WHY WHINE JUST DINE

Jason Chan
Ashirvadh Bhupathi
David Joseph
Hojun Son
Josh Chopra
Krishna Patel
Juergen Benitez
Justin Davis
Pavan Kunigiri
Ryan Van Duren

Individual contributions breakdown

Jason Chan:

- **Problem Statement- 33%**
  - Statements for Host, Waiter, Chef
- **Decomposition into Subproblems- 90%**
  - Almost all if not all, as well as changes based on Prof's and TA;s concerns.
- **User Interface Requirements- 5%**
  - Created Diagrams for "Chef", as well as table of requirements
  - Helped edit requirements for other employee subgroups
- **Project Management- 23%**
  - Recorded the work already completed in the Gantt Table, as well as help plan the roadmap for Report 2
- **Communication- 11%**
  - Helped start and lead meetings
- **Actors and Goals- 25%**
  - Helped fill out descriptions for some of the actors and goals
- **Traceability Matrix- 25%**
  - Helped create matrix and fill out the data
- **Fully Dressed Description- 33%**
  - Wrote description for UC-9
- **User Effort Estimation- 50%**
  - Wrote user scenario 2.
- **Feedback2 edits- 14%**
  - Met up with group to discuss and edit Report based off Feedback#2
- **Mapping Subsystems to Hardware- 100%**
  - Nothing much, just a small statement.
- **Project size estimation- 20%**
  - Wrote the complexities for TCF
  - Filled out description and complexity for various actors for UUCW
- **Feedback edit 3**

David Joseph

- **Glossary Of Terms - 70%**
  - All the definitions from "Android" to "Menu" and "Software Engineer" to "Waiter/Host"
  - Also added definitions for "Rewards/Discounts", "Cook Time", and "Dish Popularity"
- **User Interface Requirements - 85%**
  - Created diagrams for the "Login screen", "Owners Screens", and "Customers Screens"
  - Also created diagrams for "Waiter/Host" and "Busboy" under the employees section.
  - Worked on the Chef REQ table

- **Project Management - 1%**
    - Helped a little with creating the sheet, adding and cleaning up rows/columns.
- **Communication - 11%**
    - Helped find times to meet with most of us. Consistently meets in weekly meetings
- **Stakeholders - 100%**
- **Actors and Goals - 70%**
    - Finished most, but not all of them.
- **Traceability Matrix - 25%**
    - Helped create matrix and fill out data
- **Fully Dressed Description - 33%**
    - Worked on UC-#9
- **Feedback #2 - 14%**
    - Worked on #3
- **Identifying Subsystems - 100%**
- **Architecture Styles - 100%**
- **Feedback edit 3 - Case diagram chart - 100%**

Ashirvadh Bhupathi

- **Problem Statement - 33%**
    - Statement for Manager and Driver
- **User Interface Requirements - 5%**
    - Created diagrams for the drivers
    - Did the UI requirements that pertain to the drivers
- **Project Management - 23%**
    - Planned the work needed to be done on Report 1 part 2 and Part 3 using the Gantt Table
- **UI Diagrams - 50%**
    - Worked on UC-1 and completed UC-3
- **Fully Dressed Description**
    - Fleshed out some alternatives success scenarios
- **Project size estimation based on use case points - 25%**
    - Completed the Unadjusted weight

Juergen Benitez:

- Made edits to Manager Section of **Problem Statement**
- **Glossary of Terms**- 30%
- **Enumerated Functional Requirements**-20%
- **Project Management(Coding plan**)-30%
- **Use cases** -33%
- **Traceability Matrix** -25%
- **User Effort Estimation** -50%
- **Feedback 2 Edits** -14%

- Worked on **Project Size Estimation-40%**

Ryan Van Duren

- **Decomposition into subgroups - 100%**
    - Edited “Customer” subgroup to answer the questions posed by the TA
- **Business Goals - 100%**
    - Rewrote the Business Goals section to accurately reflect the changes made to the project.
- **System Sequence Diagrams - 50%**
    - Created for UC-10 and created UC-6 with Pavan.
- **Hardware Requirements - 100%**
    - Wrote the Hardware requirements section

Hojun Son

- **Enumerated Nonfunctional Requirements - 100%**
    - Created a table of nonfunctional requirements with priority and description for each of goals.
- **User Interface Requirements - 5%**
    - Created tables for Manager, Waiter, and Busboy.
- **Actors and Goals - 5%**
    - Helped fill out the actors and goals description.
- **Traceability Matrix- 25%**
    - Helped fill out the traceability matrix.
- **Fully Dressed Description - 33%**
    - Helped fill out a description of UC-2.
- **Feedback 2 Edits - 14%**
    - Helped edit the report based on feedback #2.
- **Project Size Estimation - 15%**
    - Calculated each technical factor value and the TCF value.

Pavan Kunigiri

- **Enumerated Functional Requirements - 80%**
    - Created Key and Table. Wrote down the description for most of the requirements. Ranked requirements for priority.
- **Project Management - 23%**
    - Recommended Gantt Chart template used and edited document with necessary information.
- **Decomposition into Subproblems - 5%**
    - Added detail to subproblems to address problems TA addressed in his email.
- **Feedback 2 - 14%**
    - Edited requirements.
- **System Sequence Diagrams - 50%**
    - Created diagram for UC-6. Created diagram for UC-5 with Ryan.

- **Global Control Flow - 100%**

Krishna Patel

- **Problem Statement - 33%**
  - Statement for Customer and Busboy
- **Cover Page - 10 %**
- **Communication - 10%**
- **Use Cases - 33%**
- **Diagram - 100%**
- **Feedback 2 edits - 14%**
- **Network Protocol - 50%**

Josh Chopra

- **Cover Page - 90%**
- **Use Cases - 33%**
- **Network Protocol - 50 %**
- **Feedback 2 edits - 14%**

Justin Davis

- **Communication - 11%**
- **Software Choices**
  - I developed the idea to use Android and Java programming language
- **Research**
  - Helped research past projects in determining what others have done to improve upon on our own
- **UI Diagrams - 50%**
  - Did half of UC-1 and all of UC-2
- **Fully Dressed Descriptions - 5%**
  - Added an alternate success scenario for UC-1

Table of Contents

Section 1: Customer Problem Statement

1a.Problem Statement

Actor - Host/Hostess:
As a host, my job is to make sure that customers are as comfortable as possible throughout the entire process of dining, or picking up takeout orders. Being one of the first impressions for customers when they enter the restaurant, it is important that I am able to conduct this process as smoothly as possible. However, especially during the times of day where we have a large number of customers flowing in and out of our restaurant, it can be quite difficult to keep track of everything. Currently, as a host I have to keep track of which tables are free, which tables are clean, which tables are reserved, and the location of all these said tables. Furthermore, as a host, another part of my job consists of responding to customer requests timely and efficiently. However, while doing other parts of my job such as welcoming in guests from the front door as well as seating them, sometimes I might not notice a table trying to get my attention. This then can lead to a bit of paranoia as I would have to constantly check on tables to make sure everything is going smoothly. Thus, it would be awesome if there was a program that allowed me to get notified when a customer requests for my presence at their table, and which table I would then have to attend. Furthermore, I would like the program to be able to keep track of all the tables, so it would be much easier to have the tables cleaned, seat the customers, etc. This would allow me to be much more efficient with my work, as I would spend a lot less time seating customers and attending to their requests.
Application Solution: Our application allows the host to see which tables are open, occupied, dirty, and reserved, as well as their corresponding location within the restaurant. The interface will essentially have a status for the table that is able to be changed by the host when a table is reserved, freed, or occupied. Furthermore, through the customer's ordering interface, they will

be able to alert the host when they require their presence. The interface on the host's end will notify him that his presence is requested, and from which table.Furthermore, to make it more efficient, during closing time, the host will be able to clean-reset the table statuses to prepare it for the next day.

Actor - Driver

As a driver my job is to deliver food in a timely manner so the food will be fresh when the customer eats the food. To do this I need to be notified immediately when the food is ready so I can promptly pick up the food from the kitchen and deliver it to the customer. I would also like detailed information on how to contact the customer in case there is a problem. Information like their address and phone number is necessary for me to do my job in an efficient way.

Application Solution: The application will require the chef to mark when the food is finished cooking. This information will then be sent to the driver so they know when to come pick up the food. Any customer who asks for delivery will need to prove their name. Address and phone number. This way the driver will always be able to contact the customer if any problems arise like for example need any help with directions. The driver will input, using the information from whatever navigation software they are using, how long it will take to deliver the food. The customer will also have the option to leave any additional notes in the app that will notify the driver of any specific instructions like leave the food in front of the door.

Actor - Manager:

As a manager the thing I value the most is efficiency. I'm always looking for opportunities to get rid of wastage of resources. A restaurant like ours has a lot of moving parts which at times can get quite chaotic. I would love to have an app that would make my life easier in keeping track of everything that is going on.

One thing I would love to improve in our business would be delivery. With the rise in Covid cases, there has been a greater demand for customers who eat from home. However, delivery from my point of view can be very expensive. I have to hire extra people to do the deliveries and any mistake that occurs can be very difficult to fix. For example if we mess up an order, normally we can quickly fix it in the restaurant. However, due to the extra time it would take in the case of delivery it would leave the customer extremely unsatisfied and frustrated. I would like the app you create to help minimize these mistakes. The app would need to keep track and give me the status of all orders at different stages of completion. Specifically I would want to know if the chef finished cooking the order and at what time it was delivered.

I manage a lot of employees and with the increase in demand for delivery that number will keep going up. Because of this I would like the app to give me the ability to track when they have clocked in and out, as well as keep track when each employee is scheduled to work in the upcoming week. The app should also automatically calculate how much money each employee has made.

I would also like the app to have bookkeeping software and have general data analysis. These features will help me assess how the day to day business is going as well as let me know what products are performing well or doing poorly. I will then use this information to make decisions on where to best invest my resources. To be more specific, I want the app to track which orders have been extremely popular as well as how much money was gained from them. Foot traffic on different days would also be very useful information.

Application Solution: The application will keep track of all orders done by the customer and relay it directly to the chef. The chef will then indicate whenever an order is finished so either the waiter can come to get the food to the customer or the delivery person can come to deliver the food. The delivery driver will need to mark on the app when the order has been delivered. The application will give the manager data on how many hours each employee has clocked in any given day as well give them the option of seeing future schedules of each employee. This should make it easier for the manager to manage their employees schedules.

For the data analysis, the app will keep track of the popularity of food and drinks in the menu. This will be defined by the number of times an item is ordered during the entire day and it will be shown in the form of a graph. The y axis will be the number of orders and the x axis will be defined by the hour of the day. The app will give more options such as seeing the number of orders by day of the week , or even by month. The use of graphs will make it easier to see trends

The application will keep track of all financial transactions that occur. The app will then use this information to produce graphs that will make it easier for the manager to see how well the business is doing. With this information the manager will be able to make better informed financial decisions.

Actor - Customer

As a customer, I expect my food and the service to be perfect. I should be getting what I paid for without any issues. When ordering food I want to know what exactly is in the food incase of allergies and for calorie counting purposes. I really don’t want to wait for the waiter to give me menus to order, I just want to explore my options before getting seated. A digital menu would be helpful, saving me the time and hassle of talking to the waiter and there being communication issues. If I had the ability to order ahead of time allowing me to cut down my wait time for my food would be fantastic. Along with ordering on my phone, if there was a way for me to know how long my food will take would be amazing, therefore if I order for take out, I know exactly when my hot food will be ready for me to pick up and enjoy. If I wanted to enjoy a nice sit down dinner at the restaurant instead of a take out order, I do not want to have long wait times and wait for hours. If I could reserve a table right on my phone and know exactly when the table will be ready, I would be able to save time and be able to do what I want until the table is ready instead of just waiting at the restaurant for an hour just to be seated. After being seated, I hate waving down waiters for help. I wish there was a way to have two buttons on my table, one to call over the waiter and another to call over the hostess incase of table issues. This would save me the time of waiting for the waiter/hostess to walk past me, instead they should be notified when I need help and then they will be able to come over to me. Due to the COVID-19 pandemic I do not always feel comfortable going into the restaurant to eat and sometimes I just want the food to come to me. If there was a delivery option right from an app where I can place my order and then track the estimated time of arrival as well as track where my delivery driver is would be very helpful.

Application solution: Our application will offer a digital touch menu with features to ensure the highest customer satisfaction. The menu will display what ingredients are in each dish ensuring all allergy concerns are accounted for. Our application will allow customers to select a table to reserve as well as let customers know about wait time so there is no uncertainty. If ordering for delivery the customer will be able to track their order, when the food is ready and when the food is on the way. They will also be able to track the driver that is delivering our food, where they are and when their food will be delivered. Being able to track your food allows for the customer to be more efficient with their time, allowing them to do whatever they want instead of just waiting for the driver to come with their food. Our application will also let customers reserve tables from their phones before coming to the restaurant. This helps them manage their time better instead of coming to the restaurant and then waiting an hour for their table after arriving. Reserving a table beforehand and the ability to track when their food will be ready will help customers be more efficient with their time.

