**BỘ CÔNG THƯƠNG**

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

**KHOA CÔNG NGHỆ THÔNG TIN**

**🙜🙢🙠🙞

**KHÓA LUẬN TỐT NGHIỆP**

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

**GVHD:**

Ths. Lâm Thị Họa Mi

Ths. Vũ Đình Ái

**SINH VIÊN THỰC HIỆN:**

Kiều Hữu Thành-2001160343

Lê Thị Ngọc Hiền-2001160222

**CHUYÊN NGÀNH:** Công nghệ phần mềm và Hệ thống thông tin

**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**

**🙜🙢🙠🙞

**KHÓA LUẬN TỐT NGHIỆP**

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

**GVHD:**

Ths. Lâm Thị Họa Mi

Ths. Vũ Đình Ái

**SINH VIÊN THỰC HIỆN:**

Kiều Hữu Thành-2001160343

Lê Thị Ngọc Hiền-2001160222

**CHUYÊN NGÀNH:** Công nghệ phần mềm và Hệ thống thông tin

**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 em 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.

Trong đề tài này, nhóm đã tiến hành chọn:

* Công nghệ lập trình là Nodejs và MongoDB.
* Thiết kế cơ sở dữ liệu bằng ngôn ngữ NoSQL.
* Thiết kế giao diện dành cho khách hàng là Android.
* Giao diện quản lý Admin, cửa hàng là ngôn ngữ lập trình PHP.
* Tích hợp Google API, Facebook API để đăng ký, đăng nhập tài khoản.
* Ứng dụng Google Map để lấy địa chỉ, tính khoảng cách.
* Ứng dụng Json Web Token để mã hóa tài khoản đăng nhập.
* Ứng dụng Firebase cho việc thông báo đơn hàng, thông tin khuyến mãi.

Từ việc xây dựng hệ thống trên, nhóm sẽ trình bày theo 3 phần như sau:

Phần 1: Cơ sở lý thuyết.

Phần 2: Triển khai đề tài.

Phần 3: Tổng kết.

**MỤC LỤC**

[**LỜI CẢM ƠN** 3](#_Toc45745750)

[**LỜI MỞ ĐẦU** 4](#_Toc45745751)

[**DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT** 7](#_Toc45745752)

[**DANH MỤC HÌNH VẼ VÀ ĐỒ THỊ** 8](#_Toc45745753)

[**PHẦN 1:** **CƠ SỞ LÝ THUYẾT** 9](#_Toc45745754)

[1.1. TÌM HIỂU CÔNG NGHỆ 9](#_Toc45745755)

[1.1.1. NODEJS 9](#_Toc45745756)

[1.1.2. MONGOBD 9](#_Toc45745757)

[1.1.3. FIREBASE 9](#_Toc45745758)

[1.1.4. HỆ THỐNG RESTFUL API 10](#_Toc45745759)

[1.1.5. JSON WEB TOKEN 12](#_Toc45745760)

[1.1.6. GOOGLE API 13](#_Toc45745761)

[1.1.7. FACEBOOK API 13](#_Toc45745762)

[1.2. LÝ THUYẾT PHÂN TÍCH 13](#_Toc45745763)

[1.2.1. CƠ SỞ DỮ LIỆU NOSQL 13](#_Toc45745764)

[1.2.2. MÔ HÌNH ERD 14](#_Toc45745765)

[1.2.3. MÔ HÌNH USECASE 15](#_Toc45745766)

[1.2.4. MÔ HÌNH DFD 15](#_Toc45745767)

[**PHẦN 2:** **TRIỂN KHAI ĐỀ TÀI** 17](#_Toc45745768)

[**CHƯƠNG 1:** **TỔNG QUAN** 17](#_Toc45745769)

[1.1. GIỚI THIỆU ĐỀ TÀI 17](#_Toc45745770)

[1.2. MỤC TIÊU PHẠM VI ĐỀ TÀI 17](#_Toc45745771)

[1.2.1. MỤC TIÊU 17](#_Toc45745772)

[1.2.2. PHẠM VI 18](#_Toc45745773)

[1.3. KHẢO SÁT HỆ THỐNG 18](#_Toc45745774)

[1.4. QUI TRÌNH XỬ LÝ NGHIỆP VỤ 18](#_Toc45745775)

[1.4.1. SƠ ĐỒ HỆ THỐNG 18](#_Toc45745776)

[1.4.2. MÔ HÌNH NGHIỆP VỤ BPM 19](#_Toc45745777)

[**CHƯƠNG 2:** **PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 25](#_Toc45745778)

[1. PHÂN TÍCH HỆ THỐNG 25](#_Toc45745779)

[1.1. MÔ HÌNH ERD 25](#_Toc45745780)

[1.2. MÔ TẢ THỰC THỂ 25](#_Toc45745781)

[2. THIẾT KẾ HỆ THỐNG 25](#_Toc45745782)

[2.1. SƠ ĐỒ PHÂN CẤP CHỨC NĂNG 25](#_Toc45745783)

[2.2. MÔ HÌNH DFD 26](#_Toc45745784)

[2.2.1. Mô hình mức ngữ cảnh 26](#_Toc45745785)

[2.2.2. Mô hình mức đỉnh 26](#_Toc45745786)

[2.2.3. Mô hình mức dưới đỉnh 27](#_Toc45745787)

[2.3. MÔ HÌNH USECASE CHỨC NĂNG 29](#_Toc45745788)

[2.3.1. USECASE KHÁCH HÀNG 29](#_Toc45745789)

[2.3.2. USECASE CỬA HÀNG 30](#_Toc45745790)

[2.3.3. USECASE QUẢN TRỊ VIÊN 31](#_Toc45745791)

[2.4. THIẾT KẾ CƠ SỞ DỮ LIỆU 32](#_Toc45745792)

[2.5. XÂY DỰNG API 39](#_Toc45745793)

[**CHƯƠNG 3:** **THIẾT KẾ GIAO DIỆN** 40](#_Toc45745794)

[3.1. GIAO DIỆN APP KHÁCH HÀNG 40](#_Toc45745795)

[3.1.1. ĐĂNG NHẬP, ĐĂNG KÝ 40](#_Toc45745796)

[3.1.2. TRANG CHỦ 41](#_Toc45745797)

[3.1.3. CỬA HÀNG MÓN ĂN 44](#_Toc45745798)

[3.1.4. ĐƠN HÀNG 48](#_Toc45745799)

[3.1.5. THÔNG TIN KHÁCH HÀNG 50](#_Toc45745800)

[3.2. GIAO DIỆN WEB ADMIN 52](#_Toc45745801)

[3.2.1. ĐĂNG NHẬP 52](#_Toc45745802)

[3.2.2. QUẢN LÝ TÀI KHOẢN ĐĂNG KÝ 53](#_Toc45745803)

[3.2.3. QUẢN LÝ CHI NHÁNH CỬA HÀNG 54](#_Toc45745804)

[3.2.4. THÔNG TIN TÀI KHOẢN CỬA HÀNG 54](#_Toc45745805)

[3.2.5. QUẢN LÝ KHUYẾN MÃI 55](#_Toc45745806)

[3.2.6. QUẢN LÝ DANH MỤC 55](#_Toc45745807)

[a) DANH MỤC LOẠI MÓN ĂN 55](#_Toc45745808)

[b) DANH MỤC CỬA HÀNG HÔM NAY 56](#_Toc45745809)

[c) DANH MỤC CỬA HÀNG TRANG CHỦ 57](#_Toc45745810)

[3.2.7. BÁO CÁO THỐNG KÊ 58](#_Toc45745811)

[3.3. GIAO DIỆN WEB CỬA HÀNG 58](#_Toc45745812)

[3.3.1. ĐĂNG NHẬP 58](#_Toc45745813)

[3.3.2. DANH SÁCH MÓN ĂN 59](#_Toc45745814)

[3.3.3. DANH SÁCH ĐƠN HÀNG 60](#_Toc45745815)

[3.3.4. DANH SÁCH KHUYẾN MÃI 60](#_Toc45745816)

[3.3.5. TÀI KHOẢN 61](#_Toc45745817)

[3.3.6. ĐƠN HÀNG THỐNG KÊ 61](#_Toc45745818)

[**PHẦN 3:** **TỔNG KẾT** 62](#_Toc45745819)

[**3.1.** **KẾT QUẢ ĐẠT ĐƯỢC** 62](#_Toc45745820)

[**3.1.1.** **VỀ PHẦN KIẾN THỨC** 62](#_Toc45745821)

[**3.1.2.** **VỀ SẢN PHẨM** 62](#_Toc45745822)

[**3.2.** **ĐIỂM HẠN CHẾ** 62](#_Toc45745823)

[**3.3.** **HƯỚNG PHÁT TRIỂN** 62](#_Toc45745824)

[**TÀI LIỆU THAM KHẢO** 63](#_Toc45745825)

# **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**

[Hình 1.1 Vòng đời của một Activity 11](file:///D:\HUFI_IT_SYSTEM_TECHNOLOGY\IT_FOODNOW_IMAGE\xxamp\htdocs\FoodNow\Foodnow_KLTN.docx#_Toc45751308)

[Hình 1.2 Dữ liệu trên Database Firebase có định dạng Json kiểu Gson 13](#_Toc45751309)

[Hình 1.3 Mô hình hoạt động của RESTful API 15](file:///D:\HUFI_IT_SYSTEM_TECHNOLOGY\IT_FOODNOW_IMAGE\xxamp\htdocs\FoodNow\Foodnow_KLTN.docx#_Toc45751310)

[Hình 1.4 Sơ đồ hệ thống FoodNow 22](#_Toc45751311)

[Hình 1.5 Quy trình khách hàng đăng nhập hệ thống 23](#_Toc45751312)

[Hình 1.6 Qui trình khách hàng đặt hàng 24](#_Toc45751313)

[Hình 1.7 Qui trình giao hàng 27](#_Toc45751314)

[Hình 1.8 Qui trình nhận hàng 27](#_Toc45751315)

[Hình 1.9 Mô hình ERD 28](#_Toc45751316)

[Hình 1.10 Mô hình phân cấp chức năng 29](file:///D:\HUFI_IT_SYSTEM_TECHNOLOGY\IT_FOODNOW_IMAGE\xxamp\htdocs\FoodNow\Foodnow_KLTN.docx#_Toc45751317)

[Hình 1.11 Mô hình mức ngữ cảnh 29](#_Toc45751318)

[Hình 1.12 Mô hình mức đỉnh 30](#_Toc45751319)

[Hình 1.13 Mô hình quản lý khách hàng 31](#_Toc45751320)

[Hình 1.14 Mô hình quản lý đặt hàng 31](#_Toc45751321)

[Hình 1.15 Mô hình quản lý hủy đơn hàng 32](#_Toc45751322)

[Hình 1.16 Mô hình quản lý cửa hàng 32](#_Toc45751323)

[Hình 1.17 Mô hình usecase khách hàng 33](#_Toc45751324)

[Hình 1.18 Mô hình usecase cửa hàng 34](#_Toc45751325)

[Hình 1.19 Mô hình usecase quản trị viên 35](#_Toc45751326)

[Hình 1.20 Cơ sở dữ liệu API 36](#_Toc45751327)

[Hình 1.21 Giao diện mở đầu hệ thống 44](#_Toc45751328)

[Hình 1.22 Giao diện đăng ký, đăng nhập 44](#_Toc45751329)

[Hình 1.23 Giao diện danh mục loại món ăn trang chủ 46](#_Toc45751330)

[Hình 1.24 Giao diện trang chủ phần đầu 46](#_Toc45751331)

[Hình 1.25 Giao diện trang chủ phần giữa 47](#_Toc45751332)

[Hình 1.26 Giao diện tìm kiếm món ăn 47](#_Toc45751333)

[Hình 1.27 Giao diện phần cuối trang chủ 48](#_Toc45751334)

[Hình 1.28 Giao diện danh sách cửa hàng tìm kiếm 49](#_Toc45751335)

[Hình 1.29 Giao diện danh sách cửa hàng gần tôi 49](#_Toc45751336)

[Hình 1.30 Giao diện danh sách cửa hàng đặt nhiều 50](#_Toc45751337)

[Hình 1.31 Giao diện thông tin cửa hàng cùng món ăn 50](#_Toc45751338)

[Hình 1.32 Giao diện danh sách loại món ăn và món ăn của cửa hàng 51](#_Toc45751339)

[Hình 1.33 Giao diện địa chỉ cửa hàng 51](#_Toc45751340)

[Hình 1.34 Giao diện món ăn trong giỏ hàng 52](#_Toc45751341)

[Hình 1.35 Giao diện ghi chú món ăn 52](#_Toc45751342)

[Hình 1.36 Giao diện đơn đặt hàng 53](#_Toc45751343)

[Hình 1.37 Giao diện chọn thời gian 53](#_Toc45751344)

[Hình 1.38 Giao diện thay đổi địa chỉ 54](#_Toc45751345)

[Hình 1.39 Giao diện chọn đặt đơn khi đã kiểm tra thông tin đầy đủ 54](#_Toc45751346)

[Hình 1.40 Giao diện đăng nhập khi khách hàng đăng xuất 58](#_Toc45752904)

[Hình 1.41 Giao diện khách hàng đăng nhập thành công 58](#_Toc45752905)

[Hình 1.42 Giao diện thông tin của khách hàng 59](#_Toc45752906)

[Hình 1.43 Giao diện chỉnh sửa địa chỉ 59](#_Toc45752907)

[Hình 1.44 Giao diện quản trị đăng nhập 60](#_Toc45752908)

[Hình 1.45 Giao diện tổng quan 60](#_Toc45752909)

[Hình 1.46 Giao diện đăng ký cho cửa hàng 61](#_Toc45752910)

[Hình 1.47 Giao diện quản lý chi nhánh cửa hàng 61](#_Toc45752911)

[Hình 1.48 Giao diện quản lý thông tin cửa hàng 62](#_Toc45752912)

[Hình 1.49 Giao diện quản lý khuyến mãi 63](#_Toc45752913)

[Hình 1.50 Giao diện quản lý danh mục loại món ăn 63](#_Toc45752914)

[Hình 1.51 Giao diện cửa hàng hôm nay 64](#_Toc45752915)

[Hình 1.52 Giao diện thêm cửa hàng vào từng nhóm thuộc cửa hàng hôm nay 64](#_Toc45752916)

[Hình 1.53 Giao diện cửa hàng trang chủ 65](#_Toc45752917)

[Hình 1.54 Giao diện thêm cửa hàng vào từng nhóm thuốc cửa hàng trang chủ 65](#_Toc45752918)

[Hình 1.55 Giao diện cửa hàng đăng nhập 66](#_Toc45752919)

[Hình 1.56 Giao diện trang tổng quan 66](#_Toc45752920)

[Hình 1.57 Giao diện danh sách loại món ăn và món ăn 67](#_Toc45752921)

[Hình 1.58 Giao diện thêm món ăn 67](#_Toc45752922)

[Hình 1.59 Giao diện danh sách đơn hàng chi tiết 68](#_Toc45752923)

[Hình 1.60 Giao diện danh sách khuyến mãi 68](#_Toc45752924)

[Hình 1.61 Giao diện thông tin tài khoản 69](#_Toc45752925)

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.
  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

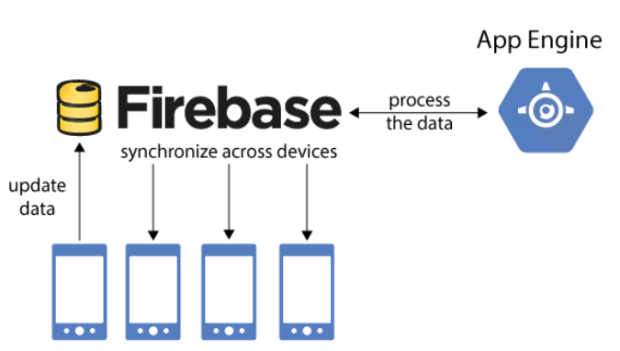
Realtime Database Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng có tương tác cơ sở dữ liệu (CRUD) một cách nhanh chóng và ngay tức thời (Realtime).

