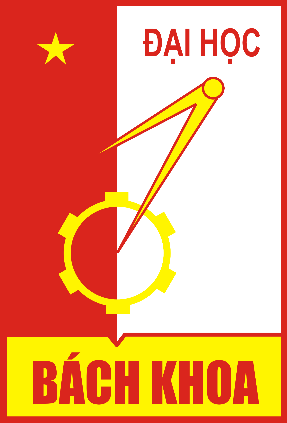


**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

****

BÀI TẬP LỚN MÔN HỌC

**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**

Đề tài: Xây dựng menu điện tử

Giáo viên hướng dẫn: PGS.TS Ngô Quỳnh Thu

Sinh viên thực hiện:

|  |  |
| --- | --- |
| Bùi Đức Quang | 20133067 |
| Phạm Thế Trung | 20134188 |
| Nguyễn Viết Việt | 20134590 |
| Nguyễn Văn Thuyên | 20133847 |

Nhóm 17 - Lớp CNTT2.03-K58

Hà Nội, 12/2015

# Phần 1: Giới thiệu đề tài

## **Tên đề tài**

Tìm hiểu đặc tả yêu cầu, phân tích thiết kế hệ thống và thiết kế một số trường hợp kiểm thử cho menu điện tử phục vụ nhà hàng (đặt món ăn, hủy món ăn, báo nhà bếp, thống kê theo ngày).

## **Mục đích**

Hiện nay, với sự phát triển không ngừng của ngành công nghệ thông tin thì việc áp dụng công nghệ vào kinh doanh nhằm tiết kiệm thời gian, giảm nhân lực, tăng hiệu quả lao động là điểu mà các chủ cửa hàng đang hướng đến. Như mọi người đều biết, thực khách chọn món ăn trên thực đơn, nhân viên phục vụ phải đứng chờ để ghi chép. Hoặc sau khi chọn xong, khách hàng phải chờ kêu phục vụ đến bàn để đọc món ăn mà mình đã chọn. Việc ghi chép thực đơn có thể sai sót dẫn đến tranh cãi sau này. Sau đó, nhân viên phục vụ phải đích thân mang thực đơn xuống nhà bếp mà quãng đường đi lại cũng khá xa. Tại nhà bếp sẽ có người quản lý để phân phối các món ăn cho đầu bếp. Khi món ăn được thực hiện xong, phải gọi cho nhân viên phục vụ để mang thức ăn cho khách. Tất cả công việc này nếu không được tổ chức tốt thì sẽ gây ách tắc, dễ nhầm lẫn, khó kiểm soát và mất thời gian. Hơn nữa, với phương cách này thì khách hàng cũng không thể biết được hiện nay các món ăn của mình đã được thực hiện tới đâu. Mặt khác, mỗi lần khách hàng cần nhân viên phục vụ có khi phải ngóng chờ để được phục vụ. Hệ thống quản lý nhà hàng, áp dụng menu điện tử là một công nghệ mới đang được phát triển tại các nhà hàng lớn, với hình thức mới lạ, nhanh gọn, tiện lợi đã giúp nhà hàng thu hút được nhiều khách hàng. Không những thế, hệ thống còn giúp giảm đáng kể nhân viên phục vụ và tránh được những sai sót không đáng có (ví dụ: nhân viên phục vụ ghi chép nhầm yêu cầu của khách, ….).

## **Mô tả:**

Menu điện tử được cài đặt trong phạm vi nhà hàng. Mỗi bàn ăn sẽ có một màn hình điện tử hiển thị thông tin về số bàn, menu. Khách hàng có xem thông tin các món ăn từ đó có thể yêu cầu đặt món, hủy món phù hợp. Thông tin thực đơn sẽ được gửi tới nhà bếp chuẩn bị và cũng được gửi tới nhân viên thu ngân để lập hóa đơn cho khách.

## **Chi tiết (Nhân sự, Thời gian, Giá thành)**

* Nhân sự:

|  |  |  |  |
| --- | --- | --- | --- |
| Thành viên | MSSV | Sô điện thoại | Email |
| Bùi Đức Quang | 20133067 | 01652375277 | quangbd.hust@mail.com |
| Nguyễn Viết Việt | 20134590 | 0962624101 | Vietbk.succeed@gmail.com |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* Thời gian: 1 tháng

|  |  |
| --- | --- |
| Tuần | Công việc |
| 1 | Tìm hiểu đặc tả yêu cầu cho hệ thống, từng thành viên nêu ý tưởng và trình bày sau đó tổng hợp để đồng bộ ý tưởng, tìm hiểu cho thiết kế hệ thống. |
| 2, 3 | Thiết kế hệ thống |
| 4 | Kiểm thử và hoàn thành sản phẩm. |

