

# ***TurboYums***



**Restaurant Automation Software Suite**

<https://fastfoood.github.io/SEWebsite/>

**Group 13**

**First Report**

Michelle Curreri

Suvranil Ghosh

Ziad Mallah

Anthony Merheb

Roshni Shah

Hersh Shrivastava

Holly Smith

Brandon Tong

Dante Torello

# **Table of Contents**

<b>1 Customer Problem Statement</b>	<b>3</b>
<b>1.1 Problem Statement</b>	<b>3</b>
<b>1.2 Glossary of Terms</b>	<b>6</b>
<b>2 System Requirements</b>	<b>7</b>
<b>2.1 Enumerated Functional Requirements</b>	<b>7</b>
<b>2.2 Enumerated Nonfunctional Requirements</b>	<b>8</b>
<b>2.3 On-Screen Appearance Requirements</b>	<b>9</b>
Customer Interface Requirements	9
Manager Interface Requirements	11
Employee Interface Requirements	12
<b>3. Functional Requirements Specification</b>	<b>12</b>
<b>3.1 Stakeholders</b>	<b>12</b>
<b>3.2 Actors and Goals</b>	<b>12</b>
Initiating Actors	12
Participating Actors	13
<b>3.3 Use Cases</b>	<b>14</b>
<b>3.3.1 Casual Description</b>	<b>14</b>
<b>3.3.2 Use Case Diagram</b>	<b>16</b>
<b>3.3.3 Traceability Matrix</b>	<b>17</b>
<b>3.3.4 Fully Dressed Description</b>	<b>18</b>
<b>3.4 System Sequence Diagrams</b>	<b>22</b>
<b>4. User Interface Specification</b>	<b>23</b>
<b>4.1 Preliminary Design</b>	<b>23</b>
Customer UI	23
<b>4.2 User Effort Estimation</b>	<b>25</b>
Customer Interface	25
Best Case Scenario (Customer Orders a Meal)	25
Worst Case Scenario (Customer Orders a Meal)	25
Employee Interface	26
Best Case Scenario (Chef Serves Customers)	26

Worst Case Scenario (Chef Serves Customers)	26
Best Case Scenario (Employee Serves Customers)	26
Worst Case Scenario (Employee Serves Customers)	27
Manager Interface	27
Best Case Scenario (Manager wants to see who is currently clocked-in)	27
Worst Case Scenario (Manager wants to see who is currently clocked-in)	28
Best Case Scenario (Manager wants to view current orders)	28
Worst Case Scenario (Manager wants to check orders)	28
<b>5. Domain Analysis</b>	29
5.1 Domain Model	29
5.1.1 Concept Definitions	29
5.1.2 Association Definitions	30
5.1.3 Attribute Definitions	31
5.1.4 Traceability Matrix	32
5.1.5 Domain Model Diagram	33
5.2 System Operation Contracts	34
Calculated Use Case Points (UCP)	37
<b>7 Plan of Work</b>	37
<b>8 References</b>	38

# **1 Customer Problem Statement**

## **1.1 Problem Statement**

### **Chef:**

Working in a kitchen can be strenuous and it can be difficult to keep track of everything that needs to be done. Orders need to be filled in a reasonable amount of time while simultaneously ensuring that any dish accommodations are accounted for. This needs to be done with accuracy or else an order could be made incorrectly or even worse, a person with a food allergy could get hurt as a result of a waiter's poor handwriting. We want to get our orders out with accuracy and speed to make sure we have a happy returning customer. When dishes are complete and the waiter does not pick it up for a few minutes, it is a pain because the food gets colder and it takes up space on the counter that could be used otherwise. Also, at the end of a busy day of serving hundreds of customers the inventory can be depleted to the point of needing restocking and it's difficult to keep track of what's expired and what I need more of. I have so many responsibilities that it can be difficult to keep track of everything in the pantry. I have to make sure that the expired ingredients are cycled out and that the low stock ingredients are replenished in time for the upcoming week. It would be helpful to have a tool that would track the expiration dates of my food to make sure everything is fresh.

It would be a great help if I had something that clearly displays all incoming orders and the modifications that have been requested by the customers. Also, if multiple dishes come in with the same ingredients, it will be helpful if the chef was notified to make more of a certain ingredient at one time versus cooking a certain part of a dish twice. This can save me more time in the kitchen and make the cooking process more efficient. Finally, a way to signal to the waitresses and waiters that the order is finished and ready to be taken to the table to reduce the time the food sits on the counter.

### **Host/Hostess:**

Whenever I am welcoming customers into our restaurant, I aim to make the start of their experience as flawless as possible. As parties of multiple people arrive, they tend to have different requests and requirements based on how many people they have and where they want to sit. Sometimes you can have a party of 8 looking for a round table in a corner, where as other times you can get a family of 4 wanting a booth. I love to find them their preferred seating, but sometimes there just isn't an available spot, and we aren't really sure how long it will be for an appropriate spot to open up.

Finding an exceptional table for guests can sometimes be a challenge, especially when I'm unsure of which tables are ready to go. When I'm welcoming customers in, it would be great if there was a way to easily keep track of which tables are currently cleaned and empty and which are occupied, without having to go through

endless reams of paper. At the same time, it'd be nice to keep track of how long each table has already been occupied. This will be a big help in providing guests with accurate estimations on when a valid seating arrangement may become available.

How will TurboYums help me? Using a handy tablet/phone I can keep on me, I can mark tables as occupied, vacant, clean, etc. as the day goes on. Whenever I seat new guests, I can mark the table as occupied. This starts a timer that lets us know how long it has been since guests have arrived there. Once the guests leave, that table can be marked as vacant, notifying a busboy that the table is ready to be cleaned. Once the busboy is finished, they can mark it as cleaned which lets me know that the table is ready to be used once again.

### **Customer (Restaurant Patron):**

I love going out to eat, one thing I don't love is how slow things can be. When I'm seated, it'd be really great if I could place an order for drinks or appetizers without waiting for a waiter to get to me, especially if the restaurant is incredibly busy this could really cut down on time spent waiting. Even if I have a special request immediately after my initial order or need to notify the waiter for anything, it can be rough getting someone's attention. Which reminds me, waiting for a server can be a major inconvenience while dining out. It's usually very difficult to get their attention unless they walk right past you, and if no one ever walks by, you might be waiting extended periods of time before anyone notices you. It can also be difficult to split a bill while out with friends, often times it doesn't get mentioned until the very end of the outing, and it becomes an awkward experience determining with the server what the best way to split the bill is. Some people prefer to split it by itemizing and some by simply splitting the total bill at the end. When this hasn't been properly planned it can be difficult to figure out how to fix it.

How will TurboYums help me? I think a tablet on the table with a menu that my party could use to place orders immediately after being seated would be a great idea. It would also be a great way to have more interactive menus that could give me more information on each item, since if I have any questions, I usually have to wait for a server to come answer. A great solution to waiting for a server to walk by to help you, would be the ability to press a button in an app to request a server visits your table. Features related to bill splitting would also really help, at the conclusion of our evening the tablet could present an interactive interface where we can leave our thoughts about our experience.

### **Waiter/Waitress:**

Working as a waiter/waitress is an extremely demanding job. Servers have to run around the restaurant taking customers orders, keeping track of which table ordered what, and returning orders. Servers have to interact with chefs/cooks and get food to people right away so the food does not get cold. They also have to enter checks when a customer or group is ready to pay. Since there are many tables in this restaurant, it is somewhat difficult to connect a certain order to a certain table, and also to determine whether a table needs to be cleaned and set up for the next customer. Unfortunately, sometimes a waiter may have to send a dish back if the customer does not like it or finds something wrong. The software should be able to account for this.

There are many ways in which a computer software can help make servers' lives easier everyday at work. A software suite could enter all the orders in a list of the servers', as well as number each table in this restaurant and match it to a given order. Additionally, when the customers are done using the table, the software would mark the table "not clean" and so the waiters will clean it or send the busboy to clean it. If the customers need anything from me, they can also have an option to send me a ping which will come through as a push notification to my device, allowing me to know they need service. Server and chef coordination are also very important. The chef should be able to enter a time of completing a dish, and the server should be able to see how much time the dish has been out. The software then sends a reminder to the waiter/waitress to serve an order. Lastly, the system could automatically take payments and tips.

### **Managers:**

Managing a restaurant and a full staff of employees is no easy task, there are many different things that I have to do throughout my day and it can be a little overwhelming at times. I am really hoping that TurboYums will be able to help me with ensuring that the restaurant is adequately stocked at all times by actively keeping track of inventory since the counts provided by the busboys never actually lineup with when we run out of supplies and therefore I am constantly guessing at when the proper time to restock is which can lead to scrambling around at the last second to fulfill a customer request.

There are also difficulties with figuring out how many employees I need to schedule for a given day and time, it seems that over the weekends we generally need more staff, but I am not sure if there are trends throughout the week and some days are busier than others because I do sometimes find that we are understaffed, but I cannot seem to find the proper balance of workers to work. Another issue with employees is keeping track of them, like whether or not they made it to their shift on time or are in the building and calculating how much to compensate the employees for their time. Any way to reduce the amount of effort that I have to put into any of these tasks would be immensely helpful.

How will TurboYums help me? Utilizing the TurboYums Android application once the order has been placed via the Android device that has the software on it, the ingredients used to cook the meal will automatically be deducted from inventory so that inventory is always up to date and accurate, if the user has a manager account then they may input a threshold value for items in inventory, and when supplies fall below that threshold a notification will be sent to the manager, informing the manager that they need to pick up more supplies. In addition to this, the application will track the traffic in the restaurant allowing management to determine if there are any trends with when people eat at the restaurant so that they are not over or under staffed. TurboYums also serves as an employee portal, allowing them to clock in and out, log break time, and will calculate the proper amount to provide to them for compensation. The application will also verify the employees' location based on IP address or GPS location so that employees can only clock in while they are on site.

