Member ID", student_member.getMemberID());
                    details.put("First name", student_member.getfName());
                    details.put("Last name", student_member.getlName());
                    details.put("Address", student_member.getaddress());
                    details.put("School", student_member.getSchool());
                    details.put("Starting Date", date.getStartingDate());
                    input.insertOne(details);

                } else if (select == 2) {
                    System.out.println("**********over 60
Member**********");
                    Over60Member over60_member = new Over60Member();
                    over60_member.setMemberID();
                    over60_member.setfstName();
                    over60_member.setlstName();
                    over60_member.setaddress();
                    over60_member.setAge();
                    over60_member.setContact_number();
                    Date date = new Date();
                    date.setStartingDate();
                    over60_member.setDefaultdate(date.getStartingDate());

                    Document details = new Document();
                    details.put("Member ID", over60_member.getMemberID());
                    details.put("First name", over60_member.getfName());
                    details.put("Last name", over60_member.getlName());
                    details.put("Address", over60_member.getaddress());
                    details.put("Age", over60_member.getAge());
```

```
                    details.put("Contact_number",
over60_member.getContact_number());
                    details.put("Starting Date", date.getStartingDate());
                    input.insertOne(details);

                } else if (select == 3) {
                    System.out.println("**********Default
Member**********");
                    DefaultMember defaultMember = new DefaultMember();
                    defaultMember.setMemberID();
                    defaultMember.setfstName();
                    defaultMember.setlstName();
                    defaultMember.setaddress();
                    Date date = new Date();
                    date.setStartingDate();
                    defaultMember.setDefaultdate(date.getStartingDate());

                    Document details = new Document();
                    details.put("Member ID", defaultMember.getMemberID());
                    details.put("First name", defaultMember.getfName());
                    details.put("Last name", defaultMember.getlName());
                    details.put("Address", defaultMember.getaddress());
                    details.put("Starting Date", date.getStartingDate());
                    input.insertOne(details);

                } else {
                    System.out.println("Invalid Input.... Please enter a
valid input");
                }


            } else if (choice == 2) {
                System.out.println("**********Sort the items in an
order**********");
                FindIterable<Document> sort_items = input.find().sort(new
BasicDBObject("First name", 1));
                for (Document info : sort_items) {
                    System.out.println(info.get("Member ID") + " - " +
info.get("First name") + " - " + info.get("Last name"));
                }

            } else if (choice == 3) {
                System.out.println("**********Print the list of
members**********");

                for (Document data : input.find()) {
                    System.out.println("Member ID      -: " +
data.get("Member ID"));
                    System.out.println("First name     -: " +
data.get("First name"));
                    System.out.println("Last name      -: " +
data.get("Last name"));
                    System.out.println("Address        -: " +
data.get("Address"));
                    System.out.println("School         -: " +
data.get("School"));
                    System.out.println("Age            -: " +
data.get("Age"));
                    System.out.println("Starting Date  -: " +
data.get("Starting Date"));
                    System.out.println("");
```

```
                }


            } else if (choice == 4) {
                System.out.println("**********Prompt the member list in
GUI**********");
                Interface.Interface(args);

            } else if (choice == 9) {
                System.out.println("**********Delete a member**********");
                System.out.println("Enter the member ID-:");
                int id = gym.nextInt();
                BasicDBObject basicDBObject = new BasicDBObject();
                basicDBObject.put("Member ID", id);
                input.deleteOne(basicDBObject);
                System.out.println("Your member's details deleted
successfully");

            } else if (choice == 0) {
                System.out.println("Exit the programme");
                break;

            } else {
                System.out.println("Please enter a valid input");
            }


        }

    }
}
```

- DefaultMember.java -:

```
import java.util.Scanner;

public class DefaultMember {

    Scanner scanner = new Scanner(System.in);

    int memberID;
    String fstName;
    String lstName;
    String address;
    String newdefaultDate;

    public void setMemberID() {
        System.out.println("Please enter your member ID  -: ");
        while (true) {
            try {
                this.memberID = scanner.nextInt();
                break;
            } catch (Exception exception) {
                System.out.println("Invalid input... Please enter the
correct input again...");
```

