

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

ĐỀ TÀI : XÂY DỰNG HỆ THỐNG GIAO ĐỒ ĂN

Giảng viên hướng dẫn :

- Nguyễn Đình Ái
- Vũ Văn Vinh

Sinh viên thực hiện :

1.2001181383- *Vũ Minh Trung- 09DHTH6*

2.2001181020-*Nguyễn Chánh Bảo-09DHTH1*

3.2001181393-*Huỳnh Tăng Anh Tú-09DHTH4*

TP. HỒ CHÍ MINH – tháng ... năm

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

ĐỀ TÀI : XÂY DỰNG HỆ THỐNG GIAO ĐỒ ĂN

Giảng viên hướng dẫn :

- Nguyễn Đình Ái
- Vũ Văn Vinh

Sinh viên thực hiện :

1.2001181383- *Vũ Minh Trung- 09DHTH6*

2.2001181020-*Nguyễn Chánh Bảo-09DHTH1*

3.2001181393-*Huỳnh Tăng Anh Tú-09DHTH4*

TP. HỒ CHÍ MINH – tháng ... năm

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi, nội dung tham khảo từ các công trình khác đều được trích dẫn rõ ràng. Các số liệu, kết quả nêu trong khóa luận là trung thực và chưa từng công bố trong bất kỳ công trình nào khác.

Sinh viên thực hiện luận văn

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Quá trình thực hiện luận văn tốt nghiệp là giai đoạn quan trọng nhất trong quãng đời mỗi sinh viên. Khóa luận tốt nghiệp là tiền đề nhằm trang bị cho chúng em những kỹ năng nghiên cứu, những kiến thức quý báu trước khi lập nghiệp.

Trước hết, chúng em xin chân thành cảm ơn quý Thầy, Cô khoa Công nghệ thông tin đã tận tình chỉ dạy và trang bị cho em những kiến thức cần thiết trong suốt thời gian ngồi trên ghế giảng đường. Làm nền tảng cho em có thể hoàn thành được bài khóa luận này.

Em xin trân trọng cảm ơn thầy Vũ Đình Ái và thầy Vũ Văn Vinh đã tận tình giúp đỡ, định hướng cách tư duy và cách làm việc khoa học. Đó là những góp ý hết sức quý báu không chỉ trong quá trình thực hiện luận văn này mà còn là hành trang tiếp bước cho em trong quá trình học tập và lập nghiệp sau này.

Và cuối cùng, xin gửi lời cảm ơn đến gia đình, bạn bè, tập thể khóa 09DHTH, những người luôn sẵn sàng sẻ chia và giúp đỡ trong học tập và cuộc sống. Mong rằng, chúng ta sẽ mãi mãi gắn bó với nhau.

Xin chúc những điều tốt đẹp nhất sẽ luôn đồng hành cùng mọi người.

Vũ Minh Trung

Nguyễn Chánh Bảo

Huỳnh Tăng Anh

MỞ ĐẦU

Dưới sự tác động của cuộc cách mạng công nghiệp 4.0 hiện nay đã ảnh hưởng nhiều đến các cách thức hoạt động mới của các cửa hàng, trung tâm thương mại lẫn cả thị trường Việt Nam có sự nhiều thay đổi tích cực và linh hoạt hơn nhiều so với lúc trước. Nắm bắt cơ hội đó mà nhiều hoạt động kinh doanh, du lịch, vui chơi giải trí, ăn uống, làm đẹp, ... cũng không ngừng mọc lên nhằm đáp ứng sự gia tăng của nhu cầu người dùng hơn.

Và cũng chính vì thế mà một hoạt động kinh doanh từ xưa đến nay không bao giờ lỗi thời, nó đã xuất hiện từ năm 2016 và đến ngày nay nhu cầu về nó không những không giảm mà còn tăng nhiều hơn nữa, hay nói cách khác đó chính là thị trường đặt đồ ăn trực tuyến đang càng ngày càng phát triển mạnh mẽ. Vì vậy mà các ứng dụng, phần mềm, dịch vụ đặt đồ ăn trực tuyến đang xuất hiện mọi lúc mọi nơi.

Chính vì bài toán này mà nhóm đã chọn Cross Platform Flutter cho thiết bị mobile nhằm hỗ trợ nhu cầu của người dùng từ đó giúp xây dựng một ứng dụng mobile cho cả hai nền tảng phổ biến nhất hiện nay là IOS và Android một cách nhanh nhất và tiết kiệm chi phí

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
UI	User Interface	Giao diện người dùng
AOT	Ahead Of Time	Trước thời hạn
JIT	Just In Time	Đúng thời hạn
QoS	Quality of service	Chất lượng dịch vụ
SDK	Software Development Kit	Bộ công cụ phát triển phần mềm
FCM	Firebase Cloud Messaging	Thông báo qua đám mây của Firebase

DANH MỤC CÁC HÌNH VẼ

Hình 1. 1 Công nghệ Flutter.....	14
Hình 1. 2 Công nghệ Angular	16
Hình 1. 3 Công nghệ mongoDB.....	19
Hình 1. 4 Công nghệ Firebase.....	21
Hình 1. 5 Quy trình hoạt động FCM.....	22
Hình 1. 6 Công nghệ nodejs.....	22
Hình 1. 7 Mô hình hoạt động Expressjs.....	23
Hình 1. 8 Biểu mẫu shoppeeFood	27
Hình 1. 9 Biểu mẫu shoppeeFood	28
Hình 2. 1 Sơ đồ use-case phần ứng dụng khách hàng.....	30
Hình 2. 2 Mô hình hệ thống của đối tác / admin.....	31
Hình 2. 3 Sơ đồ hoạt động đặt hàng.....	32
Hình 2. 4 Sơ đồ hoạt động cấp mật khẩu	32
Hình 2. 5 Sơ đồ hoạt động cập nhật cửa hàng yêu thích.....	33
Hình 2. 6 Sơ đồ hoạt động tra cứu thông tin	34
Hình 2. 7 Sơ đồ hoạt động quản lý menu.....	35
Hình 2. 8 Sơ đồ hoạt động cập nhật thông tin đặt hàng	35
Hình 2. 9 Sơ đồ hoạt động đăng ký đối tác	36
Hình 2. 10 Sơ đồ hoạt động quản lý của đối tác	36
Hình 2. 11 Sơ đồ hoạt động quản lý của admin	37
Hình 2. 12 Sơ đồ mức phân tích.....	45
Hình 3. 1 Lược đồ csdl thiết kế bằng moon moderler.....	47
Hình 3. 2 Sơ đồ lớp mức thiết kế	50
Hình 3. 3 Giao diện đăng nhập/đăng ký.....	51
Hình 3. 4 Giao diện cung cấp lại mật khẩu	52
Hình 3. 5 Giao diện trang chủ	53
Hình 3. 6 Giao diện quản lý đơn hàng	54
Hình 3. 7 Giao diện theo dõi cửa hàng yêu thích.....	54
Hình 3. 8 Giao diện thông báo và tài khoản.....	55
Hình 3. 9 Giao diện của cửa hàng	56
Hình 3. 10 Giao diện đặt đơn hàng	57
Hình 3. 11 Giao diện đăng nhập.....	58
Hình 3. 12 Giao diện đăng ký	58
Hình 3. 13 Giao diện dashboard của đối tác	59
Hình 3. 14 Giao diện quản lý hóa đơn	59
Hình 3. 15 Giao diện quản lý hóa đơn	60
Hình 3. 16 Giao diện quản lý menu	60

Hình 3. 17 Giao diện quản lý menu	61
Hình 3. 18 Giao diện thông tin của đối tác	61
Hình 3. 19 Giao diện dashboard của admin	62
Hình 3. 20 Giao diện quản lý khách hàng	62
Hình 3. 21 Giao diện quản lý khách hàng	63
Hình 3. 22 Giao diện quản lý nhân viên.....	63
Hình 3. 23 Giao diện quản lý cửa hàng.....	64
Hình 3. 24 Giao diện thông tin cá nhân	64
Hình 3. 25 Sơ đồ chức năng (3 lớp) cập nhật thông tin khách hàng.....	65
Hình 3. 26 Sơ đồ tuần tự của chức năng cập nhật thông tin.	66
Hình 3. 27 Sơ đồ chức năng (3 lớp) đặt hàng	67
Hình 3. 28 Sơ đồ tuần tự của chức năng đặt hàng.	67
Hình 3. 29 Sơ đồ chức năng (3 lớp) thêm cửa hàng yêu thích.....	68
Hình 3. 30 Sơ đồ tuần tự của chức năng thêm sản phẩm yêu thích	68
Hình 3. 31 Sơ đồ chức năng (3 lớp) thêm giỏ hàng	69
Hình 3. 32 Mô hình tuần tự của chức năng thêm vào giỏ hàng.	69
Hình 3. 33 Sơ đồ chức năng (3 lớp) thêm tài khoản	70
Hình 3. 34 Sơ đồ tuần tự của chức năng đăng ký.	70
Hình 3. 35 Sơ đồ chức năng (3 lớp) tra cứu.....	71
Hình 3. 36 Sơ đồ tuần tự của chức năng tra cứu.	71
Hình 3. 37 Sơ đồ tuần tự của chức năng đăng ký.	72
Hình 3. 38 Sơ đồ tuần tự của chức năng cập nhật thông tin.	73
Hình 3. 39 Sơ đồ tuần tự của chức năng quản lý tương tự.....	74

DANH MỤC CÁC BẢNG

Bảng 1. 1 So sánh flutter với React Native	16
Bảng 1. 2 So sánh angular & react	19
Bảng 2. 1 Đặc tả UC đăng ký dịch vụ	39
Bảng 2. 2 Đặc tả UC tra cứu thông tin	39
Bảng 2. 3 Đặc tả UC đặt hàng	40
Bảng 2. 4 Đặc tả UC thanh toán đơn hàng	41
Bảng 2. 5 Đặc tả UC thay đổi thông tin cá nhân	41
Bảng 2. 6 Thay đổi của hàng yêu thích	42
Bảng 2. 7 Đặc tả UC đăng ký tài khoản (admin/đối tác)	43
Bảng 2. 8 Đặc tả UC cập nhật thông tin (admin/đối tác)	44
Bảng 2. 9 Đặc tả UC quản lý (admin/đối tác)	45
Bảng 3. 1 Mô tả Collection Store	48
Bảng 3. 2 Mô tả Collection Customer	48
Bảng 3. 3 Mô tả Collection Order	48
Bảng 3. 4 Mô tả Document Contact	48
Bảng 3. 5 Mô tả Document Food	49
Bảng 3. 6 Mô tả Document Coupon	49
Bảng 3. 7 Mô tả Document Employee	49
Bảng 3. 8 Mô tả Document Address	49
Bảng 3. 9 Mô tả Document review	49
Bảng 3. 10 GET API	75
Bảng 3. 11 POST API	75
Bảng 3. 12 PUT API	75
Bảng 3. 13 DELETE API	76

MỤC LỤC

LỜI CẢM ƠN	4
MỞ ĐẦU	5
DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT	6
DANH MỤC CÁC HÌNH VẼ.....	7
DANH MỤC CÁC BẢNG	9
CHƯƠNG 1.....	12
1.1. GIỚI THIỆU :.....	12
1.2. CÁC HỆ THỐNG LIÊN QUAN :.....	12
1.3. MỤC TIÊU VÀ PHẠM VI ĐỀ TÀI :	12
1.3.1 Mục tiêu đề tài :	12
1.3.2 Phạm vi đề tài :	13
1.4. ĐỐI TƯỢNG NGHIÊN CỨU :.....	13
1.5. NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU :	13
1.6. CÔNG CỤ, THIẾT BỊ SỬ DỤNG:.....	14
1.6.1 Flutter :.....	14
1.6.2 Angular :	16
1.6.3 MongoDB:	19
1.6.4 Firebase:	21
1.6.5 NodeJS:.....	22
1.7. KHẢO SÁT HỆ THỐNG :	23
1.7.1 Giới thiệu sơ lược về hệ thống :	23
1.7.2 Các quy trình nghiệp vụ :.....	24
1.7.3 Các biểu mẫu thu thập được :.....	27
1.8. TỔNG KẾT HỆ THỐNG :.....	28
CHƯƠNG 2.....	29
2.1 GIỚI THIỆU :.....	29
2.2 MÔ HÌNH HÓA CHỨC NĂNG	29
2.2.1 Sơ đồ Use Case hệ thống.....	29
2.2.2 Sơ đồ Activity :	31

2.2.3	Đặc tả use case hệ thống	37
2.3	SƠ ĐỒ LỚP MỨC PHÂN TÍCH	45
2.4	TỔNG KẾT CHƯƠNG.....	45
CHƯƠNG 3	47
3.1	GIỚI THIỆU	47
3.2	THIẾT KẾ CSDL	47
3.3	SƠ ĐỒ LỚP MỨC THIẾT KẾ	50
3.4	THIẾT KẾ GIAO DIỆN HỆ THỐNG	51
3.5	THIẾT KẾ CHỨC NĂNG HỆ THỐNG	65
3.6	THIẾT KẾ API :.....	75
CHƯƠNG 4	81
4.1	KẾT QUẢ ĐẠT ĐƯỢC :.....	81
4.2	ĐÁNH GIÁ KẾT QUẢ :	82
TÀI LIỆU THAM KHẢO	83

CHƯƠNG 1

TỔNG QUAN

1.1. GIỚI THIỆU :

Trước đây khi hình thức mua bán, đặt hàng trực tuyến chưa được ứng dụng rộng rãi thì các công việc như bán hàng, mua hàng trực tiếp thì rất tốn thời gian và tài chính cũng như sự thỏa mãn được các nhu cầu từ khách hàng cũng bị hạn chế. Ngày nay nhờ sự phát triển của công nghệ thông tin thì đã giúp cho mọi người kết nối với mạng được gần gũi và mạnh mẽ hơn bao giờ hết. Cũng nhờ đó mà hình thức mua bán đặt hàng trực tuyến xuất hiện để giúp cho các cửa hàng vừa và nhỏ có thêm thu nhập và đồng thời cũng giúp cho các khách hàng thỏa mãn được nhu cầu. Chính vì vậy chúng em quyết định chọn đề tài “Xây dựng hệ thống giao đồ ăn”.

Hiện nay việc sử dụng ứng dụng đặt hàng từ các cửa hàng giao hàng trực tuyến đã không còn xa lạ với các hành khách. Bây giờ việc sử dụng ứng dụng và phần mềm đã trở thành một phần trọng yếu để phục vụ nhu cầu đời sống của khách hàng.

Do đó đòi hỏi phải có một hệ thống giao đồ ăn để có thể phục vụ khách hàng mọi lúc mọi nơi nhằm tiết kiệm thời gian và chi phí.

1.2. CÁC HỆ THỐNG LIÊN QUAN :

Ngày nay, nước chúng ta đang sống trong thời đại mà công nghệ thông tin đang trên đà phát triển mạnh mẽ, nhờ đó mà các hình thức mua bán, kinh doanh được phổ biến rộng rãi và phát triển hơn bao giờ hết. Đặc biệt là các hình thức kinh doanh, mua bán trực tuyến trên mạng.

Ngoài ra nước chúng ta đang chịu vấn nạn Covid-19 nên các hoạt động buôn bán, mua hàng trực tuyến là giải pháp thực sự cần thiết trong giai đoạn này. Có thể kể đến các cửa hàng mua bán trực tuyến đang nổi như : Lazada , Shopee, Sendo, Gojek, Now, Baemin...

1.3. MỤC TIÊU VÀ PHẠM VI ĐỀ TÀI :

1.3.1 Mục tiêu đề tài :

Khảo sát, đánh giá được hiện trạng của các ứng dụng giao hàng thực tế. Xây dựng một hệ thống phù hợp, dễ hiểu, dễ sử dụng cho người dùng phần mềm.

Hiểu được hệ thống quản lý bán hàng, cần quản lý những mảng nào để xây dựng ứng dụng cho phù hợp nhu cầu thực tiễn. Cần phải hiểu rõ và nắm bắt được các công việc hằng ngày của việc đặt đồ ăn trực tuyến

Xác định, phân tích rõ ràng các yêu cầu hệ thống, yêu cầu chức năng, quy trình nghiệp vụ

Thiết kế giao diện hệ thống bố cục tiện lợi, dễ sử dụng, dễ nhìn

Giúp bản thân có thêm khả năng sáng tạo, tư duy thông qua thiết kế giao diện tương tác với người dùng, vận dụng kiến thức bản thân vào thực tiễn.

Tiến hành và cài đặt hệ thống với các chức năng đã phân tích

Cuối cùng xây dựng tài liệu hệ thống và cách sử dụng dành cho người dùng

1.3.2 Phạm vi đề tài :

Đề tài được nghiên cứu dựa trên các ứng dụng đặt đồ ăn trực tuyến như Delivery Food ,NowFood,Baemin và các tài liệu trên web .

Từ đó áp dụng vào các mô hình vừa và nhỏ một cách hiệu quả và tối ưu. Nhóm nghiên cứu đề tài sẽ bám sát và tuân theo nghiệp vụ hoạt động của các ứng dụng giao hàng trực tuyến , đảm bảo đầy đủ các chức năng chính của phần mềm và khai thác các chức năng mới có liên quan nhằm giúp cho phần mềm thêm phần tiện ích.

1.4. ĐỐI TƯỢNG NGHIÊN CỨU :

Đối tượng nghiên cứu là các khách hàng sử dụng phần mềm đặt đồ ăn. Chủ yếu là các khách hàng ở trên toàn thành phố Hồ Chí Minh cũng như các cửa hàng đồ ăn trên cùng khu vực.

1.5. NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU :

Đọc tài liệu tham khảo trước khi khảo sát thực tế để lấy thông tin. Đọc tài liệu qua các trang mạng, video,... là bước đầu tiên và rất quan trọng với mỗi đề tài. Để cung cấp thông tin tổng quan trước cho quá trình thiết kế phần mềm.

Bước tiếp theo mà chúng em tiến hành tìm hiểu các nghiệp vụ mà một phần mềm đặt đồ ăn cần phải có. Tham khảo các giao diện, chức năng, nội dung của các ứng dụng mobile và website quản lý liên quan đến đề tài.

Tiến hành tìm hiểu và khảo sát thực tế các ứng dụng mobile liên quan đến đặt đồ ăn đã có sẵn trên thị trường như GrabFood, ShopeeFood, Baemin,...

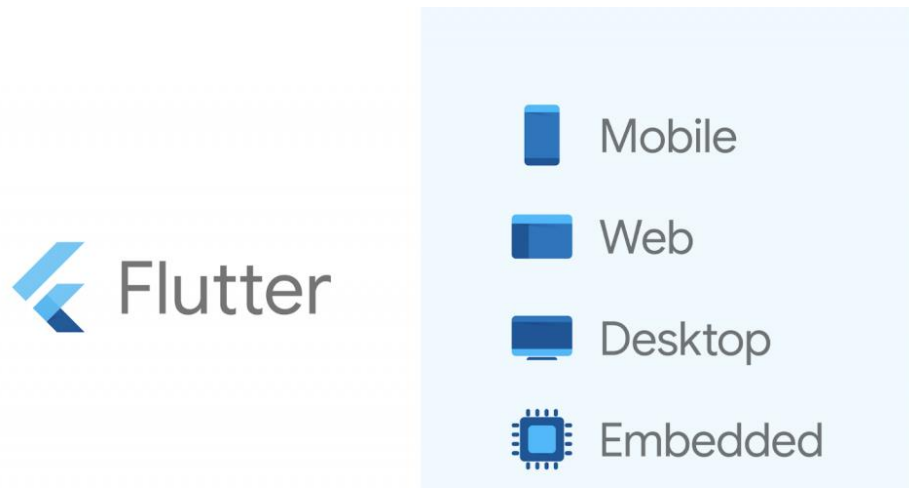
Sau khi tiến hành tìm hiểu và tham khảo chúng em sẽ tiến hành phân tích, xử lý thông tin thu được, bổ sung thêm thông tin còn thiếu. Sau đó bắt đầu triển khai thực hiện thiết kế phần mềm.

1.6. CÔNG CỤ, THIẾT BỊ SỬ DỤNG:

1.6.1 Flutter :

Flutter là mobile UI framework của Google để tạo ra các giao diện chất lượng cao trên IOS và Android trong khoảng thời gian ngắn. Flutter hoạt động với những code sẵn có được sử dụng bởi các lập trình viên, các tổ chức.

Flutter hoàn toàn miễn phí và cũng là mã nguồn mở.



Hình 1. 1 Công nghệ Flutter

a. Đặc điểm nổi bật :

Fast Development: Tính năng Hot Reload hoạt động trong milliseconds để hiển thị giao diện tới bạn. Sử dụng tập hợp các widget có thể customizable để xây dựng giao diện trong vài phút. Ngoài ra Hot Reload còn giúp bạn thêm các tính năng, fix bug tiết kiệm thời gian hơn mà không cần phải thông qua máy ảo, máy android hoặc IOS.

Expressive and Flexible UI: Có rất nhiều các thành phần để xây dựng giao diện của Flutter vô cùng đẹp mắt theo phong cách Material Design và Cupertino, hỗ trợ nhiều các APIs chuyển động, smooth scrolling...

Native Performance: Các widget của flutter kết hợp các sự khác biệt của các nền tảng ví dụ như scrolling, navigation, icons, font để cung cấp một hiệu năng tốt nhất tới IOS và Android.

Sử dụng DART. Một ngôn ngữ lập trình hướng đối tượng do Google phát triển. DART là một static type language nên nó là AOT

b. So sánh giữa flutter và react native:

Cross Platform	Ưu điểm	Nhược điểm
Flutter	<ul style="list-style-type: none"> – Mạnh về hiệu ứng, hiệu năng rất cao. – Giao tiếp gần như trực tiếp với native – Static language nhưng với syntax hiện đại, compiler linh động giữa AOT (for archive, build prod) và JIT (for development, hot reload) – Có thể chạy được giả lập mobile ngay trên web, tiện cho development. Các metric measure performance được hỗ trợ sẵn giúp developer kiểm soát tốt hiệu năng của app. – Có thể dùng để build các bundle/framework gắn và app native để tăng hiệu năng. 	<ul style="list-style-type: none"> – Bộ render UI được team author gần như viết lại, không liên quan tới UI có sẵn của Framework native, dẫn đến memory sử dụng khá nhiều. – Phải học thêm ngôn ngữ DART, bloc pattern, DART Streaming – Dù đã phát hành 1.0 chính thức, tuy nhiên còn khá mới. Một số plugin rất quan trọng như Google Map vẫn còn đang phát triển, chưa stable.
React Native	<ul style="list-style-type: none"> – Thiên về development/hotfix nhanh (hot reload, bundle injection) – Sử dụng JS (quen thuộc với nhiều developer) và có thể share business logic codebase với frontend (js). – Phát hành bởi Facebook, họ dùng cho product của họ hàng ngày nên developer 	<ul style="list-style-type: none"> – Giao tiếp với native thông qua các bridge, dễ bị bottleneck nếu không được kiểm soát tốt. – Dùng JS nên mang theo các đặc điểm của JS: rất dễ làm nhưng cũng dễ sai, dẫn tới khó maintain về sau. – Hiệu năng. animation là điểm yếu của ReactNative, muốn làm tốt phải làm từ native, tăng js chỉ gọi vào,

	<p>hưởng lợi khá nhiều từ đây.</p> <ul style="list-style-type: none"> – Hiện tại đã rất nhiều thư viện, gần như đã rất đầy đủ cho các nhu cầu app thông dụng. 	<p>setup views. Tuy nhiên với các interactive animation thì rất đau khổ.</p> <ul style="list-style-type: none"> – Không thích hợp cho các app cần năng lực tính toán cao (hash, crypto, etc).
--	--	--

Bảng 1. 1 So sánh flutter với React Native

1.6.2 Angular :

Angular là một JavaScript framework mã nguồn mở được viết bằng TypeScript được duy trì bởi Google với chức năng chính là phát triển các ứng dụng đơn trang. Angular được ra mắt vào năm 2010, là một framework sử dụng kiến trúc Model-View-Controller (MVC), bao gồm có:

Model: Thành phần nhận các input từ Controller và quản lý dữ liệu. Hãy coi nó là một bộ khung.

View: Thành phần trình bày các dữ liệu nhận được. Hãy coi nó là code HTML.

Controller: Thành phần phản hồi lại với các input và gửi dữ liệu từ input đến Model. Hãy coi nó là Javascript.



Hình 1. 2 Công nghệ Angular

a. Đặc điểm nổi bật của Angular:

Là một phần thiết yếu của hệ sinh thái Javascript, Angular là công cụ lập trình web front-end phổ biến nhất hiện nay. Mỗi phiên bản Angular đều có rất nhiều tính năng và được thường xuyên nâng cấp để trở nên thông minh hơn.

Các tính năng chung

Liên nền tảng (Cross-platform): Với Angular bạn có thể phát triển các ứng dụng web cấp tiến (Progressive Web Applications – PWA). PWA cũng cung cấp trải nghiệm như ứng dụng (app-like) cho người dùng sử dụng các công nghệ web tân tiến. Tùy theo nhu cầu thì bạn có thể triển khai một ứng dụng gốc cũng như cấp tiến. Bộ công cụ phát triển phần mềm kết hợp được gọi là Ionic có thể đưa ứng dụng của bạn lên app store hoặc triển khai nó lên mobile web như một PWA. Bạn cũng có thể phát triển ứng dụng máy tính bằng Angular.

Tốc độ cao và tối ưu hiệu năng: Angular định nghĩa lại máy ảo JavaScript hiện đại bằng cách chuyển các template thành code. Bởi vậy, code viết tay của bạn cũng có thể được tận dụng trong framework này. Điều tuyệt vời nhất của Angular đó là bạn có thể render code thành HTML&CSS. Thời gian tải của Angular nhanh hơn bất kỳ front-end framework nào ở thời điểm hiện tại. Angular tải cực kỳ nhanh với bất kỳ router thành phần mới nào. Hơn nữa, code sẽ tự động được chia ra đối với mỗi khi người dùng load và yêu cầu render.

Các ứng dụng Angular dành cho mọi người: Angular là một nền tảng front-end kỳ diệu không chỉ cho phép lập trình các ứng dụng ấn tượng, mà nó còn có thể tạo ra các hoạt ảnh cao cấp để nâng cao trải nghiệm người dùng, Các API của Angular linh hoạt đến mức các lập trình viên có thể phát triển các bố trí giao diện phức tạp cũng như các hoạt ảnh với sử dụng ít code . Hơn nữa, với các framework kiểm thử đơn vị thông minh như Jasmine và Karma, bạn có thể sửa lại code bất kỳ lúc nào, Angular có hơn 11 module kiểm thử được tích hợp sẵn để đảm bảo code không dính lỗi.

