ECSE 321 - Intro to Software Engineering
Design Specification Document - Deliverable 3

Harley Wiltzer
Camilo Garcia La Rotta
Jake Shnaidman
Robert Attard
Matthew Lesko

March 19, 2017

Contents

Part I

Unit Tests

1 Course Unit Test

METHOD UNDER TEST: Constructor
*Note: For cases in which a message, stating for a wrong input, is output from the system, then the changes/creations/deletions of objects should not persist.
Test Cases For: className Inputs

Test Cases for Constructor	className Input	cdn Input	TimeBudget Input
Test Case no.1	null	101	10.00
Test Case no.2	""	101	10.00
Test Case no.3	"1234567890"	101	10.00
Test Case no.4	" "	101	10.00

Test Cases for Constructor	Expected Output
Test Case no.1	Course name cannot be empty!
Test Case no.2	Course name cannot be empty!
Test Case no.3	(className gets saved in persistence layer)
Test Case no.4	Course name cannot be empty!

Test Cases For: cdn Inputs

Test Cases for Constructor	className Input	cdn Input	TimeBudget Input
Test Case no.5	ECSE	101 and 101	10.00
Test Case no.6	ECSE	null	10.00
Test Case no.7	ECSE	-1	10.00
Test Case no.8	ECSE	"one hundred and one"	10.00
Test Case no.9	ECSE	" "	10.00

Test Cases for Constructor	Expected Output
Test Case no.5	Cannot input Non-Unique CDN!
Test Case no.6	Cannot input empty CDN!
Test Case no.7	CDN must be positive!
Test Case no.8	Cannot input alphabetical characters for CDN!
Test Case no.9	Cannot input empty CDN!

Test Cases For: graderTimeBudget/taTimeBudget Inputs

Test Cases for Constructor	className Input	cdn Input	TimeBudget Input
Test Case no.10	ECSE	101	null
Test Case no.11	ECSE	101	""
Test Case no.12	ECSE	101	"ten"
Test Case no.13	ECSE	101	-10.00

Test Cases for Constructor	Expected Output
Test Case no.10	Cannot input empty TimeBudget!
Test Case no.11	Cannot input empty TimeBudget!
Test Case no.12	Cannot input alphabetical characters for TimeBudget!
Test Case no.13	Grader Budget must be positive! TA Budget must be positive!

METHOD UNDER TEST: addJob

Test Cases For: addJob Inputs

Test Cases	startTime	endTime	Day	Salary	Requirements	Instructor
Test Case no.14	null	null	null	null	null	null
Test Case no.15	10:00	15:00	Monday	15.00	Bachelors	Default Instructor

Test Cases	Expected Output
Test Case no.14	Invalid Input for Job!
Test Case no.15	(Job gets saved in persistence layer)

METHOD UNDER TEST: addJobAt

Test Cases For: addJobAt Inputs

Test Cases	Job1 Input	Index1 Input	Job2 Input	Index2 Input
Test Case no.16	Job1	null	Job2	null
Test Case no.17	Job1	1	Job2	1

Test Cases	Expected Output
Test Case no.16	Cannot add Job at null index!
Test Case no.17	Cannot add Job at occupied index!

METHOD UNDER TEST: addOrMoveJobAt

Test Cases For: addOrMoveJobAt Inputs

Test Cases	Job Input	Index Input	New Index Input
Test Case no.18	Job	1	2
Test Case no.19	Job	1	0

Test Cases	Expected Output
Test Case no.18	(Job gets moved to a new index successfully)
Test Case no.19	(Job gets moved to a new index successfully)

2 Job Unit Test

***Note:** Since the Classes Tutorial and Laboratory inherit from the Job class; the test cases for these two classes will be identical to the Job Class. The inputs aCourse and aInstructor are simply valid Course and Instructor inputs.

METHOD UNDER TEST: Constructor

Test Cases For: startTime and endTime Inputs

Test Cases	startTime	endTime	Day	Salary	Requirements	Course	Instructor
Test Case no.1	null	null	Monday	10.00	Bachelors	aCourse	aInstructor
Test Case no.2	""	""	Monday	10.00	Bachelors	aCourse	aInstructor
Test Case no.3	10:00	9:00	Monday	10.00	Bachelors	aCourse	aInstructor
Test Case no.4	21:00	22:00	Monday	10.00	Bachelors	aCourse	aInstructor

Test Cases	Expected Output
Test Case no.1	Input Time cannot be empty!
Test Case no.2	Input Time cannot be empty!
Test Case no.3	startTime cannot be before endTime!
Test Case no.4	Times cannot be outside working hours!

