**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH**

**Đề tài: Quản lý đặt món ăn và thanh toán đơn hàng**

**nhà hàng**

**Giảng viên hướng dẫn: Ths. Mai Văn Hà**

**Sinh viên thực hiện:**

**Nguyễn Văn Tài LỚP: 23T\_DT4 NHÓM: 23.12B**

**Nguyễn Tiến Sơn LỚP: 23T\_DT4 NHÓM: 23.12B**

**Đà Nẵng 2024-2025**

LỜI MỞ ĐẦU

Trong thời đại số hóa hiện nay, công nghệ đã và đang trở thành một phần không thể thiếu trong mọi lĩnh vực của cuộc sống, từ giáo dục, y tế, tài chính cho đến ngành dịch vụ. Đặc biệt, trong lĩnh vực quản lý nhà hàng – một ngành có tính cạnh tranh cao và yêu cầu sự chính xác, nhanh chóng – việc ứng dụng công nghệ hiện đại không chỉ giúp tối ưu hóa quy trình vận hành mà còn mang lại trải nghiệm tốt hơn cho khách hàng.

Ứng dụng “Đặt món ăn và thanh toán đơn hàng” được phát triển nhằm giải quyết những thách thức trong công tác quản lý truyền thống, vốn phụ thuộc nhiều vào thao tác thủ công, dễ gây ra sai sót và mất thời gian. Với các tính năng nổi bật như đặt món ăn, quản lý đơn hàng, quản lí thực đơn, đề xuất món ăn, thanh toán linh hoạt, và in hóa đơn tự động, ứng dụng này mang lại sự tiện lợi và chính xác vượt trội trong hoạt động hàng ngày của nhà hàng.

Không chỉ dừng lại ở việc hỗ trợ vận hành, ứng dụng còn được xây dựng như một công cụ mạnh mẽ để thu thập và phân tích dữ liệu khách hàng. Những thông tin giá trị về thói quen ăn uống, sở thích, và xu hướng tiêu dùng của thực khách sẽ được xử lý và lưu trữ có hệ thống. Đây chính là nguồn dữ liệu quý giá giúp nhà hàng đưa ra các quyết định chiến lược, từ cải tiến menu, xây dựng chương trình ưu đãi, đến hoạch định chiến lược kinh doanh dài hạn.

Với sự kết hợp giữa công nghệ và tầm nhìn chiến lược, ứng dụng “Đặt món ăn và thanh toán đơn hàng” giúp các nhà hàng vận hành hiệu quả hơn mà còn trở thành động lực thúc đẩy sự đổi mới và phát triển toàn diện của ngành dịch vụ ăn uống. Không chỉ là một giải pháp quản lý mà còn là cầu nối giúp nhà hàng tiếp cận khách hàng một cách gần gũi hơn, xây dựng mối quan hệ lâu dài và tạo ra giá trị vượt trội trong bối cảnh cạnh tranh ngày càng khốc liệt.

**MỤC LỤC**

[1](#_Toc184475790)

[LỜI MỞ ĐẦU 2](#_Toc184475791)

[DANH MỤC HÌNH VẼ 5](#_Toc184475792)

[1. GIỚI THIỆU ĐỀ TÀI 6](#_Toc184475793)

[2. PHÂN TÍCH CHỨC NĂNG HỆ THỐNG 7](#_Toc184475794)

[2.1. Quản lí thực đơn 7](#_Toc184475795)

[2.2. Đặt món 7](#_Toc184475796)

[2.3. Thanh toán 7](#_Toc184475797)

[2.4. Quản lí tài chính 7](#_Toc184475798)

[3. THIẾT KẾ CẤU TRÚC DỮ LIỆU 7](#_Toc184475799)

[3.1. Phát biểu bài toán 7](#_Toc184475800)

[3.2. Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống 8](#_Toc184475801)

[3.2.1. Thêm món vào thực đơn 8](#_Toc184475802)

[3.2.2. Xóa món khỏi thực đơn 9](#_Toc184475803)

[3.2.3. Kiểm tra doanh thu 9](#_Toc184475804)

[3.2.4. Đề xuất món ăn bán chạy nhất 10](#_Toc184475805)

[3.2.5. Đặt món ăn theo bàn 11](#_Toc184475806)

[3.2.6. Thêm món và xóa món mà người dùng chọn 12](#_Toc184475807)

[3.2.7. Thanh toán và in hóa đơn 13](#_Toc184475808)

[3.2.8. Đổi bàn 14](#_Toc184475809)

[4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG 15](#_Toc184475810)

[4.1. Cấu trúc hệ thống hướng đối tượng 15](#_Toc184475811)

[4.1.1. Các class: 15](#_Toc184475812)

[4.1.2. Các tính chất hướng đối tượng được áp dụng vào chương trình 19](#_Toc184475813)

[4.2. Kết quả 20](#_Toc184475814)

[4.2.1. Cấu trúc chương trình 20](#_Toc184475815)

[4.2.2. Giao diện chính của chương trình 21](#_Toc184475816)

[4.2.3. Kết quả thực thi của chương trình 22](#_Toc184475817)

[4.2.4. Nhận xét 32](#_Toc184475818)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 32](#_Toc184475819)

[a. Kết luận 32](#_Toc184475820)

[b. Hướng phát triển 32](#_Toc184475821)

[TÀI LIỆU THAM KHẢO 34](#_Toc184475822)

DANH MỤC HÌNH VẼ

[Hình 1: Cấu trúc chương trình 20](#_Toc184567170)

[Hình 2: Giao diện chính của chương trình 21](#_Toc184567171)

[Hình 3: Giao diện chính bên quản lí 21](#_Toc184567172)

[Hình 4: Giao diện chính bên nhân viên 22](#_Toc184567173)

[Hình 5: Chọn bàn trước khi đặt món ăn 22](#_Toc184567174)

[Hình 6: Chọn món ăn 23](#_Toc184567175)

[Hình 7: Xác nhận hóa đơn 23](#_Toc184567176)

[Hình 8: Thực đơn của quán 24](#_Toc184567177)

[Hình 9: Danh sách bàn ăn đang được sử dụng 24](#_Toc184567178)

[Hình 10: Thêm món ăn vào đơn hàng có sẵn 25](#_Toc184567179)

[Hình 11: Xóa món ăn từ đơn hàng có sẵn 26](#_Toc184567180)

[Hình 12: Tính năng đổi bàn 26](#_Toc184567181)

[Hình 13: Thông báo chưa có bàn nào được đặt 27](#_Toc184567182)

[Hình 14: In hóa đơn khi có yêu cầu thanh toán 27](#_Toc184567183)

[Hình 15: Thanh toán khi chưa có bàn nào được đặt 27](#_Toc184567184)

[Hình 16: Danh sách chức năng bên quản lí 28](#_Toc184567185)

[Hình 17: Thay đổi thông tin đăng nhập 28](#_Toc184567186)

[Hình 18: Thêm, xóa tài khoản và mật khẩu 28](#_Toc184567187)

[Hình 19: Thêm tài khoản đăng nhập mới 28](#_Toc184567188)

[Hình 20: Xóa tài khoản đăng nhập 29](#_Toc184567189)

[Hình 21: Thực đơn của nhà hàng 29](#_Toc184567190)

[Hình 22: Chỉnh sửa thực đơn 29](#_Toc184567191)

[Hình 23: Thêm món ăn mới vào thực đơn 30](#_Toc184567192)

[Hình 24: Xóa món ăn ra khỏi thực đơn 30](#_Toc184567193)

[Hình 25: Thay đổi số lượng bàn ăn 31](#_Toc184567194)

[Hình 26: Xem hóa đơn ở một khung giờ trong một ngày 31](#_Toc184567195)

[Hình 27: Xem số lượng các món được bán ra 32](#_Toc184567196)

# GIỚI THIỆU ĐỀ TÀI

Đề tài xây dựng ứng dụng đặt món ăn và thanh toán đơn hàng tập trung vào việc phát triển một công cụ số hỗ trợ hiệu quả cho nhân viên và quản lý của nhà hàng. Trong bối cảnh ngành dịch vụ ăn uống đang không ngừng phát triển, việc tối ưu hóa quy trình làm việc và quản lý trở thành một nhu cầu thiết yếu. Ứng dụng này nhằm mục đích nâng cao hiệu suất làm việc của nhân viên, giảm thiểu sai sót trong quá trình nhận và xử lý đơn hàng, đồng thời cung cấp cho quản lý nhà hàng các công cụ giám sát và phân tích hoạt động kinh doanh.

