**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙜🙢🙠🙞

**

**ĐỒ ÁN MÔN HỌC**

**XÂY DỰNG ỨNG DỤNG FOODNOW TRÊN NỀN ANDROID KẾT HỢP VỚI WEBSERVICE (NODEJS VÀ MONGODB)**

**GVHD:** Ths. Vũ Đình Ái

**SINH VIÊN THỰC HIỆN:**

Kiều Hữu Thành-2001160343

Lê Thị Ngọc Hiền-2001160222

**CHUYÊN NGÀNH:** Công nghệ phần mềm

**NIÊN KHÓA:**2016-2020

Thành phố Hồ Chí Minh, tháng 07 năm 2020

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙜🙢🙠🙞

**

**ĐỒ ÁN MÔN HỌC**

**XÂY DỰNG ỨNG DỤNG FOODNOW TRÊN NỀN ANDROID KẾT HỢP VỚI WEBSERVICE (NODEJS VÀ MONGODB)**

**GVHD:** Ths. Vũ Đình Ái

**SINH VIÊN THỰC HIỆN:**

Kiều Hữu Thành-2001160343

Lê Thị Ngọc Hiền-2001160222

**CHUYÊN NGÀNH:** Công nghệ phần mềm

**NIÊN KHÓA:**2016-2020

Thành phố Hồ Chí Minh, tháng 07 năm 2020

# **LỜI CẢM ƠN**

Sau khi hoàn thành đề tài này, nhóm chúng tôi xin gửi lời cảm ơn đến khoa Công nghệ thông tin đã tiến hành tổ chức môn Đề tài một chỉ để sinh viên có thể tiếp cận với cách thực hiện một đề tài hoàn chỉnh. Tiếp theo, nhóm xin cảm ơn đến bạn bè xung quanh đã giúp đỡ nhóm trong các vấn đề khó, trao đổi với nhóm những tài liệu quý giá để nhóm hoàn thành đề tài này. Được chia sẻ kinh nghiệm cũng như thảo luận với các bạn đã giúp nhóm tiến bộ hơn trong việc nghiên cứu dồ án.

Quan trọng nhất, nhóm muốn gửi lời cảm ơn đến giáo viên hướng dẫn là thầy Vũ Đình Ái. Thầy là người đã hướng dẫn trực tiếp, định hướng đề tài, cung cấp nguồn tài liệu tham khảo quý giá và giúp đỡ nhóm khi nhóm có những vấn đề phát sinh. Nhờ có thầy mà đề tài “Xây dựng ứng dụng FoodNow trên nền Android kết hợp với webservice” được hoàn chỉnh các vấn đề ở giai đoạn một. Cám ơn thầy và mong có thể tiếp tục được làm việc với thầy vào giai đoạn tiếp theo của đề tài.

Cuối cùng nhóm xin cảm ơn đến những thầy cô và những người đang đọc, tham khảo và nhận xét cho đề tài này. Ý kiến của mọi người chính là những kim chỉ nam hướng dẫn nhóm định hướng phát triển đề tài ở các giai đoạn tiếp theo. Từ đó nhóm có thể tạo ra được sản phẩm hoàn thiện có thể áp dụng trong đời sống và giúp đỡ mọi người.

Lời nói cuối cùng, nhóm xin chúc sức khỏe và thành công đến tất cả mọi người. Xin chân thành cảm ơn.

**SINH VIÊN THỰC HIỆN**

**Kiều Hữu Thành**

**Lê Thị Ngọc Hiền**

# **LỜI MỞ ĐẦU**

Hiện nay, ngành Thương mại điện tử đang phát triển mạnh mẽ và dần dần được sử dụng rộng rãi ở khắp mọi nơi với các ngành hàng đa dạng. Dù bất cứ đâu, với một thiết bị thông minh như điện thoại hay laptop có kết nối mạng, mọi người đã có thể mua hàng mà không phải tới tận cửa hàng. Các ngành hàng ngày càng được đa dạng hóa như sách báo, quần áo , đồ gia dụng, cho đến các loại mỹ phẩm, hóa phẩm sinh hoạt,… Và các mặt hàng liên quan đến ăn uống cũng không ngoại lệ, nhất là khi các cửa hàng ăn uống đa dạng dành cho tất cả mọi người đang bùng phát mạnh mẽ hơn hết. Vấn đề đặt ra là để tham gia được sàn Thương mại điện tử thì các cửa hàng cần một trung gian thực hiện việc giao tiếp với khách hàng trên nền tảng trực tuyến, kết nối nhanh chóng và tiện lợi nhất cho khách hàng.

