|  |
| --- |
| **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **KHOA CÔNG NGHỆ THÔNG TIN \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Logo HaUI ban chuan  **ĐỒ ÁN**  **TỐT NGHIỆP ĐẠI HỌC**  **NGÀNH KHOA HỌC MÁY TÍNH**  ***Đề tài:***  **TÌM HIỂU VỀ ANDROID, MYSQL VÀ XÂY DỰNG ỨNG DỤNG "MÓN NGON ĐƯỜNG PHỐ" TRÊN ANDROID**  **Giảng viên hướng dẫn: ThS. Nguyễn Văn Tỉnh**  **Sinh viên thực hiện: Đào Thị Hằng  Mã sinh viên: 1041060269**    ***Hà Nội, 5/2019*** |

**LỜI NÓI ĐẦU**

Hiện nay công nghệ thông tin đang được ứng dụng trong nhiều lĩnh vực nhiều ngành nghề giúp con người giảm thiếu tối đa sức lao động, tiện ích về nhiều mặt và việc áp dụng vào ẩm thực cũng ngày càng nhiều. Việc làm thế nào để có bữa ăn ngon ấm cúng bên gia đình người thân bạn bè mà không cần phải đi hỏi ý kiến mọi người cách chế biến món ăn vất vả, ngoài ra còn chỉ đường cho khách du lịch tìm được nhưng món ngon ngay tại địa điểm mình tham quan và đến nhà hàng gần đó để thưởng thức đồng thời cũng quảng bá được các món ăn ngon của Việt Nam ra quốc tế vì vậy em đã chọn đề tài: tìm hiểu về android, mysql và xây dựng ứng dụng "món ngon đường phố" trên android”. Ứng dụng này hiện em chọn nền tảng Android để xây dựng vì thị trường android chiếm hơn 80%. Em xin trân thành cảm ơn thầy Nguyễn Văn Tỉnh đã giúp đỡ em để xây dựng nên ứng dụng: tìm hiểu về android, mysql và xây dựng ứng dụng "món ngon đường phố" trên android.

Một lần nữa em xin cảm ơn các thầy cô đã tạo điều kiện cho em được học tập, nghiên cứu và hoàn thành đề tài đồ án tốt nghiệp này. Kính chúc thầy cô luôn mạnh khỏe và công tác tốt!

Hà Nội, tháng 5/2019

Sinh viên thực hiện

*Đào Thị Hằng*

Mục lục

[Danh mục các ký hiệu các chữ viết tắt 3](#_Toc8063588)

[Danh mục bảng 4](#_Toc8063589)

[Danh mục các hình vẽ, đồ thị 5](#_Toc8063590)

[CHƯƠNG 1: TÌM HIỂU VỀ ANDROID VÀ MYSQL 8](#_Toc8063591)

[1.1. Giới thiệu về hệ điều hành Android. 8](#_Toc8063592)

[1.1.1. Lịch sử ra đời hệ điều hành Android. 8](#_Toc8063593)

[1.1.2. Nền tảng Android. 9](#_Toc8063594)

[1.1.3. Kiến trúc hệ điều hành Android. 10](#_Toc8063595)

[1.2. Công cụ xây dựng ứng dụng. 13](#_Toc8063596)

[1.2.1. JDK(Java Deverlopment Kit) 13](#_Toc8063597)

[1.2.2. Android Studio. 14](#_Toc8063598)

[1.3. Các thành phần của một ứng dụng Android. 14](#_Toc8063599)

[1.4. Mô hình MVP (Model View Presenter)trong Android 20](#_Toc8063600)

[1.4.1. Khái niệm mô hình MVP(Model View Presenter) 20](#_Toc8063601)

[1.4.2. Lợi ích của mô hình MVP 21](#_Toc8063602)

[1.4.3. Triển khai MVP trên Android 22](#_Toc8063603)

[1.4.4. So sánh mô hình MVP và mô hình MVC 23](#_Toc8063604)

[1.5. MySQL 23](#_Toc8063605)

[1.5.1 Cơ sở dữ liệu 23](#_Toc8063606)

[1.5.2 Mysql cơ bản 25](#_Toc8063607)

[1.5.3 Kết nối Mysql với Android 28](#_Toc8063608)

[CHƯƠNG 2: KHẢO SÁT PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 29](#_Toc8063609)

[2.1. Khảo sát hệ thống 29](#_Toc8063610)

[2.1.1 Nội dung đề tài 29](#_Toc8063611)

[2.1.2 Đối tượng và phạm vi 29](#_Toc8063612)

[2.2.3Mô tả bài toán 29](#_Toc8063613)

[2.2 Phân tích thiết kế hệ thống 30](#_Toc8063614)

[2.2.1 Mô hình use cases 30](#_Toc8063615)

[2.2.2 Mô hình lớp 44](#_Toc8063616)

[2.2.3 Biểu đồ trình tự 45](#_Toc8063617)

[2.3 Thiết kế cơ sở dữ liệu 53](#_Toc8063618)

[2.3.1 Thiết kế cơ sở dữ liệu dựa trên hệ quản trị cơ sở dữ liệu MySQL 53](#_Toc8063619)

[2.3.2 Thiết kế giao diện 58](#_Toc8063620)

[CHƯƠNG 3: CÀI ĐẶT ỨNG DỤNG 69](#_Toc8063621)

[3.1. Yêu cầu môi trường 69](#_Toc8063622)

[3.2. Kết quả cài đặt chương trình 70](#_Toc8063623)

[3.2.1. Một số giao diện của admin và chủ nhà hàng 70](#_Toc8063624)

[3.2.2. Giao diện của client 76](#_Toc8063625)

[3.3. Ứng dụng công nghệ vào chương trình 81](#_Toc8063626)

[3.4. Đánh giá kết quả thử nghiệm 81](#_Toc8063627)

[KẾT LUẬN 83](#_Toc8063628)

[TÀI LIỆU THAM KHẢO 84](#_Toc8063629)

# DANH MỤC CÁC KÝ HIỆU CÁC CHỮ VIẾT TẮT

MVP -Model View Presenter

SSL - Secure Sockets Layer

# DANH MỤC BẢNG

[Bảng 1.1 So sánh mô hình MVP và mô hình MVC 23](#_Toc6960504)

[Bảng 1.2 Các kiểu dữ liệu 25](#_Toc6960505)

[Bảng 2.1 Luồng dữ liệu chính của ca sử dụng đăng nhập 32](#_Toc6960506)

[Bảng 2.2 Luồng dữ liệu chính ca sử dụng đăng xuất 33](#_Toc6960507)

[Bảng 2.3 Luồng dữ liệu chính ca sử dụng đăng ký tài khoản 34](#_Toc6960508)

[Bảng 2.4 Luồng dữ liệu chính của ca sử dụng đăng ký nhà hàng 35](#_Toc6960509)

[Bảng 2.5 Luồng dữ liệu chính ca sử dụng bính luận 36](#_Toc6960510)

[Bảng 2.6 Luồng dữ liệu chính ca sử dụng tìm kiếm 37](#_Toc6960511)

[Bảng 2.7 Luồng dữ liệu chính ca sử dụng xem chi tiết món ăn 38](#_Toc6960512)

[Bảng 2.8 Luồng dữ liệu chính ca sử dụng liên hệ đặt món 38](#_Toc6960513)

[Bảng 2.9 Luồng dữ liệu chính ca sử dụng chỉ đường 39](#_Toc6960514)

[Bảng 2.10 Luồng dữ liệu chính của cập nhật thông tin nhà hàng 40](#_Toc6960515)

[Bảng 2.11 Luồng dữ liệu chính ca sử dụng quản lý món ăn 41](#_Toc6960516)

[Bảng 2.12 Luồng dữ liệu chính quản lý nhà hàng 42](#_Toc6960517)

[Bảng 2.13 Luồng dữ liệu chính ca sử dụng quản lý tài khoản 42](#_Toc6960518)

[Bảng 2.14 Luồng dữ liệu chính ca sử dụng quản lý món ăn 43](#_Toc6960519)

[Bảng 2.15: Mô tả chi tiết bảng tbl\_restaurant\_comments 53](#_Toc6960520)

[Bảng 2.16: Mô tả chi tiết bảng tbl\_user 54](#_Toc6960521)

[Bảng 2.17: Mô tả chi tiết bảng tbl\_manager\_restaurant 54](#_Toc6960522)

[Bảng 2.18: Mô tả chi tiết bảng tbl\_images 55](#_Toc6960523)

[Bảng 2.19: Mô tả chi tiết bảng tbl\_restaurants 55](#_Toc6960524)

[Bảng 2.20: Mô tả chi tiết bảng tbl\_provinces 56](#_Toc6960525)

[Bảng 2.21: Mô tả chi tiết bảng tbl\_food\_comments 56](#_Toc6960526)

[Bảng 2.22: Mô tả chi tiết bảng tbl\_prices 56](#_Toc6960527)

[Bảng 2.23: Mô tả chi tiết bảng tbl\_foods 57](#_Toc6960528)

[Bảng 2.24: Mô tả chi tiết bảng tbl\_categorys 57](#_Toc6960529)

# DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

[Hình 1. 1 Một số phiên bản của hệ điều hành Android 9](#_Toc8063180)

[Hình 1. 2 Kiến trúc của hệ điều hành Android 11](#_Toc8063181)

[Hình 1. 3 Sơ đồ vòng đời của Service 16](#_Toc8063182)

[Hình 1. 4 Sơ đồ vòng đời của Activity 18](#_Toc8063183)

[Hình 1. 5 Luồng dữ liệu trong mô hình MVP 21](#_Toc8063184)

[Hình 2. 1 Biểu đồ ca sử dụng tổng quát 30](#_Toc8063475)

[Hình 2. 2 Bản vẽ use case cho actor người dùng 31](#_Toc8063476)

[Hình 2. 3 Bản vẽ use case cho Admin 31](#_Toc8063477)

[Hình 2. 4 Bản vẽ use case cho chủ nhà hàng 32](#_Toc8063478)

[Hình 2. 5 Mô hình lớp 44](#_Toc8063479)

[Hình 2. 6 Biểu đồ tuần tự đăng xuất của người dùng 45](#_Toc8063480)

[Hình 2. 7 Biểu đồ tuần tự đăng xuất của chủ nhà hàng 45](#_Toc8063481)

[Hình 2. 8 Biểu đồ đăng nhập người dùng 46](#_Toc8063482)

[Hình 2. 9 Biểu đồ đăng nhập chủ nhà hàng và admin 46](#_Toc8063483)

[Hình 2. 10 Biểu đồ tuần tự đăng ký tài khoản của người dùng 47](#_Toc8063484)

[Hình 2. 11 Biểu đồ đăng ký nhà hàng của chủ nhà hàng 47](#_Toc8063485)

[Hình 2. 12 Biểu đồ tuần tự cập nhật nhà hàng của chủ nhà hàng 48](#_Toc8063486)

[Hình 2. 13 Biểu đồ tuần tự chỉ đường 48](#_Toc8063487)

[Hình 2. 14 Biểu đồ tuần tự xem chi tiết món ăn 49](#_Toc8063488)

[Hình 2. 15 Biểu đồ tuần tự bình luận 49](#_Toc8063489)

[Hình 2. 16 Biểu đồ tuần tự liên hệ đặt món 50](#_Toc8063490)

[Hình 2. 17 Biểu đồ tuần tự tìm kiếm theo tỉnh 50](#_Toc8063491)

[Hình 2. 18 Biểu đồ tuần tự tìm kiếm theo thực đơn 51](#_Toc8063492)

[Hình 2. 19 Biểu đồ tuần tự quản lý nhà hàng của admin 51](#_Toc8063493)

[Hình 2. 20 Biểu đồ tuần tự quản lý tài khoản của admin 52](#_Toc8063494)

[Hình 2. 21 Biểu đồ quản lý thực đơn của admin 52](#_Toc8063495)

[Hình 2. 22 Biểu đồ tuần tự quản lý món ăn 53](#_Toc8063496)

[Hình 2. 23 Thiết kế giao diện Flash của app ShareFoodManager 58](#_Toc8063497)

[Hình 2. 24 Thiết kế màn hình đăng nhập củ chủ nhà hàng và admin 58](#_Toc8063498)

[Hình 2. 25 Thiết kế màn hình quản lý nhà hàng 59](#_Toc8063499)

[Hình 2. 26 Thiết kế màn hình quản lý tài khoản 59](#_Toc8063500)

[Hình 2. 27 Thiết kế màn hình quản lý thực đơn 60](#_Toc8063501)

[Hình 2. 28 Thiết kế màn hình chi tiết tài khoản 60](#_Toc8063502)

[Hình 2. 29 Thiết kế giao diện cập nhật đơn hàng 61](#_Toc8063503)

[Hình 2. 30 Thiết kế giao diện thêm thực đơn 61](#_Toc8063504)

[Hình 2. 31 Thiết kế giao diện quản lý nhà hàng 62](#_Toc8063505)

[Hình 2. 32 Thiết kế giao diện quản lý món ăn 62](#_Toc8063506)

[Hình 2. 33 Thiết kế giao diện cập nhật nhà hàng 63](#_Toc8063507)

[Hình 2. 34 Thiết kế giao diện cập nhật món ăn 63](#_Toc8063508)

[Hình 2. 35 Thiết kế giao diện thêm món ăn 64](#_Toc8063509)

[Hình 2. 36 Thiết kế giao diện Flash của app ShareFood 64](#_Toc8063510)

[Hình 2. 37 Thiết kế giao diện đăng nhập của người dùng 65](#_Toc8063511)

[Hình 2. 38 Thiết kế giao diện menu 65](#_Toc8063512)

[Hình 2. 39 Thiết kế giao diện đăng ký nhà hàng 66](#_Toc8063513)

[Hình 2. 40 Thiết kế giao diện màn hình chính 66](#_Toc8063514)

[Hình 2. 41 Thiết kế giao diện thông tin món ăn 67](#_Toc8063515)

[Hình 2. 42 Thiết kế giao diện tìm kiếm theo thực đơn 67](#_Toc8063516)

[Hình 2. 43 Thiết kế giao diện tìm kiếm theo tỉnh 68](#_Toc8063517)

[Hình 3. 1 Giao diện khởi động 70](#_Toc8063518)

[Hình 3. 2 Giao diện đăng nhập 71](#_Toc8063519)

[Hình 3. 3 Giao diện cập nhật nhà hàng 71](#_Toc8063520)

[Hình 3. 4 Giao diện quản lý món ăn 72](#_Toc8063521)

[Hình 3. 5 Giao diện cập nhật nhà hàng 72](#_Toc8063522)

[Hình 3. 6 Giao diện quản lý nhà hàng 73](#_Toc8063523)

[Hình 3. 7 Giao diện quản lý tài khoản 73](#_Toc8063524)

[Hình 3. 8 Giao diện quản lý thực đơn 74](#_Toc8063525)

[Hình 3. 9 Giao diện cập nhật thực đơn 74](#_Toc8063526)

[Hình 3. 10 Giao diện thêm thực đơn 75](#_Toc8063527)

[Hình 3. 11 Giao diện đăng xuất 75](#_Toc8063528)

[Hình 3. 12 Giao diện màn hình khởi động 76](#_Toc8063529)

[Hình 3. 13 Giao diện danh sách món ăn 76](#_Toc8063530)

[Hình 3. 14 Giao diện menu 77](#_Toc8063531)

[Hình 3. 15 Giao diện đăng nhập 77](#_Toc8063532)

[Hình 3. 16 Giao diện đăng ký 78](#_Toc8063533)

[Hình 3. 17 Giao diện tìm theo thực đơn 78](#_Toc8063534)

[Hình 3. 18 Giao diện món ăn theo tỉnh 79](#_Toc8063535)

[Hình 3. 19 Giao diện yêu thích 79](#_Toc8063536)

[Hình 3. 20 Giao diện chi tiết món ăn 80](#_Toc8063537)

[Hình 3. 21 Giao diện chỉ đường bình luận 80](#_Toc8063538)

[Hình 3. 22 Giao diện đăng ký nhà hàng 81](#_Toc8063539)

# CHƯƠNG 1: TÌM HIỂU VỀ ANDROID VÀ MYSQL

* 1. **Giới thiệu về hệ điều hành Android.**
     1. ***Lịch sử ra đời hệ điều hành Android.***

**Hệ điều hành 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 Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005.

