

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng ứng dụng giao đồ ăn

HOÀNG THỊ TUYẾT

tuyet.ht184013@sis.hust.edu.vn

Ngành Công nghệ thông tin

Chuyên ngành Công nghệ thông tin

Giảng viên hướng dẫn: ThS. Nguyễn Tiến Thành

Chữ kí GVHD

Khoa: Công nghệ thông tin

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 06/2022

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn sâu sắc đến ThS. Nguyễn Tiến Thành. Thầy là người đã hướng dẫn, giúp đỡ em rất nhiều trong quá trình nghiên cứu, thực hiện và hoàn thiện ĐATN này. Em cũng xin cảm ơn tới tất cả thầy cô đã chỉ dạy cho em những kiến thức quý báu trong suốt quá trình học tập, cung cấp một hành trang thiết thực cho tương lai sắp tới. Cuối cùng, em xin cảm ơn gia đình, bạn bè, đồng nghiệp - những người luôn ở bên cạnh, cổ vũ và động viên tinh thần, giúp em vượt qua những khó khăn trong suốt quá trình học tập tại đại học Bách Khoa Hà Nội. Do thời gian có hạn, cũng như kiến thức chuyên môn còn non kém, ĐATN của em sẽ không tránh khỏi những thiếu sót và hạn chế. Em rất mong nhận được những đánh giá, nhận xét từ thầy cô và các bạn, để từ đây nhận thấy những ưu, nhược điểm của bản thân, đúc kết và xây dựng kinh nghiệm làm việc một cách vững vàng hơn. Em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Công việc kinh doanh buôn bán đã xuất hiện từ rất lâu đời; trải qua mỗi giai đoạn, nó mang một hình thức đặc thù riêng. Trước kia, khi các công cụ hỗ trợ chưa phát triển mạnh, thì việc trao đổi mua bán chỉ diễn ra dưới hình thức mua bán trực tiếp. Từ khi khoa học công nghệ phát triển, nó đã tạo ra các loại hình thức mua bán mới, điển hình là mua bán trực tuyến. Hình thức mua bán trực tuyến đó chính là mô hình thương mại điện tử hiện nay. Nó hỗ trợ đắc lực cho người kinh doanh tiếp xúc được với nhiều khách hàng, còn với khách hàng thì công việc mua bán được diễn ra nhanh chóng, thuận lợi, tiết kiệm thời gian. Với những lợi ích kể trên, thương mại điện tử trở thành một lĩnh vực rất được quan tâm và ngày càng quan trọng cho sự phát triển kinh tế. Sự bùng nổ thương mại điện tử cũng đã đi sâu vào lĩnh vực thức ăn, đồ uống. Hầu hết các giao dịch thương mại điện tử trong lĩnh vực thức ăn, đồ uống được thực hiện thông qua các thiết bị di động như điện thoại và máy tính bảng. Vì vậy, ĐATN này hướng tới xây dựng ứng dụng giao đồ ăn với mong muốn giúp khách hàng có thể trao đổi mua bán một cách dễ dàng, thuận tiện trong thời buổi các dịch bệnh có thể bùng lên như COVID-19.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	4
2.1 Khảo sát hiện trạng	4
2.2 Tổng quan chức năng	5
2.2.1 Biểu đồ use case tổng quát	5
2.2.2 Biểu đồ use case phân rã chức năng Theo dõi đơn hàng	7
2.2.3 Biểu đồ use case phân rã chức năng Quản lý thực đơn	7
2.2.4 Biểu đồ use case phân rã chức năng Quản lý người dùng	8
2.2.5 Biểu đồ use case phân rã chức năng Quản lý quán ăn.....	9
2.2.6 Biểu đồ use case phân rã chức năng Quản lý đơn hàng.....	10
2.2.7 Quy trình nghiệp vụ	11
2.3 Đặc tả chức năng	12
2.3.1 Đặc tả use case Tìm kiếm món ăn.....	12
2.3.2 Đặc tả use case Đặt hàng	12
2.3.3 Đặc tả use case Xem lịch sử đặt hàng	13
2.3.4 Đặc tả use case Chỉnh sửa đơn hàng.....	14
2.3.5 Đặc tả use case Chỉnh sửa thông tin cá nhân	14
2.4 Yêu cầu phi chức năng	15
2.4.1 Yêu cầu UI/UX	15
2.4.2 Yêu cầu thực thi	15

2.4.3 Yêu cầu bảo mật.....	15
2.4.4 Yêu cầu phần mềm.....	15
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	16
3.1 React Native [7].....	16
3.2 Firebase	17
3.3 Ngôn ngữ JavaScript [11]	18
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ	20
4.1 Thiết kế kiến trúc.....	20
4.1.1 Lựa chọn state management	20
4.1.2 Thiết kế tổng quan.....	21
4.1.3 Thiết kế chi tiết gói	22
4.2 Thiết kế chi tiết.....	22
4.2.1 Thiết kế giao diện	22
4.2.2 Thiết kế lớp	25
4.2.3 Thiết kế cơ sở dữ liệu	28
4.3 Xây dựng ứng dụng.....	31
4.3.1 Thư viện và công cụ sử dụng.....	31
4.3.2 Kết quả đạt được	32
4.3.3 Minh họa các chức năng chính	32
4.4 Kiểm thử.....	42
4.4.1 Kiểm thử chức năng Tìm kiếm món ăn.....	42
4.4.2 Kiểm thử chức năng Đặt hàng	43
4.4.3 Kiểm thử chức năng Xem lịch sử đặt hàng.....	43
4.4.4 Kiểm thử chức năng Chỉnh sửa đơn hàng.....	43
4.4.5 Kiểm thử chức năng Chỉnh sửa thông tin cá nhân	44

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT	45
5.1 State Management	45
5.1.1 Vấn đề.....	45
5.1.2 Giải pháp [13].....	45
5.2 Lazy load Flatlist	47
5.2.1 Vấn đề.....	47
5.2.2 Giải pháp	47
5.3 Xây dựng hệ thống đặt hàng	48
5.3.1 Vấn đề.....	48
5.3.2 Giải pháp	48
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	51
6.1 Kết luận	51
6.2 Hướng phát triển.....	51
TÀI LIỆU THAM KHẢO.....	53

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quan	6
Hình 2.2	Biểu đồ usecase phân rã “Theo dõi đơn hàng”	7
Hình 2.3	Biểu đồ usecase phân rã “Quản lý thực đơn”	8
Hình 2.4	Biểu đồ usecase phân rã “Quản lý người dùng”	9
Hình 2.5	Biểu đồ usecase phân rã “Quản lý quán ăn”	10
Hình 2.6	Biểu đồ usecase phân rã “Quản lý đơn hàng”	11
Hình 2.7	Quy trình nghiệp vụ đặt hàng	12
Hình 4.1	Mô hình quản lý state trong react	20
Hình 4.2	Biểu đồ phụ thuộc gói	21
Hình 4.3	Thiết kế gói tham gia use case "Đặt hàng"	22
Hình 4.4	Thiết kế giao diện trang chủ	23
Hình 4.5	Thiết kế giao diện chi tiết quán ăn	24
Hình 4.6	Thiết kế giao diện chi tiết quán ăn	25
Hình 4.7	Thiết kế các lớp chính tham gia chức năng đặt hàng	26
Hình 4.8	Biểu đồ trình tự chức năng đặt hàng	27
Hình 4.9	Biểu đồ trình tự chức năng xác nhận đặt hàng	28
Hình 4.10	Thiết kế cơ sở dữ liệu cho hệ thống giao đồ ăn	28
Hình 4.11	Giao diện màn Trang chủ cho người mua	33
Hình 4.12	Giao diện xem chi tiết quán ăn	34
Hình 4.13	Giao diện màn giỏ hàng cho người mua	35
Hình 4.14	Giao diện thanh toán	36
Hình 4.15	Giao diện theo dõi đơn hàng cho người mua	37
Hình 4.16	Giao diện Chi tiết đơn hàng cho người mua	38
Hình 4.17	Giao diện màn quản lý thông tin cá nhân cho người mua	39
Hình 4.18	Giao diện theo dõi đơn hàng cho người bán	40
Hình 4.19	Giao diện quản lý món ăn cho người bán	41
Hình 4.20	Giao diện thống kê cho người bán	42
Hình 5.1	Màn hình trang chủ	49
Hình 5.2	Màn hình chi tiết quán ăn	50

DANH MỤC BẢNG BIỂU

Bảng 2.1	So sánh ứng dụng ShopeeFood, GrabFood, Go-Food	4
Bảng 2.2	Đặc tả usecase Tìm kiếm món ăn	12
Bảng 2.3	Đặc tả usecase Đặt hàng	13
Bảng 2.4	Đặc tả usecase Xem lịch sử đặt hàng	13
Bảng 2.5	Đặc tả usecase Chỉnh sửa đơn hàng	14
Bảng 2.6	Đặc tả usecase Chỉnh sửa thông tin cá nhân	14
Bảng 4.1	Mô tả chi tiết collection users	29
Bảng 4.2	Mô tả chi tiết collection addresses	29
Bảng 4.3	Mô tả chi tiết collection categories	30
Bảng 4.4	Mô tả chi tiết collection restaurants	30
Bảng 4.5	Mô tả chi tiết collection menus	30
Bảng 4.6	Mô tả chi tiết collection menuItems	30
Bảng 4.7	Mô tả chi tiết collection orders	31
Bảng 4.8	Mô tả chi tiết collection orderItems	31
Bảng 4.9	Danh sách thư viện và công cụ sử dụng	31
Bảng 4.10	Danh sách thư viện sử dụng trong ứng dụng giao đồ ăn	32
Bảng 4.11	Kiểm thử chức năng Tìm kiếm món ăn	42
Bảng 4.12	Kiểm thử chức năng Đặt hàng	43
Bảng 4.13	Kiểm thử chức năng Xem lịch sử đặt hàng	43
Bảng 4.14	Kiểm thử chức năng Chỉnh sửa đơn hàng	43
Bảng 4.15	Kiểm thử chức năng Chỉnh sửa thông tin cá nhân	44

DANH MỤC TỪ VIẾT TẮT

Viết tắt	Tên tiếng Anh	Tên tiếng Việt
API	Application Programming Inter- face	Giao diện lập trình ứng dụng
CSDL		Cơ sở dữ liệu
UC	Use case	Cả sử dụng
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
CNTT		Công nghệ thông tin
ĐATN		Đồ án tốt nghiệp
SV		Sinh viên
CMS	Content Management System	Hệ quản trị nội dung
UI	User Interface	Giao diện người dùng
UX	User Experience	Trải nghiệm người dùng

DANH MỤC THUẬT NGỮ

Thuật ngữ	Ý nghĩa
Shipper	Tài xế
Admin	Quản trị viên
User	Người dùng
App - Application	Ứng dụng
Frontend	Phần tương tác với người dùng
Backend	Những phần bên trong
Framework	Tập hợp các thư viện phần mềm
Freeship	Miễn phí vận chuyển
Online	Trực tuyến
Hybrid	Ứng dụng lai
Multi-platform	Đa nền tảng
Build	Xây dựng
Code	Mã nguồn
Bugs	Lỗi
Cloud	Điện toán đám mây
Service	Dịch vụ
Database	Cơ sở dữ liệu
Objects	Đối tượng
Black-Box	Hộp đen
Data	Dữ liệu
Component	Thành phần
Props	Thuộc tính
Pass	Truyền
Flatlist	Thành phần React Native tạo ra danh sách
Implement	Thực thi
Scroll	Cuộn màn hình

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

Chương này giới thiệu các vấn đề thực tế dẫn tới việc em lựa chọn đề tài, tổng quan về hệ thống giao đồ ăn online. Sau đó nêu ra mục tiêu và phạm vi đề tài, định hướng giải pháp và bố cục của đồ án.

1.1 Đặt vấn đề

Trong thời đại kinh tế ngày càng phát triển theo xu hướng toàn cầu hóa, các thiết bị công nghệ và các sàn thương mại điện tử ngày càng phát triển. Việt Nam cũng đang dần áp dụng các xu hướng đó vào cuộc sống hàng ngày. Nhu cầu về các dịch vụ trực tuyến ngày càng tăng cao, nhất là nhu cầu về dịch vụ ăn uống của con người. Ăn uống là một nhu cầu thiết yếu của mỗi người, họ đòi hỏi những đồ ăn thức uống đa dạng, tiện lợi mà không cần thiết phải đến tận các cửa hàng để mua. Với sự hối hả của thời gian, bận rộn của công việc mà con người dần dần bỏ quên đi căn bếp ở nhà mà quen thuộc dần với các hàng quán. Nhưng sự bận rộn ngày càng bủa vây khiến con người không còn có thời gian để đến các nhà hàng, quán ăn nữa. Thay vì vậy họ mong muốn đồ ăn thức uống có thể giao đến nơi mình muốn mà không cần tự mình đi mua.

Vì vậy mà các app đặt đồ ăn ra đời phục vụ nhu cầu con người. Với những ưu điểm về dịch vụ của mình, các app như ShopeeFood, Grab Food, BAEMIN, Gojek, Loship, ... ngày càng phát triển mạnh mẽ tại Việt Nam và trở thành các đối thủ cạnh tranh lớn của nhau. Tuy phát triển mạnh mẽ là vậy nhưng lòng tin của người tiêu dùng đối với các sản phẩm trên chưa nhiều nguyên nhân vì chất lượng dịch vụ trực tuyến còn thấp.

Từ những nhu cầu thực tế và các hạn chế của những ứng dụng giao đồ ăn trên thị trường, em đã thực hiện đề tài “Xây dựng ứng dụng giao đồ ăn”. Mục đích của đề tài là phát triển hệ thống có thể giúp người mua hàng đặt đồ ăn một cách dễ dàng và nhanh chóng nhất, đồng thời cho phép người bán theo dõi đơn hàng và thống kê báo cáo doanh thu của cửa hàng một cách chính xác.

1.2 Mục tiêu và phạm vi đề tài

Với những vấn đề đã trình bày trong mục 1.1, trong đồ án này, em đặt ra mục tiêu là phát triển một hệ thống có các chức năng quản lý nhà hàng và việc đặt đơn hàng. Thứ nhất, hệ thống có chức năng cho phép người mua hàng có thể tìm kiếm nhà hàng, món ăn sau đó tạo đơn hàng và tiến hành đặt hàng. Đơn hàng khi được đặt sẽ ở trạng thái chờ xác nhận và người mua hàng có thể theo dõi tình trạng đơn của mình. Người mua có thể hủy đơn nếu đơn hàng vẫn còn ở trạng thái chưa

giao hàng. Thứ hai, người bán là chủ của quán ăn sẽ có chức năng quản lý đơn hàng của quán. Người bán được phép cập nhật tình trạng đơn hàng như xác nhận đơn hàng, hủy đơn hàng. Người bán cũng có các chức năng để quản lý nhà hàng như chỉnh sửa thực đơn của quán. Thứ ba, hệ thống cung cấp chức năng thống kê doanh thu theo các mốc thời gian cụ thể giúp người bán quản lý được tình hình của quán. Cuối cùng, hệ thống có chức năng cho phép quản trị viên theo dõi thông tin của người dùng, báo cáo vi phạm và quản lý việc đặt hàng.

