# TableTapp

## ECE 499

June 23rd, 2023

University
of Victoria

**Group ID: 19**

**Faculty supervisor: Dr. Navneet Popili**

**Co-supervisor (if any): Dr. Ilamparithi Thirumarai Chelvan**

**Team Information:**

| S. No. | Name | V Number |
|---|---|---|
| 1 | Kris Lemieux | V00928684 |
| 2 | Kai Herrero | V00914547 |
| 3 | Adil | V00924050 |
| 4 | Aryan Sahel | V00935002 |

# Table of Contents

# 1.0 Introduction

## 1.1 Problem

The restaurant industry suffers from small profit margins [1] of around ~5.16% (net profit), this figure is considerably lower than other industries. Small mom and pop restaurants see even less profit and because of the pandemic some small restaurants are losing money and have not yet recovered to pre-pandemic sales figures [2]. Since the budget is so tight in small restaurants, hiring and paying an employee is a considerable expense and performance of each employee varies. A major and operating constraint in the small restaurant business is how to manage the small budget they have and finding ways to save money that are safe and don't affect service is a huge factor in determining the success of the business.

Oftentime restaurants require software solutions such as point-of-sale (POS), inventory management, employee scheduling, kitchen management, HR management and restaurant business solutions in order to efficiently provide reliable service while running their business. For most restaurants, each software solution is outsourced and purchased either individually or as an all-in-one solution. Lots of the current software solutions that small restaurants need are subscription based and can cost up to thousands of dollars monthly for a full software solution (see [3] for price breakdowns).

To ease the cost of operations in the small restaurant industry we are making a web app called TableTapp that is an all in one restaurant solution with the additional feature of contactless waiting service within the restaurant via your own device. TableTapp aims to ease restaurant expenses by easing the volume of waiting staff needed, combining all software solutions into one solution, and providing reliable and fast service to increase efficiency.

## 1.2 Scope & Goals

In order to effectively provide an all in one solution, TableTapp needs to incorporate each feature from the list of software solutions that restaurants use [sect. 1.1]. For the scope of ECE 499, our group can only implement a select set of those features. This is due to many limitations including the time constraint of 8 weeks, a workforce supply of only four group members, and other important time commitments such as other classes and work terms. Thus the select set of features we will be implementing will be limited to the customer ordering from the app use case. This will emulate a customer-waiter interaction in a restaurant. The basic breakdown in the lens of the three different stakeholders (user, front house admin, back house admin) for this interaction can be seen below in figures 1, 2, and 3.

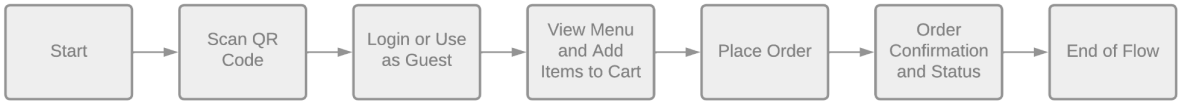


Figure 1 - Basic workflow of a customer user in TableTapp

In this workflow the user will be able to scan a QR code provided by the restaurant with their own smart device. The user will then be prompted to login to their account that they had previously set up or continue as a guest. Next the user has access to the restaurant's menu that they can select food and drink from that is added to their order (cart). Lastly the user reviews their order and confirms it and pays for the order. This action will add their order to the table's ticket which is to be sent off to the kitchen and can be viewed by the front of house as seen in the two figures below.

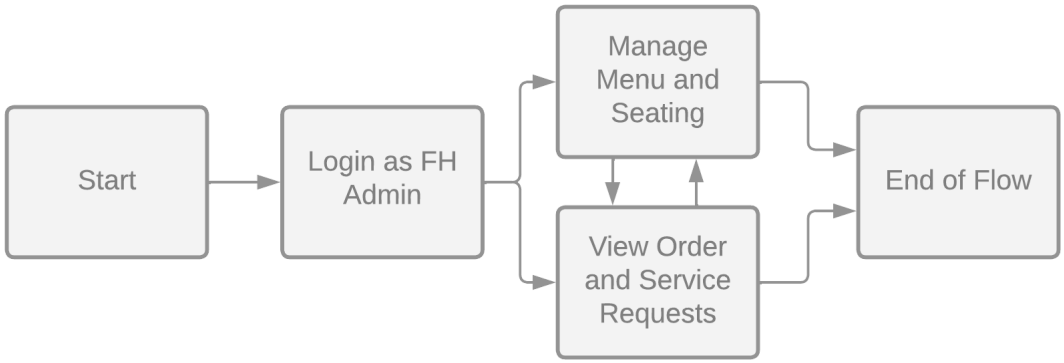


Figure 2 - Basic workflow of the front house user in TableTapp

The front house workflow is the simplest, as the goal of this product is to reduce the amount of front house staff needed to effectively run the restaurant. The front house would be run by a floor manager that monitors how full the restaurant is and takes any service requests the customers may have.

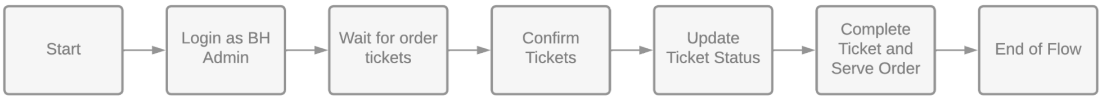


Figure 3 - Basic workflow of the back house use in TableTapp

Lastly the backhouse workflow is identical to that of a normal restaurant. The only change would be that the user's orders translate directly into order tickets that are viewed and confirmed by the back of house. The back of house also updates the status of the order ("prepping", "cooking", and "expo" etc.) that can be viewed by the customer on their device.

