**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH PYTHON**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM QUẢN LÝ CỬA HÀNG ĐỒ ĂN SÁNG**

**Giáo viên hướng dẫn: Th.S Trần Quý Nam**

**Sinh viên thực hiện:**  1. Bế Quang Hải

**Hà Nội, 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH PYTHON**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM QUẢN LÝ CỬA HÀNG ĐỒ ĂN SÁNG**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1771040011 | Bế Quang Hải | 08/09/2005 |  |  |

### 

### CÁN BỘ CHẤM THI

**Hà Nội, 2024**

**MỤC LỤC**

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 4](#_Toc182839639)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 5](#_Toc182839640)

[2.1. Thiết kế menu và dữ liệu 5](#_Toc182839641)

[2.2. Yêu cầu chức năng 6](#_Toc182839642)

[2.3. Phân tích dữ liệu 9](#_Toc182839643)

[CHƯƠNG 3. TRIỂN KHAI CHƯƠNG TRÌNH 13](#_Toc182839644)

[3.1. Khởi tạo dữ liệu và các biến chính 13](#_Toc182839645)

[3.2. Quản lý thực đơn 14](#_Toc182839646)

[3.3. Các chức năng thêm, sửa, xóa món ăn 15](#_Toc182839647)

[CHƯƠNG 4. TRIỂN KHAI VÀ KẾT QUẢ CỦA CHƯƠNG TRÌNH 18](#_Toc182839648)

[4.1. Sơ đồ khái quát cách đặt món hoạt động 18](#_Toc182839649)

[4.2. Đặt món ăn 19](#_Toc182839650)

[4.3 Thanh toán hóa đơn 22](#_Toc182839651)

[4.5. Các lựa chọn sắp xếp: 25](#_Toc182839652)

[4.6. Chức năng xem lại hóa đơn 27](#_Toc182839653)

[CHƯƠNG 5. ĐÁNH GIÁ HỆ THỐNG 29](#_Toc182839654)

[5.1. Điểm mạnh của chương trình 29](#_Toc182839655)

[5.2. Hạn chế và hướng phát triển 29](#_Toc182839656)

[5.3. Hướng phát triển: 29](#_Toc182839657)

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

Trong thời đại công nghệ 4.0, lập trình đã trở thành một trong những kỹ năng quan trọng, không chỉ giúp giải quyết các bài toán thực tiễn mà còn hỗ trợ quá trình số hóa và tự động hóa trong nhiều lĩnh vực khác nhau. Python, với sự đơn giản, linh hoạt và mạnh mẽ, đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất, đặc biệt trong việc phát triển các ứng dụng quản lý.

Bài tập lớn này được thực hiện với mục tiêu áp dụng kiến thức đã học về Python để xây dựng một ứng dụng quản lý nhà hàng. Ứng dụng không chỉ hỗ trợ quản lý thực đơn, đặt món, thanh toán mà còn giúp theo dõi doanh thu, từ đó tối ưu hóa hoạt động kinh doanh.

Thông qua bài tập, nhóm không chỉ mong muốn củng cố kỹ năng lập trình mà còn hiểu rõ hơn về cách tổ chức, xử lý dữ liệu và xây dựng giao diện thân thiện với người dùng. Đây cũng là cơ hội để khám phá tiềm năng của Python trong việc xây dựng các ứng dụng thực tế, phục vụ nhu cầu cuộc sống và công việc.

Tôi hy vọng bài tập lớn này sẽ góp phần thể hiện khả năng ứng dụng Python và mang lại giá trị thực tiễn trong lĩnh vực quản lý nhà hàng.

**Link code:**

https://github.com/Dog-Gold-VN/BTL-Python

# CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 2.1. Thiết kế menu và dữ liệu

**Triển khai:**

|  |
| --- |
| def menu\_chinh():      while True:          print(              """  1. Đặt món  2. Thanh toán  3. Quản lý thực đơn  4. In danh sách món  5. Tổng kết doanh thu  6. Xem lại hóa đơn  0. Thoát  """          )          lua\_chon = input("Chọn thao tác: ")          if lua\_chon == "1":              xu\_ly\_dat\_mon()          elif lua\_chon == "2":              xu\_ly\_thanh\_toan()          elif lua\_chon == "3":              quan\_ly\_thuc\_don()          elif lua\_chon == "4":              hien\_thi\_thuc\_don()          elif lua\_chon == "5":              tong\_ket\_doanh\_thu()          elif lua\_chon == "6":              xem\_hoa\_don()          elif lua\_chon == "0":              break          else:              print("Lựa chọn không hợp lệ.") |

**Kết quả:**

|  |
| --- |
| 1. Đặt món  2. Thanh toán  3. Quản lý thực đơn  4. In danh sách món  5. Tổng kết doanh thu  6. Xem lại hóa đơn  0. Thoát  Chọn thao tác: |

**Thiết kế dữ liệu**

Biến toàn cục

* menu\_do\_an: Dictionary lưu thực đơn, nơi key là ID\_mon và value là tuple (ten\_mon, gia\_tien).
* ban\_an: Dictionary lưu danh sách món ăn đã đặt cho từng bàn, nơi key là số bàn và value là dictionary các món đã đặt với key là ID\_mon và value là số lượng.
* doanh\_thu: Tổng doanh thu của toàn hệ thống.
* doanh\_thu\_theo\_ngay: Dictionary lưu doanh thu từng ngày, key là ngày và value là tổng doanh thu của ngày đó.
* hoa\_don: Danh sách lưu các hóa đơn đã thanh toán, mỗi hóa đơn là dictionary chứa thông tin số bàn, tổng tiền, và ngày thanh toán.

