

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH**

**SE112.K11**

**Đề tài:**

**TÌM HIỂU WEB SERVICE VÀ XÂY DỰNG ỨNG DỤNG  
THƯƠNG MẠI ĐIỆN TỬ**

Giảng viên hướng dẫn: **ĐỖ THỊ THANH TUYỀN**

Sinh viên thực hiện:

**Nguyễn Minh Hoàng**

**15520257**

*TP. Hồ Chí Minh, ngày 01 tháng 01 năm 2020*



# LỜI MỞ ĐẦU

Dịch vụ Web (Web Service) được coi là một công nghệ mang đến cuộc cách mạng trong cách thức hoạt động của các dịch vụ B2B (Business to Business) và B2C (Business to Customer). Giá trị cơ bản của dịch vụ Web dựa trên việc cung cấp các phương thức theo chuẩn trong việc truy nhập đối với hệ thống đóng gói và hệ thống kế thừa. Các phần mềm được viết bởi những ngôn ngữ lập trình khác nhau và chạy trên những nền tảng khác nhau có thể sử dụng dịch vụ Web để chuyển đổi dữ liệu thông qua mạng Internet theo cách giao tiếp tương tự bên trong một máy tính. Tuy nhiên, công nghệ xây dựng dịch vụ Web không nhất thiết phải là các công nghệ mới, nó có thể kết hợp với các công nghệ đã có như XML, SOAP, WSDL, UDDI... Với sự phát triển và lớn mạnh của Internet, dịch vụ Web thật sự là một công nghệ đáng được quan tâm để giảm chi phí và độ phức tạp trong tích hợp và phát triển hệ thống.

Thương mại điện tử (Electronic commerce - E-commerce) là hình thái hoạt động thương mại bằng phương pháp điện tử; là việc trao đổi thông tin thương mại thông qua các phương tiện công nghệ điện tử. Hoạt động thương mại gồm có trao đổi, mua bán hàng hóa, dịch vụ, xúc tiến thương mại, quảng cáo, khuyến mại... Các hoạt động này nhất thiết phải trên một hạ tầng mạng truyền tải thông tin số hóa như Mạng Internet, Website, mạng điện thoại di động... Quá trình giao dịch không nhất thiết phải có hai bên tham gia trực tiếp nhưng có hóa đơn hoặc bất cứ giấy tờ điện tử nào có thể chứng thực. Việc thanh toán dựa vào các ứng dụng về số hóa: thẻ tín dụng, thẻ trả sau, tài khoản ngân hàng online – E-banking.... Còn việc chứng thực giao dịch có thể dựa vào các luật về thương mại điện tử quốc tế hoặc vùng lãnh thổ diễn ra giao dịch, chữ ký điện tử, mã số thuế ....

Ngày nay Web service đã trở nên rất thông dụng. Nó được ứng dụng rất rộng rãi trong các lĩnh vực của cuộc sống, trong Công nghệ thông tin nói chung và trong thương mại điện tử nói riêng. Web service có thể được ứng dụng trong tất cả các hoạt động của thương mại điện tử như: quảng cáo trực tuyến, sử dụng trong các công cụ tìm kiếm, ứng dụng trong giao dịch mua bán hàng trực tuyến và trong dịch vụ thanh toán.

# LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn chân thành đến tập thể quý Thầy Cô Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM và quý Thầy Cô khoa Công Nghệ Phần Mềm đã giúp cho em có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này.

Đặc biệt, em xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới Cô **Đỗ Thị Thanh Tuyền**. Đã giúp em hoàn thành tốt báo cáo môn học của mình.

Trong thời gian một học kỳ thực hiện đề tài, em đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, em vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, em không tránh khỏi những thiếu sót. Chính vì vậy, em rất mong nhận được những sự góp ý từ phía các Thầy Cô nhằm hoàn thiện những kiến thức mà em đã học tập và là hành trang để em thực hiện tiếp các đề tài khác trong tương lai.

Em xin chân thành cảm ơn quý Thầy Cô!

*Sinh viên thực hiện*

# MỤC LỤC

A. TÌM HIỂU VỀ WEB SERVICE VÀ THƯƠNG MẠI ĐIỆN TỬ.....	9
CHƯƠNG I. TÌM HIỂU VỀ WEB SERVICE.....	10
1. Khái niệm về Web Service.....	10
2. Đặc điểm của Web Service.....	11
3. Ưu, nhược điểm của Web Service .....	12
4. Mô hình Web Service.....	14
5. Các thành phần chính của Web Service .....	14
5.1. Giao thức vận HTTP.....	15
5.2. Giao thức truyền thông SOAP .....	15
5.3. Tầng mô tả dịch vụ XML, WSDL.....	19
5.4. Universal Discovery Description and Intergration .....	22
6. Web Service hoạt động như thế nào?.....	24
7. Các loại Web Service .....	26
7.1. SOAP Web Service là gì?.....	26
7.2. RESTful Web Service là gì? .....	26
7.3. Sự khác nhau giữa REST và SOAP .....	31
8. Lý do sử dụng REST thay vì SOAP?.....	32
9. Lý do sử dụng SOAP?.....	32
10. Vấn đề an toàn cho Web Service .....	33
11. Mô hình của ứng dụng Web Service.....	35
11.1. Xây dựng một Web Service .....	35
11.2. Quy trình xây dựng một dịch vụ Web.....	36
11.3. Tích hợp Web service theo chuẩn .....	36
12. Giới thiệu về Web API.....	37
12.1. Phương thức thiết kế tự động.....	37
12.2. Web Service sử dụng ngôn ngữ đánh dấu .....	38
12.3. Khuyết điểm .....	38
12.4. Kiểm thử hồi quy trong Web Service.....	39
12.5. Web Service change management.....	39
13. JSON .....	39
CHƯƠNG II. TỔNG QUAN VỀ THƯƠNG MẠI ĐIỆN TỬ.....	44
1. Khái niệm thương mại điện tử .....	44
1.1. Thương mại điện tử theo nghĩa hẹp:.....	44
1.2. Thương mại điện tử theo nghĩa rộng: .....	44

2.	Bản chất, đặc trưng, lợi ích và hạn chế của thương mại điện tử .....	45
2.1.	Bản chất của thương mại điện tử .....	45
2.2.	Đặc trưng của thương mại điện tử .....	46
2.3.	Lợi ích và hạn chế của thương mại điện tử .....	46
3.	Nền tảng của thương mại điện tử .....	52
3.1.	Mạng viễn thông và Internet .....	52
3.2.	Các công nghệ hỗ trợ thương mại điện tử .....	56
3.3.	Ứng dụng của thương mại điện tử: .....	58
CHƯƠNG III: ỨNG DỤNG CỦA WEB SERVICE TRONG TMĐT .....		65
1.	Ứng dụng trong quảng cáo trực tuyến: .....	65
1.2.	Quảng cáo logo – banner, pop-up: .....	66
1.3.	Quảng cáo bằng đường Text link .....	66
1.4.	Quảng cáo tài trợ tại Google, Yahoo!, MSN, Altavista... ..	66
2.	Ứng dụng trong các công cụ tìm kiếm: .....	68
3.	Ứng dụng trong giao dịch, mua bán hàng trực tuyến .....	70
4.	Ứng dụng trong dịch vụ thanh toán (Payment gateway) .....	70
4.1.	Payment gateway là gì? .....	70
4.2.	Payment Gateway hoạt động thế nào? .....	71
B. ỨNG DỤNG MUA HÀNG TRỰC TUYẾN TRÊN THIẾT BỊ DI ĐỘNG .....		73
CHƯƠNG I: TỔNG QUAN ĐỀ TÀI .....		74
1.	Tổng quan ứng dụng .....	74
1.1.	Phạm vi .....	74
1.2.	Mục tiêu .....	74
1.3.	Ý nghĩa lý luận và thực tiễn của đề tài .....	74
2.	Khảo sát hiện trạng .....	75
2.1.	Hiện trạng tổ chức .....	75
2.2.	Hiện trạng về nghiệp vụ .....	76
CHƯƠNG II: ĐẶC TẢ YÊU CẦU .....		77
1.	Mô hình Use Case .....	77
2.	Danh sách các Actor .....	77
3.	Danh sách các Use Case .....	77
4.	Đặc tả Use Case .....	78
4.1.	Đặc tả Use Case “Đăng Ký” .....	78
4.2.	Đặc tả usecase “Đăng nhập” .....	78
4.3.	Đặc tả usecase “Đăng xuất” .....	78

4.4. Đặc tả Usecase “Xem danh sách hàng” .....	78
4.5. Đặc tả Usecase “Xem chi tiết hàng” .....	78
4.6. Đặc tả Usecase “Quản lý giỏ hàng” .....	79
4.7. Đặc tả Usecase “Xác nhận đơn hàng” .....	79
CHƯƠNG III: PHÂN TÍCH HỆ THỐNG.....	80
1. Activity Diagram .....	80
1.1. Đăng ký .....	80
1.2. Đăng nhập.....	80
1.3. Đăng xuất .....	80
1.4. Xem danh sách sản phẩm.....	81
1.5. Xem chi tiết sản phẩm.....	81
1.6. Quản lý giỏ hàng của User .....	81
1.7. Quá trình mua hàng của User.....	81
2. Sơ đồ tuần tự .....	82
2.1. Đăng nhập.....	82
2.2. Đăng xuất .....	82
2.3. Đăng ký .....	83
2.4. Xem chi tiết sản phẩm.....	83
2.5. Quản lý giỏ hàng.....	84
2.6. Quá trình xác nhận đơn hàng.....	84
CHƯƠNG IV: XÂY DỰNG ỨNG DỤNG .....	85
1. Mô hình hóa dữ liệu.....	85
2. Đặc tả mô hình .....	85
3. Web Service .....	87
4. Android .....	89
4.1. Package adapter .....	89
4.2. Package fragments .....	90
4.3. Package helper.....	90
4.4. Package hhhai0304.dfc: .....	93
4.5. Package item.....	96
CHƯƠNG VI: THIẾT KẾ GIAO DIỆN VÀ CHỨC NĂNG .....	97
1. Màn hình khởi động.....	97
2. Màn hình đăng nhập .....	98
3. Màn hình giao diện món ăn.....	99
4. Màn hình danh mục của ứng dụng .....	100

5. Màn hình thông tin người dùng.....	101
6. Màn hình đổi mật khẩu của người dùng.....	102
7. Màn hình đổi thông tin người dùng.....	103
8. Màn hình đăng ký tài khoản mới.....	104
9. Màn hình tìm kiếm sản phẩm.....	105
10. Màn hình sắp xếp sản phẩm.....	106
11. Màn hình giỏ hàng.....	107
12. Màn hình thông tin đơn hàng .....	110
13. Màn hình chi tiết sản phẩm.....	112
14. Màn hình thông tin nhận hàng .....	113
CHƯƠNG VII: KẾT LUẬN .....	114
1. Kết quả đạt được.....	114
2. Khó khăn và hạn chế.....	114
3. Hướng phát triển.....	115
4. Tài liệu tham khảo .....	115



## **A. TÌM HIỂU VỀ WEB SERVICE VÀ THƯƠNG MẠI ĐIỆN TỬ**

# CHƯƠNG I. TÌM HIỂU VỀ WEB SERVICE

## 1. Khái niệm về Web Service

Theo IBM: “Service is a repeatable task within a business process”. Theo đó, Service là một ứng dụng với người dùng, một thao tác được thực hiện một hoặc nhiều lần trong một tiến trình và được thực hiện bởi một hay nhiều người.

Service là một hệ thống có khả năng nhận một hay nhiều yêu cầu xử lý và sau đó đáp ứng lại bằng cách trả về một hay nhiều kết quả. Quá trình nhận yêu cầu và trả kết quả về được thực hiện thông qua các giao diện đã được định nghĩa trước đó. Thông thường việc giao tiếp này được thực hiện trên các giao diện đã được chuẩn hóa và sử dụng rộng rãi.

Một hệ thống được thiết kế theo kiểu hướng Service là một hệ thống trong đó các chức năng của hệ thống được xây dựng dựa trên các service có độ kết dính thấp. Các service trong hệ thống giao tiếp với nhau thông qua việc gửi nhận các thông điệp.

Web Service (dịch vụ web) là tập hợp các giao thức và tiêu chuẩn mở được sử dụng để trao đổi dữ liệu giữa các ứng dụng hoặc giữa các hệ thống.

Các ứng dụng phần mềm được viết bằng các ngôn ngữ lập trình khác nhau hoặc chạy trên các nền tảng khác nhau, chúng có thể sử dụng các Web Service để trao đổi dữ liệu qua lại theo cách tương tự như liên lạc giữa các quá trình trên một máy tính.

Dịch vụ Web (Web Service) được coi là một công nghệ mang đến cuộc cách mạng trong cách thức hoạt động của các dịch vụ B2B (Business to Business) và B2C (Business to Customer). Giá trị cơ bản của dịch vụ Web dựa trên việc cung cấp các phương thức theo chuẩn trong việc truy nhập đối với hệ thống đóng gói và hệ thống kế thừa. Các phần mềm được viết bởi những ngôn ngữ lập trình khác nhau và chạy trên những nền tảng khác nhau có thể sử dụng dịch vụ Web để chuyển đổi dữ liệu thông qua mạng Internet theo cách giao tiếp tương tự bên trong một máy tính. Tuy nhiên, công nghệ xây dựng dịch vụ Web không nhất thiết phải là các công nghệ mới, nó có thể kết hợp với các công nghệ đã có như XML, SOAP, WSDL, UDDI... Với sự phát triển và lớn mạnh của Internet, dịch vụ Web thật sự là một công nghệ đáng được quan tâm để giảm chi phí và độ phức tạp trong tích hợp và phát triển hệ thống. Chúng ta sẽ xem xét các Web service từ mức khái niệm đến cách thức xây dựng.

## 2. Đặc điểm của Web Service

- Self-Contained (Độc lập)

Web service độc lập vì nó không đòi hỏi các tiến trình ở phía client phải cài đặt bất cứ một thành phần nào. Ở phía server, yêu cầu để triển khai (deploy) Web Service chỉ là servlet engine, EJB container hoặc .NET runtime. Khi Web Service đã được triển khai thì phía client có thể tiêu thụ các dịch vụ mà không cần đòi hỏi phải cài đặt bất cứ một thành phần nào. Trong khi đó với các công nghệ như DCOM hay RMI, phía client phải cài đặt client stub trước khi có thể truy cập dịch vụ.

- Tự mô tả

Giao diện của Web Service được xuất bản thông qua tài liệu WSDL. Tài liệu WSDL định nghĩa cấu trúc thông điệp trao đổi và cấu trúc dữ liệu sử dụng trong thông điệp đó. Để triệu gọi dịch vụ, client chỉ cần biết cấu trúc và nội dung của thông điệp yêu cầu và đáp ứng của Web service.

- Truy cập thông qua Web

Web được xuất bản, xác định và triệu gọi thông qua Web. Web service sử dụng giao thức chuẩn của web. Mô tả dịch vụ được xuất bản bằng cách sử dụng WSDL, các dịch vụ được xác định với sự giúp đỡ của UDDI registry và triệu gọi thông qua SOAP. Tất cả những giao thức này đều dựa trên web.

- Độc lập về Ngôn ngữ, Nền tảng, Giao thức

Web service có cơ sở là tiêu chuẩn mở XML. Một Client được viết bằng bất cứ ngôn ngữ cũng nào có thể truy cập một trang web service được viết bằng bất cứ ngôn ngữ nào khác. Web service độc lập nền tảng, client và web service có thể chạy trên hai nền tảng độc lập khác nhau.

- Dựa trên chuẩn Mở

Những chuẩn này là XML, SOAP, WSDL và UDDI.

- Web Service cho phép client và server tương tác được với nhau mặc dù trong những môi trường khác nhau.

- Web Service thì có dạng mở và dựa vào các tiêu chuẩn. XML và HTTP là nền tảng kỹ thuật cho web service. Phần lớn kỹ thuật của web service được xây dựng là những dự án nguồn mở. Bởi vậy, chúng độc lập và vận hành được với nhau.

- Web Service thì rất linh động: Vì với UDDI và WSDL, thì việc mô tả và phát triển web service có thể được tự động hóa.

- Web Service được xây dựng trên nền tảng những công nghệ đã được chấp nhận.
- Web Service có dạng mô đun.
- Web Service có thể công bố (publish) và gọi thực hiện qua mạng.

Ngày nay Web Service được sử dụng rất nhiều trong những lĩnh vực khác nhau của cuộc sống như:

- Tìm kiếm các thông tin về các khách sạn ở các thành phố hoặc các trung tâm để liên hệ đặt phòng theo yêu cầu của khách hàng.
- Dịch vụ chọn lọc và phân loại tin tức: Là những hệ thống thư viện kết nối đến các web portal để tìm kiếm các thông tin từ các nhà xuất bản có chứa những từ khóa muốn tìm.
- Dịch vụ hiển thị danh sách đĩa nhạc dành cho các công ty thu thanh.
- Ứng dụng đại lý du lịch có nhiều giá vé đi du lịch khác nhau do có chọn lựa phục vụ của nhiều hãng hàng không.
- Bảng tính toán chính sách bảo hiểm dùng công nghệ Excel/COM với giao diện web.
- Thông tin thương mại bao gồm nhiều nội dung, nhiều mục tin như: Dự báo thời tiết, thông tin sức khỏe, lịch bay, tỷ giá cổ phiếu...

### **3. Ưu, nhược điểm của Web Service**

#### **Ưu điểm:**

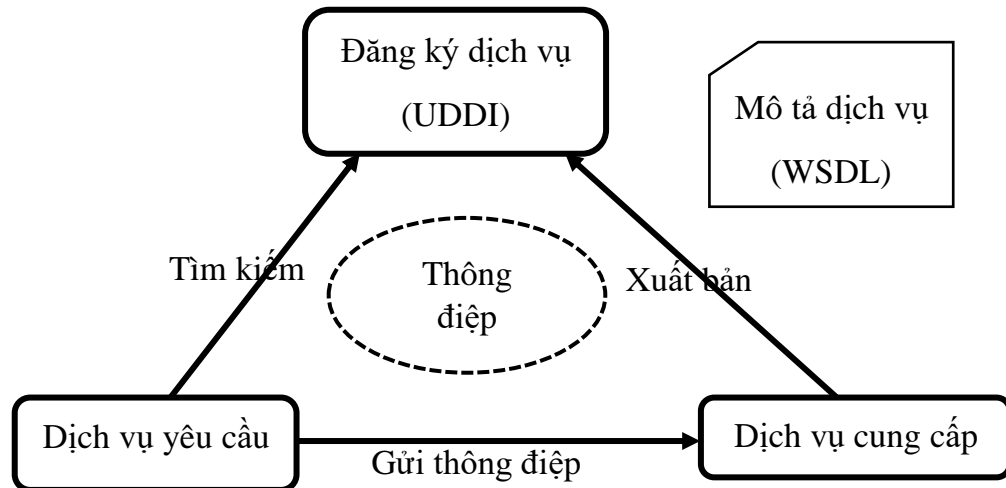
- Cho phép chương trình được viết bằng các ngôn ngữ khác nhau trên các nền tảng khác nhau giao tiếp được với nhau dựa trên một nền tảng tiêu chuẩn.
- Đơn giản (chỉ dùng URL).
- Làm việc với các giao thức chuẩn Web như XML, HTTP và TCP/IP.
- Sự an toàn của máy chủ cơ sở dữ liệu luôn được bảo mật một cách chắc chắn. Web Service làm giảm giá thành cho việc tích hợp các hệ thống khác nhau.
- Dịch vụ Web cung cấp khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau.

- Sử dụng các giao thức và chuẩn mở. Giao thức và định dạng dữ liệu dựa trên văn bản (text), giúp các lập trình viên dễ dàng hiểu được.
- Nâng cao khả năng tái sử dụng.
- Thúc đẩy đầu tư các hệ thống phần mềm đã tồn tại bằng cách cho phép các tiến trình/chức năng nghiệp vụ đóng gói trong giao diện dịch vụ Web.
- Tạo mối quan hệ tương tác lẫn nhau và mềm dẻo giữa các thành phần trong hệ thống, dễ dàng cho việc phát triển các ứng dụng phân tán.
- Thúc đẩy hệ thống tích hợp, giảm sự phức tạp của hệ thống, hạ giá thành hoạt động, phát triển hệ thống nhanh và tương tác hiệu quả với hệ thống của các doanh nghiệp khác.

#### **Nhược điểm:**

- Phụ thuộc vào tốc độ đường truyền Internet.
- Web Service thiếu cơ chế khôi phục đủ tin cậy để đảm bảo giao dịch được khôi phục lại trạng thái ban đầu trong trường hợp xảy ra sự cố.
- Số lượng các ứng dụng cộng tác cùng hoạt động sẽ ảnh hưởng tới hiệu suất tối ưu của Web Service.
- Tải trọng: ứng dụng Web Service là các ứng dụng sử dụng rất nhiều thông điệp. Khả năng bùng nổ số lượng giao dịch trao đổi sẽ làm hệ thống máy chủ ứng dụng và kiến trúc hạ tầng hệ thống thông tin của doanh nghiệp trở nên ngừng trệ.
- Vì Web Service đòi hỏi kết nối thông qua khá nhiều máy chủ trung gian cho nên băng thông/tốc độ của hạ tầng mạng và các yếu tố liên quan tới hệ thống rõ ràng có vai trò quan trọng góp phần cải thiện hiệu năng của toàn bộ các ứng dụng WS.
- Những thiệt hại lớn sẽ xảy ra vào khoảng thời gian chết của Dịch vụ Web, giao diện không thay đổi, có thể lỗi nếu một máy khách không được nâng cấp, thiếu các giao thức cho việc vận hành.
- Có quá nhiều chuẩn cho dịch vụ Web khiến người dùng khó nắm bắt.
- Phải quan tâm nhiều hơn đến vấn đề an toàn và bảo mật.

#### 4. Mô hình Web Service



Hình 1. Mô hình dịch vụ Web Service

- Nhà cung cấp đăng ký Web Service với UDDI.
- Người sử dụng tìm kiếm dịch vụ trên UDDI qua một URL thích hợp.
- UDDI trả lại một bản mô tả WSDL cho nhà cung cấp.
- Người sử dụng triệu gọi dịch vụ bằng một cuộc gọi SOAP tới nhà cung cấp.
- Nhà cung cấp trả lại kết quả của cuộc gọi SOAP cho người sử dụng.

#### 5. Các thành chính của Web Service

Sổ đăng ký
<b>UDDI</b>
Mô tả dịch vụ
<b>WSDL XML</b>
Giao thức truyền thống
<b>SOAP</b>
Giao thức giao vận
<b>HTTP</b>

Hình 2. Các thành phần chính của Web Service

XML được sử dụng để định dạng dữ liệu, SOAP được sử dụng trao đổi dữ liệu, WSDL được sử dụng để mô tả dịch vụ hiện có và UDDI được sử dụng để liệt kê các Web Service hiện có.

### ***5.1. Giao thức vận HTTP***

Tầng giao vận liên quan tới cơ chế sử dụng để chuyển yêu cầu dịch vụ và thông tin phản hồi từ phía nhà cung cấp dịch vụ tới người sử dụng dịch vụ. Có rất nhiều tiêu chuẩn sử dụng xung quanh Web Service, nhưng phổ biến nhất vẫn là giao thức HTTP.

#### **Ưu điểm:**

- HTTP là một giao thức phổ biến rộng rãi.
- Giao thức HTTP hoàn toàn mở và khai triển trên rất nhiều loại hệ thống.
- Hầu hết mọi tổ chức đều chấp nhận cho phép trao đổi thông tin dựa trên giao thức HTTP vượt qua tường lửa bảo vệ.

#### **Nhược điểm của HTTP:**

- HTTP là một giao thức đơn giản và không có tính trạng thái, và không được thiết kế đặc biệt cho mục đích vận chuyển dữ liệu của các ứng dụng.

### ***5.2. Giao thức truyền thông SOAP***

#### ***5.2.1. Khái niệm SOAP***

- SOAP là giao thức truyền thông giữa các ứng dụng.
- SOAP được thiết kế để liên lạc qua Internet và làm việc qua tường lửa.
- SOAP độc lập nền tảng, độc lập ngôn ngữ.
- SOAP dựa trên XML, đơn giản và dễ mở rộng.

#### ***5.2.2. Đặc trưng SOAP***

SOAP có những đặc trưng sau:

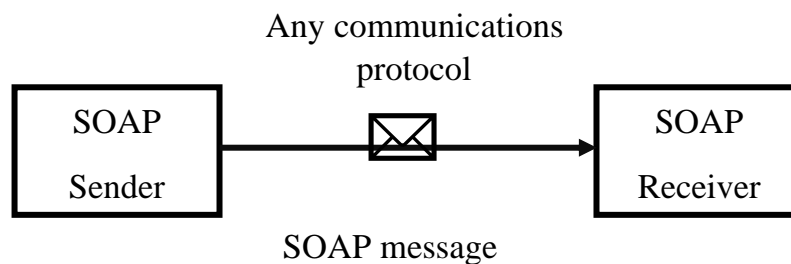
- SOAP được thiết kế đơn giản và dễ mở rộng.
- Tất cả các message SOAP đều được mã hóa sử dụng XML.
- SOAP sử dụng giao thức truyền dữ liệu riêng.
- Không có garbage collection phân tán, và cũng không có cơ chế tham chiếu. Vì thế SOAP client không giữ bất kỳ một tham chiếu đầy đủ nào về các đối tượng ở xa.

- SOAP không bị ràng buộc bởi bất kỳ ngôn ngữ lập trình nào hoặc công nghệ nào.

Vì những đặc trưng này, nó không quan tâm đến công nghệ gì được sử dụng để thực hiện miễn là người dùng sử dụng các message theo định dạng XML. Tương tự, service có thể được thực hiện trong bất kỳ ngôn ngữ nào, miễn là nó có thể xử lý được những message theo định dạng XML.

Khi trao đổi các thông điệp SOAP, có hai thành phần liên quan: Bên gửi và bên nhận. Thông điệp sẽ được chuyển từ bên gửi sang bên nhận. Đây là ý niệm đơn giản nhất trong trao đổi thông điệp SOAP. Trong nhiều trường hợp, kiểu trao đổi này không cung cấp đủ chức năng. Nhưng đây là mô hình cơ bản, dựa trên đó sẽ phát triển các mô hình trao đổi phức tạp hơn.

- SOAP là giao thức mà định nghĩa cái cách để chuyển một XML message từ A đến B dựa trên giao thức chuẩn web HTTP (hoạt động trên cổng 80) qua giao thức Internet TCP/IP.



Hình 3. Thông điệp SOAP

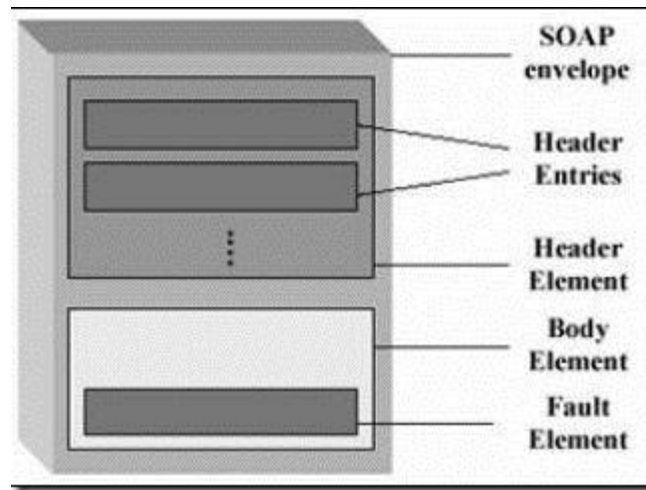
Tại sao phải có SOAP:

- Phát triển các ứng dụng cho phép các chương trình trao đổi qua Internet.
- Các ứng dụng liên lạc với nhau bằng cách sử dụng các cuộc gọi thủ tục ở xa giữa các đối tượng như DCOM, CORBA.
- SOAP cung cấp cách để liên lạc giữa các ứng dụng chạy trên các hệ điều hành khác nhau, với các công nghệ khác nhau và ngôn ngữ khác nhau.

### 5.2.3. Cấu trúc một message theo dạng SOAP

Cấu trúc một message theo dạng SOAP được mô tả như hình dưới đây:





Hình 4. Cấu trúc message SOAP

Message theo dạng SOAP là một văn bản XML bình thường gồm các phần tử sau:

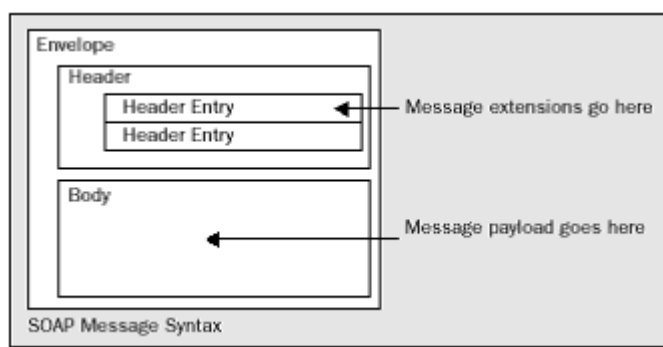
- Phần tử gốc - envelop: Phần tử bao trùm nội dung message, khai báo văn bản XML như là một thông điệp SOAP.
- Phần tử đầu trang – header: Chứa các thông tin tiêu đề cho trang, phần tử này không bắt buộc khai báo trong văn bản. Những đầu mục còn có thể mang những dữ liệu chứng thực, những chữ ký số hóa, và thông tin mã hóa, hoặc những cài đặt cho giao tác.
- Phần tử khai báo nội dung chính trong thông điệp - body, chứa các thông tin yêu cầu và phản hồi.
- Phần tử phát sinh lỗi (Fault) cung cấp thông tin lỗi xảy ra trong quá trình xử lý thông điệp.

Trong trường hợp đơn giản nhất, phần thân của SOAP message gồm có:

- Tên của message.
- Một tham khảo tới một thể hiện service.
- Một hoặc nhiều tham số mang các giá trị và mang các tham chiếu. Có 3 kiểu thông báo:
  - Request messages: Với các tham số gọi thực thi một service.
  - Response messages: Với các tham số trả về, sử dụng khi đáp ứng yêu cầu.
  - Fault messages báo tình trạng lỗi.

#### 5.2.4. Định dạng thông điệp SOAP

Một thông điệp SOAP là một văn bản XML được mô tả bởi một thành phần Envelope, chứa một thành phần Body bắt buộc và một thành phần Header không bắt buộc.



Hình 5. Cấu trúc thông điệp SOAP

Envelop là thành phần gốc của một thông điệp SOAP, nó chứa các thành phần Header và Body.

Thành phần Header là một cơ chế mở cho phép thêm các tính năng vào bên trong một thông điệp SOAP. Mỗi thành phần con của Header gọi là một Header Entry. Các Header Entry dùng để diễn giải, quy định một số ngữ nghĩa của thông điệp SOAP. Các ứng dụng có thể xử lý và định tuyến các thông điệp dựa trên thông tin header và thông tin bên trong thông điệp đó. Đây là ưu điểm mà các mô hình kiến trúc như DCOM, CORBA và RMI không có được, vì các protocol header của chúng phải được chỉ định chi tiết cho mỗi ứng dụng.

#### 5.2.5. Các kiểu truyền thông

SOAP hỗ trợ 2 kiểu truyền thông khác nhau:

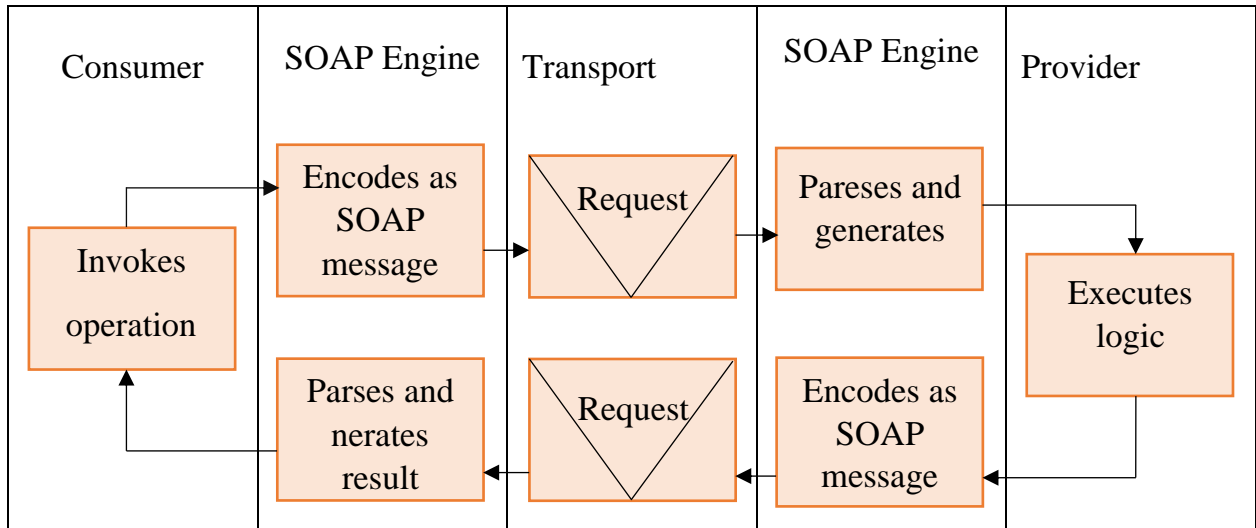
- Remote procedure call (RPC).
- Document.

#### 5.2.6. Quá trình xử lý thông điệp SOAP

Một thông điệp SOAP giúp cho khách hàng và nơi cung cấp Web Service hoàn thành những tác vụ mà không lo lắng đến sự phức tạp của việc xử lý thông điệp SOAP.

Một processor của khách hàng chuyển các lời yêu cầu phương thức vào trong một thông điệp SOAP. Thông điệp này được truyền qua tầng giao vận (HTTP và SMTP) tới processor của nơi cung cấp, tại đây thông điệp sẽ được phân tích thành lời yêu cầu

phương thức. Sau đó nơi cung cấp sẽ thực hiện những bước logic cần thiết và trả lại kết quả cho processor của nó, processor này sẽ phân tích thông tin trong thông điệp hồi đáp. Thông điệp này được truyền qua tầng giao vận tới khách hàng yêu cầu. Processor của nó phân tích thông điệp hồi đáp thành kết quả dưới dạng một đối tượng.



Hình 6. Quá trình xử lý thông điệp SOAP

### 5.3. Tầng mô tả dịch vụ XML, WSDL

#### 5.3.1. XML (eXtensible Markup Language)

##### a. Khái niệm XML:

XML là nền tảng của Web Service và được dùng để trao đổi dữ liệu.

XML là một chuẩn nổi tiếng cho việc tổ chức, lưu trữ và trao đổi dữ liệu.

XML được hỗ trợ bởi hầu hết các ngôn ngữ lập trình hiện đại (DotNet, Java...).

XML được sử dụng rộng rãi trong việc trao đổi dữ liệu trên môi trường Internet.

XML dùng các thẻ để tổ chức và lưu trữ dữ liệu.

##### b. Đặc điểm của XML:

XML là tự do và mở rộng được.

XML rất quan trọng đối với sự phát triển của web trong tương lai.

Tầm quan trọng của XML đối với tương lai của web cũng giống như tầm quan trọng của HTML đối với nền tảng của web, và XML sẽ là công cụ xử lý và truyền dữ liệu phổ biến nhất.

XML là công cụ dùng được trên mọi nền phần cứng, độc lập với phần cứng và phần mềm để truyền (trao đổi, chia sẻ) thông tin.