## 1.2 Glossary of Terms

- **Application** - A software designed to perform a group of coordinated functions, tasks or activities for the benefit of the user.
- **Bartender** - Prepares and mixes drinks that are requested by the customers.
- **Busboy** - Clears tables, take dirty dishes to the dishwasher and set up the tables to be occupied again.
- **Chef** - Makes food that is requested by the customers.
- **Customer** - Someone that comes to the restaurant as a guest to be waited on and served food.
- **Employee Portal** - Allows employees to clock into work from our system.
- **Floor Layout** - Shows how tables in the restaurant are placed and the tables status.
- **Host/Hostess** - Greets incoming customers and assigns seats to each customer.
- **Inventory** - A list of available items available for use, such as goods that are currently in stock and available property owned by the establishment.
- **Menu** - A list of food available to be prepared, cooked, and presented, including pictures and ratings.
- **Manager** - Supervises all staff on board and makes sure that everything is in working order.
- **Manager Account** - A user account that allows access to the data collected by the application as well as having extra features that allows management of employees and the restaurant.
- **Operating System** - System software on which the application will run.
- **Queue** - A first in, first out way of handling orders to ensure that food arrives in order that it is requested.
- **Rating** - A position or standing of something determined by the customers feedback.
- **Reservation** - An arrangement made in advance to secure a table.
- **Restaurant Automation** - Makes a restaurant flow more efficiently and more easily. Eliminates certain tasks that waiters would normal have to do. Uses devices with preloaded software that manages

several tasks which helps eliminate many of the required tasks that are normally done via employees.

- **Tablet** - A portable thin computer that utilizes a touchscreen as its primary user interface.
- **Tip** - A sum of money given to waiter/waitress in addition to the base price.
- **User Interface** - The visual aspect of the software that allows user interaction.
- **Waiter/Waitress** - Takes orders from customers and delivers food from the kitchen to the customer's table.

## 2 System Requirements

### 2.1 Enumerated Functional Requirements

Note: Priority 5 is more vital than Priority 1

Identifier:	Priority:	Requirement:
REQ-1	4	The application will allow customers to select an open table based on an interactive seating chart, after the user has logged in. Once selected, the table is marked as "Taken".
REQ-2	5	The application will present an interactive graphical menu once the customer is seated at the table.
REQ-3	5	The application will provide employees with an Employee Portal.
REQ-4	5	The application will keep track of all employee hours for payroll based on clock-in/clock-out reports.
REQ-5	3	The application will notify the busboy when a customer has paid their bill to indicate that they are leaving, so the busboy can clean the table.
REQ-6	3	The application will allow for managers and waiters to quickly adjust the floor map to allow for the most accurate representation of the tables on the floor at all times.
REQ-7	4	The application will allow for customers to split the bill
REQ-8	3	The application will provide the option to make a reservation at the restaurant ahead of time.
REQ-9	4	The application will notify and send the incoming orders to the chef, adding it to the end of the queue in order of time placed.



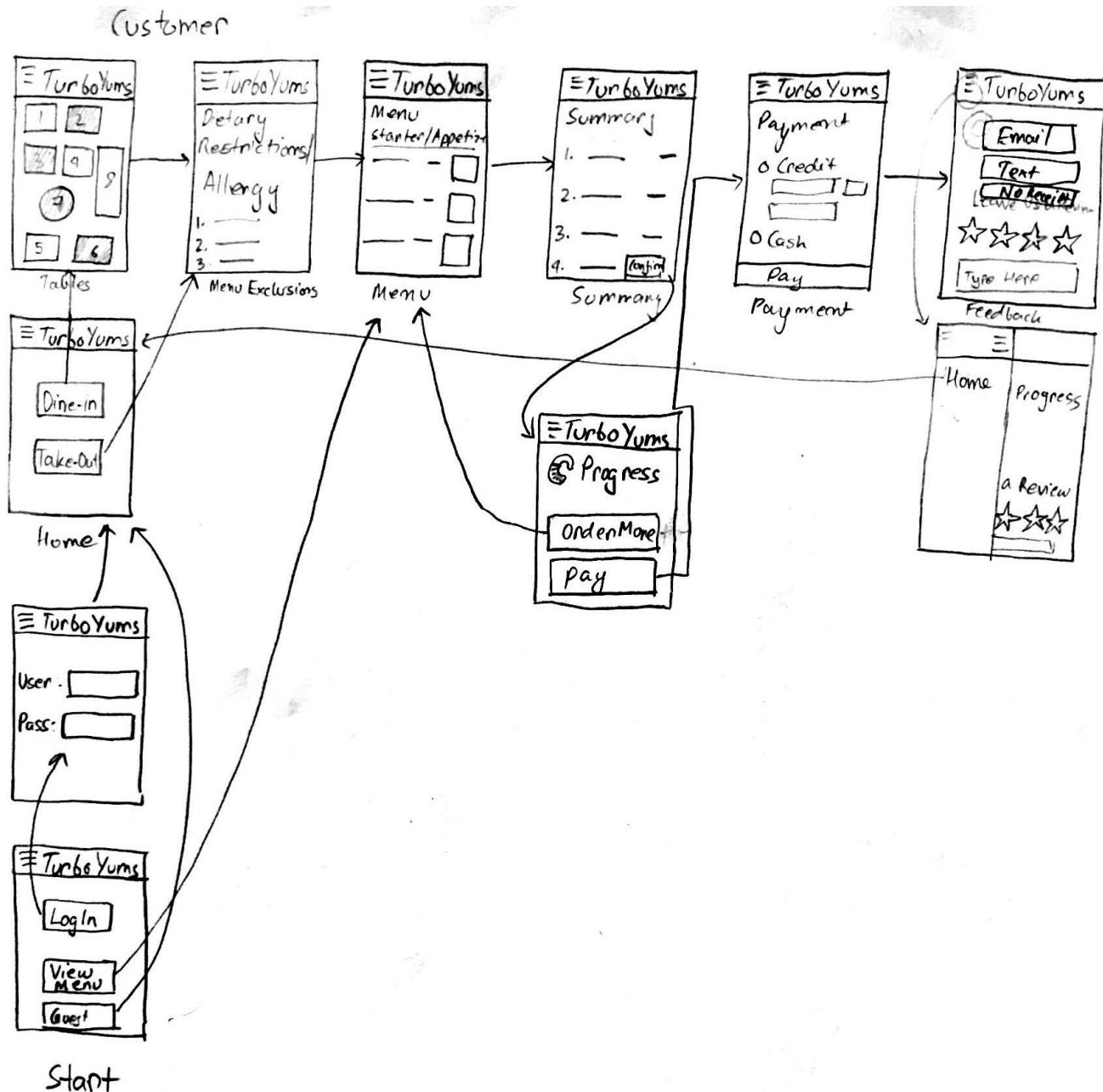
REQ-10	3	The application will allow a “Take-Out” option where the take-out orders are placed and taken out.
REQ-11	4	The application will allow the customer to sign-up to be a part of a Rewards program.
REQ-12	4	The application should add points earned every time the customer makes a purchase.
REQ-13	4	The application should deduct points and apply them appropriately to the total bill when the user decides to use them.
REQ-14	4	The application will notify the waiter/waitress when a customer’s food is ready, so they can deliver it to the designated table.
REQ-15	3	The application will allow the option for the menu to be translated, if needed.
REQ-16	5	The application will allow customers to filter menu items according to dietary restrictions and notify the chef when the order is sent.
REQ-17	2	The application will allow the customer to rate and leave comments on the menu items at the end of the meal.
REQ-18	3	The application will allow for waiters/waitresses to mark tables as “Taken”.
REQ-19	4	The application will allow the customer to alert the waiter/waitress, using a button on the application.
REQ-20	3	The application will allow a table to be marked as “Open” once the busboy has cleaned it.
REQ-21	5	The application will allow the customer to pay on the spot with a credit-card reader, if desired.
REQ-22	1	The application will give the option of receipt choice (i.e. paper, e-mail, none).

## 2.2 Enumerated Nonfunctional Requirements

Identifier:	Priority:	Requirement:
REQ-23	5	After logging in or continuing as a guest, and selecting the “Dine- in” option, the customer will be presented with a floor layout that designates the available tables as white and the unavailable tables as grey, and will allow the customer to select an available table to sit at. The application should be compatible with only Android operating systems.

REQ-24	3	The application should be easy to use.
REQ-25	3	The application should be aesthetically pleasing.
REQ-26	2	The application should be able to startup and shut down quickly.
REQ-27	3	The application should have smooth transitions from page to page.
REQ-28	4	The application should support a functioning restaurant.

