



Course Name: Software Engineering

Course Number: 14:332:452

Group: #10

Super Foods

Full Report #1

Submission date: February 23, 2021

URL: <https://github.com/BriannaSolano/superfoods2.0>

Group Members:

Robert Kulesa, Justin Chan, William Basanaga, Lindsay Wisner, Guilherme Kenji Silva
Horikawa, Saurabh Bansal, Keith Lo, Brianna Solano Aguilar, Rawad Sayah, Jeremy Kim,
Gian-Soren Morici

TABLE OF CONTENTS

Work Contribution	4
Section 1: Customer Statement of Requirements	5
1.1: Problem Statement	5
1.2: Glossary of Terms	10
Section 2: System Requirements	11
2.1: Business Goals	11
2.2: Enumerated Function Requirements	11
2.3: Enumerated Non-Functional Requirements	15
2.4: User Interface Requirements	17
Section 3: Use Cases	23
3.1: Stakeholders	23
3.2: Actors and Goals	23
3.3: Use Cases - Casual Descriptions	25
3.4: Use Cases - Diagrams	28
3.5: Traceability Matrix	30
3.6: Use Cases - Fully Dressed Descriptions	35
3.7: System Sequence Diagrams	39
Section 4: User Interface Specification	41
4.1: preliminary Design	41
4.2: User Effort Estimation	43
Section 5: System Architecture	46
5.1: Identifying Subsystems	46
5.2: Architecture Styles	48
5.3: Mapping Subsystems to Hardware	49
5.4: Connectors and Network Protocols	49
5.5: Global Control Flow	50
5.6: Hardware Requirements	51
Section 6: Project Size Estimation	52
6.1: Unadjusted Actor Weight	52
6.2: Unadjusted Use Case Weight	53
6.3: Technical Complexity Factors	56
6.4: Use Case Points	56
Section 7: Plan of Work	57
Individual Contributions Breakdown for Report	61

Project Management **63**

References **65**

Work Contribution

Every member of the project contributed equally to Report # 1

Section 1: Customer Statement of Requirements

1.1: Problem Statement

Customer #1:

One of my favorite things to do on my nights is go out to eat at a restaurant, but lately with the Covid-19 Pandemic going on, it is very hard for me to get out of the house to do so; when I do get to leave the house, I feel extremely uncomfortable. What I want is to have a contactless, efficient delivery service. I hated waiting for my food when I was physically in the restaurant, and being home I don't want to wait any longer than I have to. Even if I did get takeout delivered to me, more than half the time, the food comes cold or is thrown around in the car and the food leaks out to the outside of the container. I don't want my order arriving messy as if it was thrown around.

What could Super Foods do? Super Foods needs to create a delivery system that is very convenient, efficient, effective and above all else safe. I think that this will benefit not only myself, but even those who are compromised and cannot leave their homes in times like this. I would like to have an app where I can make my order without feeling the pressure that I was wasting someone's time like when I am ordering by phone. An app would allow me to write down my order to the finest details and have all of the requests processed without anyone mishearing me over the phone. Even though an app would have the least communication and interaction with members of the restaurant, it would still allow me to convey complicated orders with a low chance of the order being messed up. I would never have to worry about the receiver mishearing my order or forgetting a special request I make for a dish. For this, I think that it would be a good idea to have an app that orders food and has drones to fulfill the deliveries. This would limit my face to face interaction with outside people and would make me feel safer than if I were getting my order directly from a delivery driver. The drones would even make the delivery process much quicker whether I am at home or I decided to opt-in for a curbside pick up. The drones will never need to stop at stop signs or red lights. They would just fly in a straight line and deliver my food while it's still hot. The estimated time of arrival would not change because of a traffic accident or road work. On top of that, I would not even have to worry about the food falling from the seat to the floor if the delivery driver makes a sudden break. Although I think food is very important to a restaurant, the delivery aspect of a restaurant is just as important. Just imagine if the food you ordered always arrived hot and in the condition that the restaurant packed it. That would be amazing. Sometimes I have to wait forever all to get a cold meal; I think if they put in place a system and used current technologies to create the most efficient route of delivery, I would definitely be a regular customer.

Customer #2:

It's annoying, after a long day, to go drive out to my favorite restaurant, and order my favorite meal. Just to find out after some time that it can't be made correctly because the restaurant ran out of stock of a key ingredient.

What could Super Foods do? It would be nice if I could immediately be warned that an ingredient is out of stock when I click on an item that contains it, that way I can choose to order a different item. Better yet, if I see the ingredients that are needed to make my dish start to go down, I can set aside what I am doing to place an order immediately before the ingredients run out. This level of transparency gives me the power to weigh the priorities of the tasks that I need to do. If someone told me that if I placed an order within the next 5 minutes or else I wouldn't be able to order it later, I would.

Customer #3:

I enjoy going out to restaurants often, but I don't like not knowing whether what I'm eating is healthy or not. I have two young children with severe food allergies making the selection of restaurants even slimmer. Fast food and TV dinners are not what I want to be feeding my kids. Often in restaurants we enjoy high quality meals, but dishes are often overloaded with calories, salt or sugar. It would be nice if I could see the health data, especially calorie values, of the foods being offered. That way, I can make the decision to order a healthier item if I'm feeling health conscious.

What could Super Foods do? On the menu, Super Foods can implement a program to display the nutritional information of an item when selected by the customer. Indicators such as red font can indicate if there are large amounts of calories. A button will allow the customer to toggle the ability to view nutritional values of items on the menu. Superfoods can also keep track of the total calorie count of the customer's entire order, assuming the button is toggled on.

Employee:

Delivering for a company has its ups and downs. Sometimes I get tips on top of each delivery which is nice and all, but sometimes I feel like it's just not worth leaving the restaurant just to deliver a small dinner four blocks away. I have to take out my car, get stuck in traffic, waste ten minutes of my life only to receive complaints that I'm delivering cold food.

What could Super Foods do? Super Foods could make me a way to deliver these orders without having me drive to those places. They could also make me something that would allow me to not get stuck in traffic. I am thinking that a drone style delivery system would be ideal. This way I could just dispatch a drone to those time wasteful jobs and save my energy for longer hauls, or catering orders.

Manager:

Since I have been with this company, I have noticed that it is extremely difficult to keep track of the inventory we have coming in and out of the restaurant. The way that we take inventory now is that every night, after we close, we must count all of our ingredients, and that is what is going to determine what we need to order more of. This becomes very time consuming to all of those involved, and I think the time we spend counting inventory can be used in many other areas of the restaurant, not in places that are unnecessary. Manually doing the inventory also becomes very stressful because you have to make sure that your counting is exact, and it is very hard to

keep track when we can have a lot of the same item. It also becomes an issue during inventory when we have to throw away a product due to it being spoiled or not selling. Sometimes without realizing, we order too much of the same thing and end up throwing it out.

What can Super Foods do? I think Super Foods could build an inventory system for our restaurant. I think it should have some features like automatic inventory count, manual override, automatic reordering of products, and overall, I think it just needs to be organized in a good manner. When I say automatic inventory count, I want the inventory to automatically update once a shipment comes in. I think manually updating the inventory every time a product comes in would be very tedious and it would allow a lot of room for error. At the same time, we should be able to manually override the inventory count if we feel necessary. Sometimes when the chefs are in the kitchen, there is food that is dropped and needs to be thrown away before it even reaches the customer. The inventory I also think should be processed in a way that after something is ordered, it is automatically deducted from the inventory count so it has a most up to date count that we can be made aware of. The system should also be able to notify me when a product needs to be ordered again, and it should even order it for the restaurant when it reaches a certain point. This will prevent me from forgetting to order important products vital to the restaurant's production. As far as organization goes, I just want it to be easy to read and organized in a way that places all of the products into categories whether it be freezer and refrigerated foods, perishables, or even basic supplies. This would be far easier to navigate through. In the database that Super Foods creates, I think it makes most sense for me to be able to manually enter the products we have once, and from there on out it would save it. This will let me know exactly what I'm looking for based off of what I decided to name the new items. In this same database, you can add specific details about the food that will be portrayed to customers, such as the calorie count. This could go hand in hand with whether a product is out of stock or not.

Owner:

Restaurants want to provide their customers with a unique experience while also generating revenue. It's typically the same routine for every restaurant, sit, order, wait, eat. I often see adults bored on a weekend continuously debating on something fun to do and ultimately deciding on nothing. As a restaurant owner, I want to provide my customers with a fun experience that can only be given at my restaurant. Our alcoholic beverages don't receive as much attention from our customers as they should, resulting in poor sales in that area. I want my alcoholic drinks to receive more attention and be the focus of this new experience. Overall I want to give my customers an opportunity to have fun while drinking with friends through games. I want my restaurant to have a fun environment where customers continue to return.

How will Super Foods help? I want SuperFoods to have a drinking game that is both fun, easy, and built from scratch. I want customers to be required to sign in with an account to have access to the game. I believe this would incentivize my customers to create accounts and allow the game to remain unique to the restaurant. If the game were to be accessed only by account, my restaurant would gain a reputation for having an exclusive game. From the application, I

want customers to be able to select the game option from the main menu. When selected, my customers would be redirected to a website where they can select to play the drinking game. I want the drinking game to provide customers with a range of "Would you rather" questions. Customers will have the option to choose between two choices till the timer in the game runs out. From there the game randomly picks one of the options given and displays it to the customer with the text "Take a shot". At any time, customers can quit the game.

Owner:

I love the application Turbo-Yum but as a restaurant owner I feel that it does not reach its full potential. I always have a problem determining how customers feel about my restaurant. I want to know how my new waiter is performing or the new ratings of my new dish from the customers' point of view. Obtaining reviews from customers on specific things requires too much time and work that can be allocated somewhere else. Turbo-Yums doesn't provide me with a solution for this review problem. I also feel as if Turbo-Yum does not help my restaurant provide a relationship with its guests, such that they don't feel like guests but rather like regulars to my restaurant. I find it hard for my restaurant to reach out to its customers beyond just food. Not enough customers are creating accounts to the application Turbo-Yums. Rather than going through the trouble of signing in, they would rather check out as a guest. Turbo-Yums does not provide that incentive for customers to take the time and create an account with us. I find that customers come to the restaurant, order something they have not ordered before, enjoy it, but forget what it was when they return back to the restaurant since the app doesn't store this information. Rather than searching for items, they opt for something they typically ordered. I find that the app has many major features but not everyone should have access to them. Typically, new employees who are not familiar with the app interface may mistakenly access the inventory and change stock. This problem has become frequently at the restaurant and not every employee should have the same employee status and app privileges as others.

