Software Engineering (ECE 452) - Spring 2019
Group #3

Restaurant Automation Codename Adam

URL: https://github.com/sa2423/SE-2019

Date Submitted: February 10, 2019

Team Members:
Seerat Aziz
Lieyang Chen
Christopher Gordon
Alex Gu
Yuwei Jin
Christopher Lombardi
Arushi Tandon
Taras Tysovskyi
Hongpeng Zhang

Table of Contents

Contributions Breakdown

Everyone contributed equally in this part of the report #1.

Section 1: Customer Problem Statement

1.1: Problem Statement

Restaurant automation is a daunting task, as there are many variables to consider in the automation process. Moreover, it is important to maintain a balance between efficiency and hospitality as most customers associate restaurants with their ambience as opposed to mechanized efficiency. Many restaurants nowadays have started to automate their tasks, such as Panera Bread and TGIF, by streamlining the ordering and payment processes by employing tablets with specialized apps. Spyce, a restaurant in Boston, has automated its cooking process by replacing human chefs with a robot [4]. However, even the owners of Spyce seek to maintain the human touch by attempting to tailor the experience for each customer.

For this reason our project aims to automate basic tasks that customers, waiting staff and chefs have to perform in restaurants, without outright replacing them, in order to maintain the human touch that the restaurant experience depends on. This, in turn, will decrease the time it takes from customers coming in and placing their order to actually being served their food, resulting into better dining experience. Additionally, it will help the restaurant owners manage their business better through detailed statistics and analysis of trends. It will provide a way to easily see all the placed orders as well as a way to predict the amount of ingredients that will be needed for any given day, (inferred from historical data). This allows the restaurant owners to dramatically cut down the leftovers and shape their menu accordingly to what sells best. Moreover, our project will improve customer experience by recommending meals.

While the proposed solution requires additional investment in the form of electronic devices for customers and servers, as of 2018 lower-end tablets running Android operating system can be purchased for under $50 dollars, especially in larger quantities. As such, this is well within the realm of financial feasibility for medium and even small, family owned restaurants. Although our prototype is tailored towards Android operating systems, it can be implemented on the iOS operating system if restaurants do not mind paying more.

Problem: Helping customers decide on a meal to eat

A common problem that customers face when going to eat a restaurants, whether they have visited the restaurant before or not, is deciding on what food to eat. This problem is often worsened by menus with food items with very “foreign” sounding names or unfamiliar menus in general. Such problems can discourage customers from visiting the same restaurant, so we find that assisting customers with choosing meals will cause them to maintain their relationship with our establishment. As a result, this will cause our customer base to increase over time.

The Codename Adam Solution:

We seek to solve this problem by implementing a meal recommendation system, which will be enhanced by using responses from the review system. In order to implement the recommendation system, the user will be given the choice to take a short quiz before ordering their meal. This quiz will ask questions about their meal preferences (if they want a meal with cheese, rice, meat, etc.) to recommend a list of appropriate meals. This quiz component will incorporate active learning by taking the user's choices (what meal they select from the given list) into account for future recommendations. Likewise, the user will be allowed to "favorite" meals on the application, which will also be taken into account for future meal recommendations. After a user finishes their meal, they will be given a short questionnaire to rate the meal and restaurant service. This responses from this short questionnaire (the review system) will also be taken into account when recommending meals to future customers. If the customer returns, we can further streamline the ordering process, using an account system with facial recognition. During this procedure, all the customer needs to do is take a seat and look at the camera, avoiding any inconveniences such as user login. Of course, to avoid any concerns about data collection and invasion of privacy users are informed of their information being stored and can opt out of this feature at any time.

Problem: Providing customers with efficient and careful service

With the advancement of internet technologies, customers want information delivered to their screens instantly. For restaurants in particular, they would like to see today's menu, reserve a table or even order food in a matter of seconds, without ever leaving their homes or talking to people. Along with customer service, restaurants have consider their customers' dietary concerns. This includes taking religious dietary laws (such as Halal and Kosher) and fad diets (Keto and Vegan) into account. Moreover, as food allergies have become more prevalent in today's era, it has become more important for restaurants to be vigilant with their food preparation efforts.

The Codename Adam Solution:

Codename Adam provides an ecosystem of websites and mobile apps that allow customers to interact with restaurants remotely and effortlessly. Checking today's specials, getting food recommendations, reserving a table and making an order can now be done in a matter of seconds. In order for the restaurant staff to be able to keep up with the features offered online, we will also offer apps for servers and chefs to increase their productivity by being able to receive orders instantly and be notified immediately when an order is done cooking and is ready to be served. Furthermore, the app specialized for the waiting staff will provide information about the meals, such as

ingredients used, calorie information, and any possible allergy warnings. Likewise, the restaurant website (with admin privileges and the ingredient prediction system) will help restaurant owners decrease operating costs and increase profits.

Problem: Make server spend their time more efficiently

A server has a lot of things to do when they are working. For example, they need to serve for new-coming customers and stand by them to listen to their decisions on meals, they need to bring a finished meal from chef to a customer who is waiting, they need to go helping those customers who is calling a server for some specific questions. They have a lot things to do in their work, so an efficient way to use their time is really desired so that all customers can enjoy their time by being fully served.

The Codename Adam Solution:

Actually, there are some unnecessary actions that can be avoid by making our application more advanced. For example, if we can allow the customers to order their food on the table app, the time server used to spend on record customers' decisions can be saved. If we can allow customers to order food online instead of making phone calls, then the time severs use to talk to customers on phone can be saved. If we can allow customers to use the application to write down their specifications on each meal, then the overhead severs spend in transmitting the information from customers to chefs can be saved. With time saved, servers can spend more time on those customers who requests a server calling so they can be helped at once.

Problem: Maximizing profits and streamlining process for the Chef

The kitchen is the engine of a restaurant. Every customer's order must pass through the kitchen, which means that the efficiency and quality at which the kitchen operates heavily impacts the performance of the restaurant itself. As a result, it is in the restaurant's best interest to promote the efficiency of its chefs and the quality of its ingredients and to reduce wastage in its kitchen. If this can be accomplished, chefs will have more time and fresher ingredients, allowing them to produce a larger quantity of meals of higher quality, not only improving the customer's dining experience, but also increasing the restaurant's profit potential.

