FPT ACADEMY INTERNATIONAL

FPT – APTECH COMPUTER EDUCATION

Center name: ACE-HCMC-2-FPT.

Address: 391A Nam Ky Khoi Nghia, District 3, Ho Chi Minh City, Viet Nam.

RESTAURANT ORDERING MANAGEMENT

Supervisor: Ms. Le Mong Thuy

Semester: 2

Batch no: T1.2406.E1

Group no: 6

Member:

Order	Name	StudentID
1	Pham Nguyen Hung	Student1576000
2	Dinh Tung Lam	Student1565810
4	Nguyễn Đình Đức Mạnh	Student1580658

Date: May, 2025

Catalog

ACKNOWLEDGMENT

First and foremost, we would like to express our sincere gratitude to **Ms. Lê Mộng Thúy**, who has wholeheartedly guided, supported, and provided us with valuable feedback throughout the process of completing this project. Her knowledge, experience, and dedication have been a great source of motivation, enabling us to successfully accomplish our work.

We would also like to extend our heartfelt thanks to **FPT Aptech** for providing us with excellent facilities, learning materials, and a positive academic environment, which greatly contributed to our ability to research and develop this project effectively.

Last but not least, we sincerely appreciate the efforts of all our team members who have worked diligently, supported each other, and contributed to achieving the project's goals throughout the entire process.

Although we have made every effort, due to limitations in time and practical experience, some shortcomings are inevitable. We sincerely hope to receive constructive feedback from Ms. Thúy and our fellow classmates to help improve and perfect the project.

Sincerely, thank you!

SYNOPSIS

This project aims to digitalize the ordering and table reservation system for a restaurant to improve service efficiency and customer experience.

The restaurant employs various staff members including waiters, kitchen staff, bartenders, cashiers, and managers. It offers a wide range of food and drinks, each categorized into specific types. Each dish or drink belongs to exactly one category, and only managers have permission to add new items and categories to the menu.

The restaurant is divided into three main areas:

- Private Rooms, each accommodating up to 20 people.
- Standard Dining Area, which includes tables for 2, 4, or 6 people.

Each table or room is uniquely numbered and can have one of the following statuses: available, waiting for food, all food served, or payment requested.

Ordering is performed by waiters. When customers order, the waiter selects the corresponding table number in the system and chooses the desired dishes and drinks. Upon submission, the table status changes to waiting for food, and the order is routed to the kitchen

(for food) and bar (for drinks). When items are prepared, kitchen or bar staff mark them as completed, notifying the waiter to serve them. The waiter confirms each dish served. Once all items are delivered, the table status changes to all food served. Additional items can be ordered later, and the table status will update accordingly.

If any ordered items are not served within 30 minutes, a notification is sent to the kitchen.

When customers are ready to pay, the waiter requests payment through the system. The cashier is notified, and the table enters the payment requested state. The cashier then prints the invoice containing the bill ID, table number, total amount, and payment method (cash, transfer, or card). After completing the transaction, the table returns to available.

Customers can view the menu online. To reserve a table, a customer must have an account and submit a reservation form including name, phone number, arrival time, number of guests, preferred area (private room, couples area, or standard), and optional pre-ordered items. Couples area only allows reservations for two people. Reservations for private rooms require a pre-order total of at least 5 million VND and must be prepaid via bank transfer.

This project enhances restaurant management by automating ordering, reservation, kitchen coordination, and billing, resulting in faster service, improved accuracy, and a better customer experience.

FROM VIEWPOINT OF VISITORS

Visitors (customers) are the end users of the system. They are allowed to browse the food and drink menu without logging in. However, in order to reserve a table and pre-order food, they are required to create and log into a customer account.

The main functions available to customers include:

- View Menu: Customers can explore the full list of food and drink items, categorized by type.
- Table Reservation:
  - Select the preferred dining area (Private Room, Couple Area, or Standard Area).
  - Provide reservation details including name, phone number, arrival time, and number of guests.
  - If selecting the Couple Area, only 2 guests are allowed.

- If selecting a Private Room, the total value of pre-ordered food must be at least 5 million VND, and prepayment via bank transfer is required.

Pre-Order Food (Optional): Customers may choose to pre-order dishes to reduce waiting time upon arrival.

FROM VIEWPOINT OF ADMINISTRATOR

The administrator (restaurant manager) has full access to the system and is responsible for maintaining smooth operations, managing the menu, and controlling user accounts.

The main functions available to the administrator include:

Menu Management:

- Add new food and drink categories.
- Add new food and drink items.
- View detailed information of existing menu items.
- Update menu item details.
- Change item availability status:
    - Available / Unavailable.

User Account Management:

- Create new user accounts for staff (waiters, kitchen/bar staff, cashiers).
- Update user account status:
    - Active / Inactive.

These functions allow the administrator to ensure that the system is up to date, the menu is properly maintained, and all staff accounts are securely managed for efficient restaurant operations.

SCOPE OF WORK

1. Administrator (Manager) Module.

- Manage food and drink categories.
- Add, update, and deactivate food and drink items.

- Manage user accounts (create staff accounts, activate/deactivate access).

2. Waitstaff Module.

- Select tables and take customer orders (food & drinks).
- Update the number of guests at a table.
- Send orders to the kitchen or bar.
- Confirm when dishes have been served.
- Request bill payment on behalf of customers.

3. Kitchen and Bar Staff Module.

- Receive order notifications.
- Mark dishes or drinks as completed.
- Receive alerts for delayed orders (over 20 or 30 minutes not served).

4. Cashier Module.

- View payment requests from waitstaff.
- Generate and print invoices (including table number, order details, total amount, and payment method).
- Process payments (cash, card, Momo).
- Reset table status after payment is completed.

