|  |  |  |
| --- | --- | --- |
| CS3343-LA3 2014/15 Sem A December 5, 2014 | B.T.  Test Report | |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] | | Prepared By TH Kwan (52846408) |

# Contents

[Set Up 2](#_Toc405137611)

[Tear Down 2](#_Toc405137612)

[Test Cases Overview 2](#_Toc405137613)

[Test Cases 4](#_Toc405137614)

[Testing Strategy 5](#_Toc405137615)

[Coverage 6](#_Toc405137616)

[Statement Coverage in Eclemma 6](#_Toc405137617)

[“Branch” Coverage in EclEmma 7](#_Toc405137618)

[Test Results 9](#_Toc405137619)

# Set Up

Two variables, timeslots and timetable, are initialized to be used by test cases. The system output is set as a print stream such that the console output can be compared using JUnit test.

# Tear Down

The two variables defined in set up is set to point to null, as well as the system output.

# Test Cases Overview

|  |  |  |
| --- | --- | --- |
| **Test case** | **Significant related Class(es)** | **Corresponding Method(s)** |
| testSameDay | Timeslot | sameDay |
| testConstructorAndGetters | Timeslot | Constructor  getters |
| testSameBuilding | Timeslot | sameBuilding |
| testToString | Timeslot | toString |
| testOverlapTrue | Timeslot | overlap |
| testOverlapFalseSameDay | Timeslot | Overlap  sameDay |
| testOverlapFalseDiffDays | Timeslot | overlap |
| testDifference | Timeslot | difference |
| testToStringTimetable | Timetable | toString |
| testExtractByCode | Utilities | extractTimeslotsByCode |
| testExtractTimeslotsByType | Utilities | extractTimeslotsByType |
| testAllCourses | Utilities | allCourses |
| testPermutate | Utilities | permutate |
| testPermutateArrayList | Utilities | permutateArrayList |
| testGeneratePermutations | Utilities | extractTimeslotsByCode  extractTimeslotsByType  permutate  GeneratePermutations |
| testValidateInputDuplicateCrn | Utilities | validateInput |
| testValidateInputSession | Utilities | validateInput |
| testValidateInputBuilding | Utilities | validateInput |
| testValidateInputCourseStartTime | Utilities | validateInput |
| testValidateInputCourseFinishTime | Utilities | validateInput |
| testValidateInputCourseTimeRange | Utilities | validateInput |
| testValidateInputDay | Utilities | validateInput |
| testWeekday | Weekday | getDay |
| testPrintSchedule | IO | printSchedule |
| testPrintCRNs | IO | printCRNs |
| testReadRequiredConstraints | IO | readRequiredConstraints |
| testReadRequiredConstraintsFail | IO | readRequiredConstraints |
| testReadBuildingConstraints | IO | readBuildingConstraints |
| testReadBuildingConstraintsFail | IO | readBuildingConstraints |
| testReadTimeGapConstraint | IO | readTimeGapConstraint |
| testReadTimeGapConstraintFail | IO | readTimeGapConstraint |
| testReadTimeGapConstraintError | IO | readTimeGapConstraint |
| testReadTimeGapConstraintEmpty | IO | readTimeGapConstraint |
| testReadBuildingConstraintsEmpty | IO | readBuildingConstraints |
| testReadRequiredConstraintsEmpty | IO | readRequiredConstraints |
| testReadTimeConstraintsDayOff | IO | readTimeConstraints |
| testReadTimeConstraintsBefore | IO | readTimeConstraints |
| testReadTimeConstraintsAfter | IO | readTimeConstraints |
| testReadTimeConstraintsBetween | IO | readTimeConstraints |
| testReadTimeConstraintsError | IO | readTimeConstraints |
| testReadTimeslots | IO | readTimeslots |
| testReadTimeslotsFail | IO | readTimeslots |
| testRequiredConstraint | Timeslot  RequiredConstraint | Constructor  Constructor  isFulfilled |
| testTimeGapConstraint | Timeslot  TimeGapConstraint | Constructor  Constructor  isFulfilled |
| testTimeConstraint | Timeslot  TimeConstraint | Constructor  Constructor  isFulfilled |
| testBuildingConstraint | Timeslot  BuildingConstraint | Constructor  Constructor  isFulfilled |
| testRequiredAndTimeGapConstraints | RequiredConstraint  TimeGapConstraint | Constructor  isFulfilled  Constructor  isFulfilled |
| testRequiredConstraintsEmpty | IO  RequiredConstraint | readRequiredConstraints  Constructor  isFulfilled |
| testBuildingConstraintsEmpty | IO  BuildingConstraint | readBuildingConstraints  Constructor  isFulfilled |
| testTimeGapConstraintEmpty | IO  TimeGapConstraint | readTimeGapConstraint  Constructor  isFulfilled |
| testReadTimeslots | IO  Timeslot | readTimeslots  getCode  getDay |
| testReadTimeslotsAndExtractByDay | IO  Utilities  Timeslot | readTimeslots  extractTimeslotsByDay  getDay |
| testMain | Schedule | Main |
| testMainBoundByConstraint | Schedule | Main |
| testMainNoCombinations | Schedule | Main |
| testMainNoLectures | Schedule | Main |
| testMainNoTutorials | Schedule | Main |
| testMainNoArguments | Schedule | Main |

