

**KHOA CÔNG NGHỆ KỸ THUẬT -
NÔNG NGHIỆP CÔNG NGHỆ CAO**



BARIA VUNGTAU
UNIVERSITY
CAP SAINT JACQUES

BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
ĐỀ TÀI: XÂY DỰNG HỆ THỐNG ĐẶT MÓN ĂN CHO NHÂN
VIÊN TẠI NHÀ HÀNG

Trình độ đào tạo	: Đại học
Ngành	: Công nghệ thông tin
Chuyên ngành	: Lập trình ứng dụng di động và game
Khóa học	: 2018-2022
Lớp	: DH18LT
Sinh viên thực hiện	: Nguyễn Thanh Long
Mã số sinh viên	: 18033886
GVHD	: TS. Phan Ngọc Hoàng

Tp. Vũng Tàu, ngày 01 tháng 12 năm 2021

LỜI CẢM ƠN

Trước khi bắt đầu nội dung báo cáo, em xin phép gửi lời cảm ơn chân thành đến cha mẹ và gia đình đã có công sinh thành, phụng dưỡng, khuyên nhủ và bảo bọc để em có thể vững bước trên con đường học tập mà có kiến thức vững chắc để bước ra đời.

Em cũng xin gửi đến quý thầy cô tại trường đại học Bà Rịa Vũng Tàu nói riêng và trong khoa Công nghệ kỹ thuật – Nông nghiệp công nghệ cao nói riêng, thầy Hoàng, thầy Phương, thầy Trì, thầy Tình, cô Hà, cô Trang, cô Nương lời cảm ơn chân thành nhất.

Và hơn nữa, em xin gửi đến thầy Phan Ngọc Hoàng, người thầy đã hỗ trợ em trong suốt quãng đường đại học. Thầy Hoàng đã dẫn dắt và hướng dẫn tận tình em phát triển, xây dựng đề án cũng như hoàn thành bài báo cáo tốt nghiệp một cách tốt nhất.

Đồng thời xin chân thành cảm ơn Ban lãnh đạo trường Đại học Bà Rịa – Vũng Tàu đã tạo điều kiện thật tốt cho chúng em môi trường học tập lành mạnh và năng động.

Vì kiến thức bản thân còn hạn chế và trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp thầy, cô để em học thêm được nhiều kinh nghiệm và sẽ hoàn thiện bản thân đáp ứng với nhu cầu công việc thực tế trong tương lai.

Và lời cuối em xin chân thành cảm ơn!

LỜI CAM ĐOAN

Tôi xin cam đoan kết quả đạt được trong đề án là sản phẩm của riêng cá nhân, không sao chép lại của người khác. Trong toàn bộ nội dung của đề án, những điều được trình bày hoặc là của cá nhân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Vũng Tàu, ngày 01 tháng 12 năm 2021

Sinh viên thực hiện

LỜI NÓI ĐẦU

Trong thế giới ngày nay, công nghệ thông tin (CNTT) là một công cụ quan trọng hỗ trợ đắc lực cho ngành dịch vụ. Gặt hái lợi ích tối đa từ các khoản đầu tư vào công nghệ để duy trì khả năng cạnh tranh và nâng cao chất lượng dịch vụ là một trong những ưu tiên cao nhất của những cá nhân, tổ chức phát triển ngành dịch vụ ngày nay.

Thực tại hầu hết các địa điểm bán đồ ăn vừa và nhỏ hiện chưa ứng dụng CNTT vào ngành dịch vụ của mình hoặc ứng dụng công nghệ lỗi thời chưa tối ưu được chất lượng dịch vụ như công nghệ máy bán hàng POS chỉ được đặt tại một số vị trí cố định không đảm bảo tính di động và linh hoạt. Thực tế khi thực khách chọn món ăn, thức uống trên thực đơn, nhân viên phục vụ phải đứng chờ để ghi chép. Hoặc sau khi chọn xong, khách hàng phải chờ gọi phục vụ đến bàn để đọc món ăn mà mình đã chọn. Việc ghi chép các món ăn của thực khách có thể sai sót dẫn đến tranh cãi sau này và gây mất uy tín của nhà hàng.

Sau khi ghi món xong, nhân viên phục vụ phải đích thân mang danh sách món ăn xuống nhà bếp. Tại nhà bếp sẽ có người quản lý để phân phối thông tin các món ăn cho đầu bếp. Khi món ăn được thực hiện xong, phải gọi cho nhân viên phục vụ để mang thức ăn cho khách. Quãng đường đi lại nhiều lần gây mất thời gian và bố trí thêm nhân lực phục vụ khách ngoài sảnh. Tất cả công việc này nếu không được tổ chức tốt sẽ gây ách tắc, dễ nhầm lẫn, khó kiểm soát và mất thời gian.

Thực tế đã có nhiều hệ thống nhà hàng đã áp dụng công nghệ thông tin vào hệ thống đặt món ăn. Tuy những hệ thống ấy đem lại rất nhiều hiệu quả và đã khắc phục hầu hết các nhược điểm của quá trình đặt món ăn truyền thống nhưng lại có một nhược điểm lớn đó là tính di động. Hệ thống đặt món ăn hiện đại chủ yếu là những máy tính được đặt cố định một vị trí nên nhân viên nhà hàng khó linh hoạt và di động đặt món ăn cho khách tại bàn.

Vì lẽ đó, để giải quyết những hạn chế trong phong cách phục vụ truyền thống cũng như giải quyết vấn đề về tính linh hoạt của hệ thống đặt món ăn hiện nay, đề tài “*Xây dựng hệ thống đặt món ăn cho nhân viên tại nhà hàng*” được ra đời. Hướng tới những cá nhân và tổ chức có nhu cầu phát triển và nâng cấp chất lượng phục vụ trong ngành dịch vụ bán đồ ăn. Hệ thống đảm bảo khách hàng sẽ được nhân viên phục vụ đặt món tại chỗ một cách nhanh chóng, tiện lợi và chính xác.

MỤC LỤC

DANH MỤC BẢNG.....	10
DANH MỤC SƠ ĐỒ.....	12
DANH MỤC HÌNH.....	14
CHƯƠNG I: TỔNG QUAN	19
1. Lý do chọn đề tài.....	19
2. Mục tiêu và chức năng của hệ thống	19
2.1. Mục tiêu.....	19
2.2. Chức năng	19
3. Mô tả bài toán	20
CHƯƠNG II: CƠ SỞ LÝ THUYẾT	22
1. Cấu trúc hệ thống	22
1.1. Mô hình kết nối	22
1.2. Cấu trúc dữ liệu	22
1.3. Các tiêu chuẩn	23
1.4. Phương thức API.....	23
2. Công nghệ sử dụng.....	26
2.1. Flutter và Dart	27
2.1.1. Ngôn ngữ lập trình Dart [1]	27
2.1.2. Flutter framework [2].....	28
2.2. Python và Django	29
2.2.1. Ngôn ngữ lập trình Python [3].....	29
2.2.2. Django framework [4]	31
2.3. Hệ quản trị cơ sở dữ liệu SQLite [5].....	33
2.3.1. Tổng quan.....	33
2.3.2. Tại sao lại chọn SQLite?	33
2.4. Firebase	34
2.4.1. Tổng quan [6].....	34
2.4.2. Firebase Cloud Firestore [7]	34
2.4.3. Firebase Cloud Messaging [6].....	34

CHƯƠNG III: PHÂN TÍCH HỆ THỐNG	35
1. Phân tích yêu cầu đề bài	35
1.1. Yêu cầu người dùng	35
1.2. Yêu cầu Website quản trị	35
1.3. Ứng dụng di động	38
2. Phân tích hệ thống	39
2.1. Website quản trị	39
2.2. Ứng dụng di động	56
CHƯƠNG IV: THIẾT KẾ CƠ SỞ DỮ LIỆU	84
1. Sơ đồ cơ sở dữ liệu	84
2. Chi tiết cơ sở dữ liệu	84
CHƯƠNG V: XÂY DỰNG HỆ THỐNG	89
1. Xây dựng Website quản trị sử dụng Django	89
1.1. Giao diện đăng nhập	89
1.2. Giao diện chính	90
1.3. Giao diện chung của các danh mục	92
1.4. Giao diện Users (Người dùng)	98
1.5. Giao diện Tables (Bàn ăn)	105
1.6. Giao diện Send Messages (Lời nhắn)	110
1.7. Giao diện Promotions (Chương trình khuyến mãi)	115
1.8. Giao diện Categories (Loại sản phẩm)	119
1.9. Giao diện Labels (Nhãn sản phẩm)	124
1.10. Giao diện Variations (Biến thể sản phẩm)	128
1.11. Giao diện Products (Sản phẩm)	135
1.12. Giao diện Extras (Sản phẩm gọi thêm)	144
1.13. Giao diện Orders (Đơn hàng)	148
1.14. Các giao diện khác	152
2. Xây dựng ứng dụng di động	153
2.1. Tổng quan	153
2.2. Giao diện Welcome	153
2.3. Giao diện Login (Đăng nhập)	154

2.4.	Giao diện Home (Trang chủ).....	156
2.5.	Giao diện Order (Bàn ăn).....	158
2.6.	Giao diện đơn hàng.....	158
2.7.	Giao diện menu.....	159
2.8.	Giao diện chi tiết món ăn.....	162
2.9.	Giao diện giỏ hàng.....	164
2.10.	Giao diện chi tiết đơn hàng.....	166
2.11.	Giao diện hủy món.....	167
2.12.	Giao diện thanh toán.....	168
2.13.	Các giao diện chức năng xử lý đơn hàng khác.....	169
2.14.	Giao diện Chat (Nhắn tin).....	172
2.15.	Giao diện Profile.....	174
2.16.	Giao diện My Reports.....	175
2.17.	Giao diện Reports.....	177
	CHƯƠNG VI: KẾT LUẬN.....	178
1.	Kết quả đạt được.....	178
2.	Hướng phát triển.....	178
	TÀI LIỆU THAM KHẢO.....	179
	PHỤ LỤC.....	180
1.	Sơ đồ cấu trúc thư mục.....	180
1.1.	Website quản trị.....	180
1.2.	Ứng dụng di động.....	182
1.	Code xử lý.....	183
1.1.	Website quản trị.....	183
1.2.	Ứng dụng di động.....	192

DANH MỤC BẢNG

Bảng 1	Danh sách mã response API	25
Bảng 2	Danh sách chức năng Website quản trị	35
Bảng 3	Danh sách chức năng quản lý người dùng	36
Bảng 4	Danh sách chức năng quản lý loại sản phẩm	36
Bảng 5	Danh sách chức năng quản lý biến thể sản phẩm	36
Bảng 6	Danh sách chức năng quản lý nhãn sản phẩm	37
Bảng 7	Danh sách chức năng quản lý sản phẩm	37
Bảng 8	Danh sách chức năng quản lý sản phẩm gọi thêm	37
Bảng 9	Danh sách chức năng quản lý chương trình khuyến mãi	37
Bảng 10	Danh sách chức năng quản lý bàn ăn	37
Bảng 11	Danh sách chức năng quản lý lời nhắn	38
Bảng 12	Danh sách chức năng quản lý đơn hàng	38
Bảng 13	Danh sách chức năng của nhân viên	38
Bảng 14	Danh sách chức năng của quản lý	38
Bảng 15	Danh sách chức năng chung	39
Bảng 16	Danh sách các trường dữ liệu người dùng	39
Bảng 17	Danh sách các trường dữ liệu bàn ăn	40
Bảng 18	Danh sách các trường dữ liệu lời nhắn	40
Bảng 19	Danh sách các trường dữ liệu chương trình khuyến mãi	41
Bảng 20	Danh sách các trường dữ liệu loại sản phẩm	41
Bảng 21	Danh sách các trường dữ liệu nhãn sản phẩm	41
Bảng 22	Danh sách các trường dữ liệu biến thể sản phẩm	42
Bảng 23	Danh sách các trường dữ liệu sản phẩm	42
Bảng 24	Danh sách các trường dữ liệu sản phẩm gọi thêm	43
Bảng 25	Danh sách các trường dữ liệu đơn hàng	43
Bảng 26	Cơ sở dữ liệu quản lý người dùng	84
Bảng 27	Cơ sở dữ liệu quản lý bàn ăn	85
Bảng 28	Cơ sở dữ liệu quản lý loại sản phẩm	85
Bảng 29	Cơ sở dữ liệu quản lý nhãn	85
Bảng 30	Cơ sở dữ liệu quản lý loại biến thể	85
Bảng 31	Cơ sở dữ liệu quản lý biến thể	86
Bảng 32	Cơ sở dữ liệu quản lý sản phẩm	86
Bảng 33	Cơ sở dữ liệu quản lý sản phẩm biến thể	86
Bảng 34	Cơ sở dữ liệu quản lý đơn hàng	86
Bảng 35	Cơ sở dữ liệu quản lý chi tiết đơn hàng	86
Bảng 36	Cơ sở dữ liệu quản lý sản phẩm gọi thêm chi tiết đơn hàng	87
Bảng 37	Cơ sở dữ liệu quản lý chương trình khuyến mãi	87
Bảng 38	Cơ sở dữ liệu quản lý lời nhắn	87
Bảng 39	Giải thích cấu trúc thư mục website quản trị	181

Bảng 40 Giải thích cấu trúc thư mục ứng dụng di động182

DANH MỤC SƠ ĐỒ

Sơ đồ 1 Mô hình kết nối website và ứng dụng di động.....	22
Sơ đồ 2 Quy trình xử lý phương thức Resful API.....	22
Sơ đồ 3 Sơ đồ usecase website quản trị của quản lý.....	44
Sơ đồ 4 Sơ đồ tuần tự chức năng đăng nhập website quản trị.....	45
Sơ đồ 5 Sơ đồ hợp tác chức năng đăng nhập website quản trị	45
Sơ đồ 6 Sơ đồ tác nhân - usecase chức năng truy cập danh mục	46
Sơ đồ 7 Sơ đồ tuần tự chức năng truy cập danh mục	47
Sơ đồ 8 Sơ đồ hợp tác chức năng truy cập danh mục.....	47
Sơ đồ 9 Sơ đồ tác nhân - usecase chức năng thêm.....	48
Sơ đồ 10 Sơ đồ tuần tự chức năng thêm.....	48
Sơ đồ 11 Sơ đồ hợp tác chức năng thêm	49
Sơ đồ 12 Sơ đồ tác nhân - usecase chức năng tìm kiếm.....	49
Sơ đồ 13 Sơ đồ tuần tự chức năng tìm kiếm.....	50
Sơ đồ 14 Sơ đồ hợp tác chức năng tìm kiếm	50
Sơ đồ 15 Sơ đồ tác nhân - usecase chức năng hiển thị	51
Sơ đồ 16 Sơ đồ tuần tự chức năng hiển thị	51
Sơ đồ 17 Sơ đồ tác nhân - usecase chức năng chỉnh sửa	52
Sơ đồ 18 Sơ đồ tuần tự chức năng chỉnh sửa.....	53
Sơ đồ 19 Sơ đồ hợp tác chức năng chỉnh sửa	53
Sơ đồ 20 Sơ đồ tác nhân - usecase chức năng xóa.....	54
Sơ đồ 21 Sơ đồ tuần tự chức năng xóa	54
Sơ đồ 22 Sơ đồ hợp tác chức năng xóa	55
Sơ đồ 23 Sơ đồ tác nhân - usecase chức năng xuất báo cáo	55
Sơ đồ 24 Sơ đồ tuần tự chức năng xuất báo cáo	56
Sơ đồ 25 Sơ đồ hợp tác chức năng xuất báo cáo.....	56
Sơ đồ 26 Sơ đồ usecase ứng dụng di động của quản lý.....	58
Sơ đồ 27 Sơ đồ usecase ứng dụng di động của nhân viên	58
Sơ đồ 28 Sơ đồ tuần tự chức năng đăng nhập ứng dụng di động.....	59
Sơ đồ 29 Sơ đồ hợp tác chức năng đăng nhập ứng dụng di động	60
Sơ đồ 30 Sơ đồ tác nhân - usecase chức năng truy cập bàn ăn	60
Sơ đồ 31 Sơ đồ tuần tự chức năng truy cập bàn ăn.....	61
Sơ đồ 32 Sơ đồ hợp tác chức năng truy cập bàn ăn.....	61
Sơ đồ 33 Sơ đồ tác nhân - usecase chức năng thêm vào giỏ hàng.....	62
Sơ đồ 34 Sơ đồ tuần tự chức năng thêm vào giỏ hàng	63
Sơ đồ 35 Sơ đồ hợp tác chức năng thêm vào giỏ hàng.....	63
Sơ đồ 36 Sơ đồ tác nhân - usecase chức năng chỉnh sửa giỏ hàng	64
Sơ đồ 37 Sơ đồ tuần tự chức năng chỉnh sửa giỏ hàng.....	64
Sơ đồ 38 Sơ đồ hợp tác chức năng chỉnh sửa giỏ hàng.....	65
Sơ đồ 39 Sơ đồ tác nhân - usecase xác nhận đơn hàng	65

Sơ đồ 40	Sơ đồ tuần tự chức năng xác nhận đơn hàng.....	66
Sơ đồ 41	Sơ đồ hợp tác chức năng xác nhận giỏ hàng	66
Sơ đồ 42	Sơ đồ tác nhân - usecase chức năng thanh toán tiền mặt.....	67
Sơ đồ 43	Sơ đồ tuần tự chức năng thanh toán tiền mặt	68
Sơ đồ 44	Sơ đồ hợp tác chức năng thanh toán tiền mặt	68
Sơ đồ 45	Sơ đồ tác nhân - usecase chức năng chuyển bản	69
Sơ đồ 46	Sơ đồ tuần tự chức năng chuyển bản.....	70
Sơ đồ 47	Sơ đồ hợp tác chức năng chuyển bản	70
Sơ đồ 48	Sơ đồ tác nhân - usecase chức năng nhắn tin.....	71
Sơ đồ 49	Sơ đồ tuần tự chức năng nhắn tin	72
Sơ đồ 50	Sơ đồ hợp tác chức năng nhắn tin.....	72
Sơ đồ 51	Sơ đồ tác nhân - usecase chức năng xem báo cáo cá nhân.....	73
Sơ đồ 52	Sơ đồ tuần tự chức năng xem báo cáo cá nhân	74
Sơ đồ 53	Sơ đồ hợp tác chức năng xem báo cáo cá nhân	74
Sơ đồ 54	Sơ đồ tác nhân - usecase chức năng hủy món	75
Sơ đồ 55	Sơ đồ tuần tự chức năng hủy món	76
Sơ đồ 56	Sơ đồ hợp tác chức năng hủy món	76
Sơ đồ 57	Sơ đồ tác nhân - usecase chức năng hủy đơn hàng	77
Sơ đồ 58	Sơ đồ tuần tự chức năng hủy đơn hàng.....	78
Sơ đồ 59	Sơ đồ hợp tác chức năng hủy đơn hàng	78
Sơ đồ 60	Sơ đồ tác nhân - usecase chức năng đặt lại bàn ăn.....	79
Sơ đồ 61	Sơ đồ tuần tự chức năng đặt lại bàn ăn	80
Sơ đồ 62	Sơ đồ hợp tác chức năng đặt lại bàn ăn	80
Sơ đồ 63	Sơ đồ tác nhân - usecase chức năng xem báo cáo	81
Sơ đồ 64	Sơ đồ tuần tự chức năng báo cáo	82
Sơ đồ 65	Sơ đồ hợp tác chức năng báo cáo	82
Sơ đồ 66	Sơ đồ quy trình nghiệp vụ	83
Sơ đồ 67	Sơ đồ cơ sở dữ liệu tổng quát.....	84

DANH MỤC HÌNH

Hình 1 Biểu tượng Dart	27
Hình 2 Biểu tượng Flutter	28
Hình 3 Biểu tượng Python.....	29
Hình 4 Đặc điểm của ngôn ngữ python.....	30
Hình 5 Biểu tượng Django	32
Hình 6 Biểu tượng SQLite	33
Hình 7 Biểu tượng Firebase	34
Hình 8 Giao diện đăng nhập website	89
Hình 9 Giao diện đăng nhập website thiếu email.....	89
Hình 10 Giao diện đăng nhập website thiếu mật khẩu	90
Hình 11 Giao diện đăng nhập website sai email	90
Hình 12 Giao diện trang chủ website.....	91
Hình 13 Giao diện trang chủ website theo khối	91
Hình 14 Giao diện danh mục người dùng	92
Hình 15 Giao diện danh mục người dùng theo khối.....	93
Hình 16 Giao diện ẩn danh sách danh mục	94
Hình 17 Giao diện hiện danh mục	94
Hình 18 Giao diện danh sách bản ghi	95
Hình 19 Giao diện tìm kiếm.....	95
Hình 20 Giao diện xóa đa bản ghi	96
Hình 21 Giao diện xác nhận xóa đa bản ghi	96
Hình 22 Giao diện xóa đa bản ghi thành công.....	97
Hình 23 Giao diện xuất báo cáo	97
Hình 24 Ví dụ một file báo cáo	97
Hình 25 Giao diện bộ lọc	98
Hình 26 Thêm mới người dùng	98
Hình 27 Giao diện thêm người dùng lỗi.....	99
Hình 28 Giao diện mật khẩu không trùng khớp	100
Hình 29 Giao diện thêm người dùng trùng email.....	100
Hình 30 Giao diện thêm người dùng thành công	101
Hình 31 Giao diện chỉnh sửa người dùng	102
Hình 32 Giao diện "this form"	102
Hình 33 Giao diện thay đổi mật khẩu	103
Hình 34 Giao diện thay đổi mật khẩu lỗi	103
Hình 35 Giao diện thay đổi mật khẩu thành công.....	103
Hình 36 Giao diện xóa người dùng.....	104
Hình 37 Giao diện xác nhận xóa người dùng	104
Hình 38 Giao diện xóa người dùng thành công.....	105
Hình 39 Giao diện danh mục bản ăn.....	105

Hình 40	Giao diện thêm mới bàn ăn.....	106
Hình 41	Giao diện thêm mới bàn ăn lỗi.....	107
Hình 42	Giao diện thêm mới bàn ăn thành công.....	107
Hình 43	Giao diện chỉnh sửa bàn ăn.....	108
Hình 44	Giao diện chỉnh sửa bàn ăn thành công.....	108
Hình 45	Giao diện xóa bàn ăn.....	109
Hình 46	Giao diện xác nhận xóa bàn ăn.....	109
Hình 47	Giao diện xóa bàn ăn thành công.....	110
Hình 48	Giao diện lời nhắn.....	111
Hình 49	Giao diện thêm mới lời nhắn.....	111
Hình 50	Giao diện thêm lời nhắn lỗi.....	112
Hình 51	Giao diện thêm mới lời nhắn thành công.....	112
Hình 52	Giao diện chỉnh sửa lời nhắn.....	113
Hình 53	Giao diện chỉnh sửa lời nhắn thành công.....	113
Hình 54	Giao diện xóa lời nhắn.....	114
Hình 55	Giao diện xác nhận xóa lời nhắn.....	114
Hình 56	Giao diện xóa lời nhắn thành công.....	114
Hình 57	Giao diện chương trình khuyến mãi.....	115
Hình 58	Giao diện thêm mới chương trình khuyến mãi.....	116
Hình 59	Giao diện thêm mới chương trình khuyến mãi lỗi.....	116
Hình 60	Giao diện thêm mới chương trình khuyến mãi thành công.....	117
Hình 61	Giao diện chỉnh sửa chương trình khuyến mãi.....	117
Hình 62	Giao diện chỉnh sửa chương trình khuyến mãi thành công.....	118
Hình 63	Giao diện xóa chương trình khuyến mãi.....	118
Hình 64	Giao diện xác nhận xóa chương trình khuyến mãi.....	119
Hình 65	Giao diện xóa chương trình khuyến mãi thành công.....	119
Hình 66	Giao diện loại sản phẩm.....	120
Hình 67	Giao diện thêm mới loại sản phẩm.....	120
Hình 68	Giao diện thêm mới loại sản phẩm lỗi.....	121
Hình 69	Giao diện thêm mới loại sản phẩm thành công.....	121
Hình 70	Giao diện chỉnh sửa loại sản phẩm.....	122
Hình 71	Giao diện chỉnh sửa loại sản phẩm thành công.....	122
Hình 72	Giao diện xóa loại sản phẩm.....	123
Hình 73	Giao diện xác nhận xóa loại sản phẩm.....	123
Hình 74	Giao diện xóa loại sản phẩm thành công.....	123
Hình 75	Giao diện nhãn sản phẩm.....	124
Hình 76	Giao diện thêm mới nhãn sản phẩm.....	124
Hình 77	Giao diện thêm mới nhãn sản phẩm lỗi.....	125
Hình 78	Giao diện thêm mới nhãn sản phẩm thành công.....	125
Hình 79	Giao diện chỉnh sửa nhãn sản phẩm.....	126
Hình 80	Giao diện chỉnh sửa nhãn sản phẩm thành công.....	126

Hình 81 Giao diện xóa nhãn sản phẩm.....	127
Hình 82 Giao diện xác nhận xóa nhãn sản phẩm.....	127
Hình 83 Giao diện xóa nhãn sản phẩm thành công.....	127
Hình 84 Giao diện biến thể sản phẩm.....	128
Hình 85 Giao diện thêm mới biến thể sản phẩm.....	129
Hình 86 Giao diện thêm mới biến thể sản phẩm lỗi.....	130
Hình 87 Giao diện thêm mới biến thể sản phẩm lỗi 2.....	131
Hình 88 Giao diện thêm mới biến thể sản phẩm thành công.....	131
Hình 89 Giao diện chỉnh sửa biến thể sản phẩm.....	132
Hình 90 Giao diện thêm mới biến thể sản phẩm thành công.....	133
Hình 91 Giao diện xóa biến thể sản phẩm.....	134
Hình 92 Giao diện xác nhận xóa biến thể sản phẩm.....	134
Hình 93 Giao diện xóa biến thể sản phẩm thành công.....	135
Hình 94 Giao diện sản phẩm.....	135
Hình 95 Giao diện thêm mới sản phẩm.....	136
Hình 96 Giao diện lựa chọn loại sản phẩm.....	137
Hình 97 Giao diện thêm/chỉnh sửa loại sản phẩm trực tiếp.....	137
Hình 98 Giao diện đa sự lựa chọn.....	138
Hình 99 Giao diện sản phẩm biến thể.....	138
Hình 100 Giao diện thêm mới sản phẩm lỗi.....	139
Hình 101 Giao diện thêm sản phẩm thành công.....	140
Hình 102 Giao diện chỉnh sửa sản phẩm.....	141
Hình 103 Giao diện chỉnh sửa sản phẩm thành công.....	142
Hình 104 Giao diện xóa sản phẩm.....	143
Hình 105 Giao diện xác nhận xóa sản phẩm.....	144
Hình 106 Giao diện xóa sản phẩm thành công.....	144
Hình 107 Giao diện sản phẩm gọi thêm.....	144
Hình 108 Giao diện thêm mới sản phẩm gọi thêm.....	145
Hình 109 Giao diện thêm mới sản phẩm gọi thêm lỗi.....	145
Hình 110 Giao diện thêm mới sản phẩm gọi thêm thành công.....	146
Hình 111 Giao diện chỉnh sửa sản phẩm gọi thêm.....	146
Hình 112 Giao diện chỉnh sửa sản phẩm gọi thêm thành công.....	146
Hình 113 Giao diện xóa sản phẩm gọi thêm.....	147
Hình 114 Giao diện xác nhận xóa sản phẩm gọi thêm.....	147
Hình 115 Giao diện xóa sản phẩm gọi thêm thành công.....	148
Hình 116 Giao diện đơn hàng.....	148
Hình 117 Giao diện thêm mới đơn hàng.....	149
Hình 118 Giao diện thêm mới đơn hàng lỗi.....	149
Hình 119 Giao diện thêm mới đơn hàng thành công.....	150
Hình 120 Giao diện chỉnh sửa đơn hàng.....	150
Hình 121 Giao diện chỉnh sửa đơn hàng thành công.....	150

Hình 122	Giao diện xóa đơn hàng	151
Hình 123	Giao diện xác nhận xóa đơn hàng.....	151
Hình 124	Giao diện xóa đơn hàng thành công	151
Hình 125	Giao diện header website	152
Hình 126	Giao diện đăng xuất	152
Hình 127	Giao diện nút "HISTORY"	152
Hình 128	Giao diện lịch sử trống	152
Hình 129	Giao diện lịch sử	152
Hình 130	Biểu tượng ứng dụng di động.....	153
Hình 131	a,b Giao diện đăng nhập.....	154
Hình 132	a,b,c Giao diện đăng nhập.....	155
Hình 133	a,b,c Giao diện xử lý đăng nhập.....	156
Hình 134	a,b,c Giao diện trang chủ	157
Hình 135	a,b Giao diện bàn ăn	158
Hình 136	Giao diện đơn hàng trống.....	159
Hình 137	a,b Giao diện menu	160
Hình 138	Giao diện món ăn	160
Hình 139	Giao diện "View all".....	161
Hình 140	a,b Giao diện menu theo loại	161
Hình 141	a,b,c Giao diện chi tiết món ăn.....	162
Hình 142	Giao diện thêm vào giỏ hàng thất bại	163
Hình 143	a,b,c Giao diện thêm vào giỏ hàng.....	164
Hình 144	a,b,c Giao diện giỏ hàng.....	165
Hình 145	a,b Giao diện giỏ hàng.....	166
Hình 146	a,b,c Giao diện chi tiết đơn hàng.....	167
Hình 147	a,b Giao diện hủy món ăn	168
Hình 148	a,b,c Giao diện thanh toán.....	169
Hình 149	a,b Giao diện các thao tác bàn ăn.....	170
Hình 150	a,b,c Giao diện chuyển bàn ăn.....	171
Hình 151	Giao diện chuyển bàn ăn thành công.....	172
Hình 152	Giao diện nhắn tin.....	173
Hình 153	Giao diện chi tiết tin nhắn.....	174
Hình 154	a,b Giao diện Profile.....	175
Hình 155	Giao diện báo cáo cá nhân	176
Hình 156	Giao diện mốc thời gian	176
Hình 157	a,b Giao diện báo cáo	177
Hình 159	Giao diện biểu đồ	177
Hình 187	Cấu trúc thư mục website quản trị c	180
Hình 188	Cấu trúc thư mục website bản trị b	180
Hình 189	Cấu trúc thư mục website quản trị a	180
Hình 190	Cấu trúc thư mục website quản trị d.....	181

Hình 191 Cấu trúc thư mục website quản trị e	181
Hình 165 Cấu trúc thư mục website quản trị f.....	181
Hình 193 Cấu trúc thư mục ứng dụng di động c	182
Hình 194 Cấu trúc thư mục ứng dụng di động b.....	182
Hình 195 Cấu trúc thư mục ứng dụng di động a	182

CHƯƠNG I: TỔNG QUAN

1. Lý do chọn đề tài

Ngành ăn uống, dịch vụ ăn uống ở nước ta đang phát triển rất nhanh và đa dạng. Cung cấp lương thực thực phẩm cho xã hội đồng thời góp phần quan trọng vào phát triển kinh tế đất nước, nhờ đó mà việc mở rộng thị trường và phát triển chất lượng ngành là việc cấp thiết và tất yếu nhất. Thực tại việc quản lý món ăn, đơn hàng và đặt món tại các nhà hàng đang gặp nhiều khó khăn và hạn chế bởi phong cách quản lý và phục vụ truyền thống dẫn đến chất lượng phục vụ chưa được như mong muốn.

Thực tế cho thấy, những nhà hàng phát triển dịch vụ ăn uống có quy trình phục vụ chuyên nghiệp, chính xác, nhanh chóng và uy tín đều nhận được đánh giá tích cực từ phía khách hàng. Nhờ đó mà nhà hàng luôn được nhiều thực khách ủng hộ và đồng thời uy tín và chất lượng phục vụ luôn được đánh giá cao. Vì lẽ đó các nhà hàng hiện tại đã và đang khai thác giá trị của công nghệ thông tin áp dụng vào ngành dịch vụ của mình.

Bản chất của cuộc Cách mạng công nghiệp 4.0 đối với dịch vụ ăn uống là dựa trên nền tảng công nghệ số và tích hợp tất cả các công nghệ thông minh để tối ưu hóa quy trình, chất lượng phục vụ thực khách. Quản lý món ăn, đơn hàng, tối ưu quy trình đặt món và thống kê doanh thu đối với những nhà hàng lớn, phục vụ lượng thực khách đông đảo là một vấn đề vô cùng nan giải. Chính vì vậy áp dụng một hệ thống giúp nhân viên đặt món ăn cho khách hàng nhanh chóng, chính xác và linh hoạt là vô cùng cần thiết.

2. Mục tiêu và chức năng của hệ thống

2.1. Mục tiêu

Quản lý món ăn, quá trình đặt món và thống kê doanh thu, các mục tiêu bao gồm:

- Tính di động và linh hoạt.
- Số hóa việc lưu trữ thông tin món ăn, nhân viên, đơn hàng và các thông tin chương trình khuyến mãi.
- Tìm kiếm, trích xuất dữ liệu nhanh chóng và chính xác.
- Giảm thiểu việc lưu trữ giấy tờ.
- Theo dõi tình hình kinh doanh.
- Theo dõi tình hình nhân viên phục vụ.
- Hỗ trợ gửi và nhận thông báo thời gian thực về tình hình phục vụ.
- Hỗ trợ trao đổi thông tin giữa các nhân viên.

2.2. Chức năng

- Quản trị website.
- Quản lý bàn ăn.

- Quản lý thông tin khuyến mãi.
- Quản lý món ăn.
- Quản lý đơn hàng.
- Quản lý nhân viên.
- Quản lý doanh thu.
- Lưu trữ, trích xuất dữ liệu.
- Quản lý lời nhắn (gửi lời nhắn đến toàn thể nhân viên).
- Gửi tin nhắn (gửi tin nhắn riêng).
- Gửi và nhận thông báo đẩy.

3. Mô tả bài toán

Yêu cầu hệ thống đặt ra tập trung vào tính linh động, chính xác và nhanh chóng. Vì vậy hệ thống được phát triển trên hai nền tảng website quản trị và ứng dụng di động.

Mục đích của website quản trị là quản lý các thông tin như sau:

- Quản lý bàn ăn (hệ thống giữ lại tính hợp lý trong phong cách phục vụ truyền thống là phân biệt món ăn phục vụ khách hàng bằng thông tin bàn ăn của đơn hàng) :
 - Quản lý các thông tin bàn ăn: Mã, tên, trạng thái, ...
- Quản lý món ăn :
 - Quản lý loại món ăn (đối với cửa hàng phục vụ nhiều loại món ăn thì quản lý thông tin này để phân biệt món ăn là cần thiết).
 - Quản lý nhãn món ăn (mỗi món ăn được gắn nhãn để phân biệt tính chất của món ăn như : món cay, món chay,...).
 - Quản lý các thông tin món ăn: Mã, tên, hình ảnh, mô tả,...
 - Quản lý biến thể món ăn: Kích cỡ và các biến thể tùy loại món ăn (VD: Pizza có biến thể đế bánh : Dày, vừa, nhỏ)
 - Quản lý gọi thêm (VD: gọi thêm cơm trong phần cơm, gọi thêm hải sản trong một phần pizza thịt)
- Quản lý nhân viên (quản lý thông tin tài khoản đăng nhập vào ứng dụng di động):
 - Quản lý thông tin nhân viên: Mã, tên, hình ảnh, số điện thoại, email (Dùng để đăng nhập vào ứng dụng) và mật khẩu (Đăng nhập vào ứng dụng)
- Quản lý hóa đơn.
- Quản lý lời nhắn (lời nhắn cần gửi đến toàn thể nhân viên).
- Quản lý thông tin khuyến mãi (quản lý các chương trình khuyến mãi và hiển thị đến toàn thể nhân viên để tiện lợi cho việc nắm bắt thông tin).

Mục đích của ứng dụng di động là hiển thị các thông tin như lời nhắn, chương trình khuyến mãi, bàn ăn, món ăn, đơn hàng, các thông tin người dùng và xử lý các tác vụ như sau:

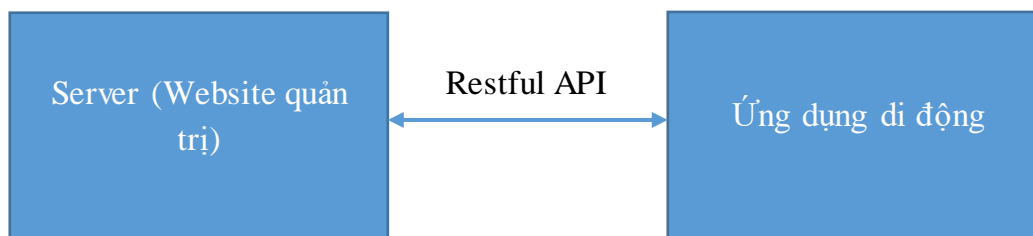
- Đặt món ăn.
- Chỉnh sửa đơn hàng (Thêm, xóa, hủy đơn hàng).
- Thanh toán đơn hàng.
- Nhắn tin riêng với nhân viên khác.

CHƯƠNG II: CƠ SỞ LÝ THUYẾT

1. Cấu trúc hệ thống

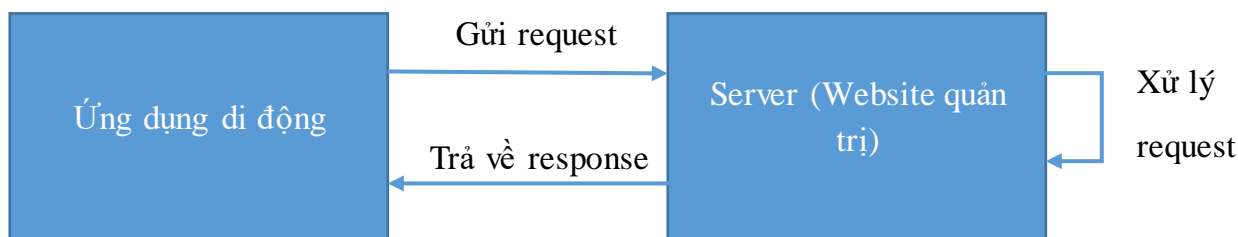
1.1. Mô hình kết nối

Hệ thống đặt món ăn cho nhà hàng được phát triển trên hai nền tảng: Ứng dụng di động và website quản trị. Ứng dụng di động giao tiếp với server (website quản trị) thông qua giao thức Restful API. Mô hình kết nối được trình bày trong sơ đồ 1:



Sơ đồ 1 Mô hình kết nối website và ứng dụng di động

Server đóng vai trò lưu trữ và xử lý dữ liệu khi có yêu cầu (Request) từ phía ứng dụng di động gửi lên server và trả về dữ liệu (Response) sau khi hoàn thành xử lý dữ liệu. Quy trình xử lý được biểu diễn trong sơ đồ 2:



Sơ đồ 2 Quy trình xử lý phương thức Restful API

1.2. Cấu trúc dữ liệu

Server và ứng dụng giao tiếp với nhau thông qua giao thức Restful API, quy trình gửi yêu cầu (request) và nhận phản hồi (response) là phương thức hoạt động duy nhất của hệ thống, trong đó request và response được xây dựng dựa trên cấu trúc dữ liệu Json được thể hiện như ví dụ sau:

```
{  
  "id": 1,  
  "name": "Pizza Rau Củ (Xốt Bơ Tỏi)",  
  "description": "Thanh nhẹ với ô liu đen tuyệt hảo, cà chua bi tươi ngon, nấm, thơm, bắp, bí ngòi và phô mai Mozzarella cho bạn bữa tiệc rau củ tròn vị",  
  "category": {  
    "id": 1,  
    "name": "Pizza",  
    "description": "Pizza",  
  }  
}
```

```
    "created_at": "2021-09-17T16:42:05.423417",  
    "updated_at": "2021-09-17T16:42:05.423417"  
  },  
}
```

Cấu trúc dữ liệu Json hỗ trợ hầu hết các kiểu dữ liệu như số (Int), chuỗi (String), ngày giờ (Datetime), ngày (Date), đối tượng (Object),...

1.3. Các tiêu chuẩn

- Tiêu chuẩn thời gian:

Định dạng : `yyy-MM-ddTHH:mm:ss.zzzzzz`

Ví dụ: 2021-09-17T16:42:05.423417

- Tiêu chuẩn dữ liệu:

Hệ thống hỗ trợ dữ liệu chuẩn Unicode (UTF-8).

- Tiêu chuẩn bảo mật kết nối:

- API kết nối được mã hóa sử dụng giao thức https với xác thực bằng Bearer Token.
- Để đảm bảo bảo mật, mặc định hệ thống sẽ không cho qua API khi chưa đăng nhập. Để kết nối người dùng nhập tài khoản và mật khẩu gửi lên hệ thống xác thực thông tin và trả về kết quả token nếu xác thực đúng.
- Khi nhận được token, có thể truy cập mọi API khi gửi request theo phương thức như sau:

```
header: {"Content-Type": "application/json", "Authorization": "Bearer [token]"}
```

1.4. Phương thức API

Cấu trúc để gửi request lên server thông qua API bao gồm: Url (đường dẫn API), header (các yêu cầu của API như token), body (dữ liệu xử lý - Json).

Các phương thức để gửi request qua API bao gồm:

- **Get:** Phương thức này yêu cầu url được chỉ định từ phương thức get và trả về một phản hồi là Future <response>. Ở đây, phản hồi là một lớp, chứa thông tin phản hồi.

Ví dụ sử dụng phương thức get:

```
http.get(url, headers: {  
    'Content-Type': 'application/json',  
    'Accept': 'application/json',  
    'Authorization': 'Bearer $token',  
});
```

- **Post:** Phương pháp này được sử dụng để gửi dữ liệu đến các tài nguyên được chỉ định. Nó yêu cầu url được chỉ định bằng cách đăng dữ liệu đã cho và trả về phản hồi dưới dạng Future <response>.

Ví dụ sử dụng phương thức post:

```
http.post(url, headers: {  
    'Content-Type': 'application/json',  
    'Accept': 'application/json',  
    'Authorization': 'Bearer $token',  
}, body);
```

- **Put:** Phương pháp này được sử dụng cho khả năng cập nhật. Nó cập nhật tất cả các biểu diễn hiện tại của tài nguyên đích với các payloads. Phương thức này yêu cầu url được chỉ định và trả về phản hồi là Future <response >

Ví dụ sử dụng phương thức put:

```
http.put(url, headers: {  
    'Content-Type': 'application/json',  
    'Accept': 'application/json',  
    'Authorization': 'Bearer $token',  
}, body);
```

- **Delete:** Phương pháp này được sử dụng để loại bỏ tất cả các tài nguyên được chỉ định.

Ví dụ sử dụng phương thức delete:


```
http.delete(url, headers: {  
    'Content-Type': 'application/json',  
    'Accept': 'application/json',  
    'Authorization': 'Bearer $token',  
});
```

Sau khi server xử lý dữ liệu sẽ trả về kết quả response và mã trạng thái trong bảng 1.

Bảng 1 Danh sách mã response API

Mã lỗi	Trạng thái	Mô tả mã lỗi
200	OK	Thành công
201	Created	Sử dụng phương thức POST tạo dữ liệu thành công
400	Bad Request	Dữ liệu được gửi đi chưa hoàn thiện hoặc thiếu
403	Forbidden	Hành động hoặc yêu cầu gửi đi không được hệ thống cấp phép
404	Not Found	Không tìm thấy dữ liệu
422	Unprocessable Entity	Có lỗi xảy ra không quá trình gửi yêu cầu
429	Too Many Requests	Ứng dụng nhận quá nhiều yêu cầu và có giới hạn
500	Internal Server Error	Gửi yêu cầu quá giới hạn thời gian

Ví dụ hệ thống trả về kết quả thất bại:

```
HTTP 400 Bad Request  
Allow: GET, POST, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept  
{  
  "name": [  
    "This field may not be blank."  
  ],  
  "description": [  
    "This field may not be blank."  
  ]  
}
```

Ví dụ hệ thống trả về kết quả thành công:

```
HTTP 200 OK
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "id": 1,
  "name": "Pizza Rau Củ (Xốt Bơ Tỏi)",
  "description": "Thanh nhẹ với ô liu đen tuyệt hảo, cà chua bi tươi ngon, nấm, thơm, bắp, bí ngòi và phô mai Mozzarella cho bạn bữa tiệc rau củ tròn vị",
  "category": {
    "id": 1,
    "name": "Pizza",
    "description": "Pizza",
    "created_at": "2021-09-17T16:42:05.423417",
    "updated_at": "2021-09-17T16:42:05.423417"
  },
}
```

2. Công nghệ sử dụng

Ứng dụng di động phát triển dành cho nhân viên nhà hàng hỗ trợ hiển thị bàn và món ăn, đặt món, hủy món, hủy bàn, chuyển bàn, thanh toán và các chức năng khác như: *Chat, báo cáo*, ... Ứng dụng được phát triển dựa trên Framework **Flutter** với ngôn ngữ lập trình **Dart**.

Ứng dụng di động hỗ trợ chức năng Chat và gửi thông báo đẩy (Push Notification) được cung cấp bởi Firebase Firestore và Firebase Message Cloud.

Website quản trị phát triển dành cho người quản lý nhà hàng hỗ trợ các chức năng như: Thêm, sửa, xóa các thông tin như: *Sản phẩm, tài khoản nhân viên, bàn ăn*,... Website quản trị được phát triển dựa trên Framework **Django** với ngôn ngữ lập trình **Python**.

2.1. Flutter và Dart

2.1.1. Ngôn ngữ lập trình Dart [1]



Hình 1 Biểu tượng Dart

2.1.1.1. Tổng quan

Dart là ngôn ngữ lập trình đa mục đích ban đầu được phát triển bởi Google và sau đó được Ecma (ECMA-408) phê chuẩn làm tiêu chuẩn. Nó được sử dụng để xây dựng các ứng dụng web, server, máy tính để bàn và thiết bị di động. Dart là một ngôn ngữ hướng đối tượng, được xác định theo lớp, với cơ chế thu gom rác (garbage-collected), sử dụng cú pháp kiểu C để dịch mã tùy ý sang JavaScript. Nó hỗ trợ giao diện (interface), tái sử dụng (mixin), trừu tượng (abstract), tham số hóa kiểu dữ liệu (generic), static typing và sound type (2 cái cuối có thể hiểu là type-safe). Dart là ngôn ngữ mã nguồn mở và miễn phí, được phát triển trên *Github*. Hiện nay Dart đã ra mắt phiên bản 2.14.1.

