14:332:452:01 Software Engineering Spring ‘14
Group 4
Part 1&2 Submission: February 16,2014

WHY WAIT

A Restaurant Automation System

http://mitulgada.wix.com/whywait

Group Members:
Amgad Armanus
Jake Chou
Mitul Gada
Avni Patel
Nirjan Thayaparan
Diego Urquiza
Christian Youssef

Project Management

All team members contributed equally.

Table of Contents

1) Customer Requirements

a.) Problem Statements

The following are problems faced in restaurants today and are divided by the different positions.

Chef

Problem -Because chefs must constantly remain in the kitchen, they cannot be bothered with the placement time of orders. There is a great inefficiency between the time customers order food to the time when they receive the food. The waiter has to take the order and deliver that order to the kitchen, then the chef must make sure that that meal are prepared on time while simultaneously preparing other meals as well. The food then sits in the kitchen until it is picked up by the waiter for table delivery. The inefficiency is even more faulted since there are multiple customers who face these same problems. For the chef, whose priority is to cook the food, it is arduous to worry about these external issues especially as there is a multiple of customers and just as many meals to cook.

Solution - The chef would want a system that would allow them to get orders automatically, prioritize the meals ordered, and then notify a waiter that the meal is ready to be taken.

Problem - Between cooking and taking orders, there is a ton of room for error when it comes to supplies. One day there can be too many carrots and another there can be too little onions. This can entail a huge loss in profit since a customer can order something and the supplies may not be present. There could also be a loss of profit if we oversupply and the ingredients are essentially wasted. Lastly, there are certain days of the week when more supplies are needed than they are on other days. These days, the chances of not having the supplies and evidently losing profit from making the food is more likely.

Solution - It would be very beneficial if chefs could log the amount of ingredients used and what is needed. Also if there was a way to analyze when more supplies are needed on certain days, this would make it easier and take away the guesswork on how much is needed.

Problem - In the kitchen, the chef is preparing multiple meals all at the same time. Be it a gourmet salmon dish or fresh bread, the chef must focus on each detail to ensure quality, but also making sure it is delivered as quickly as possible. Thus, a problem arises when the chef must focus on cooking meals and ingredients with different timings while simultaneously estimating the time for each to complete. With all the different foods that all begin cooking at different times, it is hard for the chef to keep track of the times of the food, which can lead to reduced quality in meals and slow down preparations for upcoming meals.

Solution - The chef would want the system to assist in keeping tabs on when multiple ingredients or meals are being cooked for a certain amount of time, and when they have completed. This way chefs would not have to worry about cooking time and could instead be notified when food is finished cooking.

Customer Sign-in

Problem - Restaurants these days are always trying to find a way to satisfy customers. One thing they always look at is the waiting and seating arrangements in the beginning. Often when restaurants are packed, customers that come in have to wait. The problem is, restaurants don't know exactly how long it will take so they estimate it. Restaurants want to be more organized with their table management and keep track of their customers. This also leads to another problem with the status of the tables. Restaurants want to know if tables are ready to clean immediately so they can be attended to. Restaurants will also want to know how long customers have been sitting at the table. This will help tell them if they are almost done so the busboys can clean the table. Overall the problem restaurants are trying to fix these days is the satisfaction of the customers by giving them seating faster, and the organization of all the tables so they can be attended faster.

Solution - A solution that is appealing to the restaurant will to have a customer seating PC that will be in GUI. This customer seating PC will be in real time so it can show not only the customers, but the restaurant workers all the tables occupied or not. This way new customers can pick the available tables to their liking. For example, some customers might prefer sitting near a window, or booth, etc. And since this will be in real time, the exact times of when customers were seated will show up. This will overall give a much more accurate estimate of how much longer customers will take at the table since the time is there. Customers can see this when arriving and have a good idea when customers might leave and restaurant workers can use this information to inform busboys that a table is about to be ready to clean. This organized system will not only help restaurant workers being faster and efficient, but improve customer satisfaction.

Cashier

Problem - When doing transactions, restaurants aim for a fast and errorless execution to make it easier for the customer. One thing restaurants are worried about is receipt handling. Some customers might prefer an electronic form for a receipt so they won't lose the hard copy. Many restaurants only offer hard copies for receipts which may be unsatisfying for some customers. Another problem is the amount of change given. Often times the receptionist takes longer handling the change and might even make a mistake forcing to fix the issue which ultimately slows down the business. It is in the restaurant's best interest to handle the final transaction professionally to satisfy the customers and to organized the business.

Solution - A simple solution that restaurants will love is an automated cashier tablet that will have an option to either print out a receipt, have it emailed, and even both if wanted. This option will be appealing to all customers. The customers who want the hard copy can get it and the ones who often lose things or unorganized are able to get an electronic form to access easily. This also helps the restaurant be organized since they can keep the files tabbed and easily accessible as well as saves money by printing out less receipts. Another problem it solves is the slower process and human error. By having an automatic change dispenser, it eliminates the error of the receptionist giving out the wrong change. This not only speeds up the process but also saves the transaction time which ultimately improves customer service.

BusBoy

Problem - A lot of times, busboys would be standing around waiting for tables to be cleaned once the customers leave. Restaurants around would want busboys to clean the tables as soon as the customers left to speed up the service. Also, sometimes the waiter will also need help bringing out dishes to customers with large parties. Busboys will be needed to help out the waiters. Restaurant's want to fix the problem where the busboy will always be ready and there fast to help out the customers.

Solution - A solution restaurants will need is the busboy tablet. This tablet will notify the busboys of the tables needed to clean immediately so they can get straight to it. This will speed up the business and improve customer satisfaction since the tables will be ready faster. The tablet will notify the busboys of all duties needed to perform which improves efficiency. Another example is when a waiter needs help to carry food out. The tablet will tell the busboys and they will go to help the waiters serve food making the restaurant an altogether team helping out one another to improve the business.

Manager

Problem - Restaurant managers spend a lot of their time managing inventory and manually counting food items at the end of each day in order to record data in the log book. Managers also generate payroll for each employee at the end of each week. Although it is difficult to track popular food items by manually going through the order receipts and counting the number of times an item is ordered, managers need a simple way to get statistics about food items, employee performance, restaurant revenue and average wait times.

Solution - We believe that managers should be able to view statistics about the restaurant more efficiently without doing the tedious work of gathering the information themselves. Our tools will allow the manager use these statistics to make the best business decisions. Whether its to discontinue an unpopular menu item or to speak to an employee about their lack of performance or attitude our tools will provide an accurate representation on how the restaurant is functioning. An inventory system that will monitor incoming orders and adjust the inventory accordingly is ideal for accomplishing these task. It will also allow the manager to adjust the inventory manually whenever they need to. An automated system can send a low stock alert to the manager whenever an item reaches a certain threshold.

Problem- Managing payroll can be difficult for any one person to precisely handle. Usually the Manager is in charge of payroll but he has a lot of other responsibilities to handle beside managing hours. And during stressful/hectic days the manager has a lot of deal with and has a higher chance of making mistakes on an employee's payroll. Not to mention the old fashion way of doing payroll is with paper and pencil. The same usually goes for the calendar schedule of all employee shifts. Managing the calendar can at times be a big hassle especially when shifts need to be moved around and/or updated.

Solution- Technology is the way to go for more a precise and accurate database of information. Using a server will keep track of every employee's work hours that will stored into a database that will be backed up. This means that manager can easily access all this information without having to worry about updating unless he needs to. Using automatic software that updates itself reduces the chance of pencil and paper error alongside saving the manager a lot of time he can use on running the restaurant instead. Using a calendar interface is far more simplistic, easy to use, and cleaner to use and manager. The calendar can easily be shared among workers and offer the ability to allow workers to cover or change shifts in case of emergencies.

Waiter

Problem- In a customer service oriented business, customer satisfaction is the greatest asset. A common problem restaurants face is customers having to wait a long time to be able to speak with their waiters. This is caused by many issues: the wait staff might not be the fastest people around, they are constantly going back and forth to the kitchen so they aren't around to be able to help customers, also they have so many things to do at once so, sometimes they forget that a customer didn't get their food or something. These are all things that affect a restaurant's customer satisfaction rating because no one likes having to wait for their food. When people come to a restaurant, they come hungry. So serving them as soon as possible is essential otherwise customers will be unhappy.

Solution- To fix this, waiters should get some sort of reminder every few minutes to check the tables that they were assigned to. Also, it would be great if waiters could place and check the status of the orders without having to go to the kitchen. Anything to keep the waiters on the floor and out of the kitchen would help.

Problem- Another issue is sometimes waiters don't have the best handwriting. This creates miscommunication between the customers and the chefs. Customers who order one thing and get something different, they would not be happy with that. Also, this significantly decreases profit margins because food is thrown away and more money is spent making another order.

Solution- It would be good if there could be something to make communication easier between the waiters and chefs.

Problem- Sometimes another problem is the wait staff doesn't have the entire menu memorized. So, if a customer has a question about what ingredients are in the food, waiters have to go to the kitchen to ask the chefs. This is a big waste of time, and time is money in any industry. Also, the menu changes from time to time so its difficult to keep track of what is actually on the menu.

Solution- It would be beneficial to have a menu with the ingredients listed so that the waiter wouldn't have to make the extra trip to the kitchen every time a customer has a question.