Để có thể phát triển một hệ thống đáp ứng các chức năng trên, hệ thống cần được thiết kế với một kiến trúc đáp ứng được nhiều người sử dụng cùng lúc, hiệu năng ổn định và dễ dàng mở rộng. Hơn nữa, giao diện sử dụng dành cho người dùng cũng phải được thiết kế tinh gọn, tối giản mà lại dễ sử dụng, không gây nhiều khó khăn với những người mới tiếp cận, tăng trải nghiệm cho người dùng. Ngoài ra, hệ thống cũng cần đáp ứng được các yêu cầu về kỹ thuật, chẳng hạn như tìm kiếm và sửa lỗi, bảo trì và thêm mới chức năng.

1.3 Định hướng giải pháp

Với mục tiêu phát triển một hệ thống gồm nhiều các hệ thống con tương ứng từng tác nhân sử dụng, ĐATN sẽ cố gắng xây dựng các tính năng cần có. Tuy nhiên, do thời gian của đề án có hạn nên ĐATN sẽ xây dựng ứng dụng dành cho người mua, người bán trên nền tảng mobile trước, sau này có thời gian sẽ phát triển hệ thống CMS trên nền tảng website.

Mã nguồn của ứng dụng sẽ chia làm hai phần: backend – xử lý các nghiệp vụ logic, kết nối với dữ liệu để thực hiện các tính năng; và frontend – hiển thị dữ liệu, tương tác với người dùng.

Phần backend sẽ sử dụng Firebase [1] – là một dịch vụ backend của Google cung cấp các API đơn giản. Với ưu điểm dễ dàng sử dụng giúp chúng ta trong việc phát triển ứng dụng một cách rất nhanh chóng.

Với phần frontend, để đảm bảo tốc độ phát triển của dự án và chạy được trên cả hai nền tảng iOS và Android, framework React-Native và nền tảng ngôn ngữ lập trình JavaScript được lựa chọn. Với ưu điểm phát triển dễ dàng, nhanh chóng và chạy được trên cả 2 nền tảng iOS và Android mà không cần cấu hình nhiều, framework React-Native sẽ giúp phát triển ứng dụng một cách nhanh chóng.

1.4 Bố cục đề án

Phần còn lại của báo cáo đề án tốt nghiệp này được tổ chức như sau.

Chương 2 sẽ trình bày về tổng quan các chức năng trong hệ thống đồng thời đặc tả một số use case chính.

Trong Chương 3, em sẽ trình bày về các công nghệ và công cụ được sử dụng xuyên suốt quá trình thực hiện đề án.

Trong Chương 4, em sẽ trình bày chi tiết về phân tích thiết kế hệ thống và trình bày sâu hơn về luồng hoạt động của một số nghiệp vụ chính trong hệ thống, cách thiết kế giao diện, cơ sở dữ liệu và cuối cùng là triển khai hệ thống.

Trong Chương 5 em sẽ trình bày nội dung về các giải pháp và đóng góp mà em thấy tự hào nhất, đồng thời trình bày chi tiết hơn về cách thực hiện các yêu cầu nêu ra trong Chương 2 cũng như cách giải quyết các khó khăn trong quá trình thực hiện đề án.

Trong Chương 6 em sẽ kết luận đề án, sau đó trình bày các ưu nhược điểm của hệ thống, những gì em đã và chưa làm được trong đề án tốt nghiệp lần này. Đồng thời, em sẽ đề xuất các định hướng mà em muốn phát triển hệ thống trong tương lai.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Chương 2 trình bày một số đánh giá về các sản phẩm ứng dụng giao đồ ăn hiện có trên thị trường. Qua đó, đưa ra tổng quan các chức năng em sẽ thực hiện trong hệ thống của mình và sau đó là làm rõ từng chức năng.

2.1 Khảo sát hiện trạng

Với mức độ tăng trưởng nhanh và giá trị thị trường lớn, theo báo cáo của hãng nghiên cứu thị trường Euromonitor, thị trường giao nhận đồ ăn Việt Nam tiếp tục cạnh tranh khốc liệt trong thời gian tới do cơ hội phía trước là rất nhiều. Năm 2020, lĩnh vực này sẽ đạt giá trị khoảng 38 triệu USD, với tốc độ tăng trưởng trung bình khoảng 11%/năm [2]. Và tại Việt Nam những thương hiệu lớn trong lĩnh vực này không thể kể đến các cái tên như sau: Grab với dịch vụ GrabFood, Go-Viet với dịch vụ Go-Food, Shopee với dịch vụ ShopeeFood và Baemin thương hiệu mới nổi đến từ Hàn Quốc. Sau đây em sẽ phân tích các ưu nhược điểm của từng sản phẩm.[3]

Bảng 2.1 thể hiện so sánh theo các tiêu chí của 3 ứng dụng ShopeeFood, GrabFood, Go-Food

Tiêu chí	ShopeeFood	GrabFood	Go-Food
Số lượng quán ăn	Nhiều	Bình thường	Nhiều
Thời gian giao hàng	Nhanh	Nhanh	Bình thường
Độ phủ sóng	16 Tỉnh thành	15 Tỉnh thành	2 Tỉnh thành
Chăm sóc khách hàng	Bình thường	Tốt	Bình thường
Phí ship tối thiểu	15.000đ/5km	15.000đ/5km	14.000đ/5km
Giờ hoạt động	24/7	24/7	24/7
Thanh toán thẻ/Ví điện tử	Có	Có	Không
Hình ảnh món ăn	Có	Có	Không
Đọc review tham khảo	Có	Không	Không
Đặt đơn/ chat nhóm	Có	Không	Không
Tích điểm đổi quà	Không	Có	Không

Bảng 2.1: So sánh ứng dụng ShopeeFood, GrabFood, Go-Food

Từng được biết đến là ứng dụng Now, nay đã đổi tên thành ShopeeFood [4], vẫn luôn được lòng rất nhiều người dùng. Với cơ sở dữ liệu đa dạng cùng nhiều đối tác quán ăn, người dùng sẽ dễ dàng tìm được món ăn yêu thích. ShopeeFood liên tục có mã freeship, mã đối tác hấp dẫn giúp loại bỏ yếu tố khiến người dùng e dè khi đặt mua online. Mạng lưới tài xế rộng khắp các phố phường cũng giúp thời gian giao hàng được nhanh chóng hơn nữa. Tuy nhiên, chức năng thanh toán trên ShopeeFood vẫn còn nhận nhiều phản hồi tiêu cực vì khi đặt đơn ứng dụng sẽ trừ

tiền ngay, nên lúc quán hết món hay phát sinh thêm phí thì không thể chỉnh sửa lại giá, và phải đợi một khoảng thời gian để được hoàn tiền vào thẻ.

Trong trận chiến giao đồ ăn vài năm trước, hầu như người dùng không có lựa chọn nào khác ngoài Now và chấp nhận tình trạng 1 đơn chờ đợi 1-2 tiếng nhưng với sự xuất hiện của Grab, thế cục trận chiến đã được thay đổi. Trên cơ sở xây dựng mạng lưới tài xế công nghệ, ứng dụng Grab ngày càng mở rộng hơn cho các dịch vụ đi chợ hộ, vận chuyển hàng hộ và trong đó có đặt mua thức ăn hộ. Dù hoạt động ở nhiều mảng nhưng GrabFood cũng hoàn thành rất tốt các nhu cầu của người dùng. Tuy nhiên, có vẻ các chương trình ưu đãi và mã giảm giá của Grab không thực sự quá đa dạng và yêu cầu nhiều điều kiện đi kèm.

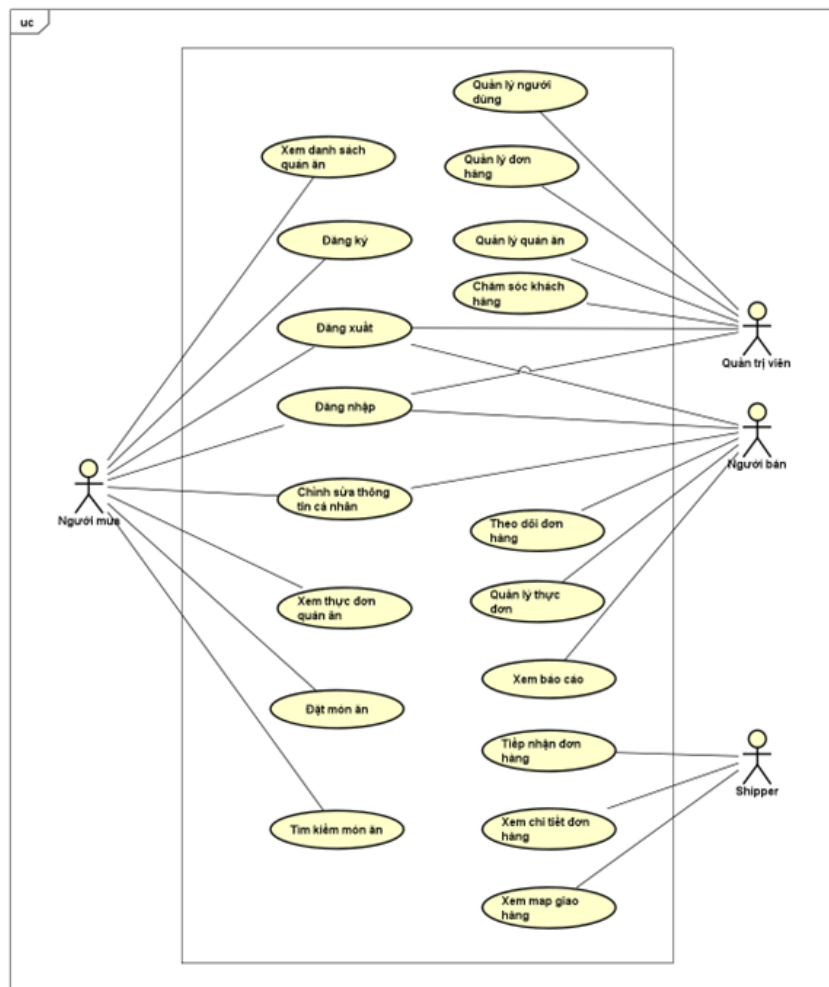
Tương tự như GrabFood [5], GoFood [6] cũng cung cấp danh sách gồm rất nhiều hàng quán đối tác để khách hàng lựa chọn. Mạng lưới tài xế cũng gọi là nhiều khi so sánh với các nền tảng khác. GoFood trực thuộc Go-Viet công ty con của tập đoàn GoJek, từng được kỳ vọng sẽ trở thành đối thủ nặng ký trong trận chiến, nhưng sau hơn 1 năm với nhiều sự thay đổi về nhân sự vẫn dậm chân tại chỗ, dần dần tụt hạng trong trận chiến giành ngôi vị dẫn đầu.

Từ những luận điểm trên, em sẽ xây dựng ứng dụng giao đồ ăn phát huy được các ưu điểm đồng thời hạn chế và khắc phục những nhược điểm của các sản phẩm đang có trên thị trường. Cụ thể các chức năng đó em sẽ trình bày trong các phần tiếp theo.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quát

Hệ thống giao đồ ăn gồm nhiều thành phần chức năng khác nhau. Hình 2.1 mô tả tổng quan toàn bộ chức năng của hệ thống.



Hình 2.1: Biểu đồ use case tổng quan

Hệ thống gồm có 4 tác nhân là người mua, người bán, shipper, quản trị viên.

Người mua cần đăng nhập để vào ứng dụng. Khi đăng nhập xong người mua có thể đặt đồ ăn bằng các tìm kiếm đồ ăn, chọn cửa hàng yêu thích, xem đồ ăn, đánh giá và nêu cảm nhận về món ăn đó. Người mua cũng có thể sửa thông tin của mình qua ứng dụng khi vào mục tài khoản.

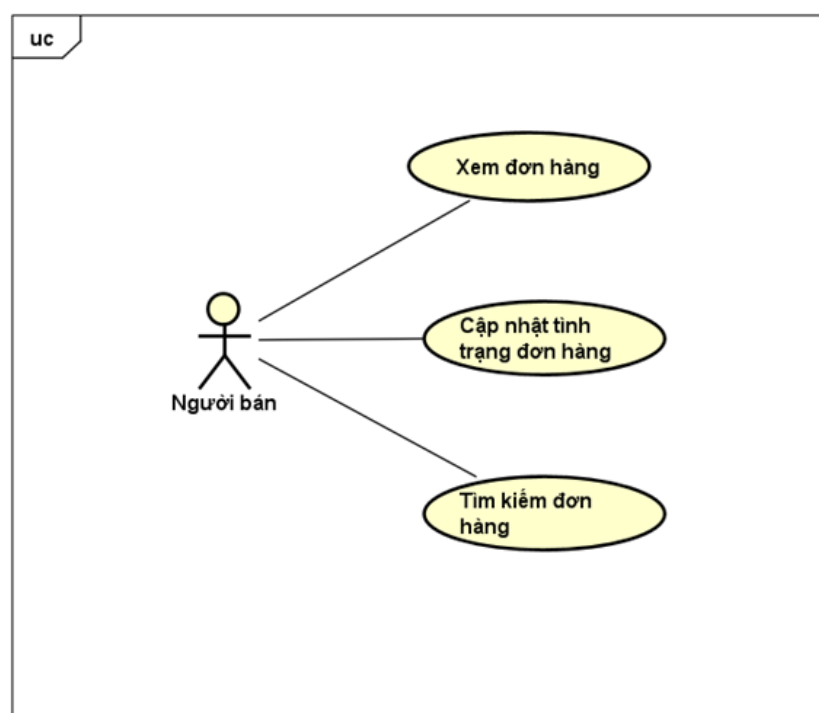
Người bán sẽ có chức năng theo dõi đơn hàng, quản lý thực đơn, xem báo cáo giúp cho người bán biết được công việc doanh thu và lợi nhuận.

Shipper có chức năng tiếp nhận đơn hàng và xác nhận nhận đơn. Để giao đơn hàng, hệ thống hỗ trợ bản đồ để giúp shipper giao hàng đến người mua một cách dễ dàng.

Quản trị viên sẽ có chức năng quản lý người dùng, quản lý đơn hàng, quản lý quán ăn và chăm sóc khách hàng.

