Report #1 - Final: Restaurant Automation

SpeedByte

(https://github.com/Cpawel/SpeedByte)

Group # 15:
Jimmy Jorge
Tim Gilligan
Jose Figueroa
Paweł Derkacz
Husen Valikrimwala
Ramaseshan Parthasarathy
Vamshikrishnan Balakrishnan

Individual Contributions Breakdown:

	Project Category	**Team Member Name**						
		Jimmy	**Tim**	**Jose**	**Pawel**	**Husen**	**Ram**	**Vam**
Responsibility Levels	***Project Management***	*80%*	*10%*				*10%*	
	Sec 1. ***Customer Statement of Requirements***	*10%*	*30%*		*50%*			*10%*
	Sec 2. ***System Requirements***		*20%*			*70%*		*10%*
	Sec 3. ***Functional Requirements Specification***	*30%*		*20%*		*40%*	*10%*	
	Sec 4. ***User Interface Specification***	*10%*	*90%*					
	Sec 5. ***Domain Analysis***	*10%*		*35%*		*20%*		*35%*
	Sec 6. ***Plan of Work***	*10%*			*35%*	*20%*	*35%*	

Table of Contents:

1. Customer Statement of Requirements (CSR):

Part a.) Problem Statement

Actor - Host/Hostess:

By being the first person the customer sees, after coming through the front door, there is a paramount of importance placed on hospitality that is served by the host. As a host, I have to try to make the customer feel welcome, and have them seated in a calm and collected manner. However, this feat can be more difficult than it sounds when factoring a large influx of customers and limited seating, along with the confusion of people moving about. If there was a program that could keep track of the open seats, and where they were located, I could serve newly entering customers much faster. What would be even better is if the customer had reserved a seat for themselves prior to arrival. That way I wouldn't even have to search for an open seat, and all I would have to do is confirm that they did indeed reserve a seat and guide them to it.

The SpeedByte solution: SpeedByte allows the customer to select their own table before even stepping foot inside the restaurant. The process is simple, the customer will select their desired time and table. If available, the customer will reserve said table at said time. Upon arrival the customer will simply give their name and the hostess will see when and where to seat them.

Actor - Manager:

Running a business is not easy by any means. If there is anything that could help minimize the amount I have to spread myself across tasks, and help me keep the business profitable, then I'm all ears. I was personally thinking about something that can help the restaurant keep track of the flow of customers, and perhaps what they ordered. That way I know when to keep extra staff on hand, and what to order more of in order to keep a stocked inventory.

That being said, there is also the issue of my having to keep tabs on my employee, to make sure that they are working in a timely manner. I hear that you have some kind of table tracker that you might be able to implement. Well, if you could add in something that keeps track of how long it takes the busboy to clean off the tables and how long it takes for the guests to be seated along with being served, that would make my day. Saves me some time from having to look over the camera records to make sure there isn’t any major slacking.

However, while all of the aforementioned is nice and dandy, I really don't want to have to juggle around another program. Sometimes I get so lost when I have to sift through so many different programs in order to manage payrolls, inventory, shift management, expenditures, et cetera. If there is any way you lot can combine all of these tools for me in one compact program, I would be eternally grateful.

The SpeedByte solution: The overall design of SpeedByte keeps track of practically every action made in the restaurant by both the employees and the customers. This makes relaying data back to the management much easier. For example, SpeedByte will be able to recognize how many orders were placed, how many dishes are ordered per order, how much of the inventory was used per order, and how many employees are on staff between a given interval of time. With knowledge of this information, SpeedByte can easily create charts for management to keep track of the day to day running of the restaurant.

Actor - Busboy:

Most people think living the life of a busboy is easy - and it is, for the most part. But, there are some days that just have me turned about, with having so many customers and all. With the hostess asking you for an informative layout of the tables in the restaurant, I was wondering if you could put in some information that shows which tables have to be cleared off and cleaned. You know, that way I could quickly come over and do whatever has to be done to make the place spiffy for the next customer.

The SpeedByte solution: SpeedByte offers an interactive display which will keep track of every step along the way of a successful dining experience. With that being said, the Busboy(s) will know when a customer has left the restaurant, which will trigger the Busboy(s) to clean the now unoccupied table, once the table is clean the Busboy(s) will be able to set the table as a clean "Available" table.

Actor - Customer:

I'm really picky about my food - I prefer knowing that the food I'm about to get is near perfection, at least to most people. Some kind of rating system might help me with decisions regarding just that. In consideration of the foregoing, I also want to know what exactly is in the food, in the case of allergies and for calorie counting purposes. All this talk about food has made me hungry, so I'll probably call somewhere for food; actually, that gives me another idea! If there was a digital menu, I wouldn't have to go through the hassle of having to speak with someone and waste time trying to understand them as they try to understand me - my phone isn't especially good, so this is more of a problem than you might think. And perhaps if I could order ahead of time, that would cut down my waiting to get the food - fantastic! Speaking of waiting, if the menu also displayed how long an item would take to make, then I would know when to arrive at the restaurant to pick up the food when it's nice and hot. Although, there are also occasions when I like to just eat at the restaurant, but the line goes through the door sometimes! If only there was a way everyone already knew where to sit and could just get to it. What if there was a way to combine some of those ideas! Hey, let's say I order off of this menu, then maybe I can also reserve a seat for myself, and pay the bill while I'm eating. That way, I already have a seat and food ready to go - presto, time saved. Man, I'm a genius, eh?

The SpeedByte solution: SpeedByte offers a digital touch menu with numerous features to ensure customer satisfaction. An integrated rating system will show the customer what previous customers thought about a specific dish, this will guide the customer into selecting a dish that will help them leaving satisfied. The menu will also display what ingredients are in each dish which will avoid any allergy issues. To remove the uncertainty of a wait time, SpeedByte has the time it will take for each dish to be made displayed on the menu. Another great feature SpeedByte provides is a table selection option. This option allows the customer to reserve a specific table at a specific time.

Actor - Chef:

Day in and day out, I work in the kitchen - being a chef isn't terrible, but boy is it stressful. Having to make sure orders are filled out in a timely fashion and keeping track of dishes that have a few modifications requested to them by the customer really does a number on me sometimes. Not to mention, sometimes what the waiter writes down looks like downright chicken scratch, it all just makes the job harder than it has to be. So, what I would like to get from you guys is something that will help me read incoming orders clearly, with any possible modifications done to them in a different color. That way, I don't have to rely on the waiter's chicken scratch, and I can clearly see what is different about this meal from the standard. Next on my list would be something to help tell the waiter that the food is ready to pick up - that's another one of my pet peeves; I have a fresh meal ready for a customer and it just sits around for 5 minutes, taking up space on the counter for more dishes and getting cold all the while.

Now since that's all settled, there is one more idea that I have, but I'm not sure if it's worth the effort. I was thinking about having the incoming orders laid out in such a way so that I could tell what ingredients are common among them. That way, I could prepare larger batches of whatever it is that the meals require, say sautéed mushrooms for instance. If an order for a burger with mushrooms came in, and then an order of creamy mushroom on fettuccine, I would like to be able to see that they share sautéed mushrooms in common, so I could make more in one go rather than having two sautéing sessions. But, that's just my own little idea - I'm not even 100% sure it would work the way I think it could.

The SpeedByte solution: Because SpeedByte is a connected application, when the customer places an order it immediately goes into a queue until it is ready to be made. Once the order is ready to be made the Chef will be able to click “Start Order.” This will bring up the recipe for each dish. Once the Chef completes the order they will simply click “Order Complete” which will signal the server to pick up the food.

Actor - Waiter/Waitress:

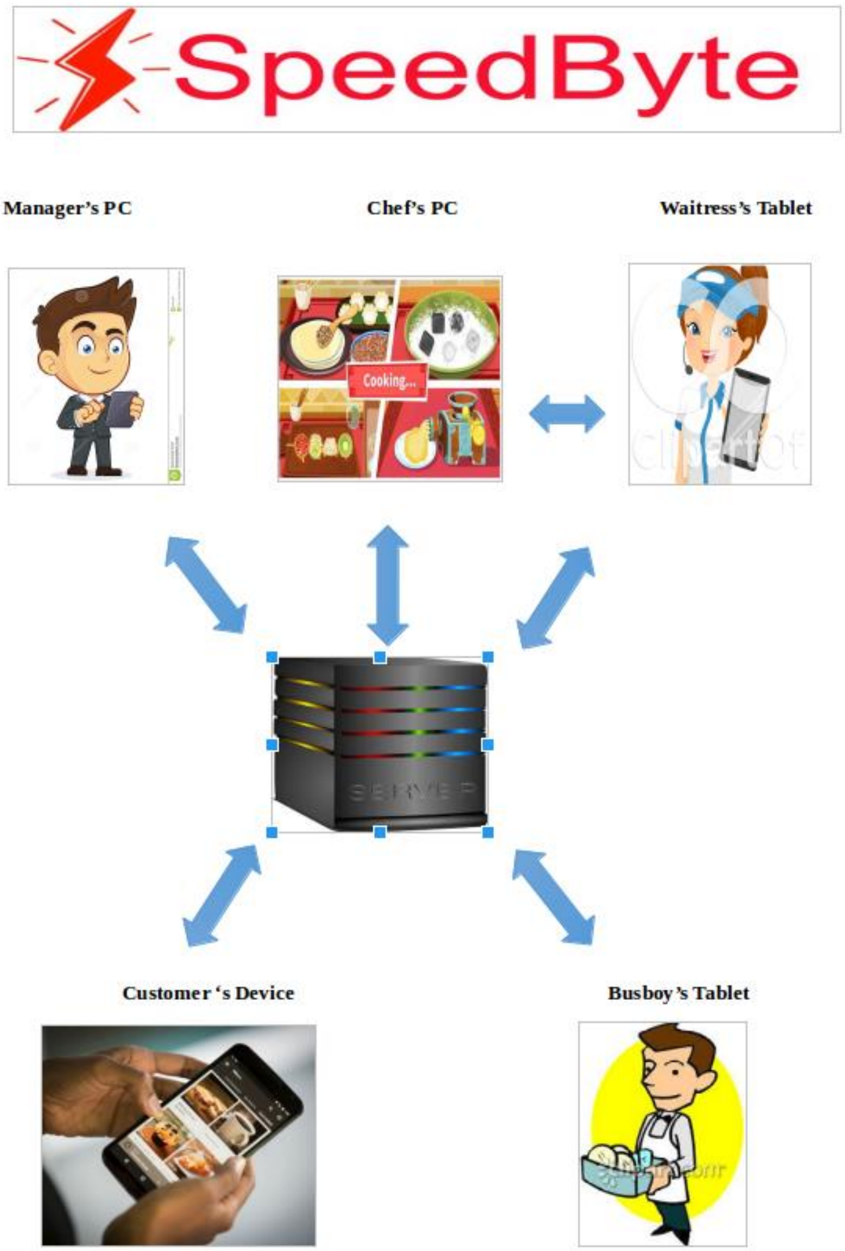
Man do I hate when the chef yells at me - he gets so agitated when I don't pick up stuff from him in the time he wants me to. It's not like I diddle-dally my way around the restaurant. I have customers to attend to as well, taking their orders and, well, waiting on them. I hear you guys are already implementing some sort of order ahead system - that is absolutely fantastic, since it will take a bit of the load off of me. I was just wondering if there was some way you can have me fill out orders digitally, and I would just send them off to the kitchen without even my having to go there. Also, if you guys could have a little list for me indicating which meal is done, and to which table it should go to, that would be fabulous. In fact, if you could have some way of indicating which table has yet to be served, or is in the middle of eating, or is wanting to pay their bill, or … well, you get the idea. Something that tells me the status of each table, so I know where to run to next that too would be extremely helpful. Maybe with all of these things done, the chef would no longer have a reason to yell at me.

