

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN



HUỲNH THẾ CƯƠNG- 15110172

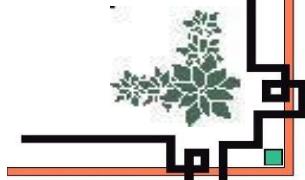
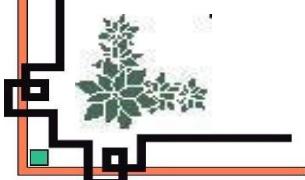
BÙI ĐỖ TRUNG TRỰC - 15110354

ĐỀ TÀI: TÌM HIỂU VÀ XÂY DỰNG ỨNG DỤNG
TÌM KIẾM QUÁN ĂN IOS

KHOÁ LUẬN CHUYÊN NGÀNH CNPM

GVHD: TS. NGUYỄN THIÊN BẢO

KHÓA 2015 - 2019



BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN



HUỲNH THẾ CƯƠNG- 15110172

BÙI ĐỖ TRUNG TRỰC - 15110354

ĐỀ TÀI: TÌM HIỂU VÀ XÂY DỰNG ỨNG DỤNG
TÌM KIẾM QUÁN ĂN IOS

KHOÁ LUẬN CHUYÊN NGÀNH CNPM

GVHD: TS. NGUYỄN THIÊN BẢO

KHÓA 2015 - 2019

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên SV thực hiện 1: Huỳnh Thế Cương

Mã số SV: 15110172

Họ và tên SV thực hiện 2: Bùi Đỗ Trung Trực

Mã số SV: 15110354

Ngành: Công nghệ thông tin

Tên đề tài: Tìm hiểu và xây dựng ứng dụng tìm kiếm quán ăn trên hệ điều hành IOS

Họ và tên giáo viên hướng dẫn: T.S Nguyễn Thiên Bảo

NHẬN XÉT:

1. Về nội dung đề tài và khối lượng công việc thực hiện:

Lý thuyết:

- Tìm hiểu ngôn ngữ Swift, nodejs
- Tìm hiểu về cơ sở dữ liệu mongodb, firebase

Hiện thực:

- Xây dựng ứng dụng trên hệ điều hành IOS về tìm kiếm quán ăn

2. Ưu điểm:

Dễ dàng sử dụng với người dùng

Cho phép tìm kiếm quán ăn theo địa chỉ và tên quán

Lưu các món ăn yêu thích

Chia sẻ quán ăn với nhiều người

2. Khuyết điểm:

Chưa có đánh giá về quán ăn

Chưa có đặt thức ăn và đặt chỗ

Hiện thông báo cố định, chưa có thêm thông báo

Chưa có mời bạn bè

3. Đánh giá loại:

4. Điểm:

TP.HCM, ngày tháng năm 2019

Giáo viên hướng dẫn

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên SV thực hiện 1: Huỳnh Thế Cường

Mã số SV: 15110172

Họ và tên SV thực hiện 2: Bùi Đỗ Trung Trực

Mã số SV: 15110354

Ngành: Công nghệ thông tin

Tên đề tài: Tìm hiểu và xây dựng ứng dụng tìm kiếm quán ăn trên hệ điều hành IOS

Họ và tên giáo viên phản biện: T.S Nguyễn Thành Sơn

NHẬN XÉT:

1. Về nội dung đề tài và khối lượng công việc thực hiện:

.....
.....
.....
.....
.....
.....
.....

2. Ưu điểm:

.....
.....
.....
.....
.....
.....

3. Khuyết điểm:

.....
.....
.....
.....
.....
.....

4. Đánh giá loại:

5. Điểm:

TP.HCM, ngày tháng năm 2019

Giáo viên phản biện

LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của bất kỳ ai khác. Trong suốt thời gian học tập ở trường, em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình và bạn bè.

Em xin chân thành gửi lời cảm ơn đến Khoa Công nghệ thông tin, trường Đại học Sư phạm kỹ thuật Thành phố Hồ Chí Minh đã tạo điều kiện cho em thực hiện đề tài này.

Em xin kính gửi đến thầy Nguyễn Thiên Bảo lời cảm ơn sâu sắc nhất. Trong khoảng thời gian qua, Thầy đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình thực hiện đề tài.

Chân thành cảm ơn tập thể Giáo viên Khoa Công nghệ thông tin, trường Đại học Sư phạm kỹ thuật Thành phố Hồ Chí Minh về những bài giảng trong suốt khóa học, về những kiến thức nền tảng cũng như những kiến thức về chuyên ngành đã giúp cho em hoàn thành tốt đề tài này.

Xin chân thành cảm ơn các anh chị và bạn bè đã ủng hộ, giúp đỡ, động viên em trong thời gian học tập và nghiên cứu.

Em xin chúc Thầy luôn thành công trong sự nghiệp giảng dạy cũng như mọi lĩnh vực trong cuộc sống. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình, chu đáo của Thầy, đến nay em đã hoàn thành được “ứng dụng trên hệ điều hành IOS về tìm kiếm quán ăn”.

Trong phạm vi khả năng cho phép, em đã rất cố gắng để hoàn thành đề tài một cách tốt nhất. Song, chắc chắn sẽ không tránh khỏi những thiếu sót. Em kính mong nhận được sự cảm thông và những ý kiến đóng góp quý báu của Thầy Cô để em học hỏi thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn trong tương lai.

Một lần nữa em xin chân thành cảm ơn!

ĐỀ CƯƠNG KHOÁ LUẬN CHUYÊN NGÀNH

Họ và tên SV thực hiện 1: Huỳnh Thế Cương

Mã số SV: 15110172

Họ và tên SV thực hiện 2: Bùi Đỗ Trung Trực

Mã số SV: 15110354

Thời gian làm tiểu luận: từ: 01/03/2019 đến: 30/06/2019

Chuyên ngành: Công Nghệ Phần Mềm

Tên tiểu luận: Tìm hiểu và xây dựng ứng dụng tìm kiếm quán ăn trên hệ điều hành iOS
GV hướng dẫn: T.S Nguyễn Thiên Bảo

Nhiệm vụ của luận văn:

1. Tìm hiểu ngôn ngữ Swift, Nodejs
2. Tìm hiểu cơ sở dữ liệu MongoDB và Firebase
3. Áp dụng để xây dựng ứng dụng tìm kiếm quán ăn trên hệ điều hành iOS

Đề cương chi tiết luận văn

MỤC LỤC

1. Phần mở đầu
 - 1.1 Sự cần thiết của đề tài
 - 1.2 Mục đích của đề tài
 - 1.3 Cách tiếp cận
 - 1.4 Kết quả dự kiến đạt được
2. Khảo sát hiện trạng
 - 2.1 Các ứng dụng tương tự
 - 2.2 Yêu cầu của ứng dụng
3. Cơ sở lý thuyết
4. Thiết kế hệ thống
 - 4.1 Đặc tả usecase
 - 4.2 Lược đồ sequence
 - 4.3 Thiết kế cơ sở dữ liệu
 - 4.4 Thiết kế giao diện
5. Hiện thực và kiểm thử hệ thống
 - 5.1 Cài đặt
 - 5.2 Kiểm thử
6. Kết luận
 - 6.1 Kết quả đạt được
 - 6.2 Ưu, nhược điểm
 - 6.3 Hướng phát triển của đề tài

KẾ HOẠCH THỰC HIỆN

| STT | Thời Gian | Công Việc | Ghi Chú |
|-----|-----------------|--|---------|
| 1 | 03/03 đến 17/03 | Tìm hiểu về mongodb và các công nghệ liên quan. Lên ý tưởng xây dựng | |
| 2 | 17/03 đến 31/03 | Khảo sát hiện trạng v, xác định yêu cầu, mô tả hệ thống Vẽ Use case | |
| 3 | 31/03 đến 07/04 | Tìm hiểu api của mongodb Xây dựng database Vẽ Sequence Diagram chính | |
| 4 | 07/04 đến 14/04 | Tạo giao diện cho Admin và các chức năng của admin | |
| 5 | 14/04 đến 28/04 | Tạo form đăng nhập, giao diện chính và đưa database lên giao diện | |
| 6 | 28/04 đến 12/05 | Tạo bản đồ cho quán ăn Đưa phần mềm vào điện thoại | |
| 7 | 12/05 đến 19/05 | Đăng kí firebase Đăng nhập và đăng kí trên ứng dụng | |
| 8 | 19/05 đến 26/05 | Tạo chức năng lưu bài viết Thay đổi thông tin cá nhân của User Tạo chức năng cho người dùng đăng bài | |
| 9 | 26/05 đến 09/06 | Tạo chức năng bình luận quán ăn Tạo chức năng tìm kiếm theo tên quán, địa chỉ và xung quanh Hoàn thiện chức năng cho người dùng đăng bài | |
| 10 | 09/06 đến 16/06 | Tạo chức năng chia sẻ quán ăn cho người khác Tạo chức năng thông báo trên điện thoại | |
| 11 | 16/06 đến 23/06 | Chỉnh sửa lại giao diện và kiểm thử | |

Ý kiến của giáo viên hướng dẫn

Ngày tháng năm 20

Người viết đề cương

(ký và ghi rõ họ tên)

Mục Lục

| | |
|--|-----------|
| DANH MỤC HÌNH ẢNH | 10 |
| Danh Mục Bảng | 11 |
| CHƯƠNG 1: PHẦN MỞ ĐẦU..... | 1 |
| 1. TÍNH CẤP THIẾT CỦA ĐỀ TÀI..... | 1 |
| 2. MỤC ĐÍCH CỦA ĐỀ TÀI..... | 1 |
| 3. CÁCH TIẾP CẬN..... | 1 |
| 4. KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC..... | 2 |
| CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG..... | 3 |
| 2.1 CÁC ỨNG DỤNG TƯƠNG TỰ | 3 |
| 2.1.1 Foody:..... | 3 |
| 2.1.2 Lozi:..... | 3 |
| 2.1.3 ClingMe:..... | 3 |
| 2.2 YÊU CẦU CỦA ỨNG DỤNG..... | 4 |
| CHƯƠNG 3: CƠ SỞ LÝ THUYẾT | 6 |
| 3.1 MONGOOSE | 6 |
| 3.1.1 Định nghĩa..... | 6 |
| 3.1.2 Các SchemaTypes của Mongoose:..... | 6 |
| 3.1.3 Mỗi loại SchemaType đều cho phép xác định: | 6 |
| 3.1.4 Thiết lập Mongoose:..... | 7 |
| 3.1.5 Định nghĩa schema cho Mongoose: | 7 |
| 3.1.6 Tạo và lưu những model của Mongoose:..... | 9 |
| 3.1.7 Xác thực dữ liệu trong schema:..... | 11 |
| 3.1.8 Một số thao tác với cơ sở dữ liệu:..... | 13 |
| 3.2 NODEJS | 14 |
| 3.2.1 Định nghĩa: | 14 |
| 3.2.2 Ứng dụng của node.js:..... | 14 |
| 3.2.3 Nhược điểm của node.js:..... | 15 |
| 3.2.4 Ưu điểm của node.js:..... | 15 |
| 3.2.5 Không nên sử dụng node.js khi:..... | 16 |
| 3.2.6 Những trường hợp nên sử dụng node.js: | 17 |
| 3.3 FIREBASE..... | 17 |
| 3.3.1 Định nghĩa | 17 |
| 3.3.2 Các dịch vụ firebase cung cấp: | 17 |
| 3.3.3 Các nhóm công cụ | 18 |
| 3.3.4 Ưu điểm, nhược điểm | 21 |
| 3.3.5 Cách cài đặt | 21 |
| CHƯƠNG 4: THIẾT KẾ HỆ THỐNG HỆ THỐNG | 22 |
| 4.1 ĐẶC TẨU USECASE | 22 |

| | | |
|--|--|-----------|
| 4.1.1 | Mô tả use case Quản lý quán ăn | 24 |
| 4.1.2 | Mô tả use case Quản lý món ăn | 27 |
| 4.1.3 | Mô tả Use Case Quản lý thông tin cá nhân và thay đổi thông tin cá nhân | 28 |
| 4.1.4 | Mô tả Use case Quản Lý Thông tin Quán ăn | 31 |
| 4.2 | LƯỢC ĐỒ SEQUENCE..... | 34 |
| 4.2.1 | Lược đồ Sequence Diagram đăng nhập | 34 |
| 4.2.2 | Lược đồ Sequence Diagram cho Đăng kí..... | 34 |
| 4.2.3 | Sequence Diagram cho sửa cửa hàng | 35 |
| 4.2.4 | Sequence Diagram cho Tìm Kiếm Quán Ăn..... | 35 |
| 4.3 | CHI TIẾT CÁC BẢNG DỮ LIỆU | 36 |
| 4.4 | THIẾT KẾ GIAO DIỆN..... | 40 |
| 4.4.1 | Màn hình giao diện và xử lý bên Admin | 40 |
| 4.4.1.1 | <i>Chức năng nghiệp vụ của Admin</i> | 40 |
| 4.4.1.2 | <i>Màn hình giao diện.....</i> | 41 |
| 4.4.2 | Màn hình và xử lý bên Guest..... | 47 |
| 4.4.2.1 | <i>Chức năng nghiệp vụ của Guest</i> | 47 |
| 4.4.2.2 | <i>Màn hình giao diện.....</i> | 47 |
| 4.4.3 | Màn hình và xử lý của User | 51 |
| 4.4.3.1 | <i>Chức năng nghiệp vụ của User:</i> | 51 |
| 4.4.3.2 | <i>Màn hình giao diện.....</i> | 52 |
| CHƯƠNG 5: HIỆN THỰC VÀ KIỂM THỬ HỆ THỐNG..... | 59 | |
| 5.1 | CÀI ĐẶT..... | 59 |
| 5.2 | KIỂM THỬ | 59 |
| CHƯƠNG 6: PHẦN KẾT LUẬN | 63 | |
| 6.1 | KẾT QUẢ ĐẠT ĐƯỢC | 63 |
| 6.2 | ƯU , NHƯỢC ĐIỂM | 63 |
| 6.2.1 | Ưu điểm..... | 63 |
| 6.2.2 | Nhược điểm:..... | 63 |
| 6.3 | HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI..... | 63 |
| TÀI LIỆU THAM KHẢO..... | 64 | |
| HƯỚNG DẪN CÀI ĐẶT | 65 | |

DANH MỤC HÌNH ẢNH

| | |
|---|----|
| Hình 1 Giao diện foody..... | 3 |
| Hình 2 Giao diện ứng dụng Lozi..... | 3 |
| Hình 3 Giao diện ứng dụng ClingMe..... | 4 |
| Hình 4 Các nhóm công cụ Firebase..... | 18 |
| Hình 5 Usecase hệ thống | 22 |
| Hình 6 Usecase Quan ly quan an | 23 |
| Hình 7 Usecase Quan ly thuc an | 23 |
| Hình 8 Usecase Quản lý thông tin cá nhân | 23 |
| Hình 9 Usecase Thay đổi thông tin cá nhân | 24 |
| Hình 10 Usecase Quan ly thong tin quan an | 24 |
| Hình 11 Sequence Diagram cho Đăng nhập | 34 |
| Hình 12 Sequence Diagram cho Đăng Kí..... | 35 |
| Hình 13 Sequence Diagram cho Sửa Cửa Hàng | 35 |
| Hình 14 Sequence Diagram cho Tìm Kiếm Quán Ăn | 36 |
| Hình 15 Sequence Diagram cho Lưu Quán Ăn..... | 36 |
| Hình 16 Cơ sở dữ liệu của ứng dụng | 37 |
| Hình 17 Thêm bài viết..... | 41 |
| Hình 18 Sửa bài viết | 42 |
| Hình 19 Xóa bài viết | 43 |
| Hình 20 Thêm thực đơn..... | 44 |
| Hình 21 Sửa thực đơn..... | 45 |
| Hình 22 Xóa thực đơn | 46 |
| Hình 23 Xem bài viết của User | 46 |
| Hình 24 BM_Search | 48 |
| Hình 25 BM_Search_Local | 49 |
| Hình 26 View_Info..... | 50 |
| Hình 27 BM_Login | 50 |
| Hình 28 BM_Register | 51 |
| Hình 29 BM_Save_Info | 53 |
| Hình 30 BM_View_Save | 53 |
| Hình 31 BM_Share_Store | 54 |
| Hình 32 BM_Comments_Store..... | 54 |
| Hình 33 BM_Post_Store | 55 |
| Hình 34 BM_Notification | 55 |
| Hình 35 BM_Edit_Name | 56 |
| Hình 36 BM_Edit_Pass | 56 |
| Hình 37 BM_Edit_Image | 57 |
| Hình 38 BM_Logout | 57 |
| Hình 39 BM_Report_Store | 58 |
| Hình 40 Mở file source code | 65 |
| Hình 41 Hình Chạy ứng dụng | 65 |
| Hình 42 Lỗi chạy ứng dụng lần đầu tiên | 65 |
| Hình 43 sửa lỗi bước 1..... | 66 |
| Hình 44 Sửa lỗi bước 2 | 66 |
| Hình 45 Sửa lỗi bước 3 | 67 |
| Hình 46 Sửa lỗi bước 4 | 67 |
| Hình 47 sửa lỗi bước 5..... | 68 |
| Hình 48 sửa lỗi bước 6 | 68 |

Danh Mục Bảng

| | |
|---|----|
| Bảng 1 Usecase Them quan an | 25 |
| Bảng 2 Usecase Sua quan an | 25 |
| Bảng 3 Usecase Xoa quan an | 26 |
| Bảng 4 Usecase Phe duyet quan an..... | 26 |
| Bảng 5 UseCase Nhan bao cao quan an | 27 |
| Bảng 6 Usecase Them mon an | 27 |
| Bảng 7 Usecase Sua mon an..... | 28 |
| Bảng 8 Usecase Xoa mon an | 28 |
| Bảng 9 Usecase Xem thong tin ca nhân..... | 29 |
| Bảng 10 Usecase Chinh sua ten ca nhan | 29 |
| Bảng 11 Usecase Chinh sua mat khau..... | 30 |
| Bảng 12 Usecase Chinh sua anh dai dien..... | 30 |
| Bảng 13 Usecase Xem thong tin quan an | 31 |
| Bảng 14 Usecase Luu quan an yeu thich | 31 |
| Bảng 15 UseCase Tim kiem quan an | 32 |
| Bảng 16 UseCase Dang tai khoan | 32 |
| Bảng 17 UseCase Binh luan quan an | 33 |
| Bảng 18 UseCase Bao cao quan an..... | 33 |
| Bảng 19 Bảng quán ăn poster của Mongoddb..... | 37 |
| Bảng 20 Bảng MoreContents của Mongodbb | 38 |
| Bảng 21 Bảng comments trong Mongodbb | 39 |
| Bảng 22 Bảng User trong Firebase | 39 |
| Bảng 23 Bảng SaveStore trong Firebase | 40 |
| Bảng 24 nghiệp vụ của admin | 40 |
| Bảng 25 Nghiệp vụ của Guest | 47 |
| Bảng 26 Nghiệp vụ của User | 51 |
| Bảng 27 TestCase ADMIN | 59 |
| Bảng 28 TestCase User | 60 |
| Bảng 29 TestCase GUEST | 62 |

CHƯƠNG 1: PHẦN MỞ ĐẦU

1.1 TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Nhu cầu đi tìm kiếm quán ăn của con người ngày càng cao và các bạn trẻ thì thích các quán ngon bổ rẻ. Vì vậy, ứng dụng của đề tài giúp khách hàng tìm kiếm các quán ăn và lưu lại các quán ăn mà mình thích, với giao diện quen thuộc với người dùng. Để đáp ứng nhu cầu và một lượng lớn các quán ăn, các dữ liệu chúng ta cần phải có một cơ sở dữ liệu lớn để

1.2 MỤC ĐÍCH CỦA ĐỀ TÀI

- Tìm hiểu về ngôn ngữ Swift, NodeJS
- Tìm hiểu về công nghệ Firebase và mongodb
- Xây dựng ứng dụng tìm kiếm quán ăn trên hệ điều hành iOS

1.3 CÁCH TIẾP CẬN

- Tạo Form đăng nhập với mongodb trên xcode.
- Viết api để kết nối từ mongodb lên app điện thoại.
- Thiết kế database để tạo bản bài viết hiển thị trên app điện thoại.
- Dùng textflight để có thể chạy app lên điện thoại thật.
- Dựng vps và domain sau đó đưa server lên để điện thoại thật có thể nhận được dữ liệu.
- Tạo bản đồ và định vị chỉ đường cho người dùng.
- Kết nối Firebase với app.
- Tạo tài khoản với Firebase và database Realtime
- Đưa dữ liệu từ FirebaseDatabase sang Mongoddb.
- Tạo chức năng đăng ký, đăng nhập bằng Firebase.
- Tạo các chức năng cho người dùng đã đăng ký tài khoản:
 - Lưu bài viết.
 - Xem bài viết đã lưu.
 - Bỏ lưu bài viết.
 - Nhận thông báo từ admin