Để giải quyết vấn đề nói trên, nhóm chúng tôi muốn xây dựng một ứng dụng trung gian trên nền tảng di động -trước tiên là Android- để các cửa hàng có thể nhận được các đơn hàng từ các khách hàng trong khu vực. Bên phía cửa hàng có thể quản lý được doanh thu, lượng sản phẩm, lượng khách hàng. Bên khách hàng có thể quản lý các đơn hàng, quản lý tài khoản, đánh giá chất lượng cho các cửa hàng.Ngoài ra, khi ứng dụng có thể hoạt động được trong thực tế còn có thể tạo việc làm cho bên thứ ba, cụ thể là những người giao hàng (shipper).

Xây dựng một ứng dụng nói chung hay các thiết bị di động nói riêng cần phải trải qua nhiều giai đoạn thực hiện. Cần xác định môi trường hoạt động, lựa chọn công nghệ lập trình, xây dựng cơ sở dữ liệu, thiết kế giao diện,… Và ở giai đoạn một này, nhóm sẽ tiến hành chọn công nghệ lập trình và thiết kế cơ sở dữ liệu, tạo nền tảng cho các giai đoạn tiếp theo.

Trong báo cáo này, nhóm sẽ trình bày theo 3 phần như sau:

Phần 1: Cơ sở lý thuyết.

Phần 2: Phân tích và thiết kế hệ thống.

Phần 3: Tổng kết.

# **MỤC LỤC**

Contents

[**LỜI CẢM ƠN** 3](#_Toc44588858)

[**LỜI MỞ ĐẦU** 4](#_Toc44588859)

[**MỤC LỤC** 5](#_Toc44588860)

[**DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT** 6](#_Toc44588861)

[**DANH MỤC HÌNH VẼ VÀ ĐỒ THỊ** 7](#_Toc44588862)

# **DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **Viết tắt** | **Tiếng anh** | **Tiếng Việt** |
| BPM | Business Process Diagram | Quy trình quản lí nghiệp vụ |
| ERD | Entity Relationship Diagram | Mô hình dòng dữ liệu |
| BFD | Business Function Diagram | Mô hình phân rã chức năng |
| DFD | Data flow diagram | Mô hình dòng dữ liệu |
| CSDL |  | Cơ sở dữ liệu |

# **DANH MỤC HÌNH VẼ VÀ ĐỒ THỊ**

1. **CƠ SỞ LÝ THUYẾT**
2. TÌM HIỂU CÔNG NGHỆ
   1. LẬP TRÌNH ANDROID

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Android, Inc. với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005.

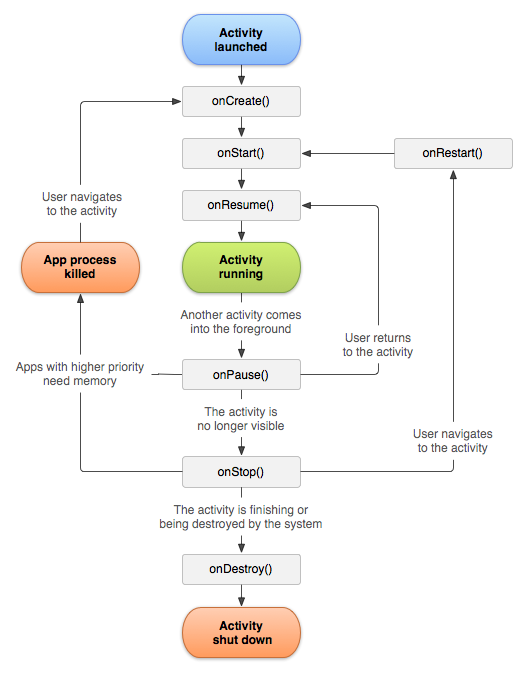
Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi. Tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt.

