| **FPT-APTECH COMPUTER EDUCATION** |
| --- |
| eProject Document |
| [Electronic Menu] |
|  |
| | **Iced Tea** | | | --- | --- | | **Group Member** | Trịnh Văn Phúc - TH2209059 - OV-7023\_ADSE  Nguyễn Thị Hương - TH2209079 - OV-7023\_ADSE  Trần Văn Tiến - TH2209070 - OV-7023\_ADSE  Nguyễn Sỹ Toàn - TH2209063 - OV-7023\_ADSE  Nguyễn Đức Lâm - TH2209061 - OV-7023\_ADSE | | **Instructor** | Nguyen Tuan | | **Batch** | T2210M | | **Semester** | 4 | |
|  |

| - Hanoi, 10/2024 - |
| --- |

<This is the template document. Replace any text in <> with your text. When you are done, there should be no <> or text surrounded by <> in this document. Remember, this purpose of this document is to be useful: not just work to get a grade.>

***Contents***

[**1. Introduction**](#) **3**

[**2. Problem Definition**](#_gjdgxs) **4**

[2.1. Problem Abstraction](#_30j0zll) 4

[2.2. The Current System](#_1fob9te) 4

[2.3. The Proposed System](#_3znysh7) 4

[2.4. Boundaries of the System](#_2et92p0) 4

[2.5. Development Environment](#) 4

[**3. Requirements and Business Flow**](#) **5**

[3.1. Customer Requirement Specification](#) 5

[3.2. Activity Diagram](#) 5

[3.3. Use Case Diagram](#) 5

[3.4. Use Case Specification](#) 5

[3.5. Other Concerns<Optional>](#) 6

[**4. Design**](#) **7**

[4.1. System Architecture](#) 7

[4.2. Class Diagram](#) 7

[4.3. Class Diagram Explanation](#) 7

[4.4. Sequence Diagram (Optional)](#) 7

[4.5. Collaboration Diagram (Optional)](#) 7

[4.6. State Diagram (Optional)](#) 7

[4.7. Entity Relationship Diagram](#_tyjcwt) 7

[4.8. Database Design](#_3dy6vkm) 7

[4.9. Algorithms (optional)](#_1t3h5sf) 7

[4.10. Others (optional)](#) 8

[**5. System Prototype**](#) **9**

[**6. Management and Project Planning**](#) **10**

[6.1. Management Approach](#) 10

[6.2. Project Plan](#) 10

[6.3. Task Sheet](#) 10

[6.4. Meeting Minutes (Optional)](#) 10

[**7. Checklists**](#_4d34og8) **11**

[7.1. Check List of Validation](#_2s8eyo1) 11

[7.2. Submission Checklist](#_17dp8vu) 11

[**8. Screenshots**](#) **11**

[**9. Coding Convention**](#) **11**

[**10. Other Concerns<Optional>**](#) **11**

[**Appendix**](#) **11**

[Glossary [Optional]](#) 12

[References [Optional]](#) 12

[Others<Optional>](#) 12

# **Introduction**

This document provides a comprehensive overview of the *Electronic Menu* project, a digital solution designed to enhance customer experience and streamline restaurant operations. The project aims to replace traditional paper menus with an intuitive electronic menu interface that allows customers to view and order items directly from their personal devices, reducing wait times and improving service efficiency.

The document is organized into several sections to facilitate easy navigation and comprehension:

* **Problem Definition**: This section discusses the current challenges faced by traditional menu systems, including inefficiencies and customer dissatisfaction. It outlines the limitations of existing practices and introduces the proposed *Electronic Menu* solution, which aims to address these issues effectively. It also defines the boundaries of the system and the development environment used for implementation.
* **Requirements and Business Flow**: Here, we specify the requirements from the customer perspective, provide an activity diagram that illustrates the workflow, and present use case diagrams that outline user interactions with the system. This section also includes detailed use case specifications, which describe the functionality expected from the system.
* **Design**: This section delves into the technical design of the *Electronic Menu*, covering the system architecture, class diagrams, and entity relationship diagrams that depict the underlying structure. It also includes a database design section and optional diagrams, such as sequence and collaboration diagrams, to illustrate interactions and states within the system.
* **System Prototype**: A prototype of the *Electronic Menu* interface is presented in this section, showcasing the design and functionality of key features.
* **Management and Project Planning**: This section outlines the management approach taken for the project, including the project plan and task assignments to ensure effective execution. Meeting minutes, if applicable, are also documented here.
* **Checklists**: To ensure thoroughness, this section includes validation and submission checklists that outline the necessary criteria for project completion and delivery.
* **Screenshots**: Visual representations of the application’s interface are provided to illustrate its functionality and usability.
* **Coding Convention**: This section details the coding standards to maintain consistency and quality in the codebase throughout the development process.
* **Other Concerns** (Optional): Additional considerations that may impact the project are addressed here.
* **Appendix**: Supplementary material related to the project is included in this section for reference.
* **Glossary** (Optional): Definitions of key terms and acronyms used throughout the document are provided for clarity.
* **References** (Optional): This section lists sources and materials referenced in the document.

