**7COM1025: Programming for Software Engineers**

**Student Name:  
Student ID:**

Table of Contents

[Introduction 3](#_Toc100955202)

[Create Class diagram for Java classes in system 3](#_Toc100955203)

[Associations and attributes 3](#_Toc100955204)

[Implement System using Junit Tests 5](#_Toc100955205)

[Assumptions 5](#_Toc100955206)

[Design of program 6](#_Toc100955207)

[Designing principles used 7](#_Toc100955208)

[Junit Tests 7](#_Toc100955209)

[Refactoring used for the system development 8](#_Toc100955210)

[References 9](#_Toc100955211)

# Introduction

The discussion is about the University Sports Centre which needs the proper software for the managing of the booking of group exercises. It is for the lessons which is then made by the students and then offering the groups with exercising lessons on Saturday and Sunday. The lessons on Yoga, Zumba and the Box Fit is where each lesson accommodate 4 students are there at the most. The lessons per day are 1 in morning, afternoon, and the last in evening. There are pricing which is for the different lessons and the pricing is for the same exercise which tend to remain same and is for the different time. The students look for the booking of the lesson needs with the check on the timetable with the selection on the lesson for the day. The student can also check on the timetable with the two ways on specifying on the days and the other for the specific exercise name. The students are then allowed for handling the change of booking, with the spaces that are available for the new elected lesson. The student can easily book the lessons as soon as they want to till there are no conflicts (Patnaik et al., 2020).

# Create Class diagram for Java classes in system

Graphical user interface, application

Description automatically generated with medium confidence

## Associations and attributes

The association is about the reviewing of the system with the ranges that are about the dissatisfied, satisfied, and the review for the rating information which is then recorded on the system. The report includes the number of students in a group with the exercising of the lessons for the average rating. The report is about the group exercises with the generation on the high income, that counts for the same exercising of the lessons together. The approach is about the report that contains the different students that tend to exercise for the group lessons and the average rating process. It is for the generating of the higher income that counts for the exercise lessons as well. The data covers the different types of the group lesson exercises with the students to review on the rating. This is for the external database which is for the output for the suitable GUI interface. There is a need for the security control with the pre-registered form (Santana et al., 2020).

Graphical user interface, application

Description automatically generated with medium confidence

With the advancement of technology, there are waterfall model which is for the Agile where the code is displayed for the production which is fast. The unit testing and integration is for the individual units that are tested for checking the units with the parameters which might or might not return on the values. The Junit is for the testing with the applications like GUIs for the required human testers. The understanding is for the automated testing with the code with the confidence. Hence, the programmer is the fast coder with the seeing of the different failed tests. The unit testing and the regression is for the form of white box with the cases that are based on the internal structure. The tester tends to choose the inputs for exploring the paths and then determine the outputs as well. Not only this, there are features for the different aspects of the project.

# Implement System using Junit Tests

Junit tutorial provides the unit testing in Java with the examples that involves the cost for the open source testing. It is for the performance unit testing and then creating the different test cases. The execution is for the tool support with the automated testing which is reliable and the test annotation is for the methods that determines about the new code for ensuring that nothing is broken. The Junit Regression Testing Framework is for the implementing on the unit testing in Java. The acceleration is for the programming speed and then increase on the quality of code. The features are for the Junit Test Framework with the Fixtures, test suites and the test runners. The fixtures are for the fixed state that are for set of objects and then there are baseline for the running tests. The software unit tests is for the helping of the developers to verify on the logic of the piece of program. The running tests are for identifying the regressions of the software with the change on the source code. It is for the understanding of applications under the test, with the test fixtures that are for the precondition testing. The unit testing is for the functionality with the assertion on different behaviours and states.

# Assumptions

The assumptions are for the integration testing with the set of components. It comes with the behaviour tests to check on the certain methods. The state testing is for the validation of result with the behaviour setting that is for the application under the test. The testing framework is for Junit and the TestNG. It comes with the separate project or the source folder to keep the test code and then there are separation from the real code. The Junit framework is for the test annotation, with the execution of the code under the test. The messages are for the asserting of statements (Machado, 2020). The Junit is for the annotations to mark the methods with the testing methods or configuring them. The assert statements are for the comparing to the actual value which is then determined by the test for the expected value.

