TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &

TRUYỀN THÔNG VIỆT HÀN

**Khoa Khoa Học Máy Tính**



ĐỒ ÁN CƠ SỞ 4

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ QUÁN ĂN THEO MÔ HÌNH CLIENT-SERVER**

Sinh viên thực hiện: **Nguyễn Hữu Thức**

Lớp: **21SE1**

Giảng viên hướng dẫn: **TS.Trần Văn Đại**

Đà Nẵng, tháng 12 năm 2023

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &

TRUYỀN THÔNG VIỆT HÀN

**Khoa Khoa Học Máy Tính**



ĐỒ ÁN CƠ SỞ 4

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ QUÁN ĂN THEO MÔ HÌNH CLIENT-SERVER**

Sinh viên: **Nguyễn Hữu Thức - 21IT109**

Giảng viên hướng dẫn: **TS. Trần Văn Đại**

Đà Nẵng, tháng 12 năm 2023

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

# 

# 

# 

# 

# 

# 

# 

# 

LỜI CẢM ƠN

Trong quá trình học tập và thực hiện đề tài, em luôn được sự quan tâm, giúp đỡ, hướng dẫn tận tình của các thầy cô trong khoa. Các thầy cô đã giảng dạy cho chúng em những kiến thức về chuyên ngành Công nghệ thông tin, để từ đó áp dụng những kiến thức đó vào thực tiễn. Với tình cảm chân thành và sâu sắc, cho phép em bày tỏ lòng biết ơn sâu sắc đến các thầy cô đã giúp đỡ để đồ án đạt được kết quả tốt đẹp.

Đặc biệt, em xin gửi lời cảm ơn chân thành nhất đến giảng viên - TS. Trần Văn Đại đã trực tiếp hướng dẫn và giúp đỡ tận tình để tạo điều kiện cho em hoàn thành đề tài này.

Với điều kiện kinh nghiệm cũng như thời gian hạn chế , đề tài này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến chân thành của các thầy cô để em có điều kiện bổ sung kinh nghiêm quý báu, nâng cao ý thức của mình, phục vụ tốt hơn công việc thực tế sau này.

Em xin chân thành cảm ơn!

*Sinh viên,*

Nguyễn Hữu Thức

MỤC LỤC

[DANH MỤC HÌNH VẼ VIII](#_Toc21256)

[Danh mục bảng IX](#_Toc32559)

[MỞ ĐẦU X](#_Toc4504)

[1. 1. Giới thiệu X](#_Toc23599)

[2. Mục tiêu của đề tài X](#_Toc25505)

[3. Nội dung và kế hoạch thực hiện X](#_Toc4505)

[4. Bố cục báo cáo XI](#_Toc3317)

[Chương 1. TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc21638)

[1. Lý thuyết môn học 1](#_Toc13459)

[1.1. Lập trình Java 1](#_Toc1308)

[1.1.1. Giới thiệu về Java 1](#_Toc17060)

[1.1.2. Giới thiệu về JavaFX 2](#_Toc14607)

[1.1.3. Ứng dụng trong đề tài 2](#_Toc21367)

[1.2. Lập trình mạng 3](#_Toc11065)

[1.2.1. Giao thức TCP 3](#_Toc3891)

[1.2.2. Socket 4](#_Toc16625)

[1.2.3. Ứng dụng trong đề tài 4](#_Toc29794)

[1.3. Cơ sở dữ liệu 5](#_Toc6914)

[1.3.1. Hệ quản lý cơ sở dữ liệu 5](#_Toc21371)

[1.3.2. Ngôn ngữ truy vấn SQL 5](#_Toc6002)

[1.3.3. Ứng dụng trong đề tài 5](#_Toc27671)

[2. Công cụ sử dụng 6](#_Toc4734)

[2.1. Visual Studio Code 6](#_Toc24693)

[2.2. SceneBuilder 7](#_Toc26741)

[2.3. Maven 8](#_Toc6778)

[2.4. Mysql 9](#_Toc1871)

[3. Thư viện bổ sung 10](#_Toc21840)

[3.1. AtlantaFX 10](#_Toc5696)

[Chương 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 11](#_Toc22856)

[1. Phân tích Use case 11](#_Toc10898)

[1.1. Danh sách Actor 11](#_Toc9152)

[1.2. Danh sách use case 12](#_Toc31875)

[1.3. Biểu đồ use case 13](#_Toc17158)

[2. Đặc tả use case 13](#_Toc25607)

[2.1. Use case đăng nhập 13](#_Toc11928)

[2.2. Use case đăng ký 14](#_Toc25750)

[2.3. Use case xem menu 14](#_Toc26167)

[2.4. Use case gọi món 14](#_Toc2162)

[2.5. Use case xem hóa đơn 15](#_Toc13979)

[2.6. Use case xem danh sách chờ món 15](#_Toc3843)

[2.7. Use case thêm món ăn 16](#_Toc14546)

[2.8. Use case sửa món ăn 16](#_Toc20289)

[2.9. Use case xóa món ăn 16](#_Toc2447)

[2.10. Use case cập nhật trạng thái món ăn được gọi 17](#_Toc30276)

[3. Sơ đồ hoạt động 18](#_Toc24563)

[3.1. Hoạt động đăng nhập 18](#_Toc19310)

[3.2. Hoạt động đăng ký 19](#_Toc10132)

[3.3. Hoạt động xem menu 20](#_Toc11982)

[3.4. Hoạt động gọi món 21](#_Toc8145)

[3.5. Hoạt động xem hóa đơn 22](#_Toc27559)

[3.6. Hoạt động xem danh sách chờ món 22](#_Toc21172)

[3.7. Hoạt động thêm món ăn 23](#_Toc2126)

[3.8. Hoạt động sửa món ăn 24](#_Toc2032)

[3.9. Hoạt động xóa món ăn 25](#_Toc17837)

[3.10. Hoạt động cập nhật trạng thái món ăn được gọi 26](#_Toc15547)

[Chương 3. XÂY DỰNG ỨNG DỤNG 27](#_Toc21554)

[1. Cấu hình máy chủ 27](#_Toc12184)

[2. Giao diện ứng dụng 27](#_Toc12538)

[2.1. Giao diện đăng nhập 27](#_Toc18249)

[2.2. Giao diện đăng ký 28](#_Toc30059)

[2.3. Giao diện menu 29](#_Toc5765)

[2.4. Giao diện xem đơn hàng 29](#_Toc29862)

[2.5. Giao diện thêm món ăn 30](#_Toc2597)

[2.6. Giao diện sửa món ăn 30](#_Toc7537)

[2.7. Giao diện cập nhật đơn hàng 31](#_Toc32224)

[3. Kết luận 31](#_Toc3211)

[TỔNG KẾT 32](#_Toc5076)

[TÀI LIỆU THAM KHẢO 33](#_Toc23176)

# **DANH MỤC HÌNH VẼ**

[Hình 1 Java 1](#_Toc26079)

[Hình 2 JavaFX 2](#_Toc25566)

[Hình 3 Giao thức TCP 3](#_Toc31928)

[Hình 4 Socket 4](#_Toc16589)

[Hình 5 Visual Studio Code 6](#_Toc9193)

[Hình 6 Scene Builder 7](#_Toc15291)

[Hình 7 Maven 8](#_Toc2824)

[Hình 8 Cấu trúc một file pom.yml đơn giản 8](#_Toc1927)

[Hình 9 MySQL 9](#_Toc14213)

[Hình 10 AtlantaFX 10](#_Toc28100)

[Hình 11 Cài đặt AtlantaFX 10](#_Toc7473)

[Hình 12 Biểu đồ use case 13](#_Toc19296)

[Hình 13 Hoạt động đăng nhập 18](#_Toc8188)

[Hình 14 Hoạt động đăng ký 19](#_Toc16515)

[Hình 15 Hoạt động xem menu 20](#_Toc25347)

[Hình 16 Hoạt động gọi món 21](#_Toc6074)

[Hình 17 Hoạt động xem hóa đơn 22](#_Toc2250)

[Hình 18 Hoạt động xem danh sách chờ món 22](#_Toc10325)

[Hình 19 Hoạt động thêm món ăn 23](#_Toc18629)

[Hình 20 Hoạt động sửa món ăn 24](#_Toc14350)

[Hình 21 Hoạt động xóa món ăn 25](#_Toc32286)

[Hình 22 Hoạt động cập nhật trạng thái món ăn 26](#_Toc8171)

[Hình 23 Giao diện đăng nhập 27](#_Toc5042)

[Hình 24 Giao diện đăng ký 28](#_Toc14437)

[Hình 25 Giao diện menu 29](#_Toc29021)

[Hình 26 Giao diện xem đơn hàng 29](#_Toc20681)

[Hình 27 Giao diện thêm món ăn 30](#_Toc20087)

[Hình 28 Giao diện sửa món ăn 30](#_Toc13791)

[Hình 29 Giao diện cập nhật đơn hàng 31](#_Toc3486)

# **DANH MỤC BẢNG**

[Bảng 1 Danh sách Actor 11](#_Toc3018)

[Bảng 2 Danh sách use case 12](#_Toc1462)

# **MỞ ĐẦU**

1. 1. Giới thiệu

Những quán ăn đang phát triển không chỉ thông qua khẩu vị ngon mắt mà còn thông qua công nghệ. Sự phổ biến của smartphone và sự thuận tiện của internet đã tạo ra một cơ hội mới cho ngành thực phẩm - cho phép quán ăn nâng cao trải nghiệm của khách hàng thông qua việc đặt món và quản lý đơn hàng trực tiếp trong quán.

