Universidade Beira Interior
Departamento de Informática

Gym Program in Java

by

Joel Tapia (47275), Joao Mendonça (47817), Manuel Garcia (45500), Gizem Nur (51182)

Professor

Maria Paula Prata de Sousa

December 30th 2022

Acknowledgments

We would like to thank the *Universidade da Beira Interior* (UBI) and our professor Paula Prata, that made this project possible, by the knowledge that she passed to us during her lessons.

Acronyms

UBI *Universidade da Beira Interior*

Contents

Contents **7**

List of Figures **9**

1 Introduction **11**
1.1 Motivation 11

2 Menu Layout **13**
2.1 Main Menu 13
2.2 User's Menu 13
2.3 Teacher's Menu 14
2.4 Administrator's Menu 14

3 Debugging **17**
3.1 Recording of System's User's name 17
3.2 Order of Output 18
3.3 Concurrent Modification Exception 18
3.4 Teacher can't sign up in his own lessons 18
3.5 Teacher's lessons access 19

4 Conclusions **21**

List of Figures

1 Recording of User's name 17
2 Automated creation of files 18
3 The teacher is not allowed to sign up in his own lessons 18
4 Lesson's Constructor 19

Chapter

1

Introduction

1.1 Motivation

This project has in mind the creation of a program that presents a beginners friendly *User Interface* designed to be used by users, teachers and administrators at the Gym. It provides different profiles with respective privileges that offer appropriate choices.

The system is able to store, modify or delete information, as well as connect the data and display a specific one, when asked, as long as it has such rights.

Chapter

2

Menu Layout

In the following pages it will show in broad strokes the functionality of our menu and the action options available to the user.

2.1 Main Menu

The main menu consists of three subdivisions which are "Leave, Login, Sign Up", the first one will serve to stop the program, the Login and Sign Up will be their own menus.

When the Login option is selected, four options will be displayed: "Cancel, User, Teacher, Administrator". The first one will be used to go back to the main menu, if we choose the other three, they are going to lead to their appropriate menus: "User Menu 2.2, Teacher Menu 2.3, Administrator Menu" 2.4, respectively, which will be explained in the following subtitles.

When the Sign Up option is selected, you will face the options "Cancel, User, Teacher". The two last options will also lead you to their respective menus, where it will be asked to insert the username and PIN (required to be of 4 integer digits to be accepted) for the account that is about to be created. Afterwards, an ID will be associated to the new account and it will be asked to remember that ID for future Logins.

2.2 User's Menu

Normal Users and Teachers (which are Users that have the privilege to teach lessons) can both Login as Users, to schedule their lessons as students. However, the Teacher will only be able to sign up for lessons from other teachers.

Besides the schedule of lessons, other options like "Check information" (check ID and name), "Check lessons" (check availability) and "Change PIN" will also be at disposal.

2.3 Teacher's Menu

The Teacher will be greeted with the options below:

1. **0- Logout:** it returns to the Main Menu.
2. **1- Check Information about myself:** shows ID and name.
3. **2- Check Lessons:** displays the lessons.
4. **3- Check Students:** presents the name and id of the students that are registered in the lessons of the teacher.
5. **4- Create Lessons:** when this option is selected the user is asked to specify the name of the new lesson and its maximum number of students.
6. **5- Choose schedule for a lesson:** the user will be asked to choose a day in the week and the id of one of his owns lessons in case he has any, the following menu will be shown. During this process, the teacher can cancel the schedule any time, by entering the number zero.
    a) **0- Cancel** cancel and go back
    b) **1- Morning:** you may choose schedule the class you previous selected in any hours available between 8 and 11 am. It will be showed if that hour is "Free" or "Occupied" and only the hours that are free will accept your request to schedule. Enter zero to go back to the Teacher's Menu 2.3.
    c) **2- Evening:** The process is the same as above, but this time, you may only schedule the class for any hour between 2 and 6 pm.

