



### Report One

#### Project Website:

<https://sites.google.com/scarletmail.rutgers.edu/restaurantautomation2018/home?authuser=1>

#### Project Blog:

<https://littlebitsse2018.wixsite.com/mysite>

#### **GROUP #4**

**Team Name:** Little Bits

**Team Members:**

- Esraa Abdelmotteleb
- Manvi Agarwal
- Johnathan Alcantara
- Lakshmi Sai Nithya Garudadri
- Kevin Honeker
- Zachary Joseph
- Fareen Pourmoussavian
- Dafna Shochat

## Table of Contents

<b>Customer Statement of Requirements .....</b>	<b>4</b>
<b>Customers:.....</b>	<b>4</b>
<b>Employees:.....</b>	<b>6</b>
<b>Managers:.....</b>	<b>6</b>
<b>Glossary of Terms.....</b>	<b>7</b>
Technical Terms:.....	7
Non Technical Terms:.....	8
<b>User Stories.....</b>	<b>10</b>
<b>System Requirements .....</b>	<b>13</b>
<b>Enumerated Functional Requirements.....</b>	<b>13</b>
<b>Enumerated Non Functional Requirements .....</b>	<b>14</b>
<b>On-Screen Appearance .....</b>	<b>15</b>
<b>Functional Requirements Specification .....</b>	<b>19</b>
<b>Stakeholders:.....</b>	<b>19</b>
<b>Actors and Goals: .....</b>	<b>20</b>
Initiating Actors .....	20
Participating actors.....	20
<b>Use Cases .....</b>	<b>21</b>
Casual Description .....	21
Use Case Diagram.....	22
Traceability Diagram .....	23
Fully-Dressed Description .....	25
<b>User Interface Specification .....</b>	<b>37</b>
<b>Preliminary Design .....</b>	<b>37</b>
Customer .....	37

## LITTLE BITS

---

Employees .....	38
Manager .....	39
<b>User Effort Estimation .....</b>	<b>40</b>
UC-1 .....	40
UC-2 .....	40
UC-3 .....	41
UC-4 .....	41
UC-5 .....	41
UC-6 .....	42
<b>Domain Analysis .....</b>	<b>44</b>
<b>Domain Model.....</b>	<b>44</b>
Concept definitions: .....	44
Association Definitions:.....	45
Attribute Definitions .....	46
Traceability Matrix .....	48
Domain Model Diagram .....	50
<b>Systems Operation Contracts .....</b>	<b>51</b>
<b>Mathematical Model.....</b>	<b>53</b>
<b>Plan of Work .....</b>	<b>57</b>
<b>Product Ownership.....</b>	<b>59</b>
<b>References .....</b>	<b>61</b>

## Customer Statement of Requirements

In the modern world, more and more businesses are automating their practices as it has proven to increase efficiency therefore increasing profit. Restaurant industry hasn't caught up yet with many restaurants still operating by taking orders manually. We propose a project to automate many aspects of restaurants using a system interface. This interface will allow easy access to customers, managers, and employees, and will change the way restaurants work. Below are some problems that restaurant owners found that could lower efficiency and lower customer satisfaction.

---

### Customers:

#### **Problem 1: Difficult to find available tables and to efficiently manage seating**

Restaurants have a limited number of tables. When there's high traffic on the restaurant, sometimes customers might end up waiting a long time to be seated or come to find out that no tables will be available for the rest of the day. A solution to this problem is a customer interface within an overall restaurant system interface. Using a mobile device or a restaurant tablet, customers should be able to acquire information about availability and expected waiting time before they can be seated. The customer can then input information about the timing and the number of people in the party. A layout of the restaurant with the tables should be available to the customer based on the information that they provided so they can pick their desired table and time. This allows the customers to make an informed decision about whether to drive to a restaurant. An estimation will be made depending on the number of customers already in the restaurant and the number of reservations already made. The app can then use party size and desired timing to prioritize customers for each table and give an approximate waiting time.

#### **Problem 2: Placing orders and getting a waiter's attention is a long process which can be optimized**

After customers are seated, a waiter has to take their order which poses two problems. The waiter isn't always ready to take the order as soon as customers are ready to place it. Therefore, customers have to wait until the waiter is nearby so they can grab the waiter's attention and place their order. When the traffic is particularly high in the restaurant, customers sometimes have to wait for a significantly long time before they can just place their orders. The other issue is when customers take varying amounts of time to place an order and the waiter has to keep coming back to check whether they're ready. This can cause some confusion between the two parties as well as can lead to a customer being annoyed and having a poor experience. In addition, the waiter ends up wasting their time by having to move between tables asking every table whether they're ready to place their order. Another issue is trying to get assistance from a waiter. Customers have to flag their waiter down and sometimes aren't seen. Using a tablet placed on each table, customers should be able to view the entire menu and place their orders as soon as they're ready. The tablet should contain menu items that are organized into different categories. The customer can then navigate through the different categories and pick their desired menu items and add them to their cart. Items can also be edited or deleted from this interface. After

## LITTLE BITS

---

they are satisfied with their items, the order can then be placed and sent directly to the chef eliminating the need for a waiter to place the order. The tablet should also allow customers to request their waiter and the waiters should be able to see the notification on their tablet. Using this system, waiters can focus more on different tasks and customer satisfaction.

### **Problem 3: Long and tedious payment process which can be optimized**

At restaurants some people take longer to eat than others. This makes it difficult for waiters to figure out when the customers are done eating and are ready to pay their bill. This means that waiters have to keep an eye on whether the customers are still eating and have to choose the right time to go ask the customers if they're done. In addition, a lot of times customers have to split the bill evenly or in proportions creating some hassle for waiters. This is a tedious task and takes a lot of time and patience and can waste time for waiters to do other things. Using the tablet placed on the customers table, they can select an option on the tablet that will show them their bill. The customers then can have the option to split the bill how they want and pay. An option for tipping will also be available to the customer, which waiters can view as well. The customers can pay using various payment methods such as credit cards, PayPal, and cash. When a customer decides to pay in cash, this will send an alert to the waiter to assist the customer.

### **Problem 4: Customers have to wait for their food to come and can get anxious**

Customers sometime have to ask the waiters about when their food is going to be ready. Having to call a waiter over can waste their time and energy which can be directed towards other things. It can also be very frustrating for the customer trying to get the waiter's attention. A solution to this problem would be after the customer places their order, the tablet on their table will give an estimation of how long it'll take for their order to be ready. This estimation will be based on different factors such as the number of chefs, waiters, customers, and the amount of orders in the system.

### **Problem 5: No proper way for customers to leave feedback and review the restaurant and their service.**

For a restaurant to be successful, it's important to get the customers feedback. Customer feedback is important to improve ambience and customer satisfaction. Customers should have the ability to express their thoughts, whether it's a positive or a negative review. Managers should also be able to respond to the customer's feedback and justify the reasons that led to the customers' substandard experience. The manager can use this feedback to improve and can also use the statistics to help manage different parts of the restaurant. We need a rating system that gives the customer an option of rating and adding comments about what they liked and what they disliked. This rating system should also have more advanced features that can show the number of times people have ordered that specific dish and reviews for that dish. Customers should also be able to rate and add reviews for certain employees who served them. This system is different from previous ones in that allows the customers to review their experience at the restaurant.

### Employees:

#### **Problem 1: Difficult to make sure orders are made in the correct order they are placed**

Manually taking the order and notifying the chef of them takes time. Chefs often go wrong in prioritizing different orders which leads to a delay in preparing a customer's order. In many cases, there are many special requests that the waiter forgets to inform the chef about, which can lead to low customer satisfaction. These problems can be resolved by a restaurant interface. In this interface, we can take order prioritization out of the chef's hands and should be able to automatically prioritize the orders on a first-come-first-serve basis. There should be different portals within the employee's subsection where the chef will be able to view the orders in the order they need to be made in. The chef will also be able to review the customer's preferences and requests. With this, the chance of making mistakes will significantly decrease since orders will go directly from customers to the chef without any intermediary person.

#### **Problem 2: Difficulty with communication between waiters and the bussers**

In today's restaurant world, it pays to be efficient. When customers have to wait for a table to be cleaned before being able to be seated it can lead to low ratings and can deter them from coming back. Generally, waiters don't have a proper system in place to communicate with the bussers. Due to this problem, restaurants waste time and possibly lose out on customers. When a customer has to wait, it can deter them from coming back and can lead to poor reviews and a poor overall experience. The waiters are usually in charge of communicating to other employees when a table has been vacated, when a table is being used, and when a table needs to be cleaned. To make this easier they need a system to properly communicate with the bussers. A solution to this issue is that within an interface, in an employee's portal, the waiters should be able to mark tables that need to be cleaned. Then bussers should be able to see this notification and then can also respond and mark when tables are cleaned and ready to go. Waiters should also have the ability to mark when a table is occupied, so there is less confusion between employees. This will save time and is an easy way for waiters and bussers to communicate

### Managers:

#### **Problem 1: Organizing inventory and managing how much to purchase to minimize waste**

In restaurants managing inventory is usually a difficult task. This is a long and tedious task which can be optimized for efficiency and speed. Supplies can also have expiration dates that can lead to heavy consequences when neglected. Restaurants also have problems where they have too little of an item, and have to refuse a customer an order or have to change the recipe. This creates an issue for the chef, as well as for the manager, and can lower customer satisfaction. By having statistics on the popularity of orders and by having the inventory being updated after each order, our managers can have an estimate on what supplies they need more or less of. This system will minimize waste and can lower the risk of having an unsatisfied customer. This system will also be able to track if inventory items are expired need to be replaced or refilled. Our system will have the ability to automatically suggest orders to be made which managers can

## LITTLE BITS

---

then decide whether to make those orders or not. This saves the manager's time and makes inventory management much more efficient and accurate.

### **Problem 2: Scheduling employees can be difficult with multiple preferences and availabilities**

One of the most common issues that restaurants face is scheduling. Making employee schedules can be hard and often time are not as efficient. Managing employee availability can be very tedious and can often lead to conflict, last minute changes to work schedule, and understaffed shifts. By tracking the restaurant's activity, our managers can determine the proper number of employees needed for certain times. With this system, employees clock in using a web clocking system. Our managers can also confirm, deny, or give availability. This system will also allow our managers to monitor any employee's availability to work so the manager can build the work schedule accordingly. This will reduce the possibility of having understaffed shifts, which results in a better overall customer experience. Our employees can also see a detailed page on their schedule for the future and past, hours logged, current pay check, and how much they will receive on a certain day.

### **Problem 3: Managing finances can be optimized and made easier for managers**

Our managers are typically given the responsibility of manually handling the finances of their respective restaurants and must make important financial decisions based on the financial health of their restaurant. This process can be cumbersome and overwhelming for some managers as the task of managing finances for any organization can become a huge problem without the help of software resources. By creating a financial management system where all income and costs incurred by the restaurant are automatically tracked, the manager's job of optimizing the use of restaurant funds will become significantly easier. The system will allow our manager to view profit projections, projected payroll costs, and keep track of restaurant transactions, such as the purchase of inventory items.

---

## Glossary of Terms

### **Technical Terms:**

**Alert** - notify or warn waiter of customer's request of assistance or attention

**Algorithm** - a process or set of rules to be followed in calculations or problem solving operation by a computer

**Automate** - technique of making a process, system, or apparatus operate without the use of human labor

**Database** - a structured set of data held in a computer, especially one that is accessible in various ways

---

## LITTLE BITS

---

**Mobile Application** - type of application software designed to run on a mobile device, such as a smartphone or tablet computer

**NFC** - (near-field communication) set of communication protocols that enable two electronic devices, one of which is usually a portable device such as a smartphone, to establish communication by bringing them within 4 cm of each other

**System** - a coordinated body of methods or a scheme or plan of procedure.

**Tablet** - wireless, portable personal computer with a touchscreen interface, smaller than a notebook computer but larger than a smartphone

### Non Technical Terms:

**Availability** - table's status of being ready to seat new customers

**Bill** - recorded account of amount of money owed for food ordered

**Chef** - person who is responsible for preparing food (or overseeing individuals who are preparing food)

**Chef portal** - A site that can be accessed by chef

**Costs** - the price paid to acquire, produce, accomplish, or maintain anything

**Customer** - any person entering the establishment with intentions of ordering food or drinks

**Employee Portal** - Schedule accessible by employees that lists their shifts and any open shifts they may cover.

**Finances** - the monetary resources of the restaurant.

**Funds** - money immediately available.

**Inventory** - a complete listing of stock on hand, such as raw materials and finished goods.

**Manager** - a person who has control or direction of the restaurant

**Notes** - comments placed by the customer pertaining to the specific preparation or ingredients of a dish

**Order** - the food the customer requests.

**Organization** - the administrative personnel or apparatus of a business

## LITTLE BITS

---

**PayPal** - electronic commerce company that facilitates payments between parties through online funds transfers

**Payroll** - a list of employees to be paid, with the amount due to each

**Prioritize** - designate or treat (something) as more important than other things based on relation to time or urgency of a situation

**Process** - a systematic series of actions directed to some end

**Profit** - the difference between the amount earned and the amount spent in buying, operating, and running a business resulting in a financial gain

**Reservation** - placing a request to hold a table for a specific group or persons by requesting a head of time

**Restaurant** - an establishment where meals are served to customers

**Scheduling** - the process of assigning shifts to employees

**Split the Bill** - the act of breaking up the check into separate checks depending on each customer's individual order with the intention for each customer to pay separately

**Software** - the programs used to direct the operation of a computer, as well as documentation giving instructions on how to use them.