How will Super Foods help? I want Super Foods to enhance Turbo-Yums so that it reaches its full potential. I want the application to provide my customers with a guide on what specific subject my restaurant looks for in a review. I want the application to allow customers to leave reviews on particular foods, on their waiters, the restaurant itself, or any comments they have. I would also want the application to notify waiters if there is a special anniversary day for a guest in the restaurant. Making them feel that my restaurant cares for its customers. I also want Super Foods to provide incentives for my customers to create an account and continuously sign in rather than continuing as a guest. Super Foods should provide a reward system for customers who create an account and sign in. Earning coupons after spending a certain amount or eating at the restaurant a certain number of times. These rewards would be tied into the account. Through the account creation, account holders should have the ability to review past orders from the restaurant. This would incentivize customers even more to create an account, so they will not forget that dish they liked the previous visit. I want Super Foods to expand on the portal and give different status to my employees. I want to have a diverse selection of status to give my employees such as waiter, bartender, chief, and manager. I want these statuses to have privileges right in the app for my employees. Such as chiefs and managers can edit inventory while waiters can only view. This would give my employees a better environment to navigate. I

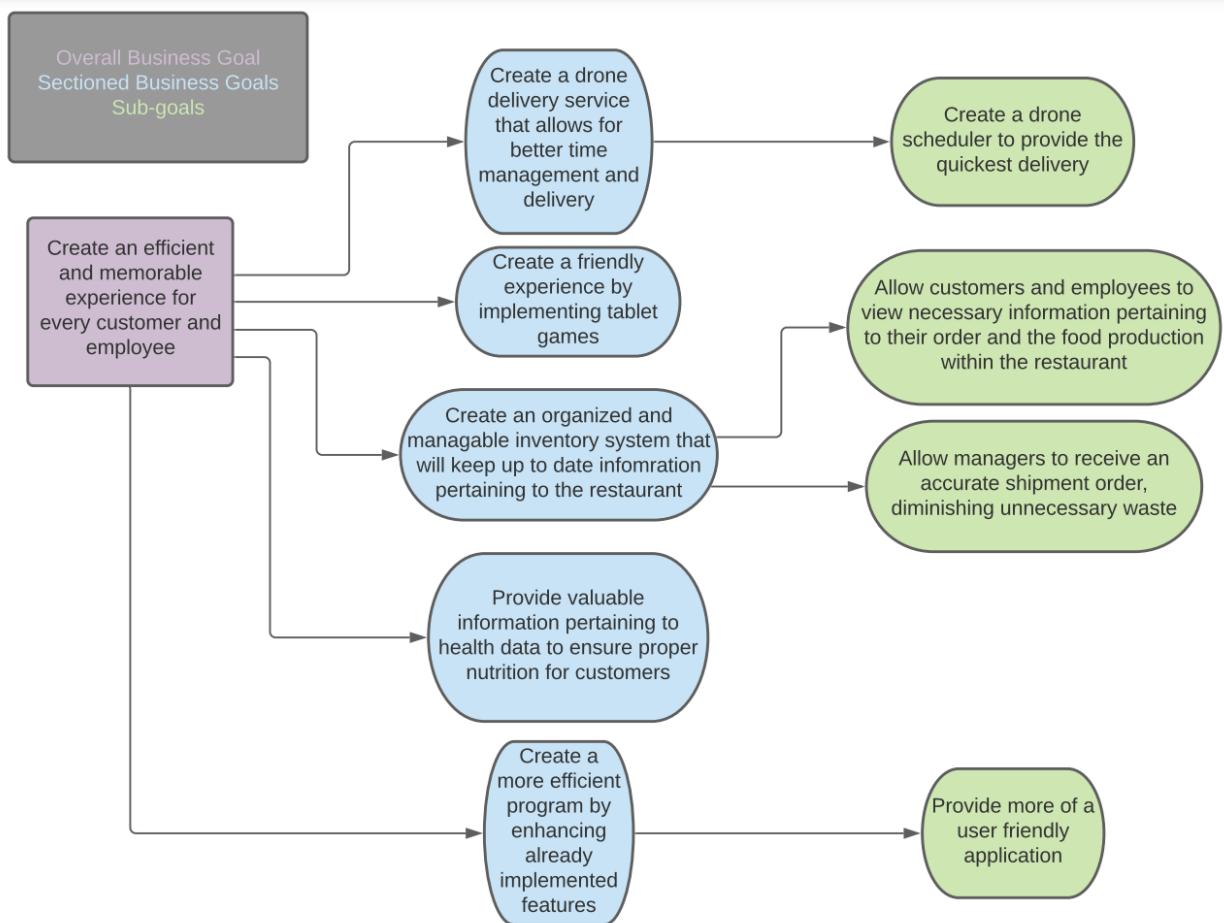
want Super Foods to be an extension of the application TurboYums. My customers and employees are already used to the TurboYums interface and I don't want to give them a new tool that they have to learn. I still want to retain some of TurboYums key features like app transactions and menu view but with new options in place.

1.2: Glossary of Terms

- **Application** - A software designed to perform a group of coordinated functions, tasks or activities for the benefit of the user
- **Calories** - Health data specific to the product
- **Drone** - Unpiloted vehicle used to delivery
- **Drop Down Menu** - Customers and employees can use this application feature which gives them options to relocate them to different sections within the application
- **Employee Portal** - Allows employees to locate inventory, drone delivery service, menu and orders
- **Inventory** - A list of available items that are able to be used, such as goods and products that are currency in stock on restaurant property
- **Operating System** - System software on which the application will run
- **Restaurant Automation** - Makes a restaurant flow more efficiently and more easily. Uses devices preloaded with software that manages several tasks which helps eliminate many of the required taste that are normally done via employees
- **Tablet** - A portable thin, touchscreen computer that displays the user interface of Super Foods Application

Section 2: System Requirements

2.1: Business Goals



2.2: Enumerated Function Requirements

2.2.1: Drone/Driven Delivery Service

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - D1	4	As a customer, I need to be able to input my delivery address so that my order can be delivered to me.
REQ - D2	5	As a manager, I need the system to determine which delivery method is optimal to reduce delivery costs.

REQ - D3	4	As a manager, I only want the drone operator to view this module.
REQ - D4	3	As a drone operator I need to log battery level and battery swaps in the app when drones leave and return so that the drone successfully delivers food and returns without losing power.
REQ - D5	3	As a drone operator, I should be able to update the inventory in the app when a spare battery is fully charged.
REQ - D6	3	As a manager, I need to authorize drone delivery before they go out for delivery so that I can manage when an order is delivered.

2.2.3: Adult “Would you Rather” Game App

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - A1	1	As a Customer, I need the ability to view the rules of the game.
REQ - A2	1	As a Customer, I need the ability to play the game after reviewing the rules.
REQ - A3	2	At the beginning of every question, the system should display a ready slide for a certain amount of time then prompt the question.
REQ - A4	4	The system should display the “would you rather” questions for a total of 15 seconds before prompting to the randomly selected answer.
REQ - A5	3	As a customer, I need to request an alternate “would you rather” statement whenever I receive an old question.
REQ - A6	5	The system should prompt for the answer of the “would you rather” question for a certain amount of time after displaying the question. This is displayed for a set time then prompted to the ready screen.
REQ - A7	5	The system should allow the user to quit during the question prompt and redirect users to main menu.

2.2.4: Inventory System

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - I1	5	As an employee, I need to view and modify inventory count so that the kitchen is aware when a product is under stocked.
REQ - I2	4	As a manager, I want my menu items to be hidden from view if an item is out of stock.
REQ - I3	4	As a manager, every time an item is ordered, I want the inventory to reflect those changes.
REQ - I4	2	As a manager, I need to receive notifications when inventory of an item is out so that I can restock for the following work night.
REQ - I5	4	As a manager , I need the Inventory to be adjusted automatically when a shipment arrives.
REQ - I6	4	As an Employee, I need to view incoming deliveries & their dates so that I can inform my customers accordingly when an item will be back on the menu.
REQ - I7	2	As a manager, I need to cancel or reschedule incoming deliveries in case our stock drastically changes.

2.2.5: Calories/Health Data

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - C1	2	As a customer, I need to view a menu with/without caloric/nutritional value so that I can feel confident in what I'm ordering.
REQ - C2	5	As a customer, I want/do not want to be aware of the calorie count of each menu item.
REQ - C3	5	As a customer, I need the system to calculate the total amount of calories of my order, allowing me to be aware of how much I'm eating.
REQ - C4	4.5	As a manager, changes to the menu should also be reflected on the caloric values.

REQ - C5	3	As a Customer, I need to be able to see dietary restrictions and common food allergies so that I am not taking any health risks.
----------	---	--

2.2.6: Turbo Yum Menu/Order System enhancement

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - T1	5	As a user, I need to view and manage my order so that my order is accurate.
REQ - T2	5	As a manager, I need to edit menu items and their description so that our menu is up to date.
REQ - T3	4	As a user, I need to create an account with my username, email, birthday & password so that I can place an order and collect rewards.
REQ - T4	3	As a manager, I need to be notified when a user fails to sign in after 5 attempts to maintain security.
REQ - T7	5	As a manager, I need to lock placing orders when the restaurant is closed.
REQ - T8	3	As a user, I need to track my order even when the manager has locked placing orders.
REQ - T9	3	As a manager, the system needs to calculate the current price of the customer's order accurately.
REQ - T11	3	As a user, I need to schedule a delivery time within the managers' allocated time frame in case I do not want the food immediately.
REQ - T12	3	As a manager, I need to manage the windows orders are being delivered.
REQ - T14	5	As a user, I need to input my card information to complete the order.
REQ - T15	1	As a user, I need to leave star ratings or comments on orders to share my opinions about the restaurant.
REQ - T16	2	As a manager, I want to reward loyal customers by giving them a reward point for any orders over \$20.
REQ - T17	2	As a manager, the system should apply a 10% coupon when a loyal customer collects 10 points as an incentive

		to order more often.
REQ - T18	1	As a manager, I want to reward points on birthdays so that customers feel valued.
REQ - T19	1	As a user, I need to view my order history in order to leave a review or place a similar order.

2.3: Enumerated Non-Functional Requirements

2.3.1: Drone/Driven Delivery Service

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - D7	5	The system should be able to get real time, local weather data that is accurate and taken from a credible source.
REQ - D8	5	The system should provide a seamless and accurate transfer from street addresses to coordinates.
REQ - D9	5	The system should provide accurate order weight data.
REQ - D10	5	The system should provide accurate data on battery status.
REQ - D11	5	The system should decide whether a drone or driver should be sent out based on data such as weather, distance, and battery status.

2.3.3: Adult “Would you Rather” Game App

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - A8	1	The system should provide smooth transitions from “would you rather” statements to another.
REQ - A9	5	Users should be able to easily understand the rules of the game.
REQ - A10	3	Users should be able to access the game across various devices.

REQ - A11	5	Users should be able to easily understand information displayed from the game.
-----------	---	--

2.3.4: Inventory System

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - I8	3	The system should be intuitive and easy to understand/use.
REQ - I9	3	The system should offer a different view depending on whether the user is an employee or customer.
REQ - I10	2	The system should be well organized in a manner that makes most sense to employees.

2.3.5: Calories/Health Data

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - C6	3	The system should not have a terribly noticeable delay in the update of the caloric values.
REQ - C7	5	The system should instantly process and display ingredients and nutritional values to the current menu when added.
REQ - C8	5	The system should provide instantaneous updates to the menu with their caloric value when a new item/ingredient is added by the kitchen staff.
REQ - C9	4	System is compatible with most recent versions of software (ie. iOS 11 or Windows 10).
REQ - C10	5	The caloric and nutritional values should be accurate and taken from a credible source

2.3.6 Turbo Yum Menu/Order System enhancement

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - T20	4	Order pricing should be accurately calculated and

		displayed.
REQ - T21	3	Should maintain an accurate record of sales/inventory loss.
REQ - T22	1	Should maintain an accurate record of rewards points.
REQ - T23	2	Menu items should be accurately organized pertaining to their group.
REQ - T24	3	Menu Items should hold accurate information pertaining to their makeup.
REQ - T25	4	Adjustments made to each item ordered should be accurate.
REQ - T26	5	Each user with a different restaurant occupation should only be able to see information pertaining to their job.

