

Integration Testing

Praveen Chekuri, Pradnya Pisal, Kartik Bhatnagar, Bill Fung, Zac Brown

We focused on the incremental, top-down approach for the Manage Menu design. This was a feasible approach for us since it organized the modules into separate entities and allowed us to check for data flow between each module. We focused on the testing of highest level modules first and worked our way down with the use of stubs for low-level modules.

The main manage menu interface contains three buttons:

1) Add Item 2) Print Menu 3) Remove Item

If the user selects “Add Item”, the link navigates to another screen. There are several text fields for the user to fill in (item name, item description, item price). To test this part of the module, we enter data in individual text fields (Ex. Cheeseburger, Tasty, \$3.50). We check for correctness of functionality by referring to the database. The item “Cheeseburger” and its properties appear in the database, since that is what the ‘Add Item’ function was implemented to do. Later on, we hope to insert more fields such as “Food Type” and create a drop down of “Ingredients List”. We are in the middle of implementation for both of these functions except they are not fully functioning yet.

Separate modules were designed, also, to automatically generate an item ID and a URL for each item. While our implementation of these methods does not take into account of adding this information into the database, we did manage to test for its functionalities.

When we ran our “Add Item” module on an Eclipse platform, an item ID of 100007 was created for item “Cheeseburger”. At the moment, food type is initialized to “Burgers” for testing purposes. Our goal is for the system to automatically generate the correct output by referring to the database and doing a quick sort to locate the specific food type and do a count of the items listed under that food type. In our case, we had already stored six other items under food type “Burgers” with item IDs from 100001-100006.

Also, for “Create URL” a separate function is called to automate a URL based on the food item added. Basically, the URL is to generate a “www.google.com/search?q=” LINK where the name of the food is inserted shortly after. For instance, when item “Cheeseburger” was added to the list, we test for the functionality of the code by checking that the URL appears (which it

does). It creates a URL for cheeseburger: "www.google.com/search?q=Cheeseburger". Our next goal for demo 2 is to have the URL and item ID inserted directly into the database.

Another module that we designed is "Print Menu" where its screen displays a list of items contained in the database. While this is the main functionality of this module, our implementation for the time being only focuses on printing out the recently added menu items on the screen. To test data correctness, the item "Cheeseburger" does successfully appear on the screen. The user, then, has the option to 1) "Edit Item" information or 2) "Remove Item" from database. For demo one, the edit module has not been implemented. However, a separate module is designed for "Remove Item". When the user selects "Remove Item" button next to the item "Cheeseburger", this should call method "Remove Item". This can easily be tested by referring to the database; likewise the item "Cheeseburger" does no longer exist. One thing that we are still working on is to get the "Print Menu" page to refresh each time an item is removed from the database. At the moment, item "Cheeseburger" is still being displayed on the list even though it is no longer in the database. However, if user tries to remove the same item twice, the system will prompt a message saying that item "Cheeseburger" is no longer in the database.

Lastly, the third button on the main Manage Menu interface is also the "Remove Button". During the actual integration testing, we happened to encounter bugs that still need to be fixed. However, the procedures for integration testing are as follows. If the user chooses to remove item right after the item has been added, it can do so without having to go to the print menu. Please keep in mind that our main purpose of "Print Menu" is to display ALL items stored in the database. So, this separate button for "Remove Item" will come in handy when the user does not have to scroll through all the items when it can instead remove the most recently added item. This functionality was tested shortly after item "Pizza" was added. While "Print Menu" displays (Cheeseburger, Pizza) and the option to remove either item, the "Remove Item" button found on the main page will know to remove item "Pizza" from the database. Also, if user tries to remove "Pizza" again from "Print Menu", the system will prompt a message saying that item "Pizza" is no longer in the database.

Unfortunately, there are several other bugs that are not being mentioned but have come to our attention and we hope to eradicate them in time for our next demo.