### ***c. Cấu trúc chung của XML***

Chúng ta có thể sử dụng trình soạn thảo bất kỳ để soạn thảo tài liệu XML, nhưng phải tuân thủ theo nguyên tắc sau:

```
<root>
  <child>
    <subchild>...</subchild>
    ....
  </child>
  ....
</root>
```

Theo định dạng trên, chúng ta thấy tuy tài liệu XML rất đơn giản nhưng quy định cũng rất chặt chẽ, tức là các tài liệu XML đều xuất phát từ nút gốc (root), và mỗi phần tử phải có thẻ mở và thẻ đóng “<tên thẻ > ... </ tên thẻ>”.

### ***d. XML được sử dụng như thế nào?***

XML thiết kế để lưu giữ, mang, và trao đổi dữ liệu nhưng không hiển thị dữ liệu.

XML dùng cho trao đổi dữ liệu.

Với XML có thể trao đổi dữ liệu giữa các hệ thống không tương thích.

### ***e. Ưu điểm của XML***

Đơn giản, ổn định, linh hoạt và có tính mở rộng cao.

XML được chấp nhận rộng rãi. Rất nhiều công cụ và tiện ích sẵn có đáp ứng nhu cầu phân tích và chuyển đổi dữ liệu XML hoặc hiển thị chúng.

### ***f. Nhược điểm của XML***

Sự phức tạp.

Chuẩn hoá: Trong khi đã tồn tại các định nghĩa tên thẻ của ngành, bạn vẫn có thể định nghĩa các thẻ không phải là tiêu chuẩn.

Dung lượng lớn.

### **5.3.2. WSDL (*Web Service Description Language*)**

#### **a. Định nghĩa**

WSDL định nghĩa cách mô tả Web Service theo cú pháp tổng quát XML, bao gồm các thông tin:

- Tên service.
- Giao thức và kiểu mã hóa.
- Loại thông tin: những thao tác, những tham số, ...
- WSDL chỉ định các đặc tính vận hành của Web Service sử dụng một tài liệu XML. Ngôn ngữ mô tả những khái niệm trả lời cho các câu hỏi sau:
  - o Cái gì (dịch vụ web làm gì)?
  - o Ở đâu (nơi chứa dịch vụ)?
  - o Như thế nào (dịch vụ có thể kích hoạt bằng cách nào)?

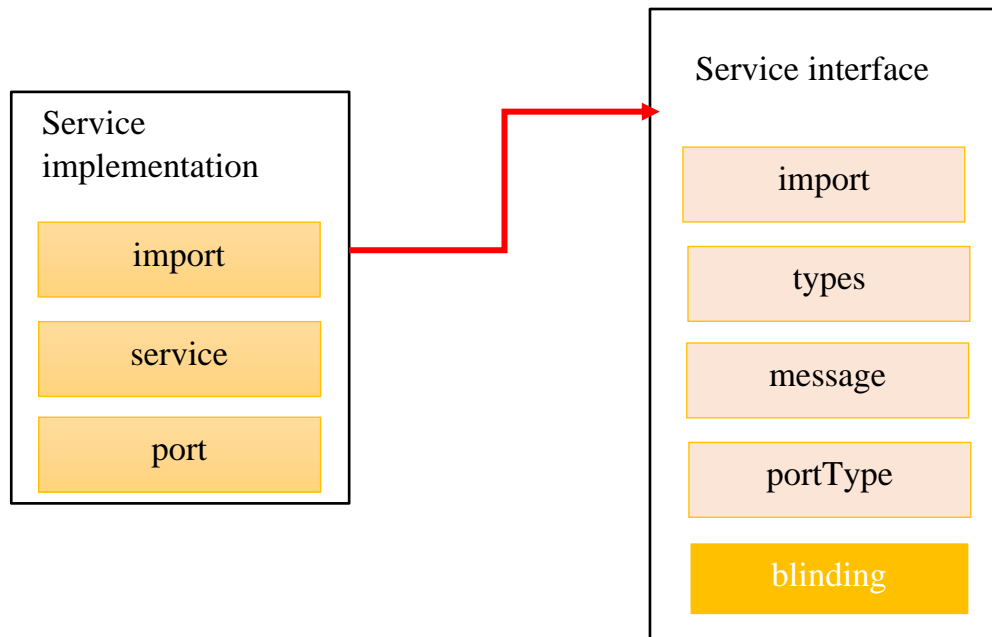
#### **b. Cấu trúc WSDL -*Web Services Description Language***

**Một WSDL hợp lệ gồm có hai phần:**

- Phần giao diện mô tả giao diện và giao thức kết nối.
- Phần thi hành mô tả thông tin để truy xuất service.

Cả 2 phần trên được lưu trong 2 tập tin XML, bao gồm:

- Tập tin giao diện service (cho phần 1).
- Tập tin thi hành service (cho phần 2).



Hình 7. Service Interface và Service Implementation.

**c. Ưu điểm của WSDL:**

Như một yêu cầu cơ bản đối với ứng dụng của bất cứ dịch vụ web, WSDL là yêu cầu bắt buộc đáp ứng nhu cầu công bố giao tiếp và thoả thuận cho các dịch vụ khác kích hoạt.

**d. Nhược điểm của WSDL:**

Tài liệu không cung cấp một số thông tin người sử dụng có nhu cầu như:

- Ai cung cấp dịch vụ ?
- Loại hình kinh doanh cung cấp dịch vụ?
- Các dịch vụ khác cùng do nhà cung cấp dịch vụ này cung cấp?
- Dịch vụ này sẽ cung cấp với chất lượng dịch vụ như thế nào?
- Đây là dịch vụ miễn phí hay có thu phí ?

## 5.4. Universal Discovery Description and Intergration

### 5.4.1. Khái niệm UDDI

UDDI là một chuẩn công nghiệp cho việc công bố và tìm kiếm thông tin về Web Service. Nó định nghĩa một khung thông tin cho phép bạn mô tả và phân loại tổ chức của bạn, dịch vụ của nó và những chi tiết kỹ thuật về giao diện của Web Service mà bạn trình bày. Khung thông tin này cho phép bạn phát hiện dịch vụ một cách thích hợp, hay giao diện của một kiểu đặc biệt, một loại hay của một hàm.

### 5.4.2. Nội dung của thư mục UDDI

Một nội dung thư mục UDDI là một tệp XML mô tả một nghiệp vụ và các dịch vụ nó chào. Có 3 phần đối với một nội dung trong thư mục UDDI, đó là:

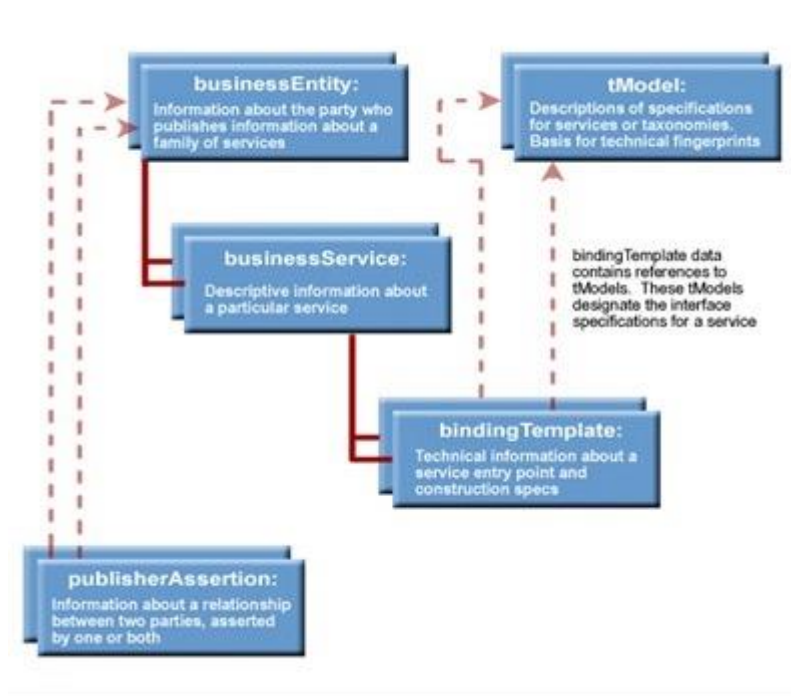
**Trang vàng -Yellow pages:** mô tả công ty chào dịch vụ: tên, địa chỉ, các thông tin liên hệ.

**Trang vàng -Yellow pages:** chứa thông tin mô tả Web Service theo những chủng loại khác nhau. Những thông tin này cho phép các đối tượng thấy Web Service theo từng chủng loại của nó.

**Trang xanh-Green Pages:** mô tả giao diện đối với dịch vụ một cách đầy đủ chi tiết cho ai đó có thể viết một ứng dụng sử dụng dịch vụ Web Service này.

### 5.4.3. Cấu trúc sổ đăng ký UDDI

UDDI cung cấp 4 cấu trúc dữ liệu mô tả dịch vụ mà nó đưa ra: BusinessEntity, BusinessService, BindingTemplate và tModels.



Hình 8. Mô hình UDDI

Như hình trên, cấu trúc thông tin bao gồm :

- **BusinessEntity:** mô tả nhà cung cấp dịch vụ, mỗi BusinessEntity có thể có nhiều cấu trúc BusinessService kết hợp với nó.

Mỗi doanh nghiệp được cấp một định danh duy nhất, ví dụ như số D-U-N-S.

- **BusinessService:** Liên kết với mỗi BusinessEntity là một danh sách các BusinessService cung cấp bởi BusinessEntity đó. Mỗi thành phần chứa thông tin mô tả về dịch vụ, về thông tin phân loại của dịch vụ và danh sách các BindingTemplate liên quan đến thông tin kỹ thuật của dịch vụ. Mỗi BusinessService cần có ít nhất một BindingTemplate.
- **BindingTemplate:** Gắn với mỗi BusinessService là một danh sách các BindingTemplate cung cấp thông tin về địa điểm có thể tìm thấy Web Service và làm cách nào để sử dụng nó.

Một cấu trúc BindingTemplate mô tả thông tin interface của Web Service và các địa chỉ URL. Mỗi BindingTemplate được định danh duy nhất thông qua số phát sinh tự động UUID lưu trong bindingKey.

- **TModel (Technical Model-Mô hình kỹ thuật):** Mục đích của tModels là dùng để liên kết đến metadata bên ngoài UDDI. Thành phần quan trọng nhất của tModels là một URL trỏ đến một tài liệu mô tả thông tin metadata. Tài liệu này có thể là tài liệu bất kỳ HTML, Word, ... tùy ý mô tả một đặc tả kỹ thuật nào đó, ví dụ như giao thức mạng, dạng thức trao đổi hoặc luật tuần tự mà thông thường nhất là file mô tả thông tin service WSDL. Có hai thuộc tính cơ bản bên trong một tModel :tModelKey đóng vai trò định danh duy nhất giữa các tModel với nhau và name dùng cung cấp một tên với đầy đủ ngữ nghĩa cho tModel.

## 6. Web Service hoạt động như thế nào?

Một ứng dụng Web Service bao gồm 2 thành phần: Client và Server giao tiếp với nhau qua giao thức HTTP.

- Client gửi yêu cầu qua các lời gọi hàm thông qua HTTP Request đến Server.
- Server gửi các kết quả được thực thi các ở hàm thông qua HTTP Request.

Mô hình hoạt động của ứng dụng Web Service gồm 3 thành phần chính:

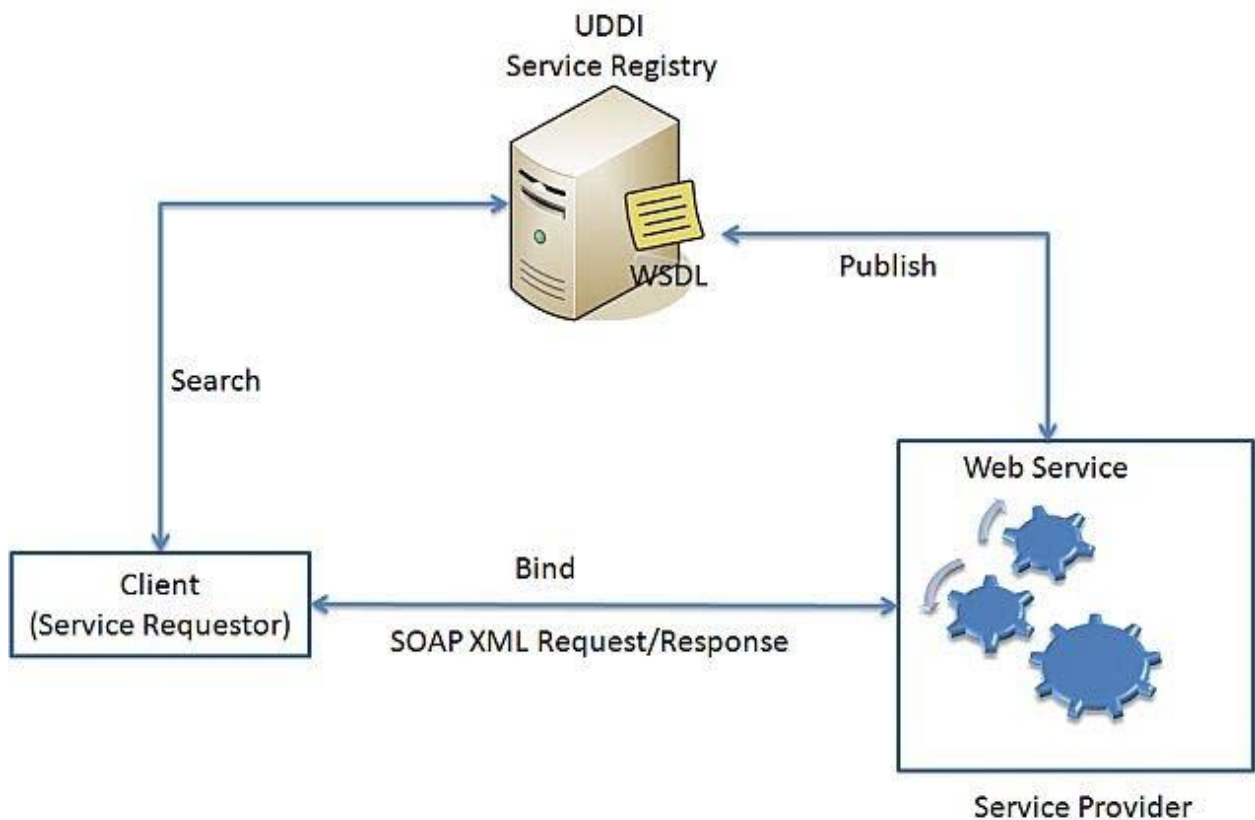
- **UDDI service registry:** Công cụ giúp nhà phát triển Web Service công bố những thông tin về Web Service của mình cho cộng đồng các nhà phát triển ứng dụng. Người



dùng sẽ dựa vào các thông tin này để sử dụng Web Service trong ứng dụng riêng của mình.

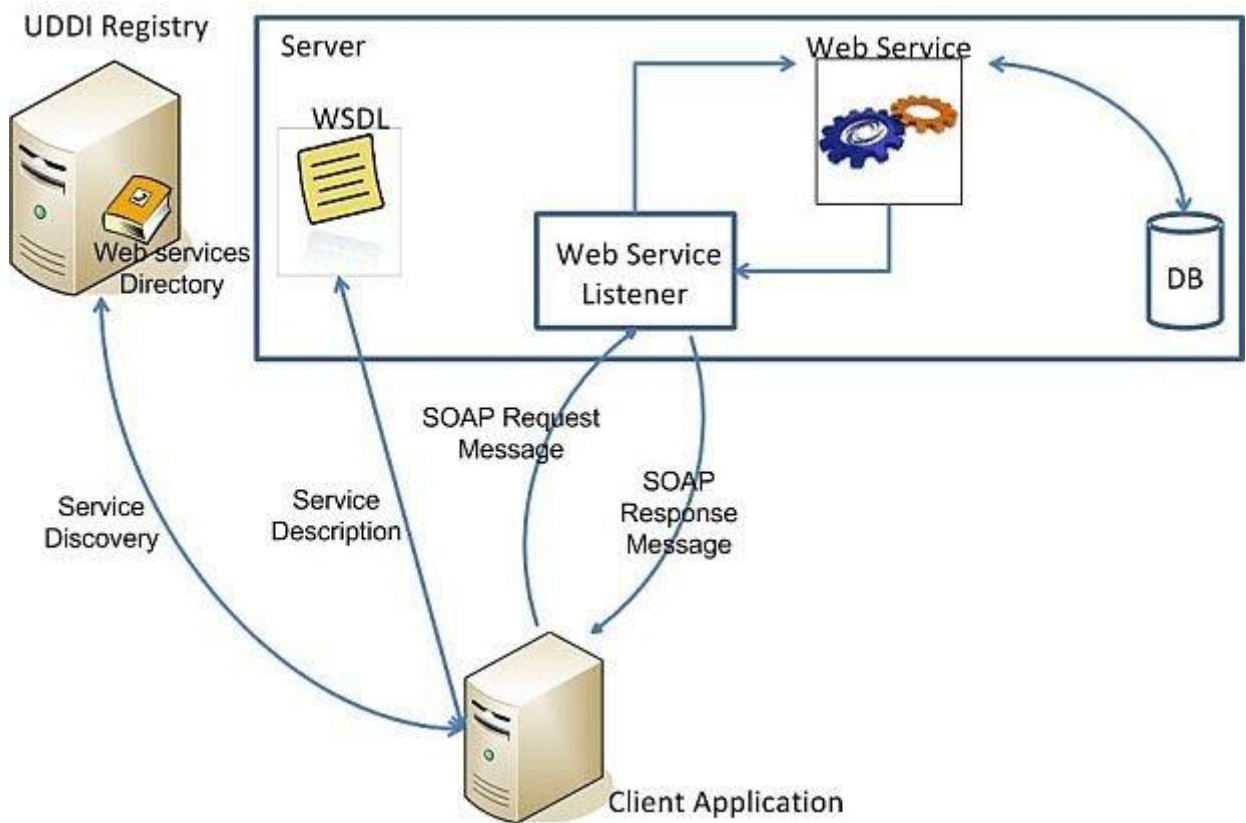
- **Web Service:** Chứa giao thức SOAP định dạng dữ liệu, tài liệu WSDL định nghĩa các hàm trong Web Service, XML để xây dựng ứng dụng phân tán.
- **Application Client:** Ứng dụng phía Client sử dụng Web Service xây dựng riêng cho mình.

Cách thức hoạt động có thể mô tả như sau: Đầu tiên, Application Client cần truy vấn các mẫu tin.



Hình 9. Mô hình hoạt động của Web Service

UDDI theo 1 thông tin nào đó (chẳng hạn tên loại) để xác định Web Service cần tìm. Khi đã xác định được Web Service cần cho ứng dụng, Client có thể lấy thông tin về địa chỉ của tài liệu WSDL của Web Service này dựa trên mẫu tin UDDI. Tài liệu WSDL sẽ mô tả cách thức liên lạc với Web Service, định dạng gói tin truy vấn và phản hồi. Dựa vào những thông tin này, Client có thể tạo những gói tin SOAP tương ứng để liên lạc với Service.



Hình 10. Cách thức hoạt động của Web Service

## 7. Các loại Web Service

Có 2 loại web service chính:

- SOAP Web Services.
- RESTful Web Services.

### 7.1. SOAP Web Service là gì?

SOAP là viết tắt của **S**imple **O**bject **A**ccess **P**rotocol.

SOAP (Simple Object Access Protocol) là giao thức sử dụng **XML** để định nghĩa dữ liệu dạng thuần văn bản (plain text) thông qua HTTP. SOAP là cách mà Web Service sử dụng để truyền tải dữ liệu. Vì dựa trên XML nên SOAP là một giao thức không phụ thuộc platform cũng như bất kì ngôn ngữ lập trình nào. Chúng ta có thể viết bằng Java, PHP, .NET, ... và triển khai trên Window, Linux, ...

### 7.2. RESTful Web Service là gì?

REST là viết tắt của **R**epresentational **S**tate **T**ransfer

REST là một loại kiến trúc phần mềm (architectural style), không phải là một protocol.

RESTful Web Service là các Web Service được viết dựa trên kiến trúc REST. REST đã được sử dụng rộng rãi để thay thế cho các Web Service dựa trên SOAP và WSDL.

Tương tự SOAP, REST cũng không phụ thuộc platform cũng như bất kỳ ngôn ngữ lập trình nào.

REST có thể sử dụng SOAP web service như một implement của nó.

REST định nghĩa các quy tắc kiến trúc để bạn thiết kế Web services, chú trọng vào tài nguyên hệ thống, bao gồm các trạng thái tài nguyên được định dạng như thế nào và được truyền tải qua HTTP, và được viết bởi nhiều ngôn ngữ khác nhau. Nếu tính theo số dịch vụ mạng sử dụng, REST đã nổi lên trong vài năm qua như là một mô hình thiết kế dịch vụ chiếm ưu thế. Trong thực tế, REST đã có những ảnh hưởng lớn và gần như thay thế SOAP và WSDL vì nó đơn giản và dễ sử dụng hơn rất nhiều.

REST tuân thủ 4 nguyên tắc thiết kế cơ bản sau:

- Sử dụng các phương thức HTTP một cách rõ ràng.
- Phi trạng thái.
- Hiển thị cấu trúc thư mục như các URLs.
- Truyền tải JavaScript Object Notation (JSON), XML hoặc cả hai.

#### ***7.2.1. Sử dụng các phương thức HTTP một cách rõ ràng***

REST đặt ra một quy tắc đòi hỏi lập trình viên xác định rõ hành động của service thông qua các phương thức của HTTP.

Các hành động của một Web Service thông thường bao gồm tạo dữ liệu (**Create**), lấy dữ liệu (**Read**), cập nhập dữ liệu (**Update**) hoặc xóa dữ liệu (**Delete**). Các hành động này còn được gọi là **CRUD**.

Thiết lập một ánh xạ 1-1 giữa các hành động (CRUD) và các phương thức **HTTP**:

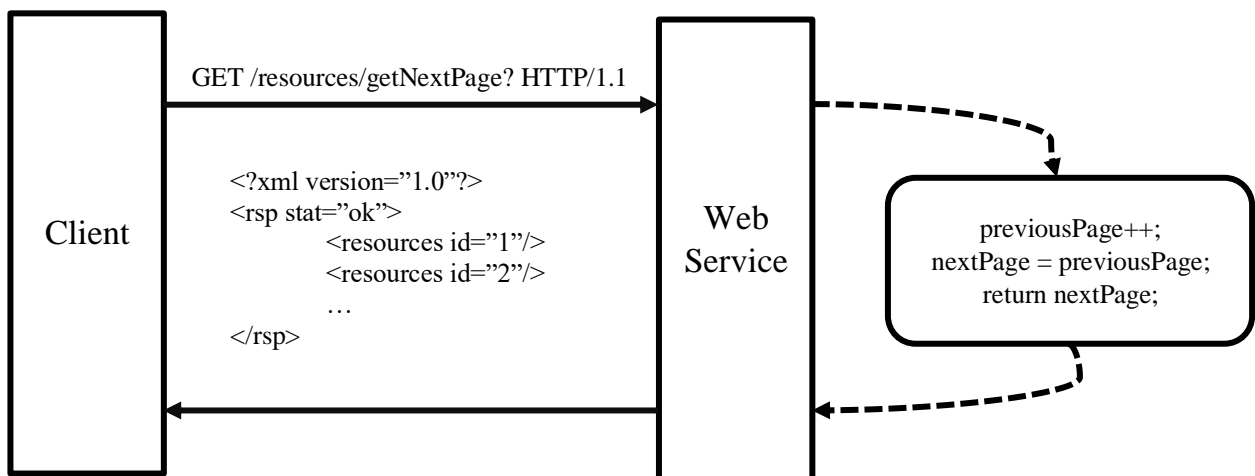
- **POST**: để tạo một tài nguyên trên Server.
- **GET**: để truy xuất một tài nguyên.
- **PUT**: để thay đổi trạng thái một tài nguyên hoặc để cập nhật nó.
- **DELETE**: để huỷ bỏ hoặc xóa một tài nguyên.

Lưu ý: các nguyên tắc ở trên là không bắt buộc, trên thực tế chúng ta có thể sử dụng phương thức GET để yêu cầu lấy, tạo, sửa hoặc xóa dữ liệu trên Server. Tuy nhiên, REST đưa ra các nguyên tắc ở trên mục đích đưa mọi thứ trở lên rõ ràng và dễ hiểu.

### 7.2.2. Phi trạng thái (Stateless)

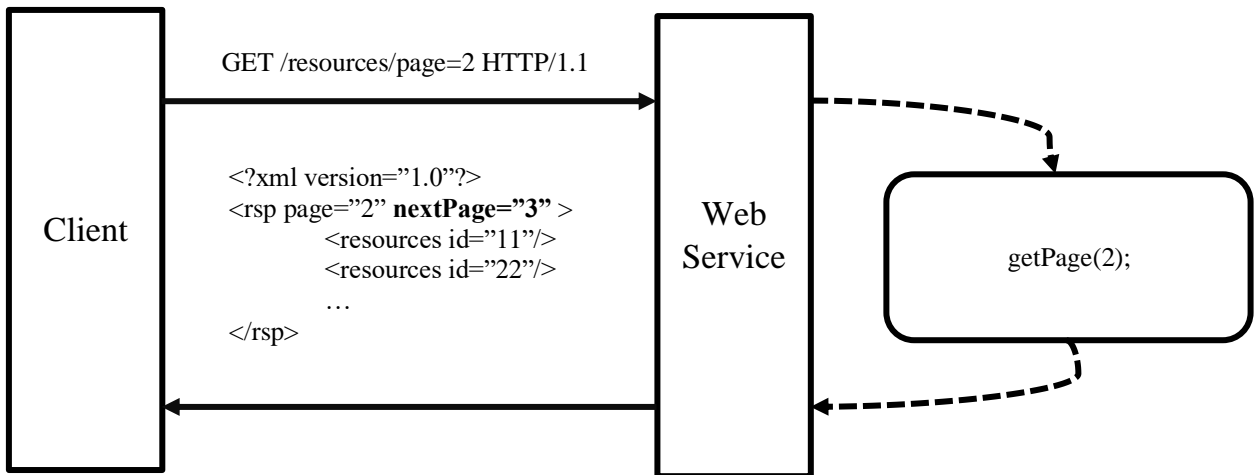
Một đặc điểm của REST là phi trạng thái (stateless), có nghĩa là nó không lưu giữ thông tin của client. Chẳng hạn bạn vừa gửi yêu cầu để xem trang thứ hai của một tài liệu, và bây giờ bạn muốn xem trang tiếp theo (sẽ là trang ba). REST không lưu trữ lại thông tin rằng trước đó nó đã phục vụ bạn trang số hai. Điều đó có nghĩa là REST không quản lý phiên làm việc (Session).

Hình dưới đây minh họa một ứng dụng có lưu trữ trạng thái, nó biết người dùng đang xem trang số mấy. Và người dùng chỉ cần yêu cầu “Trang Tiếp theo” để nhận được trang mong muốn.



Hình 11. Stateful

Với các thiết kế phi trạng thái, Client phải gửi yêu cầu rõ ràng, bao gồm số thứ tự của trang cần xem.



Hình 12. Stateless

Như vậy, các thành phần Server phi trạng thái ít phức tạp hơn để thiết kế, viết và phân bổ thông qua Server được cân bằng tải. Dịch vụ phi trạng thái không chỉ hoạt động tốt hơn, nó còn chuyển hầu hết vai trò duy trì trạng thái sang ứng dụng ở Client. Trong một dịch vụ mạng RESTful, Server chịu trách nhiệm đưa ra các phản hồi và cung cấp một cách thức cho phép Client duy trì trạng thái ứng dụng của chính nó.

### 7.2.3. *Đưa ra cấu trúc thư mục giống các URL*

REST đưa ra một cấu trúc để người dùng có thể truy cập vào tài nguyên của nó thông qua các URL, tài nguyên ở đây là tất cả những cái mà bạn có thể gọi tên được (Video, ảnh, báo cáo thời tiết,...)

Bạn cần tạo ra các REST service để nó trả về cho người dùng các nguồn tài nguyên tương ứng.

Các địa chỉ REST service cần phải thật trực quan đến mức người dùng dễ đoán. Hãy nghĩ về một địa chỉ (URI) giống như một gợi ý rõ ràng, dễ đoán rằng nó đang trỏ tới cái gì và cung cấp tài nguyên gì. Tóm lại, cấu trúc của một URI nên được đơn giản, có thể dự đoán, và dễ hiểu.

Hãy xem một URL dưới đây, nó cung cấp danh sách bài viết của một ngày cụ thể, và nó dễ hiểu đối với người dùng.

<https://gpcoder.com/posts/2019-05-20>

Một vài nguyên tắc bổ sung để lưu ý trong khi nói về cấu trúc địa chỉ của RESTful Web service là:

- Giấu các đuôi tài liệu mở rộng của bản gốc trong máy chủ (.jsp, .php, .asp), nếu có, vì vậy bạn có thể giấu một số thứ mà không cần thay đổi địa chỉ Urls.
- Để mọi thứ là chữ thường.
- Thay thế các khoảng trống bằng gạch chân hoặc gạch nối (một trong hai loại).
- Tránh các chuỗi yêu cầu càng nhiều càng tốt.
- Thay vì sử dụng mã (404 Not Found) khi yêu cầu địa chỉ cho một phân đường dẫn, luôn luôn cung cấp một trang mặc định hoặc tài nguyên như một phản hồi.

#### 7.2.4. Truyền tải XML, JSON hoặc cả hai

Khi Client gửi một yêu cầu tới web service nó thường được truyền tải dưới dạng XML hoặc JSON và thông thường nhận về với hình thức tương tự.

Đôi khi Client cũng có thể chỉ định kiểu dữ liệu nhận về mà nó mong muốn (JSON, hoặc XML,...), các chỉ định này được gọi là các kiểu MIME, nó được gửi kèm trên phần HEADER của request.

Dưới đây là các kiểu MIME phổ biến thường sử dụng với REST service:

x	Content-Type
.json	application/json
.xml	application/xml

Tham khảo thêm các MIME type khác: <https://www.freeformatter.com/mime-types-list.html>

Ví dụ: Client gửi một yêu cầu để lấy thông tin danh sách bài viết, và yêu cầu dữ liệu trả về là định dạng XML.

```
GET https://gpcoder.com/posts
authority: gpcoder.com
Accept: application/xml;q=0.9
```

Và nhận được dữ liệu:

```
<posts>
  <post>
    <date>2019/05/20</date>
    <title>Java Web service tutorial</title>
    <content>...</content>
  </post>
  <post>
    <date>2019/05/25</date>
    <title>SOAP Web Service</title>
    <content>...</content>
  </post>
  <post>
    <date>2019/05/30</date>
    <title>RESTful Web Service</title>
    <content>...</content>
  </post>
</posts>
```

### 7.3. Sự khác nhau giữa REST và SOAP

#Tiêu chí	SOAP	REST
1. Viết tắt	SOAP là viết tắt của Simple Object Access Protocol (Giao thức truy cập đối tượng đơn giản)	REST là viết tắt của REpresentational State Transfer (Chuyển giao trạng thái phản hồi)
2. Kiến trúc và giao thức	Một giao thức gửi nhận message có dạng XML.	Một loại kiến trúc bao gồm các quy tắc để thao tác với server.
3. Các dạng định dạng được hỗ trợ	SOAP là một chuẩn được tạo ra để chuẩn hóa giao tiếp giữa client và server về format, structure và method. Sử dụng WSDL để giao tiếp giữa máy chủ và máy khách. SOAP sử dụng các message dạng XML để giao tiếp với server.	Sử dụng XML hoặc JSON để gửi nhận dữ liệu.
4. Tài nguyên	SOAP sử dụng các message dạng XML để xác định các thủ tục hay tài nguyên yêu cầu. Gọi các service thông qua method RPC.	RESTful sử dụng <b>URL</b> để xác định tài nguyên mong muốn được truy cập.
5. Kết quả trả về	Không dễ đọc.	Dễ đọc vì đơn giản chỉ là text XML hoặc JSON.
6. Sử dụng giao thức HTTP	Có thể truyền qua nhiều giao thức khác nhau như HTTP, SMTP, FTP,... Tuy nhiên trong thực tế chúng ta cũng sử dụng giao thức HTTP nhiều hơn nên lợi thế này của SOAP cũng không mấy được coi trọng.	REST chỉ có thể truyền qua HTTP và nó thừa hưởng tất cả những lợi ích của giao thức HTTP, bao gồm các method như GET, POST, PUT và DELETE để thực hiện các hành động truy vấn, thêm, sửa, xóa dữ liệu.

<b>7. Client</b>	JS có thể dùng để gọi SOAP, nhưng rất khó để làm.	Quá đơn giản nếu dùng JS.
<b>8. Performance</b>	Performance không tốt bằng REST.	Performance tốt hơn SOAP, tốn ít tài nguyên CPU hơn, code ngắn gọn hơn.
<b>9. Băng thông</b>	Các message SOAP thường có độ dài và dung lượng cao hơn so với một request của RESTful, do đó nếu dùng SOAP thì bạn sẽ tốn băng thông nhiều hơn.	Tốn ít băng thông hơn SOAP.
<b>10. Bảo mật</b>	SOAP cần viết thêm các cơ sở hạ tầng để bảo mật các message và giao thức vận chuyển.	RESTful Web Service có thể được triển khai bằng các giải pháp và tiêu chuẩn truyền thống để các truy cập được phân quyền và xác thực trước khi sử dụng tài nguyên web.
<b>11. Caching</b>	Các dịch vụ web SOAP hoàn toàn bỏ qua caching web.	RESTful web service tận dụng tối đa cơ chế caching vì về cơ bản chúng dựa trên URL.
<b>12. Tiếp cận</b>	Các dịch vụ web SOAP, mọi thực thể đều tập trung vào các interface và message.	Các dịch vụ web dựa trên REST, mọi thực thể đều tập trung vào các tài nguyên.

## 8. Lý do sử dụng REST thay vì SOAP?

- REST có thể được sử dụng bởi bất kỳ client nào ví dụ: Java, C ++, Python client và thậm chí là một trình duyệt web với Ajax và JavaScript.
- REST nhẹ hơn so với SOAP, nó không yêu cầu phân tích cú pháp XML và nó cũng tiêu tốn ít băng thông hơn vì không giống như SOAP, REST không yêu cầu SOAP header cho mỗi lần request.
- SOAP là một công nghệ cũ, còn tất cả những gã khổng lồ hiện đại đang sử dụng REST, ví dụ: Google, Twitter và Flickr...
- REST rất dễ học, nó chỉ cần hiểu danh từ và động từ. Nếu bạn đã biết các phương thức HTTP thì nó thậm chí còn dễ dàng hơn.
- Java hỗ trợ tuyệt vời cho RESTful web service và cũng hỗ trợ tốt cho SOAP web service. Bạn có rất nhiều sự lựa chọn ở đây, ví dụ: Jersey, RESTlet, ...

## 9. Lý do sử dụng SOAP?

Mặc dù REST có rất nhiều ưu điểm hơn so với SOAP. Tuy nhiên, có một số lý do có thể khiến bạn muốn lựa chọn SOAP cho dịch vụ web của mình:



- **WS-Security:** SOAP không chỉ hỗ trợ SSL (giống như REST) mà còn hỗ trợ WS-Security, bổ sung thêm một số tính năng enterprise security. Hỗ trợ nhận dạng thông qua các trung gian, không chỉ là point-to-point như SSL. Nó được dùng khi muốn xây dựng những web service đảm bảo và tin cậy. Web Service Security đảm bảo cho tính an toàn, sự toàn vẹn thông điệp và tính tin cậy của thông điệp.
- **WS-AtomicTransaction:** Khi muốn có các giao dịch ACID qua một dịch vụ, bạn sẽ phải cần SOAP. Mặc dù REST có hỗ trợ các transactions, nhưng nó không toàn diện và cũng không phù hợp với ACID. REST bị hạn chế bởi HTTP nên không thể cung cấp cam kết hai pha trên các tài nguyên giao dịch phân tán, nhưng SOAP lại có thể làm được điều này. Thật may mắn các giao dịch ACID gần như không có ý nghĩa nhiều đối với các dịch vụ internet thông thường. Nhưng đôi khi các ứng dụng doanh nghiệp lại cần mức độ tin cậy giao dịch này.
- **WS-ReliableMessaging:** REST không có hệ thống báo lỗi chuẩn và mong muốn khách hàng giải quyết các lỗi communicate bằng cách retry và ... retry ... SOAP đã thành công trong việc xử lý những tình huống này và cung cấp end-to-end một cách tin cậy thông qua các trung gian SOAP.