- Thay đổi tên hiển thị
- Thay đổi mật khẩu.
- Thay đổi avatar.
- Bình luận bài viết.
- Chia sẻ bài viết
- Đăng bài viết
- Báo cáo về quán ăn

1.4 KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC

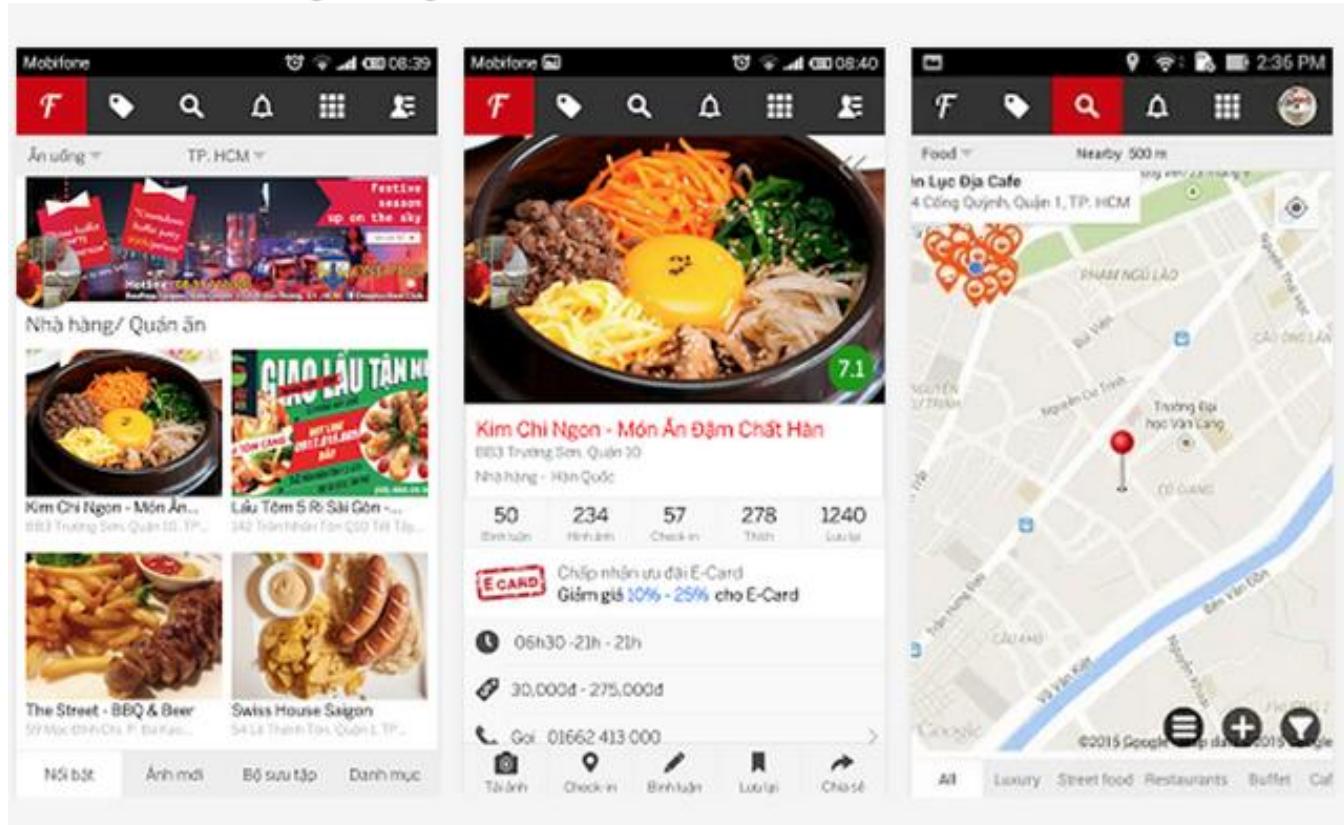
- Tạo ra ứng dụng tìm kiếm quán ăn.
 - Giúp người dùng xem được những thông tin cần thiết về quán ăn.
 - Giúp người dùng lưu lại những địa điểm mình đánh dấu.
 - Chia sẻ các quán ăn.
 - Đăng các bài về quán ăn muốn chia sẻ
 - Báo cáo về quán ăn trên ứng dụng
- Có kiến thức về các ngôn ngữ mới: Swift, nodeJs.
- Có kiến thức về các công nghệ database NoSQL mới: Firebase, mongodb.

CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG

2.1 CÁC ỨNG DỤNG TƯƠNG TỰ

2.1.1 Foody:

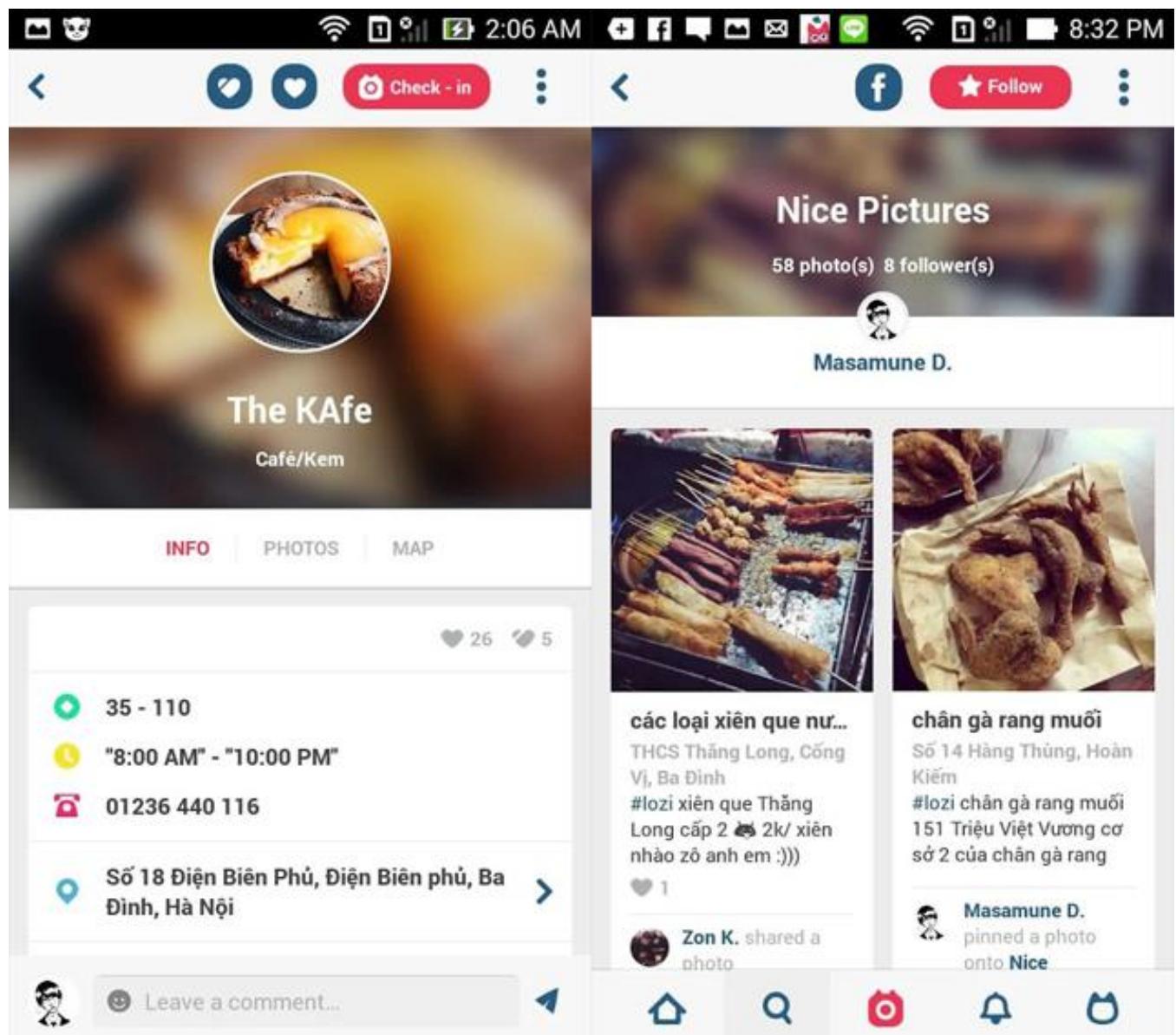
Ân tượng đầu tiên của Foody nằm ở giao diện ứng dụng được đầu tư khá tốt cũng như trải nghiệm người dùng. Điểm mạnh của Foody là không chỉ hỗ trợ giao diện web mà nó còn hỗ trợ đa nền tảng di động như iOS, Android và Windows Phone.



Hình 1 Giao diện foody

2.1.2 Lozi:

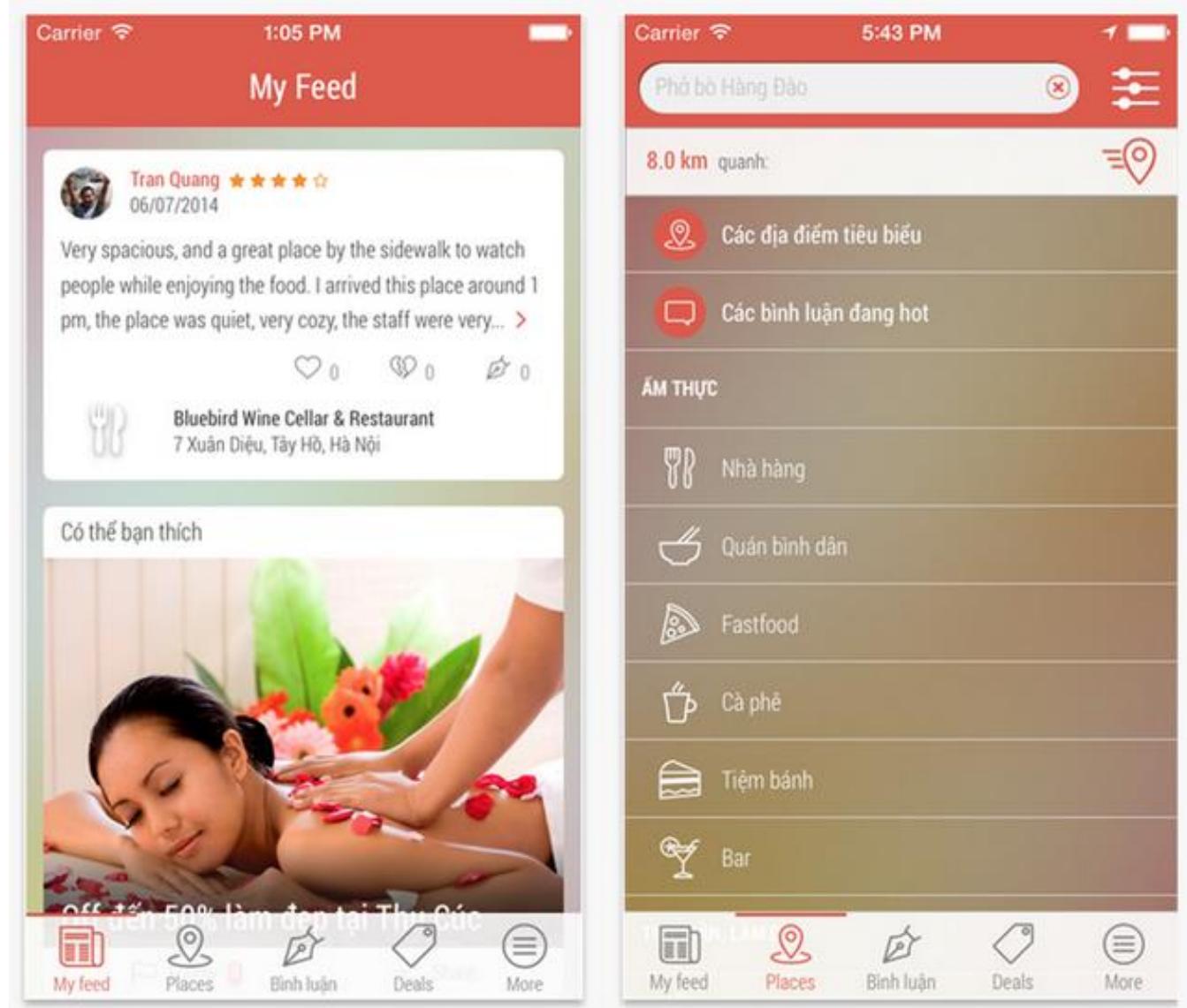
Lozi là một mạng xã hội về ăn uống, chú trọng vào chia sẻ ảnh món ăn. Ân tượng đầu tiên của người dùng đó là hình ảnh rất đẹp, sở dĩ Lozi có được những hình ảnh đẹp này là vì Lozi xây dựng cộng đồng ban đầu từ Instagram. Và cũng như tính năng của các mạng xã hội khác, người dùng sẽ chia sẻ ảnh món ăn và review của họ cũng như theo dõi bạn bè của họ.



Hình 2 Giao diện ứng dụng Lozi

2.1.3 ClingMe:

Một ứng dụng Việt thông minh với nhiều tính năng độc đáo. Ngoài việc trợ giúp tìm những địa điểm xung quanh, đưa ra gợi ý các hàng quán gần và phù hợp nhu cầu người dùng nhất dựa trên GPS, ứng dụng còn hỗ trợ tìm kiếm các cây xăng, ATM, bến xe và cả tiệm thuốc ở khắp đất nước Việt Nam. ClingMe có giao diện khá trực quan, cách bố trí hợp lý giúp người dùng dễ làm quen ngay ở lần đầu tiên.



Hình 3 Giao diện ứng dụng ClingMe

2.2 YÊU CẦU CỦA ÚNG DỤNG

Ở Việt Nam, trong những năm gần đây, việc ứng dụng tin học vào tìm kiếm quán ăn khá phổ biến. Với những phần mềm tìm kiếm quán ăn như vậy, nó tìm kiếm các quán ăn ở gần, các quán ăn ngon để mình có thể lựa chọn theo món.

Ưu điểm:

- Tìm kiếm nhanh các quán ăn gần.
- Tìm kiếm các món ăn, quán ăn theo tên
- Lưu các quán lại để có thể tìm kiếm nhanh hơn những gì mình thích

Khuyết điểm:

- Được phát triển riêng cho các nhà hàng, quán cafe – trà sữa, do đó phần mềm chưa tư vấn được các cửa hàng ở nhiều lĩnh vực

Chức năng có thể có trong ứng dụng:

❖ Guest :

Xem thông tin quán ăn

- Xem tất cả các quán ăn: tên quán, địa chỉ, thời gian mở cửa, giá cả.
- Xem bản đồ chỉ đường.
- Xem thông tin quảng cáo, khuyến mãi.
- Chia sẻ quán ăn

Đăng nhập.

Đăng ký tài khoản

Tìm kiếm:

- Theo tên quán ăn
- Theo địa chỉ quán ăn

❖ User:

Xem thông tin quán ăn:

- Xem tất cả các quán ăn: tên quán, địa chỉ, thời gian mở cửa, giá cả.
- Xem bản đồ chỉ đường.
- Xem thông tin quảng cáo, khuyến mãi.
- Bình luận về quán ăn.
- Chia sẻ quán ăn
- Báo cáo về quán ăn cho Admin

Lưu lại thông tin quán ăn yêu thích

Tìm kiếm:

- Theo tên quán ăn
- Theo địa chỉ quán ăn

Chỉnh sửa thông tin cá nhân: thay đổi tên hiển thị, thay đổi mật khẩu, đổi hình avatar

- Nhận thông báo từ admin

❖ Admin

- Thêm xóa sửa: Quán ăn, thực đơn quán ăn.
- Gửi thông báo cho người dùng.
- Phê duyệt bài viết của User muốn chia sẻ quán ăn
- Nhận báo cáo về quán ăn của User

CHƯƠNG 3: CƠ SỞ LÝ THUYẾT

3.1 MONGOOSE

3.1.1 Định nghĩa

Mongoose là một Object Document Mapper (ODM). Điều này có nghĩa là Mongoose cho phép bạn định nghĩa các object (đối tượng) với một schema được định nghĩa rõ ràng, được ánh xạ tới một MongoDB document.

3.1.2 Các SchemaTypes của Mongoose:

- String
- Number
- Date
- Buffer
- Boolean
- Mixed
- ObjectId
- Array

3.1.3 Mỗi loại SchemaType đều cho phép xác định:

- Giá trị mặc định
- Hàm xác thực
- Cho biết một trường bắt buộc
- Hàm get cho phép bạn thao tác dữ liệu trước khi nó được trả về như một object
- Hàm set cho phép bạn thao tác dữ liệu trước khi nó được lưu vào cơ sở dữ liệu
- Tạo indexes cho phép dữ liệu được nạp nhanh hơn
- Ngoài các tùy chọn phổ biến này, một số kiểu dữ liệu nhất định cho phép bạn tùy chỉnh thêm cách dữ liệu được lưu trữ và truy xuất từ một database. Ví dụ, một kiểu dữ liệu String cũng cho phép bạn chỉ định các tùy chọn bổ sung sau đây:
 - Chuyển đổi nó thành chữ thường
 - Chuyển đổi nó thành chữ in hoa
 - Cắt dữ liệu trước khi lưu lại
 - Một biểu thức chính quy có thể giới hạn dữ liệu được phép lưu trong quá trình xác thực

- Một enum có thể định nghĩa một danh sách các string (chuỗi) hợp lệ
- Các thuộc tính Number and Date đều hỗ trợ chỉ định một giá trị tối thiểu và tối đa cho phép cho field đó.
- Bạn nên quen thuộc với hầu hết 8 kiểu dữ liệu. Tuy nhiên, có một số ngoại lệ có thể xảy ra với bạn, chẳng hạn như Buffer, Mixed, ObjectId và Array.
- Kiểu dữ liệu Buffer cho phép bạn lưu dữ liệu nhị phân. Một ví dụ phổ biến về dữ liệu nhị phân sẽ là hình ảnh hoặc tệp được mã hóa, chẳng hạn như tài liệu PDF.
- Kiểu data Mixed biến thuộc một tính thành một field "anything goes" (thứ gì cũng được). Field này so sánh nhà phát triển có thể sử dụng MongoDB ra sao vì nó không có cấu trúc xác định. Hãy cảnh giác với việc sử dụng kiểu dữ liệu này vì bạn sẽ mất nhiều tính năng tuyệt vời mà Mongoose cung cấp, chẳng hạn như xác thực dữ liệu và phát hiện các thay đổi của entity để tự động biết để cập nhật thuộc tính khi lưu.
- Kiểu dữ liệu ObjectId thường chỉ định một liên kết đến một tài liệu khác trong database của bạn. Ví dụ: nếu bạn có một bộ collection về books (sách) và author (tác giả), book document có thể chứa thuộc tính ObjectId có tham chiếu đến một author cụ thể của document.
- Kiểu dữ liệu Array cho phép bạn lưu trữ các mảng giống như JavaScript. Với kiểu dữ liệu Array, bạn có thể thực hiện các chức năng phổ biến của JavaScript array, chẳng hạn như push, pop, shift, slice, v.v.

3.1.4 Thiết lập Mongoose:

- Gói mongoose được cài đặt như sau:

```
npm install mongoose --save
```

- Kết nối với MongoDB:

```
var mongoose = require('mongoose');  
mongoose.connect('mongodb://localhost/mongoose_basics');
```

Mongoose bây giờ được thiết lập và kết nối với một cơ sở dữ liệu
được gọi là `mongoose_basics`

3.1.5 Định nghĩa schema cho Mongoose:

- Ví dụ: Tạo đối tượng user có chứa hai thuộc tính: firstName và lastName:

```
var userSchema = mongoose.Schema({  
    firstName: String,  
    lastName: String  
});
```

Mở rộng ví dụ này bằng cách chuyển đổi các thuộc tính tên và họ thành các đối tượng con của thuộc tính name. Thuộc tính name sẽ bao gồm cả tên và họ. Thêm một thuộc tính created theo kiểu Date.

```
var userSchema = mongoose.Schema({  
    name: {  
        firstName: String,  
        lastName: String  
    },  
    created: Date  
});
```

- Tham chiếu giữa các schema trong MongoDB:
Tạo 2 schema: authorSchema và bookSchema:

```
var authorSchema = mongoose.Schema({  
    _id: mongoose.Schema.Types.ObjectId,  
    name: {  
        firstName: String,  
        lastName: String  
    },  
    biography: String,  
    twitter: String,  
    facebook: String,  
    linkedin: String,  
    profilePicture: Buffer,  
    created: {  
        type: Date,  
        default: Date.now  
    }  
});
```

```
var bookSchema = mongoose.Schema({  
    _id: mongoose.Schema.Types.ObjectId,  
    title: String,  
    summary: String,  
    isbn: String,  
    thumbnail: Buffer,
```

```
author: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Author'
},
ratings: [
    {
        summary: String,
        detail: String,
        numberOfStars: Number,
        created: {
            type: Date,
            default: Date.now
        }
    }
],
created: {
    type: Date,
    default: Date.now
}
});
```

Lược đồ `book` chứa một số thuộc tính của kiểu `String`. Như đã đề cập ở trên, nó chứa một tham chiếu đến lược đồ `author`. Để minh họa thêm các định nghĩa schema mạnh mẽ, schema `book` cũng chứa một `Array` các `ratings`. Mỗi xếp hạng bao gồm `summary`, `detail`, `numberOfStars`, và `createddate` (ngày tạo ra).

