

**ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

Tel. (84-511) 3736 949, Fax. (84-511) 842 771

Website: [itf.ud.edu.vn](http://itf.ud.edu.vn), E-mail: [cntt@ud.edu.vn](mailto:cntt@ud.edu.vn)



**LUẬN VĂN TỐT NGHIỆP KỸ SƯ**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**MÃ NGÀNH : 05115**

**ĐỀ TÀI:  
XÂY DỰNG ỨNG DỤNG FASTFOOD TRÊN  
NỀN ANDROID**

**Mã số:**

**Ngày bảo vệ:**

**CBHD**

**SVTH**

**LỚP**

**: Ts. Nguyễn Văn Hiệu**

**: Trương Công Khôi**

**: 11TLT-A**

**ĐÀ NẴNG, 12/2013**

# MỤC LỤC

---

# MỞ ĐẦU

## **.I BỐI CẢNH CHỌN ĐỀ TÀI**

Công việc kinh doanh buôn bán đã xuất hiện từ rất lâu, trải qua mỗi giai đoạn lịch sử nó mang một hình thức đặc thù riêng. Trước kia, khi các công cụ hỗ trợ công việc mua bán chưa phát triển mạnh, thì người kinh doanh mua bán chỉ diễn ra dưới hình thức mua bán trực tiếp. Từ khi khoa học công nghệ phát triển, nó đã tạo ra phát triển các loại hình thức mua bán mới, điển hình là mua bán trực tuyến. Hình thức mua bán trực tuyến hỗ trợ đắc lực cho người kinh doanh tiếp xúc với được nhiều khách hàng, còn khách hàng thì công việc mua bán được diễn ra nhanh chóng, thuận lợi, tiết kiệm được thời gian ...

- Hiện nay, Thương Mại Điện Tử (TMĐT) đang được nhiều quốc gia quan tâm, coi là một trong những động lực phát triển chủ yếu của nền kinh tế. TMĐT đem lại những lợi ích tiềm tàng, giúp doanh nghiệp thu được thông tin phong phú về thị trường và đối tác, giảm chi phí tiếp thị và giao dịch, tạo dựng và củng cố quan hệ bạn hàng.

- Việc mua hàng qua mạng chỉ với thủ tục đăng ký mua sắm đơn giản nhưng đem lại nhiều lợi ích: tiết kiệm và chủ động về thời gian, tránh khỏi những phiền phức khó chịu. Vậy nên việc mua bán hàng qua mạng đang rất được mọi người quan tâm.

Trên cơ sở các kiến thức được học trong nhà trường và quá trình tìm hiểu các website trong thực tế, em đã quyết định chọn đề tài “*Xây dựng ứng dụng mua bán fastfood trên nền Android*”.

- Phục vụ tốt hơn nhu cầu của khách hàng và quản lý của nhà hàng trong hoạt động kinh doanh.
- Khách hàng chỉ cần các thao tác đơn giản trên điện thoại di động là có thể đặt hàng được ngay.

## **.II MỤC ĐÍCH VÀ Ý NGHĨA CỦA ĐỀ TÀI**

---

## **II.1 Mục đích**

Ứng dụng mua bán fastfood chạy trên nền Android được thực hiện dựa trên hai mục đích chính sau :

Thứ nhất là nghiên cứu các công nghệ lập trình như : lập trình ứng dụng với Android.

Thứ hai là phát triển ứng dụng có tính thực tiễn cao, có khả năng triển khai ứng dụng vào thực tế, giúp những người có nhu cầu mua bán thức ăn nhanh có thể thao tác dễ dàng và tiện lợi.

Em hy vọng dựa trên nền tảng lý thuyết đã được thầy cô truyền thụ lại, kết hợp với sự tìm hiểu công nghệ của cá nhân, luận văn tốt nghiệp của em sẽ đạt được mục đích mong đợi.

## **II.2 Ý nghĩa của đề tài**

Trước hết, đề tài “Xây dựng ứng dụng mua bán fastfood trên nền Android” là một ứng dụng được xây dựng như một đề tài thể hiện việc áp dụng những kiến thức quý báu đã được các thầy cô của trường Đại học Bách Khoa – Đại học Đà Nẵng nhiệt tình truyền thụ lại cho chúng em. Và đặc biệt là sự theo dõi và quan tâm giúp đỡ của thầy Nguyễn Văn Hiệu trong suốt thời gian em thực hiện đề tài 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.

Thứ ba, trong thời đại công nghệ thông tin phát triển như vũ bão ngày nay, công nghệ không ngừng được cải tiến, những công nghệ mới ra đời đòi hỏi người lập trình viên phải cố gắng nghiên cứu và tìm hiểu để áp dụng vào thực tiễn. Điều này cũng được thể hiện qua sự cố gắng em khi quyết định thực hiện đề tài này với các công nghệ được coi là mới nhất hay và được dự báo sẽ phát triển mạnh mẽ trong tương lai.

Do vậy, em rất hy vọng sẽ tạo nên một sản phẩm hoàn chỉnh nhất có thể để đáp lại sự nhiệt tình giúp đỡ từ thầy cô và cả tâm huyết của em khi quyết định thực hiện đề tài này.

## **.III NHIỆM VỤ CỦA ĐỀ TÀI**

---

### III.1.Về lý thuyết

Đề tài “Xây dựng ứng dụng mua bán fastfood trên nền Android” là một ứng dụng chạy trên điện thoại sử dụng hệ điều hành Android. Ứng dụng gồm:

- Phần ứng dụng chạy trên điện thoại được phát triển trên nền Android SDK.
- Phần mềm quản lý cho phép người quản lý có thể thêm xóa cập nhật món ăn,xử lý các thông tin từ ứng dụng Android gửi về

Dưới đây là những chức năng chính mà em dự định phát triển ở phiên bản đầu tiên của ứng dụng này:

Đề tài xây dựng ứng dụng mua bán fastfood trên nền android.Khách hàng tải ứng dụng về điện thoại có sử dụng hệ điều hành android 4.0.Khách hàng chạy ứng dụng có thể xem danh sách món ăn mà nhà hàng đã cập nhật trong cơ sở dữ liệu.Khách hàng có thể chọn món và đăng ký thông tin để mua.Sau khi khách hàng chọn mua,thông tin sẽ được chuyển về lưu vào cơ sở dữ liệu thông qua Web Service.Ngoài ra,ta sẽ xây dựng một phần mềm quản lý dành cho nhà hàng để xử lý các thông tin mà khách hàng gửi về từ ứng dụng android,xử lý các yêu cầu của khách hàng.Quản lý danh mục món ăn như cập nhật,xóa,thống kê..Mục tiêu của đề tài này là.

- Tìm hiểu các hoạt động kinh doanh trong thực tế.
- Tìm hiểu cách thức xây dựng một ứng dụng trên android.
- Xây dựng *thành công ứng dụng mua bán fastfood (thức ăn nhanh) trên android.*
- Xây dựng phần mềm quản lý nhà hàng.
- Tìm hiểu cách thức xây dựng ứng dụng bán hàng
- Tìm hiểu bộ công cụ Visual Studio 2010 (chủ yếu là ngôn ngữ C#)
- Tìm hiểu ứng dụng kết nối Ksaop2 giữa Android và Web Service
- Tìm hiểu bộ Android Development Tool

---

### III.2.Về chương trình

- Chương trình thực hiện được các yêu cầu đề ra của đề tài tốt nghiệp
- Ứng dụng fastfood đã phần nào xây dựng và đáp ứng được một số chức năng chính:
  - Quản lý nhà hàng
  - Quản lý việc đặt hàng, xử lý giỏ hàng,...
  - Xử lý thống kê theo yêu cầu như: đơn hàng, hóa đơn
  - Cho phép tìm kiếm thông tin trong hệ thống,...
  - Giao diện thân thiện, dễ sử dụng.

