

TRƯỜNG ĐẠI HỌC TRÀ VINH  
KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN



**TÊN CHỦ ĐỀ:**

**XÂY DỰNG ỨNG DỤNG BẢN ĐIỆN THOẠI**

**Giáo viên hướng dẫn:**

**Nhan Minh Phúc**

**Sinh viên thực hiện:**

**Nguyễn Thị Thanh Tuyền  
Trần Nhựt Minh  
Trịnh Hải Sơn**

*Trà Vinh, 11/2015*

TRƯỜNG ĐẠI HỌC TRÀ VINH  
KHOA KỸ THUẬT VÀ CÔNG NGHỆ  
BỘ MÔN CÔNG NGHỆ THÔNG TIN



**BÁO CÁO MÔN HỌC**  
**XÂY DỰNG PHẦN MỀM HƯỚNG ĐỐI TƯỢNG**  
**Đề tài:**  
**XÂY DỰNG ỨNG DỤNG BÁN ĐIỆN THOẠI**

**GVHD: Nhan Minh Phúc**

**SVTH: Nhóm 4**

**Nguyễn Thị Thanh Tuyền      110112070**

**Trần Nhựt Minh                110112054**

**Trịnh Hải Sơn                 110112060**

## LỜI MỞ ĐẦU

Trong những năm gần đây với sự phát triển vượt trội của khoa học kỹ thuật đặc biệt là công nghệ thông tin, với những ứng dụng của công nghệ thông tin vào các lĩnh vực đã đóng góp một phần to lớn cho sự nghiệp phát triển của con người. Trong những đóng góp đó phải kể đến các ứng dụng bán hàng, ứng dụng bán hàng đang dần trở thành một phần không thể thiếu của cuộc sống. Ứng dụng bán hàng gắn bó với con người và có ảnh hưởng sâu rộng tới thói quen, sinh hoạt, giải trí của nhiều người.

Cùng với sự phát triển nhanh chóng đó thì các hình thức mua và bán hàng hóa cho mọi người ngày càng đa dạng và phát triển hơn. Các ứng dụng cũng ngày càng đa dạng và trở nên phổ biến. Trước nhu cầu đó cùng với yêu cầu môn học chúng em quyết định chọn đề tài Xây dựng ứng dụng bán điện thoại.

Với đề tài và môn học này, em xin chân thành cảm ơn sự giúp đỡ tận tình của thầy Nhan Minh Phúc. Do còn nhiều hạn chế về thời gian đề tài xây dựng không tránh khỏi những thiếu sót. Rất mong được thầy và các bạn đóng góp ý kiến để chương trình ngày càng hoàn thiện và được đưa vào sử dụng.

## LỜI CẢM ƠN

Trước tiên chúng tôi xin chân thành cảm ơn Ban Giám Hiệu nhà trường, Ban Chủ nhiệm Khoa Kỹ thuật và Công nghệ, Trường Đại học Trà Vinh đã tạo điều kiện cho chúng tôi thực hiện đề tài.

Chúng tôi xin gửi lời cảm ơn chân thành thầy Nhan Minh Phúc người đã tận tình chỉ bảo, hướng dẫn và cung cấp cho chúng tôi những kiến thức quý báu trong suốt thời gian thực hiện đề tài.

Chúng tôi xin cảm ơn chân thành tới các Thầy, Cô Khoa Kỹ thuật và Công nghệ, Trường Đại học Trà Vinh đã nhiệt tình giảng dạy, tạo điều kiện thuận lợi cho chúng tôi trong suốt quá trình học tập cũng như trong thời gian thực hiện đề tài này.

Chúng tôi rất mong nhận được các ý kiến đóng góp của quý Thầy Cô, các chuyên gia có kinh nghiệm và cùng toàn thể các bạn sinh viên để bài báo cáo này càng được hoàn thiện hơn.

*Xin chân thành cảm ơn!*

Nhóm sinh viên thực hiện

**Nhóm 4**

[illegible]

**Trà Vinh, ngày ..... tháng ..... năm 2015**  
**Giáo viên hướng dẫn**

## BẢNG PHÂN CÔNG CÔNG VIỆC

Công việc	Tuần1 (từ 12/10 đến 18/10)	Tuần 2 (từ 19/10 đến 25/10)	Tuần 3 (từ 26/10 đến 01/11)	Tuần 4 (từ 02/11 đến 08/11)	Thực hiện
Khảo sát nhu cầu					Tuyền
Đặc tả vấn đề					Son
Phân tích Use Case					Minh
Lập sơ đồ lớp					Son
Thiết kế CSDL					Son
Thiết kế giao diện					Son
Thiết kế Use case Bán Hàng					Minh
Thiết kế Use case Thanh toán					Minh
Làm bài báo cáo Word					Tuyền
...					

# MỤC LỤC

Chương 1. GIỚI THIỆU .....	1
1.1. Lý do chọn đề tài.....	1
1.2. Mục đích đề tài .....	1
1.3. Động cơ thực hiện.....	2
1.4. Các khía cạnh kỹ thuật để thực hiện dự án .....	2
Chương 2. CƠ SỞ LÝ THUYẾT .....	3
2.1. Phần mềm hướng đối tượng.....	3
2.1.1. Ưu điểm của hướng đối tượng .....	3
2.1.2. Các đặc trưng của hướng đối tượng .....	4
2.2. Giới thiệu ngôn ngữ UML .....	6
2.2.1. UML là gì? .....	6
2.2.2. Lịch sử UML .....	7
2.2.3. Các hướng nhìn của UML.....	8
Chương 3. ĐẶC TẢ VÀ PHÂN TÍCH ỨNG DỤNG .....	11
3.1. Đặc tả yêu cầu.....	11
3.2. Phân tích ứng dụng .....	12
3.2.1. Yêu cầu chức năng .....	13
3.2.2. Yêu cầu chất lượng .....	14
3.2.3. Yêu cầu về nền tảng .....	14
3.3. Phân tích ứng dụng .....	15
3.3.1. Sơ đồ Usecase .....	15
3.3.2. Sơ đồ lớp .....	16
Chương 4. THIẾT KẾ HỆ THỐNG VÀ THỰC HIỆN HÓA USECASE .....	18
4.1 Thiết kế tổng quát .....	18
4.1.1 Mô hình hệ thống .....	18
4.1.2 Thiết kế dữ liệu .....	18
4.1.3 Lựa chọn công nghệ .....	22
4.2 Thiết kế chi tiết .....	22
4.2.1 Usecase đăng nhập .....	22
4.2.2 Usecase đăng ký .....	25
4.2.3 Usecase nhập hàng .....	27
4.2.4 Usecase đổi mật khẩu.....	29
4.2.5 Usecase xóa sản phẩm.....	31
4.2.6 Usecase xóa khách hàng .....	33
4.2.7 Usecase thống kê sản phẩm .....	35
4.2.8 Usecase thống kê nhân viên .....	36
4.2.9 Usecase thống kê hóa đơn.....	37
4.2.10 Usecase thống kê khách hàng .....	39
Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	41
5.1 Kết luận.....	41
5.1.1 Kết quả đạt được .....	41
5.1.2 Hạn chế.....	42
5.2 Hướng phát triển .....	42
TÀI LIỆU THAM KHẢO .....	43

## DANH MỤC HÌNH

Hình 3.1	Lược đồ Usecase tổng quát.....	15
Hình 3.2	Sơ đồ lớp.....	16
Hình 4.1	Mô hình của hệ thống .....	18
Hình 4.2	Mô hình cơ sở dữ liệu.....	18
Hình 4.3	Giao diện của Usecase đăng nhập .....	23
Hình 4.4	Lược đồ lớp của Usecase đăng nhập .....	24
Hình 4.5	Lược đồ tuần tự của Usecase đăng nhập .....	24
Hình 4.6	Giao diện của Usecase đăng ký tài khoản .....	25
Hình 4.7	Lược đồ lớp của Usecase đăng ký .....	26
Hình 4.8	Lược đồ tuần tự của Usecase đăng ký .....	26
Hình 4.9	Giao diện của Usecase nhập hàng .....	27
Hình 4.10	Lược đồ lớp của Usecase nhập hàng.....	28
Hình 4.11	Lược đồ tuần tự của Usecase nhập hàng.....	28
Hình 4.12	Giao diện của Usecase đổi mật khẩu .....	29
Hình 4.13	Lược đồ lớp của Usecase đổi mật khẩu .....	30
Hình 4.14	Lược đồ tuần tự của Usecase đổi mật khẩu .....	30
Hình 4.15	Giao diện của Usecase xóa sản phẩm .....	31
Hình 4.16	Lược đồ lớp của Usecase xóa sản phẩm .....	32
Hình 4.17	Lược đồ tuần tự của Usecase xóa sản phẩm .....	32
Hình 4.18	Giao diện của Usecase xóa khách hàng .....	33
Hình 4.19	Lược đồ lớp của Usecase xóa khách hàng .....	34
Hình 4.20	Lược đồ tuần tự của Usecase xóa khách hàng .....	34
Hình 4.21	Lược đồ lớp của Usecase thống kê sản phẩm .....	35
Hình 4.22	Lược đồ tuần tự của Usecase thống kê sản phẩm .....	35
Hình 4.23	Lược đồ lớp của Usecase thống kê nhân viên.....	36
Hình 4.24	Lược đồ tuần tự của Usecase thống kê nhân viên.....	36
Hình 4.25	Lược đồ lớp của Usecase thống kê hóa đơn .....	37
Hình 4.26	Lược đồ tuần tự của Usecase thống kê hóa đơn.....	38
Hình 4.27	Lược đồ lớp của Usecase thống kê khách hàng .....	39
Hình 4.28	Lược đồ tuần tự của Usecase thống kê khách hàng .....	39
Hình 4.29	Lược đồ trạng thái .....	40
Hình 4.30	Lược đồ hoạt động .....	40



## DANH MỤC BẢNG

Bảng 3.1	Bảng nhu cầu chức năng.....	13
Bảng 3.2	Bảng yêu cầu chất lượng .....	14
Bảng 4.1	Mô tả chi tiết bảng Tb_HangHoa .....	19
Bảng 4.2	Mô tả chi tiết bảng Tb_User.....	20
Bảng 4.3	Mô tả chi tiết bảng Tb_NhanVien .....	20
Bảng 4.4	Mô tả chi tiết bảng Tb_HoaDon.....	21
Bảng 4.5	Mô tả chi tiết bảng Tb_KhachHang .....	21
Bảng 4.6	Mô tả chi tiết bảng Tb_ChiTietHoaDon.....	22

## Chương 1. GIỚI THIỆU

### 1.1. Lý do chọn đề tài

Hiện nay, việc mọi người sử dụng các thiết bị di động không còn quá xa lạ khi mà các công nghệ và dịch vụ di động phát triển từng ngày. Do vậy, các trung tâm mua bán điện thoại di động cũng được khai trương ở mọi nơi. Các trung tâm này có thị trường rất lớn và các nhà sản xuất điện thoại liên tục đưa ra các sản phẩm hấp dẫn khách hàng.

Một trung tâm mua bán điện thoại di động thường có nhiều hình thức: giao dịch, trao đổi,... Nguồn hàng của cửa hàng có thể do các nhà cung cấp, nhà sản xuất hoặc các công ty trong và ngoài nước cung cấp. Đối tượng bán hàng của cửa hàng gồm người tiêu dùng, các cửa hàng bán điện thoại nhỏ... Cửa hàng quản lý rất nhiều các mặt hàng như Microsoft, Samsung, Apple, Sony... Toàn cửa hàng có quầy thu ngân, quầy kế toán, quầy kỹ thuật... Vì vậy cần một chương trình quản lý để giúp nhân viên dễ dàng hơn trong việc quản lý của mình.

### 1.2. Mục đích đề tài

Xây dựng một ứng dụng bán điện thoại tương đối hoàn chỉnh phục vụ có hiệu quả cho người dùng. Đảm bảo mọi chức năng hoạt động tốt, mang lại hiệu quả cao trong quá trình sử dụng.

Việc sử dụng ngôn ngữ mô hình hóa thống nhất UML để xây dựng ứng dụng nhằm mục đích giúp mọi người có thể dễ dàng trao đổi mua bán một cách nhanh chóng. Thông qua ứng dụng người dùng có thể dễ dàng xem thông tin khách hàng, hàng hóa, thêm vào thông tin chính xác, sửa thông tin bị sai và xóa các thông tin không cần thiết, thống kê hàng hóa tồn kho để quản lý xuất nhập hàng được dễ dàng hơn.

**1.3. Động cơ thực hiện**

Nhằm góp phần xây dựng một hệ thống bán điện thoại đơn giản, thân thiện, dễ sử dụng, cho phép người dùng có thể dễ dàng xem thông tin khách hàng, hàng hóa, thêm vào thông tin chính xác, sửa thông tin bị sai và xóa các thông tin không cần thiết.

Chương trình là công cụ quản lý bán hàng hiệu quả nhằm làm đơn giản hóa việc lập các hoá đơn giấy tờ cũng như việc lưu trữ chúng, cung cấp thông tin nhanh chóng và chính xác làm cơ sở cho việc ra quyết định trong kinh doanh, phục vụ hữu hiệu cho việc quản lý và phát triển của cửa hàng.

