

CSCI 441-A

Team B

Report #2 Part 1: Restaurant Automation

October 2, 2023

[https://github.com/ivanvelocastaneda/CSCI441\\_A-Team-B-Project.git](https://github.com/ivanvelocastaneda/CSCI441_A-Team-B-Project.git)

Bjarni Jonsson

Cheikh Abdoulaye Faye

Sokhna Khady Mbacke

Ivan Velo Castaneda

## Table of Contents

### **Table of Contents**

### **Individual Contributions Breakdown**

### **Summary of Changes**

#### **1. Customer Problem Statement**

- a. Problem Statement
- b. Decomposition into Sub-Problems
- c. Glossary of Terms

#### **2. Goals, Requirements, and Analysis**

- a. Business Goals
- b. Enumerated Functional Requirements
- c. Enumerated Non Functional Requirements
- d. User Interface Requirements

#### **3. Use Cases**

- a. Stakeholders
- b. Actors and Goals
  - i. Initiating Actors
  - ii. Participating Actors
- c. Use Cases
  - i. Casual Description
  - ii. Use Case Diagram
  - iii. Traceability Matrix
  - iv. Fully Dressed Description
  - v. System Sequence Diagrams

#### **4. User Interface Specification**

- a. Preliminary Design
- b. User Effort Estimation

#### **5. System Architecture**

- a. Identifying Subsystems
- b. Architecture Styles
- c. Mapping Subsystems to Hardware

- d. Connectors and Network Protocols
- e. Global Control Flow
- f. Hardware Requirements

## **6. Project size estimation based on use case points**

## **7. Analysis and Data Modeling**

- a. Conceptual Model
  - i. Concept Definitions
  - ii. Association Definitions
  - iii. Attribute Definitions
  - iv. Traceability Matrix
- b. System Operation Contracts
- c. Data Model and Persistent Data Storage
- d. Mathematical Model

## **8. Project Management**

## **9. References**

### **Individual Contributions Breakdown**

<b>Topics</b>	<b>Bjarni</b>	<b>Cheikh</b>	<b>Ivan</b>	<b>Sokhna</b>
<b>Individual Contributions Breakdown</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>
<b>Customer Problem Statement</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>
<b>Goals, Requirements, and Analysis</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>
<b>Use Cases</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>
<b>User Interface Specification</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>

<b>System Architecture</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>
<b>Project size estimation based on use case points</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>
<b>Analysis and Data Modeling</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>
<b>Project Management</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>
<b>References</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>

## 1. Customer Problem Statement

### a. Problem Statement

#### **Manager:**

Running a restaurant and overseeing employees is a challenging responsibility. My daily tasks can potentially be overwhelming at times. Some tasks include, keep track of which employees are on duty, figure out the payroll for every pay period, ensure employees punctuality and their compensation. Any means to reduce the effort required for these tasks would be greatly appreciated.

Additionally, I should have the flexibility to update the menu with ease to ensure that we are matching the times we are actually serving and make modifications if needed in case we run out of the certain item. The restaurant's floor plan is subject to constant change, especially when large parties come in. I need to have the ability to make real-time adjustments to ensure staff have an accurate floor plan at their disposal. Farm to table serves as an employee portal, enabling employees to conveniently clock in and out while automatically calculating their compensation and hours worked. This feature will significantly make my life easier as I will no longer need to manually log employee hours or perform pay calculations. The site will handle these tasks smoothly. Furthermore, Farm to Table provides a swift interface for menu editing and floor adjustments, enhancing overall efficiency.

**Waiter/Waitress:**

Working as a waiter/waitress can be an incredibly demanding job, requiring us to be constantly on the move within the restaurant. We have to take customers orders, keep track of table orders, deliver orders quickly to ensure food stays hot and manage the process of closing out checks when customers are ready to pay their bill. With multiple tables in the restaurant, it can be a difficult process to keep track of what tables ordered what items and to determine what tables need to be cleaned and prepared for the next customer. Unfortunately, sometimes we have to send dishes back if it is not up to the customers standards or find something wrong with it. A site like Farm to Table would be immensely beneficial in allowing us to take customer's orders, keep track of them in the site, and send them directly to the kitchen. We would also love to receive notifications whenever our food is finished getting prepared so the food does not stay on the window for too long. Lastly, we would like to close down tabs and input customer's payment (whether they paid cash or card) in the site to ensure we are making the correct amount of tips at the end of the day.

**Host/Hostess/Busboy/Busgirl:**

When I am welcoming guests into the restaurant, I would like to ensure their experience goes as smoothly as possible. As groups of multiple sizes arrive, they often have specific requests and preferences regarding their seating arrangements based on the number of people in their and where they would like to seat within the restaurant. Sometimes, there might be a five-person party seeking a booth or multiple tables put together and it is our responsibility to ensure there is an available spot and they do not wait too long to get seated. However, it is occasionally uncertain how long it would be until tables become free to accommodate a party's seating preferences.

Finding an appropriate table for our guests can pose a challenge since we do not know the current status of tables. We have to physically go and check if tables are clean or not. It would be highly beneficial to have a means of easily tracking which tables are occupied, unoccupied, and need to be cleaned without having to

go through multiple stacks of paper. This information would enable us to provide guests with an estimation of when a suitable seating arrangement can be made. A site like Farm to Table can help us identify which tables are occupied, unoccupied, or need to be cleaned.

**Customer(s):**

I love going out to eat, but it can be frustrating at times. It is not the staff's fault whatsoever. Sometimes they are incredibly busy and a waiter/waitress can take a long time to get to me unless they walk past me or make direct eye contact with me. It would be great if we could have a way to get a server's attention or order drinks/appetizers without having to wait for my server to get to me. It would speed up the process of time spent waiting.

Some sort of device on the table with a menu available to my party would be great in placing orders immediately without having to wait for my server to get to me if they are busy. It would also be a great way to have an interactive menu that could offer more information on each item. The ability to have some sort of button on the device to call my server over to my table would solve the problem of waiting too long to be noticed or ask our server for help.

**Cook/Chef:**

Working in a kitchen requires a lot of patience and it can be a demanding job at times. We have to make sure the food is up to standard while simultaneously making it in a reasonable amount of time. We also have to make sure the order is accurate because it can be quite horrible if an order is made incorrectly. A person could potentially get hurt if their order contains an item they are allergic to as a result of miscommunication between us and the waiting staff. It is up to us and the waiting staff to ensure we have a happy customer because that means they would be more likely to come back in the future. It can be annoying at times trying to track a waiter/waitress down to let them know their food is ready to be delivered to their table. The food could get cold and take up space on the window, potentially risking a broken plate or the quality of the meal.

It would be of great help if we could have something to display all incoming orders. We would also love to have a way to let servers know that their food is

finished without having to yell out their names to reduce the time the food sits on the window.

Farm to Table will generate a queue of incoming orders with a timestamp so that we are aware what food orders need to be made first so that we can stay on track. It would eliminate the necessity of handwritten orders thus eliminating any confusion in the server's poor handwriting since we would digitally have the details and requests of the server's tables. The site will allow us to send servers a ping to let them know they can come and pick up their meals so that they may be taken to their tables.

b. Decomposition into Sub-problems

Already described within each problem statement.

c. Glossary of Terms

**Cook/Chef:** A person who prepares and cooks food, typically as a profession or as a skilled practitioner

**Customer:** An individual or group of individuals who visit an eating establishment to purchase and consume food and beverages

**Employee Portal:** It serves as a centralized hub where employees can access a variety of resources, tools, and information related to their employment and workplace

**Floor Plan:** Refers to the physical arrangement and design of tables, seating areas, and other elements within the dining area of a restaurant

**Host/Hostess:** An employee responsible for managing the front-of-house operations and ensuring a smooth and welcoming experience for diners

**Manager:** An individual who holds a position of authority and responsibility within an organization or business

**Menu:** A written or printed list of food and beverage items that a restaurant offers to its customers.

**Queue:** Typically refers to a line or waiting area where orders wait their turn to be cooked at a restaurant

**Restaurant Automation:** Refers to the use of technology and automated systems to streamline and improve various aspects of restaurant operations

**Screen:** Can refer to various digital displays or monitors used within the establishment for different purposes, often leveraging technology to enhance the dining experience

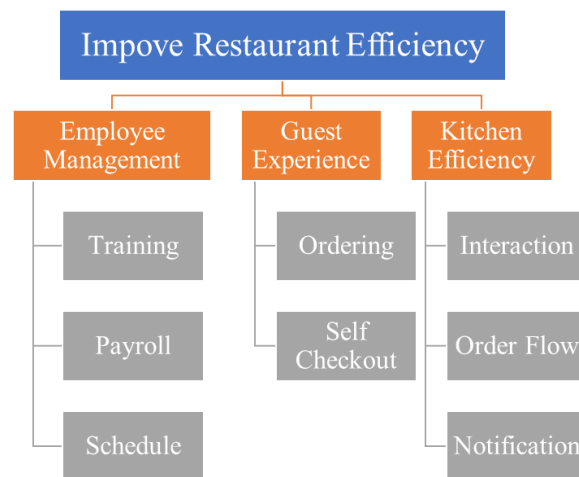
**Tip:** Refers to an additional sum of money that customers voluntarily leave for the staff as a token of appreciation for the service provided

**Waiter/Waitress:** Often referred to as a server, is an individual employed in the hospitality industry, typically at restaurants, cafes, or other dining establishments