SOAP rõ ràng là hữu ích và quan trọng. Ví dụ, nếu bạn viết một ứng dụng để giao tiếp với ngân hàng chắc chắn bạn sẽ cần phải sử dụng SOAP. Tất cả ba tính năng trên là bắt buộc đối với các giao dịch ngân hàng.

Ví dụ: nếu tôi chuyển tiền từ tài khoản này sang tài khoản khác, tôi cần phải chắc chắn rằng nó đã hoàn tất. Việc cứ cố gắng retry thực sự là quá kinh dị nếu giao dịch thành công lần đầu tiên nhưng thông báo tôi nhận được lại là thất bại.

## 10. Vấn đề an toàn cho Web Service

Dịch vụ Web liên kết và tương tác với các ứng dụng qua Internet, chính vì vậy bảo mật là một vấn đề được quan tâm khi các công ty tiến tới kết hợp ứng dụng với một dịch vụ Web. Việc đảm bảo an toàn cho dịch vụ Web là một vấn đề quan trọng, đặc biệt đối với những dịch vụ liên quan đến trao đổi tiền tệ, thông tin từ thị trường chứng khoán hay dịch vụ bán hàng qua mạng (liên quan đến trả tiền bằng tài khoản và có yêu cầu thông tin cá nhân của người dùng).

Trước khi có WS-Security (bảo mật cho dịch vụ Web) thì ý nghĩa thông thường của an toàn dịch vụ Web là bảo mật kênh truyền dữ liệu. Hiện nay, nó được thực hiện cho những SOAP/HTTP dựa trên cơ chế truyền thông điệp bằng cách sử dụng giao thức HTTPS. Không chỉ là an toàn ở mức truyền thông điệp, HTTPS còn cung cấp sự an toàn tới toàn bộ gói dữ liệu HTTP.

Mặc dù HTTPS không bao gồm tất cả các khía cạnh trong chuẩn an toàn chung cho dịch vụ Web nhưng nó đã cung cấp một lớp bảo mật khá đầy đủ với định danh, chứng thực, tính toàn vẹn thông điệp hay độ tin cậy.

#### ❖ **Đảm bảo an toàn cho dịch vụ Web:**

**Khái niệm về WS-Security:** đây là một chuẩn an toàn bao trùm cho SOAP, nó được dùng khi muốn xây dựng những dịch vụ Web toàn vẹn và tin cậy. Toàn vẹn có nghĩa là khi có một giao dịch hay khi truyền thông tin, hệ thống và thông tin sẽ không bị chặn, giao dịch sẽ không bị mất cũng như không thể có người lấy cắp được dữ liệu trên đường truyền. WS-security được thiết kế mang tính mở nhằm hướng tới những mô hình an toàn khác bao gồm PKI, Kerberos và SSL. Nó cũng đưa ra nhiều hỗ trợ cho các cơ chế an toàn khác, nhiều khuôn dạng chữ ký và công nghệ mã hóa, đảm bảo sự an toàn, toàn vẹn thông điệp và tính tin cậy của thông điệp. Tuy nhiên, WS-security cũng chưa thể đảm bảo được tất cả yêu cầu về bảo mật và an toàn thông tin, nó chỉ là một trong những lớp của giải pháp an toàn cho dịch vụ Web.

Tính toàn vẹn tạo ra một chữ ký số hóa XML dựa trên nội dung của thông điệp. Nếu dữ liệu bị thay đổi bất hợp pháp, nó sẽ không còn thích hợp với chữ ký số hóa XML đó. Chữ ký này được tạo ra dựa trên khóa mà người gửi thông điệp tạo ra, do đó người nhận chỉ nhận thông điệp khi có chữ ký sử dụng và nội dung phù hợp. Ngược lại sẽ có một thông báo lỗi. Việc chứng thực được thực hiện giữa client và server là cách chứng thực rất cơ bản (sử dụng định danh người dùng và mật khẩu).

WS-security chỉ là một trong những lớp an toàn và bảo mật cho dịch vụ Web, vì vậy cần một mô hình an toàn chung lớn hơn để có thể bao quát được các khía cạnh khác. Các thành phần được thêm có thể là WS-Secure Conversation Describes, WS-Authentication Describes, WS-Policy Describes hay WS-Trust Describes. Chúng sẽ thực hiện việc đảm

bảo an toàn hơn cho hệ thống khi trao đổi dữ liệu, mở và đóng các phiên làm việc cũng như quản lý dữ liệu cần chứng thực và chính sách chứng thực.

## **11. Mô hình của ứng dụng Web Service**

### ***11.1. Xây dựng một Web Service***

Có 4 giai đoạn chính để xây dựng một dịch vụ Web là xây dựng, triển khai, tiến hành và quản lý, trong đó:

- **Giai đoạn xây dựng:** bao gồm phát triển và chạy thử ứng dụng dịch vụ Web, xây dựng các chức năng và định nghĩa dịch vụ. Có hai cách khác nhau để tiến hành trong giai đoạn này, đó là Red-path- solid và Blue-path-dashed. Với Red- path-solid, chúng ta sẽ xây dựng một dịch vụ Web mới từ trạng thái ban đầu hoặc với một dịch vụ đã có sẵn. Từ đó, xây dựng định nghĩa service (WSDL) với các đối tượng, hàm chức năng mà chúng ta mong muốn. Nếu theo cách Blue-path-dashed, dịch vụ Web sẽ được xây dựng từ đầu hoặc từ một định nghĩa dịch vụ WSDL. Sử dụng WSDL này, xây dựng hoặc sửa đổi lại mã để thực hiện các yêu cầu mong muốn trong dịch vụ Web.
- **Giai đoạn triển khai:** công bố định nghĩa dịch vụ, xây dựng WSDL và triển khai mã thực thi của dịch vụ Web. Triển khai dịch vụ Web tới một ứng dụng phía server, sau đó sẽ công bố dịch vụ Web trên mạng Internet để các client có thể nhìn thấy. Sử dụng UDDI registry để công bố lên mạng.
- **Giai đoạn tiến hành:** tìm kiếm và gọi thực thi dịch vụ Web bởi những người dùng muốn sử dụng dịch vụ.
- **Quản lý:** Quản lý và quản trị dịch vụ, duy trì sự ổn định của dịch vụ, cập nhật thông tin mới, sửa lỗi khi nó xảy ra...

Để xây dựng một dịch vụ Web, chúng ta cần hiểu được những việc phải làm và nên bắt đầu từ đâu. Có 3 cách tiếp cận chủ yếu để xây dựng nên một dịch vụ Web, có thể từ một ứng dụng đã có (bottom-up); từ một định nghĩa dịch vụ, WSDL để phát sinh một ứng dụng mới (top-down) hoặc có thể từ một nhóm các dịch vụ Web hiện có, kết hợp lại với nhau để tạo nên các chức năng mới hoặc mở rộng thêm chức năng. Những hướng tiếp cận này dựa trên những gì mà chúng ta đã có, tùy thuộc vào yêu cầu của hệ thống, trong đó tối đa việc sử dụng lại các chức năng, các thành phần, môđun đã được xây dựng.

### ***11.2. Quy trình xây dựng một dịch vụ Web***

Quy trình xây dựng một dịch vụ Web bao gồm các bước sau:

1. Định nghĩa và xây dựng các chức năng, các dịch vụ mà dịch vụ sẽ cung cấp (sử dụng ngôn ngữ Java chẳng hạn).
2. Tạo WSDL cho dịch vụ
3. Xây dựng SOAP server
4. Đăng ký WSDL với UDDI registry để cho phép các client có thể tìm thấy và truy xuất.
5. Client nhận file WSDL và từ đó xây dựng SOAP client để có thể kết nối với SOAP server
6. Xây dựng ứng dụng phía client (chẳng hạn sử dụng Java) và sau đó gọi thực hiện dịch vụ thông qua việc kết nối tới SOAP server.

Lựa chọn một ngôn ngữ, xây dựng các tiến trình nghiệp vụ và chúng ta bắt đầu tạo nên một dịch vụ Web như ý muốn. Sau đó là cung cấp dịch vụ Web này trên Internet.

### ***11.3. Tích hợp Web service theo chuẩn***

Để có thể thành công với dịch vụ Web chúng ta phải quan tâm đến khá nhiều vấn đề, bao gồm việc triển khai, giám sát và tích hợp hệ thống. Doanh nghiệp không những phải phát triển một ứng dụng dịch vụ Web mới mà còn phải tích hợp các ứng dụng nghiệp vụ phụ trợ của họ trong kiến trúc Dịch vụ Web. Cùng với việc triển khai và tích hợp, những nhà kinh doanh và những người sử dụng kỹ thuật cũng cần có khả năng giám sát, triển khai toàn diện để đảm bảo hoạt động kinh doanh hiệu quả và tin cậy.

- **Giám sát (monitoring):** Cần hỗ trợ ở cả mức công cụ và cơ sở hạ tầng để giám sát các dịch vụ Web chạy như thế nào qua toàn bộ mạng, từ một chi nhánh con của một công ty trên mạng tới các chi nhánh khác trong công ty hay giao tiếp với doanh nghiệp khác. Kết hợp thông báo theo sự kiện với các lỗi trong luồng nghiệp vụ cho những người dùng không có kinh nghiệm giám sát dịch vụ Web và các dịch vụ kế thừa khác.

- **Xác định đường đi dữ liệu (Data routing):** Việc thiết lập đường đi của dữ liệu giữa những thành phần của dịch vụ Web hướng tới tối đa hóa khả năng sử dụng lại. Nếu coi một thành phần (component) là một đối tượng thì mỗi thể hiện (instance) của nó sẽ không quan tâm đến các thể hiện khác của cùng thành phần đó. Những thể hiện của cùng

một thành phần có thể dễ dàng được sử dụng lại trong các ứng dụng phân tán khác bởi vì chúng hoàn toàn độc lập và không phụ thuộc lẫn nhau.

- **Triển khai (Deployment):** Triển khai các dịch vụ Web có khả năng nâng cấp, điều khiển và cấu hình các thành phần từ xa thông qua mạng phân tán.
- **Quản lý (Management):** Có thể xây dựng theo kiến trúc P2P (Peer-to-Peer). Các hoạt động chính như thực thi các thành phần, định tuyến dữ liệu, xử lý luồng công việc và chuyển đổi dữ liệu được thực hiện tại các điểm cuối của mạng. Server sẽ tập trung giải quyết các hoạt động khác như quản lý, điều khiển sự kiện, chứng thực bảo mật và quản trị.
- **Cấu hình và quản lý phiên bản (Configuration and version management):** Sử dụng các công cụ linh hoạt để quản lý các phiên bản khác nhau của dịch vụ Web, cho phép các phiên bản được nâng cấp và điều khiển từ một công cụ quản lý tập trung. Kết hợp giữa ứng dụng và mạng giúp các kỹ sư triển khai có thể điều khiển các thành phần chạy trên nền tảng hệ thống phần cứng cụ thể bên trong mạng.
- **Bảo mật (Security):** các chuẩn mở như HTTP, XML, SOAP, WSDL và chuẩn bảo mật JSM được sử dụng rộng rãi khiến chúng trở thành lý tưởng để xây dựng các ứng dụng web. Đầu tiên, dịch vụ Web sử dụng những công nghệ này giống như firewall, SSL và các chứng nhận số. Dịch vụ Web thế hệ sau này sẽ kết hợp với những công nghệ có khả năng bảo mật cao hơn, giống như mã hóa XML và chứng nhận số XML.

➤ Như vậy, với một dịch vụ Web, việc giao tiếp và truyền nhận dữ liệu trở nên dễ dàng và hiệu quả hơn, đồng thời đem lại chi phí thấp hơn và tăng cường những khả năng giao tiếp thời gian thực, kết nối với mọi người trên khắp thế giới. Bản chất của nền tảng công nghệ này là kiến trúc hướng dịch vụ và sự phát triển của dịch vụ Web có tương lai rất khả quan.

## 12. Giới thiệu về Web API

Web API là một sự phát triển từ Web Service khi nền tảng của nó sử dụng nền tảng giao tiếp REST. REST API không yêu cầu các phương thức XML Web service cơ bản (WSDL, SOAP) để hỗ trợ giao diện của mình.

### 12.1. Phương thức thiết kế tự động

Công cụ tự động có thể giúp đỡ ta trong việc tạo dựng Web Service. Đối với các Service sử dụng WSDL, nó có thể tự động tạo ra WSDL cho các Class đã có sẵn (mô

hình từ dưới lên) hoặc tạo khung cho các Class cho WSDL hiện có (mô hình từ trên xuống).

Trong một số ngôn ngữ lập trình, nhà phát triển có thể sử dụng mô hình từ dưới lên viết các Class trước, sau đó sử dụng công cụ tạo WSDL để tạo thành các phương thức từ các Class đó như một Web Service. Đây là cách đơn giản để phát triển nhưng nó sẽ khó để duy trì khi các Class thường xuyên bị thay đổi.

Nhà phát triển sử dụng mô hình từ trên xuống để viết WSDL trước và sau đó sử dụng công cụ tạo code để làm khung cho Class, sẽ được hoàn thành khi cần thiết. Mô hình này thường được coi là khó hơn nhưng nó chắc chắn và bền hơn để thay đổi. Miễn là các định dạng tin nhắn giữa người gửi và người nhận không thay đổi hoặc những thay đổi giữa người gửi và người nhận không ảnh hưởng đến Web Service. Kỹ thuật này được gọi là "Contract First" từ điểm khởi đầu của WSDL.

Nhà phát triển bằng cách sử dụng Subset WSDL có thể kiểm thử Web Service và phát triển từ trên xuống.

### ***12.2. Web Service sử dụng ngôn ngữ đánh dấu***

Đây là một số Web Service sử dụng ngôn ngữ đánh dấu:

- JSON-RPC
- JSON-WSP
- Web template
- Web Services Description Language (WSDL) from the W3C
- XML Interface for Network Services (XINS) provides a POX-style Web service specification format
- Web Services Conversation Language (WSCL)
- Web Services Flow Language (WSFL) (superseded by BPEL)
- WS-MetadataExchange
- Representational state transfer (REST) versus remote procedure call (RPC)
- XML-RPC - XML - Remote Procedure Call

### ***12.3. Khuyết điểm***

Khuyết điểm của non-RESTful Web services là nó quá phức tạp và dựa vào các nhà cung cấp phần mềm lớn chứ không sử dụng các loại mã nguồn mở điển hình.

Cũng có những rủi ro về hiệu suất khi sử dụng XML Web Service như là một định dạng tin nhắn và SOAP/HTML trong đóng gói và vận chuyển.

#### ***12.4. Kiểm thử hồi quy trong Web Service***

Kiểm thử chức năng và phi chức năng của Web Service đã được hoàn thành nhờ sự trợ giúp của WSDL. Kiểm thử hồi quy được thực hiện bằng cách xác định những thay đổi được thực hiện sau đó. Kiểm thử hồi quy của Web Service có thể được phân loại theo 3 cách khác nhau, cụ thể là: Thay đổi trong WSDL, Thay đổi trong code, và Chọn lọc kiểm tra lại những hoạt động của Web Service. Chúng ta có thể nắm bắt được 3 nhu cầu trên trong 3 hình thức trong gian của tập hợp con WSDL tương ứng: Sự khác biệt WSDL, Đơn vị WSDL và giảm tải WSDL. 3 Tập hợp con WSDL sau đó được kết hợp để tạo Combined WSDL và tiếp tục được sử dụng để kiểm thử hồi quy Web Service. Điều này sẽ giúp cho việc Automated Web Service Change Management bằng cách thực hiện việc lựa chọn các trường hợp kiểm thử có liên quan để xây dựng một bộ kiểm thử giảm tải từ bộ kiểm tra cũ.

#### ***12.5. Web Service change management***

Công việc liên quan tới các hoạt động của việc sự giả lập và nắm bắt những thay đổi của Web Service. Giả lập và tính toán những thay đổi có thể thực hiện dưới hình thức hiện vật trung gian (tập hợp con WSDL). Những tính toán có chiều sâu về tác động của những thay đổi rất có ích cho việc kiểm thử, phát triển từ trên xuống và giảm tải kiểm thử hồi quy. Automated Web Service Change Management là một công cụ xác định các hoạt động tập hợp trong một tập tin WSDL để xây dựng tập hợp con WSDL.

### **13. JSON**

JSON: Là viết tắt của JavaScript Object Notation.

JSON là ngôn ngữ dùng để lưu trữ và trao đổi thông tin văn bản giống với XML tuy nhiên JSON nhỏ hơn so với XML, nhanh hơn và dễ phân tích hơn XML.

*Ví dụ:*

```
{
  "employees": [
    { "firstName": "John", "lastName": "Doe" },
    { "firstName": "Anna", "lastName": "Smith" },
    { "firstName": "Peter", "lastName": "Jones" }
  ]
}
```

JSON sử dụng ngôn ngữ JavaScript để mô tả đối tượng dữ liệu, JSON vẫn là ngôn ngữ và nền tảng độc lập so với Java.

	Giống nhau	Khác Nhau
JSON giống và khác với XML	<ul style="list-style-type: none"> <li>- JSON là văn bản trơn (không có định dạng (màu sắc, cỡ chữ,...)).</li> <li>- JSON là "tự mô tả" (người dùng có thể hiểu được).</li> <li>- JSON là phân cấp (có cấu trúc cây).</li> <li>- JSON có thể được phân tích cú pháp (parse) bởi JavaScript.</li> <li>- Dữ liệu JSON có thể được truyền đi bằng AJAX.</li> </ul>	<ul style="list-style-type: none"> <li>- Không có thẻ kết thúc.</li> <li>- Ngắn hơn.</li> <li>- Nhanh hơn để đọc và ghi.</li> <li>- Có thể được phân tích cú pháp bằng hàm dựng sẵn trong JavaScript là eval ().</li> <li>- Sử dụng mảng (Array).</li> <li>- Không dùng các từ reserve.</li> </ul>

#### **Người ta thường dùng XML khi:**

- Lấy một tài liệu XML.
- Sử dụng XML DOM để lặp qua tài liệu.
- Trích xuất các giá trị lưu trữ trong các biến.

#### **Sử dụng JSON khi:**

- Lấy một chuỗi JSON.
- Dùng hàm để chuyển đổi chuỗi JSON thành đối tượng JavaScript.

#### **Các quy luật của cú pháp JSON:**

- Dữ liệu là các cặp tên/giá trị (name/values).

```
"firstName" : "John"
```

- Dữ liệu được phân cách bằng dấu phẩy.
- Đối tượng nằm trong cặp ngoặc nhọn '{} '.

```
{ "firstName": "John" , "lastName": "Doe" }
```

- Dấu ngoặc vuông giữ mảng '[] '.



```
{
    "employees": [
        { "firstName": "John", "lastName": "Doe" },
        { "firstName": "Anna", "lastName": "Smith" },
        { "firstName": "Peter", "lastName": "Jones" }
    ]
}
```

### Ứng dụng demo kết nối giữa Android và web service viết bằng PHP dùng JSON:

Ứng dụng này sẽ dùng phương thức POST để kết nối tới web service, dùng JSON để truyền dữ liệu từ Android lên web service cũng như bắt lấy giá trị JSON từ web service trả về để hiển thị lên ứng dụng cho người dùng thấy.

Ta có 4 file php dùng làm web service tương ứng với 4 trường hợp khác nhau với hàm `json_encode` dùng để chuyển định dạng array sang định dạng JSON:

- File `vidu1.php`: Trường hợp không có kiểu trả về, không có tham số.

```
<?php
function vidu1()
{
    $a = array('Loc' => 'Hello world');
    echo json_encode($a);
}
vidu1();
?>
```

Hình 13. Ví dụ PHP - Không có kiểu trả về, không có tham số

- File `vidu2.php`: Trường hợp không có kiểu trả về, có tham số.

```
<?php
function vidu2($a)
{
    $ketqua = array ('Ten' => $a);
    echo json_encode($ketqua);
}

$b = $_POST["ten"];
vidu2($b);
?>
```

Hình 14. Ví dụ PHP - Không có kiểu trả về, có tham số

- File `vidu3.php`: Trường hợp có kiểu trả về, không có tham số.

```
<?php
function vidu3()
{
    $a = array ('Loc' => 'Hello world !');
    return json_encode($a);
}
echo vidu3();
?>
```

Hình 15. Ví dụ PHP - Có kiểu trả về, không có tham số

- File vidu4.php: Trường hợp có kiểu trả về, có tham số.

```
<?php
function vidu4($a,$b)
{
    $ketqua = array ('Ho' => $a, 'Ten' => $b);
    return json_encode($ketqua);
}
echo vidu4($_POST["ho"],$_POST["ten"]);
?>
```

Hình 16. Ví dụ PHP - Có kiểu trả về, có tham số

Trên ứng dụng, ta sẽ có 4 nút tương ứng với 4 trường hợp trên, để kết nối ứng dụng với web service ta dùng HttpPost và quá trình kết nối trao đổi dữ liệu sẽ được diễn ra thông qua AsyncTask. Ta có kết quả ứng dụng như sau:

21:16

1. Không có kiểu trả về, có tham số

KẾT QUẢ 1

2. Không có kiểu trả về, có tham số

Hai

KẾT QUẢ 2

3. Có kiểu trả về, không có tham số

KẾT QUẢ 3

4. Có kiểu trả về, có tham số

Ho

Hoang Hai

KẾT QUẢ 4

Ho Hoang Hai

Hình 17. Giao diện ứng dụng Demo

## CHƯƠNG II. TỔNG QUAN VỀ THƯƠNG MẠI ĐIỆN TỬ

### 1. Khái niệm thương mại điện tử

#### 1.1. *Thương mại điện tử theo nghĩa hẹp:*

- **Thương mại điện tử** là việc mua bán hàng hóa và dịch vụ thông qua các phương tiện điện tử, nhất là Internet và các mạng viễn thông khác. (Quan điểm vào cuối thập kỷ 90s).
- **Thương mại điện tử** là các giao dịch thương mại và hàng hóa và dịch vụ được thực hiện thông qua các phương tiện điện tử (Diễn đàn đối thoại xuyên Đại Tây Dương, 1997).
- **Thương mại điện tử** là việc thực hiện các giao dịch kinh doanh có dẫn tới việc chuyển giao giá trị thông qua các mạng viễn thông (EITO, 1997).
- **Thương mại điện tử** là việc hoàn thành bất kỳ một giao dịch nào thông qua một mạng máy tính làm trung gian mà bao gồm việc chuyển giao quyền sở hữu hay quyền sử dụng hàng hóa và dịch vụ (Cục thống kê Hoa Kỳ, 2000).

#### 1.2. *Thương mại điện tử theo nghĩa rộng:*

- **Thương mại điện tử** là toàn bộ chu trình và các hoạt động kinh doanh liên quan đến các tổ chức hay cá nhân.
- **Thương mại điện tử** là việc tiến hành hoạt động thương mại sử dụng các phương tiện điện tử và công nghệ xử lý thông tin số hóa.
- **Thương mại điện tử** bao gồm việc sản xuất, phân phối, marketing, bán hay giao hàng hóa và dịch vụ bằng các phương tiện điện tử (UNCTAD - United Nations Conference on Trade And Development , 1998).
- **Thương mại điện tử** bao gồm các giao dịch thương mại thông qua các mạng viễn thông và sử dụng các phương tiện điện tử. Bao gồm, TMĐT gián tiếp (trao đổi hàng hóa hữu hình) và TMĐT trực tiếp (Trao đổi hàng hóa vô hình).
- **Thương mại điện tử** cũng được hiểu là hoạt động kinh doanh điện tử, bao gồm: mua bán điện tử hàng hóa, dịch vụ, giao hàng trực tiếp trên mạng với các nội dung số hóa; chuyển tiền điện tử - EFT (electronic fund transfer); mua bán cổ phần điện tử - EST (Electronic bill of lading); đấu giá thương mại - Commercial auction; hợp tác thiết kế và

sản xuất; tìm kiếm các nguồn lực trực tuyến; mua sắm trực tuyến - Online procurement; marketing trực tiếp, dịch vụ khách hàng sau khi bán.

- **Thương mại điện tử** là việc làm kinh doanh thông qua mạng Internet, bán những hàng hoá và dịch vụ có thể được phân phối không thông qua mạng hoặc những hàng hoá có thể mã hoá bằng kỹ thuật số và được phân phối thông qua mạng hoặc không thông qua mạng.

- **Thương mại điện tử** gồm các giao dịch thương mại liên quan đến các tổ chức và cá nhân dựa trên việc xử lý và truyền đi các dữ kiện đã được số hoá thông qua các mạng mở (như Internet) hoặc các mạng đóng có cổng thông với mạng mở (như AOL). (OECD: Organization for Economic Cooperation and Development)

- **Thương mại điện tử** là việc thực hiện toàn bộ hoạt động kinh doanh bao gồm marketing, bán hàng, phân phối và thanh toán (MSDP) thông qua các phương tiện điện tử (UN: United Nations).

- **Thương mại điện tử** bao gồm việc sản xuất, quảng cáo, bán hàng và phân phối sản phẩm được mua bán và thanh toán trên mạng Internet, nhưng được giao nhận có thể hữu hình hoặc giao nhận qua Internet dưới dạng số hóa (WTO: World Trade Organization).

- **Thương mại điện tử** là làm kinh doanh có sử dụng các công cụ điện tử, định nghĩa này rộng, coi hầu hết các hoạt động kinh doanh từ đơn giản như một cú điện thoại giao dịch đến những trao đổi thông tin EDI phức tạp đều là TMĐT (AEC: Association for Electronic Commerce).

- **Thương mại điện tử** là việc trao đổi thông tin thương mại thông qua các phương tiện điện tử, không cần phải in ra giấy bất cứ công đoạn nào của toàn bộ quá trình giao dịch. (UNCITRAL: United Nations Conference for International Trade Law: Luật mẫu về TMĐT)

## **2. Bản chất, đặc trưng, lợi ích và hạn chế của thương mại điện tử**

### **2.1. Bản chất của thương mại điện tử**

- Thương mại điện tử gồm toàn bộ các chu trình và các hoạt động kinh doanh của các tổ chức và cá nhân được thực hiện thông qua các phương tiện điện tử.

- Thương mại điện tử phải được xây dựng trên một nền tảng vững chắc về cơ sở hạ tầng (bao gồm cơ sở hạ tầng về kinh tế, công nghệ, pháp lý và nguồn nhân lực).

## ***2.2. Đặc trưng của thương mại điện tử***

- Đối với thương mại truyền thống thì mạng lưới thông tin chỉ là phương tiện để trao đổi dữ liệu, còn đối với thương mại điện tử thì mạng lưới thông tin chính là thị trường.
- Các bên tiến hành trong giao dịch trong thương mại điện tử không tiếp xúc trực tiếp với nhau và không đòi hỏi phải biết nhau từ trước.
- Thương mại điện tử được thực hiện trong một thị trường không có biên giới (thống nhất toàn cầu) và nó tác động tới môi trường cạnh tranh toàn cầu.
- Trong hoạt động giao dịch thương mại điện tử đều có sự tham gia của ít nhất ba chủ thể, trong đó có một bên không thể thiếu được là người cung cấp dịch vụ mạng, các cơ quan chứng thực.
- Tóm lại, trong thương mại điện tử bản chất của thông tin không thay đổi. Thương mại điện tử chỉ biến đổi cách thức khởi tạo, trao đổi, bảo quản và xử lý thông tin, hoàn toàn không thay đổi những chức năng cơ bản của thông tin đối với các bên tham gia truyền thống của hợp đồng.

## ***2.3. Lợi ích và hạn chế của thương mại điện tử***

### **❖ Lợi ích:**

- **Quảng bá thông tin và tiếp thị trong thị trường toàn cầu với chi phí thấp:** bất cứ doanh nghiệp nào cũng có thể thiết lập website một cách dễ dàng bằng một số tiền tương đối nhỏ (khoảng 480.000đ cho việc thuê tên miền/1 năm, và khoảng >1.000.000đ cho việc thuê không gian máy chủ/1 năm). Nếu doanh nghiệp không thể tự thiết kế website thì có thể nhờ các công ty, dịch vụ thiết kế web với giá từ 3 triệu - 5 triệu đồng tùy theo số trang và chức năng của website. Như vậy, với khoảng < 5 triệu đồng cho 1 năm, doanh nghiệp sẽ có một website thương mại xuất hiện trên mạng Internet, nơi mà mọi người trên thế giới đều có thể truy cập và đọc được các thông tin trên website của họ. Tuy nhiên, doanh nghiệp cần phải đầu tư công sức cho việc quảng bá, giới thiệu, đăng ký website với các công cụ tìm kiếm để khách hàng dễ dàng tìm thấy website của doanh nghiệp trong hàng tỷ trang web hiện có.
- **Cung cấp dịch vụ tốt hơn cho khách hàng:** tạo cơ hội tiếp xúc trực tiếp với khách hàng, ghi nhận phản ánh, thắc mắc của khách hàng một cách nhanh chóng, tiện lợi, từ đó nắm bắt tốt hơn nhu cầu và nguyện vọng của họ, giúp cải tiến chất lượng sản phẩm, dịch vụ... làm tăng sự hài lòng của khách hàng.

- **Tăng doanh thu và giảm chi phí:** Thương mại điện tử giúp doanh nghiệp tiếp cận được với thị trường thế giới, tăng lượng khách hàng. Hơn nữa, việc tự động hóa tiến trình kinh doanh, giúp tăng hiệu quả hoạt động, giảm thời gian xử lý đơn hàng, đáp ứng kịp thời nhu cầu khách hàng, cải tiến dây chuyền cung ứng... giúp tăng doanh thu và giảm chi phí hoạt động.

- **Tạo lợi thế cạnh tranh:** trong thời đại công nghệ và Internet, các doanh nghiệp nắm bắt kịp thời sự phát triển của công nghệ và ứng dụng vào hoạt động kinh doanh sẽ tạo được lợi thế cạnh tranh trước các đối thủ. Ta nhận thấy, khi một doanh nghiệp áp dụng thương mại điện tử thì các doanh nghiệp khác cũng áp dụng, vì vậy, doanh nghiệp phải tạo được sự khác biệt cho mình dựa vào công nghệ mới, thể hiện ở sự tiện lợi, nhanh chóng, mỹ thuật và hiệu quả của website thương mại điện tử và phương thức kinh doanh.

Chúng ta có thể phân loại lợi ích của thương mại điện tử với các nhóm đối tượng:

#### ***2.3.1. Lợi ích đối với các tổ chức:***

- **Mở rộng thị trường:** Với chi phí đầu tư nhỏ hơn nhiều so với thương mại truyền thống, các công ty có thể mở rộng thị trường, tìm kiếm, tiếp cận người cung cấp, khách hàng và đối tác trên khắp thế giới. Việc mở rộng mạng lưới nhà cung cấp, khách hàng cũng cho phép các tổ chức có thể mua với giá thấp hơn và bán được nhiều sản phẩm hơn.
- **Giảm chi phí sản xuất:** Giảm chi phí giấy tờ, giảm chi phí chia sẻ thông tin, chi phí in ấn, gửi văn bản truyền thống.
- **Cải thiện hệ thống phân phối:** Giảm lượng hàng lưu kho và độ trễ trong phân phối hàng. Hệ thống cửa hàng giới thiệu sản phẩm được thay thế hoặc hỗ trợ bởi các showroom trên mạng.
- **Vượt giới hạn về thời gian:** Việc tự động hóa các giao dịch thông qua Web và Internet giúp hoạt động kinh doanh được thực hiện 24/7/365 mà không mất thêm nhiều chi phí.
- **Sản xuất hàng theo yêu cầu:** Còn được biết đến dưới tên gọi “Chiến lược kéo”, lôi kéo khách hàng đến với doanh nghiệp bằng khả năng đáp ứng mọi nhu cầu của khách hàng.
- **Mô hình kinh doanh mới:** Các mô hình kinh doanh mới với những lợi thế và giá

trị mới cho khách hàng. Một ví dụ điển hình là mô hình của Amazon.com, mua hàng theo nhóm hay đấu giá nông sản qua mạng đến các sàn giao dịch B2B là điển hình của những thành công này.

- **Tăng tốc độ tung sản phẩm ra thị trường:** Với lợi thế về thông tin và khả năng phối hợp giữa các doanh nghiệp làm tăng hiệu quả sản xuất và giảm thời gian tung sản phẩm ra thị trường.
- **Giảm chi phí thông tin liên lạc**
- **Giảm chi phí mua sắm:** Thông qua giảm các chi phí quản lý hành chính (80%); giảm giá mua hàng (5-15%) (Nguồn: [www.vnn.vn](http://www.vnn.vn))
- **Củng cố quan hệ khách hàng:** Thông qua việc giao tiếp thuận tiện qua mạng, quan hệ với trung gian và khách hàng được củng cố dễ dàng hơn. Đồng thời việc cá biệt hóa sản phẩm và dịch vụ cũng góp phần thắt chặt quan hệ với khách hàng và củng cố lòng trung thành.
- **Thông tin cập nhật:** Mọi thông tin trên web như sản phẩm, dịch vụ, giá cả... đều có thể được cập nhật nhanh chóng và kịp thời.
- **Chi phí đăng ký kinh doanh:** Một số nước và khu vực khuyến khích bằng cách giảm hoặc không thu phí đăng ký kinh doanh qua mạng. Thực tế, việc thu nếu triển khai cũng gặp rất nhiều khó khăn do đặc thù của Internet.
- **Các lợi ích khác:** Nâng cao uy tín, hình ảnh doanh nghiệp; cải thiện chất lượng dịch vụ khách hàng; đối tác kinh doanh mới; đơn giản hóa và chuẩn hóa các quy trình giao dịch; tăng năng suất, giảm chi phí giấy tờ; tăng khả năng tiếp cận thông tin và giảm chi phí vận chuyển; tăng sự linh hoạt trong giao dịch và hoạt động kinh doanh.

### ***2.3.2. Lợi ích đối với người tiêu dùng***

- **Vượt giới hạn về không gian và thời gian:** Thương mại điện tử cho phép khách hàng mua sắm mọi nơi, mọi lúc đối với các cửa hàng trên khắp thế giới.
- **Nhiều lựa chọn về sản phẩm và dịch vụ:** Thương mại điện tử cho phép người mua có nhiều lựa chọn hơn vì tiếp cận được nhiều nhà cung cấp hơn.
- **Giá thấp hơn:** Do thông tin thuận tiện, dễ dàng và phong phú hơn nên khách hàng có thể so sánh giá cả giữa các nhà cung cấp thuận tiện hơn và từ đó tìm được mức giá phù hợp nhất.



- **Giao hàng nhanh hơn với các hàng hóa số hóa được:** Đối với các sản phẩm số hóa được như phim, nhạc, sách, phần mềm.... việc giao hàng được thực hiện dễ dàng thông qua Internet.
- **Thông tin phong phú, thuận tiện và chất lượng cao hơn:** Khách hàng có thể dễ dàng tìm được thông tin nhanh chóng và dễ dàng thông qua các công cụ tìm kiếm (search engines); đồng thời các thông tin đa phương tiện (âm thanh, hình ảnh).
- **Đấu giá:** Mô hình đấu giá trực tuyến ra đời cho phép mọi người đều có thể tham gia mua và bán trên các sàn đấu giá và đồng thời có thể tìm, sưu tầm những món hàng mình quan tâm tại mọi nơi trên thế giới.
- **Cộng đồng thương mại điện tử:** Môi trường kinh doanh thương mại điện tử cho phép mọi người tham gia có thể phối hợp, chia sẻ thông tin và kinh nghiệm hiệu quả và nhanh chóng.
- **“Đáp ứng mọi nhu cầu”:** Khả năng tự động hóa cho phép chấp nhận các đơn hàng khác nhau từ mọi khách hàng.
- **Thuế:** Trong giai đoạn đầu của thương mại điện tử, nhiều nước khuyến khích bằng cách miễn thuế đối với các giao dịch trên mạng.