2.4: User Interface Requirements

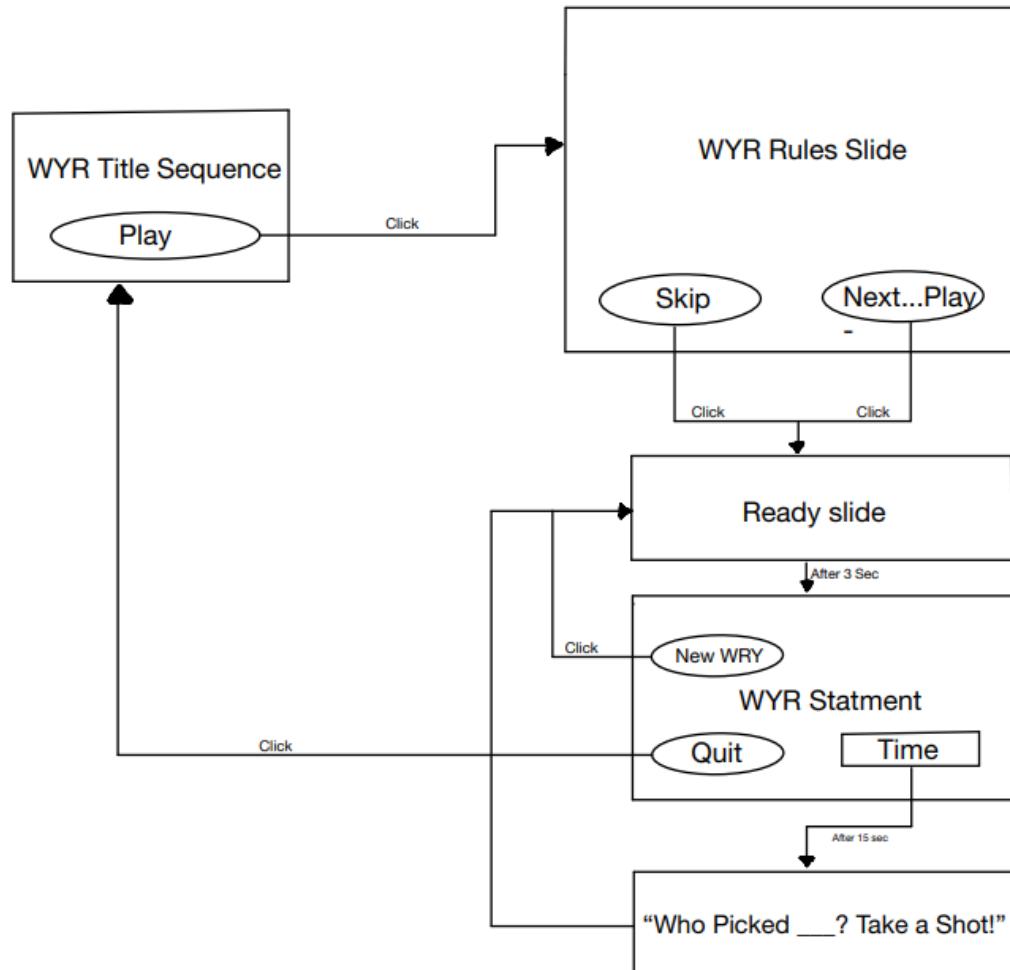
2.4.1: Drone/Driven Delivery Service

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - D12	5	The app should display to the drone operator and the manager on duty the status of each drone.
REQ - D13	2	The app should display any issues impeding a drone delivery.

2.4.3: Adult “Would you Rather” Game App

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - A12	5	The system should display a button to play in the game's main menu.
REQ - A13	2	The system should display a button to skip the tutorial or play through the slides containing the rules.
REQ - A14	2	At the beginning of every questions the system should display a ready slide .

REQ - A15	2	The system should display the would you rather questions for a total of 15 seconds.
REQ - A16	1	The application should display a timer at the bottom to allow users to see the remaining time to answer the question.
REQ - A17	4	The application should display a button during the "would you rather" question to display a new question.
REQ - A18	5	The application should display a randomly selected answer to the "would you rather question" in the middle of the user's screen.
REQ - A19	5	The application should display a quit button during the question display.



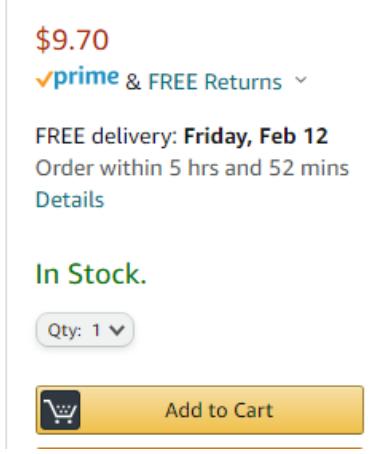
2.4.4: Inventory System

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - I11	2	Display items that are out of stock and notify the customer on the menu.
REQ - I12	3	Inventory system should be split into categories of items (hot ingredients, cold ingredients, drinks, etc)
REQ - I13	2	Employees should be able to see the status of all items stock

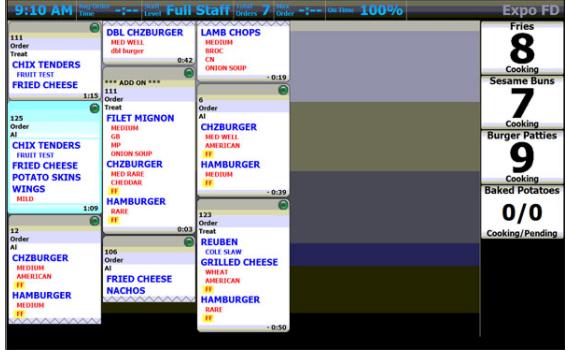
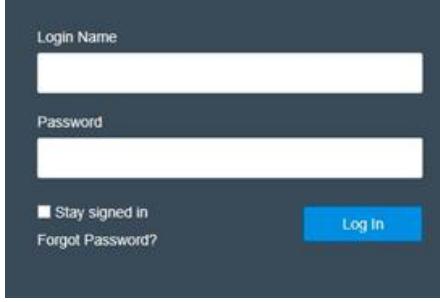
2.4.5: Calories/Health Data

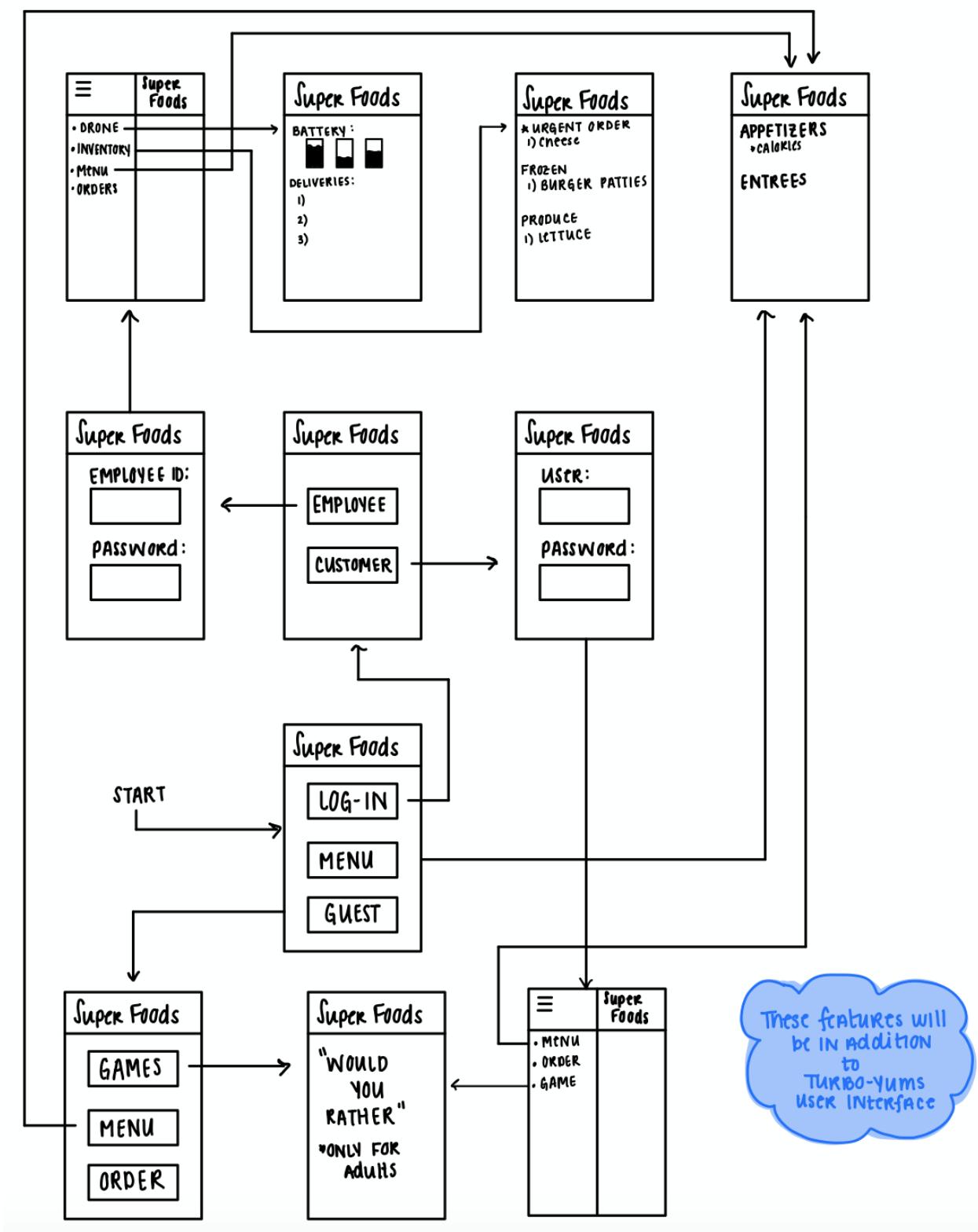
Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - C13	3	Ingredients/nutritional values should be displayed to their respective images.
REQ - C14	2	An option to hide detailed nutritional values should be available.

2.4.6 Turbo Yum Menu/Order System enhancement

Identifier	Priority (Higher number indicates higher priority)	Requirement
REQ - T27	5	<p>System should display buttons to customer to add items to cart</p> 

REQ - T28	5	<p>System should display buttons to customers to place an order.</p> <table border="1"> <tr> <td>Items:</td><td>\$147.06</td></tr> <tr> <td>Shipping & handling:</td><td>\$0.00</td></tr> <tr> <td>Amazon Photos Promo:</td><td>-\$10.00</td></tr> <tr> <td>Total before tax:</td><td>\$137.06</td></tr> <tr> <td>Estimated tax to be collected:</td><td>\$9.08</td></tr> </table>	Items:	\$147.06	Shipping & handling:	\$0.00	Amazon Photos Promo:	-\$10.00	Total before tax:	\$137.06	Estimated tax to be collected:	\$9.08
Items:	\$147.06											
Shipping & handling:	\$0.00											
Amazon Photos Promo:	-\$10.00											
Total before tax:	\$137.06											
Estimated tax to be collected:	\$9.08											
REQ - T29	3	<p>System should display options to modify ingredients.</p> <table border="1"> <tr> <td colspan="3">CUSTOMIZE YOUR CHICKEN</td> </tr> <tr> <td><input type="checkbox"/> No Cheese Subtracts 110 Cals</td> <td><input type="checkbox"/> No Mexi-Ranch Subtracts 270 Cals</td> <td><input type="checkbox"/> No Pico De Gallo Subtracts 25 Cals</td> </tr> <tr> <td><input type="checkbox"/> No Tortilla Strips Subtracts 160 Cals</td> <td colspan="2"></td> </tr> </table>	CUSTOMIZE YOUR CHICKEN			<input type="checkbox"/> No Cheese Subtracts 110 Cals	<input type="checkbox"/> No Mexi-Ranch Subtracts 270 Cals	<input type="checkbox"/> No Pico De Gallo Subtracts 25 Cals	<input type="checkbox"/> No Tortilla Strips Subtracts 160 Cals			
CUSTOMIZE YOUR CHICKEN												
<input type="checkbox"/> No Cheese Subtracts 110 Cals	<input type="checkbox"/> No Mexi-Ranch Subtracts 270 Cals	<input type="checkbox"/> No Pico De Gallo Subtracts 25 Cals										
<input type="checkbox"/> No Tortilla Strips Subtracts 160 Cals												
REQ - T30	1	<p>Managers should have the ability to include pictures in their menu items.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> </div> <div style="text-align: center;"> <p>FIESTA LIME CHICKEN</p> </div> </div>										
REQ - T31	5	<p>System should display tables to host, colored in red if a table is waiting for an order, green if the table is occupied, and neutral if a table is free, yellow if the table needs to be bussed.</p>										

		
REQ - T32	5	<p>System should display order tickets to chefs and bartenders in the order of which they came.</p> 
REQ - T33	4	<p>System should display buttons to chefs and bartenders to allow indication of a finished item.</p>
REQ - T34	4	<p>System should display one login screen for different occupations of a restaurant (Host, Chef, Bartender, Manager, Customer).</p> 



Section 3: Use Cases

3.1: Stakeholders

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

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

3.2: Actors and Goals

Initiating Actors

Actor	Role	Goal
Customer	The customer is a restaurant visitor who may choose to dine in or take out food, view the menu, order a meal, eat, and pay for service.	The goal of the customer is to have an excellent dining experience with minimal wait times and smooth service.
Guest	The guest is a customer that does not have an account and chooses to opt out of the features that come with an account, other than that a guest is the same of a customer.	The goal of the guest is to have an excellent dining experience with minimal wait time and smooth service.
Employee	The employee is any type of worker at the restaurant, except for the manager.	The goal of the employee is to provide an excellent dining experience for the customers.
Manager	The manager is the employee who has the additional responsibility of managing all of the needs of the restaurant.	The goal of the manager is to manage employees and scheduling, keeping track of inventory, and monitoring revenue and losses while they also ensure that restaurant customers are accounted for.