### IV PHƯƠNG PHÁP TRIỂN KHAI

Xây dựng hệ thống bao gồm ứng dụng chạy trên điện thoại và một phần mềm quản lý để quản lý các thông tin *cần rất nhiều thời gian, tâm huyết và cả công sức.*

*Vì vậy trong phạm vi luận văn tốt nghiệp lần này, em quyết định thực hiện các phân cơ bản đến các phần nâng cao theo các bước sau:*

- *Khảo sát lấy yêu cầu người sử dụng điện thoại về các ứng dụng mua bán trên mobile để xây dựng ý tưởng, khảo sát các trang web chuyên về kinh doanh.*
- Tiếp đến, em sẽ nghiên cứu công nghệ phát triển ứng dụng chạy trên hệ điều hành Android. Xây dựng phía client một app có thể cài đặt trên điện thoại chạy Android đáp ứng các yêu cầu cơ bản của hệ thống.
- Sau khi hoàn thành phần ứng dụng chạy trên điện thoại, em sẽ nghiên cứu đến việc lấy dữ liệu từ server về sau đó đưa lên ứng dụng. Ở đây, em sẽ xây dựng một phần mềm cho phép quản lý dữ liệu trả về của ứng dụng.

Sau đây là danh sách các công nghệ sử dụng để xây dựng ứng dụng :

- Android SDK
- C#
- Web Service
- Java

- 
- HTML
  - SQL Server

Ngoài phần mở đầu và kết luận, bố cục của đề tài còn có thêm 3 chương.

*Chương 1:* Cơ sở lý thuyết tập trung tìm hiểu các cơ sở lý thuyết liên quan, các công cụ hỗ trợ để xây dựng ứng dụng của đề tài như..

- Lý thuyết về Android
- Cơ sở dữ liệu Sql Server 2008
- Visual Studio 2010 (Web Service, Window Form)

*Chương 2:* Phân tích và thiết kế tập trung phân tích các yêu cầu, thuật toán cần xử lý ứng dụng của đề tài. Thiết kế cơ sở dữ liệu. Đặc tả sử dụng xử lý các chức năng như đặt hàng, chọn món ăn, đăng ký dành cho khách hàng. Thống kê và quản lý các danh mục bên phía nhà hàng. Gồm các danh mục

- Giới thiệu ứng dụng
- Giới thiệu các chức năng chính của ứng dụng
- Phân tích thiết kế hệ thống
- Thiết kế cơ sở dữ liệu

*Chương 3:* Xây dựng và triển khai chương trình tập trung chủ yếu vào các modul chính của chương trình, cài đặt và xây dựng ứng dụng.

---

## CHƯƠNG 1

---

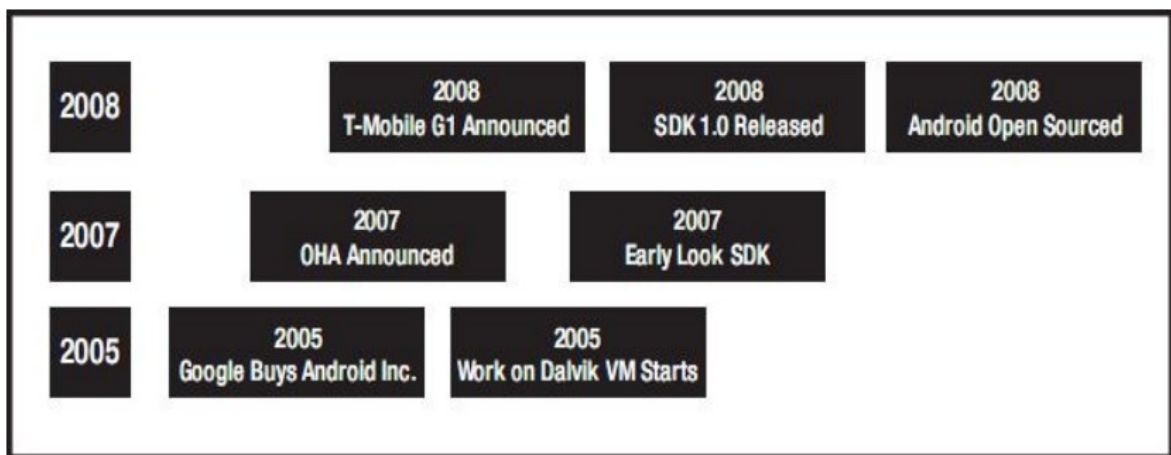
# CƠ SỞ LÝ THUYẾT

## I. Cơ sở lý thuyết liên quan đến đề tài

### I.1. Lý thuyết về Android

#### I.1.1 Lịch sử Android

Ban đầu, Android là hệ điều hành cho các thiết bị cầm tay dựa trên lõi Linux do công ty Android Inc. (California, Mỹ) thiết kế. Công ty này sau đó được Google mua lại vào năm 2005 và bắt đầu xây dựng Android Platform. Các thành viên chủ chốt ở Android Inc. gồm có: Andy Rubin, Rich Miner, Nick Sears, and Chris White..



*Hình 1. Android timeline*

Và sau tiếp, vào cuối năm 2007, thuộc về Liên minh Thiết bị Cầm tay Mã Nguồn mở (Open Handset Alliance) gồm các thành viên nổi bật trong ngành viễn thông và thiết bị cầm tay như:

Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, and Vodafone Group,...



---

Mục tiêu của Liên minh này là nhanh chóng đổi mới để đáp ứng tốt hơn cho nhu cầu người tiêu dùng và kết quả đầu tiên của nó chính là nền tảng Android. Android được thiết kế để phục vụ nhu cầu của các nhà sản xuất thiết bị, các nhà khai thác và các lập trình viên thiết bị cầm tay.