* **Mục đích thực hiện**

Đề tài “Xây dựng ứng dụng đặt món ăn và thanh toán đơn hàng” tạo ra một phần mềm hỗ trợ các nhà hàng và nhân viên dễ dàng trong việc đặt món ăn, tăng cường hiệu quả hoạt động của nhà hàng, quản lý đơn hàng dễ dàng hơn, hỗ trợ quản lý và phân tích dữ liệu khách hàng. Từ đó, nhà hàng có thể đưa ra định hướng phát triển phù hợp trong tương lai.

* **Mục tiêu**

Đề tài tạo ra một phần mềm có khả năng quản lý và thanh toán đơn hàng một cách hiệu quả và nhanh chóng. Bao gồm:

* Xây dựng giao diện đơn giản, dễ sử dụng.
* Cung cấp cho người sử dụng tính năng đặt món ăn và thanh toán đơn hàng.
* Cho phép người dùng thống kê số lượng đơn hàng, tổng số tiền đã thu trong ngày, số tiền đã nhận và tổng số hóa đơn của từng nhân viên.
* **Phạm vi và đối tượng nghiên cứu**

Phạm vi: Chú trọng vào phát triển một phần mềm cung cấp khả năng đặt món ăn , thanh toán các đơn hàng, quản lý các hóa đơn cho các nhà hàng.

Đối tượng nghiên cứu: những nhân viên, quản lý làm việc trong nhà hàng.

* **Phương pháp nghiên cứu**

Nghiên cứu tài liệu: Tham khảo các tài liệu, báo cáo, các văn bản có liên quan.

Phân tích yêu cầu**:** Phân tích các yêu cầu của nhà hàng đối với ứng dụng để xây dựng các chức năng phù hợp.

Thiết kế và lập trình: Từ những dữ liệu thu thập được, thiết kế giao diện, các chức năng của ứng dụng bằng ngôn ngữ C++.

* **Các chức năng chính**
* Quản lí thực đơn
* Đặt món
* Thanh toán
* Quản lí tài chính

# PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

## Quản lí thực đơn

* Thêm món mới: Cho phép quản lý thêm món ăn mới vào thực đơn với thông tin như tên, giá.
* Cập nhật món ăn: Chỉnh sửa thông tin món ăn hiện có, như thay đổi giá cả.
* Xóa món ăn: Xóa các món ăn không còn được phục vụ khỏi thực đơn.
* Phân loại món ăn: Phân loại các món ăn theo các danh mục như món chính, món khai vị, món tráng miệng, đồ uống.
* Đề xuất những món bán chạy nhất.

## Đặt món

* Tạo đơn hàng: có thể tạo đơn hàng mới, chọn món ăn từ thực đơn, số lượng món đó với số bàn tương ứng đã chọn.
* Tùy chọn cho món ăn: Thêm ghi chú hoặc các yêu cầu đặc biệt khi khách đặt món (ví dụ, không cay, ít đường, thêm topping,...).
* Chỉnh sửa đơn hàng: Thay đổi hoặc thêm bớt món trong đơn hàng trước khi xác nhận.

## Thanh toán

* Tính tổng hóa đơn: Tự động tính toán tổng giá trị của đơn hàng, bao gồm các khoản phụ thu như thuế, phí dịch vụ.
* Xuất hóa đơn: In hoặc xuất hóa đơn điện tử chi tiết cho khách hàng sau khi thanh toán.
* Phương thức thanh toán: Hỗ trợ thanh toán qua tiền mặt, chuyển khoản.

## Quản lí tài chính

* Báo cáo doanh thu: Tổng hợp doanh thu theo ngày giúp nhà quản lý nắm được tình hình kinh doanh.

# THIẾT KẾ CẤU TRÚC DỮ LIỆU

## Phát biểu bài toán

A, Input

* File menu.txt.
* File login.txt.
* Nhập từ bàn phím các yêu cầu.

B, Output

* Thực đơn.
* Số lần được đặt của các món ăn.
* Danh sách tất cả hóa đơn trong ngày.

## Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống

### Thêm món vào thực đơn

* Tham số đầu vào:
* Yêu cầu thêm món.
* File menu.txt.
* Tên của món mới.
* Giá của món mới.
* Đơn vị tính của món.
* Xử lí: Khi quản lí yêu cầu thêm một món mới vào thực đơn thì sẽ được nhập tên, giá, đơn vị tính của món mới. Sau đó món sẽ được tự động thêm vào cuối file thực đơn với ID tự động tăng.
* Thuật toán:
* In ra thực đơn cho quản lí xem bởi hàm view\_menu().
* Tên, giá, đơn vị tính của món mới được lưu vào biến name, price, unit.
* Mở file menu.txt, sau đó đọc dòng cuối cùng của file để lấy ra ID món cuối cùng.
* ID sẽ tăng lên một và ghi vào cuối file thông tin món mới với ID, name, unit, price.
* File menu\_suggest dùng để tổng hợp lại số lần được đặt của các món cũng sẽ được cập nhập sau đó.
* Tham số đầu ra:
* File menu.txt và menu\_suggest.txt sau khi đã cập nhật món mới.
* Thông báo đã thêm món thành công.
* Độ phức tạp: O(n).

### Xóa món khỏi thực đơn

* Tham số đầu vào:
* Yêu cầu xóa món từ người dùng.
* File menu\_suggest.txt.
* ID món muốn xóa.
* Xử lí: In ra thực đơn để người dùng xem để chọn món muốn xóa bằng cách nhập ID món đó. Sau đó tạo ra mảng chứa các thông tin của món ăn đọc ra từ file, khi đọc đến món cần xóa thì bỏ qua. Xóa hết dữ liệu trong file để ghi toàn bộ thông tin từ mảng sang file.
* Thuật toán:
* Tạo struct Dish để lưu trữ thông tin của món ăn.
* Tạo vector<Dish> , mở file menu.txt.
* In ra thực đơn bằng view\_menu(), sau đó yêu cầu người dùng nhập ID món muốn xóa.
* Đọc file menu.txt từng dòng một và lưu thông tin vào mảng, khi gặp ID món trùng với ID được nhập vào thì bỏ qua.
* Xóa hết nội dung file menu.txt và ghi hết toàn bộ nội dung trong mảng ra file.
* File menu\_suggest.txt dùng để chứa thông tin về số lần đặt của từng món cũng sẽ được cập nhật khi được gửi yêu cầu xóa món.
* Tham số đầu ra:
* File menu.txt và menu\_suggest sau khi được cập nhật.
* Thông báo xóa món thành công.
* Độ phức tạp: O(n).

