

# Testing Document and Specification

## Test Plan

ID: TestPlan1  
Sponsor Project Group 6  
CS 451, FS 2019

Salay, Tarik

Tran, David

Wolfe, Jonathan

## Introduction

This document outlines the test plan for the Online Banking Application created by Group 6 of CS451R. As outlined in the project Requirements Document (BankReq1), this system will be used as an online banking portal where users can view their recent transactions, get alerts to certain transactions that match business rules, and acquire reports data based on the alerts. The testing activities discussed in this document will verify that the software for the Online Banking Application meets the needs of the customer, as outlined in the Requirements Document(BankReq1).

User testing will involve logging in and out of users, removing/creating rules, and switching between bank accounts. System and database testing will involve how the system stores and clears session info from the website and the ability to collect and store info from the database and present it on the website. The results of this testing procedure will help gauge project success as outlined in the Project Plan (Com13).

## Terminology

Throughout this document the terms user, system, site, username/password, Session Info, Alert, Transaction, Rule and Report will be used frequently therefore, formal definitions will be given.

System- the database and web application that supports the backend of the website

User- bank customers who will exclusively use only the front-end of the website

Site- the front-end website consisting of all pages in the application

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

Session Info - Information about the user saved between pages within the site while the user is accessing the site

Alert - A message to a user about a transaction property that the user has marked as important and should be notified

Transaction - A monetary change to a bank account from a 3rd party

Rule - A criteria determining when a transaction should create an alert

Report - A collection of alerts within a designated time frame

## Items Tested

Items that will be tested during the testing phase as laid out by the Project Plan(Com13) will be but are not limited to:

- Ability for a user to login via username/password and save session info  
Test Case: 1.1
- Ability for a user to logout and clear session info  
Test Case: 1.2
- Ability for a user to create and remove rules  
Test Case: 2.1, 2.2
- Ability for a user to export a report and view its information  
Test Case: 3.1
- Ability for the system to fill an HTML table using SQL database  
Test Case: 4.1
- Ability for a user to switch between banking accounts  
Test Case: 5.1
- Ability for the system to timeout the user  
Test Case: 6.1
- Ability for the system to generate alerts based off transactions  
Test Case: 7.1
- Ability to test incorrect input (date range, input field size)  
Test Case: 8.1

## 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:

- Registering/removing new users
- Interface to add transactions

## **Approach**

The overall method to this testing procedure is manual system testing. Each test case relates to a component of a feature laid out in the Requirements Document(BankReq1). Test cases that include similar Feature methods will be broken into components and tested individually and together. For example, features including logging in to change rules, logging in to export a report, logging in to check transactions will be divided into logging in to the site, changing rules, export a report and checking transactions. The pass/fail will be tested for each case. Tests like changing rules, etc. require the logging in test case but the pass/fail for logging in will not be tested when testing cases like changing rules.

Manual system testing will occur during the final iteration once the application is complete with all features. This is not ideal and testing should be done on a regular basis but with an application this small, the effect of an error can be easily corrected without much cost.

## **Item Pass/Fail Criteria**

The minimum requirements for this software system were laid out in the Requirements Document(BankReq1) and the Project Plan(Com13) 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(TestPlan1), other test deliverables include the Test Specification(TestSpec1) which outlines the specific test cases and expected results of each test, and Test Reports(TestRep1) 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
<b>Test Plan</b>	<ul style="list-style-type: none"><li>• Analyze Requirements for System Features</li><li>• Determine Testable/Non-Testable Features</li><li>• Develop Approach/Method for testing</li><li>• Determine Task and Estimate Efforts</li><li>• Develop Schedule for Testing</li></ul>
<b>Test Specifications</b>	<ul style="list-style-type: none"><li>• Analyze Requirements</li><li>• Define Test Cases for Testable Features as Outlined by the Test Plan</li></ul>
<b>Test Reports</b>	<ul style="list-style-type: none"><li>• Implement Test Cases as Outlined by the Test Specifications</li><li>• Document Incidents and Defects</li><li>• Determine Severity of Incidents and Defects</li><li>• Determine Changes that Need to be Made to System</li><li>• Document and Submit Change Request to Developer</li></ul>