Các tính năng kỹ thuật

Kiến trúc MVC: Kiến trúc này đã được đề cập đến trong phần đầu của bài viết. Dữ liệu của ứng dụng sẽ được quản lý bởi Model, View quản lý phần hiển thị dữ liệu. Controller sẽ đóng vai trò là bộ nối giữa lớp View và lớp Model. Thông thường trong kiến trúc MVC, bạn có thể chia nhỏ ứng dụng một cách tùy ý và viết code để nối nó lại với nhau.

Tuy nhiên trong Angular, lập trình viên chỉ cần chia nhỏ ứng dụng thành các phần của MVC và framework sẽ tự làm những công việc còn lại. Điều này sẽ tiết kiệm một cách đáng kể thời gian code.

Liên kết dữ liệu hai chiều: Một trong những tính năng hiệu quả nhất của Angular đó là hệ thống liên kết dữ liệu hai chiều. Lớp View thể hiện chính xác lớp Model và hai lớp này sẽ phối hợp đồng bộ với nhau một cách hoàn hảo. Nếu bạn thay đổi bất kỳ cái gì trong Model, người dùng có thể thấy rõ thay đổi đó trong View.

Framework ít code: Angular là một framework ít code so với các công nghệ front-end khác. Bạn không cần phải viết code riêng để kết nối các lớp MVC. Và bạn cũng không cần đến các code cụ thể để xem một cách thủ công. Các chỉ thị được tách biệt với code ứng dụng. Tất cả sẽ tiết kiệm công sức và thời gian cần thiết để phát triển ứng dụng.

CDK và Angular Material: Angular nâng cấp Bộ công cụ phát triển thành phần (Component Development Kit – CDK) theo mỗi phiên bản mới. Làm mới (Refreshing) và cuộn ảo (virtual scrolling) là các tính năng tiêu biểu nhất của Angular CDK. Nó cũng hỗ trợ cho việc tải động cũng như đỡ tải DOM để tạo ra một danh sách lớn các dữ liệu hiệu năng cao. Bạn có thể import ScrollingModule hay DragDropModule vào ứng dụng.

TypeScript: Với TypeScript, Angular cung cấp một trải nghiệm liền mạch cho các lập trình viên. Nó rất hiệu quả trong việc phát hiện lỗi. TypeScript cũng tự động gắn các thiết lập của tệp root để biên dịch dễ dàng hơn. So với JavaScript, TypeScript có các kiểu liệt kê, giao diện, các lớp generic, các kiểu kết hợp, các kiểu union/intersection, các bộ điều chỉnh truy cập đặc sắc hơn.

b. So sánh giữa angular và react :

Framework	Angular	React
Khái niệm	Kiến trúc component	Kiến trúc component
	DOM thật	DOM ảo
	SSR, CSR	SSR
Cài đặt	Tương đối dễ	Tương đối khó

Data binding	Data binding 2 chiều	Data binding 1 chiều
Quản lý dependency	Sử dụng công cụ xây dựng sẵn	Yêu cầu ReactDI để quản lý
Hỗ trợ ngôn ngữ	TypeScript, CoffeeScript, JavaScript	JSX, JavaScript, TypeScript
Thoải mái khi sử dụng	Ít thoải mái, phải tuân theo các quy ước	Thoải mái hơn
Khả năng tiếp cận	Khó khăn (yêu cầu kiến thức nâng cao về JS)	Dễ (yêu cầu kiến thức trung bình về JS)

Bảng 1. 2 So sánh angular & react

1.6.3 MongoDB:

MongoDB là một chương trình cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data).



Hình 1. 3 Công nghệ mongoDB

Lợi thế của MongoDB so với các cơ sở dữ liệu dạng quan hệ(RDBMS):

- Ít Schema hơn: MongoDB là một cơ sở dữ liệu dựa trên Document, trong đó một Collection giữ các Document khác nhau. Số trường, nội dung và kích cỡ của Document này có thể khác với Document khác.
- Cấu trúc của một đối tượng là rõ ràng.
- Không có các Join phức tạp.
- Khả năng truy vấn sâu hơn. MongoDB hỗ trợ các truy vấn động trên các Document bởi sử dụng một ngôn ngữ truy vấn dựa trên Document mà mạnh mẽ như SQL.
- MongoDB dễ dàng để mở rộng.
- Việc chuyển đổi/ánh xạ của các đối tượng ứng dụng đến các đối tượng cơ sở dữ liệu là không cần thiết.
- Sử dụng bộ nhớ nội tại để lưu giữ phần công việc, giúp truy cập dữ liệu nhanh hơn.

MongoDB Atlas là cloud database của MongoDB được ra mắt vào năm 2016 chạy trên AWS, Microsoft Azure và Google Cloud Platform.

MongoDB Atlas là giải pháp phần mềm Database as a Service (DBaaS) Provider với chức năng và chi phí phù hợp cho các doanh nghiệp từ nhỏ và vừa (SMEs) tới các doanh nghiệp lớn. Phần mềm MongoDB Atlas được đánh giá cao bởi cả người dùng lẫn chuyên gia trong lĩnh vực Database Software.

Dữ liệu trong mỗi Cluster ở Atlas được lưu trữ theo cơ chế Replication, với 3 nodes: 1 master (primary) và 2 slaves (secondary)

Thật dễ dàng để bắt đầu - sử dụng GUI đơn giản để chọn kích thước phiên bản phù hợp, vùng địa lý và các tính năng bạn cần. MongoDB Atlas cung cấp:

- Các tính năng bảo mật để bảo vệ quyền truy cập vào dữ liệu của bạn.
- Nhân rộng tích hợp để luôn sẵn sàng, chịu được sự cố hoàn toàn của trung tâm dữ liệu.
- Sao lưu và khôi phục kịp thời để bảo vệ chống lại sự cố dữ liệu.
- Giám sát chi tiết để giúp bạn biết khi nào cần mở rộng quy mô. Các phiên bản bổ sung có thể được cung cấp bằng cách nhấn nút.
- Bản vá tự động và nâng cấp bằng một cú nhấp chuột cho các phiên bản chính mới của cơ sở dữ liệu, cho phép bạn tận dụng các tính năng MongoDB mới nhất và tuyệt vời nhất.
- Lựa chọn các nhà cung cấp đám mây, khu vực và các tùy chọn thanh toán.

1.6.4 Firebase:

Firebase là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

Cụ thể là những giao diện lập trình ứng dụng API đơn giản. Mục đích nhằm tăng số lượng người dùng và thu lại nhiều lợi nhuận hơn.

Đặc biệt, còn là dịch vụ đa năng và bảo mật cực tốt. Firebase hỗ trợ cả hai nền tảng Android và IOS. Không có gì khó hiểu khi nhiều lập trình viên chọn Firebase làm nền tảng đầu tiên để xây dựng ứng dụng cho hàng triệu người dùng trên toàn thế giới. Bạn có thể tìm hiểu thêm về điện toán đám mây trong bài viết:



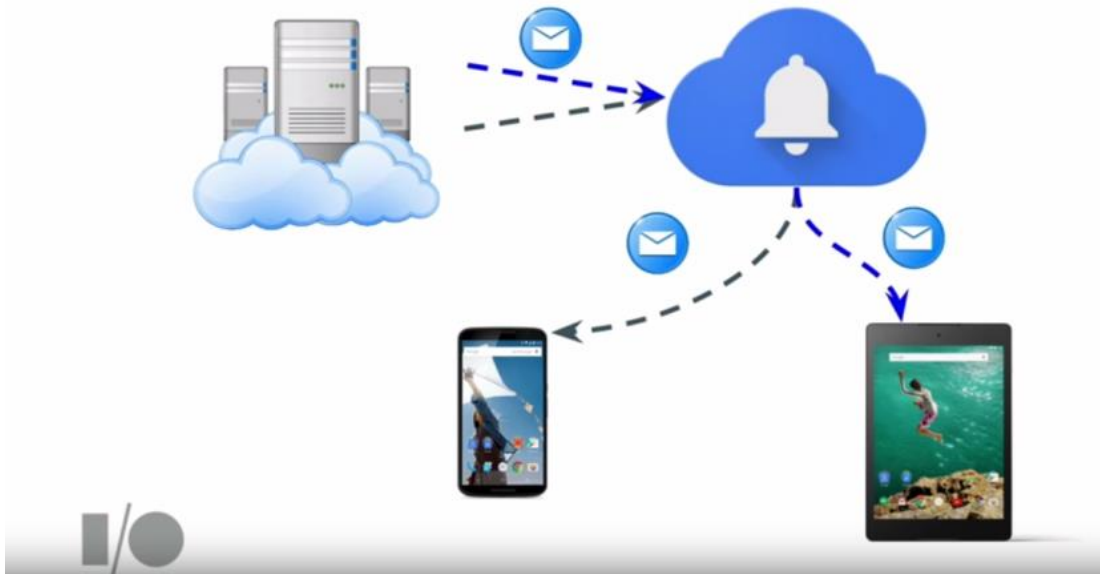
Hình 1. 4 Công nghệ Firebase

Firebase cloud messaging :

Firebase Cloud Messaging (FCM) là một dịch vụ miễn phí của Google. Thông qua FCM, nhà phát triển ứng dụng có thể gửi thông điệp một cách nhanh chóng, an toàn tới các thiết bị cài đặt ứng dụng.

Ưu điểm : thời gian triển khai nhanh, config server không nhiều, tiết kiệm chi phí.

Firebase cloud push notification hoạt động như thế nào?



Hình 1. 5 Quy trình hoạt động FCM

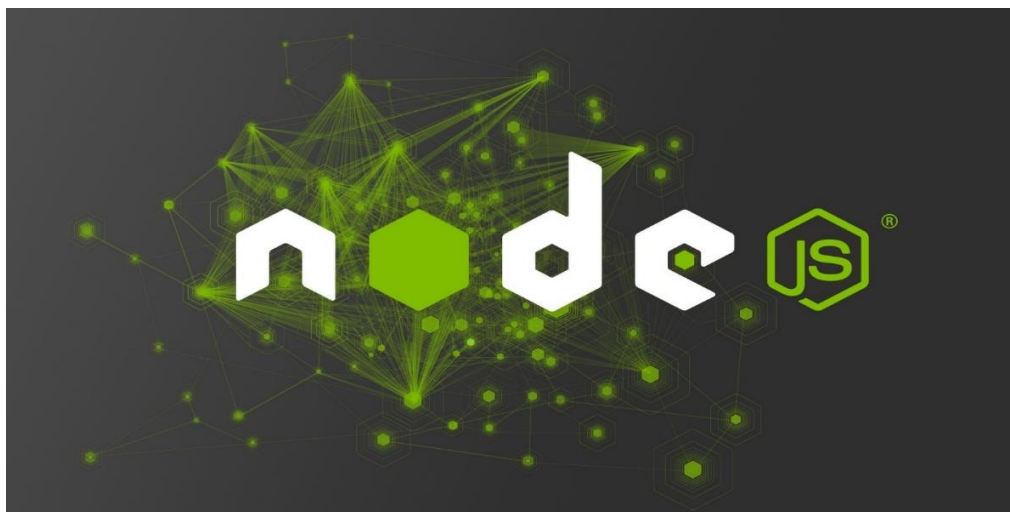
Hình trên mô tả luồng hoạt động của Firebase Cloud Messaging.

- Server sẽ soạn thảo tin nhắn cần thông báo tới người sử dụng ứng dụng.
- Firebase sẽ chịu trách nhiệm gửi tới các thiết bị cài đặt ứng dụng.

Như vậy, để các thiết bị có thể nhận được thông điệp, Firebase phải xác định thông qua một mã gọi là Token khi thiết bị cài đặt ứng dụng

1.6.5 NodeJS:

NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

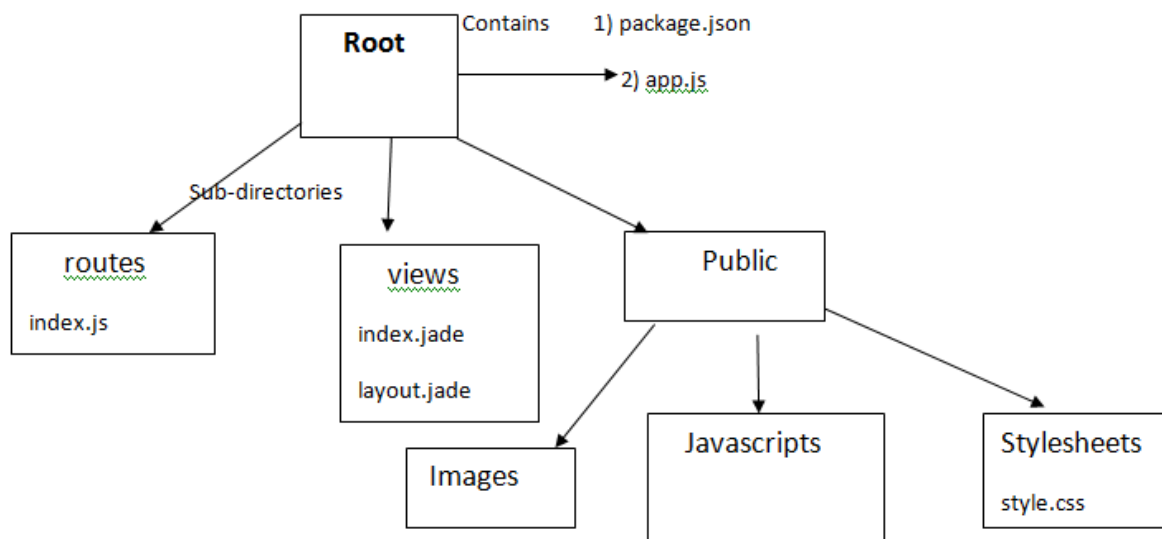


Hình 1. 6 Công nghệ nodejs

NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.

- NodeJS được viết bằng JavaScript với cộng đồng người dùng lớn mạnh. Nếu bạn cần hỗ trợ gì về NodeJS, sẽ nhanh chóng có người hỗ trợ bạn.
- Tốc độ xử lý nhanh. Nhờ cơ chế xử lý bất đồng bộ (non-blocking), NodeJS có thể xử lý hàng ngàn kết nối cùng lúc mà không gặp bất cứ khó khăn nào.
- Dễ dàng mở rộng. Nếu bạn có nhu cầu phát triển website thì tính năng dễ dàng mở rộng của NodeJS là một lợi thế cực kỳ quan trọng.

Expressjs là một framework được xây dựng trên nền tảng của **Nodejs**. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. **Expressjs** hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.



Hình 1. 7 Mô hình hoạt động Expressjs

Tổng hợp một số chức năng chính của Expressjs như sau:

- Thiết lập các lớp trung gian để trả về các HTTP request.
- Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số.

1.7. KHẢO SÁT HỆ THỐNG :

1.7.1 Giới thiệu sơ lược về hệ thống :

Dựa vào các hệ thống giao hàng trực tuyến nổi tiếng như ShopeeFood, Go, Baemin, Lazada mà nhóm chúng em đã tổng kết được các quy trình nghiệp vụ cần có bên phần ứng dụng của khách hàng và cả bên phần đối tác/quản trị.

Phần ứng dụng khách hàng bao gồm các quy trình :

- Quy trình khách hàng đăng ký sử dụng dịch vụ.
- Quy trình đặt đơn hàng.
- Quy trình tra cứu cửa hàng
- Quy trình thay đổi thông tin khách hàng.

Phần quản trị / đối tác bao gồm các quy trình :

- Quy trình đăng ký tài khoản.
- Quy trình quản lý thông tin.
- Quy trình cập nhật thông tin.

1.7.2 Các quy trình nghiệp vụ :

1.7.2.1 Quy trình đăng ký sử dụng dịch vụ (Khách hàng):

Quy trình thường diễn ra khi có khách hàng thực hiện sử dụng dịch vụ đăng ký vào ứng dụng . Đầu tiên xác nhận yêu cầu, ở đây sẽ bao gồm chủ yếu là 3 yêu cầu: yêu cầu tạo thành viên, yêu cầu cung cấp lại mật khẩu, yêu cầu đăng nhập. Yêu cầu tạo thành viên, hệ thống sẽ kiểm tra thông tin có phù hợp để tạo thành viên mới và kiểm tra xem gmail có tồn tại hay chưa , nếu chưa tồn tại thì hệ thống sẽ xác nhận thông tin và lưu trữ vào csdl,ngược lại thông báo lỗi. Yêu cầu thứ hai, khi hệ thống nhận yêu cầu cung cấp lại mật khẩu của khách hàng, hệ thống sẽ gửi mã xác nhận đến email đã đăng ký sử dụng của khách hàng và yêu cầu xác nhận thông qua 6 chữ ,nếu mã xác nhận không phù hợp hệ thống sẽ không cho phép khách hàng thay đổi mật khẩu và ngược lại hệ thống sẽ cho phép khách hàng cập nhập lại mật khẩu của mình. Yêu cầu thứ ba xảy ra khi khách hàng đã đăng ký tài khoản , lúc này khách hàng sẽ nhập thông tin vào form, nếu các thông tin hợp lệ hệ thống sẽ cho phép khách hàng vào sử dụng dịch vụ, ngược lại sẽ hiển thị thông báo lỗi .

1.7.2.2 Quy trình đặt đơn hàng (Khách hàng):

Quy trình thường diễn ra khi khách hàng có nhu cầu sử dụng dịch vụ như mua các thức ăn hoặc nước uống, các loại thực phẩm chức năng từ những cửa hàng trên khắp thành phố của bạn . Sau khi khách hàng đã cung

cấp các thông tin yêu cầu như tên, email ,điện thoại và cụ thể là địa chỉ thì hệ thống sẽ cho phép khách hàng đặt hàng trực tuyến . Khi khách hàng đặt hàng trên hệ thống , hệ thống sẽ hiển thị các món hàng mà khách hàng , tính toán tổng tiền mà khách hàng phải trả.Lúc này hệ thống sẽ yêu cầu khách hàng chọn thanh toán bằng tiền mặt hay bằng hình thức thẻ.Nếu khách hàng chọn thanh toán bằng thẻ thì hệ thống sẽ hiển thị mã để xác nhận và sau khi xác nhận xong thì đơn hàng của quý khách sẽ được tự động thanh toán trực tiếp trước khi nhận được hàng , ngược lại nếu chọn tiền mặt thì khách hàng sẽ thanh toán thông qua shipper của cửa hàng. Sau khi khách hàng đã xem và xác nhận những món hàng mình đã đặt thì lúc này hệ thống sẽ gửi thông báo cho cửa hàng đối tác trên hệ thống để cung cấp thông tin cho các shipper để đưa hàng đến cho khách hàng và đồng thời sẽ gọi điện cho khách hàng để xác nhận thông tin.Trong lúc gọi nếu khách hàng không phản hồi, hoặc nhận thấy các thông tin bị sai lệch thì đơn hàng sẽ bị hủy ngay lập tức và hoàn lại tiền nếu khách hàng sử dụng phương thức thanh toán bằng thẻ.

1.7.2.3 Quy trình tra cứu cửa hàng (Khách hàng):

Quy trình được diễn ra khi khách hàng có nhu cầu tìm kiếm thức ăn.Được thực hiện sau khi khách hàng đã đăng nhập thành công vào ứng dụng của mình. Khách hàng sẽ nhập vào từ khóa món ăn cần kiếm sau đó hệ thống sẽ kiểm tra thông tin dữ liệu bên server.Nếu tìm thấy hệ thống sẽ hiển thị ra các danh sách cửa hàng liên quan đến món ăn của khách hàng đã nhập nếu không hệ thống sẽ thông báo là không thể tìm thấy dữ liệu.

1.7.2.4 Quy trình thay đổi thông tin (Khách hàng):

Quy trình được diễn ra sau khi khách hàng đã đăng nhập vào hệ thống thành công và có nhu cầu thay đổi thông tin của mình. Khách hàng sẽ vào trang tài khoản của mình và thay đổi các thông tin.

1.7.2.5 Quy trình đăng ký tài khoản (đối tác/ admin):

Quy trình thường diễn ra khi một trong 2 đối tượng là đối tác hoặc admin có nhu cầu muốn tạo tài khoản để truy cập vào trang quản lý.

Đối với việc tạo tài khoản của đối tác thì:

- Người dùng nhập thông tin.
- Chọn đối tượng mà tài khoản được tạo nhắm tới là đối tác.
- Hệ thống sẽ kiểm tra định dạng của thông tin.

- Người dùng nhấn đăng ký.
- Hệ thống kiểm tra thông tin rồi thông báo cho người dùng.
- Chuyển hướng về trang đăng nhập.

Đối với việc tạo tài khoản của admin thì:

- Người dùng nhập thông tin.
- Chọn đối tượng mà tài khoản được tạo nhằm tới là admin.
- Hệ thống kiểm tra định dạng của thông tin.
- Người dùng nhấn đăng ký.
- Hệ thống kiểm tra thông tin rồi thông báo cho người dùng.
- Người dùng cần được admin khác cấp mật khẩu.
- Khi đã có mật khẩu thì tiến hành đăng nhập.

1.7.2.6 Quy trình quản lý thông tin(đối tác/ admin):

Quy trình thường diễn ra khi một trong 2 đối tượng là đối tác hoặc admin có nhu cầu quản lý thông tin trong quyền hạn của mình (xem – xóa – sửa):

- Hệ thống sẽ hiển thị các danh mục mà người dùng có thể quản lý:
 - Đối tác:
 - Menu.
 - Hóa đơn.
 - Admin:
 - Nhân viên.
 - Cửa hàng.
 - Khách hàng.
- Người dùng chọn một trong những danh mục mà họ có quyền.
- Hệ thống sẽ lấy thông tin từ database và sẽ trực quan hóa thông tin lấy được vào bảng thông tin.
- Người dùng có thể thực hiện các thao tác trên bảng (thêm – xóa – sửa).
- Hệ thống sẽ thực hiện cập nhật thông tin vào database cũng như là bảng thông tin.

1.7.2.7 Quy trình thực hiện cập nhật thông tin (đối tác/ admin):

Quy trình thường diễn ra khi một trong 2 đối tượng là đối tác hoặc admin có nhu cầu muốn cập nhật thông tin của mình:

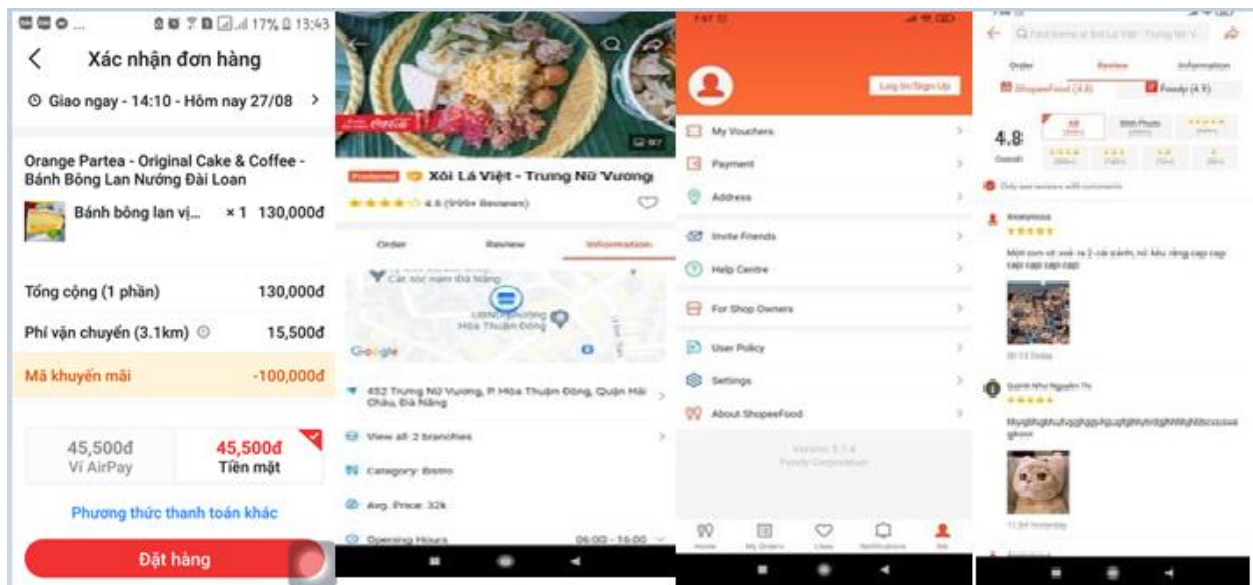
- Người dùng truy cập vào trang quản lý.

- Người dùng chọn danh mục thông tin cá nhân.
- Hệ thống tiến hành lấy thông tin dựa vào quyền người dùng hiện tại.
- Hệ thống hiển thị thông tin đã lấy vào một form.
- Người dùng nhập thông tin cần thay đổi.
- Hệ thống kiểm tra định dạng thông tin người dùng nhập vào.
- Người dùng nhấn cập nhật.
- Hệ thống kiểm tra thông tin rồi thông báo cho người dùng.
- Hiển thị thông tin được cập nhật.

1.7.3 Các biểu mẫu thu thập được :



Hình 1. 8 Biểu mẫu shoppeeFood



Hình 1. 9 Biểu mẫu shoppeeFood

1.8. TỔNG KẾT HỆ THỐNG :

Như vậy sau chương 1 tổng quan về hệ thống đã khảo sát nhóm em đã tìm hiểu được đầy đủ các yêu cầu cần thiết mà một hệ thống giao đặt hàng trực tuyến cần có, kết quả đạt được bao gồm : tổng quan rõ ràng về nguồn gốc của cửa hàng, các hệ thống liên quan , các hình ảnh và biểu mẫu thực tế để hỗ trợ cho việc phân tích và thiết kế một hệ thống giao, đặt hàng trực tuyến.

CHƯƠNG 2

PHÂN TÍCH HỆ THỐNG

2.1 GIỚI THIỆU :

Hệ thống chúng em được xây dựng gồm 2 phần :

❖ Phần ứng dụng cho khách hàng :

- Hệ thống bên phần khách hàng được xây dựng và phát triển trên nền tảng crossplatform(flutter) để phát triển cho cả 2 nền tảng phổ biến nhất hiện nay bao gồm IOS và Android.
- Hệ thống bao gồm có các chức năng như sau : đăng nhập, đăng xuất, quên mật khẩu, tra cứu, hiển thị thông tin(cửa hàng,đơn hàng,thức ăn),theo dõi các cửa hàng yêu thích, đặt đơn hàng , duyệt lịch sử đặt đơn hàng, xem giỏ hàng, thay đổi thông tin cá nhân, chức năng nhận thông báo, đánh giá bình luận cho cửa hàng.