### Kiểm tra doanh thu

* Tham số đầu vào:
* Yêu cầu xem doanh thu.
* Thời điểm cần xem (khung giờ, ngày, tháng, năm).
* File doanh thu là file tổng hợp tất cả các hóa đơn đã thanh toán trong ngày hôm đấy.
* Xử lí: Sau khi người dùng yêu cầu kiểm tra doanh thu thì sẽ nhập ngày tháng năm và sau khi mở file doanh thu thành công thì sẽ được nhập thêm khung giờ (hh – hh). Từ đó chương trình sẽ in ra màn hình toàn bộ những hóa đơn có trong thời điểm đó.
* Thuật toán:
* Người dùng sẽ nhập ngày tháng năm, từ đó sẽ tạo ra tên file dd\_mm\_yyyy.txt. Nếu file không tồn tại thì hệ thống sẽ thông báo không thể mở file.
* Sau đó, người dùng sẽ nhập khung giờ cần xem (hh hh), thông tin này sẽ được lưu vào biến start và end.
* Khởi tạo một biến check lúc đầu bằng 1. Mở file doanh thu, tìm vị trí từ “Hóa” (Hóa trong Hóa đơn) và nếu biến check bằng 1 thì lưu vị trí của nó vào biến line\_begin\_to\_print. Sau đó tìm tiếp từ “Thời gian: ”, xét xem 2 kí tự sau nó có nằm trong khung giờ cần tìm start và end hay không. Nếu có thì biến check = 0 và vị trí của dòng đấy sẽ được lưu vào biến line\_end\_to\_print. Lặp lại các bước trên cho tới khi hết file.
* Đưa con trỏ file về đầu, đọc từng dòng từ đầu cho đến vị trí dòng bằng line\_begin\_to\_print, sau đó đọc những dòng tiếp theo thì sẽ in ra màn hình cho đến khi tới vị trí dòng line\_end\_to\_print thì dừng lại.
* Đóng file.
* Tham số đầu ra:
* Tất cả các hóa đơn tại thời điểm mà người dùng muốn xem.
* Độ phức tạp: O(n).

### Đề xuất món ăn bán chạy nhất

* Tham số đầu vào:
* File menu\_suggest.txt.
* Xử lí: Một mảng động được tạo ra để lưu trữ thông tin về việc số lần từng món ăn được đặt. Sau đó tìm ra 3 món ăn có số lần đặt cao nhất thì khi in ra thực đơn sẽ in thêm kí hiệu “Best seller!!”.
* Thuật toán:
* Tạo ra một struct gồm ID món ăn, tên món ăn và số lần đặt count. Sau đó tạo một mảng để lưu các thông tin về các món ăn.
* File menu\_suggest.txt sẽ chứa các thông tin về số lần đặt của các món ăn, những món ăn chưa đặt lần nào hay mới thêm vào sẽ mặc định là 0.
* Khi chương trình thực hiện chức năng đặt đơn hàng thì sẽ đọc file đó và lưu toàn bộ thông tin vào mảng.
* Khi mà một hóa đơn được thanh toán thì số lượng món ăn trong đó sẽ tự động cập nhật với thông tin trong mảng.
* Người dùng đặt đơn hàng thì sẽ duyệt qua mảng đó để tìm ra ID 3 phần tử có lần đặt món nhiều nhất. Từ đó, khi in thực đơn ra nếu trùng ID thì sẽ in thêm kí hiệu “Best seller!!”.
* Khi người dùng thoát khỏi vai trò nhân viên thì tất cả các thông tin sẽ được cập nhât ra file menu\_suggest.txt.
* Tham số đầu ra:
* Ba món ăn được bán nhiều nhất sẽ được kí hiệu là “Best seller” khi đặt món ăn.
* Độ phức tạp: O(n).

### Đặt món ăn theo bàn

* Tham số đầu vào:
* Người dùng chọn bàn và chọn món ăn mình muốn.
* Xử lí: Khi người dùng truy cập vào mục "Nhận đơn hàng", hệ thống sẽ hiển thị danh sách các bàn trống trên màn hình (nếu tất cả các bàn đã được đặt thì hệ thống sẽ thông báo hết bàn trống và trở lại trạng thái trước khi chọn). Người dùng có thể lựa chọn một bàn bất kỳ trong số các bàn được liệt kê. Sau khi chọn bàn, giao diện sẽ tự động chuyển sang hiển thị menu để người dùng lựa chọn món ăn. Sau khi hoàn tất việc chọn món, hệ thống yêu cầu xác nhận đặt bàn. Toàn bộ thông tin về món ăn và bàn đã chọn sẽ được lưu trữ trong một mảng dữ liệu, đồng thời cập nhật trạng thái bàn thành "đã được đặt".
* Thuật toán:
* Lặp qua từng phần tử trong mảng order\_arr[] (kiểu dữ liệu Order). Kiểm tra điều kiện paid = true. Nếu bàn đã được thanh toán hoặc chưa được đặt, thêm thông tin bàn đó vào danh sách bàn trống và hiển thị trên màn hình. Nếu không có bàn nào có paid = true thì thông báo hết bàn trống.
* Khi hiển thị các bàn trống, lưu ID của từng bàn vào một vector tạm thời. Các bàn được đánh số thứ tự (STT) từ 1 trở đi khi in ra để người dùng chọn và nhập vào lựa chọn dễ dàng.
* Người dùng nhập STT của bàn muốn chọn. Nếu STT nhập không nằm trong danh sách hiển thị, báo lỗi và yêu cầu nhập lại. Khi nhập đúng, dùng STT làm chỉ số để truy xuất ID của bàn từ vector tạm.
* Sau khi lấy được ID của bàn, truy cập vào bàn đó và cho phép người dùng chọn món và lưu vào mảng number.
* Sau khi chọn xong, hiển thị yêu cầu xác nhận đặt món. Nếu người dùng từ chối, thiết lập lại mảng number về trạng thái ban đầu và không thực hiện thay đổi nào. Nếu đồng ý, cập nhật trạng thái paid của bàn đó thành false (bàn đang hoạt động).
* Trong quá trình đặt món, nếu khách hàng muốn hủy thao tác và thoát, đảm bảo hệ thống không ghi nhận thay đổi nào và quay lại trạng thái trước đó.
* Tham số đầu ra:
* Thông tin các món ăn đã chọn được tạo thành hóa đơn và chuyển đến bộ phận bếp để chuẩn bị.
* Hệ thống xác nhận bàn đã được đặt thành công và cập nhật trạng thái paid = false.
* Độ phức tạp: O(n).

### Thêm món và xóa món mà người dùng chọn

* Tham số đầu vào:
* Yêu cầu xóa món từ người dùng.
* Người dùng sẽ nhập vào STT của bàn.
* ID món muốn xóa.
* Xử lí: Hệ thống sẽ hiển thị danh sách các bàn đã được đặt nhưng chưa thanh toán (paid = false) để người dùng lựa chọn (nếu chưa có bàn nào được đặt thì hệ thống sẽ tự động thông báo và trở lại trạng thái trước khi chọn). Sau khi chọn bàn, danh sách các món ăn đã đặt tại bàn đó sẽ được hiển thị. Người dùng nhập số lượng món muốn xóa, nếu số lượng nhập lớn hơn số món hiện có, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại. Khi nhập đúng, hệ thống sẽ yêu cầu xác nhận. Nếu người dùng đồng ý, món ăn sẽ được xóa bớt theo số lượng đã nhập hoặc xóa toàn bộ nếu người dùng nhập 'all'. Nếu tất cả món ăn bị xóa, hệ thống sẽ thông báo bàn đã trống và cập nhật trạng thái bàn thành paid = true. Mọi thao tác nhập không đúng theo hướng dẫn đều được yêu cầu thực hiện lại.
* Thuật toán:
* Lặp qua từng phần tử trong mảng order\_arr[] (kiểu dữ liệu Order). Kiểm tra điều kiện paid = false. Nếu bàn chưa được thanh toán, thêm thông tin bàn đó vào danh sách bàn đã đặt và hiển thị trên màn hình. Nếu không có bàn nào có paid = false thì thông báo chưa có bàn nào được đặt.
* Khi hiển thị các bàn đã đặt, lưu ID của từng bàn vào một vector tạm thời. Các bàn được đánh số thứ tự (STT) từ 1 trở đi khi in ra để người dùng chọn và nhập vào lựa chọn dễ dàng.
* Người dùng nhập STT của bàn muốn cập nhật. Nếu STT nhập không nằm trong danh sách hiển thị, báo lỗi và yêu cầu nhập lại. Khi nhập đúng, dùng STT làm chỉ số để truy xuất ID của bàn từ vector tạm.
* Sau khi lấy được ID của bàn, truy cập vào bàn đó và cho phép người dùng chọn thêm món hoặc xóa món trong bàn đã chọn đó.
* Nếu người dùng chọn thêm món, hệ thống sẽ gọi hàm hiển thị menu và in ra danh sách các món ăn để người dùng lựa chọn. Người dùng nhập vào ID của món ăn và số lượng tương ứng. Sau khi hoàn tất, hệ thống sẽ truy cập vào bàn đang được chọn, cập nhật mảng number của bàn đó với thông tin món ăn và số lượng vừa thêm. Cuối cùng, thông báo cho người dùng rằng món ăn đã được thêm thành công.
* Nếu người dùng chọn xóa món, hệ thống sẽ hiển thị danh sách các món đã đặt tại bàn để người dùng lựa chọn món muốn xóa bớt hoặc hủy hoàn toàn. Người dùng nhập ID của món ăn và số lượng muốn xóa. Nếu nhập số món muốn xóa lớn hơn số lượng món hiện tại thì hệ thống sẽ yêu cầu nhập lại. Sau đó, hệ thống sẽ yêu cầu xác nhận. Nếu người dùng đồng ý, hệ thống truy cập vào bàn đang được chọn và cập nhật mảng number của bàn đó với số lượng món đã xóa, sau đó thông báo xóa món thành công. Ngược lại, nếu người dùng không đồng ý, mọi thay đổi sẽ bị hủy, hệ thống sẽ giữ nguyên trạng thái ban đầu và không ảnh hưởng đến dữ liệu hiện tại.
* Trong quá trình thao tác người dùng có thể thoát ra mà ảnh hưởng đến hệ thống.
* Nếu người dùng xóa hết món của bàn đó, hệ thống sẽ thông báo và chuyển paid từ false thành true.
* Tham số đầu ra:
* Mảng number của bàn được chọn được cập nhật.
* Thông báo xóa món thành công hoặc bàn đã hết món để xóa.
* Độ phức tạp: O(n).