Các thành phần cơ bản cấu tạo nên một ứng dụng Android bao gồm:

* Activity: là nền của 1 ứng dụng. Khi khởi động 1 ứng dụng Android nào đó thì bao giờ cũng có 1 main Activity được gọi, hiển thị màn hình giao diện của ứng dụng cho phép người dùng tương tác.
* Service: thành phần chạy ẩn trong Android từ lúc chạy cho tới khi thiết bị cầm tay tắt đi. Service sử dụng để update dữ liệu, đưa ra các cảnh báo (Notification) và không bao giờ hiển thị cho người dùng thấy.
* Content Provider: được tạo ra để quản lý và chia sẻ dữ liệu với các hoạt động, dịch vụ khác.
* Broadcast Recevier: Chuyên lắng nghe và chờ đợi thông báo kích hoạt hệ thống để thực hiện một số công việc. Thông điệp có thể là: pin yếu, kết nối wifi, kết nối sạc,… Broadcast receiver có thể gủi phản hồi bằng cách thưc thi activity hoặc phát một “notification” để gấy chú ý người dùng.

Quá trình hoạt động của một Activity có các giai đoạn cơ bản sau:

* Active/running: Activity đang ở foreground (xảy ra khi gọi onResume() cho tới lúc gọi onPause()), lúc này activity đang tương tác với người dùng.
* Paused: Activity bị mất focus nhưng vẫn hiển thị với người dùng, lúc này activity không thể tương tác và bị che đi một phần hay hoàn toàn.
* Stop: Activity bị che đi hoàn toàn bởi một activity khác, người dùng không thể tương tác với activity này trừ khi tắt activity đã che nó.

Hình 1.1 Vòng đời của một Activity

* 1. NODEJS

Nodejs là một **nền tảng** (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.

Nodejs được tạo ra bởi Ryan Dahl từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ. Mục tiêu ban đầu của Dahl là làm cho trang web có [khả năng push](https://vi.wikipedia.org/w/index.php?title=C%C3%B4ng_ngh%E1%BB%87_push&action=edit&redlink=1) như trong một số ứng dụng web như [Gmail](https://vi.wikipedia.org/wiki/Gmail). Sau khi thử với vài ngôn ngữ Dahl chọn Javascript vì một API Nhập/Xuất không đầy đủ. Điều này cho phép anh có thể định nghĩa một quy ước Nhập/Xuất điểu khiển theo sự kiện, non-blocking.

Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao. Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực. Và áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.

* 1. MONGOBD

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng. MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

MongoDB được công ty 10gen software bắt đầu phát triển vào năm 2007 trong khi công ty này đang xây dựng một nền tảng dịch vụ (Platform as a Service). Đến năm 2009, dự án này được tách độc lập. Năm 2013, công ty 10gen được đổi tên thành MongoDB inc.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng. Trong MongoDB collection ứng với table, còn document sẽ ứng với row.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB.

* 1. FIREBASE

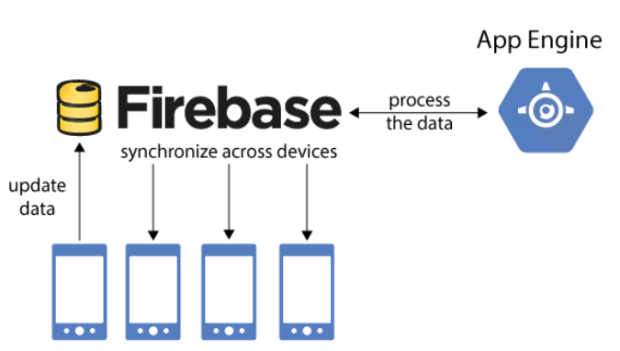
Realtime Database Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng có tương tác cơ sở dữ liệu (CRUD) một cách nhanh chóng và ngay tức thời (Realtime).

Thực ra ta cũng nên bày vẽ dịch Realtime Database Firebase ra Cơ sơ dữ liệu thời gian thực firebase

hay cũng không nên dịch Cloud ra chữ nền tảng đám mây. Cứ để Cloud là được rồi.

CRUD: Là 4 thao tác không thể thiếu với mọi ứng dụng có tương tác Cơ Sở Dữ liệu. C (Create- thêm mới), R (Retrieve – truy vấn xem dữ liệu), U (Update- cập nhật dữ liệu), D (Delete- xóa dữ liệu)

Khi có sự thay đổi dữ liệu trên Database Firebase thì ngay lập tức giao diện của bất kỳ thiết bị nào có sử dụng phần mềm này sẽ tự động cập nhật (gọi là Realtime):

[](https://duythanhcse.wordpress.com/2018/10/11/phat-trien-ung-dung-co-so-du-lieu-thoi-gian-thuc-voi-firebase-phan-1/firebase0/)Dữ liệu trên Database Firebase có định dạng JSON kiểu Gson.

