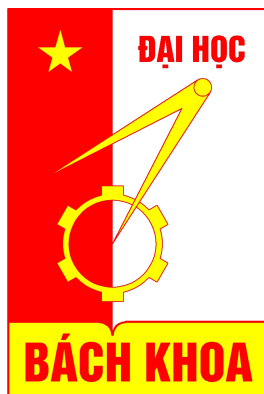


TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÁO CÁO MÔN HỌC

NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

Đề tài: Xây dựng menu điện tử phục vụ nhà hàng

Nhóm sinh viên thực hiện:

Họ và tên	MSSV	Lớp
Đỗ Thanh Bình	20130325	84307
Nguyễn Tuấn Đạt	20130856	84307
Trần Văn Đức	20131070	83444
Đặng Quang Trung	20134145	84307
Phan Anh Tú	20134501	84307

Giáo viên hướng dẫn: TS. Nguyễn Thanh Hùng

Mục lục

Mở đầu	3
1 Giới thiệu Project	4
2 Phân tích và đặc tả yêu cầu	5
2.1 Phân tích yêu cầu phần mềm menu diện tử	5
2.2 Đặc tả	5
2.2.1 Đặc tả chức năng	5
2.2.2 Máy trạng thái	5
2.2.3 Đặc tả dữ liệu	6
3 Thiết kế hệ thống	9
3.1 Biểu đồ Class	10
3.2 Biểu đồ UseCase	11
4 Code	12
4.1 Công nghệ sử dụng	12
5 Kiểm thử	14
5.1 Kiểm thử hộp đen	14
5.1.1 Phân tích đặc tả về các yêu cầu chức năng của modular thu ngân cần thực hiện.	14
5.1.2 Sử dụng kĩ thuật kiểm thử phân lớp tương đương để định nghĩa các testcase của các chức năng trên	14
5.1.3 Kiểm thử các testcase đã định nghĩa	15
5.1.4 So sánh kết quả thu được với kì vọng(lập bảng)	15
5.2 Kiểm thử hộp trắng	17
5.2.1 Đoạn code đem ra kiểm thử	17
5.2.2 Sơ đồ thuật toán và đồ thị chương trình	18
6 Kết luận	21
7 Version Control	22
Tài liệu tham khảo	23

Mở đầu

Công nghệ phần mềm là lĩnh vực được chú trọng, thu hút đông đảo kỹ sư nhất trong ngành công nghệ thông tin hiện nay. Nó đòi hỏi sự sáng tạo trong ý tưởng, nghiêm khắc trong các quy trình, chăm chỉ, bền bỉ trong công việc, và rất nhiều yếu tố khác nữa. Học phần "Nhập môn công nghệ phần mềm" là học phần rất yêu cầu đối với sinh viên, kỹ sư công nghệ phần mềm. Được học học phần này là niềm thú vị đối với chúng em.

Trong quá trình học tập, chúng em đã chọn đề tài bài tập lớn "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)." Nhóm chúng em bao gồm:

Họ và tên	email	Di động
Đỗ Thanh Bình	magic10995@gmail.com	0167 461 1215
Nguyễn Tuấn Đạt	tuandat95cbn@gmail.com	0164 399 5246
Trần Văn Đức	tv.duc95@gmail.com	0169 314 0664
Đặng Quang Trung	20134145	84307
Phan Anh Tú	20134501	84307

Chúng em cảm ơn thầy đã giảng dạy học phần và hướng dẫn chúng em làm bài tập lớn.

Chương 1

Giới thiệu Project

- Đề tài: Xây dựng menu điện tử phục vụ nhà hàng.
- Mục đích: Xây dựng phần mềm cho nhân viên phục vụ, nhà bếp và thu ngân, ứng dụng công nghệ thông tin vào đời sống hàng ngày.
- Mô tả: 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, huỷ món ăn, báo nhà bếp, thống kê theo ngày). Mô hình chi tiết sẽ trong phần đặc tả hệ thống.
- Chi tiết:
 - Nhân sự: đội ngũ kỹ sư, lập trình phần mềm (4 người) và một designer.
 - Thời gian bàn giao sản phẩm dự kiến 3 tháng.
 - Giá thành phần mềm: 50 triệu đồng. Mỗi thiết bị android cho nhân viên: 4 triệu. Server: 10 triệu. PC cho thu ngân: 5 triệu. (Coi như các thiết bị cơ bản wifi, router nhà hàng đã có sẵn).
- Rủi ro và cách giải quyết:
 - Giữ nguồn điện ổn định cho nhà hàng, có giải pháp thay thế khi mất điện.
 - Sẵn sàng cập nhật server khi cơ sở dữ liệu ngày càng lớn dần.
 - Thường xuyên bảo trì hệ thống, sửa lỗi phần mềm.