### Thanh toán và in hóa đơn

* Tham số đầu vào:
* Người dùng nhập vào số bàn muốn thanh toán.
* Xử lí: Hệ thống hiển thị danh sách các bàn đã được đặt nhưng chưa thanh toán. Người dùng nhập STT tương ứng với bàn muốn chọn. Sau khi chọn, hệ thống sẽ hiển thị danh sách các món đã đặt tại bàn đó, bao gồm thông tin: tên món, giá, số lượng, đơn vị tính, tổng tiền từng món, và tổng tiền tất cả. Nếu người dùng bấm xác nhận, hóa đơn sẽ được ghi vào file, với tên file là ngày thanh toán của hóa đơn. Nếu file đã tồn tại thì tự động ghi vào dòng tiếp theo trong file đó.
* Thuật toán:
* Lặp qua từng phần tử trong mảng order\_arr[] (kiểu dữ liệu Order). Kiểm tra điều kiện paid = false. Nếu bàn chưa được thanh toán, thêm thông tin bàn đó vào danh sách bàn đã đặt và hiển thị trên màn hình. Nếu không có bàn nào có paid = false thì thông báo chưa có bàn nào được đặt và trở lại trạng thái ban đầu.
* Khi hiển thị các bàn đã đặt, lưu ID của từng bàn vào một vector tạm thời. Các bàn được đánh số thứ tự (STT) từ 1 trở đi khi in ra để người dùng chọn và nhập vào lựa chọn dễ dàng.
* Người dùng nhập STT của bàn muốn cập nhật. Nếu STT nhập không nằm trong danh sách hiển thị, báo lỗi và yêu cầu nhập lại. Khi nhập đúng, dùng STT làm chỉ số để truy xuất ID của bàn từ vector tạm.
* Sau khi lấy được ID của bàn, truy cập vào bàn đó và hiển thị cho người dùng danh sách các món ăn đã đặt một cách chi tiết.
* Yêu cầu xác nhận từ hệ thống, nếu người dùng đồng ý thì hóa đơn sẽ được lưu vào file, thiết lập lại các phần tử của number, đặt paid của bàn lại là true.
* Nếu người dùng không đồng ý thanh toán thì hệ thống cũng sẽ không bị ảnh hưởng gì.
* Tham số đầu ra:
* In ra hóa đơn, tổng tiền cho người dùng.
* Độ phức tạp: O(n).

### Đổi bàn

* Tham số đầu vào:
* Người dùng nhập vào số bàn hiện tại và số bàn mới
* Xử lí: Hệ thống hiển thị danh sách các bàn đã được đặt và chưa thanh toán. Người dùng nhập STT tương ứng với bàn hiện tại muốn đổi. Sau khi chọn, hệ thống sẽ hiển thị danh sách các bàn còn trống ra. Người dùng chọn vào số thứ tự của bàn mới, sau đó bấm xác nhận và thông báo đổi bàn thành công.
* Thuật toán:
* Lặp qua từng phần tử trong mảng order\_arr[] (kiểu dữ liệu Order). Kiểm tra điều kiện paid = false. Nếu bàn chưa được thanh toán, thêm thông tin bàn đó vào danh sách bàn đã đặt và hiển thị trên màn hình. Nếu không có bàn nào có paid = false thì thông báo chưa có bàn nào được đặt và trở lại trạng thái ban đầu.
* Khi hiển thị các bàn đã đặt, lưu ID của từng bàn vào một vector tạm thời. Các bàn được đánh số thứ tự (STT) từ 1 trở đi khi in ra để người dùng chọn và nhập vào lựa chọn dễ dàng. Người dùng nhập vào số thứ tự bàn hiện tại. Dựa vào STT này ta truy cập được ID của bàn nhờ vào vector. Sau đó lưu ID của bàn này lại.
* Người dùng nhập STT của bàn muốn chuyển sang. Nếu STT nhập không nằm trong danh sách hiển thị, báo lỗi và yêu cầu nhập lại. Khi nhập đúng, dùng STT làm chỉ số để truy xuất ID của bàn từ vector tạm.
* Sau khi lấy được ID của 2 bàn ta chuyển giá trị các phần tử trong number của bàn cũ sang cho mảng number của bàn mới. Đồng thời thiết lập lại paid của bàn cũ là true và paid của bàn mới là false.
* Tham số đầu ra:
* Đổi bàn thành công.
* Độ phức tạp: O(n).

# PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG

## Cấu trúc hệ thống hướng đối tượng

### Các class:

* ClassMenu\_model:
* Thuộc tính:
* int ID
* string Name
* ClassMenu(kế thừa menu\_model):
* Thuộc tính:
* double price;
* string unit;
* Phương thức:
* void set\_menu(int ID, string name, string unit, double price);
* menu();
* int get\_id();
* string get\_name();
* String get\_unit();
* double get\_price();
* Class Order (kế thừa menu\_model):
* Thuộc tính:
* bool paid;
* int \*number;
* Phương thức:
* void set\_order(int, int);
* bool getPaid();
* void reset\_paid();
* string return\_name();
* int return\_id();
* int getSelectedDishes();
* void order(int, int);
* void resetNumber(int);
* vector<int> returnSelectedDishes(int);
* bool check\_number\_change(int);
* void reset\_value\_of\_element(int);
* bool return\_value\_of\_element(int, int);
* void remove\_count\_of\_element(int, int);
* void reset\_value(const int&);
* void transfer\_oldTable\_to\_newTable(vector<int>&, const int&);
* int get\_number(int);
* Order();
* ~Order();
* Class Person:
* Thuộc tính:
* string username;
* string password;
* Phương thức:
* bool Login();
* Person();

~Person();