### 2.3.3. Lợi ích đối với xã hội

- **Hoạt động trực tuyến:** Thương mại điện tử tạo ra môi trường để làm việc, mua sắm, giao dịch... từ xa nên giảm việc đi lại, ô nhiễm, tai nạn.
- **Nâng cao mức sống:** Nhiều hàng hóa, nhiều nhà cung cấp tạo áp lực giảm giá do đó khả năng mua sắm của khách hàng cao hơn, nâng cao mức sống của mọi người.
- **Lợi ích cho các nước nghèo:** Những nước nghèo có thể tiếp cận với các sản phẩm, dịch vụ từ các nước phát triển hơn thông qua Internet và thương mại điện tử. Đồng thời cũng có thể học tập được kinh nghiệm, kỹ năng... được đào tạo qua mạng.
- **Dịch vụ công được cung cấp thuận tiện hơn:** Các dịch vụ công cộng như y tế, giáo dục, các dịch vụ công của chính phủ... được thực hiện qua mạng với chi phí thấp hơn, thuận tiện hơn. Cấp các loại giấy phép qua mạng, tư vấn y tế.... là các ví dụ thành công điển hình.

#### 2.3.4. Hạn chế

- **Sự thay đổi nhanh chóng của công nghệ:** ngày nay, các công nghệ phần cứng và phần mềm thay đổi rất nhanh chóng, nếu doanh nghiệp không nắm bắt và ứng dụng kịp các công nghệ mới sẽ có thể bị tụt hậu và đánh mất lợi thế cạnh tranh. Vì vậy, các doanh nghiệp phải có bộ phận chuyên trách về quản lý và theo dõi hoạt động website thương mại điện tử, nắm bắt tình hình công nghệ và cập nhật, nâng cấp website thường xuyên cho phù hợp với tình hình phát triển chung.
- **Vấn đề bảo mật thông tin và an toàn cơ sở dữ liệu:** khi kinh doanh trên mạng, doanh nghiệp phải quan tâm nhiều đến vấn đề an ninh mạng, chẳng hạn như: sự lây lan của virus tin học, sâu Internet, hiện tượng hacker tấn công vào website... sẽ gây tê liệt và ngưng hoạt động của hệ thống trong 1 thời gian, cho đến làm sai lệch dữ liệu, xóa cơ sở dữ liệu, làm hỏng máy chủ web không thể khắc phục được... đều gây thiệt hại cho các doanh nghiệp kinh doanh thương mại điện tử, nhất là các doanh nghiệp trong lĩnh vực tài chính, ngân hàng. Vấn đề này đòi hỏi các doanh nghiệp phải chọn nhà cung cấp dịch vụ hosting an toàn, có giải pháp bảo mật tốt, thường xuyên sao lưu cơ sở dữ liệu và phối hợp với nhà cung cấp để khắc phục sự cố kịp thời.
- **Rủi ro trong thanh toán qua mạng:** các website thương mại điện tử cũng có thể gặp phải rắc rối với vấn đề thanh toán qua mạng. Chẳng hạn, khi doanh nghiệp bán hàng cho 1 khách hàng với 1 số tài khoản hợp lệ nào đó. Nếu như đây là 1 tài khoản bị đánh cắp, thì doanh nghiệp có nguy cơ mất trắng số tiền này khi phát sinh kiện tụng.
- **Thiếu cơ chế pháp lý hoàn thiện để giải quyết tranh chấp qua mạng:** Ở nước ta, luật thương mại điện tử đang trong quá trình xây dựng và hoàn thiện, nên sẽ khó khăn khi xử lý các vấn đề tranh chấp xảy ra trong thực tế nhưng chưa có trong luật. Hơn nữa, các hoạt động lừa đảo trên mạng thì không ít và ngày càng tinh vi hơn, vì vậy, ở giai đoạn đầu của kinh doanh thương mại điện tử, nếu có phát sinh các tranh chấp, kiện tụng,... thì phần thiệt thòi thường thuộc về phía các doanh nghiệp. Do đó, các doanh nghiệp phải tự bảo vệ mình bằng các biện pháp kinh doanh kết hợp với công nghệ.

<b>Hạn chế về mặt kỹ thuật</b>	<b>Hạn chế về mặt thương mại</b>
1. Chưa có tiêu chuẩn quốc tế về chất lượng, an toàn và độ tin cậy.	1. An ninh và riêng tư là hai cản trở về tâm lý đối với người tham gia thương mại điện tử.
2. Tốc độ đường truyền Internet vẫn chưa đáp ứng được yêu cầu của người dùng, nhất là trong thương mại điện tử.	2. Thiếu lòng tin và thương mại điện tử và người bán hàng trong thương mại điện tử do không được gặp trực tiếp.
3. Các công cụ xây dựng phần mềm vẫn trong giai đoạn đang phát triển.	3. Nhiều vấn đề về luật, chính sách, thuế chưa được làm rõ.
4. Khó khăn khi kết hợp các phần mềm thương mại điện tử với các phần mềm ứng dụng và các cơ sở dữ liệu truyền thống.	4. Một số chính sách chưa thực sự hỗ trợ tạo điều kiện để thương mại điện tử phát triển.
5. Cần có các máy chủ thương mại điện tử đặc biệt (công suất, an toàn) đòi hỏi thêm chi phí đầu tư.	5. Các phương pháp đánh giá hiệu quả của thương mại điện tử còn chưa đầy đủ, hoàn thiện.
6. Chi phí truy cập Internet vẫn còn cao.	6. Chuyển đổi thói quen tiêu dùng từ thực đến ảo cần thời gian.
7. Thực hiện các đơn đặt hàng trong TMĐT B2C đòi hỏi hệ thống kho hàng tự động lớn.	7. Sự tin cậy đối với môi trường kinh doanh không giấy tờ, không tiếp xúc trực tiếp, giao dịch điện tử cần thời gian.
8. Số lượng người tham gia chưa đủ lớn để đạt lợi thế về quy mô 9. Số lượng gian lận ngày càng tăng do đặc thù của thương mại điện tử. 10. Thu hút vốn đầu tư mạo hiểm khó khăn hơn sau sự sụp đổ.	

Theo nghiên cứu của CommerceNet (commerce.net), 10 cản trở lớn nhất của thương mại điện tử theo thứ tự là:

1. An toàn
2. Sự tin tưởng và rủi ro
3. Thiếu nhân lực về thương mại điện tử

#### 4. Văn hóa

5. Thiếu hạ tầng về chữ ký số hóa (hoạt động của các tổ chức chứng thực còn hạn chế)
6. Nhận thức của các tổ chức về thương mại điện tử
7. Gian lận trong thương mại điện tử (thẻ tín dụng...)
8. Các sàn giao dịch B2B chưa thực sự thân thiện với người dùng
9. Các rào cản thương mại quốc tế truyền thống
10. Thiếu các tiêu chuẩn quốc tế về thương mại điện tử

Những hạn chế trên được sắp xếp theo thứ tự giảm dần độ nghiêm trọng. Những hạn chế này kìm hãm sự phát triển của thương mại điện tử, và cũng làm giảm đi lợi ích mà thương mại điện tử mang lại. Do vậy, cần có biện pháp làm giảm hay triệt tiêu những hạn chế này.

### **3. Nền tảng của thương mại điện tử**

#### ***3.1. Mạng viễn thông và Internet***

##### ***3.1.1. Mạng LANs (Local Area Networks)***

- Các mạng cục bộ (Local area networks-LANs) nối kết các máy tính và các thiết bị xử lý thông tin khác trong một khu vực hạn chế, như văn phòng, lớp học, tòa nhà, nhà máy sản xuất, hay nơi làm việc khác. LANs đã trở nên thông dụng trong nhiều tổ chức đối với việc cung cấp các khả năng mạng viễn thông nối kết nhiều người dùng trong các văn phòng, bộ phận, và các nhóm làm việc khác.

- LANs sử dụng nhiều phương tiện viễn thông, như dây điện thoại thông thường, cáp đồng trục, hay thậm chí các hệ thống vô tuyến để nối các trạm máy vi tính và các thiết bị ngoại vi với nhau. Để giao tiếp qua mạng, mỗi PC thường có một bo mạch được gọi là một card giao tiếp mạng. Phần lớn LANs dùng một máy vi tính mạnh hơn có dung lượng đĩa cứng lớn, được gọi là file server hay máy chủ mạng, chứa chương trình hệ điều hành mạng kiểm soát viễn thông, cách dùng và chia sẻ tài nguyên mạng. Ví dụ, nó phân phối các bản sao các tập tin dữ liệu thông thường và các bộ phần mềm đến các máy vi tính khác trong mạng và kiểm soát truy cập đến các máy in và các thiết bị ngoại vi khác đã được chia sẻ.

- Topo (hay topology) mạng là sơ đồ vật lý của một mạng cục bộ.

- Mạng LAN đầu tiên sử dụng kiểu kênh/tuyến (Bus). Dây cáp mạng tạo thành một kênh duy nhất và các máy trạm nối vào đó. Để giải quyết tranh chấp, mạng bus sử dụng vài loại kỹ thuật quản lý tranh chấp. Kỹ thuật này nhằm loại bỏ dữ liệu bị hỏng do tranh chấp. Mạng có ưu điểm là thiết kế đơn giản nhưng khó kiểm soát và không thể đưa thêm một trạm vào giữa mạng.

- Kiểu hình sao (Star) giải quyết vấn đề mở rộng mạng bằng cách đưa vào một bộ tập trung là Hub, nhờ đó việc bổ sung thêm người sử dụng khá đơn giản nhưng độ tin cậy thấp vì tất cả máy tính phụ thuộc vào máy chủ trung tâm.

- Kiểu vòng (Ring) tạo ra cách duy nhất để chống tình trạng tranh chấp, một trạm làm việc chỉ có thể truyền dữ liệu khi sở hữu một thẻ bài, là đơn vị đặc biệt chạy vòng trong mạng này.

### **3.1.2. Mạng WAN**

Các mạng viễn thông bao phủ một phạm vi địa lý rộng lớn được gọi là **mạng diện rộng** (Wide Area Network-WAN). Các mạng bao phủ một thành phố lớn hay phạm vi thủ đô (Metropolitan Area Network-MAN) cũng có thể được bao gồm trong loại này. Các mạng lớn như vậy đã trở nên cần thiết cho việc thực hiện các hoạt động hàng ngày của nhiều tổ chức kinh doanh và chính phủ và những người dùng cuối của nó. Ví dụ, WAN được dùng bởi nhiều công ty đa quốc gia để chuyển và nhận thông tin giữa các nhân viên, khách hàng, nhà cung cấp, và các tổ chức khác qua nhiều thành phố, vùng, quốc gia và thế giới.

### **3.1.3. Mạng Intranet, Extranet**

- Intranet : Là mạng cục bộ dành cho các nhân viên bên trong tổ chức
  - Mạng riêng gồm nhiều LAN & WAN.
  - Sử dụng các nghi thức để liên lạc như : TCP/IP, IPX/SPX...
  - Thường có Firewalls nếu có kết nối Internet.
- Extranet: Dạng mở rộng của Intranet, cho phép kết nối từ ngoài vào.
  - Một kiểu mạng Intranet mở rộng
  - Dành cho giao tiếp với khách hàng, đại lý bên ngoài.

### **3.1.4. Mạng Internet**

- Internet: kết nối nhiều LAN, tạo khả năng truy cập mở trên toàn cầu.

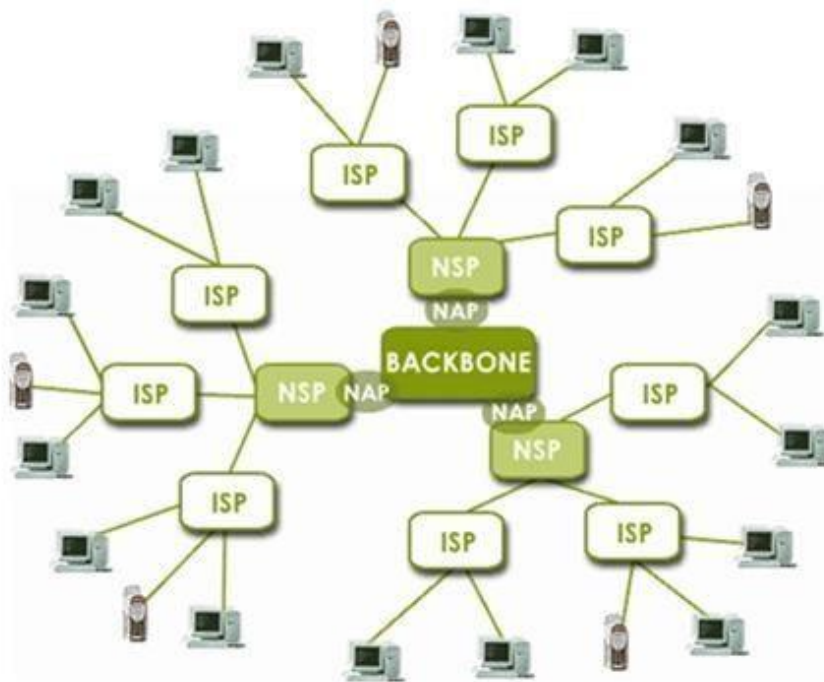
- Mạng cộng đồng diện rộng, sử dụng nghi thức TCP/IP
- Gồm nhiều Intranet kết nối bằng đường điện thoại, vệ tinh...
- Mỗi máy tính trong mạng có một địa chỉ IP duy nhất.

Chỉ cách nay vài năm, các doanh nghiệp gặp rất nhiều khó khăn trong việc chuyển giao thông tin và các ứng dụng của mình cho nhân viên, nhất là nhân viên ở những nơi xa xôi hẻo lánh, hoặc nơi ở của họ bị phân tán khắp nơi. Ngày nay, có thể dễ dàng thực hiện điều đó cho cả khách hàng, đối tác và cộng đồng, bất chấp họ ở đâu trên thế giới. Web chính là chất xúc tác cho sự thay đổi này. Tuy nhiên, hơn ba mươi năm nay nếu không có sự phát triển trong cơ sở hạ tầng mạng toàn cầu, có lẽ Web đã không thể tồn tại!

Trong khi có rất nhiều người sử dụng Internet hàng ngày nhưng chỉ ít người hiểu biết rõ ràng về các thao tác cơ bản của nó. Ở góc độ vật lý, Internet là mạng của hàng ngàn mạng được nối kết lẫn nhau. Chúng là:

- Mạng xương sống cấp độ quốc tế được nối kết với nhau.
- Các mạng con truy cập/chuyên giao thông tin.
- Hàng ngàn mạng riêng và của các tổ chức nối với rất nhiều máy chủ của các tổ chức. Các mạng này chứa nhiều thông tin thú vị.

Các mạng xương sống được kiểm soát bởi các **Nhà cung cấp dịch vụ mạng** (Network Service Providers - **NSPs**) như MCI, Sprint, UUNET/MIS... Mỗi mạng xương sống xử lý hơn 300 terabytes/tháng. Các mạng con đến từ các **Nhà cung cấp dịch vụ Internet** (Internet Service Provider - **ISPs**) trao đổi dữ liệu với **NSP tại các điểm truy cập mạng** (Network Access Points - **NAPs**). Hình sau đây minh họa các kết nối giữa ISP, NAP và các mạng xương sống.



Hình 18: Kiến trúc mạng Internet

Khi người sử dụng gửi một yêu cầu lên Internet từ máy tính của mình, nó sẽ theo mạng ISP, di chuyển qua một hay nhiều mạng xương sống và băng qua mạng ISP khác đến máy tính chứa thông tin quan tâm. Câu trả lời cho yêu cầu đó sẽ theo thứ tự lộ trình tương tự. Bất kỳ yêu cầu và kết quả trả lời nào cũng đều không theo lộ trình định sẵn. Thật vậy, chúng bị tách ra thành các gói và mỗi gói lại theo những lộ trình khác nhau. Những lộ trình này được xác định bởi các máy tính đặc biệt gọi là Router. Các Router có những bản đồ mạng trên Internet có thể cập nhật được cho phép chúng xác định đường đi cho các gói tin.

Một điều làm mọi người ngạc nhiên là "không ai chịu trách nhiệm trên Internet" và chính vì thế mà ở giai đoạn đầu các xí nghiệp rất miễn cưỡng sử dụng Internet cho mục đích kinh doanh của họ.

Vấn đề liên mạng nằm ở chỗ là làm thế nào để xây dựng một tập hợp các giao thức xử lý truyền thông giữa hai máy tính bất kỳ trở lên bằng cách sử dụng bất kỳ hệ điều hành nào và chúng được nối bởi bất kỳ phương tiện vật lý nào. Chúng ta giả định rằng không hệ thống nối kết nào có kiến thức về các hệ thống khác; không có cách nào biết được một hệ thống ở xa đang ở đâu và nó sử dụng loại phần mềm nào và chạy trên nền phần cứng nào.

## **3.2. Các công nghệ hỗ trợ thương mại điện tử**

### **3.2.1. Kiến trúc ứng dụng Client/Server:**

Các ứng dụng trên nền web thường dựa trên kiến trúc 2 lớp là client/ server. Kiến trúc client/server cho phép chia sẻ việc xử lý giữa các máy chủ hay máy trạm khác nhau. Trong đó, người sử dụng sử dụng trình duyệt từ phía máy khách (client), gửi các yêu cầu về thông tin đến máy chủ (server), máy chủ tiếp nhận yêu cầu, xử lý, truy xuất các thông tin cần thiết và gửi kết quả về phía client dưới dạng 1 trang web. Ở mô hình này máy chủ vừa cung cấp các dịch vụ truy xuất web, vừa chứa các dữ liệu cần thiết để đáp ứng các yêu cầu của máy client, điều này khiến cho dữ liệu trên máy chủ không an toàn.

Để đảm bảo an toàn dữ liệu, người ta đưa ra mô hình 3 lớp, trong đó, lớp server sẽ được tách thành web server (máy chủ xử lý ứng dụng web) và database server (máy chủ quản lý thông tin trong CSDL). Lúc này, máy client sẽ gửi các yêu cầu dịch vụ và nhận các kết quả trả về từ Web server (máy chủ cung cấp dịch vụ web). Webserver sẽ tùy theo yêu cầu của phía client mà kết nối đến Database Server (máy chủ cung cấp dữ liệu) để lấy các dữ liệu tương ứng.

Tùy theo các chức năng của ứng dụng web mà người ta có thể chia ra làm nhiều lớp khác nữa, gọi chung là mô hình n lớp.

### **3.2.2. Các ngôn ngữ phát triển ứng dụng web:**

Các ứng dụng web có thể được viết bằng ngôn ngữ HTML (web tĩnh), hoặc kết hợp với các ngôn ngữ lập trình web (gọi là các ngôn ngữ script) để thực hiện các yêu cầu xử lý và truy xuất dữ liệu, để trả về trang web có nội dung thay đổi tùy theo đối tượng và hoàn cảnh (web động).

Các ngôn ngữ script có thể là : CGI, Perl, ASP, VBScript (dựa trên ngôn ngữ Visual Basic), PHP (theo cú pháp ngôn ngữ C++), JSP, JavaScript (dựa trên ngôn ngữ Java)... Các script này có thể được quy định chạy phía máy server hoặc client. Tuy nhiên, để dụng được các script này server phải được cài đặt và cấu hình phù hợp.

Ngoài ra, các công nghệ mới như : Java Bean, Java Applet, Dot Net,... cũng được giới thiệu và sử dụng ngày càng nhiều trong lập trình web để tạo các ứng dụng xử lý ở phía server và trả về trang web cho phía client.



### 3.2.3. Cơ sở dữ liệu và ứng dụng Web:

Ngày nay, các ứng dụng web đều đòi hỏi kết nối với 1 cơ sở dữ liệu nào đó, để lưu trữ các thông tin cập nhật, cũng như các giao dịch tiến hành trên mạng. Việc kết nối CSDL của tổ chức với website TMĐT càng cần thiết hơn khi hoạt động TMĐT đã đạt đến mức độ phát triển cao, đòi hỏi phải tích hợp với các hệ thống thông tin quản lý trong tổ chức, như : hệ thống xử lý đơn hàng, hệ thống kế toán, hệ thống quản lý nhân sự, ...

Hiện nay, các doanh nghiệp thường sử dụng 1 hệ quản trị cơ sở dữ liệu để quản lý toàn bộ dữ liệu hoạt động, kinh doanh của tổ chức. Có rất nhiều hệ quản trị CSDL phổ biến hiện nay, như là: Access, Foxpro, SQL Server, MySQL, SyBase, Oracle, DB2... Các hệ QTCSDL này đều hỗ trợ mô hình CSDL quan hệ, đây là 1 mô hình CSDL phổ biến, được phát triển dựa trên cơ sở toán học là đại số quan hệ.

Các hệ QTCSDL quan hệ này đều có 2 chức năng cơ bản sau :

- **Tổ chức lưu trữ dữ liệu:** dưới dạng 1 bảng, gồm các cột (field) và các dòng (record). Các bảng thường có quan hệ với nhau, trên đó có cài đặt các cơ chế đảm bảo nhất quán và toàn vẹn dữ liệu.
- **Truy vấn dữ liệu:** sử dụng ngôn ngữ SQL là 1 ngôn ngữ theo chuẩn ANSI & ISO để truy vấn dữ liệu 1 cách nhanh chóng và hiệu quả.

Ngoài ra, các HQTCSDL còn có thể có các chức năng sau:

- Quản lý sao lưu và phục hồi dữ liệu.
- Quản lý bảo mật và cấp phát quyền cho người dùng CSDL. + Quản lý nhập, xuất và chuyển đổi dữ liệu.
- Quản lý giao tác và lưu vết cập nhật dữ liệu...

Với 1 lượng dữ liệu lớn trong CSDL vận hành (operational database), các doanh nghiệp có thể tập hợp chúng lại thành 1 kho dữ liệu tổ chức (data warehouse). Từ đó, họ có thể sử dụng các công cụ, như : suy luận tình huống (case bases reasoning), khai mỏ dữ liệu (data mining), hoặc xử lý dữ liệu trực tuyến (olap)... để phân tích dữ liệu, tái sử dụng tri thức, hoặc rút trích ra các thông tin quý giá, cần thiết cho việc ra quyết định và cải tiến các hoạt động kinh doanh.

### 3.2.4. Ngôn ngữ XML:

Ngôn ngữ đánh dấu mở rộng - XML (eXtensible Markup Language) là một kỹ thuật phát triển tương tự ngôn ngữ đánh dấu siêu văn bản HTML (HyperText Markup Language). Đây là 1 chuẩn mới về dữ liệu trên Internet, giúp cho các ứng dụng dựa trên các hệ quản trị CSDL khác nhau có thể hiểu và nói chuyện được với nhau.

Vì việc chuyển đổi dữ liệu được tiến hành qua 1 hệ thống chung (web), việc tương thích không còn là vấn đề lớn. Trong quá khứ, các công ty với các hệ thống quản lý thông tin không tương thích có thể gặp khó khăn khi thực hiện các giao dịch. XML, một chuẩn phát triển web, có thể được dùng để cải tiến sự tương thích giữa các hệ thống riêng rẽ, tạo ra các cơ hội cho thị trường mới. Ngày nay, hầu hết các ứng dụng trên web đều hỗ trợ chuẩn XML. Hơn nữa, người ta còn sử dụng XML để biểu diễn ngữ nghĩa của trang web, từ đó giúp cho việc tìm kiếm thông tin trên Internet hiệu quả và chính xác hơn.

Chẳng hạn, nhà phát triển XML có thể mã hóa dữ liệu trong một danh mục sản phẩm bằng XML. Mỗi sản phẩm trong danh mục được gán một thẻ mô tả kích thước, màu sắc, giá cả, nhà cung cấp, thời gian chờ ước lượng và chính sách giảm giá. Vì XML có thể được sử dụng với nhiều hệ thống và nền tảng, các công ty có thể cung cấp dữ liệu danh mục của nó trên nhiều địa chỉ trao đổi B2B. Tên sản phẩm, giá cả và các dữ liệu mô tả khác được định dạng tự động để phù hợp với hình thức và cảm nhận về một địa chỉ.

### ***3.3. Ứng dụng của thương mại điện tử:***

#### ***3.3.1. Các cấp độ ứng dụng TMĐT:***

Có 2 cách phân chia theo cấp độ ứng dụng TMĐT: chia theo 3 cấp độ hoặc chia theo 6 cấp độ.

Chia theo 3 cấp độ:

1. **Thương mại thông tin (i-commerce):** doanh nghiệp có website trên mạng để cung cấp thông tin về sản phẩm, dịch vụ... Các hoạt động mua, bán vẫn thực hiện theo cách truyền thống.

2. **Thương mại giao dịch (t-commerce):** doanh nghiệp cho phép khách hàng thực hiện các giao dịch đặt hàng, mua hàng qua website trên Internet, có thể có hoặc chưa có thanh toán trực tuyến.
3. **Thương mại tích hợp (c-business):** website của doanh nghiệp liên kết trực tiếp với cơ sở dữ liệu trong mạng nội bộ của doanh nghiệp, mọi hoạt động truyền dữ liệu được tự động hóa, hạn chế sự can thiệp của con người, nhờ đó làm giảm đáng kể chi phí hoạt động và tăng hiệu quả.

Chia theo 6 cấp độ:

1. **Hiện diện trên mạng:** doanh nghiệp bắt đầu có website trên mạng, tuy nhiên website chỉ rất đơn giản, cung cấp một số thông tin ở mức tối thiểu về doanh nghiệp và sản phẩm dưới dạng các trang web tĩnh và không có các chức năng phức tạp khác.

2. **Có website chuyên nghiệp:** doanh nghiệp có website với cấu trúc phức tạp hơn, có nhiều chức năng tương tác với người xem, có chức năng cập nhật nội dung, giúp người xem liên lạc trực tiếp với doanh nghiệp một cách thuận tiện.

3. **Chuẩn bị TMĐT:** doanh nghiệp bắt đầu triển khai bán hàng hay dịch vụ qua mạng. Tuy nhiên, các giao dịch trên mạng chưa được kết nối với cơ sở dữ liệu nội bộ, vì vậy việc xử lý giao dịch còn chậm và kém an toàn.

4. **Áp dụng TMĐT:** website của doanh nghiệp được kết nối trực tiếp với cơ sở dữ liệu nội bộ, mọi giao dịch đều được tự động hóa với rất ít sự can thiệp của con người, vì thế giảm đáng kể chi phí hoạt động và tăng hiệu quả kinh doanh.

5. **TMĐT không dây:** doanh nghiệp áp dụng TMĐT trên các thiết bị không dây như điện thoại di động, máy trợ giúp cá nhân kỹ thuật số (palm),... sử dụng giao thức truyền vô tuyến - wap.

6. **Tất cả trong một máy tính:** chỉ với 1 thiết bị điện tử, người ta có thể truy cập và tìm kiếm các thông tin về doanh nghiệp, sản phẩm, dịch vụ... và thực hiện các giao dịch trực tuyến mọi lúc, mọi nơi.

### 3.3.2. Các hình thức chủ yếu của TMĐT:

❖ **Thư điện tử (Electronic Mail):**

Các doanh nghiệp, các cơ quan nhà nước, các cá nhân,... gửi thư cho nhau thông qua mạng, gọi là thư điện tử, thông tin trong thư điện tử không phải tuân theo một cấu trúc định trước nào.

❖ **Thanh toán điện tử (Electronic Payment):**

Là việc thanh toán tiền thông qua bản tin điện tử (electronic message), thay cho việc giao dịch dùng tiền mặt.

**VD: Trả lương bằng cách chuyển tiền vào tài khoản, trả tiền mua hàng, thẻ tín dụng...**

- **Trao đổi dữ liệu tài chính (FEDI – Financial Electronic Data Interchange):** phục vụ cho việc thanh toán điện tử giữa các công ty giao dịch với nhau bằng điện tử.
- **Tiền mặt Internet (Internet Cash):** Là tiền mặt được mua từ một nơi phát hành (ngân hàng hoặc một số tổ chức tín dụng nào đó), sau đó được chuyển tự do sang các đồng tiền khác thông qua Internet, áp dụng trong cả phạm vi một nước cũng như giữa các quốc gia (Digital cash).
- **Túi tiền điện tử (Electroninc Purse):** là nơi để cất tiền mặt Internet, chủ yếu là thẻ thông minh (Smart card), còn gọi là thẻ giữ tiền (stored value card), tiền được trả cho bất kỳ ai đọc được thẻ đó; kỹ thuật của túi tiền điện tử tương tự như kỹ thuật áp dụng cho “tiền mặt Internet”.
- **Giao dịch ngân hàng số hóa (Digital Banking), giao dịch chứng khoán số hóa (Digital Sercurities Trading):** hệ thống thanh toán của ngân hàng là một hệ thống lớn gồm nhiều hệ thống nhỏ:
  - Thanh toán giữa ngân hàng với khách hàng.
  - Thanh toán giữa ngân hàng với các đại lý thanh toán (Siêu thị, nhà hàng)
  - Thanh toán nội bộ một hệ thống ngân hàng.
  - Thanh toán giữa hệ thống ngân hàng này với hệ thống ngân hàng khác.

**3.3.3. Trao đổi dữ liệu điện tử (EDI – Electronic Data Interchange)**

Là việc trao đổi các dữ liệu dưới dạng “có cấu trúc” (structured form) từ máy tính điện tử này sang máy tính điện tử khác, giữa các công ty hoặc đơn vị đã thỏa thuận

buôn bán với nhau, theo cách này sẽ tự động hóa hoàn toàn không cần có sự can thiệp của con người.

Theo ủy ban liên hợp quốc về luật thương mại quốc tế (UNCITRAL - United Nations Commission on International Trade Law), việc trao đổi dữ liệu điện tử được xác định như sau: “Trao đổi dữ liệu điện tử là việc chuyển giao thông tin từ máy tính điện tử này sang máy tính điện tử khác bằng phương tiện điện tử, có sử dụng một tiêu chuẩn đã được thỏa thuận để cấu trúc thông tin”.

EDI thường là các nội dung sau:

- Giao dịch kết nối.
- Đặt hàng.
- Giao dịch gửi hàng.
- Thanh toán

#### **3.3.4. Truyền dung liệu:**

Dung liệu (content) là nội dung của hàng hóa, mà không phải là bản thân vật mang nội dung đó; ví dụ: tin tức, nhạc phim, các chương trình truyền hình, truyền thanh, các phần mềm, ý kiến tư vấn, hợp đồng bảo hiểm,...

Trước đây dung liệu được trao đổi dưới dạng hiện vật, còn ngày nay nó được số hóa và truyền gửi theo mạng, được gọi là “Giao gửi số hóa” (digital delivery).

#### **3.3.5. Bán lẻ hàng hóa hữu hình (Retail of tangible goods)**

Loại hoạt động mua bán hàng hóa bán lẻ qua mạng thường được gọi là “mua hàng điện tử” (Electronic shopping) hay “mua hàng trên mạng”.

Hoạt động quảng cáo, trưng bày và giới thiệu sản phẩm của nhà người bán và hoạt động xem hàng, xác nhận mua và thanh toán của khách hàng bằng thanh toán điện tử thông qua một đối tượng trung gian là các trang web hay là các “cửa hàng ảo” (Virtual shop) của người bán trên mạng.

#### **3.3.6. Công nghệ ứng dụng trong TMDT:**

##### **❖ Hệ thống thông tin:**

Chia thành các loại:

- Hệ thống xử lý giao dịch (Transaction Processing Systems).
- Hệ thống quản trị thông tin (Management Information Systems).
- Hệ thống hỗ trợ ra quyết định (Decision Support Systems).
- Hệ chuyên gia (Expert Systems).

Có thể phân chia các giai đoạn phát triển hệ thống thông tin dựa trên chu kỳ sống của phát triển hệ thống (System Development Life Cycle) gồm **bảy giai đoạn**:

- **Xác định vấn đề, cơ hội, mục tiêu:**
  - Xác định **vấn đề** cần sự giúp đỡ của hệ thống thông tin.
  - **Cơ hội** là những tình thế có thể cho phép sử dụng hệ thống thông tin để tăng tính cạnh tranh.
  - Xác định **mục tiêu** trả lời câu hỏi “Cần phải làm những gì?”.
- **Xác định các yêu cầu thông tin:**
  - Xác định các **yêu cầu thông tin** mà người dùng hệ thống đưa ra.
  - Kết quả giai đoạn này là báo cáo chức năng tổng thể cũng như từng bộ phận.

=> Các kết quả này cần có sự thống nhất của người sử dụng và được viết ở dạng phổ thông dễ hiểu.

- **Phân tích, đề xuất hệ thống:**
  - Xác định sơ đồ dòng dữ liệu.
  - Xác định sơ đồ xử lý quyết định.

=> Kết quả giai đoạn này là một đề xuất hệ thống mới tổng hợp những gì đã phân tích, cung cấp phân tích chi phí của nhiều phương án giải pháp, và một đề xuất giải pháp tối ưu.

- **Thiết kế hệ thống mới:**
  - Thiết kế logic.
  - Thiết kế biểu mẫu và màn hình.
  - Thiết kế sơ đồ thực thể - quan hệ.
  - Thiết kế cơ sở dữ liệu.
  - Thiết kế đầu ra trên màn hình, máy in, đĩa cứng.

=> Kết quả giai đoạn này là một tài liệu mô tả chi tiết các trang màn hình vào-ra, các sơ đồ, các bảng của cơ sở dữ liệu cùng với thủ tục kiểm tra tính đúng đắn của dữ liệu.

- **Triển khai phần mềm:**

- Thiết kế các sơ đồ khối hoặc siêu thủ tục.
- Cài đặt các mã lệnh.
- Loại bỏ các sai sót.

=> Kết quả giai đoạn này là hệ thống chương trình đã cài đặt, cùng với tư liệu lập trình mô tả chi tiết các thuật giải, thủ tục, hướng dẫn sử dụng, giúp đỡ trực tuyến.

- **Kiểm định hệ thống:**

- Việc kiểm định thông qua các dữ liệu giả định và theo hai phương pháp:
- kiểm định từng bộ phận và kiểm định theo chức năng.
- Kiểm định từng bộ phận: bảo đảm từng mô đun không có sai sót.
- Kiểm định theo chức năng: nhằm bảo đảm chức năng chạy thông suốt.

- **Cài đặt và đánh giá hệ thống**

- Cài đặt tại vị trí người dùng.
- Đào tạo người dùng.
- Đổi các tập tin đang có thành các tập tin cơ sở dữ liệu.
- Đánh giá hệ thống: phải thỏa mãn những yêu cầu đã đặt ra.

