Deliverable 2

Angelica Longo

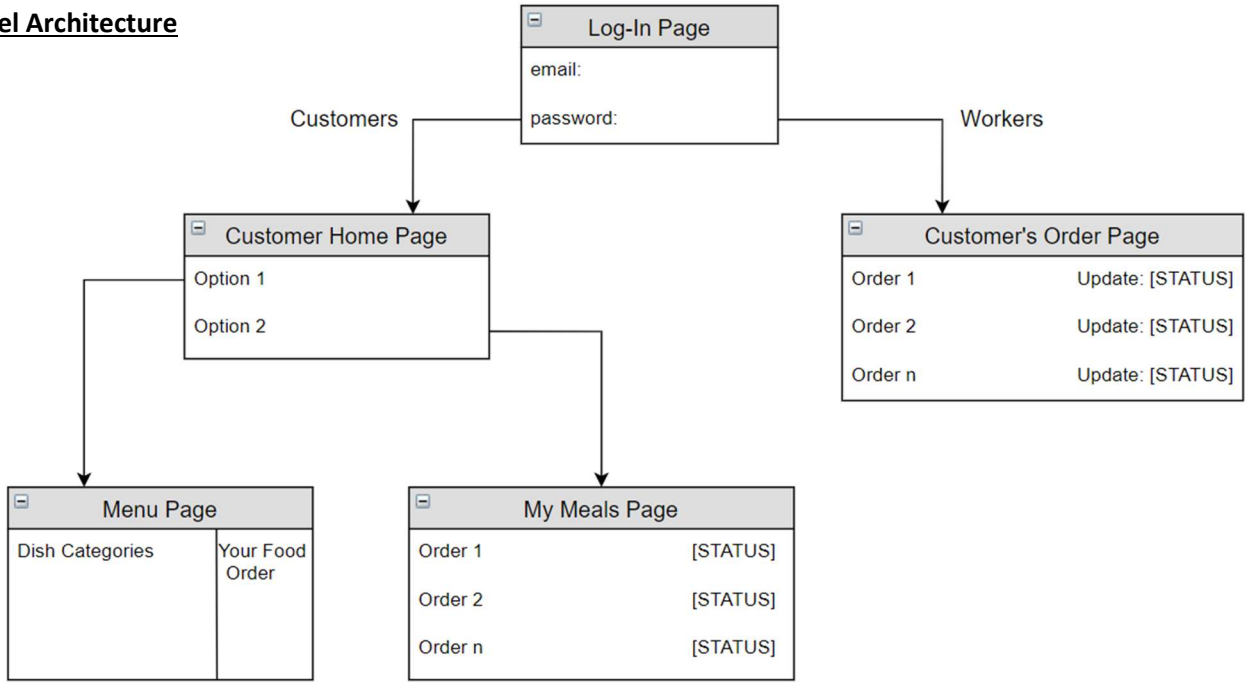
Project #1: MealStats

Fall 2020 – COP 4331C

Contents of this Document

High Level Design

High-level Architecture

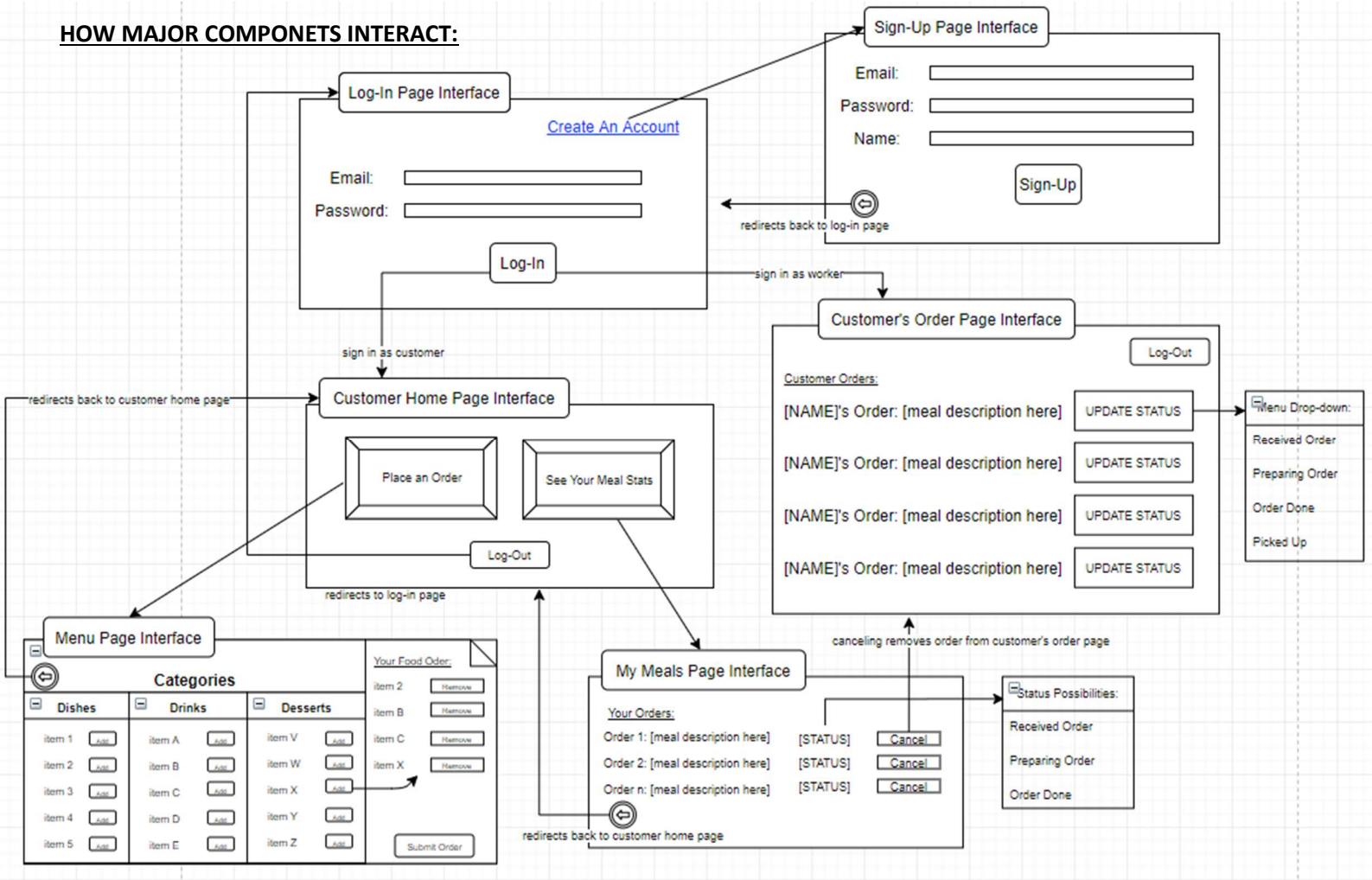


Major Components

Log-In Page	Will redirect to different page depending if the user is a worker or customer (verified by user's credentials on database)
Customer Home Page	The redirected page (from Log-In) if user signs-in as a *customer*. This page provides 2 options. Will redirect customer depending if they want to choose to place an order, or view status of previous order(s).
My Meals Page	The redirected page (from Customer Home Page) if user wants to view their status of an order(s). Any previously placed orders will be listed along with meal's status from restaurant.
Menu Page	The redirected page (from Customer Home Page) if user wants to place an order. Browse food categories/options. Vertical sidebar on the righthand side of the page for selected meals. Payment/submission button at bottom of sidebar.
Customers Order's Page	The redirected page (from Log-In) if user signs-in as a *worker*. This page provides a list of all the orders submitted from customers. Status changing options for each order.

