Programming for Software Engineers 7COM1025 Weekend Fitness club (WFC) System Using Java

Student Name: Umesh Bodala

Student ID: 21082836

Table of Contents

1	Introduction	2
2	Description of System	3
3	UML Class Diagram	3
4	Implementation Procedure	5
5	JUnit Testing Procedure	0
6	Conclusion	. 1
7	References	2
Li	st of Figures	
Fig	gure 1: Main page	6
Fig	gure 2: Appointment types	6
Fig	gure 3: Days appointment	7
Fig	gure 4: Sunday appointment	7
Fig	gure 5: Select types	7
Fig	gure 6: Review page	8
Fig	gure 7: Monthly report	9

1 Introduction

The report aims to develop the weekend fitness club system that follows the concept of Java programming language with the consideration of GUI (graphical user interface) principles. With the help of the fitness club system user or member is able to attend the Lessons in the proper manner. As it is known that Java is associated with the object-oriented programming language concept so the research aims to develop the four classes which are ZUMBA, BODY SCULPT, YOGA, and SPIN. These classes are associated with different functions. In order to visualize the development section, the research is incorporated with the UML class system. The study also includes the development and working procedure of the system and the entire system is tested with the help of Junit testing.

2 Description of System

The system aims to manage the bookings of the customer by avoiding any type of overlapping among the booking. The system is associated with different types of lessons which are YOGA, SPIN, ZUMBA, and BODYSCULPT. The system defines that each lesson can have a maximum of five bookings on each day which are Sunday and Saturday. There are only two lessons scheduled on the same day. The system is also associated with the feedback option in the five different categories. The user can select the rating of feedback from 1 to 5. The system also contains the report generation so the member of the club can access their report and progress. It is also associated with payment options.

3 UML Class Diagram

A class diagram in the UML (Unified Modelling Language) is a part of a static structure diagram used in software engineering that illustrates the classes, instances, operations, and relationships between objects in a system in order to describe the structure or hierarchy of the system. The main aim of object-oriented modelling is the class diagram. It is used for detailed modelling, which converts the business models into programming code, as well as for general conceptual modelling on the basis application's structure which is the WFC (Weekend Fitness Club). The class diagram consists of the different classes and their objects (Utsa, 2023). Basically, it is the representation of the entire code that is required for the development of the WFC system. The following figure is the representation of a class diagram that is built with the help of the draw.io software which allows for building different UML diagrams.

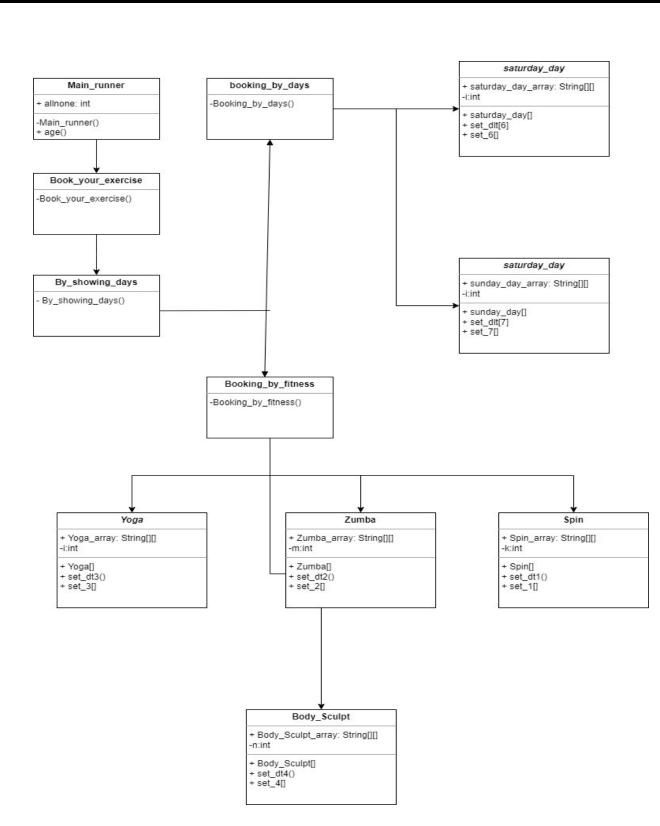


Figure 1: UML diagram

4 Implementation Procedure

The following stages are considered the necessary stage in order to develop the GU with Java Programming language. The development of GUI with the help of Java can be an efficient and easier process with the help of an integrated development environment (IDE) like NetBeans. NetBeans is defined as an open-source IDE which provides a visual designer in order to create GUIs.

