

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM



BÁO CÁO

PHẦN MỀM QUẢN LÝ CHUỖI NHÀ HÀNG

Dưới sự Hướng dẫn
Thầy Huỳnh Tuấn Anh

Sinh viên thực hiện

1. Phan Thanh Duy	16520303
2. Lê Hồng Hiến	16520361
3. Nguyễn Trung Nguyên	16520848

Lớp: SE346.K11

TP. Hồ Chí Minh

Tháng 12/2019

LỜI CẢM ƠN

Trong suốt quá trình học môn Lập trình trên thiết bị di động trước tiên nhóm chúng em xin gửi lời cảm ơn chân thành nhất đến Thầy Huỳnh Tuấn Anh. Chính nhờ sự giảng dạy nhiệt tình, tận tâm của Thầy cùng những kiến thức được Thầy chia sẻ trên lớp cùng những lời góp ý, phản biện mà nhóm có thể hoàn thành ra được một sản phẩm khá hoàn thiện.

Bên cạnh đó, nhóm cũng xin gửi lời cảm ơn chân thành đến tất cả các anh chị, các bạn trong và ngoài lớp đã kịp thời hỗ trợ, chia sẻ tài liệu và đưa ra những nhận xét, góp ý chân thành nhằm giúp đồ án được hoàn thiện hơn. Đặc biệt là các anh, chị K10 đã nhiệt tình chia sẻ kinh nghiệm và giải đáp các thắc mắc của nhóm trong suốt quá trình thực hiện đồ án.

Tuy vậy, trong quá trình thực hiện đồ án, viết báo cáo việc không tránh khỏi những thiếu sót là điều chắc chắn. Vì thế, nhóm rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy và các bạn học cùng lớp để nhóm có thể rút kinh nghiệm và hoàn thiện đồ án một cách tốt nhất.

Nhóm xin chân thành cảm ơn!

Tp. Hồ Chí Minh, tháng 12 năm 2019

Nhóm thực hiện.

NHẬN XÉT CỦA GIẢNG VIÊN

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	10
1.1 Đặt vấn đề	10
1.2 Giải pháp	11
1.3 Mục tiêu đề tài	11
1.3.1 Tìm hiểu các công nghệ mới	11
1.3.2 Xây dựng phần mềm quản lý chuỗi nhà hàng	12
1.4 Công cụ sử dụng	13
CHƯƠNG 2: PHÁT BIỂU BÀI TOÁN	15
2.1 Khảo sát hiện trạng	15
2.1.1 Sơ đồ tổ chức	15
2.1.2 Nghiệp vụ	15
a) Quy định về thời gian hoạt động của nhà hàng	15
b) Quy định về giá cả món ăn	15
c) Quy trình đặt món tại nhà hàng	16
d) Quy trình đặt món online	16
2.2 Yêu cầu chức năng	17
2.2.1 Yêu cầu lưu trữ	17
2.2.2 Yêu cầu về tính năng	17
2. Yêu cầu phi chức năng	18
CHƯƠNG 3: MÔ HÌNH USE-CASE	19
3.1 Sơ đồ Use-case	19
3.1.1 Use-case chung	19
3.1.2 Use-case Customer	20
3.1.3 Use-case Staff	21
3.1.4 Use-case Chef	21
3.1.5 Use-case Shipper	22
3.1.6 Use-case Ware manager	23
3.1.7 Use-case Accountant	23
3.1.8 Use-case Store manager	24
3.1.9 Use-case Admin	25
3.2 Danh sách các Actor	25
3.3 Danh sách các Use-case	26
3.4 Đặc tả Use-case	28

3.4.1 Đặc tả Use-case View history	28
3.4.2 Đặc tả Use-case View history detail	29
3.4.3 Đặc tả Use-case Change avatar & information	30
3.4.4 Đặc tả Use-case Change password	31
3.4.5 Đặc tả Use-case Change addresses	32
3.4.6 Đặc tả Use-case Submit new delivery bill	33
3.4.7 Đặc tả Use-case Look for daily dish	34
3.4.8 Đặc tả Use-case Pick bill to delivery	35
3.4.9 Đặc tả Use-case Submit new bill	36
3.4.10 Đặc tả Use-case Confirm bill is paid	37
3.4.11 Đặc tả Use-case Add dish to unpaid bill	38
3.4.12 Đặc tả Use-case Remove dish from unpaid bill	39
3.4.13 Đặc tả Use-case Pick pill to prepare	40
3.4.14 Đặc tả Use-case Confirm dish prepared	41
3.4.15 Đặc tả Use-case Confirm dish is out of stock	42
3.4.16 Đặc tả Use-case Confirm delivery bill is paid	43
3.4.17 Đặc tả Use-case Pick prepared bill to ship	44
3.4.18 Đặc tả Use-case Create import bill	45
3.4.19 Đặc tả Use-case Create end-day report	46
3.4.20 Đặc tả Use-case View analysis	47
3.4.21 Đặc tả Use-case Export / Print revenue report	48
3.4.22 Đặc tả Use-case Export / Print expenditure report	49
3.4.23 Đặc tả Use-case Edit import bill / Edit end-day report	50
3.4.24 Đặc tả Use-case Create Warehouse	51
3.4.25 Đặc tả Use-case Edit warehouse	52
3.4.26 Đặc Use-case Delete warehouse	53
3.4.27 Đặc tả Use-case Create voucher code / discount code	54
3.4.28 Đặc tả Use-case Edit voucher / discount code information	56
3.4.29 Đặc tả Use-case Delete voucher / discount code	57
3.4.30 Đặc tả Use-case Create user / customer	58
3.4.31 Đặc tả Use-case Edit user / customer	59
3.4.32 Đặc tả Use-case Delete use / customer	60
3.4.33 Đặc tả Use-case Create discount campaign	61
3.4.34 Đặc tả Use-case Edit discount campaign	62

3.4.35 Đặc tả Use-case Delete discount campaign	63
3.4.36 Đặc tả Use-case Add dishes to list dished of store	64
3.4.37 Đặc tả Use-case Remove dish from lish dished of store	65
3.4.38 Đặc tả Use-case Add dished to list daily dishes	66
3.4.39 Đặc tả Use-case Remove dish from list daily dishes	67
3.4.40 Đặc tả Use-case Create store / role / dish	68
3.4.41 Đặc tả Use-case Edit store / role / dish	69
3.4.42 Đặc tả Use-case Delete store / role / dish	70
CHƯƠNG 4: PHÂN TÍCH	71
CHƯƠNG 5: THIẾT KẾ DỮ LIỆU	72
5.1 Danh sách các bảng dữ liệu	74
5.2 Mô tả chi tiết từng bảng dữ liệu	76
5.2.1 Bảng User	76
5.2.2 Bảng customer	77
5.2.3 Role	78
5.2.4 Address	79
5.2.5 Bill	80
5.2.6 Delivery Bill	81
5.2.7 Bill History	82
5.2.8 Dish	82
5.2.9 DailyDish	83
5.2.10 Store	83
5.2.11 Warehouse	84
5.2.12 Daily Report	84
5.2.13 Stock_Warehouse	85
5.2.14 Stock_DailyReport	85
5.2.15 Stock	85
5.2.16 Stock Import Bill	86
5.2.17 ImportBill	86
5.2.18 DiscountCampaign	87
5.2.19 Store_DisCam	87
5.2.20 DisCam_Dish	87
5.2.21 Store_DisCode	88

5.2.22 DiscountCode	88
5.2.23 Store_VouCode	89
5.2.24 VoucherCode	89
CHƯƠNG 6: KIẾN TRÚC ỨNG DỤNG	90
6.1 Kiến trúc hệ thống	90
6.2 Mô tả chi tiết từng thành phần trong hệ thống	91
6.2.1 Client	91
6.2.2 Web Server	92
6.2.3 Web Client	93
CHƯƠNG 7: THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG	94
7.1 Thiết kế giao diện trên mobile (customer app)	94
7.1.1 Danh sách các màn hình	94
7.1.2 Mô tả từng màn hình	96
7.1.2.1 Màn hình đăng nhập	96
7.1.2.2 Màn hình đăng ký	97
7.1.2.3 Màn hình sửa mật khẩu	98
7.1.2.4 Màn hình sửa thông tin cá nhân	99
7.1.2.5 Màn hình thông tin cá nhân	100
7.1.2.6 Màn hình danh sách nhà hàng	101
7.1.2.7 Màn hình danh sách món ăn	102
7.1.2.8 Màn hình thông tin món ăn	103
7.1.2.9 Màn hình giỏ hàng	105
7.1.2.10 Màn hình tạo hóa đơn	106
7.1.2.11 Màn hình thông tin hóa đơn	108
7.1.2.12 Màn hình danh sách hóa đơn của tôi	109
7.2 Thiết kế giao diện trên mobile (staff app)	110
7.2.2 Mô tả từng màn hình	111
7.2.2.1 Màn hình danh sách món ăn hôm nay	111
7.2.2.2 Màn hình thông tin món ăn	113
7.2.2.3 Màn hình giỏ hàng	114
7.2.2.4 Màn hình danh sách hóa đơn và danh sách hóa đơn của bạn	115
7.2.2.5 Trang thông tin cá nhân	116
7.2.2.6 Sửa trang thông tin cá nhân	118

CHƯƠNG 8: CÔNG NGHỆ SỬ DỤNG	119
8.1 Flutter	119
8.1.1 Flutter là gì?	119
8.1.2 Tại sao chọn Flutter?	119
8.1.3 Firebase Storage	120
8.2 ExpressJs Framework	121
8.3 Object Relational Mapping (Orm) - TypeOrm	122
8.3.1 Orm là gì?	122
8.3.2 Tại sao nên chọn Orm	122
8.3.3 Trở ngại khi dùng Orm	123
8.3.4 TypeOrm là gì?	124
8.3.5 Tại sao chọn TypeOrm	124
8.4 GraphQL	125
8.4.1 GraphQL là gì?	125
8.4.2 Đặc điểm của GraphQL	125
8.4.3 Lợi ích của GraphQL	125
8.5 Socket.io	126
8.6 Heroku	127
8.6.1 Heroku là gì?	128
8.6.2 Đặc điểm của heroku	128
8.6.3 Tại sao chọn Heroku	129
CHƯƠNG 9: KẾT QUẢ ĐẠT ĐƯỢC	129
9.1 Kết quả	130
9.2 Hạn chế	130
9.3 Hướng phát triển	130
CHƯƠNG 10: PHÂN CÔNG CÔNG VIỆC	130
REFERENCE	132

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 Đặt vấn đề

Hiện nay, với sự phát triển của xã hội, con người sống vội hơn, nhiều người không có thời gian để tự chuẩn bị cho mình những bữa ăn nên họ chọn ăn ngoài để tiện lợi hơn, tiết kiệm thời gian hơn. Từ những quán cơm bình dân đến những nhà ăn, căn tin trong công ty, trường học hay cao cấp hơn là các nhà hàng được mở ra để phục vụ cho từng nhu cầu, thu nhập, mức sống khác nhau. Tất cả đều thu hút được đông đảo khách hàng ở nhiều phân khúc độ tuổi, tầng lớp khác nhau.

Việc lượng khách hàng càng đông khiến cho việc quản lý cửa hàng bằng hình thức kinh doanh cổ điển khiến khâu quản lý trở nên vô cùng khó khăn, tốn kém, từ đó dẫn đến độ chính xác không cao và không đồng bộ giữa các khâu. Điển hình có những hạn chế sau:

- Tra cứu thông kê, báo cáo doanh thu hằng ngày, hằng tháng gặp nhiều khó khăn.
- Lưu trữ thông tin order của khách hàng, thông tin nguyên liệu... bằng giấy tờ công kênh, không hiệu quả và dễ gây nhầm lẫn.
- Việc quản lý kho gặp nhiều khó khăn, ví dụ như việc khó có thể biết được khi nào thì nguyên liệu sắp hết để xin phiếu chi để nhập thêm hàng mới.
- Bộ phận kế toán khó có thể quản lý thu chi một cách hiệu quả khi các khâu liên quan chưa được đồng bộ.

Để giải quyết vấn đề trên, cần một giải pháp mang tính tổng quát hơn, xây dựng một hệ thống phục vụ công tác quản lý chuỗi nhà hàng. Hệ thống được xây dựng phải là một hệ thống quản lý tích hợp, lưu trữ thông tin tập trung và đáp ứng cho nhiều đối tượng

người dùng. Bên cạnh đó cũng cần chuẩn hóa quy trình các công việc, để việc ứng dụng công nghệ thông tin thực sự có hiệu quả.

1.2 Giải pháp

Để giải quyết vấn đề trên, cần một giải pháp mang tính tổng quát hơn, xây dựng một hệ thống phục vụ công tác quản lý chuỗi nhà hàng. Hệ thống được xây dựng phải là một hệ thống quản lý tích hợp, lưu trữ thông tin tập trung và đáp ứng cho nhiều đối tượng người dùng. Bên cạnh đó cũng cần chuẩn hóa quy trình các công việc, để việc ứng dụng công nghệ thông tin thực sự có hiệu quả.

Vì mỗi nhà hàng yêu cầu xuất báo cáo qua Internet, đồng thời mỗi nhà hàng đang được trang bị hệ thống Wifi tiên tiến: tốc độ cao, ổn định, ... cho khách hàng sử dụng. Do đó để tiết kiệm chi phí chúng ta có thể dùng hệ thống wifi có sẵn ở mỗi nhà hàng.

1.3 Mục tiêu đề tài

Nội dung đề tài gồm 3 phần chính: tìm hiểu về ReactJS, Flutter và xây dựng phần mềm hỗ trợ quản lý chuỗi nhà hàng.

1.3.1 *Tìm hiểu các công nghệ mới*

Mục tiêu đạt được sau dự án là tìm hiểu được các công nghệ mới dưới đây:

- Flutter và Dart: Flutter là một framework để xây dựng ứng dụng trên di động và hỗ trợ đa nền tảng. Với flutter ta chỉ cần sử dụng một ngôn ngữ lập trình duy nhất cho cả front-end và back-end là Dart
- ExpressJS: Công nghệ để làm web server đơn giản và nhanh chóng trên nền tảng nodeJs.
- TypeOrm: Công nghệ Orm trên nền tảng Nodejs và hỗ trợ tốt cho ngôn ngữ lập trình Typescript.
- NodeJs và Typescript: Nodejs là một framework viết bằng Javascript, nhưng javascript thì khá khó và tính hướng đối tượng không cao. Nên ngày nay

người ta phát triển thêm Typescript là ngôn ngữ mở rộng từ Javascript hỗ trợ hướng đối tượng tốt hơn.

- Socket.io: Để xây dựng ứng dụng realtime.
- GraphQL: Để thao tác và truy vấn dữ liệu nguồn mở cho API.

1.3.2 *Xây dựng phần mềm quản lý chuỗi nhà hàng*

Xây dựng phần mềm quản lý chuỗi nhà hàng cần đáp ứng các yêu cầu sau:

- **Chức năng dành cho người quản lý tại nhà hàng:**
 - Quản lý khâu đặt món.
 - Thêm món ăn mới vào hệ thống.
 - Định giá các món ăn.
 - Xem các báo cáo hằng ngày, hằng tuần, hằng tháng của hệ thống, thu ngân
- **Chức năng dành cho phục vụ:**
 - Order món cho khách
 - Thanh toán hóa đơn nếu khách thanh toán bằng tiền mặt
 - Phục vụ món ăn cho khách sau khi đầu đã chuẩn bị xong món ăn.
 - Dọn bàn sau khi khách dùng bữa xong.
- **Chức năng dành cho đầu bếp:**
 - Đầu bếp nhận hóa đơn và tiến hành chuẩn bị món ăn.
 - Báo cáo trạng thái món ăn
 - Báo hết món khi nguyên liệu hết.
- **Chức năng dành cho bộ phận kế toán:**
 - Tham mưu cho ban quản lý về chiến lược kinh doanh.
 - Tính giá hàng tồn kho, giá vốn hàng bán.

- Quản lý về công tác sổ sách: tổng hợp các hoá đơn, quản lý doanh thu.
- Lập báo cáo doanh thu mỗi ngày, mỗi tuần, mỗi tháng.
- Tổ chức thực hiện công tác quản lý dịch vụ phục vụ sinh viên.
- Lập kê khai thuế.

- **Chức năng dành cho bộ phận quản lý kho:**

Quản lý nhập kho:

Xuất hoá đơn nhập kho sau mỗi ngày (dựa theo thống kê từ bộ phận quản lý tồn kho và quyết định nhập xuất kho từ bên kế toán).

Quản lý quá trình nhập kho (kiểm kê, xuất hoá đơn thanh toán, chuyển hoá đơn cho bộ phận kế toán để tiến hành chi trả).

Quản lý tồn kho:

Quản lý số lượng tồn kho của thực phẩm (bao gồm cả đã chế biến và chưa chế biến).

Lập báo cáo tồn kho cho bộ phận quản lý nhập kho và bộ phận kế toán.

1.4 Công cụ sử dụng

Trong quá trình nghiên cứu, thiết kế và triển khai đồ án, nhóm đã có sử dụng một số công cụ, thư viện hỗ trợ việc phát triển đồ án:

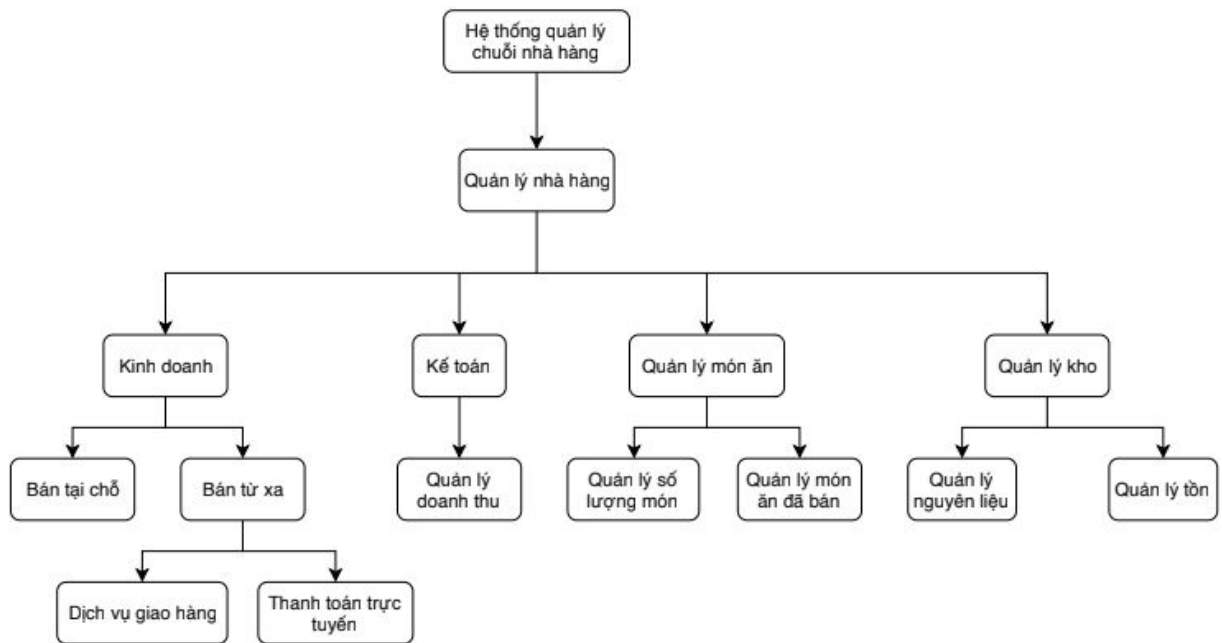
- Draw.io: Công cụ hỗ trợ phân tích và vẽ sơ đồ
- Flutter: Nền tảng hỗ trợ lập trình di động đa nền tảng.
- ExpressJS: Thư viện hỗ trợ triển khai server
- TypeOrm: Thư viện hỗ trợ chuyển đổi dữ liệu giữa các hệ thống khác nhau.
- Heroku: Nền tảng đám mây, hỗ trợ phát triển server.
- Visual studio code: IDE hỗ trợ triển khai server

- IntelliJ IDE: IDE hỗ trợ phát triển android app.

CHƯƠNG 2: PHÁT BIỂU BÀI TOÁN

2.1 Khảo sát hiện trạng

2.1.1 Sơ đồ tổ chức



2.1.2 Nghiệp vụ

a) Quy định về thời gian hoạt động của nhà hàng

- Mỗi nhà hàng sẽ có thời gian mở cửa và đóng cửa riêng
- Quán nghỉ lễ theo các quy định Nhà nước (không thông báo trước)
- Quán có thể nghỉ đột xuất vì lý do nào đó (thông báo trên app online và dán thông báo tại quán ít nhất trước một ngày).
- Quy định về mốc thời gian có thể được tùy chỉnh cho hợp lý bởi ban quản lý.

b) Quy định về giá cả món ăn

- Giá mỗi món ăn thông thường sẽ có giá mặc định.