## 2.2. Yêu cầu chức năng

**2.2.1. Đọc thực đơn**

**Mục tiêu**

* Lưu trữ thực đơn vào tệp tin (MenuFoods.txt) để dễ dàng cập nhật.
* Đọc thực đơn từ tệp tin để sử dụng trong chương trình.

**Tạo tệp (MenuFoods.txt)**

Nội dung bao gồm ID món, tên món, giá tiền.

|  |
| --- |
| 1,Phở Bò,50000  2,Bún Chả,45000  3,Cơm Tấm,40000  4,Gỏi Cuốn,30000  5,Bánh Xèo,35000 |

**Hàm đọc thực đơn**

Đọc dữ liệu từ tệp MenuFoods.txt.

* Lưu vào một dictionary (menu\_do\_an), trong đó:
* Key: ID món.
* Value: Tuple chứa tên món và giá tiền.

**Viết hàm ghi thực đơn**

* Ghi nội dung của dictionary menu\_do\_an vào tệp tin (MenuFoods.txt).
* Sử dụng để lưu lại các thay đổi khi thêm, sửa, xóa món.

**Thêm chức năng hiển thị thực đơn**

* In toàn bộ danh sách món ăn ra console với định dạng dễ nhìn.

**2.2.2. Đặt và quản lý món ăn**

**Mục tiêu**

* Nhận yêu cầu đặt món từ khách hàng.
* Quản lý danh sách món ăn của từng bàn.

**Quản lý số bàn**

Xây dựng dictionary ban\_an để lưu danh sách món theo bàn:

* Key: Số bàn.
* Value: Dictionary chứa danh sách món đã đặt và số lượng.

**Viết hàm đặt món**

* Hiển thị thực đơn.
* Nhập số bàn (1-20).
* Nhập ID món và số lượng cần đặt.
* Lưu vào dictionary ban\_an.

**Hiển thị danh sách món đã đặt**

* In toàn bộ món và số lượng đã đặt cho từng bàn.

**2.2.3. Thanh toán và quản lý doanh thu**

**Mục tiêu**

* Tính tổng tiền của từng bàn khi thanh toán.
* Lưu doanh thu vào tệp tin.
* Quản lý doanh thu theo ngày.

**Tính tổng tiền bàn ăn**

* Lấy danh sách món từ ban\_an.
* Tính tổng tiền dựa trên số lượng và giá món ăn.

**Viết hóa đơn ra tệp DoanhThu.txt**

* Ghi thông tin bàn ăn, danh sách món, tổng tiền và thời gian thanh toán.

**Cập nhật doanh thu**

* Cập nhật tổng doanh thu trong biến doanh\_thu.
* Cập nhật doanh thu theo ngày trong dictionary doanh\_thu\_theo\_ngay.

**Xóa danh sách món đã thanh toán**

* Xóa thông tin bàn ăn khỏi ban\_an sau khi thanh toán.

**2.2.4. Xem hóa đơn và tổng kết doanh thu**

**Mục tiêu**

* Xem lại các hóa đơn đã ghi.
* Tổng kết doanh thu theo ngày hoặc tổng doanh thu.

**Hiển thị hóa đơn từ tệp**

* Đọc nội dung từ DoanhThu.txt.
* In ra console theo định dạng dễ hiểu.

**Tổng kết doanh thu**

* Tổng doanh thu toàn bộ.
* Tổng doanh thu theo từng ngày (lấy từ doanh\_thu\_theo\_ngay).
* Sắp xếp hóa đơn

## 2.3. Phân tích dữ liệu

**2.3.1. Tệp dữ liệu MenuFoods.txt**

**Mô tả**

Tệp MenuFoods.txt chứa thông tin về thực đơn của nhà hàng.

* Vai trò: Lưu trữ danh sách món ăn, cho phép dễ dàng đọc/ghi và cập nhật thực đơn.
* Định dạng: Text file, mỗi dòng lưu thông tin một món ăn, các trường cách nhau bởi dấu phẩy.

**Cấu trúc tệp**

|  |
| --- |
| ID\_món, Tên\_món, Giá\_tiền |

* ID\_món: Số nguyên, định danh duy nhất cho từng món ăn.
* Tên\_món: Chuỗi, tên gọi của món ăn.
* Giá\_tiền: Số nguyên, giá của món ăn (đơn vị: VND).

**Dữ liệu trong tệp**

|  |
| --- |
| 1,Phở Bò,50000  2,Bún Chả,45000  3,Cơm Tấm,40000  4,Gỏi Cuốn,30000  5,Bánh Xèo,35000 |

**2.3.2. Tệp dữ liệu DoanhThu.txt**

**Mô tả**

Tệp DoanhThu.txt lưu trữ thông tin hóa đơn của từng lần thanh toán.

* Vai trò: Ghi lại chi tiết từng giao dịch, phục vụ việc tra cứu và tổng kết doanh thu.
* Định dạng: Text file, mỗi hóa đơn được viết thành nhiều dòng, gồm thông tin bàn, danh sách món, và tổng tiền.

