

Final Report

Wait Management System



UNSW
SYDNEY

z5315885	Ho Wan Hou	z5315885@ad.unsw.edu.au	Scrum Master
z5405329	Puquan Chen	z5405329@ad.unsw.edu.au	Developer
z5275224	Wei Chen	z5275224@ad.unsw.edu.au	Developer
z5338023	Weizhe Pan	z5338023@ad.unsw.edu.au	Developer
z5387750	Qifan Zhao	z5387750@ad.unsw.edu.au	Developer

Course Code: COMP9900

Course Title: Information Technology Project, 2023 T2

Lab: 9900W18B

Team: Gogogo

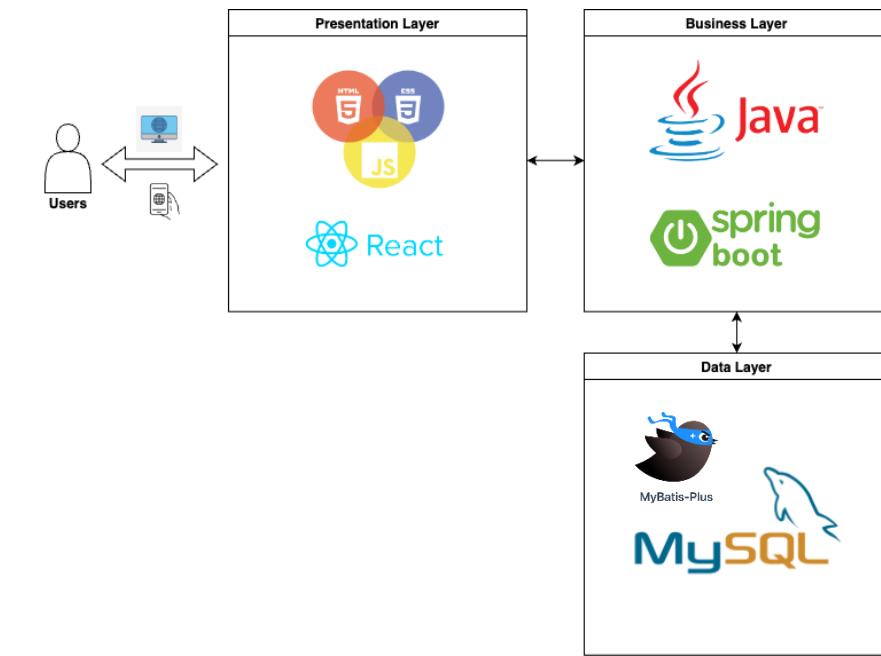
Report Submission Date: 4/8/2023

Table of Contents

1	Overview	1
1.1	System Architecture	1
1.1.1	Presentation Layer	1
1.1.2	Business Layer	1
1.1.3	Data Layer.....	2
1.2	Page Functionalities Outline	3
1.3	Project Objectives	3
2	Functionalities developed	4
2.1	Manager - Add table.....	4
2.2	Manager - Add category.....	5
2.3	Manager - Delete Category	6
2.4	Manger - Move category order.....	7
2.5	Manger - Add dish.....	8
2.6	Manger - Modify dish	9
2.7	Manger - Delete Dish	10
2.8	Customer - Select table	11
2.9	Waitstaff - Create the template for table	12
2.10	Customer - View menu	12
2.11	Customer - View details of specific dish and add dish to cart	13
2.12	Customer - View current order, Delete Dish and Place order.....	14
2.13	Waitstaff - View order information.....	15
2.14	Kitchen - Create order template, Cook, Flag once finished	16
2.15	Waitstaff - Receive dish status, serve the dish.....	18
2.16	Customer - Ask for help	19
2.17	Waitstaff – Respond to the help request.....	19
2.18	Customer - Request bill and finish meal, Provide rating	20
2.19	Waitstaff – Confirm payment.....	21
2.20	Manager - View statistics.....	22
2.21	Customer – View top rated and best sellers	25
3	Third-Party functionalities	25
3.1	Front-end	25
3.2	Back-end.....	26
3.2.1	Mybatis-plus	26
3.2.2	Postman.....	26

4	Implementation challenges	26
4.1	Front-end	26
4.1.1	Co-Debugging Frontend and Backend.....	26
4.1.2	Familiarizing with a new UI library: Ant Design	27
4.1.3	Enhancing Program Efficiency with React Hooks	27
4.2	Back-end.....	27
4.2.1	Deletion in Database	27
4.2.2	Table Join Query	27
4.2.3	Table Join Query.....	27
5	User Manual.....	28
5.1	Front-end	28
5.2	Back-end.....	28
6	Reference	30
7	Appendix.....	31
7.1	User Stories in Proposal	31

1 Overview



1.1 System Architecture

1.1.1 Presentation Layer

In the presentation layer, we have implemented the frontend using React, an immensely popular and robust open-source JavaScript library. React's principal advantage is its ability to create engaging and efficient user interfaces through a concept known as the virtual DOM, which enhances web performance by dynamically updating and rendering UI components. The declarative nature of React simplifies the development process and promotes reusability and high maintainability of code. This leads to the creation of responsive, interactive interfaces with minimal coding.

The framework is supported by a vibrant ecosystem, providing us with a rich JavaScript library that further offers flexibility in designing our frontend. Therefore, utilizing React for the presentation layer allows us to deliver an improved application speed and an enhanced user experience.

1.1.2 Business Layer

Our core framework for the business layer is Spring Boot. We use it to develop both business logic and expose RESTful APIs, essentially merging the business and interface layers in our architecture. With the help of Spring Web MVC, a module within the Spring Boot framework, we can swiftly design and implement these APIs.

For interacting with the database, Spring Data JPA is employed. This powerful tool provides a user-friendly object-relational mapping (ORM) methodology, eradicating the necessity for complex SQL queries. It lets us carry out CRUD operations smoothly and ensures security against SQL injection.

To foster a well-structured and maintainable codebase, we implement the Spring Boot Application Factory pattern. This enhances our ability to distribute logic across different files and modules, fostering more organized, readable, and collaboration-friendly code.

By exploiting the versatility of Java and the integrated capabilities of the Spring Boot framework, which include offering RESTful APIs and managing business logic, we successfully construct a sturdy and efficient business layer that substantiates our system's overall functionality.

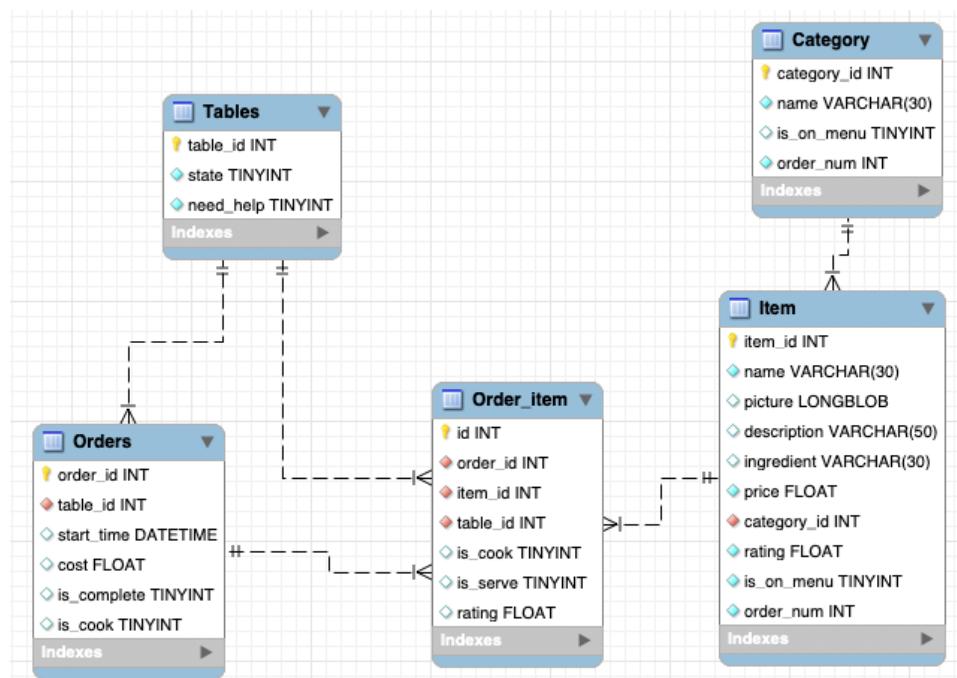
1.1.3 Data Layer

The data layer of our system heavily relies on MySQL and MyBatis Plus, integrated with Java, to efficiently manage and handle structured data. MySQL, a robust open-source relational database management system, is our primary choice for data storage. It's widely recognized for its scalability, performance, and the capacity to manage large volumes of structured data. MyBatis Plus, as an enhancement of the MyBatis persistence framework, simplifies the mapping between SQL queries and Java code. It automates CRUD operations, offers dynamic SQL generation, and supports pagination, which significantly enhances the productivity of developers and allows efficient manipulation of data.

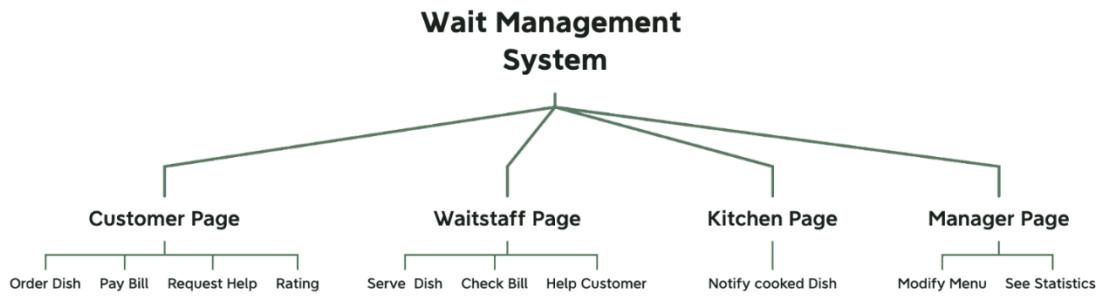
Being highly compatible with MyBatis Plus and Spring Boot, MySQL provides a seamless development experience. Java's platform-independence along with MySQL's extensive range of functionalities promotes robustness in our system. Moreover, the integration of MySQL with Spring Boot using MyBatis Plus streamlines the process of database-related operations. It abstracts complex configurations and installations, freeing developers to focus more on implementing business logic and handling database modifications.

Spring Boot enhances our data layer management with tools such as Flyway. These tools facilitate the smooth migration and evolution of database schemas while preserving data integrity. This becomes extremely useful during updates and modifications, making our data layer more maintainable and adaptable to changes.

Through integrating these technologies, our data layer can adeptly handle the intricacies of data operations, thereby delivering applications of good quality, efficiency, and security. This technological combination is particularly suitable for modern web and enterprise applications, which often demand complex data handling and operations.