Problem- Another matter in restaurants is a customer complaining about the wait time after their done eating. At this point customers want to leave as soon as possible and waiters are generally busy taking care of other customers. This process takes a long time. Waiters will come to the customers with their bill and in the mean time take care of some other work. Then they go back to get the payment, take care of some more work, and then finally give the customer a chance to give a tip. The entire process takes anywhere from 5-15 minutes. Also, a lot of times restaurants will get big groups and splitting the bill is a hassle. Sometimes one person gets charged for another person's bill or the split is incorrect. This is very taxing on the waiters who are thinking about the other tables they are waiting on and sometimes customers just walk away without paying. This increasing problem of theft is solely due to the latency between waiters giving the bill and getting the payment from customers.

Solution- If there is a way for the waiters to immediately charge the customers for their tab with a portable credit card reader that would allow the waiters to split the bills that would make it easier for both parties. This would decrease theft because payment would be taken on the spot from the customers.

b.) Glossary of Terms

Technical Terms

Database: the file where the menu items, inventory, scheduling and orders are stored.

Order Queue: a list of orders that are placed in first in, first out order. These orders are sent to the Chef PC when the chef can prepare them.

Inventory System: a system that can be accessed by the manager or the chef, this system is used to manage the inventory of food items.

Walk-in Queue: a list of table reservations that are placed in first in, first out order. These reservations are made on the Customer PC when the customer walks into the restaurant.

Graphical User Interface (GUI) - interface that allows allows easier user communication via pictures and texts

Non-Technical Terms

Customer: Any person that orders an item from the menu or walks into the restaurant.

Waiter: The person that takes the orders of the customers, enters the orders into the system and delivers the orders to the customers.

Manager: The person that is responsible for inventory management, employee scheduling, payroll and customer satisfaction.

Chef: The person that receives the orders on the Chef PC and prepares the food.

Cashier: The person that charges the customers after their meal.

Ingredient: Inventory items that are used to prepare menu item.

2) System Requirements

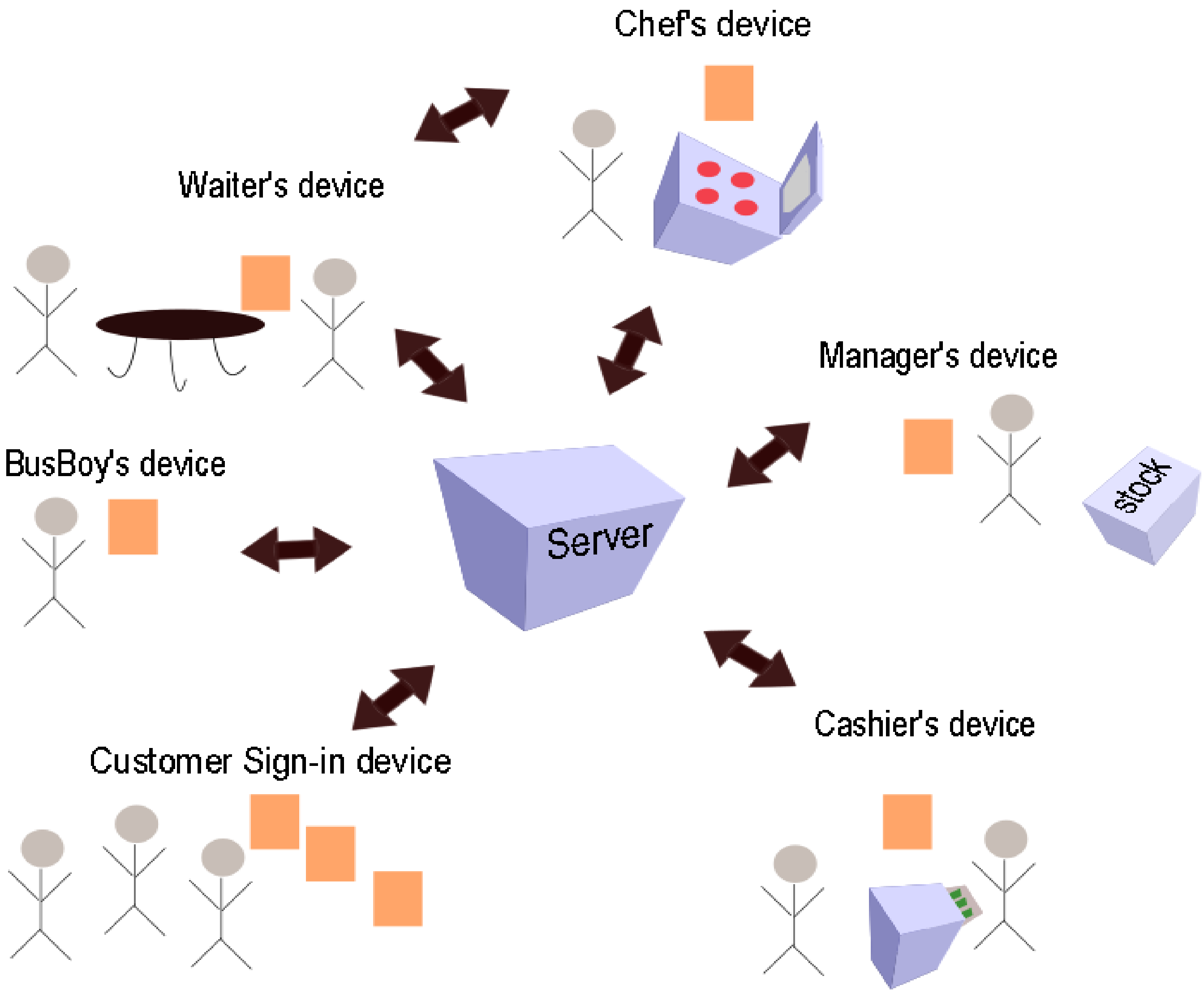


Figure 1.
This is the general flow of how the restaurant will work with the Server. Each peach rectangular object represents a tablet/device that will communicate with the server

a.) Enumerated Functional Requirements

Identifier	PW	Requirement
REQ - 1	1	The customer sign in PC shall keep track of the customers at the table and record the time they've been waiting.
REQ - 2	1	The cashier tablet shall have an option for the customer on what kind of receipt they want (printed copy or email)
REQ - 3	2	The busboy tablet shall notify the busboy what tables to clean.
REQ - 4	4	The chef PC will see customers meals prioritized by order time(when did they order)
REQ - 5	5	The chef PC shall receive orders from waiters
REQ - 6	3	The chef PC must be able to notify the waiter that a meal is ready to be delivered
REQ - 7	2	The chef PC must be able to set cooking timers and be notified when they are done
REQ - 8	4	The chef PC should notify the system to update inventory when a meal is ready to be delivered
REQ - 9	2	The manager PC shall be able to analyze and predict supply usage
REQ - 10	3	The manager PC will keep track of all employee hours for payroll alongside a calendar GUI with everyone's work schedule
REQ - 11	4	The manager PC will dynamically update inventory as orders ready to be delivered. Also, when ordering shipments of supplies, quantity of each supply with be updated.
REQ - 12	1	The manager PC will have statistics on food popularity an

		employee performances.
REQ - 13	4	The manager PC will have the ability to add, edit or remove items from the menu.
REQ - 14	2	The manager's PC shall alert the manager when the inventory has a low stock of a particular item.
REQ - 15	2	The manager's PC shall keep track of daily, weekly, monthly, and annual revenue.
REQ - 16	5	The waiter PC will allow the menu to be viewed with what ingredients are used.
REQ - 17	4	The waiter PC will be able to place separate orders per customer.
REQ - 18	1	The waiter PC will allow bills to be split and paid on the spot with credit card reader.
REQ - 19	1	The waiter PC will send reminders to user to go check on table periodically.
REQ - 20	3	The customer sign in PC will let waiter PC know which customers came first.
REQ - 21	4	The waiter PC must be able to notify the chef PC of any food allergies/special instructions

b.) Enumerated Non-Functional Requirements

Identifier	PW	Requirement
REQ - 22	1	The program should be aesthetically pleasing and meet the standard of the restaurant.
REQ - 23	5	Security measures need to be made to make sure non-users do not login.
REQ - 24	4	Users should read the manual before using the program.
REQ - 25	5	The system should be backed up in case of failures to avoid losing inventory count and orders.

REQ - 26	5	The system should have a low mean time between failures(MTBF)
REQ - 27	2	Time between different screens on the program should be minimized.
REQ - 28	1	The system should work on any type of device or PC.
REQ - 29	3	The system should be easy to use by users that are not technologically advanced.
REQ - 30	4	The system should be easy to debug.
REQ - 31	1	The system should be compatible with the assistance button.
REQ - 32	2	The device each employee uses should be sized based on what would be most convenient to them.

c.) On- Screen Appearance Requirements

Identifier	**PW**	**Requirements**
REQ - 33	4	The system shall display real time table availability.
REQ - 34	2	The system should display average wait time for the next available table.
REQ - 35	5	The system shall display menu items, ingredients and price.
REQ - 36	5	The system shall display the order total price.
REQ - 37	4	The system shall display inventory for the manager and chef.
REQ - 38	1	The system should display employee schedules.
REQ - 39	4	The system shall display order queue for the kitchen.
REQ - 40	3	The system shall display revenue, employee and order statistics for the manager.
REQ - 41	3	The system shall display available payment options.
REQ - 42	2	The system shall display a comment box for allergy/special requests.

3. Functional Requirements

3.1: Stakeholders

Restaurant Owners: The restaurant owners would be most interested in this system because it immediately affects their business and he wants to improve efficiency and customer satisfaction.

