**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Xây dựng ứng dụng giao đồ ăn**

**HÒANG THỊ TUYẾT**

tuyet.ht184013@sis.hust.edu.vn

**Ngành Công nghệ thông tin**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Nguyễn Tiến Thành |
| **Bộ môn:** | Công nghệ phần mềm |
| **Viện:** | Công nghệ thông tin – Truyền thông |
| **HÀ NỘI, 5/2022** | |

# Lời cam kết

Họ và tên sinh viên: . . . . . . . .Hoàng Thị Tuyết . . . . . . . . . . . . . . . . . . . . . . . . . . .

Điện thoại liên lạc: . . . 0975976172. . . . . . . . Email: tuyet.ht184013@sis.hust.edu.vn

Lớp: . . . . . . . CNTT-01 . . . . . . . . . . . . . . . . . Hệ đào tạo: . . Đại học đại trà . . . . . . .

Tôi –*Hoàng Thị Tuyết*– cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS. Nguyễn Tiến Thành*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Tác giả ĐATN  *Họ và tên sinh viên* |

# 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 thiết sót và hạn chế. Em rất mong nhận được những đánh giá, nhận xét từ thầy cô và các bạn, để từ đấy 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

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 nước uống, 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

[Lời cam kết ii](#_Toc44922331)

[Lời cảm ơn iii](#_Toc44922332)

[Tóm tắt iv](#_Toc44922333)

[Abstract v](#_Toc44922334)

[Mục lục vi](#_Toc44922335)

[Danh mục hình vẽ ix](#_Toc44922336)

[Danh mục bảng x](#_Toc44922337)

[Danh mục công thức xi](#_Toc44922338)

[Danh mục các từ viết tắt xii](#_Toc44922339)

[Danh mục thuật ngữ xiii](#_Toc44922340)

[Chương 1 Giới thiệu đề tài 1](#_Toc44922341)

[1.1 Đặt vấn đề 1](#_Toc44922342)

[1.2 Mục tiêu và phạm vi đề tài 2](#_Toc44922343)

[1.3 Định hướng giải pháp 2](#_Toc44922344)

[1.4 Bố cục đồ án 2](#_Toc44922345)

[Chương 2 Khảo sát và phân tích yêu cầu 4](#_Toc44922346)

[2.1 Khảo sát hiện trạng 4](#_Toc44922347)

[2.2 Tổng quan chức năng 4](#_Toc44922348)

[2.2.1 Biểu đồ use case tổng quan 4](#_Toc44922349)

[2.2.2 Biểu đồ use case phân rã XYZ 5](#_Toc44922350)

[2.2.3 Quy trình nghiệp vụ 5](#_Toc44922351)

[2.3 Đặc tả chức năng 5](#_Toc44922352)

[2.3.1 Đặc tả use case A 5](#_Toc44922353)

[2.3.2 Đặc tả use case B 5](#_Toc44922354)

[2.4 Yêu cầu phi chức năng 5](#_Toc44922355)

[Chương 3 Công nghệ sử dụng 6](#_Toc44922356)

[Chương 4 Phát triển và triển khai ứng dụng 7](#_Toc44922357)

[4.1 Thiết kế kiến trúc 7](#_Toc44922358)

[4.1.1 Lựa chọn kiến trúc phần mềm 7](#_Toc44922359)

[4.1.2 Thiết kế tổng quan 7](#_Toc44922360)

[4.1.3 Thiết kế chi tiết gói 8](#_Toc44922361)

[4.2 Thiết kế chi tiết 8](#_Toc44922362)

[4.2.1 Thiết kế giao diện 8](#_Toc44922363)

[4.2.2 Thiết kế lớp 9](#_Toc44922364)

[4.2.3 Thiết kế cơ sở dữ liệu 9](#_Toc44922365)

[4.3 Xây dựng ứng dụng 9](#_Toc44922366)

[4.3.1 Thư viện và công cụ sử dụng 9](#_Toc44922367)

[4.3.2 Kết quả đạt được 9](#_Toc44922368)

[4.3.3 Minh hoạ các chức năng chính 10](#_Toc44922369)

[4.4 Kiểm thử 10](#_Toc44922370)

[4.5 Triển khai 10](#_Toc44922371)

[Chương 5 Các giải pháp và đóng góp nổi bật 11](#_Toc44922372)

[Chương 6 Kết luận và hướng phát triển 12](#_Toc44922373)

[6.1 Kết luận 12](#_Toc44922374)

[6.2 Hướng phát triển 12](#_Toc44922375)

[Tài liệu tham khảo 13](#_Toc44922376)

[Phụ lục A-1](#_Toc44922377)

[A Hướng dẫn viết đồ án tốt nghiệp A-1](#_Toc44922378)

[A.1 Quy định chung A-1](#_Toc44922379)

[A.2 Ngành học A-2](#_Toc44922380)

[A.3 Tạo đề mục A-2](#_Toc44922381)

[A.4 Bảng biểu A-3](#_Toc44922382)

[A.5 Hình vẽ A-4](#_Toc44922383)

[A.6 Tài liệu tham khảo A-4](#_Toc44922384)

[A.7 Công thức toán học A-5](#_Toc44922385)

[A.8 Tham chiếu chéo A-5](#_Toc44922386)

[A.9 Cập nhật mục lục và tham chiếu chéo A-6](#_Toc44922387)

[A.10 In quyển đồ án tốt nghiệp A-6](#_Toc44922388)

[B Đặc tả use case B-8](#_Toc44922389)

[B.1 Đặc tả use case “Thống kê tình hình mượn sách” B-8](#_Toc44922390)

[B.2 Đặc tả use case “Đăng ký làm thẻ mượn” B-8](#_Toc44922391)

[C Công nghệ sử dụng C-8](#_Toc44922392)

[C.1 Công nghệ bảo mật dữ liệu C-8](#_Toc44922393)

[C.2 Công nghệ blockchain C-8](#_Toc44922394)

[D Thiết kế gói D-8](#_Toc44922395)

[D.1 Thiết kế gói cho kiến trúc tổng quan D-8](#_Toc44922396)

[D.2 Thiết kế gói cho chức năng “Trả sách” D-9](#_Toc44922397)

[E Thiết kế lớp E-9](#_Toc44922398)

# Danh mục hình vẽ

[**Hình 1** Ví dụ biểu đồ phụ thuộc gói 7](#_Toc27562454)

[**Hình 2** Ví dụ thiết kế gói 8](#_Toc27562455)

[**Hình 3** Ví dụ hình vẽ A-4](#_Toc27562456)

[**Hình 4** Quy cách đóng quyển A-6](#_Toc27562457)

[**Hình 5** Quy cách ghi chữ phần gáy A-6](#_Toc27562458)

[**Hình 6** Hướng dẫn thiết lập in hai mặt A-7](#_Toc27562459)

# Danh mục bảng

# Danh mục các từ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface  Giao diện lập trình ứng dụng |
| **EUD** | End-User Development  Phát triển ứng dụng người dùng cuối |
| **GWT** | Google Web Toolkit  Công cụ lập trình Javascript bằng Java của Google |
| **HTML** | HyperText Markup Language  Ngôn ngữ đánh dấu siêu văn bản |
| **CNTT** | Công nghệ thông tin |
| **ĐATN** | Đồ án tốt nghiệp |
| **SV** | Sinh viên |

# Danh mục thuật ngữ

|  |  |
| --- | --- |
| Browser | Trình duyệt |
| **Cache memory** | Bộ nhớ đệm |
| **E-commerce** | Thương mại điện tử |
| **Bloatware** | Ứng dụng nhà sản xuất tích hợp vào thiết bị |
| **Interpreter** | Trình thông dịch |
| **Compiler** | Trình biên dịch |

# 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

## 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 của 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 mà đến các nhà hàng, quán ăn nữa mà họ mong muốn đồ ăn thức uống có thể đặt hàng mà không cần đến tận nơi mà đồ ăn cũng được giao đến nơi mình muốn.

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, 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 hệ thống giao đồ ăn online”. 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 nhanh chóng, dễ dàng và chính xác.

