The Leftovers
Part 3

Nicholas Chiapputo, Khaemon Edwards, Caleb Halter,
Jacob Robbins, Ryan Vanek, Saidat Babatunde,
Ephraim Emilimor, Jordan Simmons

April 19, 2020

1 Introduction

Our restaurant management system is hosted through a live server at [1]. The home page of the website has collapsible sections that we originally used to test the backend functionality while the frontend was being developed. We have left it there to more easily showcase each of the functionalities of the system. Some things, such as messaging and editing items, are not as easy to show, however. To get to any of the four user screens, simply use the menu bar. The kitchen option will take you directly to the single kitchen page where orders are queried every 5 seconds. The manager option will take you to a login screen for managers and similarly for the server option. If you would like to test logging in, please use the employees tab on the homepage to query the employee list. Retrieve the employee list under the "Get Employees" subheading. The login for an employee uses the "_id" value as the username and the four-digit pin as a password.

When selecting the customer option, you will be taken to a login screen for the table. The idea behind this is that the restaurant management or staff will set up the systems in the morning (akin to turning the registers on). They will then set the system to the appropriate table. After the table is logged in (using ID values from 1 to 20, inclusively), the customer is unable to go back to the table login page. This prevents a customer from ordering for another table.

The source files for our project can be found at [2] under the `src/` subdirectory. This is a combination of the client and backend sources. Some packages (e.g., formidable) may be necessary to install using `npm` to run correctly. Additionally, our site is using MongoDB to control database operations and NodeJS to interface between client requests and the backend operations.

Our unit tests were performed using JEST, a JavaScript testing framework. This was done through `npm` using some non-default packages including `frisby`. These can all be tested repeatedly to ensure functionality. There may be some difficulties with creating and editing menu items as these require file uploads. The test scripts themselves have a reference to one of the images included in our source files. So, if the hierarchy is maintained, all 80 unit tests will likely pass barring any unforeseen local issues.

2 Validation Tests

Attached following this description is a table of all validation tests we carried out.To the best of our ability, we tested that each and every one of these tests passed with expected results. Admittedly, there may be some unknown issues regarding non-ideal inputs that we have yet to find and fix. However, when considering ideal inputs, we believe that all 49 of these validation tests pass correctly. Below is a list of each validation test, the corresponding requirement from our requirements document, a description of the testing steps, the expected output, the comments regarding the testing, and whether or not it passed or failed at the time of submission.

Test Case No.	Traces to Requirement No.	Test Steps	Expected Output	Comments	Pass/ Fail
1	1.1	1. From a Table screen, select order 2. Select a menu item 3. Make a substitution in the item options 4. Add the item to the order	The menu item in the order will show the requested substitution	The table can choose from a selection of options provided from the manager	Pass
2	1.2	1. From a Table screen, select order 2. Click on more info button on menu item	The price, nutrition information, allergens, and an image for the selected menu item will be displayed	The customers have to click on a menu item and it will display all the information on said item	Pass
3	1.3	1. Have an ingredient have 0 amount in inventory 2. From a Table screen, select order	The menu items that require the out of stock ingredient will not be shown	The menu item is not sent to the client as determined on the back-end.	Pass
4	1.4	1. From a Table screen, have orders previously placed that contains a drink and entree 2. Select Pay	A credit of the cost of the drink will be added to the order	Works as expected	Pass
5	1.5	1. From a Table screen, select order 2. Select a menu item 3. Write a note in note section 4. Add the item to the order	The menu item in the order will have a note section with the written note	The customer is able to leave a note to the kitchen	Pass
6	1.6	1. From a Table screen, have an order pending with items in it 2. Select send to kitchen 3. Confirm	The kitchen will have all the information in the order added to an order in their list	This feature works as expected	Pass
7	1.7	1. From a Table screen, have an order pending with items in it 2. Select send to kitchen	A prompt will be shown to the customer to confirm their order	This feature works as expected	Pass
8	1.8	1. From a Table screen, have an order pending with items in it 2. Have the system time be between 11:30pm - 8:00am	An error will show telling the customer an order can not be placed	Works as expected	Pass