Actor - Waiter/Waitress: I get scolded all the time by the chefs and managers for not being perfect at my job, yet I have to juggle so many things at a time. Despite this, they still expect me to perform all my duties perfectly and efficiently. I have to attend to customers, take their orders, fetch food from the kitchen, deliver orders to their corresponding table, and collect the bills from the customers. I always have to keep checking on orders in the kitchen to see if they are ready, and if i'm even a bit late the chef gets mad at me, while the customers are unhappy their order is taking a bit long to come out. Furthermore, I have to remember which orders are for which table, as well as which tables have been brought their food or not. Furthermore, I have to remember which tables have paid their bills or not, as well as paying attention to any waving hands stuck in my face to get my attention. Something that tells me whether an order is complete in the kitchen, as well as estimated time of completion would save a lot of trips to the kitchen checking if an order is ready. Furthermore, if there was another way for the customer to notify me that they are ready to order or if they need anything else, that would be awesome.
Application solution: This application will allow the customer the option to self-order using a digital menu, or to call the waiter to order. This saves the waiter as they will know when the table is ready to order, as well as some customers ordering for themselves. Furthermore, there will be a button on the customers interface that will alert the waiter. On the waiter's end of the interface, they will be quickly notified that a table requests for them, and which table they are seated at. Furthermore, all the orders will digitally be sent to the kitchen, saving the waiter a trip. The waiter is given an estimated completion time with each order, as well as which table the order is coming from. Once the order is complete, the waiter will be notified and they can then proceed to deliver the food to the corresponding table. Furthermore, the waiter can also see which tables are ready to pay, have paid already, or haven't paid yet.

Actor - Chef:
As a chef, I take pride in the food that I prepare for our customers. However, one thing that troubles me is how messy orders can be especially during busy hours. A lot of times orders come in with special modifications requested, and when multiple orders for the same dish come in but with different modifications, it can be confusing sometimes. Quite frankly it's hard to keep track of everything especially when I am running around the kitchen preparing the food.

Furthermore, sometimes the orders that the waiters write down are hardly legible. If I had some way to be able to have these orders more organized, as well as any customizations clear, that would be much appreciated. Another Issue that I have in the kitchen is sometimes the waiters take way too long to come in to get the food. Not only does it take up space on the countertop, and makes things cluttered, but it also causes the food to get cold. I'm a big advocate of serving food while it's hot, and I dislike it when those waiters are so slow to deliver the food. I don't entirely blame them as they are also very busy, but I would like a way to be able to notify the waiters that an order is ready so they can get it ASAP.
Application Solution: The orders will digitally be displayed in neat order for the chef to read easily. Furthermore, the chef will be able to notify the waiters when the food is ready, as well as an estimated time of when they think food will be ready once they start an order. Furthermore, and modifications to orders will be highlighted so the chef can see it clearly.

Actor - Busboy:

Being a busboy is not always the hardest job but there are always those days that can end up being stressful. The busy days where we have a lot of customers coming in and out are the worst. The hostess is relying on me to let them know when the tables are cleaned and ready for the next party to be seated. If there was an option in the application that shows which tables have been cleared off and cleaned, it would be a more efficient way to let the hostess know when the tables are ready for the next guest to enjoy.

Application Solution: Our application will offer an interactive display which will keep track of every step along the way of a successful dining experience. The busboys will be informed when a customer has left the restaurant, which will let the busboys know to clean those certain tables. Once the busboys clean the table they can update the status of the table on the application which will then let the hostess know that the table is clean and available for the next party to be seated.

1b. Decomposition into Subproblems

- Subproblem 1: Customer
    - Ordering Interface
        - Customers should be able to order food from the menu, and customize their orders.
        - Account login system where they can get rewards/discounts for coming frequently. (guest system also will be implemented)
        - Takeout/Delivery/Dine in options
    - Food tracking interface(Simplified Version)
        - Able to see when their order is being started as well as cook time, and estimated time of completion
    - Delivery tracking interface(Simplified Version)
        - Able to track when the order has left the restaurant, and estimated time of arrival.
    - Data Analysis(recommended food items)

        - Suggested food items based off popular menu choices
    - Table Occupancy interface(Availability of Table)
        - Able to see how many tables are available with X amount of seats per table.
- Subproblem 2: Manager
    - Delivery/food tracking interface(Based off completion)
        - Manager can see when an order has completed and when a delivery has been completed
        - Has admin options if he wants to see more details that the waiter and chef see.
            - Add or remove items from the menu
            - Estimated food completion time
            - Whether an order has been started
    - Table Occupancy interface(shared)
        - Able to see how many tables are available with X amount of seats per table.
        - (Admin): can update table status to any status.
    - Finance tracking(Full Version)
        - Keeps track of all the finances throughout the day, as well as bookkeeping
        - Able to see when a table has paid and how much their order was/receipts.
    - Data Analysis (Full Version)
        - Data:Food ordered by time, throughout the day/ Food ordered by day of the week.
            - A bar graph will be produced.
            - Divided by time of day and day of the week
        - Data: Number of customers at various times throughout the day/week
    - Clocking hours (Full Version)
        - Able to see which employee has clocked in at what time and how long they have been working, as well as normal shift hours.
- Subproblem 3: Employee (Waiter/Chef/Driver/Busboy)
    - Table Occupancy interface(shared)
        - Able to see which tables are occupied/not, which tables are dirty and clean.
        - Waiters: Can update table status as dirty
        - Host: Can update table status as occupied
        - Busboy: Can update table status as Clean
    - Food tracking interface(Full version)
        - Able to track all the food orders(whether they have started cooking, almost done, complete, etc.)
    - Delivery tracking interface(Full Version)

        - Able to see whether the food has been delivered and track delivery
    - Finance tracking(Whether table has payed or not)
        - Tracks the bills, and whether a table has payed or not

*All functionalities will need to be used in combination with another. Examples of functionalities that will be in combination with another would be: food tracking, payments, data from customer orders, clock in hours, table availability, and surveys/data.

1c.Glossary of Terms

Busboy
An employee at a restaurant who performs the cleaning tasks in the restaurant. This includes: clearing and cleaning tables, taking dishes and cleaning them in the kitchen, and setting tables for new customers who enter the restaurant. With this app the busboy could be notified when tables need cleaning and is able to communicate with other employees when a table is cleared and a customer could be seated there.

Chef
A chef is a professional cook who has trained in the culinary arts to prepare food for customers of a restaurant. With this app the chef would be able to receive orders and notify other employees of when orders are prepared to be served.

Clock-in Hours
Employees work for a certain amount of hours a day and the Owner/Manager needs to keep track of this information so that employees get paid accordingly. Clock-in hours will be a feature in our app to keep track of how many hours employees have been working for per day.

Cook Time
The cook time is the amount of minutes/seconds required to make a meal for a customer by a chef. With our system, the cook time is managed by someone in the kitchen (in this case the chef) and they provide estimates of times for waiters and customers so they could prepare for their meal.

Customer
A person or organization that buys goods from another person or organization. With our app, the customer can do many things such as place orders in or out of the restaurant depending on what they feel like doing.

Customer Account
For the person buying goods or services and provides them a user account which will be protected by a username and password. Using an account will allow you to order from this restaurant online and will accumulate points for their reward system.

Customer Service
An employee of the restaurant that helps customers out with all sorts of issues whether it be technical (ex. an issue with the app) or physical (ex. order was wrong).

Database
A set of data stored on our network that can be accessed by users such as Owners/Managers.

Delivery
If a customer chooses to not want to eat inside the restaurant and eat in the luxury of their own they can do so. Delivery allows a restaurant to bring a customer their food directly to their location. This type of ordering will usually amount a small additional fee (for travel expenses).

Delivery Driver
An employee working for a restaurant that brings a customer their order directly to them. They will drive their own or company owned cars to bring a customer their food. With our app they can communicate with the customer of when their food has been taken from the restaurant, when they are driving, and when they have arrived at the desired location.

Dine-In
When a customer wants to sit down and eat at the restaurant itself.

Discounts
A discount is when an organization reduces the price for customers. Using the reward system with our app, rewards earned by frequent customers can be used to reduce the price of items.

Dish Popularity
When a meal is enjoyed by many customers, its approval rating up making it a popular dish. With our app, our rating system can produce data that will be sent to Owners/Managers that show which dishes are popular based on how each customer rates each meal and based on how many customers ordered the meal. This data can help managers push certain dishes forward, adjust prices on certain meals, remove items from the menu, and even modify certain meals.

Employee
An employee is a person who works at an organization, which in this case is a restaurant. These employees vary in jobs done such as: busboy, waiter/host, or chef.

Employee Account
An employee using our app can login to their account with their given username and a created password. This account can see many things based on their occupation. Every employee can see their hours worked, and can clock in. Chefs and Waiters can see when orders have been placed, which tables have placed each order, and what each order contains. Waiters and Busboys can see which tables are occupied, available, or need to be cleaned.

Floor Layout
A floor layout is a diagram of the restaurant. For certain employees using our app, they can see whether a table is available, occupied, or dirty. They could also see which order has come from which table.

Menu
The list of food choices and drinks a customer can order. With our app customers can view the restaurant's menu to pick different options, which will be delivered to the other employees to make and bring the customer that item.

Order
A list of items from the menu that a specific customer or group of people being served has ordered.

Order Queue
The list of orders the chefs must cook. Follows a first in first out order ,so orders that are placed the soonest will get served more soon. The app will show the chef what they need to make.

Owner/Manager
The person who manages employee scheduling, payroll, customer service issues, inventory, and ensures that the restaurant is operating in an efficient manner.

Owner Account
A login for the owner and/or manager that comes with special privileges that other employees don’t have. This can include being able to update the restaurant menu or seing sales analysis.

Payment
The amount of money a customer will need to buy their order. Either through cash or card depending on what customer chooses.

Reservation
When a customer(s) decides to dine-in and schedule a table for themselves ahead of time.

Rating System
Customer feedback on either a menu item or the restaurant. Based on five stars where 1 star is less favorable and 5 stars is exceptional.

Reward System
A system where regular customers can receive special offers by buying items. Requires a user account to claim an offer or reward. The way our reward system works is whenever a customer (with an account) buys an item, they earn a small amount of reward points. Once the customer reaches a certain amount of reward points, they can use this to get a discount off menu items and possibly get items for free if enough reward points are earned.

Table Status
A table's status is either available, occupied/reserved, or dirty based on whether no one is at the table, is at a table or when a customer wants a table in advance, or a customer has just left a table which needs to be cleaned for the next customer. Using our app, employees (waiters/hosts and busboys) can view and modify a tables' status and customers can view a tables' status and reserve a table as well.

Take-Out
Take-out is when a customer does not want to sit and eat at the restaurant, and instead of being delivered to a certain location but can directly pick up their food from the store.

Tip
A tip is a small amount of money, separate from the meal amount, that can be included for a waiter for providing an enjoyable experience.

Tracking (Delivery)
Delivery tracking is used to provide a customer with information about where their food is currently. Our app will provide information to customers who order online for when their food is about to be delivered.

User Account
A private location on a network server that stores information about a user's information such as usernames, passwords, names, etc. for the app.

Waiter/Host
The host is an employee at a restaurant that seats customers when they enter the restaurant. Waiters are also employees, however they take orders directly (if the customer chooses not to order through the app) and/or serve customers with the food they order.

Section 2: System Requirements

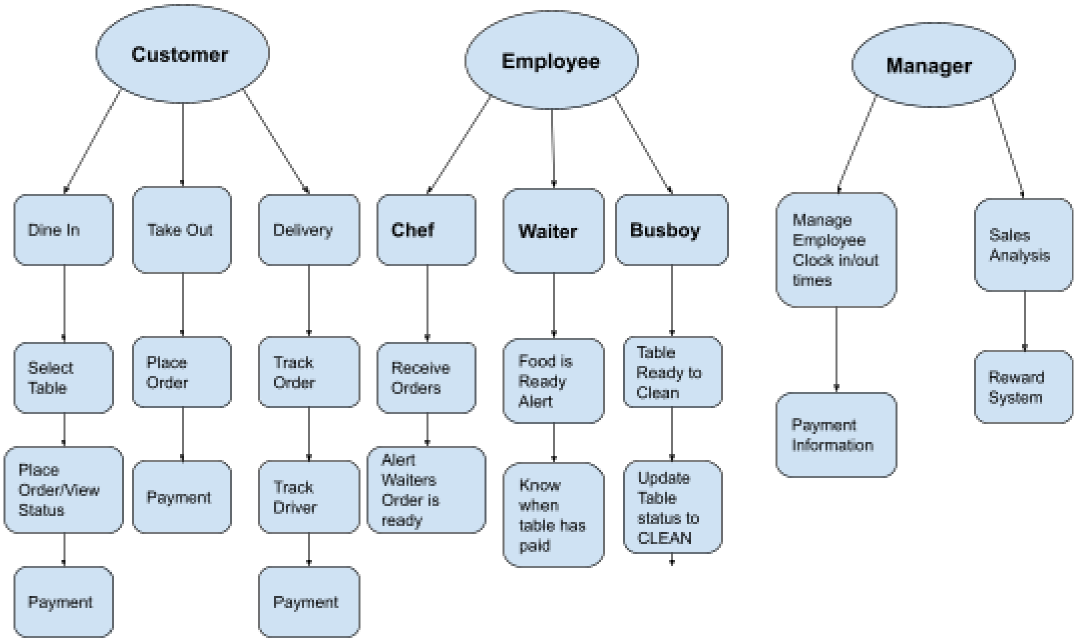
2a.Business Goals

Our application will allow the restaurant to function significantly more efficiently than a restaurant that operates without the app by providing the managerial staff, employees, and customers with a communication hub to interact in a fast, well organized manner. The goals of the app are to make the customer experience as quick and simple as possible, to maximize the efficiency with which employees can do their jobs through the automation of many tasks, and provide owners/managers with a comprehensive statistical analysis of their cash flow.

