**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)**

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

**GVHD:** Ths. Vũ Đình Ái

Ths. Lâm Thị Họa Mi

**SINH VIÊN THỰC HIỆN:**

Kiều Hữu Thành - 2001160343

Lê Thị Ngọc Hiền - 2001160222

**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)**

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

**GVHD:** Ths. Vũ Đình Ái

Ths. Lâm Thị Họa Mi

**SINH VIÊN THỰC HIỆN:**

Kiều Hữu Thành - 2001160343

Lê Thị Ngọc Hiền - 2001160222

**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 tạo điều kiện để sinh viên có thể thực hiện đề tài một cách tốt đẹp. 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 đề tài.

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 thiện.

Cuối cùng nhóm xin cảm ơn đến những thầy cô và những người đang đọc, tham khảo và nhận xét cho đề tài này. Đóng góp của mọi người chính là những kim chỉ nam hướng dẫn nhóm định hướng phát triển đề tài được hoàn thiện hơn, 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 đơn hàng, khuyến mãi, doanh thu, lượng sản phẩm, lượng khách hàng. Bên khách hàng có thể đặt hàng, hủy đơn hàng, thanh toán trực tuyến. 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 phi quan hệ NoSQL.
* Thiết kế giao diện ứng dụng dành cho khách hàng trên nền tảng Android.
* Tạo giao diện quản lý admin, cửa hàng bằng ngôn ngữ lập trình PHP (Framework CodeIgniter).
* 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, xác thực tài khoản đăng nhập.
* Ứng dụng Firebase Cloud Messaging cho việc thông báo đơn hàng, thông tin khuyến mãi.
* Sử dụng thuật toán RSA và SHA256, tích hợp Momo Payment API vào thanh toán đơn hàng trực tuyến bằng ví Momo.

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**

[PHẦN 1: CƠ SỞ LÝ THUYẾT 1](#_Toc47099610)

[1.1. TÌM HIỂU CÔNG NGHỆ 1](#_Toc47099611)

[1.1.1. NODEJS 1](#_Toc47099612)

[1.1.2. MONGOBD 1](#_Toc47099613)

[1.1.3. FIREBASE CLOUD MESSAGING 7](#_Toc47099614)

[1.1.4. JSON WEB TOKEN 8](#_Toc47099615)

[1.1.5. GOOGLE MAP API 11](#_Toc47099616)

[1.1.6. GOOGLE SIGN-IN 11](#_Toc47099617)

[1.1.7. FACEBOOK API 12](#_Toc47099618)

[1.2. LÝ THUYẾT 16](#_Toc47099619)

[1.2.1. MOMO PAYMENT API 16](#_Toc47099620)

[1.2.1.1. THUẬT TOÁN RSA 16](#_Toc47099621)

[1.2.1.2. THUẬT TOÁN SHA-256 17](#_Toc47099622)

[1.2.1.3. SƠ ĐỒ XỬ LÝ THANH TOÁN TRỰC TUYẾN VÍ MOMO 19](#_Toc47099623)

[1.2.2. LẬP TRÌNH ANDROID 19](#_Toc47099624)

[1.2.3. CODEIGNITER WEB FRAMEWORK 22](#_Toc47099625)

[1.2.4. CƠ SỞ DỮ LIỆU NOSQL 22](#_Toc47099626)

[1.2.5. HỆ THỐNG RESTFUL API 23](#_Toc47099627)

[1.2.6. MÔ HÌNH ERD 27](#_Toc47099628)

[1.2.7. MÔ HÌNH USE CASE 28](#_Toc47099629)

[1.2.8. MÔ HÌNH DFD 29](#_Toc47099630)

[KẾT PHẦN 1 30](#_Toc47099631)

[PHẦN 2: TRIỂN KHAI ĐỀ TÀI 31](#_Toc47099632)

[**CHƯƠNG 1: TỔNG QUAN** 31](#_Toc47099633)

[1.1. GIỚI THIỆU ĐỀ TÀI 31](#_Toc47099634)

[1.2. MỤC TIÊU PHẠM VI ĐỀ TÀI 32](#_Toc47099635)

[1.2.1. MỤC TIÊU 32](#_Toc47099636)

[1.2.2. PHẠM VI 33](#_Toc47099637)

[1.3. KHẢO SÁT HỆ THỐNG 33](#_Toc47099638)

[1.4. QUI TRÌNH XỬ LÝ NGHIỆP VỤ 34](#_Toc47099639)

[1.4.1. SƠ ĐỒ HỆ THỐNG 34](#_Toc47099640)

[1.4.2. MÔ HÌNH NGHIỆP VỤ BPM 35](#_Toc47099641)

[**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG** 49](#_Toc47099642)

[1. PHÂN TÍCH HỆ THỐNG 49](#_Toc47099643)

[1.1. MÔ HÌNH ERD 49](#_Toc47099644)

[2. THIẾT KẾ HỆ THỐNG 50](#_Toc47099645)

[2.1. THIẾT KẾ CƠ SỞ DỮ LIỆU 50](#_Toc47099646)

[2.2. SƠ ĐỒ PHÂN CẤP CHỨC NĂNG 52](#_Toc47099647)

[2.3. MÔ HÌNH DFD 53](#_Toc47099648)

[2.4. MÔ HÌNH USE CASE 56](#_Toc47099649)

[2.5. XÂY DỰNG API 59](#_Toc47099650)

[**CHƯƠNG 3: THIẾT KẾ GIAO DIỆN** 62](#_Toc47099651)

[3.1. GIAO DIỆN APP KHÁCH HÀNG 62](#_Toc47099652)

[3.1.1. ĐĂNG NHẬP, ĐĂNG KÝ 62](#_Toc47099653)

[3.1.2. THÔNG TIN KHÁCH HÀNG 64](#_Toc47099654)

[3.1.3. TRANG CHỦ 67](#_Toc47099655)

[3.1.4. CỬA HÀNG MÓN ĂN 70](#_Toc47099656)

[3.1.5. ĐƠN HÀNG 74](#_Toc47099657)

[3.2. GIAO DIỆN WEB ADMIN 80](#_Toc47099658)

[3.2.1. ĐĂNG NHẬP 80](#_Toc47099659)

[3.2.2. QUẢN LÝ TÀI KHOẢN ĐĂNG KÝ 82](#_Toc47099660)

[3.2.3. QUẢN LÝ CHI NHÁNH CỬA HÀNG 86](#_Toc47099661)

[3.2.4. THÔNG TIN TÀI KHOẢN CỬA HÀNG 87](#_Toc47099662)

[3.2.5. QUẢN LÝ KHUYẾN MÃI 90](#_Toc47099663)

[3.2.6. QUẢN LÝ DANH MỤC 92](#_Toc47099664)

[3.3. GIAO DIỆN WEB CỬA HÀNG 97](#_Toc47099665)

[3.3.1. ĐĂNG NHẬP 97](#_Toc47099666)

[3.3.2. DANH SÁCH ĐƠN HÀNG 99](#_Toc47099667)

[3.3.3. DANH SÁCH MÓN ĂN 100](#_Toc47099668)

[3.3.4. DANH SÁCH KHUYẾN MÃI 102](#_Toc47099669)

[3.3.5. TÀI KHOẢN 104](#_Toc47099670)

[3.3.6. ĐƠN HÀNG THỐNG KÊ 105](#_Toc47099671)

[KẾT PHẦN 2 106](#_Toc47099672)

[PHẦN 3: TỔNG KẾT 107](#_Toc47099673)

[3.1. KẾT QUẢ ĐẠT ĐƯỢC 107](#_Toc47099674)

[3.1.1. VỀ PHẦN KIẾN THỨC 107](#_Toc47099675)

[3.1.2. VỀ SẢN PHẨM 107](#_Toc47099676)

[3.2. ĐIỂM HẠN CHẾ 107](#_Toc47099677)

[3.3. HƯỚNG PHÁT TRIỂN 107](#_Toc47099678)

[TÀI LIỆU THAM KHẢO 109](#_Toc47099679)

# **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 |
| API | **Application Programming Interface** | Giao diện lập trình ứng dụng |
| UML | Unified Modeling Language | Ngôn ngữ mô hình hóa thống nhất |
| RSA | Ronald Rivest, Adi Shamir và Leonard Adleman |  |
| SHA | Secure Hash Algorithm | Thuật giải băm an toàn |
| SSL | **Secure Sockets Layer** |  |
| TSL | **Transport Layer Security** |  |
| HTTPS | **Hypertext Transfer Protocol Secure** | Giao thức truyền tải siêu văn bản an toàn |
| FCM | **Firebase Cloud Messaging** |  |
| GUI | **Graphical User Interface** |  |
| SDK | **Software Development Kit** |  |
| FBML | Facebook Markup Language | Ngôn Ngữ Đánh Dấu Facebook |

**DANH MỤC HÌNH VẼ VÀ BẢNG**

Hình 1.1. Mô hình xử lý của NodeJS 3

Hình 1.2. Mô hình xử lý của ASP.Net 4

[Hình 1.3. Cách MongoDB lưu trữ một bản ghi 5](#_Toc47077950)

[Hình 1.4. Mô hình hoạt động MongoDB 5](#_Toc47077951)

[Hình 1.5. Hoạt động Firebase Cloud Messaging 7](#_Toc47077952)

[Hình 1.6. Cấu trúc của Json Web Token 9](#_Toc47077953)

[Hình 1.7. Luồng xử lý bảo mật JWT 10](#_Toc47077954)

[Hình 1.8. Hoạt động của Google API 12](#_Toc47077955)

[Hình 1.9. Hoạt động Facebook API 13](#_Toc47077956)

[Hình 1.10. Mã hóa thông tin 17](#_Toc47077957)

[Hình 1.11. Sơ đồ xử lý thanh toán trực tuyến ví Momo 19](#_Toc47077958)

[Hình 1.12. Vòng đời của một Activity 21](#_Toc47077959)

[Hình 1.13. Mô hình hoạt động của RESTful API 25](#_Toc47077960)

[Hình 2.1.1. Sơ đồ hệ thống FoodNow 35](#_Toc47077961)

[Hình 2.1.2. Quy trình khách hàng đăng ký/ đăng nhập hệ thống 36](#_Toc47077962)

[Hình 2.1.3. Quy trình khách hàng đặt hàng 38](#_Toc47077963)

[Hình 2.1.4. Quy trình khách hàng hủy đơn hàng 39](#_Toc47077964)

[Hình 2.1.5. Quy trình giao hàng 41](#_Toc47077965)

[Hình 2.1.6. Quy trình nhận hàng 43](#_Toc47077966)

[Hình 2.1.7. Quy trình nhân viên giao hàng hủy đơn 44](#_Toc47077967)

[Hình 2.1.8. Quy trình khách hàng không nhận hàng 45](#_Toc47077968)

[Hình 2.1.9. Quy trình đăng ký làm đối tác (tài xế) hệ thống 46](#_Toc47077969)

[Hình 2.1.10. Quy trình đăng ký cửa hàng 47](#_Toc47077970)

[Hình 2.1.11. Quy trình xác nhận hủy đơn hàng 48](#_Toc47077971)

[Hình 2.2.1. Mô hình ERD 49](#_Toc47077972)

[Hình 2.2.2. Mô hình cơ sỡ dữ liệu NoSQL 51](#_Toc47077973)

[Hình 2.2.3. Mô hình phân cấp chức năng 52](#_Toc47077974)

[Hình 2.2.4. Mô hình mức ngữ cảnh 53](#_Toc47077975)

[Hình 2.2.5. Mô hình mức đỉnh 53](#_Toc47077976)

[Hình 2.2.6. Mô hình quản lý khách hàng 54](#_Toc47077977)

[Hình 2.2.7. Mô hình quản lý đặt hàng 54](#_Toc47077978)

[Hình 2.2.8. Mô hình quản hủy đơn hàng 55](#_Toc47077979)

[Hình 2.2.9. Mô hình quản lý cửa hàng 55](#_Toc47077980)

[Hình 2.2.10. Mô hình use case khách hàng 56](#_Toc47077981)

[Hình 2.2.11. Mô hình use case cửa hàng 57](#_Toc47077982)

[Hình 2.2.12. Mô hình use case quản trị viên 58](#_Toc47077983)

[Bảng 2.2.13. Thiết kế API 61](#_Toc47077984)

[Hình 2.3.1. Giao diện mở đầu hệ thống 63](#_Toc47077985)

[Hình 2.3.2. Giao diện đăng ký, đăng nhập 63](#_Toc47077986)

[Hình 2.3.3. Giao diện tài khoản đăng nhập 64](#_Toc47077987)

[Hình 2.3.4. Giao diện tài khoản đăng xuất 65](#_Toc47077988)

[Hình 2.3.5. Giao diện thông tin chi tiết khách hàng 65](#_Toc47077989)

[Hình 2.3.6. Giao diện thêm địa chỉ mới 66](#_Toc47077990)

[Hình 2.3.7. Giao diện trang chủ phần đầu 67](#_Toc47077991)

[Hình 2.3.8. Giao diện trang chủ phần giữa hôm nay 68](#_Toc47077992)

[Hình 2.3.9. Giao diện trang chủ phần giữa giải khát ở nhà 68](#_Toc47077993)

[Hình 2.3.10. Giao diện tìm kiếm món ăn 69](#_Toc47077994)

[Hình 2.3.11. Giao diện phần cuối trang chủ 69](#_Toc47077995)

[Hình 2.3.12. Giao diện danh sách cửa hàng tìm kiếm 70](#_Toc47077996)

[Hình 2.3.13. Giao diện danh sách cửa hàng gần tôi 71](#_Toc47077997)

[Hình 2.3.14. Giao diện danh sách cửa hàng đặt nhiều 71](#_Toc47077998)

[Hình 2.3.15. Giao diện thông tin cửa hàng cùng món ăn 72](#_Toc47077999)

[Hình 2.3.16. Giao diện thông tin khuyến mãi cửa hàng 72](#_Toc47078000)

[Hình 2.3.17. Giao diện danh sách loại món ăn và món ăn của cửa hàng 73](#_Toc47078001)

[Hình 2.3.18. Giao diện địa chỉ cửa hàng 73](#_Toc47078002)

[Hình 2.3.19. Giao diện món ăn trong giỏ hàng 74](#_Toc47078003)

[Hình 2.3.20. Giao diện ghi chú món ăn 74](#_Toc47078004)

[Hình 2.3.21. Giao diện đơn đặt hàng 75](#_Toc47078005)

[Hình 2.3.22. Giao diện chọn thời gian 76](#_Toc47078006)

[Hình 2.3.23. Giao diện thay đổi địa chỉ 76](#_Toc47078007)

[Hình 2.3.24. Giao diện chọn đặt đơn khi đã kiểm tra thông tin đầy đủ 77](#_Toc47078008)

[Hình 2.3.25. Giao diện chọn đặt đơn khi đã kiểm tra thông tin đầy đủ 77](#_Toc47078009)

[Hình 2.3.26. Giao diện khách hàng thanh toán bằng ví Momo thành công 78](#_Toc47078010)

[Hình 2.3.27. Giao diện hiển thị thông báo thanh toán thành công trên ví Momo 78](#_Toc47078011)

[Hình 2.3.28. Giao diện hiển thị thông báo thanh toán Momo thành công 79](#_Toc47078012)

[Hình 2.3.29. Giao diện quản trị đăng nhập 80](#_Toc47078013)

[Hình 2.3.30. Giao diện tổng quan 81](#_Toc47078014)

[Hình 2.3.31. Giao diện đăng ký cho cửa hàng thông tin cơ bản 82](#_Toc47078015)

[Hình 2.3.32. Giao diện đăng ký cho cửa hàng thông tin người đại diện 83](#_Toc47078016)

[Hình 2.3.33. Giao diện đăng ký cho cửa hàng thông tin chi tiết 83](#_Toc47078017)

[Hình 2.3.34. Giao diện đăng ký cho cửa hàng kiểm tra thông tin 84](#_Toc47078018)

[Hình 2.3.35. Giao diện đăng ký cho cửa hàng kiểm tra thông tin trước khi bấm đăng ký 84](#_Toc47078019)

[Hình 2.3.36. Giao diện thông tin tài khoản cửa hàng khi đăng ký thành công 85](#_Toc47078020)

[Hình 2.3.37. Giao diện quản lý chi nhánh cửa hàng 86](#_Toc47078021)

[Hình 2.3.38. Giao diện quản lý thông tin cửa hàng 87](#_Toc47078022)

[Hình 2.3.39. Giao diện quản lý thông tin cửa hàng cài đặt tài khoản 88](#_Toc47078023)

[Hình 2.3.40. Giao diện quản lý thông tin cửa hàng danh sách món ăn 88](#_Toc47078024)

[Hình 2.3.41. Giao diện quản lý thông tin cửa hàng danh sách khuyến mãi 89](#_Toc47078025)

[Hình 2.3.42. Giao diện quản lý thông tin cửa hàng danh sách khuyến mãi hệ thống áp dụng 89](#_Toc47078026)

[Hình 2.3.43. Giao diện quản lý khuyến mãi hệ thống 90](#_Toc47078027)

[Hình 2.3.44. Giao diện thêm mới khuyến mãi hệ thống 91](#_Toc47078028)

[Hình 2.3.45. Giao diện áp dụng khuyến mãi hệ thống cho chi nhánh 91](#_Toc47078029)

[Hình 2.3.46. Giao diện danh mục loại món ăn 92](#_Toc47078030)

[Hình 2.3.47. Giao diện danh sách các nhóm trong danh mục cửa hàng hôm nay 93](#_Toc47078031)

[Hình 2.3.48.Giao diện thêm mới một nhóm nằm trong danh mục cửa hàng hôm nay 94](#_Toc47078032)

[Hình 2.3.49. Giao diện thêm các cửa hàng vào từng nhóm trong danh mục cửa hàng hôm nay. 94](#_Toc47078033)

[Hình 2.3.50. Giao diện danh sách các nhóm trong danh mục cửa hàng trang chủ. 95](#_Toc47078034)

[Hình 2.3.51. Giao diện thêm mới nhóm trong danh mục. 96](#_Toc47078035)

[Hình 2.3.52. Giao diện thêm các cửa hàng vào từng nhóm trong danh mục trang chủ. 96](#_Toc47078036)

[Hình 2.3.53. Giao diện đăng nhập cửa hàng. 97](#_Toc47078037)

[Hình 2.3.54. Giao diện tổng quan cửa hàng. 98](#_Toc47078038)

[Hình 2.3.55. Giao diện danh sách đơn hàng cửa hàng. 99](#_Toc47078039)

[Hình 2.3.56. Giao diện danh sách loại món ăn và món ăn. 100](#_Toc47078040)

[Hình 2.3.57. Giao diện thêm mới loại món ăn. 100](#_Toc47078041)

[Hình 2.3.58. Giao diện thêm món ăn. 101](#_Toc47078042)

[Hình 2.3.59. Giao diện danh sách khuyến mãi cửa hàng. 102](#_Toc47078043)

[Hình 2.3.60. Giao diện thêm mới khuyến mãi cửa hàng. 103](#_Toc47078044)

[Hình 2.3.61. Giao diện khuyến mãi hệ thống áp dụng cho cửa hàng. 103](#_Toc47078045)

[Hình 2.3.62. Giao diện thông tin tài khoản cửa hàng. 104](#_Toc47078046)

[Hình 2.3.63. Giao diện thông tin tài khoản cửa hàng cài đặt tài khoản. 104](#_Toc47078047)

[Hình 2.3.64. Thống kê đơn hàng 105](#_Toc47078048)

# **PHẦN 1: CƠ SỞ LÝ THUYẾT**

1. **TÌM HIỂU CÔNG NGHỆ**
   1. **NODEJS**

**Node.js là một mã nguồn mở, một môi trường cho các máy chủ và ứng dụng mạng.**

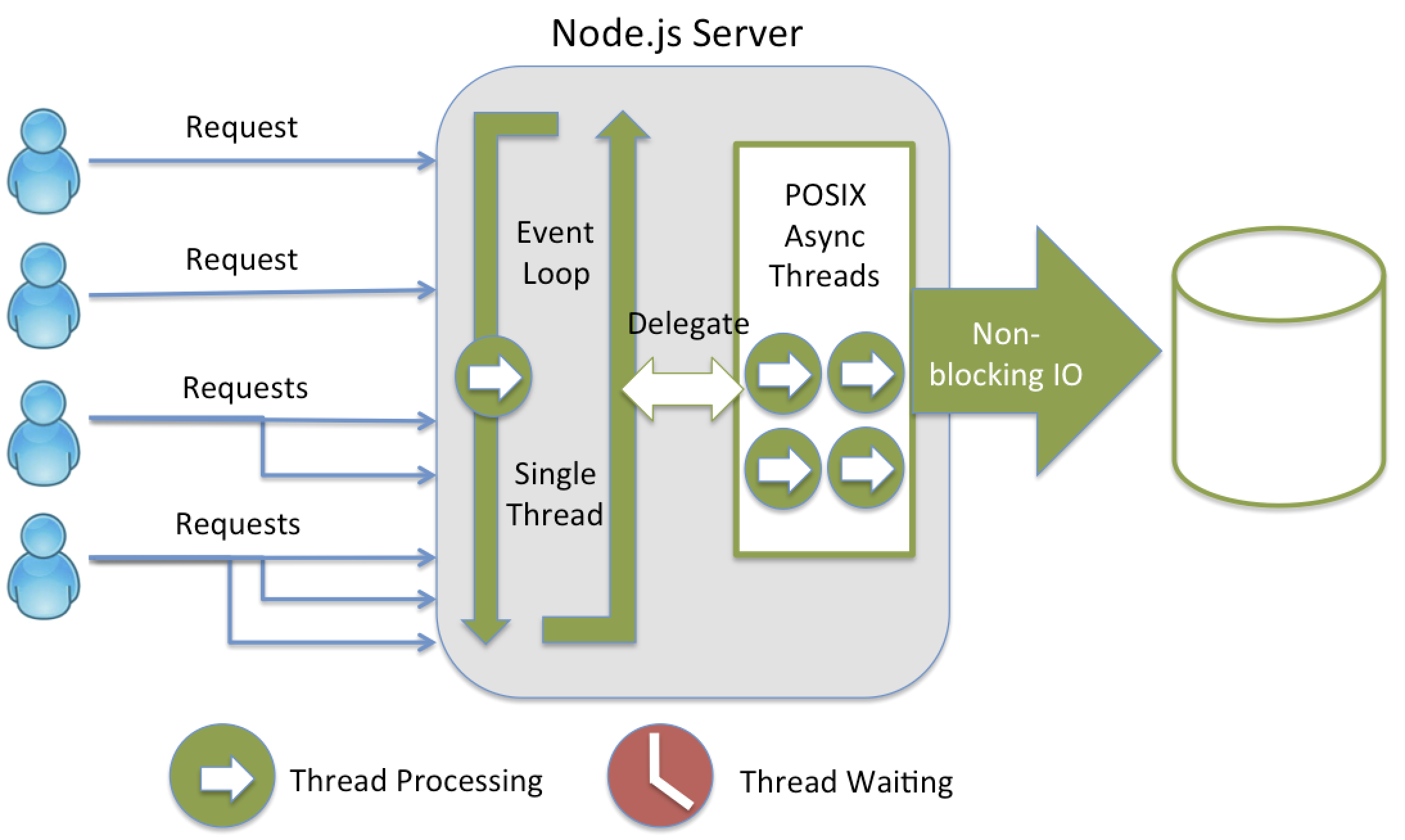
**Node.js sử dụng Google V8 JavaScript engine để thực thi mã, và một tỷ lệ lớn các mô-đun cơ bản được viết bằng JavaScript. Các ứng dụng node.js thì được viết bằn JavaScript.**

**Node.js chứa một thư viện built-in cho phép các ứng dụng hoạt động như một Webserver mà không cần phần mềm như Nginx, Apache HTTP Server hoặc IIS.**

**Node.js cung cấp kiến trúc hướng sự kiện (event-driven) và non-blocking I/O API, tối ưu hóa thông lượng của ứng dụng và có khả năng mở rộng cao**

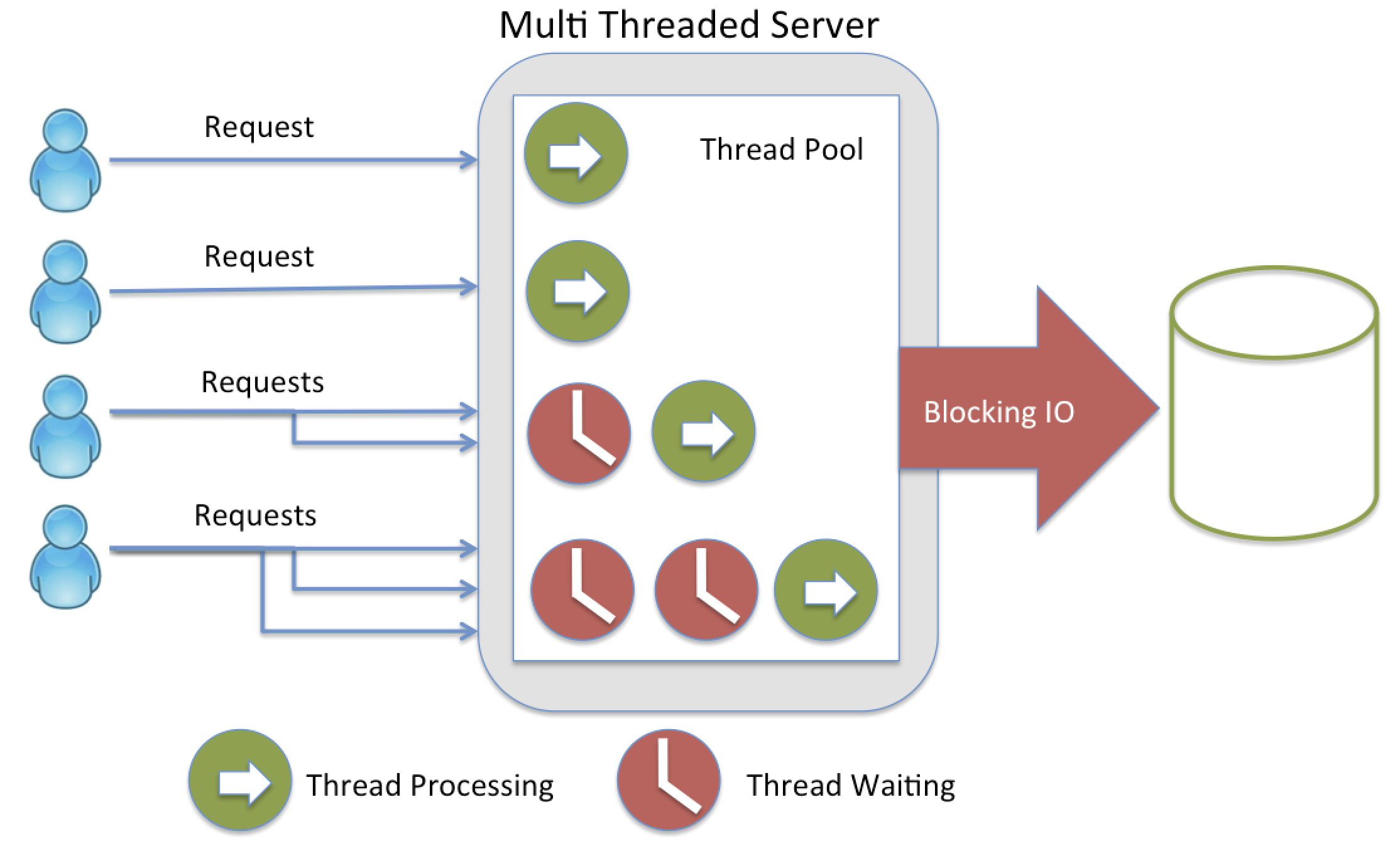
**Mọi hàm trong Node.js là không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền (background processing).**

* **Ứng dụng NodeJS**
* Xây dựng websocket server (Chat server)
* Hệ thống Notification (Giống như facebook hayTwitter)
* Ứng dụng upload file trên client
* Các máy chủ quảng cáo
* Các ứng dụng dữ liệu thời gian thực khác.
* **Ưu điểm NodeJS**
* Đặc điểm nổi bật của Node.js là nó nhận và xử lý nhiều kết nối chỉ với một single-thread. Điều này giúp hệ thống tốn ít RAM nhất và chạy nhanh nhất khi không phải tạo thread mới cho mỗi truy vấn giống PHP.
* Tận dụng ưu điểm non-blocking I/O của Javascript mà Node.js tận dụng tối đa tài nguyên của server mà không tạo ra độ trễ như PHP.
* JSON APIs: Với cơ chế event-driven, non-blocking I/O(Input/Output) và mô hình kết hợp với Javascript là sự lựa chọn tuyệt vời cho các dịch vụ Webs làm bằng JSON.
* Ứng dụng trên 1 trang (Single page Application): Với khả năng xử lý nhiều Request/s đồng thời thời gian phản hồi nhanh. Các ứng dụng bạn định viết không muốn nó tải lại trang, gồm rất nhiều request từ người dùng cần sự hoạt động nhanh để thể hiện sự chuyên nghiệp thì NodeJS sẽ là sự lựa chọn của bạn.
* Shelling tools unix: NodeJS sẽ tận dụng tối đa Unix để hoạt động. Tức là NodeJS có thể xử lý hàng nghìn Process và trả ra 1 luồng khiến cho hiệu xuất hoạt động đạt mức tối đa nhất và tuyệt vời nhất.
* Streamming Data (Luồng dữ liệu): Các web thông thường gửi HTTP request và nhận phản hồi lại (Luồng dữ liệu). Giả xử sẽ cần xử lý 1 luồng giữ liệu cực lớn, NodeJS sẽ xây dựng các Proxy phân vùng các luồng dữ liệu để đảm bảo tối đa hoạt động cho các luồng dữ liệu khác.
* Ứng dụng Web thời gian thực: Với sự ra đời của các ứng dụng di động & HTML 5 nên Node.js rất hiệu quả khi xây dựng những ứng dụng thời gian thực (real-time applications) như ứng dụng chat, các dịch vụ mạng xã hội như Facebook, Twitter,…
* Sử dụng NodeJS khi:
* Node.js thực sự tỏa sáng trong việc xây dựng RESTful API (json). Gần như không có ngôn ngữ nào xử lý JSON dễ dàng hơn Javascript, chưa kể các API server thường không phải thực hiện những xử lý nặng nề nhưng lượng concurrent request thì rất cao mà Node.js thì xử lý non-blocking.
* Những ứng dụng đòi hỏi các giao thức kết nối khác chứ không phải chỉ có http. Với việc hỗ trợ giao thức tcp, có thể xây dựng bất kỳ một giao thức custom nào đó một cách dễ dàng.
* Những ứng dụng thời gian thực.
* Những website stateful. Node.js xử lý mọi request trên cùng một process giúp cho việc xây dựng các bộ nhớ đệm chưa bao giờ đơn giản đến thế: Hãy lưu nó vào một biến global, và thế là mọi request đều có thể truy cập đến bộ nhớ đệm đó. Caching sẽ không còn quá đau đầu như trước đây, và có thể lưu cũng như chia sẻ trạng thái của một client với các client khác ngay trong ngôn ngữ, chứ không cần thông qua các bộ nhớ ngoài.
* Mô hình **xử** lý:
* Hoạt động trên lập trình không đồng bộ, cho phép máy chủ giải quyết nhiều yêu cầu đồng thời mà không chặn chúng. Mặc dù thực tế là nó chạy trên một luồng, nhưng nó sẽ theo dõi nhiều luồng nền để lên lịch cho một số tác vụ. Phương pháp này rất có lợi cho các nhà phát triển để xây dựng các ứng dụng có tính toàn diện cao.