## 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 ấn 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.

## Định hướng giải pháp

Với mục tiêu đáp ứng nhu cầu của người sử dụng, ứng dụng sẽ cố gắng đáp ứng tối đa các tính năng cần có cũng như tương thích với các thiết bị sử dụng công nghệ thông tin. Tuy nhiên, do thời gian của đồ án có hạn nên ứng dụng sẽ được triển khai trên nền tảng mobile trước, sau này có thời gian sẽ phát triển và mở rộng trên nền tảng mobile.

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 – 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.

Sử dụng mô hình MVC để xây dựng hệ thống logic, chặt chẽ dễ quản lý code và logic, giúp việc debug trở nên dễ dàng hơn và nâng cao khả năng mở rộng của dự án

## Bố cục đồ án

Phần còn lại của báo cáo đồ 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 đồ 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 đồ án.

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

# ­­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.

## 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. Và tại Việt Nam những thương hiệu lớn trong lịch 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 nỗi đến từ Hàn Quốc. Sau đây em sẽ phân tích các ưu nhược điểm của từng sản phẩm.

**Bảng 1** So sánh ứng dụng LiveChat, Tawk.to, Subiz Livechat

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

Từng được biết đến là ứng dụng Now, nay đã đổi tên thành ShopeeFood, 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.

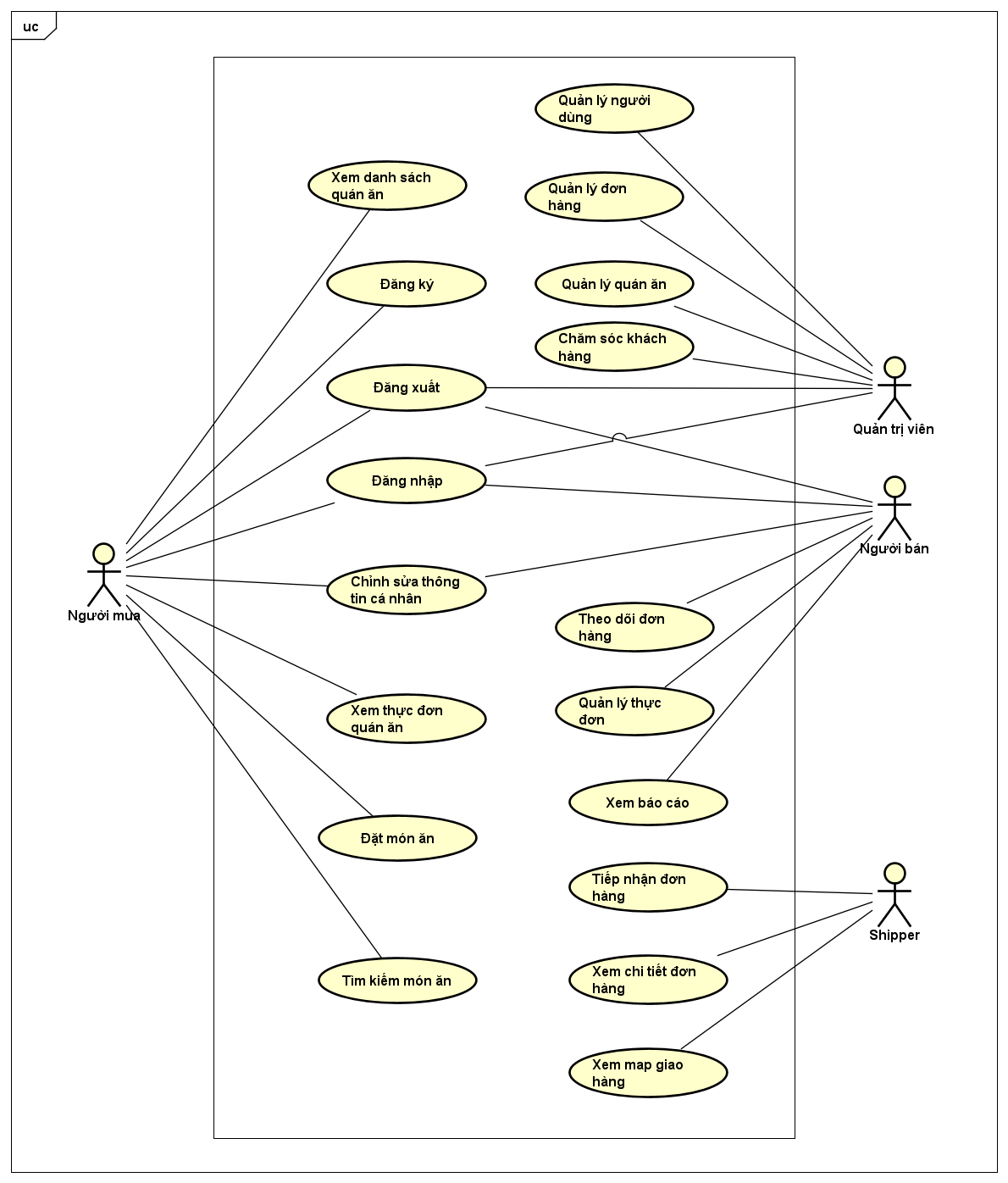
Tương tự như GrabFood, GoFood cũng cung cấp danh sách gồm rất nhiều hàng quán đối tác để khách hàng lựa chọn. Mạng lưới tài xế cũng gọi là nhiều khi so sánh với các nền tảng khác. GoFood trực thuộc Go-Viet công ty con của tập đoàn GoJek, từng được kỳ vọng sẽ trở thành đối thủ nặng ký trong trận chiến, 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 dành ngôi vị dẫn đầu.

Từ những luận điểm trên, em sẽ phát triển hệ thống quản lý và điều phối tin nhắn dựa trên các nhu cầu cần thiết nhất của từng doanh nghiệp, đáp ứng từ từ từng chức năng mà doanh nghiệp mong muốn. Cụ thể các chức năng đó em sẽ trình bày trong các phần tiếp theo.

## Tổng quan chức năng

### Biểu đồ use case tổng quan

Hệ thống giao đồ ăn gồm nhiều thành phần chức năng khác nhau. **Hình 1** mô tả tổng quan toàn bộ chức năng của hệ thống.



Hình 1 Biểu đồ tổng quan

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

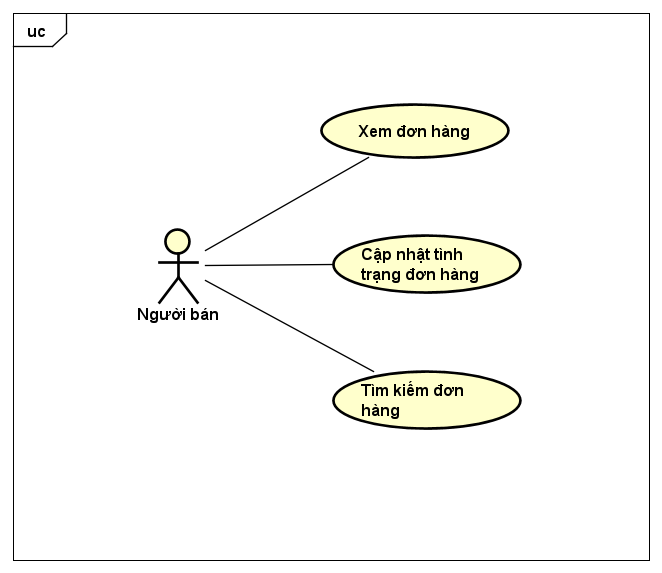
Người mua cần đăng nhập để vào ứng dụng. Khi đăng nhập xong người mua có thể đặt đồ ăn bằng các tìm kiếm đồ ăn, chọn cửa hàng yêu thích, xem đồ ăn , đánh giá và nêu cảm nhận về món ăn đó. Người mua cũng có thể sửa thông tin của mình qua ứng dụng khi vào mục tài khoản.

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.