Customers will be able to select dine in, take out, or delivery, and select a table if dining in, directly from the app. This single feature eliminates the need for an employee such as a host / hostess that would typically serve this function. When dining in, the customer will be able to make reservations, view wait time for tables, place their order, and view the status of their order, and pay their bill all directly from the app, all without the assistance of an employee. If ordering for delivery, customers will be able to track when their food is ready, when it is picked up by the driver, and give an estimated time of arrival. These features save the customer time, provide them with additional information about the status of their order, and generally improve the experience of eating at the restaurant.

The app will greatly increase the efficiency of restaurant staff by providing them with real time information about various statuses around the restaurant. The chef will instantly receive orders as they come in digitally instead of on written paper slips, which take longer and are more susceptible to miscommunication due to bad handwriting. When an order is complete, the Chef will alert the waiter / driver via the app that the order is done so they can serve the order as quickly as possible. When a table pays their bill, the status of that table will change to “dirty” and busboys will be alerted to clean that table, after which they will update the table status to clean, allowing customers to select the table again. The app allows all types of employees to instantly communicate vital information with each other, whereas a typical restaurant would waste a great deal of time-- and therefore money-- during these processes.

From the restaurant side, managers/owners will be able to look at a sales analysis containing information about how frequently each item was ordered. Additionally, managers / owners will be able to monitor when employees clock in or out, paycheck information for each employee, and employee scheduling. If the owner/manager wishes, they are able to entice customers to return by implementing a rewards system that tracks money spent by each customer and provides them with redeemable coupons to spend at the restaurant. This data will help owners make better financial decisions regarding their menu items, inventory, and payroll.



2b.Enumerated Functional Requirements

Key: Customer, Manager, Employee

Food Delivery

Key	Label	Priority	Description
🟥	REQ-1	4	The application shall provide takeout, delivery, and dine in options.
🟥	REQ-2	2	The application shall allow for the user to track order delivery.
🟥	REQ-3	4	The application shall allow the user to view the estimated arrival time of a delivery.
🟧	REQ-4	3	The application shall allow for tracking the status (cooking, almost done, complete) of food orders.
🟦	REQ-5	4	The application shall allow the manager to view delivery completion time.
🟧	REQ-6	2	The application shall allow the employee to check delivery status. `
🟧	REQ-7	4	The application shall allow the chef to notify waiters when food is ready to deliver.
🟦	REQ-8	4	The application shall allow the manager to view order completion

🟦			time.

Account Management

Key	Label	Priority	Description
🟥	REQ-9	2	The application shall allow the registered members of customers to login into their customer account.
🟥	REQ-10	2	The application shall allow the customers who do not have their accounts to create their new account.
🟥	REQ-11	2	The application should allow the customers to change their usernames or passwords.
🟥	REQ-12	3	The application should allow the customers to retrieve their username or password that they have forgotten.
🟥	REQ-13	1	The application shall allow for a rewards system tied to the accounts of customers.
🟦	REQ-14	2	The application shall allow the registered member of manager to login into his/her manager account.
🟦	REQ-15	2	The application shall allow the new manager to create the new manager account.
🟦	REQ-16	2	The application shall allow the manager to delete his/her account.
🟦	REQ-17	3	The application should allow the manager to retrieve the username or password that he/she has forgotten.
🟧	REQ-18	2	The application shall allow the registered members of employees to login into their employee account.
🟧	REQ-19	2	The application shall allow the new employees to create the new employee account.
🟧	REQ-20	3	The application should allow the manager to retrieve the username or password that he/she has forgotten.

Scheduling

Key	Label	Priority	Description
🟦	REQ-21	1	The application should allow the manager to view employee schedules
🟦	REQ-22	2	The application shall allow the manager to view when

🟦			employees have arrived and left.
🟧	REQ-23	2	The application shall allow the employees to clock in and clock out.
🟦	REQ-24	3	The application should allow the manager to create schedules for employees
🟦	REQ-25	2	The application should allow the manager to view and edit employee payroll
🟦	REQ-26	2	The application should allow the manager to view finances

Table Status

Key	Label	Priority	Description
🟥	REQ-27	3	The application shall allow the user to see which tables are available.
🟦	REQ-28	2	The application shall allow the manager to view the number of available tables and their respective number of seats.
🟦	REQ-29	2	The application shall allow the manager to view the time when a table has paid their receipt.
🟦	REQ-30	1	The application shall allow the manager to view the receipt of different tables.
🟧	REQ-31	5	The application shall allow the employee to view which tables are occupied and empty.
🟧	REQ-32	3	The application shall allow the employee to view which tables are dirty and clean.
🟧	REQ-33	4	The application shall allow the employee to check whether a table has paid their receipt or if they have yet to do so.
🟧	REQ-34	3	The application shall allow for tracking the status (cooking, almost done, complete) of food orders.

Ordering Interface

Key	Label	Priority	Description
🟥	REQ-35	5	The application shall allow customers to order food from a provided menu.
🟥	REQ-36	4	The application shall allow the user to view order start time and estimated completion time.
🟥	REQ-37	2	The application shall suggest popular menu choices
🟦	REQ-38	2	The application shall allow the manager to update the menu by removing or adding menu items.
🟦	REQ-39	5	The application should provide the customer option to pay by debit, credit, or cash

Data Analysis

Key	Label	Priority	Description
🟦	REQ-40	2	The application shall produce informational graphs using data compiled from orders.
🟦	REQ-41	1	The application shall allow the manager fast access to customer reviews.
🟦	REQ-42	2	The application shall provide data on sales and update finance accounts accordingly.

2c.Enumerated Nonfunctional Requirements

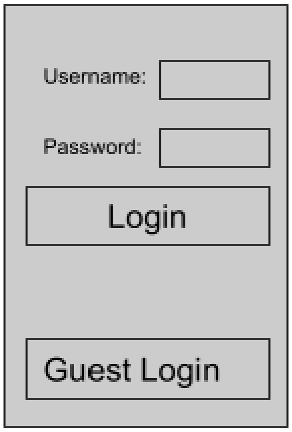
Key: Customer, Manager, Employee

Key	Label	Priority	Description
🟥	REQ-42	5	The application should be easy to use and be user friendly. Any person, even with little to no app experience should be able to use this app without too much trouble. If one does have trouble they should be able to contact customer service easily.
🟥	REQ-43	3	The application should only take a maximum of a few seconds of

🟥			delay time between each step during the ordering process.
🟥	REQ-44	5	The application should be well phrased so that all customers can easily understand functions of each part of the application.
🟥	REQ-45	4	The application should be able to accept up to 250 total customers without severe system interruption.
🟥	REQ-46	2	The application should display information to customers in high quality of resolution. For example: if their phone has a maximum resolution of 2280 x 1080, the output of our app should match that
🟥	REQ-47	3	The application should open and close almost instantaneously when a user chooses to exit the app.
🟦	REQ-48	2	The application should send the table vacancy or delivery completion information to the manager within a few seconds for efficient table/delivery flow.
🟦	REQ-49	4	The application should send the information about incomplete delivery to the manager within 5-10 minutes when delivery is not completed after the estimated time.
🟦	REQ-50	5	The application must be designed in the way the manager can handle the fast recovery when system error occurs. They can click on a button that then displays customer service information.
🟦	REQ-51	3	Security requirements in place to authenticate users and therefore user privileges (manager has advanced privileges)
🟧	REQ-52	4	The application should allow the employee to receive order information in less than 5-10 seconds after a customer finishes ordering.
🟧	REQ-53	3	The application should display order information or customer complaints with simplicity in the Order lists section, so that the employees can easily view such information.

2d.User Interface Requirements

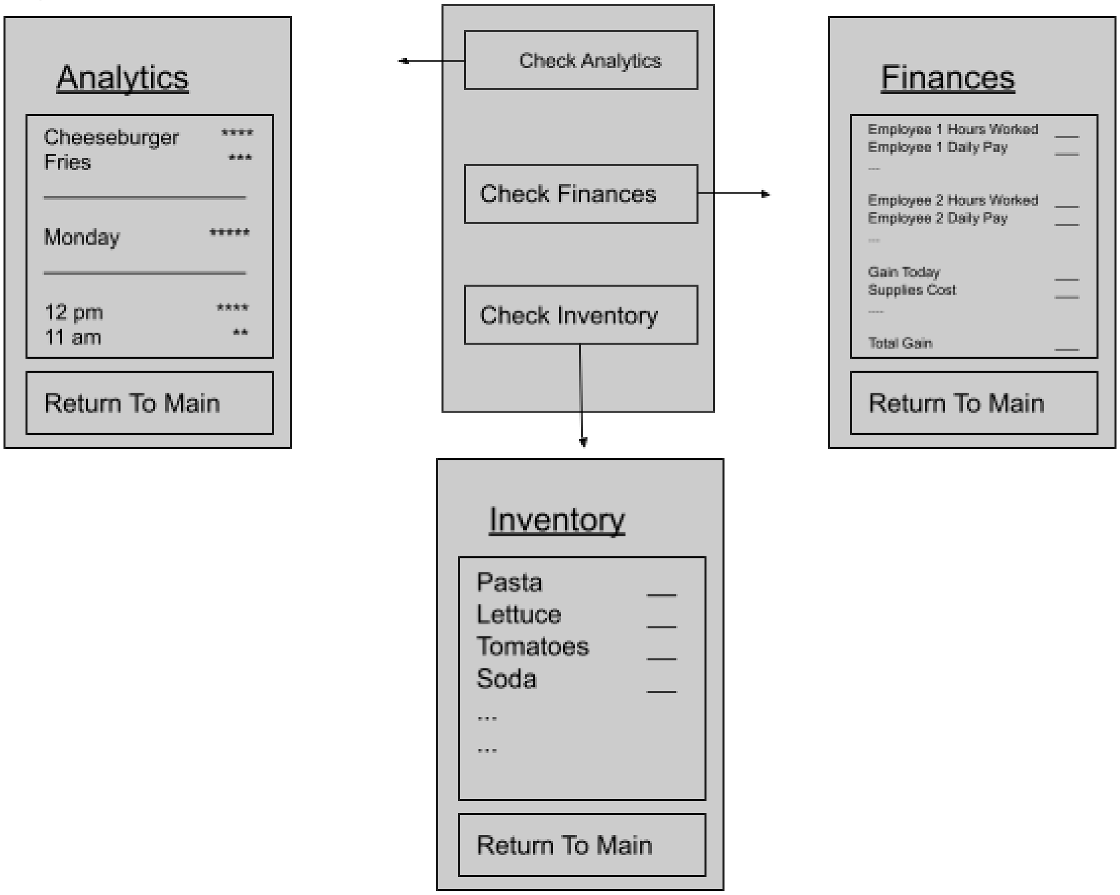
1) Screen Available to all users.Depending on the output, will load into different screens



Key: ALL

Key	Label	Priority	Description
ALL	REQ-54	3	All of the customers, managers, and employees would be able to log in to their account by typing in the username and password that they have created into the text boxes.