Test cases are placed in different files. AllTests.java is used to define and kick start a test suite. Unit tests and some integration tests are placed in the same java file. The rest of integration tests are placed in OtherIntegrationTests.java. The test cases of system test are placed in TestSchedule.java.

# Test Cases

|  |  |
| --- | --- |
| **Test case** | **Purpose** |
| testSameDay | Test whether two sessions are in the same day is correctly identified. |
| testConstructorAndGetters | Test whether the constructor and getters in Timeslot works. |
| testSameBuilding | Test whether two sessions are in the same building is correctly identified. |
| testToString | Test whether the toString method of Timeslot works. |
| testOverlapTrue | Test whether two sessions overlap with each other is correctly identified. |
| testOverlapFalseSameDay |
| testOverlapFalseDiffDays |
| testDifference | Test the calculation of the difference in time of two sessions is correct as well as the case of two sessions on different days. |
| testToStringTimetable | Test the toString method of Timetable works. |
| testExtractByCode | Test the extracted set of sessions are of the same course code. |
| testExtractTimeslotsByType | Test the extracted set of sessions are of the same type (lecture or tutorial). |
| testAllCourses | Test whether all course codes are retrieved correctly. |
| testPermutate | Test the method Permutate in Utilities class. |
| testPermutateArrayList | Test the method PermutateArrayList in Utilities class. |
| testGeneratePermutations | Test the method GeneratePermutations in Utilities class, given that the following methods are well-tested: extractTimeslotsByCode, extractTimeslotsByType, permu-tate and GeneratePermutations. |
| testValidateInputDuplicateCrn | Test the validateInput method in Utilities class given two ruplicated CRNs. |
| testValidateInputSession | Test the validateInput method in Utilities class given wrong and correct session codes. |
| testValidateInputBuilding | Test the validateInput method in Utilities class given wrong and correct building codes. |
| testValidateInputCourseStartTime | Test the validateInput method in Utilities class given wrong and correct course start times. |
| testValidateInputCourseFinishTime | Test the validateInput method in Utilities class given wrong and correct course finish times. |
| testValidateInputCourseTimeRange | Test the validateInput method in Utilities class given wrong and correct course time ranges. |
| testValidateInputDay | Test the validateInput method in Utilities class given wrong and correct days. |
| testWeekday | Test the getDay method in Weekday. |
| testPrintSchedule | Test whether the schedule printed on console is correct. |
| testPrintCRNs | Test whether the CRNs printed on console is correct. |
| testReadRequiredConstraints | Test whether the required constraints are read correctly. |
| testReadRequiredConstraintsFail |
| testReadRequiredConstraintsEmpty |
| testReadBuildingConstraints | Test whether the building constraints are read correctly. |
| testReadBuildingConstraintsFail |
| testReadBuildingConstraintsEmpty |
| testReadTimeGapConstraint | Test whether the required constraints are read correctly. |
| testReadTimeGapConstraintFail |
| testReadTimeGapConstraintError |
| testReadTimeGapConstraintEmpty |
| testReadTimeConstraintsDayOff | Test whether the time constraints are read correctly. |
| testReadTimeConstraintsBefore |
| testReadTimeConstraintsAfter |
| testReadTimeConstraintsBetween |
| testReadTimeConstraintsError |
| testReadTimeslots | Test whether the timeslots are read correctly. |
| testReadTimeslotsFail |
| testRequiredConstraint | Test whether the constructor of class RequiredConstraint and its isFulfilled method work as expected given that Timeslot is well-tested. |
| testTimeGapConstraint | Test whether the constructor of class TimeGapConstraint and its isFulfilled method work as expected given that Timeslot is well-tested. |
| testTimeConstraint | Test whether the constructor of class TimeConstraint and its isFulfilled method work as expected given that Timeslot is well-tested. |
| testBuildingConstraint | Test whether the constructor of class BuildingConstraint and its isFulfilled method work as expected given that Timeslot is well-tested. |
| testRequiredAndTimeGapConstraints | Integration test of RequiredConstraint and TimeGapConstraint. |
| testRequiredConstraintsEmpty | Integration test of IO and RequiredConstraint for the scenario of reading an empty input file. |
| testBuildingConstraintsEmpty | Integration test of IO and BuildingConstraint for the scenario of reading an empty input file. |
| testTimeGapConstraintEmpty | Integration test of IO and TimeGapConstraint for the scenario of reading an empty input file. |
| testReadTimeslots | Integration test of IO and Timeslot to verify whether the timeslots are read correctly. |
| testReadTimeslotsAndExtractByDay | Integration test of IO, Timeslot and Utilities to verify whether the timeslots read are extracted correctly. |
| testMain | System test by calling the main function, given different scenarios: normal, too many constraints, no possible combinations disregarding constraints, no lectures given for a course, no tutorials given for a course and no argument is given to run the program. |
| testMainBoundByConstraint |
| testMainNoCombinations |
| testMainNoLectures |
| testMainNoTutorials |
| testMainNoArguments |