2.1.1.2. Tại sao lại chọn Dart?

Các nhà phát triển tại Google và các nơi khác sử dụng Dart để tạo các ứng dụng chất lượng cao, quan trọng cho iOS, Android và web. Với các tính năng nhắm đến sự phát triển phía khách hàng, Dart rất phù hợp cho cả ứng dụng di động và web. Dart giúp tạo ra những giao diện đẹp và chất lượng cao trên tất cả các màn hình nhờ các ưu điểm của ngôn ngữ Dart như:

- Một ngôn ngữ được tối ưu hóa cho client.
- Framework mạnh mẽ.
- Công cụ linh hoạt.

2.1.1.3. Những ưu điểm của Dart

- **Năng suất:** Cú pháp Dart rõ ràng và súc tích, công cụ của nó đơn giản nhưng mạnh mẽ. Type-safe giúp bạn xác định sớm các lỗi tĩnh. Dart có các thư viện cốt lõi và một hệ sinh thái gồm hàng ngàn package.

- **Nhanh:** Dart cung cấp tối ưu hóa việc biên dịch trước thời hạn để có được dự đoán hiệu suất cao và khởi động nhanh trên các thiết bị di động và web.
- **Di động:** Dart biên dịch thành mã ARM và x86, để các ứng dụng di động của Dart có thể chạy tự nhiên trên iOS, Android và hơn thế nữa. Đối với các ứng dụng web, chuyển mã từ Dart sang JavaScript.
- **Dễ gần:** Dart quen thuộc với nhiều nhà phát triển hiện có, nhờ vào cú pháp và định hướng đối tượng không gây ngạc nhiên của nó.
- **Reactive:** Dart rất phù hợp với lập trình Reactive, với sự hỗ trợ để quản lý các đối tượng tồn tại trong thời gian ngắn, chẳng hạn như các widget UI, thông qua phân bổ đối tượng nhanh. Dart hỗ trợ lập trình không đồng bộ thông qua các tính năng ngôn ngữ và API sử dụng các đối tượng Future và Stream.

2.1.2. Flutter framework [2]



Hình 2 Biểu tượng Flutter

Flutter là nền tảng phát triển ứng dụng đa nền tảng cho iOS và Android do Google phát triển được sử dụng để tạo ra các ứng dụng gốc (native app) cho Google.

Flutter gồm 2 thành phần quan trọng:

SDK (Software Development Kit): Một bộ sưu tập các công cụ sẽ giúp bạn phát triển các ứng dụng của mình. Điều này bao gồm các công cụ để biên dịch mã của bạn thành mã máy gốc (mã cho iOS và Android).

Framework (UI Library based on widgets): Một tập hợp các thành phần giao diện người dùng (UI) có thể tái sử dụng (button, text inputs, slider, v.v.) giúp bạn có thể cá nhân hóa tùy theo nhu cầu của riêng mình.

2.1.2.1. Tại sao nên sử dụng Flutter?

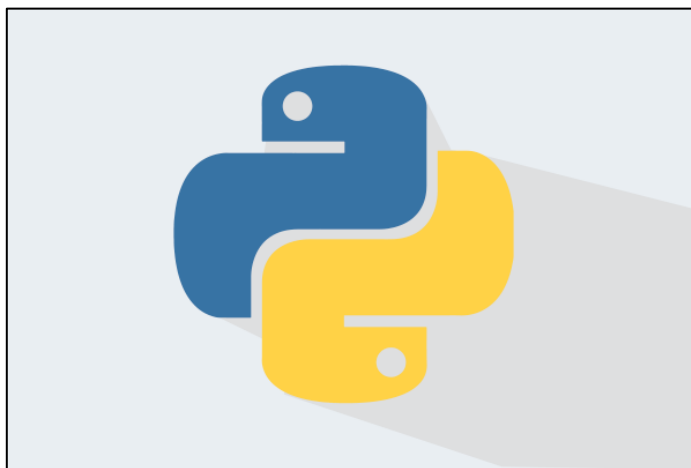
Phát triển ứng dụng nhanh chóng: Tính năng hot reload giúp người sử dụng nhanh chóng và dễ dàng thử nghiệm, xây dựng giao diện người dùng, thêm tính năng và sửa lỗi nhanh hơn. Trải nghiệm tải lại lần thứ hai, mà không làm mất trạng thái, trên emulator, simulator và device cho iOS và Android.

Giao diện người dùng đẹp và thu hút: Thỏa mãn người dùng với các widget built-in đẹp mắt của Flutter theo Material Design và Cupertino (iOS-flavor), các giao diện lập trình ứng dụng (API) chuyển động phong phú, scroll tự nhiên mượt mà và tự nhận thức được nền tảng.

Framework hiện đại: Dễ dàng tạo giao diện người dùng của bạn với framework hiện đại của Flutter và tập hợp các platform, layout và widget phong phú. Giải quyết các thách thức giao diện người dùng khó khăn của bạn với các API mạnh mẽ và linh hoạt cho 2D, animation, gesture, hiệu ứng và hơn thế nữa.

2.2. Python và Django

2.2.1. Ngôn ngữ lập trình Python [3]



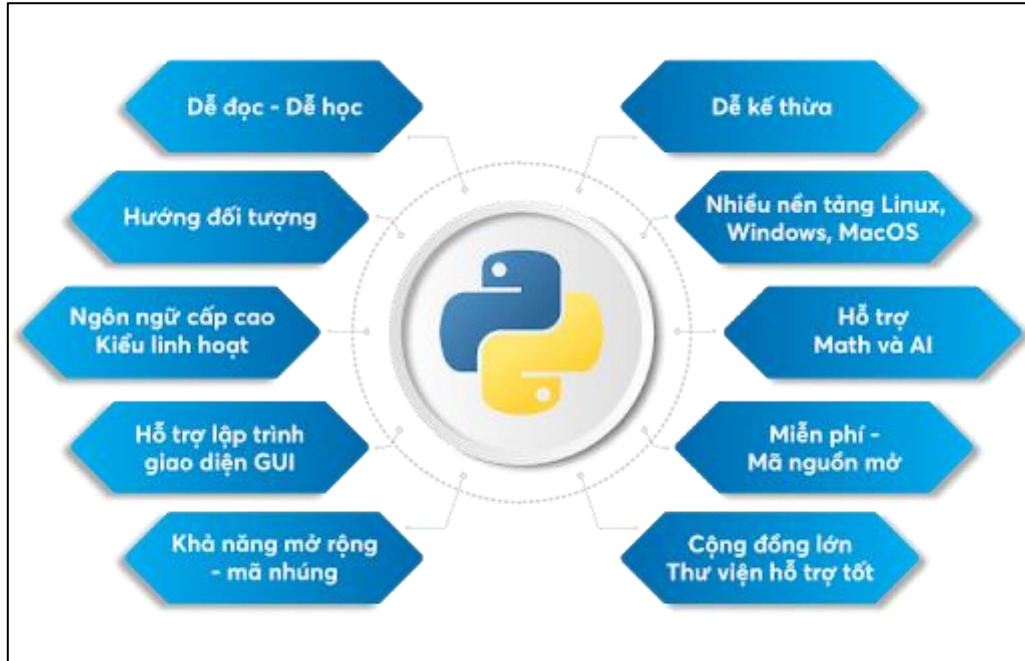
Hình 3 Biểu tượng Python

2.2.1.1. Tổng quan

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, Van Rossum đã từ chức Leader trong cộng đồng ngôn ngữ Python sau 30 năm lãnh đạo.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

2.2.1.2. Đặc điểm của ngôn ngữ python



Hình 4 Đặc điểm của ngôn ngữ python

➤ **Ngôn ngữ lập trình đơn giản, dễ học – dễ học**

Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.

➤ **Miễn phí, mã nguồn mở**

Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng cho mục đích thương mại. Vì là mã nguồn mở, người dùng không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.

➤ **Khả năng di động linh hoạt**

Giả sử bạn giả sử người dùng đã viết mã Python cho máy Windows của mình. Bây giờ, nếu người dùng muốn chạy nó trên máy Mac, người dùng không cần phải thay đổi nó như cũ. Nói cách khác, có thể lấy một mã và chạy nó trên bất kỳ máy nào, không cần phải viết mã khác nhau cho các máy khác nhau. Điều này làm cho Python trở thành một ngôn ngữ di động. Tuy nhiên, phải tránh mọi tính năng phụ thuộc hệ thống trong trường hợp này. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.

➤ **Khả năng mở rộng và có thể nhúng**

Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, người dùng có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của người dùng những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.

➤ **Ngôn ngữ thông dịch cấp cao**

Không giống như C/C++, với Python, người dùng không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Trong nội bộ, mã nguồn của nó được chuyển đổi thành một hình thức ngay lập tức được gọi là bytecode. Vì vậy, tất cả những gì cần làm là chạy đoạn code Python mà không phải lo lắng về việc liên kết với các thư viện và những thứ khác.

➤ **Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến**

Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code. Ví dụ: Bạn cần kết nối cơ sở dữ liệu MySQL trên Web server? Lập trình viên có thể nhập thư viện MySQLdb và sử dụng nó. Có các thư viện cho các biểu thức thông thường, tạo tài liệu, kiểm tra đơn vị, trình duyệt web, phân luồng, cơ sở dữ liệu, CGI, email, thao tác hình ảnh và rất nhiều chức năng khác. Vì vậy, lập trình viên có thể chắc chắn rằng nó sẽ không làm hỏng code hay ứng dụng của mình.

➤ **Hướng đối tượng**

Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, lập trình viên có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng. Python hỗ trợ cả lập trình hướng đối tượng, một trong những tính năng chính của nó. Nó cũng hỗ trợ nhiều kế thừa, không giống như Java.

2.2.1.3. Ưu điểm khi lập trình web với python

- Python là ngôn ngữ lập trình được đánh giá tương đối dễ học. Code Python ngắn gọn, dễ đọc. Nếu so sánh với các ngôn ngữ lập trình khác, điển hình như Java, code Python ngắn hơn 3-5 lần và C++ là 5-10 lần
- Học Python sẽ là bước đệm để học tiếp các ngôn ngữ lập trình khác.
- Python có tiềm năng trong lĩnh vực trí tuệ nhân tạo. Với sự linh hoạt của ngôn ngữ cùng tốc độ xử lý nhanh nhạy. Python được kỳ vọng sẽ là nền tảng cho ngành công nghiệp mũi nhọn thời đại 4.0.

2.2.2. Django framework [4]



Hình 5 Biểu tượng Django

2.2.2.1. Tổng quan

Django là một framework bậc cao của Python có thể thúc đẩy việc phát triển phần mềm thần tốc và clean, thiết kế thực dụng. Được xây dựng bởi nhiều lập trình viên kinh nghiệm, Django tập trung lớn những vấn đề phát triển Web, có thể phát triển trang web mà không cần xây dựng từ những căn bản. Đặc biệt nó miễn phí và có mã nguồn mở.

2.2.2.2. Những lợi thế của Django

- **Hoàn thiện:** Django phát triển theo tư tưởng "Batteries included" (có thể hiểu ý nghĩa là tích hợp toàn bộ, chỉ cần gọi ra mà dùng). Nó cung cấp mọi thứ cho developer không cần phải nghĩ phải dùng cái ngoài. Chúng ta chỉ cần tập trung vào sản phẩm, tất cả đều hoạt động liền mạch với nhau.
- **Đa năng:** Django có thể được dùng để xây dựng hầu hết các loại website, từ hệ thống quản lý nội dung, cho đến các trang mạng xã hội hay web tin tức. Nó có thể làm việc với framework client-side, và chuyển nội dung hầu hết các loại format (HTML, RESS, JSON, XML, ...)
- **Bảo mật:** Django giúp các developer trang các lỗi bảo mật thông thường bằng cách cung cấp framework rằng có những kỹ thuật "phải làm như vậy" để bảo vệ website. Ví dụ: Django cung cấp bảo mật quản lý tên tài khoản và mật khẩu, tránh các lỗi cơ bản như để thông tin session lên cookie, mã hóa mật khẩu thay vì lưu thẳng.
- **Dễ Scale:** Django sử dụng kiến trúc shared-nothing dựa vào component (mỗi phần của kiến trúc sẽ độc lập với nhau, và có thể thay thế hoặc sửa đổi nếu cần thiết). Có sự chia tách rõ ràng giữa các phần nghĩa là nó có thể scale cho việc gia tăng traffic bằng cách thêm phần cứng ở mỗi cấp độ: caching, servers, database servers, hoặc application servers. Nhiều web về kinh doanh đã thành công khi Django được scale đáp ứng yêu cầu của họ
- **Dễ bảo trì:** code Django được viết theo nguyên tắc thiết kế mẫu (pattern design) có thể khuyến khích ý tưởng bảo trì và tái sử dụng code. Trên thực tế, nó sự theo khái niệm Don't Repeat Yourself làm cho không có sự lặp lại không cần thiết, giảm một lượng code.
- **Tính linh động:** Django được viết bằng Python, nó có thể chạy đa nền tảng. Nó có nghĩa rằng bạn không ràng buộc một platform server cụ thể. Django được hỗ

trợ tốt ở nhiều nhà cung cấp hosting, họ sẽ cung cấp hạ tầng và tài liệu cụ thể cho hosting web Django.

2.2.2.3. Lợi ích của Django REST Framework

Django REST framework là một bộ công cụ mạnh mẽ và linh hoạt hỗ trợ trong xây dựng Web APIs.

Những lý do nên chọn Django Rest framework bao gồm:

- Hỗ trợ phương thức xác minh tài khoản.
- Dễ dàng xây dựng APIs.
- Linh hoạt trong kết nối cơ sở dữ liệu.

2.3. Hệ quản trị cơ sở dữ liệu SQLite [5]



Hình 6 Biểu tượng SQLite

2.3.1. Tổng quan

SQLite là hệ thống cơ sở dữ liệu quan hệ nhỏ gọn, hoàn chỉnh, có thể cài đặt bên trong các trình ứng dụng khác. SQLite được Richard Hipp viết dưới dạng thư viện bằng ngôn ngữ lập trình C.

2.3.2. Tại sao lại chọn SQLite?

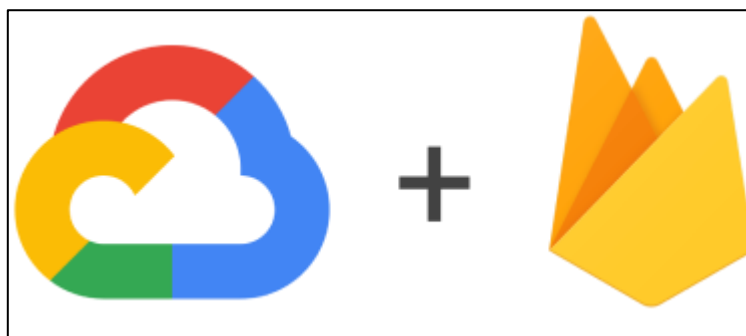
Những lí do SQLite là một hệ quản trị cơ sở dữ liệu được sử dụng rộng rãi:

- SQLite không yêu cầu một quy trình hoặc hệ thống máy chủ riêng biệt để hoạt động.
- SQLite không cần cấu hình, có nghĩa là không cần thiết lập hoặc quản trị.
- Một cơ sở dữ liệu SQLite hoàn chỉnh được lưu trữ trong một file disk đa nền tảng (cross-platform disk file).
- SQLite rất nhỏ và trọng lượng nhẹ, dưới 400KiB được cấu hình đầy đủ hoặc dưới 250KiB với các tính năng tùy chọn bị bỏ qua.
- SQLite là khép kín (self-contained), có nghĩa là không có phụ thuộc bên ngoài.
- Các transaction trong SQLite hoàn toàn tuân thủ ACID, cho phép truy cập an toàn từ nhiều tiến trình (process) hoặc luồng (thread).

- SQLite hỗ trợ hầu hết các tính năng ngôn ngữ truy vấn (query language) được tìm thấy trong tiêu chuẩn SQL92 (SQL2).
- SQLite được viết bằng ANSI-C và cung cấp API đơn giản và dễ sử dụng.
- SQLite có sẵn trên UNIX (Linux, Mac OS-X, Android, iOS) và Windows (Win32, WinCE, WinRT).

2.4. Firebase

2.4.1. Tổng quan [6]



Hình 7 Biểu tượng Firebase

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

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

2.4.2. Firebase Cloud Firestore [7]

Cloud Firestore là một cơ sở dữ liệu linh hoạt và dễ mở rộng cho mobile, web và server được phát triển từ Firebase and Google Cloud Platform. Cloud Firestore giúp cho việc đồng bộ dữ liệu giữa các ứng dụng phía client một cách nhanh chóng theo thời gian thực (Realtime) và hỗ trợ lưu offline data trong ứng dụng.

Cloud Firestore là một cloud-hosted, NoSQL database mà các ứng dụng phía client có thể trực tiếp truy cập thông qua native SDKs. Nó lưu dữ liệu theo mô hình dữ liệu NoSQL. Dữ liệu được lưu trữ trong các file tài liệu chứa các trường được ánh xạ vào các giá trị. Các file tài liệu này được lưu trữ trong các tập hợp chúng có thể sử dụng nó để tổ chức dữ liệu và truy vấn dữ liệu. Cloud Firestore hỗ trợ rất nhiều kiểu dữ liệu từ đơn giản như String, Integer hay những kiểu dữ liệu phức tạp như các nested object.

2.4.3. Firebase Cloud Messaging [6]

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

CHƯƠNG III: PHÂN TÍCH HỆ THỐNG

1. Phân tích yêu cầu đề bài

Đề tài: “Xây dựng hệ thống đặt món ăn cho nhân viên tại nhà hàng”

Hệ thống hướng đến tính tiện lợi, nhanh chóng và di động, vì vậy hệ thống được xây dựng trên hai nền tảng: website quản trị và ứng dụng di động với những tính năng xử lý những tác vụ nhất định.

Hai nền tảng được phân quyền bởi hai loại người dùng: quản lý và nhân viên.

1.1. Yêu cầu người dùng

Theo khảo sát thực tế tại đa số các nhà hàng lớn và tầm trung, người dùng hệ thống được phân chia theo hai cấp: quản lý và nhân viên.

- Quản lý là người dùng được cấp đầy đủ các quyền quản trị website và sử dụng tất cả chức năng của ứng dụng di động. Điều ấy cho phép người quản lý linh động việc kiểm soát quá trình phục vụ thực khách cũng như quản lý quá trình bán hàng, đồng thời có thể trực tiếp điều chỉnh các thông tin dữ liệu của nhà hàng.
- Nhân viên là người dùng không có quyền truy cập vào website quản trị, nhân viên chỉ có khả năng truy cập vào ứng dụng di động với tài khoản được cấp bởi người quản lý. Việc sử dụng ứng dụng di động của người dùng nhân viên cũng ở mức cơ bản, trong đó nhân viên được cấp quyền truy cập và thao tác các tính năng cơ bản.

1.2. Yêu cầu Website quản trị

1.2.1. Chức năng

Website quản trị hỗ trợ cho người dùng **quản lý**, vì vậy các chức năng chính của Website quản trị:

Bảng 2 Danh sách chức năng Website quản trị

STT	Chức năng	Diễn giải
1	Đăng nhập/Đăng xuất	Cho phép đăng nhập/đăng xuất vào website quản trị.
2	Thay đổi mật khẩu	Cho phép thay đổi mật khẩu truy cập vào website quản trị.
3	Quản lý người dùng	Cho phép thêm/xóa/sửa các thông tin người dùng. (Tài khoản quản lý và nhân viên truy cập vào website quản trị và ứng dụng di động)
4	Quản lý bàn ăn	Cho phép thêm/xóa/sửa các thông tin bàn ăn.

5	Quản lý sản phẩm	Cho phép thêm/xóa/sửa các thông tin liên quan đến sản phẩm (Loại sản phẩm, nhãn sản phẩm, sản phẩm gọi thêm, sản phẩm biến thể, ...)
6	Quản lý đơn hàng	Cho phép thêm/xóa/sửa các thông tin đơn hàng.
7.	Quản lý khuyến mãi	Cho phép thêm/xóa/sửa các thông tin khuyến mãi để hiển thị lên ứng dụng di động.
8.	Quản lý lời nhắn	Cho phép thêm/xóa/sửa các thông tin lời nhắn để hiển thị lên ứng dụng di động.

1.2.2. Yêu cầu đặt ra

Dựa vào các chức năng chính đã nói trên, yêu cầu chức năng được đặt ra như sau:

- Đăng nhập/Đăng xuất: Cho phép
- Quản lý người dùng cho phép quản lý tài khoản quản lý và nhân viên truy cập vào website quản trị và ứng dụng di động:

Bảng 3 Danh sách chức năng quản lý người dùng

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới người dùng.
2.	Chỉnh sửa	Cho phép chỉnh sửa thông tin người dùng.
3	Xóa	Cho phép xóa người dùng.
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách người dùng.
5	Xuất file báo cáo	Cho phép xuất file báo cáo thông tin người dùng.

- Quản lý sản phẩm cho phép quản lý các thông tin và các trường thông tin liên quan đến sản phẩm:
 - Quản lý loại sản phẩm cho phép quản lý các thông tin về loại sản phẩm :

Bảng 4 Danh sách chức năng quản lý loại sản phẩm

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới loại sản phẩm.
2.	Chỉnh sửa	Cho phép chỉnh sửa thông tin loại sản phẩm.
3	Xóa	Cho phép xóa loại sản phẩm.
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách loại sản phẩm.
5	Xuất file báo cáo	Cho phép xuất file báo cáo thông tin loại sản phẩm.

- Quản lý biến thể sản phẩm (sản phẩm có những biến thể như kích cỡ) cho phép quản lý các thông tin về biến thể sản phẩm:

Bảng 5 Danh sách chức năng quản lý biến thể sản phẩm

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới biến thể sản phẩm.
2.	Chỉnh sửa	Cho phép chỉnh sửa thông tin biến thể sản phẩm.
3	Xóa	Cho phép xóa biến thể.
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách biến thể.

5	Xuất file báo cáo	Cho phép xuất file báo cáo thông tin biên thể.
---	-------------------	--

- Quản lý nhãn sản phẩm (sản phẩm được gắn các nhãn như: món cay, món chay để phân biệt) cho phép quản lý các thông tin về nhãn sản phẩm:

Bảng 6 Danh sách chức năng quản lý nhãn sản phẩm

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới nhãn sản phẩm.
2.	Chỉnh sửa	Cho phép chỉnh sửa thông tin nhãn sản phẩm.
3	Xóa	Cho phép xóa nhãn sản phẩm.
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách nhãn sản phẩm.
5	Xuất file báo cáo	Cho phép xuất file báo cáo thông tin nhãn sản phẩm.

- Quản lý sản phẩm cho phép quản lý các thông tin về sản phẩm:

Bảng 7 Danh sách chức năng quản lý sản phẩm

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới sản phẩm.
2.	Chỉnh sửa	Cho phép chỉnh sửa thông tin sản phẩm.
3	Xóa	Cho phép xóa sản phẩm
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách sản phẩm.
5	Xuất file báo cáo	Cho phép xuất file báo cáo thông tin sản phẩm.

- Quản lý sản phẩm gọi thêm cho phép quản lý các sản phẩm được gọi thêm trong hóa đơn:

Bảng 8 Danh sách chức năng quản lý sản phẩm gọi thêm

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới sản phẩm gọi thêm.
2.	Chỉnh sửa	Cho phép chỉnh sửa thông tin sản phẩm gọi thêm.
3	Xóa	Cho phép xóa sản phẩm gọi thêm.
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách sản phẩm gọi thêm.
5	Xuất file báo cáo	Cho phép xuất file báo cáo thông tin sản phẩm gọi thêm.

- Quản lý khuyến mãi cho phép quản lý các thông tin về chương trình khuyến mãi:

Bảng 9 Danh sách chức năng quản lý chương trình khuyến mãi

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới khuyến mãi.
2.	Chỉnh sửa	Cho phép chỉnh sửa thông tin khuyến mãi
3	Xóa	Cho phép xóa khuyến mãi.
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách khuyến mãi.
5	Xuất file báo cáo	Cho phép xuất file báo cáo thông tin khuyến mãi.

- Quản lý bàn ăn cho phép quản lý các thông tin về bàn ăn:

Bảng 10 Danh sách chức năng quản lý bàn ăn

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới bàn ăn.

2.	Chỉnh sửa	Cho phép chỉnh sửa thông tin bàn ăn.
3	Xóa	Cho phép xóa bàn ăn.
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách bàn ăn.
5	Xuất file báo cáo	Cho phép xuất file báo cáo thông tin bàn ăn.

- Quản lý lời nhắn cho phép quản lý các thông tin về lời nhắn:

Bảng 11 Danh sách chức năng quản lý lời nhắn

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới lời nhắn.
2.	Chỉnh sửa	Cho phép chỉnh sửa thông tin lời nhắn.
3	Xóa	Cho phép xóa lời nhắn.
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách lời nhắn.
5	Xuất file báo cáo	Cho phép xuất file báo cáo thông tin lời nhắn.

- Quản lý đơn hàng cho phép quản lý các thông tin về đơn hàng:

Bảng 12 Danh sách chức năng quản lý đơn hàng

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới đơn hàng.
2.	Chỉnh sửa	Cho phép chỉnh sửa thông tin đơn hàng.
3	Xóa	Cho phép xóa đơn hàng
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách đơn hàng.
5	Xuất file báo cáo	Cho phép xuất file báo cáo thông tin đơn hàng.

1.3. Ứng dụng di động

Ứng dụng di động phát triển hỗ trợ hai loại người dùng bao gồm: **Quản lý** và **nhân viên** được cấp các quyền và chức năng như sau:

- Các chức năng của nhân viên:

Bảng 13 Danh sách chức năng của nhân viên

STT	Chức năng	Diễn giải
1	Đăng nhập/Đăng xuất	Cho phép đăng nhập/đăng xuất vào ứng dụng di động.
1	Đặt món cơ bản	Cho phép thêm món ăn vào bàn ăn.
2	Thanh toán hóa đơn	Cho phép thanh toán hóa đơn của bàn ăn.
3	Báo cáo cá nhân	Cho phép hiển thị báo cáo cá nhân (Bao gồm các báo cáo về các món đã đặt và doanh thu cá nhân theo thời gian).

- Các chức năng của quản lý:

Bảng 14 Danh sách chức năng của quản lý

STT	Chức năng	Diễn giải
1	Đăng nhập/Đăng xuất	Cho phép đăng nhập/đăng xuất vào ứng dụng di động.

1	Đặt món	Cho phép thêm món ăn, hủy món ăn, hủy đơn hàng.
2	Thanh toán hóa đơn	Cho phép thanh toán hóa đơn của bàn ăn.
3	Báo cáo cá nhân	Cho phép hiển thị báo cáo cá nhân (Bao gồm các báo cáo về các món đã đặt và doanh thu cá nhân theo thời gian).
4	Báo cáo doanh thu	Cho phép hiển thị báo cáo doanh thu, báo cáo sản phẩm, báo cáo nhân sự

➤ Các chức năng chung của ứng dụng:

Bảng 15 Danh sách chức năng chung

STT	Chức năng	Diễn giải
1	Hiển thị	Hiển thị các thông tin như: Khuyến mãi, lời nhắn, bàn ăn, món ăn, ...
2	Gửi tin nhắn	Cho phép gửi tin nhắn đến người dùng khác.
3	Nhận thông báo đẩy	Cho phép nhận thông báo đẩy về thiết bị di động.

2. Phân tích hệ thống

2.1. Website quản trị

2.1.1. Mô tả hệ thống

Website quản trị được phát triển và phân quyền duy nhất cho người dùng quản lý, vì vậy website quản trị cung cấp và hỗ trợ toàn quyền và chức năng để quản lý các thông tin danh mục liên quan đến quy trình phục vụ bán hàng:

2.1.1.1. Quản lý người dùng

Người dùng ở đây là những cá nhân có nghiệp vụ và được cấp quyền truy cập vào website quản trị và ứng dụng di động. Quản lý người dùng bao gồm quản lý các thông tin về người dùng liên quan đến các trường dữ liệu để truy cập vào hệ thống và hiển thị thông tin người dùng trên hệ thống. Người dùng hệ thống được phân quyền thành hai loại người dùng: Quản lý và nhân viên, quản lý có toàn quyền và chức năng của hệ thống, nhân viên chỉ truy cập và có quyền thao tác các chức năng cơ bản. Thông người dùng quản lý và nhân viên đều được lưu trữ với các trường dữ liệu như sau:

Bảng 16 Danh sách các trường dữ liệu người dùng

STT	Trường dữ liệu	Diễn giải
1	Email Address	Địa chỉ email người dùng (Email dùng để đăng nhập vào hệ thống).
2	Password	Mật khẩu người dùng (Dùng để đăng nhập vào hệ thống).
3	First Name	Tên người dùng.
4	Last Name	Họ người dùng.
5	Contact Phone	Số liên lạc người dùng.
6	Image	Ảnh đại diện người dùng.

7	Is Active	(True/False) Trạng thái kích hoạt của tài khoản (True: Kích hoạt, người dùng có thể truy cập tài khoản vào hệ thống; False: Hủy kích hoạt, người dùng không thể truy cập tài khoản vào hệ thống).
8	Is SuperUser	(True/False) Quyền tài khoản (True: Quyền quản lý, người dùng có toàn quyền trong hệ thống; False: Không có quyền quản lý, tức nhân viên, người dùng chỉ có quyền truy cập vào ứng dụng di động và thao tác cá chức năng cơ bản).

2.1.1.2. Quản lý bàn ăn

Bàn ăn là thông tin được lưu trữ trong mỗi hóa đơn để phân biệt các món ăn phục vụ thực khách. Quản lý bàn ăn bao gồm quản lý các thông tin về bàn ăn để trích xuất dữ liệu từ hóa đơn. Đồng thời xử lý thông tin “Status” (Trạng thái) của bàn ăn để hiển thị trạng thái của bàn ăn trên ứng dụng di động:

Bảng 17 Danh sách các trường dữ liệu bàn ăn

STT	Trường dữ liệu	Diễn giải
1	Name	Tên bàn ăn.
2	Description	Mô tả, thông tin chi tiết bàn ăn.
3	Is Active	(True/False) Trạng thái kích hoạt bàn ăn (True: Kích hoạt, bàn ăn được phép sử dụng và hiển thị trên ứng dụng di động; False: Không kích hoạt, bàn ăn không được sử dụng và không hiển thị trên ứng dụng di động)
4	Status	(Ready/Ordered) Trạng thái bàn ăn (Ready: Bàn ăn sẵn sàng, chưa có thực khách ngồi cũng như chưa có hóa đơn nào tại bàn ăn; Ordered: Bàn ăn đã đặt, đã có thực khách ngồi và đã có hóa đơn tại bàn)
5	Is In Use	(True/False) Trạng thái sử dụng bàn ăn (True: Bàn ăn đang được người dùng nào đó truy cập, không cho phép người dùng sau truy cập vào; False: Bàn ăn chưa có người dùng nào truy cập, mọi người dùng đều có quyền truy cập)

2.1.1.3. Quản lý lời nhắn:

Quản lý các thông tin về lời nhắn để gửi và hiển thị lên ứng dụng di động như một lời nhắn, một thông báo đến toàn thể nhân viên.

Bảng 18 Danh sách các trường dữ liệu lời nhắn

STT	Trường dữ liệu	Diễn giải
1	Name	Tên lời nhắn.
2	Description	Mô tả, chi tiết lời nhắn (Trường dữ liệu này được hiển thị trên ứng dụng di động).
3	Text Color	Màu chữ lời nhắn (Màu chữ lời nhắn hiển thị trên ứng dụng di động).
4	Border Color	Màu viền lời nhắn (Màu viền lời nhắn hiển thị trên ứng dụng di động).

5	Is Active	(True/False) Trạng thái kích hoạt lời nhắn (True: Kích hoạt, lời nhắn hiển thị trên ứng dụng di động; False: Không kích hoạt, lời nhắn không hiển thị trên ứng dụng di động).
---	-----------	---

2.1.1.4. Quản lý khuyến mãi:

Quản lý các thông tin về các chương trình khuyến mãi và hiển thị lên ứng dụng di động để toàn thể nhân viên nắm bắt được thông tin khuyến mãi hỗ trợ quy trình phục vụ của nhân viên.

Bảng 19 Danh sách các trường dữ liệu chương trình khuyến mãi

STT	Trường dữ liệu	Diễn giải
1	Name	Tên chương trình khuyến mãi.
2	Description	Mô tả, chi tiết chương trình khuyến mãi.
3	Image	Hình ảnh chương trình khuyến mãi.
4	Start At	Thời gian chương trình khuyến mãi có hiệu lực.
5	End At	Thời gian chương trình khuyến mãi hết hiệu lực.
6	Is Active	(True/False) Trạng thái kích hoạt chương trình khuyến mãi (True: Kích hoạt, chương trình khuyến mãi hiển thị trên ứng dụng di động; False: Không kích hoạt, chương trình khuyến mãi không hiển thị trên ứng dụng di động).

Nếu thời gian hiện tại sau “Start At” và trước “End At”, tức trong khoảng thời gian chương trình khuyến mãi có hiệu lực. Trường dữ liệu “Is Active” sẽ ở trạng thái True (Kích hoạt) và ngược lại khi thời gian hiện tại nằm ngoài khoảng thời gian hiệu lực, “Is Active” sẽ ở trạng thái False (Không kích hoạt).

2.1.1.5. Quản lý loại sản phẩm:

Quản lý các thông tin về loại sản phẩm để phân loại và hiển thị trên ứng dụng di động và sắp xếp theo loại.

Bảng 20 Danh sách các trường dữ liệu loại sản phẩm

STT	Trường dữ liệu	Diễn giải
1	Name	Tên loại sản phẩm.
2	Description	Mô tả, chi tiết loại sản phẩm.

2.1.1.6. Quản lý nhãn sản phẩm:

Quản lý các thông tin nhãn sản phẩm để gắn nhãn vào sản phẩm nhằm phân biệt sản phẩm theo tính chất như (món cay, món chay)

Bảng 21 Danh sách các trường dữ liệu nhãn sản phẩm

STT	Trường dữ liệu	Diễn giải
1	Name	Tên nhãn sản phẩm.
2	Description	Mô tả, chi tiết nhãn sản phẩm.
3	Background Color	Màu nền nhãn sản phẩm hiển thị trên sản phẩm tại ứng dụng di động.

2.1.1.7. Quản lý biến thể sản phẩm

Quản lý các thông tin về biến thể sản phẩm. Mỗi sản phẩm sẽ có loại biến thể hoặc không tùy theo loại sản phẩm. Mỗi loại biến thể sẽ có các biến thể

Ví dụ: Loại biến thể “Kích cỡ” có các biến thể như “Nhỏ”, “Vừa”, “Lớn”.

Bảng 22 Danh sách các trường dữ liệu biến thể sản phẩm

STT	Trường dữ liệu	Diễn giải
1	Name	Tên loại biến thể sản phẩm.
2	Description	Mô tả, chi tiết loại biến thể sản phẩm.
	Variation Options	Các thông tin chi tiết biến thể
1	Name	Tên biến thể
2	Description	Mô tả, chi tiết biến thể.

2.1.1.8. Quản lý sản phẩm

Quản lý các thông tin về sản phẩm. Mỗi sản phẩm sẽ có các thông tin chung và các thông tin biến thể.

Ví dụ: Sản phẩm “Com gà” có các thông tin chung như tên, hình ảnh,... và các thông tin biến thể như “Cỡ nhỏ” có giá 20.000 VNĐ và “Cỡ lớn” có giá 25.000 VNĐ

Bảng 23 Danh sách các trường dữ liệu sản phẩm

STT	Trường dữ liệu	Diễn giải
1	Name	Tên sản phẩm.
2	Description	Mô tả, chi tiết sản phẩm.
3	Category	Loại sản phẩm.
4	Labels	Nhãn sản phẩm (Mỗi sản phẩm có thể có một, nhiều hoặc không có nhãn)
5	Variations	Loại biến thể sản phẩm (Mỗi sản phẩm có thể có một, nhiều, hoặc không có loại biến thể)
6	Image	Hình ảnh sản phẩm.
7	Is Active	(True/False) Trạng thái kích hoạt sản phẩm (True: Kích hoạt, sản phẩm được phép bán và hiển thị trên ứng dụng di động; False: Không kích hoạt, sản phẩm không được phép bán và không hiển thị trên ứng dụng di động).
	Product Variation Options	Các thông tin chi tiết sản phẩm biến thể
1	Variation Options	Tên biến thể.
2	Price	Giá sản phẩm.

2.1.1.9. Quản lý sản phẩm gọi thêm

Quản lý các thông tin về sản phẩm gọi thêm. Các sản phẩm được đều được cho phép đặt sản phẩm gọi thêm trong quá trình đặt món.

Ví dụ: Sản phẩm “Cơm gà” đặt thêm “trứng” và “xúc xích”, trứng và xúc xích ở đây chính là sản phẩm gọi thêm.

Bảng 24 Danh sách các trường dữ liệu sản phẩm gọi thêm

STT	Trường dữ liệu	Diễn giải
1	Name	Tên sản phẩm gọi thêm.
2	Description	Mô tả, chi tiết sản phẩm gọi thêm.
3	Is Active	(True/False) Trạng thái kích hoạt sản phẩm gọi thêm (True: Kích hoạt, sản phẩm gọi thêm được phép bán và hiển thị trên ứng dụng di động; False: Không kích hoạt, sản phẩm gọi thêm không được phép bán và không hiển thị trên ứng dụng di động).
4	Price	Giá sản phẩm gọi thêm.

2.1.1.10. Quản lý đơn hàng

Quản lý cá thông tin về đơn hàng và các thông tin chi tiết đơn hàng. Mỗi đơn hàng sẽ có thông tin chung và chi tiết đơn hàng.

Ví dụ: Đơn hàng mã “AA1” có các thông tin chung như tên khách hàng, thời gian đặt hàng, tổng tiền và các thông tin chi tiết đơn hàng như tên sản phẩm, giá sản phẩm,...

Bảng 25 Danh sách các trường dữ liệu đơn hàng

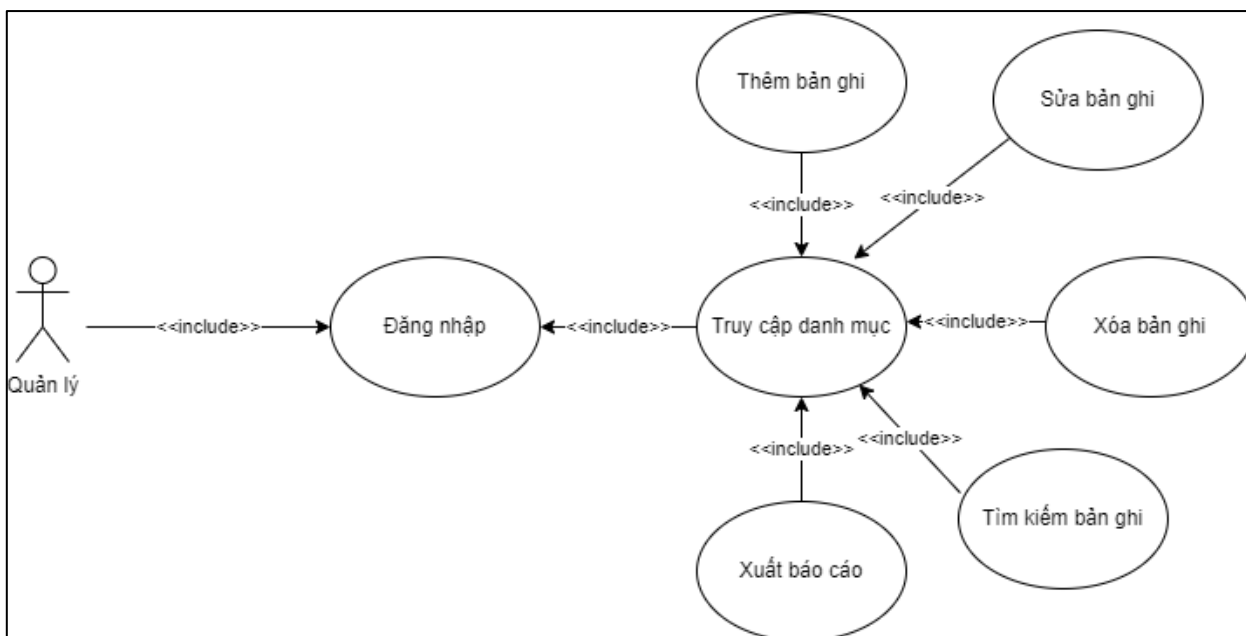
STT	Trường dữ liệu	Diễn giải
1	Table	Tên bàn ăn. (Dùng để phân biệt đơn hàng và món ăn của thực khách)
2	Paid By	Người dùng đã thanh toán (Khi nhân viên thanh toán hóa đơn thông tin người dùng đó sẽ được lưu trữ tại trường này để tiện lợi cho việc trích xuất và kiểm tra)
3	Status	(Serving/Paid/Cancelled) Trạng thái hóa đơn (Serving: Đang phục vụ, hóa đơn đang trong quá trình phục vụ, tức cho phép thêm, xóa, sửa đơn hàng và hóa đơn được hiển thị trên ứng dụng di động; Paid: Đã thanh toán, không cho phép thao tác và không hiển thị trên ứng dụng di động; Cancelled: Đã hủy, không cho phép thao tác và không hiển thị trên ứng dụng di động)
4	Total Price	Tổng giá hóa đơn (Được tính bằng tổng giá của các chi tiết hóa đơn)
	Order Items	Chi tiết hóa đơn
1	Product	Sản phẩm.
2	Varitation Options	Biến thể sản phẩm.
3	Extras	Sản phẩm gọi thêm.
4	User	Người dùng đặt món (Nhân viên đặt món sẽ được lưu trữ dữ liệu tại trường này để tiện lợi cho việc trích xuất và kiểm tra)

5	Quantity	Số lượng sản phẩm
6	Note	Ghi chú (Yêu cầu của khách hàng về món ăn như: Ít cay, nấu chín,...)
7	Total Price	Tổng giá chi tiết hóa đơn (Được tính tích giá sản phẩm, số lượng và giá sản phẩm gọi thêm)

2.1.2. Xác định use case

Website quản trị được phát triển và phân quyền duy nhất một loại người dùng đó là quản lý. Quản lý là người quản lý nhà hàng và nắm toàn quyền của website quản trị. Người quản lý có những chức năng như sau:

- Đăng nhập vào hệ thống.
- Quản trị website.
- Truy cập vào danh mục.
- Thêm bản ghi vào danh mục.
- Sửa bản ghi.
- Xóa bản ghi.
- Tìm kiếm bản ghi.
- Xuất báo cáo.



Sơ đồ 3 Sơ đồ usecase website quản trị của quản lý

2.1.3. Đặc tả use case

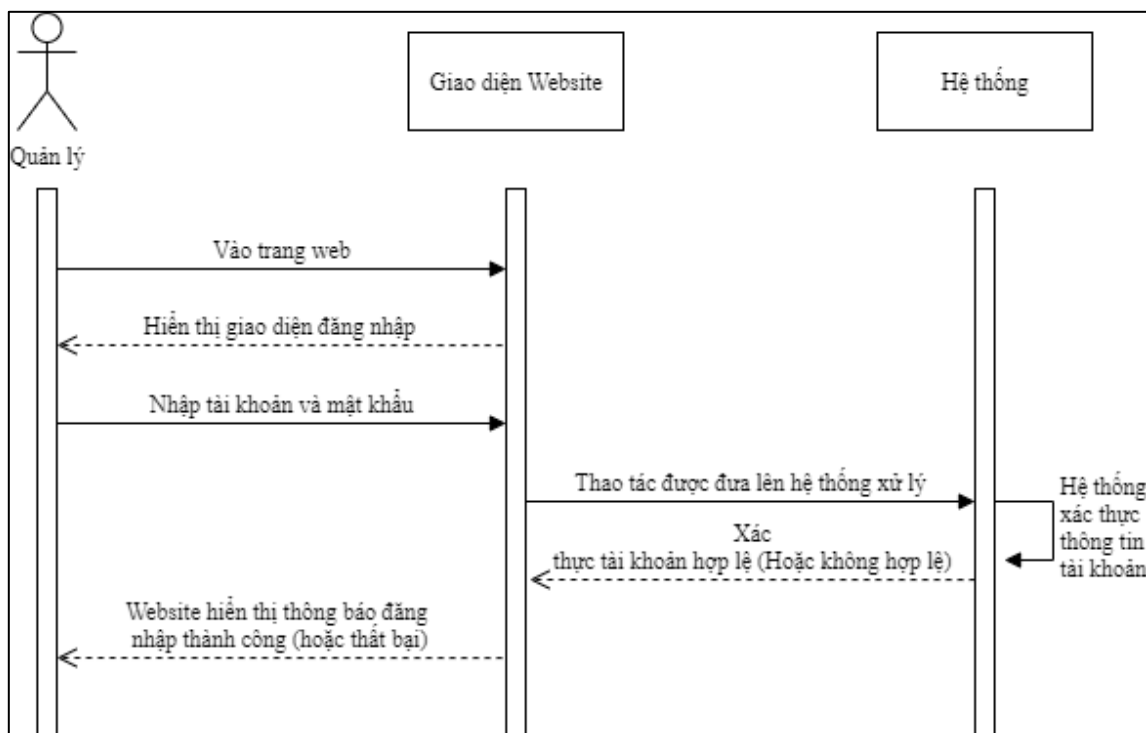
2.1.3.1. Chức năng đăng nhập

Người quản lý thực hiện chức năng đăng nhập vào hệ thống bằng cách truy cập vào website quản trị, sau đó website quản trị hiển thị giao diện đăng nhập yêu cầu nhập email và password. Người quản lý nhập đầy đủ thông tin và nhấn nút “Đăng nhập”. Nếu thông tin xác thực chính xác, website thông báo đăng nhập thành công và chuyển hướng đến giao diện chính, nếu thông tin đăng nhập không chính xác, website

thông báo lỗi cho người quản lý, không chuyển hướng và yêu cầu đăng nhập lại. Quy trình như sau:

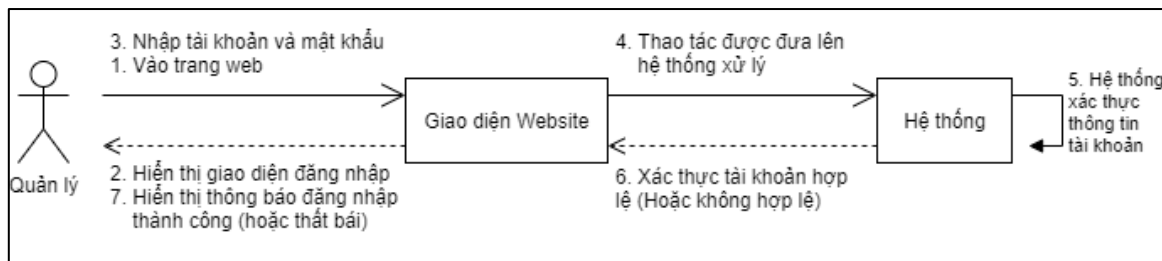
- Quản lý vào Website;
- Website hiện giao diện đăng nhập;
- Quản lý nhập tài khoản và mật khẩu;
- Thao tác được đưa lên hệ thống xử lý;
- Hệ thống xác thực thông tin tài khoản;
- Xác thực tài khoản hợp lệ (Hoặc không hợp lệ);
- Website hiển thị thông báo đăng nhập thành công (hoặc thất bại).