```
                scanner.next();
            }
        }
    }

    public void setfstName() {
        System.out.println("Please enter your first name  -: ");
        this.fstName = scanner.next();
        while (true) {
            try {
                Integer.parseInt(this.fstName);
                System.out.println("Invalid input... Please check and enter
correct first name  -: ");
                this.fstName = scanner.next();
            }catch (Exception exception) {
                break;
            }
        }
    }

    public void setlstName() {
        System.out.println("Please enter your last name  -: ");
        this.lstName = scanner.next();
        while (true) {
            try {
                Integer.parseInt(this.lstName);
                System.out.println("Invalid input... Please check and enter
correct last name  -: ");
                this.lstName = scanner.next();
            }catch (Exception exception) {
                break;
            }
        }
    }

    public void setaddress() {
        System.out.println("Please enter your address  -: ");
        this.address = scanner.next();
        while (true) {
            try {
                Integer.parseInt(this.address);
                System.out.println("Invalid input... Please check and enter
correct address  -: ");
                this.address = scanner.next();
            }catch (Exception exception) {
                break;
            }
        }
    }

    public void setDefaultdate(String defaultdate) {this.newdefaultDate =
defaultdate;}

    public String getDefaultdate() {
        return newdefaultDate;
    }

    public int getMemberID() {return memberID;}
    public String getfName() {return fstName;}
    public String getlName() {return lstName;}
```

```
    public String getaddress() {return address;}
}
```

- StudentMember.java -:

```
import java.util.Scanner;

public class StudentMember extends DefaultMember {

    Scanner student = new Scanner(System.in);

    String school;

    public void setSchool() {
        System.out.println("Please enter your school name  -: ");
        this.school = student.next();
        while (true) {
            try {
                Integer.parseInt(this.school);
                System.out.println("Invalid input... Please check and enter
your input correctly  -: ");
                this.school = student.next();
            }catch (Exception exception) {
                break;
            }
        }
    }

    public String getSchool() {return school;}

}
```

- Over 60 member.java -:

```
import java.util.Scanner;

public class Over60Member extends DefaultMember {

    Scanner over_60 = new Scanner(System.in);

    int age;
    int Contact_number;

    public void setAge() {
        System.out.println("Please enter your age  -: ");
        this.age = over_60.nextInt();
        while (true) {
            try {
                Integer.parseInt(String.valueOf(this.age));
```

```
                System.out.println("Invalid input... Please check and enter
correct age  -: ");
                this.age = over_60.nextInt();
            } catch (Exception exception) {
                break;
            }
        }
    }

    public void setContact_number() {
        System.out.println("Please enter your contact number  -: ");
        this.Contact_number = over_60.nextInt();
        while (true) {
            try {
                Integer.parseInt(String.valueOf(this.Contact_number));
                System.out.println("Invalid input... Please check and enter
correct contact number  -: ");
                this.Contact_number = over_60.nextInt();
            } catch (Exception exception) {
                break;
            }
        }
    }

    public int getAge() {return age;}
    public int getContact_number() {return Contact_number;}
}
```

- Date.java -:

```
public class Date extends DefaultMember{

    String startingDate;
    int year;
    int month;
    int date;

    public void setStartingDate() {

        while (true) {
            System.out.println("Please enter your starting year  -: ");
            try {
                this.year = scanner.nextInt();
                break;
            } catch (Exception e) {
                System.out.println("Invalid input... Please enter a correct
input...");
            }
        }

        while (true) {
            System.out.println("Please enter your starting month  -: ");
            try {
                this.month = scanner.nextInt();
                break;
```

```
            } catch (Exception e) {
                System.out.println("Invalid input... Please enter a correct
input...");
            }
        }

        while (true) {
            System.out.println("Please enter your starting date  -: ");
            try {
                this.date = scanner.nextInt();
                break;
            } catch (Exception exception) {
                System.out.println("Invalid input... Please enter a correct
input...");
            }
        }

        startingDate = year + "/" + month + "/" + date;

    }

    public String getStartingDate() {
        return startingDate;
    }
}
```

- Interface.java -:

```
import com.mongodb.client.MongoCollection;
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;
import org.bson.Document;
import static com.mongodb.client.model.Filters.eq;
import java.util.Observable;

public class Interface extends Application {

    TableView<Object> table;
    public String searchFirstName;

    public static MongoCollection<Document> input =
MyGymManager.database.getCollection("gymDatabase");

    public ObservableList<Object> getTableData() {
        ObservableList<Object> observableList =
FXCollections.observableArrayList();
        for (Document document : input.find()) {
            Data addData = new Data();
            addData.setMemberID((Integer) document.get("Member ID"));
```