## 2.3 On-Screen Appearance Requirements

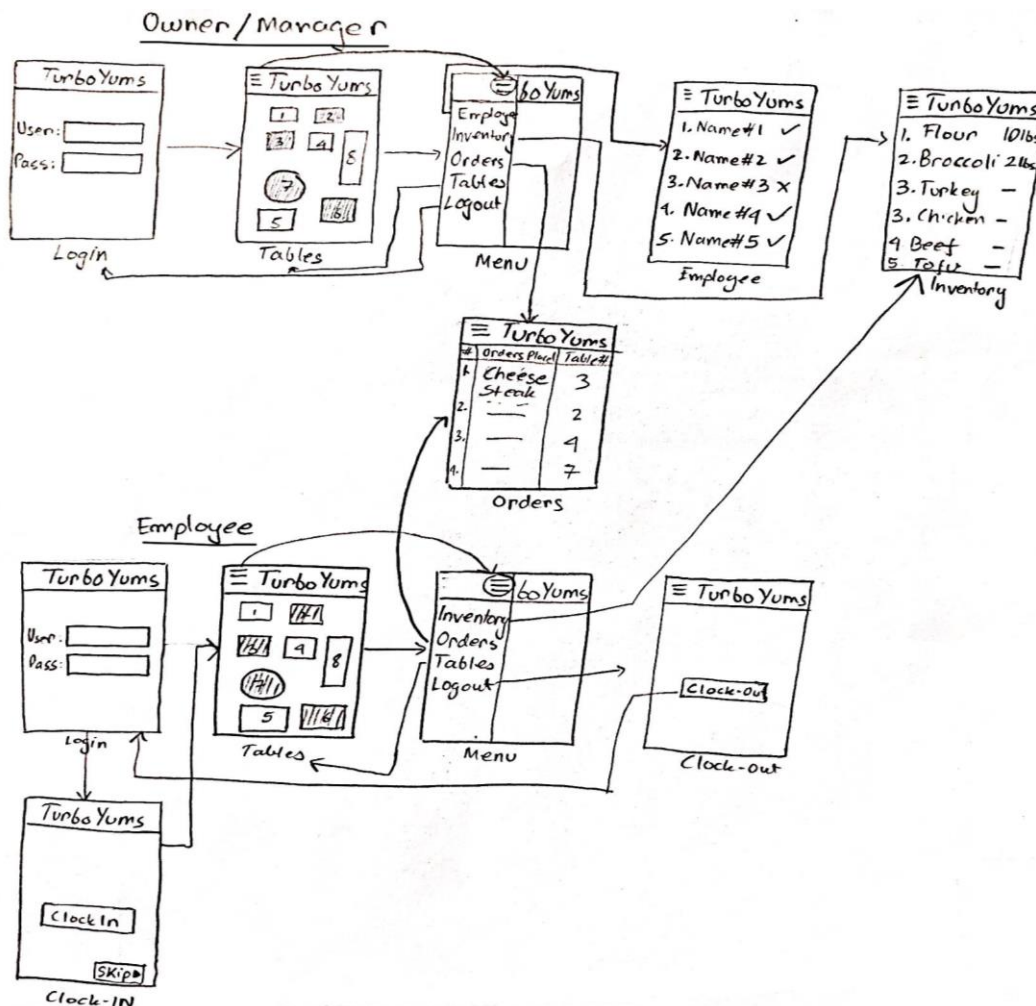


## Customer Interface Requirements

Identifier:	Priority:	Requirement:
CSREQ-1	4	Customer gets to choose among “Log In”, “Guest Login” and “View Menu”
CSREQ-2	4	Clicking on the “Login” button takes the customer to the Login Page where he/she inputs her account username and password.
CSREQ-3	5	Clicking on the “Guest Login” button takes customer to a page where he/she gets to choose between Dine-In and Take-Out.
CSREQ-4	5	Clicking on the “View Menu” button takes the customer to a Menu Page <b>without</b> “Add to Cart” option to prevent checking out without either logging in or continuing as a guest.
CSREQ-5	4	Clicking on Dine-In takes the customer to the Tables page so that he/she can choose a table to sit at.
CSREQ-6	5	Clicking on an unoccupied ready table (colored in white) takes the customer to the Menu Exclusion page where he/she can filter out the menu according to his/her dietary restrictions/allergies. Clicking on an occupied or unready table (colored in grey) is forbidden.
CSREQ-7	4	Clicking on the Take-Out button will take the customer straight to the Menu Exclusion page.
CSREQ-8	5	After the confirmation of the customers Dietary Restriction/Allergies, the customer can head to the Menu page which is already filtered out.
CSREQ-9	3	After selection from the menu, the customer gets a summary/overview of her selections in the summary page and he/she can proceed to Progress page.
CSREQ-10	1	In the Progress Page, the customer can see the progress of his/her order and Order More or Checkout/Pay.
CSREQ-11	5	Clicking on “Payment” takes customer to the Payment page where he/she can either pay using Credit Card or Cash.
CSREQ-12	2	After payment, the Receipt/Feedback page appears where the customer can choose how they want their receipt or not and rate his/her experience of ordering via the app.
CSREQ-13	3	Clicking on the Home page, the customer can go back to the dine-in/take-out page where she can place another order - let’s say, for takeout.

## Manager Interface Requirements

Identifier:	Priority:	Requirement:
MSREQ-1	5	Owner/Manager logs in with his/her Username and Password.
MSREQ-2	4	Default page of Tables appears after login to show which tables are empty and which are not.
MSREQ-3	5	Clicking on the app menu takes the owner to a list of pages available for him to view: Employees Info, Inventory, Orders, Tables and Logout.
MSREQ-4	5	Employee Info takes the owner to view a list of all his employees and shows who're working at the moment and who are not.
MSREQ-5	4	Clicking on the Tables page takes the owner to the Tables page where he/she can see which tables are occupied (grey), unclean (red) or ready to go (white).
MSREQ-6	4	Log Out takes the owner back to the Login Page.



## Employee Interface Requirements

Identifier:	Priority:	Requirement:
ESREQ-1	5	Employee logs in with his/her Username and Password.
ESREQ-2	5	The Employee clocks in or skips. The time gets logged into the server under the employee's account.
ESREQ-3	4	Default page of Tables appears after login to show which tables are empty and which are not.
ESREQ-4	4	Clicking on the app menu takes the employee to a list of pages available for him/her to view: Orders, Tables and Logout. These pages are the same as the one's under the Owner/Manager mode except for the LogOut.
ESREQ-5	4	The Employee gets to the Clock-Out Page after clicking on LogOut. This is only if the employee had clocked in. Otherwise the user will be directed straight to the Log In Page

## 3. Functional Requirements Specification

### 3.1 Stakeholders

There are many stakeholders who have an interest in this system, and ultimately its success:

- i. Restaurant Owners - Have a direct interest in using the system as it will help optimize efficiency in the restaurant and provide better service for patrons.
- ii. Employees - Have an interest in the system since it will result in a more streamlined process while working making their job much easier.
- iii. Restaurant Visitors - The system will result in an enriched dining experience for the restaurant visitor, as they will be interacting with and using the system.
- iv. Developers - Have an interest in working to design and implement solutions to create the system.

### 3.2 Actors and Goals

#### Initiating Actors

Actor	Role	Goal
Customer	The customer is a restaurant visitor who may choose to dine in or take-out food, view the menu, order a meal, eat, and pay for service.	The goal of the customer is to have an excellent dining experience with minimal wait times and smooth service.

Guest	The guest is a customer that does not have an account and chooses to opt out of the feature that come with an account, other than that a guest is the same of a customer.	The goal of the guest is to have an excellent dining experience with minimal wait times and smooth service.
Employee	The employee is any type of worker at the restaurant, except for the manager.	The goal of the employee is to provide an excellent dining experience for the customers.
Manager	The manager is the employee who has the additional responsibility of managing all needs of the restaurant.	The goal of the manager is to manage employees and scheduling, keeping track of inventory, and monitoring revenue and losses while they also ensure that restaurant customers are accounted for.

### Participating Actors

Actor	Role
Bartender	The bartender is the employee responsible for preparing and mixing drinks that are ordered by a customer. The bartender receives a queue of customer orders, and begins mixing the drinks accordingly. Once complete, the drink is brought to the customer's table via the waiter.
Busboy	The busboy is the employee responsible for cleaning the dishes and tables, and maintaining overall cleanliness of the restaurant. The busboy receives notifications from the database that customers have left their table, marking the table as dirty. Once the table is cleaned, it will be marked as clean.
Chef	The chef is the employee responsible for cooking and preparing the food that is ordered by a customer. The chef receives a queue of orders, preparing them in the order they come in. The customer is updated on the status of their order (submitted, preparing, ready).
Database	The database is a system that records a customer's order, reservation/table selection, and menu ratings. It essentially acts as the server in which the application receives most of its information for automation.
Host/Hostess	The host/hostess is the employee responsible for greeting incoming customers and assigning seats to them. In the event that the guests have already reserved a table, the host/hostess will escort them to the table and marked it as occupied. The host/hostess receives notifications when a table is marked as clean.
Waiter/Waitress	The waiter/waitress is the employee responsible for taking orders from customers and sending them to the kitchen queue, as well as serving the food when it is ready. The waiter/waitress receives notifications that a meal is ready, so that they can pick it up and serve it. They are also notified which table number to serve it to, and get notified when a customer needs additional assistance.