Sơ đồ tuần tự:



Sơ đồ 4 Sơ đồ tuần tự chức năng đăng nhập website quản trị

Sơ đồ hợp tác:



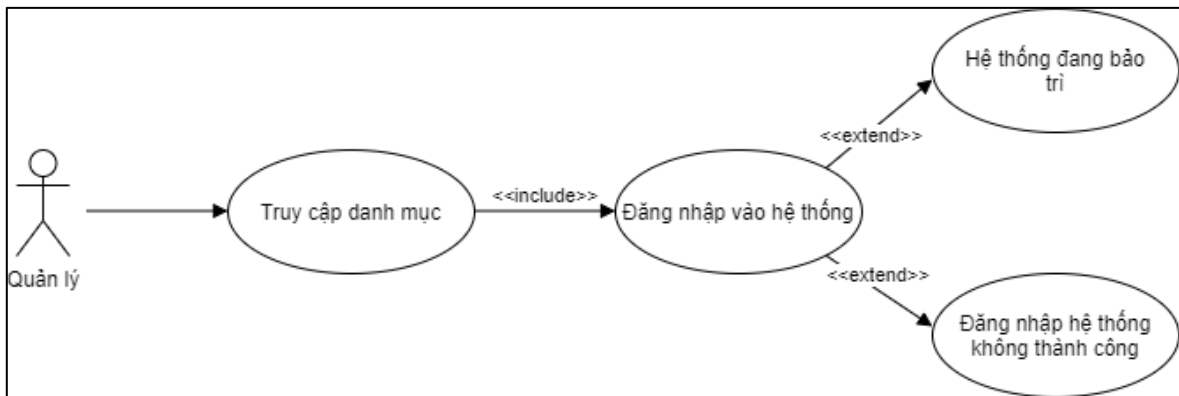
Sơ đồ 5 Sơ đồ hợp tác chức năng đăng nhập website quản trị

2.1.3.2. Chức năng truy cập danh mục

Sau khi người quản lý đăng nhập thành công, website điều hướng sang giao diện chính. Tại đây hiển thị các danh mục chính như đã phân tích trên (Người dùng, Chương trình khuyến mãi, Sản phẩm,...). Người quản lý chọn danh mục muốn truy cập, sau đó website hiển thị giao diện của danh mục. Quy trình như sau:

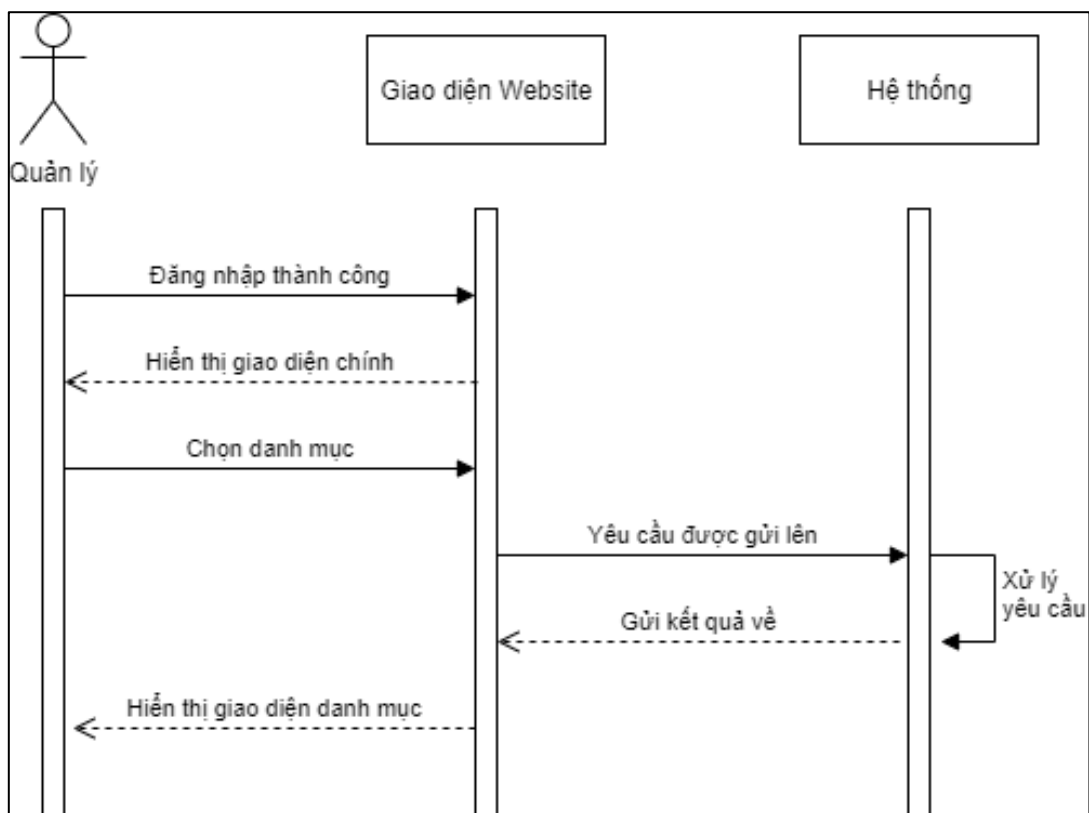
- Quản lý đăng nhập vào Website thành công;
- Website hiển thị giao diện chính;
- Quản lý chọn danh mục;
- Website gửi yêu cầu về hệ thống xử lý;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về Website.
- Website hiển thị giao diện danh mục.

Sơ đồ giữa tác nhân và use case:



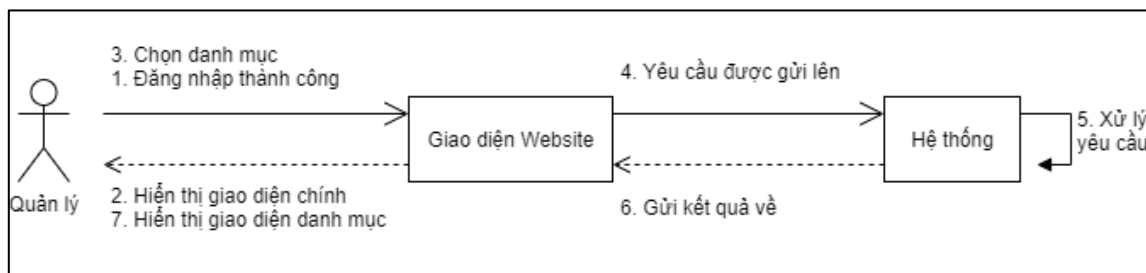
Sơ đồ 6 Sơ đồ tác nhân - usecase chức năng truy cập danh mục

Sơ đồ tuần tự:



Sơ đồ 7 Sơ đồ tuần tự chức năng truy cập danh mục

Sơ đồ hợp tác:



Sơ đồ 8 Sơ đồ hợp tác chức năng truy cập danh mục

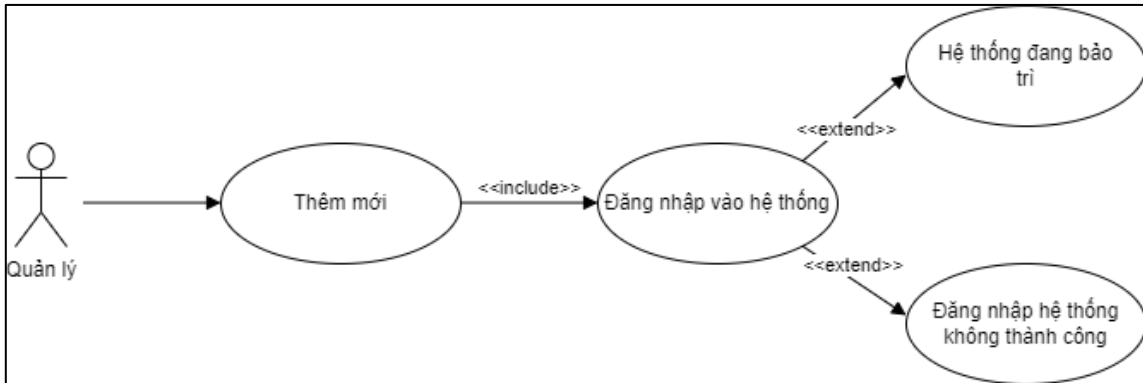
2.1.3.3. Chức năng thêm

Sau khi truy cập một danh mục thành công, người quản lý có thể thực hiện tác vụ thêm bằng cách nhấn nút “Thêm”, sau đó website hiển thị form để nhập thông tin. Người quản lý nhập đầy đủ thông tin và nhấn nút “Save” để gửi yêu cầu xử lý lên hệ thống. Nếu hệ thống xác thực thông tin hợp lệ thì hiển thị kết quả thành công, ngược lại trả về thông báo thất bại. Quy trình như sau:

- Quản lý đăng nhập vào Website thành công;
- Quản lý truy cập danh mục thành công;
- Quản lý nhấn nút “Thêm” để mở form nhập thông tin;
- Quản lý thực hiện nhập thông tin;

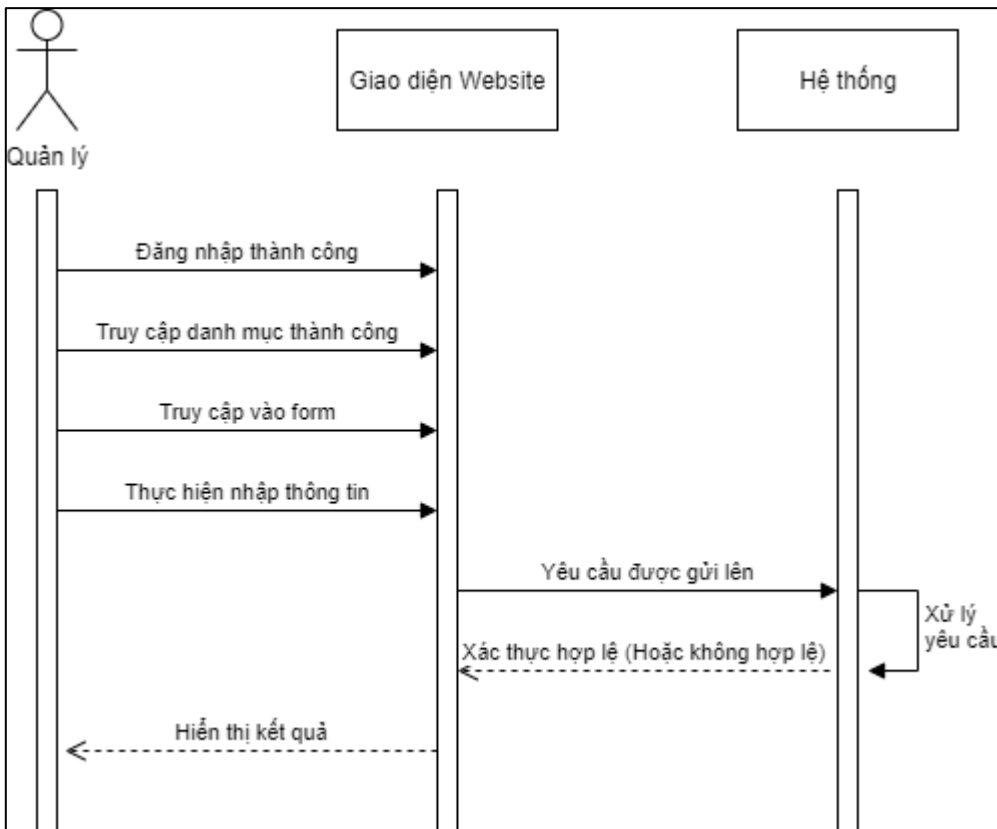
- Website gửi yêu cầu về hệ thống xử lý;
- Hệ thống xử lý yêu cầu;
- Hệ thống xác thực hợp lệ (Hoặc không hợp lệ);
- Website hiển thị kết quả

Sơ đồ giữa tác nhân và use case:



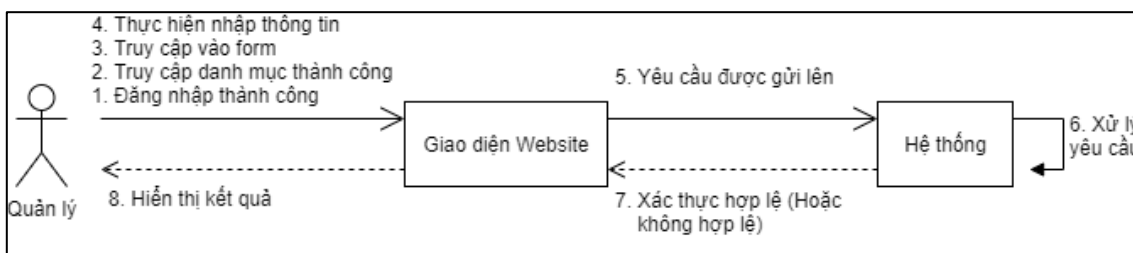
Sơ đồ 9 Sơ đồ tác nhân - usecase chức năng thêm

Sơ đồ tuần tự:



Sơ đồ 10 Sơ đồ tuần tự chức năng thêm

Sơ đồ hợp tác:



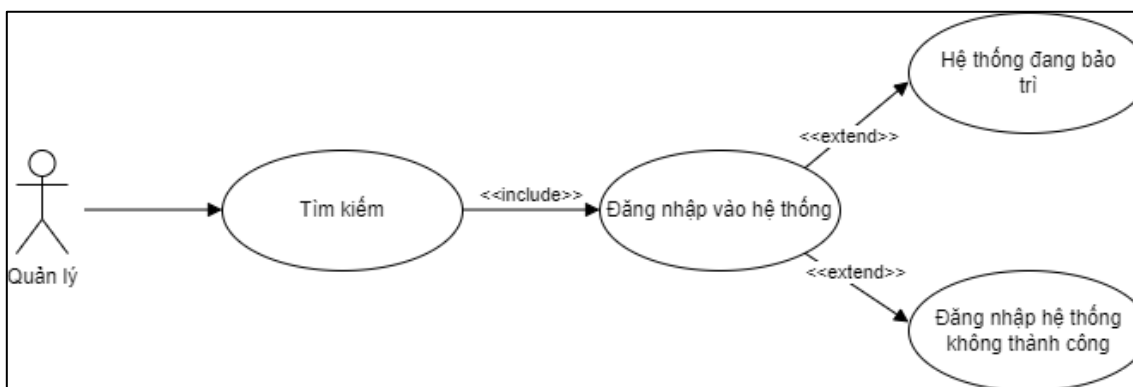
Sơ đồ 11 Sơ đồ hợp tác chức năng thêm

2.1.3.4. Chức năng tìm kiếm

Sau khi truy cập một danh mục thành công, người quản lý có khả năng thực hiện tác vụ tìm kiếm bằng cách nhập thông tin cần tìm kiếm vào thanh tìm kiếm và nhấn nút “Tìm”. Hệ thống xử lý tác vụ và gửi về kết quả tìm kiếm cho người quản lý. Quy trình như sau:

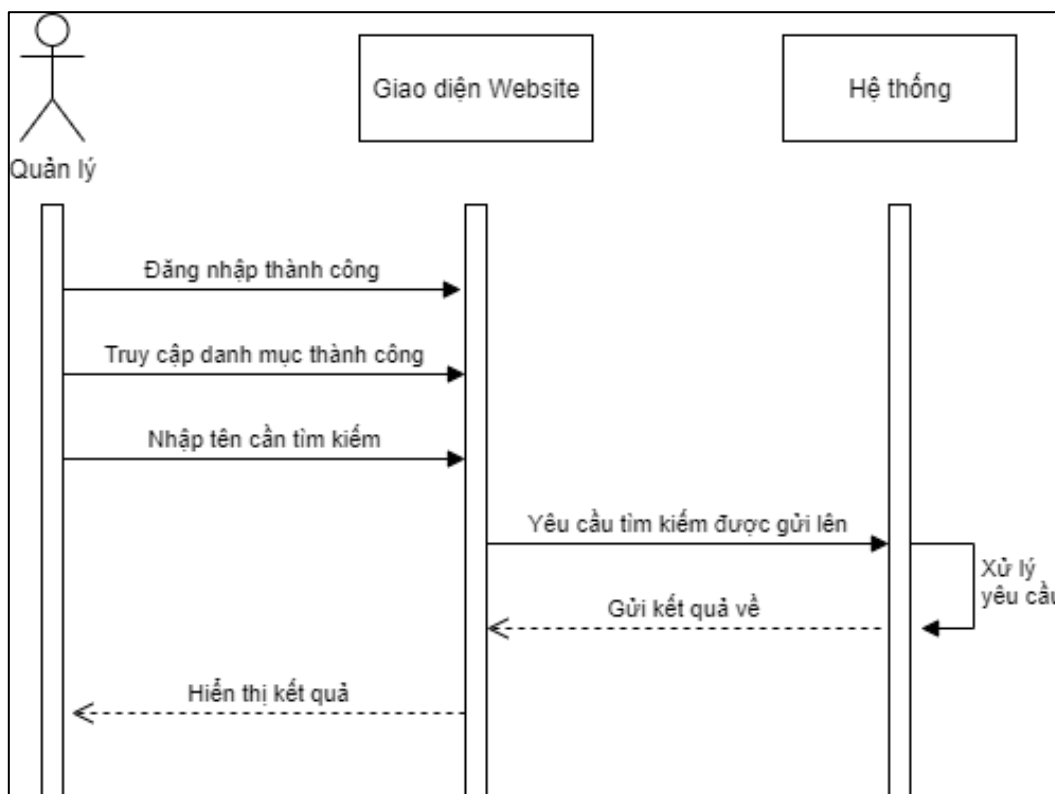
- Quản lý đăng nhập vào Website thành công;
- Quản lý truy cập danh mục thành công;
- Quản lý nhập tên cần tìm kiếm;
- Website gửi yêu cầu về hệ thống xử lý;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về Website.
- Website hiển thị kết quả

Sơ đồ giữa tác nhân và use case:



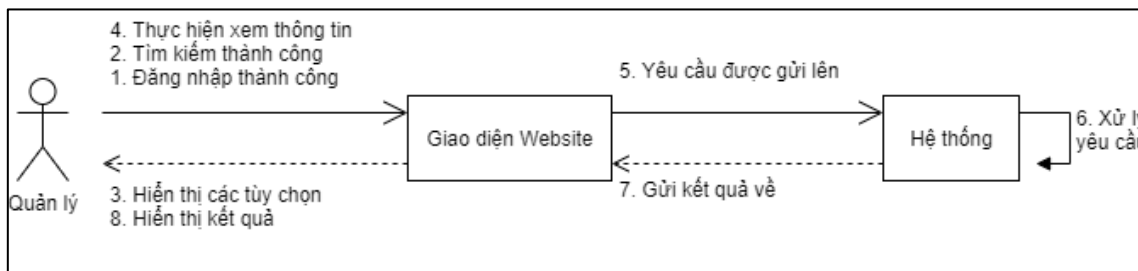
Sơ đồ 12 Sơ đồ tác nhân - usecase chức năng tìm kiếm

Sơ đồ tuần tự:



Sơ đồ 13 Sơ đồ tuần tự chức năng tìm kiếm

Sơ đồ hợp tác:



Sơ đồ 14 Sơ đồ hợp tác chức năng tìm kiếm

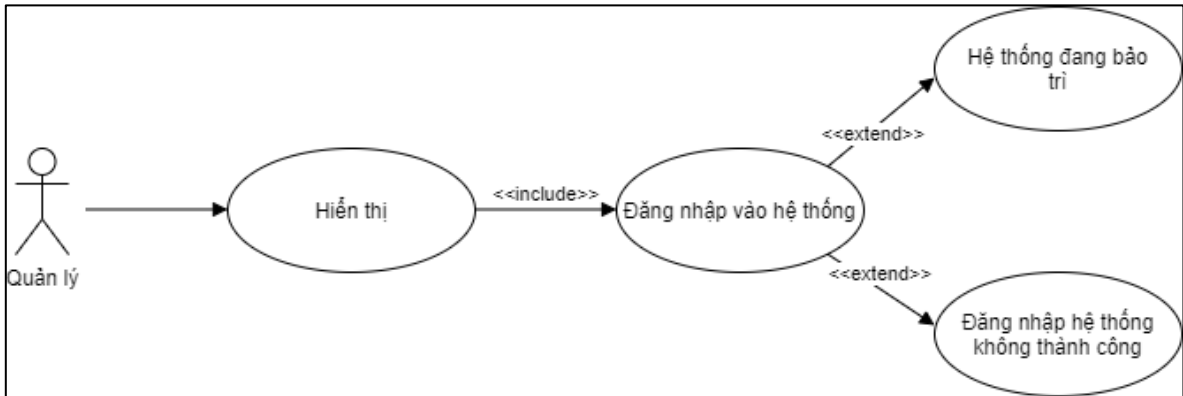
2.1.3.5. Chức năng hiển thị

Sau khi tìm kiếm thành công, người quản lý được phép truy cập vào các bản ghi dữ liệu bằng cách nhấp vào bản ghi mong muốn. Sau đó website điều hướng và hiển thị giao diện với các thông tin chi tiết của bản ghi. Quy trình như sau:

- Quản lý đăng nhập Website thành công;
- Quản lý tìm kiếm thành công;
- Website hiển thị danh sách bản ghi;
- Quản lý chọn “Xem thông tin” bằng cách nhấn vào bản ghi;
- Website gửi yêu cầu về hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi thông tin lên Website;

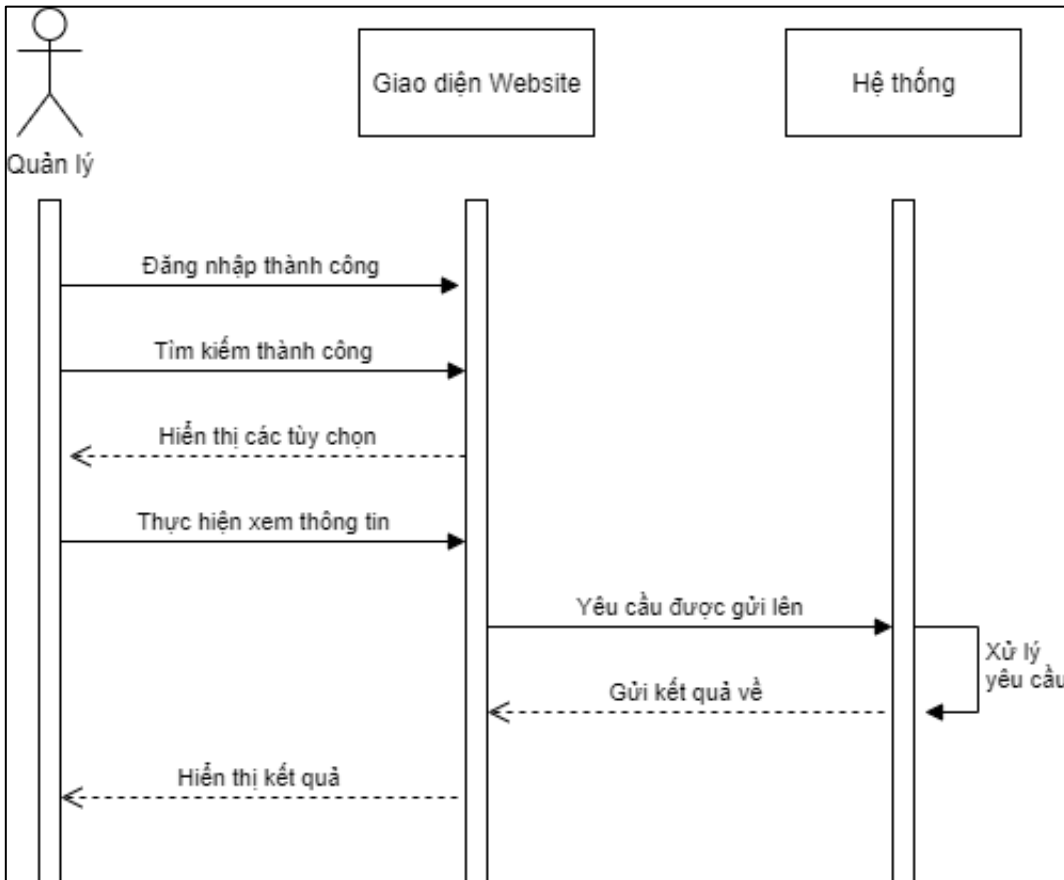
- Website hiển thị thông tin;

Sơ đồ giữa tác nhân và use case:



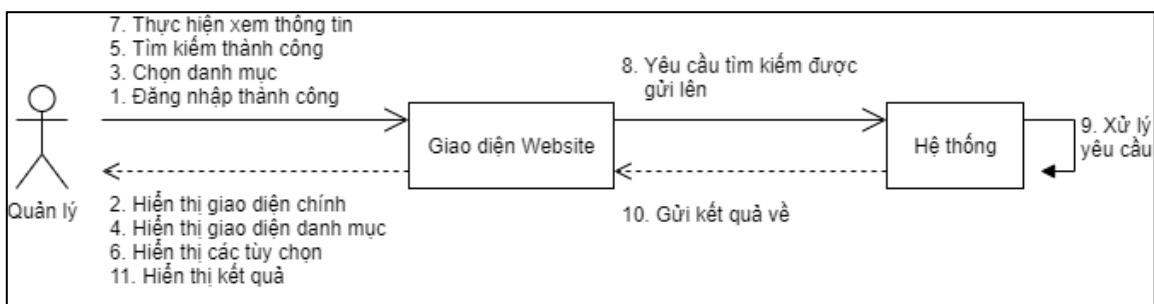
Sơ đồ 15 Sơ đồ tác nhân - usecase chức năng hiển thị

Sơ đồ tuần tự:



Sơ đồ 16 Sơ đồ tuần tự chức năng hiển thị

Sơ đồ hợp tác:

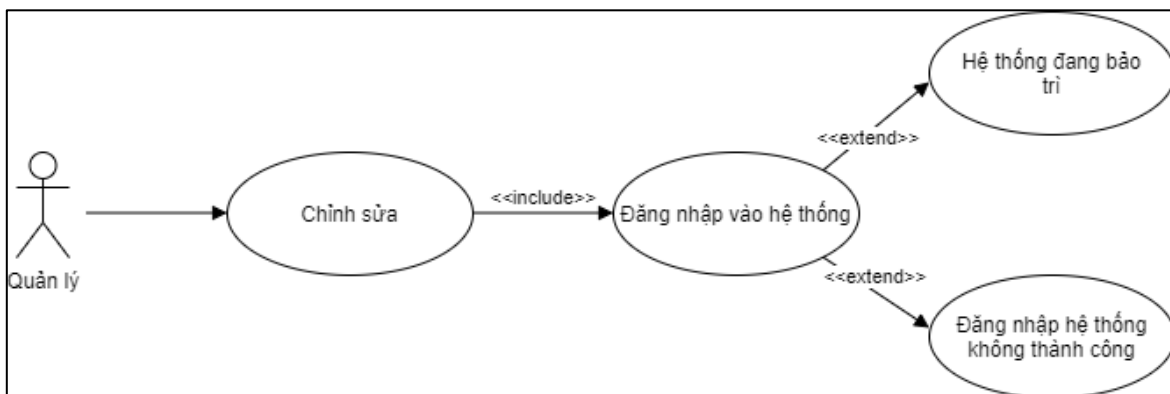


2.1.3.6. Chức năng chỉnh sửa

Sau khi truy cập vào một bản ghi thành công, tại đây người quản lý có quyền chỉnh sửa bản ghi bằng cách thay đổi các trường thông tin và nhấn nút “Lưu”. Hệ thống xử lý yêu cầu và trả về kết quả. Quy trình như sau:

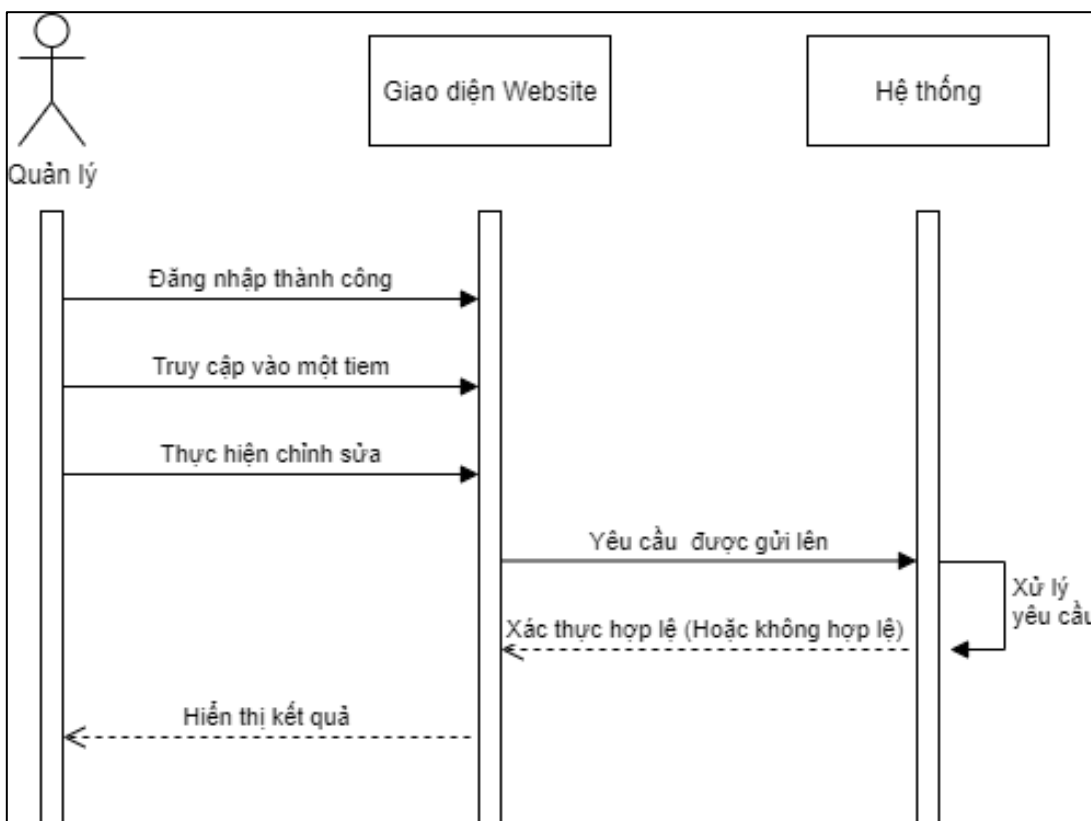
- Quản lý đăng nhập Website thành công;
- Quản lý truy cập vào một bản ghi;
- Quản lý “Chỉnh sửa” bằng cách thay đổi các trường thông tin;
- Website gửi yêu cầu về hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống xác thực hợp lệ (Hoặc không hợp lệ);
- Website hiển thị thông báo thành công (hoặc thất bại).

Sơ đồ giữa tác nhân và use case:



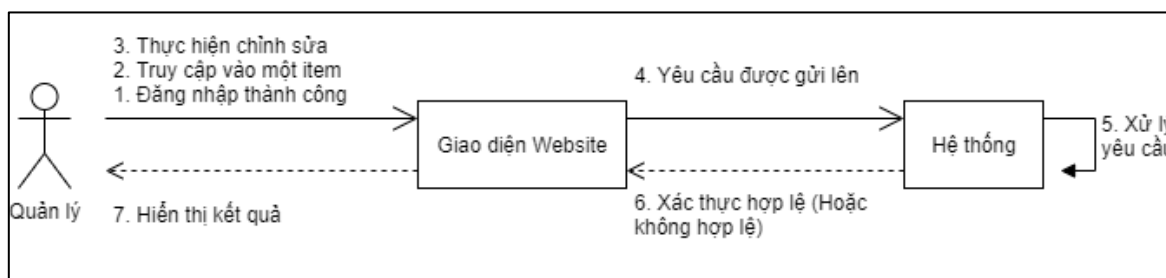
Sơ đồ 17 Sơ đồ tác nhân - usecase chức năng chỉnh sửa

Sơ đồ tuần tự:



Sơ đồ 18 Sơ đồ tuần tự chức năng chỉnh sửa

Sơ đồ hợp tác:



Sơ đồ 19 Sơ đồ hợp tác chức năng chỉnh sửa

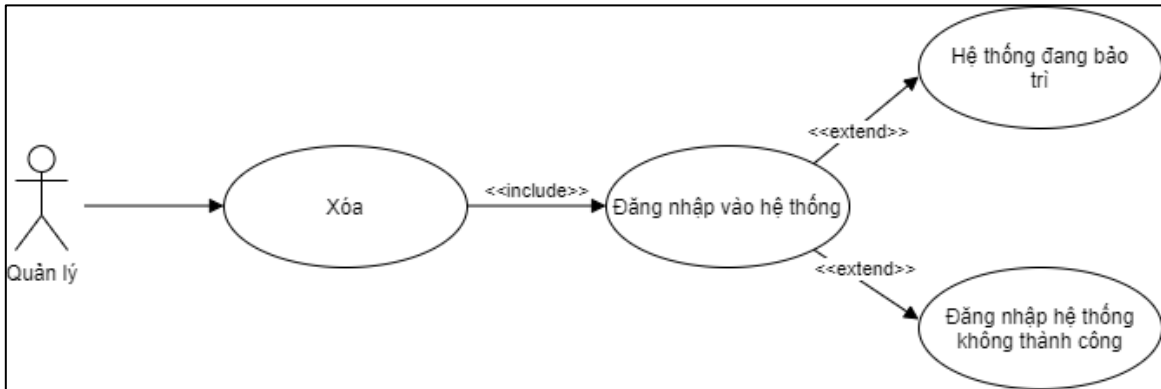
2.1.3.7. Chức năng xóa

Để thực hiện thao tác xóa một bản ghi, người quản lý cần truy cập vào một bản ghi và nhấn nút “Xóa”, hệ thống xử lý yêu cầu và trả về kết quả. Quy trình như sau:

- Quản lý đăng nhập thành công;
- Quản lý tìm kiếm thành công;
- Quản lý lựa chọn các bản ghi;
- Quản lý thực hiện “Xóa” bằng cách nhấn nút Xóa;
- Website gửi yêu cầu về hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả lên Website;

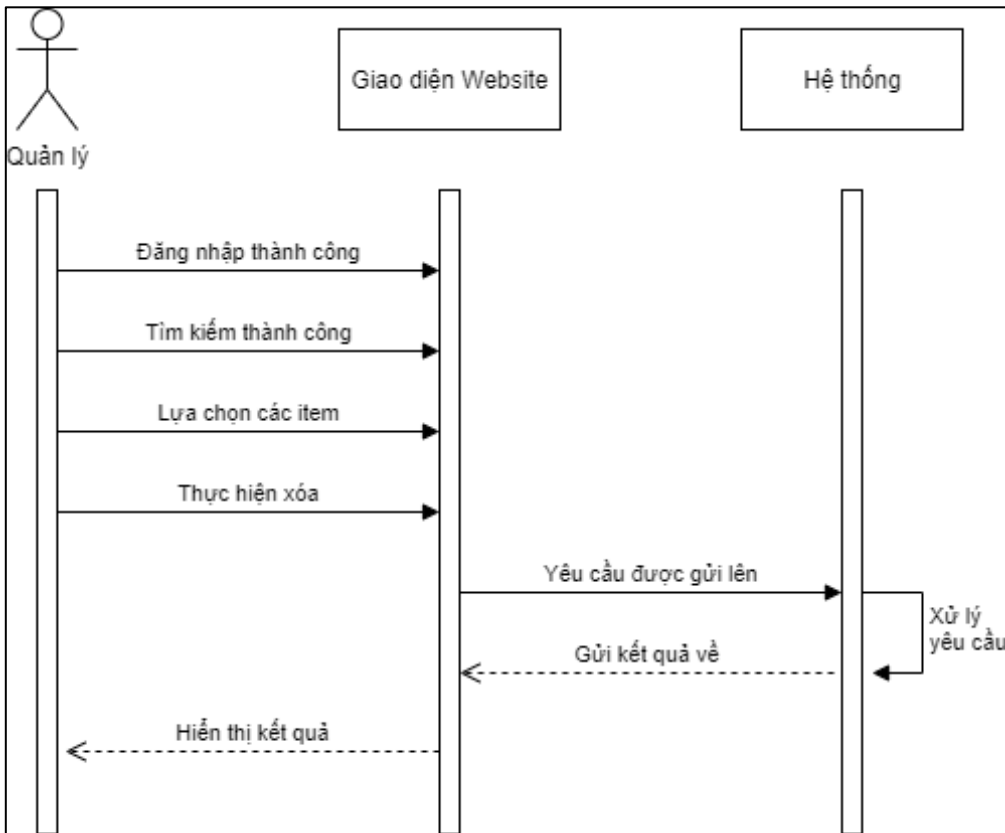
- Website hiển thị kết quả.

Sơ đồ giữa tác nhân và use case:



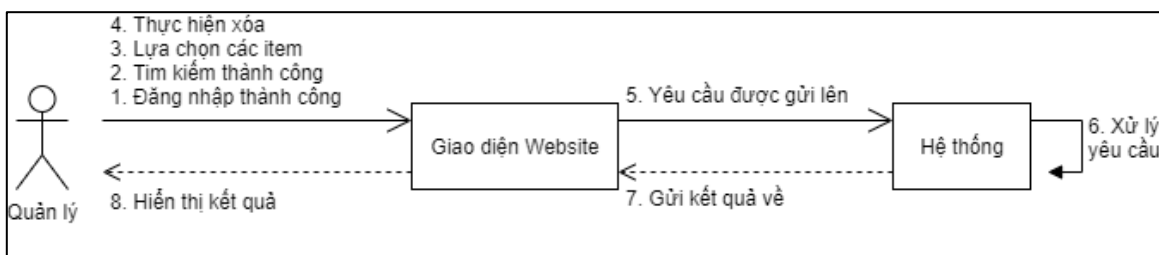
Sơ đồ 20 Sơ đồ tác nhân - usecase chức năng xóa

Sơ đồ tuần tự:



Sơ đồ 21 Sơ đồ tuần tự chức năng xóa

Sơ đồ hợp tác:



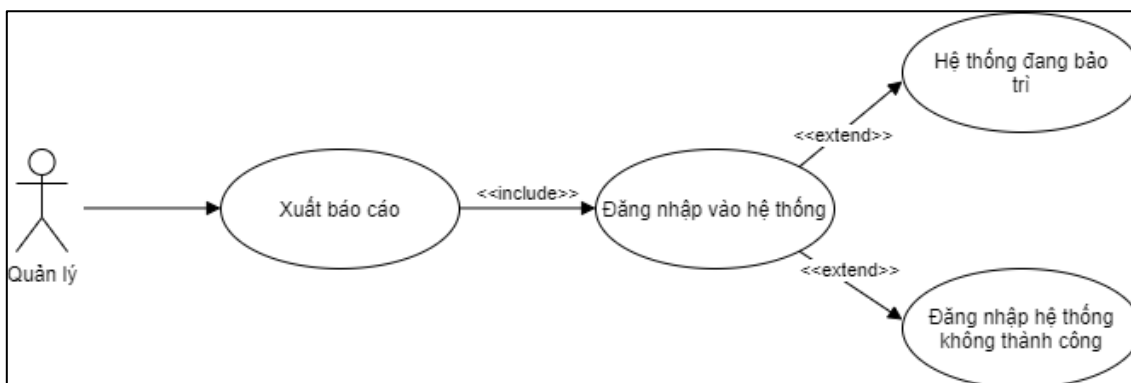
Sơ đồ 22 Sơ đồ hợp tác chức năng xóa

2.1.3.8. Chức năng xuất báo cáo

Người quản lý được cung cấp chức năng xuất báo cáo phục vụ cho công tác đánh giá chất lượng dịch vụ. Để thao tác xuất báo cáo, người quản lý cần truy cập vào một bản ghi, lựa chọn kiểu xuất file và nhấn nút “Xuất”. Hệ thống xử lý yêu cầu và trả về kết quả. Quy trình như sau:

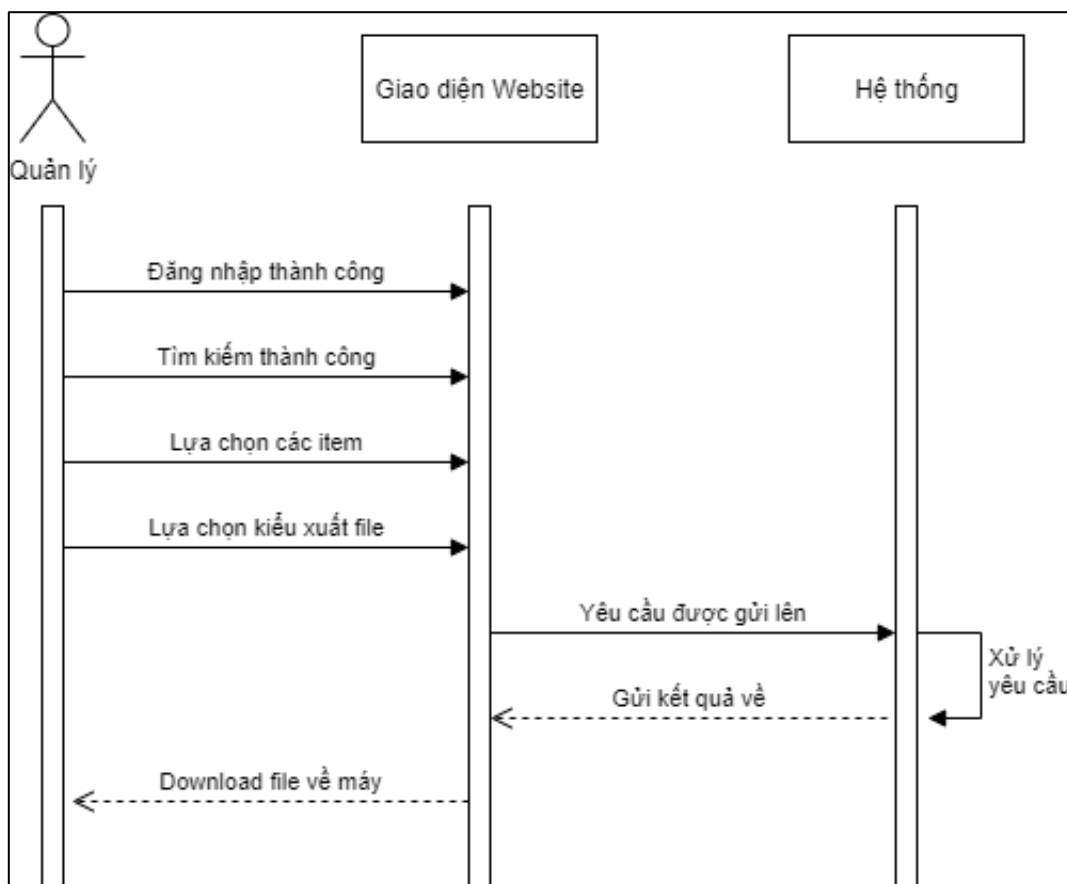
- Quản lý đăng nhập thành công;
- Quản lý tìm kiếm thành công;
- Quản lý lựa chọn các bản ghi;
- Quản lý lựa chọn kiểu xuất file;
- Website gửi yêu cầu về hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả lên Website;
- Website download file về máy.

Sơ đồ giữa tác nhân và use case:



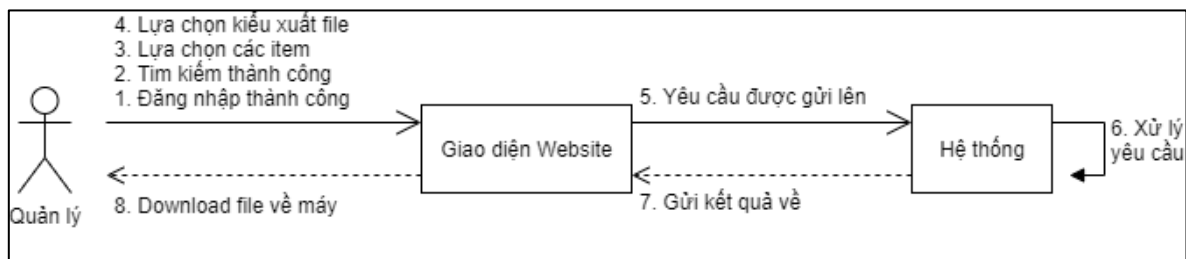
Sơ đồ 23 Sơ đồ tác nhân - usecase chức năng xuất báo cáo

Sơ đồ tuần tự:



Sơ đồ 24 Sơ đồ tuần tự chức năng xuất báo cáo

Sơ đồ hợp tác:



Sơ đồ 25 Sơ đồ hợp tác chức năng xuất báo cáo

2.2. Ứng dụng di động

2.2.1. Mô tả hệ thống

Để đăng nhập vào ứng dụng di động, người dùng sử dụng tài khoản được cấp bởi quản lý. Thông tin tài khoản đăng nhập bao gồm: Email và Password. Tài khoản này người dùng không thể tự tạo hay thay đổi.

