

Testing Document and Specification

Test Specifications

ID: TestSpec1
Sponsor Project Group 6
CS 451, FS 2019

Salay, Tarik

Tran, David

Wolfe, Jonathan

Test Case ID:	1.1
Title:	Authenticating a User with Username & Password
Feature/Subfeature:	System User Validation
Purpose:	To ensure that only persons that are customers of the bank are able to access their account information.
Initial Conditions:	User must currently be on the Online Banking website and be able to input information into the login fields.
Test Data:	Test Data will include invalid usernames like georgeWash and AbeLincoln and invalid passwords as in 123456789 and 098765432. Valid usernames and passwords will also be used but will not be listed here for security and confidentiality reasons.
Test Actions:	1. Go to the login page of the Online Banking website 2. Enter username and password 3. Select "Login"
Expected Results:	After step 3, if valid username and password are entered, user can login to the main page and view their banking information. If invalid data has been entered, a warning of invalid entry appears and step 2 should be repeated.

Test Case ID:	1.2
Title:	User Logout and Clear Session Info
Feature/Subfeature:	System User Validation
Purpose:	To ensure that bank customers can log out of the website to protect their information.
Initial Conditions:	User can be on any page within the website. The "Logout" button is available in the top right corner of the website.
Test Data:	Test Data will include a valid user login that can access the website to test the logout feature.
Test Actions:	1. After logging in, go to any page within the website. 2. View the "Logout" button in the top right. 3. Click on the "Logout" button to be logged off of the website and return back to the login screen.
Expected Results:	After step 3, the user should be back to the original login screen of the banking application and pressing the back button will not return to the prior page.

Test Case ID:	2.1
Title:	User Creates a Rule
Feature/Subfeature:	Business Rule Creation
Purpose:	To ensure that users can set up and modify rules that would be used to create alerts.
Initial Conditions:	User must have an idea of a rule to create that would be set up to generate alerts based off of transactions.
Test Data:	Category: Rules Title: Transaction Category Question: Do I want a rule that would set up an alert based off a specific category (high amount spent, certain date/time, out of area transactions, etc.) ?
Test Actions:	1. From any page in the website, the user navigates to the “Rules” page by clicking it on the navbar. 2. The user decides which rule to enable. 3. Fill in the corresponding input field (if available), then hit the corresponding checkbox. 4. Press “Apply” to apply these changes to the rule. 5. Now the alerts table will update based on the updates to the rules.
Expected Results:	After step 5, when the user returns to the alerts or home page, the alerts should be updated to match the rules that were updated in the “Rules” page.

Test Case ID:	2.2
Title:	User Removes a Rule
Feature/Subfeature:	Business Rule Removal
Purpose:	To ensure that users can modify and remove rules that would be used to create alerts.
Initial Conditions:	User must have an idea of a rule to remove that would change the generation of alerts based off of transactions.
Test Data:	Category: Rules Title: Transaction Category Question: Do I want to remove an existing rule that would set up an alert based off a specific category (high amount spent, certain date/time, out of area transactions, etc.) ?
Test Actions:	1. From any page in the website, the user navigates to the “Rules” page by clicking it on the navbar. 2. The user decides which rule to disable. 3. Uncheck the corresponding checkbox. 4. Press “Apply” to apply these changes to the rule. 5. Now the alerts table will update based on the updates to the rules.

Expected Results:	After step 5, when the user returns to the alerts or home page, the alerts should be updated to match the rules that were updated in the “Rules” page.
--------------------------	--

Test Case ID:	3.1
Title:	Exporting a Report and Viewing the Information
Feature/Subfeature:	Reports Information
Purpose:	To ensure that users are able to view reports (that are generated every month). These reports contain the alerts that were triggered within that month.
Initial Conditions:	User wants to view the alerts that were triggered within a certain month.
Test Data:	Category: Reports Title: Viewing Reports Question: Do I want to see a detailed list of the alerts that were triggered for a specific month?
Test Actions:	1. From any page in the website, the user navigates to the “Reports” page by clicking it on the navbar. 2. The user decides which month of reports to view. 3. Click the “Export” next to the report that the user wants to view detailed information about. 4. Save or open the attached excel document to view the information.
Expected Results:	After step 3, the user should be prompted with the ability to save or open the excel document that contains the alerts data for that month. After step 4, the file that was attached should contain the related transaction information that triggered a rule for that month.