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/)

* 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.

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng. REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

REST hoạt động dựa trên giao thức HTTP. Trình tự hoạt động của chúng như sau:

* GET (SELECT): truy vấn một object
* POST (CREATE): tạo object mới
* PUT (UPDATE): cập nhật thông tin cho object
* DELETE: xóa một object

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

* 1. JSON WEB TOKEN

JSON Web Mã (JWT) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA.

Thành phần của JWT:

- Header bao gồm hai phần chính: loại token (mặc định là JWT - Thông tin này cho biết đây là một Token JWT) và thuật toán đã dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA).

- Payload chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ. Có 3 loại claims thường gặp trong Payload: reserved, public và private claims.

- **Signature:** Phần chữ ký được tạo bằng cách kết hợp 2 phần Header + Payload, rồi mã hóa nó lại bằng 1 giải thuật encode nào đó, càng phức tạp thì càng tốt, ví dụ như HMAC SHA-256.

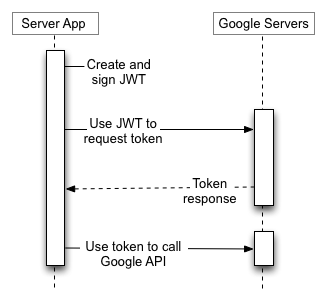
* 1. GOOGLE API

Service account xác thực đối tượng truy cập Google API’s bằng cách sử dụng OAuth trên danh nghĩa là 1 ứng dụng web thay vì là 1 người dùng. Tức là nó sử dụng chứng chỉ tài khoản dịch vụ của ứng dụng để chứng thực.

Trước khi tạo request truy cập API, cần phải có 1 access token bằng cách sử dụng các tài khoản dịch vụ đã được chứng nhận. request này là POST request với các params như grant-type và JWT(JSON Web Token) đã đăng nhập bằng key.

Các loại tham số cấp đầu tiên là 1 jet-bearer (chỉ cần 1 access token mà không cần refresh token). Param thứ 2 là JWT chứa thông tin xác thực tài khoản trong chuỗi mã hóa được định dạng trước chứa tiêu đề, tập các yêu cầu(claim set) và chữ ký. Tiêu đề định nghĩa thuật toán signin được sử dụng trong việc signing JWT, claim setchứa địa chỉ email của tài khoản dịch vụ, phạm vi, thời gian hết hạn… Chữ ký là thành phần có chữ ký của tiêu đề và claim set

Sau khi tạo request tới máy chủ ủy quyền, chúng ta nhận được một access token kéo dài ít nhất là 1 giờ. Sử dụng access token này, chúng ta có thể truy cập vào Google APIs theo cách sau : sử dụng user’s credential, cách thức như hình dưới đây :



Tiến trình trên yêu cầu ứng dụng phải tạo và ký hiệu mã hóa JWT’s, rất dễ để phạm sai lầm nghiêm trọng tác động rất lớn tới sự an toàn của ứng dụng. Thay vào đó tất cả có thể được thực hiện bởi việc sử dụng các thư viện client . Ruby chứa gem google\_api\_client cho việc lấy `access token, tạo ra các yêu cầu để truy cập vào API và phân tích kết quả.

* 1. FACEBOOK API

Facebook API là một nền tảng để xây dựng những ứng dụng cho các thành viên của mạng xã hội Facebook. API cho phép các ứng dụng sử dụng các kết nối xã hội và các thông tin hồ sơ để làm cho các ứng dụng liên quan tới nhau nhiều hơn. API cung cấp những lời gọi để lấy thông tin về người sử dụng, nhóm người dùng, bạn bè, thông báo, sự kiện và nguồn cấp dữ liệu…Cũng có thể dùng lời gọi API để cập nhật và lấy thông tin hồ sơ.

Facebook gửi phương thức POST tới máy chủ Facebook API. Nó bao gồm một số các thông số yêu cầu như api\_key của ứng dụng. Session\_key của người dùng đưa ra yêu cầu. Bên cạnh đó Facebook còn thêm vào tham số fb\_sig để thông báo ứng dụng đưa ra yêu cầu. Bằng cách này tất cả các lời gọi API sẽ được đảm bảo, Facebook có thể xác minh các yêu cầu được gửi từ một ứng dụng đã được chấp thuận. Thông tin mà Facebook sẽ trả lại là một tài liệu XML.

1. LÝ THUYẾT PHÂN TÍCH
2. CƠ SỞ DỮ LIỆU NOSQL

NoSql là một khái niệm chỉ về một lớp các hệ cơ sở dữ liệu không sử dụng mô hình quan hệ. (RDBMS). RDBMS vốn tồn tại khá nhiều nhược điểm như có hiệu năng không tốt nếu kết nối dữ liệu nhiều bảng lại hay khi dữ liệu trong một bảng là rất lớn.

NoSQL ra đời năm 1998 bởi Carlo Strozzi khi ông lập mới một hệ cơ sở dữ liệu quan hệ mã nguồn mở nhanh và nhẹ không liên quan đến SQL Vào năm 2009, Eric Evans, nhân viên của Rackspace giới thiệu lại thuật ngữ NoSql khi Johan Oskarsson của Last.fm muốn tổ chức một hội thảo về cơ sở dữ liệu nguồn mở phân tán. Thuật ngữ NoSql đánh dấu bước phát triển của thế hệ cơ sở dữ liệu mới: phân tán (distributed) + không ràng buộc (non-relational).

Về cơ bản, thiết kế của NoSQL lựa chọn mô hình lưu trữ dữ liệu theo cặp giá trị ‘key-value’. Khái niệm node được sử dụng như trong các hệ thống phân tán. Việc lưu trữ dữ liệu có thể chấp nhận trùng lặp, tức dữ liệu được nhân bản ra nhiều node khác nhau. Một yêu cầu truy vấn dữ liệu có thể gửi tới nhiều máy cùng lúc, khi một máy nào đó bị sự cố cũng không ảnh hưởng nhiều tới toàn bộ hệ thống. Để đảm bảo tính ‘thời gian thực’ trong các hệ thống xử lý lượng lớn, thông thường người ta sẽ tách biệt CSDL ra làm 2 hoặc nhiều CSDL. CSDL nhỏ đảm bảo đọc-ghi liên tục, khi đạt tới ngưỡng thời gian hoặc dung lượng, CSDL nhỏ sẽ được gộp vào CSDL lớn có thiết kế tối ưu cho tác vụ đọc.

1. MÔ HÌNH ERD

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 USECASE

Use case là một kỹ thuật được dùng trong [kỹ thuật phần mềm](https://vi.wikipedia.org/wiki/K%E1%BB%B9_thu%E1%BA%ADt_ph%E1%BA%A7n_m%E1%BB%81m) và [hệ thống](https://vi.wikipedia.org/wiki/K%E1%BB%B9_thu%E1%BA%ADt_h%E1%BB%87_th%E1%BB%91ng) để nắm bắt [yêu cầu chức năng](https://vi.wikipedia.org/w/index.php?title=Y%C3%AAu_c%E1%BA%A7u_ch%E1%BB%A9c_n%C4%83ng&action=edit&redlink=1) của hệ thống. Use case mô tả sự tương tác đặc trưng giữa người dùng bên ngoài ([actor](https://vi.wikipedia.org/w/index.php?title=Actor_(UML)&action=edit&redlink=1)) và hệ thống. Nó thể hiện ứng xử của hệ thống đối với bên ngoài, trong một hoàn cảnh nhất định, xét từ quan điểm của người sử dụng[[1]](https://vi.wikipedia.org/wiki/Use_case#cite_note-Jaco-1). Nó mô tả các yêu cầu đối với hệ thống, có nghĩa là những gì hệ thống phải làm chứ không phải mô tả hệ thống làm như thế nào. Tập hợp tất cả Use case của hệ thống sẽ mô tả tất cả các trường hợp mà hệ thống có thể được sử dụng.

Mỗi use case mô tả cách thức [actor](https://vi.wikipedia.org/w/index.php?title=Actor_(UML)&action=edit&redlink=1) tương tác với hệ thống để đạt được mục tiêu nào đó. Một hoặc nhiều [kịch bản](https://vi.wikipedia.org/w/index.php?title=K%E1%BB%8Bch_b%E1%BA%A3n_(m%C3%A1y_t%C3%ADnh)&action=edit&redlink=1) (*scenario*) có thể được tạo ra từ mỗi use case, tương ứng với chi tiết về mỗi cách thức đạt được mục tiêu nào đó. Khi mô tả Use case, người ta thường tránh dùng thuật ngữ kỹ thuật, thay vào đó họ sử dụng ngôn ngữ của [người dùng cuối](https://vi.wikipedia.org/wiki/Ng%C6%B0%E1%BB%9Di_d%C3%B9ng_cu%E1%BB%91i) hoặc [chuyên gia về lĩnh vực](https://vi.wikipedia.org/w/index.php?title=Chuy%C3%AAn_gia_v%E1%BB%81_l%C4%A9nh_v%E1%BB%B1c&action=edit&redlink=1) đó. Để tạo ra use case, cần phải có sự hợp tác chặt chẽ giữa [người phân tích hệ thống](https://vi.wikipedia.org/w/index.php?title=Ng%C6%B0%E1%BB%9Di_ph%C3%A2n_t%C3%ADch_h%E1%BB%87_th%E1%BB%91ng&action=edit&redlink=1) và người dùng cuối. Một trong những cách biểu diễn trực quan phổ biến hiện nay là [lược đồ use case](https://vi.wikipedia.org/w/index.php?title=L%C6%B0%E1%BB%A3c_%C4%91%E1%BB%93_use_case&action=edit&redlink=1) của [UML](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_m%C3%B4_h%C3%ACnh_h%C3%B3a_th%E1%BB%91ng_nh%E1%BA%A5t).

1. MÔ HÌNH DFD

Mô hình dòng dữ liệu theo trường phái phân tích có cấu trúc, chú ý nhiều đến xử lý hơn là dữ liệu. Mô hình dòng dữ liệu được tiếp cận chủ yếu theo hướng từ trên xuống, xem các hoạt động như hộp đen và quan tâm đến việc lưu trữ và xử lý thông tin giữa các hộp đen này.

Gồm các phương pháp chính như: Tom DeMacro, Yourdon & Constantine, Gane & Sarson

Các khái niệm trong mô hình:

Ô xử lý (Process): một xử lý là một hoạt động bên trong hệ thống thông tin, bao gồm:

Tạo mới thông tin

Sử dụng thông tin

Cập nhập thông tin

Hủy bỏ thông tin

Tên ô xử lý thường là một động từ.

Xử lý sẽ làm chuyển đổi dữ liệu của các dòng dữ liệu vào thành dòng dữ liệu ra.

Dòng dữ liệu (Data Flow): Dòng dữ liệu biểu diễn sự di chuyển từ thành phần này đến thành phần kia của mô hình dòng dữ liệu. Các thành phần có thể là xử lý, kho dữ liệu và đầu cuối.

Tên của dòng dữ liệu cho biết nội dung của dữ liệu di chuyển bên trong dòng dữ liệu, tên của dòng dữ liệu thông thường là cụm danh từ. Dòng dữ liệu được biểu diễn đồ họa bằng một cung với mũi tên cho biết hướng di chuyển của dòng dữ liệu.

Kho dữ liệu (Data Store): Kho dữ liệu biểu diễn vùng chứa thông tin bên trong hệ thống thông tin. Các tập tin trung gian, tập hồ sơ, bảng tra cứu, tập phiếu bằng giấy, các tạp tin lưu trữ... Có thể được biểu diễn bởi kho dữ liệu. Kho dữ liệu được biểu diễn đồ họa bằng hai đường thẳng song song.

Tên kho dữ liệu là danh từ hoặc cụm danh từ.

Đầu cuối (Terminator): Một đầu cuối là một thực thể bên ngoài độc lập với hệ thống mà có quan hệ giao tiếp với hệ thống thông tin.

Tên đầu cuối là danh từ hoặc cụm danh từ.

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 em 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.
* Tích hợp Google API, Facebook API để đăng ký, đăng nhập tài khoản.
* Ứng dụng Firebase để thông báo đơn hàng, khuyến mãi.
* Ứng dụng Google Map để tìm địa chỉ, tính khoảng cách.
* 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 bằng NoSQL, chức năng hệ thống.
* Thiết kế giao diện dành cho khách hàng, cửa hàng và quản trị viên.
* Xây dựng hệ thống API cho phép giao tiếp với dữ liệu và xử lý nghiệp vụ.