The Codename Adam Solution:

Codename Adam's ingredient prediction system helps to solve many of these problems that restaurants face. Based off of records of which menu items were ordered on the same day of the week in previous weeks, the ingredient prediction system will calculate an estimated amount of each ingredient that will be required on that specific day. This will allow chefs to determine how much of each ingredient to prepare for the

the day's dishes, reducing food wastage, preventing chefs from wasting time preparing unnecessary ingredients, and ensuring that dishes use only the freshest of ingredients. In the beginning, the chef may want to prepare a little more than what is estimated, but over time, the estimate should become more refined and more accurate, as outliers are identified and pruned from the estimation. This will be accomplished by performing the RANSAC algorithm on the amount of ingredients needed for the k most recent weeks. The estimate may also be refined by trends in dish reviews and favorites and pending reservations. An extension of this feature may further divide the ingredient prediction by time of day, to maximize the freshness of ingredients and efficient use of the chefs' time.

What makes Codename Adam a better solution for restaurant automation?

Codename Adam is a better solution for restaurant automation because unlike applications created in previous years, it provides a way for restaurants to plan their future spending through the ingredient prediction system and enhances the customer experience through the meal recommendation system. Our project is an improvement over Why W8 [1], the Fall 2018 solution to restaurant automation, because it utilizes their innovations (rating and favorites system) to create a meal recommendation system. Along with providing a list of meals frequently eaten at a restaurant, registered customers will benefit from receiving recommendations based on their personal preferences even if the menu changes. Moreover, our project is an improvement over FoodEZ [2], the Spring 2015 solution to restaurant automation, because it streamlines the meal payment process even more by providing different payment options through the customer application and table application. This reduces the time waiting staff spends on accounting for meals paid using different methods.

1.2: Glossary of Terms

Customer app: a software that customers can download on their mobile devices and use to order food from the restaurant

Table app: a software that is built inside every table devices, so dine-in customers can use it to order food, request service and pay the bill

Server app: a software that every server needs to download on their mobile devices and it will help alert everything associated with the demand of their service

Chef app: a software that every chef needs to see the list of ordered meals they need to cook

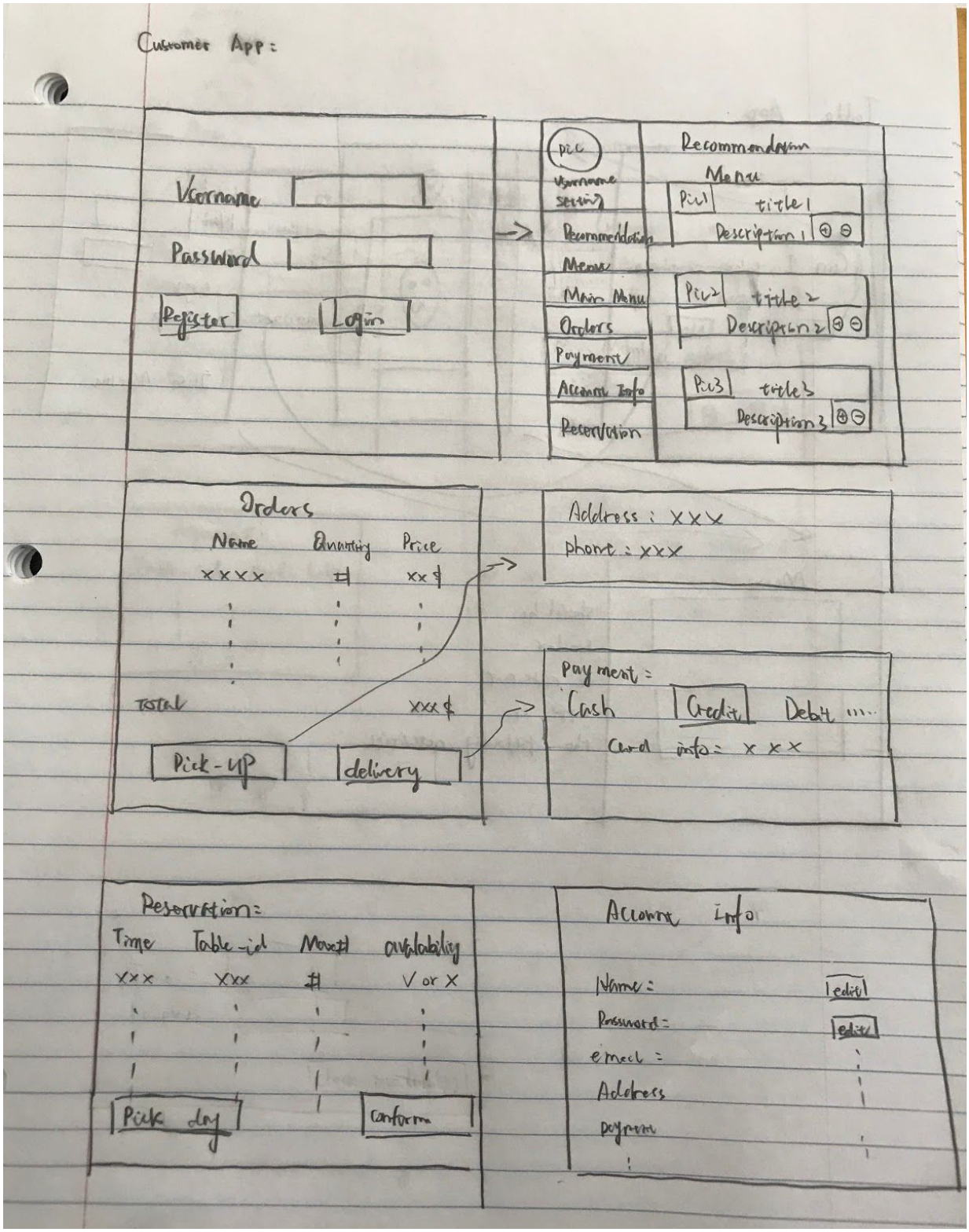
Section 2: System Requirements

2.1: Customer App

Functional Requirements:

Identifier	Priority (Higher number indicates higher priority)	Requirement
CA-01	5	As a user should be able to browse the most up to date menu
CA-02	1	All users should be able take a quiz to get meal recommendations
CA-03	5	All unregistered users should be able to register
CA-04	1	All registered users should be able to see meal suggestions based on their previous dining habits
CA-05	3	All registered users should be able to order food online, either to be delivered or to be picked up Possible
CA-06	3	All registered users should be able to reserve a table at the restaurant
CA-07	4	All registered users should be able to choose the payment methods: cash, debit or credit if they choose to order food online or to pick-up
CA-08	4	All registered users should be able to rate the all meals they ordered

On-Screen Appearance Requirements:



2.2: Table App

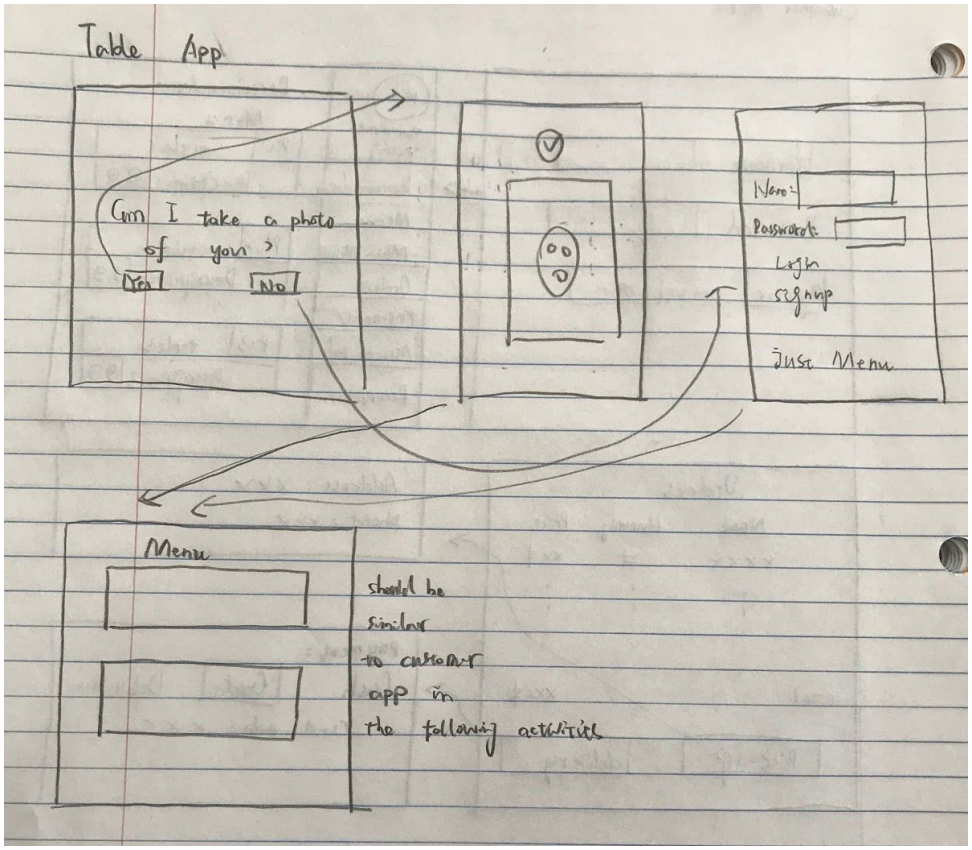
Functional Requirements:

Identifier	Priority (Higher number indicates higher priority)	Requirement
TA-01	1	The app should automatically detect when someone sits down at the table using face detection from the camera feed
TA-02	1	After detection, the app should attempt facial recognition to see if the user has authorized using facial detection before, has already dined in the restaurant before, and offer suggestions automatically
TA-03	3	If user's face is not recognized, there should be a way for users to login or register, or go on for the following steps as a non-registered guest.
TA-04	1	If the user chose to manually log in or register, the app should ask the user for the agreement of using facial recognition in the future (this option can be changed after log in, and the app will only save the data of user's face features after getting the agreement)
TA-05	5	All users should be able to view the menu
TA-06	5	All users should be able to make an order and pay for it in person
TA-07	4	All registered users should be able to make an order and have their card be charged automatically (if they have one attached to their account)
TA-08	5	All users should be able to request assistance by pressing a button

Nonfunctional Requirements:

Identifier	Priority (Higher number indicates higher priority	Requirement
TA-09	1	The user should not need to wake the table device for ordering, it should be done automatically when they sit down at the table.
TA-10	1	The user should not be waiting for the face detection and facial recognition process for too long (ideally up to 5 sec), during this process, the user should see a welcome page.
TA-11	1	All users in the restaurant should be able to use facial recognition simultaneously, without waiting for over 5 sec.
TA-12	3	All users should be able to manually log in when any part of the facial recognition system is not available or has failed
TA-13	5	The user should not encounter power shortage of the table device during the entire operating time
TA-14	5	The user should not encounter signal shortage and low transfer speed on the table device regardless of the table they choose to sit at.
TA-15	4	The user should not suffer noticeable delay when they're scrolling through the menu and click on items
TA-16	4	The user should not suffer screen tearing or low-resolution pictures when they viewing the menu.

On-Screen Appearance Requirements:



2.3: Server App

Functional Requirements:

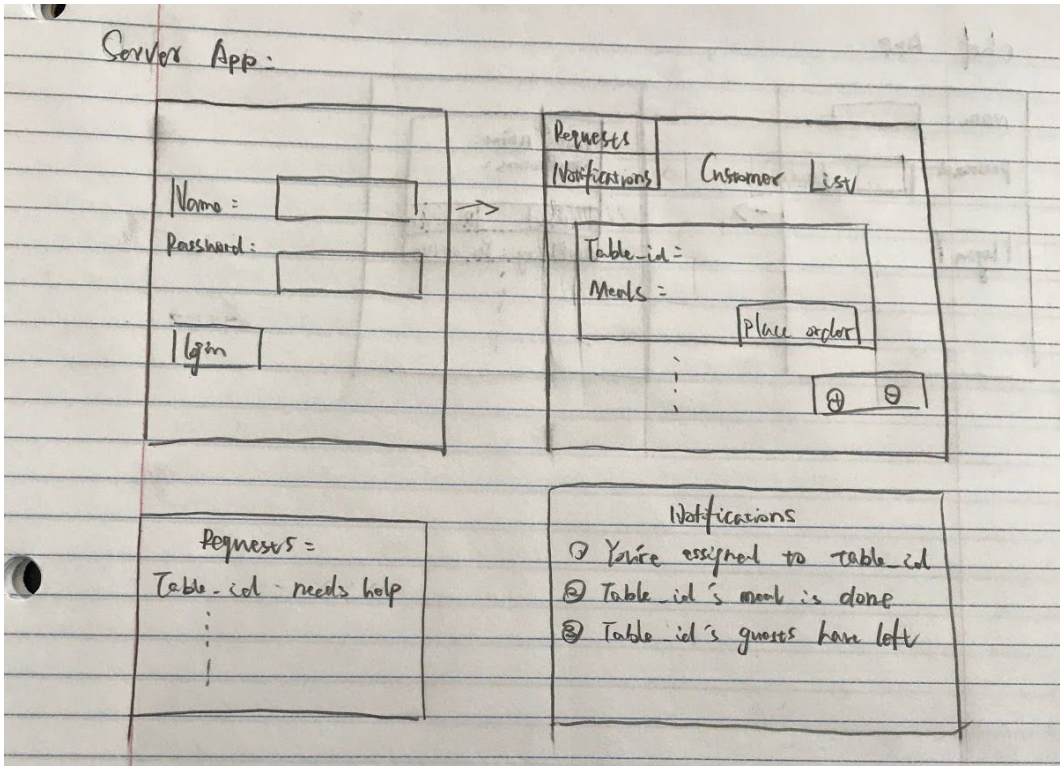
Identifier	Priority (Higher number indicates higher priority)	Requirement
SA-01	5	All users have to login before using the app
SA-02	5	All registered users should be notified when a customer requires assistance
SA-03	5	All registered users should be able to manually place an order for a customer

SA-04	5	All registered users should be notified when an order has been completed and should be served to the customer
SA-05	3	One of registered users should be assigned to serve a table when a table detects that someone has sit down.
SA-06	2	Users assigned to a table should be notified when a customer or customers have left

Nonfunctional Requirements:

Identifier	**Priority** **(Higher number indicates higher priority**	**Requirement**
SA-07	5	The app should be easy to navigate and use, since waiters have to serve multiple customers as quickly as possible.
SA-08	5	Any notifications sent by customers or chefs or the tables' facial recognition system should be shown on user's screen as fast as possible
SA-09	3	This application should be modular so that any new updates can be made easily in case new functionality is added
SA-10	3	The application should be designed to be as clean as possible so that some non-necessary or repeated information or activity can be avoided.

On-Screen Appearance Requirements:



2.4: Chef App

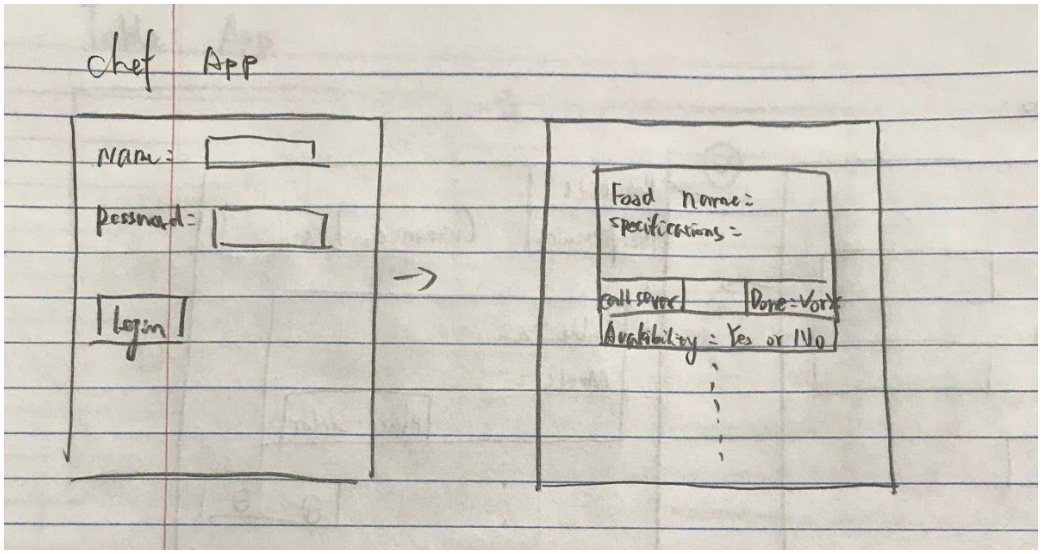
Functional Requirements:

Identifier	Priority (Higher number indicates higher priority	Requirement
CF-01	1	All users should be able to view a dynamically updated queue of orders.
CF-02	1	All users should be able to mark an order as fulfilled after it has been prepared, which would send a notification to the server app.
CF-03	1	All users should be able to view the details of an orders.
CF-04	1	Inside each order, all users should be able to see a button for calling the server so that he can talk to the server assigned to a specific order when he cannot send the order completion information to the server for whatever reason.

Nonfunctional Requirements:

Identifier	Priority (Higher number indicates higher priority)	Requirement
CF-05	5	This application should be intuitive to use
CF-06	4	This application should be modular so that any new updates can be made easily in case new functionality is added.
CF-07	4	The application should be designed to be as clean as possible so that some non-necessary or repeated information or activity can be avoided.
CF-08	1	The application should be designed to be as compact as possible, which means it won't take up to much storage on devices

On-Screen Appearance Requirements:



2.5: Recommendation & Review System

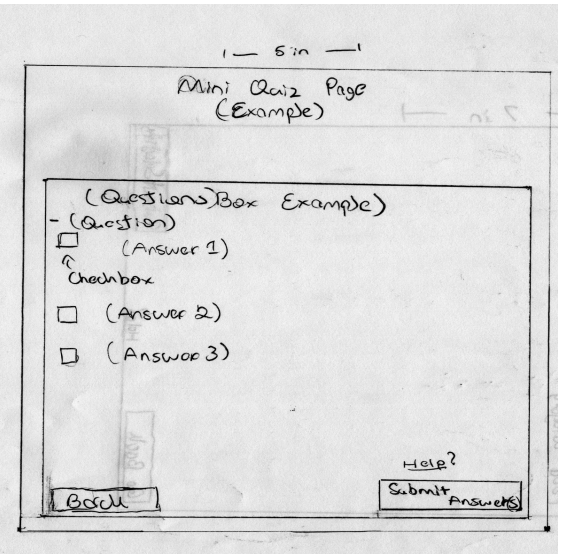
Functional Requirements:

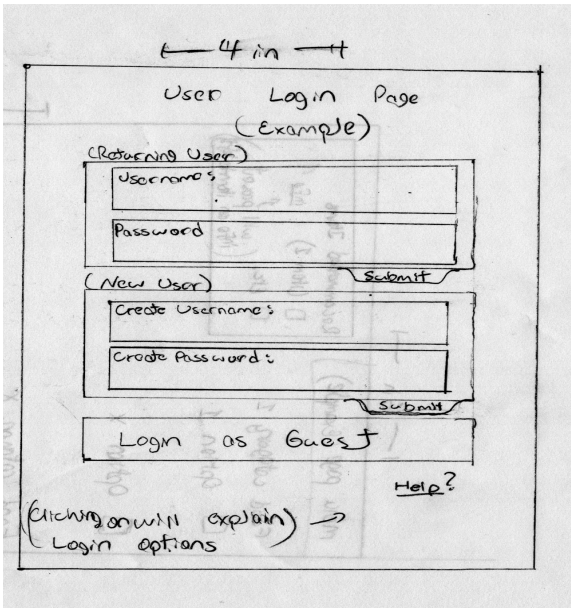
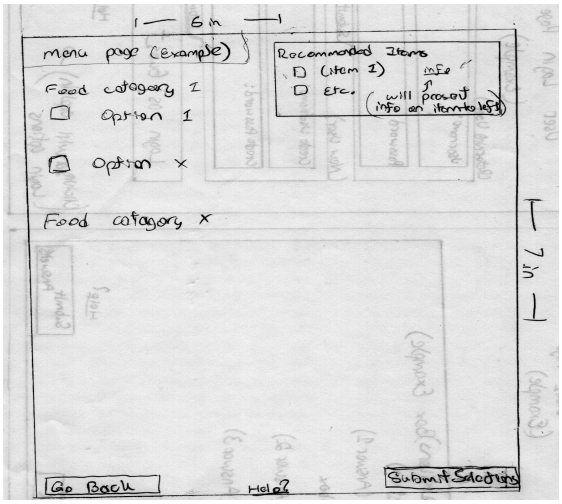
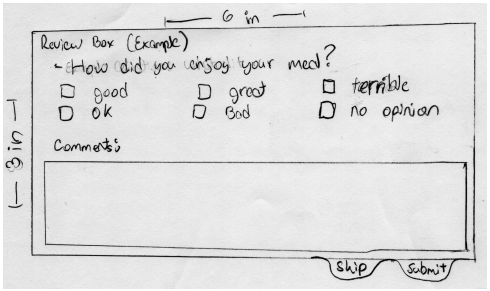
Identifier	Priority (Higher number indicates higher priority)	User Story
RR-1	1	All users should be given the option to rate their meal and customer service once they are done eating.
RR-2	1	Answers to reviews will be recorded in the database and will also be used to generate future meal recommendations for a particular user.
RR-3	1	Ratings and how often a meal is chosen will determine the popularity of the meal.
RR-4	1	Meal recommendations can be generated based on popular meals as well as previous meal choices or quiz answers of new users.
RR-5	1	As a customer, I should have the option to select a recommended meal or choose one myself. The recommendation system can be available as an option on the side menu of the application.
RR-6	1	As a new customer I would find it to be the most convenient if meals were recommended to me based on personal preferences (quiz answers).
RR-7	1	As a new customer, I should have the option to make an account (so that my quiz answers and past meal choices can be recorded for future recommendations).
RR-8	1	As a returning customer, I should have the ability to view foods that I may like based on my past meals.
RR-9	1	As a guest, I should only be able to see the restaurant's most popular meals.

Nonfunctional Requirements:

Identifier	Priority (Higher number indicates higher priority)	User Story
RR-10	3	Customers should be able to close the pop-up that requests their permission to take the meal recommendation quiz
RR-11	5	Customers should be prompted to log into their accounts if they want meals recommended to them
RR-12	5	The customer app and restaurant website should provide a list of most popular meals for users that do not decide to log in or take the meal recommendation quiz
RR-13	3	The meal recommendation quiz should not take longer than 1-2 minutes to complete
RR-14	5	The customer app and website should ask users for their permission to store information about their meals

On-Screen Requirements:

Identifier	Priority (Higher number indicates higher priority)	Requirement Description (Depicted Images Are Not Set In Stone)
RR-15	**1**	 Quiz new account holders have the option to take in order to optimize their recommended items in the menu

RR-16	**1**	 Customer login page with options to login as a returning customer, create an account as a new customer, or simply bypass as a guest customer.
RR-17	**1**	 Menu including recommendation section
RR-18	**1**	

		Rating System Box that will pop up at the end of a meal.

2.6: Ingredient Prediction System

Functional Requirements:

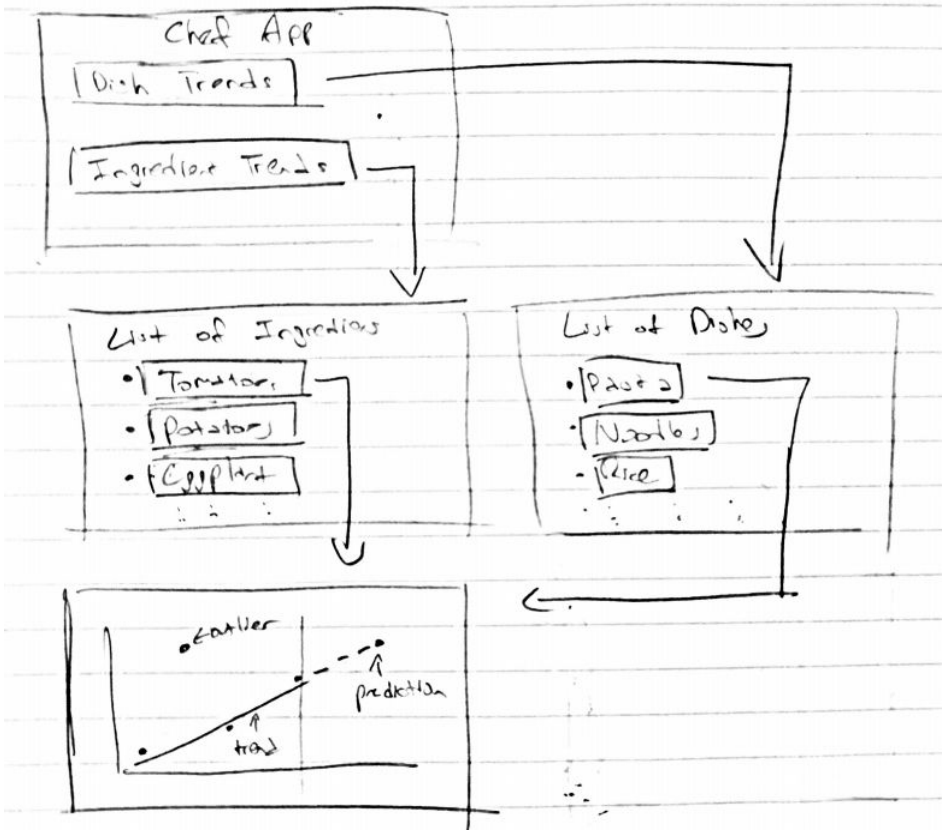
Identifier	Priority (Higher number indicates higher priority)	User Story
IP-1	5	As a chef, I want to have enough ingredients prepared for the dishes I need to cook beforehand so that I can focus on cooking the food and not on prepwork, so I want to know how much of each ingredient I should prepare.
IP-2	5	As a chef, I also don't want to I don't want to end up preparing too much of an ingredient either, since this indicates wasted time and food, since the food is most likely going to be tossed out.
IP-3	5	As a customer, I want to eat food made of fresh ingredients.
IP-4	5	As the owner, I want to know how much of each ingredient I need to purchase and how much this will cost me.
IP-5	1	As a chef, I want to know what the impact of modifying the recipes of certain dishes would be on the overall ingredient consumption.

Nonfunctional Requirements:

Identifier	Priority (Higher number indicates higher priority)	Requirement
IP-6	5	The popularity of dishes should reasonably follow a trend based on the popularity in the past. The ingredient prediction should take into account the demand of ingredients in the past.
IP-7	4	There may be outliers on certain holidays or other occurrences that we don't want to take into account when looking at overall trends.

IP-8	5	Since the amount of customers will change every day, the ingredient prediction should take into account the day of the week.
IP-9	3	While the popularity of dishes will follow a trend, over longer periods of time, the popularity may change drastically. To take this into account, we only want to look a few weeks into the past.
IP-10	2	As a chef, I do not want to prepare ingredients for dishes that are popular only late in the day while it is still early. The ingredient prediction should also take into account the time of day.
IP-11	1	The ingredient prediction should take into account other parts of the overall restaurant automation system such as reviews and reservations.

On-Screen Requirements:



2.7: Data System API

Functional Requirements:

Identifier	Priority (Higher number indicates higher priority)	Requirement
DS-01	3	The application would collect all quizes results and upload them on the database
DS-02	2	None two identical accounts are allowed
DS-03	2	The manager would be able to check the storage and the usage of the inventory
DS-04	5	After getting a request, the application would collect data from the database and return the results
DS-05	3	When the amount of certain ingredient is low, an alert would be raised and broadcasted
DS-06	5	After an order is placed, the amount of related ingredients would be reduced
DS-07	5	Store customer account information
DS-08	4	Customers can only access their own history

Nonfunctional Requirements:

Identifier	Priority (Higher number indicates higher priority)	Requirement
DS-09	3	Use clear relationships between tables while not making tables unreasonably large.
DS-10	3	Clear API that is intuitive to use.
DS-11	4	Make specific endpoints to promote separation of concerns.
DS-12	2	Easily expandable

2.8: Restaurant Website

Functional Requirements:

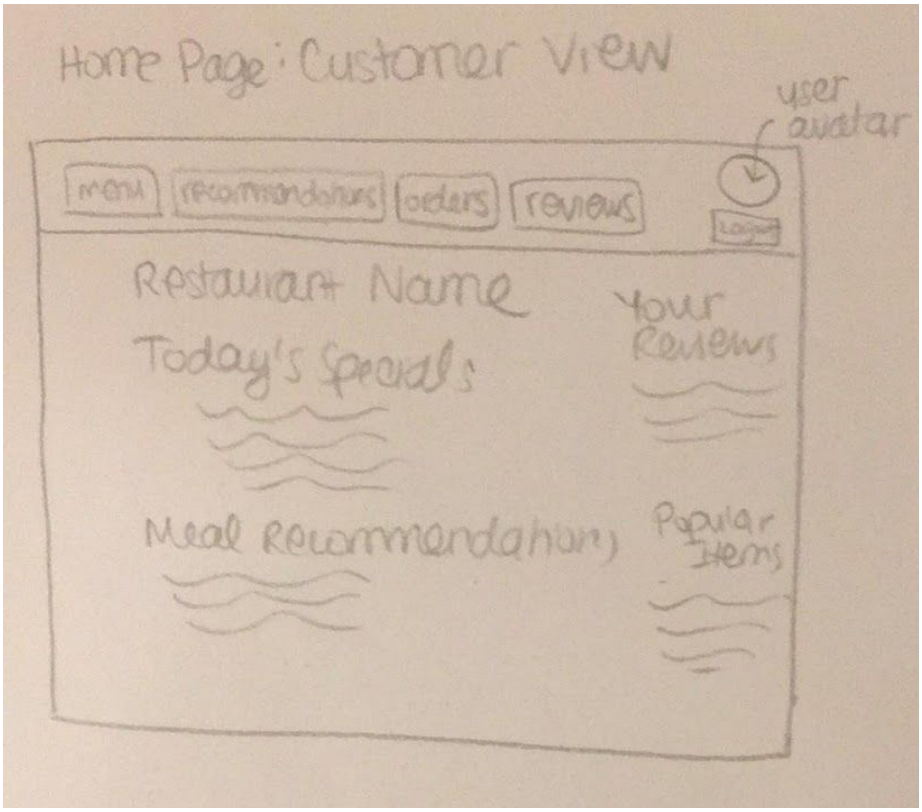
Identifier	Priority (Higher number indicates higher priority)	Requirement
RW-1	5	The website should contain information about the menu and most popular food items.
RW-2	4	Customers should be able to make reservations or take out/delivery orders (if applicable) from a page on the website.
RW-3	3	Direct order from the website would send the order description to the chef app.
RW-4	3	The website should allow users to sign in to their accounts in order to use the meal recommendation feature.
RW-5	2	The website should allow users to rate their meals and customer service received.
RW-6	5	The website should communicate with the data system via the same REST API as the mobile apps.
RW-7	4	Restaurant owner should be able to access the admin page that would display relevant statistics, such as earnings, most popular foods and ingredient usage.
RW-8	1	The website admin page should allow the owner to modify the menu, change prices, and enable/disable features.
RW-9	3	The website admin page should interface with the ingredient prediction system to show predictions of ingredient usage based on popularity trends.