## 3.3 Use Cases

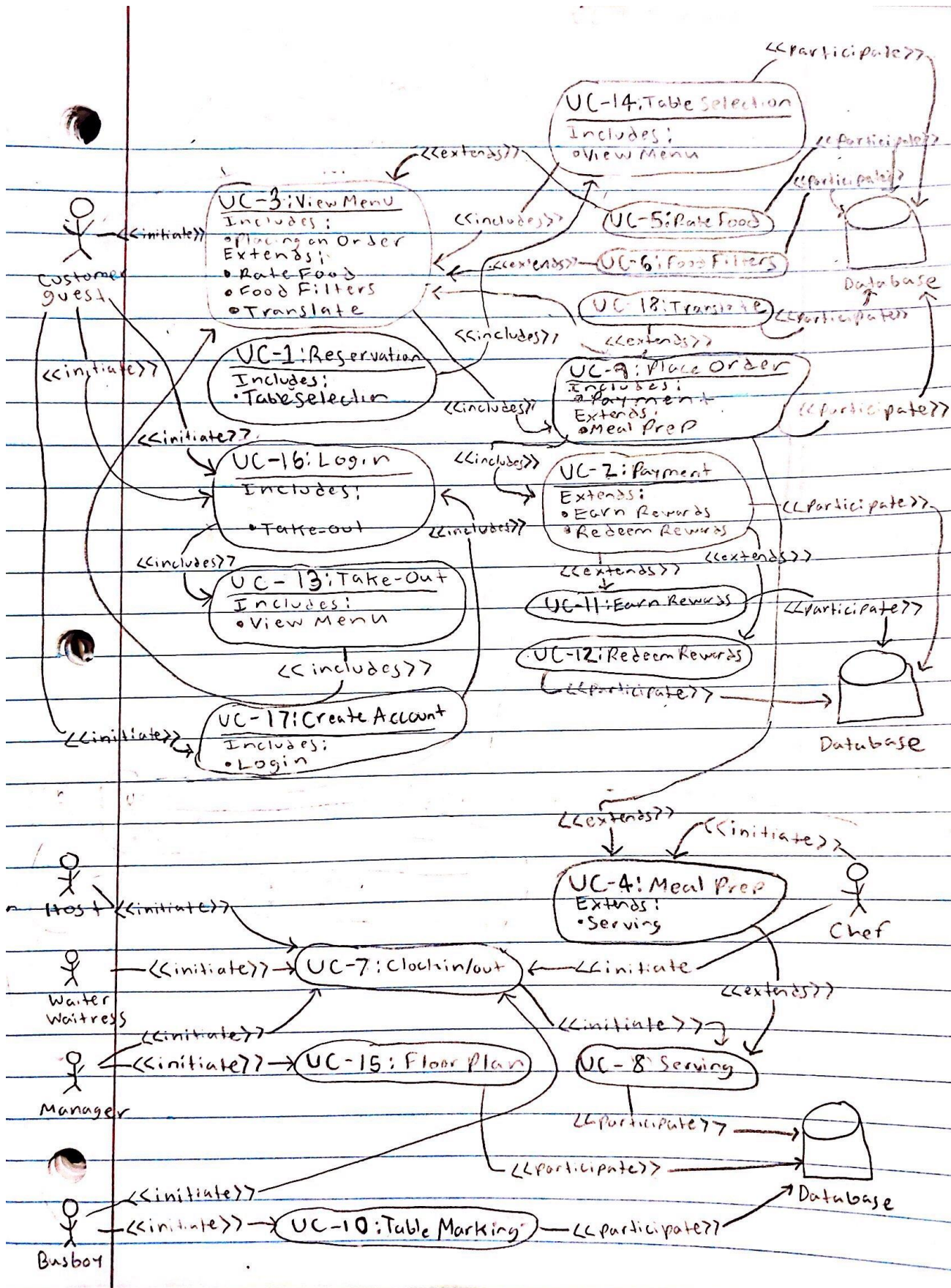
### 3.3.1 Casual Description

- **UC-1: Reservation** - Allows customers to make reservations online before they come to the restaurant to reserve a table.  
Derivations: REQ-8
- **UC-2: Payment** - Allows customers to split the bill, choose option of receipt (email, paper) and allow customer to pay on the spot with a credit-card reader (if desired)  
Derivations: REQ-5, REQ-7, REQ-12, REQ-13, REQ-21, REQ-22, CSREQ-10, CSREQ-11, CSREQ-12
- **UC-3: View Menu** - Allows customers to view the entire menu with our system.  
Derivations: REQ-2, REQ-15, REQ-16, CSREQ-4, CSREQ-8
- **UC-4: Meal Prep** - Allows chefs to be updated on the status and requirements of their orders.  
Derivations: REQ-9, REQ-16, ESREQ- 4
- **UC-5: Rate Food** - Allows customers to rate food items on the menu and view their current ratings.  
Derivations: REQ-17, CSREQ-12
- **UC-6: Food Filters** - Allows customers to filter out food according to dietary restrictions or preferences.  
Derivations: REQ-16, CSREQ-8
- **UC-7: Clocking In/Out** - Allows employees to clock in when they come to work and clock out when they go on break/go home.  
Derivations: REQ-3, REQ-4, ESREQ-1, ESREQ-2, ESREQ-4, ESREQ-5, MSREQ-4
- **UC-8: Serving** - Allows waiters/waitresses to keep track of their current orders and customer status.  
Derivations: REQ-6, REQ-14, REQ-18, REQ-19, REQ-20, ESREQ-4
- **UC-9: Placing an Order** - Allows waiter/waitresses or customers to send their order to the kitchen  
Derivations: REQ-9, ESREQ-4, CSREQ-10
- **UC-10: Table Marking** - Allows waiters/waitresses to mark a table as taken when customers sit down and allows busboys to mark a table as open once it is cleaned.  
Derivations: REQ-1, MSREQ-5, ESREQ-3, ESREQ-4

- **UC-11: Earning Rewards** - When the customer makes a purchase, they earn rewards. With a certain amount of rewards the customer can receive discounts or a free drink as an example.  
Derivations: REQ-11, REQ-12, REQ-13
- **UC-12: Redeeming Rewards** - When the customer makes a purchase, they can use points earned toward the cost of their meal.  
Derivations: REQ-12, REQ-13
- **UC-13: Take-Out** - An option for the customers that allows them to order food from the restaurant for pick up to take home.  
Derivations: REQ-10, CSREQ-7
- **UC- 14: Table Selection** - Allows the customer to select a table after either logging in or continuing as a guest by viewing the floor plan, which will have the status of the tables shown and allowing them to select from all of the open tables.  
Derivations: REQ-1, REQ-18, CSREQ-5
- **UC-15: Floor Plan Adjustment** - Allows the manager and the waiters to adjust the table layout on the floor plan by viewing the floor plan and holding their finger on a specific table for a period of time which will then allow the table to be moved around on screen and placed in the proper location.  
Derivations: REQ-6, MSREQ-2, MSREQ-3, MSREQ-5
- **UC-16: Login** - Allows users to login which will dictate which interfaces they will view and the different potential functions that they may access.  
Derivations: REQ-3, REQ-11, CSREQ-1, CSREQ-2, CSREQ-3, MSREQ-1, ESREQ-1
- **UC-17: Create Account** - Allows users to create an account so that they may get the benefits of being an account holder.  
Derivations: REQ-11, CSREQ-1
- **UC-18: Translation** - Allows user to have the option to translate the menu into another language, so they don't need a translator.  
Derivations: REQ-15



### 3.3.2 Use Case Diagram



### 3.3.3 Traceability Matrix

Requirements	PW	1	2	3	4	5	6	7	8	9	10	11	12	3	14	15	16	17	18
REQ-1	4										X				X				
REQ-2	5			X															
REQ-3	5							X									X		
REQ-4	5							X											
REQ-5	3		X																
REQ-6	3								X							X			
REQ-7	4		X																
REQ-8	3	X																	
REQ-9	4				X					X									
REQ-10	3													X					
REQ-11	4											X					X	X	
REQ-12	4		X									X	X						
REQ-13	4		X									X	X						
REQ-14	4								X										
REQ-15	3			X															X
REQ-16	5			X	X		X												
REQ-17	2					X													
REQ-18	3								X						X				
REQ-19	4								X										
REQ-20	3								X										
REQ-21	5		X																
REQ-22	1		X																
CSREQ-1	4																X	X	
CSREQ-2	4																X		
CSREQ-3	5																X		
CSREQ-4	5			X															
CSREQ-5	4														X				
CSREQ-6	5																		
CSREQ-7	4													X					
CSREQ-8	5			X			X												
CSREQ-9	3																		
CSREQ-10	1		X							X									
CSREQ-11	5		X																
CSREQ-12	2		X			X													
CSREQ-13	3																		
MSREQ-1	5																X		
MSREQ-2	4															X			
MSREQ-3	5															X			
MSREQ-4	5							X											
MSREQ-5	4										X					X			
MSREQ-6	4																		
ESREQ-1	5							X									X		
ESREQ-2	5							X											
ESREQ-3	4										X								
ESREQ-4	4				X			X	X	X	X								
ESREQ-5	4							X											

### 3.3.4 Fully Dressed Description

UC-2: Payment
<b>Related Requirements:</b> REQ-5, REQ-7, REQ-12, REQ-13, REQ-22, CSREQ-10, CSREQ-11, CSREQ-12
<b>Initiating Actor:</b> Customer
<b>Actor's Goal:</b> To pay for a completed order
<b>Participating Actors:</b> Database Waiter/Waitress
<b>Preconditions:</b> The user has loaded the system The user has logged in as a Customer and has the menu opened
<b>Postconditions:</b> The user is prompted on payment method The user is prompted on receipt option
<b>Flow of Events for Main Success Scenario:</b> <ol style="list-style-type: none"><li>→ The customer clicks on the "Payment" button on the menu screen.</li><li>← The system displays the payment menu.</li><li>← The system displays options for splitting the bill and shows payment method options.</li><li>→ The customer chooses the "Cash" button in order to pay by cash.</li><li>← The system prompts the customer with receipt options.</li><li>→ The customer chooses the "Print Receipt" button.</li><li>← The system notifies the waiter/waitress to print the receipt and bring it back to the customer.</li></ol>
<b>Flow of Events for Alternate Success Scenario:</b> <ol style="list-style-type: none"><li>→ The customer clicks on the "Payment" button on the menu screen.</li><li>← The system displays the payment menu.</li><li>← The system displays options for splitting the bill and shows payment method options.</li><li>→ The customer chooses the "Credit" button in order to pay by credit.</li><li>← The system prompts the customer with receipt options.</li><li>→ The customer chooses the "E-Mail Receipt" button.</li><li>← The system prompts the user for an e-mail address.</li><li>→ The customer types in an E-mail address and receives a receipt through E-mail.</li></ol>

UC-3: View Menu
<b>Related Requirements:</b> REQ-2, REQ-15, REQ-16, CSREQ-4, CSREQ-8
<b>Initiating Actor:</b> Customer
<b>Actor's Goal:</b> To view the menu
<b>Participating Actors:</b> Database
<b>Preconditions:</b> The user has loaded the system
<b>Postconditions:</b> The user is shown the available menu
<b>Flow of Events for Main Success Scenario:</b> <ol style="list-style-type: none"> <li>← The system gives options for "Log-In", "Guest", or "View Menu".</li> <li>→ The customer selects to login to a Rewards account.</li> <li>← The system prompts the customer to login with their information.</li> <li>→ The customer inputs information for Rewards account.</li> <li>← The system gives options for "Dine-In" or "Take-Out".</li> <li>→ The customer selects "Dine-In".</li> <li>← The system allows the customer to select a table from the seating chart.</li> <li>→ The customer sits at the table.</li> <li>← The system provides options to input Dietary Restrictions.</li> <li>→ The customer chooses to not filter the menu.</li> <li>← The system displays the available menu.</li> </ol>
<b>Flow of Events for Alternate Success Scenario:</b> <ol style="list-style-type: none"> <li>← The system gives options for "Log-In", "Sign-Up", "Guest", or "View Menu".</li> <li>→ The customer selects "View Menu".</li> <li>← The system displays the available menu.</li> </ol>

UC-7: Clocking In/Clocking Out
<b>Related Requirements:</b> REQ-3, REQ-4, ESREQ-1, ESREQ-2, ESREQ-4, ESREQ-5, MSREQ-4
<b>Initiating Actor:</b> Manager

## UC-7: Clocking In/Clocking Out (Contd.)

**Actor's Goal:**

To accurately record the amount of time that each employee spends while on the job and to make sure the employee is actually at the restaurant when they clock-in and out

**Participating Actors:**

All Employees (i.e. Bartender, Busboy, etc.)