2.2.2 Biểu đồ use case phân rã chức năng Theo dõi đơn hàng

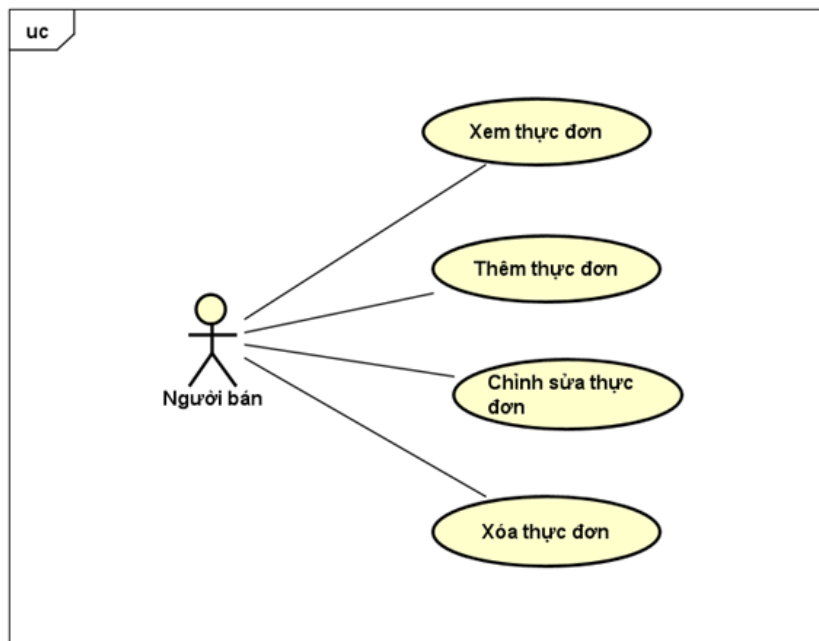
Usecase Theo dõi đơn hàng được thực hiện bởi người bán, bao gồm xem đơn hàng, cập nhật tình trạng đơn hàng và tìm kiếm đơn hàng. Chi tiết usecase được miêu tả qua Hình 2.2.



Hình 2.2: Biểu đồ usecase phân rã “Theo dõi đơn hàng”

2.2.3 Biểu đồ use case phân rã chức năng Quản lý thực đơn

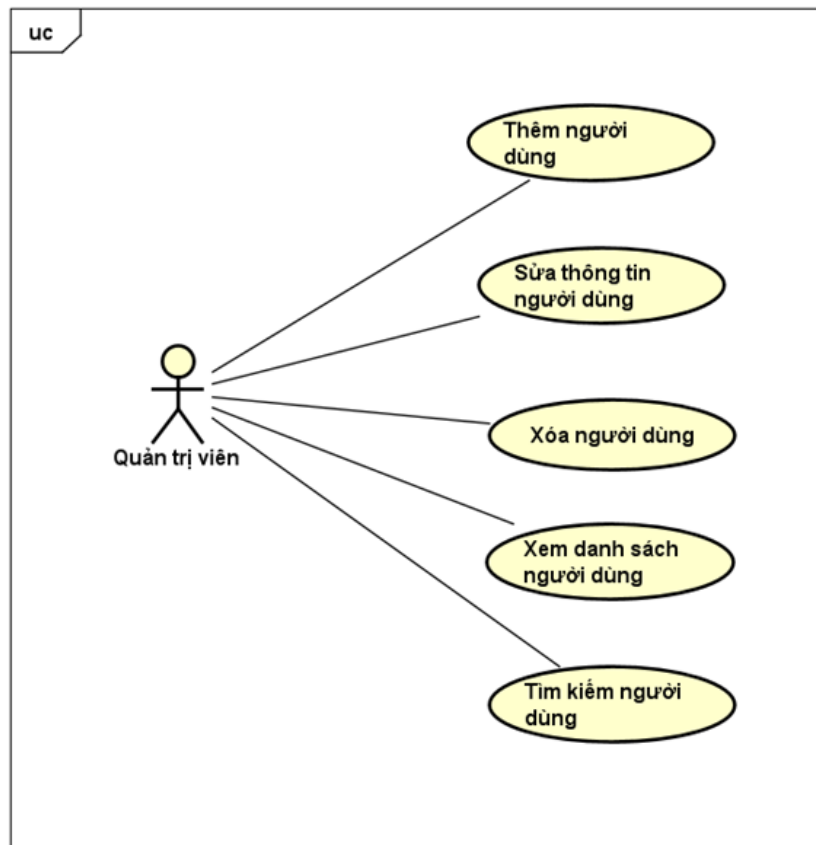
Usecase Quản lý thực đơn được thực hiện bởi người bán, bao gồm xem thực đơn, thêm thực đơn, chỉnh sửa thực đơn và xóa thực đơn. Chi tiết usecase được miêu tả qua Hình 2.3.



Hình 2.3: Biểu đồ usecase phân rã “Quản lý thực đơn”

2.2.4 Biểu đồ use case phân rã chức năng Quản lý người dùng

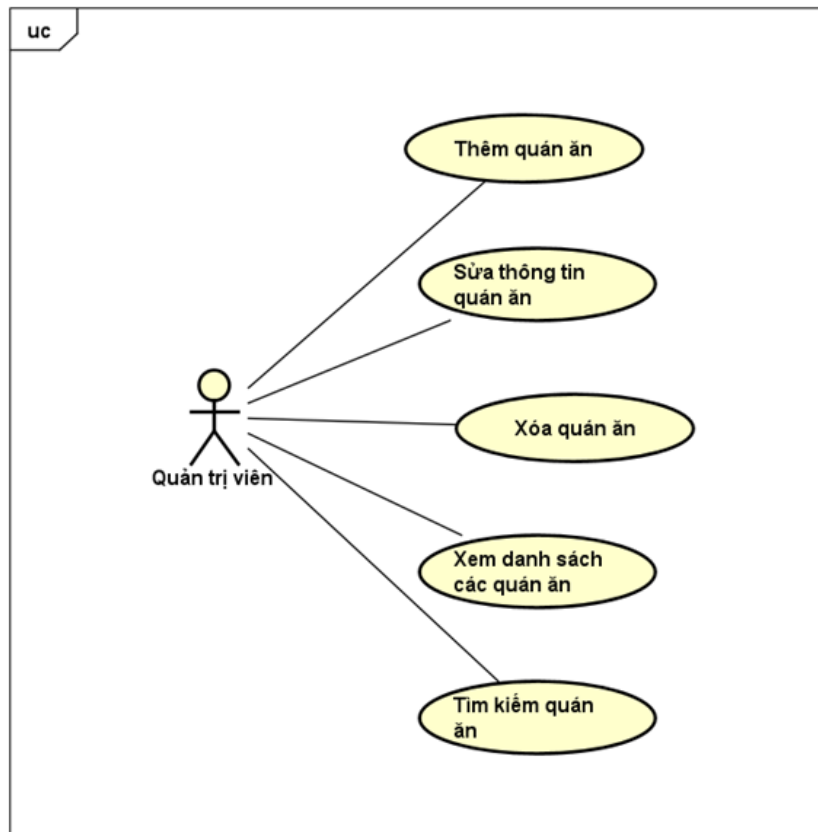
Usecase Quản lý người dùng được thực hiện bởi quản trị viên, bao gồm thêm người dùng, sửa thông tin người dùng, xóa người dùng, tìm kiếm và xem danh sách người dùng. Chi tiết usecase được miêu tả qua Hình 2.4.



Hình 2.4: Biểu đồ usecase phân rã “Quản lý người dùng”

2.2.5 Biểu đồ use case phân rã chức năng Quản lý quán ăn

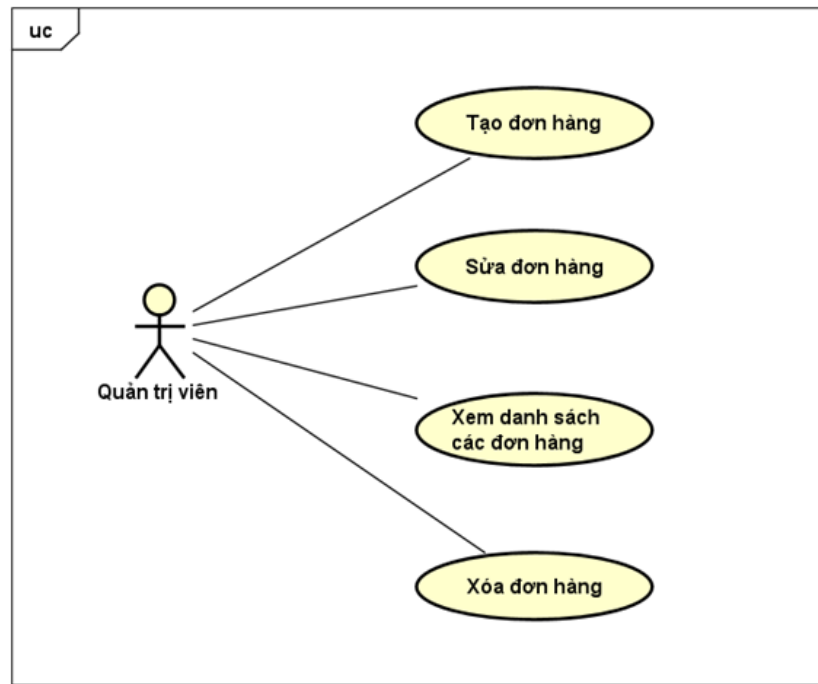
Usecase Quản lý quán ăn được thực hiện bởi quản trị viên, bao gồm thêm quán ăn, sửa thông tin quán ăn, xóa quán ăn, tìm kiếm và xem danh sách quán ăn. Chi tiết usecase được miêu tả qua Hình 2.5.



Hình 2.5: Biểu đồ usecase phân rã “Quản lý quán ăn”

2.2.6 Biểu đồ use case phân rã chức năng Quản lý đơn hàng

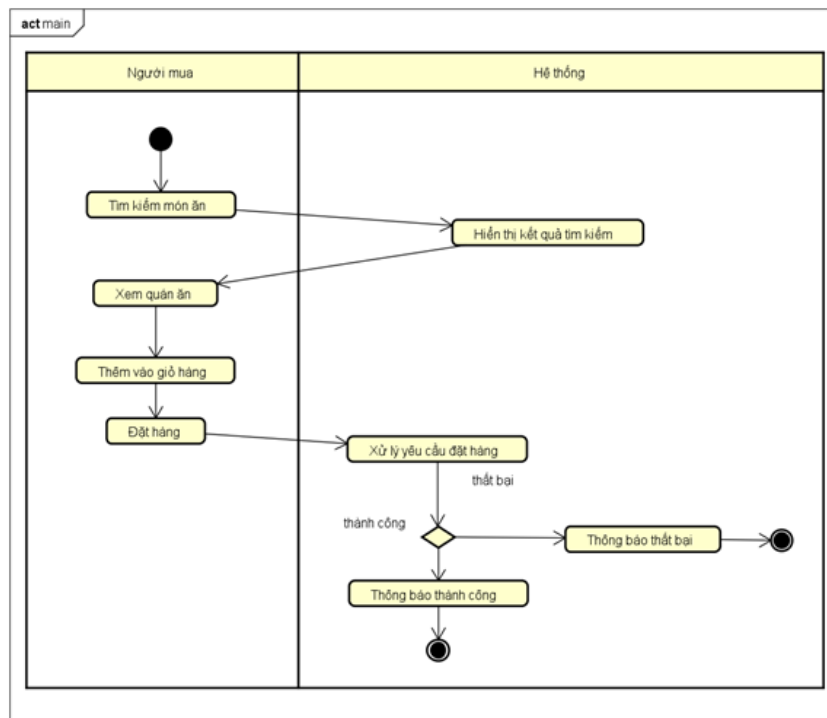
Usecase Quản lý đơn hàng được thực hiện bởi quản trị viên, bao gồm tạo đơn hàng, sửa đơn hàng, xem danh sách các đơn hàng và xóa đơn hàng. Chi tiết usecase được miêu tả qua Hình 2.6.



Hình 2.6: Biểu đồ usecase phân rã “Quản lý đơn hàng”

2.2.7 Quy trình nghiệp vụ

Hình 2.7 mô tả quy trình nghiệp vụ của chức năng đặt hàng. Khách hàng sau khi đăng nhập vào ứng dụng có thể tìm kiếm món ăn trong danh sách các món có trong hệ thống cửa hàng. Khi đã chọn được quán, khách chọn món để thêm vào giỏ hàng. Sau khi hoàn tất, khách hàng ấn đặt hàng, hệ thống chuyển đến trang thanh toán. Khách hàng ấn thanh toán để hoàn thành việc đặt hàng. Đơn hàng đặt thành công sẽ hiện ở trang đơn hàng của tôi. Khách hàng có thể xem chi tiết đơn hàng sau khi đã đặt hàng.



Hình 2.7: Quy trình nghiệp vụ đặt hàng

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case Tìm kiếm món ăn

Bảng 2.2: Đặc tả usecase Tìm kiếm món ăn

Mã Use case	UC001	Tên Use case	Tìm kiếm món ăn
Tác nhân hệ thống	Người mua		
Tiền điều kiện	Đăng nhập		
Luồng sự kiện chính	STT	Thực hiện	Hành động
	1	Người mua	Nhập món ăn muốn tìm vào ô tìm kiếm
	2	Hệ thống	Xử lý thông tin và trả về kết quả tìm kiếm
	3	Hệ thống	Hiển thị thông tin và kết quả tìm kiếm
Luồng sự thay thế	STT	Thực hiện	Hành động
	3a	Hệ thống	Thông báo lỗi trong quá trình tìm kiếm
Hậu điều kiện	Không		

2.3.2 Đặc tả use case Đặt hàng

Bảng 2.3: Đặc tả usecase Đặt hàng

Mã Use case	UC002	Tên Use case	Đặt hàng
Tác nhân hệ thống	Người mua		
Tiền điều kiện	Người mua đã tạo giỏ hàng		
Luồng sự kiện chính	STT	Thực hiện	Hành động
	1	Người mua	Ấn nút đặt hàng
	2	Hệ thống	Chuyển thông tin hàng sang màn hình đặt hàng
	3	Người mua	Kiểm tra đơn hàng, tiền, số lượng hàng và chọn thanh toán
	4	Hệ thống	Chuyển sang màn thanh toán
	5	Người mua	Kiểm tra thông tin
	6	Hệ thống	Thông báo đặt hàng thành công
Luồng sự thay thế	STT	Thực hiện	Hành động
	6a	Hệ thống	Thông báo lỗi quá trình đặt hàng
Hậu điều kiện	Không		

2.3.3 Đặc tả use case Xem lịch sử đặt hàng

Bảng 2.4: Đặc tả usecase Xem lịch sử đặt hàng

Mã Use case	UC003	Tên Use case	Xem lịch sử đặt hàng
Tác nhân hệ thống	Người mua		
Tiền điều kiện	Người mua đã tạo giỏ hàng		
Luồng sự kiện chính	STT	Thực hiện	Hành động
	1	Người mua	Chọn vào mục lịch sử
	2	Hệ thống	Chuyển sang màn lịch sử cho khách hàng
	3	Hệ thống	Hiện thị đơn hàng đã đặt cho khách hàng
	4	Người mua	Chọn vào đơn hàng muốn xem lại
	5	Hệ thống	Trả về thông tin đơn hàng muốn xem
Luồng sự thay thế	Không		