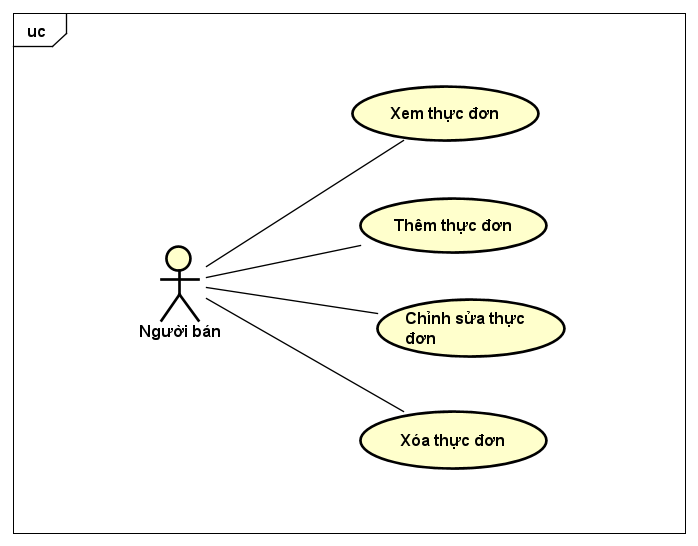
Quản trị viên sẽ có chức năng quản lý người dùng, quản lý đơn hàng, quản lý quán ăn và chăm sóc khách hàng.

### Biểu đồ usecase phân rã chức năng Theo dõi đơn hàng



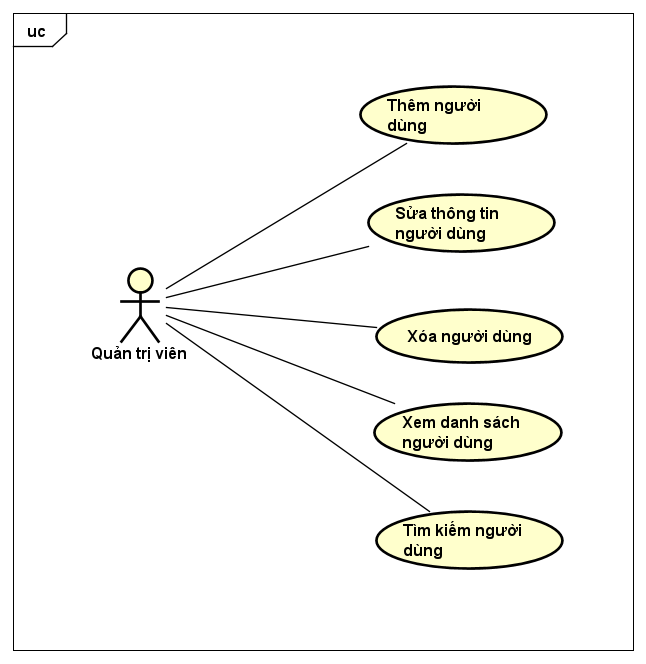
Hình 2 Biểu đồ usecase phân rã “Theo dõi đơn hàng”

### Biểu đồ usecase phân rã chức năng Quản lý thực đơn



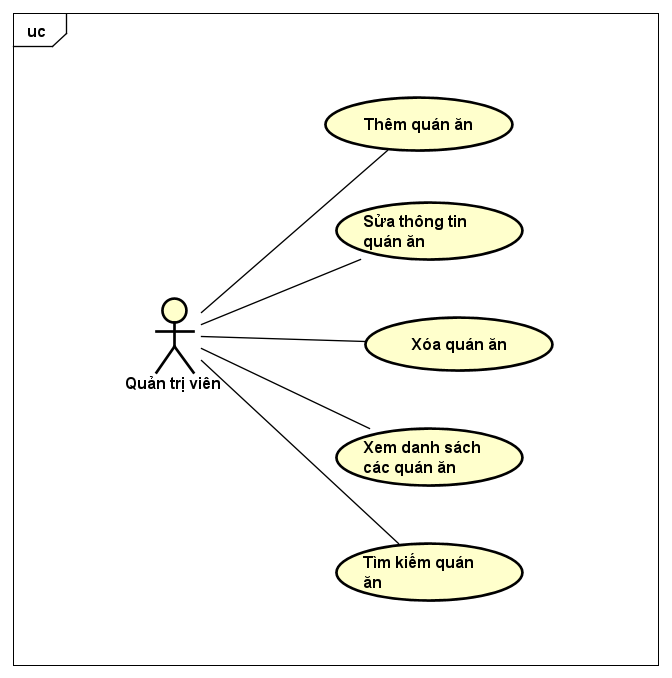
Hình 3 Biểu đồ usecase phân rã “Quản lý thực đơn”

### Biểu đồ usecase phân rã chức năng Quản lý người dùng



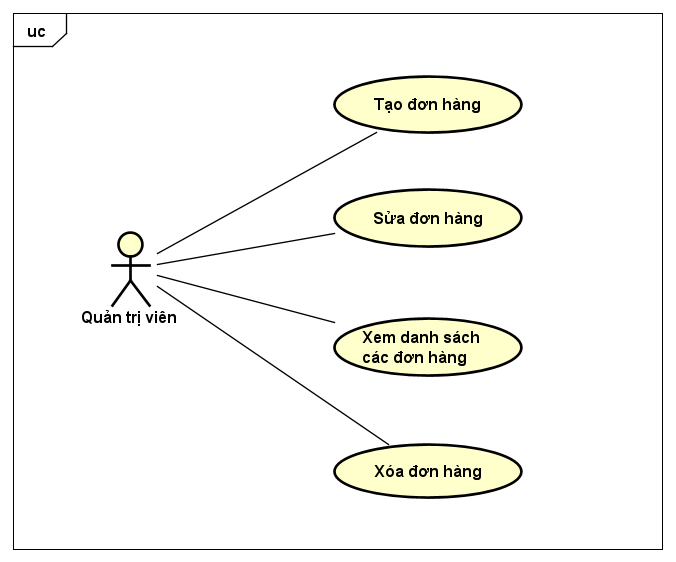
Hình 4 Biểu đồ usecase phân rã “Quản lý người dùng”

### Biểu đồ usecase phân rã chức năng Quản lý quán ăn



Hình 5 Biểu đồ usecase phân rã “Quản lý quán ăn”

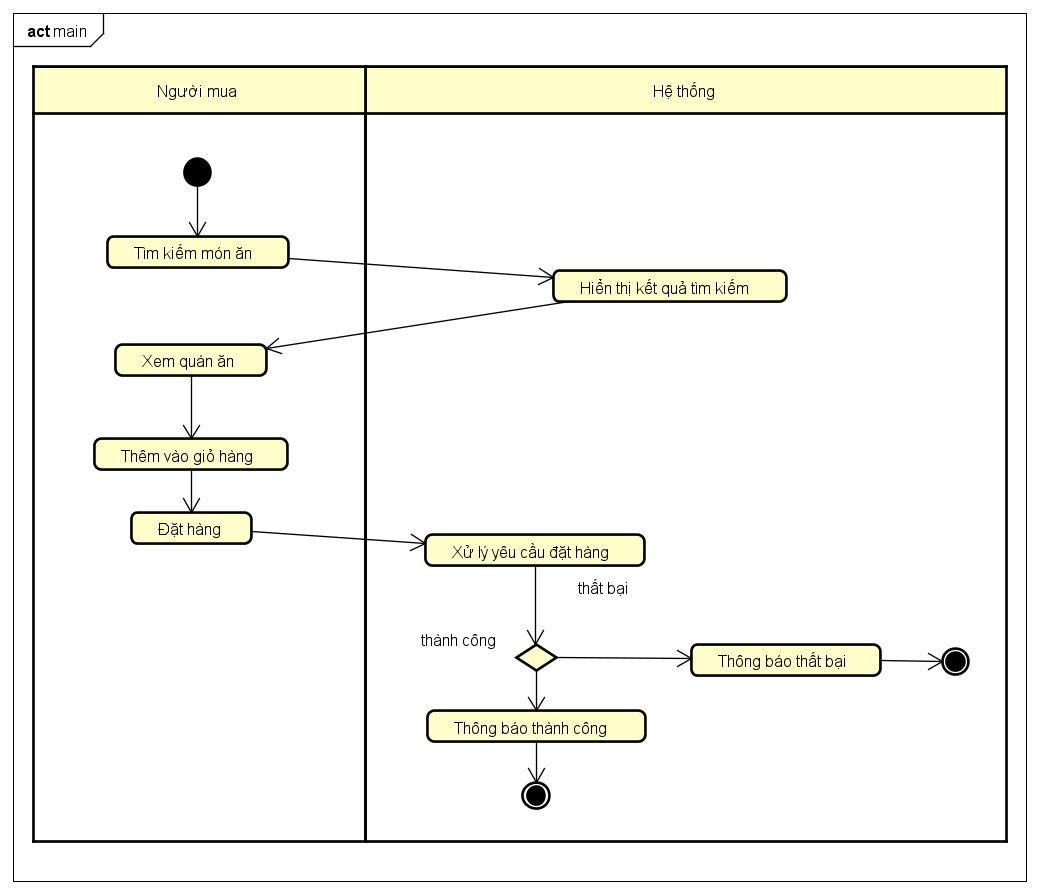
### Biểu đồ usecase phân rã chức năng Quản lý đơn hàng



