

2017

Zest-Ware

Alexander Dewey

Fahd Humayun

Shehpar Sohail

| Ama Freeman

| Nathan Morgenstern

| Dwayne Anthony

| Raphaelle Marcial

IVAN MARSIC | Professor – Software Engineering

Table of Contents

<i>Part 1.1 Customer Statement of Requirements (CSR)</i>	1
<i> 1.a. Problem Statements:</i>	1
1.a.i. Seating.....	1
1.a.ii. Ordering	2
1.a.iii. Delivering order to the kitchen/changes to or cancellation of the order	3
1.a.iv. Payment	4
1.a.v. Entertainment and Feedback.....	5
1.a.vi. Bookkeeping.....	5
1.a.vii. Management Responsibilities	6
1.a.viii. Shift Tracking	6
1.a.ix. Pin number for employees	6
1.a.x. Expense and Income tracking.....	8
1.a.xi. Checking Inventory.....	8
1.a.xii. Kitchen Requirements	9
<i> 1.b. Glossary of Terms:.....</i>	11
1.b.i. Technical Terms	11
1.b.ii. Non-Technical Terms	13
<i>Part 1.2 System Requirements</i>	15
<i> 2.a. Enumerated Functional Requirements:</i>	15
2.a.i. Customer User Stories	15
2.a.ii. Kitchen User Stories	17
2.a.iii. Managerial User Stories	18
<i> 2.b. Enumerated Non-Functional Requirements:.....</i>	20
<i> 2.c. On-Screen Appearance Requirements:.....</i>	21
<i>Part 2.3 Functional Requirements Specification.....</i>	25
<i> 3.a. Stakeholders:.....</i>	25
<i> 3.b. Actors and Goals:.....</i>	26
3.b.i. Manager:.....	26
3.b.ii. Customer, Host:	27
3.b.iii. Waiter, Chef, and Busboy:.....	28

<i>3.c. Use Cases:</i>	29
3.c.i. Casual Description.....	29
3.c.ii. Use Case Diagram.....	32
3.c.iii. Traceability Matrix.....	33
3.c.iv. Full-Dressed Description.....	36
<i>3.d. System Sequence Diagrams:</i>	45
Part 2.4. User Interface Specification.....	53
4.a. Preliminary Design and User Effort Estimation.....	53
Part 3.5 Domain Analysis.....	70
3.5.a. Domain Model.....	70
3.5.a.i. Concept Definitions.....	73
3.5.a.ii. Association Definitions.....	79
3.5.a.iii. Attribute Definitions.....	81
3.5.a.iv. Traceability Matrix	85
3.5.b. System Operational Contracts	88
3.5.c. Mathematical Model.....	92
3.6. Plan of Work	93
3.7. References.....	96
End of project responsibility breakdown.....	98

Part 1.1 Customer Statement of Requirements (CSR)

A restaurant requires the seamless collaboration of the multiple employees who work in the facility. I, the restaurant owner, require solutions to help make time consuming and mundane tasks around the restaurant faster. I would like to eliminate time wasting and increase efficiency and overall income to the restaurant.

1.a. Problem Statements:

1.a.i. Seating

At the entrance of my restaurant, a host greets a customer and asks them if they have already made a reservation of the table (by phone call). If the customer has already made a reservation, then my host asks for the customer's name and checks in the notepad (for pre-reservations) and takes them to the reserved table. To provide my customers the service of pre-reservation, the host must keep track of the reserved table so that it is not occupied by other customers at the reservation time. If the customer does not have a pre-reservation, then the host writes the customer's name on their notepad. The host checks the "dry erase diagram of tables" for an available table that is suitable for the certain party size. This type of service requires the host to make frequent trips around the restaurant to update the diagram on the blackboard. It is a major concern for my restaurant because as my host becomes occupied in additional work, they have a hard time keeping track of available tables on the diagram and my customer waits for a longer period in the waiting area. In the meantime, customers frequently approach host at the front desk asking how much longer they should wait to be seated. When the tables become available, host calls out the names of the party to direct them to their designated tables. It is time consuming and hectic for my host to call out names, especially during high patron traffic and loud noise in the waiting area.

To eliminate this inefficiency, I want a touch screen to be installed at the main entrance with my restaurant's software running on it. When my customer enters the restaurant, they would be greeted by the screen (or host at the screen). The customer (or host) would be able to control/interact with the software through the touch screen. It should have all the necessary options needed for pre-reservations or new-reservations, and a floor map for the availability of tables to guide the customers to their table. There should be another touch screen installed in the waiting area, which would notify my customers when the table is available or show the estimated wait time for the availability of the table.

This would make the seating service of my restaurant organized, lower the time consumption for the customer to be seated, and productive for my business by attracting more customers because of the well-organized/managed services.

1.a.ii. Ordering

As my customer is seated at their designated table, my waiter gives them a menu to determine their order and leaves the table to serve other customers. When the customer is ready to place an order, they either call the waiter or wait for the waiter to check back with them. Once the waiter comes back to them and takes the order, then the waiter repeats the order at least once before leaving to avoid any error or confusion, since the waiter is writing down the order using a pen and notepad. Likewise, if the customer is not ready yet to place an order, the waiter is frequently checking back with the customer if they have decided on their meal and the customer does not feel comfortable (feels like being rushed) to place an order. So, either the waiter does not have a chance to revisit the customers (while the customer is ready to place an order) or frequently checks back while the customer is still deciding, because of which it takes about fifteen to twenty minutes (can be more time consuming during high patron traffic) to completely place an order.

The current service of placing an order is a major concern for me, because I want to serve my customers with the idea: "*they want what they want and when they want it*" *strongly in mind*.

A touch screen should be installed at each table with my restaurant's software running on each one of them. The screen should provide the customers with the options to view the entire menu with images of the items, place an order, and make changes to the order.

By providing this service, it would have a great impact on the quality of services of the restaurant, as this would be such a convenience for the customer to place an order or make changes to the order. Hence, I would be able to serve the customers with the phrase: "*they want what they want and when they want it*" *in mind*.

1.a.iii. Delivering order to the kitchen/changes to or cancellation of the order

When the waiter takes the order from a customer, they take it to the kitchen and stick the order (from the notepad) with the table number on the board in the kitchen. The kitchen staff then takes the order from the board one at a time to prepare the food. Meanwhile, the waiters go back and forth to check on the status of the food. These visits of the waiter to the kitchen and not knowing if the food is ready (or not) is absurd, because, either the food would not be ready and they should go back to the floor or the food might be ready but being unaware of the prepared food – the food would be sitting in the kitchen to be taken to the table.

More concerns arise when the customer wants to make changes or wants to cancel their order. When the waiter takes the order from the customer, the waiter may only return to the customer once to serve entrées before the food is ready for the customer. This means that the customer wants assistance from the waiter, but needs to wait until the waiter notices them or the waiter is free to assist them. Once the waiter comes to the table to assist them, the waiter would note down the items being added/removed to the order (that was already placed). Now, as the waiter left the already placed order on the kitchen's notice board, they must go to the kitchen, look for the order with the specific table number, then add the changes made to the order with the order stuck to the notice board, and notify the kitchen staff about the changes made to the order. Similarly, when the customer wants to cancel their order, as mentioned they wait for assistance. When the waiter is available to assist them with the cancellation of the order, the waiter goes back to the kitchen to check if the food is not being prepared or not being placed for preparing yet. The waiter then returns to the customer to notify them about the status of their food, and that whether the order can be cancelled or not.

There should be a screen installed in the kitchen for the kitchen staff, which would get information related to the orders placed, orders cancelled, and changes made to the order (from the screens on the tables at the floor). Another screen should be installed for the waiter to check the status of the food – if being prepared to be delivered to the customer. The customer should be able to see the status of their food in process, and should be allowed to make changes or cancel the entire order through their table screen. This would reduce the wait time and the hassle by making the information flow faster and convenient between the customer, the waiter, and the kitchen staff.

1.a.iv. Payment

There is unnecessary wait time when the customer has finished eating and is waiting for the payment. The current service in my restaurant is that when a customer has finished eating, they wait for assistance from the waiter. Once the waiter is available, the customer asks for their bill. The waiter then goes to the kitchen to get the order of the specific table from the notice board (in the kitchen), which they wrote down on their note pad. They bring the order to the cashier department, who makes the bill. The waiter takes the bill, returns to the customer, and then leaves so that customer gets the payment ready. The waiter comes back to the customer in a couple of minutes to receive the payment (cash or card) from the customer, then goes to the cashier to either process the card payment or to get change for the customer. The waiter then returns to the customer with the receipt and possibly change to complete the payment process. The process to make a payment takes about ten to fifteen minutes.

Apart from all of this, there is a concern of split payment among the customers, which causes the payment process to take even longer than usual.

I would like to have the option of payment available on the screen installed at the table, so that the customer can easily make a payment through the screen if they are paying by card. If the customer is making cash payment, there should be an option that would call assistance of the waiter for the payment, so that the waiter comes with the bill rather than figuring out what the customer needs assistance with, then getting the bill, and processing the payment. The software should also allow split payment with an ease and convenience. The software should help with the information flow between the customer, the waiter, and the cashier.

Furthermore, I would also like to enable payments in the form of Bitcoin. The customer should be able to scan the table screen with their phone to complete their payment. The customer, my staff, and I should not have a hard time processing the payments i.e. the process should not require a great deal of technical expertise.

1.a.v. Entertainment and Feedback

There is lack of entertainment for the customers after the order is being placed. Many parents with young children want their meal urgently because it is hard for them to control their children – while wait for their meal to be served to them. Also, during rush hours, most of the customers get bored because the wait time for each customer is even greater.

Currently, there is no service for getting feedback from the customer.

While the customer is waiting for their meal, the software system should help them by providing them with entertainment such as using the internet, playing free games, or provide them with a feedback/survey related to the restaurant (whoever does the feedback/survey will get special offers).

1.a.vi. Bookkeeping

One of the other concerns is of keeping the record that includes the number of customers arrived, the orders placed, and the payments made. The current service of my restaurant is; the host keeps noting down the arrival time and the number of customers arrived at that time in their notepad, the waiter stores their notepad papers on which they have all the orders placed/changed/cancelled, and the cashier stores the receipts of all the payments made. It is a tedious process of bookkeeping, because most of the time the papers/receipts gets misplaced, and this is also hectic for my management to keep all the papers/receipts properly organized/managed. If we want to check record of a specific date/time, it gets frustrating by going through the papers/receipts.

The software system could help us by keeping track and storing all the ongoing activities in the restaurant related to the customer's arrivals/orders/payments, and help us to access the records of each date/time.

Summary of problems and suggestions related to customer and staff interaction

There is lack of communication among the restaurant's staff and the customer, which causes unnecessary delays and decreases quality of services provided by the restaurant. The restaurant also lacks in keeping the record of ongoing activities.

The restaurant needs to have a software system that would help with a reliable, efficient, and faster information flow among the staff and the customers, which will increase the quality of the services, the productivity, and the reputation of the restaurant. The software system should also help with the bookkeeping to make it easily accessible and manageable.

1.a.vii. Management Responsibilities

In my restaurant, my managers' role is crucial to maintaining normal operations. My managers are the sole organizer of the restaurant's employees and finances. However, both of these components consist of multiple smaller and complicated parts. Due to growing amount of responsibilities my managers have, automation would benefit them greatly by keeping their tasks to a minimum. The managers in my restaurant have to keep account of all their employees. This includes knowing their shifts, being aware of whom the employees are serving, and financing their payroll. More importantly, my managers must make sure that my customers are satisfied. This task includes obtaining feedback from them, and making any changes to accompany their needs. All these tasks lead to one main focus for my managers, to make sure that my restaurant is making profit. Observing whether the restaurant is making its quota then taking appropriate actions, and keeping track of inventory are integral when taking care of the restaurant's finances. Since my managers are the crux of the organization and maintenance of software. The software system is the solution for my managers, who must take on these seemingly insurmountable tasks.

1.a.viii. Shift Tracking

Regarding my restaurant, managing the shifts of multiple employees monthly, weekly, and even possibly daily is an extremely complex process. My employees change very often in the restaurant environment; the employees working on a Monday could be completely different from the employees working on Friday, and vice versa. Thus, my manager needs to be able to keep track of constantly changing schedules, especially if the restaurant manager changes throughout the week as well. Therefore, a system to keep track of all employees working at different times is optimal. The system should not confuse my manager and use visuals such as calendars to easily see days with, for example, too few employees.

1.a.ix. Pin number for employees

When punching into their respective shifts, my employees need a quick and effective process to do so. Keeping around punch cards is not efficient for my restaurant, as this requires my managers to check and verify the cards, leaving room for errors in calculations. These processes could cause even more time-consuming issues in the future of the restaurant. Therefore, my employees could punch into the system using a quick form of identification in the form of a memorized pin number. All workers in my restaurant will be required to punch in using this system. Also, the system should keep track of hours as my employees punch in and out.

1.a.x. Expense and Income tracking

Along with payroll and shift tracking, my managers must also keep track of the expenses and income of my restaurant. The software should clearly define the totals for a day, which include the payments from my customers. Expenses such as damages to my restaurant and incoming ingredient charges should be held in the program as well. Basic addition, subtraction, and other calculations should be accessible by my managers using the program when checking totals.

1.a.xi. Checking Inventory

When checking inventory in my restaurant, my managers and employees rely on a board in the kitchen that is updated only when an item is out of stock. This is a waste of time for my employees who could make better use helping each other, or assisting my customers. Moreover, because the board is manually updated, there is no prior warning given to my employees that the inventory is running low. No one knows about the shortage until my chef realizes this conflict and tries to flag down another employee to write on the board. This is also a problem for my customers who believe that the dish they ordered is cooking, but would be forced to order another item due to an ingredient shortage.

The software should upgrade this outdated “hanging board” method. Once my chef realizes that their inventory is running low, they should be able to put in a request through the system to inform my managers that it needs to be restocked. This would lead to a notification that would appear on all applicable screens that the item used in the respective dish is no longer available to order. For my customers who are viewing this alert, they would be notified that the dish they would like to order is no longer available. However, my chefs, waiters, and managers would be able to view both the dish and the ingredient, which is out of stock. Additionally, the system would push a request to my managers to order more through the supplier. The manager could then decide whether they want to immediately send the order, or wait until the end of service. Although it is possible to fully automate the system so that an order can be sent automatically when inventory is low, my chef should ultimately decide if they have enough ingredients along with my manager’s approval.

1.a.xii. Kitchen Requirements

There are a number of capabilities that I insist this software gives to my kitchen staff. Firstly, the waiters should be notified when they are needed by a customer, and should be able to ensure satisfaction and follow up on their services in a more efficient manner. They should also be able to cancel, edit, and make orders for tables in the event that our guests do not want to use the technology themselves. In the same respect, our waiters must be able to perform the checkout process for customers if they are uncomfortable with performing it themselves.

Furthermore, any part of the table staff, but particularly the waiters and busboys, shall be notified precisely when a meal is ready so there is a limited wait time for the guests' food to be delivered. The busboys' other table duties should be integrated in this software as well. While I cannot quite imagine a way for a computer system to help the dish washing process, the system should notify the busboys when a table needs to be cleaned after the guests have checked out, and subsequently when a table needs to be set.

The hostess needs to be able to view a frequently updated floor plan showing which tables are free for seating. With this tool, the hostess must be able to effectively seat a table and assign a waiter to the guests for further service. They should also be able to update this floor plan themselves so the rest of the staff also knows when a table has been seated. Also, they should be able to view which servers have which and how many tables, so they can appropriately assign tables based on how many tables each server has and how spread out they are. This feature will be very important in the eyes of the servers, as they make a good bit of their money in tips and they should not feel as if they are getting slighted.

Finally, the chefs need features on their end of the software that can help them more efficiently prepare the orders that are coming into the kitchen from the tables. They need to be able to receive orders on their screen in a queue system as soon as they are entered by the guests. This screen should also be updated the second any order that has been placed has been edited or canceled so that the chefs' time is compromised as little as possible. The chef also absolutely needs the ability to notify the rest of the staff when an order is ready for delivery. With up to the second notifications, the freshness and quality and the meals should be improved.

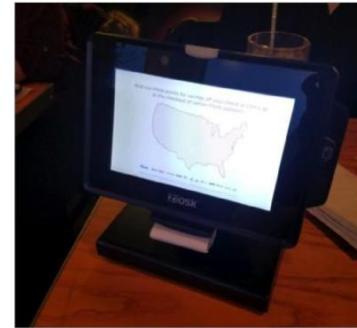
Another aspect that is perhaps not as important as the other requirements but would still go a long way in terms of customer satisfaction would be a "birthday button". This button could be implemented so that it could be pressed by guests in order to notify the staff that there is a birthday at their table. This would prompt our staff to perform our birthday routine followed by a dessert like most restaurants do. Perhaps the staff could be notified with optional information like the birthday person's first name to add into our song or his favorite dessert offered on our menu so that they can be more satisfied with the experience.

I would also like the software to include a scheduling assistant for the restaurant's employees that they can access remotely from an internet browser. In this portion of the software, the first thing the employee should be able to is, of course, view his schedule. From there, if he wants to see if someone can cover a future shift, he can choose a shift and press a button to request a cover for it. Conversely, if he is free at a certain time and would wish to pick up a shift, he can request that as well. In addition to this, the employees should have the ability to things like request off for unscheduled days.

1.b. Glossary of Terms:

1.b.i. Technical Terms

- **Tablet:** Physical interface where the “portal” will be held. Customers will be able to order, take surveys, and entertain themselves on the tablets.
- **Computer:** Another physical interface where the “portal” will be accessed from. Managers and other employees will be able to view table status, food readiness, etc.
- **Portal:** Software “hub” where managers, employees, and customers can access items relating to themselves.
- **Customer Table Screen (CTS):** A touch screen that has the restaurant’s software running and is installed to improve communication between the customer and restaurant staff. It is located at the side of every table within the restaurant and allows customers to easily perform actions such as ordering, playing games, payments, etc.



- **Customer Welcome Screen(CWS):** A screen installed at the entrance of the restaurant that displays the floor plan of the restaurant, availability of tables, and an estimate wait time for customers.
- **Customer Waiting Area Screen (CWAS):** A screen installed in the waiting room of the restaurant. It displays the estimated wait time for the availability of the table and notifies customers when the table is available.
- **Restaurant's Database:** Storage of information related to the restaurant (menus, customer order, customer payment, inventory, scheduling, payroll, order status, table status, statistics, etc.).
- **Open-source games:** Free games available for customers to play on their CTS.
- **Add/Remove:** A tab in the CTS that allows the customer to add or remove a dish from their order.
- **Activities:** A tab in the CTS that allows the customer to play open-source games, give feedback, take a survey, or use the internet.
- **Call Waiter/Waitress:** An option on the CTS for the customer to contact the waiter when assistance is needed.



- **Facial Recognition:** A system which verifies a person's identity based on their facial features. This can be used as an extra security measure.
- **Bitcoin Public Key:** The “account number” that customers will send payments to. If a customer chooses the bitcoin payment option, they will be displayed a public key to send payment to.
- **Bitcoin Private Key:** The private key is essentially the “password” of the Bitcoin funds stored on the address. This should be secured by the manager and ideally not touch the internet.



1.b.ii. Non-Technical Terms

- **Restaurant** - An establishment which serves food to customers for a profit
- **Customer:** The main source of income of the restaurant. These are third party human beings who bring business to the restaurant. Their satisfaction is important. Cumbersome tasks must be eliminated in order to ensure the frequent return of customers.
- **Manager:** The head of the restaurant, who handles numerous responsibilities ranging from employees to customers to finances. The manager oversees operations, and continuously solve problems to keep customers satisfied. A person with access to the status of employees, tables, and the finances (to name a few) of the restaurant. It is imperative to make the job of the manager simpler and more efficient.
- **Employee Payroll:** System that keeps track of the hours of working employees which include waiters and waitresses, chefs, busboys, and the manager.
- **Expenses:** Costs to be paid by the restaurant. These include, but are not limited to, payroll, ingredients, food stuffs, and damages to the restaurant.
- **Feedback:** Comments from the customer which will be used towards improving restaurant operations.
- **Finances:** The management of money.
- **Inventory:** A varying amount of ingredients used to prepare dishes in the restaurant.
- **Pin Number:** 4 to 6 digits number to be used by working members of the restaurant to punch in to work.
- **Stock:** Refers to the inventory within the restaurant.
- **Quota:** A goal pertaining to money/satisfaction that the restaurant attempts to reach.
- **Kitchen:** (The kitchen consists of chefs and busboys. The chefs prepare the food and the busboys clean and collect dishes and clear tables when the table has been vacated by customers.)
- **Customer Satisfaction:** The extent to which a customer's needs are met.
- **Waiter/Waitress:** An individual that serves customers at tables in the restaurant.