Hậu điều kiện	Không
----------------------	-------

2.3.4 Đặc tả use case **Chỉnh sửa đơn hàng**

Bảng 2.5: Đặc tả usecase Chỉnh sửa đơn hàng

Mã Use case	UC004	Tên Use case	Chỉnh sửa đơn hàng
Tác nhân hệ thống	Người bán		
Tiền điều kiện	Không		
Luồng sự kiện chính	STT	Thực hiện	Hành động
	1	Người bán	Chọn xem chi tiết đơn hàng
	2	Hệ thống	Chuyển sang màn chi tiết đơn hàng
	3	Người bán	Ấn chọn trạng thái đơn hàng là xác nhận đơn
	4	Người bán	Ấn nút cập nhật đơn hàng
	5	Hệ thống	Thông báo cập nhật thành công
Luồng sự thay thế	STT	Thực hiện	Hành động
	5a	Hệ thống	Thông báo lỗi không sửa được đơn hàng
Hậu điều kiện	Không		

2.3.5 Đặc tả use case **Chỉnh sửa thông tin cá nhân**

Bảng 2.6: Đặc tả usecase Chỉnh sửa thông tin cá nhân

Mã Use case	UC005	Tên Use case	Chỉnh sửa thông tin cá nhân
Tác nhân hệ thống	Người mua		
Tiền điều kiện	Không		

Luồng sự kiện chính	STT	Thực hiện	Hành động
	1	Người mua	Chọn vào mục thông tin cá nhân
	2	Hệ thống	Chuyển sang màn thông tin cá nhân
	3	Người mua	Nhập các thông tin thay đổi
	4	Người mua	Ấn xác nhận thay đổi
	5	Hệ thống	Xử lý và trả về thông báo thay đổi thông tin thành công
Luồng sự thay thế	STT	Thực hiện	Hành động
	5a	Hệ thống	Thông báo lỗi không sửa được thông tin cá nhân
Hậu điều kiện	Không		

2.4 Yêu cầu phi chức năng

2.4.1 Yêu cầu UI/UX

- Giao diện đẹp, đơn giản và thân thiện với người dùng.
- Thông báo sau khi thực hiện các thao tác phải rõ ràng, phân biệt giữa các tính năng.
- Đảm bảo tính dễ học và dễ sử dụng cho người mới bắt đầu.

2.4.2 Yêu cầu thực thi

- Đảm bảo hiệu năng khi có nhiều người sử dụng cùng lúc.
- Chạy ổn định 24/7.

2.4.3 Yêu cầu bảo mật

- Hệ thống bảo mật thông tin người dùng.
- Tránh được các cuộc tấn công từ tội phạm mạng.

2.4.4 Yêu cầu phần mềm

- Dữ liệu được cập nhật liên tục.
- Khả năng tương tác với người dùng tốt.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 React Native [7]

React Native là một Framework được các chuyên gia của Facebook phát triển, nó xuất phát từ nhu cầu cần phát triển những ứng dụng, cũng như chức năng có thể chạy được trên cả 2 nền tảng hệ điều hành Android và iOS một cách mượt mà nhất có thể.

Và một mục đích chính nữa là để giải quyết bài toán về hiệu năng của Hybrid, cũng như chi phí để phát triển ứng dụng trong dài hạn.

Với React Native thì các lập trình viên có thể build ra được những ứng dụng đa nền tảng (multi-platform) chứ không chỉ đơn thuần là ứng dụng dành riêng cho iOS hay Android, không phải là một Mobile Web app, không phải là HTML5 [8] app và cũng không phải là một Hybrid app.

Trước kia, để ứng dụng có thể chạy được trên 2 nền tảng thì lập trình viên phải viết ra hai bộ code riêng biệt cho mỗi hệ điều hành tương ứng, nhưng với React Native thì điều này không còn cần thiết nữa.

Facebook đã chính thức tung ra phiên bản cho người dùng là các lập trình viên (dev) trên toàn thế giới để họ chung tay phát triển.

Với React Native, lập trình viên có thể sử dụng JavaScript (hoặc có thể kết hợp với code native như Swift [9], Java [10]) để tạo ra những ứng dụng chạy được trên cả Android và iOS mà không cần phải thiết kế riêng lẻ nữa.

Ưu điểm:

- Khi sử dụng React Native thì sẽ ít phải sử dụng native code hơn.
- Ta có thể phát triển ứng dụng nhanh chóng hơn.
- Tiết kiệm nhiều chi phí và nhân lực cho các doanh nghiệp.
- Khả năng tái tạo code cao lên tới (75% - 80%).
- Duy trì ít code hơn, ít bugs hơn.
- Tận dụng nguồn nhân lực tốt hơn.
- Có cộng đồng người sử dụng rộng rãi dễ dàng tìm được sự giúp đỡ khi gặp phải bugs.
- Khi làm thì không cần trang bị quá nhiều kiến thức nền.
- Trải nghiệm người dùng rất tốt.

Nhược điểm:

- React Native chỉ hỗ trợ cho 2 nền tảng iOS và Android.
- Hiệu năng sẽ bị giảm đi thấp hơn so với ứng dụng thuần native code.
- Bảo mật của ứng dụng không cao.
- Có một số thư viện thư mục cần phải trả phí mới được trải nghiệm sử dụng.
- Khả năng tùy biến không thực sự tốt và nhanh nhạy với một vài module.

3.2 Firebase

Firebase là nền tảng được xây dựng, cung cấp bởi Google, hoạt động trên nền tảng Cloud nhằm hỗ trợ phát triển các ứng dụng trên web và di động dựa trên nguyên tắc đơn giản hóa thao tác lập trình. Các CSDL có sẵn mang đến khả năng code nhanh chóng, thuận tiện hơn. Dưới sự hỗ trợ của nền tảng này, lập trình viên hoàn toàn không cần để tâm đến backend hay server mà có cơ hội tập trung chủ yếu vào việc phát triển ứng dụng. Nền tảng này hướng đến đối tượng chủ yếu là di động, được biết đến như một backend as a service.

Firebase được sử dụng rộng rãi không phải điều ngẫu nhiên, bởi nó sở hữu rất nhiều ưu điểm tuyệt vời. Cụ thể là:

- Tốc độ cao: Firebase hỗ trợ phát triển ứng dụng với tốc độ nhanh chóng, giảm bớt thời gian phát triển và tiếp thị ứng dụng.
- Đơn giản, dễ dàng trong sử dụng: Người dùng có thể tạo tài khoản Firebase bằng tài khoản Google và cũng như sử dụng nền tảng này trong phát triển ứng dụng một cách dễ dàng.
- Một nền tảng, đa dịch vụ: Firebase cung cấp đầy đủ các dịch vụ hỗ trợ phát triển web, bạn có thể chọn database Firestore hoặc Realtime theo ý muốn.
- Là một phần của Google: Firebase đã được mua lại và hiện tại đang là một phần của Google. Nó khai thác triệt để sức mạnh và các dịch vụ sẵn có của Google.
- Tập trung phát triển chủ yếu về giao diện người dùng: Firebase cho phép lập trình viên tập trung chủ yếu vào phát triển phần giao diện người dùng nhờ kho Backend mẫu phong phú đa dạng.
- Không có máy chủ: Điều này giúp Firebase có khả năng tối ưu hóa về hiệu suất làm việc thông qua mở rộng cụm database.
- Học máy: Firebase cung cấp thư viện học máy cho lập trình viên, hỗ trợ tối đa cho việc phát triển ứng dụng

- Tạo ra lưu lượng truy cập: Với khả năng hỗ trợ tạo lập các chỉ mục, Firebase giúp nâng cao thứ hạng của ứng dụng trên bảng xếp hạng Google, từ đó tăng lượt traffic.
- Theo dõi lỗi: Firebase là công cụ phát triển, khắc phục lỗi tuyệt vời.
- Sao lưu: Firebase sao lưu thường xuyên, giúp đảm bảo tính sẵn có và bảo mật thông tin cho dữ liệu

Tuy nhiên, Firebase vẫn có nhược điểm như sau:

- Không phải mã nguồn mở
- Chỉ hoạt động trên CSDL NoSQL
- Lập trình viên không được truy cập mã nguồn
- Truy vấn chậm
- Phạm vi hoạt động bị hạn chế ở một vài quốc gia
- Một vài dịch vụ cần trả phí, chi phí dành cho Firebase tương đối lớn và không ổn định
- Thiếu Dedicated Servers và cả hợp đồng doanh nghiệp
- Không cung cấp API GraphQL
- Chỉ có thể chạy trên Google Cloud.

3.3 Ngôn ngữ JavaScript [11]

JavaScript là một ngôn ngữ lập trình website, được tích hợp và nhúng trong HTML giúp website sống động hơn. JavaScript cho phép kiểm soát các hành vi của trang web tốt hơn so với khi chỉ sử dụng mỗi HTML.

JavaScript được dùng trong rất nhiều lĩnh vực khác nhau:

- Lập trình website.
- Xây dựng ứng dụng cho máy chủ.
- Làm ra ứng dụng di động (React Native), trò chơi.

JavaScript có rất nhiều ưu điểm vượt trội hơn so với các đối thủ khác, chẳng hạn như:

- Nó được coi là ngôn ngữ lập trình quan trọng để xây dựng giao diện người dùng của các trang web
- Chạy ngay lập tức trong trình duyệt web vì không có yêu cầu biên dịch
- Cung cấp một giao diện phong phú hơn cho một trang web

- Rất linh hoạt
- Thực hiện đơn giản, dễ dàng phát hiện và sửa lỗi
- Rất nhiều tài nguyên và sự hỗ trợ của cộng đồng cực lớn
- Được sử dụng để xây dựng một loạt các ứng dụng
- Hoạt động đặc biệt tốt với các ngôn ngữ lập trình khác

Một số khuyết điểm của JavaScript có thể kể đến như:

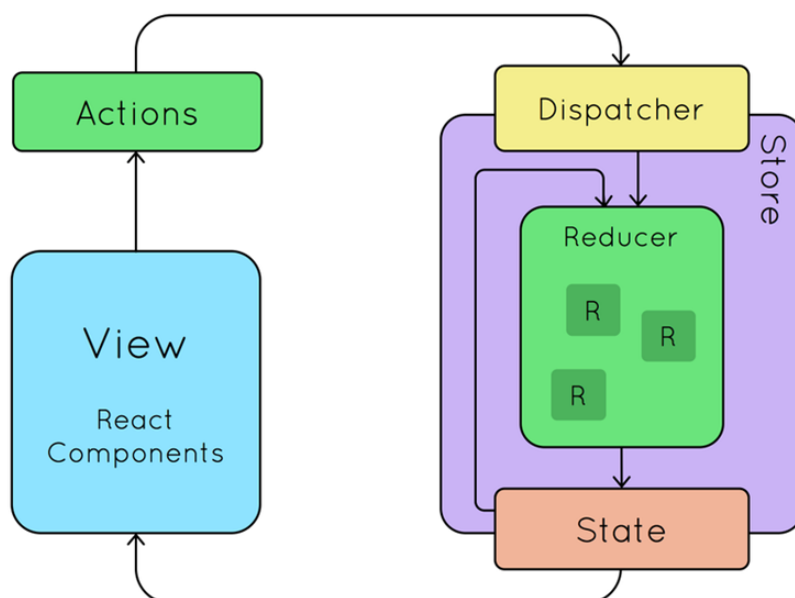
- Không có bản sao hoặc phương pháp tương đương
- Chỉ cho phép đơn thừa kế(không kế đa kế thừa được)
- Khi code thực thi trên máy người dùng, nhiều người chọn tắt JavaScript do sợ bị khai thác cho mục đích xấu
- Có thể được thông dịch khác nhau bởi các trình duyệt khác nhau

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn state management

Redux [12] là 1 thư viện JavaScript để quản lý state của ứng dụng, thường được sử dụng với JavaScript framework như React. Cơ chế hoạt động của nó được tóm gọn trong hình 4.1



Hình 4.1: Mô hình quản lý state trong react

Redux sẽ hoạt động dựa vào 3 thành phần cơ bản là: Actions, Reducers và Store.

Actions đơn giản là các events. Chúng là cách mà chúng ta send data từ app đến Redux store. Những data này có thể là từ sự tương tác của user vs app, API calls hoặc cũng có thể là từ form submission.

Reducers là các function nguyên thủy chúng lấy state hiện tại của app, thực hiện một action và trả về một state mới. Những states này được lưu như những objects và chúng định rõ cách state của một ứng dụng thay đổi trong việc phản hồi một action được gửi đến store.

Store lưu trạng thái ứng dụng và nó là duy nhất trong bất kỳ một ứng dụng Redux nào. Bạn có thể access các state được lưu, update state, và đăng ký or hủy đăng ký các listeners thông qua helper methods.

Redux được xây dựng dựa trên 3 nguyên lý như sau:

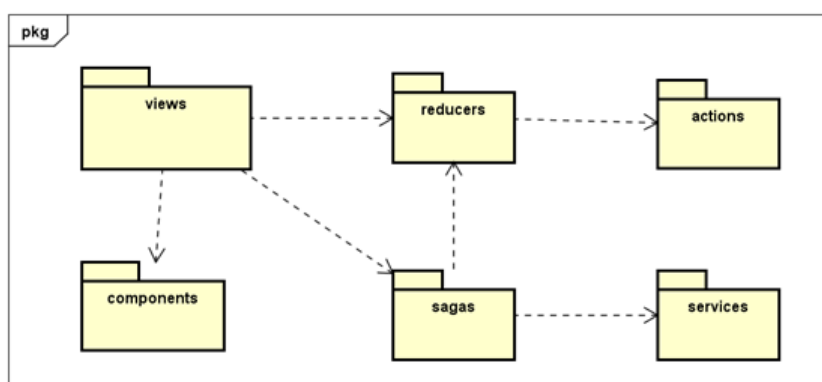
- Trạng thái (state) của toàn bộ ứng dụng được lưu trong 1 store duy nhất

là 1 Object mô hình tree.

- Chỉ có 1 cách duy nhất để thay đổi trạng thái (state) đó là tạo ra một action (là 1 object mô tả những gì xảy ra).
- Sử dụng hàm thuần túy để thay đổi. Với mục đích chỉ ra cách State được biến đổi từ Action. Vì thế, chúng ta cần sử dụng các pure function gọi là Reducer.