**Window:** A platform used to place plates in

## 2. Goals, Requirements, and Analysis

### a. Business Goals:

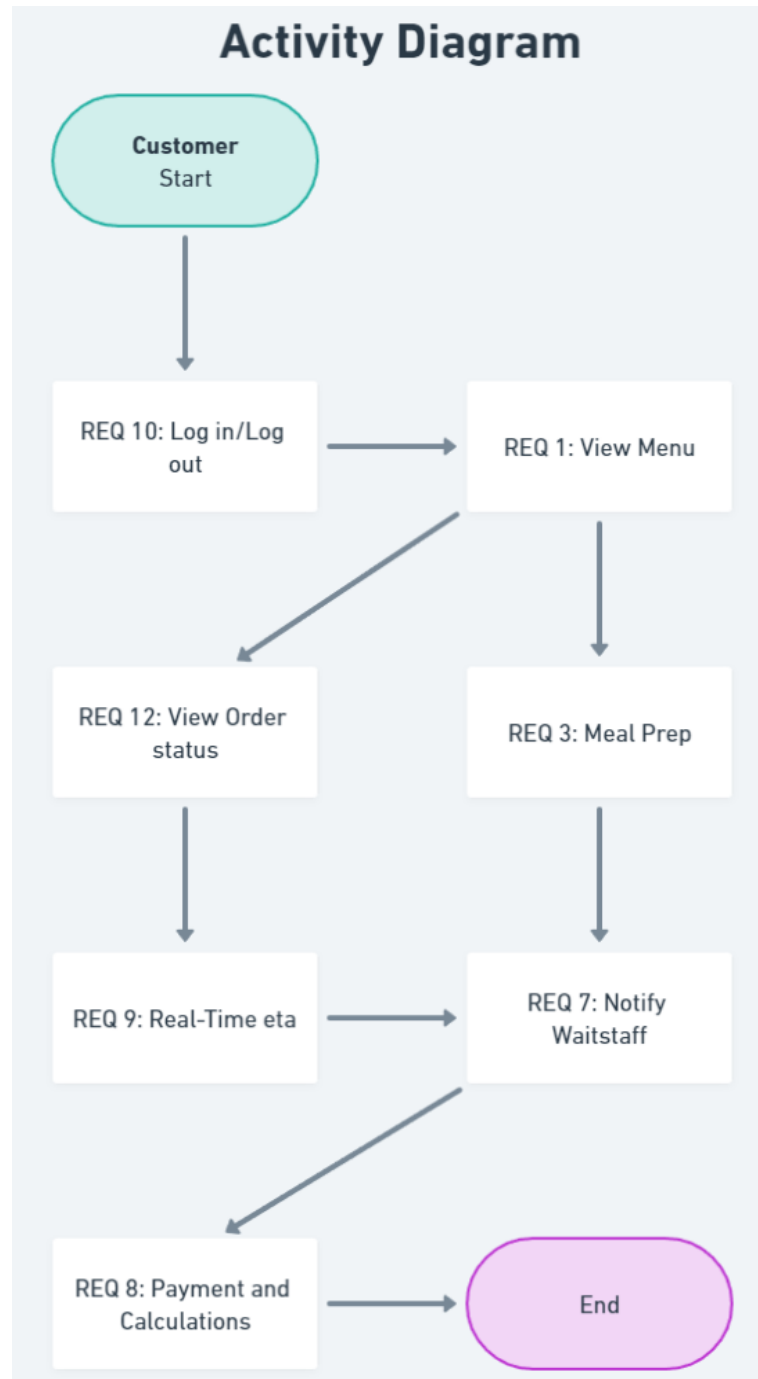


### b. Enumerated Functional Requirements:

Identifier	Priority	Requirement
REQ-1	5	The system will allow customers to select items from the menu and put in an order through an interactive screen located .
REQ-2	5	The system provides the host with an interactive screen that displays the current table layout in the restaurant.
REQ-3	4	The system will notify the chef with new orders and place them in the kitchen queue.
REQ-4	4	The system should automate and calculate employee



		hours and compensation.
REQ-5	5	The system should allow managers to easily update menu items and notify about unavailable dishes.
REQ-6	4	Employees should have the ability to clock in and out through an employee portal.
REQ-7	3	The system should offer waitstaff a notification system for when dishes are ready.
REQ-8	4	The system should digitally process checks and payments to assist waitstaff in tip calculation.
REQ-9	3	The system should provide real-time wait time estimations for guests.
REQ-10	5	The system should allow customers and employees to login and logout.
REQ-11	2	The system should allow managers to add/remove employees from the system.
REQ-12	3	The system should allow customers and employees to view an order's status.



c. Enumerated Nonfunctional Requirements:

Identifier	Priority	Requirement
NFREQ-1	5	The system provides customers with an interactive screen that displays current table layout in the restaurant.

NFREQ-2	5	The system must be secure, protecting all data especially payment and personal information in compliance with regulations.
NFREQ-3	4	The system should integrate seamlessly with existing systems or software used in the restaurant.
NFREQ-4	4	The system should be scalable to accommodate the restaurant's growth or changes.
NFREQ-5	3	The system should offer training modules or guides to assist employees in understanding the functionalities.
NFREQ-6	3	The system should have a feedback mechanism for users to report issues or provide suggestions for improvement.
NFREQ-7	4	The system must have a high availability, ensuring it remains operational during peak restaurant hours.
NFREQ-8	3	The system should offer multi-language support for diverse customer bases.
NFREQ-9	3	The system should allow customers to add extras or remove items from order.
NFREQ-10	3	The system should allow everybody to view item ingredients.
NFREQ-11	3	The system should allow everybody to create a reservation.
NFREQ-12	2	The system should allow customers to create an rewards account for points
NFREQ-13	3	Allow customers to order take out

## d. User Interface Requirements

Identifier	Priority	Requirement
USREQ-1	5	Interactive Menu Display: The interface should provide customers with a visual and interactive menu. This should include clear images of dishes, concise descriptions, and the price. The interface should be intuitive for customers to place an order directly from this menu.

USREQ-2	4	Table Layout Visualization: Hosts and waitstaff should have a graphical interface displaying the current status of all tables in the restaurant (occupied, free, needs cleaning). It should be possible to update the table status in real-time.
USREQ-3	3	Interactive Order Display: The interface should provide customers with a visual and order status screen. This should keep customers more informed of their order status.
USREQ-4	3	Customer Account Page: The interface provides customers with a web page where they can log into from home to place orders for take-out or make table reservations.

### 3. Use Cases

#### a. Stakeholders

- i. Restaurant Owners - They have an interest in using the system to help optimize efficiency in the restaurant and provide quality customer service for customers.
- ii. Employees and Managers - The system will streamline their workload of their process and make their job much easier.
- iii. Customers - The system will enrich their dining experience since they will have a chance to interact with it.
- iv. Developers - They have an interest in improving and implementing a system that would create solutions to small restaurants' problems without automation face on a daily basis.

#### b. Actors and Goals

##### i. Initiating Actors

Actor	Role	Goal
Manager	The manager is the employee in charge of managing the wait staff and the additional needs of the restaurant.	The goal of the manager is to manage employees and their schedules, keep track of inventory, monitor revenues and losses and also ensure restaurant customers have an enriching dining experience.

Employee	The employee is any type of wait staff at the restaurant, except for the manager.	The goal of the employee is to provide customers with an excellent dining experience.
Customer	The customer is a restaurant visitor who chooses to either dine in, order take out, views the menu, orders a meal, and pays for service	The goal of the customer is to have an enriching dining experience with minimal wait time and great service.

ii. Participating Actors

Chef	The chef is responsible for preparing and cooking food that is ordered by customers. They receive a queue of orders and prepare them as they come in. They then send a ping to the servers to let them know their food is ready.
Host/Hostess	The host/hostess is in charge greeting customers and seating them. They can see table status (occupied, unoccupied, dirty) through the database. After seating the customers, they then mark the table as occupied.
Waiter/Waitress	The waiter/waitress is responsible for taking orders from customers, sending them to the kitchen, and serving the food when it is ready. They receive notifications from the kitchen when their meal is ready so they can go and serve it.
Database	The database is a system that records a customer's order, table layout of the restaurant, menu options, etc. It acts as the storage of all information for our site to function properly.

c. Use Cases

i. Casual Description

**UC-1: Clocking in/Clocking out**-Allow employees to clock in when they first come in to work or after they take a break and clock out after finishing their shift or before taking a break.

**Derivations:** REQ-6

**UC-2: Log in/Log out**-Allow employees and customers to log in/log out into the system which will determine what interface they will have access to.

**Derivations:** REQ-10

**UC-3: View Menu**-Allows employees and customers to view the items on the menu and item ingredients.

**Derivations:** USREQ-1 / NFREQ-10

**UC-4: Place Order**-Allows employees and customers to place orders and add/remove items from meals.

**Derivations:** USREQ-1 / REQ-3 / NFREQ-13 / REQ-1

**UC-5: View/Update Table Status**-Allows to view the table status of all tables whether occupied, unoccupied, dirty.

**Derivations:** USREQ-2

**UC-6: Make/View all reservations**-Allow only employees to view reservations. Also allow customers to make or cancel reservations.

**Derivations:** NFREQ-11

**UC-7: Print Out Reports**-Allow managers to print out employee's reports at the end of their shift.

**Derivations:** REQ-4

**UC-8: Payment**-Allows employees and customers to split the bill and allow customers to pay on the spot (credit-card reader) if they desire it.

**Derivations:** REQ-8

**UC-9: Checking guests are happy/Rating**-Allows customers to give feedback for the service provided throughout the visit.

**Derivations:** NFREQ-6

**UC-10: Order status**-Allows employees to update and view order status, and customers to view order status.

**Derivations:** USREQ-3 / REQ-12 / REQ-2

**UC-11: Add/Remove employees from system**-Allow manager to add employee to system when hiring and remove employees from system when firing.

**Derivations:** REQ-11

**UC-12: Meal Prep**-Allows chefs to see incoming orders with a timestamp and notify employees when their order is ready.