# Design of program

Graphical user interface, text, application, Word

Description automatically generated

# Designing principles used

The testing is based on the changes where Java Unit testing is for the repeated automated tests. It is for the framework with the evolvement over the time. The common IDEs is for the Eclipse and IntelliJ which are for the testing of integration and then building system to support the annotation. The Junit is for the associated approaches with the annotations that are for the deletion of temporary files and variables. The assert methods are for the test cases with the failure of assertions as recorded. It comes with the running of the JUnit from the common line. The benefits are for the unit testing with the isolation and the focused testing, and then assuring behaviour for the methods or the pieces with addition or modification of code. The boundary analysis is about the check on the invalid and the bad input with testing on aspects for the method. The unit test, integration test is for the associated components and classes. The acceptance test is for the operation on the fully integrated system with the regression tests. The unit testing is for the output with the approaches that is not satisfactory. The assertion features are for the pre-conditions and the post-conditions with the unit testing framework (Saheb et al., 2021).

# Junit Tests

The unit tests are for the manual and automated testing with the execution of test cases with time consuming and tedious. The huge investments are for the execution manually with the less reliable approach, with the automated testing. The fast approach is about the significant approach with the lesser investment in the human resources. The reliable approach is for the automation testing and the testers can program on the bringing out of the hidden information. Junit is the unit testing framework with the test driven development with the features that are for the open source framework, with the use for the writing and running tests. The providing of the test runners is for the running tests and then allowing on the writing codes faster with the increased quality. The Unit Test Case is for the part of code with the handling of testing framework. It is for the perfect unit tests which comes through handling Java Programming language (Rizzi et al., 2018). The requirement is for the positive test and the negative test with the sub-requirements and the two test cases with the positive and negative.

The JWebUnit is for the Java based testing framework with the web applications. It wraps the existing testing framework, that is for the simple testing interface with the testing for the corrected web applications. API is for the navigation on the web applications which comes with the assertions that are for the application correctness. It comes with the methods that needs to be ready-to-use assertions with allowing on the rapid testing creation with the XML Unit which is for the Junit extension class.

# Refactoring used for the system development

The process is about the change of software system where there are no alterations of the function. The automated and semi-automated refactoring is by the rules of transformation. It comes with the operations that are defined with the code level, functional and the architecture level refactoring. The consolidation is for the statements which includes the extraction of the parameters of methods and then new classes are directed through the heuristic methods and functions. The refactoring is based on the components to segregate and the client web browser or the UI code is for the server side data manipulation. The generation is for the multi-tier application with the architecture involved with the separate high level business logic (Oda et al., 2021). The refactoring is for the improvement on the architecture design in the local scope. The refactoring process is about the system that is for the rebuilding of the same and then using the middleware deployment or alignment process through the user interface methods. The refactoring is for the offering of the significant benefits to organization that tend to seek for the streamlining and modernizing the existing code bases.

# References

Alenezi, M., Akour, M. and Al Qasem, O., 2020. Harnessing deep learning algorithms to predict software refactoring. *Telkomnika*, *18*(6), pp.2977-2982.

Martinez, M., Etien, A., Ducasse, S. and Fuhrman, C., 2020, June. Rtj: a Java framework for detecting and refactoring rotten green test cases. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings* (pp. 69-72).

Oda, T., Araki, K., Sahara, S., Chang, H.M. and Gorm, P., 2021. Refactoring for Exploratory Specification in VDM-SL. *Hugo Daniel Macedo, Casper Thule, and Ken Pierce (Editors)*, p.21.

Patnaik, A., Panigrahi, R. and Padhy, N., 2020, March. Prediction Of Accuracy On Open Source Java Projects Using Class Level Refactoring. In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)* (pp. 1-6). IEEE.

Rizzi, L., Fontana, F.A. and Roveda, R., 2018, September. Support for architectural smell refactoring. In *Proceedings of the 2nd International Workshop on Refactoring* (pp. 7-10).

Saheb Nasagh, R., Shahidi, M. and Ashtiani, M., 2021. A fuzzy genetic automatic refactoring approach to improve software maintainability and flexibility. *Soft Computing*, *25*(6), pp.4295-4325.

Santana, R., Martins, L., Rocha, L., Virgínio, T., Cruz, A., Costa, H. and Machado, I., 2020, October. RAIDE: a tool for Assertion Roulette and Duplicate Assert identification and refactoring. In *Proceedings of the 34th Brazilian Symposium on Software Engineering* (pp. 374-379).