4.1.2 Thiết kế tổng quan

Sự phụ thuộc gói của ứng dụng được thể hiện qua Hình 4.2:



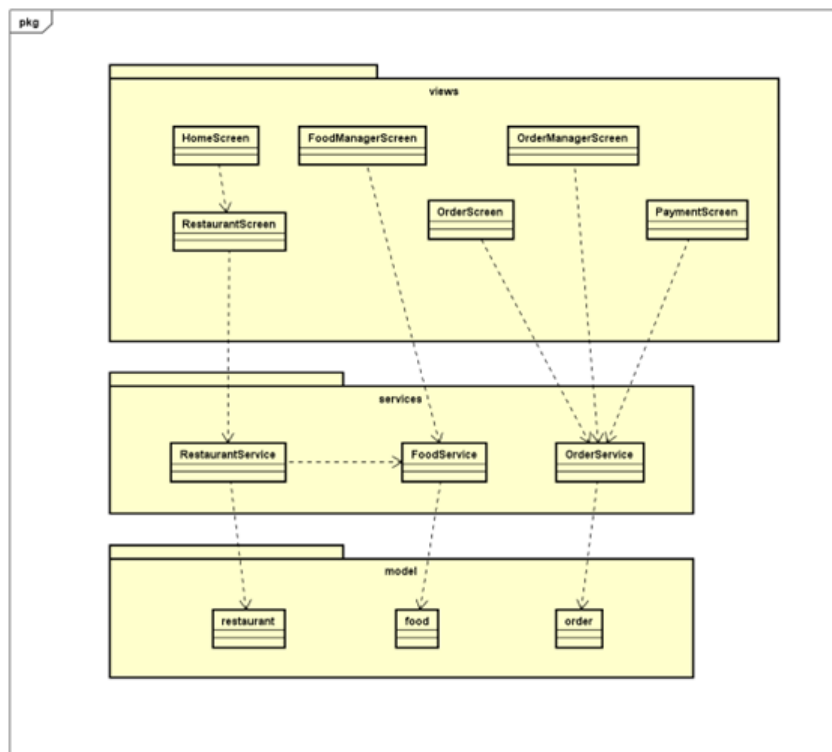
Hình 4.2: Biểu đồ phụ thuộc gói

Trong đó:

- Gói views: là nơi lưu trữ các template để hiển thị dữ liệu. Dữ liệu hiển thị được nhận từ API hoặc nhận trực tiếp từ các controller.
- Gói components: là nơi chứa các component view, có khả năng tái sử dụng.
- Gói sagas: là nơi chứa các hàm saga có chức năng nhận các action và xử lý trước khi chạy đến các hàm trong reducers.
- Gói reducers: là nơi chứa các function nguyên thủy chúng lấy state hiện tại của app, thực hiện một action và trả về một state mới. Những states này được lưu như những objects và chúng định rõ cách state của một ứng dụng thay đổi trong việc phản hồi một action được gửi đến store.
- Gói actions: là nơi chứa các events. Chúng là cách mà chúng ta send data từ app đến Redux store. Những data này có thể là từ sự tương tác của user vs app, API calls hoặc cũng có thể là từ form submission.
- Gói services: là nơi chứa các hàm để tương tác với firebase, xử lý các logic nghiệp vụ.

4.1.3 Thiết kế chi tiết gói

Hình 4.3 thể hiện thiết kế chi tiết gói cho nghiệp vụ đặt hàng. Có 3 gói là views, services, models, trong đó views là package chứa các lớp bao có chức năng tương tác với người dùng như: màn quán ăn, màn thanh toán, màn lịch sử đơn hàng,...; services là package chứa các lớp điều khiển có chức năng nhận yêu cầu của lớp bao và tương tác với firebase; models là package chứa các lớp thực thể tương ứng với các collections trong firebase.



Hình 4.3: Thiết kế gói tham gia use case "Đặt hàng"

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Sau đây là các hình vẽ thiết kế một số chức năng chính trong ứng dụng giao đồ ăn. Hình 4.11 mô tả thiết kế giao diện trang chủ chứa danh sách các quán ăn.

The wireframe illustrates the layout of a mobile application's home screen. It is divided into three main horizontal sections:

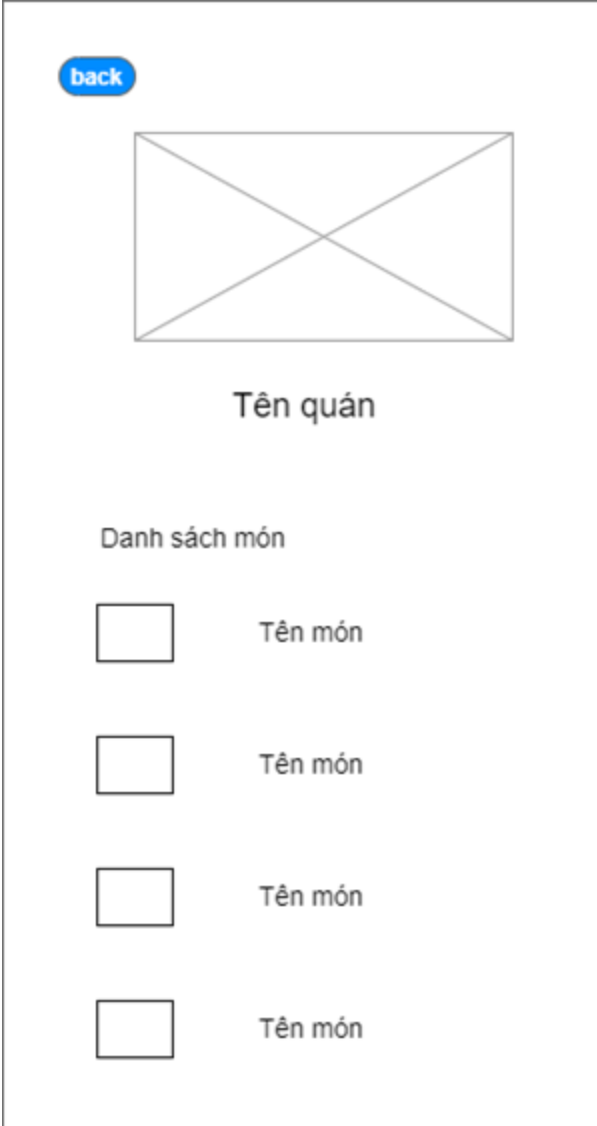
- Search Bar:** Located at the top, it consists of a text input field with the placeholder text "Tìm kiếm món ăn" (Search for food) and a "Search" button to its right.
- Danh mục (Categories):** Below the search bar, this section is labeled "Danh mục". It contains a 2x4 grid of eight empty square boxes, intended for displaying category icons.
- Danh sách quán (Restaurant List):** The bottom section is labeled "Danh sách quán". It displays a list of four restaurant entries. Each entry is composed of a small square placeholder box on the left and the text "Tên quán" (Restaurant Name) to its right.

Hình 4.4: Thiết kế giao diện trang chủ

Màn hình trang chủ được chia thành 3 thành phần con bao gồm:

- SearchBar: Thành phần hiển thị ô nhập từ khóa tìm kiếm và nút tìm kiếm.
- Danh mục: Thành phần giao diện hiển thị các danh mục cửa hàng của ứng dụng.
- Danh sách quán: Thành phần giao diện hiển thị danh sách các nhà hàng.

Hình 4.5 mô tả thiết kế giao diện chi tiết quán ăn.



back

Tên quán

Danh sách món

Tên món

Tên món

Tên món

Tên món

Hình 4.5: Thiết kế giao diện chi tiết quán ăn

Màn hình chi tiết quán ăn được chia thành 3 thành phần con bao gồm:

- Nút Back: Thành phần nút có chức năng quay về màn trước đó.
- Thông tin quán: Thành phần giao diện hiển thị ảnh mô tả và tên quán.
- Danh sách món: Thành phần giao diện hiển thị danh sách các món ăn của quán.

Hình 4.6 mô tả thiết kế giao diện thanh toán.

back Trang thanh toán

Thông tin nhận hàng

Người nhận abc

Địa chỉ HN

Danh sách món

<input type="text"/>	Tên món	x1
<input type="text"/>	Tên món	x2

Tạm tính 100,000đ

Phí áp dụng 15,000đ

Tổng tiền 115,000đ

Đặt hàng

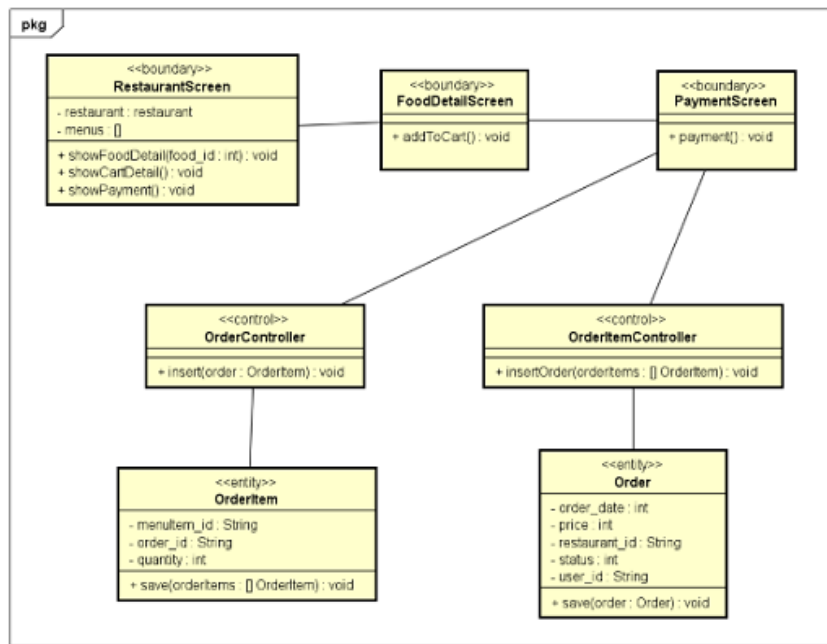
Hình 4.6: Thiết kế giao diện chi tiết quán ăn

Màn hình thanh toán được chia thành 4 thành phần con bao gồm:

- Nút Back: Thành phần nút có chức năng quay về màn trước đó.
- Thông tin nhận hàng: Thành phần giao diện hiển thị thông tin người nhận hàng gồm tên và địa chỉ.
- Danh sách món: Thành phần giao diện hiển thị danh sách các món ăn và số lượng mà người mua đã đặt.
- Thông tin đơn hàng: Thành phần giao diện hiển thị tổng số tiền người mua phải trả và nút xác nhận đặt hàng.

4.2.2 Thiết kế lớp

Hình 4.7 mô tả các lớp chính tham gia chức năng đặt hàng.



Hình 4.7: Thiết kế các lớp chính tham gia chức năng đặt hàng

a, Thiết kế chi tiết lớp RestaurantScreen (boundary)

Thuộc tính:

Tên	Kiểu dữ liệu	Mô tả
restaurant	Restaurant	Thông tin quán ăn
menus	[] Menu	Danh sách các thực đơn của quán

Phương thức:

Tên	Tham số	Kiểu dữ liệu trả về	Mô tả
showFoodDetail	food_id	void	Điều hướng đến màn xem chi tiết món của người dùng
showCartDetail	không	void	Điều hướng đến màn chi tiết giỏ hàng
showPayment	không	void	Điều hướng đến màn thanh toán

b, Thiết kế chi tiết lớp OrderController (control)

Phương thức:

Tên	Tham số	Kiểu dữ liệu trả về	Mô tả
insertOrder	order	Boolean	Tạo đơn hàng mới và lưu vào CSDL
getOrders	không	[] Order	Lấy danh sách các đơn hàng của người dùng
updateOrder	order_id	Boolean	Cập nhật thông tin của đơn hàng

c, Thiết kế chi tiết lớp Order (entity)

Thuộc tính:

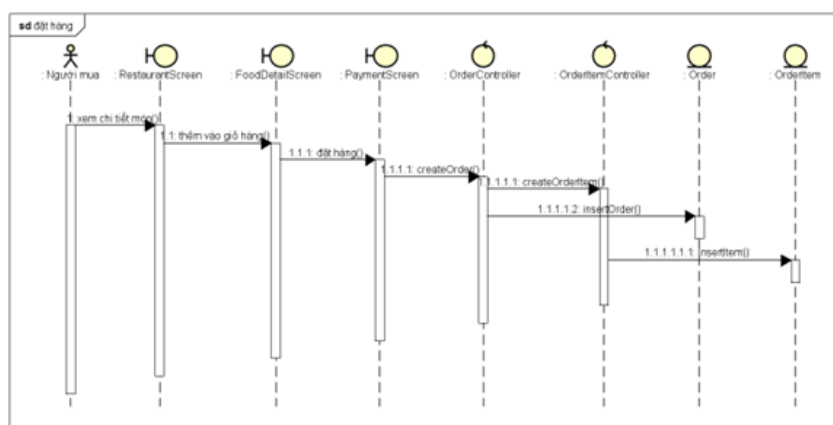
Tên	Kiểu dữ liệu	Mô tả
order_date	int	Ngày tạo đơn hàng
price	int	Tổng tiền của đơn hàng
status	int	Trạng thái của đơn hàng
restaurant_id	String	Id của nhà hàng
user_id	String	Id của người mua đặt đơn hàng

Phương thức:

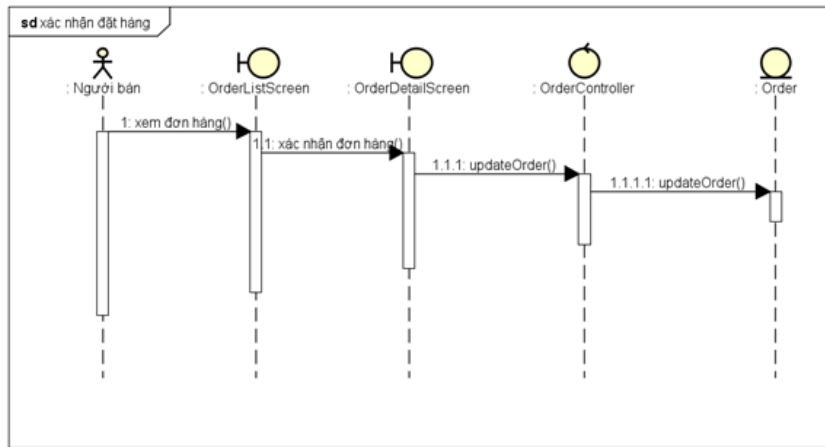
Tên	Tham số	Kiểu dữ liệu trả về	Mô tả
save	order	Boolean	Lưu đơn hàng mới vào CSDL

d, Biểu đồ trình tự các UC quan trọng

Hình 4.8 mô tả biểu đồ trình tự các lớp tham gia chức năng đặt hàng.

**Hình 4.8:** Biểu đồ trình tự chức năng đặt hàng

Hình 4.9 mô tả biểu đồ trình tự các lớp tham gia chức năng xác nhận đặt hàng.