❖ Phần ứng dụng cho quản trị và đối tác :

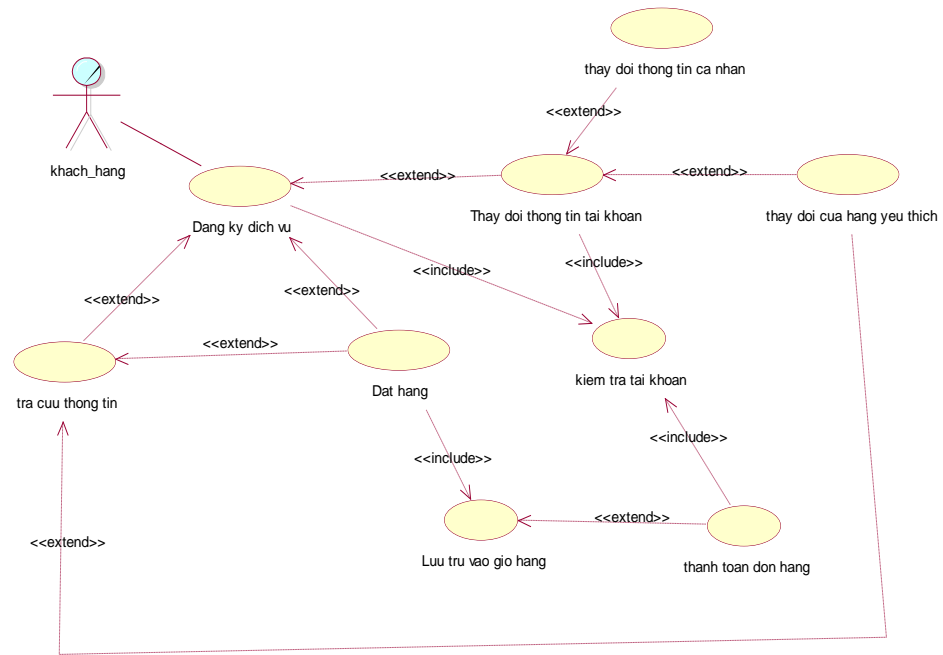
- Chức năng của đối tác bao gồm : Đăng ký tài khoản đối tác, Đăng nhập quyền đối tác, hiển thị thống kê tổng quan của cửa hàng, quản lý và thống kê menu và hóa đơn.
- Chức năng của admin bao gồm : Đăng ký / đăng nhập tài khoản admin,hiển thị thống kê tình hình tổng quan của ứng dụng, quản lý nhân viên (CRUD), quản lý khách hàng(CRUD), quản lý cửa hàng (CRUD).

Ngoài ra cả hai phần của chúng em đều sử dụng firebase notification để có thể nhận/gửi thông báo đặt đơn hàng một cách realtime. Dữ liệu của hệ thống được lưu trữ trên mongodb atlas và thông qua framework của nodejs để dựng các api cho việc giao tiếp giữa 2 phần.

2.2 MÔ HÌNH HÓA CHỨC NĂNG

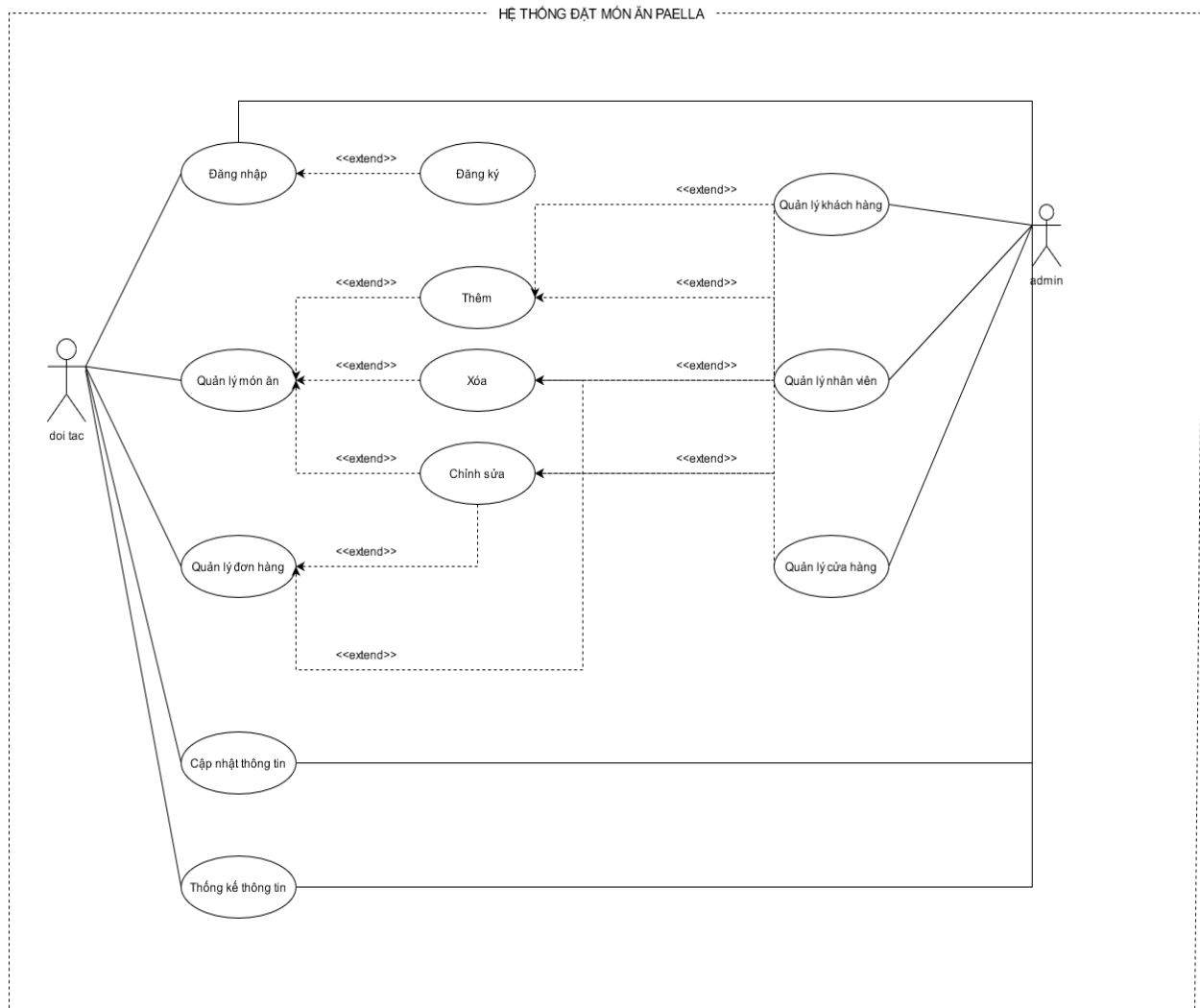
2.2.1 Sơ đồ Use Case hệ thống

a. Phần ứng dụng khách hàng :



Hình 2. 1 Sơ đồ use-case phần ứng dụng khách hàng

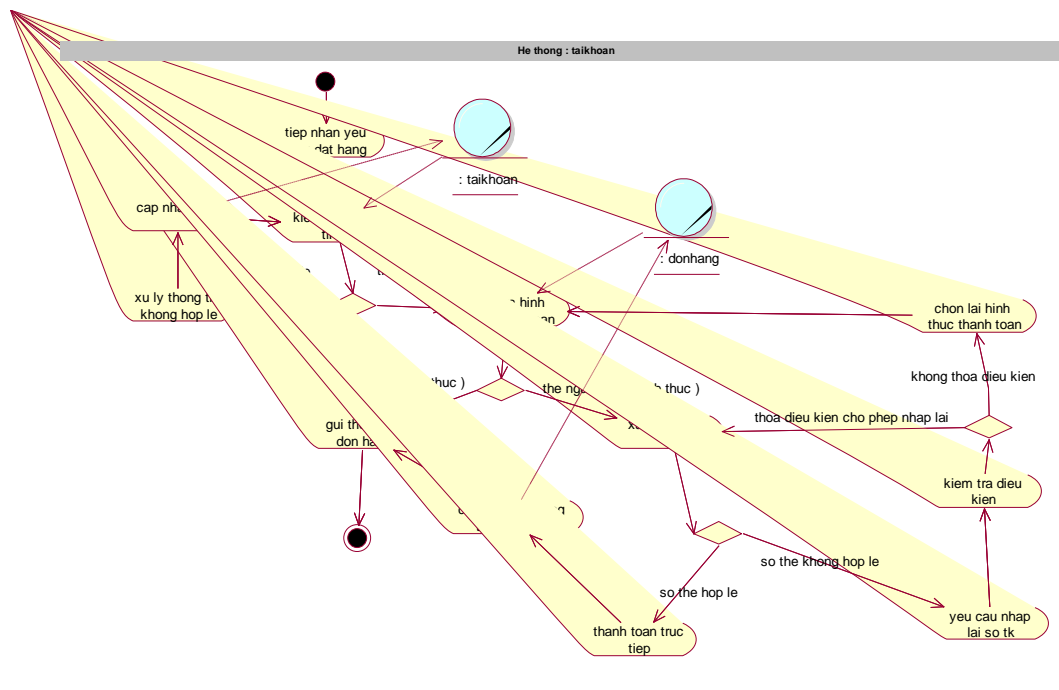
b. Phần đối tác và admin :



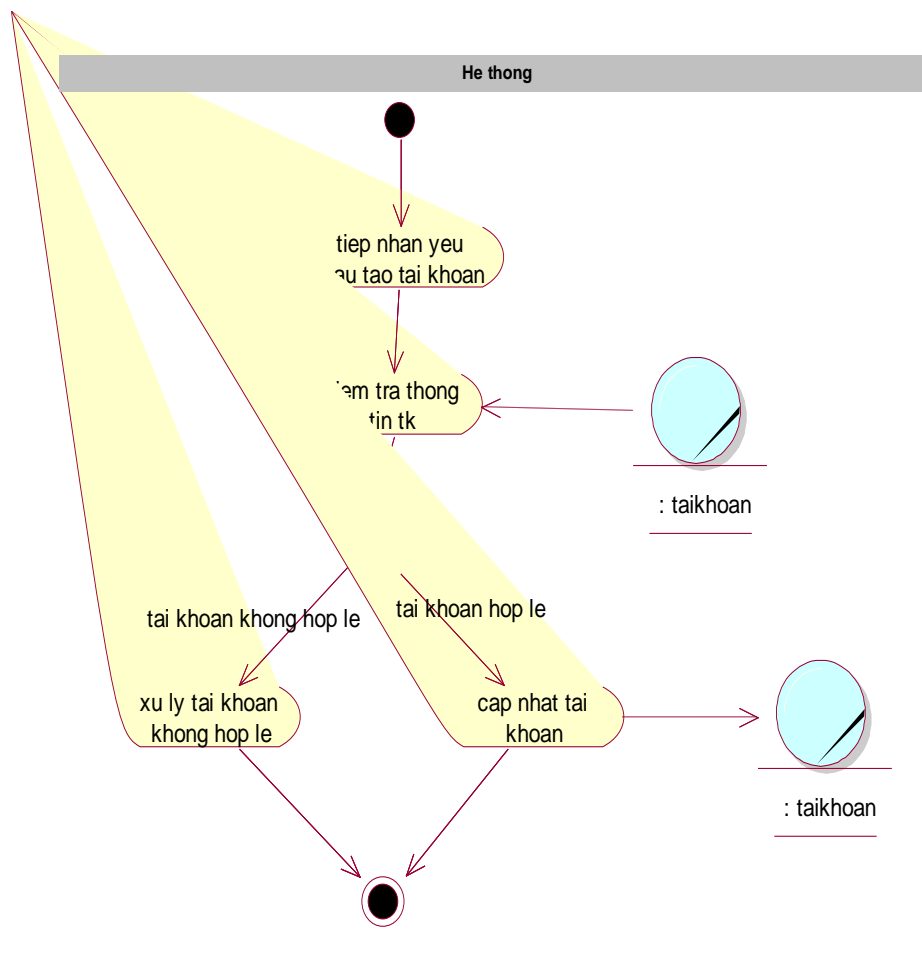
Hình 2. 2 Mô hình hệ thống của đối tác / admin.

2.2.2 Sơ đồ Activity :

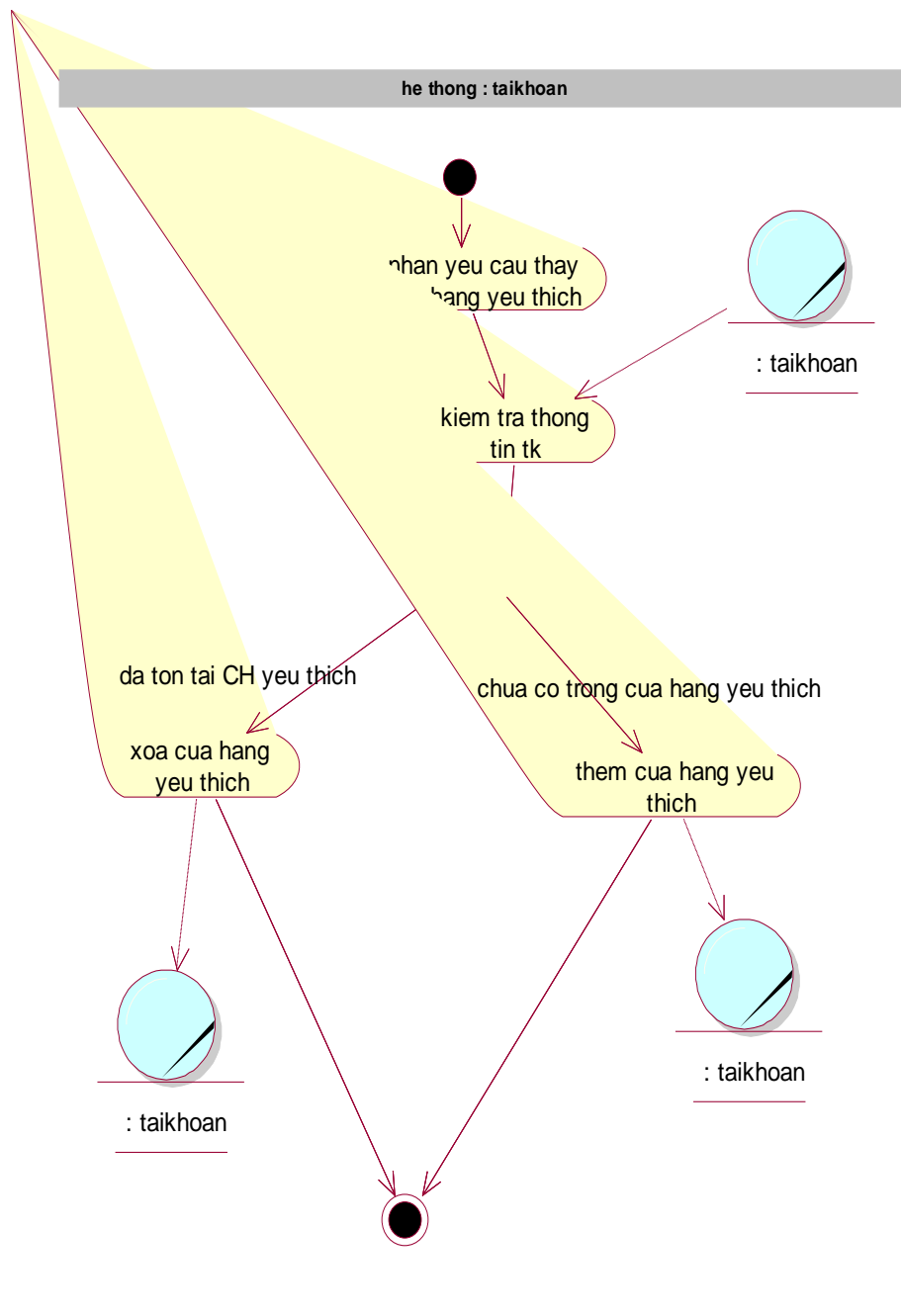
a. Phần ứng dụng khách hàng :



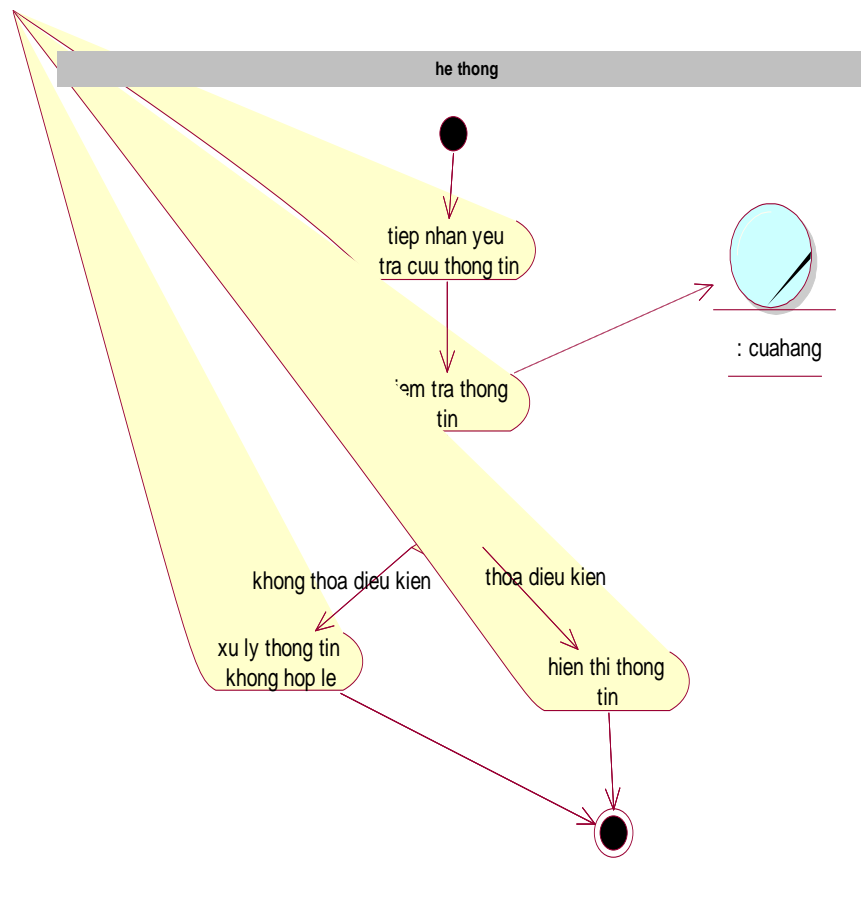
Hình 2. 3 Sơ đồ hoạt động đặt hàng



Hình 2. 4 Sơ đồ hoạt động cập mật khẩu

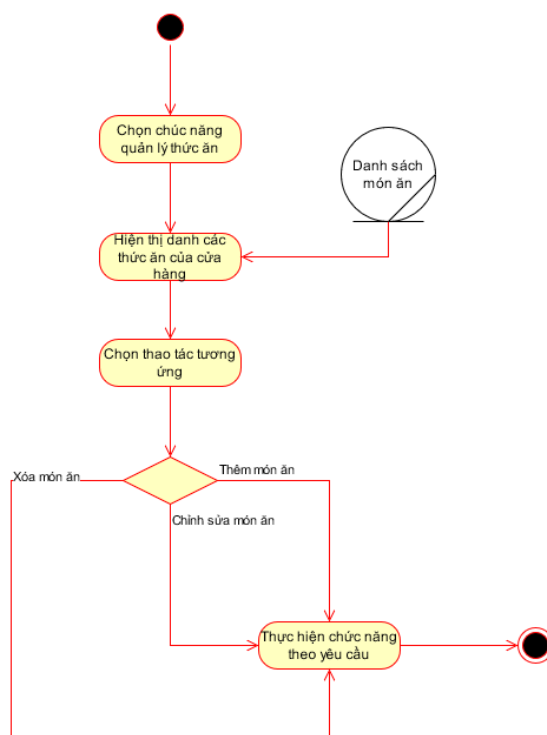


Hình 2. 5 Sơ đồ hoạt động cập nhật cửa hàng yêu thích

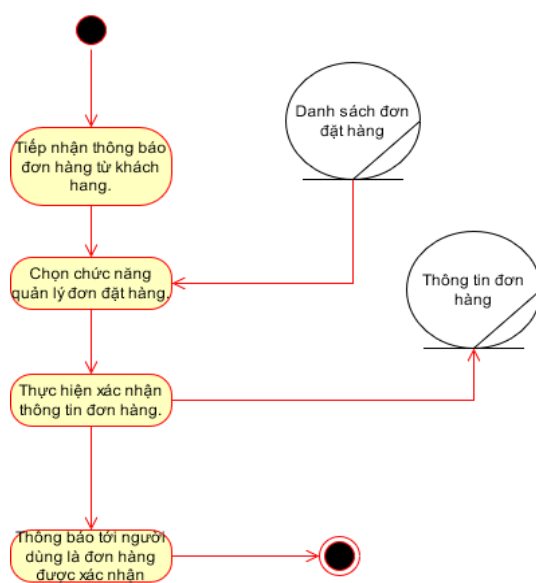


Hình 2. 6 Sơ đồ hoạt động tra cứu thông tin

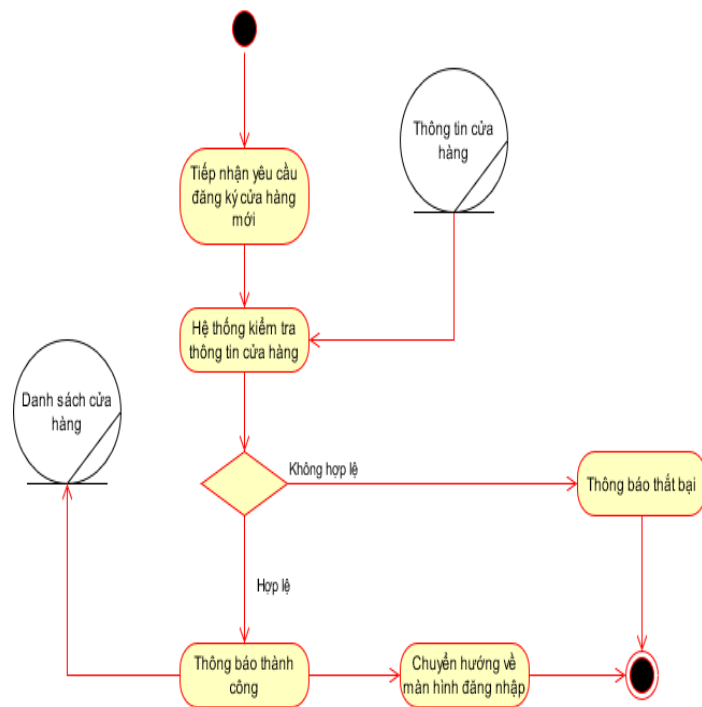
b. Phần quản trị / đối tác :



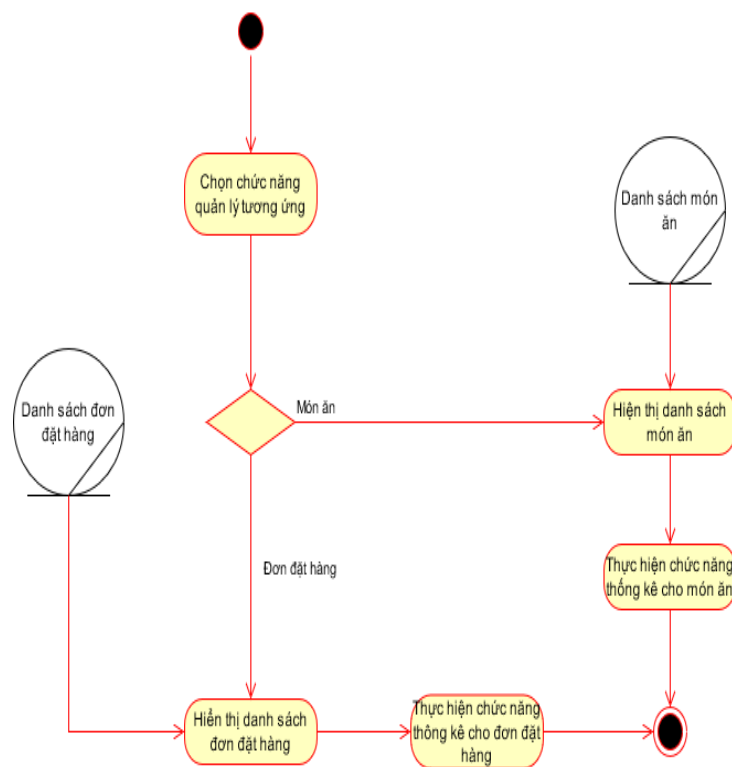
Hình 2. 7 Sơ đồ hoạt động quản lý menu



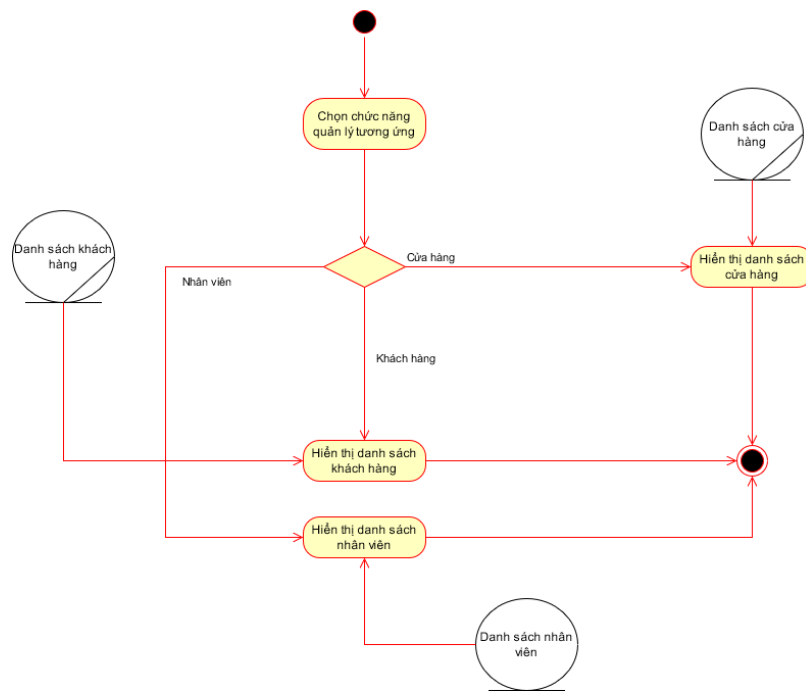
Hình 2. 8 Sơ đồ hoạt động cập nhật thông tin đặt hàng



Hình 2. 9 Sơ đồ hoạt động đăng ký đối tác



Hình 2. 10 Sơ đồ hoạt động quản lý của đối tác



Hình 2. 11 Sơ đồ hoạt động quản lý của admin

2.2.3 Đặc tả use case hệ thống

a. Phần ứng dụng khách hàng :

Tên use case	Dạng ký dịch vụ
Tóm tắt	Khi bắt đầu sử dụng, hệ thống sẽ yêu cầu đăng ký tài khoản để sử dụng, nếu đã có tài khoản thì có thể đăng nhập vào hệ thống sử dụng các dịch vụ.
Tác nhân	Khách hàng
Use case liên quan	Tra cứu thông tin, đặt hàng, kiểm tra tài khoản, thay đổi thông tin tài khoản
Dòng sự kiện chính	<ol style="list-style-type: none"> Hệ thống hiển thị 3 chức năng bao gồm đăng nhập, đăng ký và quên mật khẩu. Người dùng nhấp chọn : <ol style="list-style-type: none"> Đăng nhập Đăng ký Quên mật khẩu <ol style="list-style-type: none"> Hệ thống yêu cầu nhập email và mật khẩu.