		3. Select send to kitchen			
9	1.9	1. From a Table screen, have an order pending with items in it 2. Select send to kitchen 3. Confirm	The ingredients database will decrease by all the ingredients requested in the order	Works as expected	Pass
10	2	1. From a Table screen, select Games 2. Select a game	The selected game will start No Help, Refill, or Order Buttons will be displayed	Works as expected	Pass
11	3	1.Be on any Table screen not in kids mode 2.Select Help 3.Select Server	A Server will receive a notification to come to the aid of the customer	Table presses help and the server assigned to them receives a message	Pass
12	3	1. Be on any Table screen not in kids mode 2. Select Refill	A Server will receive a notification that the table is requesting a refill	Table is informed server has been notified and server receives a notification that the table is requesting a refill	Pass
13	4	1. From a Table screen, have orders previously placed 2. Select Pay 3. Select Payment type 4. Process Payment	The payment will be accepted and the table marked as paid	Customers are notified of a completed payment.	Pass
14	4.1	1. From a Table screen, have orders previously placed 2. Select Pay 3. Enter a valid coupon code in coupon section	The price for the order will be discounted	Successfully discounts items.	Pass
15	4.2	1. From a Table screen, have orders previously placed that contains a kids meal 2. Have the time be after 4:00 on Monday 3. Select Pay	A credit of the cost of the kids meal will be added to the order	Kid's items added to the order within the time will have their price reduced to $0.	Pass
16	4.3	1. From a Table screen, have	The order total will	Tip amount is	Pass

		orders previously placed 2. Select Pay 3. Enter tip amount in tip section on screen 4. Finish paying	increase by tip amount and the server's account tip section will increase as well	displayed on bill and server's total tips is increased by the same amount.	
17	4.4	1. From a Table screen, finish paying for all orders 2. Play game that pops up 3. If customer chooses a winning number, a coupon code will be displayed 4. Coupon can be redeemed at bill pay to remove one dessert from the bill	If won, a free coupon code will be given for a free dessert.	Game gives a coupon when customer wins. When applied, one dessert is taken off bill.	Pass
18	4.5	1. From a Table screen, have orders previously placed 2. Select Pay 3. Select Split Bill 4. Enter desired split 5. Process payments for each split	The orders will be paid for from multiple sources	Bill can be split	Pass
19	4.6	1. From a Table screen, navigate to the bill 2. Enter payment information 3. Select either email or print receipt 4. If email, enter email address 5. Select option to pay order 6. Email receipt will be sent to given email address	The customer will receive an email and physical receipt	Customer can choose email or printed receipt and skip game by pressing back button	Pass
20	5	1. From a manager screen, select table statuses	A list of accurate table statuses are shown	Tables that the server is assigned to show up under the respective status headers	Pass
21	6	1. On a logged out Server screen 2. Enter a valid server id 3. Enter the corresponding password 4. Select login	The server was logged in and can now perform server tasks	The server is capable of logging in as long as the ID and PIN is valid	Pass
22	6.1	1. On a Server screen, select log out	The server is logged out, and the elapsed time since login is added to hours worked on server account	On the server options screen, the server can click "Clock Out" to log out of the	Pass