The SpeedByte solution: Because of the integration of all components of the app, once the customer orders their food it will be relayed directly to the kitchen eliminating the middleman where things can get lost in translation. Once an order is ready, a notification will alert the server prompting them to deliver the food to the corresponding table. Once the food is delivered the server simple presses “Delivered” for the corresponding table which will remove it from the list.

Part b.) Glossary of Terms

- **Customer:** Orders food and services from the restaurant.
- **Manager:** Manages inventory, payroll, employee list and charts, customer's bill in order to provide discount due to inconvenience, log-in interface to prevent unauthorized access of restaurant management and statistics for the restaurant.
- **Waiter:** Delivers the order.
- **Chef:** Reads the order placed from a terminal in the kitchen and cooks food accordingly.
- **Busboy:** Keeps track of the dirty tables and updates once the cleaning is done.
- **Host:** Assigns seats to people who come to the restaurant. Changes the status of table to "Occupied" to "Vacant". Keeps track of clock in and clock out function for payroll purpose.
- **Add/Edit Employee:** Button only on the Management page to add or edit the information of an employee at the restaurant.
- **Manage Inventory:** Button only on the Management page of all the items required for food preparation in the restaurant.
- **Manage Payroll:** Button only on the Management page to manage the payrolls of the various employees in the restaurant.
- **Reports Screen:** Statistical and graphical data analysis of the traffic flows in the restaurant.
- **Menu:** List of dishes served in the restaurant, which will be displayed on an android application.
- **Floor Layout:** Shows all tables in the restaurant along with their respective status.
- **Order Status:** Shows whether the order of a particular table is ready to be "served" or "cooking".
- **User interface:** The visual on the computer and tablet that allows user interaction with the system.
- **Wireless Access Point:** Accesses points support Wi-Fi wireless communication standard.
- **Database:** Where all the data such as menu items, inventory, scheduling, payment, payroll, employee information, customer information and orders are stored. (http://i.stack.imgur.com/iTINi.png)
- **Order Queue:** a list of orders that are placed in (FIFO) first in, first out order. These orders are sent to the chef's PC, where the chef can prepare them.
- **Walk-in Queue:** a list of table reservations that are placed in (FIFO) first in, first out order. These reservations are made on the Customer PC when the customer walks into the restaurant.
- **Reservation System:** Customers can reserve a table using an android application.

2. System Requirements:



Part a.) Enumerated Functional Requirements

IDENTIFIER	PRIORITY	REQUIREMENT
REQ-01	5	Allow the customers to choose whether they want to dine in, take-out, or get food delivered.
REQ-02	5	Provide the customer with a click able menu.
REQ-03	3	Customer should be able to choose seating if dining.
REQ-04	5	The Chef PC should receive the order.
REQ-05	2	View estimated time of arrival for food and/or service.
REQ-06	4	Customer should be notified of order.
REQ-07	5	Customer should not be able to login as employee.
REQ-08	5	Customer should be able to choose the option to pay. (Cash or Online)
REQ-09	4	Busboy’s tablet should be able to update the floor layout by changing table’s status.
REQ-10	1	Manager’s PC shall track restaurant trends.
REQ-11	1	Manager’s PC should be able to track inventory of each dish sold over a period of time.
REQ-12	4	Chef’s PC alerts waiter/Waitress when food is ready to be served/delivered.

REQ-13	2	Busboy’s tablet shall notify the busboy what tables to clean.
REQ-14	4	Host shall Prioritize orders in a queue.
REQ-15	4	Chef’s PC shall notify the system to update inventory when a meal is ready to be delivered.
REQ-16	3	Manager’s PC should be able to access data.
REQ-17	2	Manager’s PC should be able to analyze and predict supply usage.
REQ-18	4	Manager’s PC will have the ability to edit the menu.
REQ-19	2	Manager’s PC shall alert the manager when the inventory has a low stock of a particular item.
REQ-20	3	Host will let Chef’s PC and Waiter’s PC know which customers came first.
REQ-21	1	Manager’s PC will have statistics on food popularity.
REQ-22	3	Host Should allow customer to rate each dishes
REQ-23	5	All data should be stored in the database.

Part b.) Enumerated Nonfunctional Requirements

Identifier	Priority	Requirement
REQ-24	1	The program should be user-friendly and meet the standards of the restaurant.
REQ-25	4	User Manual should be made to provide assistance.
REQ-26	5	The system should be backed up to avoid loss of data.
REQ-27	2	Safety protocols should be placed to protect the data from unauthorized users.
REQ-28	2	Operating time between screens should be minimized.
REQ-29	5	The system should have a low mean time to failure (MTTF) and high reliability components.
REQ-30	1	The system should work on any device.
REQ-31	3	The system should be relatively easy to debug.
REQ-32	2	Employee devices should be convenient and easy to operate.

Usability:

The system will be user friendly and easy to operate. The users will download an app and create a login. Manager and Employees will have their own digital device to interact with and assist the customers. The customers will have options to dine-in, take-out, or have their food delivered. The restaurant's entire menu will then be available for the customer to choose from. The order will be transferred to the manager's and chef's devices with all the information needed.

Reliability:

The system will consistently excel in performance. The initial options such as dine-in, take-out, and delivery, will be the first available options. Then, the menu option will be available for the customer to choose from. There is no chance of getting "invalid input" as the system will have all the specific options on the screen. Each user will have a unique username and password. Hence, there will be fewer risks of unauthorized and unknown users to hack into the system. All the options and security access will be available to the administration only.

Performance:

The system will be used by many customers and employees of the restaurant simultaneously, and will be able to handle it without any failures. To run the system efficiently and quickly, even with all of the possible traffic, the server will have to be efficient. To analyze the restaurant's performance for example: sales and inventory purchase, manager will have the option to compile the data all together and check the weekly/monthly performance. The restaurant will have a high-speed wireless connection for the system to perform its best.

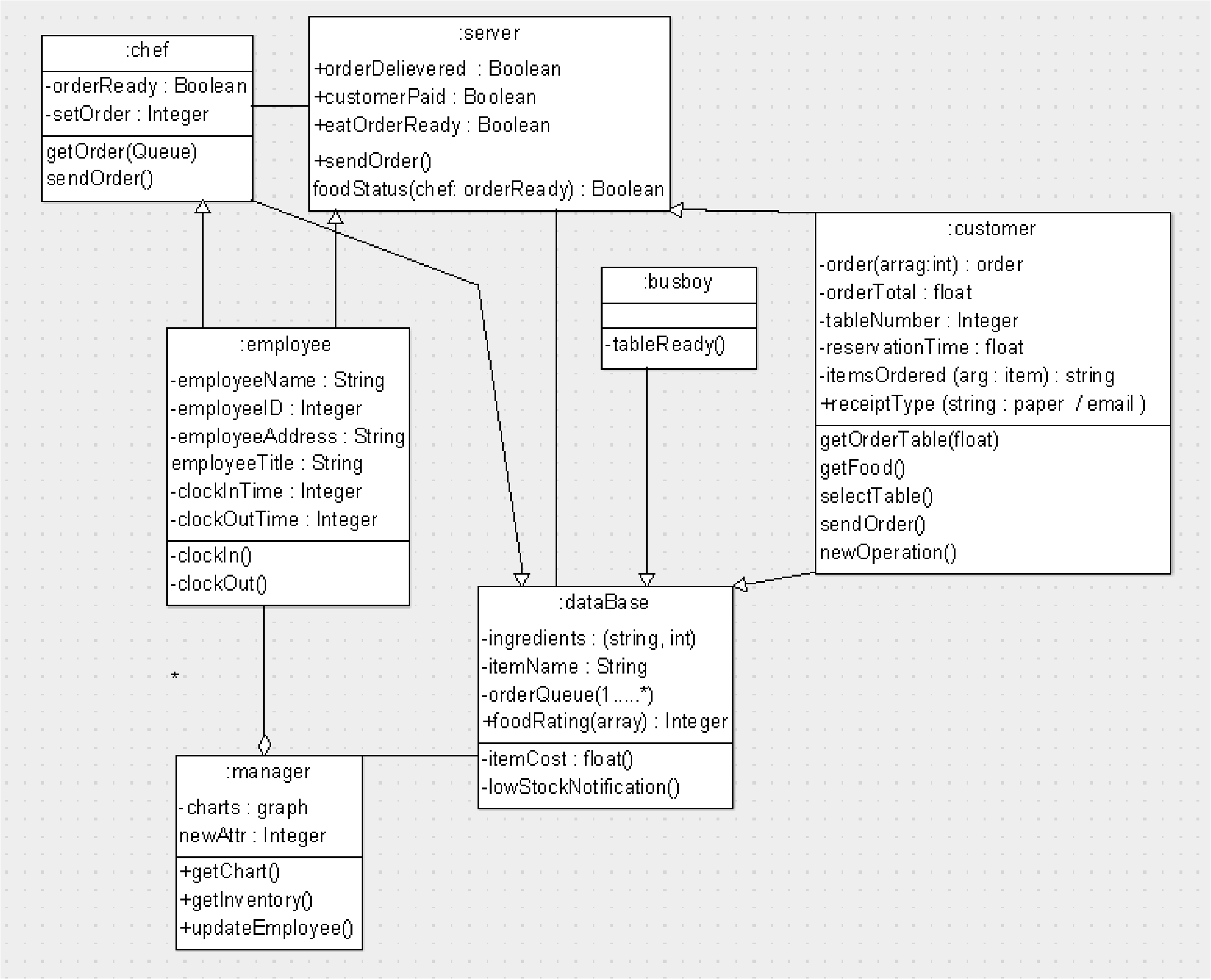
Supportability:

The system will support the modification. All the changes and replacement that an application might need in future will be allowed. For instance: change in menu, replacing the logo. Manager will have the ability to modify the menu. The restaurant manager will be the lone administrator of the system.