**Derivations:** REQ-3 / REQ-7

**UC-13: Create a Rewards Account**-Allows customers to create a rewards account.

**Derivations:** NFREQ-12 / USREQ-4

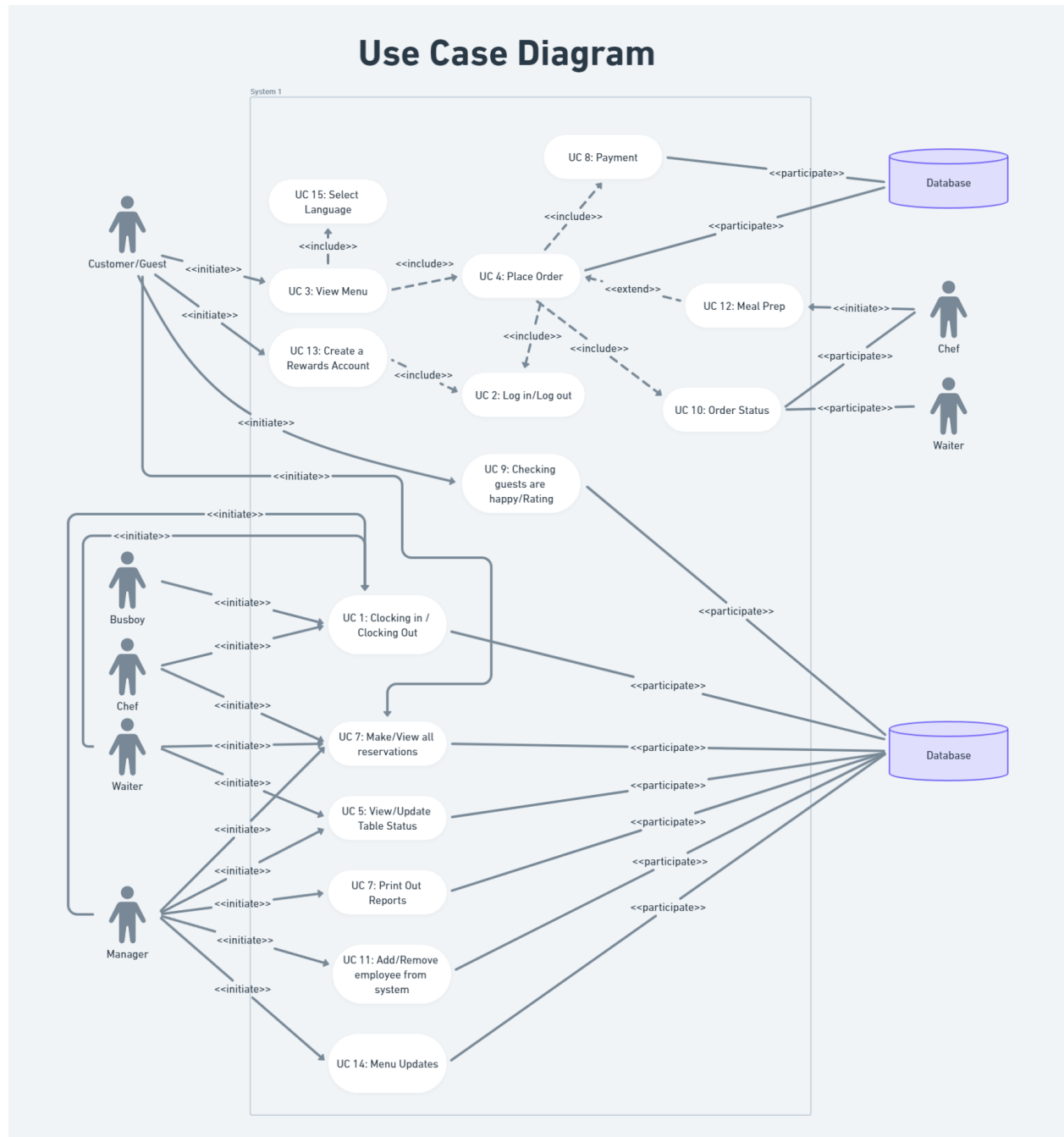
**UC-14: Menu Updates**-Allow manager to add or remove items from menu.

**Derivations:** REQ-5

**UC-15: Select Language**-Allow customers to select a different language.

**Derivations:** NFREQ-8

- ii. Use Case Diagram





## iii. Traceability Matrix

REQ't	PW	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
REQ-1	5				X											
REQ-2	5										X					
REQ-3	4				X								X			
REQ-4	4							X								
REQ-5	5														X	
REQ-6	4	X														
REQ-7	3												X			
REQ-8	4								X							
REQ-9	3															
REQ-10	5		X													
REQ-11	2											X				
REQ-12	3										X					
NFREQ-1	5															
NFREQ-2	5															
NFREQ-3	4															
NFREQ-4	4															
NFREQ-5	3															
NFREQ-6	3									X						
NFREQ-7	4															
NFREQ-8	3															X
NFREQ-9	3									X						
NFREQ-10	3			X												
NFREQ-11	3						X									
NFREQ-12	2													X		
NFREQ-13	3				X											
USREQ-1	5			X	X											
USREQ-2	4					X										
USREQ-3	3										X					
USREQ-4	3													X		

iv. Fully-Dressed Description

UC-3: View Menu
<b>Related Requirements:</b> USREQ-1, NFREQ-10
<b>Initiating Actor:</b> Employee (for assisting customers or placing orders), Customer (for placing orders or inquiries)
<b>Actor's Goal:</b> To view detailed information about menu items, including images, descriptions, prices, and ingredients
<b>Participating Actors:</b> Database, Interactive Menu Display
<b>Preconditions:</b> The restaurant's menu is updated and available in the system. The system is operational and accessible by both employees and customers.
<b>Postconditions:</b> The employee or customer has successfully viewed the desired menu items and their details.
<b>Flow of Main Success Scenario:</b> <ol style="list-style-type: none"> <li>← The employee or customer accesses the system to view the menu.</li> <li>← The database retrieves the menu items, including images, descriptions, prices, and ingredients.</li> <li>← The Interactive Menu Display presents the menu items in an organized and visually appealing manner with suggestions.</li> <li>← The employee or customer can select individual items to view more detailed information, including ingredients.</li> <li>← The employee or customer can navigate through different sections of the menu with ease.</li> </ol>
<b>Flow of Events for Alternate Success Scenario:</b> <ol style="list-style-type: none"> <li>← The employee or customer accesses the system to view the menu.</li> <li>← The database fails to retrieve the menu items due to a system error.</li> <li>← The employee or customer is informed of the system error and is asked to try again later.</li> <li>← The menu remains inaccessible until the error is resolved.</li> </ol>

UC-4: Place Order
<b>Related Requirements:</b> USREQ-1, REQ-3, NFREQ-13, REQ-1
<b>Initiating Actor:</b> Employee (for assisting customers or placing special orders), Customer (for placing orders).
<b>Actor's Goal:</b> To efficiently place an order, customize meals by adding or removing items, and ensure the kitchen is notified of the new order.
<b>Participating Actors:</b> Database, Interactive Menu Display, Kitchen Queue System.
<b>Preconditions:</b> The restaurant's menu is updated and available in the system. The system is operational and accessible by both employees and customers. The kitchen is operational and ready to receive new orders.
<b>Postconditions:</b> The order has been successfully placed and added to the kitchen queue. The customer receives a confirmation of their order.
<b>Flow of Main Success Scenario:</b> <ol style="list-style-type: none"> <li>← The employee or customer accesses the Interactive Menu Display to view the menu.</li> <li>← They select desired items, customizing them by adding or removing ingredients as needed.</li> <li>← Once the order is finalized, they confirm the order.</li> <li>← The system saves the order in the database and notifies the kitchen via the Kitchen Queue System.</li> <li>← The chef receives the new order and begins preparation.</li> <li>← The customer receives a confirmation, including order details and an estimated wait time.</li> </ol>
<b>Flow of Events for Alternate Success Scenario:</b> <ol style="list-style-type: none"> <li>← The employee or customer accesses the Interactive Menu Display to view the menu.</li> <li>← They select desired items but encounter a system error when trying to customize or confirm the order.</li> <li>← The employee or customer is informed of the system error and is asked to try again.</li> <li>← If the error persists, they may need to place the order manually or seek assistance.</li> </ol>

UC-10: Order Status
<b>Related Requirements:</b> USREQ-3, REQ-12, REQ-2
<b>Initiating Actor:</b> Employee (for updating order status), Customer (for viewing order status)
<b>Actor's Goal:</b> To keep the order process transparent and informed for both employees and customers.
<b>Participating Actors:</b> Database, Interactive Display Screen.
<b>Preconditions:</b> An order has been placed by the customer. The system is operational and accessible by both employees and customers.
<b>Postconditions:</b> The order status is updated and visible to relevant parties.
<b>Flow of Main Success Scenario:</b> <ol style="list-style-type: none"> <li>← The employee or customer accesses the system to view or update the order status.</li> <li>← The database retrieves the current status of the order.</li> <li>← If accessed by an employee, they have the option to update the order status (e.g., "preparing", "ready for pickup", "served").</li> <li>← The updated status is saved in the database.</li> <li>← The Interactive Display Screen shows the updated status to the customer.</li> </ol>
<b>Flow of Events for Alternate Success Scenario:</b> <ol style="list-style-type: none"> <li>← The employee or customer accesses the system to view the order status.</li> <li>← The database fails to retrieve the current status due to a system error.</li> <li>← The employee or customer is informed of the system error and is asked to try again later.</li> <li>← The order status remains unchanged until the error is resolved.</li> </ol>

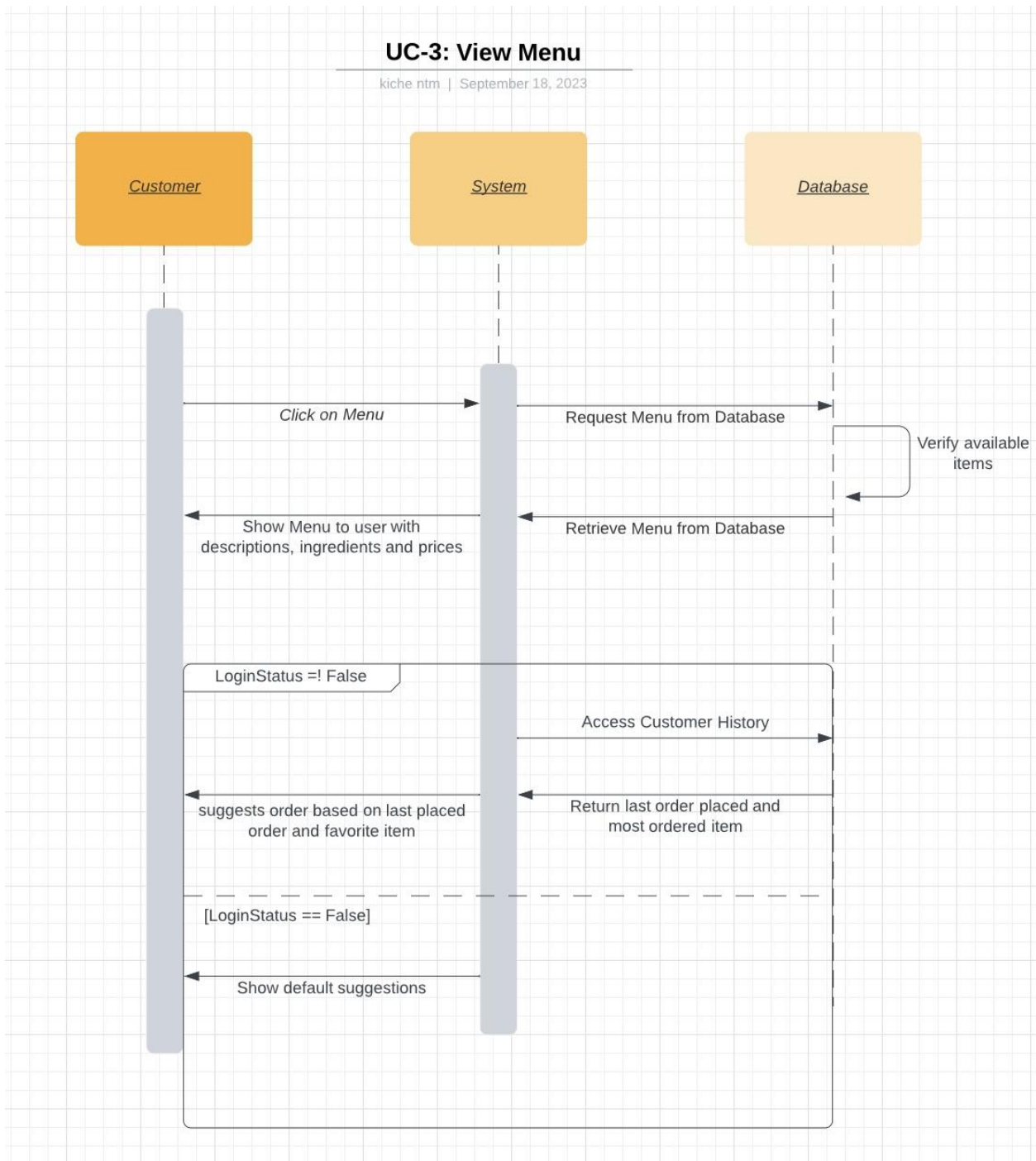
UC-12: Meal Prep
<b>Related Requirements:</b> REQ-3 / REQ-7
<b>Initiating Actor:</b>