The implementation of these workflows define our goal for this project. Any features that need to be implemented in order to facilitate these workflows will be our final product for the ECE 499 presentation. Any further features are considered bonus features. The scope of this project is also limited to the implementation of these features. For further specification of specific features and requirements needed for these workflows to be implemented, please refer to table 1 below and [sect. 3.0] for specific details.

## 1.3 EGBC Principles

The development and creation of TableTapp follows EGBC code of ethics [4] for practicing engineers. Specifically principles 3 and 4 are the most applicable when developing this product.

> *Principle 3: have regard for the common law and any applicable enactments, federal enactments, or enactments of another province;*
>
> *Principle 4: have regard for applicable standards, policies, plans, and practices established by the government or Engineers and Geoscientists BC;*

Both these principles are directly applicable to TableTapp. The user login feature requires users to provide their personal login details (email and password) as well as payment information. The storage of users' personal details are protected by law and a minimum amount of security is required by law (principle 3) when storing sensitive user data. The product we create must also be created with ethical concern as small businesses may rely on the service output of this product. Misinforming about features and service quality detracts from the reputations, standards, and practices of engineers certified by EGBC (principle 4).

## 1.4 Potential Users

The user base is initially targeted towards small restaurant businesses in Victoria, BC. Any restaurant struggling with staffing or restaurants that want to make the leap to digitized ordering systems and robust all-in-one software solutions may also be in the list of potential users for this product.

# 2.0 Objectives

Small restaurants face significant challenges in keeping their costs down, which leads to frequent staff shortages at any given time. Unfortunately, this cost-cutting measure often restricts the restaurant’s spending capabilities and also results in a decline in the quality of service.

One significant contributing factor to the struggles of small restaurants is their dependence on multiple subscriptions of software solutions offering employee

management, inventory tracking, order placement, and transaction management. Although some all-in-one solutions do exist in the market, they often have reliability and security concerns surrounding them. The financial burden due to subscription services can be particularly challenging for small businesses with limited budgets, and the risk of system failures from data breaches can be detrimental to the restaurant's efficiency, customer satisfaction, and overall success.

To address these challenges and provide a cost-effective solution, the project for ECE 499 aims to develop a web application that is capable of emulating the interactions between restaurant staff and customers in a restaurant. This app serves as a customer-operated food ordering system, thus eliminating the need for additional waitstaff and consequently reducing labor costs for the business. The ability of the application to notify the kitchen staff promptly is aimed at increasing restaurant efficiency and reducing wait times for customers. By implementing this solution, small restaurants would be able to minimize their reliance on expensive labor and multiple software subscriptions. This would also allow these businesses to allocate their financial resources more efficiently, enhance the quality of their services, and improve the restaurant's ambiance.

On the other hand, customers can conveniently place their orders and make payments directly through the platform, eliminating the need for extensive manual involvement. This automation not only saves time but also ensures accurate and streamlined transactions. By reducing the reliance on traditional order-taking methods, the app significantly speeds up the ordering process, improving customer satisfaction and enabling the restaurant to serve more customers efficiently.

Table [1] shown below highlights the key objectives for achieving the goals of the food ordering application. These requirements have also been weighted based on the criticality of the features to ensure the successful operation of the application. Overall, the objectives listed below provide the key features that would address the current problems being faced by small restaurants and improve upon the shortcomings of existing solutions in the market.

Table [1]: TableTapp Requirements

| Requirement ID | Requirement | Weight (/10) |
|---|---|---|
| FR1 | User accesses the restaurant menu by scanning a QR code (presumably posted by the restaurant at the table). | 6 |
| FR2 | The QR code given by the restaurant is mapped to the table number in the restaurant's software system. | 6 |
| FR3 | Users should be able to add and view items in the cart before placing the order. | 8 |
| FR4 | The user should be able to place an order that will be received by the restaurant software. | 10 |

| | | |
|---|---|---|
| FR5 | The user should have the option to order food with an account the user set up or order as a guest. | 4 |
| FR6 | The restaurant should be able to inventory stock in the software system that will inform which items on the menu can be ordered. | 2 |
| FR7 | The front house/back house staff should be able to view and edit a user's order. | 5 |
| FR8 | The user should get a notification on the device they ordered from if an order was changed by anyone but them. | 2 |
| FR9 | The user will get a prompt to accept/decline/change an order upon receiving the order change notification described in [FR8]. | 1 |
| FR10 | Extending the editing of an order by the staff, the kitchen should be able to select “apply to all future orders” on edits. This would target the scenario that the kitchen has run out of an item. This would also have to update the user’s menus. | 4 |
| FR11 | The user should have the option to call for the service of a front house employee if they have any specific concerns. | 8 |
| FR12 | There should be a separate admin login for the front house that is linked to the restaurant system | 9 |
| FR13 | There should be a separate admin login for back house employees that is linked to the restaurant system. | 9 |
| FR14 | There should be a prompt for inputting payment methods after the user orders. | 4 |
| FR15 | The user should be able to view the order status of their order that is updated by the KDS software. | 5 |
| FR16 | The system should have reliable back-end software. | 10 |
| FR17 | The system should have an ergonomic UI that makes the functionality of the web app accessible to all users. | 10 |
| FR19 | The system should prompt the user to log in using their credentials. If the user has already logged in, re-using the QR code will bring them to their existing order | 7 |

| | | |
|---|---|---|
| FR18 | The system should have a reliable front-end software solution. | 10 |

# 3.0 Literature Survey & Specifications

The literature survey and specifications for this project will be divided into two categories. Since TableTapp is a web app solution the project will have a front end software and a back end software for the framework of the app. The design for the front end software will then be divided into three subcategories which include the different admin and user interactions with the project. The three subcategories include:

(1) Customer facing (user) view
(2) Floor and management staff (front house) view
(3) Kitchen and Inventory (back house) view

Below are the literature survey and specifications for the front end and back end software.