* 1. HỆ THỐNG RESTFUL API

API là viết tắt của cụm từ Application Programming Interface,  là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST là viết tắt của cụm từ Representational State Transfer. REST không phải một chuẩn hay một giao thức, đây là một cách tiếp cận, một kiến trúc để viết API. Lần đẩu tiên được trình bày bởi Roy Fleding vào năm 2000 trong luận án về REST của ông. Một API được gọi là RESTful API khi đã thỏa mãn được 6 ràng buộc của kiến trúc REST:

* Máy chủ và máy khách (Server-Client): Bằng cách tách thành 2 phần là giao diện và dữ liệu, có thể tối ưu được phần giao diện người dùng trên nhiều thiết bị khác nhau và cải thiện khả năng mở rộng API bằng cách đơn giản hóa các component của máy chủ.
* Không trạng thái (Stateless): Mỗi yêu cầu gửi từ máy khách đến máy chủ đều chứa các thông tin cần thiết để xử lý và không thể được lưu lại trên máy chủ. Do đó, các trạng thái đều hoàn toàn giữ lại ở máy khách.
* Lưu trữ trên bộ nhớ cache (Cacheable): Các phản hồi từ máy chủ có thể được lưu trong bộ nhớ cache của máy khách. Các phản hồi này có thể được máy khách sử dụng lại cho các yêu cầu tương tự sau này.
* Thống nhất Interface (Uniform interface): Để có thể thống nhất interface, cần phải xác định bốn ràng buộc interface là Xác định resource, Thao tác với resource thông qua đại diện, Thông tin tự mô tả, Hypermedia là công cụ của trạng thái ứng dụng.
* Hệ thống phân lớp (Layered system): Phân lớp và xác định rõ các chức năng của của từng lớp sao cho mỗi lớp thực hiện đúng chức năng.
* Code theo yêu cầu (Code on demand): Chức năng tùy chọn. REST cho phép các máy khách mở rộng bằng cách tải xuống API và thực thi dưới dạng applet hay script, giúp đơn giản hóa bằng cách giảm số lượng chức năng chưa cần thiết.

Phương thức hoạt động của RESTful gồm 4 phương thức GET/ PUT/ POST/ DELETE. Những phương thức này được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, đọc, sửa, xóa.Mỗi phương thức sẽ gọi đến server thông qua API để chỉ thị cho server phải làm gì.

* 1. CƠ SỞ DỮ LIỆU NO SQL
  2. JSON WEB TOKEN
  3. NOTIFICATION
  4. NGÔN NGỮ PHP

1. **TRIỂN KHAI ĐỀ TÀI**
2. **TỔNG QUAN**
3. GIỚI THIỆU ĐỀ TÀI

Thương mại điện tử hiện nay đang rất phổ biến và dần trở thành xu thế mua sắm cho tương lai. Và để gia nhập xu thế này, các chuỗi cửa hàng ăn uống cần phần mềm trung gian để quảng bá cũng như giao tiếp với khách hàng trên nền tảng trực tuyến. Và để tạo ra được ứng dụng ấy, đề tài này được ra đời. Với mục đích cuối cùng là tạo ra được một hệ thống ứng dụng trực tuyến, sử dụng và hoạt động tốt trong thực tế, nhóm chúng tôi sẽ khảo sát thực tế về việc vận hành của các ứng dụng vận chuyển tương tự như Grab, GoViet,… Sau đó là phân tích hệ thống và cuối cùng là lập trình ứng dụng.

Các vấn đề đặt ra cho đề tài bao gồm:

* Tìm hiểu về lập trình Android, NodeJs, MongoDB
* Khảo sát, vạch ra các quy trình nghiệp vụ trên ứng dụng.
* Phân tích cơ sở dữ liệu cho ứng dụng
* Thiết kế hoàn thiện cơ sở dữ liệu, chức năng cửa hệ thống và giao diện.
* Xây dựng hệ thống API cho phép giao tiếp với dữ liệu và xử lý nghiệp vụ

Phần lập trình android sẽ được thực hiện ở giai đoạn tiếp theo của đề tài. Sau khi hoàn thiện, FoodNow sẽ là một phần mềm trung gian giữa khách hàng và cửa hàng. Cho phép các cửa hàng quản lý việc nhận đơn hàng, quản lý thực đơn hay thống kê đơn hàng của họ. Cho phép khách hàng đăng ký tài khoản, quản lý tài khoản, đặt món ăn và xem lại các đơn hàng của bản thân. Trong tương lai, có thể phát triển thêm các hệ thống tính điểm cho cả khách hàng và cửa hàng.

