**MentoringPOT** (Node Version)

**Functional Specification**

Jaycen Milling

Last Updated: March 18, 2019

**Overview**

**MentoringPOT** is a service that allows users to select items from the menu (hypothetical Restaurant) and then review their order, change their order, and then submit their order.

It also allows “admins” to update the menu, adding items, removing items, adding quantities to an item, changing the item name, etc.

**These specs are not all complete or functional and will need to be updated and completed.** The screenshots shown are merely to illustrate the underlying functionality. The actual functionality, look, and feel will be developed over time.

**Scenarios**

**Scenario 1**: Customer

Customer navigates to the home page and the menu is populated. The customer looks at all the items and realizes they want to order some “Cheese Fries” as an Appetizer, “Grilled Chicken Alfredo” as the main entree, and “Tea” to drink. They click the “Review Order” button and are directed to the Review order page where they realize they accidently added two “Cheese Fries” and changed to amount from two to one. They then click the “Submit Order” button and are directed to the Submit Order page and it shows them the quantity of each item they ordered, the subtotal (the overall price of all he items added together), and the total (subtotal + 7% taxes).

**Scenario 2**: Administrator

Administrator gets a shipment in to restock all the items on the menu and gets a Chicken Quesadilla (new Appetizer), Fish and Chips (new Dinner entrée), BLT Sandwich (Lunch entrée), Green Beans (a new side), and Coke (a new drink). He navigates to the Admin console where he changes the quantity of the current items to the correct amount. He then clicks the “+” button next to the Appetizers section and adds the “Chicken Quesadilla” under the item, “8.99” as the price of the item, and “6” as the quantity. Next he clicks the “+” button next to the Lunch section, he adds the “BLT Sandwich” under the item, “7.99” as the price of the item, and “10” as the quantity. The admin then clicks the “+” button next to the Dinner section and enters “Fish and Chips” under the item section, “10.99” as the price of the item, and “10” as the quantity. When the admin is finished, he clicks the “Save Menu” button and it updates the menu for when a Customer views the menu again.

**Screen by Screen Specification**

**Home Page**

The home page is the Producer and Consumer side of the Kafka Menu topic.

It Consumes the Menu topic to create the menu in the UI for the User to select certain items.

When the user has selected the items it wishes to “order” they can press the “Review Order” button that will take them to the Review Page.



**Review**

The user is taken to this screen to review their order, they can add, remove, and change the amount of the items they want. Then the hit the “Submit Order” button that will direct the user to the “Order Submitted” page (essentially a receipt), but also push the order to the Order topic in kafka.



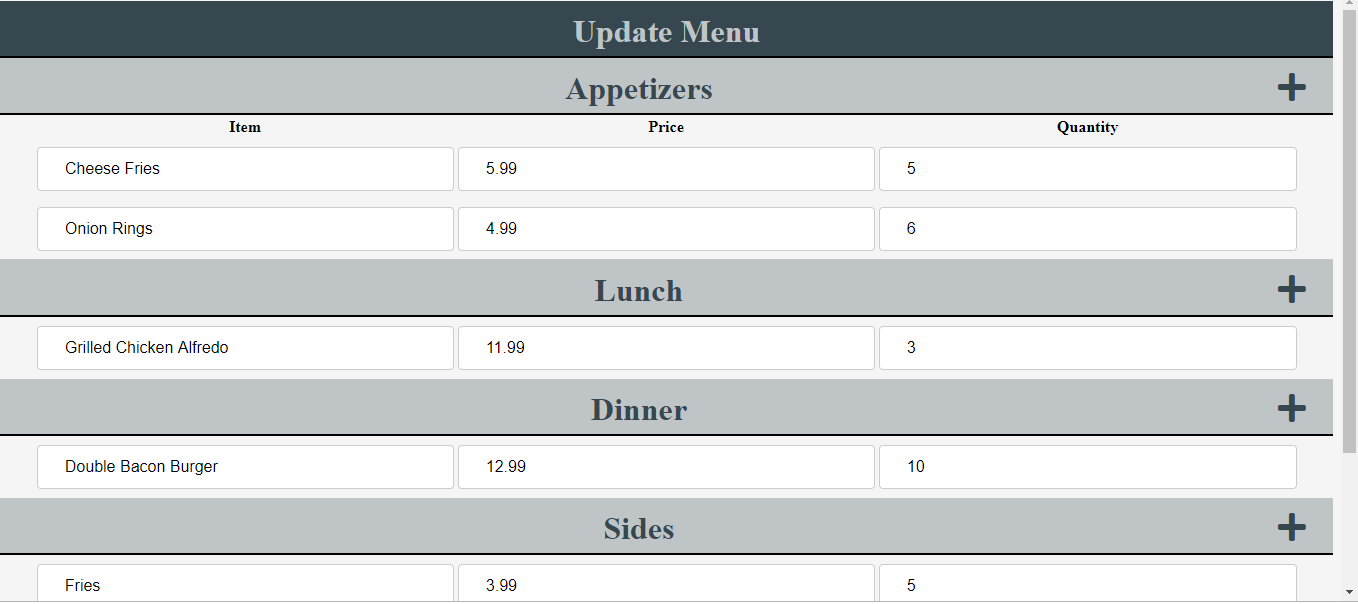
**Order (Submitted)**

This screen is more like a receipt for the Customer, it tells the user how many of each of the items they ordered, subtotal, and Overall total of their order.



**Admin (Update Menu)**

Allows the admin to update the “Item” (name), “Price”, and “Quantity” as well as adding items under each category.



**Technical Specs**

**Technologies Utilized**

1. **NodeJS**
   1. **Dependencies Utilized**
      1. **Express**
      2. **MongoClient**
      3. **Body-Parser**
      4. **EJS**
      5. **Kafka-Node**
2. **Kafka**
3. **MongoDB**

**External Interface Requirements**

1. **User Interfaces**
2. Front-end software: NodeJS version
3. Back-end software: ExpressJS, Kafka, and MongoDB
4. **Hardware Interfaces**
5. A browser which supports HTML and Javascript
6. **Software Interfaces**
7. Developed in Windows utilizing Visual Studio Code.
8. NodeJS for the Javascript runtime engine.
9. MongoDB as a backup of Kafka.
10. Kafka real-time topic monitoring.
11. **Communications Interfaces**
12. This project supports all types of web browsers.