Hệ thống FoodNow mà nhóm hoàn thành 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 danh sách món ăn, quản lý khuyến mãi hay thống kê đơn hàng của họ. Cho phép khách hàng đăng ký, đăng nhập tài khoản, quản lý tài khoản của mình, quản lý địa chỉ, đặt món ăn và xem lại các đơn hàng đã đặt, đã hủy.

1. MỤC TIÊU PHẠM VI ĐỀ TÀI
   1. MỤC TIÊU

Nhóm xây dựng hệ thống giao đồ ăn FoodNow trên nền tảng NodeJs và MongoDB từ đó thiết kế App người dùng và Web dành cho quản trị và cửa hàng gồm các chức năng:

* App dành cho người dùng:
  + Người dùng đăng nhập tài khoản bằng google hoặc facebook.
  + Người dùng xem, cập nhật thông tin.
  + Tìm kiếm cửa hàng, loại món ăn và món ăn.
  + Tìm kiếm và chọn vị trí giao hàng trên bản đồ.
  + Lựa chọn được thời gian giao hàng.
  + Thực hiện việc đặt hàng sản phẩm.
  + Xem lịch sử mua hàng (đơn đặt, đơn đã hủy).
  + Hiển thị thông báo khuyến mãi.
* Website dành cho quản trị viên:
  + Quản lý danh sách cửa hàng, chi nhánh
  + Quản lý thông tin tài khoản cửa hàng
  + Quản lý đăng ký cửa hàng.
  + Quản lý các danh mục
  + Quản lý khuyến mãi
  + Thống kê báo cáo
* Website dành cho cửa hàng:
  + Quản lý danh sách món ăn
  + Quản lý khuyến mãi
  + Quản lý danh sách đơn hàng
  + Đơn hàng thống kê
  1. PHẠM VI

Đề tài dựa trên ứng dụng Foodnow để thực hiện gồm 2 phần chính là giao diện người dùng (thiết kế App) và giao diện quản trị hệ thống Foodnow (thiết kế Web).

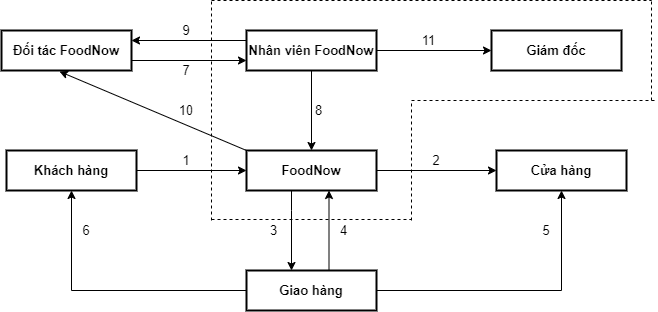
Đề tài được thực hiện theo phạm vi những mục tiêu nêu trên.

Do phạm vi và khả năng nên nhóm chỉ khảo sát và phân tích đề tài trong một giới hạn cho phép.

1. KHẢO SÁT 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. Đề tài này nhóm tập trung xử lý vào phần cửa hàng và khách hàng. 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.

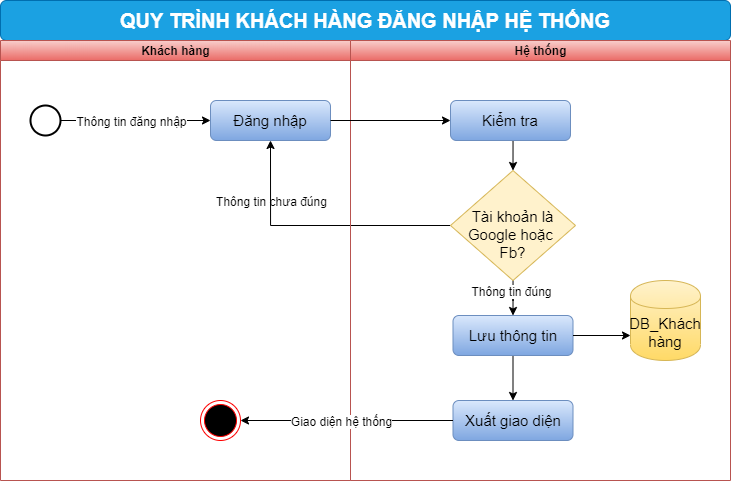
1. QUI TRÌNH XỬ LÝ NGHIỆP VỤ
   * 1. SƠ ĐỒ HỆ THỐNG



Diễn giải:

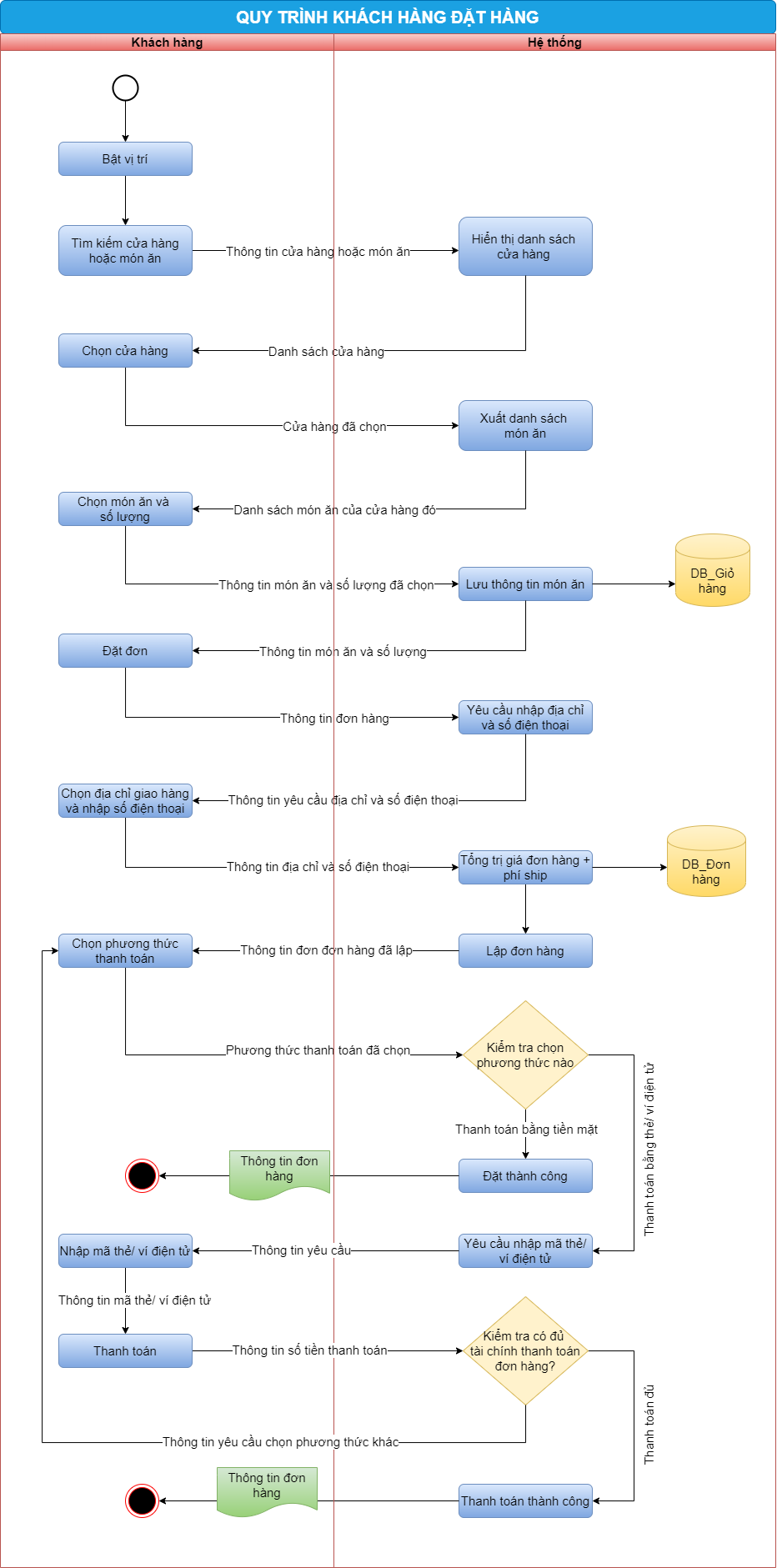
* + - 1. Khách hàng đặt hàng thông qua hệ thống FoodNow.
      2. FoodNow thông báo đến cửa hàng đơn hàng mới từ khách hàng.
      3. FoodNow thông báo đến nhân viên giao hàng nhận đơn.
      4. Nhân viên giao hàng xác nhận đơn.
      5. Nhân viên giao hàng đến cửa hàng lấy hàng.
      6. Nhân viên giao hàng đi giao hàng cho khách.
      7. Đối tác FoodNow (Cửa hàng và nhân viên giao hàng) đăng ký làm đối tác, gọi đến nhân viên hệ thống sẽ đăng ký tài khoản hoặc tự lên trang FoodNow để đăng ký làm đối tác.
      8. Nhân viên vào hệ thống để đăng ký.
      9. Nhân viên hẹn ngày ký hợp đồng.
      10. Cung cấp tài khoản cho đối tác
      11. Nhân viên tổng hợp báo cáo giám đốc.
    1. MÔ HÌNH NGHIỆP VỤ BPM
       - 1. Quy trình đăng nhập phía khách hàng

Khách hàng đăng nhập hệ thống FoodNow bằng hai tài khoản là Google hoặc Facebook mà hệ thống cung cấp. Hệ thống sẽ kiểm tra nếu đúng sẽ lưu thông tin vào dữ liệu khách hàng, rồi xuất giao diện hệ thống cho khách hàng dùng. Ngược lại đăng nhập sai thì hệ thống sẽ báo đăng nhập lại.



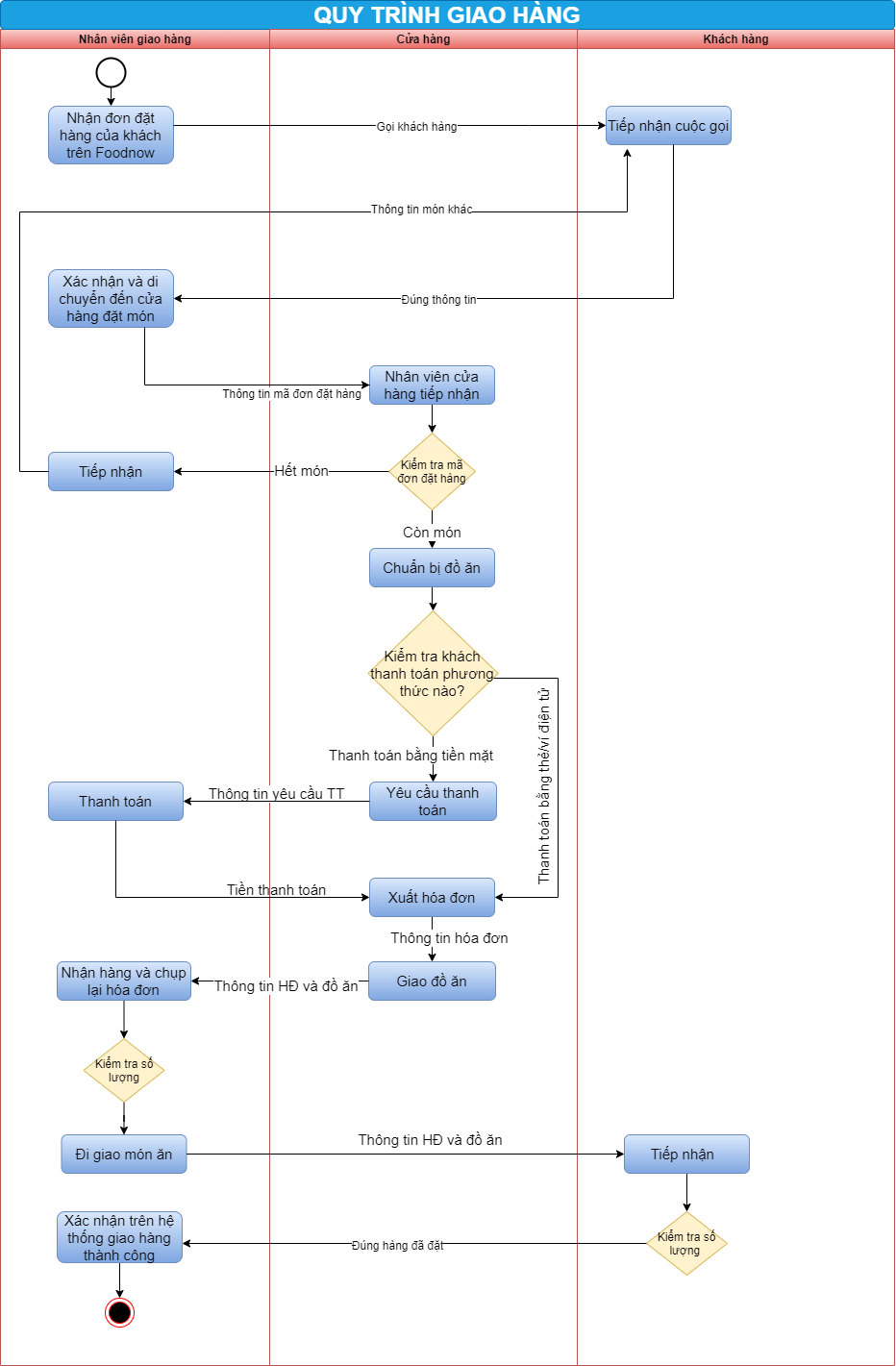
* + - * 1. Quy trình đặt hàng của khách hàng

Khách hàng sẽ bật vị trí để hệ thống xác định địa chỉ. Khách hàng tìm kiếm cửa hàng hoặc món ăn. Hệ thống sẽ hiển thị các cửa hàng mà khách hàng cần tìm kiếm. Khách hàng chọn cửa hàng và danh sách món ăn của cửa hàng hiển thị. Khách hàng chọn món ăn và món ăn đó sẽ được thêm vào giỏ hàng. Khách hàng chọn đặt đơn thì hệ thống sẽ yêu cầu cung cấp số điện thoại và địa chỉ giao hàng. Sau khi cung cấp đầy đủ thông tin thì hệ thống sẽ tính tổng giá trị đơn hàng cùng phí giao hàng và lưu vào dữ liệu đơn hàng. Khách hàng sẽ chọn phương thức thanh toán. Nếu là thanh toán tiền mặt thì hệ thống sẽ báo đặt hàng thành công. Nếu thanh toán bằng ví điện tử/ thẻ thì yêu cầu nhập mã thẻ/ ví điện tử rồi thanh toán. Hệ thống sẽ kiểm tra thanh toán đủ thì sẽ đặt hàng thành công còn nếu thanh toán thiếu sẽ quay lại bước chọn thanh toán bằng tiền mặt.