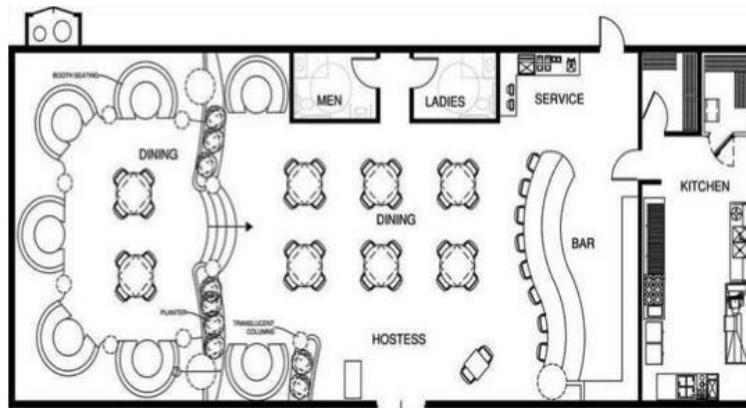
- **Busboy:** An individual who cleans and sets the table after customers have finished their meals.
- **Menu:** A list of the items that are served to the customers at the restaurant. Each item on the list is accompanied by an image for customer convenience on the CTS.
- **Payment:** An amount of money given by the customer in exchange for food or other services provided by the restaurant



- **Table Pre-Reservation:** An early booking of a table at the restaurant.
- **Table New-Reservation:** A booking of a table by walk-in.
- **Table Status (Occupied):** A table is busy being used by current customers.
- **Table Status (Available):** A table that is not busy and is able to be used by current customers.
- **Order:** Items that a customer requested from the menu.
- **Offers:** Special discounts on specific food items.
- **Order Status:** Percentage that a customer's food has been cooked at the moment.



- **Floor Plan:** A visual diagram about the positions of the tables and rooms in the restaurant.



Part 1.2 System Requirements

2.a. Enumerated Functional Requirements:

2.a.i. Customer User Stories

Table 1: Functional Requirements – Customer User Stories.

Identifier	User Story	Size	Priority
C - 01	<i>As a customer (or host), I can choose pre-reserved table or new-reservation of table.</i>	7 pts	High
C - 02	<i>As a customer (or host), I can input the name for looking up of a pre-reserved table.</i>	4 pts	High
C - 03	<i>As a customer (or host), I can input the name and party size for new-reservation of table.</i>	7 pts	High
C - 04	<i>As a customer (or host), I can view the floor map, and I can assign the preferred seating from the available tables for the party size.</i>	5 pts	High
C - 05	<i>As a customer (or host), I can make future reservations based on the availability by inputting my name, party size, time, and date.</i>	4 pts	Medium
C - 06	<i>As a host, I can view a list of all the pre-reservations for a specific time and date.</i>	4 pts	Low
C - 07	<i>As a customer, if my table is not available, I am given an estimated wait time for the availability of table, and I am guided to a waiting area.</i>	7 pts	Medium
C - 08	<i>As a customer, if (or once) my table is available, I am notified and guided to my table.</i>	9 pts	Low
C - 09	<i>As a customer, I can view the entire menu (with categories) on the table to place an order.</i>	5 pts	High
C - 10	<i>As a customer, I can add and remove items from my order.</i>	4 pts	High
C - 11	<i>As a customer, I can make changes to current order or cancel order before processing begins.</i>	6 pts	High

<i>C – 12</i>	<i>As a customer, I can see the estimated wait time after placing my order.</i>	<i>8 pts</i>	<i>Medium</i>
<i>C – 13</i>	<i>As a customer, I can see the current stage/process/status of my order.</i>	<i>6 pts</i>	<i>Medium</i>
<i>C – 14</i>	<i>As a customer, I can call a waiter for assistance.</i>	<i>2 pts</i>	<i>Medium</i>
<i>C – 15</i>	<i>As a customer, I can make payment with my card without the assistance of the waiter.</i>	<i>5 pts</i>	<i>High</i>
<i>C – 16</i>	<i>As a customer, I can make cash payments with the assistance of a waiter.</i>	<i>1 pts</i>	<i>High</i>
<i>C – 17</i>	<i>As a customer, I can make split payments.</i>	<i>3 pts</i>	<i>High</i>
<i>C – 18</i>	<i>As a customer, I can view special offers before and after placing an order.</i>	<i>2 pts</i>	<i>Low</i>
<i>C – 19</i>	<i>As a customer, I can have entertainment like use the internet, or play free games, while I am waiting for my meal.</i>	<i>7 pts</i>	<i>Medium</i>
<i>C – 20</i>	<i>As a customer, I can give feedback or take a survey, and receive an incentive bonus/offer.</i>	<i>4 pts</i>	<i>Medium</i>

2.a.ii. Kitchen User Stories

Table 2: Functional Requirements – Kitchen User Stories.

Identifier	User Story	Size	Priority
K - 01	<i>As a waiter, I can place orders for the kitchen to cook.</i>	8 pts	Medium
K - 02	<i>As a waiter, I can be notified that a customer needs assistance via my portal.</i>	3 pts	Low
K - 03	<i>As a waiter, I edit the order (cancel or add meal) for the kitchen to update.</i>	7 pts	Medium
K - 04	<i>As table side staff (waiter), I can deliver the food for the customer.</i>	6 pts	High
K - 05	<i>As a waiter, I can perform the checkout process for a customer.</i>	10 pts	Low
K - 06	<i>As a busboy, I can clean the table after the checkout process has been completed.</i>	8 pts	Medium
K - 07	<i>As chef, I can prepare a meal for table staff to deliver.</i>	10 pts	High
K - 08	<i>As a chef, I can cancel a meal in the current queue.</i>	6 pts	Low
K - 09	<i>As busboy, I can set the table for the next customer.</i>	9 pts	Medium

2.a.iii. Managerial User Stories

Table 3: Functional Requirements – Managerial User Stories.

Identifier	User Story	Size	Priority
M - 01	<i>As a manager, I should know the availability of tables in the restaurant.</i>	8 pts	Medium
M - 02	<i>As a manager, I need to know the shifts of all of my employees and be able to easily move, delete, or add shifts.</i>	10 pts	Low
M - 03	<i>As a manager, I need to be able to change the menu easily at the touch of a button.</i>	6 pts	High
M - 04	<i>As a manager, I need to know how satisfied my customers are.</i>	8 pts	High
M - 05	<i>As a manager, I must know what menu items bring in the most income and which bring in the least.</i>	7 pts	Medium
M - 06	<i>As a manager, I need a database to handle mounting expenses and the ability to total my daily, weekly, or monthly quota.</i>	9 pts	Medium
M - 07	<i>As a manager, I shall be able to see restaurant inventory.</i>	6 pts	High
M - 08	<i>As a manager, I shall be able to add/delete employees from the software.</i>	10 pts	Low
M - 09	<i>As a manager, I shall know which waiter served each table.</i>	7 pts	Medium
M - 10	<i>As a manager, I shall know which chef prepared each dish.</i>	7 pts	Medium
M - 11	<i>As a manager, I shall be able to control when to order more inventory.</i>	5 pts	High
M - 12	<i>As a manager, I shall know which busboy prepared each table.</i>	7 pts	Medium

<i>M - 13</i>	<i>As a manager, I should be able to control the music playing in the restaurant.</i>	<i>8 pts</i>	<i>Low</i>
<i>M - 14</i>	<i>As a manager, I should be able to print reports based on the restaurant's expenses, inventory, and feedback.</i>	<i>4 pts</i>	<i>Low</i>

2.b. Enumerated Non-Functional Requirements:

Table 4: Enumerated Non-Function Requirements.

Identifier	User Stories	Size	Priority
NFR - 01	<i>As a user of the software, I should be able to install on the operating system of my choice.</i>	5 pts	Low
NFR - 02	<i>As a manager, access to my portal should be limited to myself, and possibly an assistant manager.</i>	6 pts	High
NFR - 03	<i>As a user of the software, I should not experience wait times longer than one second.</i>	7 pts	Medium
NFR - 04	<i>As a chef, I should be able to read what I need to cook from about a meter away.</i>	2 pts	Medium
NFR - 05	<i>As an employee, it should take no more than two clicks/taps to get to my respective portal.</i>	5 pts	High
NFR - 06	<i>As a waiter, I should be notified when my guest's order is ready.</i>	4 pts	High
NFR - 07	<i>As a waiter, it should only take two clicks/taps to inform the system that I have delivered an item.</i>	2 pts	Medium
NFR - 08	<i>As a chef, it should only take two clicks/taps to remove an item off the queue.</i>	3 pts	Medium
NFR - 09	<i>As a chef, it should only take three clicks/taps to update the progress of a meal.</i>	4 pts	Low

2.c. On-Screen Appearance Requirements:



Figure 1: Customer Welcome Screen (CWS) and Pre-Reservations Lookup.



Figure 2: New-Reservations and Floor Map.

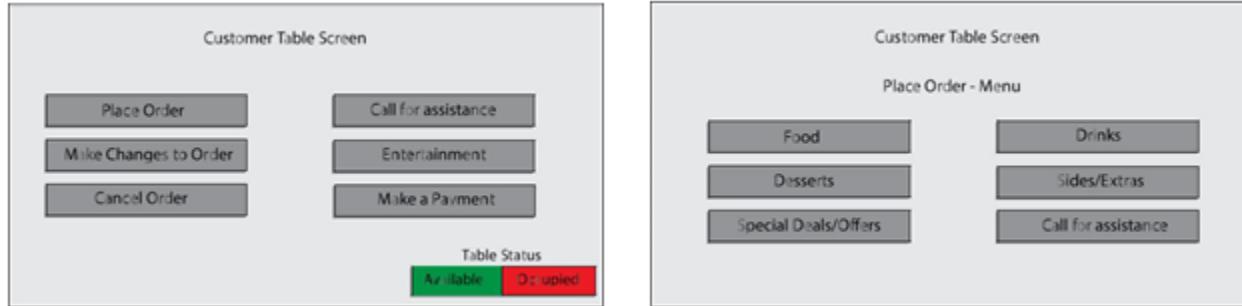


Figure 3: Customer Table Screen (CTS) and Place Order.

CWS - Pre Reservations Lookup			
By Name	<input type="text"/>		
By Date	<input type="text"/> MM/DD/YYYY		

Customer Waiting Area Screen			
Customer's Name	Party Size	Table #	Est. Wait Time
1. John Doe	X	Y	Z
2. Jane Doe	A	B	C

Figure 4: CWS - Pre-Reservations Lookup and Customer Waiting Area Screens (CWAS).

CTS - Payment			
<input type="button"/> Cash	<input type="button"/> Bitcoin		
<input type="button"/> Credit/Debit Card	<input type="button"/> Paypal		

CTS - Food Progress/Status	
Progress Bars & Indicators	

Figure 5: CTS - Payment and CTS - Food Progress Bar.



Log Out		Search for item...	Add New Item	Sorting Options
Item Name	Item Quantity		Item Status	Note
Ground Beef	5 oz.	<button>Order Item</button>	!!! Low	
Sesame Seed Buns	30 pcs.	<button>Order Item</button>	!! Moderately Low	
Lettuce Heads	50 pcs.	<button>Order Item</button>	In Stock	
Tomatoes	50 pcs.	<button>Order Item</button>	In Stock	
Mayonaisse	20 oz.	<button>Order Item</button>	!! Moderately Low	
Ketchup	20 oz.	<button>Order Item</button>	!! Moderately Low	
Mustard	1 oz.	<button>Order Item</button>	!!! Low	
American Cheese	120 pcs.	<button>Order Item</button>	In Stock	



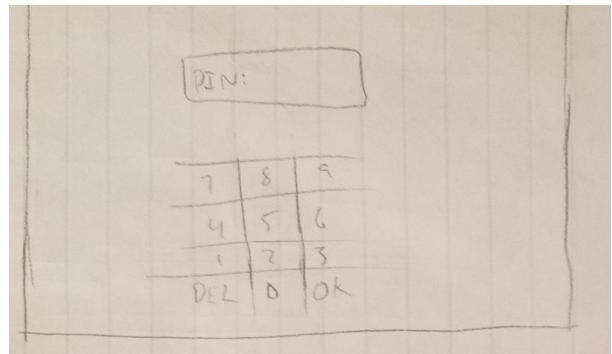
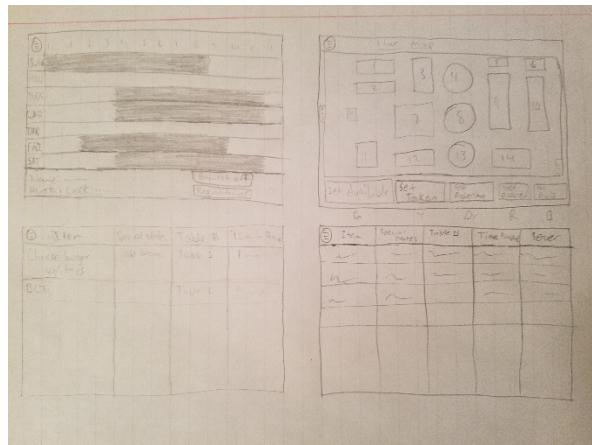


Figure: Kitchen Screen

Part 2.3 Functional Requirements Specification

3.a. Stakeholders:

- Restaurant
- Manager
- Customer
- Waiter
- Host
- Chef
- Busboy

3.b. Actors and Goals:

3.b.i. Manager:

Table 5: Manager as actor and their goal.

Actor	Type of Actor	Actor's Goal	Use Case Name
Manager	Initiating	<i>To check the availability of tables in the restaurant.</i>	<i>ShowTableAvailability (UC - 01)</i>
Manager	Initiating	<i>To check and edit shifts of employees of the restaurant.</i>	<i>EditShifts (UC - 02)</i>
Manager	Initiating	<i>To make changes to the menu.</i>	<i>Delete/Change/AddMenuItem (UC - 03)</i>
Manager	Initiating	<i>To know the satisfaction of customers based on customer surveys.</i>	<i>ShowSurveyData (UC - 04)</i>
Manager	Initiating	<i>To see the most popular menu items and what menu items bring in the most income.</i>	<i>ShowMostPopularItems, ShowHighestIncomeItems (UC - 05)</i>
Manager	Initiating	<i>To have access to a database with mounting expenses.</i>	<i>ShowExpenses (UC - 06)</i>
Manager	Initiating	<i>To total daily/weekly/monthly expenses quota.</i>	<i>SumWeek/Month/DayExpenses (UC - 07)</i>
Manager	Initiating	<i>To check the restaurant inventory and easily order low ingredients.</i>	<i>CheckInventory (UC - 08)</i>
Manager	Initiating	<i>To add or delete employees from the software system (hiring/firing).</i>	<i>Remove/AddEmployee, ChangeEmployeeData (UC - 09)</i>
Manager	Initiating	<i>To know the specific tables served by each waiter.</i>	<i>ShowActiveWaiters (UC - 10)</i>
Manager	Initiating	<i>To know which chefs prepared each dish.</i>	<i>ShowActiveChefs (UC - 11)</i>
Manager	Initiating	<i>To know the specific tables cleaned by the busboy.</i>	<i>ShowActiveBusboys (UC - 12)</i>
Manager	Initiating	<i>To change the musical ambience of the restaurant.</i>	<i>Change/AddMusic (UC - 13)</i>
Manager	Initiating	<i>To have quick and secure access to the manager's portal.</i>	<i>AccessPortal (UC - 14)</i>

<i>Manager</i>	<i>Initiating</i>	<i>To print daily reports about the restaurant (quota/feedback/inventory).</i>	<i>PrintProtal (UC - 15)</i>
----------------	-------------------	--	------------------------------

3.b.ii. Customer, Host:

Table 6: Customer, Host, and Restaurant's Database as actors and their goals.

Actor	Type of Actor	Actor's Goal	Use Case Name
<i>Customer</i>	<i>Initiating</i>	<i>To check in to pre-reservation and sit down at table.</i>	<i>PreReservation (UC - 16)</i>
<i>Customer</i>	<i>Initiating</i>	<i>To check in and be seated at the next available table.</i>	<i>NewReservation (UC - 17)</i>
<i>Restaurant's Database</i>	<i>Participating</i>	<i>To verify the reservation is valid (Table and Date are not taken).</i>	<i>VerifyReservation (UC - 18)</i>
<i>Restaurant's Database</i>	<i>Participating</i>	<i>To add a new reservation, if there are currently none at the given date and time.</i>	<i>AddNewReservation (UC - 19)</i>
<i>Restaurant's Database</i>	<i>Participating</i>	<i>To notify customer (or host) that a table has become available, and they can be seated.</i>	<i>NotifyCustomer/Host (UC - 20)</i>
<i>Customer</i>	<i>Initiating</i>	<i>To add an item from the menu to the current order.</i>	<i>AddMenuItem (UC - 21)</i>
<i>Customer</i>	<i>Initiating</i>	<i>To remove an item from the current order</i>	<i>RemoveMenuItem (UC - 22)</i>
<i>Customer</i>	<i>Initiating</i>	<i>To finalize the order and send the information to the kitchen.</i>	<i>OrderItems (UC - 23)</i>
<i>Customer</i>	<i>Initiating</i>	<i>To cancel the entire order.</i>	<i>CancelOrder (UC - 24)</i>
<i>Restaurant's Database</i>	<i>Initiating</i>	<i>To calculate the estimated wait time for the customer's order to be prepared.</i>	<i>FoodWaitTime (UC - 25)</i>
<i>Customer</i>	<i>Initiating</i>	<i>To use the internet or play free games while waiting for order.</i>	<i>Entertainment (UC - 26)</i>
<i>Customer</i>	<i>Participating</i>	<i>To find out the status of the order.</i>	<i>FoodStatus (UC - 27)</i>

<i>Customer</i>	<i>Initiating</i>	<i>To give feedback or take survey and earn an incentive bonus/offer.</i>	<i>SpecialOffers/TakeSurvey (UC - 28)</i>
<i>Restaurant's Database</i>	<i>Initiating</i>	<i>To adjust bill based on incentive bonus/offer.</i>	<i>AdjustBill (UC - 29)</i>
<i>Customer</i>	<i>Initiating</i>	<i>To make a payment with a card without the waiter's help.</i>	<i>CardPayment (UC - 30)</i>
<i>Waiter</i>	<i>Participating</i>	<i>To help customer complete the cash transaction.</i>	<i>CashPayment (UC - 31)</i>
<i>Restaurant's Database</i>	<i>Participating</i>	<i>To divide the payment of the meal accordingly among customers.</i>	<i>SplitPayment (UC - 32)</i>

3.b.iii. Waiter, Chef, and Busboy:

Table 7: Waiter, Chef, and Busboy as actors and their goals.

Actor	Type of Actor	Actor's Goal	Use Case Name
<i>Waiter</i>	<i>Initiating</i>	<i>To follow up on customer's satisfaction.</i>	<i>FollowUp (UC - 33)</i>
<i>Waiter</i>	<i>Initiating</i>	<i>To edit order if need be.</i>	<i>EditOrder (UC - 34)</i>
<i>Waiter</i>	<i>Participating</i>	<i>To deliver prepared meal.</i>	<i>DeliverMeal (UC - 35)</i>
<i>Chef</i>	<i>Participating</i>	<i>To prepare requested meal.</i>	<i>PrepareMeal (UC - 36)</i>
<i>Chef</i>	<i>Initiating</i>	<i>To cancel meal as requested.</i>	<i>CancelMeal (UC - 37)</i>
<i>Busboy</i>	<i>Participating</i>	<i>To clean table after customer departs.</i>	<i>CleanTable (UC - 38)</i>

3.c. Use Cases:

3.c.i. Casual Description

Table 8: Use cases related to Manager.