Ứng dụng hiển thị các thông tin được gửi từ website quản trị như thông tin lời nhắn và chương trình khuyến mãi theo danh sách. Chương trình khuyến mãi hiển thị theo hình ảnh, người dùng truy cập thông tin chi tiết khuyến mãi bằng thao tác nhấp vào hình ảnh.

Ứng dụng hiển thị thông tin bàn ăn theo danh sách. Mỗi bàn ăn được thể hiện trạng thái (Ready/Ordered/InUse) theo màu sắc. Ở trạng thái Ready (Bàn ăn có sẵn) và Ordered (Bàn ăn đã đặt món) người dùng được phép truy cập. Ở trạng thái InUse (Một người dùng đang truy cập) không người dùng nào khác được phép truy cập.

Khi người dùng truy cập bàn ăn thành công, bàn ăn chuyển trạng thái về InUse không cho phép người dùng khác truy cập. Tại đây ứng dụng hiển thị các thông tin về đơn hàng và cho phép người dùng chỉnh sửa đơn hàng (Đặt thêm/ Hủy món).

Người dùng ứng dụng tính năng “Nhắn tin” trong quá trình phục vụ để giao tiếp và trao đổi thông tin với nhân viên khác. Ứng dụng hiển thị những người dùng sẵn sàng để nhắn tin và những người dùng vừa nhắn tin.

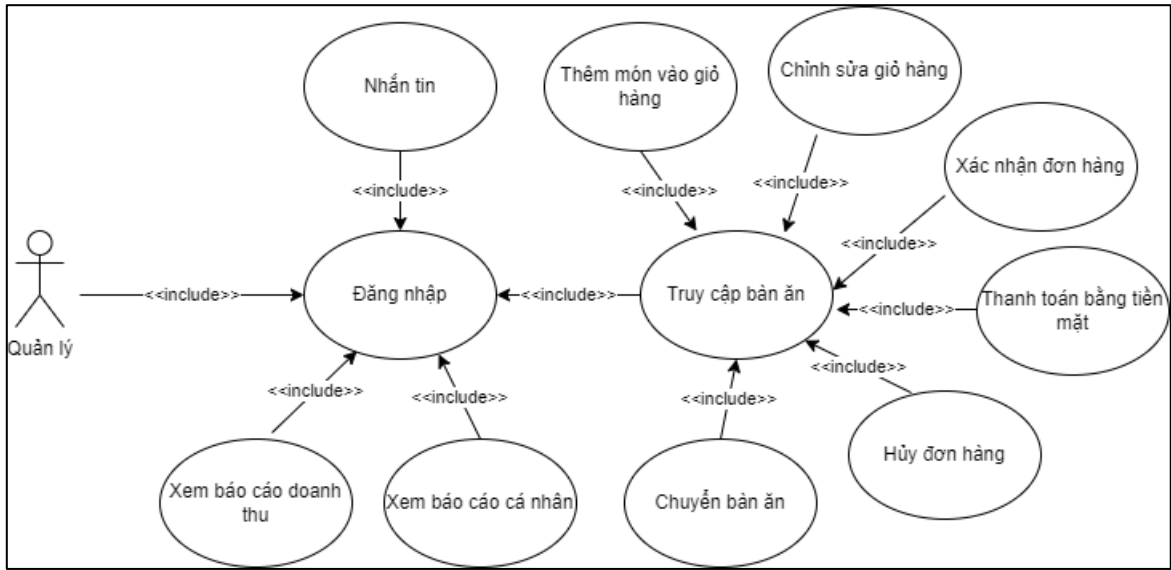
Ứng dụng xử lý và phản hồi dữ liệu thời gian thực và đồng bộ. Giải quyết các vấn đề về tính chính xác, linh hoạt, nhanh chóng và tránh xảy ra xung đột trong quá trình sử dụng.

2.2.2. Xác định use case

Ứng dụng di động phân quyền hai loại người dùng: *Quản lý* và *nhân viên*

2.2.2.1. Quản lý

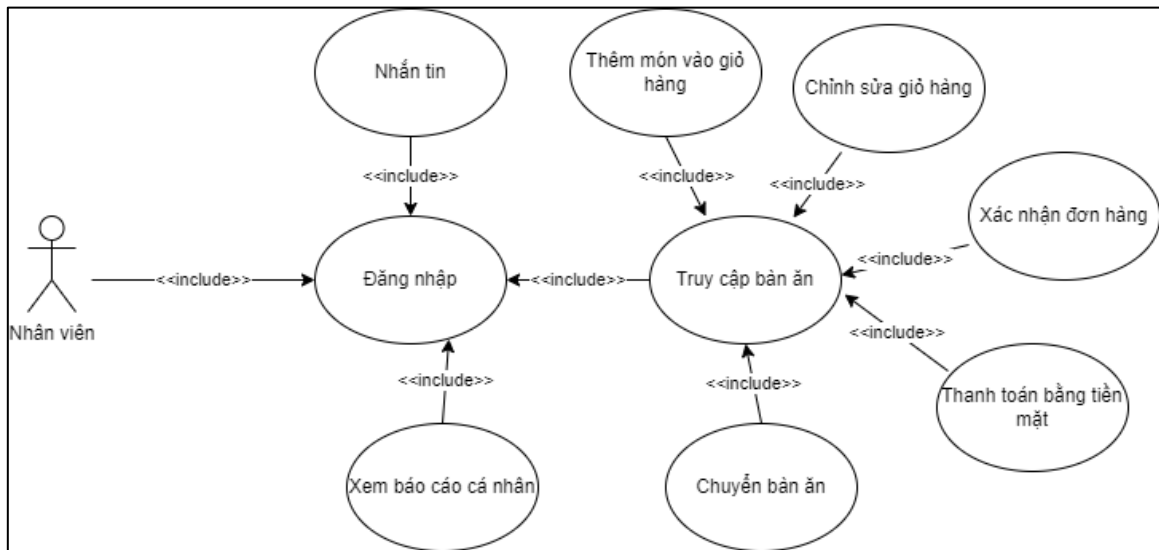
- Đăng nhập vào ứng dụng.
- Truy cập bàn ăn.
- Thêm món vào giỏ hàng.
- Chỉnh sửa giỏ hàng.
- Xác nhận đơn hàng.
- Thanh toán bằng tiền mặt.
- Hủy đơn hàng.
- Chuyển bàn ăn.
- Nhắn tin.
- Xem báo cáo cá nhân.
- Xem báo cáo doanh thu.



Sơ đồ 26 Sơ đồ usecase ứng dụng di động của quản lý

2.2.2.2. Nhân viên

- Đăng nhập vào ứng dụng.
- Truy cập bản ăn.
- Thêm món vào giỏ hàng.
- Chỉnh sửa giỏ hàng.
- Xác nhận đơn hàng.
- Thanh toán bằng tiền mặt.
- Chuyển bản ăn.
- Nhấn tin.
- Xem báo cáo cá nhân.



Sơ đồ 27 Sơ đồ usecase ứng dụng di động của nhân viên

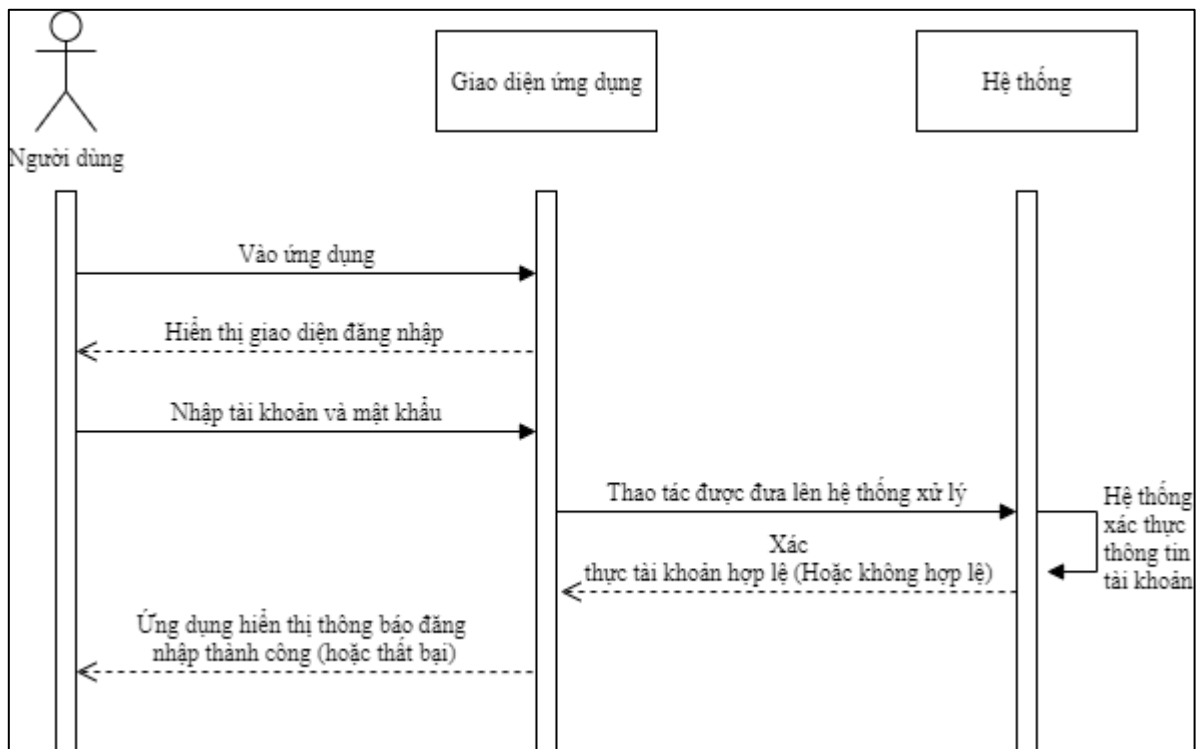
2.2.3. Đặc tả use case

2.2.3.1. Chức năng đăng nhập:

Người dùng thực hiện chức năng đăng nhập vào hệ thống bằng cách truy cập vào ứng dụng di động trên thiết bị, sau đó ứng dụng hiển thị giao diện đăng nhập yêu cầu nhập email và password. Người dùng nhập đầy đủ thông tin và nhấn nút “Đăng nhập”. Nếu thông tin xác thực chính xác, ứng dụng thông báo đăng nhập thành công và chuyển hướng đến giao diện chính, nếu thông tin đăng nhập không chính xác, ứng dụng thông báo lỗi, không chuyển hướng và yêu cầu đăng nhập lại. Quy trình như sau:

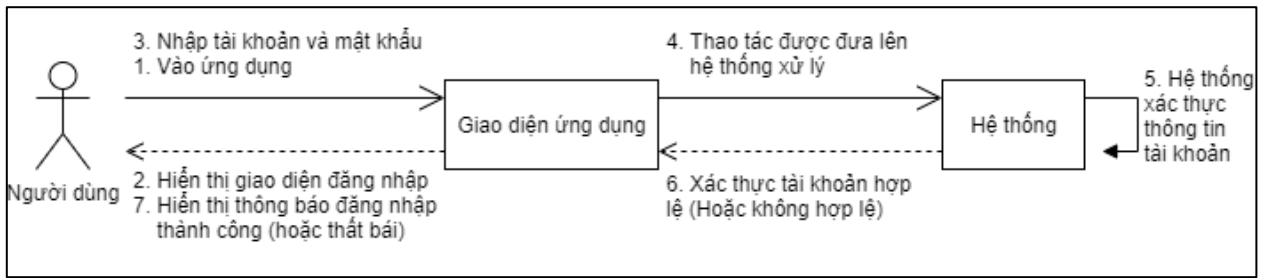
- Người dùng vào ứng dụng;
- Ứng dụng hiển thị giao diện chào mừng;
- Người dùng chọn chức năng đăng nhập;
- Người dùng nhập tài khoản và mật khẩu;
- Thao tác được đưa lên hệ thống xử lý;
- Hệ thống xác thực thông tin tài khoản;
- Xác thực tài khoản hợp lệ (hoặc không hợp lệ)
- Ứng dụng hiển thị thông báo đăng nhập thành công (hoặc thất bại),

Sơ đồ tuần tự:



Sơ đồ 28 Sơ đồ tuần tự chức năng đăng nhập ứng dụng di động

Sơ đồ hợp tác:



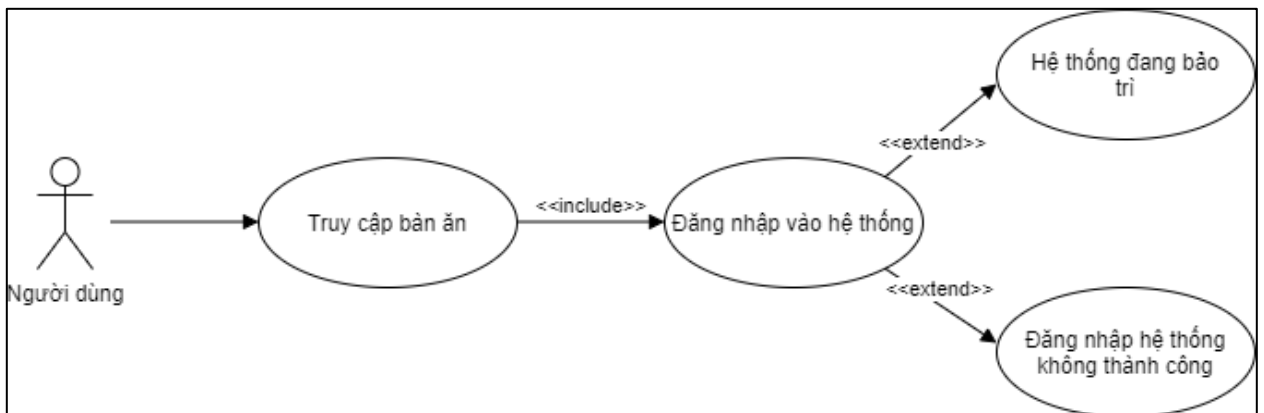
Sơ đồ 29 Sơ đồ hợp tác chức năng đăng nhập ứng dụng di động

2.2.3.2. Chức năng truy cập bàn ăn:

Sau khi đăng nhập thành công, người dùng truy cập bàn ăn bằng cách chuyển sang tab “Bàn ăn” tại màn hình chính và chọn bàn ăn để truy cập. Nếu bàn ăn ở trạng thái In Use, ứng dụng không cho phép truy cập, ngược lại người dùng được phép truy cập và ứng dụng chuyển hướng người dùng đến màn hình chi tiết bàn ăn. Hệ thống chuyển trạng thái bàn ăn về In Use. Khi thao tác tới bàn ăn thành công, người dùng nhấn “Back” chuyển về màn hình chính, bỏ trạng thái In Use của bàn ăn. Quy trình như sau:

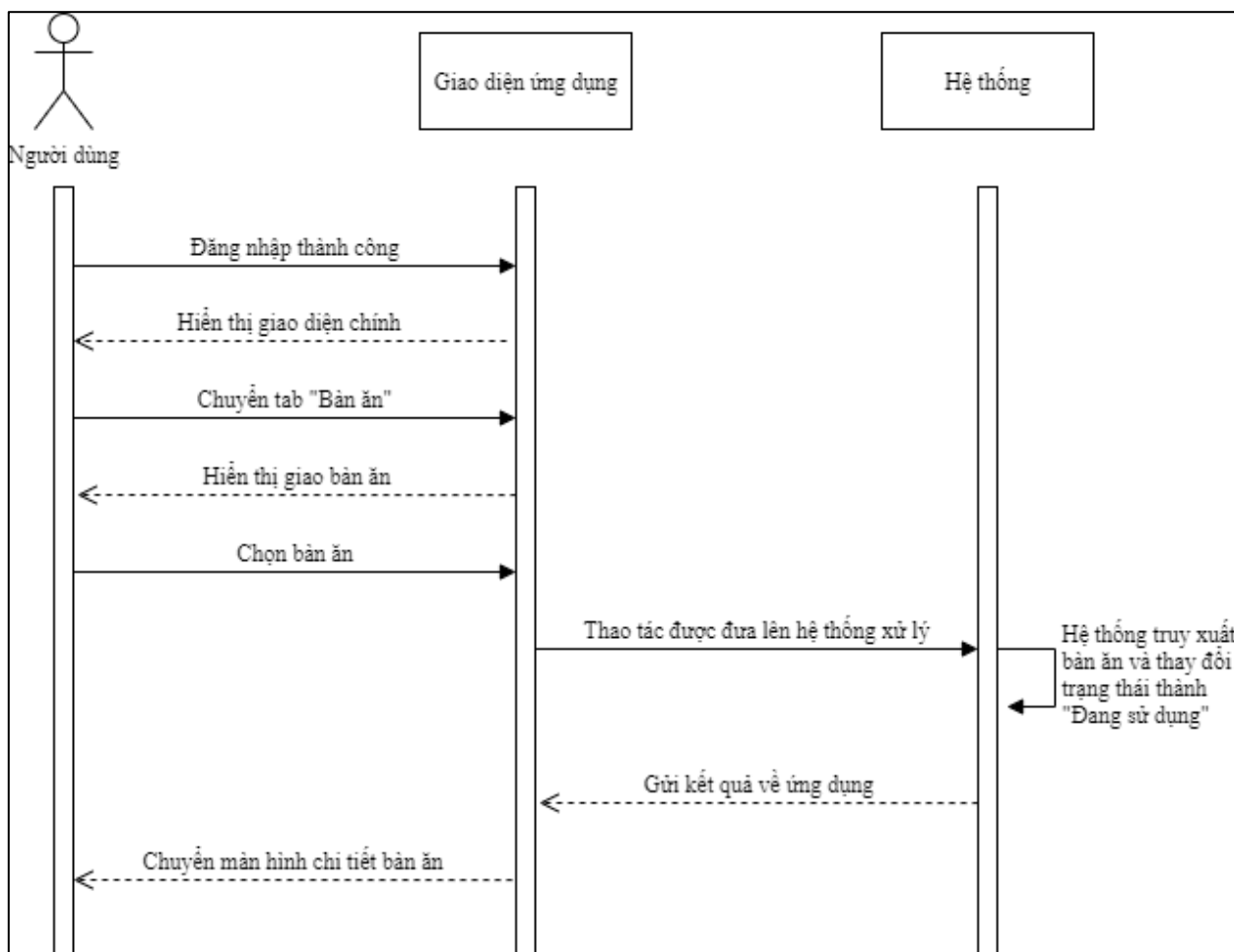
- Người dùng đăng nhập thành công;
- Ứng dụng hiển thị giao diện chính;
- Người dùng chuyển tab “Bàn ăn”;
- Ứng dụng hiển thị giao diện bàn ăn;
- Người dùng chọn bàn ăn;
- Thao tác được đưa lên hệ thống xử lý;
- Hệ thống truy xuất bàn ăn và thay đổi trạng thái bàn thành “Đang sử dụng”
- Hệ thống gửi kết quả về ứng dụng;
- Ứng dụng chuyển màn hình chi tiết bàn ăn (Chi tiết đơn hàng).

Sơ đồ giữa tác nhân và use case:



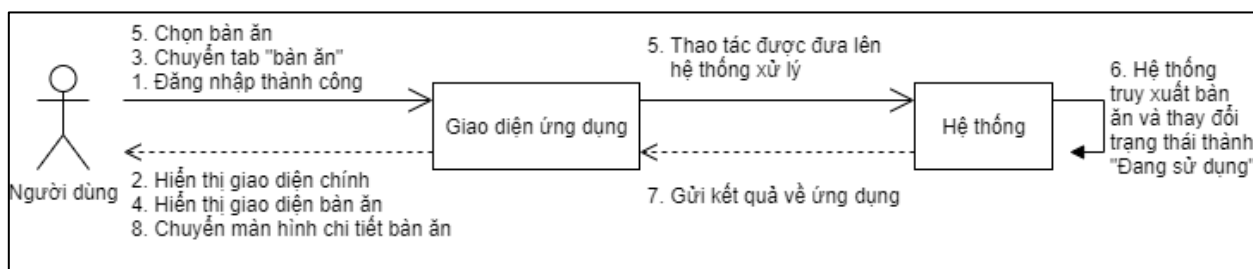
Sơ đồ 30 Sơ đồ tác nhân - usecase chức năng truy cập bàn ăn

Sơ đồ tuần tự:



Sơ đồ 31 Sơ đồ tuần tự chức năng truy cập bàn ăn

Sơ đồ hợp tác:



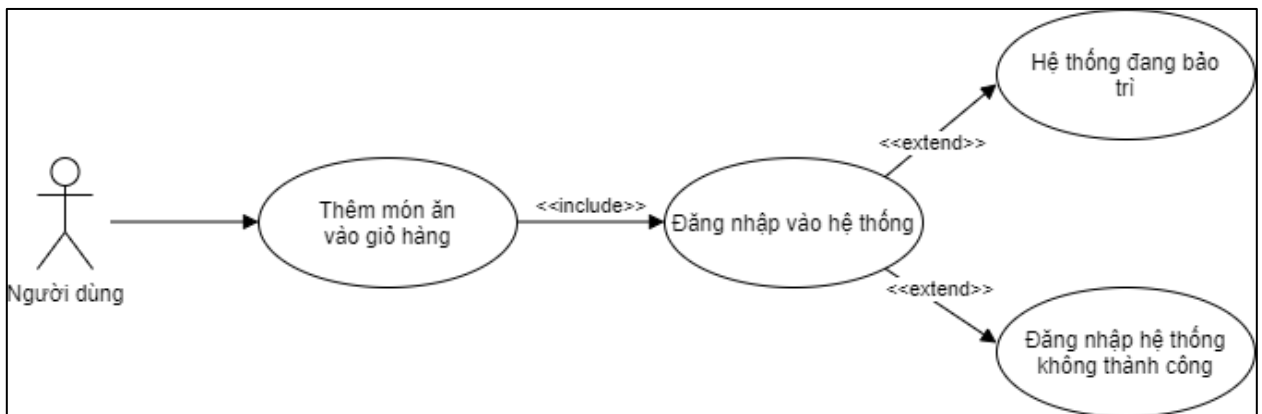
Sơ đồ 32 Sơ đồ hợp tác chức năng truy cập bàn ăn

2.2.3.3. Chức năng thêm món ăn vào giỏ hàng:

Sau khi truy cập một bàn ăn thành công. Người dùng nhấn vào nút “Order” để thực hiện quy trình đặt món. Ứng dụng hiển thị giao diện thực đơn bao gồm các món ăn được phân loại theo loại món ăn. Người dùng chọn món ăn, ứng dụng hiển thị giao diện chi tiết món ăn và các thông tin đặt món bao gồm: Số lượng, biến thể, ghi chú, món ăn gọi thêm,.. Người dùng lựa chọn các thông tin đặt món vào xác nhận. Ứng dụng xác thực thông tin thành công sau đó thêm món ăn vào giỏ hàng, ngược lại thông báo lỗi cho người dùng. Quy trình như sau:

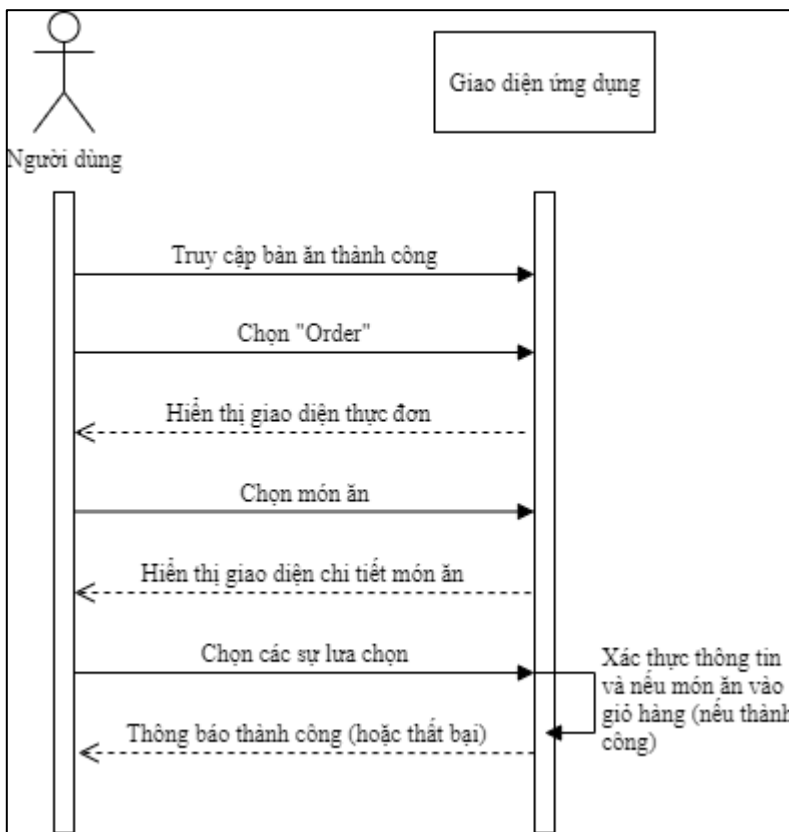
- Người dùng truy cập bàn ăn thành công;
- Người dùng nhấn vào “Order”;
- Ứng dụng hiển thị giao diện thực đơn;
- Người dùng chọn món ăn;
- Ứng dụng hiển thị giao diện chi tiết món ăn;
- Người dùng chọn các sự lựa chọn và xác nhận;
- Ứng dụng xác thực thông và lưu món ăn vào giỏ hàng (nếu thành công);
- Ứng dụng thông báo thành công (hoặc thất bại).

Sơ đồ giữa tác nhân và use case:



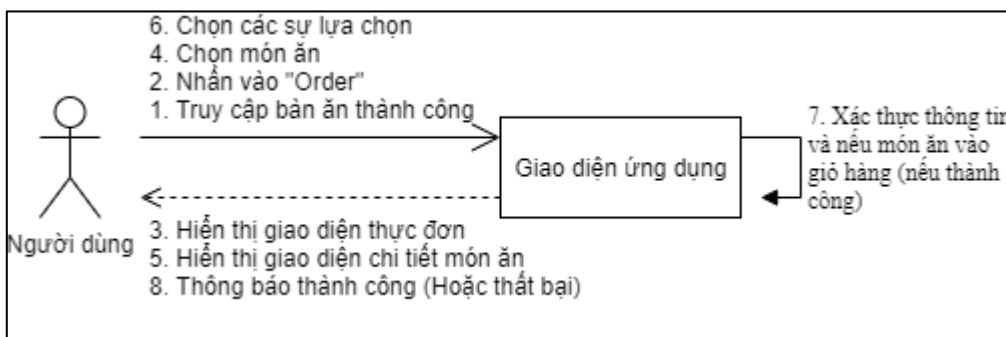
Sơ đồ 33 Sơ đồ tác nhân - usecase chức năng thêm vào giỏ hàng

Sơ đồ tuần tự:



Sơ đồ 34 Sơ đồ tuần tự chức năng thêm vào giỏ hàng

Sơ đồ hợp tác:



Sơ đồ 35 Sơ đồ hợp tác chức năng thêm vào giỏ hàng

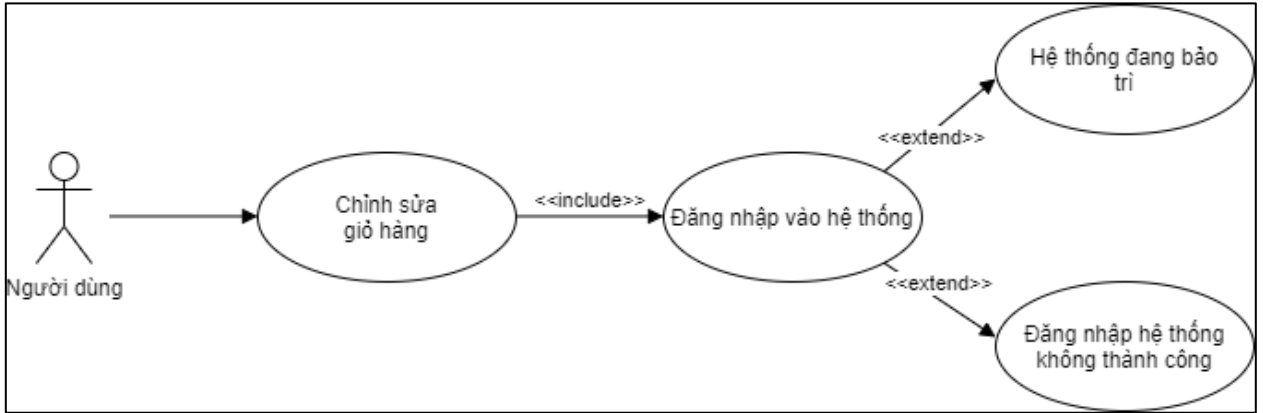
2.2.3.4. Chức năng chỉnh sửa giỏ hàng:

Sau khi người dùng thêm món ăn vào giỏ hàng thành công, người dùng trở về giao diện chi tiết đơn hàng. Nhấn “Giỏ hàng” để xem giỏ hàng. Những món ăn trong giỏ hàng được hiển thị theo danh sách kèm theo nút “Xóa” bên cạnh. Người dùng nhấn “Xóa” để xóa món ăn khỏi giỏ hàng. Quy trình như sau:

- Người dùng thêm món ăn vào giỏ hàng thành công;
- Người dùng chọn “Giỏ hàng”;
- Ứng dụng hiển thị giao diện giỏ hàng;

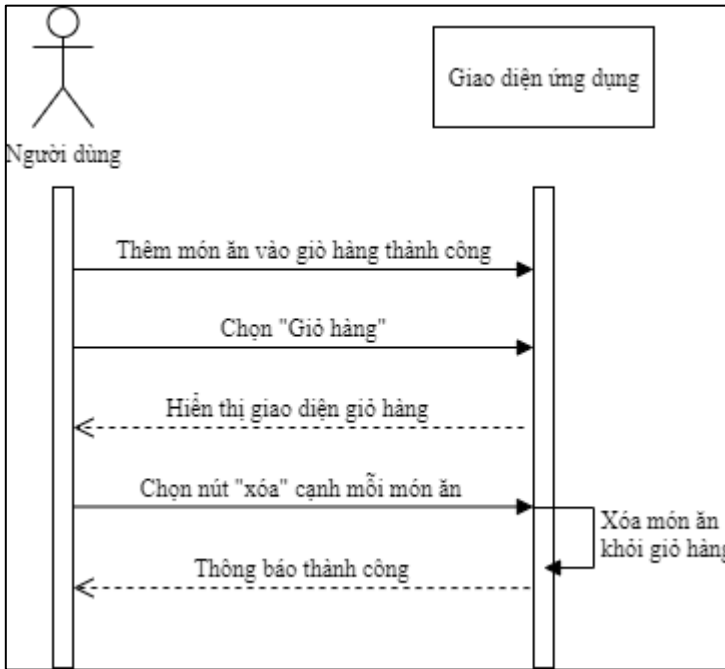
- Người dùng nhấn “Xóa” cạnh mỗi món ăn;
- Ứng dụng xóa món ăn khỏi giỏ hàng;
- Ứng dụng thông báo thành công.

Sơ đồ giữa tác nhân và use case:



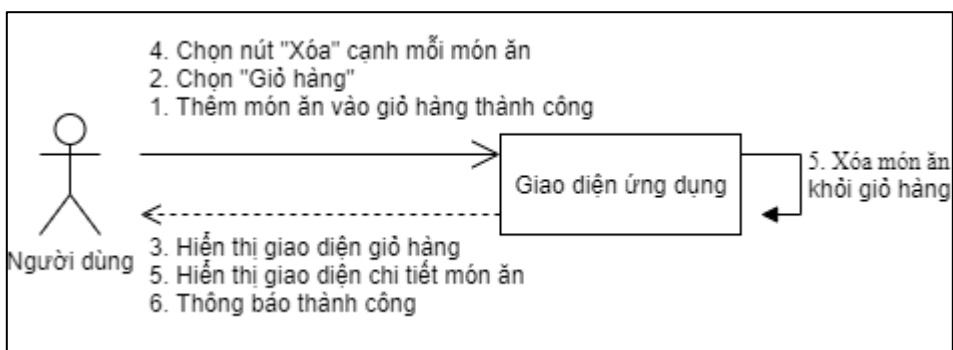
Sơ đồ 36 Sơ đồ tác nhân - usecase chức năng chỉnh sửa giỏ hàng

Sơ đồ tuần tự:



Sơ đồ 37 Sơ đồ tuần tự chức năng chỉnh sửa giỏ hàng

Sơ đồ hợp tác:



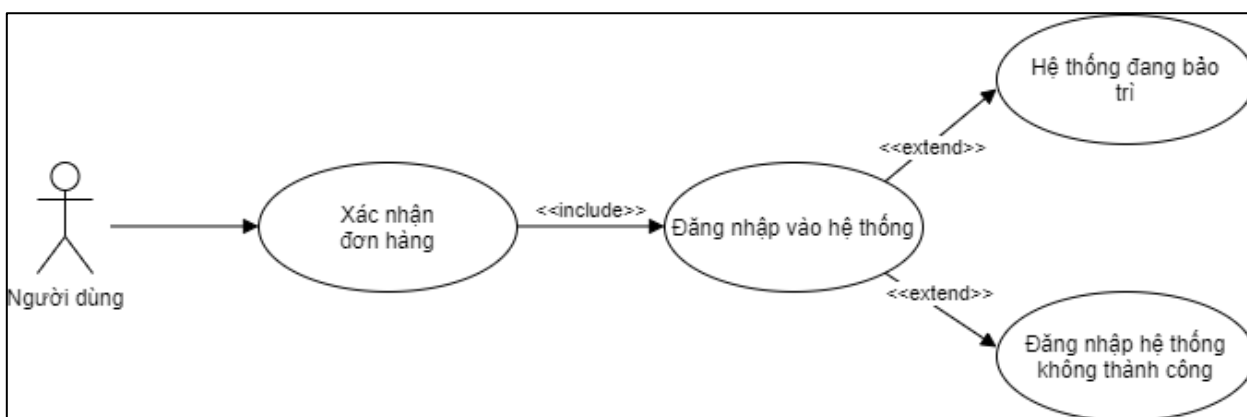
Sơ đồ 38 Sơ đồ hợp tác chức năng chỉnh sửa giỏ hàng

2.2.3.5. Chức năng xác nhận đơn hàng

Sau khi thêm món ăn vào giỏ hàng thành công, người dùng chọn “Giỏ hàng” và nhấn “Xác nhận” để ứng dụng gửi yêu cầu lên hệ thống thêm món ăn vào đơn hàng. Quy trình như sau:

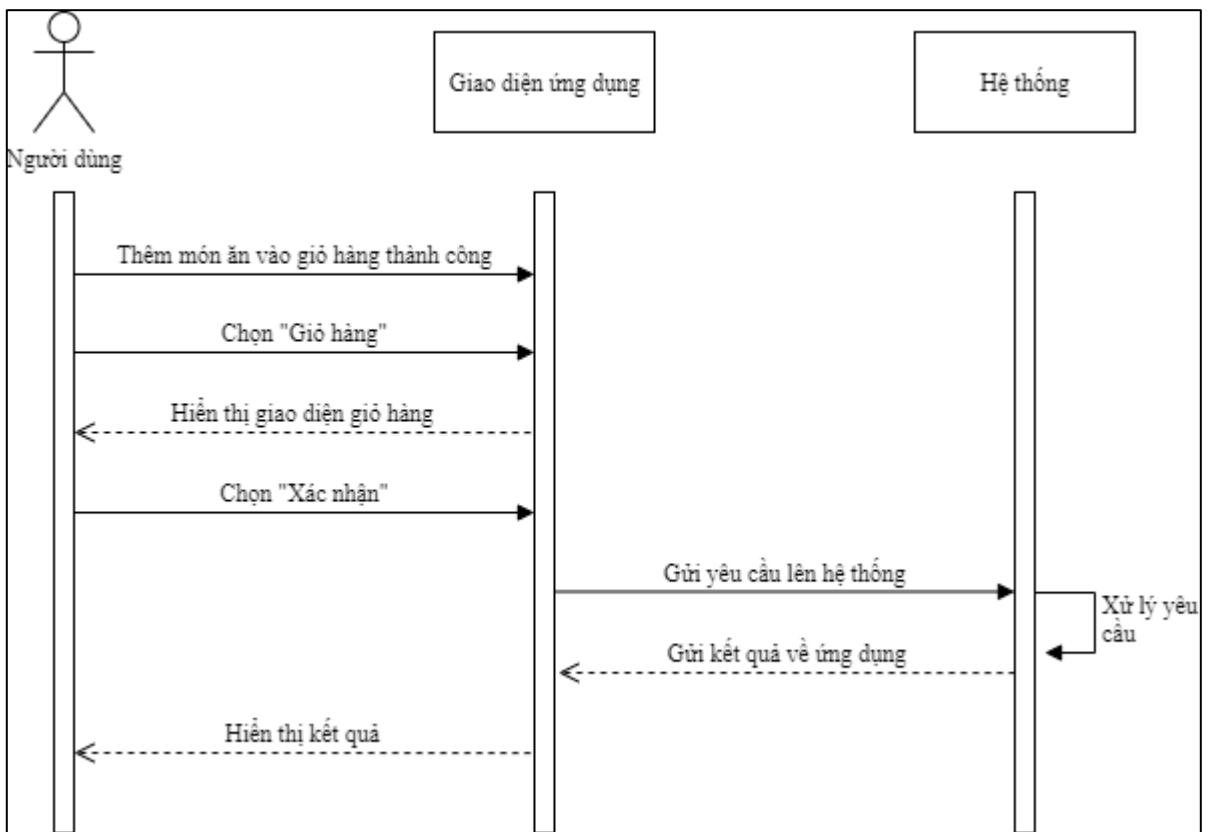
- Người dùng thêm món ăn vào giỏ hàng thành công;
- Người dùng chọn “Giỏ hàng”;
- Ứng dụng hiển thị giao diện giỏ hàng;
- Người dùng nhấn “Xác nhận”;
- Ứng dụng gửi yêu cầu lên hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về ứng dụng;
- Ứng dụng thông báo kết quả.

Sơ đồ giữa tác nhân và use case:



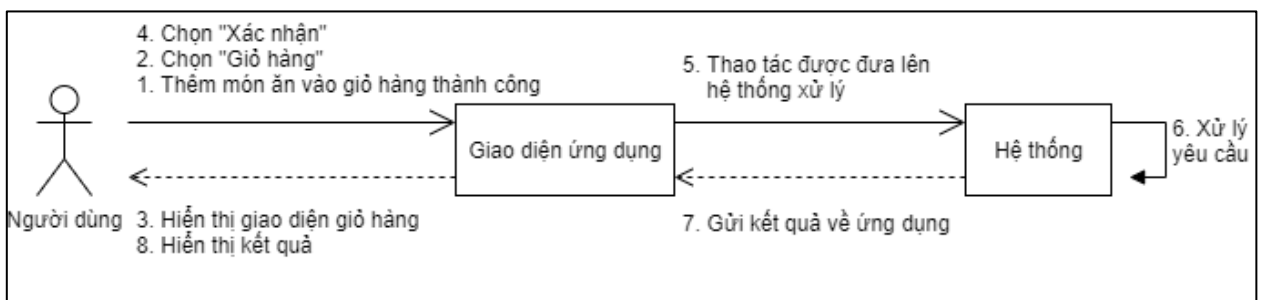
Sơ đồ 39 Sơ đồ tác nhân - usecase xác nhận đơn hàng

Sơ đồ tuần tự:



Sơ đồ 40 Sơ đồ tuần tự chức năng xác nhận đơn hàng

Sơ đồ hợp tác:



Sơ đồ 41 Sơ đồ hợp tác chức năng xác nhận giỏ hàng

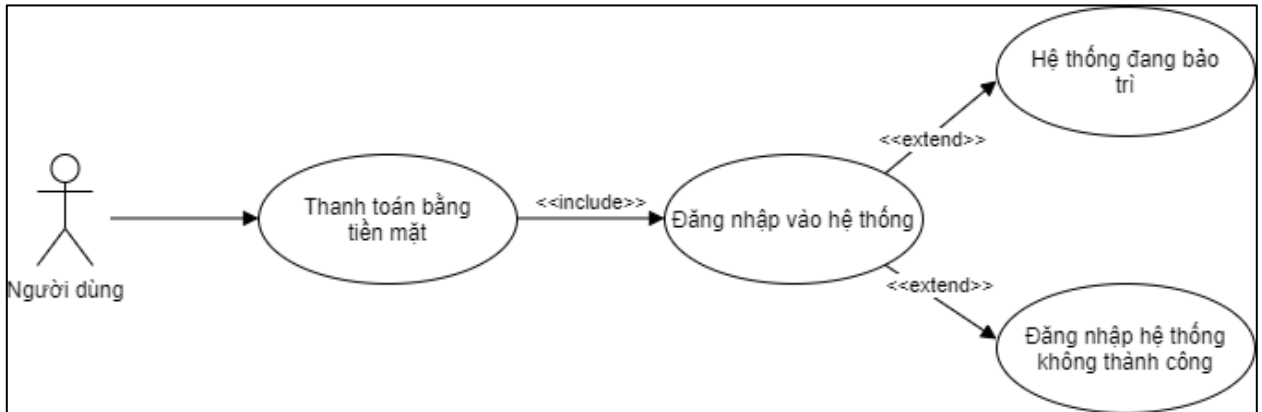
2.2.3.6. Chức năng thanh toán bằng tiền mặt:

Sau khi người dùng truy cập bàn ăn thành công, nhấn “Thanh toán”, sau đó ứng dụng điều hướng người dùng đến giao diện thanh toán. Người dùng chọn phương thức thanh toán bằng tiền mặt và nhập số tiền rồi xác nhận. Ứng dụng gửi yêu cầu xử lý lên hệ thống và gửi về kết quả. Nếu thành công chuyển trạng thái đơn hàng về “Paid” (Đã thanh toán) và trạng thái bàn ăn về “Ready” (Có sẵn). Quy trình như sau:

- Người dùng truy cập bàn ăn thành công;
- Người dùng chọn “Thanh toán”;
- Ứng dụng hiển thị giao diện thanh toán;

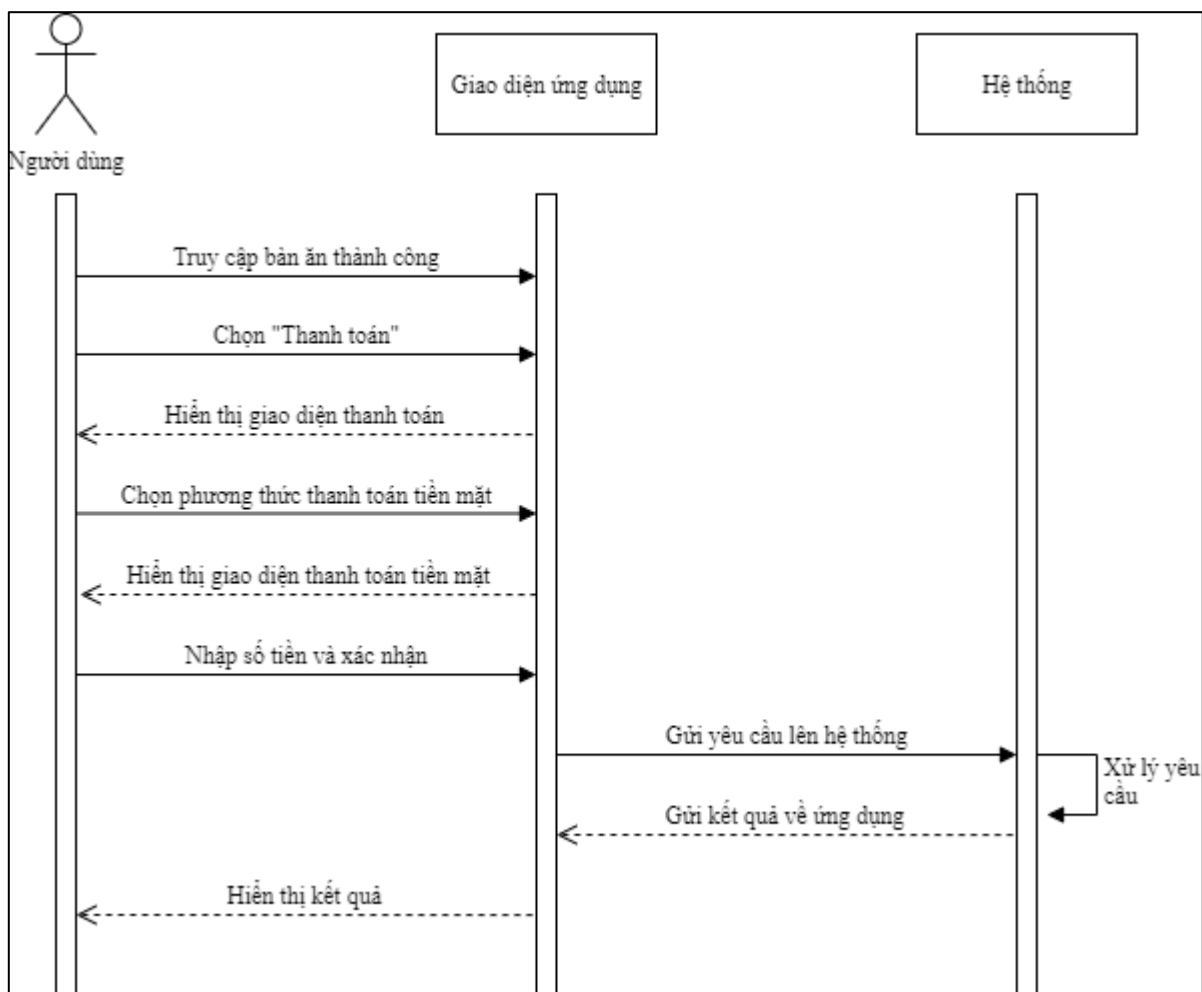
- Người dùng chọn phương thức thanh toán tiền mặt;
- Ứng dụng hiển thị giao diện thanh toán tiền mặt;
- Người dùng nhập số tiền và xác nhận;
- Ứng dụng gửi yêu cầu lên hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về ứng dụng;
- Ứng dụng hiển thị kết quả.