Chính mã nguồn mở của Android cùng với tính không ràng buộc nhiều đã cho phép các nhà phát triển thiết bị di động và các lập trình viên 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.

Cho đến hiện tại, Google đã bán ra 200.000 thiết bị Android mỗi ngày và Play Store của họ vượt mức 90.000 ứng dụng. Nhờ yếu tố mở, dễ dàng tinh chỉnh cùng sự phát triển nhanh chóng đã khiến hệ điều hành này dần trở nên phổ biến, kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng nhưng giờ đây Android đã xuất hiện trên các smart TV, máy chơi game và một số thiết bị điện tử khác.

**Kỷ nguyên của Android:**

Android bắt đầu với bản beta đầu tiên vào tháng 11 năm 2007 và phiên bản thương mại đầu tiên, Android 1.0, được phát hành vào tháng 9 năm 2008. Kể từ tháng 4 năm 2009, phiên bản Android được phát triển, đặt tên theo chủ đề bánh kẹo và phát hành theo thứ tự bảng chữ cái: Cupcake (bánh cupcake), Donut (bánh vòng), Eclair (bánh Eclair), Froyo (ya-ua lạnh), Gingerbread (bánh gừng), Honeycomb (mật ong), Ice Cream Sandwich (bánh kẹp kem), Jelly Bean (kẹo dẻo), Kitkat (bánh Kitkat), Lollipop (kẹo mút) và một số phiên bản khác cho đến tận bây giờ.



Hình 1. Một số phiên bản của hệ điều hành Android

* + 1. ***Nền tảng Android.***

**Android chiếm hơn 81% thị phần điện thoại kích hoạt toàn cầu:**

Với sự tăng trưởng mạnh mẽ của thị trường này, không chỉ các hãng sản xuất di động mà cả các nhà phân phối và những doanh nghiệp công nghệ khác cũng chen chân vào cuộc đua phát triển ứng dụng mobile để khai thác hết các tính năng của thiết bị di động đỉnh cao như: iPhone, iPad, Samsung Galaxy, Nokia Lumia, Blackberry, HTC,… Việt Nam cũng không nằm ngoài xu hướng này. Điều đó đặt ra một yêu cầu là phải có đội ngũ nhân sự lập trình giỏi để cho ra đời các tiện ích độc đáo và thiết thực, đáp ứng đòi hỏi ngày càng cao của người dùng.

**Công cụ trở thành lập trình viên:**

Lập trình Android sẽ đem lại cho chúng ta những công cụ cần thiết, những kiến thức cơ bản giúp chúng ta tiến gần hơn tới vị trí của một lập trình viên, một nhà phát triển chuyên nghiệp, hoặc chỉ đơn giản là thỏa sức sáng tạo ra những ứng dụng hữu ích cho cuộc sống.

**Kho ứng dụng Google Play Store linh hoạt.** Các ứng dụng của bạn khi phát triển xong có thể tải lên kho Google Play và cho phép tải về chỉ sau vài giờ. Ngoài ra, một ứng dụng trên Play Store đều được phản hồi nhanh chóng và xử lý linh hoạt bởi nó có thể cập nhật nhiều lần trong một ngày, mọi phản hồi về lỗi, tính tương thích…

Một ưu điểm khác của Play Store là cho phép lập trình viên phát hành ứng dụng beta để có thể nhận được các phản hồi sớm từ nhóm thử nghiệm. Điều này giúp cho bản chính thức sẽ hoàn chỉnh hơn khi người dùng tải về.

**Hỗ trợ ngôn ngữ lập trình Java:**

Hiện nay, Java là ngôn ngữ lập trình phổ biến và mạnh mẽ, sự phổ biến của ngôn ngữ này trải rộng từ mọi thiết bị đến các hệ điều hành. Do đó, nếu bạn thành thạo ngôn ngữ Java có thể học lập trình android để có cơ hội phát triển được những ứng dụng cho Android và các nền tảng khác nữa.

**Khả năng chuyển đổi ứng dụng linh hoạt.**

Một trong những tiêu chí quan trọng để các lập trình viên quyết định chọn nền tảng nào cho sự nghiệp phát triển phần mềm, ứng dụng của mình đó là nền tảng đó có khả năng chuyển đổi ứng dụng từ nền tảng này sang nền tảng khác hay không.

Các lập trình viên Android sử dụng ngôn ngữ lập trình Java, do đó có thể dễ dàng chuyển đổi sang các nền tảng di động khác như Symbian, BlackBerry và Ubuntu. Bên cạnh đó, các ứng dụng Android cũng có thể chuyển đổi để trở thành phần mềm trên Chrome OS.

* + 1. ***Kiến trúc hệ điều hành Android.***

Hệ điều hành Android là 1 ngăn xếp các thành phần phần mềm, được chia thành 5 phần và 4 lớp chính như trong hình bên dưới.



Hình 1. Kiến trúc của hệ điều hành Android

**Linux Kernel**:

Dưới cùng là lớp Linux - Linux 3.6 cùng với khoảng 115 bản vá. Lớp này cung cấp 1 cấp độ trừu tượng giữa phần cứng của thiết bị và các thành trình điều khiển phần cứng thiết yếu như máy ảnh, bàn phím, màn hình hiển thị... Đồng thời, hạt nhân (kernel) còn xử lý tất cả các thứ mà Linux có thể làm tốt như mạng kết nối và 1 chuỗi các trình điều khiển thiết bị, giúp cho giao tiếp với các thiết bị ngoại vi dễ dàng hơn.

Các thư viện:

Ở trên lớp nhân Linux là tập các thư viện bao gồm WebKit - trình duyệt Web mã nguồn mở, được biết đến như thư viện libc, cơ sở dữ liệu SQLite - hữu dụng cho việc lưu trữ và chia sẻ dữ liệu ứng dụng, các thư viện chơi và ghi âm audio, video, hay các thư viện SSL(Secure Sockets Layer) chịu trách nhiệm bảo mật internet...

**Các thư viện Android**:

Đây là các thư viện dựa trên Java phục vụ cho việc phát triển Android. Ví dụ của các thư viện này bao gồm các thư viện ứng dụng dùng để xây dựng giao diện người dùng, vẽ đồ họa hay truy cập cơ sở dữ liệu. 1 số thư viện chính của Android:

* Android.app - Cung cấp quyền truy cập vào ứng dụng và là nền tảng của tất cả ứng dụng Android.
* Android.content - Cung cấp quyền truy cập nội dung (content), truyền tải thông điệp giữa các ứng dụng hay các thành phần của ứng dụng.
* Android.database - Đựoc sử dụng để truy cập dữ liệu của content provider và cơ sở dữ liệu SQLite
* Android.opengl - giao diện các phuơng thức Java để sử dụng OpenGL ES
* Android.os - Cung cấp các ứng dụng với quyền truy cập vào các dịch vụ của hệ điều hành bao gồm thông điệp, các dịch vụ hệ thống và các giao tiếp nội tại
* Android.text - Đựoc sử dụng để hiển thị và điều chỉnh chữ trên màn hình thiết bị
* Android.view - Các thành phần cơ bản trong việc xây dựng giao diện người dùng của ứng dụng.
* Android.widget - Tập các thành phần giao diện người dùng đã được xây dựng sẵn như các nút, các nhãn (label), list view,....
* Android.webkit - Tập các lớp cho phép xây dựng khả năng duyệt web.

Android Runtime:Đây là phần thứ 3 của kiến trúc và nằm ở lớp thứ 2 từ dưới lên. Phần này cung cấp 1 bộ phận quan trọng là Dalvik Virtual Machine - là 1 loại Java Virtual Machine được thiết kế đặc biệt để tối ưu cho Android.

Dalvik VM sử dụng các đặc trưng của nhân Linux như quản lý bộ nhớ và đa luồng, những thứ mà đã có sẵn trong Java. Dalvik VM giúp mở ứng dụng Android chạy trong tiến trình riêng của nó, với các thể hiện (instance) riêng của Dalvik virtual Machine.

Android Runtime cũng cung cấp 1 tập các thư viện chính giúp các nhà phát triển ứng dụng Android có thể viết ứng dụng Android bằng Java

**Application Framework**:

Lớp Android Framework cung cấp các dịch vụ cấp độ cao hơn cho các ứng dụng dưới dạng các lớp Java. Các nhà phát triển ứng dụng được phép sử dụng các dịch vụ này trong ứng dụng của họ.

Android Framework bao gồm các dịch vụ chính sau:

* Activitty Manager - Kiểm soát tất cả khía cạnh của vòng đời ứng dụng và ngăn xếp các Activity.
* Content Providers - Cho phép các ứng dụng chia sẻ dữ liệu với các ứng dụng khác.
* Resource Manager - Cung cấp quyền truy cập vào các tài nguyên như các chuỗi, màu sắc, các layout giao diện người dùng...
* Notifications Manager - Cho phép các ứng dụng hiển thị cảnh báo và các thông báo cho người dùng.
* View System - Tập các thành phần giao diện (view) được sử dụng để tạo giao diện người dùng.

**Application**:

Lớp trên cùng của kiến trúc là Application. Các ứng dụng bạn tạo ra sẽ được cài đặt trên lớp này. Ví dụ như: Danh bạ, nhắn tin, trò chơi...

Mặc dù bài viết không liên quan trực tiếp đến việc lập trình nhưng giúp bạn hiểu sâu hơn về chính những ứng dụng mình làm ra cũng như các ứng dụng Android khác, qua đó sẽ giúp ích rất nhiều cho việc phát triển ứng dụng sau này.

* 1. **Công cụ xây dựng ứng dụng.**
     1. ***JDK(Java Deverlopment Kit)***

JDK là một bộ công cụ phát triển Java, nó dành cho những người lập trình Java để phát triển ứng dụng. Về cơ bản nó bao gồm:

* JRE (Java Runtime Environment) là một môi trường chạy ứng dụng Java.
* Javac: Một chương trình để dịch mã mà bạn viết thành mã bytecode, khi ứng dụng Java chạy nó dịch mã bytecode thành mã máy tính và thực thi, điều đó có nghĩa là bytecode chỉ là một mã trung gian.
* Archive (jar): Là một chương trình nén các file thành một file duy nhất có đuôi jar. Thường dùng để đóng gói các file class.
* Javadoc: Là một công cụ tạo ra tài liệu hướng dẫn sử dụng API.
  + 1. ***Android Studio.***

#### Android Studio là môi trường phát triển tích hợp (IDE) chính thức dành cho phát triển các ứng dụng chạy trên nền tảng Android.

Android Studio hoạt động dựa trên “IntelliJ IDEA Community Edition” cho phép lập trình viên tạo ứng dụng, dễ dàng thực hiện các thay đổi và xem trước trong thời gian thực, đồng thời cũng có khả năng tăng tốc sản phẩm, thiết kế giao diện đẹp hơn trước. Đặc biệt là tiếng Việt cũng được hỗ trợ trong **Android Studio.**

[Android](https://trainghiemso.vn/android/)**Studio** hỗ trợ một loạt các giả lập để xem trước ứng dụng, vì vậy ngay cả khi bạn không có thiết bị thử nghiệm, bạn vẫn có thể chắc chắn rằng mọi thứ đều hoạt động trơn tru. Bên cạnh đó, loạt công cụ như lời khuyên tối ưu hóa, đồ thị doanh số bán hàng, và số liệu lấy từ phân tích sẽ giúp các nhà phát triển quản lý ứng dụng đang bán của mình và tìm ra hướng đi cụ thể với từng thiết bị Android.

**Android Studio** hỗ trợ các hệ điều hành [Windows](https://trainghiemso.vn/windows/), Mac OS X và Linux, và là IDE chính thức của Google để phát triển ứng dụng Android gốc để thay thế cho Android Development Tools (ADT) dựa trên Eclipse.

* 1. **Các thành phần của một ứng dụng Android.**

**Services:**

Chạy ngầm bên dưới hay không cần hiển thị trên giao diện vì Service có chức năng giúp ứng dụng vẫn chạy được. Ví dụ khi các bạn đang nghe nhạc bằng ứng dụng nghe nhạc nhưng các bạn vẫn nghe được kể cả khi đã tắt ứng dụng đó là vì ứng dụng đó đang chạy dưới background.  Không chỉ thế các bạn còn có thể liên kết hoặc kết nối giữa một Activity với một service, ví dụ: việc download thực hiện ở service khi download một file từ trên mạng, kết quả phần trăm download sẽ trả sau đó lên activity để hiện thị cho người dùng biết.

Chú ý: Cần phân biệt giữa service và thread, service không phải thread mặc dù service chạy ở chế độ background, để tránh trường hợp sử dụng service làm ứng dụng bị đơ hoặc chậm khi sử lý các luồng dữ liệu hoặc giao diện khác thì chúng ta sử dụng và xử lý cho phù hợp tùy trường hợp.