## 3.1 Software Solution

To design a robust full-stack application, developers need to carefully select a set of development frameworks that complement the development process. The choice of frameworks and tools depends on the project scope and the specific requirements.

In the case of our project, TableTapp, we conducted extensive research and ultimately opted for the MERN stack (MongoDB, Express.js, React, Node.js) to expedite the application development process.

When it came to developing the customer-facing aspect of the application, we thoroughly explored various approaches to ensure that the required specifications were met. The customer-end interface necessitated the inclusion of a login page, menu page, order page, and order status page. Many existing applications have already implemented these pages along with their respective user experiences and user interfaces (UX/UIs).

One such application that caught our attention was the McDonald's app, as it possessed a similar structure to what we envisioned for TableTapp.

The login page in TableTapp, inspired by the McDonald's app, will allow users to enter their credentials or register for a new account. A user-friendly form with validation will be implemented to ensure the security and integrity of user data.

The menu page, a crucial component of TableTapp, will allow customers to browse through the available food and beverage options. Drawing from the McDonald's app, we will organize the menu items into categories such as appetizers, main courses, desserts,

and drinks. Each item will be displayed with an appealing image, a brief description, and the corresponding price. To enhance the browsing experience, we will incorporate filtering and sorting options based on different criteria such as popularity, price, and dietary preferences.

For the order page, we will design a simple and intuitive interface that allows users to add items to their cart, customize their orders (if applicable), and specify any special instructions or dietary requirements. We will include a summary section that displays the selected items and their quantities, as well as the total order amount. Users will be able to easily modify their order or proceed to the checkout process.

The order status page will be another important feature. This page will provide real-time updates on the status of the customer's order. Customers will be able to track the progress of their order, from preparation to delivery, and receive notifications when their order is ready to be served.

To manage the application and its associated data, a strong backend server is crucial for a smooth data flow between the user and the database. The backend will define and implement robust APIs (Application Programming Interfaces) to facilitate smooth communication between the front-end and back-end components. These APIs will handle requests from the front-end, process data, interact with the database, and provide appropriate responses.

## 3.2 Front End Design

### 3.2.1 User Interface Design

Food delivery applications have become increasingly popular, providing customers with convenient access to a wide range of restaurants and offering restaurants an additional channel for reaching customers. We will explore the pros and cons of three prominent food delivery applications: DoorDash, SkipTheDishes, and Uber Eats, from both the customer and restaurant perspectives.

Table 2: Door dash Pros and Cons [5]

| **Pros** | **Cons** |
|---|---|
| **Restaurant** | |
| Wide customer base and increased exposure for restaurants. | High commission fees, which can significantly impact restaurant profitability. |
| Comprehensive order management systems to streamline operations. | Lack of control over the delivery process and customer interactions. |

| | |
|---|---|
| Marketing support, including featured placements and sponsored listings. | Potential for negative customer reviews impacting restaurant reputation. |
| Positive customer reviews and ratings can boost restaurant reputation. | Technical glitches and occasional order errors can lead to customer dissatisfaction. |
| Opportunities for additional revenue streams. | |
| **Customer** | |
| Extensive restaurant selection, including local and national chains. | High service and delivery fees, which can make it less cost-effective. |
| User-friendly interface and intuitive app navigation. | Occasional order accuracy issues, such as missing items or incorrect orders. |
| Timely and reliable delivery service with real-time order tracking. | Limited availability in some smaller cities or rural areas. |
| Regular promotions and discounts to enhance affordability. | |
| Effective customer support and issue resolution. | |

Table 3: Skip the Dishes Pros and Cons [5]

| **Pros** | **Cons** |
|---|---|
| **Restaurant** | |
| Expanded customer reach and increased visibility for restaurants. | High commission fees and delivery charges affecting profitability. |
| Detailed analytics and reporting tools for performance tracking. | Order accuracy issues and challenges in managing large volumes of orders. |
| Marketing opportunities through promotional features and campaigns. | Potential delays and longer delivery times during peak periods. |
| Customizable menus and options for efficient order management. | Limited control over the delivery process and customer interactions. |

| | |
|---|---|
| Additional revenue streams for participating restaurants. | |
| **Customer** | |
| Wide selection of restaurants, including many local eateries. | Higher delivery fees compared to some competitors. |
| User-friendly interface with advanced search filters and dietary options. | Longer delivery times during peak hours or in certain areas. |
| Flexible delivery options, including scheduled deliveries. | Limited restaurant availability in smaller cities or less populated regions. |
| Loyalty programs and rewards for frequent customers. | |
| Good customer support and responsiveness. | |