Hình 6 Biểu đồ phân rã usecase “Quản lý đơn hàng”

### Quy trình nghiệp vụ

**Hình 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 7 Quy trình nghiệp vụ đặt hàng

## Đặc tả chức năng

### Đặc tả use case Tìm kiếm món ăn

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use case** | UC001 | **Tên Use case** | Tìm kiếm món ăn |
| **Tác nhân** | Khách | | |
| **Tiền điều kiện** | Không | | |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | |  | Khách | Nhập món ăn muốn tìm vào ô tìm kiếm | |  | Hệ thống | Xử lý thông tin và trả về kết quả tìm kiếm | |  | Hệ thống | Hiển thị thông tin và kết quả tìm kiếm | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 3a | Hệ thống | Thông báo lỗi trong quá trình tìm kiếm | | | |
| **Hậu điều kiện** | Không | | |

### Đặc tả use case Đặt hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use case** | UC002 | **Tên Use case** | Đặt hàng |
| **Tác nhân** | Khách | | |
| **Tiền điều kiện** | Không | | |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | |  | Khách | Chọn hàng cần đặt rồi chọn đặt hàng. | |  | Hệ thống | Chuyển thông tin hàng sang màn hình đặt hàng | |  | Khách | Kiểm tra đơn hàng, tiền, số lượng hàng và chọn thanh toán | |  | Hệ thống | Chuyển sang màn thanh toán | | 5 | Khách | Chọn thanh toán | | 6 | Hệ thống | Kiểm tra thông tin và thông báo thành công | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 6a. | Hệ thống | Thông báo lỗi quá trình đặt hàng | | | |

### Đặc tả use case Xem lịch sử đặt hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use case** | UC003 | **Tên Use case** | Xem lịch sử đặt hàng |
| **Tác nhân** | Khách | | |
| **Tiền điều kiện** | Không | | |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Khách | 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. | Khách | Chọn vào đơn hàng muốn xem lại | | 5 | Hệ thống | Trả về thông tin đơn hàng muốn xem | | | |
| **Luồng sự kiện thay thế** | Không | | |

### Đặc tả use case Chỉnh sửa đơn hàng

### Đặc tả use case Chỉnh sửa thông tin cá nhân

## Yêu cầu phi chức năng

### Yêu cầu về mặt kỹ thuật

* Phân quyền người sử dụng: đối với từng loại người dùng sẽ có những quyền hạn để truy cập và sử dụng những tính năng nhất định.
* Mật khẩu của người dùng phải được mã hóa trước khi lưu trữ (mã hõa theo chuẩn Bcrypt ). Thông tin của người dùng khi gửi qua Email dưới dạng tệp đính kèm cũng cần phải thiết lập mật khẩu.
* Có khả năng sao lưu và phục hồi dữ liệu cho người quản lý hệ thống.
* Các thao tác phía người dùng trong trường hợp có lỗi phải xác định được lỗi gây ra do người dùng hay do hệ thống.
* Hệ thống đảm bảo khả năng mở rộng, dễ bảo trì

### Yêu cầu về giao diện người dùng

* Giao diện đẹp, đơn giản và thân thiện với người dùng.
* Bố cục rõ ràng, phân chia rành mạch giữa các tính năng.
* Thông báo sau khi thực hiện các theo tác phải rõ rang, phân biệt giữa các tính năng.
* Đảm bảo tính dễ học và dễ sử dụng cho người mới bắt đầu.
* Co giãn theo các kích thước màn hình máy tính.

# Công nghệ sử dụng

## Ngôn ngữ JavaScript

JavaScript, hay thường được gọi tắt là JS, cùng với HTML và CSS là bộ ba tạo nên hệ thống các trang website. Nếu như HTML là bộ xương sống, CSS phủ lên lớp áo màu sắc thì JS sẽ chịu trách nhiệm cho các chuyển động tương tác với người dùng khi sử dụng website.

Là một ngôn ngữ có cú pháp rành mạch, đơn giản và dễ học, JS đang là ngôn ngữ lập trình phổ biến hơn bao giờ hết. Các trình duyệt website hiện đại đều hỗ JS, giúp cho việc trải nghiệm tương tác của người dùng khi sử dụng website trở nên sinh động và đa dạng hơn, đồng thời, nếu thao tác được xử lý ở phía máy khách hàng thì sẽ giảm thiểu lượng truy cập phải xử lý cho máy chủ. Ngoài ra, với khả năng tương tác và mở rộng cao, JS có thể được “nhúng” vào các ứng dụng website khác nhau thông qua các extension (tiện ích mở rộng), góp phần làm đơn giản hóa thao tác sử dụng các chức năng website.

Với ưu điểm chỉ dịch một lần, nhưng có thể chạy nhiều tiến trình song song, JS sẽ có thời gian biên dịch mã nguồn ngắn hơn so với các ngôn ngữ lập trình truyền thống như là Java, .NET hay PHP. Không những vậy, JS còn được sử dụng trên các máy chủ sử dụng platform NodeJS, giúp việc đồng bộ ngôn ngữ giữa máy chủ và máy khách hàng trở nên dễ dàng hơn.

Lựa chọn JS làm ngôn ngữ chủ yếu xây dựng ứng dụng giao đồ ăn đảm bảo hiệu quả cao nhất cũng như tính dễ mở rộng, dễ quản lý và tái sử dụng mã nguồn trong tương lai sắp tới

## React Native

React Native là framework xây dựng ứng dụng di động native sử dụng Javascript do Facebook phát hành. Sử dụng React Native để xây dựng ứng dụng iOS và Android chỉ cần 1 ngôn ngữ javascript duy nhất.

React Native sở hữu rất nhiều ưu điểm tuyệt vời. Cụ thể là:

* Phát triển đồng thời 2 nền tảng IOS, Andoird giúp tiết kiệm thời gian.
* Ngôn ngữ Javascript kết hợp JSX dễ dàng tiếp cận.
* Dễ dàng tái sử dụng code với Component.
* Dễ dàng chuyển sang code ReactJS
* Cộng đồng lớn và rất nhiều hướng dẫn, dễ dàng tiếp cận với những người mới bắt đầu lập trình.

Tuy nhiên nó vẫn tồn tại những nhược điểm sau:

* Hiệu năng không bằng ứng dụng native.
* Vẫn phải biết native code nếu làm sâu.