1.2 Page Functionalities Outline



1.3 Project Objectives

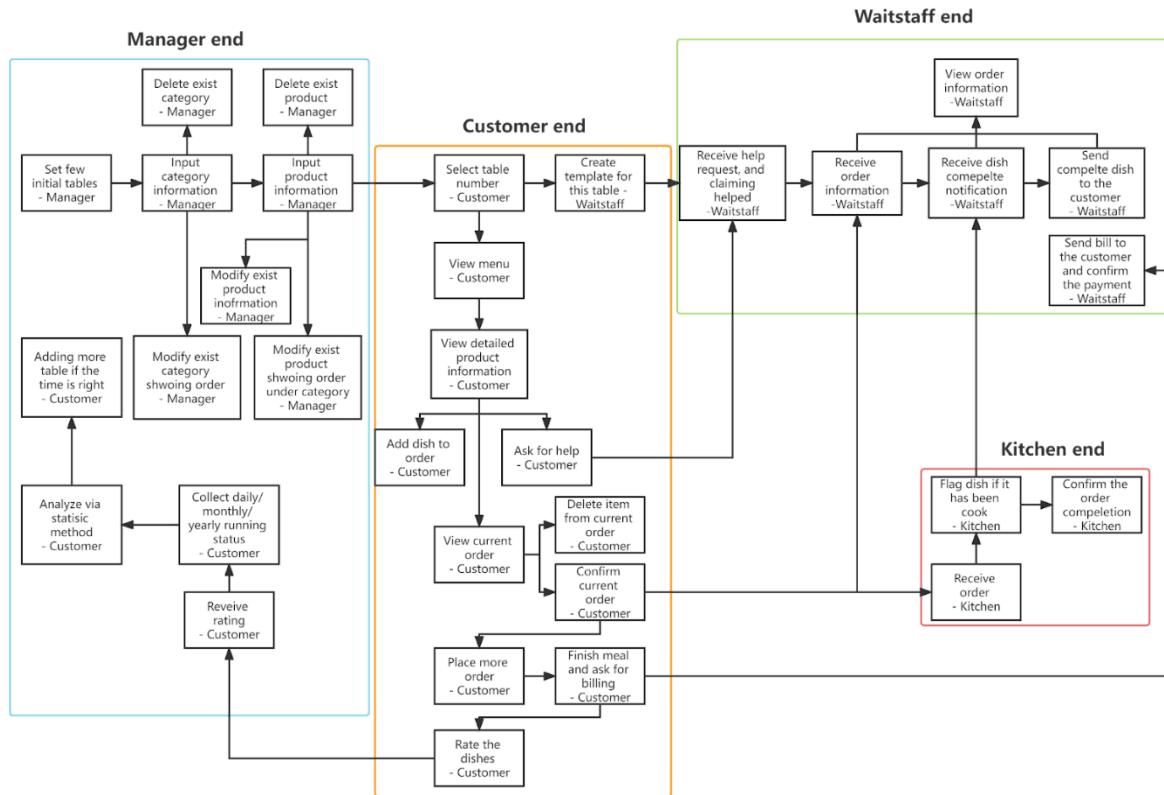
The Wait Management System aims to streamline order placement for customers, minimizing the need for staff assistance and reducing miscommunication.

1. **Customer Menu Browsing:** Customers can navigate the menu interface, where items are organized into different categories. A clear and user-friendly digital menu enhances customer's ordering experience.
2. **Customer Ordering:** Customers can select table, order items from the menu, providing them with the flexibility to choose and order their desired items at any time during their dining experience. This bypass the need to wait for a waiter's assistance, reducing the waiting time for customers.
3. **Customer Request:** Customers can request for help during the meal and request their bill once the meal is complete. They can also view the items they ordered, along with their prices, and the total amount on the bill. This enhances their dining experience.
4. **Kitchen Staff System:** The kitchen staff will have access to a time-sorted order list displayed on the screen. This allows the kitchen staff to have a clear overview of the items that need to be prepared and prioritize orders accordingly. As a result, efficiency is improved and the waiting time for customers is reduced.
5. **Waitstaff System:** Allows waitstaff to provide efficient and timely service to customers. Waitstaff can access detailed information about customers' orders, allowing them to accurately deliver the right dishes to the respective tables. They can mark notifications as completed once they have successfully served the dishes.
6. **Notification System:** Improve communication among customers, waitstaff and kitchen staff. Customers can use this system to request assistance and waitstaff will see the notifications on the waitstaff system. Kitchen staff can use this system to send the notification to wait staff to remind them the order is ready. This system improves the efficiency of communication.
7. **Manager System:** Allows manager to manage and customize the menu.
8. **Rating and Recommendation System (Novel):** Customers can rate the dishes they ordered. To assist in decision-making, the menu will display the top rated and best selling items.
9. **Analytics (Novel):** Managers can gain a comprehensive understanding of the restaurant's sales performance, enabling them to plan more effectively for the future.

User stories related to each objective are presented in Appendix 7.1.

2 Functionalities developed

In this section, we will showcase the functionalities that have been successfully implemented by presenting a step-by-step demonstration of ordering items from the menu. The process begins with setting up the menu through the manager interface.



2.1 Manager - Add table

(Addresses objective 7: Manager System)

Manager Page

Menu

- Burger
- Noodle
- Hot Pot
- Drink

Burger

Dish Name	Price	Rating	Action
Veggie Burger	\$19	★★★★★	
Pork Burger	\$12	★★★★☆	
Turkey Burger	\$10	★★★★★	
Lamb Burger	\$12	★★★★★	

Menu

- Burger
- Noodle
- Hot Pot
- Drink

Burger

Image	Name	Price	Rating	Action
	Veggie Burger	\$19	★★★★★	
	Pork Burger	\$12	★★★★☆	
	Turkey Burger	\$10	★★★★★	
	Lamb Burger	\$12	★★★★★	

The manager can set the number of tables for the restaurant using the manager's interface. By accessing the manager's homepage, they can add tables using the "Add Table" button. The system will update and display the current total number of tables available in the restaurant.

2.2 Manager - Add category

(Addresses objective 7: Manager System)

Manager Page

Burger

Image	Name	Price	Rating	Action
	Veggie Burger	\$19	★★★★★	
	Pork Burger	\$12	★★★★☆	
	Turkey Burger	\$10	★★★★★	
	Lamb Burger	\$12	★★★★★	

Manager Page

Burger

Add New Category

* Category:

Submit

Image	Name	Price	Rating	Action
	Veggie Burger	\$19	★★★★★	
	Pork Burger	\$12	★★★★☆	
	Turkey Burger	\$10	★★★★★	
	Lamb Burger	\$12	★★★★★	

Manager Page

Menu

- Burger
- Noodle
- Hot Pot
- Drink
- new category**

At the manager's end, they can easily add a new category to the menu by clicking on the dedicated "Add Category" button on the manager's homepage. Once the manager confirms the creation, the menu will automatically refresh, showing the updated menu with the new category.

2.3 Manager - Delete Category

(Addresses objective 7: Manager System)

Manager Page

Menu

- Burger
- Noodle
- Hot Pot
- Drink
- new category**

Manager Page

Menu

- Burger
- Noodle
- Hot Pot
- new category**

Menu

- Burger
- Drink
- Noodle
- Hot Pot

Page

+ Add New Dish + Add New Category + Add Table ↵ Statistics

Burger

Burger Type	Price	Rating	Action
Veggie Burger	\$19	★★★★★	trash
Pork Burger	\$12	★★★★★	trash
Turkey Burger	\$10	★★★★★	trash
Lamb Burger	\$12	★★★★★	trash

The manager can easily delete a category from the menu by clicking on the "trash" button below the category they want to remove. Once confirmed, the menu will automatically refresh, showing the updated menu without the deleted category.

2.4 Manager - Move category order

(Addresses objective 7: Manager System)

Manager Page

+ Add New Dish + Add New Category + Add Table ↵ Statistics

Burger

Burger Type	Price	Rating	Action
Veggie Burger	\$19	★★★★★	trash
Pork Burger	\$12	★★★★★	trash
Turkey Burger	\$10	★★★★★	trash
Lamb Burger	\$12	★★★★★	trash

Manager Page

+ Add New Dish + Add New Category + Add Table ↵ Statistics

Drink

Drink Type	Price	Rating	Action
Coke	\$5	★★★★★	trash
Sprite	\$5	No Rating	trash
Canada Dry	\$3	★★★★★	trash

Manager Page

+ Add New Dish + Add New Category + Add Table ↵ Statistics

Burger

Burger Type	Price	Rating	Action
Veggie Burger	\$19	★★★★★	trash
Pork Burger	\$12	★★★★★	trash
Turkey Burger	\$10	★★★★★	trash

The manager can easily change the order of categories on the menu using a simple dragging method. They can grab the category name they want to move, drag it to the desired position, and release the cursor. The menu will then automatically refresh, showing the updated order of categories.

2.5 Manager - Add dish

(Addresses objective 7: Manager System)

Manager Page

Menu

- Drink
- Burger
- Noodle
- Hot Pot

+ Add New Dish + Add New Category + Add Table ↵ Statistics

Drink

	Coke	Sprite	Canada Dry
Image			
Name	Coke	Sprite	Canada Dry
Price	\$5	\$5	\$3
Rating	★★★★★	No Rating	★★★★★
Action			

Manager Page

Menu

- Drink
- Burger
- Noodle
- Hot Pot

+ Add New Dish + Add Category

Drink

	Coke
Image	
Name	Coke
Price	\$5
Rating	★★★★★
Action	

Add New Dish

Dish Image: + Upload

* Dish Name: Bubble Tea

* Category: Drink

* Price: 10

* Description: Delicious Bubble Tea

* Ingredients: Sugar, milk and Tea

Submit

The screenshot shows the Manager Page interface. On the left, there's a sidebar with categories: Drink, Burger, Noodle, and Hot Pot, each with a trash icon. The main area is titled "Manager Page" and has tabs for "Drink", "Burger", and "Page". At the top right, a red box highlights a green success message: "Dish added successfully!". Below it are buttons for "Add New Dish", "Add New Category", "Add Table", and "Statistics". The "Drink" section displays four items: Bubble Tea (\$10), Coke (\$5), Sprite (\$5), and Canada Dry (\$3). The "Bubble Tea" card is highlighted with a red border. The "Burger" tab is selected at the bottom.

When adding a new dish to the menu from the manager's end, the process is simple and user-friendly. By accessing the manager's homepage, they can easily locate and click on the "Add Dish" button. Upon clicking the button, a form will appear, prompting the manager to input the necessary information for the new dish. The required information includes the dish name, description of the dish, price, category, ingredients and a picture.

2.6 Manager - Modify dish

(Addresses objective 7: Manager System)

This screenshot shows the Manager Page after a modification. The "Drink" section now lists five items: Bubble Tea (\$10), Coke (\$5), Sprite (\$5), Canada Dry (\$3), and a newly added item, "Bacon Cheeseburger" (\$12). The "Bacon Cheeseburger" card is highlighted with a red border. The "Burger" tab is selected at the bottom. The rest of the interface is identical to the previous screenshot, including the sidebar and top navigation.

Manager Page

The screenshot shows a modal window titled "Modify Dish" over a menu list. The modal contains fields for "Dish Image" (with a placeholder image of a bubble tea), "Dish Name" (Bubble Tea), "Price" (\$25), "Description" (Delicious Bubble Tea), "Ingredients" (Sugar, milk and Tea), and "Category" (Drink). The "Modify" button is highlighted with a red box.