Test Case ID:	4.1
Title:	Read from Database into HTML Table
Feature/Subfeature:	System Function
Purpose:	To ensure that the system can fill in the tables of the website with the corresponding information from the database.
Initial Conditions:	Database tables are filled with relevant data. The system is set up correctly to read from the database and transfer the information.
Test Data:	The database should be filled with information that can be displayed in a table within the website. The website should be correctly attached to the database.
Test Actions:	1. The developer should run test function calls in the code. 2. The system should be able to access the database and return the data that was called.

Expected Results:	After step 2, the system returns the corresponding database information that would be used by developer to display within the website.
--------------------------	--

Test Case ID:	5.1
Title:	Switching between Banking Accounts
Feature/Subfeature:	Multiple Account Support
Purpose:	To ensure that the user can view their account and other accounts (if owned). This also ensures that the system is correctly handling multiple accounts.
Initial Conditions:	User has another account setup. Multiple banking accounts should be linked to the same user for this test case.
Test Data:	Category: User Title: Checking Another Account Question: Am I able to access information from my other accounts?
Test Actions:	1. From any of the pages on the website, the user sees the dropdown to change between accounts. 2. The user decides which account they would like to view information for. 3. Click on the dropdown. 4. Select the account that the user would like to see. 5. Now information on that page should be correlated to that banking account that has been selected by the dropdown.
Expected Results:	After step 3, the system should display options that include all the bank accounts connected to the user's login. After step 5, the tables on the page should update to the other account's information.

Test Case ID:	6.1
Title:	Timing Out the User
Feature/Subfeature:	System Function (Timeout)
Purpose:	To ensure that the customer account is safe by ending the session after five minutes
Initial Conditions:	User must be already logged in and have not interacted with the system for 300 seconds by any terms
Test Data:	Customer account
Test Actions:	1. The user logs into their account. 2. After logging in, the user could be on any page in the application. 3. After 5 minutes of inactivity, the system will kick the user back to the login page.

Expected Results:	After step 2, the system will start counting each time there is no action and when the count reaches 300 seconds, it will automatically log out the user.
--------------------------	---

Test Case ID:	7.1
Title:	Generating Alerts based off Transactions
Feature/Subfeature:	Database Function (Alert Generation)
Purpose:	To ensure that the database is correctly creating the requested alerts based off of the user's rules.
Initial Conditions:	Database must be filled with transactions that can trigger rules. The user has to have rules set up that would create alerts.
Test Data:	Database: Transactions that become alerts based off rules. Rules: User sets rules that would create alerts.
Test Actions:	<ol style="list-style-type: none"> 1. From the "Rules" page, the user chooses rules (view test case 2.1 and 2.2) that are appropriate for alerts. 2. Click "Apply" on the "Rules" page once the user has finished updating the rules. 3. The database should automatically produce the alerts data that would be displayed.. 4. View "Alerts" or "Home" to see the alerts that should be created based off the set rules.
Expected Results:	After step 3, database calls events that would fill the alerts table with transactions based off rules.

Test Case ID:	8.1
Title:	Incorrect Input
Feature/Subfeature:	System Validation (Date Range, Input Field Size, etc.)
Purpose:	To ensure that the system is receiving correct and valid information to avoid logical errors or errors that could break the system.
Initial Conditions:	The developer should know what valid input should be, for example, the "amount" for a transactions is an Int32, so the amount should not exceed the Int32 limit.
Test Data:	Test Invalid Input: Negative Numbers, Numbers > Int32
Test Actions:	When retrieving information about transactions from the bank, the system should check that the information is valid before committing it to the database.
Expected Results:	If input is invalid, throw an error message and re-request the information.