Khi bạn bước vào một quán ăn hiện đại ngày nay, bạn có thể thấy người dùng không còn phải chờ đợi lâu để đặt món hoặc trả tiền. Thay vào đó, họ có thể sử dụng ứng dụng di động hoặc thiết bị tương tự để xem menu, đặt món, và thanh toán ngay tại bàn. Điều này không chỉ giúp giảm thời gian chờ đợi mà còn cải thiện hiệu suất của quán, cho phép họ xử lý đơn hàng nhanh chóng và tối ưu hóa dịch vụ.

Xây dựng ứng dụng Đặt món và quản lý đơn trong quán ăn không chỉ đem lại lợi ích cho khách hàng mà còn giúp quán ăn theo dõi đơn hàng, quản lý nguồn lực và dự báo nhu cầu một cách hiệu quả. Dự án này đại diện cho sự kết hợp giữa sáng tạo và công nghệ, biến một bữa ăn thông thường thành một trải nghiệm tiên tiến và hiện đại.

2. Mục tiêu của đề tài

Mục tiêu của đề tài nhằm tạo ra một ứng dụng giúp hỗ trợ việc tương tác giữa các nhân viên trong quán ăn và khách hàng trở nên dễ dàng hơn. Các chức năng chính bao gồm:

- Quản lý món ăn: Cho phép nhân viên tạo, chỉnh sửa và quản lý danh sách món ăn của quán.

- Gọi món: Cho phép người dùng thực hiện xem danh sách món ăn và thực hiện gọi món thông qua ứng dụng, và được cập nhật hành động dựa trên thời gian thực nâng cao tính trải nghiệm.

Giao diện người dùng: Xây dựng giao diện dễ sử dụng, cung cấp trải nghiệm ổn định.

3. Nội dung và kế hoạch thực hiện

Để thực hiện đề tài, chúng em sẽ lần lượt tiến hành các nội dung:

- Nghiên cứu tổng quan về đề tài.

- Nghiên cứu cơ sở lí thuyết của đề tài.

- Tiến hành phân tích và thiết kế

- Tiến hành xây dựng ứng dụng theo các mục tiêu đề ra

Kế hoạch thực hiện:

|  |  |
| --- | --- |
| Thời gian | Nội dung thực hiện |
| Từ 03/10/2023 đến 21/10/2023 | Giảng viên hướng dẫn và sinh viên thống nhất đề tài, nội dung và kế hoạch xây dựng đề cương đồ án. |
| Từ 25/10/2023 đến 1/11/2023 | Tìm hiểu và nghiên cứu về tổng quan và cơ sở lí thuyết của đề tài. |
| Từ 02/11/2023 đến 16/11/2023 | Phân tích thiết kế ứng dụng. |
| Từ 17/11/2022 đến 03/12/2023 | Xây dựng ứng dụng. |
| Từ 04/12/2023 đến 10 /12/2023 | Kiểm thử và hoàn thiện sản phẩm |
| Từ 11/12/2023 đến 14/12/2023 | Viết báo cáo và chuẩn bị slide |

4. Bố cục báo cáo

Sau phần *Mở đầu*, báo cáo được trình bày trong ba chương, cụ thể như sau:

Chương 1.Tổng quan về đề tài.Trong chương này, báo cáo trình bày các khái niệm, đặc điểm, các kiến thức, ngôn ngữ, công cụ sẽ sử dụng.

Chương 2. Phân tích thiết kế hệ thống. Nội dung chương bao gồm các yêu cầu, bước thiết kế để thực hiện nhằm giải quyết vấn đề.

Chương 3. Xây dựng hệ thống Chương này trình bày kết quả, chức năng cài đặt được cảu hệ thống

Cuối cùng là Kết luận, Tài liệu tham khảo và Phụ lục liên quan đến đề tài.

**Chương 1. TỔNG QUAN VỀ ĐỀ TÀI**

1. Lý thuyết môn học
   1. Lập trình Java
      1. Giới thiệu về Java



Hình 1 Java

Java là một ngôn ngữ lập trình đa nền tảng, được phát triển bởi Sun Microsystems (nay là Oracle Corporation) vào những năm 1990. Với khả năng chạy trên mọi loại hệ điều hành, Java đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới. Một số đặc điểm quan trọng của Java bao gồm:

- **Đa nền tảng (Platform Independence)**: Ứng dụng Java có thể chạy trên mọi hệ điều hành mà không cần biên dịch lại mã nguồn.

- **Tính Bảo mật Cao**: Java sử dụng mô hình bảo mật sandbox để ngăn chặn các nguy cơ bảo mật potentional.

- **Thuận Tiện cho Phát triển Ứng dụng**: Java mạnh mẽ trong Lập trình hướng đối tượng từ đó giúp việc phát triển ứng dụng trở nên thuận tiện hơn.

* + 1. Giới thiệu về JavaFX



Hình 2 JavaFX

**JavaFX** là một nền tảng phát triển ứng dụng đồ họa và giao diện người dùng (GUI) cho ngôn ngữ lập trình **Java**. Nó cung cấp các công cụ và thư viện cho việc xây dựng ứng dụng đa nền tảng với giao diện người dùng đồ họa mạnh mẽ. Dưới đây là một giới thiệu tổng quan về **JavaFX**:

- Có thể sử dụng **XML** để tạo giao diện: **JavaFX** có thể tạo giao diện người dùng sử dụng xml, ngoài ra còn có thể sử dụng **css** để khiến cho giao diện bắt mắt hơn.

- **Tương thích với nhiều nền tảng** như Window, MacOS, Linux.

* + 1. Ứng dụng trong đề tài

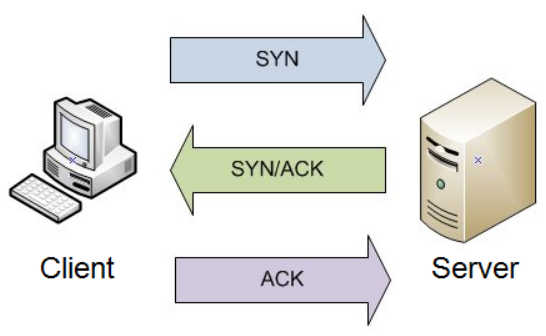
Trong ngữ cảnh của đề tài, lập trình Java sẽ đóng một vai trò quan trọng trong việc phát triển ứng dụng hoặc hệ thống. Một số ứng dụng cụ thể bao gồm:

− **Phát triển Ứng dụng Đa Nền tảng**: Sử dụng tính đa nền tảng của Java để xây dựng ứng dụng có thể chạy trên nhiều hệ điều hành khác nhau.

− **Kết nối và Giao tiếp Mạng**: Java cung cấp các thư viện mạng mạnh mẽ, cho phép xây dựng các ứng dụng có khả năng giao tiếp qua mạng.

− **Lập trình Giao diện Người dùng (GUI)**: Sử dụng JavaFX để phát triển giao diện người dùng đồ họa cho ứng dụng.

* 1. Lập trình mạng
     1. Giao thức TCP



Hình 3 Giao thức TCP

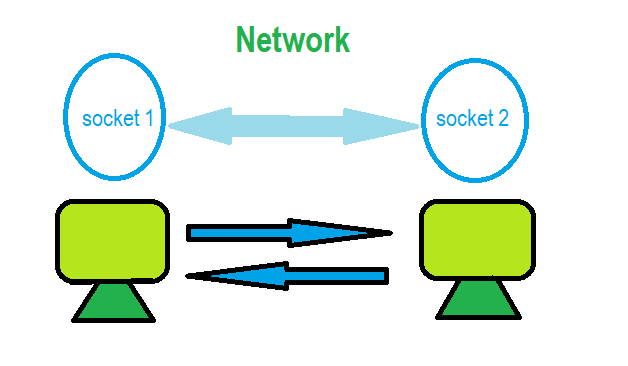
Giao thức TCP (Transmission Control Protocol) là một trong những giao thức quan trọng trong mô hình **TCP/IP**, được sử dụng để thiết lập và duy trì kết nối giữa các thiết bị trên mạng. Dưới đây là một số điểm chính về TCP:

- **Kết nối Được Thiết lập**: TCP thiết lập kết nối đường dẫn đáng tin cậy giữa nguồn và đích trước khi truyền dữ liệu.

- **Đảm Bảo Giao Tiếp Tin Cậy**: TCP cung cấp cơ chế kiểm soát lỗi và đảm bảo rằng dữ liệu được truyền đi mà **không bị mất hoặc bị lẫn lộn**.

- **Chia Dữ Liệu thành Các Gói**: Dữ liệu được chia thành các gói nhỏ để truyền đi qua mạng và được **gửi theo thứ tự đảm bảo**.

* + 1. Socket



Hình 4 Socket

**Socket** là một cơ chế trong lập trình mạng **cho phép hai máy tính truyền thông với nhau thông qua mạng**.

Dưới đây là một số điểm quan trọng về socket:

- **Giao Diện Lập Trình Ứng Dụng (API)**: Socket cung cấp API để phát triển ứng dụng có khả năng giao tiếp qua mạng.

- **Kết nối Client-Server**: Socket được sử dụng để thiết lập kết nối giữa client và server, cho phép truyền và nhận dữ liệu.

* + 1. Ứng dụng trong đề tài

Trong ngữ cảnh của đề tài, lập trình mạng sẽ chủ yếu tập trung vào việc xây dựng giao tiếp giữa các thành phần của hệ thống.