5. Customer Module.

View the food and drink menu.

Register and log in to a customer account.

Reserve tables by selecting area (Private Room, Couple Area, or Standard Area).

Pre-order dishes (optional).

Prepay via bank transfer when required (e.g., Private Room reservation over 5 million VND).

HARDWARE AND SOFTWWARE REQUIREMENTS

1. Hardware requirement.

Client-Side (Customer, Staff, Admin Interface)

- Device: Laptop / Desktop / Tablet / Smartphone

- Processor: Intel Core i3 or higher
- RAM: 4 GB or higher
- Storage: 512 MB free space (for cache & browsing)
- Display: Minimum 1280×720 resolution
- Internet Connection: Required (Stable broadband or Wi-Fi)

Server-Side

- Processor: Intel Core i5 or higher (Quad Core recommended)
- RAM: Minimum 8 GB (16 GB recommended for scalability)
- Storage: SSD with at least 100 GB free space
- Network: High-speed internet with static IP or cloud-based hosting
- Operating System: Ubuntu 20.04 LTS / CentOS / Windows Server 2019

2. Software requirement.

Frontend (Client-side)

- Framework: ReactJS (v17 or above)
- Languages: HTML5, CSS3, JavaScript (ES6+)
- Libraries/Tools: Axios, Bootstrap 5 (or Material UI), React Router, Redux (optional)

Backend (Server-side)

- Framework: Laravel (v8 or above)
- Language: PHP (v7.4 or above)
- Web Server: Apache / Nginx
- Authentication: Laravel Sanctum / Passport (optional)
- API Format: RESTful API (JSON response)

Database

- DBMS: MySQL (v5.7 or above)
- Tool for DB Management: phpMyAdmin / MySQL Workbench

Development Tools

- IDE: Visual Studio Code / PhpStorm / Sublime Text
- Package Managers: Composer (for PHP), npm/yarn (for ReactJS)
- Version Control: Git (GitHub/GitLab/Bitbucket)

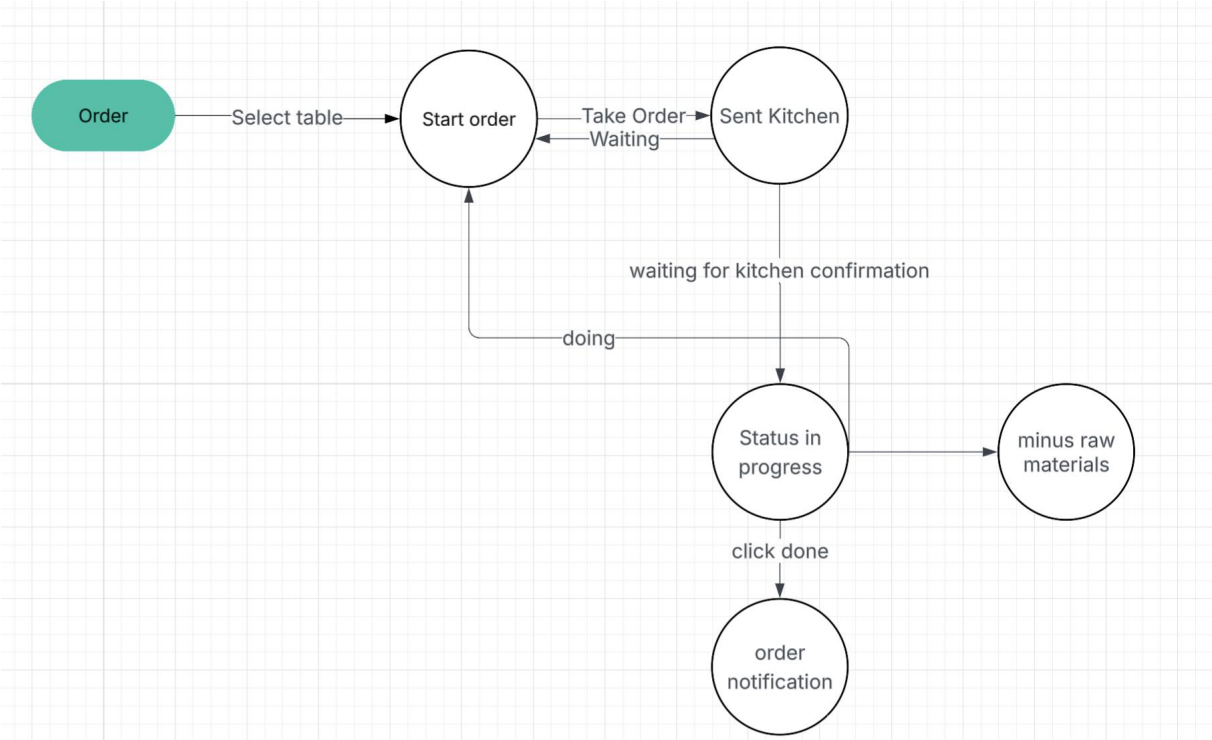
TASKSHEET 1

Project: RESTAURANT ORDERING MANAGEMENT		**PrJ Name**	**Act Plan Prepare By**	**Date of preparation of activity plan**			
No	**Task**			**Start date**	**Actual days**	**Team member name**	**Status**
1	**Acknowledgment**	Restaurant ordering management	Hưng	6 May, 2025	1	Hưng	Completed
2	**Synopsis**			6 May, 2025	1	Lâm	Completed
3	**From viewpoint of visitors**			6 May, 2025	1	Khang	Completed
4	**From viewpoint of Administrator**			6 May, 2025	1	Mạnh	Completed
5	**Scope of work**			6 May, 2025	1	Hưng, Lâm	Completed
6	**Hardware requirement**			6 May, 2025	1	Hưng	Completed
7	**Software requirement**			6 May, 2025	1	Mạnh	Completed
8	**Task sheet 1**			6 May, 2025	1	Hưng	Completed
Date: 26 April, 2025							
Signature of Instructor Ms. Le Mong Thuy				Signature of team Leader Pham Nguyen Hung			

