Lab Report for Software Engineering course
Lab 3: Starbubucks coffee online retailing system v2.0

Wang, Chen	Liu, Jiaxing	Huang, Jiani	Tang, Xinyue
16307110064	17302010049	17302010063	16307110476

School of Software
Fudan University

March 27, 2019

Contents

Chapter 1

Overview of this Lab

1.1 The Objectives of the Project

We are going to try to learn to use the code unit testing tools, e.g. JUnit/JMock, in this lab so as to experience test-driven development and the influence of the change in code quality and requirements on the process of development.

The specifications, division of work and the detailed implementation of the work is shown in the sections below.

1.2 Specifications of the Lab

In this lab, we are required to accomplish the tasks including designing and implementing new interfaces, performing unit tesing design and development and some other works. Specifically, the works are in the form of the following parts:

1. Implement the interfaces as required in the lab requirements;
2. Perform code unit testing design and develop code unit test;
3. Link the code commits with the project work items in the project planning;
4. Develop in groups based on Git and hand in the lab report.

   Our group members acted actively in their own roles together to finish the whole project and below are the detailed working results of our group.

1.3 The division of work in the team

1.3.1 Division of work: Huang, Jiani

Build the JUnit environment and write all required environmental methods except the *@test* method, such as *@before* and *@after*; write the *@test* method for the *login* method.

1.3.2 Division of work: Liu, Jiaxing

Write the *checkName* and *checkPassword* test method.

1.3.3 Division of work: Wang, Chen

Complete the *checkName* and *checkPassword* methods.

1.3.4 Division of work: Tang, Xinyue

Finish the test method of *signUp*, *checkStatus* and *cost*.

Chapter 2

Tools adopted for quality analysis

2.1 Junit

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks which is collectively known as xUnit that originated with SUnit.

JUnit is linked as a JAR at compile-time; the framework resides under package junit.framework for JUnit 3.8 and earlier, and under package org.junit for JUnit 4 and later.

2.2 JMock

JMock is a library that supports test-driven development of Java code with mock objects.

Mock objects help you design and test the interactions between the objects in your programs.

The jMock library:

1. makes it quick and easy to define mock objects, so you don't break the rhythm of programming.
2. lets you precisely specify the interactions between your objects, reducing the brittleness of your tests.
3. works well with the autocompletion and refactoring features of your IDE
4. plugs into your favourite test framework is easy to extend.

Chapter 3

Features added to the project

3.1 User name checking

3.1.1 Requirements for user name checking

The username should satisfy the following requirements:

1. The username must start with **starbb_**;
2. The username can consist of **letters**, **numbers** and **underline**, excluding any other symbols;
3. The username should have a length greater than or equal to 8 and less than 50.

3.1.2 Interface for the checking method

```
/**                                                    1
 * Check whether the given name is valid              2
 *                                                    3
 * @param name the given name to check                4
 * @return whether the name is valid                  5
 */                                                   6
boolean checkName(String name);                       7
```

3.2 Password checking

3.2.1 Requirements for user name checking

The password should satisfy the following requirements:

1. The password can consist of **letters**, **numbers** and _, excluding any other symbols;
2. The password must consist of all the three types, i.e. **letters**, **numbers** and _, excluding any other symbols;

3. The password should have a length greater than or equal to 8 and less than 100.

3.2.2 Interface for the checking method

```
/**                                                    1
 * Check whether the given password is valid           2
 *                                                     3
 * @param password the given password to check         4
 * @return whether the password is valid               5
 */                                                    6
boolean checkPassword(String password);                7
```

Chapter 4

Features tested in the project

4.1 Login method

The whole test for login can be divided into two functions: the one for login successfully and the other for login failure.

4.1.1 Login successfully

assertTrue method is used in this test function.

4.1.2 Login failed

assertEquals method is used in this test function. If we fail to login, a run-time exception will be thrown. Therefore, we should compare the message of exception with the expected string.

4.2 Sign up method

4.3 Username checking method

4.4 Password checking method

4.5 Status checking method

4.6 Cost checking method

Bibliography

[1] Wikipedia contributors. (2018, December 24). Version control. In *Wikipedia, The Free Encyclopedia*. Retrieved 06:12, March 10, 2019, from `https://en.wikipedia.org/w/index.php?title=Version_control&oldid=875227317`

[2] Wikipedia contributors. (2019, March 10). Systems development life cycle. In *Wikipedia, The Free Encyclopedia*. Retrieved 06:13, March 10, 2019, from `https://en.wikipedia.org/w/index.php?title=Systems_development_life_cycle&oldid=887015682`

[3] Stolen, L. H. (1999). Distributed control system. *international telecommunications energy conference.*

[4] Murayama, T. (1991). Distributed Control System. *international conference on advanced robotics robots in unstructured environments.*

[5] Wikipedia contributors. (2019, March 6). Distributed control system. In *Wikipedia, The Free Encyclopedia*. Retrieved 06:18, March 10, 2019, from `https://en.wikipedia.org/w/index.php?title=Distributed_control_system&oldid=886468871`