

<http://sce2.umkc.edu/BIT/burris/cs451/test-plan.html>

Test Case Specification

Team 6

April 7, 2017

Prepared by:
Jeremy Troshynski
Emily Thomas

Table of Contents

1. Introduction 8
2. Test Cases: Banking Application 8

Revision History

Version	Date	Name	Description
1	4/7/17	Jeremy Troshynski	Initial Document
2	4/12/17	Emily Thomas	Updated test cases

Introduction

This document provides the test cases to be carried out for the Commerce Bank Application. Each item to be tested is represented by an individual test case. Each case details the input and expected outputs.

1. Test Cases: Overview

Test ID	Overview 1
Title	Data for current spending is accurately displayed in the donut chart
Feature	Concurrent data on Transactions page and Overview page
Objective	Compare the totals for each category on each page. Test for accuracy
Setup	The application is running correctly from the information in the database
Test Data	All transactions for a user
Test Actions	<ol style="list-style-type: none">1. Start Application2. Add values for each category3. Compare transactions with overview
Expected Results	Values are accurate between transaction categories and overview page

Test ID	Overview 2
Title	Data for badges earned is accurately displayed in the sidebar
Feature	Concurrent data on Badges page and Overview page
Objective	Compare the badges earned on both pages. Test for accuracy

Setup	The application is running correctly from the information in the database
Test Data	All badges for a user
Test Actions	<ol style="list-style-type: none"> 1. Start Application 2. Add values for each category 3. Compare transactions with overview
Expected Results	Values are accurate between badges and overview page

Test ID	Overview 3
Title	Current goal data is accurately displayed
Feature	Concurrent data on Goals page and Overview page
Objective	Compare the goals on both pages. Test for accuracy
Setup	The application is running correctly from the information in the database
Test Data	All goals for a user
Test Actions	<ol style="list-style-type: none"> 1. Start Application 2. Record values of current goals 3. Compare goals with overview
Expected Results	Values are accurate between goals and overview page

Test ID	Overview 4
Title	Current account balance is accurately displayed
Feature	Accurate data for the account balances from the database
Objective	Compare the data from the database and the overview page
Setup	The application is running correctly from the information in the database
Test Data	All accounts and account data for a user
Test Actions	<ol style="list-style-type: none"> 1. Start Application 2. Record values of current goals 3. Compare goals with overview
Expected Results	Values are accurate between database and overview page

2. Test Cases: Badges

Test ID	Badges 1
---------	----------

Title	User earns the correct badge for an achievement
Feature	Earn a badge as a user
Objective	Confirm that the user correctly received the badge, and the badge displays on all relevant pages
Setup	The application is running correctly from the information in the database
Test Data	<Badge Earned>
Test Actions	1. Start Application 2. Earn Badge 3. Navigate to the Badges page to view newly earned badge
Expected Results	The new badge is shown on the badges page, as well as the overview page

Test ID	Badges 2
Title	User views badges earned
Feature	List of past badges for viewer to see
Objective	Confirm that all badges earned appear on the badges page
Setup	The application is running correctly from the information in the database
Test Data	<All Badges Earned>
Test Actions	1. Start Application 2. Navigate to the Badges page to view all earned badges
Expected Results	The user is able to see all earned badges on the badges page and overview page (testers may want to keep personal list of sample set of badges earned and check it against all the badges that appear from the database)

3. Test Cases: Savings/Goal Page

Test ID	Savings 1
Title	User adds a new savings goal
Feature	A new savings goal is added to the list of savings goals
Objective	Confirm that the savings goal has been added with all the correct information
Setup	The application is adding information from a form to the database
Test Data	<The savings goal information in the form and in the database>
Test Actions	1. Start Bank Application

	<ol style="list-style-type: none"> 2. Navigate to the Goals page 3. Click on Add New Goal button 4. Fill out the Goal form 5. Submit the Form
Expected Results	The user should see the new savings goal added to the list of goals with the same goal information as filled out in the form

Test ID	Savings 2
Title	User edits a savings goal
Feature	The savings goal will be updated correctly with the new information
Objective	Confirm that the savings goal has been updated with the correct information
Setup	The application is adding information from a form to the database
Test Data	<The savings goal information in the form and in the database>
Test Actions	<ol style="list-style-type: none"> 6. Start Bank Application 7. Navigate to the Goals page 8. Click on Edit Goal button 9. Edit the Goal form 10. Submit the Form
Expected Results	The user should see the new savings goal information added to the list of goals with the same goal information as filled out in the form

4. Test Cases: Trends Page

Test ID	Trends 1
Title	Verify graphical integrity
Feature	Trends page correctly displays transaction data in a graphical format
Objective	Check the graphical representation of the transaction data and verify that it is correctly categorizing and plotting the data
Setup	The application is correctly adding information from a form to database
Test Data	All transactions and categories for a user
Test Actions	<ol style="list-style-type: none"> 1. Start the Application 2. Navigate to the trends page 3. Verify the graph is representing the data correctly for the given transactions within a category

Expected Results	The graph should correctly display the data for each given transaction
------------------	--

5. Test Cases: Transactions Page

Test ID	Transactions 1
Title	Marking of over budget transactions
Feature	When a user goes over-budget, a transaction becomes marked as red.
Objective	Confirm that all transactions that are over-budget for the month are marked as red
Setup	The application is running correctly from the information in the database
Test Data	All transactions for a user
Test Actions	<ol style="list-style-type: none"> 1. Log into the application 2. Compare budget for a category with transactions for that category
Expected Results	All transactions that are past the budgeting constraints are marked as red

Test ID	Transactions 2
Title	Altering a transaction category
Feature	Selecting a transaction and editing its category
Objective	Confirm that the user is able to edit a category associated with a transaction
Setup	The application is running correctly from the information in the database
Test Data	All transactions and categories for a user
Test Actions	<ol style="list-style-type: none"> 1. Log into the application 2. Select a transaction 3. Edit the transaction category 4. Refresh the page and verify the change has occurred
Expected Results	The transaction should keep its edited category and display that in all future visits to the transaction page

Test ID	Transactions 3
Title	Filtering by all transaction categories

Feature	Selecting a category and viewing all transactions from that category
Objective	Confirm proper filtering
Setup	The application is running correctly from the information in the database
Test Data	All transactions and categories for a user
Test Actions	<ol style="list-style-type: none"> 1. Log into the application 2. Select a transaction category 3. Click Filter 4. Verify correct transactions are displayed 5. Repeat for all other category filters
Expected Results	The transactions table should display all transactions for a particular category.

6. Test Cases: Budget Page

Test ID	Budget 1
Title	Editing a budget category
Feature	A user should be able to select a budget category and edit all features, such as the name, description, and amount.
Objective	Confirm that all data for the budget category is altered correctly
Setup	The application is running correctly from the information in the database
Test Data	All budget categories for a user
Test Actions	<ol style="list-style-type: none"> 1. Log into the application 2. Click Edit for a budget category 3. Alter all information for the category 4. Save the changes and verify the changes were applied correctly
Expected Results	The altered budget category should display the expected changes

Test ID	Budget 2
Title	Adding a budget category
Feature	A user should be able to add a new budget category to their budget
Objective	Confirm that all data for the budget category is saved to the application
Setup	The application is running correctly from the information in the database

Test Data	All budget categories for a user
Test Actions	<ol style="list-style-type: none">1. Log into the application2. Click Add New Category3. Enter in all information for the category5. Save the changes and verify the changes were applied correctly
Expected Results	The new budget category should display all of the entered information

