Report #1:

ChefBoyRD

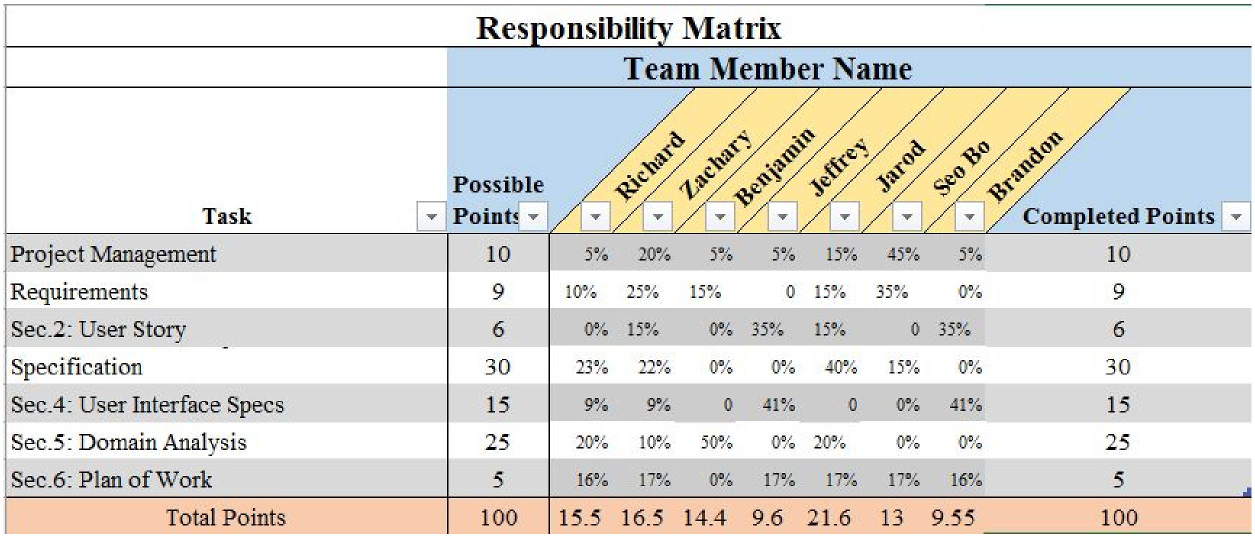
Restaurant Automation

GROUP #6:
Richard Ahn
Zachary Blanco
Benjamin Chen
Jeffrey Huang
Jarod Morin
Seo Bo Shim
Brandon Smith

GITHUB & WEBSITE:
https://github.com/ZacBlanco/ChefBoyRD
http://blanco.io/ChefBoyRD

SUBMISSION DATE:
February 19, 2017

Individual Contributions Breakdown



Responsibility Matrix									
	Team Member Name								
Task	Possible Points	Richard	Zachary	Benjamin	Jeffrey	Jarod	Seo Bo	Brandon	Completed Points
Project Management	10	5%	20%	5%	5%	15%	45%	5%	10
Requirements	9	10%	25%	15%	0	15%	35%	0%	9
Sec.2: User Story	6	0%	15%	0%	35%	15%	0	35%	6
Specification	30	23%	22%	0%	0%	40%	15%	0%	30
Sec.4: User Interface Specs	15	9%	9%	0	41%	0	0%	41%	15
Sec.5: Domain Analysis	25	20%	10%	50%	0%	20%	0%	0%	25
Sec.6: Plan of Work	5	16%	17%	0%	17%	17%	17%	16%	5
Total Points	100	15.5	16.5	14.4	9.6	21.6	13	9.55	100

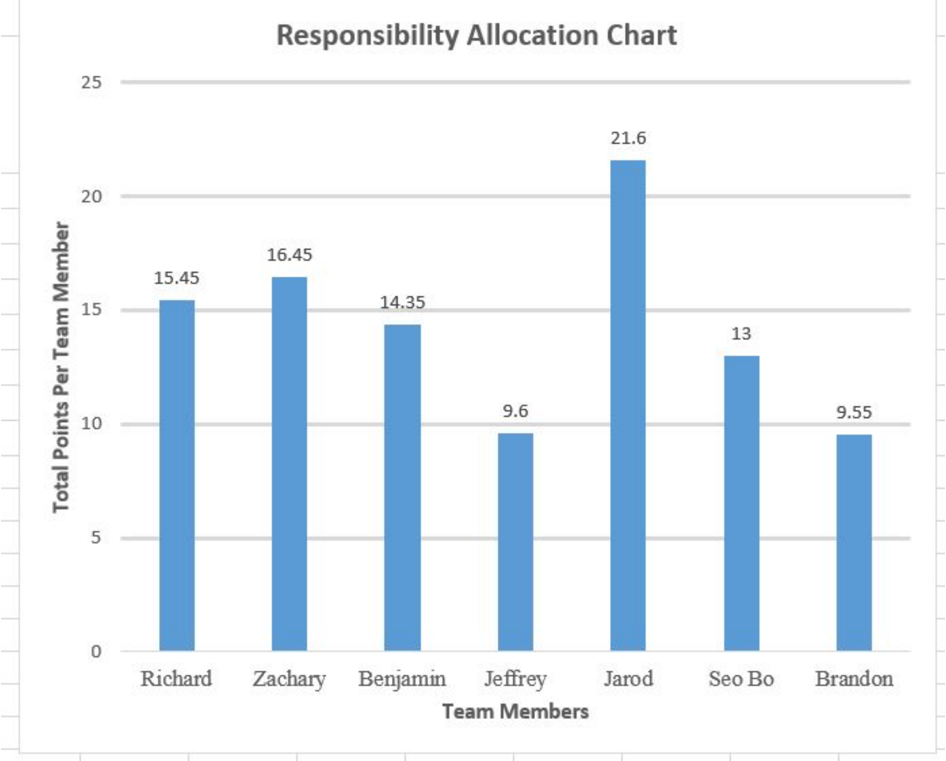


Table of Contents

1. Customer Statements of Requirements

a. Problem Statement

As an established restaurant, we wish to incorporate more technology to help run the restaurant more smoothly and to help us compete in the saturated restaurant market. Our understanding is that with the help of technology we are able to track information on large scales. The broad problem that restaurants like us face is creating meaningful contextual relationships using this information, which can then be analyzed to improve the restaurant overall. We would like to have the proposed software application address some specific problems. The problems we would like to address are improving the customer experience and increasing the efficiency of running the restaurant and bringing more customers, which ultimately leads to the financial success of the restaurant.

PREDICTING INGREDIENT USAGE

Chef:

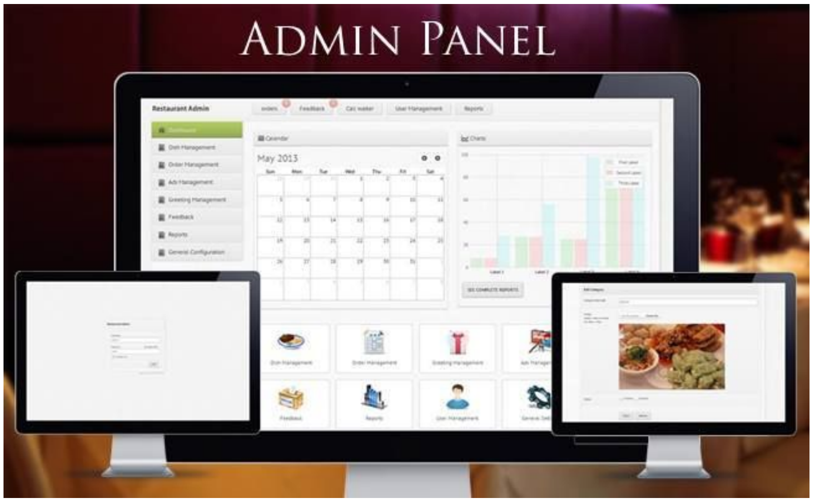
As a chef, one of my responsibilities is gauging the amount of food that needs to be prepared for a typical day. This plays a crucial role in the day-to-day operation of a typical food business. As the restaurant gets busier throughout the day, our kitchen has to predict which dishes will be ordered and begin preparing them in advance to keep up with customer demand. However, if we predict incorrectly, we risk preparing too many dishes and wasting both time and money with the wasted dishes. According to the NRDC, approximately 40% of food gets wasted, and a major contributor to that number is restaurants.[1] Our kitchen can attempt to prepare fewer dishes, but if we get too much business and have not made enough dishes, our customers will have to wait longer to receive their orders. This makes customers unhappy and discourages them from returning to our restaurant in the future.

With prior experience, it is possible to predict (to a certain degree) which dishes and how much food should be prepared from day to day. However, this method is very prone to error for even experienced chefs. A system that can analyze our restaurant's metrics and order histories would be extremely useful, as it will use this data to determine the amount of food we should prepare for a given day. This would not only reduce our restaurant's waste but also increase our profit margins. This allows us to invest more into our business and make our customers and employees happier.

As a chef, if I am constantly busy around the kitchen and preparing food, we would like this predictive system to not only be available within the software application; but also available to use and communicate with through a voice-activated service such as Amazon Alexa. Being able to utilize this feature through Amazon Alexa's voice services means I will be able to work in the kitchen while receiving updates and predictions from the service on the day's food production. It should be able to provide the same type of estimates or similar to the ones available by the software application.

Manager:

We would like to reduce waste and excessive use of ingredients, and what would help is we can view exactly how much of each individual ingredients were used over different time spans. With the aid of this system, chefs can more accurately estimate the ingredients needed on a particular day, expediting the food preparation process and minimizing the waiting time for customers to receive their orders. We would like the application to give us detailed analyses and reports about the restaurant's order history over the course of days, weeks, months or years. We would like the application to display order history information in an easy-to-view format, such as dashboards with graphs and charts [Fig 1a-1]. This system is also helpful for me as a manager so that I can understand the day-to-day performance and long-term trends of the restaurant. The ability to see this data helps me determine overall business trajectory and make informed decisions to keep the restaurant financially successful.



[Fig 1a-1] Administrative Panel

We would also like the application to use the accumulated data to give the chef an estimate - an actual number - of the type and quantity of specific ingredients that are required throughout a typical day. The estimates should be able to tell how much of certain ingredients will be consumed within certain hours of business operation. Ultimately, the quantities for food preparation will be determined by the chef, but the system's estimate should serve as a general benchmark from which the chef can base his decisions.

Overall, this should result in a more efficient use of food resources, especially with time-sensitive ingredients. It can also allow us to prepare food in advance with more time-fragile, but fresher, ingredients.

We do however recognize that one of the drawbacks of using the predictive software is the lack of reliability in its initial stage due to insufficient data. Therefore, we would like the system to provide a simple interface to allow us to sync our historical records from our existing POS system to the new one. This way, we can quickly get the prediction software running as best as possible. This preliminary step of obtaining data is extremely important because a predictive software that is unreliable for the first few weeks or even months is highly undesirable.

Overall, this system can help us serve our customers faster by telling us how much food to prepare at a given time. This system will also allow our business to waste fewer ingredients, resulting in greater profits for our restaurant. Our managers will also be provided with a clear and simple visualization of their restaurant's financial status to make their own analyses and business decisions. Eventually, the profits from this system can be invested back into our business to provide higher quality ingredients, higher employee salaries, or even lower prices for customers.

CUSTOMER FEEDBACK

Waiter:

Customer feedback is very important in how I'm presenting myself to my customers. Some nights I receive awful tips, or no tips at all and I have no idea why this is the case. Though every customer is different, if I had some concrete feedback to work with, I could improve my customer service skills or communicate any problems to the management. Different customers use different signals to get my attention when they want something. Usually people will wave me over or call out when I pass. Sometimes when the customer thinks that I am being inattentive or am ignoring them, it is just that I didn't recognize that the signal they were using indicated that they wanted my attention. If people communicated more details about their expectations and the signals they used instead of simply rating the service as excellent or poor, it would allow me to learn these new signals and prevent me from repeating my mistakes.

Chef:

When I test new dishes, it would be very helpful to know if people like them. It's never easy to find out what exactly the customers are thinking, but if people are leaving unhappy or are absolutely raving about my dishes, they usually make their opinions clear. My waiters and the manager already do a good job of communicating what the customers think about my dishes, but it would be nice to have something concrete to work with.

Customer:

The waiter usually asks us to complete an online survey about our experience when we receive the check. The website is written on the receipt, and there are usually prizes, but I've never actually completed one. Entering the website on mobile is cumbersome so I prefer to respond on a home computer. However, by the time I've gotten home I've usually forgotten about the receipt. Sometimes going to a restaurant isn't the last thing I do while I am out and I don't remember the details about my dining experience by when I return home I would be happy to offer my opinion if the restaurant provided a solution so that customers could take the survey before leaving the restaurant.

Manager:

A critical task that I have to perform as a manager is finding the source of problems that I see. If the restaurant is only half-full on a Friday night, I have to find the cause behind the problem. Could it be an employee? Is there something wrong in the kitchen? This is where customer feedback is useful, where I can learn the perspective of the customers and act accordingly to find solutions. In fact, it would be even more helpful to have contextual information about a specific feedback. Who was on shift during this time? What did they order?

I need to have improved methods of collecting feedback, because the current methods do not suffice. Verbal feedback is easy to understand, but difficult to quantify. Online surveys require patrons to visit a website and submit a specific form that restricts users entry. Though they are detailed, they are unpopular because they are so long!

Our previous implementation was to use a suggestion box, but the results were underwhelming. Very few customers offered suggestions and the results were disorganized. We have switched to an online form that requires customers to create an account to participate. To compensate for the added inconvenience, we provide incentives to respond. These prize rewards are also advertised on receipts. We have received more feedback as a result, but the increase in responses does not justify the cost of the prizes.

I want to improve the quantity and quality of the feedback collected, perhaps make feedback submission convenient to the customer and easily trackable [Fig 1a-2]. Then once I can receive the feedback, I have to make business decisions based on them and eventually report them to the restaurant's owner. Tracking feedback helps me by providing evidence for my decision-making process. If I need to justify my managerial decisions to the owner, it would be extremely helpful to have supporting data, such as the customer feedback.

[Fig 1a-2]Word Cloud

I also want to make sure that feedback is sincere and addresses a problem. I would like the ability to keep track of feedback and determine whether the feedback is useful or not.

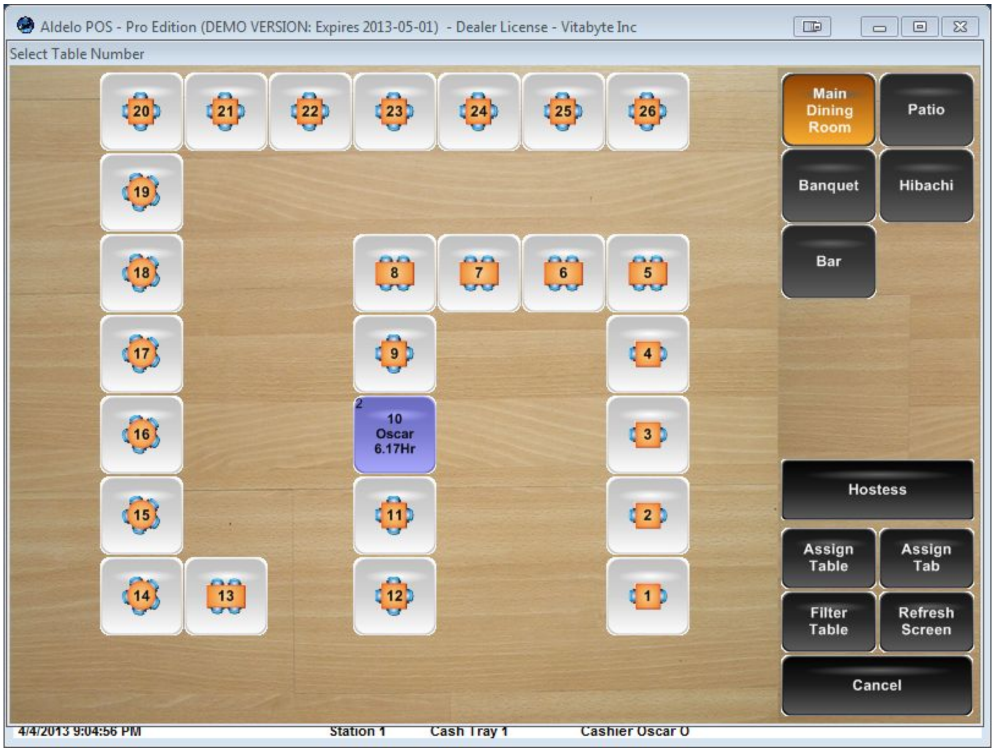
If a customer is blatantly dissatisfied with a service, in my experience they usually communicate this directly to me or the owner. However, this situation can allow minor or subtle problems to exist undetected. What if customers have feedback that is not substantial enough to warrant calling me or a staff member over?

This could be something like: the table is wobbly, the light is too dim at times, the seats feel a little hard. Individually, these bits of feedback may not be important, but collectively they could make a difference. Some of these items may not be important in the big scheme of things, but making these subtle changes are key to providing an excellent customer experience that exceeds expectations.

TABLE MANAGEMENT

Hostess:

As a hostess, one issue that I constantly deal with is seating customers. Each table seats a different number of people and different-sized groups of people arrive all at different times. Updating this is a tedious, error-prone process, especially during peak hours when the restaurant is busiest. I want to be able to quickly assign people to tables, and change the table locations according to how it looks in the restaurant [Fig 1a-3].



Aldelo POS - Pro Edition (DEMO VERSION: Expires 2013-05-01) - Dealer License - Vitabyte Inc

Select Table Number

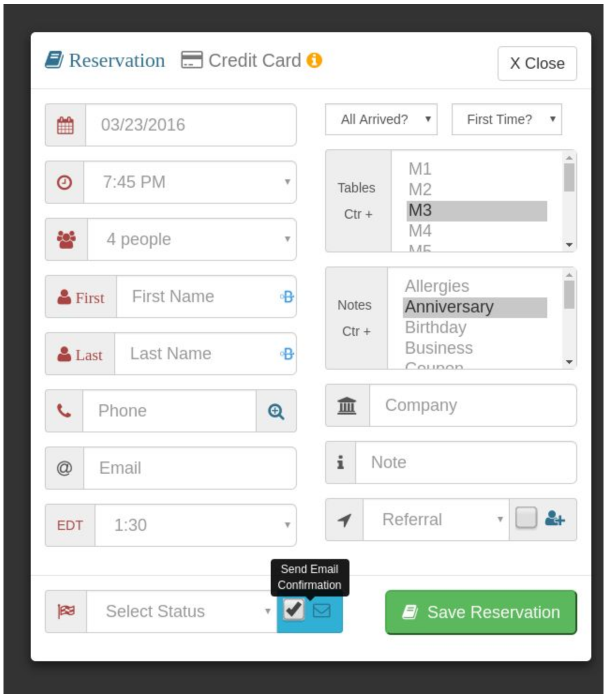
4/4/2013 9:04:56 PM Station 1 Cash Tray 1 Cashier Oscar O

[Fig 1a-3] Table View

Customer:

Long wait times at restaurants are inconvenient and annoying to deal with. There are times I’ve had to wait outside because there were too many people lined up inside the restaurant.

But that’s what I always reserve my spot at restaurants now. I usually call in, or make a reservation online. Even then, I do get a wait time estimate but I can wait in the comfort of my home and plan accordingly. But some of these online reservation systems are not convenient at all, I have to enter my full name, address, credit card number, when I just wanted to reserve a spot [Fig 1a-4]. And there are times where after I reserve a spot and come into the restaurant, I find out that my spot wasn’t reserved at all. It would be nice if I could be more confident that the online-reservation is working. A simple confirmation message with a generic message just doesn’t do it for me.



03/23/2016

7:45 PM

4 people

Tables Ctr +: M1, M2, M3, M4

Notes Ctr +: Allergies, Anniversary, Birthday, Business

EDT 1:30

[Fig 1a-4] Reservation Form

Also, the waiting times are not very accurate and can be inconvenient. I'll be told that I have to wait an hour, then when I arrive on time, I'm told I have to wait another 30 minutes! It would be nice to see where I am in line, and how many people are before me.

Overall, I just want to ensure that I'm not wasting my time when I decide to eat out at a restaurant. I'd like to have short, accurate waiting times, with minimal errors in how I'm being seated.