Sơ đồ giữa tác nhân và use case:



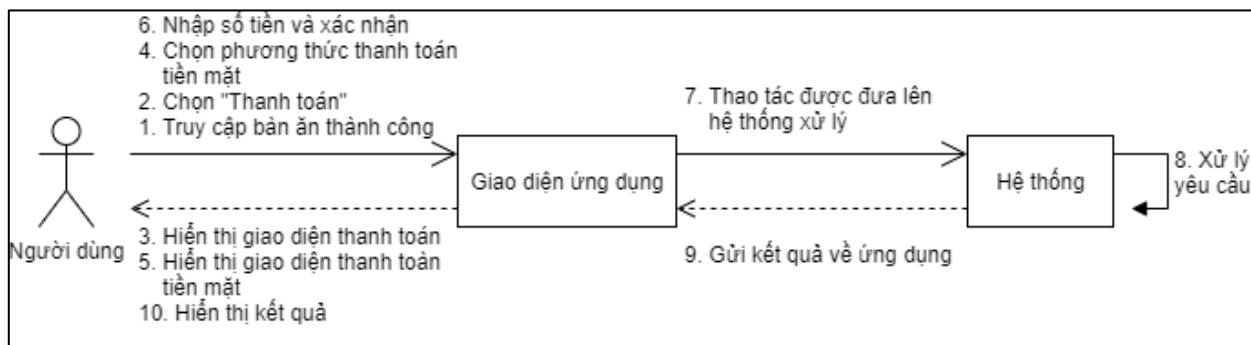
Sơ đồ 42 Sơ đồ tác nhân - usecase chức năng thanh toán tiền mặt

Sơ đồ tuần tự:



Sơ đồ 43 Sơ đồ tuần tự chức năng thanh toán tiền mặt

Sơ đồ hợp tác:



Sơ đồ 44 Sơ đồ hợp tác chức năng thanh toán tiền mặt

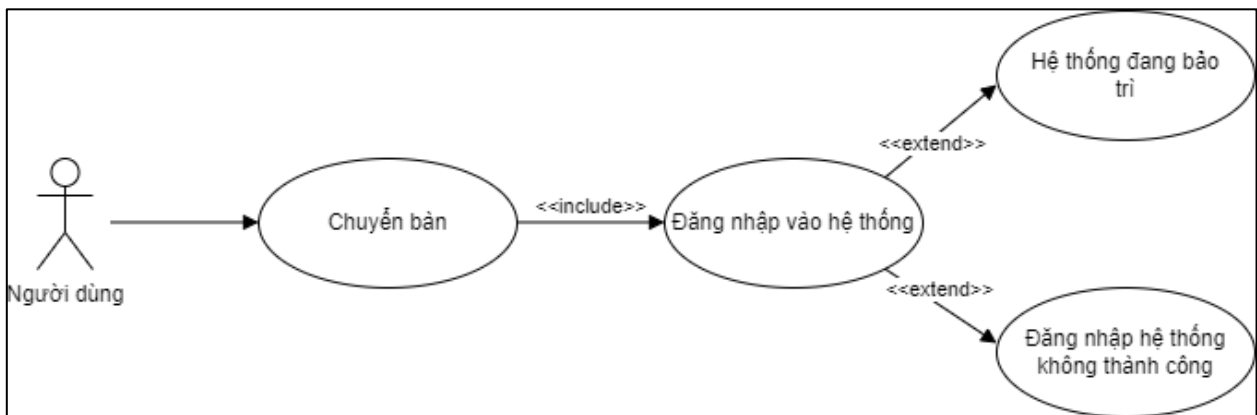
2.2.3.7. Chức năng chuyển bàn:

Tại tab “Bàn ăn” ứng dụng hiển thị danh sách bàn ăn. Người dùng nhấn giữ bàn ăn và ứng dụng hiển thị các sự lựa chọn như “Mở”, “Thanh toán”, “Hủy” và “Chuyển bàn”. Người dùng chọn “Chuyển bàn”, ứng dụng hiển thị giao diện chuyển bàn với các thông tin bao gồm bàn đích (Bàn ăn được chuyển đến) và bàn đích này được hiển

thị theo danh sách các bàn ăn còn trống. Người dùng chọn một bàn ăn đích và xác nhận, ứng dụng gửi yêu cầu lên hệ thống và trả về kết quả. Quy trình như sau:

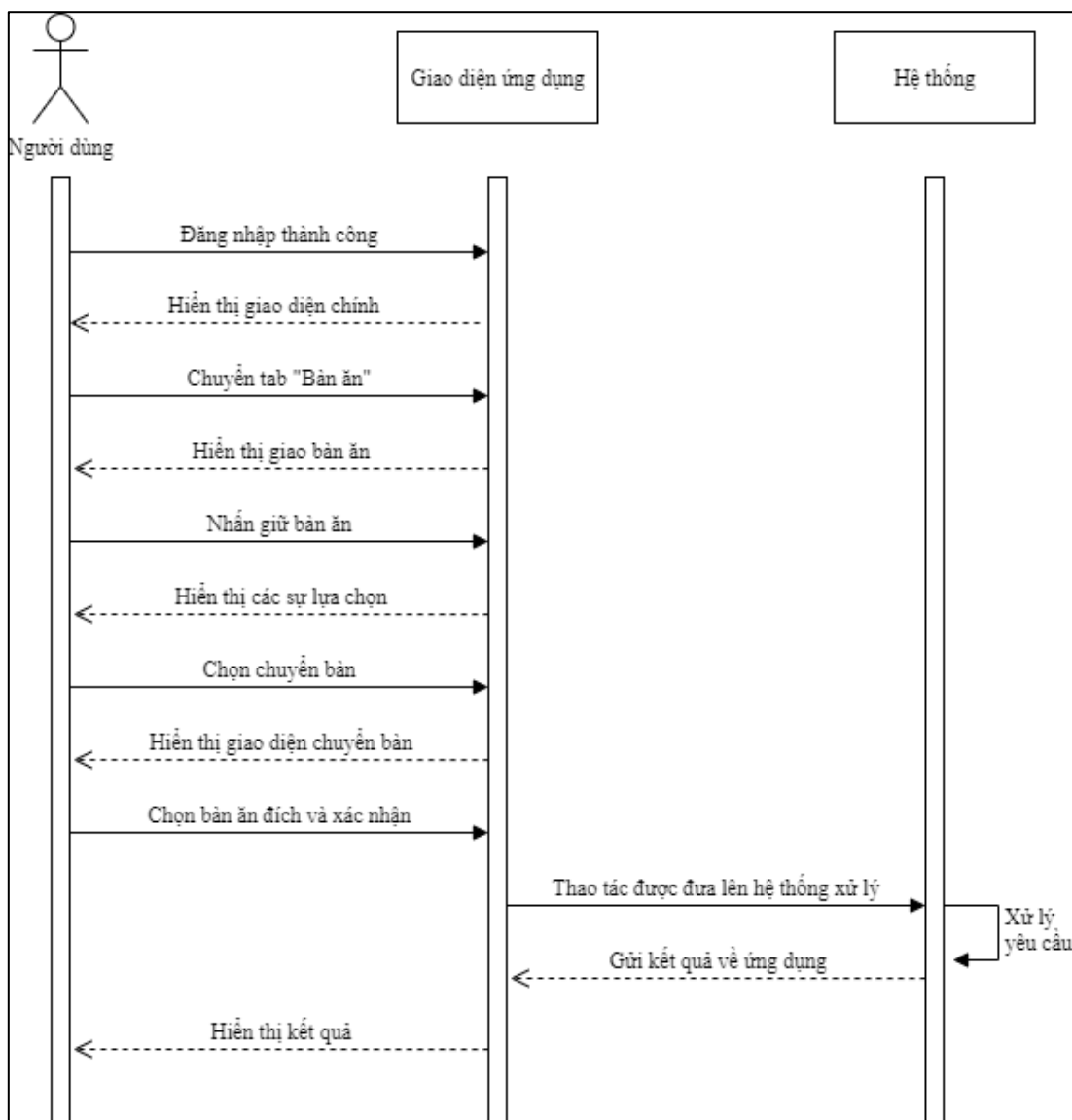
- Người dùng đăng nhập thành công;
- Ứng dụng hiển thị giao diện chính;
- Người dùng chuyển tab “Bàn ăn”;
- Người dùng nhấn giữ bàn ăn;
- Ứng dụng hiển thị các sử lựa chọn;
- Người dùng chọn chuyển bàn;
- Ứng dụng hiển thị giao diện chuyển bàn;
- Người dùng chọn bàn ăn đích và xác nhận;
- Ứng dụng gửi yêu cầu lên hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về ứng dụng;
- Ứng dụng hiển thị kết quả.

Sơ đồ giữa tác nhân và use case:



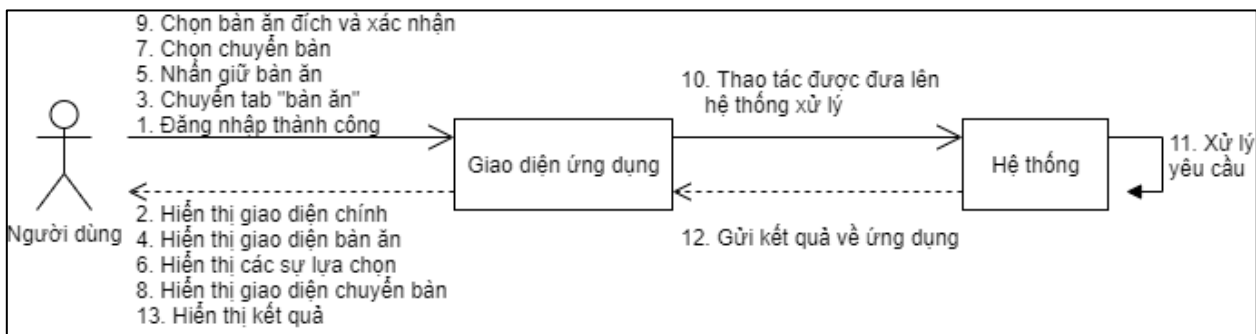
Sơ đồ 45 Sơ đồ tác nhân - usecase chức năng chuyển bàn

Sơ đồ tuần tự:



Sơ đồ 46 Sơ đồ tuần tự chức năng chuyển bàn

Sơ đồ hợp tác:



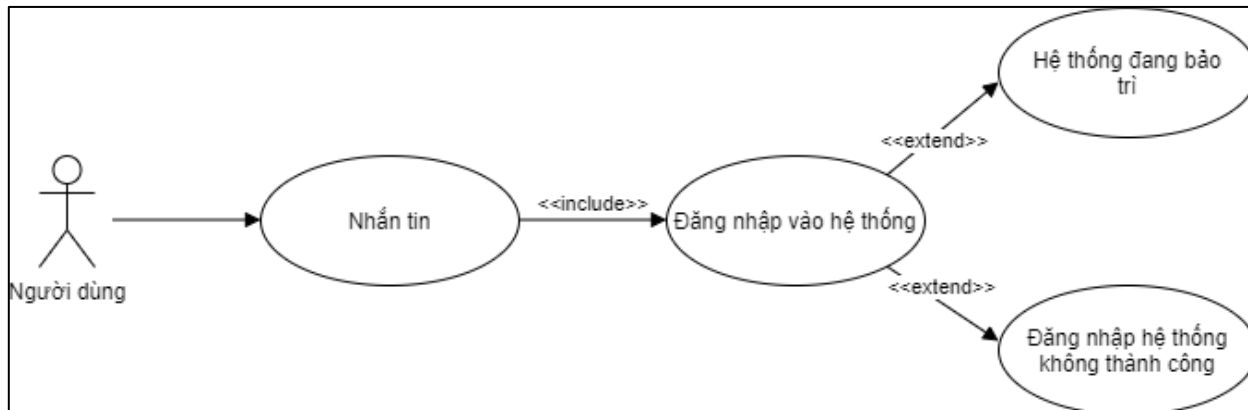
Sơ đồ 47 Sơ đồ hợp tác chức năng chuyển bàn

2.2.3.8. Chức năng nhắn tin:

Để thao tác với chức năng nhắn tin, người dùng chuyển tab “Nhắn tin” tại màn hình chính. Ứng dụng hiển thị giao diện nhắn tin với danh sách người dùng khác và danh sách người dùng vừa nhắn tin. Người dùng chọn một người dùng khác để liên lạc. Ứng dụng hiển thị giao diện chi tiết tin nhắn với thông tin các tin nhắn đã nhận và gửi. Người dùng gửi tin nhắn với thao tác nhập tin nhắn và nhấn gửi. Ứng dụng gửi tin nhắn đến người dùng khác thông qua hệ thống. Quy trình như sau:

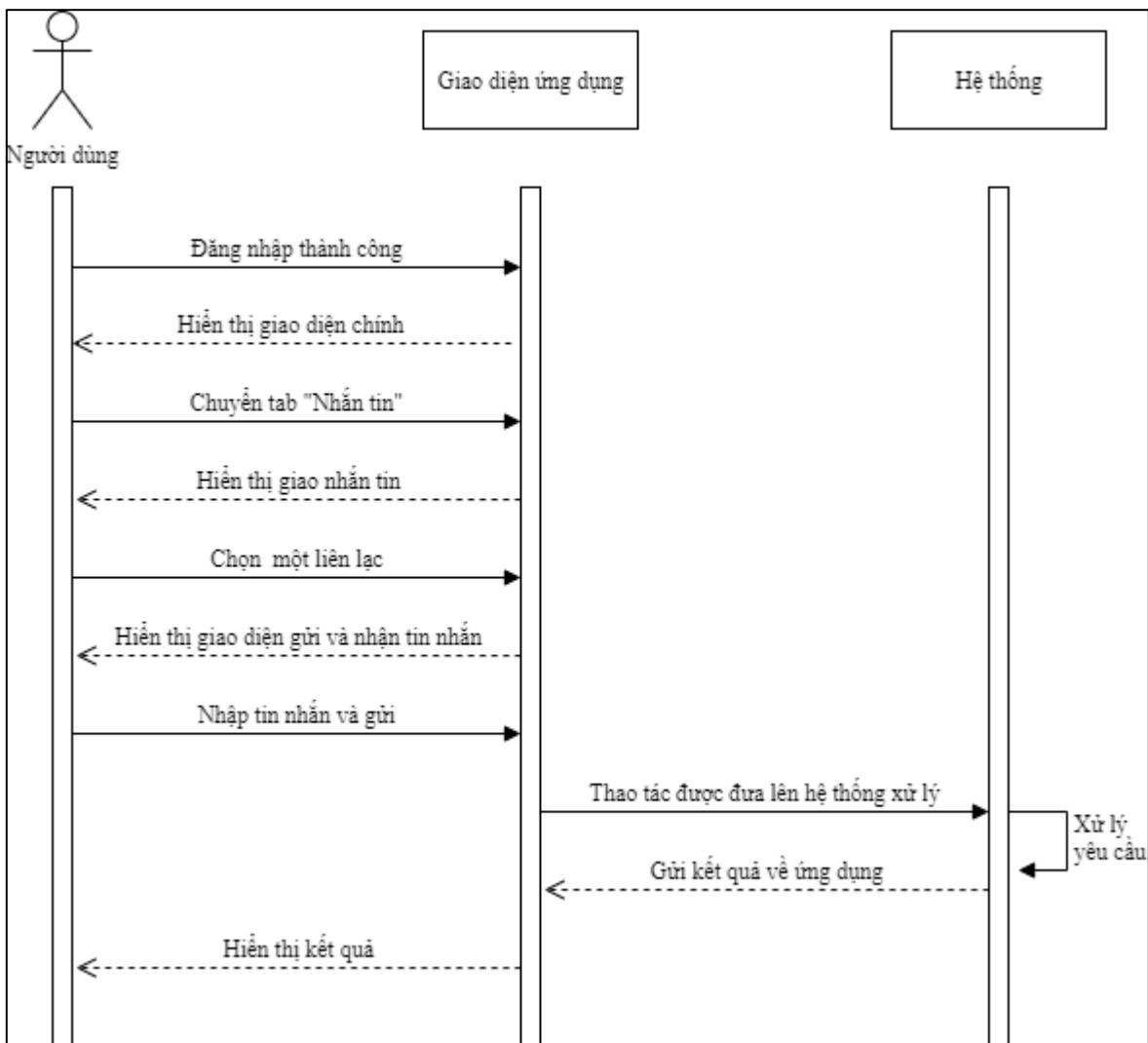
- Người dùng đăng nhập thành công;
- Ứng dụng hiển thị giao diện chính;
- Người dùng chuyển tab “Nhắn tin”;
- Ứng dụng hiển thị giao diện nhắn tin;
- Người dùng chọn một liên lạc;
- Ứng dụng hiển thị giao diện gửi và nhận tin nhắn;
- Người dùng nhập tin nhắn và gửi;
- Ứng dụng gửi yêu cầu lên hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về ứng dụng;
- Ứng dụng hiển thị kết quả.

Sơ đồ giữa tác nhân và use case:



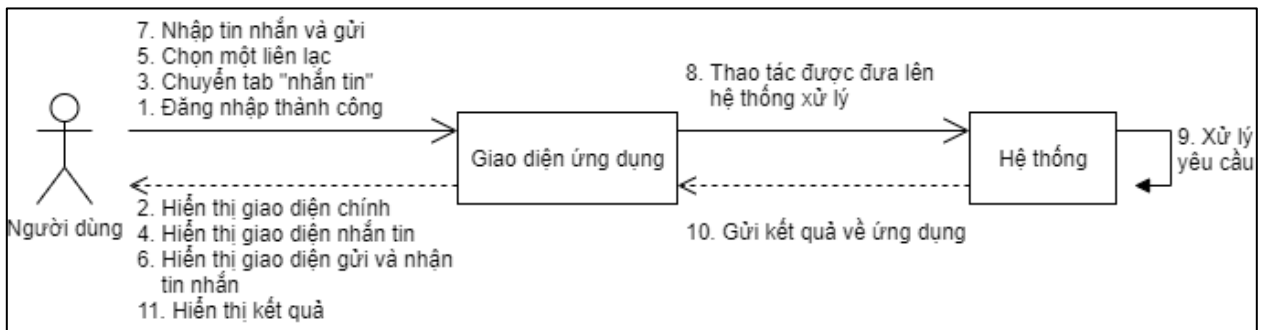
Sơ đồ 48 Sơ đồ tác nhân - usecase chức năng nhắn tin

Sơ đồ tuần tự:



Sơ đồ 49 Sơ đồ tuần tự chức năng nhắn tin

Sơ đồ hợp tác:



Sơ đồ 50 Sơ đồ hợp tác chức năng nhắn tin

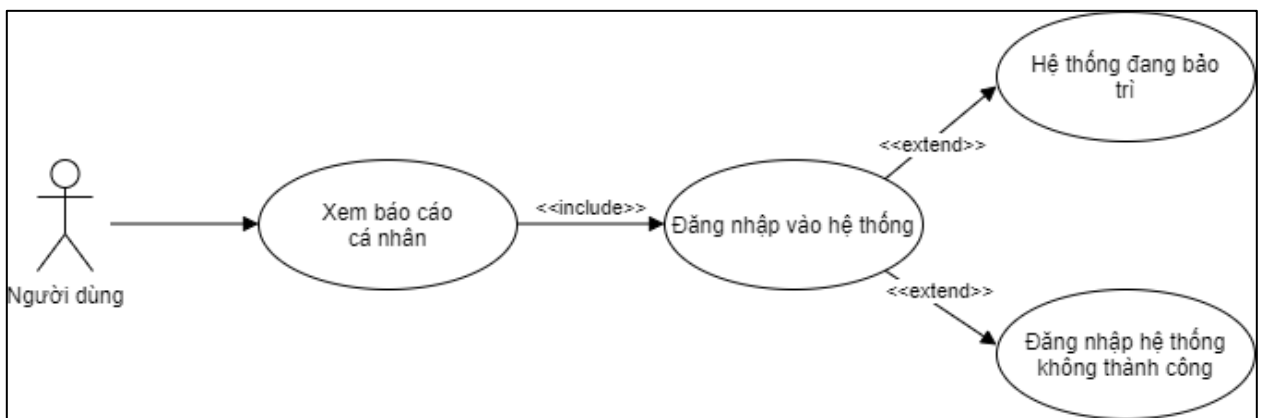
2.2.3.9. Chức năng xem báo cáo cá nhân:

Sau khi đăng nhập thành công, người dùng chuyển tab “Profile” tại giao diện chính. Ứng dụng hiển thị các lựa chọn, người dùng chọn “My Report” để gửi yêu cầu

hiển thị báo cáo cá nhân. Ứng dụng hiển thị giao diện với các thông tin như: doanh thu cá nhân, thống kê món ăn,... Quy trình như sau:

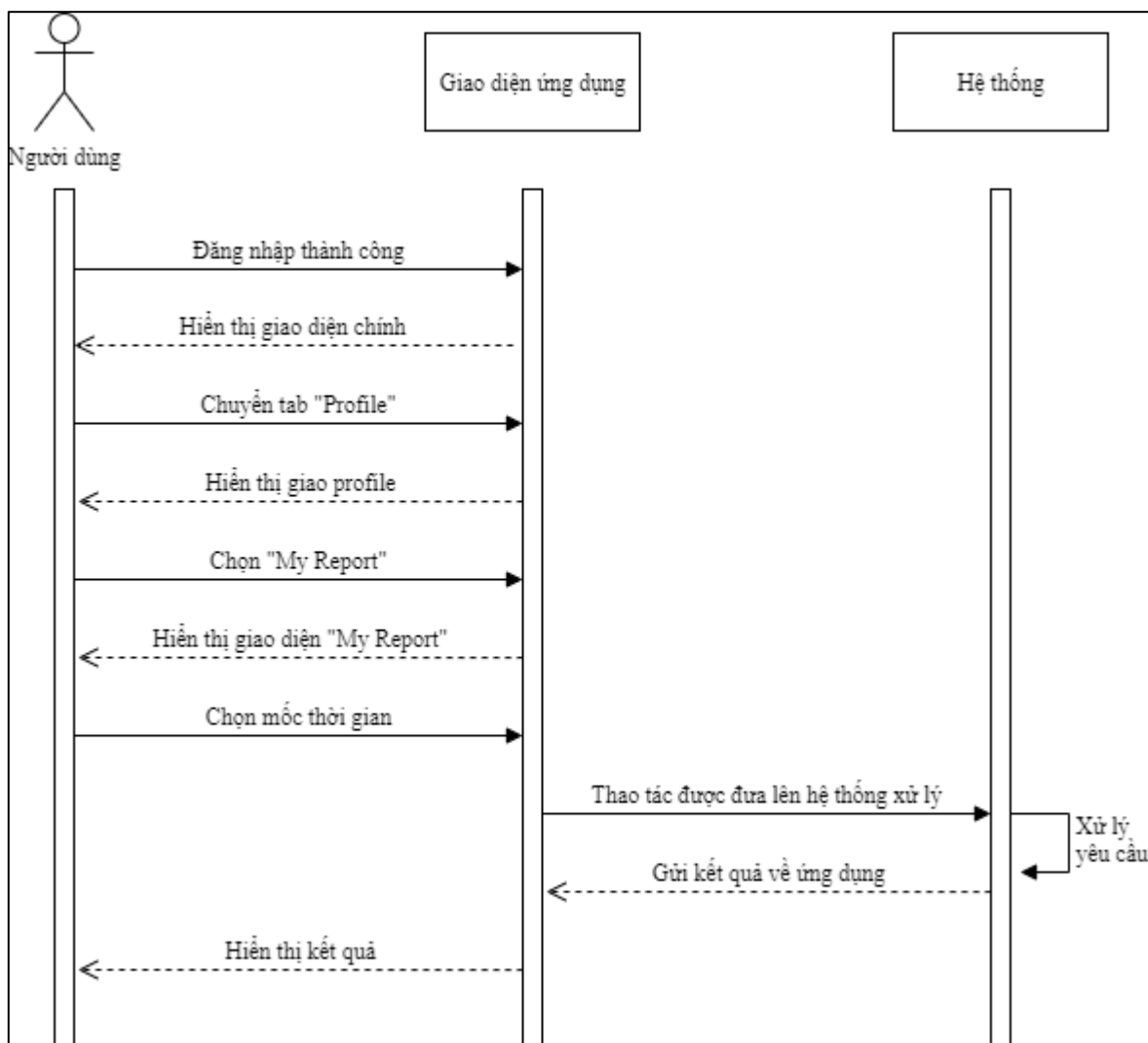
- Người dùng đăng nhập thành công;
- Ứng dụng hiển thị giao diện chính;
- Người dùng chuyển tab “Profile”;
- Ứng dụng hiển thị giao diện profile;
- Người dùng chọn “My Report”;
- Ứng dụng hiển thị giao diện “My Report”;
- Người dùng chọn mốc thời gian;
- Ứng dụng gửi yêu cầu lên hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về ứng dụng;
- Ứng dụng hiển thị kết quả.

Sơ đồ giữa tác nhân và use case:



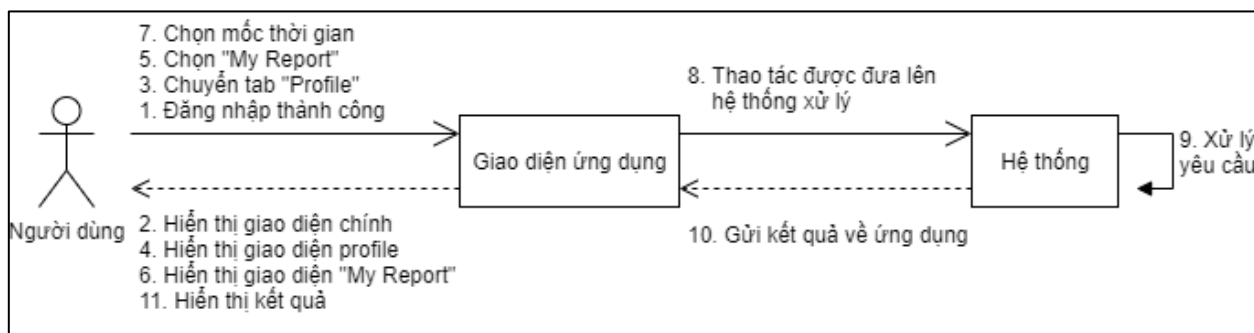
Sơ đồ 51 Sơ đồ tác nhân - usecase chức năng xem báo cáo cá nhân

Sơ đồ tuần tự:



Sơ đồ 52 Sơ đồ tuần tự chức năng xem báo cáo cá nhân

Sơ đồ hợp tác:



Sơ đồ 53 Sơ đồ hợp tác chức năng xem báo cáo cá nhân

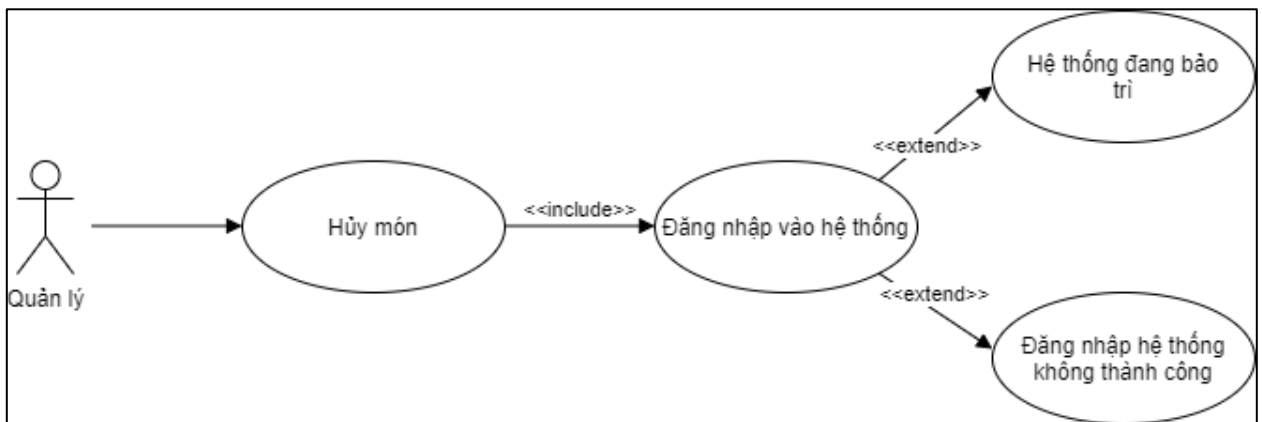
2.2.3.10. Chức năng hủy món:

Trong quá trình phục vụ, khi thực khách yêu cầu hủy món ăn. Người quản lý truy cập vào một bàn ăn vào chọn món ăn muốn hủy. Người quản lý nhấn “Hủy” và

ứng dụng gửi yêu cầu lên hệ thống xử lý và gửi kết quả về người quản lý. Quy trình như sau:

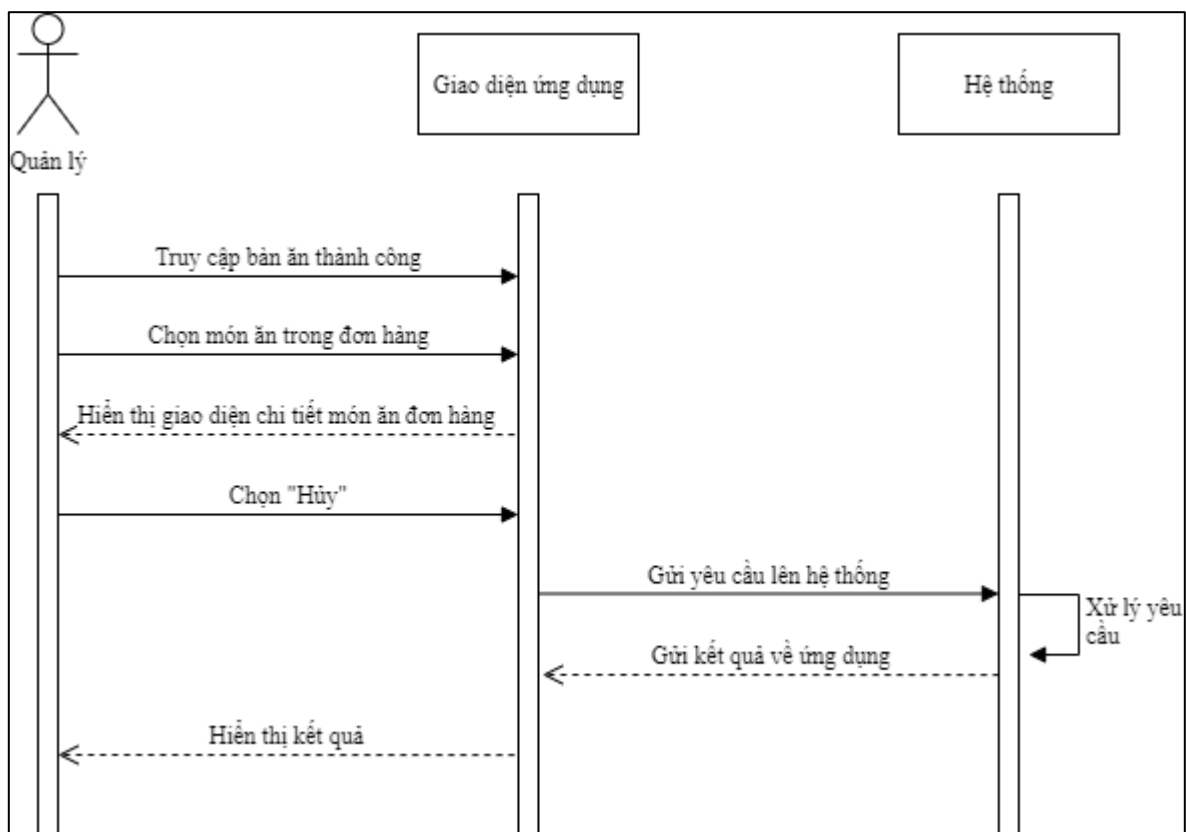
- Quản lý truy cập bàn ăn thành công;
- Quản lý chọn món ăn trong đơn hàng;
- Ứng dụng hiển thị giao diện chi tiết món ăn đơn hàng;
- Quản lý chọn “Hủy”;
- Ứng dụng gửi yêu cầu lên hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về ứng dụng;
- Ứng dụng hiển thị kết quả.

Sơ đồ giữa tác nhân và use case:



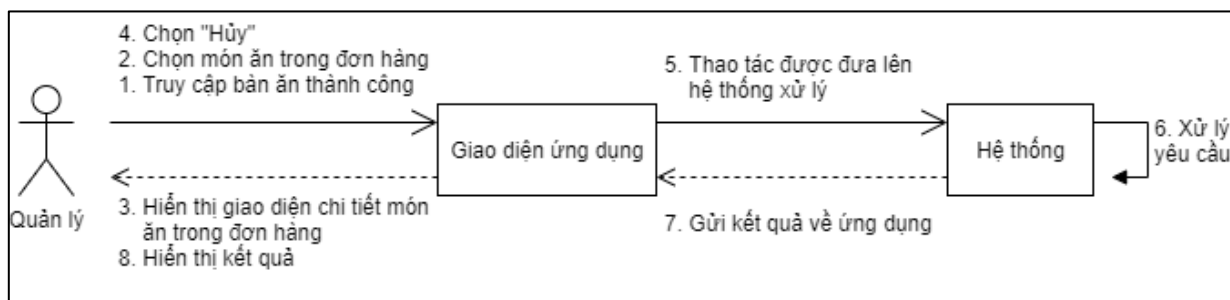
Sơ đồ 54 Sơ đồ tác nhân - usecase chức năng hủy món

Sơ đồ tuần tự:



Sơ đồ 55 Sơ đồ tuần tự chức năng hủy món

Sơ đồ hợp tác:



Sơ đồ 56 Sơ đồ hợp tác chức năng hủy món

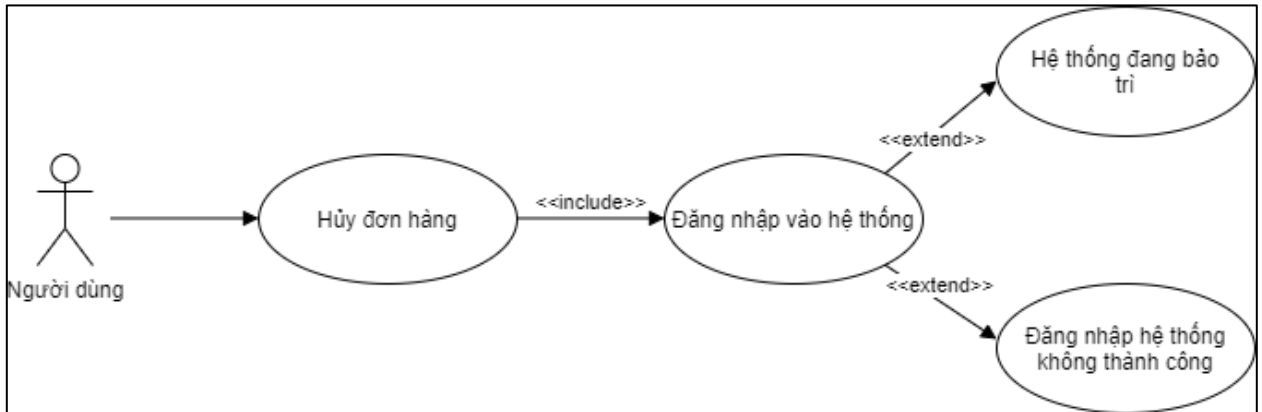
2.2.3.11. Chức năng hủy đơn hàng:

Trong quá trình phục vụ, khi có vấn đề dẫn đến yêu cầu hủy đơn hàng. Người quản lý truy cập vào tab “Bàn ăn” và nhấn giữ bàn ăn. Quản lý chọn “Hủy” để thực hiện tác vụ hủy đơn hàng. Ứng dụng gửi yêu cầu lên hệ thống và trả về kết quả. Quy trình như sau:

- Quản lý đăng nhập thành công;
- Ứng dụng hiển thị giao diện chính;
- Quản lý chuyển tab “Bàn ăn”;
- Ứng dụng hiển thị giao diện bàn ăn;

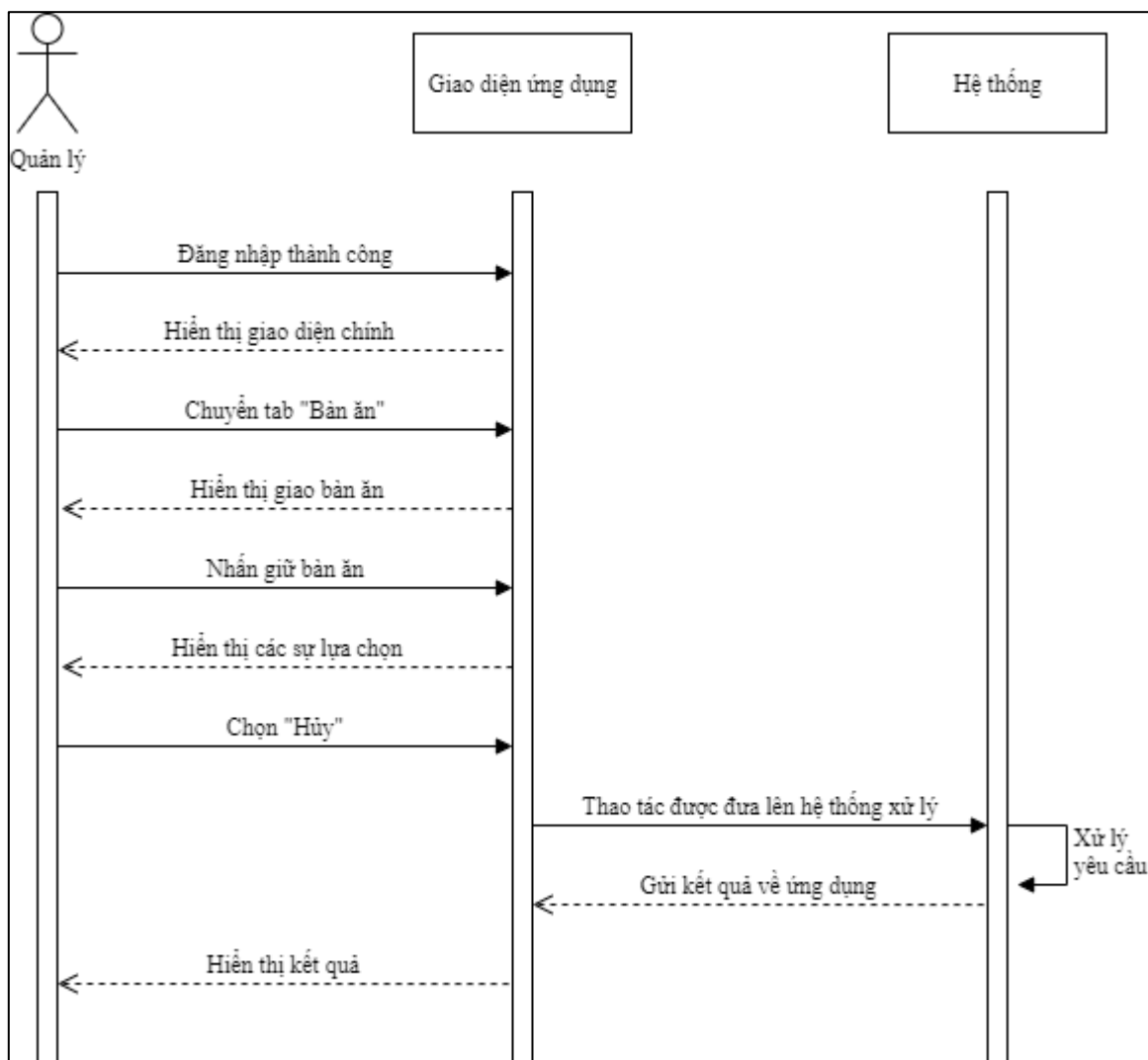
- Quản lý nhân giữ bàn ăn;
- Ứng dụng hiển thị các sự lựa chọn;
- Quản lý chọn “Hủy”;
- Ứng dụng gửi yêu cầu lên hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về ứng dụng;
- Ứng dụng hiển thị kết quả.

Sơ đồ giữa tác nhân và use case:



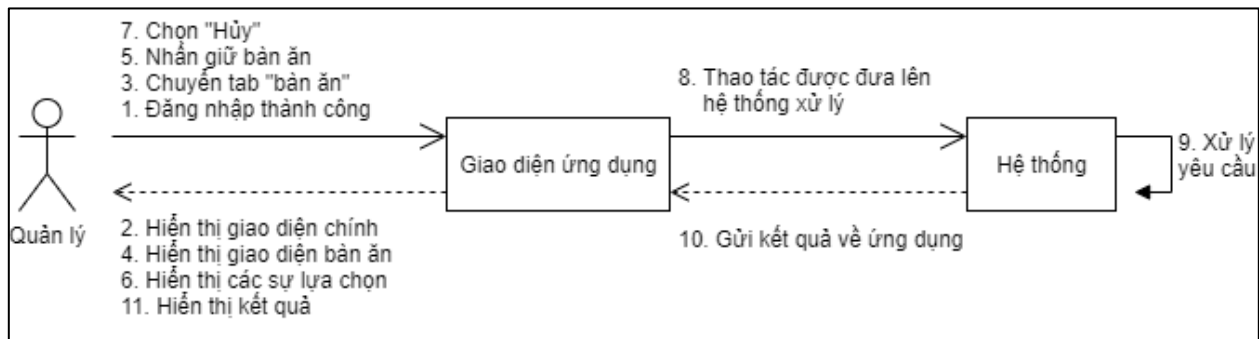
Sơ đồ 57 Sơ đồ tác nhân - usecase chức năng hủy đơn hàng

Sơ đồ tuần tự:



Sơ đồ 58 Sơ đồ tuần tự chức năng hủy đơn hàng

Sơ đồ hợp tác:



Sơ đồ 59 Sơ đồ hợp tác chức năng hủy đơn hàng

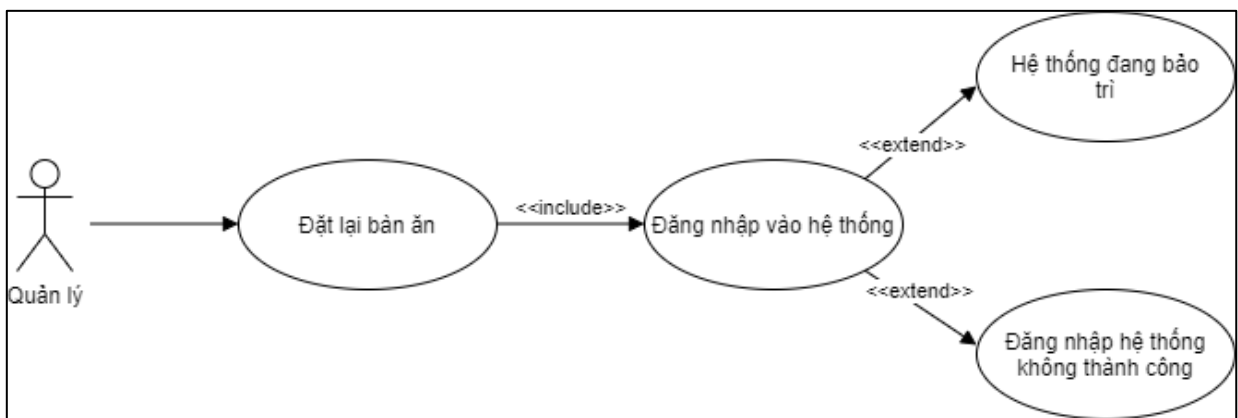
2.2.3.12. Chức năng đặt lại bàn ăn:

Để xử lý vấn đề khi đang truy cập bàn ăn người dùng đột ngột thoát ứng dụng. Ứng dụng cho phép thao tác đặt lại bàn ăn (Tức bỏ trạng thái In Use). Người quản lý

nhấn giữ bàn ăn và chọn “Đặt lại”, ứng dụng gửi yêu cầu lên hệ thống và gửi kết quả về. Quy trình như sau:

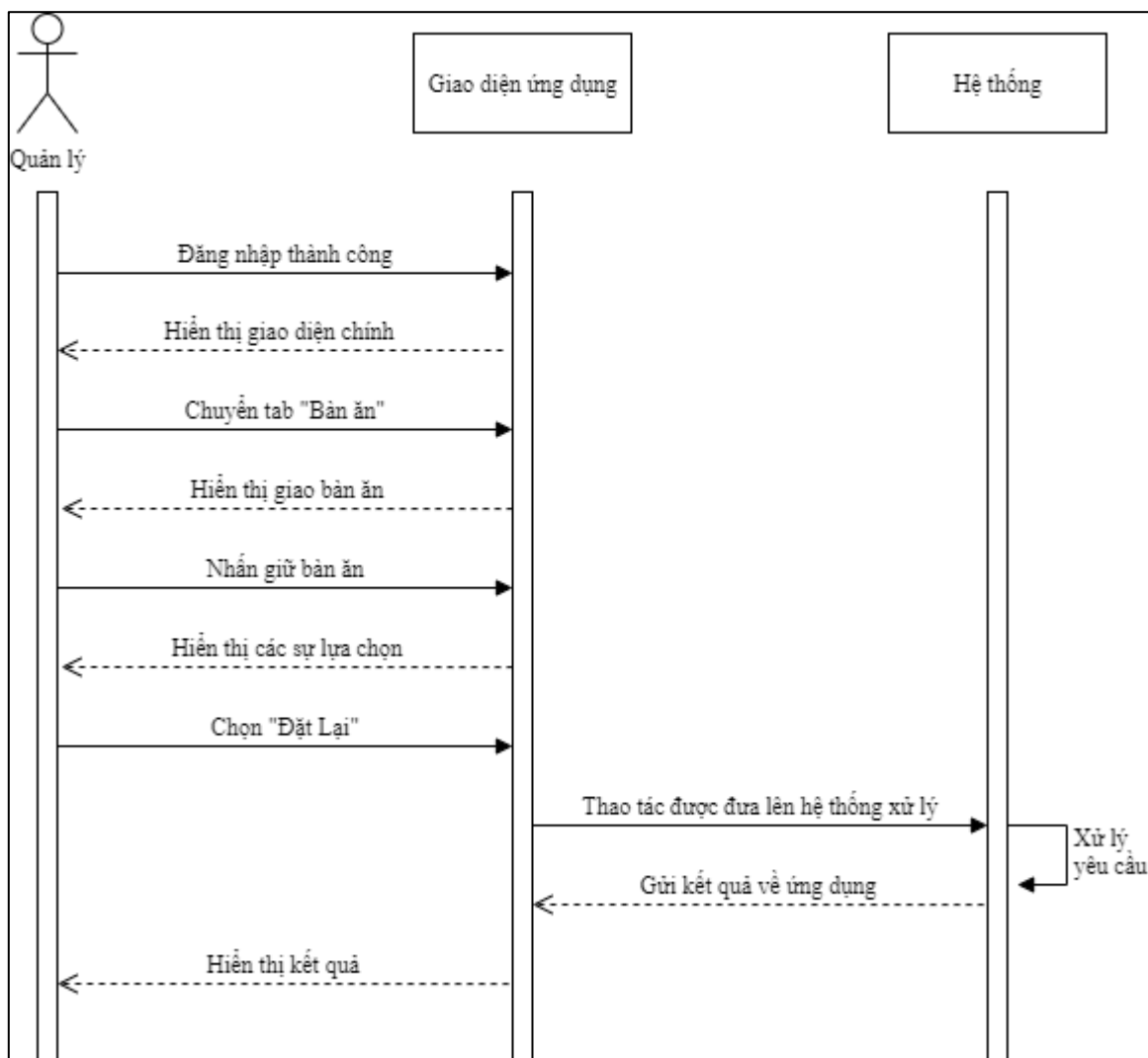
- Quản lý đăng nhập thành công;
- Ứng dụng hiển thị giao diện chính;
- Quản lý chuyển tab “Bàn ăn”;
- Ứng dụng hiển thị giao diện bàn ăn;
- Quản lý nhấn giữ bàn ăn;
- Ứng dụng hiển thị các sự lựa chọn;
- Quản lý chọn “Đặt lại”;
- Ứng dụng gửi yêu cầu lên hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về ứng dụng;
- Ứng dụng hiển thị kết quả.