*Hình 1.1. Mô hình xử lý của NodeJS*

* Với ASP.NET: Nó hoạt động trên mô hình đồng bộ và đa luồng, tức là nó xử lý liên tiếp tất cả các yêu cầu có trong nhóm luồng. Nếu tổng số nhiệm vụ nhiều hơn số chủ đề có sẵn, họ phải đợi cho đến khi nó trở nên miễn phí. Điều này đòi hỏi nhiều không gian lưu trữ hơn và giảm tốc độ xử lý ứng dụng.



*Hình 1.2. Mô hình xử lý của ASP.Net*

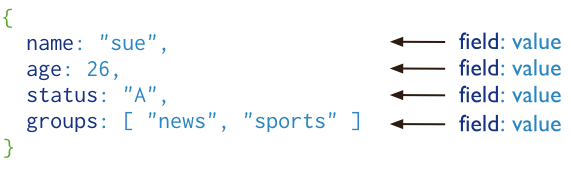
* 1. **MONGOBD**

**MongoDB**là một mã nguồn mở và là một tập tài liệu dùng cơ chế NoSQL để truy vấn, nó được viết bởi ngôn ngữ C++. Chính vì được viết bởi C++ nên nó có khả năng tính toán với tốc độ cao chứ không giống như các hệ quản trị CSDL hiện nay.

Mỗi một table (bảng dữ liệu) trong SQL sử dụng thì trong MongoDB gọi là collection (tập hợp)

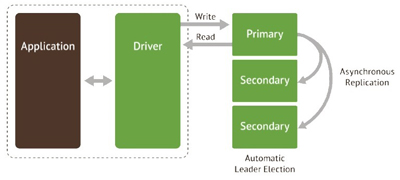
Mỗi một record (bản ghi) trong MongoDB được gọi là document (tài liệu).

Một bản ghi của MongoDB được lưu trữ dưới dạng document (tài liệu), nó được ghi xuống với cấu trúc field (trường) và value (giá trị). Nó giống như là một đối tượng JSON có dạng như sau:



*Hình 1.3. Cách MongoDB lưu trữ một bản ghi*

Điều đó có thể dễ dàng ép kiểu sang ***array*** (mảng) để lập trình các ứng dụng một cách dễ dàng hơn. Nói một cách dễ hiểu thì mỗi một bản ghi của MongoDB là một mảng dữ liệu riêng biệt bao gồm các cặp key, value khác nhau do đó cách lưu trữ của MongoDB là phi cấu trúc dữ liệu.

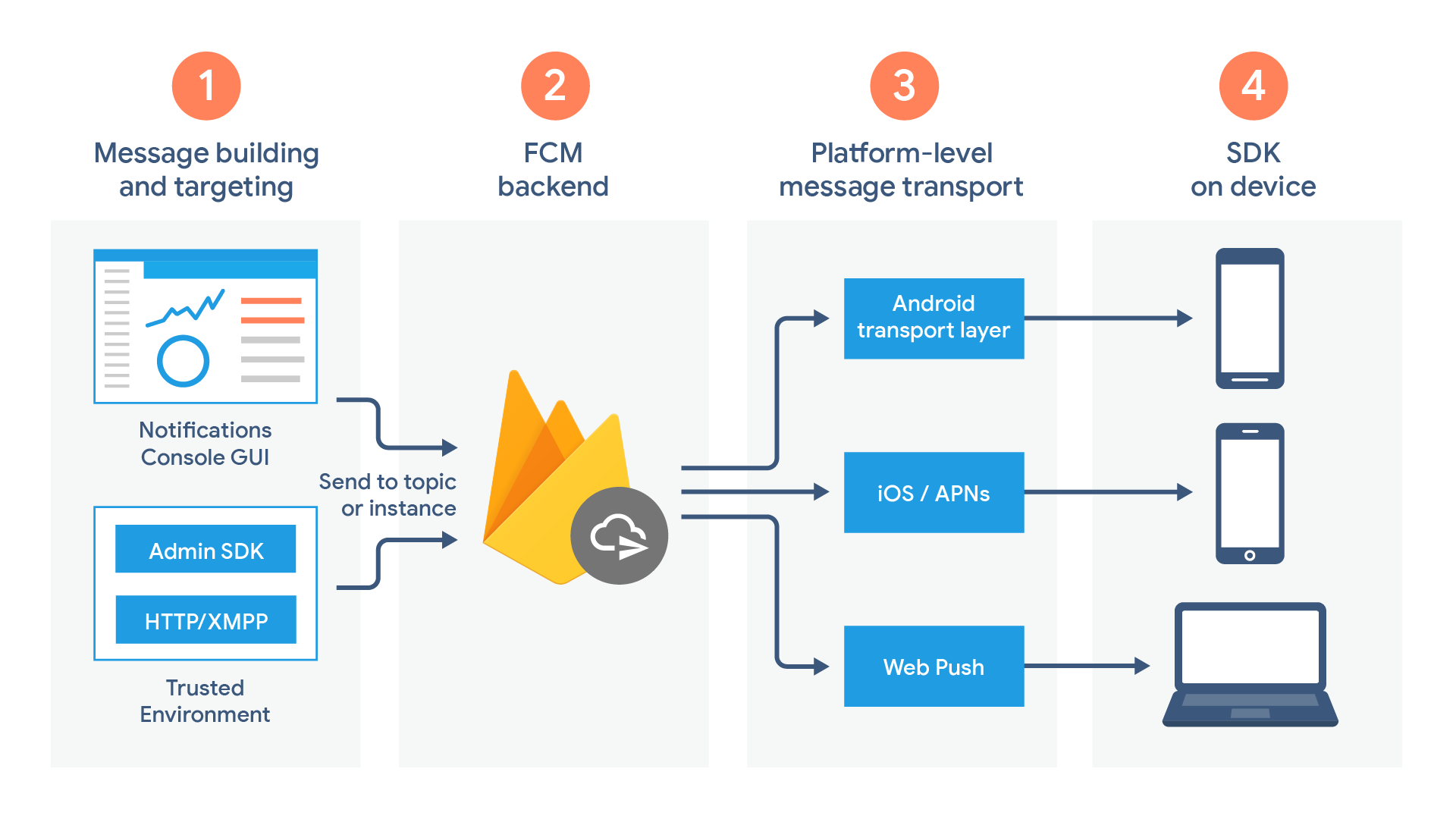


*Hình 1.4. Mô hình hoạt động MongoDB*

* MongoDB hoạt động dưới một tiến trình ngầm service luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.
* Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau.
* Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
* Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.
* Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.
* Ưu điểm của MongoDB
* Document oriented
* Hiệu suất cao
* Tính sẵn sàng cao – Nhân rộng
* Khả năng mở rộng cao – Sharding
* Năng động – Không có lược đồ cứng nhắc.
* Linh hoạt – thêm / xóa trường có ít hoặc không ảnh hưởng đến ứng dụng
* Dữ liệu không đồng nhất
* Không joins
* Phân phối được
* Biểu diễn dữ liệu trong JSON hoặc BSON
* Hỗ trợ không gian địa lý (Geospatial)
* Tích hợp dễ dàng với BigData Hadoop
* Ngôn ngữ truy vấn dựa trên tài liệu mạnh mẽ như SQL
* Các bản phân phối cloud như AWS, Microsoft, RedHat, dotCloud và SoftLayer, v.v … Trên thực tế, MongoDB được xây dựng cho cloud. Kiến trúc mở rộng quy mô tự nhiên của nó, được kích hoạt bởi sharding, liên kết tốt với quy mô và sự nhanh nhẹn có được nhờ điện toán đám mây.
* Sử dụng MongoDB
* Danh mục sản phẩm thương mại điện tử.
* Blog và quản lý nội dung.
* Phân tích thời gian thực và ghi nhật ký tốc độ cao, bộ nhớ đệm và khả năng mở rộng cao.
* Quản lý cấu hình.
* Duy trì dữ liệu dựa trên vị trí – Dữ liệu không gian địa lý.
* Các trang web di động và mạng xã hội.
* Phát triển yêu cầu dữ liệu.
* Mục tiêu không chặt chẽ – thiết kế có thể thay đổi theo thời gian.
  1. **FIREBASE CLOUD MESSAGING**

Firebase Cloud Messaging (FCM) dựa trên tập hợp các thành phần sau đây để xây dựng, vận chuyển và nhận tin nhắn:

(1) Công cụ để soạn hoặc xây dựng yêu cầu tin nhắn. Trình soạn thảo thông báo cung cấp tùy chọn dựa trên GUI để tạo yêu cầu thông báo. Để tự động hóa hoàn toàn và hỗ trợ cho tất cả các [loại tin nhắn](https://firebase.google.com/docs/cloud-messaging/concept-options#notifications_and_data_messages) , bạn phải xây dựng các yêu cầu tin nhắn trong [môi trường máy chủ](https://firebase.google.com/docs/cloud-messaging/server) đáng tin cậy hỗ trợ SDK quản trị Firebase hoặc các giao thức máy chủ FCM. Môi trường này có thể là Chức năng đám mây cho Firebase, Google App Engine hoặc máy chủ ứng dụng của riêng bạn.

*Hình 1.5. Hoạt động Firebase Cloud Messaging*

(2) Chương trình phụ trợ FCM, (trong số các chức năng khác) chấp nhận yêu cầu tin nhắn, thực hiện các tin nhắn thông qua các chủ đề và tạo siêu dữ liệu tin nhắn như ID tin nhắn.

(3) Lớp vận chuyển ở cấp độ nền tảng, định tuyến tin nhắn đến thiết bị được nhắm mục tiêu, xử lý việc gửi tin nhắn và áp dụng cấu hình dành riêng cho nền tảng khi thích hợp. Lớp vận chuyển này bao gồm:

* Lớp vận chuyển Android (ATL) cho các thiết bị Android có dịch vụ Google Play
* Dịch vụ thông báo đẩy của Apple (APN) cho thiết bị iOS
* Giao thức đẩy web cho các ứng dụng web

(4) SDK FCM trên thiết bị của người dùng, nơi thông báo được hiển thị hoặc thông báo được xử lý theo trạng thái nền trước / nền của ứng dụng và mọi logic ứng dụng có liên quan.

* **Lưu lượng vòng đời gửi và nhận tin nhắn.**

Máy chủ ứng dụng gửi tin nhắn đến ứng dụng khách:

(1) Tin nhắn được soạn, trong môi trường soạn thảo Thông báo hoặc môi trường tin cậy và yêu cầu tin nhắn được gửi đến phụ trợ FCM.

(2) Phần cuối FCM nhận yêu cầu tin nhắn, tạo ID tin nhắn và siêu dữ liệu khác và gửi nó đến lớp vận chuyển cụ thể của nền tảng.

(3) Khi thiết bị trực tuyến, tin nhắn được gửi qua lớp vận chuyển dành riêng cho nền tảng đến thiết bị.

(4) Trên thiết bị, ứng dụng khách sẽ nhận được tin nhắn hoặc thông báo.

* 1. **JSON WEB TOKEN**

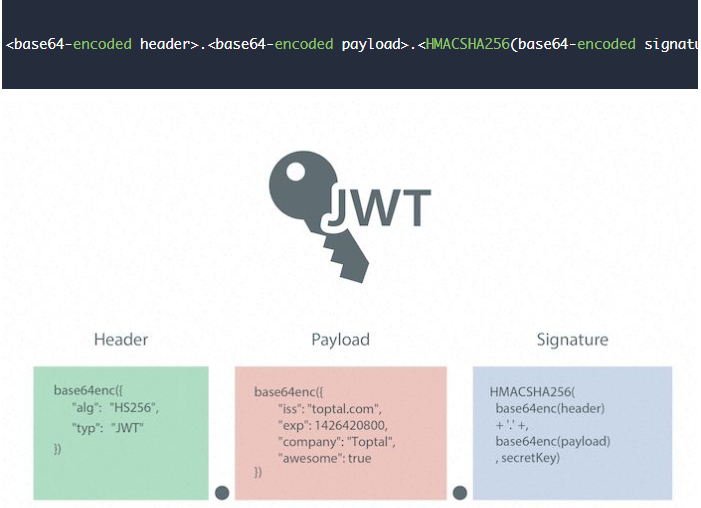
JSON Web Token (JWT) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán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

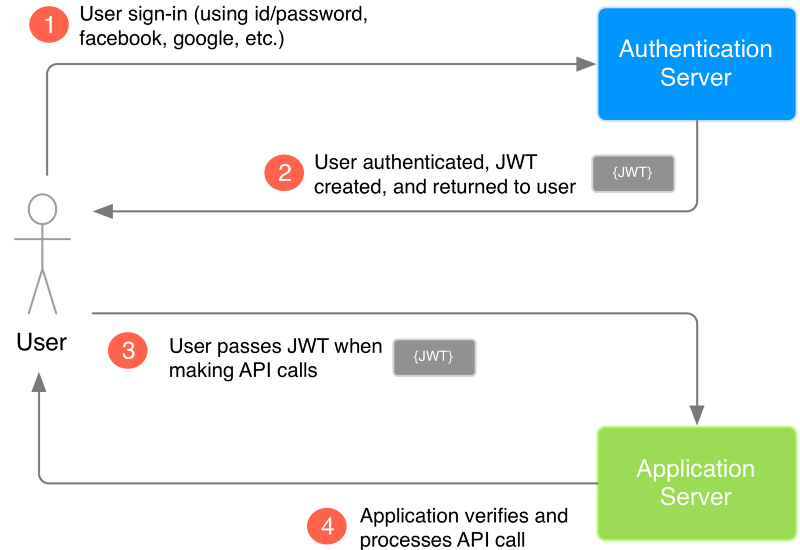
- Payload chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ. Có 3 loại claims thường gặp trong Payload: reserved, public và private claims.

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



*Hình 1.6. Cấu trúc của Json Web Token*

**Luồng xử lý của 1 hệ thống sử dụng bảo mật JWT**



*Hình 1.7. Luồng xử lý bảo mật JWT*

1. User thực hiện login bằng cách gửi id/password hay sử dụng các tài khoản mạng xã hội lên phía Authentication Server (Server xác thực)
2. Authentication Server tiếp nhận các dữ liệu mà User gửi lên để phục vụ cho việc xác thực người dùng. Trong trường hợp thành công, Authentication Server sẽ tạo một JWT và trả về cho người dùng thông qua response.
3. Người dùng nhận được JWT do Authentication Server vừa mới trả về làm “chìa khóa” để thực hiện các “lệnh” tiếp theo đối với Application Server.
4. Application Server trước khi thực hiện yêu cầu được gọi từ phía User, sẽ verify JWT gửi lên. Nếu OK, tiếp tục thực hiện yêu cầu được gọi.
   1. **GOOGLE MAP API**

Google Maps là một ứng dụng bản đồ trực tuyến trên web được cung cấp miễn phí bởi Google. Nó có thể dùng để nhúng vào các trang web của bên thứ ba thông qua Google Maps API. Nó cung cấp rất nhiều những tiện ích nâng cao giúp người dùng tìm kiếm đường đi, các địa điểm, các địa danh...

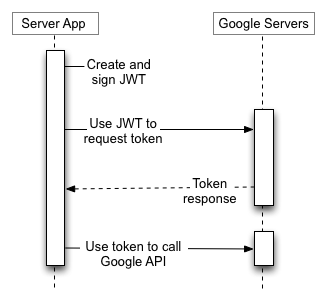
Map API là một phương thức cho phép 1 website B sử dụng dịch vụ bản đồ của website A (gọi là Map API) và nhúng vào website của mình (site B). Site A ở đây là google map, site B là các website cá nhân hoặc tổ chức muốn sử dụng dịch vụ của google (di chuột, room, đánh dấu trên bản đồ…)

Các ứng dụng xây dựng trên maps được nhúng vào trang web cá nhân thông qua các thẻ java scripts do vậy việc sử dụng API google rất dễ dàng.

* Một số ứng dụng của Google Map API:
* Đánh dấu các địa điểm trên bản đồ kèm theo thông tin cho địa điểm đó : khu vui chơi giải trí, nhà hàng khách sạn, cây ATM, bệnh viện, trường học,…
* Chỉ dẫn đường đến các địa điểm cần tìm (đường tối ưu và nhiều lựa chọn khác), chỉ dẫn đường giao thông công cộng, có thể là các địa điểm cung cấp như trên. Ở đây sử dụng các service google cung cấp.
* Khoanh vùng khu vực: các trung tâm kinh tế, khu đô thị, khu ô nhiễm…
* Tình trạng giao thông các khu vực, từ đó đưa ra các giải pháp.
  1. **GOOGLE SIGN-IN**

Google Sign-in là một hệ thống xác thực an toàn giúp giảm gánh nặng đăng nhập cho người dùng, bằng cách cho phép họ đăng nhập bằng tài khoản Google của họ cùng một tài khoản họ đã sử dụng với Gmail, Play và các dịch vụ khác của Google.

Đăng nhập Google cũng là cửa ngõ để kết nối với người dùng và dịch vụ của Google một cách an toàn. Cung cấp cho người dùng cơ hội thanh toán bằng Google Pay, chia sẻ với các liên hệ trên toàn Google của họ, lưu tệp vào Drive, thêm sự kiện vào lịch và hơn thế nữa. Tích hợp các API và dịch vụ lấy người dùng làm trung tâm của Google trong ứng dụng của bạn để giúp người dùng của bạn hành động và chuyển đổi.



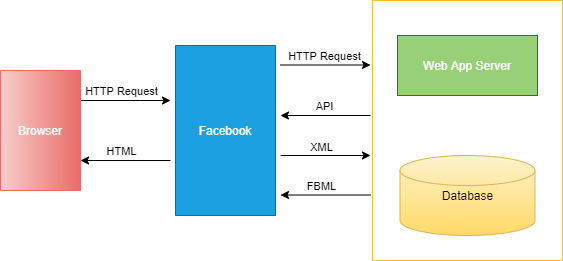
*Hình 1.8. Hoạt động của Google API*

* 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. API sử dụng giao thức RESTful và các hồi đáp được trả lại dưới dạng XML.

**Cách thức hoạt động của Facebook API .**

API là nền tảng do Facebook cung cấp cho người viết ứng dụng để dễ dàng trong việc tạo ứng dụng và đảm bảo người viết ứng dụng không can thiệp quá sâu vào hệ thống của Facebook. API cho phép thực thi nền tảng thông qua các phương thức được định nghĩa. Thông qua các lời gọi API, người tạo ứng dụng có thể lấy thông tin về user, groups, photo,… mà họ cần.



*Hình 1.9. Hoạt động Facebook API*

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.

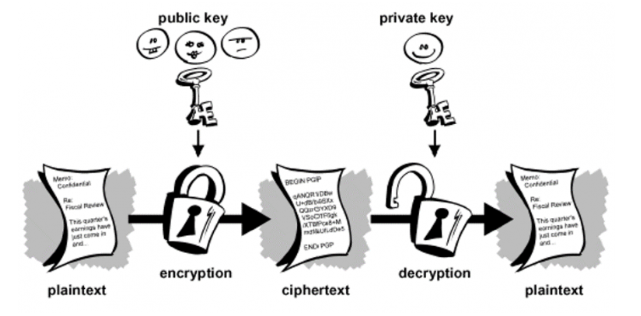
**Một số phương thức của Facebook API**

* **User**: Facebook API cung cấp một số phương thức cho phép nhà phát triển ứng dụng truy cập vào một số thông tin người dùng:
* Getinfo(): Lấy thông tin của người dùng, như name (họ tên), first\_name, last\_name, id, email… Ví dụ lấy họ tên người dùng ta dùng: API.user.getinfo().name sẽ trả về họ tên user. Một số thông tin cần được cho phép của người dùng thì mới có thể lấy được, ví dụ email, address, phone…
* GetLoggedInUser(): Trả về uid của người hiện tại đang đăng nhập ứng dụng. GetStandardInfo(): Lấy thông tin cơ bản của người dùng như name, first\_name, last\_name…
* Friends: Khi phát triển ứng dụng, việc hiển thị danh sách bạn bè của người dùng là rất phổ biến. Để giải quyết vấn đề này API có các hàm giúp người viết ứng dụng lấy thông tin về bạn bè của người dùng.
* areFriend(): Kiểm tra xem 2 user của phải là bạn của nhau hay không. Ở đây truyền vào 2 đối số là 2 uid hoặc 2 user hoặc 2 danh sách (list) user, uid. Kiểu trả về là dạng boolean. Ví dụ: bool result = API.friends.areFriend(123123,3990023);
* get(): Trả lại danh sách uid của người dùng là bạn của người dùng hiện tại.
* getAppUsers(): Lấy danh sách uid của người dùng đang sử dụng ứng dụng của bạn.
* getAppUsersObjects(): Trả về danh sách các user đang sử dụng ứng dụng.
* getLists(): Trả về freindslisst của người dùng hiện tại.
* **Groups**: Làm việc với nhóm cũng là một phần rất quan trọng của ứng dụng. API đã cung cấp các phương thức:
* Get(): Trả về các nhóm theo tiêu chí lọc như nhóm có liên kết với một người dùng.
* getMember(): Trả về danh sách thành viên của một nhóm. Kết quả trả về là một mảng đa chiều với chiều đầu tiên là kiểu thành viên (admin, member, officer, not\_replied). Notifications: Facebook cho phép gửi và nhận các thông báo trong ứng dụng với các hàm API. Có 2 hàm chính:
* Get():Trả về một mảng đa chiều với các đối tượng là các loại thông báo: tin nhắn, giới thiệu nhóm, giới thiệu sự kiện, yêu cầu kết bạn, chia sẻ, poke).
* getList(): Trả về tất cả thông tin của người dùng hiện tại. Photos: Với hơn 60 triệu bức ảnh được thêm vào mỗi tuần bởi người dùng Facebook, các nhà phát triển Facebook đã cung cấp một nhóm phương thức API để giúp các nhà phát triển ứng dụng dễ dàng thao tác với các ảnh của người dùng.
* addTag(): Thêm một thẻ đi kèm thông tin cho một bức ảnh. Thẻ chỉ có thể thêm vào những bức ảnh đang chờ giải quyết thuộc sở hữu của người dùng cụ thể hoặc người dùng hiện hành. Ứng dụng với quyền mở rộng photo\_upload có thể thêm thẻ vào bất kì bức ảnh nào thuộc quyền sở hữu của người dùng.
* createAlbums():Tạo và trả về một album mới thuộc quyền sở hữu của một người dùng cụ thể hoặc người dùng hiện hành.
* Get(): Trả về tất cả các ảnh theo tiêu chí lọc.
* getAlbum(): Trả về siêu dữ liệu về tất cả các album ảnh được tải lên bởi người dùng cụ thể. Phương thức này trả về thông tin của tất cả các album dựa theo tiêu chí lọc.
* getTags(): Trả về danh sách người dùng được đánh dấu của các hình ảnh cụ thể.
* upload(): upload một ảnh thuộc sở hữu của người dùng cụ thể hoặc người dùng hiện hành và trả về là một ảnh mới. Các loại ảnh hỗ trợ như: gif, jpg, png, psd, tiff, jp2, iff, wbmp, xbm.
* **Profile**: Để dễ dàng tương tác với các thiết lập thông tin trong trang hồ sơ của người dùng, nhóm hàm này đã cung cấp 2 phương thức:
* setFBML(): Thiết lập FBML cho hồ sơ của người dùng hoặc trang Facebook , bao gồm: nội dung của hộp thông tin trên nhãn Wall, hộp thông tin cho nhãn Boxes.
* getFBML(): Trả lại FBML của hộp thông tin hiện đang thiết lập cho cấu hình của người dùng.
* **Stream:** Để thực hiện các thao tác cập nhật thông tin như đăng bài, bình luận,... tuy nhiên phải tuân theo quyền được phép thực hiện các thao tác từ phía người khác. Nó gồm các phương thức cơ bản như:
* addComment(): Cho phép thêm bình luận vào bài đã được đăng trên tường của người nào đó.
* addLike(): Dùng để like một bài viết nào đó, đối số truyền vào chính là id của bài viết đó – post\_id.
* getComments(): Lấy danh sách các comment trong một bài bình luận, tham số truyền vào chính là post\_id – id bài cần lấy thông tin comment.
* Publish(): Cho phép đăng lên tường ai đó, nội dung đăng có thể là đoạn text hoặc kèm theo hình ảnh hay link,…
* Remove(): Xóa bài đăng nào đó, thông số truyền vào là post\_id là id của bài cần xóa, có thể kèm theo uid để xác định xóa bài của người nào.
* removeComment(): Là xóa bình luận của người dùng nào đó trong một bài bình luân, thông số truyền vào là comment\_id là id của lời bình luận cần xóa.
* removeLike(): Tương tự như trên, removeLike() dùng để unlike bài viết nào đó mà trước đó đã like.

1. **LÝ THUYẾT**
2. **MOMO PAYMENT API**
3. **THUẬT TOÁN RSA**

Mã hóa RSA là thuật toán mã hóa được sử dụng phổ biến hiện nay. Mã hóa RSA là mã hóa khóa công khai (public-key encryption), tức là RSA sử dụng 2 khóa khác nhau để mã hóa và giải mã: 1 khóa công khai (public key – mọi người đều biết) và 1 khóa bí mật (private key – chỉ có người nhận biết). Bất kỳ ai cũng có thể sử dụng khóa công khai để mã hóa dữ liệu cần gửi. Tuy nhiên, để giải mã dữ liệu đó, cần phải có khóa bí mật, và chỉ có người nhận mới có khóa đó. Khác với mã hóa khóa đối xứng (symmetric-key encryption), khóa bí mật không bao giờ được truyền ra ngoài, do đó dữ liệu đã mã hóa, dù bị nghe trộm, cũng không thể bị giải mã. RSA được dùng trong mọi tình huống cần bảo mật và bảo đảm, bao gồm giao thức SSL/TSL (kể cả HTTPS) và chứng chỉ điện tử (digital certificate).

Hoạt động của RSA dựa trên 4 bước chính: sinh khóa, chia sẻ khóa, mã hóa và giải mã.



*Hình 1.10. Mã hóa thông tin*

* Bản rõ (plaintext) : là thông tin chưa được mã hóa hay thông tin ở dạng có thể hiểu được.
* Mã hóa (Encryption): là hành động xáo trộn bản rõ để chuyển thành bản mã.
* Bản mã (ciphertext hay encrypted text) : là thông tin đã được mã hóa hay thông
* tin ở dạng đã bị xáo trộn.
* Giải mã (Decryption): là hành động giải xáo trộn bản mã để chuyển thành bản rõ.
* Mã hóa sử dụng một thuật toán (Algorithm) để mã hóa thông tin.

1. **THUẬT TOÁN SHA-256**

SHA-256 là thuật toán băm bảo mật 256 bit và dùng để tạo ra các hàm băm không thể đảo ngược và duy nhất. Số lượng hàm băm có thể có càng lớn, thì xác suất để hai giá trị sẽ tạo ra cùng một giá trị băm càng nhỏ.

Thuật toán SHA-256 tạo ra một mã băm có cố định 256-bit (32-byte) gần như duy nhất. Một chuỗi băm được tạo ra không thể được tính toán trở lại. Điều này làm cho nó phù hợp để xác nhận mật khẩu, thách thức xác thực hàm băm, chống giả mạo, chữ ký số.

SHA-256 là một trong những hàm băm kế tiếp đối sau SHA-1 và là một trong những hàm băm mạnh nhất hiện tại.

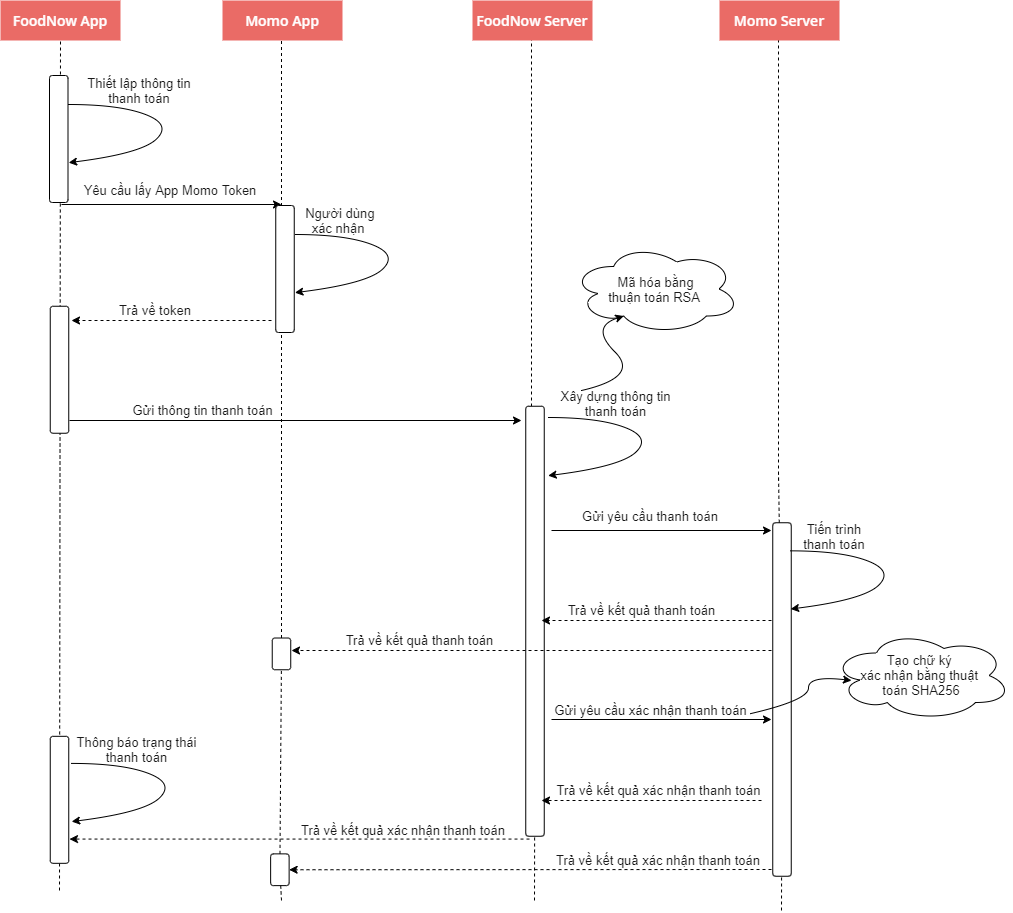
Trong thuật toán SHA, người ta xác định giá trị băm ban đầu, thông tin có thể sử dụng để biến đổi giá trị ban đầu này là dữ liệu đầu vào.