Category	Dish Name	Price	Description	Ingredients	Category
Drink	Bubble Tea	\$10	No Rating	Sugar, milk and Tea	Drink
Burger	Canada Dry	\$3	★★★★★	Canada Dry GINGER ALE	Burger

Manager Page

The screenshot shows a list of drink items: Bubble Tea (\$25, No Rating), Coke (\$5, ★★★★★), Sprite (\$5, No Rating), and Canada Dry (\$3, ★★★★★). Each item has a card with its name, price, rating, and a "Modify" button.

Category	Dish Name	Price	Description	Ingredients	Category
Drink	Bubble Tea	\$25	No Rating	Sugar, milk and Tea	Drink
Drink	Coke	\$5	★★★★★	Coca-Cola	Drink
Drink	Sprite	\$5	No Rating	Sprite	Drink
Drink	Canada Dry	\$3	★★★★★	Canada Dry GINGER ALE	Drink

By clicking the "Modify" button on each dish card, a form will appear with the previous patch information already filled in. This convenient feature saves the manager time as they can directly make the necessary changes to the dish details within the form. Once confirmed, the menu will automatically refresh, displaying the updated information for that dish.

2.7 Manager - Delete Dish

(Addresses objective 7: Manager System)

Manager Page

The screenshot shows a modal window titled "Modify Dish" over a menu list. The modal contains fields for "Dish Image" (placeholder image), "Dish Name" (Bubble Tea), "Price" (\$25), "Description" (Delicious Bubble Tea), "Ingredients" (Sugar, milk and Tea), and "Category" (Drink). The "Delete" button is highlighted with a red box.

Category	Dish Name	Price	Description	Ingredients	Category
Drink	Bubble Tea	\$10	No Rating	Sugar, milk and Tea	Drink
Burger	Canada Dry	\$3	★★★★★	Canada Dry GINGER ALE	Burger

Manager Page

Menu

- Drink
- Noodle
- Hot Pot
- Burger

Drink

Item	Description	Rating	Action
Coke	\$5	★★★★★	
Sprite	\$5	No Rating	
Canada Dry	\$3	★★★★★	

In the same dish card, the manager can click the “Delete” button to delete the dish. Once confirmed, the menu will automatically refresh, removing that specific dish.

2.8 Customer - Select table

(Addresses objective 2: Customer Ordering)



WELCOME

+ Select:

2
3
4
5
6
7
8
9



WELCOME

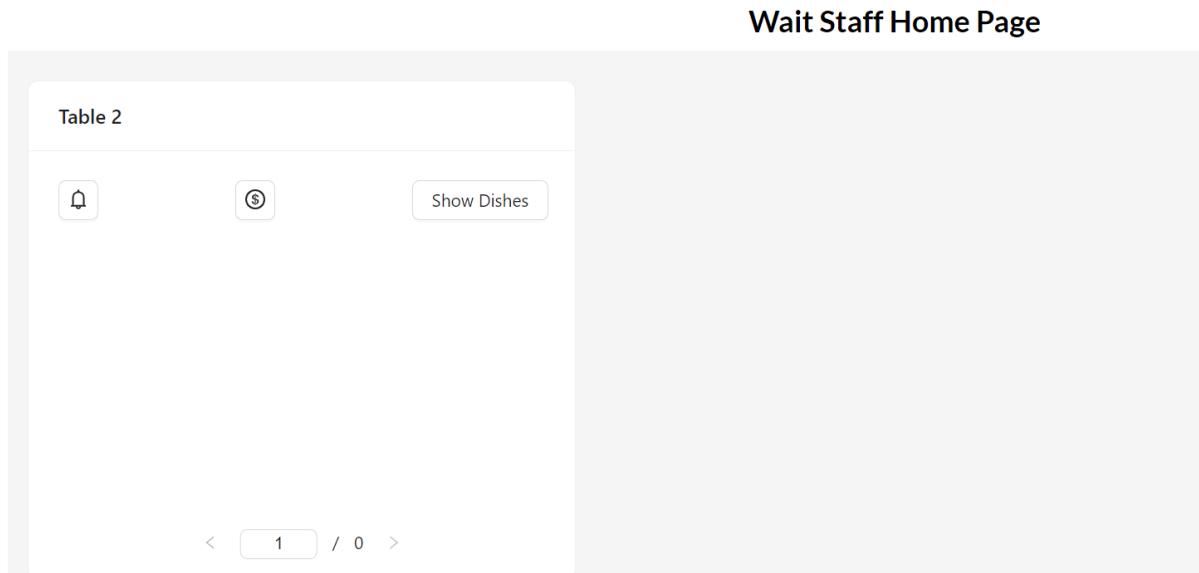
+ Select:



From the customer's end, we can select the table easily by the drop down menu.

2.9 Waitstaff - Create the template for table

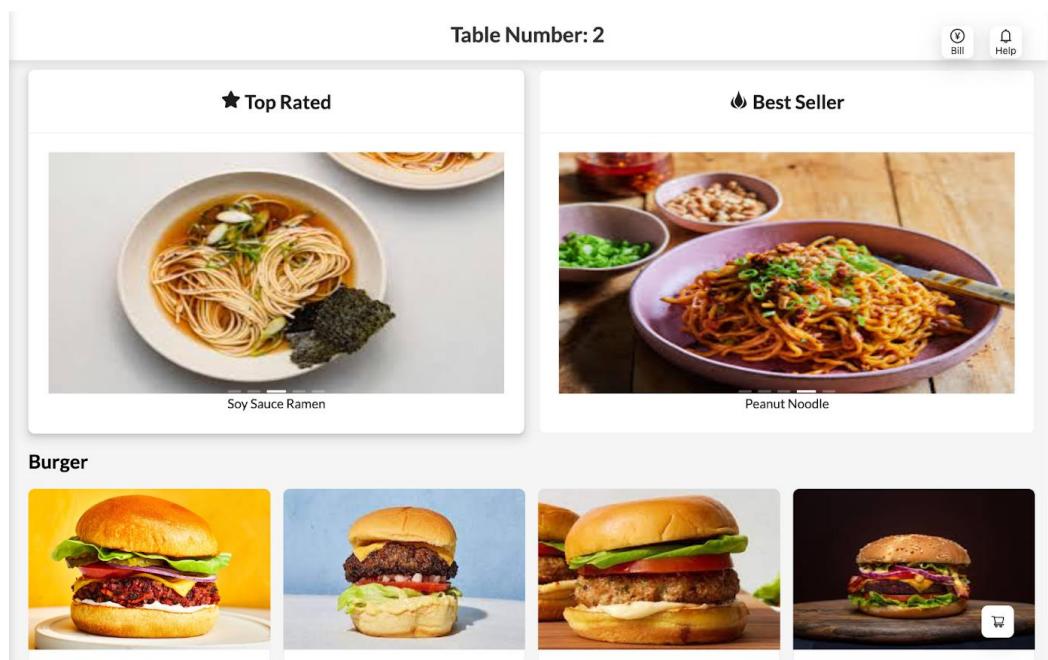
(Addresses objective 3: Waitstaff System)



Once the table has been selected, the waitstaff interface will load a template to store information about that table during the dining.

2.10 Customer - View menu

(Addresses objective 1: Customer Menu Browsing)



Customers have two convenient options for viewing the menu. They can simply scroll up and down to explore all the items, or they can click on the categories listed on the left-hand side to directly jump to a specific category of interest.

2.11 Customer - View details of specific dish and add dish to cart

(Addresses objective 1 & 2: Customer Menu Browsing & Customer Ordering)

Menu

- Burger
- Noodle
- Hot Pot
- Drink

Hot Pot

Image	Name	Price	Rating
	Japanese Hotpot	\$25	★★★★★
	Korean Hotpot	\$30	No Rating
	RamYun Hotpot	\$35	No Rating
	Chongqing Hotpot	\$30	★★★★★

Drink

Image	Name	Price	Action
	Coke	\$5	
	Sprite	\$5	
	Canada Dry	\$3	

Menu

- Burger
- Noodle
- Hot Pot
- Drink

Hot Pot

Image	Name	Price	Rating
	Habachi Noodle	\$12	★★★★★
	Japanese Hotpot	\$25	★★★★★
	Korean Hotpot	\$30	No Rating
	Chongqing Hotpot	\$30	★★★★★

Korean Hotpot

Description: Korean Hotpot, delicious

Ingredients: Korean Hotpot

Price: 30

Add Dish

Customers can view the detailed information of the chosen dish by clicking the desired dish. And after that, they can choose to add that dish to cart or keep viewing by clicking the mask area.

2.12 Customer - View current order, Delete Dish and Place order (Addresses objective 2: Customer Ordering)

Menu

- Burger
- Noodle
- Hot Pot
- Drink

Habachi Noodle
\$12

★★★★★

Hot Pot

Japanese Hotpot
\$25

★★★★★

Korean Hotpot
\$30

No Rating

RamYun Hotpot
\$35

No Rating

Chongqing Hotpot
\$30

★★★★★

Drink

A shopping cart icon is located in the top right corner.

Menu

- Burger
- Noodle
- Hot Pot
- Drink

Habachi Noodle
\$12

★★★★★

Hot Pot

Korean Hotpot * 2
\$60

Total Cost
\$60

Place Order

Japanese Hotpot
\$25

★★★★★

RamYun Hotpot
\$35

No Rating

Chongqing Hotpot
\$30

★★★★★

Drink

A shopping cart icon is located in the top right corner.

Menu

- Burger
- Noodle
- Hot Pot
- Drink

Habachi Noodle
\$12

★★★★★

Hot Pot

Korean Hotpot * 2
\$60

Total Cost
\$60

Place Order

Japanese Hotpot
\$25

★★★★★

RamYun Hotpot
\$35

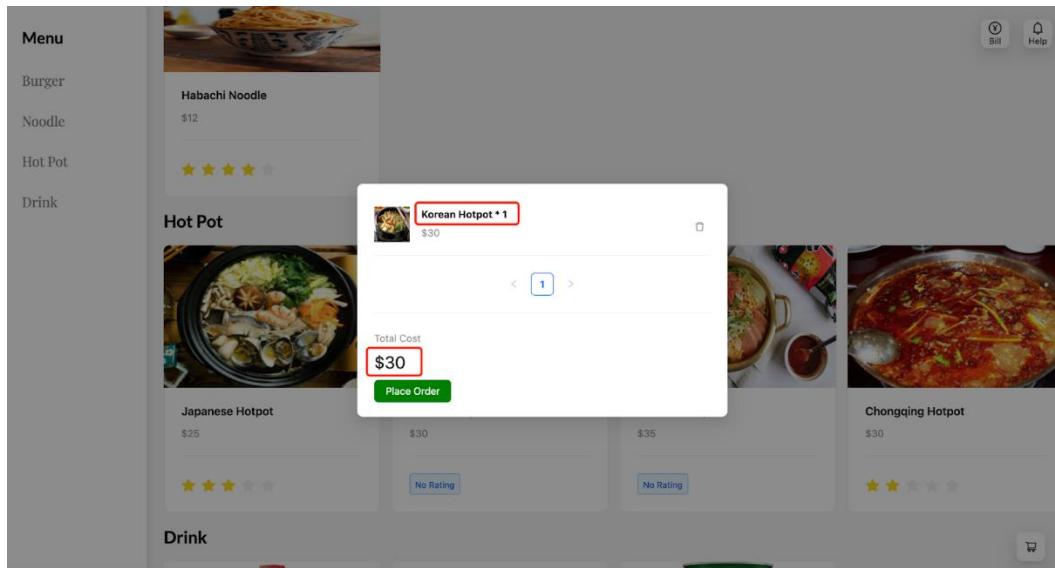
No Rating

Chongqing Hotpot
\$30

★★★★★

Drink

A delete icon (red square with a white minus sign) is highlighted over the Korean Hotpot order in the shopping cart overlay.