Khi sử dụng service chúng ta sẽ kế thừa từ lớp cha là: Service

Một service là một thành phần của ứng dụng, thể hiện mong muốn ứng  dụng thực hiện các hành động trong khi không tương tác với người dùng hoặc cung cấp chức năng cho các ứng dụng khác sử dụng. Nói một cách đơn giản, service là các tác vụ (task) chạy ngầm dưới hệ thống nhằm thực hiện một nhiệm vụ nào đó.  Mỗi class Service phải chứa thẻ <service> được khai báo trong file AndroidManifext.xml. Services có thể được bắt đầu bởi Context.startService() và Context.bindservice()

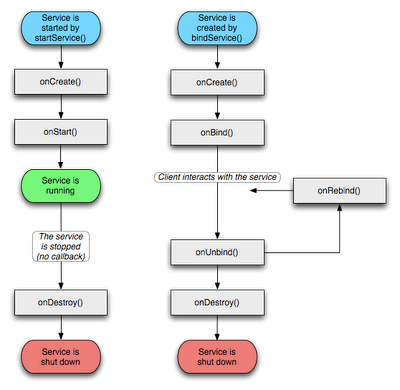
Cũng như các đối tượng của ứng dụng khác, services chạy trên luồng (Thread) của tiến trình chính. Có nghĩa là nếu service dự định chạy các hành vi có cường độ lớn hoặc các hành vi có thể gây nghẽn mạch, nó sẽ tự sinh ra luồng (Thread) của chính nó để làm thực hiện các tác vụ được đặc tả trong Service.

Service có chứa 2 đặc trưng cơ bản sau:

* Là một chức năng để ứng dụng thông báo với hệ thống về việc mà nó muốn thực hiện ở phía dưới nền (ngay cả khi người dùng không tương tác trực tiếp tới ứng dụng). Điều này tương ứng với việc gọi phương thức Context.startservice(), nó sẽ yêu cầu hệ thống lên lịch cho service để chạy cho đến khi bị chính service hay người nào đó dừng nó lại.
* Là một chức năng cho phép ứng dụng để cung cấp các chức năng của nó cho các ứng dụng khác. Điều đó tương ứng với việc gọi Context.bindService(), cho phép một long-standing connection được tạo ra để service có thể tương tác với nó.

Khi một Service được tạo ra, việc hệ thống cần làm là tạo ra thành phần và gọi hàm onCreate() và bất kỳ lệnh callback thích hợp nào trên luồng chính. Nó phụ thuộc vào Service thi hành hành vi thích hợp nào, chẳng hạn tạo luồng thứ hai để thực hiện tiến trình.

Có 2 lý do để một service được chạy bởi hệ thống. Bằng cách gọi phương thức Context.startService() và Context.bindService(). Nếu người dùng gọi phương thức Context.startService() thì hệ thống sẽ nhận về một đối tượng service (tạo mới nó và gọi hàm onCreate() nếu cần thiết) và sau đó lại gọi phương thức onStartCommand(Intent, int ,int) với các biến được cung cấp bởi client. Service mới được tạo ra này sẽ chạy cho tới khi có lệnh gọi phương thức Context.stopService() hoặc stopSelf(). Dù cho một service có được start bao nhiêu lần đi nữa thì nó sẽ bị tắt ngay khi phương thức Context.stopService() hoặc phương thức stopSelf() được gọi.Service có thể dùng phương thức stopSelf(int) để đảm bảo rằng service sẽ không được tắt cho đến khi Intent đã khởi tạo nó được xử lý.



Hình 1. 3 Sơ đồ vòng đời của Service

**Activities:**

Activity là một màn hình đơn với giao diện trên đó. Và điểm tiếp xúc với người dùng cũng được xem như là một Activity. Bạn sẽ thấy giao diện trên màn hình hiển thị cho chúng ta xem khi đã cài và chạy ứng dụng này.  Activty thực hiện các chức năng cần thiết trên đó, chuyển đổi qua lại giữa các màn hình giao diện/ chức năng, giúp người dùng tương tác với hệ thống.

Chúng ta sẽ kết thừa từ lớp cha của nó là Activity khi thường sử dụng Activity.

Là thành phần tối quan trọng của bất kỳ một ứng dụng Android nào. Thuật ngữ Activity chỉ một việc mà người dùng có thể thực hiện trong một ứng dụng Android. Do gần như mọi activity đều tương tác với người dùng, lớp Activity đảm nhận việc tạo ra một cửa sổ (window) để người lập trình đặt lên đó một giao diện UI với setContentView(View). Một activity có thể mang nhiều dạng khác nhau: Một cửa sổ toàn màn hình (full screen window), một cửa sổ floating (với windowsIsFloating) hay nằm lồng bên trong 1 activity khác (với ActivityGroup).

Có 2 phương thức mà gần như mọi lớp con của Activity đều phải hiện thực:

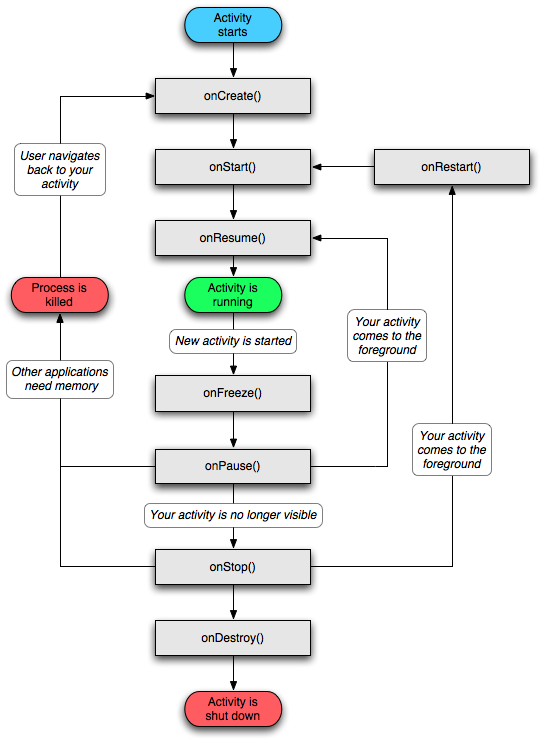
* onCreate(Bundle) - Nơi khởi tạo activity. Quan trọng hơn, đây chính người lập trình gọi setContentView(int) kèm theo layout để thể hiện UI của riêng mình. Đồng thời còn có findViewById(int) giúp gọi các widget (buttons, text boxes, labels,..) để dùng trong UI.
* onPause() - Nơi giải quyết sự kiện người dùng rời khỏi activity. Mọi dữ liệu được người dùng tạo ra tới thời điểm này cần phải được lưu vào ContentProvider.

Để có thể sử dụng Context.startActivity(), mọi lớp activity đều phải được khai báo với tag <activity> trong file AndroidManifest.xml.

Các activity được quản lí dưới dạng các activity stack - First-In-Last-Out: Khi một activity mới được khởi tạo, nó sẽ được đưa lên trên cùng stack, các activity khác muốn chạy trên nền (foreground) trở lại thì cần phải chờ tới khi Activity mới này kết thúc.

Một Activity có 4 trạng thái:

* Active hay Running: Khi một activity đang chạy trên màn hình.
* Paused: Khi một activity vẫn đang chạy trên màn hình nhưng đang bị một activity trong suốt (transparent) hay không chiếm toàn màn hình hiển thị phía trên. Tuy vẫn lưu trữ dữ liệu, nhưng các paused activity này sẽ bị hệ thống bắt chấm dứt khi đang thiếu bộ nhớ trầm trọng.
* Stopped: Khi một activity bị che khuất hoàn toàn bởi một activity khác. Tuy vẫn lưu trữ dữ liệu, nhưng các stopped activity này sẽ thường xuyên bị hệ thống bắt chấm dứt để dành chỗ cho các tiến trình khác.
* Killed hay Shut down: Khi một activity đang paused hay stopped, hệ thống sẽ xóa activity ấy ra khỏi bộ nhớ.



Hình 1. 4 Sơ đồ vòng đời của Activity

Dựa vào lược đồ trên, thấy được có 3 vòng lặp quan trọng sau:

* Vòng đời toàn diện (Entire Lifetime): Diễn ra từ lần gọi onCreate(Bundle) đầu tiên và kéo dài tới lần gọi onDestroy() cuối cùng.
* Vòng đời thấy được (Visible Lifetime): Diễn ra từ khi gọi onStart() và kéo dài tới khi gọi onStop(). Ở vòng đời này, activity được hiển thị trên màn hình nhưng có thế không tương tác với người dùng ở trên nền. Các phương thức onStart() và onStop() có thể được gọi nhiều lần.
* Vòng đời trên nền (Foreground Lifetime): Diễn ra từ khi gọi onResume() và kéo dài tới khi gọi onPause(). Ở vòng đời này, activity nằm trên mọi activity khác và tương tác được với người dùng. Một activity có thể liên tục thay đổi giữa 2 trạng thái paused và resumed, chẳng hạn khi thiết bị sleep hay một intent mới được đưa tới.

**Broadcast receivers:**

Trong nhiều trường hợp Broadcast receiver được sử dụng, ví dụ: Ngoài sử dụng binding các bạn có thể sử dụng broadcast để gửi dữ liệu, có thể chuyển dữ liệu từ service lên. Hoặc để tạo ra notification trên màn hình để báo cho người dùng biết thì ví dụ như trong ứng dụng như hẹn giờ, khi đến giờ hẹn, ứng dụng sẽ sử dụng broadcast báo thức.

Chúng ta kế thừa từ BroadcastReceiver khi sử dụng broadcast receiver.

Có 2 cách để đăng ký Android broadcast receiver

* Một là phương thức tĩnh khi đăng ký broadcast receiver trong ứng dụng thông qua file AndroidManifest.xml.
* Hai là phương thức động để đăng ký broadcast receiver thông qua việc sử dụng method Context.registerReceiver(). Các broadcast receiver được đăng ký động có thể được gỡ bởi methodunregisterReceiver().

Tránh những task chạy lâu

Chúng ta cần tránh những task chạy lâu trong các Broadcast receiver. Đối với các task lâu chúng ta có thể chạy service trong receiver. Các receiver được đăng ký động sẽ được gọi trong UI thread. Các receiver được đăng ký động sẽ hạn chế bất cứ thao tác UI (giao diện người dùng), vì vậy phương thức onReceive() sẽ được chạy ở tốc độ cao nhất. Application có thể trở nên chậm chạp và phát sinh lỗi tồi tệ “Application Not Responding”.

**Content providers:**

Để ứng dụng có thể lưu trữ dữ liệu trên hệ thống thì Content provider là thành phần dùng để quản lí các cách lưu trữ dữ liệu. Khi xây dựng các ứng dụng cần lưu trữ vào SQLite chúng ta sẽ biết cụ thể về thành phần này. Ví dụ, trong các ứng dụng từ điển các bạn sẽ thấy dữ liệu, từ vựng mà các bạn tra hiện thị, thì dữ liệu hiện thị đó được lưu trữ trong SQLite và Content provider gọi để lấy ra cho người dùng xem. Ngoài ra thành phần này còn thực hiện các chức năng sửa, xóa dữ liệu, thêm,…

Android cung cấp sẵn content providers cho một số kiểu dữ liệu thông dụng như âm thanh, video, thông tin sổ điện thoại,... Người lập trình cũng có thể tự tạo ra các class con (subclass) của Content Provider để lưu trữ kiểu dữ liệu của riêng mình.

Cách thức hoạt động của Content Provider:

Mọi content provider, tuy có thể khác nhau về cách thức lưu trữ dữ liệu, đều sử dụng chung một interface để tìm và trả về dữ liệu, bao gồm việc thêm, bớt và sửa dữ liệu. Việc này được thực hiện thông qua các đối tượng ContentResolver khi gọi getContentResolver()

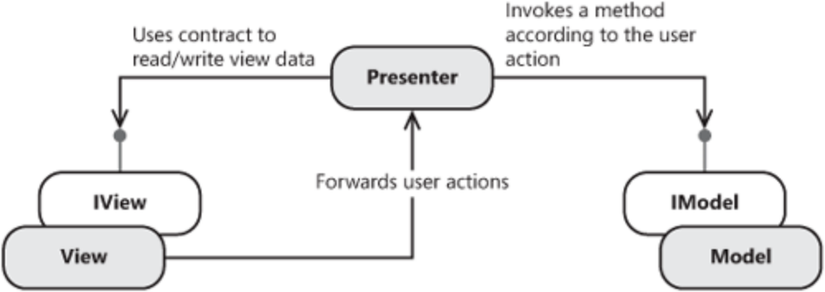
ContentResolver cr = getContentResolver();

* 1. **Mô hình MVP (Model View Presenter)trong Android**
     1. ***Khái niệm mô hình MVP(Model View Presenter)***

MVP là một mô hình kiến trúc hướng giao diện người dùng, được thiết kế để tạo thuận lợi cho việc kiểm thử đơn vị (unit testing) và tăng tính tách biệt giữa tầng dữ liệu và tầng hiển thị dữ liệu trong mô hình MVC.

Mô hình MVP cho phép tách tầng trình diễn Presenter ra khỏi tầng dữ liệu Model, vì vậy tương tác với giao diện được tách biệt với cách chúng ta biểu diễn nó trên màn hình View, hay nói cách khác, tất cả logic khi người dùng tương tác được tách ra và đưa vào tầng trình diễn. Thiết kế lý tưởng nhất là với cùng một logic được áp dụng cho nhiều View khác nhau và hoán đổi được cho nhau.

* Model là một interface xác định cách mà dữ liệu được hiển thị trong giao diện người dùng.
* View là một giao diện người dùng thụ động hiển thị dữ liệu (Model) và tiếp nhận tương tác người dùng và truyền tới để Presenter xử lý tương tác.
* Presenter được ví như trung gian. Khi người dùng tương tác với View, Presenter tiếp nhận tương tác người dùng và cập nhật Model. Khi Model được cập nhật hay có thay đổi, Presenter lấy dữ liệu từ Model, định dạng và đưa tới View để hiển thị.



Hình 1. Luồng dữ liệu trong mô hình MVP

* + 1. ***Lợi ích của mô hình MVP***

Trong Android, có một vấn đề phát sinh từ thực tế là các cơ chế xử lý tương tác trong Android được kết hợp chặt chẽ giữa giao diện người dùng và xử lý, truy cập dữ liệu.

Việc kết hợp này có thể giúp giảm thiểu lượng code trong ứng dụng cũng như gộp các xử lý, logic trong ứng dụng vào cùng một nơi. Nhưng đối với một ứng dụng liên tục phát triển, hay một ứng dụng lớn, việc này khiến cho lượng code trên mỗi logic trở nên rất lớn, các logic xen lần, chồng chéo lên nhau, rất khó cho việc đọc hiểu, bảo trì cũng như mở rộng.

Phân tầng, phân lớp ứng dụng có thể giảm đi hiệu năng do ứng dụng cần thêm nhiều tài nguyên, nhưng tính khả chuyển của ứng dụng cũng tăng gấp nhiều lần, không những thế còn mở rộng khả năng cho phép kiểm thử từng phần của ứng dụng trở nên dễ dàng hơn. Chính vì thế cân nhắc giữa hiệu năng và tính dễ dàng mở rộng, bảo trì là hết sức quan trọng.