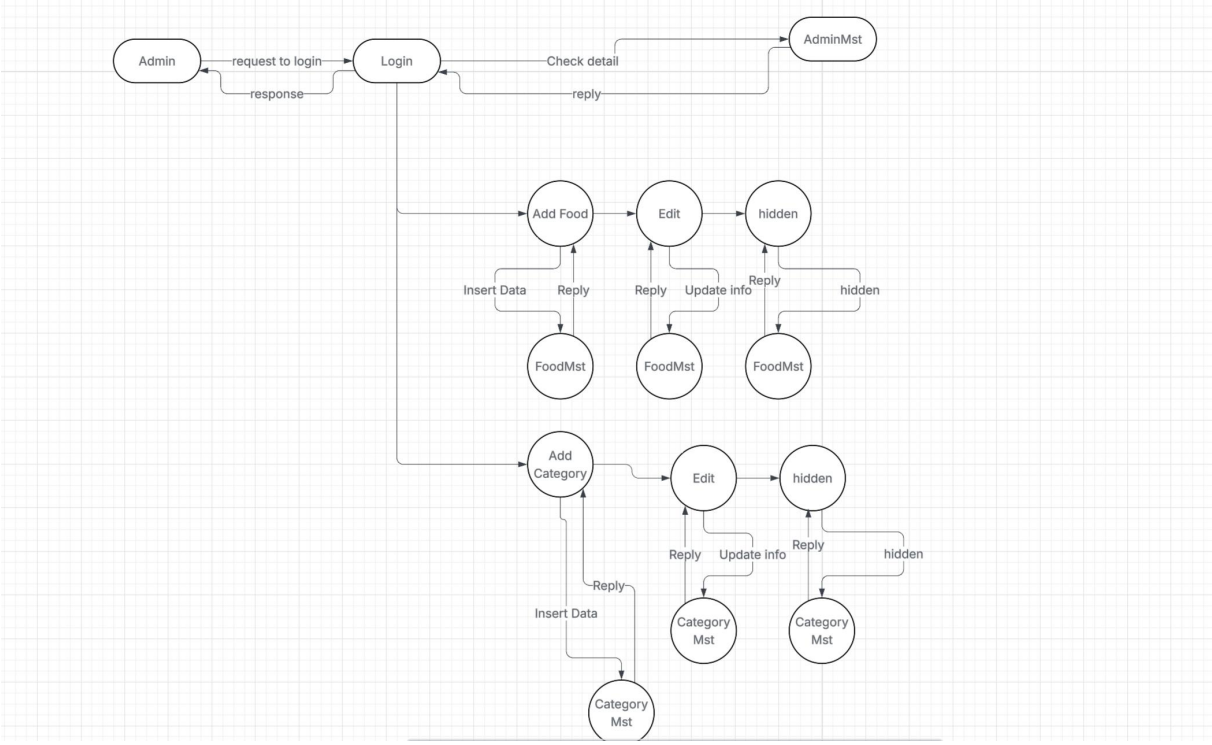
PART 2. SYSTEM DESIGN ANALYSIS

1. Data flow Diagram - DFD.

Order - DFD



Admin - DFD



2. Integrity constraints.

2.1. Table roles.

Column Name	Data Type	Key	Notes
id	INT	PRIMARY KEY	Auto-incremented, uniquely identifies each role.
role_code	VARCHAR(20)	UNIQUE	Must be unique and not null; used as the role code.
role_name	VARCHAR(50)		Must not be null; stores the name of the role.

- PRIMARY KEY ensures that each record is unique.
- UNIQUE ensures that no two roles have the same role_code.
- NOT NULL ensures that a value must be provided for that column.

2.2. Table User.

Column Name	Data Type	Key	Notes
id	INT	PRIMARY KEY	Auto-incremented, uniquely identifies each user.
staff_code	VARCHAR(20)	UNIQUE	Optional but must be unique if provided; used as a unique staff code.
name	VARCHAR(100)		Required; stores the full name of the user.
phone	VARCHAR(10)	UNIQUE	Required; must be a unique 10-digit phone number.
password	VARCHAR(255)		Required; stores the hashed password.
role_id	INT	FOREIGN KEY	Required; references id in the roles table to assign a user role.

2.3. Table Categories.

Column Name	Data Type	Key	Notes

id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each category.
name	VARCHAR(255)		Required (NOT NULL); name of the food or drink category.
description	ENUM('Drink', 'Food')		Required; specifies whether the category is a drink or food.

2.4. Table Food.

Column Name	**Data Type**	**Key**	**Notes**
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each food item.
name	VARCHAR(255)		Required (NOT NULL); name of the food item.
price	DECIMAL(10,2)		Required (NOT NULL); price of the food item with 2 decimal places.
category_id	INT	FOREIGN KEY	Optional; references id in the categories table to indicate the category.

2.5. Table Area.

Column Name	**Data Type**	**Key**	**Notes**
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each dining area.
name	ENUM('phòng riêng', 'phòng thường')		Required (NOT NULL); type of area: private room (phòng riêng) or regular room (phòng thường).