* Giá thành:

## **Rủi ro và cách giải quyết**

*Rủi ro:*

* Không thống nhất được ý tưởng chung trong xây dựng ý tưởng hệ thống, mỗi người sẽ có cách suy nghĩ khác nhau đối với hệ thống đề ra.
* Thời gian quá ngắn sẽ khiến cho việc thiết kế hệ thống không được hoàn chỉnh như ý muốn.
* Quá trình kiểm thử không đưa ra được tất cả các trường hợp dễ phát sinh lỗi khi vận hành hệ thống trong thực tế.

*Hướng giải quyết:*

* Tìm hiểu chung nhất về một hướng ý tưởng, tổng hợp từng phần hay nhất để đưa ra một hệ thống logic hợp lý, đảm bảo yêu cầu, tiết kiệm thời gian và chi phí.
* Lựa chọn ngôn ngữ thiết kế và công nghệ áp dụng hợp lý để hệ thống vận hành tốt nhất và dễ hiểu với người dung nhất.
* Vấn đề kiểm thử sẽ tìm hiểu từ những thành viên thiết kế kiểm thử đến lấy ý kiến thực tế để kiểm thử cho việc kiểm thử bao quát được nhiểu vấn đề hơn.

# Phần 2: Đặc tả yêu cầu

## **Giới thiệu hệ thống**

### *Hoạt động nghiệp vụ của menu điện tử*

Theo hoạt động nghiệp vụ của nhà hàng, hoạt động của menu điện tử được tóm tắt như sau:

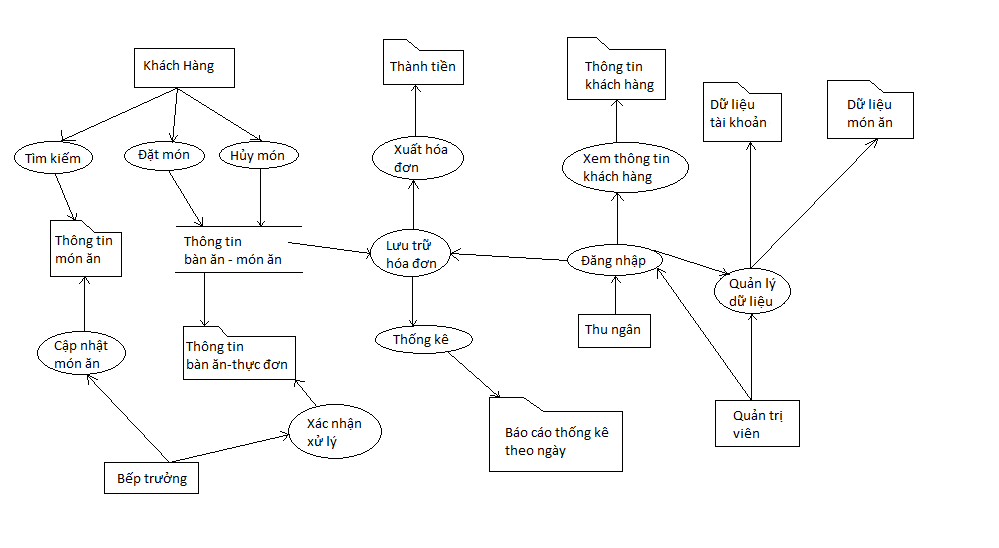
* Nhà hàng cần có một máy chủ và mạng nội bộ kết nối giữa các bộ phận như nhà bếp, yêu cầu từ các máy trạm, quầy thu ngân với nhau.
* Nhà hàng sẽ có một cơ sở dữ liệu để lưu thông tin các món ăn mà nhà hàng có thể đáp ứng được, lưu thông tin hóa đơn nhập, xuất(nhập nguyên liệu).
* Không gian nhà hàng được thiết kế gồm nhiều bàn ăn, mỗi bàn ăn sẽ có số bàn và có màn hình menu điện tử gắn với nó (máy trạm), ở đây khách hàng sẽ có thể đặt món, hủy món và thông tin này sẽ được gửi cho các bộ phận đó là nhà bếp và quầy thu ngân.
* Quầy thu ngân cũng có một màn hình điện tử giúp trực quan hóa cho nhân viên giúp theo dõi khách hàng và thanh toán thống kê về quá trình kinh doanh.
* Nhà bếp cũng có một màn hình điện tử theo dõi các bàn ăn để phục vụ quá trình nấu ăn theo thực đơn của các bàn ăn.
* Khi thanh toán thì nhân viên thu ngân sẽ in hóa đơn từ cơ sở dữ liệu cho khách hàng.