Trong Android, kiểm thử là một vấn đề khó khăn vì mối liên kết chặt chẽ giữa giao diện, logic và dữ liệu. MVP tách biệt giao diện khỏi dữ liệu, chia ứng dụng ra thành ít nhất ba lớp khác nhau, từ đó có thể kiểm thử một các độc lập. Với MVP ta có thể kiểm thử đo đạc một cách tối đa các logic của ứng dụng giúp cho dev có thể quản lý code một cách dễ dàng.

* + 1. ***Triển khai MVP trên Android***

Có rất nhiều biến thể cũng như phương pháp triển khai MVP, tất cả mọi người có thể điều chỉnh mô hình này tùy theo nhu cầu và cách họ cảm thấy thoải mái hơn. Các mô hình này, về cơ bản khác nhau ở số lượng chức năng mà tầng Presenter đảm nhận.

**Tầng trình diễn – Presenter**:

Tầng trình diễn có trách nhiệm như một middle- man giữa  View và Model. Nó lấy dữ liệu từ Model, định dạng và trả về cho View. Nhưng không giống như MVC, nó cũng quyết định những gì sẽ xảy ra khi người dùng tương tác với View, hay nói cách khác nó hàm chứa logic ứng dụng.

**Tầng logic dữ liệu – Model**:

Trong một ứng dụng với thiết kế kiến trúc tốt, mô hình này sẽ chỉ là một gateway giữa tầng domainvà tầng business logic. Trong mô hình Clean Architecture của Uncle Bob, Model sẽ là một interactor thự thi một use case. Để đơn giản, ở đây Model đơn thuần được nhìn nhận như một data source - cung cấp dữ liệu chúng ta muốn hiển thị trong giao diện ứng dụng.

Tầng giao diện – View:

View, thường được implement bởi một Activity (hoặc có thể là một Fragment, một View ... tùy thuộc vào cấu trúc ứng dụng), Activity này sẽ chứa một thuộc tính là một lớp Presenter. Lý tưởng nhất Presenter nên được cung cấp bởi một Dependency Injection framewok như Dagger, nhưng trong trường hợp ứng dụng không sử dụng một thư viện hay framework như vậy ta hoàn toàn có thể tạo ra các đối tượng Presenter này.

* + 1. ***So sánh mô hình MVP và mô hình MVC***

Bảng 1. So sánh mô hình MVP và mô hình MVC

|  |  |
| --- | --- |
| MVP(Model, View, Presenter) | MVC(Model, View, Control) |
| - Tầng view là tầng duy nhất nhận tương tác.  - Logic xử lý dữ liệu tương tác cũng như xử lý dữ liệu hiển thị được tách ra trong tầng presenter.  - Việc kết hợp này giảm thiểu lượng code trong ứng dụng cũng như gộp các logic trong ứng dụng vào một nơi.  - Nhưng đối với một ứng dụng liên tục phát triển hay một ứng dụng lớn việc này khiến lượng code trên mỗi logic là rất lớn, các logic chồng chéo lên nhau rất khó cho việc đọc hiểu cũng như việc bảo trì.  - Dễ dàng trong việc kiểm thử một cách độc lập. | - Cả 2 tầng view và control đều có thể tương tác được với người dùng.  - Logic xử lý dữ liệu tương tác và xử lý dữ liệu hiển thị được gộp chung vào trong controler.  - Việc gộp chung vào một controler sẽ khó kiểm soát lượng code cũng như gây khó khăn trong việc nâng cấp và cập nhật.  - Việc kiểm thử ở mô hình này chiếm thời gian cũng như công sức nhiều hơn so với mô hình MVP. |

* 1. **MySQL**

### *1.5.1 Cơ sở dữ liệu*

MySQL là hệ thống cơ sở dữ liệu phổ biến nhất được sử dụng với PHP.

* MySQL là một hệ thống cơ sở dữ liệu được sử dụng trên web
* MySQL là một hệ thống cơ sở dữ liệu chạy trên máy chủ
* MySQL là lý tưởng cho cả ứng dụng nhỏ và lớn
* MySQL rất nhanh, đáng tin cậy và dễ sử dụng
* MySQL sử dụng SQL chuẩn
* MySQL biên dịch trên một số nền tảng
* MySQL miễn phí tải về và sử dụng
* MySQL được phát triển, phân phối và hỗ trợ bởi Tập đoàn Oracle
* MySQL được đặt theo tên con gái của người đồng sáng lập Monty Widenius: My.

Dữ liệu trong cơ sở dữ liệu MySQL được lưu trữ trong các bảng. Bảng là tập hợp các dữ liệu liên quan và nó bao gồm các cột và hàng.

Cơ sở dữ liệu rất hữu ích để lưu trữ thông tin một cách phân loại. Một công ty có thể có một cơ sở dữ liệu với các bảng sau:

Cơ sở dữ liệu rất hữu ích để lưu trữ thông tin một cách phân loại. Một công ty có thể có một cơ sở dữ liệu với các bảng sau:

* Nhân viên
* Các sản phẩm
* Khách hàng
* Đơn đặt hàng

Để làm việc với cơ sở dữ liệu chúng ta cần khởi động Apache và Cơ sở dữ liệu lên. Chúng ta đang sử dụng Xampp vì thế chúng ta sẽ khởi động chương trình MySQL trong Xampp Controll Panel Application. Tiếp đến bạn mở trình duyệt lên và truy cập vào hệ thống quản trị cơ sở dữ liệu thông qua  địa chỉ : <http://localhost/phpmyadmin>.

Trong cơ sở dữ liệu thì chưa rất nhiều thông tin có thể là thông tin cùng loại hoặc thông tin khác loại. Để có thể quản lý được toàn bộ các thông tin này trong cơ sở dữ liệu một cách khoa học và hiệu quả, chúng ta sử dụng một bảng dữ liệu dùng để quản lý những thông tin cùng loại hay có cùng kiểu nội dung.

Mỗi một bảng dữ liệu lại chứa các cột dữ liệu hay còn gọi là các trường dữ liệu khác nhau. Một cột sẽ lưu trữ một thông tin trong bảng.

Thuật ngữ trong mysql:

* NULL : Giá trị của trường cho phép rỗng
* AUTO\_INCREMENT : Cho phép giá trị tự động tăng, thường sử dụng cho trường ID
* UNSIGNED: Giá trị phải là số nguyên dương
* PRIMARY KEY: Cho phép nó là khóa chính trong bảng.

Các kiểu dữ liệu:

Bảng 1. Các kiểu dữ liệu

|  |  |
| --- | --- |
| **Kiểu dữ liệu** | **Mô tả** |
| **Char** | Định dạng chuỗi có chiều dài từ 0 đến 255 |
| **Varchar** | Định dạng chuỗi có chiều dài từ 0 đến 255, chuỗi dữ liệu ký tự unicode |
| **Text** | Định dạng chuỗi có chiều dài từ 0 đến 65535 |
| **Longtext** | Định dạng chuỗi có chiều dài từ 0 đến 4294967215 |
| **Int** | Định dạng số có chiều dài từ 0 đến 4294967215 |
| **Float** | Định dạng số thập phân có chiều dài nhỏ |
| **Double** | Định dạng số thập phân có chiều dài lớn |
| **Date** | Định dạng thời gian theo định dạng YYYY-MM-DD |
| **Date Time** | Định dạng thời gian theo định dạng YYYY-MM-DD HH:MM:SS |

### *1.5.2 Mysql cơ bản*

Chúng ta có hai loại chú thích trong SQL:

* Chú thích dòng : Cách chú thích này thường được sử dụng để chú thích một đoạn mã ngắn ở cuối dòng hoặc ghi chú trên mỗi dòng lệnh.

--Nội dung chú thích tại đây

Tất cả những nội dung giữa cặp dấu “–” và cuối dòng đều sẽ được coi là chú thích và không được thực thi.

* Chú thích đoạn : Cách chú thích này thường được sử dụng để vô hiệu hóa một khối lệnh nào đó.

/\*Nội dung chú thích nằm ở đây\*/

Tất cả nội dung nằm giữa cặp dấu /\* và \*/ đều được coi là chú thích và sẽ không được thực thi.

*Phân cấp trong Mysql:*

Trong MySQL có khá nhiều thứ : DATABASE(cơ sở dữ liệu), TABLE(Bảng dữ liệu), RECORD(bản ghi).

RECORD chính là thứ chúng ta quan tâm nhất, nó chính là dữ liệu, ta sẽ đưa vào hoặc lấy ra.

*Các lệnh khởi tạo:*

* Tạo cơ sở dữ liệu

CREATE DATABASE ten\_csdl

CHARACTER SET utf8

COLLATE utf8\_unicode\_ci

* Khởi tạo bảng dữ liệu

CREATE TABLE ten\_bang(ten\_cot thuoc\_tinh(gia\_tri), ten\_cot thuoc\_tinh(gia\_tri),…

,ten\_cot thuoc\_tinh(gia\_tri), PRIMARY KEY(ten\_cot))

* Thêm một cột mới vào một bảng cho trước

ALTER TABLE ten\_bang ADD ten\_cot thuoc\_tinh(gia\_tri)AFTER ten\_cot

* Loại bỏ một cột

ALTER TABLE ten\_bang DROP ten\_cot

* Thêm bản ghi vào bảng dữ liệu

INSERT INTO ten\_bang(ten\_cot\_1, ten\_cot\_2, …, ten\_cot\_n)

                   VALUES(gia\_tri\_1, gia\_tri\_2, …, gia\_tri\_n)

* Lọc và hiển thị dữ liệu

SELECT \* FROM ten\_bang

* Lọc tất cả bản ghi với thông tin của một số cột nhất định

SELECT ten\_cot\_1,ten\_cot\_2,…,ten\_cot\_n FROM ten\_bang

* Lọc hạn chế bản ghi theo một điều kiện nào đó sử dụng mệnh đề WHERE

SELECT \*FROM ten\_bang WHERE ten\_cot=gia\_tri

* Sửa bản ghi

UPDATE FROM ten\_bang SET ten\_cot\_1 = gia\_tri

                         ten\_cot\_2 = gia\_tri\_2

                          …

                         ten\_cot\_n = gia\_tri\_n

                     WHERE ten\_cot = gia\_tri

* Xóa bản ghi

DELETE FROM ten\_bang WHERE ten\_cot=gia\_tri

* Toán tử AND

 Đối với toán tử AND, nếu muốn câu truy vấn trả về bản ghi có dữ liệu thì tất cả các điều kiện được kết hợp bởi toán tử AND cần phải đúng. Chỉ cần một điều kiện sai thì toàn bộ câu truy vấn sẽ trả về dữ liệu rỗng.

FROM ten\_bang

WHERE dieu\_kien\_1 AND dieu\_kien\_2 AND dieu\_kien\_3,...;

* Toán tử OR

Đối với toán tử OR, chỉ cần một trong các điều kiện được kết hơp đúng thì sẽ có bản ghi trả về.

SELECT cot\_1, cot\_2, ...

FROM ten\_bang

WHERE dieu\_kien\_1 OR dieu\_kien\_2 OR dieu\_kien\_3,...;

* Từ khóa LIKE

Từ khóa LIKE dùng để tìm kiếm sự tồn tại của một mẫu nào đó trong bảng dữ liệu

SELECT ten\_cot FROM ten\_bang WHERE ten\_cot LIKE mau\_can\_tim

* Từ khóa ORDER BY

Từ khóa ORDER BY dùng để sắp xếp các bản ghi tăng dần hoặc giảm dần theo một trường nào đó.

Sắp sếp tăng dần

SELECT ten\_cot FROM ten\_bang ORDER BY ten\_cot ASC

Sắp sếp giảm dần

SELECT ten\_cot FROM ten\_bang ORDER BY ten\_cot DESC

**Khóa ngoại (ForeignKey)**

Foreign key là mối quan hệ giữa hai bảng và mối quan hệ này ta hay gọi là cha – con. Nghĩa là nếu bảng A có một trường liên kết tới bảng B thì lúc này bảng B đóng vai trò là cha và bảng A đóng vai trò là con.

Nối hai bảng với INNER JOIN

  Quá trình nối hai bảng INNER JOIN sẽ trả lại cho chúng ta một bảng mới chứa các bản ghi có các trường chung trong điều kiện.

SELECT ten\_cot FROM ten\_bang\_1 INNER JOIN tean\_bang\_2

                 ON ten\_bang\_1.ten\_cot\_chung = ten\_bang\_2.ten\_cot\_chung

### *1.5.3 Kết nối Mysql với Android*

Các bước thực hiện:

*Tao cơ sở dữ liệu trên Mysql:*

Sử dụng phpmyadmin từ **http://localhost/phpmyadmin/** để tạo một cơ sở dữ liệu và môt bảng sử dụng câu lệnh

## CHƯƠNG 2: KHẢO SÁT PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

**2.1. Khảo sát hệ thống**

### *2.1.1 Nội dung đề tài*

Xây dựng hệ thống món ngon đường phố giúp mọi người có thể tìm các món ngon theo tỉnh thành và thực đơn nhanh nhất tiện ích nhất khi đi chơi, mở tiệc, … và ngoài ra còn giúp các bà các mẹ các chị, …vào bếp một cách dễ dàng.

Đây là một hệ thống chứa một lượng lớn thông tin về các tỉnh thành trên cả nước, món ăn nổi tiếng theo tỉnh thành, cách chế biến món ăn đó và thông tin cửa hàng bán món ăn đó. Người truy cập vào ứng dụng có thể sử dụng chức năng tìm kiếm món ăn theo tỉnh thành, tìm kiếm theo thực đơn, xem chi tiết món ăn và cách nấu. Người dùng có thể bình luận góp ý về món ăn hay cũng như chức năng của ứng dụng. Admin có quyền thêm, sửa, xóa dữ liệu về thực đơn, khóa tài khoản nhà hàng hoặc khách hàng. Khi đi chơi bạn đang không biết món ở đó có những món ngon gì thì ứng dụng cũng sẽ cung cấp cho người dùng về thông tin món ngon và cửa hàng nổi tiếng về món đó.

***2.1.2 Đối tượng và phạm vi***

Tất cả mọi người có nhu cầu tìm hiểu về các món ngon đường phố, cửa hàng bán món ngon đó. Dành cho những người nội chợ, khách du lịch.

***2.2.3Mô tả bài toán***

Xây dựng một hệ thống cho phép người dùng tìm các món ăn ngon đường phố theo tỉnh thành.