Use Case Number	Use Case Name	Description
<i>UC - 01</i>	<i>ShowTableAvailability</i>	<i>The manager view available tables in the dining area.</i>
<i>UC - 02</i>	<i>EditShifts</i>	<i>The manager controls employees' shifts.</i>
<i>UC - 03</i>	<i>Delete/Change/AddMenuItem</i>	<i>The manager changes the menu to their desire.</i>
<i>UC - 04</i>	<i>ShowSurveyData</i>	<i>The manager views customer feedback to make any changes.</i>
<i>UC - 05</i>	<i>ShowMostPopularItems, ShowHighestIncomeItems</i>	<i>The manager views the most successful items.</i>
<i>UC - 06</i>	<i>ShowExpenses</i>	<i>The manager views the expenses made by the restaurant.</i>
<i>UC - 07</i>	<i>SumWeek/Month/DayExpenses</i>	<i>The manager sums the total amount of expenses during a given period.</i>
<i>UC - 08</i>	<i>CheckInventory</i>	<i>The manager views inventory, and can order low stock items.</i>
<i>UC - 09</i>	<i>Remove/AddEmployee, ChangeEmployeeData</i>	<i>The manager adds or deletes any information pertaining to employees.</i>
<i>UC - 10</i>	<i>ShowActiveWaiters</i>	<i>The manager views which waiters are serving which customers.</i>
<i>UC - 11</i>	<i>ShowActiveChefs</i>	<i>The manager views which chefs prepared each customer's dish.</i>
<i>UC - 12</i>	<i>ShowActiveBusboys</i>	<i>The manager views which busboys are cleaning each table.</i>
<i>UC - 13</i>	<i>Change/AddMusic</i>	<i>The manager changes the music per the mood of the restaurant.</i>
<i>UC - 14</i>	<i>AccessPortal</i>	<i>The manager accesses the manager portal without any inconveniences.</i>

<i>UC – 15</i>	<i>PrintProtal</i>	<i>The manager prints out reports describing the restaurant's success.</i>
----------------	--------------------	--

Table 9: Use cases related to Customer

Use Case Number	Use Case Name	Description
<i>UC – 16</i>	<i>PreReservation</i>	<i>The customer either makes a pre-reservation for a specific date, time and table or checks in for the reservation already made.</i>
<i>UC – 17</i>	<i>NewReservation</i>	<i>The customer makes a new reservation for the desired table per the party size of the customer.</i>
<i>UC – 18</i>	<i>VerifyReservation</i>	<i>Restaurant's database verifies the reservation by checking through the list of reservations being placed in a queue per the date, time, and table #.</i>
<i>UC – 19</i>	<i>AddNewReservation</i>	<i>Restaurant's database adds new reservation to the queue after verifying the reservation.</i>
<i>UC – 20</i>	<i>NotifyCustomer/Host</i>	<i>The restaurant's database notifies the customer for the availability of the table and guides them to their table to be seated.</i>
<i>UC – 21</i>	<i>AddMenuItem</i>	<i>The customer adds an item to the list of the items to be placed for the order.</i>
<i>UC – 22</i>	<i>RemoveMenuItem</i>	<i>The customer removes an item added to the list of items in the order.</i>
<i>UC – 23</i>	<i>OrderItems</i>	<i>The customer places an order after reviewing the items in the entire order and confirms the order.</i>
<i>UC – 24</i>	<i>CancelOrder</i>	<i>The customer cancels an order while in the process of placing it or within a certain time after placing it.</i>
<i>UC – 25</i>	<i>FoodWaitTime</i>	<i>The restaurant's database calculates an estimate time for when the customer's order will be ready.</i>
<i>UC – 26</i>	<i>Entertainment</i>	<i>The customer uses the internet or play games while their food is being prepared by the kitchen.</i>
<i>UC – 27</i>	<i>FoodStatus</i>	<i>The customer checks the stage/process/status of the food as it is being prepared.</i>
<i>UC – 28</i>	<i>SpecialOffers/TakeSurvey</i>	<i>The customer gives feedback or takes a survey to receive a special bonus/offer.</i>

<i>UC – 29</i>	<i>AdjustBill</i>	<i>The restaurant's database adjusts the price of the meals per any special bonus/offer.</i>
<i>UC – 30</i>	<i>CardPayment</i>	<i>The customer makes a payment with their card through the system.</i>
<i>UC – 31</i>	<i>CashPayment</i>	<i>The waiter helps customer make payment through cash.</i>
<i>UC – 32</i>	<i>SplitPayment</i>	<i>The restaurant's database divides the bill.</i>

Table 10: Use cases related to Waiter, Host, Chef, and Busboy

Use Case Number	Use Case Name	Description
<i>UC – 33</i>	<i>FollowUp</i>	<i>The waiter follow ups on the customer for their assistance.</i>
<i>UC – 34</i>	<i>EditOrder</i>	<i>The waiter assist the customer in editing the order.</i>
<i>UC – 35</i>	<i>DeliverMeal</i>	<i>The waiter delivers the meal to the customer when it is ready.</i>
<i>UC – 36</i>	<i>PrepareMeal</i>	<i>The chef prepares the meal according to the orders in the queue.</i>
<i>UC – 37</i>	<i>CancelMeal</i>	<i>The chef cancels the order as requested.</i>
<i>UC – 38</i>	<i>CleanTable</i>	<i>The busboy cleans the table when the customer has done payment and leaves the table.</i>

3.c.ii. Use Case Diagram

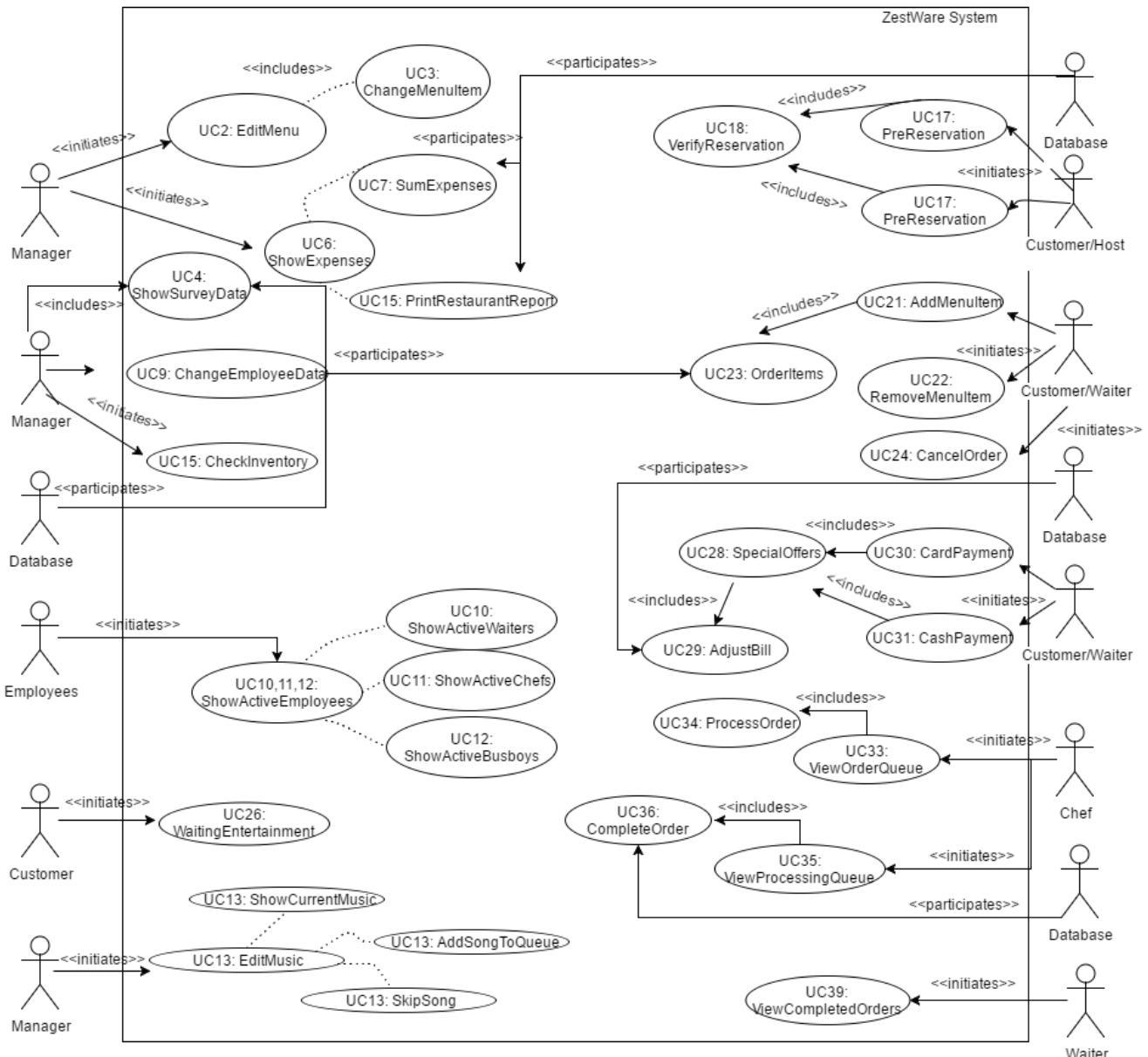


Figure 6: Use Case Diagram.

3.c.iii. Traceability Matrix

Priority: Low – PW = 1, Medium – PW = 2, High – PW = 3

Table 11: Traceability Matrix for Manager.

REQ		PW	UC														
			01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
M	01	2	X														
	02	1		X													
	03	3			X												
	04	3				X											
	05	2					X										
	06	2						X	X								
	07	3								X							
	08	1									X						
	09	2										X					
	10	2											X				
	11	3													X		
	12	2												X			
	13	1													X		
	14	1														X	
Max PW			2	1	3	3	2	2	2	3	1	2	2	2	1	3	1
Total PW			2	1	3	3	2	2	2	3	1	2	2	2	1	3	1

Table 12: Traceability Matrix for Customer.

REQ		PW	UC															
			16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
C	01	3	X		X													
	02	3	X		X													
	03	3		X		X												
	04	3		X			X											
	05	2		X			X											
	06	1	X	X	X													
	07	2		X				X										
	08	1		X				X										
	09	3							X	X	X							
	10	3						X	X	X								
	11	3							X	X	X							
	12	2										X			X			
	13	2										X			X			
	14	2									X				X			X
	15	3														X		
	16	3															X	
	17	3																X
	18	1														X		
	19	2										X						
	20	2													X	X		
Max PW			3	3	3	3	2	3	3	3	3	2	2	2	2	2	3	3
Total PW			7	12	7	8	3	6	9	9	5	4	2	6	2	3	3	5

Table 13: Traceability Matrix for Kitchen.

REQ		PW	UC					
			33	34	35	36	37	38
K	01	2		X				
	02	1	X					
	03	2		X				
	04	3			X			
	05	1	X					
	06	2						X
	07	3				X		
	08	1					X	
	09	2						X
Max PW			1	2	3	3	1	2
Total PW			2	4	3	3	1	4

3.c.iv. Full-Dressed Description

Use Case UC - 02: EditShifts

Related Requirements: From user stories: M-2

Initiating Actor: Manager

Actor's Goal: To easily maintain and edit the shifts and information of employees in the restaurant's database.

Participating Actors: Database

Preconditions: The manager is authorized personnel that can access the shift editing portal. There is a database with existing information of the restaurant.

Postconditions: The shifts or information of employees shall be correctly and immediately updated in the database.

Flow of Events for Main Success Scenario:

← 1. **System** prompts actor for username and password or actor uses facial recognition software for access to the portal.

→ 2. **Manager** provides necessary credentials or displays face for reading.

← 3. **System** verifies the actor and prompts the actor if they would like to (a) view current timetable, or (b) edit current timetable.

← 3a. **System** displays current timetable for review by actor.

← 3b. **System** displays timetable with options to (a) delete, (b) add, or (c) change the employees in the shift table.

→ 4. The **manager** chooses one of the options.

→ 4a. For delete, the **manager** chooses a shift and chooses an employee's name to remove from the timetable of the day.

← 4a. The **system** removes the employee from the shift, updates the timetable and the database, and confirms the action.

→ 4b. For add, the **manager** chooses a shift and is given a list of sorted employee names to choose for the shift.

← 4b. The **system** adds the chosen name to the shift, updates the database, and confirms the action.

→ 4c. For changing the table, the **manager** chooses an employee name and can replace the current employee with another employee's name in the time table.

← **4c.** *The system replaces the employee, updates the database, and confirms the action.*

→ **5.** *The manager is prompted to either continue editing, print report, or exit portal.*

Flow of events for Extensions (Alternate Scenarios):

← **1.** *System (a) detects incorrect credentials, (b) takes down attempt, and (c) prompts for actor to try once more for access.*

→ **2.** *Manager provides credentials or face for identification once more.*

Use Case UC - 08: CheckInventory

Related Requirements: From user stories: M-7, M-11

Initiating Actor: Manager

Actor's Goal: To access the inventory and see which items are in stock and which need to be ordered. To be able to easily and quickly order any quantity of low stock items. To update the inventory database when new orders come in.

Participating Actors: Database

Preconditions: The manager is authorized personnel that can access the inventory portal. There is a database with existing information of the restaurant's inventory.

Postconditions: The items ordered will be purchased and the database updated.

Flow of Events for Main Success Scenario:

← 1. **System** prompts actor for username and password or actor uses facial recognition software for access to the inventory portal.

→ 2. **Manager** provides necessary credentials or displays face for reading.

← 3. **System** verifies the actor and prompts the actor if they would like to (a) view the inventory, (b) see low items first, or (c) order food items.

← 3a. **System** displays current inventory for review by actor.

← 3b. **System** displays inventory with options to (a) delete, (b) add, or (c) order the particular food item.

→ 4. **The manager** chooses one of the options.

→ 4a. For delete, the **manager** chooses an item to remove entirely from the inventory database.

← 4a. The **system** removes the item from the list, updates the list and the database, and confirms the action.

→ 4b. For add, the **manager** enters a food item name and is then prompted to order a certain amount of the food item.

← 4b. The **system** adds the chosen item to the inventory and sets up an order list for the manager to confirm afterwards.

→ 4c. For ordering items, the **manager** chooses an item and is prompted to select an amount.

← 4c. The **system** orders the food items, updates the database, and confirms with the actor.

Flow of events for Extensions (Alternate Scenarios):

← **1. System** (a) detects incorrect credentials, (b) takes down attempt, and (c) prompts for actor to try once more for access.

→ **2. Manager** provides credentials or face for identification once more.

Use Case UC - 16: PreReservation

Related Requirements: C-01, C-02, C-04, C-06

Initiating Actor: Customer, Host

Actor's Goal: To check-in to a preservation and be seated at their respective table.

Participating Actors: Database

Preconditions: The customer has previously made a reservation either through the phone or through the online form.

Postconditions: The reservation should be removed from the database, and the table.

Flow of Events for Main Success Scenario:

- 1. **System (CWS)** greets the customer with a simple interface allowing them to select pre-reservation, or new-reservation.
- ← 2. **Customer or Host** select the pre-reservation button on the (CWS).
- 2a. **Customer or Host** enters the name and party size of the reservation into the (CWS).
- 3. **Restaurant database** verifies there is a reservation under the given name.
- 4. **CWS** says welcome to Zest-Ware [Customer's Name]!", and the customer is guided to their table.

Flow of events for Extensions (Alternate Scenarios):

- 4a. System is not able to verify the given name for a previously made reservation.
- ← 1. **CWS** (a) detects incorrect credentials, (b) prompts the actor to enter credentials one more time.
- 2. **Customer** enters name and party size to create a new reservation.

Use Case UC - 17: NewReservation

Related Requirements: C-03, C-04, C-05, C-06, C-07, C-08.

Initiating Actor: Customer or Host.

Actor's Goal: To get seated at a table.

Participating Actors: Customer Welcome Screen (CWS), Floor Mapping Screen (FMS).

Preconditions: CWS displays option for new reservation.

Postconditions: Customer gets seated at available/desired table.

Flow of Events for Main Success Scenario:

- 1. **Customer/host** selects the option "New Reservation".
- ← 2. **System** prompts for customer's name and party size.
- 3. **Customer/host** enters name and party size.
- ← 4. **System** (a) gets the available table from the FMS, (b) displays the floor map and available table. For example, "Table #X available" and highlights the available table.
- 5. **Customer/host** selects "Confirm Table Reservation".
- ← 6. **System** (a) displays "Table Reservation Confirmed, you may now be seated at table #X," (b) signals to floor mapping system to change status of table #X to "Occupied".

Flow of Events for Extensions (Alternate Scenarios):

- 4a. System senses that the table is not available, so table cannot be assigned
- ← 1. **System** (a) displays that the table is not available and the wait time for the next available, (b) displays the floor map, (c) prompts user to cancel reservations or wait for available table.
 - 2. **Customer/host** selects the wait option.
 - ← 3. **System** (a) displays message to wait in waiting area, (b) signals floor mapping system for next customer in queue to be seated.

Use Case UC - 23: OrderItems

Related User Stories: C-09, C-10, C-11.

Initiating Actor: Customer.

Actor's Goal: To place an order.

Participating Actor: Customer Table Screen (CTS), Kitchen Staff Screen (KSS).

Preconditions: The CTS displays the menu of all the items.

Postconditions: The order is placed and is sent to the KSS.

Flow of Events for Main Success Scenario:

- 1. **Customer** selects the option "Order Items" on the CTS.
- ← 2. **System** displays the menu items with categories.
- 3. **Customer** selects desired item from the icons.
- ← 4. **System** (a) lists the customer's desired items, (b) signals customer to confirm order.
- 5. **Customer** selects "Confirm Order".
- ← 6. **System** (a) signals the order placed to KSS, (b) displays estimated wait time for order. For example, "Estimated wait time for Table #X is: Y minutes".

Flow of Events for Extensions (Alternate Scenarios):

- 5. Customer wants to make changes (add/remove item) to order.
 - ← 1. **System** gives "Make changes to the order" option.
 - 2. **Customer** selects the option "Make changes to the order".
 - ← 3. **System** displays menu items with categories.
 - 4. **Customer** (a) adds/removes items from menu, (b) confirms order.

Use Case UC – 30: CardPayment

Related User Stories: C-15.

Initiating Actor: Customer.

Actor's Goal: To get bill and make a payment.

Participating Actor: Customer Table Screen(CTS), Bitcoin, Waiter's Screen.

Preconditions: The order has been placed and delivered.

Postconditions: The payment is successful.

Flow of Events for Main Success Scenario:

- 1. **Customer** selects "Payment" option.
- ← 2. **System** prompts user for "Cash Payment," "Card Payment," "Bitcoin," or "Split Payment".
- 3. **Customer** selects "Card Payment".
- ← 4. **System** prompts the user to swipe card or insert chip.
- 5. **Customer** inserts or swipes card via CTS.
- ← 6. **System** (a) processes payment, (b) confirms payment, (c) prints bill/receipt.

Flow of Events for Extensions (Alternate Scenarios):

6. System is unable to process payment. For example, denial of card.

- ← 1. **System** (a) displays "Unsuccessful payment," (b) signals waiter's screen for assistance.

Use Case : Checkout

Related User Stories: K – 02, K – 05, K – 06, K – 09

Initiating Actor: Customer, Waiter, Database, Busboy.

Actor's Goal: Ensure customer pays, table is cleaned and available for the next arriving customer.

Participating Actor: Customer Table Screen(CTS), Bitcoin, Waiter's Screen.

Preconditions: Customer is satisfied, Customer finished meal Customer pays before departing.

Postconditions: Update database, request busboy.