**Statistics** - the numerical facts or data themselves.

**Supplies** - a quantity of something on hand or available for use.

**Table Availability Schedule** - A database that shows the availability of the tables in the restaurant.

**Tip** - leaving gratuity of 15% - 20% of the check as a reflection of adequate service provided by the waiter

**Traditional Restaurant** - refers to how restaurants used to be run before technology took over

**Traffic** - refers to high volume of people or orders arriving in a short time span

**Transactions** - the exchange of money between two consenting persons; i.e. customer and employee.

**Waiter** - a man/woman whose job is to serve customers at their tables in a restaurant

**Waste** - to fail or neglect to use

---

## User Stories

**Customer:**

Identifier	User story	Size
ST-C-1	I can order dishes at my pace without having to wait for a waiter to take my order each time	2
ST-C-2	Based on the size of my party I can reserve table(s) ahead of time and check their availability	4
ST-C-3	I can add/drop ingredients/allergens by my choice while ordering	4
ST-C-4	I can rate the ambience and service according to my satisfaction	2
ST-C-5	I can split the bill among our group and can tip if the service was worth it.	3
ST-C-6	I can see the estimated time for my order to be ready	2

**General Employee:**

IDENTIFIER	USER STORY	SIZE
ST-E-1	I should be able to see when my shifts are for the month	5
ST-E-2	I should be able to give preferences for shifts	4
ST-E-3	I should be able to clock in and out for my shifts	4
ST-E-4	I can create reservations for customers	2

## LITTLE BITS

---

### Waiter/Waitress:

IDENTIFIER	USER STORY	SIZE
ST-W-1	I can see which tables are mine	4
ST-W-2	I can see when a customer needs assistance	5
ST-W-3	I can mark tables are being used	5
ST-W-4	I can mark when customers leave the tables and are dirty	4
ST-W-5	I can see when a customer leaves a tip	2

### Chef

IDENTIFIER	USER STORY	SIZE
ST-CH-1	As a chef, I have a chef portal that I can log into.	2
ST-CH-2	As a chef, I can see (in the chef portal) the orders customers placed in the order I need to make them.	4
ST-CH-3	As a chef, I can see the special requests customers make about each order so I can modify the order to meet the customer's request.	2

### Busser

IDENTIFIER	USER STORY	SIZE
ST-B-1	I should be able to mark when tables are clean	4
ST-B-2	I can see when a table is dirty and when it is vacated	3

**Manager**

IDENTIFIER	USER STORY	SIZE
RT-M-1	I want to see my inventory to see if i am running low on items and see what I need to order	5
RT-M-2	I want to build a schedule to see when my employees will be working	4
RT-M-3	I want to see if items are expired to prevent the kitchen from using those items and use fresher ingredients	3
RT-M-4	I want to be able to change the rate of pay for any individual employee	3
RT-M-5	I want to be able to change the prices of the items on the menu	3
RT-M-6	I want to be able to view the popularity of each item on the restaurant's menu	4
RT-M-7	I want to be able to see my funds to determine my financial situation	5

# System Requirements

## Enumerated Functional Requirements

Identifier	User story	Size
ST-C-1	I can order dishes at my pace without having to wait for a waiter to take my order each time	2
ST-C-2	Based on the size of my party I can reserve table(s) ahead of time and check their availability	4
ST-C-3	I can add/drop ingredients/allergens by my choice while ordering	4
ST-C-4	I can rate the ambience and service according to my satisfaction	2
ST-C-5	I can split the bill among our group and can tip if the service was worth it.	3
ST-C-6	I can see the estimated time for my order to be ready	2
ST-E-1	I can log and log out of my portal	2
ST-E-2	I can view tables that need to be cleaned and mark when they are cleaned	4
ST-E-3	I can see when tables need assistance from a waiter	5
ST-E-4	I can see when an order comes and in what order people placed their orders.	6
ST-E-5	I can input my preferred schedule and can see when I am scheduled	5
ST-M-1	As a manager, I want to see my inventory to see if I am running low on items and see what I need to order	2
ST-M-2	As a manager, I want to build a schedule to see when my employees will be working	3
ST-M-3	As a manager, I want to see if items are expired to prevent the kitchen from using those items and use fresher ingredients	7
ST-M-4	As a manager, I want to be able to change the rate of pay for any individual employee	5
ST-M-5	As a manager, I want to be able to change the prices of the items on the menu	8

## LITTLE BITS

---

ST-M-6	As a manager, I want to be able to view the popularity of each item on the restaurant's menu	6
ST-M-7	As a manager, I want to be able to see my funds to determine my financial situation	1
RT-M-8	As a manager, I want to be able to monitor floor traffic at my restaurant	4

## Enumerated Non Functional Requirements

**Functionality:** This software can be increased in size and used across many different size platform restaurants. This system allows for customers to be able to make reservations and seating's according to their preference. It also allows employees to make reservations for customers if they so choose. We also have created the ability for employees to adjust scheduling based on their availability. We also have allowed for a integrated user experience across the customer, employee, and manager segments. All are intertwined and rely on each other. The employees can see when an order went in and the order in which it was placed. It also maximizes efficiency by allowing bussers to mark when tables are clean and waiters to mark when tables need to be cleaned. However,

**Usability:** We have designed this system for 10 tables, five Waiters, two Chefs, two Bussers, one Manager. This will be able to be used for chain restaurants such as TGI Fridays, On The Border, Applebee's, etc. We will also be providing the users access to a training software, which if they need assistance on how to use our system, they can use the training system.

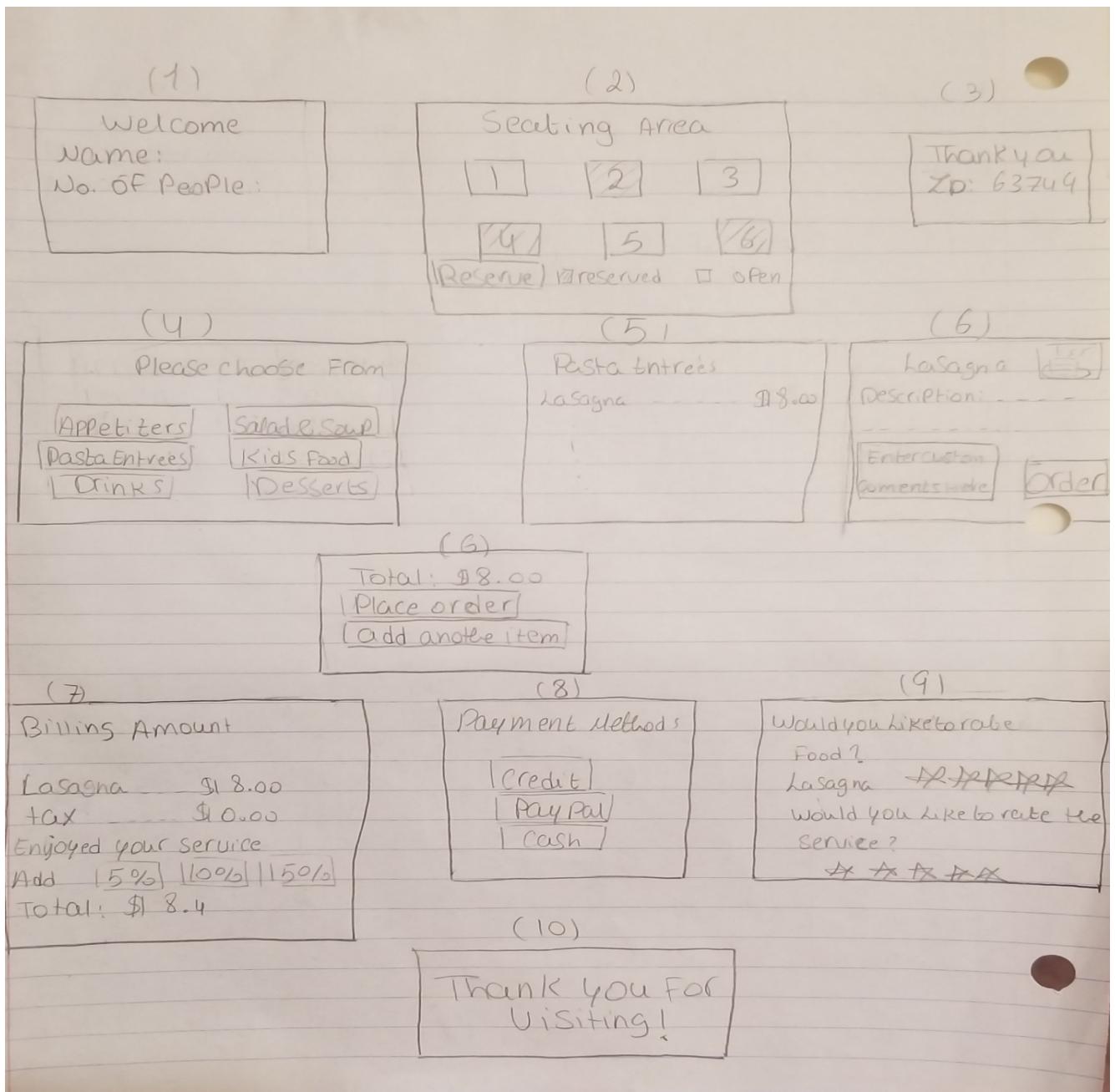
**Reliability:** We created this system so that restaurants can be more efficient and focus on the customer satisfaction. If a system is down, then it can lower customer satisfaction and cause harm to the restaurant. In order to mediate this potential issue, we will be providing 24-hour customer support, dedicated to helping the restaurants with any potential problems. We also provide technicians that can come on site to help manage the software and help manage updates.

**Performance:** This system will run in real time with a possible three seconds delay. The response time for our system will be nearly close to immediate, with the possibility of slight delay.

**Supportability:** This system will be monitored using GitHub. Any necessary changes can be done using GitHub. We can manage updates, problems, and can see who made what changes. Our system will be compatible with any tablets with access to the internet. This application will be used on primarily tablets.

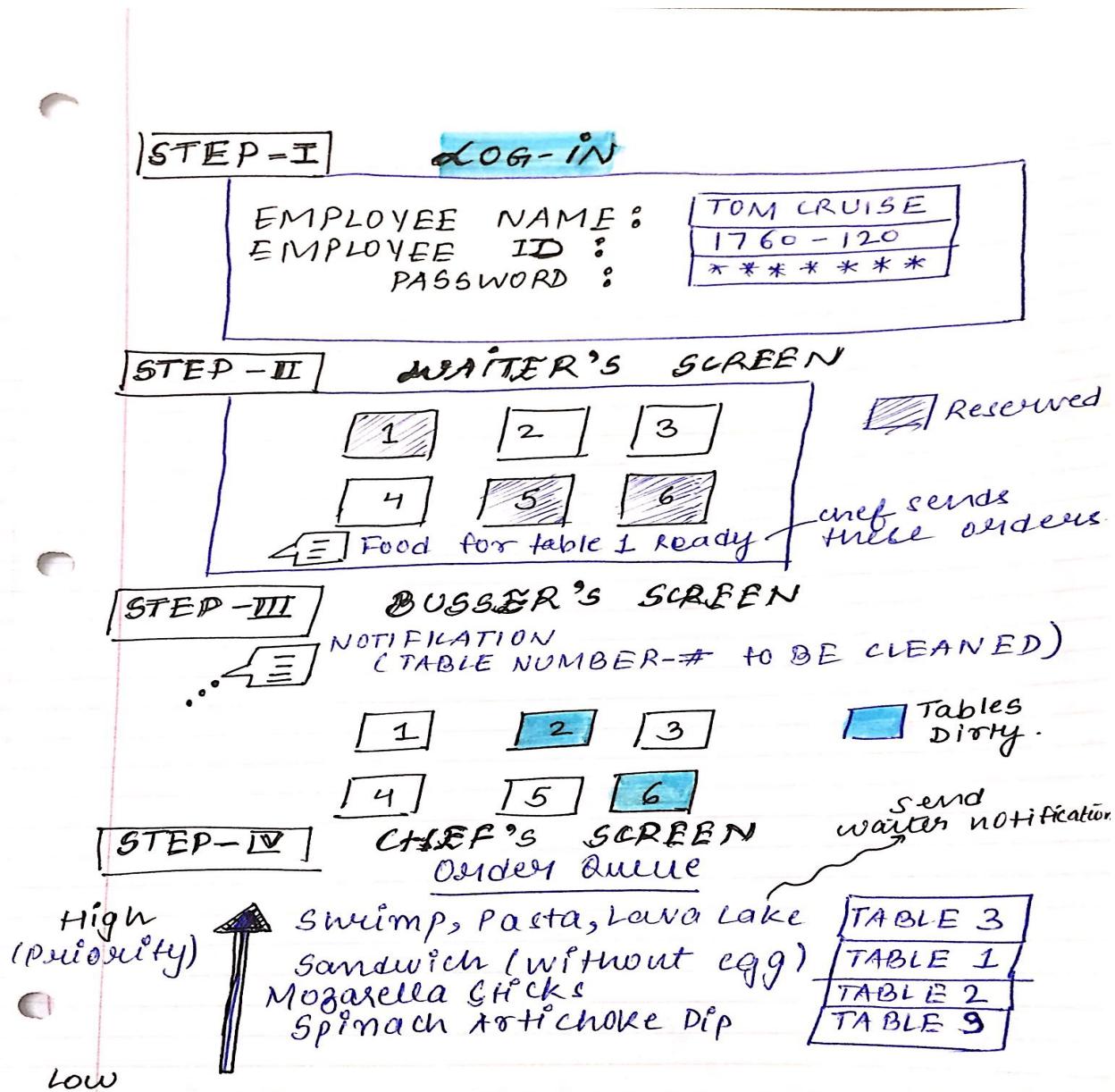
## On-Screen Appearance

**Customer:** The customers will have the ability to see open tables, and reserve tables for their preferred time. This interface also provides the customer the ability to call a waiter for assistance. The customers can also look at the menu and make their selections and pay at the end of their meal. Finally, the customers can pay their bill and leave reviews to help improve the restaurant.



## LITTLE BITS

**Employees:** This page shows how waiters will see tables and be able to mark when a table is dirty. It also shows which tables are reserved. It will also show a user's log in information. Finally, our system incorporates a chef's portal, showing the order in which customers orders come in.