Mongoose cho phép bạn tạo ra các schema có tham chiếu đến các schema khác, hoặc như trong ví dụ trên với thuộc tính `ratings`, nó cho phép bạn tạo một `Array` các thuộc tính con có thể chứa trong một schema có liên quan (như `book` đến `author`) hoặc nội tuyến như trong ví dụ trên (với `book` cho một `Array ratings`).

3.1.6 Tạo và lưu những model của Mongoose:

Với ví dụ ở trên, khi các schema của `author` và `book` chứng minh tính linh hoạt của Mongoose, chúng ta sẽ tiếp tục sử dụng các schema đó và lấy ra một model `Author` và `Book` từ chúng.

```
var Author = mongoose.model('Author', authorSchema);
var Book = mongoose.model('Book', bookSchema);
```

Một Model của Mongoose, khi được lưu lại, tạo một Document trong MongoDB với các thuộc tính như đã được định nghĩa từ schema mà nó bắt nguồn.

Tiếp theo sẽ là lưu các objects vào trong các Model đã được định nghĩa bằng cách sử dụng phương thức `save` của Model.:

```
var jamieAuthor = new Author {
  _id: new mongoose.Types.ObjectId(),
  name: {
    firstName: 'Jamie',
    lastName: 'Munro'
  },
  biography: 'Jamie is the author of ASP.NET MVC 5 with Bootstrap and Knockout.js.',
  twitter: 'https://twitter.com/',
  facebook: 'https://www.facebook.com/'
};
```

```
jamieAuthor.save(function(err) {
  if (err) throw err;

  console.log('Author successfully saved.');
```

```
var mvcBook = new Book {
  _id: new mongoose.Types.ObjectId(),
  title: 'ASP.NET MVC 5 with Bootstrap and Knockout.js',
  author: jamieAuthor._id,
  ratings:[{
    summary: 'Great read'
  }]
};

mvcBook.save(function(err) {
```

```
if (err) throw err;  
console.log('Book successfully saved.');//  
});  
  
var knockoutBook = new Book {  
    _id: new mongoose.Types.ObjectId(),  
    title: 'Knockout.js: Building Dynamic Client-Side Web  
    Applications',  
    author: jamieAuthor._id  
};  
  
knockoutBook.save(function(err) {  
    if (err) throw err;  
  
    console.log('Book successfully saved.');//  
});  
});
```

3.1.7 Xác thực dữ liệu trong schema:

Ví dụ:

```
var authorSchema = mongoose.Schema({  
    _id: mongoose.Schema.Types.ObjectId,  
    name: {  
        firstName: {  
            type: String,  
            required: true  
        },  
        lastName: String  
    },
```

```
biography: String,  
twitter: {  
    type: String,  
    validate: {  
        validator: function(text) {  
            return text.indexOf('https://twitter.com/') === 0;  
        },  
        message: 'Twitter handle must start with https://twitter.com/'  
    }  
},  
facebook: {  
    type: String,  
    validate: {  
        validator: function(text) {  
            return text.indexOf('https://www.facebook.com/') === 0;  
        },  
        message: 'Facebook must start with https://www.facebook.com/'  
    }  
},  
linkedin: {  
    type: String,  
    validate: {  
        validator: function(text) {  
            return text.indexOf('https://www.linkedin.com/') === 0;  
        },  
        message: 'LinkedIn must start with https://www.linkedin.com/'  
    }  
},
```

```
profilePicture: Buffer,  
created: {  
    type: Date,  
    default: Date.now  
}  
});
```

3.1.8 Một số thao tác với cơ sở dữ liệu:

- Tìm kiếm:

Giới thiệu về Mongoose sẽ không hoàn chỉnh nếu không có ví dụ về tìm kiếm bản ghi (record) và cập nhật một hoặc nhiều thuộc tính trên đối tượng đó.

Mongoose cung cấp một số chức năng khác nhau để tìm dữ liệu cho một Model cụ thể. Các hàm find, findOne và findById.

Ví dụ 1: Hàm find và findOne đều chấp nhận một object làm đầu vào cho phép các tìm kiếm phức tạp. tìm tất cả các sách chứa string "mvc" trong tiêu đề.

```
Book.find({  
    title: /mvc/i  
}).exec(function(err, books) {  
    if (err) throw err;
```

```
    console.log(books);  
});
```

Việc gọi hàm find cũng được liên kết với các phương thức truy vấn khác, chẳng hạn như where, and, or, limit, sort, any, v.v.

Sau khi áp dụng hàm find, thứ tự của các hàm khác không quan trọng vì tất cả các hàm có kết nối được biên dịch cùng nhau thành một truy vấn duy nhất và không được thực hiện cho đến khi hàm exec được gọi.

Ví dụ 2: truy vấn một author cụ thể bằng `_id` của họ.

Hàm `findById` được thực hiện hơi khác một chút với hàm `find`. Hàm này thực hiện ngay lập tức và chấp nhận một hàm callback, thay vì cho phép một chuỗi kết nối các hàm.

```
Author.findById('59b31406beefa1082819e72f', function(err, author) {  
    if (err) throw err;
```

```
    console.log(author);
  });
- Tìm kiếm và cập nhật:  
Mongoose cũng cung cấp hai hàm bổ sung để tìm kiếm một đối tượng  
và lưu lại cùng lúc với các hàm được đặt tên khá phù  
hợp: findByIdAndUpdate và findOneAndUpdate
```

Ví dụ:

```
Author.findByIdAndUpdate('59b31406beefa1082819e72f',
  { linkedin: 'https://www.linkedin.com/in/cnpm' },
  function(err, author) {
  if (err) throw err;

  console.log(author);
});
```

3.2 NODEJS

3.2.1 Định nghĩa:

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 bằng 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

Mỗi 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).

3.2.2 Ứng dụng của node.js:

- 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

3.2.3 Nhược điểm của node.js:

- Ứng dụng nặng tốn tài nguyên Nếu bạn cần xử lý các ứng dụng tốn tài nguyên CPU như encoding video, convert file, decoding encryption... hoặc các ứng dụng tương tự như vậy thì không nên dùng NodeJS (Lý do: NodeJS được viết bằng C++ & Javascript, nên phải thông qua thêm 1 trình biên dịch của NodeJS sẽ lâu hơn 1 chút). Trường hợp này bạn hãy viết 1 Addon C++ để tích hợp với NodeJS để tăng hiệu suất tối đa !
- NodeJS và ngôn ngữ khác NodeJS, PHP, Ruby, Python .NET ... thì việc cuối cùng là phát triển các App Web. NodeJS mới sơ khai như các ngôn ngữ lập trình khác. Vậy nên bạn đừng hi vọng NodeJS sẽ không hơn PHP,Ruby,Python... ở thời điểm này. Nhưng với NodeJS bạn có thể có 1 ứng dụng như mong đợi, điều đó là chắc chắn !

3.2.4 Ưu điểm của node.js:

- 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. Ngoài ra, 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) Nếu bạn định viết 1 ứng dụng thể hiện trên 1 trang (Gmail?) NodeJS rất phù hợp để làm. 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 sử 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, ...

3.2.5 Không nên sử dụng node.js khi:

- Xây dựng các ứng dụng hao tốn tài nguyên: Bạn đừng mơ mộng đến Node.js khi bạn đang muốn viết một chương trình convert video. Node.js hay bị rơi vào trường hợp thắt cổ chai khi làm việc với những file dung lượng lớn.
- Một ứng dụng chỉ toàn CRUD: Node.js không nhanh hơn PHP khi bạn làm các tác vụ mang nặng tính I/O như vậy. Ngoài ra, với sự ổn định lâu dài của các webserver script khác, các tác vụ CRUD của nó đã được tối ưu hóa. Còn Node.js? Nó sẽ lòi ra những API cực kỳ kỳ ngớ ngẩn.
- Khi bạn cần sự ổn định trong ứng dụng của bạn: Chỉ với 4 năm phát triển của mình (2009-2013), version của Node.js đã là 0.10.15 (hiện tại tới thời điểm này là v0.10.35). Mọi API đều có thể thay đổi – một cách không tương thích ngược – hãy thật cẩn thận với những API mà bạn đang dùng, và luôn đặt câu hỏi: “Khi nó thay đổi, nó sẽ ảnh hưởng gì đến dự án của tôi?”
- Và quan trọng nhất: Bạn chưa hiểu hết về Node.js Node.js cực kỳ nguy hiểm trong trường hợp này, bạn sẽ rơi vào một thế giới đầy rẫy cạm bẫy, khó khăn. Với phần lớn các API hoạt động theo phương thức non-blocking/async việc không hiểu rõ vấn đề sẽ làm cho việc xuất hiện những error mà thậm chí bạn không biết nó xuất phát từ đâu? Và tệ hơn nữa: Khi cộng đồng Node.js chưa đủ

lớn mạnh, và sẽ ít có sự support từ cộng đồng. Khi mà phần lớn cộng đồng cũng không khá hơn bạn là bao.

3.2.6 Những trường hợp nên sử dụng node.js:

- 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. Chẳng còn gì thích hợp hơn Node.js trong trường hợp này!
- Những ứng dụng đòi hỏi các giao thức kết nối khác chứ không phải chỉ có http. Với việc hỗ trợ giao thức tcp, từ nó bạn có thể xây dựng bất kỳ một giao thức custom nào đó một cách dễ dàng.
- Những ứng dụng thời gian thực: Không phải nói vì Node.js dường như sinh ra để làm việc này!
- 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 bạn có thể lưu cũng như chia sẻ trạng thái của một client với các client khác ngay trong ngôn ngữ, chứ bạn không cần thông qua các bộ nhớ ngoài!

3.3 FIREBASE

3.3.1 Định nghĩa

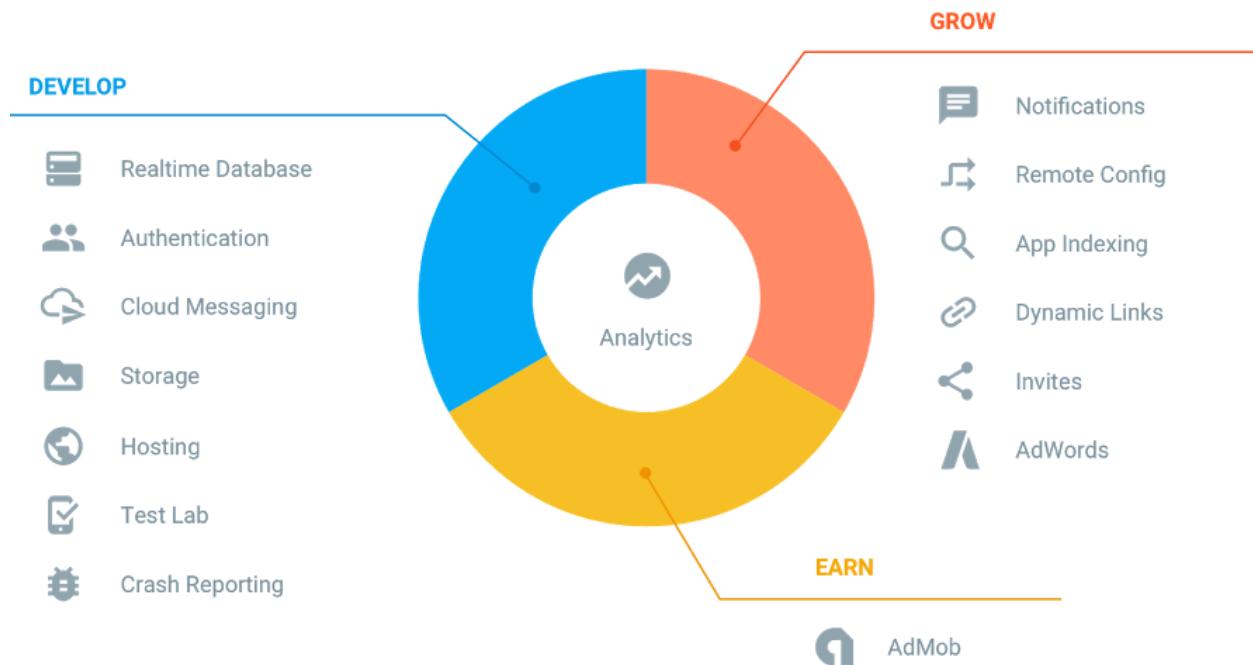
Firebase là một dịch vụ hệ thống backend được Google cung cấp sẵn cho ứng dụng Mobile của bạn, với Firebase bạn có thể rút ngắn thời gian phát triển, triển khai và thời gian mở rộng quy mô của ứng dụng mobile mình đang phát triển. Hỗ trợ cả 2 nền tảng Android và IOS, Firebase mạnh mẽ, đa năng, bảo mật và là dịch vụ cần thiết đầu tiên để xây dựng ứng dụng với hàng triệu người sử dụng.

3.3.2 Các dịch vụ firebase cung cấp:

Firebase cung cấp cho chúng ta công cụ Firebase Analytics và 2 nhóm sản phẩm chính tập trung vào 2 đối tượng là:

- ❖ Develop & test your app: phát triển và kiểm thử các ứng dụng được thiết kế.
- ❖ Grow & engage your audience: Phân tích dữ liệu và tối ưu hóa trải nghiệm đối với người dùng.

3.3.3 Các nhóm công cụ



Hình 4 Các nhóm công cụ Firebase

- ❖ Công cụ Develop & test your app
 - Realtime database: Lưu trữ và đồng bộ dữ liệu người dùng thời gian thực, các ứng dụng hỗ trợ tính năng này có thể lưu trữ và lấy dữ liệu từ máy chủ trong tích tắc. Các dữ liệu được lưu trữ trong hệ thống cơ sở dữ liệu hỗ trợ NoSQL và được đặt trên nền tảng máy chủ Cloud, dữ liệu được ghi và đọc với thời gian thấp nhất tính bằng mili giây. Nền tảng này hỗ trợ đồng bộ hóa dữ liệu của người dùng kể cả khi không có kết nối mạng, tạo nên trải nghiệm xuyên suốt bất chấp tình trạng kết nối internet của người sử dụng. Realtime Database của Firebase hỗ trợ: android, ios, web, c++, unity, và cả xamarin
 - Crashlytics: Hệ thống theo dõi và lưu trữ thông tin lỗi của ứng dụng đang chạy trên máy người dùng. Các thông tin lỗi này được thu thập một cách toàn diện và ngay tức thời. Cách trình bày hợp lý với từng chu trình hoạt động đến khi xảy ra lỗi, các báo cáo trực quan giúp người phát triển có thể nắm bắt và xử lý kịp thời các lỗi chính của ứng dụng.
 - Cloud Firestore: Lưu trữ và đồng bộ dữ liệu giữa người dùng và thiết bị – ở quy mô toàn cầu – sử dụng cơ sở dữ liệu noSQL được lưu trữ trên hạ tầng cloud.

Cloud Firestore cung cấp cho bạn tính năng đồng bộ hóa trực tuyến và ngoại tuyến cùng với các truy vấn dữ liệu hiệu quả. Tích hợp với các sản phẩm Firebase khác cho phép bạn xây dựng các ứng dụng thực sự ngay cả khi kết nối internet bị gián đoạn.

- Authentication: Quản lý người dùng một cách đơn giản và an toàn. Firebase Auth cung cấp nhiều phương pháp để xác thực, bao gồm email và mật khẩu, các nhà cung cấp bên thứ ba như Google hay Facebook, và sử dụng trực tiếp hệ thống tài khoản hiện tại của bạn. Xây dựng giao diện của riêng bạn hoặc tận dụng lợi thế của mã nguồn mở, giao diện người dùng tùy biến hoàn toàn.
- Cloud Functions: Mở rộng ứng dụng của bạn bằng mã phụ trợ tùy chỉnh mà không cần quản lý và quy mô các máy chủ của riêng bạn. Các chức năng có thể được kích hoạt bởi các sự kiện, được phát sinh ra bởi các sản phẩm Firebase, dịch vụ Google Cloud hoặc các bên thứ ba có sử dụng webhooks.
- Cloud Storage: Lưu trữ và chia sẻ nội dung do người dùng tạo ra như hình ảnh, âm thanh và video với bộ nhớ đám mây mạnh mẽ, đơn giản và tiết kiệm chi phí được xây dựng cho quy mô của Google. Các Firebase SDK cho Cloud Storage thêm tính năng bảo mật của Google để tải lên và tải tệp cho các ứng dụng Firebase của bạn, bất kể chất lượng mạng.
- Hosting: Đơn giản hóa lưu trữ web của bạn với các công cụ được thực hiện cụ thể cho các ứng dụng web hiện đại. Khi bạn tải lên nội dung web, chúng tôi sẽ tự động đẩy chúng đến CDN toàn cầu của chúng tôi và cung cấp cho họ chứng chỉ SSL miễn phí để người dùng của bạn có được trải nghiệm an toàn, đáng tin cậy, độ trễ thấp, dù họ ở đâu.
- Test Lab: Chạy thử nghiệm tự động và tùy chỉnh cho ứng dụng của bạn trên các thiết bị ảo và vật lý do Google cung cấp. Sử dụng Firebase Test Lab trong suốt vòng đời phát triển của bạn để khám phá lỗi và sự không nhất quán để bạn có thể cung cấp một trải nghiệm tuyệt vời trên nhiều thiết bị.
- Performance Monitoring: Chẩn đoán các vấn đề về hiệu suất ứng dụng xảy ra trên thiết bị của người dùng của bạn. Sử dụng dấu vết để theo dõi hiệu suất của các phần cụ thể trong ứng dụng của bạn và xem chế độ xem tổng hợp trong bảng điều khiển Firebase. Luôn cập nhật thời gian khởi động của ứng dụng và theo dõi các yêu cầu HTTP mà không cần viết bất kỳ mã nào.

Công cụ Grow & engage your audience

- Google Analytics: Phân tích thuộc tính và hành vi của người dùng trong một bảng điều khiển đơn để đưa ra các quyết định sáng suốt về lộ trình sản phẩm của bạn.

Nhận thông tin chi tiết về thời gian thực từ báo cáo hoặc xuất dữ liệu sự kiện thông qua bạn tới Google BigQuery để phân tích tùy chỉnh.

- Cloud Messaging: Gửi tin nhắn và thông báo cho người dùng qua các nền tảng Android, iOS và web một cách miễn phí. Bạn có thể gửi tin nhắn đến các thiết bị, nhóm thiết bị hoặc các chủ đề hoặc phân đoạn người dùng cụ thể. Nhắn tin đám mây Firebase (FCM) thậm chí là các ứng dụng lớn nhất, cung cấp hàng trăm tys thu mỗi ngày.
- Predictions: Firebase Predictions áp dụng deep learning máy học với dữ liệu phân tích của bạn để tạo các nhóm người dùng năng động dựa trên hành vi dự đoán. Các nhóm người dùng này có thể được sử dụng để nhắm mục tiêu trong các sản phẩm khác như thông báo, Cấu hình từ xa và nhiều hơn nữa.
- Dynamic Links: Sử dụng Liên kết động để cung cấp trải nghiệm người dùng tùy chỉnh cho iOS, Android và web. Bạn có thể sử dụng chúng để hỗ trợ web di động để thúc đẩy chuyển đổi ứng dụng gốc, người dùng chia sẻ người dùng, các chiến dịch xã hội và tiếp thị và hơn thế nữa. Dynamic Links cung cấp cho bạn các thuộc tính bạn cần để hiểu rõ hơn về tăng trưởng di động của bạn. Đây là dịch vụ thay thế goo.gl của Google.
- Remote Config: Tùy chỉnh cách ứng dụng của bạn hiển thị cho mỗi người dùng. Thay đổi giao diện, triển khai các tính năng dần dần, chạy thử nghiệm A / B, cung cấp nội dung tùy chỉnh cho người dùng nhất định hoặc thực hiện các cập nhật khác mà không cần triển khai một phiên bản mới-tất cả từ bảng điều khiển Firebase. Giám sát tác động của những thay đổi của bạn và thực hiện các điều chỉnh chỉ trong vài phút.
- Invites: Cho phép người dùng chia sẻ tất cả các khía cạnh của ứng dụng của bạn, từ mã giới thiệu đến nội dung yêu thích, qua email hoặc SMS. Giải pháp out-of-the-box này hoạt động với Google Analytics for Firebase, để bạn biết khi người dùng mở hoặc cài đặt một ứng dụng qua lời mời.
- App Indexing: Thu hút lại người dùng bằng các ứng dụng đã cài đặt của họ với tích hợp Google Tìm kiếm này. Nếu người dùng có ứng dụng của bạn và họ tìm kiếm nội dung có liên quan, họ có thể khởi chạy nó trực tiếp từ kết quả. Nếu người dùng chưa có ứng dụng của bạn, một thẻ cài đặt sẽ xuất hiện khi họ tìm kiếm các ứng dụng tương tự.
- AdMob: Kiếm tiền bằng cách hiển thị quảng cáo hấp dẫn cho khán giả toàn cầu. AdMob có tất cả những gì bạn cần để thực hiện chiến lược kiếm tiền trên lớp bậc nhất và để tối đa hóa doanh thu do mỗi người dùng tạo ra. Nó có thể được điều

chính cho ứng dụng của bạn, và API của nó được xây dựng để tích hợp các định dạng quảng cáo phong phú một cách dễ dàng.