Chương 2

Phân tích và đặc tả yêu cầu

2.1 Phân tích yêu cầu phần mềm menu điện tử

- **Chức năng:** Đặt món ăn, hủy món ăn, báo nhà bếp, thống kê theo ngày.
- **Phạm vi:** nhà hàng bao gồm các bộ phận: bếp, nhà ăn, quản lý.
- **Kỹ thuật:** Sử dụng mạng lan wifi kết nối giữa các thiết bị, quản lý theo Id.
- **Đối tượng tham gia sử dụng:** Khách hàng, nhân viên phục vụ, nhà bếp và quầy quản lý.
- **Kịch bản:** Khách hàng ngồi tại bàn x đặt món, hủy món qua thiết bị điện tử của nhân viên, nhân viên gửi danh sách đến nhà bếp, nhà bếp bắt đầu làm và check các món đã làm, khách hàng có thể hủy bỏ món, sau khi ăn xong, khách hàng ra quầy quản lý thanh toán. Quầy quản lý có dữ liệu tại bàn x và tính toán, thu tiền của khách.
- **Quy mô:** Nhiều nhân viên, nhiều bếp, nhiều bàn và nhiều quầy quản lý.

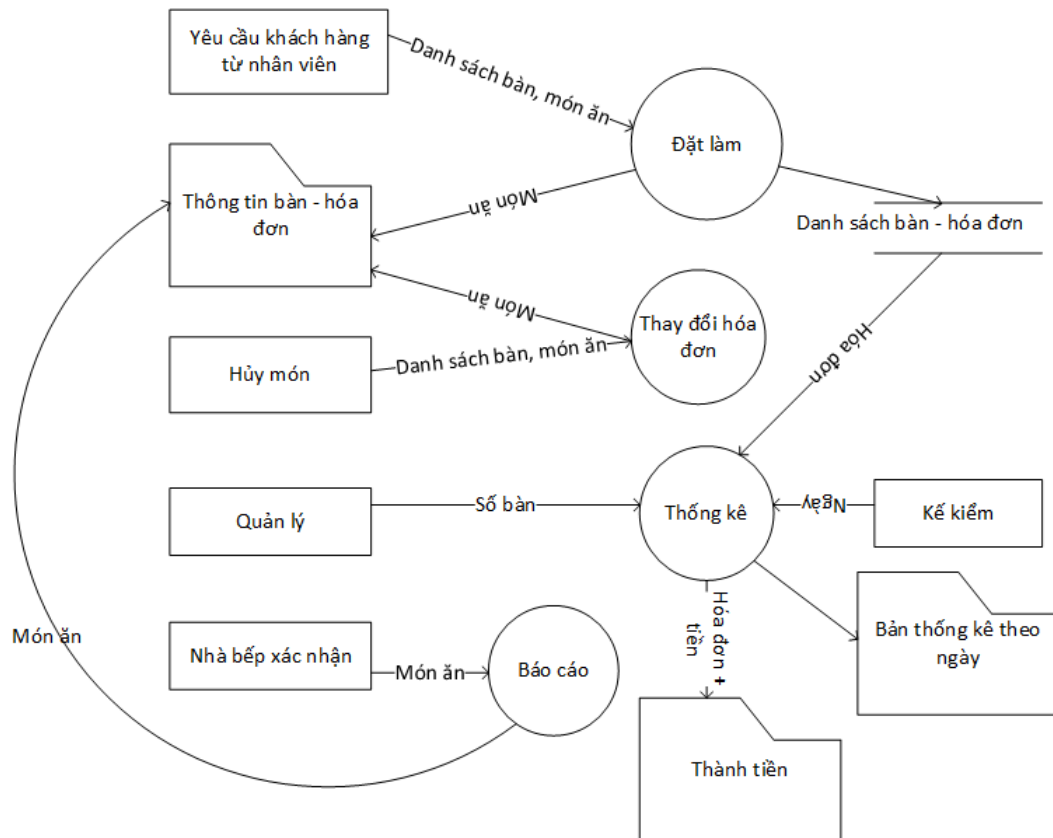
2.2 Đặc tả

2.2.1 Đặc tả chức năng

Biểu đồ luồng dữ liệu:

2.2.2 Máy trạng thái

Sử dụng máy trạng thái để làm rõ nghĩa hơn cho trạng thái đối tượng sử dụng trong sơ đồ