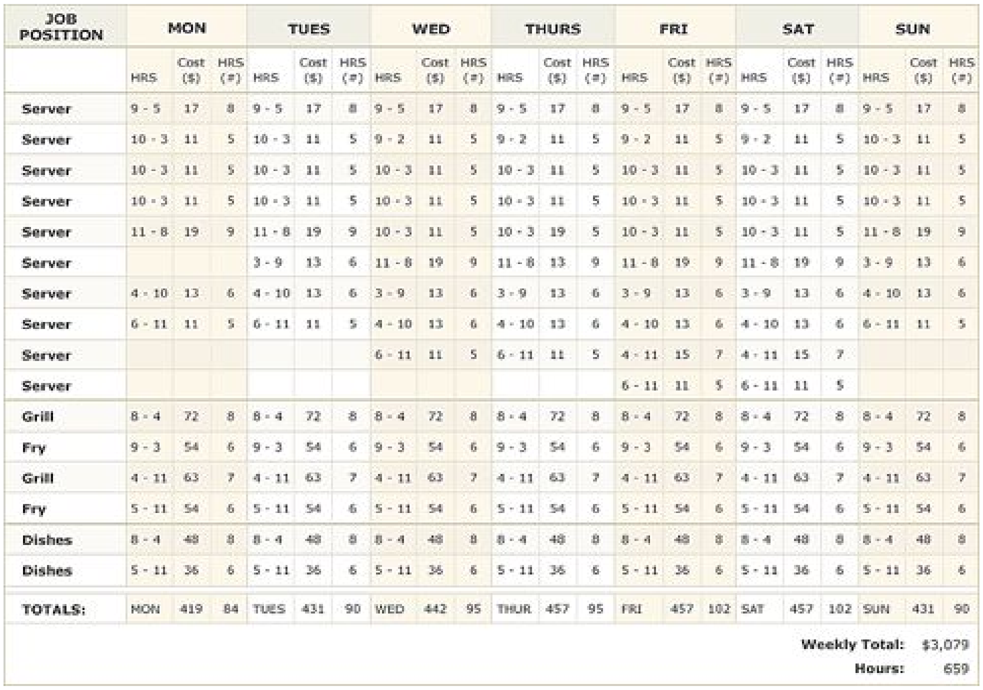
SHIFT MANAGEMENT

Manager:

As a manager, I have to maintain all of the logistics of the business. This often means that I have to handle tasks such as having repetitive conversations with employees about when they can come into work. In addition, there are times I have to call them up to see if they are available on specific days. I wish to put more responsibility in the hands of the employees.

The big problem is that I have to manage all this and tell employees their shift times [Fig 1a-5]. It is my responsibility to ensure the restaurant is staffed, but sometimes this can lead to ambiguity. For example I may forget to tell an employee they are scheduled for a specific time. Or they may claim that I did not notify them. Also if employees cover for each other, I'd like this to be clear to both parties and me. Overall, an organized system will reduce business liability.

Something like an online shift scheduling tool will be useful. So if shifts are changed, or if an employee has an emergency, I can notify all employees of an open shift. Internet access has become increasingly more common in the past decade, so it is now a more reasonable request to ask employees to check online for their shift times.



JOB POSITION	MON			TUES			WED			THURS			FRI			SAT			SUN		
	HRS	Cost ($)	HRS (#)	HRS	Cost ($)	HRS (#)	HRS	Cost ($)	HRS (#)	HRS	Cost ($)	HRS (#)	HRS	Cost ($)	HRS (#)	HRS	Cost ($)	HRS (#)	HRS	Cost ($)	HRS (#)
Server	9 - 5	17	8	9 - 5	17	8	9 - 5	17	8	9 - 5	17	8	9 - 5	17	8	9 - 5	17	8	9 - 5	17	8
Server	10 - 3	11	5	10 - 3	11	5	9 - 2	11	5	9 - 2	11	5	9 - 2	11	5	9 - 2	11	5	10 - 3	11	5
Server	10 - 3	11	5	10 - 3	11	5	10 - 3	11	5	10 - 3	11	5	10 - 3	11	5	10 - 3	11	5	10 - 3	11	5
Server	10 - 3	11	5	10 - 3	11	5	10 - 3	11	5	10 - 3	11	5	10 - 3	11	5	10 - 3	11	5	10 - 3	11	5
Server	11 - 8	19	9	11 - 8	19	9	10 - 3	11	5	10 - 3	19	5	10 - 3	11	5	10 - 3	11	5	11 - 8	19	9
Server				3 - 9	13	6	11 - 8	19	9	11 - 8	13	9	11 - 8	19	9	11 - 8	19	9	3 - 9	13	6
Server	4 - 10	13	6	4 - 10	13	6	3 - 9	13	6	3 - 9	13	6	3 - 9	13	6	3 - 9	13	6	4 - 10	13	6
Server	6 - 11	11	5	6 - 11	11	5	4 - 10	13	6	4 - 10	13	6	4 - 10	13	6	4 - 10	13	6	6 - 11	11	5
Server							6 - 11	11	5	6 - 11	11	5	4 - 11	15	7	4 - 11	15	7			
Server													6 - 11	11	5	6 - 11	11	5			
Grill	8 - 4	72	8	8 - 4	72	8	8 - 4	72	8	8 - 4	72	8	8 - 4	72	8	8 - 4	72	8	8 - 4	72	8
Fry	9 - 3	54	6	9 - 3	54	6	9 - 3	54	6	9 - 3	54	6	9 - 3	54	6	9 - 3	54	6	9 - 3	54	6
Grill	4 - 11	63	7	4 - 11	63	7	4 - 11	63	7	4 - 11	63	7	4 - 11	63	7	4 - 11	63	7	4 - 11	63	7
Fry	5 - 11	54	6	5 - 11	54	6	5 - 11	54	6	5 - 11	54	6	5 - 11	54	6	5 - 11	54	6	5 - 11	54	6
Dishes	8 - 4	48	8	8 - 4	48	8	8 - 4	48	8	8 - 4	48	8	8 - 4	48	8	8 - 4	48	8	8 - 4	48	8
Dishes	5 - 11	36	6	5 - 11	36	6	5 - 11	36	6	5 - 11	36	6	5 - 11	36	6	5 - 11	36	6	5 - 11	36	6
TOTALS:	MON	419	84	TUES	431	90	WED	442	95	THUR	457	95	FRI	457	102	SAT	457	102	SUN	431	90

Weekly Total: $3,079
Hours: 659

[Fig 1a-5] Employee Schedule

An organized online system like this would also be useful because I can keep track of everyone's hours easily. I will be able to calculate payroll automatically. Typically the software used for managing shifts and payroll are offline. This means I have to manually enter in shifts, which again is very repetitive.

Also, keeping a record of employee shifts as well as a record of orders will be useful in tracking when someone makes an error. If errors are being made continuously, either with the order or server, an organized record can help keep track of employee underperformance.

Employee:

I need a convenient way to be able to see my shifts. I may forget when I have a shift scheduled. I may need my shift covered for a non-emergency or emergency reason, and I may want to pick up more shifts in a week. It would be extremely convenient to have a website where I can access all this information and make changes. This means I can access this on my phone or desktop whenever I want to, instead of having to call my manager every time I want to make a change.

b. Glossary of Terms

Logistics: related to the coordination of different employees of the restaurant in order to provide a desired business goal.

Chefs: A critical member of the kitchen staff who directs kitchen activities including food preparation, cooking, and presentation. Also responsible for creating new recipes.

Customer: A person who uses, or intends to use the services and products of the restaurant.

Customer Satisfaction: A measurement of whether a patron’s enjoyment or satisfaction of the restaurant experience meets or exceeds standards. This is impacted by quality of service, food quality, wait time, and overall experience.

Dish(es): A complete food item produced by the kitchen staff.

Feedback: Information that signifies, to any level of detail, a positive, neutral, or negative response to products or services.

Food preparation: A process executed by the kitchen staff between the time the food ingredients are stored, and cooked.

General Employee: A term used to refer to any sort of employee working at the restaurant. This includes but is not exclusive to: waiters, hosts, cooks, manager, etc.

Host/hostess: restaurant employee in charge of greeting customers at the door, and seating arrangements in general

Ingredients: The edible components of a prepared dish before being cooked.

POS (Point-of-Sale): A system that a modern restaurant may use to manage its customer orders,process them to collect payment,and send the orders to the kitchen. These systems often include the cash registers and monitors that employees use to enter in orders.

Reservation: A customer declares and notifies the restaurant that they are arriving at a specified date and time. They usually must be made with customer’s name, contact info, date/time, and number of guests.

Restaurant Portal: Where employees can access their role-specific functions in the software.

Employee Portal: Where individual employees can login and manage their shifts, contact information, non-role specific information.

Server: This term will be used to refer to the computer hardware system that hosts the software. This does not refer to the waiter/waitress who is commonly called a server.

2. User Stories

Higher size values indicate a longer expected implementation time.

Role: Manager (ST-MA)

Identifier	User Story	Size (1-10)
ST-MA-1	I do not need to manually enter orders or ingredient lists apart from the POS system	7
ST-MA-2	I can pull up economic information (net gain, net loss, revenue, expenses, etc.) at ease	6
ST-MA-3	I can see the inventory that we currently have to make sure that we have enough ingredients and/or materials to continue the quality of service.	5
ST-MA-4	I can receive alerts pertaining to abrupt critical information such as negative feedback, a negative revenue trend, etc	4
ST-MA-5	I can see the customer feedback on specific employees and dishes served	8
ST-MA-6	I can view feedback the detailed feedback that customers send in	5
ST-MA-7	I can see customer feedback organized into positive/negative categories	7
ST-MA-8	I can see who is working at the moment	2
ST-MA-9	I can find openings in the schedule and ask employees to come and work in those openings	4

Role: Chef (ST-CH)

Identifier	User Story	Size
ST-CH-1	I can receive recommendations on how to prepare the proper amount of ingredients for the restaurant at a certain time	10

ST-CH-2	I can view the current inventory levels to ensure there is enough for the next dish/day	5
ST-CH-3	I can see and update inventory levels as needed and the manager can be notified	4
ST-CH-4	I can see a historic record of what and when dishes have been made	5
ST-CH-5	I can view customer feedback about the meals I prepare	7

Role: Host (ST-HO)

Identifier	User Story	Size
ST-HO-1	I can check in both customers that made reservations and customers waiting in line, and then update table availability	5
ST-HO-2	I can cancel a reservation and remove customers from the queue list	4
ST-HO-3	I can see all current reservations and their phone number in case I need to call them	4
ST-HO-4	I can manually add an open reservation, specifying the customer's info and arrival info.	2
ST-HO-5	I can use a generated reservation number to refer to a customer's reservation.	2
ST-HO-6	I can easily keep track of moved and merged tables	7
ST-HO-7	I can see the current estimated waiting time for customers in line, to give customers an estimate.	5

Role: Customer (ST-CU)

Identifier	User Story	Size
ST-CU-1	I can reserve a table with ease	4

ST-CU-2	I can cancel a reservation without reason	3
ST-CU-3	I can receive a confirmation to make sure that the information that I provided for the reservation is accurate.	2
ST-CU-4	I can get a text confirmation of my reservation so that I can be confident my reservation was made	4
ST-CU-5	I can be seated in a timely fashion when I arrive at the establishment	3
ST-CU-6	I can easily and privately submit feedback about my experience at any time	8
ST-CU-7	I can easily leave a rating for my overall experience	3
ST-CU-8	I can leave detailed feedback to the restaurant	6

Role: General Employee (ST-GE)

Identifier	User Story	Size
ST-GE-1	I can see my work shift and hours worked in daily, weekly, monthly, yearly format as well as wage.	6
ST-GE-2	I can see who else is working at the same time that I will be working	3
ST-GE-3	I can request absence from work	3
ST-GE-4	I can post my shift and take coverages for other people	4
ST-GE-5	I can easily log into my account in the restaurant portal	3

3. Functional Requirements Specification

a. Stakeholders

One of our major stakeholders are restaurant managers and chefs. Since their goal is to raise the efficiency of their restaurant and maximize profits, our product would be very appealing to managers. Chefs in particular will find the freedom of having a voice-operated service that tells them what ingredients they need to prepare very useful.

Some equally important stakeholders also include the companies that developed the POS systems that our target restaurants use. These companies would be highly interested in a product that plugs in and adds powerful statistical analysis to their POS systems. Some popular POS systems include Vend or Bindo.

Dining customers will also find our product attractive since we provide the feature of allowing reservations to be made quickly online. In fact, our whole system is designed to reduce the wait time for all of these customers, so we believe that customers will be very supportive of our product.

Overall, we believe that all participants in the flow of a restaurant will benefit from our system.

b. Actors and Goals

Chef:

Chefs are initiators. Their primary goal is to initiate a call through either a physical interaction of the device or through the voice-operating service to give them a list of orders or ingredients to prepare.

Manager:

Managers are initiators. Their primary goal is to initiate a call to give them various statistics of the restaurant’s order history for their own analysis. Additionally, they also have access to see which orders need to be prepared.

Order:

Orders are participators. They’re passive actors that are only used as data for our use cases.

Host:

Hosts are mainly participators. Their primary goal is to manage customer interaction before they arrive, and just when they arrive at the restaurant. This includes setting up reservations and table management.

General Employee:

General employees are initiators. They initiate access into the employee portal, where they can manage their work shifts.

c. Use Cases

i. Casual Descriptions

Food/Dish Use Case (FD-UC)

FD-UC1

- Manager or System Sync Settings <<initiates>> the use case
- Current Day's Order Data <<participates>> by providing order data
- Order Data Database <<participates>> by providing and receiving order data and the results of the prediction algorithm
- Administrative Portal <<observes>> by receiving access to the order data and prediction algorithm results

Throughout the day, data relating to the dishes ordered and the associated times is tracked. At the end of the day, this information can be added to the database used for predictive purposes by a manager, or by the program itself. The current data in the database is compiled with the new data and is then analyzed by the prediction algorithm. The results of this process are stored back into the database.

FD-UC2

- Manager or Chef <<initiates>> the use case
- Current Day's Order Data <<participates>> by providing order data
- Order Data Database <<participates>> by providing historical order data
- Administrative Portal <<observes>> by receiving the historical order data

As the data from orders is tracked on a day-to-day basis as a chef or manager it is useful to be able to view historical trends on this data. It can be used not only to make predictions but to make business decisions in terms of marketing and designing the menu for the establishment. By providing easy access to historical data the research and investment required in order to make the most informed decision is lowered.

Feedback Use Cases (FB-UC)

FB-UC1

- Customer <<initiates>> the use case
- Feedback Database <<participates>> by providing and receiving data
- Administrative Portal <<observes>> by noting any significant data trends

A customer completes and submits the in-store survey. The response data, as well as the time of submission and table number, are recorded in the feedback database. Data from the database is retrieved to be compiled with the new submission, and the result is analyzed for trends of exceptionally poor or excellent ratings. If any such notable tendencies are found, they are displayed on the administrative portal.

FB-UC2

- Customer <<initiates>> the use case
- Feedback Database <<participates>> by providing and receiving feedback data
- Administrative Portal <<observes>> by receiving access to the sorted responses and word cloud

A customer submits an open-ended response relating to their experience at the restaurant. If possible, the submission is sorted into categories based on its content, indicating which part of the dining experience the feedback relates to. General information regarding the responses, such as a word cloud, is accessible through the administrative portal.

Table Reservation Use Case (TR-UC)

TR-UC1

- Customer <<initiates>> the use case and optionally receives a reservation confirmation message
- Seating Schedule Database <<participates>> by providing the current seating schedule and updating this data if necessary
- Host/Hostess Interface <<observes>> by noting any changes that occur in the seating schedule

A customer calls the restaurant to make a reservation, indicating party size and time of arrival. The current table schedule is accessed and compared with the request to see if an opening is available. If so, the table schedule is updated to include the new reservation and a confirmation is sent to the customer. If not, the customer is offered an alternative time. The customer may accept or provide a new time of arrival for their party, repeating the process until either a reservation is accepted or the customer does not provide another request.

TR-UC2

- Customer <<initiates>> the use case
- Seating Schedule Database <<participates>> by providing the current seating schedule and updating this data if necessary
- Host/Hostess Interface <<observes>> by noting any changes that occur in the seating schedule

A customer requests to be seated at the restaurant without a reservation, indicating party size. The table scheduling database is accessed and compared with the request to determine if a table is available. If so, the customer is seated and the table schedule is updated to reflect that the table is in use. Otherwise, the customer is allowed to wait for the next available table, placing them into a queue. The table schedule is updated to indicate that the next expected table vacancy will be reserved for the customers currently waiting in the queue.

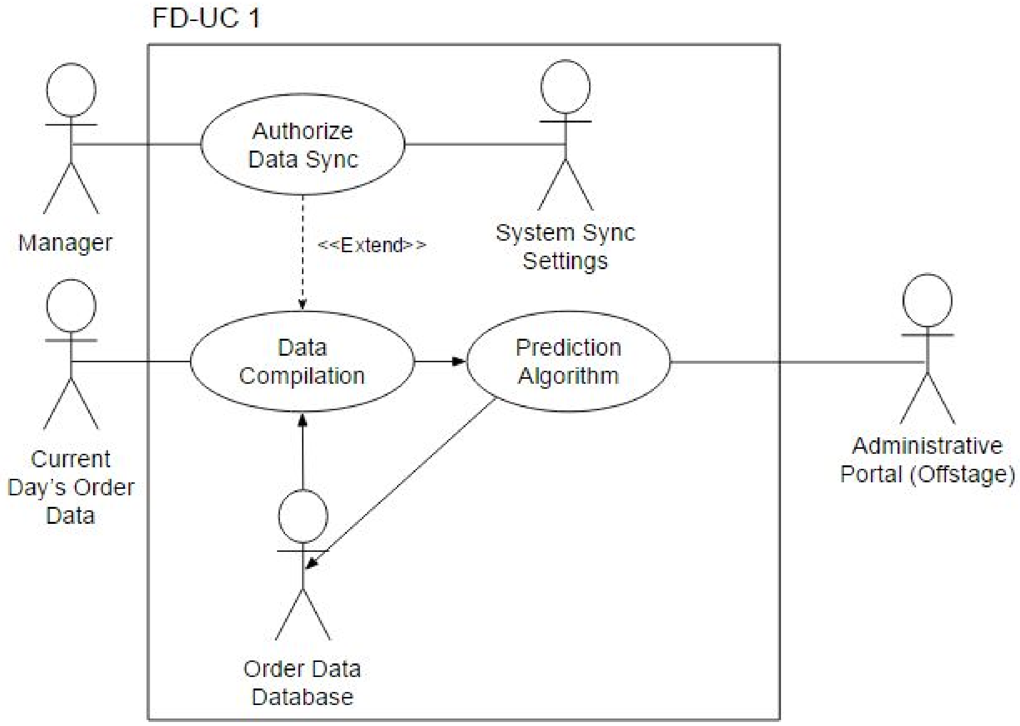
Manage Shifts Use Case (MS-UC)