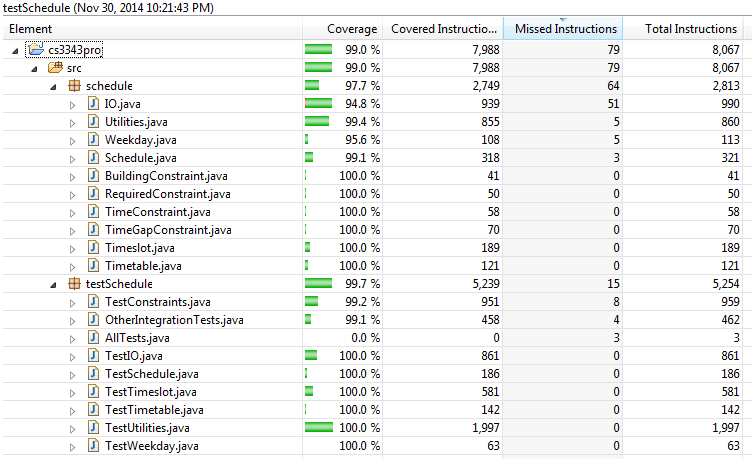
# Testing Strategy

The testing strategy used in this project is bottom-up.

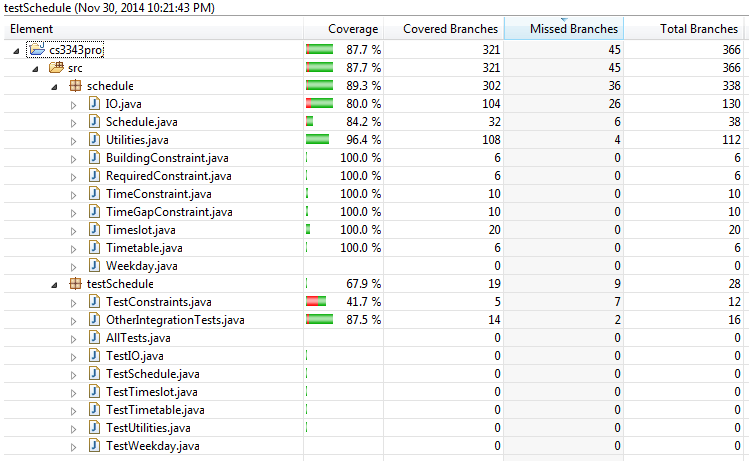
Unit tests of Timeslot and Weekday were first performed. Then the integration tests were performed after unit tests. For Utilities, Timetable, RequiredConstraint, TimeGapConstraint, TimeConstraint and BuildingConstraint classes, as they all depend on class Timeslots, the test should be performed after the unit test of Timeslot. The integration test of IO, Weekday and constraint classes are performed after testing the 4 constraint classes, as well as integrating with Timeslot and Utilities classes. The system tests were performed at last by calling the main function.

# Coverage

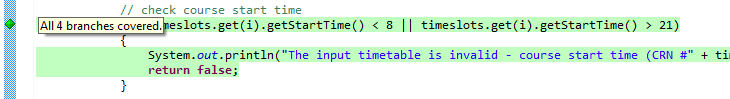
## Statement Coverage in Eclemma



## “Branch” Coverage in EclEmma

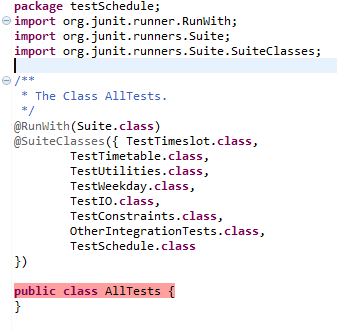


The “branch” coverage calculated by EclEmma covered more than the branch coverage defined in this course. Here is an example:



The condition is “the start time is less than 8 or greater than 21”. There are 4 possibilities in terms of condition. In this course, branch coverage is defined as the entire decision of an if-statement, i.e. 2 branches for this example.

Nevertheless, our statement coverage is high, as well as the “branch” coverage. This represents that our codes are well-tested. Please note that, the 0% coverage of AllTests.java is due to its emptiness (as illustrated in the following figure).



# Test Results