Scanned by CamScanner

STEP-V      SCHEDULE

\* can only view -

Time Spots	Mon	Tue	Wed	Thu	Fri	Sat	Sun
8:00 - 10:00 am	TOM	C	B	A	Z	Y	-
10:00 - 12:00 pm	T	-	W	-	-	-	-
12:00 - 2:00 pm	-	Q	-	P	.	.	S
!	R	.	U.	V	-	.	.
8:00 - 10:00 pm	-	-	-	-	J	M	N

STEP-VI      CLOCK-IN / CLOCK OUT

PUNCH TYPE :-

Clock in	v
Clock out	
Meal	
Clock in	

enter punch

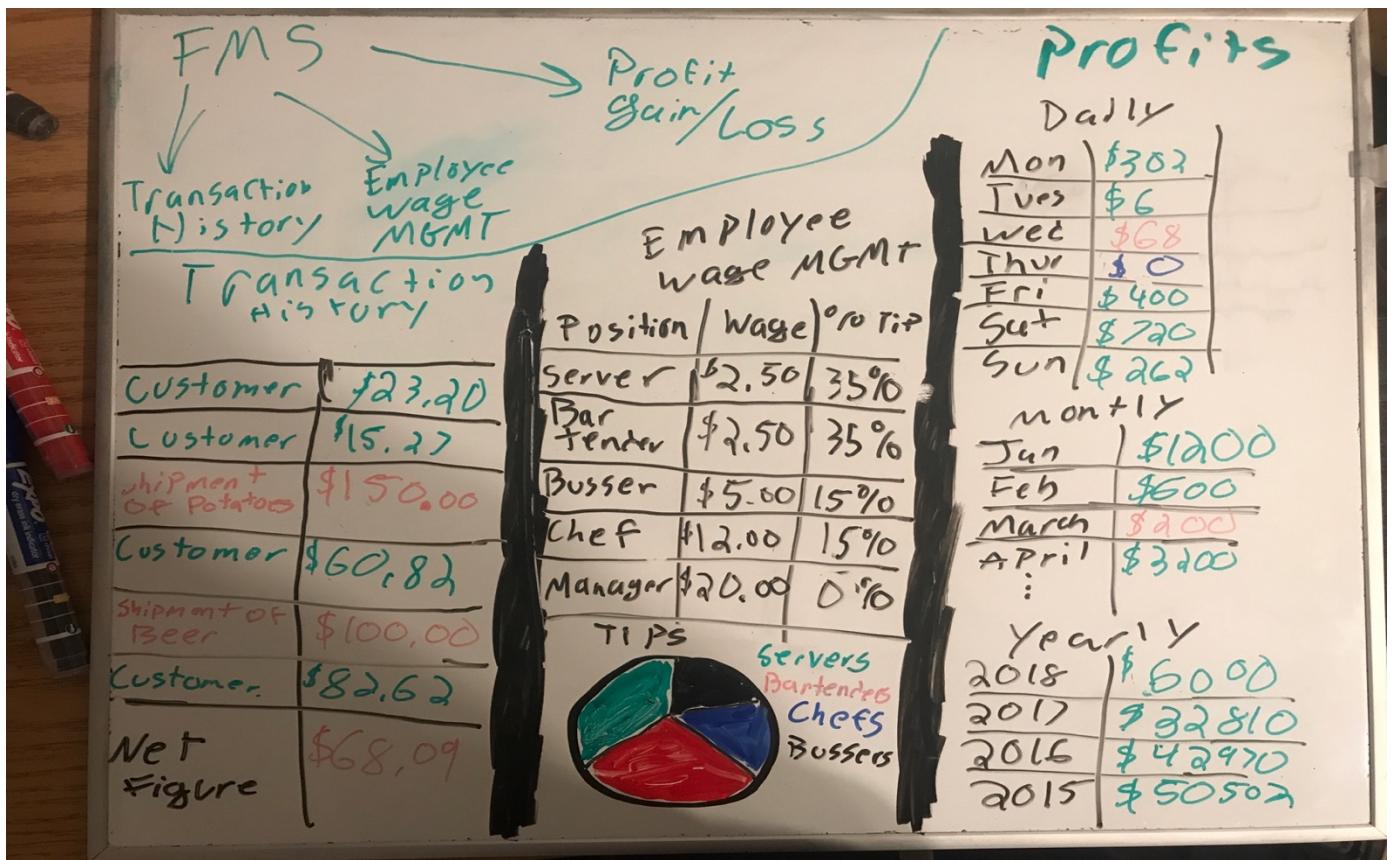
## LITTLE BITS

---

**Manager:** Here is how the scheduling would look once preferences are inputted. You also have an inventory showing what items are there, when they expire, and how much you have. Finally, there is a Financial description, giving various details on different financial statements.

Staff	Mon 18	Tue 19	Wed 20	Thu 21	Fri 22	Sat 23	Sun 24
Employee 1 Position	3:00 AM - 7:00 AM Manager		3:00 PM - 7:00 PM Manager				
Employee 2 Position				11:00 AM - 3:00 PM Supervisor			
Employee 3 Position		11:00 AM - 3:00 PM Supervisor		11:00 AM - 3:00 PM Supervisor			
Employee 4 Position	3:00 AM - 7:00 AM Trainee		7:00 AM - 3:00 PM Trainee				
Employee 5 Position	3:00 AM - 7:00 AM Assistant Manager					11:00 AM - 3:00 PM Assistant Manager	
Employee 6 Position		3:00 AM - 7:00 AM Trainee					
Employee 7 Position	3:00 PM - 7:00 PM Manager	7:00 AM - 3:00 PM Manager		3:00 PM - 7:00 PM Manager		7:00 AM - 3:00 PM Manager	
Employee 8 Position	3:00 AM - 7:00 AM Sales Associate				7:00 AM - 3:00 PM Sales Associate		
Employee 9 Position		3:00 AM - 7:00 AM Cashier	3:00 PM - 7:00 PM Cashier	3:00 PM - 7:00 PM Cashier		11:00 AM - 3:00 PM Cashier	
Employee 10 Position							
Employee 11 Position	3:00 AM - 7:00 AM Manager		3:00 PM - 7:00 PM Manager				
Employee 12 Position							
Employee 13 Position		11:00 AM - 3:00 PM Sales Associate		11:00 AM - 3:00 PM Sales Associate			
Employee 14 Position	3:00 AM - 7:00 AM Trainee		7:00 AM - 3:00 PM Trainee				
Employee 15 Position	3:00 AM - 7:00 AM Assistant Manager					11:00 AM - 3:00 PM Assistant Manager	

Item	Quantity	Expiration date	Date Added
△ Tomatoes	10 lbs	2/10	1/31
Salt	8 lbs	N/A	1/30
△ Cheese	1 lb	2/20	2/5
△△△ Onions	0 lbs	N/A	N/A
△△△ Potatoes	3 lbs	2/7	1/27
△△ Eggs	<1 Dozen	2/10	2/1



## Functional Requirements Specification

### Stakeholders:

There are a multitude of stakeholders that are a major part of our team. The first set of stakeholders that are directly involved are the users. The users include; customers, waiters, bussers, chefs, and managers. Customers do not have access to the portal that employees have, rather they will have access to an interface which allows them to place orders, notify waiters, and see different information regarding their order. They are not direct users, however the customer experience and satisfaction is the highest priority of our team. Employees will have a separate portal that contains their job specific functions. Another set of stakeholders will be the technicians. Technicians interests involve the maintenance of the system as well as updates with the interface. Another group of stakeholders will be the administration, focused on the promotion of our interface. Finally, we have a customer service team. This teams interest lies in helping the restaurants using our system with basic technical problems that can be fixed over the phone.

### Actors and Goals:

#### Initiating Actors

##### Bussers:

Role - The employee who is in charge of the cleanliness of the restaurant.

Goal - Manage the status of tables, and clean tables that are marked dirty on the app.

##### Chef:

Role - The employee who is in charge of the kitchen and preparing meals.

Goal - Manage the ingredient inventory, manage the menu, and prepare meals on the order queue.

##### Customer:

Role - The restaurant visitor who orders food and drink from the restaurant for dining-in or taking-out.

Goal - Place, pay for, and receive orders quickly, and choose whether to eat-in or take-out.

##### Manager:

Role - The employee who manages the entire restaurant.

Goal - Manage employees and scheduling, keep track of inventory, and keep track of profits/losses.

### Participating actors

#### Waiter:

Role – The restaurant employee that handles customer interactions such as adjusting bill, getting table set, and bringing food

Goal – To bring out food in a timely manner once it is done, to make sure tables are empty and ready for use, and to handle any customer needs

#### MySQL Database:

Role - The object that stores information needed for the system.

Goal - Store and modify information pertaining to inventory, employees, profits, losses etc..

## Use Cases

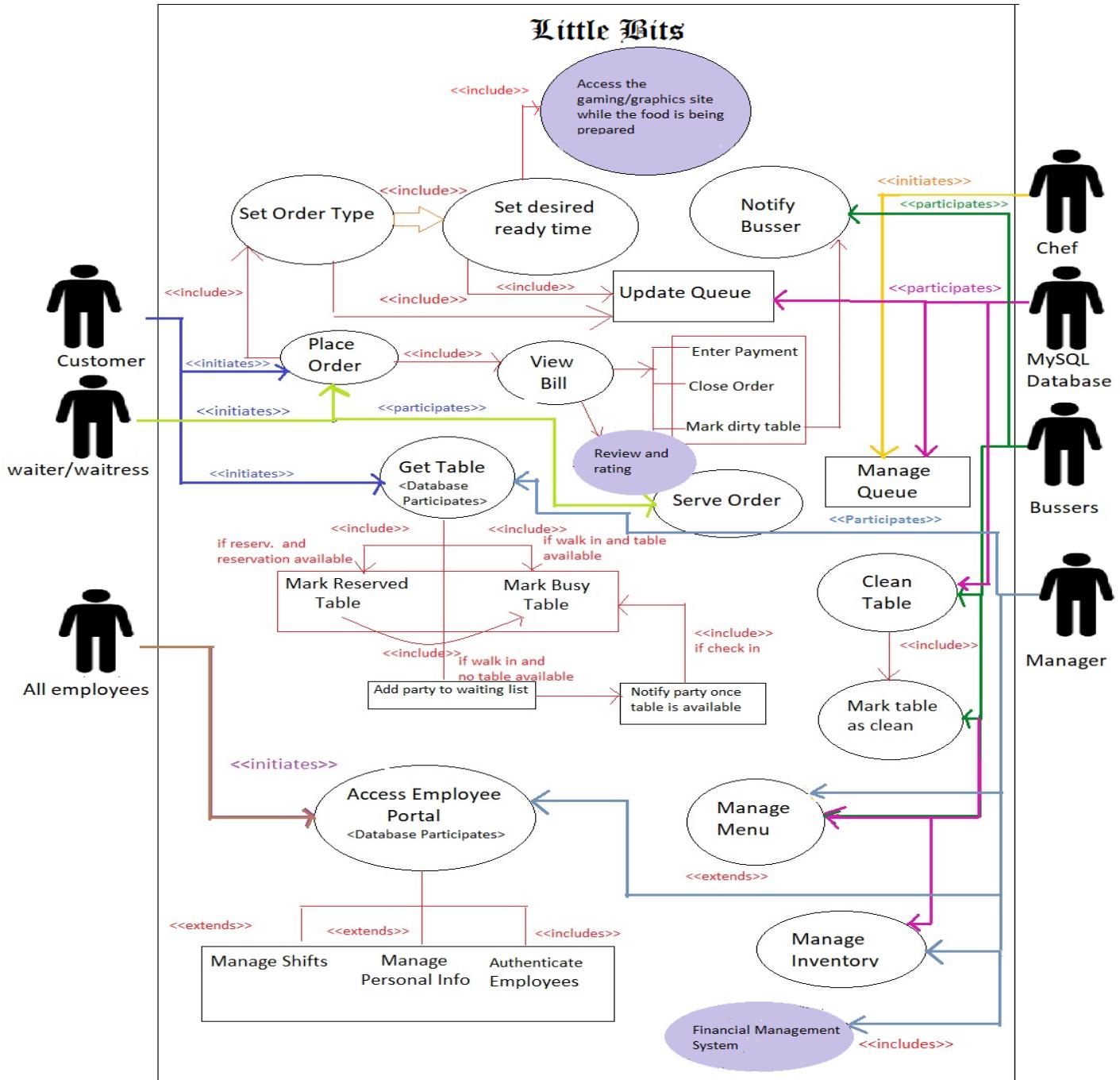
### Casual Description

Our team's user stories will serve as the casual descriptions for the use cases. Included is a weighting table to show level of effort and difficulty.

User Case	Use Case Weight
<u>UC-1</u>	<u>6</u>
<u>UC-2</u>	<u>8</u>
<u>UC-3</u>	<u>8</u>
<u>UC-4</u>	<u>6</u>
<u>UC-5</u>	<u>7</u>
<u>UC-6</u>	<u>8</u>

# LITTLE BITS

## Use Case Diagram



## Traceability Diagram

User stories	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6
ST-C-1						
ST-C-2						
ST-C-3						
ST-C-4						
ST-C-5						
ST-C-6						
ST-E-1						
ST-E-2						
ST-E-3						
ST-E-4						
ST-W-1						
ST-W-2						
ST-W-3						
ST-W-4						
ST-W-5						
ST-CH-1						
ST-CH-2						
ST-CH-3						
ST-B-1						
ST-B-2						
RT-M-1						
RT-M-2						

## LITTLE BITS

---

RT-M-3						
RT-M-4						
RT-M-5						
RT-M-6						
RT-M-7						
RT-M-8						

## Fully-Dressed Description

Use Case UC-1: Managing Orders

**Related Stories:** ST-C-1, ST-C-6, ST-CH-2

**Initiating Actor:** MySQL Database

**Actor's Goal:** Orders placed and sent to the chef's tablet and the database are then queued and an algorithm then calculates estimated time for the order to be ready and displays it on to the customer's tablet.