Participating Actors

Actor	Role
Busboy	The busboy is the employee responsible for cleaning the dishes and tables, and maintaining overall cleanliness of the restaurant. The busboy can see from the database that customers have left their table, marking the table as dirty. Once the table is cleaned, it can be marked as clean.
Chef	The chef is the employee responsible for cooking and preparing the food that is ordered by a customer. The chef receives a queue of orders, preparing them in the order they come in. The customer is updated on the status of their order (submitted, preparing, ready).
Database	The database is a system that records a customer's order, table selection, and menu options. It essentially acts as persistent storage for all information that needs to be stored for our application to function.
Host/Hostess	The host/hostess is the employee responsible for greeting incoming customers and assigning seats to them. In the event that the guests have already selected a table, the host/hostess will escort them to the table. The host/hostess can see from the database when a table is marked as clean. If the customer has not yet selected a table, the host can seat him and mark the table as occupied.
Servers	The server is the employee responsible for taking orders from customers and sending them to the kitchen queue, as well as serving the food when it is ready. The waiter/waitress receives notifications that a meal is ready, so that they can pick it up and serve it. They are also notified which table number to serve it to, and get notified when a customer needs additional assistance.
System	The system will hold responsibility for many calculations needed to perform certain tasks. This may include calculating all necessary information to support drone usage. The goal of the system is to produce information to allow drone deliveries to be efficient, and give the best customer experience when receiving their orders.
Drone Operator	The drone operator is the one responsible for driving the drones to the necessary locations. They will be notified by the system where the drone needs to go, and it will also tell the drone operator which drone needs to be used for that specific delivery based on battery usage. They are the ones responsible for utilizing and maintaining drones.
Drone	The drone is piloted by the drone operator. The drone will be responsible for delivering food to a customer specified address.
Display	The display will change depending on the actions that the customer or employee inputs into the system.

3.3: Use Cases - Casual Descriptions

Drone/Driven Delivery Service

- **UC-1: Data Input** - Allows user to input delivery address
Derivations: REQ-D1, REQ-T8
- **UC-2: Check Battery Level** - Allows drone operator to track battery levels
Derivations: REQ-D4, REQ-D5, REQ-D10
- **UC-3: Drone Tracking** - Allows the drone operator to track the drones
Derivations: REQ-D3, REQ-D4, REQ-D6, REQ-D12,
- **UC-4: Track Emergencies** - Allows the drone operator to track issues with drones
Derivations: REQ-D13, REQ-D14
- **UC-5: Order Tracking** - Allows the customer to track the location of their order
Derivations: REQ-T8
- **UC-6: Decision Making** - Informs the drone operator if drones should be used or not
Derivations: REQ-D2, REQ-D7, REQ-D8, REQ-D9, REQ-D10, REQ-D11, REQ-D14

Adult “Would you Rather” Game App

- **UC-7: Start the Game** - Allows the customer to start the game from the game menu.
Derivations: REQ-A1, REQ-A12
- **UC-8 : Tutorial** - Allows the customer to migrate through tutorial slides at their command explaining the rules of the game.
Derivations: REQ-A2, REQ-A9
- **UC-9: Play** - Allows the customer to leave the tutorial and play the game.
Derivations: REQ-A2, REQ-A3, REQ-A4, REQ-A6, REQ-A13
- **UC-10: Next** - Allows the customer to skip a would you rather question if the was repeated
Derivations: REQ-A5, REQ-A17
- **UC-11: End** - Allows the customer to end the game during a “Would You Rather” statement
Derivations: REQ-A7, REQ-A19

Inventory

- **UC-12: Shipment Deliveries** - Orders that have been placed and delivered to fulfill inventory needs
Derivations: REQ-I5, REQ-I6, REQ-I7
- **UC-13: View Inventory Count** - Allows employees to have an up to date count of the inventory
Derivations: REQ-I1, REQ-I2, REQ-I3
- **UC-14: Inventory Display:** A screen type display system with options to click on inventory list depending on the category and displays: Green(Over half), Yellow(Halfway), Red(Low)
Derivations: REQ-I2, REQ-I8, REQ-I9, REQ-I10, REQ-I12, REQ-C7

- **UC-15: Low Stock Alert:** Easy display alert system that notifies the manager of items in low stock and the manager has to acknowledge the alert to get rid of the notification
Derivations: REQ-I4, REQ-I11

Calories/ Health Data

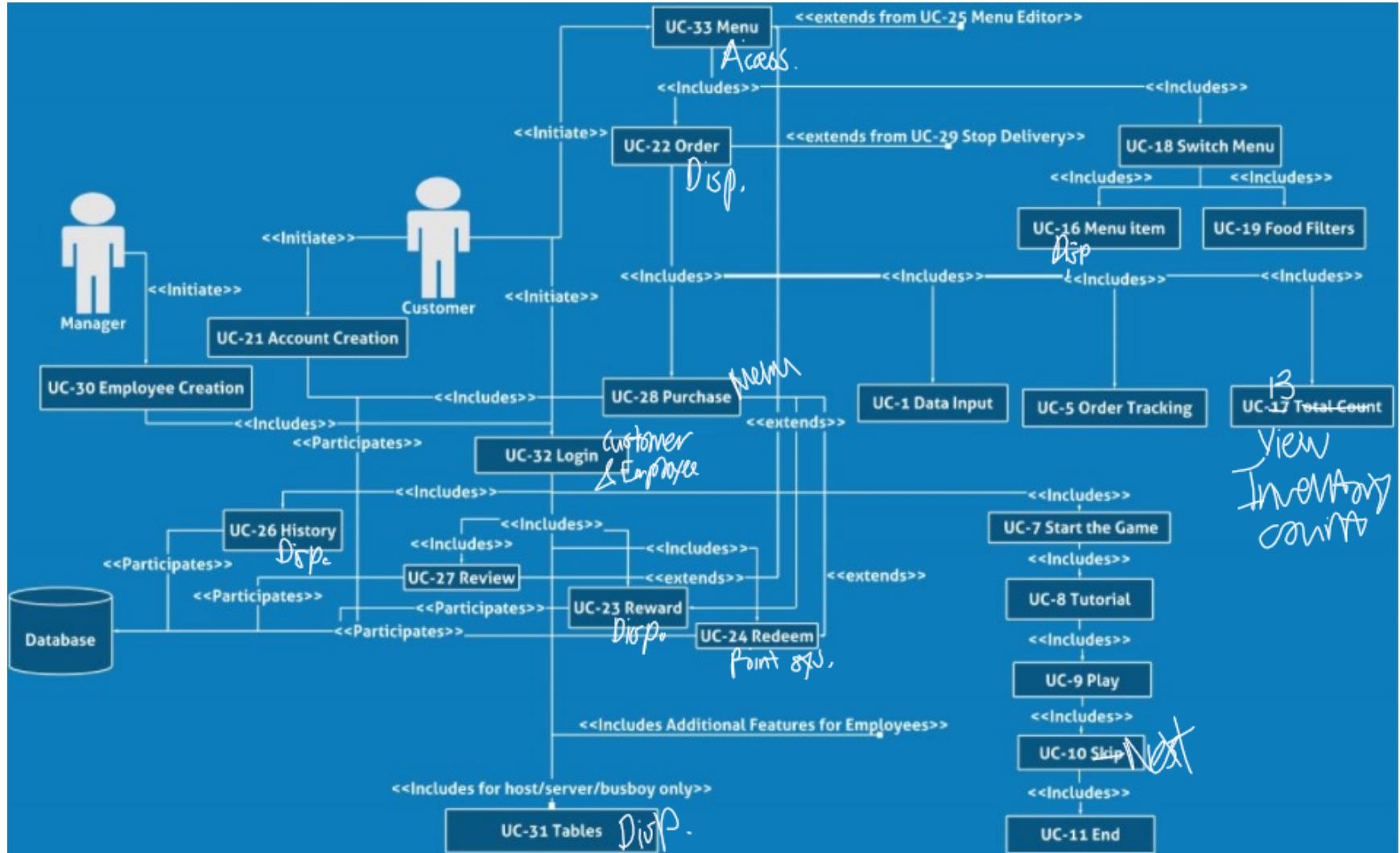
- **UC-16: Displaying Menu Item:** Customers should be able to view calories for each menu item
Derivations: REQ-C2, REQ-C10, REQ-C11, REQ-C12, REQ-C13
- **UC-17: Display Total Customers:** Customers should be able to view the total calorie count of their order
Derivations: REQ-C3, REQ-C6, REQ-C10, REQ-C12
- **UC-18: Switch Menu:** Allows customers to switch between menus with and without calories and nutritional information
Derivations: REQ-C1, REQ-C13, REQ-C14
- **UC-19: Add New item:** Employees should be able to manually add ingredients and their caloric values through the inventory
Derivations: REQ-C4, REQ-C8, REQ-C9, REQ-C10, REQ-C12
- **UC-20: Display Food Filters:** Allows customers to view menu items containing or not containing certain ingredients according to dietary restrictions
Derivations: REQ-C5