**Preconditions:**

The user has loaded the system

**Postconditions:**

The user has successfully clocked in/out

The database saves the information for future use

**Flow of Events for Main Success Scenario:**

1. ← The system asks the employee to login.
2. → The employee logs in as regular employee at the beginning of the shift.
3. ← The system prompts the employee for their login information.
4. → The employee logs in with their information.
5. ← The system keeps track of the employee's clock-in data.
6. ← The system verifies that the employee is actually in the restaurant by checking the IP address and GPS coordinates.
7. → When a break is needed, the employee chooses the break option and clocks out for their break.
8. → After break is completed, the employee clocks back in.
9. ← The system keeps track of the employee's break data.
10. ← The system verifies that the employee is actually in the restaurant by checking the IP address.
11. → When the employee has completed the shift for the day, the employee will clock out/log out.
12. ← The system keeps track of the employee's clock-out data.
13. ← The system verifies that the employee is actually in the restaurant by checking the IP address.

**Flow of Events for Alternate Success Scenario:**

1. ← The system asks the employee to login
2. → The manager logs in as manager at the beginning of the shift.
3. ← The system prompts the manager for their login information.
4. → The manager logs in with their information.
5. ← The system keeps track of the manager's clock-in data.
6. ← The system verifies that the employee is actually in the restaurant by checking the IP address.
7. → When a break is needed, the manager chooses the break option and clocks out for their break.
8. → After break is completed, the manager clocks back in.
9. ← The system keeps track of the manager's break data.
10. ← The system verifies that the employee is actually in the restaurant by checking the IP address.
11. → When the manager has completed the shift for the day, the manager will clock out/log out.
12. ← The system keeps track of the manager's clock-out data.
13. ← The system verifies that the employee is actually in the restaurant by checking the IP address.

## UC-11: Earning/Redeeming Rewards

**Related Requirements:**

REQ-11, REQ-12, REQ-13

**Initiating Actor:**

Customer

**Actor's Goal:**

To receive coupons/discounts based off the amount of purchases at restaurant

**Participating Actors:**

Database

**Preconditions:**

The user has loaded the system

The user intends to make a purchase

**Postconditions:**

The user has earned an appropriate amount of points for how much was spent

**Flow of Events for Main Success Scenario:**

1. ← The system asks the customer at the beginning to login/create account
2. → The database locates where that user points are
3. ← The customer chooses all the food items and confirms order
4. ← The customer asks to see coupon/discount that are available based off points
5. → The database updates point and price if customer uses coupon
6. ← Customer confirms order
7. → The database calculates the amount of points of based off amount paid
8. → The database takes the new points and adds it to the users previous points

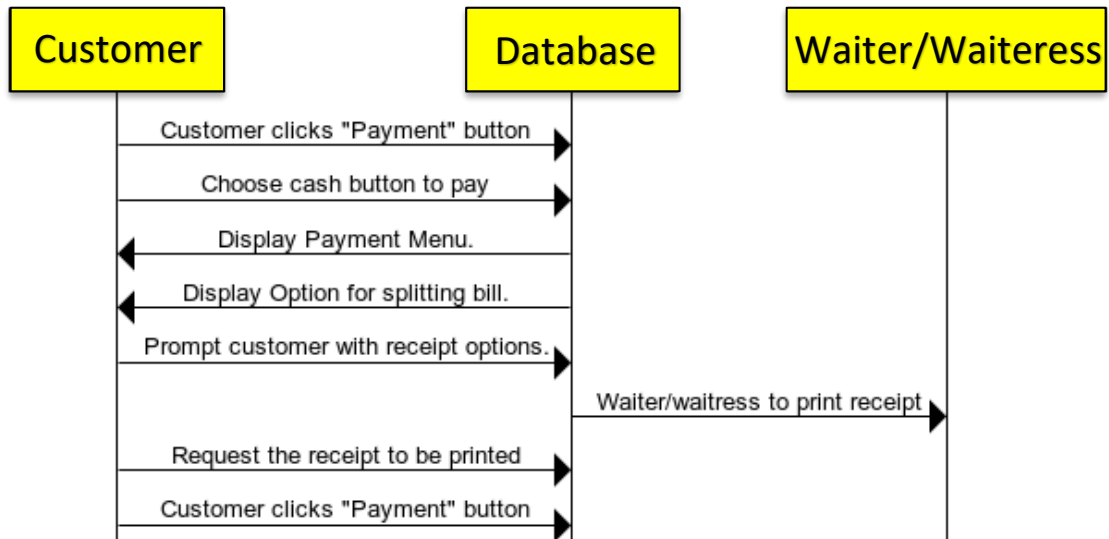
**Flow of Events for Alternate Success Scenario:**

1. ← The system prompts the customer at the beginning to login/create account
2. → The database locates where the user points are based off account
3. → The customer confirms order
4. ← The database checks if user is eligible for coupon/discounts
5. → Customer can choose which coupon/discount to use for the order
6. ← Database adjust points if coupon is used
7. → Customer pays for order
8. ← The database calculates the amount of points based off price
9. ← The database adds points to users account

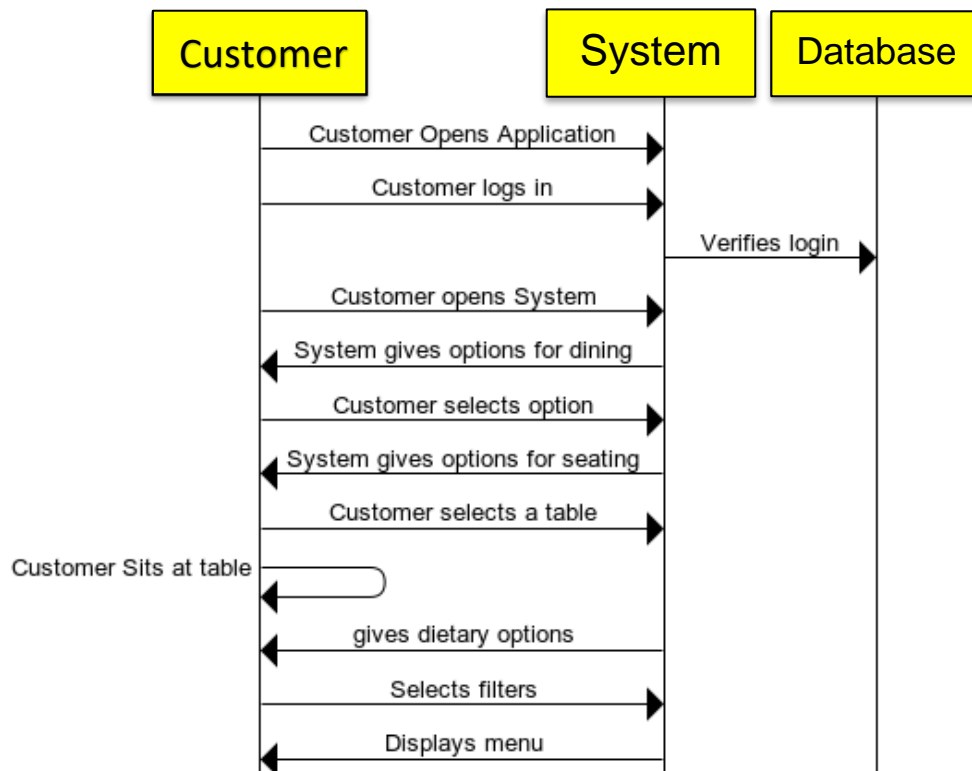


## 3.4 System Sequence Diagrams

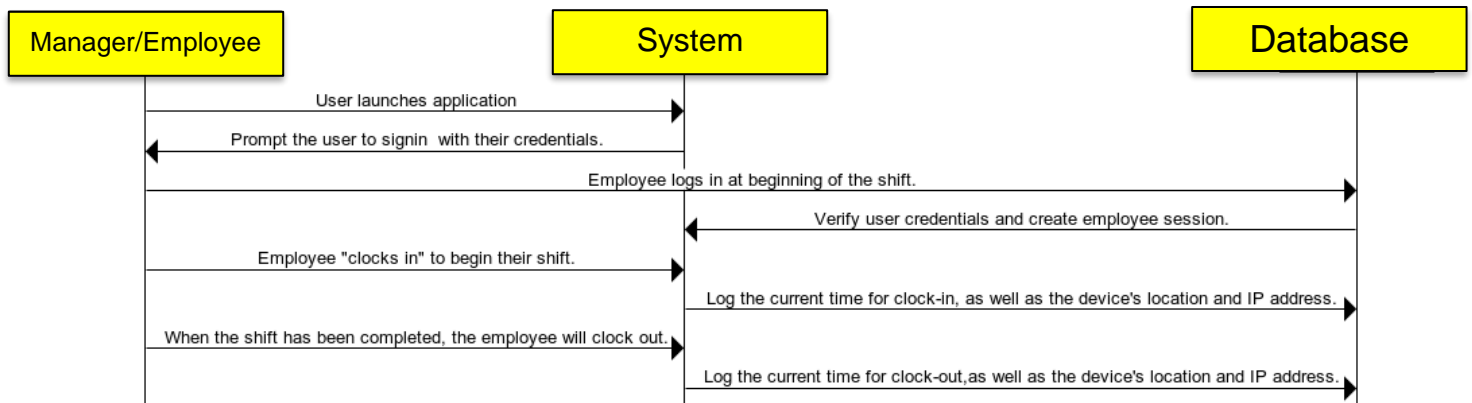
### UC-3: Payment



### UC-4: View Menu



#### UC-7: Clocking In/Clocking Out



## 4. User Interface Specification

### 4.1 Preliminary Design

#### Customer UI

This is a step by step process on how a customer can use TurboYums to order food for takeout or for dine-in, pay for the food, get the option of getting the receipt for the order (or not) and leave a feedback of his/her experience.