Architectural Styles: The Publish-Subscribe architectural style applies to the MealStats system since restaurant workers using the system can change the status of a customer's submitted order. The customers are notified and can then view the updated status every time it has been changed when they log-in. The Server-Client architectural style also applies to the MealStats system. The server component would refer to the workers who update, and process orders submitted by the customers. The client component would refer to the customers, who can submit meal orders and check their meals status by signing into their MealStats account at any time.

HOW MAJOR COMPONETS INTERACT:



System Interfaces	Description/Inputs	Outputs
Log-In Page	• Text fields for user to enter their email and password • "Log-In" button (this will redirect to *Customer Home Page* if the user is a customer, or *Customers Order's Page* if the user is a worker) • "Create Account" link for new users to create an account (this will redirect to the *Sign-Up Page*)	• Error pop-up for entering incorrect credentials
Sign-Up Page	• Text fields for user to enter their email, password, and name for account • "Sign-Up" button (this will flag a pop-up) • "Back" button (this will redirect to the *Log-In Page*)	• Error pop-up for not entering credentials and pressing the "Sign-Up" button • Confirmation pop-up for entering credentials (pressing "OK" will redirect to *Log-In Page*)
Customer Home Page	• "Place an Order" button to be redirected to the *Menu Page* • "See your Meal Stats" button to be redirected to the *My Meals Page* • "Log-Out" button (this will redirect to the *Log-In Page*)	• Confirmation pop-up before logging out
Menu Page	• Banner labels for "Dishes", "Drinks", and "Desserts" • "Add" button next to each food option displayed (this is will add the selected food item to the "Your Food Order" sidebar) • "Submit Order" button at the bottom of the sidebar (this will flag a pop-up) • "Back" button (this will redirect to the *Customer Home Page*)	• Confirmation pop-up before submitting order (notify about tax) (pressing "OK" will redirect to *Customer Home Page*)
My Meals Page	• "Cancel" button next to each order (this will cancel the order submission and flag a pop-up) • "Back" button (this will redirect to the *Customer Home Page*)	• Confirmation pop-up before canceling an order (the order will also be removed from the *Customers Order's Page*)
Customers Order's Page	• "Update Status" drop-down menus next to each order • "Log-Out" button (this will redirect to the *Log-In Page*)	• Confirmation pop-up when changing an order status to "Picked Up" • Confirmation pop-up before logging out

---

Design Issues

The ***reliability*** of the MealStats system ensures that its databased is backed-up so that if an error occurs, or if the system crashes for any reason, there will be no loss of data. The design of the system is purposefully made to be simplistic and minimal so that the software may be ***reused*** among any restaurant that wishes to use this online ordering system. While the software design is made simple, its system is a basic template so that it may be improved to adapt to other attributes and be ***maintained*** easily. MealStats has a system in which the extent of ***testing*** is made with ease. The design avoids large classes and includes consistent small methods (short units of code) that make assessments straightforward. Since code complexity is minor, the processing speed, data transfer rate, network bandwidth and throughput, are all expected to ***perform*** properly under an unexpected workload. The online ordering system is designed for ***portability*** and made for any device that supports a web browser. The security aspects of the system include email and password verification, along with confirmation pop-ups before any payments are made.

What technical difficulties do you expect to encounter?

Regarding technical difficulties, I expect to have the most trouble on setting up an encrypted database. This database will have to be backed-up frequently and confirm whether a user already has an existing account, is a customer, or is a worker. Another technical issue includes how to make the web application public on any browser, and not just on the developed local machine. However, the use of tutorial videos will likely be of most help in figuring out how to accomplish issues regarding database set-up.

What design trade-offs did you make in your selection of the architecture?