Table 4: UberEats Pros and Cons [5] [6]

| **Pros** | **Cons** |
|---|---|
| **Restaurant Side** | |
| Large customer base and increased market exposure for restaurants. | High commission fees, impacting restaurant margins. |
| Seamless integration with the Uber platform, leveraging existing user base. | Challenges in maintaining order accuracy and timely delivery. |
| Efficient order management systems and streamlined operations. | Limited control over the delivery process and customer interactions. |
| Marketing and promotion opportunities through targeted campaigns. | Technical issues or app outages affecting order management. |
| Option to leverage Uber's delivery fleet for streamlined logistics. | |
| **Customer Side** | |
| Extensive coverage with a large network of participating restaurants. | Surge pricing during high-demand periods, leading to increased costs. |

| | |
|---|---|
| Easy-to-use interface and seamless integration with the Uber app. | Some customers have reported issues with order accuracy and missing items. |
| Fast delivery service with real-time tracking and estimated arrival times. | Limited availability in certain rural or remote areas. |
| Regular promotions and discounts for cost-conscious customers. | |
| Reliable and efficient customer support. | |

While these platforms offer convenience, a wide selection of restaurants, and efficient delivery services, they also come with challenges such as high fees, order accuracy issues, and limited control over the delivery process. Understanding the pros and cons from comparable platforms helps to shed light on possible problem points in TableTapp that can be addressed before deployment.

### 3.2.2 Back House Interface Design

Back house kitchen display software (KDS) refers to a restaurant assistance software designed to streamline and enhance the operations of a restaurant's (back house). It encompasses a range of features and functionalities, such as inventory management, recipe management, order tracking, and staff scheduling, all aimed at improving efficiency, reducing errors, and maximizing productivity in the kitchen. This software helps in automating various tasks, optimizing ingredient usage, minimizing waste, and facilitating seamless communication among kitchen staff, ultimately leading to smoother operations and improved customer satisfaction.

On inspection of current KDS software, platforms such as Toast KDS [7] offer a visually distinctive and easy to read GUI for their KDS. Toast offers features for: order ticketing, ticket time, table number, ticket status, menu item (including item modifications) and a history of previously completed tickets. Lightspeed KDS [8] also include these features but include additional statistics on average time to order completion and amount of tickets currently in progress. Lightspeed is a software based solution which can be run on multiple smart devices and different operating systems and Toast requires a purchase of a dedicated device to host the KDS software.

For our purposes the KDS system should be connected with the customer ordering software in two ways. Firstly when the table places an order ticket, the kitchen should be able to see the newly created ticket in the KDS. Next the kitchen should be able to change the status of the ticket from, "not started" to "started" to "expo" (finishing touches and sides) and that status can be viewed by the table that placed the order ticket. For efficiency in the kitchen, the KDS should display tickets that have specific orders and specific items from the menu displayed as well as any modifications made to the order (

such as “no tomato” or “additional sauce” etc.). Lastly the KDS should include a system that timestamps the ticket so the kitchen knows how long a ticket has been active for.

### 3.2.3 Front House Interface Design

The major operations of the restaurant such as inventory tracking, table management, employee management, food ordering, and purchasing are handled mostly by the staff working in the lobby and dining area of a restaurant, also known as “front of house” (FOH). These operations are usually facilitated through Point of Sale (POS) and management apps that the restaurant can subscribe to, which are not only expensive but also require staff to undertake special training for each of the applications the restaurant chooses to use [9]. Although the cost of these apps may be viewed as an investment by large-scale restaurants, local businesses usually cannot afford to incur such costs for their daily operations. Moreover, the shortage of staff at small businesses proves to be another major constraint to their success, which depend heavily on building a reliable customer base [10]. To solve these challenges surrounding local businesses and to prevent them from incurring the cost of hiring additional skilled workers, the FOH system built into TableTapp relies on combining the features of existing management apps with POS systems while providing a user-friendly experience to restaurant workers.

One of the major reasons for small-scale businesses to opt out of using software-based POS systems is their expensive subscription plans. With the global POS market being valued at approximately 12 billion USD in 2022, the vendors for these services are now able to charge additional prices for updates to their systems [11]. This not only further puts small-scale businesses at a disadvantage, but also prevents them from investing in resources that would increase their productivity. Due to this, the TableTapp FOH system is built on a web interface. This enables free upgrades to users while maintaining the productive interface offered by software-based services. Further, the design of the FOH system has been selected such that restaurant staff can use it without the need for any additional training. The use of informative headings and easily understandable tabs would aid in bridging any skill gaps that may exist in the workforce while allowing restaurants to customize the components of the FOH system based on the needs of their business.

Another key feature of the TableTapp FOH system is the ability for restaurant owners to view the analytics for their business and devise effective strategies to promote their growth. Restaurants would be able to keep track of their customers using an encrypted and secured database and offer incentive programs to retain their existing market.

# 4.0 Team Duties & Project Planning

Development of the application follows a scrum sprint methodology with two week sprints spread over two and a half months. We use the software platform Trello to keep track of all of our tasks from week to week. Sprints are planned at the weekly sprint planning meeting after a standup and discussion of possible problems, then tickets are drafted and put in the backlog. Once everyone has agreed on the tickets and their priority

members are assigned to the tickets under their milestone. Members' roles are shown below.

Table 7: Member roles

| **Member** | **Roles** |
|---|---|
| Kris | Customer, Scrum master, technical support, UX/UI, Full Stack Development |
| Kai | Back of house, Graphic Designer, Front-end developer |
| Aryan | Front of house, Market Researcher, Front-end developer |
| Adil | Backend developer |

## 4.1 Milestone breakdown

There are 4 main milestones each of which have sub milestones.