	<ul style="list-style-type: none"> b. Hệ thống yêu cầu cung cấp các thông tin (sdt,email,mật khẩu). c. Hệ thống yêu cầu nhập email <p>4. a. Khách hàng chọn “Login” b. Khách hàng chọn “Create” c. Khách hàng chọn “Send”</p> <p>5. a.b.c. Hệ thống thực hiện UC chức năng kiểm tra tài khoản</p> <p>6. a. Hệ thống cho phép vào home page b. Hệ thống thực hiện tạo tài khoản c. Khách hàng nhập mã xác nhận mà hệ thống đã gửi</p> <p>7. b. Hệ thống cho phép vào home page c. Khách hàng nhập lại mật khẩu mới</p> <p>8. c. Hệ thống thực hiện việc cập nhập lại mật khẩu của khách hàng.</p>
Dòng sự kiện phụ	<p>Tại bước 3 : nếu nhập sai yêu cầu định dạng các thông tin yêu cầu thì hệ thống sẽ yêu cầu nhập lại.</p> <p>Tại bước 4:</p> <ul style="list-style-type: none"> a. Nếu nhập sai mật khẩu, hệ thống sẽ thông báo yêu cầu sử dụng chức năng quên mật khẩu tại bước 2c b. Nếu email đã tồn tại, hệ thống sẽ hiển thị thông báo , yêu cầu đổi lại email hoặc sử dụng chức năng 2c. <p>Tại bước 6 :</p> <ul style="list-style-type: none"> c. Nếu khách hàng nhập sai mã hệ thống sẽ bắt yêu cầu nhập lại .
Điều kiện tiên quyết	Khách hàng muốn sử dụng dịch vụ
Hậu điều kiện	Hình thành 1 tài khoản cho khách hàng hoặc cho phép khách hàng vào trang chủ.

Bảng 2. 1 Đặc tả UC đăng ký dịch vụ

Tên use case	Tra cứu thông tin
Tóm tắt	Trong quá trình sử dụng, khách hàng có thể tìm kiếm các thông tin cửa hàng
Tác nhân	Khách hàng
Use case liên quan	Đăng ký sử dụng dịch vụ, đặt hàng, kiểm tra tại quầy, thay đổi cửa hàng yêu thích
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống yêu cầu khách hàng nhập vào thức ăn muốn tìm 2. Khách hàng điền tên thức ăn . 3. Khách hàng chọn icon “tìm kiếm” 4. Hệ thống hiển thị lên các cửa hàng có chứa thức ăn đó.
Dòng sự kiện phụ	<p>Tại bước 3: nếu người dùng không nhập thì hệ thống sẽ yêu cầu không được để trống.</p> <p>Tại bước 4 : nếu hệ thống không tìm thấy cửa hàng nào sẽ thông báo thức ăn của khách hàng không có bán</p>
Điều kiện tiên quyết	Khách hàng phải đăng ký/đăng nhập và muốn tìm kiếm thức ăn.
Hậu điều kiện	Hiển thị các cửa hàng có liên quan hoặc thông báo không tìm thấy cửa hàng nào.

Bảng 2. 2 Đặc tả UC tra cứu thông tin

Tên use case	Đặt hàng
Tóm tắt	Trong quá trình sử dụng, khách hàng muốn gọi món từ cửa hàng
Tác nhân	Khách hàng

Use case liên quan	Dang ky su dung dich vu,tra cuu thong tin,Luu vao gio hang
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống hiển thị các danh sách món ăn từ cửa hàng đã chọn 2. Khách hàng chọn các món ăn bằng cách chọn dấu cộng hoặc trừ tùy theo số lượng khách hàng muốn gọi. 3. Hệ thống thực hiện UC lưu trữ vào giỏ hàng 4. Hệ thống ghi nhận thông tin giỏ hàng
Dòng sự kiện phụ	Tại bước 4 nếu khách hàng không chọn chức năng thanh toán thì sau khi thoát ứng dụng hoặc đăng xuất giỏ hàng sẽ mất các món đã gọi.
Điều kiện tiên quyết	Khách hàng phải đăng ký dịch vụ và chọn món ăn.
Hậu điều kiện	Lưu trữ được các món ăn mà khách hàng đã chọn.

Bảng 2. 3 Đặc tả UC dat hang

Tên use case	Thanh toan don hang
Tóm tắt	Khi khách hàng đã có hàng trong giỏ hàng thì khách hàng có thể dùng chức năng thanh toán để hoàn tất việc đặt hàng
Tác nhân	Khách hàng
Use case liên quan	Luu tru gio hang vao gio,kiem tra tai khoan
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống hiển thị ra các danh sách món ăn và tính tổng tiền của các món ăn. 2. Khách hàng cung cấp các thông tin cần thiết 3. Khách hàng xem rồi , chọn chức năng thanh toán 4. Hệ thống thực hiện UC thanh toan don hang 5. Hệ thống thực hiện gửi đơn hàng với thông tin khách hàng tới cửa hàng .

Dòng sự kiện phụ	<p>Tại bước 3 nếu khách hàng không có cung cấp địa chỉ thì hệ thống sẽ báo lỗi quay lại bước 2</p> <p>Tại bước 2 khách hàng có thể thay đổi địa chỉ, áp voucher cho đơn hàng</p>
Điều kiện tiên quyết	Khách hàng phải có giỏ hàng trong giỏ hàng.
Hậu điều kiện	Hoàn tất được việc gửi thành công giỏ hàng tới cửa hàng.

Bảng 2. 4 Đặc tả UC thanh toán đơn hàng

Tên use case	Thay đổi thông tin cá nhân
Tóm tắt	Khi khách hàng muốn cập nhật thông tin cá nhân của mình
Tác nhân	Khách hàng
Use case liên quan	thay đổi thông tin tài khoản
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng chọn “Me”. 2. Hệ thống hiển thị các thông tin khách hàng 3. Khách hàng thay đổi các fields cần đổi 4. Khách hàng chọn “Cập nhật” 5. Hệ thống thực hiện UC thay đổi thông tin cá nhân 6. Hệ thống cập nhật lại thông tin khách hàng
Dòng sự kiện phụ	Tại bước 4 nếu khách hàng cung cấp các fields không đúng với các yêu cầu định dạng thì hệ thống thông báo lỗi
Điều kiện tiên quyết	Khách hàng phải đăng nhập sử dụng dịch vụ
Hậu điều kiện	Hoàn tất được việc cập nhật lại thông tin cá nhân của khách hàng.

Bảng 2. 5 Đặc tả UC thay đổi thông tin cá nhân

Tên use case	Thay doi cua hang yeu thich
Tóm tắt	Khi khách hàng muốn theo dõi các cửa hàng yêu thích của mình
Tác nhân	Khách hàng
Use case liên quan	thay doi thong tin tai khoan
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. a. Khách hàng chọn “Favorite”. b. Khách hàng chọn cửa hàng để like 2. a. Hệ thống hiển thị các danh sách cửa hàng đã like. b. Khách hàng chọn like cửa hàng. 3. Khách hàng chọn cửa hàng để thay đổi trạng thái yêu thích -> bỏ thích, chưa thích -> thích 4. Hệ thống thực hiện UC thay doi cua hang yeu thich 5. Hệ thống cập nhật lại thông tin các cửa hàng yêu thích
Dòng sự kiện phụ	Tại bước 1 khách hàng phải có ít nhất 1 cửa hàng đã yêu thích.
Điều kiện tiên quyết	Khách hàng phải đăng nhập sử dụng dịch vụ
Hậu điều kiện	Hoàn tất được việc cập nhật lại thông tin yêu thích cửa hàng của khách hàng.

Bảng 2. 6 Thay doi cua hang yeu thich

b. Phần đối tác / admin :

Tên use case	Đăng ký tài khoản
Tóm tắt	Khi muốn truy cập vào trang quản lý cửa hàng / admin thì người dùng phải có tài khoản để đăng nhập vào trang
Tác nhân	Cửa hàng / Admin
Use case liên quan	Đăng nhập

Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng thực hiện chức năng đăng ký. 2. Hệ thống yêu cầu người dùng nhập thông tin đăng ký. 3. Người dùng xác nhận thông tin. 4. Người dùng bấm đăng ký. 5. Hệ thống kiểm tra thông tin. 6. Hệ thống thông báo là tài khoản được tạo thành công.
Dòng sự kiện phụ	<p>Tại bước 2 : nếu nhập sai yêu cầu định dạng các thông tin yêu cầu thì hệ thống sẽ yêu cầu nhập lại.</p> <p>Tại bước 5:</p> <ol style="list-style-type: none"> a. Nếu như thông tin email đã có người đăng ký thì hệ thống thông báo bại cho người dùng. b. Nếu thông tin hợp lệ thì chuyển sang bước thứ 6. c. Nếu người dùng chọn đăng ký là admin thì phải có người khác cấp mật khẩu cho.
Điều kiện tiên quyết	Đối tác / admin muốn tạo tài khoản.
Hậu điều kiện	Đối tác / admin có tài khoản để truy cập vào trang quản lý.

Bảng 2. 7 Đặc tả UC đăng ký tài khoản (admin/đối tác)

Tên use case	Cập nhật thông tin
Tóm tắt	Khi đối tác / admin muốn cập nhật thông tin của họ.
Tác nhân	Cửa hàng / admin
Use case liên quan	Đăng ký
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng chọn chức năng cập nhật thông tin. 2. Hệ thống hiển thị thông tin hiện tại của người dùng. 3. Người dùng có thể nhập thông tin cần thay đổi.

	<p>4. Người dùng bấm cập nhật thông tin.</p> <p>5. Hệ thống kiểm tra thông tin.</p> <p>6. Hệ thống thông báo là cập nhật thành công.</p>
Dòng sự kiện phụ	<p>Tại bước 3: Nếu người dùng nhập sai dạng thông tin thì hệ thống sẽ thông báo cho người dùng.</p> <p>Tại bước 4:</p> <ul style="list-style-type: none"> a. Nếu người dùng nhập thông tin email đã tồn tại thì hệ thống sẽ thông báo thất bại cho người dùng. b. Nếu thông tin hợp lệ thì sẽ chuyển sang bước 6.
Điều kiện tiên quyết	Đối tác / admin muốn cập nhật thông tin của họ.
Hậu điều kiện	Thông tin của đối tác / admin được cập nhật trong hệ thống.

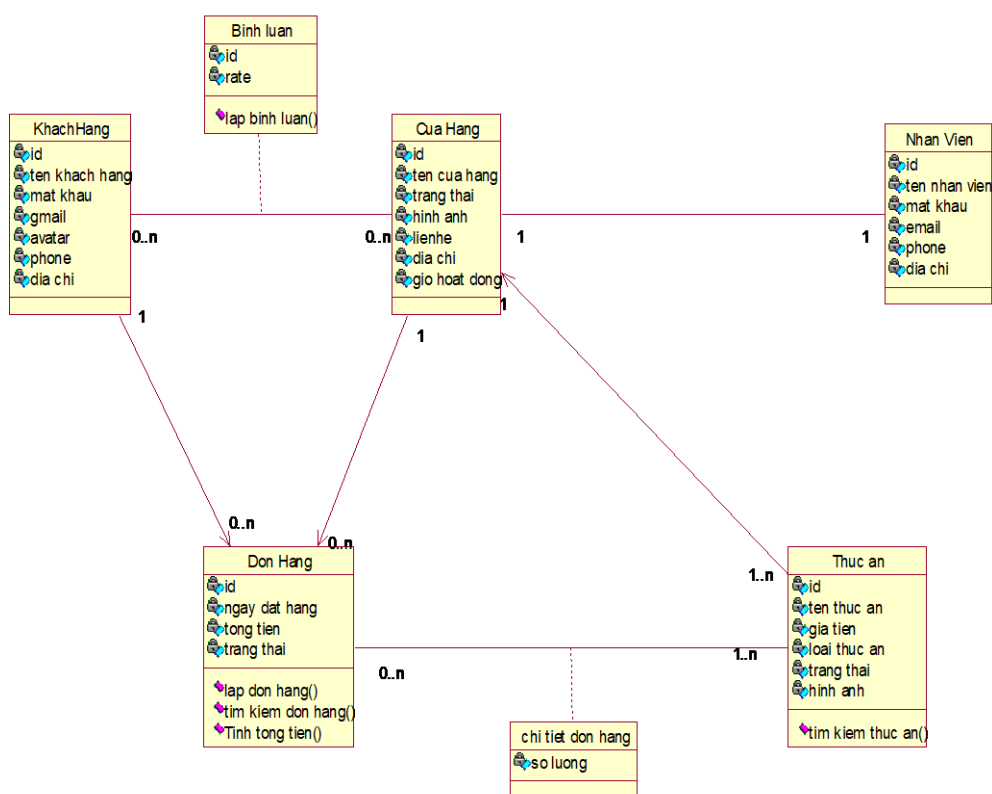
Bảng 2. 8 Đặc tả UC cập nhật thông tin (admin/đối tác)

Tên use case	Quản lý
Tóm tắt	Khi đối tác / admin muốn quản lý thông tin trong quyền hạn của họ.
Tác nhân	Cửa hàng / admin
Use case liên quan	
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống kiểm tra quyền của người dùng. 2. Hệ thống hiển thị các quyền quản lý tương ứng. 3. Người dùng chọn một trong các chức năng quản lý. 4. Hệ thống hiển thị thông tin tương ứng. 5. Hệ thống thực hiện thống kê thông tin tương ứng.
Dòng sự kiện phụ	Tại bước 2: Hệ thống dựa vào quyền của người dùng để hiển thị.

	<p>Tại bước 4:</p> <p>2.1 Hệ thống sẽ dựa vào quyền của người dùng để hiển thị thông tin theo quyền hạn cho phép.</p> <p>2.2 Nếu hệ thống phát hiện người dùng chưa đăng nhập thì sẽ không cho phép điều hướng vào trang quản lý.</p>
Điều kiện tiên quyết	Đối tác / admin muốn quản lý thông tin và xem thống kê.
Hậu điều kiện	

Bảng 2. 9 Đặc tả UC quan ly (admin/doi tac)

2.3 SƠ ĐỒ LỚP MỨC PHÂN TÍCH



Hình 2. 12 Sơ đồ mức phân tích

2.4 TỔNG KẾT CHƯƠNG

Như vậy sau chương 2 nhóm chúng em đã hoàn tất việc xác định được các nghiệp vụ cần thiết và chuyển đổi thành các mô hình tương ứng. Trong các mô hình

nhóm chúng em đã có các sơ đồ UC hệ thống và các bảng đặc tả chi tiết của các sơ đồ UC hệ thống. Từ các mô hình trên nhóm chúng em đã hoàn thành việc thiết kế một sơ đồ lớp mức phân tích hoàn chỉnh.

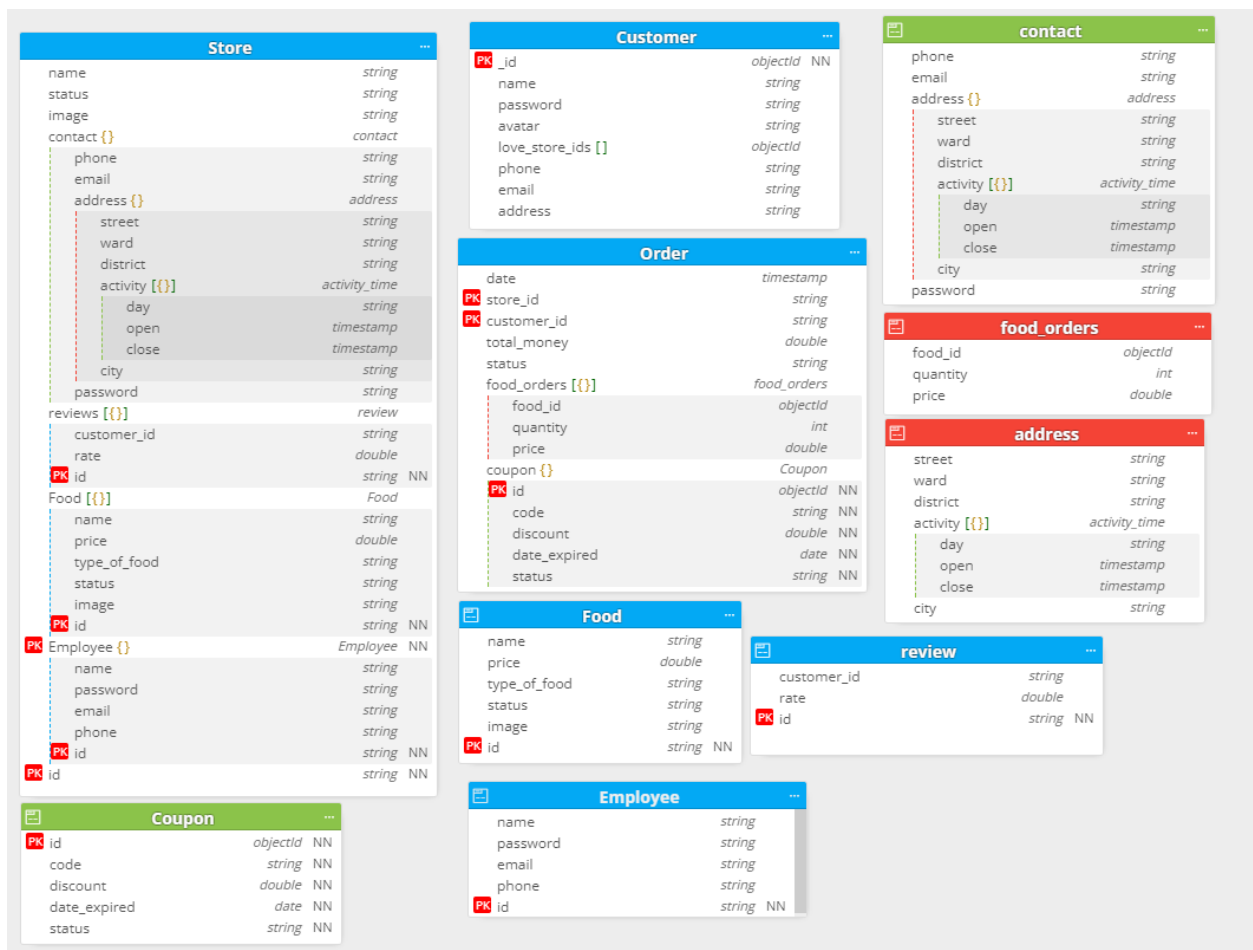
CHƯƠNG 3

THIẾT KẾ HỆ THỐNG

3.1 GIỚI THIỆU

Trong giai đoạn này, nhóm chúng em sẽ thiết kế cơ sở dữ liệu phi quan hệ (NoSQL). Nhờ việc sử dụng csdl mongodb thì việc lưu trữ các thông tin với cấu trúc phức tạp đã trở nên dễ dàng hơn nhờ đó mà việc lấy dữ liệu từ hệ thống cũng đã trở nên thuận lợi hơn rất nhiều.

3.2 THIẾT KẾ CSDL



Hình 3. 1 Lược đồ csdl thiết kế bằng moon modeler

Giải thích :

- Trong lược đồ có 3 collection bao gồm : Store, Customer và Order. Trong đó có các document được nhúng bên trong. Các document bao gồm (7) : Coupon, Food, Employee, review, address, food_order và contact.

Bảng mô tả cho từng collection và document tương ứng :

- **Collection Store :**

NAME	DATATYPE	ARR	ID	NN
name	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
status	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
image	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
contact	contact (document)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
reviews	review (document)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Food	Food (document)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Employee	Employee (document)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
id	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Bảng 3. 1 Mô tả Collection Store

- **Collection Customer :**

NAME	DATATYPE	ARR	ID	NN
_id	objectId	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
name	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
password	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
avatar	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
love_store_ids	objectId	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
phone	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
email	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
address	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Bảng 3. 2 Mô tả Collection Customer

- **Collection Order :**

NAME	DATATYPE	ARR	ID	NN
date	timestamp	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
store_id	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
customer_id	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
total_money	double	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
status	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
food_orders	food_orders (document)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
coupon	Coupon (document)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Bảng 3. 3 Mô tả Collection Order

- **Document Contact :**

NAME	DATATYPE	ARR	ID	NN
phone	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
email	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
address	address (document)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
password	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Bảng 3. 4 Mô tả Document Contact

- **Document Food :**

NAME	DATATYPE	ARR	ID	NN
name	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
price	double	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
type_of_food	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
status	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
image	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
id	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Bảng 3. 5 Mô tả Document Food

- **Document Coupon :**

NAME	DATATYPE	ARR	ID	NN
id	objectId	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
code	string	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
discount	double	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
date_expired	date	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
status	string	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Bảng 3. 6 Mô tả Document Coupon

- **Document Employee :**

NAME	DATATYPE	ARR	ID	NN
name	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
password	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
email	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
phone	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
id	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Bảng 3. 7 Mô tả Document Employee

- **Document Address :**

NAME	DATATYPE	ARR	ID	NN
street	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ward	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
district	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
activity	activity_time (document)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
city	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

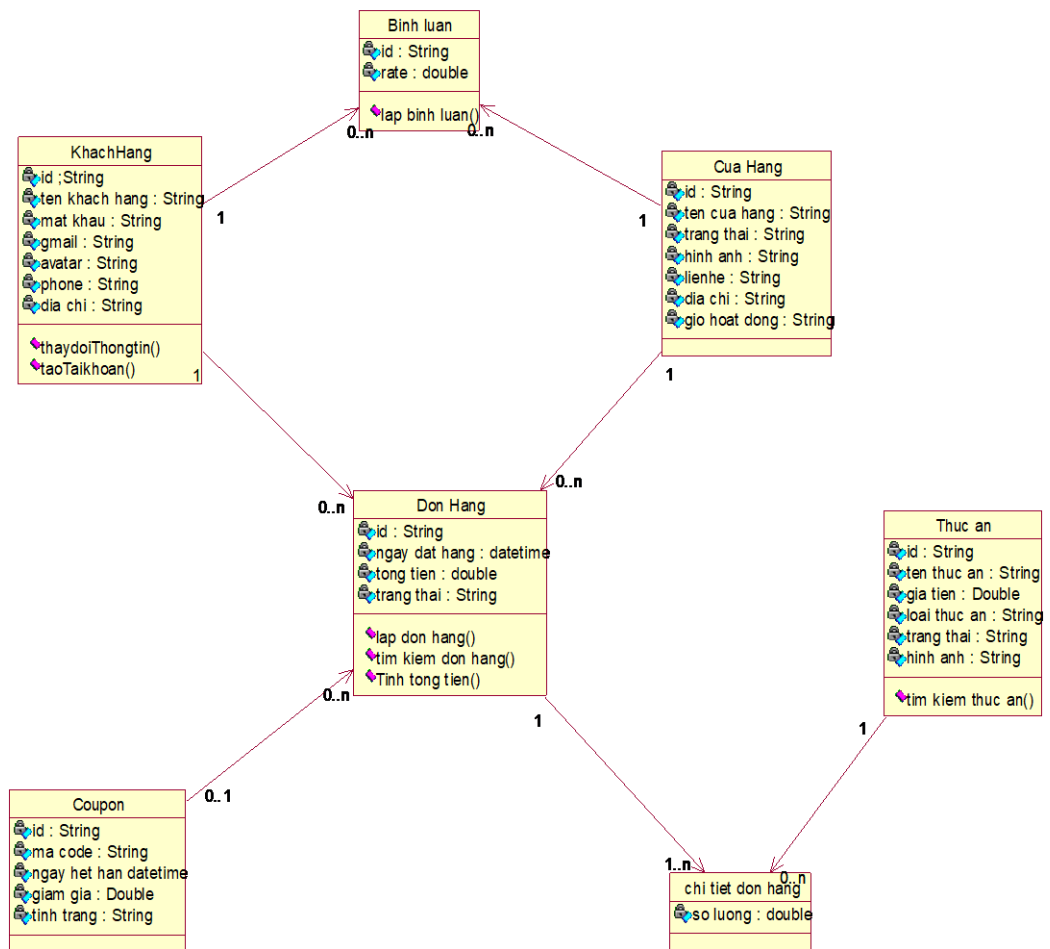
Bảng 3. 8 Mô tả Document Address

- **Document Review :**

NAME	DATATYPE	ARR	ID	NN
customer_id	string	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
rate	double	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
id	string	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Bảng 3. 9 Mô tả Document review