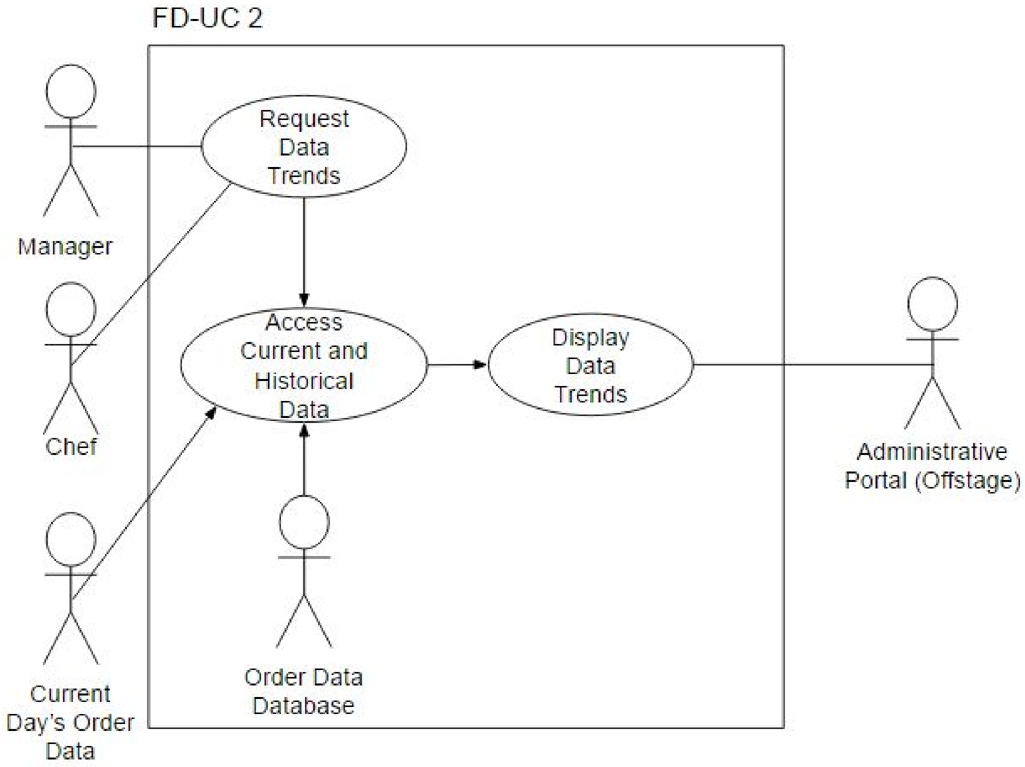
MS-UC1

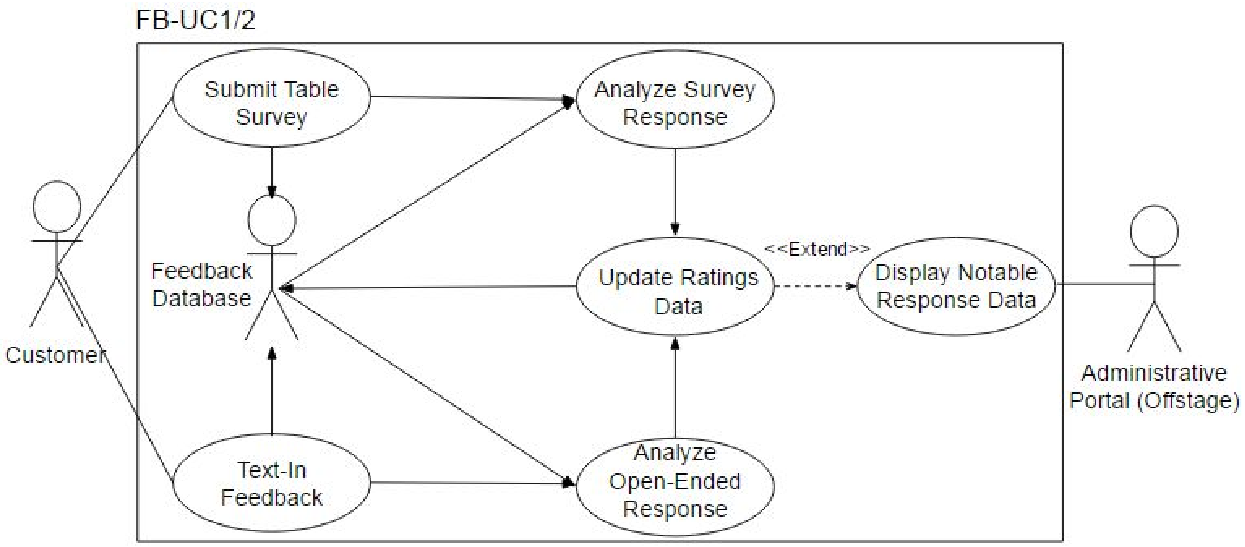
- General Employee <<initiates>> the use case by logging into the employee portal
- Employee Schedule Database <<participates>> by providing the current employee work shift schedule and updating this data if necessary
- Manager <<participates>> by accessing the updated employee schedule, creating the initial schedules, and making any necessary manual overrides.

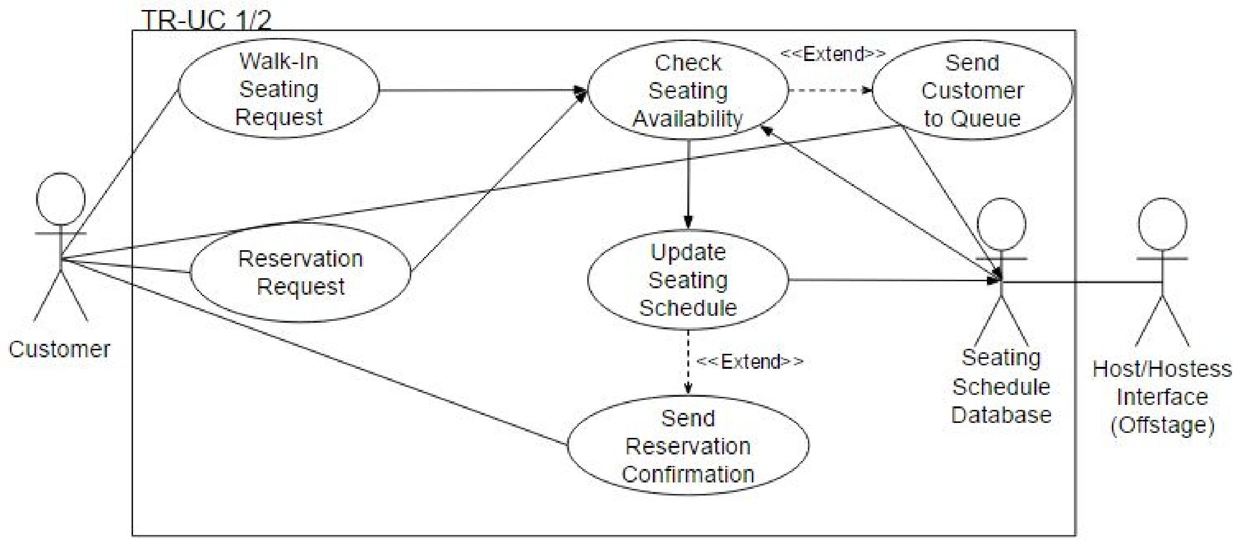
General employees can log in and access an employee portal where they may add, drop their work shifts. This in turn notifies the manager. The manager is who initially schedules these work shifts to the employees, and reserves the power to modify or change these shifts.

ii. Use Case Diagrams

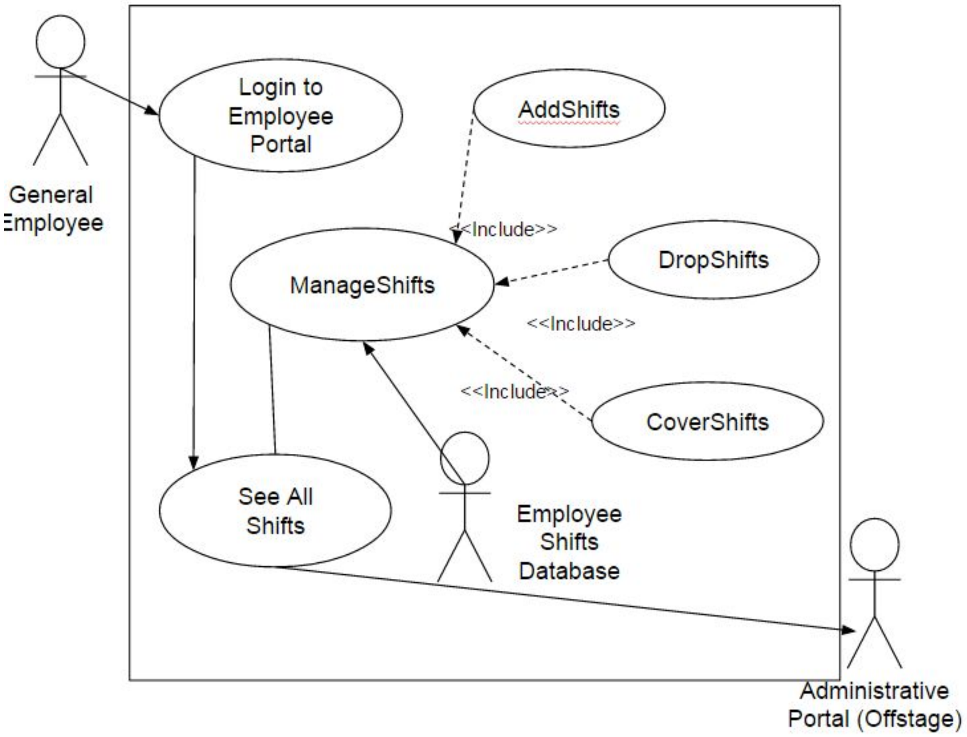





FB-UC1/2
Submit Table Survey
Analyze Survey Response
Customer
Feedback Database
Update Ratings Data
<<Extend>>
Display Notable Response Data
Administrative Portal (Offstage)
Text-In Feedback
Analyze Open-Ended Response


TR-UC 1/2
Walk-In Seating Request
Check Seating Availability
<<Extend>>
Send Customer to Queue
Customer
Reservation Request
Update Seating Schedule
Seating Schedule Database
Host/Hostess Interface (Offstage)
<<Extend>>
Send Reservation Confirmation

MS-UC 1


General
Employee
Login to Employee Portal
AddShifts
<<Include>>
ManageShifts
DropShifts
<<Include>>
<<Include>>
CoverShifts
See All Shifts
Employee Shifts Database
Administrative Portal (Offstage)

iii. Traceability Matrix

	ST Size	FD-UC1	FD-UC2	FB-UC1	FB-UC2	TR-UC1	TR-UC2	MS-UC1
ST-MA-1	7	1	1					
ST-MA-2	6		1					
ST-MA-3	5		1					
ST-MA-4	4	1						
ST-MA-5	8			1				
ST-MA-6	5				1			
ST-MA-7	7			1	1			
ST-MA-8	2							1
ST-MA-9	4							1
ST-CH-1	10	1						
ST-CH-2	5	1	1					
ST-CH-3	4	1	1					
ST-CH-4	5		1					
ST-CH-5	7			1				
ST-HO-1	5					1	1	
ST-HO-2	4					1	1	
ST-HO-3	4					1		
ST-HO-4	5					1	1	
ST-HO-5	2					1		
ST-HO-6	7					1	1	
ST-HO-7	5						1	
ST-CU-1	4					1		
ST-CU-2	3					1		
ST-CU-3	2					1		
ST-CU-4	4					1		
ST-CU-5	3					1	1	
ST-CU-6	8				1			
ST-CU-7	3			1				
ST-CU-8	6				1			
ST-GE-1	6							1
ST-GE-2	3							1
ST-GE-3	3							1

ST-GE-4	**4**							1
ST-GE-5	**3**							1
	201	30	32	25	26	43	29	25

*Each User Story was counted multiple times (in overall total) if it was resolved by multiple Use Cases

iv. Fully-Dressed Descriptions

Use Case FD-UC1: Data Synchronization

Related Reqs: ST-MA-1, ST-MA-4, ST-CH-1, ST-CH-2, ST-CH-3

Initiating Actor: Manager or System Synchronization Settings

Actor's Goal: Update the data being used for orders prediction to include new data.

Participating Actors: Order Database, Current Day's Order Data

Preconditions: Current day's order data has been collected and not imported into learning database table. Manager has been authenticated.

Postconditions: The system will make sure the current day's data is moved into the database. We ensure that we do not duplicate data.

Flow of Events for Main Success Scenario:

1. → Manager or System Settings <<initiates>> the request by either manually requesting the synchronization through the Manager's UI or the System's scheduler.
2. ← The system will respond by moving the data into the database table where the prediction algorithm learns from new data. All data should be successfully moved away from the current day's order data.

Flow of Events for Extensions:

1. → Actor requests synchronization when there is no data
    a. ← System responds letting the user know there is no data to copy
2. → Actor attempts to request synchronization without authentication/authorization
    a. ← Return error message letting the actor know that they must be authorized to perform the operation.
3. → The database server could not be found or connect.
    a. ← The system lets the user know that the database could not be connected. Notifies the user whether the database service can be started.
    b. → Actor waits a specified amount of time for system to start the database.
    c. ← Database has not started
    d. → Actor is given the option to cancel the synchronization operation and to call support.

Use Case FD-UC2: Orders Prediction

Related Reqs: ST-MA-1, ST-MA-2, ST-MA-3, ST-CH-2, ST-CH-3, ST-CH-4

Initiating Actor: Manager or Chef

Actor's Goal: Determine the number of ingredients that will be consumed in a given time period.

Participating Actors: Order Database, Current Day's Order Data

Preconditions: Order data from previous day is stored within the order database. Manager or chef has been authenticated

Postconditions: The system responds with a set of data based on its prediction algorithm. The system will store its prediction to determine how accurate it the prediction was for future use.

Flow of Events for Main Success Scenario:

1. → Chef or Manager <<initiates>> the request either through the use interface or the voice service by providing the day and time that the actor would like information on.
2. ← The system responds with data representing the ingredients needed for the orders in a given period of time

Flow of Events for Extensions:

1. → Actor requests data for a time in the past
    a. ← System responds with invalid time exception
2. → Actor requests prediction when there is no order database connected or the order database is empty
    a. ← System responds with a message notifying the user that it can't make predictions without data.

Use Case UC-FB-1: Table Survey Analysis

Related Reqs: ST-MA-5, ST-MA-7, ST-CH-5, ST-CU-7

Initiating Actor: Customer

Actor's Goal: Provide feedback relating to specific aspects of dining experience.

Participating Actors: Feedback Database, Administrative Portal

Preconditions: Time and table number that are submitted with user survey will allow the system to allocate ratings to the appropriate employees using the employee schedule.

Postconditions: Feedback is stored and reflected in employee ratings. If certain thresholds are crossed, the manager is notified.

Flow of Events for Main Success Scenario:

1. → Customer <<initiates>> survey analysis by submitting a survey after a meal.
2. ← System requests feedback data from Feedback Database.
3. ← Feedback data from the Feedback Database is accessed for use in analysis.
4. ← Old and new data are analyzed by the program.
5. ← The Feedback Database is updated to reflect the new rating values.

Flow of Events for Extensions:

1. ← If a significant event occurs regarding the new rating totals after analysis, the Administrative Portal displays a notification.

Use Case TR-UC1: Table Reservation

Related Reqs: ST-HO-1, ST-HO-2, ST-HO-3, ST-HO-4, ST-HO-5, ST-HO-6, ST-CU-1, ST-CU-2, ST-CU-3, ST-CU-4, ST-CU-5

Initiating Actor: Customer

Actor's Goal: Make an online reservation including time and date of arrival and party size.

Participating Actors: Host/Hostess Interface, Seating Schedule Database

Preconditions: Customer is on the website. The restaurant is open.

Postconditions: Customer's reservation is set in the database and able to be seen by the host

Flow of Events for Main Success Scenario:

→ 1. Customer accesses ChefBoyRD website, and clicks reservation tab.

← 2. System responds by requesting reservation data from main server.

→ 3. Customer populates information about their reservation. Date/time, phone number.

← 4. System checks the reservation information and confirms the slot is empty.

← 5. System updates reservation information to the database.

← 6. System sends the user a reservation confirmation.

Flow of Events for Extensions:

→ 1. Customer requests a reservation that is already filled

← a. The system responds by checking if the reservation is available.

← b. The system notifies customer that it is not a valid reservation.

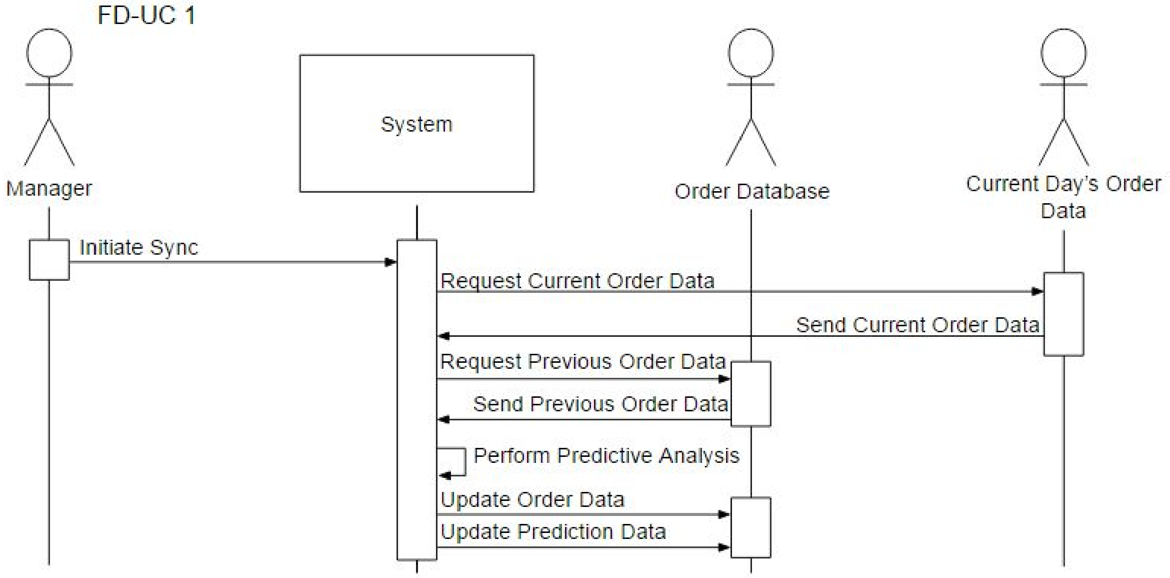
→ 2. Customer wishes to cancel a reservation

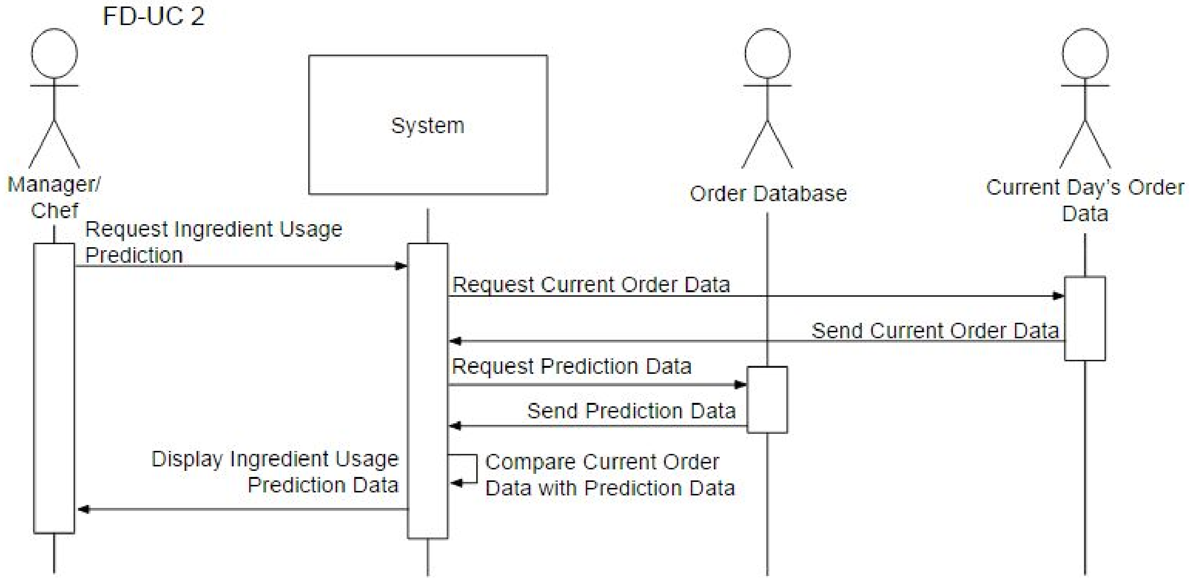
← a. The system responds by checking to see if the reservation matches the customer who is requesting it. (By phone number)

← b. Upon confirmation, the system makes a change to the reservations list and updates the database.

