## Calvin Hew 207885H

**Assessed Exercise 3: Boyd-Orr Sports Centre**

The final of the program is complete with regards to addressing the requirements highlighted in specification document. The program successfully displays the current schedule of classes and an attendance report which are both read from text files. It can process the creation/allocation of a new class to an available time slot, as well as deletion of an existing class provided the class ID is given as an input.

A number of assumptions were considered when the program was written. Though it was assumed that the ID assigned to a Fitness Class would be unique, no specific check for the format in which the ID should be in as it is just expected to be a short and unique String. Thus, it is assumed that the user will pick a sensible format (such as a combination of the class name, tutor and a number as used in specification). Another assumption was made regarding the format of the text files that were being read in or written to output files. For example, every AttendancesIn.txt file should contain exactly 5 records of attendance and no more/less. There is also no maximum or minimum capacity of attendance for any given class. Thus, it is assumed that these files are manually checked and verified prior to using the program.

Tests were conducted using the AttendancesIn.txt and ClassesIn.txt which were provided in the setup files.

*Appendix 1* shows the interface containing the initial schedule which displays the classes or available times from 9am to 3pm (constructed from the ClassesIn.txt). It is expected that the schedule should display empty/available slots from 12-13 and 13-14, whilst the rest of the time slots should be populated with classes. The attendance report for the initial schedule (constructed from AttendancesIn.txt) is also show. The report should contain a total of 5 attendance records, representing the attendance over 5 weeks and an average value. An overall average attendance figure considering all classes is also provided.

*Appendix 2* provides a test of successfully adding a new Fitness Class to an available time slot. It is expected that **zumba** taught by **laura** should appear at the time slot rom 12-13. The updated attendance report will hold a record of this new class (initialised at 0, 0, 0, 0, 0) and the overall attendance average will change to 8.73, taking into consideration the additional class.

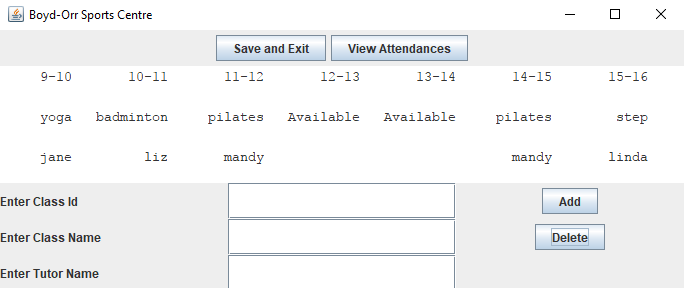
*Appendix 3* tests for the scenario in which a duplicate ID is entered by the user or if no time slot is available when adding a new Fitness Class. It is expected that an error message will occur for each of these instances.

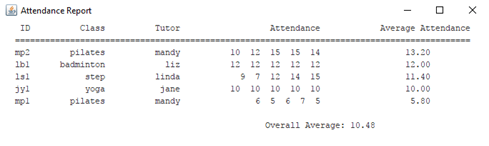
*Appendix 4* provides a test of successfully deleting an existing Fitness Class, with the schedule and report updated accordingly. It is expected that **badminton** taught by **liz** should be removed from the time slot 10-11 if processed successfully.

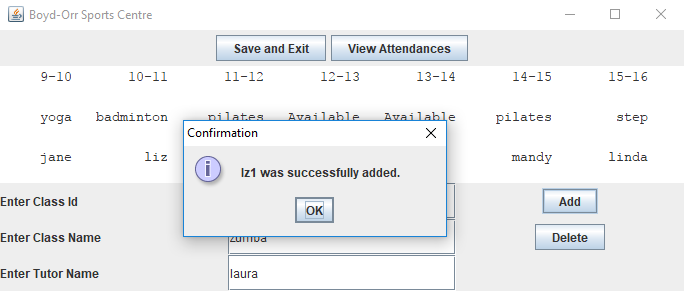
*Appendix 5* highlights the case in which the deletion of an existing class failed due to an unidentified ID input from the user. It is expected that an error message will notify the user of this issue.

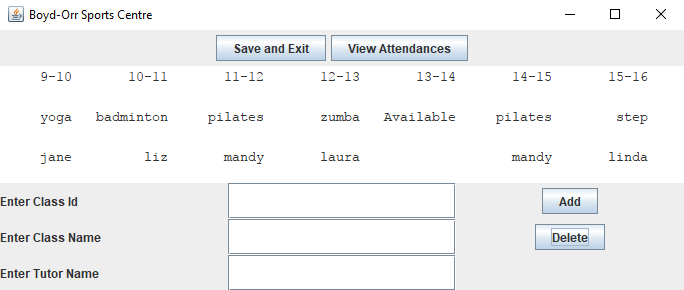
*Appendix 6* provides additional cases in which an exception may be caught due to having empty input fields: no class ID, name or tutor when adding a new Fitness Class or no class ID when deleting an existing Fitness Class. In these instances, an error message will notify the user of the particular missing field.

