FIT5171

Project Assignment 2

Integration Testing & Regression Testing

Yue Su 29848411
Yunzhong Liu 29052122
Kaixiang Zhang 25631179

Test Plan Identifier

This test plan is provided by the group 34.

Introduction

This report is about the second step of a project which is a Java-based Web application related to rockets. In this report, we are going to introduce the functionality we implemented and integration testing approach. At the end, an evidence of our regression testing framework will be shown as a bunch of screenshots.

Techniques and Test Strategy

Junit5**:** The major testing frameworks.

Test-Driven Development (TDD): is the major method which was being used in the testing and new function development.

First of all, we extended code based on the given code, the we started writing test. The first test has been written and then we check if it's successful, if the test successes, we start another one, but if the test fails, a production code would be produced. Then we ran all tests, if they are all success, we would start a new test, if one or more test fail, we would write production code again until all test pass.

Functionality

The main functionalities are in RocketMiner.java, there are 5 functions we added which have shown in table:

Function	Description
List<Rocket> mostLaunchedRockets(**int** k)	Sort k Rockets and return the list of k most active rockets.
List<LaunchServiceProvider> mostReliableLaunchServiceProviders(**int** k)	sort k LaunchServiceProvider then Returns the top-k most reliable launch service providers as measured by percentage of successful launches.
public String dominantCountry(String orbit)	Find the dominant country who has the most launched rockets in an orbit.
public List<Launch> mostExpensiveLaunches(**int** k)	Find the list of k most expensive launches
public List<LaunchServiceProvider>	Find the list of k launch service

highestRevenueLaunchServiceProviders(**int** k, **int** year)	providers who has the highest sales revenue

The last functionality is in Neo4jDAO.java which is

public <T **extends** Entity> **void** delete(T entity).

This function used to delete the data from the database.

Other functions we add are in the rest classes and they are all extended based on the setters. In another word, we added method to check if a parameter is null.

Testing Details

Testing Type

Type: Function testing

Manual/Automated: Automated testing.

Testing object

The object of this testing includes Two parts, RocketMiner class and Neo4jDAO test.

RocketMinerUnitTest has been changed in following:

RocketMiner Class (Testing)		
Attribute	**Testing**	**Method**
FindTopMostActiveRockets	shouldReturnTopMostActiveRockets(int k)	Use hash set load all rockets and sort them. Then check if Top Most Active Rockets has returned
MostReliableLaunchServiceProviders	shouldReturnMostReliableLaunchServiceProviders(int k)	load all launch service providers into a list then sort it according to each one's ratio. Then check if the Most Reliable Launch Service Providers has been returned
TheMostExpensiveLaunches	shouldReturnTheMostExpensiveLaunches	load all launch into a list then sort it according to each one's price. Then check if the launches are sorted by the price by comparing it to a sorted list.
TheCountryWhichHasTheMostLaunchesOnThisOrbit	shouldReturnTheCountryWhichHasTheMostLaunchesOnThisOrbit	load all launch into a list then sort it according to the certain Orbit. Then check if the most launches are in the orbit.
FinfHighestRevenueLaunchServiceProviders	shouldReturnHighestRevenueLaunchServiceProviders	load all launch service providers into a list then sort it according to each one's revenue. Then check if the Highest Revenue Launch Service Providers has been returned

In Neo4jDAOUnitTest, we add two test which are shouldDeleteLaunchServerProviderDirectly() and

shouldDeleteUserWithoutDeleteRocket(). They are used to delete Launch Server Provider and delete Rocket.