### **3.3.7. Công nghệ Web:**

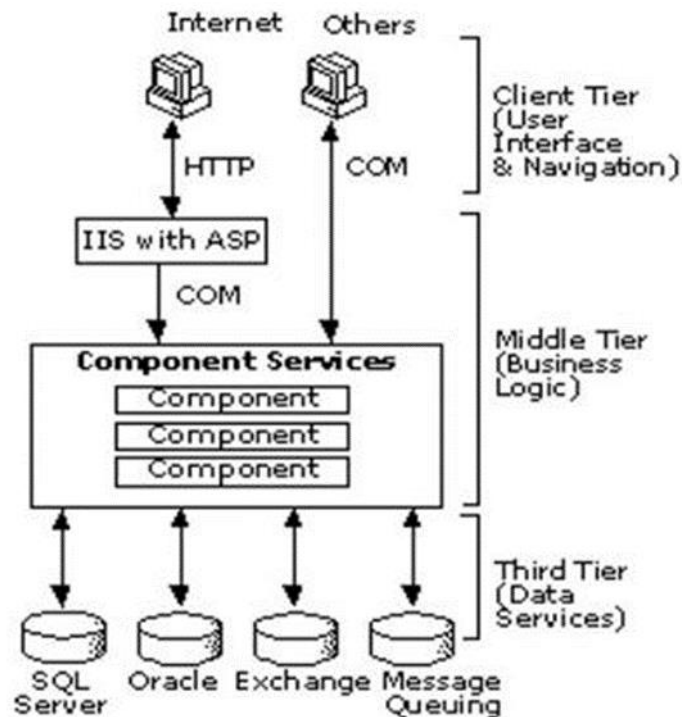
Hệ thống thông tin dựa trên web là hệ thống được thiết kế dựa trên: Các dịch vụ Internet, chủ yếu dùng trình duyệt web (browser) phía người dùng;

Dùng mô hình client-server để thiết kế các ứng dụng phía client và phía server. Công nghệ phần mềm dựa trên web còn gọi là **công nghệ web**.

- **Công nghệ Client-Server:**

- **Client/server** là mô hình mạng trong đó **máy chủ xử lý các yêu cầu từ máy khách**. - Thiết kế **multi-tier**: Một ứng dụng có 3 nhiệm vụ *nhập dữ liệu, ghi nhớ dữ liệu, xử lý dữ liệu*. Do đó người ta chia ra làm 3 tier (hoặc nhiều tier).

- **Client Tier** Giao diện người dùng hay lớp trình diễn.
- **Middle Tier** Các thành phần mô phỏng logic của tổ chức (ASP). - **Third Tier** Hệ thống quản trị Cơ sở dữ liệu (DBMS).



Hình 19. Mô tả công nghệ Client-Server

- **Data Binding:**
  - Data binding cho phép đưa dữ liệu từ bên ngoài chèn vào trong trang web.
  - Dữ liệu có thể lấy từ các biến, các trường, biểu mẫu, cơ sở dữ liệu.
  - Dùng data binding, có thể quản trị dữ liệu từ xa ngay trong trình duyệt.
- **Ứng dụng Client-Side:**
  - Các ứng dụng trong phạm vi máy client, được hỗ trợ bởi trình duyệt. Có thể dùng các phần tử HTML: text, images, multimedia, script,...
  - Ứng dụng client-side không xử lý đầy đủ tính năng của một hệ ứng dụng.
- **Ứng dụng Server-Side:**
  - Ứng dụng server-side là cốt lõi của hệ ứng dụng, nó bao gồm thu nhận yêu cầu của client, thực hiện các xử lý và trả lại kết quả cho client.
  - Công nghệ: CGI, ASP, PHP, JSP,...



## **CHƯƠNG III: ỨNG DỤNG CỦA WEB SERVICE TRONG TMDT**

Ngày nay các dịch vụ web đã trở nên rất thông dụng. Nó được ứng dụng rất rộng rãi trong các lĩnh vực của cuộc sống, trong Công nghệ thông tin nói chung và trong thương mại điện tử nói riêng. Những ứng dụng của Web service trong đời sống có thể kể tên đó là:

- Dịch vụ chọn lọc và phân loại tin tức: là những hệ thống thư viện kết nối đến các web portal để tìm kiếm các thông tin từ các nhà xuất bản có chứa những từ khóa muốn tìm.
- Dịch vụ hiển thị danh sách đĩa nhạc dành cho các công ty thu thanh.
- Ứng dụng đại lý du lịch có nhiều giá vé đi du lịch khác nhau do có chọn lựa phục vụ của nhiều hãng hàng không.
- Bảng tính toán chính sách bảo hiểm dùng công nghệ Excel/COM với giao diện web
- Thông tin thương mại bao gồm nhiều nội dung, nhiều mục tin như: dự báo thời tiết, thông tin sức khỏe, lịch bay, tỷ giá cổ phiếu,...
- Những giao dịch trực tuyến cho cả B2B và B2C như: đặt vé máy bay, làm giao kèo thuê xe.
- Hệ thống thông tin dùng Java để tính toán tỷ giá chuyển đổi giữa các loại tiền tệ. Hệ thống này sẽ được các ứng dụng khác dùng như một web service

Còn trong lĩnh vực Thương mại điện tử nói riêng, Web service có thể được ứng dụng trong tất cả các hoạt động của thương mại điện tử như: quảng cáo trực tuyến, sử dụng trong các công cụ tìm kiếm, ứng dụng trong giao dịch mua bán hàng trực tuyến và trong dịch vụ thanh toán.

### **1. Ứng dụng trong quảng cáo trực tuyến:**

Trong thế giới rộng lớn của Internet với hàng tỷ người đang sử dụng hàng ngày, rất nhiều trong số họ thường đọc quảng cáo trực tuyến khi lướt web và đó là cơ hội lớn để các công ty tiến hành các chiến dịch quảng cáo tiếp thị hình ảnh công ty, sản phẩm - dịch vụ trực tuyến trên Internet.

Việc quảng cáo trên mạng đem lại cho người mua quảng cáo rất nhiều lợi ích mà các hình thức quảng cáo truyền thống khác không thể có được hoặc có nhưng hiệu quả đem lại không cao.

Quảng cáo trực tuyến có rất nhiều ưu điểm như: khả năng nhắm chọn khách hàng, khả năng có thể theo dõi hiệu quả của việc quảng cáo; tính linh hoạt và khả năng phân phối rộng rãi; tính tương tác nhanh gọn, trong một phạm vi rộng, với số lượng người sử dụng rất đông đảo trên mạng Internet. Quảng cáo trực tuyến có các hình thức thông dụng như sau:

### ***1.2. Quảng cáo logo – banner, pop-up:***

Đặt logo hoặc banner quảng cáo trên các website nổi tiếng, những website có lượng khách hàng truy cập lớn hay những website được rank cao trên Google là phổ biến và là cách quảng cáo trực tuyến hiệu quả nhất hiện nay. Nó không những quảng bá được thương hiệu mà còn nhắm đến các khách hàng tiềm năng trên Internet. Hoạt động này ít có sự tham gia của Web service.

### ***1.3. Quảng cáo bằng đường Text link***

Là đặt quảng cáo bằng chữ có đường link đến website hay sản phẩm dịch vụ của người mua quảng cáo, họ phải có tiêu đề cho đoạn quảng cáo, địa chỉ website, thông tin giới thiệu về website hay quảng cáo sản phẩm dịch vụ để họ có thể đăng ký vào bất kỳ danh bạ nào trên Internet phục vụ việc tra cứu, tìm kiếm sản phẩm dịch vụ của các cỗ máy chủ tìm kiếm. Lợi ích của hình thức quảng cáo này là thường xuyên có hàng trăm ngàn robot và spider của các search engine truy cập vào và sẽ tự động cập nhật website của người mua quảng cáo lên search engine trong vòng chỉ có ba đến bảy ngày. Hoạt động này được thực hiện dựa trên Web Service.

### ***1.4. Quảng cáo tài trợ tại Google, Yahoo!, MSN, Altavista...***

Chúng ta lâu nay vẫn sử dụng Google, Yahoo, MSN hay Altavista - những công cụ tìm kiếm mạnh nhất với việc cung cấp miễn phí các công cụ: Tìm kiếm thông tin, Email, Tin tức, Chat, Điện thoại Internet,...mà tất cả đều miễn phí – kinh phí của họ dựa trên những hoạt động quảng cáo đi kèm với việc tìm kiếm của người dùng.

Nếu chúng ta chịu khó quan sát sẽ dễ dàng nhận ra: Trên mỗi trang kết quả tìm kiếm, với bất cứ từ khoá nào, đều xuất hiện một đến vài mẫu quảng cáo được đóng

trong khung ở bên phải màn hình hoặc nằm trên cùng và dưới cùng của trang đầu (hay trang 2, 3) kết quả tìm kiếm. Một nguyên tắc để nhận biết những kết quả tìm thấy đó là quảng cáo là ở dòng chữ: "**Sponsors**" (Nhà tài trợ). "Sponsorship Advertising" (quảng cáo tài trợ) là một thuật ngữ mới được áp dụng vào Internet từ năm 1996 bởi Hotwire và tiếp theo là AOL (American Online).

Đây là hệ thống quảng cáo mới nên nó có tính năng thông minh, nhắm chọn. Khi khách hàng đánh vào một từ khoá bất kỳ trong ô tìm kiếm là các cỗ máy tìm kiếm lập tức mang một nhà tài trợ (sponsors) có liên quan đặt ngay lên đầu kết quả tìm kiếm. Công việc này thường được dựa trên nền tảng và cách thức hoạt động của Web service. Có hai cách mà nhà quảng cáo có thể chọn một để trả cho các Search Engine: một là trả theo tổng số lần truyền phát quảng cáo - thường được gọi là CPM (Cost per Impression - trả tiền cho một ngàn lần truyền phát quảng cáo). Cách này thực hiện như sau: Google hay Yahoo! Sử dụng Web service cùng một vài công nghệ khác để cho ra kết quả tìm kiếm của người dùng đi kèm nhà tài trợ (quảng cáo) mỗi khi có một từ khoá liên quan được tìm kiếm. Kết quả sẽ hiển thị trên hàng bên phải hay trên cùng, hay dưới cùng của trang 1, trang 2 hay trang 3 kết quả tìm kiếm (tùy theo số tiền người mua quảng cáo Bid - đấu giá) kèm theo là hàng chữ đậm màu: SPONSORS (nhà tài trợ). Ngoài ra, Google, Yahoo còn thiết đặt nhiều địa chỉ quảng cáo tại các site có nhiều người truy cập theo từng chủ đề nhất định (hoặc có thể sử dụng UDDI – là các trang vàng của Web Service). Nếu người dùng vào một website, thấy một mẫu quảng cáo có dòng chữ nhỏ phía trên hay phía dưới: "Ads by Google", "Goooooogle...." hay "Advertisment by Yahoo!" thì đó chính là "Sponsorship advertising".

Sau cách phải trả cho mỗi một ngàn lần truyền phát quảng cáo như vậy, người mua quảng cáo có thể chọn cách thứ 2: chỉ phải trả tiền nếu có khách hàng nhấn vào dòng quảng cáo của mình (để liên kết đến website của họ). Mỗi một lần khách hàng bấm vào dòng quảng cáo của họ được gọi là một "**Nhấn**" (Click). Phương thức quảng cáo này đang ngày càng thông dụng và được gọi với cái tên: CPC (Cost per click - trả tiền cho mỗi "nhấn") hay PPC (Pay per click).

## **2. Ứng dụng trong các công cụ tìm kiếm:**

Các search engine hiện đại ngày này đều cung cấp các web services để cho phép các ứng dụng có thể truy xuất trực tiếp đến các server của các engine, điều này cho phép chúng ta có thể xây dựng được các chức năng tìm kiếm khác nhau theo những yêu cầu cụ thể trên các ứng dụng của chúng ta. Và đặc biệt là, ngày nay việc xây dựng các ứng dụng trên mạng đang được triển khai rất nhiều, thì việc xây dựng các chức năng tìm kiếm theo các yêu cầu khác nhau là thực sự cần thiết.

Các công cụ tìm kiếm ngày nay sử dụng phương pháp tạo chỉ mục cho các websites, các hình ảnh, các nhóm, các thư mục theo một tiêu chí riêng để các kết quả tìm kiếm tương đồng với câu truy vấn của người dùng nhất bằng các thuật toán và các phương pháp tạo chỉ mục phù hợp.

### **❖ Công cụ tìm kiếm google:**

Trong những năm gần đây, công cụ tìm kiếm Google ngày càng được phát triển và mở rộng để trở thành công cụ tìm kiếm thông dụng nhất trên web. Ngoài những chức năng tìm kiếm thông thường, Google còn cho phép tìm kiếm theo các tiêu chí khắt khe hơn như là tìm kiếm theo đúng yêu cầu, tìm kiếm theo quyền truy xuất và có thể yêu cầu kết quả trả về là những trang web đều sử dụng một ngôn ngữ nào đó theo sự chọn lựa của người yêu cầu tìm kiếm ban đầu. Tuy nhiên chỉ có một số nhỏ người dùng web biết hết được sức mạnh thực lực của công cụ google, thông thường người dùng thường sử dụng google để tìm dựa trên từ khóa, các câu và đôi khi là các từ khóa; các câu người dùng yêu cầu tìm kiếm lại rất mơ hồ, chung chung, do đó có thể kết quả trả về cho người dùng có thể không chính xác, không tốt hoặc chung chung không đi đúng vào trọng tâm của yêu cầu tìm kiếm. Chính vì thế công cụ google phải có phần hướng dẫn cho người dùng để cho người dùng có thể nhanh chóng tìm thấy chính xác nội dung họ cần. Trong mục tiêu đưa thông tin đến người dùng và cho phép người dùng có thể tìm kiếm thông tin từ ngay bên trong ứng dụng của họ thì google đã phát triển google web services để cho phép tích hợp công cụ google vào trong các ứng dụng, điều này có nghĩa là google web services cho phép các ứng dụng gửi các yêu cầu tìm kiếm đến trực tiếp đến google server và nhận lại các kết quả. Điều này mở ra hướng mới là cho phép người dùng truy cập trực tiếp vào các hàm của google cung cấp và từ đó người dùng có thể xây dựng

riêng cho mình một công cụ tìm kiếm tốt nhất theo đúng yêu cầu của họ. Hiện tại các chức năng của google web service cung cấp là tìm kiếm, cache page và kiểm tra chính tả.

Các lập trình viên có thể phát triển các ứng dụng trong rất nhiều ngôn ngữ khác nhau, có thể là lựa chọn Java, .Net, C... để xây dựng ứng dụng và thực hiện kết nối tới các hàm được cung cấp bởi Google web service, việc giao tiếp được thực hiện thông qua SOAP. Khi đã thực hiện được kết nối các ứng dụng có thể thực hiện tất cả các chức năng mà google dịch vụ đã cung cấp như là tìm kiếm hoặc kiểm tra chính tả, các kết quả trả về sẽ được ứng dụng sử dụng cho các mục đích khác nhau của các ứng dụng. Bộ GoogleAPI cung cấp tất cả các hàm cho phép ứng dụng thực hiện các chức năng như là web site google.com đã cung cấp.

Trên mô hình này người dùng sẽ truy xuất đến Google thông qua ứng dụng có tên là **Cedasit Google**, ứng dụng này được xây dựng bằng .NET. Để triển khai ứng dụng này, người dùng phải có tài khoản ‘gmail’ để nhận license key cho việc sử dụng các hàm API, nếu họ chưa có tài khoản ‘gmail’, có thể đăng kí tài khoản tại địa chỉ: <https://www.google.com/accounts/NewAccount>. Sau khi phát triển ứng dụng, quá trình truy vấn thông tin được bắt đầu khi người dùng nhập vào các từ khóa hay một thông tin gì đó cần truy vấn, Cedasit Google sẽ lấy yêu cầu của người dùng và gửi đến Google.com bằng SOAP request, tại Google.com sau khi nhận được yêu cầu sẽ tiến hành truy xuất thông tin thông qua các cụm máy server, và sau đó gửi trả kết quả về cho ứng dụng thông qua SOAP response (SOAP server), và công việc cuối của Cedasit Google là hiển thị kết quả truy vấn của người dùng. Quá trình này được thực hiện trong suốt đối với người dùng, họ chỉ cần biết nhập vào yêu cầu cần truy vấn và đợi cho đến khi có được kết quả trả về.

So sánh kết quả thực hiện tìm kiếm của các chức năng trên ứng dụng được xây dựng từ googleAPI do Google.com cung cấp so với kết quả tìm kiếm trực tiếp tại website google.com chúng ta thấy rằng kết quả tìm kiếm là như nhau chỉ có thứ tự hiển thị là khác nhau, do đó việc xây dựng ứng dụng tìm kiếm cho riêng ứng dụng của chúng ta từ các hàm do web service của các công cụ tìm kiếm như google.com cung

cấp chẳng hạn thì cho kết quả rất tốt không khác gì so với thực hiện trực tiếp tại các công cụ tìm kiếm này.

Việc sử dụng web services của google để xây dựng công cụ tìm kiếm với 3 chức năng là tìm kiếm bình thường, tìm kiếm kết hợp, tìm kiếm theo host là một minh chứng cho thấy việc sử dụng ngày càng rộng rãi công nghệ web services trong các ứng dụng đặc biệt là các ứng dụng trên mạng để xây dựng các ứng dụng các chức năng theo những yêu cầu riêng của ứng dụng.

### **3. Ứng dụng trong giao dịch, mua bán hàng trực tuyến**

TMĐT cơ bản là việc mua bán hàng hóa, dịch vụ trên Internet hay nói một cách khác đó là các cửa hàng ảo trên Internet. Chính nhờ giảm thiểu các trung gian trong mua bán hàng hóa, dịch vụ và tiết kiệm chi phí trong việc mở cửa hàng và trưng bày sản phẩm, những doanh nghiệp chỉ dựa vào phương pháp truyền thống (trực tiếp gặp gỡ khách hàng) nay cũng có kế hoạch mở các trang web TMĐT để bán hàng và giới thiệu sản phẩm hay ít nhất cũng tham gia quảng cáo trên mạng Internet.

Do việc mua bán qua Internet hay TMĐT là hoàn toàn tự động nên yêu cầu về cơ sở hạ tầng trong hệ thống tài chính, ngân hàng và các công cụ thanh toán cũng phải đồng bộ, hiện đại, thông suốt và bảo đảm. Bằng những công nghệ phần mềm hiện đại và bản chất tương tác hai chiều của Internet, các công ty và doanh nghiệp còn có thể thu nhập được các thông tin về cá nhân người tiêu dùng và xu hướng tiêu dùng của họ thông qua các thủ tục đăng ký, đặt hàng, giao hàng và thanh toán, trong đó vai trò của Web service là rất quan trọng. Nó giúp cho việc cập nhật những sản phẩm-dịch vụ mới một cách nhanh chóng, thuận tiện, mang tính đồng bộ hóa cao giữa các hệ thống.

### **4. Ứng dụng trong dịch vụ thanh toán (Payment gateway)**

#### **4.1. Payment gateway là gì?**

Payment gateway là dịch vụ cho phép thanh toán trực tuyến ở các site thương mại điện tử như e-businesses, online retailers, bricks and clicks. Nó tương tự như một POS (Thiết bị thanh toán điện tử đầu cuối) khi thanh toán online. Payment gateway cho phép mã hóa các thông tin nhạy cảm như số CreditCard, để đảm bảo thông tin có thể bảo mật và giao dịch thuận tiện giữa người bán và người mua.

Đơn giản hơn, Payment Gateway là công cụ nối liền website thương mại điện tử của với account merchant. Gateway đơn giản chỉ có chức năng thanh toán, tuy nhiên rất nhiều nhà cung cấp dịch vụ Gateway đã cung cấp thêm các tiện ích kèm theo như shopping carts, Web hosting hay merchant accounts (Tài khoản thanh toán của doanh nghiệp khi tham gia TMĐT)

#### **4.2. Payment Gateway hoạt động thế nào?**

Khi một người mua đặt lệnh mua một đối tượng sản phẩm tại payment gateway trên website, payment gateway sẽ xử lý một loạt các tiến trình thanh toán ngầm bên trong như sau:

- Khách hàng đặt lệnh bằng cách nhập thông tin trên thẻ tín dụng và bấm vào nút ví dụ như "Xác nhận mua" trên website.
- Thông tin sẽ được mã hóa và chuyển từ Trình duyệt của khách hàng đến webserver của người bán, để thực hiện được tiến trình này, bắt buộc server phải được hỗ trợ bởi tính năng mã hóa SSL (Secure Socket Layer).
- Người bán sẽ chuyển các thông tin giao dịch đó cho payment gateway (thường là tự động). Đó lại là một tiến trình thực hiện mã hóa SSL khác của payment gateway server.
- Payment gateway sẽ nhận thông tin chi tiết về giao dịch đó và chuyển nó cho trung tâm xử lý tại acquiring bank của người bán.
- Trung tâm xử lý này sẽ chuyển thông tin giao dịch tới trung tâm thanh toán của CreditCard đó (Visa/Master)
- Trong trường hợp là thẻ American Express hay Discover Card, thì Amex và Dis kiêm luôn vai trò của acquiring bank và trực tiếp xử lý lệnh từ payment gateway (gộp 2 bước vào 1).
- Trung tâm thanh toán thẻ sẽ chuyển thông tin giao dịch tới ngân hàng phát hành thẻ.
- Ngân hàng phát hành thẻ sẽ kiểm tra tính sở hữu và gửi thông tin phản hồi tới payment gateway theo tiến trình ngược lại một mã phản hồi. Mã phản hồi đó cung cấp thông tin như chấp nhận hay không chấp nhận, lý do trong trường hợp không chấp nhận (như không đủ tiền, hay ko liên kết được với tài khoản ngân hàng .v.v.)

- Payment gateway nhận mã phản hồi này và chuyển nó tới website và thể hiện bằng một thông báo dễ hiểu đối với người mua và bán.
- Tất cả các công việc đó chỉ diễn ra trong vòng 2-3 giây.
- Người bán sau đó phải chuyển hàng hoặc đáp ứng lệnh bán trước khi được quyền yêu cầu một lệnh hoàn thành giao dịch.
- Sau đó người bán nhập các thông tin cần thiết về việc đáp ứng lệnh bán và lưu vào một "batch" để gửi tới acquiring bank để hoàn thành giao dịch.
- Acquiring bank sau đó sẽ nạp tiền vào một tài khoản người bán chỉ định.
- Tiến trình từ việc yêu cầu lệnh đến khi hoàn thành giao dịch tiền về tài khoản cuối cùng mất chừng 3 ngày (trung bình).

Các payment gateway thường cung cấp sẵn các form, các công cụ tự động tính thuế và tự động hoàn thành hồ sơ gửi tới trung tâm xử lý. Đặc biệt là các công cụ chống giả mạo như geolocation, velocity pattern analysis, delivery address verification, computer finger printing technology, identity morphing detection, AVS checks.



## **B. ỨNG DỤNG MUA HÀNG TRỰC TUYẾN TRÊN THIẾT BỊ DI ĐỘNG**

# CHƯƠNG I: TỔNG QUAN ĐỀ TÀI

## 1. Tổng quan ứng dụng

### 1.1. Phạm vi

Xây dựng ứng dụng mua thức ăn nhanh trên thiết bị di động chạy hệ điều hành Android 4.4 trở lên.

Dữ liệu sản phẩm được chứa trong MySQL database đặt trên Host. Sử dụng RESTful Web Service viết bằng PHP gửi nhận dữ liệu giữa MySQL database trên Host và thiết bị Android.

### 1.2. Mục tiêu

- Áp dụng các kiến thức đã học để xây dựng một ứng dụng trên thiết bị di động để phục vụ mục đích đặt hàng nhanh chóng qua thiết bị di động.
- Xây dựng RESTful Web Service viết bằng PHP để truyền tải dữ liệu giữa MySQL Database trên Host với thiết bị Android.
- Ứng dụng có thể đăng ký tài khoản, đăng nhập, đăng xuất cũng như chỉnh sửa thông tin tài khoản lẫn mật khẩu.
- Giao diện đơn giản nhưng thân thiện với người dùng.
- Phân loại các sản phẩm để người dùng tìm kiếm theo nhu cầu của mình dễ hơn.
- Có thể tìm kiếm theo tên sản phẩm.
- Sắp xếp sản phẩm theo tên hoặc theo giá.
- Hiện thị các đơn hàng đã đặt để người dùng dễ dàng xem lại những đơn hàng của mình cũng như trạng thái của các đơn hàng.
- Có thể gửi tin nhắn thông báo từ trên web xuống các thiết bị android đã cài ứng dụng để giúp cho các chiến dịch quảng cáo, khuyến mãi của bên bán hàng thuận lợi hơn.

### 1.3. Ý nghĩa lý luận và thực tiễn của đề tài

**Ý nghĩa lý luận:** Đề tài là tài liệu tham khảo rất hữu ích cho những ai muốn tìm hiểu về Web Service và Android. Bên cạnh đó chưa có nhiều tài liệu tham khảo về Android bằng tiếng Việt, do đó em muốn nghiên cứu muốn đóng góp hiểu biết của mình để làm giàu thêm tài liệu tham khảo cho chủ đề này.

Thứ hai là những kinh nghiệm quý báu mà em có được trong quá trình thực hiện đề tài sẽ là hành trang tuyệt vời giúp ích rất nhiều cho công việc sau này của mình.

**Ý nghĩa thực tiễn:** Sự phát triển và phồn vinh của một nền kinh tế không còn chỉ dựa vào nguồn tài nguyên thiên nhiên và nguồn lao động, mà ở mức độ lớn được quyết định bởi trình độ công nghệ thông tin và tri thức sáng tạo. Cùng với xu thế đó, TMĐT xuất hiện đã làm thay đổi bộ mặt kinh tế thế giới bởi những ảnh hưởng to lớn của mình:

- Làm thay đổi tính chất của nền kinh tế mỗi quốc gia và nền kinh tế toàn cầu.
- Làm cho tính tri thức trong nền kinh tế ngày càng tăng lên và tri thức đã thực sự trở thành nhân tố và nguồn lực sản xuất quan trọng nhất, là tài sản lớn nhất của một doanh nghiệp.
- Mở ra cơ hội phát huy ưu thế của các nước phát triển sau để họ có thể đuổi kịp, thậm chí vượt các nước đã đi trước.
- Xây dựng lại nền tảng, sức mạnh kinh tế quốc gia và có tiềm năng làm thay đổi cán cân tiềm lực toàn cầu.
- Rút ngắn khoảng cách về trình độ tri thức giữa các nước phát triển với các nước đang phát triển.

Cách mạng hoá marketing bán lẻ và marketing trực tuyến.

## **2. Khảo sát hiện trạng**

### ***2.1. Hiện trạng tổ chức***

Thương mại điện tử là một phần không thể thiếu trong cuộc sống ngày nay, khi con người ngày càng có nhu cầu về mua bán trao đổi nhưng diện tích hay yếu tố khách quan không cho phép, vậy làm thế nào để tiết kiệm chi phí nhất?

Hiện nay trên thị trường đã có những website chuyên cập nhập tình hình mua bán trên các trang thương mại điện tử của Tiki, Shoppe, SenDo, Lazada ... về những mặt hàng đang được ưa thích nhất, bán chạy nhất, giá giảm tốt nhất ... Nhưng chỉ mới dừng lại ở mức độ là website mà mọi người phải truy cập vào nó và mọi thứ dường như trở nên khá vất vả.

Nếu người dùng muốn một nơi vừa có thể xem nhưng hot deal trên những trang thương mại điện tử nổi tiếng, vừa muốn xem nhưng tin đăng bán hàng cũ của những người dùng khác để thuận tiện cho việc mua hàng của mình thì phải làm thế nào? Và hiện nay trên thị trường chưa có một website hay ứng dụng tích hợp những điều đó lại với nhau.

Và trên thị trường hiện nay những ứng dụng hay website mua bán đồ cũ đều chưa tích hợp định vị vị trí của người mua và người bán, chưa tích hợp được chỉ đường từ người mua tới người bán hay ngược lại. Việc này góp phần làm người dùng tin tưởng lẫn nhau hơn mua hàng và thuận tiện hơn trong giao dịch, mua bán, trao đổi.

Đối với những mặt hàng mà người dùng muốn đăng bán kể cả mới hay cũ, khi người dùng muốn tìm một sản phẩm mà người khác muốn giới thiệu cho mình xem thì làm thế nào để người dùng có thể tìm sản phẩm đó một cách nhanh nhất và tốn ít công sức nhất?

Xuất phát từ những nhu cầu nói trên, để có thể tạo ra một nơi tổng hợp được những hot deal trên tất cả những trang thương mại điện tử vừa là nơi mà người dùng có thể xem và đăng bán những mặt hàng cũ của mình như nhà đất, điện thoại, đồ nội thất, xe máy, xe hơi...

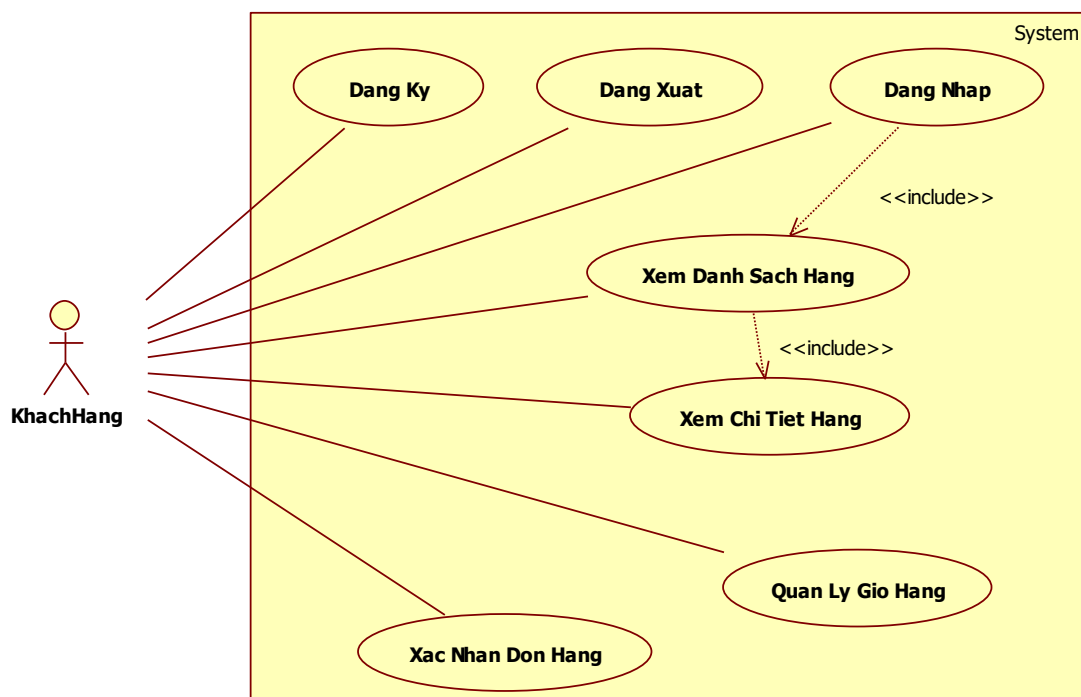
## ***2.2. Hiện trạng về nghiệp vụ***

Ban tổ chức của ứng dụng cần thực hiện việc tin học hóa việc quản lý các sản phẩm trên các trang thương mại điện tử với các nghiệp vụ sau: Tạo sản phẩm hotdeal dựa trên cách sản phẩm của trang thương mại điện tử, xóa những sản phẩm hết khuyến mãi và sửa những sản phẩm có thay đổi khuyến mãi.

0.0. 

## CHƯƠNG II: ĐẶC TẢ YÊU CẦU

### 1. Mô hình Use Case



Hình 20. Mô hình Use Case

### 2. Danh sách các Actor

STT	Tên Actor	Ý nghĩa/Ghi chú
1	User	Người sử dụng các chức năng đăng nhập, đăng xuất, đổi mật khẩu, mua bán hàng.

