|  |
| --- |
| CSCI 45200 – IUPUI SPRING 2014 |
| The Healthy People Pharmacy System |
| Assignment 4 – Quality Assurance |
|  |
| **Naeem Tai** |
| **4/23/2014** |

|  |
| --- |
| This document is prepared while performing Quality Assurance for the Healthy People Pharmacy system. This document contains the list of test cases, discussion of the role of a test framework, and discussion of Test-Driven Development.. |

# List of Test Cases

* testOperator
* testSubtotal
* testTax
* testTotal
* testChangeDue

# Role of a Test Framework

Test Framework – JUnit (xUnit)

We used the JUnit test framework to test the Register class of our system. The Register class is the most important class in our system. It is a source of user interaction, and all input and output for the system is managed through this class. For the prototype created in the Software Design phase, we tested four main functions, which actually cover the whole system interaction, of the Register class.

We performed the test by creating only one instance of the register class in the RegisterTester, JUnit Test Case, class. In this class, we also created all the necessary instances such as items and an employee. We assigned this employee to the register, performed a sale by scanning items, and made a payment. We manually calculated the subtotal, tax, total, and change due for a constant amount tendered. We then invoked methods of the register class and checked if they return the same values as we manually calculated. Our test cases included checking for assigned operator to the register, subtotal for the current sale on the register, tax for the current sale on the register, total for the current sale on the register, and change due after making cash payment for the current sale on the register.

In this iteration, we performed very simple tests by performing all the calculations manually prior to executing the tests. In the next iteration, we plan to make our test class more complex by repeatedly executing tests and programmatically setting the desired outcomes instead of performing endless manual calculations. This will allow us to quickly test our system in multiple scenarios.

# Test Driven Development

In this phase of the software development process of our system, we developed the system by following the Test-Driven Development process. We iteratively developed the system by writing and executing one test case at a time. After writing each test case, we immediately executed the test case and expected it to fail. After it failed, we developed all the necessary components to make the test case succeed, and executed the test case until it succeeded. We performed this routine for all five test cases we identified to test our system in this iteration. We also ordered the test cases in terms of their complexity, so that each test case would require us to further develop the system. By doing this, we made sure that our system is able to pass the required tests. As a side effect, we also prevented ourselves from writing any unnecessary code.

By following the Test-Driven Development process, we were confident and satisfied with our development of the system. This is because we first wrote the test cases, and then developed the system accordingly. As developers, we are confident that our code works as expected under the circumstances defined by the test cases.

The resulted system of the Test-Driven Development was the same as the system we developed in the Software Design phase earlier. In our case, there was no difference between two developments. We only needed to provide a get method in the Register class to retrieve the current sale. We used this method in the RegisterTester class when testing subtotal, tax, and total for the current sale. This method was only necessary to properly execute the test cases related to the current sale. Aside from the tester class, we would not need this method for the final deployment of the system.

Overall, our experience with Test-Driven Development was positive. It forced us to actually write quality test cases. Additionally, it granted us satisfaction with our code. In the end, we were confident that our system was able to pass the test cases because we had developed the system accordingly and executed the test cases until our system pass them.