- 1. **Installation of NetBeans**: The first Stage refers for download and installing the NetBeans on the machine. NetBeans is open source and it is available free on the official website, and installation of Netbeans is considered as an easy process.
- 2. **New project creation**: After the installation of NetBeans, the next stage refers to creating a new project that can be performed by selecting "Java" from the category list and "Java Application" as the type of project. It also includes the allocation of project names then leads to finishing the project.
- 3. **Jframe Frame**: After creating the new project, the next stage refers to the creation of a new JFrame form. Then there is a requirement for right-clicking on the project name and then selecting "New" > "JFrame Form" from the menu of context. The last stage refers to the finishing of the stage.
- 4. **Design of GUI**: As the JFrame form is created, then the visual designer will open in the auto manner. We design the UI to drag and drop components' text fields, like buttons, and labels onto the form. Then the last stage refers to arranging the components in order to create the desired layout.
- 5. **Customization of components**: As the components are arranged, the next stage is referred to the customization of such items. After that, we change the properties of every component like the font, size, and color. You can also add event listeners to the components by double-clicking on them.
- 6. **Write the code**: After the completion of the design of the GUI, the next stage refers for write the code that will implement the logic of WFC. We can also switch to the "Source" view in NetBeans in order to write the code in an automatic manner. So, it can be stated that NetBeans

is the easier and more efficient procedure for writing the code.

- 7. **Test of GUI**: After the completion of the coding section there is a requirement for testing so we adopted the Junit testing method for testing the GUI. Apart from the method, we can also test the application by pressing the "Run" button in NetBeans (Apache NetBeans, 2017).
- 8. Snap shots of the program



Figure 2: Main page



Figure 3: Appointment types

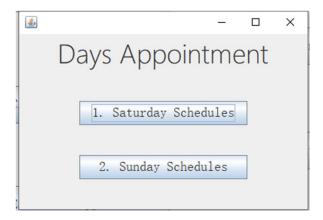


Figure 4: Days appointment



Figure 5: Sunday appointment

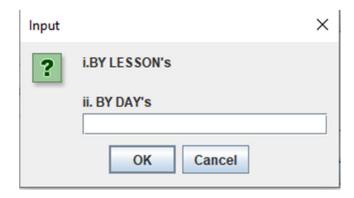


Figure 6: Select types

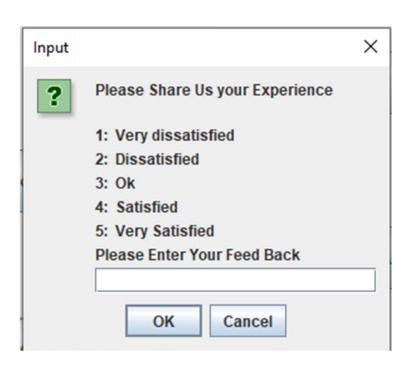


Figure 7: Review page

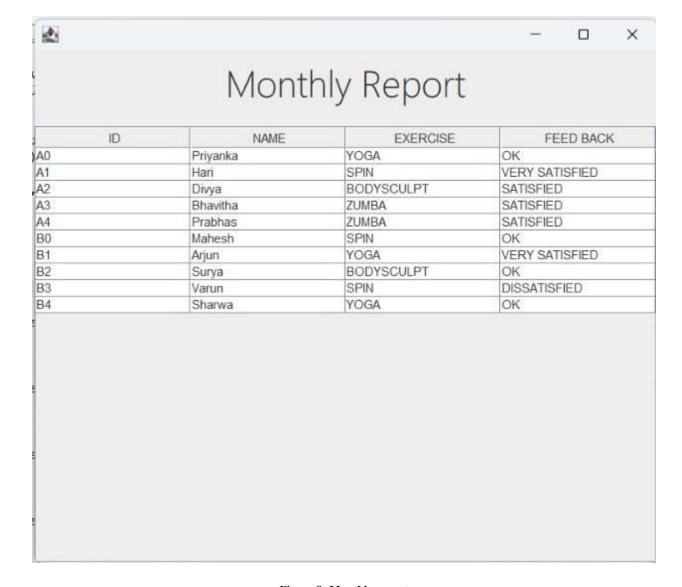


Figure 8: Monthly report

5 JUnit Testing Procedure

In order to test the GUI, we adopted the Junit method so created a main function in the different test cases. As the test cases are able to execute successfully then it can be stated that the application is running in the proper manner (Toure et al., 2014).

```
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;
public class Main_runnerTest {
   Main_runner run1;
    public Main_runnerTest() {
    @BeforeEach
    public void setUp() {
        run1 = new Main_runner();
    @AfterEach
    public void tearDown() {
        run1 = null;
    @Test
    public void testMain() {
        String name="Bhavesh";
        int a = run1.amount(name);
        assertEquals(500,a);
```

Figure 9: JUnit testing

6 Conclusion

The research with practical and theoretical analysis is enough to state that programming languages have the ability to solve various complex manner in an efficient manner by developing the software. The study is conducted as part of the technical report for defining the WFC system. The research includes the various concepts of software engineering which are the UML class diagram, the Use of the NetBeans platform, programming with Java, and Junit testing. The research is enough able to manage the different lessons as per the requirement of the business.

Version Control:

https://github.com/ub22aac/21082836 umesh

7 References

Apache NetBeans. (2017). *Introduction to GUI Building*. [online] Apache.org. Available at: https://netbeans.apache.org/kb/docs/java/gui-functionality.html [Accessed 22 Apr. 2023].

Toure, F., Badri, M. and Lamontagne, L., (2014). A metrics suite for JUnit test code: a multiple case study on open source software. *Journal of Software Engineering Research and Development*, 2(1), pp.1-32.

Utsa. (2023). *UML Class Diagrams*. [online] Available at: http://www.cs.utsa.edu/~cs3443/uml/uml.html [Accessed 22 Apr. 2023].