Software Designers: The software designers will also be interested since it will be their job to design it the best to their ability and sell it to people who need it.

Customers: The customers are the ones who will rely on the system to order their food and for their food to arrive as quickly as possible while also desiring the best possible restaurant experience

Restaurant Employees: The employees will have an interest in the system because they will rely on it to ease the burden of their responsibilities and help create more efficiency within the restaurant

3.2: Actors and Goals

Initiating Actors

Manager: The employee in charge of the restaurant who must manage the system. The goal of the manager is to make sure the system is up-to-date, properly functioning, keep track of employee information and activities, and manage the restaurant

Waiter: The employee who must attend to customers and service them. The goal of the waiter is to take and deliver the orders to the customers.
Chef: The employee who must cook the food. The goal of the chef is to make the food and maintain ingredients.

Customer: The person who places an order. The goal of the customer is to place their order and pay for their meal.

Customer Sign-In: The employee who checks in customers. The goal of the customer sign in is to seat customers faster and efficiently.
Cashier: The employee who processes checks. The goal of the cashier is to make sure the customer pays their bill in the quickest and most efficient way.

Busboy: The employee who cleans the tables. The goal of the busboy is to clean a table right after a customer leaves and to help the waiters delivering food if necessary.

Participating Actors:

Database: Stores the data of the system. The goal of the database is to update inventory, keep track of employee information, and manage any other data

Timer: Keep track of cooking time for an order. The goal of the timer is to make sure an order/ingredient is cooked properly

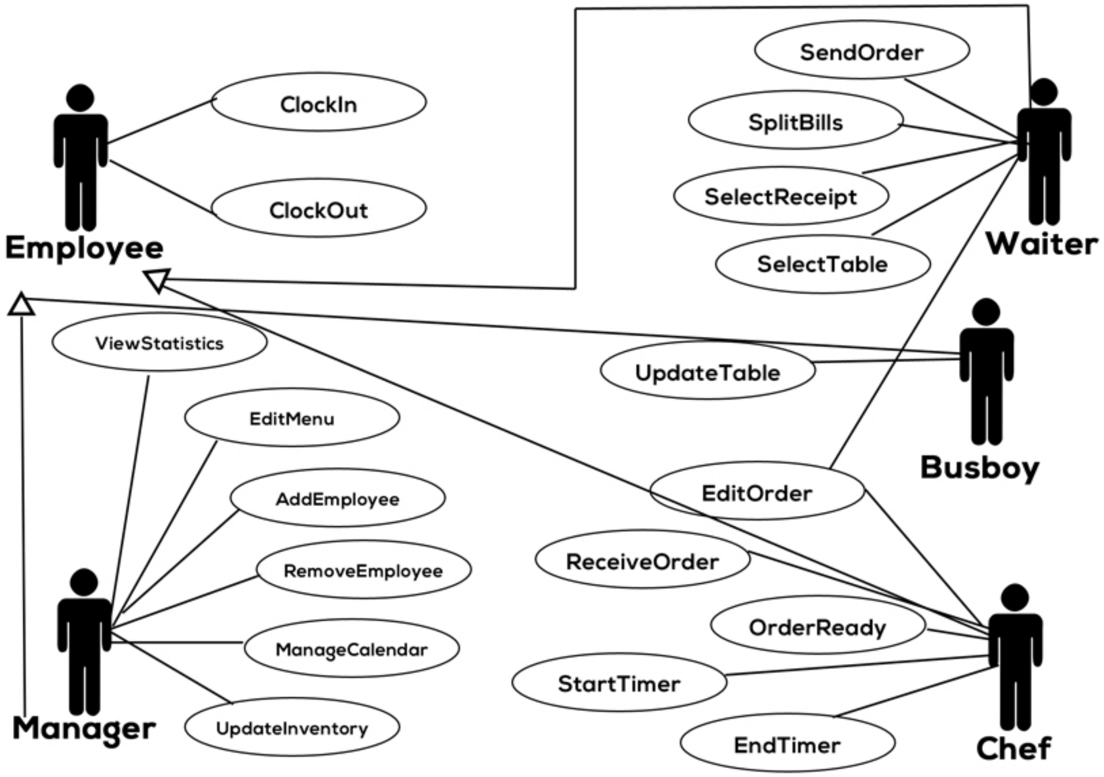
3.3: Use Cases

a) Casual Descriptions

Use Case #	**Use Case Name**	**Description**
UC -1	UpdateTable	The waiters can change the status of all tables to available or not
UC -2	SelectTable	The tables will be shown and customers can select which one to sit
UC -3	SelectReceipt	The option of receipt with either paper or email is given.
UC - 4	StartTimer	The chef will start the timer to cook an order
UC - 5	EndTimer	The timer will end, notifying the chef
UC - 6	ReceiveOrder	The chef receives the order which is automatically prioritized
UC - 7	OrderReady	The order is ready to be delivered to the customer and inventory must also be update
UC - 8	EditOrder	The chef can edit any order or remove them as well
UC - 9	SendOrder	Waiter places customer order to the system
UC - 10	SplitBills	Waiter charges customer

		using a credit card swiper, which can be used to split bills if necessary
UC - 11	ManageCalendar	Manager can make changes to the Calendar GUI in order to move/add/remove shifts
UC - 12	UpdateInventory	Manager can manually update the Inventory system. System will also update inventory as orders are placed and as supplies are ordered.
UC - 13	EditMenu	Manager can modify menu items or add/remove an item to the menu
UC - 14	ViewStatistics	Manager can look at all stats of the restaurant ranging from food popularity to payroll
UC - 15	AddEmployee	Enter a new employee into the system
UC - 16	RemoveEmployee	Remove an employee from the system
UC - 17	ClockIn	Employee clocks into their shift
UC - 18	ClockOut	Employee clocks out of their shift

b) Use Case Diagram



c) Traceability Matrix

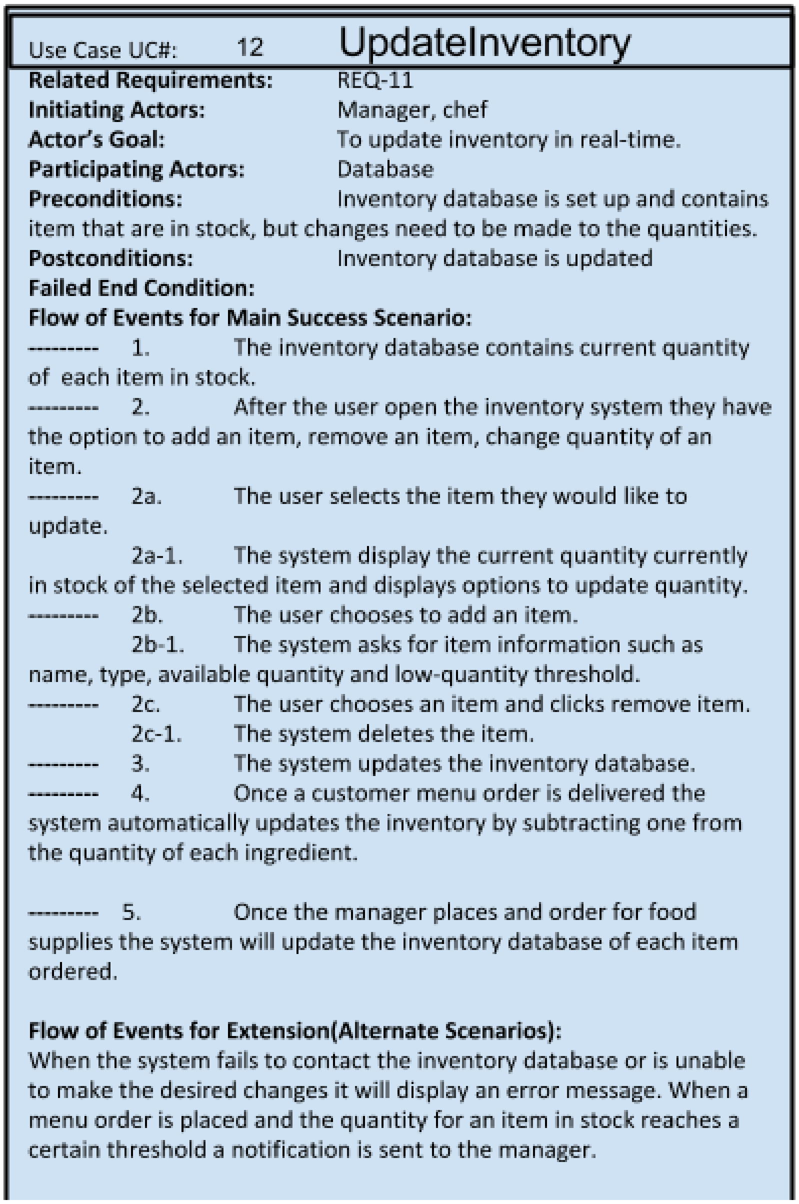
	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16
RQ1	X	X														
RQ2		X	X							X						
RQ3	X	X														

RQ4						x			x							
RQ5						x		x	x							
RQ6							x	x				x				
RQ7				x	x			x								
RQ8								x				x				
RQ9												x	x	x		
RQ10											x				x	x
RQ11												x	x	x		
RQ12											x	x	x	x	x	x
RQ13				x	x			x				x	x	x		
RQ14							x					x	x			
RQ15										x	x	x	x	x	x	x
RQ16								x					x			
RQ17						x			x	x						
RQ18			x							x						
RQ19	x															
RQ20		x														
RQ21						x		x	x							
RQ22											x			x		
RQ23						x	x	x				x	x		x	x
RQ24					x	x	x	x				x	x		x	x
RQ25						x	x		x			x		x		
RQ26					x	x	x	x				x	x			
RQ27							x	x				x	x			
RQ28											x	x	x	x	x	x
RQ29						x	x		x			x		x		
RQ30		x				x	x		x		x	x		x		

RQ31						**X**	**X**	**X**	**X**			**X**		**X**		
RQ32						**X**		**X**	**X**				**X**	**X**		
RQ33	**X**	**X**								**X**						
RQ34	**X**	**X**								**X**						
RQ35						**X**		**X**	**X**				**X**	**X**		
RQ36			**X**			**X**	**X**			**X**				**X**		
RQ37							**X**	**X**				**X**	**X**	**X**		
RQ38											**X**					
RQ39				**X**	**X**	**X**	**X**									
RQ40											**X**			**X**		
RQ41			**X**							**X**				**X**		
RQ42								**X**	**X**			**X**	**X**			
MX PW	**4**	**4**	**5**	**4**	**5**	**5**	**5**	**5**	**5**	**5**	**4**	**5**	**5**	**5**	**5**	**5**

d) Fully Dressed Descriptions with Sequence Diagrams

Manager



Use Case UC#: 12 **UpdateInventory**

Related Requirements: REQ-11

Initiating Actors: Manager, chef

Actor's Goal: To update inventory in real-time.