2.6. Table Tables.

Column Name	**Data Type**	**Key**	**Notes**
id	INT	PRIMARY	Auto-incremented; uniquely identifies

		KEY	each table.
area_id	INT	FOREIGN KEY	Optional; references id in the area table to assign the table to a specific dining area.
table_number	VARCHAR(5)		Required (NOT NULL); table identifier (e.g., T01, A5).
capacity	INT		Required; number of people the table can accommodate.
status	ENUM('trống', 'chờ món', 'đã lên hết món', 'chờ thanh toán')		Optional; default is 'trống'; indicates the current status of the table.

2.7. Table Reservations.

Column Name	**Data Type**	**Key**	**Notes**
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each reservation.
user_id	INT	FOREIGN KEY	References id in the users table; identifies who made the reservation.
table_id	INT	FOREIGN KEY	References id in the tables table; the table being reserved.
arrival_time	DATETIME		Required (NOT NULL); date and time of arrival.
guest_count	INT		Required; number of guests expected.
pre_order_total	DECIMAL(10,2)		Optional; total amount of any pre-ordered food/drinks; defaults to 0.

2.8. Table Orders

Column Name	**Data Type**	**Key**	**Notes**
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each order.

table_id	INT	FOREIGN KEY	References id in the tables table; indicates the table where the order was placed.
staff_id	INT	FOREIGN KEY	References id in the users table; staff member handling the order.
order_time	DATETIME		Defaults to current timestamp; time the order was placed.
status	ENUM('chờ món', 'đã lên hết món', 'chờ thanh toán')		Optional; default is 'chờ món'; indicates the current state of the order.

2.9. Table order_details.

Column Name	**Data Type**	**Key**	**Notes**
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each order item.
order_id	INT	FOREIGN KEY	References id in the orders table; links the item to a specific order.
food_id	INT	FOREIGN KEY	References id in the food table; specifies which food item was ordered.
quantity	INT		Required (NOT NULL); quantity of the food item ordered.
kitchen_status	ENUM('đang chờ', 'xác nhận làm', 'đã xong')		Optional; defaults to 'đang chờ'; tracks preparation status in the kitchen.
created_at	DATETIME		Automatically set to current time; when the order item was added.
delivered_at	DATETIME		Nullable; when the dish was delivered to the customer (can be updated later).

2.10. Table Bill.

Column Name	Data Type	Key	Notes
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each bill.
order_id	INT	FOREIGN KEY	References id in the orders table; links the bill to the corresponding order.
cashier_id	INT	FOREIGN KEY	References id in the users table; identifies the cashier who processed the bill.
total_amount	DECIMAL(10,2)		Required (NOT NULL); total amount to be paid.
payment_method	ENUM('tiền mặt', 'chuyển khoản', 'thẻ')		Required (NOT NULL); payment method used: cash, transfer, or card.
paid_at	DATETIME		Defaults to CURRENT_TIMESTAMP; time the bill was paid.

2.11. Table Promotions.

Column Name	Data Type	Key	Notes
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each promotion.
code	VARCHAR(20)	UNIQUE, NOT NULL	Unique promotion code used by customers.
description	VARCHAR(255)		Optional; short description of the promotion.
discount_percent	INT	CHECK	Must be between 0 and 100; indicates the discount percentage.
start_date	DATE	NOT NULL	The date when the promotion becomes valid.
end_date	DATE	NOT NULL	The date when the promotion expires.
active	BOOLEAN		Defaults to TRUE; indicates whether the promotion is currently

			active.

2.12. Table inventory_receipts.

Column Name	Data Type	Key	Notes
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each inventory receipt record.
staff_id	INT	FOREIGN KEY	References id in the users table; identifies the staff who created the receipt.
receipt_date	DATETIME		Defaults to CURRENT_TIMESTAMP; records the date and time of stock entry.
note	TEXT		Optional; additional notes or remarks about the inventory receipt.

2.13. Table ingredients.

Column Name	Data Type	Key	Notes
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each ingredient.
name	VARCHAR(100)	NOT NULL	Name of the ingredient.
unit	VARCHAR(20)	NOT NULL	Unit of measurement (e.g., kg, gram, liter).

category_id	INT	FOREIGN KEY	References id in ingredient_categories; categorizes the ingredient.
stock_quantity	DECIMAL(10,2)		Quantity currently in stock; def

2.14. Table ingredient_categories.

Column Name	Data Type	Key	Notes
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each ingredient category.
name	VARCHAR(100)	NOT NULL	Name of the ingredient category (e.g., Food, Spice, Beverage, etc.).