Sơ đồ giữa tác nhân và use case:



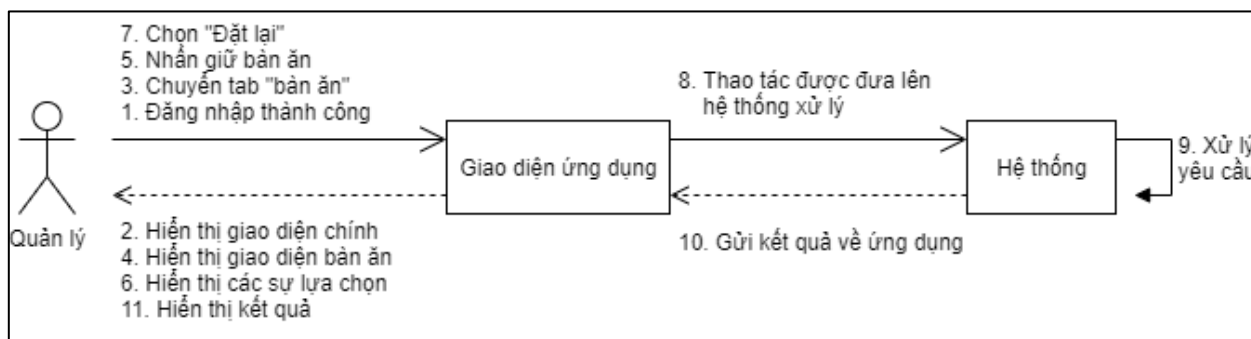
Sơ đồ 60 Sơ đồ tác nhân - usecase chức năng đặt lại bàn ăn

Sơ đồ tuần tự:



Sơ đồ 61 Sơ đồ tuần tự chức năng đặt lại bàn ăn

Sơ đồ hợp tác:



Sơ đồ 62 Sơ đồ hợp tác chức năng đặt lại bàn ăn

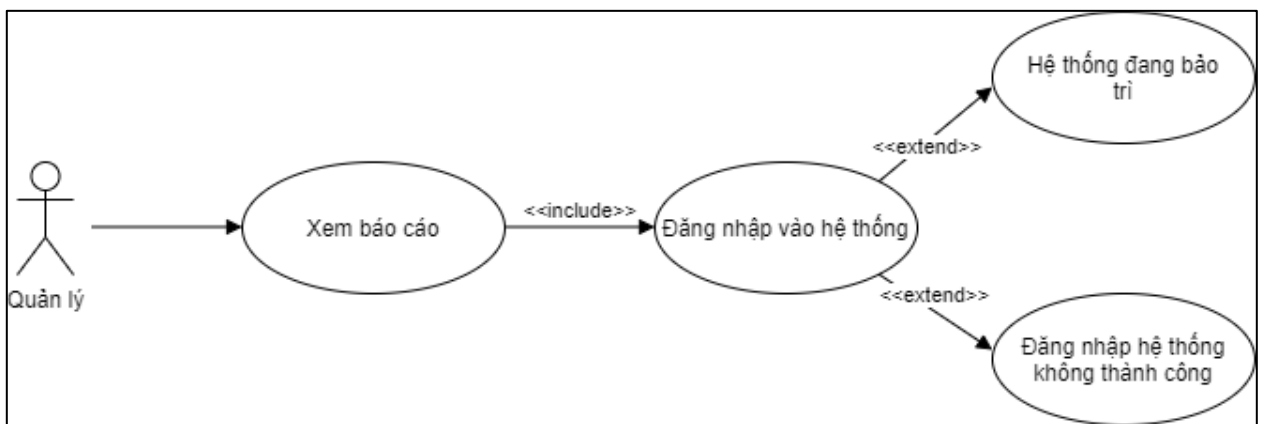
2.2.3.13. Chức năng xem báo cáo:

Sau khi đăng nhập thành công, quản lý chuyển tab “Profile” tại giao diện chính. Ứng dụng hiển thị các lựa chọn, người dùng chọn “Report” để gửi yêu cầu hiển thị

báo cáo. Ứng dụng hiển thị giao diện với các thông tin như: biểu đồ doanh thu, doanh thu, thống kê món ăn, thống kê nhân viên ,...Quy trình như sau:

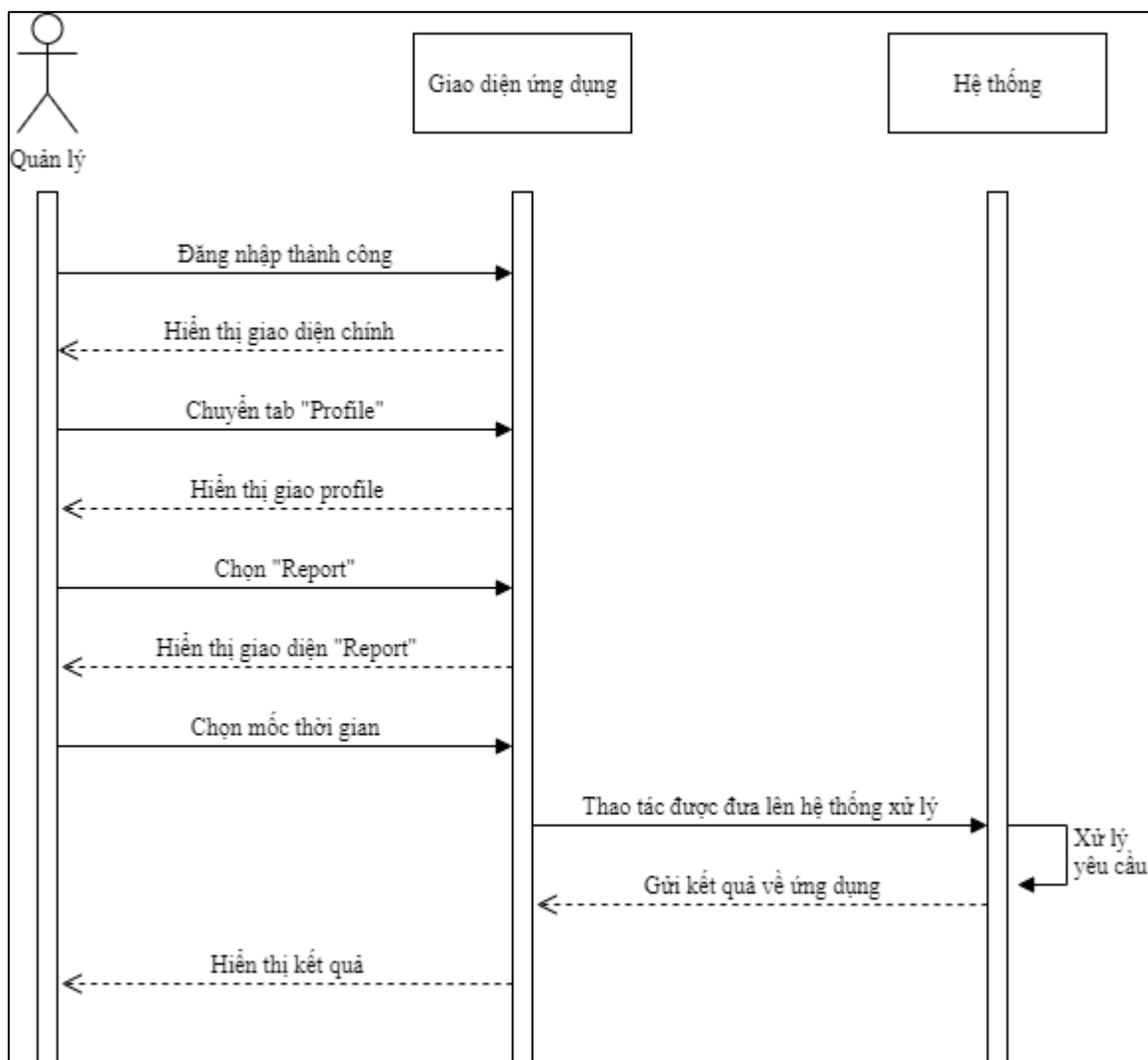
- Quản lý đăng nhập thành công;
- Ứng dụng hiển thị giao diện chính;
- Quản lý chuyển tab “Profile”;
- Ứng dụng hiển thị giao diện profile;
- Quản lý chọn “Report”;
- Ứng dụng hiển thị giao diện “Report”;
- Quản lý chọn mốc thời gian;
- Ứng dụng gửi yêu cầu lên hệ thống;
- Hệ thống xử lý yêu cầu;
- Hệ thống gửi kết quả về ứng dụng;
- Ứng dụng hiển thị kết quả.

Sơ đồ giữa tác nhân và use case:



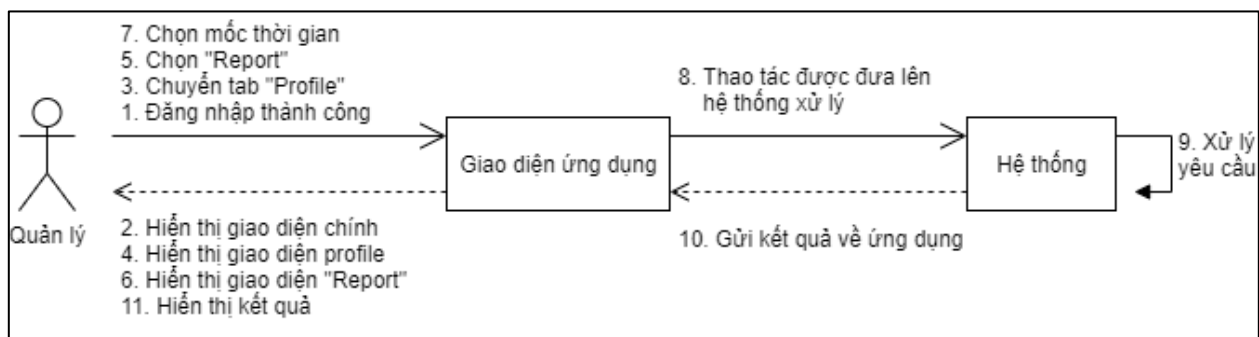
Sơ đồ 63 Sơ đồ tác nhân - usecase chức năng xem báo cáo

Sơ đồ tuần tự:



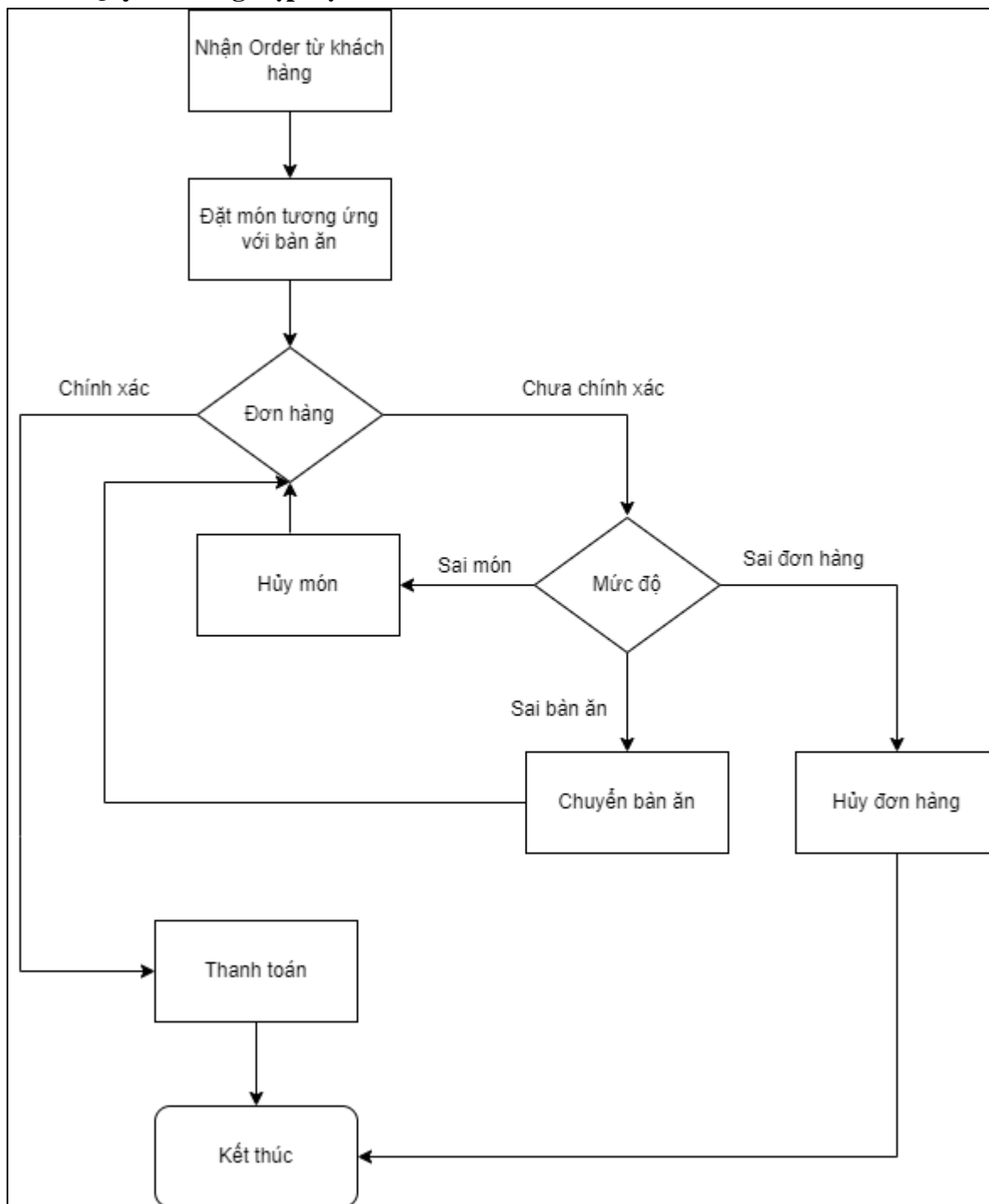
Sơ đồ 64 Sơ đồ tuần tự chức năng báo cáo

Sơ đồ hợp tác:



Sơ đồ 65 Sơ đồ hợp tác chức năng báo cáo

2.2.4. Quy trình nghiệp vụ

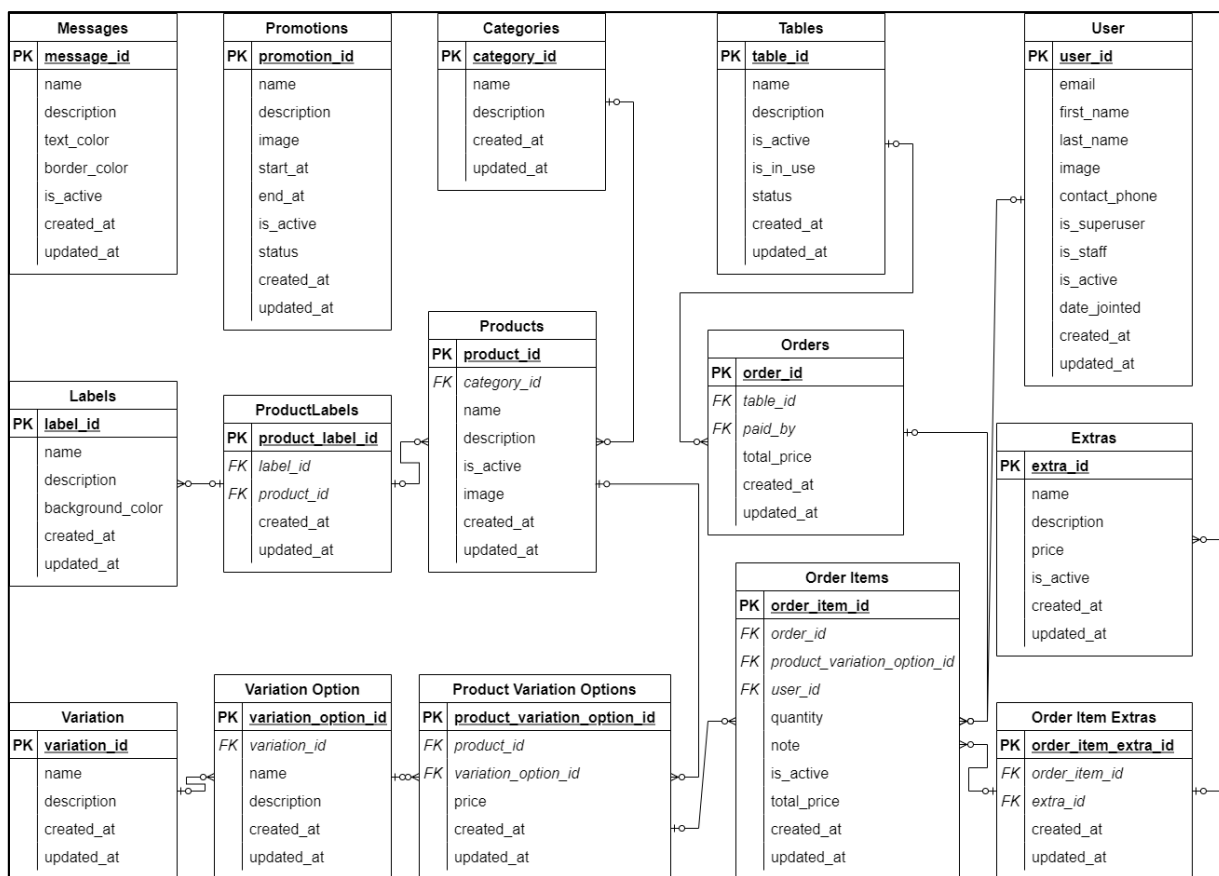


Sơ đồ 66 Sơ đồ quy trình nghiệp vụ

CHƯƠNG IV: THIẾT KẾ CƠ SỞ DỮ LIỆU

1. Sơ đồ cơ sở dữ liệu

Sơ đồ cơ sở dữ liệu phân tích nhằm lưu trữ thông tin Người dùng, Bàn ăn, Loại sản phẩm, Nhãn sản phẩm, Biển thể loại sản phẩm, Biển thể sản phẩm, Sản phẩm, Sản phẩm gọi thêm, Đơn hàng, Chi tiết đơn hàng, Chương trình khuyến mãi, Lời nhắn được thể hiện như sơ đồ 67.



Sơ đồ 67 Sơ đồ cơ sở dữ liệu tổng quát

2. Chi tiết cơ sở dữ liệu

Lưu ý: Các trường **In đậm gạch dưới** là khóa chính của bảng, **In đậm nghiêng gạch dưới** là khóa ngoại liên kết tới các bảng dữ liệu khác, viết thường là các trường dữ liệu cần lưu trong các bảng.

➤ Quản lý người dùng

Bảng 26 Cơ sở dữ liệu quản lý người dùng

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Users	user_id	Integer	Mã người dùng
2		email	Char(100)	Email người dùng
3		first_name	Nvarchar(100)	Tên người dùng

4		last_name	Nvarchar(100)	Họ người dùng
5		image	Binary	Hình đại diện người dùng
6		contact_phone	Char(11)	Số điện thoại người dùng
7		is_superuser	Boolean	Quyền quản lý
8		is_staff	Boolean	Quyền nhân viên
9		is_active	Boolean	Trạng thái kích hoạt
10		date_joined	DateTime	Thời gian tạo
11		created_at	DateTime	Thời gian tạo
12		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý bàn ăn

Bảng 27 Cơ sở dữ liệu quản lý bàn ăn

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Tables	table_id	Integer	Mã bàn ăn
2		name	Nvarchar(100)	Tên bàn ăn
3		description	Nvarchar(100)	Mô tả bàn ăn
4		is_active	Boolean	Trạng thái kích hoạt
5		is_in_use	Boolean	Trạng thái trạng sử dụng
6		status	String	Trạng thái
7		created_at	DateTime	Thời gian tạo
8		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý loại sản phẩm

Bảng 28 Cơ sở dữ liệu quản lý loại sản phẩm

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Categories	category_id	Integer	Mã loại sản phẩm
2		name	Nvarchar(100)	Tên loại sản phẩm
3		description	Nvarchar(100)	Mô tả loại sản phẩm
4		created_at	DateTime	Thời gian tạo
5		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý nhãn

Bảng 29 Cơ sở dữ liệu quản lý nhãn

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Labels	label_id	Integer	Mã nhãn sản phẩm
2		name	Nvarchar(100)	Tên nhãn sản phẩm
3		description	Nvarchar(100)	Mô tả nhãn sản phẩm
4		background_color	Char(10)	Màu nền nhãn sản phẩm
5		created_at	DateTime	Thời gian tạo
6		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý loại biến thể

Bảng 30 Cơ sở dữ liệu quản lý loại biến thể

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Variations	variation_id	Integer	Mã loại biến thể
2		name	Nvarchar(100)	Tên loại biến thể

3		description	Nvarchar(100)	Mô tả loại biến thể
4		created_at	DateTime	Thời gian tạo
5		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý biến thể

Bảng 31 Cơ sở dữ liệu quản lý biến thể

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Variation Options	<u>variation_option_id</u>	Integer	Mã biến thể
2		<u>variation_id</u>	Integer	Mã loại biến thể
3		name	Nvarchar(100)	Tên biến thể
4		description	Nvarchar(100)	Mô tả biến thể
5		created_at	DateTime	Thời gian tạo
6		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý sản phẩm

Bảng 32 Cơ sở dữ liệu quản lý sản phẩm

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Products	<u>product_id</u>	Integer	Mã sản phẩm
2		<u>category_id</u>	Integer	Mã loại sản phẩm
3		name	Nvarchar(100)	Tên sản phẩm
4		description	Nvarchar(100)	Mô tả sản phẩm
5		image	Binary	Hình ảnh sản phẩm
6		created_at	DateTime	Thời gian tạo
7		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý sản phẩm biến thể

Bảng 33 Cơ sở dữ liệu quản lý sản phẩm biến thể

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Product Variation Options	<u>product_variation_option_id</u>	Integer	Mã sản phẩm biến thể
2		<u>product_id</u>	Integer	Mã sản phẩm
3		<u>variation_option_id</u>	Integer	Mã biến thể
4		price	Numeric	Giá
5		created_at	DateTime	Thời gian tạo
6		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý đơn hàng

Bảng 34 Cơ sở dữ liệu quản lý đơn hàng

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Orders	<u>order_id</u>	Integer	Mã đơn hàng
2		<u>table_id</u>	Integer	Mã bàn ăn
3		<u>paid_by</u>	Integer	Mã người dùng
4		price	Numeric	Tổng giá
5		created_at	DateTime	Thời gian tạo
6		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý chi tiết đơn hàng

Bảng 35 Cơ sở dữ liệu quản lý chi tiết đơn hàng

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Order Items	<u>order_item_id</u>	Integer	Mã chi tiết đơn hàng
2		<u>order_id</u>	Integer	Mã đơn hàng
3		<u>product_variation_option_id</u>	Integer	Mã biến thể sản phẩm
4		<u>user_id</u>	Integer	Mã người dùng
5		quantity	Integer	Số lượng
6		note	Nvarchar(100)	Ghi chú
7		total_price	Numeric	Tổng giá
8		is_active	Boolean	Trạng thái kích hoạt
9		created_at	DateTime	Thời gian tạo
10		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý sản phẩm gọi thêm của chi tiết đơn hàng

Bảng 36 Cơ sở dữ liệu quản lý sản phẩm gọi thêm chi tiết đơn hàng

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Order Item Extras	<u>order_item_extra_id</u>	Integer	Mã sản phẩm gọi thêm chi tiết đơn hàng
2		<u>order_item_id</u>	Integer	Mã chi tiết đơn hàng
3		<u>extra_id</u>	Integer	Mã sản phẩm gọi thêm
4		created_at	DateTime	Thời gian tạo
5		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý chương trình khuyến mãi

Bảng 37 Cơ sở dữ liệu quản lý chương trình khuyến mãi

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Promotions	<u>promotion_id</u>	Integer	Mã chương trình khuyến mãi
2		name	Nvarchar(100)	Tên chương trình khuyến mãi
3		description	Nvarchar(100)	Mô tả chương trình khuyến mãi
4		image	Binary	Hình ảnh chương trình khuyến mãi
5		start_at	DateTime	Thời gian hiệu lực chương trình khuyến mãi
6		end_at	DateTime	Thời gian hết hạn chương trình khuyến mãi
7		is_active	Boolean	Trạng thái kích hoạt
8		created_at	DateTime	Thời gian tạo
9		updated_at	DateTime	Thời gian chỉnh sửa

➤ Quản lý lời nhắn

Bảng 38 Cơ sở dữ liệu quản lý lời nhắn

STT	Tên bảng	Tên cột	Kiểu	Diễn giải
1	Messages	<u>message_id</u>	Integer	Mã lời nhắn
2		name	Nvarchar(100)	Tên lời nhắn
3		description	Nvarchar(100)	Mô tả lời nhắn
4		text_color	Char(10)	Màu sắc chữ
5		border_color	Char(10)	Màu sắc viền

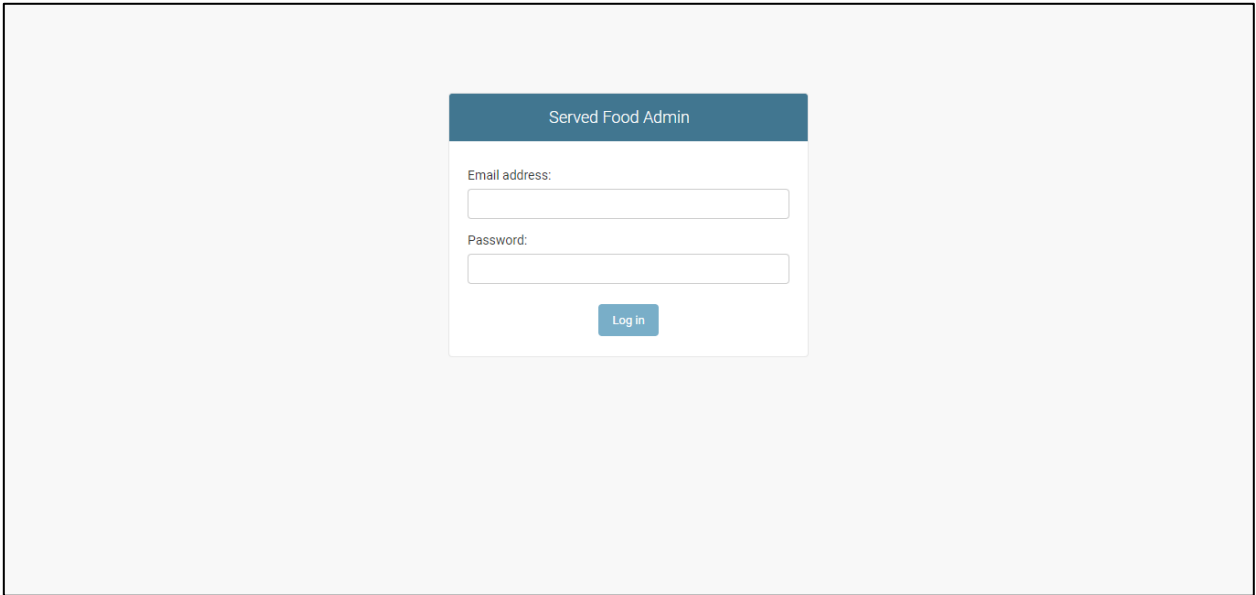
6		is_active	Boolean	Trạng thái kích hoạt
7		created_at	DateTime	Thời gian tạo
8		updated_at	DateTime	Thời gian chỉnh sửa

CHƯƠNG V: XÂY DỰNG HỆ THỐNG

1. Xây dựng Website quản trị sử dụng Django

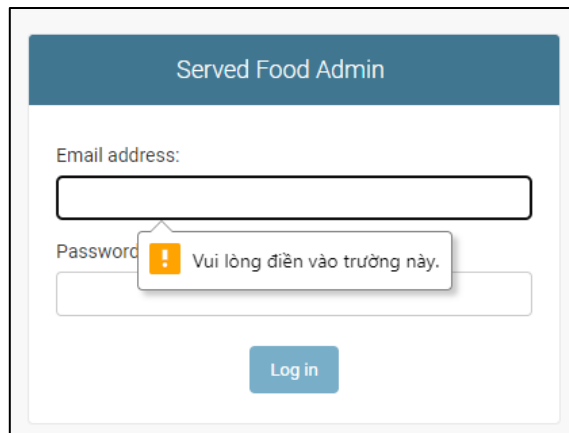
1.1. Giao diện đăng nhập

Khi người quản lý truy cập vào website quản trị lần đầu, hiển thị đầu tiên là giao diện đăng nhập. Chức năng đăng nhập yêu cầu các thông tin bao gồm: Email và Password để truy cập vào hệ thống. Giao diện đăng nhập như hình 8.

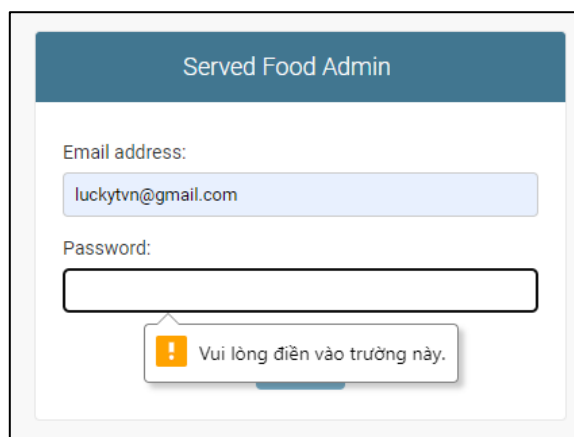


Hình 8 Giao diện đăng nhập website

Quản lý phải điền đầy đủ thông tin Email và Password, nếu không thành công giao diện sẽ hiển thị như hình 9 và 10



Hình 9 Giao diện đăng nhập website thiếu email



Served Food Admin

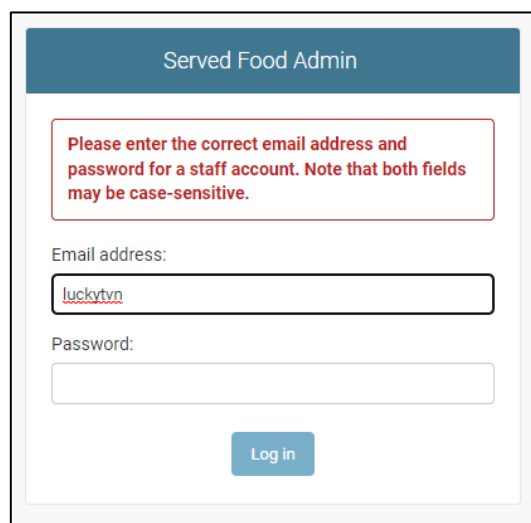
Email address:
luckytvn@gmail.com

Password:

Vui lòng điền vào trường này.

Hình 10 Giao diện đăng nhập website thiếu mật khẩu

Trong trường hợp người quản lý đã điền đầy đủ thông tin, hệ thống sẽ xác thực các trường thông tin. Cụ thể hệ thống sẽ kiểm tra tính hợp lệ của Email và tính chính xác của Password. Giao diện sẽ hiển thị như hình 11



Served Food Admin

Please enter the correct email address and password for a staff account. Note that both fields may be case-sensitive.

Email address:
luckytvn

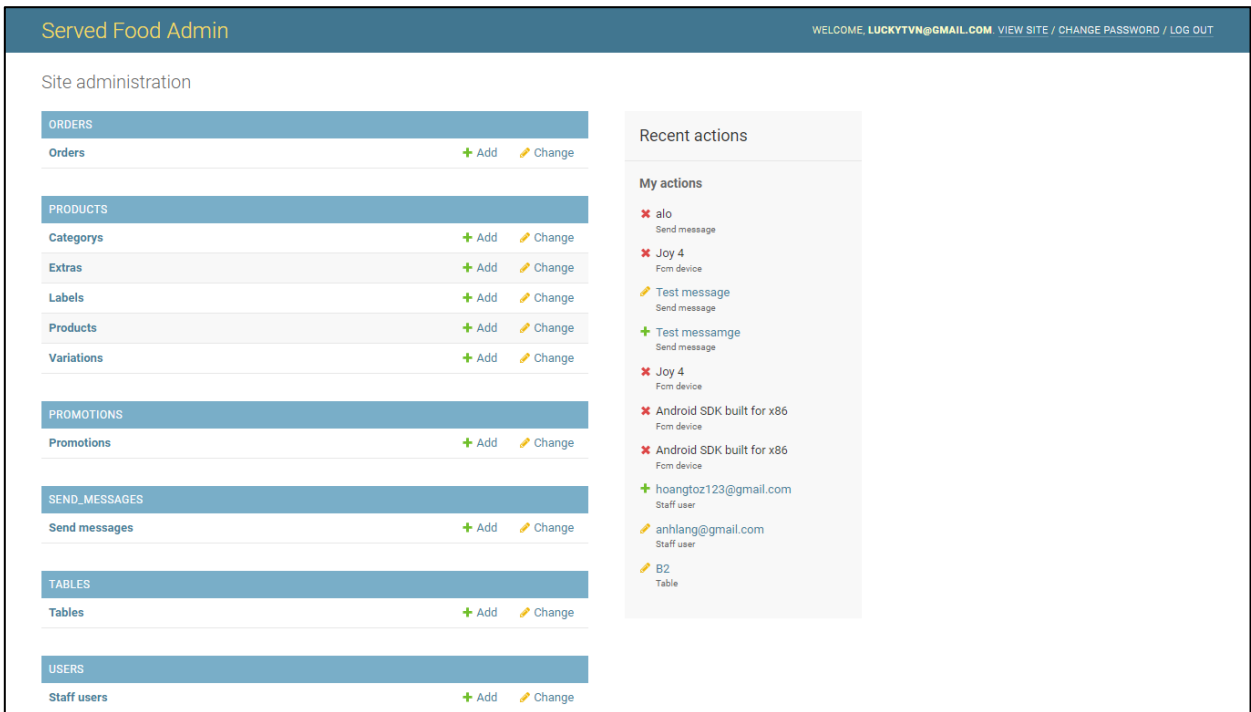
Password:

Log in

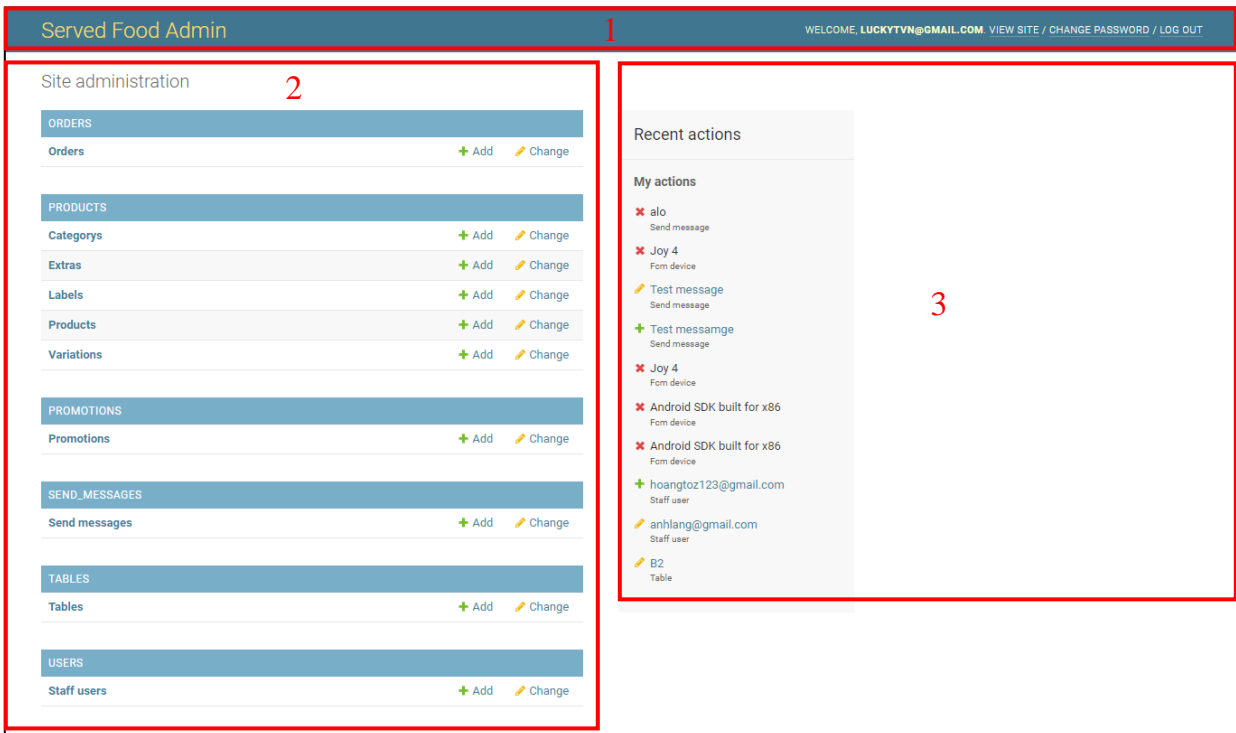
Hình 11 Giao diện đăng nhập website sai email

1.2. Giao diện chính

Sau khi đăng nhập thành công, website quản trị sẽ điều hướng đến giao diện chính. Giao diện chính như hình 12.



Hình 12 Giao diện trang chủ website
 Giao diện chính được chia ra thành các khối như hình 13



Hình 13 Giao diện trang chủ website theo khối
 Như hình 13, giao diện chính sẽ được chia ra thành 3 khối. Trong đó:

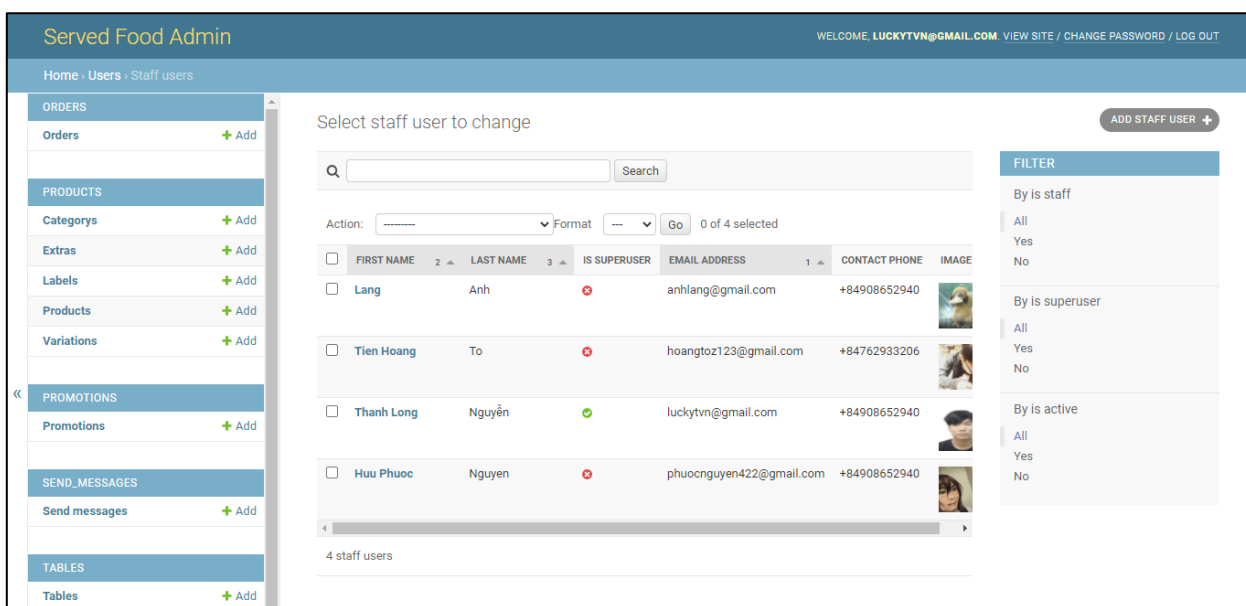
1. Header: Bao gồm tiêu đề website quản trị, lời chào đến người dùng và các tính năng như : View Site – Truy cập vào website giới thiệu (Hiện chưa phát triển), Change Password – Thay đổi mật khẩu, Log Out – Đăng xuất.
2. Danh mục: Bao gồm các danh mục quản lý như: Orders - Đơn hàng; Products - Các danh mục liên quan đến sản phẩm: Category (Loại sản phẩm), Extras (Sản phẩm gọi thêm), Labels (Nhãn sản phẩm), Products (Sản phẩm), Variations (Biến thể sản phẩm); Promotions – Chương trình khuyến mãi; Send Messages – Lời nhắn; Tables – Bàn ăn; Users – Người dùng (Tài khoản người dùng).

Mỗi danh mục sẽ bao gồm các danh mục con, chẳng hạn như danh mục cha “PRODUCTS” có các danh mục con “Category”, “Extras”,... Để truy cập vào danh mục, quản lý cần phải nhấp vào tên danh mục. Người quản lý có thể thêm hoặc thay đổi bản ghi thông qua Shortcut: Add, Change. Recent Actions: Bản ghi lại lịch sử thao tác quản lý dữ liệu như thêm, sửa, xóa.

3. Lịch sử thao tác sẽ được hiển thị chi tiết các thông tin bao gồm: Tên bản ghi, Danh mục bản ghi, và các kí hiệu: Dấu X màu đỏ - hành động xóa; Cây bút màu vàng – hành động chỉnh sửa; Dấu cộng màu xanh lá – hành động thêm. Với những hành động thêm và chỉnh sửa được hiển thị với tên bản ghi màu xanh dương, người quản lý có thể trực tiếp truy cập vào bản ghi bằng cách nhấp vào tên bản ghi.

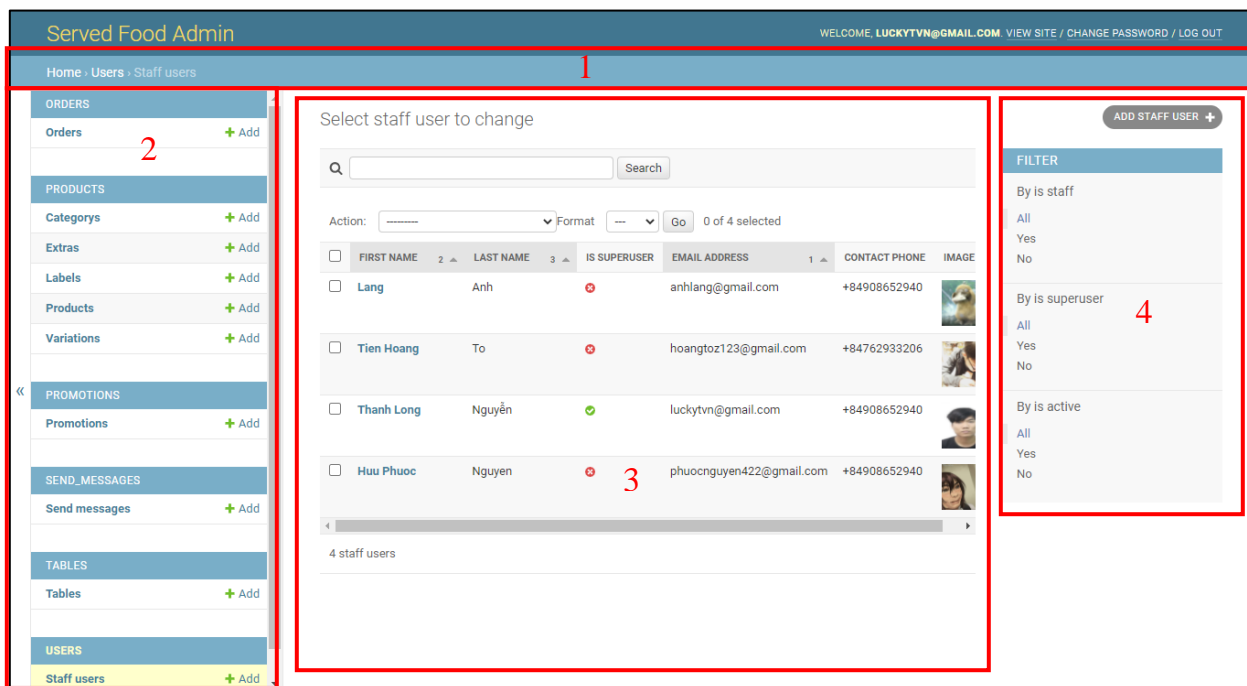
1.3. Giao diện chung của các danh mục

Ở đây sử dụng giao diện danh Users làm đại diện cho cách hiển thị giao diện của các danh mục khác. User tức người dùng trong hệ thống, cũng như tài khoản đăng nhập vào hệ thống. Người quản lý sau khi truy cập vào danh mục Staff Users, website hiển thị giao diện như hình 14.



