**Testing Document and Specification**

Test Plan

Group 1, Commerce Bank Project

CS 451R 2021

|  |  |  |
| --- | --- | --- |
| Acharya, Sushant | Kenney, Kole | Loewen, Ozzie |
| Starr-Timberlake, Thomas |  |  |

# Introduction

This document is for the benefit of Commerce Bank to help and detail what is being tested in the Web Application being created for their bank in terms of a transaction site. Previously stated in the Requirements Document, this system is aimed at providing Commerce Bank and their customers with an application that allows them to track, list, and edit their banking transactions and notifications. The testing activities discussed in this document will verify that the software for Transactions Application as well as the Notifications created for Commerce Bank, meeting the requirements laid out in the Requirements Document.

The testing routine will test the interactions that occur on the Transaction Application, the system’s response to adding and editing transaction, registering a new user, receiving notifications, and editing notification settings. The results of this testing procedure will enable the creators of this system to gauge project success outlined in the definition of success in previous documents.

# Terminology

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

System- the database and bulletin board that supports the backend of the website

User- In this case users are Commerce Banking members as well as group members who test the system.

Customer- These are people who have a an account on our Application to use the Transactions and Notifications.

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 Macro Project Plan will be but are not limited to:

* The ability for the user to Register a New User into the Transaction Web Application.

Test Case: 4.1.1.1, 4.1.1.2

* Ability for a user to login via username and password

Test Case: 4.1.1.2

* Ability for a user to login and create a Transaction.

Test Case: 4.1.2.1

* Ability user to login and edit a Transaction.

Test Case: 4.1.2.2

* Ability for the user to add a new Notification.

Test Case: 4.1.3.1

* Ability for the user to edit a Notification.

Test Case: 4.1.3.2

* Ability for the user to update their Profile.

Test Case: 4.1.5.1, 4.1.5.2, 4.1.5.3,

# 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, they have been implemented using the trial-and-error methodology, rather than a methodical testing plan. These features are items that do not vary as far as input, such as the items listed below:

* The CSS design aspects
* Querying the database
* Home Menu links

# Approach

The overall method to this testing procedure is manual system testing. Each Test Case was created based on what was plotted through both designing the application and the use of the application. Test cases that include similar Feature methods will be tested together. This would mean features regarding manage profile settings, the transaction page features, the notifications features, as well as log in features. Test cases such as these test the database manipulation as well as the security of the account. 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 feature responsible for the Manage Profile settings will be the most complex and time-consuming testing. This will have to do with the validity of the password and login as well as the changing of this password. Since the password is what allows access to the user’s account, this will need to have the most rigorous testing process.

Manual system testing has been a priority from iteration one, and the creation of the application. Allowing the user to be recognized and testing for the account settings. This testing process will continue through all iteration as to keep the system up to date and not create an abundance of technical debt. 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 from being discussed with the creators of the software.

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. These fails are also discussed in ‘Touch Base’ meetings and developer workshops to ensure this was the best approach or if something else should be tried.

# Test Deliverables

In addition to the Test Plan, other test deliverables include the Test Cases which outlines the specific test cases and expected results of each test, and Test Incident 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 * Add comprehensive notes for future developers to reference whilst PR-ing. |
| **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 * Review Plan, Specifications, to ensure the report contains the information for next phase of development. |