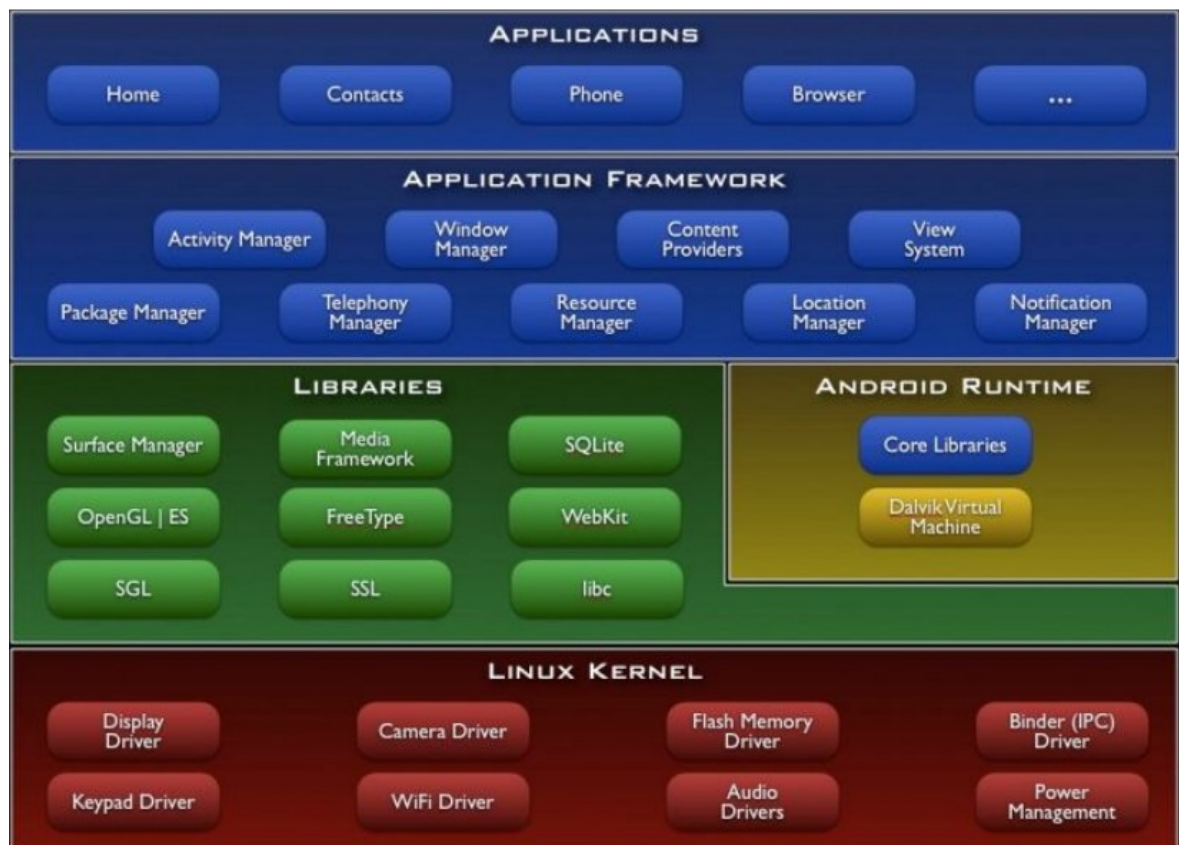
Phiên bản SDK lần đầu tiên phát hành vào tháng 11 năm 2007, hãng T-Mobile cũng công bố chiếc điện thoại Android đầu tiên đó là chiếc T-Mobile G1, chiếc smartphone đầu tiên dựa trên nền tảng Android. Một vài ngày sau đó, Google lại tiếp tục công bố sự ra mắt phiên bản Android SDK release Candidate 1.0. Trong tháng 10 năm 2008, Google được cấp giấy phép mã nguồn mở cho Android Platform.

Khi Android được phát hành thì một trong số các mục tiêu trong kiến trúc của nó là cho phép các ứng dụng có thể tương tác được với nhau và có thể sử dụng lại các thành phần từ những ứng dụng khác. Việc tái sử dụng không chỉ được áp dụng cho các dịch vụ mà nó còn được áp dụng cho cả các thành phần dữ liệu và giao diện người dùng.

Vào cuối năm 2008, Google cho phát hành một thiết bị cầm tay được gọi là Android Dev Phone 1 có thể chạy được các ứng dụng Android mà không bị ràng buộc vào các nhà cung cấp mạng điện thoại di động. Mục tiêu của thiết bị này là cho phép các nhà phát triển thực hiện các cuộc thí nghiệm trên một thiết bị thực có thể chạy hệ điều hành Android mà không phải ký một bản hợp đồng nào. Vào khoảng cùng thời gian đó thì Google cũng cho phát hành một phiên bản vá lỗi 1.1 của hệ điều hành này. Ở cả hai phiên bản 1.0 và 1.1 Android chưa hỗ trợ soft-keyboard mà đòi hỏi các thiết bị phải sử dụng bàn phím vật lý. Android cố định vấn đề này bằng cách phát hành SDK 1.5 vào tháng Tư năm 2009, cùng với một số tính năng khác. Chẳng hạn như nâng cao khả năng ghi âm truyền thông, vật dụng, và các live folder..

## **1.1.2 KIẾN TRÚC CỦA ANDROID**

Mô hình sau thể hiện một cách tổng quát các thành phần của hệ điều hành Android. Mỗi một phần sẽ được đặc tả một cách chi tiết dưới đây.



Hình 2. Cấu trúc stack hệ thống Android

### 1.1.3 Application framework

Bằng cách cung cấp một nền tảng phát triển mở, Android cung cấp cho các nhà phát triển khả năng xây dựng các ứng dụng cực kỳ phong phú và sáng tạo. Nhà phát triển được tự do tận dụng các thiết bị phần cứng, thông tin địa điểm truy cập, các dịch vụ chạy nền, thiết lập hệ thống báo động, thêm các thông báo để các thanh trạng thái, và nhiều, nhiều hơn nữa.

Nhà phát triển có thể truy cập vào các API cùng một khuôn khổ được sử dụng bởi các ứng dụng lõi. Các kiến trúc ứng dụng được thiết kế để đơn giản hóa việc sử dụng lại các thành phần; bất kỳ ứng dụng có thể xuất bản khả năng của mình và ứng dụng nào khác sau đó có thể sử dụng những khả năng (có thể hạn chế bảo mật được thực thi bởi khuôn khổ). Cơ chế này cho phép các thành phần tương tự sẽ được thay thế bởi người sử dụng.

Cơ bản tất cả các ứng dụng là một bộ các dịch vụ và các hệ thống, bao gồm:

- Một tập hợp rất nhiều các View có khả năng kế thừa lẫn nhau dùng để thiết kế phần giao diện ứng dụng như: gridview, tableview, linearlayout,...

- 
- Một “Content Provider” cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác (chẳng hạn như Contacts) hoặc là chia sẻ dữ liệu giữa các ứng dụng đó.
  - Một “Resource Manager” cung cấp truy xuất tới các tài nguyên không phải là mã nguồn, chẳng hạn như: localized strings, graphics, and layout files.
  - Một “Notification Manager” cho phép tất cả các ứng dụng hiển thị các custom alerts trong status bar.

Activity Manager được dùng để quản lý chu trình sống của ứng dụng và điều hướng các activity.

#### **I.1.4 Library**

Android bao gồm một tập hợp các thư viện C/C++ được sử dụng bởi nhiều thành phần khác nhau trong hệ thống Android. Điều này được thể hiện thông qua nền tảng ứng dụng Android. Một số các thư viện cơ bản được liệt kê dưới đây:

- **System C library:** a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices.
- **Media Libraries** - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.
- **Surface Manager** – Quản lý việc truy xuất vào hệ thống hiển thị.
- **LibWebCore** - a modern web browser engine which powers both the Android browser and an embeddable web view.
- **SGL** - the underlying 2D graphics engine.
- **3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer.
- **FreeType** - bitmap and vector font rendering.
- **SQLite** - a powerful and lightweight relational database engine available to all applications.