**Cấu trúc tệp**

|  |
| --- |
| === Hóa Đơn Bàn [số\_bàn] - Ngày [ngày\_giờ] ===  Món: [tên\_món], Số lượng: [số\_lượng], Giá: [giá tiền] VND, Tổng: [thành\_tiền] VND  ...  Tổng tiền: [tổng tiền bàn] VND\ |

**Ví dụ dữ liệu trong tệp**

|  |
| --- |
| === Hóa Đơn Bàn 1 - Ngày 2024-11-15 17:29 ===  Món: Phở Bò, Số lượng: 1, Giá: 50000 VND, Tổng: 50000 VND  Món: Bún Chả, Số lượng: 5, Giá: 45000 VND, Tổng: 225000 VND  Món: Bánh Xèo, Số lượng: 8, Giá: 35000 VND, Tổng: 280000 VND  Tổng tiền: 555000 VND    === Hóa Đơn Bàn 3 - Ngày 2024-11-18 12:33 ===  Món: Bánh Xèo, Số lượng: 23, Giá: 35000 VND, Tổng: 805000 VND  Tổng tiền: 805000 VND |

**2.3.3. Cấu trúc lưu trữ dữ liệu trong chương trình**

**1. Lưu trữ thực đơn (menu\_do\_an)**

Kiểu dữ liệu: Dictionary.

Cấu trúc:

|  |
| --- |
| menu\_do\_an = {  ID\_món: (Tên\_món, Giá\_tiền),  ...  } |

Ví dụ:

|  |
| --- |
| menu\_do\_an = {  1: ("Phở Bò", 50000),  2: ("Bún Chả", 45000),  3: ("Cơm Tấm", 40000),  4: ("Gỏi Cuốn", 30000),  5: ("Bánh Xèo", 35000)  } |

**2. Lưu trữ thông tin bàn ăn (ban\_an)**

Kiểu dữ liệu: Dictionary.

Cấu trúc:

|  |
| --- |
| ban\_an = {  Số\_bàn: {  ID\_món: Số\_lượng,  ...  },  ...  } |

Ví dụ:

|  |
| --- |
| ban\_an = {  1: {1: 2, 2: 1}, # Bàn 1: 2 phần Phở Bò, 1 phần Bún Chả  3: {4: 3} # Bàn 3: 3 phần Gỏi Cuốn  } |

**3. Lưu trữ doanh thu theo ngày (doanh\_thu\_theo\_ngay)**

Kiểu dữ liệu: Dictionary.

Cấu trúc:

|  |
| --- |
| doanh\_thu\_theo\_ngay = {  "YYYY-MM-DD": Tổng\_doanh\_thu,  ...  } |

Ví dụ:

|  |
| --- |
| doanh\_thu\_theo\_ngay = {  "YYYY-MM-DD": Tổng\_doanh\_thu,  ...  } |

**4. Lưu trữ danh sách hóa đơn (hoa\_don)**

Kiểu dữ liệu: List of dictionaries.

Cấu trúc:

|  |
| --- |
| hoa\_don = [  {  "so\_ban": Số\_bàn,  "tong\_tien": Tổng\_tiền,  "ngay": "YYYY-MM-DD HH:MM"  },  ...  ] |

Ví dụ:

|  |
| --- |
| hoa\_don = [  {"so\_ban": 1, "tong\_tien": 150000, "ngay": "2024-11-15 18:00"},  {"so\_ban": 2, "tong\_tien": 190000, "ngay": "2024-11-15 19:00"}  ] |

# CHƯƠNG 3. TRIỂN KHAI CHƯƠNG TRÌNH

## 3.1. Khởi tạo dữ liệu và các biến chính

**3.1.1. Giải thích vai trò của các biến toàn cục**

Trong chương trình, các biến toàn cục được sử dụng để lưu trữ trạng thái dữ liệu dùng chung cho toàn bộ các chức năng. Chúng bao gồm:

|  |  |
| --- | --- |
| Tên biến | Vai trò |
| Ten\_file | Đường dẫn đến tệp tin chứa thực đơn (MenuFoods.txt). |
| Menu\_do\_an | Lưu danh sách món ăn hiện tại, cấu trúc là từ điển {ID\_món: (tên\_món, giá\_tiền)}. |
| Ban\_an | Lưu trạng thái đặt món của các bàn ăn, cấu trúc là từ điển {số\_bàn: {ID\_món: số\_lượng}}. |
| Doanh\_thu | Tổng doanh thu từ đầu chương trình. |
| Doanh\_thu\_theo\_ngay | Lưu tổng doanh thu theo từng ngày, cấu trúc là từ điển {ngày: doanh\_thu}. |
| Hoa\_don | Danh sách hóa đơn đã thanh toán, mỗi hóa đơn là một từ điển {"so\_ban": ..., "tong\_tien": ..., "ngay": ...}. |

**3.1.2. Cách khởi tạo dữ liệu ban đầu từ tệp tin**

Dữ liệu ban đầu của chương trình được khởi tạo bằng cách đọc từ tệp tin MenuFoods.txt.

Hàm doc\_menu\_tu\_tap\_tin(ten\_file) đảm nhiệm việc này, với chi tiết như sau:

Mục tiêu: Đọc dữ liệu từ tệp MenuFoods.txt và khởi tạo biến menu\_do\_an.

Luồng hoạt động:

* Mở tệp MenuFoods.txt ở chế độ đọc (r) với mã hóa UTF-8.
* Đọc từng dòng, loại bỏ ký tự thừa và tách dữ liệu theo dấu phẩy ,.
* Mỗi dòng dữ liệu được phân tích và lưu vào menu\_do\_an theo cấu trúc:

|  |
| --- |
| {ID\_món: (tên\_món, giá\_tiền)} |

* Nếu tệp không tồn tại hoặc gặp lỗi, trả về từ điển trống và thông báo lỗi.

## 3.2. Quản lý thực đơn

Phần quản lý thực đơn bao gồm các chức năng chính: đọc và ghi thực đơn từ tệp tin, thêm/sửa/xóa món ăn trong thực đơn, và hiển thị thực đơn cho người dùng. Dưới đây là phân tích chi tiết và cách triển khai từng chức năng.

