Testing Document and Specification

Test Reports

Commerce Bank Group 3
CS 451R

Reed Heinrich
Connor Larson

Landon Volkmann
Tayler Singleton

Samuel Feye

Introduction

This document outlines the outcome of completed system tests and unit tests. Incidents, defects, and problems we ran into will be listed here.

Incidents

This section defines the incidents discovered while performing various tests on the application. Some defects or problems were not discovered during testing

Incident ID	1
Description	Error message kept popping up when attempting to start the application.
Originator	Tayler Singleton while writing the User Guide
Discover Date	April 24, 2020
Severity	Low
Steps Required to Produce Incident	1. Open terminal 2. Run "npm start" in the backend 3. Run "npm start" in the normal terminal 4. Application will crash
Responder	Landon Volkmann, Samuel Feye - developers
Current Status	Closed
Cause	The backend development branch had updated the database, but the "Development" branch was trying to be used while its records indicated that the database still contained the outdated data, when it did not. Threw error.
Resolution	Updated database and development branch with said database data.
Addressed Date	April 24, 2020
Creation Phase	Development
Detection Phase	Creating User Guide
Correction Time	10 mins

Incident ID	Test Case 3.2-A
Description	Created a new account with new sign in information, added new transactions to this new account, and it continued the transaction ID numbering from the previous account, i.e. the first account left off at ID = 3, and when a new transaction was added on the new account, it started at 4.

Originator	Reed Heinrich -Tester
Discover Date	April 26, 2020
Severity	Low
Steps Required to Produce Incident	1. Create an account with a sign in and password 2. Add any number of transactions (in this case, 3) 3. Logout 4. Create new account with sign in and password 5. Add new transactions, and the transaction ID will start the number at 4.
Responder	Landon Volkmann, Samuel Feye - Developers
Current Status	At time of submission of test plan, OPEN
Cause	The data is being sent to the same database column, so according to the database, it’s just another transaction, not in a separate entity/account.
Resolution	Currently discussing resolution, but one method would be to hide the transaction ID from the user and keep it in the backend. Another would be a lot harder, to try and keep transaction ID’s in a separate table or location for each user, but in the end this would probably not be worth it, but this is an ongoing discussion.
Addressed Date	TBD
Creation Phase	Development
Detection Phase	Iteration 5 Testing
Correction Time	TBD

Incident ID	3
Description	Error message pops up while trying to boot up application with new API implementation. Hook errors with React.
Originator	Landon Volkmann
Discover Date	April 11th, 2020
Severity	Low-Medium
Steps Required to Produce Incident	1. Attempt to boot up application.
Responder	Samuel Feye, Connor Larson - developers
Current Status	Closed
Cause	We were attempting to call the dispatch in the same component that needs that information to render.
Resolution	Tried dispatching the call in the dashboard, then passed the data to notifications.
Addressed Date	April 24, 2020
Creation Phase	Development Iteration 4
Detection Phase	Development Iteration 4
Correction Time	1+ hour

Defects

Most defects were found as they were implemented and fixed on the spot, or at least during that iteration. We did a great job as a team of heading into a new iteration with little to no defects to our knowledge. Any defects that were found were handled and taken care of. To our knowledge now, there are no defects that harm the application's success.

Summary

The testing procedure to date has been conducted using manual system testing and concurrent unit testing. Features tested include but are not limited to:
Unit Testing (found in "test" folder in project):

- Unit Test 1: Makes sure that the correct data was loaded into the database using the .sql files. Specifically, this unit test makes sure all Account info is loaded in correctly
  - Test function: public void test_select_acc_info()
- Unit Test 2: Makes sure that the setTransaction call works in the backend and that all necessary data is loaded correctly.
  - Test function: public void test_set_transaction()
- Unit Test 3: Assures that notification triggers' type and description are inserted correctly.
  - Test function: public void test_notification_trigger_descr()
- Unit Test 4: Assures that notification triggers are actually created and all necessary values are in place.
  - Test function: public void test_create_notification_trigger()

Login Page

- Ability for a user to login via email and password
  - Test Case: 1.1
- Ability for a user to sign up and create their own account through the login page
  - Test Case: 1.2
- Ability for a user to save their email+password combination for next time
  - Test Case: 1.1
- Ability for a user to hit forgot password and have a refresh link sent to their email
  - Test Case: 1.3

- On launch, the Login page is correctly displayed.
    - Test Case: 1.1
- Password is masked and not visible to users
    - Test Case: 1.1

Dashboard

- Upon login, user is taken to dashboard
    - Test Case: 2.1
- Dashboard displays both the “Notifications” section at the top and the “Transactions” section below it
    - Test Case: 2.1
- From the dashboard/home screen, a user has the ability to select the “Transaction” tab and the “Notification Settings” tab on the left side of the screen
    - Test Case: 2.1
- Functional “Logout” button is visible at the top right corner of the screen
    - Test Case: 2.1

Transactions

- When a user clicks on the “Transactions” tab, the transactions list is displayed
    - Test Case: 3.1
- User has the ability to sort transactions by date - either by newest or oldest
    - Test Case: 3.2
- User has the option to turn on/off Dense Padding, which scales down the list to fit more on the screen.
    - Test Case: 3.2
- Each header is displayed correctly above the right columns: ID, Date, Amount, Type of Charge, Balance, and Description
    - Test Case: 3.1
- Each row is selectable - user has ability to select one or all rows and delete the transaction
    - Test Case: 3.2
- User has the ability to search for a transaction in the search field, and they can select which column they are searching through
    - Test Case: 3.2
- User has the ability to add a transaction
    - Test Case: 3.2
- User has the ability to export the transactions to a .csv file
    - Test Case: 3.3
- User has the ability to select the amount of rows they want displayed in one page (default 5)
    - Test Case: 3.2

Notifications

- When a user clicks on the “Notification Settings” tab, they are taken to the Notification Settings screen
    - Test Case: 4.1
- The functionality is the same for all four sections (transaction, balance, description, recurring alert): when the user clicks the “add” button, they are prompted to enter a value (if an amount, it’s a number, if it’s a description, it’s a word/some words)
    - Test Case: 4.1
- If a user no longer wants to receive a type of notification, they have the ability to delete a notification by clicking the ‘X’
    - Test Case: 4.2

Further system testing will be done as needed as we implement some new features during this final milestone. Luckily, these changes are mostly cosmetic and adding a few tweaks here and there, so ideally a majority of testing has been completed. We have one ongoing incident that will likely be resolved very soon.