Trong trường hợp SHA-256, dữ liệu đầu vào được chia ra thành nhiều phần, mỗi phần có độ dài 64 byte.

* Dữ liệu được chia ra được gọi là “message block” (khối thông báo, thông tin).
* Khi dữ liệu đầu vào vượt quá 64 byte, nhiều message block sẽ được tạo thành.
* Tiếp theo, message block đầu tiên được sử dụng làm thay đổi giá trị băm ban đầu để tính ra giá trị băm mới.
* Sau đó, nó được xử lý theo phương thức giá trị băm tạo thành sẽ được biến đổi bằng message block tiếp theo để cho ra giá trị mới, và tiếp tục lặp lại với lần lượt các message block,…
* Giá trị băm được đưa ra bởi message block cuối là kết quả cuối cùng.

Cứ như vậy, bằng cách sử dụng dữ liệu đầu vào được chia ra thành những phần 64 byte và biến đổi giá trị băm, chúng ta có thể thu được những giá trị có cùng độ dài bit bất kể độ dài dữ liệu đầu vào.

1. **SƠ ĐỒ XỬ LÝ THANH TOÁN TRỰC TUYẾN VÍ MOMO**



*Hình 1.11. Sơ đồ xử lý thanh toán trực tuyến ví Momo*

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 1.12. Vòng đời của một Activity*

1. **CODEIGNITER WEB FRAMEWORK**

CodeIgniter là framework PHP mạnh mẽ với footprint rất nhỏ, dành cho các lập trình viên cần một bộ công cụ đơn giản và thông minh để tạo ra các ứng dụng web với đầy đủ các tính năng. [Codeigniter Framework](https://blog.webico.vn/codeigniter-framework-la-gi-tong-quan-ve-codeigniter-framework/) cũng được đánh giá là Framework có tốc độ hoạt động nhanh, hiệu quả.

Đặc điểm nổi bật:

* Được thiết kế theo mô hình MVC, nó giúp cho việc thiết kế, xử lý và bảo trì mã nguồn dễ dàng, đồng thời tăng khả năng mở rộng của phần mềm. CodeIgniter vận dụng mô hình này trong thiết kế, giúp tách biệt các tập tin giao diện với các tập tin xử lý dữ liệu, nâng cao khả năng quản lý và dễ bảo trì.
* Nhỏ gọn, tốc độ nhanh, miễn phí.
* Hệ thống thư viện phong phú, hỗ trợ search engine optimization.
* Bảo mật tốt.

1. **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.

* **Các loại cơ sở dữ liệu**
* Key-value: Cơ sở dữ liệu key-value có khả năng phân vùng cao và cho phép mở rộng theo tỷ lệ mà các loại cơ sở dữ liệu khác không thể đạt được. Các trường hợp sử dụng như chơi game và công nghệ quảng cáo cực kỳ phù hợp cho mô hình key-value.
* Tài liệu: Sự phổ biến của cơ sở dữ liệu tài liệu đã tăng lên vì các nhà phát triển có thể duy trì dữ liệu, bằng cách sử dụng cùng định dạng mô hình tài liệu trong mã ứng dụng. DynamoDB và MongoDB là các cơ sở dữ liệu tài liệu phổ biến cung cấp các API mạnh và trực quan để phát triển linh hoạt [impotenciastop.pt](https://impotenciastop.pt/).
* Biểu đồ: Mục đích của cơ sở dữ liệu đồ thị là giúp dễ dàng xây dựng và chạy các ứng dụng hoạt động với các bộ dữ liệu được kết nối cao.
* In-memory: Phù hợp với các ứng dụng trò chơi và công nghệ quảng cáo có các trường hợp sử dụng như bảng xếp hạng, cửa hàng online và phân tích thời gian thực yêu cầu thời gian phản hồi tính bằng đơn vị micro giây và có lưu lượng truy cập tăng đột biến bất cứ lúc nào.

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.

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:

*Hình 1.13. Mô hình hoạt động của RESTful API*

* 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.

* **HTTP GET**

Sử dụng các yêu cầu GET **để chỉ truy xuất đại diện / thông tin tài nguyên** - và không sửa đổi nó theo bất kỳ cách nào. Vì các yêu cầu GET không thay đổi trạng thái của tài nguyên, đây được coi là **các phương thức an toàn** .

Nếu URI yêu cầu đề cập đến một quy trình sản xuất dữ liệu, thì đó là dữ liệu được tạo sẽ được trả về làm thực thể trong phản hồi và không phải là văn bản nguồn của quy trình, trừ khi văn bản đó là đầu ra của quy trình.

Đối với bất kỳ API HTTP GET đã cho nào, nếu tìm thấy tài nguyên trên máy chủ, thì nó phải trả về mã phản hồi HTTP (200-OK) cùng với phần thân phản hồi, thường là nội dung XML hoặc JSON (do tính chất độc lập với nền tảng của chúng).

Trong trường hợp tài nguyên KHÔNG được tìm thấy trên máy chủ thì nó phải trả về mã phản hồi HTTP 404 (NOT FOUND). Tương tự, nếu xác định rằng chính yêu cầu GET không được hình thành chính xác thì máy chủ sẽ trả về mã phản hồi HTTP 400 (BAD REQUEST).

* **HTTP POST**

Sử dụng POST APIs để tạo tài nguyên cấp dưới mới , ví dụ: tệp phụ thuộc vào thư mục chứa nó hoặc một hàng phụ thuộc vào bảng cơ sở dữ liệu. Khi nói một cách nghiêm túc về REST, các phương thức POST được sử dụng để tạo một tài nguyên mới vào bộ sưu tập tài nguyên.

Lý tưởng nhất là nếu tài nguyên đã được tạo trên máy chủ gốc, mã phản hồi là HTTP 201 (Created) và chứa một thực thể mô tả trạng thái của yêu cầu và tham chiếu đến tài nguyên mới và tiêu đề [vị trí](https://en.wikipedia.org/wiki/HTTP_location) .

Nhiều lần, hành động được thực hiện bởi phương thức POST có thể không dẫn đến một tài nguyên có thể được xác định bởi một URI. Trong trường hợp này, mã phản hồi HTTP 200 (OK) hoặc 204 (No Content) là trạng thái phản hồi thích hợp.

Các phản hồi cho phương thức này không được lưu trong bộ nhớ cache , trừ khi phản hồi bao gồm các trường tiêu đề [kiểm soát bộ](https://en.wikipedia.org/wiki/Web_cache#Cache_control)[đệm](https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html) hoặc hết [hạn](https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html) thích hợp .

* **HTTP PUT**

Sử dụng PUT APIs chủ yếu để cập nhật tài nguyên hiện có (nếu tài nguyên không tồn tại, thì API có thể quyết định tạo tài nguyên mới hay không). Nếu tài nguyên mới được tạo bởi API PUT, máy chủ gốc PHẢI thông báo cho tác nhân người dùng thông qua phản hồi mã phản hồi HTTP 201 (Created) và nếu tài nguyên hiện có bị sửa đổi, 200 (OK) hoặc 204 (No Content) để cho biết hoàn thành yêu cầu .

Nếu yêu cầu chuyển qua bộ đệm và URI yêu cầu xác định một hoặc nhiều thực thể hiện được lưu trong bộ nhớ cache. Phản hồi cho phương pháp này không được lưu trữ .

* **HTTP DELETE**

Như tên được áp dụng, DELETE APIs được sử dụng để xóa tài nguyên (được xác định bởi URI yêu cầu).

Phản hồi thành công của các yêu cầu DELETE là phản hồi HTTP code 200 (OK) nếu phản hồi bao gồm một thực thể mô tả trạng thái, 202 (Accepted) nếu hành động đã được xếp hàng hoặc 204 (No Content) nếu hành động đã được thực hiện nhưng phản hồi không bao gồm thực thể.

Nếu DELETE một tài nguyên, nó sẽ bị xóa khỏi bộ sưu tập tài nguyên. Việc liên tục gọi DELETE API trên tài nguyên đó sẽ không thay đổi kết quả. Tuy nhiên, việc gọi DELETE trên tài nguyên lần thứ hai sẽ trả về 404 (NOT FOUND) vì nó đã bị xóa.

Nếu yêu cầu chuyển qua bộ đệm và URI yêu cầu xác định một hoặc nhiều thực thể hiện được lưu trong bộ nhớ cache. Phản hồi cho phương pháp này không được lưu trữ .

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 USE CASE**

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.

# **KẾT PHẦN 1**

Từ việc tìm hiểu công nghệ, nhóm đã ứng dụng công nghệ vào đề tài như sau:

* Tích hợp Google Map:
  + Trong việc tìm kiếm các cửa hàng gần khách hàng nhất.
  + Khách hàng có thể thêm, thay đổi địa chỉ.
  + Tính khoảng cách từ cửa hàng đến vị trí của khách hàng, để khách hàng dễ dàng lựa chọn cửa hàng mình cần đặt món. Sắp xếp được các cửa hàng theo cấp độ tăng dần theo khoảng cách.
* Tích hợp Google API cụ thể là Google Sign-in
  + Đăng ký, đăng nhập tài khoản.
* Tích hợp Facebook API
  + Đăng ký, đăng nhập tài khoản.
* Ứng dụng Json Web Token
  + Mã hóa bảo mật tài khoản.
* Ứng dụng Firebase Cloud Messaging
  + Thông báo đơn hàng mới đến cửa hàng.
  + Thông báo đến khách hàng là cửa hàng đã xác nhận/ hủy đơn hàng.
  + Thông báo đến cửa hàng là khách hàng đã hủy đơn hàng.
  + Hiển thị thông báo khuyến mãi của cửa hàng đến khách hàng.
  + Gửi thông báo đến khách hàng thanh toán thành công bằng ví Momo.
* Sử dụng thuật toán RSA và SHA256, tích hợp Momo Payment API vào thanh toán trực tuyến bằng ví Momo.

**PHẦN 2: TRIỂN KHAI ĐỀ TÀI**

## **CHƯƠNG 1: TỔNG QUAN**

1. **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 đáp ứng nhu cầu phục vụ tận tay của khách hàng, 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 Now, Grab, GoViet, Baemin,… 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 Json Web Token để bảo mật, mã hóa tài khoản.
* Ứng dụng Firebase để thông báo đơn hàng, khuyến mãi.
* Tích hợp Google Map để tìm địa chỉ, tính khoảng cách.
* Sử dụng thuật toán RSA và SHA256 vào thanh toán trực tuyến bằng ví Momo.
* 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 chức năng nghiệp vụ của hệ thống.

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, hủy 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, đặt món ăn, xem được thông báo khuyến mãi của cửa hàng, hệ thống, xem lại các đơn hàng đã đặt, đã hủy. Ngoài phần mềm trung gian giữa khách hàng và cửa hàng. Nhóm cũng thiết kế website dành cho quản trị viên hệ thống để quản lý các cửa hàng, cấp tài khoản cho cửa hàng, quản lý các danh mục, quản lý khuyến mãi.

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

Nhóm xây dựng hệ thống 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.
  + Thanh toán trực tuyến bằng ví Momo.
  + 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 Now và các ứng dụng khác như Baemin, GrabFood, Foody,… để thực hiện gồm: phân tích hệ thống, thiết kế ứng dụng và lập trình các chức năng nghiệp vụ.

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**

Để tạo hệ thống, nhóm đã tiến hành khảo sát ứng dụng vận chuyển hàng của Now vì hệ thống này có nhiều điểm tương đồng, dễ khảo sát và gần với thực tế. Với một hệ thống lớn sẽ đầy đủ các quy trình từ phía người dùng tác động đến hệ thống.

Now.vn (tiền thân là Deliverynow), một thành viên của mạng xã hội ẩm thực lớn nhất tại Việt Nam - Foody và được biết đến là một trong những đối thủ nặng ký trong lĩnh vực đặt món và giao hàng trực tuyến.

Now hoạt động trên cơ chế liên kết với các nhà hàng, quán ăn, khi có đơn hàng được đặt, hệ thống giao hàng riêng biệt của Now sẽ tiến hành mua và giao đến cho khách hàng.

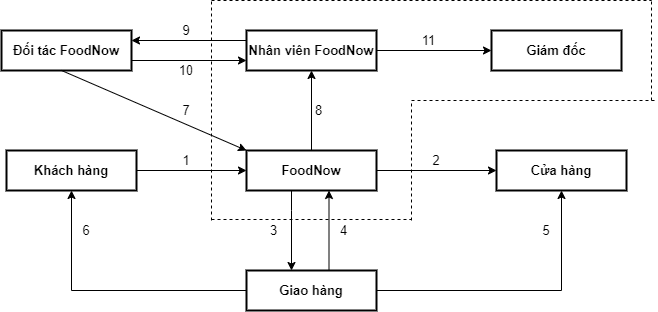
Dù khởi động muộn hơn so với các đối thủ trong cùng lĩnh vực, Now vẫn tạo ra được sự khác biệt nhờ sở hữu các lợi thế: Hệ thống cửa hàng đa dạng cho đối tượng giới trẻ và tận đụng được các phản hồi, đánh giá hữu ích từ Foody.

Bên cạnh đó, Now ngày càng được nâng cấp thêm nhiều tính năng mới, đáp ứng nhu cầu sử dụng rộng rãi của khách hàng Việt như: Đặt bàn, mua thực phẩm, hoa, bia, rượu, giặt ủi,…

Cùng với sự kiện được Sea (Singapore) mua lại, Now đang định hướng cho mình chiến lược phát triển mảng gọi xe. Mục tiêu trong tương lai của Now là trở thành nhà cung cấp cấp dịch vụ hằng ngày cho khách hàng trên toàn khu vực Đông Nam Á.

Hệ thống Now cũng như GrabFood, Baemin,… 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.

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

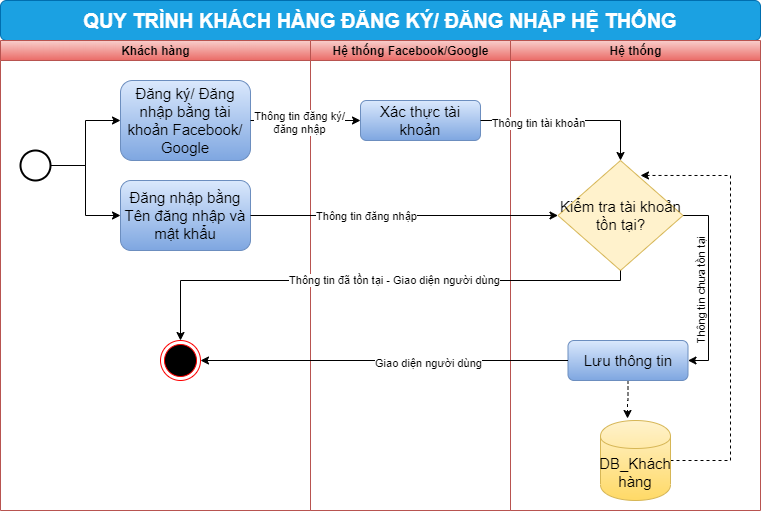


*Hình 2.1.1. Sơ đồ hệ thống FoodNow*

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 sẽ đăng ký làm đối tác trên trang hệ thống FoodNow.
      8. Hệ thống sẽ thông báo đến cho nhân viên trong hệ thống.
      9. Nhân viên hẹn ngày ký hợp đồng.
      10. Đối tác ký hợp đồng và được cấp tài khoản để đăng nhập hệ thống.
      11. Nhân viên tổng hợp báo cáo giám đốc.
    1. MÔ HÌNH NGHIỆP VỤ BPM
       - 1. Quy trình đăng ký, đăng nhập phía khách hàng

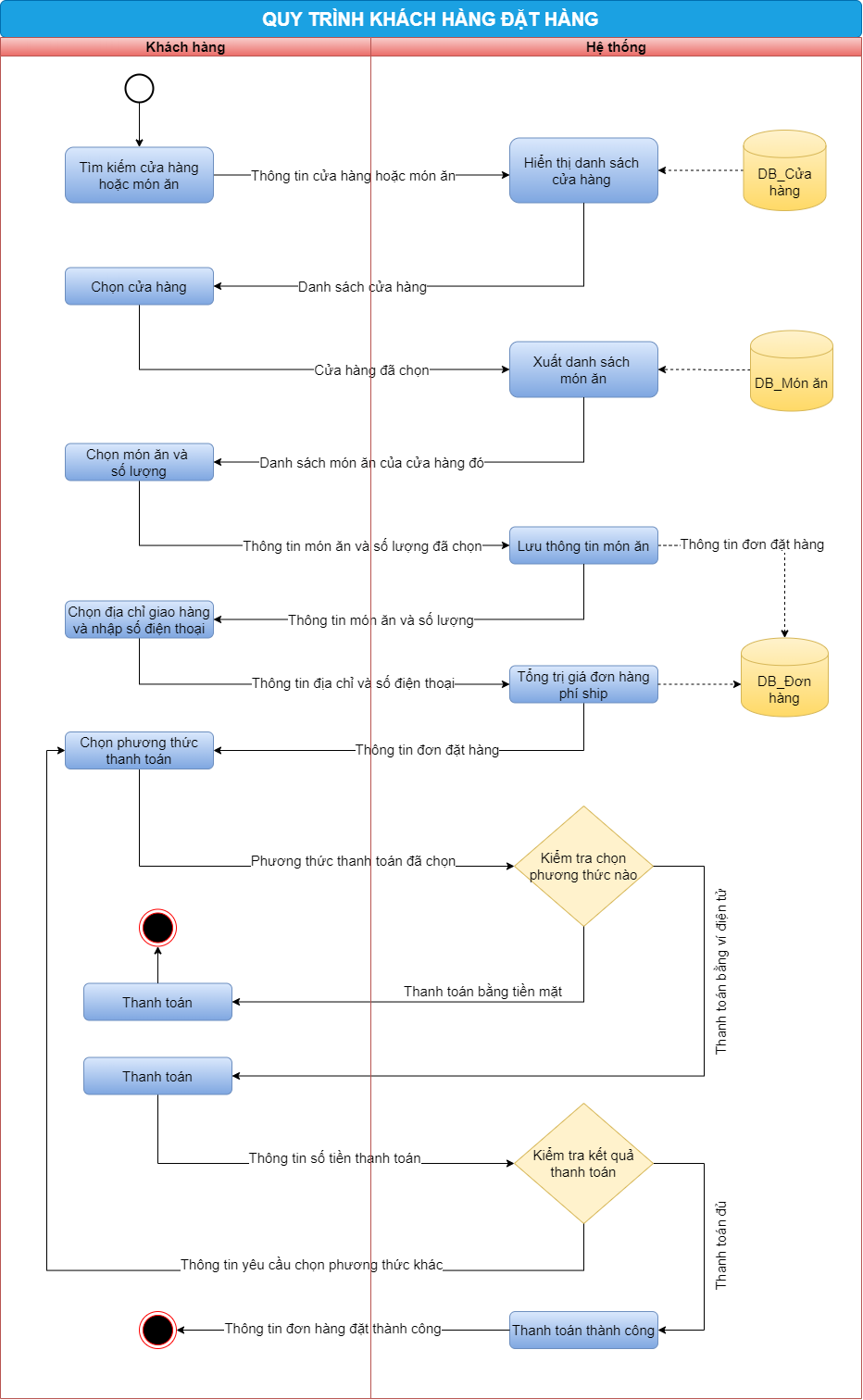
Khách hàng vào hệ thống lần đầu sẽ đăng ký tài khoản một trong hai tài khoản là Google hoặc Facebook mà hệ thống cung cấp. Hệ thống Google/ Facebook sẽ xác thực tài khoản. Hệ thống FoodNow sẽ kiểm tra tài khoản này chưa có trong hệ thống, hệ thống sẽ lưu thông tin xuống dữ liệu khách hàng và trả về cho khách hàng giao diện ứng dụng. Khi khách hàng đăng nhập bằng tài khoản Google/ Facebook, hệ thống Google/ Facebook sẽ xác thực tài khoản, hệ thống FoodNow kiểm tra tài khoản đã có và trả về cho khách hàng giao diện người dùng. Việc đăng ký bằng tài khoản Google/ Facebook hệ thống FoodNow cũng sẽ cung cấp cho người dùng một tài khoản gồm tên đăng nhập và mật khẩu nằm trong thông tin người dùng. Nếu khách hàng không đăng nhập bằng hai tài khoản Google/Facebook, khách có thể đăng nhập bằng tên đăng nhập và mật khẩu hệ thống cấp.



*Hình 2.1.2. Quy trình khách hàng đăng ký/ đăng nhập hệ thống*

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

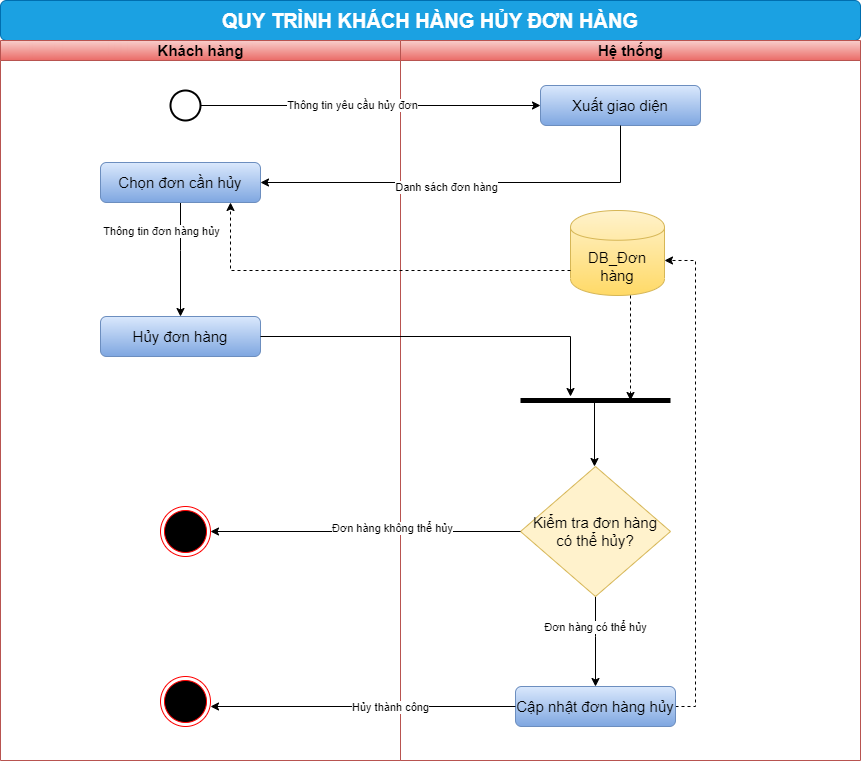
Khách hàng tìm kiếm cửa hàng hoặc món ăn. Hệ thống sẽ hiển thị các cửa hàng mà khách hàng cần tìm kiếm. Khách hàng chọn cửa hàng và danh sách món ăn của cửa hàng hiển thị. Khách hàng chọn món ăn và món ăn đó sẽ được thêm vào giỏ hàng và lưu vào dữ liệu đơn hàng. Khách hàng muốn giao hàng sẽ phải nhập địa chỉ và số điện thoại. 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, khách hàng bấm chọn thanh toán, hệ thống sẽ kiểm tra kết quả thanh toán là thành công và thông báo cho khách hàng, nếu không thành công khách hàng sẽ chọn lại phương thức thanh toán.



*Hình 2.1.3. Quy trình khách hàng đặt hàng*

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

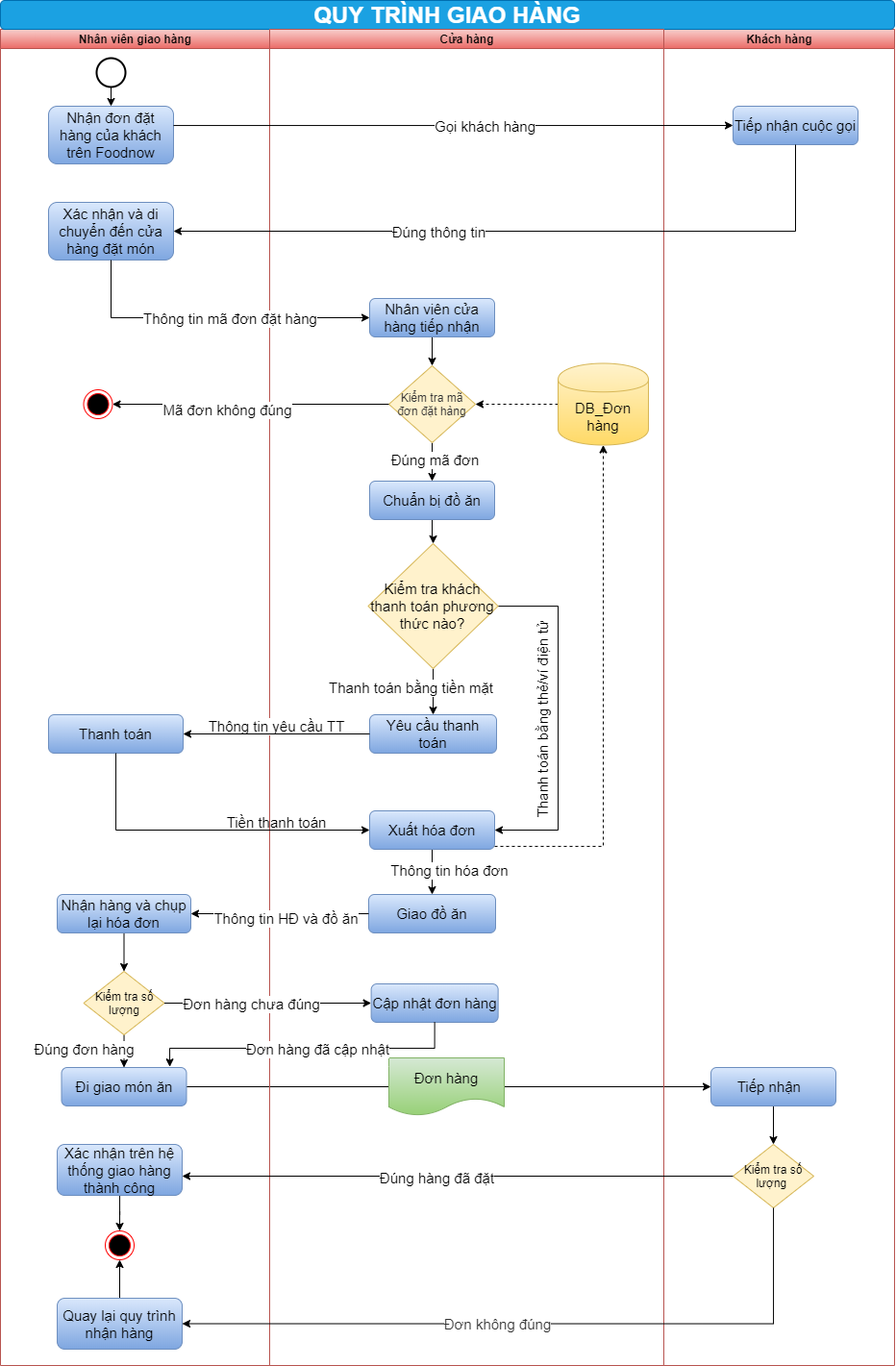
Khách hàng yêu cầu hủy đơn hàng sẽ vào đơn hàng đang đến, hệ thống sẽ xuất các danh sách đơn hàng mà khách hàng đặt đang chờ xác nhận từ phía cửa hàng. Khách hàng chọn đơn hàng cần hủy. Hệ thống sẽ kiểm tra đơn hàng, nếu đơn hàng đã được cửa hàng xác nhận thì đơn hàng không thể hủy sẽ thông báo cho khách hàng. Đơn hàng chưa được cửa hàng xác nhận thì khách hàng hủy thành công và hệ thống sẽ cập nhật đơn hàng cho khách hàng, chuyển đơn hàng hủy vào lịch sử.



*Hình 2.1.4. Quy trình khách hàng hủy đơn hàng*

* + - * 1. Quy trình giao hàng

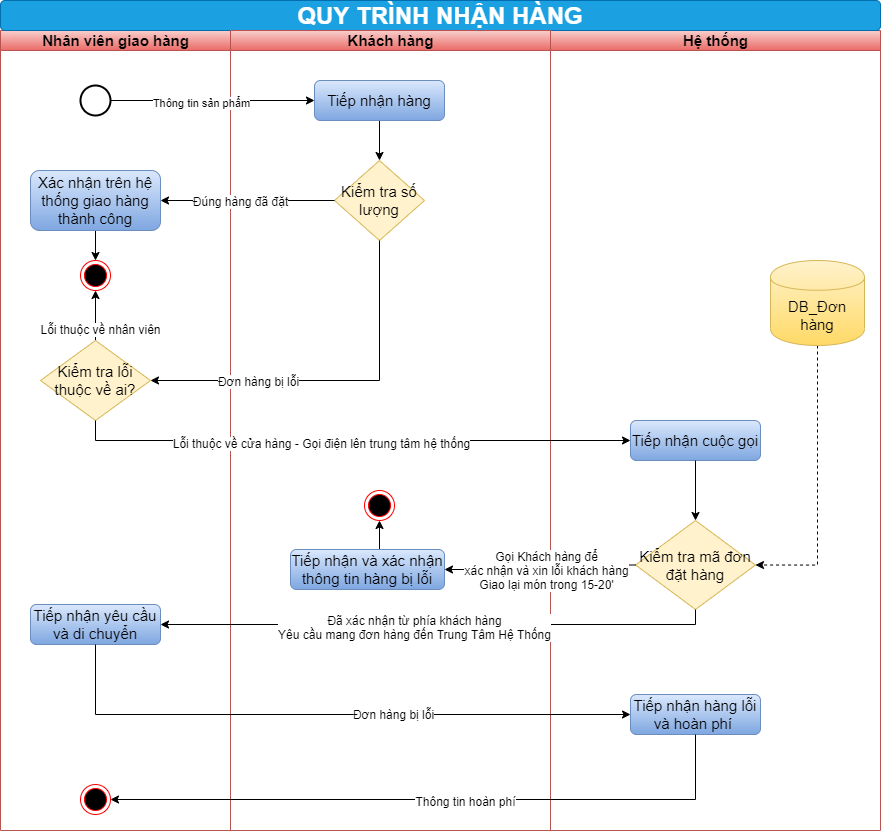
Nhân viên giao hàng(NVGH) nhận đơn của khách hàng trên hệ thống. NVGH sẽ gọi cho khách hàng để xác thực thông tin và di chuyển đến cửa hàng để lấy món. Nhân viên cửa hàng sẽ kiểm tra mã đơn. Đúng mã đơn cửa hàng sẽ chuẩn bị món. Trước khi giao món cho nhân viên giao hàng, cửa hàng sẽ kiểm tra khách hàng 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 đã thanh toán bằng ví, cửa hàng chỉ xuất hóa đơn và giao đồ ăn. NVGH nhận món và chụp lại hóa đơn, kiểm tra lại số lượng. Nếu đơn hàng đúng NVGH đi giao món ăn cho khách hàng, nếu kiểm tra số lượng và hóa đơn chưa đúng, NVGH yêu cầu cửa hàng cập nhật lại đơn 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. Trường hợp đơn đến khách hàng không đúng NVGH sẽ làm theo quy trình nhận hàng của khách hàng.