- AdWords: Có được thông tin và nút kéo lại người dùng với khả năng của Google. Bạn có thể chạy quảng cáo trên Tìm kiếm, hiển thị và video cũng như nhắm mục tiêu phân khúc người dùng cụ thể mà bạn xác định trong Google Analytics for Firebase. Cải thiện nhắm mục tiêu quảng cáo và tối ưu hóa hiệu suất chiến dịch của bạn. Tóm lại: Hiện project của mình chỉ tích hợp tính năng Cloud Messaging gửi tin nhắn và thông báo cho người dùng qua các nền tảng Android, iOS. Nên chúng ta sẽ tìm hiểu cách tích hợp Firebase vào project của mình.

3.3.4 Ưu điểm, nhược điểm

- **Ưu điểm:**

Cung cấp chức năng như phân tích, cơ sở dữ liệu, báo cáo hoạt động và báo cáo các sự cố lỗi để bạn có thể dễ dàng phát triển, định hướng ứng dụng của mình vào người sử dụng nhằm đem lại các trải nghiệm tốt nhất cho họ.

Được google hỗ trợ và cung cấp trên nền tảng phần cứng với quy mô rộng khắp thế giới, được các tập đoàn lớn và các ứng dụng với triệu lượt sử dụng từ người dùng.

Tập trung trong một giao diện website đơn giản, các ứng dụng này hoạt động độc lập nhưng liên kết dữ liệu phân tích chặt chẽ.

- **Nhược điểm**

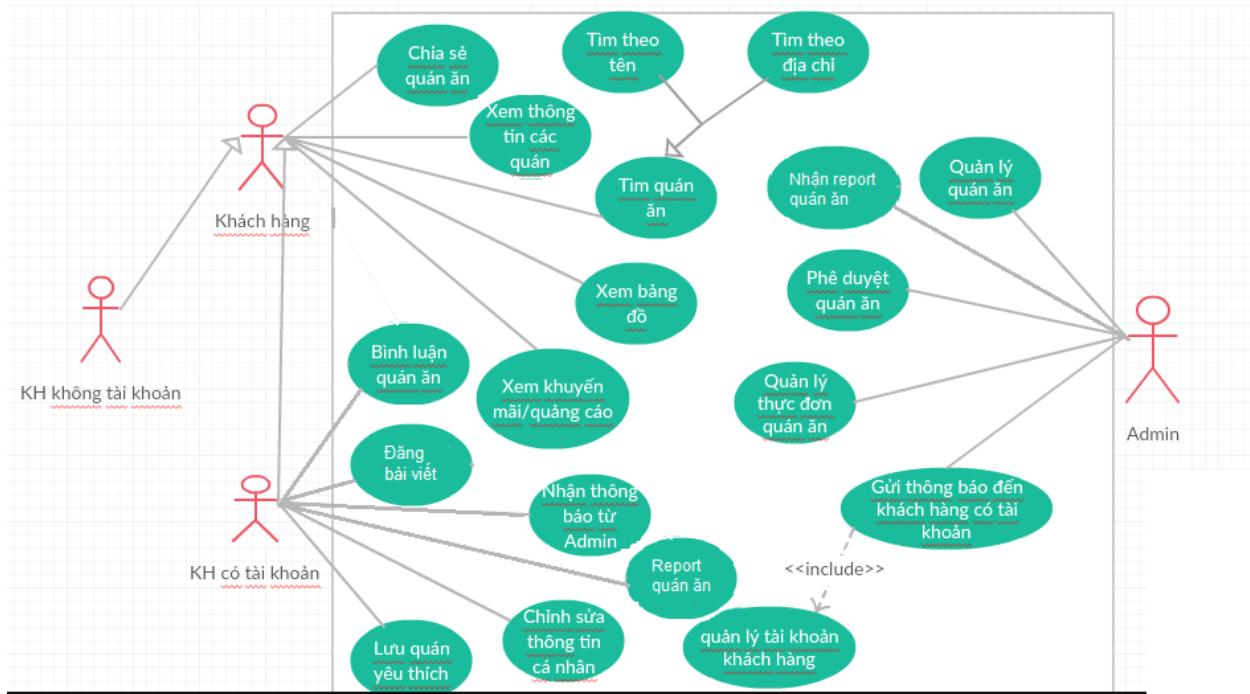
Cơ sở dữ liệu của họ được tổ chức theo kiểu trees, parent-children, không phải là kiểu Table nên những ai đang quen với SQL có thể sẽ gặp khó khăn từ mức độ chút tới khá nhiều.

3.3.5 Cách cài đặt

- 1.Thêm firebase vào ứng dụng
- 2.Thêm SDK
3. Khởi tạo firebase trong ứng dụng
- 4.Pods

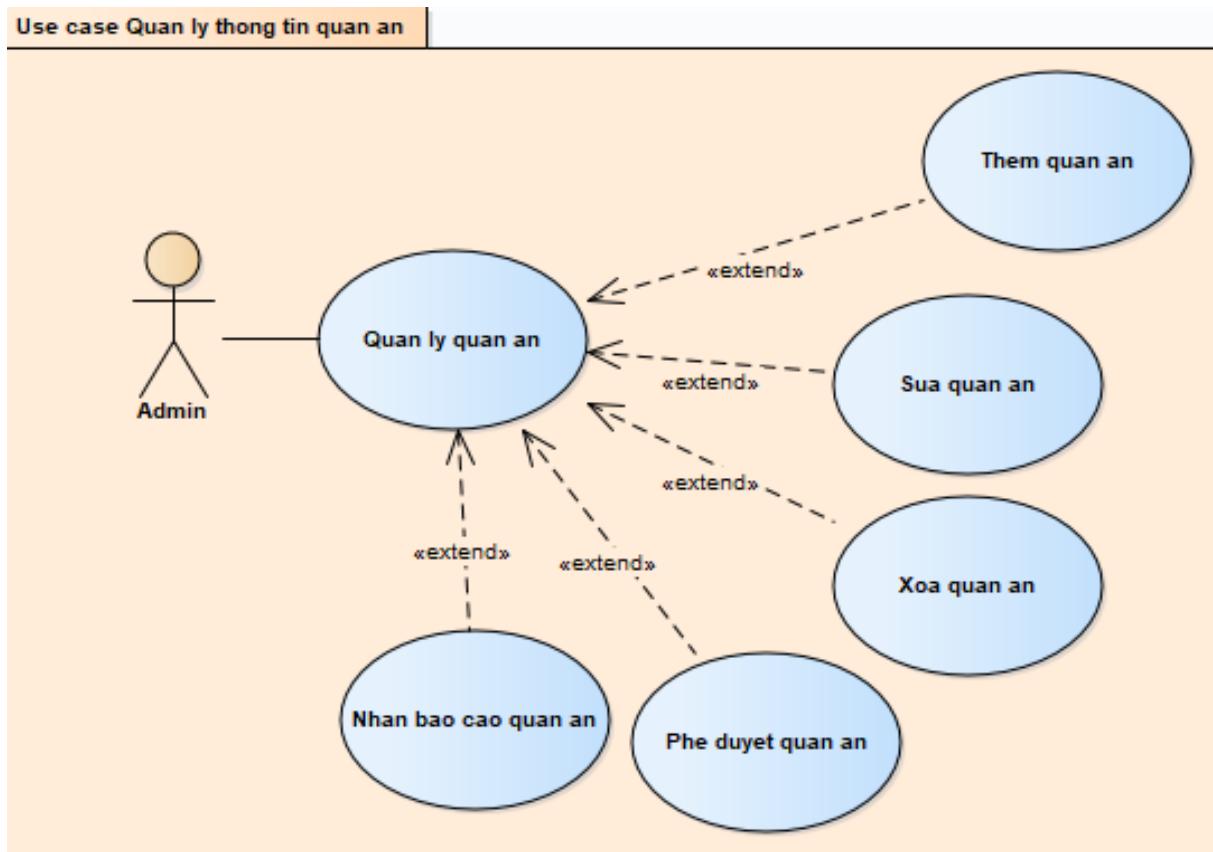
CHƯƠNG 4: THIẾT KẾ HỆ THỐNG HỆ THỐNG

4.1 ĐẶC TẢ USECASE

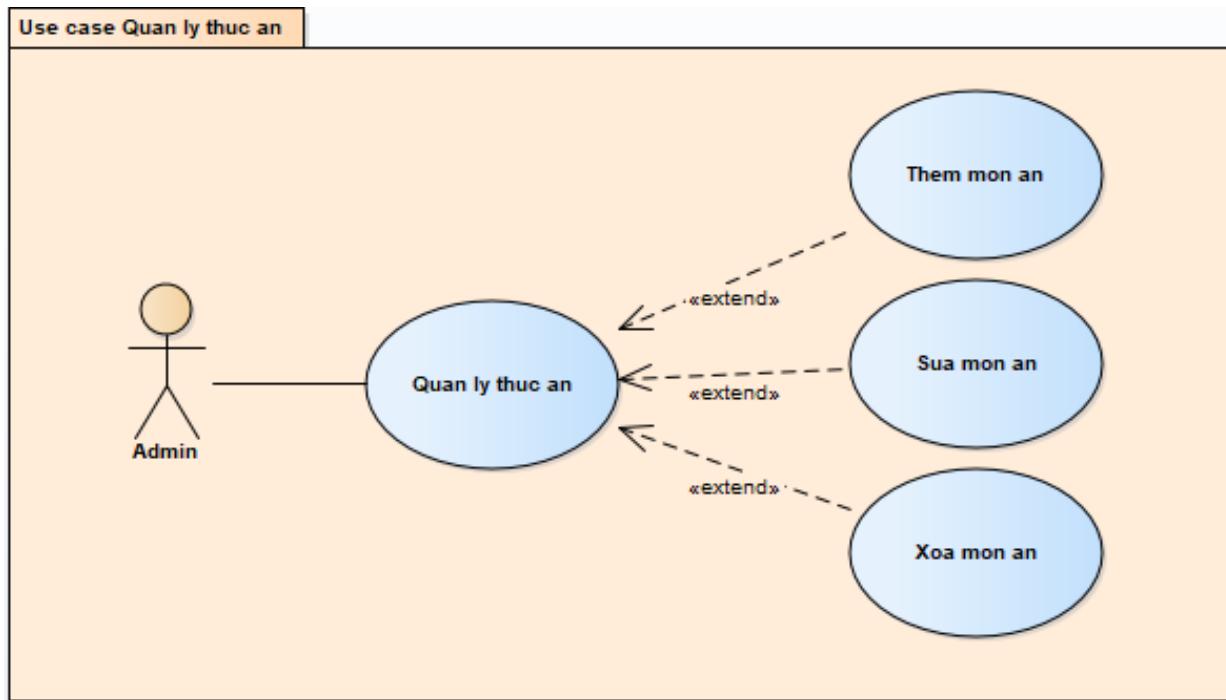


Hình 5 Usecase hệ thống

Admin

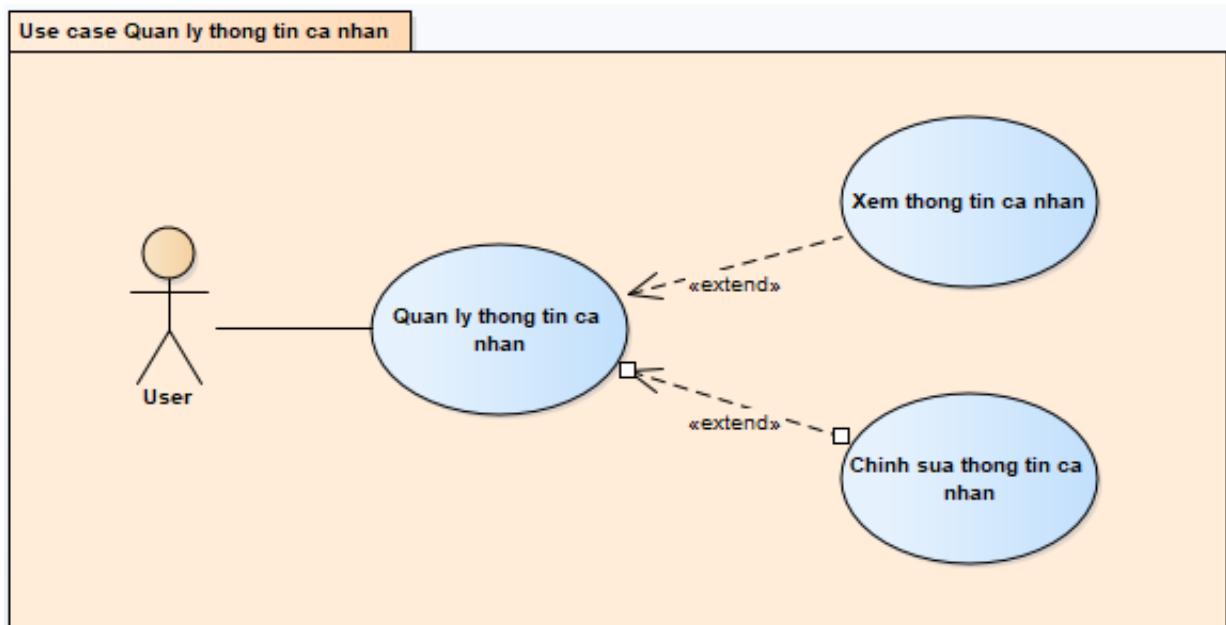


Hình 6 Usecase Quan ly quan an

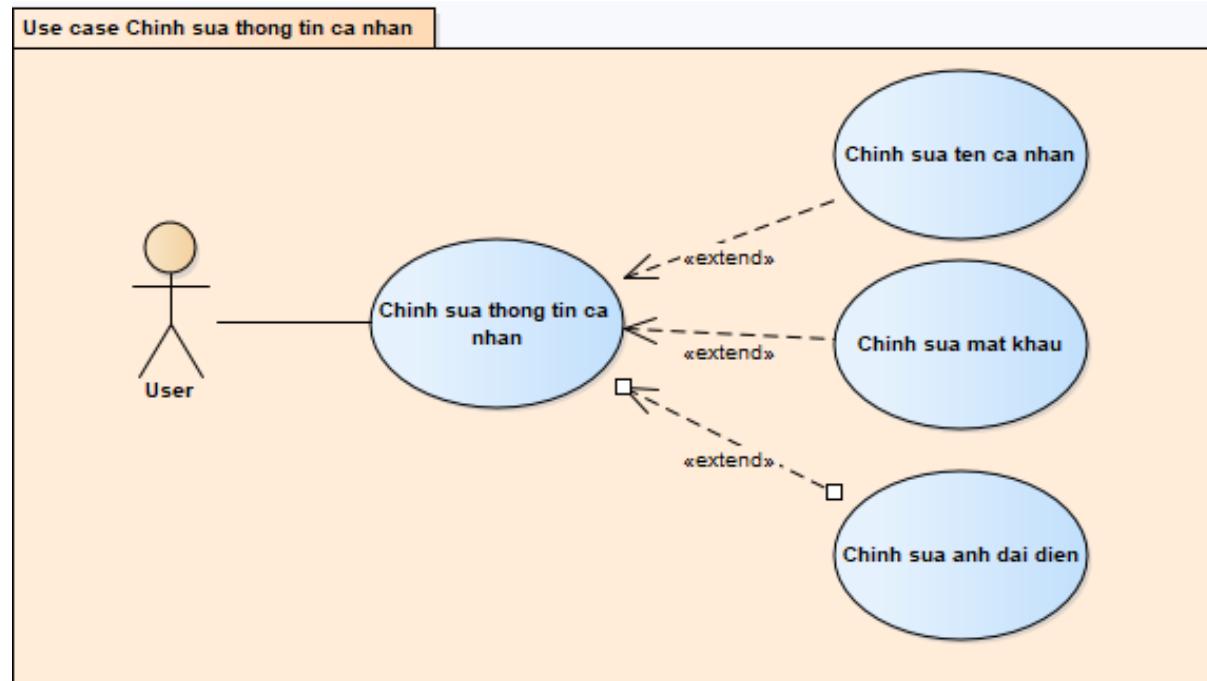


Hình 7 Usecase Quan ly thuc an

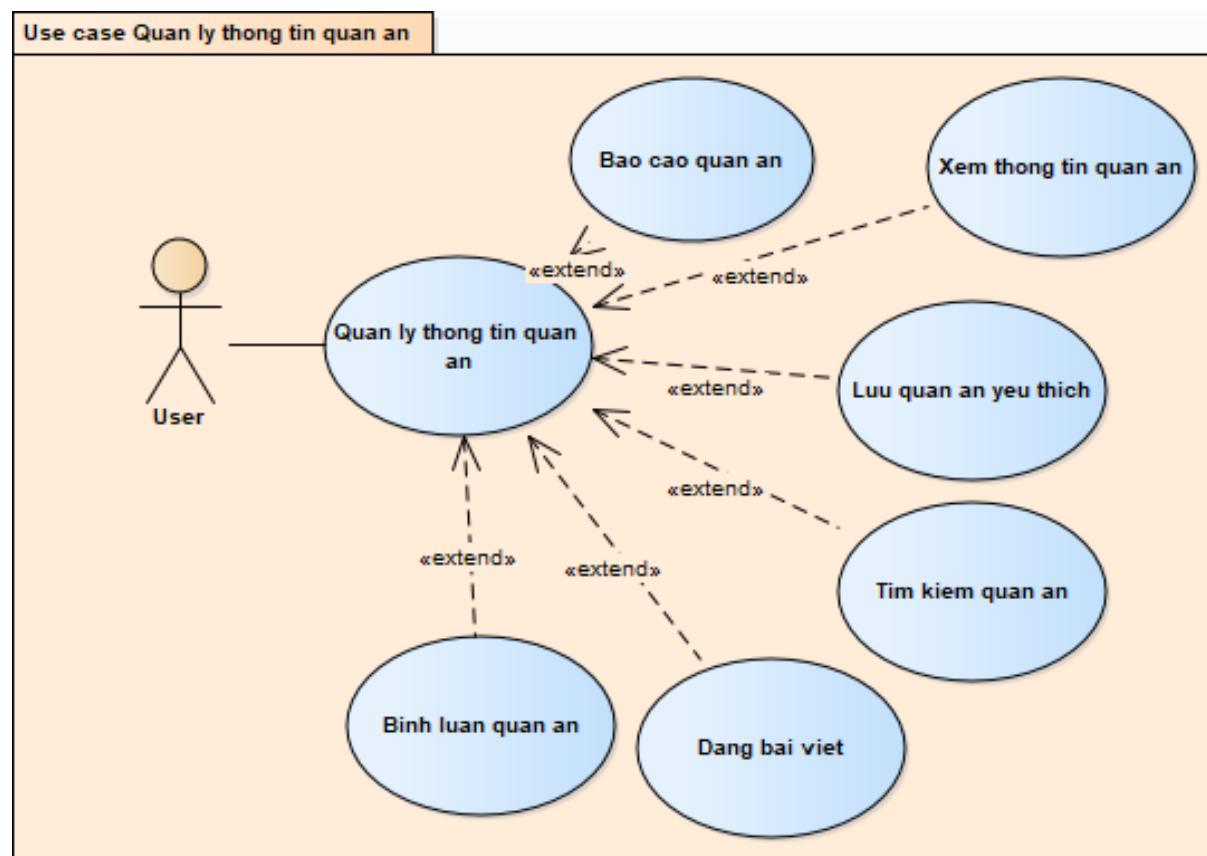
Khách hàng



Hình 8 Usecase Quản lý thông tin cá nhân



Hình 9 Usecase Thay đổi thông tin cá nhân



Hình 10 Usecase Quản lý thông tin quán ăn

4.1.1 Mô tả use case Quản lý quán ăn

Bảng 1 Usecase Them quan an

| Use Case Them quan an | |
|-----------------------|--|
| Mô tả | Cho phép Admin có thể thêm quán ăn từ trang web |
| Tác nhân kích hoạt | Admin |
| Tiền điều kiện | Biết được đường link để vào trang web |
| Các bước thực hiện | <p>(1) Tại màn hình chính của trang web Chọn “Thêm Cửa Hàng”</p> <p>(2) Điền những thông tin cần thiết URL HINH ANH, Tên cửa hàng , địa chỉ, vĩ độ, kinh độ, bài viết giới thiệu</p> <p>(3) Bấm vào “Submit”</p> |