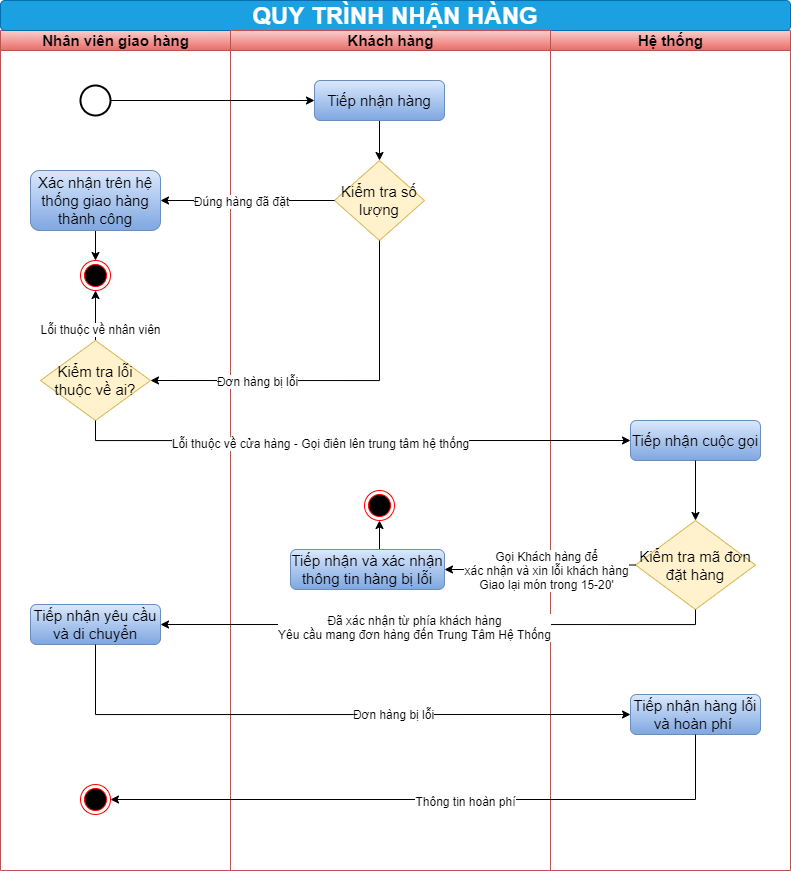


* + - * 1. Quy trình hủy đơn hàng của khách hàng
        2. Quy trình giao hàng

Nhân viên giao hàng(NVGH) nhận đơn của khách hàng trên hệ thống. NVGH sẽ gọi cho khách hàng để xác thực thông tin và di chuyển đến cửa hàng để lấy món. Nhân viên cửa hàng sẽ kiểm tra mã đơn, trường hợp món hết sẽ báo cho NVGH. NVGH sẽ gọi điện cho khách hàng chọn món khác có giá tương đương, khách hàng đồng ý. Cửa hàng chuẩn bị món. Trước khi giao món cửa hàng sẽ kiểm tra khách hàng thanh toán bằng phương thức nào, nếu là tiền mặt thì cửa hàng sẽ yêu cầu NVGH thanh toán, cửa hàng xuất hóa đơn và giao đồ ăn. Nếu là thẻ đã thanh toán rồi, cửa hàng chỉ xuất hóa đơn và giao đồ ăn. NVGH nhận món và chụp lại hóa đơn, kiểm tra lại số lượng rồi đi giao món ăn cho khách hàng. Khi khách hàng tiếp nhận, NVGH phải xác nhận trên hệ thống là giao hàng thành công thì phí giao hàng mới được thanh toán cho nhân viên.

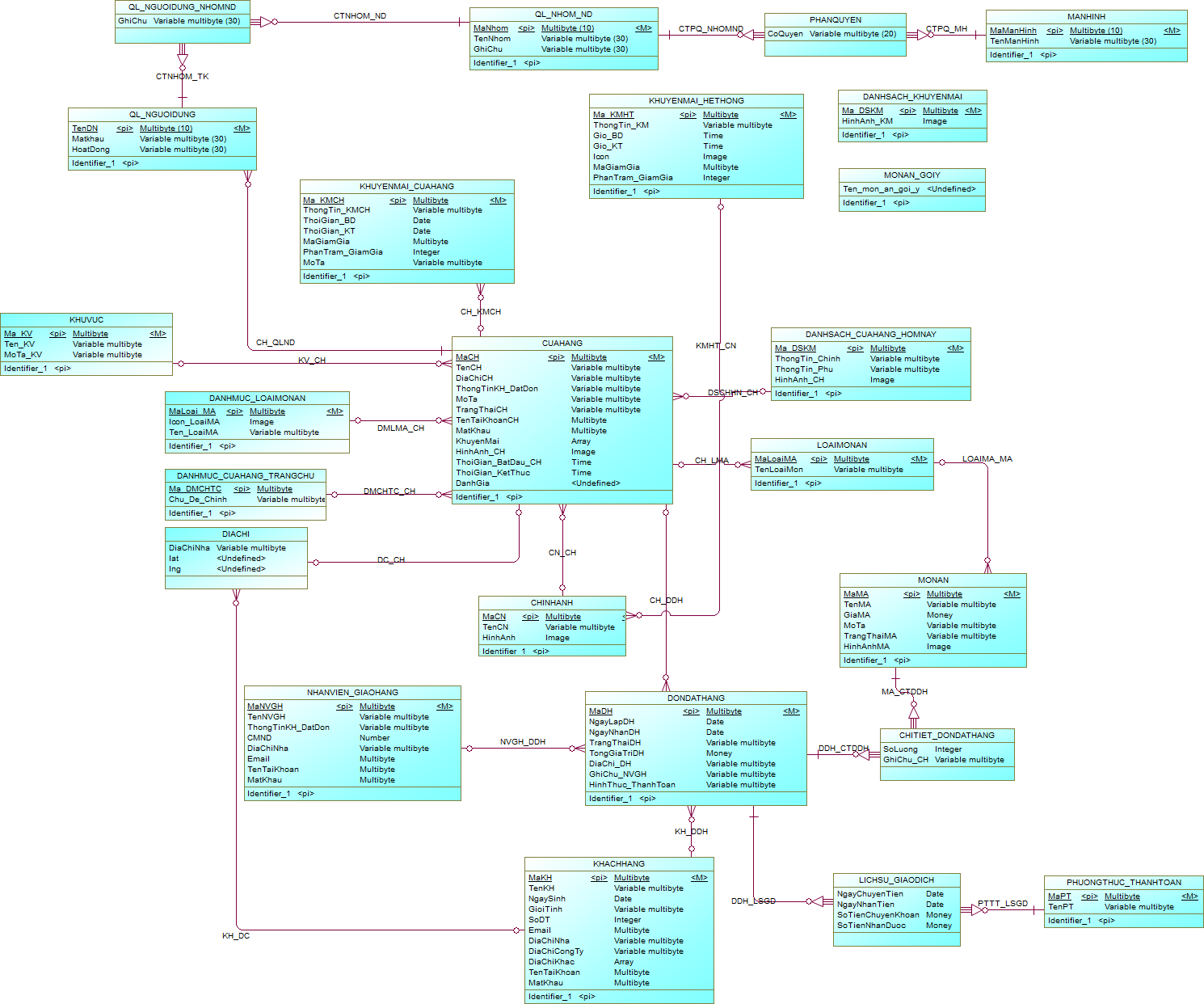


* + - * 1. Qui trình nhận hàng



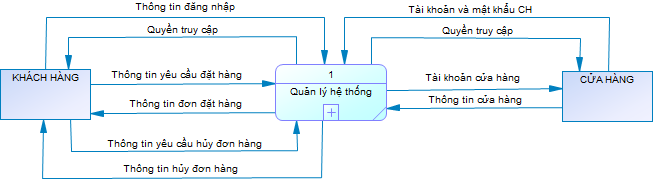
* + - * 1. Quy trình đăng ký cho phía cửa hàng.
        2. Quy trình xác nhận, hủy đơn hàng phía cửa hàng.

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

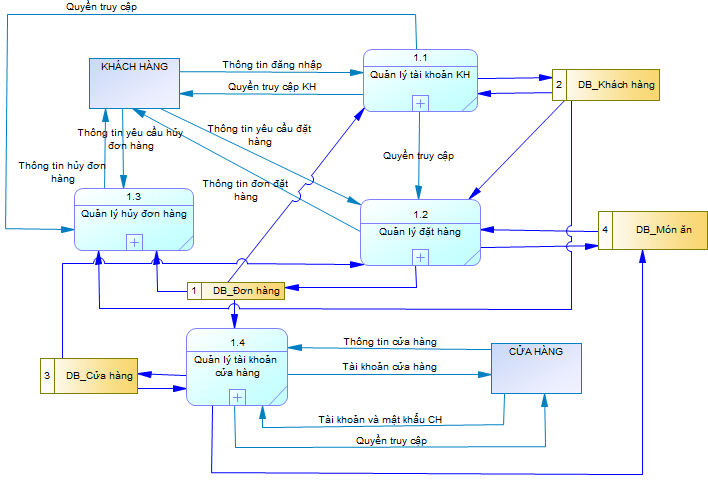


* 1. MÔ TẢ THỰC THỂ

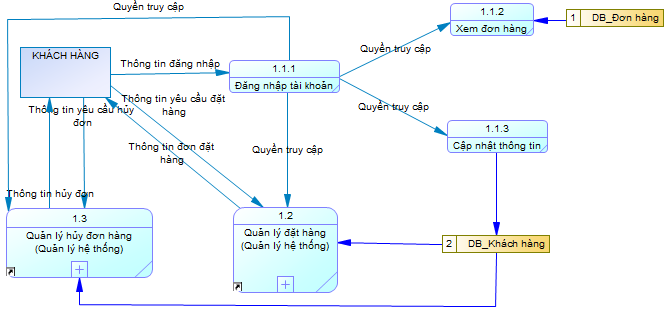
1. THIẾT KẾ HỆ THỐNG
   1. SƠ ĐỒ PHÂN CẤP CHỨC NĂNG
   2. MÔ HÌNH DFD
      1. Mô hình mức ngữ cảnh



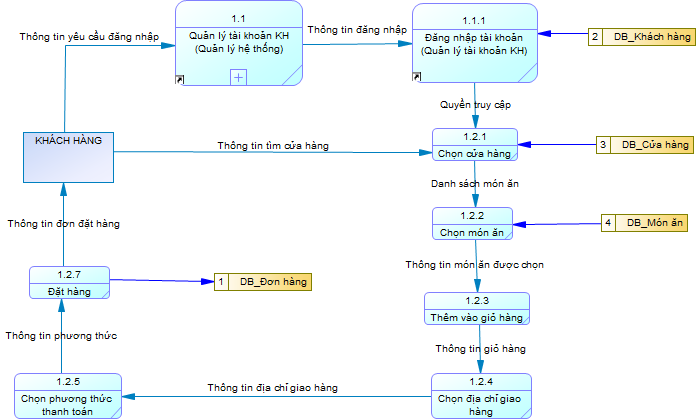
* + 1. Mô hình mức đỉnh



* + 1. Mô hình mức dưới đỉnh
       - 1. Mô hình quản lý khách hàng



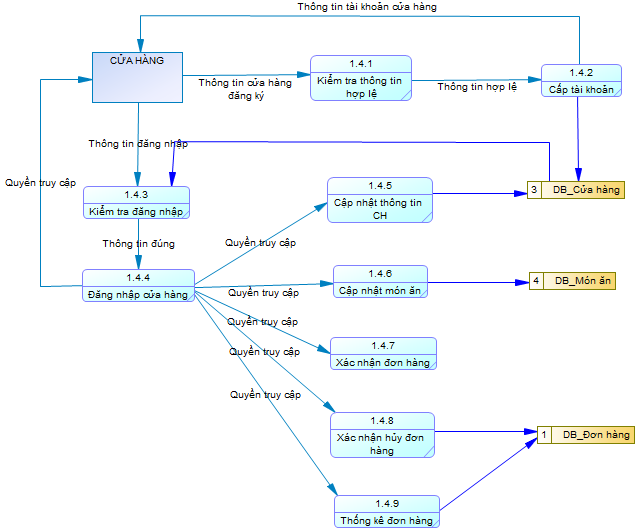
* + - * 1. Mô hình quản lý đặt hàng



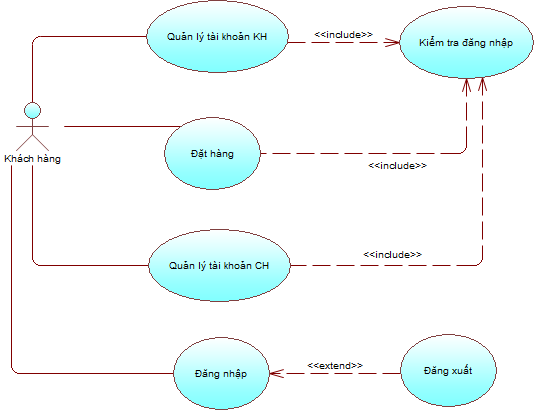
* + - * 1. Mô hình quản lý hủy đơn hàng



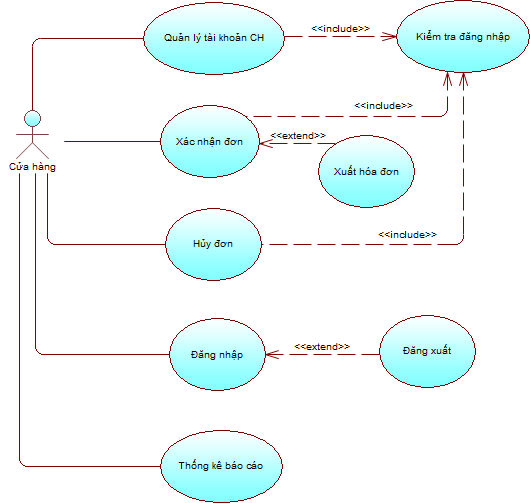
* + - * 1. Mô hình quản lý cửa hàng



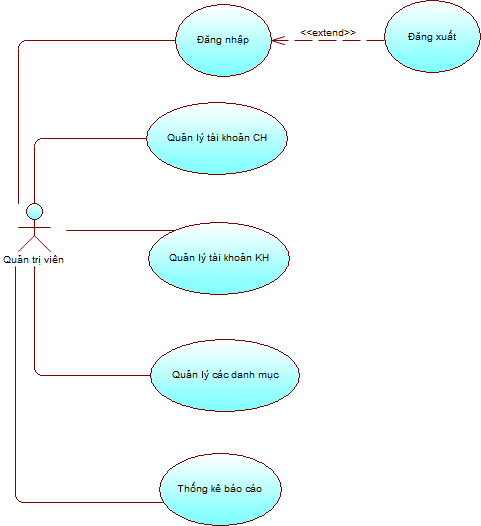
* 1. MÔ HÌNH USECASE CHỨC NĂNG
     1. USECASE KHÁCH HÀNG



* + 1. USECASE CỬA HÀNG

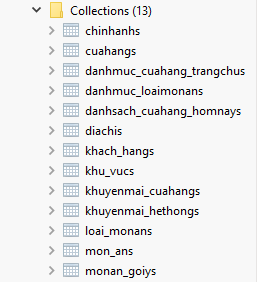


* + 1. USECASE QUẢN TRỊ VIÊN



* 1. THIẾT KẾ CƠ SỞ DỮ LIỆU

Từ việc thiết kế cơ dữ liệu bằng mô hình ERD, nhóm đã chuyển sang ngôn ngữ NoSQL để thiết kế cơ sở dữ liệu cho phù hợp với đề tài nghiên cứu.