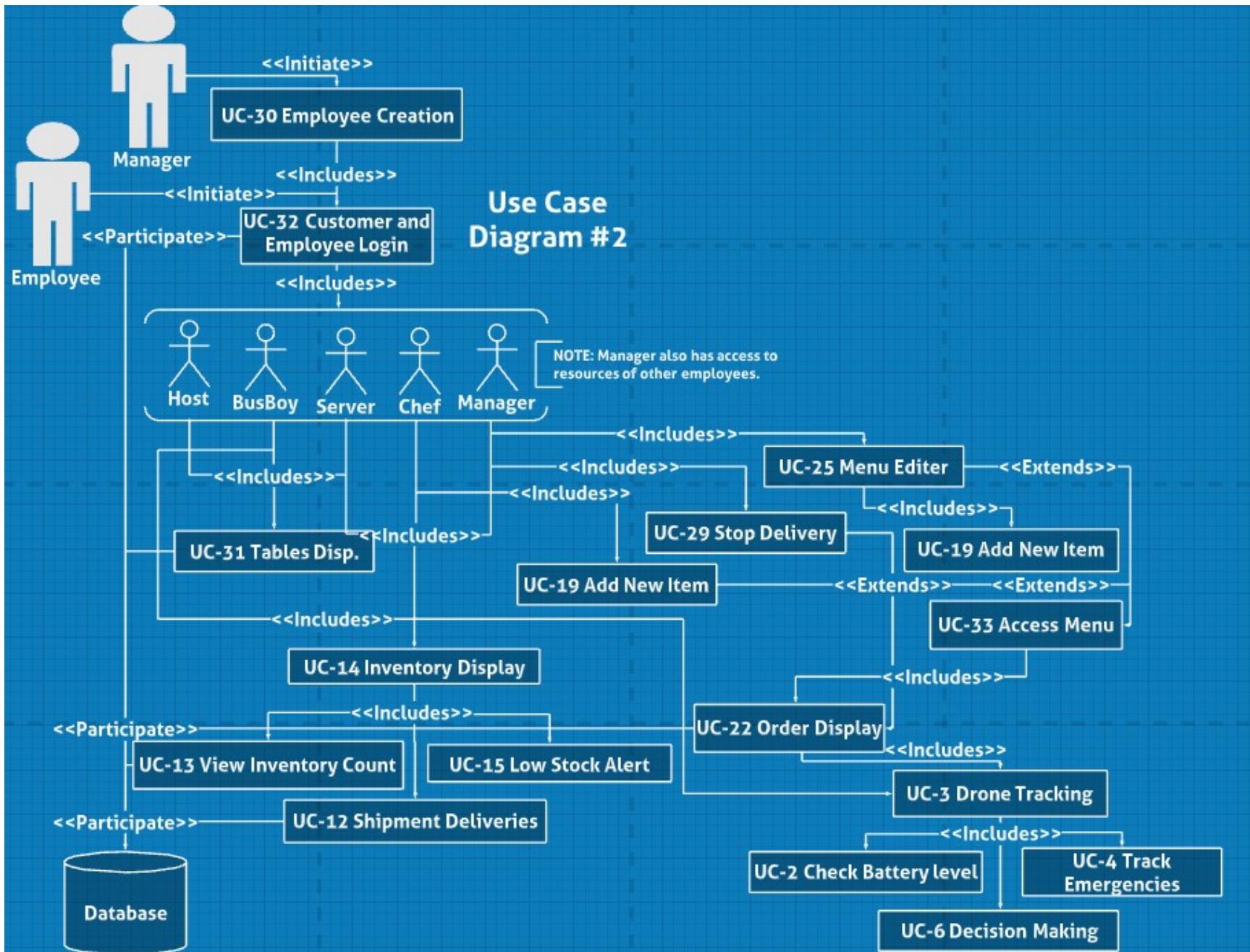
Turbo Yum Enhancements

- **UC-21: Account Creation** - Allows the user to create an account containing their username, password, email, and birthday, where they may receive account holder benefits.
Derivations: REQ-T3
- **UC-22: Order Display**- Allows the customer to view and manage their order.
Derivations: REQ-T1, REQ-T7, REQ-T8, REQ-T9, REQ-T11, REQ-T20, REQ-T24
- **UC-23: Reward Display**- Allows the customer to earn reward points making purchases and special occasions from their account.
Derivations: REQ-T3, REQ-T16, REQ-T18, REQ-T22
- **UC-24: Point Redemption System**- Allows the customer to redeem points towards a coupon for their next meal from their account.
Derivations: REQ-T3, REQ-T17, REQ-T22
- **UC-25: Menu Editor** - Allows the manager to select an item from the menu which will then allow them to change the description and availability of the item.
Derivations: REQ-T2, REQ-T6, REQ-T30, REQ-I4
- **UC-26: History Display** - Allows the customer to view past orders that have been made through their account.
Derivations: REQ-T3, REQ-T19
- **UC-27: Review Menu**- Allows the customer to leave a star rating and comments on different categories.
Derivations: REQ-T15

- **UC-28: Purchase Menu-** Allows customers to input card information and complete a transaction when placing an order.
Derivations: REQ-T14, REQ-T15, REQ-T28
- **UC-29: Stop Delivery -** Allows the manager to select a time to stop customers from selecting delivery when ordering.
Derivations: REQ-T12, REQ-T13
- **UC-30: Employee Creation -** Allows the manager to create employee accounts containing their username, password, email, SS and other proper credentials. The manager will also set the employee status which determines the features available to the employee.
Derivations: REQ-T3, REQ-T26, REQ-T-34
- **UC-31: Tables Display -** Allows host to view the table layout based on a color coded scheme where each color on table status.
Derivations: REQ-T31
- **UC-32: Customer and Employee Login-** Allow users to login in with their accounts with different functions based on the role of their account.
Derivations: REQ-T3, REQ-T4, REQ-T16, REQ-T19, REQ-T34, REQ-I9
- **UC-33: Access Menu-** Allow users to view the menu, food and beverage selection and reviews from other users.
Derivations: REQ-T15, REQ-T20 REQ-T23, REQ-T24, REQ-T27, T29, REQ-T30

3.4: Use Cases - Diagrams





3.5: Traceability Matrix

REQ-T16	5					x			x
REQ-T17	1						x		
REQ-T18	2					x			
REQ-T19	1						x		x
REQ-D7	5		x						
REQ-D8	5		x						
REQ-D9	5		x						
REQ-D10	5	x	x						
REQ-D11	5		x						
REQ-A8	1			x					
REQ-A9	5			x					
REQ-A11	5			x					
REQ-I8	3				x				
REQ-I9	3				x				x
REQ-I10	2				x				
REQ-C6	3					x			
REQ-C7	5				x				
REQ-C8	5						x		
REQ-C9	4						x		
REQ-C10	5					x	x	x	
REQ-T20	4							x	
REQ-T21	3			x					
REQ-T22	1						x	x	

REQ-T31	5																			x		
REQ-T32	5																			x		
REQ-T33	4																			x		
REQ-T34	4																		x	x		

3.6: Use Cases - Fully Dressed Descriptions

UC-6: Decision Making
Related Requirements: REQ-D2, REQ-D7, REQ-D8, REQ-D9, REQ-D10, REQ-D11, REQ-D13
Initiating Actor: Customer
Actor's Goal: Provide the system with the proper credentials such as address, for the system to perform the proper decision making for delivery method.
Participating Actors: Database
Preconditions: Customer must place an order Customer must enter address being delivered to
Postconditions: Customer will have order confirmation after processing payment
Flow of Events for Main Success Scenario: <ol style="list-style-type: none">1. → Customer enters delivery information including street address2. ← System converts street address into coordinates3. ← System decides drone delivery is the proper form of transportation4. ← System notifies customer time of delivery5. ← System assigns drone to delivery, taking into account battery parameters6. ← System notifies employee any issue pertaining to a drone delivery
Flow of Events for Alternate Success Scenario: <ol style="list-style-type: none">1. → Customer enters delivery information including street address2. ← System converts street address into coordinates3. ← System decides that car delivery is the proper use of transportation4. ← System notifies customer time of delivery

UC-22: Order Display

Related Requirements:

REQ-T1, REQ-T7, REQ-T8, REQ-T9, REQ-T11, REQ-T20, REQ-T24

Initiating Actor:

Customer

Actor's Goal:

To view and manage their order

Participating Actors:

Database

Preconditions:

Customer logs into application

Customer must click order from the drop down menu

Postcondition:

Customer is able to view and manage their orders

Flow of Events for Main Success Scenario:

1. → Customer signs in to application
2. → Customer selects "Order" where they are prompted to the menu
3. ← System displays menu with button next to each item offering "Add to order"
4. → Customer selects items to add to their order
5. → Customer clicks on cart
6. ← System shows items in cart, an order total, and options for delivery
7. → Customer chooses option for delivery via press button
8. ← System prompts customer to enter delivery method
9. → Customer enters all necessary information including name and address
10. ← System prompts customer to payment
11. → Customer enters card information or can select "Cash"
12. ← System prompts user to verify information
13. → Customer verifies information
14. ← System confirms order

Flow of Events for Alternate Success Scenario:

1. → Customer signs in to application
2. → Customer selects "Order" where they are prompted to the menu
3. ← System displays menu with button next to each item offering "Add to order"
4. → Customer does not select items to add to their order
5. → Customer terminates or reallocates themselves within the application

UC-32 Customer and Employee Login
Related Requirements: REQ-T34, REQ-I9, REQ-T16, REQ-T19, REQ - T4, REQ -T3
Initiating Actor: All users (customers, employees, managers, etc.)
Actor's Goal: To login to the application to receive account benefits
Participating Actors: Database
Preconditions: The user has the system loaded The user has their login credentials available
Postconditions: The user is prompted to the main menu with a selection of variable features based on account role (employee status, customer)
Flow of Events for Main Success Scenario: <ol style="list-style-type: none"> 1. → The user clicks what role they are from the main menu (employee or customer) 2. → The system displays the login screen with the login in credentials specified 3. ← The user enters their login credentials such as email and password 4. → The database verifies login and account role (employee or customer) 5. → The system prompts the user to the main menu
Flow of Events for Alternate Success Scenario: <ol style="list-style-type: none"> 1. → The customer clicks “customer” from the main menu 2. → The system displays the login screen with the login in credentials specified 3. ← The customer clicks on create a new account 4. → The system prompts the user to the account creation screen 5. ← The system display the credentials birthday, email, and password 6. → The customer inputs the credentials 7. ← The system stores the information to the database 8. ← The system prompts the users to the main menu

UC-33: Accessing Menu

Related Requirements:

REQ-T15, REQ-T20, REQ-T23, REQ-T24, REQ-T27, REQ-T29

Initiating Actor:

Customer

Actor's Goal:

Have a clear understanding and is able to properly utilize the menu

Participating Actors:

Database

Preconditions:

Customer must log into the system using their credentials or chose an option as guest
Customer must select Menu from the drop down bar

Postconditions:

The user is shown the available menu

Flow of Events for Main Success Scenario:

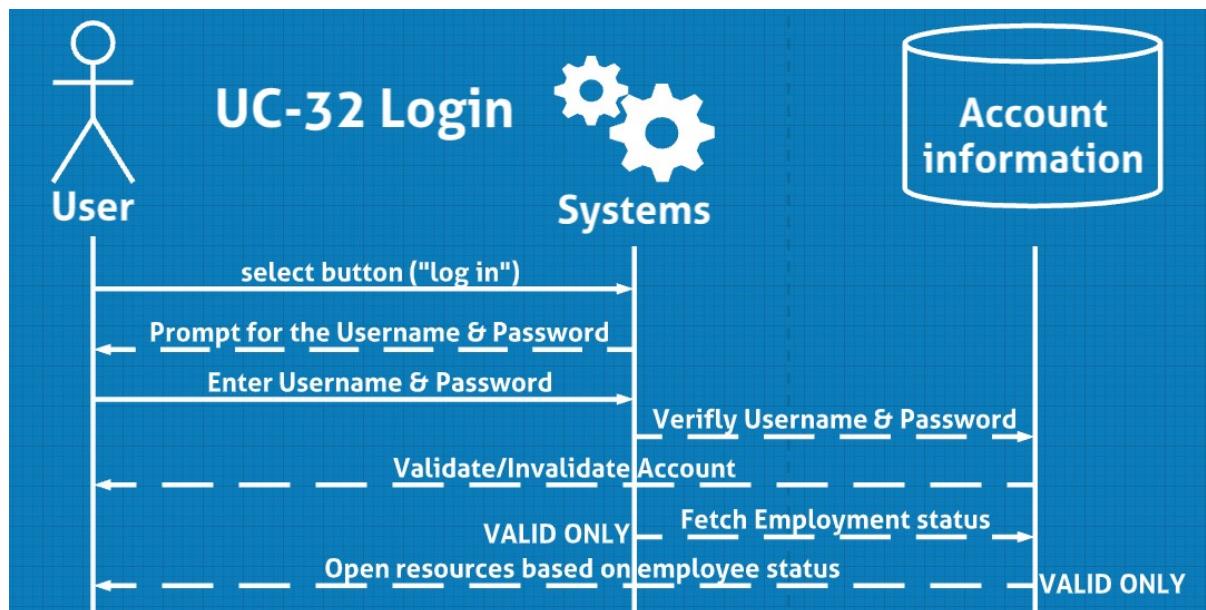
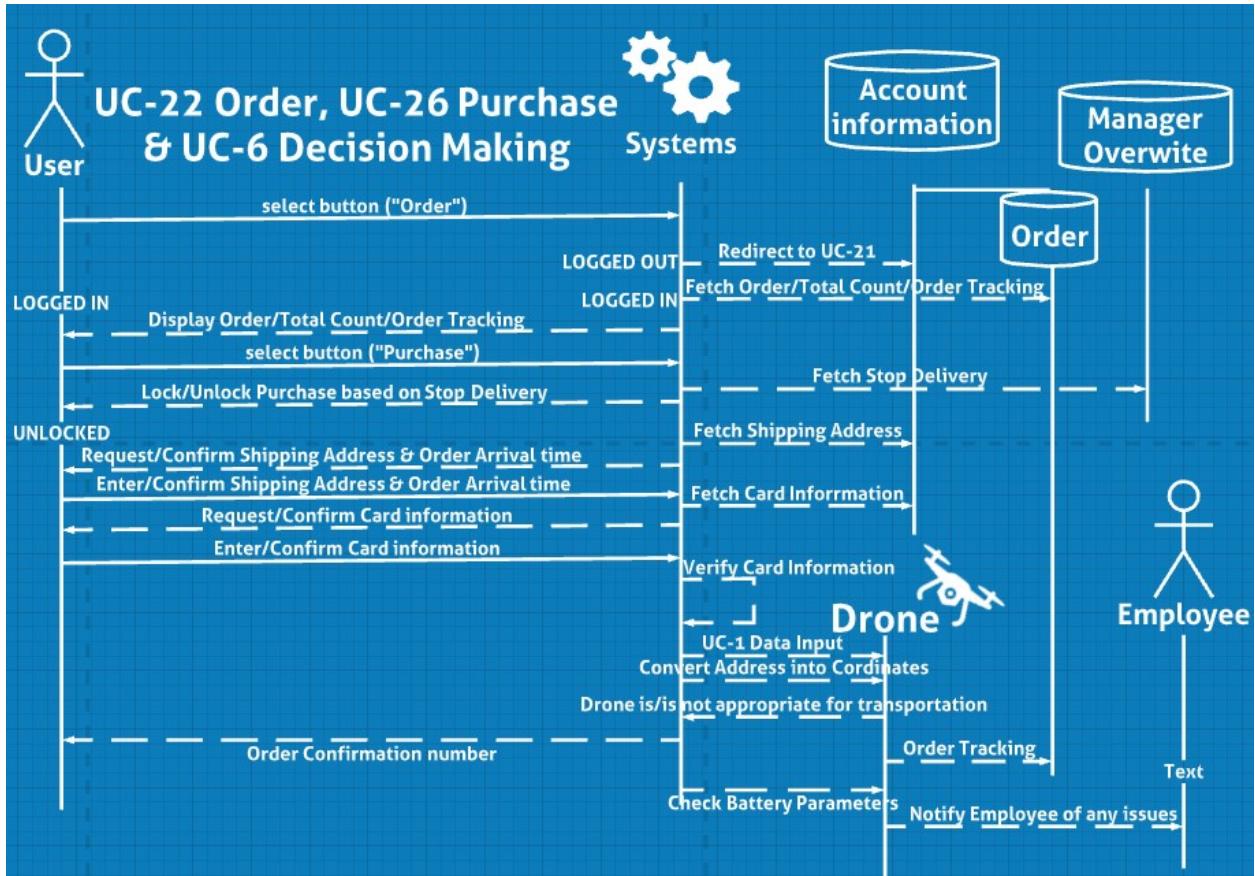
1. ← The customer clicks “customer” from the main menu
2. ← System prompts users to login with their credentials
3. → Customer inputs information
4. ← System confirms login credentials with the database
5. ← System prompts user to home screen
6. → Customer chooses “Menu” from the drop down bar
7. ← The system displays the available menu for viewing purposes

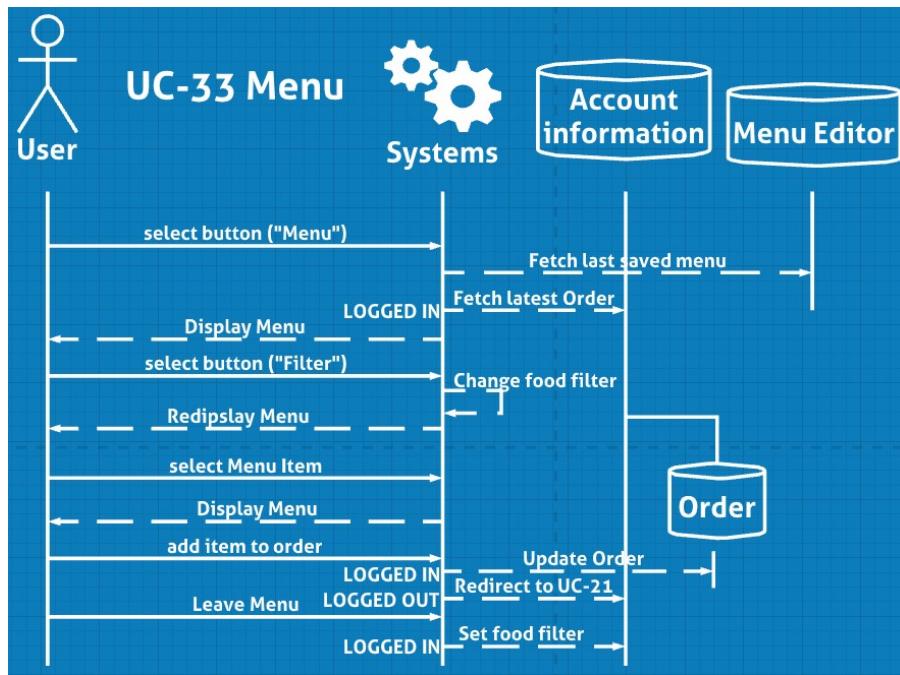
Flow of Events for Alternate Success Scenario:

1. ← The customer clicks “customer” from the main menu
2. ← System prompts users to login with their credentials
3. → Customer inputs information
4. ← System confirms login credentials with the database
5. ← System prompts user to home screen
6. → Customer chooses “Order” from the drop down bar
7. ← The system displays the available menu for ordering purposes

3.7: System Sequence Diagrams

The user is a customer in the following sys. seq. diagrams





Section 4: User Interface Specification

4.1: preliminary Design

UC-32: Login



- **Login Page**

- This page allows the user to sign in or create an account
- If the user is signing in, they can enter their email address and password then click on sign in. When the user signs in, they are prompted to the CustomerHomePage or EmployeeHomePage depending on what type of user they are.
- If the customer wants to register, the user can select "Don't Have an Account? Sign Up!" where they are prompted to the RegisterInformationPage. Here they enter their email, password, and birthday, then click on create account.

- **Customer Registration Page**

- When the customer wants to register, they select "Don't Have an Account? Sign Up!", and they are prompted to the **Customer Registration Page**. Here the customer enters their email, password, and birthday to create the account. The user then clicks on "Create Account" and is prompted to the CustomerHomePage

Customer Registration



UC: 33 - Menu



UC: 22 - Order



- **Menu**

- This Page allows user to view the menu and details on an item such as price, calories and a description
- The user arrives at the main menu of the application where they decide whether they want to sign in, register, or enter as a guest. Once an option is selected and in some cases credentials are entered, the user is prompted to the home screen.
- At the home screen the user can click on order from the drop down bar and is promoted to the menu.
- From the menu, the user may scroll through the selection of items that the restaurant has available to order.

- **Order**

- This page allows the customer to view and manage their order
- Customers access this page from the menu page shown above. The customer selects the cart button on the menu screen and is prompted to the cart.
- The customer is allowed to view their order by scrolling down and is able to toggle the switch to remove an item. The customer is also allowed to view their order and select a delivery option.
- The customer is promoted to the checkout page to enter address and other information

4.2: User Effort Estimation

CUSTOMER

Best Case Scenario

1. New customer chooses to log-in as a guest. (1 Click)
2. Chooses Menu. (1 click)
3. selects food category. (1 Click)
4. selects food item. (1 Click)
5. adds item to cart. (1 Click)
6. clicks on cart. (1 Click)
7. chooses to in store (1 Click)
8. chooses payment option as cash (1 Click)
9. selects the “No Receipt” option. (1 Click)
10. chooses not to rate their experience. (1 Click)

Total = 10 Clicks

Worst Case Scenario (note: capital letters represent constants)

1. Enters Username of ‘U’ characters (U Clicks)
2. Enters Password of ‘P’ characters (P Clicks)
3. chooses the login option (1 Click)
4. enters wrong information (repeats steps 1-3 ‘n’ times)
5. opens menu (1 click)
6. chooses appetizer category (1 click)
7. chooses appetizer (1 click)
8. skims through ‘A’ appetizers (repeats steps 7 ‘A’ times)
9. makes ‘x’ modifications to appetizer (x clicks)
10. adds app to cart. (1 Click)
11. orders for ‘m’ people (repeats steps 7,9,10 ‘m’ times)
12. chooses drink category (1 click)
13. chooses drink (1 click)
14. skims through ‘D’ drinks (repeats steps 13 D times)
15. adds drink to cart with ‘s’ specifications. (1+s Clicks)
16. chooses entree category (1 click)
17. chooses entree (1 click)
18. skims through ‘E’ entrees (repeats step 17 E times)
19. makes ‘y’ modifications to entree (y clicks)

20. adds entree to cart. (1 Click)
21. orders for 'm' people (repeat steps 12-13,15-17,19-20 'm' times)
22. Goes back to main screen. (1 Click)
23. chooses to play game (1 click)
24. chooses tutorial and continues to game after (2 clicks)
25. chooses to play 'r' rounds (r Clicks)
26. exits game to main menu (1 click)
27. clicks on cart. (1 Click)
28. chooses to in store (1 Click)
29. chooses payment option as card (1 Click)
30. chooses card number box and enters card number (1+12 Clicks)
31. chooses cvv box and enters cvv (1+3 Clicks)
32. chooses expiration box date and enters expiration date (1+4 Clicks)
33. selects the "Print Receipt" option. (1 Click)
34. chooses star rating. (1 Click)
35. chooses to leave 'c' character review. ('c' Clicks)

Total = $n*(U+P+1) + 1 + 1 + A + m*(2+x) + 1 + D + m*(2+s) + 1 + E + m*(2+y) + 1 + 4 + r + 27 + c$

= $n*(U+P+1) + m*(6 + x + s + y) + A + D + E + 36$ clicks

EMPLOYEES

Best Case Scenario

1. Enters Username of 'U' characters (U Clicks)
2. Enters Password of 'P' characters (P Clicks)
3. chooses the login option (1 Click)
4. enters wrong information (repeats steps 1-3 'n' times)
5. no tables
6. Signs out (1 click)
7. Confirms sign out (1 click)

Total = $n*(U+P+1) + 2$ clicks

Worst Case Scenarios

CHEF/BARTENDER

1. Enters Username of 'U' characters (U Clicks)

2. Enters Password of 'P' characters (P Clicks)
3. chooses the login option (1 Click)
4. enters wrong information (repeats steps 1-3 'n' times)
5. confirms completion of 'i' items (i clicks)
6. Signs out (1 click)
7. Confirms sign out (1 click)

Total = $n*(U+P+1) + i + 2$ clicks

HOST

1. Enters Username of 'U' characters (U Clicks)
2. Enters Password of 'P' characters (P Clicks)
3. chooses the login option (1 Click)
4. enters wrong information (repeats steps 1-3 'n' times)
5. chooses table to occupy (1 click)
6. confirms occupation (1 click)
7. seats 't' tables (repeats 5-6 't' times)
8. Signs out (1 click)
9. Confirms sign out (1 click)