				system.	
23	7	1. On a Server screen, select order 2. Input number of table server is ordering for 3. Order as if customer	An order is placed for the desired table	The server may select a table and choose to view and modify their current order.	Pass
24	8	1. On a Server screen, select Process Payment 2. Input number of table server is paying for 3. Process payment as if customer	The payment is processed and the table total decreases by amount paid	The server may select a table and choose to process their payment.	Pass
25	9	1. On a Server screen, select Comp Item 2. Select table who has ordered placed 3. Select item in table's orders to be comped 4. Confirm	The orders total is decreased by comped item amount and the comp is recorded on server account	When processing payment for a table, the server can choose to comp the bill by a certain amount.	Pass
26	10	1. On a Server screen, select the Help button	The manager will receive a notification that the server needs help	Server is notified that a message is sent. Manager receives message on page reload.	Pass
27	11	1. Have orders submitted from various tables	The kitchen screen displays the list of orders ordered by time sent.	The kitchen displays the orders in chronological order	Pass
28	11.1	1. Have order submitted and shown on kitchen order screen 2. Select complete next to menu item in order	The meal is marked as completed on order screen	The order is turned to the color green to let a cook know they completed it.	Pass
29	12	1. Have order marked as complete by kitchen 2. Server refreshes page and is shown notification that kitchen has finished order for table	The server for the table who placed the order will receive a order complete notification	Works as expected. After kitchen marks order as complete and server refreshes page, server gets a notification for the order complete with table number	Pass

30	13	1. From the kitchen screen, select the “Help” button to call a manager 2. From the kitchen screen, select the “Notify Server” button on the relevant order to notify the server who is assigned to the table that sent the order	The chosen employee will receive a help notification from kitchen	The Kitchen can only send a notification to the manager with the help button, but there is a notify server button which also sends a help request.	Pass
31	14.1	1. From manager’s main menu, select “Menu” 2. Navigate to a menu item 3. Select “Remove” next to the menu item 4. Select “Yes” to confirm decision	The menu will no longer show the deleted one	You have to click on the “Menu” option then select a food item and then click the remove button	Pass
32	14.1	1. From manager’s main menu, select “ Menu” 2. Select “Add ” 3. Input new item’s information 4. Click the “Upload Image” button and select an image on the local disc 5. Click “Submit” button	The menu will show the new menu item	You won’t see that the new menu Item has been created until you recheck the other tabs	Pass
33	14.2	1. From manager’s main menu, select “Menu” 2. Select menu item to be edited 3. Click “Edit” button next to data item to be changed 4. Edit data item 5. Click “Save” button	The menu items details will show change when viewed now	You have to click on a menu item then press “Load Edit Form” and a form will open at the bottom of the page	Pass
34	15.1	1. From manager’s main menu, select “Inventory” 2. Select “Add Item” 3. Input new item’s name and quantity 4. Click “Submit” button	The new ingredient will be shown in inventory with inputted quantity	Works as expected	Pass
35	15.1	1. From manager’s main menu, select “Inventory” 2. Click “Remove Items” button 3. Click “Remove” button next to an ingredient	Removed item will no longer be shown in inventory	Works as expected.	Pass

		4. Select “Yes” to confirm			
36	15.2	1. From manager’s main menu, select “Inventory” 2. Click “Edit” button next to an item 3. Type in a new quantity 4. Click “Save” button	The shown count for the changed ingredient will be changed	Works as expected.	Pass
37	16	1. From manager’s main menu, select “Staff” 2. Click “Add Employee” button 3. Enter a name for new employee in name field 4. Enter an ID for new employee in ID field 5. Click “Submit” button	The new employee account will be visible on “Staff” screen	Works as expected although. Id’s are randomly generated.	Pass
38	16	1. From manager’s main menu, select “Staff” 2. Select “Remove” next to an employee account 3. Select “Yes” to confirm	The selected item will no longer be visible on “Staff” screen	Works as expected.	Pass
39	17	1. From manager’s main menu, select Staff	A table showing hours worked, total tips, and number of comps for each employee will be shown	Works as expected.	Pass
40	18	1. From manager’s main menu, select Stats	A table showing a list of all menu items ordered by number of times purchased and the times purchased amount will be shown	Works as expected.	Pass
41	19	1. From manager’s main menu, select Create Coupon 2. Choose coupon effect 3. Select Expiration Date 4. Confirm	A useable coupon code will be created	Works as expected. Expiration Date Format is unspecific.	Pass
42	20.1	1. Customer reaches the login menu screen. 2. Customer selects “create new account.” 3. Customer enters personal information	Customer’s loyalty reward account is created.	Works as expected	Pass