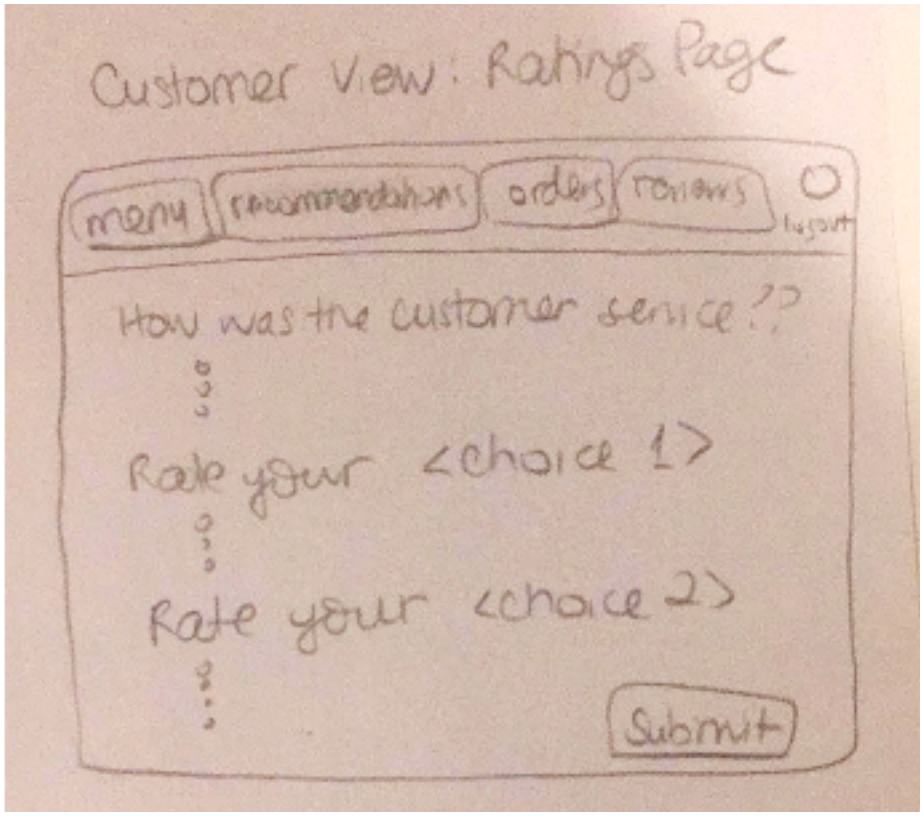
Nonfunctional Requirements:

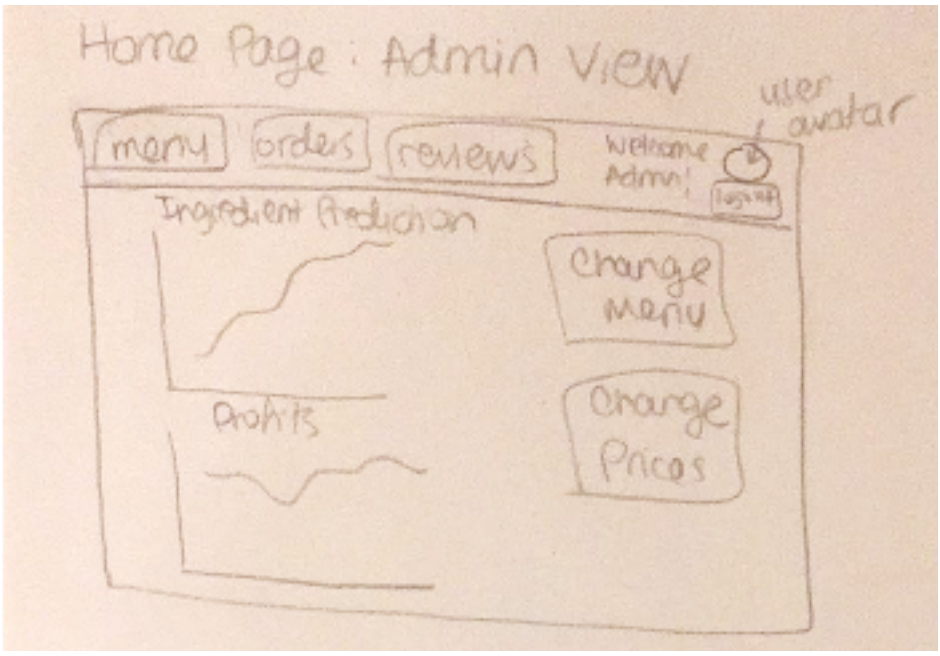
Identifier	Priority (Higher number indicates higher priority)	Requirement
RW-10	4	The website should be lightweight and not rely on too many