Hình 2.1: Biểu đồ luồng dữ liệu



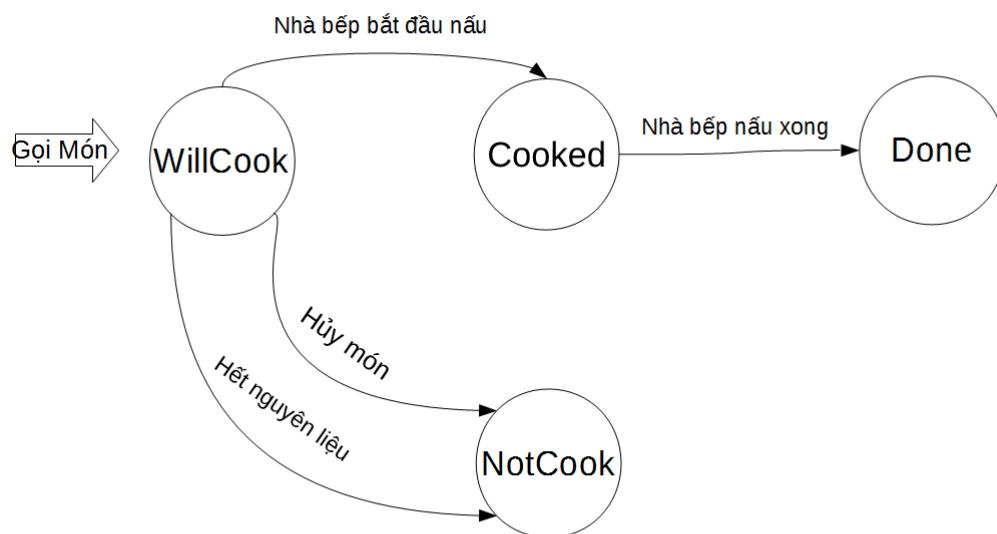
Hình 2.2: Máy trạng thái bàn

2.2.3 Đặc tả dữ liệu

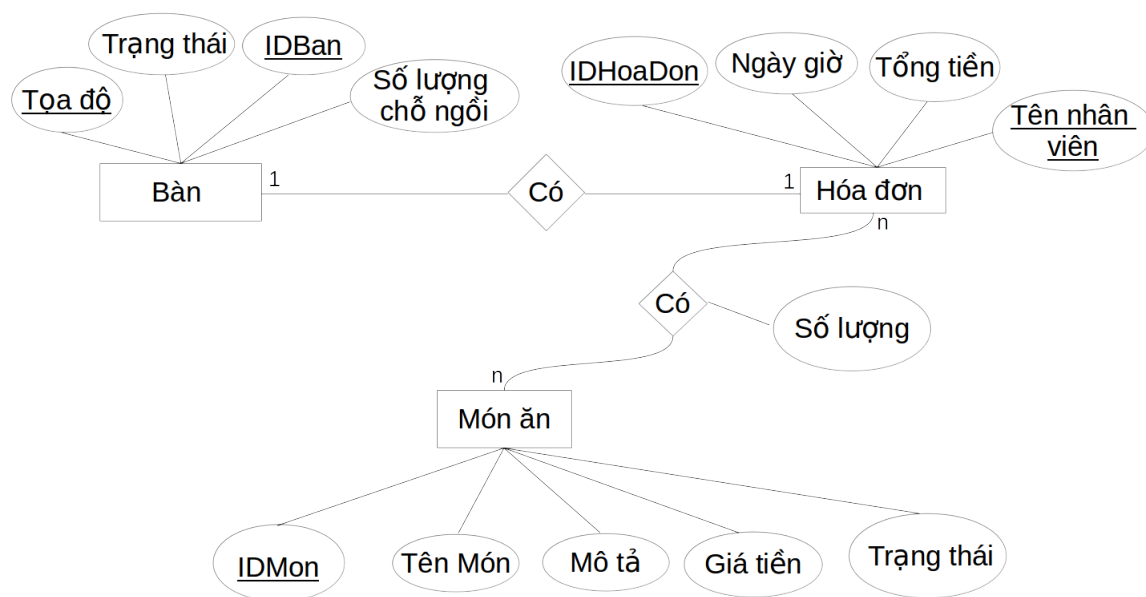
Mô hình thực thể liên kết

Mô hình quan hệ (Bảng lưu trong CSDL)

Ban(IDBan, ToaDo, SoLuongChoNgoi, TrangThai, IDHoaDon)