Bảng 2 Usecase Sua quan an

| Use Case Sua quan an | |
|----------------------|--|
| Mô tả | Cho phép Admin có thể sửa quán ăn từ trang web |
| Tác nhân kích hoạt | Admin |
| Tiền điều kiện | Biết được đường link để vào trang web |
| Các bước thực hiện | <p>(1) Tại màn hình chính của trang web Chọn “Sửa Cửa Hàng”</p> <p>(2) Điền những thông tin cần thiết URL HINH ANH, ID , Tên cửa hàng , địa chỉ, vĩ độ, kinh độ, bài viết giới thiệu</p> <p>(3) Bấm vào “Submit”</p> |

Bảng 3 Usecase Xoa quan an

| Use Case Xoa quan an | |
|----------------------|--|
| Mô tả | Cho phép Admin có thể xoá quán ăn từ trang web |
| Tác nhân kích hoạt | Admin |
| Tiền điều kiện | Biết được đường link để vào trang web |
| Các bước thực hiện | <ul style="list-style-type: none"> (1) Tại màn hình chính của trang web Chọn “Xoá Cửa Hàng” (2) Điền những thông tin cần thiết ID (3) Bấm vào “Submit” |

Bảng 4 Usecase Phe duyet quan an

| Use Case Phe duyet quan an | |
|----------------------------|--|
| Mô tả | Cho phép Admin có thể xem bài viết của user và phê duyệt |
| Tác nhân kích hoạt | Admin |
| Tiền điều kiện | Biết được đường link để vào trang web |
| Các bước thực hiện | <ul style="list-style-type: none"> (1) Tại màn hình chính của trang web Chọn “Xem Bài Viết Của User” (2) Điền những thông tin cần thiết Vĩ Đô, Kinh Đô, ID. (3) Bấm vào “Submit” |

Bảng 5 UseCase Nhan bao cao quan an

| Use Case Nhan bao cao quan an | |
|-------------------------------|--|
| Mô tả | Cho phép Admin có thể xem báo cáo về quán ăn mà User gửi về cho Admin |
| Tác nhân kích hoạt | Admin |
| Tiền điều kiện | Biết được đường link để vào trang web |
| Các bước thực hiện | <ul style="list-style-type: none"> (1) Tại màn hình chính của trang web Chọn “Report” (2) Xem mã ID của quán ăn, Tên quán ăn, Report về quán ăn |

4.1.2 Mô tả use case Quản lý món ăn

Bảng 6 Usecase Them mon an

| Use Case Them mon an | |
|----------------------|--|
| Mô tả | Cho phép Admin có thể thêm món ăn từ trang web |
| Tác nhân kích hoạt | Admin |
| Tiền điều kiện | Biết được đường link để vào trang web |
| Các bước thực hiện | <ul style="list-style-type: none"> (1) Tại màn hình chính của trang web Chọn “Thêm Món Ăn” (2) Điền những thông tin cần thiết URL HINH ANH, ID Cửa Hàng, Món Ăn, Giá. (3) Bấm vào “Submit” |

Bảng 7 Usecase Sua mon an

| Use Case Sua mon an | |
|---------------------|--|
| Mô tả | Cho phép Admin có thể sửa món ăn từ trang web |
| Tác nhân kích hoạt | Admin |
| Tiền điều kiện | Biết được đường link để vào trang web |
| Các bước thực hiện | <ul style="list-style-type: none"> (1) Tại màn hình chính của trang web Chọn “Sửa Món ăn” (2) Điền những thông tin cần thiết ID, ID Tục Đơn, URL HINH ANH, Tên Món Ăn (3) Bấm vào “Submit” |

Bảng 8 Usecase Xoa mon an

| Use Case Xoa mon an | |
|---------------------|--|
| Mô tả | Cho phép Admin có thể Xoá món ăn từ trang web |
| Tác nhân kích hoạt | Admin |
| Tiền điều kiện | Biết được đường link để vào trang web |
| Các bước thực hiện | <ul style="list-style-type: none"> (1) Tại màn hình chính của trang web Chọn “Xoá Món Ăn” (2) Điền những thông tin cần thiết ID Cửa hàng, ID món ăn (3) Bấm vào “Submit” |

4.1.3 Mô tả Use Case Quản lý thông tin cá nhân và thay đổi thông tin cá nhân

Bảng 9 Usecase Xem thông tin cá nhân

| Use Case Xem thông tin cá nhân | |
|--------------------------------|---|
| Mô tả | Cho phép khách hàng xem thông tin cá nhân |
| Tác nhân kích hoạt | Khách hàng |
| Tiền điều kiện | Đăng nhập được tài khoản |
| Các bước thực hiện | (1) Tại màn hình chính của ứng dụng Chọn “Tài khoản” (2) Xem thông tin cá nhân gồm Ảnh đại diện, tên của bạn, email của bạn, thay đổi tên của bạn, thay đổi mật khẩu, thay đổi Avatar |

Bảng 10 Usecase Chinh sua ten ca nhan

| Use Case Chinh sua ten ca nhan | |
|--------------------------------|--|
| Mô tả | Cho phép khách hàng chỉnh sửa tên khách hàng |
| Tác nhân kích hoạt | Khách hàng |
| Tiền điều kiện | Đăng nhập được tài khoản |
| Các bước thực hiện | (1) Tại màn hình chính của ứng dụng Chon “Tài khoản” (2) Bấm vào “Thay Đổi tên của bạn” (3) Điền tên muốn đổi của bạn (4) Bấm “Ok” |

Bảng 11 Usecase Chinh sua mat khau

| Use Case Chinh sua mat khau | |
|-----------------------------|---|
| Mô tả | Cho phép khách hàng thay đổi mật khẩu của khách hàng |
| Tác nhân kích hoạt | Khách hàng |
| Tiền điều kiện | Đăng nhập được tài khoản |
| Các bước thực hiện | <ul style="list-style-type: none"> (1) Tại màn hình chính của ứng dụng Chọn “Tài khoản” (2) Bấm vào “Thay đổi mật khẩu” (3) Bấm mật khẩu của bạn cần đổi (4) Nhập lại mật khẩu cần đổi (5) Bấm “Ok” |

Bảng 12 Usecase Chinh sua anh dai dien

| Use Case Chinh sua anh dai dien | |
|---------------------------------|---|
| Mô tả | Cho phép khách hàng thay đổi ảnh đại diện |
| Tác nhân kích hoạt | Khách hàng |
| Tiền điều kiện | Đăng nhập được tài khoản |
| Các bước thực hiện | <ul style="list-style-type: none"> (1) Tại màn hình chính của ứng dụng Chọn “Tài khoản” (2) Bấm vào “Thay Đổi Avatar” (3) Chọn ảnh bạn muốn đổi (4) Bấm “Choose” |

4.1.4 Mô tả Use case Quản Lý Thông tin Quán ăn

Bảng 13 Usecase Xem thông tin quán ăn

| Use Case Xem thông tin quán ăn | |
|--------------------------------|---|
| Mô tả | Cho phép khách hàng xem thông tin quán ăn |
| Tác nhân kích hoạt | Khách hàng |
| Tiền điều kiện | Đăng nhập được tài khoản |
| Các bước thực hiện | (1) Tại màn hình chính của ứng dụng Chọn “Home” (2) Xem thông tin các quán ăn có trong ứng dụng (3) Bấm vào các quán ăn sẽ hiện ra “Map” và các món ăn |

Bảng 14 Usecase Lưu quán ăn yêu thích

| Use Case Lưu Quán ăn yêu thích | |
|--------------------------------|--|
| Mô tả | Cho phép khách hàng lưu các quán ăn yêu thích |
| Tác nhân kích hoạt | Khách hàng |
| Tiền điều kiện | Đăng nhập được tài khoản |
| Các bước thực hiện | (1) Tại màn hình chính của ứng dụng Chọn “Home” (2) Bấm vào quán ăn mà mình yêu thích (3) Bấm vào “Lưu” phía bên phải ở phía trên |

Bảng 15 UseCase Tim kiem quan an

| Use Case Tim kiem quan an | |
|---------------------------|---|
| Mô tả | Cho phép khách hàng tìm kiếm quán ăn theo tên quán ăn hoặc địa chỉ của quán |
| Tác nhân kích hoạt | Khách hàng |
| Tiền điều kiện | Đăng nhập được tài khoản |
| Các bước thực hiện | <ul style="list-style-type: none"> (1) Tại màn hình chính của ứng dụng Chọn “Home” (2) Điền vào thông tin cần tìm vào search bar (3) Chọn tìm theo “tên quán” hoặc ”địa chỉ” |

Bảng 16 UseCase Dang tai khoan

| Use Case Dang tai khoan | |
|-------------------------|--|
| Mô tả | Cho phép khách hàng đăng bài viết về quán ăn mà mình muốn |
| Tác nhân kích hoạt | Khách hàng |
| Tiền điều kiện | Đăng nhập được tài khoản |
| Các bước thực hiện | <ul style="list-style-type: none"> (1) Tại màn hình chính của ứng dụng Chọn “Home” (2) Nhấn vào dấu “+” phía trên, bên phải (3) Nhấn vào dấu “Thêm” phía trên, bên phải (4) Điền những thông tin cần thiết của quán ăn và món ăn (5) Bấm “Chia sẻ bài viết này” |

Bảng 17 UseCase Bình luận quan an

| Use Case Bình luận quan an | |
|----------------------------|---|
| Mô tả | Cho phép khách hàng bình luận được quán ăn mà mình muốn |
| Tác nhân kích hoạt | Khách hàng |
| Tiền điều kiện | Đăng nhập được tài khoản |
| Các bước thực hiện | <ol style="list-style-type: none">(1) Tại màn hình chính của ứng dụng Chọn “Home”(2) Bấm vào quán ăn mình muốn(3) Kéo xuống phần bình luận(4) Dièn những thông tin cần thiết(5) Bấm vào hình máy bay giấy bên phia phải |

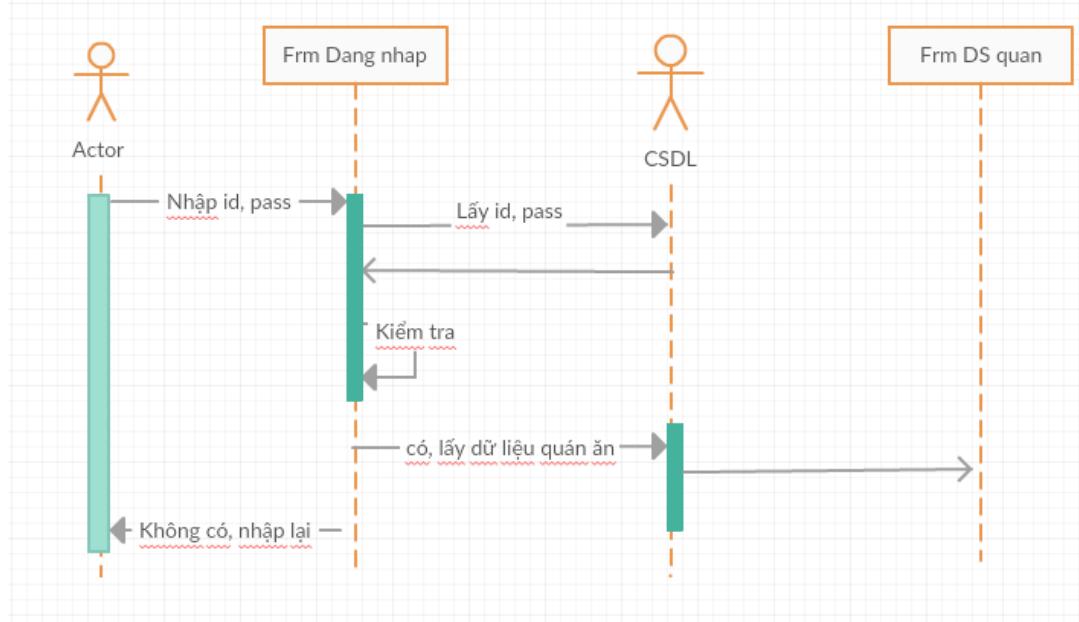
Bảng 18 UseCase Báo cáo quan an

| Use Case Báo cáo quan an | |
|--------------------------|--|
| Mô tả | Cho phép khách hàng báo cáo về quán ăn mà mình muốn về cho Admin |
| Tác nhân kích hoạt | Khách hàng |
| Tiền điều kiện | Đăng nhập được tài khoản |

| | |
|--------------------|---|
| Các bước thực hiện | (1) Tại màn hình chính của ứng dụng Chọn “Home” (2) Bấm vào quán ăn mình muốn (3) Bấm vào “report” phía trên bên phải (4) Điền những thông tin cần thiết (5) Bấm “Gửi” |
|--------------------|---|

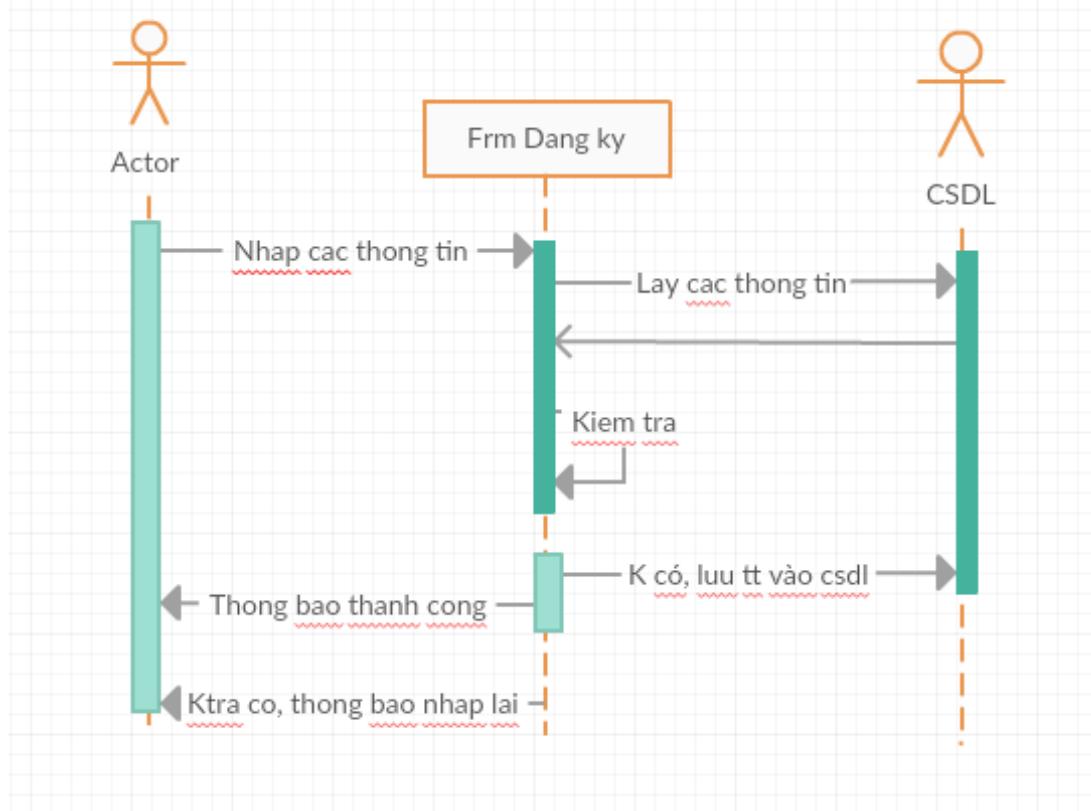
4.2 LUẬC ĐỒ SEQUENCE

4.2.1 Lược đồ Sequence Diagram đăng nhập



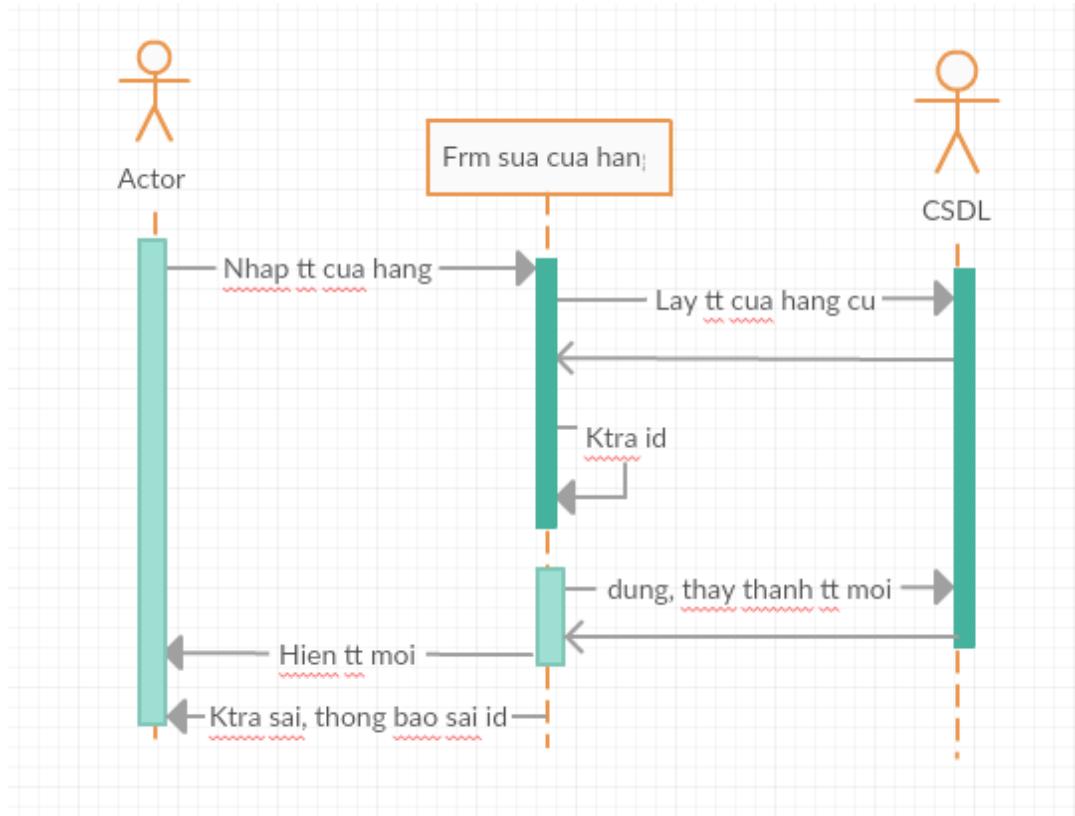
Hình 11 Sequence Diagram cho Đăng nhập

4.2.2 Lược đồ Sequence Diagram cho Đăng ký



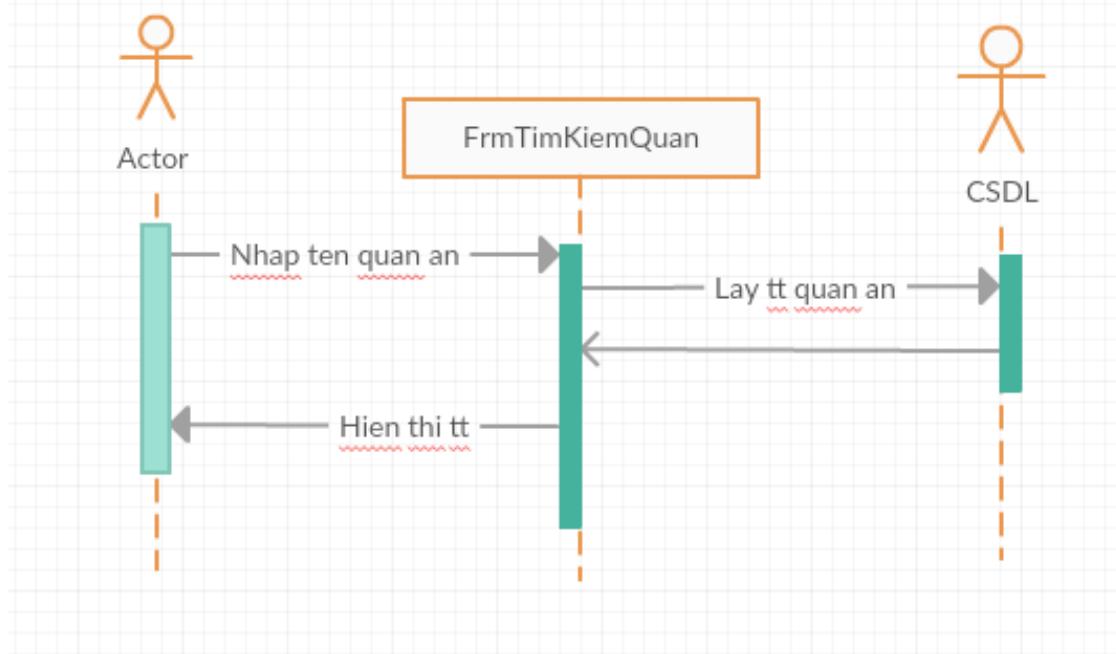
Hình 12 Sequence Diagram cho Đăng Kí

4.2.3 Sequence Diagram cho sửa cửa hàng



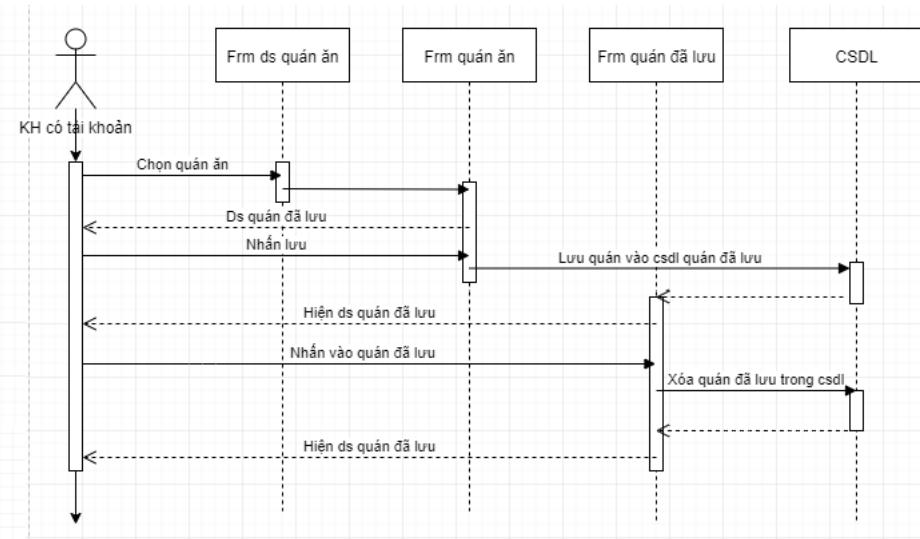
Hình 13 Sequence Diagram cho Sửa Cửa Hàng