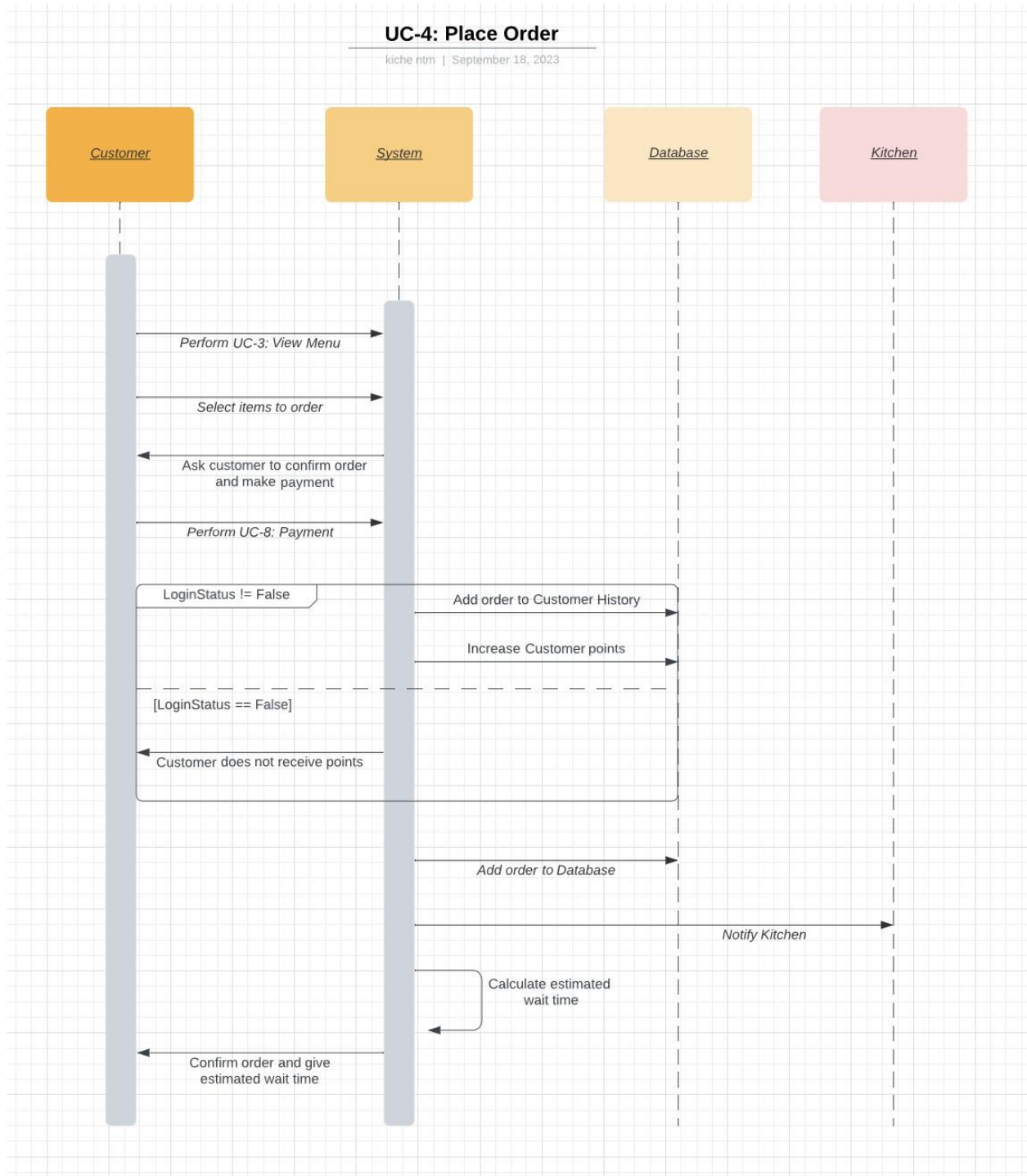
Chef
<b>Actor's Goal:</b> Efficiently prepare meals in accordance with customer order, and notify the waitstaff when dishes are ready for serving.
<b>Participating Actors:</b> Kitchen Queue System, Notification System, Waitstaff, .
<b>Preconditions:</b> The restaurant must be open and actively serving customers. Necessary ingredients and equipment should be available in the kitchen. The Kitchen Queue System is functional and displays incoming orders. The Notification System is set up to alert waitstaff.
<b>Postconditions:</b> Meals are prepared and ready to be served. Order status is updated to "Prepared" in the order management system and the waitstaff has been notified that the dish is ready for serving.
<b>Flow of Main Success Scenario:</b> <ol style="list-style-type: none"> <li>← The chef accesses the Kitchen Queue System to view incoming orders.</li> <li>← Orders are displayed with a timestamp indicating when they were placed.</li> <li>← The chef begins preparing the dishes based on the order of arrival and priority.</li> <li>← Once a dish is ready, the chef uses the Notification System to alert the relevant waitstaff.</li> <li>← The waitstaff receives the notification and proceeds to serve the dish to the customer.</li> </ol>
<b>Flow of Events for Alternate Success Scenario:</b> <ol style="list-style-type: none"> <li>← The chef accesses the Kitchen Queue System to view incoming orders.</li> <li>← The system fails to display new orders due to a technical glitch.</li> <li>← The chef informs the management about the system error.</li> <li>← Until the system is restored, manual communication may be required between the waitstaff and the chef to manage orders.</li> </ol>

<b>UC-13: Create a Rewards Account</b>
<b>Related Requirements:</b> NFREQ-12, USREQ-4
<b>Initiating Actor:</b> Customer

<b>Actor's Goal:</b> To avail discounts based off their spending at the restaurant
<b>Participating Actors:</b> Database
<b>Preconditions:</b> The user gets into and loads the system The user plans to complete a purchase
<b>Postconditions:</b> The user has earned an appropriate amount of points based on how much they spent
<b>Flow of Main Success Scenario:</b> <ol style="list-style-type: none"> <li>← At the beginning, the system prompts the customer to either create an account or log in.</li> <li>← The database will retrieve the customer's information with their current point balance.</li> <li>← The customer selects their desired food items and proceeds to confirm the order.</li> <li>← The customer enters payment.</li> <li>← The database is updated with the customer's points based on their transaction.</li> </ol>
<b>Flow of Events for Alternate Success Scenario:</b> <ol style="list-style-type: none"> <li>← At the beginning, the system prompts the customer to either create an account or log in.</li> <li>← The database will retrieve the customer's information with their current point balance.</li> <li>← The customer selects their desired food items and proceeds to confirm the order.</li> <li>← The customer does not complete or enters payment.</li> <li>← The database indicates that the transaction is incomplete and points have not been added.</li> </ol>