Hình 2.3: Máy trạng thái món ăn



Hình 2.4: Sơ đồ thực thể liên kết

MonAn(IDMon, TenMon, MoTa, GiaTien, TrangThai)

MonAnTheoBan(IDHoaDon, IDMon, SoLuong)

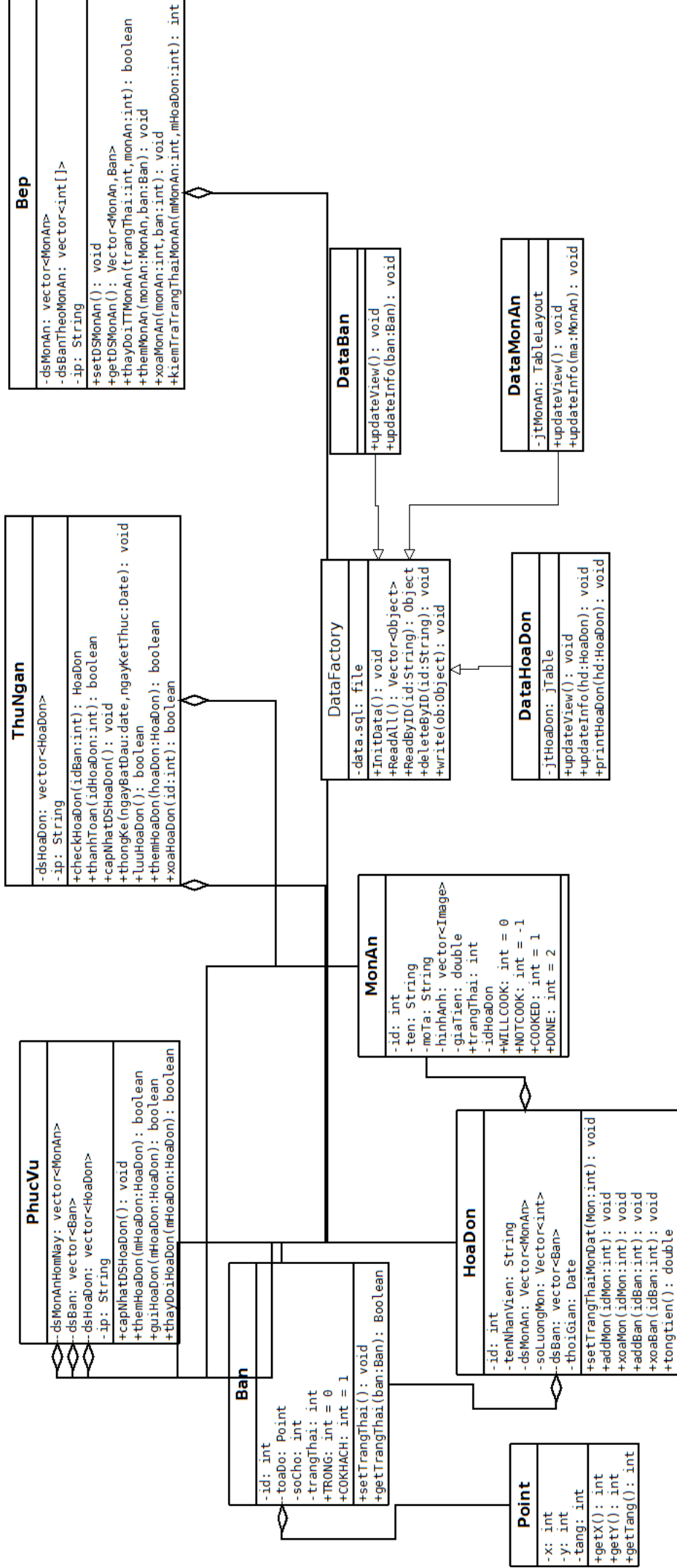
- Bảng **Ban** bao gồm các thuộc tính **IDBan**, **ToaDo** để xác định vị trí của bàn trong nhà hàng; thuộc tính **SoLuongChoNgoi** cho biết bàn có bao nhiêu chỗ để sắp xếp khách cho phù hợp; thuộc tính **TrangThai** cho biết bàn còn trống hay đã có khách (chỉ nhận giá trị 0 hoặc 1 tương ứng với bàn còn trống và đã có khách); thuộc tính **IDHoaDon** là khóa ngoài tham chiếu đến thuộc tính **IDHoaDon** của bảng **HoaDon** để xác định hóa đơn tương ứng với từng bàn.
- Bảng **HoaDon** bao gồm các thuộc tính **IDHoaDon** phân biệt các hóa đơn của các khách khác nhau; thuộc tính **NgayGio** cho biết hóa đơn đó được lập vào ngày nào; thuộc tính **TongTien** cho biết giá trị của hóa đơn (tổng số tiền mà khách hàng phải thanh toán), thuộc tính này có thể tính từ thuộc tính
- **GiaTien** của bảng **MonAn** nhưng để thuận tiện cho việc thống kê (tránh phải truy xuất CSDL nhiều lần) ta tách riêng thành một thuộc tính; thuộc tính **TenNhanVien** cho biết tên nhân viên lập hóa đơn.
- Bảng **MonAn** bao gồm các thuộc tính **IDMonAn**, **TenMon** để xác định tên món ăn, thuộc tính **MoTa** để mô tả chi tiết về món ăn, thuộc tính **GiaTien** cho biết giá của món ăn đó; thuộc tính **TrangThai** cho biết nhà hàng hôm nay có món đó không (chỉ nhận giá trị 0 (tương ứng với không) hoặc 1 (tương ứng với có)).
- Bảng **MonAnTheoBan** liệt kê danh sách các món ăn và số lượng của từng món của 1 bàn