- Giá các món đặt biệt được bộ phận quản lý đặt trước.
- Lưu ý giá các món ăn thông thường có thể được thay đổi theo yêu cầu của ban quản lý.

c) *Quy trình đặt món tại nhà hàng*

- Khi vào nhà hàng, khách hàng sẽ được nhân viên mang cho một hóa đơn điện tử gồm các món ăn ngày hôm nay nhà hàng có bán (menu được hiển thị trên điện thoại di động của nhân viên, menu có kèm giá và hình ảnh minh họa).
- Khách hàng tiến hành chọn các món yêu thích (số lượng món phải lớn hơn hoặc bằng số lượng người của bàn đó).
- Khách hàng chọn phương thức thanh toán: Thanh toán trực tuyến hay thanh toán bằng tiền mặt.
- Nếu khách hàng chọn thanh toán tiền mặt thì sẽ đưa tiền mặt cho nhân viên phục vụ.
- Hệ thống sẽ tự cập nhật hoá đơn đồng thời phân chia cho bộ phận quản lý món đã bán.
- Sau khi món ăn hoàn thành thì nhân viên mang thức ăn, tráng miệng, khăn lau, hóa đơn, tiền thừa (nếu có) cho khách hàng.

d) *Quy trình đặt món online*

- Khách hàng tiến hành đăng nhập tài khoản vào app Resman mobile customer (đăng ký trực tiếp thông qua app nếu chưa có tài khoản).
- Trong ứng dụng, khách hàng sẽ thấy được một hoá đơn điện tử gồm các món ăn trong hôm nay nhà hàng có bán (menu có kèm giá và hình ảnh minh họa).
- Khách hàng tiến hành chọn các món ưu thích.

- Khách hàng chọn phương thức thanh toán: thanh toán trực tuyến.
- Hệ thống sẽ tự cập nhật hoá đơn đồng thời phân chia cho bộ phận đầu bếp.
- Sau khi hoàn thành món ăn, đầu bếp sẽ cập nhật trạng thái trên hệ thống là đã hoàn thành món ăn, hệ thống sẽ thông báo cho khách thông qua app là món ăn đã được chuẩn bị.
- Nhân viên sẽ chuyển giao món ăn cho shipper và cập nhật hệ thống là đang giao hàng, hệ thống sẽ thông báo cho khách hàng thông qua app là đang giao hàng.
- Shipper giao hàng cho khách.

2.2 Yêu cầu chức năng

2.2.1 Yêu cầu lưu trữ

Là yêu cầu quan trọng nhất của hệ thống, là cơ sở để thực hiện phần lớn các tính năng của hệ thống. Hệ thống cần lưu trữ những thông tin sau:

- Thông tin về khách hàng: Lưu những thông tin cá nhân về khách hàng như họ tên, ngày sinh, thông tin liên lạc, lịch sử sử dụng dịch vụ, ...
- Thông tin về món ăn: Lưu những thông tin về món ăn như tên món, trạng thái, hình ảnh, nguyên liệu, đánh giá...
- Thông tin về hoá đơn: Lưu trữ chi tiết hoá đơn như ngày lập, trạng thái thanh toán, trạng thái chuẩn bị, khách hàng, danh sách món, ...
- Thông tin về kho: Lưu trữ thông tin nhập xuất kho.

2.2.2 Yêu cầu về tính năng

Ứng dụng bao gồm nhiều lớp tính năng cho nhiều loại người dùng khác nhau, nên được chia thành 2 loại ứng dụng riêng: ứng dụng Web client với đầy đủ tính năng, và ứng dụng di động với 1 vài tính năng cần thiết. Cả 2 loại ứng dụng đều được kết nối với một máy chủ và có khả năng xử lý thời gian thực.

Một vài tính năng cần được đáp ứng:

- Chức năng truy vấn: hệ thống phải đáp ứng được các truy vấn từ người dùng như: tìm món ăn, tìm thông tin cá nhân, tìm lịch sử sử dụng, tìm thông tin hoá đơn, ...
- Chức năng thống kê: giúp quản lý cửa hàng nắm bắt được tình hình chung về việc buôn bán của cửa hàng.
- Chức năng cập nhật: hệ thống phải đảm bảo chức năng cập nhật được thuận tiện và tránh tối đa lỗi do sơ xuất của người dùng.

2. Yêu cầu phi chức năng

Yêu cầu về giao diện

- Giao diện sử dụng ngôn ngữ tiếng Việt, đẹp, thân thiện và dễ sử dụng.
- Tối ưu hoá các thao tác người dùng.

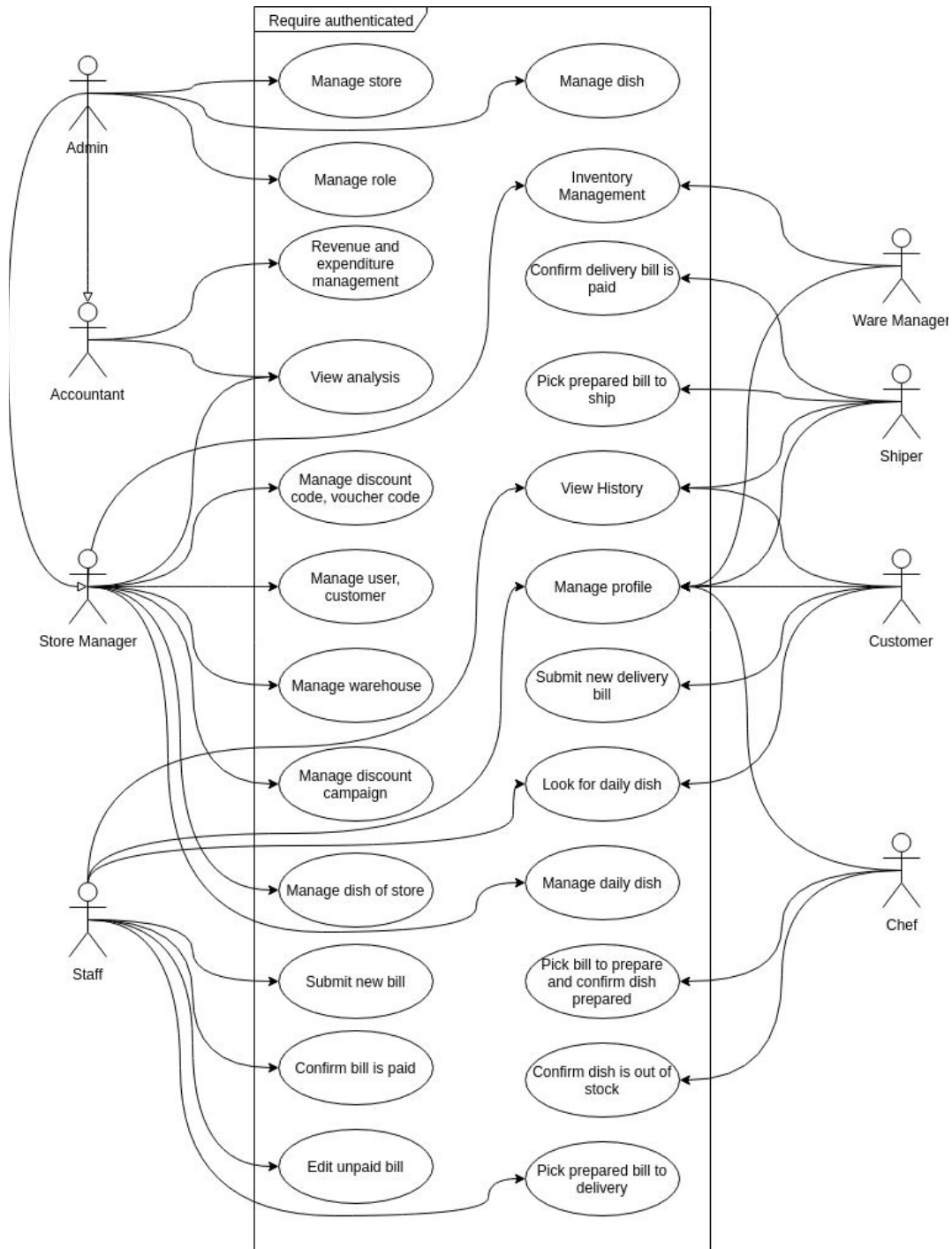
Yêu cầu về hệ thống

- Đối với ứng dụng điện thoại, cần chạy được trên nhiều nền tảng như android, ios.
- Đối với ứng dụng để bàn cần chạy ổn định, không bị lỗi, hoạt động tốt trên các hệ điều hành cũ.
- Đối với máy chủ cần chạy thời gian thực ổn định, không lỗi bảo mật, đảm bảo tốc độ.