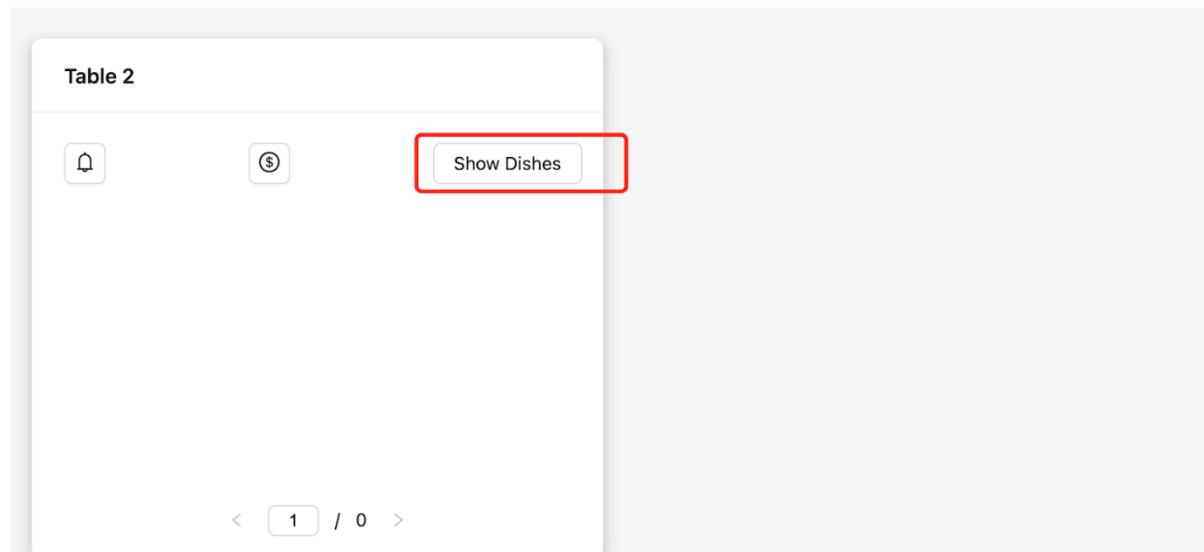


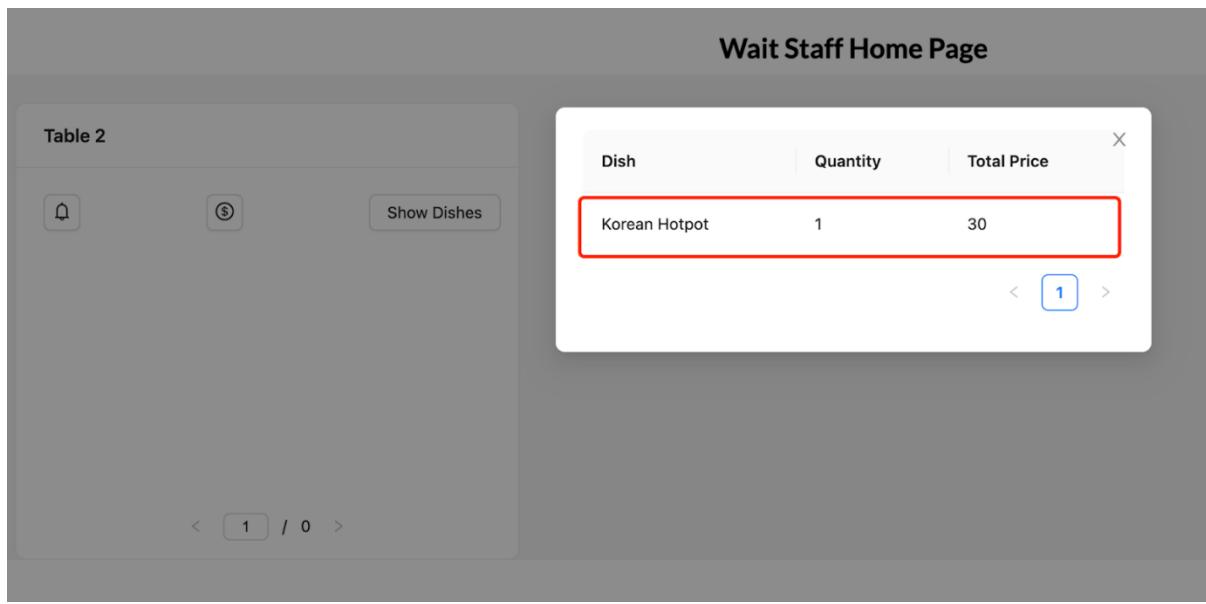
Customers can access their current order by clicking the cart button located in the bottom right corner. From there, they have the option to either remove items from the order or proceed with the placement. After deletion, the price and order information will automatically refresh. If they decide to place the order, it will be concluded and forwarded to the kitchen and waitstaff. For the customer's convenience, they can continue adding new items to a fresh order.

2.13 Waitstaff - View order information

(Addresses objective 5: Waitstaff System)

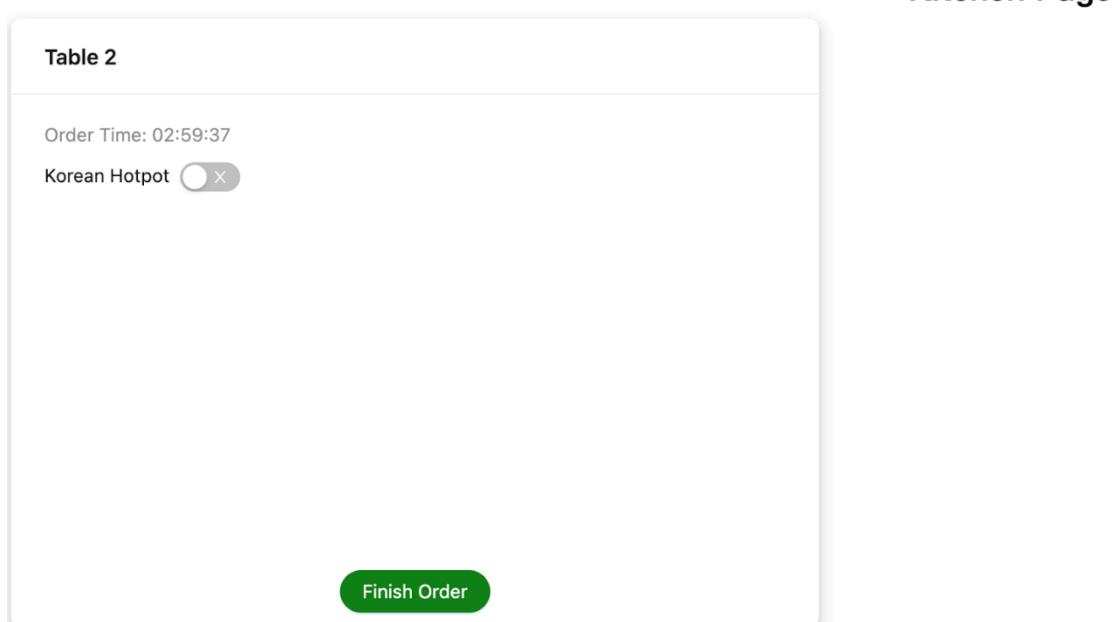
Wait Staff Home Page



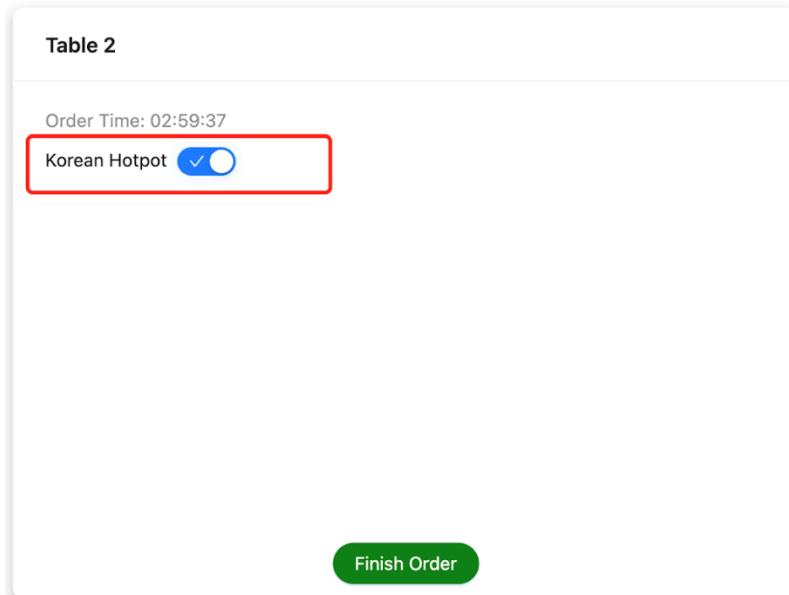


Waitstaff can access and view the table's order information.

2.14 Kitchen - Create order template, Cook, Flag once finished (Addresses objective 4 &6: Kitchen Staff System & Notification System)



Kitchen Page



Kitchen Page



Waiting for orders...

Once the customer places an order, the kitchen receives the order information. As the kitchen finishes preparing a dish, they can mark it as "prepared," and this status update will be sent to the waitstaff's interface. Additionally, if there is a mistake or a wrong click, they can quickly revert the status back to false. Once all the dishes in an order have been prepared and are ready for service, the waitstaff can choose to finish the order. Finishing the order helps to save screen space.