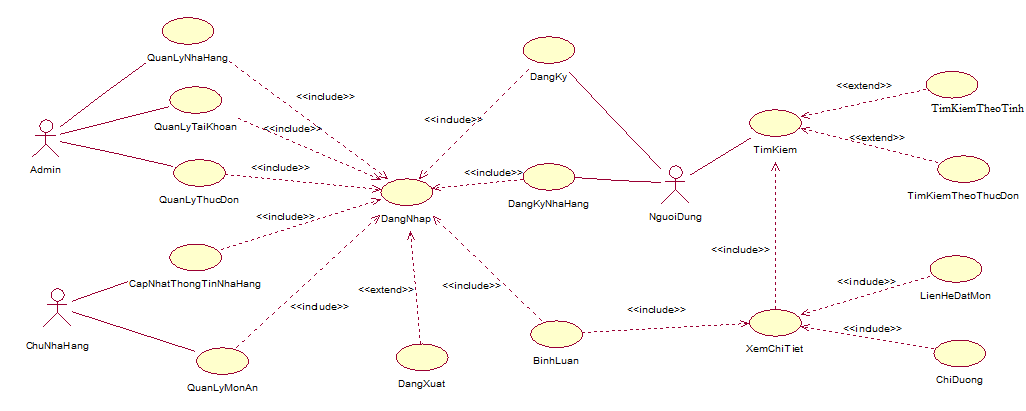
* Người dùng tìm món ăn theo tỉnh, tìm theo tên món ăn, nội dung bao gồm: tên tỉnh, tọa độ, mô tả đôi nét về tỉnh đó, tên món ăn, mô tả món ăn, công thức nấu.
* Khách du lịch dễ dàng tìm được món ngon cần thưởng thức mà không cần hướng dẫn viên du lịch.
* Giúp người vào bếp có thêm bí quyết nấu ăn.
* Hệ thống cung cấp cho người dùng về thông tin cửa hàng bán món đó.Ngoài các chức năng đó hệ thống còn cung cấp cho admin tính năng thêm , sửa, xóa các món ngon mới, cập nhật nhà hàng và cho phép người dùng bình luận về món ăn và chức năng để phát triển ứng dụng.

Để thực hiện điều này hệ thống cần đạt được các yêu cầu sau:

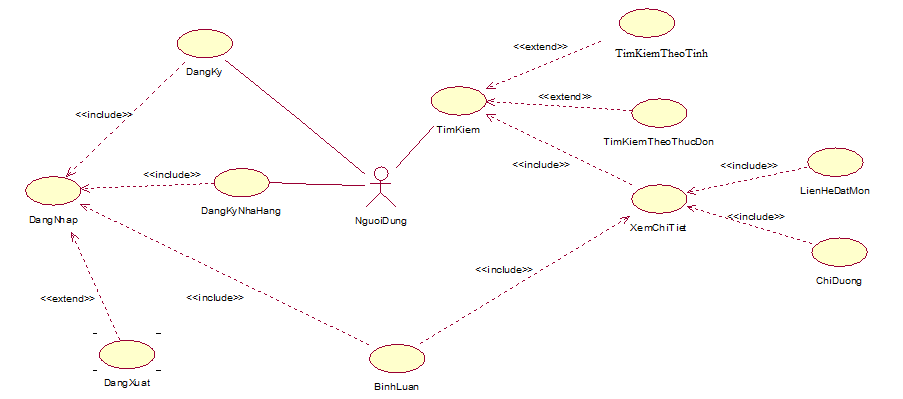
* Hệ thống có thông tin về các tỉnh thành trên cả nước, danh sách món ăn ngon theo tỉnh, cách thức chế biến, thông tin cửa hàng. Ở đây ngoài lưu thông tin ra có thể thêm các chức năng như xác nhận thông tin người dùng đăng ký,
* Hệ thống cho phép tìm kiếm món ăn theo loại món ăn hoặc theo tên món ăn.

**2.2 Phân tích thiết kế hệ thống**

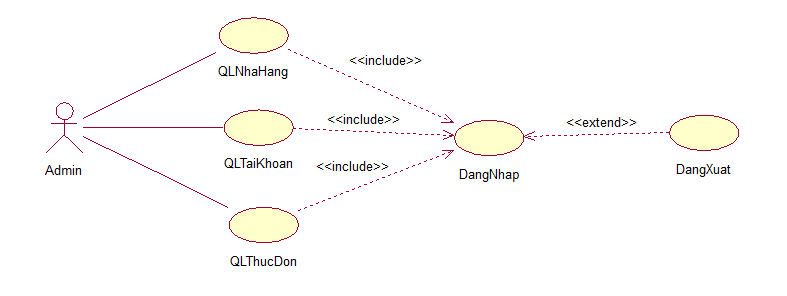
* + 1. ***Mô hình use cases***

**

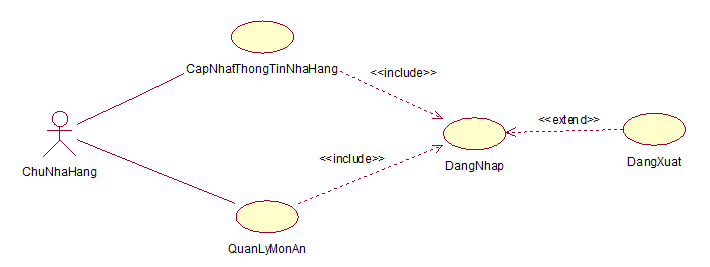
Hình 2. Biểu đồ ca sử dụng tổng quát



Hình 2. Bản vẽ use case cho actor người dùng



Hình 2. Bản vẽ use case cho Admin



Hình 2. Bản vẽ use case cho chủ nhà hàng

*Đặc tả các ca sử dụng*

Đặc tả ca sử dụng đăng nhập

*Tên ca sử dụng*: Ca sử dụng đăng nhập

*Mục đích*: Đảm bảo xác thực thông tin người sử dụng và an toàn bảo mật hệ thống

*Tác nhân, mô tả chung*:

Tác nhân: Người dùng, chủ nhà hàng, người quản trị.

Mô tả chung: Người dùng (chủ nhà hàng, quản trị hệ thống) khi muốn thao tác với hệ thống như bình luận, cập nhật thông tin, quản lý thông tin, khóa tài khoản,… phải đăng nhập vào hệ thống.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính của ca sử dụng đăng nhập

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn menu | - |
|  | 2. Hiển thị form menu |
| 3. Chọn chức năng đăng nhập | - |
|  | 4. Hiển thị form đăng nhập |
| 5. Nhập thông tin đăng nhập | - |
| 6. Gửi thông tin đã nhập đến hệ thống | - |
|  | 7. Kiểm tra thông tin đăng nhập, nếu đúng thì cho phép truy cập, sai thì thông báo yêu cầu nhập lại |

*Luồng thay thế*: Nếu đăng nhập thành công hệ thống dựa trên thông tin đăng nhập sẽ đồng thời phân quyền tùy theo loại người dùng hoặc admin. Nếu đăng nhập thất bại, hệ thống sẽ hiện thông báo cho người dùng và yêu cầu đăng nhập lại.

*Các yêu cầu cụ thể*: Tên đăng nhập và mật khẩu khớp với thông tin đã đăng ký.

*Điều kiện trước*: Nguời dùng đã đăng ký tài khoản trong hệ thống

*Điều kiện sau*: Thông báo đăng nhập thành công hoặc thất bại

*Các điểm mở rộng*: Đăng xuất.

Đặc tả ca sử dụng đăng xuất

*Tên ca sử dụng*: Ca sử dụng đăng xuất

*Mục đích*: Đảm bảo xác thực thông tin người sử dụng và an toàn bảo mật hệ thống

*Tác nhân, mô tả chung*:

Tác nhân: Người dùng, chủ nhà hàng, người quản trị.

Mô tả chung: Người dùng (chủ nhà hàng, quản trị hệ thống) khi muốn bảo mật thông tin tài khoản phải đăng xuất ra khỏi tài khoản.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính ca sử dụng đăng xuất

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chức năng đăng xuất | - |
|  | 2. Hiển thị nút đăng xuất |
| 3. Chọn nút đăng xuất | - |
|  | 4. Hiển thị thông báo xác nhận thực sự actor có muốn thoát không |
| 5. Chọn gửi xác nhận | - |
| - | 6. Kiểm tra thông tin |

*Luồng thay thế*: Không có .

*Các yêu cầu cụ thể*: Chọn đăng xuất và xác nhận thông báo.

*Điều kiện trước*: Nguời dùng đã đăng nhập tài khoản trong hệ thống

*Điều kiện sau*: Thông báo đăng xuất thành công hoặc thất bại

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng đăng ký tài khoản:

*Tên ca sử dụng*: Ca sử dụng đăng ký tài khoản:

*Mục đích*: Đảm bảo xác thực thông tin người sử dụng và an toàn bảo mật hệ thống

*Tác nhân, mô tả chung*:

Tác nhân: Người dùng, chủ nhà hàng.

Mô tả chung: Cho phép xem đăng ký tài khoản của hệ thống.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính ca sử dụng đăng ký tài khoản

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chức năng đăng ký tài khoản | - |
|  | 2. Hiển thị form đăng ký |
| 3. Chọn nhập thông tin đăng ký | - |
| 4. Chọn nút đăng ký | - |
| - | 5. Hệ thống thông báo kết quả của quá trình nhập thông tin. Nếu không chính xác thì thực hiện luồng thay thế. |
| - | 6. Hệ thống nhập thông tin đăng ký vào danh sách tài khoản. |

*Luồng thay thế*: Nếu thông tin không chính xác thì hệ thống sẽ báo không chính xác, yêu cầu khách hàng nhập lại và tiếp tục vào form đăng ký.

*Các yêu cầu cụ thể*: Chọn đăng ký.

*Điều kiện trước*: Không có

*Điều kiện sau*: Thông báo đăng ký thành công hoặc thất bại

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng đăng ký nhà hàng

*Tên ca sử dụng*: Ca sử dụng đăng ký tài nhà hàng:

*Mục đích*: Đảm bảo xác thực thông tin nhà hàng và an toàn bảo mật hệ thống

*Tác nhân, mô tả chung*:

Tác nhân: Người dùng.

Mô tả chung: Cho phép xem đăng ký nhà hàng của hệ thống.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính của ca sử dụng đăng ký nhà hàng

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chức năng đăng ký nhà hàng | - |
| - | 2. Hiển thị form đăng ký |
| 3. Chọn nhập thông tin đăng ký | - |
| 4. Chọn nút đăng ký | - |
| - | 5. Hệ thống thông báo kết quả của quá trình nhập thông tin. Nếu không chính xác thì thực hiện luồng thay thế. |
| - | 6. Hệ thống nhập thông tin đăng ký vào danh sách nhà hàng. |

*Luồng thay thế*: Nếu thông tin không chính xác thì hệ thống sẽ báo không chính xác, yêu cầu khách hàng nhập lại và tiếp tục vào form đăng ký nhà hàng.

*Các yêu cầu cụ thể*: Chọn đăng ký.

*Điều kiện trước*: Không có

*Điều kiện sau*: Thông báo đăng ký thành công hoặc thất bại

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng người dùng, chủ nhà hàng bình luận

*Tên ca sử dụng*: Ca sử dụng người dùng, chủ nhà hàng bình luận :

*Mục đích*: Góp ý kiến về món ăn, cảm nhận về ứng dụng.

*Tác nhân, mô tả chung*:

Tác nhân: Người dùng, chủ nhà hàng.

Mô tả chung: Cho phép bình luận món ăn của hệ thống.

*Luồng sự kiện chính*:

Bảng 2. 5 Luồng dữ liệu chính ca sử dụng bính luận

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chức năng bình luận | - |
| - | 2. Hiển thị bình luận |
| 3. Chọn nhập thông tin bình luận | - |
| 4. Chọn nút gửi bình luận | - |
| - | 5. Hệ thống thông báo kết quả của quá trình nhập thông tin. Nếu không chính xác thì thông báo lỗi. |
| - | 6. Hệ thống nhập thông tin bình luận vào danh sách bình luận. |

*Luồng thay thế*: Không có.

*Các yêu cầu cụ thể*: Chọn bình luận và gửi bình luận.

*Điều kiện trước*: Người dùng phải đăng nhập tài khoản, xem chi tiết món ăn.

*Điều kiện sau*: Thông báo bình luận thành công hoặc thất bại.

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng: Tìm kiếm

*Tên ca sử dụng*: Tìm kiếm :

*Mục đích*: Tìm hiểu về món ăn cách chế biến công thức, tham khảo giá và gọi điện đặt hàng.

*Tác nhân, mô tả chung*:

Tác nhân: Người dùng, chủ nhà hàng.

Mô tả chung: Tìm kiếm món ăn theo tỉnh, theo loại món ăn.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính ca sử dụng tìm kiếm

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chức năng tìm kiếm | - |
| - | 2. Hiển thị form tìm kiếm |
| 3. Chọn tìm kiếm theo tỉnh hoặc theo loại món ăn | - |
| - | 4. Hiển thị món ăn |
| 5. Chọn món ăn | - |
| - | 6. Hệ thống kiểm tra kết quả và hiển thị. |

*Luồng thay thế*: Không có.

*Các yêu cầu cụ thể*: Chọn tìm kiếm theo tỉnh hoặc theo loại món ăn.

*Điều kiện trước*: Không có.

*Điều kiện sau*: Không có.

*Các điểm mở rộng*: Tìm kiếm theo tỉnh, tìm kiếm theo loại món ăn.

Đặc tả ca sử dụng: Xem chi tiết món ăn

*Tên ca sử dụng*: Xem chi tiết món ăn :

*Mục đích*: Tìm hiểu về món ăn cách chế biến công thức, tham khảo giá và gọi điện đặt hàng, tìm đến nhà hàng có bán món ăn đó.

*Tác nhân, mô tả chung*:

Tác nhân: Người dùng, chủ nhà hàng.

Mô tả chung: Xem thông tin chi tiết về món ăn.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính ca sử dụng xem chi tiết món ăn

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn xem chi tiết | - |
| - | 2. Hiển thị form chi tiết món ăn |

*Luồng thay thế*: Không có.

*Các yêu cầu cụ thể*: Chọn xem món ăn.

*Điều kiện trước*: Tìm kiếm.

*Điều kiện sau*: Không có.

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng: Gọi điện liên hệ với nhà hàng để đặt món ăn.

*Tên ca sử dụng*: Gọi điện liên hệ với nhà hàng để đặt món ăn:

*Mục đích*: Đặt mua món ăn đó.

*Tác nhân, mô tả chung*:

Tác nhân: Người dùng.

Mô tả chung: Gọi điện đến nhà hàng để đặt mua món ăn.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính ca sử dụng liên hệ đặt món

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn gọi cho nhà hàng | - |
| - | 2. Trả về kết quả gọi điện |

*Luồng thay thế*: Không có.

*Các yêu cầu cụ thể*: Chọn gọi điện đặt món ăn.

*Điều kiện trước*: Tìm kiếm, xem chi tiết món ăn.

*Điều kiện sau*: Không có.

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng: Chỉ đường đến nhà hàng.

*Tên ca sử dụng*: Chỉ đường đến nhà hàng:

*Mục đích*: Tìm đến nhà hàng để thưởng thức, đặt mua món ăn.