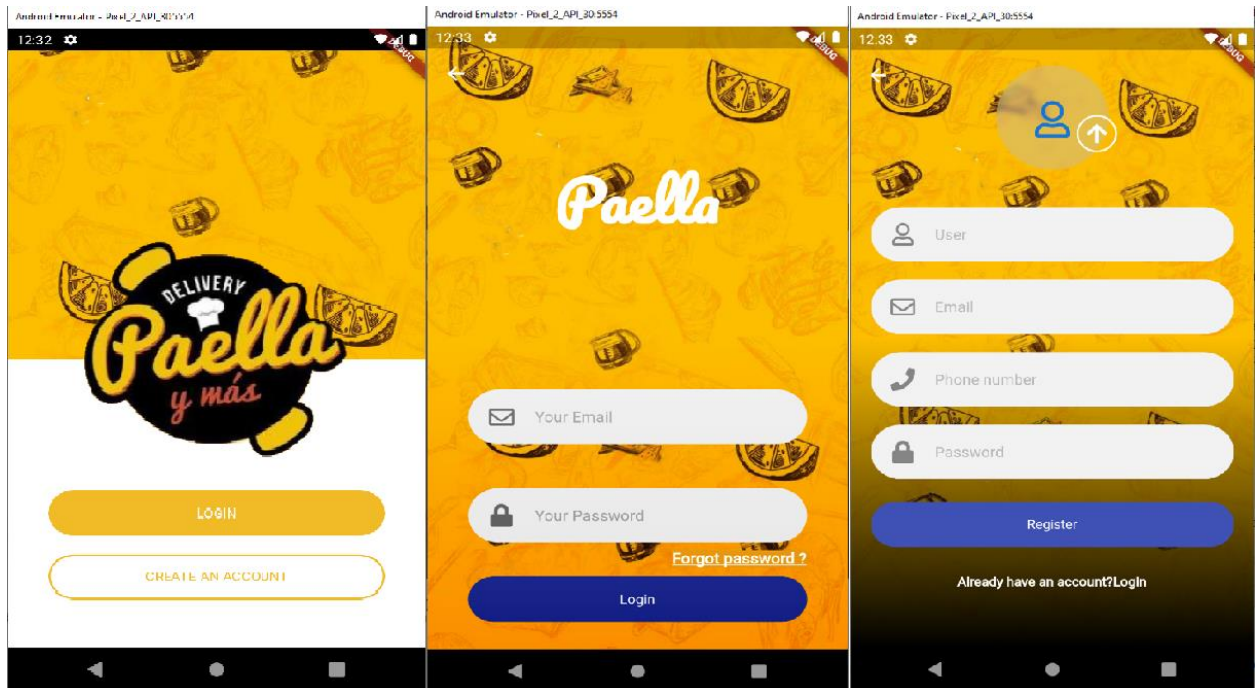
3.3 SƠ ĐỒ LỚP MỨC THIẾT KẾ



Hình 3. 2 Sơ đồ lớp mức thiết kế

3.4 THIẾT KẾ GIAO DIỆN HỆ THỐNG

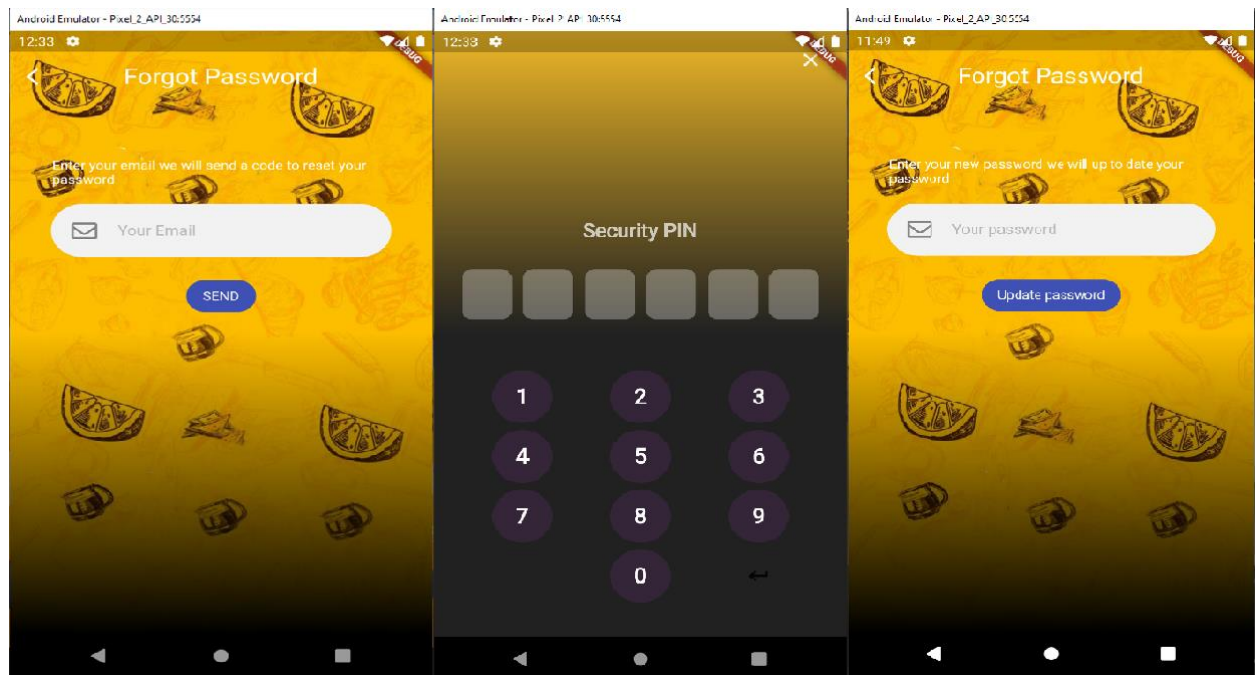
a. Phần ứng dụng khách hàng:



Hình 3. 3 Giao diện đăng nhập/đăng ký

Giao diện đăng nhập : yêu cầu khách hàng điền thông tin email và mật khẩu để vào trang chủ sau khi đăng xuất khỏi ứng dụng hoặc lần đầu đăng nhập vào ứng dụng

Giao diện đăng ký : Xuất hiện khi người dùng nhấn Create an account , yêu cầu khách hàng cung cấp các thông tin như user,email,phone và mật khẩu.

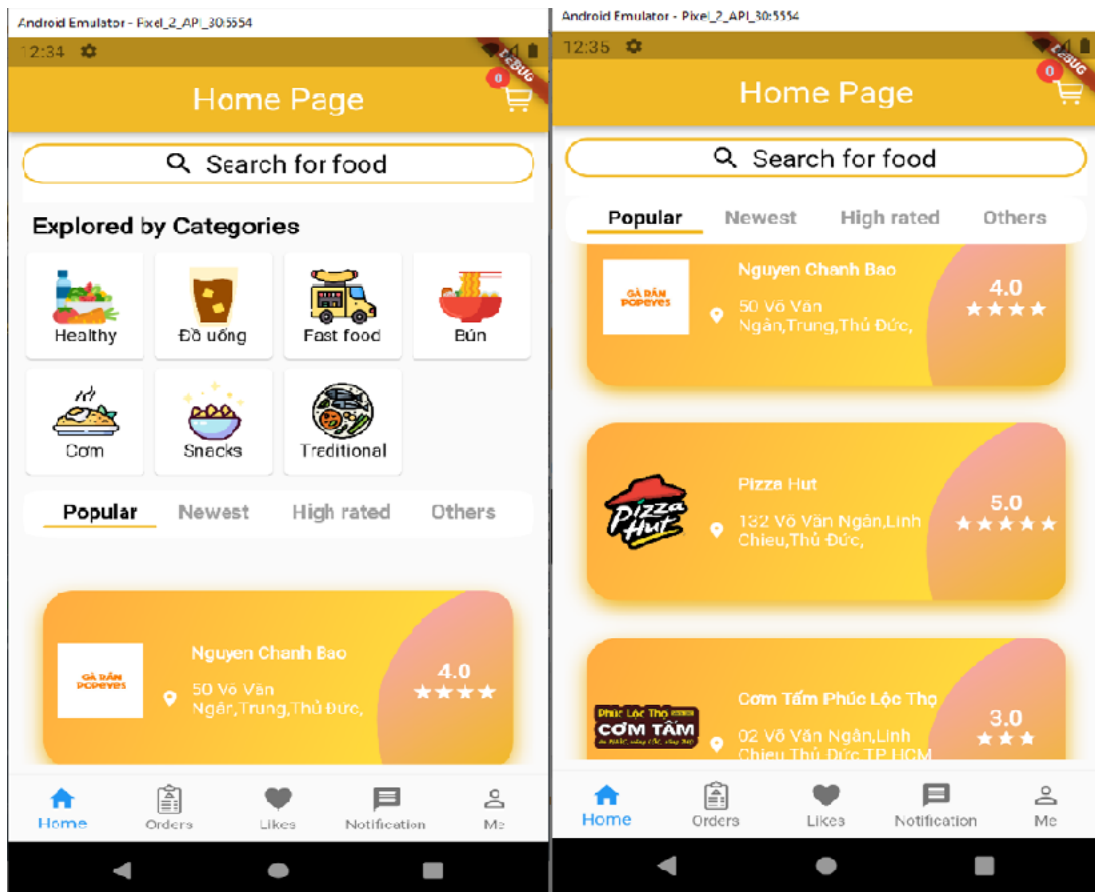


Hình 3. 4 Giao diện cung cấp lại mật khẩu

Giao diện quên mật khẩu : yêu cầu người dùng cung cấp email và nhấn gửi. Lúc này hệ thống sẽ gửi mã 6 chữ số bên gmail của người dùng.

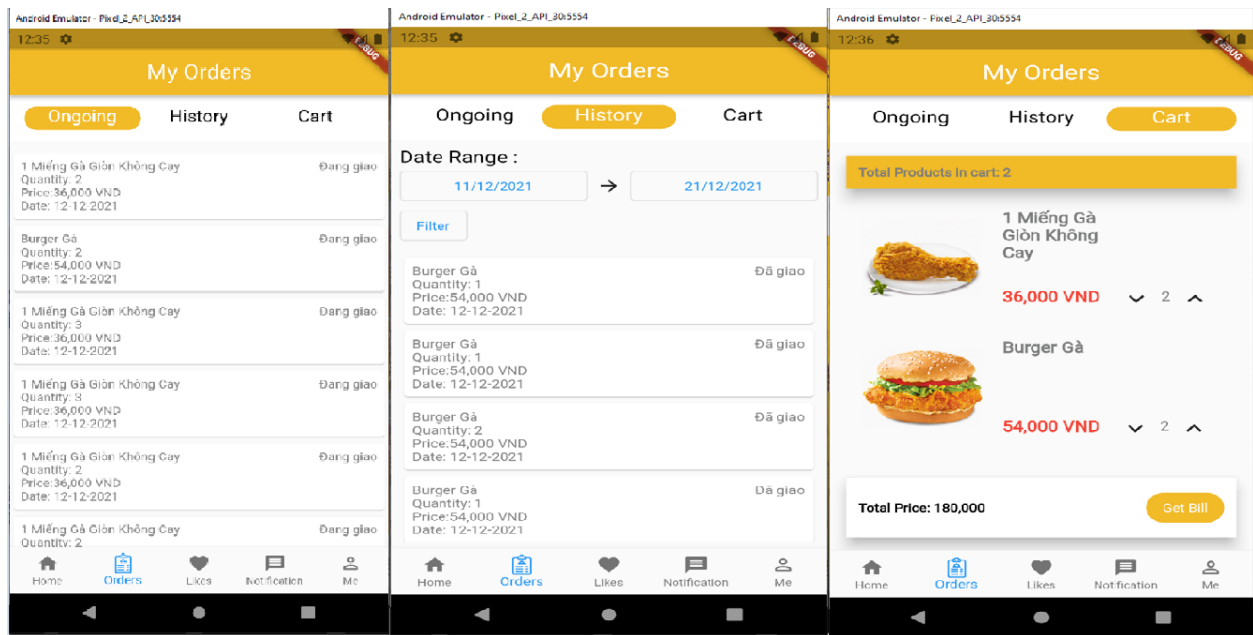
Giao diện nhập mã OTP: khi khách hàng nhận được mã thì hệ thống sẽ yêu cầu nhập 6 mã để xác minh

Giao diện cập nhật mật khẩu : Sau khi đã nhập 6 mã thành công , trang cập nhật mật khẩu sẽ hiện lên và khách hàng có thể thay đổi mật khẩu của mình



Hình 3. 5 Giao diện trang chủ

Giao diện trang chủ : Sau khi khách hàng đã đăng nhập thành công thì hệ thống sẽ cho người dùng vào trang chủ. Ở đây khách hàng có thể thấy được các thẻ lọc theo loại thức ăn, kéo xuống phía dưới là danh sách các cửa hàng được sắp xếp theo từng loại như Popular, Newest, High rated, Others. Và phía dưới cùng là các tab.

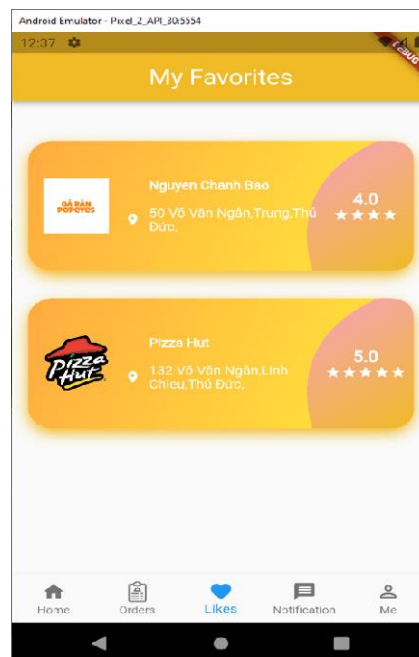


Hình 3. 6 Giao diện quản lý đơn hàng

Giao diện Ongoing: hiển thị các đơn hàng đang được giao tới cho khách hàng.

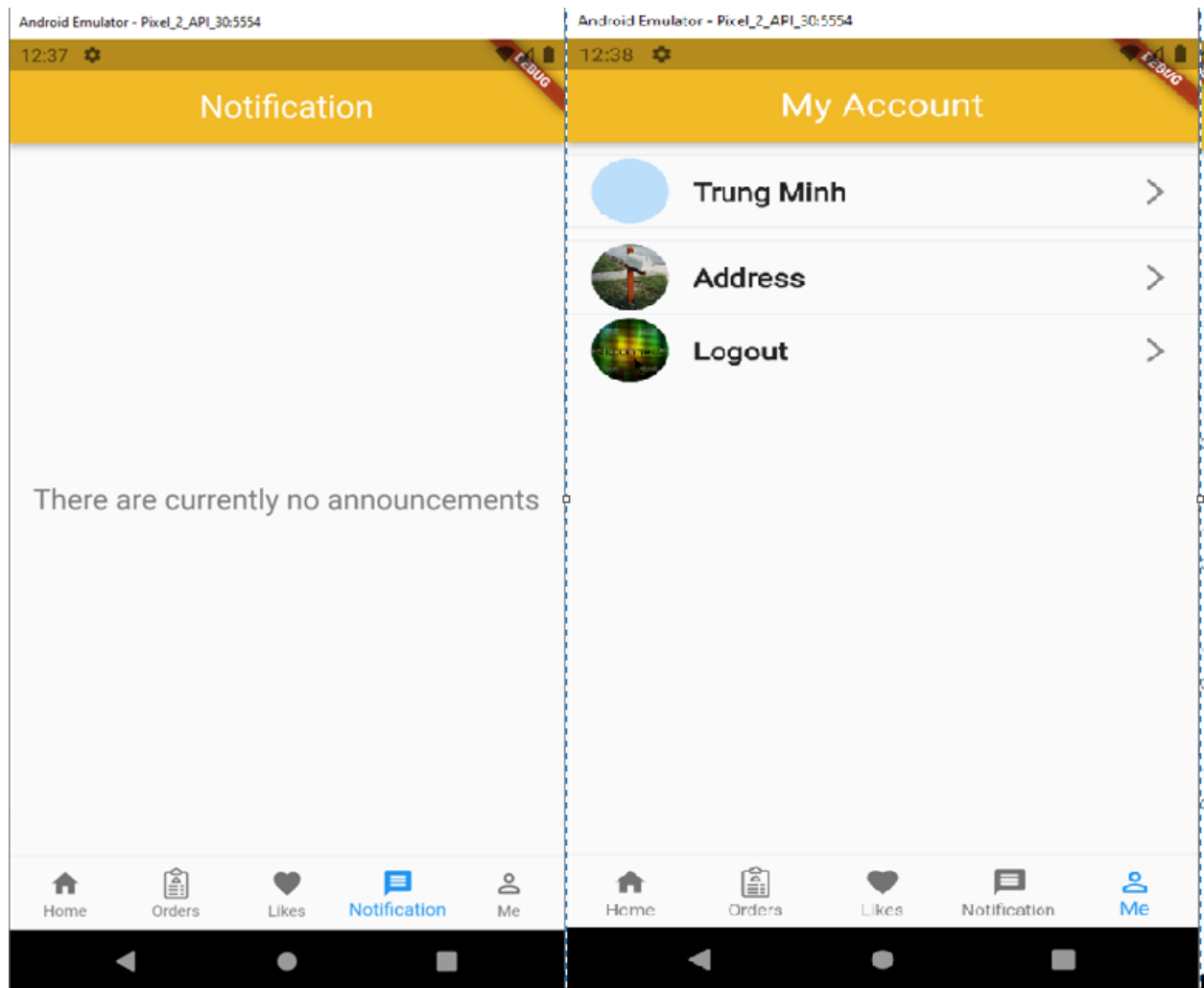
Giao diện History: cho phép người dùng hiển thị các đơn hàng đã mua từ ngày đến ngày.

Giao diện Cart : hiển thị các món ăn mà khách hàng đã đặt và cho phép khách hàng in hóa đơn.



Hình 3. 7 Giao diện theo dõi cửa hàng yêu thích

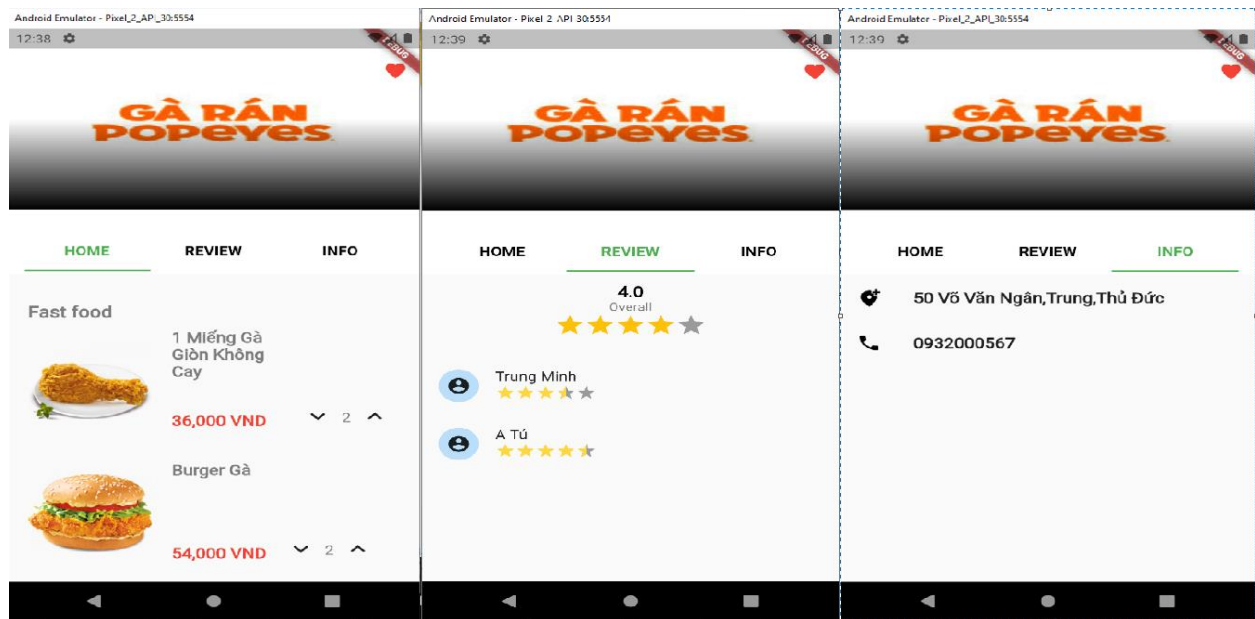
Giao diện cửa hàng yêu thích : cho phép người dùng theo dõi các cửa hàng đã like



Hình 3. 8 Giao diện thông báo và tài khoản

Giao diện thông báo : Hiển thị thông báo đơn hàng đã tới hay chưa

Giao diện tài khoản : cho phép khách hàng xem lại thông tin của mình hoặc thay đổi cập nhật.



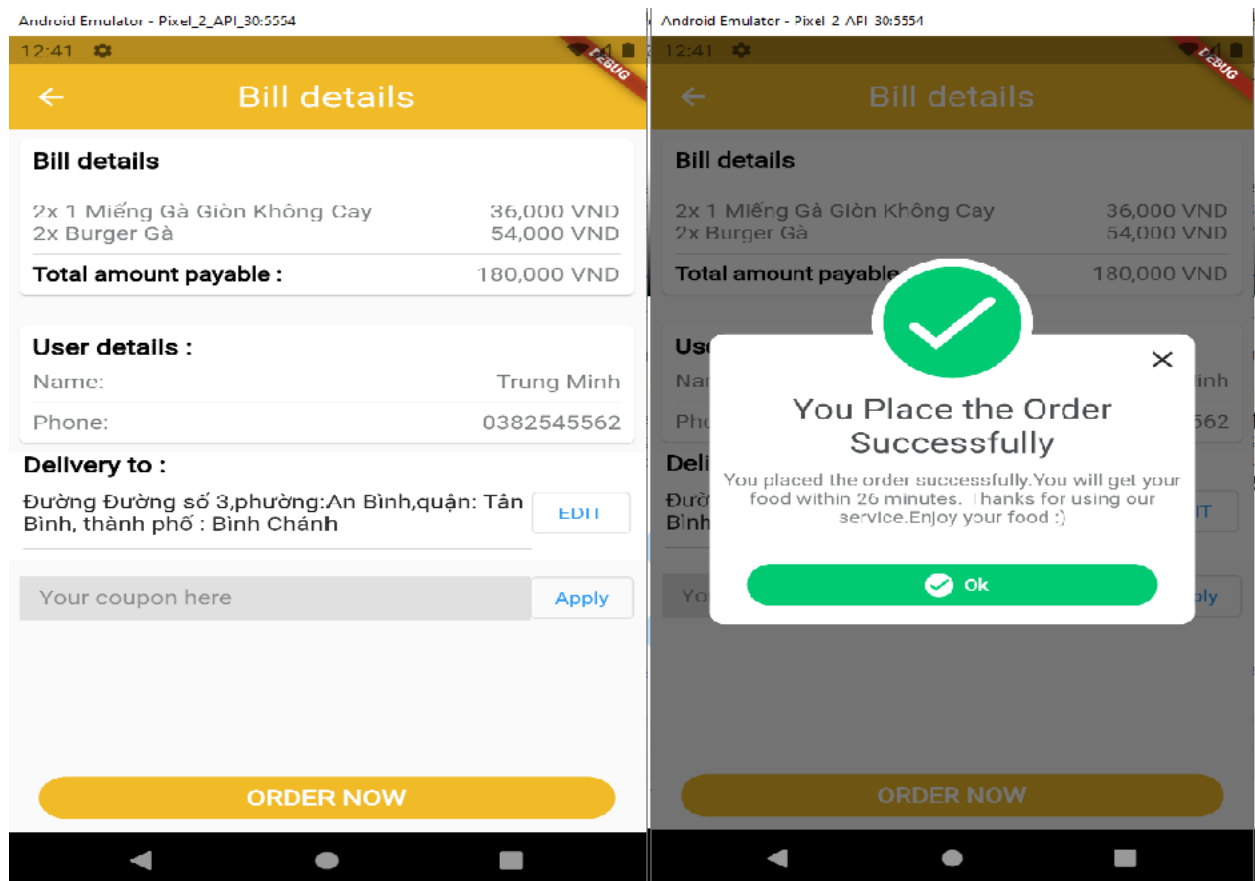
Hình 3. 9 Giao diện của cửa hàng

Giao diện trang chủ(Home) : hiển thị các danh sách món của hàng đang có theo loại.

Giao diện bình luận/đánh giá(Reviews): hiển thị danh sách đánh giá bởi các khách hàng.

Giao diện thông tin(Info): Hiển thị thông tin như địa chỉ và số điện thoại của cửa hàng.

Ngoài ra bên góc phải còn có chỗ cho khách hàng bấm để theo dõi cửa hàng.

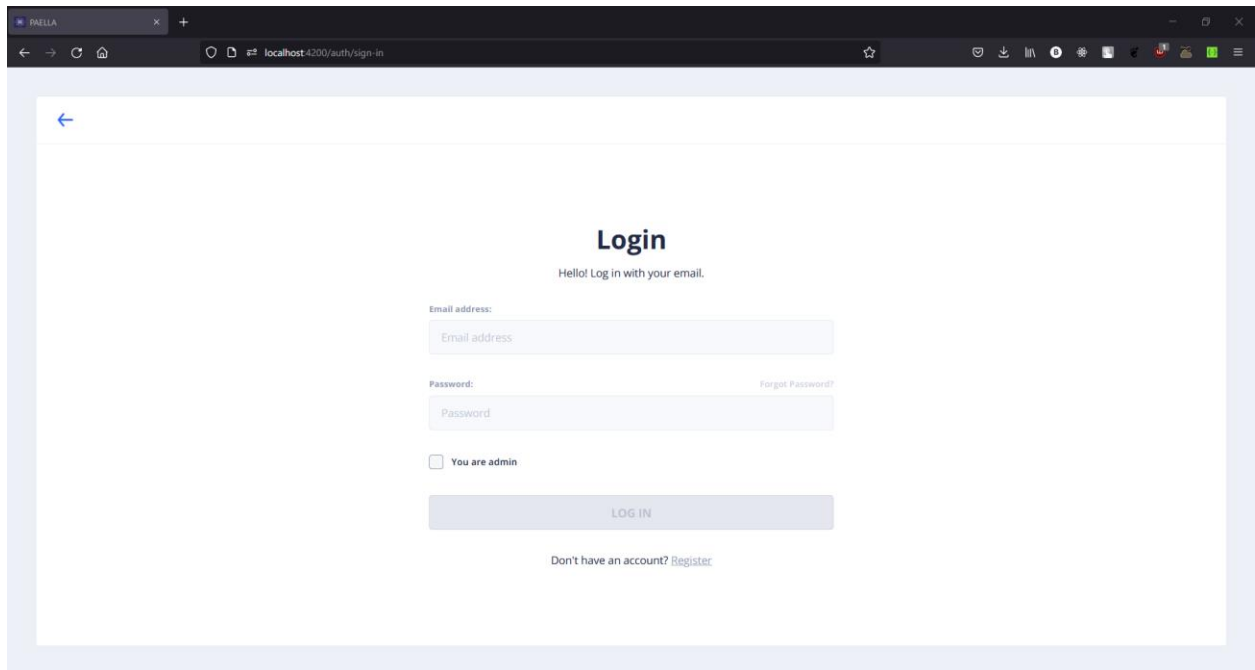


Hình 3. 10 Giao diện đặt đơn hàng

Giao diện hóa đơn đặt hàng : giao diện này hiển thị sau khi khách hàng đã có thức ăn trong giỏ hàng và bấm get bill . Cho phép khách hàng xem thông tin , đơn giá ,sau khi đặt hàng xong.