**1.4. Các khía cạnh kỹ thuật để thực hiện dự án**

- Phân tích hệ thống bằng phương pháp phân tích hướng đối tượng UML.
- Hệ quản trị cơ sở dữ liệu SQL 2008
- Ngôn ngữ lập trình C#

## Chương 2. CƠ SỞ LÝ THUYẾT

### 2.1. Phần mềm hướng đối tượng

Với phương pháp hướng đối tượng, chúng ta chia ứng dụng ra thành nhiều thành phần nhỏ, hay các đối tượng, các đối tượng này tương đối độc lập với nhau. Ứng dụng sẽ được xây dựng bằng cách ghép nối các đối tượng này lại với nhau. Như trong trò chơi xây lâu đài bằng các mẫu gỗ, bước đầu tiên là tạo một vài loại mẫu gỗ cơ bản, từ đó tạo nên các khối xây dựng căn bản, khi đã có các khối xây dựng đó, ta có thể ghép ráp chúng lại với nhau để tạo thành lâu đài. Tương tự như vậy trong phương pháp hướng đối tượng, khi đã xây dựng một số đối tượng căn bản, ta có thể ghép chúng lại với nhau để tạo ứng dụng của mình.

Mô hình hóa và thiết kế hướng đối tượng là một cách suy nghĩ về vấn đề sử dụng các mô hình được tổ chức xung quanh các khái niệm thế giới thực. Cấu trúc nền tảng là đối tượng, nó kết hợp cả cấu trúc dữ liệu và hành vi vào trong một thực thể đơn. Các mô hình hướng đối tượng là có ích cho việc hiểu vấn đề, việc trao đổi với người dùng, mô hình hoá các tổ chức kinh doanh, chuẩn bị tài liệu và thiết kế chương trình cùng cơ sở dữ liệu.

#### 2.1.1. Ưu điểm của hướng đối tượng

Đễ hiểu đối với người sử dụng vì có các thành phần là các đối tượng và khái niệm đời thực.

Đễ nâng cấp hệ thống khi có nhu cầu cần thêm chức năng khác, dễ khắc phục khi có sự cố xảy ra.

Loại bỏ được những dư thừa, trùng lặp trong việc xây dựng ứng dụng.

Một trong những ưu điểm quan trọng bậc nhất của phương pháp phân tích và thiết kế hướng đối tượng là tính tái sử dụng: bạn có thể tạo các thành phần (đối tượng) một lần và dùng chúng nhiều lần sau đó. Giống như việc bạn có thể tái sử dụng các khối xây dựng (hay bản sao của nó) trong một toà lâu đài, một ngôi nhà ở, một con tàu vũ trụ, bạn cũng có thể tái sử dụng các thành phần (đối tượng) căn bản trong các thiết kế hướng đối tượng cũng như

code của một hệ thống kế toán, hệ thống kiểm kê, hoặc một hệ thống đặt hàng.

Phương pháp hướng đối tượng giúp chúng ta xử lý các vấn đề phức tạp trong phát triển phần mềm và tạo ra các thể hệ phần mềm có khả năng thích ứng và bền chắc.

### 2.1.2. Các đặc trưng của hướng đối tượng

#### 1. Đối tượng (Objects)

Khi thiết kế một chương trình theo tư duy hướng đối tượng người ta sẽ không hỏi “vấn đề này sẽ được chia thành những hàm nào” mà là “vấn đề này có thể giải quyết bằng cách chia thành những đối tượng nào”. Tư duy theo hướng đối tượng làm cho việc thiết kế được “tự nhiên” hơn và trực quan hơn. Điều này xuất phát từ việc các lập trình viên cố gắng tạo ra một phong cách lập trình càng giống đời thực càng tốt. Nếu ngoài đời có cái công nông thì khi thiết kế ta cũng bê nguyên cả cái công nông vào trong chương trình, và như vậy chương trình là tập hợp tất cả các đối tượng có liên quan với nhau. Tất cả mọi thứ đều có thể trở thành đối tượng trong OOP, nếu có giới hạn thì đó chính là trí tưởng của bạn. Đối tượng là một thực thể tồn tại trong khi chương trình chạy. Nó có các thuộc tính (attributes) và phương thức (methods) của riêng mình.

#### 2. Lớp (Classes)

Trong khi đối tượng là một thực thể xác định thì lớp lại là một khái niệm trừu tượng. Có thể so sánh lớp như “kiểu dữ liệu còn” đối tượng là “biến” có kiểu của lớp.

#### 3. Sự đóng gói và trừu tượng hóa dữ liệu (Encapsulation & Data Abstraction)

Muốn truy cập vào các thành phần dữ liệu bắt buộc phải thông qua phương thức, và các phương thức này tạo ra một giao diện để đối tượng giao tiếp với bên ngoài. Giao diện này giúp cho dữ liệu được bảo vệ và ngăn chặn những truy cập bất hợp pháp, đồng thời tạo ra sự thân thiện cho người dùng. Ví dụ:

nếu như trong C, một chuỗi được lưu trữ trong một mảng str nào đó, muốn biết độ dài của chuỗi ta phải gọi hàm `strlen()` trong thư viện thì trong C++, nếu str là một đối tượng thuộc lớp string thì tự nó “biết” kích thước của mình, và chỉ cần gọi `str.size()` hoặc `str.length()` là nó sẽ trả về độ dài của chuỗi str. Người dùng

hoàn toàn không cần biết cài đặt chi tiết bên trong lớp string như thế nào mà chỉ cần biết “giao diện” để có thể giao tiếp với một đối tượng thuộc lớp string là ok. Điều này dẫn đến sự trừu tượng hóa dữ liệu. Nghĩa là bỏ qua mọi cài đặt chi tiết và chỉ quan tâm vào đặc tả dữ liệu và các phương thức thao tác trên dữ liệu. Đặc tả về lớp `Công_nông` ở trên cũng là một sự trừu tượng hóa dữ liệu.

#### 4. Sự kế thừa (Inheritance)

Những ý tưởng về lớp dẫn đến những ý tưởng về kế thừa. Trong cuộc sống hàng ngày chúng ta thấy rất nhiều ví dụ về sự kế thừa (tất nhiên là không phải thừa kế về tài sản). Ví dụ: lớp động vật có thể phân chia thành nhiều lớp nhỏ hơn như lớp côn trùng, lớp chim, lớp động vật có vú, không có vú ... blah blah ... hay lớp phương tiện có thể chia thành các lớp nhỏ hơn như xe đạp, xe thô, xe tăng, xích lô, ... Các lớp nhỏ hơn được gọi là lớp con (subclass) hay lớp dẫn xuất (derived class) còn các lớp phía trên gọi là lớp cha (super class) hay lớp cơ sở (base class).

Một nguyên tắc chung là các lớp con sẽ có các đặc điểm chung được thừa hưởng từ các lớp cha mà nó kế thừa. Ví dụ lớp côn trùng và động vật có vú đều sẽ có những đặc điểm chung của lớp động vật. Và do đó ta chỉ cần bổ sung những đặc điểm cần thiết thay vì viết lại toàn bộ code. Điều này giảm gánh nặng cho các lập trình viên và do đó góp phần giảm chi phí sản xuất cũng như bảo trì, nâng cấp phần mềm.

## 5. Tính đa hình và sự quá tải (Polymorphism & Overloading)

Giả sử ta xây dựng một lớp String để “đúc” ra các đối tượng lưu trữ chuỗi ký tự, ví dụ ta có 3 đối tượng s1, s2, s3 thuộc lớp String. Ta muốn thiết kế lớp String sao cho câu lệnh: s3=s1+s2;

Sẽ thực hiện việc nối chuỗi s2 vào đuôi chuỗi s1 rồi gán kết quả cho chuỗi s3. Nếu như vậy công việc lập trình trông sẽ “tự nhiên” hơn. Nhưng thật không may ngôn ngữ lập trình không cung cấp sẵn điều này.

Sử dụng các toán tử (operators) + và = như trên sẽ gây lỗi. Tuy nhiên C++ cung cấp một cơ chế cho phép lập trình viên “định nghĩa lại” các toán tử này để dùng trong các mục đích khác nhau. Việc định nghĩa lại cách sử dụng toán tử được gọi là “quá tải toán tử” (operator overloading). Một số người gọi nó là “nạp chồng toán tử” hay “đa năng hóa toán tử”.

## 2.2. Giới thiệu ngôn ngữ UML

### 2.2.1. UML là gì?

UML là sự kế vị của làn sóng phân tích và thiết kế hướng đối tượng (object-oriented analysis and design - OOA&D) xuất hiện vào cuối những năm 80 và đầu những năm 90. UML là sự thống nhất phương pháp của ba tác giả Booch, Rumbaugh (OMT), và Jacobson, nhưng phạm vi của nó được mở rộng hơn. UML đã qua một tiến trình chuẩn hóa bởi OMG (Object Management Group - OMG) và hiện nay là một chuẩn của OMG.

UML là ngôn ngữ trực quan được dùng trong quá trình phát triển các hệ thống phần mềm, nó là một ngôn ngữ đặc tả hình thức (formal specification language).

- Ngôn ngữ: chức năng của UML cũng là một phương tiện để trình bày và trao đổi tri thức.
- Mô hình hóa: UML dùng các mô hình (các biểu đồ) để trình bày các khía cạnh khác nhau của hệ thống, giúp ta hiểu được các đặc trưng yêu cầu của hệ thống trong mối tương quan với thế giới thực.

- Trực quan hóa hệ thống: Các mô hình của UML phản ánh một cách trực quan các thành phần và các mối quan hệ trong hệ thống, giúp người dùng hình dung hệ thống một cách dễ dàng.
- Làm tài liệu (Documentation) : UML thể hiện được các yêu cầu của hệ thống, quá trình thực hiện, kiểm thử và cả các kế hoạch của dự án... giúp cho việc quản trị, thu thập tài liệu cho dự án.

UML không phải là một ngôn ngữ lập trình trực quan, nhưng những mô hình của nó có thể nối kết với nhiều ngôn ngữ lập trình khác. Tức ta có thể ánh xạ từ một mô hình của UML sang một ngôn ngữ lập trình hướng đối tượng, như C++, Visual Basic, Java... hay ngay cả với các bảng của một cơ sở dữ liệu quan hệ, hoặc với cơ sở dữ liệu hướng đối tượng. Nhiều phần mềm hỗ trợ phân tích và thiết kế hướng đối tượng với UML có khả năng phát sinh mã từ các mô hình của UML.

### **2.2.2. Lịch sử UML**

Những năm đầu của thập kỷ 90 có rất nhiều phương pháp phân tích, thiết kế hệ thống hướng đối tượng và cùng với chúng là các ký hiệu riêng cho từng phương pháp. Số lượng các phương pháp trong khoảng từ 10 đã lên đến gần 50 trong những năm từ 1989 đến 1994. Ba phương pháp phổ biến nhất là OMT (Object Modeling Technique) [James Rumbaugh], Booch91 [Grady Booch] và OOSE (Object-Oriented Software Engineering) [Ivar Jacobson]. Mỗi phương pháp đều có những điểm mạnh và yếu. Như OMT mạnh trong phân tích và yếu ở khâu thiết kế, Booch91 thì mạnh ở thiết kế và yếu ở phân tích. OOSE mạnh ở phân tích các ứng xử, đáp ứng của hệ thống mà yếu trong các khâu khác.


Do các phương pháp chưa hoàn thiện nên người dùng rất phân vân trong việc chọn ra một phương pháp phù hợp nhất để giải quyết bài toán của họ. Hơn nữa, việc các ký hiệu khác nhau của các phương pháp đã gây ra những sự mập mờ, nhầm lẫn khi mà một ký hiệu có thể mang những ý nghĩa khác nhau trong mỗi phương pháp. Ví dụ như một hình tròn được tô đen biểu hiện một multiplicity trong OMT lại là một aggregation trong Booch). Thời kỳ này còn được biết đến với tên gọi là cuộc chiến giữa các phương pháp. Khoảng đầu năm 94, Booch đã cải tiến phương pháp của mình trong đó có ứng dụng những ưu điểm của các phương pháp của



Rumbaugh và Jacobson. Tương tự Rumbaugh cũng cho đăng một loạt các bài báo được biết đến với tên gọi phương pháp OMT-2 cũng sử dụng nhiều ưu điểm của phương pháp của Booch. Các phương pháp đã bắt đầu hợp nhất, nhưng các kí hiệu sử dụng ở các phương pháp vẫn còn nhiều điểm khác biệt.

Cuộc chiến này chỉ kết thúc khi có sự ra đời của UML – một ngôn ngữ mô hình hóa hợp nhất. Bằng cách hợp nhất các kí hiệu sử dụng trong khi phân tích, thiết kế của các phương pháp đó, UML cung cấp một nền tảng chuẩn trong việc phân tích thiết kế. Có nghĩa là các nhà phát triển vẫn có thể tiến hành theo phương pháp mà họ đang sử dụng hoặc là có thể tiến hành theo một phương pháp tổng hợp hơn (do thêm vào những bước ưu điểm của từng phương pháp). Nhưng điều quan trọng là các ký hiệu giờ đây đã thống nhất và mỗi ký hiệu chuẩn của tổ chức OMG (Object Management Group) vào tháng 7-1997.