**Participating Actors:** Customer, Chef

**Preconditions:**

- The restaurant menu is visually displayed on the customer's tablet
- Incoming orders will be displayed visually on the chef's tablet
- An algorithm calculates estimated time-based on queued orders

**Postconditions:**

- The expected preparation time will be visually displayed on the customer's tablet

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

→ An order is placed by a customer and is sent to the database

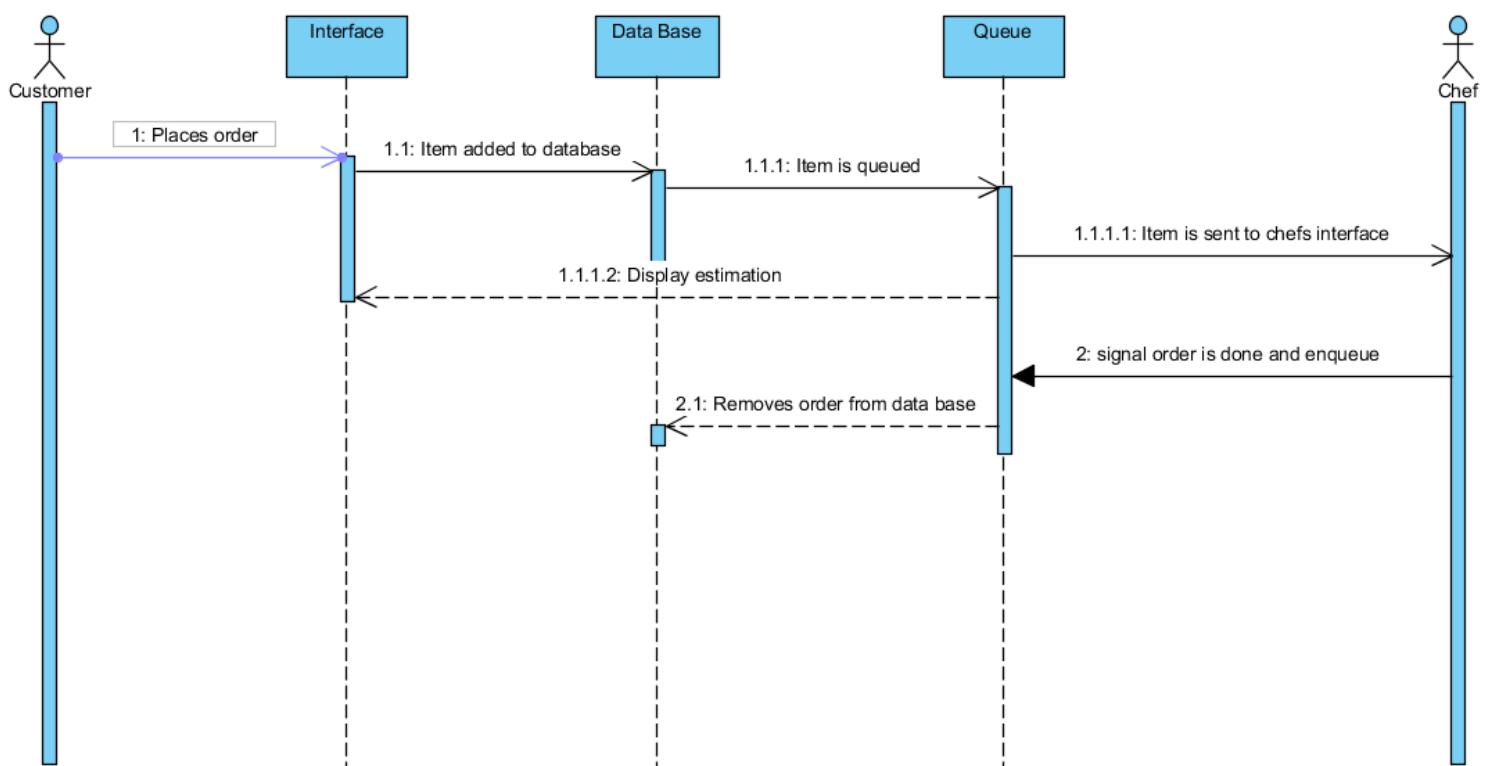
← The database registers an order has been made and places the order in the order queue

→ The database displays the next order in the queue to the chef tablet for preparation

← An algorithm calculates an estimated time for preparation based on the orders queued and displays the calculated time to the customer's display

← Chef signals that the order is done and is removed from the queue

→ Order gets removed from database



### Use Case UC-2: Traffic Monitoring

**Related Stories:** ST-C-2, ST-E-4, ST-W-1, ST-W-3, ST-W-4, ST-B-1

**Initiating Actor:** Customer

**Actor's Goal:** Have the ability to place reservations and place orders

**Participating Actors:** Waiter, Chef, Busser, Manager

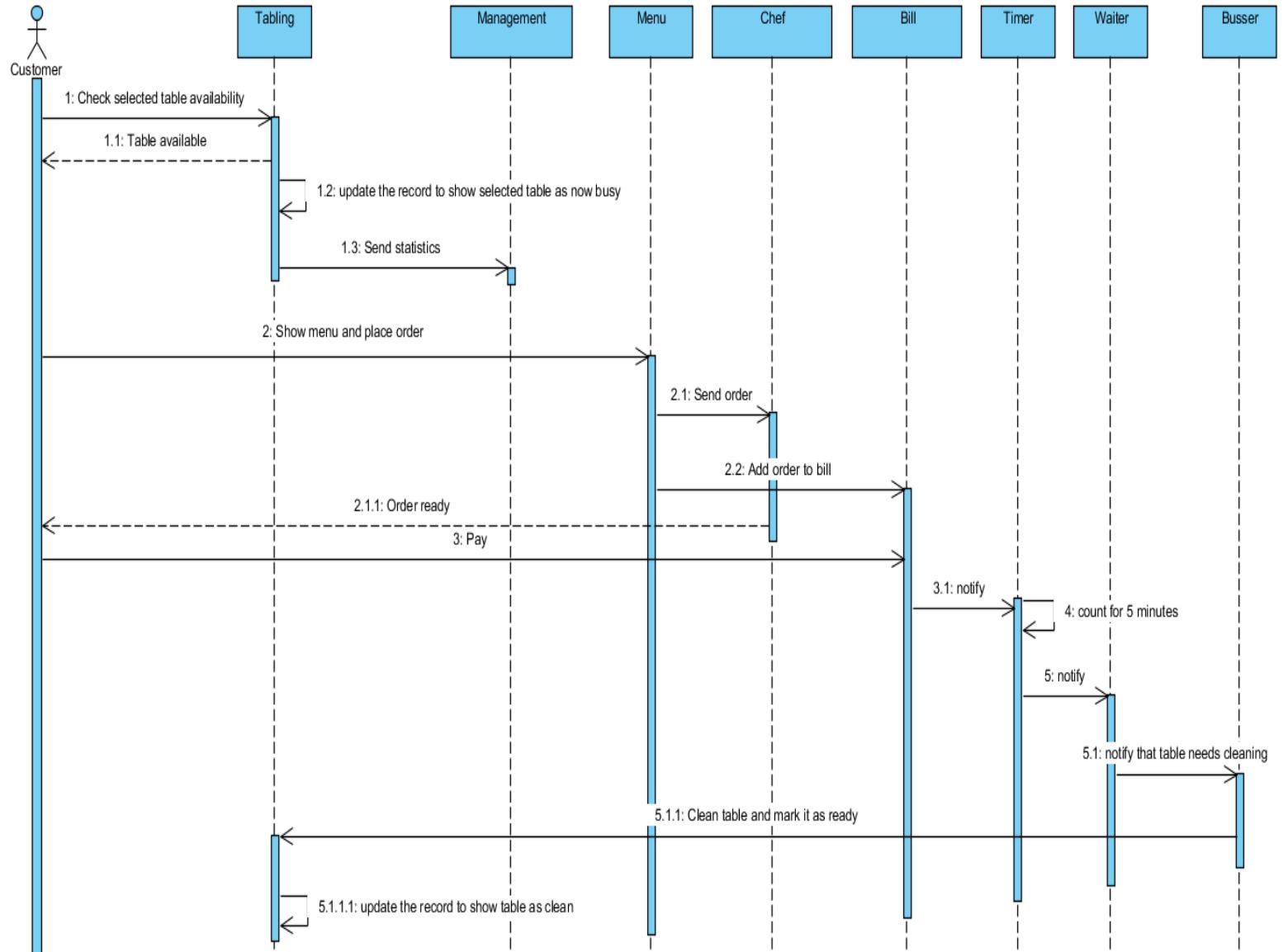
**Preconditions:** The user creates a reservation, come to the restaurant, and places an order

**Postconditions:** The waiter can see when a table is done and bussers can mark when tables are cleaned.

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

- The customer goes online and selects a table and creates a reservation
- ← The system alerts the restaurant that this table needs to be reserved
- ← The system updates itself to show the table as now reserved.
- ← The system adjusts the algorithm that monitors traffic and gives statistics to managers.
- The customer shows up and sees menu and places order
- The chef clicks that the order is finished
- ← The system notifies the customer that the order is done, and the waiter is notified to bring the order to the table
- The customers finish the meal and pay their bill and then proceeds to leave the restaurant
- ← The timer counts for 5 minutes
- ← The timer notifies the waiter to check if the table is vacant
- The waiter marks that the table is vacated and dirty
- ← The busser is notified that a table is dirty and needs to be cleaned
- The busser cleans the table, and marks that the table is ready to be used
- ← The system is updated with an open table that can be viewed in the table availability

## LITTLE BITS



Use Case UC-3: Customer Ordering

**Related Stories:** ST-C-1, ST-C-3, ST-C-5, ST-C-6, ST-CH-2

**Initiating Actor:** Customer

**Actor's Goal:** The ability to view the menu, ordered desired food and pay the bill

**Participating Actors:** Waiter, Chef and Manager

**Preconditions:** A tablet open on the customer interface

**Postconditions:**

- The customer can make the order
- The customer can pay the bill after consumption.

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

→ The customer shows up and sees menu and places order

→ Customer picks the option to see the menu in the application.

← The system shows all the food and drink items in their respective categories, as well as the current order's bill on the right side.

→ Customer chooses which food category they would like to add to the order.

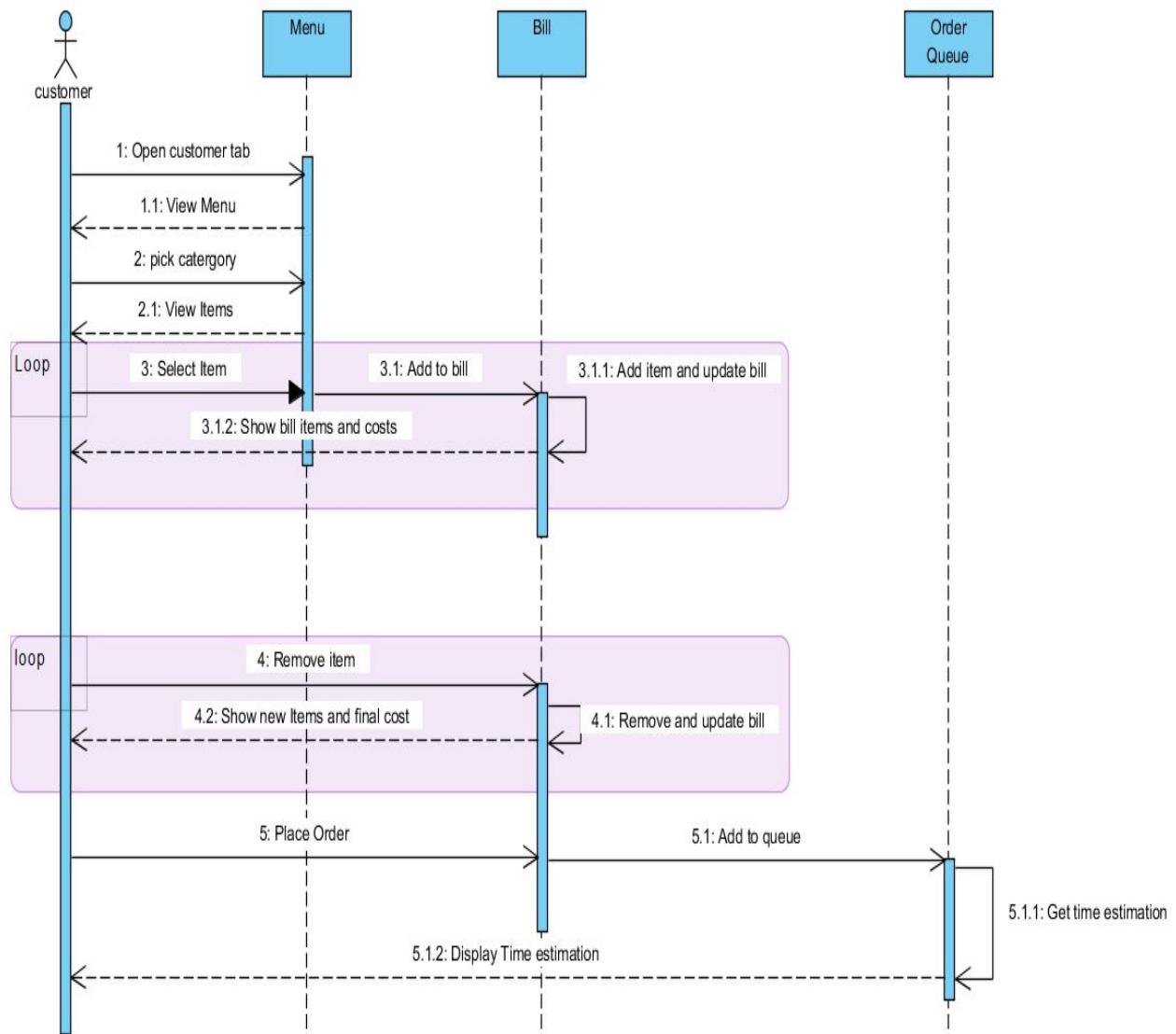
← The system displays all the items associated with the picked category and an option to add to order.

→ Customer selects the option to add the desired item to the order.

← The bill is updated and the customer has the option to place order or add new item

← The bill is updated and the order is sent to the chef's tablet in the order it was placed in the restaurant

← An estimation of the time to be ready is given to the customer



### Use Case UC-4: Payment Process

**Related Stories:** ST-C-5, ST-W-5

**Initiating Actor:** Customer

**Actor's Goal:** To pay their bill successfully

**Participating Actors:** Waiter and Customer

**Preconditions:** The Customer has clicked and opened the Pay Your Bill tab and has chosen one of the methods of payment.

**Postconditions:** The customer pays the bill with or without the Waiter's assistance depending on which option the customer chooses.

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

→ The customer chooses the mode of payment of his bill (i.e Credit Card, PayPal, Cash).

← The system opens up the tab according to the decision made by the customer.

→ The customer can either choose to split the bill or to pay all of it alone.

→ Based on the bill amount, the system notifies the waiter tipping amount to the customer.