Cụ thể:

- **Giao Tiếp Giữa Client và Server**: Sử dụng socket TCP để thiết lập kết nối đáng tin cậy giữa client và server, cho phép truyền thông và trao đổi dữ liệu.

- **Quản lý Kết nối và Truyền Dữ liệu**: Xử lý việc mở, đóng, và duy trì kết nối, cũng như đảm bảo dữ liệu được truyền đi mà không bị mất.

- **Giao thức TCP**: Đảm bảo độ tin cậy của giao tiếp.

* 1. Cơ sở dữ liệu
     1. Hệ quản lý cơ sở dữ liệu

**Hệ quản lý Cơ sở Dữ liệu** là một phần quan trọng trong việc lưu trữ, quản lý và truy xuất dữ liệu.

Dưới đây là một số điểm quan trọng về DBMS:

- **Định Nghĩa**: Hệ quản lý Cơ sở Dữ liệu là một hệ thống phần mềm dùng để tạo, quản lý, và cập nhật cơ sở dữ liệu.

- **Quản lý Dữ Liệu**: DBMS quản lý dữ liệu thông qua các thao tác như thêm, sửa, xóa, và truy xuất dữ liệu.

- **Bảo Mật và Quyền Truy Cập**: Cung cấp các cơ chế bảo mật để đảm bảo chỉ những người được phép có quyền truy cập vào dữ liệu.

* + 1. Ngôn ngữ truy vấn SQL

**SQL** (**Structured Query Language**) là một ngôn ngữ được sử dụng để tương tác với cơ sở dữ liệu.

Dưới đây là một số điểm chính về **SQL**:

- **Câu Lệnh SQL Cơ Bản**: **SQL** bao gồm các câu lệnh như SELECT, INSERT, UPDATE, DELETE để thực hiện các thao tác trên dữ liệu.

- **Truy Vấn Dữ Liệu**: **SQL** cho phép truy vấn dữ liệu từ cơ sở dữ liệu dựa trên các điều kiện và mối quan hệ.

- **Quản Lý Cơ Sở Dữ Liệu**: **SQL** cũng sử dụng để tạo, cập nhật và quản lý cấu trúc của cơ sở dữ liệu.

* + 1. Ứng dụng trong đề tài

Trong phạm vi của đề tài, cơ sở dữ liệu sẽ đóng vai trò quan trọng trong việc lưu trữ và quản lý thông tin.

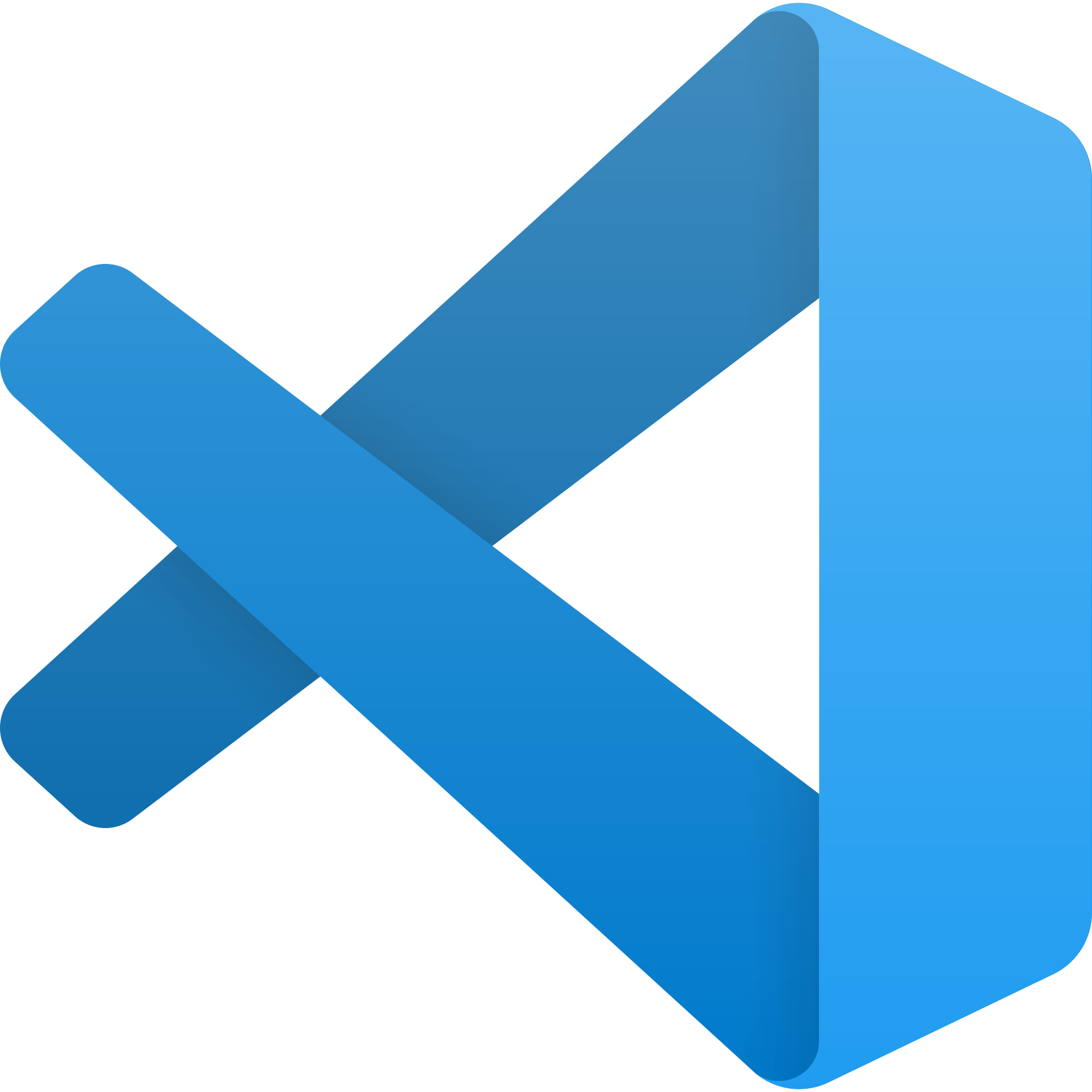
Cụ thể:

- **Thiết Kế Cơ Sở Dữ Liệu**: Xây dựng cấu trúc cơ sở dữ liệu phản ánh các yêu cầu của hệ thống và đảm bảo hiệu suất tối ưu.

- **Tương Tác với DBMS**: Sử dụng SQL để truy xuất và cập nhật dữ liệu từ cơ sở dữ liệu, bao gồm các thao tác như truy vấn thông tin và lưu trữ dữ liệu mới.

- **Bảo Mật Dữ Liệu**: Triển khai các biện pháp bảo mật cơ sở dữ liệu để đảm bảo rằng chỉ những người có quyền mới có thể truy cập và thay đổi dữ liệu.

1. Công cụ sử dụng
   1. Visual Studio Code



Hình 5 Visual Studio Code

**Visual Studio Code** là một trình soạn thảo mã nguồn được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có chức năng nổi bật cú pháp (syntax highlighting), tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, người dùng có thể thay đổi theme, phím tắt, và các tùy chọn khác.

Visual Studio Code được dựa trên Electron, một nền tảng được sử dụng để triển khai các ứng dụng Node.js máy tính cá nhân chạy trên động cơ bố trí Blink. Mặc dù nó sử dụng nền tảng Electron nhưng phần mềm này không phải là một bản khác của Atom, nó thực ra được dựa trên trình biên tập của Visual Studio Online (tên mã là "Monaco").

Trong cuộc khảo sát vào năm 2022 trên Stack Overflow, Visual Studio Code được xếp hạng là trình biên tập mã phổ biến nhất, với 74.48% của 71,010 người trả lời tuyên bố sử dụng nó.

Với những tính năng vượt trội chúng em đã chọn **Visual Studio Code** để tiến hành hoàn thiện đề tài.

* 1. SceneBuilder



Hình 6 Scene Builder

**SceneBuilder** là một công cụ đồ họa được sử dụng để thiết kế giao diện người dùng (GUI) cho ứng dụng **JavaFX**. Được cung cấp bởi Oracle, SceneBuilder giúp người phát triển tạo và quản lý các thành phần của giao diện người dùng một cách trực quan, thường là thông qua việc kéo và thả các phần tử trực tiếp trên giao diện.

Các tính năng chính của SceneBuilder bao gồm:

**Giao Diện Đồ Họa Trực Quan**: SceneBuilder cung cấp một môi trường đồ họa và trực quan để thiết kế giao diện người dùng mà không cần viết mã.

**Kéo và Thả Phần Tử**: Người phát triển có thể kéo và thả các phần tử như nút, trường văn bản, bảng và các thành phần khác để tạo giao diện người dùng theo ý muốn.

**Quản Lý Cấu Trúc FXML**: SceneBuilder tạo và quản lý file FXML, là một ngôn ngữ đánh dấu XML được sử dụng để định nghĩa giao diện người dùng trong JavaFX.

**Xem Trước Giao Diện**: Người phát triển có thể xem trước giao diện người dùng ngay trong SceneBuilder để kiểm tra trước cách nó sẽ hiển thị khi chạy ứng dụng.

**Tích Hợp với Môi Trường Phát Triển**: SceneBuilder tích hợp chặt chẽ với các môi trường phát triển như IntelliJ IDEA và NetBeans, giúp dễ dàng mở và chỉnh sửa các file FXML từ trong môi trường phát triển.