*Tác nhân, mô tả chung*:

Tác nhân: Người dùng.

Mô tả chung: Chỉ đường đến nhà hàng mình muốn đến.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính ca sử dụng chỉ đường

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn chỉ đường | - |
| - | 2. Trả về kết quả chỉ đường |

*Luồng thay thế*: Không có.

*Các yêu cầu cụ thể*: Chọn chỉ đường đến nhà hàng.

*Điều kiện trước*: Tìm kiếm, xem chi tiết món ăn.

*Điều kiện sau*: Không có.

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng: Cập nhật thông tin nhà hàng

*Tên ca sử dụng*: Cập nhật thông tin nhà hàng:

*Mục đích*: Thay đổi bảo mật thông tin chính xác với hệ thống.

*Tác nhân, mô tả chung*:

Tác nhân: Chủ nhà hàng.

Mô tả chung: Cập nhật thông tin nhà hàng với hệ thống.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính của cập nhật thông tin nhà hàng

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn cập nhật thông tin | - |
| - | 2. Hiển thị form cập nhật thông tin |
| 3. Chọn nhập thông tin cập nhật | - |
| 4. Chọn nút cập nhật | - |
| - | 5. Hệ thống thông báo kết quả của quá trình cập nhật thông tin. Nếu không chính xác thì thông báo lỗi. |
| - | 6. Hệ thống nhập thông tin cập nhật vào danh sách nhà hàng. |

*Luồng thay thế*: Không có.

*Các yêu cầu cụ thể*: Chọn cập nhật và gửi cập nhật.

*Điều kiện trước*: Chủ nhà hàng phải đăng nhập tài khoản.

*Điều kiện sau*: Thông báo cập nhật thành công hoặc thất bại.

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng: Quản lý món ăn

*Tên ca sử dụng*: Quản lý món ăn:

*Mục đích*: Thay đổi bảo mật thông tin món ăn chính xác với hệ thống.

*Tác nhân, mô tả chung*:

Tác nhân: Chủ nhà hàng.

Mô tả chung: Cập nhật, thêm , xóa thông tin món ăn với hệ thống.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính ca sử dụng quản lý món ăn

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn quản lý món ăn | - |
| - | 2. Hiển thị form quản lý thông tin món ăn |
| 3. Chọn thêm hoặc sửa hoặc xóa món ăn | - |
| - | 4. Hiển thị form thêm, sửa hoặc kiểm tra xóa món ăn |
| 5. Nếu thêm hoặc sửa thì nhập nội dung thông tin món ăn |  |
| - | 6. Hệ thống nhập thông tin cập nhật vào danh sách món ăn. |

*Luồng thay thế*: Không có.

*Các yêu cầu cụ thể*: Chọn thêm, cập nhật, xóa và gửi yêu cầu đến hệ thống.

*Điều kiện trước*: Chủ nhà hàng phải đăng nhập tài khoản.

*Điều kiện sau*: Thông báo, thêm, cập nhật, xóa thành công hoặc thất bại.

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng: Quản lý nhà hàng

*Tên ca sử dụng*: Quản lý nhà hàng:

*Mục đích*: Cho phép nhà hàng có được hoạt động không.

*Tác nhân, mô tả chung*:

Tác nhân: Người quản trị.

Mô tả chung: Gửi yêu cầu khóa nhà hàng đến hệ thống.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính quản lý nhà hàng

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn quản lý nhà hàng | - |
| - | 2. Hiển thị form quản lý nhà hàng |
| 3. Chọn khóa tài khoản nhà hàng | - |
| - | 4. Kiểm tra và gửi kết quả thông báo thành công hay lỗi |

*Luồng thay thế*: Không có.

*Các yêu cầu cụ thể*: Chọn xem và khóa tài khoản nhà hàng.

*Điều kiện trước*: Người quản trị phải đăng nhập tài khoản.

*Điều kiện sau*: Thông báo khóa nhà hàng thành công hoặc thất bại.

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng: Quản lý tài khoản

*Tên ca sử dụng*: Quản lý tài khoản:

*Mục đích*: Cho phép tài khoản có được hoạt động không.

*Tác nhân, mô tả chung*:

Tác nhân: Người quản trị.

Mô tả chung: Gửi yêu cầu khóa tài khoản đến hệ thống.

*Luồng sự kiện chính*:

Bảng 2. Luồng dữ liệu chính ca sử dụng quản lý tài khoản

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn quản lý tài khoản | - |
| - | 2. Hiển thị form quản lý tài khoản |
| 3. Chọn khóa tài khoản người dùng, chủ nhà hàng | - |
| - | 4. Kiểm tra và gửi kết quả thông báo thành công hoặc lỗi |

*Luồng thay thế*: Không có.

*Các yêu cầu cụ thể*: Chọn xem và khóa tài khoản người dùng, chủ nhà hàng.

*Điều kiện trước*: Người quản trị phải đăng nhập tài khoản.

*Điều kiện sau*: Thông báo khóa tài khoản người dùng, chủ nhà hàng thành công hoặc thất bại.

*Các điểm mở rộng*: Không có.

Đặc tả ca sử dụng: Quản lý loại món ăn

*Tên ca sử dụng*: Quản lý loại món ăn:

*Mục đích*: Thay đổi bảo mật thông tin loại món ăn chính xác với hệ thống.

*Tác nhân, mô tả chung*:

Tác nhân: Người quản trị.

Mô tả chung: Cập nhật, thêm , xóa thông tin loại món ăn với hệ thống.

*Luồng sự kiện chính*:

Bảng 2. 14 Luồng dữ liệu chính ca sử dụng quản lý món ăn

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. Chọn quản lý loại món ăn | - |
| - | 2. Hiển thị form quản lý thông tin loại món ăn |
| 3. Chọn thêm hoặc sửa hoặc xóa loại món ăn | - |
| - | 4. Hiển thị form thêm, sửa hoặc kiểm tra xóa loại món ăn |
| 5. Nếu thêm hoặc sửa thì nhập nội dung thông tin loại món ăn |  |
| - | 6. Hệ thống nhập thông tin cập nhật vào danh sách loại món ăn. |

*Luồng thay thế*: Không có.

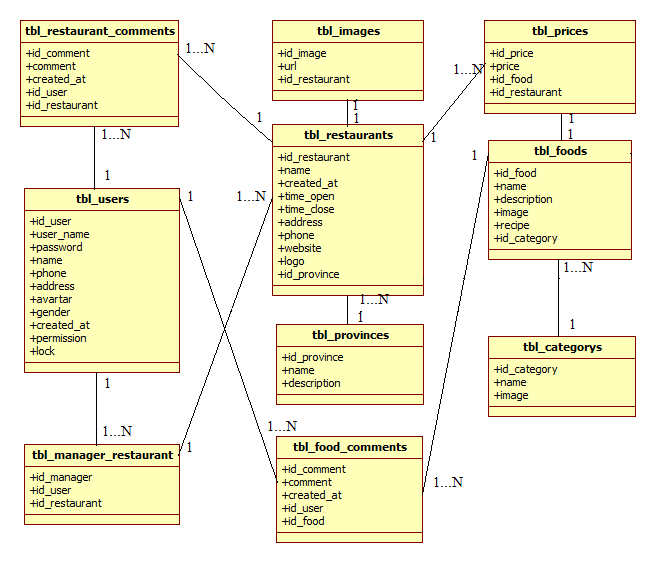
*Các yêu cầu cụ thể*: Chọn thêm, cập nhật, xóa và gửi yêu cầu đến hệ thống.

*Điều kiện trước*: Người quản trị phải đăng nhập tài khoản.

*Điều kiện sau*: Thông báo, thêm, cập nhật, xóa thành công hoặc thất bại.

*Các điểm mở rộng*: Không có.

### *Mô hình lớp*



Hình 2. Mô hình lớp

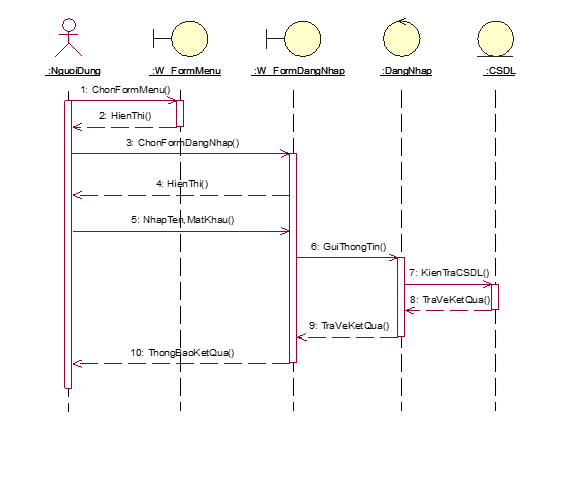
* + 1. ***Biểu đồ trình tự***

**

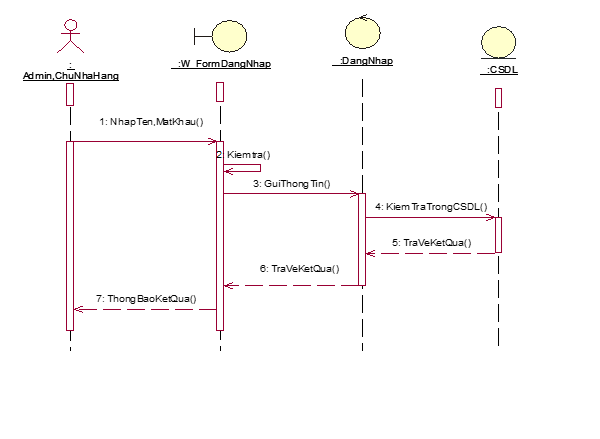
Hình 2. Biểu đồ tuần tự đăng xuất của người dùng

**

Hình 2. Biểu đồ tuần tự đăng xuất của chủ nhà hàng

**

Hình 2. Biểu đồ đăng nhập người dùng

**

Hình 2. Biểu đồ đăng nhập chủ nhà hàng và admin

**

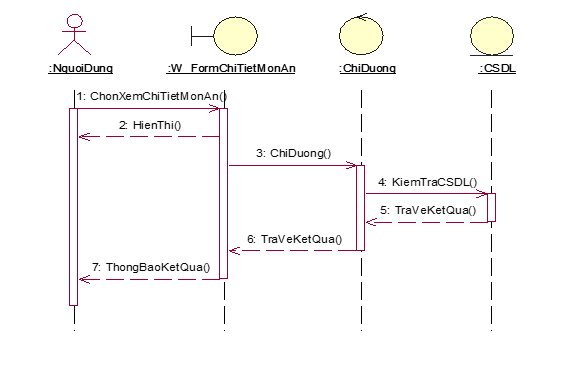
Hình 2. Biểu đồ tuần tự đăng ký tài khoản của người dùng

**

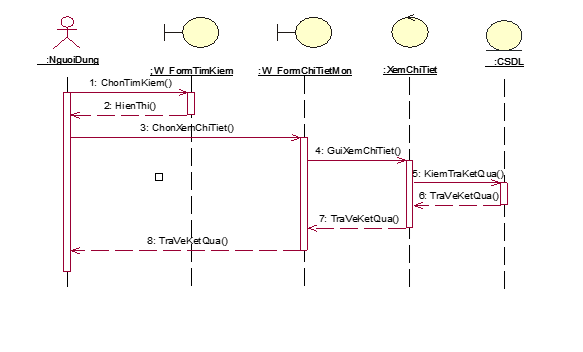
Hình 2. Biểu đồ đăng ký nhà hàng của chủ nhà hàng

**

Hình 2. Biểu đồ tuần tự cập nhật nhà hàng của chủ nhà hàng

**

Hình 2. Biểu đồ tuần tự chỉ đường



Hình 2. Biểu đồ tuần tự xem chi tiết món ăn

**

Hình 2. Biểu đồ tuần tự bình luận

**

Hình 2.  Biểu đồ tuần tự liên hệ đặt món

**

Hình 2. Biểu đồ tuần tự tìm kiếm theo tỉnh

**

Hình 2. Biểu đồ tuần tự tìm kiếm theo thực đơn

**

Hình 2. Biểu đồ tuần tự quản lý nhà hàng của admin

**

Hình 2. Biểu đồ tuần tự quản lý tài khoản của admin

**

Hình 2. Biểu đồ quản lý thực đơn của admin

**

Hình 2. Biểu đồ tuần tự quản lý món ăn

* 1. ***Thiết kế cơ sở dữ liệu***
     1. ***Thiết kế cơ sở dữ liệu dựa trên hệ quản trị cơ sở dữ liệu MySQL***
* *Mô tả chi tiết các bảng*

Bảng 2. : Mô tả chi tiết bảng tbl\_restaurant\_comments

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu | Chú thích | Mô tả |
| **id\_comment** | **int** | **Primary key** | **Mã số của bình luận nhà hàng** |
| comment | text | Not null | Nội dung bình luận |
| created\_at | text | Not null | Ngày tạo bình luận |
| id\_user | int(11) | Not null | Mã số của người dùng bình luận |
| id\_restaurant | int(11) | Not null | Mã số của nhà hàng bình luận |

Bảng 2. : Mô tả chi tiết bảng tbl\_user

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu | Chú thích | Mô tả |
| **id\_user** | **int (11)** | **Primary key** | **Mã người dùng** |
| user\_name | varchar(50) | Not null | Tên tài khoản người dùng đăng ký |
| password | varchar(50) | Not null | Mật khẩu người dùng |
| name | varchar(50) | Not null | Tên người dùng |
| phone | varchar(12) | Not null | Số điện thoại người dùng |
| address | text | Not null | Địa chỉ người dùng |
| avartar | text | Not null | Ảnh đại diện người dùng |
| gender | varchar(5) | Not null | Giới tính người dùng |
| created\_at | text | Not null | Ngày tạo |
| permission | int(11) | Not null | Trạng thái hoạt động của người dùng |
| lock | int(11) | Not null | Khóa người dùng |

BBảng 2. : Mô tả chi tiết bảng tbl\_manager\_restaurant

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu | Chú thích | Mô tả |
| **id\_manager** | **int(11)** | **Primary key** | **Mã quản lý nhà hàng** |
| id\_user | int(11) | Not null | Mã số tài khoản người dùng |
| id\_restaurant | int(11) | Not null | Mã nhà hàng |

Bảng 2. : Mô tả chi tiết bảng tbl\_images

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu | Chú thích | Mô tả |
| **id\_image** | **int (11)** | **Primary key** | **Mã số ảnh nhà hàng** |
| url | text | Not null | Đường dẫn link ảnh nhà hàng |
| id\_restaurant | int(11) | Not null | Mã số nhà hàng |