2.15. Table food_ingredients.

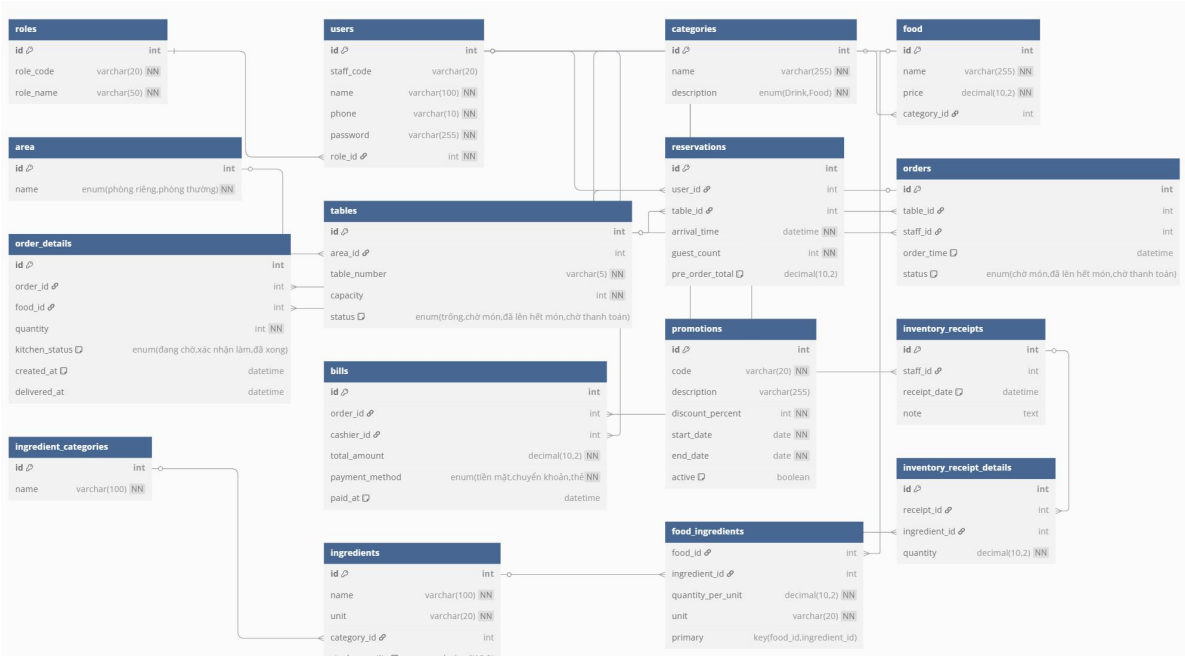
Column Name	Data Type	Key	Notes
food_id	INT	PRIMARY KEY (part)	References id in food; identifies the food item.
ingredient_id	INT	PRIMARY KEY (part)	References id in ingredients; identifies the ingredient used in the food item.
quantity_per_unit	DECIMAL(10,2)	NOT NULL	Quantity of the ingredient required per unit of food.
unit	VARCHAR(20)	NOT NULL	Unit of measurement (e.g., gram, ml, piece) for the ingredient in the food recipe.

2.16. Table inventory_receipt_details.

Column Name	Data Type	Key	Notes
id	INT	PRIMARY KEY	Auto-incremented; uniquely identifies each inventory receipt detail.
receipt_id	INT	FOREIGN KEY	References id in inventory_receipts; identifies the

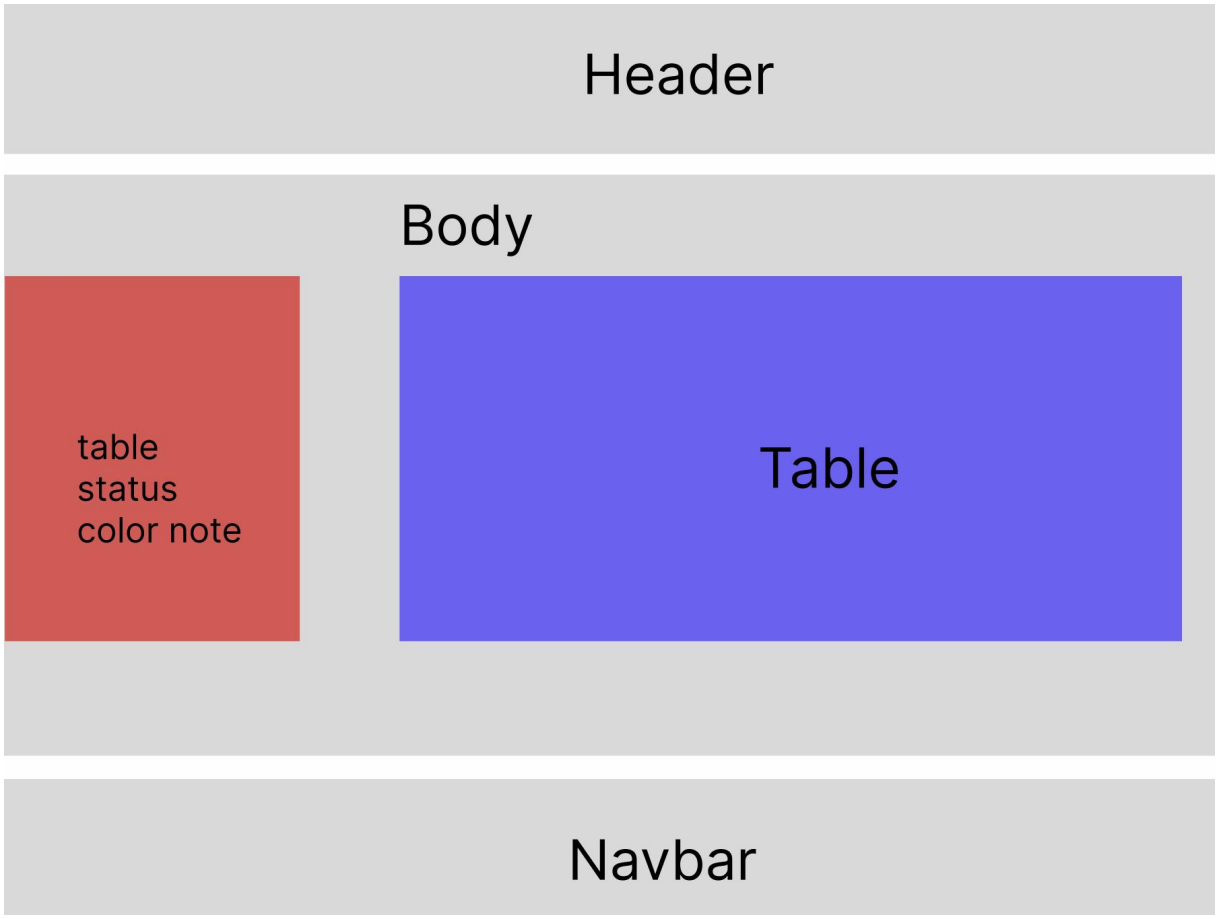
			inventory receipt.
ingredient_id	INT	FOREIGN KEY	References id in ingredients; identifies the ingredient received.
quantity	DECIMAL(10,2)	NOT NULL	Quantity of the ingredient received in the inventory receipt.