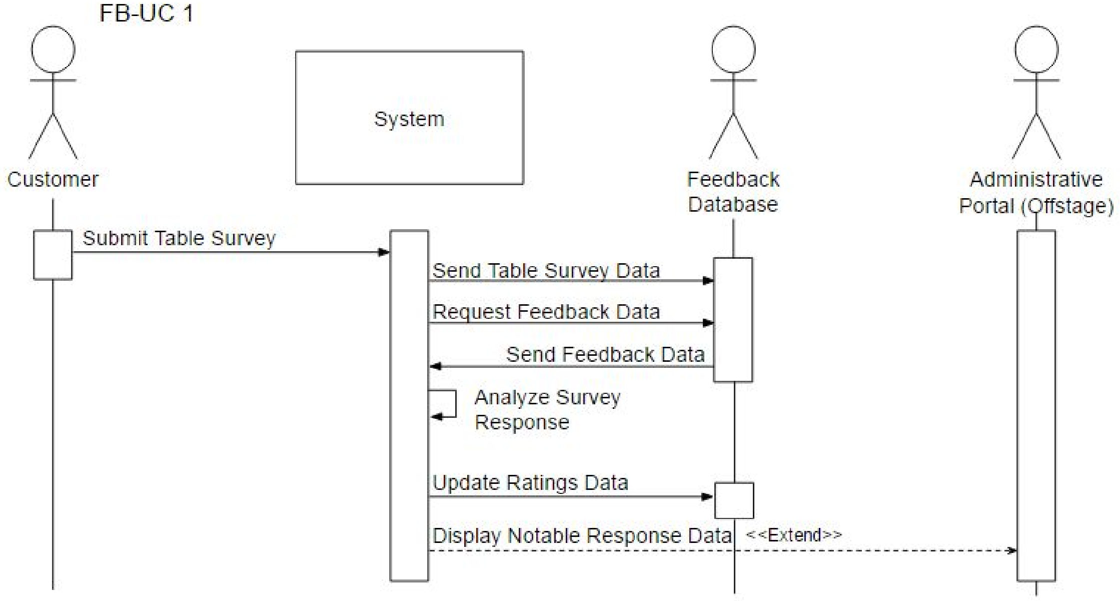
← c. The system sends a confirmation to the user that reservation has been cancelled.

d. System Sequence Diagrams

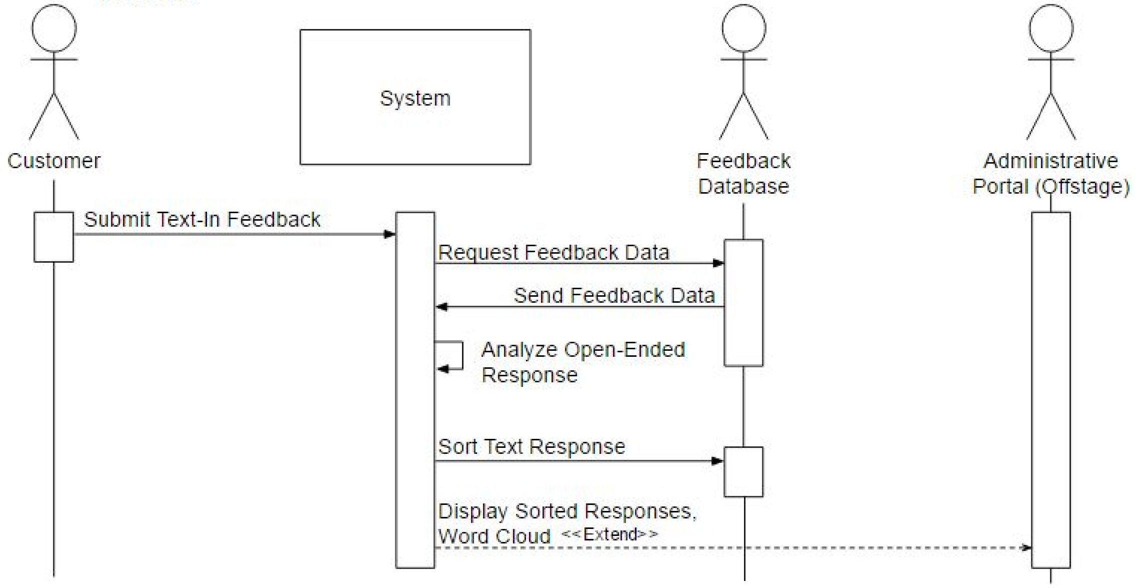


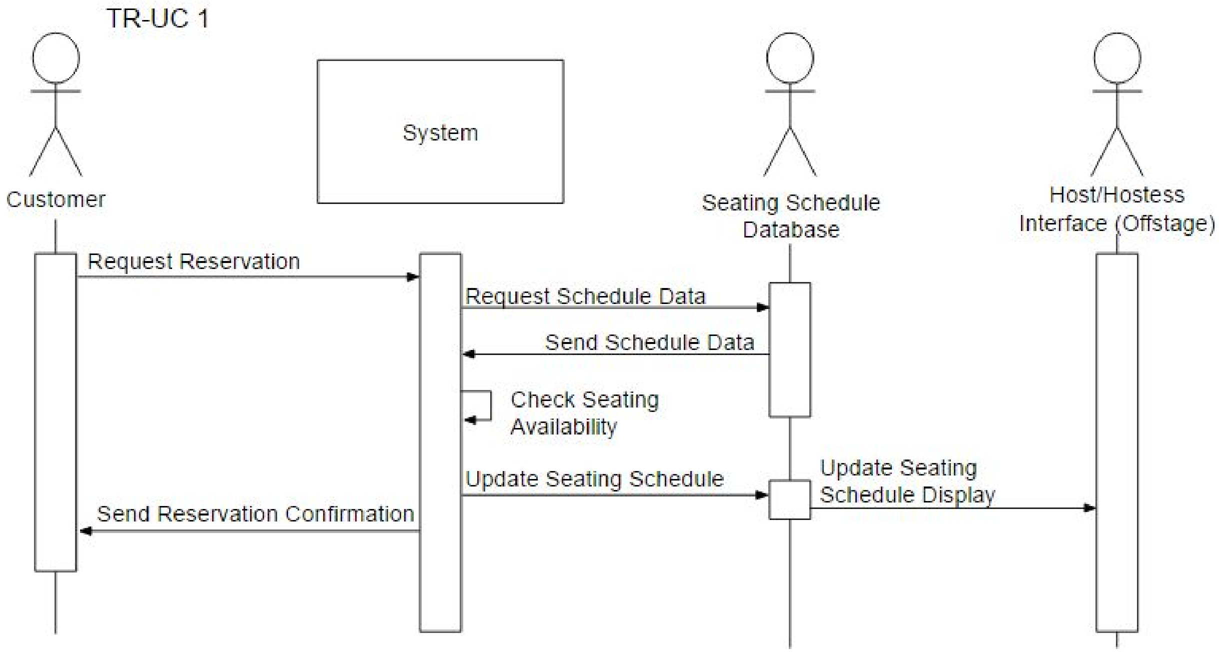

FD-UC 2
Manager/ Chef
System
Order Database
Current Day's Order Data
Request Ingredient Usage Prediction
Request Current Order Data
Send Current Order Data
Request Prediction Data
Send Prediction Data
Compare Current Order Data with Prediction Data
Display Ingredient Usage Prediction Data

FB-UC 1


Customer
System
Feedback Database
Administrative Portal (Offstage)
Submit Table Survey
Send Table Survey Data
Request Feedback Data
Send Feedback Data
Analyze Survey Response
Update Ratings Data
Display Notable Response Data
<<Extend>>

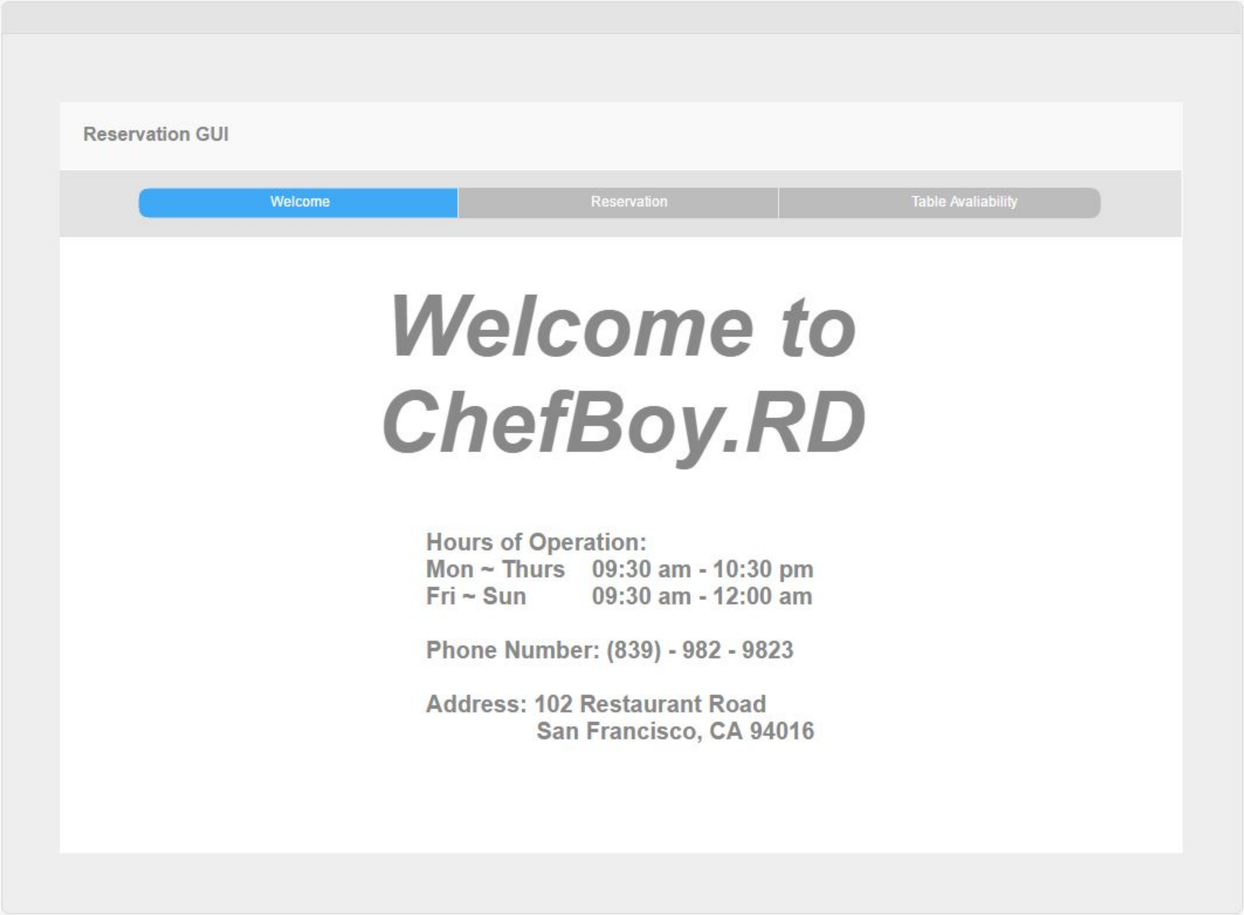
FB-UC 2


Customer
System
Feedback Database
Administrative Portal (Offstage)
Submit Text-In Feedback
Request Feedback Data
Send Feedback Data
Analyze Open-Ended Response
Sort Text Response
Display Sorted Responses, Word Cloud
<<Extend>>


TR-UC 1
Customer
System
Seating Schedule Database
Host/Hostess Interface (Offstage)
Request Reservation
Request Schedule Data
Send Schedule Data
Check Seating Availability
Update Seating Schedule
Update Seating Schedule Display
Send Reservation Confirmation

4. User Interface Specification

a. Preliminary Design



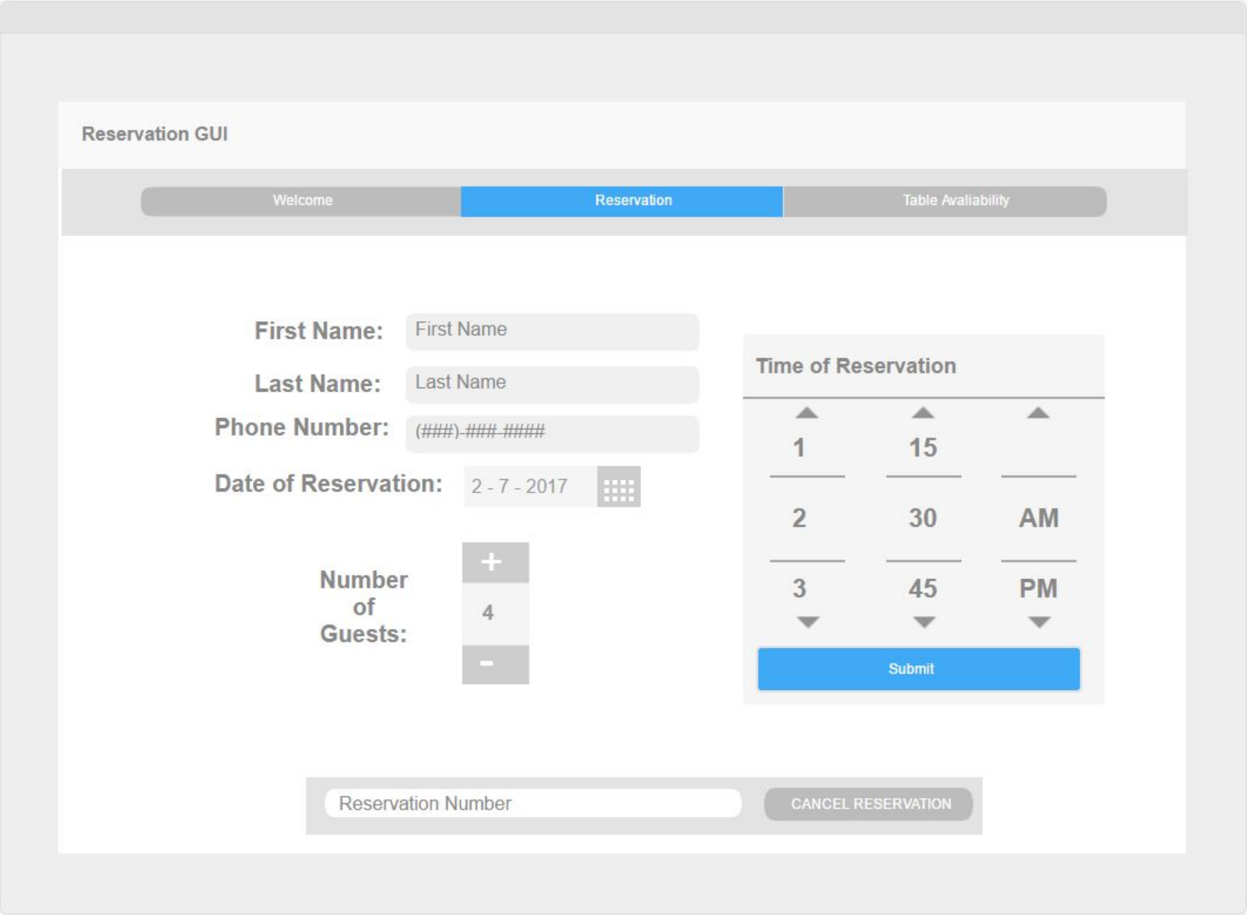
Welcome to ChefBoy.RD

Hours of Operation:
Mon ~ Thurs 09:30 am - 10:30 pm
Fri ~ Sun 09:30 am - 12:00 am

Phone Number: (839) - 982 - 9823

Address: 102 Restaurant Road
San Francisco, CA 94016

Figure 1: Customer welcome screen to give basic information about the restaurant such as address, phone number, opening and closing hours, reservation times, and table availability.



First Name: First Name

Last Name: Last Name

Phone Number: (###)-###-####

Date of Reservation: 2 - 7 - 2017

Number of Guests: 4

Reservation Number

Figure 2: Customers make reservations by typing their first and last name, phone number, the number of guests to be seated, the date of reservation and the time of reservation. Customers can also cancel reservations if they have their reservation number

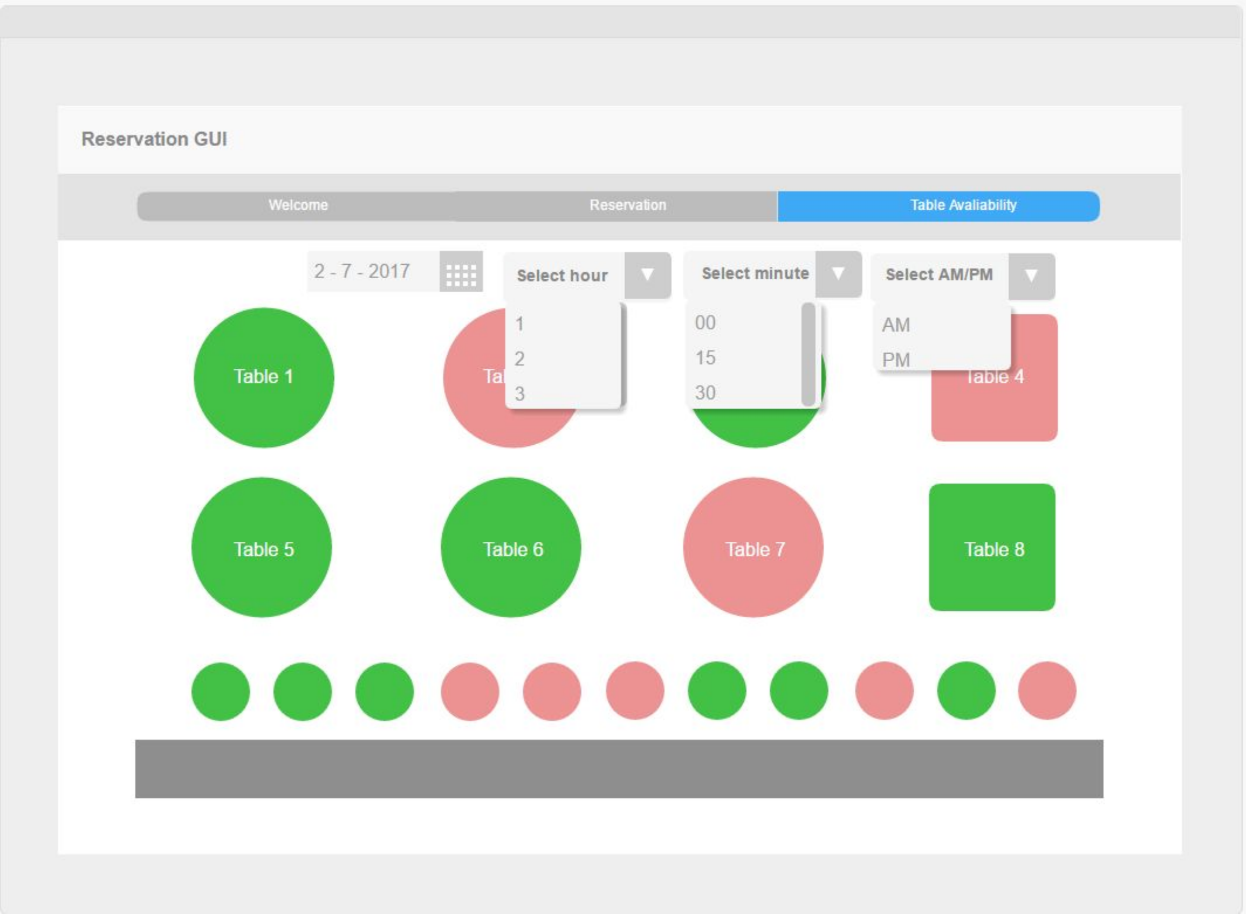


Figure 3: Customers can see the tables that are available depending the date. Red tables signify that the table is reserved or occupied and green tables signify that they are available. Smaller seating options are indicated along the bar and tables 4 and 5 are normally used to accommodate larger groups of guests (seating will vary from restaurant to restaurant).