This structured organization allows for a comprehensive understanding of the *Electronic Menu* project, ensuring that all relevant information is presented clearly and logically for stakeholders, developers, and any other interested parties.

# **Problem Definition**

## ***Problem Abstraction***

The traditional menu systems used in restaurants and cafes often lead to various inefficiencies that negatively impact customer satisfaction and operational effectiveness. During peak hours, many establishments experience overwhelming customer volumes, resulting in long wait times for service. Customers often express frustration while waiting in line, which can lead to complaints and a negative dining experience.

In addition, the manual nature of taking orders can lead to errors and misunderstandings between staff and customers. This not only affects order accuracy but also increases the workload on servers, further exacerbating delays and inefficiencies. Furthermore, the inability to easily update menu items or prices necessitates frequent printing costs, leading to additional financial burdens for restaurant owners.

To address these issues, there is a clear need for a modernized solution that enhances the customer experience while streamlining internal operations. The proposed *Electronic Menu* system aims to provide a user-friendly interface that allows customers to browse the menu, place orders, and make payments directly from their personal devices. This system will not only reduce wait times and improve order accuracy but also enable restaurants to update their menus quickly and cost-effectively. By implementing this solution, restaurants can enhance operational efficiency, reduce staffing pressures, and ultimately increase customer satisfaction and revenue.

## ***The Current System***

**System Description:**The current system utilizes traditional paper menus and manual order-taking by staff, where customers must wait in line to place their orders. Changes to the menu require reprinting, which is time-consuming and costly.

**System Evaluation:**The existing system is ineffective during high-traffic periods, causing long queues and dissatisfaction among customers. The manual process is prone to errors, which can negatively impact the overall dining experience and operational efficiency.

## ***The Proposed System***

**System Objective:**The *Electronic Menu* system aims to enhance the dining experience by providing a digital solution that allows customers to view and interact with the menu using their personal devices. It seeks to reduce wait times, improve order accuracy, and enable real-time menu updates.

**Main Functions:**

* Customers can scan a QR code to access the digital menu on their smartphones.
* Customers can place orders directly from their devices, eliminating the need for manual order-taking by staff.
* Restaurant staff can easily update menu items, prices, and images in real time without incurring printing costs.
* Integration with payment gateways allows customers to make secure payments directly through the system.

## ***Boundaries of the System***

**The system includes:**

* User accessibility via personal devices using QR codes.
* Streamlined order placement directly from customer devices.
* Real-time menu management by restaurant staff.
* Payment processing through secure payment gateways.
* Synchronization with the restaurant's existing POS system for accurate order tracking.

**The system does not include:**

* Delivery functionalities or any third-party integrations beyond payment processing.
* Features for managing inventory or staff scheduling.

## ***Development Environment***

***Operating System:*** *Windows*

***Programming Language:*** *Java, JavaScript*

***Frameworks:*** *Spring Boot (for backend), React.js (for frontend)*

***Styling:*** *Tailwind CSS*

***Build Tool:*** *Vite*

***Database:*** *MySQL*

***API Testing Tool:*** *Postman*

*●* ***Development Tools:*** *Visual Studio Code, intellij, Git version control tool.*

# **Requirements and Business Flow**

This section details the functional and non-functional requirements for the *Electronic Menu* system, emphasizing critical aspects that will influence software implementation. The goal is to present a compact yet comprehensive summary, highlighting business processes, customer needs, and system interactions.