**Hàm đọc thực đơn (doc\_menu\_tu\_tap\_tin)**

Chức năng: Đọc dữ liệu thực đơn từ tệp tin MenuFoods.txt và lưu vào biến menu\_do\_an.

Triển khai:

|  |
| --- |
| def doc\_menu\_tu\_tap\_tin(ten\_file):      menu = {}      try:          with open(ten\_file, "r", encoding="utf-8") as tap:              for line in tap:                  if line.strip():                      parts = line.strip().split(",")                      ID\_mon = int(parts[0])                      ten\_mon = parts[1]                      gia\_tien = int(parts[2])                      menu[ID\_mon] = (ten\_mon, gia\_tien)          return menu      except FileNotFoundError:          print(f"Tệp {ten\_file} không tồn tại.")          return {}      except Exception as e:          print(f"Lỗi khi đọc thực đơn: {e}")          return {} |

**Hàm ghi thực đơn (ghi\_menu\_vao\_tap\_tin)**

Chức năng: Ghi dữ liệu từ biến menu\_do\_an vào tệp tin MenuFoods.txt để lưu trữ.

Triển khai:

|  |
| --- |
| def ghi\_menu\_vao\_tap\_tin(menu, ten\_file):      try:          with open(ten\_file, "w", encoding="utf-8") as tap:              for ID\_mon, (ten\_mon, gia\_tien) in menu.items():                  tap.write(f"{ID\_mon},{ten\_mon},{gia\_tien}\n")          print("Thực đơn đã được cập nhật.")      except Exception as e:          print(f"Lỗi khi ghi thực đơn: {e}") |

## 3.3. Các chức năng thêm, sửa, xóa món ăn

**Quy trình quản lý thực đơn (quan\_ly\_thuc\_don)**

Chức năng:

Cho phép người dùng chọn các thao tác thêm, sửa, xóa món hoặc hiển thị thực đơn.

Triển khai:

|  |
| --- |
|  |

**3.3.1. Thêm món ăn (them\_mon)**

Chức năng: Thêm một món ăn mới vào thực đơn.

Triển khai:

|  |
| --- |
| def them\_mon():      try:          ID\_mon = int(input("Nhập số món (ví dụ: 1, 2, 3, ...): "))          ten\_mon = input("Nhập tên món: ")          gia\_tien = int(input("Nhập giá tiền của món (VND): "))          if ID\_mon in menu\_do\_an:              print("Món này đã tồn tại trong thực đơn.")          else:              menu\_do\_an[ID\_mon] = (ten\_mon, gia\_tien)              ghi\_menu\_vao\_tap\_tin(menu\_do\_an, ten\_file)              print(f"Đã thêm món {ten\_mon} vào thực đơn.")      except ValueError:          print("Dữ liệu nhập không hợp lệ. Vui lòng nhập lại.") |

**3.3.2. Sửa món ăn (sua\_mon)**

Chức năng: Sửa thông tin của một món ăn đã có trong thực đơn.

Triển khai:

|  |
| --- |
| def sua\_mon():      try:          ID\_mon = int(input("Nhập số món cần sửa: "))          if ID\_mon not in menu\_do\_an:              print("Món này không có trong thực đơn.")          else:              ten\_mon = input("Nhập tên món mới: ")              gia\_tien = int(input("Nhập giá tiền mới của món (VND): "))              menu\_do\_an[ID\_mon] = (ten\_mon, gia\_tien)              ghi\_menu\_vao\_tap\_tin(menu\_do\_an, ten\_file)              print(                  f"Đã cập nhật món {ID\_mon} với tên mới: {ten\_mon} và giá mới: {gia\_tien} VND."              )      except ValueError:          print("Dữ liệu nhập không hợp lệ. Vui lòng nhập lại.") |

**3.3.3. Xóa món ăn (xoa\_mon)**

Chức năng: Xóa một món ăn khỏi thực đơn.

Triển khai:

|  |
| --- |
| def xoa\_mon():      try:          ID\_mon = int(input("Nhập số món cần xóa: "))          if ID\_mon not in menu\_do\_an:              print("Món này không có trong thực đơn.")          else:              del menu\_do\_an[ID\_mon]              ghi\_menu\_vao\_tap\_tin(menu\_do\_an, ten\_file)              print(f"Đã xóa món {ID\_mon} khỏi thực đơn.")      except ValueError:          print("Dữ liệu nhập không hợp lệ. Vui lòng nhập lại.") |

**3.3.4. Hiển thị thực đơn**

Hàm hiển thị thực đơn (hien\_thi\_thuc\_don)

Chức năng: In danh sách các món ăn có trong thực đơn cùng giá tiền.

Triển khai:

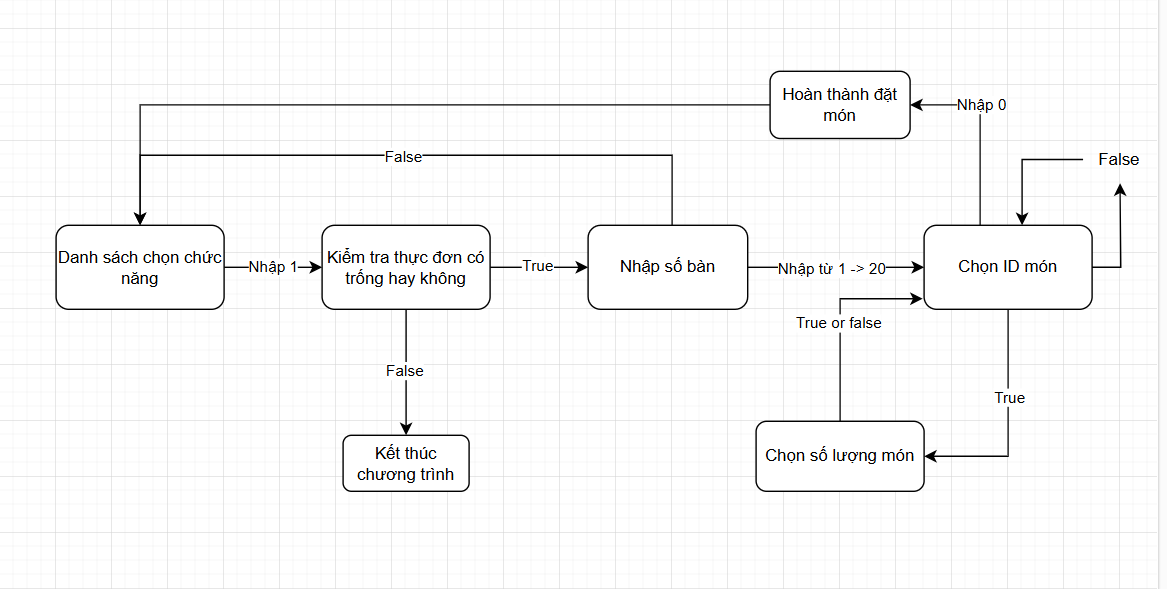
|  |
| --- |
| def hien\_thi\_thuc\_don():      if not menu\_do\_an:          print("Thực đơn hiện tại trống.")      else:          print("\n=== Thực Đơn ===")          for ID\_mon, (ten\_mon, gia\_tien) in menu\_do\_an.items():              print(f"{ID\_mon}. {ten\_mon} - {gia\_tien} VND") |

**Kết quả chạy thử:**

|  |
| --- |
| **Chọn thao tác: 4**  **=== Thực Đơn ===**  **1. Phở Bò - 50000 VND**  **2. Bún Chả - 45000 VND**  **3. Cơm Tấm - 40000 VND**  **4. Gỏi Cuốn - 30000 VND**  **5. Bánh Xèo - 35000 VND** |

# CHƯƠNG 4. TRIỂN KHAI VÀ KẾT QUẢ CỦA CHƯƠNG TRÌNH

## 4.1. Sơ đồ khái quát cách đặt món hoạt động

****

*Mô tả sơ đồ hoạt động của chức năng đặt món*

**Các bước hoạt động chính:**

1. **Danh sách chọn chức năng:**
   * Người dùng bắt đầu với giao diện danh sách các chức năng. Nếu người dùng chọn chức năng đặt món (nhập số 1), hệ thống chuyển sang bước tiếp theo.
2. **Kiểm tra thực đơn:**
   * Hệ thống kiểm tra xem thực đơn có món ăn nào không.
   * **Nếu thực đơn trống:** Chương trình kết thúc.
   * **Nếu thực đơn không trống:** Tiếp tục yêu cầu nhập số bàn.
3. **Nhập số bàn:**
   * Người dùng nhập số bàn muốn đặt món.
   * Số bàn phải hợp lệ (khoảng 1 đến 20). Nếu số bàn không hợp lệ, hệ thống yêu cầu nhập lại.
4. **Chọn ID món ăn:**
   * Người dùng chọn **ID món ăn** từ thực đơn.
   * **Nếu nhập 0:** Kết thúc quá trình đặt món và quay lại bước chọn chức năng.
   * **Nếu ID không hợp lệ:** Yêu cầu nhập lại.
5. **Chọn số lượng món:**
   * Sau khi chọn ID món hợp lệ, người dùng nhập số lượng món muốn đặt.
6. **Xử lý lặp lại:**
   * Sau khi một món được đặt thành công, hệ thống quay lại bước **Chọn ID món ăn**, cho phép người dùng đặt thêm món khác hoặc hoàn tất việc đặt món.
7. **Hoàn thành đặt món:**
   * Khi người dùng nhập **0** ở bước chọn ID món ăn, hệ thống kết thúc quy trình đặt món cho bàn hiện tại và quay lại danh sách chọn chức năng.

**Giải thích các luồng chính trong sơ đồ:**

* **Luồng chính (True):**
  + Xử lý khi thực đơn có sẵn và người dùng nhập đúng thông tin (ID món hợp lệ, số bàn hợp lệ, số lượng hợp lệ).
* **Luồng xử lý lỗi (False):**
  + Xử lý khi:
    - Thực đơn trống.
    - Người dùng nhập sai số bàn.
    - Người dùng nhập sai ID món ăn.
* **Quay lại (Nhập 0):**
  + Người dùng có thể thoát khỏi quá trình đặt món bằng cách nhập **0**.

**Điểm nổi bật trong sơ đồ:**

* Sơ đồ thể hiện rõ tính tuần hoàn và kiểm tra điều kiện ở mỗi bước, đảm bảo người dùng không nhập sai hoặc thiếu thông tin.
* Cấu trúc đơn giản nhưng dễ hiểu, giúp lập trình viên dễ dàng chuyển đổi thành các hàm xử lý cụ thể.

## 4.2. Đặt món ăn

Phần đặt món ăn bao gồm xử lý yêu cầu từ người dùng, quản lý số lượng món đã đặt theo từng bàn, và hiển thị danh sách món ăn đã đặt.

**1. Cách xử lý yêu cầu đặt món từ người dùng**

Khi người dùng yêu cầu đặt món:

1. Người dùng nhập số ID của món ăn và số lượng cần đặt.
2. Kiểm tra xem món ăn có tồn tại trong thực đơn không.
3. Cập nhật thông tin món ăn vào danh sách món đã đặt của bàn ăn tương ứng.

