|  |  |
| --- | --- |
| **Project Name: Budgeting App** | |
| **Test Case** | |
| **Test Case ID:** N/A | **Test Designed by:** Jonathan Pinder |
| **Test Priority (Low/Medium/High):** High | **Test Designed date:** 11/25/2017 |
| **Module Name: App Buttons, Login** | **Test Executed by:** Noah Lone  **Module Testing by**: Samina Hirani |
| **Test Title:** Verify buttons work and database connects | **Test Execution date:** 11/27/2017  **Retested on:** 11/29/17 |
| **Description:** Testing all of the buttons and making sure they run the queries when clicked. |  |
|  |  |
|  |  |
| **Pre-conditions:** Downloaded Github files, MSSQL running, and Eclipse | |
| **Dependencies:** | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Launching the app/Signing Up | Opening app | User should be able to sign-up with username and password, and then log in. | All data input from the user’s SignUp page is added to the database respectively. | Pass |  |
| 2 | Log In to app | Button response | User should log in successfully and the Main Page GUI should pop up. | Button was responsive. User was taken to the Main Page | Pass |  |
| 3 | Salary/Savings Goal button | Button response and query execution | The update button is clicked and the input data is sent to the database | The data is sent to the username’s database. |  |  |
| 4 | Expenses Button/Window | Button Response | The expenses button should create a column in the user’s database that is named after the user’s input. They will then enter the cost of that expense which will be sent to the database | The button works successfully. | Pass |  |
| 5 | Display Button | Button response | The display button should pull the data from the database and display it in the window to the right. | The button displays the user’s details. | Pass |  |
|  |  |  |  |  |  | ` |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes/Pre-conditions** |
| 6 | Leave “Username” text box on “User Login” page as blank and click “Login” | Login error message on “User Login” page | The message should be displayed as “Login Error” | The message displayed as “Login Error” | Pass |  |
| 7 | Enter an invalid value (ex :1234) in “Username” text box | Login error message on “User Login” page | The message should be displayed as “Login Error” | The message displayed as “Login Error” | Pass | Created a Username of invalid value(ex:1234) before logging in |
|  |  | User’s name should be displayed along with his current saving goal and his saving for that year | The message should be displayed as “Hello alex  This is your current yearly salary: 1000.0  This is your current savings goal: 50.0  This is how much you can save a year :600.0” | The message displayed as “Hello alex  This is your current yearly salary: 1000.0  This is your current savings goal: 50.0  This is how much you can save a year :600.0” | Pass | Created a username (ex: alex) and entered the salary (ex:1000) and savings goal (ex: 50). Clicked on “Expenses” button and entered name of expense (ex: Internet) and expense (ex:50) |
| 8 | Click “Update” button and “Display” button on “Main” page. |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Post-conditions:** Errors can occur with each button depending on the user’s input. Entering letters into fields that are supposed to take in numbers can cause errors. The Display button should also display the user’s expenses, but it does not do that right now.