Name: Ikemefuna Ezechukwu Student number: 2060576

Programming IT

Assessed Exercise Report 3

The program designed matches the specifications required.

There have been assumptions made that have affected implementation. Each fitness class has been assumed to have an ID that is short (3 letters long) and unique. Each class has been assumed to have a name and a tutor, with both names being only one word (no spaces). Each class has been assumed to have a unique start time between the hours 9 – 15.

The Fitness class list is manipulated by the addition and deletion fitness classes.

The program functions properly. The time table is displayed once the program is opened, and the view attendance button displays a report of the fitness classes in the list (See Figure 1). When the report window is closed, the time table window is still opened.

The save and exit button, writes the id, name, tutor name, and start time of a class to a file (See Figure 2).

Several inputs were used to test the robustness of the program.

**When adding classes**

No data was inputted in any of the text fields. This resulted in an error message asking that all fields must be filled, and a stop of the add function (as expected). See Figure 3.

An ID longer or shorter than 3 letters was inputted as part of the required information. As a result, an error message was given, and the add function stopped running (as expected). See Figure 4.

Inputting a space in any of the text field resulted in appropriate error messages, and a stop of the add function (as expected). See Figure 5.

Adding a class with an ID that already exists in the list resulted in a warning message informing that the class exists, and a stop of the add function (as expected). See Figure 6.

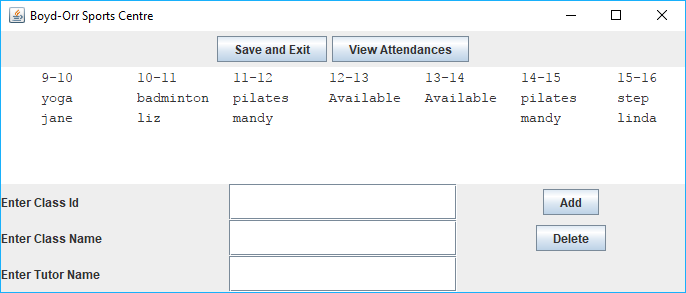
Adding a class with a name and/or a tutor that exists, but a unique ID, resulted in proper functioning (as expected). See Figure 7.

Adding a class when the list was full resulted in a warning message informing that the list full, and a stop in the add function. See Figure 8.

**When deleting classes**

Deleting an existing class resulted in proper functioning (as expected). See Figure 9.

Deleting a non-existing class resulted in an error message informing that the class does not exist, and then a stop in the delete function (as expected). See Figure 10.