### *Yêu cầu hệ thống*

* + Menu điện tử được xây dựng để phục vụ các yêu cầu sau:
* Giúp khách hàng có thể tra cứu thông tin về các món ăn tại các máy trạm như nguyên liệu làm món ăn, tác dụng của món ăn, giá tiền, ….
* Giúp giảm những thủ tục lằng nhằng của nhân viên phục vụ như ghi thông tin đặt món của khách hàng, chạy vào đưa cho nhà bếp, …. Nhân viên thu ngân chỉ việc thu tiền theo hóa đơn lấy từ hệ thống.
* Giúp bếp trưởng có thể cập nhập lại thông tin các món ăn theo ngày dựa vào nguyên liệu hiện có.
* Thống kê được tiền theo ngày, tháng, năm, ….
  + Các yêu cầu phi chức năng:
* Hệ thống triển khai theo giao diện gần gũi trên mạng nội bộ.
* Thông tin thống kê phải đảm bảo tính chính xác, khách quan.

## **Đặc tả yêu cầu**

* 1. *Biểu đồ luồng dữ liệu*

**

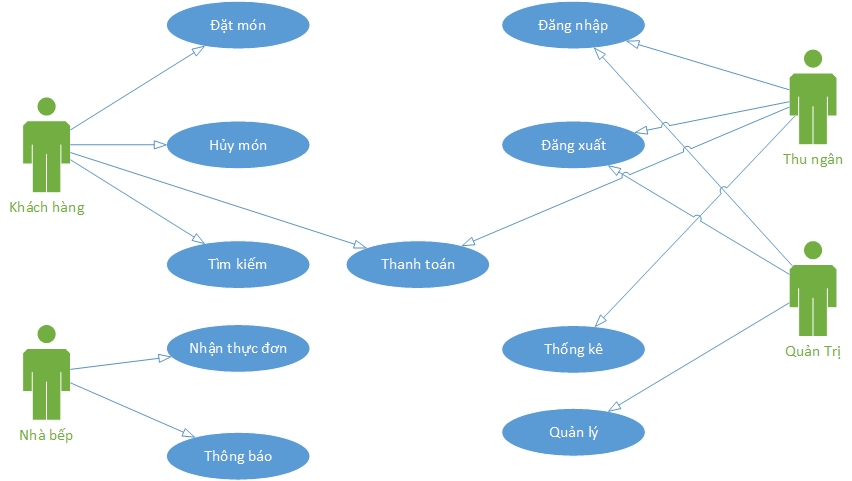
Hình 1. Biểu đồ luồng dữ liệu

Từ những thông tin khái quát đã nêu trên, ta có thể tóm tắt các chức năng và luồng dữ liệu như sau:

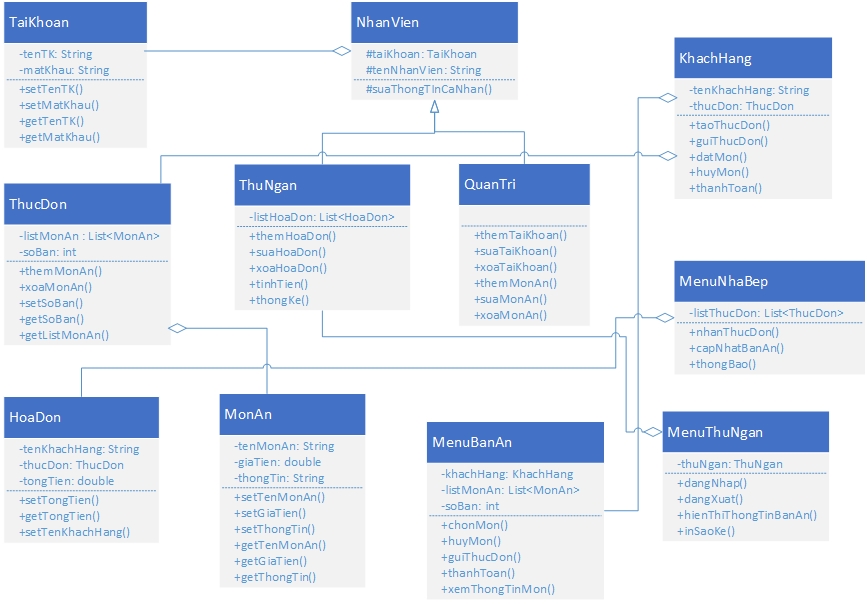
* Khách hàng có thể gửi yêu cầu đặt món, hủy món với hệ thống. Hệ thống sẽ lưu trữ thông tin bàn của khách hàng và các món ăn mà khách hàng đã đặt để xử lý yêu cầu. Ngoài ra khách hàng có thể tìm kiếm các món ăn mà mình ưa thích và hệ thống trả ra thông tin món ăn mà khách hàng đang tìm kiếm.
* Bếp trưởng sau khi nhận được yêu cầu từ khách hàng thông qua hệ thống thì phải xử lý và xác nhận. Bếp trưởng cũng là người phải cập nhập thông tin về các món ăn ở thực đơn theo định kì.
* Thu ngân sau khi nhận được yêu cầu thanh toán từ khách hàng thì sẽ lấy thông tin hóa đơn từ hệ thống và xuất hóa đơn cho khách, thu ngân cần có báo cáo thống kê theo ngày cho chủ nhà hàng để từ đó đưa ra các chiến lược kinh doanh hợp lý hơn. Lưu ý, thu ngân cần phải đăng nhập để thực hiện các chức năng của mình.
* Quản trị hệ thống (là chủ cửa hàng hoặc được chủ cửa hàng ủy quyền), là người quản lý các tài khoản của nhân viên cũng như các thông tin chung của nhà hàng. Người quản trị phải có tài khoản admin để thực hiện các chức năng của mình.

# Phần 3: Phân tích thiết kế

1. **Biểu đồ use-case**

****

1. **Biểu đồ hoạt động**
2. **Biểu đồ lớp**
   1. *Biểu đồ*



* 1. *Phân tích*
  + Lớp TaiKhoan:
* Chức năng:

Là lớp thể hiện một tài khoản giúp nhân viên nhà hàng có thể đăng nhập vào hệ thống của menu điện tử dùng cho hoạt động thu ngân hay hoạt động quản trị hệ thống.

* Thuộc tính:

**tenTK** là thuộc tính thể hiện tên mà người dùng muốn đăng nhập.

**matKhau** là thuộc tính thể hiện mật khẩu của tài khoản người dùng.

* Phương thức:

**setTenTK()** là phương thức giúp cho quá trình tạo tài khoản.

**setMatKhau()** là phương thức giúp cho quá trình tạo mật khẩu.

**getTenTK()** là phương thức giúp lấy về tên tài khoản.

**getMatKhau()** là phương thức giúp lấy về mật khẩu.

* + Lớp ThucDon:
* Chức năng:

Là lớp thể hiện chức năng, vai trò như một thực đơn của khách hàng khi gọi món giống như thực đơn mà khi nhân viên đứng bên cạnh khách hàng để ghi chép vào thực đơn.

* Thuộc tính:

**listMonAn** là một đối tượng của lớp List thể hiện thuộc tính là danh sách các món ăn có trong thực đơn của khách hàng.

**soBan** là thuộc tính kiểu int thể hiện số thứ tự bàn ăn trong nhà hàng.

* Phương thức:

**themMonAn()** là phương thức giúp thêm món ăn vào danh sách thực đơn.

**xoaMonAn()** là phương thức giúp xóa đi một món ăn trong thực đơn.

**getListMonAn()** là phương thức giúp trả về danh sách các món ăn trong thực đơn của khách hàng.

**setSoBan()** là phương thức giúp tạo số bàn có trong thực đơn của khách hàng.

**getSoBan()** là phương thức trả về số thứ tự bàn ăn có trong thực đơn của khách hàng.

* + Lớp HoaDon:
* Chức năng:

Là lớp thể hiện chức năng, vài trò như một hóa đơn sử dụng cho quá trình thanh toán của khách hàng cho nhà hàng.

* Thuộc tính:

**tenKhachHang** là thuộc tính thể hiện tên của khách hàng với hóa đơn này.

**thucDon** là đối tượng của lớp ThucDon thể hiện thực đơn đối với hóa đơn này.

**tongTien** là thuộc tính thể hiện tổng số tiền mà khách hàng phải thanh toán đối với thực đơn này.

* Phương thức:

**setTongTien()** là phương thức giúp tạo số tiền đối với hóa đơn này.

**getTongTien()** trả về tổng số tiền của hóa đơn này.

**getTenKhachHang()** trả về tên khách hàng của hóa đơn này.

* + Lớp MonAn:
* Chức năng:

Là một lớp thể hiện như một món ăn trong thực đơn của nhà hàng.

