Bryan Jimenez

Assignment 4 Design

To begin testing procedures, I will start assessing my application with unit tests. Using JUnit testing that is native in the Android Studio IDE, I will be building the algorithms that will execute all computation while testing the functionality of the code as it is developed to ensure that all errors are handled before integrating any other modules to interact with the computations. This JUnit unit testing can also be applied heavily when developing the Optical Character Recognition (OCR) module as I am new to the concept and want to ensure the correct functionality is implemented to function properly with the rest of the system. Once these modules have been tested, I will continue to test the functionality of all input boxes on all activity screens to ensure that the input is restricted to only letter characters in the ‘Name’ fields and double numbers in the ‘Price’ fields. The last functionality to be tested in unit testing is the ‘MyTab’ screen which mainly test the ‘Edit’ menu to ensure that entries are managed properly. To continue from the unit testing stage I will be taking a bottoms-up approach to ensure that the interfaces will be transferring and displaying the correct information. Before going to test the API’s interaction with the system, I will ensure that when the system saves the information from the input data structures to the ‘MyTab’ data structures properly. This will be the focus of the overall app functionality, as well as saving the ‘MyTab’ data to the device. Once this is completed, system testing will begin. At this stage, I will be testing the system with the interface as my focus by primarily going through multiple test runs through the activities which can be seen in the UI Mockup Diagrams and State Chart Diagrams attached. Security will not be so much of a focus as the information being handled is discrete and therefore does not require extensive security measures if it can be saved directly to the phone. The app will not necessarily need to be connected to server if the OCR function does not require login information and have the software initially be an input calculator that can store entries to a device. If the circumstances allow it, I will work on getting the information onto the server to communicate with other users. An even greater reach would be to invest in getting a direct Venmo integration that allows users to pay through the app.

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality Tested | Inputs | Expected Outputs | Actual Outputs |
| User Input Data Struct. | Multiple tests of entering different names to see if data is handled to match each person to its item and each item to its price | Data saves in different, easily accessible entries that allow the user later access |  |
| User Input Fields | Entries with multiple names, names with similar structure, etc. | Each person matches creates different entry in data structure |  |
| Calculation of individual totals | Multiple double values as well as the items that were split between two or more people | Algorithms effectively calculate all items between people to bring the correct totals with all circumstances considered |  |
| Add Entry in MyTab | Multiple entries for people from the initial use. Entries that are added after a calculation has been completed. | Saves all entries during an individual total calculation as well as maintaining running totals to names listed in MyTab |  |
| Delete Entry in MyTab | Select names to remove from MyTab | Data updates successfully, MyTab updated. |  |