## 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 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:

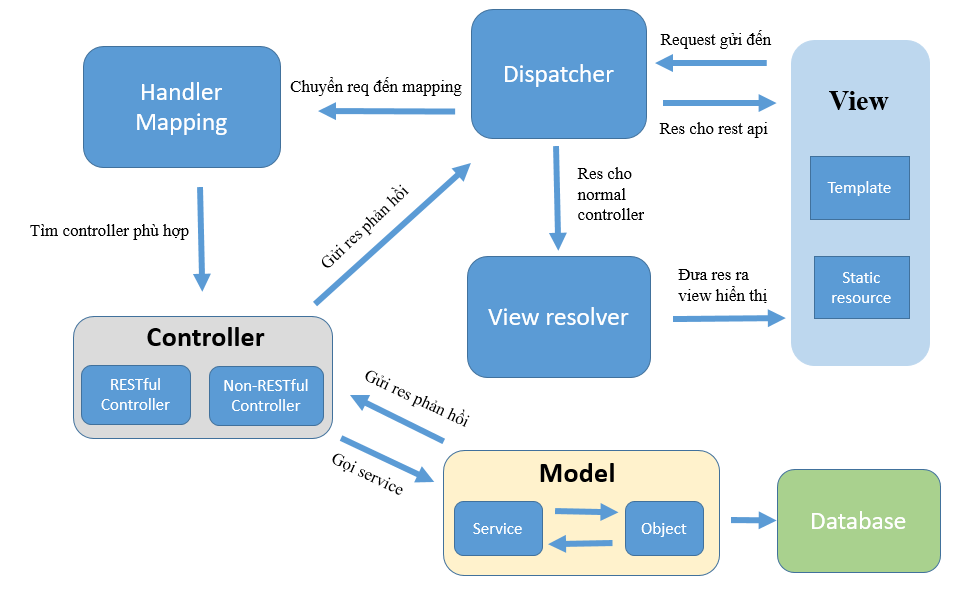
* Không phải mã nguồn mở
* Chỉ hoạt động trên CSDL NoSQL
* Lập trình viên không được truy cập mã nguồn
* Truy vấn chậm
* Phạm vi hoạt động bị hạn chế ở một vài quốc gia
* Một vài dịch vụ cần trả phí, chi phí dành cho Firebase tương đối lớn và không ổn định
* Thiếu Dedicated Servers và cả hợp đồng doanh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

# Phát triển và triển khai ứng dụng

## Thiết kế kiến trúc

### Lựa chọn kiến trúc phần mềm

Mô hình MVC (Model – View – Controller) là một kiến trúc xây dựng phần mềm khá phổ biến. Theo mô hình này, mỗi ứng dụng sẽ được tách thành ba phần riêng biệt, đảm nhận các nhiệm vụ khác nhau. Model xử lý các truy vấn dữ liệu từ database; View tạo ra các giao diện bằng dữ liệu gửi từ Controller; Controller là nơi nơi điều hướng dữ liệu cũng như xử lý các nghiệp vụ logic, kết nối giữa Model và View.



Hình 8 Mô hình MVC áp dụng trong ứng dụng

Trong xây dựng ứng dụng đặt lịch khám bệnh, mô hình MVC được thể hiện theo trình tự sau:

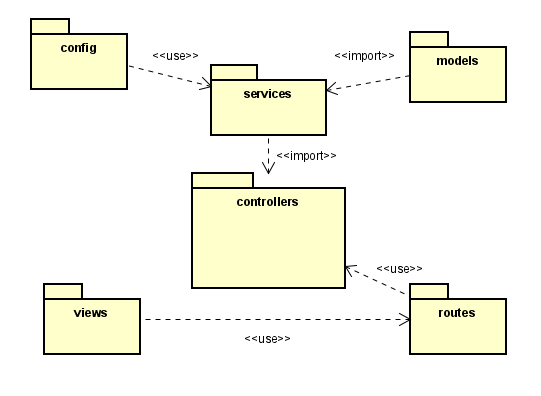
Người dùng truy vào một chức năng (tương ứng với một URL) của ứng dụng sẽ tạo ra một lời gọi tới server. Dipatcher tiếp nhận yêu cầu, chuyển tiếp tới Mapping để tìm ra Controller phù hợp. Controller sẽ chịu trách điều hướng dữ liệu, gọi tới Model để yêu cầu dữ liệu. Tại đây, các Service sẽ được dùng tới, tham chiếu tới các đối tượng model để truy xuất dữ liệu từ database. Sau khi đã có dữ liệu, Model sẽ gửi phản hồi cho Controller. Controller lại chuyển tiếp về cho Dipatcher. Trong trường hợp request được gọi từ REST API, dữ liệu sẽ được trả trực tiếp về View, sau đấy render lên dữ liệu cho người dùng. Còn nếu request được phản hồi từ Controller không sử dụng REST API, View sẽ nhận dữ liệu phản hồi từ Controller để render lại giao diện người dùng.

Sử dụng mô hình MVC trong thiết kế kiến trúc có những ưu điểm sau:

1. Các chức năng ứng dụng được tách riêng biệt ra các bộ phận khác nhau, điều này mang tới sự rõ ràng, rành mạch trong kiến trúc xây dựng.
2. Đảm bảo tính tái sử dụng cũng như phân chia nhiệm vụ cụ thể của từng bộ phận trong ứng dụng.

Dễ bảo trì và mở rộng ứng dụng trong tương lai

### Thiết kế tổng quan



Hình 9 Biểu đồ phụ thuộc gói

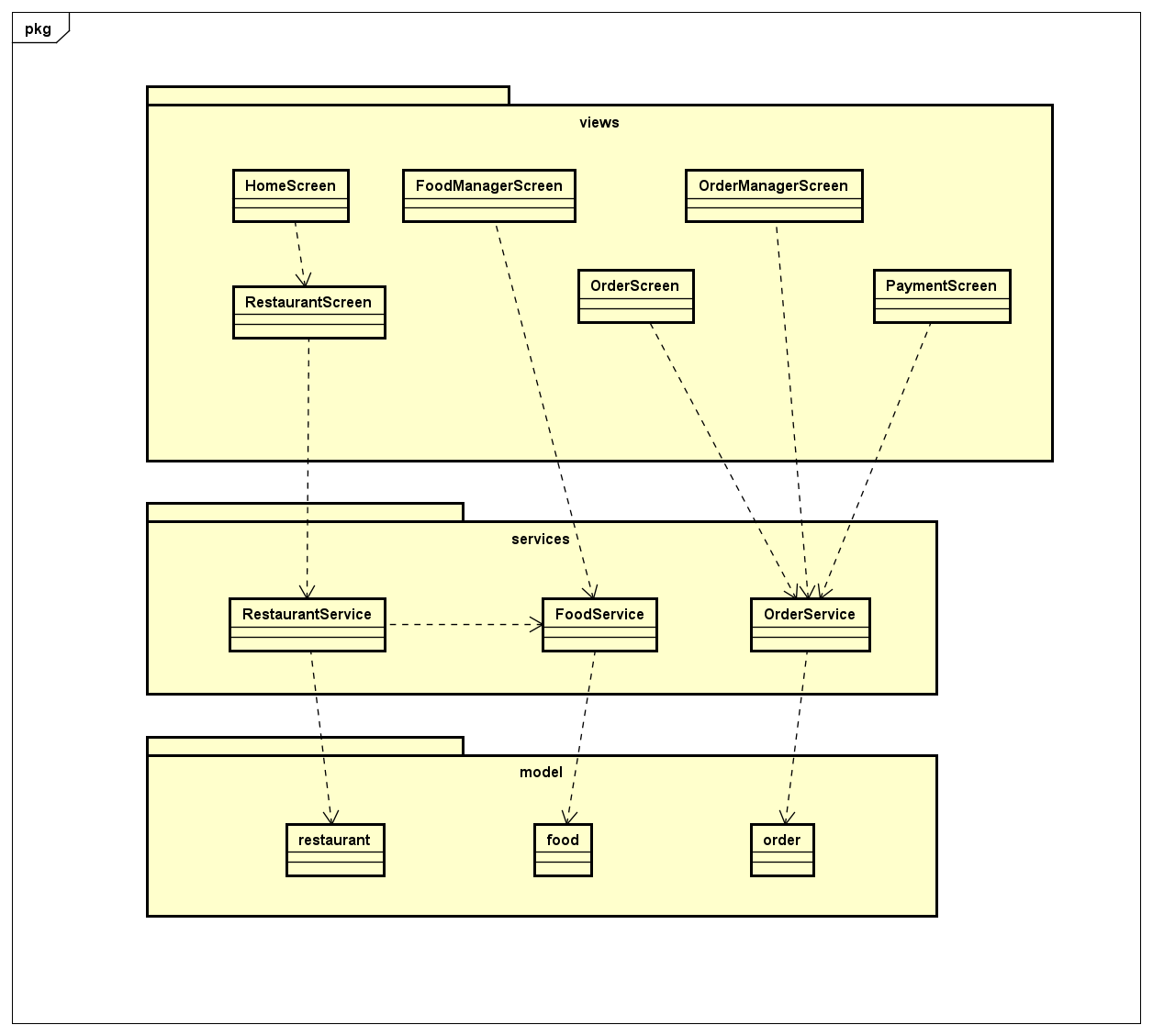
Sự phụ thuộc gói của ứng dụng được thể hiện qua Hình 9, trong đó:

* Gói view: là nơi lưu trữ các template để hiện thị dữ liệu. Dữ liệu hiển thị được nhận từ API hoặc nhận trực tiếp từ các controller.
* Gói routes: chia thành web và api. Các route web là những route chịu trách nhiệm cho hiển thị dữ liệu từ controller. Các route api sẽ nhận dữ liệu từ controller, sau đấy hiển thị lên view.
* Gói controller: xử lý các request từ routes. Các dữ liệu nhận từ request sẽ được kiểm tra, xử lý trước khi chuyển cho Service.
* Gói service: Nhận dữ liệu từ controller, sử dụng dữ liệu từ config và model (nếu cần thiết), chịu trách nhiệm truy vấn và xử lý đối với cơ sở dữ liệu.

Ngoài các gói chính trên, khi xây dựng ứng dụng đặt lịch khám bệnh, còn có thêm các gói: lang (hiển thị ngôn ngữ), helper (các hàm dùng chung), migration, seeder (tạo bảng và dữ liệu mẫu cho database).

### Thiết kế chi tiết gói

Hình 2 thể hiện thiết kế chi tiết gói cho nghiệp vụ đặt hàng. Có 3 gói là views, service, model, 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,…; service là package chứa các lớp điều khiển có chức năng nhận yêu cầu của lớp bao và tương tác với firebase; model là package chứa các lớp thực thể tương ứng với các collections trong firebase.



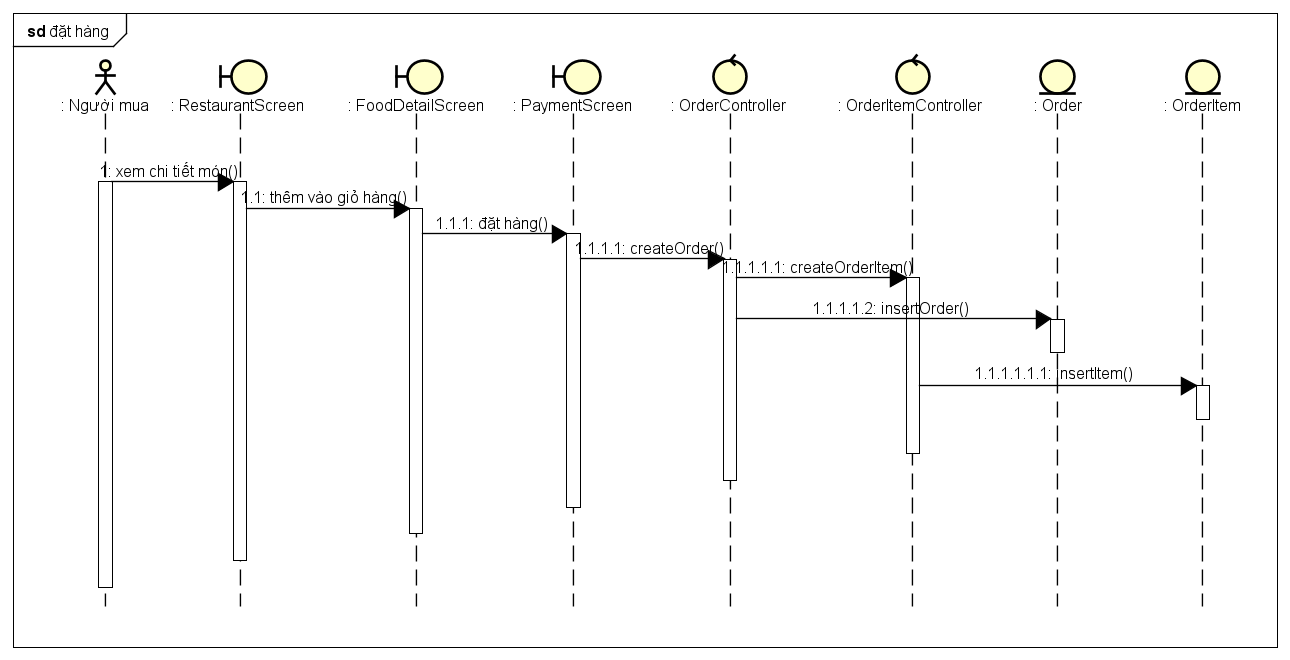
Hình 10 Thiết kế gói

## Thiết kế chi tiết

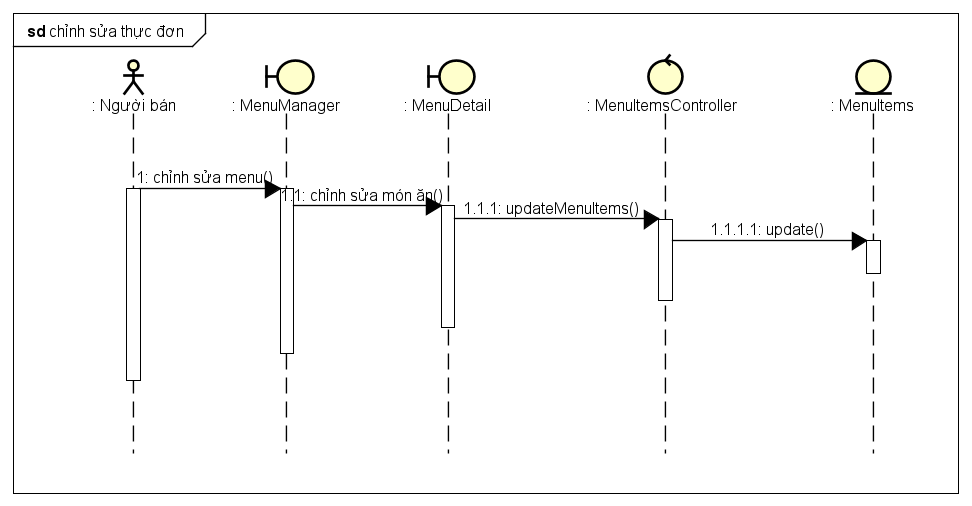
### Thiết kế giao diện

Phần này có độ dài từ hai đến ba trang. Sinh viên đặc tả thông tin về màn hình mà ứng dụng của mình hướng tới, bao gồm độ phân giải màn hình, kích thước màn hình, số lượng màu sắc hỗ trợ, v.v. Tiếp đến, sinh viên đưa ra các thống nhất/chuẩn hóa của mình khi thiết kế giao diện như thiết kế nút, điều khiển, vị trí hiển thị thông điệp phản hồi, phối màu, v.v. Sau cùng sinh viên đưa ra một số hình ảnh minh họa **thiết kế giao diện cho các chức năng quan trọng nhất**. Lưu ý, sinh viên không nhầm lẫn giao diện thiết kế với giao diện của sản phẩm sau cùng.