#### **I.1.5 ANDROID EMULATOR**

---

Android SDK và Plugin Eclipse được gọi là một Android Developer Tool (ADT). Các Android coder sẽ cần phải sử dụng công cụ IDE (Integrated Development Environment) này để phát triển, debugging và testing cho ứng dụng. Tuy nhiên, các coder cũng có thể không cần phải sử dụng IDE mà thay vào đó là sử dụng command line để biên dịch và tất nhiên là vẫn có Emulator như thường.

Android Emulator được trang bị đầy đủ hầu hết các tính năng của một thiết bị thật. Tuy nhiên, một số đã bị giới hạn như là kết nối qua cổng USB, camera và video, nghe phone, nguồn điện giả lập và bluetooth.

Android Emulator thực hiện các công việc thông qua một bộ xử lý mã nguồn mở, công nghệ này được gọi là QEMU (<http://bellard.org/qemu/>) được phát triển bởi Fabrice Bellard.

## **I.2. Mô hình thiết kế Model View Controller (MVC) trong Asp Net**

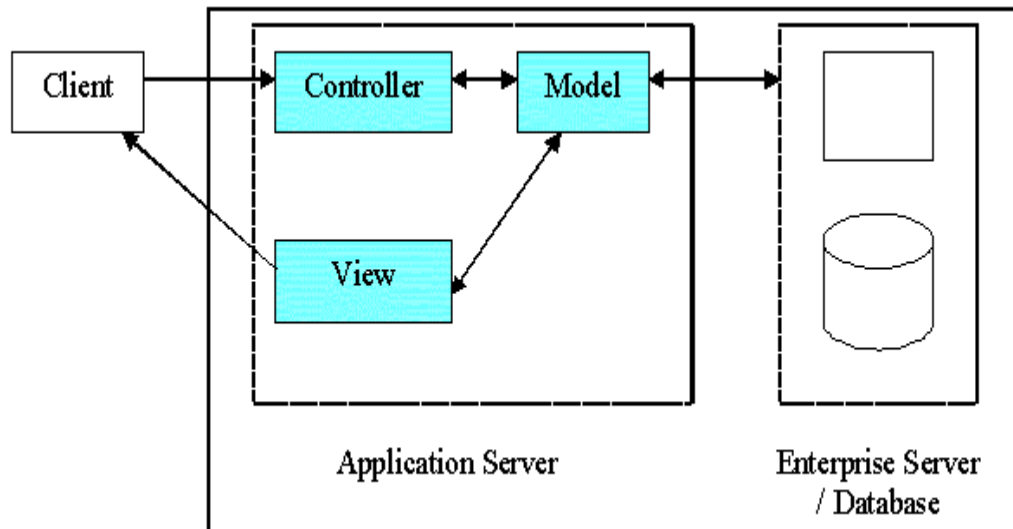
### **I.2.1 Model View Controller (MVC) là gì?**

MVC là tên một phương pháp chia nhỏ một ứng dụng thành ba thành phần để cài đặt, mỗi thành phần đóng một vai trò khác nhau và ảnh hưởng lẫn nhau, đó là models, views, và controllers.

- **Model (DAL – Data Access Layer):** Models trong các ứng dụng dựa trên MVC là những thành phần có nhiệm vụ lưu trữ thông tin, trạng thái của các đối tượng, thông thường nó là một lớp được ánh xạ từ một bảng trong CSDL. Lấy ví dụ, chúng ta có lớp Product được sử dụng để mô tả dữ liệu từ bảng Products trong SQL, bao gồm ProductID, OrderDate...
- **Views (Web – Website):** Views, nó chính là các thành phần chịu trách nhiệm hiển thị các thông tin lên cho người dùng thông qua giao diện. Thông thường, các thông tin cần hiển thị được lấy từ thành phần Models. Ví dụ, đối tượng Product có một "Edit" view bao gồm các textboxes, các dropdowns và checkboxes để chỉnh sửa các thuộc tính của sản phẩm; có một "Display" view gồm 2 dòng, cột dòng là ProductID, dòng sau là OrderDate... để xem thông tin về sản phẩm.
- **Controllers (Core):** Controllers trong các ứng dụng kiểu MVC chịu trách nhiệm xử lý các tác động về mặt giao diện, các thao tác đối với models, và cuối cùng là chọn một view thích hợp để hiển thị ra màn hình. Trong kiến trúc

---

MVC, view chỉ có tác dụng hiển thị giao diện mà thôi, còn điều khiển dòng nhập xuất của người dùng vẫn do Controllers đảm trách.



**Hình 3: Mô hình MVC**

- Khi người sử dụng ra một lệnh (gõ câu lệnh, bấm nút chuột, bấm phím, chọn menu ...), lệnh này sẽ được gửi tới phần Controller.
- Phần Điều khiển sẽ khởi tạo phần Model (nếu cần thiết), gửi các yêu cầu tới phần Model để thực hiện.
- Căn cứ trên các lệnh và thông tin nhận được từ lệnh, phần Model sẽ đảm nhận việc lấy thông tin và cập nhật thông tin trong các hệ thống khác, ví dụ trong các Enterprise Server như Application Server, Mail Server, Database Server, từ hệ thống .
- Sau khi hoàn thành việc thu thập, cập nhật thông tin, Model sẽ truyền những thông tin cần thiết về phần Controller.
- Lúc này, phần Controller sẽ quyết định chọn thành phần nào trong phần View để hiển thị dữ liệu ra cho người dùng.

- 
- Phần View khi làm nhiệm vụ hiện thông tin cho người dùng cũng có thể truy cập các thông tin hiển thị từ Model, hoặc gửi thông tin hiển thị tới Model. Trường hợp này xảy ra khi Model chứa các thông tin có thể dùng để hiện trực tiếp, ví dụ một danh sách khách hàng, hoặc một danh sách các e-mail trong mailbox. Khi phần View hiện thông tin, nó có thể báo cho phần Model biết nó đang hiện phần nào của thông tin, ví dụ như “Đang hiện thông tin khách hàng từ 20 đến 40”. Những thông tin loại này không cần thiết phải gửi qua trung gian Controller

### **I.2.2            Đặc điểm của Model View Controller (MVC)**

- Cho phép sự chia tách rành mạch giữa các thành phần, mỗi thành phần đều có khả năng được kiểm tra riêng biệt. Tất cả các mối liên hệ chính yếu trong MVC đều có giao diện để hiển thị và đều có thể xây dựng thành các mô hình, và bao gồm interface-based IRequest/IHttpResponse. Bạn có thể chạy unit test cho ứng dụng mà không phải chạy các Controllers trong ASP.NET process, điều này làm unit test chạy nhanh hơn. Bạn có thể sử dụng unit testing framework nào cũng được, kể cả NUnit, MBUnit, MS Test...
- Được thiết kế để có thể dễ mở rộng và tiện dụng, tất cả mọi thứ trong MVC framework đều có thể nhanh chóng được thay đổi, tùy chọn hóa. Ví dụ bạn có thể viết một View và sử dụng cho nhiều ứng dụng khác nhau, đây là tính pluggable.
- Bao gồm thành phần ánh xạ URL cho phép bạn xây dựng các ứng dụng thân thiện với bộ máy tìm kiếm và với người dùng. Lấy ví dụ tôi có thể ánh xạ URL /products/edit/4 là một trang cho phép chỉnh sửa thông tin về các sản phẩm, hoặc là /Blogs/scottgu/10-10-2007/SomeTopic/ để xem các thông tin.