* Class Manager (kế thừa class Person):
* Phương thức:
* Manager();
* void change\_account();
* void edit\_menu();
* void add\_dish();
* void remove\_dish(int id\_delete);
* void check\_revenue();
* void view\_menu();
* void display\_func\_manager();
* void change\_table();
* void add\_dish\_to\_file\_suggest(string name, int ID);
* void remove\_dish\_from\_file\_suggest(int id\_delete);
* void view\_dish\_count();
* Class Staff (kế thừa class Person):
* Phương thức:
* void display\_func\_staff();
* void receiveOrder();
* void viewMenu();
* void updateOrderStatus();
* void print\_bill();
* void change\_table();
* Class Data:
* Thuộc tính:
* int order\_count;
* int table\_count;
* menu \*menu\_arr;
* menu\_bill \*order\_arr;
* best\_seller \*best\_seller;
* Phương thức:
* data\_order();
* void identify\_menu\_table\_count();
* void allocate\_memory();
* void set\_element\_order();
* void load\_menu\_from\_file(const string &filename);
* void startProgram();
* void getMenu();
* int getOneDish(int);
* void saveSelectedDish(int, int, int);
* bool check\_table(int);
* int get\_table(bool);
* int get\_id\_table(int);
* int get\_id\_dish(int);
* int getDishes(int);
* void deleteOrderedDishes(int);
* void get\_paid(int);
* bool check\_dishSelection(int);
* bool check\_IfDishSelected(int);
* void remove\_dishes(int, int);
* void remove\_dish(int, int, int);
* bool compare\_count(int, int, int);
* void pay\_and\_print\_bill(int&, string&);
* void logic\_change\_table(int&, int&);
* int return\_order\_count();
* void load\_best\_seller();
* void set\_value\_to\_best\_seller(int);
* void write\_to\_file\_menu\_suggest();
* ~data\_order();

### Các tính chất hướng đối tượng được áp dụng vào chương trình

* Tính kế thừa:
* Class Manager và Staff đã kế thừa thuộc tính username và password, phương thức login() từ lớp cơ sở Person.
* Class Menu và Order đều kế thừa các thuộc tính và phương thức của class Menu\_model.
* Tính đa hình
* Class Menu\_model là class trừu tượng, hàm set\_attribute() là hàm thuần ảo và nó được định nghĩa ở 2 lớp dẫn xuất.
* Tính bao đóng
* Các thuộc tính ở các class đều là private và protected, bên ngoài sẽ chỉ làm việc thông qua những phương thức (set, get, ...)
* Tính trừu tượng
* Class menu\_model là class trừu tượng và class này sở hữu phương thức trừu tượng là set\_attribute(). Hai lớp dẫn xuất Menu và Order sẽ phải định nghĩa lại theo cách riêng.

## Kết quả

### Cấu trúc chương trình

A diagram of a company

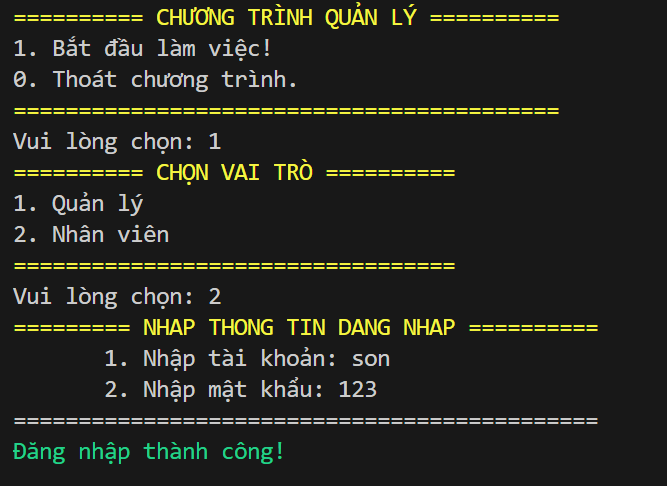
Description automatically generated

Hình 1: Cấu trúc chương trình

### Giao diện chính của chương trình

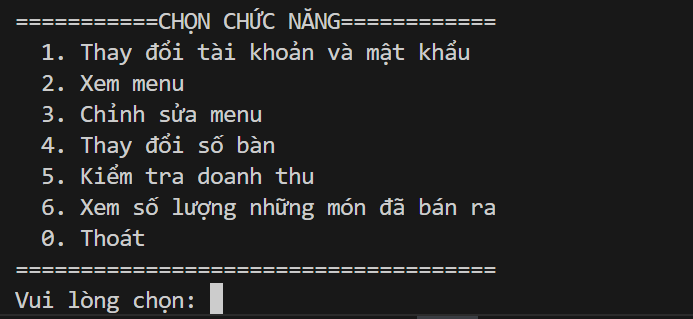
* Giao diện chung khi chuẩn bị vào làm việc:

Hình 2: Giao diện chính của chương trình



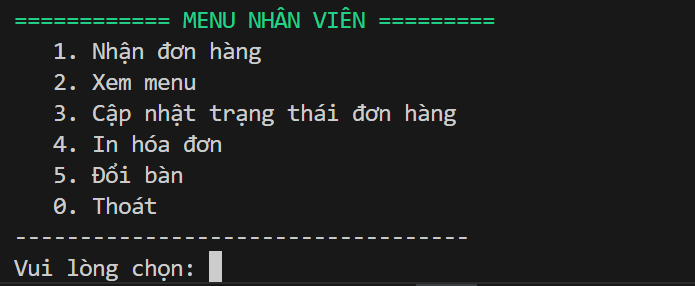
* Giao diện chính bên quản lý:

Hình 3: Giao diện chính bên quản lí

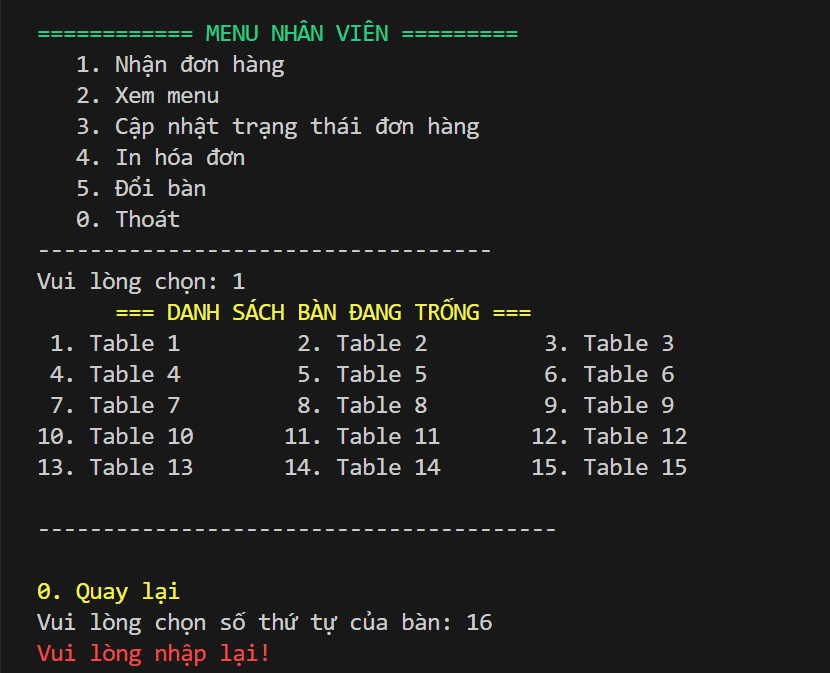


* Giao diện chính bên nhân viên:

Hình 4: Giao diện chính bên nhân viên



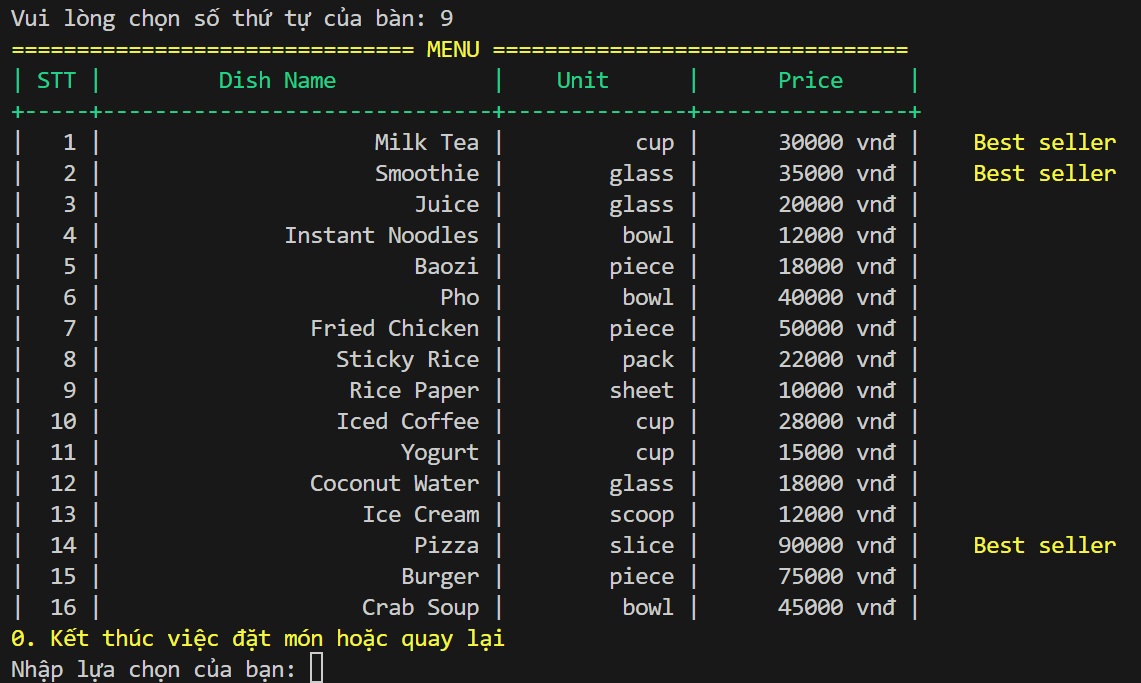
### Kết quả thực thi của chương trình

* Kết quả thực thi bên nhân viên:
* Kết quả tính năng nhận đơn hàng: chọn sai sẽ yêu cầu nhập lại

Hình 5: Chọn bàn trước khi đặt món ăn

* Còn nếu chọn đúng sẽ có kết quả thế này:

Hình 6: Chọn món ăn



* Tiếp tục chọn món, số lượng của món và sau đó xác nhận:

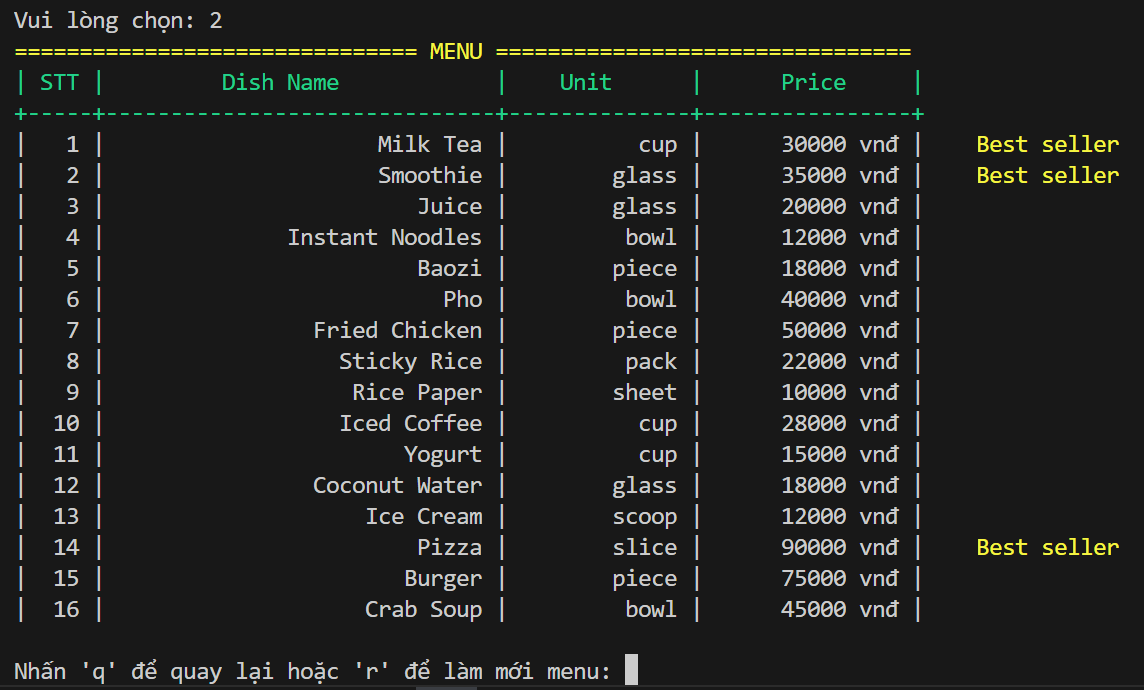
Hình 7: Xác nhận hóa đơn

A computer screen with text and numbers

Description automatically generated

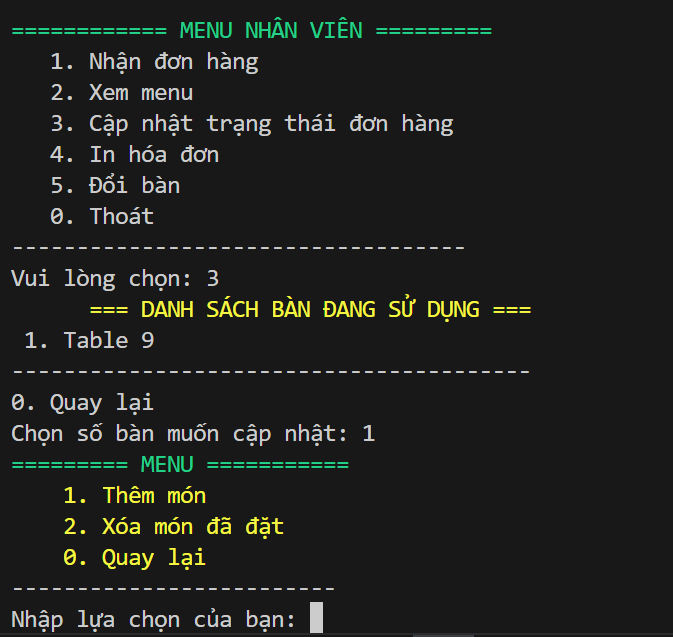
* Kết quả tính năng xem menu:

Hình 8: Thực đơn của quán



* Kết quả tính năng cập nhật đơn hàng:

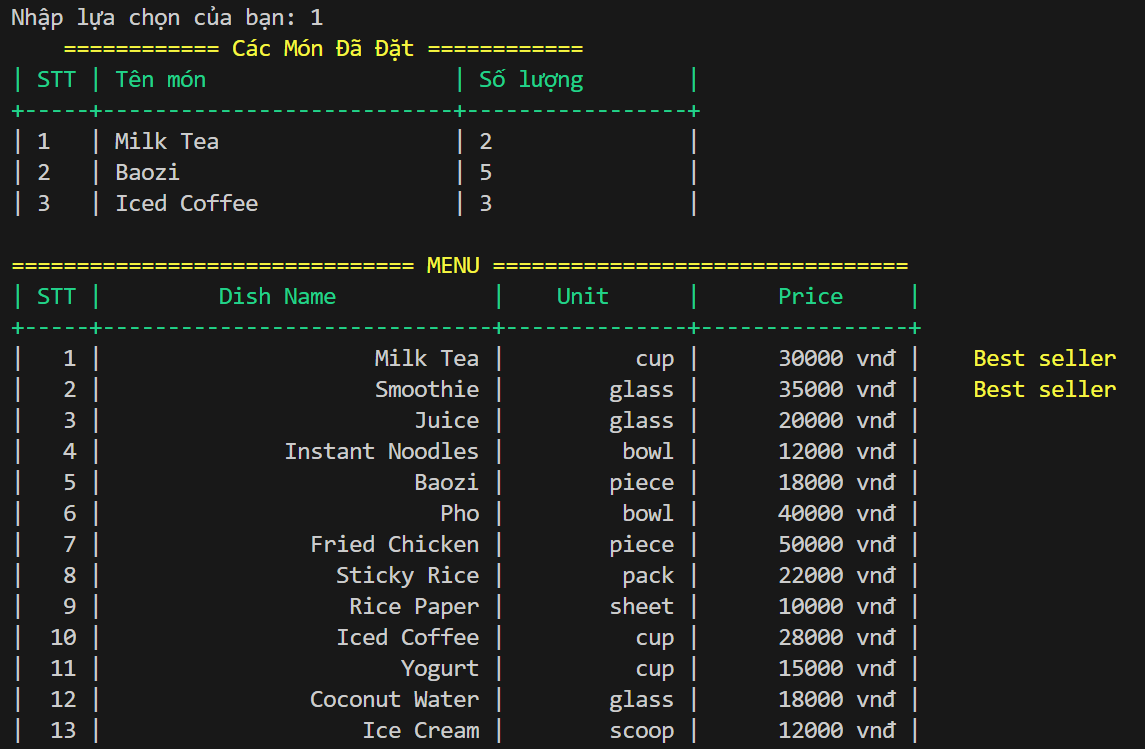
Hình 9: Danh sách bàn ăn đang được sử dụng