Total = $n*(U+P+1) + 2*t + 2$ clicks

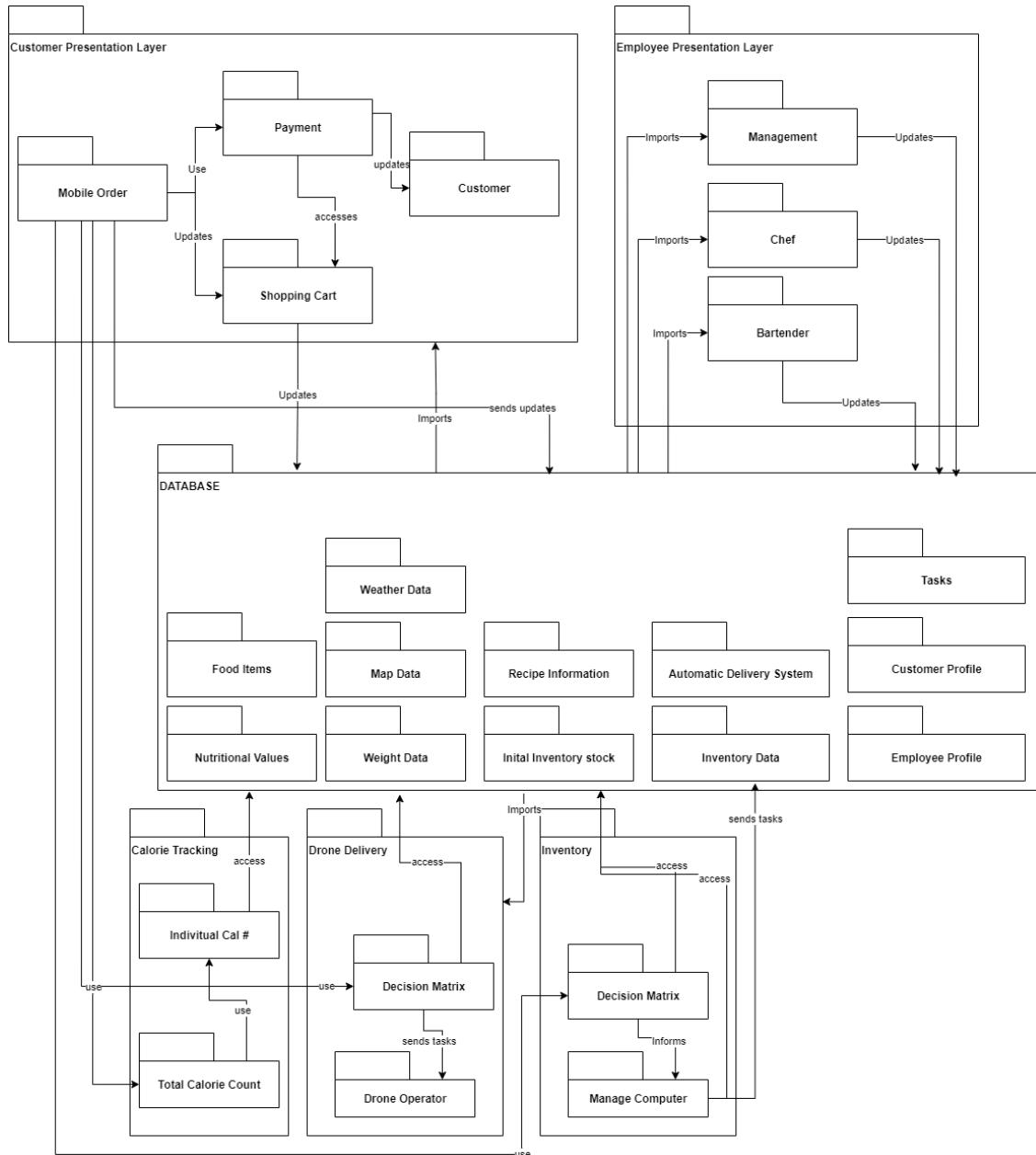
MANAGER

1. Enters Username of 'U' characters (U Clicks)
2. Enters Password of 'P' characters (P Clicks)
3. chooses the login option (1 Click)
4. enters wrong information (repeats steps 1-3 'n' times)
5. selects inventory (1 click)
6. selects ingredient (1 click)
7. enters quantity (3 clicks - Assuming can reach hundreds range)
8. updates quantity (1 click)
9. repeats for 'i' ingredients (repeats steps 5-8 'i' times)
10. Signs out (1 click)
11. Confirms sign out (1 click)

Total = $n*(U+P+1) + 6*i + 2$ clicks

Section 5: System Architecture

5.1: Identifying Subsystems



The business component in the project comes under two umbrellas.

1. Under Inventory subsystem, the manager computer is able to access the Supplier Database. Within the supplier database is the inventory data: the cost of ordering an ingredient, cost of shipping and handling, and payment method. This part of the business is the backend where it shows the ordering and handling costs.
2. Under the App subsystem, customers are able to order the food. This part of the business is the customer facing where customers can order food and use their payment method to pay for the food.
3. One last business logic is the overhead cost of running the kitchen, but none of our subsystems handle that. This part of the business logic comes under the manager/owner and their method of paying the overhead.

5.2: Architecture Styles

Our project's architectural style will be composed of a few common architectural styles. Primarily, the application will make use of the client-server model. The application server will handle data storage/requests from the many application clients, which can be customers or restaurant staff. Customers interact with the client to do things like order their food, play a game, or view the status of their delivery. Restaurant staff interact with the client to do things like clock in/out and view & manage orders. The server will receive and process all of these interactions and send back to the client any requested data. Clients can only interact with the server, and not directly to other clients.

The Super Foods application will make use of layered architecture to easily separate the features needed by customers from features needed by restaurant staff. This will simplify the design considerably and allow us to make changes to one layer without affecting the others. It also allows the different roles to interact with the application at different levels of abstraction. For example, while managers should be able to modify the restaurant menu from the menu page, other staff and customers should not see this feature in their application layer.

Lastly, the application will follow a reactive style architecture. Reactive systems are responsive (low latency), resilient (stays responsive in face of failure), elastic (stays responsive under varying workload), and message driven (pass messages asynchronously between components). The application will be composed of smaller components that will work as independently as possible in order to ensure responsiveness, resiliency, and elasticity.

5.3: Mapping Subsystems to Hardware

Our project will have two components that will run on hardware, as explained above: clients and server. The server will run on a computer, and the many clients will connect and interact with the server. The server will manage the database and its components, and process requests from the clients. The clients will run as applications on mobile devices, or as web applications. Most mobile devices will be able to use the application, and most desktop computers with a high speed internet connection should be able to run the server side application.

5.4: Connectors and Network Protocols

Our system will be having the client interact with the server through a mobile application. To accomplish this, our group is implementing the REST network protocol, Representational State Transfer. A REST API will perform actions on the server based on HTTP POST and GET requests. As described in the Architecture Styles section, the system is making use of a client-server architecture which REST complements well and makes the experience for the user as straightforward as possible. In contrast to SOAP, the other possible protocol, REST supports the data format JSON which offers better support for browser clients and is becoming more popular due to its ease of use and modern practices. For these reasons, REST is the preferred choice.

5.5: Global Control Flow

Executive Orderness: The SuperFoods main system is *event-driven*. Excluding the linear fashion of logging in, the user can take a variety of actions in whatever order they please (View menu, place an order, view order history, leave a review, play a game) but when breaking it up into subsystems, the process of ordering and the “Would You Rather?” Adult Game are *procedure driven* where the user only has a few paths to take, either continue or cancel/exit.

Time Dependency: The subsystem managing drone operation is the only subsystem that involves *time dependency*. The manager is able to set a time window for customers to place an order. When a customer includes their shipping address for a delivery order, the system will also prompt a request for the order arrival time. If the order arrival time exists within the time window set by the manager, then the system needs to calculate how long it will take for the kitchen to finish preparing the food and for the drone to deliver it. Subtract the order arrival time by those values and the system will know when to prompt the kitchen to start preparing the food and allocate a drone for the specific time of the delivery. Implementing time-dependency into the order and drone subsystems is paramount for a restaurant to deliver food effectively.

The “Would You Rather?” Adult Game subsystems will also be *time dependent*, constrained by another time window set by a manager. This is important when the restaurant is held up from closing because customers are playing the game and a manager is unavailable to turn the game off right away, the manager can be proactive by setting a time for the game to be locked from users automatically. Whenever a customer presses the next button (UC-10) the system will compare the real time with the manager's time window. If it is outside the window, the game will redirect to the main menu. At every entrance to the main menu, the system will also disable the game button if the real time is outside the time window.

Concurrency: We will utilize multiple threads to account for each individual order. Since there will be different threads handling different operations on our server possibly at the same time, multithreading will make sure no orders are lost and their dependencies will be accounted for.

5.6: Hardware Requirements

Our SuperFoods application will be able to run on mobile devices such as smartphones and tablets that are supported by a network connection via Wifi, ethernet, or Mobile Data. The layout of the application will be identical for all device types and models. The application is not intended to be used on a computer, so there is no developed client or layout for laptops or desktops. The application will be compatible with the android operating system and able to run on a range of phones such as the Google Pixel and Nexus phones. A minimum storage capacity of 100 mb is required. Our application would require mobile devices to be at least operating in the android operating system “Android Pie”. Also to be able to run our application a minimum of a 5.6” phone screen is required with a resolution of 1080x 2220. A minimum recommended internet connection is 2.5 Mbps to run SuperFoods smoothly. For phone specifications we would require the phone to at least Qualcomm Snapdragon 835 processor, the phone to be equipped with a gyrometer and accelerometer, and wireless audio capabilities.

Section 6: Project Size Estimation

6.1: Unadjusted Actor Weight

Actor Type	How to recognize the Actor Type	Weight
Simple	The actor is interacting through the defined API (application programming interface)	1
Average	The actor is a person interacting through a protocol or through a test based interface	2
Complex	The actor is a person interacting with the user interface	3

Actor Name	Description of Relevant Characteristics	Complexity	Weight
Customer	Customer is interacting with the user interface	Complex	3
Guest	Guest is interacting with the user interface	Complex	3
Employee	Employee is interacting with the user interface	Complex	3
Manager	Manager is interacting with the user interface	Complex	3
System	System is interacting through a protocol	Average	2
Busboy	Busboy is interacting with the user interface	Complex	3
Chef	Chef is acting with the user interface	Complex	3
Database	Database is a system interacting	Average	2
Host/Hostess	Host/Hostess is interacting with the user interface	Complex	3
Servers	Server is interacting with the user interface	Complex	3
Drone Operator	Drone Operator is interaction with the user interface	Complex	3

$$\text{UAW}(\text{SuperFoods}) = 2 * \text{Average} + 9 * \text{Complex} = 2(2) + 9(3) = 31$$

6.2: Unadjusted Use Case Weight

Actor Type	How to recognize the Actor Type	Weight
Simple	- Simple user interface - Only one participating actor (and initiating actor) - Number of steps for the success scenario: ≤ 3	5
Average	- Moderate interface design - Two or more participating actors - Number of steps for the success: $< 4, < 7$	10
Complex	- Complex user interface or processing - Three or more participating actors - Number of steps for the success scenario: ≥ 7	15

Use Case	Description	Category	Weight
UC-1: Data Input	Simple user interface. 1 step for the main success scenario. 2 participating actors (Customer, Database).	Simple	5
UC-2: Check Battery Level	Simple user interface. 1 step for the main success scenario. 2 participating actors (Employee, Database).	Simple	5
UC-3: Drone Tracking	Moderate interface design. 4 steps for the main success scenario. 2 participating actors (Drone Operator, System).	Average	10
UC-4: Track Emergencies	Moderate interface design. 3 steps for the main success scenario. 2 participating actors (Drone Operator, System).	Average	10
UC-5: Order Tracking	Moderate user interface. 2 steps for the main success scenario. 2 participating actors (Customer, Database).	Average	10
UC-6: Decision Making	Moderate interface design. 6 steps for the main success scenario. 1 participating actor (Database).	Average	10
UC-7: Start the Game	Simple user interface. 1 step for the main success scenario. 2 participating actors (Customer, Database).	Simple	5
UC-8: Tutorial	Moderate interface design. 5 steps for the main success scenario. Two participating actors (Customer, Database).	Average	10