2.15 Waitstaff - Receive dish status, serve the dish (Addresses objective 5 & 6: Waitstaff System & Notification System)

Wait Staff Home Page

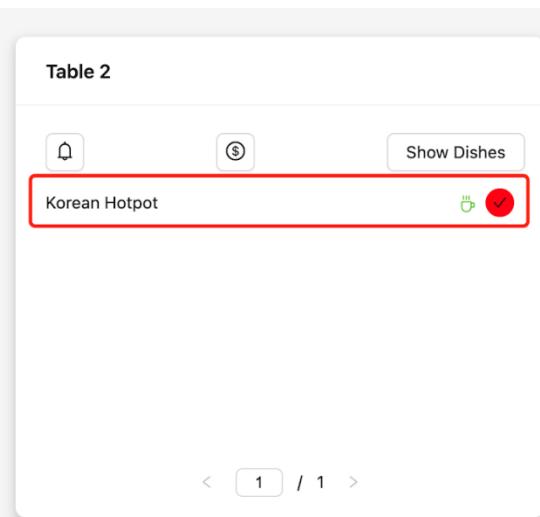


Table 2

Korean Hotpot ✓

< 1 / 1 >

Wait Page ✓ Dish served

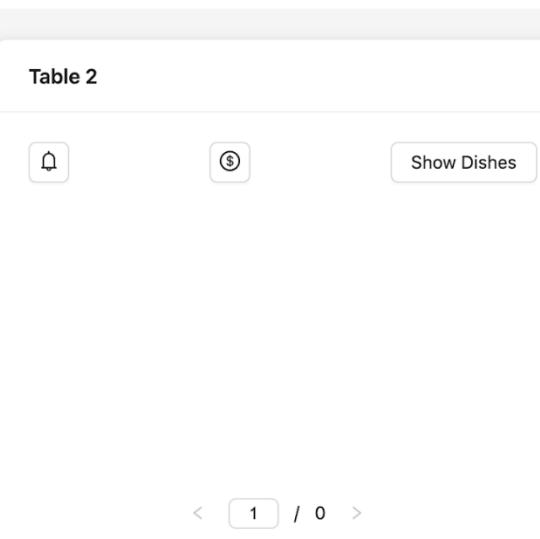


Table 2

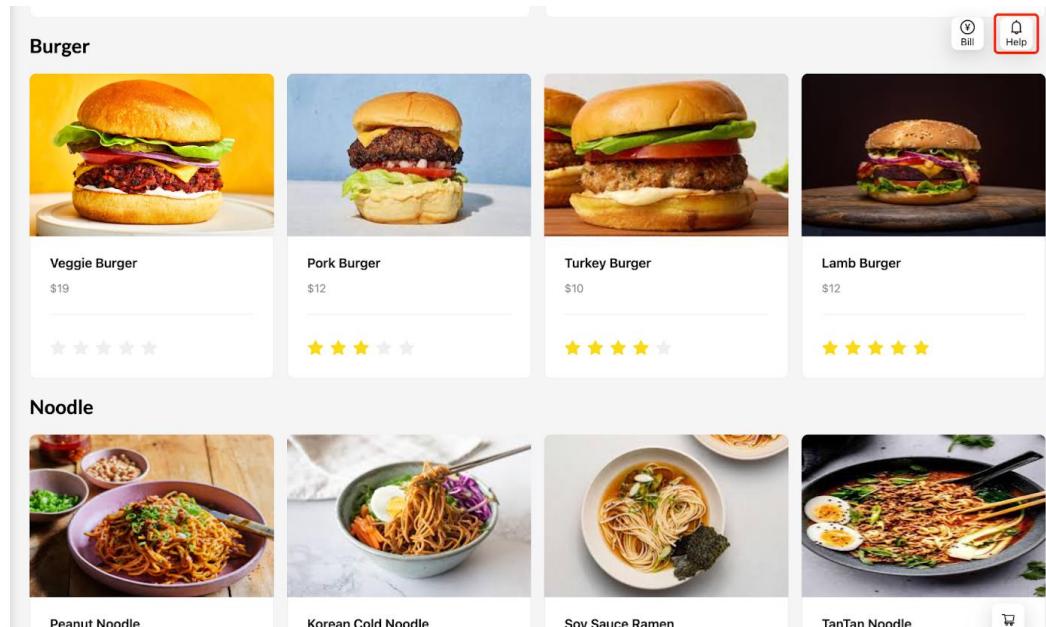
Dish served

< 1 / 0 >

On the waitstaff's template, only the cook dishes are displayed. If the waitstaffs wish to see the full order, they can simply press the "Show Dishes" button, which will reveal all the dishes for this table. When a waitstaff becomes available, they can claim the task of delivering a cooked dish by pressing the red button located on the right side of each cook dish.

2.16 Customer - Ask for help

(Addresses objective 3 & 6: Customer Request & Notification System)



Customers can call for help by clicking the ring button on the top right position.

2.17 Waitstaff – Respond to the help request

(Addresses objective 5 & 6: Waitstaff System & Notification System)

Wait Staff Home Page

The wait staff can promptly respond to a customer's help request by clicking the yellow ring button, which indicates that the customer on a specific table requires assistance. Once a wait staff claims the help request by clicking the button, the other waitstaff interfaces will no longer be notified about this particular help request.

2.18 Customer - Request bill and finish meal, Provide rating

(Addresses objective 8: Rating and Recommendation System)

Menu

Burger

Noodle

Hot Pot

Drink

Burger

- Veggie Burger \$19 ★★★★★
- Pork Burger \$12 ★★★★★
- Turkey Burger \$10 ★★★★★
- Lamb Burger \$12 ★★★★★

Noodle

- Peanut Noodle
- Korean Cold Noodle
- Soy Sauce Ramen
- TanTan Noodle Cart

Menu

Burger

Noodle

Hot Pot

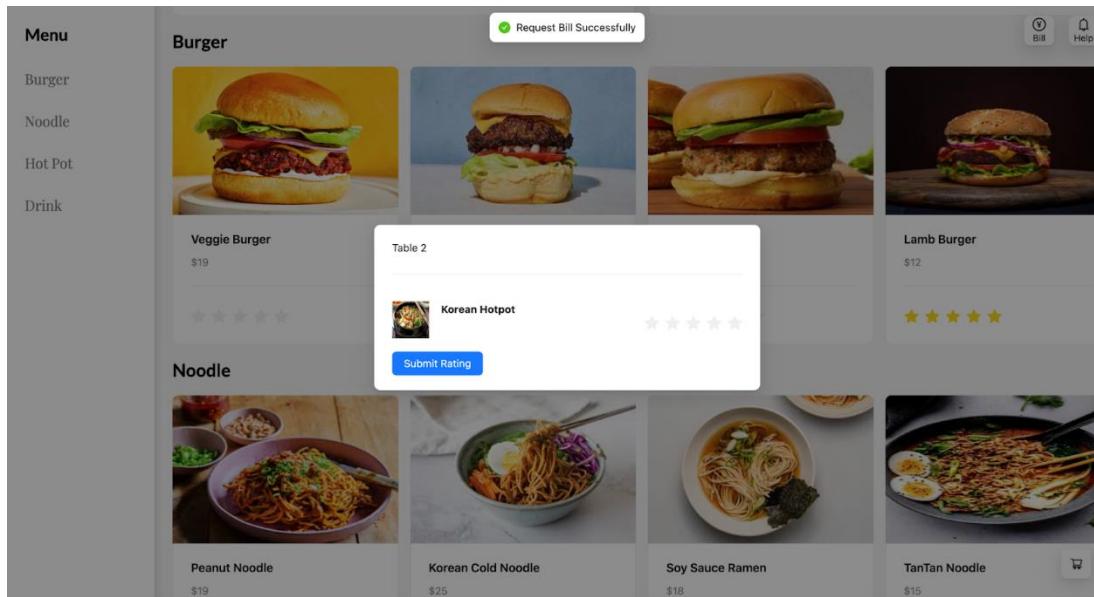
Drink

Burger

- Veggie Burger \$19 ★★★★★
- Korean Hotpot *1** Price: 30
- Total Cost (Not include un placed order)(AUD) 30
- < >
- Finish Meal**

Noodle

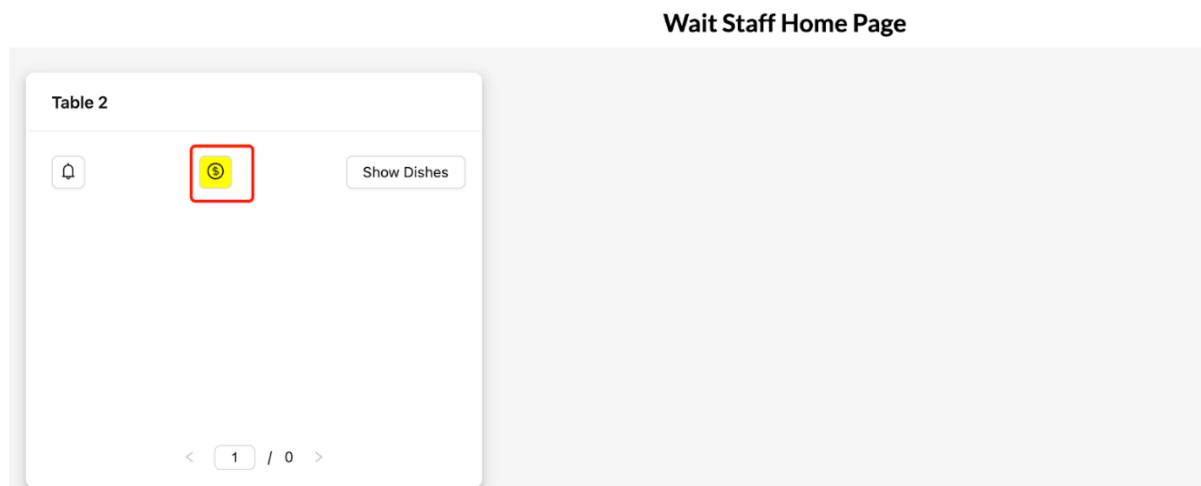
- Peanut Noodle
- Korean Cold Noodle
- Soy Sauce Ramen
- TanTan Noodle Cart



During the meal, customers have the option to view the full order by clicking the "Bill" button. Once they have reviewed the order and confirmed that all dishes have been served, they can choose to finish the meal and request the bill. While waiting for the bill, customers also have the opportunity to rate the dish they ordered. Once the waitstaff confirms the payment, the customer's interface will automatically redirect to the selecting table page.

2.19 Waitstaff – Confirm payment

(Addresses objective 5: Waitstaff System)



When the dollar button for a table turns yellow, it indicates a bill request from the customers. After sending the bill, the waitstaff can confirm the payment by clicking the dollar button to mark the order as finished for that table.