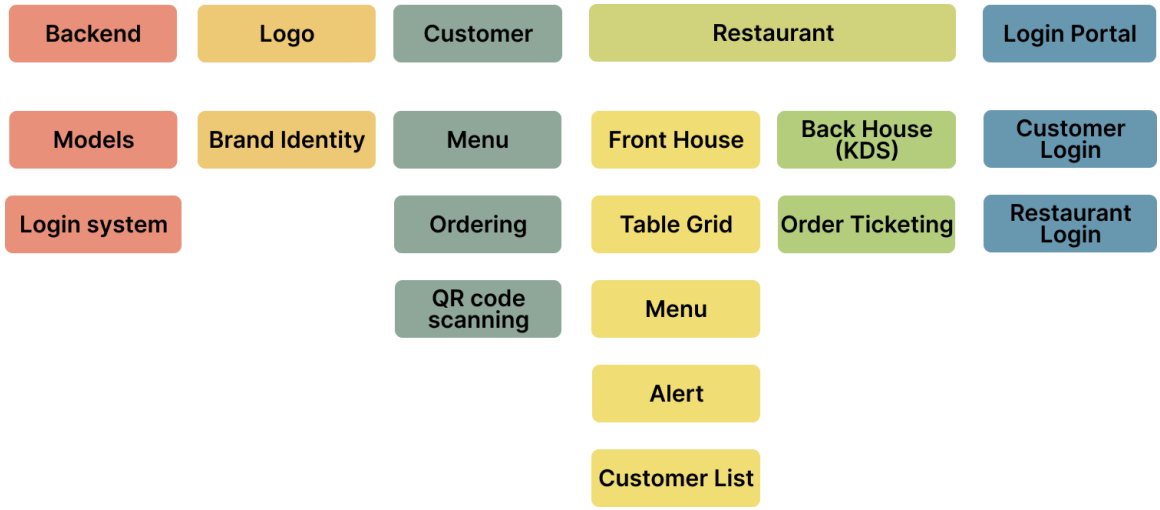


Figure 4: Milestone breakdown

Customer Sub Milestones:

The customer design is based on market research of similar applications in the industry namely food delivery and Fast Food. Since the target market is North American the applications chosen for market research all have a large market share in the Americas. UI wireframes are designed using the chakra UI library while taking key layouts and work

flows from the market research. The customer side as follows the workflow in figure K1 is broken up into 6 main views.

(1) Scanner & check in
(2) Menu
(3) Cart
(4) Item details
(5) Waiting on others
(6) Order Status

Implementation of each view is done using Typescript React and the Chakra UI library. The first view to be implemented is to be the qr code scanner view which acts as the entry point for the app. Next the menu, item details and cart are to be completed, all of which will have data pulled from the corresponding database model. Finally the login, order status, and order waiting views will be completed.

Front of House Sub Milestones:

The front of house design is based on market research of existing products and the desired features as explained during interviews with industry professionals. The UI mock-ups are designed on Figma and provide a template for the finalised design, which is programmed using Typescript React. The development process is hierarchical, starting with the most crucial requirements. The Menu page is implemented first as this would allow the restaurant staff to manage the items available to customers. Following this, the Table page is designed to allow the staff to manage logistics surrounding each table in the restaurant. Once this is completed, the Alerts page is to be designed to allow the restaurant staff to take requests from customers who need assistance. Finally, the Customers page is to be designed that would allow the restaurant owners to run analytics on their business. Throughout the design process, the template designed using Figma is used and the implementations are made available to the team for further reviews and suggestions.

Back of House Sub Milestones:

Back of house entails designing a UI that the kitchen staff can view and update. This includes all the specifications listed in [sect. 3.2.2] above. To complete the backhouse models milestone the interface UI with the full suite of specifications for this scope will have to be completed. This includes:

- Ticketing receiving and notification tab section
- Ongoing ticket order interface
- Timestamp data implementation on ticket orders
- Visual layout planning for KDS interface (done in Figma)

Users will also need a kitchen staff login interface (the same as front of house interface) to access the KDS system, this is another milestone Implementing these will be done

using Typescript React and planned in Figma. The UI elements of the back of house view will follow with the rest of the software and use Chakra UI elements. The ticket viewing UI will be designed first and then categorization of the tickets next. Finally timestamping and kitchen login will be implemented.

### 4.2 Bottlenecks

The major bottlenecks that have been identified at this juncture are as follows. The largest hindrance is the majority of the team learning the stack. This has added on learning time into the equation and was taken into account when planning out the first couple sprints. Along the similar thread we have not obtained a client as of yet which makes testing the application in real life circumstances more difficult. To remedy this, random users are surveyed with a build of the app after every sprint to get insight that the team might not have. Additionally restaurant workers are also being surveyed to gather meaningful feedback that will have an impact on the end users UX experience. Yet another barrier is the time constraints on the project. Platforms of this scale are usually not developed in just under 3 months, however the team has narrowed the scope to minimum viable product in hopes of having a full functional prototype for July 28th.

Some more technical bottlenecks  such as advanced querying and image storage are not ideal but are not critical to the functionality of the application. Certain workflows in our application will require more advanced querying than what our current api is set up for. For example when a customer first scans the QR code the customer needs to be added to the table model, and a new order connected to the table must be created they cannot be done with one query. It is best practice not to store images in MongoDB and based on our 512MB of storage that would not be advisable. Therefore, another hosting service must be used to store the images and mongoDB will hold a reference to their location.

Overall these bottlenecks do not stall the project in any way however they do add complexity to the planning of tasks.

## 5.0 Milestones Completed & Progress Made

### 5.1 Milestones achieved

The following milestones have been completed:

- Logo
- Backend Model Creation
- Customer menu
- QR code scanning
- Customer ordering

Note that the first week was intended for planning, milestones will be completed at a more rapid pace now that planning is done.