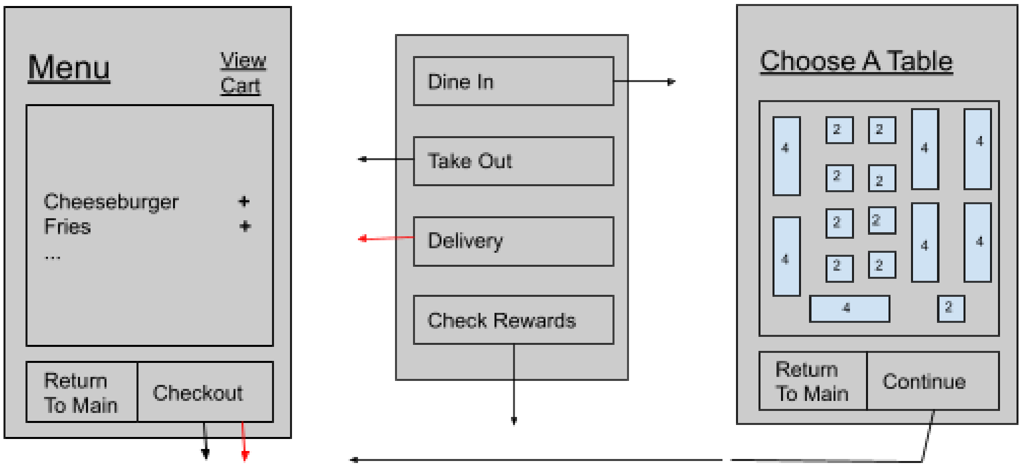
2) Owners Screens

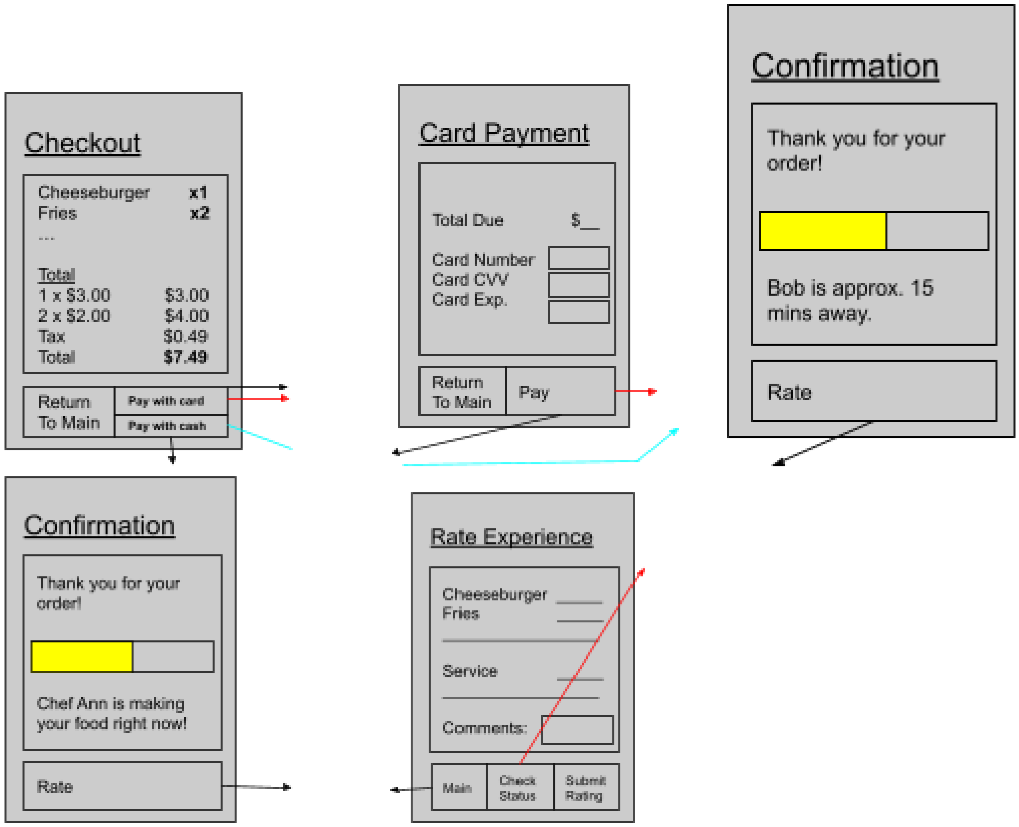


Key: Manager

Key	Label	Priority	Description
🟦	REQ-55	3	On the main screen, the manager would be able to look at analytics, finances, or inventory conditions of the restaurant by clicking on the button of corresponding sections.
🟦	REQ-56	4	In the finance page, the manager would be able to see overall total revenue gained as well as a list of the outcomes and incomes. The manager would be able to return to the main page after finishing looking at information.
🟦	REQ-57	4	In the inventory page, the manager can track the current stock of the ingredients used in the restaurant by real time. The manager would be able to return to the main page after finishing looking at information.
🟦	REQ-58	4	In the analytics page, the manager can track various types of statistics of total number sold for each menu, number of customers visited each day in a week, and number of customers visited for each time in a day.

3) Customers Screens (Difference between logged in user and guest will be reward system visibility)





Key: Customer

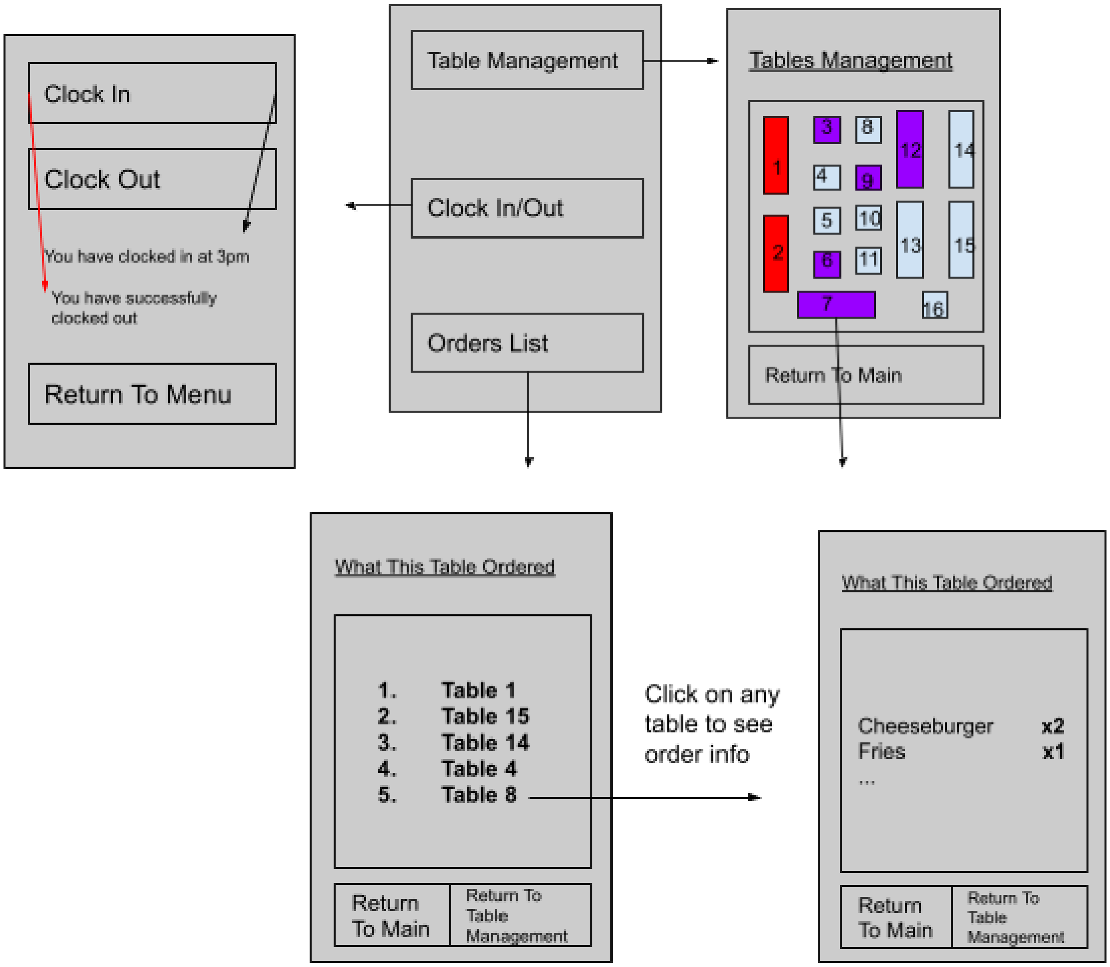
Key	Label	Priority	Description
🟥	REQ-59	3	On the menu, the customer should be able to View the Menu and add Food items to their Cart, as well as access a “Cart” and Checkout screen.
🟥	REQ-60	2	After logging in the Customer has the choice to click on Dine in, Take out, Delivery, and check their rewards.
🟥	REQ-61	4	In the Checkout page the customer can view their selected items, as well as the cost per item and total cost. They can also choose to return to main to cancel it, or pay with card/cash.
🟥	REQ-62	3	On the card payment screen the customer can input their card info to pay.
🟥	REQ-63	3	After selecting dine in, the customer can then choose their table from a table select page, from any available and clean tables.
🟥	REQ-64	3	There are several confirmation pages for various types of customers(Dine in, Delivery, Take out). It should show approximate time it takes for the order to be complete as well as an option to rate the service.

🟥	REQ-65	2	After finishing their meal customers get an option to rate their meal. Customers will also be allowed to give feedback to the chef through the comment section.

4) Employees Screens

Waiter/Host:

Click on a table to see what they ordered.
For now:
Red = Cleaning Required
Purple = Occupied
Blue = Open

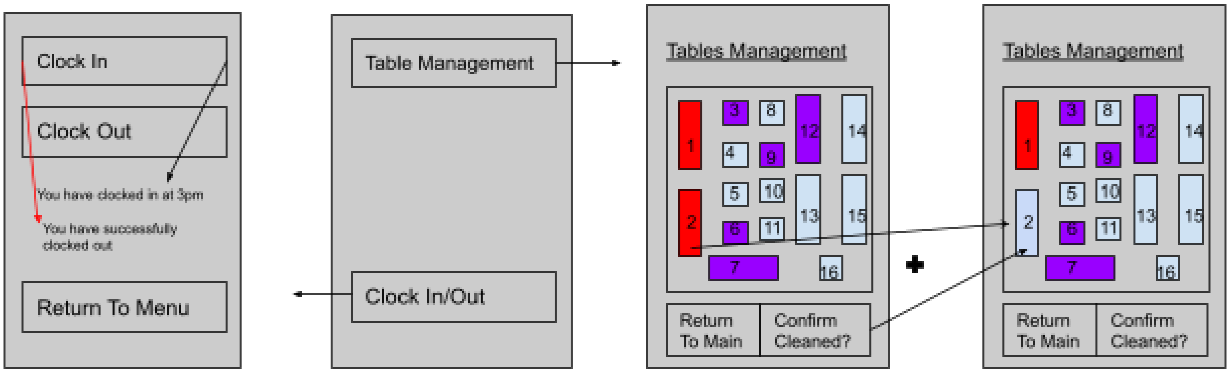


Key: Employee

Key	Label	Priority	Description
🟧	REQ-66	3	The waiter would have the choice to view the table management page, clock in/out page, or list of orders page by clicking on the corresponding buttons.

🟧	REQ-67	3	In the clock in/out page, the waiter would be able to notify their attendance through the clock in button and get off of the restaurant through clicking on the clock out button. The waiter would be able to return to the main menu after clicking the clock in or clock out button.
🟧	REQ-68	4	In the tables management page, the waiter would be able to view the current status for each table being in available, occupied, or dirty condition. The busboy would be able to return to the main page upon viewing the page.
🟧	REQ-69	5	In the orders list page, the waiter can track the tables that customers in such tables have confirmed the order. By clicking each table, the waiter can see the menu that customers in each table have ordered.

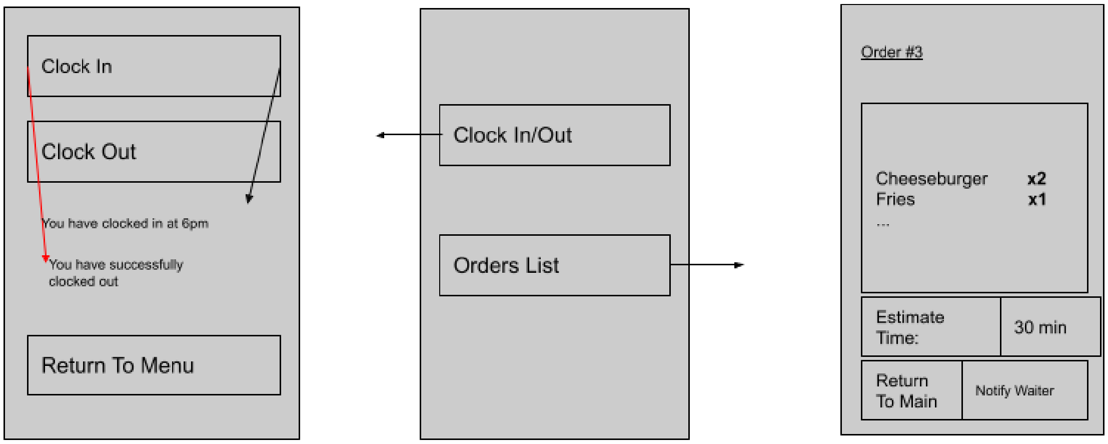
Busboy:



Key: Employee

Key	Label	Priority	Description
🟧	REQ-70	3	The busboy would have the choice to view the table management page or clock in/out page by clicking on the corresponding buttons.
🟧	REQ-71	3	In the clock in/out page, the busboy would be able to notify their attendance through the clock in button and get off of the restaurant through clicking on the clock out button. The busboy would be able to return to the main menu after clicking the clock in or clock out button.
🟧	REQ-72	4	In the tables management page, the busboy would be able to view the current status for each table being in available, occupied, or dirty condition. The busboy would be able to return to the main page or click the certain table to change the status of that table to be in available condition. The status of that table would be turned into an available condition upon clicking the “Confirm Cleaned?” button.

Chef:



Key: Employee

🟧	REQ-73	2	The chef should be able to login with his/her credentials to see the chef main page.
🟧	REQ-74	4	On the menu, the chef has two options, one being able to clock in and clock out for the manager to see how many hours a day the chef has worked, and the other being an orders list for the chef to produce food in its given order for the customer.
🟧	REQ-75	3	After pressing the “Clock In/Out” option on the main screen, the chef should be able to Clock in or clock out based on when he/she arrives and when he/she ends work.
🟧	REQ-76	4	After pressing the “Orders List” option on the main screen, the chef should be able to see what customers have ordered (sorted based on when each order was placed). The chef should also be able to put out estimates (possibly preexisting) and notify waiters when orders are done.

Driver

Key: Employee

Key	Label	Priority	Description
🟧	REQ-77	3	On the menu screen, the driver has the option of looking at the food status or the Delivery/customer information.
🟧	REQ-78	4	On the Food Status screen the driver will see if there is any food that is ready to be picked up from the kitchen for delivery.
🟧	REQ-79	4	On the Delivery/customer information screen the driver will have to

🟧			input how long it will take them to deliver the food. This screen will also give the driver information about the customer and ways they can contact them.