2.20 Manager - View statistics

(Addresses objective 9: Analytics)

Manager Page

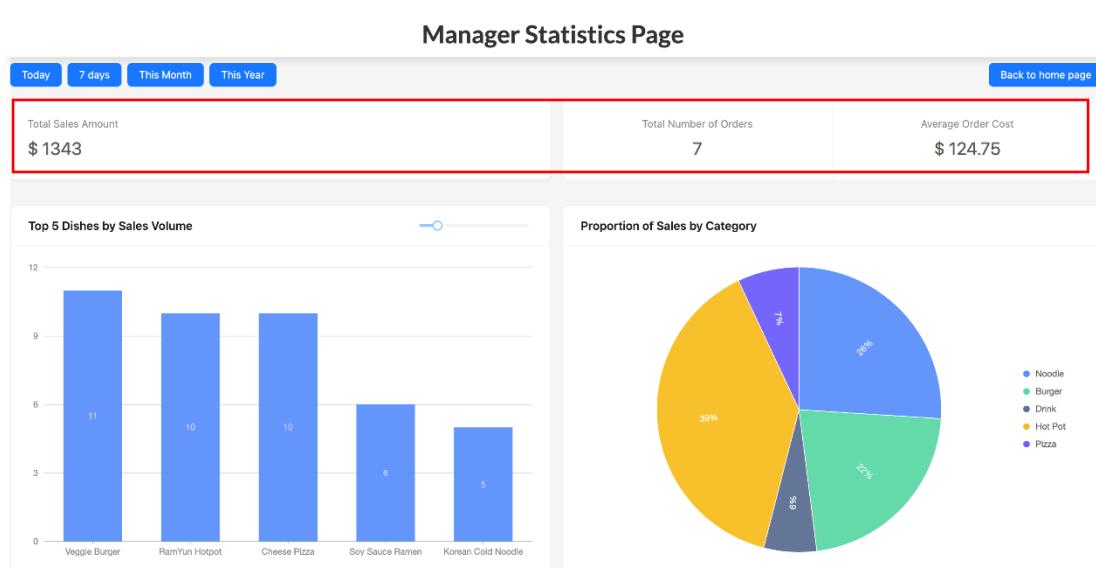
Menu

- Burger
- Soup
- Wrap
- Snacks
- Drinks

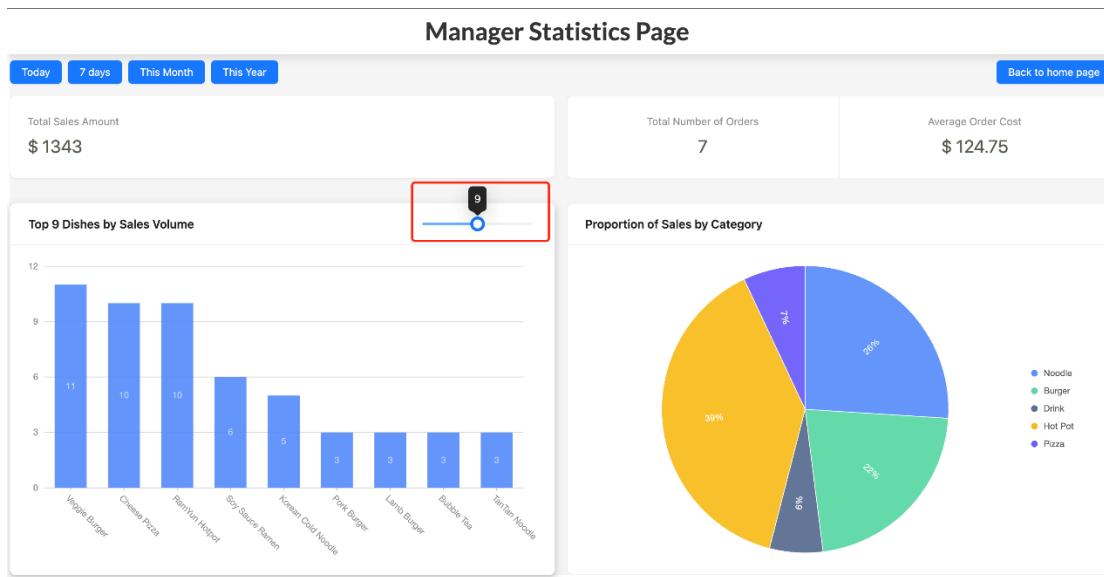
Burger

Image	Name	Price	Rating	Action
	Chicken Burger	\$12	★★★★★	
	Fish Burger	\$10.5	★★★★★	
	Cheeseburger	\$12	★★★★★	

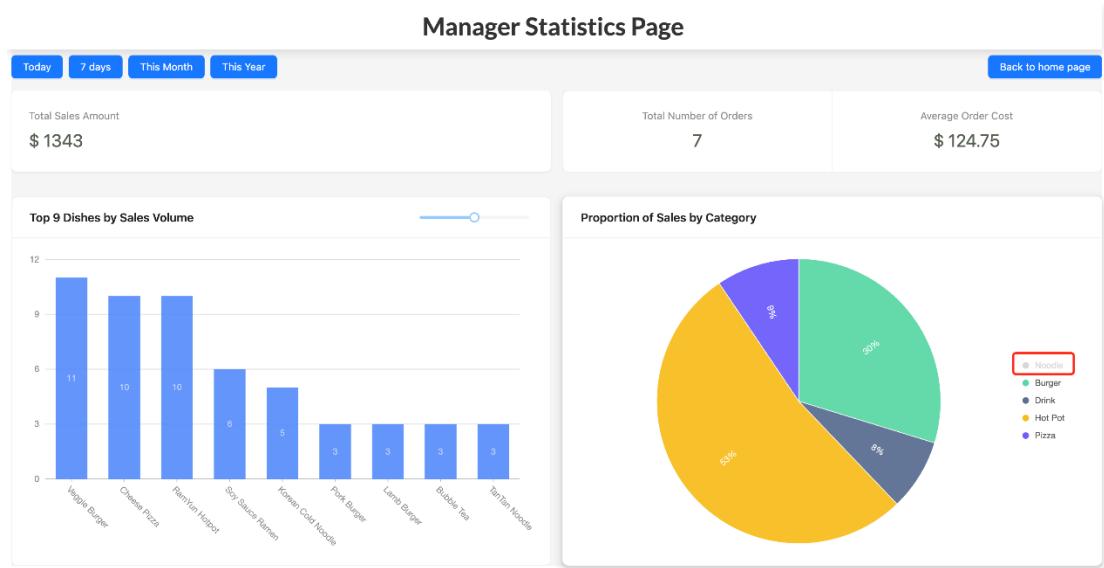
Statistics



The manager can view the statistical information by clicking the statistic button. In the statistical section, the manager can view the total sales amount, total number of orders, average order cost on the first row.

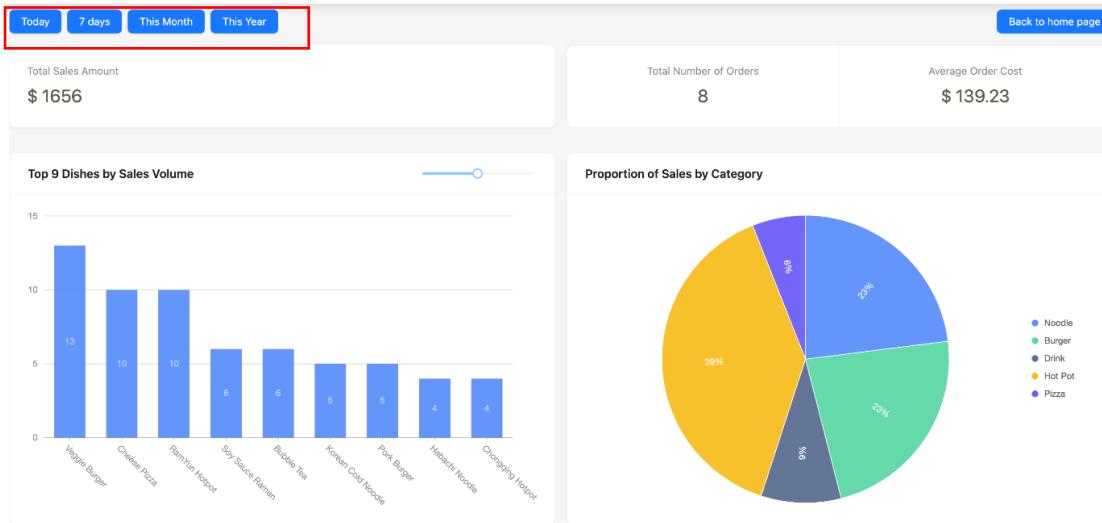


On the second row of the interface, top dishes are showcased based on their sales volume, and they appear on the left-hand side. The manager has the flexibility to adjust the number of top dishes displayed on the screen using a slider. By sliding the slider, the manager can change the "Top x" value. Dishes are displayed in descending order.



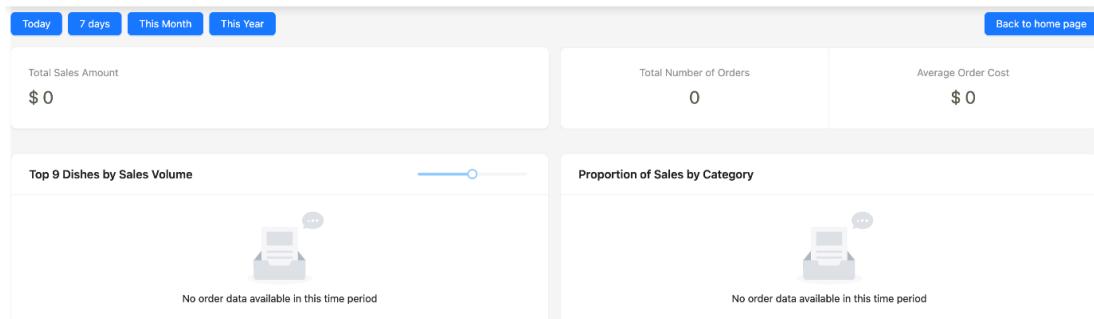
On the right-hand side of the interface, the manager can access a pie chart displaying the proportion of sales for each category. This chart provides a visual representation of how much revenue each category contributes to the overall sales. Moreover, the manager has the option to interact with the pie chart by deselecting specific categories.

Manager Statistics Page



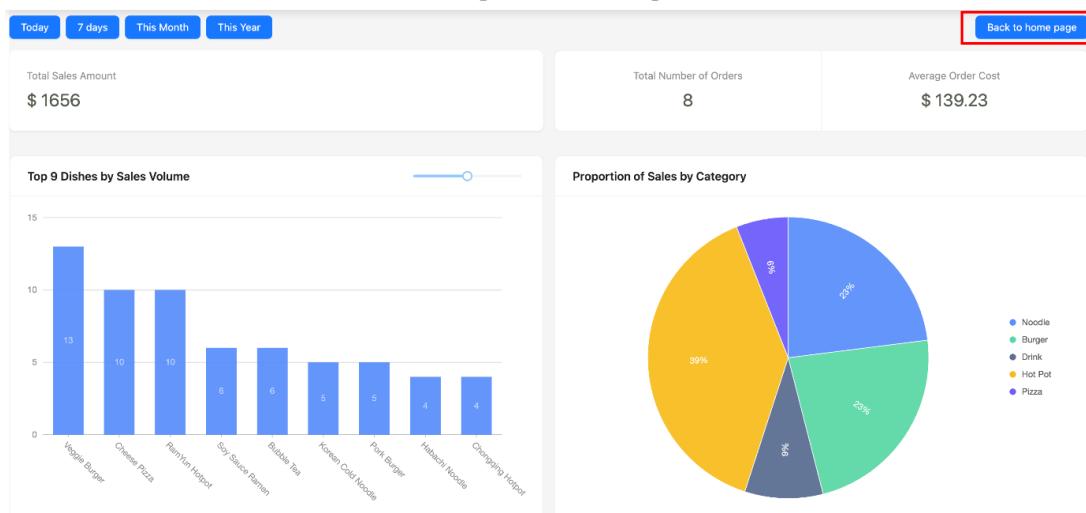
The system offers multiple options for displaying statistics, giving managers the flexibility to analyze data over different time frames.

