**Testing Document and Specification**

Test Plan

Corner Club

CS 451R, Fall 2021

|  |  |  |
| --- | --- | --- |
| McNall, Wes | Halderwood, Zach | Williams, Ryan |
| Chacey, Ian | Nguyen, Jason | McNish, Colin |

# Introduction

This document outlines the test plan for the Commerce Bank Online Banking application designed by the Corner Club development team. As outlined in the project Requirements Document, this system needs to provide a medium for Commerce Bank customers to view and interact with their bank account’s transactions through the web browser. This system will also provide a mechanism for users to be informed about new transactions based on custom rulesets. The testing activities discussed in this document will verify that the software system 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 online banking system’s response to viewing, filtering, sorting, adding, downloading, and deleting banking account transaction data. The results of this testing procedure will enable the creators of this system to gauge project success as outlined in the Macro Project Plan.

# Terminology

Throughout this document the terms system, user, site, email/password, account, and profile will be used frequently, therefore, formal definitions will be given.

System – the database that supports the backend of the website

User – an individual accessing the web application; whether this refers to an authenticated user depends on the context of the test case.

Site – the front-end website

Email/Password – unique identifiers that authenticate and validate a user

Account – a Commerce Bank customer’s bank account from which all online banking financial information is

Profile – a validated user’s online banking service which is unique to that 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:

* Ability for a user to register and login via email and password and use their online banking service

Test Case: 4.1.1, 4.1.2

* Ability for a user to view, search, add, delete, page, download, and manipulate transaction history data

Test Case: 4.2.1, 4.2.2, 4.2.3, 4.2.4

* Ability for users to change their login credentials securely

Test Case: 4.3.1

# 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. Examples of these features include registering for a new online banking login, logging in to view transaction history, etc. Test cases such as these test the security features of the system along with the ability to view and 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 viewing/manipulating transaction info will be tested together but in separate test cases. Test cases surrounding transaction info will produce results that an individual can validate visually.

Manual system testing will continue throughout the second and third 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 Macro 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 |