## 5.2 Progress made

### 5.2.1 Planning and Refining (Ongoing)

Creating an idea and making a plan for a product is one of the most crucial, the Prateno principle of design states that 80% of your time should be spent planning while the remaining 20% is spent on execution. We have completed the planning phase of our project which includes:

(1) Initial idea creation meeting
(2) Requirements planning and weighting
(3) Development stack choice (MERN)
(4) Backend creation
(5) Frontend creation
(6) Initial interfaces for menu and front of house implemented
(7) QR code support
(8) Work log 1 and progress report 1

### 5.2.2 UX/UI Design

Front of House Design

For the implementation of the front of house system, first a mock-up of the design UI was created. This mock-up incorporated the desirable features of a restaurant FOH system, as found in the market research conducted by the team. Following the initial design mock-up, a finalised design template was created, which gave more insight into precise UI features and pop-up windows seen by the users upon button clicks. Once this was done, the design moved into the implementation phase, starting with the Menu page for the FOH system.

The Menu page has been developed using React and Typescript in the 2nd sprint of the project. It is primarily responsible for displaying the current menu items, which include an item name, item description, its corresponding picture, and its corresponding price. The user can edit existing items by clicking on them, or add new items to the menu using a button click and following the same template as existing items. On top of this, the page is also responsible for displaying a summary of the orders from various tables in the restaurant, which can be expanded by a button click to reveal a more in-depth view of the items ordered by a table and the respective receipt. Currently, these features have all been

incorporated into the design, and require the layout to be optimized and kept consistent with the rest of the design elements of the application, such as themes, fonts, and colours.

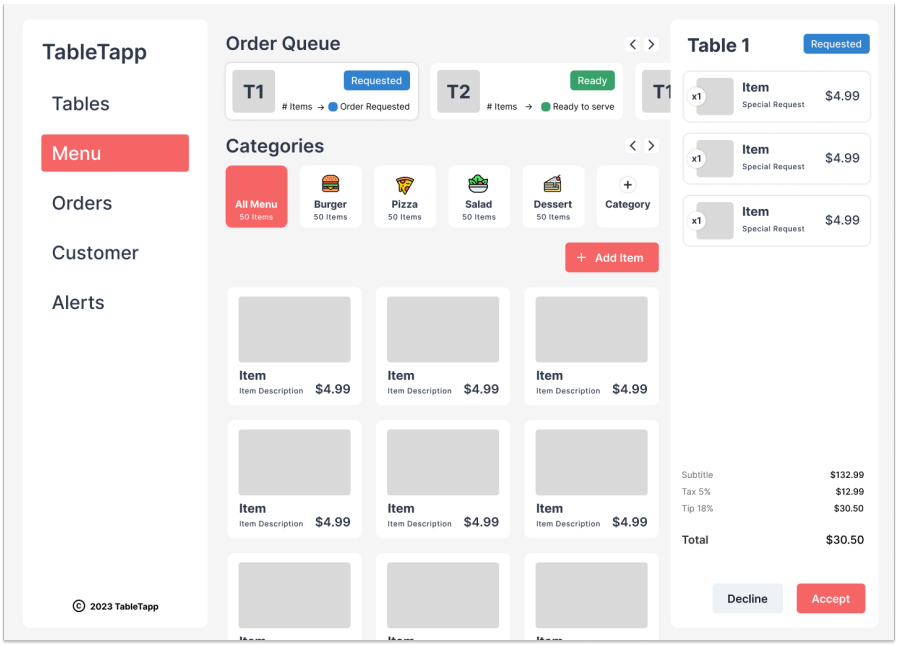


Figure 5 - Restaurant front of house menu view

In addition to the Menu page, the development of the Tables page of the FOH system has also begun, and its implementation is expected to be completed by the beginning of the 3rd sprint. Once the required changes are made to the Menu page and the design theme has been applied to the Tables page, the remaining pages will be developed in a similar manner during the 3rd and 4th sprints of the projects to ensure that all necessary deliverables are met.