Section 3: Use Cases

3a) Stakeholders

There will be a number of stakeholders needed for this app to be used efficiently. The stakeholders include: **the Owner and/or Manager, the Waiter/Waitress and the Host, the Busboy, the Chef, the Driver, and the Software Developers.**

The restaurant owner and/or manager can use this app to make their business more efficient by allowing their employees to benefit from an effective app which would allow them to communicate with each other without having to waste time. The customers, who are the most valuable asset to the restaurant, could also benefit from this app due its nature of being simple and effective allowing them to choose how they want to spend their time and money at this restaurant. Lastly, the software developers are the key to a great app for this restaurant. Without them, the app will not be able to run smoothly, or even exist! That being said, software developers will help build the app even further, creating more features based on how the owner and or managers sees fit to make a more efficient method for employees and suitable experience for customers.

3b) Actors and Goals

Initiating actors

Actor	**Role**	**Goal**
Customer (Account)	A customer with an account will be able to choose any of the available methods of ordering food such as dine-in, take-out, or delivery. They could also reserve a table before arriving at the restaurant and place/track orders as well. Lastly, customers	A customer, when using our app, should have a great experience using our and promote more business in the restaurant especially with a reward system. This actor will likely be one who returns

	with accounts will gain and spend rewards based on what they have purchased previously.	many times.
Customer (Guest)	A guest will be able to choose any of the available methods of ordering food such as din-ine, take-out or delivery. They could also reserve a table before arriving.	A customer without an account should have a similar experience to one with an account without earning rewards. This actor will be those trying out this restaurant likely.
Manager	A manager with an account will be able to add and remove items from the menu.	The manager will be able to easily alter the menu using the interface.
Waiter/Waitress/Host	A waiter, waitress, or host should be able to serve cooked foods to the customers who ordered such foods. They should be able to tell customers and busboys when tables become occupied, free, or need cleaning.	Through the application, the waiter can serve foods to the right table so that the miscommunication or situation in which foods are served to wrong tables would not take place. The waiter can welcome new customers to the vacant table with time saved using the application.
Chef	The chef should be able to notify the waiter when an order is ready to be picked up	The goal is to have efficient communication between the chef and the waiter so that the food can be delivered as quickly as possible.
Busboy	A busboy should be able to notify other employees and customers of when a table is cleaned.	The busboy will be able to notify others of when tables open up providing a more fluent system of restaurant automation.
Driver	A driver should be able to tell a customer where they are and approx. how much time it will take to reach them.	The driver should be able to communicate with a customer allowing customers to be satisfied when an order arrives based on ongoing

		information.

Participating actors

Actor	Role	Goal
Waiter/Waitress/Host	A waiter, waitress, or host should be able to receive orders, track which foods are ordered from each table, and check the vacant tables.	Through the application, the waiter, waitress, or host would be able to look at all of the orders received from each of the tables so that foods can be served to proper tables. The waiter, waitress, or host can also track for the non-dirty vacant tables that customers can be seated on.
Chef	A chef should be able to view customer orders and begin cooking meals depending on when each order was placed.	Through the app, the chef will be given clear and concise instructions about when each item was placed and who ordered the meal to be more productive and efficient.
Busboy	A busboy should be able to check when a table is dirty, occupied or clean.	This function allows busboys to be proactive and instead of waiting around looking for tables to be clean, can be notified when a table needs to be cleaned.
Driver	A driver should be able to check when an order is ready for pickup.	This allows drivers to be notified of when an order is ready to be picked up saving time.
Manager	The manager should be able to view seating statuses and order statuses.	This allows them to oversee the restaurant to make sure everything is running smoothly.

3c) Use Cases

3c.i) Casual Description

Food Delivery-

- **Use Case 1(UC1) Types of orders :** The applications will allow a customer to place an order for Dine in, Take out, or Delivery.
  Responds to REQ 1
- **Use Case 2(UC2) Order Status:** The waiters and chefs will be able to communicate to each other when an order is ready to be delivered. The manager can also keep track of completion of orders.
  Responds to REQ 7
- **Use Case 3(UC3) Track Order :** The applications will allow a customer, manager, and employee to view the order status.
  Responds to REQ 2,3,4,5,6,8,36

Scheduling-

- **Use Case 4(UC4) Clock In:** The application will allow for employees to check in when they are starting their shift.
  Responds to REQ 22, 23
- **Use Case 5 (UC5) Clock Out:** The application will allow for employees to check out when they are finished working their shift.
  Responds to REQ 22, 23
- **Use Case 6 (UC6) Employee Schedule:** The application will allow for managers to create and manage weekly schedules for employees for when they have to work.
  Responds to REQ 21, 24
- **Use Case 7 (UC7) Payroll:** The manager will be able to view payrolls for employees as well as be able to pay them.
  Responds to REQ 25
- **Use Case 8(UC8) Finance:** The manger will be able to view the finances of the company and their analytics.
  Responds to REQ 26

Account Management-

- **Use Case 9(UC9) Login :** The applications will allow a manager, customer or employee to login, or change password to their account.
  Responds to REQ 9,11,12,13,14,17,18,20
- **Use Case 10(UC10) Logout :** The applications will allow a manager, customer or employee to logout from their account.

- **Use Case 11(UC11) Register:** Allows a guest to create an account so that he or she can access the application features. The employees can also create their corresponding employee accounts.
  Responds to REQ 10,15,19
- **Use Case 12(UC12) Delete:** Allows a manager to delete an account so that the important restaurant information can be kept secure when managers quit the job or the manager account gets hacked.
  Responds to REQ 16
- **Use Case 13(UC13) Edit:** Any user is able to edit certain information in their account, after having inputted their username and password(login).
  Responds to REQ:38

Order Interface-

- **Use Case 14(UC14) Take Out** : The applications will allow a customer to place and pick up an order.
  Responds to REQ: 35,
- **Use Case 15(UC15) Dine in** : The applications will allow a customer to choose to dine in.
  Responds to REQ:35
- **Use Case 16(UC16) Delivery** : The applications will allow a customer to place a delivery
  Responds to REQ:35
- **Use Case 17(UC17) Payment:** Allows the customer to pay for their meal, after which the chef will begin cooking.
  Responds to REQ: 39
- **Use Case 18(UC18) Rate Food:** This application will allow the customer to rate the quality of the food on a scale from 1(worst) to 5 (best).
  Responds to REQ 37

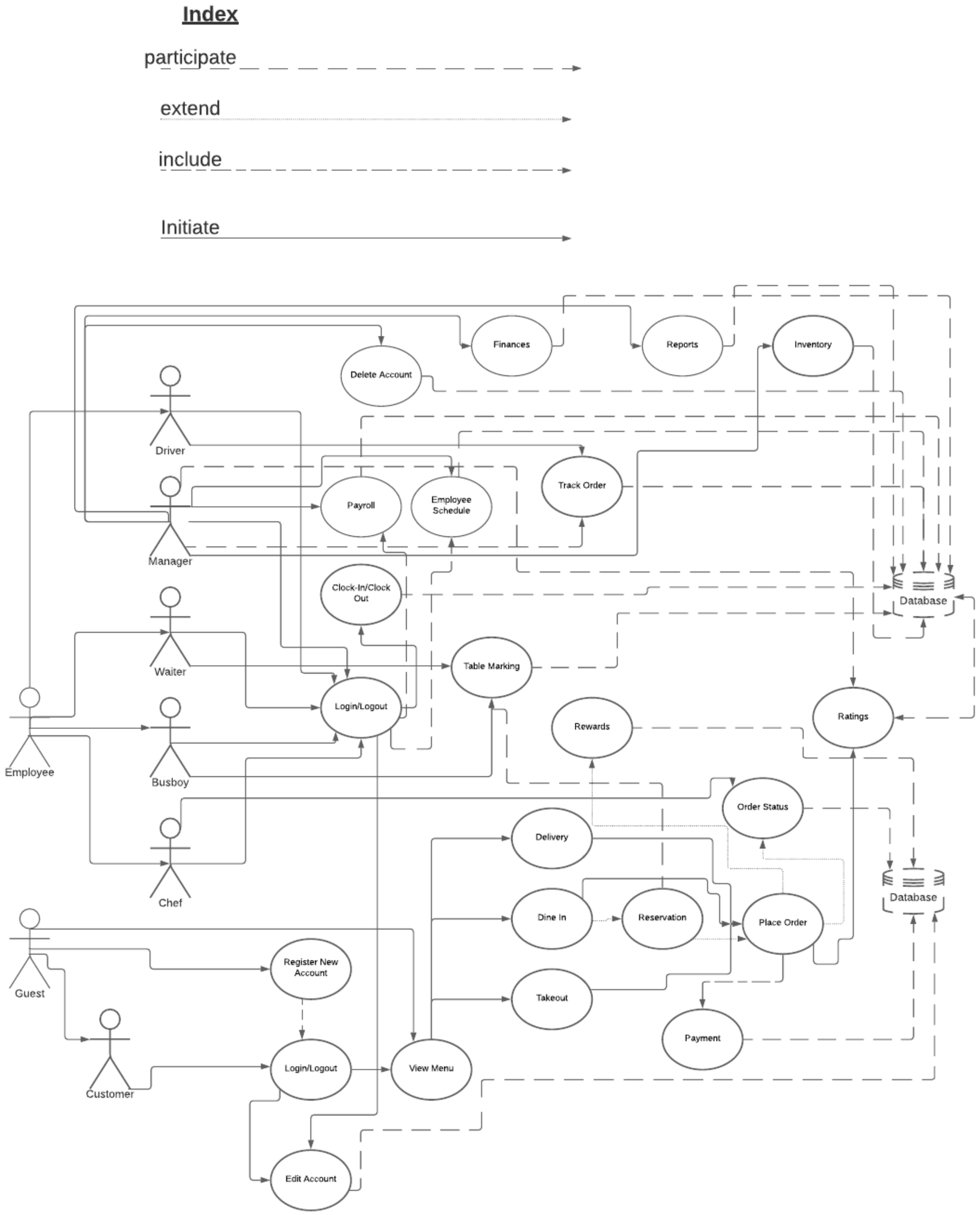
Table Status-

- **Use Case 19(UC19) Reservation :** Allow the customer to reserve a table for a later date or time right from our application without having to call the restaurant.
  Responds to REQ: 27
- **Use Case 20(UC20) Check Tables :** Allows employees to check a table's status like the number of seats at the table, if it's open , if it's clean , and customer payment status.
  Responds to REQ: 28, 29, 30, 31, 32, 33, 34

Data Analysis-

- **Use Case 21(UC21) Reports :** The application will make reports which allows managers to view important information aspects about the restaurant generated from the daily customers in easy to follow tables, charts and graphs.
  Responds to REQ:40,41,42

3c.ii) Use Case Diagram



3c.iii) Traceability Matrix

	W	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16	UC17	UC18	UC19	UC20	UC21
REQ1	4	X																				
REQ2	2			X																		
REQ3	4			X																		
REQ4	3			X																		
REQ5	4			X																		
REQ6	2			X																		
REQ7	4		X																			
REQ8	4			X																		
REQ9	2									X												
REQ10	2											X										
REQ11	2									X												
REQ12	3									X												
REQ13	1									X												
REQ14	2									X												
REQ15	2											X										
REQ16	2												X									
REQ17	3									X												
REQ18	2									X												
REQ19	2											X										
REQ20	3									X												
REQ21	1						X															
REQ22	2				X	X																
REQ23	2				X	X																
REQ24	3						X															
REQ25	2							X														
REQ26	2								X													

REQ27	3																			X		
REQ28	2																				X	
REQ29	2																				X	
REQ30	1																				X	
REQ31	5																				X	
REQ32	3																				X	
REQ33	4																				X	
REQ34	3																				X	
REQ35	5														X	X	X					
REQ36	4		X																			
REQ37	2																		X			
REQ38	2													X								
REQ39	5																	X				
REQ40	2																					X
REQ41	1																					X
REQ42	2																					X
Total	W	4	8	19	4	4	4	2	2	18	0	6	2	2	5	5	5	5	2	3	20	5

https://docs.google.com/spreadsheets/d/1Qa80kjc8waaJeg_0TDn6YPhRqyDUuFDs_g3kihCL1tM/edit#gid=0

3c.iv) Fully-Dressed Description

UC-9: Login
Related Requirements: REQ-9,REQ-11, REQ-12,REQ-13,REQ-14,REQ-15,REQ-16,REQ-17,REQ-18,REQ-20
2Initiating Actors: Manager, Employees, Customers

Actor Goals:
Grant access to corresponding accounts in the system

Participating actors: database

Preconditions: User already has account created

Postconditions:

Flow of Events for Main Success Scenario:

1. →The user opens the application to view the login/guest user page
2. →The user clicks to login
3. ←The system prompts for account information
4. →The user inputs their account username and password
5. ←The systems the displays the corresponding interface depending on the user

Flow of Events for Alternate Success Scenario 1 (Password reset):

1. →The user opens the application to view the login/guest user page
2. →The user clicks to login
3. ← The system prompts for account information
4. →The user inputs an account username and password
5. ←The system displays that incorrect information was inputted and prompts the user to try again or prompted to change their password
6. →The user inputs their correct account username and password (4 and 5 can loop) or clicks to change their password.
7. ←The systems then displays the corresponding interface depending on the user

Flow of Events for Alternate Success Scenario 2 (Created an account):

1. →The user opens the application to view the login/guest user page
2. →The user clicks to login
3. ←The system prompts for account information
4. →The user clicks forgot password
5. ←The system prompts for email to send password reset info to

UC-21: Reports
Related Requirements: - REQ-40, REQ-41, REQ-42,
Initiating Actor: Manager
Actor’s Goal - Allows the manager to view important aspects of a restaurant’s data generated daily by customers in the form of tables, charts, and graphs.
Participating Actors: Database
Preconditions: Manager has signed into the app.
Postconditions: Manager is shown information.
Flow of Events for Main Success Scenario: 1. → The manager opens the application to the homepage 2. → The manager clicks on “Sign-In”. Is prompted with a “Login Page”. 3. ← The manager enters his/her username and password. Then they can hit login to enter their account. 4. → The manager is brought to their “Manager/Owner Homescreen”. The manager can now click on the “View Analytics” 5. → The manager is presented with information regarding restaurant data in the form of tables, charts and graphs.