1. MỤC TIÊU PHẠM VI ĐỀ TÀI
   1. MỤC TIÊU
   2. PHẠM VI
2. KHẢO SÁT HỆ THỐNG
   1. GIỚI THIỆU SƠ LƯỢC HỆ THỐNG

Hệ thống FoodNow dựa trên các hệ thống quản lý việc giao nhận hàng hiện nay. Điểm chung của các hệ thống này chính là có các bên trung gian là khách hàng, cửa hàng và tài xế. Vì nhóm muốn tập trung xử lý vào phần cửa hàng và khách hàng nên bên tài xế sẽ phát triển sau. Để tạo hệ thống, nhóm đã tiến hành khảo sát ứng dụng vận chuyển hàng của Grab vì hệ thống này có nhiều điểm tương đồng, dễ khảo sát và gần với thực tế.

* 1. QUI TRÌNH XỬ LÝ NGHIỆP VỤ
     + 1. SƠ ĐỒ HỆ THỐNG
       2. MÔ HÌNH NGHIỆP VỤ BPM

1. **PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**
2. PHÂN TÍCH HỆ THỐNG
   1. MÔ HÌNH ERD
      1. KHÁI NIỆM

Mô hình thực thể kết hợp là mô hình được sử dụng nhiều nhất trong việc thiết kế quan niệm dữ liệu. Mô hình thực thể kết hợp bao gồm các khái niệm thực thể, thuộc tính, mối kết hợp. Về sau để bổ sung và mở rộng hơn mô hình thực thể có thêm một số khái niệm khác như thuộc tính kết hợp, cấu trúc cây tổng quát hóa.

Các thành phần cơ bản của mô hình thực thể kết hợp (TT - KH)

Thực thể: Mục tiêu của mô hình dữ liệu là mô tả các đối tượng của thế giới thực cùng các đặc trưng của nó. Khái niệm thực thể là khái niệm được dùng trong mô hình quan niệm dữ liệu để mô tả các thực thể của thế giới thực. Thực thể biểu diễn lớp các đối tượng của thế giới thực.

Thực thể phân thành hai loại:

Đối tượng hữu hình: các đối tượng có thể quan sát được.

Đối tượng vô hình: không cảm nhận được một cách trực quan.

Thực thể được biểu diễn bằng hình hình chữ nhật.