Chương 3

Thiết kế hệ thống

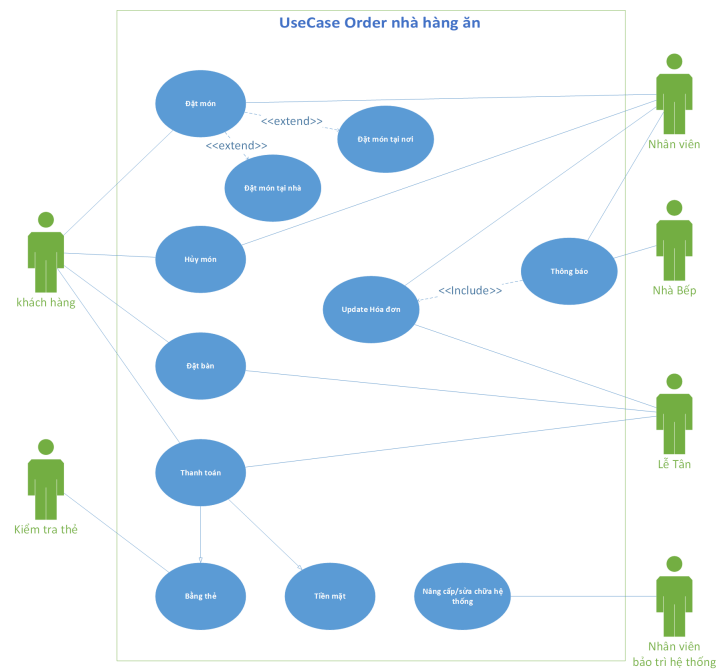
Mô hình hóa phần mềm với UML:

3.1 Biểu đồ Class



Hình 3.1: Class - Diagram

3.2 Biểu đồ UseCase



Hình 3.2: UseCase - Diagram

Chương 4

Code

4.1 Công nghệ sử dụng

- Sử dụng biểu đồ DFD, FSO và thực thể liên kết để phân tích yêu cầu.
⇒ Dễ hiểu, trực quan cho cả khách hàng lẫn đội ngũ kỹ sư.
- Sử dụng biểu đồ Use-case, UML classes để thiết kế hệ thống.
⇒ Đơn giản, quen thuộc và đầy đủ.
- Lập trình bằng ngôn ngữ Java, sử dụng gói Java Swing để thiết kế giao diện cho thu ngân.
⇒ Java là ngôn ngữ hỗ trợ công nghệ hướng đối tượng mạnh, chạy đa nền tảng, phổ biến nhất hiện nay trong việc xây dựng các phần mềm.
⇒ Java Swing hỗ trợ xây dựng giao diện nhanh, hiệu quả, hỗ trợ kéo thả.
- Cơ sở dữ liệu dùng hệ quản trị cơ sở dữ liệu MySQL.
⇒ Đơn giản, chi phí thấp.
- Xây dựng trên nền tảng Android (mỗi nhân viên trạng bị máy tính bảng android để giao tiếp với hệ thống), ngôn ngữ Java trên Android platform.
- Kết nối giữa client và server dựa vào mạng wifi, đặt ip tĩnh cho các thiết bị.
⇒ Nhẹ nhàng, linh động.
Trao đổi thông tin giữa client và server thông qua socket, giao thức TCP.
Socket là một cổng logic nằm giữa process ứng dụng và end-transport protocol. Có hai loại socket thường dùng là Stream Socket (dựa trên giao thức TCP) và Datagram Socket (dựa trên giao thức UDP). Một TCP/IP socket bao gồm một địa chỉ IP và một cổng. Địa chỉ IP dùng để xác định máy tính trên mạng còn cổng được sử dụng để xác định tiến trình.
Ở đây ta sử dụng giao thức TCP để đảm bảo dữ liệu truyền đến nơi nhận một cách đáng tin cậy và đúng thứ tự.
Giải thuật cho client:

- Xác định địa chỉ server
- Tạo socket
- Kết nối đến server
- Gửi nhận dữ liệu
- Đóng kết nối

Giải thuật cho server

- Tạo socket, đăng ký với hệ thống
- Đặt socket ở chế độ chờ, lắng nghe kết nối
- Khi có request từ client, chấp nhận kết nối, tạo một process con để xử lý. Quay lại trạng thái chờ, lắng nghe kết nối mới.

Công việc của process con

- Nhận thông tin kết nối từ client
- Giao tiếp với client
- Đóng kết nối và kết thúc process con

⇒ An toàn, bảo mật, tiện dụng.

Chương 5

Kiểm thử

5.1 Kiểm thử hộp đen

Đối tượng kiểm thử là một modular thu ngân trong hệ thống order phục vụ nhà hàng ăn.

5.1.1 Phân tích đặc tả về các yêu cầu chức năng của modular thu ngân cần thực hiện.

Modular thu ngân là một modular thanh toán tiền cho khách hàng gồm có chức năng chính:

- Kiểm tra hóa đơn của khách hàng thông qua dữ liệu vào là **idban** và trả về hóa đơn tương ứng với bàn đó.
- Thanh toán hóa đơn đầu vào là **idhoadon** và trả về kết quả đã thanh toán hay chưa.

5.1.2 Sử dụng kỹ thuật kiểm thử phân lớp tương đương để định nghĩa các testcase của các chức năng trên

Mỗi lần có khách thanh toán, modular thu ngân sẽ thực hiện các chức năng trên và đưa ra kết quả theo bảng sau:

- Chức năng kiểm tra hóa đơn:
 - Dữ liệu vào là **idban**.
 - Kết quả mong muốn là hóa đơn tương ứng với bàn đó.

ID Bàn	Kết quả mong đợi
<1	Lỗi
1 - 100	Hóa đơn tương ứng với ID bàn.
>100	Lỗi
String	Lỗi

- Chức năng thanh toán:
 - Dữ liệu vào là ID hóa đơn.
 - Dữ liệu đầu ra thanh toán thành công hay chưa(true/false)

ID hóa đơn	Kết quả mong đợi
< 1	Chưa thanh toán thành công
≥ 1	Hóa đơn tương ứng với ID bàn.
String	Chưa thanh toán thành công

5.1.3 Kiểm thử các testcase đã định nghĩa

Với mỗi trường hợp ta lấy đại diện thay vì kiểm tra hết các trường hợp.

- Chức năng kiểm tra hóa đơn: lấy các testcase đại diện sau
 1. test1 { input = 0,output = lỗi }
 2. test2 { input = 1,output = Hóa đơn }
 3. test3 { input = 101,output = lỗi }
 4. test4 { input = String,output = lỗi }
- Chức năng thanh toán: lấy các testcase đại diện sau
 1. test1 { input = 0,output = chưa thanh toán thành công }
 2. test2 { input = 1,output = thanh toán thành công }
 3. test3 { input = String,output = thanh toán chưa thành công }

5.1.4 So sánh kết quả thu được với kì vọng(lập bảng)

- chức năng kiểm tra hóa đơn (KTHD)
- chức năng thanh toán(TT)

Testcase ID	Testcase description	Testcase procedures		Testcase Expected Result	Status
		Step to perform	Step Expected Result		
KTHD1	Kiểm tra lỗi hóa đơn khi idban là số nguyên < 0	Điền giá trị idban $\leq 0(-1)$ vào filed có nhả idban rồi click vào nút kiểm tra	Kết quả thu được màn hình báo lỗi(Không có bàn đó)		
KTHD2	Kiểm tra đưa ra hóa đơn khi idban 1-100	Điền giá trị $0 \leq \text{idban} \leq 100(10)$ vào filed có nhả idban rồi click vào nút kiểm tra	Kết quả thu được màn hình đưa ra 1 hóa đơn tương ứng với bàn đó		
KTHD3	Kiểm tra lỗi hóa đơn khi idban là số nguyên > 100	Điền giá trị idban $> 100(102)$ vào filed có nhả idban rồi click vào nút kiểm tra	Kết quả thu được màn hình báo lỗi(Không có bàn đó)		
KTHD4	Kiểm tra lỗi hóa đơn khi idban là một kí tự hoặc xâu kí tự khác số nguyên	Điền giá trị idban là xâu kí tự hoặc kí tự khác số nguyên(abc)vào filed có nhả idban rồi click vào nút kiểm tra	Kết quả thu được màn hình báo lỗi(Không có bàn đó)		