* Thuộc tính:

**tenMonAn** là thuộc tính kiểu String thể hiện tên của món ăn.

**giaTien** là thuộc tính kiểu double thể hiện giá thành của món ăn này.

**thongTin** là thuộc tính kiểu String thể hiện những thông tin về món ăn như các thành phần hay các công dụng đối với sức khoe của món ăn...

* Phương thức:

**setTenMonAn()** là phương thức giúp tạo tên món ăn.

**getTenMonAn()** là phương thức trả về tên món ăn.

**setGiaTien()** là phương thức tạo giá tiền cho món ăn.

**getGiaTien()** là phương thức trả về giá tiền của món ăn.

**setThongTin()** là phương thức viết các thông tin cho món ăn.

**getThongTin()** là phương thức trả về các thông tin của món ăn.

* + Lớp NhanVien:
* Chức năng:

Là lớp trưu tượng thể hiện như tổng quát về nhân viên dùng làm lớp cha cho việc thể hiện của các lớp nhân viên con.

* Thuộc tính:

**taiKhoan** đối tượng của lớp TaiKhoan là thuộc tính thể hiện tài khoản của nhân viên nhà hàng.

**tenNhanVien** là thuộc tính thể hiện tên của nhân viên nhà hàng.

* Phương thức:

**suaThongTinCaNhan()** là phương thức giúp nhân viên có thể sửa thông tin của mình, tài khoản hoặc tên được lưu trữ trong cơ sở dữ liệu.

* + Lớp ThuNgan:
* Chức năng:

Là lớp được thừa kế của lớp NhanVien có chức năng như một nhân viên thu ngân của nhà hàng, để thanh toán với khách hàng hay thông kê việc kinh doanh của nhà hàng.

* Thuộc tính:

Được thừa hưởng những thuộc tính của lớp NhanVien

**listHoaDon** là đối tượng của lớp list, thuộc tính có chức năng quản lý các hóa đơn của nhà hàng phục vụ cho quá trình thanh toán với khách hàng.

* Phương thức:

**themHoaDon()** là phương thức có chức năng thêm hóa đơn vào trong danh sách hóa đơn của mình quản lý.

**suaHoaDon()** là phương thức giúp sửa những thông tin trong hóa đơn của khách hàng.

**xoasHoaDon()** là phương thức giúp xóa hóa đơn khi đã thông kê hoặc hủy khi không cần dùng đến để tiết kiệm bộ nhớ.

**tinhTien()** là phương thức giúp tính toán số tiền mà khách hàng cần phải thanh toán với mình.

**thongKe()** là phương thức giúp thông kê việc kinh doanh của nhà hàng trong ngày rồi có thể truy xuất tới cơ sở dữ liệu.

* + Lớp QuanTri:
* Chức năng:

Là lớp được thừa kế từ lớp NhanVien đóng vai trò là một nhân viên quản trị về mặt cơ sở dữ liệu của nhà hàng (các tài khoản, các món ăn, ...) có quyên như một admin.

* Thuộc tính:

Được thừa kế từ lớp NhanVien.

* Phương thức:

**themTaiKhoan()** là phương thức giúp tạo và thêm các tài khoản và phân quyền cho các tài khoản, như tài khoản của nhân viên thu ngân hay tài khoản admin.

**suaTaiKhoan()** là phương thức có chức năng sửa các thông tin tài khoản truy xuất tới cơ sở dữ liệu.

**xoaTaiKhoan()** là phương thức có chức năng xóa các tài khoản.

**themMonAn()** là phương thức thêm món ăn vào trong cơ sở dữ liệu.

**suaMonAn()** là phương thức giúp sửa các thông tin của món ăn của nhà hàng trong cơ sở dữ liệu.

**xoaMonAn()** là phương thức giúp xóa đi các món ăn mà nhà hàng không còn kinh doanh nữa.

* + Lớp KhachHang:
* Chức năng:

Là một lớp thể hiện chức năng như một khách hàng đến nhà hàng để ăn uống tham gia vào quá trình kinh doanh.