**Triển khai hàm dat\_mon:**

|  |
| --- |
|  |

**Kết quả chạy thử:**

|  |
| --- |
| **Nhập số bàn (1-20): 5**  **=== Đặt Món ===**  **1. Phở Bò - 50000 VND**  **2. Bún Chả - 45000 VND**  **3. Cơm Tấm - 40000 VND**  **4. Gỏi Cuốn - 30000 VND**  **5. Bánh Xèo - 35000 VND**  **Nhập ID món (0 để kết thúc): 1**  **Nhập số lượng món Phở Bò: 2**  **Đã thêm 2 phần Phở Bò vào bàn 5.**  **Nhập ID món (0 để kết thúc): 3**  **Nhập số lượng món Cơm Tấm: 3**  **Đã thêm 3 phần Cơm Tấm vào bàn 5.**  **Nhập ID món (0 để kết thúc): 0**  **Kết thúc đặt món.**  **=== Danh Sách Món Đã Đặt Bàn 5 ===**  **- Phở Bò: 2 phần**  **- Cơm Tấm: 3 phần** |

## 4.3 Thanh toán hóa đơn

1. Ghi hóa đơn vào tệp

Khi thực hiện thanh toán, hóa đơn của bàn sẽ được ghi vào tệp DoanhThu.txt để lưu trữ lịch sử giao dịch.

2. Tính toán tổng tiền và doanh thu theo ngày

Chức năng tính toán tổng tiền cho từng bàn, đồng thời cập nhật doanh thu tổng quát và doanh thu theo ngày.

3. Cập nhật trạng thái bàn ăn

Sau khi thanh toán, trạng thái của bàn được cập nhật, danh sách món ăn đã đặt sẽ được xóa khỏi biến quản lý.

**Triển khai:**

|  |
| --- |
| def xu\_ly\_thanh\_toan():      global doanh\_thu      danh\_sach\_ban = hien\_thi\_ban\_chua\_thanh\_toan()      if not danh\_sach\_ban:          return      try:          so\_ban = int(input("\nNhập số bàn cần thanh toán: "))          if so\_ban not in danh\_sach\_ban:              print(f"Bàn {so\_ban} không có món ăn hoặc không tồn tại.")              return      except ValueError:          print("Vui lòng nhập số bàn hợp lệ.")          return      tong\_tien = 0      ngay\_gio\_hien\_tai = datetime.now().strftime("%Y-%m-%d %H:%M")      if ngay\_gio\_hien\_tai.split(" ")[0] not in doanh\_thu\_theo\_ngay:          doanh\_thu\_theo\_ngay[ngay\_gio\_hien\_tai.split(" ")[0]] = 0      print(f"\n=== Đơn Hàng Bàn {so\_ban} ===")      try:          with open("BTL/DoanhThu.txt", "a", encoding="utf-8") as file:              file.write(f"\n=== Hóa Đơn Bàn {so\_ban} - Ngày {ngay\_gio\_hien\_tai} ===\n")              for ID\_mon, so\_luong in ban\_an[so\_ban].items():                  ten\_mon, gia\_tien = menu\_do\_an[ID\_mon]                  thanh\_tien = gia\_tien \* so\_luong                  tong\_tien += thanh\_tien                  print(f"{ten\_mon} - {so\_luong} phần - {thanh\_tien} VND")                  file.write(                      f"Món: {ten\_mon}, Số lượng: {so\_luong}, Giá: {gia\_tien} VND, Tổng: {thanh\_tien} VND\n"                  )              file.write(f"Tổng tiền: {tong\_tien} VND\n")              print(f"Tổng tiền: {tong\_tien} VND")      except Exception as e:          print(f"Lỗi khi ghi hóa đơn: {e}")      doanh\_thu += tong\_tien      doanh\_thu\_theo\_ngay[ngay\_gio\_hien\_tai.split(" ")[0]] += tong\_tien      hoa\_don.append(          {"so\_ban": so\_ban, "tong\_tien": tong\_tien, "ngay": ngay\_gio\_hien\_tai}      )      ban\_an[so\_ban].clear()      print(f"Đã thanh toán thành công cho bàn {so\_ban}.") |

**Kết quả chạy thử:**

|  |
| --- |
| === Danh Sách Bàn Chưa Thanh Toán ===  - Bàn 5  Nhập số bàn cần thanh toán: 5  === Hóa Đơn Bàn 5 ===  Phở Bò - 2 phần - 100000 VND  Cơm Tấm - 3 phần - 120000 VND  Tổng tiền: 220000 VND  Đã thanh toán thành công cho bàn 5. |

**Hóa đơn ghi vào file:**

|  |
| --- |
| === Hóa Đơn Bàn 5 - Ngày 2024-11-15 14:00 ===  Món: Phở Bò, Số lượng: 2, Giá: 50000 VND, Tổng: 100000 VND  Món: Cơm Tấm, Số lượng: 3, Giá: 40000 VND, Tổng: 120000 VND  Tổng tiền: 220000 VND |

**4.4. Tổng kết doanh thu**

**1. Xem tổng doanh thu theo ngày**

Tính tổng doanh thu từ tất cả các hóa đơn đã thanh toán, chia theo ngày.

**2. Sắp xếp hóa đơn theo thời gian và giá trị thanh toán**

Hiển thị danh sách hóa đơn với các tiêu chí sắp xếp:

* Thời gian: Gần nhất -> Lâu nhất hoặc Lâu nhất -> Gần nhất.
* Giá trị thanh toán: Lớn nhất -> Bé nhất hoặc Bé nhất -> Lớn nhất.