Test Cases For: Requirements Inputs

Test Cases	startTime	endTime	Day	Salary	Requirements	Course	Instructor
Test Case no.5	10:00	12:00	Monday	10.00	null	aCourse	aInstructor
Test Case no.6	10:00	12:00	Monday	10.00	""	aCourse	aInstructor
Test Case no.7	10:00	12:00	Monday	10.00	" "	aCoursee	aInstructor

Test Cases	Expected Output
Test Case no.5	Requirements cannot be empty!
Test Case no.6	Requirements cannot be empty!
Test Case no.7	Requirements cannot be empty!

Test Cases For: Course and Instructor Inputs

Test Cases	startTime	endTime	Day	Salary	Requirements	Course	Instructor
Test Case no.8	10:00	12:00	Monday	10.00	PHD Student	null	aInstructor
Test Case no.9	10:00	12:00	Monday	10.00	PHD Student	aCourse	null

Test Cases	Expected Output
Test Case no.8	Invalid Course Input!
Test Case no.9	Invalid Instructor Input!

Test Cases For: Salary Inputs

Test Cases	startTime	endTime	Day	Salary	Requirements	Course	Instructor
Test Case no.10	10:00	12:00	Monday	null	PHD Student	aCourse	aInstructor
Test Case no.11	10:00	12:00	Monday	” ”	PHD Student	aCourse	aInstructor
Test Case no.12	10:00	12:00	Monday	”ten10”	PHD Student	aCourse	aInstructor
Test Case no.13	10:00	12:00	Monday	-10	PHD Student	aCourse	aInstructor

Test Cases	Expected Output
Test Case no.10	Salary cannot be empty!
Test Case no.11	Salary cannot be empty!
Test Case no.12	Salary cannot have alphabetical characters!
Test Case no.13	Salary cannot be negative!

Test Cases For: Day Inputs

Test Cases	startTime	endTime	Day	Salary	Requirements	Course	Instructor
Test Case no.14	10:00	12:00	null	10.00	PHD Student	aCourse	aInstructor
Test Case no.15	10:00	12:00	Saturday	10.00	PHD Student	aCourse	aInstructor

Test Cases	Expected Output
Test Case no.14	Date cannot be empty!
Test Case no.15	Date cannot be a weekend day!

3 Profile

Part II

Integration Tests

The integration tests are split into two parts. One part is the Java which tests the functionality of the controllers that operate in the Desktop and Mobile application in order to test the integration of multiple entity classes. The other is the PHP which tests how numerous classes can be integrated together.

4 Java

All together, the Java code has tested that all classes function harmoniously as they are needed to within our Mobile and Desktop application.

4.1 Profile Controller

This controller tests the integration of the profile class with the courses class as well as persistence and profile manager class. This test achieved 100% code coverage using ECLEmma.

It tested for null and empty input in all fields within the controller and ensured that it was handled. It also ensured that profiles that had incorrect input were not added to the system (were not persisted in the XML)

4.2 Course Controller

This controller tests the integration of the course class with the course controller, course manager and persistence. This test achieved 99.2% code coverage.

It tested for null and empty input in all String fields in the controller and ensured that it was handled. It also checked to see that CDNs, and time budgets could not be negative. Most importantly, the integration test ensured that exceptional input did not propagate into the persistence layer.

4.3 Application Controller

This controller tests the integration of the application class with the application controller, application manager, profile manager, profile controller, persistence, course class, profile class, and job class. This test achieved 100% code coverage

It tested that null and empty input in all String fields in the controller as well as negative salaries were handled by exceptions. It also tested that these errors did not propagate into persistence.

5 PHP

5.1 Job Integration

Given that the web client is inteded to be used by an instructor, the integration testing relates the job class to all the other classes in various scenarios and use-cases by utilising their controllers. While the unit test for profile and course used some implementation of their respective controllers, here we try to use only the controllers as much as possible for all interactions with the job class

and the application controller, which handles the operations needed for the job class to be created and used.