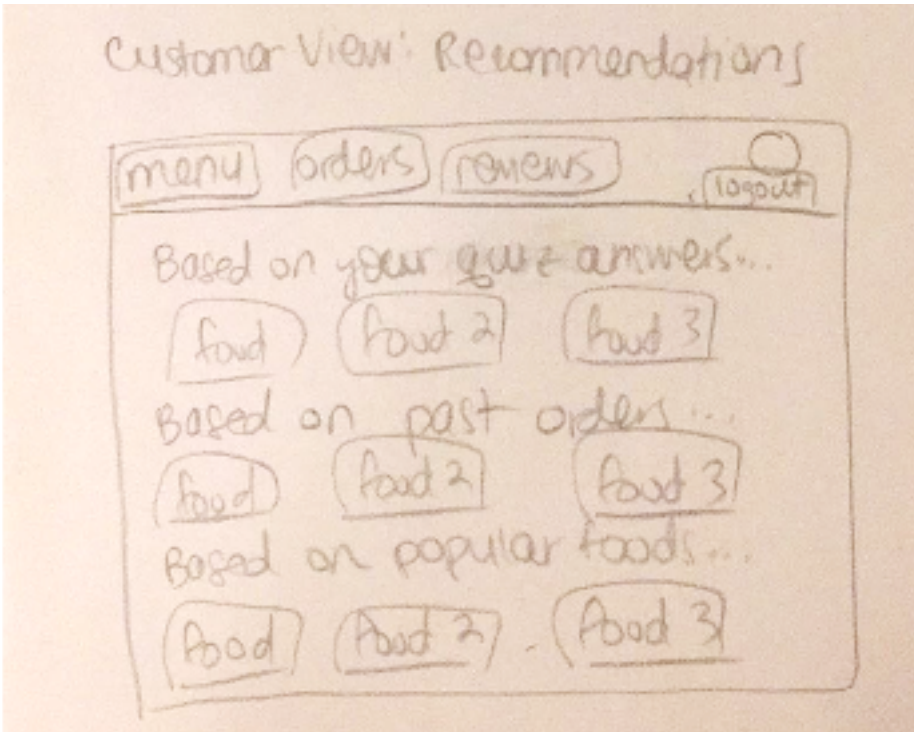
		3rd party libraries
RW-11	5	The website should be connected to the same database that the ecosystem of apps is connected to
RW-12	4	The website UI should be intuitive and user friendly
RW-13	5	The website admin page should only be accessible by a user that has logged in with admin credentials.
RW-14	4	The website should expose API endpoints to be used by the apps

On-Screen Requirements:




Customer View: Ratings Page
menu
recommendations
orders
reviews
logout
How was the customer service??
Rate your <choice 1>
Rate your <choice 2>
Submit


Home Page: Admin View
user avatar
menu
orders
reviews
Welcome Admin!
logout
Ingredient Reduction
Change Menu
Profits
Change Prices



Section 3: Project Management

Functionality

Our team will be split up into subdivisions of groups of 1, 2, or 3 people. The subdivisions that we believe will require more work will have 3 people, while simpler tasks may have only one. Here is the breakdown of the project tasks and who is assigned to each one:

#	Feature Subteams	Names
1	Mobile Applications	• Taras Tysovskyi • Lieyang Chen
2	Restaurant Website	• Arushi Tandon
3	Facial Recognition System	• Hongpeng Zhang
4	Database and API Design	• Yuwei Jin • Chris Lombardi • Taras Tysovskyi
5	Recommendation and Review Systems	• Chris Gordon • Seerat Aziz

6	Ingredient Prediction and Reservation Systems	• Alex Gu • Hongpeng Zhang

**Note that some developers are listed twice, we believe having one "expert" on two groups will help integrate the groups together. For example, Taras will integrate the Mobile app and the Database/Accounts together.

Timeframe

API: before working on everything else we need to agree on the API endpoints our server will expose. We plan for this to be done in the first 2 weeks. At this point front-end teams can start developing the apps and websites, while the back end team can work on actually writing the code that implements the API.

Mobile Applications: we are hoping to have basic customer, table, chef and serve apps done within the first 5 weeks. Since all the heavy lifting is done by the server, the apps will be just a front end that makes HTTP calls to the server and displays the data it receives from it. Chefs app is the simplest one to make, since it only has one view that displays the current orders. Table and Customer apps share a lot of the same views (menu, order, recommendations) which would allow us to reuse a lot of the same code.

Facial Recognition System: We first need to test the accuracy of a third-party software that can be integrated with our Android app (2-3 weeks). If the accuracy rate of the third-party software is low, then we will use OpenCV to implement this (2-3 weeks).

Recommendation and Review Systems: We first need to define our menu items and identify attributes that can be used in the quiz (1 week). Then we will develop an algorithm that takes the user's selected meals, "favorited" meals, and meal reviews to generate more accurate suggestions for the future (2-3 weeks). Moreover, we also have to design a review system that would work best for this subproblem (1 week).

Ingredient Prediction System: The bulk of the logic that the ingredient prediction system will use is based off of trends in order history, so it will make heavy use of the central database. As a starting point, the database structure will have to be well defined (~1 week). Beyond that, implementation and refinement of the prediction algorithm will be the next step (2 weeks). Any extensions to this system should be relatively simple, with only some minor tuning and adjustments needed (1 week).

Customer Reservation Systems: The customer reservation system is relatively simple, only requiring some database entries to be made on the back end. (1 week) The majority of work involving the reservation system lies in interfacing with other aspects of the overall restaurant automation system (~1-2 weeks).

Restaurant website: The overall structure of the restaurant website needs to be established first. The static elements of the restaurant website, such as the menu of the restaurant and the ordering and reservation pages will be completed first (2 weeks). The basic functionalities of the admin console, such as the statistics and ingredient

prediction pages will be the next step (2 weeks). Since the website will provide the front-end UI for many of the component systems, interfacing with the various systems of the Restaurant Automation system will be completed throughout the semester as they achieve various degrees of completion. Additional features on the admin console, such as editing the menu will be stretch goals (2+ weeks).

Section 4: References

1. Why W8 - Fall 2018 Restaurant Automation Project
2. FoodEZ - Spring 2015 Restaurant Automation Project
3. Professor Marsic's Restaurant Automation Project Description (https://www.ece.rutgers.edu/~marsic/books/SE/projects/Restaurant/)
4. Spyce (Boston restaurant with a robotic chef) - https://www.greenbiz.com/article/full-service-automation-restaurants-changing-food-industry