

SENG 438 - Software Testing, Reliability, and Quality

Lab. Report #1 – Introduction to Testing and Defect Tracking

Group #: 3

Student Names:

Haniya Ahmed

Apostolos Scondrianis

Beau McCartney

Josh Vanderstoop

Table of Contents

- 1 Introduction 1
- 2 High-level description of the exploratory testing plan 1
- 3 Comparison of exploratory and manual functional testing 1
- 4 Notes and discussion of the peer reviews of defect reports 1
- 5 How the pair testing was managed and team work/effort was divided 1
- 6 Difficulties encountered, challenges overcome, and lessons learned 1
- 7 Comments/feedback on the lab and lab document itself 1

Introduction

The objective of this lab focuses on four main goals: familiarizing ourselves with the SUT and the defect tracking system, exploratory testing, manual scripted testing, and regression testing using an ATM software program versions 1.0 and 1.1. This will assist in understanding some fundamentals of testing through hands-on experience and also understand the difference between different types of testing. The entire process will implement industrial defect tracking systems, processes, and practices.

Prior to this lab, our group had only the introductory knowledge of the three different testing types introduced in this lab and no prior industrial experience. The lab allowed practical application of all our understandings while also allowing us an opportunity to experience BackLog and all its uses in defect tracking.

High-level description of the exploratory testing plan

Group 1:

The outline for high level requirements provides an extensive list of properties that this system must contain. These range from simple transaction methods to pseudo-confirmations with the bank's internal systems. To explore this program and test its functionality, we will begin by conducting tests that we would regularly do

when interacting with a real ATM. In doing so, the functionality that is required by consumers will be tested thoroughly. Some of these tests will include: Withdrawals from all account types Deposits to all account types Transfers between all account type Amount inquiries for all account types. Through these tests, we aim to test most every function available to the user, and compare the results from what our expectations would be while using an ATM, as well as the expectations outlined in the requirements section. Another form of testing that will be done is from the perspective of an employee of the banking company, who would aim to ensure that this system is following procedures correctly and will not produce a loss of assets for the bank or consumers. This will be mainly completed by monitoring the transaction log and receipts printed after each of the above tests are completed.

Group 2:

Our plan is to initially test if we are able to access our banking account using both card #s. We will attempt to insert invalid and valid credentials and observe how the system reacts. We will then test a deposit, to see if the money will be deposited properly. We will do that on all 3 types of account, Checking, Savings, Money Market, check the balances of the accounts, and see if they match. For the checking account, we will check and see if when deposits or withdrawals are made on one card, if they show up on both cards, i.e. check to see if actually both cards connect to the same checking account. Then we will test each transaction type, Withdrawal, Deposit, Transfer, Balance Inquiry. In between, of each transaction we will ask for a balance, retrieve card, and reinsert and see if the balances match the expected result. We will also check to see if we attempt to withdraw more money than the atm holds, what the result will be. We will test and see if the logs match the transactions. The full testing process is described in our repository by the following txt file :

<https://github.com/seng438-winter-2022/seng438-a1-Apostolos-sc/blob/main/Group%20testing%20-%20Notes%20for%202nd%20group%20of%20group%203/ATMv1.0-Exploratory-Testing-Pair-2.txt>

Comparison of exploratory and manual functional testing

Exploratory testing is a software testing technique performed without prior planning or documentation, and therefore does not require a test script. Contrastingly, manual functional testing is a software testing technique that is a little more rigorous and involves pre-designed test cases for different functionalities. Throughout our lab, our exploratory testing was a little more spontaneous and free-ranged; however, we did not find defects that differentiated much from the manual scripted testing, but rather, greater ranges of the defects found. Exploratory testing definitely allowed more in-depth understanding of the errors and allowed us to notice how some defects affect other parts of the program.

Notes and discussion of the peer reviews of defect reports

Once the pairs had completed their respective testing sessions, our group came together to combine the results. Using a Google Doc we wrote in all of the bugs we had found and removed the duplicates, then one member from each pairing entered the information into Backlog to create our final results.

How the pair testing was managed and team work/effort was divided

For Josh and Haniya's group, we divided the work by alternating through the manual test cases. Haniya took the last 20 test cases, and Josh took the first 20, and then the other would spend time making notes about any defects noticed. Then we spent over 30 minutes making a generic plan of our exploratory testing by revisiting the requirements of the software program, and then testing for errors together with one person sharing the screen.

For Apostolos and Beau... As a group, we had a few meetings and came with a plan on how to attempt to perform exploratory testing on the software provided to us. We came with 23 tests for our exploratory testing plan. Apostolos took care of the initial testing phase performing the 23 exploratory tests and Beau performed the regression testing on all 23 exploratory tests and 40 tests suggested by the manual script testing suite provided to us. Furthermore, we saved our results into a text file which we shared with Josh and Haniya's group. Then Josh and Haniya took care of the backlog inputting. We looked over at the backlog and entries that Haniya and Josh inputted after looking at our test results and their test results, and together with beau we did some editing on the entries themselves. Then we worked together on the report and filled out what we could.

Difficulties encountered, challenges overcome, and lessons learned

In pairs:

Haniya and Josh: This assignment definitely allowed us to spend the time overcoming the learning curve of working together and working with BackLog. Josh spent time testing the test cases on version 1.0 and I spent time writing details about defects. However, I didn't write directly into the BackLog as we went which added extra time overcoming the learning curve of working with BackLog and further extended the amount of time spent on completing the defect report.

Apostolos and Beau : This assignment had a few challenges. First one was understanding the assignment document that was given to us. It was lengthy and the exploratory testing took us more than 30 minutes. In addition getting familiar with the backlog was challenging and we realized during the demo that the TA would have liked all of us to have created entries in the backlog software. Next time we will make sure to have a single meeting where we explain to each other what was added and why in the backlog and get everyone to add some entries in it. In addition, we figure that we need to be more concise and follow the same system when creating a new entry in the backlog and when performing testing and logging our results.. This will help us be more organized, communicate effectively with the rest of the team and complete the next assignment using a better workflow.

Together: We did not encounter many difficulties when coming together, it allowed us to be able to analyze and understand some defects better and improve our defect report. However, task division and revision of manual and exploratory testing could have been done more efficiently to avoid redundant and unnecessary revision. We also figured out we can do batch updates which saved a lot of time!

Comments/feedback on the lab and lab document itself

Josh: The lab was well designed and linear in its completion, however the formatting of the lab document resulted in some minor confusion for myself as to the expectations of exploratory testing.

Haniya: It was interesting operating with a software product designed by someone else to specifically test for defects. I definitely think that the lab document could have been a little bit less ambiguous to save a lot of time spent on figuring out details, but I hope the lessons learned from this lab can be applied to the next.

Apostolos: The lab was a bit unambiguous at some points and it would have been nicer to see the tests for manual scripting appear in the section where we are asked to perform the manual test scripting instead of the references. Maybe for the demo next time we can have the TA create zoom breakout sessions that are self assigned and have the title of our groups, that way if we crash we are able to rejoin without the TA having to accept us into the meeting and without having another 20 students listening to our demo which was pretty stressful.

Beau: The last few test cases were literally impossible to carry out, because you can't attempt transactions until you enter the correct pin for the card. For the demonstration, the document stated each group member stated one failing and one successful test case. There was no mention of the TA asking for specific scenarios, or choosing the tests we demo. If we're going to be interviewed, please state so in the future.