**Hỗ Trợ CSS Styling**: Cho phép người phát triển áp dụng và xem trước các hiệu ứng CSS trực tiếp từ giao diện của SceneBuilder.

SceneBuilder giúp tăng cường quá trình phát triển JavaFX bằng cách làm cho việc thiết kế giao diện trở nên đơn giản và trực quan hơn.

* 1. Maven



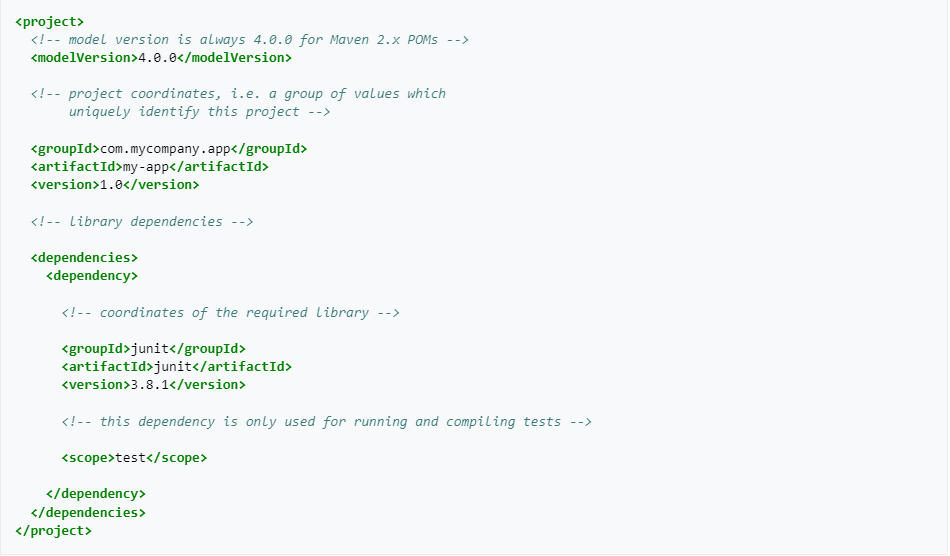
Hình 7 Maven

**Maven** là công cụ quản lý và thiết lập tự động 1 dự án phần mềm. Chủ yếu dùng cho các lập trình viên **Java**, nhưng nó cũng có thể được dùng để xây dựng và quản lý các dự án dùng C#, Ruby, Scala hay ngôn ngữ khác.

**Maven** dùng khái niệm **Project Object Model (POM)** để mô tả việc build project, các thành phần phụ thuộc và các module. Nó định nghĩa trước các target cho việc khai báo task, trình biên dịch, đóng gói và thứ tự hoạt động để mọi việc diễn ra tốt nhất.

Maven tải các thư viện, plug-in từ 1 hoặc nhiều repositories. Maven cung cấp việc hỗ trợ tải file từ repository trung tâm của **Maven 2** và các repositories khác. Cũng có thể upload các sản phẩm lên repository sau khi đã build xong.

Maven được xây dựng dựa trên kiến trúc plugin - cho phép sử dụng bất kỳ ứng dụng điều khiển nào thông qua đầu vào tiêu chuẩn. Về mặt lý thuyết, điều này cho phép bất kỳ ai cũng có thể viết các plugin để giao tiếp với các build tool (trình biên dịch, unit test tool, etc) cho bất kỳ ngôn ngữ nào khác. Nhưng trong thực tế, hỗ trợ và sử dụng cho các ngôn ngữ khác đã bị thu hẹp.



Hình 8 Cấu trúc một file pom.yml đơn giản

* 1. Mysql

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến được sử dụng rộng rãi trong cả môi trường doanh nghiệp và cộng đồng phần mềm mã nguồn mở. Được phát triển, duy trì và hỗ trợ bởi Oracle Corporation, MySQL đã trở thành một trong những công cụ quản lý dữ liệu hàng đầu trên thị trường.



Hình 9 MySQL

Với sự linh hoạt, hiệu suất cao và khả năng mở rộng, MySQL đã chứng minh được tính ưu việt của mình trong việc quản lý dữ liệu từ những ứng dụng web cá nhân đến các hệ thống doanh nghiệp quy mô lớn.

Đặc Điểm Quan Trọng của MySQL:

**Tính Ứng Dụng Rộng Rãi**: MySQL là một giải pháp cơ sở dữ liệu linh hoạt và mạnh mẽ, được sử dụng cho các ứng dụng web, di động, nhúng, cũng như trong lĩnh vực của dữ liệu doanh nghiệp và các ứng dụng máy chủ.

**Tính Năng Mở Rộng**: Với khả năng mở rộng linh hoạt, MySQL có thể mở rộng dự án từ những hệ thống nhỏ đến các cấu trúc phức tạp có tải lớn mà vẫn duy trì hiệu suất ổn định.

**Hiệu Suất Cao**: MySQL được tối ưu hóa để cung cấp hiệu suất cao và xử lý dữ liệu nhanh chóng. Cơ sở dữ liệu có thể được tinh chỉnh để đáp ứng các yêu cầu khắt khe về tải công việc.

**Cộng Đồng Lớn và Hỗ Trợ**: MySQL có một cộng đồng người dùng rộng lớn, cung cấp tài liệu, diễn đàn, và hỗ trợ kỹ thuật phong phú từ các chuyên gia trên khắp thế giới.

**Bảo Mật và Đáng Tin Cậy**: MySQL cung cấp các cơ chế bảo mật mạnh mẽ để bảo vệ dữ liệu. Nó cũng có các tính năng sao lưu và khôi phục để đảm bảo an toàn và tin cậy cho cơ sở dữ liệu.

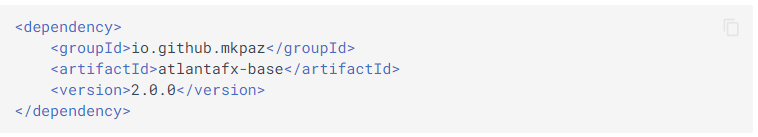
1. Thư viện bổ sung
   1. AtlantaFX



Hình 10 AtlantaFX

Là một thư viện mã nguồn được chia sẻ bởi mkpaz mở cung cấp một giao diện dễ nhìn, sinh động kèm theo nhiều Component cho JavaFX giúp cho việc tạo giao diện trở nên dễ dàng hơn.

Với khả năng hỗ trợ cho Maven cũng như SceneBuilder thì đây là một lựa chọn tốt để kết hợp với những công cụ ở trên giúp nâng cao tiến độ hoàn thành đề tài.





Hình 11 Cài đặt AtlantaFX

# **Chương 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

1. Phân tích Use case
   1. Danh sách Actor

|  |  |
| --- | --- |
| Tên Actor | Ý nghĩa |
| User | Người dùng của hệ thống |
| Staff | Nhân viên cửa hàng |
| Admin | Quản lý cửa hàng |

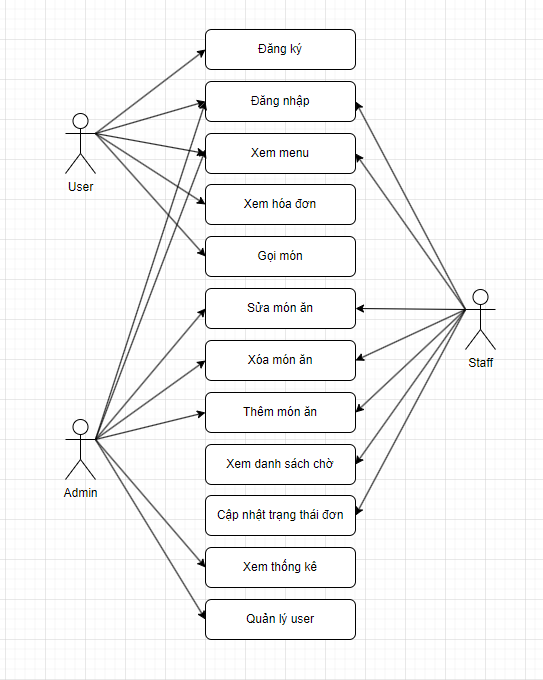
Bảng 1 Danh sách Actor

* 1. Danh sách use case

|  |  |
| --- | --- |
| Tên use case | Ý nghĩa |
| Đăng nhập | Cho phép người dùng đăng nhập để sử dụng hệ thống |
| Đăng ký | Cho phép người dùng đăng ký để sử dụng hệ thống |
| Xem menu | Cho phép người dùng xem danh sách món ăn của quán |
| Gọi món | Cho phép người dùng gửi yêu cầu gọi món đến nhân viên |
| Xem hóa đơn | Cho phép người dùng xem danh sách các món ăn đã gọi |
| Xem danh sách chờ món | Cho nhép nhân viên xem danh sách các món ăn mà người dùng gọi |
| Thêm món ăn | Cho phép nhân viên thêm món ăn vào menu |
| Sửa món ăn | Cho phép nhân viên sửa thông tin món ăn |
| Xóa món ăn | Cho phép nhân viên xóa món ăn khỏi menu |
| Cập nhật trạng thái món ăn được gọi | Cho phép nhân viên cập nhật lại trạng thái của món ăn mà người dùng gọi |
| Xem thống kê | Cho phép quản lý xem thông tin doanh thu, ... |
| Quản lý user | Cho phép quản lý quản lý người dùng sử dụng hệ thống |