Participating Actors: Database

Preconditions: Inventory database is set up and contains item that are in stock, but changes need to be made to the quantities.

Postconditions: Inventory database is updated

Failed End Condition:

Flow of Events for Main Success Scenario:

--------- 1. The inventory database contains current quantity of each item in stock.

--------- 2. After the user open the inventory system they have the option to add an item, remove an item, change quantity of an item.

--------- 2a. The user selects the item they would like to update.

2a-1. The system display the current quantity currently in stock of the selected item and displays options to update quantity.

--------- 2b. The user chooses to add an item.

2b-1. The system asks for item information such as name, type, available quantity and low-quantity threshold.

--------- 2c. The user chooses an item and clicks remove item.

2c-1. The system deletes the item.

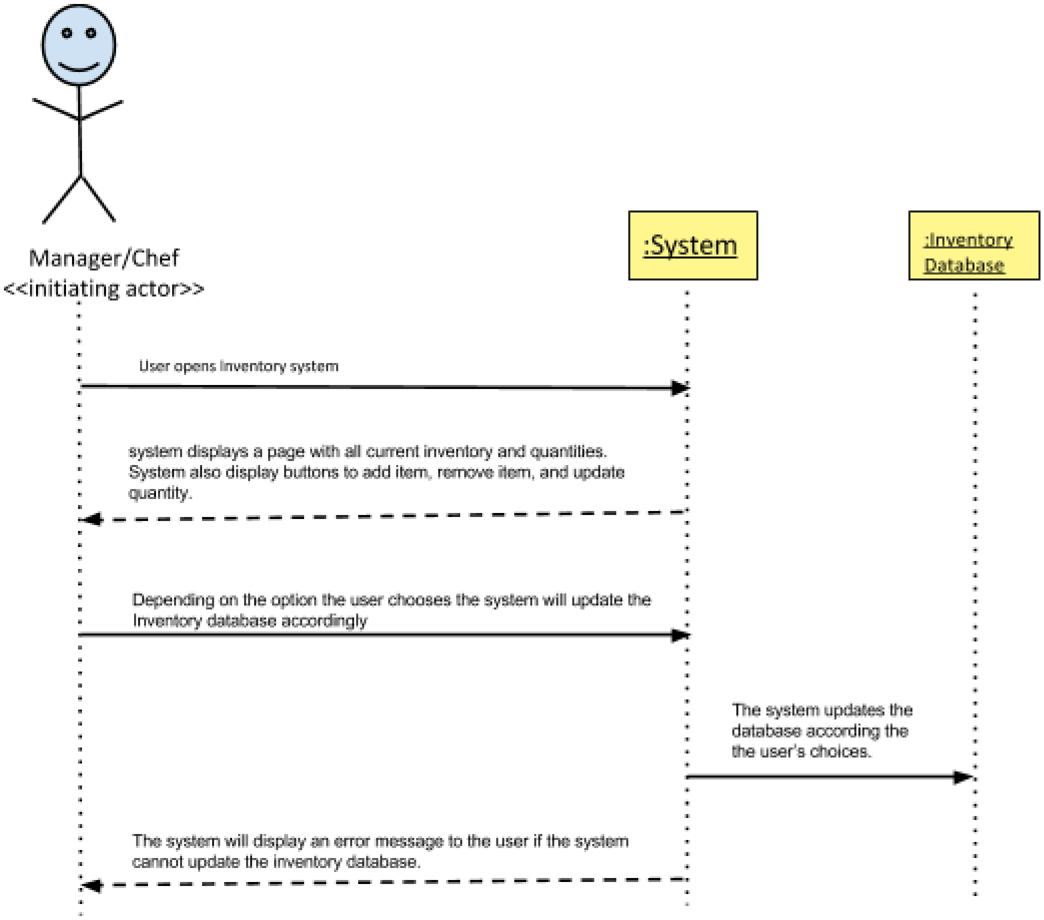
--------- 3. The system updates the inventory database.

--------- 4. Once a customer menu order is delivered the system automatically updates the inventory by subtracting one from the quantity of each ingredient.

--------- 5. Once the manager places and order for food supplies the system will update the inventory database of each item ordered.

Flow of Events for Extension(Alternate Scenarios):

When the system fails to contact the inventory database or is unable to make the desired changes it will display an error message. When a menu order is placed and the quantity for an item in stock reaches a certain threshold a notification is sent to the manager.



User opens Inventory system

system displays a page with all current inventory and quantities. System also display buttons to add item, remove item, and update quantity.

Depending on the option the user chooses the system will update the Inventory database accordingly

The system updates the database according the the user's choices.

The system will display an error message to the user if the system cannot update the inventory database.

Customer Sign-In/Busboy

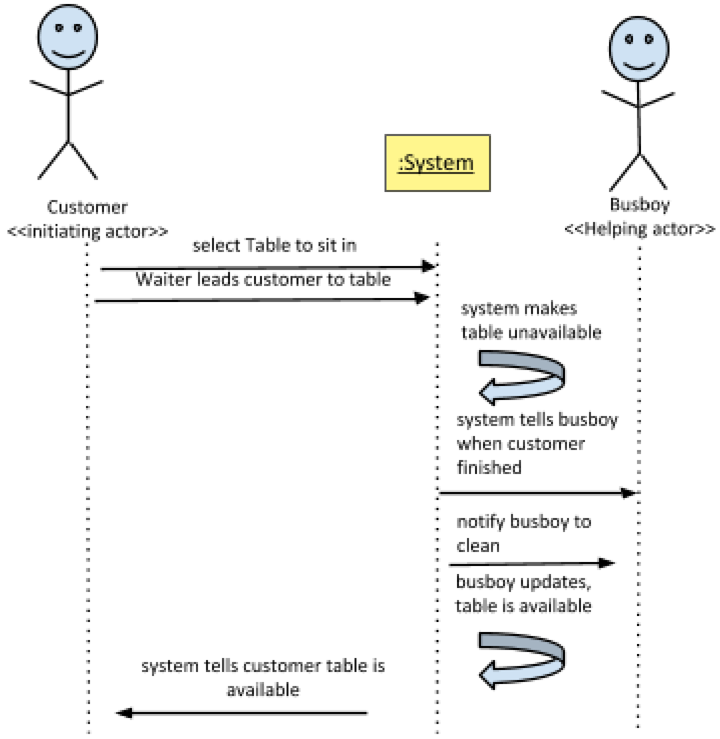
Use Case UC#:	SelectTable
Related Requirements:	Select Table, Update Table
Initiating Actors:	Customer, busboy
Actor's Goal:	To select and update the tables
Participating Actors:	Customer, Waiters, busboys
Preconditions:	Table is ready to be selected
Postconditions:	Table is ready to be cleaned
Failed End Condition:	

Flow of Events for Main Success Scenario:

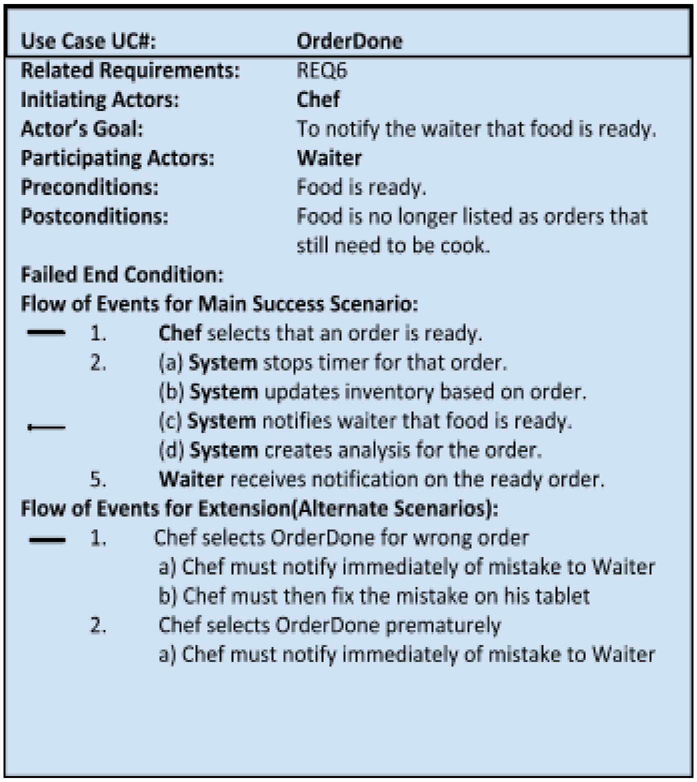
1. Customer chooses table to sit in
2. Waiter leads customer to sit there and table becomes unavailable
3. Customer finishes eating and leaves
4. Busboy cleans the table to prepare for next customer
5. Table becomes clean and busboy notifies system that table is ready to be used again

Flow of Events for Extension(Alternate Scenarios):

Customer selects unavailable table. System notifies customer that they cant sit there. Customer is forced to wait or select another table.