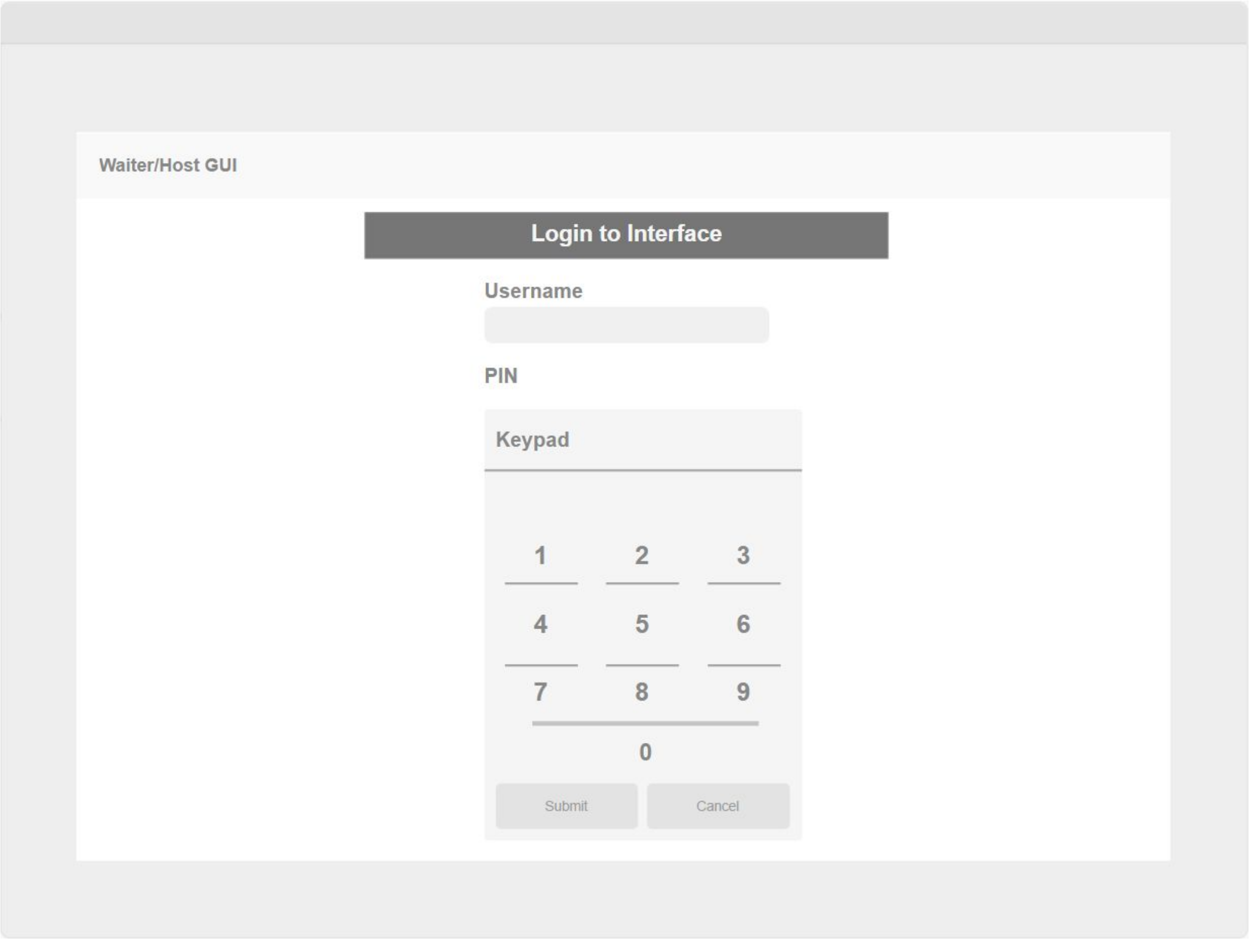


Figure 4: Hosts and waiters log-in into the interface using a username and a pin for easy access. This will prevent any overlapping information between waiters and hosts. Having unique operators for an interface will prevent miscommunication.

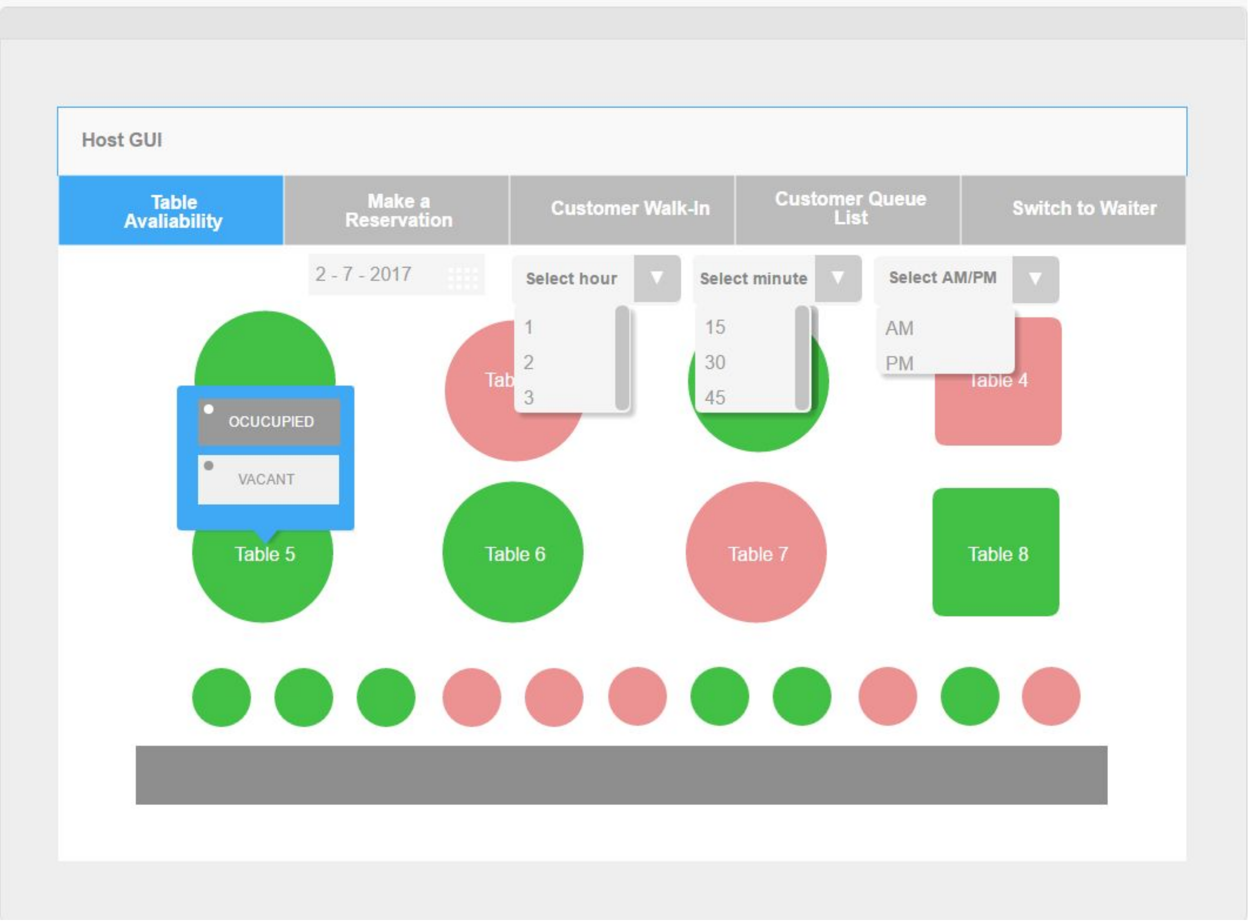
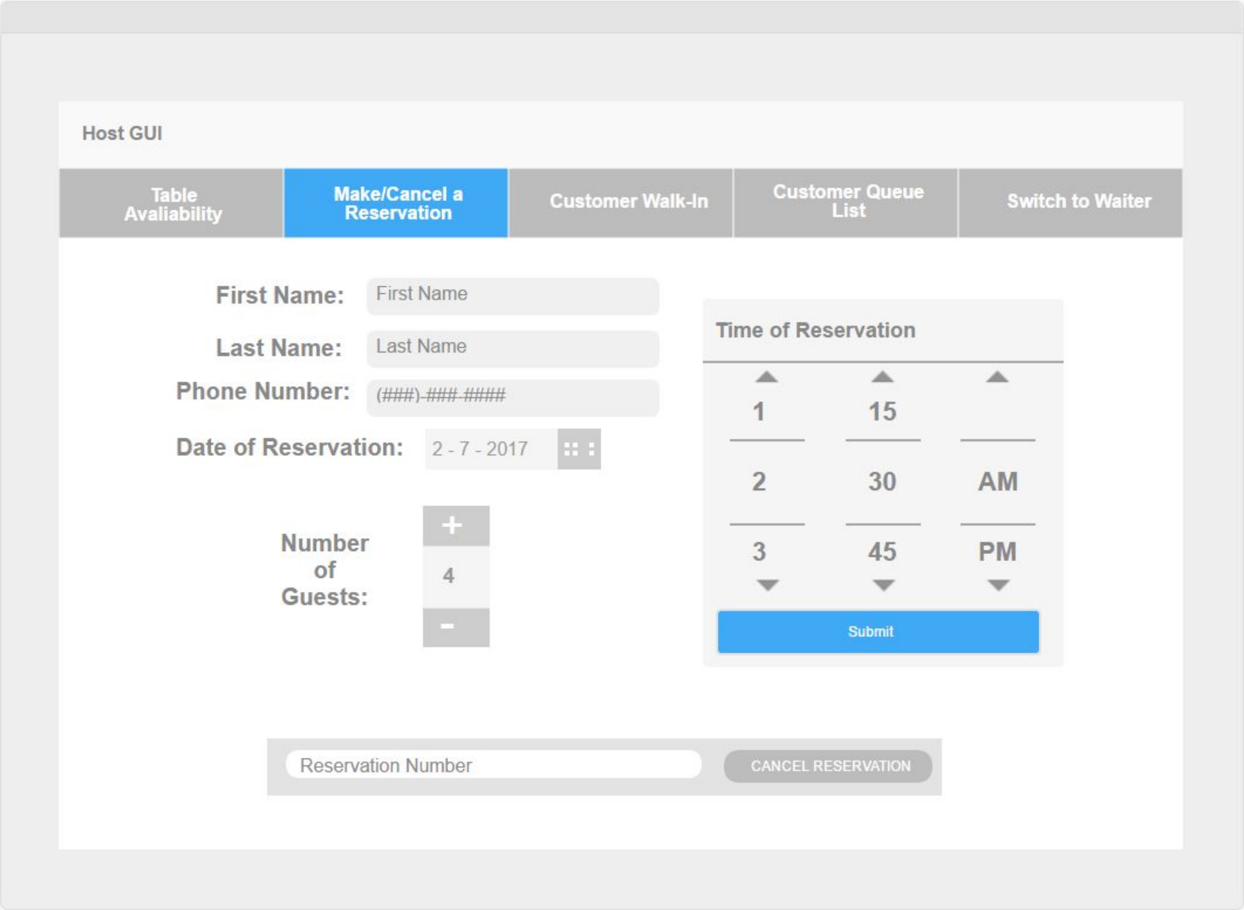


Figure 5: Host will have the visual representation of the restaurant table availability just to allow them to have a general idea of how populated their restaurant is. From the navigation bar, the host also has the ability to make a reservation, manage customer walk-ins, and see the customer queue list. The host is also able to put the table into occupied or vacant mode depending on if the customer wants to sit at a different table as shown in the blue pop-up bubble.



First Name: First Name

Last Name: Last Name

Phone Number: (###)-###-####

Date of Reservation: 2 - 7 - 2017

Number of Guests: 4

Time of Reservation

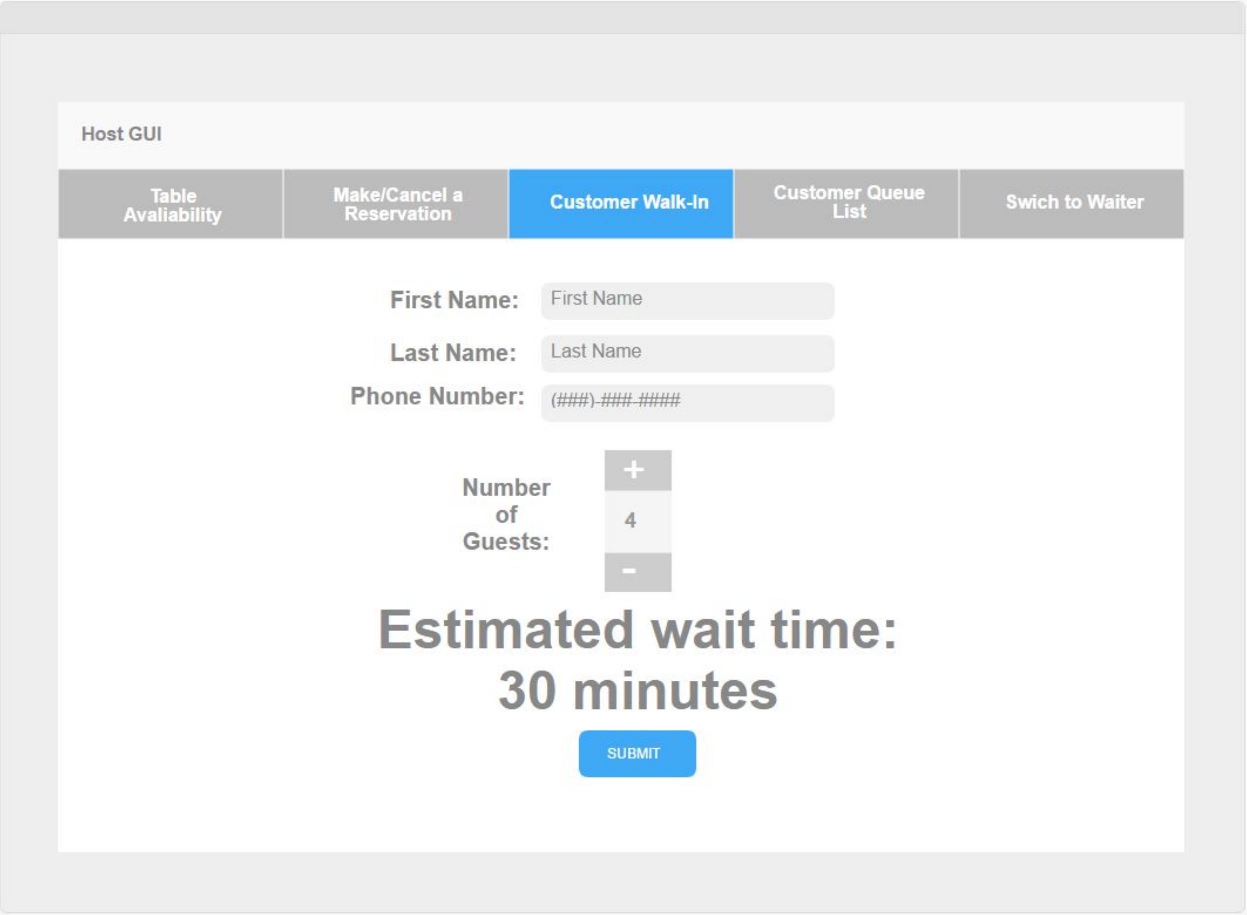
1 15

2 30 AM

3 45 PM

Reservation Number

Figure 6: Host has access to making a reservation in order to accommodate for the people that call the restaurants in order to make a reservation instead of going online and making a reservation there. The host will ask the customer for their first and last name, their phone number, the number of guests, and the time of reservation. If the customer wants to cancel the reservation, the host can either input the reservation number and cancel the reservation via this method or a method to be discussed in the customer queue list.



First Name: First Name

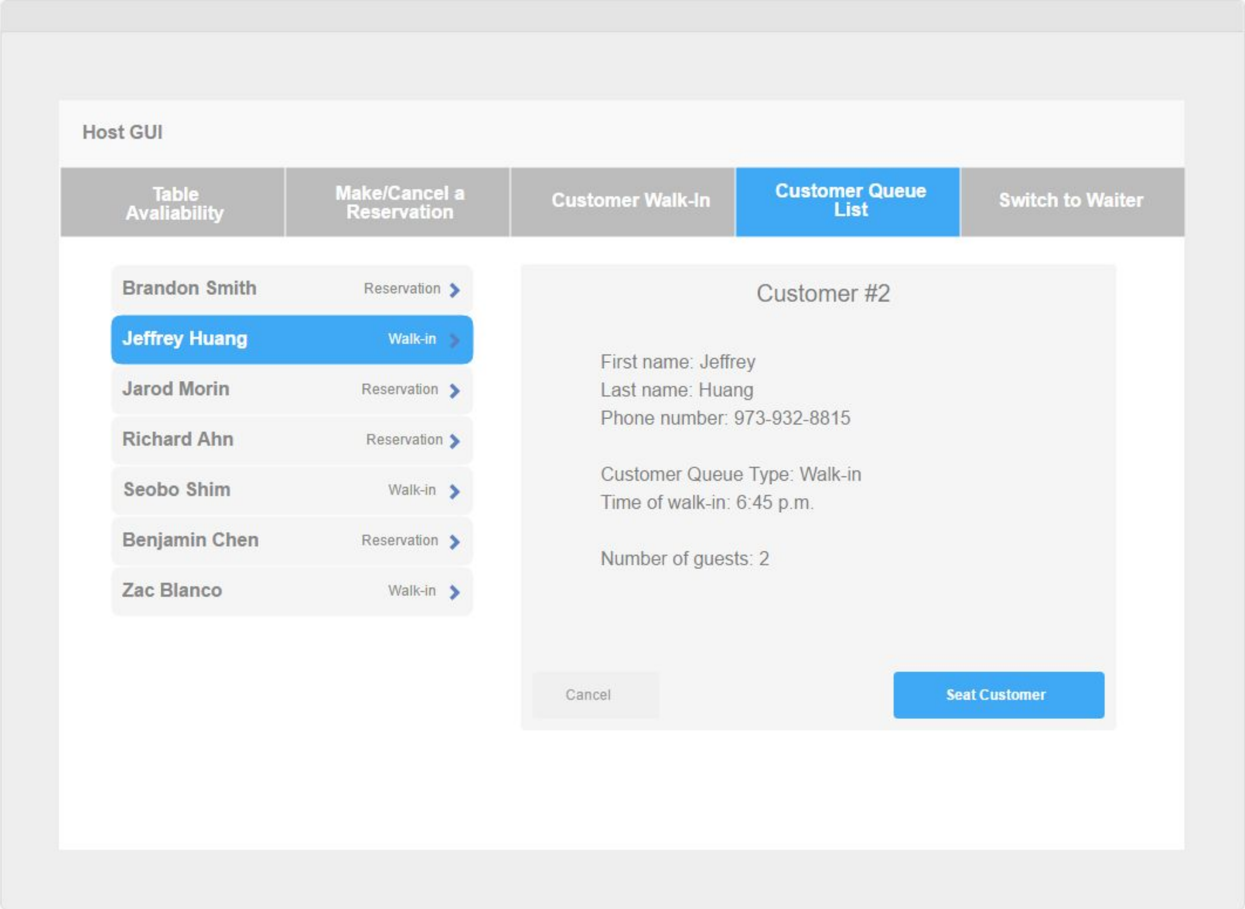
Last Name: Last Name

Phone Number: (###)-###-####

Number of Guests: 4

Estimated wait time:
30 minutes

Figure 7: Customers that walk into the restaurant that haven’t reserved a table should be seated accordingly by the host. If there are no tables available shown on the table availability, then the customer should be put into the customer queue list using the form shown above. If the customer needs to be placed into the queue list, there should be an estimated amount of time that the customer needs to wait in order to take a seat at a table.



Brandon Smith	Reservation
Jeffrey Huang	Walk-in
Jarod Morin	Reservation
Richard Ahn	Reservation
Seobo Shim	Walk-in
Benjamin Chen	Reservation
Zac Blanco	Walk-in

Customer #2

First name: Jeffrey
Last name: Huang
Phone number: 973-932-8815

Customer Queue Type: Walk-in
Time of walk-in: 6:45 p.m.

Number of guests: 2

Figure 8: Host has access to the list of people that are currently in the customer queue list in order to keep track of the reservation times. The program will determine the order of which the customers can be seated. Upon selecting a customer in the list, the customer's reservation/walk-in information will be displayed for easier understanding of seating options. The final option "Switch to Waiter" allows the host to switch between the waiter and the host role depending on the situation.