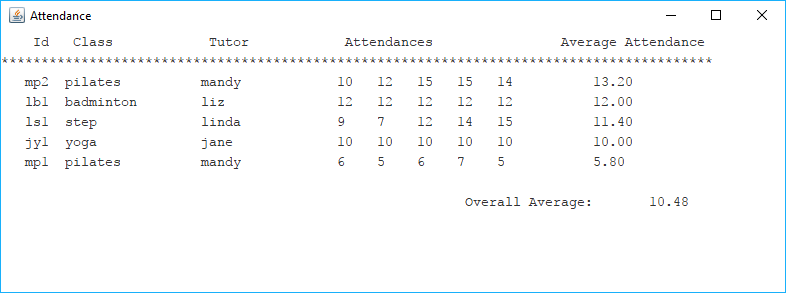


Figure 1: The time table display, and the report display

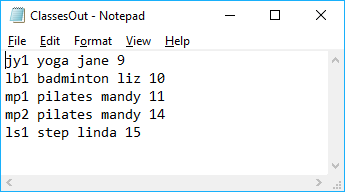
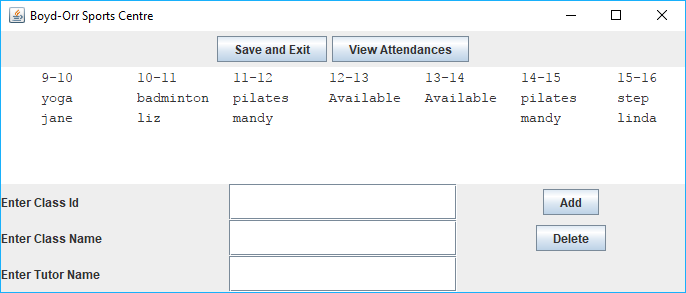
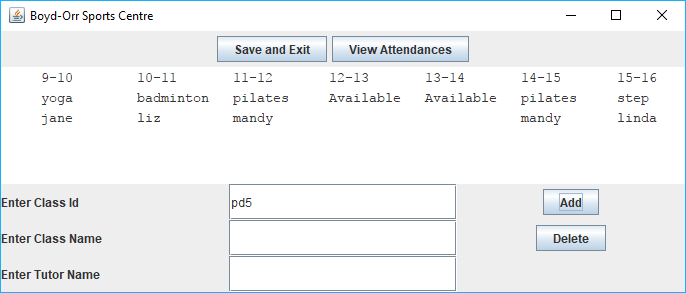
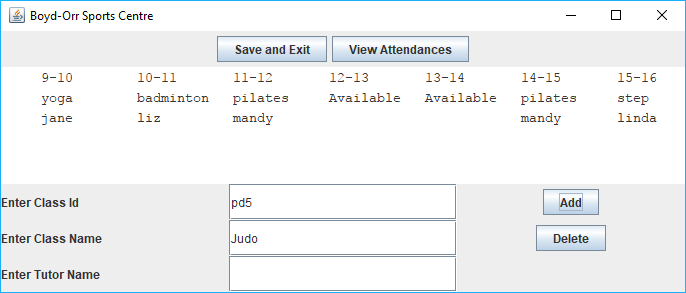
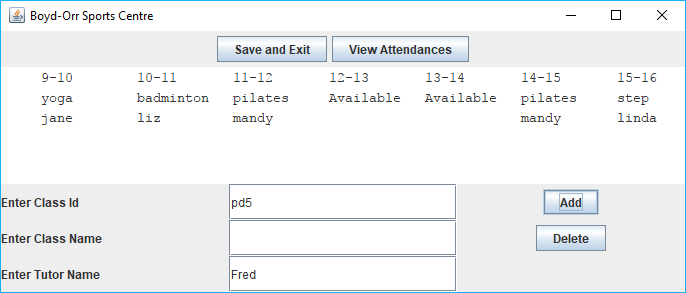


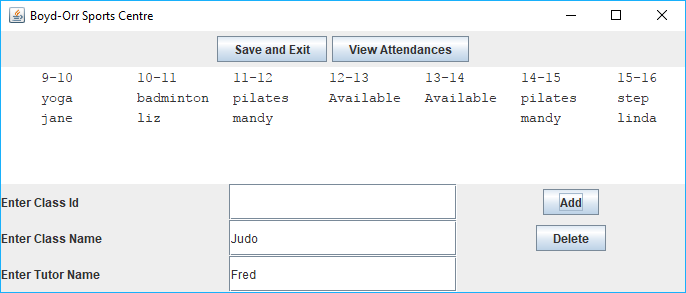
Figure 2: The file produced by clicking the save and exit button. The classes were not manipulated.

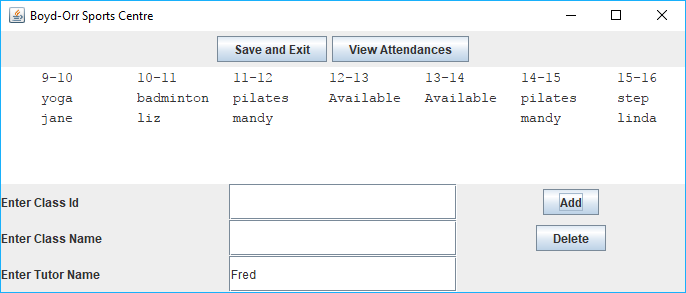












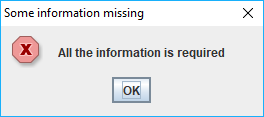
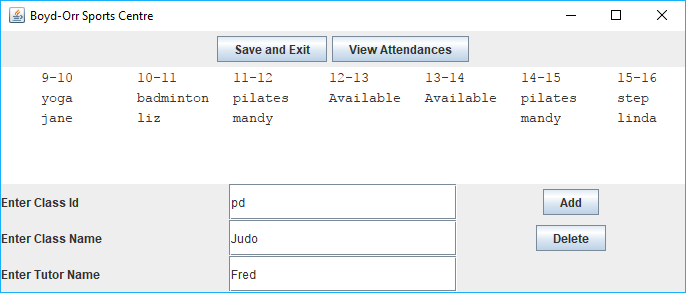
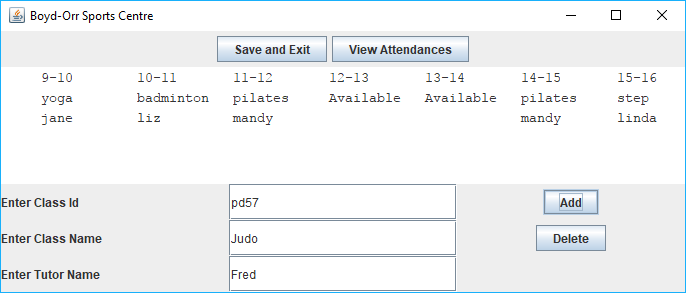


Figure 3: No text field is allowed to be empty upon adding.