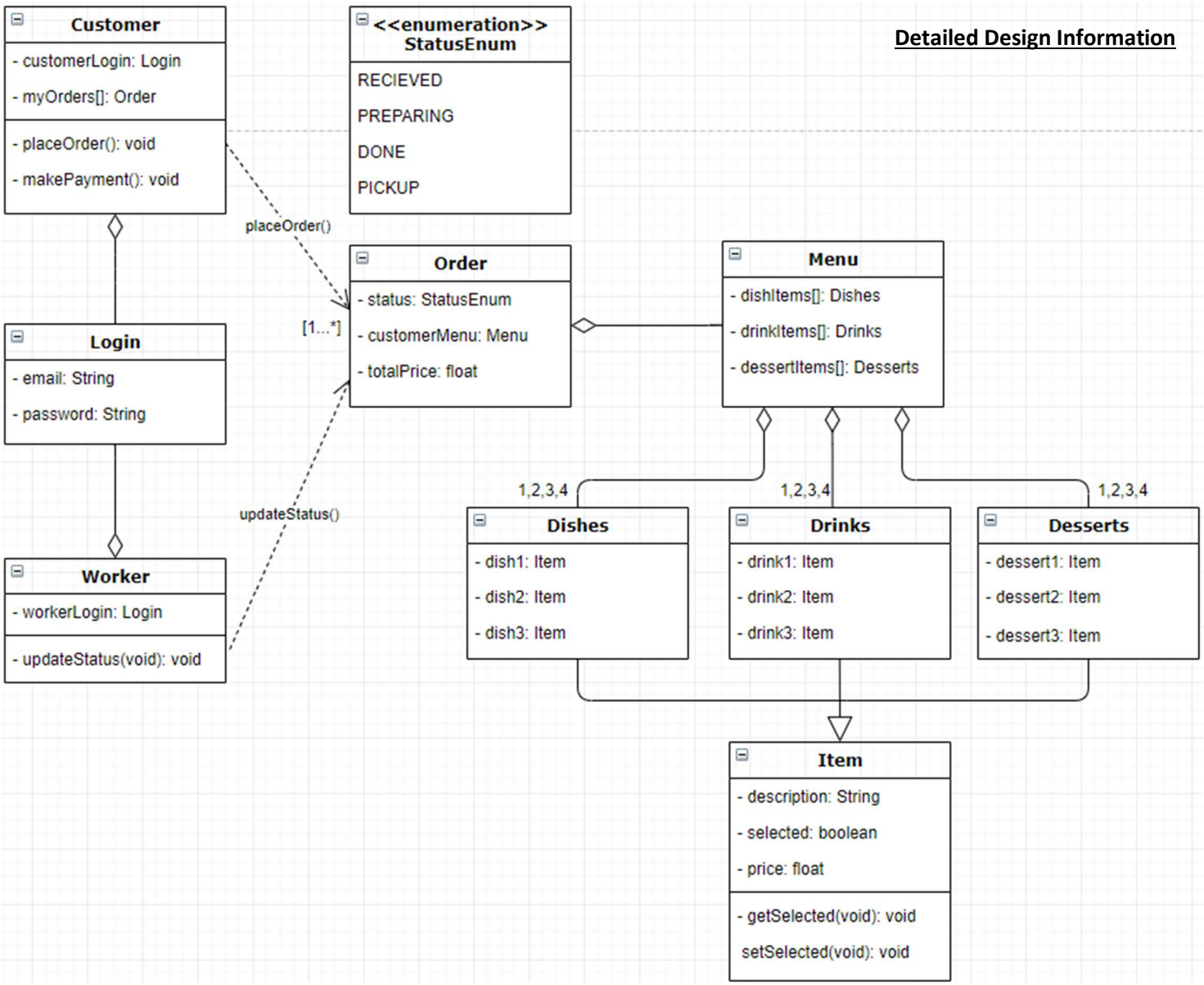
A design trade-off that I decided to implement, was being redirected to two different webpages depending on if you signed in as a customer or a worker. The rationale for this being the different uses of the same software. Since workers will only need to update the status of customer's orders, there is no need for them to view the entire website's menu options and will thus not be available to view them. A customer will have to make an account to log-in, whereas workers will all share the same log-in information that is already built-in to the system. The idea behind this is to increase productivity, so that workers will only need to log-in, update any orders as needed, and log-out when done. A possible technical risk for having all the restaurant workers sharing the same built-in log-in ID and password, may be conflict when trying to update customer status all at the same time.

