Kester Moore 1911247

Manual Tests

Test #1

**Test Purpose**: Test if Workout API Data was being called and stored correctly

**Actions Performed**: Created the URI and inserted link to API

**Expected Outcome**: Fields get the data

**Actual Outcome**: Error, the API seems to be having issues in getting the necessary information

Test #2

**Test Purpose** Test Workout API again

**Actions Performed**: Modified URL and updated API Key

**Expected Outcome:** Fields get the data

**Actual Outcome**: Error Provided, data from the API still not being accessed correctly

Test #3

**Test Purpose**: Test Crypto API

**Actions Performed**: Switched to Crypto code and coded in the URI and called the URL without a key(not needed)

**Expected Outcome:**  Fields get the data

**Actual Outcome**: Fields get the data successfully

Test #4

**Test Purpose**: Test if data from the API is stored correctly in the database created

**Actions Performed**: Coded in the database code and wrote an if statement based upon the lab code to call upon the database or the web API

**Expected Outcome:**  Crypto data is stored in the database

**Actual Outcome**: Crypto data is stored in the database

Test #5

**Test Purpose**: Test StockList

**Actions Performed**: Coded in calls in displayStockData

**Expected Outcome:**  Replace the TextField with the json version of the data

**Actual Outcome**: Caused a Crash

Test #6

**Test Purpose**: Test StockList again

**Actions Performed**: Recoded and reviewed with lab material

**Expected Outcome:**  Replace the TextField with the json version of the data

**Actual Outcome**: Data has been replaced

Test #7

**Test Purpose**: Test APIParser to see if data is being processed correctly

**Actions Performed**: Coded Parser and called it in displayStockData

**Expected Outcome:**  Parses the data that passes through it

**Actual Outcome**: Seems to have worked correctly

Test #8

**Test Purpose**: Test StockRepository

**Actions Performed**: Coded Repository based upon lab material

**Expected Outcome:**  Everything works and all methods execute as expected

**Actual Outcome**: Methods execute as expected

Test #9

**Test Purpose**:  Final Test of Database and API

**Actions Performed**: Made small changes to the calling of the API and how the database stores data.

**Expected Outcome:**  All data gets saved correctly and the download of the API works correctly and shows the data collected

**Actual Outcome**: Data is shown correctly, however one parameter symbol seems to pull a null every time

Test #10

**Test Purpose**:  Cause of the Null

**Actions Performed**: Searched through the code and found a typo

**Expected Outcome:**  All data gets saved correctly and the download of the API works correctly and shows the data collected

**Actual Outcome**: Data is shown correctly and no null values are displayed

Test #11

**Test Purpose**:  Test new pages created

**Actions Performed**: Created new pages to make the finding of the crypto currency the focus of the API.

**Expected Outcome: Pages are accessible easily**

**Actual Outcome**: New pages are accessible

Test #12

**Test Purpose**: Test if the Goals List stores the user input correctly

**Actions Performed**: Coded in the storing and the getText uses

**Expected Outcome:**  User inputted goal gets shown in the List

**Actual Outcome**: Crash is caused and then continuously crashes

Test #13

**Test Purpose**: Test if the Goals List stores the user input correctly

**Actions Performed**: Replaced TextView with EditView

**Expected Outcome:**  User inputted goal gets shown the List

**Actual Outcome**: User input is now shown

Test #13

**Test Purpose**: Test if Workout Planner page stores user input correctly

**Actions Performed**: Carried over code from the Home Page and changed it to suit the Workout Page

**Expected Outcome: Stores and displays the users input**

**Actual Outcome**: User input is displayed but requires fixes to the string spacing

Test #14

**Test Purpose**: Test if the Workout List String spacing issues are fixed

**Actions Performed**: Fixed the toString methods

**Expected Outcome:**  User inputted goal gets shown in the List

**Actual Outcome**: User input is saved and shown correctly

Test #14

**Test Purpose**: Test if the Calorie Calculator gets the user inputs

**Actions Performed**: Coded in the user input collection once the button is clicked

**Expected Outcome:**  User inputs get taken

**Actual Outcome**: User inputs get taken

Test #15

**Test Purpose**: Test the Calorie Calculator math with the user inputs

**Actions Performed**: Coded in the math to calculate the calories and incorporated the user inputs

**Expected Outcome:**  User get shown caloric intake necessary once the button is pressed

**Actual Outcome**: Crash is caused

Test #16

**Test Purpose**: Test the Calorie Calculator math with the user inputs

**Actions Performed**: Converted all numerical values into ints

**Expected Outcome:**  User inputted numbers can calculated with the equation

**Actual Outcome**: Text Field is replaced with amount of calories needed

Test #17

**Test Purpose**: Test the currency conversion through settings

**Actions Performed**: Tried calling the Cryptos method of changing currencies

**Expected Outcome:**  Change the amount of the Price from USD to selected currency

**Actual Outcome**: Crash is caused. After many tries this feature has been removed

Test #18

**Test Purpose**: Test Overall navigation, storing, showing and API calling of the app

**Actions Performed**: Checked through xml and java for any errors

**Expected Outcome:**  Navigation works, data is stored correctly and showed and the API calls are correct

**Actual Outcome**: Navigation works, data is mostly stored and showed and the API calls are correct

Test #19

**Test Purpose**: Test Overall navigation, storing, showing and API calling of the app

**Actions Performed**: Changed referencing of certain fields and xml placement

**Expected Outcome:**  Navigation works, data is stored correctly and showed and the API calls are correct

**Actual Outcome**: Application kept terminating but after a data wipe and a cold boot the AVD began to work. Navigation works, data is stored and showed and the API calls are correct