Hình 4.9: Biểu đồ trình tự chức năng xác nhận đặt hàng

4.2.3 Thiết kế cơ sở dữ liệu

Thiết kế cơ sở dữ liệu cho hệ thống giao đồ ăn được thể hiện ở Hình 4.10.



Hình 4.10: Thiết kế cơ sở dữ liệu cho hệ thống giao đồ ăn

Chi tiết thiết kế các bảng như sau: (tổng số 8 bảng)

- Bảng users: lưu trữ các thông tin của người sử dụng, bao gồm id, name, email, password, phone, image, address_id, role(xác định là người mua hay người bán).
- Bảng addresses: lưu trữ các thông tin của địa chỉ, bao gồm id, city, district, street.
- Bảng restaurants: lưu trữ các thông tin của nhà hàng, bao gồm id, name, image, address_id, category_id.

- Bảng categories: lưu trữ các danh mục nhà hàng, bao gồm id, name, image.
- Bảng menus: lưu trữ các thông tin của thực đơn, bao gồm id, name, restaurant_id.
- Bảng menuItems: lưu trữ các thông tin của món ăn, bao gồm id, name, image, price, menu_id.
- Bảng orders: lưu trữ các thông tin của đơn hàng, bao gồm id, user_id, restaurant_id, price, status.
- Bảng orderItems: lưu trữ các thông tin của chi tiết đơn hàng, bao gồm id, order_id, menuItem_id, quantity.

Từ biểu đồ thực thể liên kết, em đã thiết kế cơ sở dữ liệu phi quan hệ NoSQL sử dụng Cloud Firestore như Hình 4.10 . Cơ sở dữ liệu gồm có 8 collections đó là users, addresses, categories, restaurants, menus, menuItems, orders, orderItems. Chi tiết collection users được trình bày trong bảng sau:

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	String	id của người dùng
name	String	tên người dùng
email	String	địa chỉ email người dùng
password	String	mật khẩu người dùng
image	String	ảnh đại diện của người dùng
role	String	vai trò người dùng(người mua hoặc người bán)

Bảng 4.1: Mô tả chi tiết collection users

Chi tiết collection addresses được trình bày trong bảng sau:

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	String	id của address
city	String	tên tỉnh/ thành phố
district	String	tên quận/ huyện
street	String	tên đường

Bảng 4.2: Mô tả chi tiết collection addresses

Chi tiết collection categories được trình bày trong bảng sau:

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	String	id của category
name	String	tên danh mục
title	String	tiêu đề danh mục
image	String	ảnh mô tả danh mục

Bảng 4.3: Mô tả chi tiết collection categories

Chi tiết collection restaurants được trình bày trong bảng sau:

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	String	id của nhà hàng
name	String	tên nhà hàng
address_id	String	id địa chỉ của nhà hàng
category_id	String	id danh mục của nhà hàng
image	String	ảnh mô tả của nhà hàng
user_id	String	id chủ nhà hàng

Bảng 4.4: Mô tả chi tiết collection restaurants

Chi tiết collection menus được trình bày trong bảng sau:

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	String	id của menu
restaurant_id	String	id của nhà hàng
name	String	tên của thực đơn

Bảng 4.5: Mô tả chi tiết collection menus

Chi tiết collection menuItems được trình bày trong bảng sau:

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	String	id của món ăn
menu_id	String	id của thực đơn chứa món ăn
name	String	tên món ăn
image	String	ảnh mô tả của món ăn
price	String	giá món ăn

Bảng 4.6: Mô tả chi tiết collection menuItems

Chi tiết collection orders được trình bày trong bảng sau:

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	String	id của đơn hàng
order_date	String	ngày đặt hàng
status	Number	trạng thái của đơn hàng
menuItem_id	String	id của món ăn
price	Number	tổng tiền của đơn hàng
user_id	String	id người đặt hàng

Bảng 4.7: Mô tả chi tiết collection orders

Chi tiết collection orderItems được trình bày trong bảng sau:

Tên trường	Kiểu dữ liệu	Ý nghĩa
id	String	id của chi tiết đơn hàng
menuItem_id	String	id của món ăn
order_id	String	id của đơn hàng
quantity	String	Số lượng

Bảng 4.8: Mô tả chi tiết collection orderItems

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Trong quá trình phát triển hệ thống, em đã sử dụng một số công cụ hỗ trợ được em liệt kê trong Bảng 4.9.

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual Studio Code	https://code.visualstudio.com/
Ngôn ngữ lập trình	Javascript	https://www.w3schools.com/js/default.asp
Thư viện lập trình giao diện phía frontend	React Native	https://reactnative.dev/
Nền tảng cung cấp API do Google phát triển	Firebase	https://firebase.google.com/

Bảng 4.9: Danh sách thư viện và công cụ sử dụng

Ngoài ra em còn sử dụng một số thư viện bên thứ 3 việc phát triển hệ thống được dễ dàng, tập trung tối đa vào phần nghiệp vụ. Các thư viện được em liệt kê ở Bảng 4.10

Thư viện	Phiên bản	Mục đích
Axios	0.19.0	Thư viện gửi HTTP request
Redux	4.1.2	Thư viện quản lý state
React-redux	7.2.8	Kết nối React Component đến Redux store
React-native-firebase	14.9.3	Thư viện kết nối với SDK Firebase native cho cả iOS và Android

Bảng 4.10: Danh sách thư viện sử dụng trong ứng dụng giao đồ ăn

4.3.2 Kết quả đạt được

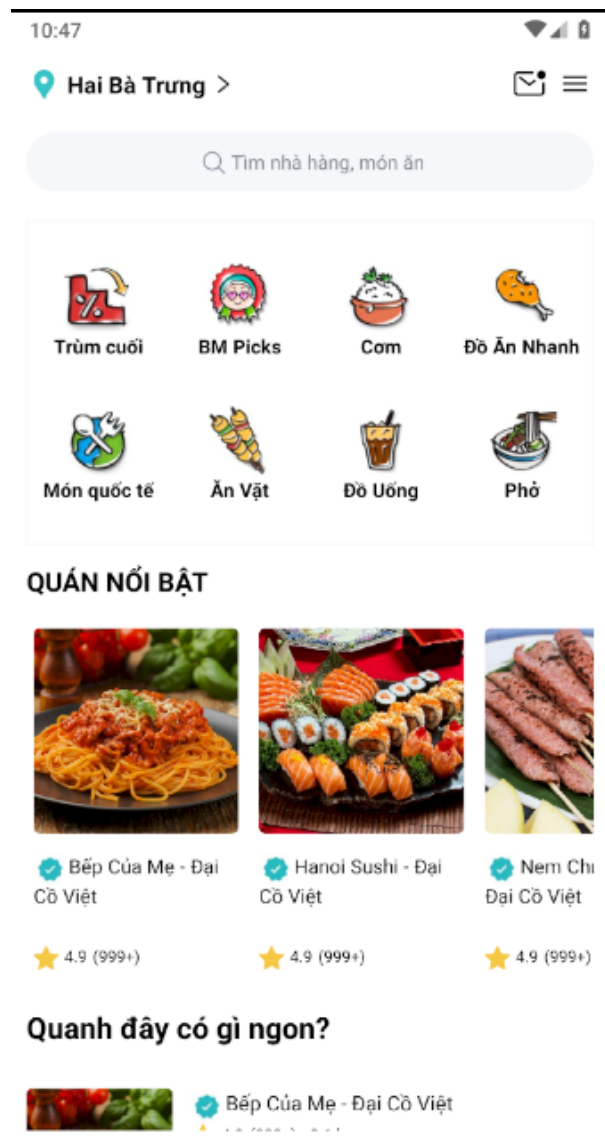
Ứng dụng xây dựng đáp ứng tính năng sử dụng cho 2 nhóm người dùng là người mua và người bán.

Người mua hàng có thể xem danh sách các quán ăn và tìm kiếm món ăn. Sau khi chọn được món ăn, khách hàng sẽ thêm vào giỏ hàng và tiến hành bước thanh toán. Đơn hàng được thanh toán thành công sẽ có trạng thái chờ xác nhận và người mua có thể xem lịch sử đơn hàng đã đặt.

Hệ thống cung cấp chức năng quản lý nhà hàng cho người bán. Họ có thể xem danh sách các đơn hàng của quán. Người bán cũng có chức năng được chỉnh sửa các món ăn trong quán đồng thời có thể xem thống kê doanh thu nhà hàng theo mốc thời gian cụ thể.

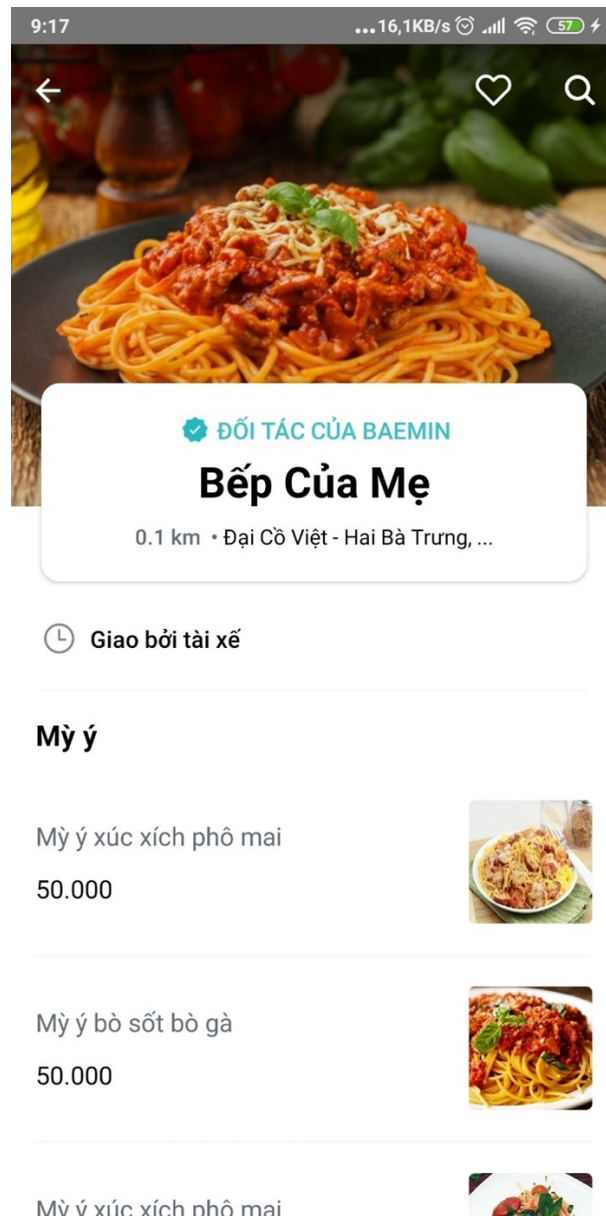
4.3.3 Minh họa các chức năng chính

Hình 4.11 thể hiện giao diện của màn hình Trang chủ cho người mua. Trang chủ hiển thị danh mục các nhà hàng, danh sách các nhà hàng gợi ý cho người mua



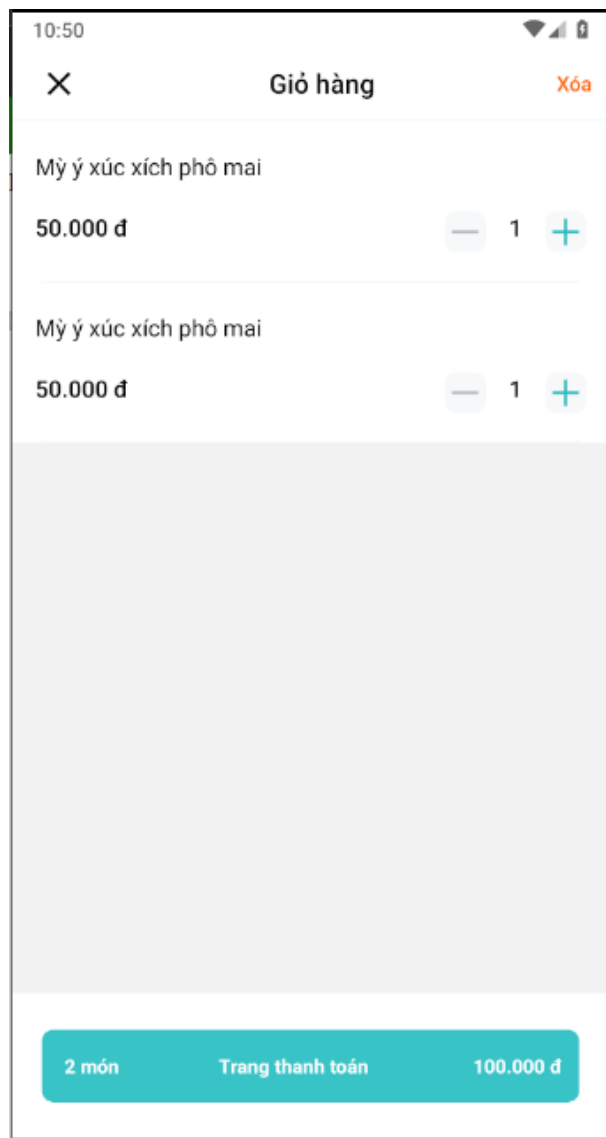
Hình 4.11: Giao diện màn Trang chủ cho người mua

Hình 4.12 thể hiện giao diện của màn hình Xem chi tiết quán ăn cho người mua. Người mua sẽ chọn món rồi thêm vào giỏ hàng.



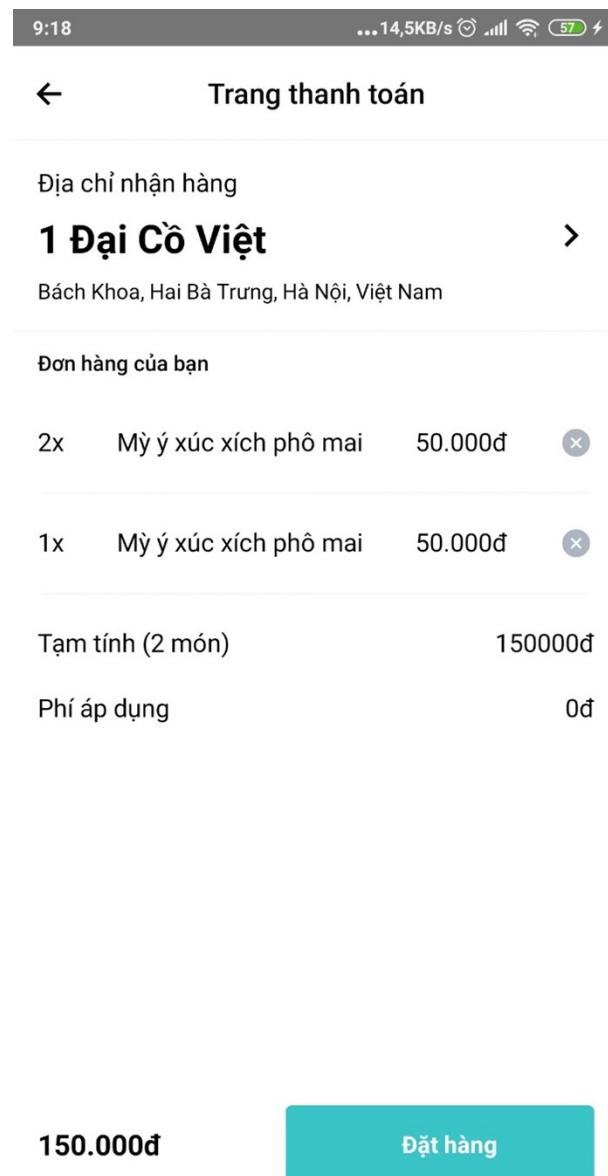
Hình 4.12: Giao diện xem chi tiết quán ăn

Hình 4.13 thể hiện giao diện của màn hình Quản lý giỏ hàng cho người mua. Người mua có thể thêm sửa xóa món ăn trong giỏ hàng.