Manager Statistics Page



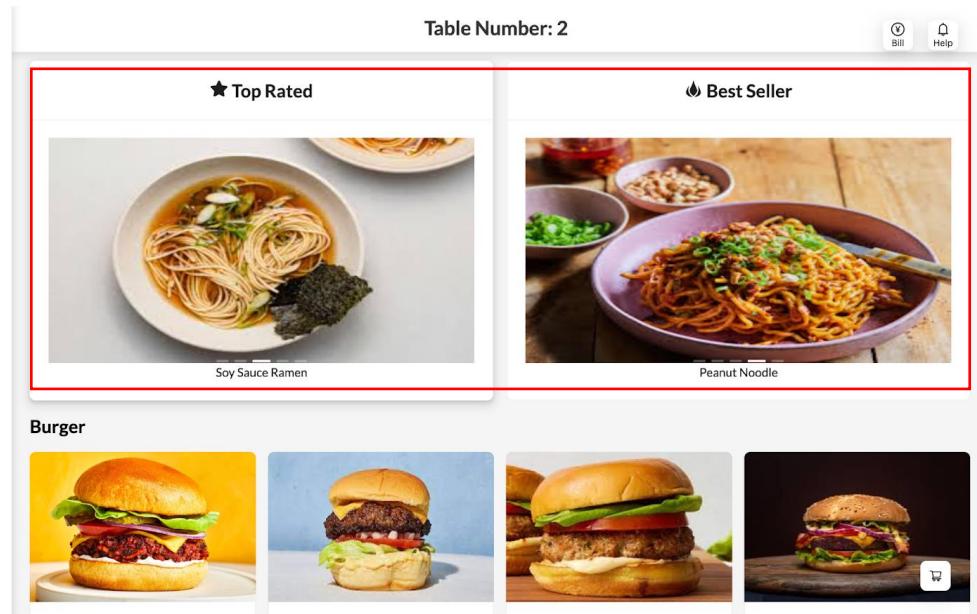
The screenshot above illustrates a scenario where there is no data available for display.

Manager Statistics Page



The manager can easily navigate back to the home page by clicking the top-right button.

2.21 Customer – View top rated and best sellers (Addresses objective 8: Rating and Recommendation System)



Customers can view the top-rated and best-selling dishes on the carousel. The carousel feature showcases the top 5 dishes, providing a visually engaging way for customers to discover the most popular and highly recommended items available at the restaurant. This helps customers make informed choices and enhances their dining experience.

3 Third-Party functionalities

3.1 Front-end

In our frontend development process, we employed Ant Design (Ant Group and Ant Design Community, 2019), an open-source React UI library offering a suite of high-quality components that follow the Ant Design System's principles. As a comprehensive recipe recommendation system, we took advantage of Ant Design's capabilities to make our pages more visually appealing, emphasizing vibrant colors and notable animation effects.

Ant Design provides a range of practical components, supporting extensive customizability including theme modification, internal styling, and color alteration. Furthermore, we utilized Ant Design's data visualization library, Ant Design Charts, for our statistical page, creating intuitive and interactive graphs to better represent data and provide valuable insights. This use of Ant Design enhances the user experience with its professional-quality design and functionality.

The license file on the GitHub repository (ant-design, 2015) confirms that Ant Design is licensed under the terms of the MIT License. The MIT License permits commercial use, modification, and distribution of the software while requiring users to include copyright and permission notices in their copies. This licensing model provides us with flexibility and customizability for our project. Additionally, there are no licensing fees or upfront costs associated with using the library, making it cost-effective for deployment in the market.

3.2 Back-end

3.2.1 Mybatis-plus

For the interaction between the database and our application, we use Mybatis-plus (Baomidou, 2023) as our tool. Mybatis-plus has 3 characters:

1. Non-intrusive: only enhancements do not change, the introduction of it will not affect the existing project.
2. Low loss: Basic CURD is automatically injected upon startup, performance is basically loss-free, and direct object-oriented operation.
3. Powerful CRUD operations: built-in general Mapper, general Service, only through a small amount of configuration can achieve most of the single table CRUD operations, more powerful condition constructor, to meet all kinds of use needs.

With Mybatis-plus, we can simply just write Java code in our service implementation class to interact with the database. This approach does not need us to create .xml files and write SQL sentences in a traditional way. Using Mybatis-plus greatly improves our development efficiency.

Similar to Ant Design, MyBatis-Plus is also licensed under the MIT License (Baomidou, 2020), providing us with similar benefits.

3.2.2 Postman

And for the testing purpose, we use Postman 1.1.1 (Postman, 2021) as our testing tool. Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so we can create better APIs—faster. Since we use the Agile method to develop this application, we need to do the unit test each week. Post provides massive convenience for our unit testing. We can easily test our APIs to see whether it produces the correct outputs.

The free tier of Postman is completely open to use for commercial purposes (Postmanaut, 2019). The licensing terms of Postman enable the project team to leverage a powerful API testing tool without additional financial burden. It supports the Agile development process and aids in delivering a reliable and well-tested APIs.

4 Implementation challenges

4.1 Front-end

4.1.1 Co-Debugging Frontend and Backend

A notable challenge we faced during frontend development was co-debugging with the backend. Our system required numerous interfaces to be designed and adjusted, which made it a demanding task. Further, backend development was progressing concurrently, making use of mock data for temporary interface content. However, co-debugging led to significant changes in terms of port, form, and templates.

After a tough initial co-debugging session, we held a retrospective meeting to clarify potential backend data requirements. This helped us optimize logic for preset configurations and streamline the co-debugging process, making it more efficient and reducing unnecessary time expenditure.

4.1.2 Familiarizing with a new UI library: Ant Design

Another challenge we encountered was the utilization of an unfamiliar UI library, Ant Design. As a relatively new tool for us, it required an understanding of its various components, customizations, and application principles. It involved a steep learning curve that demanded additional time and effort for the team to become proficient and comfortable using it for interface design and data visualization effectively. However, with continuous learning and practice, we were able to navigate this challenge, harnessing Ant Design's capabilities to enhance our frontend development.

4.1.3 Enhancing Program Efficiency with React Hooks

Another hurdle we faced was improving program efficiency. To address this, we utilized React Hooks. Despite their potential to significantly streamline code, enhance readability, and simplify state and lifecycle management in functional components, initially, our team had limited familiarity with using Hooks. This posed a challenge as they required a different mindset and approach compared to class components. However, through diligent learning and application, we effectively integrated React Hooks into our project, successfully improving the overall efficiency of our program.

4.2 Back-end

4.2.1 Deletion in Database

In our application, the manager is able to remove categories or items on the menu. At first, we designed to set the foreign key in other tables which connects with primary keys in category(id) and item(id) on delete cascade. But this will cause a problem because we still need the information of orders including deleted items or categories to implement our analysis functionality.

At last, we decided to add an attribute “is_on_menu” for the category table and item table. This not only can guarantee the customer cannot view the deleted categories or items on the menu anymore, and also retain the information of orders in our database.

4.2.2 Table Join Query

Unfortunately, Mybatis-plus does not support the query which needs table join operation. So we searched about many methods and decided to use Mybatis-plus-join to solve this problem. This is an extension package of Mybatis-plus, which provides the table join functionality. We just need to import the dependency of Mybatis-plus in our Maven pom.xml, and then we can do the table join operation easily with its function.

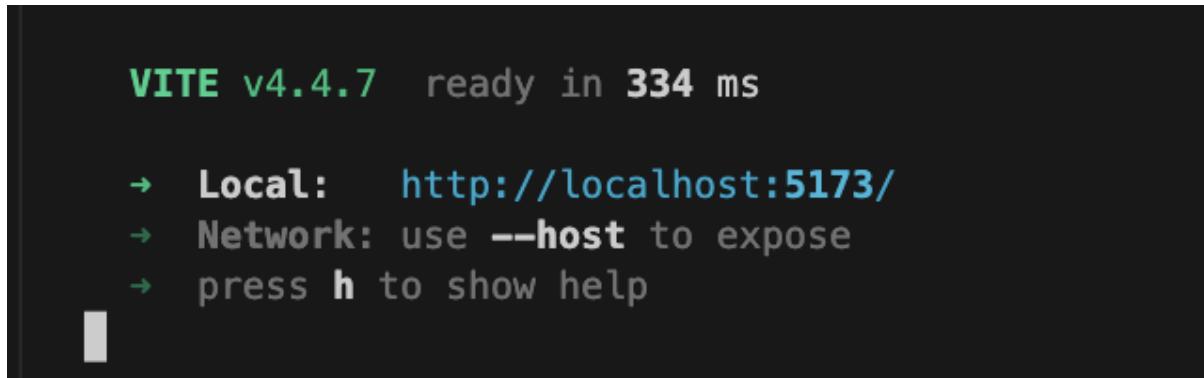
4.2.3 Table Join Query

In the wait-staff and kitchen-staff end of our application, as well as the analysis page, the front-end requires a long body of integrated json data within just one API. To solve this challenge, we created several VOs and many DTOs. VO is the final format of returned data, which can be customized. And DTO is an entity class for data transfer. In one API, we do SQL queries several times, and save the returned data in the corresponding DTOs. At last, we get the data from the DTOs and save them into VO.

5 User Manual

5.1 Front-end

1. Make sure you have the following dependencies: npm, node, yarn.
2. Go to our project root folder
3. Use command lines
 - a. `cd frontend`
 - b. `npm install --legacy-peer-deps` or `yarn install`
 - c. `npm run dev` or `yarn run dev`
4. The frontend should be running on localhost:5173



```
VITE v4.4.7 ready in 334 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h to show help
```

5.2 Back-end

1. On Linux
 - a. Download files
 - i. Download the 9900.sql file from the path `/Installation/9900.sql`
 - ii. Download the waitSys.jar file and open a terminal in this path `/Installation/waitSys.jar`
 - b. Install JDK20 could follow the steps 1-3 from
<https://ubuntuhandbook.org/index.php/2022/03/install-jdk-18-ubuntu/>

```
qifan@ubuntu:~/Downloads$ sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk-20/bin/java 1
update-alternatives: using /usr/lib/jvm/jdk-20/bin/java to provide /usr/bin/java (java) in auto mode
qifan@ubuntu:~/Downloads$ sudo update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk-20/bin/javac 1
update-alternatives: using /usr/lib/jvm/jdk-20/bin/javac to provide /usr/bin/javac (javac) in auto mode
qifan@ubuntu:~/Downloads$ sudo update-alternatives --install /usr/bin/jar jar /usr/lib/jvm/jdk-20/bin/jar 1
update-alternatives: using /usr/lib/jvm/jdk-20/bin/jar to provide /usr/bin/jar (jar) in auto mode
qifan@ubuntu:~/Downloads$ sudo update-alternatives --config java
There is only one alternative in link group java (providing /usr/bin/java): /usr/lib/jvm/jdk-20/bin/java
Nothing to configure.
qifan@ubuntu:~/Downloads$ sudo update-alternatives --config javac
There is only one alternative in link group javac (providing /usr/bin/javac): /usr/lib/jvm/jdk-20/bin/javac
Nothing to configure.
qifan@ubuntu:~/Downloads$ sudo update-alternatives --config jar
There is only one alternative in link group jar (providing /usr/bin/jar): /usr/lib/jvm/jdk-20/bin/jar
Nothing to configure.
qifan@ubuntu:~/Downloads$ sudo apt install sudo
Command 'udo' not found, but can be installed with:
sudo apt install sudo

qifan@ubuntu:~/Downloads$ sudo update-alternatives --config jar
There is only one alternative in link group jar (providing /usr/bin/jar): /usr/lib/jvm/jdk-20/bin/jar
Nothing to configure.
qifan@ubuntu:~/Downloads$ java -version
java version "20.0.2" 2023-07-18
Java(TM) SE Runtime Environment (build 20.0.2+9-78)
Java HotSpot(TM) 64-Bit Server VM (build 20.0.2+9-78, mixed mode, sharing)
```