3. Entity Relationship Diagram (ERD).

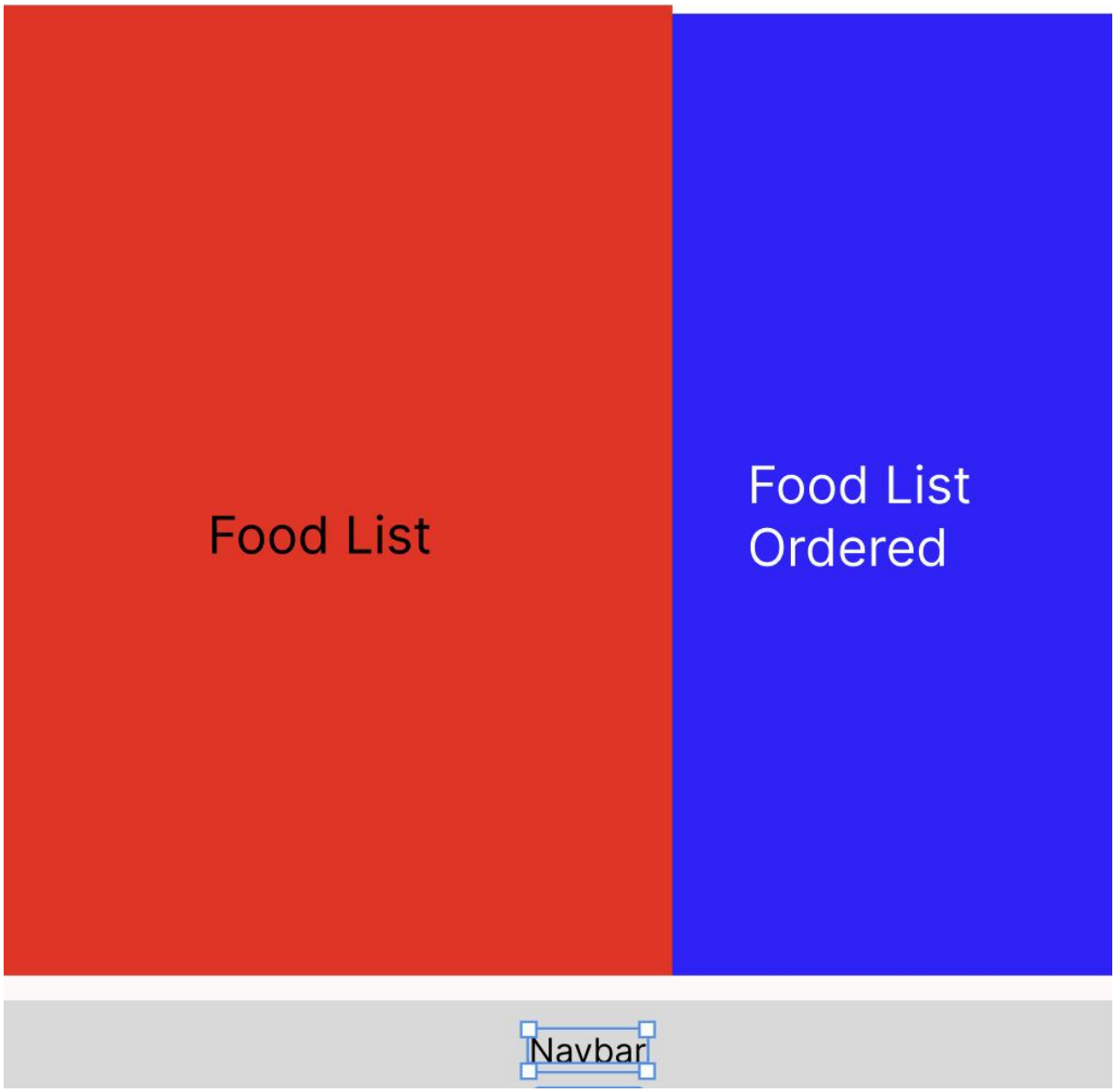


4. Prototypes

4.1. Order page.

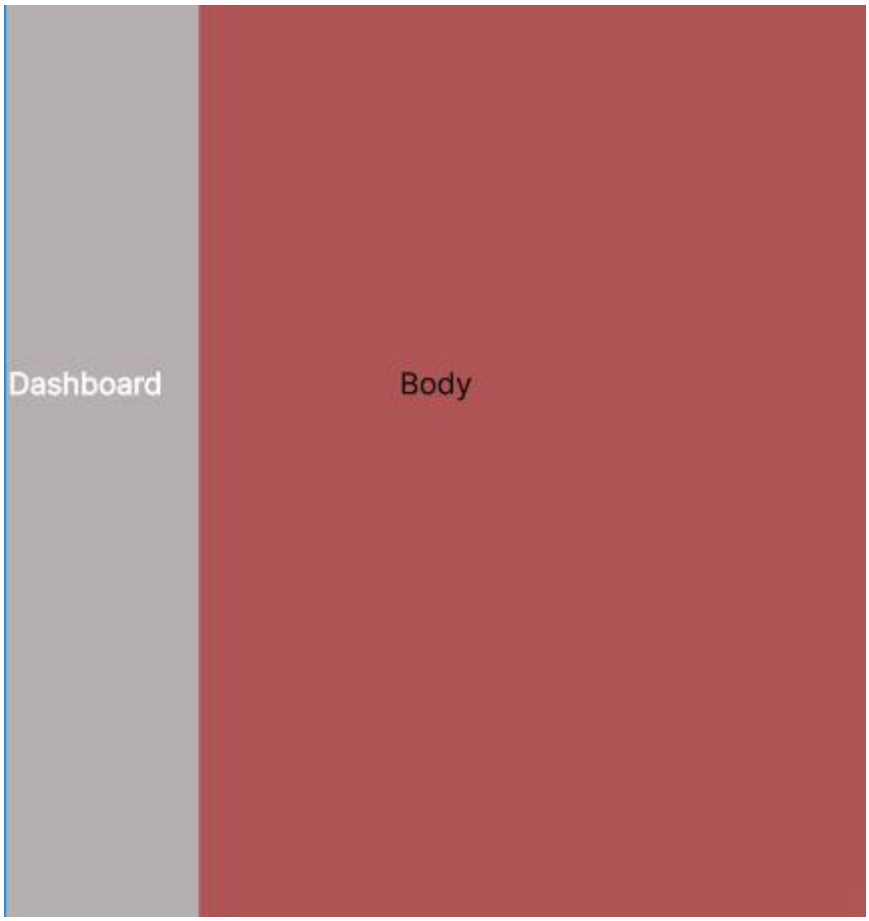


4.2. Order detail page.



4.3. Kitchen page.

4.4. Admin page.



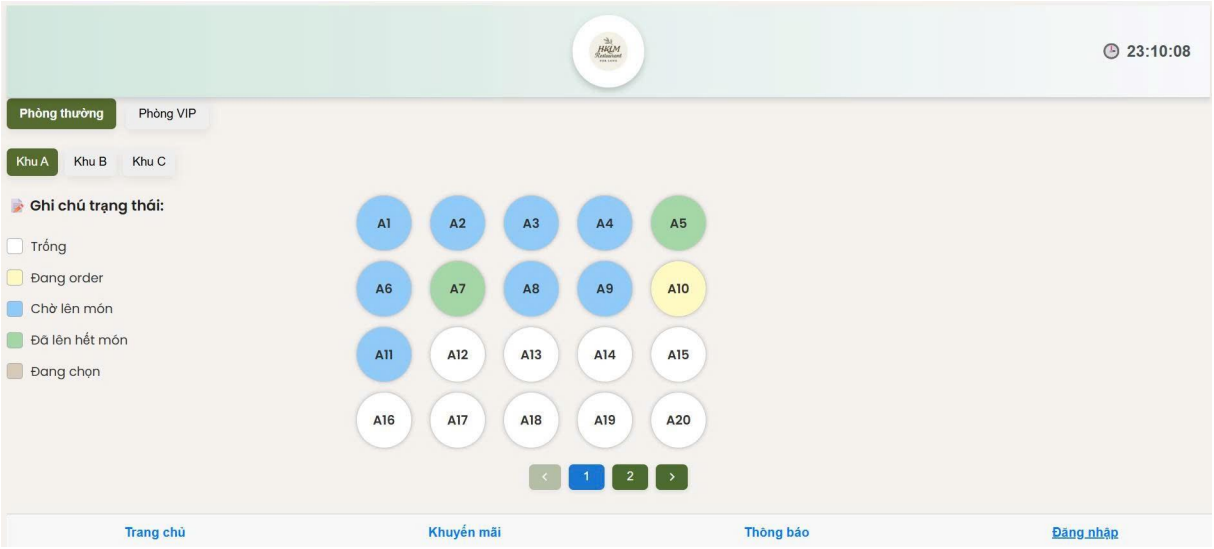
TASKSHEET 2

<table>
<tr><td colspan="2">Project:
RESTAURANT ORDERING MANAGEMENT</td><td rowspan="2">PrJ Name</td><td rowspan="2">Act Plan Prepare By</td><td colspan="4">Date of preparation of activity plan</td></tr>
<tr><td>No</td><td>Task</td><td>Start date</td><td>Actual days</td><td>Team member name</td><td>Status</td></tr>
<tr><td>1</td><td>Data flow Diagram</td><td rowspan="5">Restaurant ordering management</td><td rowspan="5">Hưng</td><td>6 May, 2025</td><td>4</td><td>Hung, Lam</td><td>Completed</td></tr>
<tr><td>2</td><td>Integrity constraints</td><td>6 May, 2025</td><td>4</td><td>Manh, Lam</td><td>Completed</td></tr>
<tr><td>3</td><td>ERD</td><td>6 May, 2025</td><td>4</td><td>Hung, Manh</td><td>Completed</td></tr>
<tr><td>4</td><td>Prototypes</td><td>6 May, 2025</td><td>4</td><td>Lam</td><td>Completed</td></tr>
<tr><td>5</td><td>Task sheet 2</td><td>6 May, 2025</td><td>1</td><td>Hưng</td><td>Completed</td></tr>
<tr><td colspan="8">Date: 10 May, 2025</td></tr>
<tr><td colspan="3">Signature of Instructor