Upon opening the app, the user is greeted by a splash screen which should look like Fig 1:



Fig 1

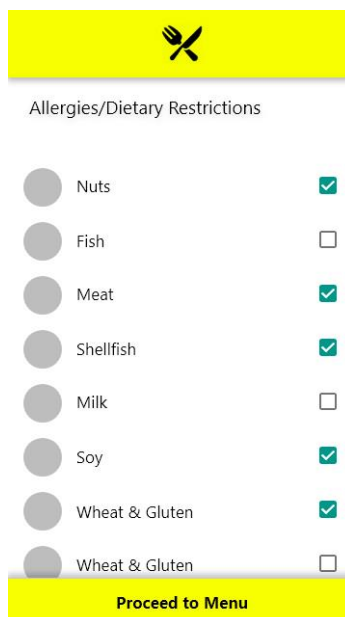


Fig 2

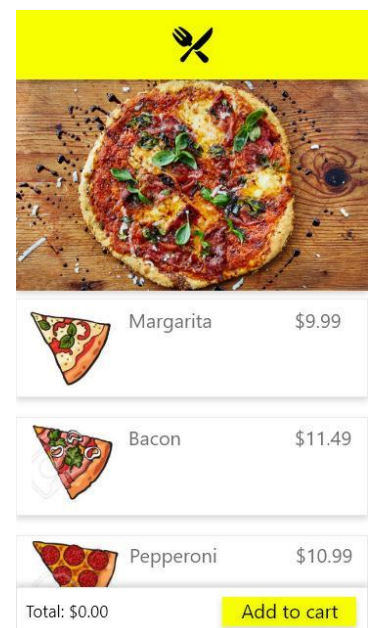


Fig 3

Immediately after the splash screen, the user is asked to choose his/her allergies so that the menu can be filtered out as shown in Fig 2.



Fig 3 shows the filtered menu where the user can either check on/scroll through the restaurant menu minus the items the user is allergic to/restricted from eating or add to cart for ordering.

If the user adds to cart and then proceeds to the next screen, he/she can choose between dine-in/takeout as shown in Fig 4.



Fig 4

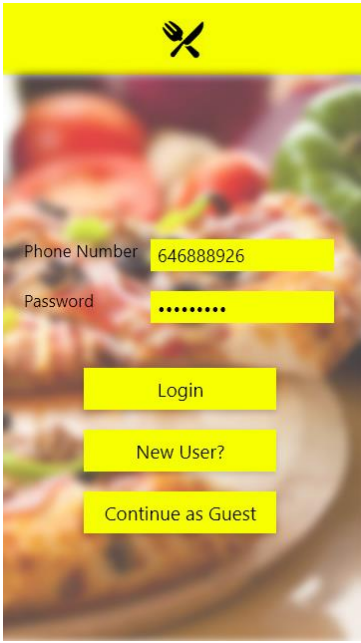


Fig 5

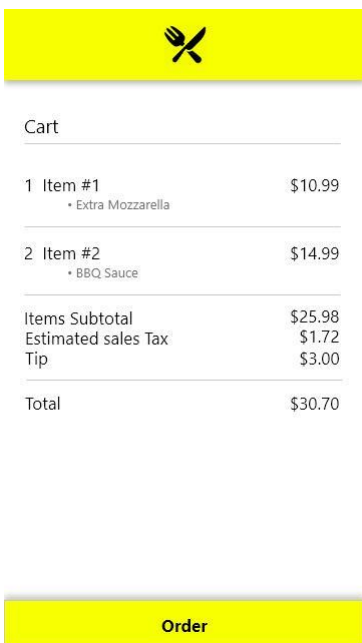


Fig 6

Thereafter the user can either login, create new account or continue as guest on the screen shown in Fig 5 After logging in, the user sees the summary/overview of his/her cart as in Fig 6 after which he/she can place the order which leads to the screen shown in Fig 7.

In this screen (Fig 7), the user can see the progress of the order that has been placed or order more or ask for assistance if he/she might.

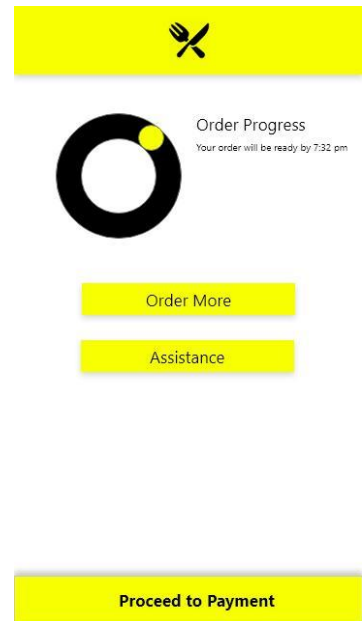


Fig 7

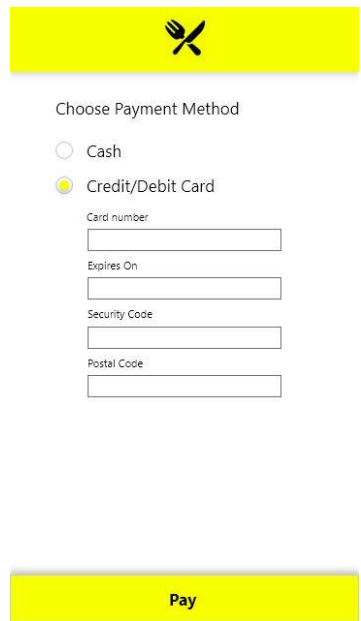


Fig 8

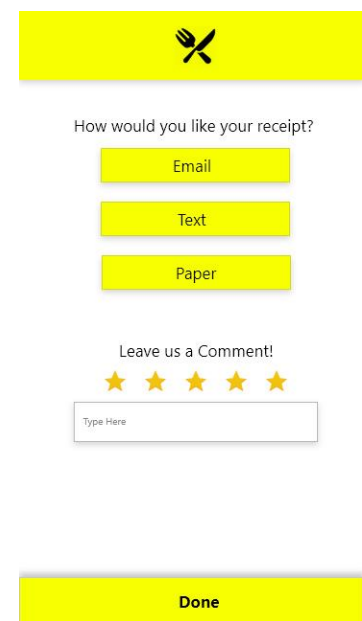


Fig 9

Proceeding to the Payment page as in Fig 8, the user gets to choose between paying via cash or paying via credit/debit card.

Henceforth as shown in Fig 8, the user is asked if he/she would like a receipt for the order either in form of email, text, or cash.

After the screen on Fig 8, the user is done.

## 4.2 User Effort Estimation

### Customer Interface

The customer effort estimation largely depends on the amount of menu filtering done, menu items ordered, and payment options - there are various payment methods and receipt options.

#### Best Case Scenario (Customer Orders a Meal)

1. The customer chooses to log-in as a guest. (1 Click)
2. The customer chooses not to filter by Dietary Restrictions. (1 Click)
3. The menu opens and the customer selects one meal to order. (1 Click)
4. The customer clicks on the "Payment" option. (1 Click)
5. The customer opts to pay with cash. (1 Click)
6. The customer pays with cash.
7. The customer selects "No Receipt" for the receipt option. (1 Click)
8. The customer chooses not to rate their meal and leaves. (1 Click)

**Total** = 7 Clicks

#### Worst Case Scenario (Customer Orders a Meal)

1. The customer opens the system at home and selects a time to make a reservation. (1 Click)
2. The customer selects a table to sit at. (1 Click)
3. The customer arrives at the restaurant and is seated.
4. The customer chooses to sign-up as a Rewards member. (1 Click)
5. The customer types in a username of "N" length and password of "P" length to create an account. (N + P Clicks)
6. The customer selects the enter button. (1 Click)
7. The customer chooses to filter the menu, adding "X" Dietary Restriction filters. (X Clicks)
8. The menu opens and the customer selects "Y" meals to order. (Y Clicks)
9. The customer clicks on the "Payment" option. (1 Click)
10. The customer selects to split the bill. (1 Click)
11. Each customer pays selects their payment method and pays (N Clicks)

12. Each customer selects "E-Mail Receipt" for the receipt option. (N Clicks)
13. Each customer types in an E-mail address of "Z" length. (Z Clicks)
14. The customers choose to rate all of their meals. (Y Clicks)

**Total** = 6 + N + P + X + 2Y + N\*(2 + Z) Clicks

## Employee Interface

The employee effort estimation depends on various different instances - largely, based upon customer input.

### **Best Case Scenario (Chef Serves Customers)**

1. The chef logs in with username of length X. (X Clicks)
2. The chef logs in with password of length Z. (Z Clicks)
3. The chef selects the enter button. (1 Click)
4. The chef selects clock in. (1 Click)
5. The chef selects "Orders". (1 Click)
6. The chef reads the menu items in the preparation queue and clicks "Start Dish". (1 Click)
7. The chef completes cooking the dish and clicks "Dish Complete". (1 Click)
8. The chef selects log-out. (1 Click)
9. The chef selects clock out. (1 Click)

**Total** = 5 + X + Z + 2 \*#dishes Clicks

### **Worst Case Scenario (Chef Serves Customers)**

1. The chef logs in with username of "N" length. (N Clicks)
2. The chef logs in with password of "Z" length. (X Clicks)
3. The chef selects the enter button. (1 Click)
4. The chef selects clock in. (1 Click)
5. The chef selects "Orders". (1 Click)
6. The chef reads the menu items in the preparation queue and clicks "Start Dish". (1 Click)
7. The chef completes cooking the dish and clicks "Dish Complete". (1 Click)
8. Employee takes B breaks throughout the shift. Selects "Start Break" and "End Break". (2B Clicks)
9. The chef selects logout. (1 Click)
10. The chef selects clock out. (1 Click)

**Total** = 7 + N + X + Z + 2\*#dishes + 2B Clicks

### **Best Case Scenario (Employee Serves Customers)**

1. Employee logs in with username of length X. (X Clicks)
2. Employee logs in with password of Z. (Z Clicks)

3. Employee selects enter button. (1 Click)
4. Employee selects clock in. (1 Click)
5. Server sees that all dishes for a table are prepared and clicks "Order delivered". This happens N times through the shift. (N Clicks)
6. Customers pay for the meal using credit/debit card and the payment is processed automatically.
7. Employee selects logout. (1 Click)
8. Employee selects clock out. (1 Click)

**Total** = 4 + X + Z + N (Clicks)

#### **Worst Case Scenario (Employee Serves Customers)**

1. Employee logs in with username of length X. (X Clicks)
2. Employee logs in with password of Z. (Z Clicks)
3. Employee selects enter button. (1 Click)
4. Employee selects clock in. (1 Click)
5. Server sees that all dishes for a table are prepared and clicks "Order delivered". This happens N times through the shift. (N Clicks)
6. Employee is sent push notification by a table and selects "okay". (A Clicks)
7. Customers pay for the meal using cash and server selects "meal paid". ( Y Clicks)
8. Employee takes B breaks throughout the shift. Selects "Start Break" and "End Break". (2B clicks)
9. Employee selects logout. (1 Click)
9. Employee selects clock out. (1 Click)

**Total** = 4 + X + Z + N + A + Y + 2B Clicks

#### **Manager Interface**

The manager effort estimation depends on what the manager wants to view on the tablet.