v. System Sequence Diagrams

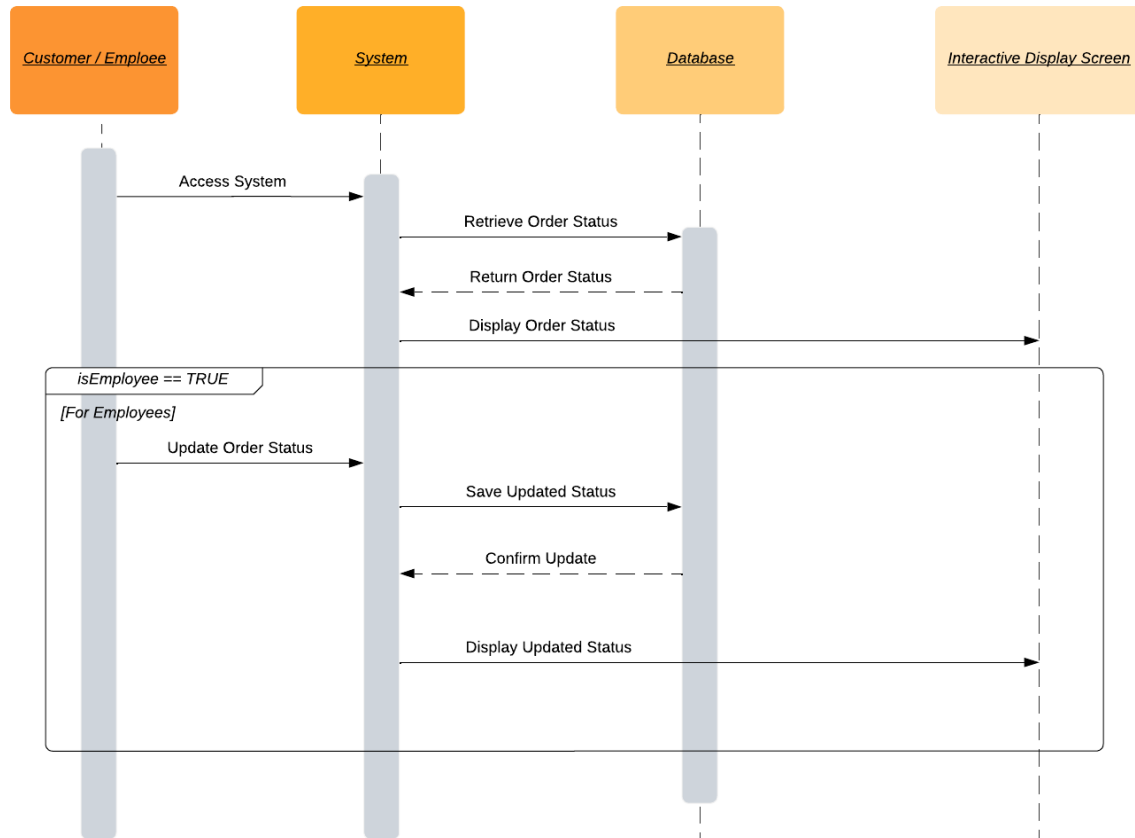


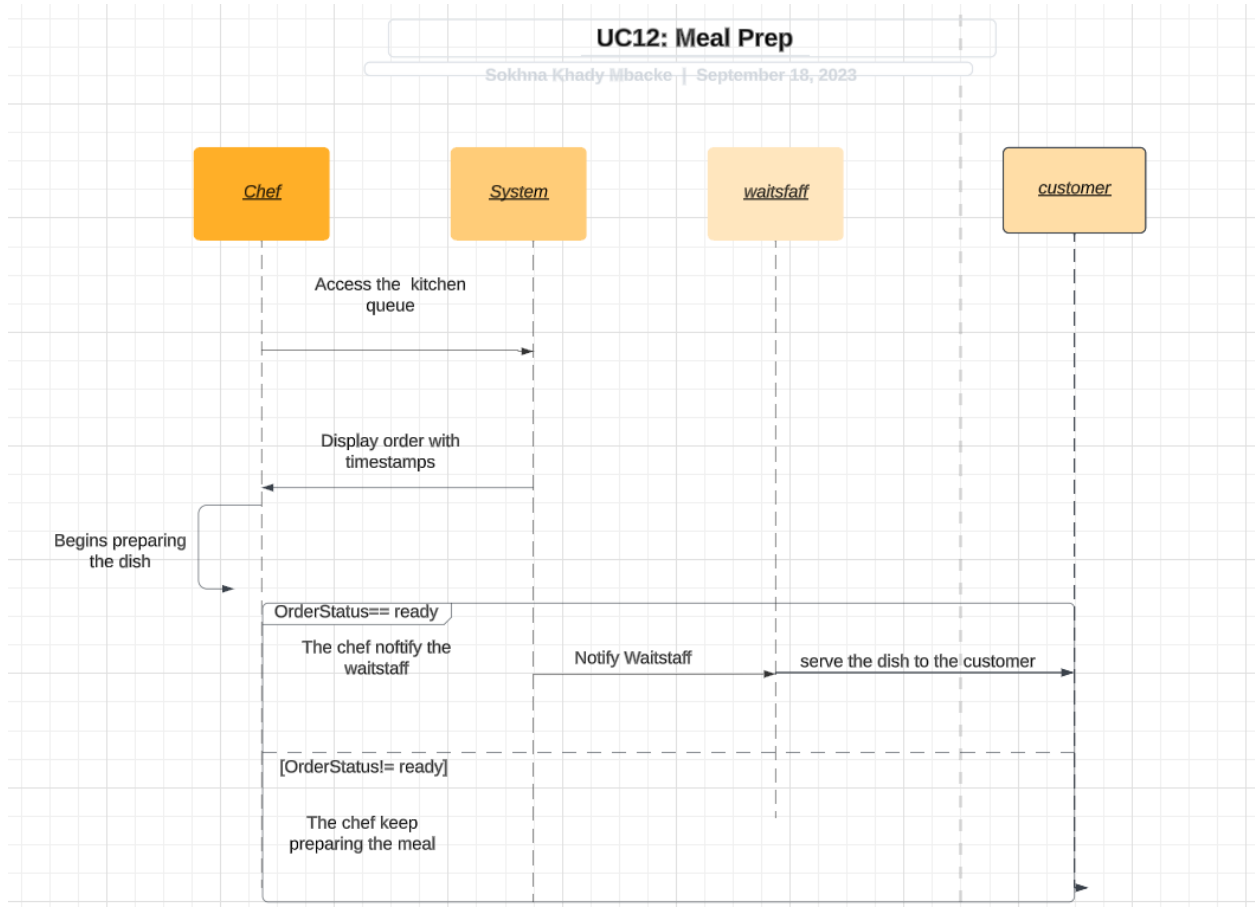


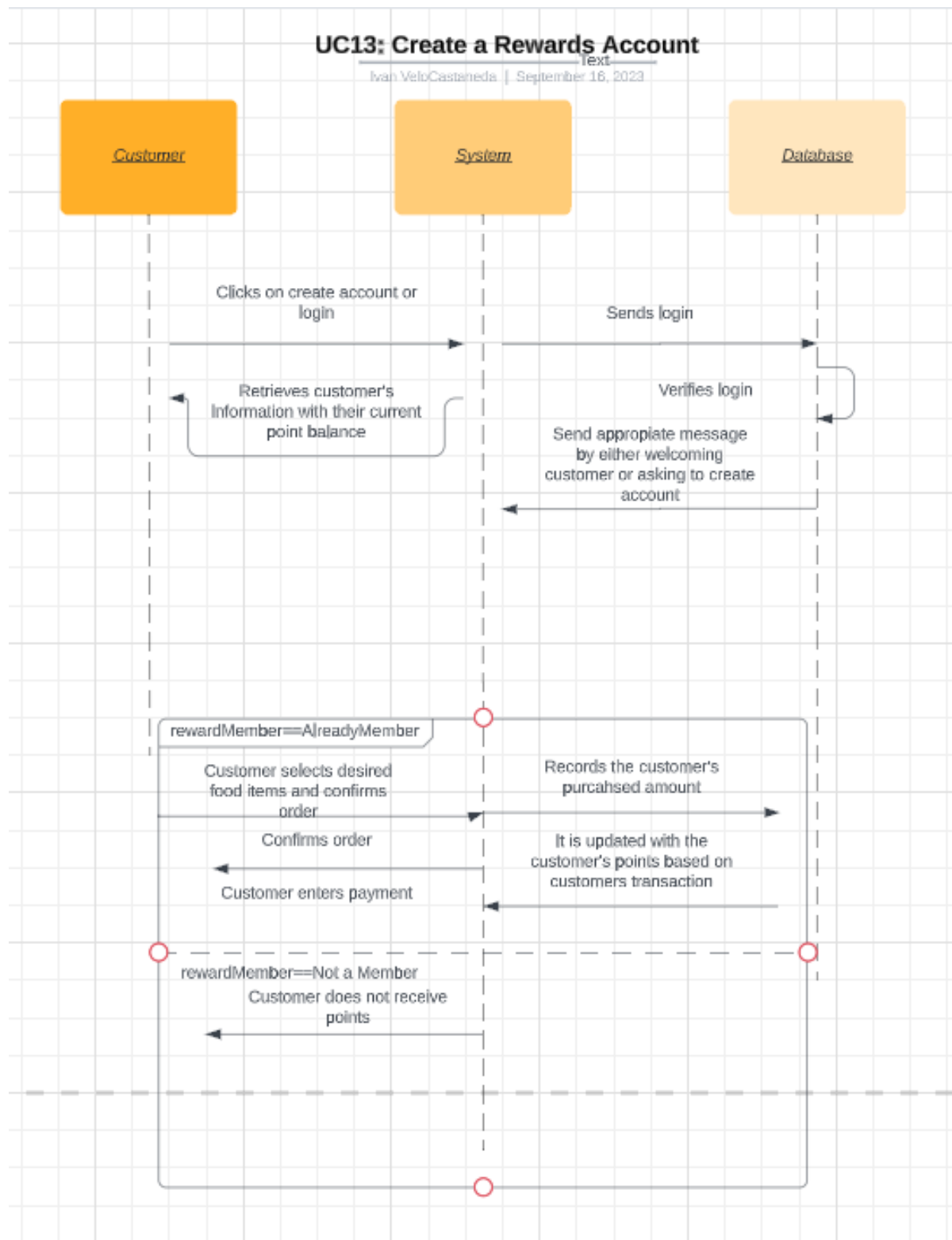


### UC-10: Order Status

Bjarni Jonsson | September 18, 2023







## 4. User Interface Specification

### 1. Preliminary Design

#### 1. View Menu User Interface Specification

- a. After successfully signing in to their role-specific interface, managers and servers will be able to navigate through the restaurant's menu using intuitive categories which will be available to them by clicking on the "View Menu" button. Each menu item will be presented with essential details such as name, description, price, and a detailed list of ingredients.

#### 2. Place Order User Interface Specification

- a. Within their role specific interface, managers and servers will have the capability to manage customers orders and navigate the restaurant's table layout. When the user clicks on the "Place Order" directs users to the table layout view, which presents an overview of all the tables within the restaurant. Users can interact with this layout by selecting tables to take customer orders.
- b. Furthermore, users can efficiently manage their orders by clicking on the "View Tables" button, which displays a list of current tables and their associated details.

#### 3. Order Status User Interface Specification

- a. Within their role specific interface, managers, servers and customers will have the capability to see the status of their existing orders in the restaurant. When the user clicks on the "Order Status" button, they will be able to update or check the status of orders in real-time. This will allow wait staff and customers to stay informed about order progress and effectively manage customer expectations throughout their dining experience. It is important to note that customers will not have the capability to update orders. They will only be able to see their order progress.

#### 4. Meal Prep User Interface Specification

- a. Within their role specific interface, chefs will be able to see incoming orders sent by servers. The Chef's specific interface

will enable chefs to view detailed order information and effectively communicate with servers when an order is prepared and ready for delivery to the respective table. When they click on the “Orders” button, they will see detailed information about each order, including meal items, special requests, and table number.

- b. Once a table’s order is ready, chefs will have the ability to let a server know when their food is ready by clicking on the “Ready” button. This action notifies the respective server that the order is prepared and ready to be served to the designated table.