**Triển khai:**

|  |
| --- |
| def tong\_ket\_doanh\_thu():      print("\n=== Tổng Kết Doanh Thu ===")      print("\nDanh sách các hóa đơn đã thanh toán:")      for hoa\_don\_entry in hoa\_don:          print(              f"Bàn {hoa\_don\_entry['so\_ban']} - {hoa\_don\_entry['tong\_tien']} VND - Ngày: {hoa\_don\_entry['ngay']}"          )      print(          f"Doanh thu hôm nay: {doanh\_thu\_theo\_ngay.get(datetime.now().strftime('%Y-%m-%d'), 0)} VND"      )      print("\n1. Sắp xếp theo thời gian (từ gần nhất đến lâu nhất)")      print("2. Sắp xếp theo thời gian (từ lâu nhất đến gần nhất)")      print("3. Sắp xếp theo giá thanh toán (từ lớn nhất đến bé nhất)")      print("4. Sắp xếp theo giá thanh toán (từ bé nhất đến lớn nhất)")      lua\_chon\_sap\_xep = input("Nhập lựa chọn sắp xếp: ")      if lua\_chon\_sap\_xep == "1":          hoa\_don.sort(key=lambda x: x["ngay"], reverse=True)      elif lua\_chon\_sap\_xep == "2":          hoa\_don.sort(key=lambda x: x["ngay"])      elif lua\_chon\_sap\_xep == "3":          hoa\_don.sort(key=lambda x: x["tong\_tien"], reverse=True)      elif lua\_chon\_sap\_xep == "4":          hoa\_don.sort(key=lambda x: x["tong\_tien"])      else:          print("Lựa chọn không hợp lệ.")      for hoa\_don\_entry in hoa\_don:          print(              f"Bàn {hoa\_don\_entry['so\_ban']} - {hoa\_don\_entry['tong\_tien']} VND - Ngày: {hoa\_don\_entry['ngay']}"          ) |

**Kết quả chạy thử:**

|  |
| --- |
| Chọn thao tác: 5  === Tổng Kết Doanh Thu ===  Danh sách các hóa đơn đã thanh toán:  Bàn 1 - 235000 VND - Ngày: 2024-11-18 15:45  Bàn 3 - 215000 VND - Ngày: 2024-11-18 15:47  Bàn 6 - 290000 VND - Ngày: 2024-11-18 15:47  Doanh thu hôm nay: 740000 VND |

## 4.5. Các lựa chọn sắp xếp:

1. Sắp xếp theo thời gian (từ gần nhất đến lâu nhất)

2. Sắp xếp theo thời gian (từ lâu nhất đến gần nhất)

3. Sắp xếp theo giá thanh toán (từ lớn nhất đến bé nhất)

4. Sắp xếp theo giá thanh toán (từ bé nhất đến lớn nhất)

**Lựa chọn sắp xếp 1. Sắp xếp theo thời gian (từ gần nhất đến lâu nhất):**

|  |
| --- |
| Nhập lựa chọn sắp xếp: 1  Bàn 3 - 215000 VND - Ngày: 2024-11-18 15:47  Bàn 6 - 290000 VND - Ngày: 2024-11-18 15:47  Bàn 1 - 235000 VND - Ngày: 2024-11-18 15:45 |

**Lựa chọn sắp xếp 2. Sắp xếp theo thời gian (từ lâu nhất đến gần nhất):**

|  |
| --- |
| Nhập lựa chọn sắp xếp: 2  Bàn 1 - 235000 VND - Ngày: 2024-11-18 15:45  Bàn 3 - 215000 VND - Ngày: 2024-11-18 15:47  Bàn 6 - 290000 VND - Ngày: 2024-11-18 15:47 |

**Lựa chọn sắp xếp 3. Sắp xếp theo giá thanh toán (từ lớn nhất đến bé nhất):**

|  |
| --- |
| Nhập lựa chọn sắp xếp: 3  Bàn 6 - 290000 VND - Ngày: 2024-11-18 15:47  Bàn 1 - 235000 VND - Ngày: 2024-11-18 15:45  Bàn 3 - 215000 VND - Ngày: 2024-11-18 15:47 |

**Lựa chọn sắp xếp 4. Sắp xếp theo giá thanh toán (từ bé nhất đến lớn nhất):**

|  |
| --- |
| Nhập lựa chọn sắp xếp: 4  Bàn 3 - 215000 VND - Ngày: 2024-11-18 15:47  Bàn 1 - 235000 VND - Ngày: 2024-11-18 15:45  Bàn 6 - 290000 VND - Ngày: 2024-11-18 15:47 |

## 4.6. Chức năng xem lại hóa đơn

Chức năng này giúp người dùng xem lại chi tiết các hóa đơn đã được ghi vào tệp DoanhThu.txt. Khi gọi chức năng này, chương trình sẽ mở tệp, đọc các dòng chứa thông tin hóa đơn và hiển thị chúng trên màn hình.