*Hình 2.1.5. Quy trình giao hàng*

* + - * 1. Qui trình nhận hàng từ phía khách hàng

Nhân viên giao hàng(NVGH) sẽ giao sản phẩm, khách hàng tiếp nhận và kiểm tra số lượng. Nếu đúng đơn đã đặt, NVGH sẽ xác nhận trên hệ thống là giao hàng thành công. Trường hợp khách hàng kiểm tra số lượng thấy đơn hàng bị lỗi, NVGH sẽ kiểm tra lỗi. Nếu lỗi thuộc về NVGH thì tự chịu trách nhiệm với đơn hàng. Nếu lỗi thuộc về cửa hàng, NVGH sẽ gọi lên trung tâm hệ thống. Hệ thống sẽ tiếp nhận cuộc gọi, kiểm tra lại mã đơn hàng của NVGH đang giao cùng những thông tin mà NGVH cung cấp. Hệ thống sẽ gọi cho khách hàng để xác nhận và khách hàng có thể đặt đơn khác. Hệ thống khi đã xác nhận từ phía khách hàng sẽ yêu cầu NVGH mang đơn hàng đến trung tâm, NVGH tiếp nhận và di chuyển. Hệ thống tiếp nhận đơn hàng lỗi và hoàn chi phí lại cho NVGH.



*Hình 2.1.6. Quy trình nhận hàng*

* + - * 1. Quy trình nhân viên giao hàng hủy đơn

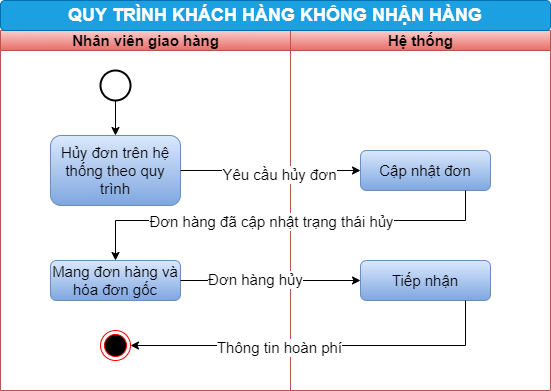
Nhân viên giao hàng(NVGH) chọn biểu tượng chat trên hệ thống, hệ thống sẽ xuất danh sách chat, NVGH chọn kênh hủy đơn, hệ thống yêu cầu nhập mã đơn cần hủy và NVGH sẽ nhập mã đơn theo cú pháp. Ngoài việc chọn biểu tượng chat, NVGH có thể chọn nút hủy đơn có trên hệ thống, hệ thống sẽ xuất danh sách lý do muốn hủy, NVGH chọn lý do sẽ được hệ thống xác nhận rồi kết thúc.



*Hình 2.1.7. Quy trình nhân viên giao hàng hủy đơn*

* + - * 1. Quy trình khách hàng không nhận đơn

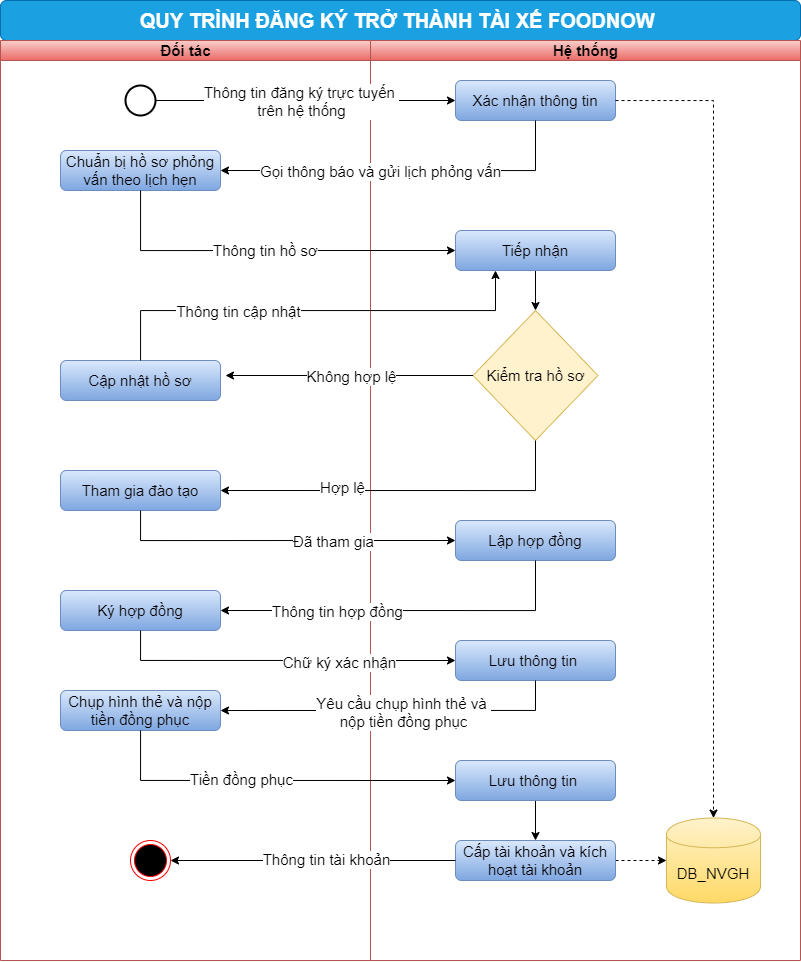
Khi khách hàng không nhận hàng, nhân viên giao hàng(NVGH) hủy đơn theo như đúng quy trình trên hệ thống, hệ thống sẽ cập nhật lại đơn và yêu cầu nhân viên giao hàng mang đơn lên hệ thống. Hệ thống tiếp nhận và hoàn phí lại cho NVGH.



*Hình 2.1.8. Quy trình khách hàng không nhận hàng*

* + - * 1. Quy trình đăng ký làm nhân viên giao hàng hệ thống

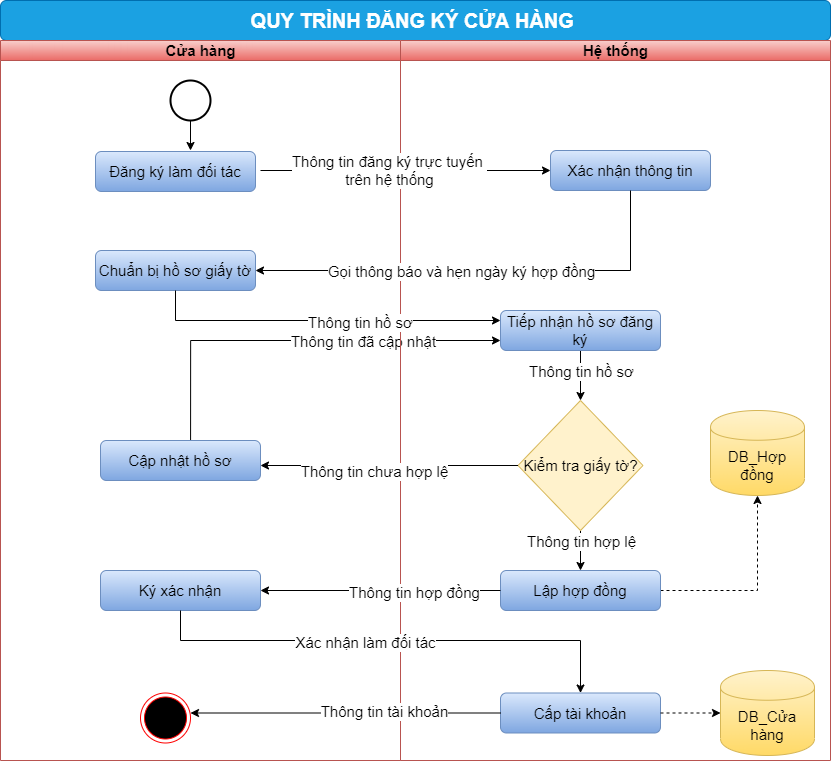
Đối tác muốn trở thành nhân viên giao hàng của hệ thống. Đối tác sẽ đăng ký trực tiếp trên trang hệ thống. Thông tin được gửi đến hệ thống, hệ thống sẽ gọi cho đối tác và hẹn ngày phỏng vấn. Khi đến phỏng vấn, đối tác mang theo hồ sơ để đăng ký. Hệ thống sẽ kiểm tra giấy tờ, nếu thông tin chưa hợp lệ sẽ yêu cầu đối tác cập nhật. Khi thông tin hợp lệ, hệ thống sẽ yêu cầu đối tác tham gia đào tạo. Sau khi hoàn thành việc đào tạo, hệ thống sẽ lập hợp đồng và đối tác ký xác nhận. Sau đó sẽ chụp hình thẻ và nộp tiền đồng phục. Hệ thống sẽ cấp cho đối tác tài khoản để đăng nhập hệ thống. Tài khoản sẽ được lưu vào dữ liệu nhân viên giao hàng.



*Hình 2.1.9. Quy trình đăng ký làm đối tác (tài xế) hệ thống*

* + - * 1. Quy trình đăng ký cho phía cửa hàng.

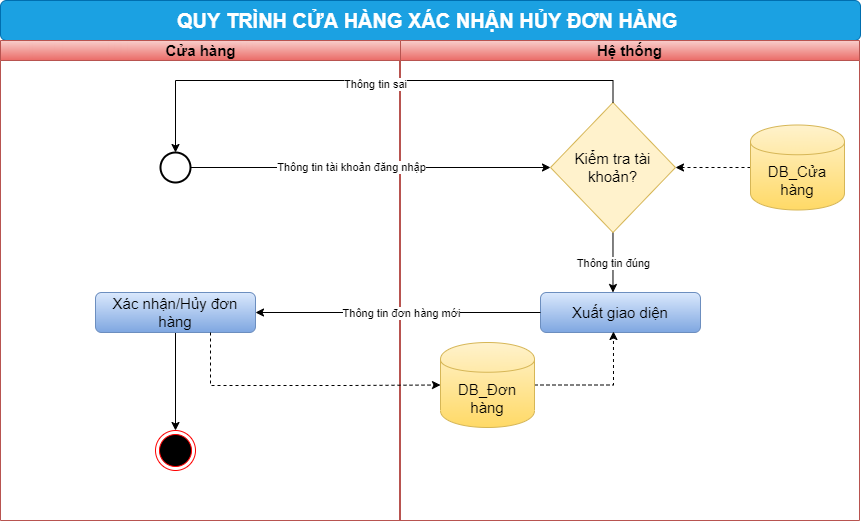
Cửa hàng sẽ gửi thông tin đăng ký đến hệ thống. Hệ thống xác nhận thông tin và gọi điện cho cửa hàng hẹn ngày ký hợp đồng là đối tác. Khi tới ký hợp đồng sẽ mang theo giấy tờ đăng ký để đảm bảo quyền lợi đôi bên. Hệ thống tiếp nhận hồ sơ và kiểm tra. Nếu chưa hợp lệ thì cửa hàng cần cập nhập lại. Nếu đã hợp lệ, hệ thống lập hợp đồng, sau đó sẽ cấp tài khoản để cửa hàng đăng nhập vào hệ thống. Tài khoản sẽ được lưu vào dữ liệu cửa hàng và hợp đồng sẽ được lưu vào dữ liệu hợp đồng.



*Hình 2.1.10. Quy trình đăng ký cửa hàng*

* + - * 1. Quy trình xác nhận, hủy đơn hàng phía cửa hàng.

Cửa hàng sẽ đăng nhập vào hệ thống với tài khoản mà hệ thống cung cấp. Hệ thống sẽ kiểm tra tài khoản nếu chưa đúng, cửa hàng cần đăng nhập lại. Nếu đúng tài khoản, hệ thống xuất giao diện hiển thị danh sách các đơn hàng mới. Cửa hàng sẽ chọn xác nhận hoặc hủy đơn hàng. Danh sách đơn hàng sẽ được lưu vào dữ liệu đơn hàng.

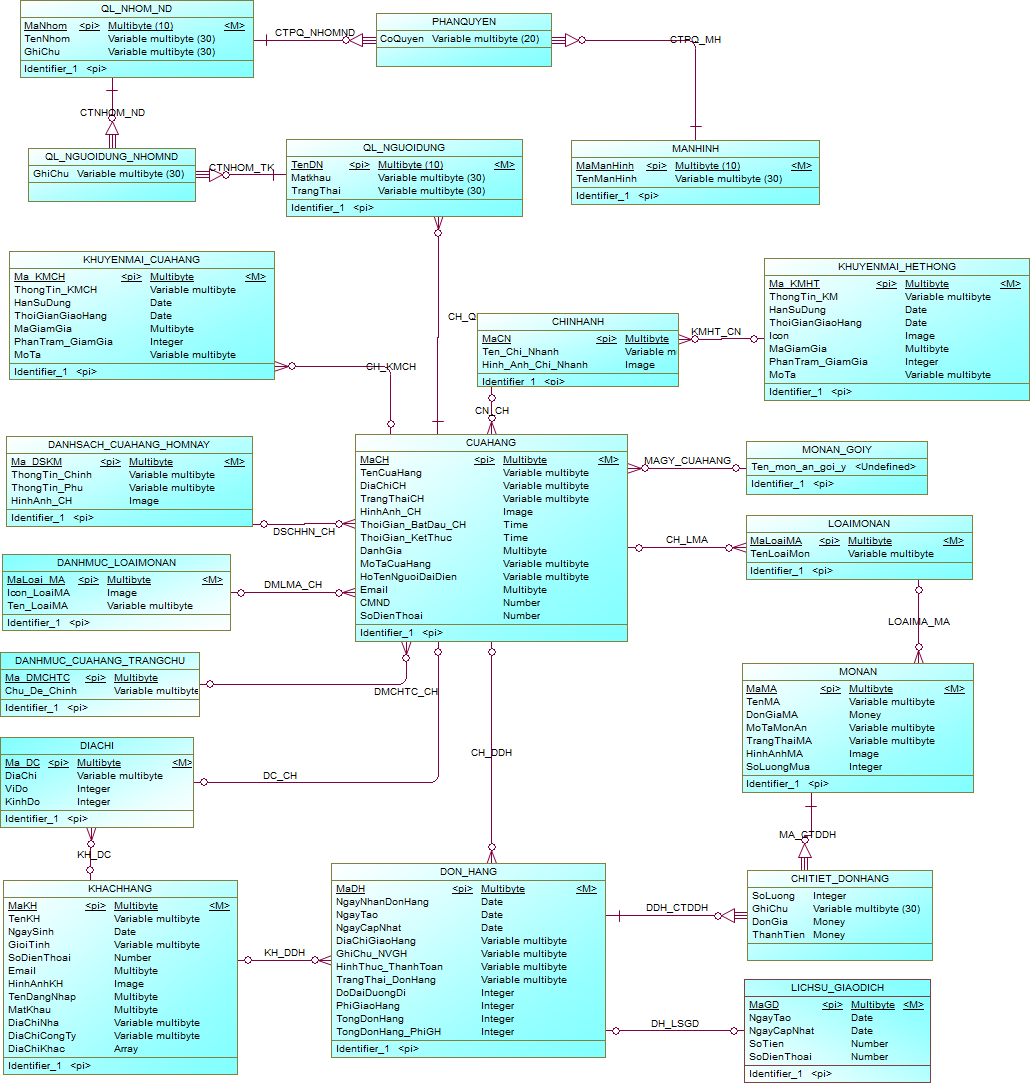


*Hình 2.1.11. Quy trình xác nhận hủy đơn hàng*

**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

1. **PHÂN TÍCH HỆ THỐNG**
   1. **MÔ HÌNH ERD**

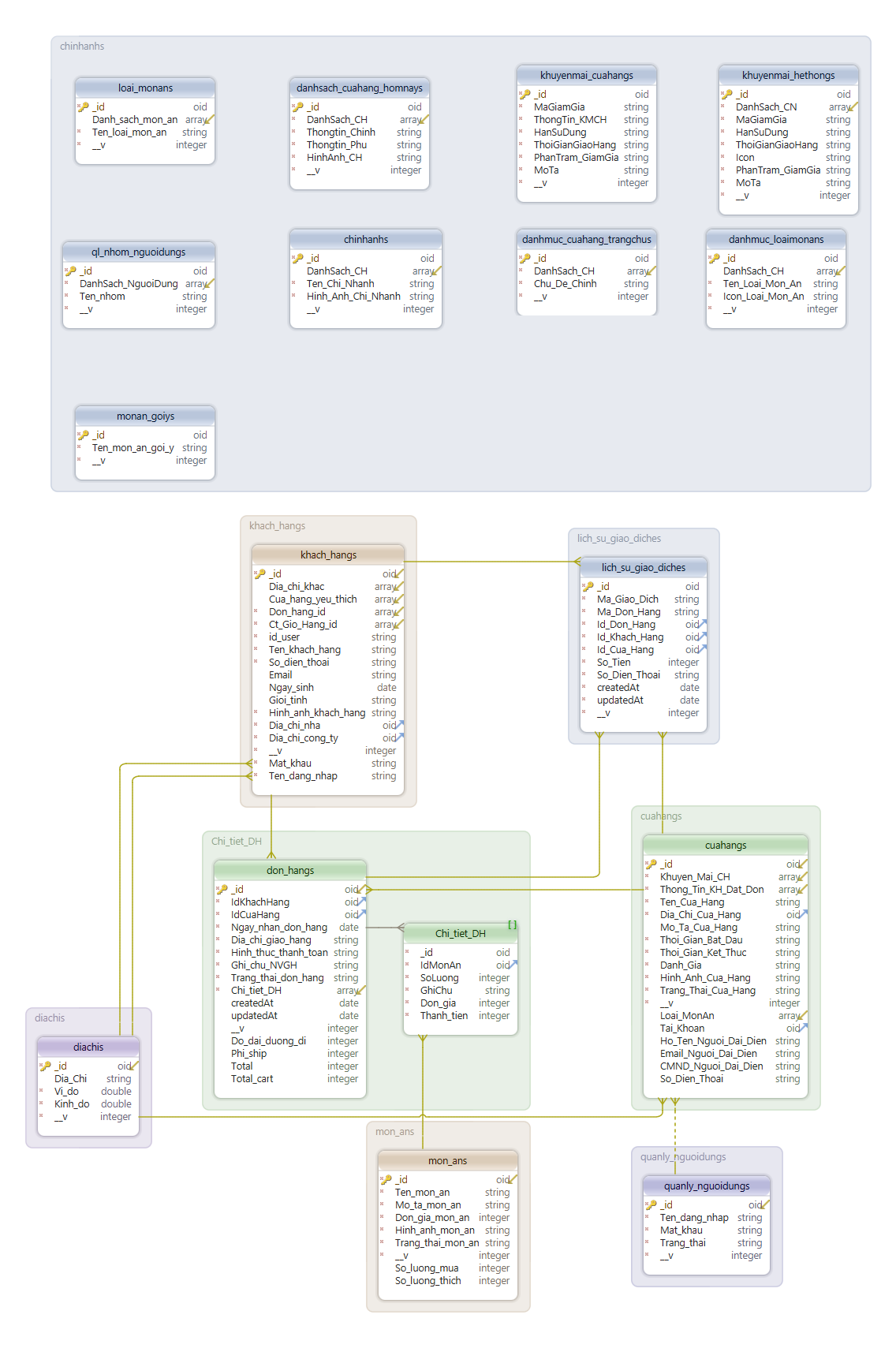
Từ việc khảo sát hệ thống và triển khai các quy trình nghiệp vụ, nhóm tiến hành phân tích và thiết kế hệ thống tập trung cho phần khách hàng và cửa hàng.



*Hình 2.2.1. Mô hình ERD*

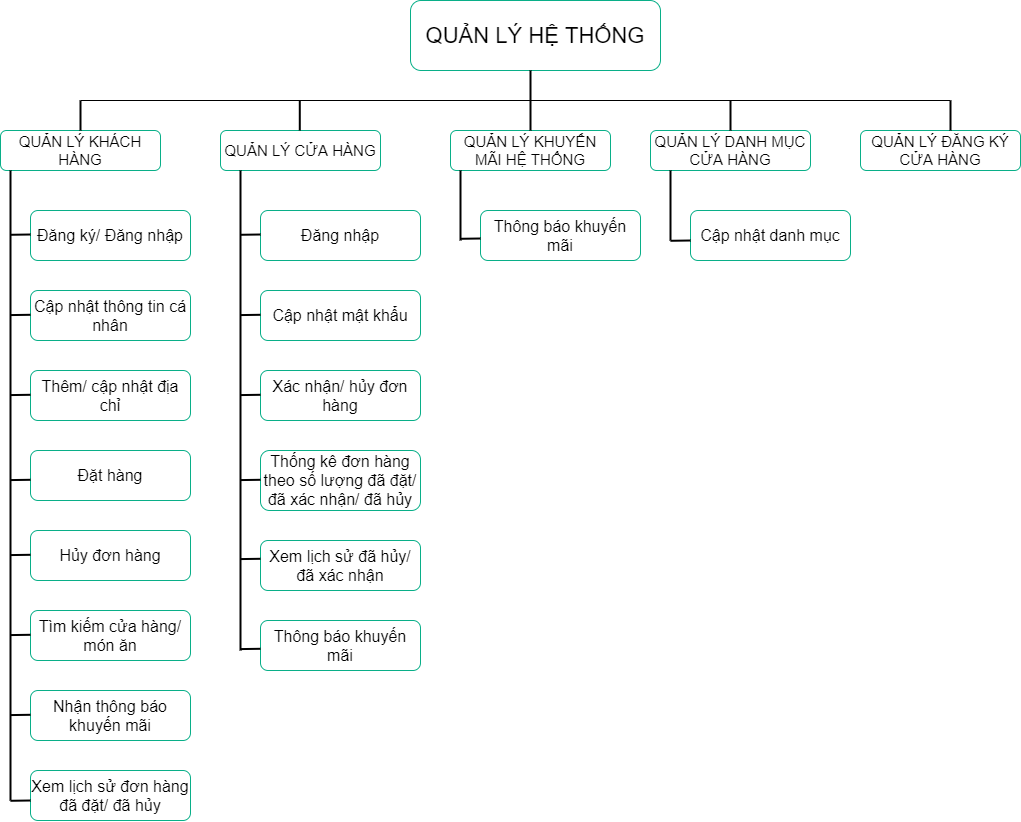
1. **THIẾT KẾ HỆ THỐNG**
   1. **THIẾT KẾ CƠ SỞ DỮ LIỆU**

Từ việc phân tích bằng mô hình ERD, nhóm đã thiết kế cơ sở dữ liệu các bảng bằng NOSQL để phù hợp với đề tài nghiên cứu.



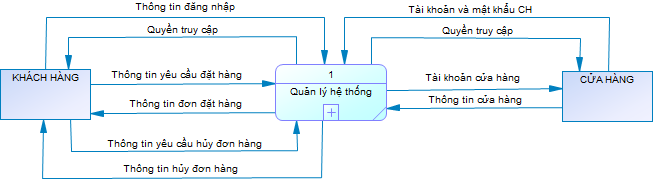
*Hình 2.2.2. Mô hình cơ sỡ dữ liệu NoSQL*

* 1. **SƠ ĐỒ PHÂN CẤP CHỨC NĂNG**



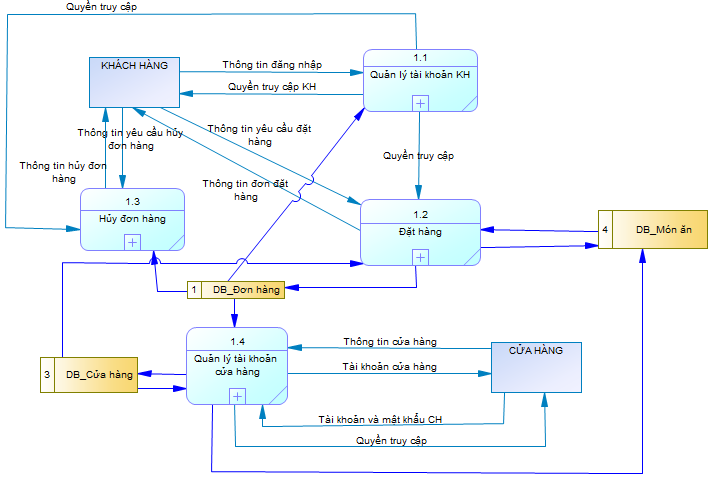
*Hình 2.2.3. Mô hình phân cấp chức năng*

* 1. **MÔ HÌNH DFD**
     1. Mô hình mức ngữ cảnh



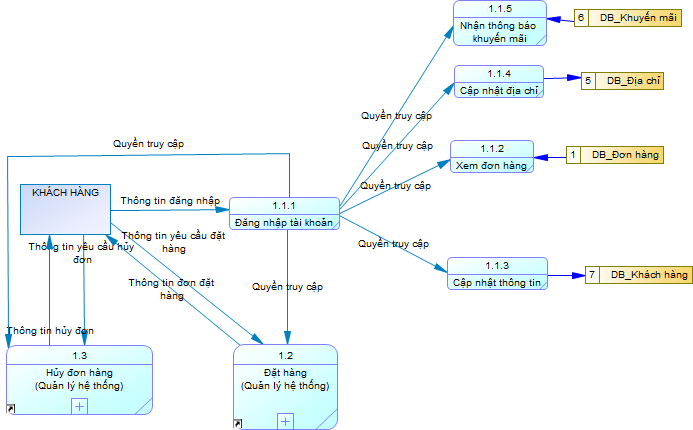
*Hình 2.2.4. Mô hình mức ngữ cảnh*

* + 1. Mô hình mức đỉnh



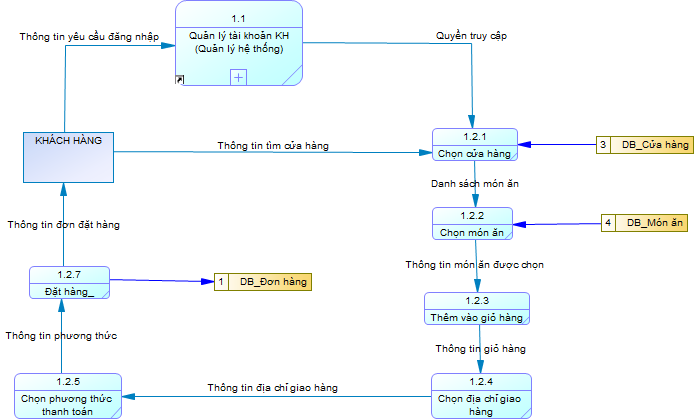
*Hình 2.2.5. Mô hình mức đỉnh*

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



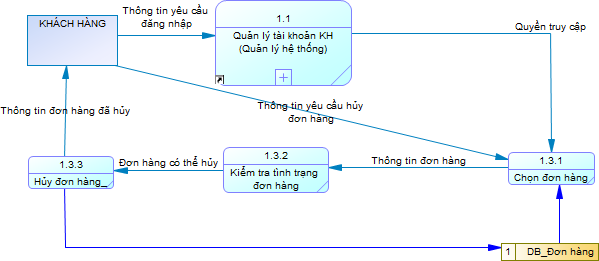
*Hình 2.2.6. Mô hình quản lý khách hàng*