- c. Install MySQL version 8
 - i. `sudo apt install mysql-server`
 - ii. `sudo mysql`
 - iii. `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'rootroot';`
- d. Initialize the database

- i. mysql -u root -p
 - ii. create database waitsys;
 - iii. use waitsys;
 - iv. source {path of 9900.sql file}

```
qifan@ubuntu:~/Downloads$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.33-0ubuntu0.20.04.4 (Ubuntu)

copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database waitsys;
Query OK, 1 row affected (0.01 sec)

mysql> use waitsys;
Database changed
mysql> source /home/qifan/Desktop/Folder/9900.sql
Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec) I

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

e. Run the back-end

- i. `java -jar waitSys.jar`

The screen should look like this.

2. On Windows and MacOS

- a. Follow the steps on the BackEnd_ReadMe.doxc from
/Installation/BackEnd_ReadMe.doxc

3. Backend Docker Deployment

- a. Make sure that your system has installed docker and docker-compose.
 - b. Go to our project code root folder

- se command lines

 - i. cd backend-docker
 - ii. docker

```

> docker-compose up -d
[+] Building 41.7s (17/17) FINISHED
=> [mysql internal] Load .dockerignore
=> => transferring context: 2B
=> [mysql internal] Load build definition from Dockerfile.mysql
=> => transferring dockerfile: 335B
=> [mysql internal] Load metadata for docker.io/library/mysql:8.1.0
=> [mysql auth] Library/mysql:pull token for registry-1.docker.io
=> [mysql 1/3] FROM docker.io/library/mysql:8.1.0@sha256:6a5dbd2819e36048669639811461f27fee48dale22039e5d31f4273a20d542f6
=> => copying docker.io/library/mysql:8.1.0@sha256:6a5dbd2819e36048669639811461f27fee48dale22039e5d31f4273a20d542f6
[+] Running 4/4
  ✓ Network test_app-network  Created
  ✓ Volume "test_db_data"    Created
  ✓ Container test-mysql-1   Started
  ✓ Container test-app-1    Started

```

- d. It will automatically build the backend image for you, then run the backend container in localhost:8080 with a detached mode .
- e. use `docker ps` command to check whether containers are running or not.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
4a5b95d576a1	test-app	"java -jar waitSys.jar"	15 minutes ago	Up 15 minutes	0.0.0.0:8080->8080/tcp	test-app-1
5c40d1c57c3c	test-mysql	"docker-entrypoint.s..."	15 minutes ago	Up 15 minutes	3306/tcp, 33060/tcp	test-mysql-1

6 Reference

Ant Group and Ant Design Community. (2019). *Ant Design - A UI Design Language and React UI library*. Ant.design. <https://ant.design/>

ant-design. (2015). ant-design/LICENSE at master · ant-design/ant-design. GitHub. <https://github.com/ant-design/ant-design/blob/master/LICENSE>

Baomidou. (2020). awesome-mybatis-plus/LICENSE at master · baomidou/awesome-mybatis-plus. GitHub. <https://github.com/baomidou/awesome-mybatis-plus/blob/master/LICENSE>

Baomidou. (2023). *MyBatis-Plus*. Mybatis.plus. <https://mybatis.plus/en/>

Postman. (2021). *Postman / The Collaboration Platform for API Development*. Postman. <https://www.postman.com/>

Postmanaut. (2019, July 12). Pricing - Who needs to have Pro and Enterprise. Postman Community. <https://community.postman.com/t/pricing-who-needs-to-have-pro-and-enterprise/6508>

7 Appendix

7.1 User Stories in Proposal

Objective 1: Customer Menu Browsing

Customers can navigate the menu interface, where items are organized into different categories. A clear and user-friendly digital menu enhances customer's ordering experience.

User Stories	Acceptance Criteria
As a customer, I want to have a menu where all items are categorized based on their categories so that I can quickly find the items I want to order.	<ul style="list-style-type: none">- Customer should be able to view all items available in the menu.- Each item should be properly classified and grouped.
As a customer, I want to see pictures and details of the items so that I can make informed decisions about what to order.	<ul style="list-style-type: none">- Picture of each item should be clearly displayed.- A description and ingredients of each item should be provided.- Price of each item should be clearly displayed.

Objective 2: Customer Ordering

Customers can order items from the menu, providing them with the flexibility to choose and order their desired items at any time during their dining experience. This bypass the need to wait for a waiter's assistance, reducing the waiting time for customers.

User Stories	Acceptance Criteria
As a customer, I want to select my table number so that the restaurant can identify my order and serve the food to the correct table.	<ul style="list-style-type: none">- The customer can enter their table number before navigating the menu interface.- Only the available table numbers are visible for selection.
As a customer, I want to place my order for the chosen items online so that I can bypass the need to wait for a waiter's assistance.	<ul style="list-style-type: none">- Customer can place an order for items whenever they are ready to do so in the menu browsing interface.
As a customer, I want to add items from the menu during the meal so that I can order what I want at any time.	<ul style="list-style-type: none">- Customer can add items to their order at any time during the meal.

Objective 3: Customer Request

Customers can view the items they ordered, along with their prices, and the total amount on the bill. They can also request their bill once the meal is complete.

User Stories	Acceptance Criteria
As a customer, I want to view the items in my order and the total price of the meal so that I can check these information anytime I want.	<ul style="list-style-type: none">- Customers can view the details of their order, including:<ul style="list-style-type: none">o Item nameso Price for each itemo Quantity of each itemo Total Price
As a customer, I want to request the bill once the meal is complete so that I can then go to the front counter to pay.	<ul style="list-style-type: none">- The checkout button is easily accessible and visible to the customer.
As a customer, I want to request assistance from a waiter so that I can get help.	<ul style="list-style-type: none">- Customer can request service through the notification button on the menu.

Objective 4: Kitchen Staff System

The kitchen staff will have access to a time-sorted order list displayed on the screen. This allows the kitchen staff to have a clear overview of the items that need to be prepared and prioritize orders accordingly. As a result, efficiency is improved and the waiting time for customers is reduced.

User Stories	Acceptance Criteria
As a kitchen staff, I want to see a time-sorted order list so that I can prioritize my work.	<ul style="list-style-type: none">- A time-sorted order list is displayed on the Kitchen Staff interface.- The list is updated when there are new orders.
As a kitchen staff, I want to see all the items in each order so that I can decide what to cook.	<ul style="list-style-type: none">- Items in each order are displayed clearly. This includes:<ul style="list-style-type: none">o Item Nameo Quantity
As a kitchen staff, I want to mark an order item as prepared once it is ready for serving so I can remove it from the list.	<ul style="list-style-type: none">- There is a button next to each item.- Kitchen staff can press the button and mark the item as completed.

	<ul style="list-style-type: none"> - Once all items in the order is completed, the order should be removed from the Kitchen Staff Interface.
--	---

Objective 5: Wait Staff System

The wait staff system aims to let wait staff serve customers in time. Wait staff should be able to see the information of customers' orders and deliver it to the customers. They can mark the notification once it is completed. This system makes convenience for wait staff to serve customers.

User Stories	Acceptance Criteria
As a wait staff, I want to see the notification icon of each table so that I know which tables need assistance.	<ul style="list-style-type: none"> - Wait Staff can view a notification icon next to each table on the Wait Staff interface.
As a wait staff, I want to mark customer request notifications as completed so that other wait staff members are aware and can avoid duplicating the same task.	<ul style="list-style-type: none"> - Wait Staff can mark customer request notifications as complete. - Wait staff interface is updated after a request is marked as complete.
As a wait staff, I want to see the information about the prepared items and the table number so that I can know which item should be delivered and which table should be delivered to.	<ul style="list-style-type: none"> - Wait staff can view the detail of item-serve notifications. This includes: <ul style="list-style-type: none"> o Table number o Item name

Objective 6: Notification System

The notification system aims to improve communication among customers, wait staff and kitchen staff. Customers can use this system to request assistance and the wait staff will see the notifications on the wait staff system. Kitchen staff can use this system to send the notification to wait staff to remind them the order is ready. This system improves the efficiency of communication.

User Stories	Acceptance Criteria
As a customer, I want to notify a Wait Staff when I need assistance so that my needs can be addressed promptly.	<ul style="list-style-type: none"> - When the customer press the request assistance notification button, this information should be transmitted to the Wait Staff Interface.
As a kitchen staff, I notify Wait Staff once items are ready for serving so that they can deliver the prepared items to the respective tables in a timely manner.	<ul style="list-style-type: none"> - When the Kitchen staff marks a prepared item as completed, this information should be transmitted to the Wait Staff Interface.

Objective 7: Manager System

The manager system aims to let manager manage and customize the menu. The manager can use this system to add or remove categories and items. And the manager also can put categories and items in self-defined order. This system enables managers to manage the menu whenever he wants.

User Stories	Acceptance Criteria
As a manager, I want to add new categories and items to the menu so that customers can have more choices.	<ul style="list-style-type: none"> - Manager can add a new category to the menu with a self-defined name - Manager can add a new items to a category with a self-defined name, description, picture, ingredient and price.
As a manager, I want to remove items from the menu so that customers cannot choose the unavailable items.	<ul style="list-style-type: none"> - Manager can remove a specified item from the menu.
As a manager, I want to update the description, ingredient, picture, category, or cost to the menu so that customers can get latest and correct information of the menu.	<ul style="list-style-type: none"> - Manager can update the description, ingredient, picture, category, or price of an item - The updated information should be shown after the customer refresh the page
As a manager, I want to update the order in which menu items are shown within a category so that customers can see the items I want them to see first.	<ul style="list-style-type: none"> - Manager should be able to modify the order of items within a category - The order of items can be modified by clicking arrows
As a manager, I want to update the order in which menu categories are shown on the menu so that customers can see the categories I want them to see first.	<ul style="list-style-type: none"> - Manager should be able to modify the order of categories - The order of categories can be modified by clicking arrows

Objective 8: Rating and Recommendation System (Novel)

Customers can rate every item they ordered after they requested the bill. An average rating for each item will be calculated. To assist in decision-making, the menu also highlights the top five highest-rated items.

User Stories	Acceptance Criteria
--------------	---------------------

As a customer, I want to see the top 5 highest-rated items so that I can order popular items that are highly recommended by other customers.	- The menu interface includes an option for customers to view the top 5 highest-rated items.
As a customer, I want to rate every item I ordered so that I can express my feelings.	- Customer can rate the items they ordered after checkout by clicking the stars.

Objective 9: Analytics (Novel)

Managers can view the sales situation and turnover of all items within a specific period. This feature allows managers to gain a comprehensive understanding of the restaurant's sales performance, enabling them to plan more effectively for the future.

User Stories	Acceptance Criteria
As a manager, I want to see the sales of each item so that I can identify popular items, enabling me to make decisions about menu optimization and promotional strategies.	- Manager should be able to see the sales of each item within a given time slot.
As a manager, I want to see the sales data so that I can have a comprehensive understanding of the restaurant's operations.	- The sales data should be presented in a clear and easily understandable format, such as charts, graphs, or tables.