Hình 4.13: Giao diện màn giỏ hàng cho người mua

Hình 4.14 thể hiện giao diện của màn hình Đặt hàng cho người mua. Người mua sẽ tiến hành xác nhận đặt đơn hàng.



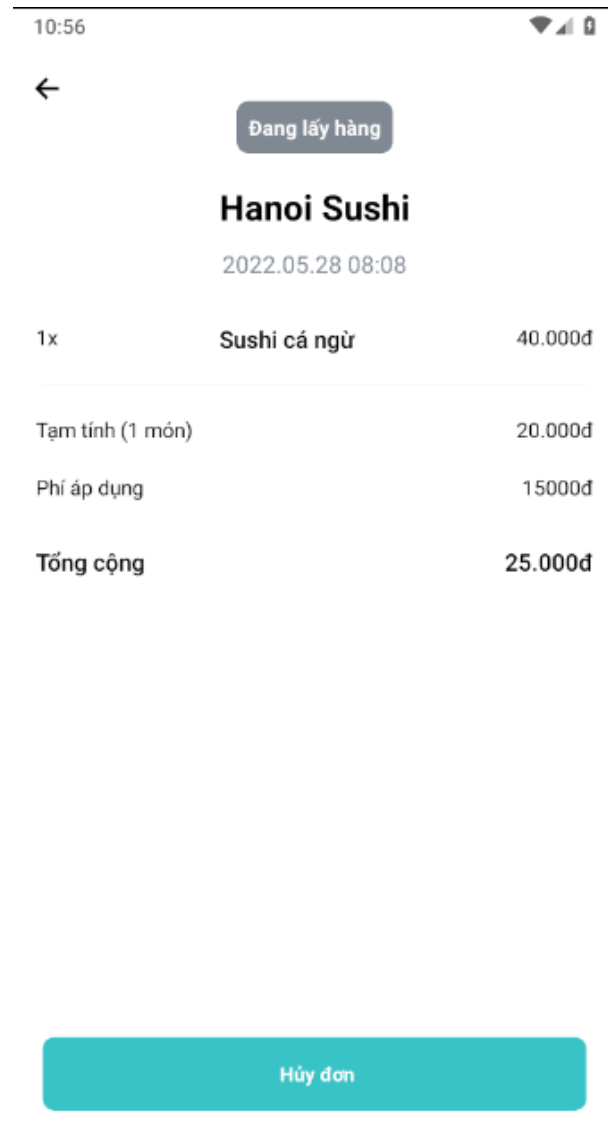
Hình 4.14: Giao diện thanh toán

Hình 4.15 thể hiện giao diện của màn hình Xem lịch sử đơn hàng cho người mua. Người mua xem chi tiết đơn đã đặt và có thể hủy đơn hàng nếu đơn hàng vẫn ở trạng thái chờ xác nhận.



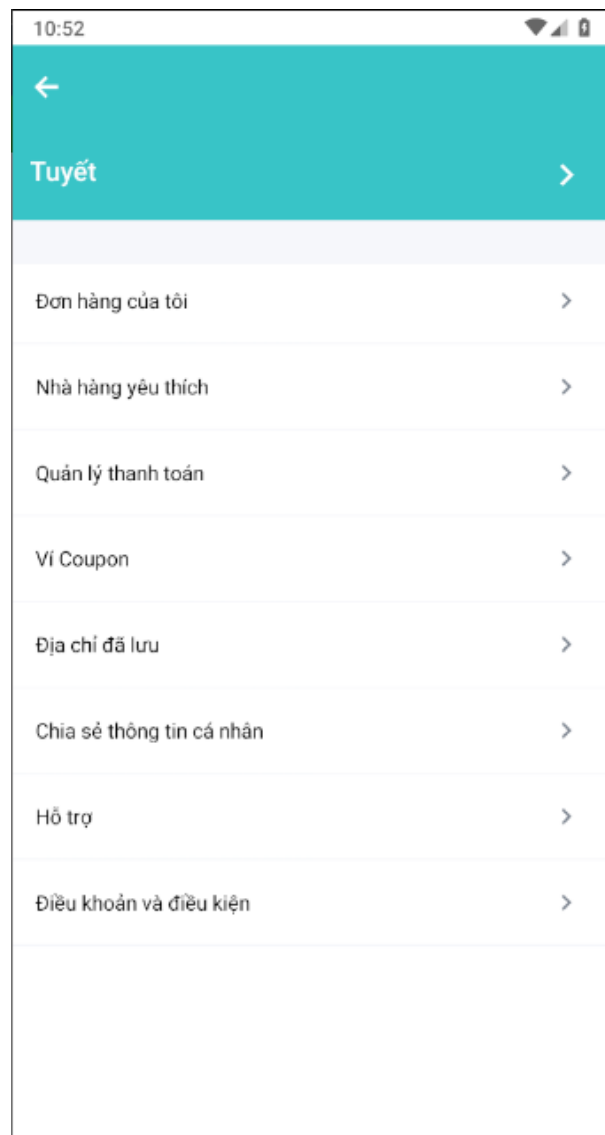
Hình 4.15: Giao diện theo dõi đơn hàng cho người mua

Hình 4.16 thể hiện giao diện của màn hình Chi tiết đơn hàng cho người mua.



Hình 4.16: Giao diện Chi tiết đơn hàng cho người mua

Hình 4.17 thể hiện giao diện của màn hình Quản lý thông tin cá nhân cho người mua.



Hình 4.17: Giao diện màn quản lý thông tin cá nhân cho người mua

Hình 4.18 thể hiện giao diện của màn hình Theo dõi đơn hàng cho người bán. Người bán xem chi tiết đơn mà khách đã đặt và có thể ấn xác nhận đơn hàng.

9:18 ...23,2KB/s 57%

Đơn hàng

#2vRg...LaaV	Đã giao hàng
135.000 đ 2 món	2022.05.31 09:20
#hWF9...rnrG	Đã giao hàng
85.000 đ 2 món	2022.05.29 10:46
#x0yT...iRel	Đang giao hàng
85.000 đ 1 món	2022.05.29 10:46

Đơn hàng

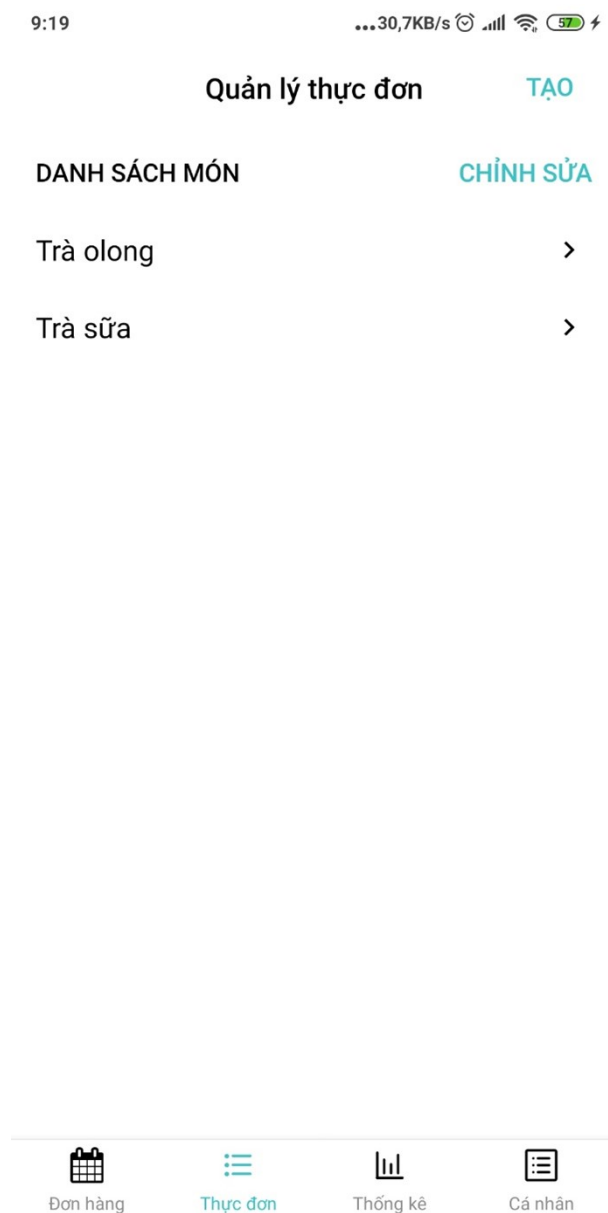
Thực đơn

Thống kê

Cá nhân

Hình 4.18: Giao diện theo dõi đơn hàng cho người bán

Hình 4.19 thể hiện giao diện của màn hình Quản lý món ăn cho người bán. Người bán có thể chỉnh sửa thực đơn, chỉnh sửa món ăn hoặc thêm mới món ăn.



Hình 4.19: Giao diện quản lý món ăn cho người bán

Hình 4.20 thể hiện giao diện của màn hình Thống kê doanh thu cho người bán. Người bán có thể xem doanh thu theo mốc thời gian xác định.

Hình 4.20: Giao diện thống kê cho người bán

4.4 Kiểm thử

Đồ án thực hiện bước kiểm thử sản phẩm sau cùng bằng kỹ thuật Black-Box. Chi tiết kiểm thử một số chức năng được mô tả phía dưới.

4.4.1 Kiểm thử chức năng Tìm kiếm món ăn

Bảng 4.11: Kiểm thử chức năng Tìm kiếm món ăn

Đầu vào	Đầu ra	Kết quả
Người dùng không nhập từ khóa	Hệ thống không trả về danh sách kết quả	Đạt
Người dùng nhập từ khóa	Hệ thống trả về danh sách kết quả theo từ khóa người dùng nhập	Đạt

Đầu vào	Đầu ra	Kết quả
---------	--------	---------

4.4.2 Kiểm thử chức năng Đặt hàng**Bảng 4.12:** Kiểm thử chức năng Đặt hàng

Đầu vào	Đầu ra	Kết quả
Người dùng chọn món, số lượng rồi ấn thêm vào giỏ hàng	Hiển thị trạng thái của giỏ hàng khi người dùng đã thêm món	Đạt
Người dùng ấn vào xem chi tiết giỏ hàng	Hệ thống trả về giao diện đúng với thông tin người dùng chỉnh sửa	Đạt
Người dùng ấn nút đặt hàng	Hệ thống xử lý và trả về kết quả đặt hàng thành công và đi đến trang chủ	Đạt

4.4.3 Kiểm thử chức năng Xem lịch sử đặt hàng**Bảng 4.13:** Kiểm thử chức năng Xem lịch sử đặt hàng

Đầu vào	Đầu ra	Kết quả
Người dùng ấn xem danh sách đơn hàng của tôi	Hệ thống lấy dữ liệu và trả về danh sách đơn hàng cho người dùng	Đạt
Người dùng ấn xem chi tiết đơn hàng	Hệ thống xử lý yêu cầu và đi đến trang chi tiết đơn hàng tương ứng	Đạt
Người dùng ấn hủy đơn đối với đơn hàng đang ở trạng thái chờ xác nhận	Hệ thống xử lý và thông báo hủy thành công, giao diện chi tiết đơn hàng thay đổi	Đạt

4.4.4 Kiểm thử chức năng Chỉnh sửa đơn hàng**Bảng 4.14:** Kiểm thử chức năng Chỉnh sửa đơn hàng

Đầu vào	Đầu ra	Kết quả
Người dùng đăng nhập với vai trò là người bán	Hệ thống trả về danh sách đơn hàng của quán tương ứng với tài khoản email người bán đã đăng nhập	Đạt

Đầu vào	Đầu ra	Kết quả
Người bán sẽ ấn nút Xác nhận đơn hàng với những đơn đang chờ xác nhận	Hệ thống xử lý và trả về thông báo cập nhật tình trạng đơn hàng thành công	Đạt

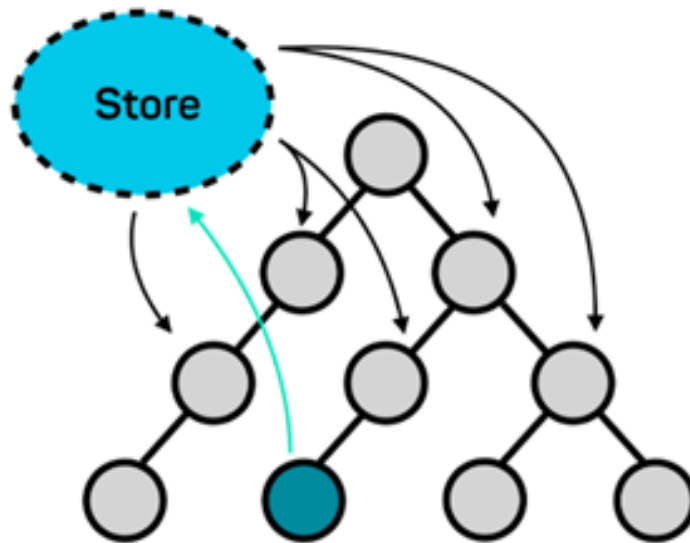
4.4.5 Kiểm thử chức năng Chỉnh sửa thông tin cá nhân

Bảng 4.15: Kiểm thử chức năng Chỉnh sửa thông tin cá nhân

Đầu vào	Đầu ra	Kết quả
Người dùng ấn chỉnh sửa số điện thoại và nhập số không hợp lệ	Hệ thống thông báo số điện thoại không hợp lệ	Đạt
Người dùng ấn xem thông tin cá nhân	Hệ thống hiển thị thông tin người dùng bao gồm: email, số điện thoại, tên,...	Đạt

Để có thể truyền dữ liệu một cách tối ưu và đơn giản hơn chúng ta sẽ lưu dữ liệu vào một store, từ đó cấp phát dữ liệu cho các component cần thiết. Lúc này store sẽ đóng vai trò trung gian, nó có nhiệm vụ chứa và phân phát dữ liệu.