		4. Customer selects “create account.”			
43	20.2	1. Customer reaches the rewards screen. 2. Customer enters loyalty ID. 3. Customer selects “Login.”	Customer’s loyalty reward account is successfully accessed.	Works as expected	Pass
44	20.3	1. Customer has entered loyalty ID on rewards screen 2. Customer is prompted to re-order item from last order 3. If customer clicks image, item is added to order 4. Customer is taken to order page	Customer places a previously ordered item in their cart, ready to be ordered.	Works as expected. Previous item is now in current order.	Pass
45	21	1. Customer selects Place Order. 2. Customer selects drinks 3. Customer selects desired drink from list 4. Customer selects add item 5. Drink is added to cart	The menu item in the order will show the reduced happy hour price based on local time.	Works perfectly.	Pass
46	22	1. Customer reaches media selection screen. 2. Customer searches for desired media. 3. Customer selects desired media. 4. Customer hits play	The requested media will start to play.	Works as expected.	Pass
47	23	1. Customer reaches payment screen 2. Customer enters desired tip amount in tip box. 3. Customer selects ’leave note’ field and types feedback to be given to the server. 4. Customer finishes the payment process.	Server receives notification with typed feedback.	Works well. Tip amount as well as note are visible.	Pass
48	24	1. From manager’s main menu, select Schedule 2. Manager enters ID of employee being scheduled 3. Manager inputs all shifts the employee will work into schedule	A schedule for selected employee with imputed shifts is generated.	Employee’s can be scheduled for any date. Works using military time. Schedule for the current week is all that's	Pass

		4. Manager selects generate schedule.		displayed however.	
49	25	1. Customer opens menu 2. Customer selects category 3. Most popular item in that category is promoted	The customer sees the most popular item in a chosen category	When selecting any category, the most ordered option is given a text box and placed at the top of the list to show popularity.	Pass

3 Unit Tests

The unit tests were performed using the JavaScript testing framework JEST. This was done through `npm` using some non-default packages including `frisby`. These can all be tested repeatedly to ensure functionality. There may be some difficulties with creating and editing menu items as these require file uploads. The test scripts themselves have a reference to one of the images included in our source files. So, if the hierarchy is maintained, all 80 unit tests will likely pass barring any unforeseen local issues. All of the scripts used for testing are located in the `src/UnitTests/` directory from [2].

The results of the tests are included with this document in the file 'results.txt'. This file contains a prettified JSON-formatted result directly from JEST. The attributes The first 10 attributes represents the result of all test cases and test suites (each test suite is a file, and there are test cases within each file). The results of this show that we performed 80 tests in 11 test suites. All of these passed.

The test results for each unit test begins in the array on line 31 with the attribute "testResults". Each JSON object in this array is a single test suite. To explain the structure, we will consider the first object from line 32 to line 88. The "assertionResults" object is an array of JSON objects where each object represents a single test. Each of these objects has a "fullName" attribute that is a description of what the test is performing. For example, the first object in the first test suite, beginning on line 34 and going through line 41, is has the description "Get the list of orders from the order database." This test queries the server's order database and tests for a proper response. Some tests may have similar names due to their actions being similar. For clarification, please view the relevant source file in the Unit Tests directroy `src/UnitTests` from [2].

Each object also has a "status" attribute whose value representes the result of the test. Since all of our tests passed, this attribute has the value "passed" for each of the 80 test cases. At the end of each test suite JSON block (e.g., lines 83-89 for the first suite), there is another "status" attribute. Since all of our test suites passed, this will have the value "passed" for each of the 11 test suites. This pattern continues for the rest of the results document, showing the passing status of each of the test cases and test suites.

References

[1] [Online].Available: http://64.225.29.130

[2] [Online].Available: https://github.com/NickChiapputo/TheLeftovers/tree/master/src