* + - * 1. Mô hình quản lý đặt hàng



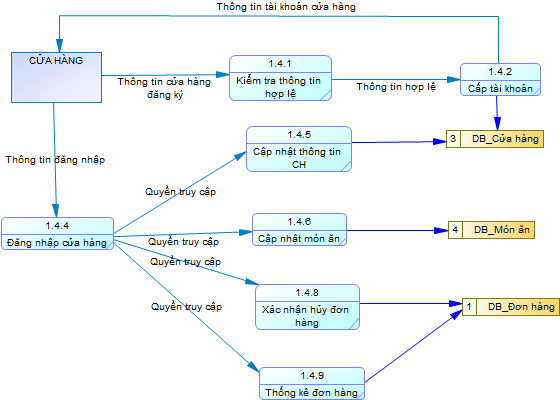
*Hình 2.2.7. Mô hình quản lý đặt hàng*

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



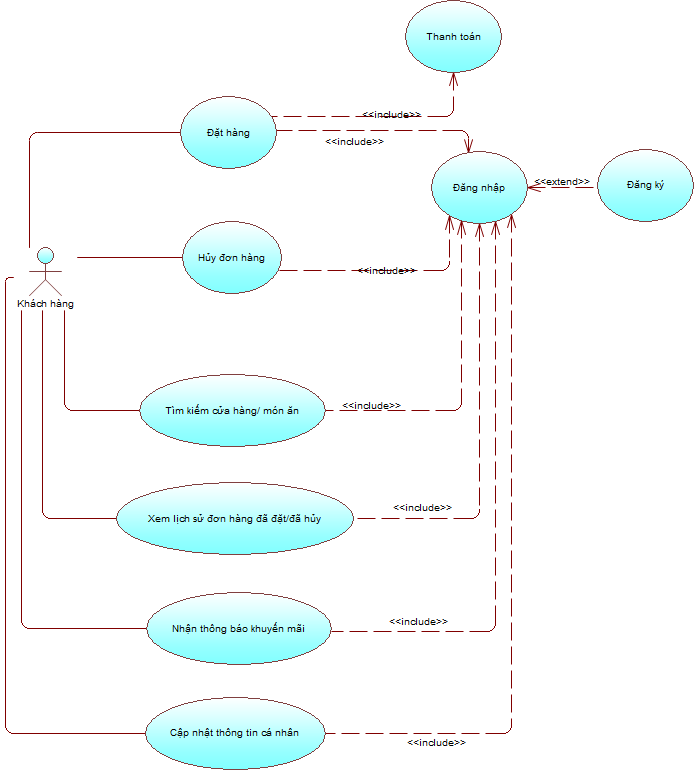
*Hình 2.2.8. Mô hình quản hủy đơn hàng*

* + - * 1. Mô hình quản lý cửa hàng



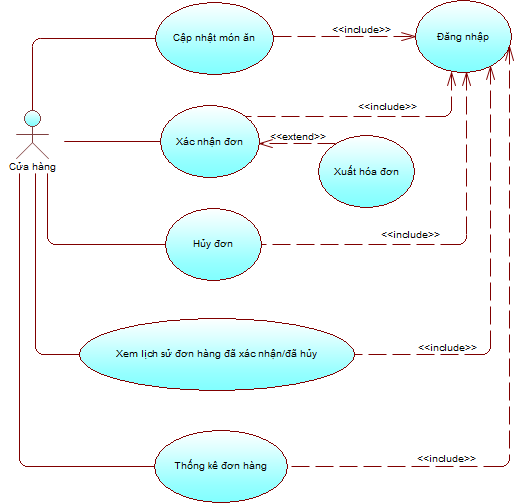
*Hình 2.2.9. Mô hình quản lý cửa hàng*

* 1. **MÔ HÌNH USE CASE**
     1. **USE CASE KHÁCH HÀNG**



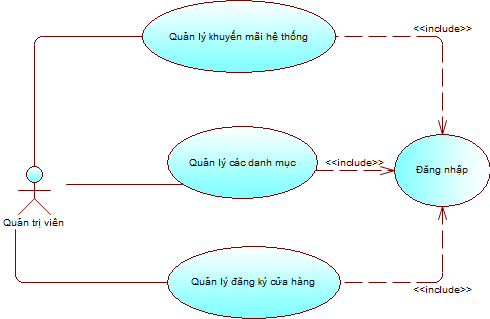
*Hình 2.2.10. Mô hình use case khách hàng*

* + 1. **USE CASE CỬA HÀNG**



*Hình 2.2.11. Mô hình use case cửa hàng*

* + 1. **USE CASE QUẢN TRỊ VIÊN**



*Hình 2.2.12. Mô hình use case quản trị viên*

* 1. **XÂY DỰNG API**

Từ việc phân tích các mô hình chức năng, nhóm thiết kế API các tính năng cho hệ thống FoodNow theo bảng sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Phương thức** | **Đường dẫn** | **Mô tả** |
| 1 | GET | http:// 192.168.137.1:3000/getDanhmucloaimonan | Lấy danh sách các mục loại món ăn |
| 2 | POST | [http://192.168.137.1:3000/getDanhSachCuaHang\_LoaiMonAn](http://localhost:3000/getDanhSachCuaHang_LoaiMonAn) | Lấy danh sách các cửa hàng thuộc danh mục loại món ăn |
| 3 | POST | http://192.168.137.1:3000/themXoaCuaHang\_DanhMuc\_LoaiMonAn | Thêm xóa cửa hàng khỏi danh mục loại món ăn |
| 4 | GET | http://192.168.137.1:3000/Danhsachcuahanghomnayhienthi | Lấy danh sách các nhóm trong phần đầu trang chủ |
| 5 | POST | http://192.168.137.1:3000/Danhsachcuahanghomnay\_hienthicuahang | Lấy danh sách các cửa hàng nằm trong từng nhóm thuộc phần đầu trang chủ |
| 6 | POST | http://192.168.137.1:3000/addDanhsachcuahanghomnay | Thêm một nhóm trong phần đầu trang chủ |
| 7 | DELETE | http://192.168.137.1:3000/deleteDanhmuccuahanghomnay | Xóa một nhóm trong phần đầu trang chủ |
| 8 | POST | http://192.168.137.1:3000/getDanhSachCuaDanhMucHomNay | Lấy danh sách các cửa hàng nằm trong từng nhóm thuộc phần đầu trang chủ |
| 9 | POST | [http://192.168.137.1:3000/themXoaCuaHang\_DanhMuc\_Cuahanghomnay](http://localhost:3000/themXoaCuaHang_DanhMuc_Cuahanghomnay) | Thêm hoặc xóa các cửa hàng trong từng nhóm thuộc phần đầu trang chủ |
| 10 | GET | http://192.168.137.1:3000/Hienthitendanhmuccuahangtrangchu | Lấy danh sách các nhóm nằm trong phần thân trang chủ |
| 11 | POST | http://192.168.137.1:3000/Hienthicuahangtrongdanhmuctrangchu | Lấy danh sách các cửa hàng nằm trong từng nhóm thuộc phần thân trang chủ |
| 12 | DELETE | http://192.168.137.1:3000/deleteDanhmuccuahangtrangchu | Xóa một nhóm trong phần thân trang chủ |
| 13 | POST | http://192.168.137.1:3000/getDanhSachCuaDanhMuc | Lấy danh sách các cửa hàng nằm trong từng nhóm thuộc phần thân trang chủ |
| 14 | POST | http://192.168.137.1:3000/themXoaCuaHang\_DanhMuc | Thêm hoặc xóa các cửa hàng trong từng nhóm thuộc phần thân trang chủ |
| 15 | GET | http://192.168.137.1:3000/Danhsachkhuyenmaihethong | Lấy danh sách các khuyến mãi hệ thống |
| 16 | POST | http://192.168.137.1:3000/Hienthichinhanh\_thuockhuyenmai | Lấy danh sách các chi nhánh thuộc khuyến mãi hệ thống áp dụng |
| 17 | POST | http://192.168.137.1:3000/addKhuyenmaihethong | Thêm khuyến mãi hệ thống mới |
| 18 | DELETE | http://192.168.137.1:3000/deleteKhuyenmaihethong | Xóa khuyến mãi hệ thống mới |
| 19 | POST | http://192.168.137.1:3000/getDanhSachChiNhanhKMHT | Lấy danh sách các chi nhánh |
| 20 | POST | http://192.168.137.1:3000/themXoaChiNhanh\_DanhMuc\_KhuyenMaiHe  Thong | Thêm hoặc xóa chi nhánh khỏi khuyến mãi hệ thống |
| 21 | GET | http://192.168.137.1:3000/getKMHeThongCuaCuaHang | Lấy danh sách khuyến mãi hệ thống áp dụng cho cửa hàng |
| 22 | POST | http://192.168.137.1:3000/Danhsachmonan\_cuahang | Lấy danh sách các món ăn thuộc cửa hàng |
| 23 | POST | http://192.168.137.1:3000/them\_loaimonan | Thêm loại món ăn |
| 24 | POST | http://192.168.137.1:3000/them\_monan | Thêm món ăn |
| 25 | POST | [http://192.168.137.1:3000/getTaikhoancuahang](http://localhost:3000/getTaikhoancuahang) | Lấy thông tin tài khoản cửa hàng |
| 26 | POST | [http://192.168.137.1:3000/KhuyenMaiCuaHang](http://localhost:3000/KhuyenMaiCuaHang) | Lấy danh sách các khuyến mãi cửa hàng |
| 27 | POST | [http://192.168.137.1:3000/addKhuyenmaicuahang](http://localhost:3000/addKhuyenmaicuahang) | Thêm khuyến mãi cửa hàng mới |
| 28 | DELETE | [http://192.168.137.1:3000/deleteKhuyenmaicuahang](http://localhost:3000/deleteKhuyenmaicuahang) | Xóa khuyến mãi cửa hàng |
| 29 | POST | [http://192.168.137.1:3000/addCuaHang](http://localhost:3000/addCuaHang) | Thêm cửa hàng mới |
| 30 | POST | [http://192.168.137.1:3000/cuahangs\_chinhanh](http://localhost:3000/cuahangs_chinhanh) | Lấy danh sách cửa hàng thuộc một chi nhánh |
| 31 | POST | [http://192.168.137.1:3000/addChiNhanh](http://localhost:3000/addChiNhanh) | Thêm mới chi nhánh |
| 32 | GET | [http://192.168.137.1:3000/chinhanh](http://localhost:3000/chinhanh) | Lấy danh sách tất cả các chi nhánh |
| 33 | DELETE | [http://192.168.137.1:3000/deleteChiNhanh](http://localhost:3000/deleteChiNhanh) | Xóa chi nhánh |
| 34 | POST | [http://192.168.137.1:3000/addKhachHang](http://localhost:3000/addKhachHang) | Thêm mới khách hàng |
| 35 | POST | [http://192.168.137.1:3000/DangNhap](http://localhost:3000/DangNhap) | Khách hàng đăng nhập |
| 36 | POST | [http://192.168.137.1:3000/getGioHang](http://localhost:3000/getGioHang) | Chi tiết giỏ hàng của 1 tài khoản |
| 37 | POST | [http://192.168.137.1:3000/CTGioHang](http://localhost:3000/CTGioHang) | Hiển thị chi tiết giỏ hàng |
| 38 | POST | [http://192.168.137.1:3000/getDonHang](http://localhost:3000/getDonHang) | Đơn hàng của khách hàng |
| 39 | POST | [http://192.168.137.1:3000/DonHang](http://localhost:3000/DonHang) | Thông tin đơn hàng |
| 40 | POST | [http://192.168.137.1:3000/CTDonHang](http://localhost:3000/CTDonHang) | Hiển thị chi tiết đơn hàng |
| 41 | POST | [http://192.168.137.1:3000/HienThiCuaHang\_KhachHangYeuThich](http://localhost:3000/HienThiCuaHang_KhachHangYeuThich) | Hiển thị danh sách các cửa hàng khách hàng yêu thích |
| 42 | POST | [http://192.168.137.1:3000/Hienthithongtin\_taikhoankhachhang](http://localhost:3000/Hienthithongtin_taikhoankhachhang) | Hiển thị thông tin khách hàng |
| 43 | POST | [http://192.168.137.1:3000/Themcuahangyeuthich\_khachhang](http://localhost:3000/Themcuahangyeuthich_khachhang) | Khách hàng thêm cửa hàng yêu thích vào danh sách yêu thích của mình |
| 44 | POST | [http://192.168.137.1:3000/Dangnhapadmin](http://localhost:3000/Dangnhapadmin) | Quản trị viên đăng nhập |

|  |  |  |  |
| --- | --- | --- | --- |
| 45 | POST | [http://192.168.137.1:3000/fragment\_home](http://localhost:3000/fragment_home) | Lấy data cho fragment home gồm các khối thông tin sau  + thông tin về khuyến mãi của hệ thống + Gợi ý các cửa hàng - title "Hôm nay ăn gì ?" + Danh mục - detail danh mục cửa hàng của hệ thống FoodNow |
| 46 | POST | [http://192.168.137.1:3000/chonSanPham](http://localhost:3000/chonSanPham) | Cập nhật giỏ hàng |
| 47 | POST | [http://192.168.137.1:3000/datHang](http://localhost:3000/datHang) | Đặt hàng |
| 48 | POST | [http://192.168.137.1:3000/Hienthicuahangduocapdungkhuyenmai](http://localhost:3000/Hienthicuahangduocapdungkhuyenmai) | Hiển thị các cửa hàng được KMHT áp dụng |
| 49 | POST | [http://192.168.137.1:3000/chitietDonHang](http://localhost:3000/chitietDonHang) | Lấy thông tin chi tiết đơn hàng |
| 50 | POST | [http://192.168.137.1:3000/danhSachDongHang](http://localhost:3000/danhSachDongHang) | Lấy danh sách đơn hàng đươc khách hàng đặt |
| 51 | POST | [http://192.168.137.1:3000/donHangMoi](http://localhost:3000/donHangMoi) | Lấy danh sách đơn hàng của cửa hàng |
| 52 | POST | [http://192.168.137.1:3000/getTaikhoancuahang](http://localhost:3000/getTaikhoancuahang) | Lấy thông tin tài khoản cửa hàng |
| 53 | POST | [http://192.168.137.1:3000/capnhatmatkhau\_cuahang](http://localhost:3000/capnhatmatkhau_cuahang) | Cập nhật mật khẩu cửa hàng |
| 54 | POST | [http://192.168.137.1:3000/capNhatTrangThaiDonHang](http://localhost:3000/capNhatTrangThaiDonHang) | Cập nhật trạng thái đơn hàng của khách hàng |
| 55 | POST | [http://192.168.137.1:3000/cuaHangCapNhatTrangThaiDonHang](http://localhost:3000/cuaHangCapNhatTrangThaiDonHang) | Cửa hàng xác nhận hoặc hủy đơn |
| 56 | POST | [http://192.168.137.1:3000/Danhsachmonan\_cuahang](http://localhost:3000/Danhsachmonan_cuahang) | Hiển thị danh sách món ăn: Loại món ăn và món ăn của cửa hàng trên app |
| 57 | POST | [http://192.168.137.1:3000/Hienthimonan\_chonloaimonan](http://localhost:3000/Hienthimonan_chonloaimonan) | Hiển thị danh sách món ăn khi chọn loại món ăn |
| 58 | POST | [http://192.168.137.1:3000/getKMHeThongCuaCuaHang](http://localhost:3000/getKMHeThongCuaCuaHang) | Lấy danh sách khuyến mãi mà hệ thống áp dụng cho cửa hàng |
| 59 | POST | [http://192.168.137.1:3000/kichHoatCuaHang](http://localhost:3000/kichHoatCuaHang) | Kích hoạt trạng thái cửa hàng |
| 60 | POST | [http://192.168.137.1:3000/danhSachDonHang\_KhachHang](http://localhost:3000/danhSachDonHang_KhachHang) | Lấy danh sách đơn hàng của khách hàng:  đơn hàng tạm, đơn hàng đã đặt đang chờ xác nhận, đơn hàng cửa hàng xác nhận |
| 61 | POST | [http://192.168.137.1:3000/themDiaChi](http://localhost:3000/themDiaChi) | Thêm địa chỉ cho khách hàng |
| 62 | POST | [http://192.168.137.1:3000/getDanhSachCuaHangKMHT](http://localhost:3000/getDanhSachCuaHangKMHT) | Lấy danh sách tất cả cửa hàng được khuyến mãi hệ thống áp dụng |
| 63 | POST | [http://192.168.137.1:3000/Hienthiketqua\_timkiemmonan](http://localhost:3000/Hienthiketqua_timkiemmonan) | Hiển thị kết quả tìm kiếm gồm danh sách cửa hàng và món ăn cửa hàng đó |
| 64 | POST | [http://192.168.137.1:3000/cuaHangThongKeDonHang](http://localhost:3000/cuaHangThongKeDonHang) | Lấy thông tin thống kê của cửa hàng và gửi về thông tin thống kê các đơn hàng của cửa hàng |

*Bảng 2.2.13. Thiết kế API*

**CHƯƠNG 3: THIẾT KẾ GIAO DIỆN**

1. **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. Từ hai tài khoản khi đăng nhập, hệ thống sẽ lưu lại thông tin và cung cấp một tên đăng nhập và mật khẩu.

|  |  |
| --- | --- |
|  |  |
| *Hình 2.3.1. Giao diện mở đầu hệ thống* | Sau khi kết thúc màn hình mở đầu hệ thống sẽ chuyển đến màn hình đăng ký đăng nhập. |
| *Hình 2.3.2. Giao diện đăng ký, đăng nhập* | Khách hàng đăng nhập để hệ thống tạo tài khoản cho lần đầu vào hệ thống. Đăng nhập bằng Google hoặc Facebook mà hệ thống cung cấp.  *Nhóm đã ứng dụng Google API, Facebook API để lấy thông tin tài khoản khi khách hàng đăng ký/ đăng nhập.*  *Ứng dụng Json Web Token để mã hóa tài khoản khi khách hàng đăng nhập bằng tên đăng nhập được cấp trong thông tin tài khoản khách hàng.* |

* 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.

|  |  |
| --- | --- |
| *Hình 2.3.3. Giao diện tài khoản đăng nhập* | Khi khách hàng đăng xuất và muốn đăng nhập lại sẽ vào tài khoản để đăng nhập. |

|  |  |
| --- | --- |
| *Hình 2.3.4. Giao diện tài khoản đăng xuất* | Khi khách hàng đăng nhập thành công sẽ thấy thông tin vừa đăng nhập. |
| *Hình 2.3.5. Giao diện thông tin chi tiết khách hàng* | Khách hàng có thể xem chi tiết thông tin của mình và có thể cập nhật thông tin.  *Nhóm ứng dụng Json Web Token để mã hóa tài khoản cung cấp cho khách hàng tên đăng nhập và mật khẩu.* |
| *Hình 2.3.6. Giao diện thêm địa chỉ mới* | Khách hàng có thể thêm địa chỉ mới hoặc thay đổi địa chỉ nhà, địa chỉ công ty.  *Tích hợp Google Map để khách hàng có thể thay đổi địa chỉ .* |

* 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.

|  |  |
| --- | --- |
| *Hình 2.3.7. Giao diện trang chủ phần đầu* | Vị trí hiện tại sẽ được cập nhật khi khách hàng bật vị trí.  *Nhóm tích hợp Google Map để lấy vị trí hiện tại 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. |

|  |  |
| --- | --- |
| *Hình 2.3.8. Giao diện trang chủ phần giữa hôm nay* | Hôm nay ăn gì là gồm các nhóm chứa các cửa hàng  FoodNow đi chợ hiển thị danh sách các cửa hàng |
| *Hình 2.3.9. Giao diện trang chủ phần giữa giải khát ở nhà* | Tương tự Giải khát ở nhà và Ăn mì bún, phở sẽ hiển thị các cửa hàng. |
| *Hình 2.3.10. Giao diện tìm kiếm món ăn* | Khi chọn ô tìm kiếm sẽ hiển thị các tìm kiếm gợi ý hoặc khách hàng tự nhập.  *Khi kết quả hiển thị cho tìm kiếm này là danh sách các cửa hàng có địa chỉ gần khách hàng, được sắp xếp theo cấp độ tăng dần về khoảng cách. Nhóm tích hợp Google Map.* |
| *Hình 2.3.11. Giao diện phần cuối trang chủ* | Quanh đây có gì ngon sẽ hiển thị các cửa hàng có vị trí gần khách hàng nhất.  Nếu trường hợp khách hàng không bật vị trí. Các cửa hàng sẽ hiển thị theo địa chỉ của khách hàng.  *Nhóm tích hợp GoogleMap vào việc tính khoảng cách để hiển thị cửa hàng gần địa chỉ 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.

|  |  |
| --- | --- |
| *Hình 2.3.12. Giao diện danh sách cửa hàng tìm kiếm* | 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.  *Nhóm tích hợp GoogleMap vào việc tính khoảng cách để hiển thị cửa hàng gần khách hàng nhất, sắp xếp theo cấp độ tăng dần khoảng cách.* |

|  |  |
| --- | --- |
| *Hình 2.3.13. Giao diện danh sách cửa hàng gần tôi* | Khi khách hàng chọn tab Gần tôi sẽ hiển thị danh sách các cửa hàng gần vị trí của khách hàng. |
| *Hình 2.3.14. Giao diện danh sách cửa hàng đặt nhiều* | 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. |
| *Hình 2.3.15. Giao diện thông tin cửa hàng cùng món ăn* | Sau khi chọn cửa hàng mong muốn, thông tin khuyến mãi cùng các loại món ăn và món ăn cũng được hiển thị. |
| *Hình 2.3.16. Giao diện thông tin khuyến mãi cửa hàng* | Khách hàng có thể xem chi tiết thông báo khuyến mãi. |
| *Hình 2.3.17. Giao diện danh sách loại món ăn và món ăn của cửa hàng* | 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.  Từng loại món ăn sẽ hiển thị trên tab. |
| *Hình 2.3.18. Giao diện địa chỉ cửa hàng* | Khách hàng có thể xem được địa chỉ cửa hàng và hệ thống sẽ chuyển sang Google Map nếu khách hàng chọn vào chỉ đường.  *Nhóm tích hợp Google Map hiển thị địa chỉ cửa hàng.* |
| *Hình 2.3.19. Giao diện món ăn trong giỏ hàng* | Tại món ăn mà khách hàng lựa chọn có thể thêm ghi chú cho món ăn.  Khi chọn giao hàng thì hệ thống sẽ tiến hàng lập đơn đặt hàng. |
| *Hình 2.3.20. Giao diện ghi chú món ăn* | Phần ghi chú cho món ăn sẽ giới hạn là 500 ký tự. |

* 1. **ĐƠN HÀNG**

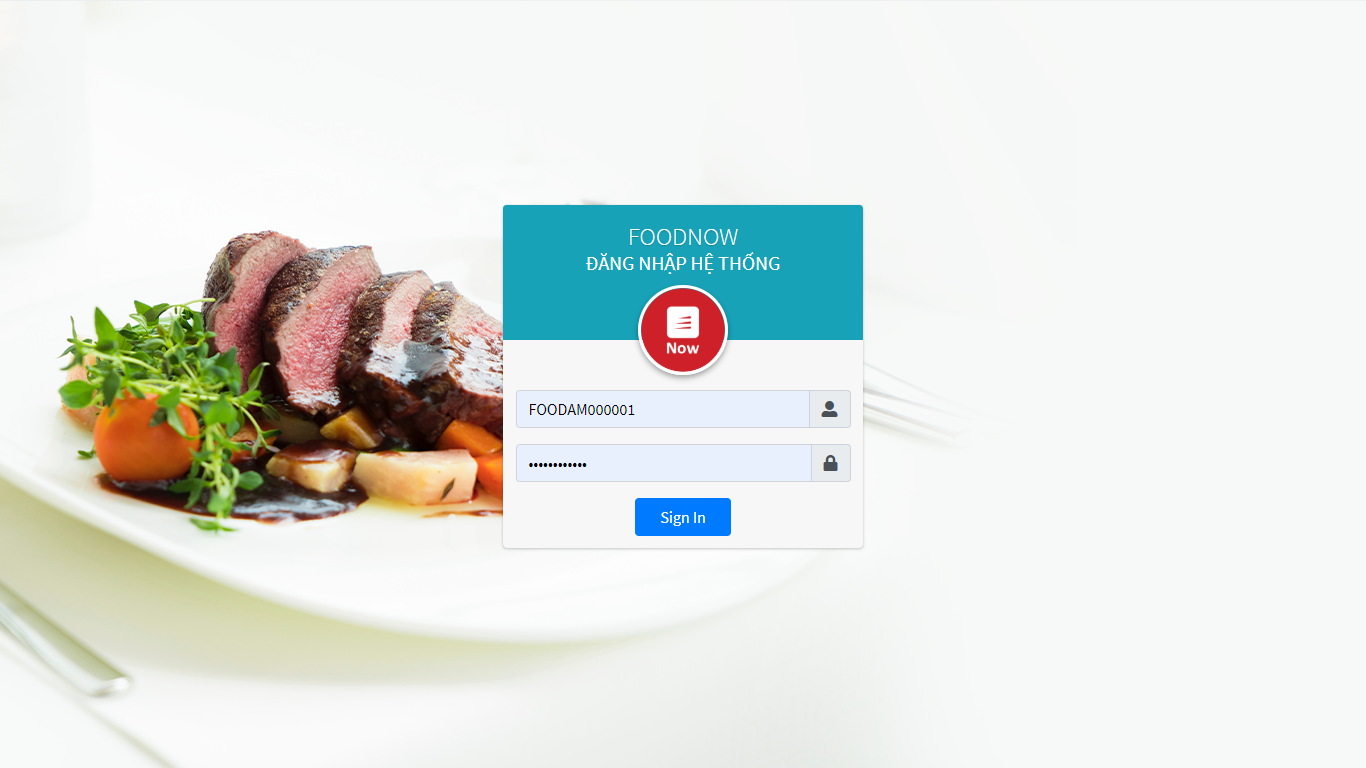
Sau khi khách hàng chọn giao hàng thì hệ thống sẽ tiến hành lập đơn đặt hàng. Khách hàng sẽ kiểm tra lại thông tin rồi bấm chọn đặt hàng.

|  |  |
| --- | --- |
| *Hình 2.3.21. Giao diện đơn 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. |

|  |  |
| --- | --- |
| *Hình 2.3.22. Giao diện chọn thời gian* | Tại đây khách hàng có thể thay đổi khung giờ giao hàng. |
| *Hình 2.3.23. Giao diện thay đổi địa chỉ* | Khách hàng có thể chọn lại địa chỉ trước khi đặt hàng.  *Nhóm tích hợp Google Map để khách hàng có thể thay đổi địa chỉ.* |
| *Hình 2.3.24. Giao diện chọn đặt đơn khi đã kiểm tra thông tin đầy đủ* | Sau khi kiểm tra đầy đủ thông tin khách hàng sẽ chọn phương thức thanh toán rồi bấm đặt hàng.  *Nhóm sử dụng thuật toán RSA và SHA256 vào thanh toán trực tuyến bằng ví Momo để mã hóa.*  *.* |
| *Hình 2.3.25. Giao diện chọn đặt đơn khi đã kiểm tra thông tin đầy đủ* | Khi khách hàng đặt hàng, bên cửa hàng sẽ xác nhận nếu như còn món. Cửa hàng sẽ hủy đơn hàng nếu như hết món ăn.  *Nhóm ứng dụng Firebase Cloud Messaging để thông báo đến ứng dụng khách hàng.* |
| *Hình 2.3.26. Giao diện khách hàng thanh toán bằng ví Momo thành công* | Chi tiết giao dịch thành công khi thanh toán bằng ví Momo |
| *Hình 2.3.27. Giao diện hiển thị thông báo thanh toán thành công trên ví Momo* | Thông báo giao dịch thực hiện thanh toán bên ví Momo thành công. |
| *Hình 2.3.28. Giao diện hiển thị thông báo thanh toán Momo thành công* | Thông báo đến khách hàng thanh toán thành công bằng ví Momo. |

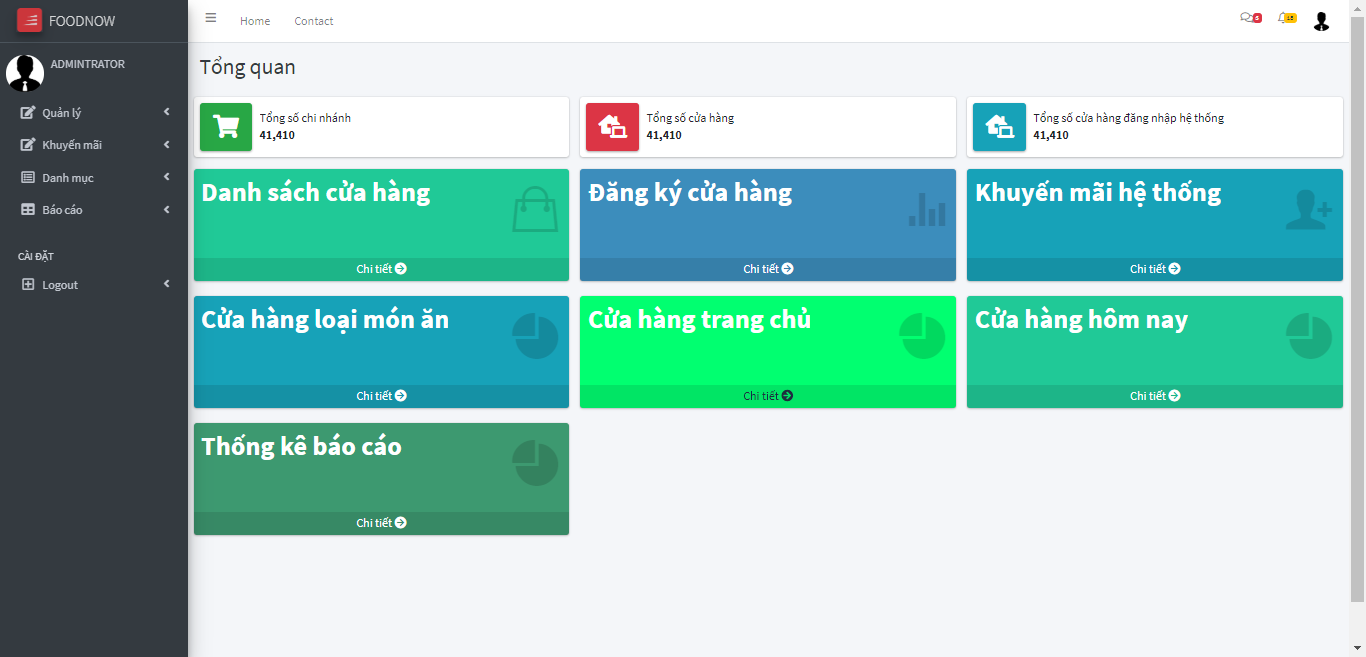
1. **GIAO DIỆN WEB ADMIN**
   1. **ĐĂNG NHẬP**

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



*Hình 2.3.29. Giao diện quản trị đăng nhập*

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



*Hình 2.3.30. Giao diện tổng quan*

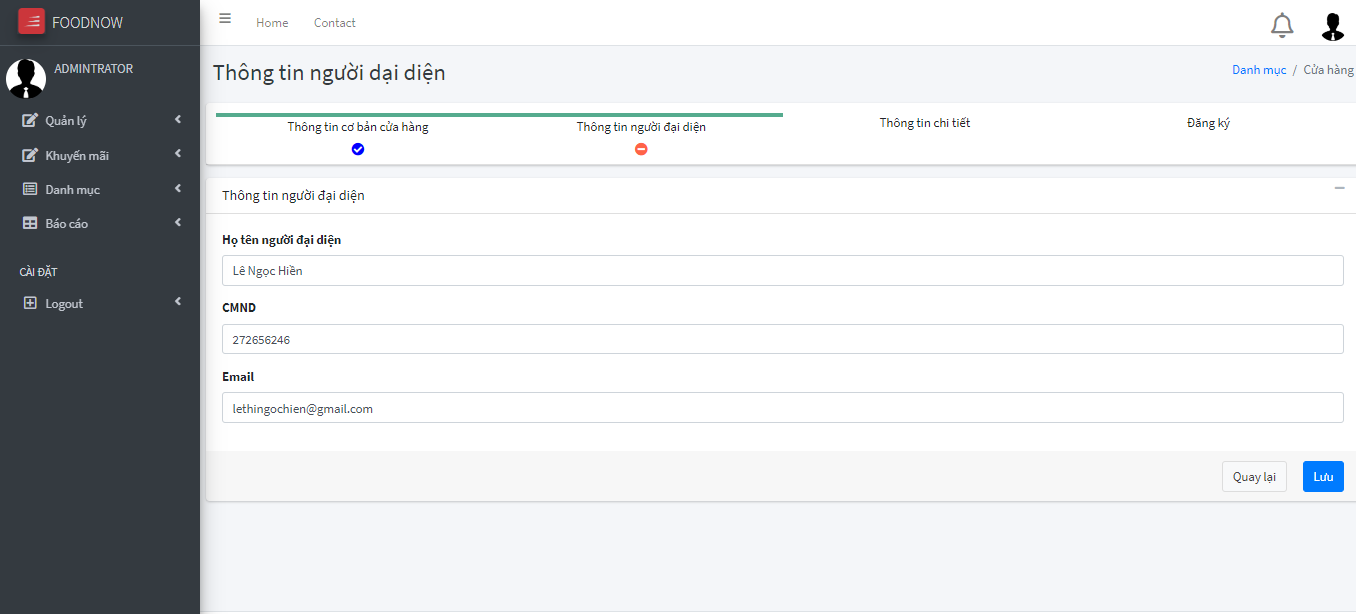
* 1. **QUẢN LÝ TÀI KHOẢN ĐĂNG KÝ**

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.