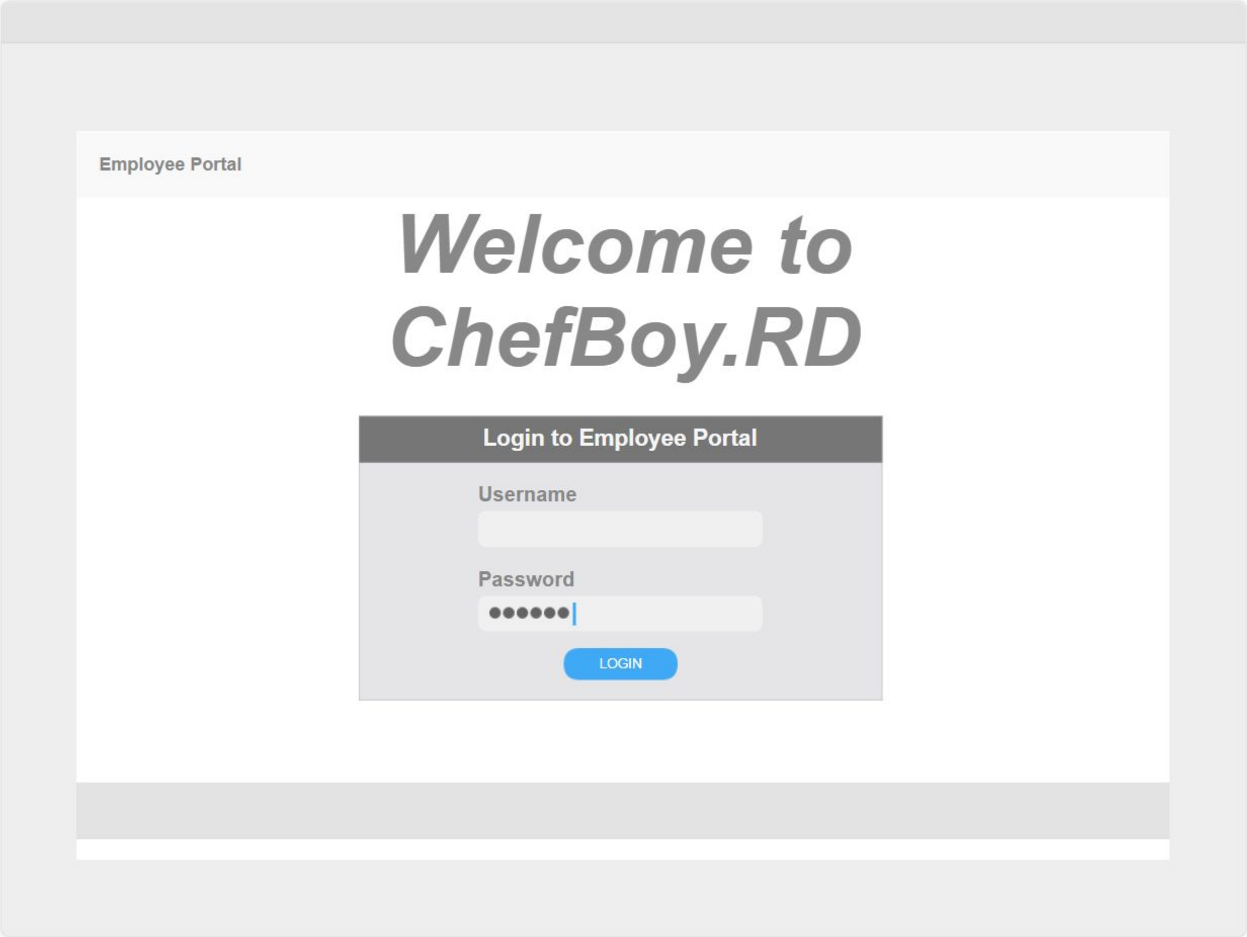
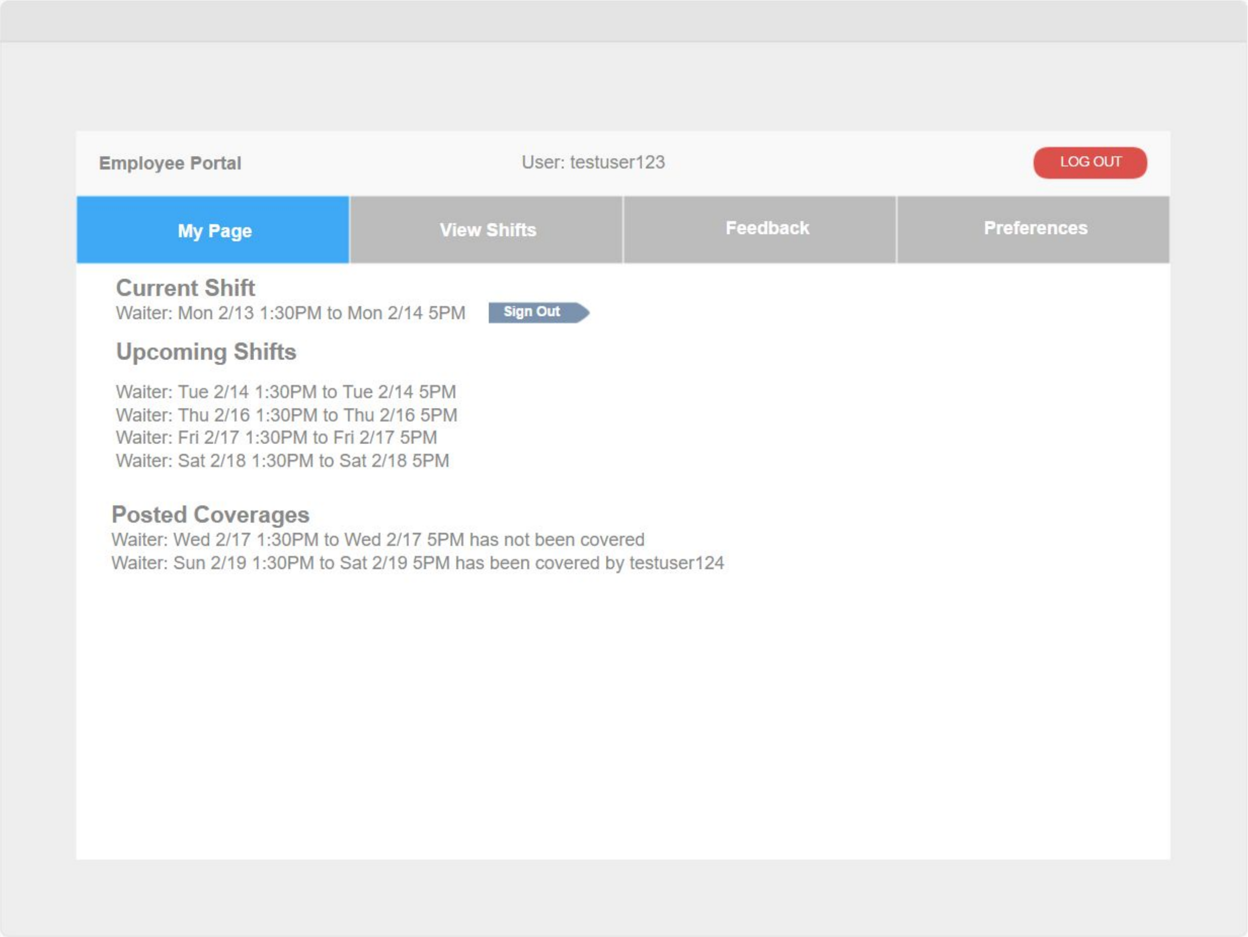


Figure 9: Employees are able to login via a web portal. When hired, an employee will be assigned specific roles which correlate to which menus the employee is able to access after successfully signing in. Username and password will be determined by the manager in terms of how secure they want their login information to be.



Current Shift

Waiter: Mon 2/13 1:30PM to Mon 2/14 5PM

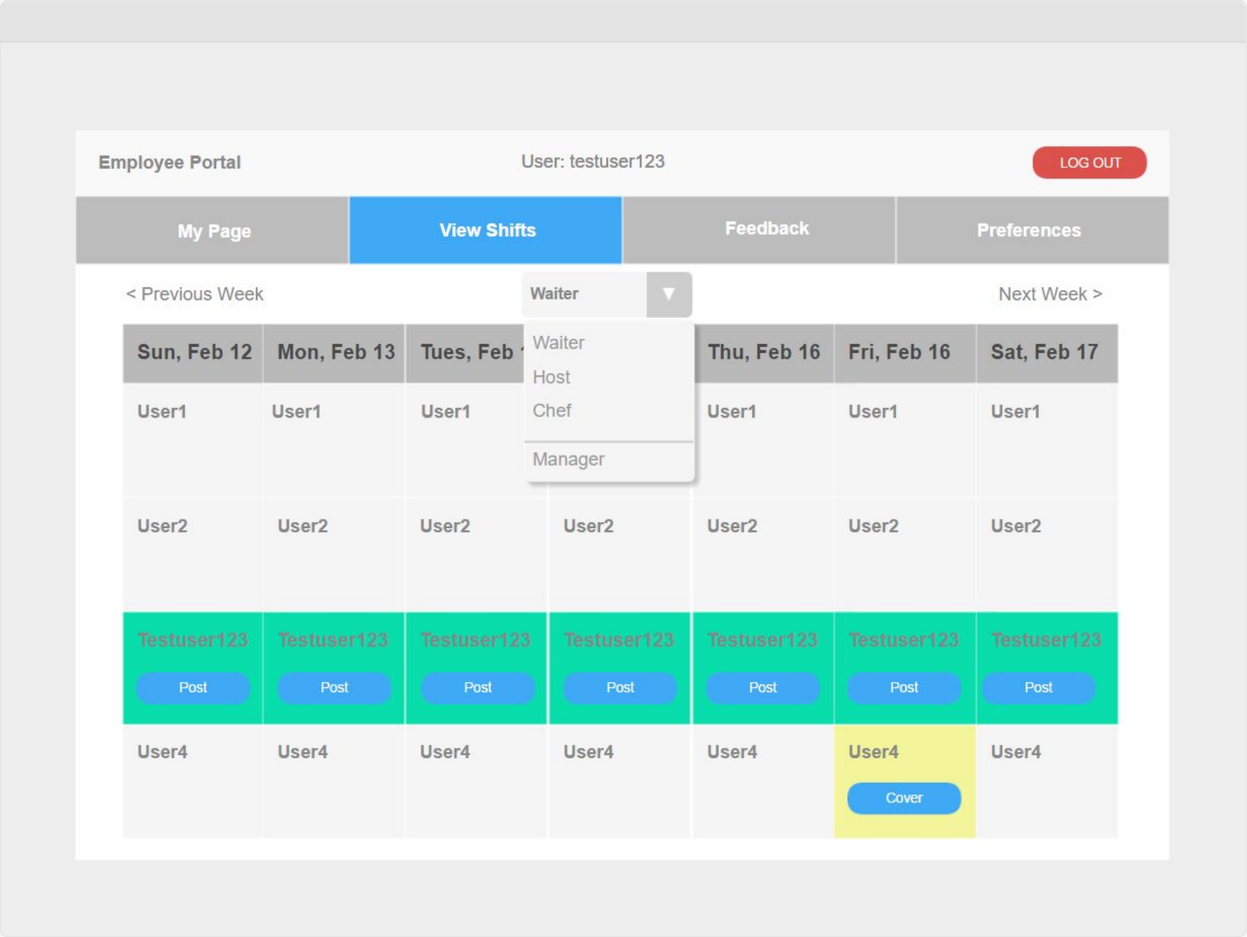
Upcoming Shifts

Waiter: Tue 2/14 1:30PM to Tue 2/14 5PM
Waiter: Thu 2/16 1:30PM to Thu 2/16 5PM
Waiter: Fri 2/17 1:30PM to Fri 2/17 5PM
Waiter: Sat 2/18 1:30PM to Sat 2/18 5PM

Posted Coverages

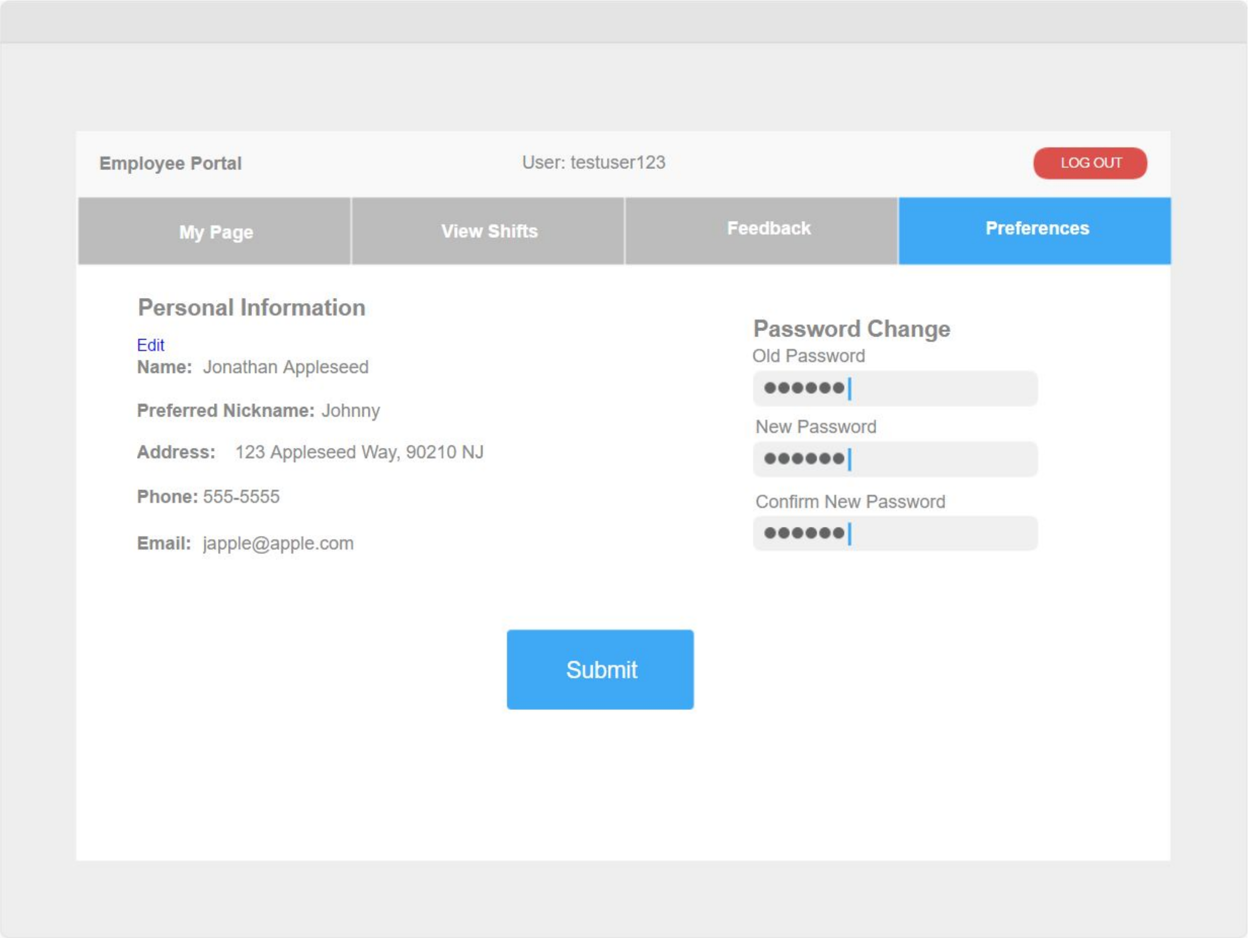
Waiter: Wed 2/17 1:30PM to Wed 2/17 5PM has not been covered
Waiter: Sun 2/19 1:30PM to Sat 2/19 5PM has been covered by testuser124

Figure 10: Within the employee portal pages will be available to see which shifts they have signed up for, which shift they may currently be on (if any) as well as requests for others to cover their shift.



Sun, Feb 12	Mon, Feb 13	Tues, Feb 1		Thu, Feb 16	Fri, Feb 16	Sat, Feb 17
User1	User1	User1		User1	User1	User1
User2	User2	User2	User2	User2	User2	User2
Testuser123 Post	Testuser123 Post	Testuser123 Post	Testuser123 Post	Testuser123 Post	Testuser123 Post	Testuser123 Post
User4	User4	User4	User4	User4	User4 Cover	User4

Figure 11: Employees can view the current shifts for their role and who is scheduled for each one. If any shifts are open or need more assistance then they will be able to claim the shifts and the manager(s) will be notified.



Employee Portal

User: testuser123

Personal Information

Name: Jonathan Appleseed

Preferred Nickname: Johnny

Address: 123 Appleseed Way, 90210 NJ

Phone: 555-5555

Email: japple@apple.com

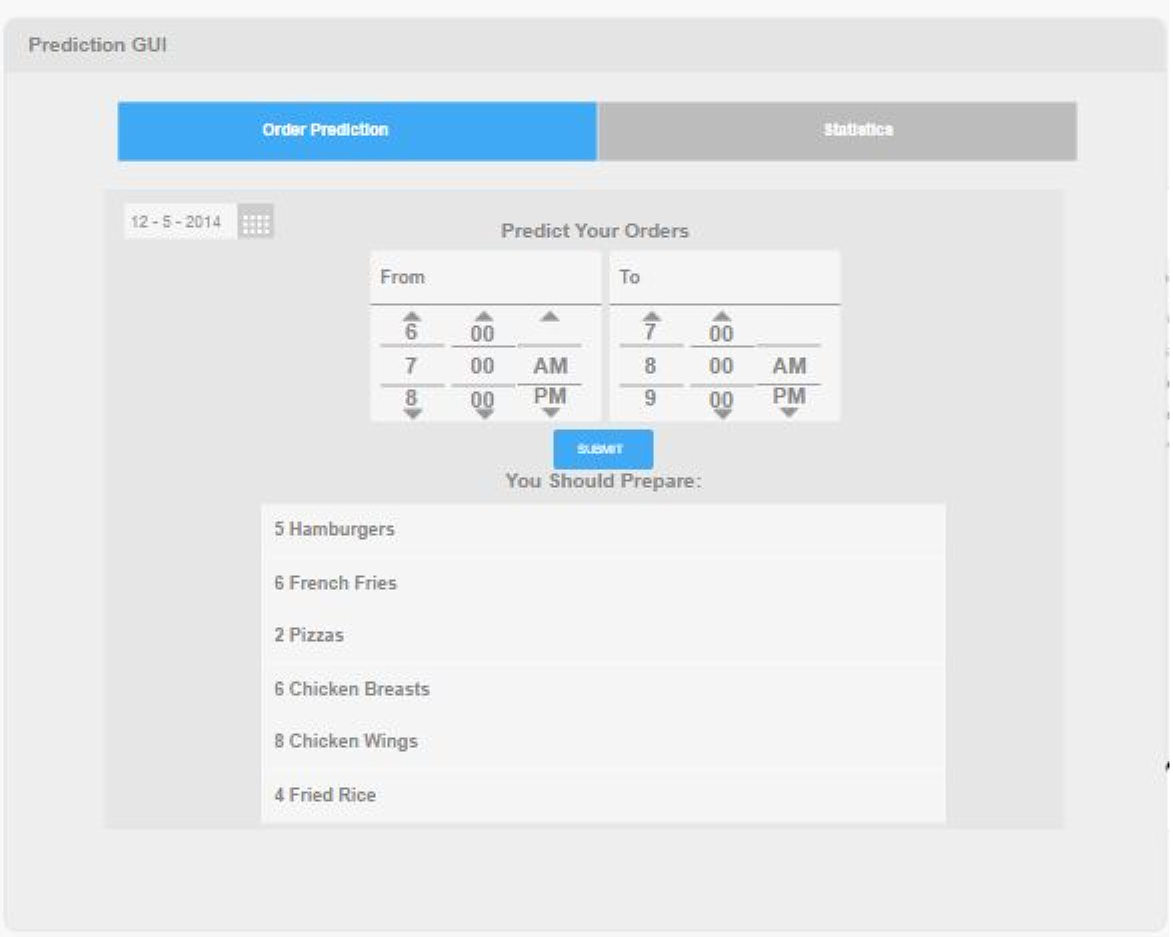
Password Change

Old Password: ●●●●●●

New Password: ●●●●●●

Confirm New Password: ●●●●●●

Figure 12: Employees will be able to manage their account. They will be able to set their own password and view the information that the business has on file about them. Any information that is going to be changed will be edited by the manager in person.



Predict Your Orders

You Should Prepare:

5 Hamburgers

6 French Fries

2 Pizzas

6 Chicken Breasts

8 Chicken Wings

4 Fried Rice

Figure 13: The chef and manager are able to access the prediction interface which will attempt to give a best-guess estimate of the food which will be ordered within a given time period. This gives the chefs a good idea of how much food they need to cook as well as the amount of inventory that the manager may need to order for a given day.