**Triển khai:**

|  |
| --- |
| def xem\_hoa\_don():      try:          with open("BTL/DoanhThu.txt", "r", encoding="utf-8") as file:              content = file.readlines()              if not content:                  print("Không có hóa đơn nào được lưu trong tệp.")                  return              print("\n=== Tất Cả Hóa Đơn ===")              for line in content:                  print(line.strip())      except FileNotFoundError:          print("Tệp DoanhThu.txt không tồn tại.")      except Exception as e:          print(f"Lỗi khi đọc tệp: {e}") |

**Dữ liệu trong file DoanhThu.txt:**

|  |
| --- |
| === Hóa Đơn Bàn 1 - Ngày 2024-11-15 17:29 ===  Món: Phở Bò, Số lượng: 1, Giá: 50000 VND, Tổng: 50000 VND  Món: Bún Chả, Số lượng: 5, Giá: 45000 VND, Tổng: 225000 VND  Món: Bánh Xèo, Số lượng: 8, Giá: 35000 VND, Tổng: 280000 VND  Tổng tiền: 555000 VND  === Hóa Đơn Bàn 2 - Ngày 2024-11-18 07:43 ===  Món: Bún Chả, Số lượng: 2, Giá: 45000 VND, Tổng: 90000 VND  Món: Cơm Tấm, Số lượng: 3, Giá: 40000 VND, Tổng: 120000 VND  Tổng tiền: 210000 VND  === Hóa Đơn Bàn 1 - Ngày 2024-11-18 15:45 ===  Món: Phở Bò, Số lượng: 2, Giá: 50000 VND, Tổng: 100000 VND  Món: Bún Chả, Số lượng: 3, Giá: 45000 VND, Tổng: 135000 VND  Tổng tiền: 235000 VND  === Hóa Đơn Bàn 6 - Ngày 2024-11-18 15:47 ===  Món: Cơm Tấm, Số lượng: 5, Giá: 40000 VND, Tổng: 200000 VND  Món: Bún Chả, Số lượng: 2, Giá: 45000 VND, Tổng: 90000 VND  Tổng tiền: 290000 VND |

**Giải thích**

* Mở tệp và đọc dữ liệu: Chương trình sử dụng open("BTL/DoanhThu.txt", "r", encoding="utf-8") để mở tệp và đọc nội dung của nó. Các dòng được đọc vào danh sách content.
* Kiểm tra tệp rỗng: Nếu tệp không chứa dữ liệu, thông báo "Không có hóa đơn nào được lưu trong tệp." sẽ được hiển thị.
* Hiển thị hóa đơn: Chương trình lặp qua mỗi dòng của tệp và in ra nội dung hóa đơn. line.strip() được sử dụng để loại bỏ các ký tự trắng thừa ở đầu và cuối mỗi dòng.
* Xử lý lỗi: Nếu tệp không tồn tại hoặc có lỗi khi mở tệp, chương trình sẽ thông báo lỗi thích hợp

# CHƯƠNG 5. ĐÁNH GIÁ HỆ THỐNG

## 5.1. Điểm mạnh của chương trình

**Tính đầy đủ:** Chương trình đáp ứng đầy đủ các chức năng cơ bản của hệ thống quản lý nhà hàng như đặt món, thanh toán, quản lý thực đơn, xem doanh thu, và xem lại hóa đơn.

**Dễ sử dụng:** Giao diện dòng lệnh đơn giản, dễ hiểu, người dùng có thể dễ dàng thực hiện các thao tác mà không cần phải có kỹ năng lập trình.

**Khả năng mở rộng:** Chương trình dễ dàng mở rộng, có thể thêm các chức năng mới như tính toán chiết khấu, quản lý nhân viên, thống kê sản phẩm bán chạy, v.v.

**Lưu trữ dữ liệu an toàn**: Dữ liệu được lưu trong các tệp tin đơn giản như MenuFoods.txt và DoanhThu.txt, dễ dàng sao lưu và phục hồi.

## 5.2. Hạn chế và hướng phát triển

**Không có giao diện người dùng đồ họa (GUI):** Hệ thống chỉ sử dụng giao diện dòng lệnh, có thể khó sử dụng với người không quen với các ứng dụng dạng này.

**Thiếu chức năng báo cáo chi tiết:** Chương trình chỉ cho phép xem tổng doanh thu và các hóa đơn, nhưng chưa có chức năng phân tích báo cáo chi tiết như sản phẩm bán chạy nhất theo tháng, theo nhân viên, v.v.

**Quản lý bàn ăn không linh hoạt:** Chương trình chỉ hỗ trợ tối đa 20 bàn ăn. Hệ thống có thể gặp khó khăn nếu số lượng bàn tăng lên hoặc nếu có nhiều chi nhánh.

**Lỗi có thể xảy ra khi thao tác với tệp:** Nếu tệp tin DoanhThu.txt hoặc MenuFoods.txt bị xóa hoặc bị hỏng, chương trình sẽ gặp lỗi và không thể hoạt động.

## 5.3. Hướng phát triển:

**Phát triển giao diện người dùng GUI:** Chuyển từ giao diện dòng lệnh sang giao diện người dùng đồ họa (GUI) sử dụng các thư viện như Tkinter hoặc PyQt để dễ dàng sử dụng hơn.

**Thêm các tính năng phân tích và báo cáo:** Chương trình có thể mở rộng để cung cấp các báo cáo chi tiết hơn về doanh thu, món ăn bán chạy, chiết khấu, v.v.

**Hỗ trợ nhiều chi nhánh và bàn ăn không giới hạn:** Cải thiện hệ thống để có thể quản lý bàn ăn và doanh thu cho nhiều chi nhánh.

**Sử dụng cơ sở dữ liệu:** Thay vì lưu trữ thông tin trong các tệp văn bản, có thể chuyển sang sử dụng cơ sở dữ liệu như SQLite hoặc MySQL để dễ dàng truy vấn và bảo mật dữ liệu hơn.

**TÀI LIỆU THAM KHẢO**

**[1].** [**https://www.w3schools.com/python/python\_dictionaries.asp**](https://www.w3schools.com/python/python_dictionaries.asp)

**[2].** [**https://docs.python.org/3/library/datetime.html**](https://docs.python.org/3/library/datetime.html)

**[3]. Google**

**[4]. Chat GPT**