Bảng 2. : Mô tả chi tiết bảng tbl\_restaurants

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu | Chú thích | Mô tả |
| **Id\_restaurant** | **int(11)** | **Not null** | **Mã số nhà hàng** |
| name | text | Not null | Tên nhà hàng |
| created\_at | text | Not null | Ngày thành lập nhà hàng |
| time\_open | varchar(6) | Not null | Thời gian mở cửa nhà hàng |
| time\_close | varchar(6) | Not null | Thời gian đóng cửa nhà hàng |
| address | text | Not null | Địa chỉ nhà hàng |
| phone | varchar(20) | Not null | Số điện thoại liên hệ nhà hàng |
| website | text | Not null | Trang web nhà hàng |
| logo | text | Not null | Hình ảnh logo nhà hàng |
| id\_province | int(11) | Not null | Mã số tỉnh của nhà hàng |

Bảng 2. : Mô tả chi tiết bảng tbl\_provinces

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu | Chú thích | Mô tả |
| **Id\_province** | **int(11)** | **Primary key** | **Mã số tỉnh** |
| name | text | Not null | Tên của tỉnh |
| description | text | Not null | Miêu tả về tỉnh |

Bảng 2. : Mô tả chi tiết bảng tbl\_food\_comments

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu | Chú thích | Mô tả |
| **id\_comment** | **int(11)** | **Primary key** | **Mã số bình luận món ăn** |
| comment | text | Not null | Nội dung bình luận món ăn |
| created\_at | text | Not null | Ngày tạo bình luận của món ăn |
| id\_user | int(11) | Not null | Mã số tài khoản bình luận món ăn |
| id\_food | int(11) | Not null | Mã số món ăn được bình luận |

Bảng 2. : Mô tả chi tiết bảng tbl\_prices

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu | Chú thích | Mô tả |
| **id\_price** | **int(11)** | **Primary key** | **Mã số giá của món ăn thuộc các nhà hàng** |
| price | int(11) | Not null | Gía món ăn thuộc nhà hàng đó |
| id\_food | int(11) | Not null | Mã số món ăn |
| id\_restaurant | int(11) | Not null | Mã số nhà hàng |

Bảng 2. : Mô tả chi tiết bảng tbl\_foods

|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu | Chú thích | Mô tả |
| **id\_food** | **int(11)** | **Primary key** | **Mã số của món ăn** |
| name | text | Not null | Tên món ăn |
| description | text | Not null | Miêu tả món ăn |
| image | text | Not null | Đường dẫn link ảnh món ăn |
| recipe | text | Not null | Công thức, cách chế biến món ăn đó |
| id\_catsgory | int(11) | Not null | Mã số loại món ăn |

Bảng 2. : Mô tả chi tiết bảng tbl\_categorys

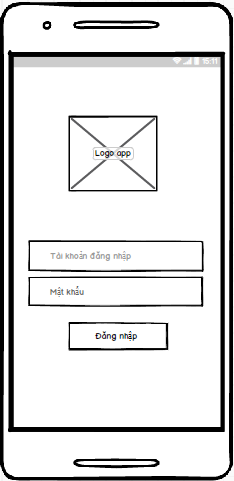
|  |  |  |  |
| --- | --- | --- | --- |
| Trường | Kiểu | Chú thích | Mô tả |
| **id\_category** | **int(11)** | **Primary key** | **Mã số loại món ăn** |
| name | text | Not null | Tên loại món ăn |
| image | text | Not null | Đường dẫn link ảnh của loại món ăn |

### 

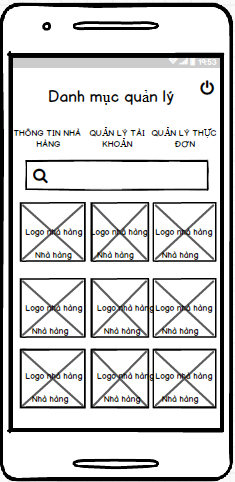
### Thiết kế giao diện



Hình 2. Thiết kế giao diện Flash của app ShareFoodManager



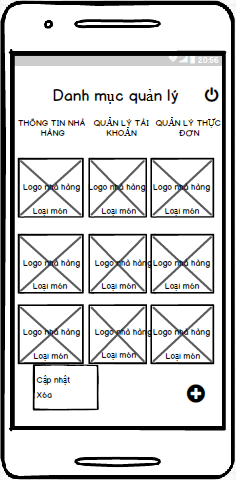
Hình 2. Thiết kế màn hình đăng nhập củ chủ nhà hàng và admin



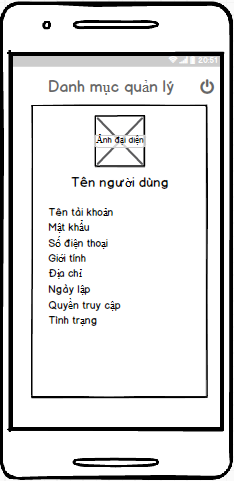
Hình 2. Thiết kế màn hình quản lý nhà hàng



Hình 2. Thiết kế màn hình quản lý tài khoản



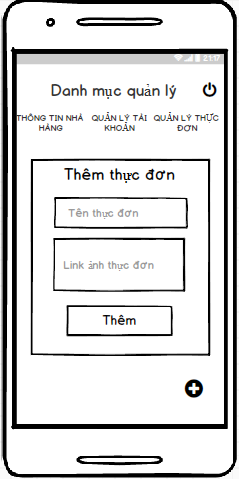
Hình 2. Thiết kế màn hình quản lý thực đơn



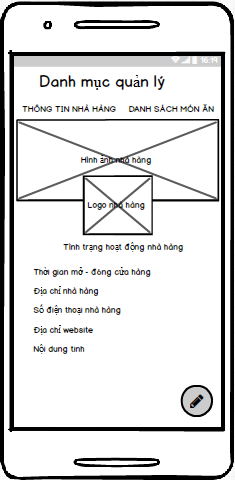
Hình 2. Thiết kế màn hình chi tiết tài khoản



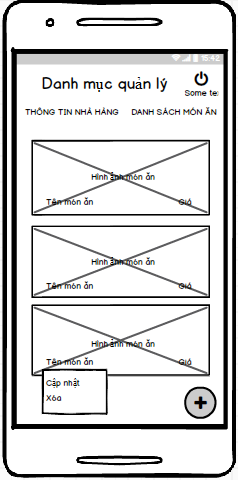
Hình 2. Thiết kế giao diện cập nhật đơn hàng



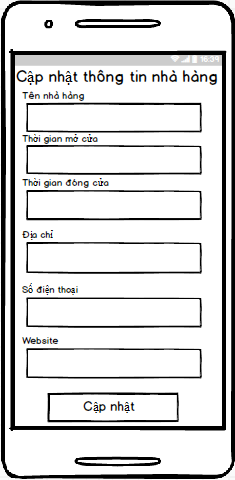
Hình 2. Thiết kế giao diện thêm thực đơn



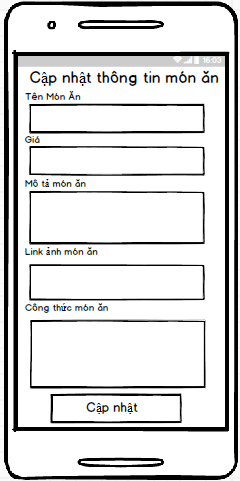
Hình 2. Thiết kế giao diện quản lý nhà hàng



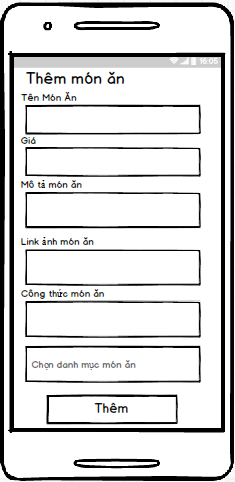
Hình 2. Thiết kế giao diện quản lý món ăn



Hình 2. Thiết kế giao diện cập nhật nhà hàng



Hình 2. Thiết kế giao diện cập nhật món ăn

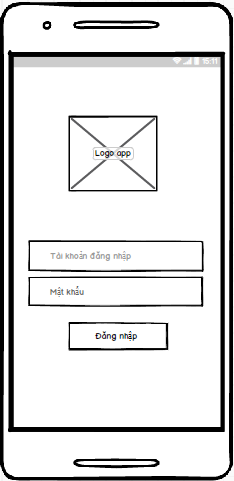


Hình 2. Thiết kế giao diện thêm món ăn

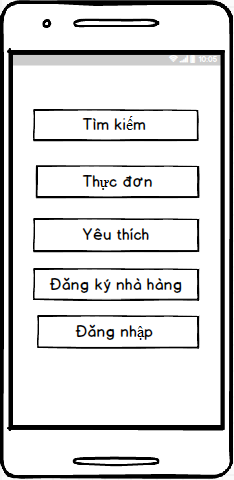
Thiết kế giao diện client



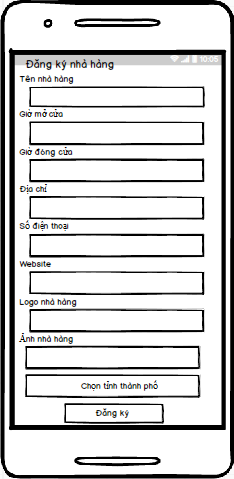
Hình 2. Thiết kế giao diện Flash của app ShareFood



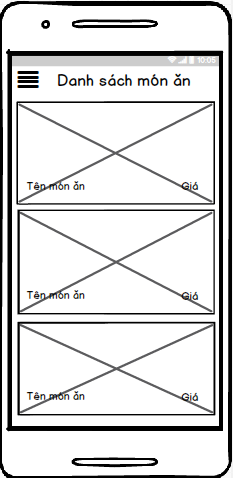
Hình 2. Thiết kế giao diện đăng nhập của người dùng



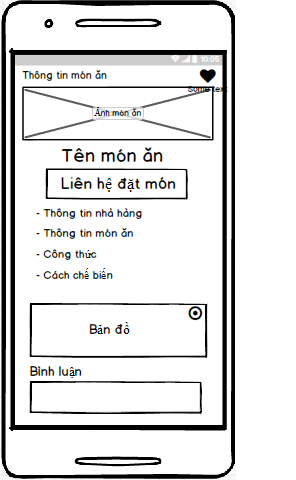
Hình 2. 38 Thiết kế giao diện menu



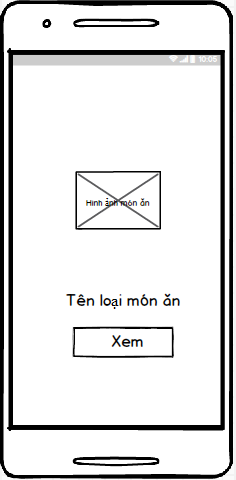
Hình 2. Thiết kế giao diện đăng ký nhà hàng



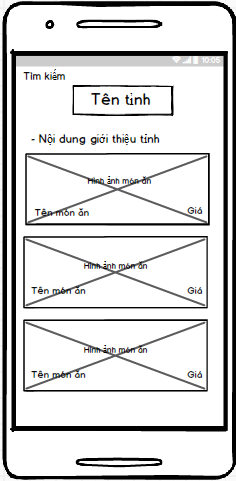
Hình 2. Thiết kế giao diện màn hình chính



Hình 2. Thiết kế giao diện thông tin món ăn



Hình 2. Thiết kế giao diện tìm kiếm theo thực đơn



Hình 2. Thiết kế giao diện tìm kiếm theo tỉnh

# CHƯƠNG 3: CÀI ĐẶT ỨNG DỤNG

* 1. ***Yêu cầu môi trường***
* Android Studio là môi trường phát triển tích hợp (Integrated Development Environment – IDE) chính thức cho việc phát triển ứng dụng cho nền tảng Android. Nó được cung cấp miễn phí theo giấy phép Apache v2.0, có sẵn các phiên bản dành cho Windows, Mac OS X và Linux. Ngoài ra, ta cũng có thể dùng Eclipse để thay thế, tuy nhiên cần lưu ý là chỉ có Android Studio mới được hỗ trợ chính thức từ Google.
* Android Software Development Kit (Android SDK) bao gồm một tập các công cụ hỗ trợ cho việc lập trình trên nền tảng Android như trình biên dịch, các thư viện, máy ảo, tài liệu, source code mẫu… Thông thường, Android SDK sẽ được tích hợp vào Android Studio hoặc Eclipse để hỗ trợ cho việc phát triển ứng dụng Android.
* Để bắt đầu cho việc phát triển ứng dụng Android, ta có thể tải về bộ Android Studio Bundle mà bên trong nó bao gồm Android Studio, Android SDK và các thành phần khác tại trang chủ  [http://developer.android.com.](https://developer.android.com/studio/index.html)
* Yêu cầu hệ thống để cài đặt Android Studio cho từng hệ điều hành như sau:  
  Windows OS X Linux

Hệ điều hành

Windows 10/8.1/8/7/Vista/2003/XP (32 or 64 bit)

Mac OS X 10.8.5 trở lên

Máy tính bàn GNOME hoặc KDE trên Ubuntu hoặc Fedora hoặc GNU/Linux Debian  
RAM 2 GB RAM trở lên, khuyến cáo 4 GB RAM

Dung lượng đĩa cứng

500 MB dung lượng đĩa cứng

Khoảng trống cho Android SDK

Ít nhất 1 GB cho Android SDK, máy ảo, file tạm…

JDK Java Development Kit (JDK) 7 hoặc cao hơn Độ phân giải màn hình 1280×800 trở lên

* Giả lập thiết bị

Android SDK bao gồm một giả lập (emulator) cho thiết bị Android chạy trên máy tính.

Chạy trên máy ảo Genymotion, Memu, …

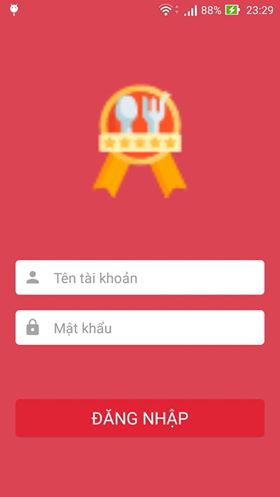
Chạy trên máy thật yêu cầu mặc định android 4.2 trở lên, có dây cáp kết nối và có cài đặt driver cho máy thì khi đó điện thoại mới kết nối được với chương trình để chạy. Đặt biệt chỉ chạy được trên điện thoại chạy hệ điều hành android còn ios, linux, … sẽ không chạy được.

* 1. ***Kết quả cài đặt chương trình***
     1. ***Một số giao diện của admin và chủ nhà hàng***



Hình 3. Giao diện khởi động

Chờ màn hình chạy 5 giây sẽ sang màn hình đăng nhập

**

Hình 3. Giao diện đăng nhập

Đăng nhập thành công hệ thống sẽ báo thành công và vào màn hình quản lý nhà hàng và món ăn.