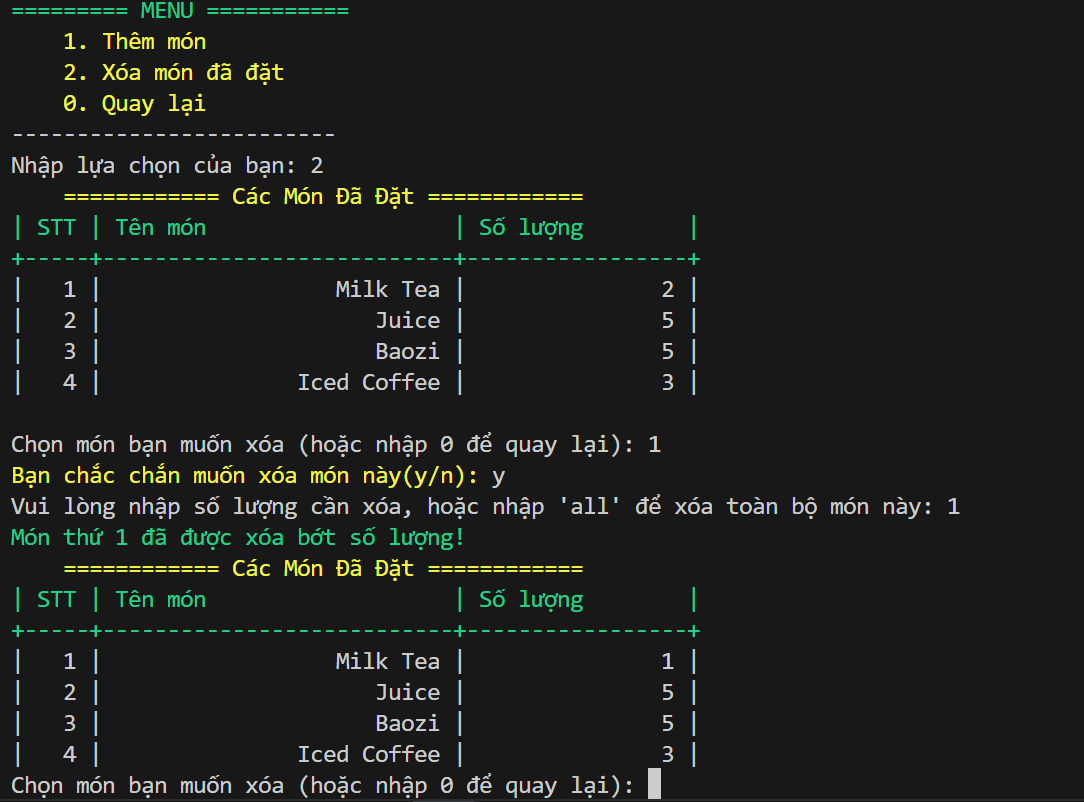


* A screen shot of a computer

  Description automatically generatedThêm mới:

Hình 10: Thêm món ăn vào đơn hàng có sẵn

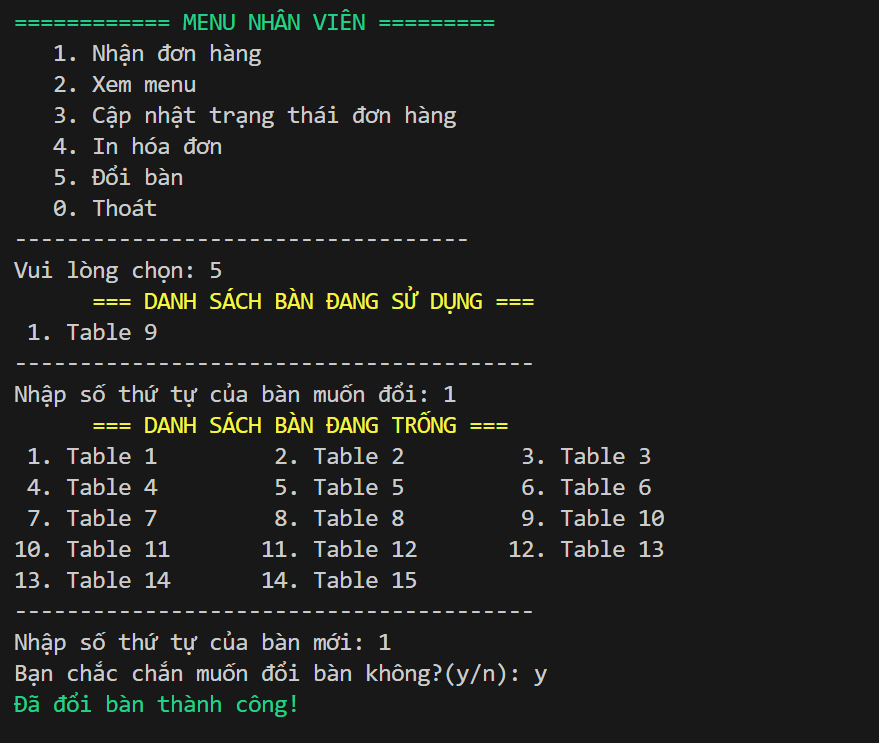


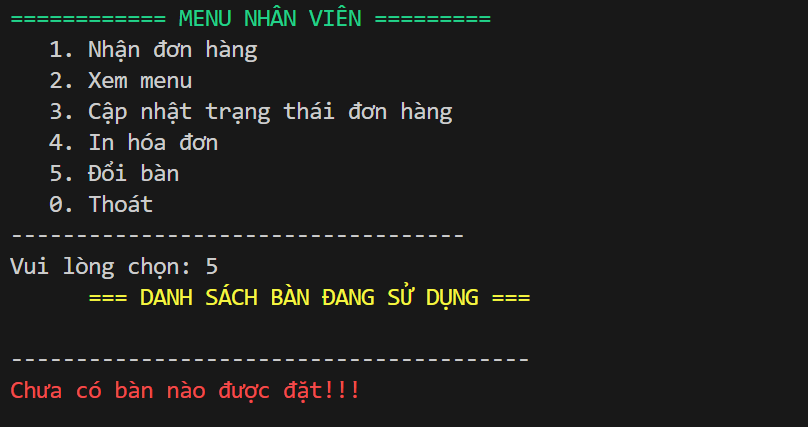
* Xóa món:

Hình 11: Xóa món ăn từ đơn hàng có sẵn

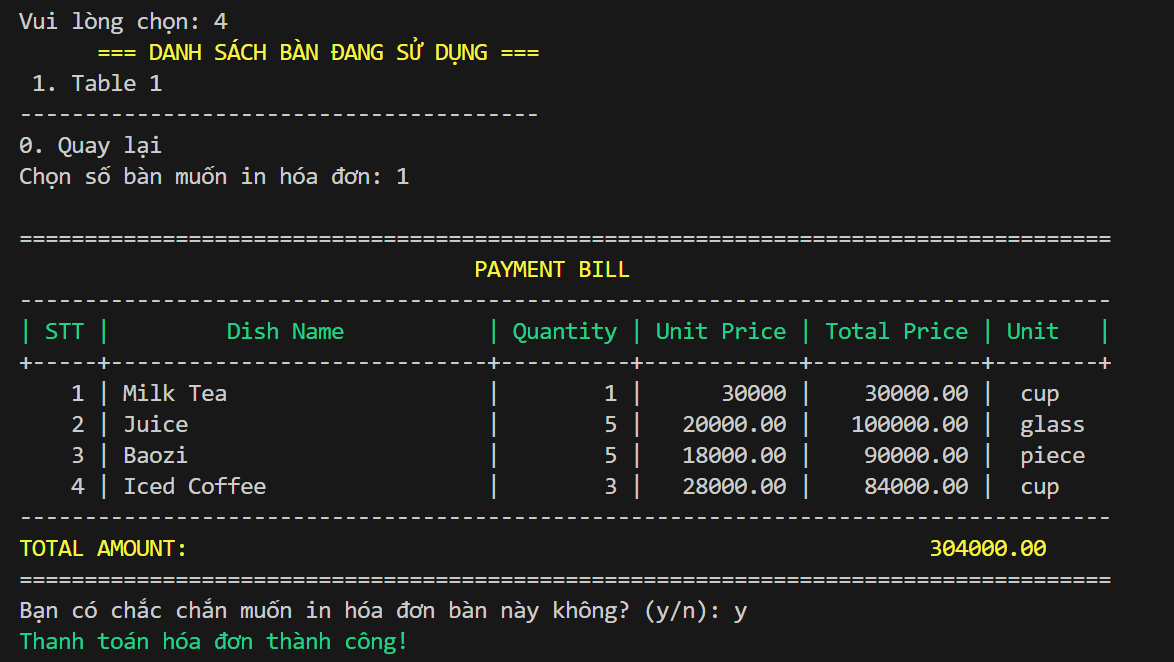
* Kết quả tính năng đổi bàn:

Hình 12: Tính năng đổi bàn



* Nếu chưa có bàn nào được đặt:

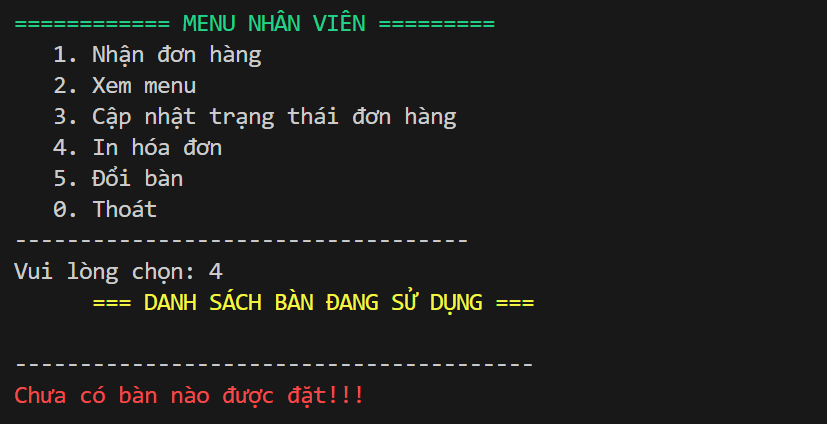
Hình 13: Thông báo chưa có bàn nào được đặt

* Kết quả tính năng in hóa đơn:

Hình 14: In hóa đơn khi có yêu cầu thanh toán

* Chưa có bàn nào được đặt:

Hình 15: Thanh toán khi chưa có bàn nào được đặt



* Kết quả thực thi bên quản lí:
* Hiển thị danh sách chức năng:

Hình 16: Danh sách chức năng bên quản lí

A screenshot of a computer

Description automatically generated

* Chức năng thay đổi tài khoản và mật khẩu:
* Chọn thay đổi thông tin của quản lí hay là nhân viên

Hình 17: Thay đổi thông tin đăng nhập

A screen shot of a computer

Description automatically generated

* Chọn thêm hay xóa tài khoản

Hình 18: Thêm, xóa tài khoản và mật khẩu

A black screen with yellow and orange text

Description automatically generated

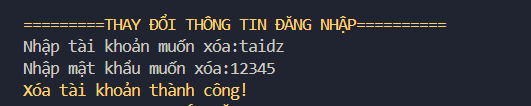
* Thêm tài khoản thành công

Hình 19: Thêm tài khoản đăng nhập mới

A black background with yellow and orange text

Description automatically generated

* Xóa tài khoản thành công



Hình 20: Xóa tài khoản đăng nhập

* Chức năng xem thực đơn

A screenshot of a menu

Description automatically generated

Hình 21: Thực đơn của nhà hàng

* Chức năng chỉnh sửa menu

A black background with yellow and orange text

Description automatically generated

Hình 22: Chỉnh sửa thực đơn

* Khi thêm một món ăn

A screenshot of a computer menu

Description automatically generated

Hình 23: Thêm món ăn mới vào thực đơn

* Khi xóa một món ăn

A screenshot of a menu

Description automatically generated

Hình 24: Xóa món ăn ra khỏi thực đơn

* Chức năng thay đối số lượng bàn ăn

A screenshot of a computer screen

Description automatically generated

Hình 25: Thay đổi số lượng bàn ăn

* Chức năng xem hóa đơn trong ngày

Hình 26: Xem hóa đơn ở một khung giờ trong một ngày

A screenshot of a computer

Description automatically generated

* Chức năng xem số lượng các món đã được bán ra

Hình 27: Xem số lượng các món được bán ra

A screenshot of a menu

Description automatically generated

### Nhận xét

Chương trình hoạt động hiệu quả, có tính ứng dụng cao trong thực tế. Thiết kế giao diện đẹp, dễ nhìn, thân thiện với người dùng, thuận tiện trong khi sử dụng, thích hợp với mô hình nhà hàng, quán café, …

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Dự án quản lý món ăn và thanh toán đơn hàng là một bước tiến quan trọng trong việc hiện đại hóa ngành dịch vụ ăn uống, mang lại một giải pháp hiệu quả trong việc tối ưu hóa quy trình phục vụ tại các nhà hàng, quán cà phê, và quán trà sữa... Hệ thống được thiết kế nhằm giảm thiểu các lỗi thủ công, cải thiện tốc độ phục vụ và tăng cường sự hài lòng của khách hàng. Thông qua việc áp dụng lập trình hướng đối tượng và phân tách rõ ràng các chức năng cho từng đối tượng sử dụng (quản lý và nhân viên), hệ thống đã thể hiện tính linh hoạt, dễ bảo trì, và khả năng mở rộng trong tương lai. Dự án cũng chứng minh được khả năng ứng dụng công nghệ để nâng cao hiệu quả quản lý và phát triển kinh doanh.

## Hướng phát triển

Thêm class Date để ghi nhận thời gian vào/ra ca, tính toán số giờ làm việc, lương và tăng ca, giúp tối ưu hóa phân bổ nguồn nhân lực và kiểm soát hiệu suất nhân viên. Thiết kế giao diện hiện đại, dễ sử dụng, giúp nhân viên thao tác nhanh chóng và khách hàng dễ dàng chọn món hoặc thanh toán.

Thêm class Customer để lưu lịch sử mua hàng, sở thích và triển khai các chiến dịch quảng cáo hiệu quả như tích điểm, giảm giá và ưu đãi. Thêm tính năng báo cáo doanh thu, lợi nhuận với biểu đồ trực quan giúp nhà quản lý phân tích và ra quyết định chiến lược dễ dàng hơn.

TÀI LIỆU THAM KHẢO

[1] Tài liệu OOP, Ths. Mai Văn Hà

[2] Tài liệu OOP, Ts. Lê Thị Mỹ Hạnh

[3] https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/

[4] https://viblo.asia/p/tim-hieu-ve-lap-trinh-huong-doi-tuong-object-oriented-programming-bxjvZwlBGJZ