Implementation Requirements:

Android studio will be used to create our app. Node.JS will be used to create a server. Firebase will be used to implement a database to hold user/pass information.

Class Diagram:



Part c.) On-Screen Appearance Requirements

Identifier	Priority	Requirement
REQ-33	5	SpeedByte will display a current seating chart of available tables to the customer.
REQ-34	3	SpeedByte color schemes will be visually appealing and appetizing.
REQ-35	4	Menu will display ingredients, wait time, and rating
REQ-36	5	SpeedByte will display interactive charts to manager tracking inventory, sales, etc.
REQ-37	3	Will display current order's recipe to chef
REQ-38	5	Will display what tables are "Available, Occupied, or Need to be Cleaned" to employees.
REQ-39	1	Will include pictures of each menu item
REQ-40	3	Scroll down menus for all options to eliminate unnecessary clicks

User Interface:

Use Case Scenario

- Customer opens app and has three options (Dine-in, Takeout, and Delivery).
- Customer chooses Dine-in option.
- Customer is asked to choose a reservation time.
- Customer selects "Eat Now."
- Customer is presented with a seating chart displaying available tables.
- Customer chooses table 4.
- Hostess checks the name and seats the customer.
- Customer is brought to the menu screen.
- Customer chooses "Pasta Faggioli" with a rating of 4.8/5 stars and wait time of 10 minutes.
- Customer is brought food after a ten minute wait.
- Customer eats food and pays bill.
- Customer is presented with a request to rate the dish.
- Customer rates dish 5 stars.
- Customer leaves the restaurant satisfied.

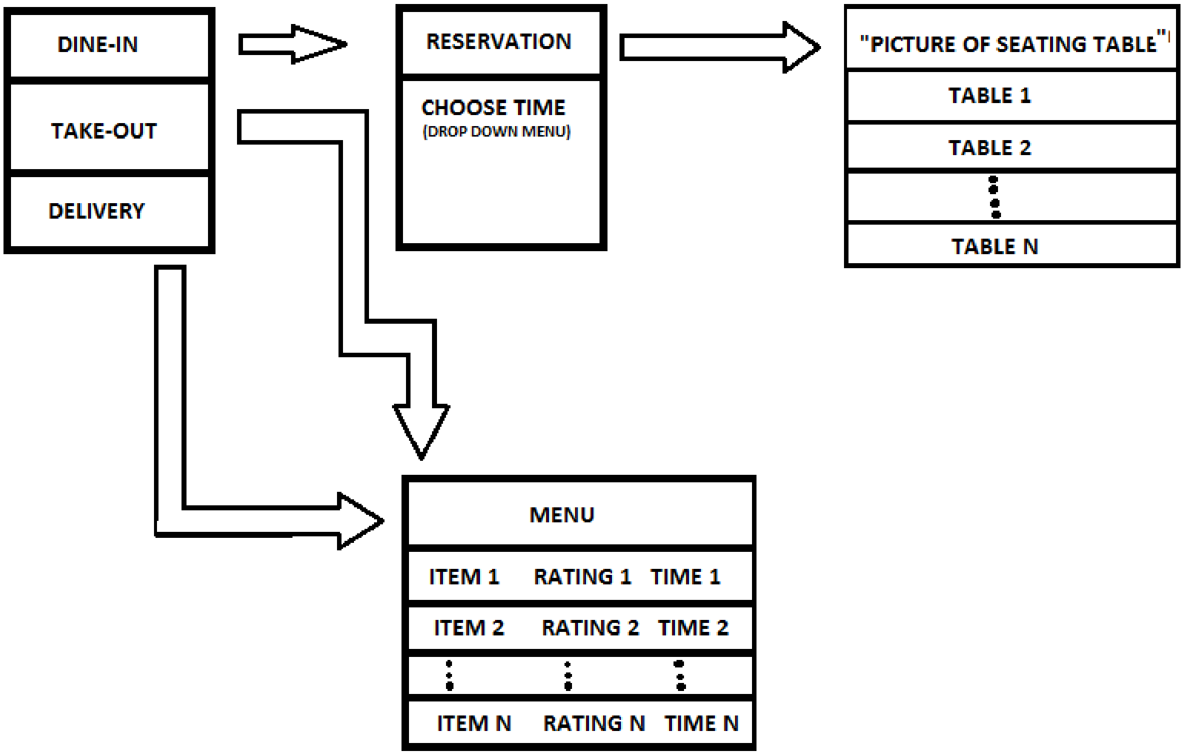


Figure: A sample on-screen interface for SpeedByte

3. Functional Requirements Specification:

Part a.) <u>Stakeholders:</u>

The following are the stakeholders who will have the most interest to design and run the system efficiently.

- ➢ **Restaurant owner**
- ➢ **Employees (Manager, Waitress, Busboy, Host, Chef and other employees.)**
- ➢ **Software Designer (Designer, Programmer and Manager)**

Part b.) <u>Actors and Goals:</u>

The following are the actors who will have to perform their duty as assigned (in chart below)