Detailed Design

Detailed Design Issues

Design Issue	Result/Reasoning
reliability	Ensuring that the MealStats databased is backed-up so that if an error occurs, or if the system crashes for any reason, there will be no loss of data.
reusability	Designing of the system is purposefully made to be simplistic and minimal so that the software may be reused among any restaurant that wishes to use the MealStats online ordering system.
maintainability	The simplistic design of the software is a basic prototype so that it may be improved to adapt to other attributes and be maintained easily.
testability	Avoiding large classes and includes consistent small methods (short units of code) that make assessments straightforward.
performance	Under an unexpected workload, having minor code complexity will allows the processing speed, data transfer rate, network bandwidth and throughput to perform more appropriately.
portability	The online ordering system is web application designed for convenience and made for any device that supports a web browser.
security	The safety measures of the system include email and password verification, along with confirmation pop-ups before any payments are made. This is will allow the user to be aware of when their personal data is being accessed.

Detailed Design Information



Trace of Requirements to Design

Requirement ID	Requirement Description	Architecture Reference	Design Reference
FUNCT-1	Shall prompt user to login or create an account	Log-In Page, Sign-up Page	Login
FUNCT-2	Shall provide customers with a webpage of dish categories	Menu Page	Menu
FUNCT-3	Shall provide workers with a list of submitted orders from customers	Customer's Order Page	Worker
UHF-1	Shall support users who are restaurant workers	Customer's Order Page	Worker
UHF-2	Shall support users who are restaurant customers	Customer Home Page, Menu Page, My Meals Page	Customer
DATA-1	Shall automatically add a 6.5% tax fee to every order submitted	Menu Page	Order

Test Plan

Overall Objective for Software Test Activity

I plan to test the following:

- Making an account on MealStats
    - When a new customer wants to make an account, they need to enter an email, password, and name. This action should add the user's credentials to the MealStats database and allows them to log-in.
- Correct redirecting from log-in page
    - From the log-in page, if the user enters a specific ID for the email and password that is already built into the system, then they should be redirected as a **worker** to the *Customer's Order Page*. If the user enters the email and password that they created into the log-in page, then they should be redirected as a **customer** to the *Customer Home Page*.
- Adding items to order (and total)
    - When a customer selects an item from the Menu Page, it should be added to the sidebar titled "Your Food Order", and each item selected will be added to the list along with the total price added up from each item.
- Submitting orders
    - Once a customer has finished selecting items from the menu and hits the "Submit Order" button, this should add their order to the *Customer's Order Page* and then redirect the customer to the *Customer Home Page.*
- Status updates
    - On tion to mark a status for each order. The action of changing the status should also be visible to the customer on their *My Meals Page* each time their order status is updated by a worker.
- Removing an order/Pick up
    - From the *My Meals Page*, if the customer selects the "Cancel" button next to an order, the order be will removed from both the *My Meals Page* and the *Customer's Home Page* visible to the workers. In addition, once a worker has updated an order status to "Picked Up", this will also remove the order from both the *My Meals Page* and the *Customer's Home Page.*

Description of Test Environment

MealStats will be personally developed and manually tested in the Eclipse IDE for web developers within a Windows 64-bit operating system using the Java programming language and the The Apache Tomcat Java servlet container tool.

Rather than testing the web application system on a specific browser such as Google Chrome, Firefox, Safari, or Internet Explorer, a manual testing environment will be used. This testing environment differs from the environment in which the software will operate, such that the Apache Tomcat Java servlet container tool will be utilized inside Eclipse for its ability to render web pages. These web pages use Java Server page coding, and the developing software is intended to be made available on any web browser across all platforms.

Stopping Criteria

If errors are found during testing, and the error is a minor mistake, I will simply record it for later correcting. If a larger error is found during testing (one that doesn't correctly execute the objectives mentioned above), then I will commit to stop testing and fix that problem.