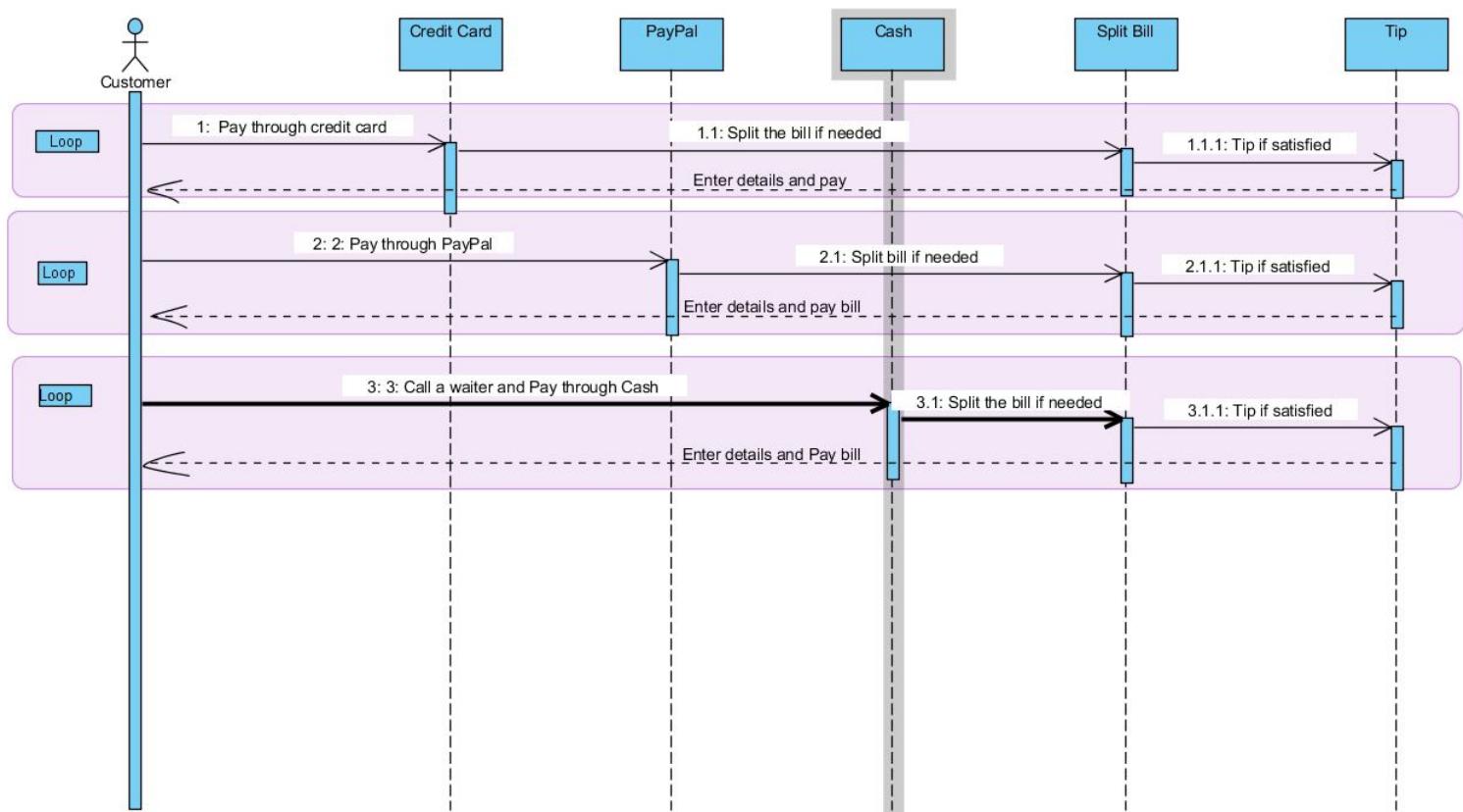
→ The customer makes a decision whether to tip or not.

← If the customer chooses to tip, the billing amount plus the tipping amount gets credited to the restaurant. If not, the restaurant receives the bill.

← The system opens the review screen after notifying the customer that his payment was made.

## LITTLE BITS

---



Use Case UC-5: Managing Employees

**Related Stories:** ST-E-1, ST-CH-1, ST-M-2, ST-M-4

**Initiating Actor:** General Employee

**Actor's Goal:** Have the ability to see their portal for their specific page which gives them their interface.

**Participating Actors:** General Employees

**Preconditions:** The user logs in which automatically clocks them in.

**Postconditions:**

- The user saves his preferences
- The user is automatically clocked out.

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

→ The employee logs in to the system

← The system automatically clocks in the employee and opens the portal depending on their position

→ The employee then can see his or her schedule and can mark preferences for time availability

← The system takes into account all the time preferences and creates a list of times for the manager to see

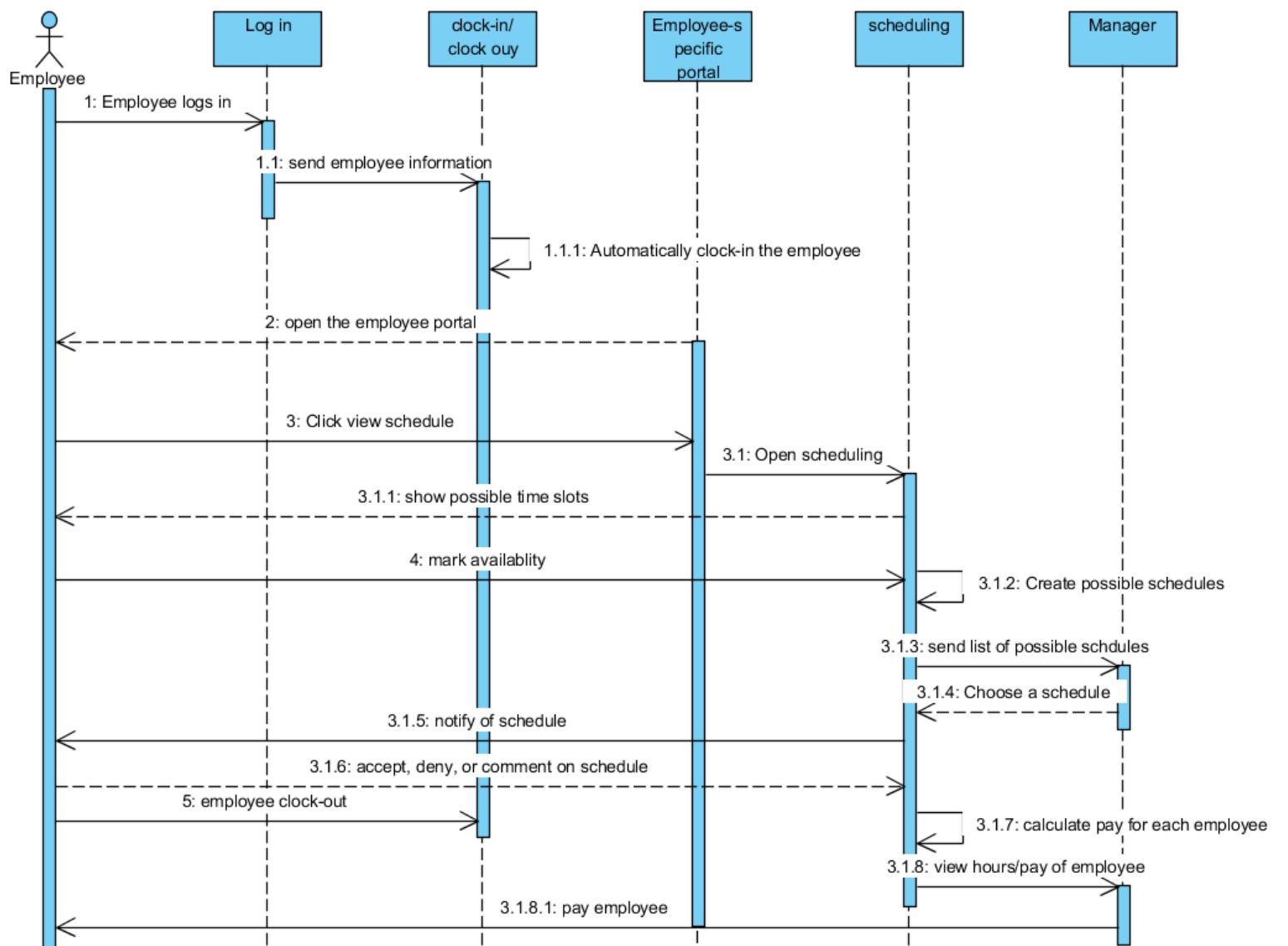
→ The manager then makes the schedule and posts it for employees to see

← Employees can then accept, deny, or comment if they can work or if the schedule is correct

→ The employee clocks out

← The manager can see the hours and pay for the employees, and has the ability to adjust the pay

← The system calculates the pay stubs for employees to see based on logged hours



Use Case UC-6: Managing Restaurant

**Related Stories:** ST-M-1, ST-M-3, ST-M-5, ST-M-6, ST-M-7, ST-M-8

**Initiating Actor:** Manager

**Actor's Goal:** To have control over the restaurant and its employees. Managers need to know the inventory and the financial situation of the restaurant to and adjust menu items and pricing as well as employee pay accordingly. Managers must also be aware of customer traffic.

**Participating Actors:** Database

**Preconditions:** Manager opens portal and sees all the available options

**Postconditions:**

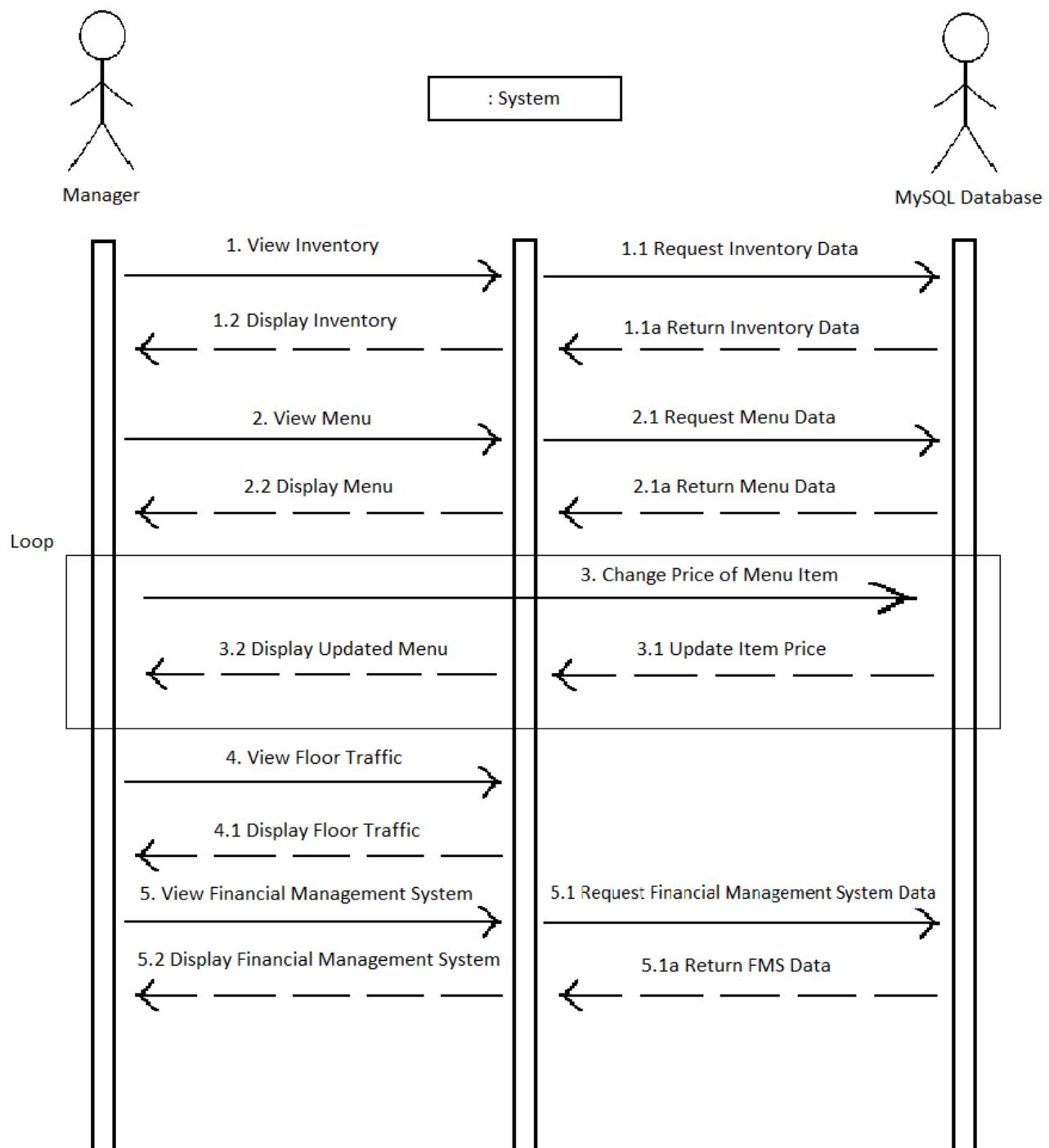
- The manager looks at the systems recommendations for inventory
- The manager selects their preferences and can make adjustments

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

- 1) Manager views inventory database to see that the restaurant is running low on a certain ingredient
- ← 2) The database indicates that the ingredient is still fresh, but the manager determines that the supply will not last the night
- 3) Manager increases the price of items on the menu that utilize said ingredient
- 4) Manager views popularity of menu items on the database that use that ingredient as well as floor activity to determine how quickly the supply will deplete
- 5) Manager marks appropriate items as unavailable once the restaurant runs out of that ingredient
- 6) Manager views and edits financial information that pertains to the restaurant

## LITTLE BITS

---



# User Interface Specification

## Preliminary Design

### Customer

### Reservation

Name: _____	Seating Area	Thank you for reserving a table!									
No. of People: _____	<table border="1"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>Occupied</td><td>Available</td><td>Reserve</td></tr> </table>	1	2	3	4	5	6	Occupied	Available	Reserve	
1	2	3									
4	5	6									
Occupied	Available	Reserve									
<input type="button" value="Submit"/>											

### Ordering

Appetizers	Soups & Salads	Pasta Entrée		Shrimp Scampi
Pasta Entrees	Kids Food	Shrimp Scampi.....\$9.00 Lasagna.....\$8.00 Mac and Cheese.....\$9.00 Gnocchi Skillet.....\$11.00 Noddle Bowl.....\$11.00 Chicken and Linguini.....\$9.00		
Drinks	Desserts			Description: Shrimp sautéed in a garlic sauce, tossed with green onion, parsley, and angel hair pasta.
		<input type="text" value="Enter custom comments here"/>		<input type="button" value="Order"/>