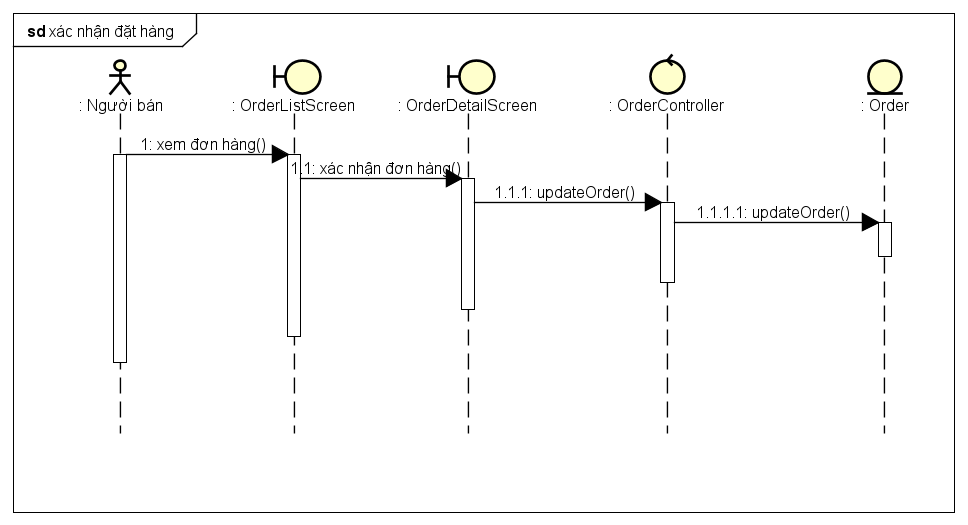
### Thiết kế lớp



Hình 11 Biểu đồ trình tự usecase “Đặt hàng”



Hình 12 Biểu đồ trình tự usecase “Chỉnh sửa thực đơn”



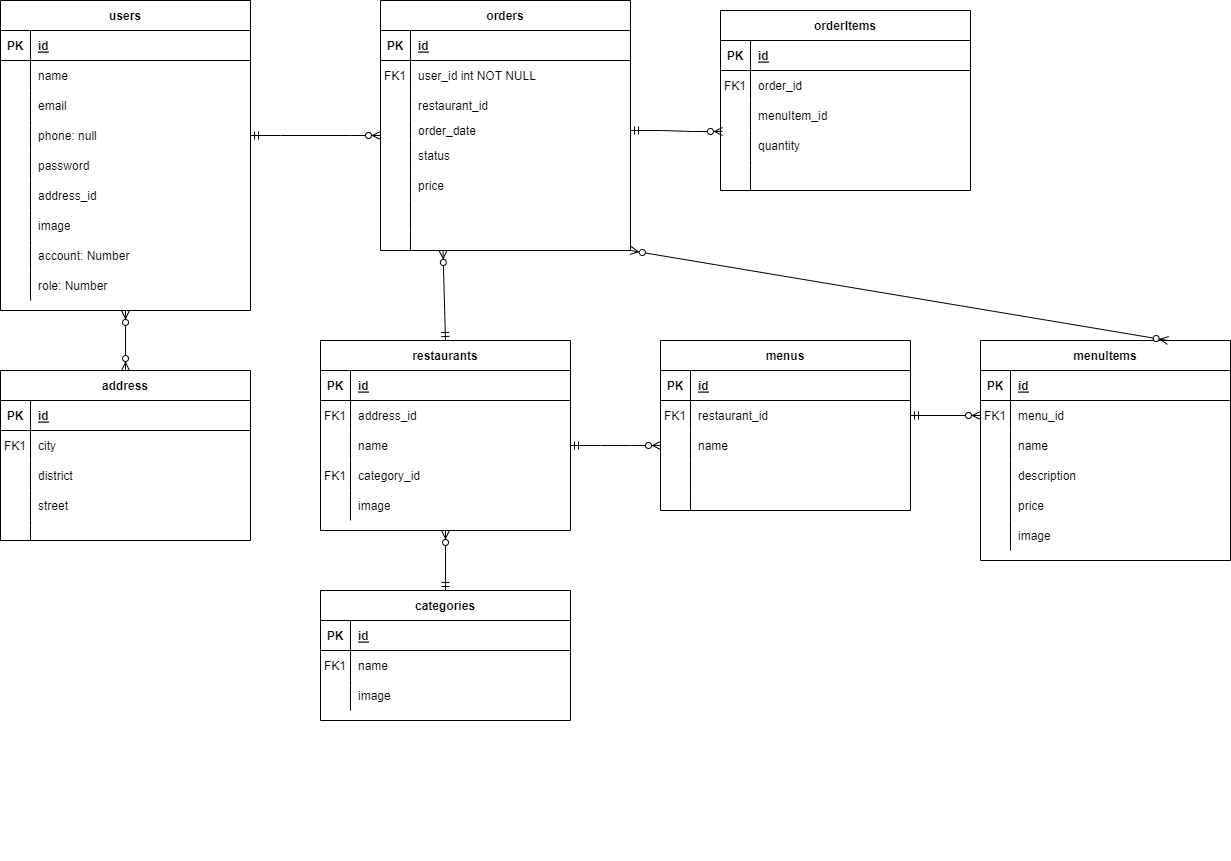
Hình 13 Biểu đồ trình tự usecase “Xác nhận đặt hàng”

### Thiết kế cơ sở dữ liệu

Thiết kế cơ sở dữ liệu cho hệ thống giao đồ ăn được thể hiện ở Hình 19.

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.



Hình 14 Thiết kế cơ sở dữ liệu cho hệ thống giao đồ ăn

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 14**. Cơ sở dữ liệu gồm có 8 collections đó là users, addresses, categories, restaurants, menus, menuItems, orders, orderItems. Chi tiết của các trường trong các collections em sẽ trình bày cụ thể ở **Bảng 15**.

**Bảng 15** Thiết kế chi tiết cơ sở dữ liệu của ứng dụng giao đồ ăn

|  |  |  |  |
| --- | --- | --- | --- |
| **Collection** | **Tên trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
| users | id | String | ID của ứng dụng quản lý chứa khách hàng. |
| name | String | Loại khách hàng (Anonymous, Facebook, Zalo, Viber,  Telegram). |
| email | String | Thông tin chung của khách hàng (fullname, dateOfBirth, gender, avatar, gender). |
|  | password | String |  |
|  | image | String |  |
|  | role | Number |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Collection** | **Tên trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
| addresses | id | String | ID của ứng dụng quản lý chứa khách hàng. |
| city | String | Loại khách hàng (Anonymous, Facebook, Zalo, Viber,  Telegram). |
| district | String | Thông tin chung của khách hàng (fullname, dateOfBirth, gender, avatar, gender). |
|  | street | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Collection** | **Tên trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
| categories | id | String | ID của ứng dụng quản lý chứa khách hàng. |
| name | String | Loại khách hàng (Anonymous, Facebook, Zalo, Viber,  Telegram). |
| title | String | Thông tin chung của khách hàng (fullname, dateOfBirth, gender, avatar, gender). |
|  | image | String |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Collection** | **Tên trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
| restaurants | id | String | ID của ứng dụng quản lý chứa khách hàng. |
| address\_id | String | Loại khách hàng (Anonymous, Facebook, Zalo, Viber,  Telegram). |
| category\_id | String | Thông tin chung của khách hàng (fullname, dateOfBirth, gender, avatar, gender). |
|  | image | String |  |
|  | name |  |  |
|  | user\_id |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Collection** | **Tên trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
| menus | id | String | ID của ứng dụng quản lý chứa khách hàng. |
| restaurant\_id | String | Loại khách hàng (Anonymous, Facebook, Zalo, Viber,  Telegram). |
| name | String | Thông tin chung của khách hàng (fullname, dateOfBirth, gender, avatar, gender). |

|  |  |  |  |
| --- | --- | --- | --- |
| **Collection** | **Tên trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
| menuItems | id | String | ID của ứng dụng quản lý chứa khách hàng. |
| menu\_id | String | Loại khách hàng (Anonymous, Facebook, Zalo, Viber,  Telegram). |
| name | String | Thông tin chung của khách hàng (fullname, dateOfBirth, gender, avatar, gender). |
|  | image | String |  |
|  | price | Number |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Collection** | **Tên trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
| orders | id | String | ID của ứng dụng quản lý chứa khách hàng. |
| user\_id | String | Loại khách hàng (Anonymous, Facebook, Zalo, Viber,  Telegram). |
| restaurant\_id | String | Thông tin chung của khách hàng (fullname, dateOfBirth, gender, avatar, gender). |
|  | order\_date | String |  |
|  | status | Number |  |
|  | price | Number |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Collection** | **Tên trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
| orderItems | id | String | ID của ứng dụng quản lý chứa khách hàng. |
| menuItem\_id | String | Loại khách hàng (Anonymous, Facebook, Zalo, Viber,  Telegram). |
| order\_id | String | Thông tin chung của khách hàng (fullname, dateOfBirth, gender, avatar, gender). |
|  | quantity | String |  |

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

Sinh viên liệt kê các công cụ, ngôn ngữ lập trình, API, thư viện, IDE, công cụ kiểm thử, v.v. mà mình sử dụng để phát triển ứng dụng. Mỗi công cụ phải được chỉ rõ phiên bản sử dụng. SV nên kẻ bảng mô tả tương tự như Bảng 1. Nếu có nhiều nội dung trình bày, sinh viên cần xoay ngang bảng.