Giải thích về các thuộc tính của từng collection trong CSDL:

* **CHI\_NHANH**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| Ten\_Chi\_Nhanh | String |  |
| Hinh\_Anh\_Chi\_Nhanh | String |  |
| Tai\_Khoan | ObjectID |  |
| DanhSach\_CH | ArrayObjectId |  |

* **CUA\_HANG**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| Ten\_Cua\_Hang | String |  |
| Mo\_Ta\_Cua\_Hang | String |  |
| Thoi\_Gian\_Bat\_Dau | String |  |
| Thoi\_Gian\_Ket\_Thuc | String |  |
| Ho\_Ten\_Nguoi\_Dai\_Dien | String |  |
| Email\_Nguoi\_Dai\_Dien | String |  |
| CMND\_Nguoi\_Dai\_Dien | String |  |
| Danh\_Gia | String |  |
| Hinh\_Anh\_Cua\_Hang | String |  |
| Trang\_Thai\_Cua\_Hang | String |  |
| Tai\_Khoan | ObjectId |  |
| Khuyen\_Mai\_CH | ArrayObjectId |  |
| Loai\_MonAn | ArrayObjectId |  |
| Thong\_Tin\_KH\_Dat\_Don | ArrayObjectId |  |

* **KHACH\_HANG**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| id\_user | ObjectId | Khóa chính |
| Ten\_khach\_hang | String |  |
| So\_dien\_thoai | String |  |
| Email | String |  |
| Ngay\_sinh | Date |  |
| Gioi\_tinh | String |  |
| Hinh\_anh\_khach\_hang | String |  |
| Dia\_chi\_nha | ObjectId |  |
| Dia\_chi\_cong\_ty | ObjectId |  |
| Dia\_chi\_khac | ArrayObjectId |  |
| Cua\_hang\_yeu\_thich | ArrayObjectId |  |
| Don\_hang\_id | ArrayObjectId |  |
| Ct\_Gio\_Hang\_id | ArrayObjectId |  |

* **LOAI\_MON\_AN**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectId | Khóa chính |
| Ten\_loai\_mon\_an | String |  |
| Danh\_sach\_mon\_an | ArrayObjectId |  |

* **MON\_AN**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectId | Khóa chính |
| Ten\_mon\_an | String |  |
| Mo\_ta\_mon\_an | String |  |
| Don\_gia\_mon\_an | Number |  |
| Hinh\_anh\_mon\_an | String |  |
| Trang\_thai\_mon\_an | String |  |
| So\_luong\_mua | Number |  |
| So\_luong\_thich | Number |  |

* **KHUYENMAI\_CUAHANG**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectId | Khóa chính |
| ThongTin\_KMCH | String |  |
| NgayBD | Date |  |
| NgayKT | Date |  |
| MaGiamGia | String |  |
| PhanTram\_GiamGia | String |  |
| MoTa | String |  |

* **KHUYENMAI\_HETHONG**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectId | Khóa chính |
| MaGiamGia | String |  |
| NgayBD | Date |  |
| NgayKT | Date |  |
| Icon | String |  |
| PhanTram\_GiamGia | String |  |
| DanhSach\_CN | ArrayObjectID |  |

* **KHU\_VUC**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| Ten\_khu\_vuc | String |  |
| Danhs\_sach\_cua\_hang | String |  |

* **DIACHI**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| Dia\_Chi | String |  |
| Vi\_do | Number |  |
| Kinh\_do | Number |  |

* **DON\_HANG**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| IdKhachHang | ObjectID |  |
| IdCuaHang | ObjectID |  |
| Ngay\_nhan\_don\_hang | Date |  |
| Dia\_chi\_giao\_hang | String |  |
| Do\_dai\_duong\_di | Number |  |
| Phi\_ship | Number |  |
| Total\_cart | Number |  |
| Total | Number |  |
| Hinh\_thuc\_thanh\_toan | String |  |
| Ghi\_chu\_NVGH | String |  |
| Trang\_thai\_don\_hang | String |  |
| Chi\_tiet\_DH | ArrayObjectID |  |

* **CHI\_TIET\_DON\_HANG**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| IdMonAn | ObjectID | Khóa chính |
| SoLuong | Number |  |
| GhiChu | GhiChu: String, |  |
| Don\_gia | Number |  |
| Thanh\_tien | Number |  |

* **DANHMUC\_CUAHANG\_TRANGCHU**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| Chu\_De\_Chinh | String |  |
| DanhSach\_CH | ArrayObjectID |  |

* **DANHMUC\_CUAHANG\_LOAIMONAN**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| Ten\_Loai\_Mon\_An | String |  |
| Icon\_Loai\_Mon\_An | String |  |
| DanhSach\_CH | ArrayObjectID |  |

* **DANHMUC\_CUAHANG\_HOMNAY**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| Thongtin\_Chinh | String |  |
| Thongtin\_Phu | String |  |
| HinhAnh\_CH | String |  |
| DanhSach\_CH | ArrayObjectID |  |

* **MONAN\_GOIY**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| Ten\_mon\_an\_goi\_y | String |  |

* **QL\_NHOM\_NGUOIDUNG**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| Ten\_nhom | String |  |
| DanhSach\_NguoiDung | ArrayObjectID |  |

* **QUANLY\_ NGUOIDUNG**

|  |  |  |
| --- | --- | --- |
| **Tên thực thể** | **Kiểu dữ liệu** | **Ghi chú** |
| \_id | ObjectID | Khóa chính |
| Ten\_dang\_nhap | String |  |
| Mat\_khau | String |  |
| Trang\_thai | String |  |

* 1. XÂY DỰNG API

Từ việc thiết kế cơ sở dữ liệu cho hệ thống FoodNow. Nhóm đã xây dựng các API cho hệ thống theo bảng sau:



1. **THIẾT KẾ GIAO DIỆN**
2. GIAO DIỆN APP KHÁCH HÀNG
   1. ĐĂNG NHẬP, ĐĂNG KÝ

Khi khách hàng vào hệ thống lần đầu sẽ hiển thị các màn hình chào mừng đến với hệ thống. Sau đó khách hàng sẽ đăng nhập bằng tài khoản google hoặc facebook mà hệ thống cung cấp. Khách hàng sẽ cung cấp số điện thoại và địa chỉ.

|  |  |
| --- | --- |
|  |  |
|  | Sau khi kết thúc màn hình mở đầu hệ thống sẽ chuyển đến màn hình đăng ký đăng nhập. |
|  | Khách hàng đăng nhập để hệ thống tạo tài khoản. Đăng nhập bằng google hoặc facebook mà hệ thống cung cấp.  Sau khi đăng nhập thì khách hàng sẽ cung cấp số điện thoại và địa chỉ. |

* 1. THÔNG TIN KHÁCH HÀNG

Khi khách hàng đăng ký bằng một trong hai tài khoản google hoặc facebook thì khách hàng có thể vào phần tài khoản để xem thông tin. Tại đây khách hàng có thể cập nhật thông tin của mình.

|  |  |
| --- | --- |
|  | Khi khách hàng đăng xuất và muốn đăng nhập lại sẽ vào tài khoản và đăng nhập. |
|  | Khi khách hàng đăng nhập thành công sẽ thấy thông tin vừa đăng nhập. |
| Hình 1.0.2 Giao diện thông tin của khách hàng | Khách hàng có thể xem chi tiết thông tin của mình và có thể cập nhật thông tin. |
|  | Khách hàng có thể thêm địa chỉ mới hoặc thay đổi địa chỉ nhà, địa chỉ công ty. |

* 1. TRANG CHỦ

Sau khi vào được hệ thống, FoodNow sẽ hiển thị giao diện trang chủ để cho khách hàng tìm kiếm cửa hàng, món ăn. Tại đây FoodNow sẽ cập nhật vị trí hiện tại mà khách hàng bật vị trí. Các cửa hàng gần vị trí sẽ hiển thị.

|  |  |
| --- | --- |
|  | Vị trí hiện tại sẽ được cập nhật khi khách hàng bật vị trí  Tại ô tìm kiếm khách hàng có thể tìm kiếm cửa hàng hoặc món ăn  Các slider hình ảnh khuyến mãi. Khi chọn sẽ hiển thị danh sách các cửa hàng nằm trong khuyến mãi đó  Danh mục 8 loại nhóm món ăn phân biệt để khách hàng dễ dàng lựa chọn mà không phải tìm kiếm  Khuyến mãi hệ thống áp dụng cho các cửa hàng |
|  | Hôm nay ăn gì là gồm các nhóm chứa các cửa hàng  FoodNow đi chợ hiển thị danh sách các cửa hàng |
|  | Tương tự Giải khát ở nhà và Ăn mì bún, phở sẽ hiển thị các cửa hàng. |
|  | Khi chọn ô tìm kiếm sẽ hiển thị các tìm kiếm gợi ý hoặc khách hàng tự nhập |
|  | Quanh đây có gì ngon sẽ hiển thị các cửa hàng có vị trí gần khách hàng nhất. |

* 1. CỬA HÀNG MÓN ĂN

Khi khách hàng lựa chọn cửa hàng, thông tin cửa hàng sẽ hiển thị cùng với các loại món ăn và món ăn. Tại đây khách hàng sẽ thêm món bằng cách chọn dấu cộng hoặc bỏ món bằng dấu trừ. Món ăn được chọn sẽ thêm vào giỏ hàng. Với món ăn được chọn có thể thêm ghi chú cho món ăn đó. Sau khi chọn giao hàng thì hệ thống sẽ lập đơn đặt hàng cho khách hàng. Việc khách hàng sẽ cập nhật lại địa chỉ hoặc số điện thoại nếu như có thay đổi.

|  |  |
| --- | --- |
|  | Khi khách tìm kiếm món mà mình yêu cầu, hệ thống sẽ cung cấp danh sách các cửa hàng tương tự món cần tìm. Trong từng cửa hàng sẽ hiển thị các món ăn cửa hàng đó để khách hàng dễ dàng chọn lựa. |
|  | Khi khách hàng chọn tab Gần tôi sẽ hiển thị danh sách các cửa hàng gần vị trí của khách hàng. |
|  | Khi khách hàng chọn tab Đặt nhiều sẽ hiển thị danh sách các cửa hàng có lượng khách đặt mua nhiều gần vị trí của khách hàng nhất. |
|  | Sau khi chọn cửa hàng mong muốn, các loại món ăn và món ăn cũng được hiển thị.  Tại đây khách sẽ tìm kiếm món và chọn món cần mua bằng cách thêm dấu cộng là món sẽ được thêm vào giỏ hàng.  Khi chọn giao hàng thì hệ thống sẽ tiến hàng lập đơn đặt hàng. |
|  | Từng loại món ăn sẽ hiển thị trên tab. |
|  | Khách hàng có thể xem được địa chỉ cửa hàng và hệ thống sẽ chuyển sang googlemap nếu khách hàng chọn vào chỉ đường. |
|  | Tại món ăn mà khách hàng lựa chọn có thể thêm ghi chú cho món ăn. |
|  | Phần ghi chú cho món ăn sẽ giới hạn là 500 ký tự. |

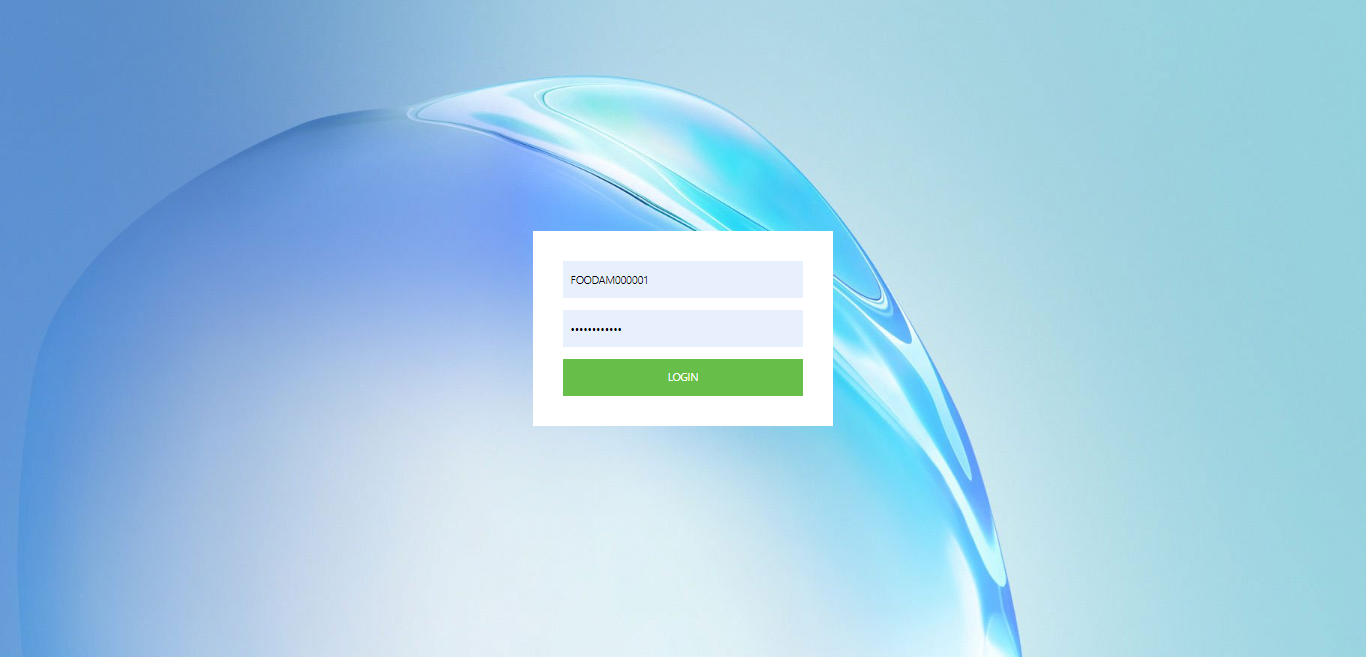
* 1. ĐƠN HÀNG

Sau khi khách hàng chọn giao hàng thì hệ thống sẽ tiến hành lập đơn đặt hàng. Khách hàng sẽ kiểm tra lại thông tin rồi bấm chọn đặt hàng.