* Thuộc tính:

**tenKhachHang** là thuộc tính thể hiện tên của khách hàng

**thucDon** là đối tượng của lớp ThucDon thể hiện thực đơn của khách hàng đó.

* Phương thức:

**taoThucDon()** là phương thức để tạo một thực đơn khi khách hàng bắt đầu chọn món.

**guiThucDon()** là phương thức để khách hàng gửi thực đơn cho nhà bếp phục vụ quá trình nấu ăn.

**datMon()** là phương thức mà khi khách hàng đặt món sẽ thêm vào trong thực đơn của khách hàng.

**huyMon()** là phương thức khi khách hàng hủy món ăn nào đó sẽ xóa bỏ trong thực đơn của khách hàng.

**thanhToan()** là phương thức để khách hành liên kết với nhân viên thu ngân để thanh toán với khách hàng.

* + Lớp MenuBanAn:
* Chức năng:

Là một lớp tạo nên giao diện cho từng bàn ăn giúp giao tiếp với khách hàng.

* Thuộc tính:

**khachHang** là đối tượng của lớp KhachHang để thực hiện các chức năng đặt món hủy món cho mình.

**listMonAn** là đối tượng của lớp List thể hiện danh sách các món ăn cho khách hàng chọn và hủy được lấy từ trong cơ sở dữ liệu.

**soBan** là thuộc tính thể hiện số thứ tự của bàn ăn này trong nhà hàng.

* Phương thức:

**chonMon()** là phương thức để đối tượng khách hàng chọn món ăn của mình.

**huyMon()** là phương thức để đối tượng khách hàng hủy các món ăn đã chọn của mình.

**guiThucDon()** là phương thức để đối tượng khách hàng gửi thực đơn đi khi đã chọn xong các món và chắc chắn sẽ dùng thực đơn đó.

**thanhToan()** là phương thức để đối tượng khách hàng gọi quây thu ngân để thanh toán hóa đơn.

**xemThongTinMon()** là phương thức giúp khách hàng xem các thông tin về món ăn mình chọn.

* + Lớp MenuThuNgan:
* Chức năng:

Là một lớp thể hiện một màn hình tại quầy thu ngân để giao tiếp với nhân viên thu ngân.

* Thuộc tính:

**thuNgan** là đối tượng của lớp ThuNgan lớp này sẽ sử dụng đối tượng này để thực hiện các chức năng của nhân viên thu ngân.

* Phương thức:

**dangNhap()** là phương thức giúp nhân viên đăng nhập để sử dụng hệ thống.

**dangXuat()** là phương thức giúp nhân viên đăng xuất khỏi hệ thống.

**hienThiThongTinBanAn()** là phương thức giúp hiển thị các thông tin của một bàn ăn.

**inSaoKe()** là phương thức khi nhân viên thanh toán với khách hàng sẽ in ra hóa đơn và thông kê lại quá trình kinh doanh.

* + Lớp MenuNhaBep:
* Chức năng:

Là một lớp giúp thể hiện một màn hình hiển thị trong nhà bếp để phục vụ các yêu cầu của khách hàng.

* Thuộc tính:

**listThucDon** là đối tượng của lớp List thể hiện danh sách các thực đơn phục vụ cho nhà hàng.

* Phương thức:

**nhanThucDon()** là phương thức để nhận các thực đơn khi khách hàng gửi yêu cầu tới nhà bếp.

**capNhatBanAn()** là phương thức để cập nhận luân phiên việc nhận các thực đơn của nhà hàng tới nhà bếp.

**thongBao()** là phương thức để thông báo các vấn đề tới các bàn ăn của khách hàng.

# Phần 4: Lựa chọn công nghệ

1. Ngôn ngữ thiết kế

Áp dụng ngôn

1. Công nghệ

# Phần 5: Kiểm thử

1. Kiểm thử hộp đen

Các chức năng cần kiểm thử: Đặt món, hủy món, báo nhà bếp, thống kê theo ngày.

Giả sử danh sách các món ăn của nhà hàng gồm: “Bánh bao chiên”, “Lẩu cá riêu hồng”, “Tôm nướng”, …

Số lượng tối đa nhà hàng đáp ứng được cho mỗi món là: 100 xuất.

* 1. Chức năng đặt món

Ta có bảng test-case:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Input | Output | Kết quả |
| 1 | TenMon = “Bánh bao chiên”  SoLuong = 0 | Thông báo: “Mời bạn nhập số lượng” | Đặt món không thành công. |
| 2 | TenMon = “Lẩu cá riêu hồng”  SoLuong = 5 | Thông báo: “Đã cập nhật thực đơn” | Đặt món thành công. |
| 3 | TenMon = “Tôm nướng”  SoLuong = 101 | Thông báo: “Quá số lượng đáp ứng” | Đặt món không thành công. |

* 1. Chức năng hủy món

Giả sử khách hàng đã đặt các món theo số lượng: 6 xuất.