TT1	Kiểm tra lỗi thanh toán hóa đơn không thành công đầu vào là số nguyên < 1	Ấn nút thanh toán điền vào trường field có nhãn $idhoaddon < 1(0)$ vào rồi click 'Ok'	Kết quả thu được màn hình báo chưa thanh toán thành công		
TT2	Kiểm tra lỗi thanh toán hóa đơn thành công đầu vào là số nguyên ≥ 1	Ấn nút thanh toán điền vào trường field có nhãn $idhoaddon \geq 1(10)$ vào rồi click 'Ok'	Kết quả thu được màn hình báo chưa thanh toán thành công		
TT3	Kiểm tra lỗi thanh toán hóa đơn không thành công đầu vào là xâu hoặc chuỗi kí tự khác số nguyên	Ấn nút thanh toán điền vào trường field có nhãn $idhoaddon$ một xâu hoặc kí tự khác số nguyên(abc) vào rồi click 'Ok'	Kết quả thu được màn hình báo chưa thanh toán thành công		

5.2 Kiểm thử hộp trắng

5.2.1 Đoạn code đem ra kiểm thử

```

public boolean thayDoiTTMonAn(int trangThai) {

    boolean xd=false;

    if (dsMonAn==null) return false;

    if((trangThai<-1) && (trangThai >2)) return false;

    Iterator<MonAn> it = dsMonAn.iterator();

    while(it.hasNext()){

        if (checkTT(it)==false) {

            it.settrangThai(trangThai);
            xd=true
        }
        it.next();
    }
    if (xd==false) System.out.printf("Khong thay doi gi");
}

```

```

public boolean thayDoiTTMonAn(int trangThai) {
    boolean xd=false; ①
    if (dsMonAn==null) return false; ②
    if((trangThai<-1) && (trangThai >2)) return false; ③
    ④
    Iterator<MonAn> it = dsMonAn.iterator(); ⑤
    ⑥
    while(it.hasNext()){ ⑦
        if (checkTT(it)==false) { ⑧
            it.settrangThai(trangThai);
            xd=true ⑨
        }
        it.next(); ⑩
    }
    ⑪
    if (xd==false) System.out.printf("Khong thay doi gi"); ⑫
    return true; ⑬
}

```

Hình 5.1: Phân tích cấu trúc

```

    return true;
}

```

5.2.2 Sơ đồ thuật toán và đồ thị chương trình

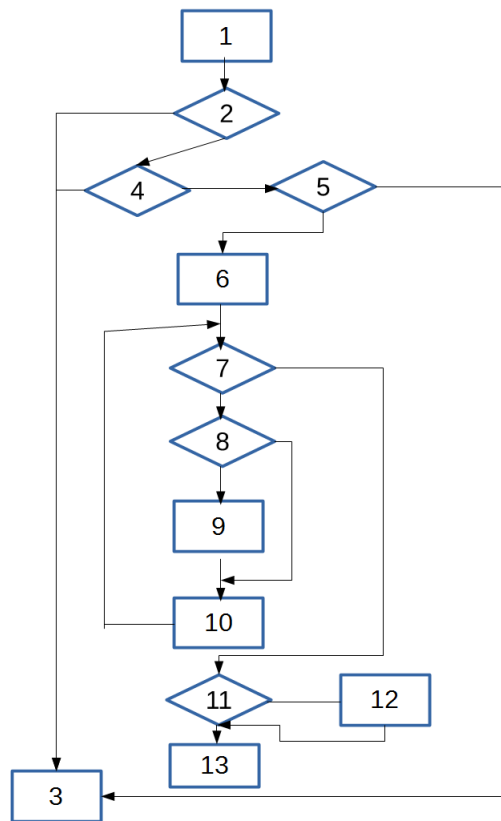
Ta xây dựng sơ đồ thuật toán và đồ thị luồng chương trình.

Dựa vào đồ thị chương trình ta thiết kế các testcase kiểm thử cho modul nhưng theo thực tế code có 2 trường hợp theo đồ thị không thể xảy ra tổng cộng ta có 5 trường hợp kiểm thử cho modul này.