Part c.) <u>Use Cases:</u>

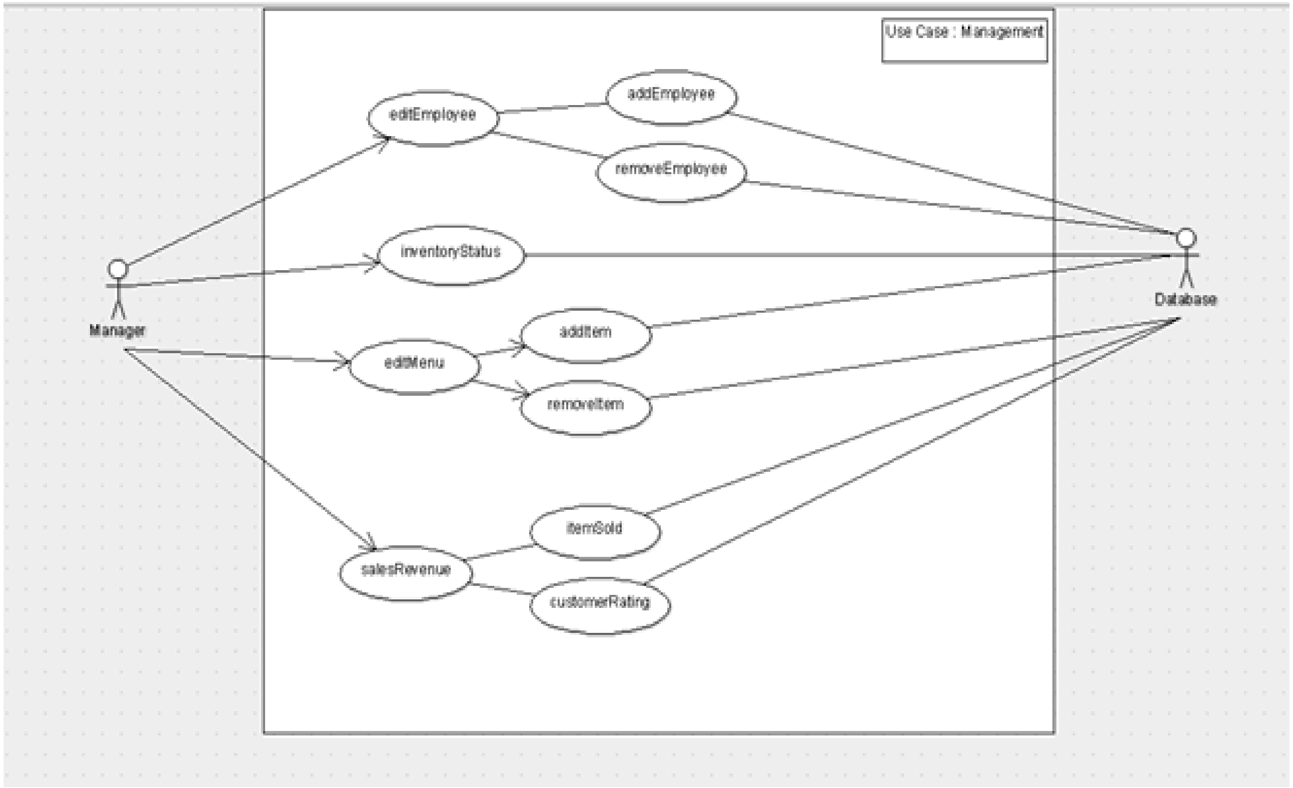
i.) Casual Description

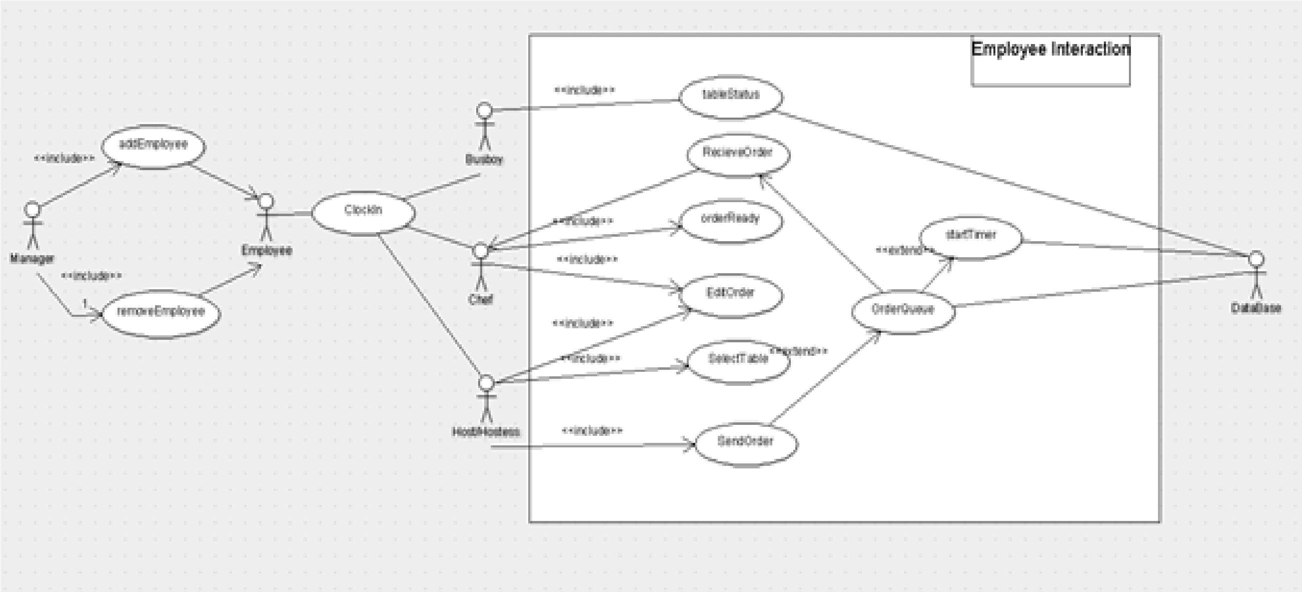
Actor	Goal	Use case	Casual Description	Type
Customer	Select Table	UC-1	Customer gets an option to choose a table.	Initiating
Customer	Place an order	UC-2	To choose dishes to place an order.	Initiating
Manager	Create Employee Account	UC-3	Create login for newly hired employee.	Initiating
Manager	Edit Employee Account	UC-4	Edit information of current employees. For ex: Adding/changing First name, Last name, Address, Contact detail, SSN, etc.	Initiating
Manager	Add Account	UC-5	Add more employee accounts.	Initiating

Manager	Delete Account	UC-6	Remove existing accounts of the employees who no longer work in the restaurant.	Initiating
Manager	Edit Menu	UC-7	Add/Delete Menu.	Initiating
Manager	Track Traffic	UC-8	Keep track of flowing traffic (customer).	Initiating
Manager	Manage Inventory	UC-9	Keep Track of Inventory.	Initiating
Manager	Manage Payroll	UC-10	Keep track of clock in/clock out of all the employees and do the payroll weekly accordingly.	Initiating
Manager	Manage Payment	UC-11	To verify if the payment is received online or not. To collect cash payment if a customer pays cash.	Initiating
All Employee	Login	UC-12	All employees will have to login to work.	Participating
Employee (Waitress)	Delivering order	UC-13	Employee will deliver the order. (Be it dine-in, take out or delivery)	Initiating
Busboy	Cleaning	UC-14	To clean and prepare a table right after a customer leaves.	Initiating
Chef	Prepare the order and Signal for a pick up	UC-15	To cook the food and signal the employee once the order is	Initiating

			prepared and ready to be delivered.	
Host	Recognizing an input or waiting number	UC-16	To recognize use's input and assign a vacant table to a customer. If the table is not available, it assigns a waiting number.	Initiating
Host	Update Floor layout	UC-17	To change the status of all tables to Available or Occupied (Avail. => Occupied) (Occupied => Avail.)	Initiating
Database	Storing the data	UC-18	To update inventory, keep track of employee information, and manage any other data.	Participating

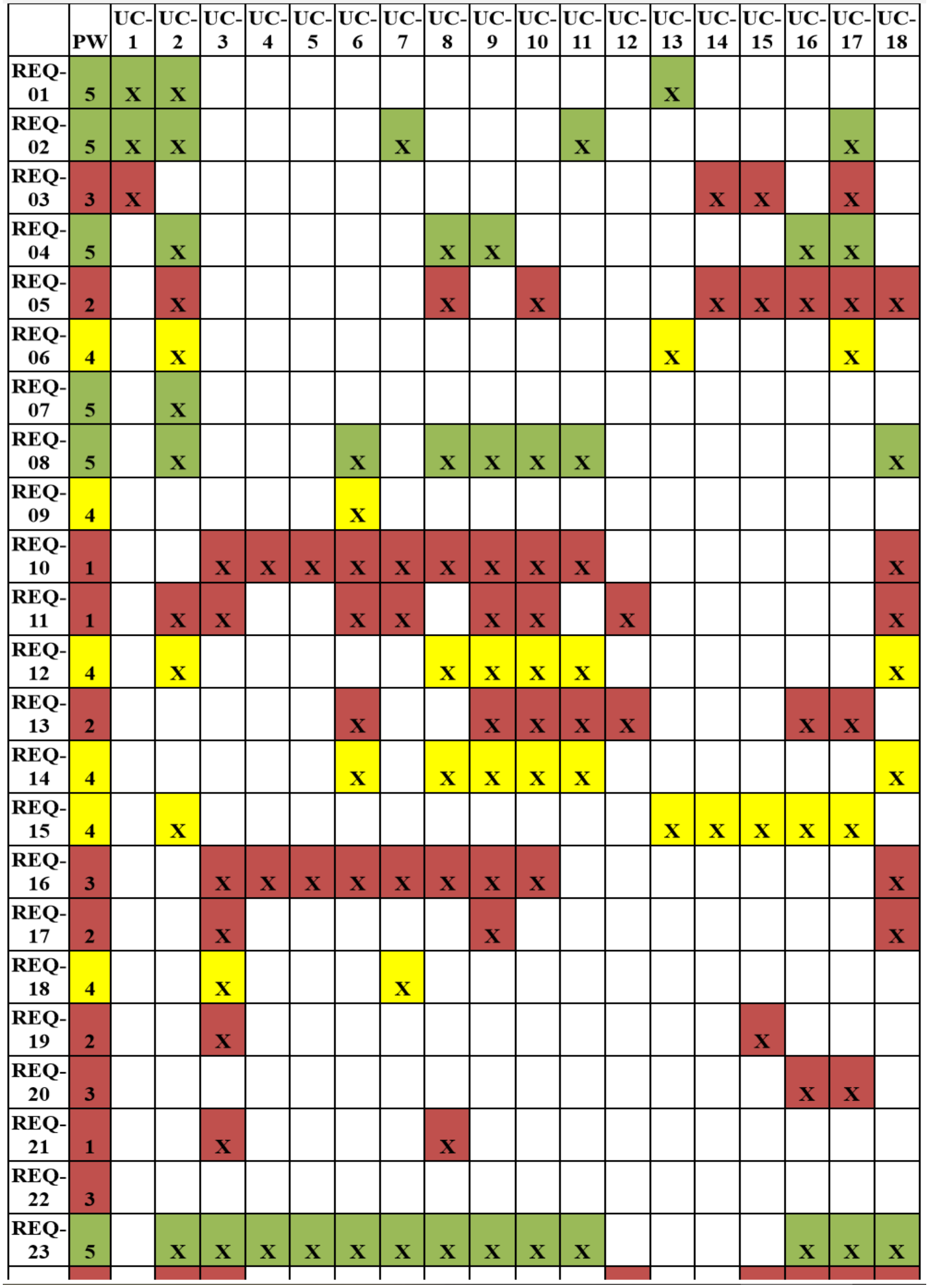
ii.) Use Case Diagram





iii.) Traceability Matrix

Green = highest priority. Yellow = second highest priority. Red = least priority.



	PW	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12	UC-13	UC-14	UC-15	UC-16	UC-17	UC-18
REQ-01	5	X	X											X					
REQ-02	5	X	X					X				X						X	
REQ-03	3	X													X	X		X	
REQ-04	5		X						X	X							X	X	
REQ-05	2		X						X		X				X	X	X	X	X
REQ-06	4		X											X				X	
REQ-07	5		X																
REQ-08	5		X				X		X	X	X	X							X
REQ-09	4						X												
REQ-10	1			X	X	X	X	X	X	X	X	X							X
REQ-11	1		X	X			X	X		X	X		X						X
REQ-12	4		X						X	X	X	X							X
REQ-13	2						X			X	X	X	X				X	X	
REQ-14	4						X		X	X	X	X							X
REQ-15	4		X											X	X	X	X	X	
REQ-16	3			X	X	X	X	X	X	X	X								X
REQ-17	2			X						X									X
REQ-18	4			X				X											
REQ-19	2			X												X			
REQ-20	3																X	X	
REQ-21	1			X					X										
REQ-22	3																		
REQ-23	5		X	X	X	X	X	X	X	X	X	X					X	X	X

REQ-24	1		X	X									X			X	X	X	X
REQ-25	4		X	X															
REQ-26	5			X															X
REQ-27	2		X	X															X
REQ-28	2		X	X															
REQ-29	5	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	X
REQ-30	1	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	X
REQ-31	3	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	
REQ-32	2			X	X	X	X	X	X	X	X	X	X			X	X	X	
REQ-33	1	X																X	
REQ-34	3	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	
REQ-35	2	X	X																
REQ-36	1		X	X					X	X	X								
REQ-37	3															X			
REQ-38	1														X		X	X	
REQ-39	5	X	X					X											
REQ-40	3	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	

iv.) Fully-Dressed Description

The following are the few important use cases along with their goals, Main Success Scenario and Alternate Success Scenario.

Customer/Employee/Host/Busboy/Database

Use Case: UC-1, UC-2, UC-17, UC-14	Select table (Dine in), Place an order, and Update floor layout.
Initiating Actors	Customer, Host, Busboy
Participating Actors	Employees, Chef, Database
Goal	The app will allow placing an order and choosing a table to dine in the restaurant.
Preconditions	The user already has an account created and is willing to dine-in the restaurant.
Post conditions	The Host assigns a table or a waiting number.
Main Scenario: ➢ Customer opts to dine-in and then chooses a table. ➢ System updates the floor layout once the table is chosen (changes the status of table from “available” to “occupied.”) ➢ The Host brings customer to the chosen table. ➢ Customer places order. ➢ Chef receives order and begins preparing order. ➢ Chef finishes preparing order, and signals the employees the food is ready. ➢ Employee receives signal and brings prepared order to the customer’s table (employee confirms order delivery). ➢ Customer finishes eating and pays the bill. ➢ System notifies the busboy to clean the table. ➢ Busboy cleans and prepares the table for future customers. ➢ Busboy notifies the system by changing table’s status to “available”. ➢ System updates the floor layout with the new available table(s).	
Alternate Scenario: ➢ Customer selects unavailable table. ➢ Host recognizes user’s input and notifies customer with a message “Table is occupied” and asks user to wait or to choose another table. ➢ If customer opts to wait, Host updates the waiting queue in order; else employee updates the floor layout once the table is chosen.	