## ***Customer Requirement Specification***

The customer requires a comprehensive electronic menu system for a restaurant that enables efficient order management, payment processing, and customer feedback collection. The system should cater to various user roles, such as **Manager**, **Chef**, **Waiter**, and **Customer**, each with their respective functionalities. The main goals are to streamline the ordering process, enhance the dining experience, and support managerial tasks like coupon management, sales reporting, and staff oversight.

Key functionalities include:

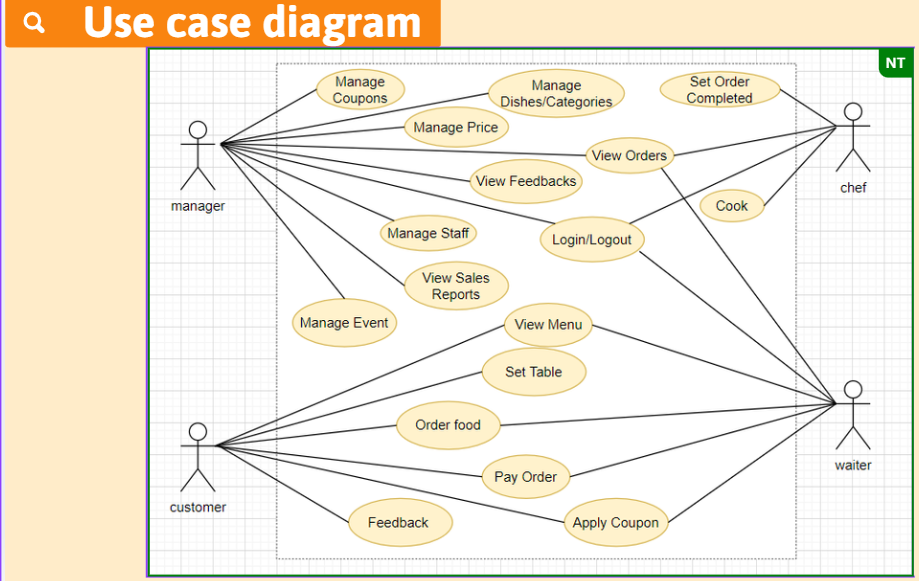
* **Digital Menu Access**: Customers can access a digital menu by scanning a QR code at their table.
* **Order Placement**: Customers can place orders directly through their devices, minimizing the need for interaction with waitstaff.
* **Real-time Menu Updates**: Restaurant staff can manage the menu (add, update, remove items) in real time, without reprinting costs.
* **Secure Payment Options**: The system should support secure payment options, enabling customers to pay directly on the platform.
* **System Synchronization**: Integration with the system for real-time order updates and accurate tracking.
* **Scalable Architecture**: The system should handle high traffic during peak hours without performance degradation.

## ***Activity Diagram***

The following activity diagrams illustrate core business processes:

1. **Digital Menu Access and Order Placement**: Customers scan a QR code to view the digital menu on their device, browse available items, and place orders directly.
2. **Order Fulfillment and Real-time Updates**: Once the customer places an order, the system updates the POS in real time, notifying the kitchen for preparation and sending updates to the waitstaff.
3. **Secure Payment Processing**: After dining, customers can pay securely through the platform, which syncs with the POS for accurate transaction records.
4. **Real-time Menu Management by Staff**: Restaurant staff can add, update, or remove menu items in real time, instantly reflecting these changes on customers' devices.
5. **System Scalability for Peak Times**: The system architecture ensures smooth operation during high-traffic periods, supporting many simultaneous users without performance issues.

## ***Use Case Diagram***

**

## ***Use Case Specification***

*<Write down all non-trivial use cases. This should reflect what you get when your team does the system analysis. Use the template to write the detailed specification for use cases>*