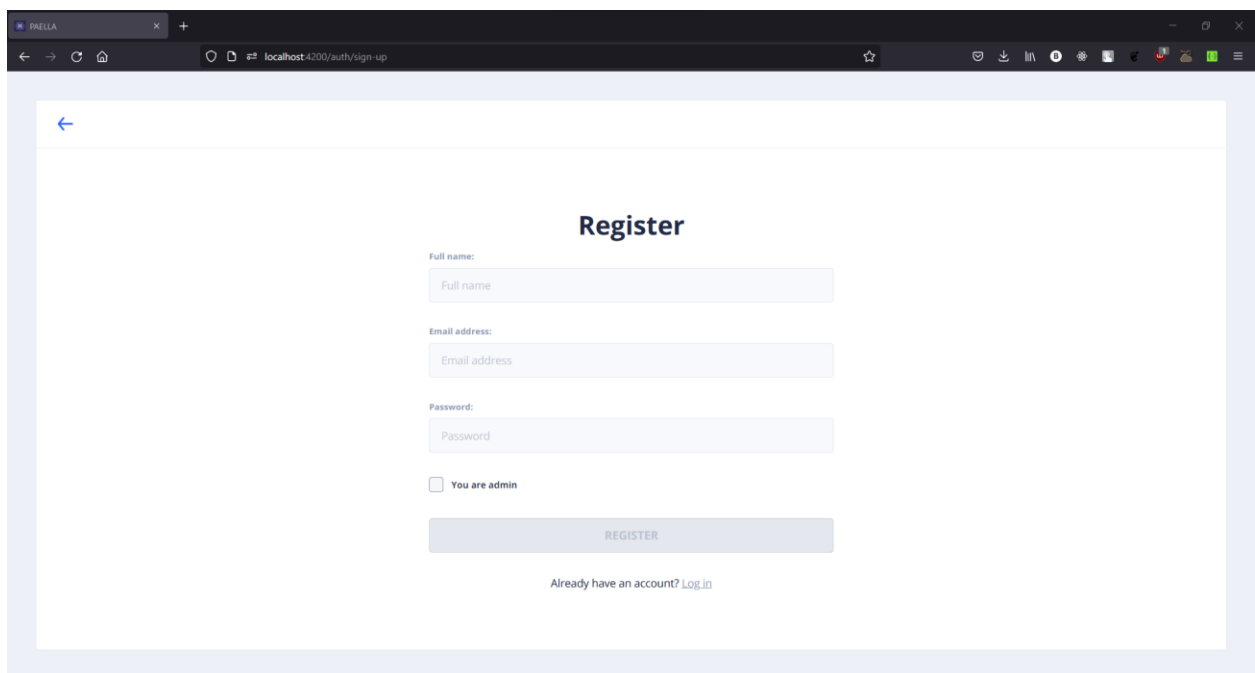
Giao diện thông báo đặt hàng thành công : hiển thị thông báo đã đặt hàng sau khi nhấn ORDER NOW.

b. Phần giao diện của quản trị :



The screenshot shows a web browser window with the address bar displaying "localhost:4200/auth/sign-in". The page has a light blue header with a back arrow. The main content area is white and contains a "Login" form. The form includes a title "Login", a subtitle "Hello! Log in with your email.", and two input fields: "Email address" and "Password". There is a "Forgot Password?" link next to the password field. Below the input fields is a checkbox labeled "You are admin" and a "LOG IN" button. At the bottom, there is a link "Don't have an account? Register".

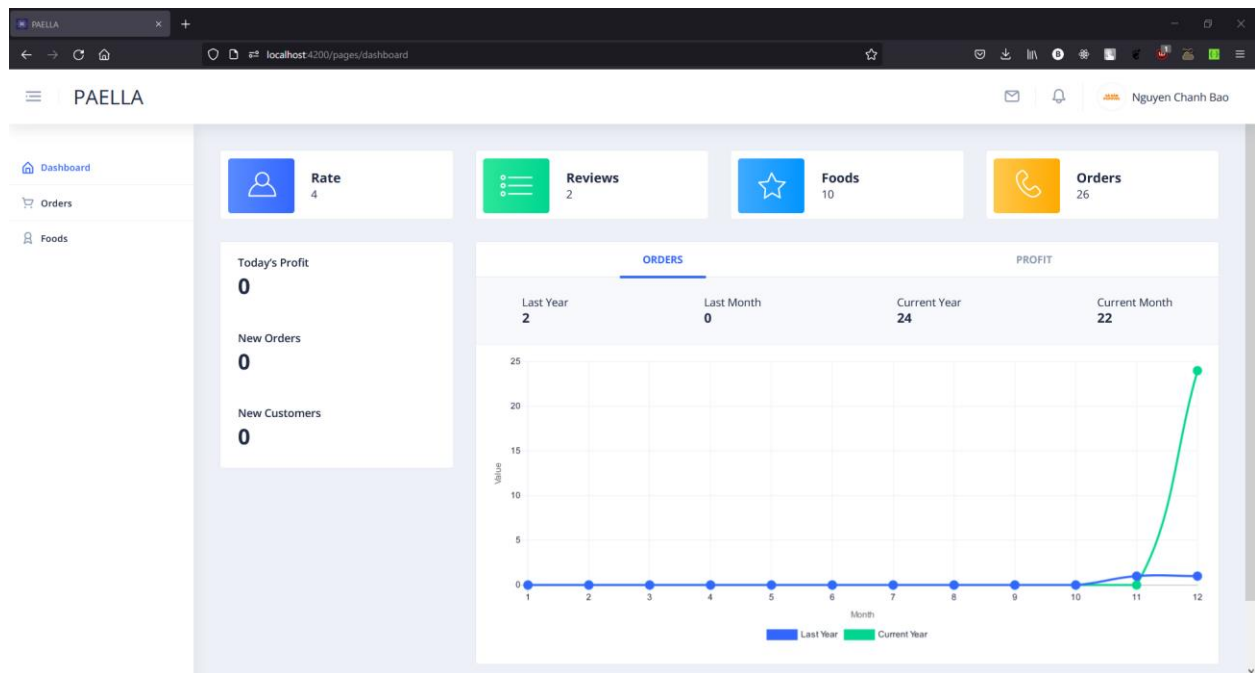
Hình 3. 11 Giao diện đăng nhập



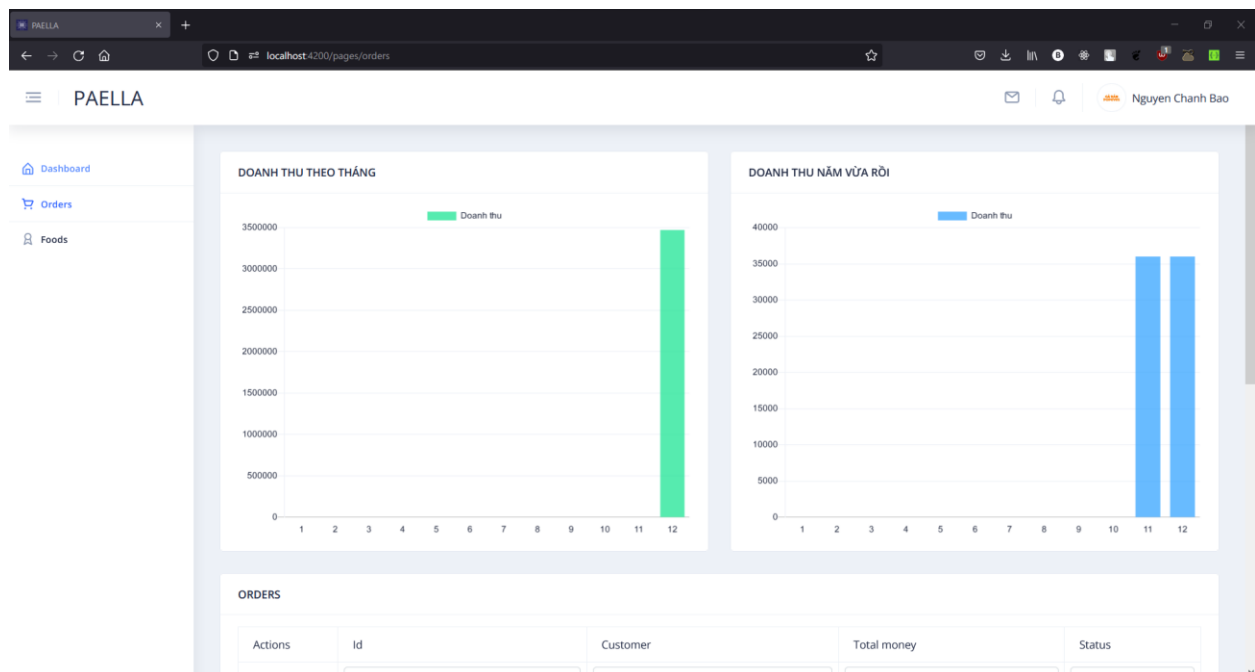
The screenshot shows a web browser window with the address bar displaying "localhost:4200/auth/sign-up". The page has a light blue header with a back arrow. The main content area is white and contains a "Register" form. The form includes a title "Register", a subtitle "Full name:", and three input fields: "Full name", "Email address", and "Password". There is a "Forgot Password?" link next to the password field. Below the input fields is a checkbox labeled "You are admin" and a "REGISTER" button. At the bottom, there is a link "Already have an account? Log in".

Hình 3. 12 Giao diện đăng ký

1. Đối tác :



Hình 3. 13 Giao diện dashboard của đối tác



Hình 3. 14 Giao diện quản lý hóa đơn

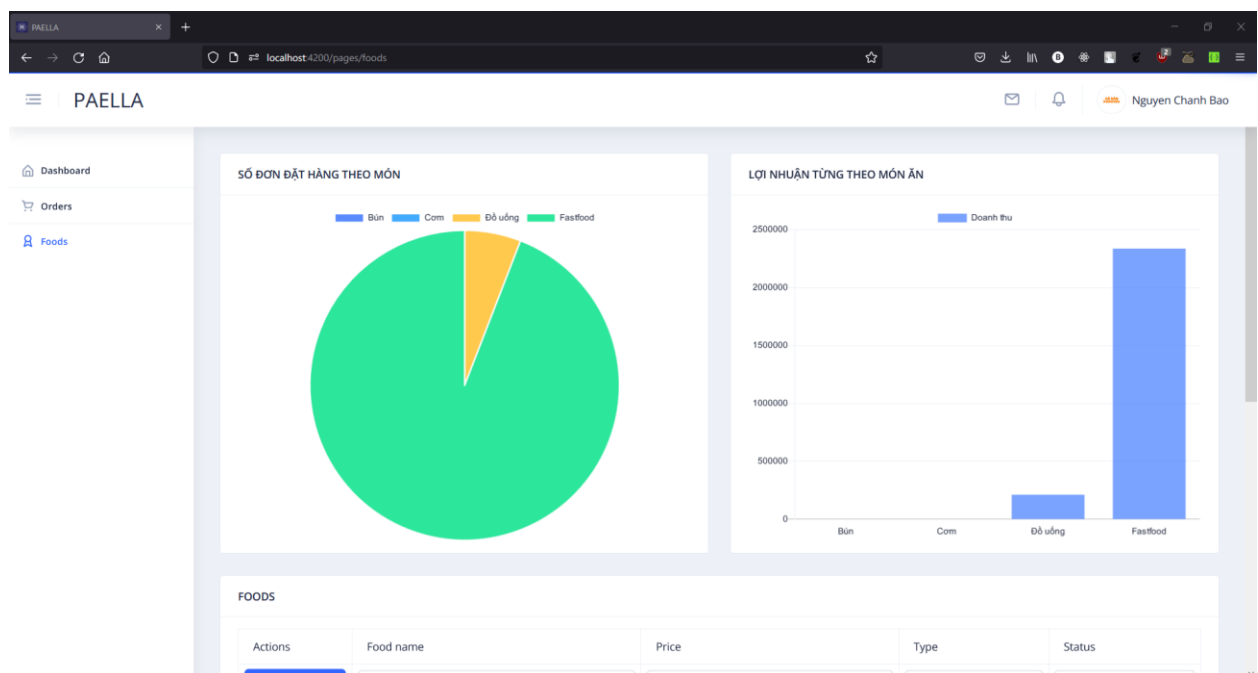
PAELLA

Dashboard Orders Foods

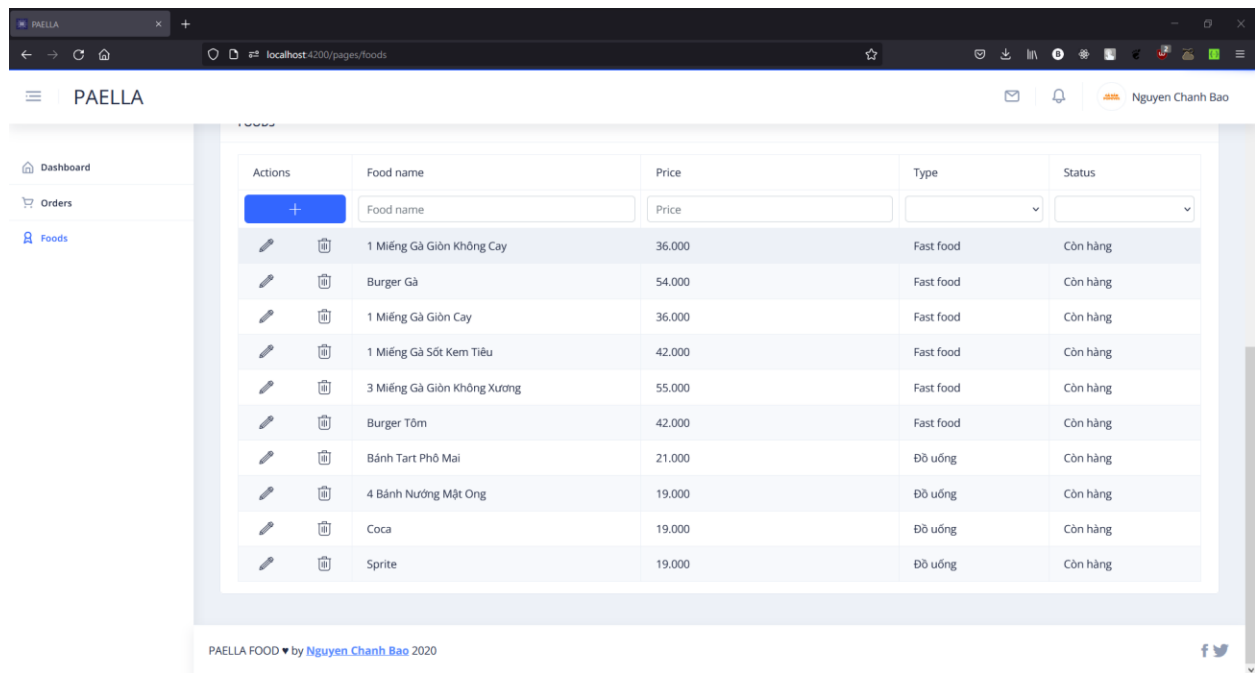
	Id	Customer	Total money	
	6180f1c6199a5ad7f9cffbfc	killuazoldyck.1520@gmail.com	36.000	Đã giao
	6187c70003d5d8d0a8d2ee19	killuazoldyck.1520@gmail.com	36.000	Đã giao
	618a1f2905eb274eb8de4273	killuazoldyck.1520@gmail.com	30.000	Đã giao
	618a1f4105eb274eb8de4276	killuazoldyck.1520@gmail.com	36.000	Đã giao
	61ae157d4bd48d22292852b0	trungvm01@gmail.com	108.000	Đã giao
	61b5a87f8c298a82da181f5e	trungvm01@gmail.com	54.000	Đã giao
	61b5a8b18c298a82da182132	trungvm01@gmail.com	54.000	Đã giao
	61b5a9238c298a82da1825f5	trungvm01@gmail.com	108.000	Đã giao
	61b5a93f8c298a82da18275a	trungvm01@gmail.com	54.000	Đã giao
	61b5a9dc8c298a82da182ef4	trungvm01@gmail.com	54.000	Đã giao

PAELLA FOOD ♥ by [Nguyễn Chanh Bao](#) 2020

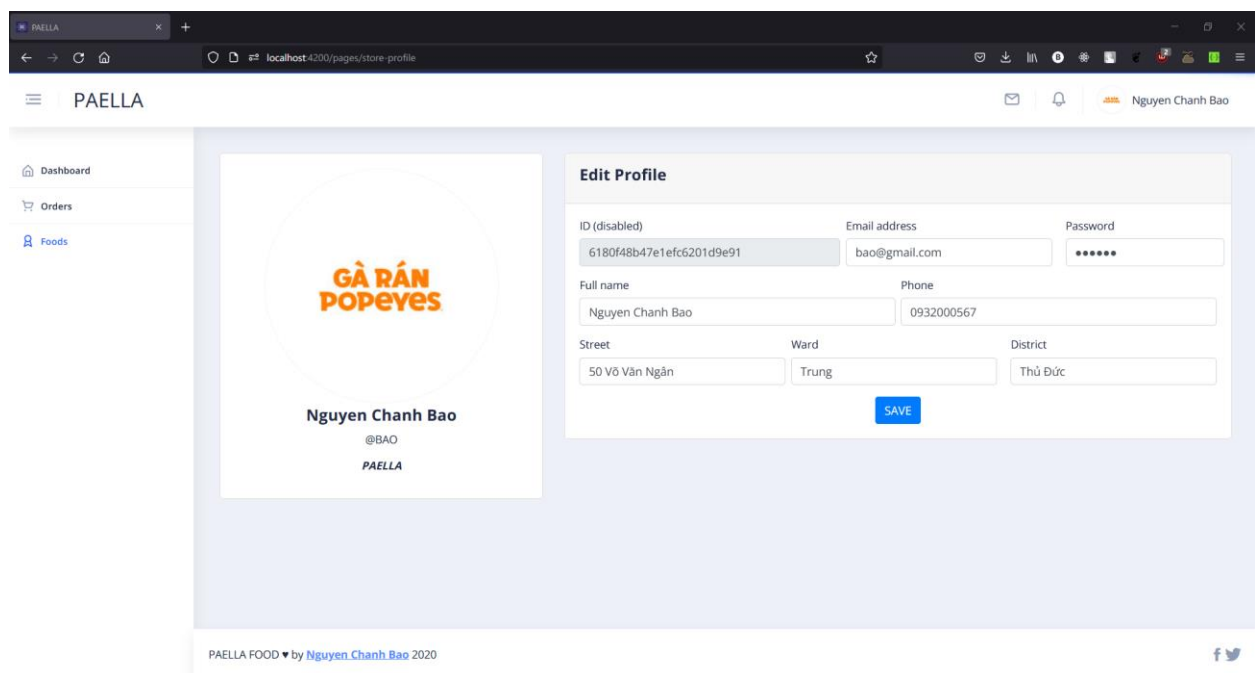
Hình 3. 15 Giao diện quản lý hóa đơn



Hình 3. 16 Giao diện quản lý menu

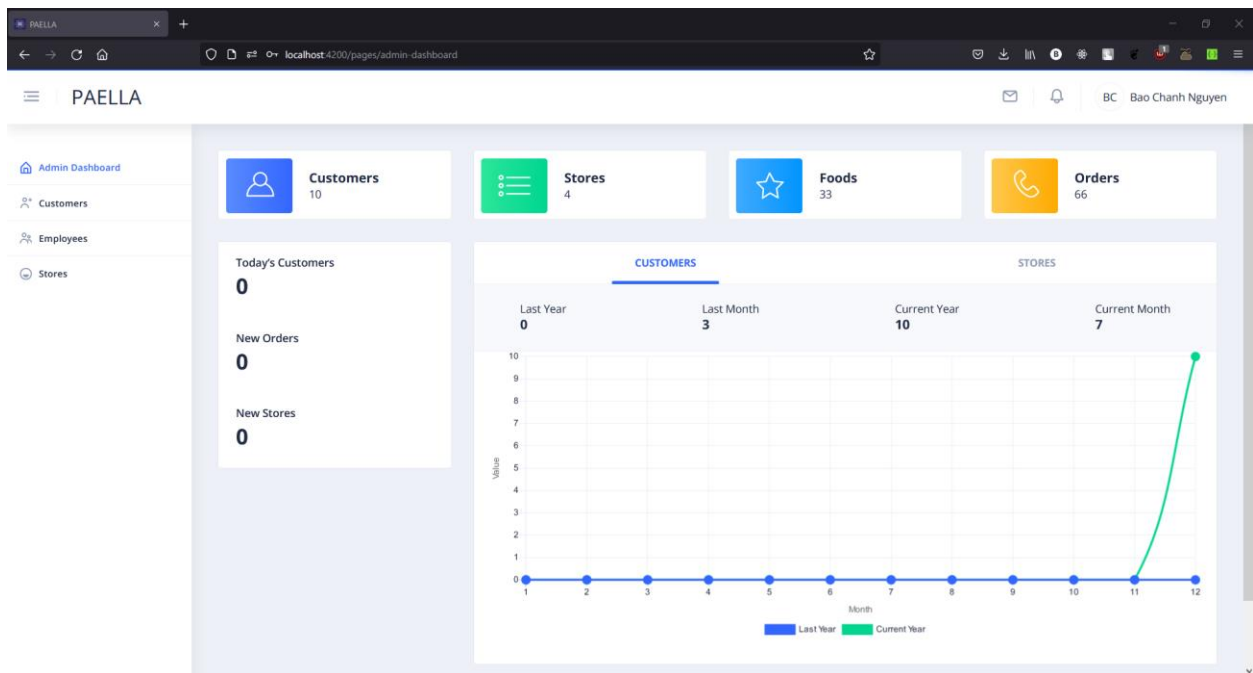


Hình 3. 17 Giao diện quản lý menu

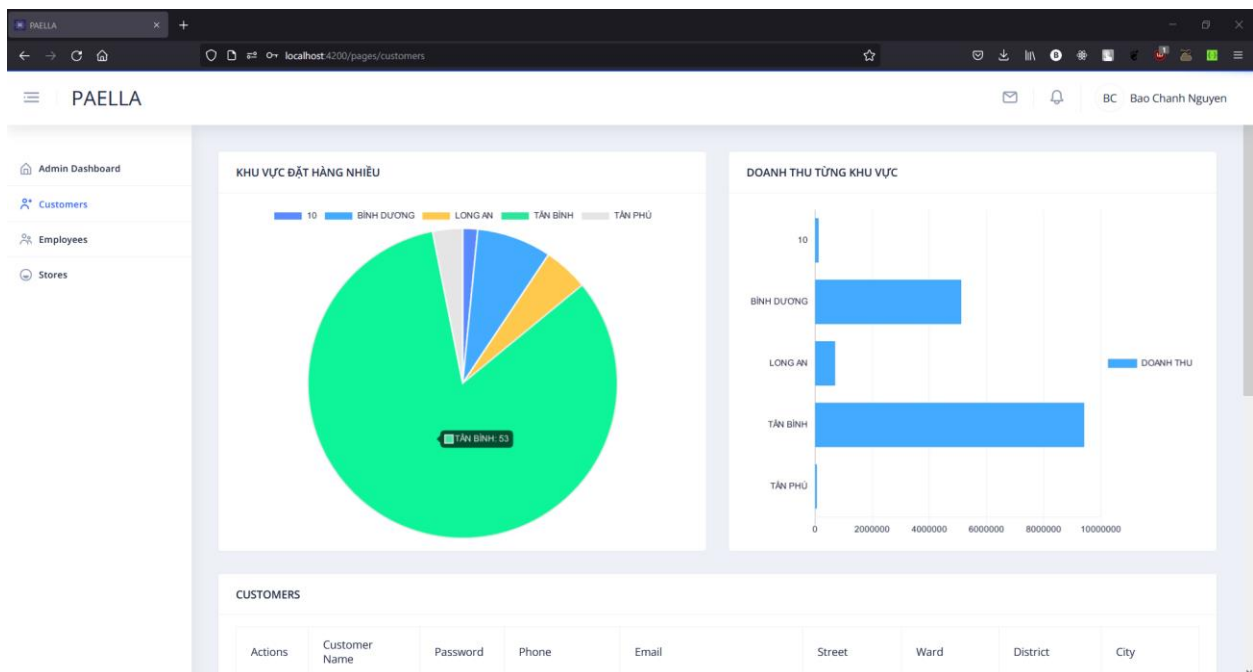


Hình 3. 18 Giao diện thông tin của đối tác

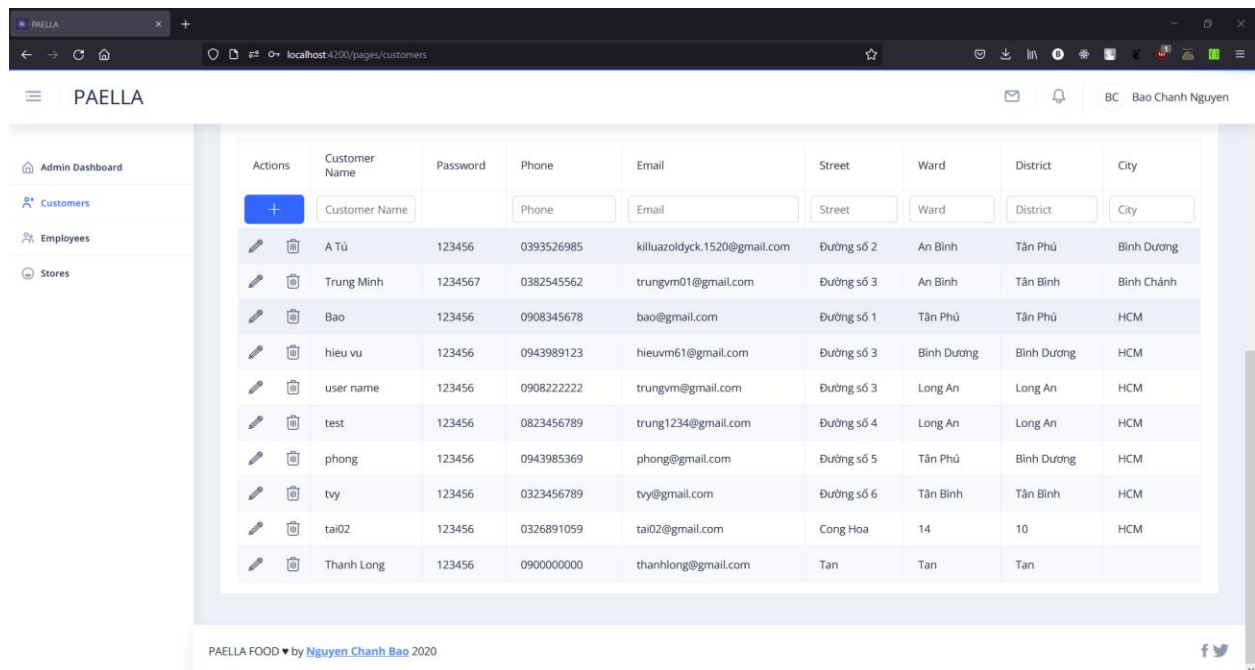
2. Admin :