### 2.2.3. Các hướng nhìn của UML

 Hướng nhìn (View): Hướng nhìn chỉ ra những khía cạnh khác nhau của hệ thống cần phải được mô hình hóa. Một hướng nhìn không phải là một bản vẽ, mà là một sự trừu tượng hóa bao gồm một loạt các biểu đồ khác nhau. Chỉ qua việc định nghĩa của một loạt các hướng nhìn khác nhau, mỗi hướng nhìn chỉ ra một khía cạnh riêng biệt của hệ thống, người ta mới có thể tạo dựng nên một bức tranh hoàn thiện về hệ thống.

Mỗi một hướng nhìn được miêu tả trong một loạt các biểu đồ, chứa đựng các thông tin nêu bật khía cạnh đặc biệt đó của hệ thống. Trong thực tế khi phân tích và thiết kế rất dễ xảy ra sự trùng lặp thông tin, cho nên một biểu đồ trên thực tế có thể là thành phần của nhiều hướng nhìn khác nhau. Khi nhìn hệ thống từ nhiều hướng nhìn khác nhau, tại một thời điểm có thể người ta chỉ tập trung vào một khía cạnh của hệ thống. Một biểu đồ trong một hướng nhìn cụ thể nào đó cần phải đủ độ đơn giản để tạo điều kiện giao tiếp dễ dàng, để dính liền với các biểu đồ khác cũng như các hướng nhìn khác, làm sao cho bức tranh toàn cảnh của hệ thống được miêu tả bằng sự kết hợp tất cả các thông tin từ tất cả các hướng nhìn. Một biểu đồ chứa các kí


hiệu hình học mô tả các phần tử mô hình của hệ thống. UML có tất cả các hướng nhìn sau:

- Use case View: đây là hướng nhìn chỉ ra khía cạnh chức năng của một hệ thống, nhìn từ hướng tác nhân bên ngoài, nêu lên chức năng tổng quát của hệ thống.
- Logical View - hướng nhìn logic: chỉ ra chức năng sẽ được thiết kế bên trong hệ thống như thế nào, qua các khái niệm về cấu trúc tĩnh cũng như ứng xử động của hệ thống.
- Process View: xử lý các vấn đề liên quan đến giao tiếp và đồng bộ hóa trong hệ thống.
- Deployment view - Hướng nhìn triển khai: Thể hiện các vấn đề liên quan đến việc triển khai hệ thống, sơ đồ triển khai về mặt vật lý của hệ thống.
- Implementation View : Thể hiện các vấn đề liên quan đến việc tổ chức các thành phần trong hệ thống.

✚ Biểu đồ (Diagram): Biểu đồ là các hình vẽ miêu tả nội dung trong một hướng nhìn. UML có tất cả 9 loại biểu đồ khác nhau được sử dụng trong những sự kết hợp khác nhau để cung cấp tất cả các hướng nhìn của một hệ thống. Bao gồm:

- Use Case Diagrams - Biểu đồ Use Case.
- Class Diagrams - Biểu đồ lớp.
- Object Diagrams - Biểu đồ đối tượng.
- State Diagrams - Biểu đồ trạng thái.
- Sequence Diagrams - Biểu đồ tuần tự.
- Collaboration Diagrams - Biểu đồ cộng tác.
- Activity Diagrams - Biểu đồ hoạt động.
- Component Diagrams - Biểu đồ thành phần.

- Deployment Diagrams - Biểu đồ triển khai.

 **Phần tử mô hình (Model element):** Các khái niệm được sử dụng trong các biểu đồ được gọi là các phần tử mô hình, thể hiện các khái niệm hướng đối tượng quen thuộc. Ví dụ như lớp, đối tượng, thông điệp, cũng như các quan hệ giữa các khái niệm này, bao gồm cả liên kết, phụ thuộc, khái quát hóa. Một phần tử mô hình thường được sử dụng trong nhiều biểu đồ khác nhau, nhưng nó luôn luôn có chỉ một ý nghĩa và một ký hiệu.

## Chương 3. ĐẶC TẢ VÀ PHÂN TÍCH ỨNG DỤNG

### 3.1. Đặc tả yêu cầu

#### Mô tả qui trình các công việc liên quan đến đề tài

Các hoạt động chính của cửa hàng gồm nhập hàng vào kho, xuất hàng từ kho ra quầy, quản lý các quầy, báo cáo số lượng và doanh thu hàng bán, báo cáo tồn kho trong tháng. Việc quản lý công việc của cửa hàng được phân cấp quản lý theo từng bộ phận như sau:

**Cửa hàng trưởng:** Là người chịu trách nhiệm trước công ty về hoạt động kinh doanh và kế hoạch thực hiện công việc của siêu thị.

**Nhân viên bán hàng:** Là nhân viên của công ty trực tiếp bán hàng đến người dùng. Nhân viên này có nhiệm vụ quản lý các mặt hàng trên quầy mình phụ trách, cuối ca phải kiểm tra và báo cáo số lượng hàng tồn trên quầy để giao ca.

**Bộ phận thu ngân:** Tính tiền của khách hàng, mọi số lượng bán ra phải được nhập vào máy tính để cuối ca nhân viên quản lý sẽ thu lại dữ liệu của họ đã nhập vào.

**Bộ phận quản lý:** Có nhiệm vụ tổng kết số liệu do bộ phận bán hàng và thu ngân chuyển đến, cân đối số lượng hàng hóa trên quầy và số lượng bán ra để tìm ra sai sót giữa hai khâu này nếu có. Sau đó số liệu sẽ chuyển sang cho bộ phận kế toán để tính doanh thu.

**Thủ kho:** Quản lý công việc xuất nhập hàng hóa của cửa hàng.

**Bộ phận kế toán:** Thực hiện điều chỉnh giá các mặt hàng và phân tích số liệu xuất nhập, cố vấn cho cửa hàng trưởng các kế hoạch thực hiện mới.

**Bộ phận kỹ thuật:** Thực hiện việc kiểm tra điện thoại cho khách hàng khi có yêu cầu và làm công việc bảo hành cho khách hàng. Việc mất mát hàng hóa và người chịu trách nhiệm sẽ được phát hiện ngay nhờ sự phân công rạch ròi từng người, từng bộ phận và nhờ vào số liệu mà bộ phận quản lý thu được từ bộ phận thu ngân và bộ phận bán hàng. Mỗi nhân viên tùy theo nhiệm vụ của mình chỉ được xem báo cáo liên quan đến công việc, mà không được xem chương trình của người khác.

### 3.2. Phân tích ứng dụng

**Nhập hàng:** Khi có nhu cầu nhập một số hàng hóa mới, nhân viên kinh doanh cần lập một đơn hàng nhập gồm mã đơn hàng nhập, tên mặt hàng cần nhập và các thông tin về mặt hàng như nhà sản xuất, số lượng, giá nhập. Sau khi hàng được nhập về kiểm tra các thông tin của mặt hàng và cần lưu trữ các thông tin về mặt hàng đã nhập vào kho dữ liệu. Đơn hàng nhập cũng được lưu trữ để thuận tiện cho việc thanh toán.

**Xuất hàng:** Khi nhận được yêu cầu mua hàng của khách thì nhân viên kinh doanh thêm mới và cập nhật các thông tin của khách hàng vào cơ sở dữ liệu để tiện liên lạc làm ăn, đồng thời cần lập một đơn hàng xuất gồm mã đơn hàng xuất, tên khách hàng, địa chỉ, tên mặt hàng bán, số lượng, giá bán. Cuối cùng đơn hàng được lưu vào kho dữ liệu.

**Bán hàng:** Bộ phận bán hàng chuyển hàng từ kho ra quầy bán. Quầy hàng là nơi trưng bày nhiều mặt hàng cùng chủng loại. Cách tổ chức này giúp nhân viên bán hàng dễ kiểm soát hàng hóa ở quầy để bổ sung khi hết hàng. Hàng ngày, căn cứ vào tình hình bán hàng của công ty, bộ phận nghiệp vụ sẽ thống kê để biết được mặt hàng nào bán chạy, mặt hàng nào bán không chạy, mặt hàng nào đã hết hoặc còn ít trong kho, từ đó đưa ra yêu cầu nhập thêm hàng mới, hoặc xuất hàng từ kho ra quầy đối với mặt hàng quầy còn ít.

**Thanh toán:** Các nhân viên kế toán dựa trên các đơn hàng được lưu trữ trong cơ sở dữ liệu gồm cả đơn hàng nhập và đơn hàng xuất. Thanh toán đơn hàng nhập được thực hiện với nhà sản xuất trong đó sẽ tính toán và đưa ra các số liệu cần thanh toán đối với nhà thanh toán như số lượng hàng, đơn giá, số tiền đã thanh toán, số tiền còn nợ nhà sản xuất. Việc thanh toán đơn hàng xuất được thực hiện tương tự với khách hàng nhưng việc thanh toán tiền sẽ khác ở chỗ khách hàng lẻ thì không được nợ, còn các khách hàng là đối tác làm ăn thường xuyên thì có thể nợ.

**Thống kê:** Theo thời gian định kỳ, việc kinh doanh cần thống kê lại các số liệu về hàng hóa cũng như các hóa đơn nhập xuất, sau đó đưa ra những con số cụ thể như số lượng hàng tồn kho, số dư nợ với nhà sản xuất, số dư nợ của khách hàng quen, dựa trên các số liệu thống kê được đưa ra tình hình kinh doanh trong thời gian

đó. Dựa trên các số liệu thống kê, và tình hình kinh doanh trong một khoảng thời gian định kỳ để lập các báo cáo về tình trạng kinh doanh cho người quản lý.

**Tìm kiếm:** chức năng tìm kiếm cho phép tìm kiếm mặt hàng và tìm kiếm khách hàng. Khi tìm kiếm mặt hàng có thể tìm kiếm theo tên mặt hàng, tên nhà sản xuất hoặc đặc điểm của mặt hàng. Khi tìm kiếm khách hàng có thể tìm kiếm theo tên khách hàng.

### 3.2.1. Yêu cầu chức năng

ID	Nhu cầu chức năng
R1	Đăng nhập, đăng xuất, giúp cho nhân viên bán hàng tính tiền
R2	Lưu trữ thông tin khách hàng
R3	Lưu trữ thông tin các nhân viên
R4	Lưu trữ thông tin hàng hóa
R5	Thống kê
R6	In hóa đơn

**Bảng 3.1**      Bảng nhu cầu chức năng

**3.2.2. Yêu cầu chất lượng**

ID	Yêu cầu chất lượng
Q1	Phân quyền đối với người dùng
Q2	Giao diện trực quan, dễ sử dụng
Q3	Dễ dàng cập nhật hệ thống khi có thay đổi về các đối tượng thông tin
Q4	Dễ dàng sao lưu, phục hồi dữ liệu
Q5	Không gây mất mát về dữ liệu khi có sự cố hệ thống
Q6	Dễ bảo trì, nâng cấp khi có nhu cầu
Q7	Luôn đảm bảo tính truy cập

**Bảng 3.2**      Bảng yêu cầu chất lượng**3.2.3. Yêu cầu về nền tảng**

Cấu hình tối thiểu cho thiết bị:

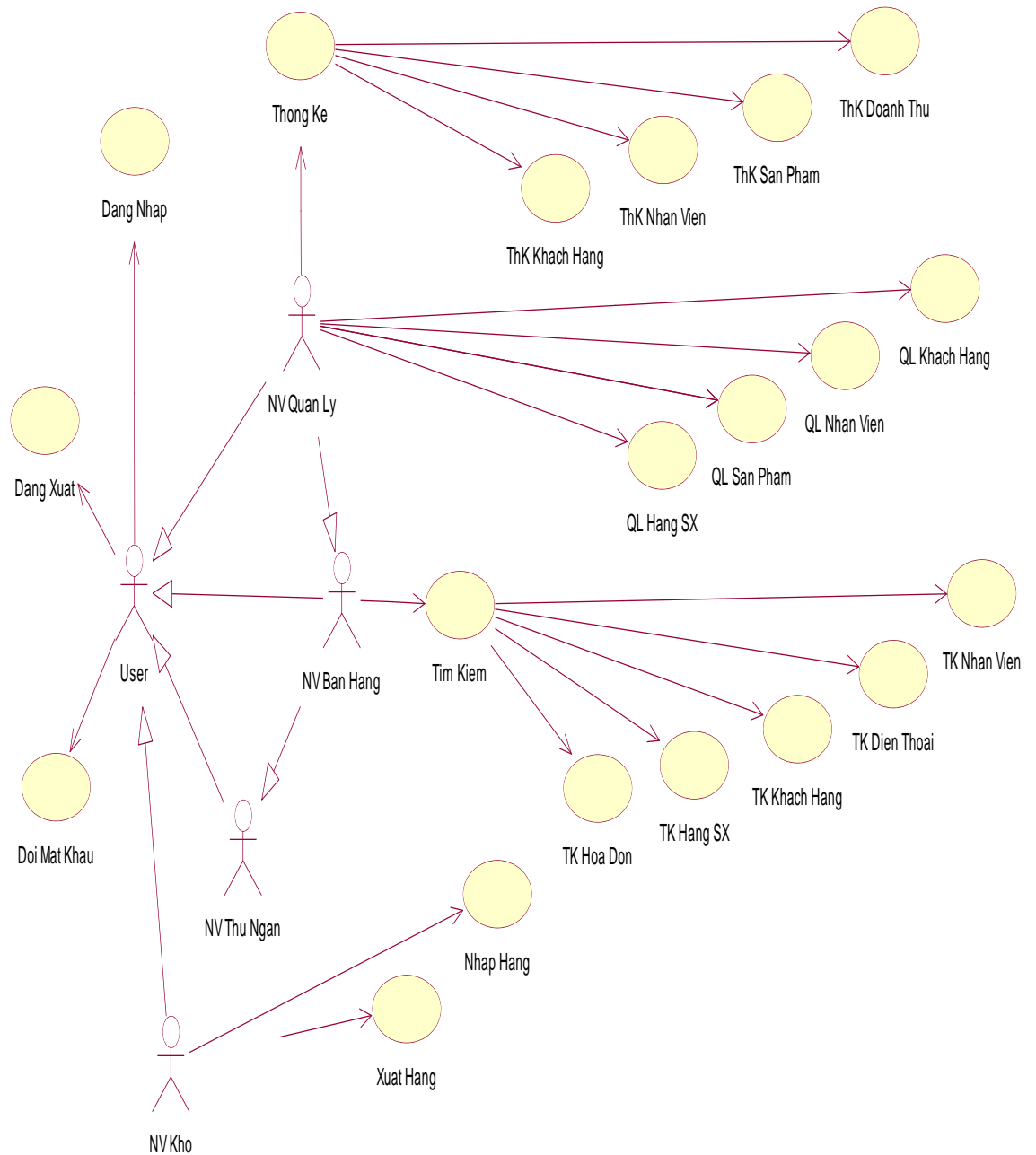
- 2Ghz processor
- 1GB RAM
- 8Gb dung lượng ổ đĩa
- Window XP, Window Vista, Window.