4.2.4 Sequence Diagram cho Tìm Kiếm Quán Ăn



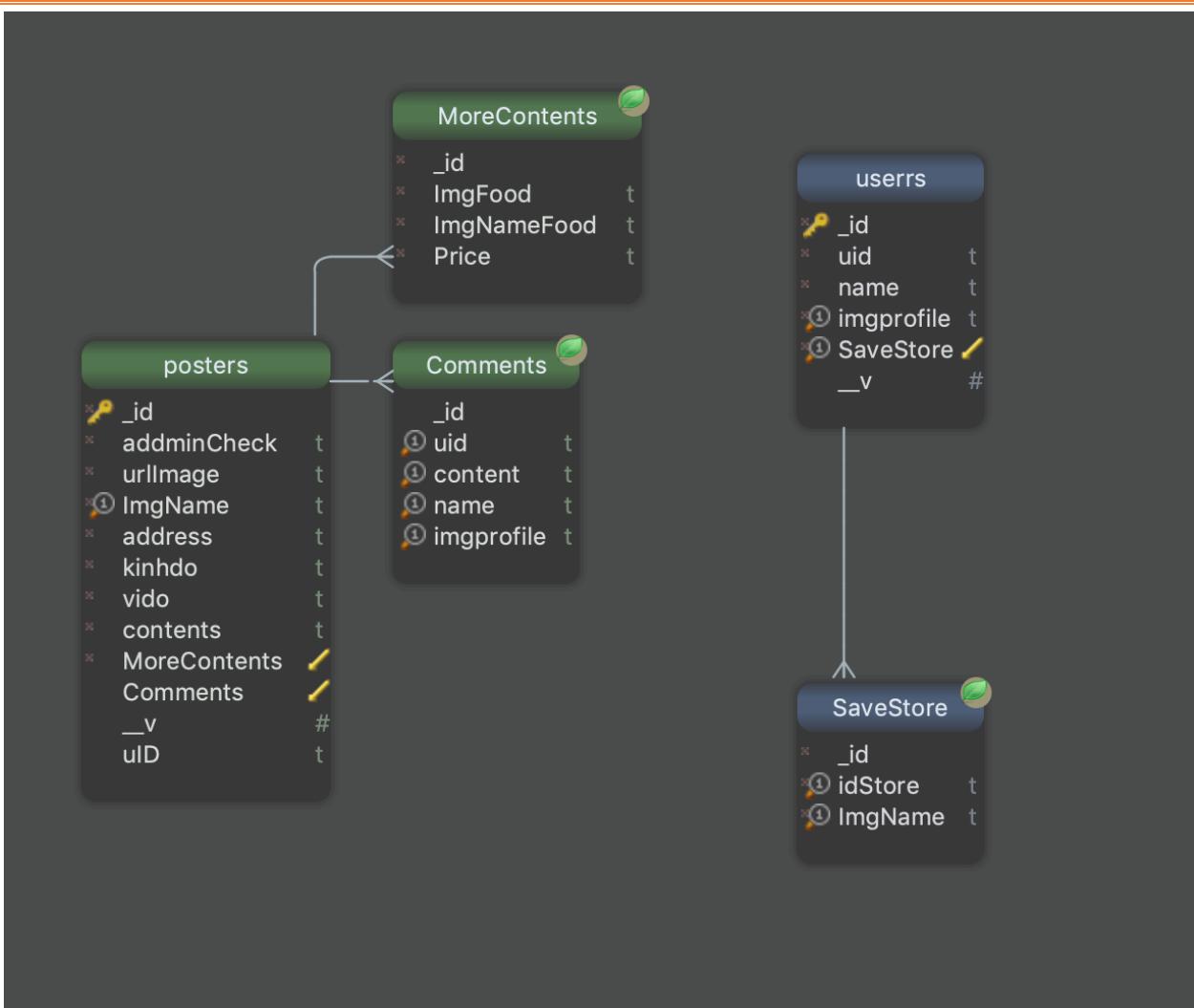
Hình 14 Sequence Diagram cho Tìm Kiếm Quán Ăn

4.2.5 Sequence Diagram cho Lưu Quán Ăn



Hình 15 Sequence Diagram cho Lưu Quán Ăn

4.3 CHI TIẾT CÁC BẢNG DỮ LIỆU



Hình 16 Cơ sở dữ liệu của ứng dụng

- **poster:**

Bảng 19 Bảng quán ăn poster của Mongodb

| Field name | Data type | Description |
|------------|-----------|---|
| id | ObjectId | Id mongodb |
| adminCheck | String | Admin kiểm tra người dùng đăng bài viết |
| urlImage | String | Địa chỉ hình món ăn |

| | | |
|--------------|----------|-------------------------------------|
| ImgName | String | Tên của quán ăn |
| Address | String | Địa chỉ của quán ăn |
| Kinhdo | String | Kinh độ của quán ăn |
| Vido | String | Vĩ độ của quán ăn |
| Contents | String | Mô tả quán ăn |
| moreContents | String | Mô tả thực đơn quán ăn |
| Comments | String | Mô tả bình luận |
| uID | ObjectId | Xác định người dùng sở hữu bài viết |

- MoreContents

Bảng 20 Bảng MoreContents của Mongodb

| Field name | Data type | Description |
|-------------|-----------|---------------------|
| id | ObjectId | Id mongodb |
| ImgFood | String | Địa chỉ hình món ăn |
| ImgNameFood | String | Tên món ăn |

| | | |
|-------|--------|----------------|
| Price | String | Số tiền món ăn |
|-------|--------|----------------|

- Comments

Bảng 21 Bảng comments trong Mongoddb

| Field name | Data type | Description |
|------------|-----------|---------------------------------|
| Id | ObjectId | ID mongodb |
| Uid | String | Mã member |
| Content | String | Bình luận của người dùng |
| Name | String | Tên của người dùng đã bình luận |
| Imgprofile | String | Url Hình ảnh người dùng |

- Userrs

Bảng 22 Bảng User trong Firebase

| Field name | Data type | Description |
|------------|-----------|-------------------------|
| id | ObjectId | Id mongodb |
| uid | String | Mã member |
| Name | String | Tên người dùng |
| Imgprofile | String | Url Hình ảnh người dùng |
| Savestore | String | Các quán ăn được lưu |

Bảng 23 Bảng SaveStore trong Firebase

- SaveStore

| Field name | Data type | Description |
|------------|-----------|-------------|
| _id | ObjectId | Id mongodb |
| idStore | String | Mã quán ăn |
| ImgName | String | Tên quán ăn |

4.4 THIẾT KẾ GIAO DIỆN

4.4.1 Màn hình giao diện và xử lý bên Admin

4.4.1.1 Chức năng nghiệp vụ của Admin

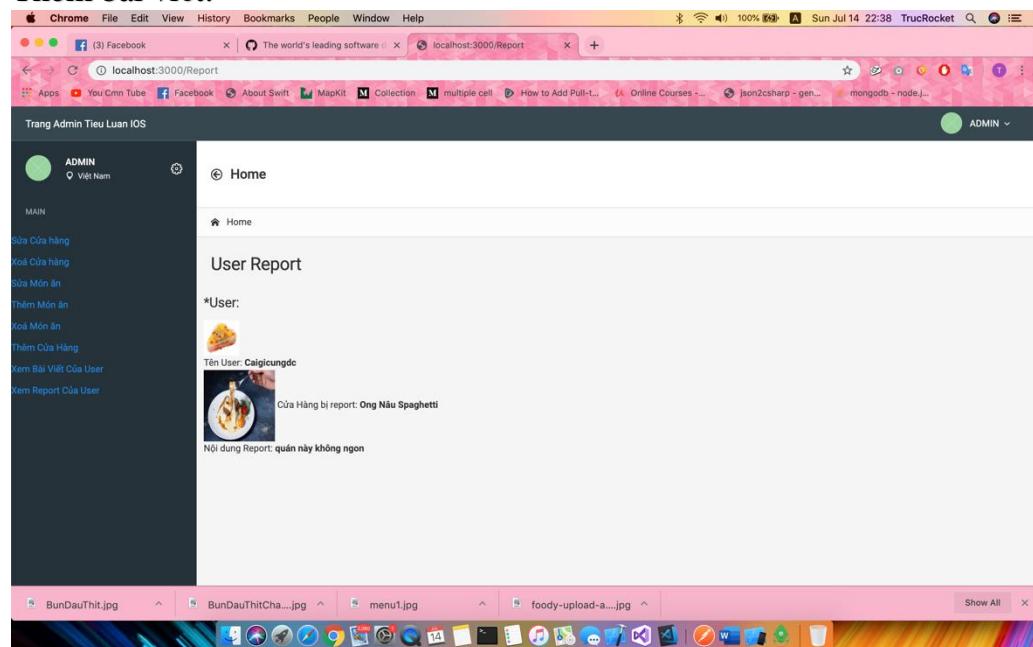
Bảng 24 nghiệp vụ của admin

| ST T | Tên xử lý | Loại công việc | Quy định/Công thức liên quan | Ghi chú |
|---------|---------------|-------------------|---------------------------------------|--|
| 1 | Xóa Bài viết | Xóa dữ liệu | http://localhost:3000/removeS tore | Xóa bỏ cửa hàng khỏi database |
| 2 | Thêm bài viết | Lưu trữ | http://localhost:3000/addStore | Thêm quán ăn |
| 3 | Sửa Bài viết | Lưu trữ | http://localhost:3000/editStore | Trường hợp nhập thông tin địa danh bị sai thì có thể chỉnh sửa. |
| 4 | Xóa Thực đơn | Xóa dữ liệu | http://localhost:3000/removeF ood | Những món ăn không được bán nữa thì sẽ được xóa |

| | | | | |
|---|----------------------------------|---------------|---------------------------------|---|
| 5 | Sửa thực đơn | Lưu trữ | http://localhost:3000/editFood | Những món ăn cần được thay đổi tên hoặc giá hoặc hình ảnh thì sẽ được chỉnh sửa lại |
| 6 | Thêm thực đơn | Lưu trữ | http://localhost:3000/addFood | Thêm thực đơn vào cửa hàng |
| 7 | Xem bài viết của User | Lưu trữ | http://localhost:3000/UserShare | Phê duyệt quán ăn do các User gửi về |
| 8 | Nhận báo cáo của User về quán ăn | Xem thông tin | | Nhận báo cáo của User về quán ăn |

4.4.1.2 Màn hình giao diện

- Thêm bài viết:



Hình 17 Thêm bài viết

- Sửa bài viết

Edit Store

URL HINH ANH:

ID:

Tên Cua Hang:

Địa chỉ:

Vi Do

Kinh Do:

Bài viết giới thiệu:

Submit

Hình 18 Sửa bài viết

- Xóa bài viết:

Delete Store

ID:

Hình 19 Xóa bài viết

- Thêm thực đơn

Add Menu

URL HINH ANH:

ID Cua Hang

MonAn

Gia

Submit

Hình 20 Thêm thực đơn

- Sửa thực đơn

Edit Menu

ID:

ID Thuc Don:

URL HINH ANH:

Tên Mon An:

Gia:

Submit

Hình 21 Sửa thực đơn

- Xóa thực đơn

Delete Store

ID Cửa hàng:

ID Món Ăn:

Submit

Hình 22 Xóa thực đơn

- Xem bài viết của User

Submit Store User Share

Quán Ăn: Thành Mập - Chân Gà Rút Xương Ngâm Sả Tắc



Địa chỉ: 220/68/5, âu cơ, quận tân bình, thành phố hồ chí minh

Thông tin quán ăn: 8:00 - 23:59 15,000đ - 35000đ

Nhập vĩ độ:

Nhập kinh độ:

Thực Đơn quán ăn



Tên Món: chân gà

Giá: 35.000đ

Submit

Hình 23 Xem bài viết của User

4.4.2 Màn hình và xử lý bên Guest

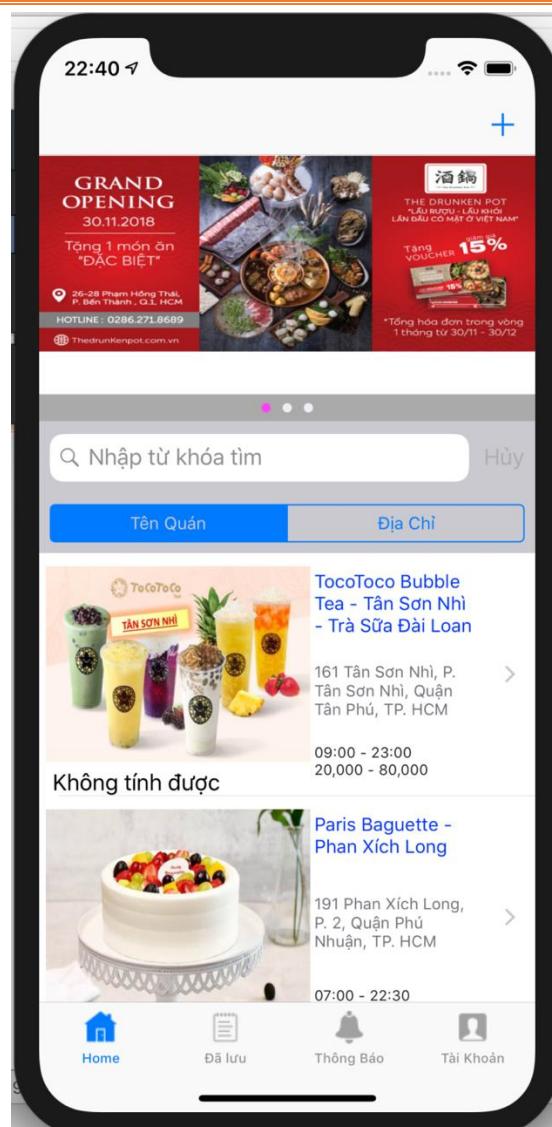
4.4.2.1 Các năng nghiệp vụ của Guest

Bảng 25 Nghiệp vụ của Guest

| STT | Tên xử lý | Điều kiện gọi thực thi | Mô tả | Biểu mẫu liên quan |
|-----|---|---|--|--------------------|
| 1 | Tìm kiếm theo tên quán | Ghi vào textbox sẽ tự động tìm kiếm | Khi nhập tên quán ăn xong list quán ăn sẽ hiển thị ra | BM_Search |
| 2 | Tìm kiếm theo địa chỉ quán | Chọn button Địa chỉ Sau đó ghi vào textbox sẽ tự động tìm kiếm | Khi nhập địa chỉ vào list quán ăn có địa chỉ trùng với địa chỉ nhập sẽ hiển thị ra | BM_Search_Local |
| 3 | Xem (Hình ảnh quán cáo, tên quán ăn, địa chỉ, giá cả, bản đồ chỉ đường) | Xem thông tin | Tất cả thông tin sẽ được hiển thị trên màn hình | BM_View_Info |
| 4 | Login with gmail password | Bấm vào tài khoản | Chuyển qua button login để nhập email và password | BM_Login |
| 5 | Register | Bấm vào tài khoản | Màn hình Register sẽ xuất hiện | BM_Register |

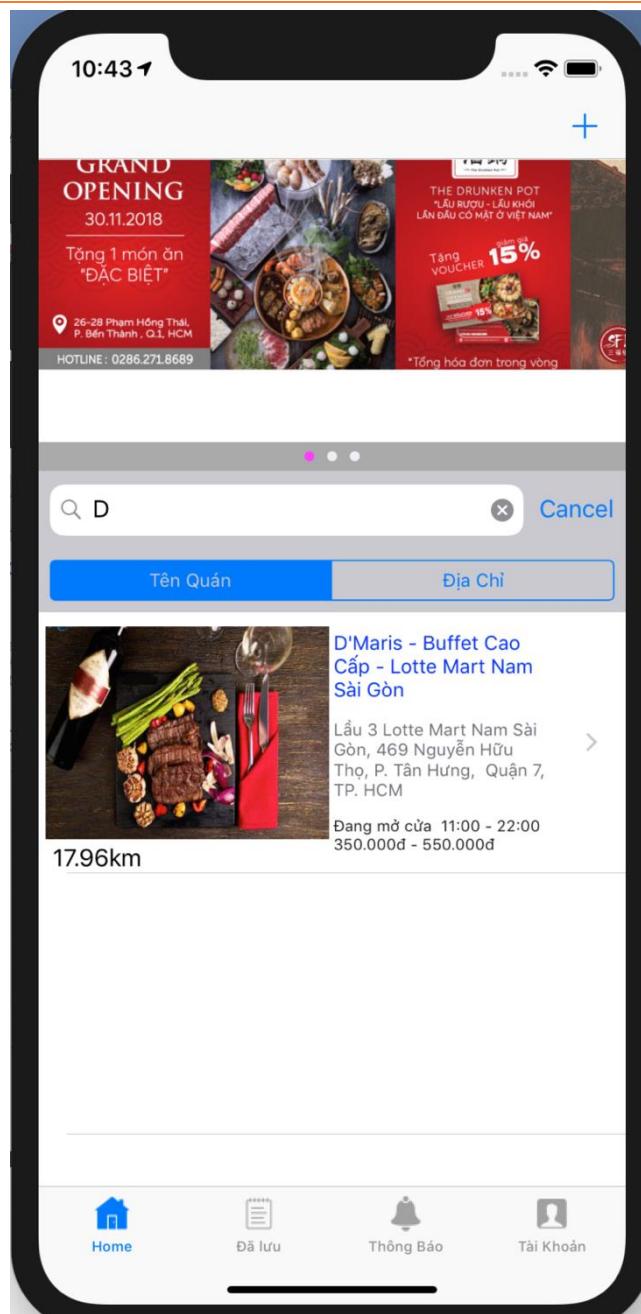
4.4.2.2 Màn hình giao diện

- BM_Search:



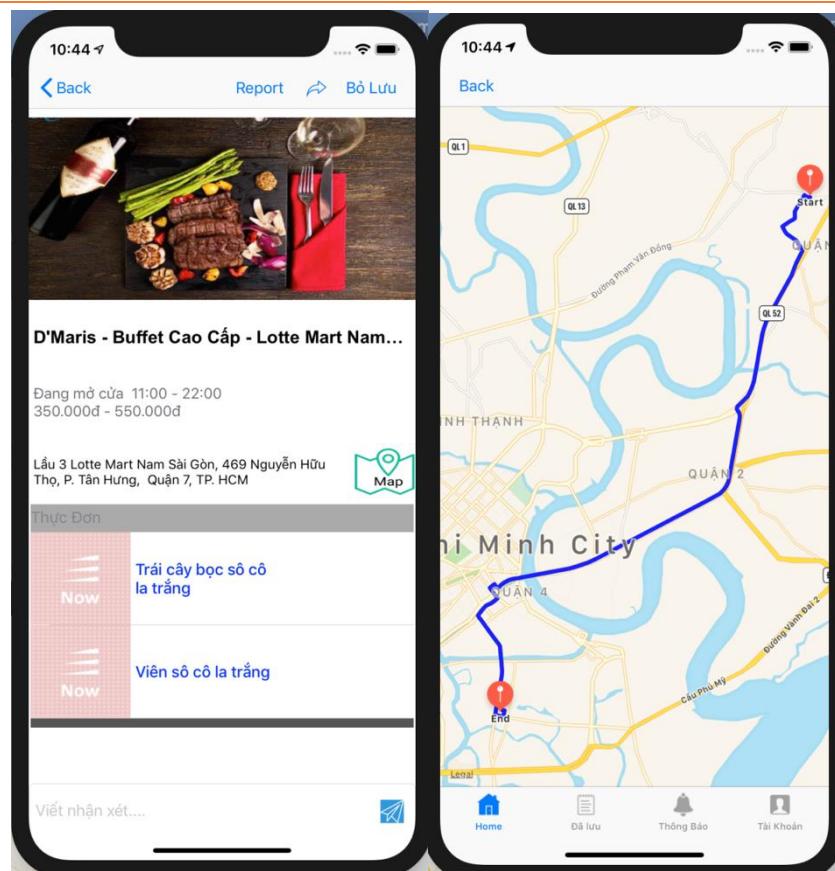
Hình 24 BM_Search

- BM_Search_Local



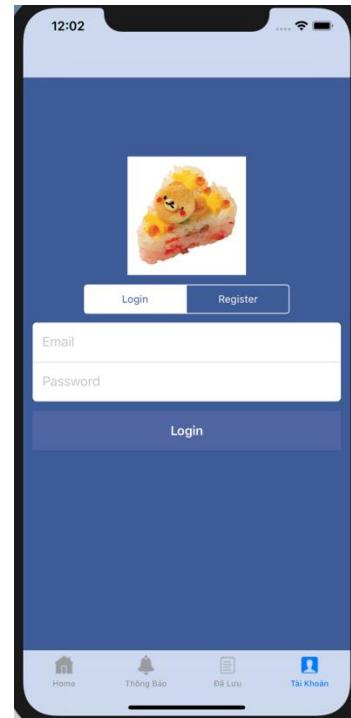
Hình 25 BM_Search_Local

- View Info



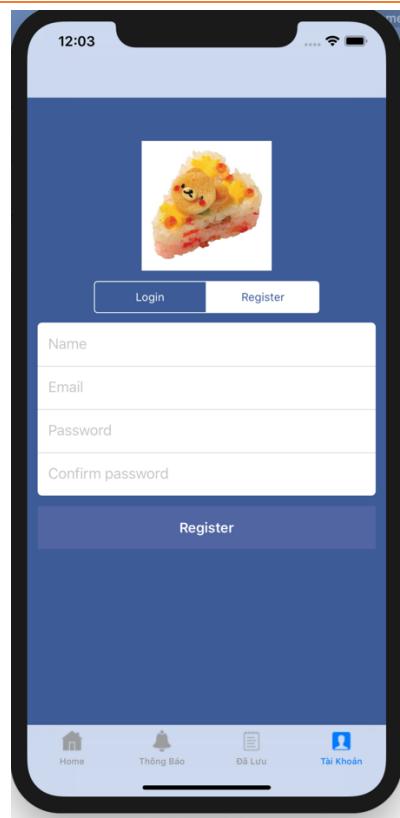
Hình 26 View_Info

- BM_Login



Hình 27 BM_Login

- BM_Register



Hình 28 BM_Register

4.4.3 Màn hình và xử lý của User

4.4.3.1 Các chức năng nghiệp vụ của User:

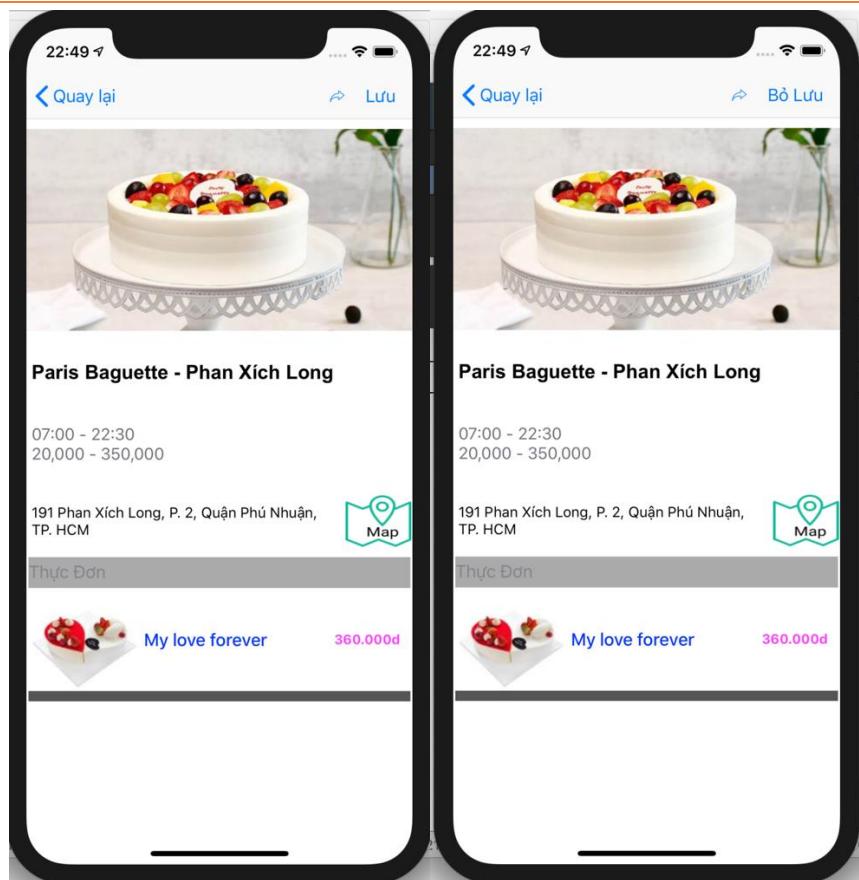
Bảng 26 Nghiệp vụ của User

| STT | Công việc | Loại công việc | Ghi chú | Biểu mẫu liên quan |
|-----|--|----------------|---------|--------------------|
| 1 | Tìm kiếm theo tên quán | Tìm kiếm | | BM_Search |
| 2 | Tìm kiếm theo địa chỉ | Tìm kiếm | | BM_Search_Local |
| 3 | View Info (Tên quán ăn, địa chỉ, xem bản đồ, xem thực đơn) | Xem thông tin | | BM_View_Info |
| 4 | Lưu bài viết | Lưu trữ | | BM_Save_Info |

| | | | | |
|----|-----------------------|---------------|--|-------------------|
| 5 | Chia sẻ bài viết | | | BM_Share_Store |
| 6 | Bình luận quán ăn | Lưu Trữ | | BM_Comments_Store |
| 7 | Xem bài viết đã lưu | Xem thông tin | | BM_View_Save |
| 8 | Đăng bài viết | Lưu trữ | | BM_Post_Store |
| 9 | Nhận thông báo | Xem thông tin | | BM_Notification |
| 10 | Report | Lưu trữ | | BM_Report_store |
| 11 | Thay đổi tên hiển thị | Lưu trữ | | BM_Edit_Name |
| 12 | Thay đổi mật khẩu | Lưu trữ | | BM_Edit_Pass |
| 13 | Thay đổi avatar | Lưu trữ | | BM>Edit_I mage |
| 14 | Logout | | | BM_Logout |