---

- MVC framework hỗ trợ sử dụng các các tập tin. ASPX, ASCX và Master như là thành phần View, điều đó có nghĩa là bạn vẫn có thể sử dụng các tính năng của ASP.NET như master pages, <%= %> snippets, server controls, templates, data-binding, localization... Tuy nhiên nó không sử dụng mô hình post-back từ giao diện gửi đến server nữa, thay vào đó, bạn có thể chủ động đưa những post-back từ giao diện đó đến thẳng lớp Controller. Tóm lại, không còn viewstate hay là page lifecycle còn tồn tại trong mô hình MVC.

### **I.2.3 Lợi điểm của Model View Controller (MVC)**

- Nó bắt buộc phải tạo ra một sự rãnh mạch trong liên kết giữa 3 thành phần models, views và controllers trong ứng dụng. Duy trì được mối quan hệ rãnh mạch rõ ràng này khiến việc kiểm tra ứng dụng trở nên dễ dàng hơn, đồng thời trách nhiệm của từng thành phần được định nghĩa rõ ràng và cả ba đều làm việc ăn ý với nhau.
- Nó cho phép bạn dễ dàng bảo trì sự phân tách giữa các thành phần trong ứng dụng, đồng thời cũng rất tốt trong công việc testing

## **I.3. Hệ quản trị cơ sở dữ liệu Sql Server 2008**

Khi SQL Server 2008 ra đời, thì những kiến thức về SQL 2005 thực sự không đủ để bạn làm chủ được SQL 2008. Khả năng và độ linh hoạt của SQL Serve 2008 được phản ánh rất rõ trong công cụ Management Studio mới và BIDS.

### **I.3.1 NÂNG CAO BẢO MẬT**

- Bảo mật nhóm thư mục hệ thống
- Bắt buộc chính sách mật khẩu

- 
- Tách biệt gián đồ và người dùng
  - Tự động tạo chứng nhận cho SSL

### **I.3.2 TĂNG CƯỜNG KHẢ NĂNG QUẢN LÝ**

Những công cụ quản lý mới

- *Những công cụ quản lý mới:* Công cụ mới SQL Server Configuration Manager cho phép bạn kiểm soát các dịch vụ kết hợp với SQL Server 2008. Nó có thể thay thế cho Services Manager và công cụ cấu hình mạng cho Server và Client. Bạn cũng có thể kiểm soát một số dịch vụ khác như: SQL Server, SQL Agent, SQL Server Analysis Services, DTS Server (Cho SQL Server Integration Services), Full - Text Search, SQL Browser.
- *Profiler:* Cho phép bạn phân tích những vấn đề về hiệu suất thực thi trong SQL Server 2005. Ví dụ, Profiler mở các tập tin truy vết mà bạn đã lưu trong hệ thống tập tin để bạn xem lại và phân tích các quá trình SQL Server mà bạn quan tâm. Profiler có thể biểu diễn thông tin truy vết ở dạng đồ thị để bạn có thể dễ dàng xem điều gì đã xảy ra. Nó có thể nhận dữ liệu được ghi lại bởi Windows Performance Monitor. Bạn có thể hiển thị dữ liệu dạng đồ thị, xem hiệu suất thực thi trên khoảng thời gian đã chọn. Từ đồ thị, bạn có thể truy cập đến điểm có vấn đề.
- *SQL Server Agent:* Những khả năng của SQL Server Agent, thành phần hỗ trợ cho các tác vụ đã được lập thời gian biểu, được nâng cao. ví dụ, số tác vụ đồng thời mà SQL Server Agent có thể chạy được tăng lên. SQL 2000 chỉ dùng SQL Agent trong những tác vụ liên quan đến cơ sở dữ liệu CSDL. còn trong 2008, SQL Server Agent thực thi các tác vụ cho Analysis Services và Integration Services. SQL Server Agent dùng Windows Management Instrumentation (WMI), cho phép bạn viết mã tránh thực thi tác vụ, như khi đĩa cứng đầy thì các tác vụ vẫn được thực thi thành công.
- *Cấu hình động:* Trong SQL Server 2008, bạn có thể thực hiện bất kỳ Thay đổi cấu hình nào mà không cần khởi động lại SQL Server, kể cả khi bạn đang chạy trên Windows Server 2003. Bên cạnh đó, bạn cũng có thể thay đổi áp lực CPU và I/O nếu bạn cần, có thể thêm nóng bộ nhớ cho Server nếu bạn có phần cứng thích hợp.

### **I.4. Mô hình thiết kế web service,sql server và ứng dụng android**



---

### **I.4.1 Khái niệm về web service và Soap2**

Webservice là một dịch vụ cung cấp cơ chế triệu gọi các đối tượng từ xa thông qua giao thức HTTP cùng với cơ chế truyền tải định dạng đối tượng theo công nghệ XML.

Chính vì sử dụng giao thức HTTP của Web nên giờ đây các lời gọi trở nên đơn giản và thông qua được các rào cản về tường lửa. Để đảm bảo điều này, một giao thức mới là SOAP (Simple Object Access Protocol) ra đời để hỗ trợ cho Web services. SOAP được định nghĩa dựa trên giao thức chuẩn HTTP, SOAP cho phép dữ liệu chuyển đi bằng HTTP và định dạng theo chuẩn XML. Các lời gọi hàm tham số truyền hàm, dữ liệu trả về từ hàm, tất cả đều được chuyển sang dạng XML và có thể dễ dàng xử lý bởi tất cả các ngôn ngữ. Một thế mạnh khác đó là nếu các đối tượng phân tán xây dựng trên mô hình Web services sẽ có thể triệu gọi lẫn nhau, bất chấp đối tượng đó được viết trên ngôn ngữ Java của Sun hay .NET của Microsoft. Hiện tại, SOAP được coi là một sự thay đổi lớn kể từ khi COM, RMI, CORBA ra đời.

### **I.4.2 Giới thiệu về Xstream**

XStream là một công cụ giúp chuyển các đối tượng hay những thể hiện của những lớp Java qua dạng XML hay ngược lại. Nó là một mã nguồn mở, được thiết lập từ tháng giêng năm 2004.

Trong một đề án IT đôi khi bạn cần phải chuyển các đối tượng của các lớp Java có chứa thông tin và đưa nó qua dạng XML. Việc làm này để giúp mang thông tin từ hệ thống này qua hệ thống khác bằng những gói hay tập tin XML (giả sử các hệ thống này viết bằng ngôn ngữ Java). Nó cũng giúp bạn tránh được nhiều phiền toái như cách sắp đặt chuyển kiểu cho hai dữ liệu giữa hai hệ thống. Do đó dùng dạng XML như là phương tiện trao đổi dữ liệu giữa hai hệ thống là cách hữu hiệu nhất. Sau khi hệ thống đã nhận được dữ liệu nằm ở dạng XML rồi, thì việc kế tiếp là người lập trình chỉ chuyển chúng về các đối tượng Java để phù hợp với ngôn ngữ mà hệ thống đó đang dùng. Công cụ XStream giúp bạn thực hiện được giải pháp vừa nói ở trên. Nếu bạn không dùng XML như là phương tiện trao đổi dữ liệu, thì trong Java cũng có cách đưa đối tượng Java từ nơi này sang nơi khác là dùng Serialize. Bài này không nói đến Serialize, mà chỉ

---

nói đến công cụ XStream. Tất nhiên, ngoài XStream ra cũng có một công cụ nữa có chức năng tương còn được biết đến với cái tên Castor.