Flow of Events for Main Success Scenario:

→ **1.** Ensure Customer Satisfaction.

← **2.** Customer Requests Bill.

→ **3.** Returns the Bill.

→ **4.** Ensures Finished Eating.

→ **5.** Ensures Customer Pays.

→ **6.** Request Busboy

← **7.** Busboy cleans the table.

← **8.** Makes seat as available.

→ **9.** Update Database.

3.d. System Sequence Diagrams:

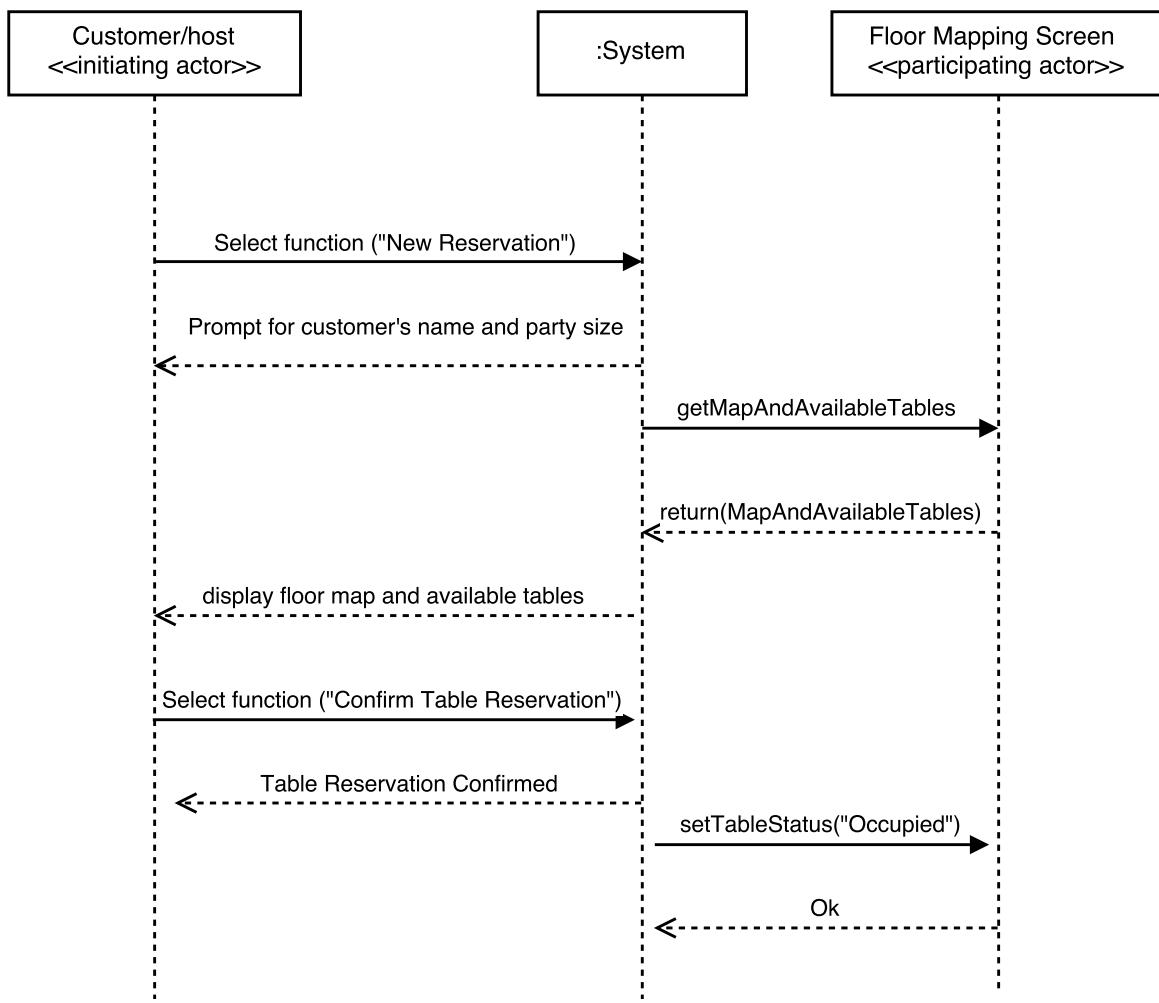


Figure 7: UC- 17 (NewReservation) Main Success Scenario

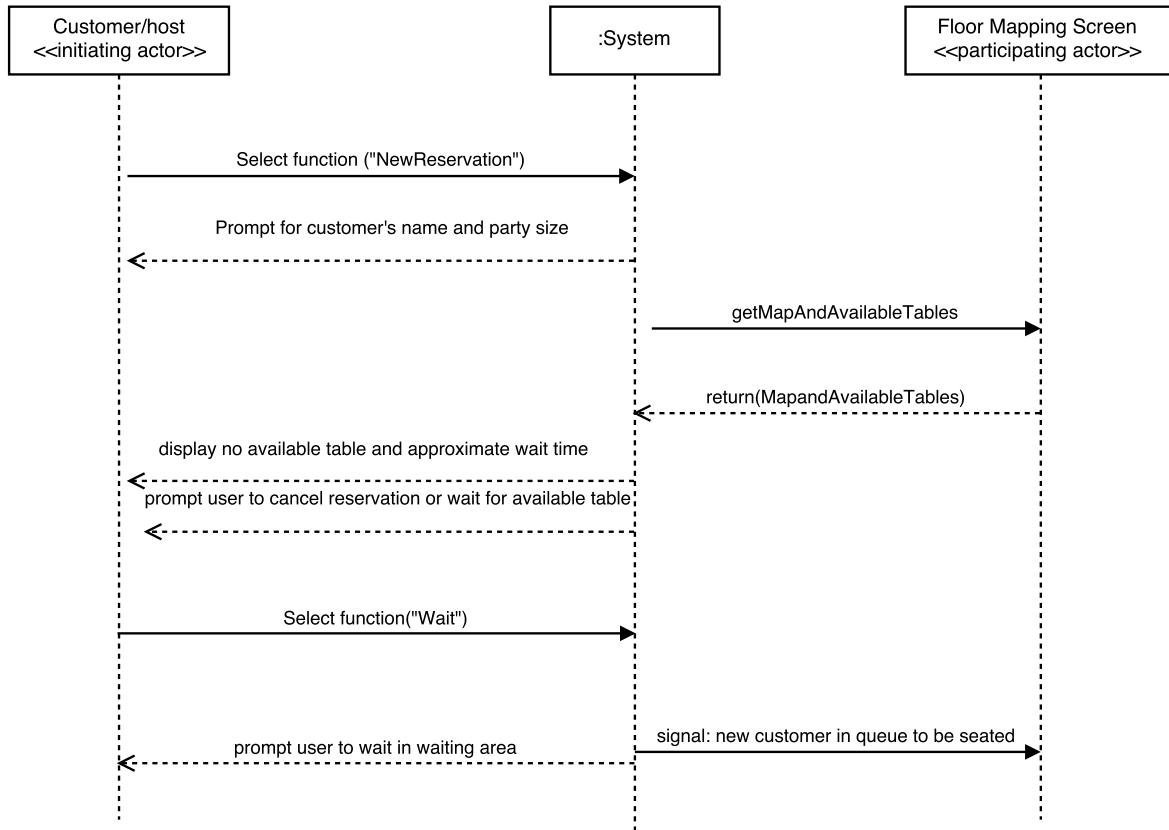


Figure 8: UC - 17 (NewReservation) Alternate Scenario

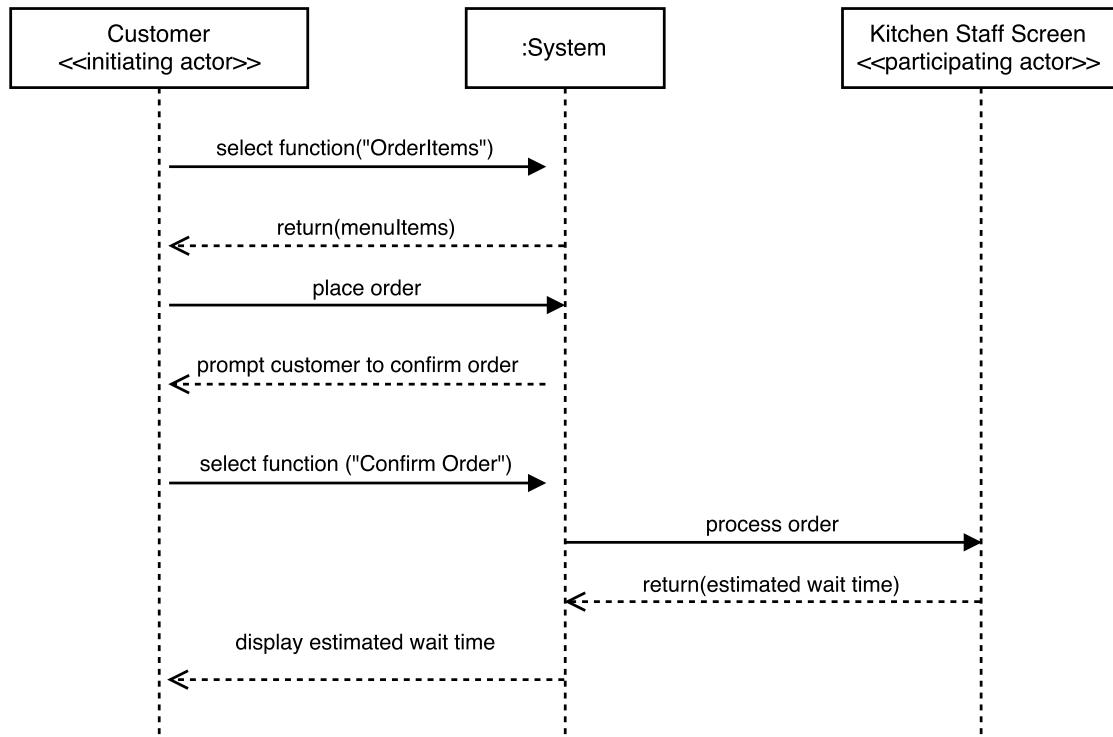


Figure 9: UC - 23 (OrderItems) Main Success Scenario

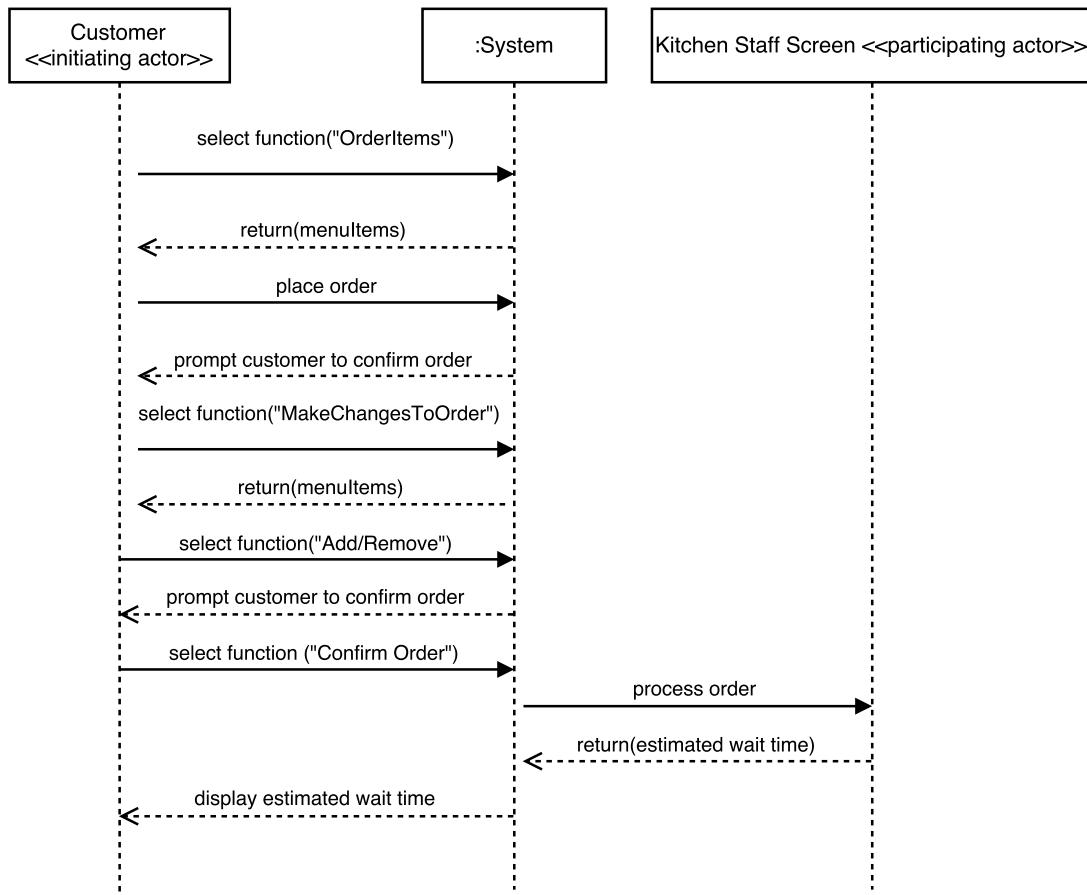


Figure 10: UC - 23 (OrderItems) Alternate Scenario

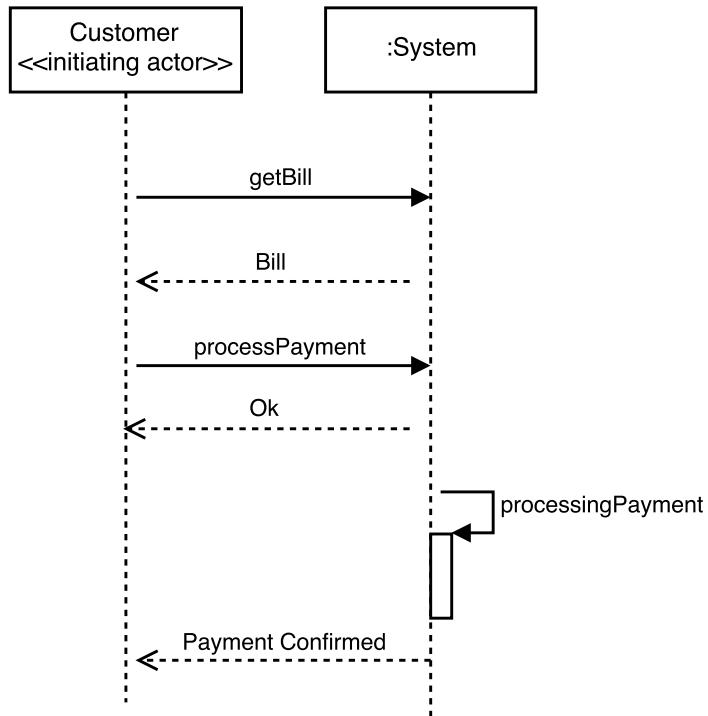


Figure 11: UC - 30 (CardPayment) Main Success Scenario

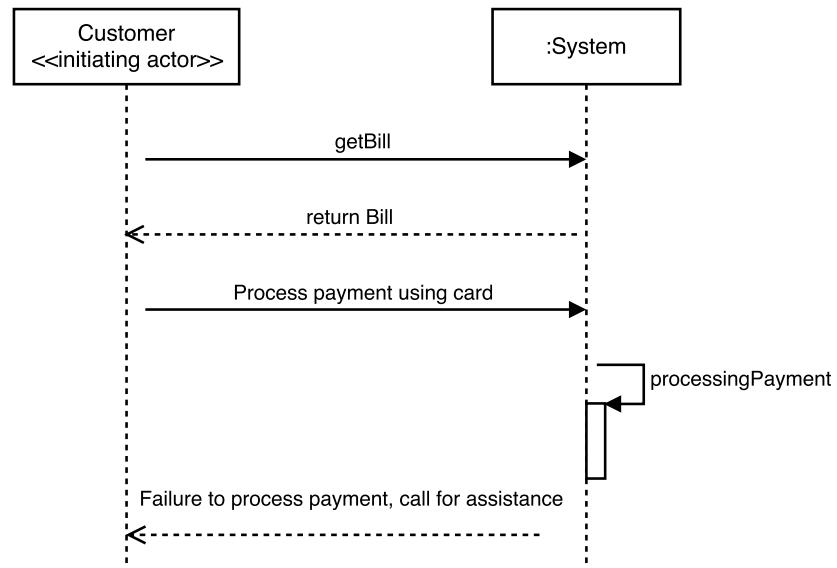


Figure 12: UC - 30 (CardPayment) Alternate Scenario

Use Case 9 : Edit Shifts

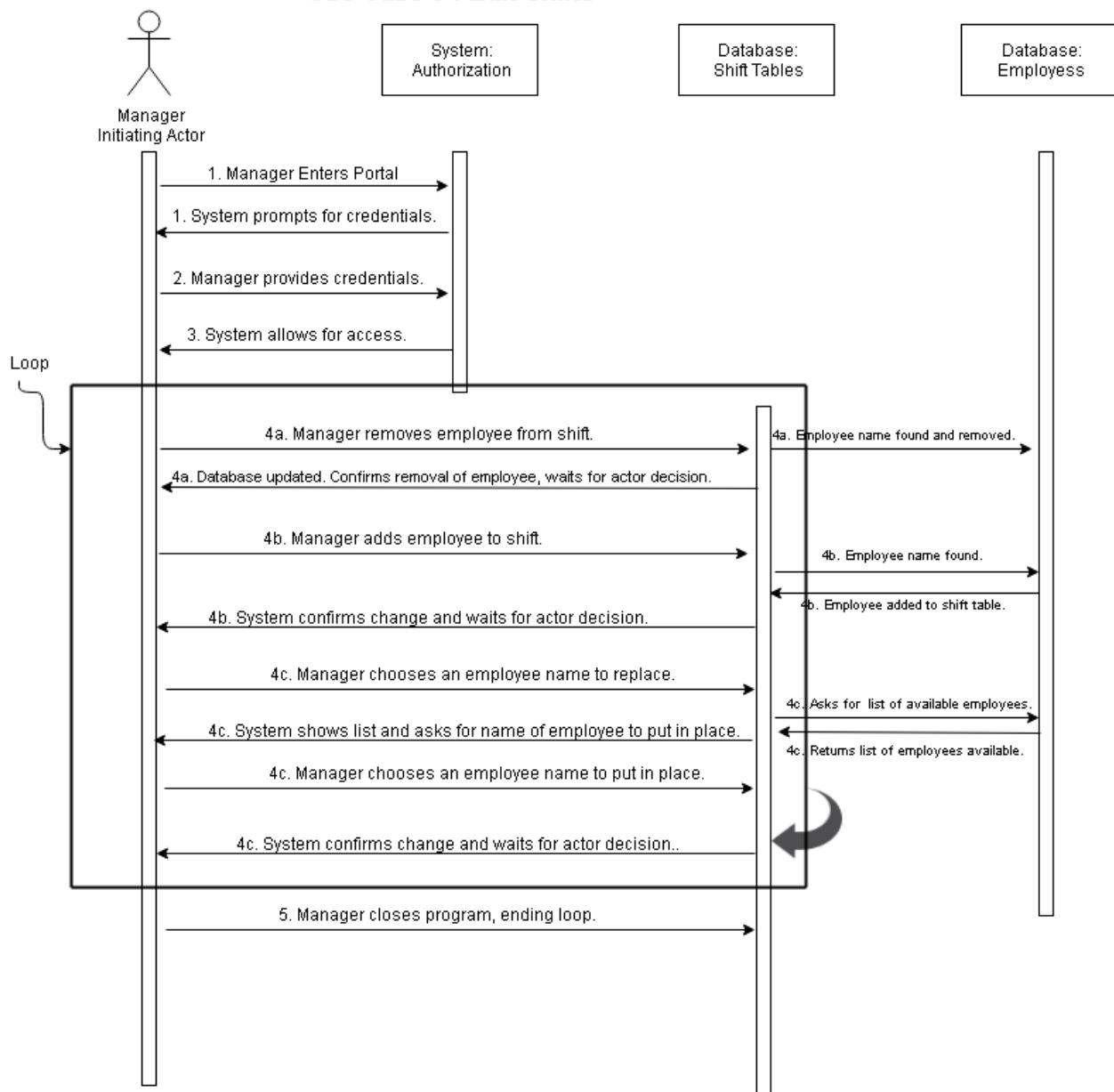


Figure 13: UC - 09 (EditShifts) Main Success Scenario

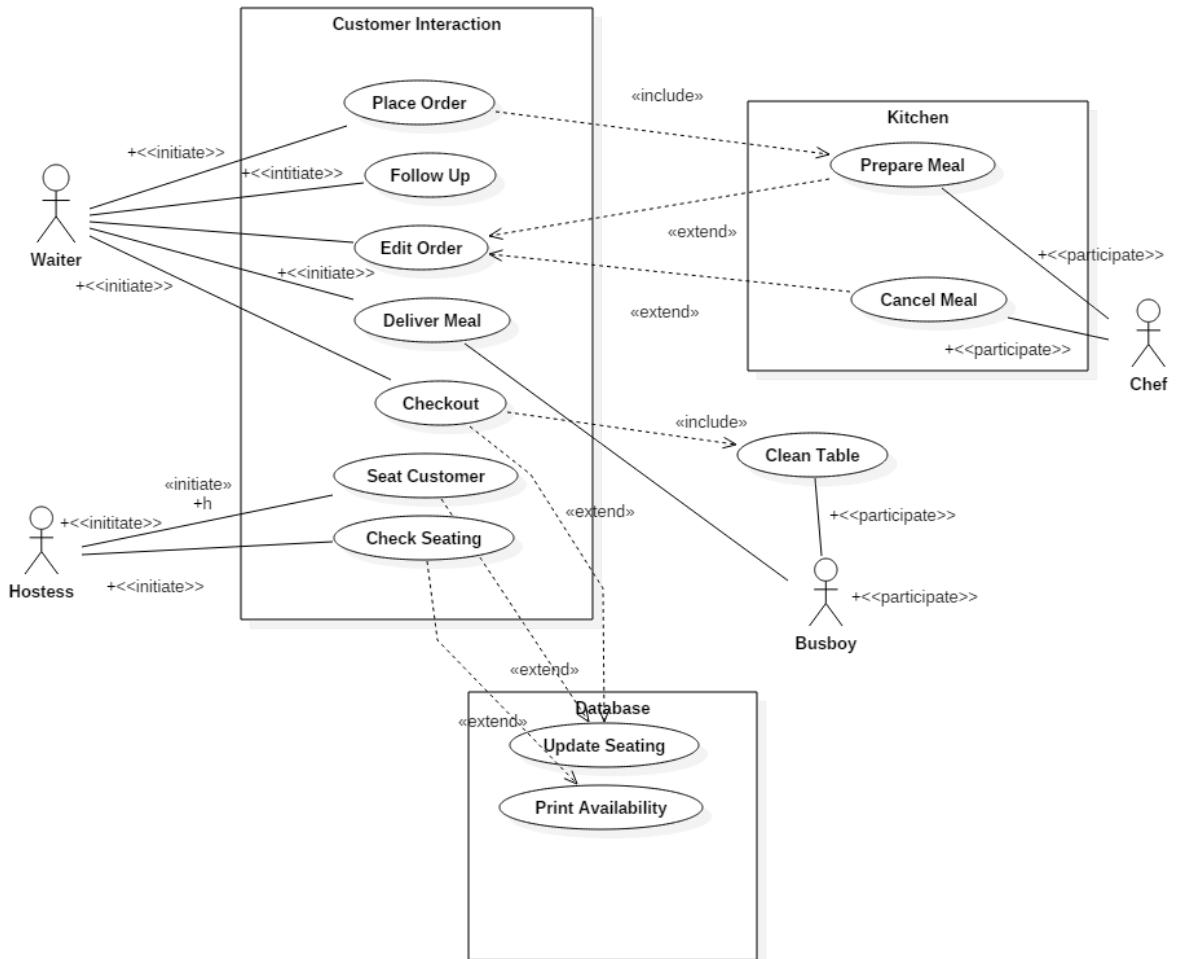


Figure 14: Kitchen Use Case

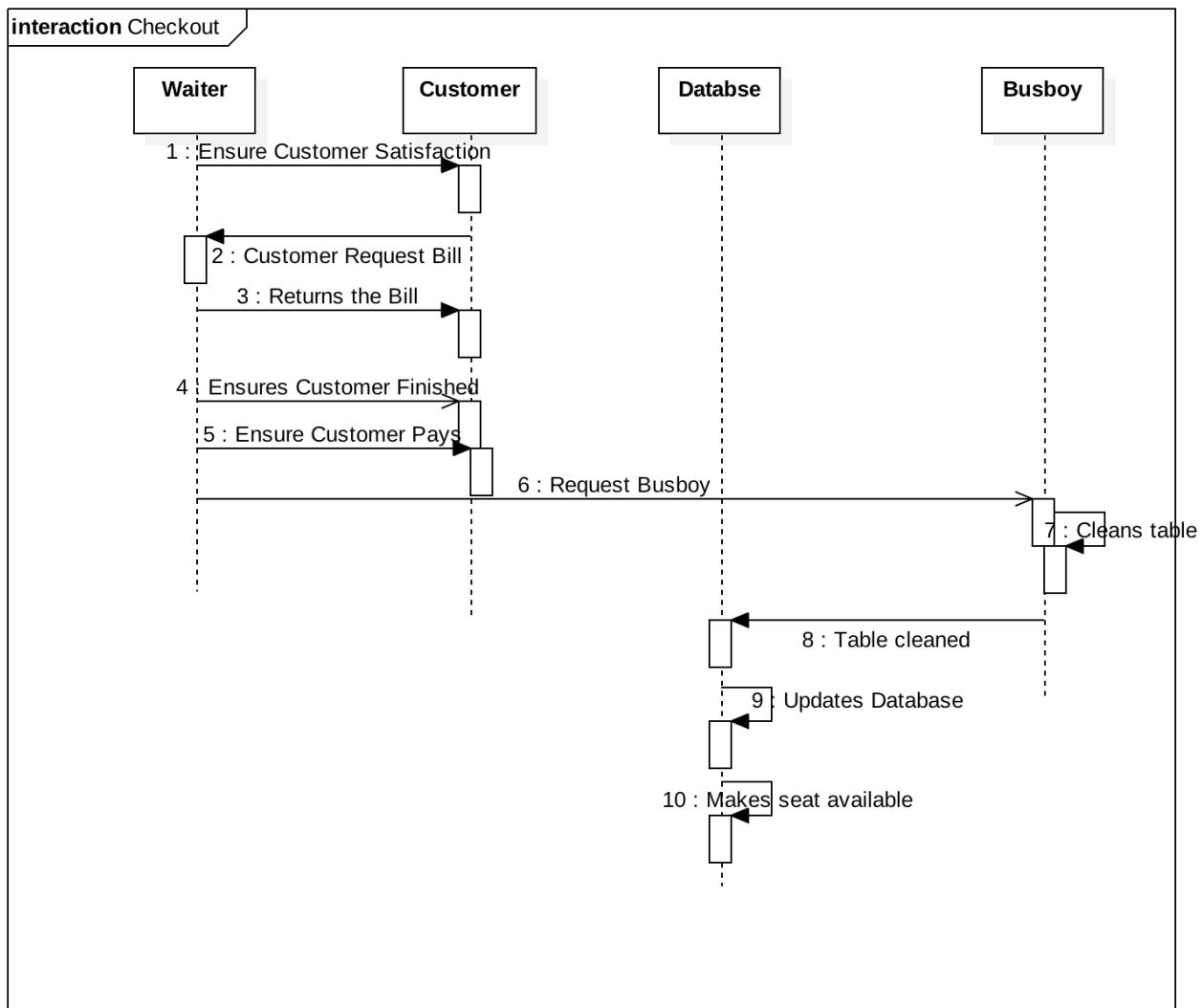
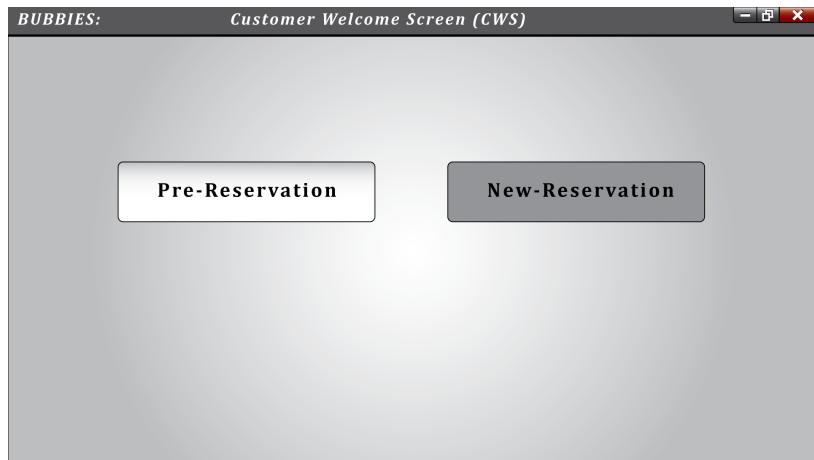


Figure 15: Use Case Checkout

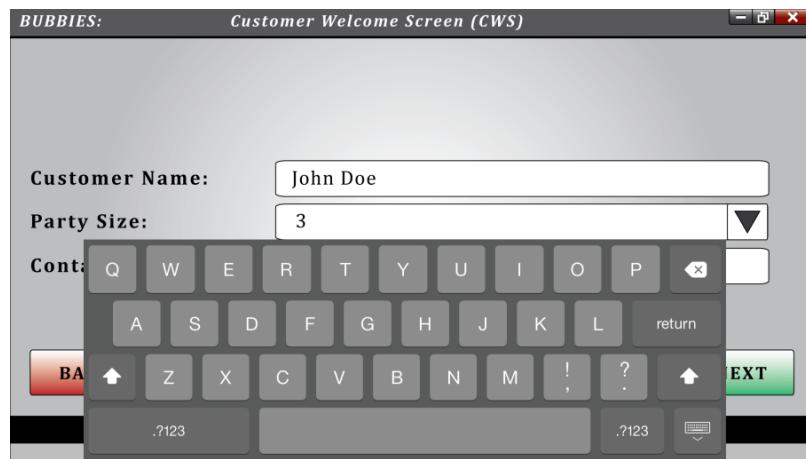
Part 2.4. User Interface Specification

4.a. Preliminary Design and User Effort Estimation

New-Reservation



*Main Screen of the CWS: The Customer/Host can tap on either
Pre-Reservation or New-Reservation*



After tapping the New-Reservation button/icon, the system prompts the user (either customer or host) for the customer's name, and their party size.

The customer's name can easily be entered by tapping on the blank white field/column in front of the "Customer Name:" and then an on-screen keyboard will appear through which the user will enter the name.