### Payment

Billing Amount	Credit Card	Credit Card	PayPal	Cash
Pasta.....\$11.00 Cake.....\$4.00 Total.....\$15.00	<input type="button" value="Credit Card"/>	Please swipe your card to the right!	<input type="button" value="PayPal"/>	Waiter on the way!
	<input type="button" value="PayPal"/>		<input type="text" value="Please enter your PayPal email below"/>	
	<input type="button" value="Cash"/>		<input type="text" value="Enter email here"/>	

### Review & Comment

How would you rate your food?  How would you rate the service?  Comments? <input type="text"/>	Thank you for visiting!
---	-------------------------

## Employees

# Login

Login

Employee ID: \_\_\_\_\_

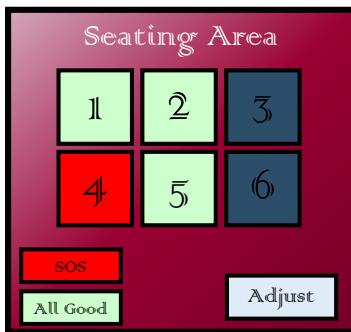
Password: \_\_\_\_\_

Submit

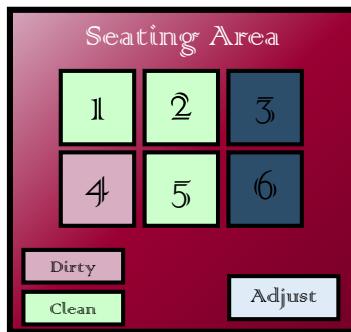
Schedule      Clock In

Order Que/  
Seating Area      Clock Out

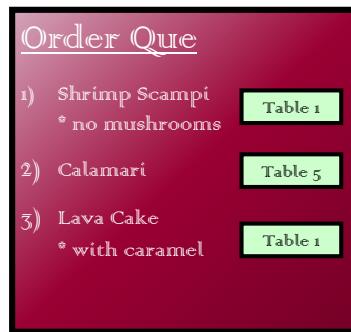
## Waiter's Screen



## Busser's Screen



## Chef's Screen



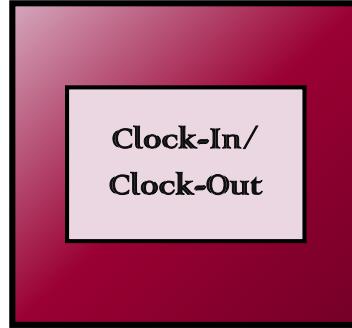
## Schedule

Schedule

Class Schedule for November 2012 - February 2013

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
4:00-7:00 AM	FIT Builder / Cycling	CHE Athlete / Cycling	FIT Builder / Cycling	CHE Athlete / Run	Power Board / Cycling	Indoor / Outdor Rides
7:00-8:00 AM		Drop-In / Open Fitness Class * 8:30		Drop-In / Open Fitness Class * 7:30		Indoor / Long Board
8:00-9:00 AM						Indoor / Long Board
9:00-10:00 AM						Indoor / Long Board
11:00-12:00		Plates / Yoga		Plates / Yoga		
12:00-1:00 PM						
1:00-2:00 PM	Cycling Class	Gone & Flexibility	Cycling Class	Gone & Flexibility		

## Clock In/Out



**Manager**

# Login

Login

Employee ID: \_\_\_\_\_

Password: \_\_\_\_\_

**Submit**

Inventory	Restaurant Traffic
Finances	Menu and Ratings
Scheduling	

**Inventory Screen**

**Finances Screen**

**Scheduling Screen**

Add Items	Update Items
Remove Items	View Items

Transaction History	Profits
Employee Wages	

**Restaurant Traffic Screen**

**Menu and Ratings Screen**

Name	price	rating
Shrimp Scampi	9.00	4.3
Lasagna	8.00	4.3
Mac and Cheese	9.00	3.7
Gnocci Skillet	11.00	
Noodle Bowl	11.00	
Chicken and Linguini	9.00	3.9

**Add Item**

**View Items**

**Profits**

**Transaction History**

**Employee Wages**

Item: \_\_\_\_\_

Quantity: \_\_\_\_\_

Expiration: \_\_\_\_\_

**ADD**

Item	Quantity	Expiration date	Date Added
Tomatoes	0.5kg	N/A	1/20
Salt	0.5kg	N/A	1/20
Cheese	1kg	2/20	2/20
Olive Oil	0.5kg	N/A	N/A
Potatoes	2kg	2/27	2/27
Eggs	12 dozen	2/20	2/20

Daily	Monthly	Yearly
January.....\$545		
February.....\$690		
March.....\$560		

Shipment	Customer
Table 5.....\$75.97	
Table 3.....\$17.58	
Table 6.....\$105.69	

Server.....\$2.50
Bartender.....\$2.50
Busser.....\$5.00
Chef.....\$12.00
Manager.....\$20.00

## User Effort Estimation

### **UC-1**

UC-1 effort estimation is covered by other use case effort estimations.

### **UC-2**

Scenario 1: User reserves a table:

Navigation:

1. Customer clicks the “Reserve” button.
2. Customer enters the number of people in their party.
3. Customer selects the desired table.
4. Customer clicks “ok”

Scenario 2: Chef Prepares Order:

Navigation:

1. Select the option to go to the queue.
2. Chef selects an order from the queue, the items within the order (including attached notes or ingredients alterations) are displayed
3. Chef selects “in progress” once the order preparation is about to begin
4. Chef selects “complete” once the order is ready for pick-up

Scenario 3: User pays the bill:

Navigation:

1. Click the “Pay” button
2. Click “Split the bill” if user desires to
3. Enter the number of people among whom the bill will be split
4. Enter the payment amount or percentage for payer
5. Click “Pay”

Scenario 4: Busboy Cleans Table:

Navigation:

1. Select the option to go to the table layout.
2. Busboy selects a table that is “ready for cleaning” and click on it once to change its status to “cleaning in progress”
3. Busboy clicks on it once more to change its status to “ready for seating” upon completion of cleaning

## LITTLE BITS

---

### UC-3

Scenario 1: Ordering food

Navigation:

1. Customer selects desired food category
2. Customer selects desired food item
3. Customer then writes needed comments and places the order
4. Customer then chooses their desired payment process
5. Customer enters payment details or calls waiter for cash payments

### UC-4

Scenario 1: Payment of bill through Credit Card

Navigation:

1. Customer selects the payment option as ‘Credit Card’
2. Customer has a choice to ‘split the bill’ if needed.
3. Customer can either choose to ‘tip the waiter’ or not.

Scenario 2: Payment of bill through PayPal

Navigation:

1. Customer selects the payment option as ‘PayPal’
2. Customer has a choice to ‘split the bill’ if needed.
3. Customer can either choose to ‘tip the waiter’ or not.

Scenario 3: Payment of bill through Cash

Navigation:

1. Customer chooses to pay the bill by ‘Cash’.
2. The waiter arrives to collect the bill.
3. Customer has a choice to ‘split the bill’ if needed.
4. Customer can either choose to ‘tip the waiter’ or not.

### UC-5

Scenario 1: Submitting employee availability

Navigation:

1. Employee clicks “Login”
2. Employee enters username and password
3. Employee clicks “Login”
4. Employee clicks “View Schedule”
5. Employee clicks on the time slots they are available
6. Employee clicks “Submit Availability”

## LITTLE BITS

---

### UC-6

Scenario 1: Viewing Inventory

Navigation:

1. Manager selects “Login”
2. Manager enters username and password
3. Manager selects “Login”
4. Manager selects “Inventory”
5. Manager selects “View Items”

Scenario 1.2: Add Item to Inventory

Navigation:

1. Manager selects “Login”
2. Manager enters username and password
3. Manager selects “Login”
4. Manager selects “Inventory”
5. Manager selects “Add Items”
6. Manager inputs data corresponding to that item and hits Add

Scenario 1.3: Delete item from Inventory

Navigation:

1. Manager selects “Login”
2. Manager enters username and password
3. Manager selects “Login”
4. Manager selects “Inventory”
5. Manager selects “Remove Items”
6. Manager selects item that is going to be removed

Scenario 2: View Profits

Navigation:

1. Manager clicks “Login”
2. Manager enters username and password
3. Manager clicks “Login”
4. Manager selects “Finances”
5. Manager selects “Profits”

## LITTLE BITS

---

### Scenario 3: View Transaction History

Navigation:

1. Manager clicks “Login”
2. Manager enters username and password
3. Manager clicks “Login”
4. Manager selects “Finances”
5. Manager selects “Transaction History”

### Scenario 4: View Employee Wages

Navigation:

1. Manager clicks “Login”
2. Manager enters username and password
3. Manager clicks “Login”
4. Manager selects “Finances”
5. Manager selects “Employee Wages”

### Scenario 5: View Employee Schedule

Navigation:

1. Manager clicks “Login”
2. Manager enters username and password
3. Manager clicks “Login”
4. Manager selects “Scheduling”

### Scenario 6: View Restaurant Traffic

Navigation:

1. Manager clicks “Login”
2. Manager enters username and password
3. Manager clicks “Login”
4. Manager selects “Restaurant Traffic”

### Scenario 7: View Menu and Ratings

Navigation:

1. Manager clicks “Login”
2. Manager enters username and password
3. Manager clicks “Login”
4. Manager selects “Menu and Ratings”

## Domain Analysis

### Domain Model

#### Concept definitions:

Responsibility Description	Type	Concept
<b>R-01:</b> Customer accesses the menu on his tab to place order.	D	OrderItem
<b>R-02:</b> System sends customer's order to the restaurant.	K	Order
<b>R-03:</b> System knows and displays status of the order for customer viewing	K	OrderStatus
<b>R-04:</b> Customer accesses the bill payment options on his tab.	D	Check
<b>R-05:</b> System keeps track of entire table's bill and tip calculations.	K	Check
<b>R-06:</b> In case payment using cash is selected, the waiter returns to system to retrieve check	D	Check
<b>R-07:</b> System receives payment and notifies busboy to clean table.	D	CleanTable
<b>R-08:</b> Chef receives all the order in the order queue	D	Chef
<b>R-09:</b> Chef accesses the queue and modifies it according to the order status.	D	OrderQueue
<b>R-10:</b> Chef displays the order status for every customer	K	OrderDisplay
<b>R-11:</b> Chef notifies the waiter when an order is ready	K	Waiter
<b>R-12:</b> Order queue is updated after each order is prepared and ready to be placed on the customer's table	D	OrderStatus
<b>R-13:</b> Notify busser to clean the table after a customer has left/has paid.	K	CleanTable, Busser
<b>R-14:</b> Waiter handles the table reservation system and updates it	D	TableRecord
<b>R-15:</b> Coordinate actions of concepts associated with this use case and delegate the work to other concepts	D	Controller
<b>R-16:</b> Render the retrieved records into an HTML document for sending to the actor's Web browser for display	D	PageMaker

## LITTLE BITS

---

<b>R-17:</b> HTML document that shows the actor the current context, what actions can be done, and outcomes of the previous actions	K	InterfacePage
<b>R-18:</b> Change employee information including position, status, wage, contact information, etc.	D	InfoChanger
<b>R-19:</b> Notify Manager about different inventory alerts	D	Notifier
<b>R-20:</b> Prepare a database query that best matches the actor's search criteria and retrieve financial information, employee records, and inventory information from the database	D	Database Connection
<b>R-21:</b> List of Menu item ratings for further investigation and comment descriptions	K	Investigation Request
<b>R-22:</b> Display floor traffic of the restaurant	K	TrafficDisplay
<b>R-23:</b> Prediction of display of profit projections	D	Predictor

### Association Definitions:

Concept Pair	Association Description	Association Name
Customer ↔ Order	Customer orders items and places order.	Conveys Requests/ Requests Save
Customer ↔ Rate	Customer rates food items and customer experience	Provides data
Time Estimation ↔ Customer	Customer receives estimation on food readiness	Provides data
Customer ↔ Waiter	Customer requests waiter's assistance	Conveys Request
Chef ↔ OrderQueue	Chef receives the order items from the order queue	Receives Data
Chef ↔ OrderDisplay	Customer can see the order status on his tablet displayed by the customer	Provides data
Waiter ↔ OrderStatus	Waiter periodically requests updates of the ordered items so that when the order is ready, they can deliver it the right table	Requests Updates
Waiter ↔ TableRecord	Waiter marks the table as occupied when the customers seat themselves.	Provides Data

## LITTLE BITS

---

Chef ↔ OrderStatus	Chef passes updates to order status. (Notifies if the order is ready or not)	Provides Data
Busser ↔ CleanTable	Busser cleans the table once the payment has been processed.	Receives Data
Busser ↔ TableRecord	Busser updates the status of tables online once they have been cleaned up.	Provide Data
Controller ↔ Page Maker	Controller passes requests to PageMaker and receives back pages prepared for displaying	Convey Requests
Page Maker ↔ Database Connection	Database Connection passes data retrieved data to PageMaker to render them for display	Provides Data
Investigation Request ↔ Database Connection	Database Connection receives the menu items and corresponding ratings and sends it to Investigation Request to display more details about each menu items	Show Data
Database Connection ↔ Notifier	Database Connection passes the inventory information to Notifier which displays any inventory-related notifications	Notify
InfoChanger ↔ Database Connection	Database Connection passes retrieved data to the InfoChanger to change employee information	Change Info
Controller ↔ Database Connection	Controller passes search requests to Database Connection	Conveys Requests
Page Maker ↔ Interface Page	Page Maker prepares the Interface Page	Prepares
Database Connection ↔ Predictor	Database Connection receives financial information and sends it to Predictor to estimate future profits	Estimates

## Attribute Definitions

Concept	Attributes	Attribute Description
OrderItem	Name	Name of menu item
	Cost	Cost of menu item
	Category	Category of menu items