:System
Customer
<<initiating actor>>
Busboy
<<Helping actor>>
select Table to sit in
Waiter leads customer to table
system makes table unavailable
system tells busboy when customer finished
notify busboy to clean
busboy updates, table is available
system tells customer table is available

Chef



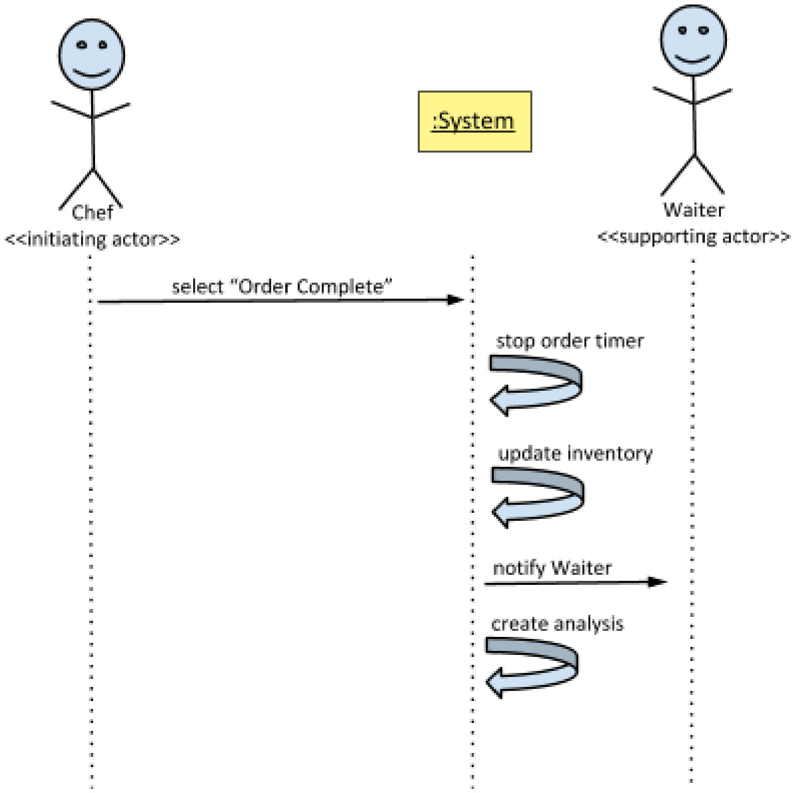
Use Case UC#:	**OrderDone**
Related Requirements:	REQ6
Initiating Actors:	**Chef**
Actor's Goal:	To notify the waiter that food is ready.
Participating Actors:	**Waiter**
Preconditions:	Food is ready.
Postconditions:	Food is no longer listed as orders that still need to be cook.
Failed End Condition:	

Flow of Events for Main Success Scenario:

— 1. **Chef** selects that an order is ready.
2. (a) **System** stops timer for that order.
(b) **System** updates inventory based on order.
← (c) **System** notifies waiter that food is ready.
(d) **System** creates analysis for the order.
5. **Waiter** receives notification on the ready order.

Flow of Events for Extension(Alternate Scenarios):

— 1. Chef selects OrderDone for wrong order
a) Chef must notify immediately of mistake to Waiter
b) Chef must then fix the mistake on his tablet
2. Chef selects OrderDone prematurely
a) Chef must notify immediately of mistake to Waiter



Waiter

Use Case UC#: SendOrder

Related Requirements: REQ-17

Initiating Actors: Waiter

Actor's Goal: To notify the chef to make the food.

Participating Actors: **Chef**

Preconditions: Food has to be on menu.

Successful End Conditions: Order has been sent to Chef's PC

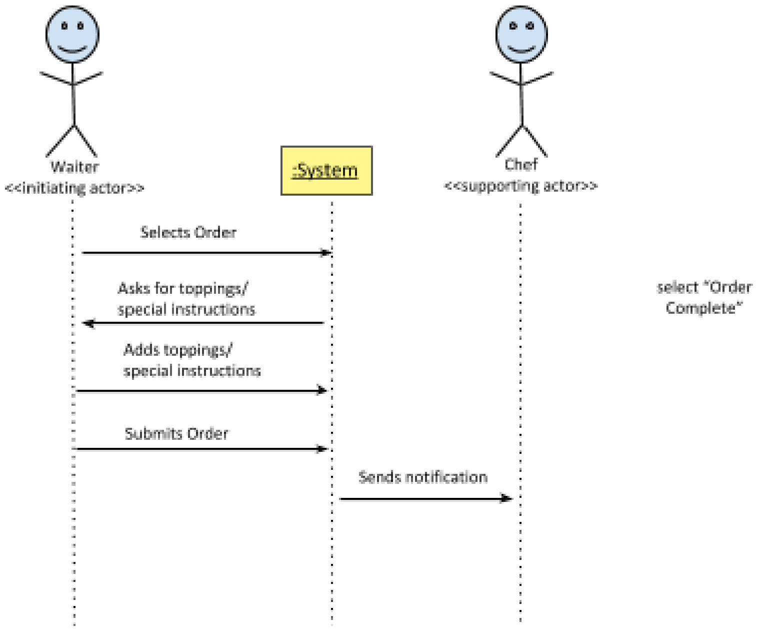
Failed End Condition: Order did not get sent to Chef's PC and waiter is notified of failure.

Flow of Events for Main Success Scenario:

— 1. **Waiter** selects what customer wants to order.
— 2. **System** asks for any extra toppings and/or special instruction.
— 3. **Waiter** adds extra toppings and/or special instruction.
— 4. **Waiter** submits order.
— 5. **System** sends notification of order to **Chef.**

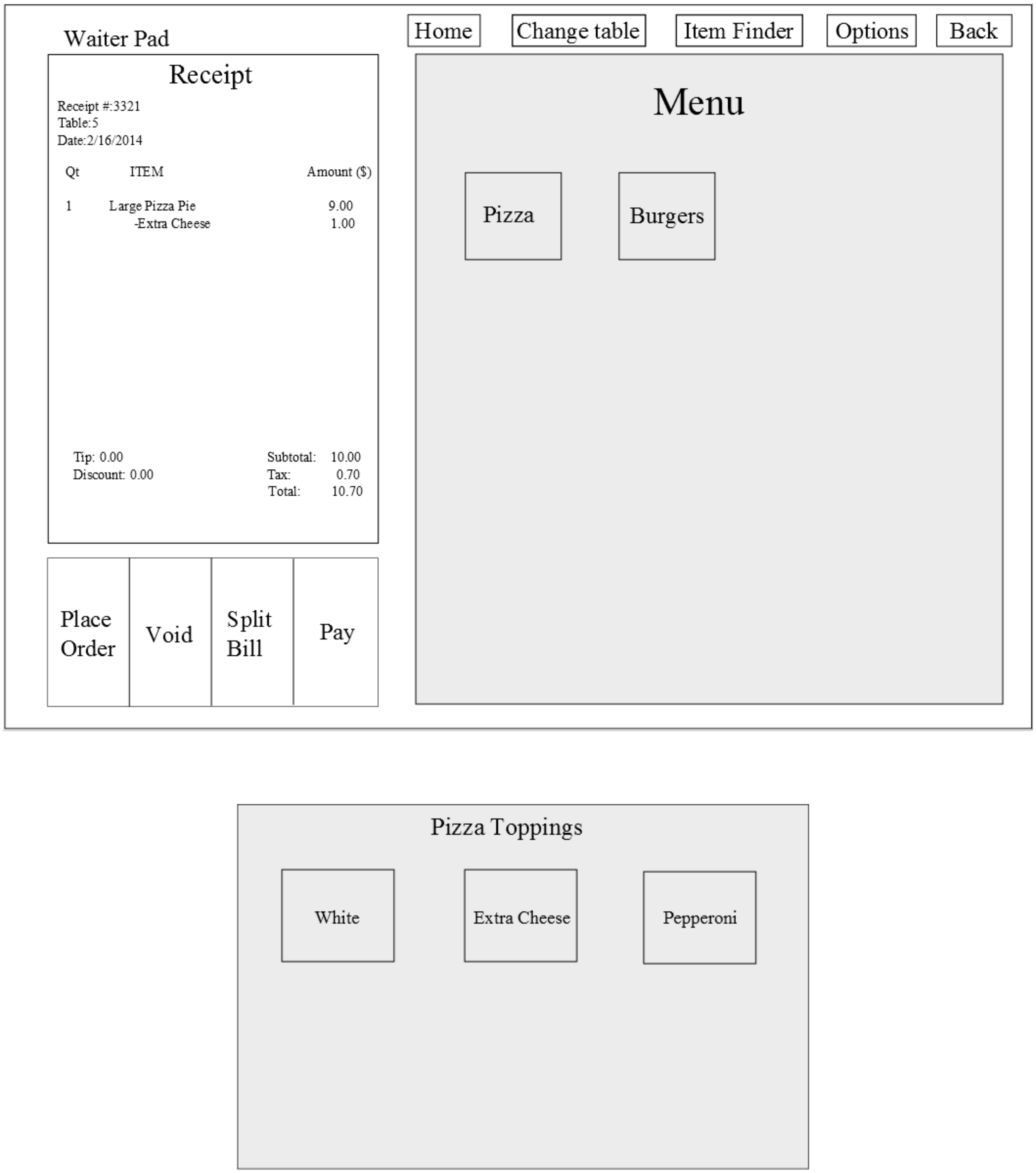
Flow of Events for Extension(Alternate Scenarios):

1a. **Waiter** selects what customer wants to order, but is no longer in **System**.
— 1. Waiter requests customer to make a different selection.