BUBBIES: Customer Welcome Screen (CWS)

Customer Name: John Doe

Party Size: (Select size)

Contact # (optional):

1	▼
2	▲
3	≡
4	▼
5	▲
6	≡

BUBBIES: Customer Welcome Screen (CWS)

Customer Name: John Doe

Party Size: (Select size)

Contact # (optional):

1	▼
2	▲
3	≡
4	▼
5	▲
6	≡

The party size is entered by tapping on the drop down menu and then selecting the size, if the size is larger the drop down menu can be scrolled down for higher sizes.

BUBBIES: Customer Welcome Screen (CWS)

Customer Name: John Doe

Party Size: 3

Contact # (optional): (____)-____-_____

BACK **NEXT**

NEW RESERVATION

BUBBIES: Customer Welcome Screen (CWS)

Customer Name: John Doe

Party Size: 3

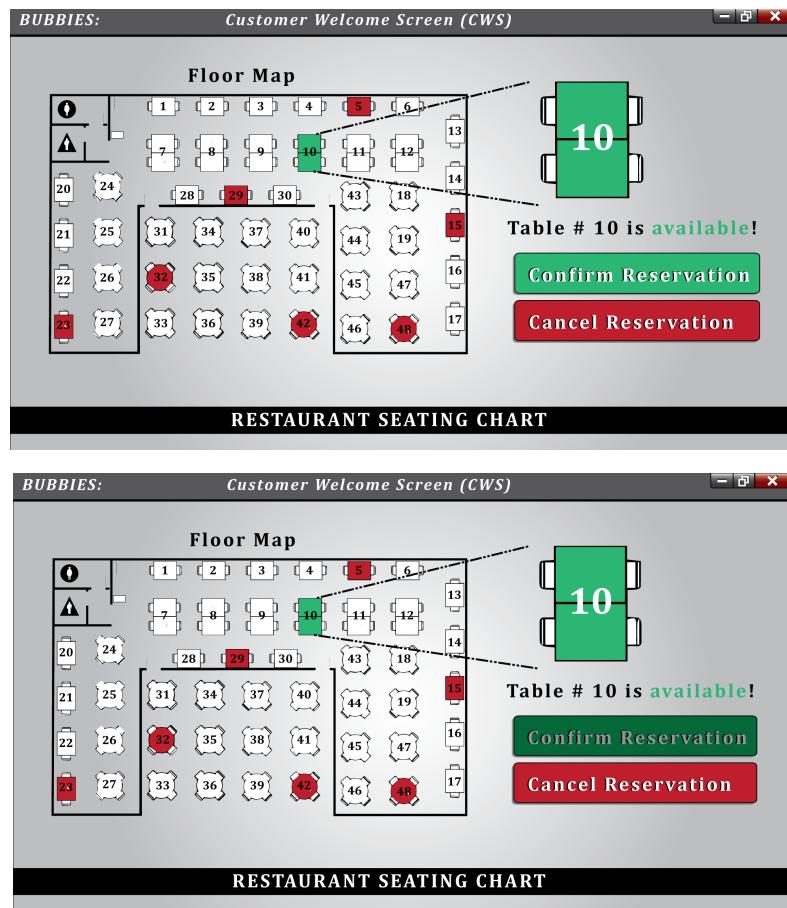
Contact # (optional): (____)-____-_____

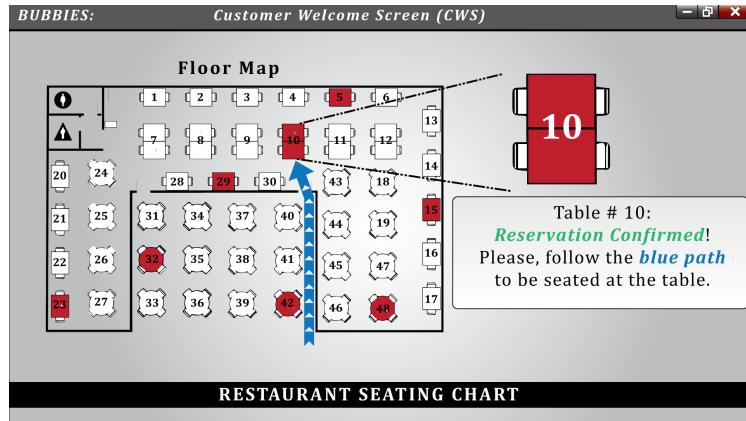
BACK **NEXT**

NEW RESERVATION

The contact number is optional so the user does not need to necessarily input the contact the number unless they want to.

The user taps "NEXT" after the name and size is entered and the system will check for the available tables according to the party size.





The system would change the status of the table to occupied, displays “reservation confirmed” message, and guides the customer to their table by displaying a path from where the user’s current location in the restaurant to their reserved table.

Approximately 5 touches/taps (excluding the tapping of customer’s name on the on-screen keyboard) to make new-reservation.

Other User Interfaces for Placing Order, Payments, Managerial, and Kitchen.

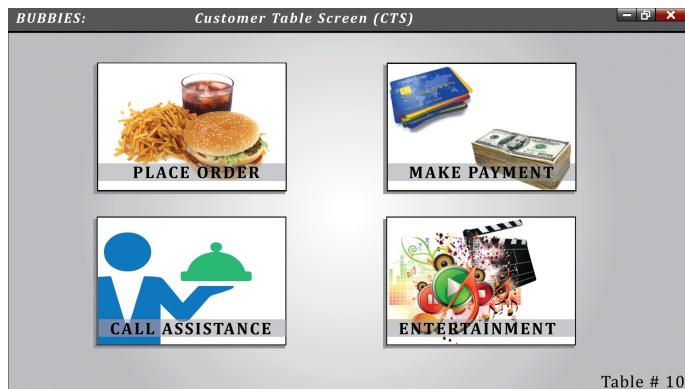


Table # 10

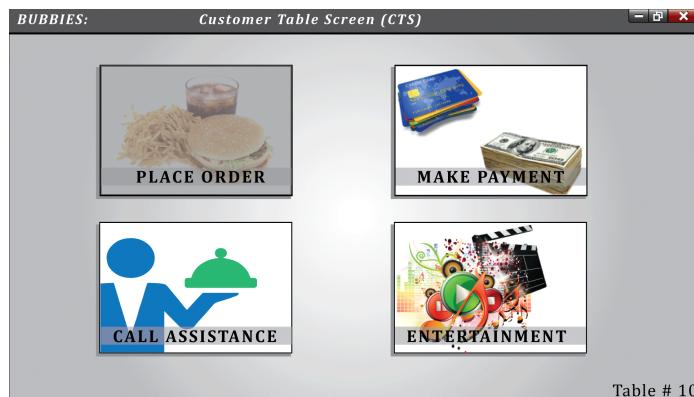
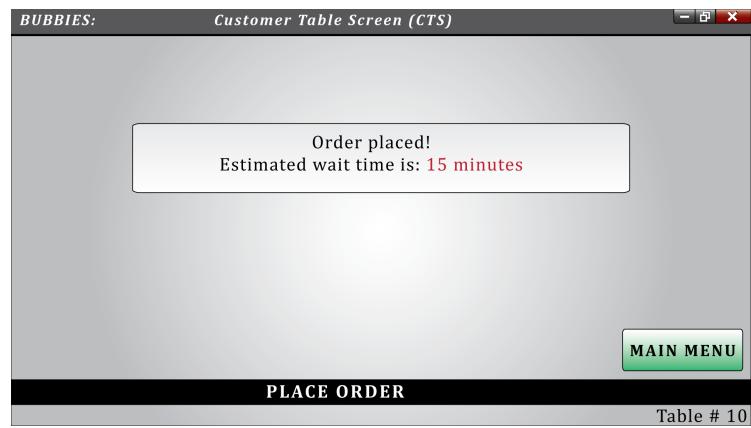
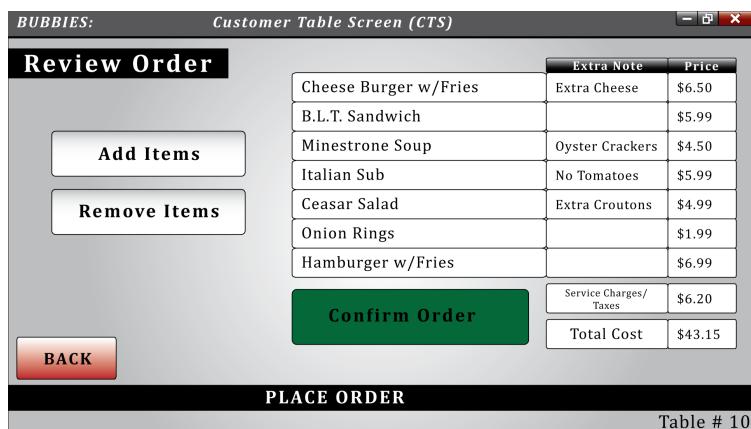
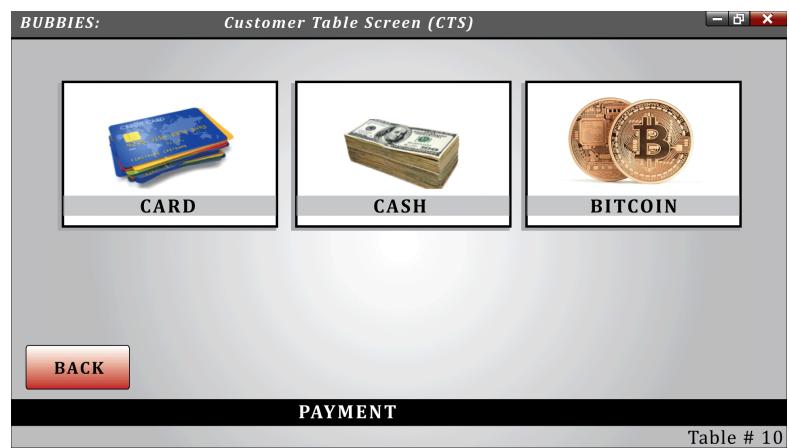
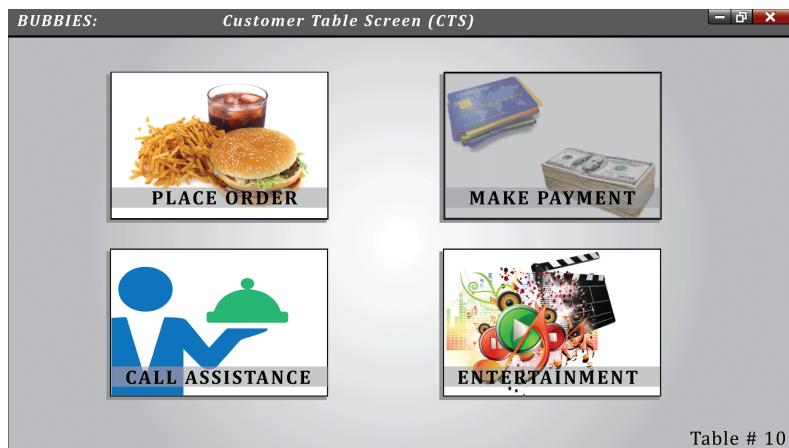