Hình 14 Giao diện danh mục người dùng

Giao diện được chia thành các khối theo hình 15.

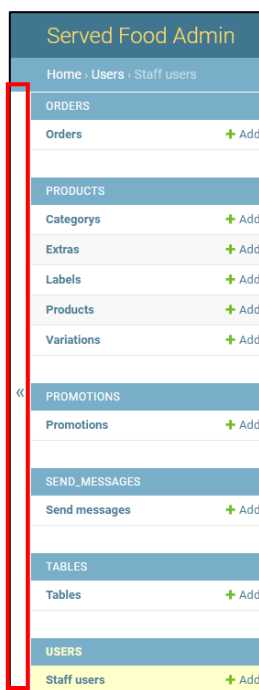


Hình 15 Giao diện danh mục người dùng theo khối

Như hình 15, giao diện sẽ được chia thành 4 khối:

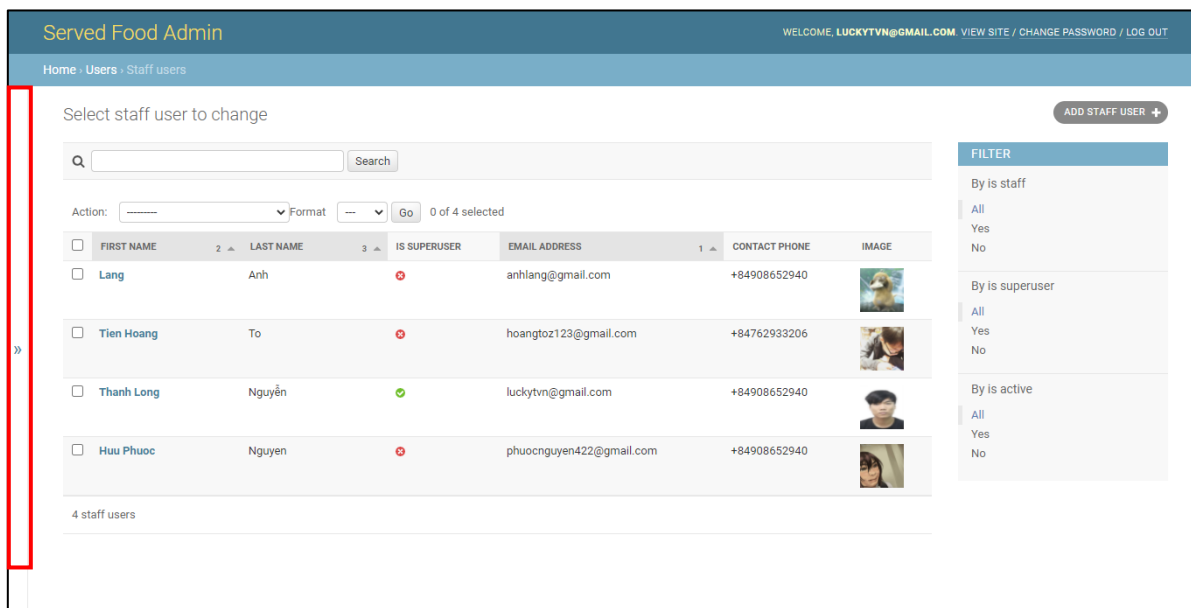
1. Đường dẫn danh mục: Ví dụ: Home/Users/Staff Users – Đường dẫn từ Home – Trang chủ đến danh mục cha Users đến danh mục con Staff Users.
2. Danh sách danh mục: Hiện thị danh mục cha và các danh mục con.

Màu nền danh mục con được hiển thị màu vàng là dấu hiệu nhận biết danh mục con đang được truy cập đến.



Hình 16 Giao diện ẩn danh sách danh mục

Như hình 16, ngay bên cạnh các danh mục là nút ẩn danh mục đi có kí hiệu <<. Thao tác này nhằm mở rộng diện tích hiển thị của khối (3), (4).



Hình 17 Giao diện hiện danh mục

Sau khi ẩn danh mục đi, giao diện sẽ hiển thị như hình 17. Nút ẩn danh mục chuyển thành nút hiện danh mục với kí hiệu >>.

3. Danh sách bản ghi: Gồm danh sách các bản ghi, chức năng tìm kiếm, chức năng xóa đa bản ghi (xóa nhiều bản ghi) và chức năng xuất file báo cáo như hình 18.

<input type="checkbox"/>	FIRST NAME	LAST NAME	IS SUPERUSER	EMAIL ADDRESS	CONTACT PHONE	IMAGE
<input checked="" type="checkbox"/>	Huu Phuoc	Nguyen		phuocnguyen422@gmail.com	+84908652940	
<input type="checkbox"/>	Lang	Anh		anhlang@gmail.com	+84908652940	
<input checked="" type="checkbox"/>	Thanh Long	Nguyễn		luckytvn@gmail.com	+84908652940	
<input type="checkbox"/>	Tien Hoang	To		hoangtoz123@gmail.com	+84762933206	

4 staff users

Hình 18 Giao diện danh sách bản ghi

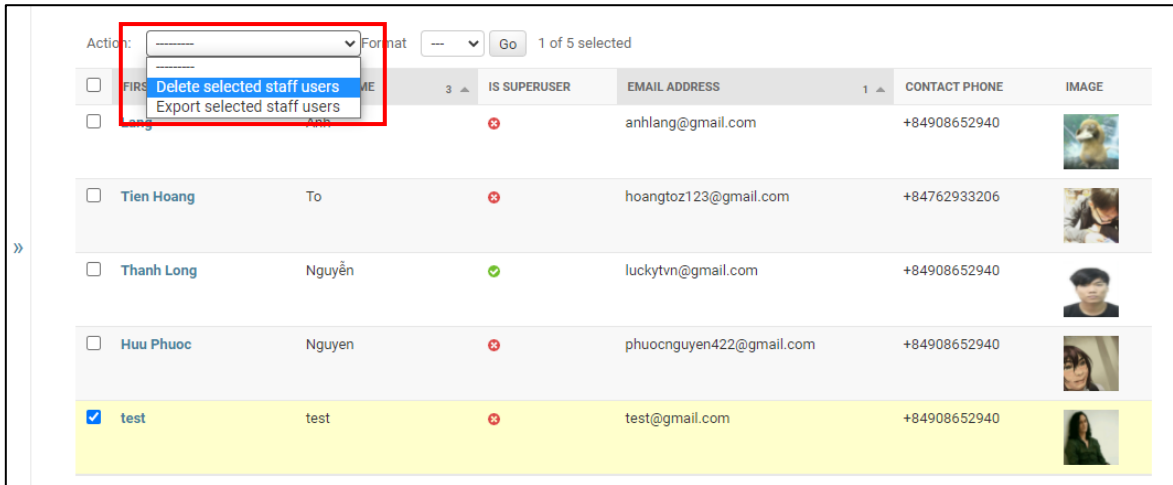
Hình 18 thể hiện một danh sách bản ghi, bao gồm chức năng Sorting (Sắp xếp); tính năng chọn nhiều bản ghi, mỗi bản ghi được chọn sẽ hiển thị dấu tích ở checkbox và giao diện như hình, nếu quản lý tích chọn vào checkbox ở trên thanh heading, tất cả bản ghi sẽ được chọn.

<input type="checkbox"/>	FIRST NAME	LAST NAME	IS SUPERUSER	EMAIL ADDRESS	CONTACT PHONE	IMAGE
<input type="checkbox"/>	Thanh Long	Nguyễn		luckytvn@gmail.com	+84908652940	

1 staff user

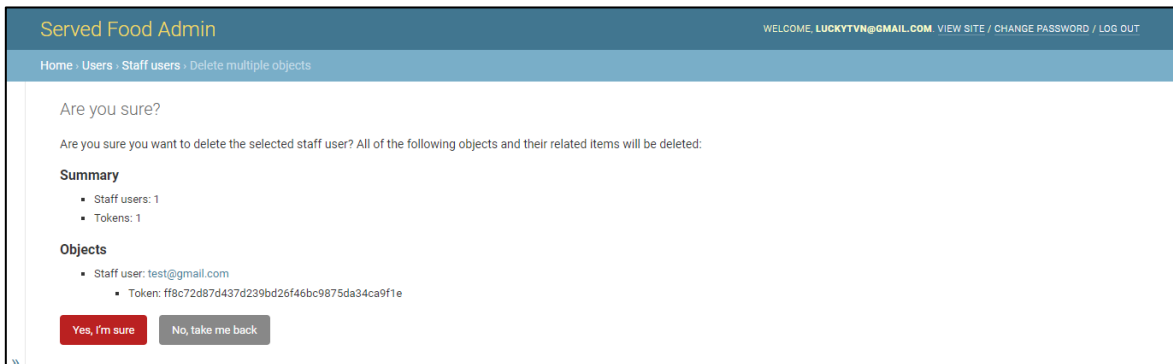
Hình 19 Giao diện tìm kiếm

Hình 29 thể hiện chức năng tìm kiếm. Trên thanh tìm kiếm, người quản lý nhập thông tin cần tìm kiếm và nhấn “Search”, kết quả trả về những bản ghi có phù hợp với dữ liệu tìm kiếm. Bên cạnh thanh tìm kiếm trả về kết quả số bản ghi tìm thấy trên tổng số bản ghi - 1 result (4 total). Khi nhấn vào “4 total” website trả về kết quả danh sách đầy đủ bản ghi.



Hình 20 Giao diện xóa đa bản ghi

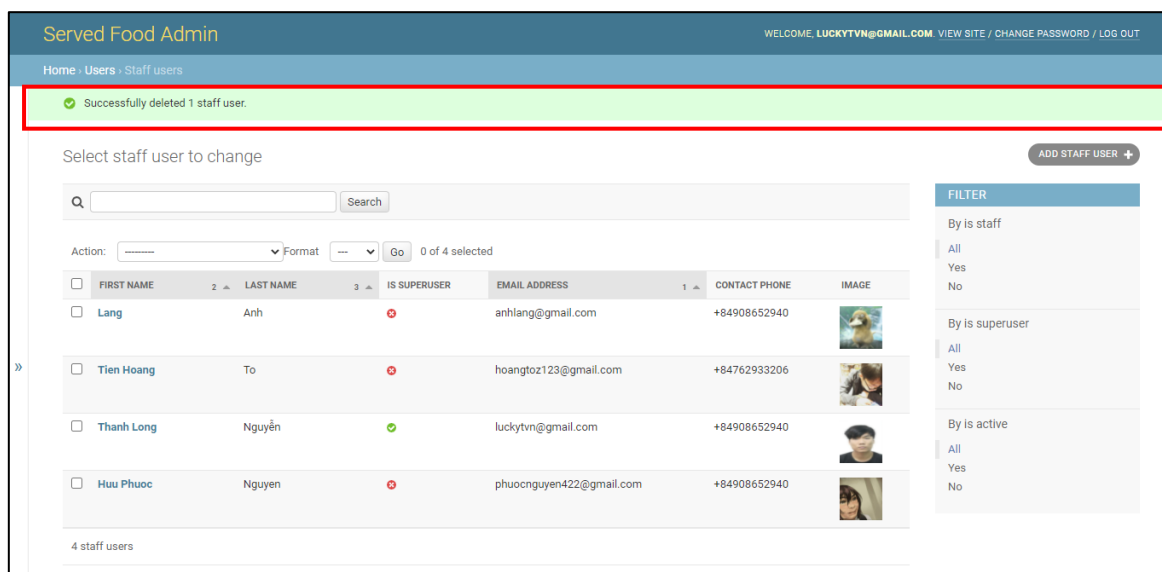
Hình 20 thể hiện tính năng xóa đa bản ghi, quản lý chọn “Delete selected staff users” ở thanh Action và nhấn “Go”, với những bản ghi được chọn tương ứng, website xử lý tác vụ và yêu cầu quản lý xác nhận như hình 21.



Hình 21 Giao diện xác nhận xóa đa bản ghi

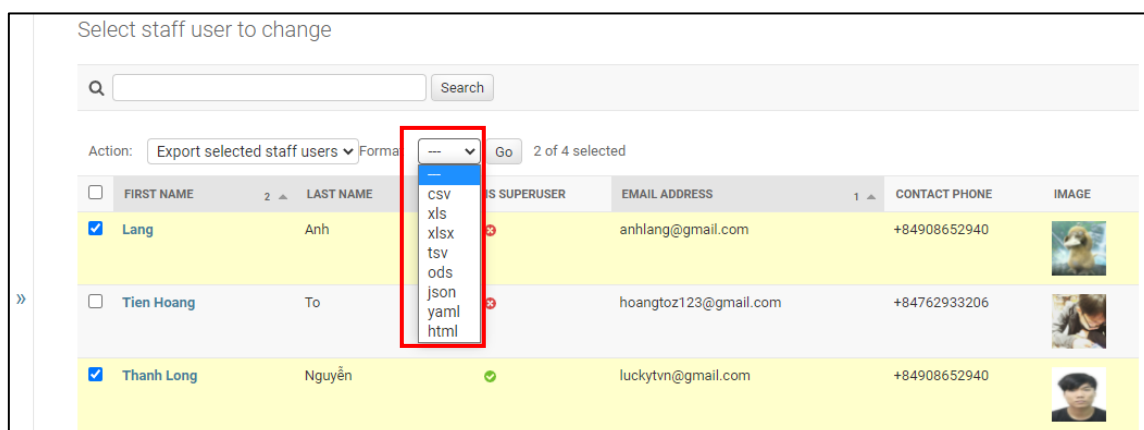
- Summary: Tóm tắt thông tin các bản ghi xóa.
- Objects: Chi tiết các bản ghi xóa.
- Nút “Yes, I’m sure”: Xác nhận thao tác xóa.
- Nút “No, take me back”: Từ chối xóa và trở về giao diện trước.

Sau khi xác nhận thao tác xóa, hệ thống xóa bản ghi khỏi cơ sở dữ liệu và hiển thị giao diện với kết quả thành công như hình 22.



Hình 22 Giao diện xóa đa bản ghi thành công

Tiếp theo đến với chức năng xuất file báo cáo, giao diện thao tác như hình 23.



Hình 23 Giao diện xuất báo cáo

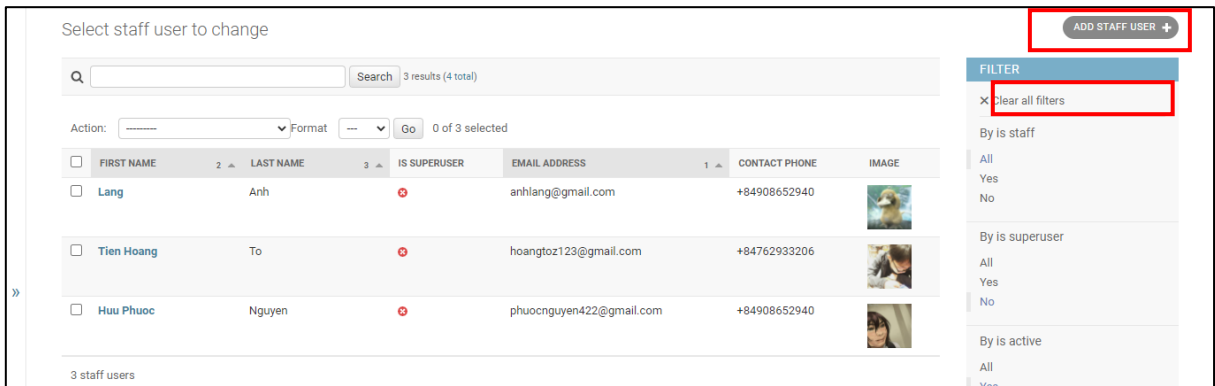
Hình 23 thể hiện một danh kiểu file mà người quản lý muốn xuất. Sau khi người quản lý chọn một kiểu file và nhấn “Go”. Hệ thống sẽ cho phép người quản lý tải một file dữ liệu về máy.

	A	B	C	F	G	H	I	J	K	L	N	O
1	id	password	last_login	email	first_name	last_name	image	contact_p	is_superuser	is_staff	is_active	date_joined
2	2	pbkdf2_sha256	9/16/2021 22:05	anhlang@gmail.com	Lang	Anh	46690586_46690586	8.49E+10	0	1	1	9/16/2021 22:05
3	1	pbkdf2_sha256	11/27/2021 8:46	luckytvn@gmail.com	Thanh Long	Nguyễn	GetImage	8.49E+10	1	1	1	9/16/2021 18:29
4												

Hình 24 Ví dụ một file báo cáo

Hình 24 thể hiện một file dữ liệu kiểu .csv sau khi tải từ website về máy.

4. Filter: Bộ lọc. Quản lý chọn “All”, “Yes” hoặc “No” theo các trường thông tin và website lọc theo các dữ liệu ấy trả về danh sách bản ghi tương ứng. Mô tả giao diện như hình 25.



Hình 25 Giao diện bộ lọc

Theo hình 25, những lựa chọn All, Yes, No khi được lựa chọn sẽ hiển thị màu xanh đồng thời hiển thị “Clear all filter” – Xóa bỏ các filters để trả danh sách bản ghi về ban đầu. Tại đây người quản lý có thể chọn nút “ADD STAFF USER” để tạo mới một bản ghi.

1.4. Giao diện Users (Người dùng)

1.4.1. Giao diện thêm mới User

Sau khi người quản lý chọn “ADD STAFF USER” hay bất cứ phím tắt nào để thêm một người dùng. Giao diện sẽ được thể hiện như hình 26.

Hình 26 Thêm mới người dùng

Để tạo mới một người dùng, người quản lý cần nhập đầy đủ và chính xác các thông tin với “Password” và “Confirm Password” là mật khẩu và xác nhận mật khẩu, 2 trường thông tin này phải nhập giống nhau để xác thực mật khẩu đúng và mật khẩu sẽ bị ẩn thành dấu “*”. “Is Active” là trạng thái kích hoạt của tài khoản, không được kích hoạt không thể sử dụng, “Is superuser” để xác định người dùng có phải là quản lý hay không, “Is staff” để xác định người dùng có phải nhân viên hay không. Nếu người quản lý nhập không chính xác thông tin, website hiển thị các lỗi như hình 27.

The screenshot shows the 'Served Food Admin' interface. At the top, there is a navigation bar with the title 'Served Food Admin' and links for 'WELCOME, LUCKYTVN@GMAIL.COM', 'VIEW SITE / CHANGE PASSWORD / LOG OUT'. Below the navigation bar, the breadcrumb trail is 'Home > Users > Staff users > Add staff user'. The main content area is titled 'Add staff user' and includes a sub-header 'First, enter a username and password. Then, you'll be able to edit more user options.' A red-bordered box contains the message 'Please correct the errors below.' Below this, there are three input fields: 'Email address:', 'Password:', and 'Confirm Password:'. Each field has a red border and the text 'This field is required.' above it. Below these fields is a section titled 'Personal info' with three more input fields: 'First name:', 'Last name:', and 'Contact phone:'. Each of these fields also has a red border and the text 'This field is required.' above it. The 'Image:' field has a red border and the text 'Chon tệp' (Choose file) and 'Không có tệp nào được chọn' (No files selected). Below the 'Personal info' section is a 'Permissions' section with three checkboxes: 'Is active' (checked), 'Is superuser' (unchecked), and 'Is staff' (unchecked). At the bottom of the form, there is a 'Table:' label with a dropdown menu and a plus sign. At the very bottom of the form, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Hình 27 Giao diện thêm người dùng lỗi

Trong trường hợp “Password” và “Confirm Password” không trùng khớp, website báo lỗi như hình 28.

Served Food Admin

WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home · Users · Staff users · Add staff user

Add staff user

First, enter a username and password. Then, you'll be able to edit more user options.

Please correct the error below.

Email address:

Password:

Your passwords must match

Confirm Password:

Hình 28 Giao diện mật khẩu không trùng khớp

Mỗi một người dùng sẽ có một email riêng biệt, tức các người dùng không thể trùng email với nhau. Trường hợp quản lý tạo một người dùng mới trùng email, website báo lỗi như hình 39.

Served Food Admin

WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home · Users · Staff users · Add staff user

Add staff user

First, enter a username and password. Then, you'll be able to edit more user options.

Please correct the error below.

Staff user with this Email address already exists.

Email address:

Password:

Confirm Password:

Hình 29 Giao diện thêm người dùng trùng email

Khi nhập đầy đủ và chính xác các thông tin, người quản lý có 3 sự lựa chọn: Save and add another – Lưu và tạo thêm một người dùng khác; Save and continue editing – Lưu và ở trạng thái chỉnh sửa; Save – Lưu. Giao diện được mô tả như hình 30.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Users > Staff users > Add staff user

✔ The staff user "test@gmail.com" was changed successfully. You may add another staff user below.

Add staff user

First, enter a username and password. Then, you'll be able to edit more user options.

Email address:

Password:

Confirm Password:

Personal info

First name:

Last name:

Contact phone:

Image: Không có tệp nào được chọn

Permissions

Is active

Is superuser

Is staff

Table:

Hình 30 Giao diện thêm người dùng thành công

1.4.2. Giao diện chỉnh sửa User

Khi người quản lý muốn chỉnh sửa một bản ghi người dùng, tại giao diện danh sách bản ghi, người quản lý chọn một bản ghi và website hiển thị giao diện như hình 31. Tại đây quản lý điều chỉnh các thông tin cần thay đổi chọn lưu.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Users > Staff users > test@gmail.com

✔ The staff user "test@gmail.com" was added successfully. You may edit it again below.

Change staff user HISTORY

test@gmail.com

Email address: test@gmail.com

Password: algorithm: pbkdf2_sha256 iterations: 260000 salt: 1733VV***** hash: CkVnry*****
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name: test

Last name: test

Contact phone: +84908652940

Image: Currently: 176080530_1226274554470624_5085427267940184655_n.jpg
Change: Không có tệp nào được chọn

Permissions

Is active

Is superuser

Is staff

Table:

Hình 31 Giao diện chỉnh sửa người dùng

Change staff user

test@gmail.com

Email address: test@gmail.com

Password: algorithm: pbkdf2_sha256 iterations: 260000 salt: 1733VV***** hash: CkVnry*****
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using **this form**.

Hình 32 Giao diện "this form"

Tại hình 32 hiển thị "this form" cho phép người quản lý thay đổi mật khẩu đi nhấp chọn vào. Giao diện như hình 33.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home · Users · Staff users · test@gmail.com · Change password

Change password: test@gmail.com

Enter a new password for the user **test@gmail.com**.

Password:

Your password can't be too similar to your other personal information.
 Your password must contain at least 8 characters.
 Your password can't be a commonly used password.
 Your password can't be entirely numeric.

Password (again):

Enter the same password as before, for verification.

»

Hình 33 Giao diện thay đổi mật khẩu

- Hình 33 thể hiện quy tắc đặt mật khẩu.
- Người quản lý sau khi nhập mật khẩu và nhấn Change Password, kết quả trả về như hình 34 hoặc 35.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home · Users · Staff users · test@gmail.com · Change password

Change password: test@gmail.com

Please correct the error below.

Enter a new password for the user **test@gmail.com**.

Password:

Your password can't be too similar to your other personal information.
 Your password must contain at least 8 characters.
 Your password can't be a commonly used password.
 Your password can't be entirely numeric.

Password (again):

The two password fields didn't match.
 Enter the same password as before, for verification.

»

Hình 34 Giao diện thay đổi mật khẩu lỗi

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home · Users · Staff users · test@gmail.com

✔ Password changed successfully.

Change staff user

test@gmail.com

Email address:

Password: algorithm: pbkdf2_sha256 iterations: 260000 salt: xmrVST***** hash: B5a1eH*****
 Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Hình 35 Giao diện thay đổi mật khẩu thành công

1.4.3. Giao diện xóa User

Trong trường hợp người quản lý muốn xóa một bản ghi người dùng, người quản lý cần truy cập vào bản ghi và chọn “Delete” như hình 36.

The screenshot shows the 'Change staff user' page for a user named 'test@gmail.com'. The page includes fields for 'Email address', 'Password', 'Personal info' (First name, Last name, Contact phone), and 'Image'. At the bottom, there are three buttons: 'Delete' (highlighted with a red box), 'Save and add another', and 'Save and continue editing'. The 'Delete' button is a red rectangle with white text.

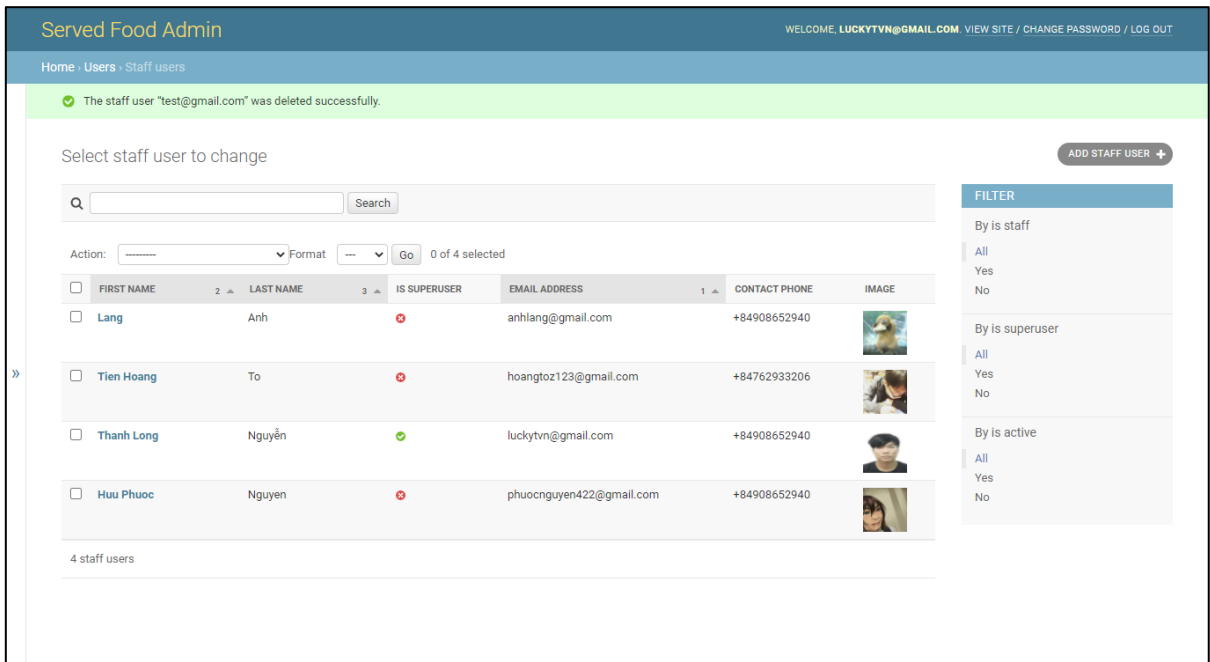
Hình 36 Giao diện xóa người dùng

Sau khi nhấn “Delete” hệ thống yêu cầu người quản lý xác nhận như hình 37.

The screenshot shows a confirmation dialog titled 'Are you sure?'. It asks: 'Are you sure you want to delete the selected staff user? All of the following objects and their related items will be deleted:'. Below this, it shows a 'Summary' section with 'Staff users: 1' and 'Tokens: 1'. Under 'Objects', it lists 'Staff user: test@gmail.com' and 'Token: ff8c72d87d437d239bd26f46bc9875da34ca9f1e'. At the bottom, there are two buttons: 'Yes, I'm sure' (red) and 'No, take me back' (grey).

Hình 37 Giao diện xác nhận xóa người dùng

Người quản lý chọn “Yes, Im sure” xác nhận yêu cầu xóa hoặc “No, take me back” để hủy yêu cầu và trở về giao diện trước đó. Giao diện xóa thành công hiển thị như hình 38.

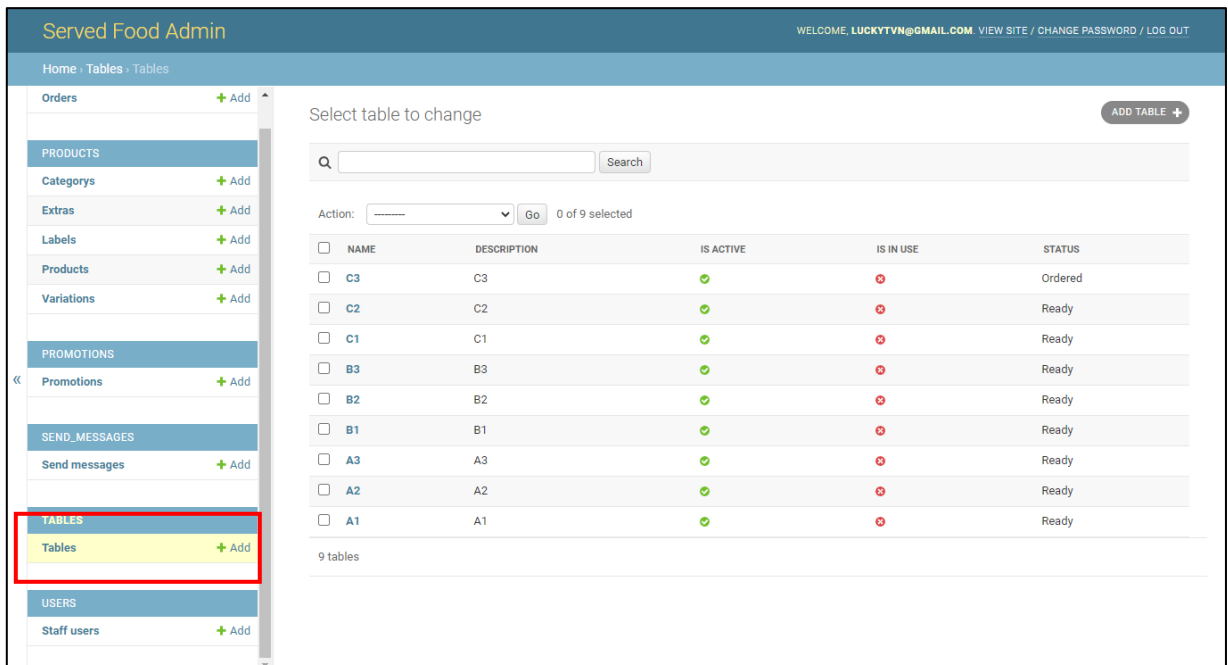


Hình 38 Giao diện xóa người dùng thành công

1.5. Giao diện Tables (Bàn ăn)

1.5.1. Giao diện chính Tables

Người quản lý chọn danh mục “Tables” thanh hiển thị các danh mục phía bên trái. Giao diện như hình 39.

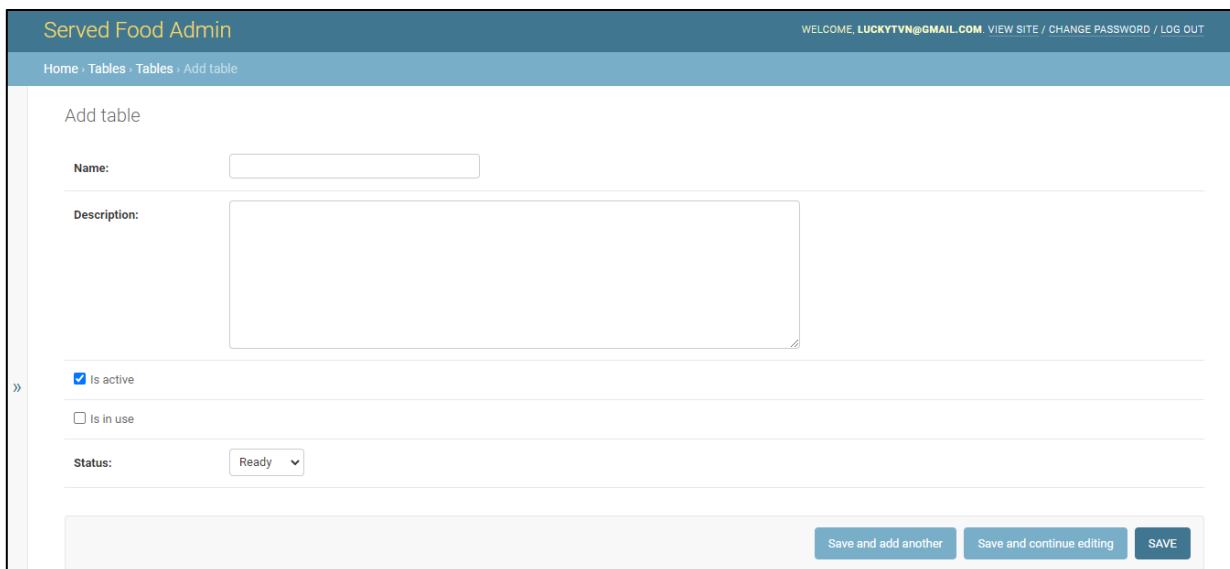


Hình 39 Giao diện danh mục bàn ăn

Danh sách bàn ăn được thể hiện với các thông tin name - tên, description - mô tả, is in use – trạng thái đang sử dụng, status – trạng thái bàn ăn (ready - có sẵn, ordered – đã đặt món).

1.5.2. Giao diện thêm mới Table

Để thêm mới bàn ăn, người quản lý chọn “ADD TABLE”. Giao diện thêm mới như hình 40.



The screenshot shows a web application interface for managing tables. The header includes the title 'Served Food Admin' and user information 'WELCOME, LUCKYTVN@GMAIL.COM' with links for 'VIEW SITE', 'CHANGE PASSWORD', and 'LOG OUT'. The breadcrumb trail is 'Home > Tables > Tables > Add table'. The main form is titled 'Add table' and contains the following fields:

- Name:** A text input field.
- Description:** A large text area.
- Is active:** A checked checkbox.
- Is in use:** An unchecked checkbox.
- Status:** A dropdown menu currently set to 'Ready'.

At the bottom of the form, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Hình 40 Giao diện thêm mới bàn ăn

Để thêm mới yêu cầu nhập đầy đủ thông tin, nếu không website hiển thị kết quả như hình 41.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Tables > Tables > Add table

Add table

Please correct the errors below.

Name: This field is required.

Description: This field is required.

Is active

Is in use

Status: Ready ▾

Hình 41 Giao diện thêm mới bàn ăn lỗi

Nếu người quản lý nhập đầy đủ thông tin và chọn lưu, hệ thống lưu bản ghi vào cơ sở dữ liệu và hiển thị kết quả thành công như hình 42.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Tables > Tables

✔ The table "D1" was added successfully.

Select table to change ADD TABLE +

Q

Action: ----- ▾ 0 of 10 selected

<input type="checkbox"/>	NAME	DESCRIPTION	IS ACTIVE	IS IN USE	STATUS
<input type="checkbox"/>	D1	D1	✔	✘	Ready
<input type="checkbox"/>	C3	C3	✔	✘	Ordered
<input type="checkbox"/>	C2	C2	✔	✘	Ready
<input type="checkbox"/>	C1	C1	✔	✘	Ready
<input type="checkbox"/>	B3	B3	✔	✘	Ready
<input type="checkbox"/>	B2	B2	✔	✘	Ready
<input type="checkbox"/>	B1	B1	✔	✘	Ready
<input type="checkbox"/>	A3	A3	✔	✘	Ready
<input type="checkbox"/>	A2	A2	✔	✘	Ready
<input type="checkbox"/>	A1	A1	✔	✘	Ready

10 tables

Hình 42 Giao diện thêm mới bàn ăn thành công

1.5.3. Giao diện chỉnh sửa Table

Để chỉnh sửa một bản ghi thông tin bàn ăn, người quản lý chọn một bản ăn để truy cập, giao diện chi tiết một bản ăn như hình 43.

The screenshot shows the 'Change table' form for table 'D1'. The form includes the following fields and controls:

- Name:** Input field containing 'D1'.
- Description:** Text area containing 'D1'.
- Is active:** Checked checkbox.
- Is in use:** Unchecked checkbox.
- Status:** Dropdown menu set to 'Ready'.
- Buttons:** 'Delete' (red), 'Save and add another' (blue), 'Save and continue editing' (blue), and 'SAVE' (blue).

Hình 43 Giao diện chỉnh sửa bàn ăn.

Sau khi thay đổi các trường thông tin hợp lý, người quản lý chọn lưu và website gửi về kết quả thành công như hình 44.

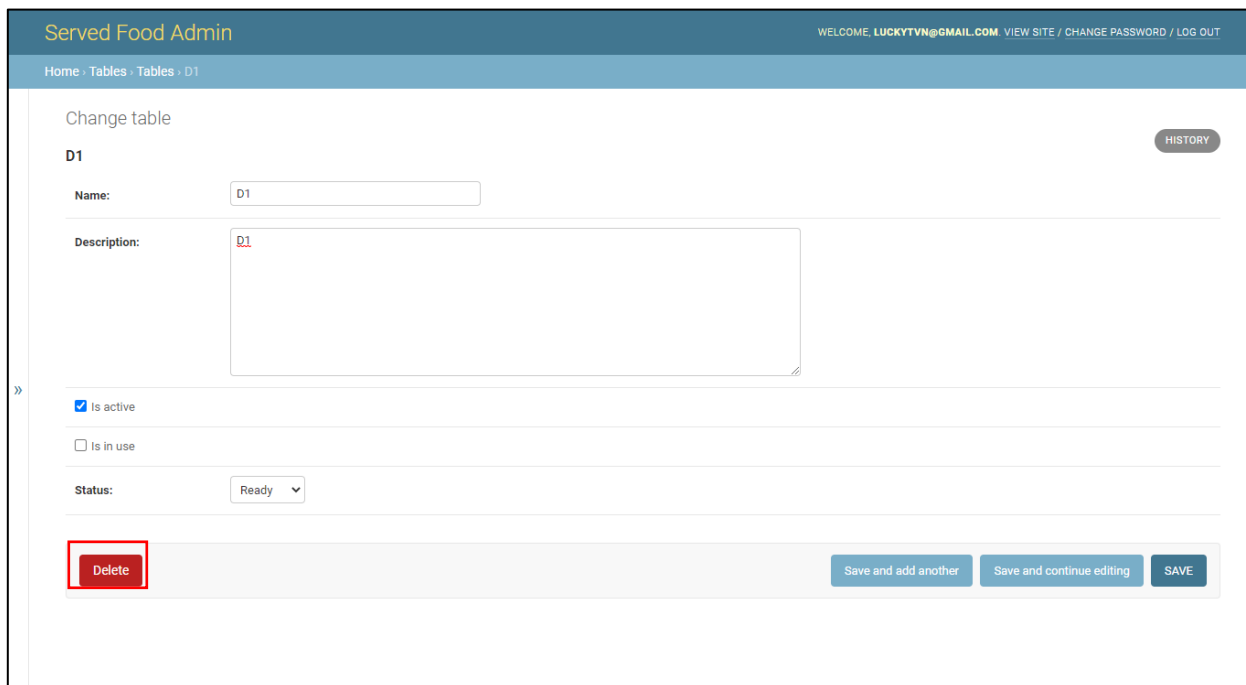
The screenshot shows the 'Select table to change' page with a success message and a table of tables. The success message is: "The table 'D1' was changed successfully." The table below lists 10 tables, with the 'D1 change' entry highlighted in red.

NAME	DESCRIPTION	IS ACTIVE	IS IN USE	STATUS
<input type="checkbox"/> D1	D1 change	✓	✗	Ready
<input type="checkbox"/> C3	C3	✓	✗	Ordered
<input type="checkbox"/> C2	C2	✓	✗	Ready
<input type="checkbox"/> C1	C1	✓	✗	Ready
<input type="checkbox"/> B3	B3	✓	✗	Ready
<input type="checkbox"/> B2	B2	✓	✗	Ready
<input type="checkbox"/> B1	B1	✓	✗	Ready
<input type="checkbox"/> A3	A3	✓	✗	Ready
<input type="checkbox"/> A2	A2	✓	✗	Ready
<input type="checkbox"/> A1	A1	✓	✗	Ready

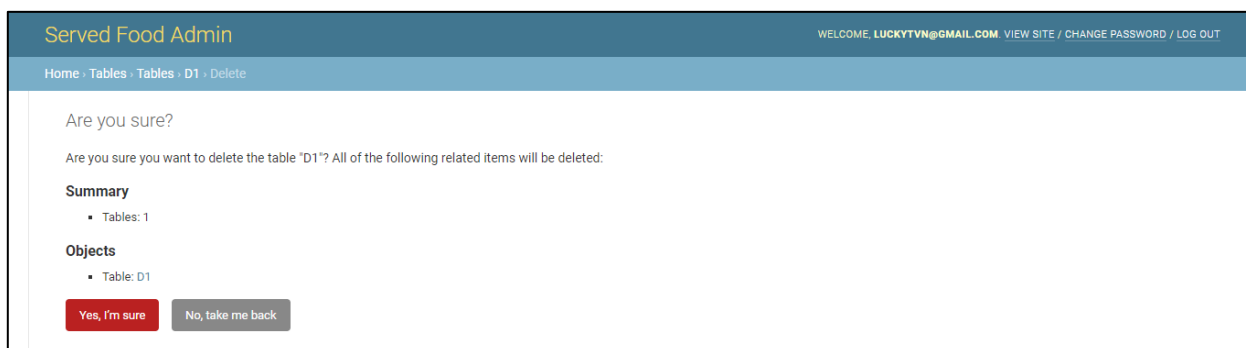
Hình 44 Giao diện chỉnh sửa bàn ăn thành công

1.5.4. Giao diện xóa Table

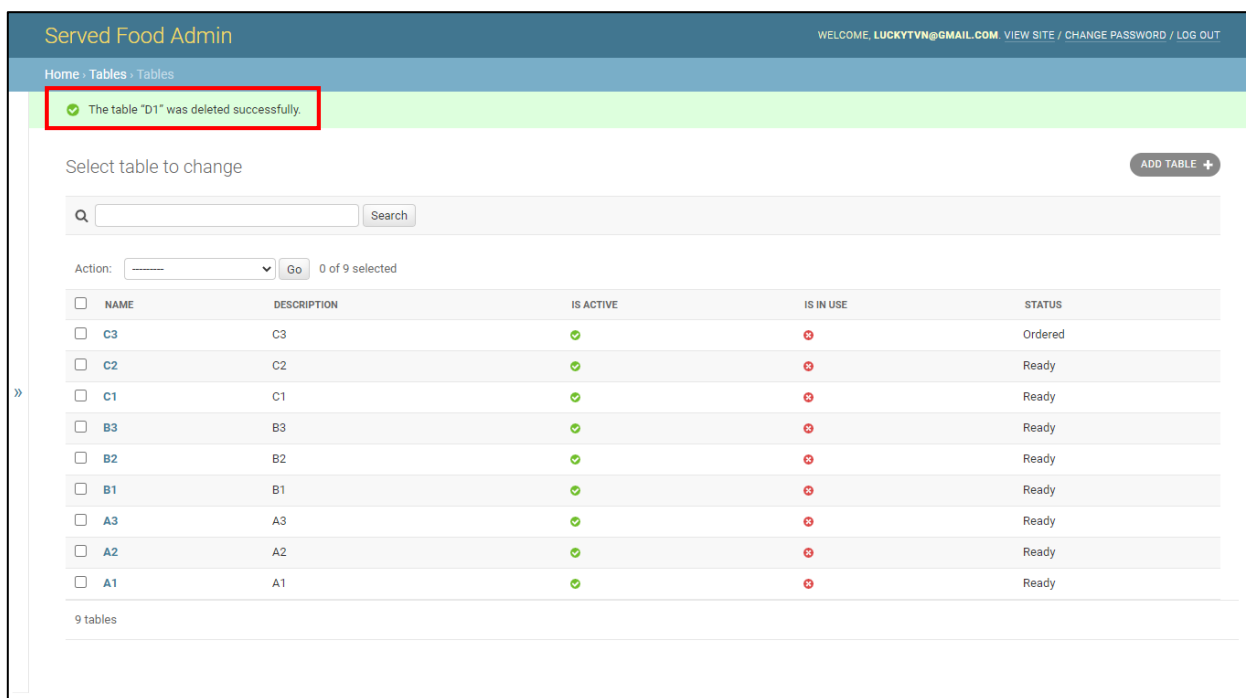
Để xóa một bản ghi bàn ăn, người quản lý truy cập bản ăn đó và chọn “Delete” như hình 45.



Hình 45 Giao diện xóa bàn ăn
Website yêu cầu xác nhận tác vụ xóa như hình 46.



Hình 46 Giao diện xác nhận xóa bàn ăn
Để xác nhận tác vụ xóa, người quản lý chọn “Yes, Im sure” hoặc “No, take me back” để hủy tác vụ. Giao diện xóa thành công hiển thị như hình 47.