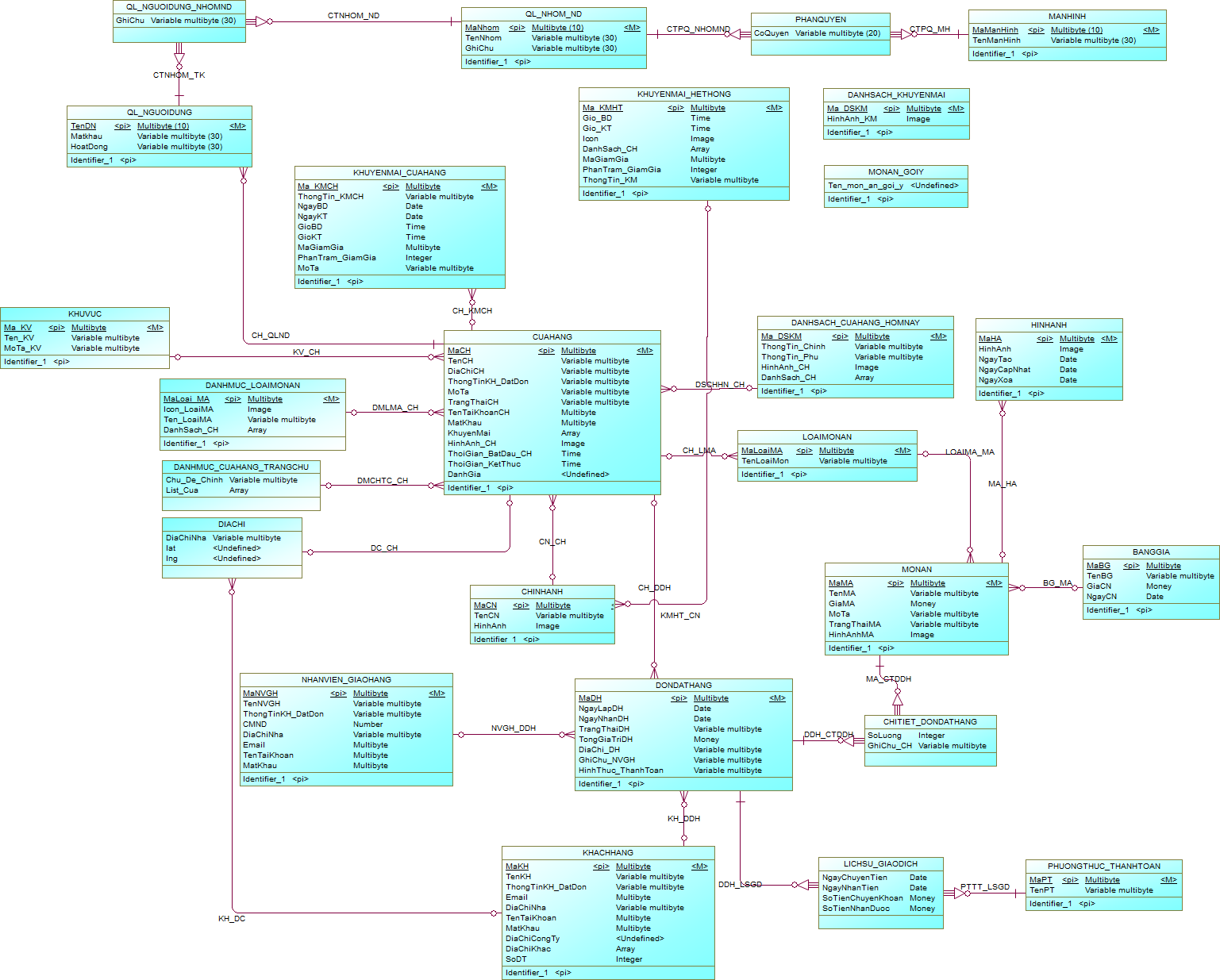
Thuộc tính: Khi mô hình hóa thế giới thực người ta thường đặc tả thực thể là các đối tượng của thế giới thực. Mỗi thực thể đều có các tính chất riêng biệt đặc trưng cho từng thực thể, các tính chất đặc trưng này chính là thuộc tính của thực thể đó. Mỗi kiểu thuộc tính là thể hiện một tập hợp tất cả những giá trị của các trường hợp của những thuộc tính đặc trưng của thực thể. Một thực thể có thể không có một thuộc tính nào cả, nhưng một thực thể như vậy ít có ý nghĩa. Do đó một thực thể có ít nhất một thuộc tính.

Khóa của một kiểu thực thể: là một hoặc tập hợp các thuộc tính không lặp mà các trường hợp của nó xác định các trường hợp cụ thể của thực thể.

Mối kết hợp: Mối kết hợp biểu diễn sự kết hợp giữa hai hay nhiều thực thể. Mối kết hợp có thể có trên nhiều thực thể. Một mối kết hợp khác là mối kết hợp giữa một thực thể với chính nó.

Mô hình thực thể kết hợp được vẽ trên phần mềm Power Designer.

* + 1. MÔ HÌNH



* 1. MÔ TẢ THỰC THỂ

1. THIẾT KẾ HỆ THỐNG
   1. SƠ ĐỒ PHÂN CẤP CHỨC NĂNG
   2. MÔ HÌNH DFD
   3. MÔ HÌNH USECASE CHỨC NĂNG
   4. THIẾT KẾ CƠ SỞ DỮ LIỆU
   5. XÂY DỰNG API
2. **THIẾT KẾ GIAO DIỆN**
3. GIAO DIỆN APP KHÁCH HÀNG
   1. ĐĂNG NHẬP, ĐĂNG KÝ
   2. TRANG CHỦ
   3. CỬA HÀNG MÓN ĂN
   4. ĐƠN HÀNG
   5. THÔNG TIN KHÁCH HÀNG
4. GIAO DIỆN WEB ADMIN
   1. QUẢN LÝ TÀI KHOẢN ĐĂNG KÝ
   2. QUẢN LÝ CHI NHÁNH CỬA HÀNG
   3. QUẢN LÝ KHUYẾN MÃI
   4. QUẢN LÝ DANH MỤC
   5. BÁO CÁO
5. **TỔNG KẾT**