Cùng xem hình minh hoạ bên dưới.

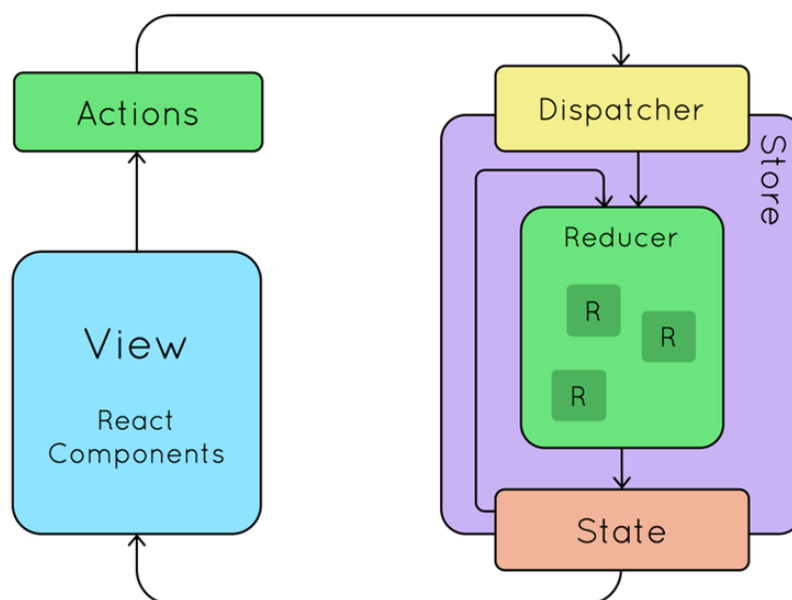


Giả sử bạn muốn chia sẻ dữ liệu từ component A tới component C thì chỉ cần đẩy state vào Store và Store sẽ cấp phát dữ liệu cho component C. Đây là mô hình mà Redux sử dụng, qua đó việc kiểm soát dữ liệu sẽ dễ dàng và tối ưu hơn.

Cách hoạt động của Redux:

Store được coi là phần quan trọng nhất trong Redux, nó có nhiệm vụ lưu trữ và phân phát dữ liệu cho các component. Trong store bao gồm các thành phần như dispatcher (có nhiệm vụ kích hoạt các action bên trong reducer), reducer có nhiệm vụ xử lý các hành động được gửi đến. Sau đây là mô hình cách thức hoạt động của Redux.

Sau khi một action được thực thi, dispatcher sẽ được kích hoạt và gửi đến reducer một action. Lúc này reducer thực hiện hành động dựa vào action được gửi đến. Sau đó, đồng thời lưu lại giá trị của state mới vào trong store và trả về state mới đó.



5.2 Lazy load Flatlist

5.2.1 Vấn đề

Hãy tưởng tượng rằng bạn bước vào một quán nhậu:

Bạn gọi một loạt 10 món ăn, bạn đợi quán chế biến và mang thức ăn lên cho bạn.

- Bạn đợi thật lâu và dùng một cái quán bê lên cả 10 món một lúc.
- Bạn có thật sự mong đợi điều trên không?

Đó cũng tương tự như vấn đề về trải nghiệm người dùng khi sử dụng ứng dụng điện thoại di động.

Khi dữ liệu cần tải về quá lớn và ứng dụng sẽ phải chờ đến khi tải hết dữ liệu, lúc đó người dùng cảm thấy không thoải mái.

5.2.2 Giải pháp

Trong DATN này, phần Flatlist chứa danh sách các quán ăn được gọi từ Firebase về. Giải pháp để tăng trải nghiệm người dùng trong trường hợp này là sử dụng Lazy load.

Lazy loading [14] là kỹ thuật thực hiện trì hoãn (defer) tải các tài nguyên không quan trọng (non-critical resources) vào thời điểm tải trang (page load time). Thay vì tải ngay lập tức, các tài nguyên không quan trọng này chỉ tải vào thời điểm cần thiết (moment of need).

Ưu điểm của Lazy Loading:

- Điểm cộng lớn của Lazy Loading là sự ảnh hưởng đến hiệu suất ứng dụng bởi

chức năng chỉ tải những dữ liệu cần thiết và trong tầm nhìn của users.

- Giúp hạn chế sự chậm trễ xảy ra khi tải đồng loạt dữ liệu trong ứng dụng.
- Lazy loading là kỹ thuật tải hiệu quả nhất được sử dụng phổ biến ngày nay, nó không bao giờ mất chi phí băng thông cho những gì không tải xuống, không xem, không nhìn thấy. Nếu lưu lượng truy cập là mối quan tâm lớn nhất của bạn, thì lazy loading chính là một lựa chọn lí tưởng.

Cách implement Lazy load cho Flatlist đó là lấy dữ liệu Flatlist theo page-size khi scroll. Tức là mỗi khi người dùng muốn xem thêm dữ liệu thì sẽ phải scroll màn hình và dữ liệu cũng chỉ lấy 1 số lượng được xác định.

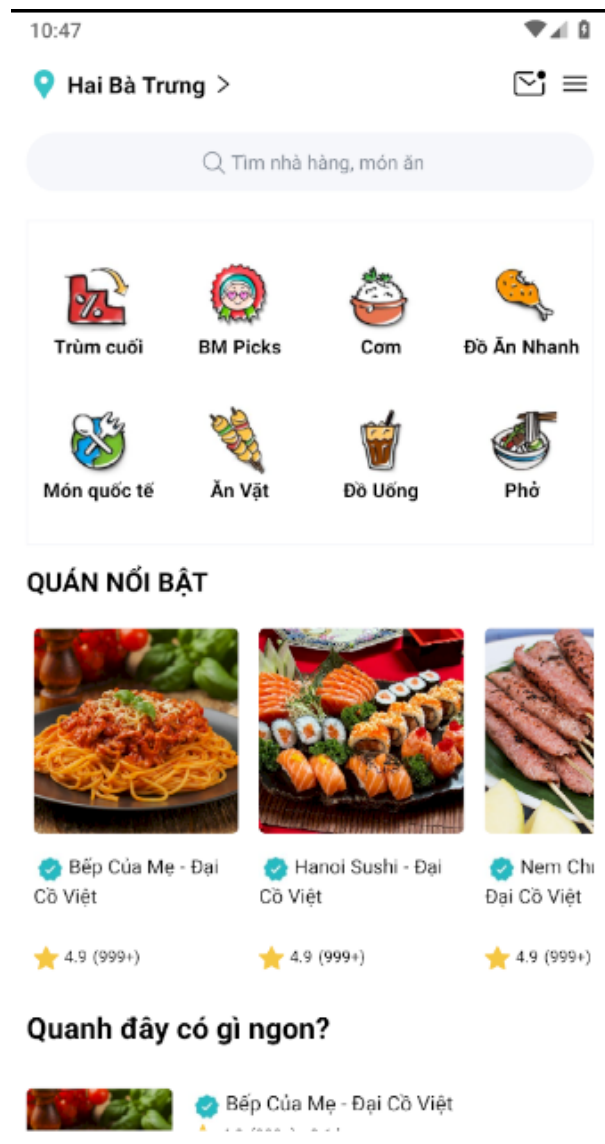
5.3 Xây dựng hệ thống đặt hàng

5.3.1 Vấn đề

Trong quá trình người mua đặt hàng, việc đầu tiên là tìm kiếm món ăn mà mình mong muốn là rất quan trọng. Danh sách các món ăn cần phải sắp xếp một cách khoa học và dễ dàng cho người dùng để họ tìm kiếm một cách nhanh chóng nhất.

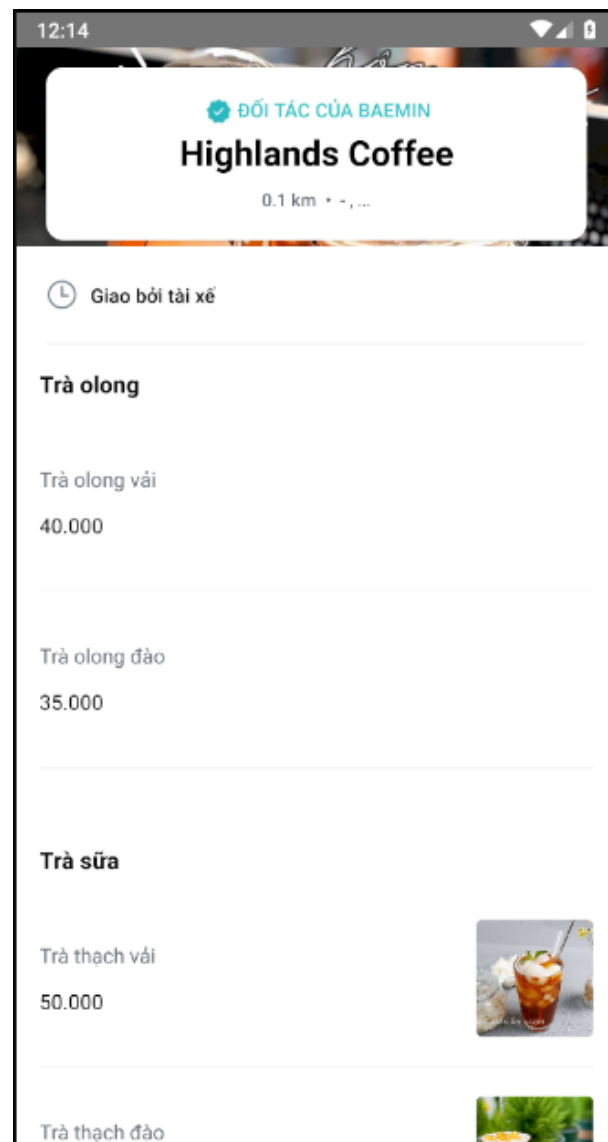
5.3.2 Giải pháp

Để xây dựng một danh sách các nhà hàng, món ăn hợp lý nhất, em đã tham khảo cách tổ chức các quán ăn của những app phổ biến hiện nay. Sau khi tìm hiểu thì em đã quyết định sắp xếp các quán ăn theo từng danh mục loại quán ăn cụ thể và được hiển thị ở đầu trang chủ. Tiếp đó là danh sách các quán ăn nổi bật được hệ thống chọn ra và đề xuất cho người dùng. Hình 5.1 minh họa trang chủ:



Hình 5.1: Màn hình trang chủ

Trong chi tiết của quán ăn, các món ăn cũng được sắp xếp theo các danh mục loại thực đơn cụ thể. Hình 5.2 minh họa chi tiết quán được chia món theo các loại thực đơn:



Hình 5.2: Màn hình chi tiết quán ăn

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Sau quá trình làm đồ án, em đã xây dựng được ứng dụng giao đồ ăn. Ứng dụng giải quyết được yêu cầu bài toán đặt ra, đó là cho phép người mua chọn món ăn và đặt hàng online, giúp người mua thuận tiện trong việc mua bán cũng như người bán tiếp cận được với nhiều khách hàng hơn. Trong quá trình xây dựng ứng dụng giao đồ ăn, em đã được củng cố cũng như áp dụng các kiến thức đã học để tiếp thu các công nghệ thực tế giúp phát triển và xây dựng ứng dụng.

Các ưu điểm nổi bật của ứng dụng xây dựng có thể kể đến như là

- Hệ thống cung cấp giao diện sắp xếp món ăn theo danh mục giúp người mua dễ dàng lựa chọn.
- Hệ thống cho phép người mua đặt hàng nhanh chóng cũng như theo dõi đơn mua và hủy đơn dễ dàng.
- Hệ thống cung cấp các chức năng quản lý nhà hàng cho người bán cùng với chức năng thống kê doanh thu theo mốc thời gian cụ thể.

Tuy nhiên, trong quá trình xây dựng ứng dụng, do thời gian và kinh phí không cho phép, ứng dụng vẫn còn những nhược điểm như là:

- Chưa xây dựng web quản lý dành cho quản trị viên.
- Chưa xây dựng ứng dụng dành cho shipper.
- Ứng dụng chưa cung cấp bản đồ giúp người mua tìm kiếm món ăn và theo dõi đơn hàng thuận tiện hơn.
- Chưa có chức năng gợi ý món ăn đối với từng khách hàng.
- Chưa có chức năng đánh giá đơn hàng.

6.2 Hướng phát triển

Trong tương lai, khi ứng dụng được đầu tư thời gian cũng như kinh phí xây dựng, em dự định sẽ mở rộng theo các hướng sau:

- Xây dựng trang web quản lý cho quản trị viên hệ thống.
- Xây dựng ứng dụng mobile dành cho shipper.
- Xây dựng chức năng xem lộ trình tài xế.
- Kết hợp thanh toán điện tử.
- Xây dựng tính năng gợi ý món ăn đối với từng khách hàng.

- Xây dựng tính năng đánh giá đơn hàng.

TÀI LIỆU THAM KHẢO

- [1] *Learn rn firebase*. [Online]. Available: <https://rnfirebase.io/> (visited on 07/30/2022).
- [2] *Báo cáo của euromonitor*. [Online]. Available: <https://vnexpress.net/cuoc-dua-giao-thuc-an-tan-noi-o-viet-nam-3776814.html> (visited on 07/30/2022).
- [3] *So sánh các app*. [Online]. Available: <https://tinhthe.vn/thread/cuoc-chien-giao-nhan-thuc-an-uu-nhuoc-diem-cua-tung-doi-thu-ban-chon-gi.3026495/> (visited on 07/30/2022).
- [4] *Thông tin shopeefood*. [Online]. Available: <https://shopeefood.vn/> (visited on 07/30/2022).
- [5] *Thông tin grab*. [Online]. Available: <https://www.grab.com/vn/> (visited on 07/30/2022).
- [6] *Thông tin gofood*. [Online]. Available: <https://gofood.vn/> (visited on 07/30/2022).
- [7] *Learn react-native*. [Online]. Available: <https://reactnative.dev/> (visited on 07/30/2022).
- [8] *Learn html*. [Online]. Available: <https://www.w3schools.com/html/> (visited on 07/30/2022).
- [9] *Learn swift*. [Online]. Available: <https://developer.apple.com/swift/> (visited on 07/30/2022).
- [10] *Learn java*. [Online]. Available: <https://dev.java/learn/> (visited on 07/30/2022).
- [11] *Learn javascript*. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (visited on 07/30/2022).
- [12] *Learn redux*. [Online]. Available: <https://redux.js.org/> (visited on 07/30/2022).
- [13] *Redux giải quyết vấn đề gì?* [Online]. Available: <https://hocdauthau.com/redux-la-gi-tai-sao-lai-ung-dung-trong-reactjs/> (visited on 07/30/2022).
- [14] *Lazy-load giải quyết vấn đề gì?* [Online]. Available: <https://tmarketing.vn/tong-quan-lazy-loading-la-gi/> (visited on 07/30/2022).