## LITTLE BITS

---

<b>Order</b>	Customer info	Information about the customer: name or ID
	Order number	The ID number created for each customer order
	Table number	Number associated with the customers table
	Check	The most recent customers bill
<b>Check</b>	ID	Customers Associated ID
	Items	Food items ordered
	Cost for each items	Cost for each item ordered
	Total Cost	The total cost for all items ordered
	Tip	Customers preferred tip amount payment
<b>Manager</b>	Name	Name of Manager
	Manager ID	Manager has a unique identification credential associated with them for login
	Privileges	Manager has specific privileges including menu modifications, hire/terminate employees, track profit/loss, etc.t0
<b>Waiter</b>	Name	Name of the waiter
	Position	Position held by the employee at the restaurant
	Employee ID	Each employee has a unique identification credential associated with them for login
	Contact Information	The employee's address, email address, phone number, etc.
	Table Association	The tables to which the waiter is serving
<b>OrderQueue</b>	Orders	Orders are placed on a queue within the system, which can be accessed by the chef
<b>TableStatus</b>	Clean and Empty	Customers can be seated in it
	In Use	There are customers seated in the table
	Dirty	There are no customers seated in the table and it needs to be cleaned

## LITTLE BITS

---

<b>Notifier</b>	Inventory quantity	Signals to user(manager) that items in the inventory are running low
<b>Investigation Request</b>	Menu	List of menu items, ratings, and comments for investigation
<b>InfoChanger</b>	Employee change	Change employee attributes and values.
<b>PageMaker</b>	Records	Information retrieved from the database to be rendered
	Document	HTML document created to display retrieved information to the user
<b>InterfacePage</b>	Document	HTML document displaying to the actor the current context, what actions can be done, and outcomes of the previous actions
<b>TrafficDisplay</b>	Traffic	Information relative to the floor activity in the restaurant
<b>Predictor</b>	Customer Traffic	Information pertaining to the amount of customers coming to the restaurant
	Finances	Information relative to the profits and losses as well as payroll costs

## Traceability Matrix

Use case	Order	TimeEstimation	OrderItem	Check
UC-1	X		X	
UC-2		X		
UC-3	X		X	
UC-4			X	
UC-5			X	X
UC-6	X		X	

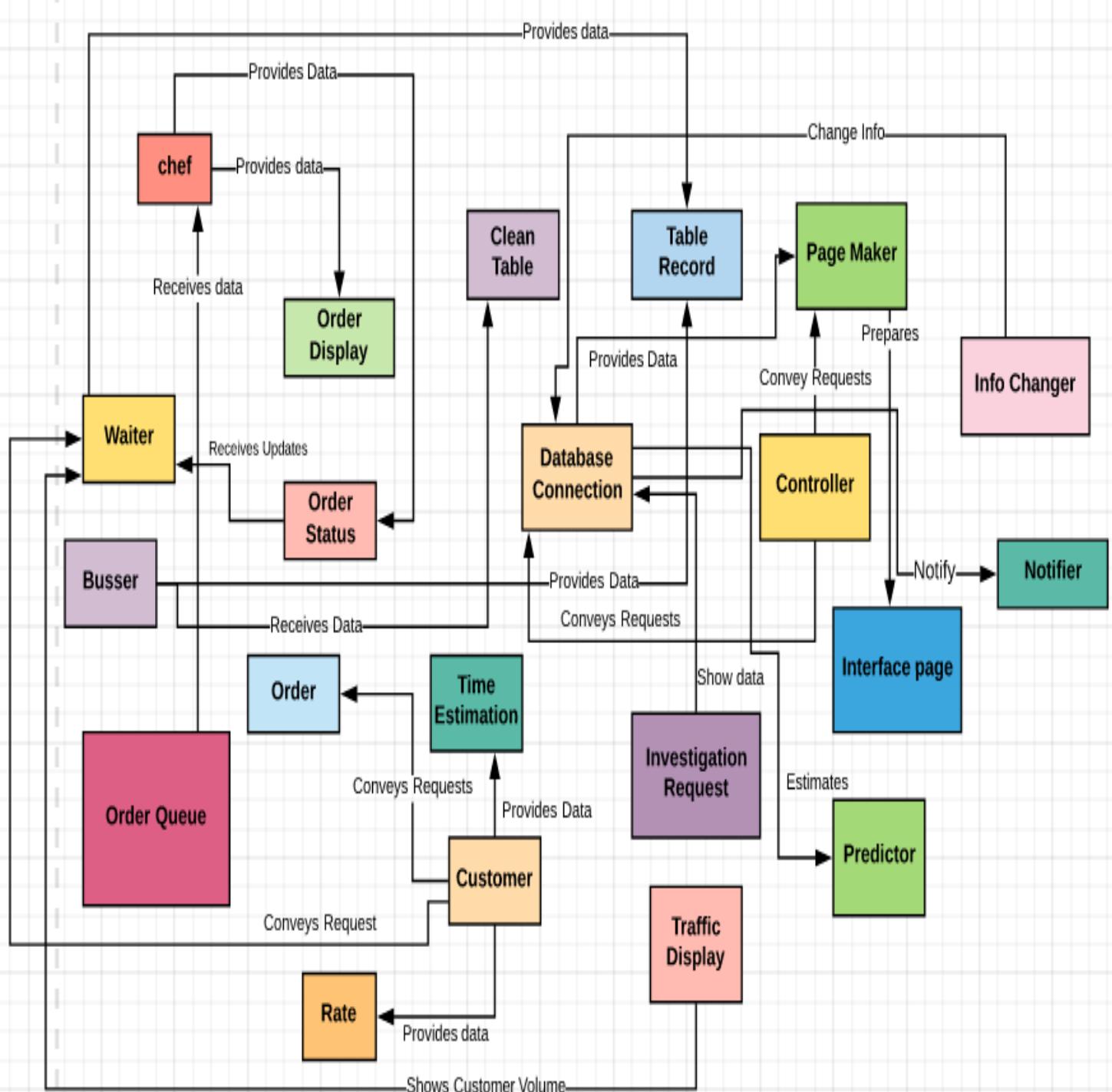
## LITTLE BITS

---

Use case	OrderQue	Order Display	Order Status	CleanTable	TableRecord	Order	Check
UC-1	X	X				X	X
UC-2				X	X		
UC-3		X				X	X
UC-4	X					X	X
UC-5	X						
UC-6	X	X	X	X	X		

Use Case	Controller	Page Maker	Interface Page	Info Change	Notifier	Database Connection	Investigation Request	Traffic Display	Predictor
UC-1	X	X	X		X	X	X		
UC-2	X	X	X			X	X	X	
UC3	X	X	X			X	X		
UC-4	X	X	X		X	X	X		
UC5	X	X	X	X		X	X		
UC6	X	X	X			X	X	X	X

## Domain Model Diagram



## Systems Operation Contracts

<b>Name:</b>	Order Management
<b>Responsibilities:</b>	Allow the customer to place an order and to view when said order will be prepared
<b>Use Cases:</b>	UC-1
<b>Exceptions:</b>	None
<b>Preconditions:</b>	Menu is displayed on the customer's tablet
<b>Postconditions:</b>	Order queue is displayed on the chef's tablet; order for designated table is displayed on the waiter's tablet

<b>Name:</b>	Traffic Monitoring
<b>Responsibilities:</b>	Place reservation, Place Orders, Mark Tables Empty, Mark Tables Cleaned
<b>Use Case:</b>	UC-2
<b>Exception:</b>	None
<b>Preconditions:</b>	The user creates a reservation, come to the restaurant, and places an order
<b>Post-conditions:</b>	The waiter can see when a table is done and bussers can mark when tables are cleaned

<b>Name:</b>	Customer Ordering
<b>Responsibilities:</b>	View the menu, ordered desired food and pay the bill
<b>Use Cases:</b>	UC-3
<b>Exceptions:</b>	None
<b>Preconditions:</b>	A tablet open on the customer interferences
<b>Postconditions:</b>	The customer is able to order and get assisted with food orders.

## LITTLE BITS

---

<b>Name:</b>	Payment Process
<b>Responsibilities:</b>	Pay the bill successfully
<b>Use Cases:</b>	UC-4
<b>Exceptions:</b>	None
<b>Preconditions:</b>	The Customer has clicked and opened the Pay Your Bill tab and has chosen one of the methods of payment.
<b>Postconditions:</b>	The customer pays the bill with or without the Waiter's assistance depending on which option the customer chooses.

<b>Name:</b>	Managing Employees
<b>Responsibilities:</b>	See Schedule, Log In/Out, Adjust Schedule Preferences, Auto Clock In/Out
<b>Use Case:</b>	UC-5
<b>Exception:</b>	None
<b>Preconditions:</b>	The user logs in which automatically clocks them in
<b>Post-conditions:</b>	Preferences are saved and user logs out

<b>Name:</b>	Restaurant Management
<b>Responsibilities:</b>	To allow the manager of the restaurant to control employee scheduling and pay rates, view inventory status, manage finances, and monitor floor traffic in the restaurant
<b>Use Cases:</b>	UC-6
<b>Exceptions:</b>	None
<b>Preconditions:</b>	Manager login page is displayed on the manager's tablet
<b>Postconditions:</b>	Manager changes to previously stored database information is saved; profit projection is calculated and displayed as desired by the manager

## Mathematical Model

### Table Reserving Model:

This algorithm uses the party size and compares it against the available tables to see which table should be designated to the customer. The algorithm utilizes a sorted list (which is sorted and prioritized by a first come first serve basis) to take input from and then compares the party sizes of all customers and find a suitable table.

```
while (sorted list of customers does not equal 0)
{
    while (iterating through the list of available tables)
    {
        while (iterating through sorted list of customers)
        {
            if (size of table is equal to a party size in the customer list)
            {
                //Display customer name and table on the tablet
                //Along with map of restaurant highlighting the table
            }
            else if(size of table is greater than party size and no other party
                    matches      table size)
            {
                //Display customer name and table on tablet
                //Along with map of restaurant highlighting the table
            }
        }
    }
}
```

### **Bill payment Algorithm:**

As customer place their order the bill amount gets added up according to the price and the quantity the item ordered. When the customer wants to pay the bill, the total bill amount is displayed with three modes of payment. Based on the customer's preferred payment method, the algorithm displays the corresponding payment page where the customer has the option to tip and rate the food and share his/her customer experience.

```
while (bill of customer not equal to zero)
{
    if(payment method equals credit card)
    {
        // Display Tipping option according to the bill amount
        //calculate bill amount accordingly and display payment page on tablet
        // Acknowledge payment
        //Display rating option
    }
    if(payment method equals paypal)
    {
        // Display Tipping option according to the bill amount
        //calculate bill amount and display paypal page
        // Acknowledge payment
        //Display rating option
    }
    if(payment method equals cash)
    {
        //send notification to athe table waiter to assist the customers
        //Display rating option
    }
}
```

### Order Status Algorithm:

Each customer's order is placed in an order queue. As soon as the customer places an order, it is appended in the queue following the FIFO sequence and the corresponding table number is recorded. When the order is just placed, it is appended in the queue and the order status says "Being prepared...". When the chef receives the customer's order, that is, the order is at the beginning of the order queue, the order status is immediately changed to "Cooking...". As soon as the chef is done preparing, the order status changes to "Order is ready".

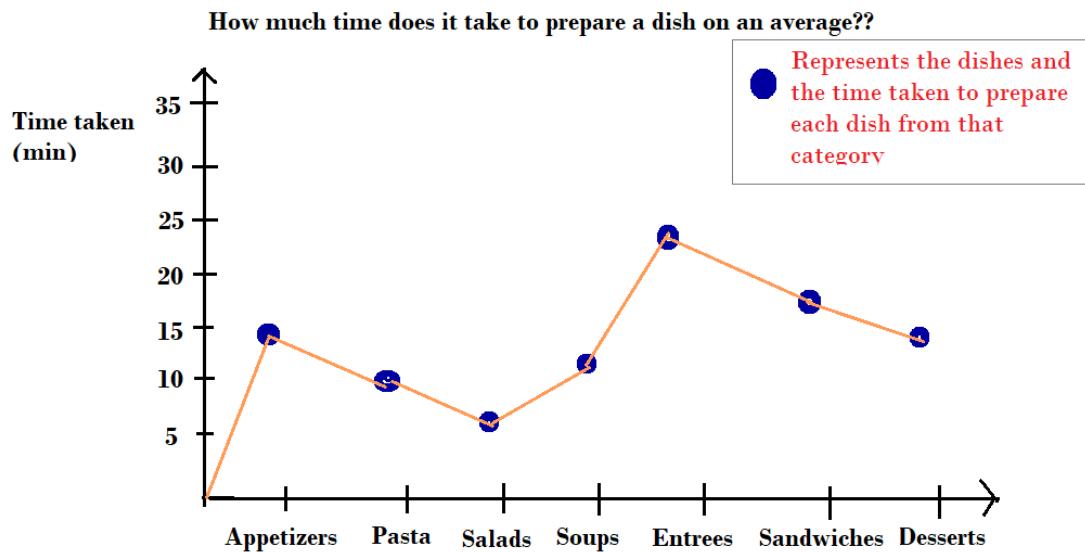
```
class OrderQueue:  
    def __init__(self):  
        self.orderItems = []  
    def isEmpty(self):  
        return self.orderItems == []  
    def enqueue(self, orderItems)  
        self.orderItems.insert(0,orderItems)  
    def dequeue(self):  
        return self.orderItems.pop()  
    def size(self):  
        return len(self.orderItems)  
  
q = OrderQueue()  
def orderStatus:  
    while(q.size!=0)  
        if(q.isEmpty == True):  
            print("No orders in progress")  
        else:  
            if(q.size ==1)  
                print("Cooking...")  
                setTimer()  
                //as soon as the timer goes off, print("Order is ready")  
                q.dequeue()  
                //notify waiter  
                print("No more remaining orders")  
            else:  
                print("order number 1 out of q.size() is cooking")  
                print("(q.size-1) orders are currently in progress")  
                setTimer()  
                //as soon as the timer goes off, print("order is ready")  
                //notify waiter  
                q.dequeue()  
                q.enqueue(order #x)
```

### Polynomial Regression:

For the financial management system, a mathematical model will be used in making predictions on the level of the profit the restaurant will experience given the customer traffic in a given amount of time and the popularity of the restaurant's menu items. With two variables being quantity of items sold and total income collected, we will create an estimation of total profits collected over a time period specified by the user, the manager. Polynomial regression is a technique which takes the form:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_h X^h$$

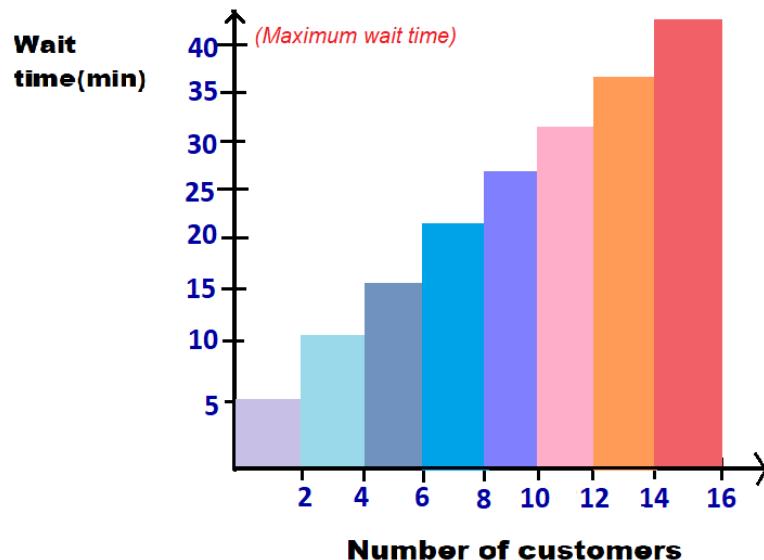
Where  $X$  represents the total quantity of items sold,  $Y$  represents the total income collected by said menu items,  $\in$  represents the error and  $h$  represents the degree of the polynomial. The manager will be able to choose the degree of the polynomial to use in making a projection on the profits of the restaurant. Using established data points taken over a period of time, two months for example, the system will begin to make calculations on profit projections for months and/or years in advance, given input for the desired time period from the manager.