Manager/Employee/Database

<table>
<tr><td>Use Case: UC-4</td><td>Edit Employee Account</td></tr>
<tr><td>Initiating Actor</td><td>Manager</td></tr>
<tr><td>Participating Actor</td><td>Employee and Database</td></tr>
<tr><td>Goal</td><td>Edit existing employee account in the database.</td></tr>
<tr><td>Preconditions</td><td>An employee entry exists in the database whose information needs to be updated.</td></tr>
<tr><td>Post conditions</td><td>Employee information is updated.</td></tr>
<tr><td colspan="2"><b>Main Success Scenario:</b>
➢ Manager selects the interface to edit the existing employee account in the database. For Instance: To change First name, Last name, DOB, Address, password, SSN, etc.
➢ Manager and system must have SSN.
➢ Validate the fields.
➢ Employee information gets updated.</td></tr>
<tr><td colspan="2"><b>Alternate Scenario:</b>
➢ Manager selects the interface to edit the information for an existing employee account in the database and a same name account with all the same information for example: First name, Last name, DOB and SSN already exists.
➢ System notifies an error message “Information cannot be updated and stored as same account already exists more than once”.
➢ Manager verifies and deletes one same account from the database.
➢ Manager accesses the sole account to update and save the information again.
➢ System allows and information gets updated and stored.</td></tr>
</table>

Chef/Employee/Database

Use Case UC-15	Preparing the Order
Initiating Actor	Chef
Participating Actor	Employee and Database
Goal	To cook the order and notify the employee when it is ready.
Preconditions	Customer places an order and chef receives the order.
Post conditions	The employee is notified when the food is ready to be delivered to the table. Plus, each time an order is prepared, the corresponding inventory items are decreased by the pre-specified quantity.

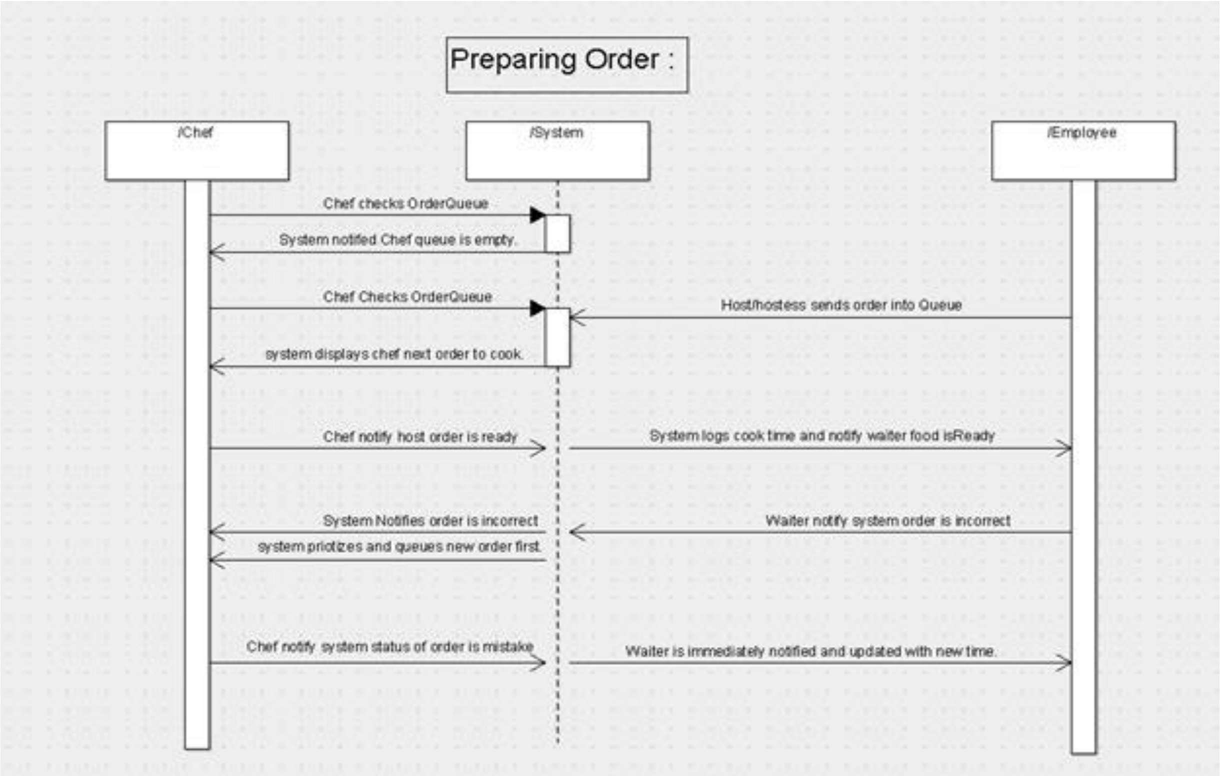
Main Success Scenario:

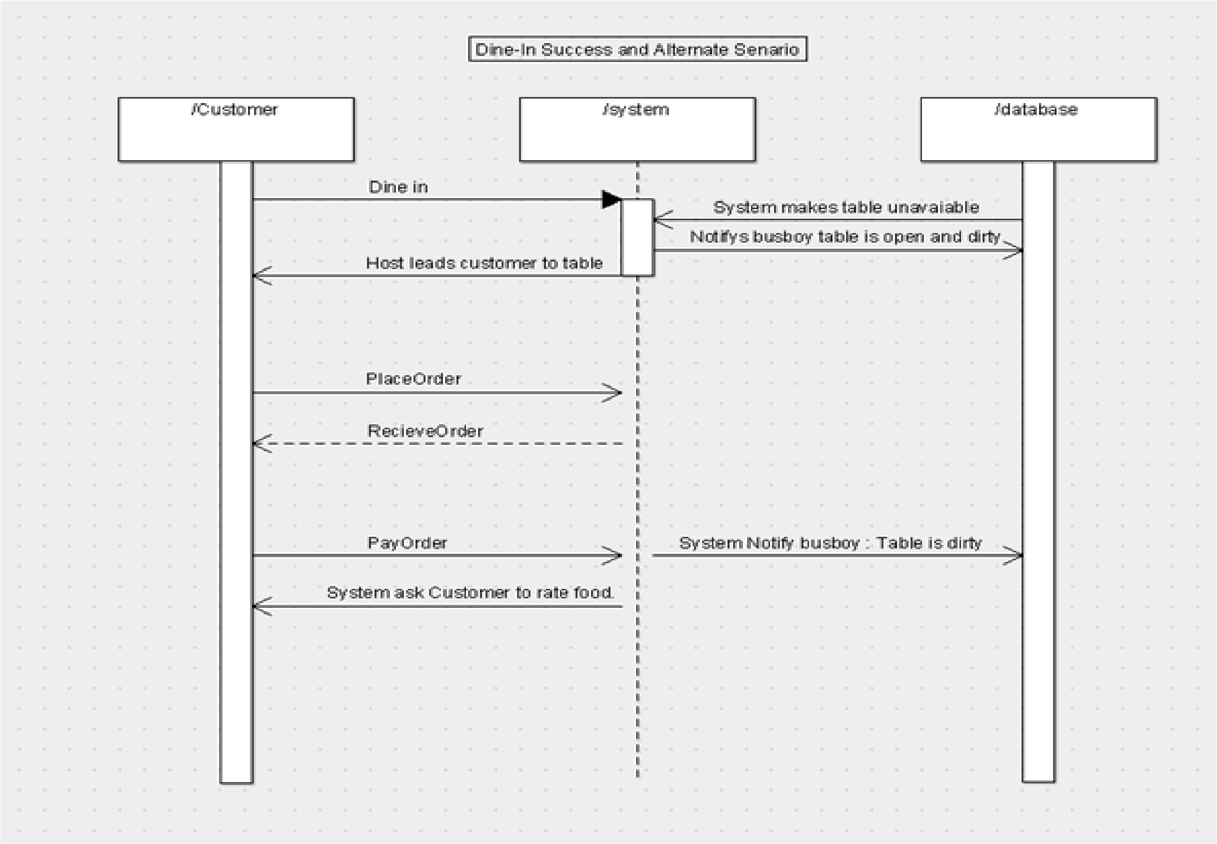
- Chef receives an order
- Once the order is prepared, Chef clicks on the button “ready”.
- System changes the status of the order to “ready” and notifies the employee.
- Employee picks up the order and delivers to the customer.
- System updates the inventory in the database.

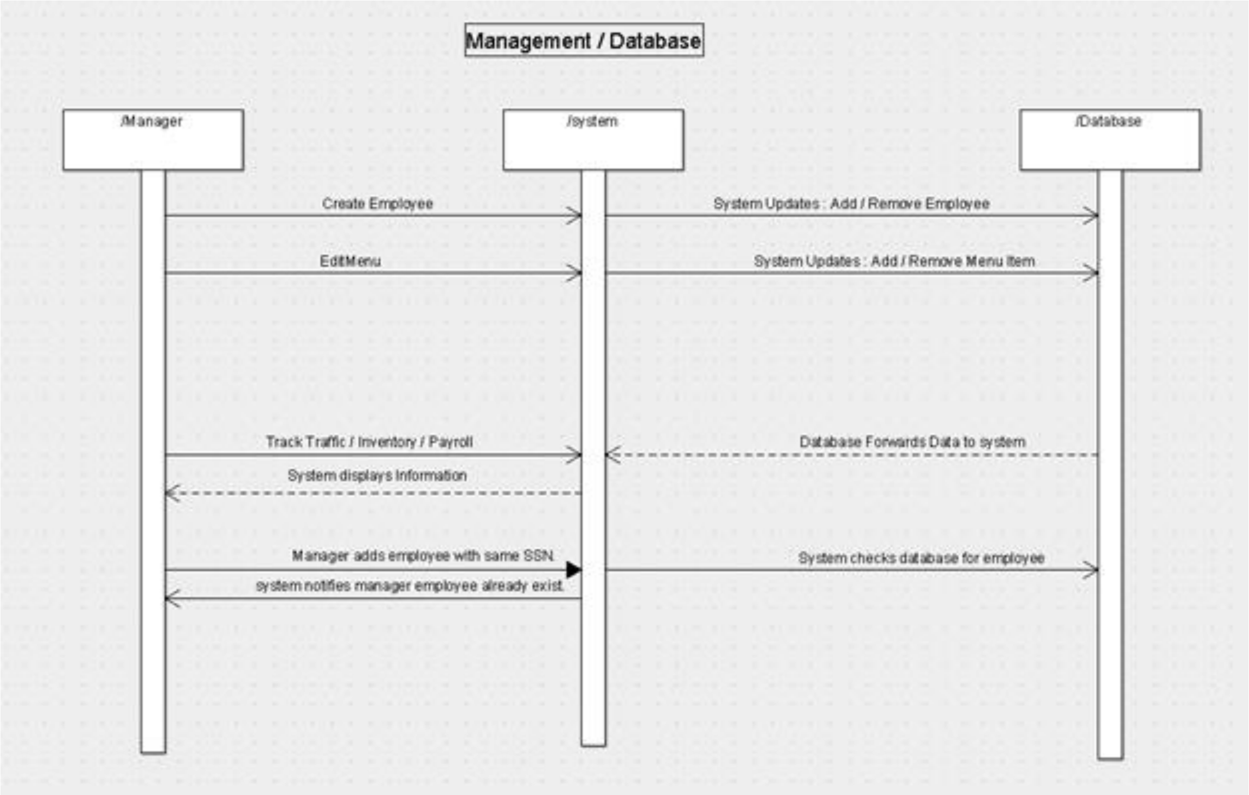
Alternate Scenario:

- Chef receives an order and mistakenly changes the status of an order to “ready” with in few seconds.
- Chef must contact employee immediately regarding the mistake.
- Chef must fix the mistake on his tablet.
- System updates employee regarding the change in status of the order.
- System updates the inventory in the database.

Part d.) System Sequence Diagrams:


Preparing Order :
/Chef
/System
/Employee
Chef checks OrderQueue
System notifed Chef queue is empty.
Chef Checks OrderQueue
Host/hostess sends order into Queue
system displays chef next order to cook.
Chef notify host order is ready
System logs cook time and notify waiter food isReady
System Notifies order is incorrect
Waiter notify system order is incorrect
system priotizes and queues new order first
Chef notify system status of order is mistake
Waiter is immediately notified and updated with new time.

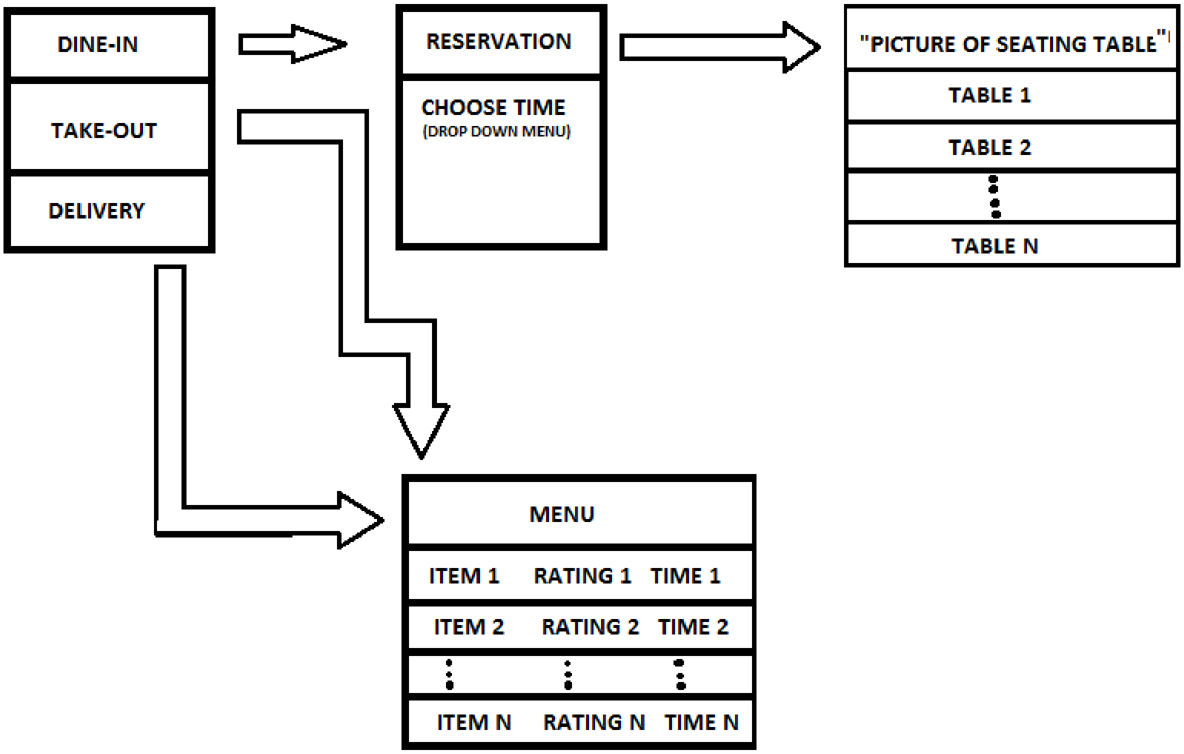

Dine-In Success and Alternate Senario
/Customer
/system
/database
Dine in
System makes table unavaiable
Notifys busboy table is open and dirty
Host leads customer to table
PlaceOrder
RecieveOrder
PayOrder
System Notify busboy : Table is dirty
System ask Customer to rate food.


Management / Database
/Manager
/system
/Database
Create Employee
System Updates : Add / Remove Employee
EditMenu
System Updates : Add / Remove Menu Item
Track Traffic / Inventory / Payroll
Database Forwards Data to system
System displays Information
Manager adds employee with same SSN.
System checks database for employee
system notifies manager employee already exist.

4. User Interface Specification:

Part a.) Preliminary Design

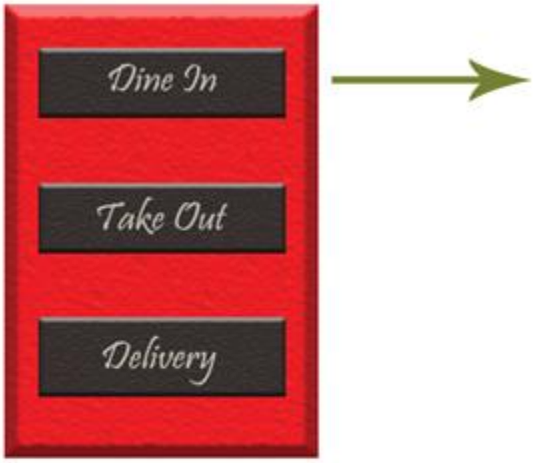
When designing an application the designer must keep the user in mind at all times. This requires the designer to make the interface as simple as possible for the user while still carrying out the expected functions.



The above diagram shows the flow that the customer will experience depending on which options are chosen.

Design Concept Art:

Below is the concept art of the diagram which will be seen by the user and fill the entire screen. The user will be able to select one of the below options which will bring them to another screen depending on their selection.



After creating an account the user interface is quite simple requiring a few simple clicks that are described below.

Customer Information Inputting (Step by Step):

The user opens the app and is brought to a screen consisting of three options (Dine in, Takeout, Delivery). Depending on which option is chosen they are brought to the next screen.

<u>Dine In Option (Customer)</u>

Choosing the dine-in screen will bring the user to the reservations drop down menu where “Eat now” or times in increments of 15 are presented. The user clicks on an available time which brings them to “Table Selection Screen” where they may select an available table simply by clicking on a drop down menu listing the table numbers. After choosing a table the customer is presented with a reservation code which is presented to the hostess. Hostess confirms reservation and the menu is brought up consisting of the menu items, as well as their ratings, and the wait times of each corresponding dish. Customer selects item(s) by clicking on each desired item, and then clicks place order. After the server brings the food to the table the customer is sent the bill and asked if they wish to rate their meal by clicking a star representing one through 5. Customer is sent receipt and enjoys the rest of their day.

Takeout Option (Customer)

After selecting the “Takeout” option the customer is brought directly to the menu where they select which items by simply clicking on each item they desire. After selecting their items they click “Checkout.” This brings them to their bill where they can pay immediately or wait until they pick it up. After paying for the meal the customer is asked to rate it if they choose to do so by clicking a star from one to five. The customer closes the app and enjoys the rest of the day.

Delivery Option (Customer)

After choosing the delivery option the customer is brought directly to the menu. The menu where they select which items by simply clicking on each item they desire. After selecting their items they click “Checkout.” This brings them to their bill where they can pay immediately. After the meal is delivered the user is asked to rate the meal from one to five if they please. User closes the app and enjoys the rest of the day.

Chef Interface:

Shows an example of an order placed by a table, the chef has the option to move to the next order in the queue, the previous order, or to signal the server that the order is ready for the customer.