#### **5. Create a Rewards Account User Interface Specification**

- a. Creating a rewards account will allow customers to earn points through purchases which will earn them points for recent transactions. They will have access to see a detailed transaction history detailing earned and redeemed for transparency.
- b. After creating a rewards account and log in using their credentials, a dashboard displaying the user’s current points balance and summary of their rewards program status. There will be a clear visualization of earned, redeemed, and expiring points.
- c. Customers will also be able to manipulate their user profile settings which can be used to update their contact information, change their password, etc. Such a program aims to empower customers and foster their customer engagement and loyalty within the restaurant’s rewards program.

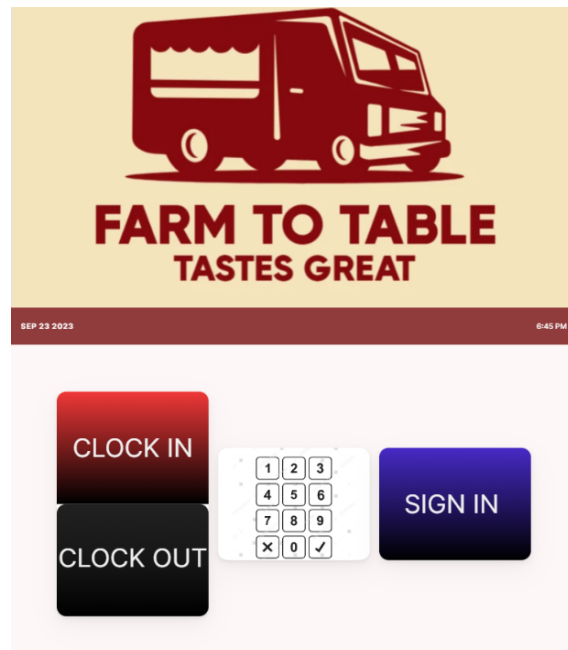
#### **6. Reports User Interface Specification**

- a. A sales report that provides a summary of daily, weekly, or yearly sales.
- b. An inventory report that tracks the levels of ingredients in the kitchen and supplies.
- c. A labor cost report that analyzes salaries and wages. It will also compare labor costs to sales to determine labor cost percentages. Lastly, it would aid to optimize staffing levels to maintain efficiency and control costs.

- d. A food cost report that calculates the cost of ingredients and compares food costs to revenue to calculate food cost percentage. It will help adjust menu pricing and control food expenses.
- e. A profit and loss report that provides an overview of revenues, costs, and expenses. It will help analyze net profit or loss over a specific period.
- f. A customer feedback report that summarizes reviews, comments, and ratings. It will help identify areas of improvement based on customer feedback and enhance customer satisfaction and the refinement of services.
- g. A payment and transactions report that will track payment methods (cash, credit card, rewards payments) used by customers. It will provide an insight into popular payment methods.

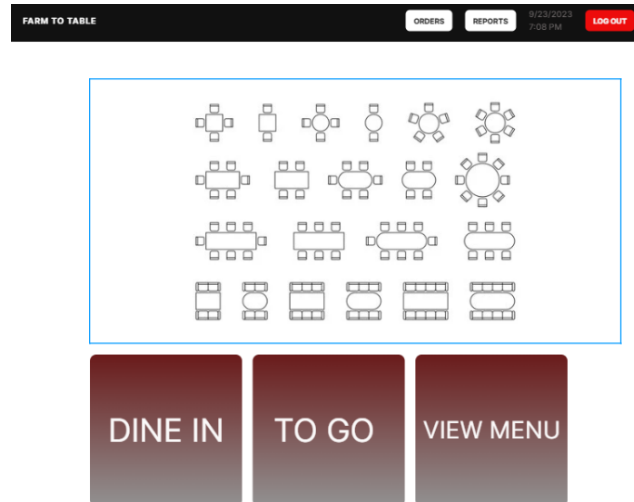
## 2. User Effort Estimation

*Figure 1: Potential Home Page of the System*

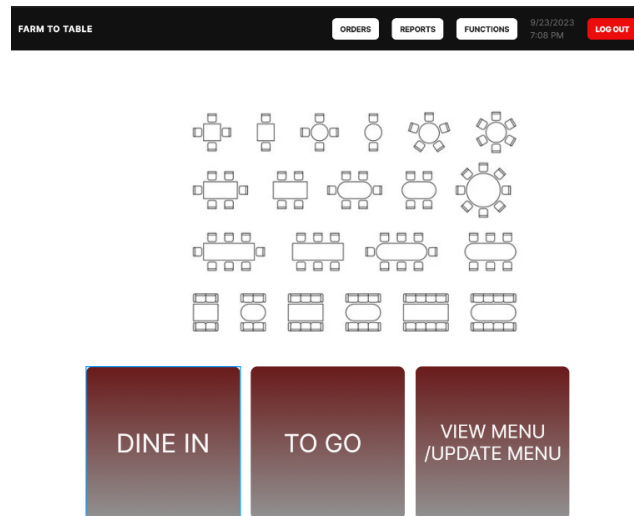


*Figure 2: Potential Design for Servers Interface*

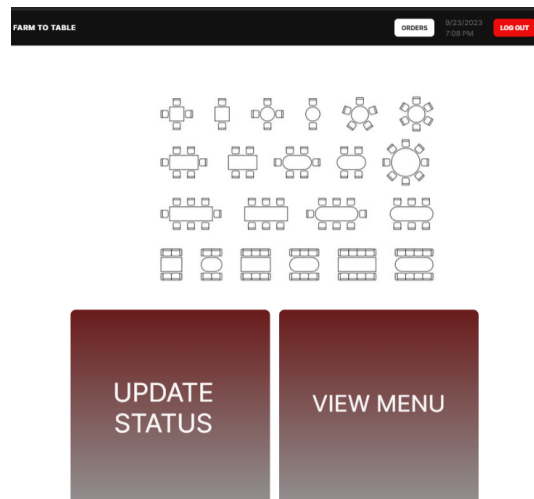
## Farm to Table



*Figure 3: Potential Design for Managers Interface*



*Figure 4: Potential Design for Host/Hostess Interface*



1. **Clock in/Clock out:** The following sequence of actions would allow an employee with the code “6254” to either clock in or clock out with a total of 2 mouse clicks and 4 keystrokes.

**NAVIGATION:** Total 2 mouse clicks, as follows

- a. Click on the “Clock in” or “Clock out” button.
- b. Click on the “Confirm” button.

**DATA ENTRY:** Total of 4 keystrokes, as follows

- a. Press “6”
- b. Press “2”
- c. Press “5”
- d. Press “4”

2. **Log in:** The following sequence of actions would allow an employee with the code “6254” to log in into the system to place an order with a total of 1 mouse click and 4 keystrokes

**NAVIGATION:** Total 1 mouse click, as follows

- a. Click on the “Sign in” button.

**DATA ENTRY:** Total of 4 keystrokes, as follows

- a. Press “6”
- b. Press “2”
- c. Press “5”
- d. Press “4”

3. **Log out:** The following sequence of actions would allow an employee to log out of the system for another employee to use with a total of 1 mouse click and 0 keystrokes.

**NAVIGATION:** Total 1 mouse click, as follows

- a. Click on the “Sign out” button.

**DATA ENTRY:** None

4. **View Menu:** The following sequence of actions would allow an employee to view the menu and item ingredients with a total of mouse 1 click and 0 keystrokes.

**NAVIGATION:** Total 1 mouse click, as follows

- a. Click on the “View Menu” button.

**DATA ENTRY:** None



5. **Place Order:** The following sequence of actions would allow an employee with the code “6254” to place an order for table 1 with a total of 8 mouse clicks and 4 keystrokes. The following order is as follows:

2 coffees and 2 classic burgers with french fries.

**NAVIGATION:** Total 8 mouse clicks, as follows

- a. Click on the “Sign in” button.
- b. Click on the “Dine in” button or “To go” button.
- c. Click on the “Table 1” icon.
- d. Click on the “Coffee” button twice under the “Beverages” section.
- e. Click on the “Classic Burger” button twice under the “Burgers/Sandwiches” section.
- f. Click on the “French Fries” button twice under the “Sides” section.
- g. Click on the “Confirm” button.
- h. Click on the “Send Order” button.

**DATA ENTRY:** Total of 4 keystrokes, as follows

- i. Press “6”
- j. Press “2”
- k. Press “5”
- l. Press “4”

If the user is already signed in, then the data entry portion of this scenario would not be necessary.

6. **View and Update Table Status:** The following sequence of actions would allow an employee with the code “6254” to view and update the status of a table with a total of 6 mouse clicks and 4 keystrokes. In this case, we want to update the status of table 1 from “dirty” to “unoccupied”.

**NAVIGATION:** Total 6 mouse clicks, as follows

- a. Click on the “Sign in” button.
- b. Click on the “View Tables button.
- c. Click on the “Table 1” icon.
- d. Click on the “Update Status” button.
- e. Select “Unoccupied” from the drop down menu.

f. Click “Confirm”.

**DATA ENTRY:** Total of 4 keystrokes, as follows

g. Press “6”

h. Press “2”

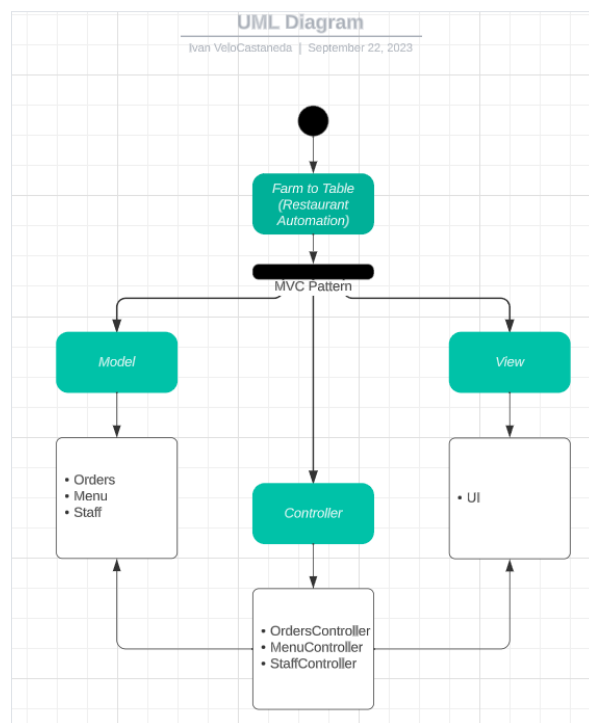
i. Press “5”

j. Press “4”

**Part 3:**

## 5. System Architecture

### a. Identifying Subsystems



The subsystem above displays our three layer architecture which consists of Model, View, and Controller. Each layer has a specific responsibility to the entire system. The Model consists of the data business logic of the system, including orders, menus, and staff. The View contains the user interface responsible for displaying the menu, taking orders, and managing employees. It is also responsible for user interactions. Lastly, the Controller acts as an intermediary between the model and view, handling user input, updating the model, and ensuring the view reflects the changes accordingly. There are specific controllers for different entities like orders, menus, and staff, managing the flow of data and actions.