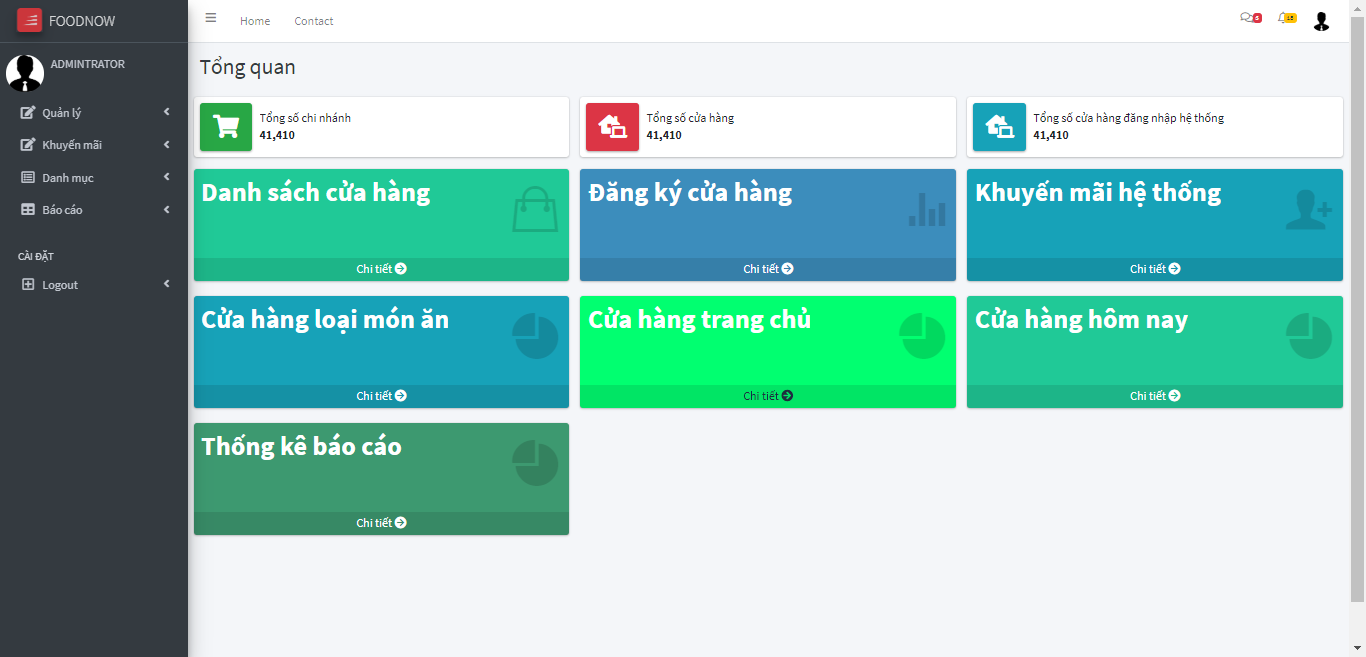
|  |  |
| --- | --- |
|  | Khi hệ thống lập đơn đặt hàng, khách hàng cần kiểm tra lại số điện thoại, địa chỉ giao trước khi bấm chọn đặt đơn.  **Khách hàng cũng có thể xóa món tại đây khi có thay đổi.** |
|  | Tại đây khách hàng có thể thay đổi khung giờ giao hàng. |
|  | Khách hàng có thể chọn lại địa chỉ trước khi đặt hàng. |
|  | Sau khi kiểm tra đầy đủ thông tin khách hàng sẽ chọn phương thức thanh toán rồi bấm đặt hàng. |

1. GIAO DIỆN WEB ADMIN
   1. ĐĂNG NHẬP

Quản trị viên sẽ nhập tên tài khoản và mật khẩu để đăng nhập vào hệ thống.



Sau khi đăng nhập thành công sẽ hiển trang tổng quan để quản trị viên dễ dàng lựa chọn.



* 1. QUẢN LÝ TÀI KHOẢN ĐĂNG KÝ

Quản trị viên sẽ nhập các thông tin cơ bản cho cửa hàng. Những thông tin này được cửa hàng cung cấp. Thuộc tính địa chỉ sẽ cho chọn trên bản đồ hoặc tự nhập. Sau đó phải xác nhận địa chỉ mới được lưu.



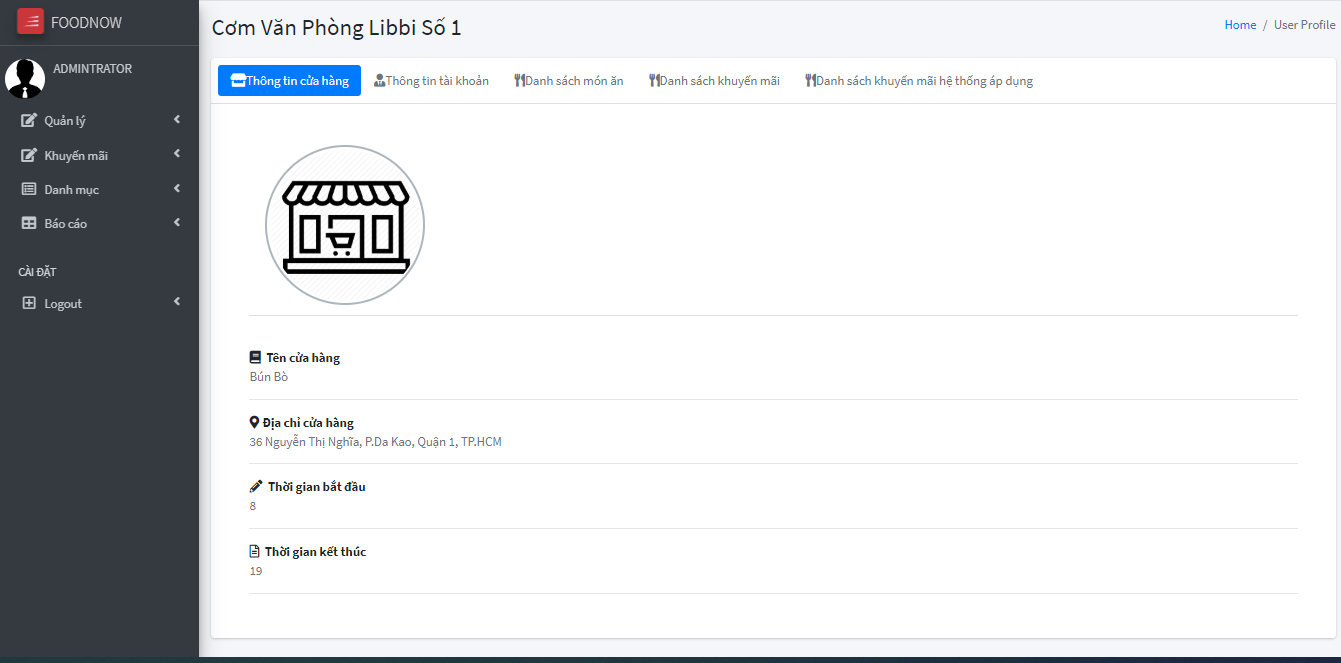
* 1. QUẢN LÝ CHI NHÁNH CỬA HÀNG

Quản trị viên có thể xem được cửa hàng vừa thêm thuộc chi nhánh nào. Cửa hàng vừa được thêm sẽ ở trạng thái chưa kích hoạt. Nếu chọn kích hoạt thì cửa hàng sẽ có quyền đăng nhập vào hệ thống để bắt đầu bán hàng. Tại danh sách cửa hàng, để có thể xem chi tiết cửa hàng đó thì quản trị chọn cửa hàng cần xem.



* 1. THÔNG TIN TÀI KHOẢN CỬA HÀNG

Khi quản trị viên chọn cửa hàng cần xem chi tiết, giao diện thông tin tài khoản của cửa hàng sẽ hiển bao gồm: Thông tin cửa hàng, Cài đặt tài khoản, Danh sách món ăn, Danh sách khuyến mãi, Danh sách khuyến mãi hệ thống áp dụng cho cửa hàng.



* 1. QUẢN LÝ KHUYẾN MÃI

Giao diện khuyến mãi hệ thống sẽ hiển thị các chi nhánh khi quản trị chọn vào dòng khuyến mãi. Quản trị có thể thêm, xóa khuyến mãi, thêm khuyến mãi áp dụng cho chi nhánh bằng cách chọn vào dấu cộng. Hiển thị danh sách các chi nhánh và quản trị sẽ chọn vào ô cần áp dụng.



* 1. QUẢN LÝ DANH MỤC
     1. DANH MỤC LOẠI MÓN ĂN

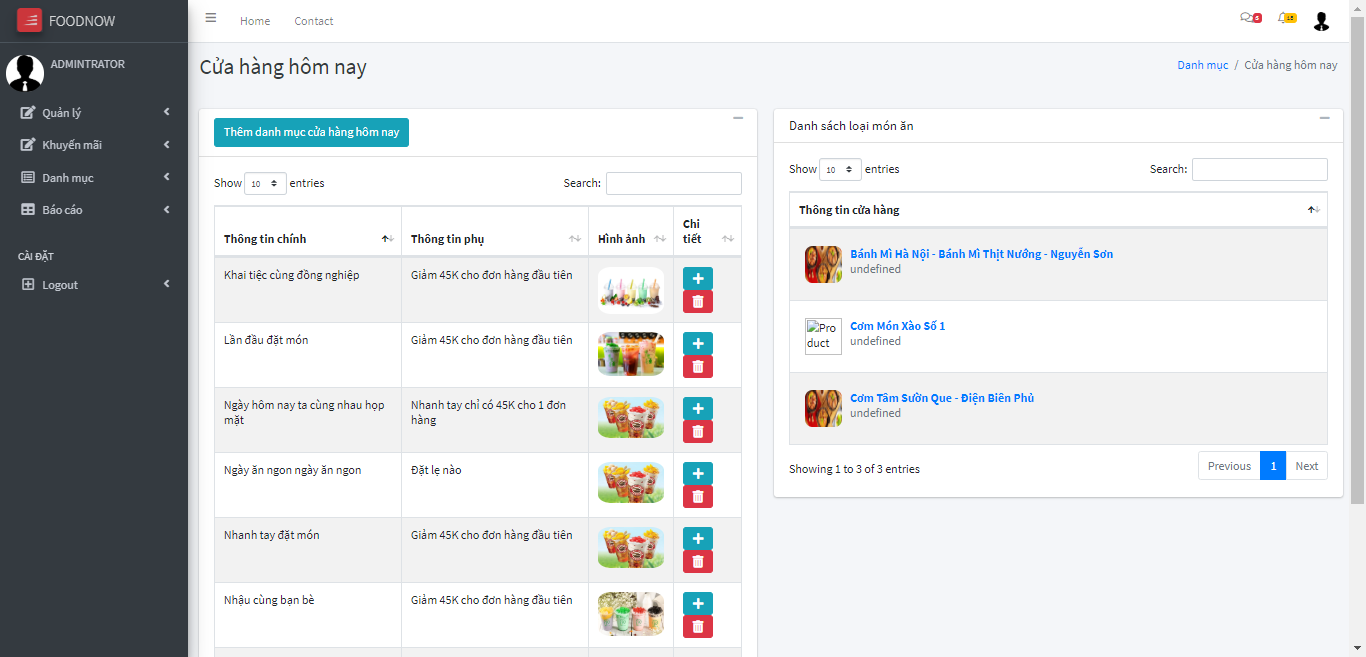
Danh mục loại món ăn gồm 8 loại, mỗi loại sẽ chứa các cửa hàng thuộc loại món đó.