### 3. Danh sách các Use Case

STT	Tên Usecase	Ý nghĩa/Ghi chú
1	Đăng nhập	Đăng nhập
2	Đăng xuất	Đăng Xuất
3	Đăng ký	Đăng ký người dùng mới
4	Đổi mật khẩu	Đổi mật khẩu
5	Xem danh sách hàng	Xem danh sách hàng
6	Xem chi tiết hàng	Xem chi tiết một món hàng nào đó

7	Quản lý giỏ hàng	Quản lý giỏ hàng đã thêm vào và chuẩn bị đặt hàng
8	Xác nhận đơn hàng	Xác nhận đơn hàng đã được đặt

#### 4. Đặc tả Use Case

##### 4.1. Đặc tả Use Case “Đăng Ký”

Use Case Đăng ký	
Mô tả	Khách hàng đăng ký
Actors	Khách hàng
Điều kiện	

##### 4.2. Đặc tả usecase “Đăng nhập”

Use Case Đăng nhập	
Mô tả	Khách hàng đăng nhập vào ứng dụng
Actors	Khách hàng
Điều kiện	Khách hàng đã có tài khoản

##### 4.3. Đặc tả usecase “Đăng xuất”

Use Case Đăng xuất	
Mô tả	Khách hàng đăng xuất khỏi ứng dụng
Actors	Khách hàng
Điều kiện	Đã đăng nhập vào hệ thống

##### 4.4. Đặc tả Usecase “Xem danh sách hàng”

Use Case Xem danh sách hàng	
Mô tả	Xem danh sách sản phẩm trong ứng dụng
Actors	Khách hàng
Điều kiện	Đã đăng nhập vào hệ thống

##### 4.5. Đặc tả Usecase “Xem chi tiết hàng”

Use Case Xem chi tiết hàng	
Mô tả	Xem chi tiết của 1 sản phẩm trong ứng dụng
Actors	Khách hàng
Điều kiện	Đăng nhập vào hệ thống và chọn 1 sản phẩm muốn xem

#### ***4.6. Đặc tả Usecase “Quản lý giỏ hàng”***

<b>Use Case Quản lý giỏ hàng</b>	
Mô tả	Khách hàng quản lý giỏ hàng của mình
Actors	Khách hàng
Điều kiện	Đăng nhập vào hệ thống và có sản phẩm trong giỏ hàng

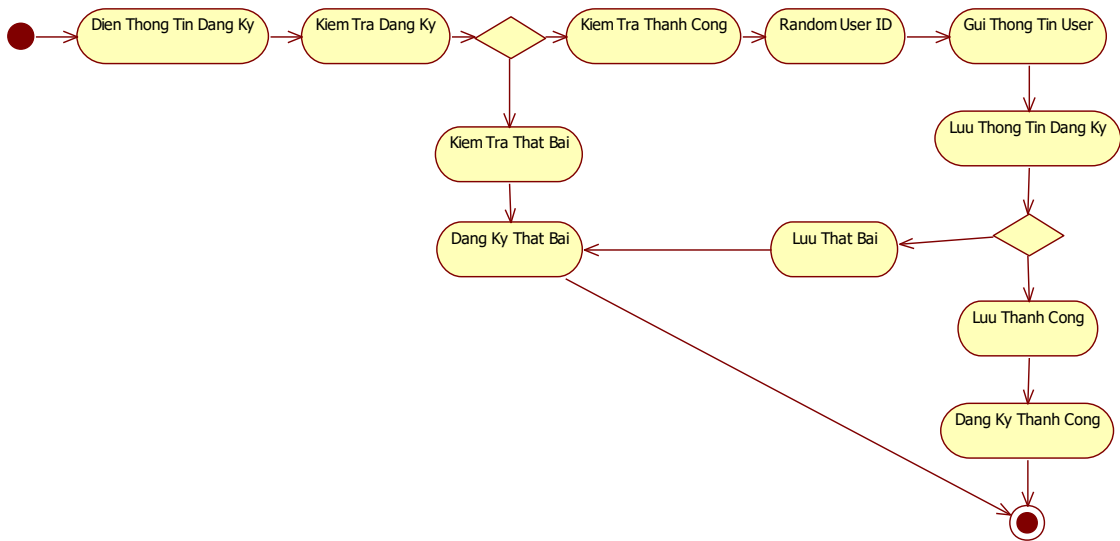
#### ***4.7. Đặc tả Usecase “Xác nhận đơn hàng”***

<b>Use Case Xác nhận đơn hàng</b>	
Mô tả	Khách hàng mua hàng và xác nhận đơn hàng
Actors	Khách hàng
Điều kiện	Đăng nhập vào hệ thống và có sản phẩm trong giỏ hàng

## CHƯƠNG III: PHÂN TÍCH HỆ THỐNG

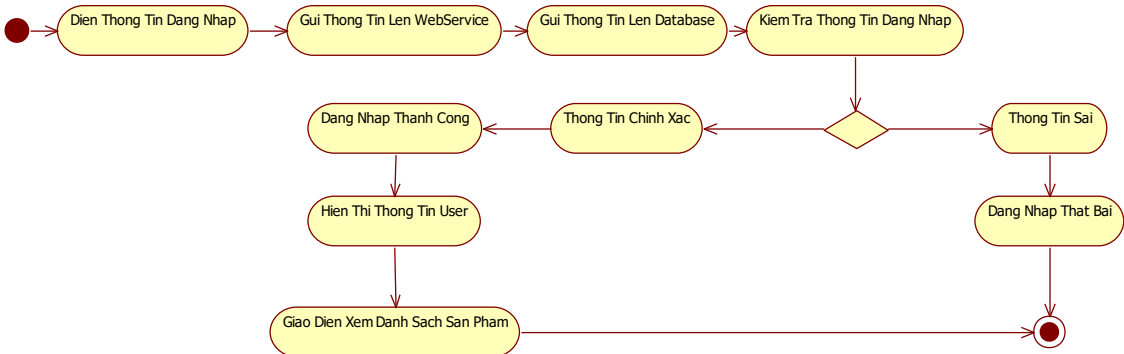
### 1. Activity Diagram

#### 1.1. Đăng ký



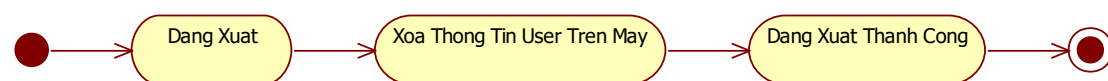
Hình 21. Mô hình Activity thể hiện quá trình đăng ký

#### 1.2. Đăng nhập



Hình 22. Mô hình Activity thể hiện quá trình đăng nhập

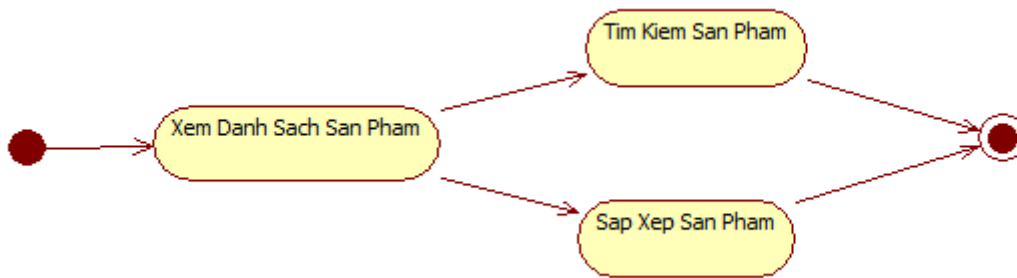
#### 1.3. Đăng xuất



Hình 23. Mô hình Activity thể hiện quá trình đăng xuất



#### 1.4. Xem danh sách sản phẩm



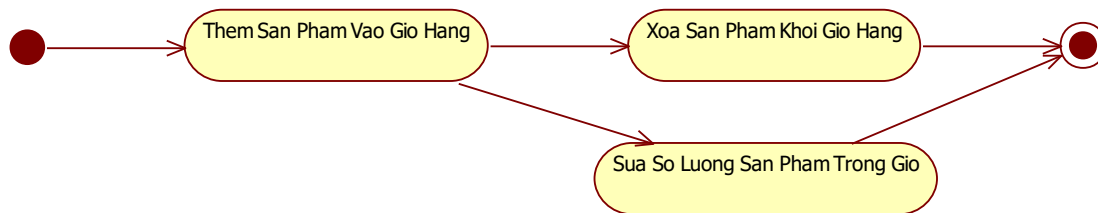
Hình 24. Mô hình Activity thể hiện quá trình xem danh sách sản phẩm

#### 1.5. Xem chi tiết sản phẩm



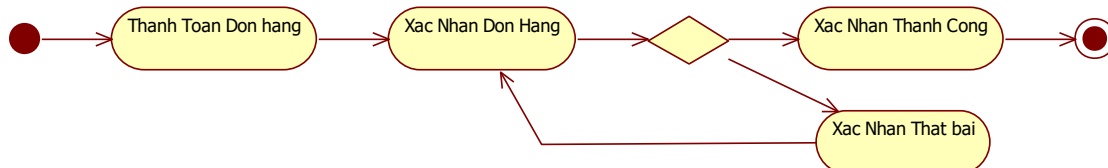
Hình 25. Mô hình Activity thể hiện quá trình xem chi tiết sản phẩm

#### 1.6. Quản lý giỏ hàng của User



Hình 26. Mô hình Activity thể hiện quá trình quản lý giỏ hàng của User

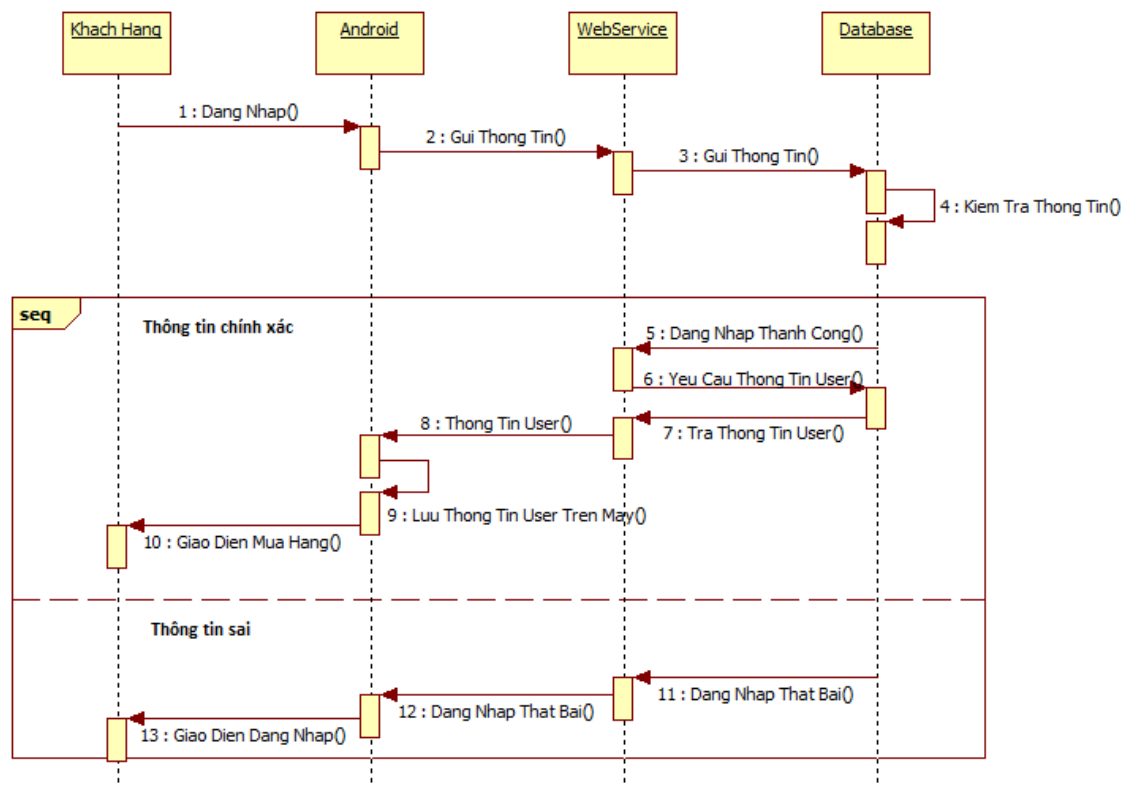
#### 1.7. Quá trình mua hàng của User



Hình 27. Mô hình Activity thể hiện quá trình mua hàng của User

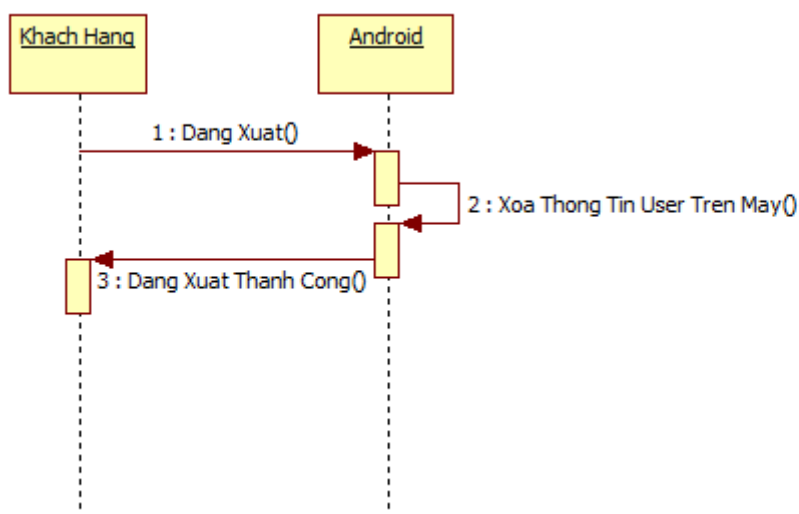
## 2. Sơ đồ tuần tự

### 2.1. Đăng nhập



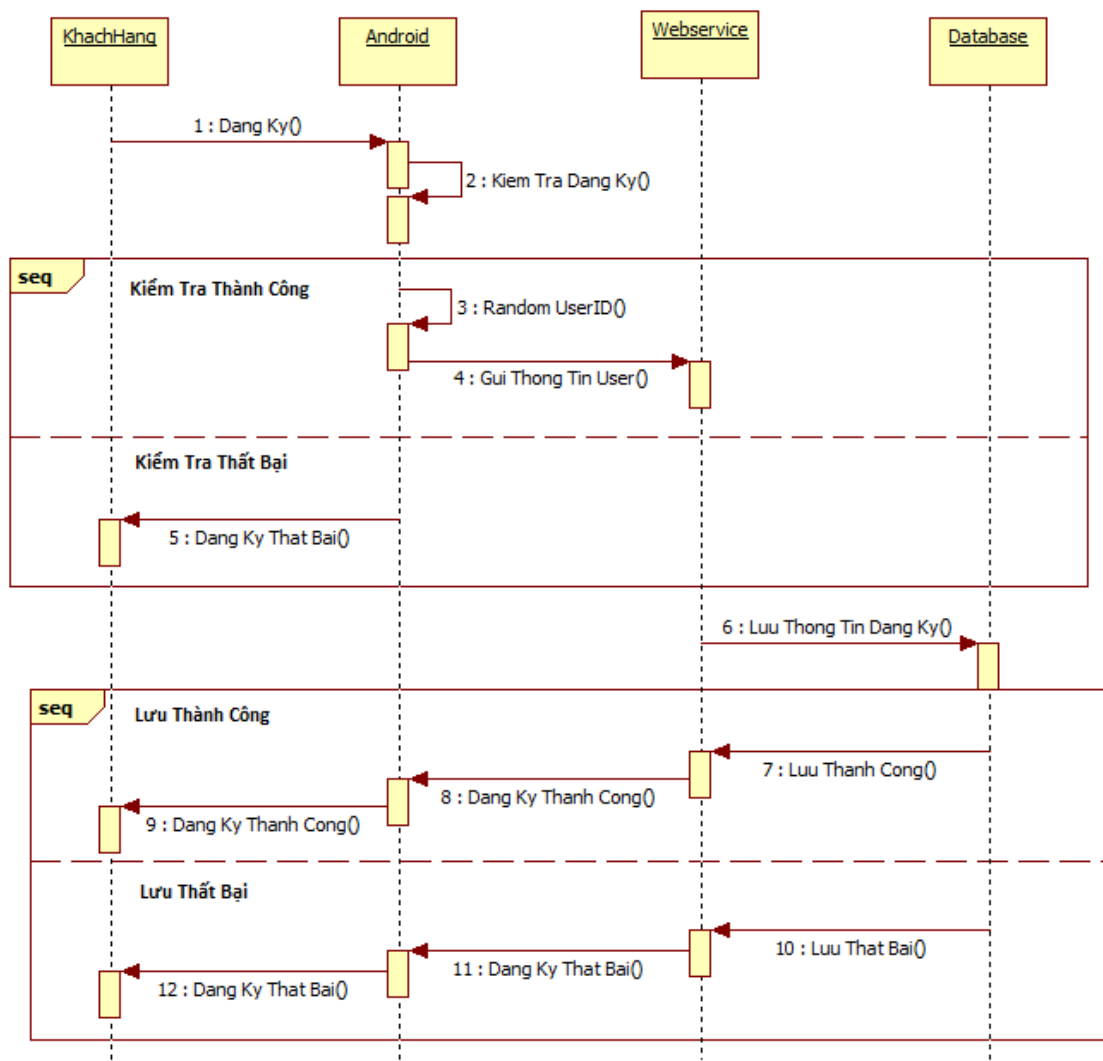
Hình 28. Mô hình Sequence thể hiện quá trình đăng nhập

### 2.2. Đăng xuất



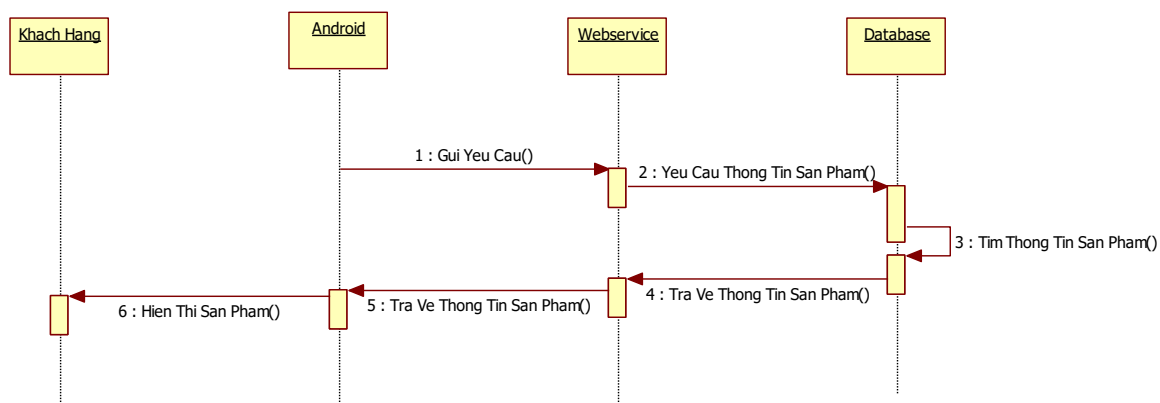
Hình 29. Mô hình Sequence thể hiện quá trình đăng xuất

### 2.3. Đăng ký



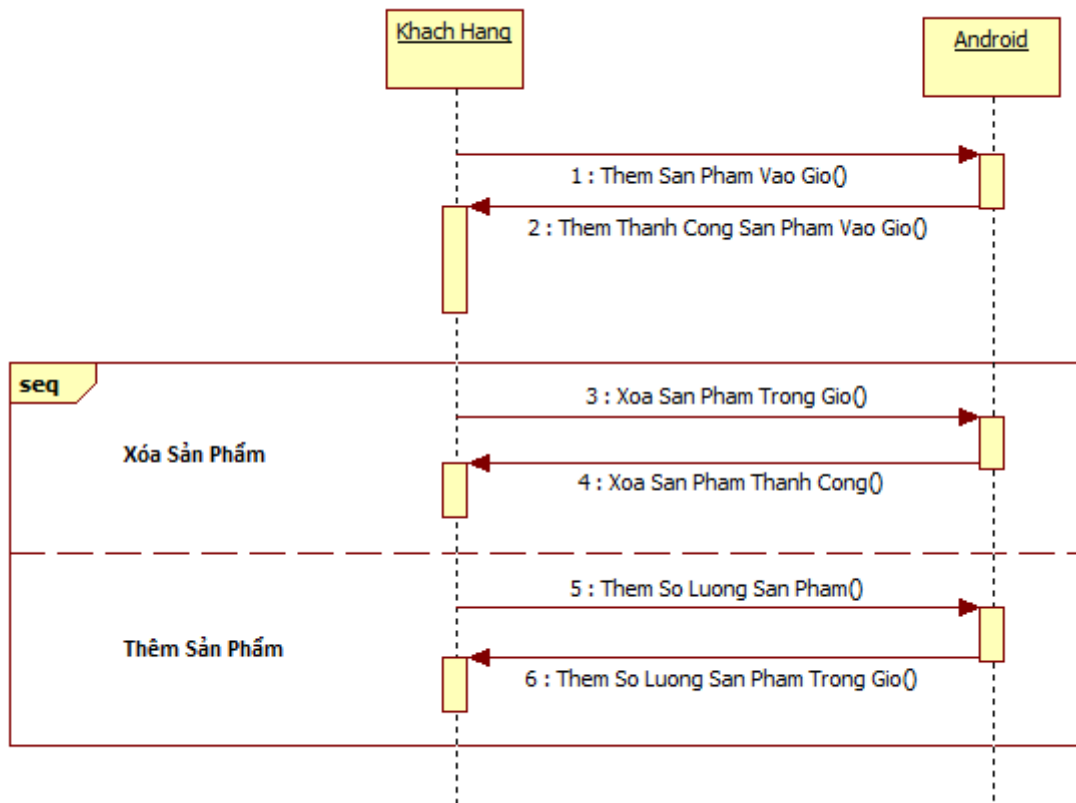
Hình 30. Mô hình Sequence thể hiện quá trình đăng ký

### 2.4. Xem chi tiết sản phẩm



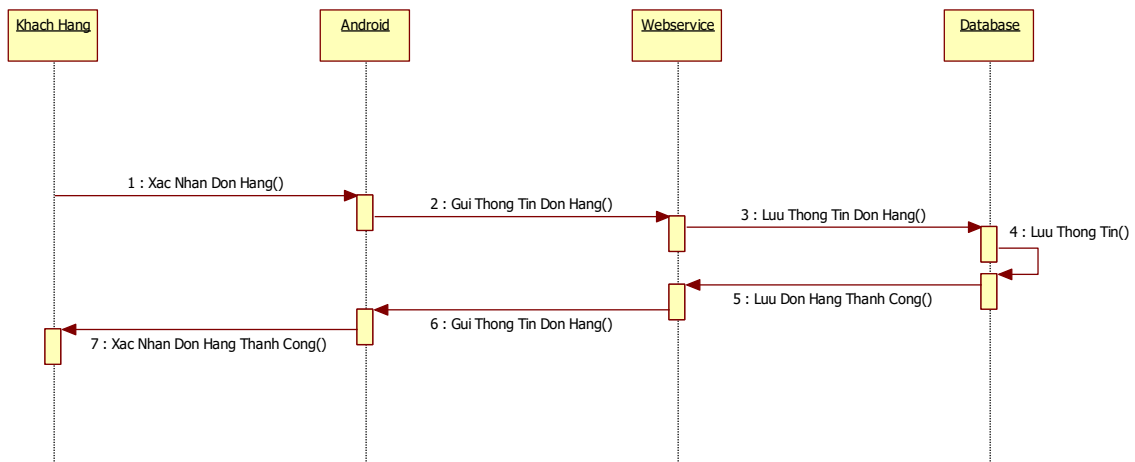
Hình 31. Mô hình Sequence thể hiện quá trình xem chi tiết sản phẩm

## 2.5. Quản lý giỏ hàng



Hình 32. Mô hình Sequence thể hiện quá trình quản lý giỏ hàng

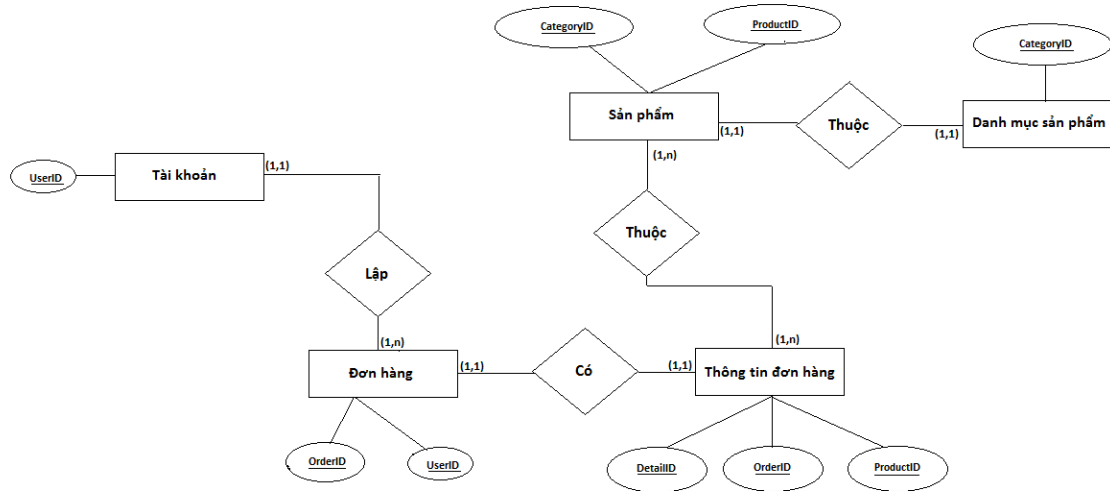
## 2.6. Quá trình xác nhận đơn hàng



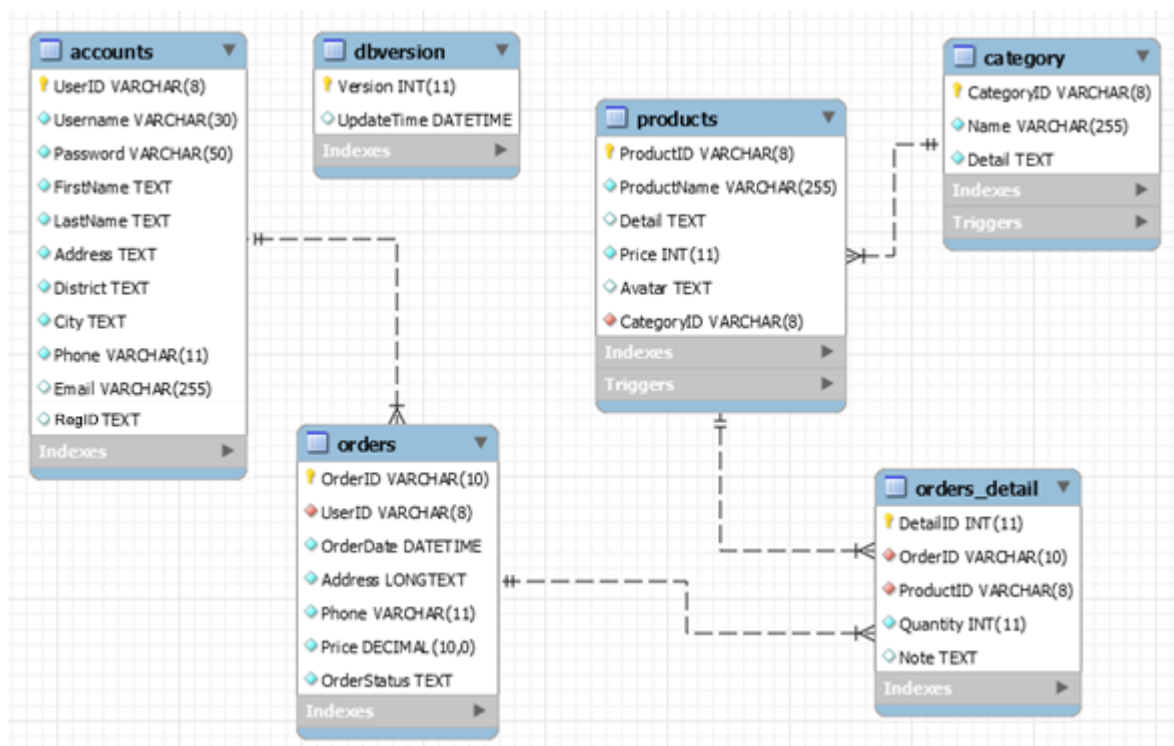
Hình 33. Mô hình Sequence thể hiện quá trình xác nhận đơn hàng

## CHƯƠNG IV: XÂY DỰNG ỨNG DỤNG

### 1. Mô hình hóa dữ liệu



Hình 35. Mô hình quan hệ giữa các bảng của Database



Hình 36. Mô hình database của ứng dụng

### 2. Đặc tả mô hình

Bảng “accounts\_detail”

Tên thuộc tính	Kiểu dữ liệu	Null	Ý nghĩa
UserID	varchar(8)	Không để trống	Mã khách hàng
Username	varchar(30)	Không để trống	Tài khoản
Password	varchar(50)	Không để trống	Mật khẩu
FirstName	text	Không để trống	Tên khách hàng
LastName	text	Không để trống	Họ và tên lót
Address	text	Không để trống	Địa chỉ
District	text	Không để trống	Quận
City	text	Không để trống	Thành phố
Phone	varchar(11)	Không để trống	Số điện thoại
Email	varchar(255)	Có thể để trống	Địa chỉ email
RegID	text	Có thể để trống	ID được tạo ra từ GCM để nhận tin nhắn GCM.

Bảng “category”

Tên thuộc tính	Kiểu dữ liệu	Null	Ý nghĩa
CategoryID	varchar(8)	Không để trống	Mã danh mục sản phẩm
Name	varchar(255)	Không để trống	Tên danh mục sản phẩm
Detail	text	Không để trống	Thông tin danh mục sản phẩm

Bảng “dbversion”

Tên thuộc tính	Kiểu dữ liệu	Null	Ý nghĩa
Version	int(11)	Không để trống	Phiên bản ứng dụng
UpdateTime	datetime	Có thể để trống	Thời gian cập nhật phiên bản

Bảng “orders”

Tên thuộc tính	Kiểu dữ liệu	Null	Ý nghĩa
OrderID	varchar(10)	Không để trống	Mã đơn hàng
UserID	varchar(8)	Không để trống	Tài khoản
OrderDate	Datetime	Không để trống	Ngày lập đơn hàng
Address	longtext	Không để trống	Địa chỉ khách hàng
Phone	varchar(11)	Không để trống	Số điện thoại khách hàng
Price	decimal(10,0)	Không để trống	Tổng giá trị của đơn hàng
OrderStatus	text	Không để trống	Trạng thái đơn hàng

Bảng “orders\_detail”

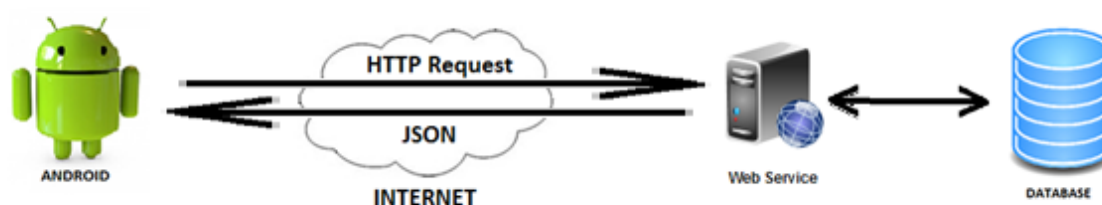
Tên thuộc tính	Kiểu dữ liệu	Null	Ý nghĩa
DetailID	int(11)	Không để trống	Mã chi tiết đơn hàng
OrderID	varchar(10)	Không để trống	Mã đơn hàng
ProductID	varchar(8)	Không để trống	Mã sản phẩm
Quantity	int(11)	Không để trống	Số lượng
Note	text	Có thể để trống	Ghi chú thêm cho sản phẩm

Bảng “products”

Tên thuộc tính	Kiểu dữ liệu	Null	Ý nghĩa
ProductID	varchar(8)	Không để trống	Mã sản phẩm
ProductName	varchar(255)	Không để trống	Tên sản phẩm
Detail	text	Có thể để trống	Thông tin sản phẩm
Price	int(11)	Không để trống	Giá tiền của sản phẩm
Avatar	text	Có thể để trống	Ảnh sản phẩm
CategoryID	varchar(8)	Không để trống	Mã danh mục sản phẩm

### 3. Web Service

Web Service thực hiện chức năng lấy dữ liệu trên Database



Hình 37. Mô hình Android kết nối với Web Service để lấy dữ liệu trên Database

Mô hình trên thể hiện việc ứng dụng gửi yêu cầu đến Web Service để Web Service kết nối và lấy dữ liệu từ Database trả về chuỗi JSON cho ứng dụng. Từ đó ứng dụng phân tích chuỗi JSON và trả ra kết quả cho người dùng.

Web Service trong ứng dụng này được viết bằng ngôn ngữ PHP và có tất cả 11 File PHP chính

Bảng Các File PHP chính

Tên File	Chức năng	Dữ liệu trả về
<b>config.php</b>	Khai báo thông tin về host database để kết nối.	Không có dữ liệu trả về.
<b>login.php</b>	Kết nối với Database và kiểm tra thông tin đăng nhập.	Trả về chuỗi JSON có thông tin của người dùng đã đăng nhập thành công.
<b>register.php</b>	Kết nối với Database và thực hiện thêm tài khoản của người dùng vào Database khi người dùng đăng ký.	Trả về chuỗi JSON thông báo trạng thái đăng ký.
<b>update_account.php</b>	Kết nối Database và cập nhật lại thông tin của người dùng khi người dùng sửa tài khoản.	Trả về chuỗi JSON thông báo trạng thái sửa tài khoản.
<b>changepass.php</b>	Kết nối Database và cập nhật lại mật khẩu mới khi người dùng thay đổi mật khẩu.	Trả về chuỗi JSON thông báo trạng thái thay đổi mật khẩu.
<b>gcm.html</b>	Giao diện để nhập nội dung tin nhắn Push, kết nối với gcm_engine.php để gửi tin.	Không có dữ liệu trả về.
<b>gcm_engine.php</b>	Sử dụng Google Cloud Messaging để gửi tin nhắn Push đến các thiết bị Android.	Không có dữ liệu trả về.
<b>getallproducts.php</b>	Kết nối Database và lấy thông tin sản phẩm trong Database.	Trả về chuỗi JSON hiển thị thông tin sản phẩm.
<b>orders.php</b>	Kết nối Database và thêm sản phẩm vào giỏ hàng.	Trả về chuỗi JSON thông báo trạng thái thêm sản phẩm vào giỏ hàng.
<b>get_orders.php</b>	Kết nối Database và lấy thông tin đơn hàng của người dùng đang đăng nhập.	Trả về chuỗi JSON hiển thị thông tin đơn hàng.



<b>get_category.php</b>	Kết nối Database và lấy thông tin của danh mục sản phẩm.	Trả về chuỗi JSON hiển thị các danh mục sản phẩm.
<b>get_orders_detail.php</b>	Kết nối Database và lấy thông tin những sản phẩm có trong giỏ hàng của người dùng đang đăng nhập.	Trả về chuỗi JSON hiển thị thông tin sản phẩm có trong giỏ hàng.
<b>getversion.php</b>	Kết nối Database và lấy ra phiên bản cao nhất của ứng dụng.	Trả ra chuỗi JSON hiển thị phiên bản cao nhất của ứng dụng và thời gian cập nhật.

#### 4. Android

Ứng dụng có tổng cộng 5 package:

##### 4.1. Package adapter

Chứa các class extends BaseAdapter.

Class	Hàm quan trọng	Kiểu	Ý nghĩa
<b>CartAdapter:</b> là Adapter của giỏ hàng, nơi xuất giao diện cho ListView trong class CartActivity.	<b>getView</b> (final int position, View convertView, ViewGroup parent)	View	Khởi tạo giao diện của từng sản phẩm trong giao diện giỏ hàng.
	<b>createOptionDialog</b> (String name, final String idForDel)	void	Tạo dialog để chỉnh sửa số lượng hay xóa một sản phẩm trong giỏ hàng.
	<b>createUpdatequantityDialog</b> (final String idForUpdate)	void	Tạo dialog để chỉnh sửa số lượng của 1 sản phẩm trong giỏ hàng
<b>DrawerListAdapter :</b> là Adapter của menu.	<b>getView</b> (int position, View convertView, ViewGroup parent)	View	Khởi tạo giao diện của từng mục trong menu chính.
<b>ProductsApdater:</b> là Adapter của danh sách hàng, nơi xuất giao diện danh sách	<b>getView</b> (final int position, View convertView, ViewGroup parent)	View	Khởi tạo giao diện hiển thị của từng dòng sản phẩm trong Trang chủ và các trang Category khác.

các món ăn trên trang chủ hay bất cứ fragment nào có danh sách hàng.	<b>setProductImage</b> (ImageView imgProduct, String avatarUrl)	void	Hiển thị ảnh của các sản phẩm trong list.
--	--	------	---

#### 4.2. Package fragments

Chứa các class extends Fragment.

Class	Hàm quan trọng	Kiểu	Ý nghĩa
<b>BaseFragment:</b> là giao diện chính của trang chủ và các danh sách hàng có thể chọn từ menu bên trái như Tổng hợp, Phần ăn combo, Gà rán...	<b>onCreateView</b> (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)	View	Khởi tạo giao diện danh sách các sản phẩm cho các Fragment kế thừa nó.
	<b>setProductsAdapter</b> (String search)	void	Thêm các sản phẩm vào ListView và hiển thị chúng ra giao diện người dùng.
	<b>showDialog()</b>	void	Tạo dialog để sắp xếp danh sách sản phẩm.
<b>FragmentAbout:</b> là giao diện của trang Giới thiệu chọn từ menu bên trái.	<b>onCreateView</b> (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)	View	Khởi tạo giao diện của trang Giới thiệu.

#### 4.3. Package helper

Chứa các class có các chức năng nhỏ lẻ để hỗ trợ cho các class chính.

Class	Hàm quan trọng	Kiểu	Ý nghĩa
<b>ConnectionCheckerHelper:</b> chứa phương thức dùng để kiểm tra xem người dùng có đang kết nối internet không và đưa ra thông báo.	<b>createInternetDialog()</b>	void	Tạo một dialog thông báo cho người dùng khi không có kết nối mạng.
	<b>isOnline()</b>	boolean	Kiểm tra xem người dùng có đang kết nối mạng hay không.
	<b>showDialog()</b>	void	Tạo dialog để sắp xếp danh sách sản phẩm.

<b>CovertHelper:</b> chứa các phương thức giúp định dạng hoặc xử lý những vấn đề có liên quan đến chuỗi.	<b>boDauChuo</b> i (String s)	String	Bỏ dấu tiếng Việt của chuỗi đưa vào.
	<b>boDauChar</b> (char ch)	char	Bỏ dấu tiếng Việt của 1 ký tự.
	<b>chuanHoaPrice</b> (String s)	String	Thêm đơn vị đồng và dấu phẩy ngăn cách phần ngàn cho giá tiền.
	<b>rutGon</b> (String s)	String	Rút gọn những tên sản phẩm nào quá dài để cho việc hiển thị đẹp hơn.
	<b>randomID</b> (int size)	String	Sinh ID ngẫu nhiên cho tài khoản mới đăng ký và mã đơn hàng mới đặt.
<b>DatabaseHelper:</b> tạo SQLite database trên máy và chứa các phương thức giúp tương tác với database.	<b>onCreate</b> (SQLiteDatabase db)	void	Khởi tạo các bảng trong Database.
	<b>themCategory</b> (CategoryItem ca)	void	Thêm Category vào bảng Category.
	<b>themProduct</b> (Products pd)	void	Thêm sản phẩm mới vào bảng Product.
	<b>taoCookie</b> (Cookie ck)	void	Thêm thông tin chi tiết của tài khoản đã đăng nhập thành công vào bảng Cookie.
	<b>themCart</b> (CartItem it)	void	Thêm sản phẩm mới vào giỏ hàng (lưu trữ trong bảng Orders).
	<b>getCartItem()</b>	ArrayList<CartItem>	Lấy danh sách dữ liệu của giỏ hàng chứa trong bảng Orders.
	<b>layCookie()</b>	Cookie	Lấy dữ liệu về thông tin chi tiết của tài khoản đã đăng nhập trong bảng Cookie.
	<b>getAllCategory()</b>	ArrayList<CategoryItem>	Lấy danh hết sách các Category chứa trong bảng Category.