Tại phần quản lý địa chỉ này, nhóm đã tích hợp Google Map để hiển thị bản đồ sẽ dễ dàng *tìm kiếm địa chỉ hoặc nhập địa chỉ tại ô nhập.*

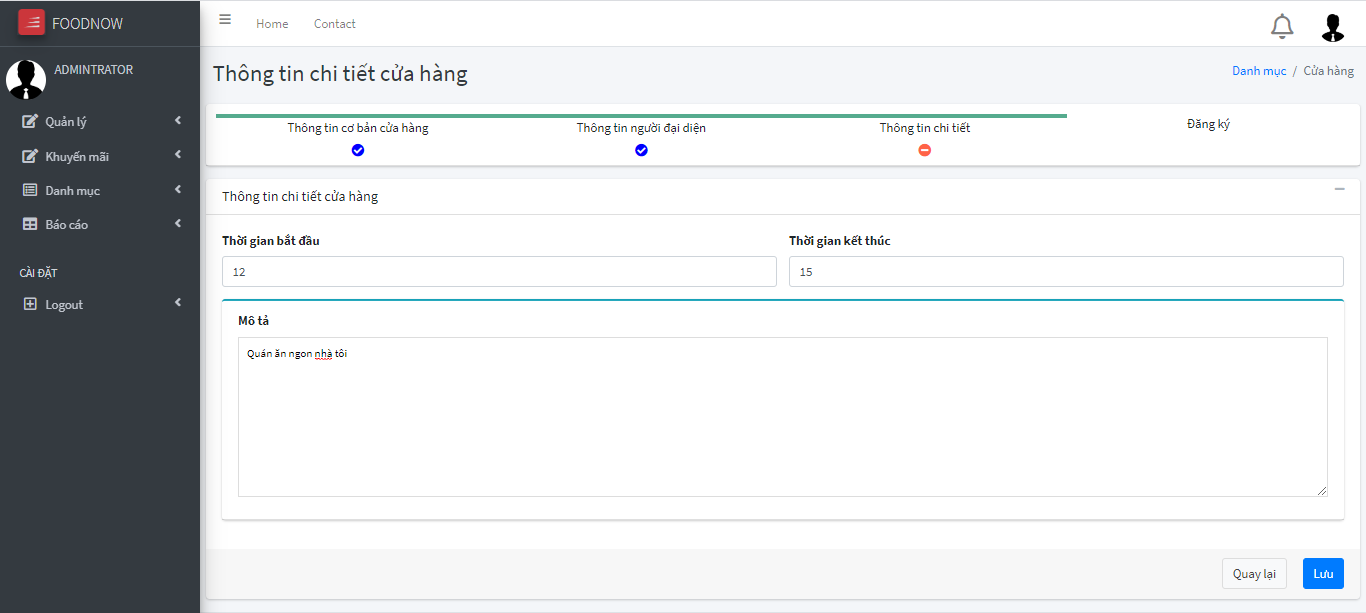
*Hình 2.3.31. Giao diện đăng ký cho cửa hàng thông tin cơ bản*

Trong phần thông tin người đại diện, quản trị sẽ điền các thông tin bắt buộc gồm: Họ tên người đại diện, chứng minh nhân dân, email rồi mới được lưu.



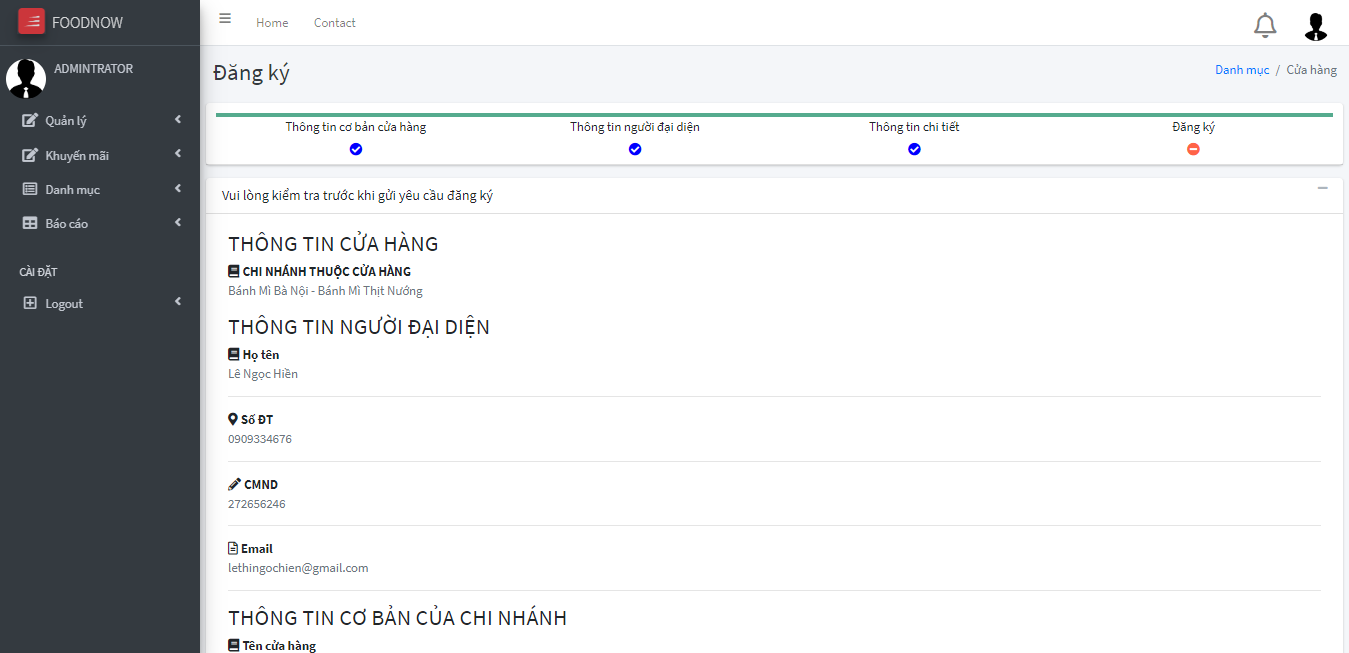
*Hình 2.3.32. Giao diện đăng ký cho cửa hàng thông tin người đại diện*

Quản trị sẽ tiếp tục điền thông tin chi tiết cửa hàng gồm: Thời gian bắt đầu, thời gian kết thúc và mô tả cửa hàng rồi mới được lưu.

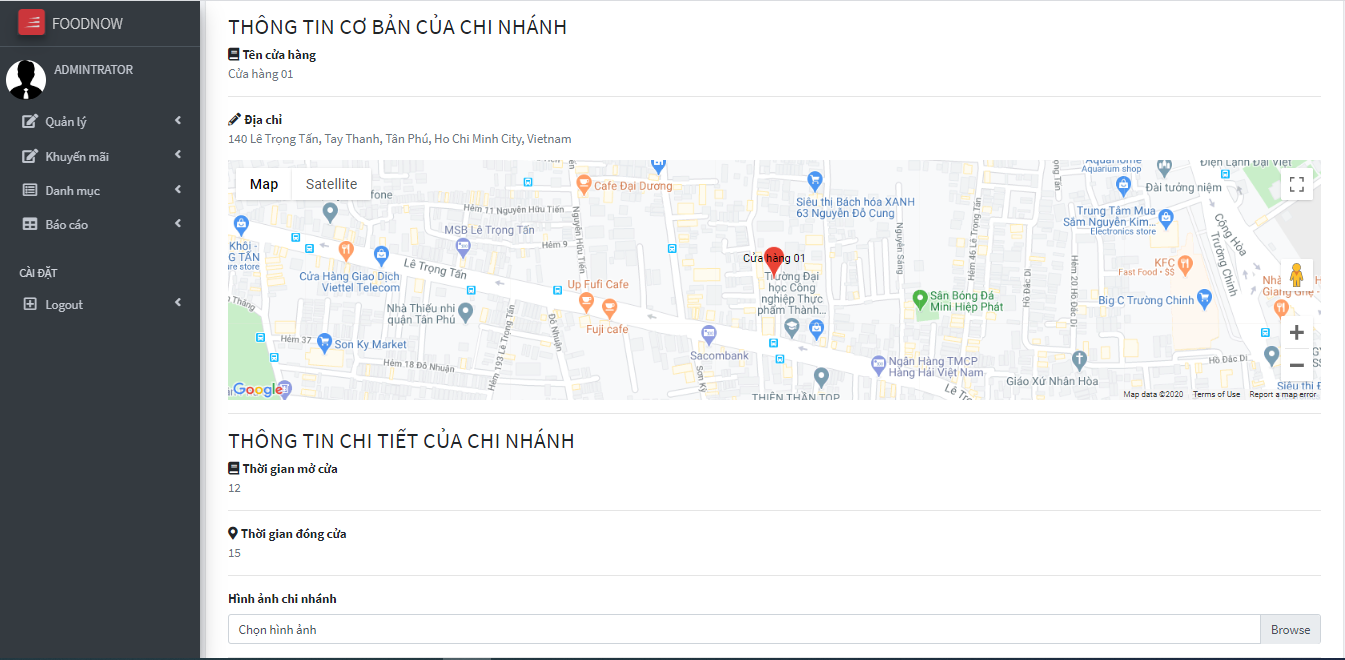


*Hình 2.3.33. Giao diện đăng ký cho cửa hàng thông tin chi tiết*

Phần đăng ký là bước cuối cùng để kiểm tra lại thông tin trước khi bấm đăng ký.

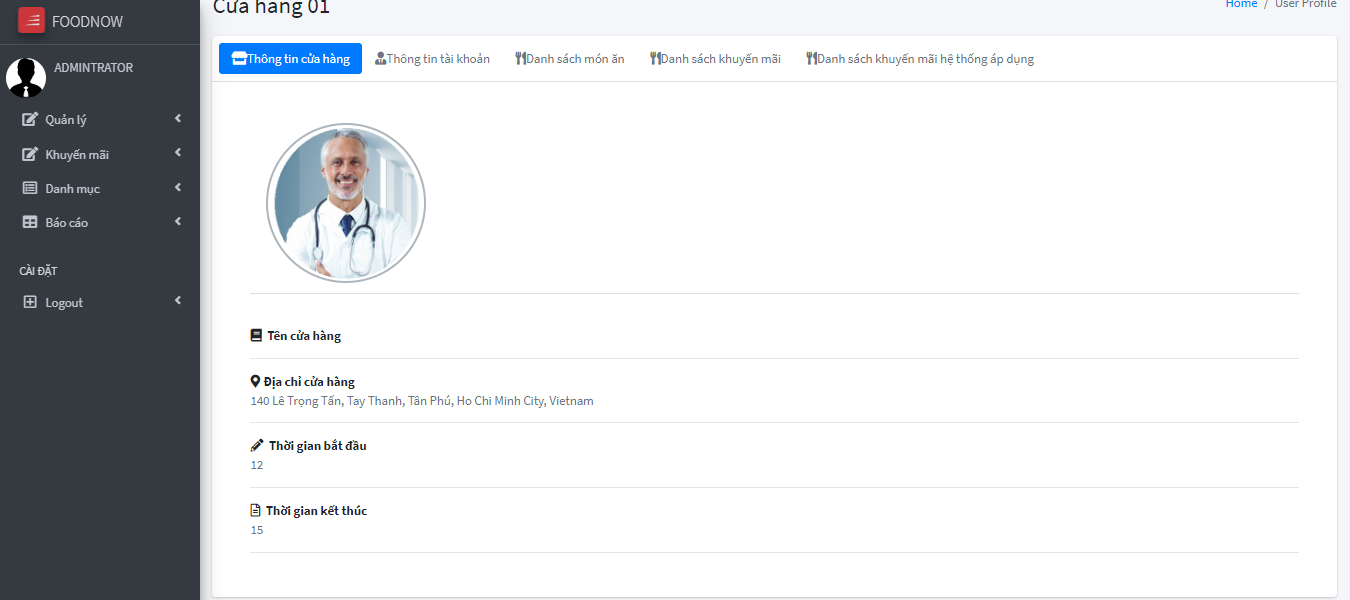


*Hình 2.3.34. Giao diện đăng ký cho cửa hàng kiểm tra thông tin*



*Hình 2.3.35. Giao diện đăng ký cho cửa hàng kiểm tra thông tin trước khi bấm đăng ký*

Khi bấm đăng ký xong sẽ tự động chuyển đến trang thông tin tài khoản cửa hàng để xem thông tin.



*Hình 2.3.36. Giao diện thông tin tài khoản cửa hàng khi đăng ký thành công*

* 1. **QUẢN LÝ CHI NHÁNH CỬA HÀNG**

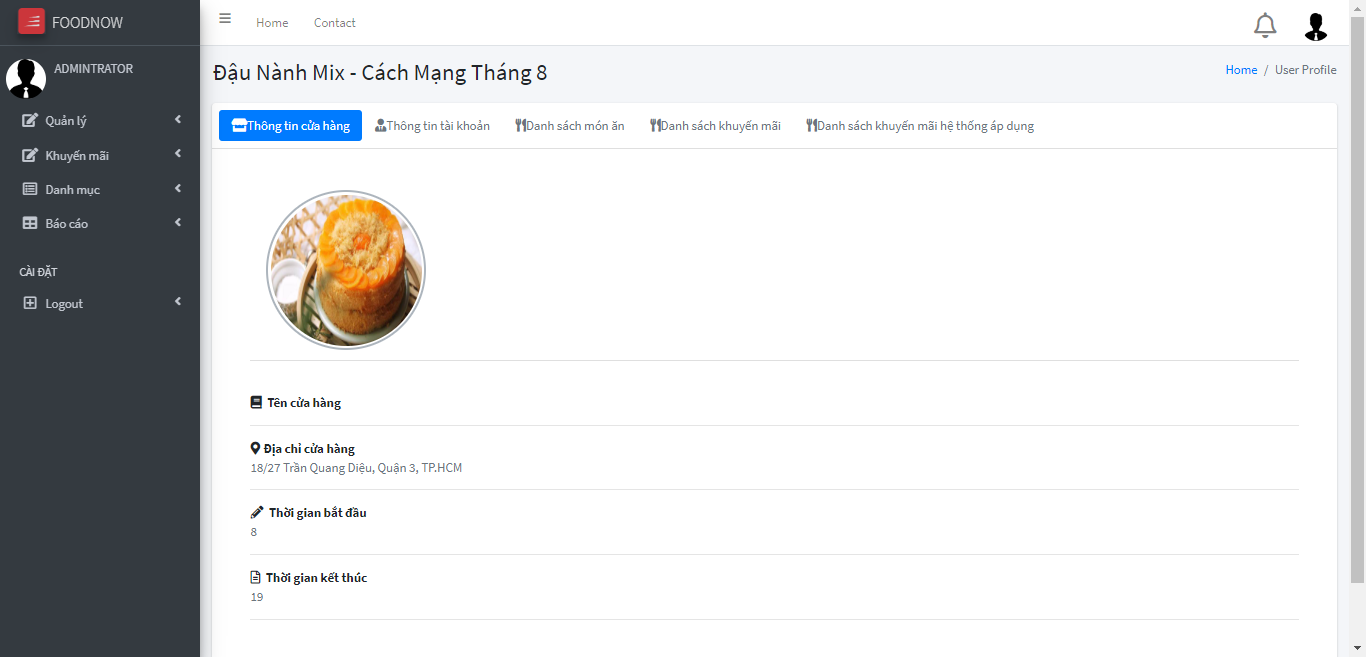
Quản trị viên có thể xem được cửa hàng vừa thêm thuộc chi nhánh nào. Cửa hàng vừa được thêm sẽ ở trạng thái chưa kích hoạt. Nếu chọn kích hoạt thì cửa hàng sẽ có quyền đăng nhập vào hệ thống để bắt đầu bán hàng. Tại danh sách cửa hàng, để có thể xem chi tiết cửa hàng đó thì quản trị chọn cửa hàng cần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



*Hình 2.3.37. Giao diện quản lý chi nhánh cửa hàng*

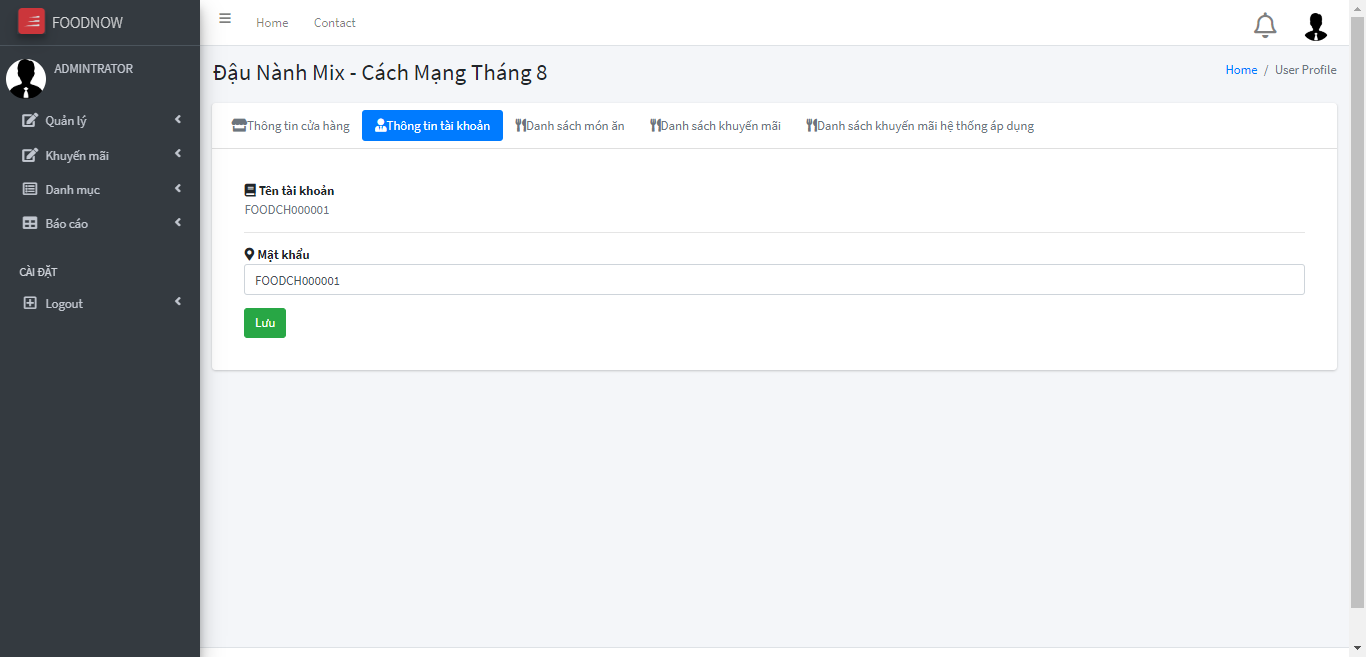
* 1. **THÔNG TIN TÀI KHOẢN CỬA HÀNG**

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.



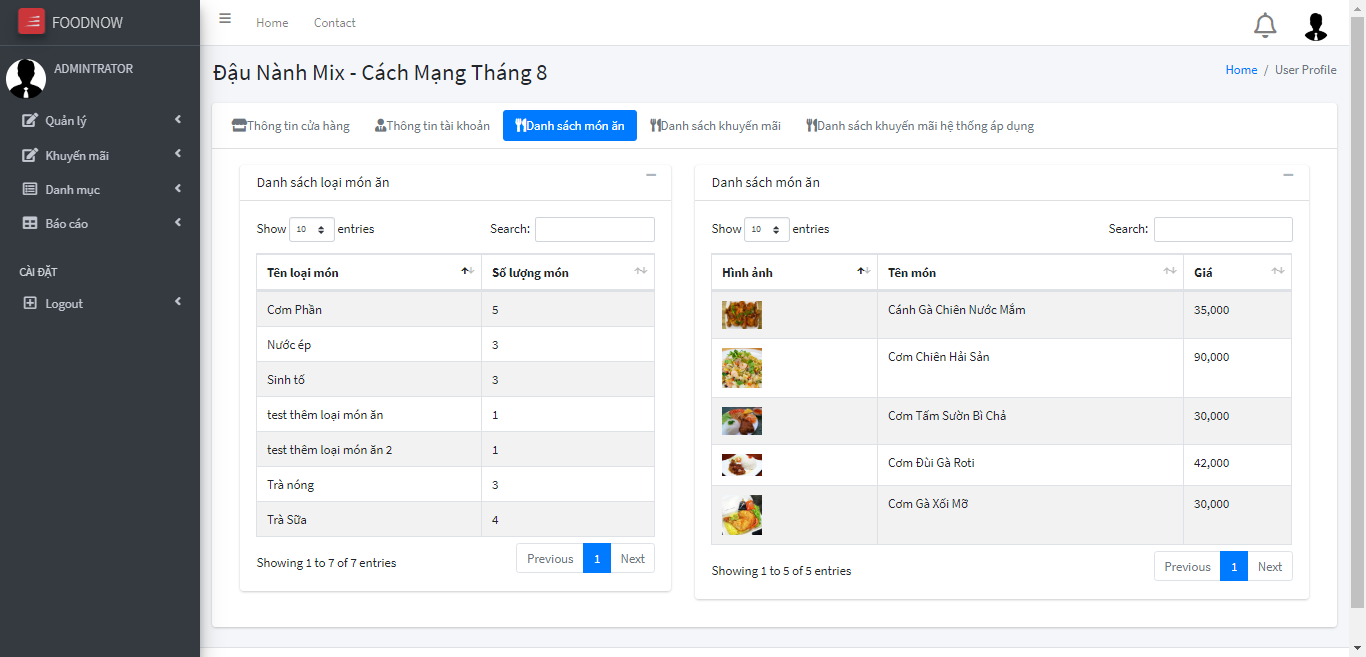
*Hình 2.3.38. Giao diện quản lý thông tin cửa hàng*

Tại đây quản trị có thể cấp lại mật khẩu khi cửa hàng không nhớ mật khẩu.



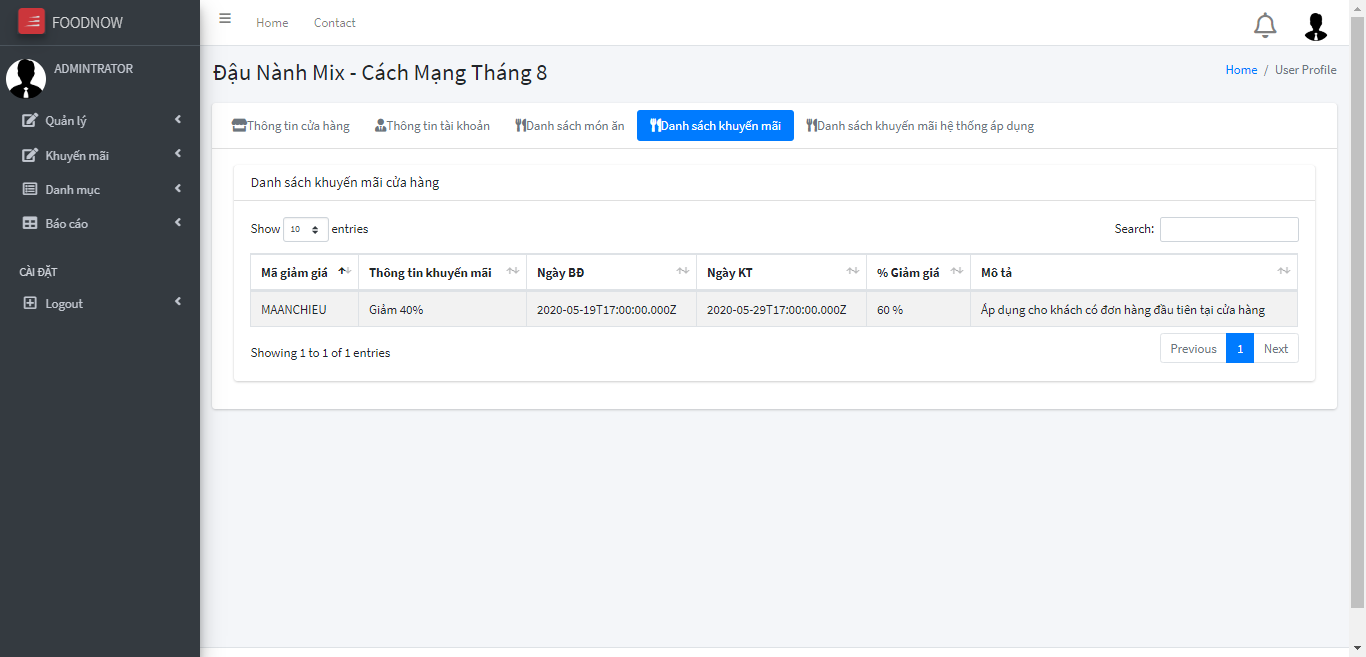
*Hình 2.3.39. Giao diện quản lý thông tin cửa hàng cài đặt tài khoản*

Quản trị có thể xem danh sách món ăn của cửa hàng.



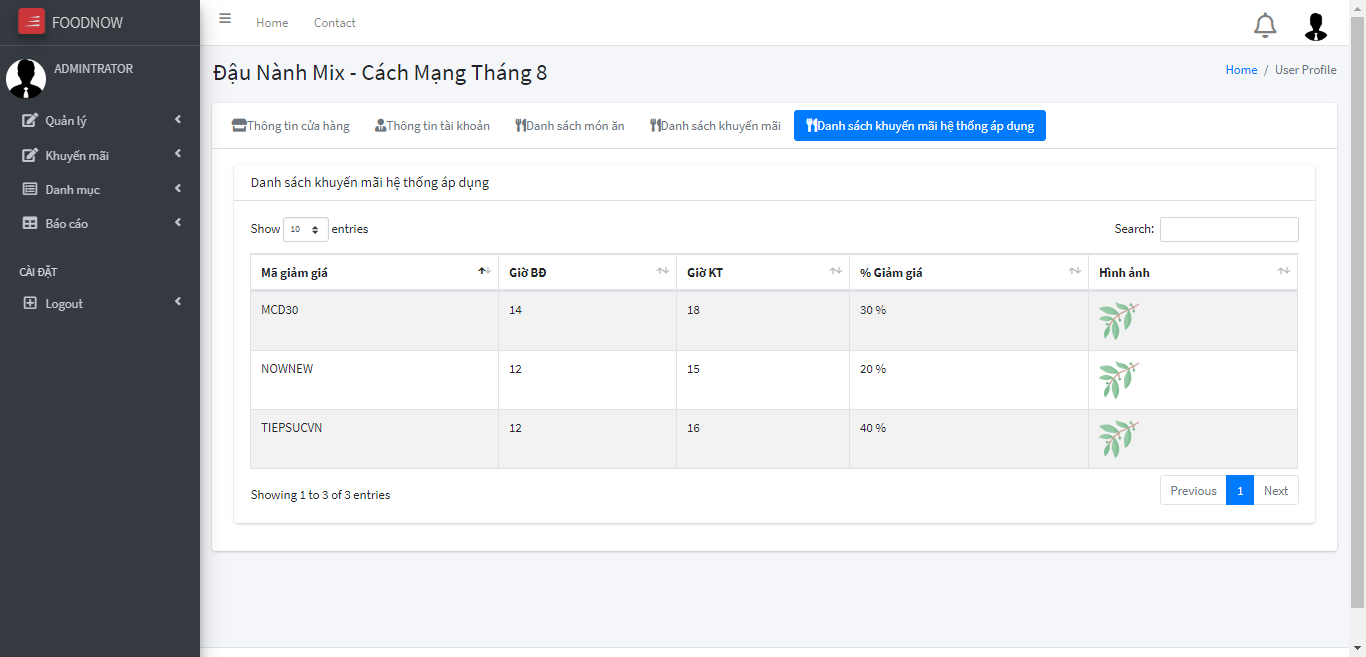
*Hình 2.3.40. Giao diện quản lý thông tin cửa hàng danh sách món ăn*

Quản trị có thể xem danh sách khuyến mãi mà cửa hàng áp dụng.



*Hình 2.3.41. Giao diện quản lý thông tin cửa hàng danh sách khuyến mãi*

Quản trị có thể xem khuyến mãi mà hệ thống áp dụng xuống cửa hàng.