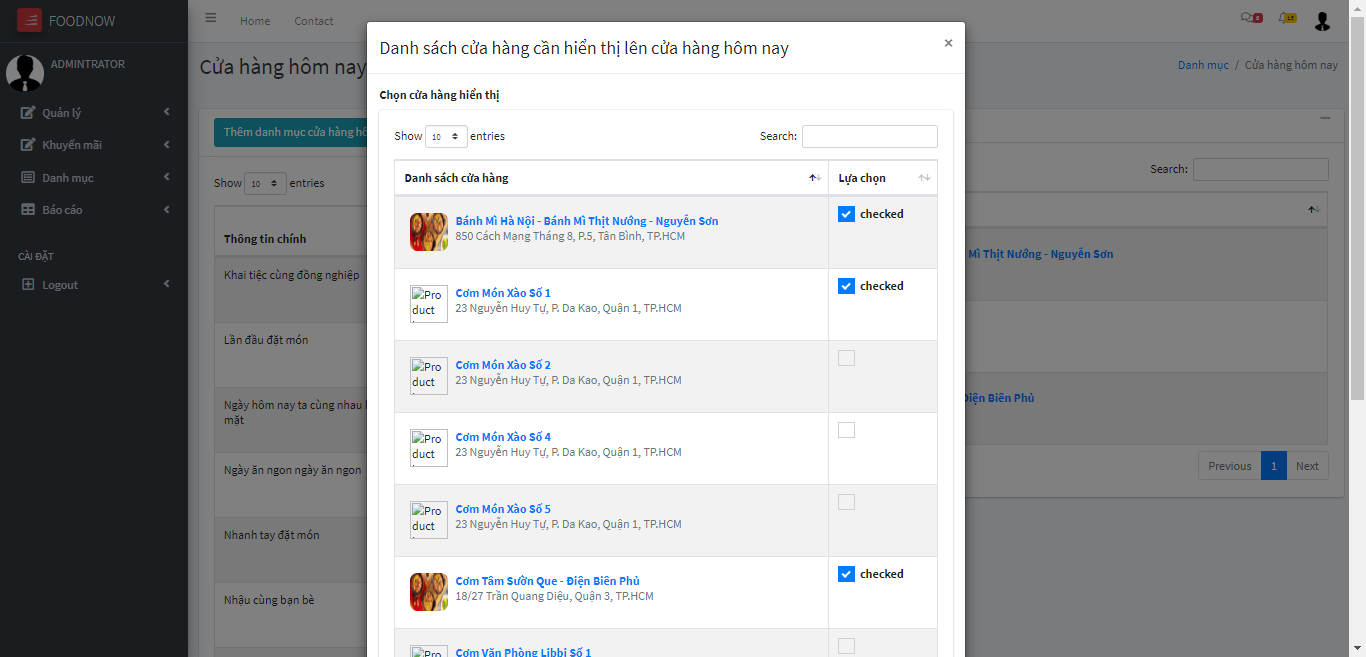
Trong từng loại sẽ hiển thị danh sách các cửa hàng, những cửa hàng được đánh dấu là những cửa hàng sẽ hiển thị trong loại món đó. Còn những cửa hàng không được đánh dấu sẽ không hiển thị trên loại món đó.



* + 1. DANH MỤC CỬA HÀNG HÔM NAY

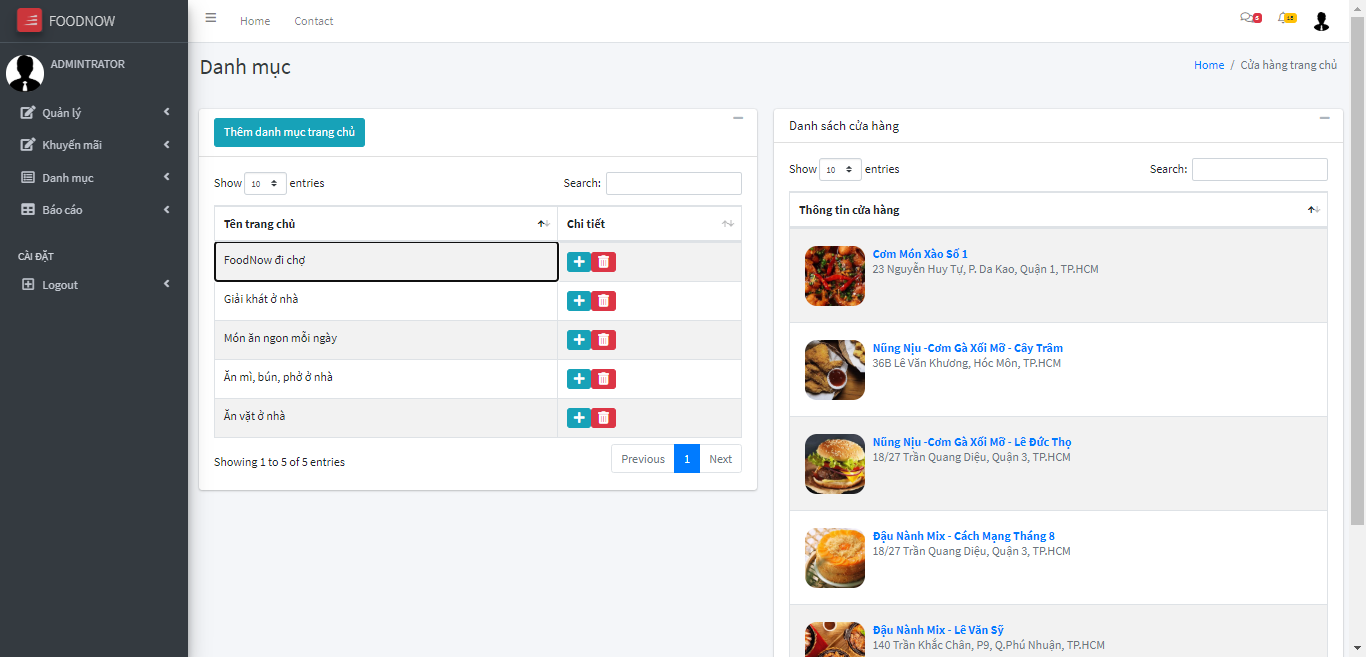
Danh mục cửa hàng hôm nay gồm các nhóm chứa cửa hàng hiển thị. Quản trị viên có thể thêm mới, xóa cho danh mục này.Đồng thời có thể thêm các cửa hàng trong nhóm bằng cách chọn dấu cộng, khi đó sẽ hiển thị tất cả các danh sách cửa hàng và quản trị viên sẽ đánh dấu cửa hàng cần hiển thị hoặc bỏ đánh dấu cửa hàng.

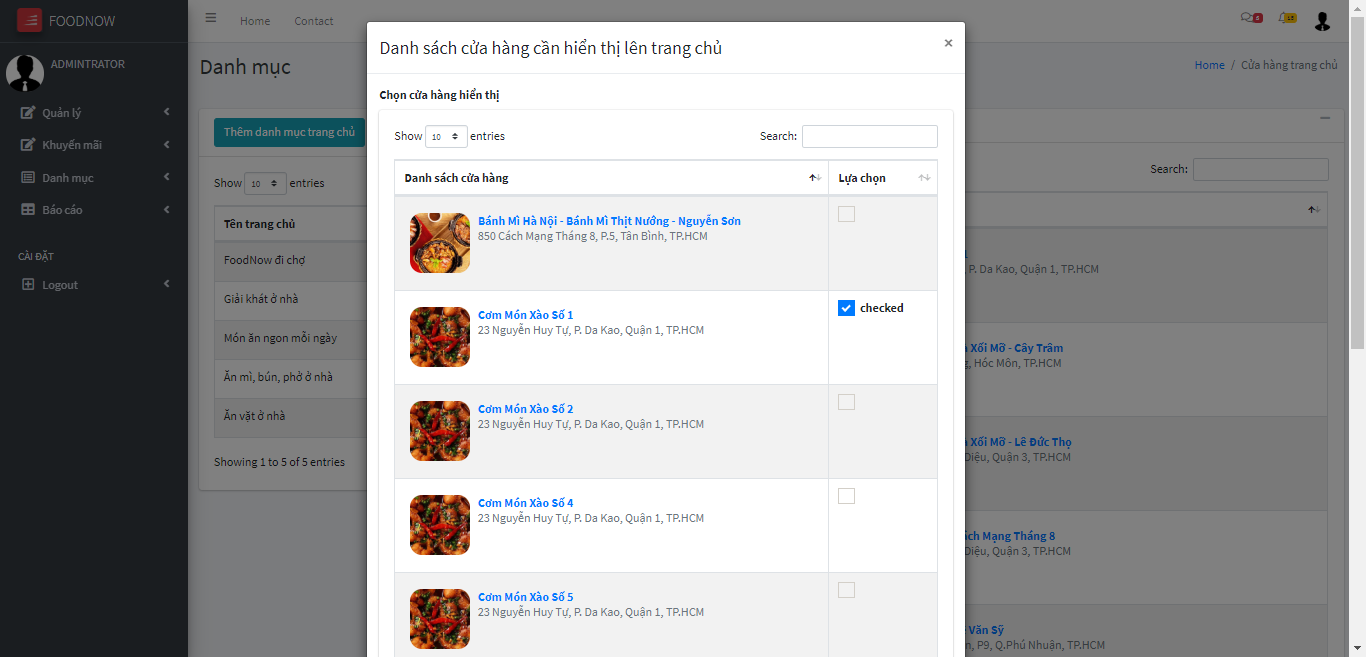




* + 1. DANH MỤC CỬA HÀNG TRANG CHỦ

Danh mục cửa hàng trang chủ gồm các nhóm chứa cửa hàng hiển thị. Quản trị viên có thể thêm mới, xóa cho danh mục này.Đồng thời có thể thêm các cửa hàng trong nhóm bằng cách chọn dấu cộng, khi đó sẽ hiển thị tất cả các danh sách cửa hàng và quản trị viên sẽ đánh dấu cửa hàng cần hiển thị hoặc bỏ đánh dấu cửa hàng.



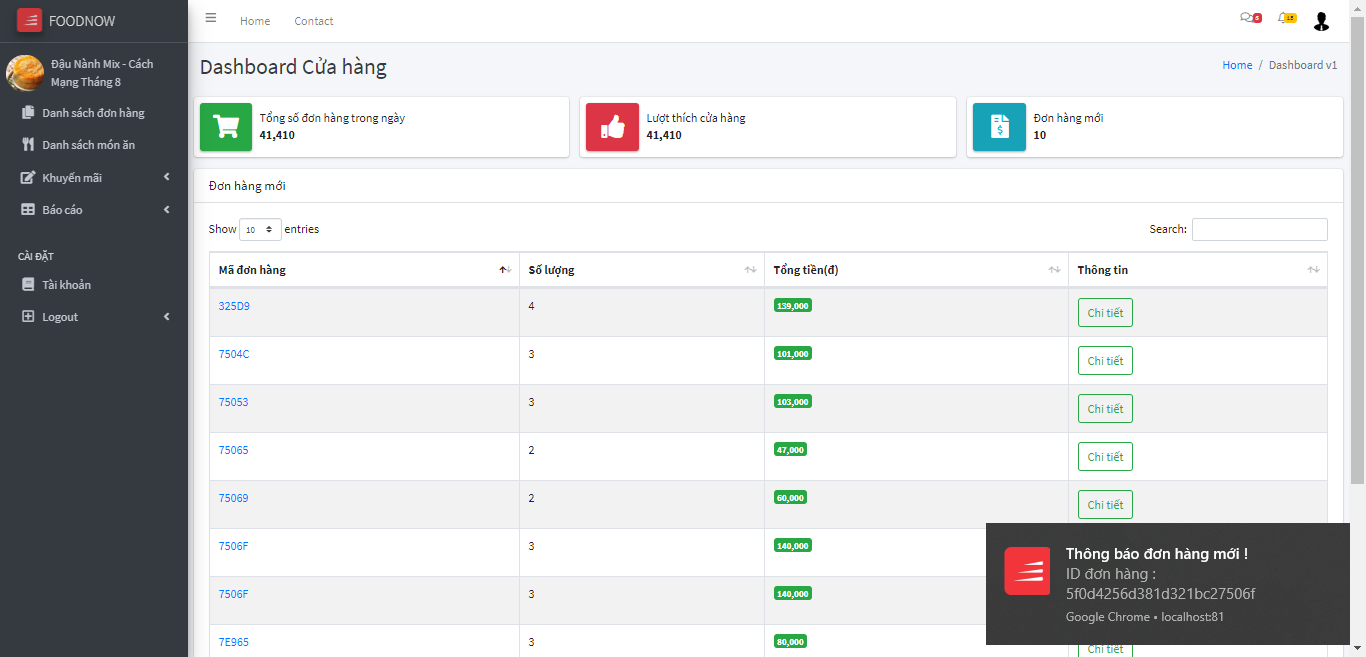


1. GIAO DIỆN WEB CỬA HÀNG
   1. ĐĂNG NHẬP

Cửa hàng sẽ đăng nhập bằng tài khoản được hệ thống cấp.



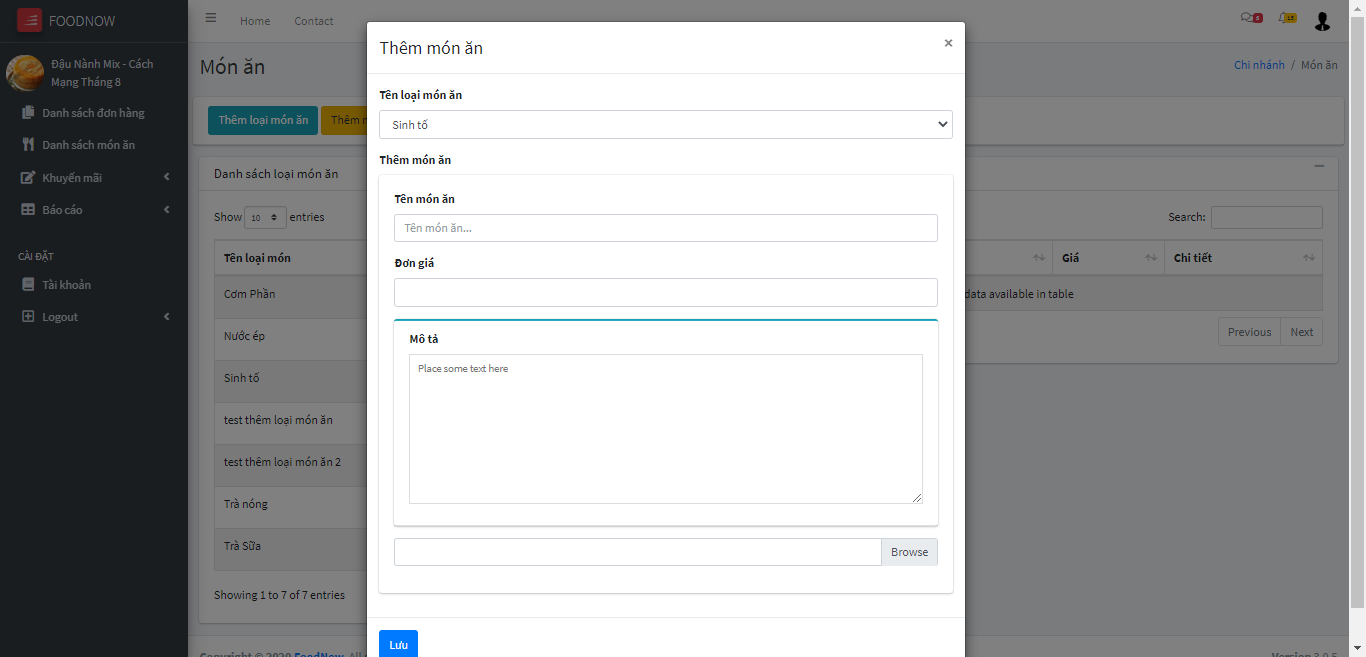
Sau khi đăng nhập thành công sẽ hiển thị trang tổng quan để cửa hàng dễ theo dõi các đơn hàng mới đến.