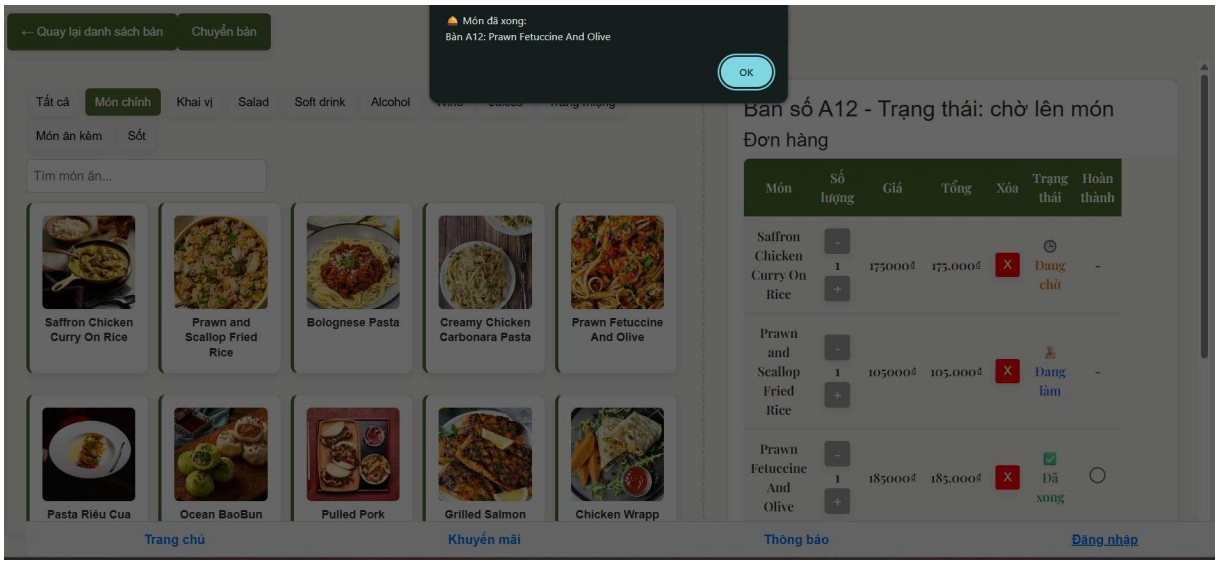
Ms. Le Mong Thuy</td><td colspan="5">Signature of team Leader

Pham Nguyen Hung</td></tr>
</table>

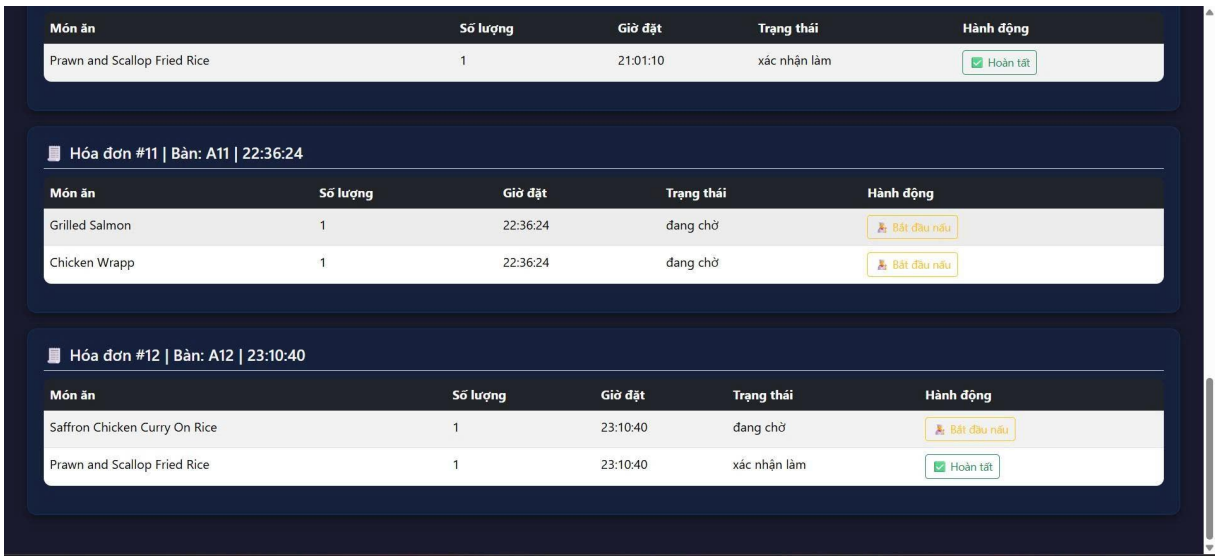
Part 3. Side Map



First, select a table



After selecting a table, press start ordering, after ordering, press send to kitchen



When sending to the kitchen, the kitchen confirms making the dish, then the interface on the order side is making. When the kitchen presses done, the order side will display a notification that the dish is ready.

TASKSHEET 3

Project: RESTAURANT ORDERING MANAGEMENT		**PrJ Name**	**Act Plan Prepare By**	**Date of preparation of activity plan**			
No	**Task**			**Start date**	**Actual days**	**Team member name**	**Status**
1	**CRUD Food**	Restaurant ordering management	Hưng	11 May, 2025	2	Lâm	Completed
2	**CRUD Category**			14 May, 2025	2	Lâm	Completed
3	**Order (add item, send item)**			17 May, 2025	4	Lâm	Completed
4	**Kitchen (Confirm dish in progress, confirm dish done)**			11 May, 2025	4	Hưng	Completed
5	**Payment**			16 May, 2025	4	Hưng	Completed
6	**Warehouse (Import/Export)**			11 May, 2025	8	Manh	Completed
7	**Desigh UX/UI**			22 May, 2025	8	Hưng, Mạnh, Lâm	Completed
8	**Task sheet 2**			28 May, 2025	1	Hưng	Completed
Date: 28 May, 2025							
Signature of Instructor Ms. Le Mong Thuy				Signature of team Leader Pham Nguyen Hung			