*<Use case template:*

| **USE CASE SPECIFICATION** | | | | |
| --- | --- | --- | --- | --- |
| **Use-case No.** | UC001 | **Use-case Version** | | 1.0 |
| **Use-case Name** | Order Food | | | |
| **Author** | Project Team Members | | | |
| **Date** | 27/10/2024 | **Priority** | High | |
| **Actor:** Customer, Waiter, Manager, Chef  **Summary:**  This use case describes the process where a customer places an order through the Electronic Menu. The customer selects menu items and submits the order, which is then processed by the restaurant staff and kitchen.  **Goal:**  Enable customers to place food orders directly through the Electronic Menu system and send orders to the kitchen for preparation.  **Triggers**  *The customer decides to place an order after viewing the* ***Electronic Menu****.*    **Preconditions:**  The customer is seated at a table in the restaurant.  The customer has accessed the **Electronic Menu** by scanning a QR code.  Menu items are available and updated in the system.  **Post Conditions:**  The order is successfully placed and sent to the kitchen.  The customer receives confirmation of their order.  The order status is updated in the system for tracking and preparation.  **Main Success Scenario:**   1. **Customer** scans the QR code on the table to open the **Electronic Menu** on their device. 2. **Customer** browses the menu and selects desired items. 3. **Customer** reviews the order and adds any special requests if needed. 4. **Customer** submits the order. 5. **System** receives the order and displays it to the **Chef** for preparation. 6. **System** confirms the order to the customer and provides an estimated preparation time. 7. **Chef** begins preparing the items based on the order details received.     **Alternative Scenario:**   * Alternative Path for Waitstaff Ordering:  1. **Customer** asks **Waiter** to place an order on their behalf. 2. **Waiter** accesses the **Electronic Menu** via a staff device and selects items as requested by the **Customer**. 3. The **Waiter** submits the order through the system. 4. **System** processes the order as in the main scenario and confirms it with the **Customer**.   **Exceptions:**  If the **System** encounters an internal error during order processing, an error message is displayed, and the **Customer** can retry or request assistance from the **Waiter**.  **Relationships:**  Includes: View Menu, Apply Coupon, Pay Order  Extends: Set Order Completed  **Business Rules:**  Menu items and prices are regularly updated and synchronized with the system.  Orders must be accurately relayed to the kitchen to minimize errors.  The **System** must handle multiple orders simultaneously without delay or lag during peak hours. | | | | |
|  | | | | |

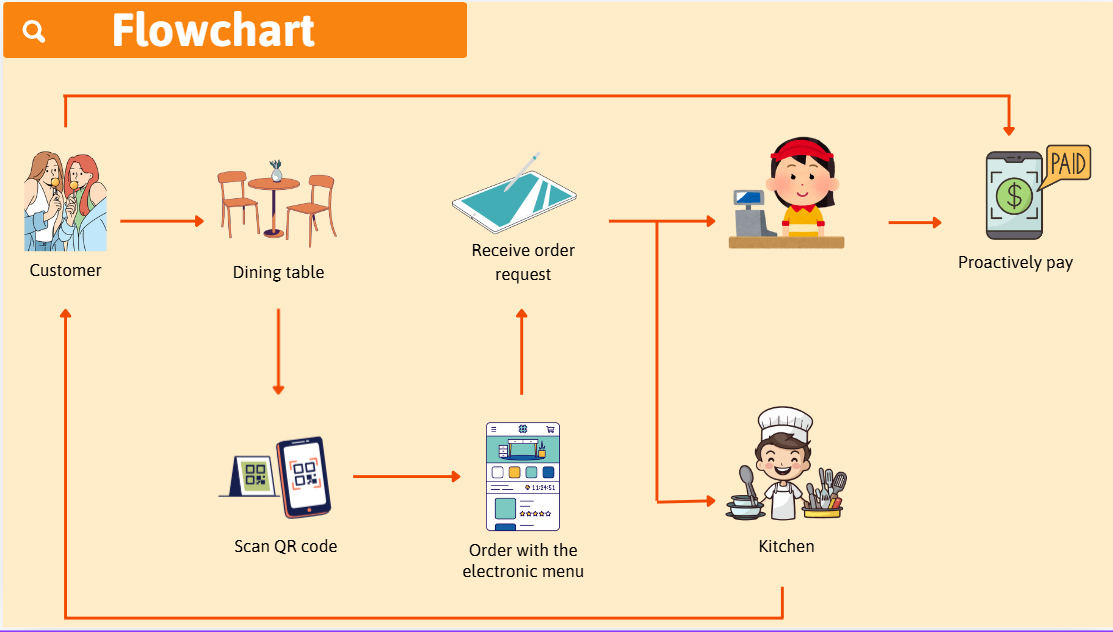
## ***Other Concerns<Optional>***

*<You can list here all other concerns about the business or the requirements if needed>*