* 1. DANH SÁCH MÓN ĂN

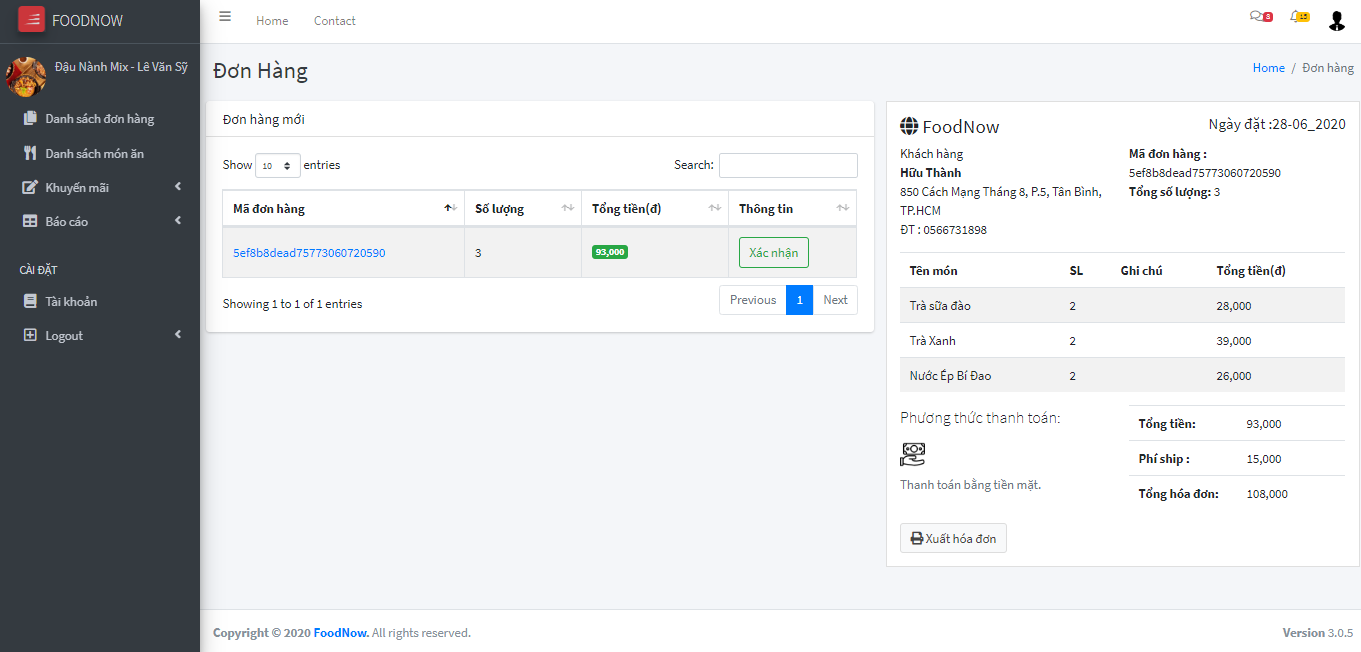
Cửa hàng có xem danh sách loại món ăn và món ăn của từng loại. Cửa hàng có thể thêm mới, xóa loại món ăn và món ăn.





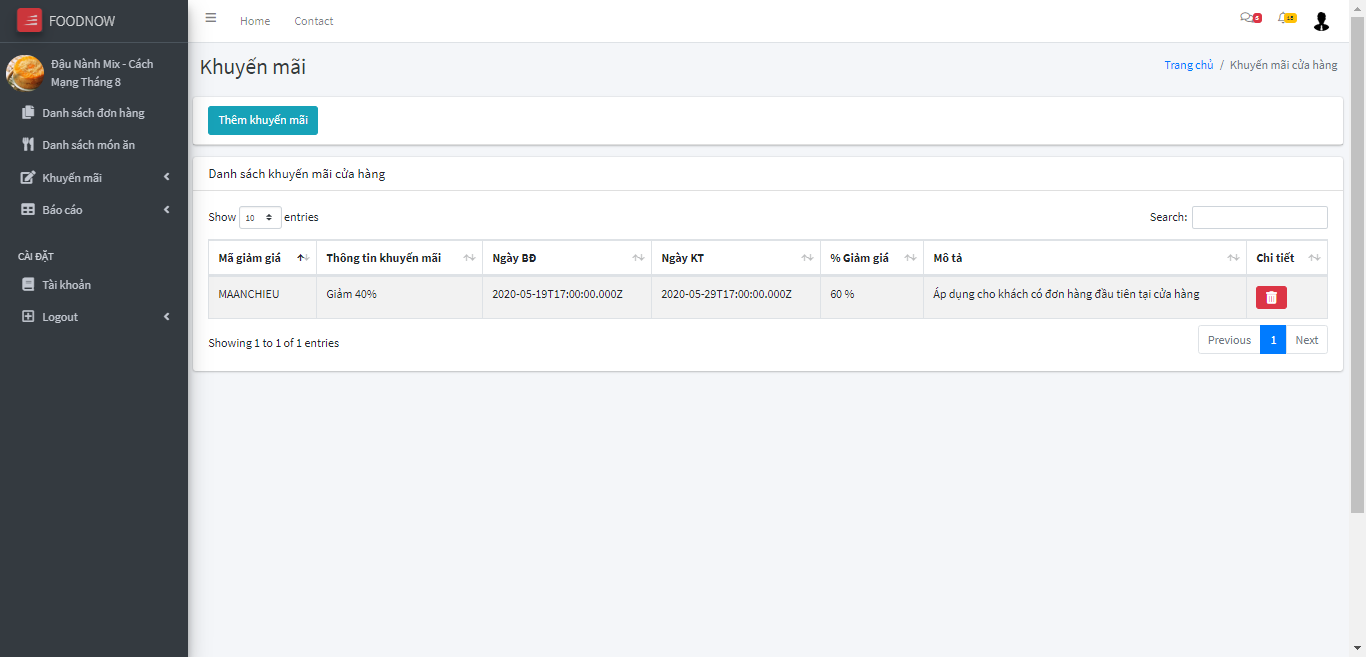
* 1. DANH SÁCH ĐƠN HÀNG

Khi cửa hàng vào danh sách đơn hàng khi chọn đơn hàng nào sẽ hiển thị chi tiết đơn hàng đó.



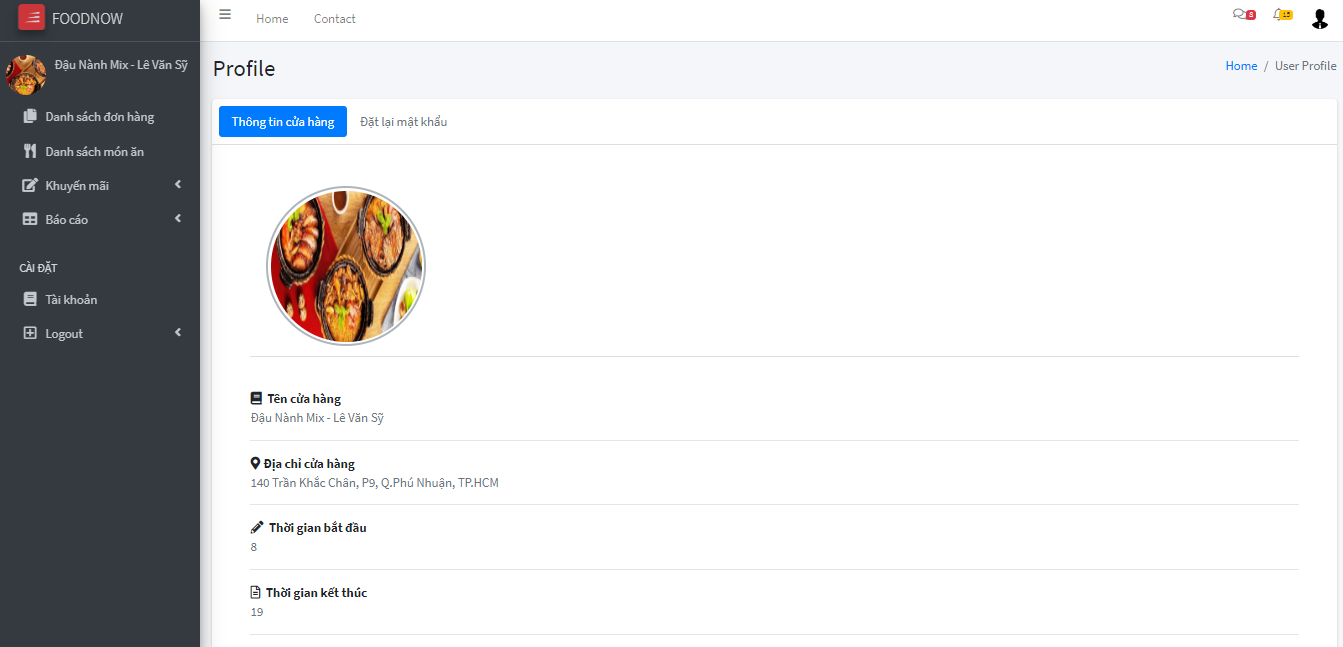
* 1. DANH SÁCH KHUYẾN MÃI

Cửa hàng có thể thêm mới, xóa khuyến mãi.



* 1. TÀI KHOẢN

Cửa hàng có thể vào tài khoản để xem thông tin đồng thời có thể cập nhật lại mật khẩu của mình.



1. **TỔNG KẾT**
2. **KẾT QUẢ ĐẠT ĐƯỢC**
3. **VỀ PHẦN KIẾN THỨC**

Về mặt kiến thức, nhóm đã đạt được các kết quả sau:

* Cách lập trình một ứng dụng Android đơn giản.
* Kiến thức về lập trình NodeJS.
* Kiến thức về cơ sở dữ liệu NoSql và MongoDB
* Kiến thức tạo API và sử dụng trong app.
* Hiểu thêm về nghiệp vụ vận hành của một app giao hàng.
* Phân tích thiết kế có hệ thống và chuyên nghiệp hơn.
* Nâng cao khả năng tự nghiên cứu, làm việc nhóm.

1. **VỀ SẢN PHẨM**

* Bản phân tích thiết kế hệ thống.
* Một hệ thống dạng RESTful API.
* Một bản demo sản phẩm

1. **ĐIỂM HẠN CHẾ**

Do thời gian tìm hiểu và quy mô hệ thống khá lơn nên sẽ xảy ra một số sai sót trong khâu phân tích và thiết kế.

Do lần đầu tiếp cận với công nghệ mới là NodeJS và MongoDB nên nhóm có thể có một số sai sót trong khâu tạo API cho hệ thống. Code có thể chưa được sắp xếp một cách khoa học và còn nhiều thiếu sót.

1. **HƯỚNG PHÁT TRIỂN**

Từ đề tài lần này, nhóm muốn phát triển lên thành một ứng dụng hoàn chỉnh sử dụng trong thực tế. Để có thể phát triển thì nhóm đặt ra các yếu cầu sau cho sản phẩm cuối cùng:

* Sản phẩm được xử lý lỗi một cách chỉnh chu và hoàn thiện.
* Sản phẩm phải chạy run-time trên mọi thiết bị.
* Sản phẩm ngoài app trên thiết bị di động phải có website quản lý cho bên cửa hàng.
* Sản phẩm đảm bảo thực hiện đúng các chức năng chính như: đăng nhập, đăng ký, đặt hàng, cập nhật thông tin tài khoản,…
* Sản phẩm sử dụng cho cửa hàng phải có các tính năng thống kê cần thiết.

# **TÀI LIỆU THAM KHẢO**

**TÀI LIỆU TIẾNG VIỆT:**

**Sách**

[1]Ths. Hoàng Thị Liên Chi, Ths. Nguyễn Văn Lễ: (2015)Bài giảng Phân tích thiết kế hệ thống thông tin, Trường Đại học Công nghiệp Thực Phẩm TPHCM, TPHCM.

**Website**

[2] techtalk.vn: (2018)Một cái nhìn tổng quan về NodeJS [online], viewed 20/10/ 2019, from< <https://techtalk.vn/mot-cai-nhin-tong-quan-nhat-ve-nodejs.html> >

[3] thanhtv.96: (2018)MongoDB là gì? Cơ sở dữ liệu phi quan hệ, viewed 20/10/2019, from <<https://viblo.asia/p/mongodb-la-gi-co-so-du-lieu-phi-quan-he-bJzKmgoPl9N> >

[4] Tin học Khoa Phạm: (2016) Hướng dẫn tự học lập trình NodeJS cơ bản, viewed 02/11/2019,from <<https://www.youtube.com/playlist?list=PLqEKeWbzk0aT9eght3b3f98qf5mnq_ABO>>

**TÀI LIỆU TIẾNG ANH:**

**BOOK:**

[5].John Horton: (2015) Android Programming for Beginners, Birmingham, UK.

[6].Dwan Griffiths & David Griffiths: (2015) Head first Android development, O’Relly Media Inc, United States of America.

[7].Ian Darwin (2017) Android cookbook, O’Relly Media Inc, United States of America.

[8]. Kristina Chodorow (2013) MongoDB The Definitive Guide, O’Relly Media Inc, United States of America.

[9]. Fernando Doglio (2018) Rest api development with node.js, Canelones, Uruguay

[10] MarkLogic Server: (2019) NodeJS Application Developẻ’s Guide [online], viewed 01/11/2019. From < <https://docs.marklogic.com/8.0/guide/node-dev.pdf> >