4. User Interface Specification

4.1 Preliminary Design(draft images)



Receipt #:3321
Table:5
Date:2/16/2014

Qt	ITEM	Amount ($)
1	Large Pizza Pie	9.00
	-Extra Cheese	1.00

Tip: 0.00
Discount: 0.00

Subtotal: 10.00
Tax: 0.70
Total: 10.70

Figure 1
Image of the waiter's GUI(graphical user interface) and the image below is the pop up interface if the user would hit the pizza option on the menu.

We are going to look at the Use case #9 *SendOrder* from the perspective of the waiter as a user. In figure 1 above**(Page25)**, after the waiter gets an order from the customer he can use the Menu section to navigate to a specific item and what toppings they would like. In this particular case the customer choose to order a 'Large Pizza Pie' and requested 'Extra Cheese.' So using the menu the user would navigate to the 'Pizza' item that will invoke a pop up that will have all of the available toppings. After Selecting a topping the receipt section will automatically update to show a draft of the Receipt which can be used for the user to easily repeat back or edit the order. When the order is ready to be placed, the user can 'Place Order' which will invoke the *SendOrder* function. This will send a signal to the server and then back to the chef.

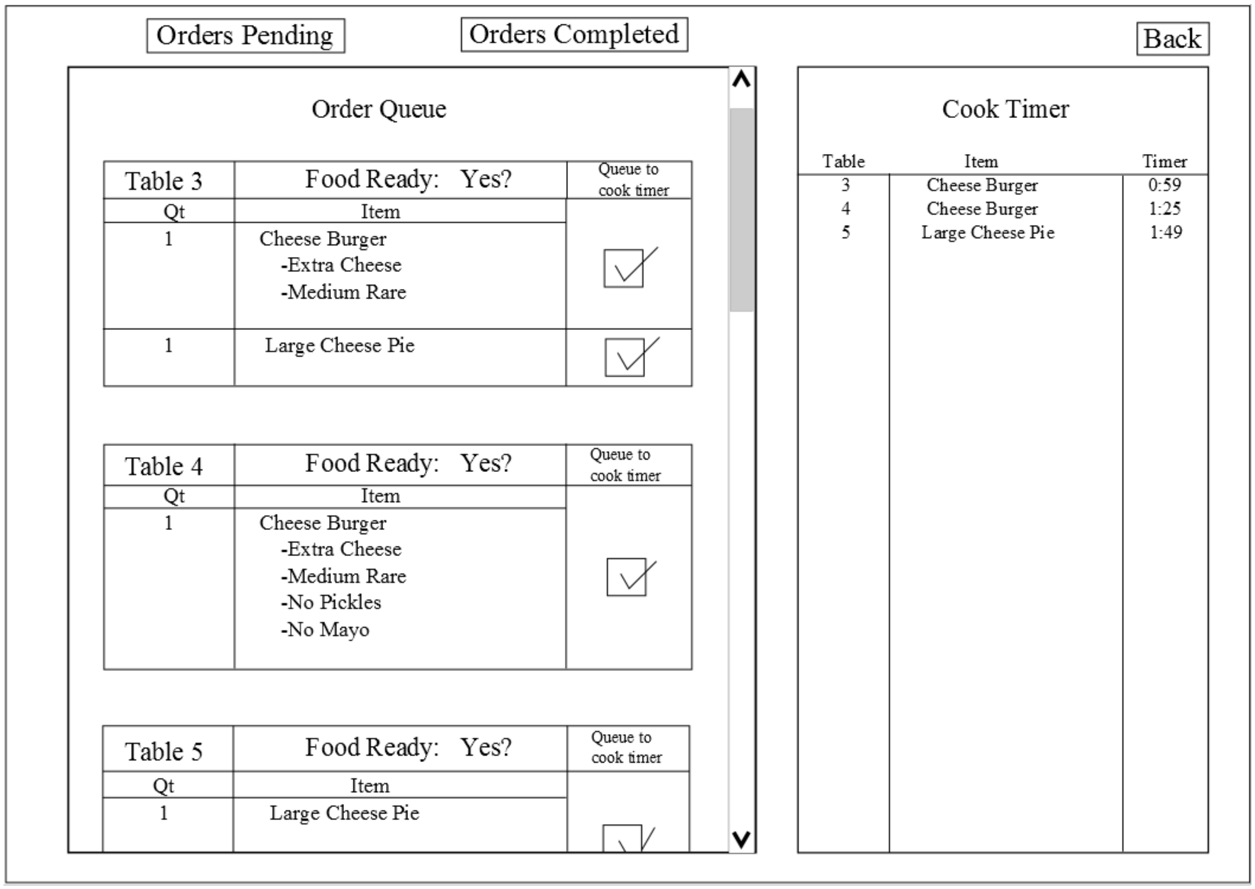


Table 3	Food Ready: Yes?	Queue to cook timer
Qt	Item	
1	Cheese Burger -Extra Cheese -Medium Rare	☑
1	Large Cheese Pie	☑

Table 4	Food Ready: Yes?	Queue to cook timer
Qt	Item	
1	Cheese Burger -Extra Cheese -Medium Rare -No Pickles -No Mayo	☑

Table 5	Food Ready: Yes?	Queue to cook timer
Qt	Item	
1	Large Cheese Pie	☑

Table	Item	Timer
3	Cheese Burger	0:59
4	Cheese Burger	1:25
5	Large Cheese Pie	1:49

Figure 2
Image of the Chef's GUI and all his/her incoming orders.

The Chef will be receiving automatic updates of new orders once the system receives *SendOrder.* In figure 2 above, the interface will automatically update itself on new incoming orders based off of Use Case #6 *RecieveOrder*(more detail later). Here the Chef can Queue in items to the Cook Timer once prepared and ready to cook. After the chef feels that the order is ready they can use the 'Food Ready: Yes?' button in order to invoke Use Case #7 *OrderReady.* The Waiter's pad will receive a pop up notification with the table # in which the food is ready. After the order has been sent the interface will update and move the order into the 'Orders Completed' tab.

4.2 User Effort Estimation:

Scenario 1:

Customer Sign in PC

1. Customer wanting to select a table to sit in. (2 taps)
A. Customer selects the table they want
B. Customer taps choose table.

2. Busboy updating the table. (2 taps)
A. Busboy selects the table
B. Busboy taps table is available

Scenario 2:

Manager Tablet

1. Manager ordering supplies (5 taps)
A. Manager selects inventory section
B. Manager selects item
C. Manager selects order
D. Manager picks quantity
E. Manager confirms order

2. Manager deleting an item (4 taps)
A. Manager selects inventory section
B. Manager selects an item
C. Manager selects delete
D. Manager confirms the deletion

3. Manager adding an item (4 taps)
A. Manager selects inventory section
B. Manager selects add item
C. Manager types in the item
D. Manager confirms the addition

4. Manager adding an employee (multiple taps)
A. Manager selects employee list
B. Manager selects add employee
C. Manager fills out information about employee
D. Manager confirms the addition

5. Manager deleting an employee (3 taps)
A. Manager selects employee list
B. Manager clicks on employee that he/she wants to delete
C. manager confirms the deletion

Scenario 3:

Waiter Tablet

1. Waiter splitting the bill (3 taps)
A. Waiter selects table(if not already on the current table)
B. Waiter selects split bill
C. Waiter selects how many people to split bill for

2. Waiter sending the order (multiple taps)
A. Waiter selects table(if not already on the current table)
B. Waiter selects food/drink items depending on what the customer asks
C. Waiter adds/removes toppings from item
D. Waiter confirms the order

3. Waiter selects table where customer is sitting (1 tap)
A. Waiter selects table

4. Waiter selects receipt (4 taps)
A. Waiter selects table(if not already on the current table)
B. Waiter selects pay and charges all customers on respective table
C. Waiter gives receipt option in form of paper or email
D. Waiter confirms the option

5. Waiter cancels current order (3 taps)
A. Waiter selects table(if not already on the current table)
B. Waiter selects Void
C. Waiter selects Yes on 'Are you sure?' pop up

6. Waiter quick search (food/drink)items (2 taps)
A. Waiter selects quick search and can type in any item
B. Item will appear in menu section where he can select the item

7. Waiter logs out/clocks out
A. Waiter selects option
B. waiter can choose whether he wants to clock out or log out

Scenario 4:

Any Employee Tablet (hitting the 'home' or 'back key)

1. Employee clocks in the system (2 tap)
A. Employee clicks clock in
B. Employee enters his/her own specific code
C. Employee presses done