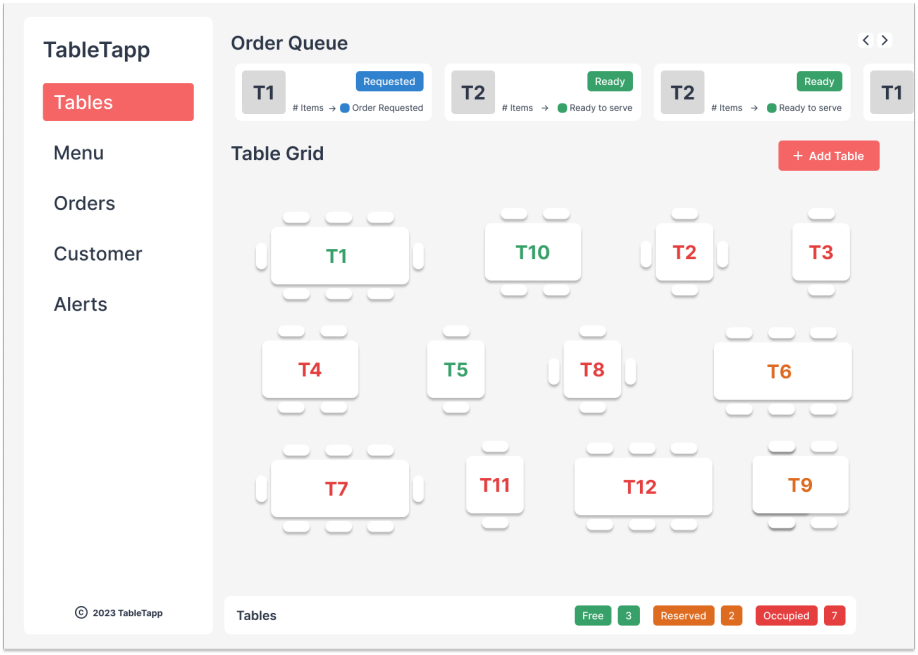


Figure 6 - Restaurant front of house table view

Back of House

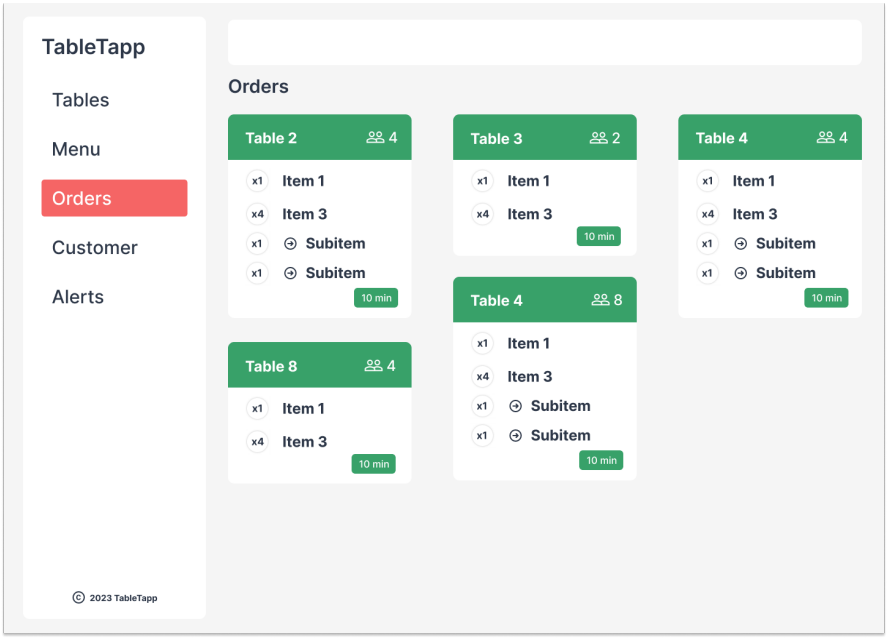


Figure 7 - Restaurant back of house orders view

Customer Design

The Design for the customer side was first designed chakra UI components with the layout was mainly inspired by doordash figma. For ease of implementation and

component reusability the major views use a table like component for layout which can be seen below in figure 8. The React implementation of the design for the menu, cart, item details and scanner have all been completed, along with api connection to the database.