Thư viện XStream có thể tải tại <http://xstream.codehaus.org/index.html>

### **I.4.3 Thao tác với web service trong Android**

Cách gọi hàm từ webservice dotNet trong Android như sau:

Input: các tham số kiểu String

Output: giá trị kiểu String

```
public static Result addUser(int from, String username, String
    display_name)throws Exception {
    String SOAP_ACTION = "http://tempuri.org/AddUsername";
    String METHOD_NAME = "AddUsername";
    String NAMESPACE = "http://tempuri.org/";
    String URL = "http://10.0.2.2:1217/Service1.asmx?
op=AddUsername";
    SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);

    request.addProperty("from", from);
    request.addProperty("username", username);
    request.addProperty("display_name", display_name);

    SoapSerializationEnvelope envelope =
        new SoapSerializationEnvelope(SoapEnvelope.VER11);
    envelope.dotNet = true;
    envelope.setOutputSoapObject(request);
    Trans trans = new Trans(URL);
    trans.call(SOAP_ACTION, envelope);
    SoapPrimitive result = (SoapPrimitive) envelope.getResponse();
    if (result.equals("-1")) {
        return Result.EXCEPTION;
    } else if (result.equals("0")) {
```

---

```
        return Result.FAILED;
    } else {
        return Result.SUCCEEDED;
    }
}
```

Các biến SOAP\_ACTION, METHOD\_NAME, NAMESPACE, URL để xác định tên phương thức, port mà webservice đang sử dụng,...

Phương thức addProperty(String var\_name, String value) có 2 tham đối.

Tham đối thứ nhất là tên biến cần truyền tham trị vào và tham đối thứ 2 là giá trị của tham biến.

Sau khi thực hiện lệnh gọi hàm trans.call(SOAP\_ACTION, envelope); thì các giá trị truyền vào sẽ được chuyển đổi thành XML và truyền lên webservice.

Kết quả trả về thông qua đối tượng SoapPrimitive hoặc SoapObject. Đối với giá trị trả về là một kiểu chuỗi thì có thể thực hiện ép kiểu trực tiếp nhưng còn đối với giá trị trả về là một kiểu danh sách thì đối tượng SoapObject cho phép ta có thể duyệt tới từng phần tử trong danh sách. Thực chất đó là một quá trình mã hoá và giải mã một nội dung XML mà đã được SOAP hỗ trợ.

Phương thức dưới đây sẽ mô tả cách nhận về dữ liệu kiểu danh sách từ webservice:

```
public static ArrayList<UserInfo> getListUsername(int
userid)throws
    Exception {
    String SOAP_ACTION = "http://tempuri.org/GetListUsernameOf";
    String METHOD_NAME = "GetListUsernameOf";
    String NAMESPACE = "http://tempuri.org/";
    String URL =
        "http://10.0.2.2:1217/Service1.asmx?op=GetListUsernameOf";
    SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
    request.addProperty("userid", userid);
    SoapSerializationEnvelope envelope =
        new SoapSerializationEnvelope(SoapEnvelope.VER11);
```

---

```
envelope.dotNet = true;
envelope.setOutputSoapObject(request);
Trans trans = new Trans(URL);
trans.call(SOAP_ACTION, envelope);
SoapObject resultsRequestSOAP = (SoapObject) envelope.bodyIn;
SoapObject a = (SoapObject) resultsRequestSOAP.getProperty(0);
int count = a.getPropertyCount();
ArrayList<UserInfo> bki = new ArrayList<UserInfo>();
for (int i = 0; i < count; i++) {
    SoapObject so = (SoapObject) a.getProperty(i);
    String us = so.getProperty("display_name").toString();
    String rname = so.getProperty("username").toString();
    String id = so.getProperty("id").toString();
    bki.add(new UserInfo(id, us, rname));
}
return bki;
}
```

## II. Nghiên cứu thực tiễn

- Cách thức hoạt động mua bán hiện nay ở các website mua bán trực tuyến trong nước và ngoài nước

+ Các website trong nước:

Website bán thức ăn nhanh của công ty KFC <http://www.kfcvietnam.com.vn/>

Website bán hàng đa cấp: <http://www.sieuthinhanh.com/>

Website quà tặng quảng cáo: <http://www.quatangquangcao.net/>

Website tặng họa: <http://dienhoa.vn/>

Website tặng quà cd: <http://www.quatangcd.com/>

Website hoa tươi quà tặng: <http://www.hoatuoiquatang.com/>

+ Các website nước ngoài:

<http://www.buy.com/>, <http://www.ebay.com/>, <http://www.amazon.com/>

- 
- Cách web service ứng dụng vào phần mềm và ứng dụng của android.

## PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### I. PHÂN TÍCH CÁC CHỨC NĂNG CHÍNH CỦA ỨNG DỤNG

Một số chức năng chính của ứng dụng:

#### I.1. Hiện thị danh sách thông tin món ăn

- Chức năng này sẽ hiển thị tất cả danh sách thông tin của món ăn khi chạy ứng dụng.
- Mỗi món ăn sẽ được liệt kê trong một danh sách bao gồm hình ảnh, tên, giá bán, mô tả thông tin món ăn đó.
- Người dùng có thể lọc danh sách món ăn.

#### I.2. Chức năng tìm kiếm món ăn.

- Chức năng này cho phép người dùng có thể tìm ra những món ăn có nội dung đúng với từ khóa.

#### I.3. Chức năng đăng ký tài khoản

- Chức năng này cho phép người sử dụng đăng ký một tài khoản để sử dụng đặt hàng cần mua trên ứng dụng.
- Thông tin đăng ký bao gồm tên mã, tên đầy đủ, ngày sinh, địa chỉ và số điện thoại để có thể đặt mua món ăn.

#### I.4. Chức năng đặt hàng

- Chức năng này cho phép người sử dụng đã đăng ký mã.
- Người dùng có thể chọn món hàng mình mua, hiển thị vào giỏ hàng, nhập số lượng cần mua và nhập mã mà mình đã đăng ký với hệ thống để đặt mua những món mình chọn.

#### I.5. Các chức năng của người quản lý bên nhà hàng

- Cập nhật danh sách món ăn (Thêm, sửa, xóa).
- Quản lý danh sách khách hàng đã đăng ký.
- Xem danh sách đơn đặt hàng của các khách hàng đã đặt.
- Thống kê.

---

## **.II PHÂN TÍCH VÀ ĐẶT TẢ CÁC YÊU CẦU**

### **II.1. Biểu đồ ca sử dụng(Usecase)**

#### **II.1.1. Khách hàng**

#### **II.1.2.Quản lý**

### **II.2. Đặt tả ca sử dụng**

#### **II.2.1.Khách hàng**

##### **II.2.1.1. Chọn món ăn**

- Tác vụ này dành cho khách hàng thực hiện khi muốn chọn một món ăn vào đơn đặt hàng của mình.

Luồng dữ liệu vào:

+ Mã món ăn

- Khi khách hàng chọn món ăn thì hệ thống sẽ hiển thị các thông tin về món ăn (tên món ăn, giá, số tiền, số lượng (khách hàng nhập vào text) trong giỏ hàng. Khi khách hàng thay đổi số lượng món ăn sau đó chọn mục cập nhật thì số lượng món ăn sẽ thay đổi tương ứng.