Table # 10



Table # 10



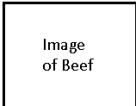


Item Name: Ground Beef	<input type="button" value="Order Item"/>	<input type="button" value="Delete Item"/>
	Notes	
Item Status: !!! LOW	Menu Item Ingredient is Used In	
Item Quantity: 5 oz.	<ul style="list-style-type: none">• Hamburger• Meatloaf• Meatballs	
<input type="button" value="Back to List"/>		

Item Name: Ground Beef	From Supplier: XXXXXXX	
	Enter Qt.	<input type="text"/>
Item Status: !!! LOW	Price Per Oz.	<input type="text"/>
Item Quantity: 5 oz.	Total	<input type="text"/>
<input type="button" value="Back to List"/>		

Item Name: Ground Beef	From Supplier: XXXXXXX	
	Enter Qt.	<input type="text" value="500 oz."/>
Item Status: !!! LOW	Price Per Oz.	<input type="text" value="\$1.00"/>
Item Quantity: 5 oz.	Total	<input type="text" value="\$500.00"/>
<input type="button" value="Back to List"/>		

Item Name: Ground Beef	From Supplier: XXXXXXX
	Enter Qt. 500 oz.
	Price Per Oz. \$1.00
	Total \$500.00
Item Status: !!! LOW	Purchase
Item Quantity: 5 oz.	
	

Item Name: Ground Beef	From Supplier: XXXXXXX
	Enter Qt. 500 oz.
	Price Per Oz. \$1.00
	Total \$500.00
Item Status: !!! LOW	Purchase
Item Quantity: 5 oz.	
	

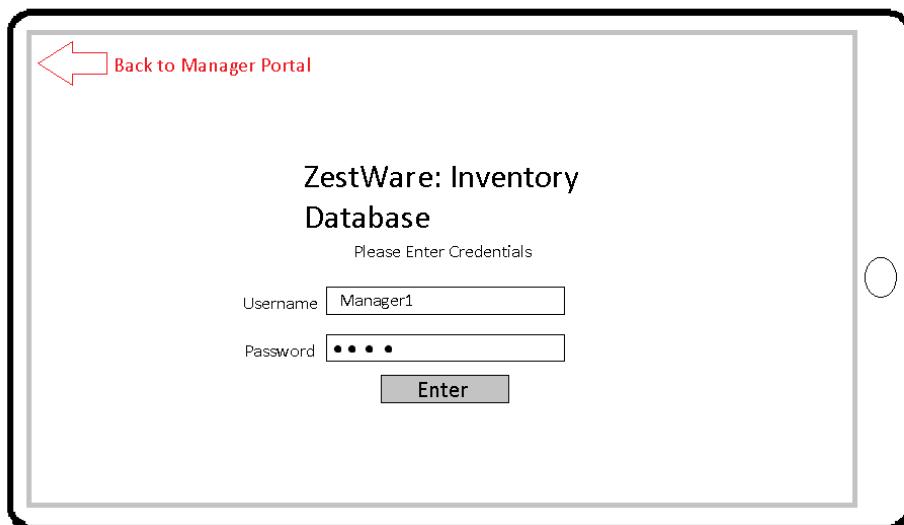
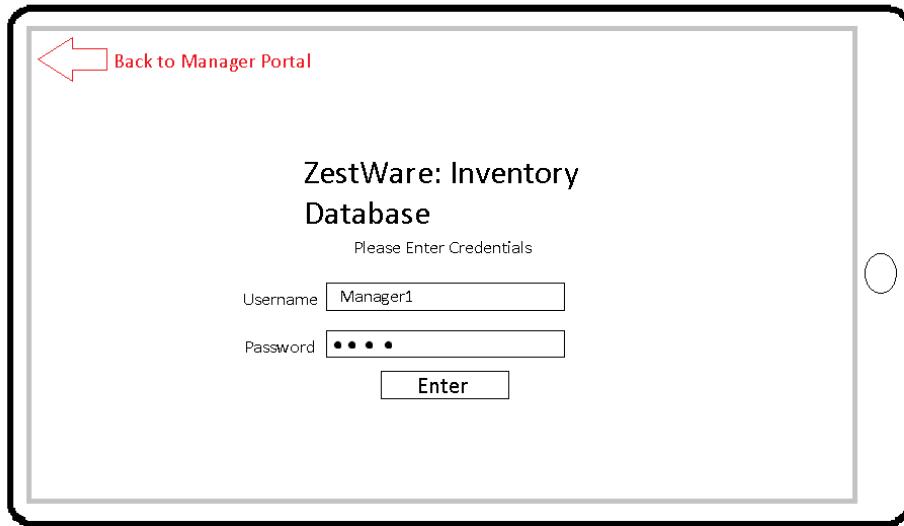
Item Name: Ground Beef	From Supplier: XXXXXXX
	Enter Qt. 500 oz.
	Purchase 500 oz. of Ground Beef for \$500.00 from Supplier XXXXXX?
Item Status: !	<input type="button" value="Yes"/> <input type="button" value="No"/>
Item Quantity: 5 oz.	
	

Item Name: Ground Beef	From Supplier: XXXXXXX
	Enter Qt. 500 oz.
Purchase 500 oz. of Ground Beef for \$500.00 from Supplier XXXXXXX?	
Item Status: !!! LOW	<input type="button" value="Yes"/> <input type="button" value="No"/>
Item Quantity: 5 oz.	
	

Item Name: Ground Beef	From Supplier: XXXXXXX
	Enter Qt. 500 oz.
	Price Per Oz. \$1.00
	Total \$500.00
Item Status: !!! LOW	Purchase
Item Quantity: 5 oz.	Order Status: Ordered
	

Item Name	Item Quantity		Item Status	Note
Ground Beef	5 oz.	Order Item	!!! Low	
Sesame Seed Buns	30 pcs.	Order Item	!! Moderately Low	
Lettuce Heads	50 pcs.	Order Item	In Stock	
Tomatoes	50 pcs.	Order Item	In Stock	
Mayonaisse	20 oz.	Order Item	!! Moderately Low	
Ketchup	20 oz.	Order Item	!! Moderately Low	
Mustard	1 oz.	Order Item	!!! Low	
American Cheese	120 pcs.	Order Item	In Stock	

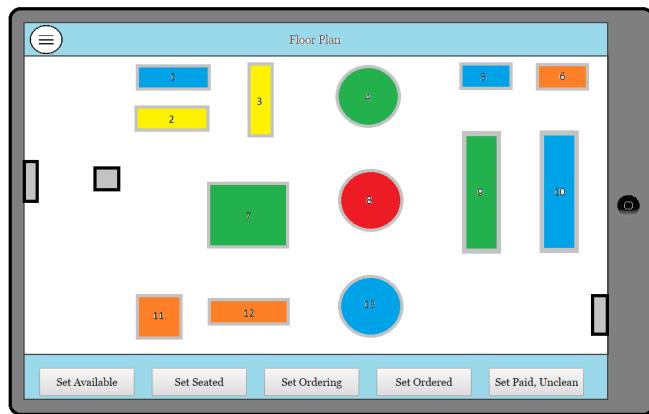
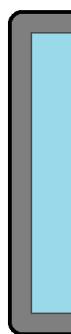
Item Name	Item Quantity		Item Status	Note
Ground Beef	5 oz.	Order Item	!!! Low	
Sesame Seed Buns	30 pcs.	Order Item	!! Moderately Low	
Lettuce Heads	50 pcs.	Order Item	In Stock	
Tomatoes	50 pcs.	Order Item	In Stock	
Mayonaisse	20 oz.	Order Item	!! Moderately Low	
Ketchup	20 oz.	Order Item	!! Moderately Low	
Mustard	1 oz.	Order Item	!!! Low	
American Cheese	120 pcs.	Order Item	In Stock	



The interface shows a list of food items on the left and a detailed view of the queue on the right.

	Item	Special Notes	Table #	Time in Queue
Cheese Fries	Cheese Burger Fries	Extra Cheese	Table 3	7 Minutes
B.L.T. S	B.L.T. Sandwich		Table 2	5 Minutes
Minestr	Minestrone Soup, Side Order			2 Minutes
Italian S	Italian Sub			2 Minutes
Caesar S	Caesar Salad			1 Minute
Onion R	Onion Rings, Side Order		Table 3	30 Seconds
Hamburg Fries	Hamburger		Table 6	Just Now

Buttons at the bottom of the queue view: PREPARED, CANCEL, EDIT.



Item	Special Notes	Table #	Time Finished	Assigned Server
Cheese Burger Fries	Fries Out ASAP, Don't Wait for Burger	Table 3	3 minutes ago	Ashley
B.L.T. Sandwich		Table 2	2 minutes ago	John
Minestrone Soup, Side Order			1 minute ago	Jane
Italian Sub			1 minute ago	John
Caesar Salad			1 minute ago	Ashley
Onion Rings, Side Order		Table 6	30 seconds ago	Ashley
Hamburger Fries		Table 3	just now	Jane

(≡)	12	1
SUN		
MON		
TUES		
WED		
THURS		
FRI		
SAT		

NAME: Matt Dam
Hours this Week: :

Item	Special Notes	Table #	Time Finished	Assigned Server
Cheese Burger Fries	Fries Out ASAP, Don't Wait for Burger	Table 3	3 minutes ago	Ashley
B.L.T. Sandwich		Table 2	2 minutes ago	John
Minestrone Soup, Side Order	Oyster Crackers	Table 4	2 minutes ago	Jane
Italian Sub	No Tomatoes	Table 7	2 minutes ago	John
Caesar Salad	Extra Croutons	Table 7	1 minute ago	Ashley
Onion Rings, Side Order		Table 6	30 seconds ago	Ashley
Hamburger Fries		Table 3	just now	Jane