UC-15: Dine-In
Related Requirements: - REQ-35

Initiating Actors:

- Customers

Actor’s Goal:

- Allows the customers (Account and Guest) to order the table to have a seat.

Participating Actors:

- Waiter/Waitress/Host, Busboy, Database

Preconditions:

- Customers have signed into the application.

Postconditions:

- The customers are shown choices of the available tables to reserve.

Flow of Events for Main Success Scenario:

1. →
   The customers open the application to the entry page.
2. ←
   The application gives options for customers to sign in with their accounts or as guests.
3. →
   The customers click the “accounts” button.
4. ←
   The application shows up the spaces for inputs of user-ids and passwords.
5. →
   The customers can type in their user-ids and passwords in order to successfully sign in.
6. ←
   The application shows the “Customer Homescreen” with choices of dine-in, take-out, delivery, or check rewards.
7. →
   The customers click the “dine-in” option.

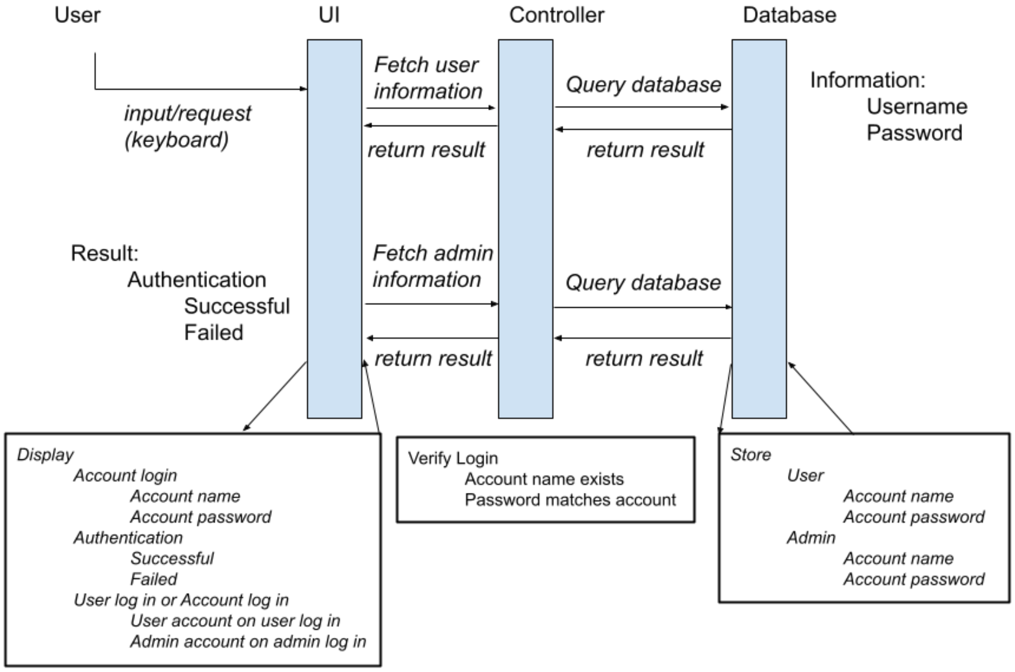
8. ←
   The application shows the screen of the “Dine-In” section with vacant, available tables shown.
9. →
   The customers can reserve one of the available tables.
10. ←
   The application notifies the successful reservation of the table.

Flow of Events for Alternate Success Scenario:

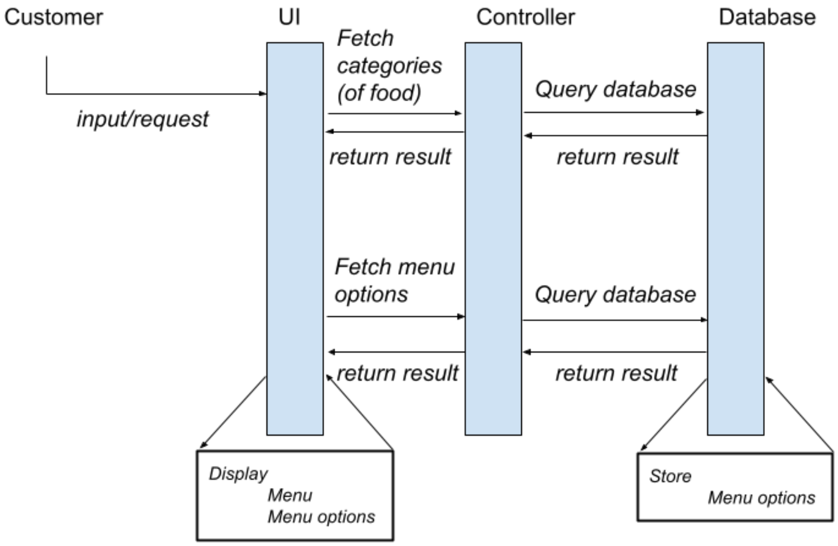
1. →
   The customers open the application to the entry page.
2. ←
   The application gives options for customers to sign in with their accounts or as guests.
3. →
   The customers click the “guest” button.
4. ←
   The application shows the “Customer Homescreen” with choices of dine-in, take-out, delivery, or check rewards.
5. →
   The customers click the “dine-in” option.
6. ←
   The application shows the screen of the “Dine-In” section with vacant, available tables shown.
7. →
   The customers can reserve one of the available tables.
8. ←
   The application notifies the successful reservation of the table.