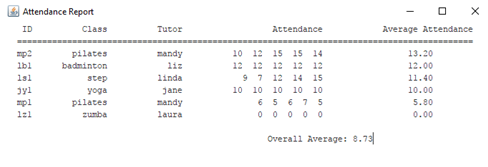
## Appendices

***Appendix 1: Displaying initial schedule and attendance report***

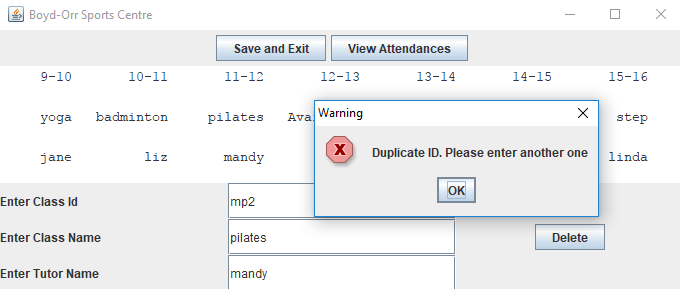


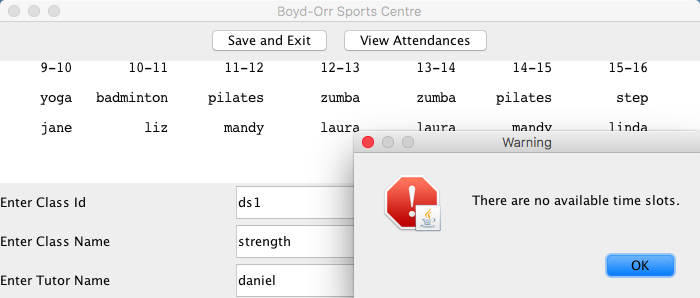
***Appendix 2: Successfully adding a new Fitness Class***



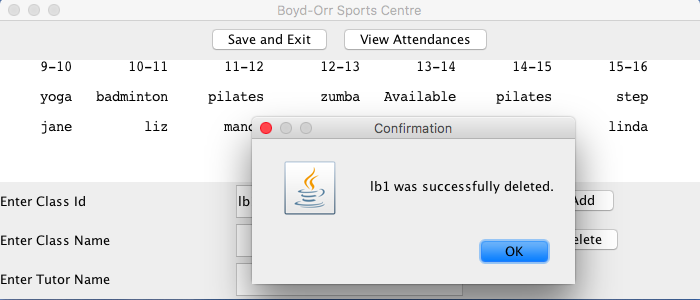


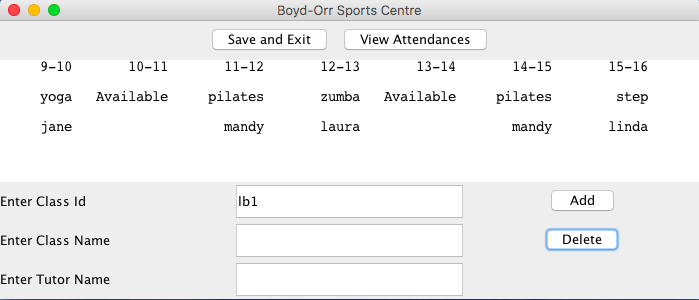
***Appendix 3: Unsuccessfully adding a new Fitness Class (duplicate ID and no time slots)***



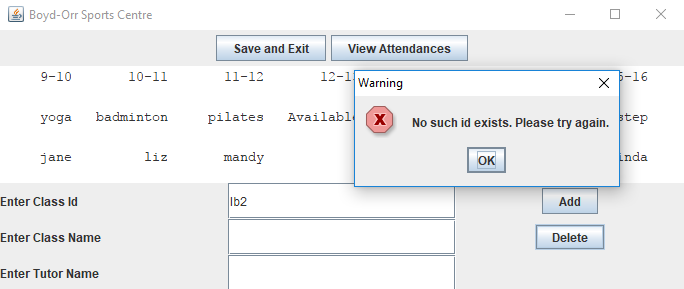


***Appendix 4:Successfully deleting a Fitness Class***





***Appendix 5: Unsuccessfully deleting a Fitness Class***



***Appendix 6: Error Messages for empty input fields***