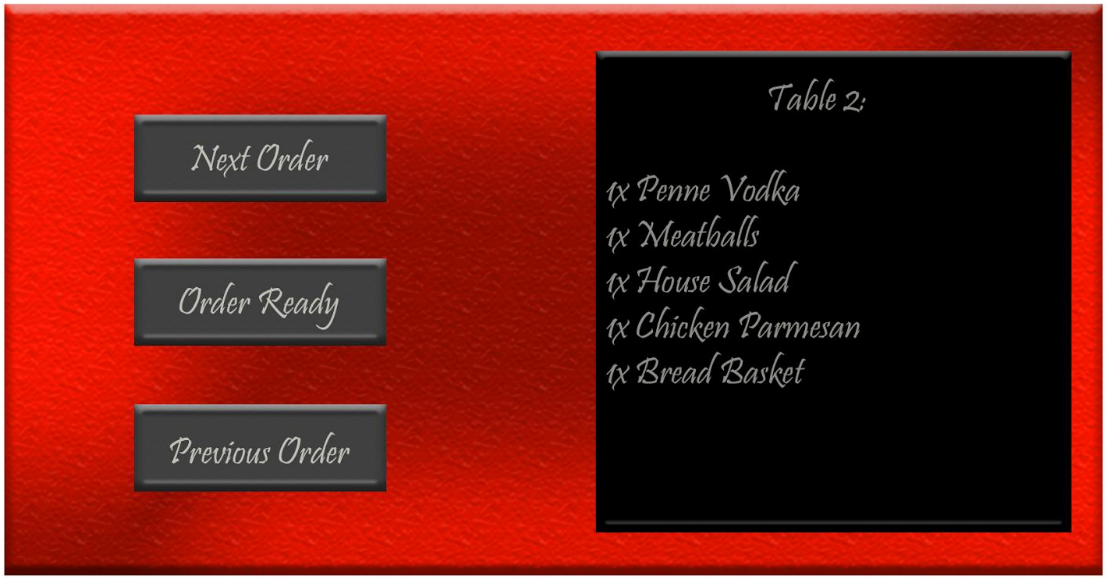
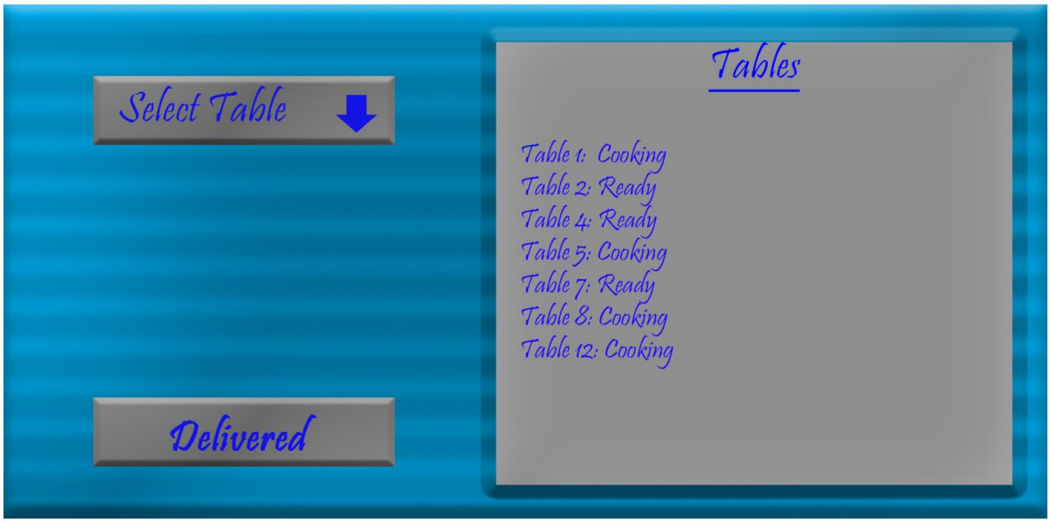


Table 2:

1x Penne Vodka
1x Meatballs
1x House Salad
1x Chicken Parmesan
1x Bread Basket

Server Interface:

Once the Chef begins an order the corresponding table number will appear on the screen and show the server that the meal is being prepared. Once the Chef completes the order the display turns from “Cooking” to “Ready.” The Server now can deliver the food to the correct table. Once delivered the server will click on the “Select Table” drop down menu to select the table to which they just delivered. Once selected the server clicks the delivered Button removing the table from the list.



Tables

Table 1: Cooking
Table 2: Ready
Table 4: Ready
Table 5: Cooking
Table 7: Ready
Table 8: Cooking
Table 12: Cooking

Part b.) User Effort Estimation

Customer Interface:

The effort estimation varies greatly depending on the number of meals that the user orders as well as the dining option that is chosen.

Best Case Scenario:

- Customer selects Delivery. (1 Click)
- Customer selects one dish. (1 Click)
- Customer clicks checkout. (1 click)
- Customer pays cash.
- Customer selects not to rate the meal and instead closes the app.

Total clicks = 3

Worst Case Scenario:

- Customer selects Dine In option. (1 click)
- Customer selects a time from the scroll down menu. (1 click)
- Customer selects a table. (1 click)
- Customer selects N menu items (N clicks)
- Customer selects check out. (1 click)
- Customer rates each item. (N Click)
- Customer closes app.

Total clicks = 4 + 2*N

Employee Interface:

The employee interface depends on different variables, like customer input.

Chef:

- Chef reads menu items that are in queue to be prepared and clicks “Start dish.” (1 click)
- Chef completes dish and chooses “Dish complete.” (1 click)

Best and Worst Case Scenario:
Total clicks = 2

Server:

- ➢ Server sees that all dishes for table are prepared and clicks “brought to table.” (1 Click)
- ➢ Server collects money from customer and selects “Meal Paid.” (1 Click)

Worst and Best Case Scenario:
Total clicks = 2

Management Interface:

The management interface receives data from all of the other interfaces and logs them appropriately. The manager will be able to keep track of statistics such as inventory, income, and amount of customers, within a certain time period. These can be displayed as graphs for ease of use.

Manager Selects Statistics:

- ➢ Manager opens app and is presented with clickable options for each statistic.
- ➢ Manager chooses an item to observe. (1 click)
- ➢ Manager is brought to a graph of that items progress hour by hour for the current day.
- ➢ Manager has the option to change time by selecting a preset time being (hourly, daily, weekly, monthly, and yearly). (1 Click)
- ➢ Manager Closes the app.

Best Case Scenario:
Total clicks = 1

Worst Case Scenario (per item):
Total clicks = 2

5. Domain Analysis

Part a.) Domain Model

i) Concept Definitions

Responsibility Description	Type	Concept Name
Populating reservation list with customers that have already ordered and selected a seat	K	Host/Hostess Menu
Check if seat selected by customer is available, waiting for new input if seat chosen is unavailable.	D	Customer Menu/Server
Recording of all information that is being sent, regarding food ordered, and time and day of order.	D	Server
Displaying in a coherent way all data that was collected, in the form of an infographic.	K	Server
Keep a backlog of authorized users and their passwords for different restaurants (for employee log-ins).	K	Server
Handle attempts to log in as an employee, and which menu to display for that employee.	D	Server
Acquire and display current items that have been ordered (in order for the chef to have a list of meals to make).	K	Chef Menu
Update seats in order to specify whether it is available, occupied, or in need of cleaning.	D	Server
Inform busboy when a table updates from occupied, so that it may be cleaned off.	K	Busboy Menu
Prompt users for a rating of the meal they enjoyed, and record the result.	D	Customer Menu
Display current stock to authorized user, and allow the user to modify the data.	D	Manager Menu
Display current employees to authorized user, and allow the user to make modifications (possibly to only users below authorized users standing - to prevent users from updating their own wages and such).	D	Manager Menu

Remind waiter of any tables that still have open orders, so that customers can get their orders in.	K	Waiter Menu
For deliveries and take-outs, have an active timer ticking down the ETA.	K	Customer Menu
For the chef, when he finishes making the meal, have him update the meal status so that the waiter is called to pick up the food.	D	Chef Menu and Waiter Menu
Have system allow for splitting the bill, and calculate the necessary splits.	D	Waiter Menu and Customer Menu
Modify food items on the menu from authorized user	K	Manager Menu

ii) Association Definitions

Concept Pair	Association Description	Association Name
Manager Menu ⇔ Server	An authorized user can prompt the server, from the Manager Menu, to deliver statistics on customer flux. From the Manager Menu, the user can also update information server-side, regarding other employees, food items, updates to stock, etc.	Send statistics
Host/Hostess Menu ⇔ Customer Menu	The customer menu sends updates to the Host's/Hostess' Menu regarding a reservation, if the customer has chosen to dine-in	Show reservation
Chef Menu ⇔ Waiter Menu	Chef sends information to waiter regarding the status of food, so that the waiter can pick it up in a timely fashion. The waiter delivers any orders from customers that don't use the app.	Sends status of food
Chef Menu ⇔ Customer Menu	When the customer orders through the app, the Chef's menu gets updated with their order. The Chef's menu updates the customer's menu to display how the food is coming along.	Update Food Status
Server ⇔ Customer Menu	Server keeps track of what the customer ordered, at what time, in what quantity, in order to have local data be as up to date as possible.	Receive Food Information
Waiter Menu ⇔ Busboy Menu ⇔	When the waiter has received payment, the table the payment was received from updates to the "dirty" status, thus indicating to the busboy that a table needs	Review Table Status

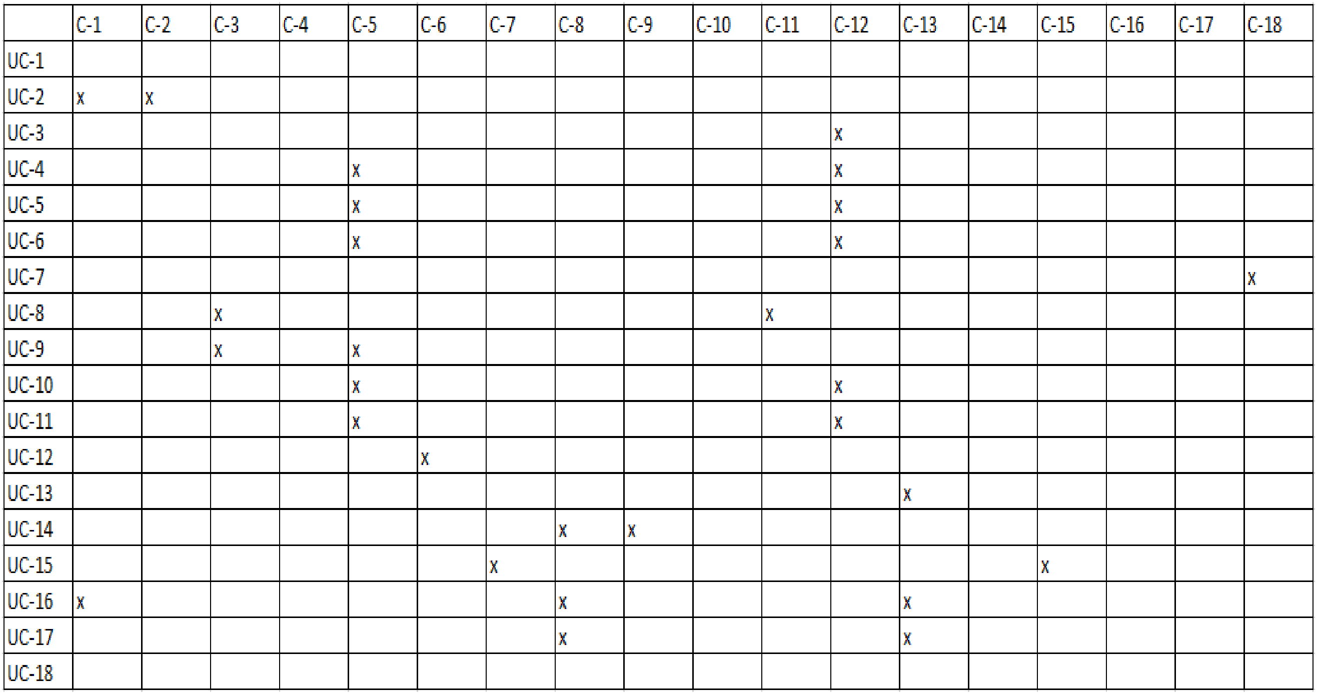
Host/Hostess /Customer Menu	cleaning. When the table is cleaned off, the busboy menu sends an update to the Host/Hostess and Customer menus indicating the table is available.	
Host/Hostess Menu ⇔ Customer Menu	When the host/hostess takes care of seating non-app users, those seats become updated to unavailable on the Host/Hostess Menu along with the Customer Menu.	Update Reservation