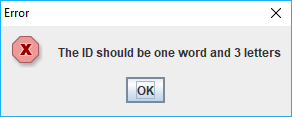
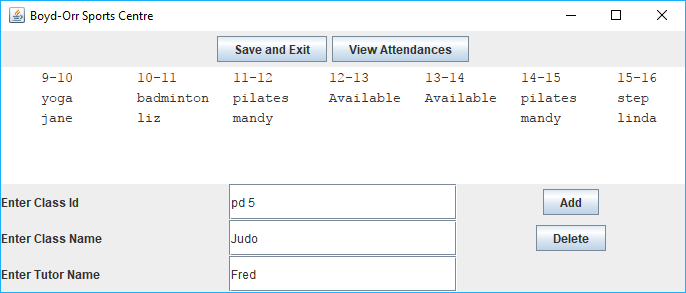
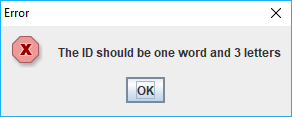
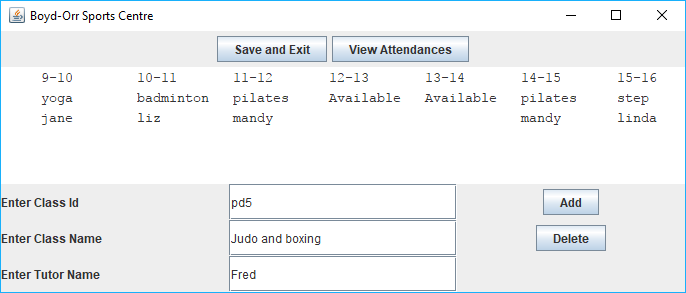
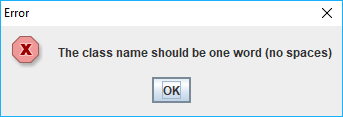


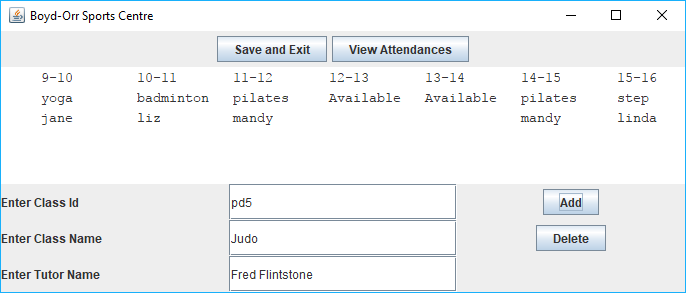
Figure 4: ID has to be three letters.











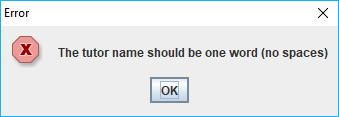
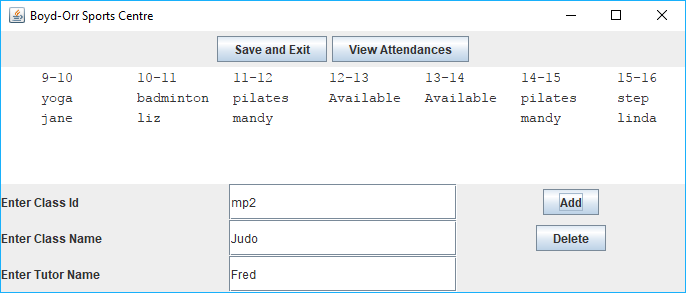