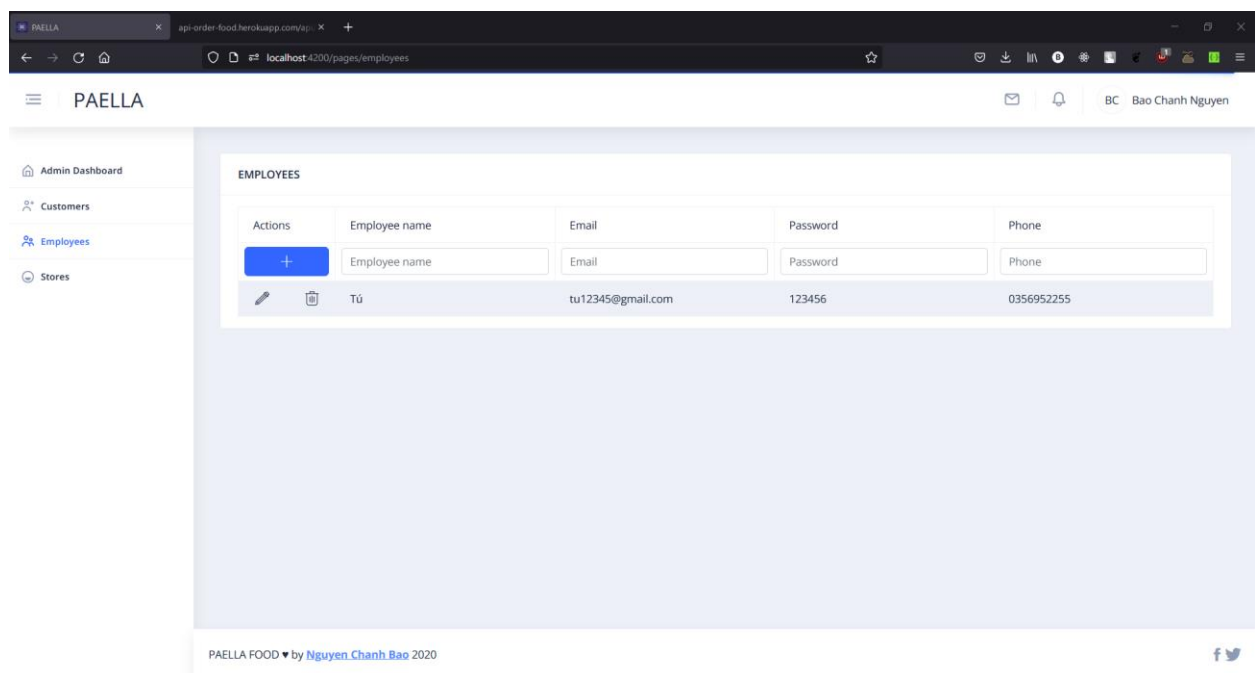
Hình 3. 19 Giao diện dashboard của admin



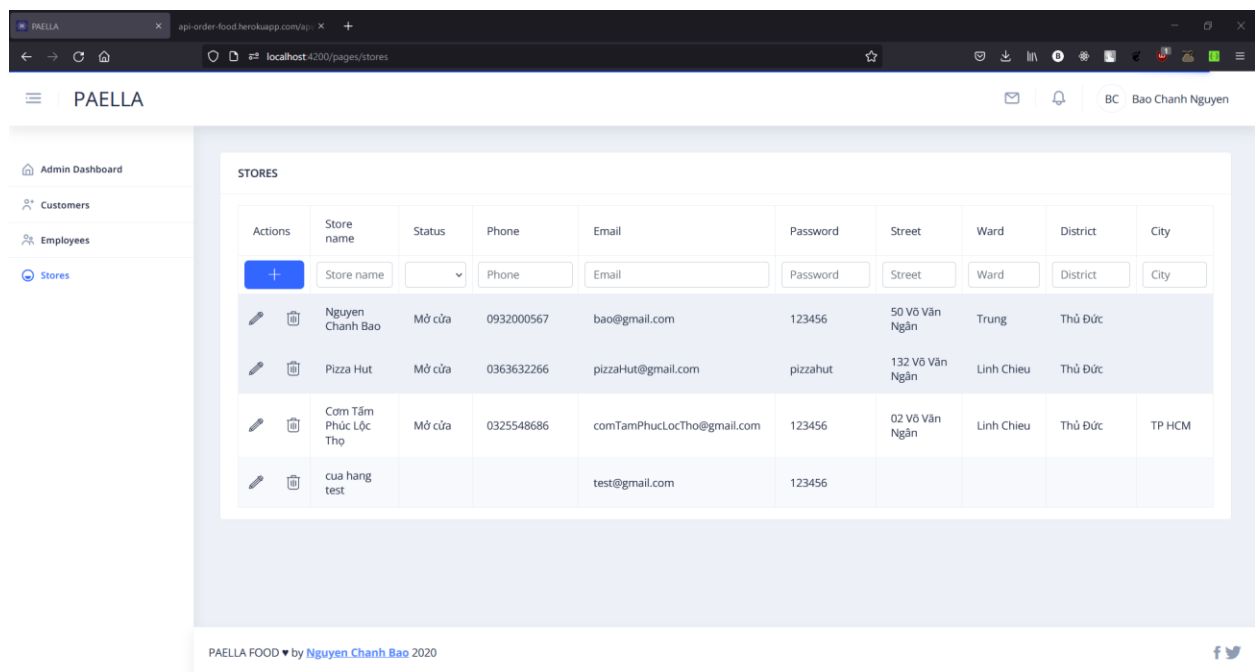
Hình 3. 20 Giao diện quản lý khách hàng



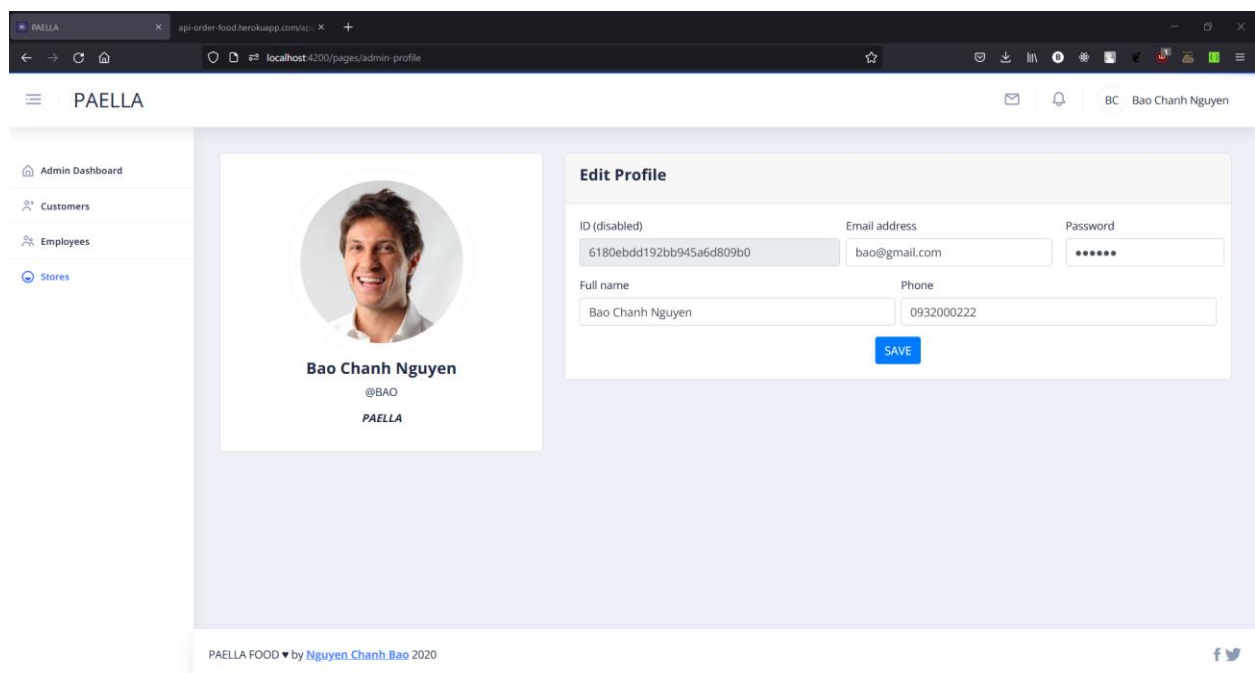
Hình 3. 21 Giao diện quản lý khách hàng



Hình 3. 22 Giao diện quản lý nhân viên



Hình 3. 23 Giao diện quản lý cửa hàng

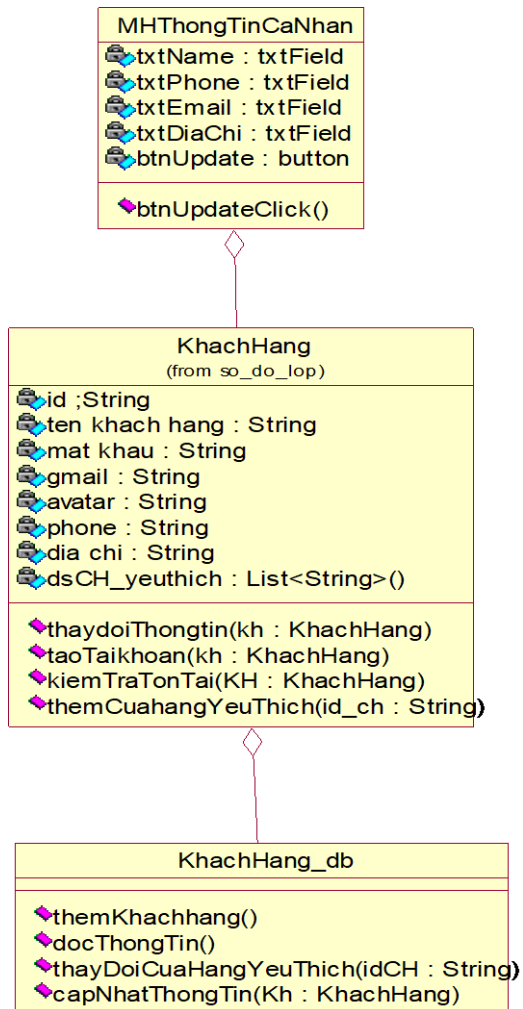


Hình 3. 24 Giao diện thông tin cá nhân

3.5 THIẾT KẾ CHỨC NĂNG HỆ THỐNG

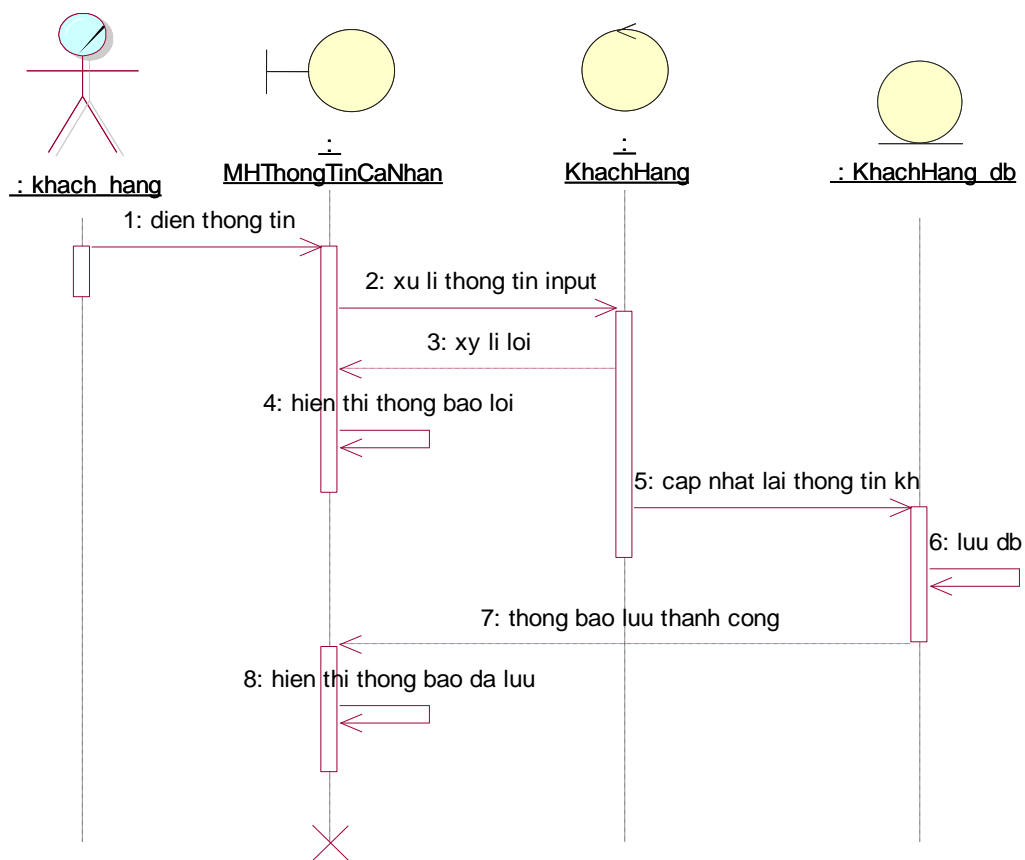
a. Phần ứng dụng khách hàng :

Chức năng cập nhật thông tin cá nhân :



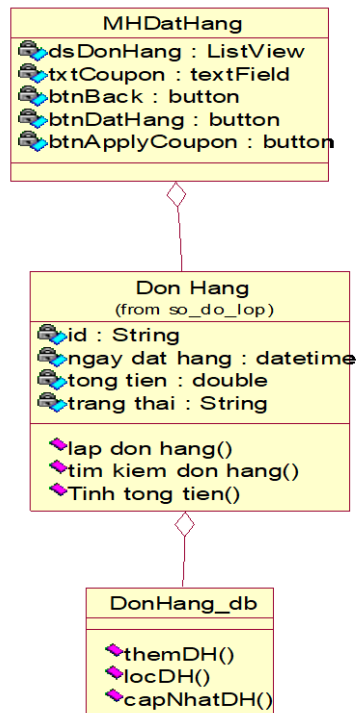
Hình 3. 25 Sơ đồ chức năng (3 lớp) cập nhật thông tin khách hàng

Sơ đồ tuần tự :



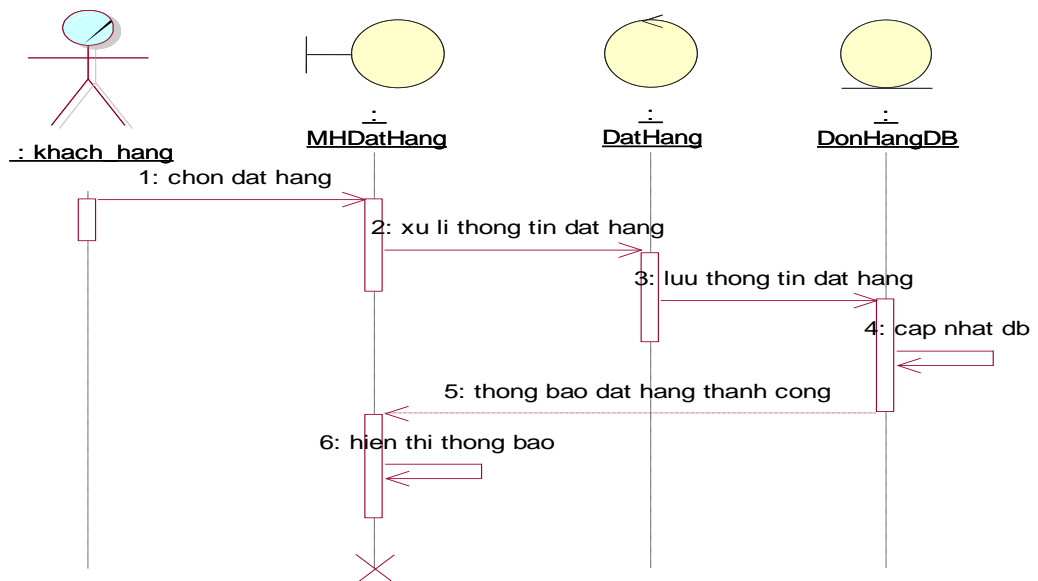
Hình 3. 26 Sơ đồ tuần tự của chức năng cập nhật thông tin.

Chức năng đặt hàng :



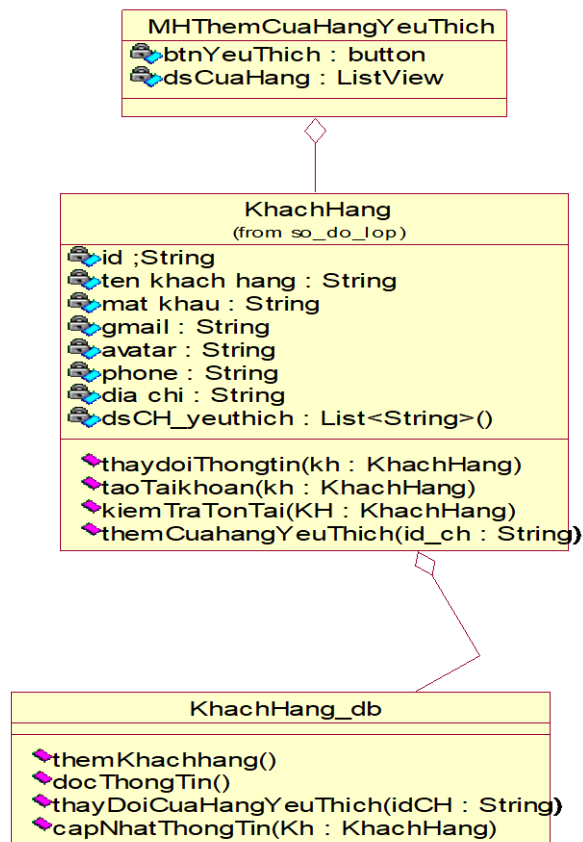
Hình 3. 27 Sơ đồ chức năng (3 lớp) đặt hàng

Sơ đồ tuần tự :



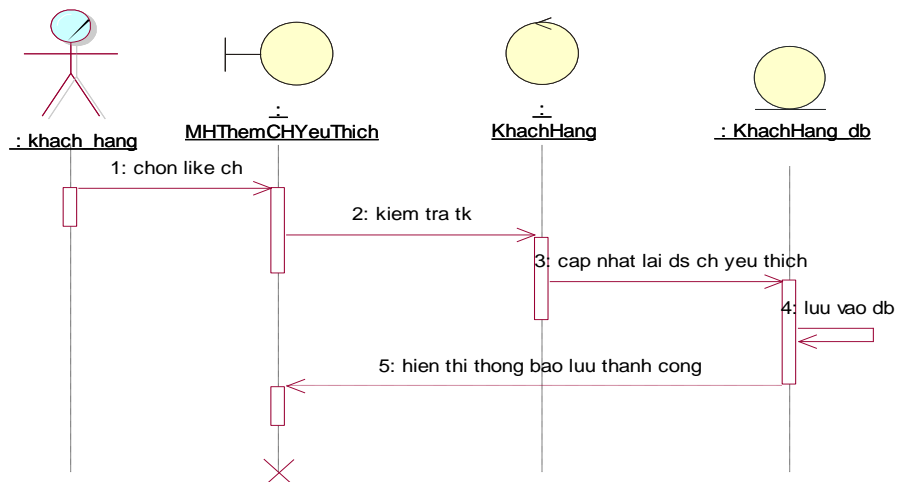
Hình 3. 28 Sơ đồ tuần tự của chức năng đặt hàng.

Chức năng Thêm cửa hàng yêu thích :



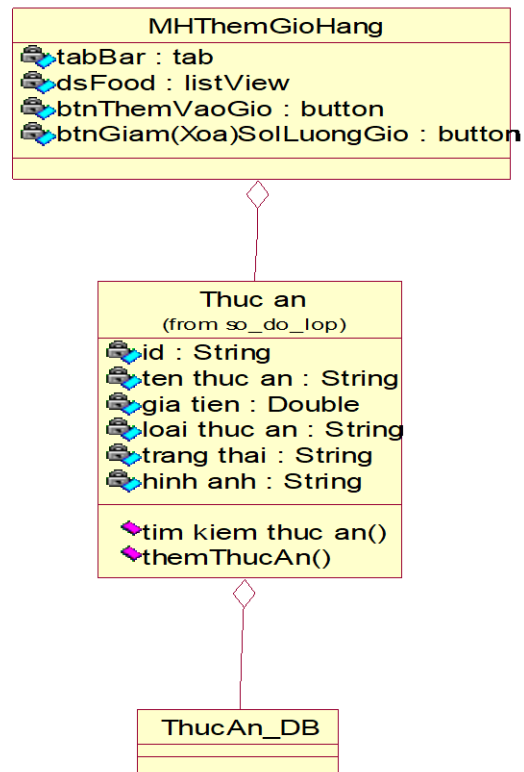
Hình 3. 29 Sơ đồ chức năng (3 lớp) thêm cửa hàng yêu thích

Sơ đồ tuần tự :



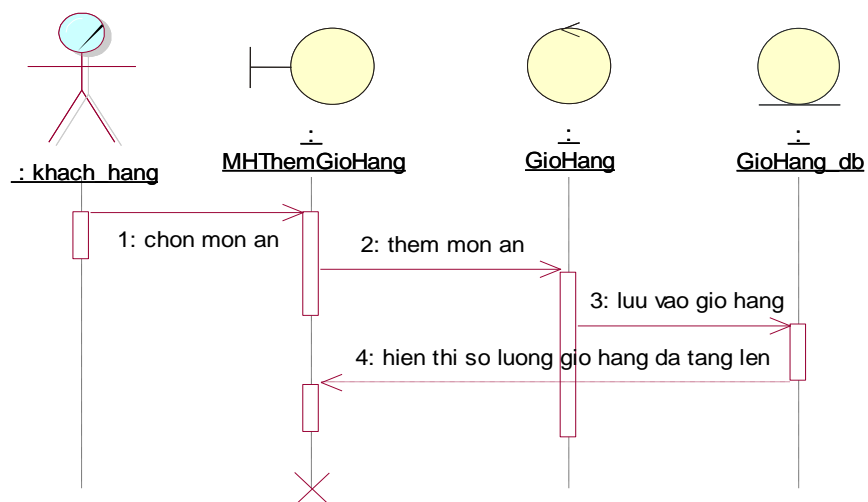
Hình 3. 30 Sơ đồ tuần tự của chức năng thêm sản phẩm yêu thích

Chức năng thêm vào giỏ hàng :



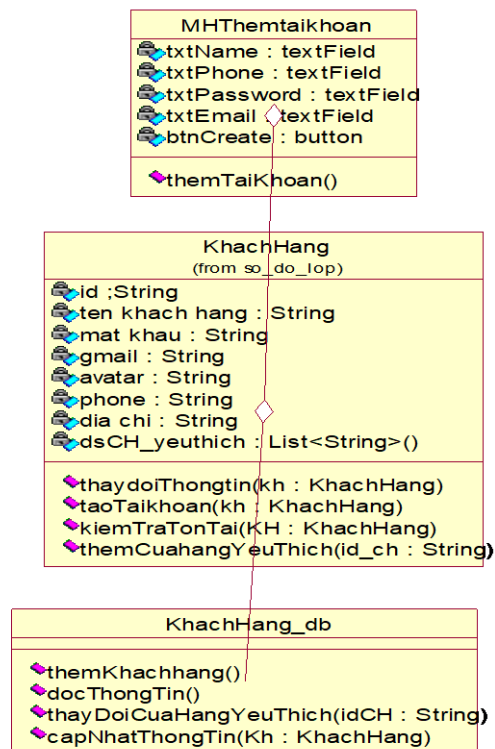
Hình 3. 31 Sơ đồ chức năng (3 lớp) thêm giỏ hàng

Sơ đồ tuần tự :



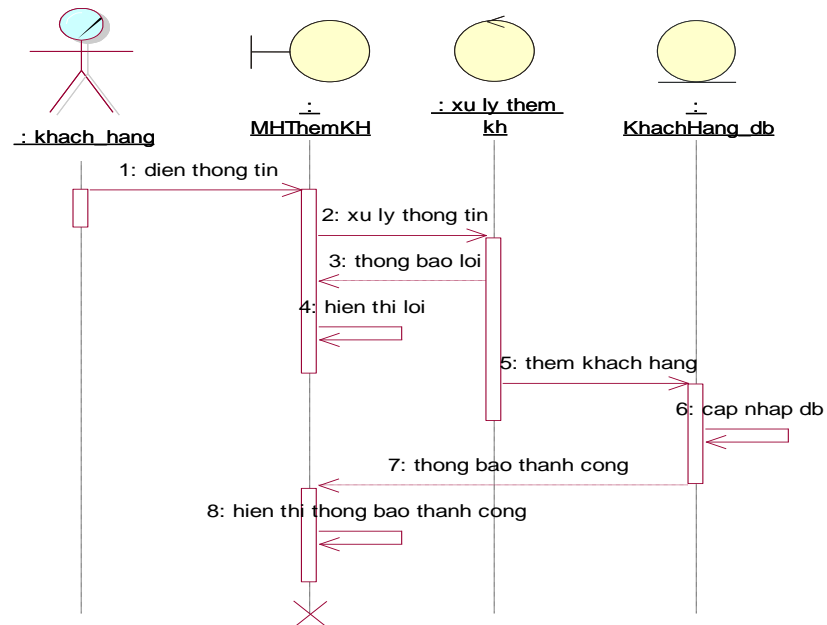
Hình 3. 32 Mô hình tuần tự của chức năng thêm vào giỏ hàng.

Chức năng thêm tài khoản :



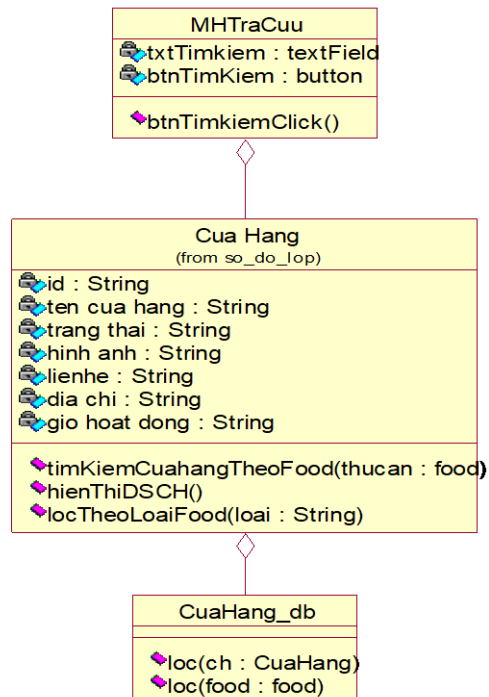
Hình 3. 33 Sơ đồ chức năng (3 lớp) thêm tài khoản

Sơ đồ tuần tự:



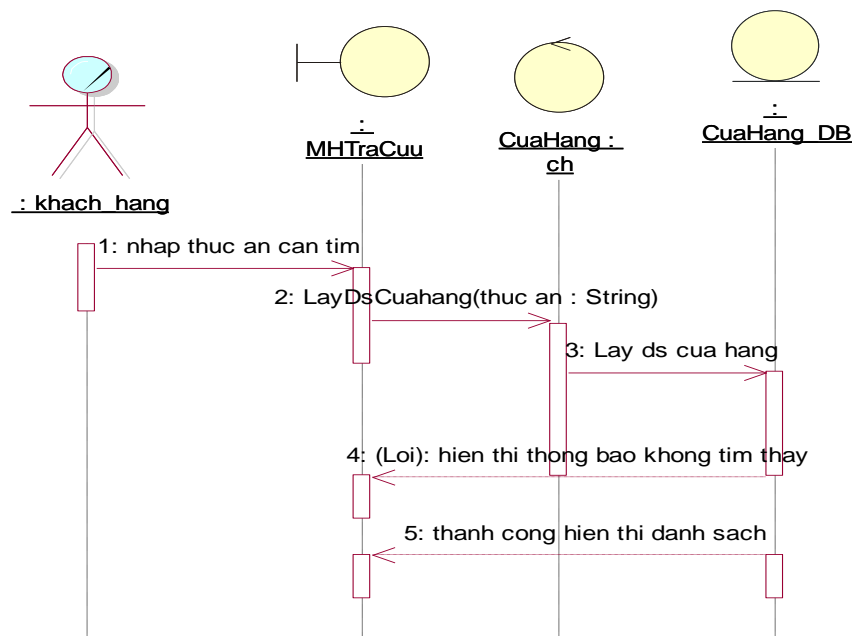
Hình 3. 34 Sơ đồ tuần tự của chức năng đăng ký.