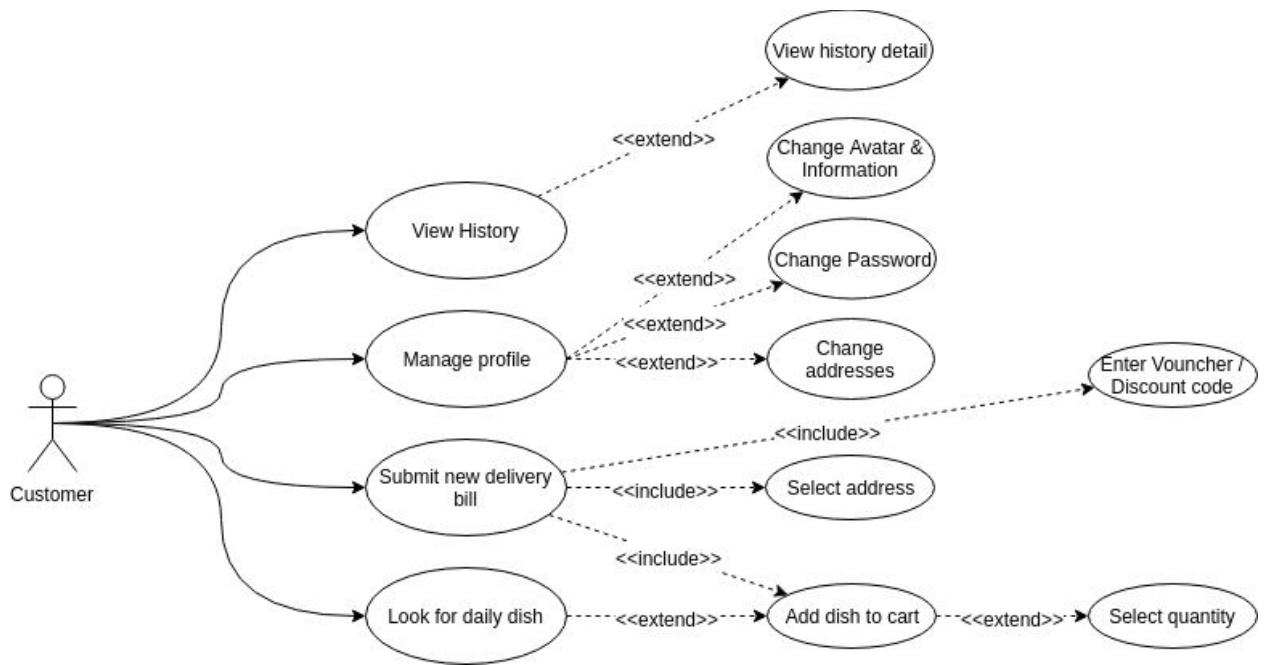
CHƯƠNG 3: MÔ HÌNH USE-CASE

3.1 Sơ đồ Use-case

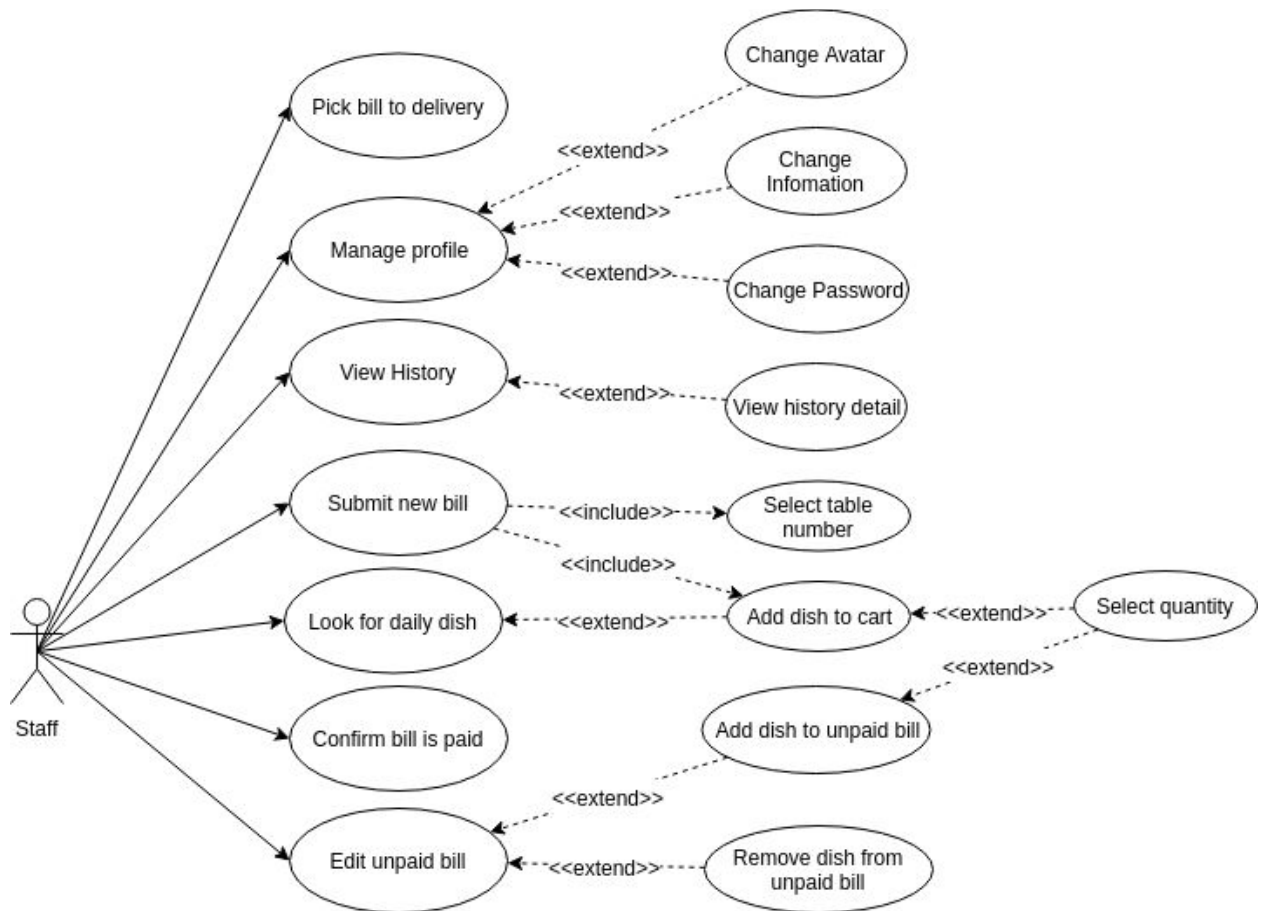
3.1.1 Use-case chung



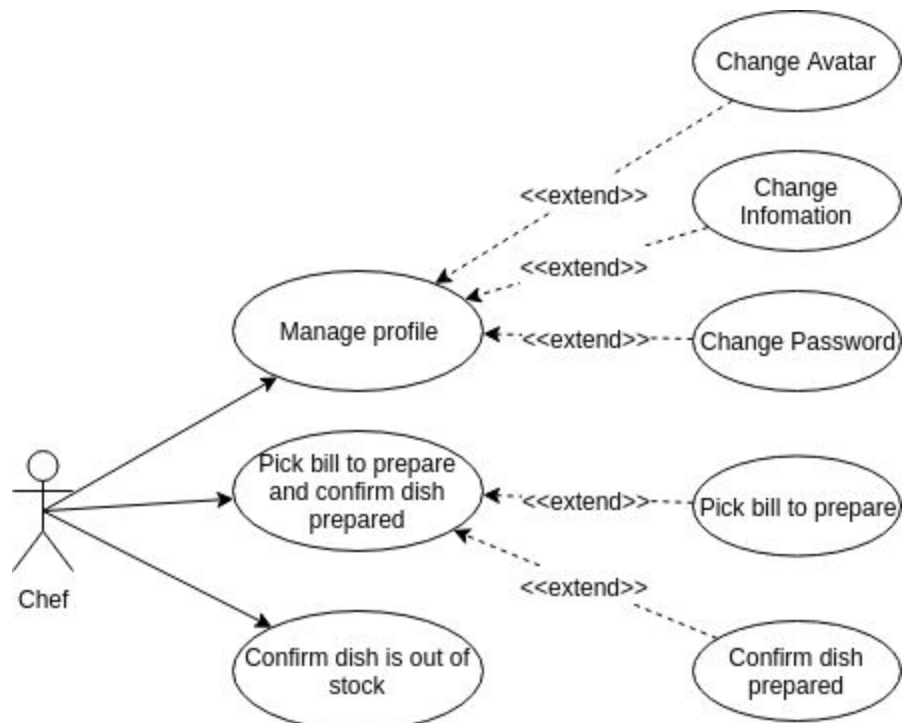
3.1.2 Use-case Customer



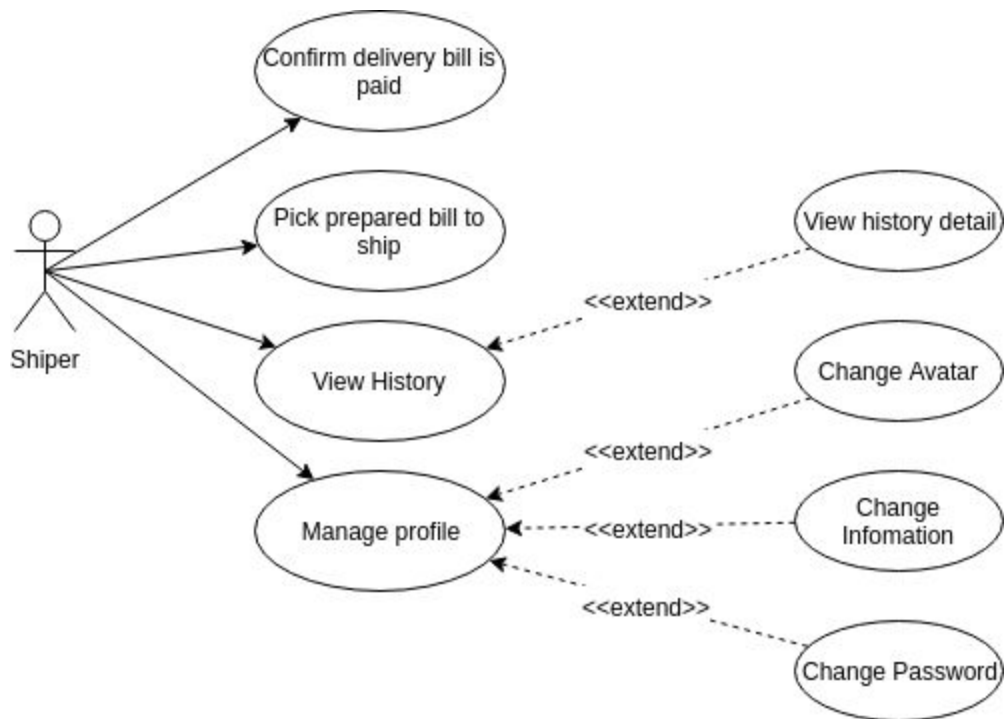
3.1.3 Use-case Staff



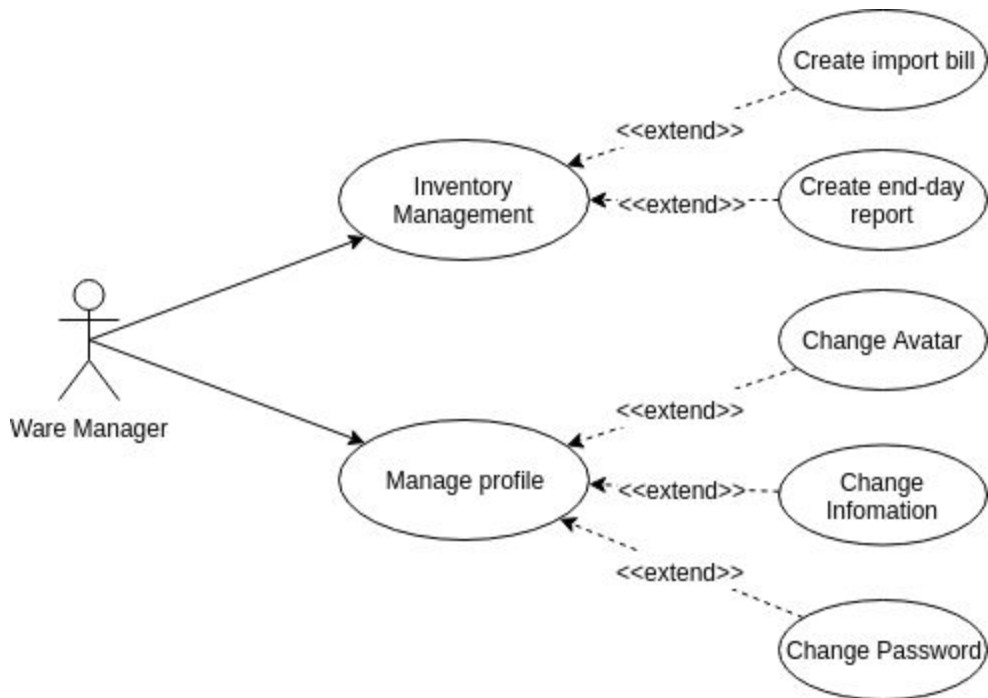
3.1.4 Use-case Chef



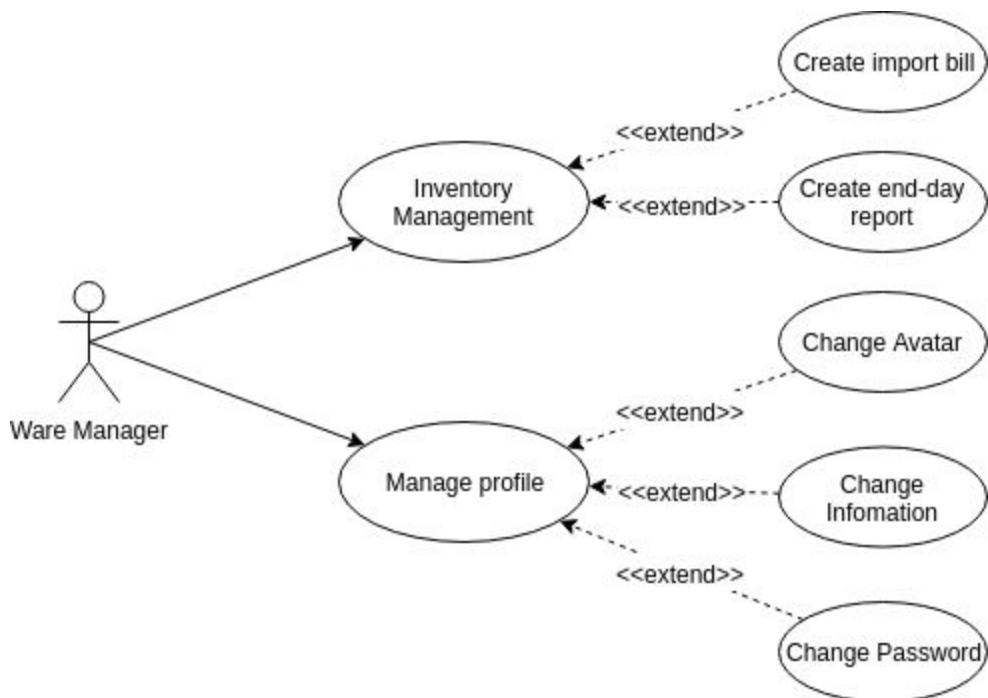
3.1.5 Use-case Shipper



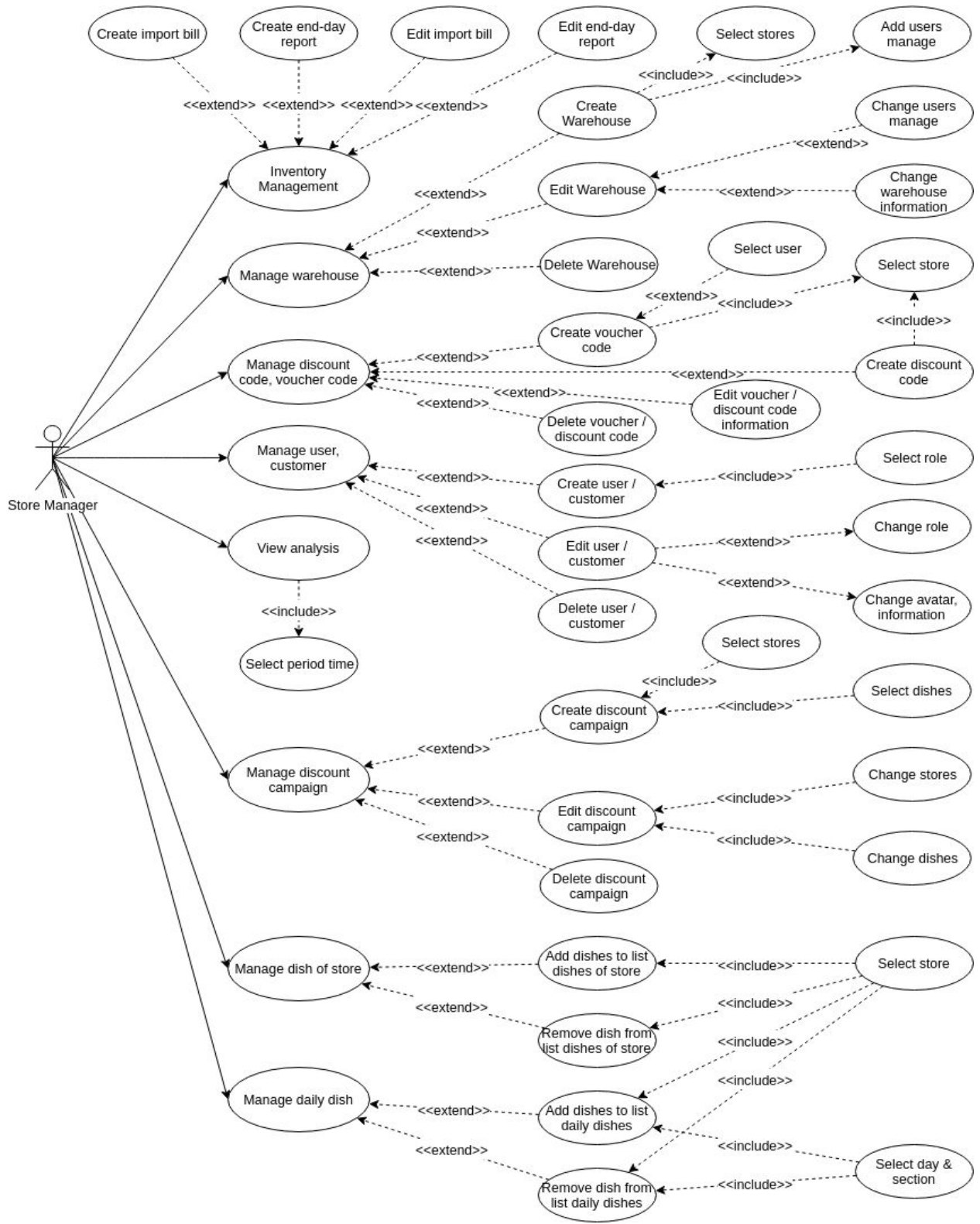
3.1.6 Use-case Ware manager



3.1.7 Use-case Accountant



3.1.8 Use-case Store manager



3.1.9 Use-case Admin

3.2 Danh sách các Actor

STT	Tên Actor	Ý nghĩa/Ghi chú
1	Admin	Quản trị viên, hoặc là chủ của chuỗi nhà hàng, quán ăn
2	Store Manager	Nhân viên cấp cao, người quản lý một cửa hàng, hoặc nhiều cửa hàng.
3	Accountant	Nhân viên kế toán, người quản lý thu chi và xuất hóa đơn báo cáo.
4	Ware Manager	Nhân viên quản lý kho, người tạo hóa đơn nhập kho và kiểm kê kho vào cuối mỗi ngày.
5	Staff	Nhân viên phục vụ, người sẽ mang menu ra khi khách tới quán và phục vụ món ăn ra bàn cũng như thu tiền.
6	Chef	Đầu bếp hoặc người chuẩn bị món ăn. Người sẽ nhận thông tin món ăn từ hóa đơn và chuẩn bị các món ăn đó để phục vụ.
7	Shipper	Nhân viên giao hàng của cửa hàng.
8	Customer	Là những khách hàng sử dụng ứng dụng để đặt món ăn.

3.3 Danh sách các Use-case

STT	Tên Use-case	Ý nghĩa/Ghi chú
1	Manage store	Quản lý thêm xóa sửa thông tin các chi nhánh (cửa hàng).
2	Manage role / dish	Quản lý thêm xóa sửa thông tin các vai trò (hoặc món ăn) trong ứng dụng.
3	Inventory Management	Quản lý nhập xuất kho.
4	Revenue and expenditure management	Quản lý thu chi.
5	View analysis	Xem thống kê, phân tích trong một khoảng thời gian.
6	Confirm delivery bill is paid	Xác nhận hóa đơn giao hàng đã thanh toán.
7	Pick prepared bill to ship	Chọn hóa đơn giao hàng để giao.
8	Manage discount code, voucher code	Quản lý mã giảm giá và voucher giảm giá.
9	View History	Xem lịch sử mua hàng (đối với khách hàng), lịch sử thao tác với hóa đơn (đối với nhân viên).
10	Manage user and customer	Quản lý nhân viên (user) và khách hàng (customer)
11	Manage profile	Quản lý thông tin cá nhân.
12	Manage warehouse	Quản lý thêm xóa sửa thông tin nhà kho.
13	Submit new delivery bill	Tạo hóa đơn giao hàng.
14	Manage discount campaign	Quản lý chiến dịch giảm giá: chiến dịch giảm giá sẽ thay đổi giá của sản phẩm trong chiến dịch từ ngày bắt đầu đến ngày kết thúc.
15	Look for daily dish	Xem và tìm kiếm món ăn hôm nay.
16	Manage dish of store	Quản lý danh sách món ăn của cửa hàng.

17	Manage daily dish	Quản lý món ăn hôm nay.
18	Submit new bill	Tạo hóa đơn mới.
19	Pick bill to prepare and confirm dish prepared	Chọn hóa đơn để chuẩn bị món ăn và xác nhận món ăn đã chuẩn bị xong.
20	Confirm bill is paid	Xác nhận hóa đơn đã thanh toán.
21	Confirm dish is out of stock	Xác nhận món ăn đã hết hôm nay.
22	Edit unpaid bill	Sửa hóa đơn chưa thanh toán.
23	Pick prepared bill to delivery	Chọn hóa đơn để mang ra bàn.

3.4 Đặc tả Use-case

3.4.1 Đặc tả Use-case View history

Tóm tắt:

Usecase được sử dụng bởi staff, shipper, customer, chef. Chức năng chính là xem lại danh sách lịch sử sử dụng (lịch sử tương tác với hóa đơn).

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn mở menu.
3. Chọn xem lịch sử.
4. Hệ thống chuyển sang trang xem lịch sử.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên hoặc khách hàng.

Hậu điều kiện:

3.4.2 Đặc tả Use-case View history detail

Tóm tắt:

Usecase được sử dụng bởi staff, shipper, customer, chef. Chức năng chính là xem lại chi tiết một lịch sử.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn mở menu.
3. Chọn xem lịch sử.
4. Hệ thống chuyển sang trang xem lịch sử.
5. Chọn lịch sử cần xem
6. Hệ thống chuyển sang trang xem chi tiết lịch sử

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên hoặc khách hàng.

Hậu điều kiện:

3.4.3 Đặc tả Use-case Change avatar & information

Tóm tắt:

Usecase được sử dụng bởi staff, shipper, customer, chef. Chức năng chính là sửa thông tin cá nhân.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn mở menu.
3. Chọn sửa thông tin.
4. Hệ thống chuyển sang sửa thông tin.
5. Chọn nút sửa avatar.
6. Chọn chụp ảnh hoặc chọn ảnh từ thư viện.
7. Điền các thông tin khác.
8. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Nếu có lỗi trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên hoặc khách hàng.

Hậu điều kiện:

- Trang chỉnh sửa phải hiển thị loading cho đến khi update thành công.
- Tải lại trang và hiển thị với thông tin mới với thông báo cập nhật thành công.

3.4.4 Đặc tả Use-case Change password

Tóm tắt:

Usecase được sử dụng bởi staff, shipper, customer, chef. Chức năng chính là để đổi mật khẩu.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn mở menu.
3. Chọn đổi mật khẩu.
4. Hệ thống chuyển sang sửa mật khẩu.
5. Nhập mật khẩu cũ.
6. Nhập mật khẩu mới.
7. Nhập lại mật khẩu mới.
8. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Thông báo lỗi nếu sai mật khẩu cũ.
- Thông báo lỗi nếu mật khẩu mới không đủ điều kiện.
- Thông báo lỗi nếu mật khẩu mới không khớp với nhập lại.
- Nếu có lỗi trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên hoặc khách hàng.

Hậu điều kiện:

- Trang chỉnh sửa phải hiển thị loading cho đến khi update thành công.
- Tải lại trang và hiển thị thông báo cập nhật thành công.

3.4.5 *Đặc tả Use-case Change addresses*

Tóm tắt:

Usecase được sử dụng bởi customer. Chức năng chính là để thêm, xóa hoặc sửa những địa chỉ cá nhân.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn mở menu.
3. Chọn sửa địa chỉ.
4. Hệ thống chuyển sang trang sửa địa chỉ.
5. Điền địa chỉ vào Input Field và chọn nút thêm để thêm địa chỉ mới.
6. Chọn nút xóa tại mỗi địa chỉ để xóa địa chỉ.
7. Nhấn giữ vào địa chỉ để sửa địa chỉ.
8. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Nếu có lỗi trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên hoặc khách hàng.

Hậu điều kiện:

- Trang chỉnh sửa phải hiển thị loading cho đến khi update thành công.
- Tải lại trang và hiển thị thông báo cập nhật thành công.

3.4.6 *Đặc tả Use-case Submit new delivery bill*

Tóm tắt:

Usecase được sử dụng bởi customer. Chức năng chính là để tạo ra một hóa đơn giao hàng hay nói cách khác là đặt hàng trực tuyến.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn thêm các món ăn vào giỏ hàng (Use-case Add dish to cart).
3. Chọn xem giỏ hàng.
4. Hệ thống chuyển sang trang xem giỏ hàng.
5. Chỉnh sửa số lượng nếu cần (Use-case Select quantity).
6. Nhập mã giảm giá / voucher.
7. Chọn địa chỉ giao hàng.
8. Chọn xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Thông báo lỗi nếu mã giảm giá / voucher không có hiệu lực.
- Thông báo lỗi nếu món ăn đã hết.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho khách hàng.

Hậu điều kiện:

- Trang chỉnh sửa phải hiển thị loading cho đến khi update thành công.
- Tải lại trang và hiển thị thông báo cập nhật thành công.

3.4.7 Đặc tả Use-case Look for daily dish

Tóm tắt:

Usecase được sử dụng bởi staff, customer và khách vãng lai (không cần đăng nhập). Chức năng chính là tìm kiếm món ăn hôm nay.

Dòng sự kiện chính:

1. Đăng nhập (nếu dùng app của nhân viên).
2. Tại trang chủ, chọn tìm kiếm.
3. Điền từ khóa.
4. Chọn tìm kiếm.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho khách hàng, và app mobile dành cho nhân viên.

Hậu điều kiện:

- Trang chỉnh sửa phải hiển thị loading cho đến khi update thành công.
- Tải lại trang và hiển thị thông báo cập nhật thành công.

3.4.8 Đặc tả Use-case Pick bill to delivery

Tóm tắt:

Usecase được sử dụng bởi staff. Chức năng chính là chọn hóa đơn đã được chuẩn bị để mang ra bàn.

Dòng sự kiện chính:

1. Đăng nhập với tài khoản nhân viên.
2. Tại trang chủ, chọn Hóa đơn đã chuẩn bị xong (Chỉ hiển thị khi có hóa đơn mới được chuẩn bị xong).
3. Hệ thống chuyển sang trang danh sách hóa đơn đã chuẩn bị xong.
4. Chọn nút Giao tại hóa đơn để xác nhận giao.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên phục vụ.

Hậu điều kiện:

3.4.9 Đặc tả Use-case Submit new bill

Tóm tắt:

Usecase được sử dụng bởi staff. Chức năng chính là để tạo ra một hóa đơn hay nói cách khác là đặt món cho khách vào nhà hàng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn thêm các món ăn vào giỏ hàng (Use-case Add dish to cart).
3. Chọn xem giỏ hàng.
4. Hệ thống chuyển sang trang xem giỏ hàng.
5. Chỉnh sửa số lượng nếu cần (Use-case Select quantity).
6. Nhập số bàn.
7. Chọn xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Thông báo lỗi nếu món ăn đã hết.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên phục vụ.

Hậu điều kiện:

- Hiện thị loading khi nhấn xác nhận.
- Trả về trang chi tiết hóa đơn nếu tạo thành công.

3.4.10 Đặc tả Use-case Confirm bill is paid

Tóm tắt:

Usecase được sử dụng bởi staff. Chức năng chính là để nhân viên xác nhận đã thu tiền cho một hóa đơn.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn Danh sách hóa đơn đã hoàn thành.
3. Hệ thống chuyển sang trang Danh sách hóa đơn đã hoàn thành.
4. Chọn nút Check tại hóa đơn cần xác nhận thanh toán.
5. Chọn xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Thông báo lỗi nếu hóa đơn đã được xác nhận thanh toán (Trường hợp bất đồng bộ).
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên phục vụ.

Hậu điều kiện:

- Hiện thị loading khi nhấn xác nhận.
- Tải lại trang Danh sách hóa đơn đã hoàn thành với thông báo thành công.

3.4.11 *Đặc tả Use-case Add dish to unpaid bill*

Tóm tắt:

Usecase được sử dụng bởi staff. Chức năng chính là để nhân viên thêm món ăn trong một hóa đơn chưa thanh toán nếu có yêu cầu từ khách hàng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn mở Menu.
3. Chọn danh sách hóa đơn hiện tại.
4. Hệ thống chuyển sang trang Danh sách hóa đơn hiện tại.
5. Chọn hóa đơn cần sửa.
6. Hệ thống sẽ chuyển sang trang chi tiết hóa đơn.
7. Chọn thêm món.
8. Hệ thống sẽ chuyển sang trang chọn món.
9. Chọn món.
10. Hệ thống sẽ chuyển ngược trở lại trang chi tiết với món đã thêm.
11. Chỉnh sửa số lượng nếu cần (Use-case Select quantity)
12. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên phục vụ.

Hậu điều kiện:

- Hiện thị loading khi nhấn lưu.
- Tải lại trang Danh sách hóa đơn đã hoàn thành với thông báo thành công.

3.4.12 Đặc tả Use-case Remove dish from unpaid bill

Tóm tắt:

Usecase được sử dụng bởi staff. Chức năng chính là để nhân viên xóa món ăn trong một hóa đơn chưa thanh toán nếu có yêu cầu từ khách hàng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn mở Menu.
3. Chọn danh sách hóa đơn hiện tại.
4. Hệ thống chuyển sang trang Danh sách hóa đơn hiện tại.
5. Chọn hóa đơn cần sửa.
6. Hệ thống sẽ chuyển sang trang chi tiết hóa đơn.
7. Chọn nút xóa tại mỗi món ăn.
8. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên phục vụ.

Hậu điều kiện:

- Hiện thị loading khi nhấn lưu.
- Tải lại trang Danh sách hóa đơn đã hoàn thành với thông báo thành công.

3.4.13 Đặc tả Use-case Pick pill to prepare

Tóm tắt:

Usecase được sử dụng bởi chef. Chức năng chính là để đầu bếp chọn hóa đơn để chuẩn bị món.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút check tại hóa đơn để đánh dấu chuẩn bị cho hóa đơn.
3. Hệ thống sẽ hiển thị dialog xác nhận.
4. Chọn xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên đầu bếp.

Hậu điều kiện:

- Chuyển sang trang chi tiết hóa đơn.

3.4.14 Đặc tả Use-case Confirm dish prepared

Tóm tắt:

Usecase được sử dụng bởi chef. Chức năng chính là để đầu bếp xác nhận đã chuẩn bị xong một món nào đó trong hóa đơn.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn vào hóa đơn trong Danh sách hóa đơn đang chuẩn bị.
3. Hệ thống sẽ chuyển sang trang chi tiết hóa đơn.
4. Chọn vào Nút hoàn thành tại mỗi món ăn.
5. Chọn xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên đầu bếp.

Hậu điều kiện:

- Chuyển sang trang chi tiết hóa đơn.

3.4.15 Đặc tả Use-case Confirm dish is out of stock

Tóm tắt:

Use-case được sử dụng bởi chef. Chức năng chính là để đầu bếp xác nhận là món ăn đã hết.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn mở menu.
3. Chọn Danh sách món ăn.
4. Hệ thống sẽ chuyển trang Danh sách món ăn.
5. Chọn Nút xác nhận hết tại mỗi món ăn để xác nhận.
6. Hệ thống sẽ hiện dialog xác nhận.
7. Chọn xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên đầu bếp.

Hậu điều kiện:

- Tải lại trang.