2. Employee clocks out of the system (1 tap)
A. Employee clicks clock out and enters code
B. Employee enters his/her own specific code
C. Employee presses done

3. Employee can check table layout (1 tap) if privileged user

A. Employee clicks table icon

Scenario 5:

Chef Tablet

1. Chef selects item that is ready to be queued for cook timer (1 taps)

A. Chef toggles a check mark button next to the item that is ready

2. Chef selects order ready (1 tap)

A. Chef selects Food Ready: Yes? when current meal is ready for the table

3. Chef switches between Order Pending or Order Complete (1 tap)

A. Chef can hit the corresponding tab labeled ‘Order Pending’ or ‘Order Complete

Scenario 6:

Cashier Tablet

1. Customer wants to pay (4-5 taps)

A. Customer selects pay transaction
B. Waiter hits pay button
C. Select cash or credit
D. Select paper or email receipt then submit
E. (if email route) enter email then submit

5. Domain Analysis

a) Domain Model

i) Concept Definitions

Responsibility	Type	Concept
R-01: Prompt customer to choose what table they would like to sit at	D	Table
R-02: Update status of table (Available, Not Available, Dirty)	D	Table
R-03: Prompt waiter to go to customer’s table, give menus	K	Menu

R-04: Prompt waiter to take customer’s order and send to Chef’s Tablet	K	Order
R-05: Notification telling waiter to check on customer’s table	K	customerNotification
R-06: Notification telling waiter to pick up customer’s order from kitchen	K	Order
R-07: Update status of customer’s order (Ready or Not Ready)	D	Order
R-08: Notify Chef of customer’s order	K	Order
R-09: Notification to manager when items are: almost finished, almost expired, finished, and expired	K	Inventory
R-10: Display list of inventory (including quantity)		Inventory
R-11: Change Inventory (add, remove, modify expiration date)	D	Inventory
R-12: Prompt waiter to get payment from customers including if customer wants to split bill	K	Payment
R-13: Prompt waiter to ask customer if they want receipt printed or emailed	K	Payment
R-14: Displays employee information (position, status, wage, contact info, add, remove) as well as status (Working or Off)		EmployeeInfo
R-15: Change employee information (position, status, wage, contact info, add, remove)	D	EmployeeInfo
R-16: Change menu (add, remove, modify)	D	Menu
R-17: Display Statistics		Statistics
R-18: Employees can either log in or out of the system	D	Clock
R-19: System that the waiter uses as the liaison between the customer and the chef	D	WaiterTablet
R-20: System that customers choose their table with	D	CustomerSigninTablet

R-21: System that the manager uses which oversees everything happening in the store	D	ManagerTablet
R-22: System that chef uses for all food updates	D	ChefTablet
R-23: System that the Busboy uses for all table updates	D	BusBoyTablet

ii) Association Definitions

Concept pair	Association Description	Association Name
CustomerSigninTablet < - > Table	CustomerSigninTablet updates the table based on the Table's availability	Updates the table status
WaiterTablet < - > Table	Table updates the WaiterTablet and prompts waiter to get menus for customer	Gives customers menu
WaiterTablet < - > customerNotification	customerNotification sends notification to WaiterTablet to go to table and place order	Placing order
ChefTablet < - > Order	Order sends a notification of order to ChefTablet	Sends Order
WaiterTablet < -> Order	Order sends a notification to WaiterTablet to pick up food from kitchen	Notifies waiter to pick up food
WaiterTablet < - > customerNotification	customerNotification sends a notification to WaiterTablet to check on table, give bill, and get payment	Pay bill
WaiterTablet < - > Payment	Payment sends notification to WaiterTablet for the waiter to ask the customer if they want a paper receipt or e-receipt using	Print Receipt
BusBoyTablet < - > Payment	Payment updates BusBoyTablet that	Marks table as

	the table is dirty after customer pays	dirty
ManagerTablet < - > Inventory	ManagerTablet uses Inventory to add, remove, and modify expiration date of inventory using Inventory	Updates Inventory
ManagerTablet < - > Inventory	Inventory sends a notification to MangerTablet based on status (almost expired, almost finished, expired, and finished) using inventoryNotification	Notifies manager about inventory
ManagerTablet < - > EmployeeInfo	ManagerTablet uses EmployeeInfo to update employee information (position, status, wage, contact info, add, remove) using updateEmployeeInfo	Updates employee information
WaiterTablet < - > Menu	WaiterTablet can display ingredients and modify customer's orders on Menu	update/display menu
ManagerTablet < - > Statistics	ManagerTablet uses Statistics to display item popularity, wait time, and dine in time	displays statistics of store
ManagerTablet < - > Clock	For use of the ManagerTablet, Manager uses his specific employee number and Clocks in or out	Clock in or out
ChefTablet < - > Clock	For use of the ChefTablet, Chef uses his specific employee number and Clocks in or out	Clock in or out
BusBoyTablet < - > Clock	For use of the BusBoyTablet, Busboy uses his specific employee number and Clocks in or out	Clock in or out
WaiterTablet < - > Clock	For use of the WaiterTablet, Waiter uses his specific employee number and Clocks in or out	Clock in or out

iii) Attribute Definitions

Concept	**Attribute**	**Description**
1. Table	selectTable	Where the customer sits and eats
2. Sit in Tablet	OrderQueue getFood	requests customers send in Subsistence customers consume
3. Employee	OrderQueue	Worker at the restaurant
4. Order	getFood OrderQueue	Items ordered by customer Order is added to the kitchen’s order queue
5. Notification	TabletAlert	Alert that is sent from customer’s table to waiter’s PC
6. Notification	KitchenAlert	Alert that is sent from the kitchen the waiter’s PC
foodStatus		
7. order	OrderQueue	Order is added to the kitchen’s order queue
8. inventory	LowStockNotification	The system notifies the manager about low stock items.
9. Inventory	InventoryCheck	The system queries the inventory database to check on an item’s stock.
10. Inventory	StockItems	Items in stock are added, removed or updated in the inventory database.
11. Payment	OrderTotal	Displays the order total for the waiter

	PaymentType	Displays accepted payment types
12. Receipt	RecieptType	Provide an option to print a paper receipt or email the receipt to the customer or both.
13. Employee	EmployeeName	Stores employee name in payroll database
	EmployeeID	Stores employee ID in payroll database
	EmployeeAddress	Stores employee address in payroll database
	EmployeeTitle	Allows the manager to update an employee’s title such as, waiter, busboy, chef, manager.
14. updateEmployee	EmployeeName	Allows manager to update employee name.
	EmployeeID	Allows manager to update employee ID.
	EmployeeAddress	Allows manager to update employee address.
15. Menu	AddItem	Add an item to the menu
	RemoveItem	Remove an item from the menu
	SuspendItem	Temporary suspend item from menu due to an out of stock ingredient.
	Ingredients	Update an item’s ingredients on the menu.
16. Statistics	ItemPopularity	The number of times an item was ordered.

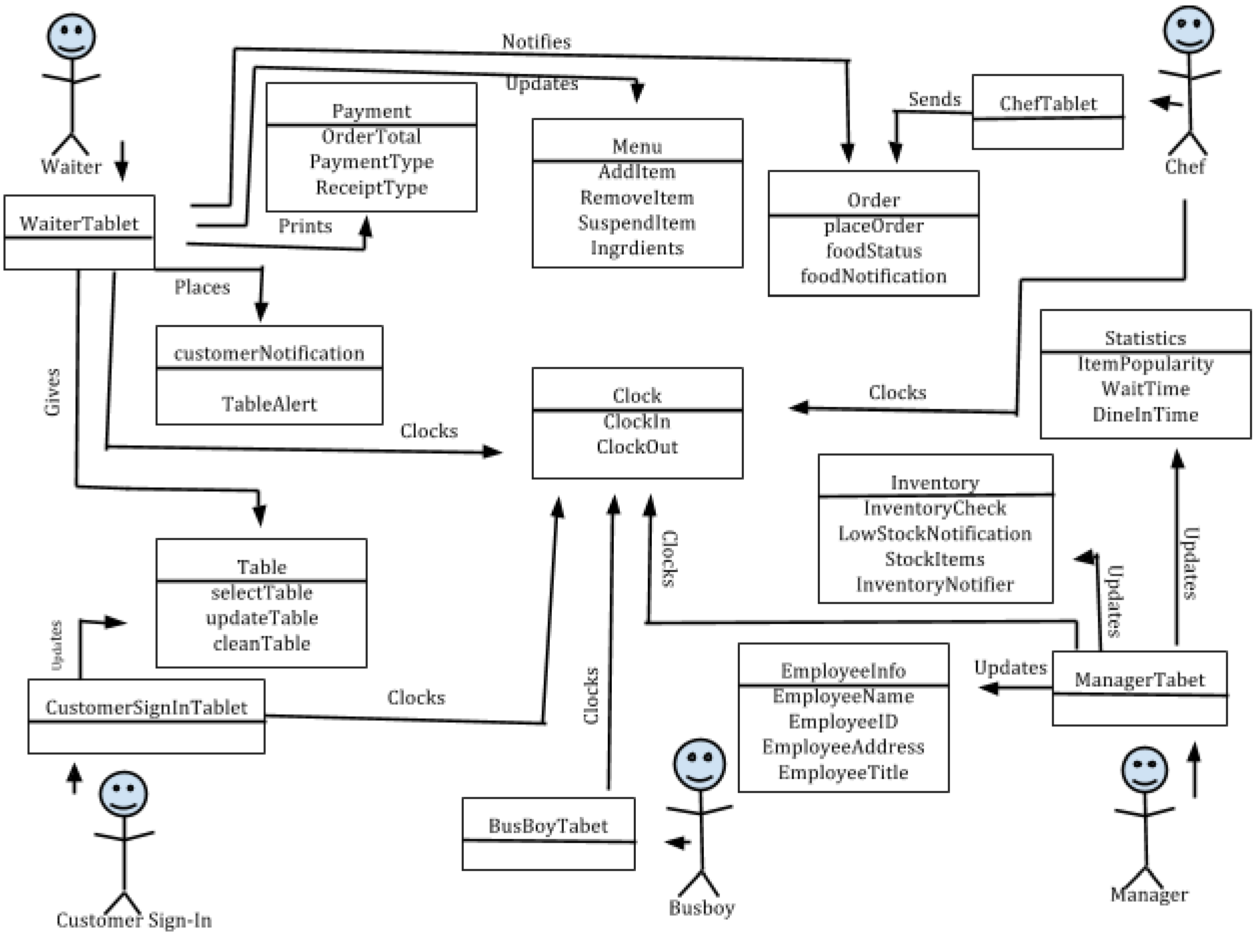
	WaitTime DineInTime	An average wait time for each party to be seated. The average time each party stays at the restaurant.
17. Clock	ClockIn ClockOut	when workers start their shift when workers end their shift