Furthermore, we add many test in other unit test classes to test if an object is null. Details are listed in following table:

Test	Description
whenEnterNullrockets()	It must have valid, non-empty name, country and launch service provider attribute values in rocket.
shouldThrowExceptionWhenSetRevenueToNull()	test if the revenue of a lsp is null
whenEnterNullLaunchOutcome()	test if the outcome of a launch is null
shouldReturnFalseWhenPasswordFormatIsWrong()	the password should have letter and number,and the size of password should from 8 to 16 characters
shouldReturnFalseWhenEmailIsNotInTheRightFormat()	when email not in the right format, for example there is no '@' or '.', return false
shouldThrowExceptionWhensetLastnameToNull()	test if the last name of an user is null
shouldThrowExceptionWhensetFirstnameToNull()	test if the first name of a user is null
whenEnterNullPrice()	test if the price of a launch is null
whenEnterNullFunction()	test if the function of a launch is null
whenEnterNullOrbit()	test if the orbit of a launch is null
whenEnterNullLaunchSited()	test if the LaunchSited of a launch is null
whenEnterNullPayload()	test if the payload of a launch is null
whenEnterNullLaunchServiceProvider()	test if the lsp of a launch is null
whenEnterNullLaunchVehicle()	test if the vehicle of a launch is null
whenEnterNullLaunchDate()	test if the lauch date of a launch is null
shouldThrowExceptionWhenNameToNull()	It must have a valid, non-empty name attribute value in Rocket.
shouldThrowExceptionWhenCountryToNull()	It must have a valid, non-empty country attribute value in Rocket.
shouldThrowExceptionWhenManufacturerToNull()	It must have a valid manufacturer value in Rocket, which points to a LaunchServiceProvider object

Testing framework

The maven test result shows in figure 1.

```
[INFO] Tests run: 18, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 3.155 s - in rockets.mining.RocketMinerUnitTest
[INFO] Running rockets.model.UserUnitTest
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.049 s - in rockets.model.UserUnitTest
[INFO] Running rockets.model.LaunchUnitTest
[INFO] Tests run: 10, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.015 s - in rockets.model.LaunchUnitTest
[INFO] Running rockets.model.RocketTest
[INFO] Tests run: 10, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.024 s - in rockets.model.RocketTest
[INFO] Running rockets.model.LaunchServiceProviderUnitTest
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.019 s - in rockets.model.LaunchServiceProviderUnitTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 64, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 28.357 s
[INFO] Finished at: 2019-04-29T00:59:39+10:00
[INFO] Final Memory: 45M/160M
```

Figure 1

Additionally, the test of RocketMinerUnitTest and Neo4jDAOUnitTest are shown below as well in figure 2 and figure 3.

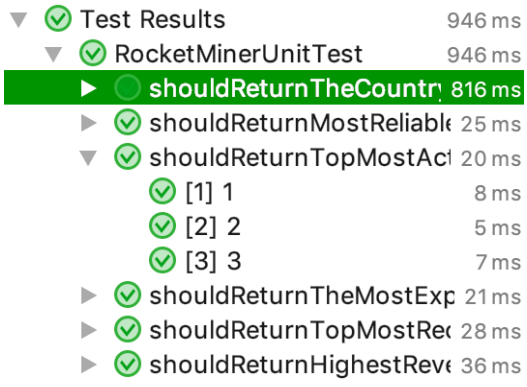

Figure 2

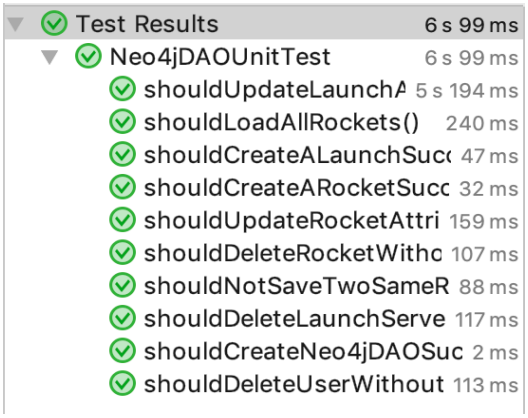

Figure 3

Team member's Contribution

Su: one TODO test in RocketMiner class and its JUnit test, tests of various set functions, Report

Victor: two TODO test in RocketMiner class and its JUnit test, Report

Kaixiang: Built database, setup Jenkins, three TODO in Neo4jDAO and RocketMiner class and its JUnit test