	<b>xoaCookie()</b>	void	Xóa bảng Cookie.
	<b>xoaCategory()</b>	void	Xóa bảng Category.
	<b>xoaAllProducts()</b>	void	Xóa bảng Products.
	<b>xoaCart()</b>	void	Xóa bảng Orders.
	<b>deleteCart</b> (String id)	void	Xóa một sản phẩm trong giỏ hàng (bảng Orders)
	<b>getAllProducts</b> (String cot, boolean tangdan, String category, String search)	ArrayList<Products>	Lấy danh sách các sản phẩm trong bảng Products.
	<b>updateCartQuantity</b> (String id, int quantity)	void	Cập nhật lại số lượng của 1 sản phẩm trong giỏ hàng (bảng Orders).
<b>DetailDialogHelper:</b> tạo dialog trả về danh sách chi tiết của đơn hàng trong mục quản lý đơn hàng.	<b>createDetailDialog</b> (String id)	void	Khởi tạo dialog hiển thị chi tiết đơn hàng trong mục Quản lý đơn hàng.
	<b>getDetail</b>	AsyncTask<Void, Void, Void>	Khởi tạo AsyncTask để tải thông tin chi tiết đơn hàng từ host về và hiển thị lên dialog ở trên.
<b>JSONParser:</b> phương thức kết nối giữa android với webservice và trả về định dạng JSON từ webservice (nếu có).	<b>getJsonFromUrl</b> (String url, List<NameValuePair> params)	JSONObject	Kết nối với web service trên Host để gửi và nhận thông tin giữa Android và Host.
<b>PHPUrl:</b> chứa các url dẫn đến các file php trên Host.	Class này chỉ chứa các biến String để lưu trữ các url cho các class khác gọi tới sử dụng.		

<b>ProductDialog Helper:</b> giúp tạo dialog chi tiết món ăn khi người dùng click vào món ăn trên danh sách.	<b>createProductDialog</b> (final String pName, String pDetail, final String pAvatar, String pPrice, final String pID)	void	Khởi tạo dialog hiển thị chi tiết sản phẩm khi click vào trên Trang chủ cũng như các Category khác.
	<b>updatePrice</b> (int quantity)	void	Cập nhật giá tiền của sản phẩm khi có sự thay đổi về số lượng.
<b>SetProductListHelper:</b> định giá trị cho ListView ở các giao diện danh sách hàng.	<b>getProductsListItem()</b>	ArrayList<ProductsListItem>	Đặt giá trị cho mỗi dòng sản phẩm trên giao diện.

#### 4.4. Package *hhhai0304.dfc*:

Chứa các Activity của ứng dụng.

Class	Hàm quan trọng	Kiểu	Ý nghĩa
<b>CartActivity:</b> giao diện giỏ hàng.	<b>onCreate</b> (Bundle savedInstanceState)	void	Khởi tạo giao diện cho trang Giỏ hàng.
	<b>createOrderDialog</b> (final Cookie cookie)	void	Khởi tạo dialog để chọn thông tin giao hàng sau khi thanh toán đơn hàng.
	<b>addOrder</b>	AsyncTask<Void, Void, Void>	Khởi tạo AsyncTask để gửi thông tin đơn hàng đã xác nhận thanh toán lên Host.
<b>ChangePasswordActivity:</b> giao diện đổi mật khẩu.	<b>onCreate</b> (Bundle savedInstanceState)	void	Khởi tạo giao diện cho trang Đổi mật khẩu và kiểm tra xem mục Mật khẩu mới với Nhập lại Mật khẩu mới có trùng khớp nhau chưa.
	<b>changePassword</b>	AsyncTask<Void, Void, Void>	Khởi tạo AsyncTask để gửi thông tin cập nhật mật khẩu mới lên Host.

<b>EditActivity:</b> giao diện đổi thông tin người dùng.	<b>onCreate</b> (Bundle savedInstanceState)	void	Khởi tạo giao diện cho trang Chính sửa thông tin người dùng.
	<b>doEdit</b>	AsyncTask <Void, Void, Void>	Khởi tạo AsyncTask để gửi thông tin cập nhật chi tiết tài khoản của người dùng lên Host.
<b>GcmBroadcastReceiver:</b> đăng ký Broadcast Receiver cho ứng dụng.	<b>onReceive</b> (Context context, Intent intent)	void	Khởi tạo Wakeful Broadcast Receiver.
<b>GcmIntentService:</b> là service luôn chạy ngầm để bắt sự kiện khi có tin nhắn từ GCM tới.	<b>onHandleIntent</b> (Intent intent)	void	Bắt sự kiện khi có tin nhắn từ GCM tới.
	<b>generateNotification</b> (Context context, String message)	void	Hiển thị thông báo lên thanh thông báo của thiết bị khi có tin nhắn GCM.
<b>LoginActivity:</b> giao diện đăng nhập.	<b>onCreate</b> (Bundle savedInstanceState)	void	Khởi tạo giao diện cho trang Đăng nhập.
	<b>checkLogin</b>	AsyncTask <Void, Void, Void>	Khởi tạo AsyncTask để gửi thông tin Tài khoản và Mật khẩu vừa nhập lên Host để kiểm tra, nếu chính xác thì cho phép người dùng vào giao diện Trang chủ.
<b>MainActivity:</b> giao diện trang chủ và menu bên trái.	<b>onCreate</b> (Bundle savedInstanceState)	void	Khởi tạo giao diện cho Trang chủ và khởi tạo Menu chính (bên trái).
	<b>onCreateOptionsMenu</b> (Menu menu)	boolean	Khởi tạo nút Giỏ hàng trên thanh ActionBar.
	<b>selectItemFromDrawer</b> (final int position)	void	Đặt sự kiện khi người dùng chọn trong Menu chính.
	<b>KhaiBao()</b>	void	Khai báo các View cũng như khởi tạo các giá trị cho Trang chủ và Menu.

	<b>createAccountDialog()</b>	void	Khởi tạo dialog hiển thị thông tin chi tiết của Tài khoản đã đăng nhập cũng như các chức năng liên quan đến tài khoản như Đổi mật khẩu, Đăng xuất, Chỉnh sửa thông tin.
<b>OrderManagerActivity:</b> giao diện quản lý đơn hàng.	<b>onCreate</b> (Bundle savedInstanceState)	void	Khởi tạo giao diện cho trang Quản lý đơn hàng.
	<b>getOrders</b>	AsyncTask <Void, Void, Void>	Khởi tạo AsyncTask để tải về các đơn hàng thuộc sở hữu của tài khoản đã đăng nhập.
	<b>addOrderView</b> (LinearLayout llView, final String OrderID, String OrderDate, final String Address, final String Phone, int Price)	void	Tạo giao diện của 1 đơn hàng trong danh sách các đơn hàng.
<b>RegisterActivity:</b> giao diện đăng ký tài khoản.	<b>onCreate</b> (Bundle savedInstanceState)	void	Khởi tạo giao diện cho trang Đăng ký.
	<b>doRegister</b>	AsyncTask <Void, Void, Void>	Khởi tạo AsyncTask để gửi thông tin đăng ký lên Host.
<b>SplashScreenActivity:</b> giao diện màn hình khởi động của ứng dụng. Tại đây sẽ tiến hành tải các dữ liệu và hình ảnh về các món ăn, hàng.	<b>onCreate</b> (Bundle savedInstanceState)	void	Khởi tạo giao diện cho Màn hình khởi động.
	<b>loadData</b>	AsyncTask <Void, Void, Void>	Tải về thông tin của Phiên bản hiện tại trong Host rồi so sánh với Phiên bản hiện tại trong máy, nếu trùng khớp thì cho người dùng thao tác tiếp, nếu không thì bắt đầu tải thông tin của tất cả các sản phẩm từ trên Host về và lưu trữ trong SQLite Database trên máy Android.

	<b>downloadProductAvatar</b> (String url, String fname)	void	Tải về tất cả hình ảnh của các sản phẩm về lưu trữ trong bộ nhớ máy Android.
	<b>setVersion</b> (String version)	void	Đặt lại thông tin phiên bản trên máy sau khi đã cập nhật bản mới.
	<b>getVersion()</b>	String	Lấy phiên bản hiện tại từ file text lưu trong máy.

#### 4.5. Package item

Chứa cái class item làm khuôn mẫu cho các adapter

Class	Ý nghĩa
<b>CartItem</b>	Khung của các sản phẩm giỏ hàng.
<b>Cookie</b>	Khung của thông tin user đã đăng nhập.
<b>NavItem</b>	Khung của menu bên trái.
<b>Products</b>	Khung của sản phẩm.
<b>ProductsListItem</b>	Khung của món ăn trong ListView trên các giao diện danh sách món ăn.
<b>OrderManagerItem</b>	Khung của các đơn hàng trong trang Quản lý đơn hàng.

Ngoài ra, ứng dụng còn sử dụng thư viện Android Universal Image Loader để hỗ trợ hiển thị ảnh.

- Trang chủ: <https://github.com/nostra13/Android-Universal-Image-Loader>
- Mô tả: đây là một thư viện hỗ trợ hiển thị ảnh rất hay, nó hỗ trợ được nhiều nguồn ảnh và có cơ chế tải ảnh không đồng bộ (kéo màn hình đến đâu thì tải ảnh đến đó chứ không một lần tải hết tất cả các ảnh).

```
"http://site.com/image.png" // from Web
"file:///mnt/sdcard/image.png" // from SD card
"file:///mnt/sdcard/video.mp4" // from SD card (video thumbnail)
"content://media/external/images/media/13" // from content provider
"content://media/external/video/media/13" // from content provider (video thumbnail)
"assets://image.png" // from assets
"drawable://" + R.drawable.img // from drawables (non-9patch images)
```

Hình 38. Các định dạng mà thư viện hỗ trợ



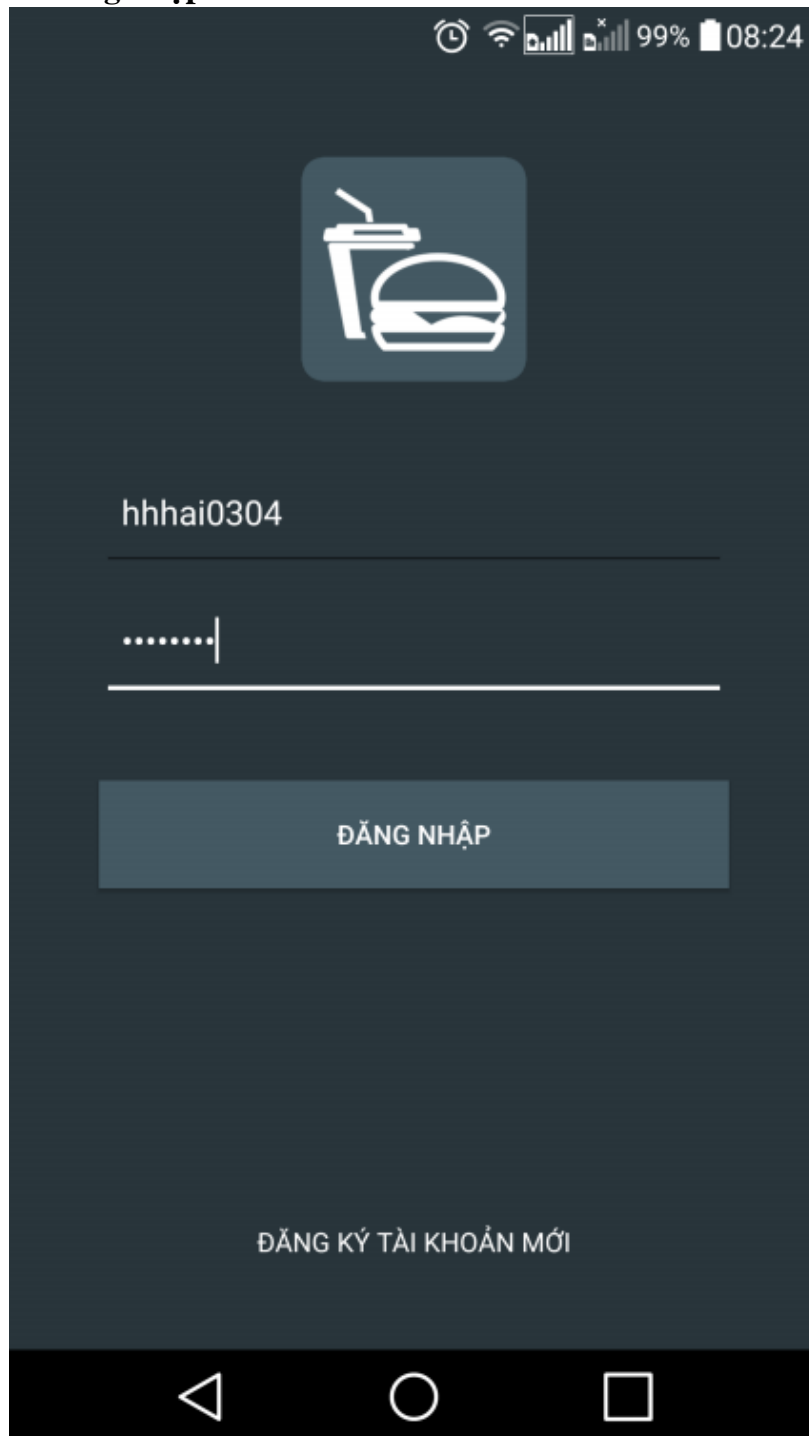
## CHƯƠNG V: THIẾT KẾ GIAO DIỆN VÀ CHỨC NĂNG

### 1. Màn hình khởi động



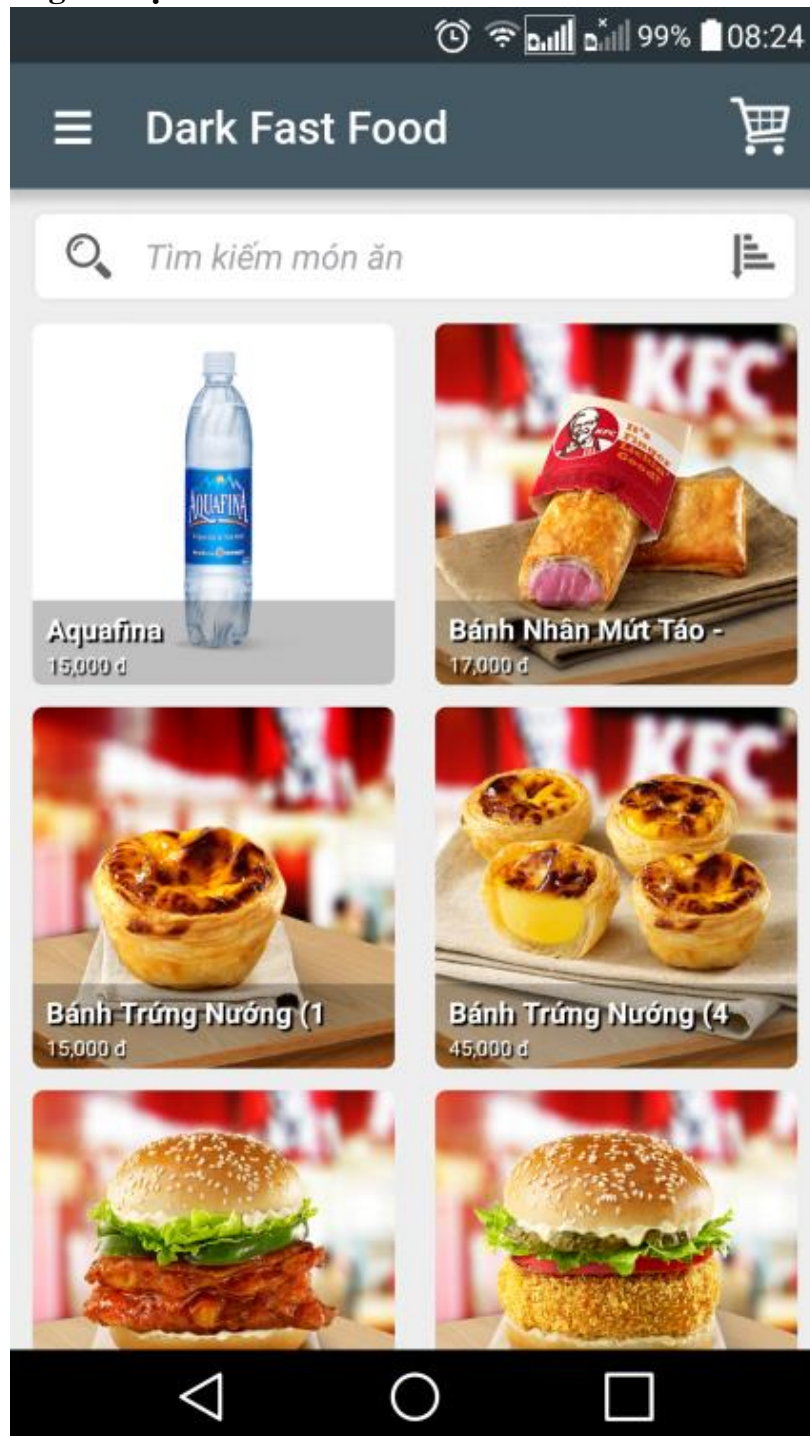
Hình 39. Màn hình khởi động

## 2. Màn hình đăng nhập



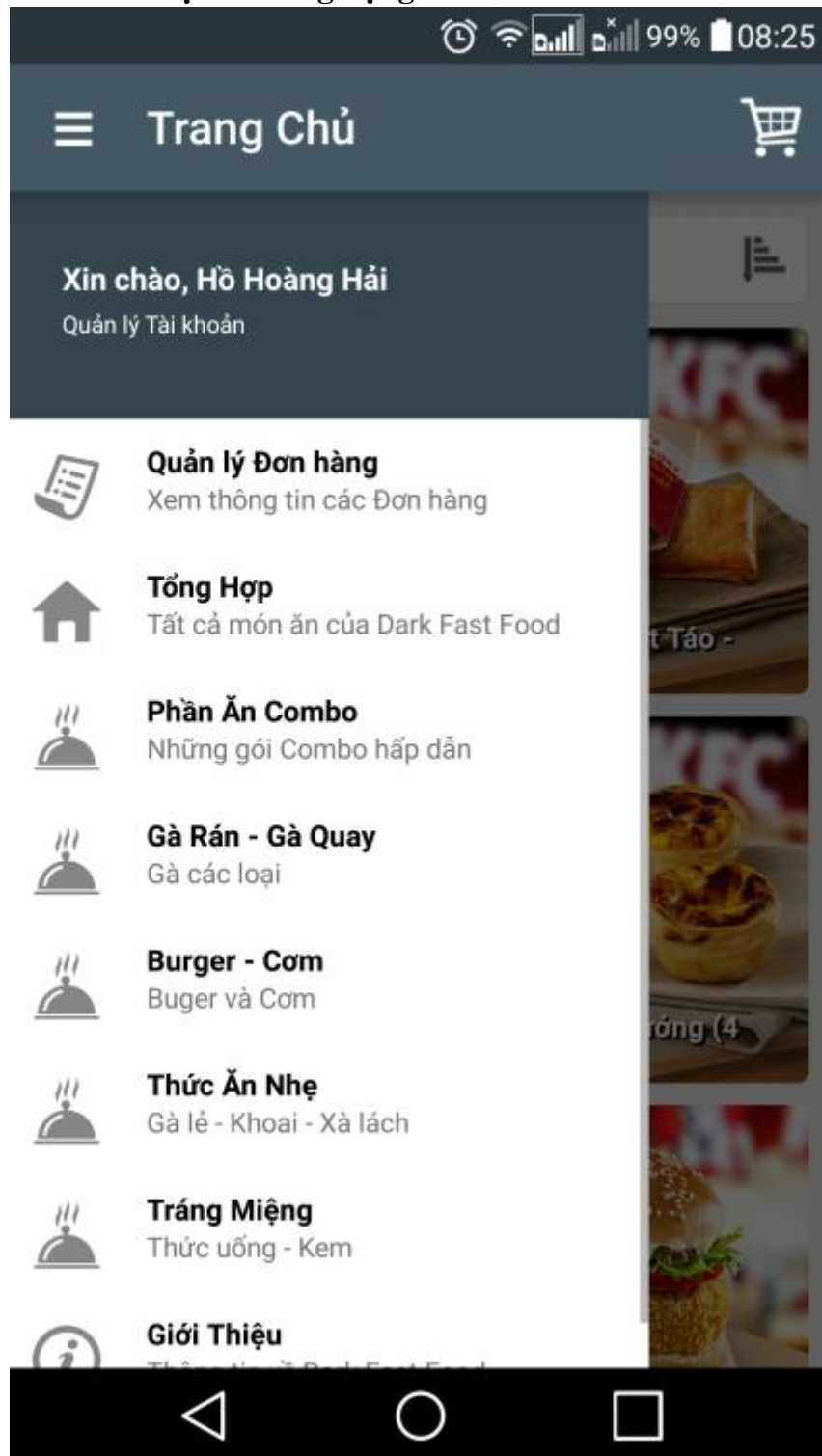
Hình 40. Giao diện đăng nhập của ứng dụng

### 3. Màn hình giao diện món ăn



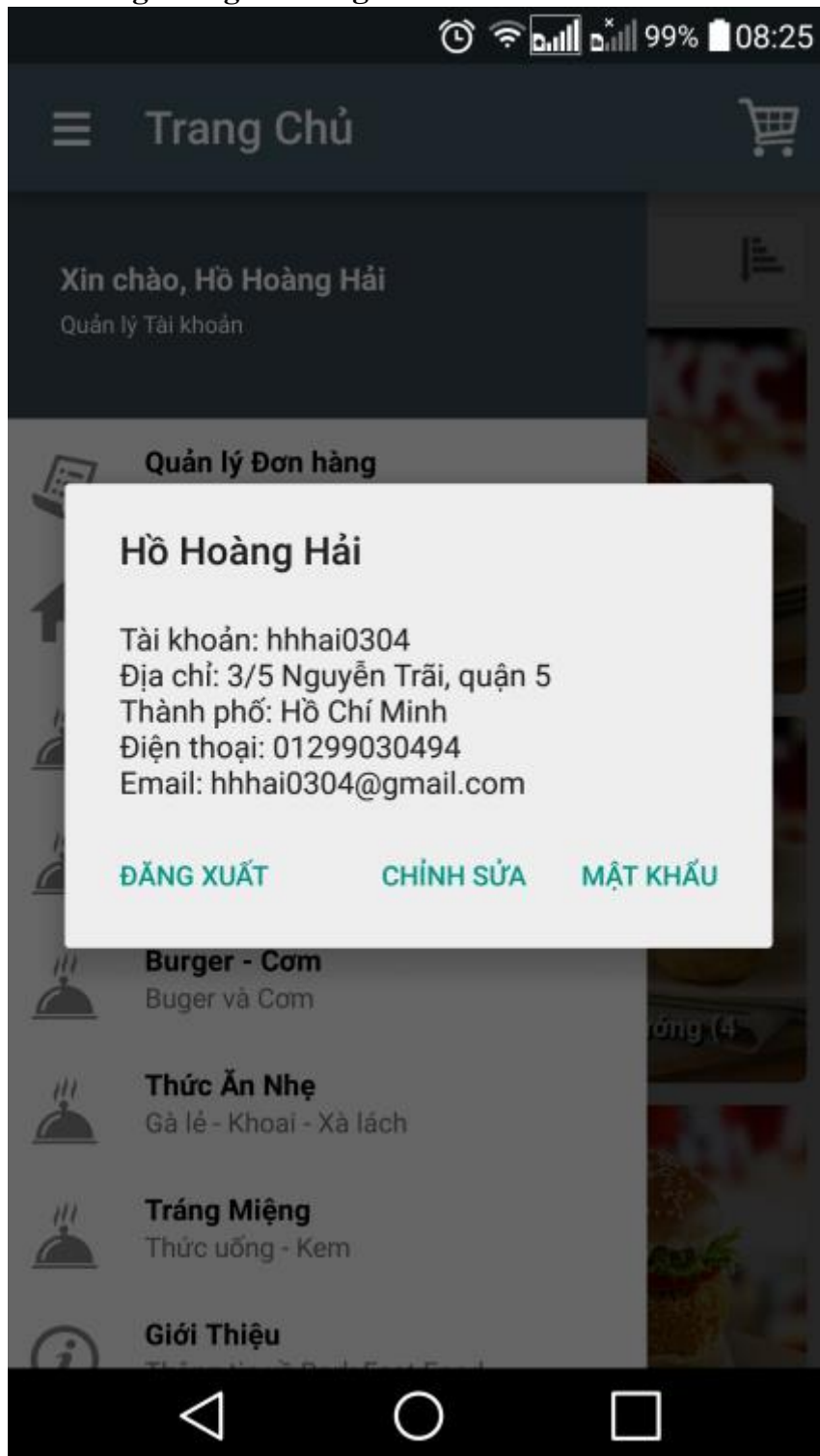
Hình 41. Giao diện món ăn của ứng dụng

#### 4. Màn hình danh mục của ứng dụng



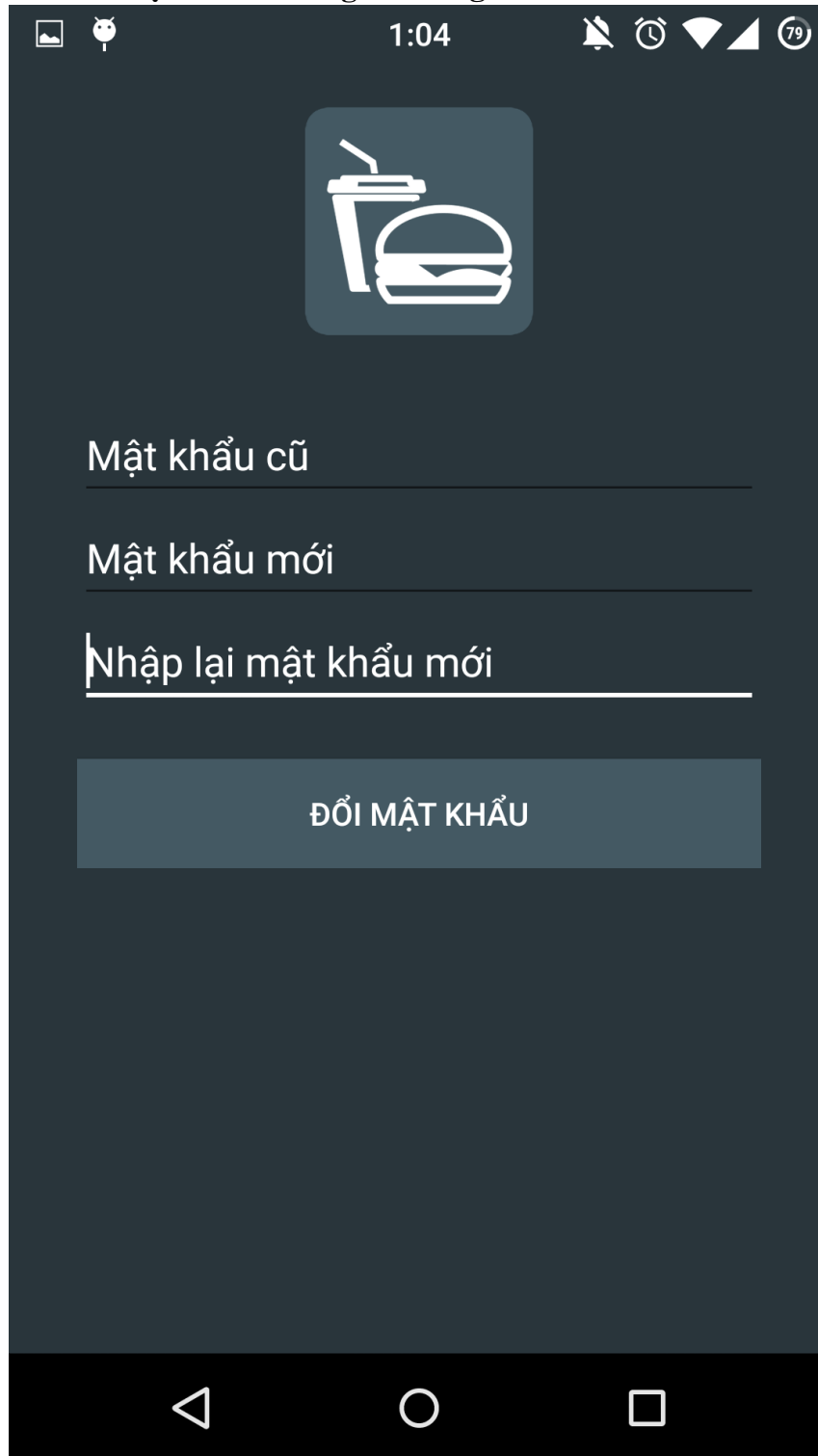
Hình 42. Phần danh mục của ứng dụng gồm có “Danh mục món ăn” và “Tên của User” đăng nhập

## 5. Màn hình thông tin người dùng



Hình 43. Thông tin của người dùng

## 6. Màn hình đổi mật khẩu của người dùng



Mật khẩu cũ

Mật khẩu mới

Nhập lại mật khẩu mới

ĐỔI MẬT KHẨU

Hình 44. Đổi mật khẩu của người dùng

## 7. Màn hình đổi thông tin người dùng

Tên      Họ

Địa chỉ

Quận      Hồ Chí Minh ▼

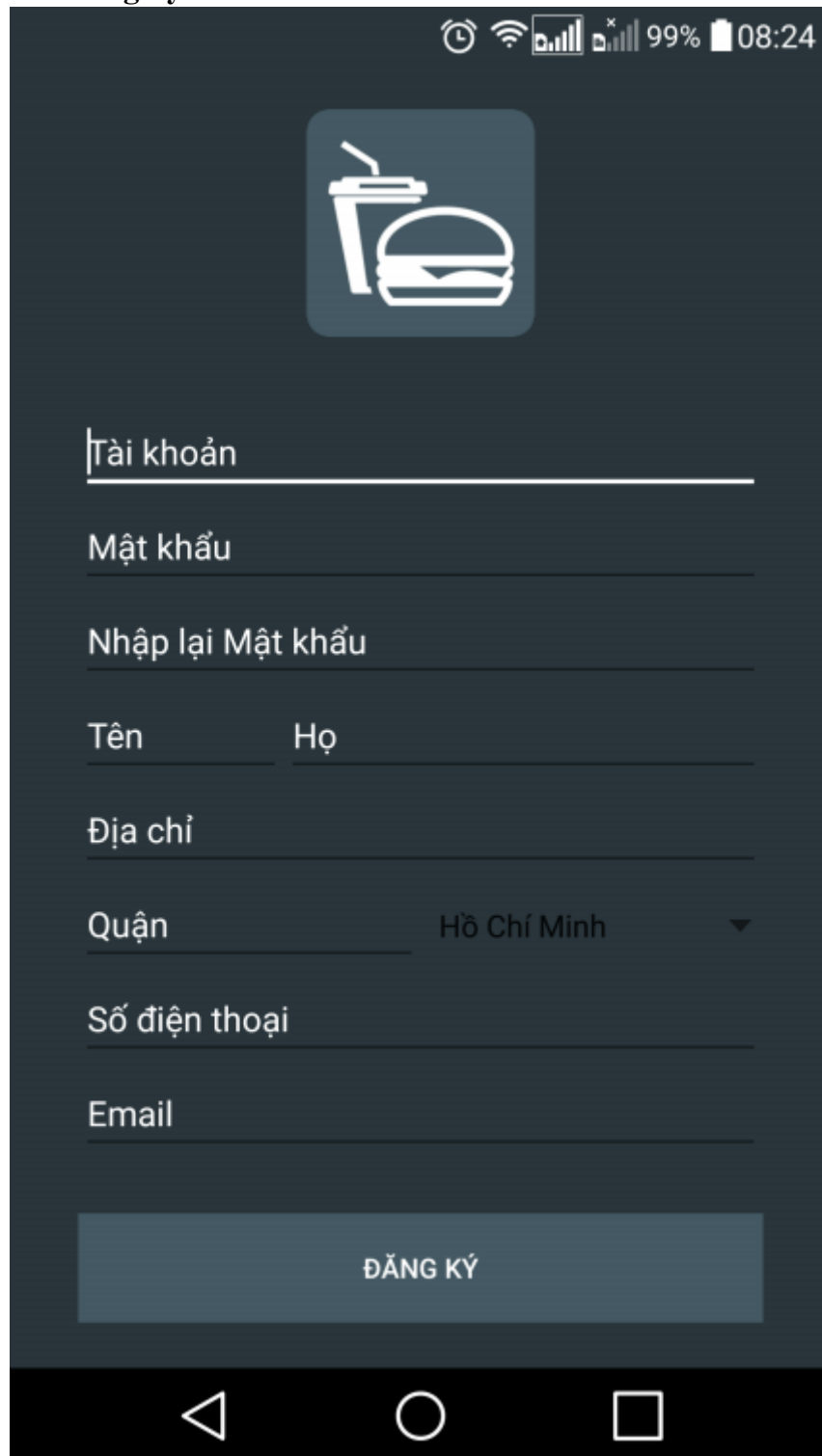
Số điện thoại

Email

CHỈNH SỬA THÔNG TIN

Hình 45. Đổi thông tin người dùng

## 8. Màn hình đăng ký tài khoản mới

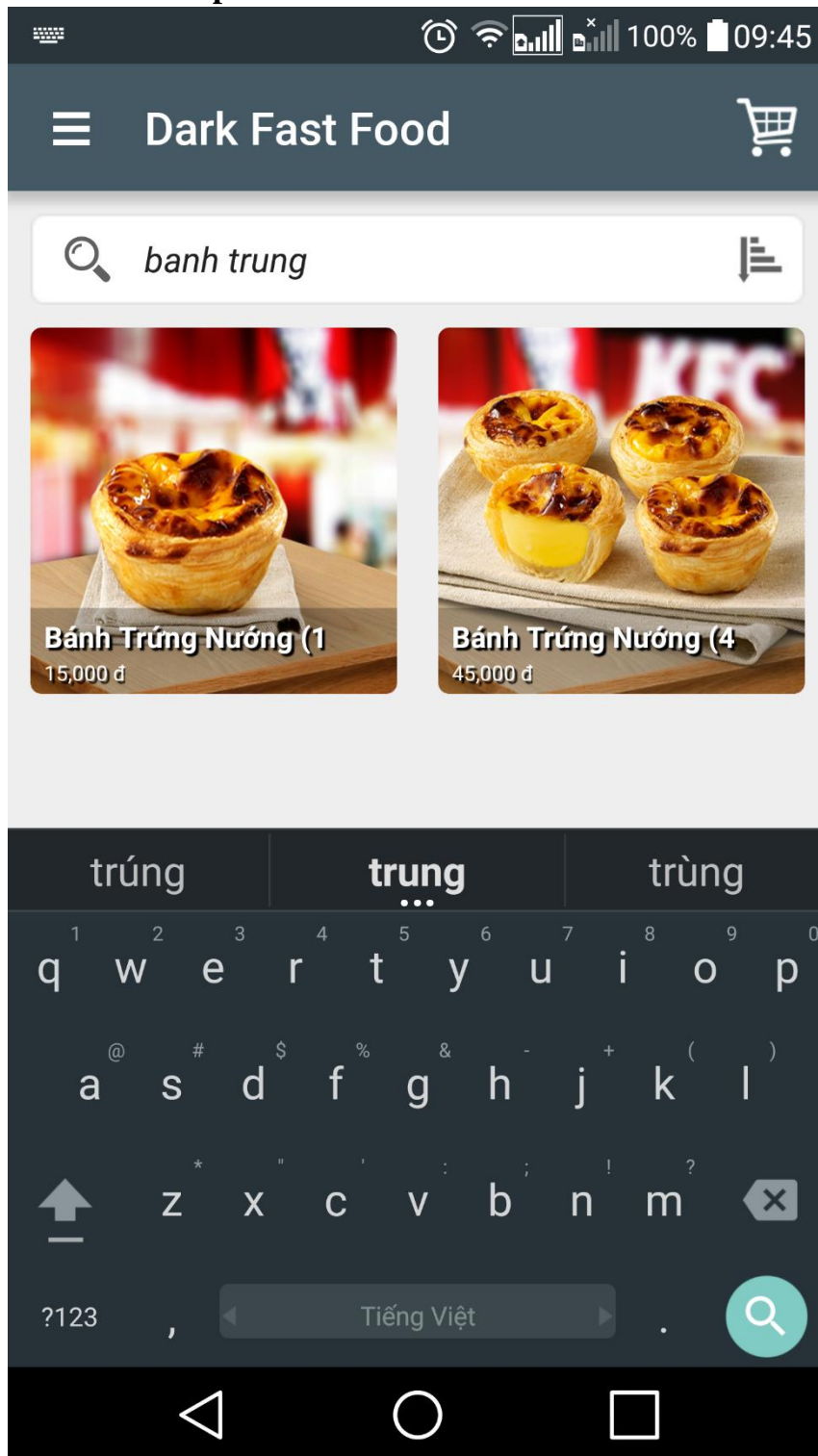


The screenshot shows a mobile application interface for creating a new account. At the top, there is a status bar with icons for alarm, Wi-Fi, cellular signal, and battery level at 99%, along with the time 08:24. Below the status bar is a header section featuring a logo of a cup of coffee and a burger. The main content area consists of several input fields for registration information: 'Tài khoản' (Username), 'Mật khẩu' (Password), 'Nhập lại Mật khẩu' (Repeat Password), 'Tên' (First Name) and 'Họ' (Last Name), 'Địa chỉ' (Address), 'Quận' (District) with a dropdown menu currently showing 'Hồ Chí Minh', 'Số điện thoại' (Phone Number), and 'Email'. At the bottom of the form is a large button labeled 'ĐĂNG KÝ' (Register). The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Hình 46. Đăng ký tài khoản mới