**

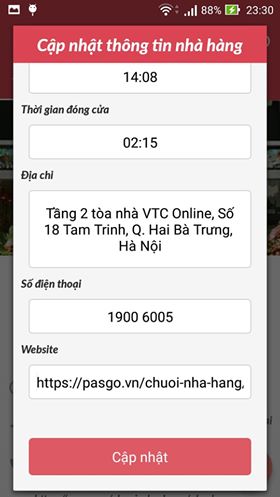
Hình 3. Giao diện cập nhật nhà hàng

Xem chi tiết nhà hàng và click nút bút màu trắng là cập nhật nhà hàng.

**

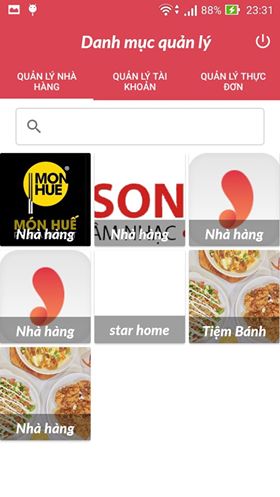
Hình 3. Giao diện quản lý món ăn

Hiển thị danh sách món ăn, click dấu cộng là thêm món ăn, sửa và xóa món ăn là click vào từng món.

**

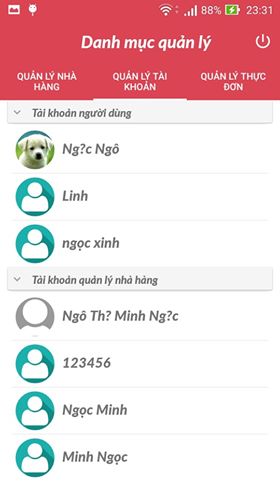
Hình 3. Giao diện cập nhật nhà hàng

Điền thông tin cập nhật nhà hàng.

**

Hình 3. Giao diện quản lý nhà hàng

Quản lý nhà hàng, tài khoản, thực đơn bằng cách vuốt ngang.

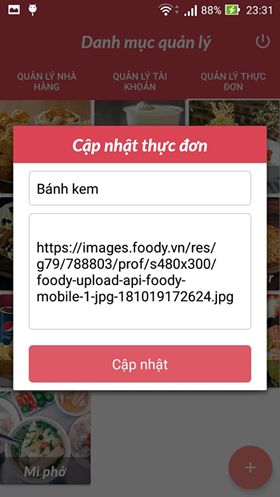
**

Hình 3. Giao diện quản lý tài khoản

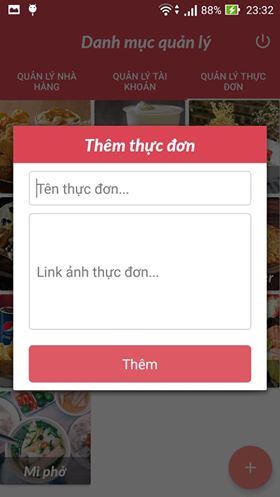
**

Hình 3. Giao diện quản lý thực đơn

Click vào nút cộng để thêm thực đơn còn sửa với xóa thì nhấn giữ vào từng thực đơn đó.

**

Hình 3. Giao diện cập nhật thực đơn

**

Hình 3. 10 Giao diện thêm thực đơn

**

Hình 3. Giao diện đăng xuất

* + 1. *Giao diện của client*

**

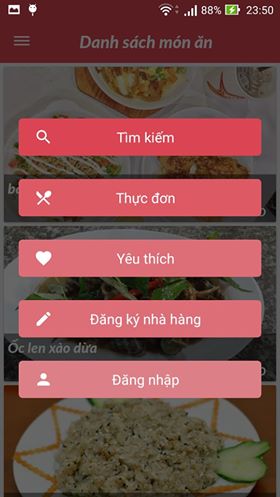
Hình 3. Giao diện màn hình khởi động

*Giao diện sau khi bật app.*

**

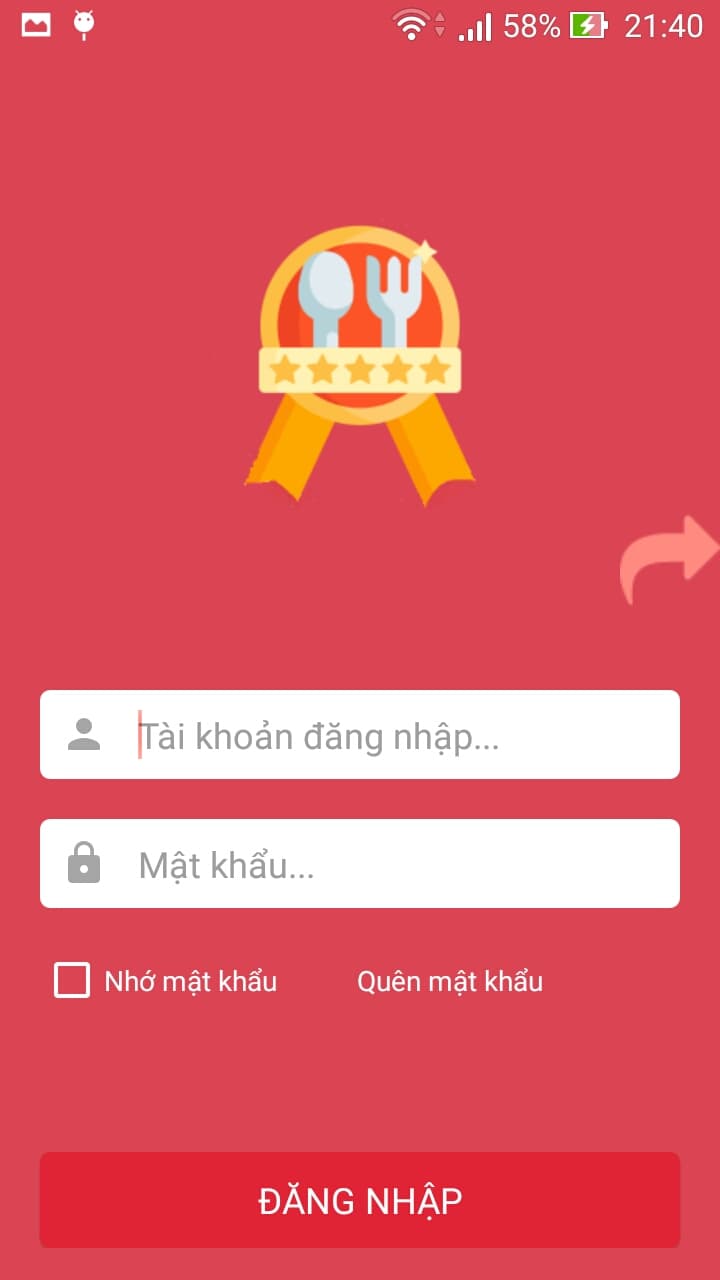
Hình 3. Giao diện danh sách món ăn

Danh Sách các món ăn và click vào từng món ăn sẽ sâu ra chi tiết món ăn đó và thông tin về nhà hàng, chỉ đường.

**

Hình 3. 14 Giao diện menu

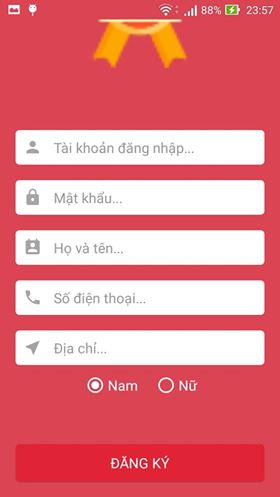
Để chọn chức năng của app.



Hình 3. Giao diện đăng nhập

Kết quả màn hình đăng nhập nếu sai hoặc đúng thì màn hình sẽ trả về một dòng thông báo.

Khi đăng nhập có lưu mật khẩu và quên mật khẩu.

**

Hình 3. Giao diện đăng ký

Màn hình đăng ký chọn giới tính và không để trống bất ký một trường nào để đăng nhập.

**

Hình 3. Giao diện tìm theo thực đơn

Mũi tên thể hiện tìm theo thực đơn bằng cách vuốt ngang màn hình.

**

Hình 3. Giao diện món ăn theo tỉnh

Tìm kiếm theo tỉnh sẽ có thông tin tỉnh và món ăn ở đó.

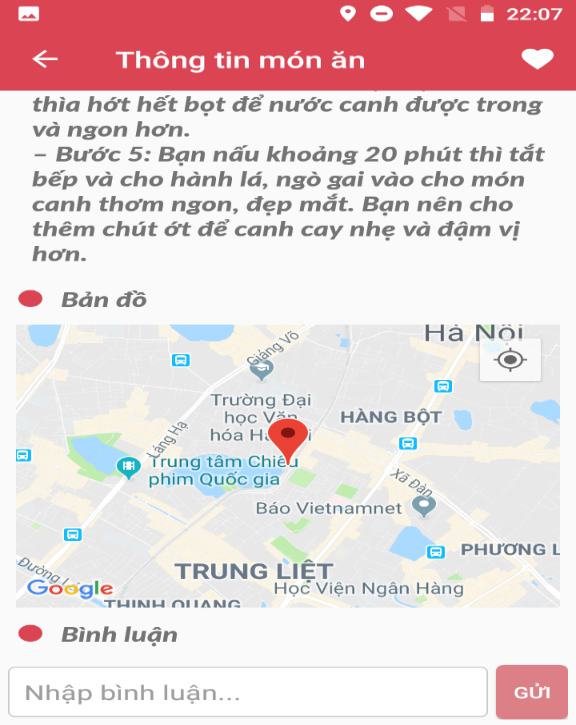
**

Hình 3. Giao diện yêu thích

**

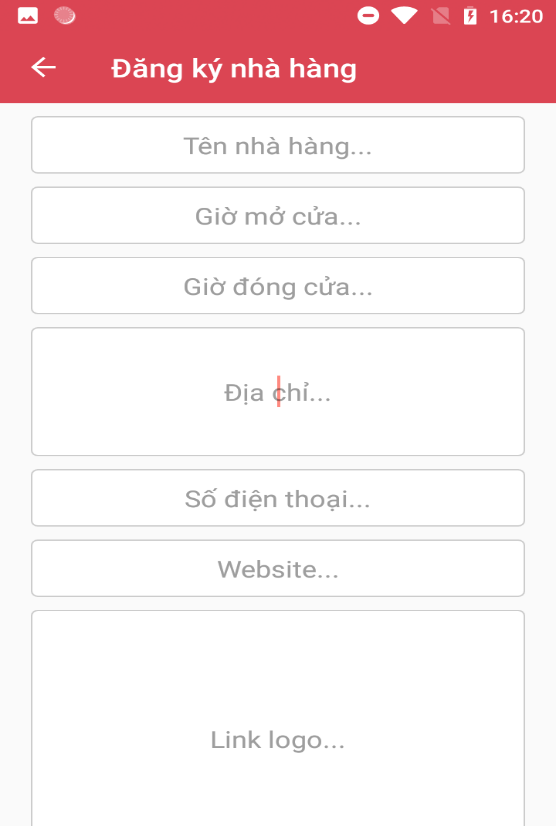
Hình 3. Giao diện chi tiết món ăn

Gọi điện thoại liên hệ đặt hàng món ăn khi click vào liên hệ đặt món.



Hình 3. Giao diện chỉ đường bình luận

Nhấn vào nút bên dưới chữ Hà Nội sẽ xâu ra vị trí nhà hàng.

**

Hình 3. Giao diện đăng ký nhà hàng

* 1. ***Ứng dụng công nghệ vào chương trình***

Áp dụng android studio 3.2.1, sử dụng MySql để hoạt động app online. Viết API bằng Java Spring. Hiện nay có nhiều cách lấy dữ liệu từ mysql để hiển thị nhưng chúng em đã chọn retrofit để lấy dữ liệu hiển thị (thư viện này giúp developers không phải viết nhiều code như trước và dùng java Spring. Sử dụng thư viện lombok cũng để giảm phải code nhiều, thư viên glide để load ảnh, Shimmer là một thư viện cung cấp một cách dễ dàng để thêm hiệu ứng load data cho ứng dụng Android, ….

* 1. ***Đánh giá kết quả thử nghiệm***

Chương trình thực hiện được các yêu cầu đề ra của đề tài thực tập tốt nghiệp.

Ứng dụng Share Food đã phần nào đáp ứng được một số chức năng chính như:

Tìm kiếm món ăn theo tỉnh, tìm kiếm món ăn theo loại món ăn, tìm đến nhà hàng có bán món ăn đó, bình luận món ăn để chia sẻ ý kiến. Hiển thị những món ăn tại nơi mình đang ở qua vị trí định vị hiện tại. Thêm yêu thích.

Admin quản lý dễ dàng nhà hàng, tài khoản, thực đơn.

Chủ nhà hàng quản lý cập nhật thông tin nhà hàng và đồng thời quản lý món ăn của nhà hàng mình.

Giao diện thân thiện, dễ sử dụng.

**KẾT LUẬN**

*Kết quả đạt được*

Sau thời gian thực hiện đề tài, chương trình đã hoàn thành và đạt được một số kết quả sau:

* + Xây dựng thành công ứng dụng món ngon đường phố cho người dùng và các chủ nhà hàng.
  + Tìm hiểu và nắm khá rõ về các công cụ xây dựng ứng dụng Android và Webservice như: Android Studio, MySQL, API, Java Spring.
  + Giao diện của chương trình thân thiện, dễ sử dụng.
  + Ứng dụng đã giúp người dùng tiết kiệm thời gian, công sức để có được một sản phẩm ưng ý. Dễ dàng quản lý thông tin chi tiết người dùng, các thông tin về nhà hàng, món ăn, thực đơn.
  + Thực hiện kiểm thử phần mềm ở mức độ lập trình.

*Hạn chế của đề tài*

* + Chưa thể xử lý được các địa chỉ chi tiết đến từng ngõ trên bản đồ.
  + Server của ứng dụng chưa được thử nghiệm trên mạng Internet.

*Hướng phát triển của đề tài*

Với nỗ lực không ngừng, em đã cố gắng hoàn thành yêu cầu đề tài.

**TÀI LIỆU THAM KHẢO**

[1]. Android MVP for beginners

<https://android.jlelse.eu/android-mvp-for-beginners-25889c500443>

truy cập cuối cùng ngày 20/04/2019

[2].Facebook for developers

<https://developers.facebook.com/docs/facebook-login/android/>

truy cập cuối cùng ngày 21/04/2019

[3].Google Sign-in for Android

<https://developers.google.com/identity/sign-in/android/sign-in>

truy cập cuối cùng ngày 21/04/2019

[4]. Maps SDK for Android

<https://developers.google.com/maps/documentation/android-sdk/start>

truy cập lần cuối ngày 15/04/2019

[5].Trần Duy Thanh, “*Phát triển ứng dụng di động cơ bản*”, Nhà xuất bản ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH, 2019.

[6].Trần Duy Thanh, “*Phát triển ứng dụng di động nâng cao*”, Nhà xuất bản ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH, 2019