### b. Architecture Styles

For our website, we will use the Model-view-controller pattern, MVC pattern, a widely adopted architectural approach for web development. Since some team members are already familiar with this pattern, it is a well grounded choice. Given that a restaurant

system involves a diverse functionalities such as order processing, menu management, and staff operations, the MVC pattern will provide an organized framework to handle these tasks efficiently. It will allow for flexibility and scalability, enabling different members to work on separate components simultaneously and accommodate changes in the restaurant's operations. Restaurants often have different user interfaces for customers, staff, and management. It will also allow the development of multiple views tailored to each user group, providing a more personalized and intuitive user experience. Lastly, it will allow management to modify or update specific components without impacting the entire system as they may need to make changes to menu offerings, or operational processes, leading to cost-effective maintenance and upgrades.

In addition to MVC, our website will adopt the Client-Server model, an architecture where the server centrally manages and provides data to various client interfaces. The system will include different interfaces for customers, servers, chefs, and managers. In our case, these interfaces will cater to customers, servers, chefs, and managers, each offering distinct functionalities such as ordering, making reservations, and menu management. The one-to-many relationship in the Client-Server model will ensure efficient data handling and interaction.

Furthermore, our website will integrate the Peer-to-Peer model to streamline customer payments. This model will enable separate processing of payments, facilitating secure fund transfers from the customer's bank account to the restaurant's account. This approach will optimize the payment process, enhancing the overall transaction experience.

### c. Mapping Subsystems to Hardware

Yes, our system needs to run on multiple computers. Here's the breakdown:

1. **Server Subsystem (Node.js with Express and MySQL):** This will be hosted on a dedicated machine. This subsystem is responsible for handling all the business logic, database operations, and serving API endpoints. Given the need for fast response times and the potential for high concurrent requests, especially during peak restaurant hours, it's crucial to have this on a robust server machine.
2. **Database Subsystem (MySQL):** The database will be hosted on a separate dedicated machine. This separation ensures that the database operations do not bottleneck the server's performance, and it also provides an added layer of security. By isolating the database, we can implement stricter access controls and monitoring.
3. **Client Subsystem (Web Browser):** This will run on various devices, including desktops, laptops, tablets, and smartphones. The client subsystem is responsible for rendering the user interface and interacting with the server subsystem via API calls. It will cater to different user interfaces for customers, staff, and management.

### d. Connectors and Network Protocols

Given that our system runs on multiple machines, we will primarily use the following communication protocols:

1. **HTTP/HTTPS:** This will be the primary protocol for communication between the client (web browser) and the server (Node.js with Express). We chose HTTP/HTTPS because it's a standard protocol for web applications, ensuring compatibility across various devices and browsers. HTTPS will be used to encrypt the data during transmission, providing an added layer of security, especially for sensitive operations like payments and user authentication.

2. **MySQL Protocol (via JDBC in Node.js):** For communication between the server subsystem and the database subsystem, we will use the MySQL protocol through JDBC connectors in Node.js. This choice is due to our use of MySQL as the database system. The JDBC connector provides a standardized way to connect Node.js applications to MySQL databases, ensuring efficient and secure data operations.
  3. **Peer-to-Peer Protocols:** For the Peer-to-Peer model used in customer payments, specific protocols will be adopted depending on the payment gateway or service we integrate. This ensures secure and direct fund transfers between customer accounts and the restaurant's account.
- e. Global Control Flow
- **Execution orderliness:** The system follows an event-driven fashion. It does not follow a strict linear procedure where every user has to go through the same steps. Instead, it waits for events triggered by user interactions. Users can generate actions in different orders based on their needs and preferences.
  - **Time dependency:** There are no specific timers in the system, and it is not a real-time system with strict time constraints. It responds to events as they occur, without periodic time requirements.
- f. Hardware Requirements

The following hardware requirements are essential to ensure the reliable and efficient operation of our system.

- **Screen display:** The client subsystem (web browsers) requires screens with varying resolutions, as it runs on devices such as desktops, laptops, tablets, and smartphones. Specific display requirements may vary based on device types and screen sizes.
- **Disk storage:** Both the server subsystem and the database subsystem will require adequate disk storage. The requirements will depend on the volume of data and system usage.
- **Communication network:** the system relies on a stable communication network with a minimum network bandwidth of 56 Kbps for smooth data exchange between clients and the server, especially during peak restaurant hours.
- **Database server:** The dedicated machine hosting the MySQL database subsystem should have sufficient computing resources to handle database operations efficiently.
- **Security Measures:** Adequate security measures, such as firewalls, intrusion detection systems, and encryption protocols should be in place to safeguard data integrity and protect against threats.

## 6. Project size estimation based on use case points

Identifier	Number of actors	Number of requirements	Complexity	Weight
------------	------------------	------------------------	------------	--------

## Farm to Table

UC-1: Clocking in /Clocking out	2: Employees and database	1: REQ-6	Simple	1
UC-2: Log in/Log out	3: Employees, customers and database	1: REQ-10	Average	2
UC-3: View Menu	3: Employees, customers and database	2: NFREQ-10 and USREQ-1	Complex	3
UC-4: Place Order	3: Employees, customers and database	4: REQ-1, REQ-3, NFREQ-13, USREQ-1	Complex	3
UC-5: View/Update Table Status	3: Employees and customers	1: USREQ-2	Average	2
UC-6: Make/View all reservations	3: Employees, customers and database	1: NFREQ-11	Average	2
UC-7: Print Out Reports	2: Manager and database	1: REQ-4	Simple	1
UC-8: Payment	3: Employees, customers and database	1: REQ-8	Average	2
UC-9: Checking guests are happy/Rating	2: Customers and database	2: NFREQ-6 and NFREQ-9	Average	2
UC-10: Order status	2: Customers and employees	3: REQ-2, REQ12, and USREQ-3	Complex	3
UC-11: Add/Remove employees from the system	2: Manager and database	1: REQ-12	Simple	1
UC-12: Meal Prep	2: Chefs and employees	2: REQ-3 and REQ-7	Average	2
UC-13: Create a Rewards Account	2: Customers and database	2: NFREQ-12 and USREQ-4	Average	2
UC-14: Menu Updates	2: Manager and database	1: REQ-5	Simple	1

UC-15: Select Language	2: Customers and database	1:NREQ-9	Simple	1
------------------------	---------------------------	----------	--------	---

$$5 * \text{Simple} + 7 * \text{Average} + 3 * \text{Complex} = 28$$

## 7. Analysis and Domain Modeling

### a. Conceptual Model

Show the process of deriving the domain model and then draw the diagram. Provide text description of:

#### i. Concept definitions

Responsibility	Type	Concept
<b>R1:</b>	Facilitate communication among customer, chef, server, and busboy	Communicator
<b>R2:</b>	Show choices for customer, waiter, chef, and manager	Interface
<b>R3:</b>	Guide the customer in choosing a table	Table Status
<b>R4:</b>	Block selection of unavailable tables	Table Status
<b>R5:</b>	Ensure reserved tables are not chosen before the reserved time	Table Status
<b>R6:</b>	Update table status upon customer actions (selection, departure, cleaning by busboy)	Table Status
<b>R7:</b>	Design and layout of the restaurant's seating and table arrangement	Floorplan
<b>R8:</b>	Update floorplan based on changes in table arrangements or renovations	Floorplan
<b>R9:</b>	Display a visual representation of the restaurant's layout to staff and customers	Floorplan
<b>R10:</b>	Ensure efficient flow and movement within the restaurant based on the floorplan	Floorplan

<b>R11:</b>	Line up orders for chef's preparation	Order Queue
<b>R12:</b>	Show orders that are ready to be served	Serving
<b>R13:</b>	Indicate when an order is received, in preparation, ready, or delivered	Order Status
<b>R14:</b>	Notify the customer of any changes in their order status	Order Status
<b>R15:</b>	Allow staff to update the status of an order	Order Status
<b>R16:</b>	Provide a history of all order statuses for a particular order	Order Status
<b>R17:</b>	Oversee transaction processing	Payment System
<b>R18:</b>	Manage the flow of data between the user interface and the system	Controller
<b>R19:</b>	Ensure proper routing of user requests to appropriate system components	Controller
<b>R20:</b>	Validate user inputs before processing	Controller
<b>R21:</b>	Handle errors and provide appropriate feedback to users	Controller
<b>R22:</b>	Oversee database interactions	DB Connection
<b>R23:</b>	Maintain employee login details and work hours	Employee Account
<b>R24:</b>	Save customer login details	Customer Profile
<b>R25:</b>	Keep track of customer's reward points based on past orders	Customer Profile
<b>R26:</b>	Showcase favorites, top picks, and frequently ordered items in customer/user profile	Customer Profile
<b>R27:</b>	Record the customer's order in the database	Customer Profile
<b>R28:</b>	Manage and allocate reward points to customers based on their purchases	Reward System

<b>R29:</b>	Offer special promotions or discounts to customers through the reward system	Reward System
<b>R30:</b>	Allow customers to redeem their reward points for meals or discounts	Reward System
<b>R31:</b>	Provide a dashboard for customers to view and manage their reward points	Reward System

ii. Association definitions

Concept Pair	Association Description	Association Name
Customer Account $\Leftrightarrow$ DB Connection	Retrieve customer's data from the database	DBQuery
Customer Account $\Leftrightarrow$ Interface	Display customer's options	Display
Interface $\Leftrightarrow$ Controller	Allows users to interact with the system	User Action
Communicator $\Leftrightarrow$ DB Connection	Modify or insert data into the database	DBUpdate
Communicator $\Leftrightarrow$ DB Queue	Allows user to send orders and queue to the database	DBQuery
Controller $\Leftrightarrow$ Order Status	Allow the staff to update or view the status of their food orders	Update Order Status View Order Status
Controller $\Leftrightarrow$ Table Status	Allows users to view the status of a table	View Table Status
Controller $\Leftrightarrow$ Payment System	Allows users to complete payments	Make Payment
Payment System $\Leftrightarrow$ DB Connection	Stores payments records of past transactions in the database	Record Payment
Interface $\Leftrightarrow$ DB Connection	Retrieve the data from the database for the users	DBQuery
Customer Profile $\Leftrightarrow$ DB Connection	Store earned points in the database	Reward System