3.4.16 Đặc tả Use-case Confirm delivery bill is paid

Tóm tắt:

Usecase được sử dụng bởi shipper. Chức năng chính là để xác nhận đã thu tiền khách hàng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, tại danh sách hóa đơn đang giao, chọn hóa đơn.
3. Hệ thống sẽ chuyển sang trang chi tiết hóa đơn.
4. Chọn Xác nhận thanh toán.
5. Điền số tiền và ghi chú nếu cần.
6. Chọn xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên shipper.

Hậu điều kiện:

- Trả lại trang chủ.

3.4.17 *Đặc tả Use-case Pick prepared bill to ship*

Tóm tắt:

Usecase được sử dụng bởi shipper. Chức năng chính là để chọn hóa đơn giao hàng đã chuẩn bị xong để giao.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, tại danh sách hóa đơn đã hoàn thành, chọn nút Giao hàng tại hóa đơn muốn giao.
3. Hệ thống sẽ hiển thị dialog xác nhận.
4. Chọn Xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý nếu hóa đơn đã được người khác giao (Lỗi bất đồng bộ).
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên shipper.

Hậu điều kiện:

- Tải lại trang chủ.

3.4.18 Đặc tả Use-case Create import bill

Tóm tắt:

Usecase được sử dụng bởi Ware Manager, hoặc Store Manager. Chức năng chính là để tạo ra phiếu nhập kho vào mỗi buổi sáng hằng ngày.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Nhập hàng.
3. Hệ thống sẽ chuyển sang trang Tạo hóa đơn nhập hàng.
4. Điền thông tin.
5. Chọn Tạo.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên ware manager hoặc store manager.

Hậu điều kiện:

- Trả lại trang chủ.

3.4.19 *Đặc tả Use-case Create end-day report*

Tóm tắt:

Usecase được sử dụng bởi Ware Manager hoặc Store Manager. Chức năng chính là để tạo ra báo cáo số lượng cuối ngày.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Báo cáo cuối ngày.
3. Hệ thống sẽ chuyển sang trang Tạo báo cáo.
4. Điền thông tin.
5. Chọn Tạo.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.
- Nếu có lỗi khác trả về từ server thì thông báo lỗi.

Tiền điều kiện:

- Sử dụng app mobile dành cho nhân viên.
- Đăng nhập bằng tài khoản nhân viên ware manager hoặc store manager.

Hậu điều kiện:

- Trả lại trang chủ.

3.4.20 Đặc tả Use-case View analysis

Tóm tắt:

Usecase được sử dụng bởi Accountant, Store Manager hoặc Admin. Chức năng chính là để xem các thống kê của cửa hàng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang bảng điều khiển (Bảng điều khiển hiển thị 1 vài thống kê nhỏ) hoặc trang thống kê (Hiển thị đầy đủ các thống kê).
3. Chọn cửa hàng cần xem (Mặc định là cửa hàng đầu tiên trong list cửa hàng).
4. Chọn khoảng thời gian cần xem (Mặc định là ngày).
5. Chọn Xem.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản nhân viên accountant, store manager hoặc admin.

Hậu điều kiện:

- Tải lại trang.

3.4.21 *Đặc tả Use-case Export / Print revenue report*

Tóm tắt:

Use-case được sử dụng bởi Accountant, Store Manager hoặc Admin. Chức năng chính là để xuất hoặc in báo cáo doanh thu.

Dòng sự kiện chính:

1. Đăng nhập.
2. Chọn Xuất báo cáo tại sidebar.
3. Hệ thống sẽ mở trang xuất báo cáo.
4. Chọn cửa hàng cần xem (Mặc định là cửa hàng đầu tiên trong list cửa hàng).
5. Chọn khoảng thời gian cần xem (Mặc định là ngày).
6. Chọn Xuất Excel hoặc in báo cáo doanh thu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản nhân viên accountant, store manager hoặc admin.

Hậu điều kiện:

- Không có.

3.4.22 *Đặc tả Use-case Export / Print expenditure report*

Tóm tắt:

Usecase được sử dụng bởi Accountant, Store Manager hoặc Admin. Chức năng chính là để xuất hoặc in báo cáo chi tiêu.

Dòng sự kiện chính:

1. Đăng nhập.
2. Chọn Xuất báo cáo tại sidebar.
3. Hệ thống sẽ mở trang xuất báo cáo.
4. Chọn cửa hàng cần xem (Mặc định là cửa hàng đầu tiên trong list cửa hàng).
5. Chọn khoảng thời gian cần xem (Mặc định là ngày).
6. Chọn Xuất Excel hoặc in báo cáo chi tiêu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản nhân viên accountant, store manager hoặc admin.

Hậu điều kiện:

- Không có.

3.4.23 Đặc tả Use-case *Edit import bill / Edit end-day report*

Tóm tắt:

Usecase được sử dụng bởi Store Manager. Chức năng chính là để sửa các phiếu nhập kho hoặc báo cáo cuối ngày.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Kho ở Sidebar.
3. Hệ thống sẽ chuyển sang trang Quản lý kho.
4. Chọn Kho cần sửa.
5. Hệ thống sẽ chuyển sang trang chi tiết kho.
6. Chọn danh sách phiếu nhập hoặc báo cáo cuối ngày.
7. Chọn Sửa tại phiếu cần sửa.
8. Điền thông tin.
9. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager.

Hậu điều kiện:

- Tải lại trang chi tiết kho.

3.4.24 *Đặc tả Use-case Create Warehouse*

Tóm tắt:

Usecase được sử dụng bởi Store Manager. Chức năng chính là để tạo Nhà kho cho Cửa hàng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Kho ở Sidebar.
3. Hệ thống sẽ chuyển sang trang Quản lý kho.
4. Chọn tạo Kho.
5. Điền thông tin.
6. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager.

Hậu điều kiện:

- Tải lại trang quản lý kho.

3.4.25 Đặc tả Use-case Edit warehouse

Tóm tắt:

Usecase được sử dụng bởi Store Manager. Chức năng chính là để sửa thông tin Nhà kho.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Kho ở Sidebar.
3. Hệ thống sẽ chuyển sang trang Quản lý kho.
4. Chọn nút sửa Kho.
5. Điền thông tin.
6. Sửa nhân viên quản lý (Nếu cần).
7. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager.

Hậu điều kiện:

- Tải lại trang quản lý kho.

3.4.26 *Đặc Use-case Delete warehouse*

Tóm tắt: Usecase được sử dụng bởi Store Manager. Chức năng chính là để xóa Nhà kho.
Dòng sự kiện chính: <ol style="list-style-type: none">1. Đăng nhập.2. Tại trang chủ, chọn nút Kho ở Sidebar.3. Hệ thống sẽ chuyển sang trang Quản lý kho.4. Chọn nút xóa tại Kho cần xóa.5. Hệ thống sẽ hiển thị dialog xác nhận.6. Chọn Xác nhận.
Dòng sự kiện phụ: <ul style="list-style-type: none">• Xử lý đăng nhập không thành công.• Xử lý lỗi thông tin không hợp lệ.
Tiền điều kiện: <ul style="list-style-type: none">• Sử dụng web admin.• Đăng nhập bằng tài khoản store manager.
Hậu điều kiện: <ul style="list-style-type: none">• Tải lại trang quản lý kho.

3.4.27 *Đặc tả Use-case Create voucher code / discount code*

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để tạo voucher hoặc mã giảm giá.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Voucher/Mã giảm giá.
3. Hệ thống sẽ mở trang Voucher/Mã giảm giá.
4. Chọn tạo Voucher/Mã giảm giá.
5. Chọn nút tạo.
6. Điền thông tin.
7. Chọn những cửa hàng áp dụng.
8. Chọn User (Đối với voucher) nếu có.
9. Chọn Tạo.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang Voucher/Mã giảm giá.

3.4.28 *Đặc tả Use-case Edit voucher / discount code information*

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để sửa thông tin voucher hoặc mã giảm giá.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Voucher/Mã giảm giá.
3. Hệ thống sẽ mở trang Voucher/Mã giảm giá.
4. Chọn sửa tại mỗi Voucher/Mã giảm giá.
5. Điền thông tin.
6. Chọn những cửa hàng áp dụng.
7. Chọn User (Đối với voucher) nếu có.
8. Chọn Lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang Voucher/Mã giảm giá.

3.4.29 Đặc tả Use-case Delete voucher / discount code

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để xóa voucher hoặc mã giảm giá.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Voucher/Mã giảm giá.
3. Hệ thống sẽ mở trang Voucher/Mã giảm giá.
4. Chọn xóa tại mỗi Voucher/Mã giảm giá.
5. Chọn Xác nhận tại dialog hiển thị.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang Voucher/Mã giảm giá.

3.4.30 *Đặc tả Use-case Create user / customer*

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để tạo user / customer.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút User / Customer.
3. Hệ thống sẽ mở trang User / Customer.
4. Chọn nút Tạo User / Customer.
5. Điền thông tin.
6. Chọn avatar.
7. Chọn Role (Đối với User).
8. Chọn tạo.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang User / Customer.

3.4.31 *Đặc tả Use-case Edit user / customer*

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để sửa thông tin user / customer.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút User / Customer.
3. Hệ thống sẽ mở trang User / Customer.
4. Chọn nút sửa tại mỗi User / Customer.
5. Điền thông tin.
6. Chọn avatar (Nếu cần).
7. Chọn Role (Đối với User) nếu cần.
8. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang User / Customer.

3.4.32 Đặc tả Use-case Delete use / customer

Tóm tắt:

Use-case được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để xóa user / customer.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút User / Customer.
3. Hệ thống sẽ mở trang User / Customer.
4. Chọn nút xóa tại mỗi User / Customer.
5. Chọn xác nhận tại dialog hiển thị.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang User / Customer.

3.4.33 Đặc tả Use-case Create discount campaign

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để tạo chiến dịch giảm giá.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Chiến dịch giảm giá.
3. Hệ thống sẽ mở trang Chiến dịch giảm giá.
4. Chọn tạo mới.
5. Điền thông tin.
6. Chọn những cửa hàng áp dụng.
7. Chọn những món ăn và thêm giá trị giảm giá riêng (Nếu cần).
8. Chọn tạo.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang Chiến dịch giảm giá.

3.4.34 Đặc tả Use-case Edit discount campaign

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để sửa chiến dịch giảm giá.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Chiến dịch giảm giá.
3. Hệ thống sẽ mở trang Chiến dịch giảm giá.
4. Chọn sửa tại chiến dịch cần sửa.
5. Điền thông tin sửa.
6. Chọn những cửa hàng áp dụng.
7. Chọn hoặc xóa những món ăn và thêm giá trị giảm giá riêng (Nếu cần).
8. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý lỗi thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang Chiến dịch giảm giá.

3.4.35 *Đặc tả Use-case Delete discount campaign*

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để xóa chiến dịch giảm giá.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Chiến dịch giảm giá.
3. Hệ thống sẽ mở trang Chiến dịch giảm giá.
4. Chọn xóa tại chiến dịch cần xóa.
5. Xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang Chiến dịch giảm giá.

3.4.36 *Đặc tả Use-case Add dishes to list dished of store*

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để thêm món ăn vào danh sách món ăn của cửa hàng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Cửa hàng.
3. Hệ thống sẽ mở trang Danh sách cửa hàng User quản lý.
4. Chọn Danh sách món ăn.
5. Chọn Thêm mới.
6. Hệ thống sẽ hiển thị Dialog với danh sách các món ăn.
7. Chọn thêm tại món ăn cần thêm.
8. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang Danh sách món ăn.

3.4.37 *Đặc tả Use-case Remove dish from lish dished of store*

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để xóa món ăn vào danh sách món ăn của cửa hàng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Cửa hàng.
3. Hệ thống sẽ mở trang Danh sách cửa hàng User quản lý.
4. Chọn Danh sách món ăn.
5. Chọn xóa tại món ăn cần xóa.
6. Xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang Danh sách món ăn.

3.4.38 *Đặc tả Use-case Add dished to list daily dishes*

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để thêm món ăn vào danh sách món ăn sẽ bán theo ngày chọn.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Món ăn hằng ngày trong Sidebar.
3. Hệ thống sẽ mở trang Danh sách món ăn hằng ngày.
4. Chọn Thêm món ăn.
5. Hệ thống sẽ hiển thị Dialog với danh sách các món ăn của cửa hàng.
6. Chọn thêm tại các món cần thêm.
7. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang Danh sách món ăn hằng ngày.

3.4.39 Đặc tả Use-case Remove dish from list daily dishes

Tóm tắt:

Usecase được sử dụng bởi Store Manager hoặc Admin. Chức năng chính là để xóa món ăn trong danh sách món ăn sẽ bán theo ngày chọn.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Món ăn hằng ngày trong Sidebar.
3. Hệ thống sẽ mở trang Danh sách món ăn hằng ngày.
4. Chọn xóa tại món ăn cần xóa.
5. Xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản store manager hoặc Admin.

Hậu điều kiện:

- Tải lại trang Danh sách món ăn hằng ngày.

3.4.40 Đặc tả Use-case Create store / role / dish

Tóm tắt:

Usecase được sử dụng bởi Admin. Chức năng chính là để tạo các cửa hàng /vai trò /món ăn trong ứng dụng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Cửa hàng (hoặc Vai trò hoặc Món ăn) trong Sidebar.
3. Hệ thống sẽ mở trang Danh sách Cửa hàng (hoặc Vai trò hoặc Món ăn).
4. Chọn thêm mới.
5. Điền thông tin.
6. Chọn tạo.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản Admin.

Hậu điều kiện:

- Tải lại trang Danh sách Cửa hàng (hoặc Vai trò hoặc Món ăn).

3.4.41 *Đặc tả Use-case Edit store / role / dish*

Tóm tắt:

Usecase được sử dụng bởi Admin. Chức năng chính là để sửa thông tin cửa hàng /vai trò /món ăn trong ứng dụng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Cửa hàng (hoặc Vai trò hoặc Món ăn) trong Sidebar.
3. Hệ thống sẽ mở trang Danh sách Cửa hàng (hoặc Vai trò hoặc Món ăn).
4. Chọn Cửa hàng (hoặc Vai trò hoặc Món ăn) cần sửa.
5. Điền thông tin.
6. Chọn lưu.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.
- Xử lý thông tin không hợp lệ.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản Admin.

Hậu điều kiện:

- Tải lại trang Danh sách Cửa hàng (hoặc Vai trò hoặc Món ăn).

3.4.42 *Đặc tả Use-case Delete store / role / dish*

Tóm tắt:

Usecase được sử dụng bởi Admin. Chức năng chính là để xóa cửa hàng /vai trò /món ăn trong ứng dụng.

Dòng sự kiện chính:

1. Đăng nhập.
2. Tại trang chủ, chọn nút Cửa hàng (hoặc Vai trò hoặc Món ăn) trong Sidebar.
3. Hệ thống sẽ mở trang Danh sách Cửa hàng (hoặc Vai trò hoặc Món ăn).
4. Chọn xóa tại Cửa hàng (hoặc Vai trò hoặc Món ăn) cần sửa.
5. Xác nhận.

Dòng sự kiện phụ:

- Xử lý đăng nhập không thành công.

Tiền điều kiện:

- Sử dụng web admin.
- Đăng nhập bằng tài khoản Admin.

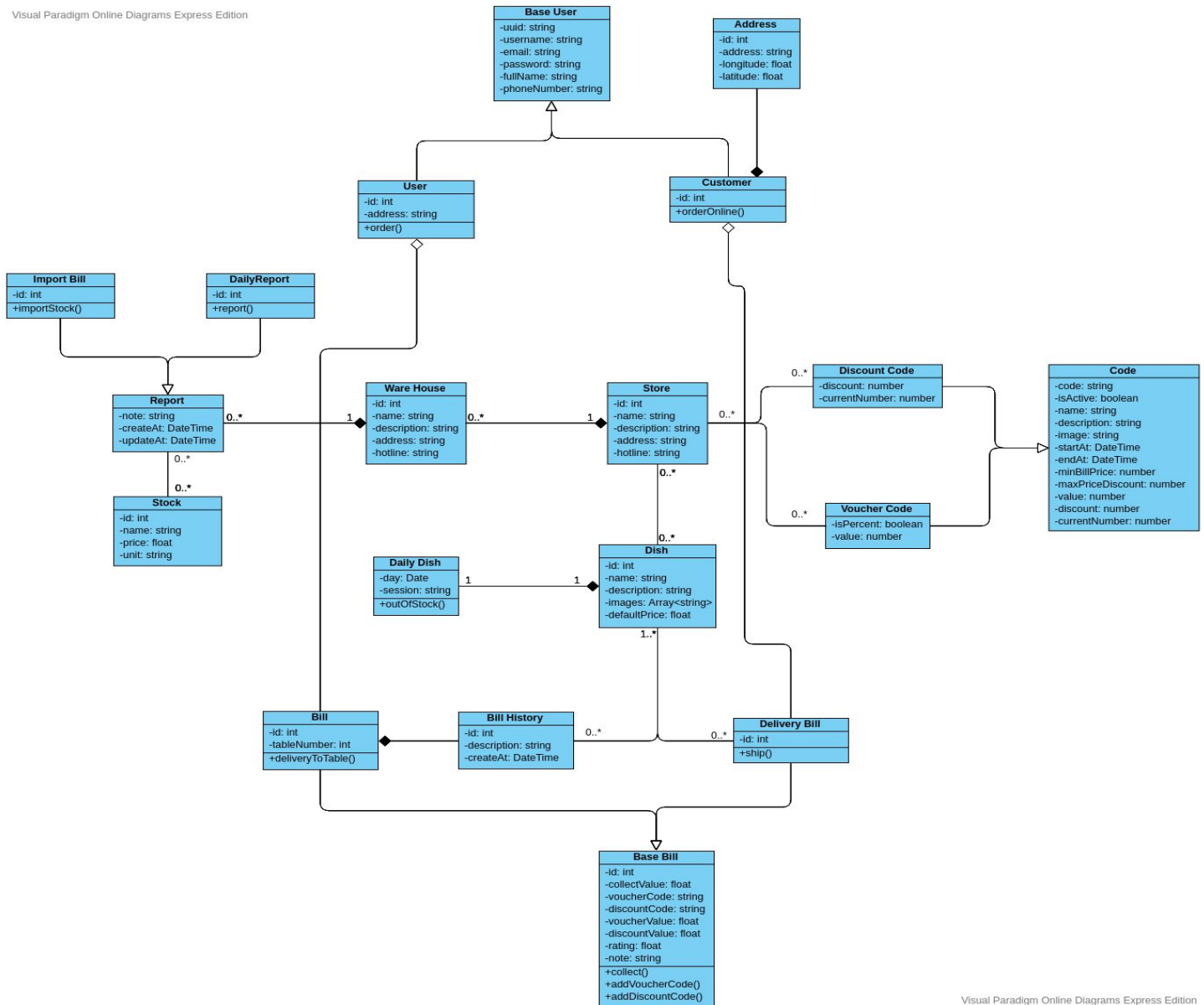
Hậu điều kiện:

- Tải lại trang Danh sách Cửa hàng (hoặc Vai trò hoặc Món ăn).