3d) System Sequence Diagrams

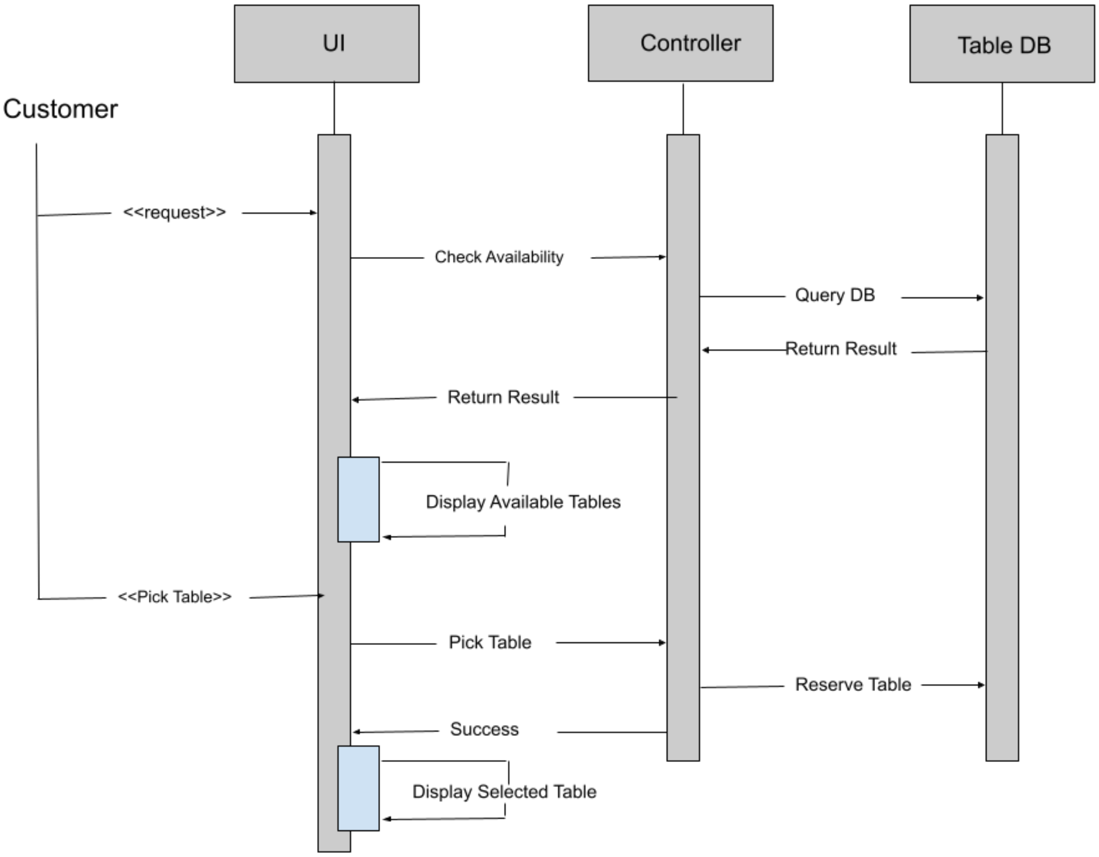
UC-5 Login



UC-6 Menu



UC-10 Table Selection

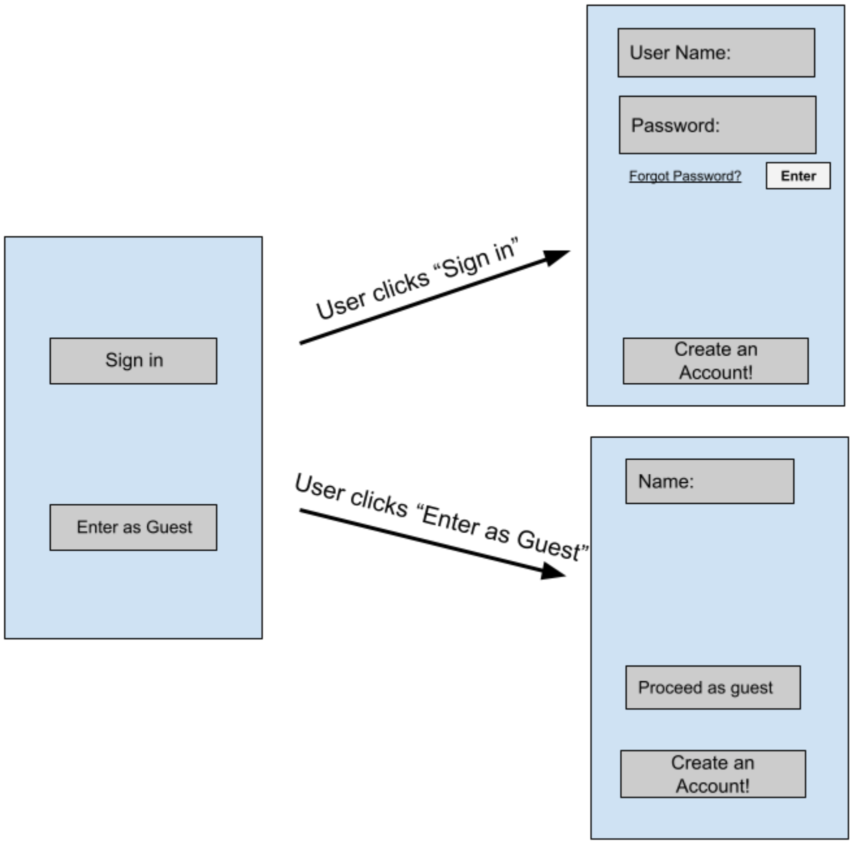


Section 4: User Interface Specification

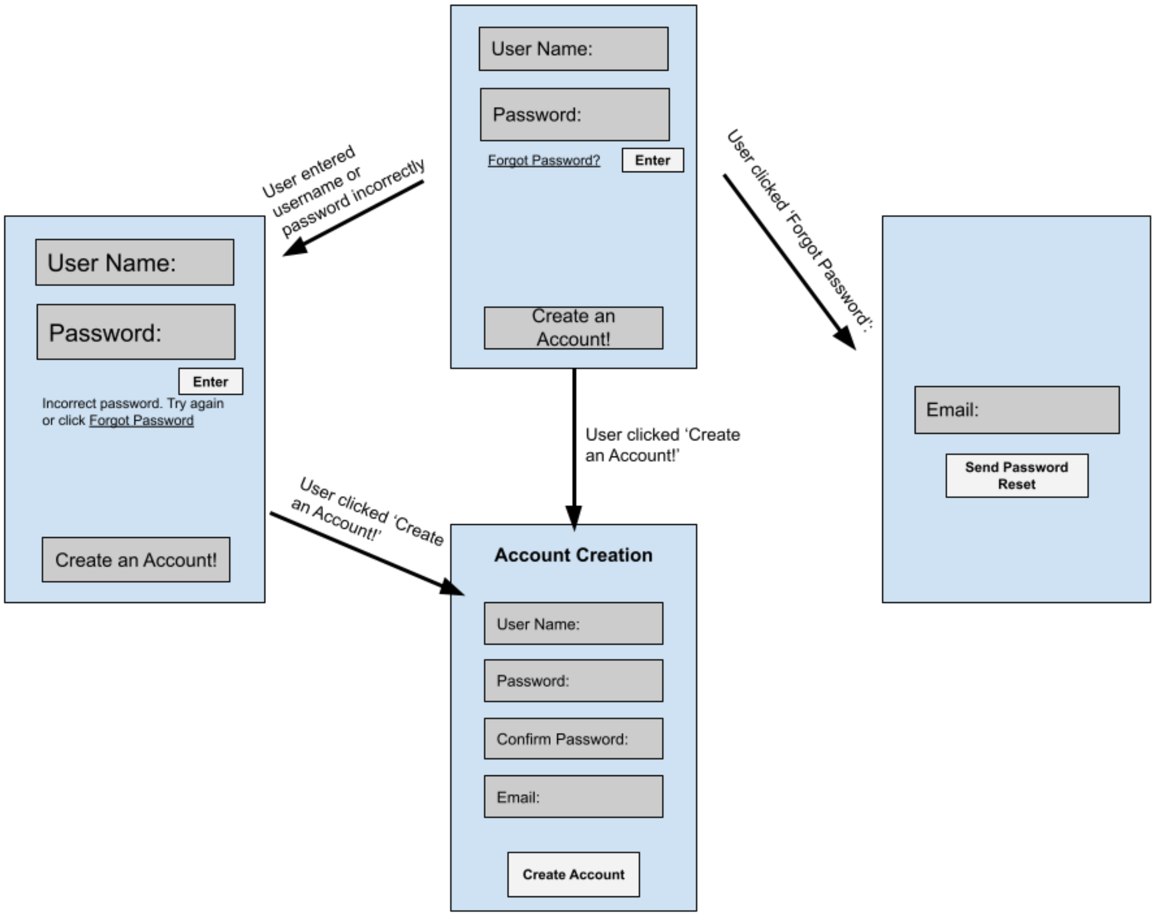
4a) Preliminary Design

UC-1 - Login

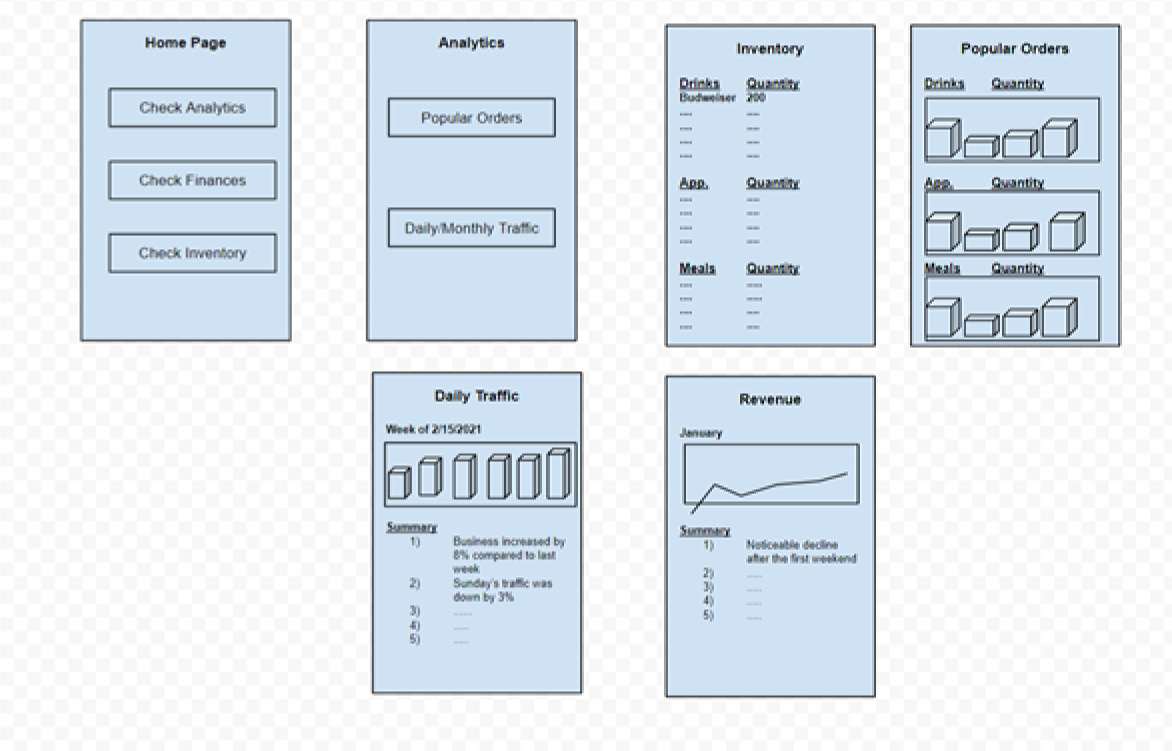
User is shown two options, sign in and enter as guest



If a user clicks “Enter as Guest”, they enter the application. If a user clicks ‘Sign in’ they are prompted with a sign in screen where the user can enter a username and password. If they enter a wrong username or password it will prompt an error. Users also will have the option to click ‘forget password’. Also, users have the option to create an account.



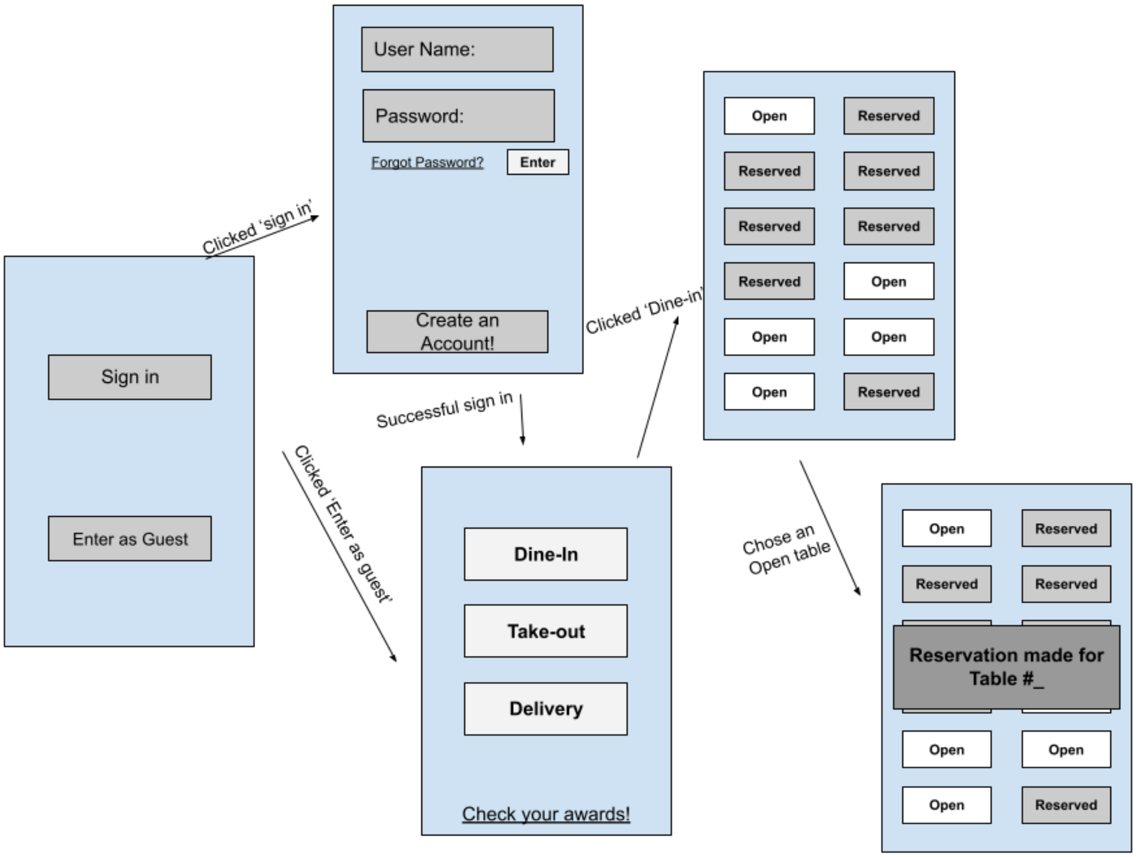
UC-9 - Reports



Summary
1) Business increased by 8% compared to last week
2) Sunday's traffic was down by 3%
3) ......
4) ......
5) ......

Summary
1) Noticeable decline after the first weekend
2) ......
3) ......
4) ......
5) ......

UC-2 - Dine-in



4b) User Effort Estimation

Usage Scenario 1: A customer who already has an account will login and order a cheeseburger, fries, and a drink for take out.

1.**Open application and click login**:1 click/touch
2.**Login**: 3 mouse clicks/touches and a variable number of keystrokes
    a.Customer clicks username
    b.Customer enters their username
    c.Customer clicks on password

d.Customer enters their password
e.Customer clicks Login

3.**Place Order**: For given situation 4 clicks.touches
a.Click Takeout option
b.From shown menu add cheeseburger,fries,and drink
(may require scrolling and at least 3 clicks)

4.**Payment** (At least 3 clicks)
a.Click checkout
b.Click pay with cash
c.A confirmation screen will be shown and user clicks rate
d.Optional comments, Main ,check status, and submit rating options

Or

4.**Payment** (credit card):at least 6 clicks and a variable number of keystrokes
a.Click checkout
b.click pay with card
c.Click card number and enter information
d.click card cvv and enter CVV
e.click Card Exp and enter expiration date
f.click pay
g.Optional comments, Main ,check status, and submit rating options

Cash: 50% data entry and 50%navigation
Card: 65% data entry and 35% navigation
Navigation refers to clicking/touching a button that takes a user to a new screen. Data entry refers to clicking/touching a certain field and entering the relevant information such as menu choices, card information, and login.

Usage Scenario 2: An employee(Host) who is registered in the system will login and clock in for the work day, as well as access the Table Management interface to reserve a table.

1.**Open application and click login**:1 click/touch
2.**Login**: 3 mouse clicks/touches and a variable number of keystrokes
a.Employee clicks username
b.Employee enters their username
c.Employee clicks on password
d.Employee enters their password
e.Employee clicks Login

3.**Clock in**

a. Employee clicks to clock in
b. Employee clicks to return to menu

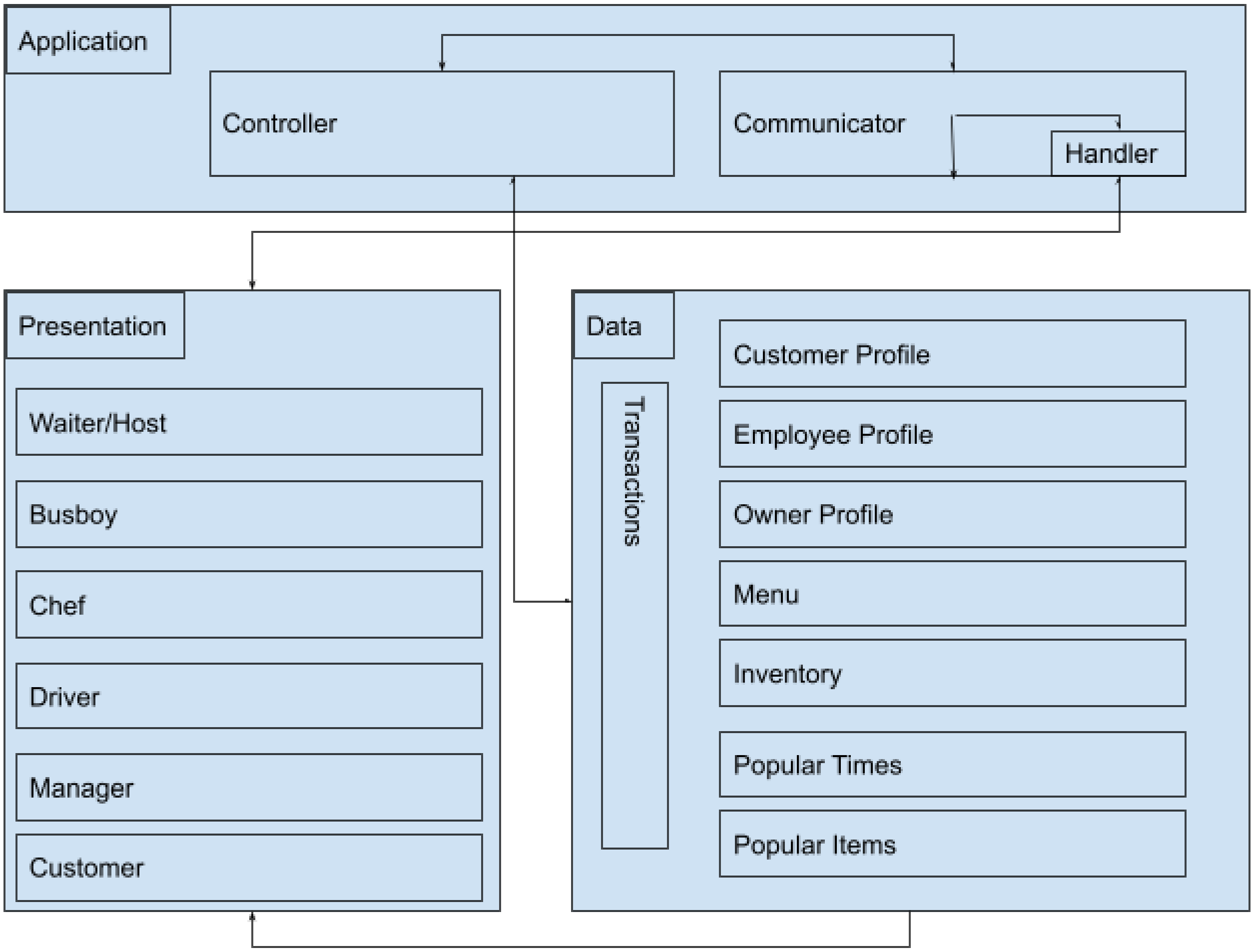
4.**Reserve table**

a. Employee clicks Table Management
b. Employee clicks on table to be reserved

c. Employee updates status to reserved
d. Employee clicks back to Menu.