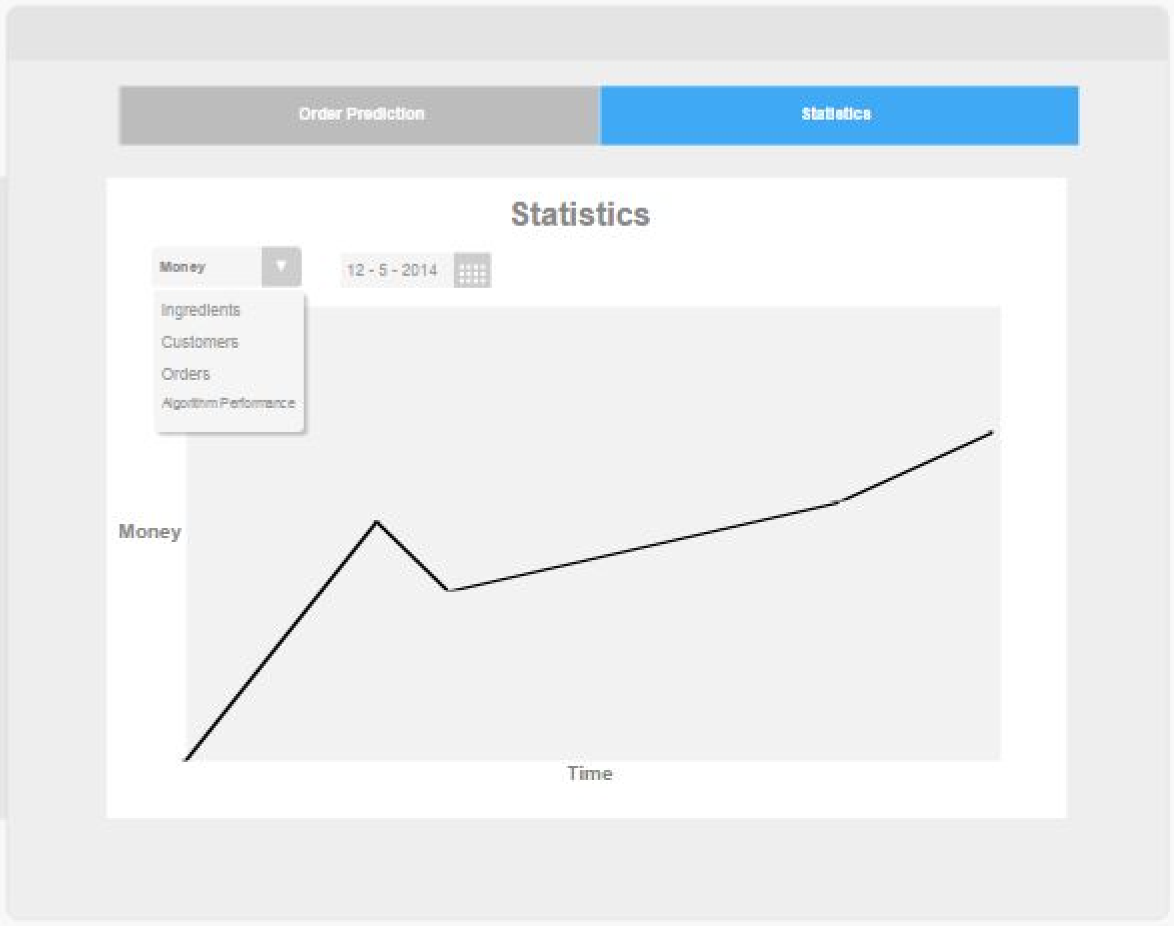


Figure 14: Managers can also get a look at the business statistics which are reported based on different metrics that the restaurant might see. This may include, but is not limited to: The amount of types of different ingredients used over time, the number of orders over time on a given day, and the performance of the prediction algorithms over time.

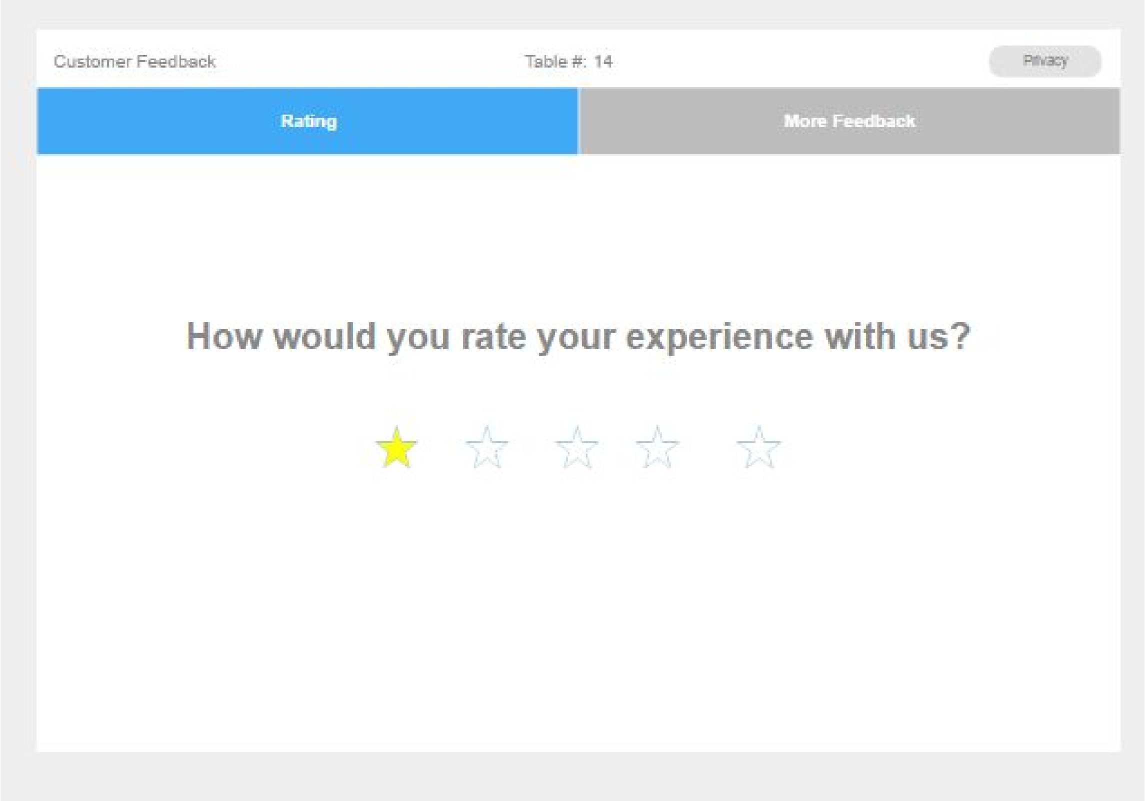
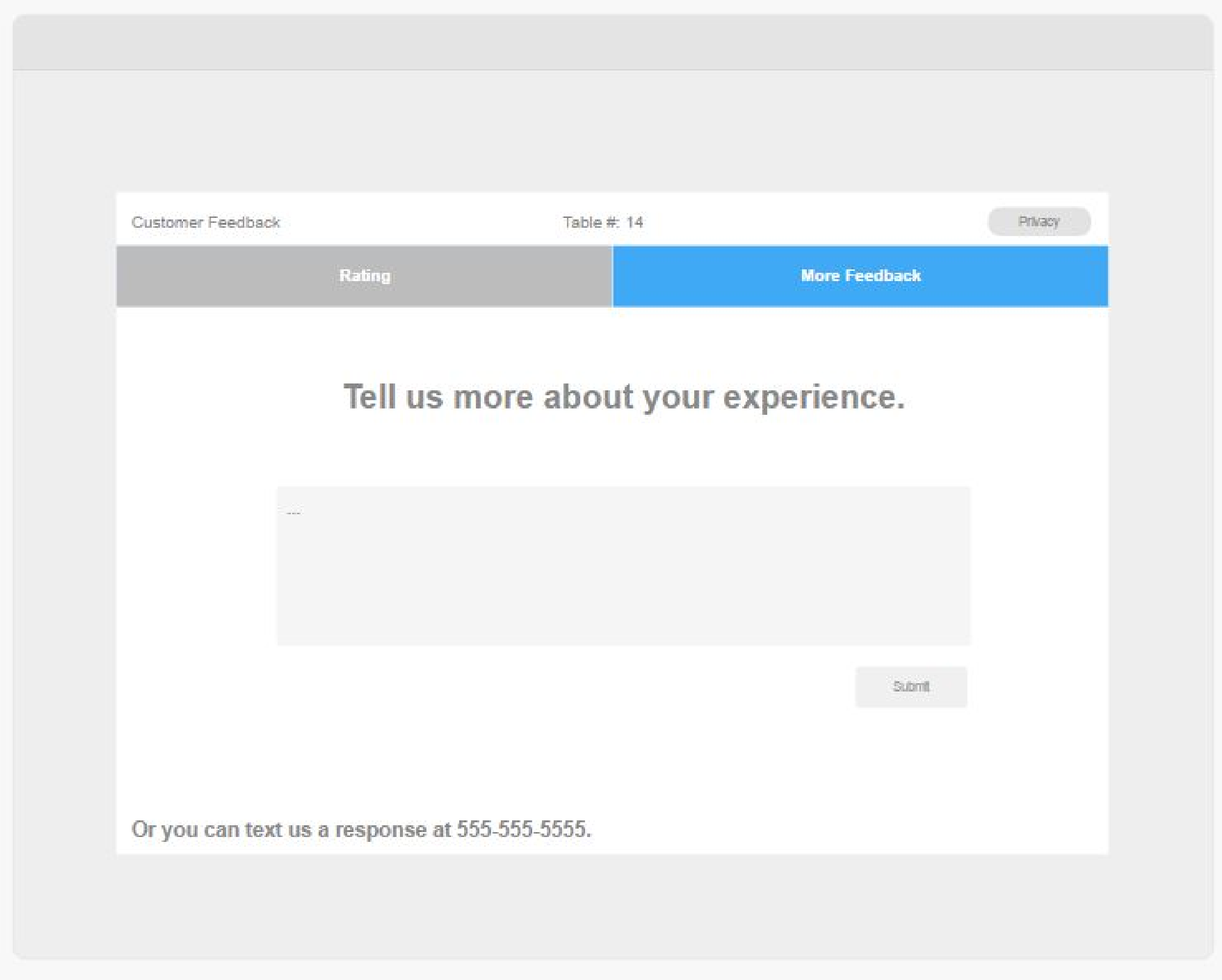


Figure 15: After receiving their order and finishing customers will be presented with a simple screen where they can rate the service they received at the restaurant. This can give managers a general idea of the service which is being provided at given times. This can also be applied to each waiter to see which waiters give the best or worst service to customers.



Customer Feedback

Table #: 14

Tell us more about your experience.

Or you can text us a response at 555-555-5555.

Figure 16: Customer feedback is one of the most important things in a restaurant in order to improve their customer service. They can type their response into the textbox that is provided and then submit their response.

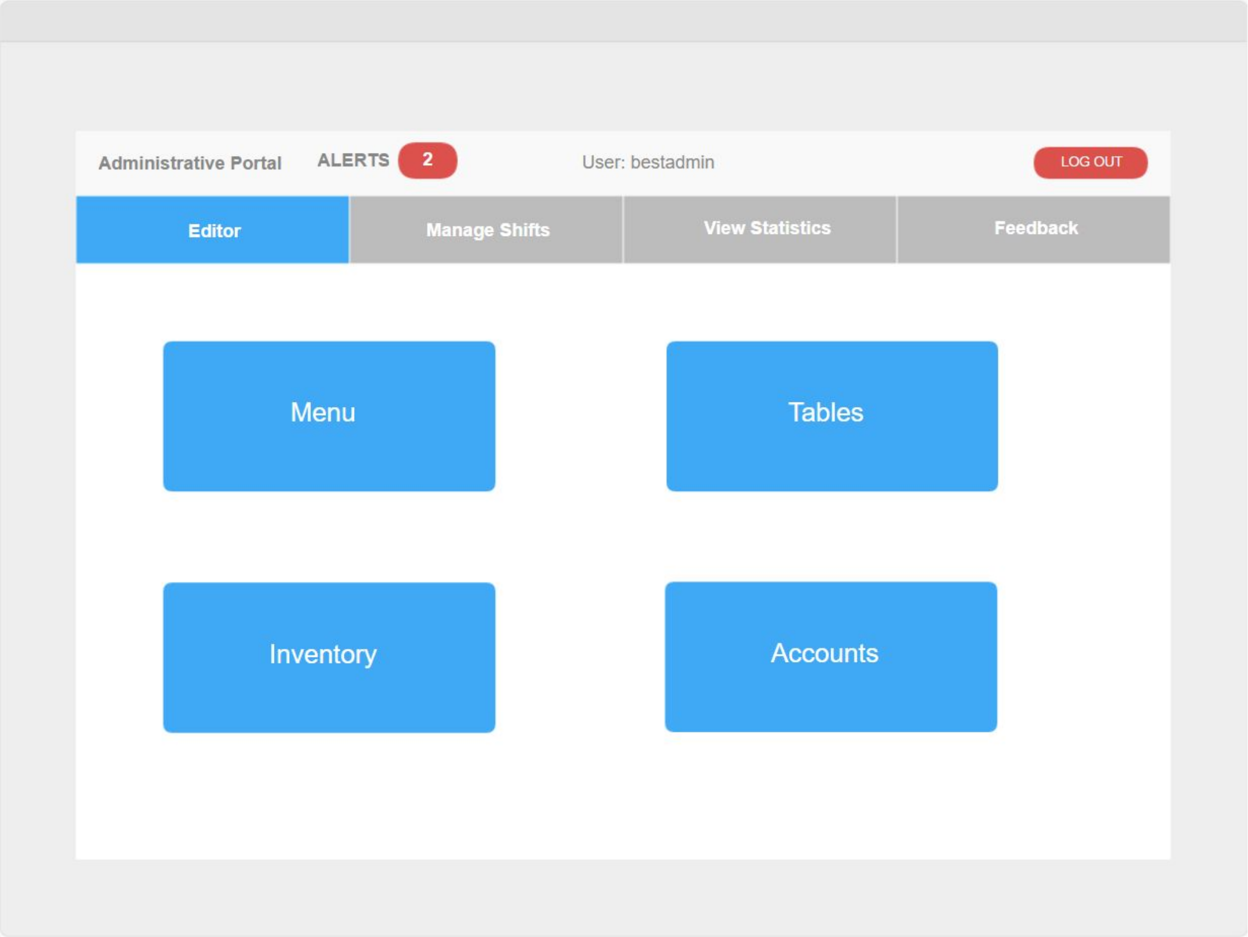


Figure 17: The administrative portal allows managers to edit the menus, the tables, inventory and accounts when they are required to do so. They also can manage shifts when an employee calls in sick or find openings to see if they need to hire more people.

b. User Effort Estimation

Customer/Host reserving a table: Total 2 clicks/presses 10-50 keystrokes

1. Select reservation tab
2. Typing their first name, last name, phone number, selecting date, number of guests, time of reservation.
3. Finalizing reservation by hitting submit

Customer/Host canceling a reservation: Total 3 clicks/presses

1. Select reservation tab
2. Typing their reservation number
3. Hitting "Cancel Reservation"

Customer checking Table Availability: Total 8 - 10 clicks/presses

1. Select Table Availability tab
2. Selecting date, hour, minute, a.m./p.m.

Customer leaving a Rating: Total 2 clicks/presses

1. Select correct tab (Rating)
2. Press appropriate star level

Customer leaving a feedback: Total 3 clicks/presses. 10- 200 keystrokes

1. Select correct table (More Feedback)
2. Selecting the feedback box and typing their response
3. Hitting submit to finalize the submission

Chef view order predictions: Total 2 clicks/presses

1. Select Predictions section from Chef Dashboard
2. Select correct tab (view predictions)

Manager view statistics: Total 2-3 clicks/presses

1. Select Statistics section from Chef Dashboard
2. Select correct tab
3. Select desired statistic from the dropdown

*Number of keystrokes can vary widely, depending on the type of information being entered.
Lower bound for number of clicks/presses: 22
Upper bound for number of clicks/presses: 25
Lower bound for number of keystrokes: 20
Upper bound for number of keystrokes: 250
On average: clicks/presses account for 14.8% of input, while keystrokes account for 85.2%

5. Domain Model

The domain model is a model designed to help someone understand how the different parts of the system works. It uses requirements analysis to show user interaction with the system (external behavior) as well as how the elements of the system interact with each other (internal behavior). The domain model is constructed based on several sources. For this project, the model is mostly based off of the use cases. Additional sources include: studying the work/problem domain, knowledge base of software designs, and our own past experiences with software design.

a. Concept Definitions

Responsibility Description	Type	Concept Name
RS1. Receives and holds order data	D	Database
RS2. Send order data	D	OrderData
RS3. Analyze data using prediction algorithm	D	AnalyzeData
RS4. Store data and algorithm results in database	D	Database
RS5. Display order data for user	D	Database
RS6. Store feedback	D	FeedbackData
RS7. Provide survey for customer	D	Survey
RS8. Receive survey and store response in feedback database	D	FeedbackData
RS9. Sort and analyze feedback database for trends	D	AnalyzeFeedback
RS10. Provide real-time seating information	D	Seating
RS11. Receive reservation from customer and check against seating	D	Reservation
RS12. Provide reservation confirmation message	D	ConfirmMessage

b. Association Definitions

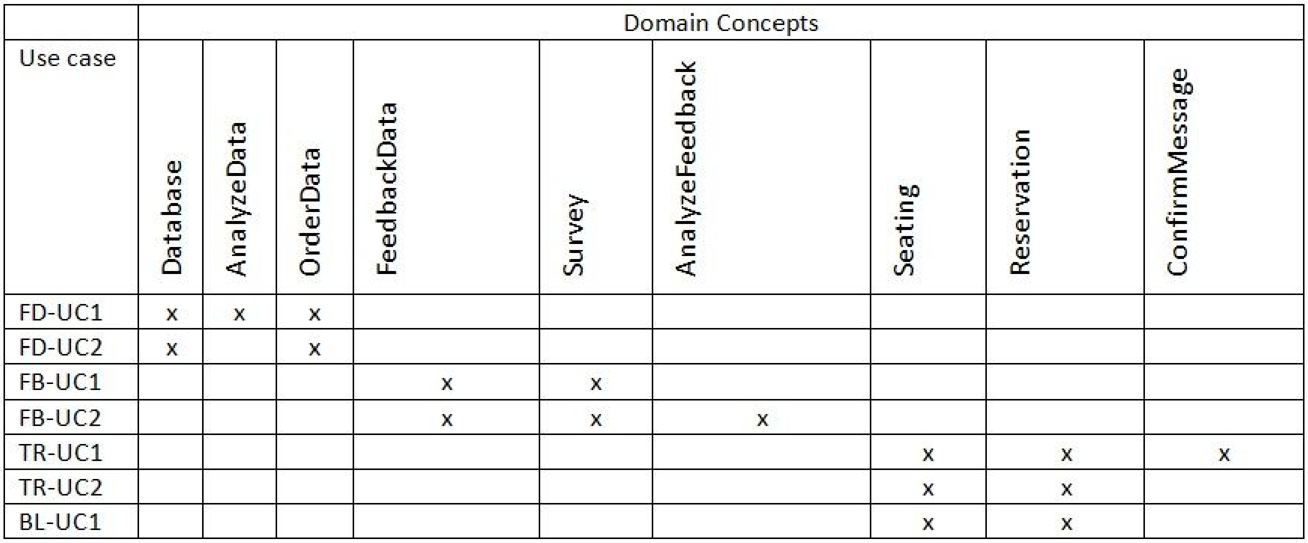
Concept Pair	Association description	Association name
OrderData <-> Database	Sends order data to database	Sends data
Database<->AnalyzeData	Analyzes the order history data using prediction algorithm	Analyzes data
FeedbackData<->DataTrends	DataTrends tracks data from feedback	Analyzes data
Survey<->FeedbackData	Survey sends data to FeedbackData	Stores data
FeedbackData<->AnalyzeFeedback	Analyzes survey data in Feedback	Analyzes data
Seating<-> Reservation	Check if seating is available and send message to customer	Reserves seats
Reservation <-> ConfirmMessage	Send confirmation message to customer	Sends message

3. Attribute Definitions

Concept	Attributes	Attribute description
AnalyzeData	Prediction Algorithm	Runs data through a prediction algorithm
Database	Customer data	contains order history data
	Display	Shows the user the order history and algorithm results
OrderData	Order data	Customer's order data
FeedbackData	Customer feedback data	Contains survey results (feedback) from customers
AnalyzeFeedback	Sorting algorithm	Analyzes and sorts customer feedback