2.4 Administrator's Menu

To Enter the Administrator's Menu, you will need to Login as an Administrator with the PIN "0000".

After you successfully Login, following options will be revealed:

1. **0- Cancel:**

2. **1- Check users**
3. **2- Check teachers**
4. **3- Delete account**

By checking either the users or the teachers, you will have the power to check the information (ID and name), as well as search a specific user or teacher by ID.

Chapter

3

Debugging

In this chapter, we are about to explore the algorithms that made possible the correct functioning of the program, and avoided bugs, such as the following.

3.1 Recording of System's User's name

The following code records the System's User's name of the machine, so it can be used in any Windows machine.

```
4
5  public class Main {
6
7      private static String so = System.getProperty("user.name");
8
9      public static void main(String[] args) {
10
11
12
13         ArrayList<User> users = new ArrayList<>();
14         try {
15             ObjectInputStream isUser = new ObjectInputStream(new FileInputStream(
16                     "C:\\Users\\"+ so +"\\Desktop\\User.dat"));
17             int last = isUser.readInt();
18             User.setLast(last);
19             users = (ArrayList<User>) isUser.readObject();
20         } catch (IOException e) {
21             System.out.println(e.getMessage());
22         } catch (ClassNotFoundException e) {
23             System.out.println(e.getMessage());
24         }
25
```

Figure 1: Recording of User's name

3.2 Order of Output

When we were outputting the content that our program generated, we change the order of the output functions and so the program sometimes wouldn't read the outputted files correctly, so we needed to type the writeInt() method before we typed the writeObject() into the code, and that fixed the bug

```
123
124         try {
125             ObjectOutputStream osUsers = new ObjectOutputStream(
126                     new FileOutputStream("C:\\Users\\"+so+"\\Desktop\\User.dat"));
127             osUsers.writeInt(User.getLast());
128             osUsers.writeObject(users);
129             osUsers.flush();
130         } catch (IOException e) {
131             System.out.println(e.getMessage());
132         }
133         try {
134             ObjectOutputStream osLesson = new ObjectOutputStream(
135                     new FileOutputStream("C:\\Users\\"+so+"\\Desktop\\Lesson.dat"));
136             osLesson.writeInt(Lesson.getLast());
137             osLesson.writeObject(lessons);
138             osLesson.flush();
139         } catch (IOException e) {
140             System.out.println(e.getMessage());
141         }
142
```

Figure 2: Automated creation of files

3.3 Concurrent Modification Exception

We needed to import the *concurrent* hash map because when we were accessing an hash map in the Admin Class, we sometimes removed objects from the hash map, and that would cause the *ConcurrentModificationException*, causing the program to crash and that was fixed by using this special type of hash map: *import java.util.concurrent.ConcurrentHashMap*.

3.4 Teacher can't sign up in his own lessons

Below, you can see that prevented the Teacher's from signing up in their own lessons, by using a *return* in the void method, to achieve that.

```
if(idsToWeekday.get(weekdayChoosen).hmMorningLessons.get(answer).getTeacherId()== id)
{
    System.out.println("You can't sign up for your own lesson!");
    return;
}
```

Figure 3: The teacher is not allowed to sign up in his own lessons

3.5 Teacher's lessons access

When we were programming and testing the code, we found it very difficult to access the Teacher's lessons. Everything turned simpler when we added the *teacherId* parameter to the Lesson's Constructor.

```
32    public Lesson(String name, String teacher, int capacity, int teacherId) {
33        this.name = name;
34        this.capacity = capacity;
35        last++;
36        id = last;
37        this.teacher = teacher;
38        this.teacherId= teacherId;
39        usersInLesson = new ArrayList<>(capacity);
40    }
```

Figure 4: Lesson's Constructor

Chapter

4

Conclusions

This project challenged our Java skills, making us find ways to avoid bugs and deliver a great User Interface, where it's easy for new users to check, create and sign up for lessons in the gym. We also added the Administrator Profile (Privileged) that can be used to manage and moderate the gym.