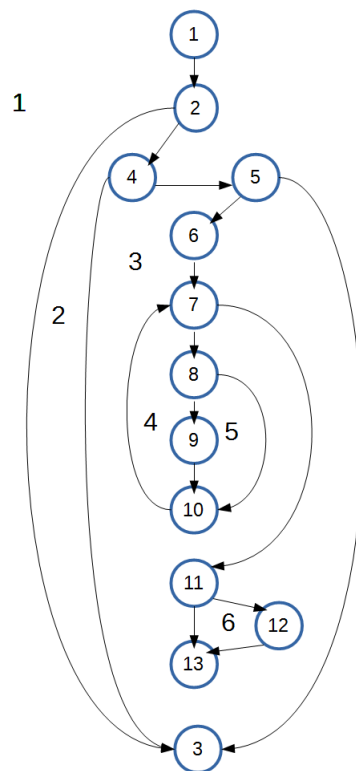
Testcase ID	Testcase description	Testcase procedures		Testcase Expected Result	Status
		Step to perform	Step Expected Result		
HT1	1-2-3	Cho dsMonAn trống	trả về false		
HT2	1-2-4-3	trangThai=-2	trả về false		
HT3	1-2-4-5-3	trangThai 4	trả về false		

HT4	1-2-4-5-6-7-8-10-7-11-13	Trường hợp này không thể xảy ra			
HT5	1-2-4-5-6-7-8-10-7-11-12-13	Không chọn sao cho <code>check(it) = true</code> cho <code>dsMonAn</code> có 2 món	in ra màn hình "không thay đổi gì cả" và trả về true		
HT6	1-2-4-5-6-7-8-9-10-7-11-13	chọn để <code>check(id) = false</code> cho <code>dsMonAn</code> 2 món	trả về true		
HT7	1-2-4-5-6-7-8-9-10-7-11-12-13	Trường hợp không thể xảy ra			

Bảng 5.4: Bảng testcase hộp trắng



Hình 5.2: Sơ đồ thuật toán



Hình 5.3: Đồ thị chương trình

Chương 6

Kết luận

Hệ thống menu điện tử xây dựng còn khá hạn chế, đối với từng mô hình nhà hàng khác nhau cần sửa đổi nhiều về nhân sự, kích thước bàn,...Ngoài ra, trong quá trình code sẽ phát sinh ra rất nhiều trường hợp, tiểu tiết mà chúng em chưa xét đến được.

Qua bài tập lớp này, chúng em đã rèn luyện được kiến thức:

- Làm việc nhóm, kỹ năng quản lý dự án thông qua github.com.
- Bổ trợ kỹ năng phân tích, thiết kế hướng đối tượng.
- Phát triển dự án theo hệ thống, từng bước một: phân tích, đặc tả, thiết kế, lập trình, kiểm thử.
- Sử dụng các công cụ, biểu đồ trong từng khâu thiết kế như DFD, FSO, Use-case, UML classes.

Chương 7

Version Control

<https://github.com/peace195/NMCPNP>

peace195 / NMCPNP

Watch 5 Star 1 Fork 3

Code Issues 0 Pull requests 0 Pulse Graphs

Sort:20151

78 commits 1 branch 0 releases 5 contributors

Branch: master New pull request New file Find file HTTPS https://github.com/peace195/ Download ZIP

tuandat95cbn Merge pull request #9 from tuandat95cbn/master

File	Commit	Time
Baocao	Update	
ChonCongNge	update cong nghe	
PhanTich	bo sung thuc tinh cho cac thuc the	
ThietKe	Sua lai Usecase UseCase1	3 days ago
report	update thu bao cao	2 days ago
README.md	Update README.md	27 days ago
UML-edited.dia	uml update them	2 days ago
UML-edited.png	uml update them	2 days ago
dat.txt	update file .txt	a month ago
duc.txt	Them may trang thai mon an va may trang thai table	28 days ago
trung.txt	chinh sua file DuLieu	27 days ago
tu.txt	Them may trang thai mon an va may trang thai table	28 days ago

No matches were found for "https://github.com/peace195/NMCPNP".

tuandat95cbn committed 2 days ago

Merge pull request #8 from tuandat95cbn/master

update thu bao cao

Sua lai Usecase UseCase1

Merge pull request #7 from tuandat95cbn/master

sua UML

Commits on Jan 5, 2016

bo sung thuc tinh cho cac thuc the

sua UML

Commits on Jan 4, 2016

update uml

update lai

Commits on Jan 3, 2016

Tài liệu tham khảo

[1] Slide "Nhập môn công nghệ phần mềm", TS.Nguyễn Thanh Hùng