*Hình 2.3.42. Giao diện quản lý thông tin cửa hàng danh sách khuyến mãi hệ thống áp dụng*

* 1. **QUẢN LÝ KHUYẾN MÃI**

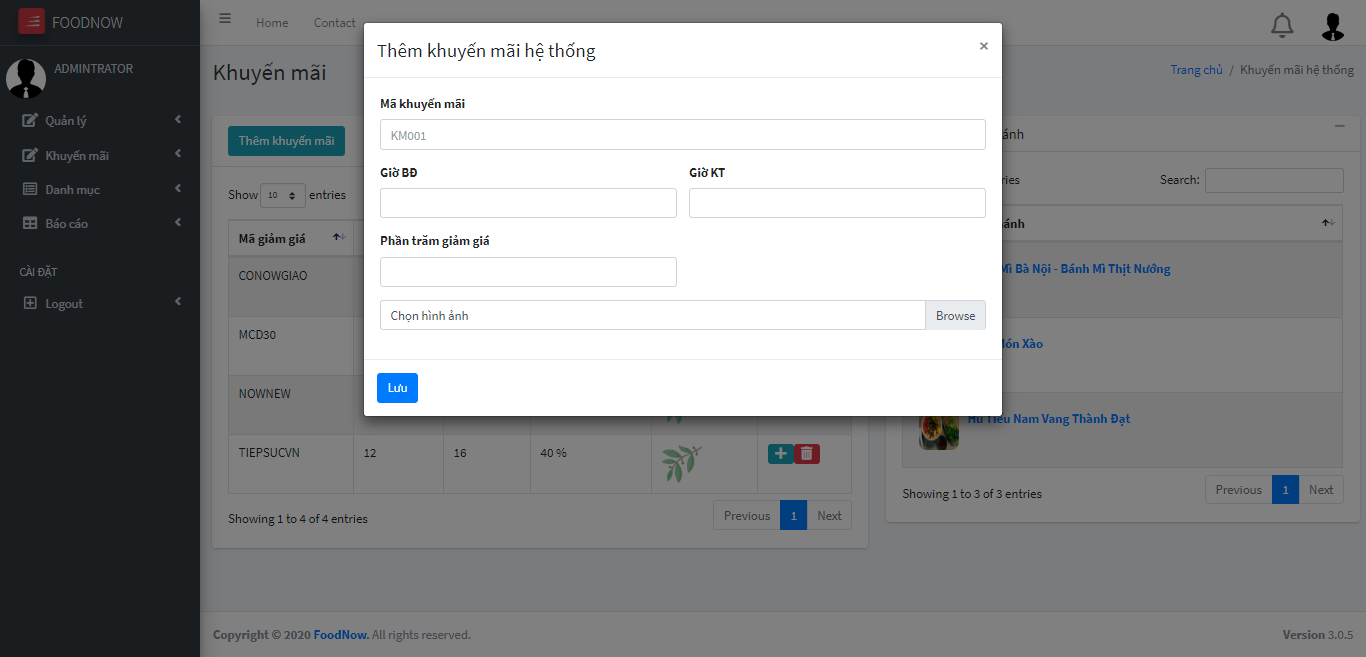
Giao diện khuyến mãi hệ thống sẽ hiển thị các chi nhánh khi quản trị chọn vào dòng khuyến mãi. Quản trị có thể thêm, xóa khuyến mãi, thêm khuyến mãi áp dụng cho chi nhánh bằng cách chọn vào dấu cộng. Hiển thị danh sách các chi nhánh và quản trị sẽ chọn vào ô cần áp dụng.

*Khi quản trị áp dụng những khuyến mãi này xuống cửa hàng, thông báo khuyến mãi sẽ gửi đến khách hàng. Nhóm đã ứng dụng Firebase Cloud Messaging để gửi thông báo.*



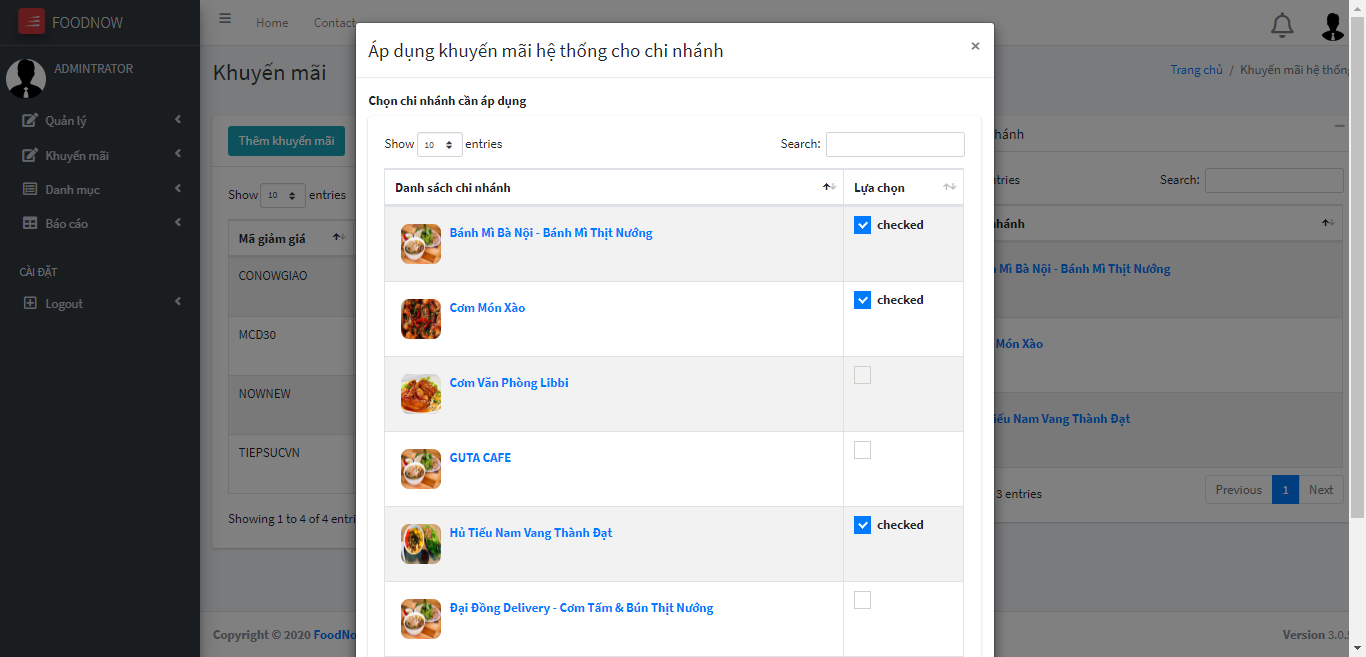
*Hình 2.3.43. Giao diện quản lý khuyến mãi hệ thống*

Quản trị có thể thêm mới khuyến mãi.



*Hình 2.3.44. Giao diện thêm mới khuyến mãi hệ thống*

Quản trị có thể áp dụng khuyến mãi hệ thống cho chi nhánh.



*Hình 2.3.45. Giao diện áp dụng khuyến mãi hệ thống cho chi nhánh*

* 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.

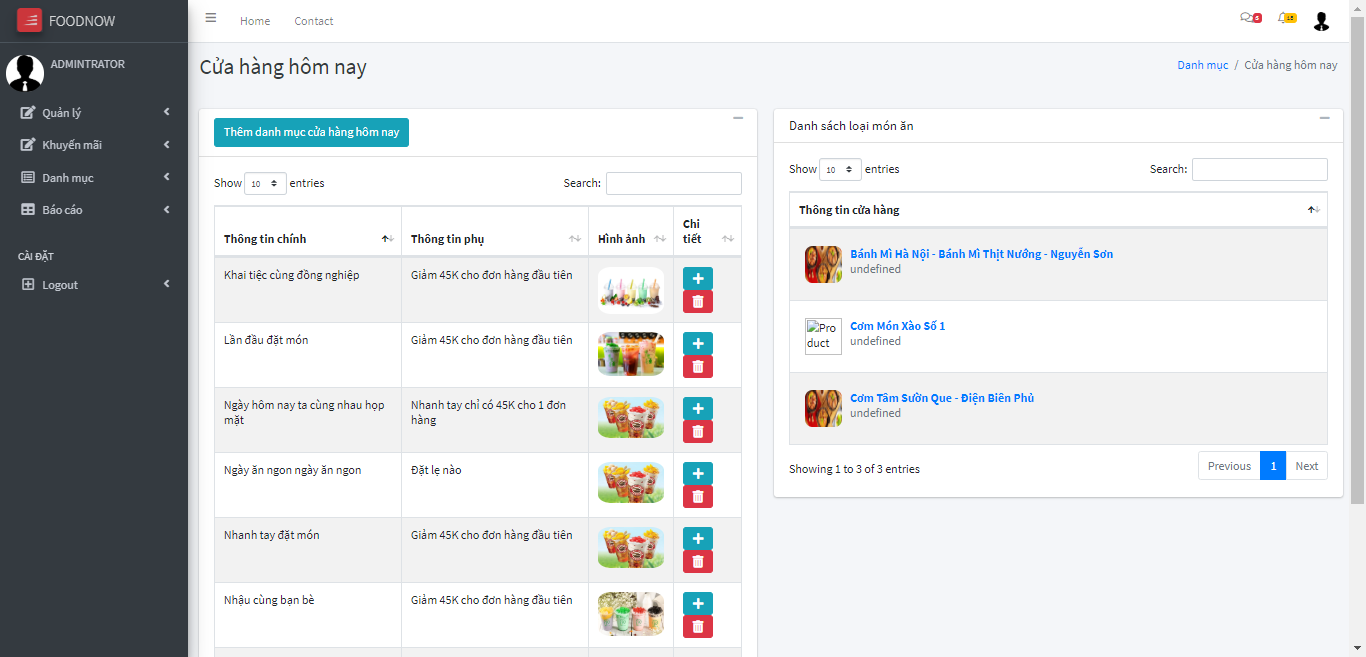
Trong từng loại sẽ hiển thị danh sách các cửa hàng, những cửa hàng được đánh dấu là những cửa hàng sẽ hiển thị trong loại món đó. Còn những cửa hàng không được đánh dấu sẽ không hiển thị trên loại món đó.



*Hình 2.3.46. Giao diện danh mục loại món ăn*

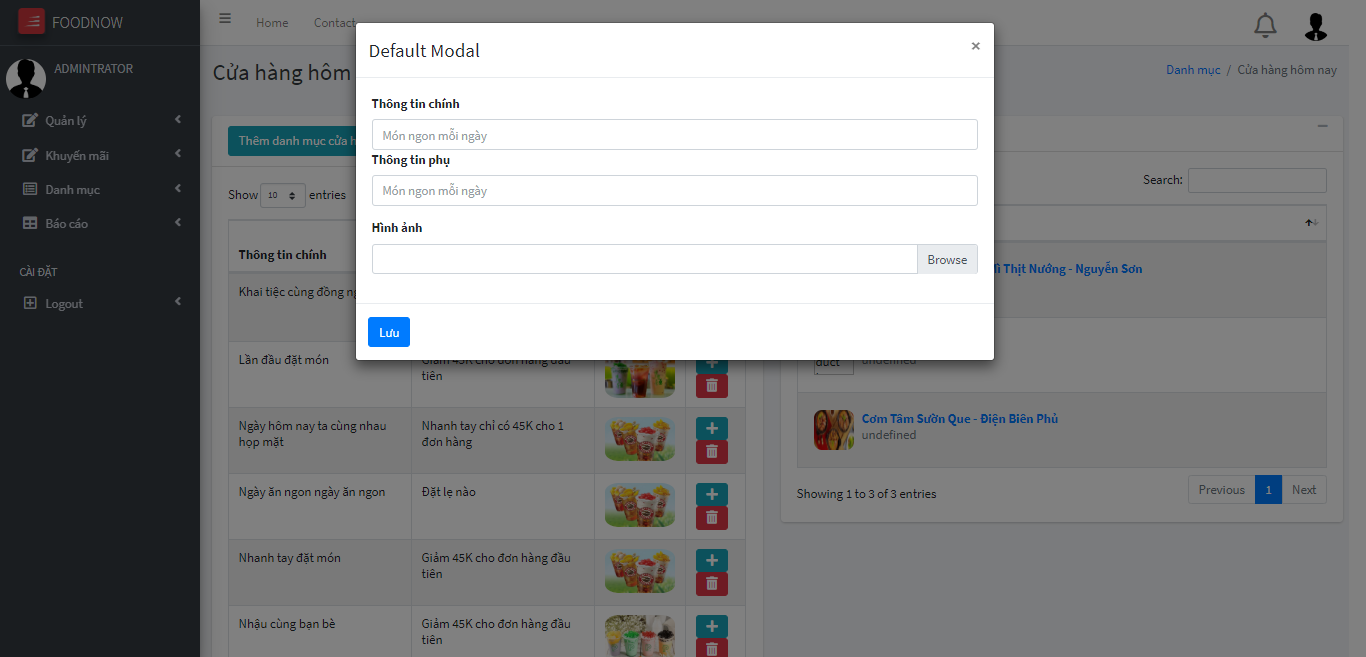
* + 1. **. DANH MỤC CỬA HÀNG HÔM NAY**

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.



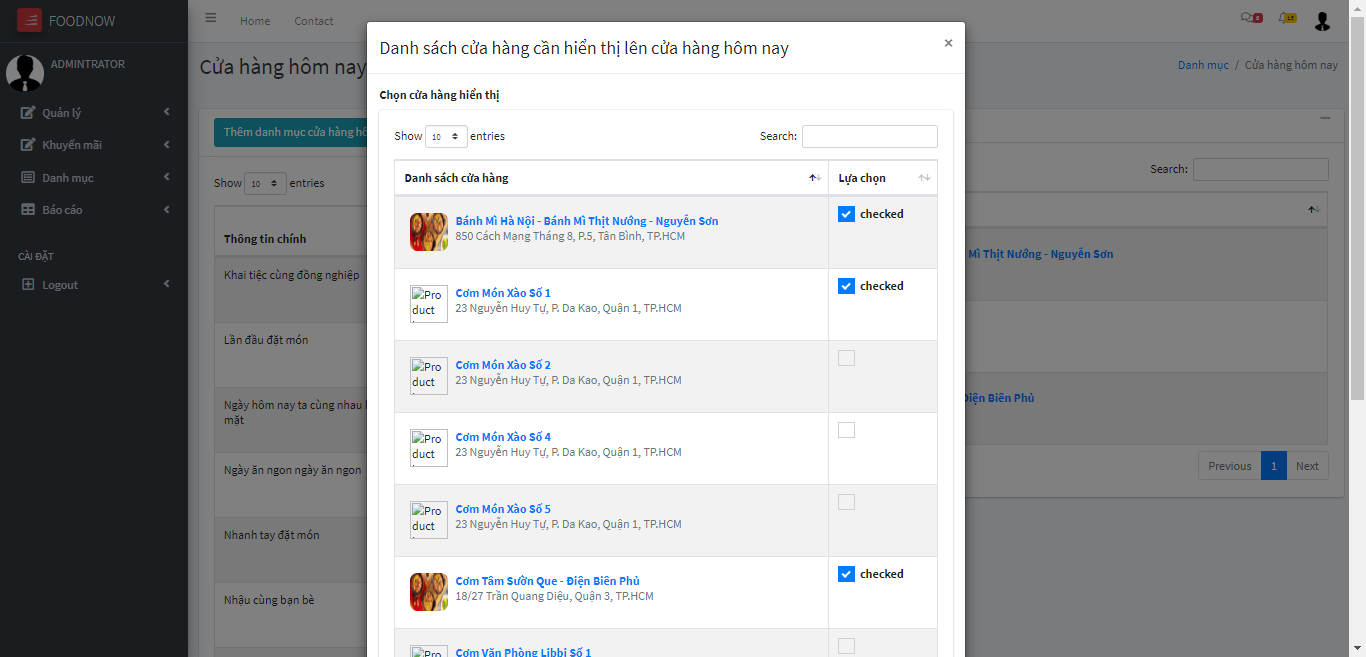
*Hình 2.3.47. Giao diện danh sách các nhóm trong danh mục cửa hàng hôm nay*

Quản trị có thể thêm mới một nhóm trong danh mục.



*Hình 2.3.48. Giao diện thêm mới một nhóm nằm trong danh mục cửa hàng hôm nay*

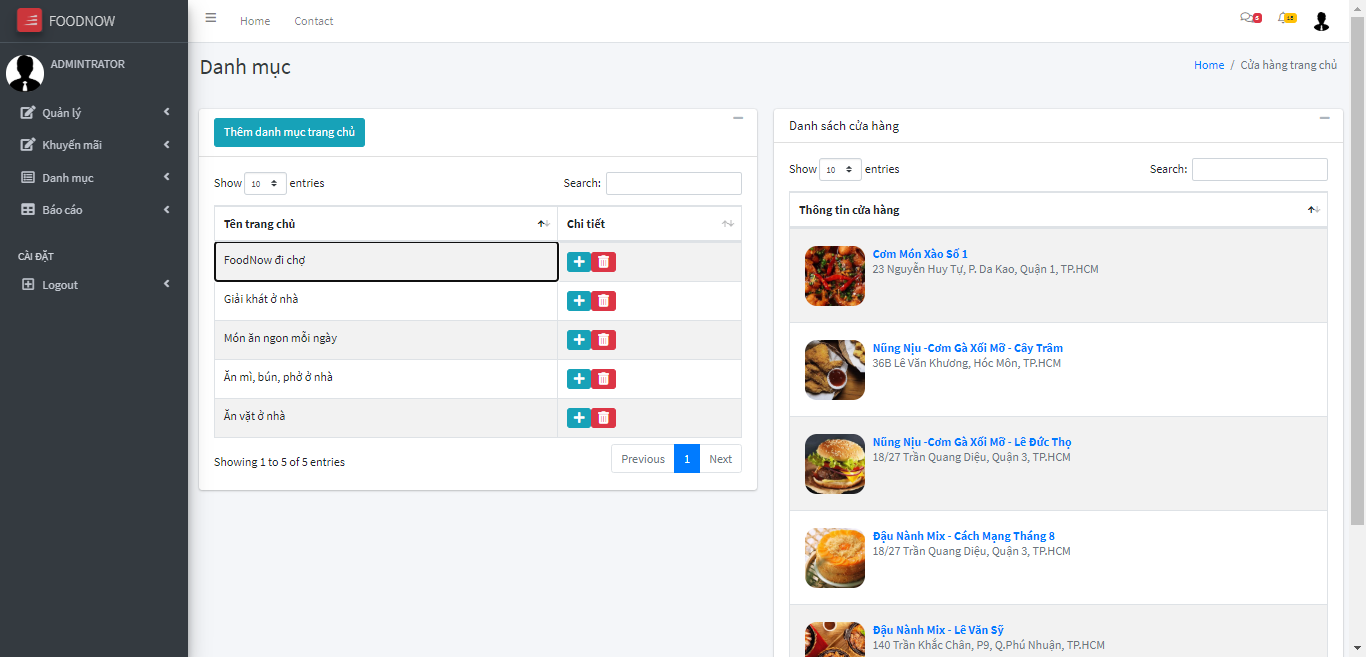
Quản trị có thể thêm các cửa hàng vào từng nhóm trong danh mục.



*Hình 2.3.49. Giao diện thêm các cửa hàng vào từng nhóm trong danh mục cửa hàng hôm nay.*

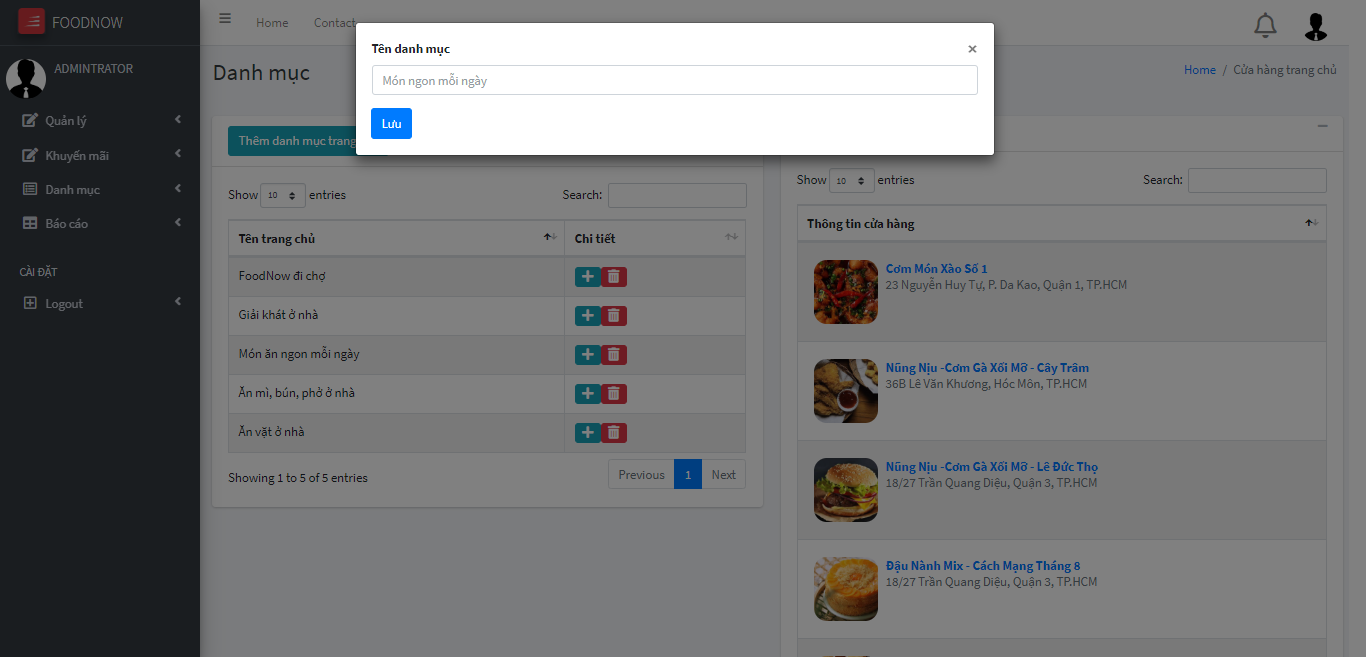
* + 1. . **DANH MỤC CỬA HÀNG TRANG CHỦ**

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.



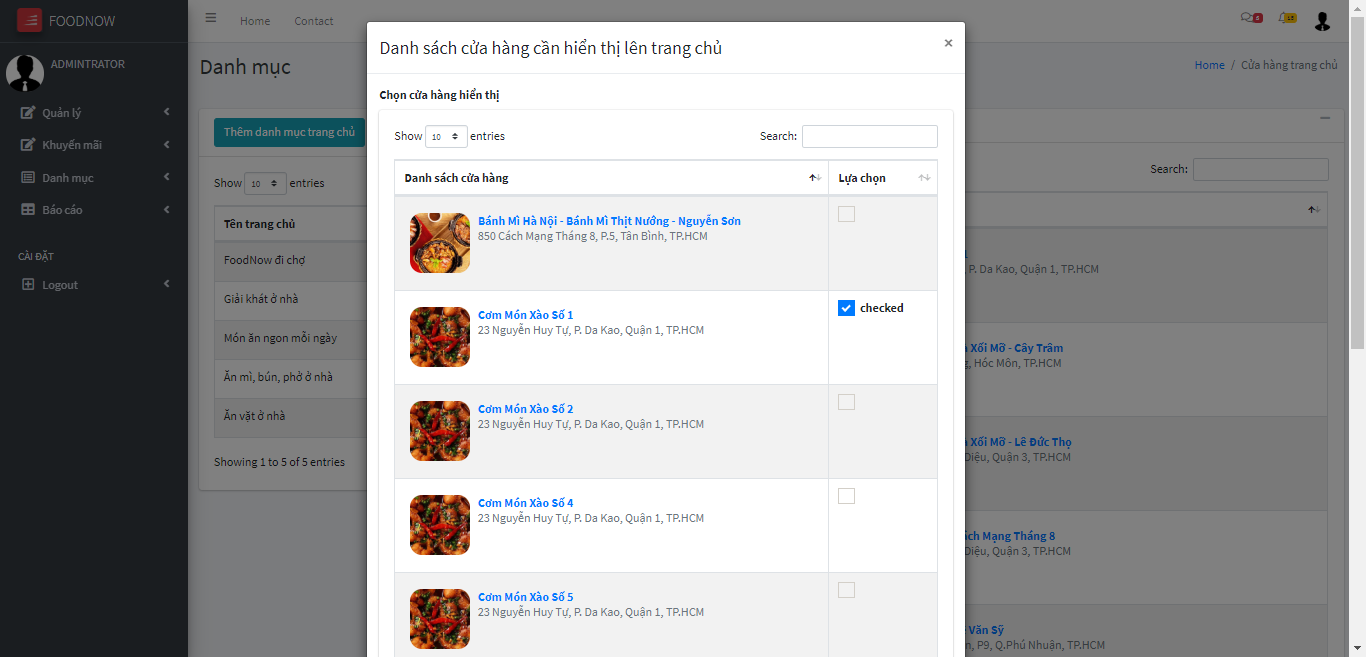
h*Hình 2.3.50. Giao diện danh sách các nhóm trong danh mục cửa hàng trang chủ.*

Quản trị có thể thêm mới nhóm trong danh mục.



*Hình 2.3.51. Giao diện thêm mới nhóm trong danh mục.*

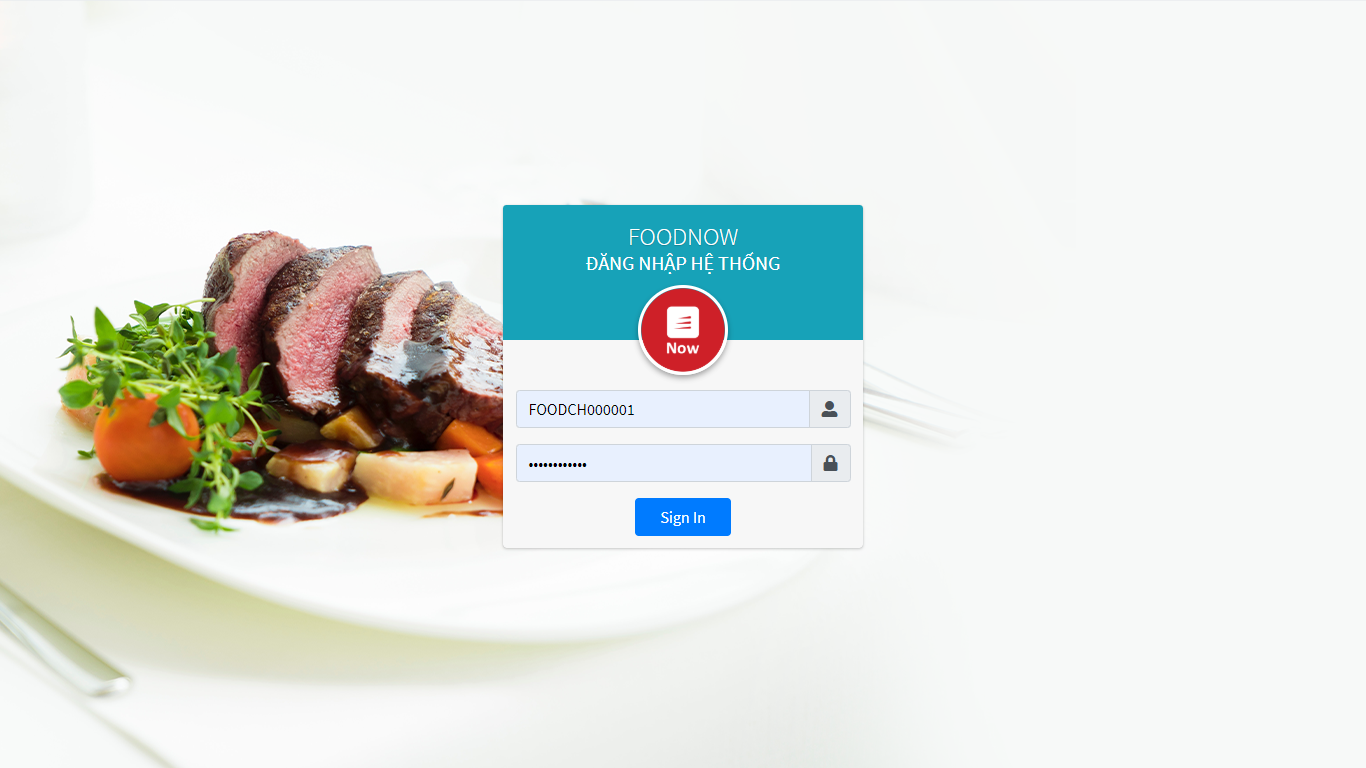
Quản trị có thể thêm các cửa hàng vào danh mục.



*Hình 2.3.52. Giao diện thêm các cửa hàng vào từng nhóm trong danh mục trang chủ.*

1. **GIAO DIỆN WEB CỬA HÀNG**
   1. **ĐĂNG NHẬP**

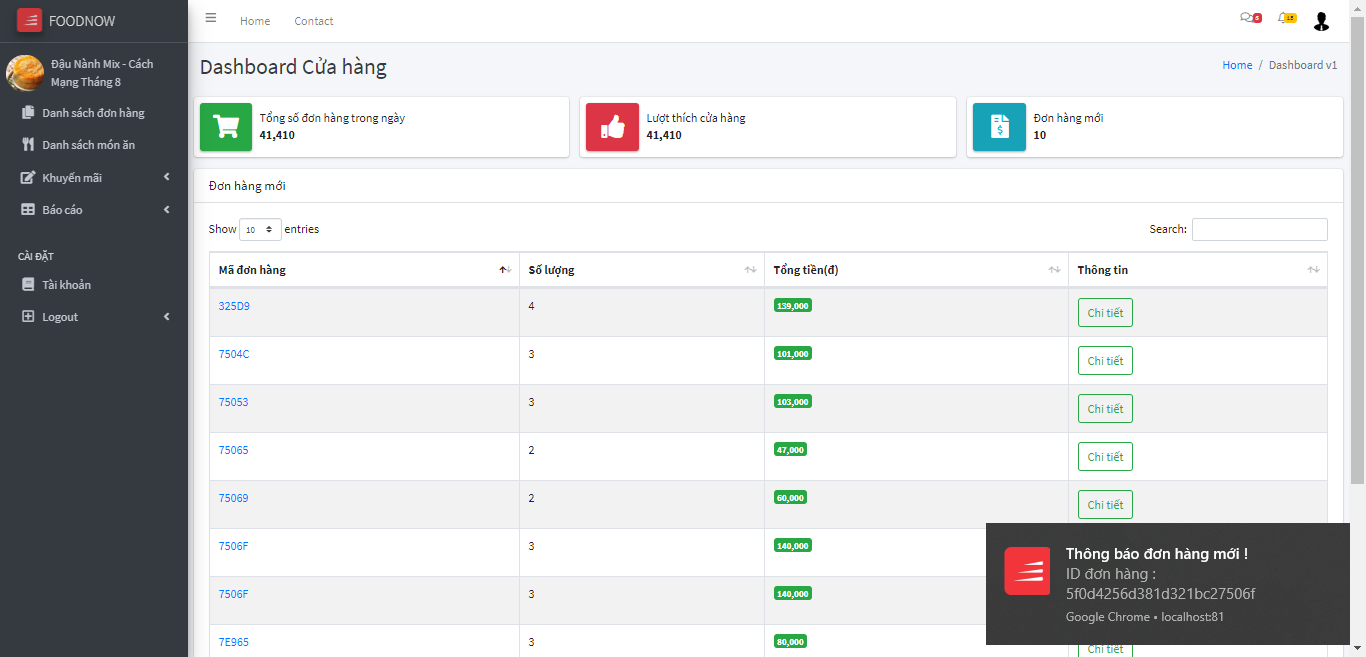
Cửa hàng sẽ đăng nhập bằng tài khoản được hệ thống cấp.