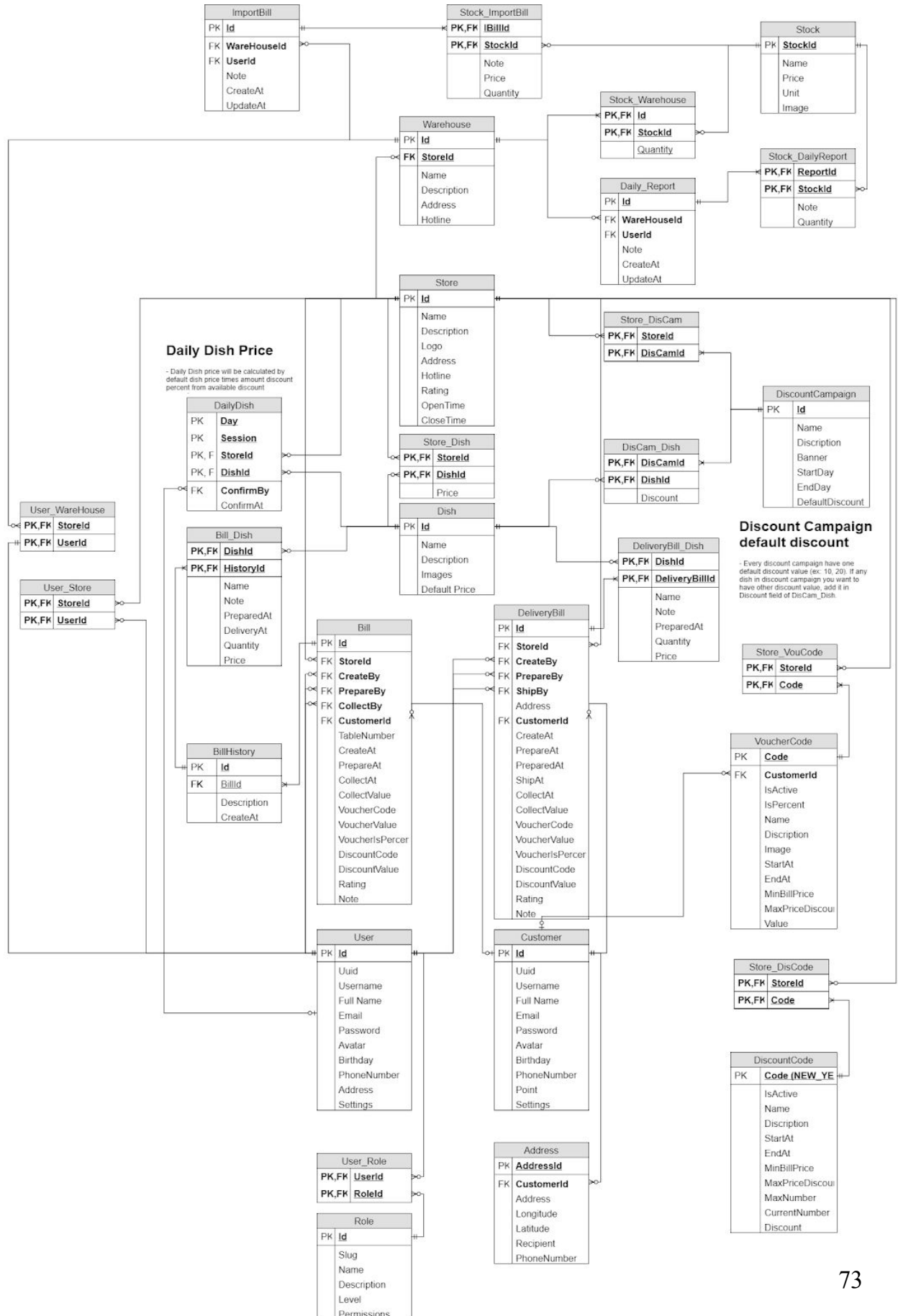
CHƯƠNG 4: PHÂN TÍCH

Sơ đồ lớp (mức phân tích)

Visual Paradigm Online Diagrams Express Edition



CHƯƠNG 5: THIẾT KẾ DỮ LIỆU



5.1 Danh sách các bảng dữ liệu

STT	Tên bản dữ liệu	Diễn giải
1	User	Tài khoản đăng nhập
2	Customer	Tài khoản đăng nhập của khách hàng
3	Role	Vai trò
4	Address	Địa chỉ của khách hàng
5	Bill	Hóa đơn
6	DeliveryBill	Hóa đơn delivery
7	BillHistory	Lịch sử hóa đơn
8	Dish	Món ăn
9	DailyDish	Món ăn hằng ngày
10	Store	Cửa hàng
11	Warehouse	Nhà kho
12	DailyReport	Báo cáo hằng ngày

13	Stock_Warehouse	Số lượng nguyên liệu ở kho
14	Stock_DailyReport	Báo cáo số lượng nguyên liệu hằng ngày
15	Stock	Tên nguyên liệu
16	Stock_ImportBill	Xuất hóa đơn nguyên liệu
17	ImportBill	Xuất hóa đơn
18	DiscountCampaign	Chiến dịch giảm giá
19	Store_DisCam	Chiến dịch giảm giá cho từng cửa hàng
20	DisCam_Dish	Chiến dịch giảm giá cho từng món
21	Store_DisCode	Mã giảm giá ở cửa hàng
22	DiscountCode	Mã giảm giá
23	VoucherCode	Phiếu giảm giá
24	Store_VouCode	Phiếu giảm giá áp dụng cho từng cửa hàng

5.2 Mô tả chi tiết từng bảng dữ liệu

5.2.1 Bảng User

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	Mã người dùng tự tăng
2	username	varchar	Tên tài khoản
3	fullname	varchar	Tên người dùng
4	email	varchar	Email
5	password	varchar	Mật khẩu
6	avatar	varchar	Ảnh đại diện
7	birthday	date	Ngày sinh
8	phonenumber	varchar	Số điện thoại
9	address	Varchar	Địa chỉ
10	settings	json	///

5.2.2 *Bảng customer*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	Mã khách hàng
2	username	varchar	Tên tài khoản của khách hàng
3	fullname	varchar	Tên của khách hàng
4	email	Varchar	Email
5	password	varchar	Mật khẩu
6	avatar	varchar	Ảnh đại diện
7	birthday	date	Ngày sinh
8	phonenumber	varchar	Số điện thoại của khách hàng
9	settings	json	///

5.2.3 *Role*

STT	Thuộc tính	kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	Mã vai trò
2	slug	varchar	//.
3	name	varchar	Tên
4	description	varchar	Miêu tả
5	level	integer	//.
6	permissions	text[]	Danh sách các quyền

5.2.4 *Address*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	
2	address	varchar	Địa chỉ
3	longitude	double	Kinh độ
4	latitude	double	Vĩ độ
5	customerId	integer	Mã khách hàng

5.2.5 *Bill*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>Id</u>	integer	
2	deleteAt	datetime	Xóa món lúc
3	tableNumber	integer	Số bàn
4	createAt	datetime	Tạo hóa đơn lúc
5	prepareAt	datetime	Chuẩn bị món lúc
6	collectAt	datetime	
7	collectValue	money	
8	voucherCode	varchar	Mã voucher
9	voucherValue	integer	Giá trị voucher
10	voucherIsPercent	boolean	
11	discountCode	varchar	Mã discount
12	discountValue	integer	Giá trị discount
13	rating	double	
14	note	varchar	
15	deleteById	integer	
16	CreateById	inteder	
17	prepareById	integer	
18	collectById	integer	
19	customerId	integer	

5.2.6 Delivery Bill

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	
2	createAt	datetime	Tạo hóa đơn lúc
3	prepareAt	datetime	Đang được chuẩn bị lúc
4	preparedAt	datetime	Đã được chuẩn bị lúc
5	shipAt	datetime	Giao món lúc
6	collectAt	datetime	
7	collectValue	money	
8	address	text	
9	longitude	double	Kinh độ
10	latitude	double	Vĩ độ
11	voucherCode	varchar	Mã voucher
12	voucherValue	integer	Giá trị voucher
13	voucherIsPercent	boolean	
14	discountCode	varchar	Mã discount
15	discountValue	integer	Giá trị discount
16	rating	double	
17	note	varchar	
18	customerId	integer	
19	deleteById	integer	
20	prepareById	integer	
21	shipById	integer	

5.2.7 *Bill History*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	id	integer	Mã bill history
2	billId	integer	Mã hóa đơn
3	description	varchar	Mô tả
6	createAt	datetime	Tạo hóa đơn lúc
7	userId	integer	

5.2.8 *Dish*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	Mã món ăn
2	name	varchar	Tên món
3	description	varchar	Mô tả
4	Images	text	Hình ảnh
5	defaultPrice	money	Giá mặc định

5.2.9 *DailyDish*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>day</u>	datetime	Ngày
2	<u>session</u>	enum	
3	<u>storeId</u>	integer	Mã cửa hàng
4	<u>dishId</u>	integer	mã món ăn
5	confirmAt	datetime	
6	confirmById	integer	

5.2.10 *Store*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	Mã nhà hàng
2	name	varchar	Tên nhà hàng
3	description	varchar	Mô tả
4	logo	varchar	Hình ảnh nhà hàng
5	address	varchar	Địa chỉ nhà hàng

5.2.11 Warehouse

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	Mã warehouse
2	name	varchar	Tên
3	description	varchar	Mô tả
4	address	varchar	Địa chỉ
5	hotline	varchar	Số điện thoại
6	storeId	integer	Mã nhà hàng

5.2.12 Daily Report

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	Mã daily report
2	note	varchar	Ghi chú
3	createAt	datetime	Xuất daily report lúc
4	updateAt	datetime	Cập nhật daily report lúc
5	warehouseId	integer	Mã warehouse
6	userId	integer	Mã user

5.2.13 *Stock_Warehouse*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>warehouseId</u>	integer	Mã warehouse
2	<u>stockId</u>	integer	Mã stock
3	quantity	integer	Số lượng stock

5.2.14 *Stock_DailyReport*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>dailyReportId</u>	integer	Mã daily report
2	<u>stockId</u>	integer	Mã stock
3	note	varchar	Ghi chú
4	quantity	integer	Số lượng

5.2.15 *Stock*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	Mã stock
2	name	varchar	tên stock
3	price	money	Giá
4	unit	varchar	
5	image	varchar	Hình ảnh

5.2.16 Stock Import Bill

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>importBillId</u>	integer	Mã nhập hóa đơn
2	<u>stockId</u>	integer	Mã stock
3	note	varchar	Ghi chú
4	quantity	integer	Số lượng
5	price	integer	Giá

5.2.17 Import Bill

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	Mã import bill
2	note	varchar	Ghi chú
3	createAt	datetime	Tạo import bill lúc
4	updateAt	datetime	Cập nhật bill lúc
5	warehouseId	integer	Mã warehouse
6	userId	integer	Mã user

5.2.18 *DiscountCampaign*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>id</u>	integer	Mã discount campaign
2	name	varchar	Tên
3	description	varchar	Mô tả
4	banner	varchar	banner
5	startAt	datetime	Ngày bắt đầu
6	endAt	datetime	Ngày kết thúc
7	defaultDiscount	integer	Giá trị của discount

5.2.19 *Store_DisCam*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>discountCampaignId</u>	integer	Mã discount campaign
2	<u>storeId</u>	integer	Mã cửa hàng

5.2.20 *DisCam_Dish*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>discountCampaignId</u>	integer	Mã discount campaign
2	<u>dishId</u>	integer	Mã món ăn
3	discount	integer	Giá trị discount

5.2.21 Store_DisCode

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>discountCodeCode</u>	varchar	Mã của mã discount
2	<u>storeId</u>	integer	Mã nhà hàng

5.2.22 DiscountCode

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>code</u>	varchar	discount code
2	isActive	boolean	còn sử dụng được hay không
3	name	varchar	Tên
4	description	varchar	Mô tả
5	startAt	datetime	Ngày bắt đầu
6	endAt	datetime	Ngày kết thúc
7	minBillPrice	money	Giá tối thiểu của hóa đơn để có thể áp dụng discount
8	maxPriceDiscount	money	mức giảm giá tối đa
9	maxNumber	integer	Số tối đa
10	discount	integer	giá trị discount

5.2.23 *Store_VouCode*

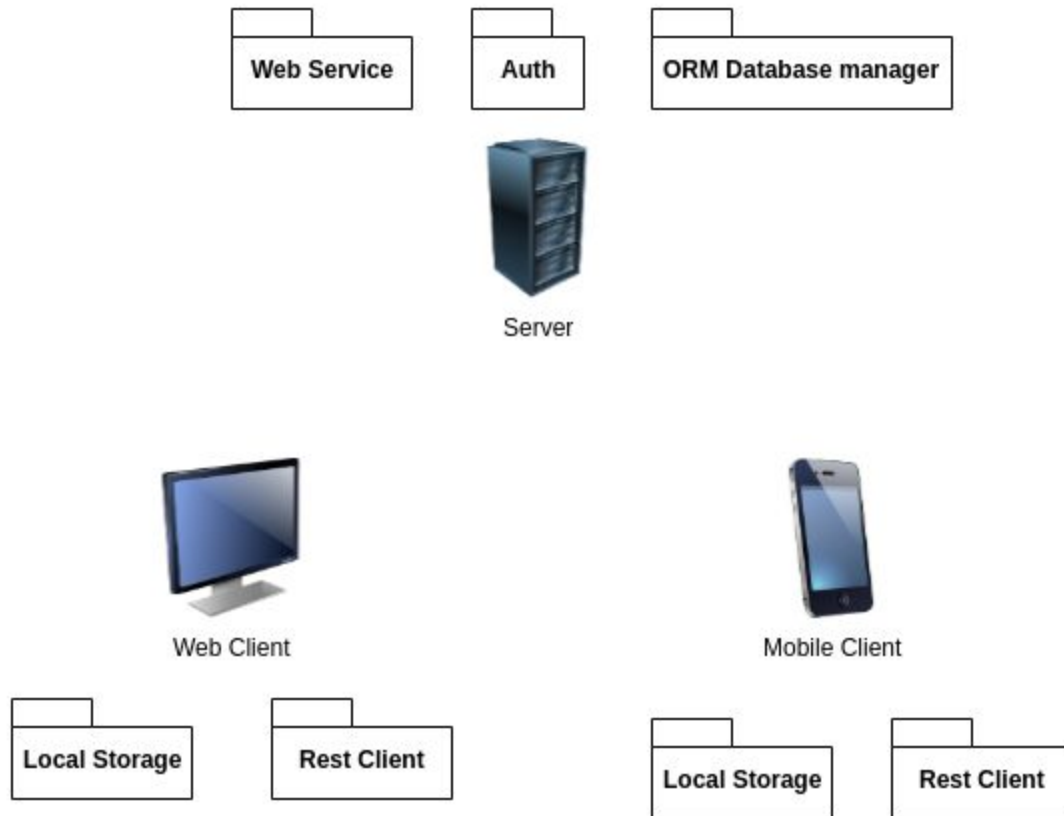
STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>voucherCodeCode</u>	varchar	Mã của Mã voucher
2	<u>storeId</u>	integer	Mã nhà hàng

5.2.24 *VoucherCode*

STT	Thuộc tính	Kiểu dữ liệu	Ghi chú
1	<u>code</u>	varchar	Mã voucher
2	isActive	boolean	Mã voucher còn hiệu lực hay không
3	isPercent	boolean	
4	name	varchar	Tên
5	description	varchar	Mô tả
6	image	varchar	Hình ảnh
7	startAt	datetime	Ngày bắt đầu
8	endAt	datetime	Ngày kết thúc
9	minBillPrice	money	Giá tối thiểu có thể áp dụng voucher
10	maxPriceDiscount	money	
11	value	integer	Giá trị discount

CHƯƠNG 6: KIẾN TRÚC ỨNG DỤNG

6.1 Kiến trúc hệ thống

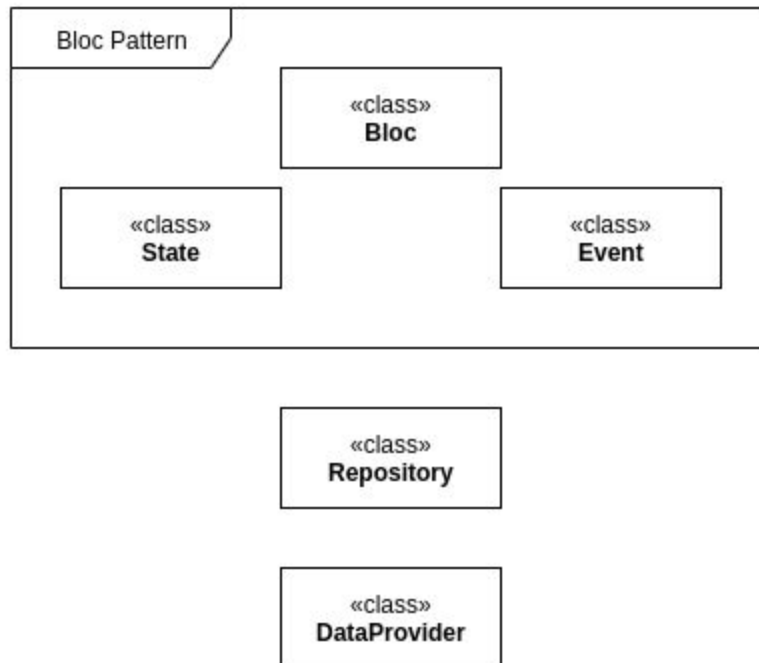


Thành phần	Diễn giải
Web Service	Web Service
Auth	Xác thực và phân quyền
ORM Database manager	Dùng ORM quản lý database
Local Storage	Lưu trữ trên thiết bị client
Rest Client	Giao tiếp với server

6.2 Mô tả chi tiết từng thành phần trong hệ thống

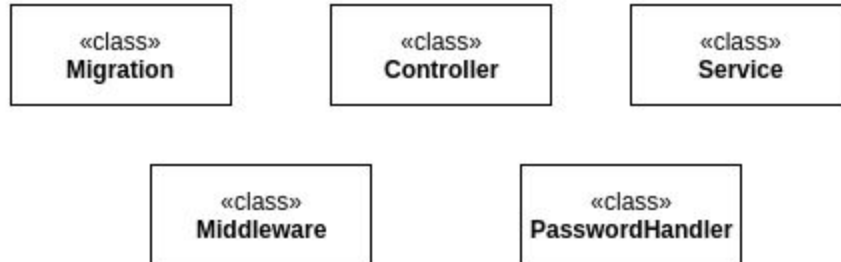
6.2.1 Client

Client mobile áp dụng mẫu thiết kế Bloc. Để quản lý trạng thái của ứng dụng, dưới đây là sơ đồ lớp theo mô hình Bloc:



Lớp đối tượng	Diễn giải
Bloc	Nhận sự kiện từ User, thay đổi trạng thái
State	Trạng thái
Event	Sự kiện
Repository	Lớp trung gian để giao diện giao tiếp với server
Dataprovider	Lớp ảo để các lớp kế thừa thực hiện nhiệm vụ giao tiếp với database cho một chức năng nào đó. Ví dụ: UserProvider, ProductProvider...

6.2.2 Web Server



Lớp đối tượng	Diễn giải
Migration	Quản lý version database
Controller	Lớp trung gian giữa Router và Service
Service	Lớp thực hiện gọi Modal giao tiếp với database
Middleware	Lớp chức năng thêm vào Router để xác thực phân quyền
PasswordHandle	Thao tác với mật khẩu

6.2.3 *Web Client*



Lớp đối tượng	Diễn giải
Service	Lớp ảo, các lớp kế thừa sẽ giao tiếp với server restful
Router	Lớp để trỏ tới các trang public
PrivateRouter	Lớp để trỏ tới các trang xác thực phân quyền để truy cập

CHƯƠNG 7: THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG

7.1 Thiết kế giao diện trên mobile (customer app)

7.1.1 Danh sách các màn hình

STT	Tên màn hình	Loại màn hình	Chức năng
1	Màn hình đăng nhập	Màn hình nhập	Đăng nhập
2	Màn hình đăng ký	Màn hình nhập	Đăng ký
3	Màn hình sửa mật khẩu	Màn hình nhập	Sửa mật khẩu
4	Màn hình sửa thông tin cá nhân	Màn hình hiển thị và nhập thông tin	Sửa thông tin cá nhân
5	Màn hình thông tin cá nhân	Màn hình hiển thị	Hiển thị thông tin cá nhân
6	Màn hình danh sách cửa hàng	Màn hình hiển thị và select	Hiển thị danh sách nhà hàng khi chưa chọn nhà hàng
7	Màn hình danh sách món ăn	Màn hình hiển thị	Hiển thị danh sách các món ăn hàng ngày tại cửa hàng đã chọn
8	Màn hình thông tin món ăn	Màn hình hiển thị	Hiển thị thông tin chi tiết món ăn như mô tả, giá, review hoặc có thể thêm món ăn vào danh sách món ăn yêu thích
9	Màn hình giỏ hàng	Màn hình hiển thị	Hiển thị danh sách món ăn đã chọn, tại đây có thể thêm, bớt số lượng của từng món hoặc xóa món ăn khỏi danh sách.
10	Màn hình tạo hóa đơn	Màn hình hiển thị và	Hiển thị danh sách món

		nhập	ăn cũng như số lượng từng món, có thể thêm discount, voucher nếu có. Select địa chỉ để nhân viên có thể giao hàng.
11	Màn hình thông tin hóa đơn	Màn hình hiển thị và nhập	Hiển thị chi tiết hóa đơn như món ăn, số lượng từng món, tổng tiền.
12	Màn hình danh sách hóa đơn của tôi	Màn hình hiển thị	Hiển thị danh sách các hóa đơn của khách hàng đã order.

7.1.2 Mô tả từng màn hình

7.1.2.1 Màn hình đăng nhập

Màn hình cho phép customer đăng nhập vào app.



7.1.2.2 Màn hình đăng ký

Màn hình cho phép customer đăng ký khi chưa có tài khoản.