## LITTLE BITS

---

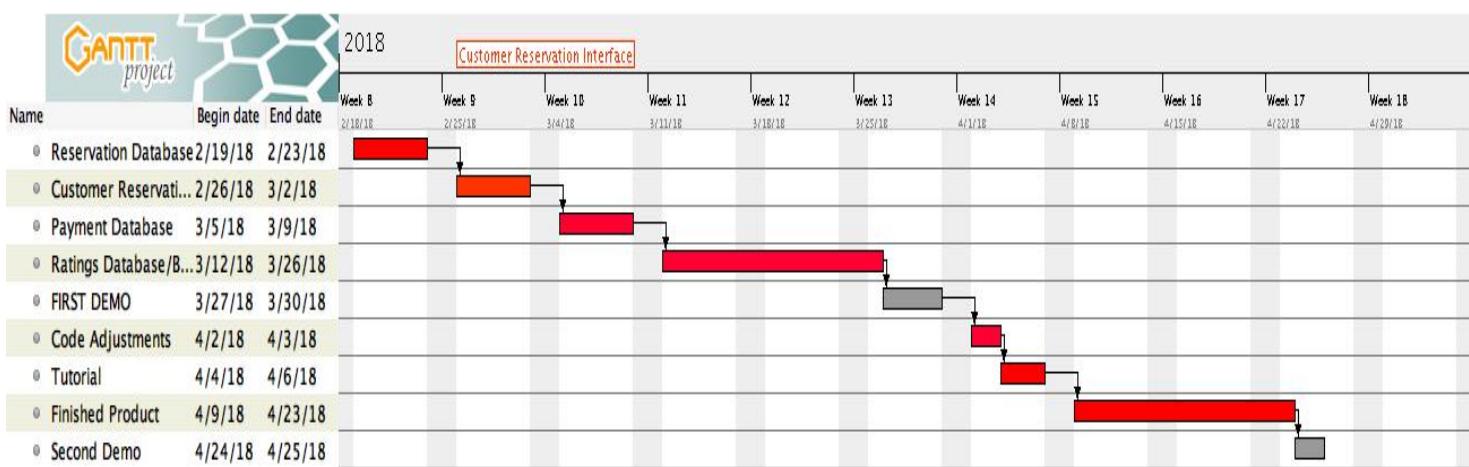
AVERAGE WAIT TIME OF CUSTOMERS



---

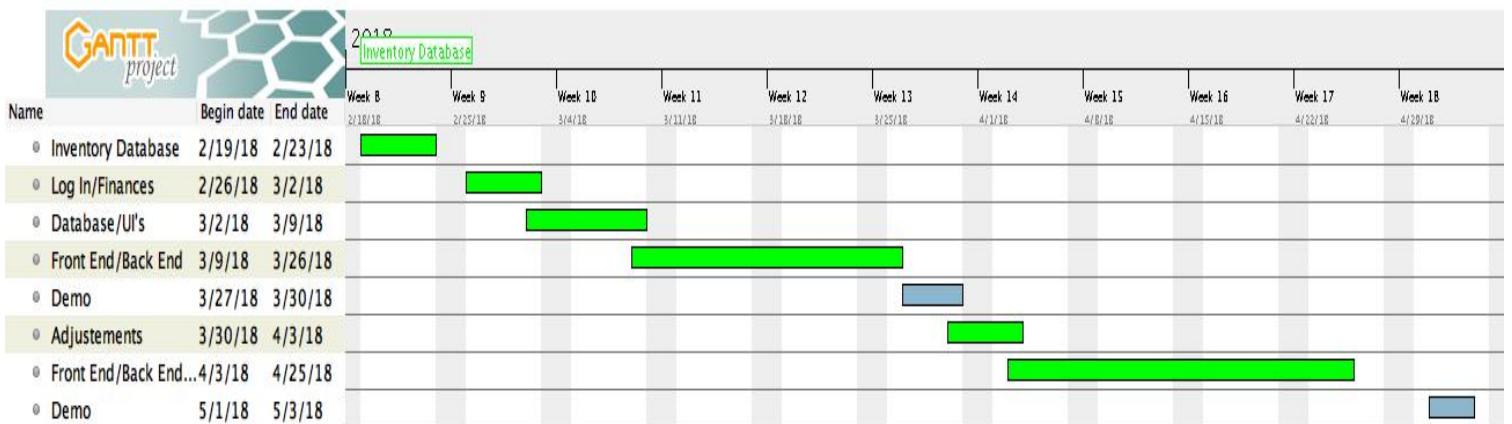
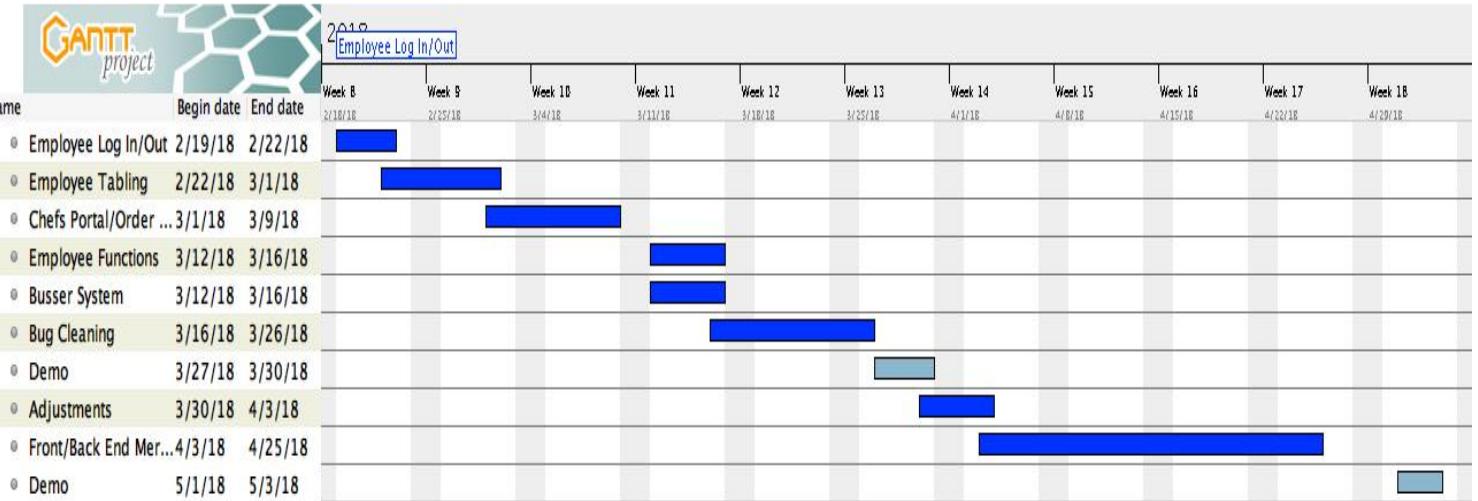
## Plan of Work

Key:



## LITTLE BITS

---



## Product Ownership

CUSTOMERS	EMPLOYEES (WAITERS/ CHEFS/BUSERS)	MANAGER
Reservation ( <a href="#">Python</a> FIFO)	Log in/ Log out (Web clock) ( <a href="#">Javascript</a> )	Inventory ( <a href="#">Java</a> , <a href="#">MySQL</a> )
Ordering ( <a href="#">Python</a> FIFO, <a href="#">MySQL</a> )	Tabling (Waiter) ( <a href="#">Java</a> / <a href="#">Python</a> )	Managing the employee system ( <a href="#">Python</a> )
Payment w/ tipping ( <a href="#">Python+Javascript</a> )	Ordering ( <a href="#">MySQL</a> )	Financial management system ( <a href="#">Python</a> )
Review and comments ( <a href="#">Javascript</a> )	Reservation ( <a href="#">Python</a> )	Log in / Log out ( <a href="#">Javascript</a> )
Hagar, Nitya, and Dafna	Manvi, Zach, and Esraa	John, Fareen, and Kevin

**Dafna, Hagar, and Nitya:**

Completed tasks: Created the first draft of user interface expectations and step by step page layout.

Current tasks: Create a functional reservation system for the customers' convenience.

Future tasks: Create databases to address each function:

- Reservations and selection of specific tables
- Ordering through the digital menu and adding specifications
- Showing the bill and averaging a tip
- Reviewing and leaving comments regarding the food and experience
- SOS button to call the waiter

## LITTLE BITS

---

### Zach, Manvi, and Esraa:

Completed tasks: Created the first draft of user interface expectations and step by step page layout.

Current tasks: Create a functional tabling system for the employees

Future tasks: Create databases to address each function:

- Tabling
- Ordering Queue for Waiters and Chefs
- Merging functions from Customer Team with Employee Team such as waiter assistance, or payment
- Bussing system to mark tables clean or dirty

### John, Kevin, and Fareen:

Completed tasks: Created the first draft of user interface expectations and step by step page layout.

Current tasks: Create a functional inventory system for the managers' convenience.

Future tasks: Create databases to address each function:

- Finances
- Inventory suggestions
- Traffic Monitoring

Section	Sub-Section	Pts	Esraa Abdelmotteleb	Manvi Agarwal	John Alcantara	Hagar Elshentenawy	Lakshmi Sai Nitya Garudadri	Kevin Honeker	Zachary Joseph	Fareen Pourmoussavian	Dafna Shochat	Teams
Project Management [10]	Vision	2	11%	11%	11%	11%	11%	11%	11%	11%	11%	Customer
	Quality	4	8%	8%	8%	8%	8%	8%	36%	8%	8%	Employee
	Communication	2	9%	12%	11%	11%	11%	11%	12%	11%	11%	Manager
Customer Statement of Requirements [9]	Leadership	2	8%	8%	17%	17%	8%	8%	17%	8%	8%	
	Problem Statement	6	11%	11%	11%	33%	0%	11%	11%	11%	0%	
System Requirements [6]	Glossary of Terms	3	11%	11%	17%	0%	0%	17%	11%	0%	33%	
	Enumerated Functional Req.	2	0%	15%	11%	0%	33%	11%	18%	11%	0%	
	Enum. Nonfunctional Req.	2	0%	14%	33%	0%	33%	0%	19%	0%	0%	
Functional System Requirements [30]	On-Screen Appearance Req.	2	0%	33%	0%	17%	17%	17%	0%	17%	0%	
	Stockholder/Actor Goals	2	0%	17%	33%	11%	11%	0%	17%	0%	11%	
	UC: Casual Description	7	11%	11%	11%	11%	11%	11%	11%	11%	11%	
	UC: Diagram	5	0%	100%	0%	0%	0%	0%	0%	0%	0%	
	UC: Traceability Matrix	3	0%	17%	11%	19%	10%	11%	17%	11%	4%	
	UC: Fully-Dressed Description	9	0%	17%	17%	17%	17%	8%	17%	8%	0%	
	System Sequence Diagram	5	33%	0%	33%	17%	17%	0%	0%	0%	0%	
User Interface Specification [15]	Preliminary Design	10	0%	0%	0%	0%	0%	15%	0%	5%	80%	
	Effort Estimation	5	33%	0%	0%	17%	17%	0%	0%	33%	0%	
Domain Analysis [25]	Domain Model Derivation	5	0%	90%	10%	0%	0%	0%	0%	0%	0%	
	DM: Concept Definitions	2	0%	12%	11%	0%	33%	11%	21%	11%	0%	
	DM: Association Definitions	3	0%	33%	0%	33%	0%	0%	0%	33%	0%	
	DM: Attribute Definitions	3	33%	0%	17%	33%	0%	0%	0%	17%	0%	
	DM: Traceability Matrix	3	0%	33%	0%	0%	33%	33%	0%	0%	0%	
	System Operation Contract	4	0%	0%	33%	0%	33%	0%	33%	0%	0%	
	Mathematical Model	4	0%	33%	33%	17%	17%	0%	0%	0%	0%	
	Plan of Work	3	0%	0%	0%	0%	33%	0%	33%	33%	0%	
Plan of Work [5]	PoW: Product Ownership	2	3%	15%	11%	0%	0%	11%	15%	11%	33%	
	<b>TOTAL</b>	<b>100</b>	7.168	19.76	12.22	10.96	11.84	7.4975	10.25	9.1525	11.78	

## References

<http://www.ece.rutgers.edu/~marsic/books/SE/projects/Restaurant/2015-g3-report3.pdf>

<https://neutrium.net/mathematics/least-squares-fitting-of-a-polynomial/>

<https://onlinecourses.science.psu.edu/stat501/node/324>

<http://interactivepython.org/runestone/static/pythonds/BasicDS/ImplementingaQueueinPython.html>