#### **Best Case Scenario (Manager wants to see who is currently clocked-in)**

1. The manager selects login. (1 Click)
2. Manager logs in with username of length X. (X Clicks)
3. Manager logs in with password of length Z. (Z Clicks)
4. Manager selects enter button. (1 Click)
5. Manager selects clock in. (1 Click)
6. Manager selects the three bars button to see more options. (1 Click)
7. Manager selects Employee Info. (1 Click)
8. Manager selects three bars button to see more options. (1 Click)
9. Manager selects Logout. (1 Click)
10. Manager selects Clock out. (1 Click)

**Total = 8 + X + Z Clicks**

**Worst Case Scenario (Manager wants to see who is currently clocked-in)**

1. The manager selects login. (1 Click)
2. Manager logs in with username of length X. (X Clicks)
3. Manager logs in with password of length Z. (Z Clicks)
4. Manager selects enter button. (1 Click)
5. Manager selects clock in. (1 Click)
6. Manager selects the three bars button to see more options. (1 Click)
7. Manager selects tables on accident. (1 Click)
8. Manager selects the three bars button to see more options. (1 Click)
9. Manager selects Employee Info. (1 Click)
10. Manager selects three bars button to see more options. (1 Click)
11. Manager selects Logout. (1 Click)
12. Manager selects Clock out. (1 Click)

**Total = 10 + 2X + 2Z Clicks**

**Best Case Scenario (Manager wants to view current orders)**

1. The manager selects manager when they are logging in. (1 Click)
2. Manager logs in with username of length X. (X Clicks)
3. Manager logs in with password of length Z. (Z Clicks)
4. Manager selects enter button. (1 Click)
5. Manager selects clock in. (1 Click)
6. Manager selects the three bars button to see more options. (1 Click)
7. Manager selects Orders. (1 Click)
8. Manager sees no tables have been waiting extremely long for an order, Manager selects three bars to see more options. (1 Click)
9. Manager selects Logout. (1 Click)
10. Manager selects Clock out. (1 Click)

**Total = 8 + X + Z**

**Worst Case Scenario (Manager wants to check orders)**

1. The manager selects manager when they are logging in. (1 Click)
2. Manager logs in with username of length X. (X Clicks)
3. Manager logs in with password of length Z. (Z Clicks)
4. Manager selects enter button. (1 Click)
5. Manager selects clock in. (1 Click)
6. Manager selects the three bars button for more options. (1 Click)
7. Manager selects Tables on accident. (1 Click)

8. Manager selects three bars button. (1 Click)
9. Manager selects Order. (1 Click)
10. Manager sees one order has been sitting in the queue for over 30 minutes, manager notifies chef to speed up. (1 Click)
11. Manager selects three bars button. (1 Click)
12. Manager selects Logout. (1 Click)
13. Manager selects Clock out. (1 Click)

**Total** = 11 + X + Z Clicks

## 5. Domain Analysis

### 5.1 Domain Model

#### 5.1.1 Concept Definitions

Responsibility	Type	Concept
<b>R1:</b> Coordinate activity between the customer, chef, waiter, busboy, etc.	D	Communicator
<b>R2:</b> Prompt the customer to select a table	D	Table Status
<b>R3:</b> Display the options for the customer, waiter, chef and manager respectively,	D	Interface
<b>R4:</b> Queue incoming orders for the chefs and bartenders to prepare	D	Order Queue
<b>R5:</b> Store employee login information, along with hours worked	K	Employee Profile
<b>R6:</b> Prevent invalid table selections	D	Table Status
<b>R7:</b> Handle payment processing	D	Payment System
<b>R8:</b> Store customer Reward points based upon previous orders	K	Customer Profile
<b>R9:</b> Display change of table status when a customer reserves a table, selects a table, leaves, and when a busboy cleans a table	D	Table Status
<b>R10:</b> Displays current orders to serve	K	Serving
<b>R11:</b> Store customer login information	K	Customer Profile
<b>R12:</b> Protect reserved tables from being selected before reservation time	D	Table Status
<b>R13:</b> Display favorites, top rated and most ordered under customer/user profile information	D	Customer Profile

<b>R14:</b> Store the customer order in the database	K	Customer Profile
<b>R15:</b> Manage interactions with the database	K	DB Connection
<b>R16:</b> Display filtered menu using user input from the Allergies/Dietary Restriction	D	Controller
<b>R17:</b> Allow manager/server to rearrange the floorplan of the restaurant to accommodate needs	D	Floorplan

### 5.1.2 Association Definitions

Concept Pair	Association Description	Association Name
Customer Profile ↔ DB Connection	Fetch customer's data from the database	QueryDB
Customer Profile ↔ Interface	Display customer's option	Display
Interface ↔ Controller	Allow the user to interact with the app.	User Action
Communicator ↔ DB Connection	Modify or insert data into the database.	UpdateDB
Communicator ↔ Order Queue	Send order and queue to the chef	QueryDB
Controller ↔ Food Status	Allow the user (staff) to update or view the food status.	Update Food Status View Food Status
Controller ↔ Table Status	Allow user to view table status	View Table Status
Controller ↔ Payment System	Allow user to complete the payment	Make Payment
Payment System ↔ DB Connection	Store payment record in the database	Record Payment
Interface ↔ DB Connection	Get the data from the database for the user	QueryDB
Customer Profile ↔ DB Connection	Stores earned rewards/points in the database	Reward System
Table Status ↔ Interface	Displays the current table layout with the status of the tables	Display
Employee Profile ↔ Interface	Displays the employee option	Display
Payment System ↔ Customer Profile	Make updates to point balance for the user.	Update Rewards
Floorplan ↔ Controller	Allow the user to adjust a table in the layout	Floorplan Change

### 5.1.3 Attribute Definitions

Concept	Attribute	Description
Customer Profile	accountUsername	Associated username of the customer (email). Guest account is assigned if no account.
	accountPassword	Password of user account
Interface	confirmOrder	Allows the user to confirm order after choosing food items
	receipt	Allows the user to choose what way they would like to receive the receipt (email, text, paper)
	rateMeal	Allows the user to rate a meal, and then have that rating stored and displayed
	tableStatus	Provides the user with the up-to-date status of each table in the restaurant. Tables can exist in 3 states: available, occupied and dirty
Payment System	paymentMade	Updates system depending on whether the payment has been made.
Food Status	orderStatus	Allows the chef to update the status of the current order being cooked
	orderReady	Allows chef to signal to waiter/waitress and customers that the order is finished
Order Queue	chefQueue	Keeps track of submitted food orders in the order that they were submitted for the chef to follow
	bartenderQueue	Keeps track of submitted drink orders in the order that they were submitted in order for the bartender to follow
Controller	tableList	Shows the customer the current tables available
	tableConfirm	Table is greyed out once it is picked by the customer and stays grey until cleaned
	paymentMade	Once the customer pays, the table is then confirmed and the order is initiated in the kitchen
Communicator	customerMeal	Each meal that the customer orders is stored in the database



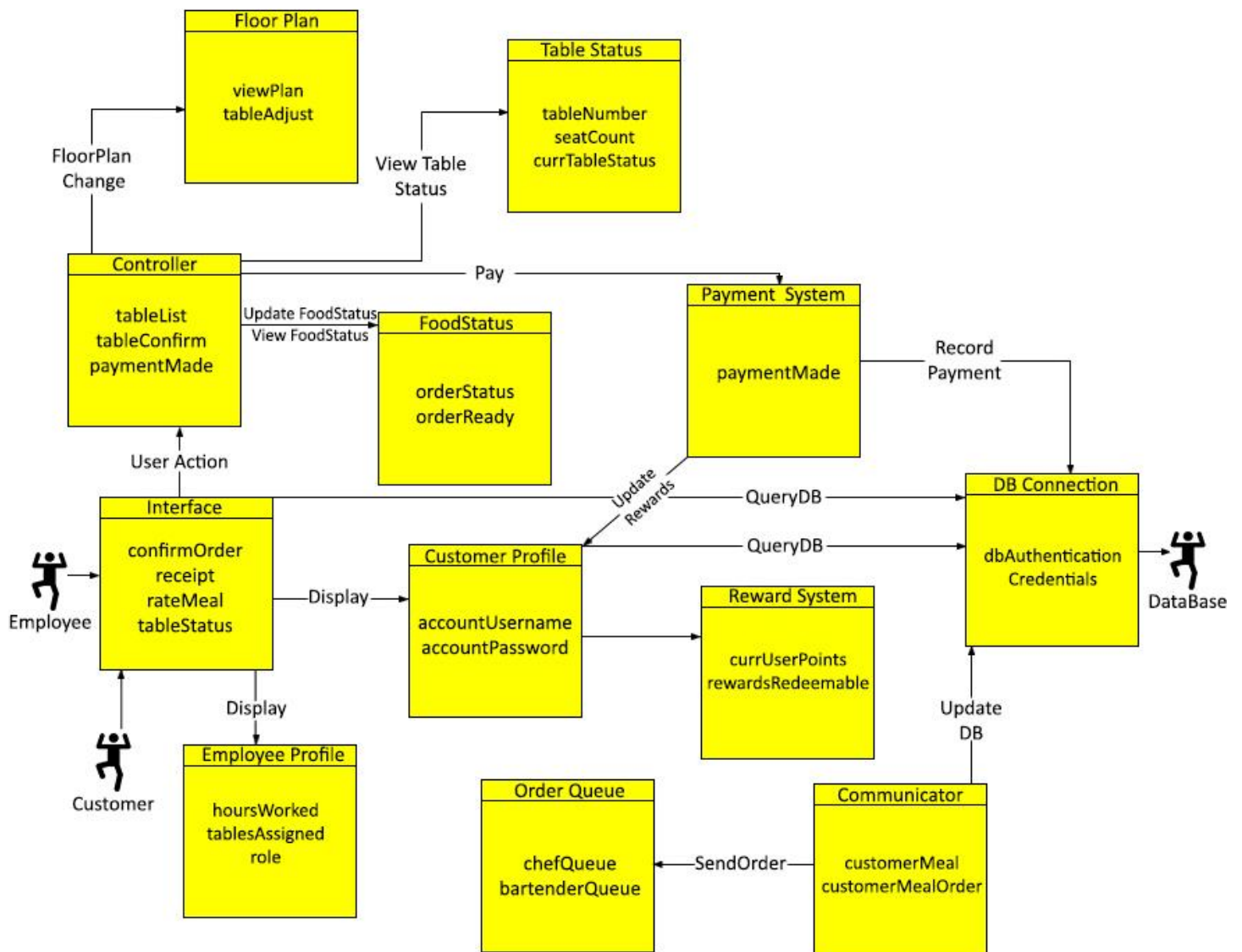
	customerMealOrder	Meals in the queue are ordered and sent to a specific chef, to balance the chef work load
Floorplan	viewPlan	Displays the current layout of the restaurant
	tableAdjust	Allows the manager/server to adjust a table in the restaurant layout
Employee Profile	hoursWorked	Displays hours a particular employee has worked in a particular payment cycle
	tablesAssigned	Shows what tables the employee has been assigned recently (threshold to be determined)
	role	Role of the employee: food server, chef/cook, barista, caterer etc.
Table Status	tableNumber	Displays the table number
	seatCount	Max number of people that can be seated at a particular table
	currTableStatus	Is table empty(white), occupied(grey) or uncleaned(red)?
Reward System	currUserPoints	Displays present user reward points balance
	rewardsRedeemable	Allows user to redeem available rewards
DB Connection	dbAuthenticationCreds	Stores the DB login for authorized personnel