Chức năng tra cứu :



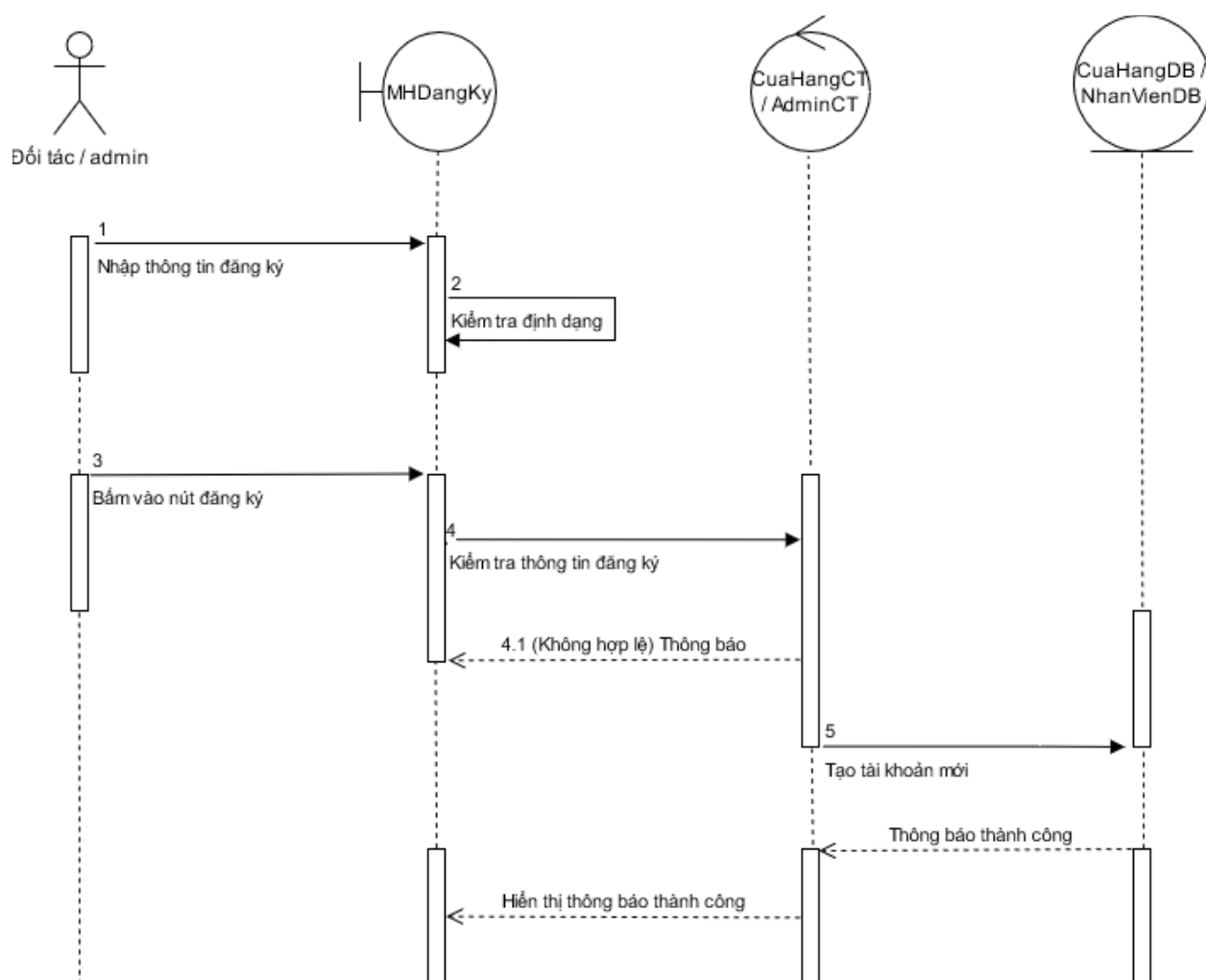
Hình 3. 35 Sơ đồ chức năng (3 lớp) tra cứu

Sơ đồ tuần tự :

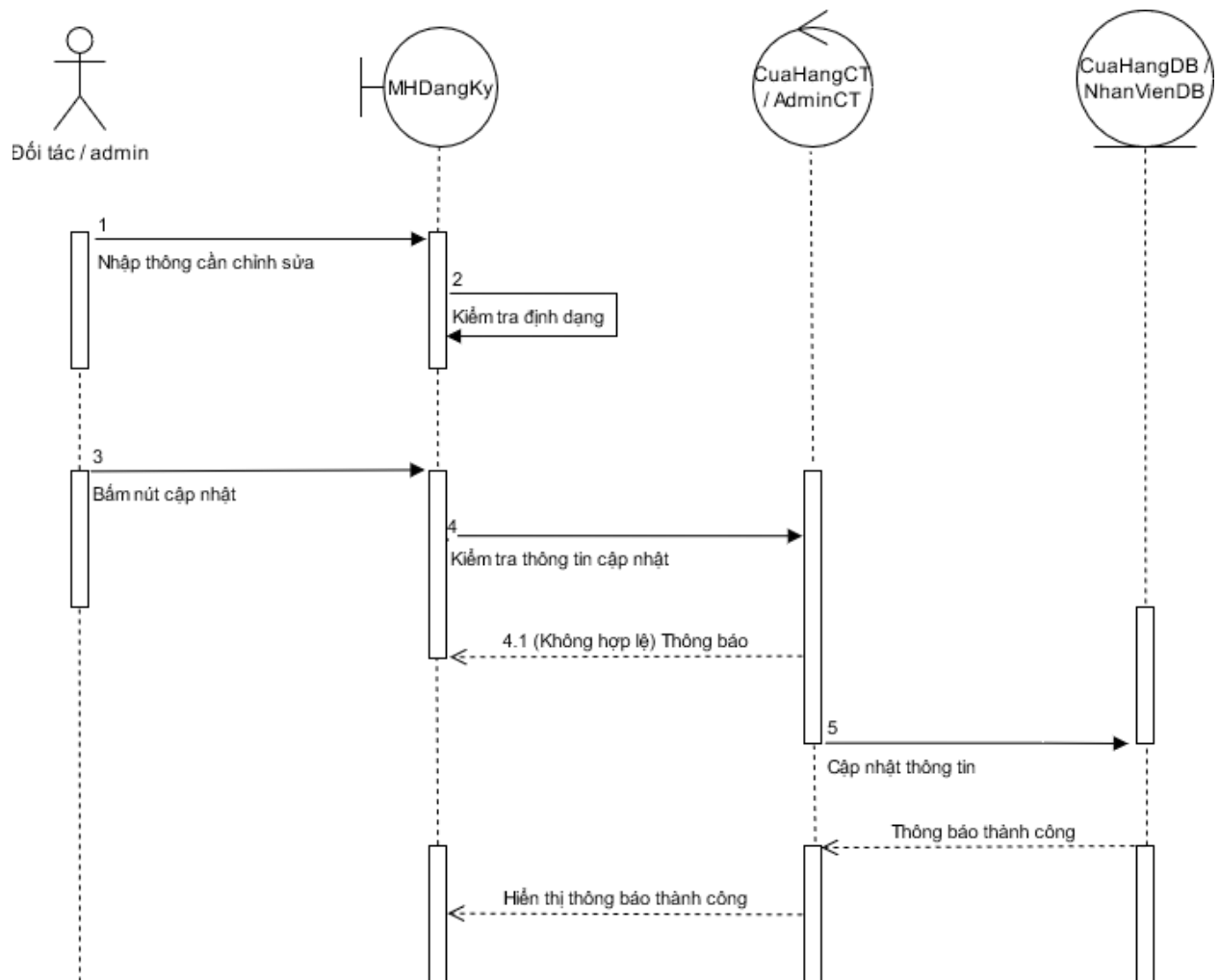


Hình 3. 36 Sơ đồ tuần tự của chức năng tra cứu.

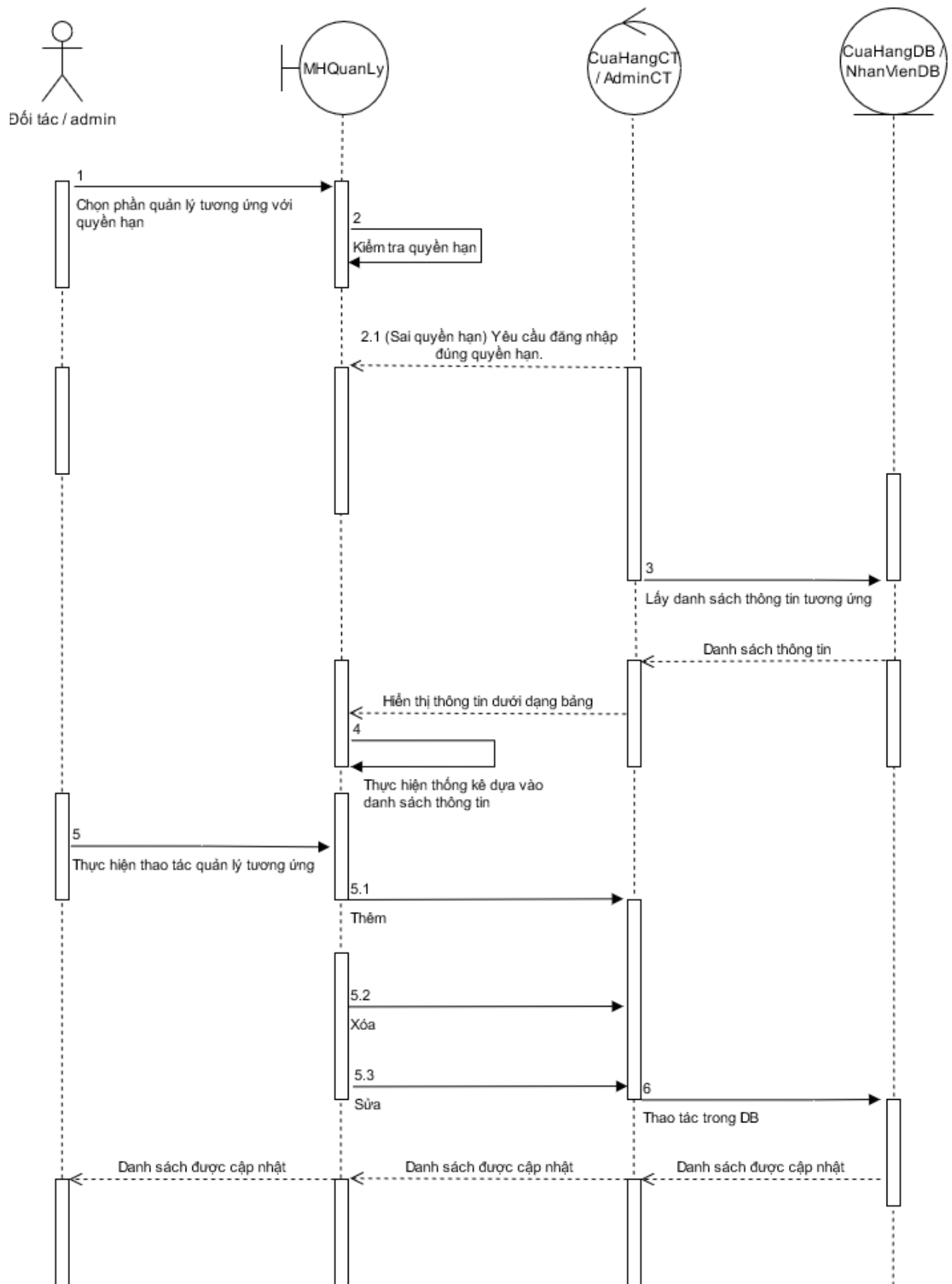
b. Đối tác/admin :



Hình 3. 37 Sơ đồ tuần tự của chức năng đăng ký.



Hình 3. 38 Sơ đồ tuần tự của chức năng cập nhật thông tin.



Hình 3. 39 Sơ đồ tuần tự của chức năng quản lý tương tự

3.6 THIẾT KẾ API :

Link API: <https://api-order-food.herokuapp.com/api>

CRUD to Collections in MongoDB Atlas

- GET

ENDPOINT	CHỨC NĂNG
/customers	Lấy thông tin tất cả khách hàng
/customers/:id	Lấy thông tin khách hàng theo id
/foods	Lấy thông tin tất cả món ăn
/foods/:id	Lấy thông tin món ăn theo id
/coupons	Lấy thông tin tất cả khuyến mãi
/coupons/:id	Lấy thông tin khuyến mãi theo id
/stores	Lấy thông tin tất cả cửa hàng
/stores/:id	Lấy thông tin cửa hàng theo id
/orders	Lấy thông tin tất cả hoá đơn
/orders/:id	Lấy thông tin hoá đơn theo id
/employees	Lấy thông tin tất cả nhân viên
/employees/:id	Lấy thông tin nhân viên theo id

Bảng 3. 10 GET API

- POST

ENDPOINT	CHỨC NĂNG
/customers	Tạo mới một khách hàng
/foods	Tạo mới một món ăn
/coupons	Tạo mới một khuyến mãi
/stores	Tạo mới một cửa hàng
/orders	Tạo mới một hoá đơn
/employees	Tạo mới một nhân viên

Bảng 3. 11 POST API

- PUT

ENDPOINT	CHỨC NĂNG
/customers/:id	Cập nhật khách hàng theo id
/foods/:id	Cập nhật món ăn theo id
/coupons/:id	Cập nhật khuyến mãi theo id
/stores/:id	Cập nhật cửa hàng theo id
/orders/:id	Cập nhật hoá đơn theo id
/employees/:id	Cập nhật nhân viên theo id

Bảng 3. 12 PUT API

- DELETE

ENDPOINT	CHỨC NĂNG
/customers/:id	Xoá khách hàng theo id
/foods/:id	Xoá món ăn theo id
/coupons/:id	Xoá khuyến mãi theo id
/stores/:id	Xoá cửa hàng theo id
/orders/:id	Xoá hoá đơn theo id
/employees/:id	Xoá nhân viên theo id

Bảng 3. 13 DELETE API

I. Nghiệp vụ.

1. Customer

- /customer/sign-in

Method: POST

Input: {

“email”: “email”,

“password”: “password”

}

Output: token

- /customer/sign-in/:token

Method: GET

Input: token

Output: true/false

- /customer/getFavoriteStores

Method: POST

Input: {

“love_store_ids” : [“StoreID”,...]

}

Output: [{ObjectStore},...]

- /customer/getOrders/:customer_id

Method: GET

Input: customer_id

Output: [{ObjectOrder},...]

- /customer/searchFood/:text

Method: GET

Input: text

Output: [{ObjectStore},...]

- /customer/reviewStore

Method: POST

Input: {

“storeID”: “store_id”,

“customer_id”: “customer_id”,

“rate”: “rate”

}

Output: message result update

- **/customer/searchByCategory/:category**

Method: GET

Input: name of category

Output: [{ObjectStore},...]

- **/customer/searchByNewest**

Method: GET

Input: None

Output: [{ObjectStore},...]

- **/customer/filterOrdersDayToDay**

Method: POST

Input: customer_id, startDay, endDay

Output: [{ObjectOrder},...]

- **/customer/forgotPassword**

Method: POST

2. Admin

- **/auth/sign-in**

Method: POST

Input: {

“email”: “email”,

“password”: “password”,

“isAdmin”: true/false

}

Output: token

- **/admin/sign-in/:token**

Method: GET

Input: email

Output: Message info email sent

- **/customer/checkVerificationCode**

Method: POST

Input: code, customer_id

Output: true/false

- **/customer/resetPassword**

Method: POST

Input: customer_id, password

Output: Message

- **/customer/getReviewsOfStore/:store_id**

Method: GET

Input: store_id

Output: [{ObjectReview},...]

Input: token

Output: true/false

- **/admin/getOrdersCurrentDate**

Method: GET

Input: None

Output: [{ObjectOrder},...]

- **/admin/getOrdersCurrentWeek**

Method: GET

Input: None

Output: [{ObjectOrder},...]

- **/admin/getOrdersCurrentMonth**

Method: GET

Input: None

Output: [{ObjectOrder},...]

- **/admin/getOrdersCurrentYear**

Method: GET

Input: None

Output: [{ObjectOrder},...]

- **/admin/getOrdersLastWeek**

Method: GET

Input: None

Output: [{ObjectOrder},...]

- **/admin/getOrdersLastMonth**

Method: GET

Input: None

Output: [{ObjectOrder},...]

- **/admin/getOrdersLastYear**

Method: GET

Input: None

Output: [{ObjectOrder},...]

- **/admin/getNewCustomersCurrentDate**

Method: GET

Input: None

Output: [{ObjectCustomer},...]

- **/admin/getNewCustomersCurrentMonth**

Method: GET

Input: None

Output: [{ObjectCustomer},...]

- **/admin/getNewCustomersCurrentYear**

Method: GET

Input: None

Output: [{ObjectCustomer},...]

- **/admin/getNewCustomersLastMonth**

Method: GET

Input: None

Output: [{ObjectCustomer},...]

- **/admin/getNewCustomersLastYear**

Method: GET

Input: None

Output: [{ObjectCustomer},...]

- **/admin/getNewStoresCurrentDate**

Method: GET

Input: None

Output: [{ObjectStore},...]

- **/admin/getNewStoresCurrentMonth**

Method: GET

Input: None

Output: [{ObjectStore},...]

- **/admin/getNewStoresCurrentYear**

Method: GET

Input: None

Output: [{ObjectStore},...]

- **/admin/getNewStoresLastMonth**

Method: GET

Input: None

Output: [{ObjectStore},...]

- **/admin/getNewStoresLastYear**

Method: GET

Input: None

Output: [{ObjectStore},...]

- **/admin/revenueMonthOfYear**

Method: GET

Input: None

Output: [int,...]

3. Store

- **/store/sign-in/:token**

Method: GET

Input: token

Output: true/false

- **/store/getOrders/:store_id**

Method: GET

Input: store_id

Output: [{ObjectOrder},...]

- **/store/getOrdersCurrentDate/:store_id**

Method: GET

Input: store_id

Output: [{ObjectOrder},...]

- **/store/getOrdersCurrentWeek/:store_id**

Method: GET

Input: store_id

Output: [{ObjectOrder},...]

- **/store/getOrdersCurrentMonth/:store_id**

Method: GET

Input: store_id

Output: [{ObjectOrder},...]

- **/store/getOrdersCurrentYear/:store_id**

Method: GET

Input: store_id

Output: [{ObjectOrder},...]

- **/store/getOrdersLastWeek/:store_id**

Method: GET

Input: store_id

Output: [{ObjectOrder},...]

- **/store/getOrdersLastMonth/:store_id**

Method: GET

Input: store_id

Output: [{ObjectOrder},...]

- **/store/getOrdersLastYear/:store_id**

Method: GET

Input: store_id

Output: [{ObjectOrder},...]

- **/store/updateOrderDelievered**

Method: POST

Input: order_id

Output: Message

CHƯƠNG 4

CÀI ĐẶT THỰC NGHIỆM VÀ ĐÁNH GIÁ

Với mô hình 3 lớp được xây dựng ở chương 3 thì nhóm chúng em đã thiết kế được xong được giao diện của ứng dụng. Để giải quyết được các vấn đề nghiệp vụ đưa ra như lưu trữ database, thông báo, authorize thì nhóm chúng em đã tích hợp sử dụng các công nghệ như Firebase notification, mongodb, heroku, nodejs, angular.

4.1 KẾT QUẢ ĐẠT ĐƯỢC :

Sau hơn 4 tháng nghiên cứu và làm việc nghiêm túc, trong sự theo dõi và hướng dẫn trực tiếp của thầy Vũ Đình Ái và thầy Vũ Văn Vinh, nhóm chúng em đã hoàn thành khóa luận đúng hạn, hoàn chỉnh một ứng dụng di động, một website với các chức năng khá đầy đủ, đáp ứng được các yêu cầu và mục tiêu đã đề ra và một tài liệu báo cáo đính kèm. Tổng kết lại những gì làm được trong 4 tháng qua, chúng em đã làm được :

- Đã thực hiện tìm hiểu về Cross Platform Flutter, là một yếu tố quan trọng trong việc hoàn thành đề tài cũng như bổ sung kiến thức về Flutter với IOT phục vụ hỗ trợ bổ sung kiến thức và áp dụng vào việc làm.
- Ứng dụng sử dụng firebase Notification để có thể gửi các thông báo nhanh và ít tốn kém.
- Tìm hiểu được design-pattern của kiến trúc Blocprovider.
- Thiết kế ứng dụng Web chuẩn SPA với framework Angular.
- Biết cách giao tiếp với Server trong Angular thông qua API.
- Thiết kế được một số chức năng thống kê và trực quan hóa thông tin thông qua đồ thị.
- Rèn luyện và trau dồi kỹ năng làm việc nhóm, cũng như việc triển khai một dự án theo quy trình. Đồng thời nâng cao kỹ năng hoàn thiện bài báo cáo cũng như trình bày sản phẩm trước mọi người.

4.2 ĐÁNH GIÁ KẾT QUẢ :

Với 4 tháng nghiên cứu thì nhóm chúng em cũng đã hoàn thành công việc được giao. Tuy nhiên vẫn còn vài bất cập như :

- Khi thử nghiệm trên nền tảng IOS như bảo mật của nền tảng quá cao nên nhóm chúng em chưa thể đánh giá hiệu năng cũng như sự ổn định của phần mềm trên nền tảng.
- Những chức năng còn mang tính phổ thông đơn giản, chưa cung cấp các chức năng như thanh toán thẻ tín dụng, đăng nhập với google, tích hợp google map, đăng tải hình ảnh bình luận.
- Giao diện người dùng dễ nhìn nhưng còn hạn chế về sự sáng tạo và tối ưu UI/UX.
- Phần thống kê của ứng dụng Web còn chậm. Một phần là do phải gọi API tới server để có thông tin thống kê. Một phần nữa là do những hạn chế trong việc thiết kế Database mà dẫn đến phải gọi API nhiều lần để có đủ thông tin thống kê.

TÀI LIỆU THAM KHẢO

AI VU DINH Document-Google Drive [Online] // Document-Google Drive. - <https://drive.google.com/drive/u/0/folders/1tzxeclpu5njssC6YR1MpExH2FJLSbmtP>.

bão Mất Firebase là gì?Giải pháp lập trình [Trực tuyến] // Firebase là gì?Giải pháp lập trình. - 27 12 2021. - <https://wiki.matbao.net/firebase-la-gi-giai-phap-lap-trinh-khong-can-backend-tu-google/>.

Dat Tran Hướng dẫn bắt notification bằng FCM [Trực tuyến] // VIBLO. - 10 7 2017. - <https://viblo.asia/p/huong-dan-ban-notification-bang-fcm-firebase-cloud-messaging-api-63vKj6yAK2R>.

Hoang Nguyen Duc Nguyen Duc Hoang - Youtube [Online] // Nguyen Duc Hoang - Youtube. - 2020. - <https://www.youtube.com/channel/UCfjYtjTW69DWMLOOMzFJ5cg?app=desktop>.

NGOC VU TUAN Giới thiệu về Flutter [Trực tuyến] = Giới thiệu về flutter // Viblo. - 3 20 2018. - <https://viblo.asia/p/gioi-thieu-ve-flutter-bWrZnNxrZxw>.

akveo What is Nebular? [Online] // ngx-dashboard. - <https://akveo.github.io/nebular/docs/getting-started/what-is-nebular#what-is-nebular>.

Angular [Online]. - <https://angular.io/start>.

Codevolution Angular Tutorial Youtube [Online] // Youtube. - <https://www.youtube.com/watch?v=0eWrpsCLMJQ&list=PLC3y8-rFHvwhBRAGFinJR8KHlrCdTkZcZ>.

Education 200lap Flutter Tutorial 2021 for Beginners: English App [Online] // 200lab Education-Youtube. - 2021. - <https://www.youtube.com/channel/UCXc-GBhPcGrzwkZt67h4suQ?app=desktop>.

Flutterly Flutterly -youtube [Online] // Flutterly -youtube. - <https://www.youtube.com/channel/UC5PYcSe3to4mtm3SPCUMjvw>.

JavaScript Mastery Build a REST API with Node JS and Express | CRUD API Tutorial [Online]. - 2020. - <https://www.youtube.com/watch?v=l8WPWK9mS5M>.

MomentJS MomentJS Document [Online]. - 2021. - <https://momentjs.com/>.

Mongoose Mongoose Document [Online]. - 2021. - <https://mongoosejs.com/docs/guide.html>.

Most Programming with Mosh - Youtube [Online] // How to build a REST API with Node js & Express. - 2018. - <https://www.youtube.com/watch?v=pKd0Rpw7O48>.

Nodemy ExpressJS & NodeJS - Xây dựng hệ thống server chuẩn RESTFUL [Online]. - 2021. -

https://www.youtube.com/playlist?list=PLodO7Gi1F7R1GMefX_44suLAaXnaNYMyC.

Programmers Lab Push notification for Flutter apps using FCM and NodeJS [Online]. -

https://www.youtube.com/playlist?list=PLCk3qj3leTQ0g6m8gDyVwqr_h-qxhJF02.

Tips Javascript How to send emails with Node.js with Nodemailer | GMAIL | OAUTH2 [Online]. - <https://www.youtube.com/watch?v=aqMPRCNzkbq&t=822s>.