- Khách hàng có thể thao tác với nút xóa

-

Luồng dữ liệu ra

- Hiển thị thông tin về Giỏ hàng
- Phát thảo giao diện

##### **II.2.1.2.**

##### **II.2.1.3. Đăng ký**

- Tác vụ này dành cho khách hàng muốn đăng kí thông tin tài khoản

Luồng dữ liệu vào:

+ Mã khách hàng

---

+ Tên khách hàng

+ Địa chỉ

Số nhà-Tổ/Phường

Quận

+ Số điện thoại

Xử lý hệ thống:

-

→ Mã khách hàng: được nhập từ bàn phím (ít nhất là 6 kí tự, nhiều nhất là 25 kí tự). Khi khách hàng chọn mục đồng ý thì mã khách hàng của khách hàng được tạo ra bao gồm: chuỗi kí tự khách hàng nhập vào (Ví dụ: người quản lý nhập vào Truongcongkhoei thì hệ thống sẽ tạo ra mã thẻ khách hàng là Truongcongkhoei)

→ Tên khách hàng (nhập họ tên phải có khoảng cách)

→

→

-

Luồng dữ liệu ra:

- Thông báo kết quả

+ Thông báo chưa nhập đầy đủ các thông tin đăng kí

+ Thông báo tài khoản đã tồn tại

- Thông báo đã đăng nhập thành công. Và vào màn hình chính của chương trình

- Phát thảo giao diện

**Giao diện khách hàng đăng ký tài khoản**

## **II.2.2. Quản lý**



---

### **II.2.1.1. Cập nhật thông tin món ăn**

Tác vụ này thực hiện khi người quản lý muốn đưa thông tin món ăn, thêm, xóa, sửa khi cần thiết

Luồng dữ liệu vào:

- Thông tin món ăn: (Hình món ăn, Mã món ăn, Tên món ăn, Giá bán, Mô tả)  
Xử lý hệ thống

+ Khi chọn vào thao tác với chức năng thêm mới thì hệ thống hiển thị giao diện cho phép nhập thông tin món ăn, sau khi nhập, hệ thống sẽ kiểm tra tính hợp lệ của mã món ăn mới nhập (không được trùng với mã món ăn đã tồn tại trong dữ liệu )

→

→

→

+ Khi người quản lý chọn mục xóa thì hệ thống sẽ hiển thị bảng yêu cầu xác nhận mật khẩu quản lý, Sau khi người quản lý xác nhận mật khẩu thì hệ thống sẽ kiểm tra mật khẩu trong dữ liệu của hệ thống có phải là của quản lý hay không? Nếu phải thì hệ thống sẽ hiển thị bảng yêu cầu xác nhận xóa thông tin món ăn.

Luồng dữ liệu ra:

- Thông báo hợp thoại quá trình cập nhật thành công/ thất bại
  - + Hiển thị giao diện cho chức năng sửa thông tin món ăn
  - + Hiển thị giao diện cho phép thêm mới món ăn

Phát thảo giao diện

**Giao diện thông tin món ăn**

---

### II.2.1.2.

#### Thông tin món ăn

-

Luồng dữ liệu vào:

- Tên món ăn

Xử lý hệ thống:

-

- Khi người quản lý chọn mục sửa thì hệ thống sẽ chuyển đến chức năng sửa thông tin món ăn ở tác vụ cập nhật thông tin món ăn

- Khi người quản lý chọn mục xóa thì hệ thống sẽ hiển thị bảng yêu cầu xác nhận mật khẩu quản lý, Sau khi người quản lý xác nhận mật khẩu thì hệ thống sẽ kiểm tra mật khẩu trong dữ liệu của hệ thống có phải là của quản lý hay không? Nếu phải thì hệ thống sẽ hiển thị bảng yêu cầu xác nhận xóa thông tin món ăn.

Luồng dữ liệu ra:

- Hiển thị giao diện thống kê các món ăn người quản lý cần tìm

Phát thảo giao diện:

-

Luồng dữ liệu vào

- Mã hóa đơn

Xử lý hệ thống

- Sau khi quản lý nhập mã hóa đơn cần tìm và chọn mục tìm kiếm thì hệ thống sẽ tìm kiếm trong dữ liệu hệ thống tất cả các mã hóa đơn (là các

---

Luồng  
dữ liệu  
ra:

- Hiển thị thông tin chi tiết về đơn hàng có mã đơn hàng vừa nhập

Phát  
thảo  
giao  
diện

### **Giao diện tìm kiếm hóa đơn**

#### **II.2.1.3. Xem thông tin**

- Tác vụ này thực hiện khi người quản lý muốn xem, in hóa đơn của khách hàng

Luồng dữ liệu vào:

+ Mã hóa đơn

Xử lý hệ thống

- Quản lý nhập mã hóa muốn xem và chọn mục xem. Thì hệ thống sẽ tìm kiếm các hóa đơn có mã hóa đơn trong dữ liệu hệ thống?

+ Nếu mã hóa đơn không được tìm thấy trong cơ sở dữ liệu thì hệ thống sẽ thông báo hóa đơn này không tồn tại và cho phép nhập lại

+ Nếu mã hóa đơn được tìm thấy có trong cơ sở dữ liệu thì hệ thống sẽ cho hiển thị thông tin của hóa đơn cần xem.

- Ngoài ra, tác vụ này có thể được chuyển đến từ tác vụ thống kê hóa đơn khi quản lý click xem chi tiết của giao diện thống kê hóa đơn, hệ thống sẽ hiển thị ra thông tin chi tiết của hóa đơn:

- Thông tin trong hóa đơn gồm:

- 
- + Thông tin đơn hàng: mã hóa đơn, ngày giao
  - + Thông tin khách hàng: tên khách hàng, địa chỉ khách hàng, số điện thoại khách hàng
  - + Thông tin món ăn: tên món ăn, đơn giá, số lượng
  - + Số tiền trong đơn hàng: tiền món ăn, số lượng, tổng tiền

Cách tính: Thành tiền = số lượng \* đơn giá

$$\text{Tổng tiền} = \sum (\text{Thành tiền})$$

Luồng dữ liệu ra:

- Hiện thị giao diện hóa đơn

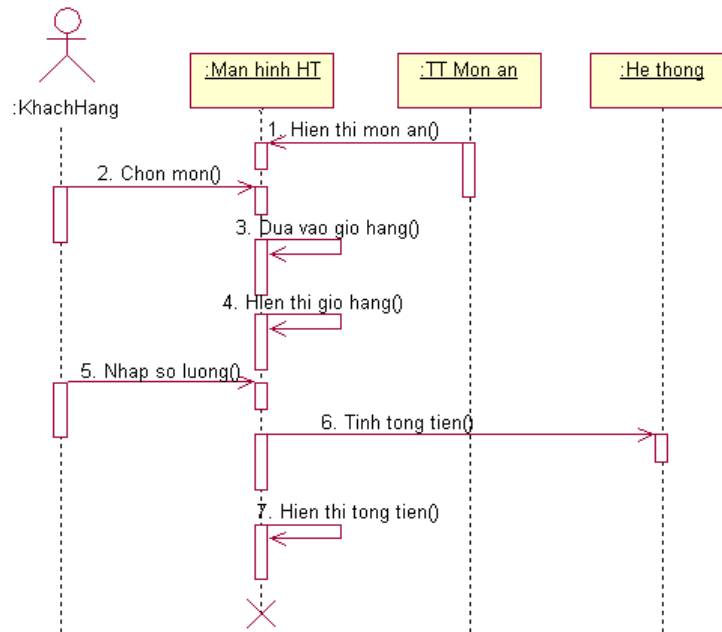
#### **II.2.1.4. Xóa thông tin tài khoản khách hàng**