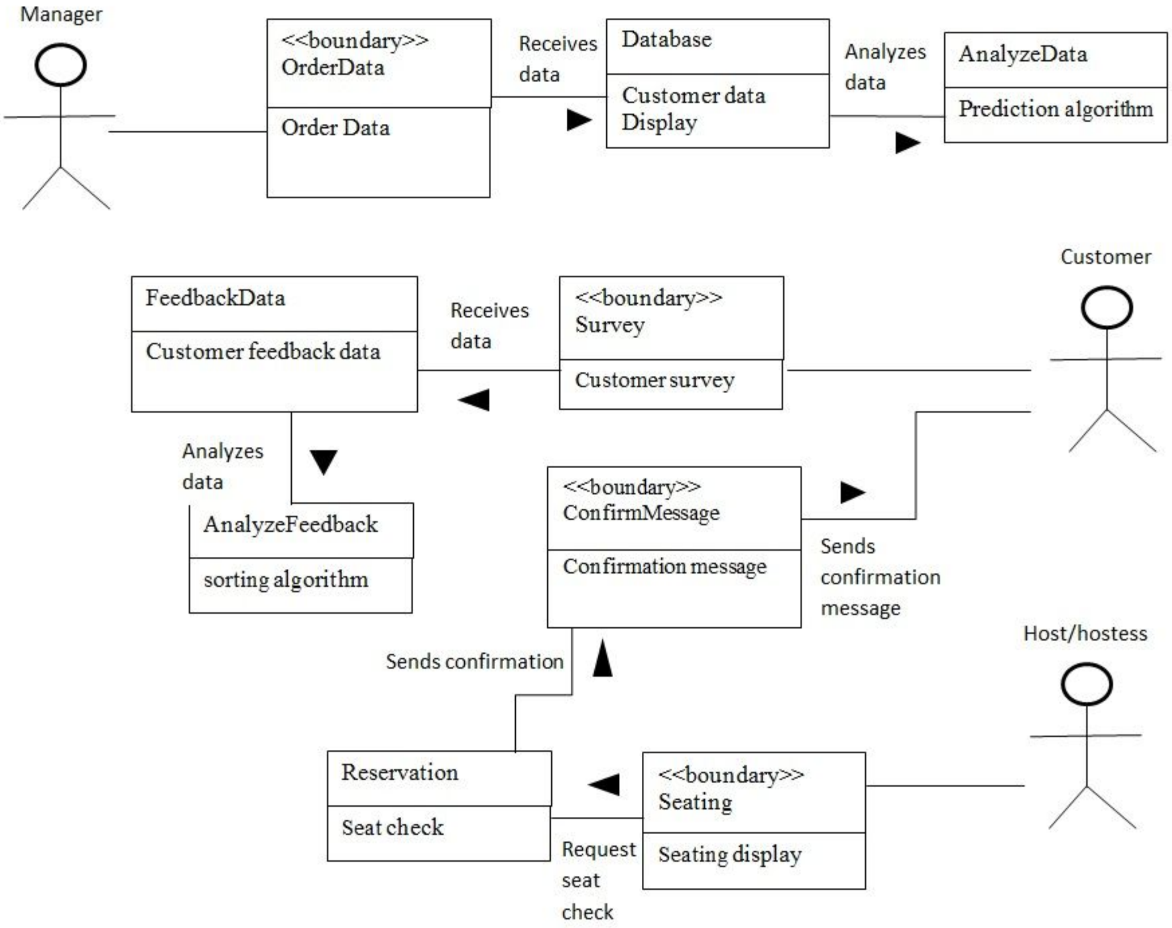
Survey	Customer survey	A survey for the customer to give feedback
Seating	Seating display	Shows seating information in real-time to the user
Reservation	Seat check	Receives reservation request from customer and checks for available seats
ConfirmMessage	Confirmation message	Sends confirmation message to customer

c. Traceability Matrix



	Domain Concepts								
Use case	Database	AnalyzeData	OrderData	FeedbackData	Survey	AnalyzeFeedback	Seating	Reservation	ConfirmMessage
FD-UC1	x	x	x						
FD-UC2	x		x						
FB-UC1				x	x				
FB-UC2				x	x	x			
TR-UC1							x	x	x
TR-UC2							x	x	
BL-UC1							x	x	

d. Domain Model Diagram



e. System Operation Contracts

FB-OC-1

Operation: submitTableSurvey()

Cross References: FB-UC-1

Preconditions: The customer has completed their dining experience and has filled out the survey.

Postconditions: - The data from the completed survey has been received and is ready for use by the system.

FB-OC-2

Operation: sendTableSurvey(surveyData)
Cross References: FB-UC-1
Preconditions: The survey response data is available to the system.
Postconditions: - The data from the completed survey has been recorded in the feedback database.

FB-OC-3
Operation: requestFeedbackData() + sendFeedbackData(feedbackData)
Cross References: FB-UC-1, FB-UC-2
Preconditions: The feedback database contains accurate and up-to-date data.
Postconditions: - The relevant feedback data from the database is available for use by the system.

FB-OC-4
Operation: analyzeSurveyResponse(surveyData)
Cross References: FB-UC-1
Preconditions: The survey data and relevant prior feedback data are available for analysis.
Postconditions: - New rating values have been averaged with prior ratings to create an up-to-date result.

FB-OC-5
Operation: updateRatingsData(ratingsData)
Cross References: FB-UC-1
Preconditions: The ratings data has been analyzed and is up-to-date.
Postconditions: - The feedback database has been updated to match the current data.

FB-OC-6
Operation: displayNotableResponseData() << extend>>
Cross References: FB-UC-1
Preconditions: The updated ratings information has been analyzed for any notable results.
Postconditions: - The administrative portal has received a notification in the event that the ratings data exceeds certain value thresholds.

FB-OC-7
Operation: submitTextInFeedback()
Cross References: FB-UC-2
Preconditions: The customer has completed their text-in response.
Postconditions: - The data from the response has been received and is ready for use by the system.

FB-OC-8
Operation: analyzeOpenEndedResponse(textInResponse)
Cross References: FB-UC-2
Preconditions: The text-in response and relevant prior feedback data are available for analysis.
Postconditions: - Response has been parsed for keywords and sentiment.
- Appropriate category for the response has been identified.
- New response is accurately depicted in the word cloud.

FB-OC-10
Operation: sortTextResponse(textInResponse)
Cross References: FB-UC-2
Preconditions: The response has been analyzed and an appropriate sorting category is selected.
Postconditions: - The feedback database has received the most recent text-in response.
- The new response is placed within the category selected through analysis.

FB-OC-11
Operation: displaySortedResponsesWordCloud() <<extend>>
Cross References: FB-UC-2
Preconditions: The word cloud and database have been updated to reflect current data..
Postconditions: - Up-to-date information of both sorted responses and the wordcloud are available through the administrative portal.

TR-OC-1
Operation: requestReservation()
Cross References: TR-UC-1
Preconditions: The customer has a desired time and party size for their reservation.
Postconditions: - The desired time and party size are known by the system.

TR-OC-2
Operation: requestScheduleData() + sendScheduleData()
Cross References: TR-UC-1
Preconditions: A reservation request is currently in progress.
Postconditions: - The current schedule is available to the system for use.

TR-OC-3
Operation: checkSeatingAvailability(time, partySize)
Cross References: TR-UC-1

Preconditions: The desired time and party size are known by the system and the current schedule is available.

Postconditions: - Availability of desired reservation is known.

TR-OC-4

Operation: updateSeatingSchedule()

Cross References: TR-UC-1

Preconditions: A reservation has been made for an available table at a known time.

Postconditions: - The seating schedule database has been updated to include the new reservation.

TR-OC-5

Operation: sendReservationConfirmation(time, tableNumber)

Cross References: TR-UC-1

Preconditions: A reservation for a specific time and table number has been successfully made.

Postconditions: - Customer has received confirmation of their reservation.

TR-OC-6

Operation: updateSeatingScheduleDisplay()

Cross References: TR-UC-1

Preconditions: The seating schedule database has been changed.

Postconditions: - The seating schedule display has been updated to match the database.

FD-OC-1

Operation: initiateSync()

Cross References: FD-UC-1

Preconditions: A system sync is desired and started by either the manager or the program itself.

Postconditions: - The system sync is initiated.

FD-OC-2

Operation: requestCurrentOrderData() + sendCurrentOrderData(currentOrderData)

Cross References: FD-UC-1, FD-UC-2

Preconditions: The current day’s order data contains accurate and up-to-date information.

Postconditions: - The relevant current order data is available for use by the system.

FD-OC-3

Operation: requestPreviousOrderData() + sendPreviousOrderData(orderData)

Cross References: FD-UC-1, FD-UC-2

Preconditions: The order database contains accurate and up-to-date information.

Postconditions: - The relevant previous order data is available for use by the system.

FD-OC-4

Operation: performPredictiveAnalysis(surveyData)

Cross References: FD-UC-1

Preconditions: Both current and prior order data are available for analysis.

Postconditions: - A new prediction is generated using the combined data.

FD-OC-5

Operation: updateOrderData(currentOrderData) + updatePredictionData(predictionData)

Cross References: FD-UC-1

Preconditions: A predictive analysis has been performed and its resultss must be recorded.

Postconditions: - The order database is updated to include the current day's order data.
- The order database prediction is updated to the most recent prediction.

FD-OC-6

Operation: requestIngredientUsagePrediction()

Cross References: FD-UC-2

Preconditions: Authorized user desires access to prediction data.

Postconditions: - The system begins accessing prediction data for use.

FD-OC-7

Operation: compareCurrentOrderDataWithPredictionData(currentOrderData, predictionData)

Cross References: FD-UC-2

Preconditions: Both current order data and most recent prediction are available for comparison.

Postconditions: - The relationship between the orders that have been placed and the orders that were expected is known.

FD-OC-8

Operation: displayIngredientUsagePredictionData(predictionData)

Cross References: FD-UC-2

Preconditions: The prediction data has been fetched from the database and compared with current orders.

Postconditions: - Raw prediction data is displayed to the user.
- Comparison between expected and current results is displayed to the user.

f. Mathematical Models

We are attempting to model the number and types of orders in a restaurant. These two features are interesting because it involves a classification problem intertwined within a continuous time series estimation problem.

The first thing that we should note is that the amount of business a restaurant gets can be approximately modeled by a periodic function. This is because restaurants have opening and closing hours every day. As most people know restaurants tend to have busier hours around meal times. This may include but is not limited to early morning for breakfast, midday for lunch, and early to late evening for dinner. However, we see a similar cyclic pattern almost every day. Some days might have larger peaks than others but the same cyclic patterns persist.

The number of people ordering at a restaurant at any given time is determined by the type of restaurant, the time of day, the day of the week, as well as possibly the season of the or month of the year as well. For example, an ice cream shop is much more likely to get business in the summer than in the winter months. Our system will look at these features through various models to analyze, predict, and refine our estimates. Different models have different advantages and disadvantages, so the system will compare the performances of each model and pick accordingly.

In order to implement these models, we need to format our inputs with the features the algorithm will analyze. Our current structure is to have our input feature vector contain the hour, day, month, and year. Our output feature will be the number of items for the particular order. Since our output feature will only contain one value, we will repeat this process for each order on the menu. The hours are represented as an integer from 0 to 23, the days range from 0 to 30, the months range from 0 to 11, and the year is just represented as is.

$$x = [hour, day, month, year]$$

$$y \ = \ [number\ of\ items]$$

This first model we plan on using stems from regression analysis with sinusoids. Due to the cyclic trends of restaurants orders we believe that using a sum of sinusoids can lead to better prediction. The model stems from a fourier analysis of the 24-hour period of daily food consumption. To fit this multivariate model, we used this model function where b is a vector of our model parameters.

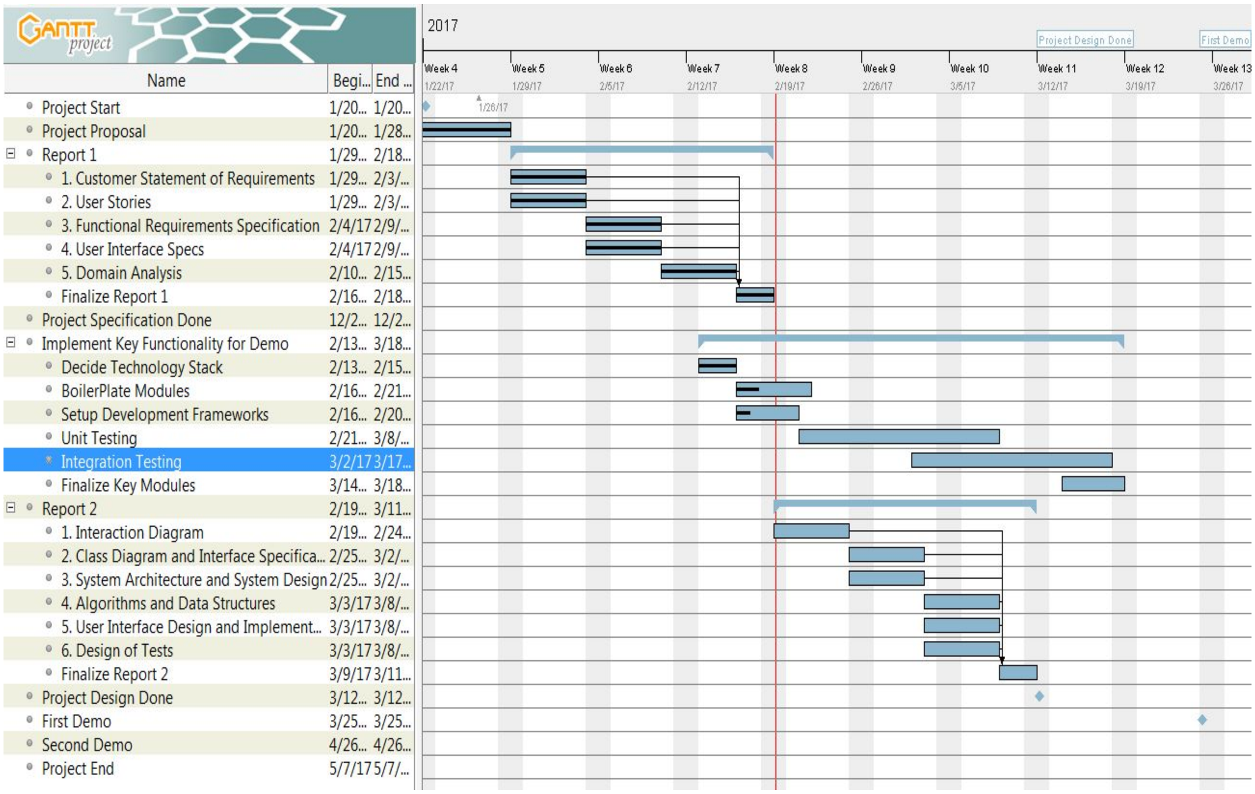
$$modelFunc \ = \ \sum_{i=1}^{length(x)} [b_{i,1} sin(x(:,i)/b_{i,2} + b_{i,3}) + b_{i,4} cos(x(:,i)/b_{i,5} + b_{i,6})] + b$$

To pick our initial parameter values, we simply guessed some random values that we believed would somewhat estimate the fitted line.

The second models attempts to model with regression using a polynomial function. This function may work better because it may be easier to account for the spikes which may occur as a result of seasonal or weekly trends as opposed to the sinusoidal series model. To fit this polynomial model, we used this function instead where m is the complexity of our polynomial.

$$modelFunc\ =\ \sum_{i=1}^{length(x)} [\sum_{j=1}^{m} b_{i,j} x(:,i)^{j}]\ + b$$

6. Plan of Work



Name	Begi...	End ...
Project Start	1/20...	1/20...
Project Proposal	1/20...	1/28...
Report 1	1/29...	2/18...
1. Customer Statement of Requirements	1/29...	2/3/...
2. User Stories	1/29...	2/3/...
3. Functional Requirements Specification	2/4/17	2/9/...
4. User Interface Specs	2/4/17	2/9/...
5. Domain Analysis	2/10...	2/15...
Finalize Report 1	2/16...	2/18...
Project Specification Done	12/2...	12/2...
Implement Key Functionality for Demo	2/13...	3/18...
Decide Technology Stack	2/13...	2/15...
BoilerPlate Modules	2/16...	2/21...
Setup Development Frameworks	2/16...	2/20...
Unit Testing	2/21...	3/8/...
Integration Testing	3/2/17	3/17...
Finalize Key Modules	3/14...	3/18...
Report 2	2/19...	3/11...
1. Interaction Diagram	2/19...	2/24...
2. Class Diagram and Interface Specifica...	2/25...	3/2/...
3. System Architecture and System Design	2/25...	3/2/...
4. Algorithms and Data Structures	3/3/17	3/8/...
5. User Interface Design and Implement...	3/3/17	3/8/...
6. Design of Tests	3/3/17	3/8/...
Finalize Report 2	3/9/17	3/11...
Project Design Done	3/12...	3/12...
First Demo	3/25...	3/25...
Second Demo	4/26...	4/26...
Project End	5/7/17	5/7/...

Short-term:

Subproblem A team - Richard and Zac:

Over the next few weeks, we plan to accomplish:

1. Configurable interface for the manager to view restaurant trends. (2-3 weeks)
2. Prediction service builtin to the manager and chef interfaces in order to see the daily predictions.
    a. Attempting to simulate real restaurant business trends (1 week)
    b. Collecting simulation data, storing, and accessing the data via DB (1-2 weeks)
    c. Training models to verify against our simulated data (2-3 weeks)
    d. Creating a user interface to easily make predictions and train the models (3 weeks)

Subproblem B team - Ben, Seo Bo, and Jarod:

Over the next few weeks, we plan to accomplish:

1. SMS interfacing for feedback and for reservation confirmation
2. Feedback parsing + organization (working demo)
3. Feedback forms design + delivery of form to backend. Unique ID for feedback Storage.

Subproblem C team - Brandon, Jeffrey:

Over the next few weeks, we place to accomplish:

1. Basic reservation interface for customer. Host will receive reservation info
2. Table management GUI for host.

Common Goals

1. Basic interface for high priority users (Manager, Chef, Host)
    a. Manager can see ingredients list + feedback
    b. Chef can see ingredients list of dishes and predictions
    c. Host can see basic GUI with tables, reservation list.
2. All employees can log in and out of website.
    a. Simple restaurant portal login (1-4 digit PIN)
3. Basic Employee Portal
    a. Can sign up for shifts.

Product ownership:

Each team will be responsible for the following features and qualitative property of the subproblem solutions. Throughout the course of the project, the following teams will implement the following functional features and qualitative properties of our project:

Subproblem A team - Richard and Zac:

- A simple easy-to-use interface to view the restaurant business trends
- A predictive service to determine the amount of food that will be needed on a given day.
- An interface to create custom dashboards with analytics based on manager preferences
- Voice commands to report to managers and chefs.

Subproblem B team - Ben, Seo Bo, and Jarod:

- A convenient post-meal survey that customers can complete to provide restaurant feedback.
- Feedback submissions remain confidential via each user being given a unique non-identifiable ID
- Customer receive a confirmation of feedback entry. (maybe include some of the organizing data that was parsed)
- Manager receives a summary and analysis of the feedback included
- Send out notification of action from management to customers who provided relevant feedback

Subproblem C team - Brandon, Jeffrey:

- An efficient and convenient system that allows customers to provide information and reserve a seat.
- Server notification that the table is now reserved as well as text/QR notification that the table is reserved for a guest.
- Manager/Host will receive seating confirmation and reserve the seating to prevent walk-in customers from taking the seat.
- Queueing of walk-in customers depends on the size of the group that comes in, basically table size demand queue.
- Designing a webpage for reservation systems, which implements a graphical interface that allows customers to reserve seating with the knowledge of where they are sitting.

7. References

- [1] - "Statistics on Food Waste in the United States", https://www.nrdc.org/issues/food-waste
- [Fig 0] ChefBoyRD Icon “Chef Boy” https://img.clipartfest.com/91a3a64a56686f2093016774a3f0ad63_boy-chef-pizza-chef-boy-clipart_236-236.jpeg
- [Fig 1a-1] Administrative Panel - “ UI for admin interface” “https://codecanyon.net/item/restaurant-order-mobile-app-android-ios/7668912
- [Fig 1a-2] Word Cloud - “Word Cloud“ http://theartofdoing.com/wp-content/uploads/2013/03/Chang2-copy.jpg
- [Fig 1a-3] Table View - “Table Management interface for POS software” http://www.vitabyte.com/wp-content/uploads/2012/11/Aldelo-Table-Management.png
- [Fig 1a-4] Reservation Form - “Online Reservation form” https://tableagent.com/static/img/screenreservation.png
- [Fig 1a-5] Employee Schedule - “Employee Schedule and Payroll“ https://s-media-cache-ak0.pinimg.com/736x/aa/11/bc/aa11bcdaafbaf8cc6927f6807ff848cd.jpg
- Excel or FluidUI (https://www.fluidui.com/) used to create charts/graphs, screen mockups.
- Plan of Work done using GanttProject (http://www.ganttproject.biz/)