Table Status ⇔ Interface	Displays the table layout with the proper status of the tables	Display
Employee Account ⇔ Interface	Displays the employee's actions	Display
Payment System ⇔ Customer Account	Update the points balance for the users	Update Rewards
Floorplan ⇔ Controller	Allow the manager to adjust the table layout of the restaurant	Floorplan Layout

## iii. Attribute Definitions

Concept	Attribute	Description
Customer Account	userEmail userPassword	Associated email and password of the customer. Guest account is assigned if there is no account.
Interface	sendOrder Receipt rateFood tableStatus	Allow the user to send orders after choosing food items. Allow the user to choose which way they would like to receive their receipt(email, paper) Allow the user to rate their meal and then have that rating stored and displayed Provides the user with the up-to-date status of each table in the restaurant. 2 possible states, "occupied" and "empty"
Payment System	paymentMade	Updates the system on whether or not a payment has been made
Order Status	orderStatus orderReady createdAt updatedAt	Allows the chef to update the status of orders being cooked. Allows the chef to send a signal to the waiter/waitress and customers that the order is finished. Allows the chef to see what time the order was sent. Allows the chef to see what

		time the order was updated.
Order Queue	chefQueue	Keeps track of incoming food orders in the order that they were submitted for the chef to follow.
Controller	tableList tableConfirm	Shows the employees the current tables available. Table is marked in green once it is picked by an employee and it turns back to white when customers leave.
Communicator	customerOrder	Each customer order is stored in the database.
Floorplan	viewPlan tableAdjust	Displays the current layout of the restaurant. Allows the manager to adjust the layout by either adding tables or deleting tables.
Employee Account	hoursWorked timeClockedIn timeClockedOut role	Display the employee's worked hours in a particular payment cycle. Display the employee's id. Display the time at which the employee clocked in on a particular day. Display the time at which the employee clocked out on a particular day. Display the role of the employee.
Table Status	tableStatus	Display if the table is empty(white) or occupied (green).
Reward System	userBalance redeemPoints	Display the current reward points balance of the user. Allow the user to redeem points.
DB Connection	dbAuthenticationCreds userID	Store the DB login for authorized personnel. Display the employee's id.

iv. Traceability matrix — show how your use cases map to your domain concepts.

Use Case	PW	Domain Concepts											
		Customer Account	Interface	Payment System	Order Status	Order Queue	Controller	Communicator	Floor plan	Employee Account	Table Status	Reward System	DB Connection
UC-1			X							X			X
UC-2		X	X							X			X
UC-3			X										X
UC-4		X	X		X	X		X				X	X
UC-5			X		X	X	X		X		X		
UC-6		X	X								X		
UC-7										X			X
UC-8		X		X								X	X
UC-9		X											X
UC-10					X	X							
UC-11										X			X
UC-12			X		X	X							X
UC-13		X								X			X
UC-14			X							X			X
UC-15			X										X

b. System Operation Contracts

Operation	View Menu
Use Case:	UC-3
Preconditions:	<ul style="list-style-type: none"> <li>The user logs in to their account.</li> <li>Manager and server select the “View Menu” button on the bottom right</li> </ul>
Postconditions:	The user is taken to the menu interface

Operation	Place Order
Use Case:	UC-4
Preconditions:	<ul style="list-style-type: none"> <li>The user logs into their account</li> <li>Manager and server select the “Dine in” button or “To Go” button</li> <li>Manager and server select the table number and enter the number of guest at the table</li> </ul>
Postconditions:	The user is taken to the place order interface

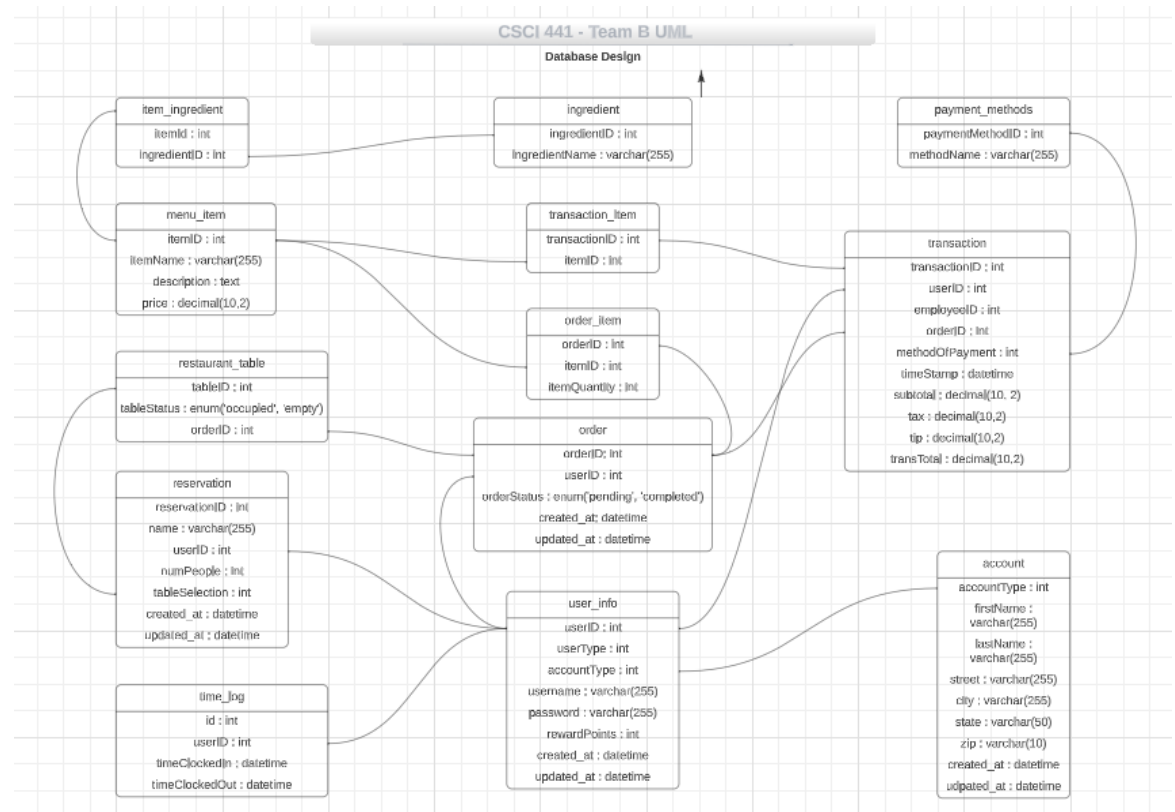
Operation	Order Status
Use Case:	UC-10
Preconditions:	<ul style="list-style-type: none"> <li>The user logs into their account</li> <li>Manager, customer and chef select the “Order Status” button on the bottom left</li> </ul>
Postconditions:	The user is taken to the order status interface where their order is displayed

Operation	Meal Prep
Use Case:	UC-12
Preconditions:	<ul style="list-style-type: none"> <li>The user logs into the system</li> <li>The chef selects the “View</li> </ul>

	Orders” button on the bottom left
Postconditions:	The user is taken to the chef interface where all the orders are displayed along with their status

Operation	Create Rewards Account
Use Case:	UC-13
Preconditions:	<ul style="list-style-type: none"> <li>The user logs into their rewards account</li> <li>The user makes a purchase</li> </ul>
Postconditions:	The user has earned rewards points based on how much they spent

### c. Data Model and Persistent Data Storage



Our system will not require persistent data storage beyond a single system execution.

Initially, the user's data will be stored in designated tables based on their role. For employees, the system will retain essential information such as first and last names, a

unique 4-digit identifier (last four digits of their social security number), salary, hours worked, work schedule, turnover time, sales, credit card tips, cash tips, and food percentages. Customers' data will consist of first and last names, email addresses, passwords, and earned reward points. In the case of managers, the system will store restaurant transactions for easy access, enabling them to track total revenue and profit. Additionally, managers will have the ability to personalize the menu by adding or removing items based on their sales performance. Crucial data to be stored includes aggregate ratings for each menu item, item prices, item names, recipes, and item IDs or numbers. All this data will reside in a MySQL database hosted on our web server.

## 8. Project Management

### **History of Work**

#### **August 21-September 4**

After getting together after class and forming the team, we created a discord to brainstorm ideas for the project and establish a line of communication. After brainstorming for a day or 2, we all came up to the conclusion of doing a restaurant automation project since one of the members already had experience working at a restaurant and the rest of the members felt connected to such an idea. We got together on the 31st of August to write up the proposal and discuss our strengths and weaknesses to decide what we are all capable of accomplishing.

#### **September 5-September 10**

We used the feedback from the proposal to plan the next few steps of the project. Some of the feedback helped come up with possible solutions the professor brought to our attention. We also took into consideration previous projects and reports and the functions they created. We could not get together during this time but everyone contributed to their assigned parts of the report. Bjarni started working on the database design using a UML diagram. He also created a SQL query for the database.

#### **September 11-September 17**

We will discuss Bjarni's database design during this time and the SQL query database he created. Also, we will brainstorm user interface ideas and the appearance of the site. We hope to come up with a model that best suits what we want our site to look like and accomplish. Lastly, we started working on part 2 of Report 1 and divided up the work.

**September 18-October 1**

We will take this time to continue discussing models and agree on a model that we are all comfortable and happy with. We will also start implementing the user interface and database.

**October 2-November 2**

We will continue implementing the user interface and database.

**November 3-November 19**

We hope that by this time, we will have a fully implemented model and get started with the testing part of the project.

**November 20-December 8**

We will take this time to fix bugs and present our final model to the class.

9. References

**Links to previous restaurant automation reports**

<https://www.ece.rutgers.edu/~marsic/books/SE/projects/Restaurant/>

**Software Engineering Project Report**

<https://www.ece.rutgers.edu/~marsic/Teaching/SE/report1.html>

**Restaurant Automation Project Description**

<https://www.ece.rutgers.edu/~marsic/books/SE/projects/Restaurant/RestaurantAutomation.pdf>

**10 Common Software Architectural Patterns in a nutshell**

<https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013>