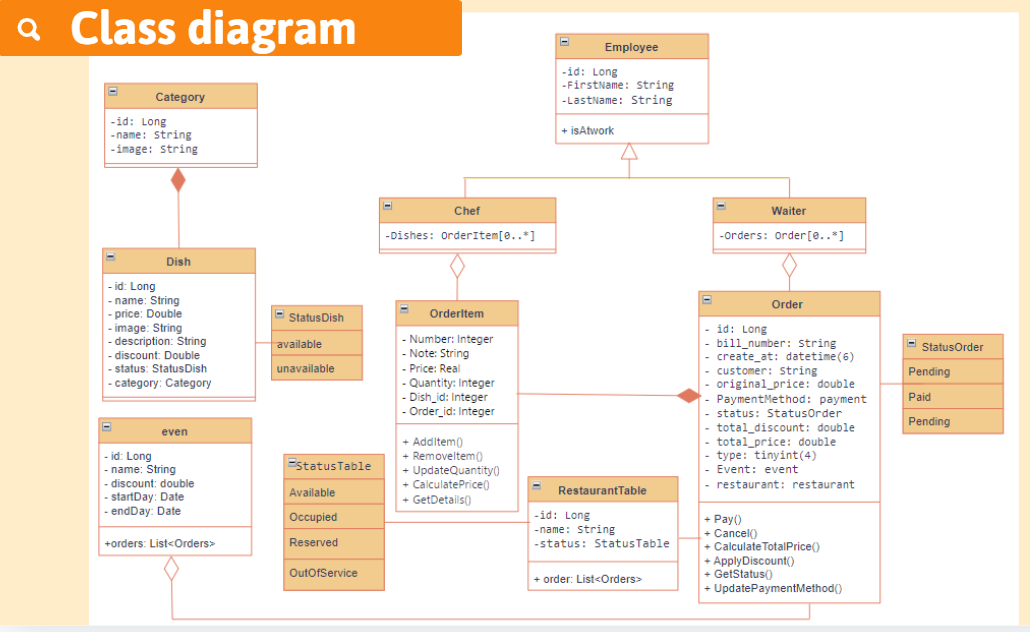
**Design**

*[This section shows the design of the system. This could be a part of the Developers Manual]*

## ***System Architecture***

**

## ***Class Diagram***



## ***Class Diagram Explanation***

The aim of the class diagram is to model an application's static vision. The only diagrams that can be explicitly converted to object-oriented languages and therefore commonly used at the time of development are class diagrams. UML diagrams, such as the operation diagram, will only give the application's sequence flow, but the class diagram is a little different. It is the most common UML in the world.

## ***Sequence Diagram (Optional)***

*<for important and complex interactions, protocols or algorithms, sequence diagrams should be drawn for clearing the details and supporting the system implementation. This section is optional>*

## ***Collaboration Diagram (Optional)***

*<for important and complex interactions, collaboration diagrams should be drawn for clearing the details and supporting the system implementation. This section is optional>*