Seat Customer

1. NAVIGATION: total 1 tap

a. Tap “Floor Plan” button on the home screen

2. DATA ENTRY: total 2 taps

a. Tap on an available table

b. Tap on “Set Seated” button

Prepare Meal

1. NAVIGATION: total 2 taps

a. Tap “Chef Screen” button on the home screen

b. Tap anywhere on the order in the queue that's been prepared

2. DATA ENTRY: total 1 tap

a. Tap “Prepared” button to send it to the server's queue

Deliver Meal

1. NAVIGATION: total 2 taps

a. Tap “Server Screen” button on the home screen

b. Tap anywhere on the order that needs to delivered

2. DATA ENTRY: total 1 tap

a. Tap “Delivered” button to remove item from the queue

Clean Table

1. NAVIGATION: total 1 tap

a. Tap “Floor Plan” button on the home screen

2. DATA ENTRY: total 2 taps

a. Tap on the table that has been cleaned

Part 3.5 Domain Analysis

3.5.a. Domain Model

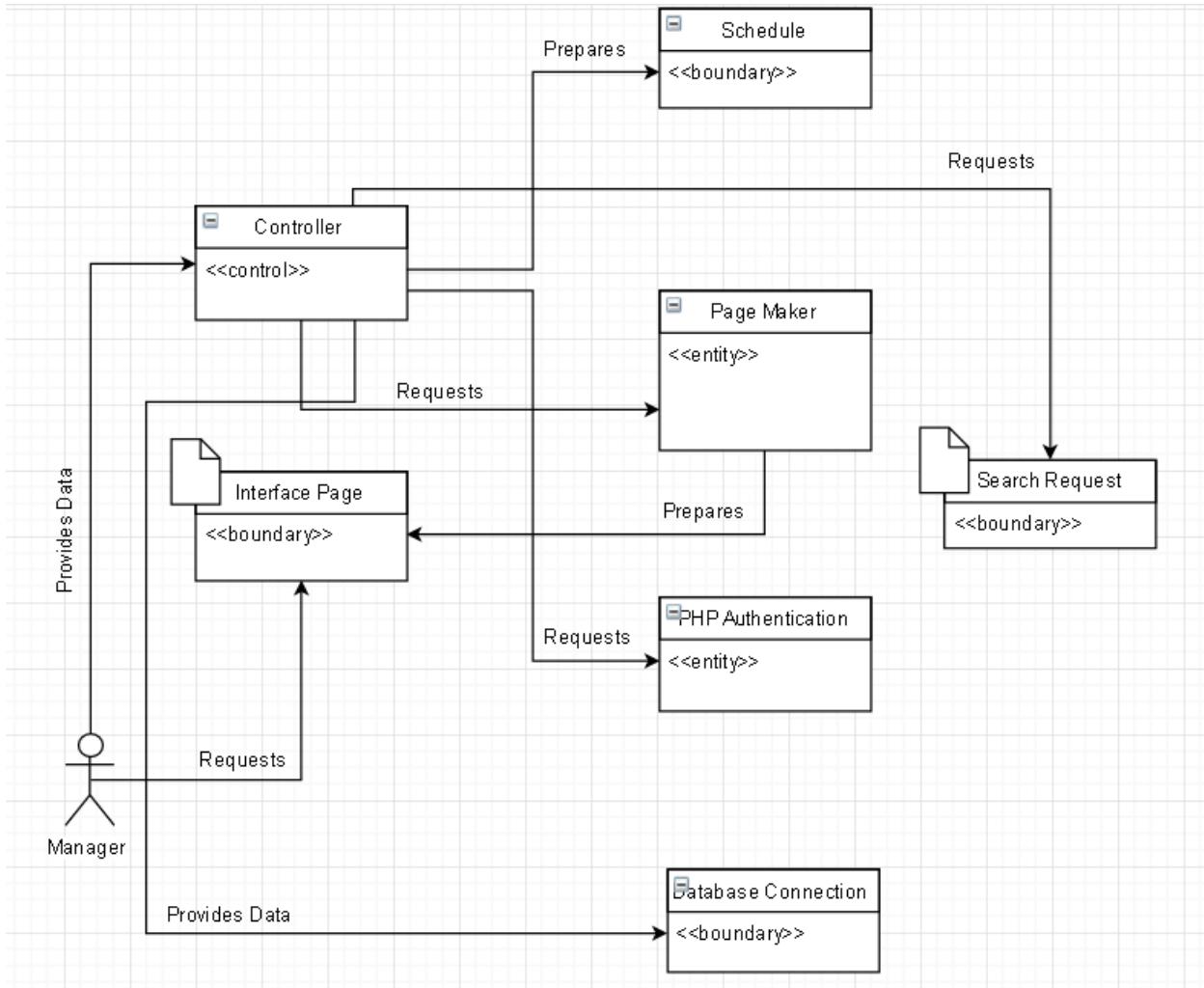


Figure 16: Managerial

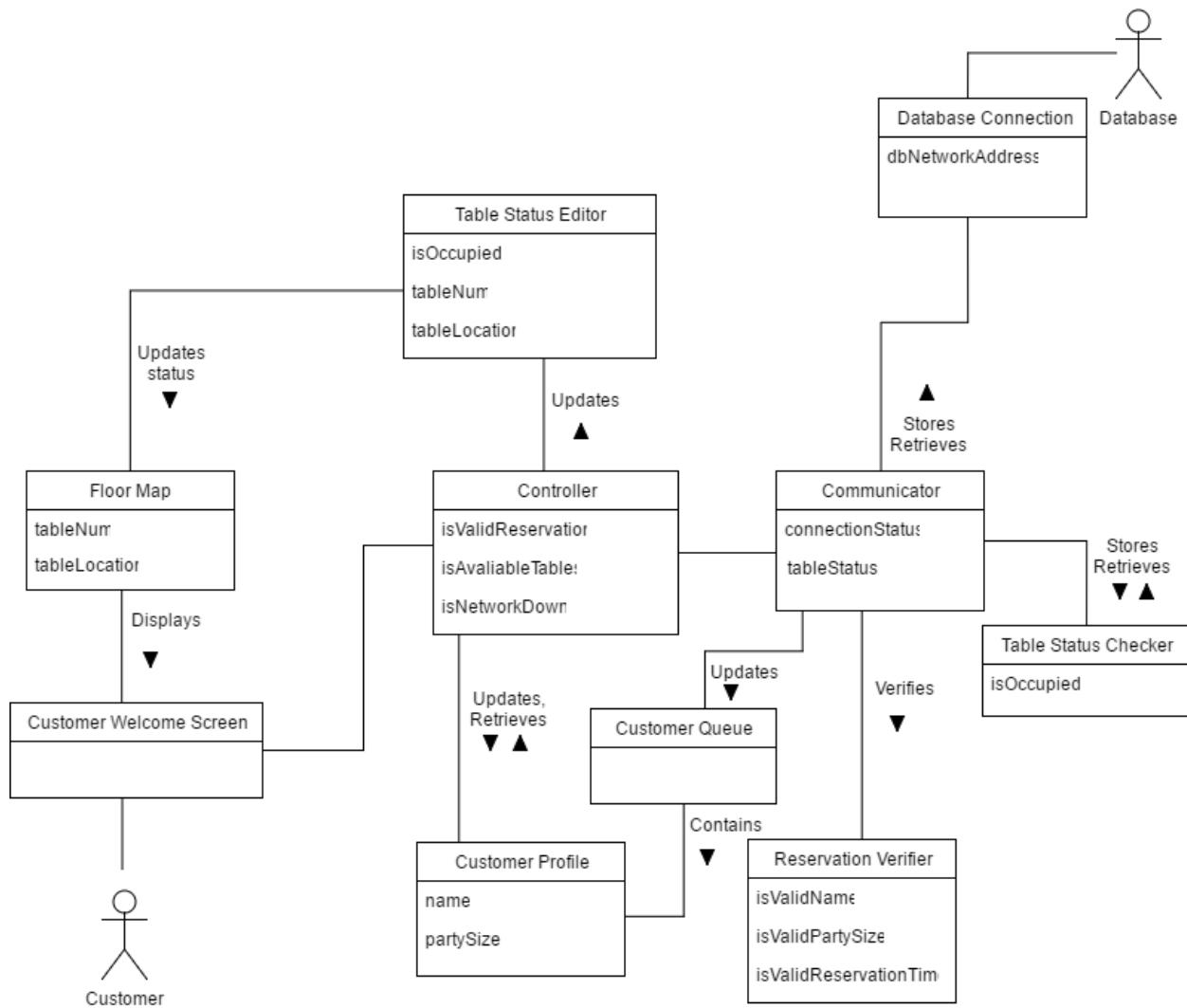


Figure 17: Customer

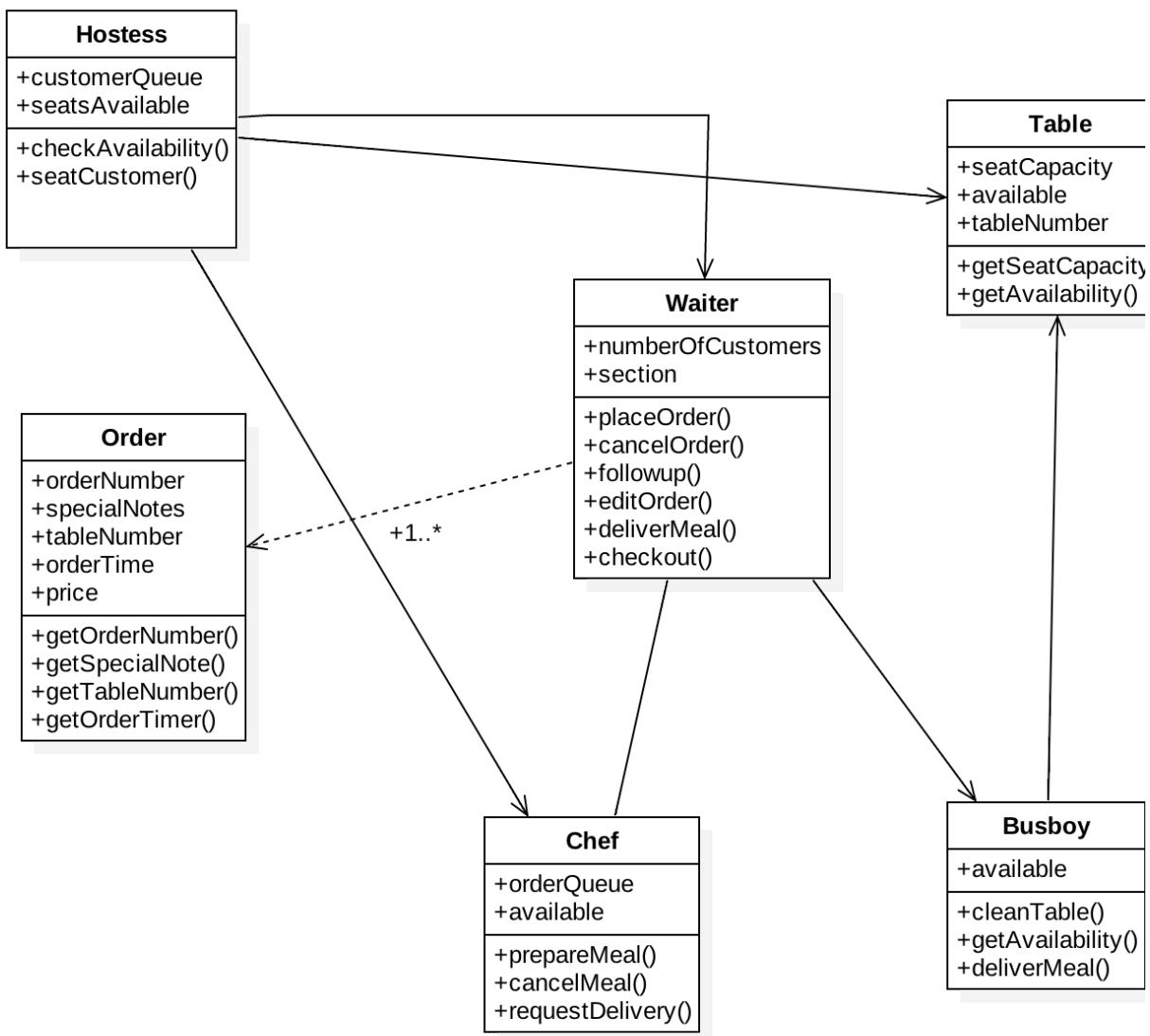


Figure 18: Kitchen

3.5.a.i. Concept Definitions

Table 14: UC - 02: EditShifts

Responsibility Description	Type	Concept Name
<i>R1: User of the program. Activates multiple interactions with the system via touch or click.</i>	<i>D</i>	<i>Manager</i>
<i>R2: Renders UI for shift schedule activated by the manager. For rendering Tablet or Desktop UI.</i>	<i>D</i>	<i>Page Maker</i>
<i>R3: Provides access to various databases that hold information about the shift table and employees.</i>	<i>D</i>	<i>Database Connection</i>
<i>R4: Confirms authentication for access to classified restaurant information in the Access Portal.</i>	<i>K</i>	<i>PHP Authentication</i>
<i>R5: Display shift table page for manager. Shown only to manager or others with administrative powers in the restaurant that passed the authentication phase.</i>	<i>K</i>	<i>Interface Page</i>
<i>R6: Notifies Manager of pending and nearing events such employees punchin in for their shift.</i>	<i>D</i>	<i>Notifier</i>
<i>R7: Allows Manager to search for restaurant related objects such as employee names.</i>	<i>K</i>	<i>Search Request</i>
<i>R8: Filter Manager's search request based on given criteria.</i>	<i>D</i>	<i>Postprocessor</i>
<i>R9: Archive all previous documents about the employees and past shift dates.</i>	<i>D</i>	<i>Archiver</i>
<i>R10: Employee schedule in database meant for interaction with Manager.</i>	<i>K</i>	<i>Schedule</i>
<i>R11: Coordinates actions of concepts. Conveys Manager interactions to other aspects of the program.</i>	<i>D</i>	<i>Controller</i>

Table 15: UC - 08: CheckInventory

Responsibility Description	Type	Concept Name
<i>R1: User of the program. Activates multiple interactions with the system via touch or click. Can view and edit employee shifts, menu, inventory, and manage restaurant expenses.</i>	D	<i>Manager</i>
<i>R2: Renders UI for inventory database activated by the manager. For rendering Tablet or Desktop UI.</i>	D	<i>Page Maker</i>
<i>R3: Provides access to inventory database. Used to retrieve and store information in database.</i>	D	<i>Database Connection</i>
<i>R4: Confirms authentication for access to classified restaurant information in the Access Portal.</i>	K	<i>PHP Authentication</i>
<i>R5: Display inventory stock page for manager. Shown only to manager or others with administrative powers in the restaurant that passed the authentication phase.</i>	K	<i>Interface Page</i>
<i>R6: Notifies Manager of pending and nearing events such as low stock.</i>	D	<i>Notifier</i>
<i>R7: Allows Manager to search for restaurant inventory items and ingredients.</i>	K	<i>Search Request</i>
<i>R8: Filter Manager's search request based on given criteria.</i>	D	<i>Postprocessor</i>
<i>R9: Archive all previous documents and ordering details. Keep past receipts of inventory item purchases for future reference.</i>	D	<i>Archiver</i>
<i>R10: Coordinates actions of concepts. Conveys Manager interactions to other aspects of the program.</i>	D	<i>Controller</i>

Table 16: UC – 17: NewReservation

Responsibility	Type	Concept
<i>R1: Coordinate actions of making a new reservation of a table and distribute the work to other concepts</i>	<i>D</i>	<i>Table Controller</i>
<i>R2: Display new reservation option for the customer</i>	<i>D</i>	<i>Interface</i>
<i>R3: Prompt the customer to input their name and party size</i>	<i>D</i>	<i>Table Controller</i>
<i>R4: Touch keyboard for customer to enter information</i>	<i>K</i>	<i>Touch Keyboard</i>
<i>R5: Keep the name and party size of the customer</i>	<i>K</i>	<i>Customer Profile</i>
<i>R6: Check the status of every table present in the restaurant (occupied, available)</i>	<i>D</i>	<i>Table Status Checker</i>
<i>R7: Map the available tables with their ID numbers</i>	<i>K</i>	<i>Floor Map</i>
<i>R8: Highlight the available table in the floor map</i>	<i>D</i>	<i>Table Highlighter</i>
<i>R9: Verify customer reservation</i>	<i>D</i>	<i>Reservation Verifier</i>
<i>R10: Prompt customer to be seated at table</i>	<i>D</i>	<i>Table Controller</i>
<i>R11: Change status of highlighted table</i>	<i>D</i>	<i>Table Status Editor</i>
<i>R12: Store the updated status of the table in the database</i>	<i>D</i>	<i>Communicator</i>
<i>R13: Estimate wait time for next available table</i>	<i>D</i>	<i>Table Timer</i>
<i>R14: Queue next customer to be seated</i>	<i>K</i>	<i>Customer Queue</i>

Table 17: UC - 23: OrderItems

Responsibility	Type	Concept
<i>R1: Coordinate actions of placing an order and distribute the work to other concepts</i>	<i>D</i>	<i>Order Controller</i>
<i>R2: Display ordering items tab for the customer</i>	<i>D</i>	<i>Interface</i>
<i>R3: Keep a list of dishes/drinks available</i>	<i>K</i>	<i>Menu</i>
<i>R4: Add dish/drink to ordered list</i>	<i>D</i>	<i>Order Item</i>
<i>R5: List of customer's ordered items</i>	<i>K</i>	<i>Ordered List</i>
<i>R6: Verify customer's order</i>	<i>D</i>	<i>Order Verifier</i>
<i>R7: Send customer's order to chef to process and table ID</i>	<i>D</i>	<i>Communicator</i>
<i>R8: Estimate wait time for order</i>	<i>D</i>	<i>Food Timer</i>
<i>R9: Keep the status of the customer's order as it is being processed</i>	<i>K</i>	<i>Order Status</i>
<i>R10: Display stage/progress of ordered items to customer</i>	<i>D</i>	<i>Interface</i>
<i>R11: Remove an item from ordered list</i>	<i>D</i>	<i>Ordered List Editor</i>
<i>R12: Store the updated ordered list in database</i>	<i>D</i>	<i>Communicator</i>

Table 18: UC - 30: CardPayment

Responsibility	Type	Concept
<i>R1: Coordinate actions of making a payment and distribute the work to other concepts</i>	D	<i>Payment Controller</i>
<i>R2: Keep list of ordered items and their cost</i>	K	<i>Bill</i>
<i>R3: Prompt user to swipe card or insert chip</i>	D	<i>Interface</i>
<i>R4: Read the card</i>	D	<i>Card Reader</i>
<i>R5: Verify payment information</i>	D	<i>Payment Verifier</i>
<i>R6: Print receipt of payment being processed successfully</i>	D	<i>Receipt Printer</i>
<i>R7: Paper that lists ordered items and total amount successfully paid</i>	K	<i>Receipt</i>
<i>R8: Call waiter for assistance</i>	D	<i>Waiter Notification</i>

Table 19: Kitchen, waiter

Responsibility	Type	Concept
<i>R1: Confirm chef is on duty</i>	<i>S</i>	<i>Chef</i>
<i>R2: Ensure Customer is satisfactory</i>	<i>C</i>	<i>Waiter</i>
<i>R3: Ensure meal has been updated</i>	<i>C</i>	<i>Waiter</i>
<i>R4: Confirm meal cancellations</i>	<i>C</i>	<i>Waiter</i>
<i>R5: Confirm meal added</i>	<i>C</i>	<i>Waiter</i>
<i>R6: Ensure meal quality</i>	<i>C</i>	<i>Chef</i>
<i>R7: Confirm meal goes to correct customer</i>	<i>C</i>	<i>Waiter</i>
<i>R8: Customer has finished meal</i>	<i>C</i>	<i>Waiter</i>
<i>R9: Customer pays for service</i>	<i>C</i>	<i>Waiter</i>
<i>R10: Busyboy has been notified</i>	<i>C</i>	<i>Host</i>
<i>R11: Table cleaned</i>	<i>C</i>	<i>Busboy</i>
<i>R12: Busyboy is available</i>	<i>A</i>	<i>Busboy</i>

3.5.a.ii. Association Definitions

Concept Pair	Association Description	Association Name
---------------------	--------------------------------	-------------------------

Table 19: Managerial

Concept Pair	Association Description	Association Name
<i>Manager ⇄ Controller</i>	<i>Manager provides credentials to be conveyed and checked.</i>	<i>Provides Data</i>
<i>Controller ⇄ PHP Authentication</i>	<i>Controller requests the confirmation of the individual logging in.</i>	<i>Requests</i>
<i>Controller ⇄ Page Maker</i>	<i>After confirmation, authorized individual is provided with a page displaying options to restaurant information.</i>	<i>Requests</i>
<i>Page Maker ⇄ Interface Page</i>	<i>Page information is sent to interface for Manager to interact with. For "Edit Shifts" use case, Manager is provided with employees and calendar.</i>	<i>Prepares</i>
<i>Controller ⇄ Database Connection</i>	<i>All search requests for employees get sent from the controller to the database.</i>	<i>Provides Data</i>
<i>Manager ⇄ Interface Page</i>	<i>Manager interacts with the user interface and selects an employee name on the calendar.</i>	<i>Requests</i>
<i>Controller ⇄ Database Connection</i>	<i>Information about the employee is displayed in UI for manager to observe. This data is provided from the database.</i>	<i>Provides Data</i>
<i>Controller ⇄ Search Request</i>	<i>Any searches made by manager are searched for in the database and returned to the UI for display.</i>	<i>Requests</i>
<i>Controller ⇄ Schedule</i>	<i>Changes in database are displayed in the schedule provided to the Manager.</i>	<i>Prepares</i>

Table 20: Customer

<i>Controller ↔ Table Status Editor</i>	<i>Controller updates table status to occupied or unoccupied</i>	<i>Updates</i>
<i>Controller ↔ Customer Profile</i>	<i>Controller adds or removes customer profile, also retrieves for verification</i>	<i>Updates, Retrieves</i>
<i>Communicator ↔ Table Status Checker</i>	<i>Retrieves most up to date table status(Occupied, or Unoccupied),</i>	<i>Stores, Retrieves</i>
<i>Communicator ↔ Customer Queue</i>	<i>Communicator updates customer queue with waiting customers</i>	<i>Updates</i>
<i>Communicator ↔ DB Connection</i>	<i>Communicator stores local customer profile into the database and retrieves customer profile from database</i>	<i>Stores, Retrieves</i>
<i>Communicator ↔ Reservation Verifier</i>	<i>Checks the customer name, party size, and current time to verify there is a pre existing reservation with the customer profile.</i>	<i>Verifies</i>
<i>Table Status Editor ↔ Floor Map</i>	<i>Table Status Editor updates the status of the floor map to be displayed to the customer</i>	<i>Updates status</i>

Table 21: Host, Chef, Busboy, Waiter

Concept Pair	Association Description	Association Name
<i>Hostess-Water</i>	<i>Hostess assigns a customer and table to a waiter</i>	<i>Provides Data</i>
<i>Watier-Chef</i>	<i>Waiter add an order to the chef order queue</i>	<i>Provides Data</i>
<i>Chef-Busboy</i>	<i>Chef request busboy to deliver food</i>	<i>Request Notify</i>
<i>Watier-Busboy</i>	<i>Request table to clean</i>	<i>Request Notify</i>
<i>Waiter-Order</i>	<i>Waiter creates a order</i>	<i>Generates</i>
<i>Hostess-Table</i>	<i>Hostess assigns table</i>	<i>Update</i>

3.5.a.iii. Attribute Definitions

Table 22: Managerial

Concept	Attributes	Attribute Description
<i>Employee Profile</i>	<i>employeeName</i>	<i>Name of employee of restaurant.</i>
	<i>employeeSalary</i>	<i>Salary of employee of restaurant.</i>
	<i>employeeStatus</i>	<i>Status of whether the employee is currently in or not.</i>
<i>Manager Credentials</i>	<i>managerName</i>	<i>Name of authorized manager.</i>
	<i>managerPassword</i>	<i>Password of authorized manager used for access to manager portal.</i>
<i>Item</i>	<i>numberInStock</i>	<i>Quantity of item or ingredient in stock.</i>
	<i>itemPrice</i>	<i>Price of item of ingredient in stock.</i>
	<i>itemStatus</i>	<i>Stock status of item or ingredient.</i>
<i>Shift Calendar</i>	<i>changeEmployeeShift</i>	<i>Allows manager to change the employee working a particular shift.</i>
	<i>deleteEmployeeShift</i>	<i>Allows manager to remove an employee from a particular shift.</i>
	<i>addEmployeeShift</i>	<i>Allows manager to add an employee to a particular shift.</i>
	<i>viewWorkMonth</i>	<i>Manager may view a specific month.</i>
	<i>viewWorkDay</i>	<i>Manager may view a specific day.</i>
	<i>viewWorkWeek</i>	<i>Manager may view a specific week.</i>
<i>Finance Table</i>	<i>viewFinancialMonth</i>	<i>Show estimated finances for the month.</i>
	<i>viewFinancialDay</i>	<i>Show estimated finances for the day.</i>
	<i>viewFinancialWeek</i>	<i>Show estimated finances for the week.</i>
	<i>sortExpenses</i>	<i>Sort expenses in descending or ascending order.</i>
	<i>viewRestaurantBudget</i>	<i>Show budget of restaurant to compare with expenses.</i>
	<i>viewRestaurantStanding</i>	<i>Show bank statement of restaurant.</i>
<i>Menu</i>	<i>changeMenuItem</i>	<i>Allows manager to change an aspect of a menu item.</i>
	<i>deleteMenuItem</i>	<i>Allows manager to remove a menu item.</i>
<i>Notifier</i>	<i>inventoryStatus</i>	<i>Notifies Manager of status of inventory items. (Is the item low in stock or soon to be low?)</i>
	<i>lunchRushStatus</i>	<i>Notifies Manager that lunch rush is coming soon.</i>

	<i>employeeCheckInStatus</i>	<i>Notifies Manager of employees checking in.</i>
<i>Archiver</i>	<i>employeeInformation</i>	<i>Past employee information available for review by manager.</i>
	<i>restaurantFinances</i>	<i>Past restaurant bills and invoices to be reviewed by Manager.</i>
	<i>menuChanges</i>	<i>Past menus and changes to be reviewed by Manager.</i>
	<i>inventoryItems</i>	<i>Past inventory items to be reviewed by Manager.</i>
<i>Postprocessor</i>	<i>searchParameters</i>	<i>Saves previous search parameters made by the Manager.</i>
<i>Search Request</i>	<i>searchParameters</i>	<i>Based on saved search parameters. Also takes new search parameters.</i>
	<i>userIdentities</i>	<i>Searches user credentials in database for access to Manager Portal.</i>

Table 23: Customer

Concept	Attribute	Attribute Description
<i>Customer Profile</i>	<i>name</i>	<i>Used to distinguish between reservations</i>
	<i>partySize</i>	<i>Needed to determine if table is available for the number of people and the estimated wait time for next available table that fits the party size</i>
<i>Table Status Checker</i>	<i>isOccupied</i>	<i>Current table status (occupied, available)</i>
<i>Floor Map</i>	<i>tableNum</i>	<i>Holds the identity of the available table</i>
<i>Reservation Verifier</i>	<i>isValidName</i>	<i>Determines if the given customer has a reservation</i>
	<i>isValidPartySize</i>	<i>Determines if the customer has a reservation with the given party size.</i>
	<i>isReservationTime</i>	<i>Verifies the Reservation time with the current time</i>
<i>Table Status Editor</i>	<i>isOccupied</i>	<i>Describes the current status of the table (available, occupied)</i>
	<i>tableNum</i>	<i>Holds the identity of the table whose status needs to be changed</i>
	<i>tableLocation</i>	<i>Holds the location of the table whose status needs to be changed</i>
<i>Table Timer</i>	<i>waitTime</i>	<i>Estimates the wait time based off of the current customer queue</i>
<i>Customer Queue</i>	<i>customerQueue</i>	<i>The current queue of customers waiting to be seated</i>
	<i>queueSize</i>	<i>Holds the number of people in queue waiting to be seated</i>
<i>Menu</i>	<i>foodItem</i>	<i>Name of food item currently available</i>
	<i>price</i>	<i>The cost of the food item</i>
	<i>icon</i>	<i>Describes the food item through a picture</i>
	<i>category</i>	<i>Used to divide food items into categories i.e. salads, desserts, side orders, soups</i>
<i>Ordered Item</i>	<i>foodItem</i>	<i>Needed to identify the food item to be added to order</i>
	<i>tableNum</i>	<i>Holds the identity of the table that desires to order item</i>
	<i>addItem</i>	<i>Needed to add item to order</i>
	<i>category</i>	<i>Describes category the item is from in menu</i>
<i>Ordered List</i>	<i>foodItem</i>	<i>Holds the names of the items customer ordered</i>
	<i>price</i>	<i>The cost of each food item</i>
	<i>tableNum</i>	<i>Holds the identity of the table that placed order</i>