Ta có bảng test-case:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Input | Output | Kết quả |
| 1 | HuyMon = “Bánh bao chiên”  SoLuong = 0 | Thông báo: “Mời bạn nhập số lượng cần hủy” | Hủy món không thành công. |
| 2 | HuyMon = “Lẩu cá riêu hồng”  SoLuong = 2 | Thông báo: “Đã cập nhật thực đơn” | Hủy món thành công. |
| 3 | HuyMon = “Tôm nướng”  SoLuong = 7 | Thông báo: “Quá số lượng đã đặt” | Hủy món không thành công. |
| 3 | HuyMon = “Tôm nướng”  SoLuong = 6 | Thông báo: “Bạn đã xóa bỏ món” | Hủy món không thành công. |

* 1. Chức năng báo nhà bếp

Giả sử khách hàng đã đặt và hủy món thành công.

Ta có bảng test-case:

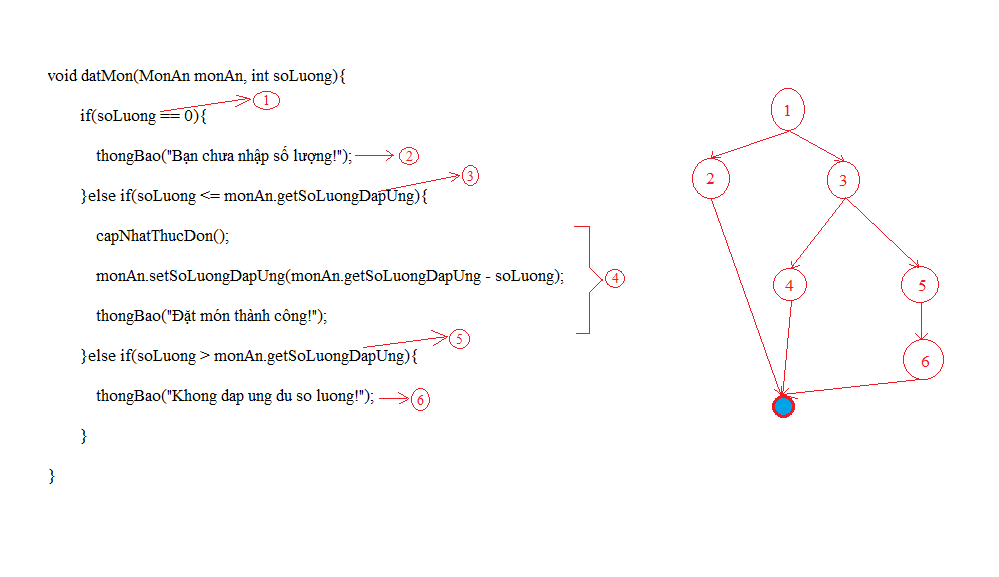
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Input | Output | Kết quả |
| 1 | Thực đơn đã đặt gồm:  Số bàn: 6.  Món: “Bánh bao chiên” SL: 8  Món: “Lẩu cá riêu hồng” SL: 4  Món: “Tôm nướng”  SL: 8 | Hiển thị trên màn hình của nhà bếp:  Số bàn: 6.  Món: “Bánh bao chiên”  SL: 8  Món: “Lẩu cá riêu hồng” SL: 4  Món: “Tôm nướng”  SL: 8 | Báo nhà bếp thành công. |
| 2 | Thực đơn rỗng | Thông báo: “Bạn chưa chọn món” | Báo nhà bếp không thành công. |

* 1. Chức năng thống kê theo ngày

Giả sử cú pháp nhập ngày thống kê là: DD/MM/YYYY

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Input | Output | Kết quả |
| 1 | Nhập: 2015/8/1 | Thông báo: “ Bạn phải nhập đúng cú pháp” | Thống kê không thành công |
| 2 | Nhập: 8/1/2015 | Thông báo: “ Thống kê thành công” | Thống kê thành công |

1. Kiểm thử hộp trắng
   1. Chức năng đặt món



Hình 1: Code chức năng đặt món

Đồ thị trên có 3 đường thi hành tuyến tính độc lập cơ bản là:

* + 1 🡪 2
  + 1 🡪 3 🡪 4
  + 1 🡪 3 🡪 5 🡪 6

Giả sử số lượng đáp ứng của món ăn là 100 xuất.

Thiết kế test-case:

* Test-case cho đường thứ nhất

soLuong = 0 (1)

Kết quả kỳ vọng: đưa ra thông báo “Bạn chưa nhập số lượng”. (2)

* Test-case cho đường thứ 2

soLuong <= 100 (3) với 100 giá trị phù hợp.