- Tác vụ này thực hiện khi quản lý muốn xóa thông tin tài khoản khách hàng uồng dữ liệu vào:
- Mã khách hàng  
Xử lý hệ thống

**Giao diện xóa tài khoản khách hàng**

### **II.3. Biểu đồ tuần tự**

#### **II.3.1. Chọn món ăn**



Hình 54: Biểu đồ tuần tự chọn món ăn

### II.3.2. Đặt hàng

### II.3.3. Đăng ký

### II.3.4. Cập nhật thông tin món ăn

## II.4. Biểu đồ lớp

Hình 78: Biểu đồ lớp

## II.5. Xây dựng cấu trúc cơ sở dữ liệu

### II.5.1. Danh sách dữ liệu sơ cấp

Stt	Dữ liệu	Giải thích
1	HìnhMN	Hình ảnh món ăn
2	MaMN	Mã món ăn
3	TenMN	Tên món ăn
4	Gia	Giá món ăn
5	SoLuong	Số lượng món ăn
6	SoTien	Số Tiền món ăn ( Giá * Số lượng)
7	TenKH	Tên khách hàng
8	ĐT	Điện thoại

9	Diachi	Số nhà – Tổ / Phường-Thành phố
10	MaKH	Mã khách hàng
11	MoTa	Mô tả món ăn
12	MaDH	Mã đơn hàng
13	MaHD	Mã hóa đơn(Chính là mã đơn hàng, khi khách hàng đã thanh toán)
14	TongCong	Tổng tiền trong đơn hàng(Giá*Số lượng)
15	Ngaysinh	Ngày sinh khách hàng

## II.5.2. Danh sách từ điển dữ liệu:

Stt	Dữ liệu	Loại dữ liệu	Khóa	Công thức tính
1	HinhMN	nvarchar(20)		
2	MaMN	nvarchar(20)	PK	
3	TenMN	nvarchar(20)		
4	Gia	numeric(18, 0)		
5	MoTa	ntext		
6	SoLuong	int		
7	SoTien	Numeric		Gia*SoLuong
8	TenKH	nvarchar(25)		
10	ĐiệnThoai	ntext		
12	NgàySinh	Smalldatetime		
15	MaKH	nvarchar(20)	PK	
26	MaDH	nvarchar(20)	PK	
27	MaHD	nvarchar(20)		
28	TongCong	Numeric		(Giá*Số lượng)

## II.6. Thiết kế cơ sở dữ liệu

### II.6.1. Thiết kế các table

Bảng: TblMonngon				
	Tên trường			
HinhMN	Text			
MaMN	nvarchar(20)			
TenMN	nvarchar(20)			
Gia	numeric(18, 0)			

MoTa	ntext		
<b>Bảng:</b> <b>TblTTHoaDon(Thông tin hóa đơn)</b>			
<i>Tên trường</i>			
MaHD	nvarchar(20)		
MaKH	nvarchar(20)		
NgàyGH	smalldatetime		
<b>Bảng:</b> <b>tbCTHoaDon(Chi tiết hóa đơn)</b>			
<i>Tên trường</i>	<i>Loại</i>		
MaDH	nvarchar(20)		
MaMH	nvarchar(20)	FK	
Soluong			
<b>Bảng:</b> <b>TblKhachhang</b>			
<i>Tên trường</i>			
MaKH	nvarchar(20)		
MaNV	nvarchar(25)		
TenKH	nvarchar(25)		
Ngàysinh	smalldatetime		
Diachinh	nvarchar(20)		
Diarnoi	nvarchar(20)		

Ghi

chú:

PF:

Khóa

chính, FK:

khóa

ngoại

---

## **XÂY DỰNG VÀ TRIỂN KHAI CHƯƠNG TRÌNH**

**I. Các modul chính của chương trình**

**II. Cài đặt chương trình**

**III. Kết quả đạt được**



---

# KẾT LUẬN

## I. Kết quả đạt được

### I.1 Về kiến thức

- Sau thời gian thực hiện đồ án tốt nghiệp em đã đạt được nhiều tiến bộ cả về mặt tìm hiểu, nghiên cứu lý thuyết lẫn kỹ năng lập trình. Có thể nói, thông qua đồ án tốt nghiệp, em đã đạt được:
- Hiểu biết nhiều hơn về các kỹ thuật và kỹ xảo trong lập trình Java
- Phân tích thiết kế theo hướng đối tượng theo hướng chuyên nghiệp hơn.
- Hiểu hơn về nghiệp vụ bán hàng trực tuyến.
- Nâng cao tinh thần tự học, tự nghiên cứu

### I.2. Về chương trình

- Chương trình thực hiện được các yêu cầu đề ra của đề tài tốt nghiệp
- Ứng dụng fastfood đã phần nào xây dựng và đáp ứng được một số chức năng chính:
- Quản lý nhà hàng
- Quản lý việc đặt hàng, xử lý giỏ hàng,...
- Xử lý thống kê theo yêu cầu như: đơn hàng, hóa đơn
- Cho phép tìm kiếm thông tin trong hệ thống,...
- Giao diện thân thiện, dễ sử dụng.

## II. Hạn chế

- Do thời gian thực hiện phân tích và thiết kế hệ thống là tương đối hạn chế so với một đề tài tương đối rộng và phong phú nên không tránh khỏi những thiếu sót nhất định. Bên cạnh đó, chương trình còn một số chức năng chưa hoàn thiện và chính xác.
- Đây là một ứng dụng dựa trên mô hình client/server với số lượng người dùng khá lớn. Để đáp ứng được điều này, hệ thống đòi hỏi một máy chủ mạnh và ổn định.

- 
- Tuy nhiên, việc xây dựng và vận hành một máy chủ sẽ tốn rất nhiều công sức và chi phí. Đây thực sự là một khó khăn.

### **III. Hướng phát triển**

- Để tiếp tục phát triển đề tài này và có thể áp dụng trong thực tế, em nhận thấy cần phải tiếp tục thực hiện một số công việc sau:
  - Xử lý các lỗi chệch hơn trước khi đưa vào sử dụng.
  - Mở rộng bài toán cho nhiều doanh nghiệp riêng biệt
  - Linh hoạt hơn trong các sự kiện và yêu cầu của khách hàng
  - Bổ sung nhiều hơn các báo cáo, thống kê.
  - Hoàn thiện các chức năng phù hợp với thực tế hoạt động kinh doanh
  - Hỗ trợ các công việc liên quan đến kinh doanh (như: Thuế , ...)
  - Phát triển các tính năng hỗ trợ thanh toán (ATM , PayNet ,.....)

---

# **PHỤ LỤC**

## **TÀI LIỆU THAM KHẢO**