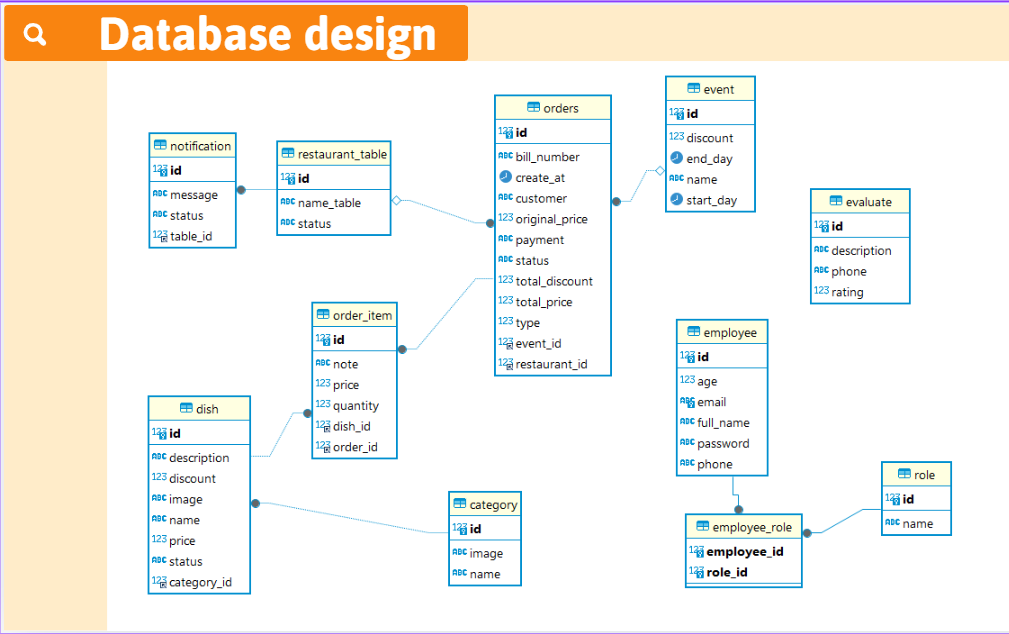
## ***State Diagram (Optional)***

*<put all state diagrams here>*

## ***Entity Relationship Diagram***

*<Provide the ERD Diagram for the system here. If your team uses file or in-memory storage facility instead of database, replace this section by ‘Data Structures’>*

## ***Database Design***

**

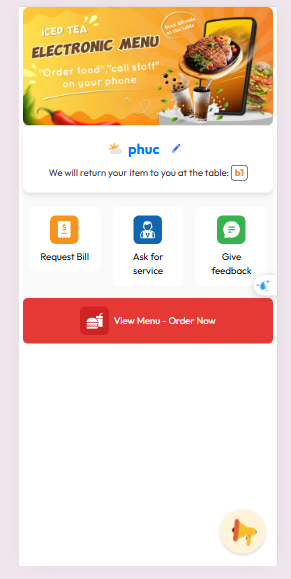
## ***Algorithms (optional)***

*<Provide the detailed description about algorithms used in the system. You can use Flow Chart or Activity Diagram to represent algorithms. Focus on the important and complex algorithms>*

## ***Others (optional)***

*<Any design concerns or diagrams can be put here>*

# **System Prototype**

**

# **Management and Project Planning**

## ***Management Approach***

Our team opted for a hybrid management approach, combining elements of self-management with a designated team lead. This structure allows for both individual ownership and collaborative decision-making.

Task Delegation:

● We leverage a collaborative task management tool (e.g., Trello, Asana) where project tasks are listed and clearly defined.

● Team members can self-select tasks based on their strengths and availability.

● The team lead facilitates discussions and ensures a balanced workload.

Team Meetings:

● We conduct regular team meetings (e.g., weekly or bi-weekly) depending on project needs.

● Meetings focus on:

● Project updates: Each member provides a brief progress report on their assigned tasks.

● Discussion and problem-solving: We address any challenges or roadblocks encountered and collaboratively brainstorm solutions.

● Decision-making: For crucial choices, the team lead facilitates a discussion to reach a consensus.

Communication:

● We maintain open communication channels through a designated team chat platform (e.g., Slack, Discord) for ongoing discussions and quick updates.

● The team lead is available for individual consultations and to provide guidance as needed.

This hybrid approach fosters a sense of ownership and accountability among team members while ensuring clear direction and support from the team lead.

## ***Project Plan***

[Electronic Menu](https://docs.google.com/spreadsheets/d/14mf8kZTfxHESIyjiicKJfUhN1ltiiTzteKwXMOONDJE/edit?gid=748029736#gid=748029736)

## ***Task Sheet***

*<Write down the tasks in Task Sheet-compatible format, this Task Sheet works as the activity report of the project or the plan of the project (not recommended); see eProject Guide for detailed Task Sheet>*

## ***Meeting Minutes (Optional)***

*<Put all minutes of your team meetings here*

# 

# **Checklists**

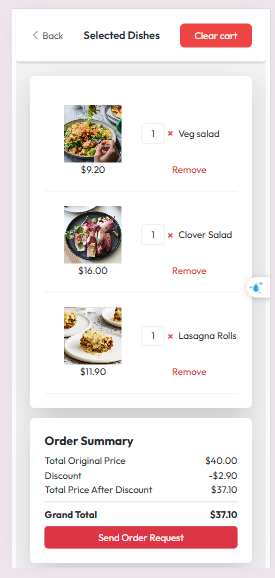
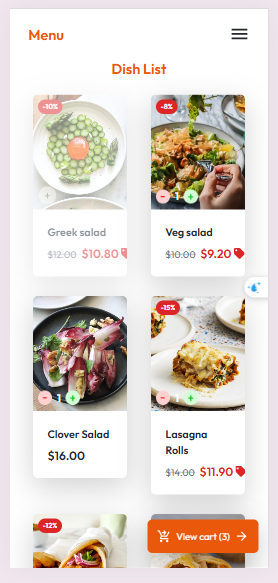
## ***Check List of Validation***