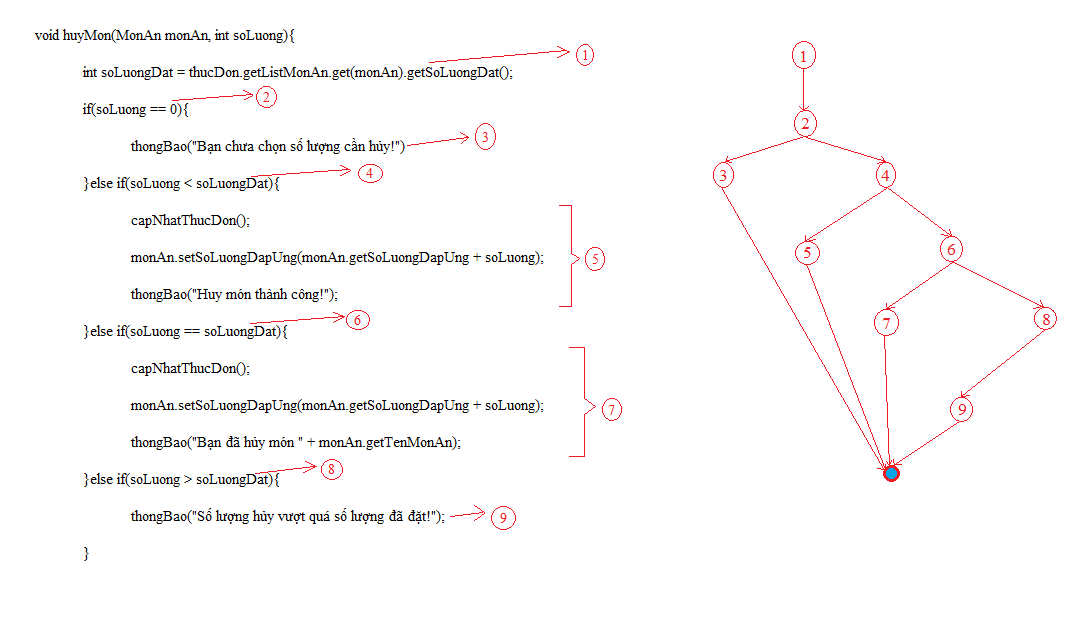
Kết quả kỳ vọng: cập nhập thực đơn ở lớp thực đơn(thêm món ăn yêu cầu và số lượng vào thực đơn). Giảm số lượng đáp ứng ở món ăn một lượng bằng soLuong. Thông báo cho khách hàng “Đặt món thành công”(4)

* Test-case cho đường thứ 3

soLuong > 100 (5)

Kết quả kỳ vọng: đưa ra thông báo “Không đáp ứng được đủ số lượng”

* 1. Chức năng hủy món



Hình 2: Code chức năng hủy món

Đồ thị trên có 4 đường thi hành tuyến tính độc lập:

* 1 🡪 2 🡪 3
* 1 🡪 2 🡪 4 🡪 5
* 1 🡪 2 🡪 4 🡪 6 🡪 7
* 1 🡪 2 🡪 4 🡪 6 🡪 8 🡪 9

Giả sử khách hàng đã đặt 10 xuất.

Thiết kế test-case:

* soLuongDat = 10 (1)
* Test-case cho đường thứ nhất

soLuong = 0 (2)

Kết quả kỳ vọng: đưa ra thông báo “Bạn chưa chọn số lượng cần hủy” (3)

* Test-case cho đường thứ 2

soLuong < 10 (4) có 9 giá trị phù hợp

Kết quả kỳ vọng: cập nhập thực đơn(giảm số lượng món ăn đã đặt đi một lượng bằng soLuong). Số lượng đáp ứng của món ăn tăng lên một lượng bằng soLuong. Đưa ra thông báo “Hủy món thành công”.(5)

* Test-case cho đường thứ 3

soLuong = 10 (6)

Kết quả kỳ vọng: cập nhập thực đơn(xóa món ăn đã đặt ra khỏi thực đơn). Số lượng đáp ứng của món ăn tăng lên một lượng bằng 10. Đưa ra thông báo “Bạn đã hủy món ” + tên món ăn. (7)

* Test-case cho đường thứ 4

soLuong > 10 (8)

Kết quả kỳ vọng: đưa ra thông báo “Số lượng hủy vượt quá số lượng đã đặt” (9).