Section 5: System Architecture

5a) Identifying Subsystems



The subsystem for our application is shown above, and consists of three layers which is the Application Layer, the Presentation Layer, and the Data Layer.

Data Layer: Uses the database and contains information about the Profiles of each user, the menu items, the inventory, the transactions that occur throughout the day, and lastly the popular food items and customer visit times.

Presentation Layer: Consists of the different users such as Waiter/Host, Busboy, Chef, Driver, Manager, and Customers. This layer takes information from the database and displays it to the user's screen depending on the type of user they are.
Application Layer: This layer contains a Controller, which takes tasks between packages and will use the Communicator to provide permissions between the packages and use the handler to load the events as per instruction. This layer also deals with business logic, and will contain objects that are able to implement the business functions. The use cases will all be implemented through this way in the application layer.

5b) Architectural Styles

Our application will have three layers and consists of a Data Layer, a Presentation Layer, and an Application Layer. This 3 layered architecture allows the Data layer to be run in the background providing resources alongside with the Application layer to produce what is seen by the user in the Presentation layer allowing a front-end/back-end situation.

This type of architectural style is the most practical for our application since each layer it will be running alongside our application's code. Our application will run using Java and build/edited through Android Studio which allows us to design the graphical user interface directly through this program. The application will be using a database, which will be called through and API which will allow users to get information as soon as it is edited to help get quick and accurate data.

Our application will also use a client/server architectural style which will be used to update our database frequently. This architecture will allow us to return information whenever a client requests to the server. The server, in our case, would be the database with additional application logic. Our database will be used to keep track of menu items, transactions, popularity, location (delivery and open tables), and employee/manager/customer profiles (and their permissions within the app).

5c) Mapping Subsystem to Hardware

No, all our subsystems can be run on the same server. However, there will be a downloadable mobile application where customers can order delivery, takeout, make reservations, look at reviews, etc. There will also be a version of the application for the manager/employees side, as they will obviously have a different interface.

5d) Network Protocol

The network protocol to be used for our purposes is normal sockets. The reason we are going in this direction of transmitting information is because all the backend information processing will be done on the server computer. The application will mainly serve as little more than a graphical frontend for the end user. The front end of the application will handle performing the requests. The user interacts with the application and when they perform a task, a request will be sent to a server where the information will be processed. For example, if the user orders food after clicking confirm, the order will be sent to the server which will respond back with information including estimated wait time, which is necessary for the client to display to the user.

5e) Global Control Flow

Execution Orderliness
The application is event-driven and the user has full control over the application's use. From the menu, the application will branch to show different features the application provides as per the user's command.

Time Dependency
The system is dependent on time. All applications are synced to real time in order to keep track of when menu orders (inside the restaurant) and delivery times (outside the restaurant) are placed. The system also inputs the current time to estimate when certain features will be completed. These features include estimated delivered time as well as estimating when a table will be empty- to name a few.

Concurrency
Data is maintained in sync as requests are sent to the database. The database will update with every request and send relevant information to be displayed on the app interface.

5f) Hardware Requirements

This software will run on smartphones and tablets with a touch screen running an Android operating system. The targeted minimum version is API 23: Marshmallow which would allow the program to run on 62.6% of devices. Devices will require at least 1MB of storage space to install the application. The minimum resolution required to display the application is 640 x 480 pixels. The minimum bandwidth required to access the server and database is 56 kbps. The minimum RAM required to run the application is 1GB. The software is intended to run on mobile devices with a touchscreen interface, so there is no desktop version of the application. Both smartphones and tablets will use the same version of the application.

Section 6:Project size estimation based on use case points

Relevant Equations

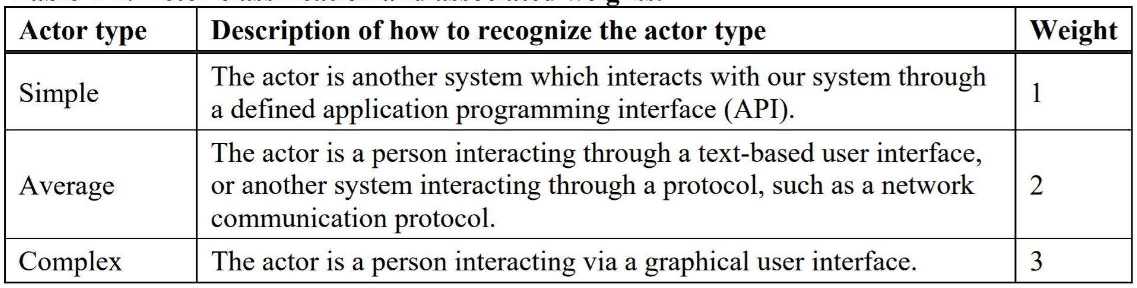
UCP(use case points) = Unadjusted Use Case Points (UUCP) * Technical Complexity Factor (TCF) * Environment Complexity Factor (ECF)

Note : Here ECF will be 1

UUCP = Unadjusted Actor Weight (UAW) + Unadjusted Use Case Weight (UUCW

Finding UAW

Table 4-1: Actor classification and associated weights.



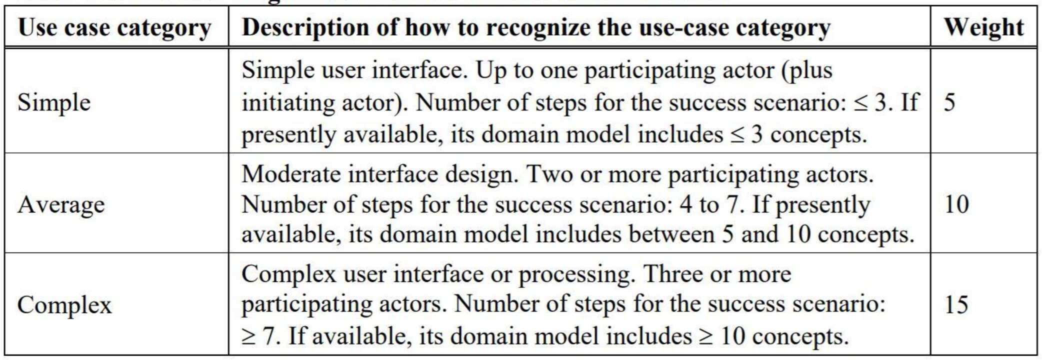
Actor type	Description of how to recognize the actor type	Weight
Simple	The actor is another system which interacts with our system through a defined application programming interface (API).	1
Average	The actor is a person interacting through a text-based user interface, or another system interacting through a protocol, such as a network communication protocol.	2
Complex	The actor is a person interacting via a graphical user interface.	3

Actor Name	Description	Complexity	Weight
Customer	Customer is interacting via graphical interface	complex	3
Manager	Same as customer	complex	3
Waiter/Waitress	Same as customer	complex	3
Host/Hostess	Same as customer	complex	3
Chef	Same as customer	complex	3
Busboy	Same as customer	complex	3
Driver	Same as customer	complex	3

UAW = 1*(0) + 2(0) + 3(7) = 21

Finding UUCW

Table 4-3: Use case weights based on the number of transactions.



Use case category	Description of how to recognize the use-case category	Weight
Simple	Simple user interface. Up to one participating actor (plus initiating actor). Number of steps for the success scenario: ≤ 3. If presently available, its domain model includes ≤ 3 concepts.	5
Average	Moderate interface design. Two or more participating actors. Number of steps for the success scenario: 4 to 7. If presently available, its domain model includes between 5 and 10 concepts.	10
Complex	Complex user interface or processing. Three or more participating actors. Number of steps for the success scenario: ≥ 7. If available, its domain model includes ≥ 10 concepts.	15

Use Case	Description	Category(Simple,Average,or Complex)	Weight(5,10,15)
UC1:Types of Orders	One initiating actor with more than 3 steps.	Average	10
UC2:Order Status	A couple actors involved. A logged in user should be able to do this in a couple steps.	Average	10
UC3:Track Order	A couple actors involved (customer, manager, and employee). A couple steps for logged in users.	Average	10
UC4:Clock In	One initiating actor and a couple steps.	Simple	5
UC5:Clock Out	Same as Clock in.	Simple	5
UC6:Employee Schedule	1 involved actor, and a couple steps (Login,click employees option, and edit)	Simple	5
UC7:Payroll	Almost the same as UC6	Simple	5

UC8:Finance	1 involved actor(manager) and several steps to check the finances	Average	10
UC9:Login	One initiating actor and a few simple steps.	Simple	5
UC10:Logout	Same as Login	Simple	5
UC11:Register	A couple involved actors (customers or employees) and a few steps (Login, create account, enter info)	Average	10
UC12:Delete	A couple actors involved and several steps after login..	Average	10
UC13:Edit	An initiating actor and a couple steps after the login to make changes to the account.	Simple	5
UC14:Take Out	1 initiating actor, several participating actors.Steps: login choose order, pay, rate.	Average	10
UC15:Dine In	1 initiating actor, 4 participating actors, and 10 steps in the success scenario.	Complex	15
UC16:Delivery	Almost the same as takeout	Average	10
UC17:Payment	One initiating actor and several steps since a user has to make an order first.	Average	10
UC18:Rate Food	Customer rates at the end of their order	Simple	5
UC19:Reservation	1 initiating actor,	Average	10

	several participating actors and steps:Login,select Dine In, choose table, make order, and pay.		
UC20:Check Tables	A couple involved actors (employees) and a few steps to check.	Average	10
UC21:Reports	1 participating actor and a couple steps.	Simple	5

UUCW = 5(9) + 10(11) + 15(1) = 45 + 110 + 15 = 170

UUCP = 170 + 21 = 191

Finding TCF

Technical factor	Description	Weight	Perceived Complexity	Calculated Factor (weight * perceived complexity)
T1	Distributed system (running on multiple machines)	2	Average (3)	2*3 = 6
T2	Performance objectives (are response time and throughput performance critical?)	1	Average (3)	1*3 = 3
T3	End user efficiency	1	Simple (0)	1*0 = 0
T4	Complex internal	1	Complex(5)	1*5 = 5

	processing			
T5	Reusable design or code	1	Average (3)	1*3 = 3
T6	Easy to install	0.5	Simple (0)	0.5*0 = 0
T7	Easy to use	0.5	Simple (0)	0.5*0 = 0
T8	Portable	2	Average (3)	2*3 = 6
T9	Easy to change(add or modify features)	1	Average (3)	1*3 = 3
T10	Concurrent use	1	Simple (0)	1*0 = 0
T11	Special security features	1	Average (3)	1*3 = 3
T12	Provides direct access for third parties (the system will be used from multiple sites in different organizations)	1	Average (3)	1*3 = 3
T13	Special user training facilities are required.	1	Simple (0)	1*0 = 0
Technical Factor Total				**32**

Constant 1 = 0.6
Constant 2 x Technical Factor Total = 0.01(32)
TCF = Constant-1 + Constant-2 x Technical Factor Total
TCF = 0.6 + 0.01(32) = 0.92.

Use Case Points:
Use Case Points = UUCP * TCF * ECF = 191*0.92*1 = 176

Project Management

Coding milestones plan
February- early March : Implement the login and clock in/out interfaces. Each of the subgroups should implement at least one important feature and then test and debug as needed.

Customer: Implement dine-in and/or takeout along with payment.

Manager: Implement analytics. Start with finances or inventory features.

Employee:
Chef:Implement orders feature
Waiter:Implement orders list feature
Busboy and/or waiter: Implement table management feature

March 23(date of first demo)

March 23 - Early/Mid April: Implement the other important features and improve upon previous features as needed. Hopefully, by this point some of the following have already been started.
Customer:Implement delivery and rewards system.
Manager: Finances and inventory related features.
Employee:Implement the driver interface

Week of April 20 (date of second demo):

April 20-May7th: Each subgroup should make any last changes and/or implement any remaining features.More testing, debugging.

IMPORTANT: GANTT CHART
https://docs.google.com/spreadsheets/d/1Rx3MXICmaH2qySpwvPafgbbn4g_-N3drkd9JR5_Ikc4/edit?usp=sharing

References

https://content.sakai.rutgers.edu/access/content/group/6951386f-dee3-4222-b5c9-7ec22bd74fe5/Software%20Engineering%20Course%202021-SE.pdf

https://www.ece.rutgers.edu/~marsic/Teaching/SE1/report1.html

http://eceweb1.rutgers.edu/~marsic/books/SE/projects/Restaurant/2018f-g4-report3.pdf https://www.ece.rutgers.edu/~marsic/books/SE/projects/Restaurant/2019-g13-report3.pdf

http://eceweb1.rutgers.edu/~marsic/books/SE/projects/Restaurant/

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