Phần mềm:

- Microsoft Visual Studio 2008
- Netframework 3.5
- Microsoft SQL Server 2008

### 3.3. Phân tích ứng dụng

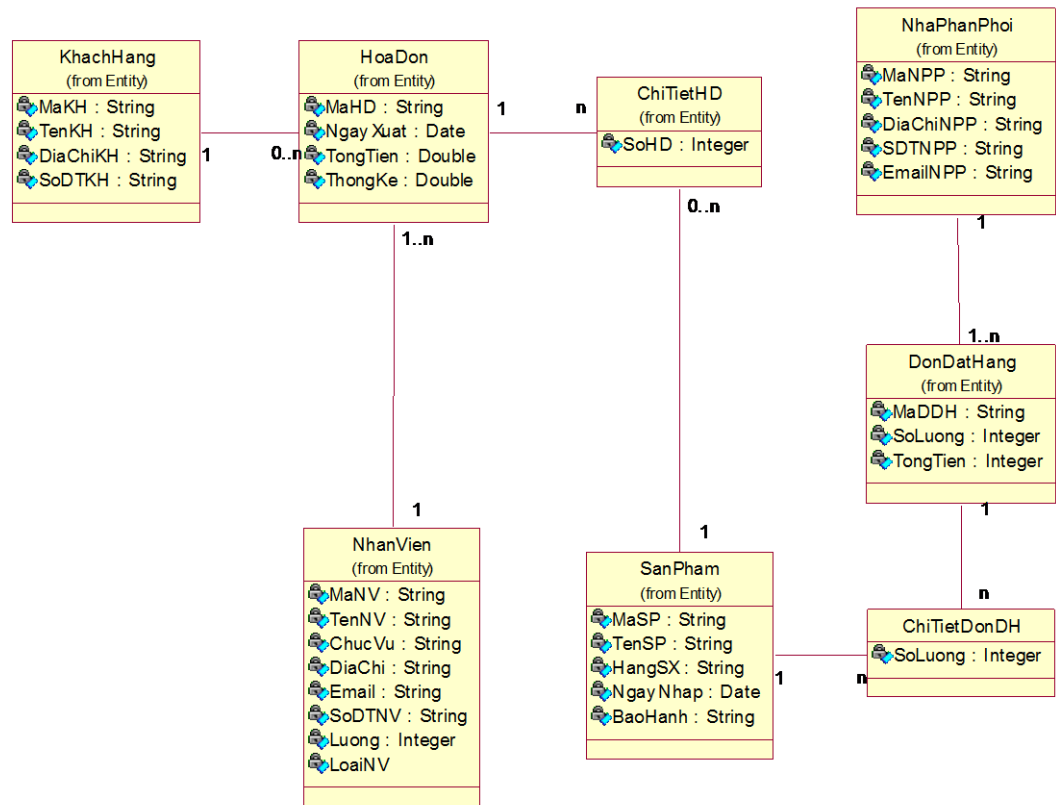
### 3.3.1 Sơ đồ Usecase



### Hình 3.1      Lược đồ Usecase tổng quát



### 3.3.2 Sơ đồ lớp



**Hình 3.2** Sơ đồ lớp

❖ Nhân viên: NhanVien

- Mã nhân viên: MaNV
- Tên nhân viên: TenNV
- Chức vụ: ChucVu
- Địa chỉ: DiaChi
- Email: Email
- Số điện thoại nhân viên: SoDTNV
- Lương: Luong

❖ Khách hàng: KháchHang

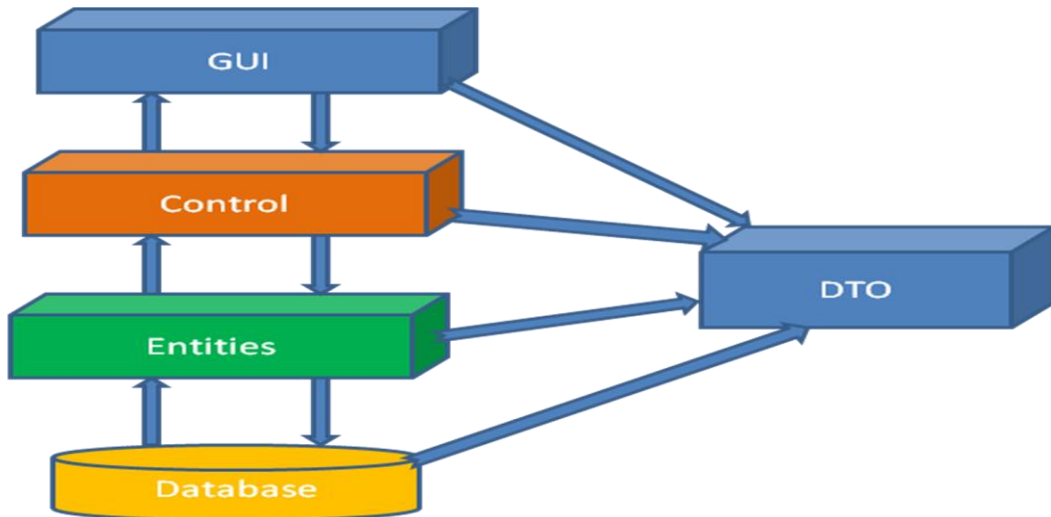
- Mã khách hàng: MaKH
- Tên khách hàng: TenKH
- Địa chỉ khách hàng: DiaChiKH
- Số điện thoại khách hàng: SoDTKH

- ❖ Sản phẩm: SanPham
  - Mã sản phẩm: MaSP
  - Tên sản phẩm: TenSP
  - Hãng sản xuất: HangSX
  - Ngày nhập: NgayNhap
  - Bảo hành: BaoHanh
- ❖ Nhà phân phối: NhaPhanPhoi
  - Mã nhà phân phối: MaNPP
  - Tên nhà phân phối: TenNPP
  - Địa chỉ nhà phân phối: DiaChiNPP
  - Số điện thoại nhà phân phối: SDTNPP
  - Email nhà phân phối: EmailNPP
- ❖ Hóa đơn: HoaDon
  - Mã hóa đơn: MaHD
  - Ngày xuất: NgayXuat
- ❖ Đơn đặt hàng: DonDatHang
  - Mã đơn đặt hàng: MaDDH
  - Số lượng: SoLuong
  - Tổng tiền: TongTien
- ❖ Thống kê: ThongKe
  - Doanh thu: DoanhThu
  - Hàng tồn: HangTon
  - Ngày thống kê: NgayTK
- ❖ Chi tiết hóa đơn: ChiTietHD
  - Số hóa đơn: SoHD

## Chương 4. THIẾT KẾ HỆ THỐNG VÀ THỰC HIỆN HÓA USECASE

### 4.1 Thiết kế tổng quát

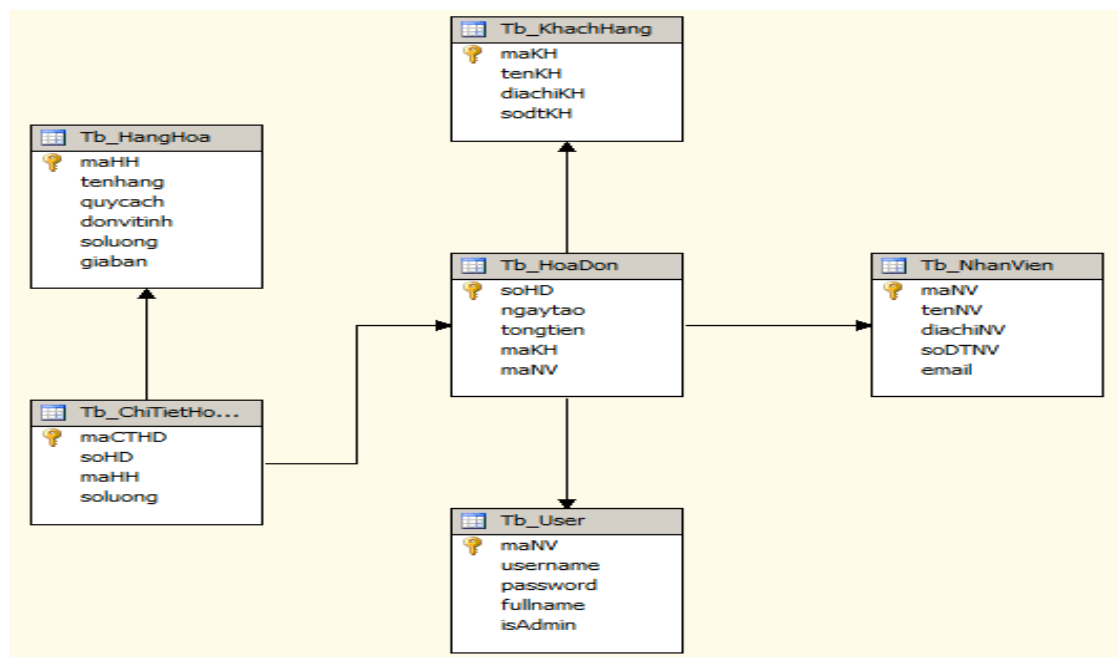
#### 4.1.1 Mô hình hệ thống



Hình 4.1 Mô hình của hệ thống

### 4.1.2 Thiết kế dữ liệu

#### 4.1.2.1 Mô hình cơ sở dữ liệu



Hình 4.2 Mô hình cơ sở dữ liệu

**4.1.2.2 Danh sách các bảng**

- Tb\_HangHoa(maHH, tenhang, quycach, donvitinh, soluong, giaban)
- Tb\_User(maNV, username, password, fullname, isAdmin)
- Tb\_NhanVien(maNV, tenNV, diachiNV, soDTNV, email)
- Tb\_HoaDon(soHD, ngaytao, tongtien, maKH, maNV)
- Tb\_KhachHang(maKH, tenKH, diachiKH, sodtKH)
- Tb\_ChiTietHoaDon(maCTHD, soHD, maHH, soluong)

**4.1.2.3 Mô tả chi tiết các bảng**

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	maHH	int	Khóa chính	Mã hàng hóa
2	tenhang	string		Tên hàng
3	quycach	string		Quy cách
4	donvitinh	string		Đơn vị tính
5	soluong	int		Số lượng
6	giaban	real		Giá bán

**Bảng 4.1** Mô tả chi tiết bảng Tb\_HangHoa

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	maUser	int	Khóa chính	Mã tài khoản
2	username	string		Tên đăng nhập
3	password	string		Mật khẩu đăng nhập
4	fullname	string		Tên đầy đủ
5	isAdmin	int		Phân quyền

**Bảng 4.2** Mô tả chi tiết bảng Tb\_User

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	maNV	int	Khóa chính	Mã nhân viên
2	tenNV	string		Tên nhân viên
3	diachiNV	string		Địa chỉ nhân viên
4	soDTNV	string		Số điện thoại nhân viên
5	email	string		Tài khoản email

**Bảng 4.3** Mô tả chi tiết bảng Tb\_NhanVien

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	soHD	int	Khóa chính	Số hóa đơn
2	ngaytao	date		Ngày tạo
3	tongtien	real		Tổng tiền
4	maKH	int		Mã khách hàng
5	maNV	int		Mã nhân viên

**Bảng 4.4** Mô tả chi tiết bảng Tb\_HoaDon

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	maKH	int	Khóa chính	Mã khách hàng
2	tenKH	string		Tên khách hàng
3	diachiKH	string		Địa chỉ khách hàng
4	sodtKH	string		Số điện thoại khách hàng

**Bảng 4.5** Mô tả chi tiết bảng Tb\_KhachHang

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc	Diễn giải
1	maCTHD	int	Khóa chính	Mã chi tiết hóa đơn
2	soHD	int		Số hóa đơn
3	maHH	int		Mã hàng hóa
4	soluong	int		Số lượng

**Bảng 4.6** Mô tả chi tiết bảng Tb\_ChiTietHoaDon

#### 4.1.3 Lựa chọn công nghệ

- Trong bài báo cáo này chúng em sử dụng công nghệ Rational Rose kết hợp với Visual Studio 2008 và Microsoft SQL Server 2008 để xây dựng website và hệ quản trị cơ sở dữ liệu.