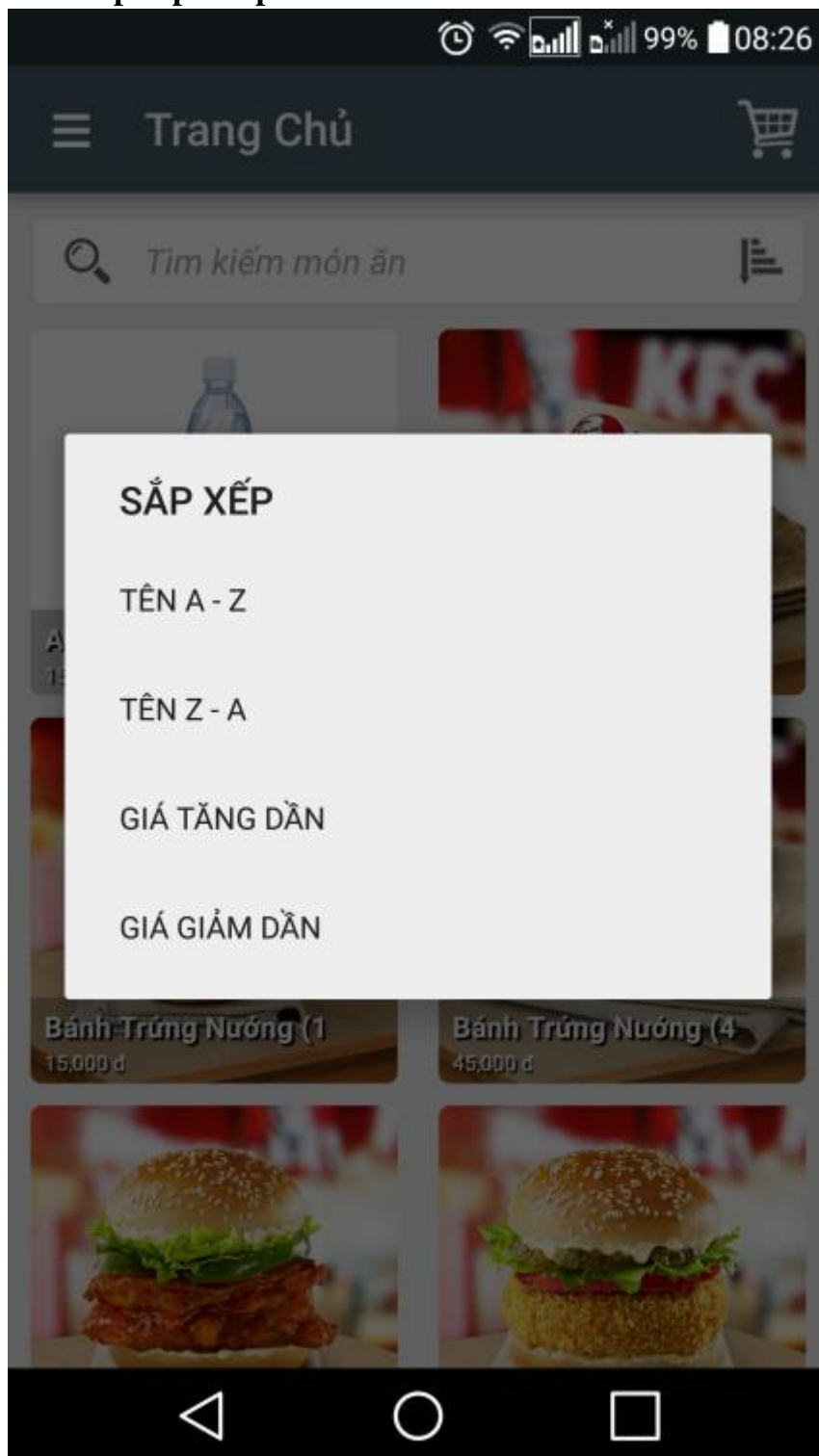


## 9. Màn hình tìm kiếm sản phẩm



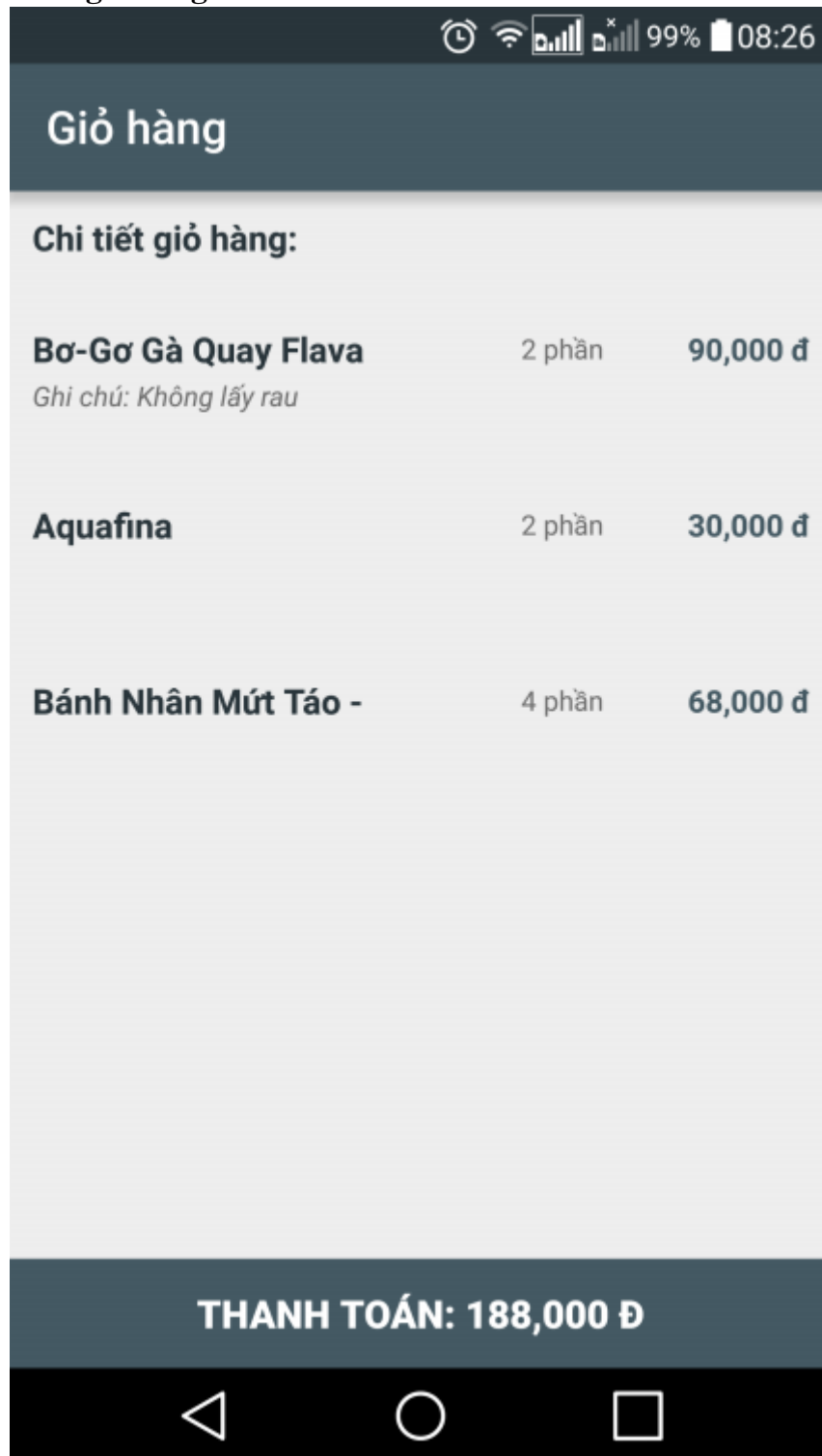
Hình 47. Tìm kiếm sản phẩm

## 10. Màn hình sắp xếp sản phẩm

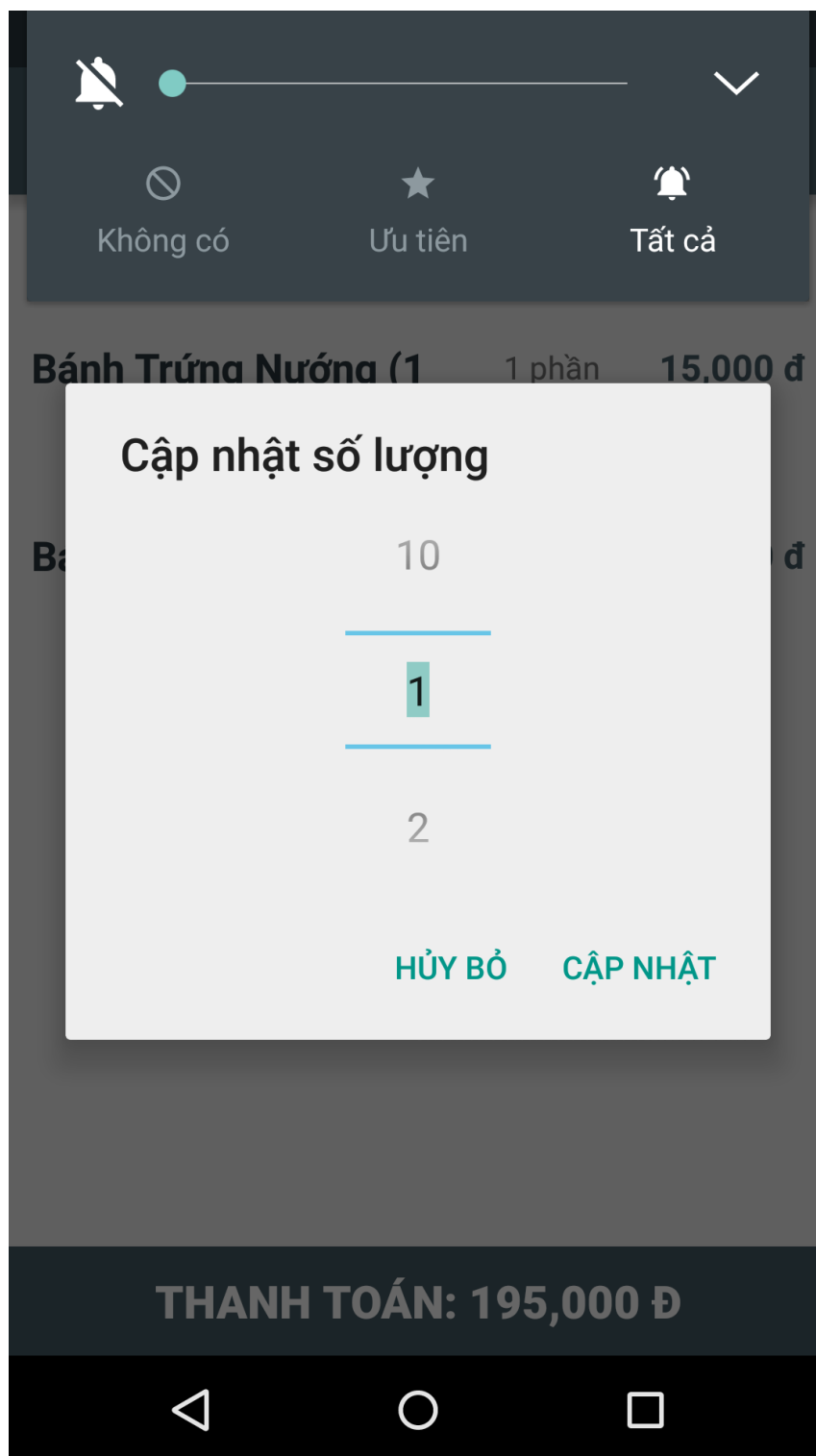


Hình 48. Sắp xếp sản phẩm

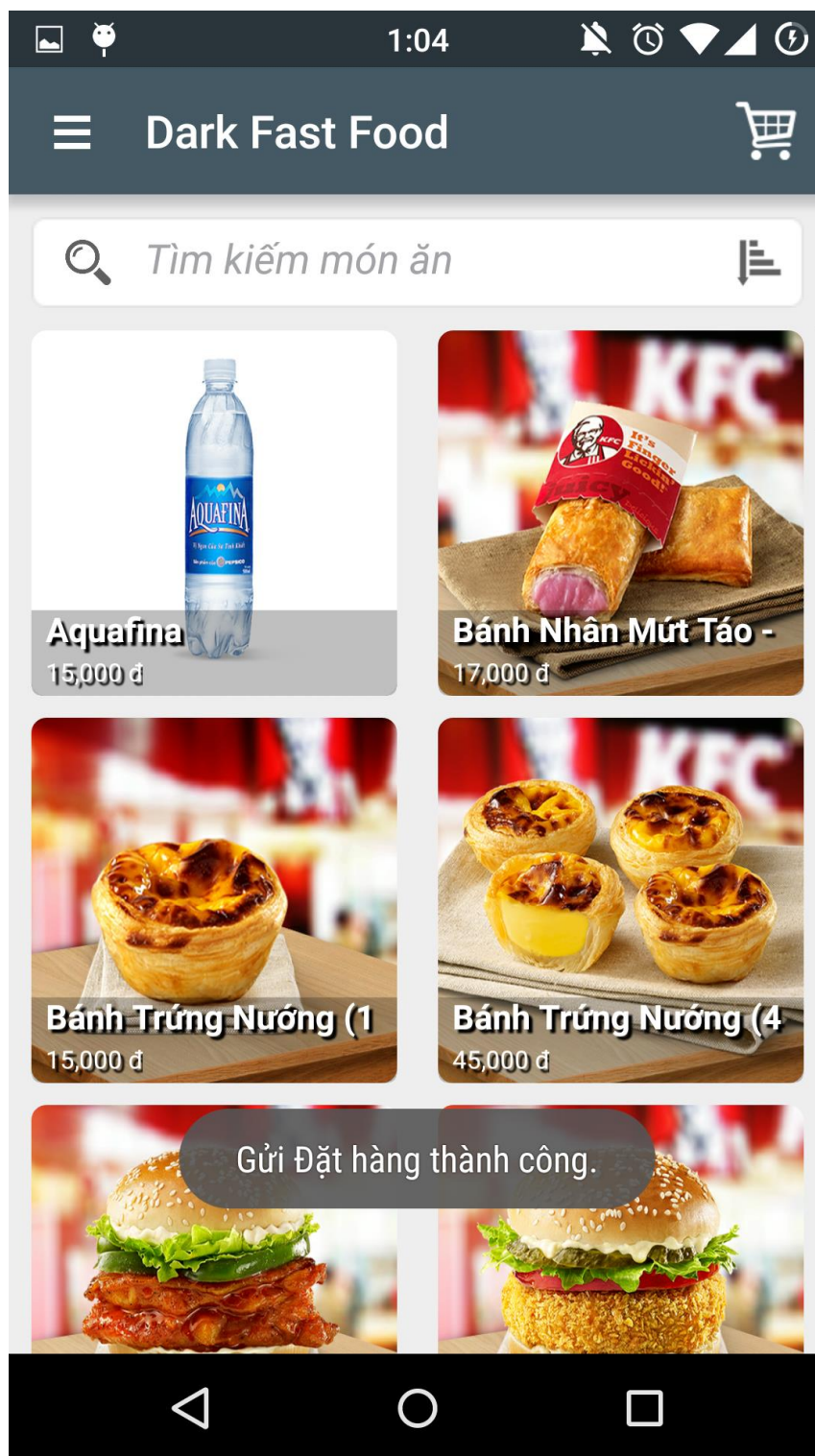
## 11. Màn hình giỏ hàng



Hình 49. Thông tin giỏ hàng

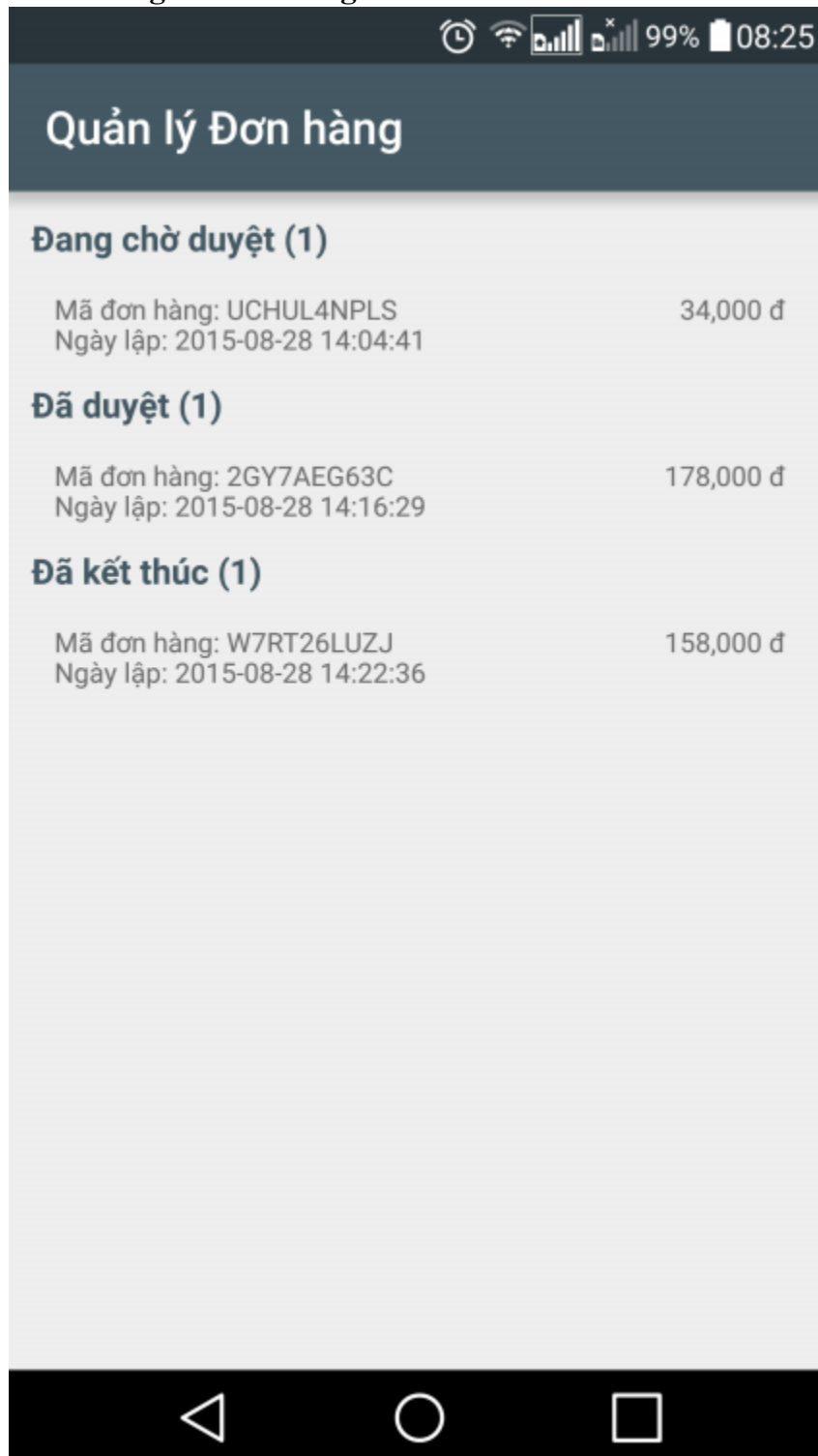


Hình 50. Thay đổi số lượng sản phẩm cần mua

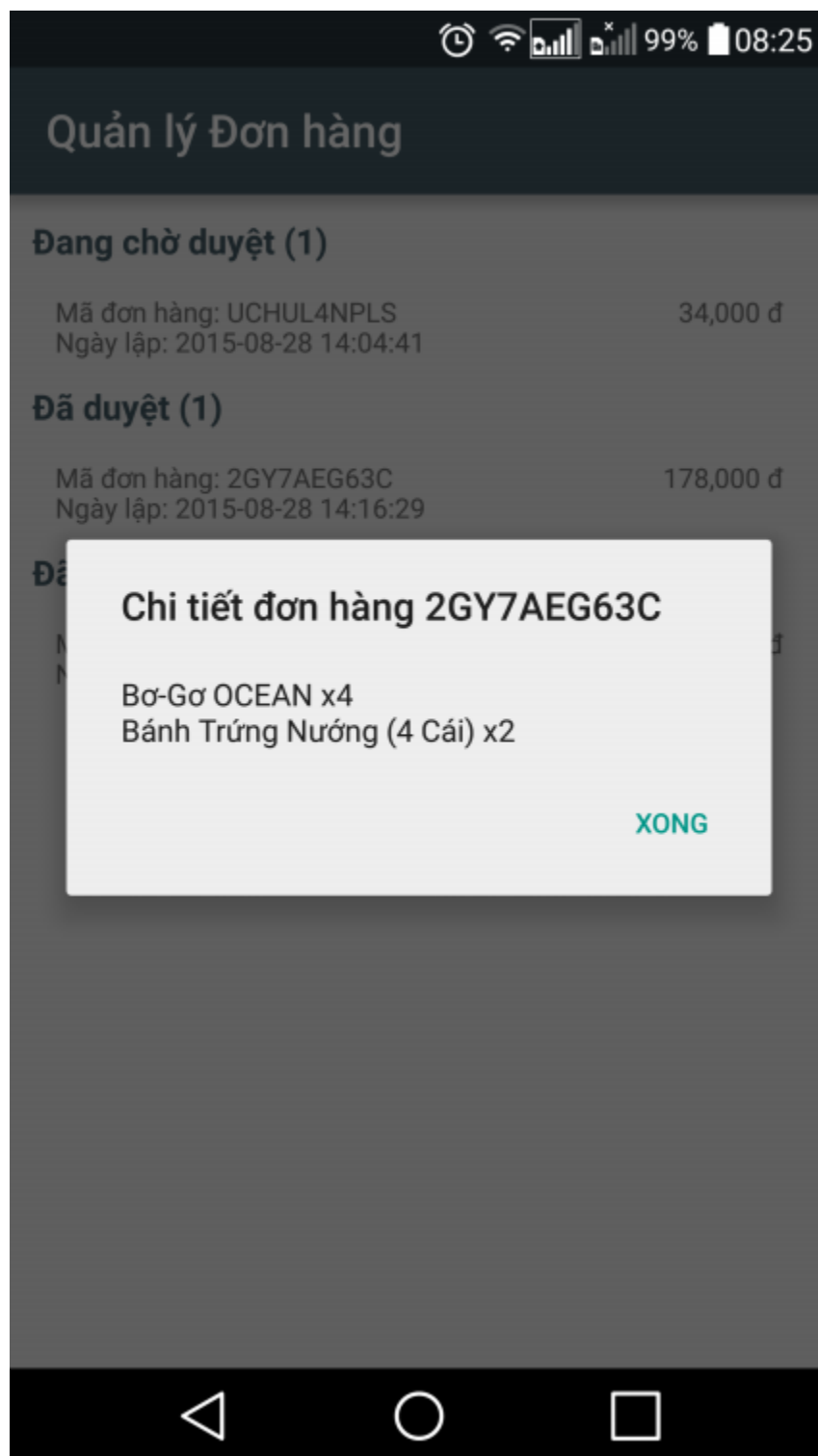


Hình 51. Sau khi thanh toán sản phẩm thì sẽ trở lại giao diện sản phẩm kèm thông báo đã đặt hàng thành công

## 12. Màn hình thông tin đơn hàng

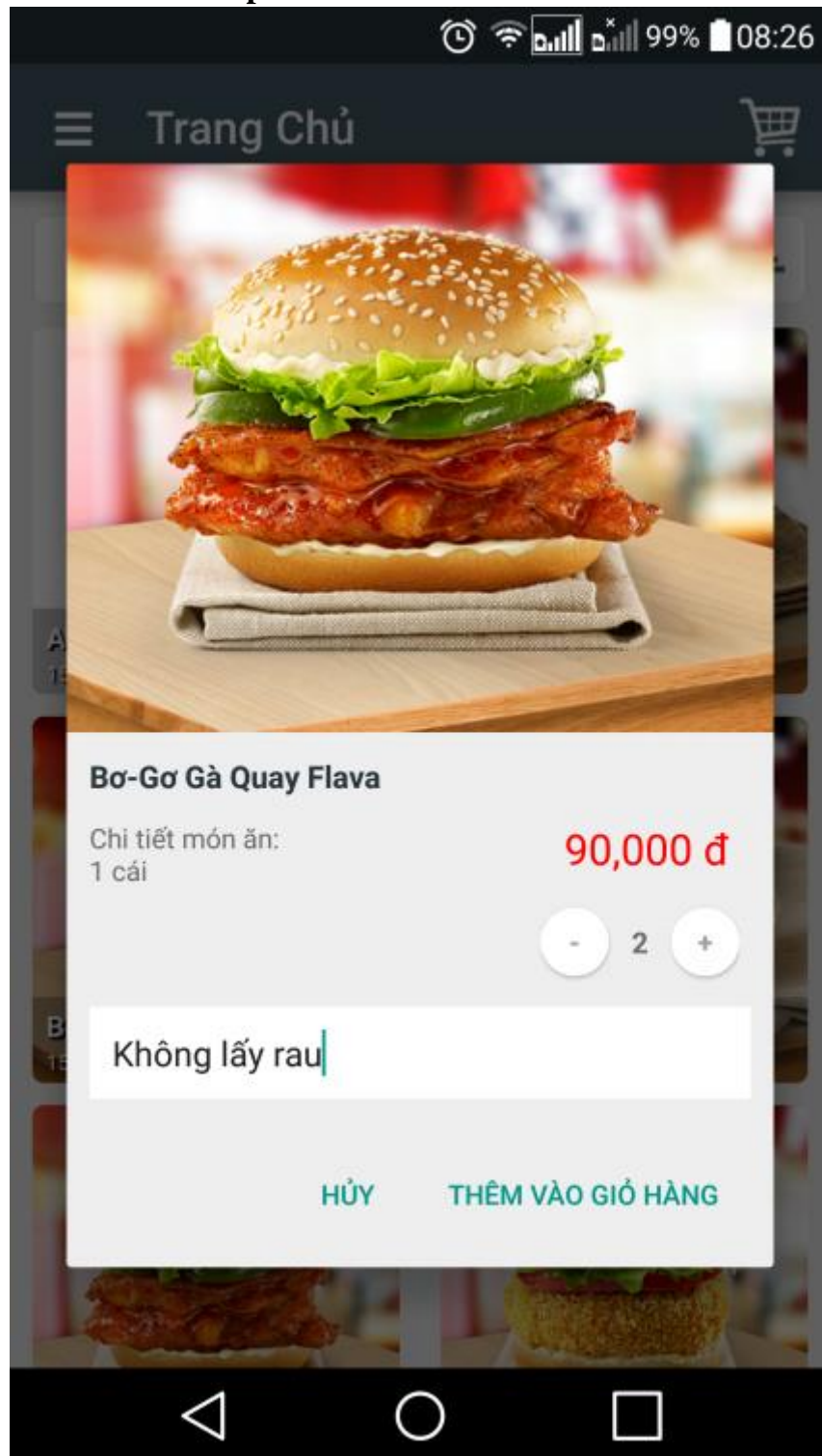


Hình 52. Thông tin đơn hàng của khách hàng



Hình 53. Chi tiết đơn hàng

### 13. Màn hình chi tiết sản phẩm



Hình 54. Chi tiết sản phẩm



#### 14. Màn hình thông tin nhận hàng

The screenshot shows a mobile application interface for a shopping cart. At the top, the status bar displays the time 08:26, 99% battery, and various connectivity icons. The app's header is dark with the text "Giỏ hàng" (Shopping Cart). Below the header, the section "Chi tiết giỏ hàng:" (Cart details:) lists items. One item, "Bơ-Gơ Gà Quay Flava", is shown with a quantity of "2 phần" and a price of "90,000 đ". A modal dialog is open in the center, titled "Địa chỉ giao hàng" (Delivery address). It contains two radio buttons: the first is for "3/5 Nguyễn Trãi, Quận 5, Hồ Chí Minh" and the second, which is selected, is for "Địa chỉ khác" (Different address). Below the radio buttons, the section "Thông tin giao hàng:" (Delivery information:) contains two input fields: "Địa chỉ" (Address) and "Số điện thoại" (Phone number). At the bottom of the modal are two buttons: "HỦY" (Cancel) and "XÁC NHẬN" (Confirm). The background of the app shows a total amount "THANH TOÁN: 188,000 Đ" at the bottom.

**Giỏ hàng**

**Chi tiết giỏ hàng:**

<b>Bơ-Gơ Gà Quay Flava</b>	2 phần	90,000 đ
----------------------------	--------	----------

**Địa chỉ giao hàng**

☐ 3/5 Nguyễn Trãi, Quận 5, Hồ Chí Minh

☒ Địa chỉ khác

**Thông tin giao hàng:**

Địa chỉ

Số điện thoại

HỦY XÁC NHẬN

**THANH TOÁN: 188,000 Đ**

Hình 55. Thông tin nhận hàng

## CHƯƠNG VI: KẾT LUẬN

### 1. Kết quả đạt được

- Đề tài này đã làm rõ được khái niệm web service, nền tảng, kiến trúc và xây dựng được mô hình ứng dụng của web service. Từ đó, đưa ra được những đặc điểm, đặc trưng và chức năng điển hình của web service.
- Đề tài đã tổng hợp và làm rõ những định nghĩa, khái niệm, bản chất, đặc trưng, nền tảng và ứng dụng của Thương mại điện tử. Tìm hiểu sự liên quan giữa web service và thương mại điện tử, nêu lên được những ứng dụng của web service trong thương mại điện tử.
- Tổng hợp thành một tài liệu chuẩn về những khái niệm, đặc điểm, chức năng, nền tảng công nghệ, ứng dụng và sự liên hệ của web service trong thương mại điện tử. Từ đó, ứng dụng vào thực tế là xây dựng một bản demo đơn giản về web service.
- Ứng dụng chạy ổn định, hoàn thiện đầy đủ các chức năng như đã phân tích trong nội dung thực hiện.
- Giao diện thân thiện, dễ sử dụng.
- Ứng dụng mang tới cho người dùng một sự trải nghiệm mới, vừa là nơi người dùng có thể xem tất cả những ghi chú trong điện thoại của mình, vừa là nơi để người dùng có thể chia sẻ những ghi chú với người yêu
- Ứng dụng tích hợp nhiều công nghệ mới góp phần làm thay đổi và mang lại sự tiện lợi cho người dùng.
- Hoàn thành đồ án đúng hạn. Hoàn tất việc khảo sát hiện trạng, phân tích, thiết kế và cài đặt phần mềm.
- Sử dụng ở mức căn bản các công cụ quản lý code bằng Github với tool SmartGit.
- Nhóm đoàn kết hoàn thành tốt các công việc.

### 2. Khó khăn và hạn chế

#### - Khó khăn:

- Trong quá trình thực hiện đề tài, khó khăn lớn nhất là thiết kế cơ sở dữ liệu để thực hiện đầy đủ các chức năng như đã phân tích trong nội dung thực hiện, cụ thể là chức năng đồng bộ dữ liệu giữa hai User.
- Xây dựng ứng dụng tách biệt Backend và FrontEnd nên mất khá nhiều thời gian để triển khai.

#### - Hạn chế:

- Vì thời gian có hạn nên ứng dụng chỉ hoàn thiện được các chức năng ở mức chưa chuyên sâu, chỉ có thể áp dụng trong phạm vi vừa và trung bình.
- Chưa tích hợp real-time nên còn hạn chế, cụ thể khi hai tài khoản đồng bộ tài khoản khi cập nhập thông tin nội dung nhưng mà tài khoản đồng bộ còn lại không update lại ngay.

### 3. Hướng phát triển

- Tích hợp Google Maps vào sản phẩm: để bên bán hàng có thể biết vị trí của người dùng ở đâu để giao hàng cũng như đưa ra các chiến lược kinh doanh phù hợp với từng khu vực.
- Làm thêm các phần khuyến mãi cho sản phẩm: để bên bán hàng có thể đưa ra các chiến lược khuyến mãi tiếp cận người dùng cũng như người dùng có được các khuyến mãi để lựa chọn mua hàng.
- Thêm bình luận, đánh giá cho sản phẩm: để người dùng có cái nhìn khách quan hơn về các sản phẩm dựa trên các bình luận, đánh giá.
- Tạo các mã ưu đãi cho khách hàng theo dạng QR Code: QR Code là điểm mạnh của các thiết bị di động hay máy tính bảng, có các mã ưu đãi QR Code sẽ làm khách hàng muốn sử dụng ứng dụng này hơn.

### 4. Tài liệu tham khảo

- [1] Đặng Văn Đức (2002). *Phân tích thiết kế hướng đối tượng bằng UML*. Nhà xuất bản giáo dục.
- [2] Refsnes Data (1998). *PHP 5 Tutorial*. A Norwegian software development and consulting company.
- [3] Huỳnh Lê Trung Toàn (2013). *Học PHP cơ bản*.  
<<http://2school.vn/hoc-php-co-ban-truc-tuyen.html>>
- [4] User1451549 (2012). *I can not get registration ID from Android GCM*.  
<<http://stackoverflow.com/questions/11713363/i-can-not-get-registration-id-from-android-gcm>>
- [5] Android Example. *Android Push Notifications using GCM*.  
<[http://androidexample.com/Android\\_Push\\_Notifications\\_using\\_Google\\_Cloud\\_Messaging\\_GCM/index.php?view=article\\_discription&aid=119&aaid=139](http://androidexample.com/Android_Push_Notifications_using_Google_Cloud_Messaging_GCM/index.php?view=article_discription&aid=119&aaid=139)>
- [6] Patflynn (2015). *"App Engine Backend with Google Cloud Messaging" Template*.  
<<https://github.com/GoogleCloudPlatform/gradle-appengine-templates/tree/master/GcmEndpoints>>
- [7] Google (2015). *Cloud Messaging*.  
<<https://developers.google.com/cloud-messaging/#manifest>>

**HẾT**

---