```
            addData.setfstName((String.valueOf(document.get("First
name"))));
            observableList.add(addData);

        }
        return observableList;
    }

    public ObservableList<Object> getTableSearch() {
        ObservableList<Object> search =
FXCollections.observableArrayList();
        for (Document document : input.find(eq("First name",
searchFirstName))){
            Data addData = new Data();
            addData.setMemberID((Integer) document.get("Member ID"));
            addData.setfstName(String.valueOf(document.get("First name")));
            addData.setlstName(String.valueOf(document.get("Last name")));
            search.add(addData);

        }
        return search;
    }

    @Override
    public void start(Stage primaryStage) throws Exception {
        primaryStage.setTitle("Gym Manager System");

        Label lblTitle = new Label("Gym Management System");
        lblTitle.setLayoutX(270);
        lblTitle.setLayoutY(40);
        lblTitle.setStyle("-fx-font: normal bold 25px 'serif'");

        TableColumn<Object, Integer> Column1 = new TableColumn<>("Member
ID");
        Column1.setCellValueFactory(new PropertyValueFactory<>("Member
ID"));
        Column1.setMinWidth(90);

        TableColumn<Object, String> Column2 = new TableColumn<>("First
name");
        Column2.setCellValueFactory(new PropertyValueFactory<>("First
name"));
        Column2.setMinWidth(200);

        TableColumn<Object, String> Column3 = new TableColumn<>("Last
name");
        Column3.setCellValueFactory(new PropertyValueFactory<>("Last
name"));
        Column3.setMinWidth(200);

        TextField txtsearch = new TextField();
        txtsearch.setPromptText("First name");
        txtsearch.setLayoutX(430);
        txtsearch.setLayoutY(540);
        txtsearch.setOnMouseExited(event -> searchFirstName =
txtsearch.getText());

        Button btnsearch = new Button("Search");
        btnsearch.setLayoutX(590);
        btnsearch.setLayoutY(540);
        btnsearch.setOnAction(event -> searchTable());
```

```
        table = new TableView<>();
        table.setLayoutX(160);
        table.setLayoutY(100);
        table.setItems(getTableData());
        table.getColumns().add(Column1);
        table.getColumns().add(Column2);
        table.getColumns().add(Column3);



        Pane root = new Pane();
        root.getChildren().add(lblTitle);
        root.getChildren().add(txtsearch);
        root.getChildren().add(btnsearch);
        root.getChildren().add(table);

        primaryStage.setScene(new Scene(root, 800, 600));
        primaryStage.show();

    }

    void searchTable() {table.setItems(getTableSearch());}

    public static void Interface(String[] args) {
        launch(args);
    }
}
```

- Data.java -:

```
public class Data extends Interface {

    int memberID;
    String fstName;
    String lstName;

    public void setMemberID(int memberID) {this.memberID = memberID;}
    public void setfstName(String fstName) {this.fstName = fstName;}
    public void setlstName(String lstName) {this.lstName = lstName;}
    public int getMemberID() {return memberID;}
    public String getfstName() {return fstName;}
    public String getlstName() {return lstName;}

}
```

Test Plan

01). Add a new member - **Yes**

- When we are considering about the adding a new member, we able to add them in 3 categories. "**Student member, Default member & Over 60 member"** are them.
- After you selected your choice, you can add a new member throughout that and the details of him or her will be saved in the database.

02). Sort the items in an order - **Yes**

- When you choosing this function by entering the number which belongs to this, you will be able to see the list of details that you have entered to the program by ascending order.

03). Print the list of members - **Yes**

- If you give **number 3** as your choice when you running this program, you can see a detail list of members that you entered to the system.

04). Open the member list in GUI - **NO**

- After you entered **number 4,** you can see the list of the details of the members in **Graphical interface**. There is only designing part in below code because there are several errors in running the code and none of the member details save in there. **But MongoDB system is able to save the member details.**

05). Delete member - **Yes**

- If you want to delete member details that you want, you can do that after entered **number 9** to the program.

06). Exit the program - **Yes**

- You want to quit the programme, you can quit by entered **number 0** in the console.