- Công nghệ này giúp chúng ta thiết kế các chương trình quản lý nhanh chóng và hiệu quả.

### 4.2 Thiết kế chi tiết

#### 4.2.1 Usecase đăng nhập

- ❖ Vai trò use case: Use case đăng nhập cho phép nhân viên công ty đăng nhập vào hệ thống để thực hiện một số thao tác thêm, sửa, xóa,...

## ❖ Kịch bản use case

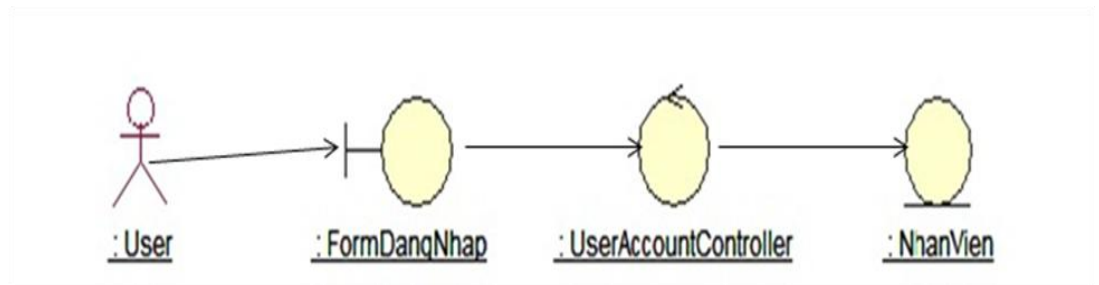
Tác nhân	Hệ thống
1. Người sử dụng mở Form đăng nhập	2. Form đăng nhập được mở
3. Người sử dụng nhập vào tên đăng nhập và mật khẩu	
4. Chọn nút đăng nhập	5. Hệ thống kiểm tra và gửi thông báo cho người đăng nhập biết đăng nhập có thành công không
6. Kết thúc quá trình đăng nhập	

## ❖ Giao diện của use case:

**Hình 4.3** Giao diện của Usecase đăng nhập

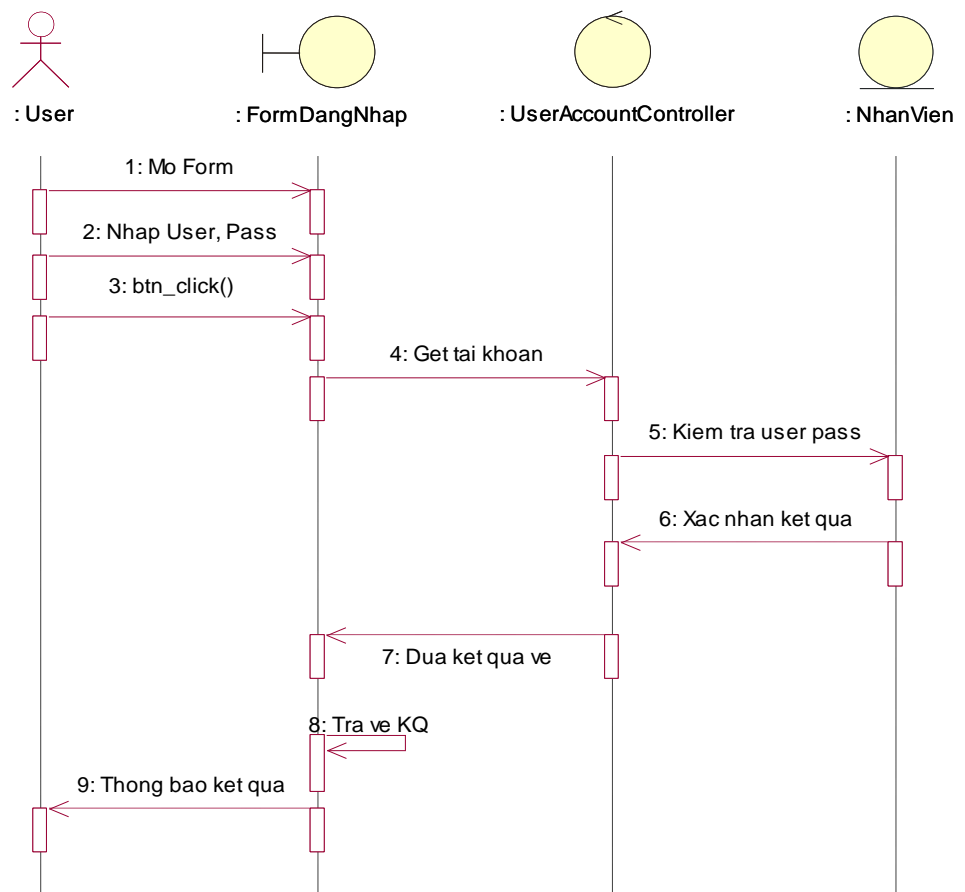


❖ Lược đồ lớp của use case:



**Hình 4.4** Lược đồ lớp của Usecase đăng nhập

❖ Lược đồ tuần tự của use case:



**Hình 4.5** Lược đồ tuần tự của Usecase đăng nhập

#### 4.2.2 Usecase đăng ký

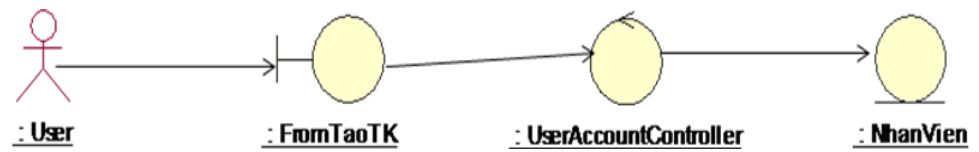
- ❖ Vai trò use case: Use case đăng ký cho phép nhân viên có thể tạo tài khoản mới để đăng nhập vào ứng dụng.
- ❖ Kịch bản use case:

Tác nhân	Hệ thống
1. Người sử dụng mở Form đăng ký	2. Form đăng ký được mở
3. Người sử dụng nhập vào các thông tin như: tên đăng nhập, mật khẩu, họ tên và chọn có là admin hay không	
4. Chọn nút đăng ký	
6. Kết thúc quá trình đăng ký	5. Hệ thống kiểm tra và gửi thông báo cho biết đăng ký có thành công không

- ❖ Giao diện của use case:

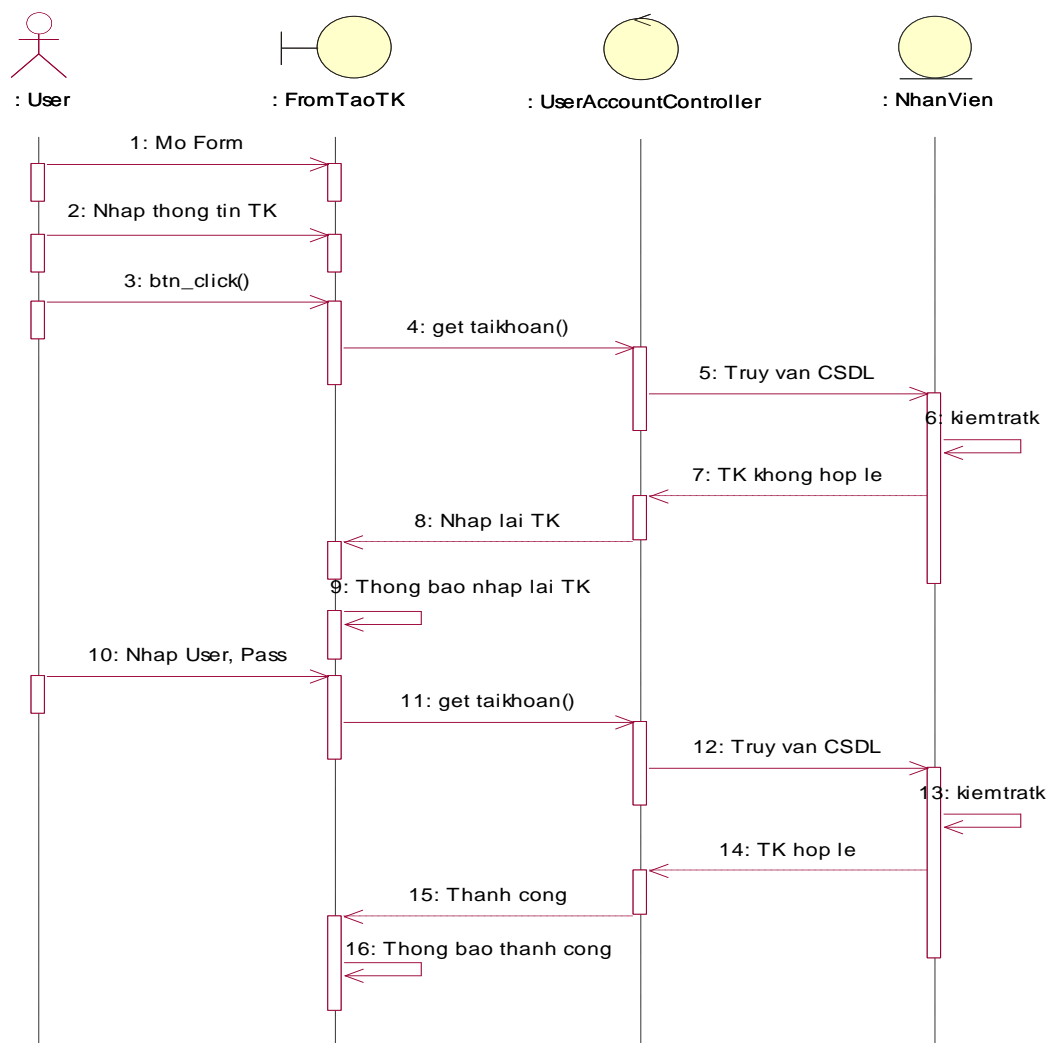
**Hình 4.6** Giao diện của Usecase đăng ký tài khoản

❖ Lược đồ lớp của use case:



**Hình 4.7** Lược đồ lớp của Usecase đăng ký

❖ Lược đồ tuần tự của use case:



**Hình 4.8** Lược đồ tuần tự của Usecase đăng ký

#### 4.2.3 Usecase nhập hàng

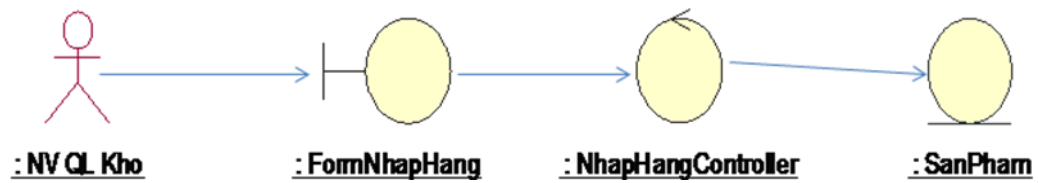
- ❖ Vai trò use case: Use case nhập hàng cho phép nhân viên có thể thêm sản phẩm mới vào hệ thống.
- ❖ Kịch bản use case:

Tác nhân	Hệ thống
1. Người sử dụng mở Form nhập hàng	2. Form nhập hàng được mở
3. Người sử dụng nhập vào các thông tin của sản phẩm	
4. Chọn nút thêm	
6. Kết thúc quá trình nhập hàng	5. Hệ thống kiểm tra và gửi thông báo cho biết nhập hàng có thành công không

- ❖ Giao diện của use case:

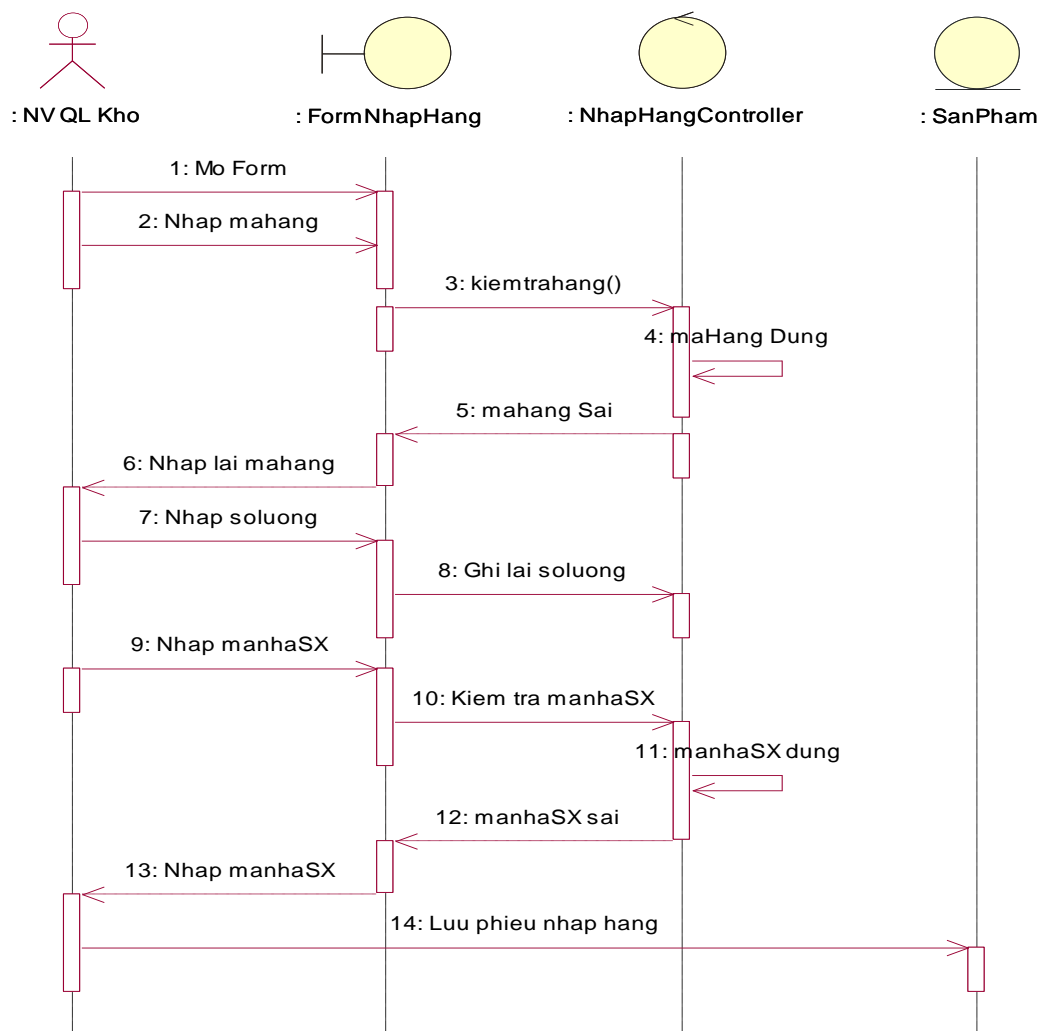
**Hình 4.9** Giao diện của Usecase nhập hàng

❖ Lược đồ lớp của use case:



**Hình 4.10** Lược đồ lớp của Usecase nhập hàng

❖ Lược đồ tuần tự của use case:



**Hình 4.11** Lược đồ tuần tự của Usecase nhập hàng

#### 4.2.4 Usecase đổi mật khẩu

- ❖ Vai trò use case: Use case đổi mật khẩu cho phép nhân viên có thể thay đổi mật khẩu của tài khoản đang sử dụng.
- ❖ Kịch bản use case:

Tác nhân	Hệ thống
1. Người sử dụng mở Form đổi mật khẩu	2. Form đổi mật khẩu được mở
3. Người sử dụng nhập vào các thông tin mật khẩu cũ và mật khẩu mới	
4. Chọn nút thay đổi	5. Hệ thống kiểm tra và gửi thông báo cho biết đổi mật khẩu có thành công không
6. Kết thúc quá trình đổi mật khẩu	

- ❖ Giao diện của use case:

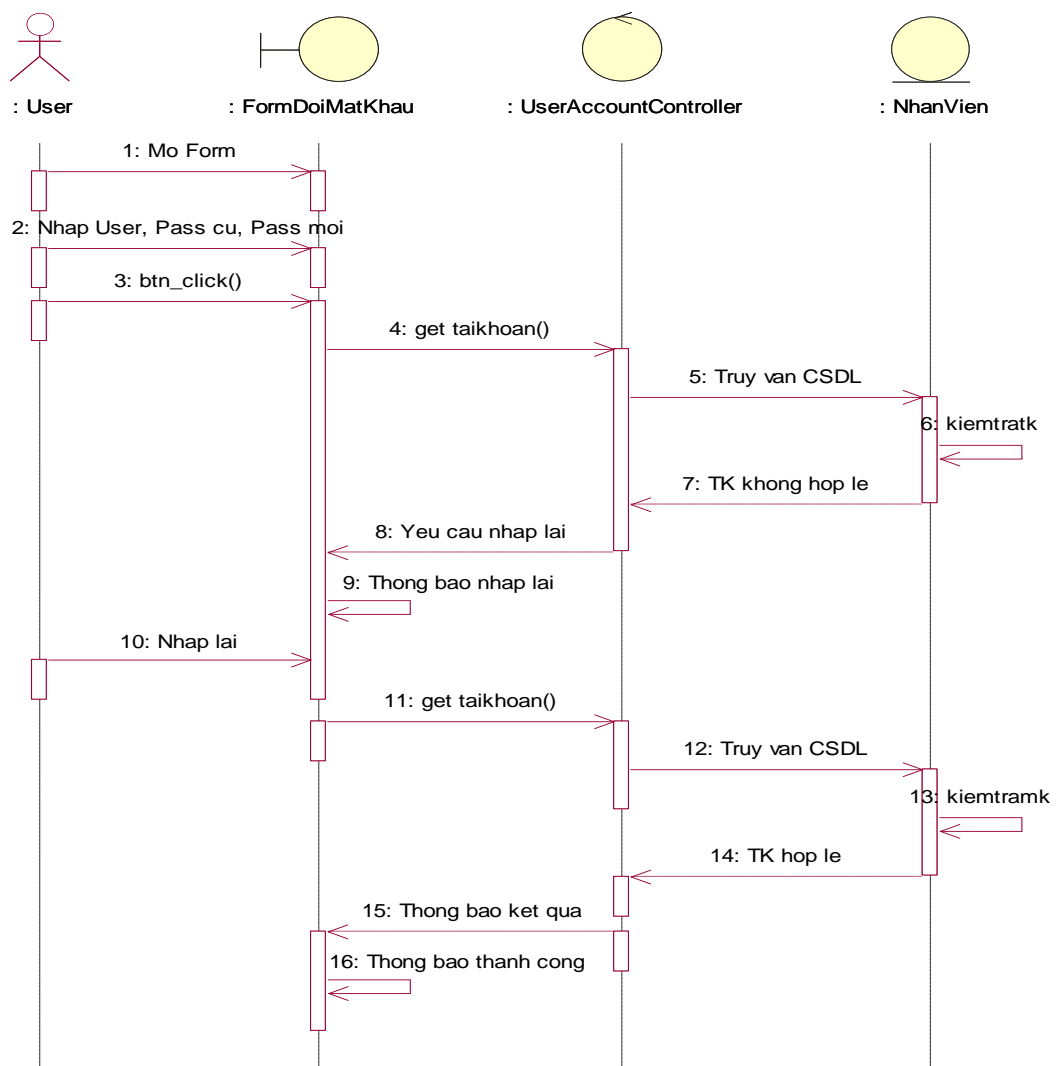
**Hình 4.12** Giao diện của Usecase đổi mật khẩu

❖ Lược đồ lớp của use case:



**Hình 4.13** Lược đồ lớp của Usecase đổi mật khẩu

❖ Lược đồ tuần tự của use case:



**Hình 4.14** Lược đồ tuần tự của Usecase đổi mật khẩu

#### 4.2.5 Usecase xóa sản phẩm

- ❖ Vai trò use case: Use case xóa sản phẩm cho phép nhân viên có thể xóa các sản phẩm khi không cần thiết lưu trữ nữa.
- ❖ Kịch bản use case:

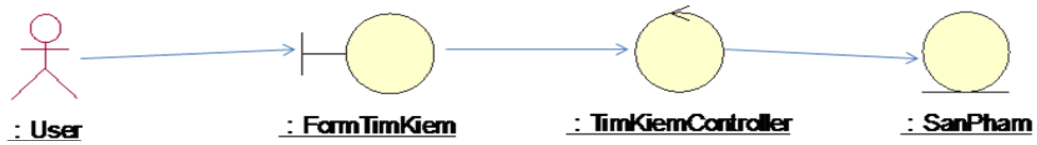
Tác nhân	Hệ thống
1. Người sử dụng mở Form xóa sản phẩm	2. Form xóa sản phẩm được mở
3. Người sử dụng tìm kiếm sản phẩm cần xóa	
4. Chọn nút xóa	
6. Kết thúc quá trình xóa sản phẩm	5. Hệ thống kiểm tra và gửi thông báo cho biết xóa sản phẩm có thành công không

- ❖ Giao diện của use case:

**Hình 4.15** Giao diện của Usecase xóa sản phẩm

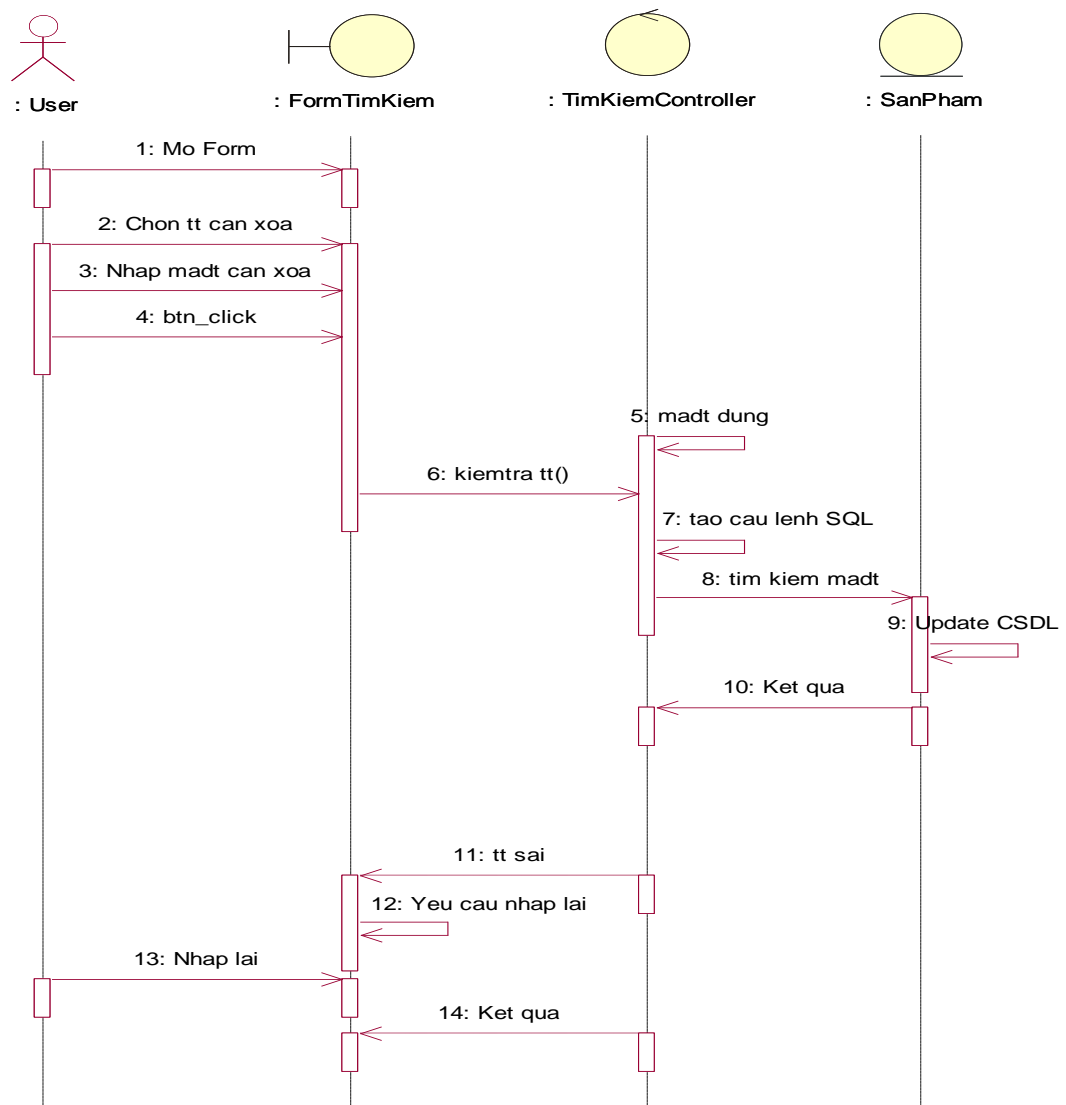


❖ Lược đồ lớp của use case:



**Hình 4.16** Lược đồ lớp của Usecase xóa sản phẩm

❖ Lược đồ tuần tự của use case:



**Hình 4.17** Lược đồ tuần tự của Usecase xóa sản phẩm

#### 4.2.6 Usecase xóa khách hàng

- ❖ Vai trò use case: Use case xóa khách hàng cho phép nhân viên có thể xóa các khách hàng khi không cần thiết lưu trữ nữa.
- ❖ Kịch bản use case:

Tác nhân	Hệ thống
1. Người sử dụng mở Form xóa khách hàng	2. Form xóa khách hàng được mở
3. Người sử dụng tìm kiếm khách hàng cần xóa	
4. Chọn nút xóa	
6. Kết thúc quá trình xóa khách hàng	5. Hệ thống kiểm tra và gửi thông báo cho biết xóa khách hàng có thành công không

- ❖ Giao diện của use case:

**THÔNG TIN KHÁCH HÀNG**

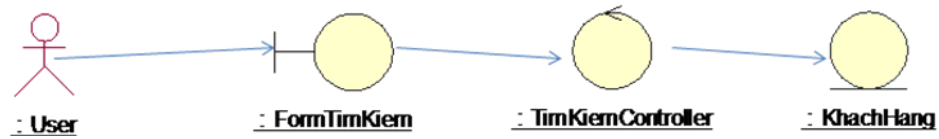
Mã khách hàng:  Địa chỉ:

Tên khách hàng:  Số điện thoại:

	MaKH	TenKH	DiaChiKH	SoDTKH
▶	1	Trần Nhứt Minh	Cảng Long - Trà ...	0989123456
	2	Huỳnh Kim Đức	Phường 6 - Trà Vi...	0989654321
	3	Phan Quốc Tín	Phường 8 - Trà Vi...	0919123654
	4	Trịnh Hải Sơn	Cầu Kê - Trà Vinh	0979456789
	5	Nguyễn Thị Than...	P5 - Trà Vinh	0978123456

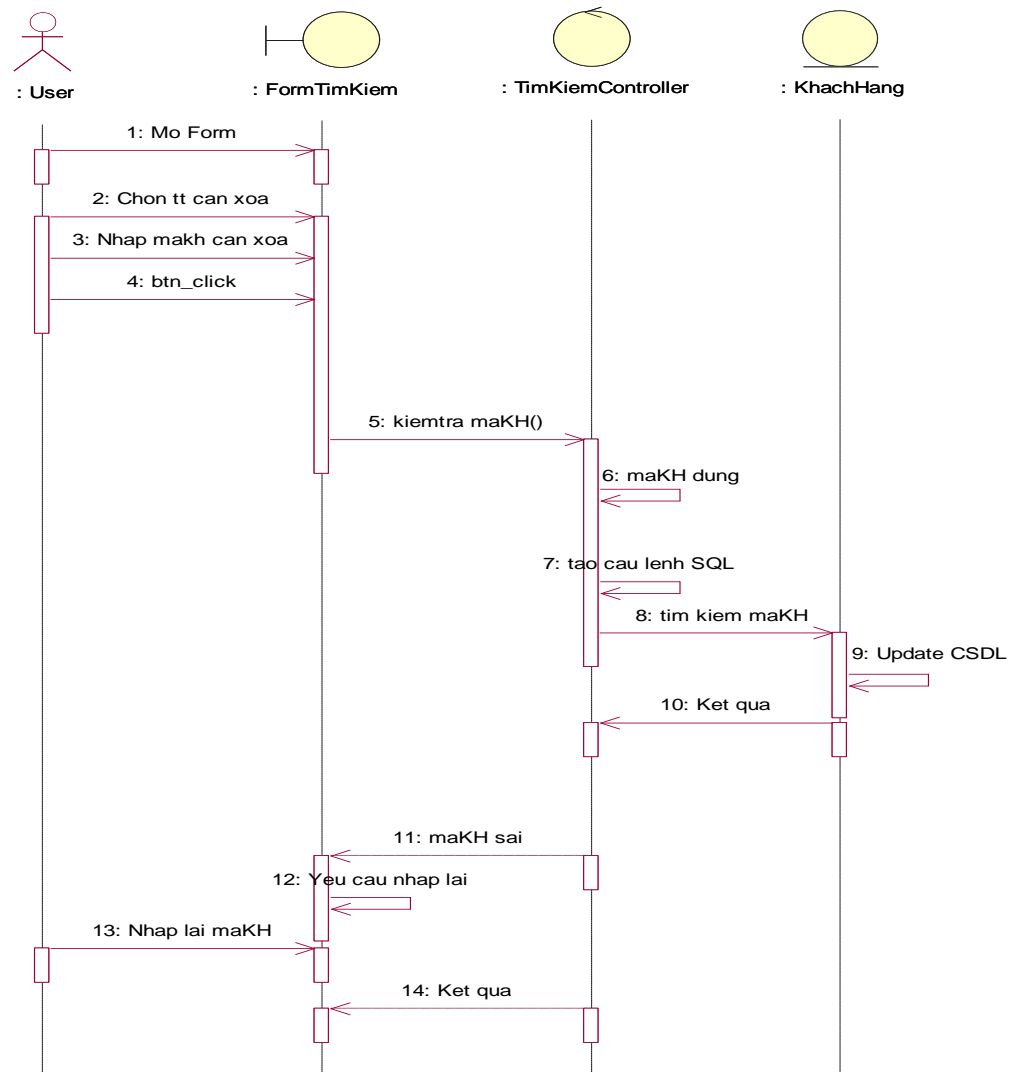
**Hình 4.18** Giao diện của Usecase xóa khách hàng

❖ Lược đồ lớp của use case:



**Hình 4.19** Lược đồ lớp của Usecase xóa khách hàng

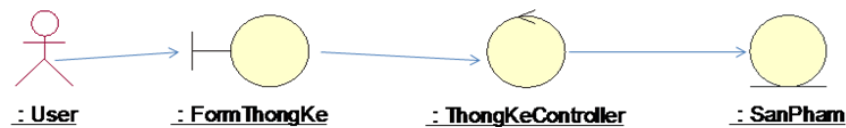
❖ Lược đồ tuần tự của use case:



**Hình 4.20** Lược đồ tuần tự của Usecase xóa khách hàng

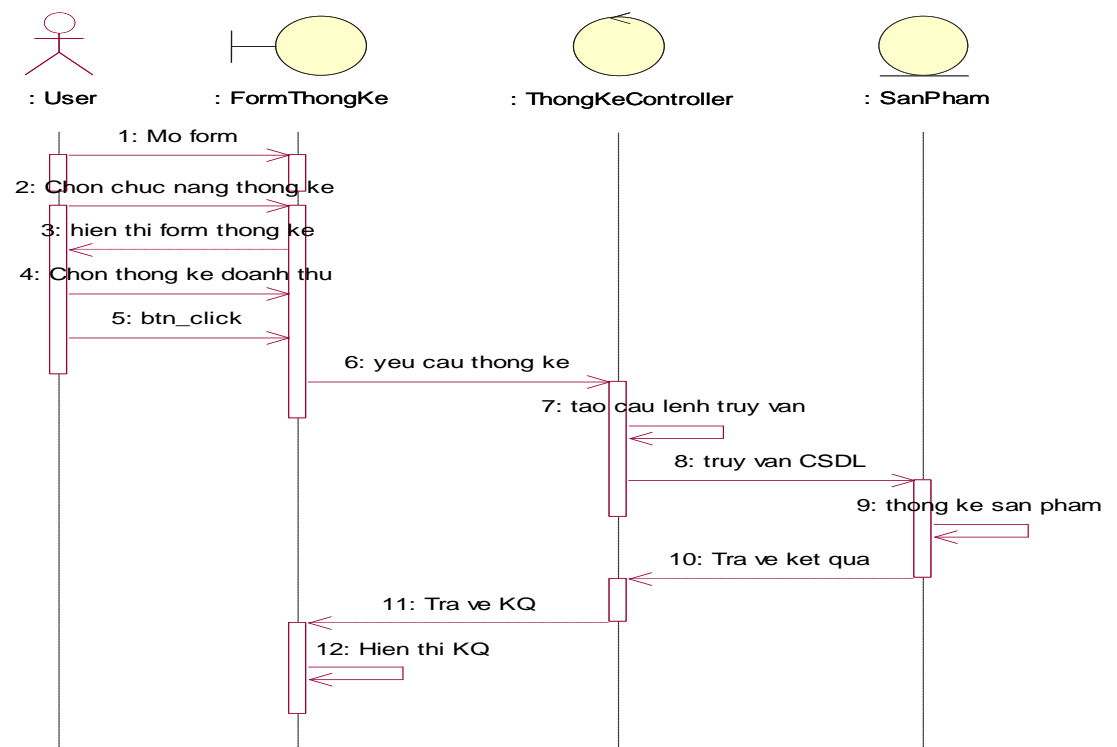
#### 4.2.7 Usecase thống kê sản phẩm

- ❖ Vai trò use case: Use case thống kê sản phẩm cho phép nhân viên có thể thống kê sản phẩm dễ dàng.
- ❖ Giao diện của use case:
- ❖ Lược đồ lớp của use case:



**Hình 4.21** Lược đồ lớp của Usecase thống kê sản phẩm

- ❖ Lược đồ tuần tự của use case:



**Hình 4.22** Lược đồ tuần tự của Usecase thống kê sản phẩm

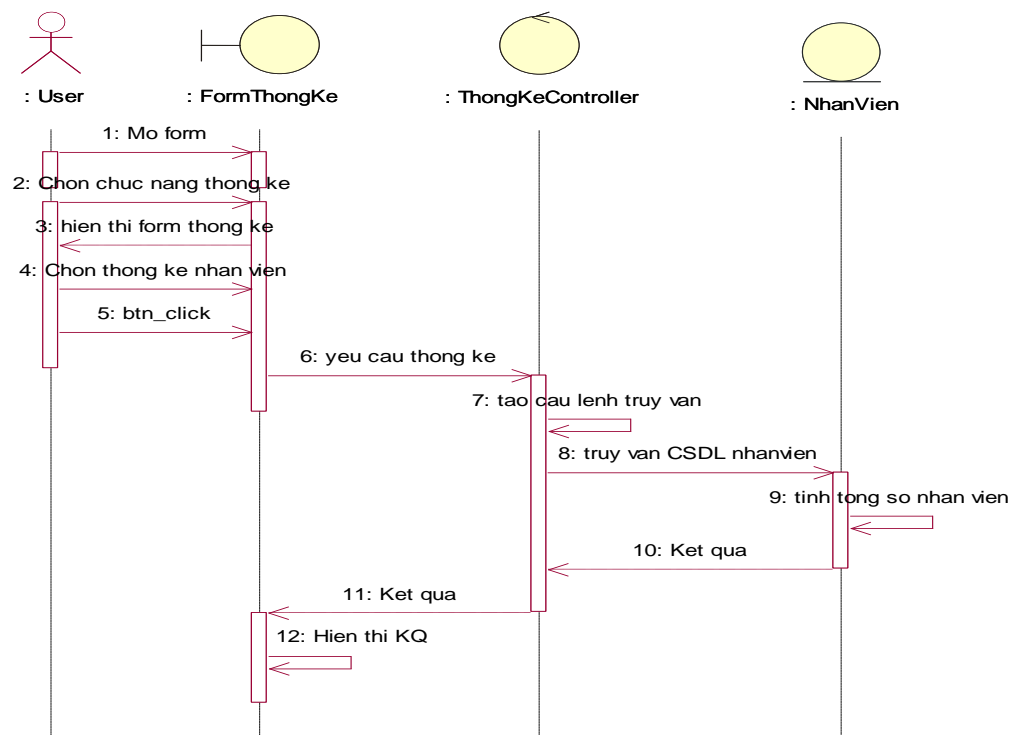
#### 4.2.8 Usecase thống kê nhân viên

- ❖ Vai trò use case: Use case thống kê nhân viên cho phép xem thông tin tất cả nhân viên.
- ❖ Giao diện của use case:
- ❖ Lược đồ lớp của use case:



**Hình 4.23** Lược đồ lớp của Usecase thống kê nhân viên

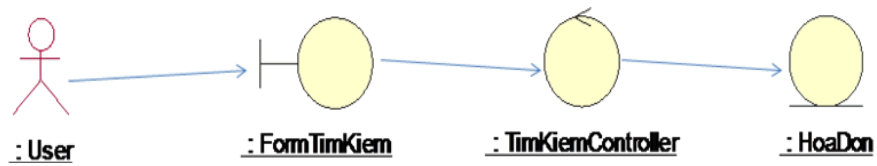
- ❖ Lược đồ tuần tự của use case:



**Hình 4.24** Lược đồ tuần tự của Usecase thống kê nhân viên

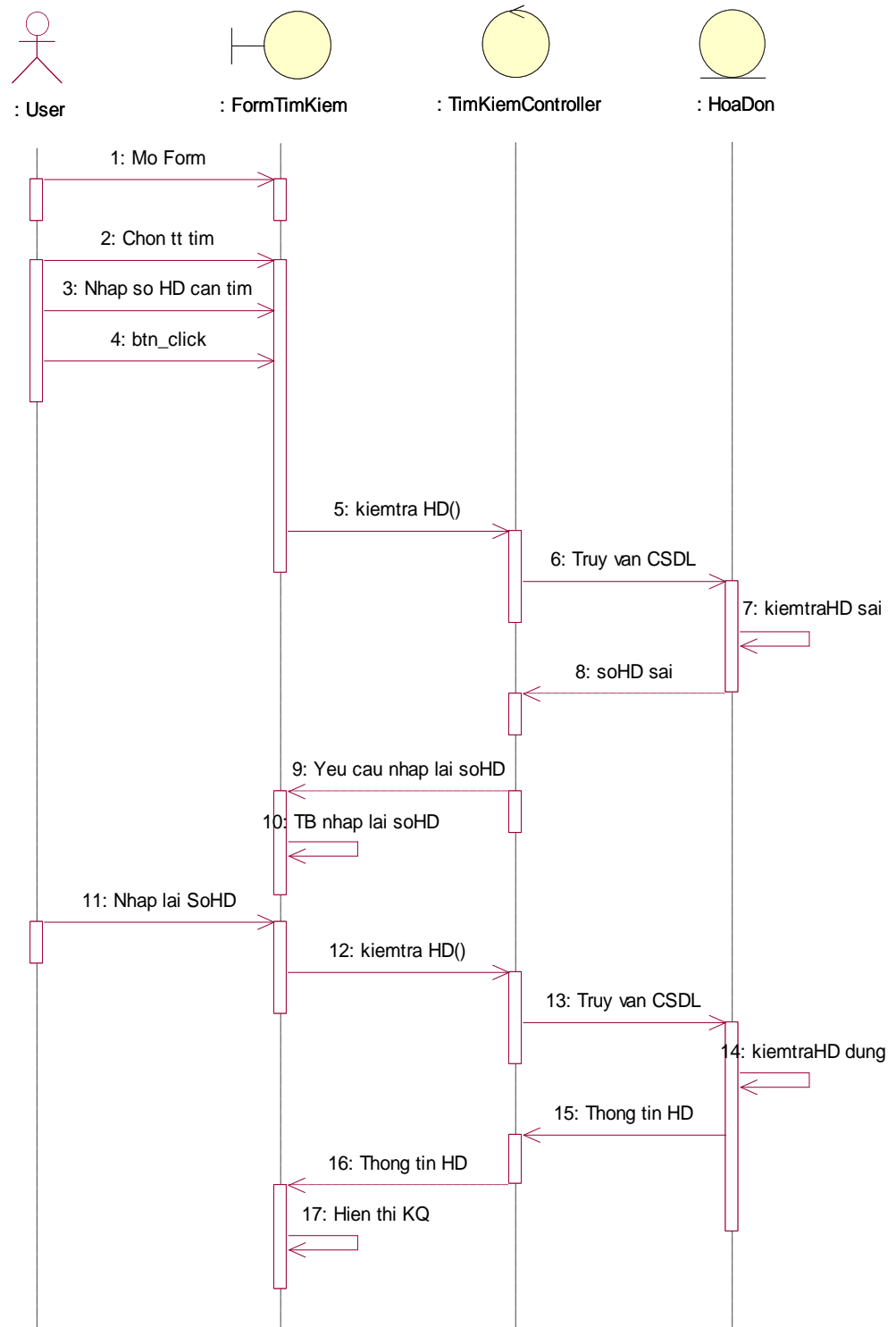
#### 4.2.9 Usecase thống kê hóa đơn

- ❖ Vai trò use case: Use case thống kê hóa đơn cho phép xem thông tin tất cả hóa đơn đã xuất.
- ❖ Giao diện của use case:
- ❖ Lược đồ lớp của use case:



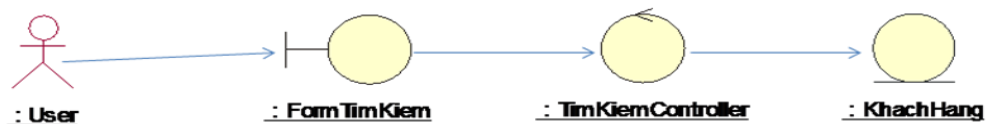
**Hình 4.25** Lược đồ lớp của Usecase thống kê hóa đơn

❖ Lược đồ tuần tự của use case:

**Hình 4.26** Lược đồ tuần tự của Usecase thống kê hóa đơn

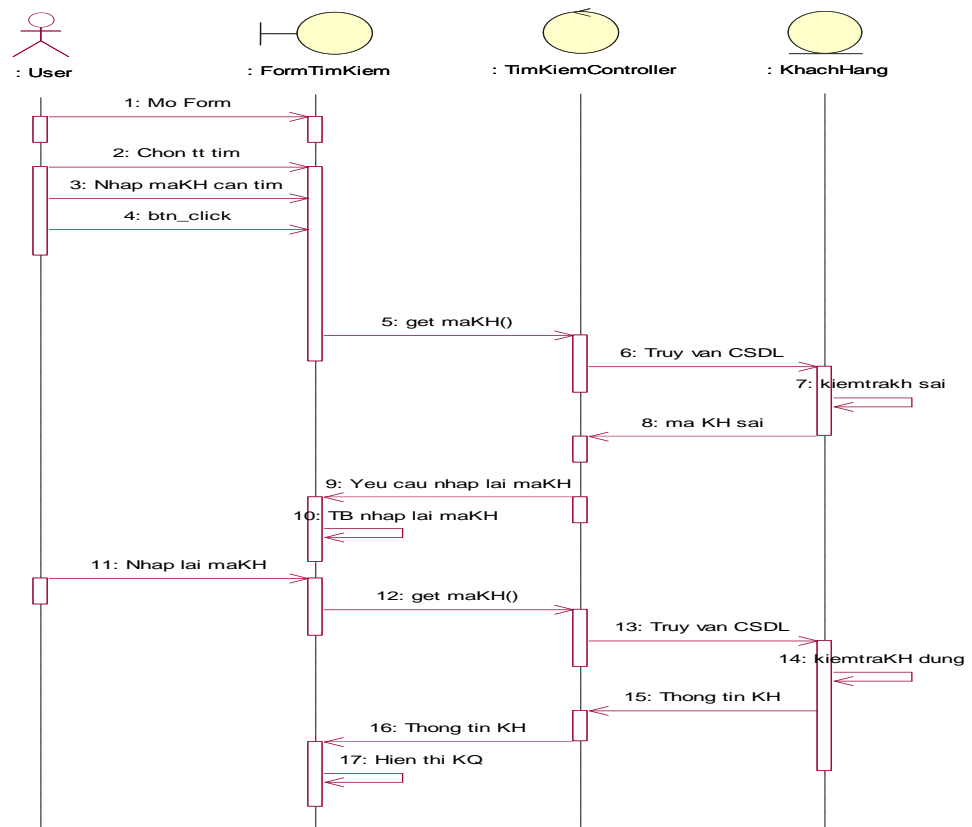
#### 4.2.10 Usecase thống kê khách hàng

- ❖ Vai trò use case: Use case thống kê khách hàng cho phép xem thông tin tất cả khách hàng.
- ❖ Giao diện của use case:
- ❖ Lược đồ lớp của use case:



**Hình 4.27** Lược đồ lớp của Usecase thống kê khách hàng

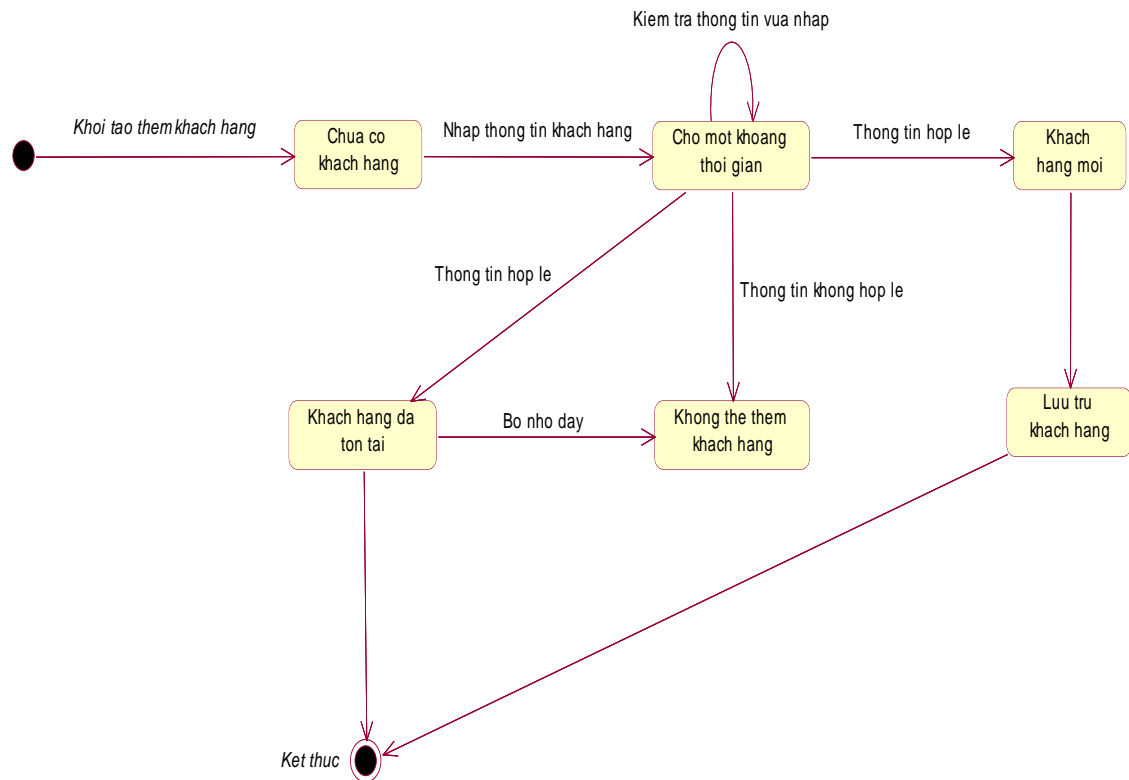
- ❖ Lược đồ tuần tự của use case:



**Hình 4.28** Lược đồ tuần tự của Usecase thống kê khách hàng

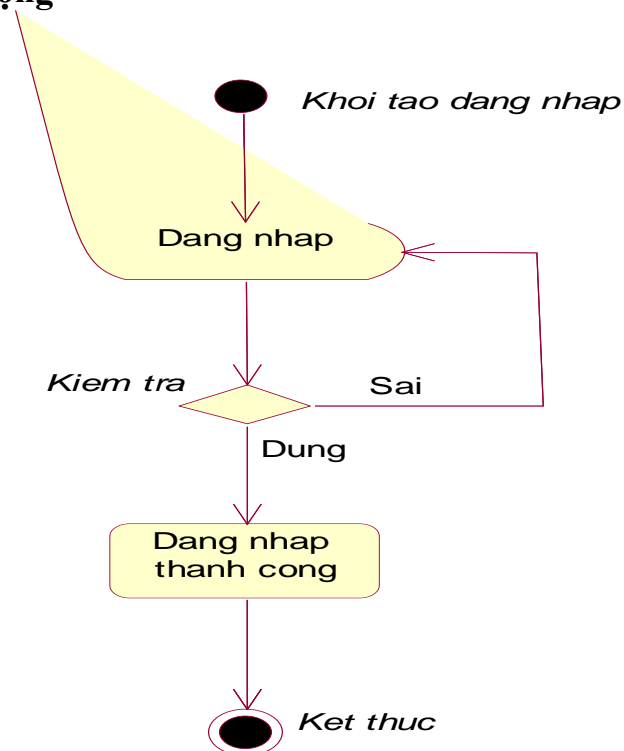


## Lược đồ trạng thái



Hình 4.29 Lược đồ trạng thái

## Lược đồ hoạt động



Hình 4.30 Lược đồ hoạt động

## **Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **5.1 Kết luận**

#### **5.1.1 Kết quả đạt được**

❖ Ứng dụng bán điện thoại đáp ứng được các chức năng:

- Đăng nhập vào ứng dụng
- Tạo tài khoản người dùng
- Thay đổi mật khẩu
- Tìm kiếm sản phẩm theo tên, mã
- Tìm kiếm khách hàng
- Thêm khách hàng
- Thêm sản phẩm
- Sửa và xóa khách hàng
- Sửa và xóa sản phẩm
- Thống kê hóa đơn
- Thống kê sản phẩm tồn
- Thống kê khách hàng

❖ Kinh nghiệm:

- Kỹ năng thu thập thông tin từ Internet, khả năng vận dụng kiến thức đã học vào thực tế.
- Nâng cao khả năng làm việc nhóm
- củng cố và nâng cao các kiến thức chuyên môn, đặc biệt là lĩnh vực phân tích thiết kế hệ thống thông tin, cơ sở dữ liệu, thiết kế và lập trình web cùng các kỹ năng về đồ họa ứng dụng

### **5.1.2 Hạn chế**

❖ Bên cạnh các kết quả đạt được thì hệ thống cũng còn một số hạn chế nhất định:

- Xây dựng ứng dụng với nhiều ngôn ngữ khác nhau
- Chưa đảm bảo trong việc lỗi ứng dụng
- Ứng dụng chưa đa dạng hóa chức năng

### **5.2 Hướng phát triển**

- Xây dựng thêm chế độ bảo mật an toàn hơn.
- Mở rộng xây dựng ứng dụng lớn hơn với đa dạng các chức năng để phục vụ cho các công ty lớn.
- Phát triển ứng dụng thông minh hơn.

## TÀI LIỆU THAM KHẢO

### Sách

Nhan Minh Phúc, Bài giảng xây dựng phần mềm hướng đối tượng.

### Internet

1. <https://vi.scribd.com/doc/2516102/13/L%E1%BB%8Bch-s%E1%BB%AD-ra-%C4%91%E1%BB%9Di-c%E1%BB%A7a-UML>
2. <https://huynhminhkhoa.wordpress.com/c-shap-vb-net/nhung-dac-trung-co-ban-cua-oop/>
3. [http://123doc.org/doc\\_search\\_title/1310607-giao-an-bai-giang-bai-tap-xay-dung-phan-mem-huong-doi-tuong.htm](http://123doc.org/doc_search_title/1310607-giao-an-bai-giang-bai-tap-xay-dung-phan-mem-huong-doi-tuong.htm)
4. <http://tailieu.vn/doc/de-tai-xay-dung-phan-mem-quan-ly-ban-hang-tai-cong-ty-cp-thuong-mai-phu-xuyen-bang-ngon-ngu-lap-tri-1310389.html>
5. <http://kcntt.duytan.edu.vn/Home/ArticleDetail/vn/161/2040/bieu-do-use-case-trong-phan-tich-thiet-ke-huong-doi-tuong>