**Testing Document and Specification**

Test Plan

Commerce Project – Group 4

CS 451R

|  |  |  |
| --- | --- | --- |
| Lara, Harrison R. | Williams, Daudi | Hubbard, Madison |
|  |  |  |

# Introduction

This document outlines the test plan for the University of Missouri-Kansas City Computer Science 451R Commerce Project. As outlined in the project Requirements Document, this system needs to provide a medium for Commerce Bank customers to utilize their accounts online. This system will also provide a mechanism for users to add transactions, create accounts, notifications about a specific account and export account data. The testing activities discussed in this document will verify that the software for the Commerce Bank project meets the needs of the customer by verifying that the requirements for this system, as outlined in the Requirements Document, are met.

The testing routine will test the Commerce Bank project systems response to creating a user account, the user being able to login, navigating to any account detail page, being able to add new transactions and view current account data. The user should be able to export all the data for an account and create a notification based off a series of rules. The results of this testing procedure will enable the creators of this system to gauge project success as outlined in the Project Plan.

# Terminology

Throughout this document the terms user, detail page, site and username/password will be used frequently therefore, formal definitions will be given.

User- people affiliated with Commerce Bank as a customer and will exclusively use only the front-end of the website

Detail Page – this is a page which is dedicated to displaying a specific accounts information, data grid, ability to export and add transactions to the specific account.

Site- the front-end website (main page)

Username/Password- unique identifiers that authenticate and validates a user

# Items Tested

Items that will be tested during the testing phase as laid out by the Project Plan will be but are not limited to:

* Ability for a user to register/ create an account

Test Case: 4.1.2

* Ability for a user to login via username and password

Test Case: 4.1.1

* Ability for a user to view their dashboard/ account snapshots and navigate to an account detail page

Test Case: 4.2.1, 4.3.1

* Ability for a user to navigate to detail page and view all their account data and page, filter, sort their transaction history

Test Case: 4.3.1

* Ability to export all transaction data from a detail page

Test Case: 4.3.3

* Ability to add a new transaction to an account via the detail page

Test Case: 4.3.2

* Ability to navigate to the notifications page and view preferences

Test Case: 4.4.1

* Ability to create, edit and delete a notification rule then trigger the notification so it appears on the dashboard

Test Case: 4.4.1, 4.3.1

* Ability to remove a notification from the dashboard

Test Case: 4.2.1

* Ability for a user to logout from anywhere in the application or via the side navigation menu

Test Case: 4.2.1

# Items Not Tested

There are features that will not be included in the current testing procedure. This does not mean that these features will not be implemented, but that they have not been implemented and are not available for testing. Those features include but are not limited to:

* Administrative Features that include adding/removing a user
* Ability to update user profile and personal information
* Querying the database

# Approach

The overall method to this testing procedure is manual system testing. Each test case created will have a direct link to the requirements as laid out in the Requirements Document. Test cases that include similar Feature methods will be tested together. Test cases such as these test the security features of the system along with the ability to submit information to the database. Each test case will test the security feature with invalid and valid data (usernames and passwords) to ensure that user requirement of valid users for these features is met.

The features that specify adding a transaction and export all account data will be tested separately and together to confirm all data is accurately reported back from the database in a quick succession. Ability to filter, sort and page transactions will verify the system’s ability to produce results that an individual can validate visually.

Manual system testing will continue throughout everyone iteration of the project. For each iteration, both old and newly implemented features will be tested. Adding new features or functionality can sometimes interfere with the functionality of old features and to ensure product/project success, all features implemented should function as intended throughout the life of the software.

# Item Pass/Fail Criteria

The minimum requirements for this software system were laid out in the Requirements Document and the Project Plan outlined what the creators of the software considered project success.

Implemented features that meet the requirements as determined by the customer, meaning the feature does what the user wants it to do with very little difficulty, passes the testing procedure. Difficulty, as used here, is determined by user comprehension and user ability to use the feature with little to no training.

Features that contain major defects will fail the testing procedure and will be documented via an incident report and turned over to the developer for investigation and revision.

# Test Deliverables

In addition to the Test Plan, other test deliverables include the Test Specification which outlines the specific test cases and expected results of each test, and Test reports which is comprised of Incidents, Defects and Changes.

# Testing Tasks

The following list the testing deliverables and the activities required to produce the deliverable.

|  |  |
| --- | --- |
| **Deliverables** | **Activities** |
| **Test Plan** | * Analyze Requirements for System Features * Determine Testable/Non-Testable Features * Develop Approach/Method for testing * Determine Task and Estimate Efforts * Develop Schedule for Testing |
| **Test Specifications** | * Analyze Requirements * Define Test Cases for Testable Features as Outlined by the Test Plan |
| **Test reports** | * Implement Test Cases as Outlined by the Test Specifications * Document Incidents and Defects * Determine Severity of Incidents and Defects * Determine Changes that Need to be Made to System * Document and Submit Change Request to Developer |