**Bảng 1** Danh sách thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| IDE lập trình | Eclipse Oxygen 64 bit | http://www.eclipse.org/ |
| v.v. | v.v. | v.v. |

### Kết quả đạt được

Sinh viên trước tiên mô tả kết quả đạt được của mình là gì, ví dụ như các sản phẩm được đóng gói là gì, bao gồm những thành phần nào, ý nghĩa, vai trò?

Sinh viên cần thống kê các thông tin về ứng dụng của mình như: số dòng code, số lớp, số gói, dung lượng toàn bộ mã nguồn, dung lượng của từng sản phẩm đóng gói, v.v. Tương tự như phần liệt kê về công cụ sử dụng, sinh viên cũng nên dùng bảng để mô tả phần thông tin thống kê này.

### Minh hoạ các chức năng chính

Sinh viên lựa chọn và đưa ra màn hình cho các chức năng chính, quan trọng, và thú vị nhất. Mỗi giao diện cần phải có lời giải thích ngắn gọn. Khi giải thích, sinh viên có thể kết hợp với các chú thích ở trong hình ảnh giao diện.

## Kiểm thử

Phần này có độ dài từ hai đến ba trang. Sinh viên thiết kế các trường hợp kiểm thử cho hai đến ba chức năng quan trọng nhất. Sinh viên cần chỉ rõ các kỹ thuật kiểm thử đã sử dụng. Chi tiết các trường hợp kiểm thử khác, nếu muốn trình bày, sinh viên đưa vào phần phụ lục.

Sinh viên sau cùng tổng kết về số lượng các trường hợp kiểm thử và kết quả kiểm thử. Sinh viên cần phân tích lý do nếu kết quả kiểm thử không đạt.

## Triển khai

Sinh viên trình bày mô hình và/hoặc cách thức triển khai thử nghiệm/thực tế. Ứng dụng của sinh viên được triển khai trên server/thiết bị gì, cấu hình như thế nào. Kết quả triển khai thử nghiệm nếu có (số lượng người dùng, số lượng truy cập, thời gian phản hồi, phản hồi người dùng, khả năng chịu tải, các thống kê, v.v.)

# Các giải pháp và đóng góp nổi bật

Chương này có độ dài tối thiểu 5 trang, tối đa không giới hạn. Sinh viên cần trình bày tất cả những nội dung đóng góp mà mình thấy tâm đắc nhất trong suốt quá trình làm ĐATN. Đó có thể là một loạt các vấn đề khó khăn mà sinh viên đã từng bước giải quyết được, là giải thuật cho một bài toán cụ thể, là giải pháp tổng quát cho một lớp bài toán, hoặc là mô hình/kiến trúc hữu hiệu nào đó được sinh viên thiết kế.

Chương này là **cơ sở quan trọng** để các thầy cô đánh giá sinh viên. Vì vậy, sinh viên cần phát huy tính sáng tạo, khả năng phân tích, phản biện, lập luận, tổng quát hóa vấn đề và tập trung viết cho thật tốt.

Mỗi giải pháp hoặc đóng góp của sinh viên cần được trình bày trong một mục độc lập bao gồm ba mục con: (i) dẫn dắt/giới thiệu về bài toán/vấn đề, (ii) giải pháp, và (iii) kết quả đạt được (nếu có).

Sinh viên lưu ý **không trình bày lặp lại nội dung**. Những nội dung đã trình bày chi tiết trong các chương trước không được trình bày lại trong chương này. Vì vậy, với nội dung hay, mang tính đóng góp/giải pháp, sinh viên chỉ nên tóm lược/mô tả sơ bộ trong các chương trước, đồng thời tạo tham chiếu chéo tới đề mục tương ứng trong Chương 5 này. Chi tiết thông tin về đóng góp/giải pháp được trình bày trong mục đó.

Ví dụ, trong Chương 4, sinh viên có thiết kế được kiến trúc đáng lưu ý gì đó, là sự kết hợp của các kiến trúc MVC, MVP, SOA, v.v. Khi đó, sinh viên sẽ chỉ mô tả ngắn gọn kiến trúc đó ở Chương 4, rồi thêm các câu có dạng: “Chi tiết về kiến trúc này sẽ được trình bày trong phần 5.1”.

# Kết luận và hướng phát triển

## Kết luận

Sinh viên so sánh kết quả nghiên cứu hoặc sản phẩm của mình với các nghiên cứu hoặc sản phẩm tương tự.

Sinh viên phân tích trong suốt quá trình thực hiện ĐATN, mình đã làm được gì, chưa làm được gì, các đóng góp nổi bật là gì, và tổng hợp những bài học kinh nghiệm rút ra nếu có.

## Hướng phát triển

Trong phần này, sinh viên trình bày định hướng công việc trong tương lai để hoàn thiện sản phẩm hoặc nghiên cứu của mình.

Trước tiên, sinh viên trình bày các công việc cần thiết để hoàn thiện các chức năng/nhiệm vụ đã làm. Sau đó sinh viên phân tích các hướng đi mới cho phép cải thiện và nâng cấp các chức năng/nhiệm vụ đã làm.

# Tài liệu tham khảo

Lưu ý: Sinh viên không được đưa bài giảng/slide, các trang Wikipedia, hoặc các trang web thông thường làm tài liệu tham khảo.

Một trang web được phép dùng làm tài liệu tham khảo **chỉ khi** nó là công bố chính thống của cá nhân hoặc tổ chức nào đó. Ví dụ, trang web đặc tả ngôn ngữ XML của tổ chức W3C https://www.w3.org/TR/2008/REC-xml-20081126/ là TLTK hợp lệ.

Có năm loại tài liệu tham khảo mà sinh viên phải tuân thủ đúng quy định về cách thức liệt kê thông tin như sau. Lưu ý: các phần văn bản trong cặp dấu < > dưới đây chỉ là hướng dẫn khai báo cho từng loại tài liệu tham khảo; sinh viên cần xóa các phần văn bản này trong ĐATN của mình.

<**Bài báo đăng trên tạp chí khoa học**: Tên tác giả, tên bài báo, tên tạp chí, volume, từ trang đến trang (nếu có), nhà xuất bản, năm xuất bản>

1. Hovy E. H., Automated Discourse Generation Using Discourse Structure Relations, Artificial Intelligence, Elsevier Science Publishers, 63: 341-385, 1993.

<**Sách**: Tên tác giả, tên sách, volume (nếu có), lần tái bản (nếu có), nhà xuất bản, năm xuất bản>

1. Peterson L. L. and Davie B. S., Computer Networks: A Systems Approach, 2nd ed., Mogran-Kaufmann, 1999.
2. Nguyễn Thúc Hải, Mạng máy tính và các hệ thống mở, Nhà xuất bản giáo dục, 1999.

<**Tập san Báo cáo Hội nghị Khoa học**: Tên tác giả, tên báo cáo, tên hội nghị, ngày (nếu có), địa điểm hội nghị, năm xuất bản>

1. Poesio M. and Di Eugenio B., Discourse Structure and Anaphoric Accessibility, In Proc. of the ESSLLI Workshop on Information Structure, Discourse Structure and Discourse Semantics, Helsinki, 2001.