Figure 5



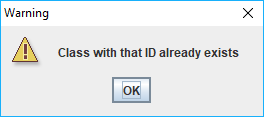
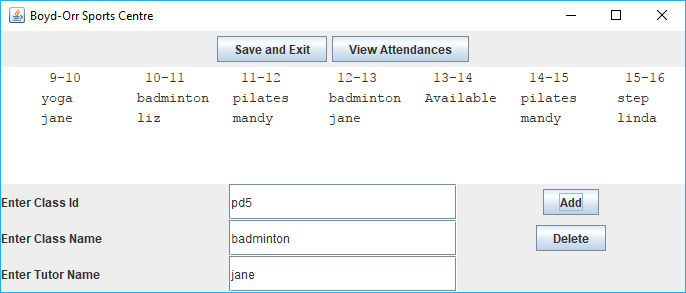
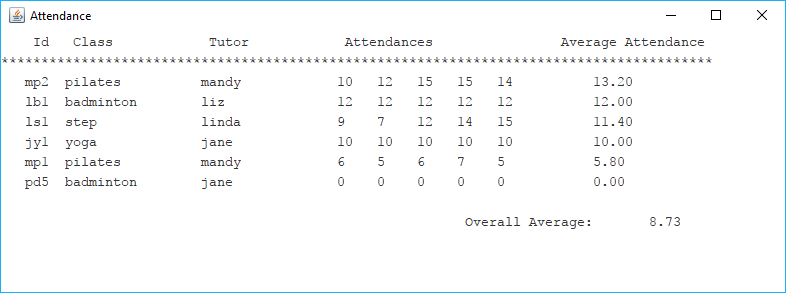


Figure 6: ID has to be unique.





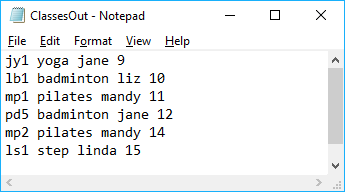
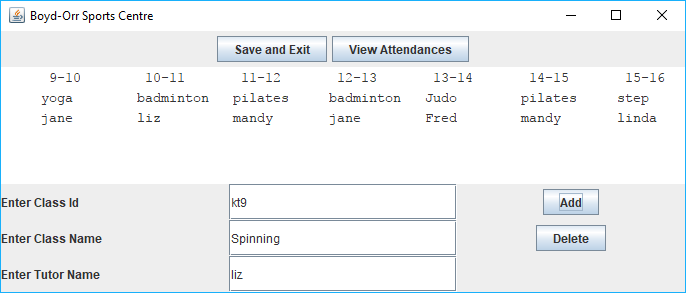


Figure 7.



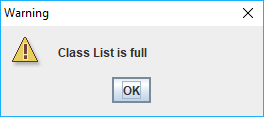
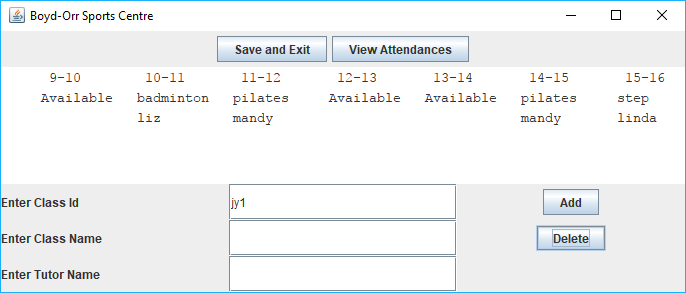
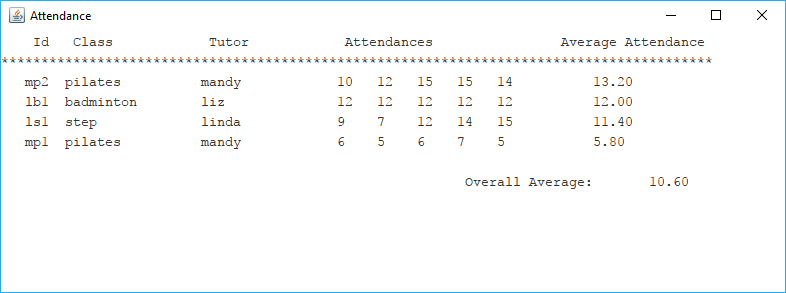


Figure 8.





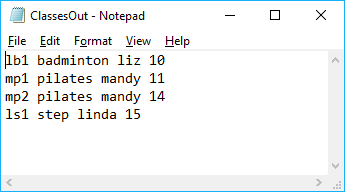


Figure 9: Delete function works according to specifications.

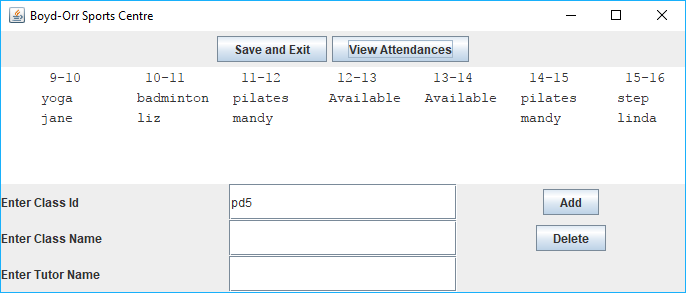




Figure 10: Cannot delete a non-existing class.