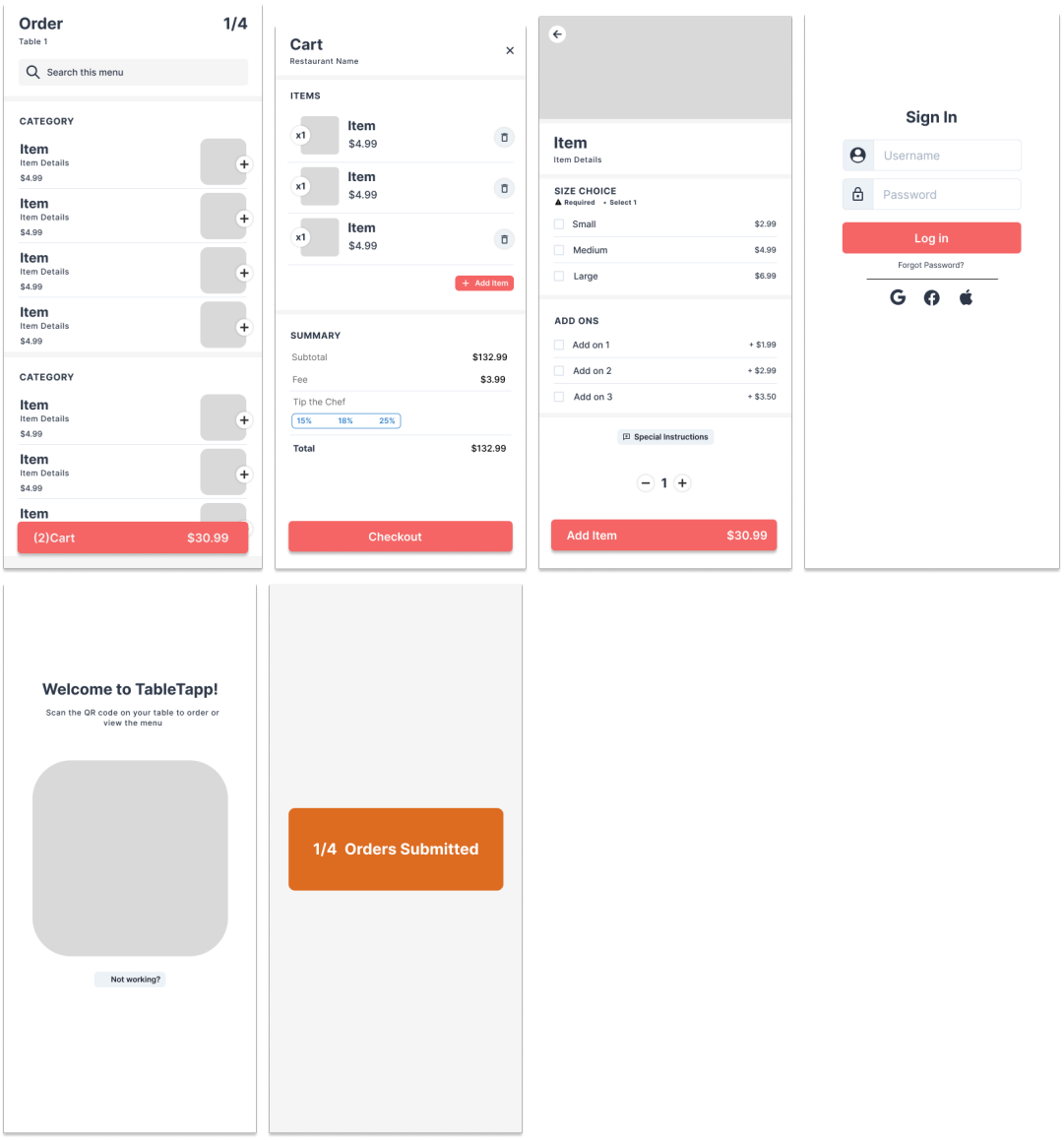


Figure 8 - Customer views UI

Data Models

The application leverages MongoDB as its primary database for storing customer, vendor, and order-related data. In order to establish communication between the application's backend and the database, data models need to be devised to ensure proper parsing and understanding by the database. These data models were among the initial key components designed during the app development process, as they form the foundation

for most other functionalities. The team has developed multiple data models, each dedicated to representing a specific aspect of the application and appropriately storing the corresponding data. The following data models have been introduced:

1. Customer Model
2. Vendor Model
3. Order Model
4. OrderItem Model
5. Item Model
6. Menu Model
7. AddOns Model
8. Category Model
9. Cart Model

By implementing these data models, the application can effectively organize and manage customer, vendor, order, and menu-related information, facilitating smooth operations and enhancing the overall user experience.

# 6.0 Summary & Future Work

Small restaurants face significant challenges in managing their operations and maintaining profitability. The costs associated with traditional restaurant management systems and the need for a large staff often eat into their profit margins, making it difficult for these businesses to grow. In order to address this issue, our team has developed an innovative all-in-one solution application, TableTapp.

TableTapp aims to eliminate the high overhead costs of using multiple software systems and the burden of maintaining an excessive number of staff members. By leveraging automation technology, TableTapp streamlines the order placement and payment processes, minimizing the need for unnecessary customer/staff interactions. This not only helps reduce operational costs but also enhances the overall efficiency of the business.

Our team just finished the customer-side development of the application, and our next objective is to work on integrating the front and backhouse interfaces over the course of the next three sprints. Simultaneously, we will be focusing on building the back-end infrastructure responsible for managing the routing and data flow of the application. Additionally, we have plans to incorporate an authentication system into the application, enabling users to securely log in and access their personalized accounts. This will enhance security and provide a personalized user experience.

# 7.0 References


[1] T. Segal, “Profit margins for the food and beverage sector,” Investopedia, https://www.investopedia.com/ask/answers/071015/what-profit-margin-usual-company-food-and-beverage-sector.asp (accessed Jun. 23, 2023).

[2] G. McIntyre, “Why B.C. restaurants are still suffering post-pandemic,” Vancouver Sun, https://vancouversun.com/business/local-business/why-bc-restaurants-are-still-suffering-post-pandemic (accessed Jun. 24, 2023).

[3] Nicolette Kier As a small business owner, “How much do restaurant Pos Systems Cost & How to choose in 2023,” Merchant Maverick, https://www.merchantmaverick.com/how-to-choose-a-restaurant-pos-system/ (accessed Jun. 23, 2023).

[4] “Code of ethics,” Engineers and Geoscientists BC, https://www.egbc.ca/Complaints-Discipline/Code-of-Ethics/Code-of-Ethics (accessed Jun. 23, 2023).

[5] M. King and R. Delucci, “Third-party delivery services: Pros/Cons & How They Work,” Fit Small Business, https://fitsmallbusiness.com/pros-cons-third-party-delivery-services (accessed Jun. 23, 2023).

[6] E. Team, “The Pros and cons of becoming an uber eats restaurant partner,” Escoffier, https://www.escoffier.edu/blog/food-entrepreneurship/the-pros-and-cons-of-becoming-an-uber-eats-restaurant-partner/ (accessed Jun. 23, 2023).

[7] Kitchen Display Systems (KDS) | toast pos, https://pos.toasttab.com/hardware/kitchen-display-system (accessed Jun. 24, 2023).

[8] “Kitchen display system,” Lightspeed, https://www.lightspeedhq.com/pos/restaurant/kitchen-display-system/ (accessed Jun. 23, 2023).

[9] “3 reasons why your employees hate your pos system,” Logic Controls, https://logiccontrols.com/3-reasons-employees-hate-pos-system/ (accessed Jun. 23, 2023).

[10] A. Lucas and I. Krietzberg, “Restaurants are short-staffed, and that’s taking a big toll on customers and workers alike,” CNBC, https://www.cnbc.com/2022/07/17/customer-service-suffers-at-short-staffed-restaurants-as-covid-takes-toll.html (accessed Jun. 23, 2023).

[11] “Point-of-sale software market size & share report, 2030,” Point-of-Sale Software Market Size & Share Report, 2030, https://www.grandviewresearch.com/industry-analysis/point-of-sale-pos-software-market (accessed Jun. 23, 2023).