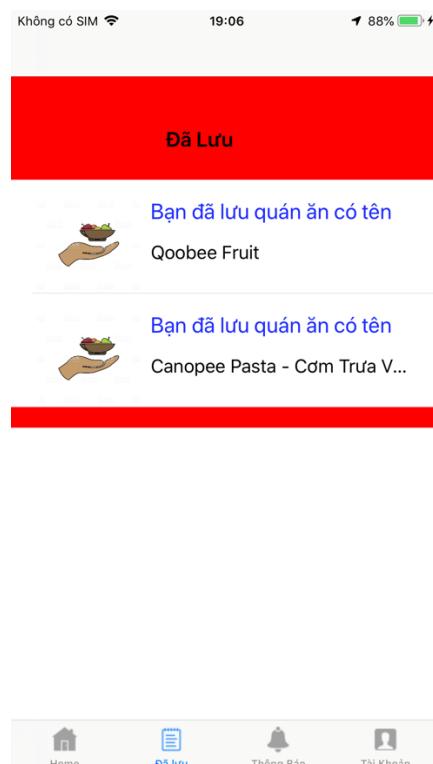
4.4.3.2 Màn hình giao diện

- BM_Save_Info



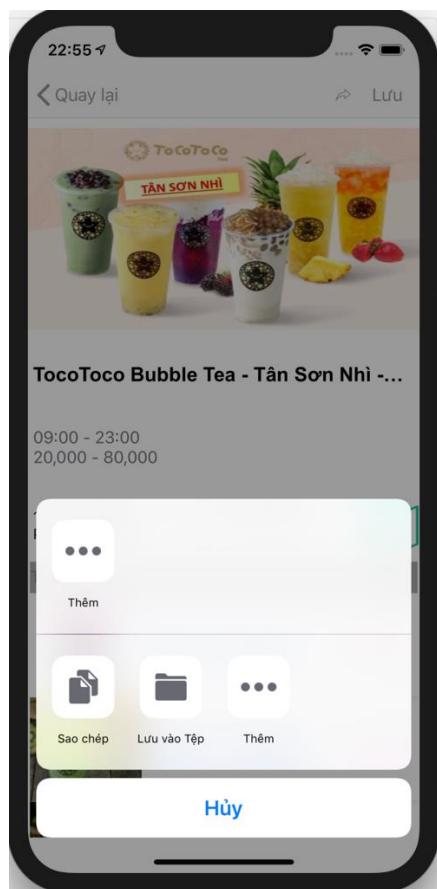
Hình 29 BM_Save_Info

- BM_View_Save



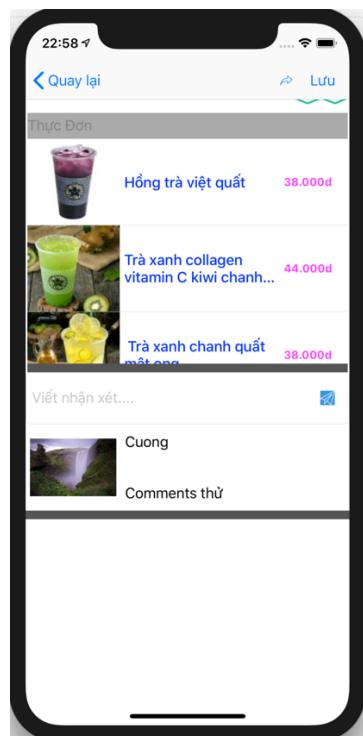
Hình 30 BM_View_Save

- BM_Share_Store



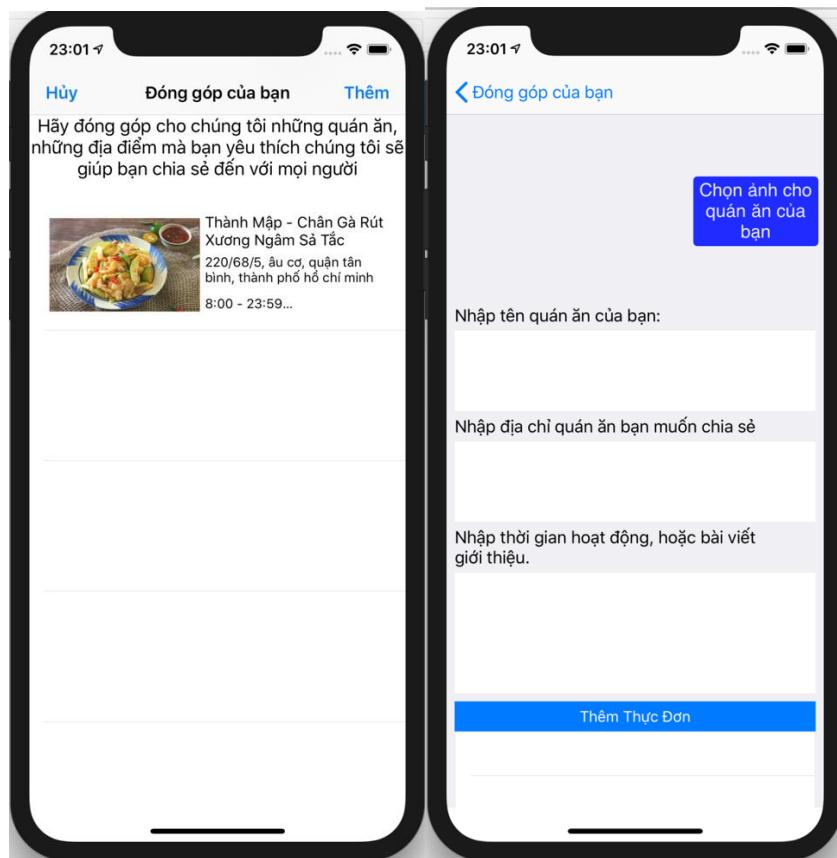
Hình 31 BM_Share_Store

- BM_Comments_Store



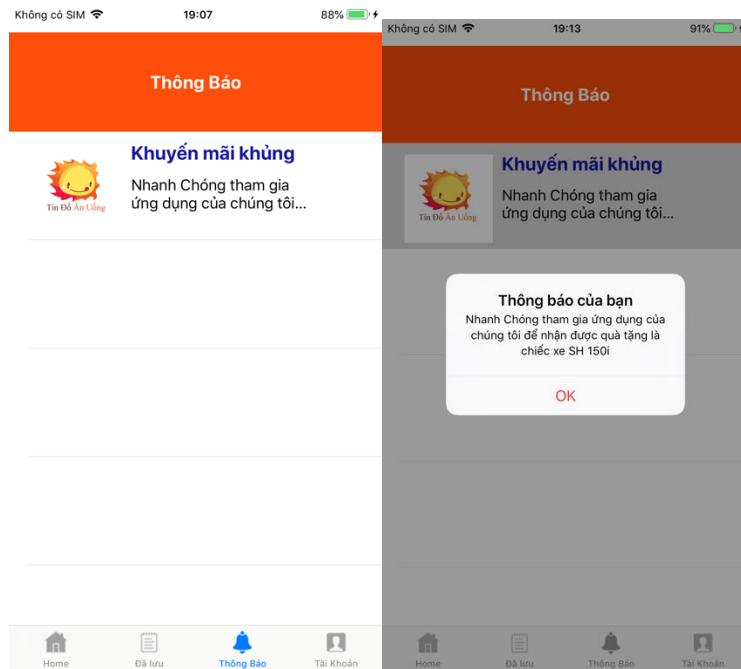
Hình 32 BM_Comments_Store

- BM_Post_Store



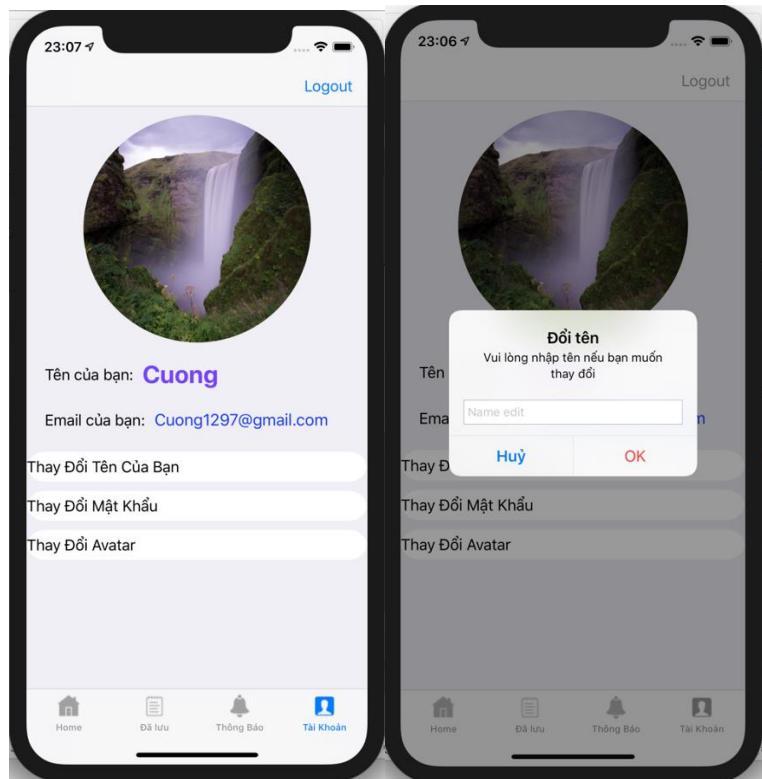
Hình 33 BM_Post_Store

- BM_Notification



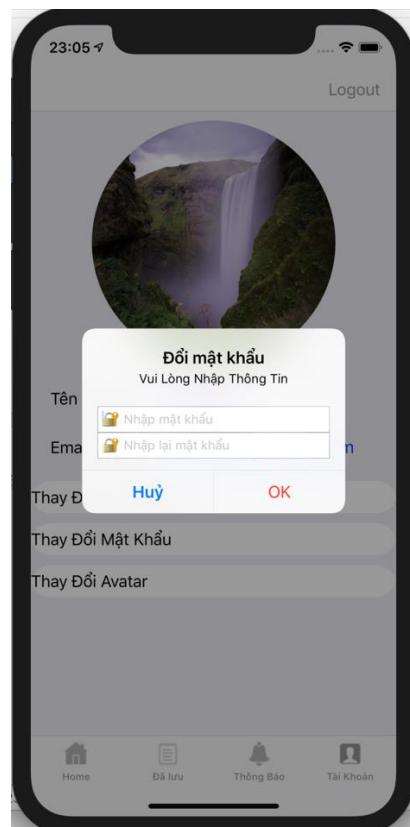
Hình 34 BM_Notification

- BM_Edit_Name



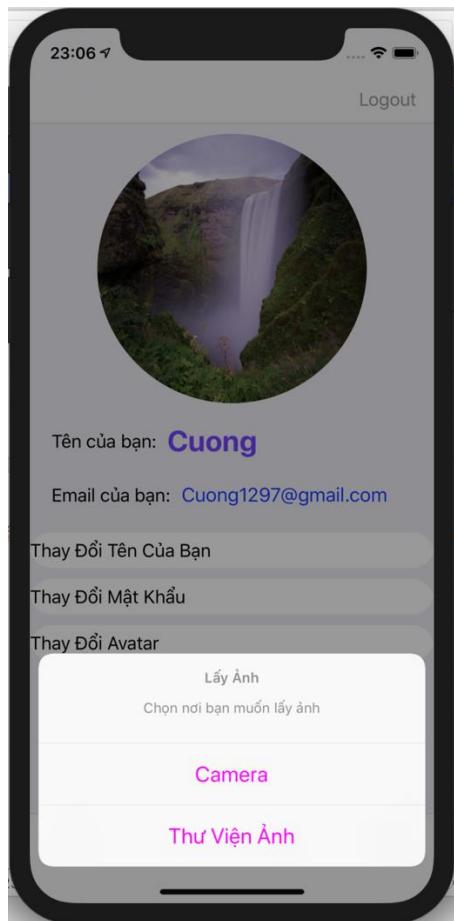
Hình 35 BM_Edit_Name

- BM_Edit_Pass



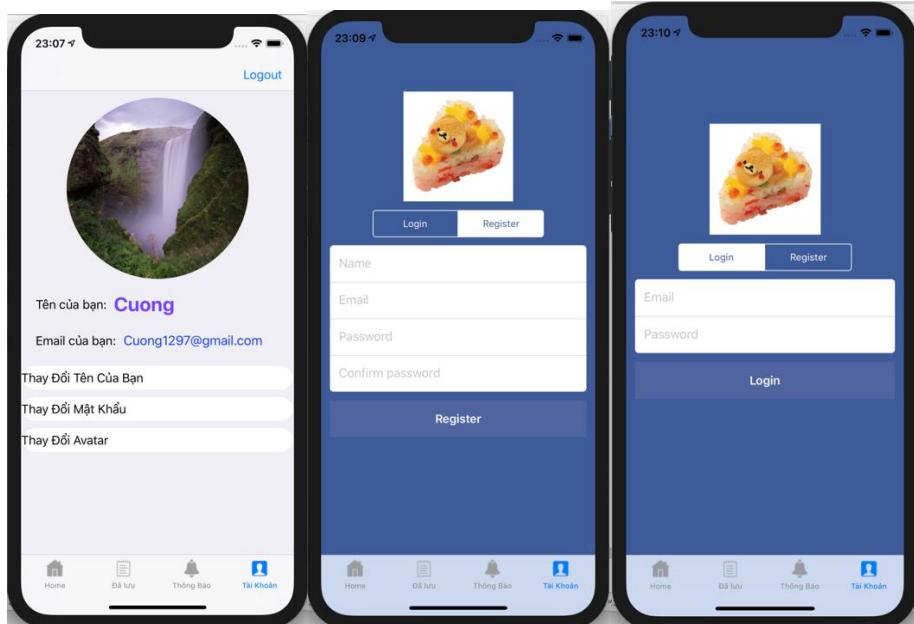
Hình 36 BM_Edit_Pass

- BM_Edit_Image



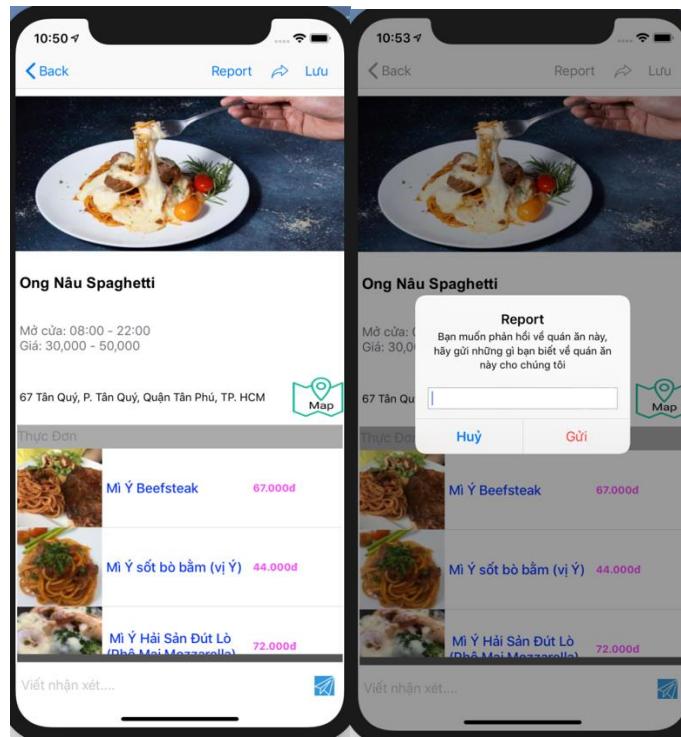
Hình 37 BM>Edit_Image

- BM_Logout



Hình 38 BM_Logout

- **BM_Report_Store**



Hình 39 BM_Report_Store

CHƯƠNG 5: HIỆN THỰC VÀ KIỂM THỬ HỆ THỐNG

5.1 CÀI ĐẶT

Sử dụng Xcode version 10.1 với ngôn ngữ Swift 3

Mongodb version 4.0.4

Firebase version 8.2

Nodejs version 8.12.0

5.2 KIỂM THỬ

Bảng 27 TestCase ADMIN

| Nội dung test | Testcase Id | Mô tả |
|----------------|---------------|--|
| TESTCASE ADMIN | | |
| Thêm cửa hàng | ADDSTORE_1 | Kiểm tra khi admin điền thông tin URL HINH ANH, Tên Cua Hang, Địa chỉ, Vi Do, Kinh Do, Bài Viết Giới Thiệu |
| | ADDSTORE_2 | Kiểm tra khi admin điền thiếu thông tin của cửa hàng |
| Xoá cửa hàng | DELETESTORE_1 | Kiểm tra khi admin điền id cửa hàng |
| | DELETESTORE_2 | Kiểm tra khi admin chưa điền vào id cửa hàng |
| Sửa cửa hàng | EDITSTORE_1 | Kiểm tra khi admin điền URL HINH ANH, ID, Tên cua hàng, địa chỉ, vi do, kinh do , bài viết giới thiệu |
| | EDITSTORE_2 | Kiểm tra khi admin điền thiếu thông tin của id cửa hàng |
| Thêm món ăn | ADDDISH_1 | Kiểm tra khi admin điền URL HINH ANH, ID cửa hàng, Monan, gia |

| | | |
|-----------------------|-----------------|--|
| | ADDDISH_2 | Kiểm tra khi admin điền thiếu thông tin của món ăn |
| Xoá món ăn | DELETEDISH_1 | Kiểm tra khi admin điền ID cửa hàng, ID món ăn |
| | DELETEDISH_2 | Kiểm tra khi admin điền thiếu thông tin của món ăn |
| Sửa món ăn | EDITDISH_1 | Kiểm tra khi admin điền ID cửa hàng, id món ăn, URL HINH ANH, tên món ăn,gia |
| | EDITDISH_2 | Kiểm tra khi admin điền thiếu thông tin của món ăn |
| Xem bài viết của User | APPROVED_POST_1 | Kiểm tra khi admin điền đủ thông tin của bài viết |
| | APPROVED_POST_2 | Kiểm tra khi admin điền thiếu dữ liệu |

Bảng 28 TestCase User

| Nội dung test | Testcase Id | Mô tả |
|---------------|--------------------|---|
| TESTCASE USER | | |
| Đăng nhập | LOGIN_1 | Kiểm tra khi người dùng đăng nhập bằng email mà chưa điền thông tin email và mật khẩu |
| | LOGIN_2 | Kiểm tra khi người dùng đăng nhập khi nhập sai email hoặc mật khẩu |
| | LOGIN_3 | Kiểm tra khi người dùng điền thông tin email và mật khẩu |
| Tìm kiếm | SEARCH_NAMESTORE_1 | Kiểm tra khi người dùng điền đúng tên quán |

| | | |
|----------------------------------|-----------------------|---|
| | SEARCH_NAMESTORE_2 | Kiểm tra khi người dùng điền không đúng tên quán |
| | SEARCH_ADDRESS_1 | Kiểm tra khi người dùng điền đúng địa chỉ |
| | SEARCH_ADDRESS_2 | Kiểm tra khi người dùng điền không đúng địa chỉ |
| Danh sách các quán ăn | LIST_STORE_1 | Kiểm tra khi người dùng xem danh sách các cửa hàng |
| Danh sách nội dung trong quán ăn | LIST_STORE_CONTENTS_1 | Kiểm tra khi người dùng xem bản đồ hướng dẫn đường đi, danh sách các món ăn |
| Danh sách các quán ăn đã lưu | LIST_STORE_SAVE_1 | Kiểm tra khi người dùng xem danh sách các quán ăn đã lưu |
| Danh sách Thông báo | LIST_NOTIFICATION_1 | Kiểm tra khi người dùng xem thông báo của app |
| Bình luận vào quán ăn | COMMENTS_STORE_1 | Kiểm tra khi người dùng bình luận vào quán ăn |
| | COMMENTS_STORE_2 | Kiểm tra khi người dùng không bình luận vào quán ăn |
| Đăng bài viết về quán ăn | POST_STORE_1 | Kiểm tra khi người dùng điền thông tin đầy đủ |
| | POST_STORE_2 | Kiểm tra khi người dùng nhập thiếu thông tin |
| Chia sẻ quán ăn | SHARE_STORE_1 | Kiểm tra khi người dùng chia sẻ quán ăn |
| Chỉnh sửa thông tin cá nhân | EDIT_PROFILE_1 | Kiểm tra khi người dùng chỉnh sửa thông tin cá nhân khi có nội dung chỉnh sửa |
| | EDIT_PROFILE_2 | Kiểm tra khi người dùng chỉnh sửa thông tin cá nhân khi có không nội dung chỉnh sửa |
| Đăng xuất | LOG_OUT_1 | Kiểm tra khi người dùng đăng xuất |

Bảng 29 TestCase GUEST

| Nội dung test | Testcase Id | Mô tả |
|----------------------------------|-----------------------|---|
| TESTCASE GUEST | | |
| Đăng ký | REGISTER_1 | Kiểm tra khi người dùng đăng ký khi điền đầy đủ thông tin Tên, email, mật khẩu, hình ảnh |
| | REGISTER_2 | Kiểm tra khi người dùng đăng ký khi điền chưa đầy đủ thông tin Tên, email, mật khẩu, hình ảnh |
| Danh sách các quán ăn | LIST_STORE_1 | Kiểm tra khi người dùng xem danh sách các cửa hàng |
| Danh sách nội dung trong quán ăn | LIST_STORE_CONTENTS_1 | Kiểm tra khi người dùng xem bản đồ hướng dẫn đường đi, danh sách các món ăn |
| Danh sách Thông báo | LIST_NOTIFICATION_1 | Kiểm tra khi người dùng xem thông báo của app |

CHƯƠNG 6: PHẦN KẾT LUẬN

6.1 KẾT QUẢ ĐẠT ĐƯỢC

- Tạo ra một ứng dụng tìm kiếm và lưu lại cái quán ăn đã được admin thêm vào.
- Có kiến thức về cách xử lý database của mongodb và xử lý được Firebase build lên điện thoại thật.
- Có kiến thức về server: Tạo được một Virtual Private Server(VPS) và đưa database mongodb lên.
- Vững kiến thức về lập trình swift.

6.2 ƯU , NHƯỢC ĐIỂM

6.2.1 Ưu điểm

- Ứng dụng dễ dàng tương tác.

6.2.2 Nhược điểm:

- Ứng dụng sử dụng vps yếu vì vậy chạy còn chậm.
- Thao tác nhanh sẽ bị bug thoát khỏi ứng dụng.
- Chưa kiểm soát triệt để lỗi trong ứng dụng.

6.3 HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

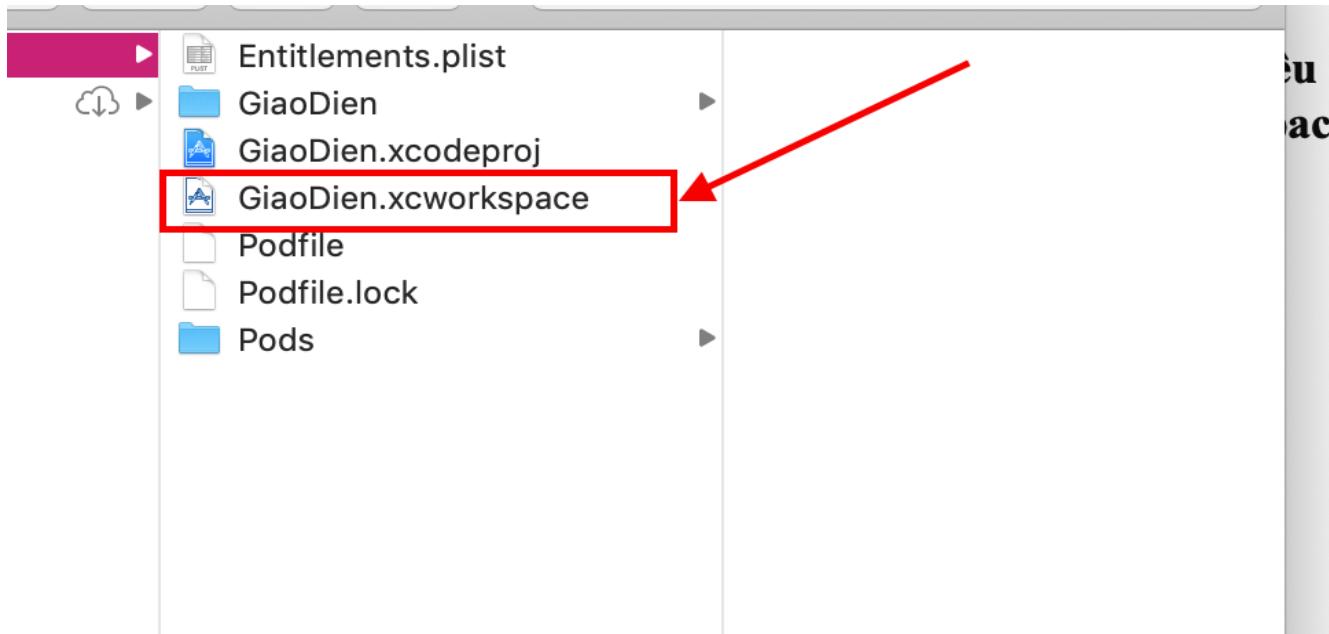
- Ứng dụng đã thực hiện được lưu bài viết, và bỏ lưu bài viết, và đăng được bài viết của mình. Từ đó có thể phát triển lên cho phép người dùng có tài khoản có thể:
 - o Đánh giá bài viết
 - o Sửa hoặc xóa đi bài viết của mình.
- Phát triển đại hàng order online và đặt chỗ trên ứng dụng.
- Phát triển thành mạng xã hội cho người dùng tương tác.

TÀI LIỆU THAM KHẢO

- <https://stackoverflow.com>
- <https://www.mongodb.com/>
- <https://mongoosejs.com/>
- <https://docs.mongodb.com/>
- <https://firebase.google.com/docs/ios/setup?authuser=0>
- <https://code.tutsplus.com/vi/articles/an-introduction-to-mongoose-for-mongodb-and-nodejs--cms-29527>
- <https://code.tutsplus.com/vi/articles/an-introduction-to-mongoose-for-mongodb-and-nodejs--cms-29527>

HƯỚNG DẪN CÀI ĐẶT

- Tải file source code về. Yêu cầu phải chạy trên máy macbook
- Mở file có đuôi .xcworkspace



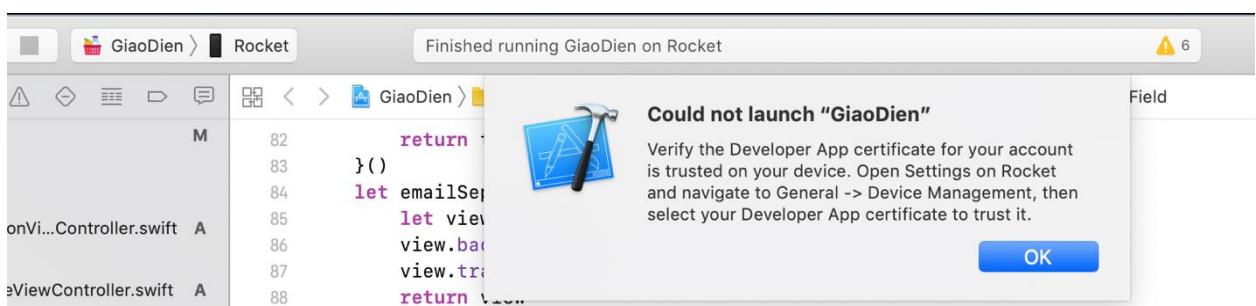
Hình 40 Mở file source code

- Kết nối iphone với macbook, sau đó build ứng dụng lên



Hình 41 Hình Chạy ứng dụng

- Đối với thiết bị chạy lần đầu tiên sẽ xuất hiện lỗi



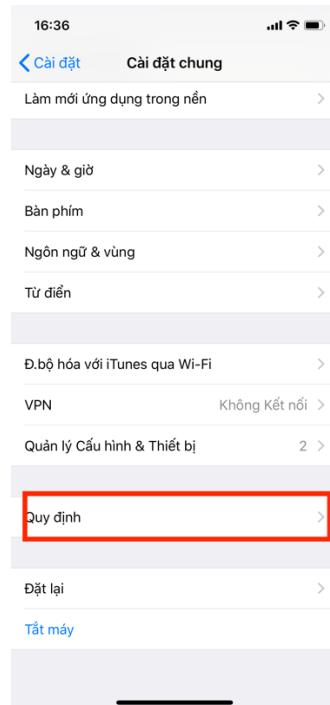
Hình 42 Lỗi chạy ứng dụng lần đầu tiên

- Bấm ok sau đó vào thiết bị iphone làm cái bước sau

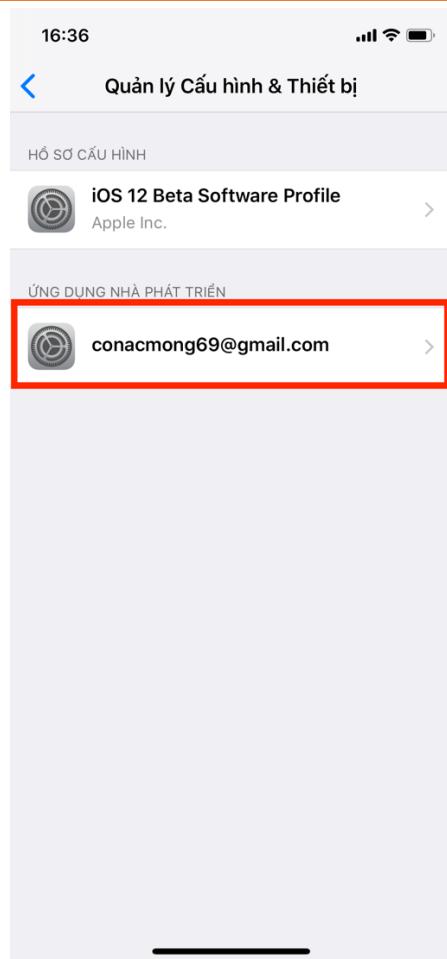
Tìm hiểu và xây dựng ứng dụng tìm kiếm quán ăn



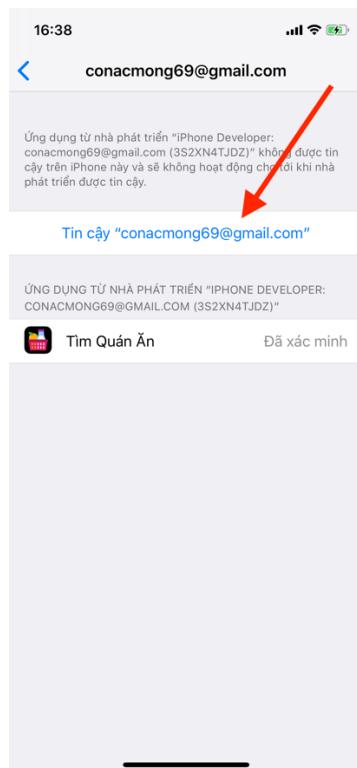
Hình 43 sửa lỗi bước 1



Hình 44 Sửa lỗi bước 2

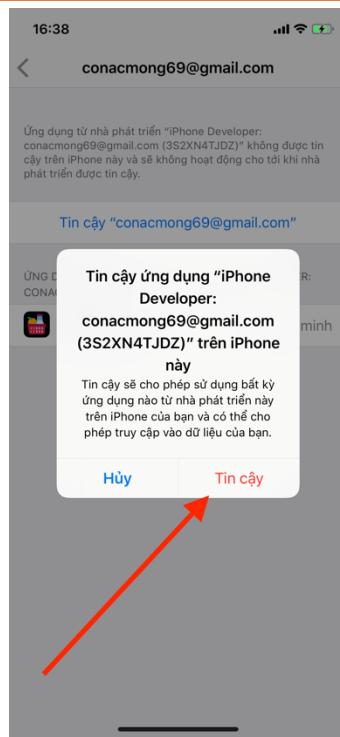


Hình 45 Sửa lỗi bước 3



Hình 46 Sửa lỗi bước 4

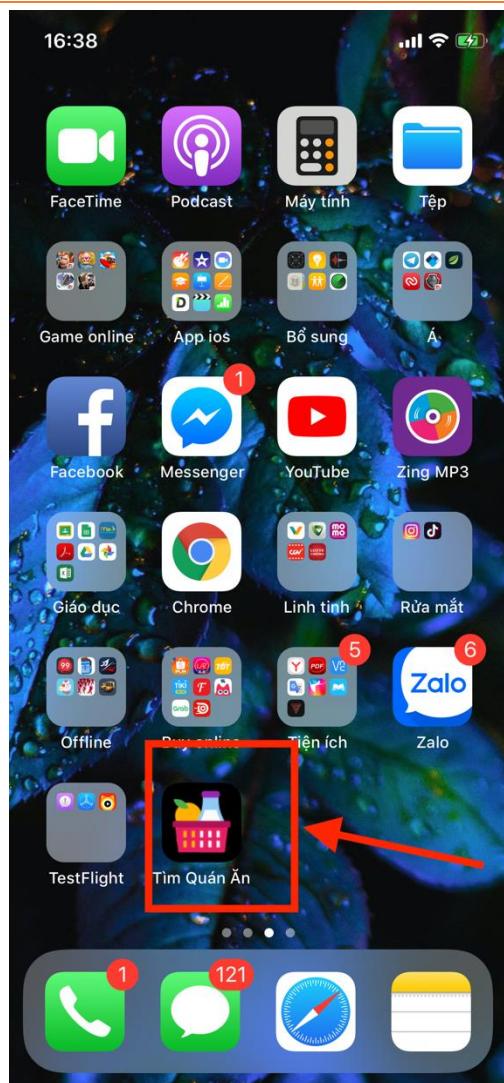
Tìm hiểu và xây dựng ứng dụng tìm kiếm quán ăn



Hình 47 sửa lỗi bước 5



Hình 48 sửa lỗi bước 6



Hình 49 Sửa lỗi bước 7

- Sau đó bấm command+R để chạy lại ứng dụng lên iphone của mình.

Ngày tháng năm 2019

Người viết đè cương

Ý kiến của giáo viên hướng dẫn