FoodCircles

Tên đăng nhập

Email

Mật khẩu

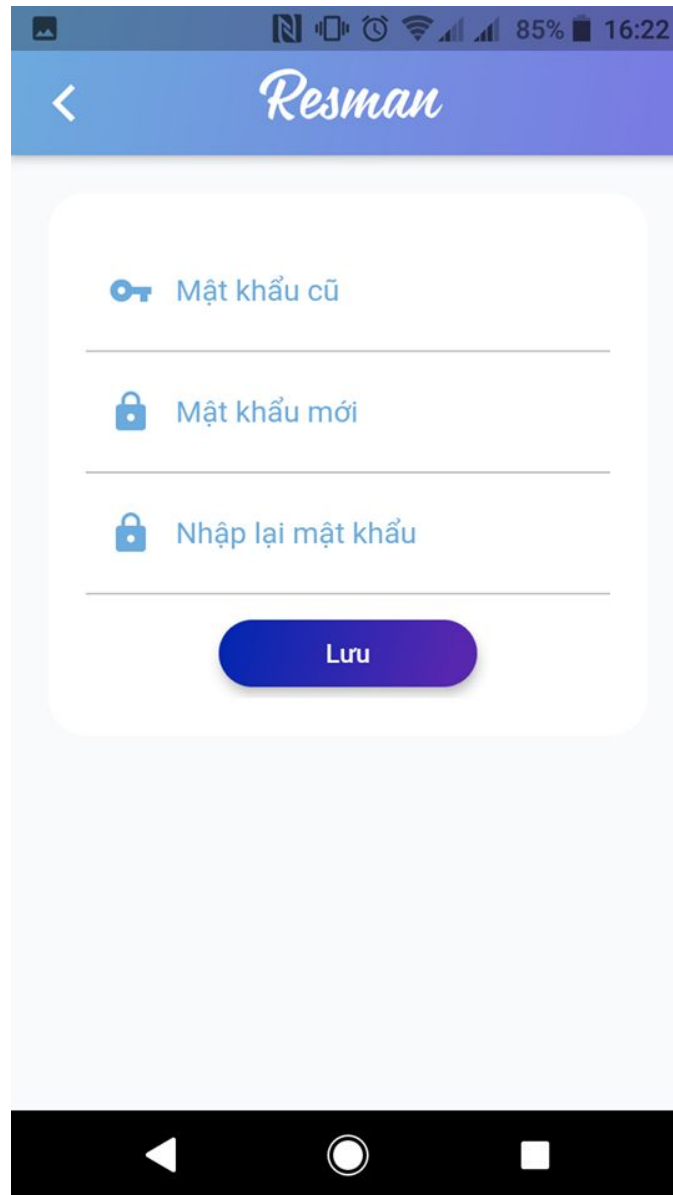
Nhập lại mật khẩu

Đăng ký

[Đăng nhập](#)

7.1.2.3 Màn hình sửa mật khẩu

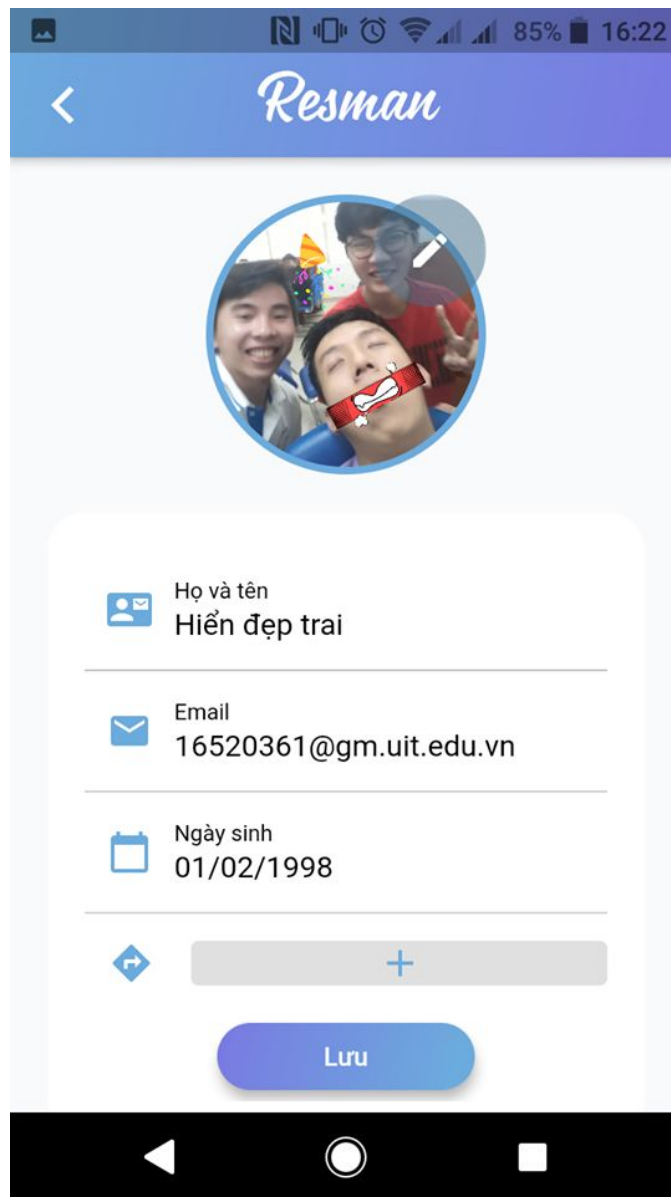
Có thể thay đổi mật khẩu tại màn hình này



The screenshot displays the 'Resman' application interface for changing a password. The top status bar indicates a battery level of 85% and a time of 16:22. The app's header is blue, featuring a back arrow on the left and the 'Resman' logo in the center. The main content area is a white rounded rectangle with three input fields, each preceded by an icon: a key for 'Mật khẩu cũ' (Old password), a lock for 'Mật khẩu mới' (New password), and another lock for 'Nhập lại mật khẩu' (Repeat password). A blue 'Lưu' (Save) button is positioned below these fields. The bottom of the screen shows the standard Android navigation bar.

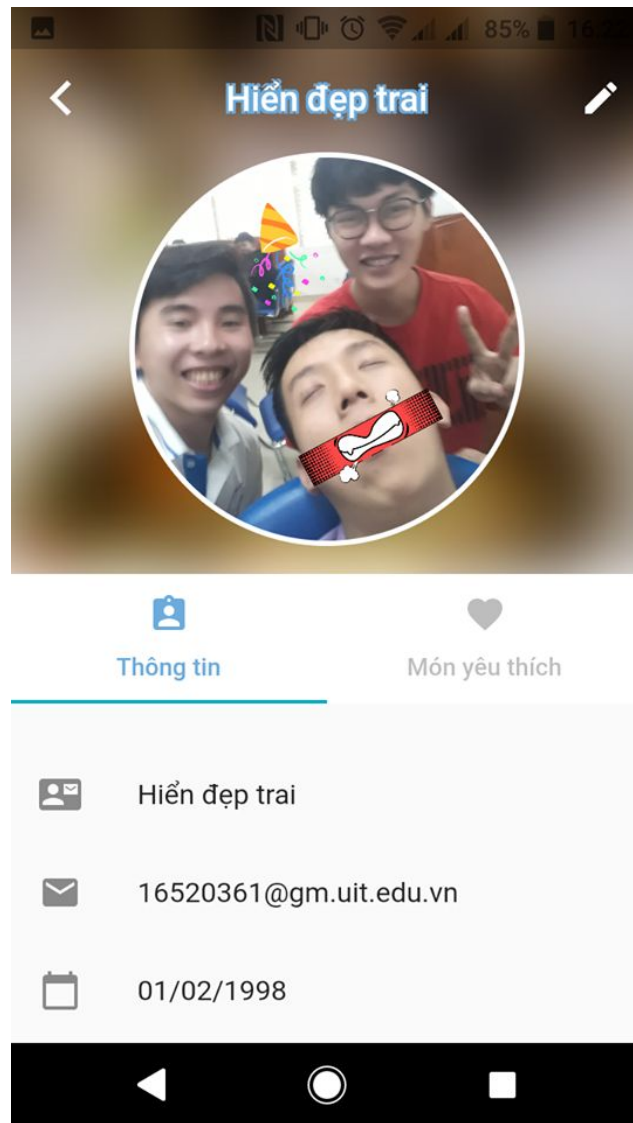
7.1.2.4 Màn hình sửa thông tin cá nhân

Thay đổi thông tin cá nhân như họ tên, ảnh đại diện, email... tại màn hình này.



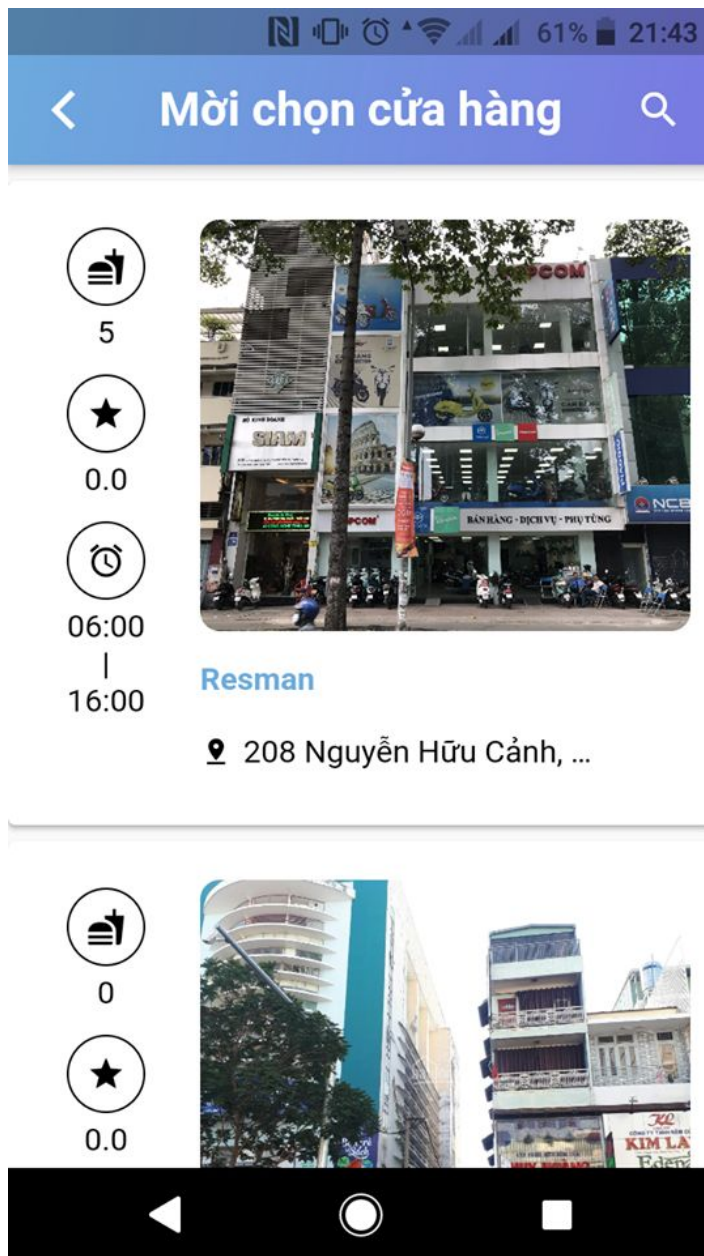
7.1.2.5 Màn hình thông tin cá nhân

Hiển thị thông tin cá nhân như ảnh đại diện, họ tên, email, ngày sinh, món ăn yêu thích.



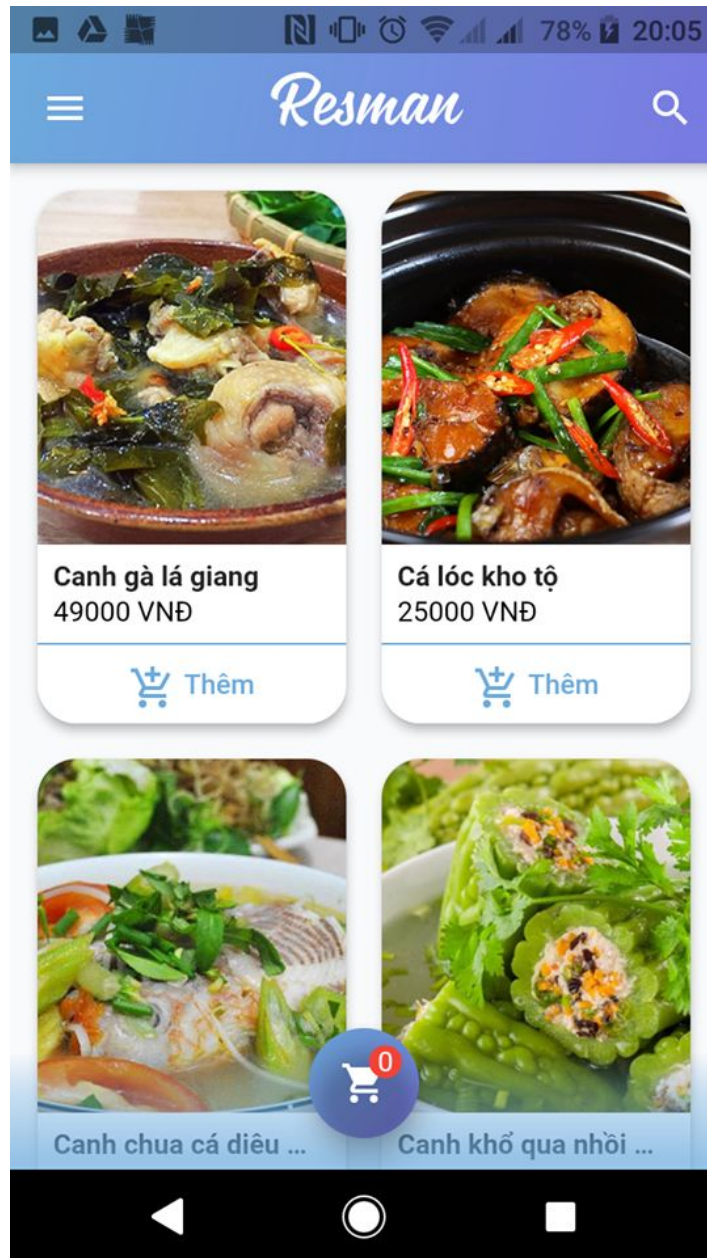
7.1.2.6 Màn hình danh sách nhà hàng

Hiển thị danh sách nhà hàng để chọn, gồm các thông tin của nhà hàng như logo, tên, địa chỉ, giờ mở cửa, đóng cửa...



7.1.2.7 Màn hình danh sách món ăn

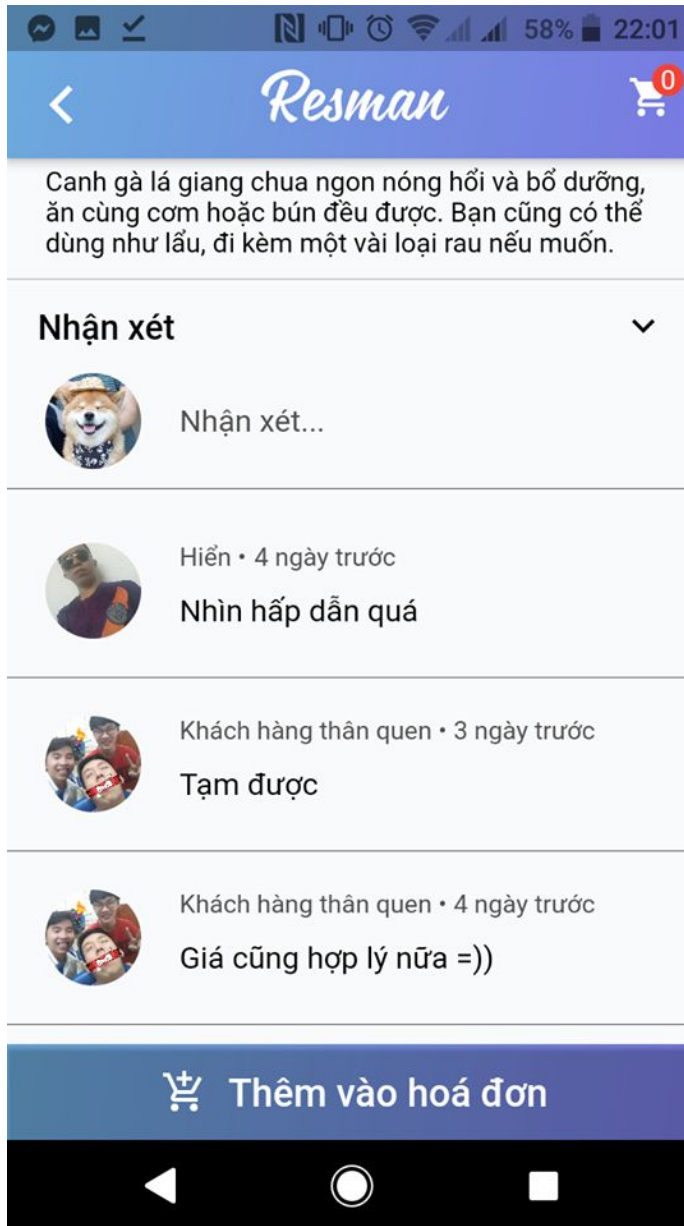
Màn hình hiển thị danh sách món ăn hàng ngày của quán gồm tên món ăn, giá, hình ảnh món ăn.



7.1.2.8 Màn hình thông tin món ăn

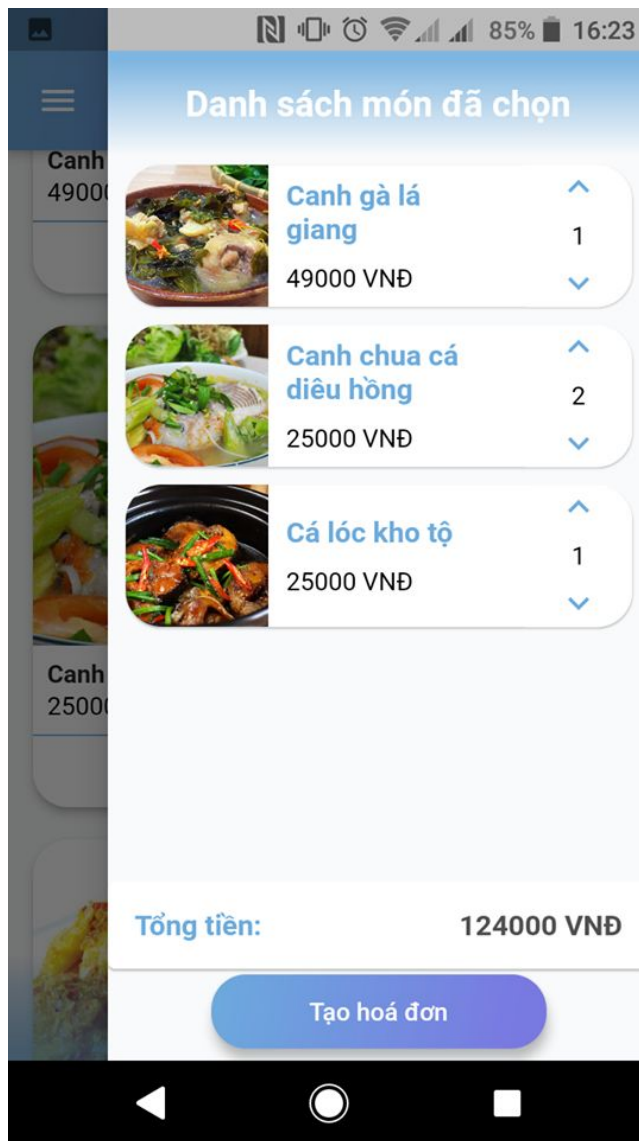
Hiển thị thông tin chi tiết món ăn như hình ảnh, mô tả, giá, review hoặc thêm vào danh sách món ăn yêu thích.





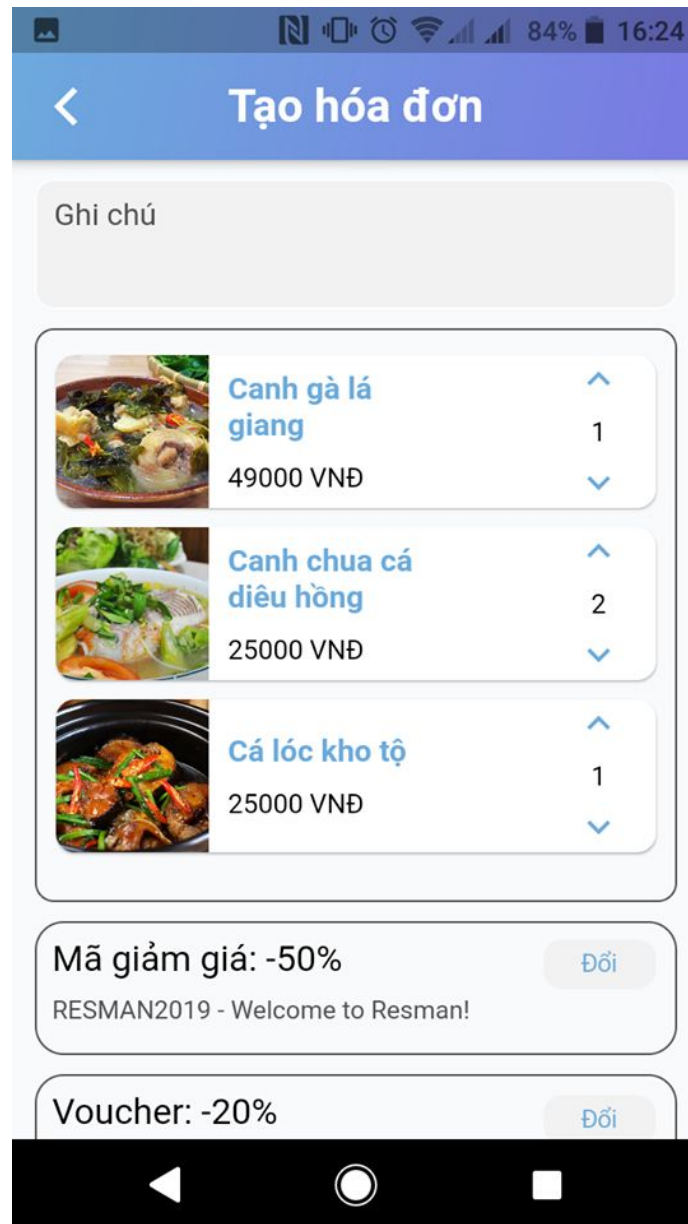
7.1.2.9 Màn hình giỏ hàng

Hiện thị danh sách các món ăn cũng như số lượng món ăn được thêm vào giỏ hàng, tổng tiền mặc định.