*Hình 2.3.53. Giao diện đăng nhập cửa hàng.*

Sau khi đăng nhập thành công sẽ hiển thị trang tổng quan để cửa hàng dễ theo dõi các đơn hàng mới đến. Cửa hàng chọn vào chi tiết tự động sẽ đến trang danh sách đơn hàng để cửa hàng xác nhận đơn, đồng thời có thể xuất hóa đơn cho đơn đó.

*Tại đây nhóm đã ứng dụng Firebase cloud messaging để hiển thị thông báo của đơn hàng khách hàng đặt đến.*

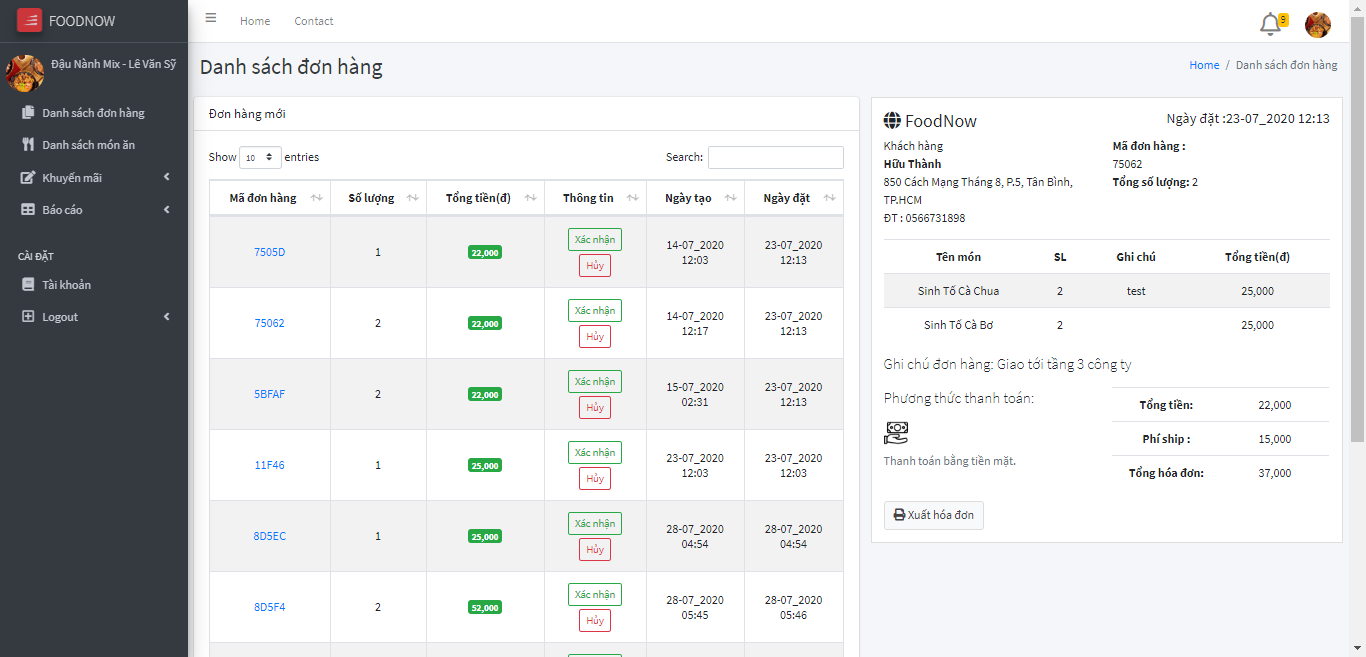


*Hình 2.3.54. Giao diện tổng quan cửa hàng.*

* 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.

*Tại đây nhóm ứng dụng Firebase cloud messaging khi cửa hàng xác nhận/ hủy đơn hàng sẽ thông báo cho khách hàng. Trong từng đơn cửa hàng sẽ xem được món ăn và xuất hóa đơn.*



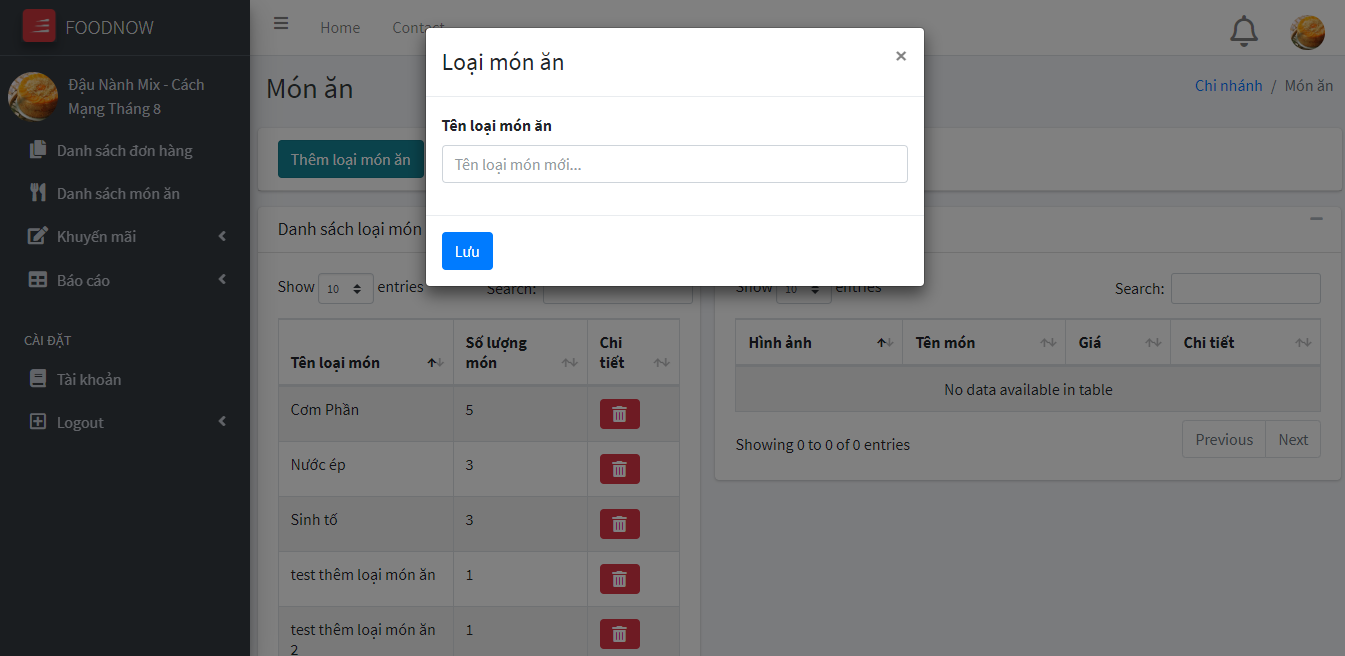
*Hình 2.3.55. Giao diện danh sách đơn hàng cửa hàng.*

* 1. **DANH SÁCH MÓN ĂN**

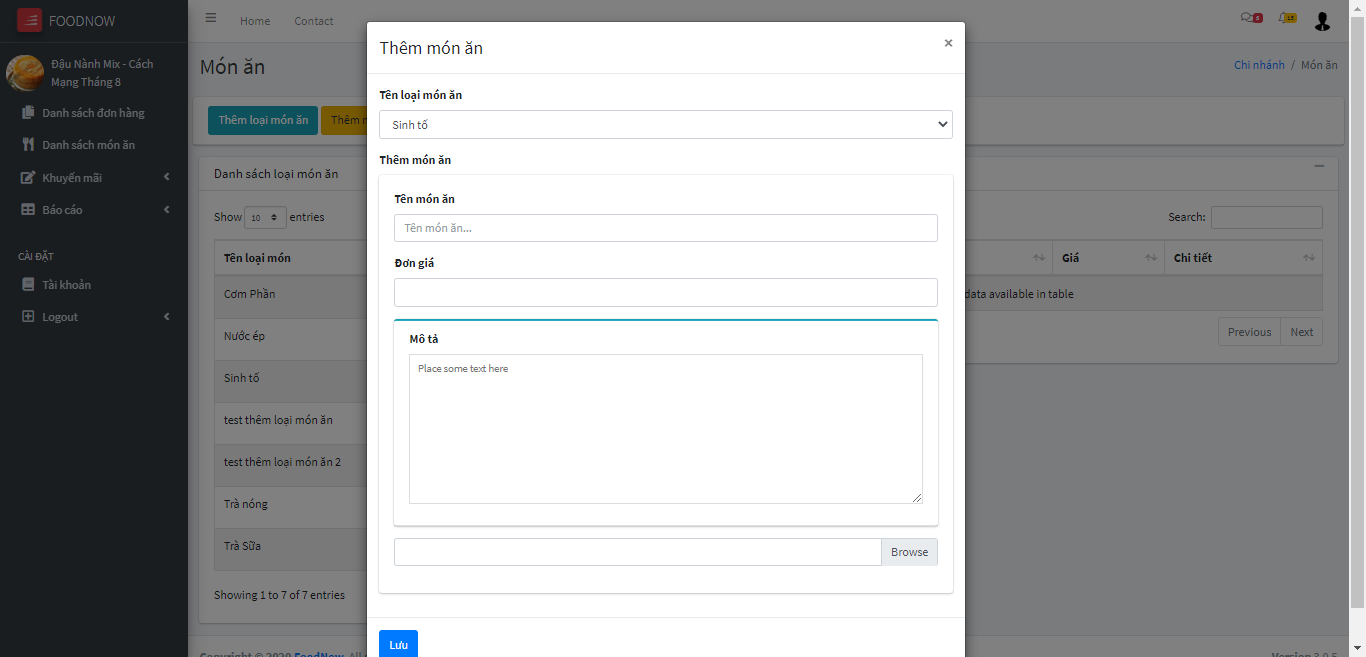
Cửa hàng có thể xem danh sách loại món ăn và món ăn của từng loại. Cửa hàng có thể thêm mới, xóa loại món ăn và món ăn.



*Hình 2.3.56. Giao diện danh sách loại món ăn và món ăn.*



*Hình 2.3.57. Giao diện thêm mới loại món ăn.*

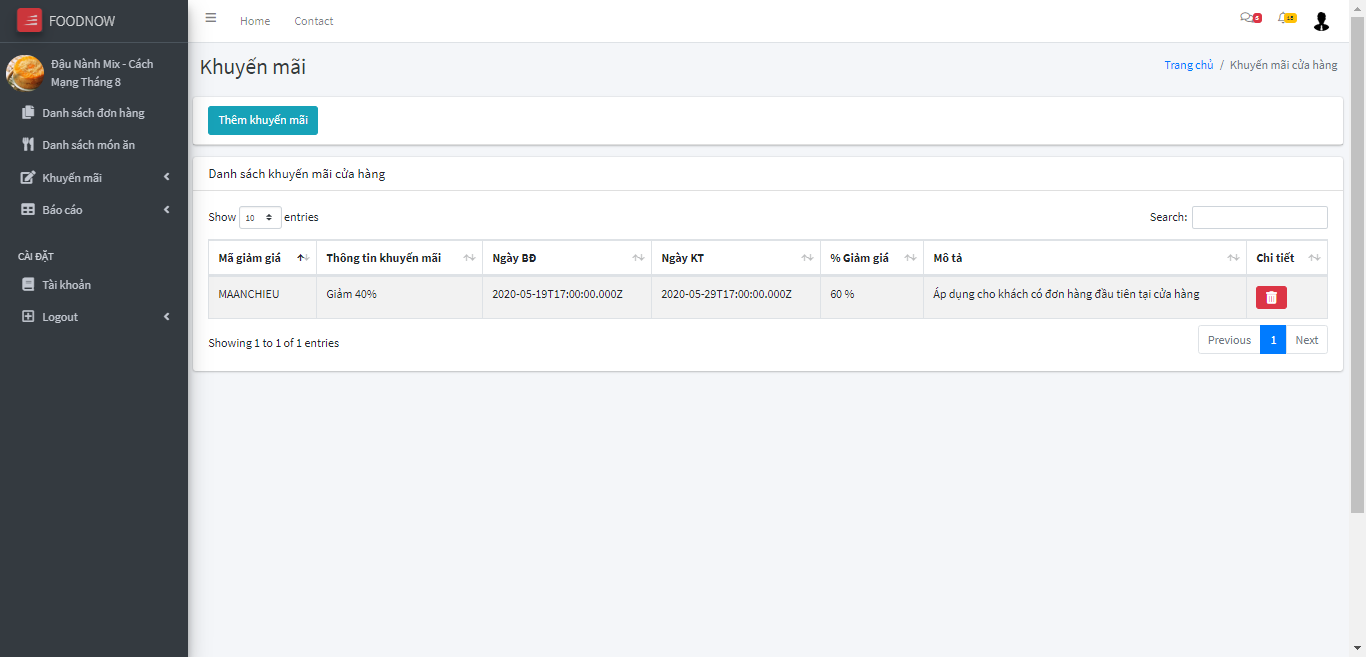


*Hình 2.3.58. Giao diện thêm món ăn.*

* 1. **DANH SÁCH KHUYẾN MÃI**

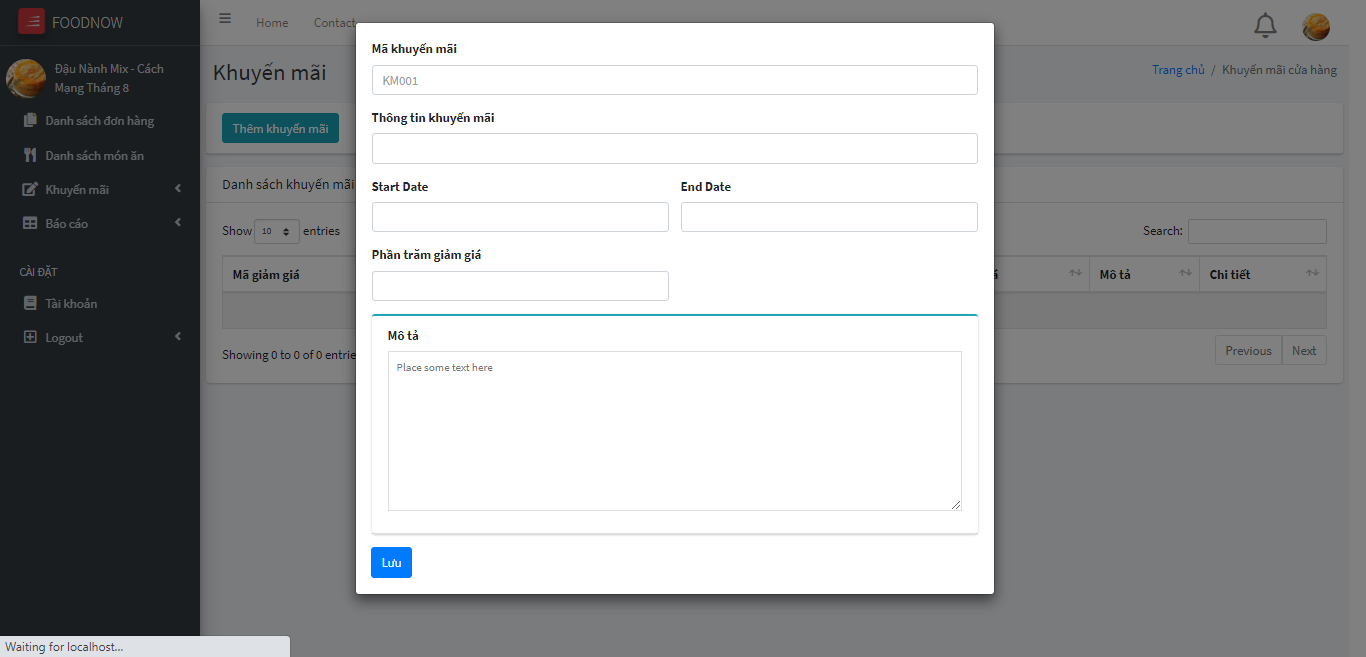
Cửa hàng có thể thêm mới, xóa khuyến mãi.

*Tại đây khi cửa hàng thêm khuyến mãi mới sẽ thông báo đến khách hàng yêu thích cửa hàng. Nhóm đã ứng dụng Firebase Cloud Messaging để thông báo.*



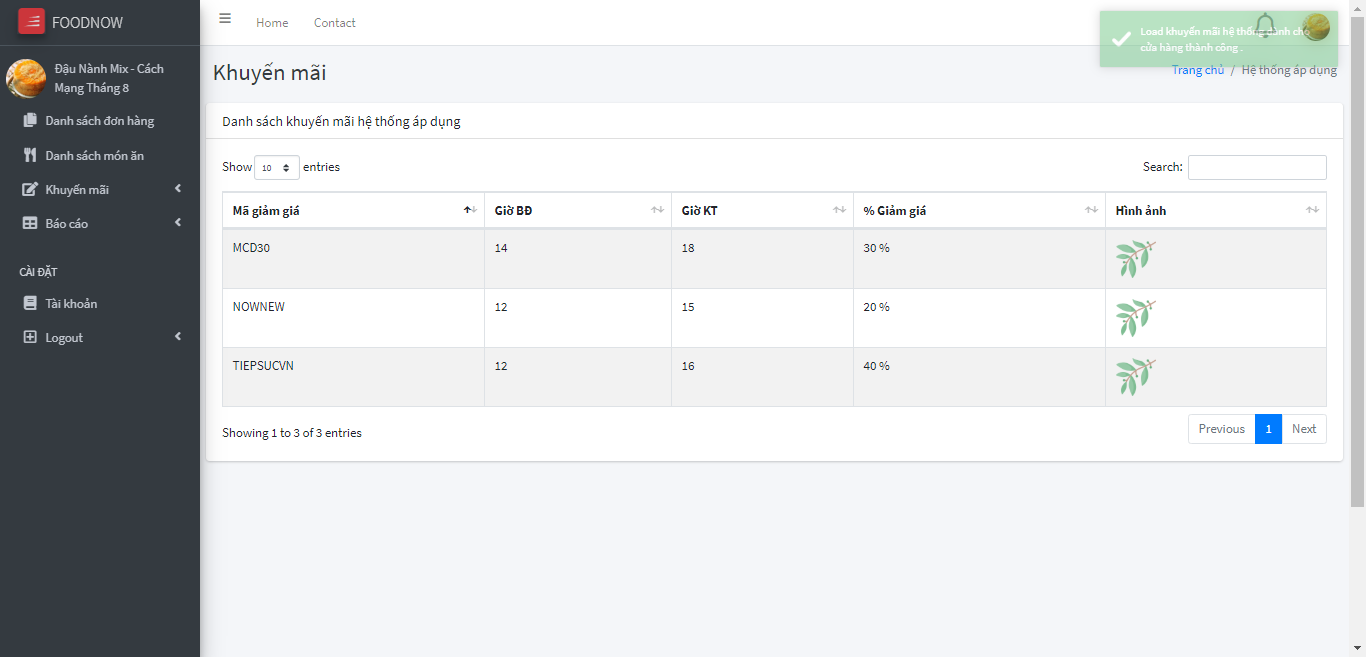
*Hình 2.3.59. Giao diện danh sách khuyến mãi cửa hàng.*

Cửa hàng có thể thêm khuyến mãi khi điền đầy đủ thông tin.



*Hình 2.3.60. Giao diện thêm mới khuyến mãi cửa hàng.*

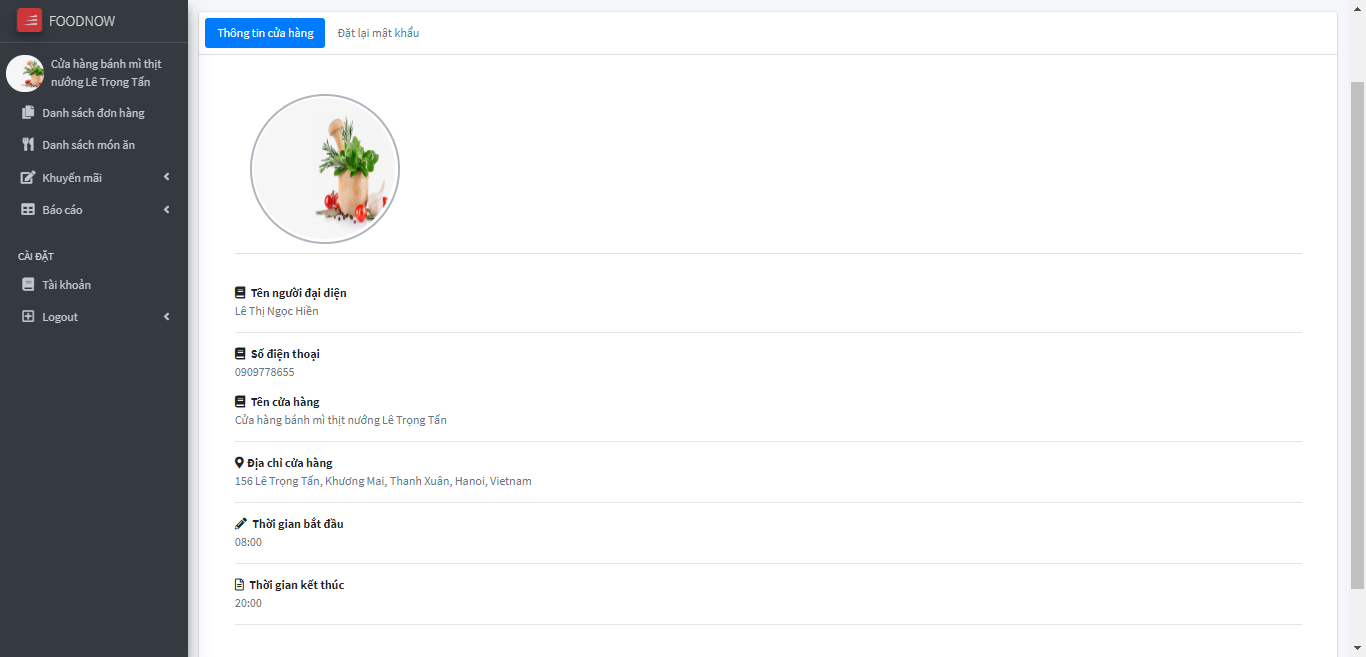
Cửa hàng có thể xem khuyến mãi hệ thống áp dụng cho cửa hàng.



*Hình 2.3.61. Giao diện khuyến mãi hệ thống áp dụng cho cửa hàng.*

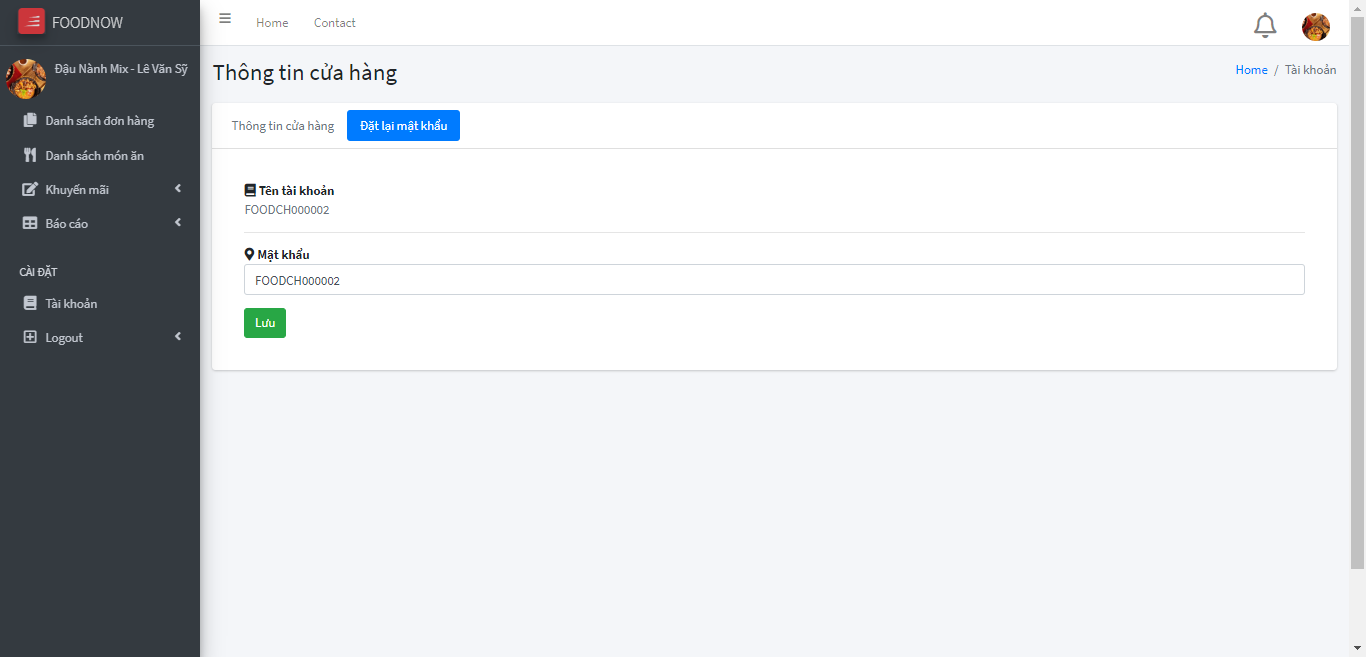
* 1. **TÀI KHOẢN**

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

**

*Hình 2.3.62. Giao diện thông tin tài khoản cửa hàng.*

Cửa hàng có thể thay đổi mật khẩu của mình.

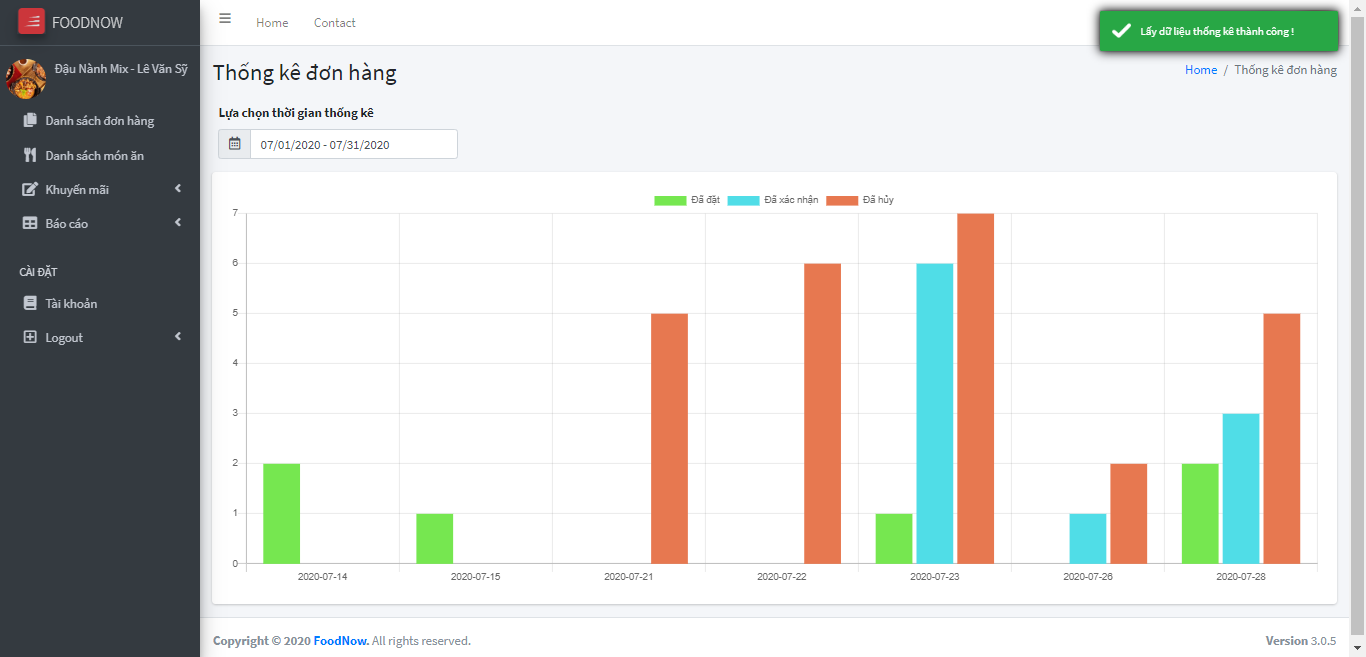


*Hình 2.3.63. Giao diện thông tin tài khoản cửa hàng cài đặt tài khoản.*

* 1. **ĐƠN HÀNG THỐNG KÊ**

Cửa hàng sẽ xem được tổng số đơn hàng đã xuất hóa đơn, đã hủy thông qua biểu đồ này để theo dõi được tình hình của cửa hàng.

*Tại phần biểu đồ này nhóm đã ứng dụng biểu đồ để thống kê đơn hàng gồm: đơn hàng đã đặt, đã xác nhận và đã hủy. Khi cửa hàng chọn ngày cần xem, biểu đồ sẽ thể hiện theo đúng số ngày chọn trên biểu đồ.*



*Hình 2.3.64. Thống kê đơn hàng*

**KẾT PHẦN 2**

* Trong phần triển khai đề tài, qua việc phân tích và thiết kế, nhóm đã xây dựng nên một ứng dụng hoàn chỉnh gồm app khách hàng và website quản trị viên, cửa hàng.
* Về phần phân tích: Xây dựng các quy trình nghiệp vụ.
* Về phần thiết kế hệ thống:
* Thiết kế cơ sở dữ liệu dưới dạng NoSql.
* Thiết kế API chức năng.
* Về phần thiết kế giao diện:
* Thiết kế ứng dụng dành cho khách hàng
* Thiết kế web dành cho cửa hàng và quản trị viên.
* Lập trình chức năng cho ứng dụng.

**PHẦN 3: TỔNG KẾT**

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

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

* Cách lập trình một ứng dụng Android.
* Kiến thức về lập trình NodeJS.
* Kiến thức về cơ sở dữ liệu NoSql và MongoDB
* Kiến thức tạo API sử dụng trong app và web.
* Ứng dụng Google API, Facebook API vào đăng ký, đăng nhập tài khoản.
* Ứng dụng Firebase cho phần thời gian thực.
* Ứng dụng Google Map vào việc tính khoảng cách và tìm địa chỉ.
* Sử dụng thuật toán RSA và SHA256 vào thanh toán trực tuyến bằng ví Momo.
* 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 lớn 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 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 sẽ có thêm app giao hàng dành cho nhân viên giao hàng.
* 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:**

[5] MOMO PAYMENT APIs DOCUMENT, from <<https://business.momo.vn/assets//docs/api/MoMo-Payment-APIs-v2.1.5.pdf?version=16>>

[6] FACEBOOK API, from<<https://developers.facebook.com/docs/facebook-login/overview>>

[7] NODEJS, from <<https://www.sohamkamani.com/blog/javascript/2019-03-29-node-jwt-authentication/>>

[8] MOGODB, from < <https://docs.mongodb.com/manual/core/document/>>

**BOOK:**

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

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

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

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

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

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