iii) Attribute definitions

Concept	**Attribute**	**Description**
1. Host Menu	listReservations	List of customer's reservation
2. Customer Menu	tableStatus	Checks to see if table is clean and available.
3. Server	listItem orderTime	Lists how many items ordered on the menu Provides Time and day of ordered food.
4. Server	graph	Provides and builds graphs for management.
5. Server	checkUser checkPassword	Checks if user is authorized to access data Checks if user input correct password associated with account.
6. Server (?)	userLogin showMenu	User Logins into account Provides appropriate menu for employee
7. Chef Menu	getOrder	List food ordered from orderQueue
8. Server	tableStatus	Provides and updates status of table.
9. Busboy Menu	getTableStatus	Displays table status if empty and if it is clean/dirty
10. Customer Menu	rateItem	Allows user to rate meal
11. Manager Menu	getInventory orderInventory	List current stock Add items to inventory

12. Manager Menu	displayEmployee editEmployee	Displays employee data to user Modify employee wage/address/name
13. Waiter Menu	openOrder	Checks if table has an open order.
14. Customer Menu	orderTime	Provides estimated time of food delivery for deliveries and take-out
15. Chef Menu	orderReady	Updates status of order
16. Waiter Menu	orderStatus	Checks if item ordered by customer is ready.
17. Waiter Menu / Customer Menu	splitBill	Allows user to split bill in multiple parts.
18. Manager Menu	editMenu	Modify food menu from authorized user.

iv) Traceability Matrix



	C-1	C-2	C-3	C-4	C-5	C-6	C-7	C-8	C-9	C-10	C-11	C-12	C-13	C-14	C-15	C-16	C-17	C-18
UC-1																		
UC-2	x	x																
UC-3												x						
UC-4					x							x						
UC-5					x							x						
UC-6					x							x						
UC-7																		x
UC-8			x								x							
UC-9			x		x													
UC-10					x							x						
UC-11					x							x						
UC-12						x												
UC-13													x					
UC-14								x	x									
UC-15							x								x			
UC-16	x							x					x					
UC-17								x					x					
UC-18																		

Part b.) System Operation Contracts

Name:	Select Table
Responsibilities:	To select and update tables to either sit customers or to be cleaned
Use case:	UC-1
Exceptions:	None
Preconditions:	Table must be empty, either before cleaning or after cleaning
Post conditions:	Table is ready to be cleaned or table can now sit customers

Name:	Place an Order
Responsibilities:	To place an order and notify the chef.
Use case:	UC-2
Exceptions:	None
Preconditions:	Order is placed by all the users
Post conditions:	Order is placed in queue and chef is notified

Name:	Create Employee Account
Responsibilities:	To create a new account for a newly hired employee
Use case:	UC-3
Exceptions:	Only administration is allowed to create an employee account
Preconditions:	Administrator is allowed to create an account.
Post conditions:	Employee account is created.

Name:	Edit Menu
Responsibilities:	To modify the menu
Use case:	UC-7
Exceptions:	Only administration is allowed to edit menu
Preconditions:	Menu is available to be updated.
Post conditions:	Menu is updated.

Name:	Manage Inventory
Responsibilities:	Update the ingredients based on users input
Use case:	UC-9
Exceptions:	Item that does not exist in the database cannot be updated
Preconditions:	Inventory database has each item either 0 or n amounts
Post conditions:	Inventory either increments or decrements the quantity of item(s). Inventory will notify if there is an item out of stock.

Name:	Manage Payment
Responsibilities:	To prepare the bill
Use case:	UC-11
Exceptions:	Only administration is allowed to manage payment
Preconditions:	Table ordered has been completed and all the items are delivered.
Post conditions:	Payment is received.

Name:	Login
Responsibilities:	To allow users to input their ID and password
Use case:	UC-12
Exceptions:	Unauthorized with invalid inputs users cannot login in the system.
Preconditions:	Login information is available.
Post conditions:	User is either login or rejected based on the user valid input.

Name:	Delivering Order
Responsibilities:	Item status is changed by chef and waiter is notified that order is ready to be delivered
Use case:	UC-13
Exceptions:	None
Preconditions:	Order is waiting to be prepared by chef.
Post conditions:	Chef notifies the waitress that the order is ready to be delivered. Inventory is updated.

Name:	Cleaning
Responsibilities:	Table status has been changed to clean.
Use case:	UC-14
Exceptions:	Only the busboy is allowed to change the status of the table to clean (available) and dirty (unavailable).
Preconditions:	Table is dirty (unavailable).
Post conditions:	Table status is changed to clean (available).

Part c.) Mathematical Model

Algorithm for Customer's order:

If the customer wants to dine in, he/she will be able to select that option. Then the customer will be given a list of tables to choose from. After that, a menu will be displayed which the customer will be able to select items to be added to the order list. After the customer finalizes their order, they will be provided with an approximate wait time. If the customer wants to either take out or have to food delivered, the items from the menu will need to be selected to add to the order list. If the delivery option is chosen, the customer also need to enter in the address of delivery. In both of these options, the customer will be given a total wait time.

```
while(phone number is not valid){
   Customer enters his/her phone number
}

if(customer is dining in){
   while(tables are not full)              //customer selects table
                  Customer can select the table and make changes if wanted
         while(items are being selected)      //customer selects menu item
      Chef sees order and order is added to customer's orderList
   if(food is done cooking)                //food notification
                  Notify customer and remove item from orderList
}
if(customer is ordering takeout){
   while(items are being selected)
      The selected items will be added to the orderList and sent to the chef
         For the list of items,
      Provides estimated time of preparation and start countdown timer
}
if(customer is ordering delivery){
   while(items are being selected)
      The selected items will be added to the orderList and sent to the chef
         Provide a food preparation + delivery time to the customer
}
```

Algorithm for Employee/Busboy/Chef:

The waitress/busboy will be able to see which tables need cleaning. The cook will receive an updated list of the customers' order lists. And when the food is prepared, the cook will signal the waiter to serve the food to the customer.

```
while(customers are ordering food){
   if(food is not prepared){
      The cook prepares the food and signals the waiter through the app that the food is done
   }


   if(customer finishes eating and pays for food){
      Waiter cleans the table that the customer was at
   }
}
```

6. Plan of Work

We will continue to split up the work done evenly for each part of the reports. After report 1, we will dedicate the rest of our time focusing more on coding our actual application and continuing to be progressive by working on the domain analysis, and the rest of the parts to come.

Functional Feature and Description	Start Date	End Date
Rating System - collecting information from customers on how well the food was prepared, and their rating of it from 1 to 5.	3/13/2017	3/20/2017
Payment System - allow customers to pay through the app, and give them the ability to add tip and split the bill.	2/20/2017	3/20/2017
Seating Chart - have an interactive chart that shows available seats to customers, and “dirty” seats to busboys.	2/27/2017	3/20/2017
Menu - an interactive menu so that customers can view items in greater detail, seeing ingredients, calories, estimated cook time, rating, etc.	2/20/2017	3/13/2017
Food Wait Time - implementing an active system that updates how long the customer has before their food is finished.	3/1/2017	3/20/2017
Inventory Tracker - implementing a system that reduces the amount of raw food in stock based on what has been ordered. A manager can update the inventory through the app as well.	2/27/2017	3/13/2017
Food / Customer Tracker - implementing a system that keep track of the flux of app using customers on a daily to hourly basis, and what they have ordered.	2/27/2017	3/20/2017
Reservation List - App using customers that wish to dine in will be placed on a reservation list, so that they may be easily seated. The hostess will have access to this list to confirm the reservation.	2/27/2017	3/6/2017
Table Alerts - alerting busboys that a table has been recently vacated so that they may come in and clean it up.	2/27/2017	3/6/2017
Waiter Related Alerts - alerting waiters to make sure to check up on tables and confirming the customer is satisfied, and about any food that is ready to be delivered to a table.	2/27/2017	3/13/2017
SpeedByte Server - backend for the app, will have to receive, send, and manipulate data.	2/20/2017	3/20/2017

Plan of Work and Ownership:

Outlined below are the teams, and the proposed work plan over the course of the next few weeks.

TEAM	CODE NAME	MEMBERS
Administration/Management	Team α	Ram, Jose
Employees/Waiters	Team β	Vam, Tim
Customers	Team ¢	Jimmy, Pawel, Husen

SHORT TERM PLAN OF WORK	TEAM
Create a communication bridge between manager and company material via a server	Team β, Team ¢
Create an interactive customer order menu	Team α, Team β
Implement the available table seating chart	Team ¢
Administration information and data trends	Team α
Customer reservations via data capture and storage will populate a list	All

Team Member	**Work Done**	**Current Work**	**Work To Do**
Team α	Functional Requirement and Domain Model	Create an interactive customer order menu	Work on payment system for employees and inventory tracker
Team β	User Interface Specification, Mathematical Model, and UI diagrams	Create a communication bridge between manager and company material via a server	Finalize User Interface and customer menu
Team ¢	CSR, System Requirements, Functional Specification, Plan of Work	Implement the available table seating chart	Finalize waiting order in queue, table availability, and bill payment

7. References

http://www.ece.rutgers.edu/~marsic/Teaching/SE/report1.html

http://www.ece.rutgers.edu/~marsic/Teaching/SE/syllabus.html

http://www.ece.rutgers.edu/~marsic/books/SE/projects/Restaurant/

http://www.ece.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf

www.openclipart.org

Adobe PhotoShop

Argo for UML diagrams