UC-9: Play	Simple user interface. 1 step for the main success scenario. 2 participating actors (Customer, Database).	Simple	5
UC-10: Next	Simple user interface. 1 step for the main success scenario. 2 participating actors (Customer, Database).	Simple	5
UC-11: End	Simple user interface. 1 step for the main success scenario. 2 participating actors (Customer, Database).	Simple	5
UC-12: Shipment Deliveries	Moderate interface design. 5 steps for the main success scenario. 2 participating actors (Manager, Database).	Average	10
UC-13: View Inventory Count	Simple user interface. 1 step for the main success scenario. 2 participating actors (Employee, Database).	Simple	5
UC-14: Inventory Display	Simple user interface. 4 steps for the main success scenario. 2 participating actors (Employee, Database).	Average	10
UC-15: Low Stock Alert	Simple user interface. 2 steps for the main success scenario. 2 participating actors (Manager, Database).	Simple	5
UC-16: Displaying Menu Item	Simple user interface. 1 step for the main success scenario. 2 participating actors (Customer, Database).	Simple	5
UC-17: Display Total Count	Simple user interface. 1 step for the main success scenario. 2 participating actors (Customer, Database).	Simple	5
UC-18: Switch Menu	Simple user interface. 2 steps for the main success scenario. 2 participating actors (Customer, Database).	Simple	5
UC-19: Add New Item	Simple user interface. 4 steps for the main success scenario. 2 participating actors (Manager, Database).	Average	10
UC-20: Display Food Filters	Simple user interface. 8 steps for the main success scenario. 2 participating actors (Customer, Database).	Average	10
UC-21: Account Creation	Simple user interface. 4 steps for the main success scenario. 2 participating actors (Employee, Database).	Simple	5

UC-22: Order Display	Complex user interface. 14 steps for the main success. 4 participating actors (Database, Customer, Manager, Employee).	Complex	15
UC-23: Reward Display	Simple user interface. 2 steps for the main success scenario. 2 participating actors (Database, Customer).	Simple	5
UC-24: Point Redemption System	Simple user interface. 2 steps for the main success scenario. 2 participating actors (Database, Customer).	Simple	5
UC-25: Menu Editor	Simple user interface. 4 steps for the main success scenario. 2 participating actors (Manager, Database).	Average	10
UC-26: History Display	Simple user interface. 2 steps for the main success scenario. Two participating actors (Customer, Database).	Simple	5
UC-27: Review Menu	Simple user interface. 1 step for the main success scenario. 2 participating actors (Customer, Database).	Simple	5
UC-28: Purchase Menu	Simple user interface. 4 steps for the main success scenario. 2 participating actors (Customer, Database).	Average	10
UC-29: Stop Delivery	Simple user interface. 2 steps for the main success scenario. 2 participating actors (Manager, Database).	Simple	5
UC-30: Employee Creation	Simple user interface. 4 steps for the main success scenario. 2 participating actors (Employee, Database).	Simple	5
UC-31: Tables Display	Simple user interface. 2 steps for the main success scenario. 2 participating actors (Server, Database).	Simple	5
UC-32: Customer and Employee Login	Simple user interface. 4 steps for the main success scenario. 2 participating actors (Customer/Employee, Database).	Simple	5
UC-33: Access Menu	Simple user interface. 5 steps for the main success scenario. 2 participating actors (Customer, Database)	Average	5

UUCW(SuperFoods) =

$$20 * \text{Simple} + 12 * \text{Average} + 1 * \text{Complex} = 20 * 5 + 12 * 10 + 1 * 15 = 235$$

6.3: Technical Complexity Factors

Technical Factor	Description	Weight	Perceived Complexity	Calculated Factor (Weight * Perceived Complexity)
TC1	Users expect good performance, nothing exceptional	1	3	$1 * 3 = 3$
TC2	End-user expects efficiency; no exceptional demands	1	3	$1 * 3 = 3$
TC3	Ease is very important to every user	0.5	5	$0.5 * 5 = 2.5$
TC4	Easy to make changes	1	3	$1 * 3 = 3$
TC5	No unique training needed	1	0	$1 * 0 = 0$
TC6	Direct access for third parties	1	5	$1 * 5 = 5$
TC7	No requirement for reusability	1	0	$1 * 0 = 0$
TC8	Concurrent use is required	1	4	$1 * 4 = 4$
Technical Factor Total:				20.5

$$TCF = 0.6 + 0.01 * \text{Technical Factor Total} = 0.6 + 0.01 * 20.5 = 0.805$$

6.4: Use Case Points

$$UCP = UUCP * TCF * ECF$$

$$UUCP = UAW + UUCW = 31 + 235 = 266$$

$$TCF = 0.805$$

$$ECF = 1$$

$$UCP = 266 * 0.805 * 1 = 214.13 = 214 \text{ use case points}$$

Section 7: Plan of Work

	Demo 1						Demo 2					
Group	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
	2/21 - 2/27	2/28 - 3/6	3/7 - 3/13	3/14 - 3/20	3/21 - 3/27	3/28 - 4/3	4/4 - 4/10	4/11 - 4/17	4/18 - 4/24	4/25 - 5/1	5/2 - 5/8	5/9 - 5/15
Group 1	UC-6	UC-6	UC-6	UC-6	UC-6	Demo Week	UC-6	UC-3	UC-1	UC-1	UC-2	Demo Week
		UC-1	UC-1	UC-1	UC-4			UC-4			UC-4	
								UC-5				
Group 2	UC-32	UC-10	UC-33	UC-22	UC-28	Demo Week	UC-21	UC-25	UC-31	UC-23	UC-27	Demo Week
	UC-7	UC-11					UC-30	UC-29		UC-24	UC-28	
	UC-9	UC-8										
Group 3	UC-13	UC-13	UC-13	UC-14	UC-14	Demo Week	UC-15	UC-15	UC-15	UC-12	UC-12	Demo Week
		UC-14	UC-14						UC-12			
Group 4	UC-16	UC-16	UC-16	UC-19	UC-16	Demo Week	UC-16	UC-20	UC-18	UC-19	UC-16	Demo Week
			UC-17		UC-17		UC-19				UC-17	

Drones, Inventory, Calories: Figure out a Menu

Group 1(Drone):

- I. Week of 2/21
 - A. Plan out decision matrix
- II. Week of 2/28
 - A. Code out decision matrix with manual test conditions
- III. Week of 3/7
 - A. Create GUI
- IV. Week of 3/14
 - A. Distance Geocoding integration(google api or texas a&m)
- V. Week of 3/21
 - A. Weather data integration
- VI. Week of 3/28
 - A. Demo Week
- VII. Week of 4/4
 - A. Food Weight integration
- VIII. Week of 4/11
 - A. Test System
- IX. Week of 4/18
 - A. User interaction with the system
- X. Week of 4/25
 - A. Employee interaction with the system
- XI. Week of 5/2
 - A. Final system testing
- XII. Week of 5/9
 - A. Demo Week

Group 2 (Games and Enhancements):

Week of 2/21

- A. Creating the login such that the user can login and use the features
- B. Creating a majority of the "Would You Rather" game such that it is playable

Week of 2/28

- A. Finish working on the game

Week of 3/7

- A. Creating the item Menu

Week of 3/14

- A. Creating the order

Week of 3/21

- A. Creating the Purchase Process

Week of 3/28

- A. Demo Week

Week of 4/4

- A. Creating the Account creation for customer and employees

Week of 4/11

- A. Creating the Menu Editor and Stop Delivery
- Week of 4/18
 - A. Creating the Table
- Week of 4/25
 - A. Create the reward Display where customers can see can earn reports
 - B. Create the rewards redemption system
- Week of 5/2
 - A. Creating the interface that allows the customer to view order history
 - B. Creating the interface that allows the customer to leave review
- Week of 5/9
 - A. Demo Week

Group 3(Inventory):

- I. Week of 2/21
 - A. Plan out decision matrix
- II. Week of 2/28
 - A. Create counter for restaurant ingredients
- III. Week of 3/7
 - A. Create a way of adding more ingredients to the counter
- IV. Week of 3/14
 - A. Create a display to display counter
- V. Week of 3/21
 - A. Add color coding for display
- VI. Week of 3/28
 - A. Demo Week
- VII. Week of 4/4
 - A. Plan out Manager Database and how it interacts with the rest of the system
- VIII. Week of 4/11
 - A. Create a system that automatically subtract ingredients from the database depending on the items cooked during the day
- IX. Week of 4/18
 - A. Create an alert system for the Manager to be reminded about low ingredients
- X. Week of 4/25
 - A. Create a way for the manager to order straight from the supplier
- XI. Week of 5/2
 - A. Testing the system
- XII. Week of 5/9
 - A. Demo Week

Group 4(Calories):

- I. Week of 2/21
 - A. Learn MySQL/ SQL(database programming language)

- II. Week of 2/28
 - A. Create database of possible ingredients with nutritional information
- III. Week of 3/7
 - A. Coding functions to access individual calories and calculate total calories
- IV. Week of 3/14
 - A. Coding manager computer function to add/update inventory
- V. Week of 3/21
 - A. Polish/review
- VI. Week of 3/28
 - A. Demo Week
- VII. Week of 4/4
 - A. Update interface with new menu items and their nutritional information
- VIII. Week of 4/11
 - A. Create function to remove foods containing dietary restrictions/allergies
- IX. Week of 4/18
 - A. Create toggle button to switch between menu with/out calories
- X. Week of 4/25
 - A. Confirm system compatibility and reduce system delay in displaying new Menu items
- XI. Week of 5/2
 - A. Polishing and Testing the system
- XII. Week of 5/9
 - A. Demo Week

Individual Contributions Breakdown for Report

Project Management

The group coordinated with each other and set meetings that accommodated everyone's schedule. Websites like when2meet provided a method to apply everyone's schedule for a set amount of days onto a table format. This allowed for easy selection of a timeframe with all members. Applications like discord are used to have project-related conversations divided by sections such as groups, ideas, and announcements. Discord also allowed us to have a platform to conduct meetings through voice and video calls. Meetings are conducted once a week with everyone to coordinate the direction of where the project is heading. Meetings are also conducted between respective groups as needed by members. Documentations for each meeting and reports are stored onto Google Docs, separated into their respective folders such as by group number, large meetings, and reports.

The report was done equally as a collective group. A meeting is scheduled such that every member of the project meets on Discord and completes the report together. Each subproblem group is in charge of describing and writing their system requirements. The user interface is completed together as a project team to incorporate everyone's ideas. This gives us a foundation such that no conflicting problems occur when all subproblems are brought together. Upon completion of the report, everyone conducts a review of the report and provides comments. The comments are taken into consideration and incorporated into the final product. A final review of the report is conducted and later on submitted.

Meetings are objective oriented such that they can be as productive as possible. During our initial week, we were able to create the basic framework of our project design. Sub groups were created and members were assigned to each group based on member's interest. Meetings are conducted between each sub group and each group set objectives for themselves. Documentation for each meeting conducted for each sub group is stored in goal drive and contains the progress and status of the group. These documentations can be viewed by all members so they can view the activities of the other groups. Code created by the group will be stored in GitHub such that everyone has access to it.

Sub-Groups	Functionalities	Members	Skills
Group 1	Drone/Driven Delivery System	Keith Lo	C++, Python, Java, Solidworks, Impact
		Guilherme Silva	C++, Python, Java
Group 2	Tablet Games Turbo-Yums Enchantments	Rawad Sayah	C, Java, UX/I Design, Gamer
		Gian-Soren Morici	Linux, C++, Management, Creativity4

		Brianna Solano Aguilar	C++, Java, SolidWorks, Arduino, Raspberry Pi
Group 3	Inventory	Saurabh Bansal	C++, Python, Java
		Robert Kulesa	Linux, C, Java, Python, Git/Version Control
		Lindsay Wisner	C++, Design, AutoCAD, SolidWorks, Team Management
Group 4	Calories/Health Data	William Basanaga	C++, Java, Python
		Jeremy Kim	Java, C++, AutoCad
		Justin Chan	C++, Java

References

- TurboYums Project
<https://turboyums.github.io/SEWebsite>
- The Coding Train
Author: The Coding Train
Title: 8.1: What is HTML? - p5.js Tutorial
Link:
<https://www.youtube.com/watch?v=URSH0QpxKo8&list=PLRqwX-V7Uu6bl1SlcCRfLH79HZrFAtBvX&index=1>
- The Reactive Manifesto
Author: Jonas Bonér, Dave Farley, Roland Kuhn, and Martin Thompson.
Title: The Reactive Manifesto
Link: <https://www.reactivemanifesto.org/>