Bảng 2 Danh sách use case

* 1. Biểu đồ use case

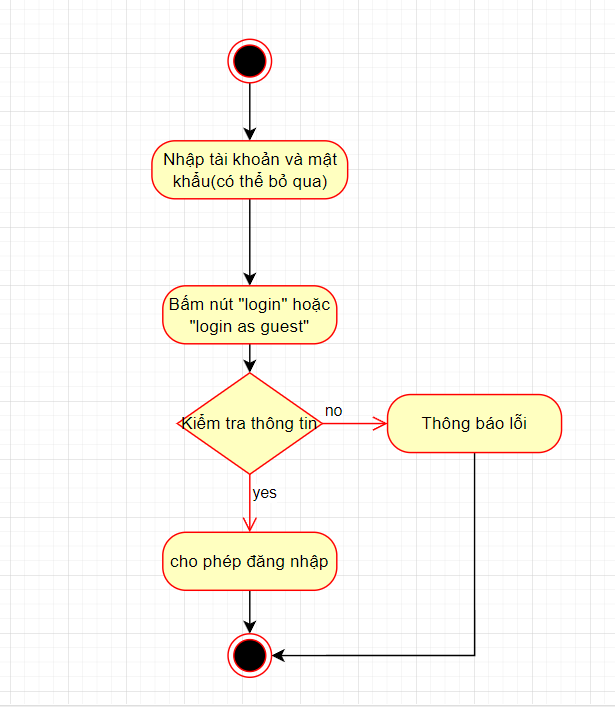


Hình 12 Biểu đồ use case

1. Đặc tả use case
   1. Use case đăng nhập

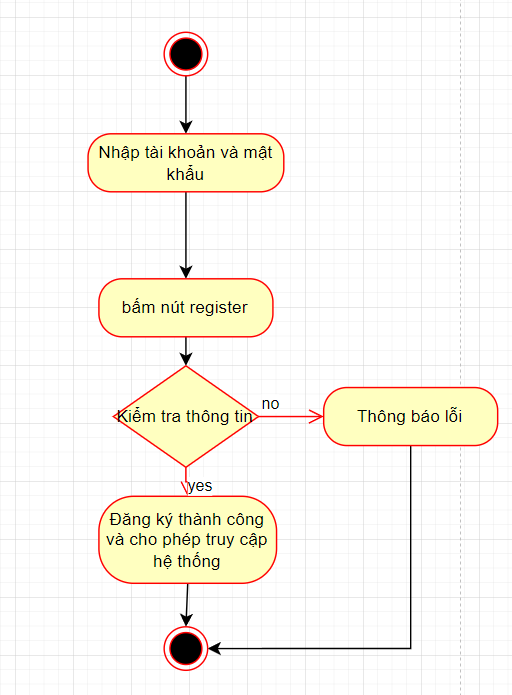
* Mô tả: Use case này cho phép người dùng sử dụng thông tin đăng nhập (tài khoản và mật khẩu) để truy cập hệ thống.
* Các bước thực hiện
  + Người dùng nhập tên tài khoản và mật khẩu.
  + Người dùng chọn nút đăng nhập.
  + Hệ thống kiểm tra thông tin, cho phép truy cập nếu hợp lệ.
* Các kịch bản
  + Kịch bản chính: người dùng điền đúng thông tin và được phép truy cập hệ thống.
  + Kịch bản lỗi: Người dùng nhập sai thông tin và không được truy cập.
  1. Use case đăng ký
* Mô tả: Use case này cho phép người dùng đăng ký tài khoản mới để truy cập vào hệ thống
* Các bước thực hiện:
  + Người dùng bấm vào nút đăng ký.
  + Hệ thống hiển thị form đăng ký.
  + Người dùng nhập thông tin vào form đăng ký.
  + Người dùng chọn nút đăng ký.
  + Hệ thống kiểm tra thông tin và tạo tài khoản mới nếu hợp lệ.
* Các kịch bản:
  + Kịch bản chính: Người dùng điền thông tin đăng ký đúng và tài khoản mới được tạo.
  + Kịch bản lỗi: người dùng nhập sai thông tin, tài khoản không được tạo.
  1. Use case xem menu
* Mô tả: Use case cho phép người dùng xem danh sách món ăn trong quán
* Các bước thực hiện:
  + Người dùng chọn tab Order trên thanh Tabs.
  + Hệ thống hiển thị danh sách món ăn trên giao diện .
* Các kịch bản:
  + Kịch bản chính: Hệ thống hiển thị danh sách món ăn trên giao diện.
  1. Use case gọi món
* Mô tả: Use case cho phép người đùng chọn món và tiến hành gửi yêu cầu đến nhân viên.
* Các bước thực hiện:
  + Người dùng chọn tab Order trên thanh Tabs.
  + Hệ thống hiển thị danh sách món ăn trên giao diện.
  + Người dùng chọn món ăn trên giao diện, thay đổi số lượng và nhấn vào nút “Add”.
  + Hệ thống tiến hành gửi yêu cầu đến server.
  + Server xử lý và trả về đơn hàng hiện tại.
* Các kịch bản:
  + Kịch bản chính: Người dùng gửi yêu cầu thành công và món được thêm vào danh sách chờ.
  + Kịch bản lỗi: Có lỗi xảy ra với thông tin món và món không thêm vào danh sách chờ.
  1. Use case xem hóa đơn
* Mô tả: Use case cho phép người dùng xem danh sách món ăn trên đơn hàng hiện tại.
* Các bước thực hiện:
  + Người dùng chọn tab “Bill” trên thanh Tabs.
  + Hệ thống hiển thị danh sách món ăn và các thông tin trên bảng.
* Các kịch bản:
  + Kịch bản chính: Hiển thị danh sách món ăn đã gọi lên giao diện người dùng.
  1. Use case xem danh sách chờ món
* Mô tả: Use case cho phép nhân viên xem danh sách các món ăn được gọi trong danh sách chờ.
* Các bước thực hiện:
  + Nhân viên chọn tab “Bill” trên thanh Tabs.
  + Hệ thống hiển thị danh sách món ăn kèm theo id của người gọi món để dễ dàng phục vụ.
* Các kịch bản:
  + Kịch bản chính: Hiển thị danh sách những đơn hàng đã gọi lên giao diện.
  1. Use case thêm món ăn
* Mô tả: Use case cho phép Nhân viên hoặc Quản lý có thể thêm món ăn vào menu.
* Các bước thực hiện:
  + Nhân viên hoặc Quản lý chọn tab “Add Item” trên thanh Tabs.
  + Tiến hành nhập thông tin của món ăn.
  + Nhấn nút “Add” để gửi yêu cầu đến Server.
  + Server xử lý và trả về kết quả.
* Các kịch bản:
  + Kịch bản chính: Server trả về thành công, menu được cập nhật ở tất cả các người dùng.
  + Kịch bản lỗi: Server trả về thất bại và báo thông tin lỗi cho Nhân viên hoặc Quản lý thực hiện yêu cầu.
  1. Use case sửa món ăn
* Mô tả: Use case cho phép Nhân viên hoặc Quản lý sửa thông tin món ăn có trong menu.
* Các bước thực hiện:
  + Nhân viên hoặc Quản lý chọn tab “Edit Item” trên thanh Tabs.
  + Chọn Món ăn cần sửa.
  + Thực hiện sửa thông tin món ăn.
  + Nhấn nút “Update” để gửi yêu cầu đến Server.
  + Server xử lý yêu cầu và trả về kết quả.
* Các kịch bản:
  + Kịch bản chính: Thành công, cập nhật lại toàn bộ menu ở các người dùng.
  + Kịch bản lỗi: Lỗi, thông báo lỗi cho Nhân viên Hoặc Quản lý đang thực hiện yêu cầu.
  1. Use case xóa món ăn
* Mô tả: Use case cho phép nhân viên hoặc quản lý xóa món ăn khỏi menu.
* Các bước thực hiện:
  + Nhân viên hoặc quản lý chọn tab “Edit Item” trên thanh Tabs.
  + Chọn Món ăn cần xóa.
  + Nhấn nút “Xóa” để gửi yêu cầu đến Server.
  + Server xử lý yêu cầu và trả về kết quả.
* Các kịch bản:
  + Kịch bản chính: Thành công, cập nhật lại toàn bộ menu ở các người dùng.
  + Kịch bản lỗi: Lỗi, thông báo lỗi cho Nhân viên hoặc quản lý đang thực hiện yêu cầu.
  1. Use case cập nhật trạng thái món ăn được gọi
* Mô tả: Use case cho phép nhân viên thực hiện thay đổi trạng thái món ăn đang trong danh sách chờ thành “Done” hoặc “Cancle”.
* Các bước thực hiện:
  + Nhân viên chọn tab “Bill” trên thanh Tabs.
  + Nhân viên chọn đơn hàng cần cập nhật.
  + Nhấn nút “Done” nếu như đơn hàng đã được hoàn thành và giao đến người dùng, nhấn nút “Cancle” nếu như đơn hàng có vấn đề phát sinh và gửi yêu cầu đến Server.
  + Server xử lý và trả về kết quả.
* Các kịch bản:
  + Kịch bản chính: Thành công, tiến hành cập nhật trạng thái tại người dùng đã đặt đơn và toàn bộ nhân viên.
  + Kịch bản lỗi: Lỗi, thông báo lỗi cho nhân viên thực hiện yêu cầu.