Concept	Attributes	Attribute Description
<i>Busboy</i>	<i>Table Information</i>	<i>Used to check for unclean tables</i>
	<i>Complete Order List</i>	<i>Used to deliver orders to guests</i>
<i>Waiter</i>	<i>Table Information</i>	<i>Used to check for guests in-need</i>
	<i>Complete Order List</i>	<i>Used to deliver orders to guests</i>
<i>Hostess</i>	<i>Table Information</i>	<i>Used to seat guests</i>
<i>Chef</i>	<i>Requested Order List</i>	<i>Used to prepare requested meals</i>

3.5.a.iv. Traceability Matrix

Table 24: Managerial

REQ	<i>PW</i>	<i>UC-1</i>	<i>UC-2</i>	<i>UC-3</i>	<i>UC-4</i>	<i>UC-5</i>	<i>UC-6</i>	<i>UC-7</i>	<i>UC-8</i>	<i>UC-9</i>	<i>UC-10</i>	<i>UC-11</i>	<i>UC-12</i>	<i>UC-13</i>	<i>UC-14</i>	<i>UC-15</i>
<i>M-1</i>	3	X														
<i>M-2</i>	2		X													
<i>M-3</i>	4			X												
<i>M-4</i>	5				X											
<i>M-5</i>	3					X										
<i>M-6</i>	3						X	X								
<i>M-7</i>	5								X							
<i>M-8</i>	1									X						
<i>M-9</i>	3										X					
<i>M-10</i>	3										X					
<i>M-11</i>	2													X		
<i>M-12</i>	3											X				
<i>M-13</i>	1													X		
<i>M-14</i>	1														X	
<i>Maw PW</i>		3	2	4	5	3	3	3	5	1	3	3	3	1	2	1
<i>Total PW</i>		3	2	4	5	3	3	3	5	1	3	3	3	1	2	1

Use Case	PW	Domain Concepts									
		EmployeeProfile	ManagerCredentials	Item	ShiftCalendar	FinanceTable	Menu	Notifier	Archiver	PostProcessor	SearchRequest
UC1	3		X					X			
UC2	2	X	X		X				X	X	X
UC3	4		X				X		X		
UC4	5		X						X		
UC5	3		X	X			X		X	X	X
UC6	3		X			X		X	X	X	X
UC7	3		X			X			X		
UC8	5		X	X				X			
UC9	1	X	X		X						
UC10	3	X	X		X			X		X	X
UC11	3	X	X		X			X		X	X
UC12	3	X	X		X			X		X	X
UC13	1		X							X	X
UC14	2	X	X								
UC15	1		X						X		X
Max PW		14	42	8	12	6	7	20	21	18	19
Total PW		14	42	8	12	6	7	20	21	18	19

The traceability matrix above relates the domain concepts with the use cases. Along with this information are the priority weights of the use cases. This determines how important a use case is in the final product of the program. Many of the managerial tasks of a restaurant are important and of high priority, therefore many of the proposed cases are to be implemented.

Table 25: Customer

REQ		PW	C O N T R O L L E R	C O M M U N I C A T O R	I N T E R F A C E	F L O O R	T A B L E	C U S T O M E R	C U S T O M E R	R E S E R V A T I O N	D A T A B A S E
UC	16	12	X	X	X	X		X		X	X
	17	20	X	X	X		X	X			X
	18	12	X	X	X					X	X
	19	13	X	X	X			X	X		X
	20	5	X	X	X	X	X				X
Max PW			20	20	20	12	20	20	13	12	20
Total PW			62	62	62	17	25	45	13	24	62

3.5.b. System Operational Contracts

Operation: Edit Shifts

Responsibility: Maintain and edit shifts, and employee information in restaurant's database.

Use Cases: UC-2

Exceptions: None

Preconditions:

- The manager is authorized personnel that can access the shift editing portal.
- There is a database with existing information of the restaurant.

Postconditions:

- The shifts or information of employees shall be correctly and immediately updated in the database.

Operation: Check Inventory

Responsibility: Access the current inventory and see which items need to be restocked. Conveniently order any quantity of low stock items, and update the inventory database when new orders come in.

Use Cases: UC-8

Exceptions: None

Preconditions:

- The manager is authorized personnel that can access the inventory portal.
- There is a database with existing information of the restaurant's inventory.

Postconditions:

- The items ordered will be purchased and the database updated.

Operation: PreReservation**Responsibility:****Use Cases:** UC-16**Exceptions:** Reserved table is taken due to it being incorrectly assigned.**Preconditions:**

- Customer has made a reservation by phone or through the online form, and their reservation is stored in the restaurant database.

Postconditions:

- Customer successfully checks in to the Restaurant and is seated at their table
- The customers pre-reservation is removed from the database

Operation: NewReservation**Responsibility:****Use Cases:** UC-17**Exceptions:** None**Preconditions:**

- None

Postconditions:

- Customer is put into Customer Queue

Operation: CardPayment**Responsibility:****Use Cases:** UC-17**Exceptions:** Card Declined, Card Read Error**Preconditions:**

- The total cost of the bill has been calculated

Postconditions:

- Payment is confirmed and customer is free to leave.
- Waiter/Busboy receive notification of successful payment, notifying them the table will soon be ready to be cleaned.

Operation: BitcoinPayment

Responsibility:

Use Cases: UC-??

Exceptions: None

Preconditions:

- Total cost of the bill has been calculated in USD and converted to BTC
- There is a public key address in the restaurant database to send the BTC to.

Postconditions:

- Payment is confirmed and customer is free to leave.
- Waiter/Busboy receive notification of successful payment, notifying them the table will soon be ready to be cleaned.

Operation: Place Order

Responsibility: Send guest orders to the chef's cooking queue

Use Cases: US-1

Exceptions: None

Preconditions:

- The guests have been seated at an available table space.
-

Postconditions:

- The chef receives and prepares the order.
-

Operation: Checkout

Responsibility: Ensure the guest pays for his meal through standard payment options before leaving

Use Cases: US-5

Exceptions: None

Preconditions:

- The customer has finished their meal and was satisfied with the service.
 - The customer has yet to depart.
-

Postconditions:

- The sales database is updated
 - The busboy is notified that a table needs to be cleaned.
-

3.5.c. Mathematical Model

Manager – Food Cost Formula:

Evaluating how each type of restaurant expense compares to the total expenses is often confusing. This food cost percentage formula gives managers an idea of how much the restaurant is spending on food in a given time period. The ending inventory must be taken after each service for the formula to remain accurate.

$$\text{Food Cost \%} = \frac{(\text{Beginning Inventory} + \text{Purchases} - \text{Ending Inventory})}{\text{Food Sales}}$$

30% food cost means that you spent 30 cents for every dollar you took in sales.

Bitcoin Price:

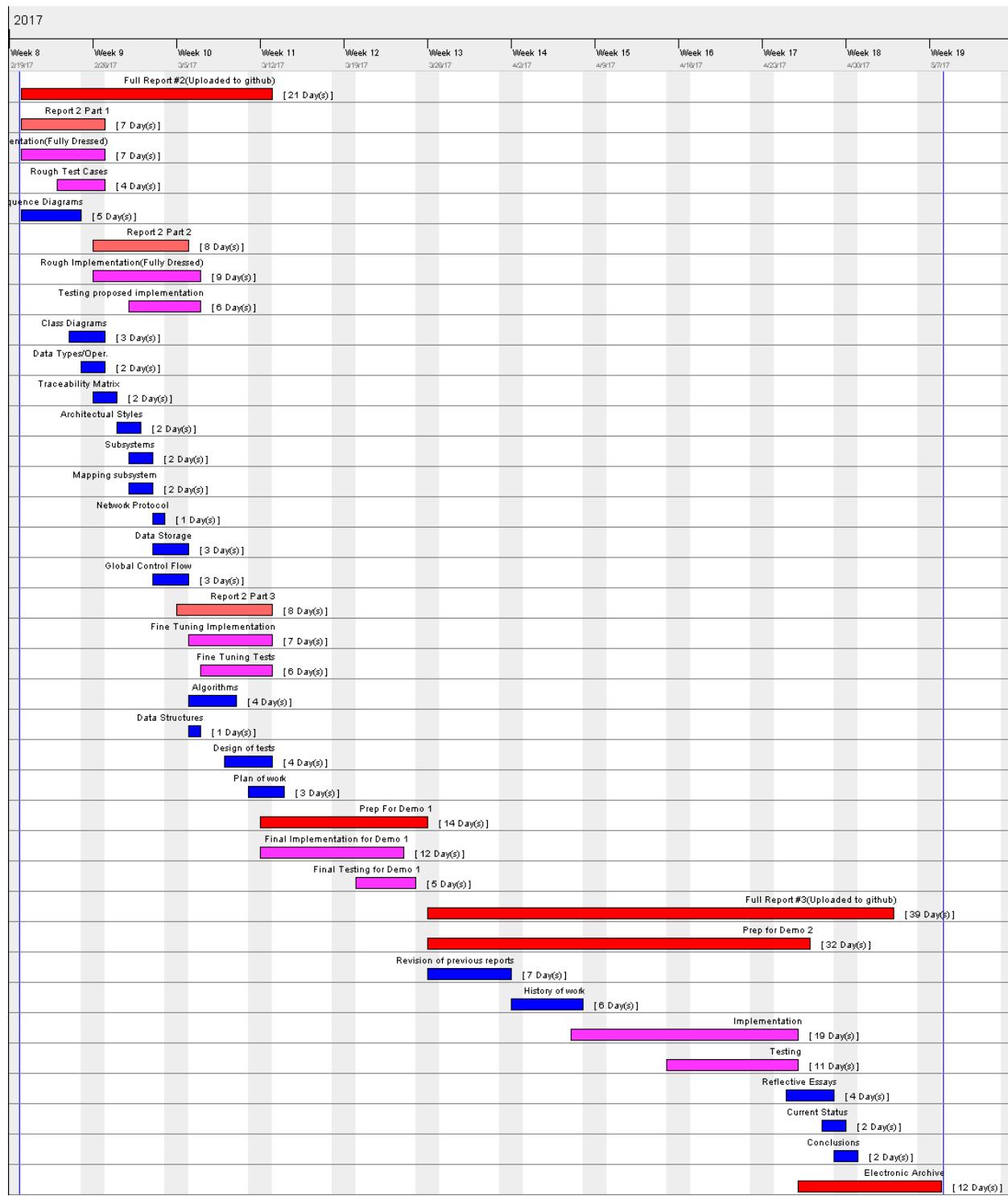
Calculating the price of a meal in BTC will be determined by pulling the exchange rate from an established exchange source such as Coinbase, or Bitstamp. These exchanges offer API's that can be integrated into the ZestWare system.

$$\text{Total Meal Cost BTC} = \frac{\text{Total Meal Cost USD}}{\text{BTC Exchange Rate USD}}$$

Where the Total Meal Cost USD includes the tax rate of the respective jurisdiction.

3.6. Plan of Work

 GANTT project		
Name	Begin date	End date
• Full Report #2(Uploaded to github)	2/20/17	3/12/17
• Report 2 Part 1	2/20/17	2/26/17
• Rough Implementation(Fully Dressed)	2/20/17	2/26/17
• Rough Test Cases	2/23/17	2/26/17
• Sequence Diagrams	2/20/17	2/24/17
• Report 2 Part 2	2/26/17	3/5/17
• Rough Implementation(Fully Dressed)	2/26/17	3/6/17
• Testing proposed implementation	3/1/17	3/6/17
• Class Diagrams	2/24/17	2/26/17
• Data Types/Oper.	2/25/17	2/26/17
• Traceability Matrix	2/26/17	2/27/17
• Architectural Styles	2/28/17	3/1/17
• Subsystems	3/1/17	3/2/17
• Mapping subsystem	3/1/17	3/2/17
• Network Protocol	3/3/17	3/3/17
• Data Storage	3/3/17	3/5/17
• Global Control Flow	3/3/17	3/5/17
• Report 2 Part 3	3/5/17	3/12/17
• Fine Tuning Implementation	3/6/17	3/12/17
• Fine Tuning Tests	3/7/17	3/12/17
• Algorithms	3/6/17	3/9/17
• Data Structures	3/6/17	3/6/17
• Design of tests	3/9/17	3/12/17
• Plan of work	3/11/17	3/13/17
• Prep For Demo 1	3/12/17	3/25/17
• Final Implementation for Demo 1	3/12/17	3/23/17
• Final Testing for Demo 1	3/20/17	3/24/17
• Full Report #3(Uploaded to github)	3/26/17	5/3/17
• Prep for Demo 2	3/26/17	4/26/17
• Revision of previous reports	3/26/17	4/1/17
• History of work	4/2/17	4/7/17
• Implementation	4/7/17	4/25/17
• Testing	4/15/17	4/25/17
• Reflective Essays	4/25/17	4/28/17
• Current Status	4/28/17	4/29/17
• Conclusions	4/29/17	4/30/17
• Electronic Archive	4/26/17	5/7/17



3.7. References

- Anon, 2015. Opportunities To Automate Your Restaurant. Restaurant Engine. Available at: <http://restaurantengine.com/automate-your-restaurant/> [Accessed January 29, 2017].
- Banderpalle, S. et al., 2015. Food-E-Z: Report 3 Part 1: Restaurant Automation.
- Armanus, A. et al., 2014. Why Wait Final Report: Restaurant Automation.
- Marsic, I., Software Engineering,
- Anon, What is a Gantt chart? What is a Gantt Chart? Gantt Chart Information, history and Software. Available at: <http://www.gantt.com/> [Accessed February 17, 2017].
- Anon, 2014. How are public and private keys related to the wallet? Bitcoin Stack Exchange. Available at: <http://bitcoin.stackexchange.com/questions/19950/how-are-public-and-private-keys-related-to-the-wallet> [Accessed February 17, 2017].
- Anon, Bitcoin - Open source P2P money. Bitcoin - Open source P2P money. Available at: <https://bitcoin.org/en/> [Accessed February 18, 2017].
- Anon, how to make vertical column headers in google docs. how to make vertical column headers in google docs. Available at: <http://www.quikstarts.com/Internet/Google-Docs/how-to-make-vertical-column-headers-in-google-docs/> [Accessed February 19, 2017].
- Anon, 2014. 7 Surprisingly Common Problems That Restaurants Face. Restaurant Engine. Available at: <http://restaurantengine.com/common-problems-restaurants-face/> [Accessed February 19, 2017].
- Tips and Tools for Controlling Your Food Cost : How to Calculate Food Cost | StarChefs.com. 2017. Tips and Tools for Controlling Your Food Cost : How to Calculate Food Cost | StarChefs.com. [ONLINE] Available at: <http://www.starchef.com/cook/business-tips/how-to-calculate-food-cost>. [Accessed 18 February 2017].
- Anon, 2014. 7 Surprisingly Common Problems That Restaurants Face. Restaurant Engine. Available at: <http://restaurantengine.com/common-problems-restaurants-face/> [Accessed February 19, 2017].
- Seating Chart: https://cloud.smartdraw.com/editor.aspx?templateId=7bb33318-f178-419d-90a7-a341353ff7d7#docId=Floor_Plan.sdr&ownerUserId=15406313&depo=2
- Food: https://upload.wikimedia.org/wikipedia/commons/2/2e/Fast_food_meal.jpg
- <http://jejeprime.com/wp-content/uploads/2015/11/restaurant-awards-2-www.jejeprime.com.jpg>
- Drinks: http://www.spiegelau-perfectservecollection.com/uploads/tx_templavoila/header_drinks_3.jpg
- Waiter icon: <http://www.free-icons-download.net/images/restaurant-waiter-icon-64769.png>
- Entertainment icon: <https://www.downtowntomsriver.com/fullcalendar/0images/00-entertainment.png>

Side orders: <http://bonappetitpizzeria.com/wp-content/uploads/2014/04/appetizers1-600x270.jpg>

Desserts: <http://clv.h-cdn.co/assets/15/12/980x490/landscape-1426613152-54eb7ec16c821-brown-sugar-balsamic-strawberry-sauce-recipe-clx0514-s2.jpg>

Specials: <http://www.mugshotzpewaukee.com/wp-content/uploads/2013/12/Daily-Specials.jpg>

Bitcoin: https://2.bp.blogspot.com/-L0mOtpX1yDU/WFwWfsMT_pl/AAAAAAABRAI/Ka6QwvhpnUk4tV-tlptbhAjy4coBLQ5awCLcB/s1600/Two-Bitcoins.png

Keyboard:

https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwi3v9HTmozSAhVI5CYKHf3qAhgQjBwIBA&url=http%3A%2F%2Fonsongapp.s3.amazonaws.com%2Fmanual%2Fa6dd435d078cc248355ecd53f14758c3.png&psig=AFQjCNFlKLAUD0z_d0SeqHVXC2s0tdh9RQ&ust=1487045586660946

End of project responsibility breakdown

Shehpar Sohail, Fahd Humayun, Nathan Morgenstern:

- Github e-archive management
- Customer table screen interface
- Customer welcome screen interface
- Customer waiting area screen interface
- Design floor map
- Design and implementation of tests associated with payments, reservation, and ordering on the customer side
- Design and implementation of payment processing functions, reservation capability, and ordering.
- Integrating the customer side of the restaurant with the kitchen/waiter side.
- Implementing Bitcoin Payment processor, and converting the exchange rate to USD

Ama Freeman and Raphaelle Marcial:

- Interview of restaurants for problem domain
- Introduction and conclusion
- Manager portal interface
- Inventory interface
- Manager/general restaurant glossary terms

Alexander Dewey and Dwayne Anthony:

- Restaurant layout view and functionality for staff
- Chef's queue for preparing orders and sending them to waiter
- Waiter's queue for orders that need to be delivered
- Table-status changing ability for all members of staff (declaring if they're free, occupied, clean, unclean, etc.)
- Schedule viewer for all members of staff
- Request cover and request off capabilities for all staff
- Clock-in capabilities for all members of staff
- Remote order editing for waiters
- Waiter's the ability to process transaction in cash for guest
- Waiter's ability to print receipts for guests if they wish to have it

Entire report broken down into Managerial, Customer, and Kitchen section.

Managerial part

R E S P O N S I B I L 	MANAGER	Team Member Name	
		AMA	RAPHAELLE
<i>Project Management</i>		50%	50%
<i>Sec.1: Customer Statement of Requirements</i>		50%	50%
<i>Sec.2: System Requirements</i>		50%	50%
<i>Sec.3: Functional Requirements Specification</i>		40%	60%
<i>Sec.4: User Interface Specs</i>		60%	40%
<i>Sec.5: Domain Analysis</i>		50%	50%
<i>Sec.6: Plan of Work</i>		50%	50%

Customer part

	Fahd Humayun	Shehpar Sohail	Nathan Morgenstern
Project Management	98%	1%	1%
Customer Statement of Requirements	40%	30%	30%
System Requirements	33%	33%	33%
Functional Requirements Specification	20%	40%	10%
User Interface Specification	75%	15%	10%
Domain Analysis	20%	40%	40%
Interaction Diagrams	10%	80%	10%
Plan of Work	33%	33%	33%
References	20%	20%	60%

Kitchen, host, chef, waiter, busboy part

	Alex	Dwayne
Project Management	70%	30%
Sec 1: Customer Statement of Requirements	100%	0%
Sec 2: System Requirements	20%	80%
Sec 3: Functional Requirements Specification	0%	100%
Sec 4: User Interface Specs	80%	20%
Sec 5: Domain Analysis	30%	70%