iv) Traceability Matrix

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17
UC-1		X	X														
UC-2	X																
UC-3												X					
UC-4								X									
UC-5						X											
UC-6						X		X									
UC-7						X											
UC-8				X													
UC-9				X													
UC-10												X					
UC-11													X				
UC-12									X	X	X						

UC-13														X		
UC-14															X	
UC-15													X			
UC-16													X			
UC-17																X
UC -18																X

v) Domain Model Diagram



b) System Contracts

Name:	Update Inventory
Responsibilities:	To update inventory in real- time
Use Case:	UC- 12
Exceptions:	none
Preconditions:	Inventory database is already created and already inputted with dat Each item has either 0 or n amount.
Postconditions:	Inventory either increments or decrements the quantity of item(s). Inventory will notify if there is an item out of stock

Name:	Select Table
Responsibilities:	To select and update tables to either sit customers or to be cleaned
Use Case:	UC- 2
Exceptions:	none
Preconditions:	Table must be empty, either before cleaning or after cleaning
Postconditions:	Table is ready to be cleaned or table can now sit customers

Name:	Send Order
Responsibilities:	To notify chef to make the meal for the customer
Use Case:	UC- 9
Exceptions:	None
Preconditions:	All items to be ordered are from the menu. Customer(s) notifies waiter what to order and waiter creates the order(s)
Postconditions:	Chef receives the orders in the order that they were made

Name:	Order Ready
Responsibilities:	To notify the waiter that the meal is ready to be delivered to the customer and to automatically update inventory

Use Case:	UC- 7
Exceptions:	None
Preconditions:	Food is ready to be delivered. Inventory is ready to be updated
Postconditions:	Meal is no longer listed as an order to be cooked. Inventory is updated

c) Mathematical Models

Chef: Algorithm for Orders

Orders must be cooked based on the time they come in. Customers who order their food first should receive their food before customers who order later. Also orders from a single table should come out at the same time. In order to do this, orders should be placed into a queue where a single tables orders are placed into one location in the queue. These table orders should then be placed into a sub-queue for just the table. New orders get placed at the end of the queue.

```
while ( queue is not empty) {
    if (table order is at the beginning of queue)
        place table orders into a sub-queue based on longest cooking time
    while( table order is not complete) {
        start cooking timer for order in the front of the sub-queue
        start cooking order in the front of the sub-queue
        if (chef is idle)
            start cooking next order in the sub-queue
        if (order is complete)
            end cooking timer for the order in the sub-queue
            remove item from sub-queue and start next order in the sub-queue
    }
    remove table order from the beginning of the queue and start next table order
}
```

“Chef is idle” - The chef is able to start another order because the current order is being cooked and does not require the chef’s attention. This can happen more than once per order so the chef should be able to see the list of the all items in the sub-queue until the order is completed.

Customer sitting: Order to sit customers and clear tables

Customers are sat down based on first come. The customer will be able to select any available table they choose. Once selected, the table will become unavailable to other customers. Customers and waiters will see the real time status of all the tables. Once the customer is finished, the waiter or busboy will update the system to show that the table is available again.

```
print (array{all tables})
        if( table has no customer )
                show table as green and available
        else( table has customer )
                show table as red and unavailable

if (customer chooses table)
        change table to red and unavailable
if (customer leaves table)
        change table to green and available
```

Manager Inventory prediction (I Will paraphrase and reference before 3pm 2/23)

For the manager inventory prediction algorithm, we decided to use what Group #1 from Spring 2013 semester started. Algorithms are difficult to formulate and much harder to perfect as it can be an ongoing process even after the system is done. So as a team we are going to try and improve on what last year's team did.

The predictions will be based off of a recursive algorithm that will estimate ingredient usage for foods. We can say that 'i' will be ingredients for food on any day 'n' which will be based off of estimated previous usage(day n-1,n-2,…,n-7) and weekly usage (week w-1,w-2,w-3,w-4). Going with a four week schedule helps to better predict popular seasons such as fall, winter, spring, and summer since the popularity of items will fluctuate.

For predictions of daily ingredient usage, every time an order is placed it will tally up the popularity of ingredient 'i' until the end of the day 'n'. Weekly tallies of ingredients will be based on what the manager decided to manually order at the end of each week 'w.' These prediction will take some time (about 1-2 months) to begin working properly as the system will need some data to predict off of.

Below is an idea of how the prediction will be calculated based off of last year's formula that we will be looking to improve. The daily usage will take into account the day you are on over the average of all the other days to give the manager an estimate. The weekly average will be based off of what the manager ordered and will be estimated over month periods.

Day n proportional usage:

$$\frac{U(n-7)}{(U(n-1)+U(n-2)+U(n-3)+U(n-4)+U(n-5)+U(n-6)+U(n-7))}$$

Weekly average :

$$\frac{U(w-1)+U(w-2)+U(w-3)+U(w-4)}{4}$$

Ui(n) =

$$\frac{U(n-7)}{(U(n-1)+U(n-2)+U(n-3)+U(n-4)+U(n-5)+U(n-6)+U(n-7))} * \frac{U(w-1)+U(w-2)+U(w-3)+U(w-4)}{4}$$

Note* U(week) is the actual weekly usage taken from the inventory system's data, and NOT a value returned by the recursive function.

6. Plan of Work:

Task	Start Date	Deadline	Estimated time(days)	Responsibility
Part 1 (Interaction Diagrams)		**3/2/2014**		
a. Interaction Diagrams	2/25/2014	3/1/2014	4D	ALL
b. Describe bubble diagram	2/25/2014	3/1/2014	4D	ALL
Part 2 (Class Diagram and System Architecture)		**3/9/2014**		
a.Class Diagram	3/4/2014	3/6/2014	2D	MG
b. Data Types and Operation Signatures	3/4/2014	3/6/2014	2D	AP
c. Traceability Matrix	3/4/2014	3/6/2014	2D	NT
System Architecture				
a. Architectural Styles	3/4/2014	3/6/2014	2D	ALL
b. Identifying Subsystems	3/4/2014	3/6/2014	2D	JC
c.Mapping Subsystems to Hardware	3/4/2014	3/6/2014	2D	AA
d.Persistent Data Storage	3/4/2014	3/6/2014	2D	MG

e.Network Protocol	3/4/2014	3/6/2014	2D	NT,AP
f.Global Control Flow	3/4/2014	3/6/2014	2D	JC
g.Hardware Requirements	3/4/2014	3/6/2014	2D	CY
Part 3		**3/9/2014**		
a.Algorithms	3/5/2014	3/6/2014	1D	ALL
b.Data Structures	3/6/2014	3/7/2014	1D	AA,DU
c.User Interface Design and Implementation	3/6/2014	3/7/2014	1D	DU,CY
d.Design of Tests	3/6/2014	3/7/2014	1D	ALL
f. Merge Project	3/7/2014	3/8/2014	1D	ALL
Full Report# 2	3/11/2014	**3/15/2014**	3D	ALL
First Demo	3/16/2014	**3/28/2014**	10D	ALL
Reflective essay	3/29/2014	**4/1/2014**	1D	ALL
Part 1 (Sections)	3/30/2014	**4/4/2014**	3D	ALL
Second Demo	4/5/2014	**4/7/2014**	2D	ALL
Full Report# 3 (rest of sections)	4/5/2014	**4/8/2014**	2D	ALL

AA= Amgad Armanus
JC= Jake Chou
MG= Mitul Gada
AP= Avni Patel
NT= Nirjan Thayaparan
DU= Diego Urquiza
CY= Christian Youssef
ALL= all team members

7. References:

"Concepts: Requirements." *Concepts: Requirements*. Polytechnique Montreal, 2012. Web. 5 Feb. 2014. -(Used for Non-Functional Requirements)

Nick Leshi. (2010). *Good Restaurants Come and Go.* Available: http://open.salon.com/blog/kikstad/2010/06/25/good_restaurants_come_and_go . Last accessed 8th Feb 2014. -(Used for Cover picture).

Group#1 Spring 2013. (2013). Inventory usage rate estimation and run-out date estimation. *Auto-Serve*. 1 (2), 75-76.