For each software test activity mentioned above, test cases will be applied such that all possible outcomes are accounted for. For example, when testing the creation of a new account, I will test the case in which an account is made successfully, incorrectly (this should cause a pop-up error), and the creation of an account that already exists (this should also cause pop-up error to the user).

The software will be "good enough to deliver" once all known errors for each objective are found and are no longer errors. Instead, when something happens incorrectly or when the user does an unregistered interaction with the system, error message pop-ups will notify the user.

Description of Individual Test Cases

- Test Conditions: See Test Environment above

Test Objective	Test Description	Expected Results
Making an account on MealStats	Will test the acceptance of an email, password, and name as input from the *Sign-Up Page*. There will be three text fields for the inputs to be taken in as strings.	Creating a MealStats account successfully should result in the email, password and name being saved onto the database as string values. To check if the execution was done correctly, the same credentials entered in the *Log-In Page* should redirect to the *Customer Home Page*.
Correct redirecting from log-in page	Will test the correct redirection pages for both customers and workers from the *Log-In Page*. There will be two text fields for the email and password to be taken in as strings.	When a customer enters their email and password created from the *Sign-Up Page*, then the expected result should be the same one mentioned above. When entering the built-in email and password into the Log-In Page (for example: worker@mealstats.com and worker123) this will redirect all workers to the *Customer's Order Page*. (Note: workers will know this since they do not create accounts).
Adding items to order (and total)	Will test that when an item is selected from the menu, it appears as a list on the vertical sidebar titled "Your Food Order". Each item includes a floating-point value that represents its cost.	This feature works correctly if each item selected from the menu gets added to a list along with the total price added up from each item (and another total showing the additional 6.5% tax).

Submitting orders	Will test that the "Submit Order" button performs as expected and redirects to anticipated page afterwards.	If the submit button executes correctly, then a confirmation pop-up will notify the user of the additional tax being added to the order. Pressing "OK" on the pop-up will then add the order to the *Customer's Order Page* and redirect the customer to the *Customer Home Page*.
Status updates	Will test the feature of updating the status of an order using a drop-down menu on the *Customer's Order Page*.	A worker updating an order's status correctly from the *Customer's Order Page* should result in the ability for the customer to view the same status from their *My Meals Page* each time it is changed.
Removing an order/Pick up	Will test the removal of a listed order when selecting the "Cancel" button, as well as selecting "Picked Up" from the status update drop-down menu.	When a customer selects the "Cancel" button next to an order from their *My Meals Page*, it is expected that the order will be removed from both the *My Meals Page* and the *Customer's Home Page* visible to the worker. The same result is also expected to happen when a worker updates an order status to "Picked Up".

Trace of Individual Test Cases to Requirements

Requirement ID	Requirement Description	Test Case Reference	Status
FUNCT-1	Shall prompt user to login or create an account	N/A	Succeeded
FUNCT-2	Shall provide customers with a webpage of dish categories	N/A	In progress
FUNCT-3	Shall provide workers with a list of submitted orders from customers	N/A	In progress
PE-1	Shall run on all modern/smart devices which support internet access and contain a built-in web browser	RES-1	In progress
UHF-1	Shall support users who are restaurant workers	FUNCT-1, FUNCT-3	In progress
UHF-2	Shall support users who are restaurant customers	FUNCT-1, FUNCT-2	In progress
DOC-1	Shall make use the of GitHub for visual control and configuration management	N/A	Succeeded
DATA-1	Shall automatically add a 6.5% tax fee to every order submitted	N/A	In progress
RES-1	Shall operate on any web browser	PE-1	In progress
SECRTY-1	Shall maintain a backup of all the customer's and worker's credentials	FUNCT-1	In progress
QA-1	Shall be personally documented which includes citations on reports, documentation in the code itself, testing, and applying improvements to finished features	N/A	In progress

Appendices: N/A