7.1.2.10 Màn hình tạo hóa đơn

Hiện thị danh sách món ăn cũng như số lượng từng món, có thể thêm discount, voucher nếu có, tổng tiền sau khi được giảm giá. Select địa chỉ để nhân viên có thể giao hàng.



☰ 📷 ✍️ N 📶 🔔 📶 57% 🔋 22:02

< **Tạo hóa đơn**



Mã giảm giá: -50% [Đổi](#)

RESMAN2019 - Welcome to Resman!

Voucher: [Chọn](#)

Chưa chọn voucher!

Địa chỉ: [Đổi](#)

tân phú, q9

Thành tiền:

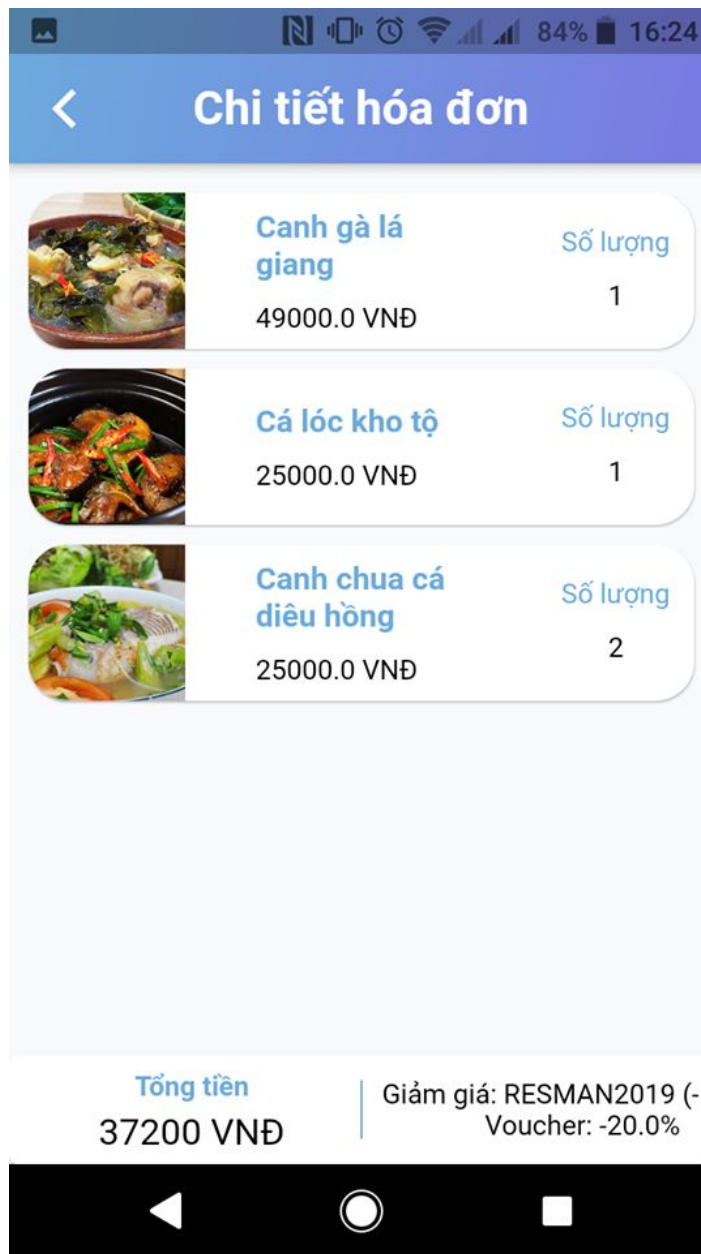
~~74000 VNĐ~~ **37000 VNĐ**

Tạo hóa đơn

◀ ○ ▶

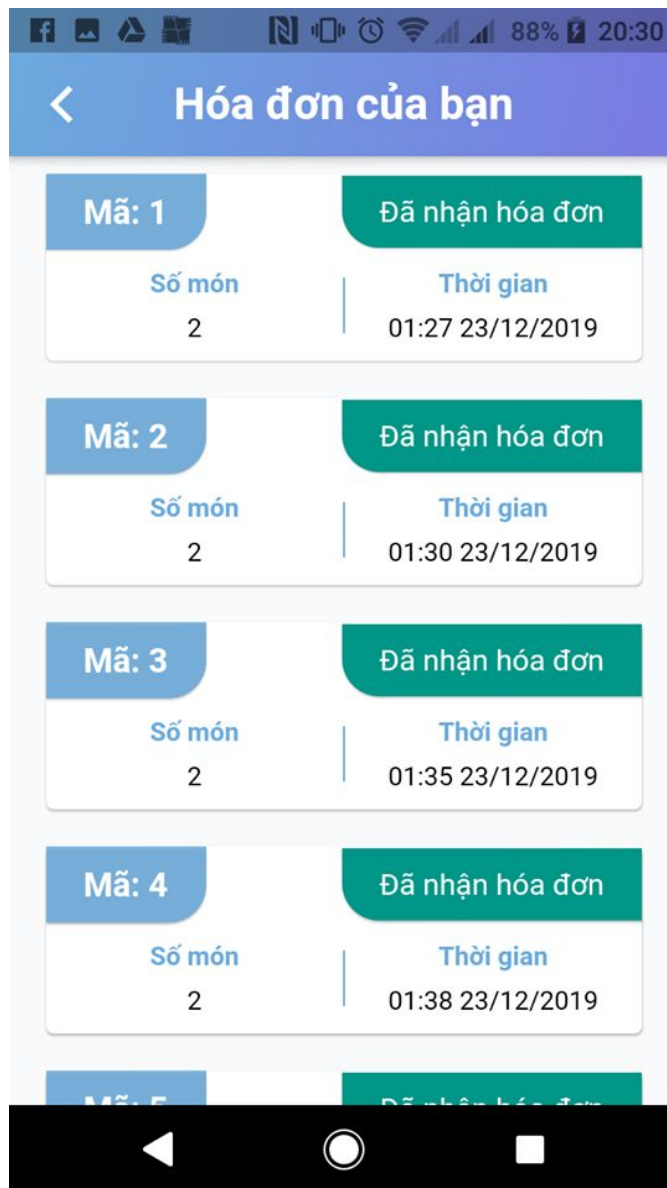
7.1.2.11 Màn hình thông tin hóa đơn

Hiển thị chi tiết hóa đơn như món ăn, số lượng từng món, tổng tiền.



7.1.2.12 Màn hình danh sách hóa đơn của tôi

Hiện thị danh sách các hóa đơn của khách hàng đã order.



7.2 Thiết kế giao diện trên mobile (staff app)

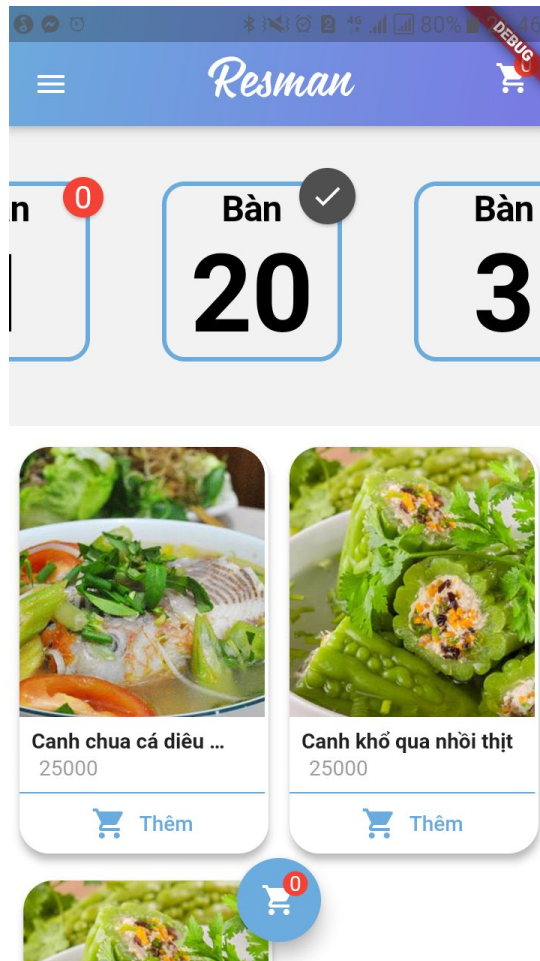
7.2.1 Danh sách các màn hình

STT	Tên trang	Loại màn hình	Chức năng
1	Trang danh sách món ăn hôm nay	Màn hình hiển thị	Hiển thị món ăn hôm nay
2	Trang thông tin món ăn	Màn hình hiển thị	Hiển thị thông tin món ăn
3	Trang giỏ hàng	Màn hình hiển thị	Hiển thị danh sách món đã chọn, thêm số lượng hoặc xoá món
4	Trang thông tin hoá đơn	Màn hình hiển thị	Hiển thị danh sách món trong hoá đơn, và thành tiền
5	Trang danh sách hoá đơn	Màn hình hiển thị	Hiển thị danh sách hoá đơn cho các loại nhân viên khác nhau
6	Trang thông tin cá nhân	Màn hình hiển thị	Hiển thị thông tin cá nhân
7	Trang sửa thông tin cá nhân	Màn hình nhập thông tin	Sửa thông tin cá nhân
8	Trang sửa mật khẩu	Màn hình nhập thông tin	Sửa mật khẩu
9	Trang đăng nhập	Màn hình nhập	Đăng nhập

7.2.2 Mô tả từng màn hình

7.2.2.1 Màn hình danh sách món ăn hôm nay

Hiển thị danh sách món ăn hôm nay kèm theo nút thêm dưới mỗi món ăn để chọn thêm vào giỏ hàng.



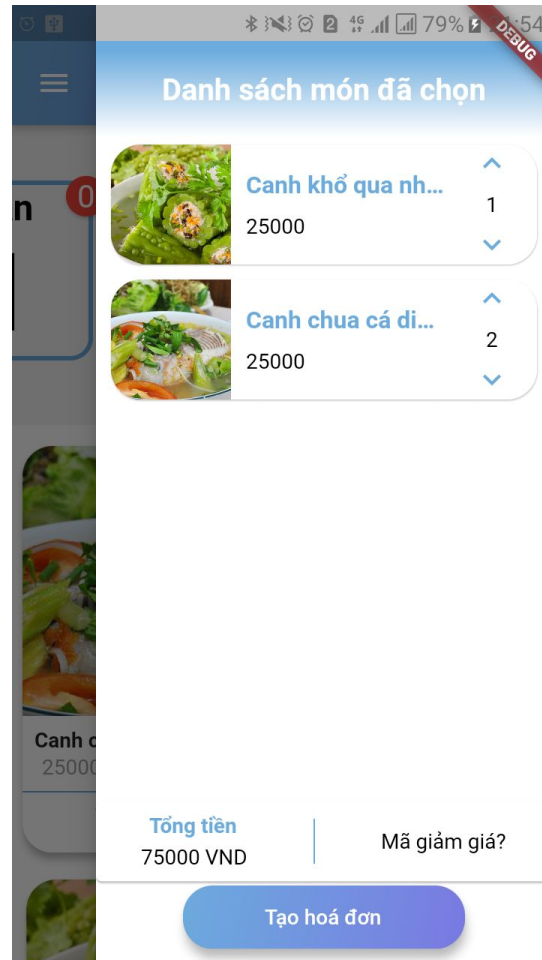
7.2.2.2 Màn hình thông tin món ăn

Hiển thị thông tin món ăn và có nút đặt món ở cuối trang để thêm vào giỏ hàng.



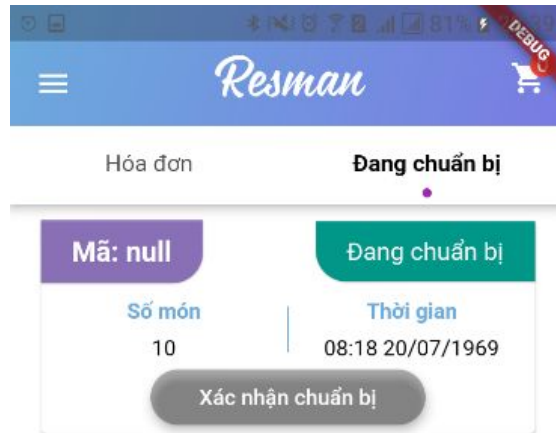
7.2.2.3 Màn hình giỏ hàng

Hiển thị danh sách các món ăn đã thêm vào giỏ hàng, và các nút tăng giảm số lượng cho mỗi món ăn. Cuối cùng là nút tạo hóa đơn để tạo hóa đơn từ các món ăn trong giỏ hàng.



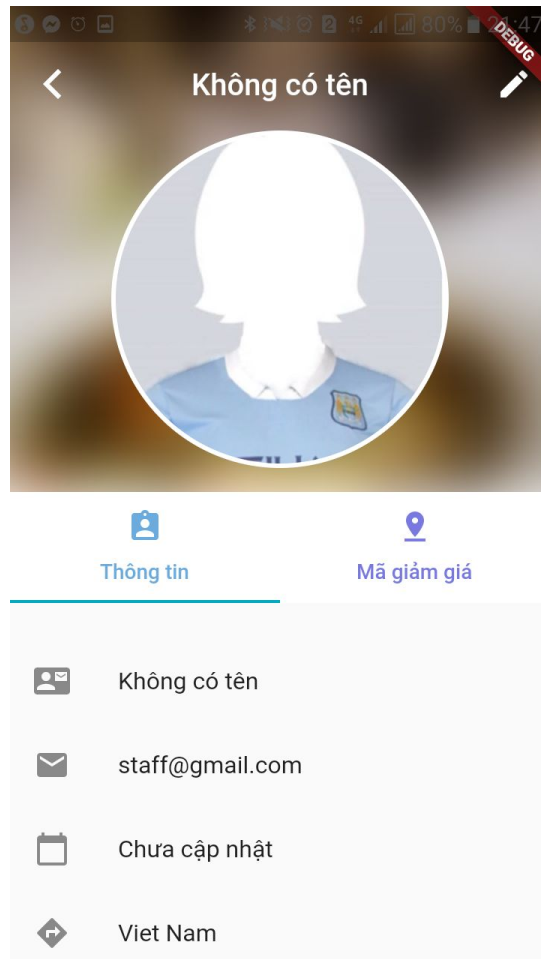
7.2.2.4 Màn hình danh sách hóa đơn và danh sách hóa đơn của bạn

Cả 2 trang đều hiển thị danh sách hóa đơn. Với trang danh sách hóa đơn chưa chuẩn bị và danh sách hóa đơn bạn đang chuẩn bị.

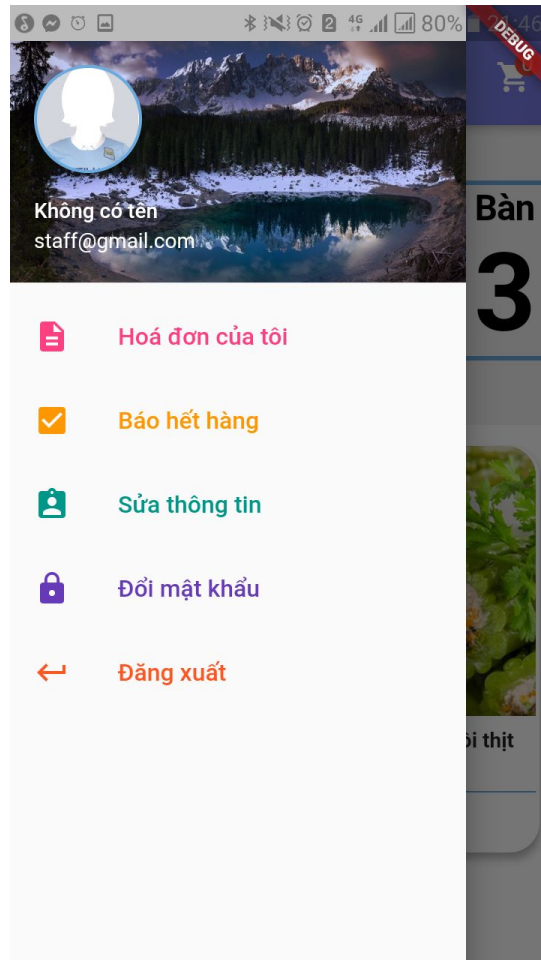


7.2.2.5 Trang thông tin cá nhân

Trang thông tin cá nhân để hiển thị thông tin cá nhân và danh sách các món yêu thích.

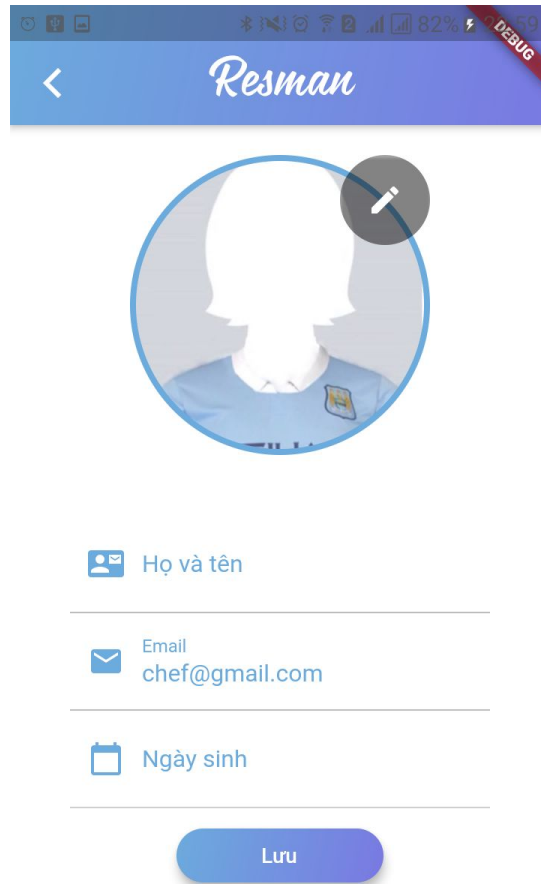


Và một drawer bar cũng hiển thị thông tin nhưng không chi tiết và các nút để trở đến các chức năng khác.



7.2.2.6 Sửa trang thông tin cá nhân

Màn hình bao gồm các trường để sửa thông tin cá nhân như email, tên hiển thị, ngày sinh, ảnh đại diện.



CHƯƠNG 8: CÔNG NGHỆ SỬ DỤNG

8.1 Flutter

8.1.1 *Flutter là gì?*

Flutter là một công cụ mới được cung cấp bởi Google cho phép các nhà phát triển xây dựng các ứng dụng đa nền tảng có thể được thực hiện trong các hệ thống khác nhau chẳng hạn như Android hay iOS chỉ với một codebase chung.

Công cụ này được xây dựng trong C và C++ và cung cấp một cơ chế rendering 2D, một functional - reactive framework là React-inspired, và một tập hợp các Material Design widget. Nó hiện đang được phát hành đến version 1.5.

Cách tiếp cận Flutter's là khác nhau từ các solution khác, ví dụ Cordova chạy trên một WebView là HTML, CSS và Javascript. Không giống như những công cụ này, nó chỉ sử dụng Dart như một ngôn ngữ lập trình duy nhất. Dart là khá dễ dàng để tìm hiểu và nếu bạn có kiến thức Java, 75% của công việc được gần như hoàn tất và làm quen với Dart sẽ chỉ mất một vài ngày.

Ứng dụng sẽ không thực thi mã Dart trực tiếp. Theo thời gian, một ứng dụng được release xây dựng, mã sẽ được biên dịch để local nhận được hiệu suất như một kết quả tốt hơn và đáp ứng giao diện người dùng tốt hơn. Trong khi phát triển trong chế độ debug (kiểm tra đối với các lỗi tiềm năng) Flutter cũng thực hiện một số nhiệm vụ mà có thể làm cho các ứng dụng chạy chậm hơn. Nếu trải qua tình trạng

này, Flutter sẽ cho biết cách đặt một dải ruy băng màu đỏ đầu ngay trong màn hình với dòng chữ "Slow Mode" ở trên đó.