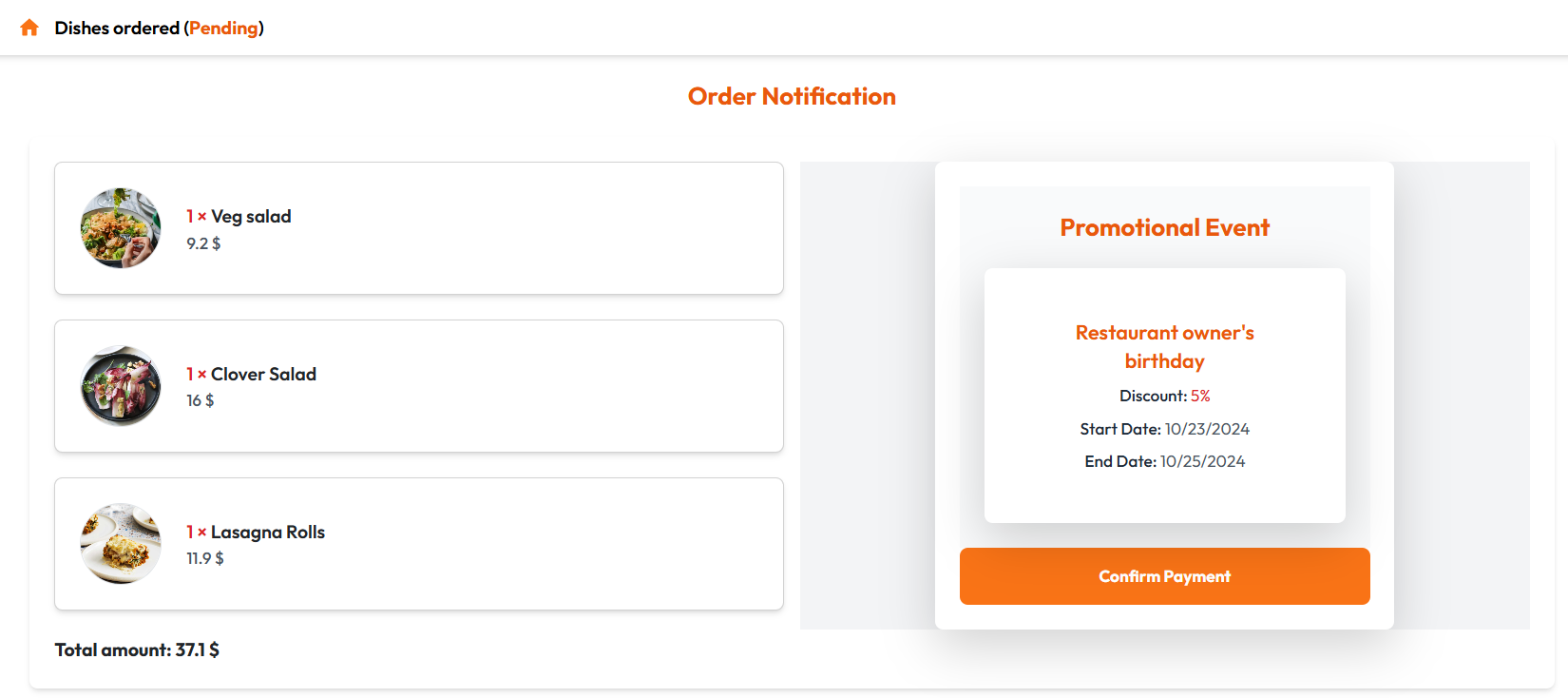
*< Put the checklist here; describe how it is used and the resulted checklist>*

## ***Submission Checklist***

*< Put the checklist here; describe how it is used and the resulted checklist>*

# **Screenshots**

**

**

# **Coding Convention**

*<Provide the coding convention for your team. If you simply want to use the existing code standard(s) such as ‘Java Code Convention’, you can refer to it\them by name or URL>*

# **Other Concerns<Optional>**

*<If you have any other information you want to add to this document, place it here. This could include thoughts on the eProject, improvements, etc.>*

# **Appendix**

## ***Glossary [Optional]***

*<Place all definitions or abbreviation used in this document >*

## ***References [Optional]***

*<Place all referenced materials used in this document >*

## ***Others<Optional>***