1. Sơ đồ hoạt động
   1. Hoạt động đăng nhập



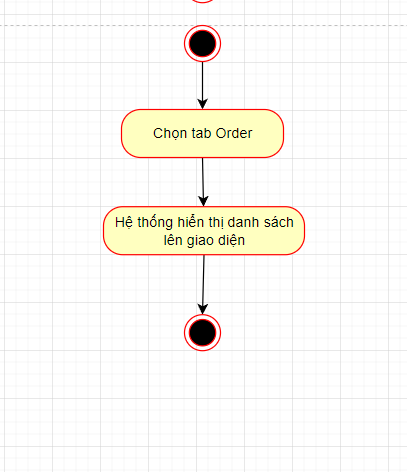
Hình 13 Hoạt động đăng nhập

* 1. Hoạt động đăng ký



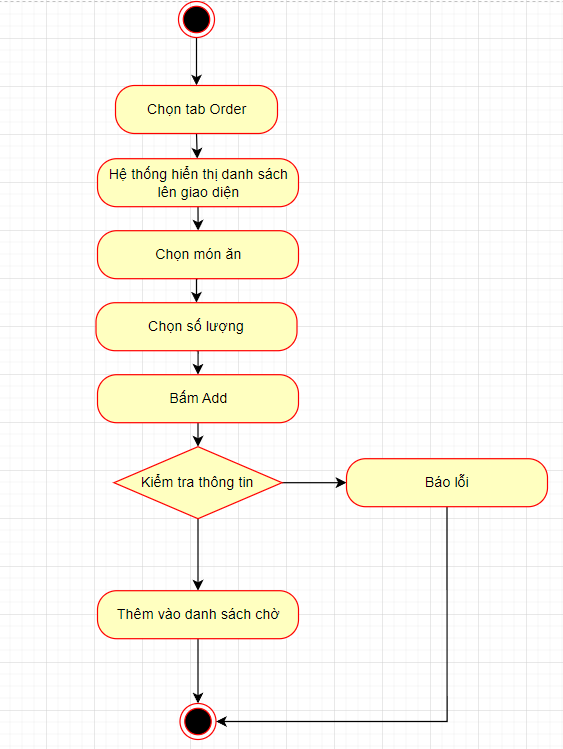
Hình 14 Hoạt động đăng ký

* 1. Hoạt động xem menu



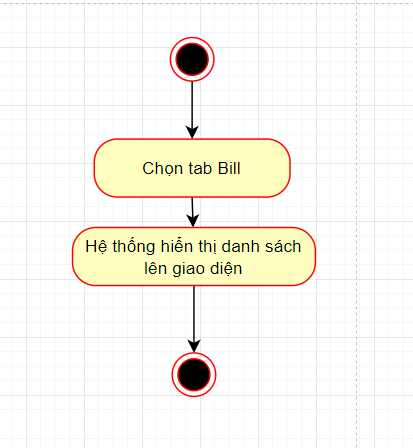
Hình 15 Hoạt động xem menu

* 1. Hoạt động gọi món



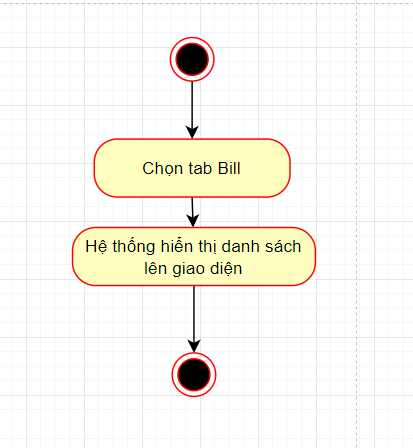
Hình 16 Hoạt động gọi món

* 1. Hoạt động xem hóa đơn



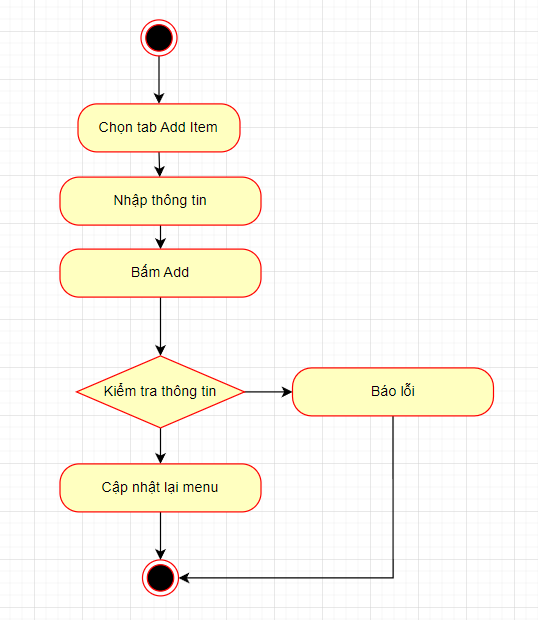
Hình 17 Hoạt động xem hóa đơn

* 1. Hoạt động xem danh sách chờ món



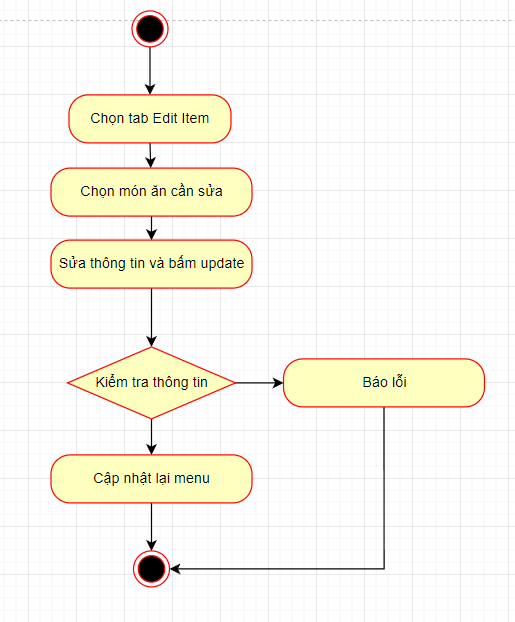
Hình 18 Hoạt động xem danh sách chờ món

* 1. Hoạt động thêm món ăn



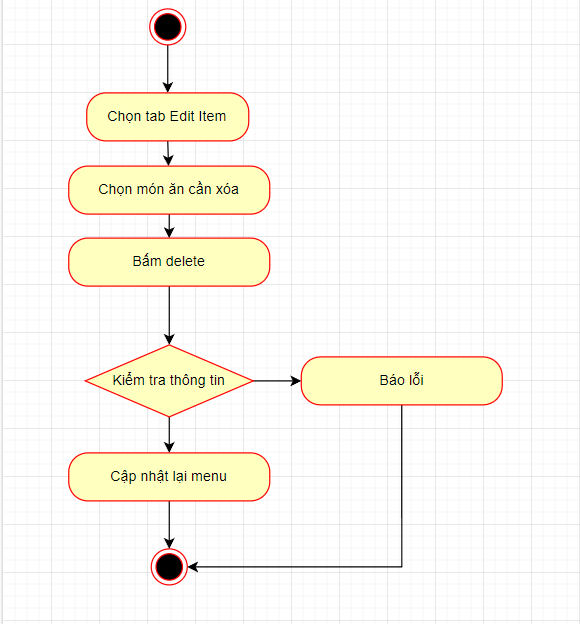
Hình 19 Hoạt động thêm món ăn

* 1. Hoạt động sửa món ăn



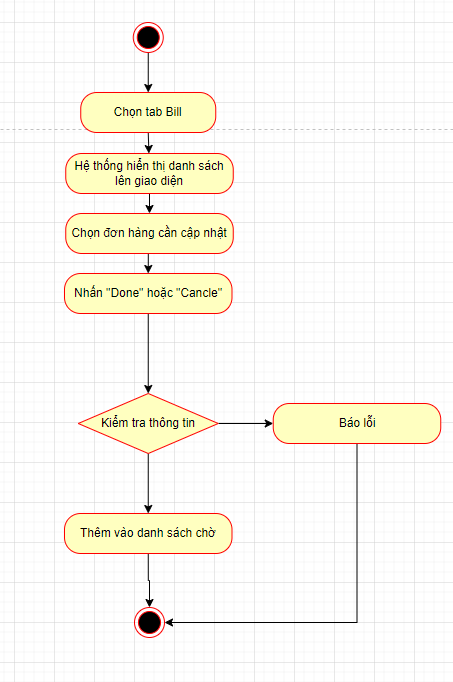
Hình 20 Hoạt động sửa món ăn

* 1. Hoạt động xóa món ăn



Hình 21 Hoạt động xóa món ăn

* 1. Hoạt động cập nhật trạng thái món ăn được gọi



Hình 22 Hoạt động cập nhật trạng thái món ăn

# **Chương 3. XÂY DỰNG ỨNG DỤNG**

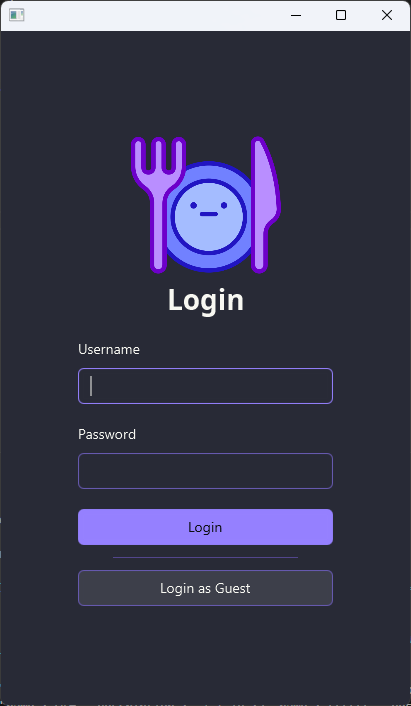
1. Cấu hình máy chủ

Ứng dụng đã được chạy thử trên thiết bị có thông tin:

* Hệ điều hành Window 11 Pro 64-bit (10.0, Build 22621)
* CPU Intel(R) Xeon(R) CPU E5-2560 v2 @ 2.60GHz (16 CPUs)
* Memory: 32GB

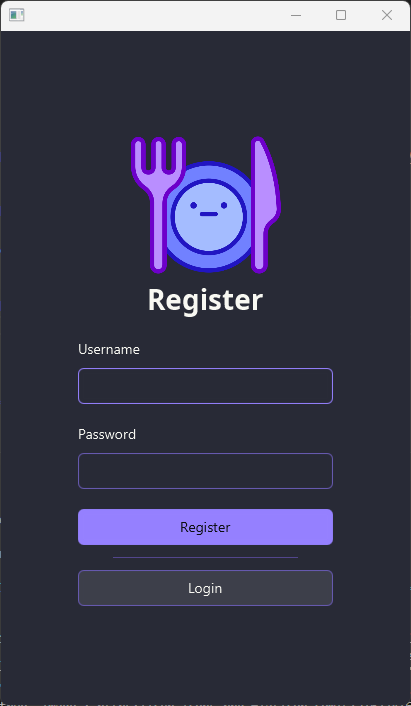
Quả trình chạy thử hoạt động khá ổn định tuy nhân có một số sự chậm trễ trong quá trình thử nghiệm.