8.1.2 Tại sao chọn Flutter?

Sẽ là nhiều hơn việc chỉ tạo ra cả Android và iOS ứng dụng với một dự án duy nhất, rất ít code so với các native programming ở cả hai nền tảng do Flutter high expressiveness.

Performance và đáp ứng giao diện người dùng.

Một tính năng tốt là Flutter theo định hướng Material Design và cung cấp rất nhiều thông số kỹ thuật của nó. Google cũng sử dụng Flutter để phát triển giao diện người dùng trong hệ thống mới của họ được gọi là Fuchsia.

8.1.3 Firebase Storage

Firebase Storage là dịch vụ được xây dựng cho mục đích lưu trữ và quản lý các nội dung mà người dùng ứng dụng tạo ra như ảnh, videos hay dữ liệu dạng file.

Firebase Storage cung cấp các API cho việc upload và download các file từ app của bạn một cách bảo mật và bạn không cần quan tâm đến chất lượng đường truyền mạng.

Firebase Storage được xây dựng trên nền tảng Google Cloud Platform nên có nhiều lợi thế.

Một số điểm mạnh:

- **Robust:** Firebase Storage thực hiện việc upload và download không phụ thuộc vào chất lượng đường truyền mạng hơn nữa các quá trình đó có thể bắt đầu lại khi bị tạm dừng giúp tiết kiệm thời gian và băng thông.
- **Secure:** Được tích hợp Firebase Authentication cho việc bảo mật nên dễ dàng quản lý quyền truy cập vào các files.
- **Scalable:** Firebase Storage được xây dựng trên nền tảng Google Cloud Platform nên khả năng mở rộng có thể lên đến hàng Petabyte dữ liệu. Cả hai nền tảng ứng dụng desktop và mobile đều sử dụng Firebase Storage để lưu trữ hình ảnh như ảnh đại diện của người dùng, các hình ảnh của món ăn.

8.2 ExpressJs Framework

Express js là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Nó rất dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web.

Expressjs có vô số các package hỗ trợ nên các bạn không phải lo lắng khi làm việc với Framework này.

Về performance: Express cung cấp thêm về các tính năng (feature) để dev lập trình tốt hơn. Chứ không làm giảm tốc độ của NodeJS.

Và hơn hết, các Framework nổi tiếng của NodeJS hiện nay đều sử dụng ExpressJS như một core function, chẳng hạn: SailsJS, MEAN...

8.3 Object Relational Mapping (Orm) - TypeOrm

8.3.1 *Orm là gì?*

Trong lập trình hướng đối tượng chúng ta tương tác với các thành phần dữ liệu (Model) là các đối tượng trong khi hầu hết các cơ sở dữ liệu đang sử dụng phổ biến hiện nay lại đang là CSDL quan hệ. Mà CSDL quan hệ lưu dữ liệu theo dòng (bản ghi – Record) trong các bảng dữ liệu (table). Do vậy cần một kỹ thuật chuyển đổi giữa hai mô hình này để quá trình lập trình được thuận lợi hơn.

ORM là kỹ thuật chuyển đổi dữ liệu giữa các hệ thống khác (không phải là mô hình hướng đối tượng) sang các đối tượng trong ngôn ngữ lập trình hướng đối tượng. Trong trường hợp này chúng ta chuyển dữ liệu từ các dòng dữ liệu trong CSDL quan hệ sang đối tượng.

Để cài đặt kỹ thuật này đơn giản là chúng ta cài đặt một lớp thực hiện việc kết nối CSDL và các chức năng tương tác với CSDL như Select, Insert, Delete, Update và các tương tác trên các đối tượng dữ liệu có quan hệ với nhau.

8.3.2 *Tại sao nên chọn Orm*

Portable – tính năng động: ORM được sử dụng để bạn viết cấu trúc của bạn một lần và lớp ORM sẽ xử lý câu lệnh cuối cùng phù hợp với DBMS được cấu hình. Đây là một lợi thế tuyệt vời khi thao tác truy xuất dữ liệu đơn giản như giới hạn được thêm vào dưới dạng ‘limit 0,100’ ở cuối câu lệnh Select trong MySQL, trong khi đó với cách truy xuất thông thường bạn phải viết là ‘Select Top 100 From Table’ trong MS SQL.

Nesting Of Data – truy xuất lồng dữ liệu: trong trường hợp database bạn có nhiều bảng và các bảng này liên hệ rồi rắc phức tạp về dữ liệu thì ORM sẽ tự

động lấy dữ liệu cho bạn một cách đơn giản (ở đây đang nói tới việc đơn giản trong truy xuất dữ liệu, không bàn về vấn đề tối ưu truy xuất).

Single Language – không cần biết SQL: thật vậy với nguyên lý thiết kế là ánh xạ toàn bộ dữ liệu lấy được từ DBMS sang bộ nhớ nên việc thao tác truy xuất bây giờ chỉ phụ thuộc vào ngôn ngữ lập trình bạn đang sử dụng, bạn chẳng cần quan tâm phía đằng sau của ORM sẽ làm gì sinh ra mã SQL như thế nào khi truy xuất SQL, và kết quả là chúng ta chỉ cần nhuần nhuyễn ngôn ngữ lập trình đang dùng.

Adding is like modifying – thêm sửa dữ liệu là như nhau: đối với ORM, nó không phân biệt giữa thêm mới và cập nhật mọi tác vụ có liên quan đến sửa đổi hay chèn dữ liệu đều được xem là định nghĩa thêm mới, hai tác vụ này được xem như là một.

8.3.3 Trở ngại khi dùng Orm

Slow – chậm: trong tất cả các bài đánh giá và so sánh về tốc độ truy xuất dữ liệu bạn sẽ thấy ORM dù cho được tối ưu đến đâu cũng sẽ không bao giờ nhanh hơn được so với truy vấn trực tiếp từ SQL.

Tunning – khó điều chỉnh: Tối ưu hóa các câu lệnh SQL là điều nên làm đối với các dữ liệu lớn, tuy nhiên đối với ORM quá trình này rất khó khăn và phức tạp, bởi vì bạn phải tuân theo thiết kế của ORM và ưu điểm mà nó mang lại là hạn chế và cô lập quá trình thao tác dữ liệu trực tiếp lên DBMS.

Complex Queries – truy vấn phức tạp: mặc dù được thiết kế để đơn giản hóa tuy nhiên một vài ORM thao tác dữ liệu rất rối rắm, các đoạn code rất phức tạp và khó hiểu hơn nhiều lần so với SQL thuần.

Studying – học tập phức tạp: Đa phần các hệ thống với các tác vụ nặng và phức tạp thư viện ORM thuần túy được cung cấp không đảm bảo về hiệu năng, và bạn cùng các thành viên trong team chắc chắn phải nghiên cứu từng dòng code để tối ưu cho hệ thống của mình.

8.3.4 TypeOrm là gì?

TypeOrm là một Orm có thể chạy trên nền tảng NodeJs, Browser Cordova, PhoneGap, Ionic, React Native, NativeScript, Expo và Electron platforms. Nó có thể viết bằng Typescript hoặc Javascript (ES5, ES6, ES7, ES8).

Mục tiêu là luôn cập nhật để hỗ trợ những phiên bản mới nhất của Javascript và cung cấp những tính năng mới để giúp các nhà phát triển phát triển bất kỳ loại phần mềm nào sử dụng database từ phần mềm nhỏ với vài bảng tới những phần mềm lớn với hàng trăm bảng.

TypeOrm hỗ trợ cả Active Record và Data Mapper pattern, không giống như các loại Orm javascript khác hiện nay. Điều này có nghĩa là bạn có thể ghi dữ liệu nhanh hơn, dễ dàng mở rộng và duy trì hơn.

Typeorm chịu ảnh hưởng nhiều từ các loại Orm khác như Hibernate, Doctrine và Entity Framework.

8.3.5 Tại sao chọn TypeOrm

Lý do chọn TypeOrm mà không chọn các Orm khác cũng rất phổ biến với Nodejs như Sequence là:

- TypeOrm hỗ trợ hoàn toàn cho TypeScript và project này được xây dựng và thiết lập để code, test và deploy hoàn toàn bằng Typescript nên nó rất phù hợp so với Sequence. Bởi vì sequence chưa có hỗ trợ dành cho Typescript.
- TypeOrm hỗ trợ rất nhiều nền tảng Sql khác nhau, có tài liệu đầy đủ và chi tiết, tốc độ truy xuất cao, và đặc biệt là rất dễ học và cài đặt.
- TypeOrm hỗ trợ tính năng Migration. Một tính năng vô cùng hữu hiệu để quản lý các phiên bản của cơ sở dữ liệu.

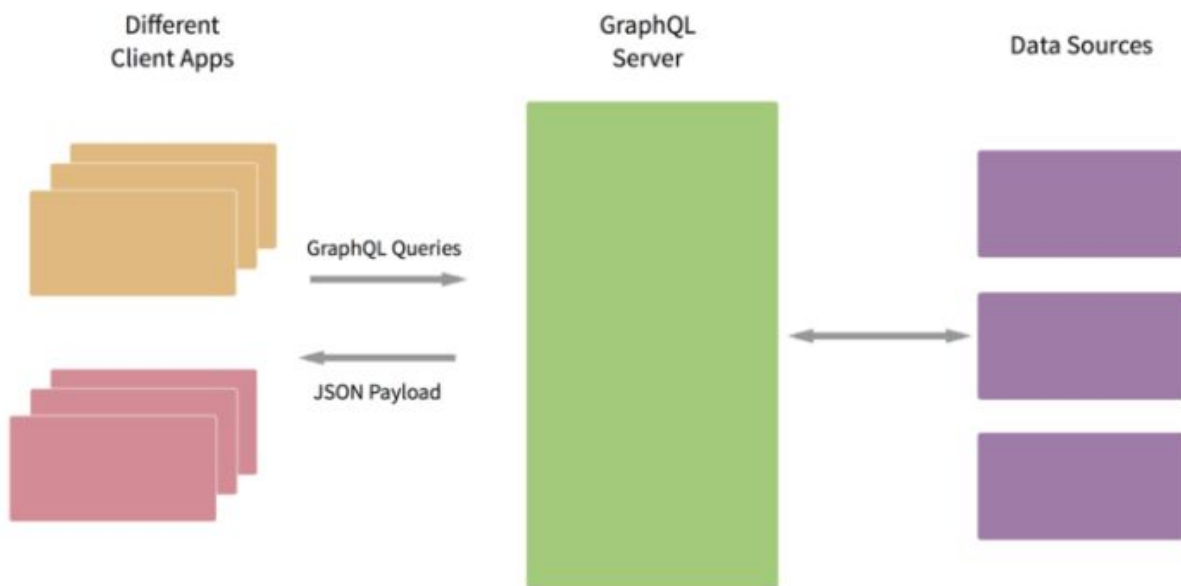
8.4 GraphQL

8.4.1 GraphQL là gì?

Được Facebook giới thiệu cùng với Relay tại React.js Conf 2015, GraphQL là một ngôn ngữ query cho API dùng để viết các câu API một cách uyển chuyển chính xác những gì cần có, trong các application hiện đại nhiều lớp với nhiều thành phần phụ thuộc được sử dụng ngày càng phổ biến hiện nay. Trên thực tế GraphQL đã được sử dụng vài năm trước trong ứng dụng trên mobile của Facebook.

8.4.2 Đặc điểm của GraphQL

Client truy vấn đến máy chủ GraphQL bằng các truy vấn với đặc điểm: format của dữ liệu trả về được mô tả trong câu truy vấn và được định nghĩa ở phía client thay vì ở server. Nói đơn giản hơn, đây là truy vấn hướng client, cấu trúc dữ liệu không khô cứng 1 khuôn mẫu từ server (REST API) mà thay đổi theo từng ngữ cảnh sao cho hiệu quả nhất đối với client mà chỉ cần dùng duy nhất 1 endpoint.



8.4.3 Lợi ích của GraphQL

Lấy Facebook làm ví dụ. Facebook phải quản lý vô số data source và API clients mà **REST API** lại lộ khuyết điểm thiếu linh hoạt do tính chất dựa trên tài nguyên cố định, dẫn đến trường hợp “nghẽn cổ chai” thường thấy. Chính vì vậy, thay vì có đến hàng tá “endpoint” dư thừa, Facebook đã nghĩ đến giải pháp chỉ dùng một “endpoint” thông minh với khả năng tiếp thu những Query phức tạp rồi đưa ra output data với loại type tùy theo yêu cầu của client. Nói đơn giản hơn, đây là truy vấn hướng client, cấu trúc dữ liệu không khô cứng 1 khuôn mẫu từ server (**REST API**) mà thay đổi theo từng ngữ cảnh sao cho hiệu quả nhất đối với client. Ngoài ra, GraphQL hoàn toàn có thể create, update, delete, nhưng với cấu trúc sáng sủa và cấu trúc phân tầng nên lại càng thuận lợi cho lập trình viên phía client.

8.5 Socket.io

Socket.io là gì?

Để xây dựng một ứng dụng realtime cần sử dụng socket.io. Socket.io sẽ giúp các bên ở những địa điểm khác nhau kết nối với nhau, truyền dữ liệu ngay lập tức thông qua server trung gian. Socket.io có thể được sử dụng trong nhiều ứng dụng như chat, game online, cập nhật kết quả của một trận đấu đang xảy ra,...

Socket.io không phải là một ngôn ngữ, mà chỉ là 1 công cụ giúp thực hiện những ứng dụng realtime. Vì thế, không thể sử dụng socket.io để thay thế hoàn toàn cho một ngôn ngữ, mà phải sử dụng kết hợp với một ngôn ngữ khác. Ngôn ngữ đó có thể là php, **asp.net**, nodejs,...

8.6 Heroku

8.6.1 Heroku là gì?

Heroku là nền tảng đám mây cho phép các công ty xây dựng, phân phối, theo dõi và mở rộng ứng dụng - Heroku cung cấp cách nhanh nhất để chuyển từ ý tưởng sang URL, bỏ qua tất cả những cơ sở hạ tầng phức tạp khó hiểu.

Heroku là cách nhanh nhất để trở thành một công ty ứng dụng. Heroku là một dịch vụ cho phép công ty dành thời gian để phát triển và triển khai các ứng dụng ngay lập tức để tạo ra giá trị.

Heroku là một trải nghiệm tuyệt vời dành cho developer. Heroku giúp các developer chỉ cần tập trung vào chuyên môn của mình đó là phát triển ứng dụng.

8.6.2 Đặc điểm của heroku

Bảo mật: Heroku được xây dựng trên sự tin tưởng và bảo mật. Sự tin cậy và minh bạch là nguyên tắc cốt lõi tại Heroku. Heroku cung cấp khả năng theo dõi threat liên tục, vá lỗ hổng hệ điều hành tự động và vận hành ops suốt ngày đêm. Hơn 23 tỷ yêu cầu mỗi ngày, đã tạo hơn 7 triệu ứng dụng, hơn 150 dịch vụ hỗ trợ.

Dữ liệu: Heroku là một nền tảng dành cho data và apps - cung cấp một database-as-a-service an toàn, có khả năng mở rộng với rất nhiều công cụ dành cho nhà phát triển như database followers, forking, dataclip và health checks tự động. Dữ liệu giữ một vai trò quan trọng của bất kỳ ứng dụng nào, từ dữ liệu khách hàng đến các dữ liệu về dịch vụ, ứng dụng và dữ liệu luôn đi đôi với nhau. Hệ sinh thái phong phú của Heroku bao gồm Heroku Postgres - một dịch vụ cơ sở dữ liệu được tích hợp sẵn. Các developer không cần phải tìm hiểu cách cung cấp cơ sở dữ liệu tối ưu thông qua trial và error, thay vào đó họ có quyền truy cập

ngay lập tức vào cơ sở dữ liệu có khả năng mở rộng cao với tính năng rollback hỗ trợ ứng dụng và development style.

Một hệ sinh thái dịch vụ: Heroku là một hệ sinh thái của các dịch vụ đám mây, có thể được sử dụng ngay lập tức để mở rộng các ứng dụng với các dịch vụ được quản lý hoàn toàn. Sử dụng một dịch vụ chất lượng cao giúp nhà phát triển có thể xây dựng nhiều hơn, nhanh hơn bằng cách sử dụng các dịch vụ đáng tin cậy cung cấp các chức năng mà họ cần. Heroku Add-ons là các dịch vụ được quản lý hoàn toàn, được tích hợp để sử dụng với Heroku. Chúng được cung cấp và thu nhỏ trong một lệnh và cho phép các nhà phát triển mở rộng khả năng của một ứng dụng. Hơn 150 Tiện ích có sẵn, cung cấp các dịch vụ từ cơ sở dữ liệu đến cảnh báo thời gian hoạt động, dịch vụ nhắn tin hoặc sao lưu tự động nhằm thực hiện search, metrics hoặc mail delivery.

8.6.3 Tại sao chọn Heroku

- Heroku có cung cấp một gói dịch vụ free với chất lượng rất tốt và rất phù hợp cho các đồ án môn học của sinh viên.
- Heroku cũng hỗ trợ sẵn cho các dịch vụ dữ liệu. Ở đây dự án này sử dụng Postgresql được Heroku hỗ trợ sẵn và tối ưu với các biến môi trường cung cấp sẵn cho dự án.
- Heroku có tính năng tự động deploy app mỗi khi bạn có thay đổi. Điều này cực kỳ hữu ích đối với các developer để tập trung vào code mà không phải lo đến các vấn đề khác.
- Heroku hỗ trợ cực kỳ tốt với Nodejs và cũng hỗ trợ tốt với nodejs được thiết lập để viết với typescript.

CHƯƠNG 9: KẾT QUẢ ĐẠT ĐƯỢC

9.1 Kết quả

Sau quá trình làm phân tích, thiết kế và triển khai dự án trong vòng 3 tháng, kết quả đạt được của cả nhóm là:

- Đáp ứng được đầy đủ các tính năng chính.
- Nắm quy trình và nghiệp vụ quản lý, cụ thể ở đây là quản lý chuỗi nhà hàng.
- Nắm được một số công nghệ mới, ngôn ngữ mới và một số framework
- Sử dụng các công cụ quản lý công việc như trello, github
- Nắm được kiến thức đặc tả, phân tích yêu cầu phần mềm

9.2 Hạn chế

- Giao diện và logic vẫn còn nhiều thiếu sót.
- Một số tính năng phức tạp vẫn chưa được đưa vào đồ án, do có ít thời gian và kiến thức chưa đủ, cần tìm hiểu thêm.
- Chưa kiểm tra được tính thực tiễn khi đưa vào sử dụng.

9.3 Hướng phát triển

- Xây dựng, cải tiến giao diện thân thiện, đẹp mắt hơn.
- Áp dụng quét mã QR để tương tác giữa nhân viên và khách hàng, áp dụng thanh toán trực tuyến.
- Giải quyết các vấn đề logic mà nhóm mắc phải mà chưa thể giải quyết được trong thời gian thực hiện đồ án.

CHƯƠNG 10: PHÂN CÔNG CÔNG VIỆC

	Lê Hồng Hiền	Phan Thanh Duy	Nguyễn Trung Nguyên
Phân tích nghiệp vụ, yêu cầu	✓	✓	✓
Thiết kế cơ sở dữ liệu	✓	✓	✓
Cài đặt cơ sở dữ liệu	✓		
Viết báo cáo	✓	✓	✓
Thiết kế Diagram			
Use case	✓	✓	✓
Đặc tả use-case	✓	✓	✓
Xây dựng Web admin			
Front-end	✓		
Back-end	✓		
Xây dựng Resman-mobile-customer			
Front-end	✓		✓
Back-end	✓		✓
Xây dựng Resman-mobile-staff			
Front-end		✓	
Back-end		✓	
Deploy Server			
Heroku	✓		

REFERENCE

Avoid inspect source reactjs

https://www.reddit.com/r/reactjs/comments/8bagkp/how_can_i_hide_my_source_code/

<https://stackoverflow.com/questions/51984146/how-to-disable-source-maps-for-react-js-application>

Authentication

<https://techtalk.vn/authentication-trong-spa-reactjs-vuejs-nen-luu-token-o-dau-co-nhung-van-de-lien-quan-den-security.html>

Restaurant

<https://www.cet.edu.vn/quan-tri-nha-hang-khach-san/ky-nang/quy-trinh-va-nghiep-vu-phuc-vu-cho-nhan-vien-nha-hang>

Online payment

<https://github.com/naustudio/node-vn-payments>

Express Unit Testing

<https://gitlab.com/hienlh/restaurant-management/wikis/Express-Unit-Testing>

Flutter

<https://flutter.dev/>

<https://medium.com/flutter-community/parsing-complex-json-in-flutter-747c46655f51>

Socket.io

<https://socket.io>

GraphQL

<https://graphql.org/learn/>

<https://www.apollographql.com/>