Unit testing for inventory.java:

Using an SQL database modifier to access jdbc:mysql://resturantdb.cul7akmhbeku.us-west-2.rds.amazonaws.com:3306/menudb","root","password", the table “ingredients” can be accessed and modified. Then, inventory.java can be run independently, and changes that were made in the database should be reflected in the listings. Similarly, adjustments made in the program should be reflected in the database.

Unit testing for addInventory.java:

Using an SQL database modifier to access jdbc:mysql://resturantdb.cul7akmhbeku.us-west-2.rds.amazonaws.com:3306/menudb","root","password", the table “ingredients” can be accessed. Then, addInventory.java can be run independently, a new item can be added, and then the database can be accessed to ensure that the new item was added to the ingredients table.

Unit testing for menuStats.java:

Using an SQL database modifier to access jdbc:mysql://resturantdb.cul7akmhbeku.us-west-2.rds.amazonaws.com:3306/menudb","root","password", the table “timedb” can be accessed and modified. Then, menuStats.java can be run independently, and changes that were made in the database should be reflected in the listings.

Unit testing for Kitchen:

Run demo as described in README1.txt, but select “yes” when asked to run with unit testing.

Unit testing for viewBudget.java:

As commented in the code within the “Confirm Edit” button’s actionperformed function, when changing the values of amount budgeted and amount spent, the array holding the values are updated with the new values. Instead of just displaying the values from the user input, the values are first updated in the array and are retrieved through the corresponding array index in order to perform the subtraction operation needed for amount remaining. This means that the value displayed in the fourth column of the table depends on the values updating in the array correctly. Run the viewBudget.java file on its own to test.

Unit testing for editMenu.java

I added multiply items to the menu to see if the added items were reflected on the waiters menu, and the chef's menu page. Also looked at the menu table in MySQL too see if the items were added to the table. I then edited a few of the items on the current menu, and then checked the MySQL table to see if the changes occurred. Lastly for removing items, I first removed one item, and then again checked to see if the waiter and chef menu page reflected this, and then looked into the MySQL table to see if the item was removed. I then removed more than one item at a time, I removed 3 at once, too see again if the changes occurred on the waiter and chef menu page. I also then checked the MySQL table to see if the multiple items were removed from the table as well.

Unit testing for server receiving completed order notification from chef.

To test to see that the server received the information that the chef was done with a specific order. I made the server print out “Success”. Whenever the chef clicked “done” on an order, and the server receives the information from the chef.