1. Giao diện ứng dụng
   1. Giao diện đăng nhập



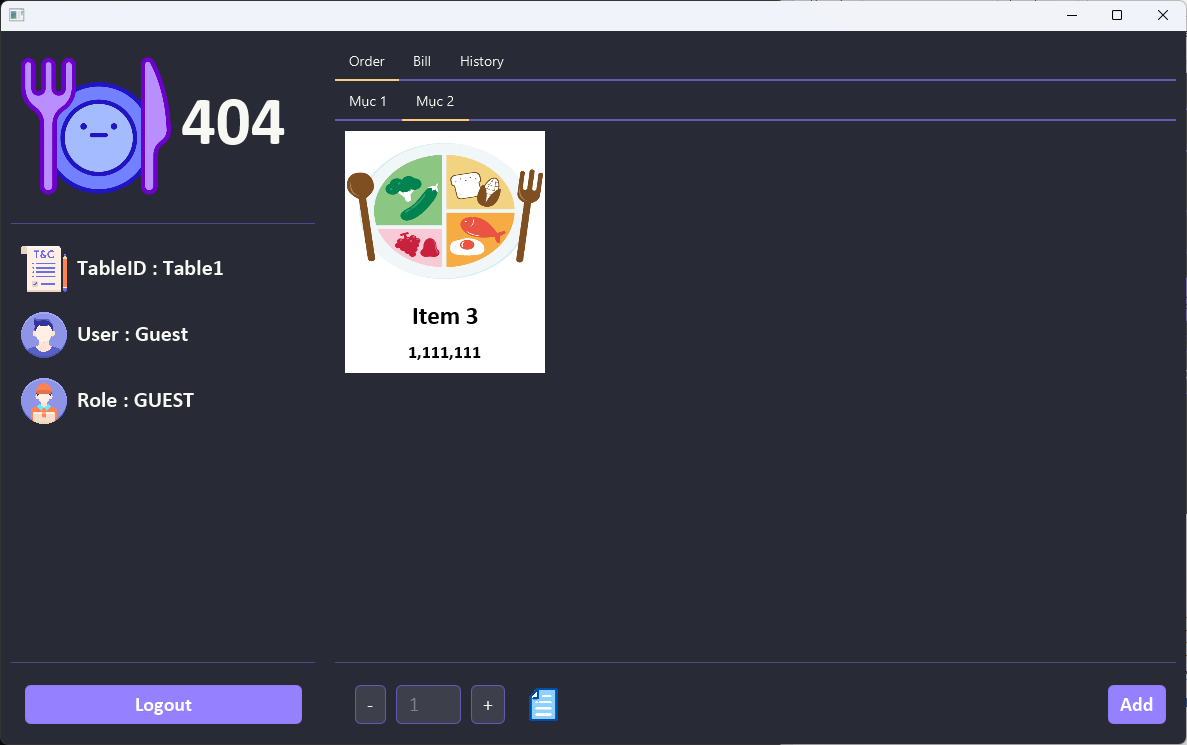
Hình 23 Giao diện đăng nhập

* 1. Giao diện đăng ký



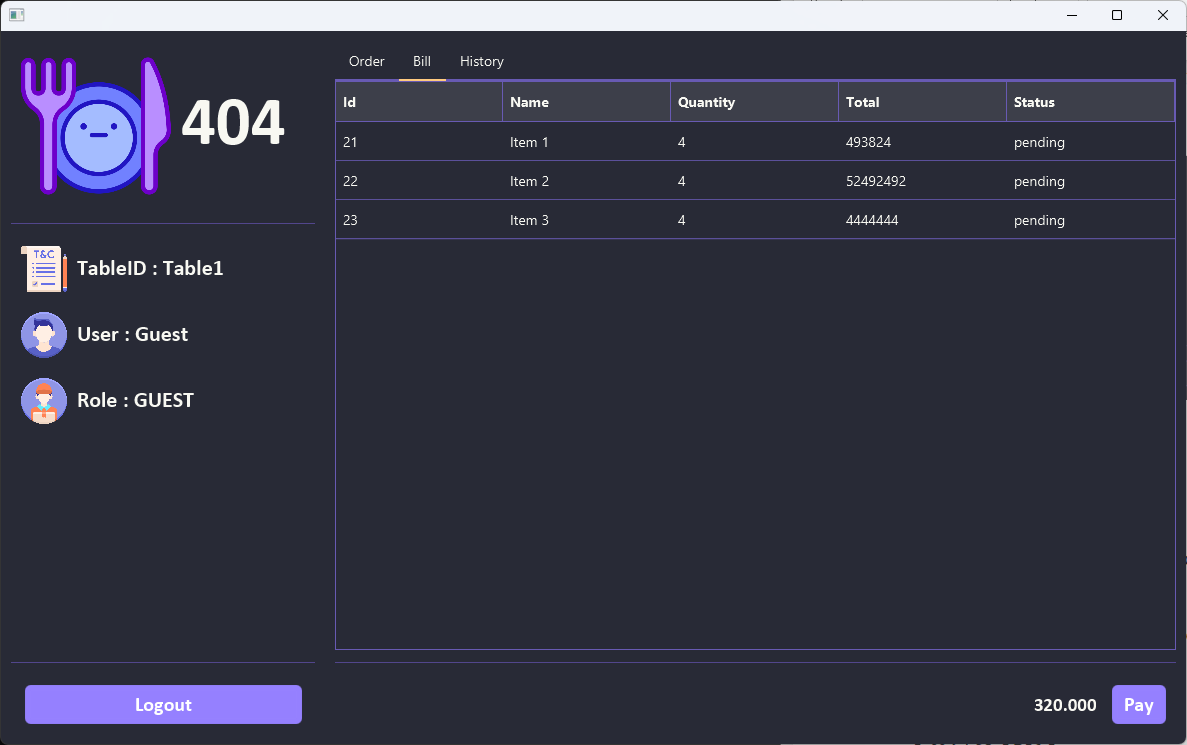
Hình 24 Giao diện đăng ký

* 1. Giao diện menu



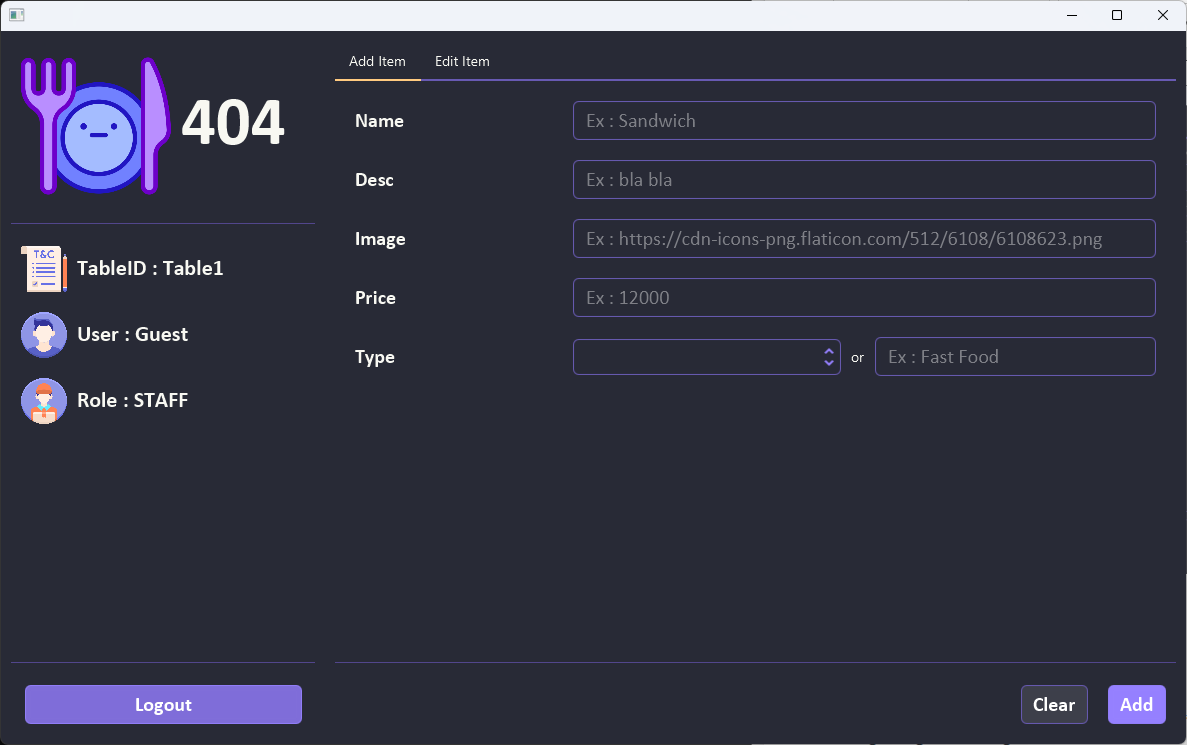
Hình 25 Giao diện menu

* 1. Giao diện xem đơn hàng



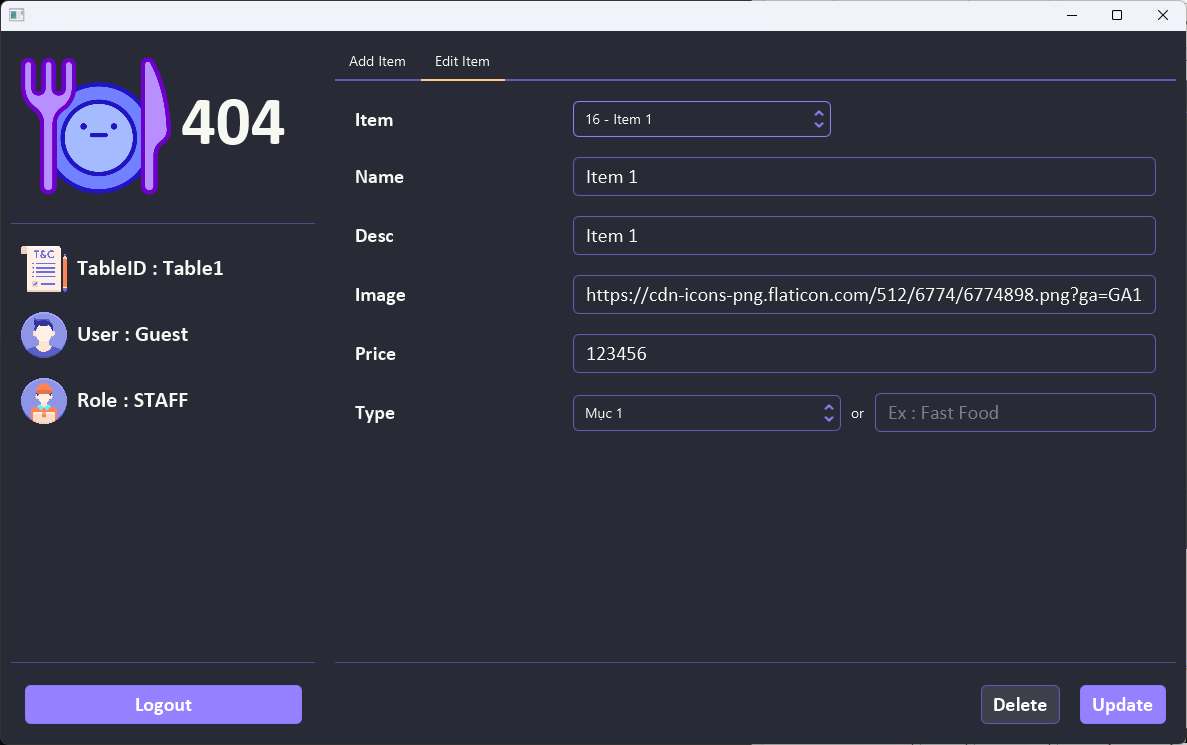
Hình 26 Giao diện xem đơn hàng

* 1. Giao diện thêm món ăn



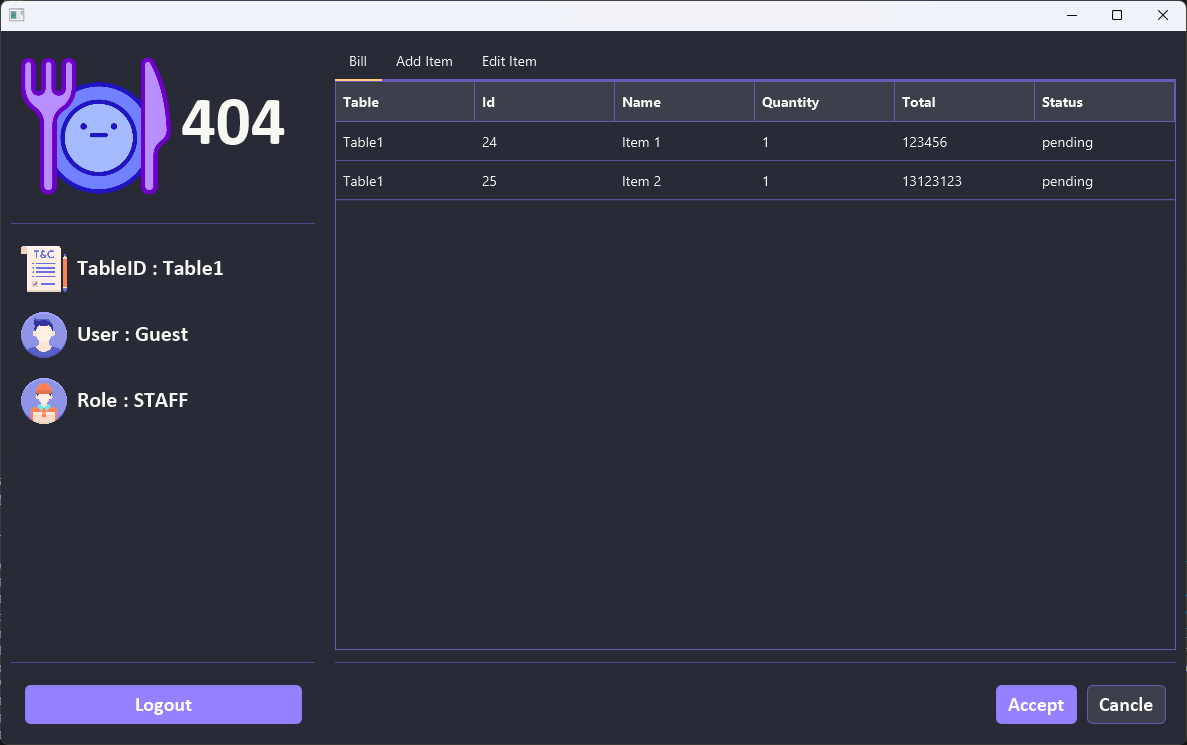
Hình 27 Giao diện thêm món ăn

* 1. Giao diện sửa món ăn



Hình 28 Giao diện sửa món ăn

* 1. Giao diện cập nhật đơn hàng



Hình 29 Giao diện cập nhật đơn hàng

1. Kết luận

Tuy vẫn còn một số sai sót nhưng ứng dụng đã đạt được một số thành tự sau:

* Ứng dụng hoạt động ổn định.
* Giao diện người dùng dễ sử dụng.
* Khả năng tương tác thời gian thực ổn định.
* Chương trình có database

# **TỔNG KẾT**

Dự án xây dựng ứng dụng đặt món bằng Java sử dụng mô hình Client-Server đã mang lại nhiều kết quả đáng kể và cung cấp những kiến thức về phát triển phầm mềm. Qua việc tập trung vào việc xây dựng các chức năng của ứng dụng, dự án đã đạt được những điểm cốt lõi sau:

* Tính Hoàn Thiện Của Hệ Thống: Hệ thống đã triển khai thành công các chức năng cơ bản của ứng dụng. Các chức năng này hoạt động ổn định và phản hồi nhanh chóng với người dùng.
* Giao Diện Người Dùng Thân Thiện: Giao diện người dùng được thiết kế để đảm bảo trải nghiệm người dùng tốt nhất có thể, đồng thời tối ưu hóa việc thao tác và tương tác với các chức năng của ứng dụng.
* Quản Lý Dữ Liệu Hiệu Quả: Việc quản lý dữ liệu được thực hiện một cách an toàn và hiệu quả thông qua việc sử dụng MySQL. Dữ liệu được lưu trữ và truy xuất một cách đáng tin cậy, đồng thời đảm bảo bảo mật thông tin người dùng.
* Thách Thức và Học Hỏi: Trong quá trình phát triển, chúng ta đã đối mặt với những thách thức về thiết kế, triển khai và tương tác giữa các thành phần. Những thách thức này đã làm tăng cường kiến thức và kỹ năng của đội ngũ phát triển.
* Hướng Phát Triển Tương Lai: Mặc dù đã chạy ổn định trên nền tảng Window nhưng vẫn chưa thể phát triển sang nền tảng Mobile, nên có khả năng trong tương lại dự án sẽ được triển khai trên nền tảng mobile và người dùng có thể sử dụng chính thiết bị của họ để truy cập menu.

# **TÀI LIỆU THAM KHẢO**

[1. Wikipedia](https://en.wikipedia.org/wiki/Scrum_(software_development))

[2. Viblo.asia](https://viblo.asia/p/tong-quan-ve-scrum-phan-i-1VgZvExRKAw" \l "_scrum-master-ho-tro-cho-to-chuc-9)