#### 5.1.4 Traceability Matrix

Use Case	PW	Domain Concepts									
		Customer Profile	Interface	Table Status	Payment System	Food Status	Order Queue	Controller	Communicator	Employee Profile	Floor Plan
UC-1	3	X	X	X					X		X
UC-2	3	X	X		X				X		
UC-3	5	X	X						X		
UC-4	3					X	X				
UC-5	4		X								
UC-6	5							X			
UC-7	5									X	

UC-8	3					X					
UC-9	3	X									
UC-10	4			X							X
UC-11	4	X									
UC-12	4	X				X					
UC-13	3	X									
UC-14	4			X							X
UC-15	3			X							X
UC-16	5	X									
UC-17	4	X									
UC-18	3		X								

### 5.1.5 Domain Model Diagram



## 5.2 System Operation Contracts

Operation	Payment
Use Case:	UC-2
Preconditions:	<ul style="list-style-type: none"><li>• The user has loaded the system</li><li>• The user has logged in as a Customer and has the menu opened.</li></ul>
Postconditions	<ul style="list-style-type: none"><li>• The user is prompted on payment method.</li><li>• The user is prompted on receipt option.</li></ul>

Operation:	View Menu
Use Case:	UC-3
Preconditions:	<ul style="list-style-type: none"><li>• The user has logged into their account.</li><li>• a. Manager and waiter/waitress select the app options button in the top left, and then select the restaurant menu option</li><li>• b. Customer logs in with credentials or as guest and immediately sees the menu.</li></ul>
Postconditions	The viewer is presented with the menu interface.

Operation:	Clocking In/Clocking Out
Use Case:	UC-7
Preconditions:	<ul style="list-style-type: none"><li>• The user has loaded the system</li></ul>
Postconditions	<ul style="list-style-type: none"><li>• The user has successfully clocked in/out</li><li>• The database saves the information for future use</li></ul>

Operation:	Earning/Redeeming Rewards
Use Case:	UC-11
Preconditions:	<ul style="list-style-type: none"><li>• The user has loaded the system</li><li>• The user intends to make a purchase</li></ul>
Postconditions:	<ul style="list-style-type: none"><li>• The user has earned an appropriate amount of points for how much was spent</li></ul>

## 6. Project Size Estimation

### Use Case Weights

Use Case	Description	Category	Weight
Reservation (UC-1)	Simple User Interface. Requires navigation through menus to place reservation.	Average	10
Payment (UC-2)	Simple User Interface. Requires light navigation and a payment system in place.	Average	10
View Menu (UC-3)	Simple User Interface. Requires minimal navigation as user selects desired menu items.	Average	10
Meal Prep (UC-4)	Simple User Interface. Requires that statuses be updated by the user.	Average	10
Rate Food (UC-5)	Simple User Interface. Requires small text box and rating scale for user to input thoughts.	Simple	5
Food Filters (UC-6)	Complex User Interface. Requires user input and database references to mark filtered out items.	Complex	15
Clock In/Out (UC-7)	Simple User Interface. Requires small clock in/clock out button to user that shows current status and time clocked in/out.	Simple	5
Serving (UC-8)	Complex User Interface. Requires that the waiter is able to see and update current order statuses as well as receive additional notifications from customers.	Complex	15
Ordering (UC-9)	Simple User Interface. Requires the system to send requested orders to the kitchen.	Simple	5
Table Marker (UC-10)	Simple User Interface. Requires that the system allows user to change table statuses.	Simple	5
Earn Rewards (UC-11)	Simple User Interface. Requires that the system keeps track of the customer's transactions and gives reward credit accordingly.	Average	10
Redeem Rewards (UC-12)	Simple User Interface. Requires that the system allows user to redeem available reward credit for rewards.	Average	10
Take-Out (UC-13)	Simple User Interface. Requires that the system allows the customer to view the menu and submit an order remotely.	Average	10
Table Selection (UC-14)	Simple User Interface. Requires that the system allows the customer to select a current table and see its availability.	Average	10
Floor Plan Adjustment (UC-15)	Complex User Interface. Requires that the system allows the employee to edit the current table layout via GUI interactions.	Complex	15
Login (UC-16)	Simple User Interface. Requires that the system allows users to login and create accounts.	Simple	5

Create Account (UC-17)	Simple User Interface. Requires that the system allows users to create an account with credentials.	Simple	5
Translation (UC-18)	Simple User Interface. Requires that the system can display all menu options in a different language.	Average	10

Simple Cases Weight = 5	Average Cases Weight = 10	Complex Cases Weight = 15	Unadjusted Use Case Points (UUCP)
6	9	3	165

### Technical Complexity Factors (TCFs)

Technical Factor	Description	Weight	Complexity	Calculated Factor (Weight * Complexity)
T1	Distributed, database access system	2	3	6
T2	Users expect good performance but nothing exceptional	1	3	3
T3	System is expected to be efficient to avoid service falling behind	1	4	4
T4	Internal Processing isn't too complex	1	1	1
T5	Light reusability for other features/product customers	1	2	2
T6	Exceptionally Easy Installation	0.5	1	0.5
T7	Ease of Use is very important	0.5	1	0.5
T8	No portability outside of Android	2	0	0
T9	Nothing to really maintain	1	0	0
T10	Parallel Processing is required	1	3	3
T11	Account Security is a concern.	1	3	3
T12	No access for 3rd parties.	1	0	0
T13	Minimal Training required.	1	1	1
<b>Factor Total</b>				24

Technical Factor Total (TF)	TCF Calculation = $0.6 + (TF/100)$
24	0.84

### Calculated Use Case Points (UCP)

UUCP	TCF	Use Case Points (UCP) = UUCP * TCF
165	0.84	138.6

## 7 Plan of Work

We will continue to split up the work done evenly amongst the group for the future submissions of reports. After finishing report 1, we will assign more specific tasks to different individuals when coding our applications features, and will continue looking for things to improve upon.

Use Case	Functional Feature and Description	Start Date	End Date
UC-4 UC-6 UC-7 UC-11	Menu <ul style="list-style-type: none"> <li>Food rating</li> <li>Food filter</li> <li>Placing an Order</li> </ul>	02/26/19	03/25/19
UC-3	Payment	02/26/19	03/25/19
UC-14 UC-15	Earning/Redeeming Rewards	02/26/19	03/25/19
UC-8	Clocking In/Out <ul style="list-style-type: none"> <li>User profile</li> <li>Employee portal</li> </ul>	02/26/19	03/25/19
UC-18 UC-13 UC-17	Table Layout <ul style="list-style-type: none"> <li>Table adjustment</li> <li>Table selection</li> <li>Table Status</li> </ul>	03/26/19	04/22/19
UC-2	Reservation	03/26/19	04/22/19
UC-5	Meal Prep	03/26/19	04/22/19
UC-12	Seating Guests	03/26/19	04/22/19

## 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
Menu	Team A	Brandon, Holly & Roshni
Rewards and Payment	Team B	Dante, Suvranil & Ziad
Clock in/Clock out	Team C	Anthony, Hersh & Michelle

Short Term Plan of Work	Team
Create an interactive customer menu	Team A
Create a menu filtering system	Team A
Create a rating food system	Team A
Create an order from the menu	Team A
Implement a payment system	Team B
Create a rewards system	Team B
Implement an Employee Portal that allows for clocking-in and clocking-out	Team C

## 8 References

- Marsic, Ivan. "Software Engineering." *Professor Zoran Gajic - Home Page*, [www.ece.rutgers.edu/~marsic/Teaching/SE/](http://www.ece.rutgers.edu/~marsic/Teaching/SE/).
- Dimitrovski, Stephan, et al. *Why W8*. 2018, *Why W8*.  
<https://www.ece.rutgers.edu/~marsic/books/SE/projects/Restaurant/2018f-g4-report3.pdf>
- Hanov Solutions Inc., "WebSequenceDiagrams"  
<https://www.websequencediagrams.com/>