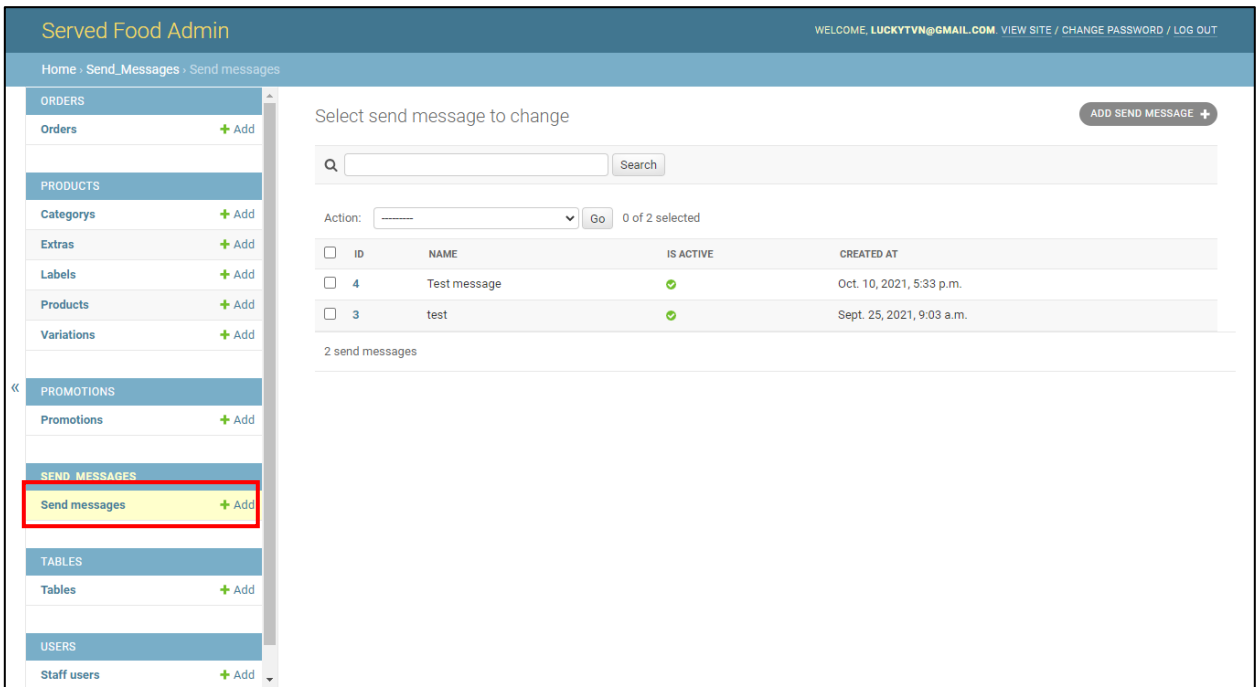


Hình 47 Giao diện xóa bàn ăn thành công

1.6. Giao diện Send Messages (Lời nhắn)

1.6.1. Giao diện chính Send Messages

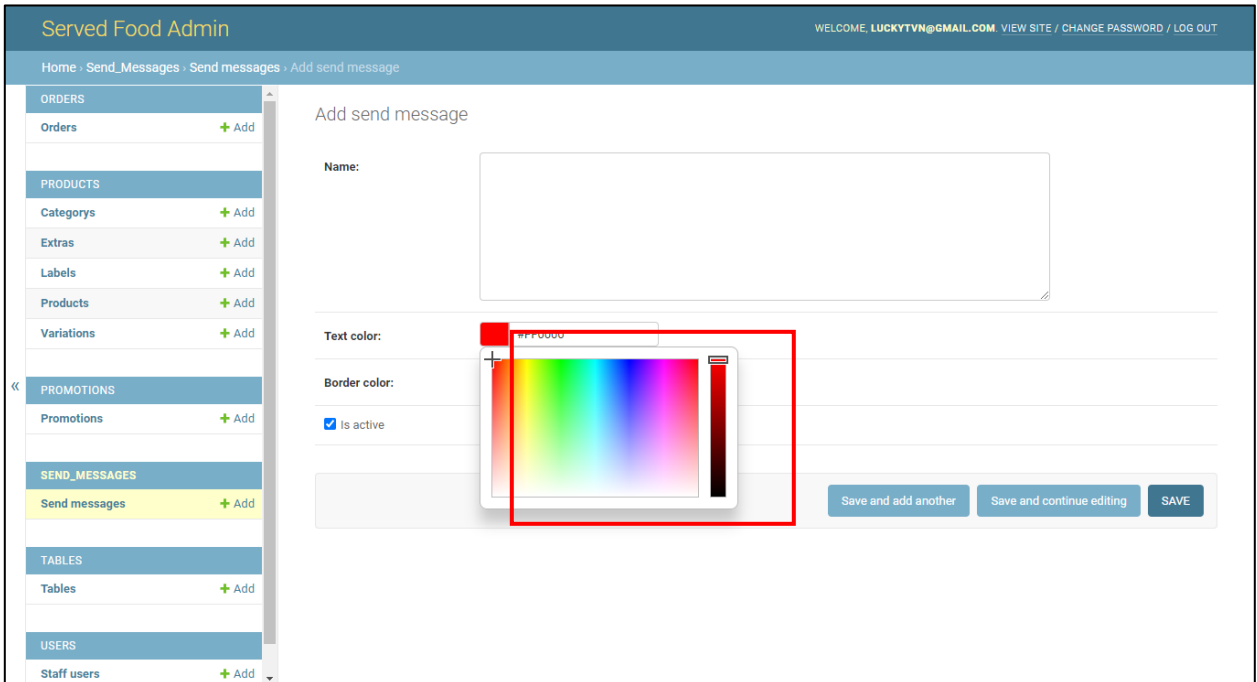
Để truy cập vào giao diện Send Messages, người quản lý chọn “Send messages” tại danh sách các danh mục phía bên trái, giao diện chính của Send Messages hiển thị danh sách lời nhắn với các thông tin như Name – Lời nhắn, Is Active – Trạng thái kích hoạt, Create At – Thời gian tạo lời nhắn, giao diện như hình 48.



Hình 48 Giao diện lời nhắn

1.6.2. Giao diện thêm mới Send Message

Để thêm mới một bản ghi, người quản lí chọn “ADD SEND MESSAGE”, giao diện website hiển thị như hình 49.



Hình 49 Giao diện thêm mới lời nhắn

Để thêm mới một bản ghi Send Message yêu cầu nhập đầy đủ thông tin và chọn màu sắc hiển thị trên ứng dụng di động. Trường hợp người quản lý chưa nhập đầy đủ thông tin, website báo lỗi như hình 50.

Served Food Admin

WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Send_Messages > Send messages > Add send message

ORDERS

- Orders + Add

PRODUCTS

- Categorys + Add
- Extras + Add
- Labels + Add
- Products + Add
- Variations + Add

PROMOTIONS

- Promotions + Add

SEND_MESSAGES

- Send messages + Add

TABLES

- Tables + Add

USERS

- Staff users + Add

Add send message

Please correct the error below.

Name: This field is required.

Text color:

Border color:

Is active

Save and add another Save and continue editing SAVE

Hình 50 Giao diện thêm lời nhắn lỗi

Khi người quản lý nhập chính xác các thông tin và chọn lưu, hệ thống lưu bản ghi lên cơ sở dữ liệu và hiển thị giao diện như hình 51.

Served Food Admin

WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Send_Messages > Send messages

The send message "test" was added successfully.

Select send message to change ADD SEND MESSAGE +

Q Search

Action: Go 0 of 3 selected

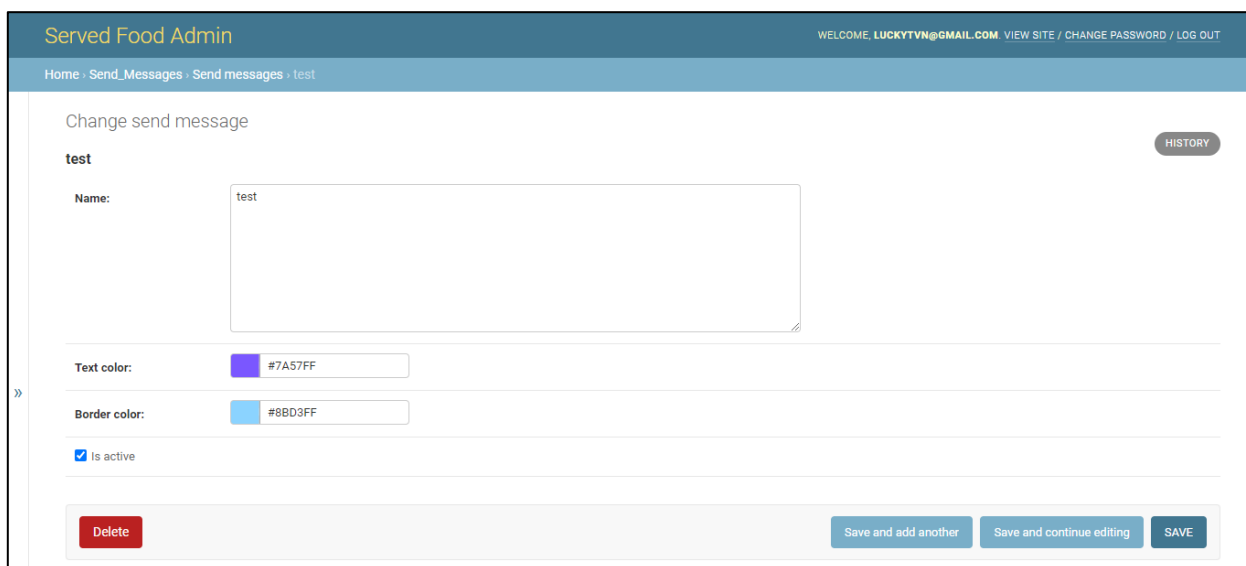
ID	NAME	IS ACTIVE	CREATED AT
<input type="checkbox"/> 5	test	<input checked="" type="checkbox"/>	Nov. 27, 2021, 4:33 p.m.
<input type="checkbox"/> 4	Test message	<input checked="" type="checkbox"/>	Oct. 10, 2021, 5:33 p.m.
<input type="checkbox"/> 3	test	<input checked="" type="checkbox"/>	Sept. 25, 2021, 9:03 a.m.

3 send messages

Hình 51 Giao diện thêm mới lời nhắn thành công

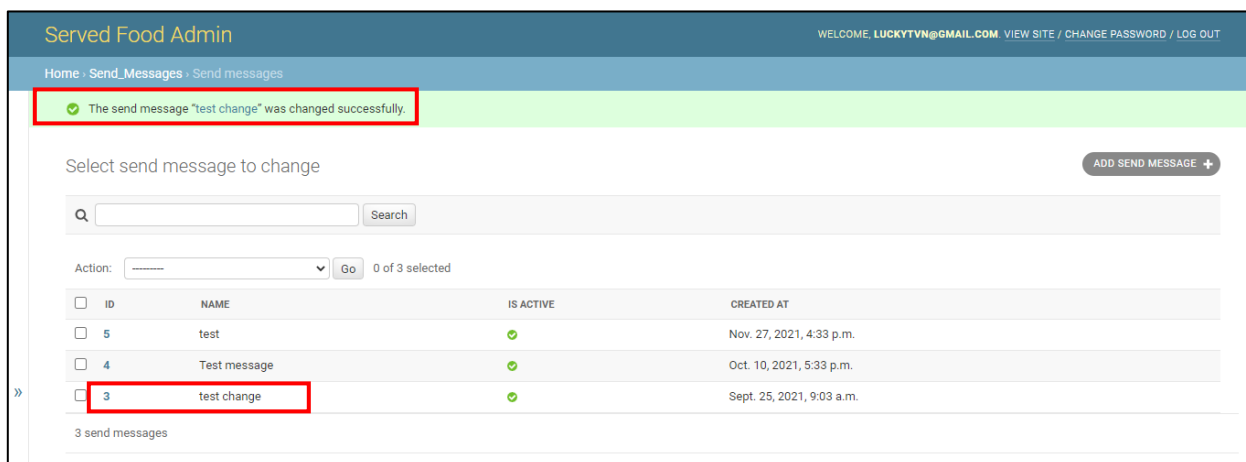
1.6.3. Giao diện chỉnh sửa Send Message

Để chỉnh sửa một bản ghi thông tin lời, người quản lý chọn một bản ăn để truy cập, giao diện chi tiết một lời nhắn như hình 52.



Hình 52 Giao diện chỉnh sửa lời nhắn

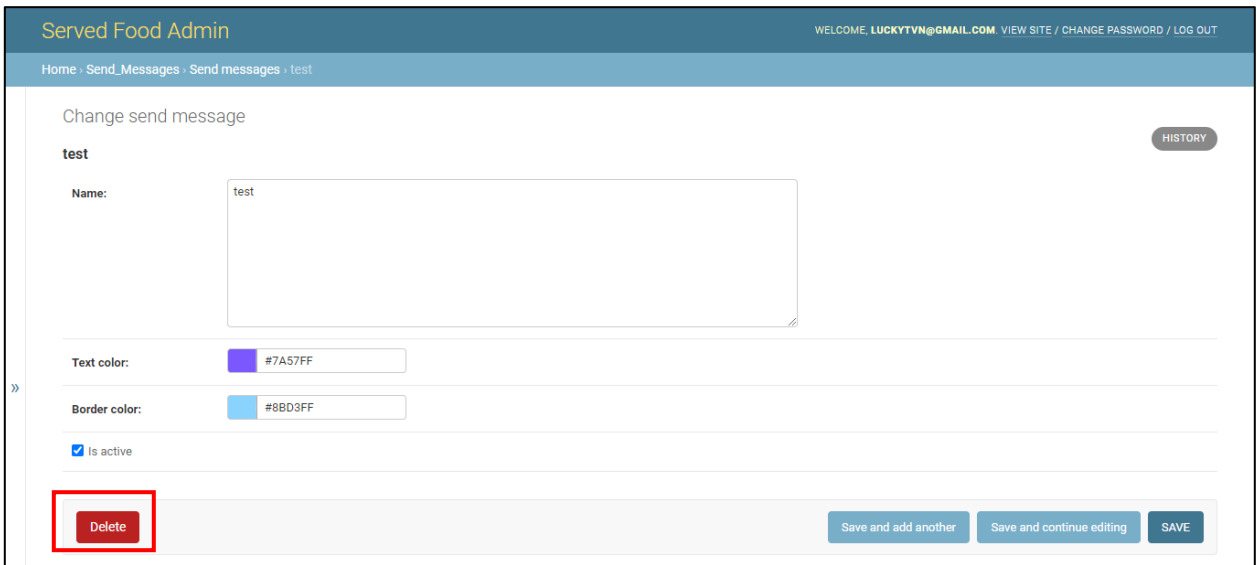
Sau khi chỉnh sửa các thông tin hợp lý, người quản lý chọn lưu và có website hiển thị kết quả như hình 53.



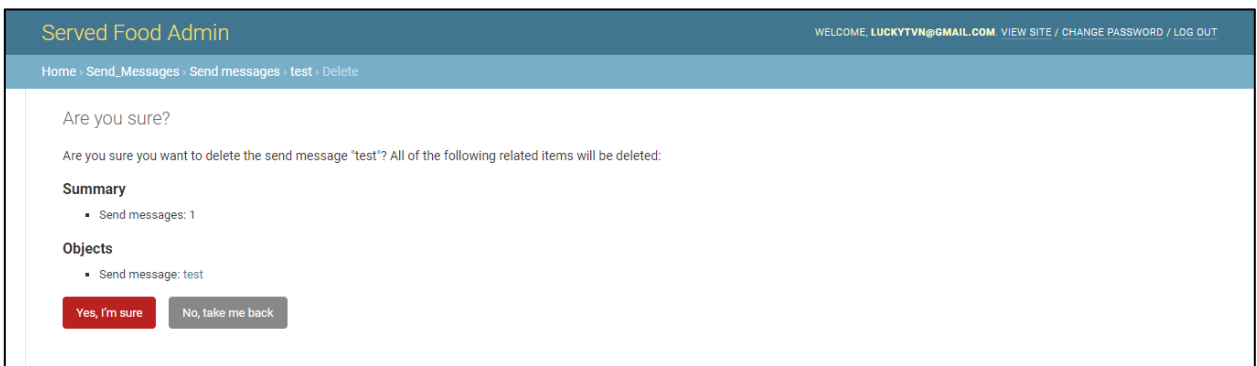
Hình 53 Giao diện chỉnh sửa lời nhắn thành công

1.6.4. Giao diện xóa Send Message

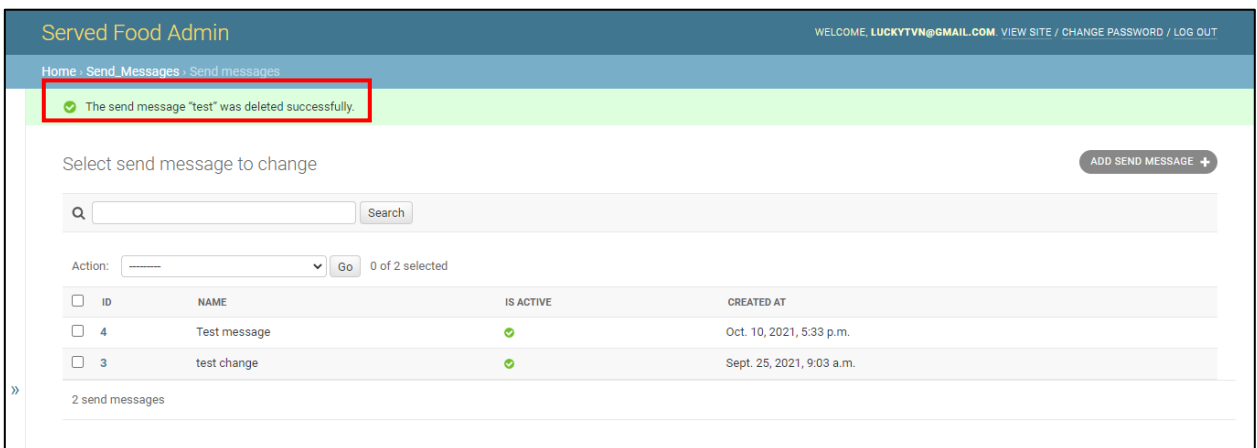
Để xóa một bản ghi Send Message, người quản lý truy cập vào bản ghi đó và chọn “Delete” như hình 54.



Hình 54 Giao diện xóa lời nhắn
Website yêu cầu người quản lý xác thực tác vụ xóa bản ghi như hình 55.



Hình 55 Giao diện xác nhận xóa lời nhắn
Người quản lý chọn “Yes, Im sure” để xác thực tác vụ xóa hoặc “No, take me back” để hủy yêu cầu. Giao diện xóa thành công như hình 56.



Hình 56 Giao diện xóa lời nhắn thành công

1.7. Giao diện Promotions (Chương trình khuyến mãi)

1.7.1. Giao diện chính Promotions

Người quản lý chọn danh mục Promotions tại danh sách các danh mục phía bên trái để truy cập vào danh mục. Giao diện hiển thị các thông tin gồm Name – Tên chương trình khuyến mãi; Description – Mô tả; Is Active – Trạng thái kích hoạt; Status – Trạng thái (ready – sẵn sàng : chưa áp dụng; in progress – trong tiến trình : đang áp dụng; stopped – ngưng : không áp dụng); Start At – Thời gian bắt đầu; End At – Thời gian kết thúc như hình 57.

The screenshot shows the 'Served Food Admin' interface. The top navigation bar includes 'Home - Promotions > Promotions' and a user greeting 'WELCOME LUCKYTVN@GMAIL.COM'. A sidebar on the left contains menu items: ORDERS, PRODUCTS, SEND_MESSAGES, and TABLES, each with an 'Add' button. The 'PROMOTIONS' section is highlighted in red. The main content area shows a notification: 'The promotion "Combo Thêm Bạn Thêm Ngon 2" was changed successfully.' Below this is a 'Select promotion to change' section with a search bar and an 'ADD PROMOTION +' button. A table lists four promotions with columns: NAME, DESCRIPTION, IS ACTIVE, IMAGE, STATUS, START AT, and END AT. The table data is as follows:

NAME	DESCRIPTION	IS ACTIVE	IMAGE	STATUS	START AT	END AT
Combo Thêm Bạn Thêm Ngon 3	Dành cho 5-6 người 01 Pizza ALC (cỡ lớn) 01 Xà Lách Trộn Cá Ngừ Và Thịt Xông Khô/ Bánh Cuốn Phở Mai/ Khoai Tây Chiên/ Bắp Nướng Phở Mai Thịt Xông Khô/ Com Chiên Hải Sản/ Mỹ Ý Thịt Xông Khô Xốt Kem/ Mỹ Ý Bò Bằm Xốt Cà Chua 01 Canh Gà Nướng BBQ (4 miếng)/ Canh Gà Chiên Xốt Cay Hàn Quốc (4 miếng)/ Canh Gà Chiên Xốt Trứng Mòn (4 miếng) 01 Chai Nước Ngọt Pepsi/ 7UP/ Mirinda 1.5L	Yes		ready	Dec. 1, 2021	Dec. 31, 2021
Combo Thêm Bạn Thêm Ngon 2	Dành cho 3-4 người 01 Pizza ALC (cỡ vừa) 01 Xà Lách Trộn Cá Ngừ Và Thịt Xông Khô/ Bánh Cuốn Phở Mai/ Khoai Tây Chiên/ Bắp Nướng Phở Mai Thịt Xông Khô 01 Com Chiên Hải Sản/ Mỹ Ý Thịt Xông Khô Xốt Kem/ Mỹ Ý Bò Bằm Xốt Cà Chua 02 Lon Nước Ngọt Pepsi/ 7UP/ Mirinda	Yes		in_progress	Nov. 27, 2021	Dec. 31, 2021
Combo Thêm Bạn Thêm Ngon 1	Dành cho 2 người 01 Pizza Truyền Thống (cỡ vừa) 01 Mỹ Ý Bò Bằm Xốt Cà Chua/ Xà Lách Trộn Cá Ngừ Và Thịt Xông Khô/ Bắp Nướng Phở Mai Thịt Xông Khô 01 Lon Pepsi/ 7UP/ Mirinda can	Yes		stopped	Sept. 17, 2021	Sept. 30, 2021
Mua 1 Được 3	Mua 1 TẶNG 2: Mua 01 bánh Pizza cỡ vừa được tặng ngay 01 Pepsi/7Up/Mirinda 0.5 L cùng 01 Pizza Phở Mai Cao Cấp (P)/ 01 Khoai Tây Chiên/ 01 Bánh Cuốn Phở Mai	Yes		stopped	Sept. 17, 2021	Oct. 31, 2021

Hình 57 Giao diện chương trình khuyến mãi

1.7.2. Giao diện thêm mới Promotion

Người quản lý chọn “ADD PROMOTION” để thêm mới một bàn ghi, giao diện như hình 58.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Promotions > Promotions > Add promotion

Add promotion

Name:

Is active

Description:

Image: Không có tệp nào được chọn

Start at: Today | 📅

End at: Today | 📅

Hình 58 Giao diện thêm mới chương trình khuyến mãi

Trường hợp người quản lý nhập chưa đầy đủ thông tin, website báo lỗi như hình 59.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Promotions > Promotions > Add promotion

Add promotion

Please correct the errors below.

This field is required.

Name:

Is active

This field is required.

Description:

This field is required.

Image: Không có tệp nào được chọn

This field is required.

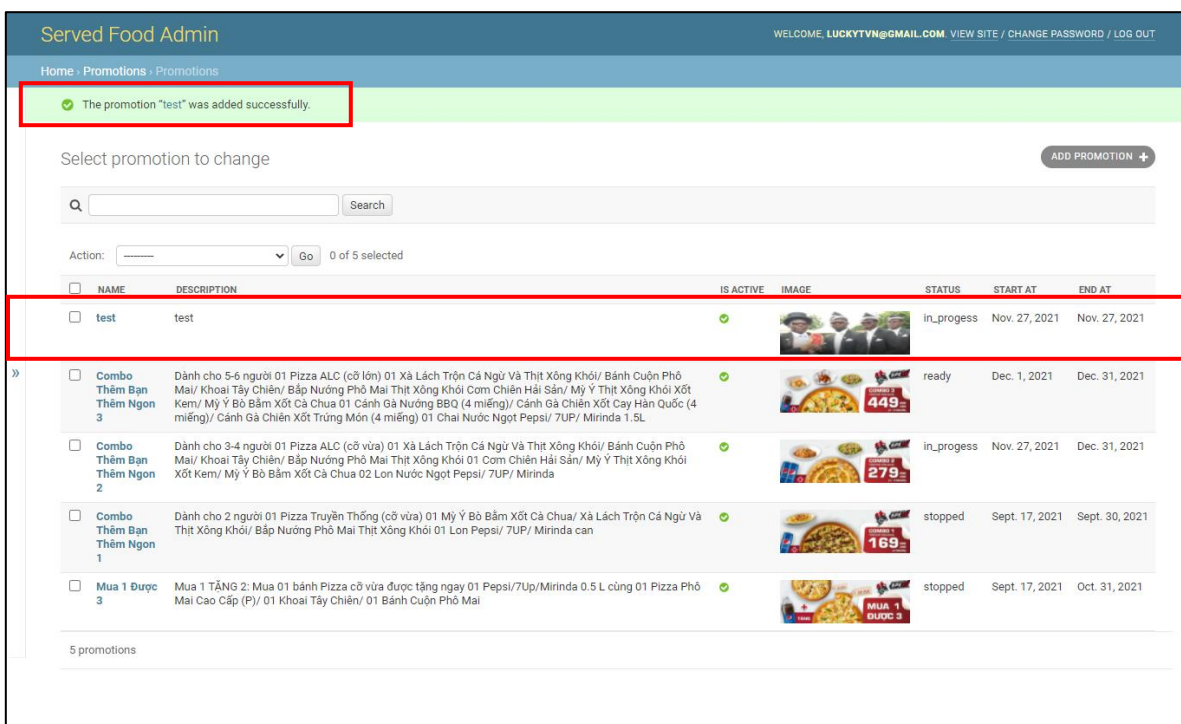
Start at: Today | 📅

This field is required.

End at: Today | 📅

Hình 59 Giao diện thêm mới chương trình khuyến mãi lỗi

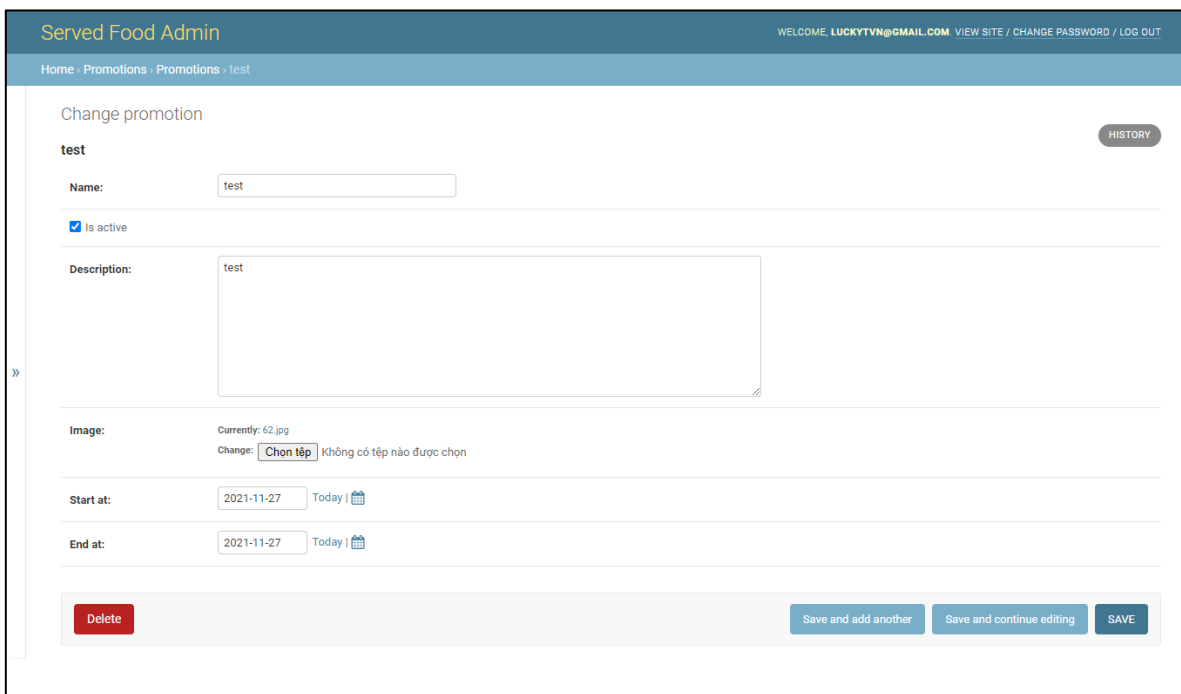
Nếu người quản lý nhập đầy đủ thông tin, website thông báo thêm thành công như hình 60.



Hình 60 Giao diện thêm mới chương trình khuyến mãi thành công

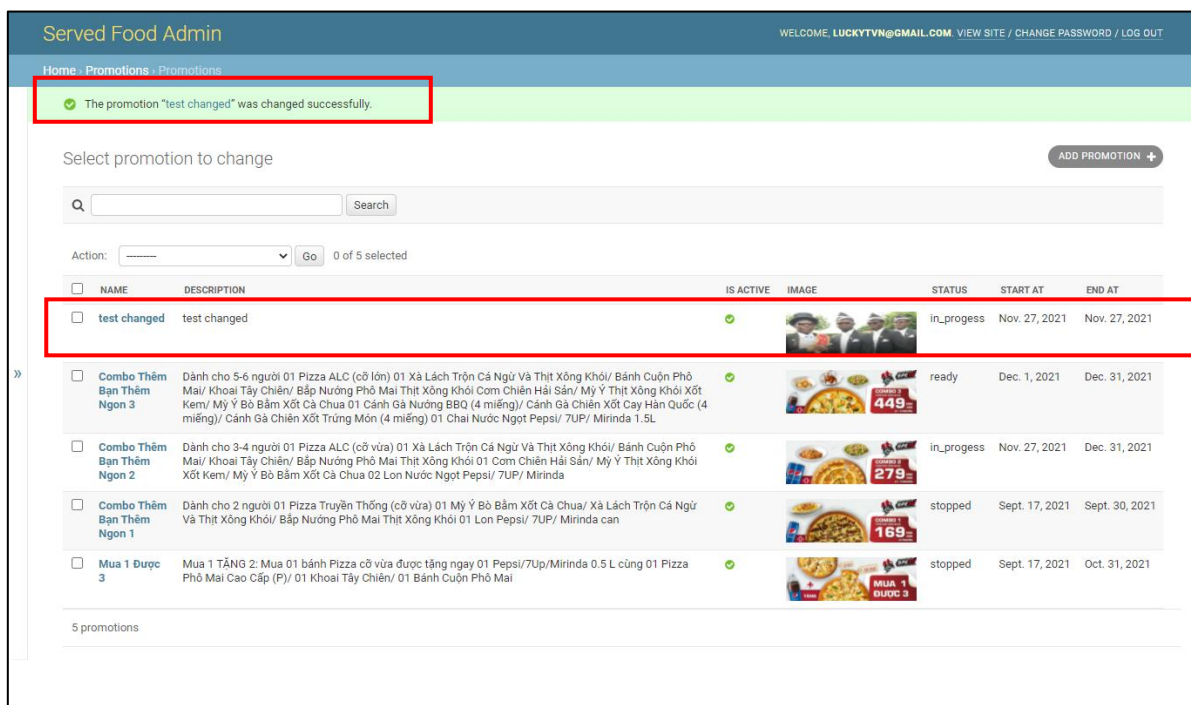
1.7.3. Giao diện chỉnh sửa Promotion

Để chỉnh sửa một bản ghi, người quản lý chọn một bản ghi Promotion, website hiển thị giao diện chi tiết như hình 61.



Hình 61 Giao diện chỉnh sửa chương trình khuyến mãi

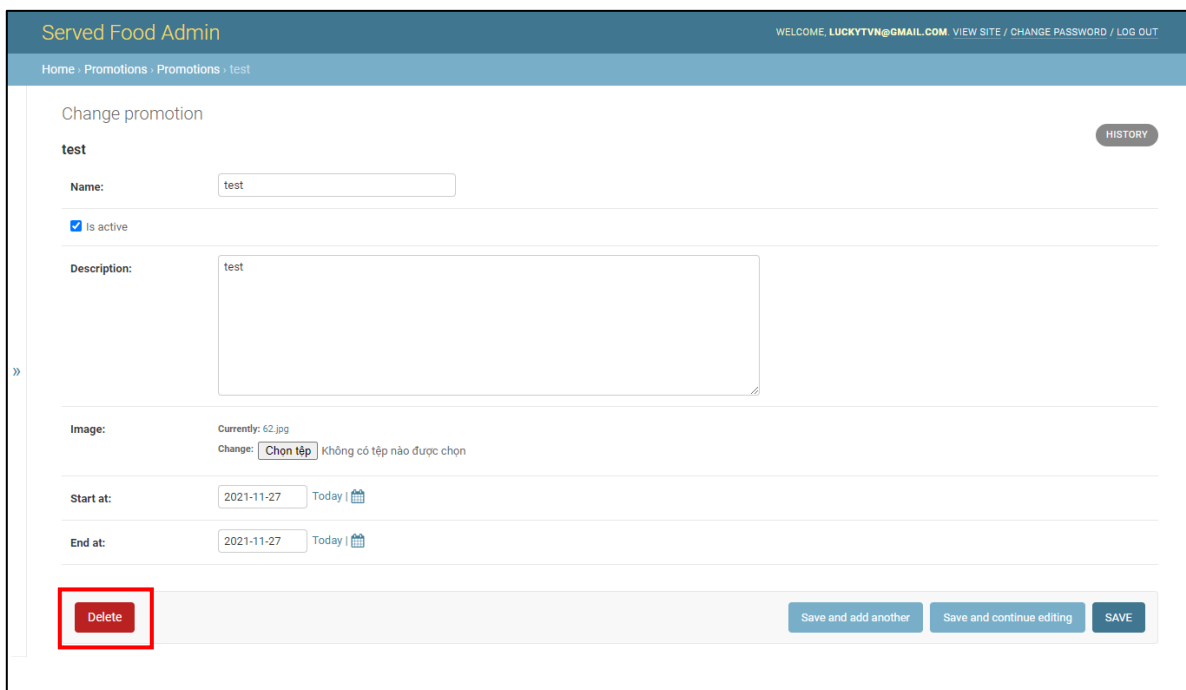
Sau khi chỉnh sửa các thông tin hợp lý, người quản lý chọn lưu. Hệ thống gửi thông báo thành công. Giao diện như hình 62.



Hình 62 Giao diện chỉnh sửa chương trình khuyến mãi thành công

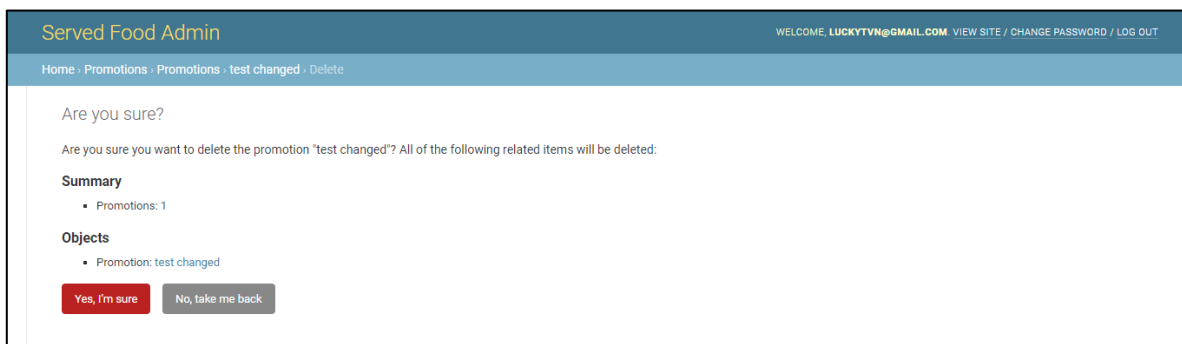
1.7.4. Giao diện xóa Promotion

Để thực hiện thao tác xóa, người quản lý chọn “Delete” như hình 63.



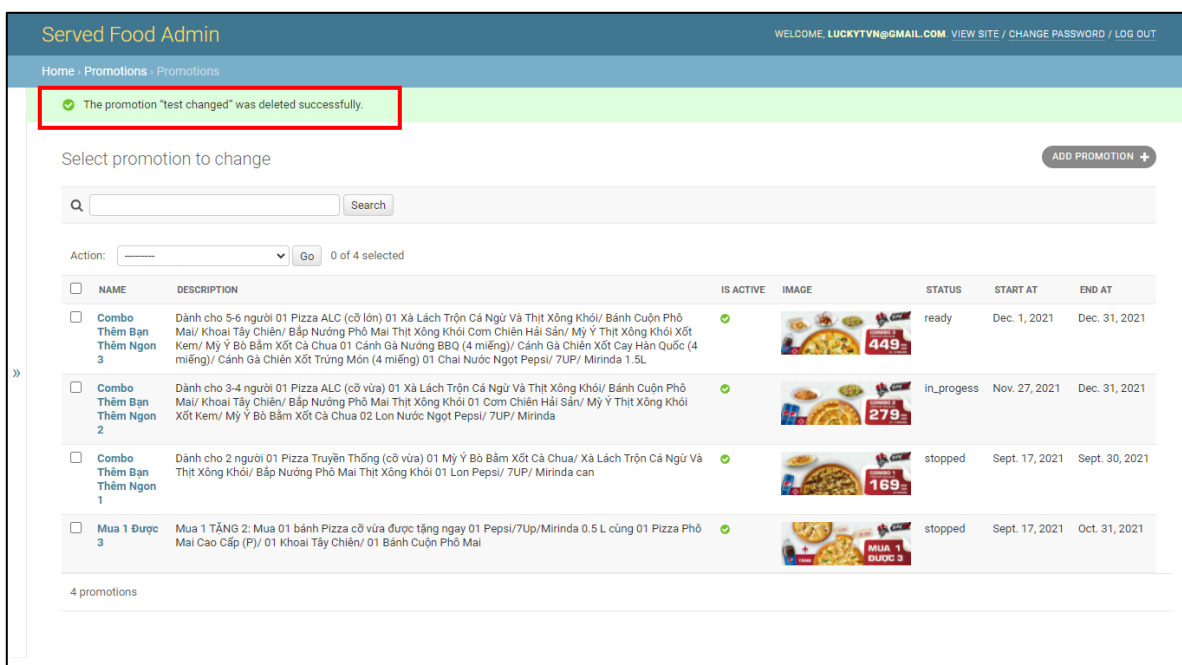
Hình 63 Giao diện xóa chương trình khuyến mãi

Hệ thống yêu cầu người quản lý xác nhận xóa như hình 64.



Hình 64 Giao diện xác nhận xóa chương trình khuyến mãi

Sau khi người quản lý xác nhận xóa, hệ thống xóa bản ghi khỏi cơ sở dữ liệu và hiển thị kết quả như hình 65.

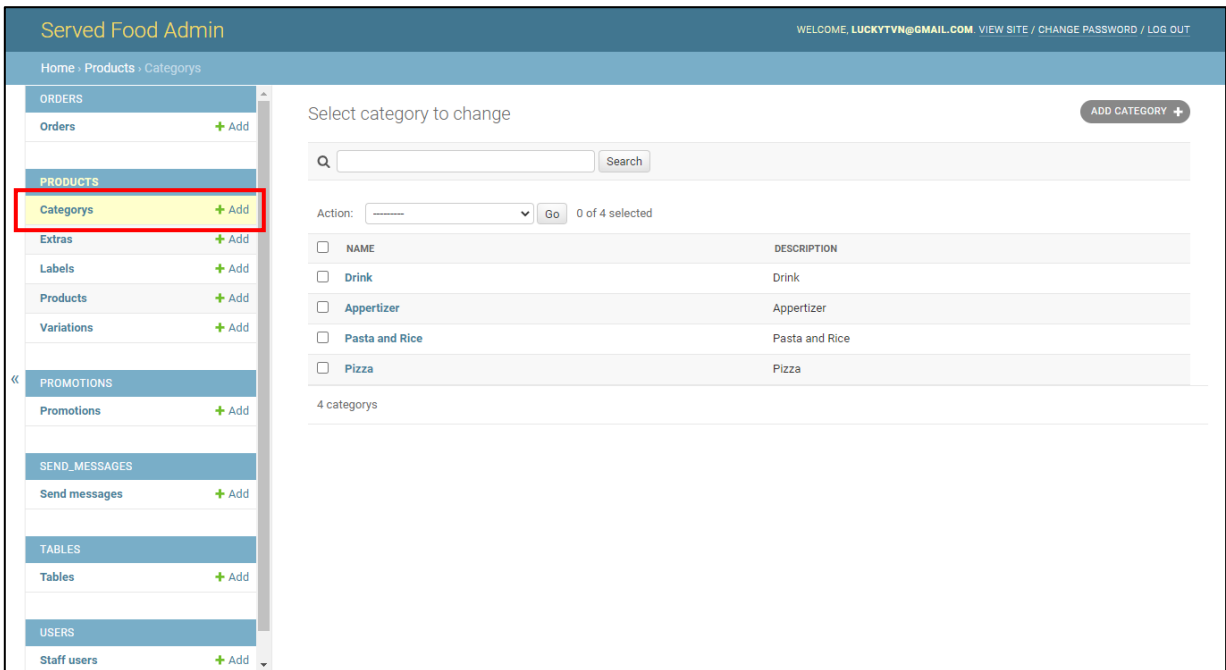


Hình 65 Giao diện xóa chương trình khuyến mãi thành công.

1.8. Giao diện Categories (Loại sản phẩm)

1.8.1. Giao diện chính Categories

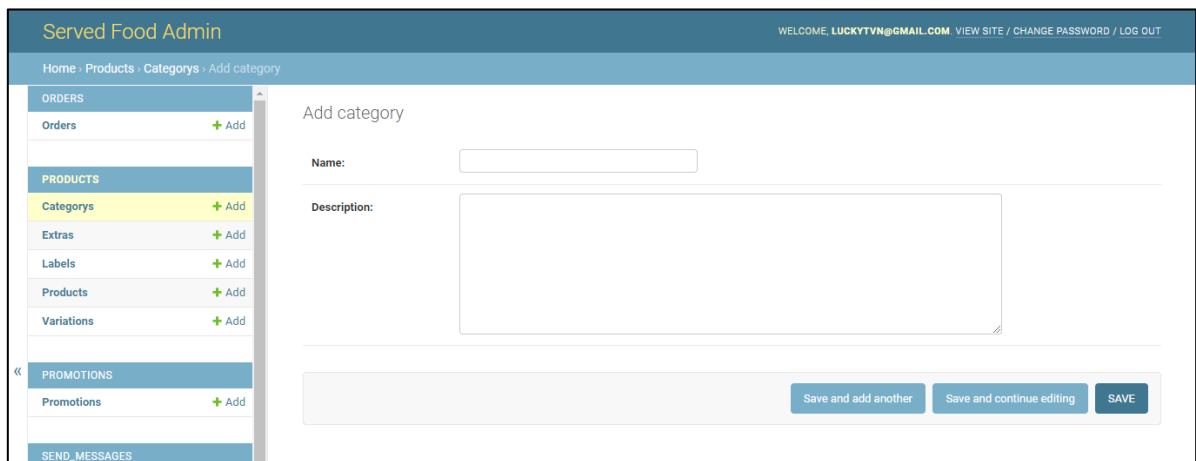
Người quản lý chọn danh mục Categories tại danh sách các danh mục bên trái, giao diện hiển thị như hình 66.



Hình 66 Giao diện loại sản phẩm

1.8.2. Giao diện thêm mới Category

Người quản lý chọn “ADD CATEGORY” để thêm mới một bản ghi. Giao diện như hình 67.



Hình 67 Giao diện thêm mới loại sản phẩm

Trong trường hợp người quản lý nhập chưa đầy đủ thông tin, website báo lỗi như hình 68.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Categorys > Add category

Add category

Please correct the errors below.

Name: This field is required.

Description: This field is required.

Save and add another Save and continue editing SAVE

Hình 68 Giao diện thêm mới loại sản phẩm lỗi

Khi người quản lý nhập đầy đủ thông tin và chính xác, giao diện thông báo thành công như hình 69.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Categorys

The category "test" was added successfully.

Select category to change ADD CATEGORY +

Q Search

Action: Go 0 of 5 selected

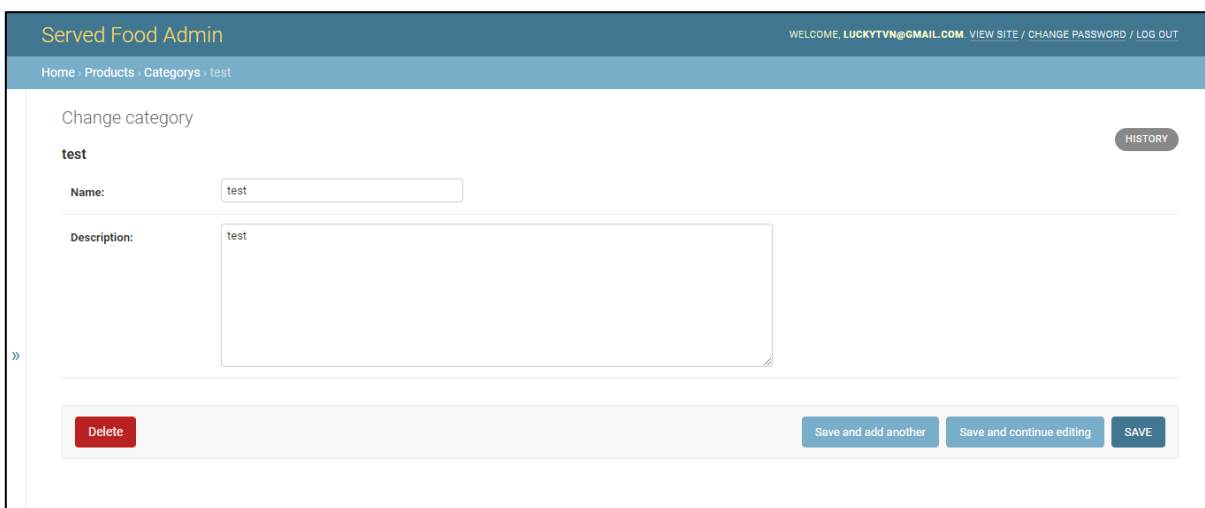
<input type="checkbox"/>	NAME	DESCRIPTION
<input type="checkbox"/>	test	test
<input type="checkbox"/>	Drink	Drink
<input type="checkbox"/>	Appertizer	Appertizer
<input type="checkbox"/>	Pasta and Rice	Pasta and Rice
<input type="checkbox"/>	Pizza	Pizza

5 categorys

Hình 69 Giao diện thêm mới loại sản phẩm thành công.

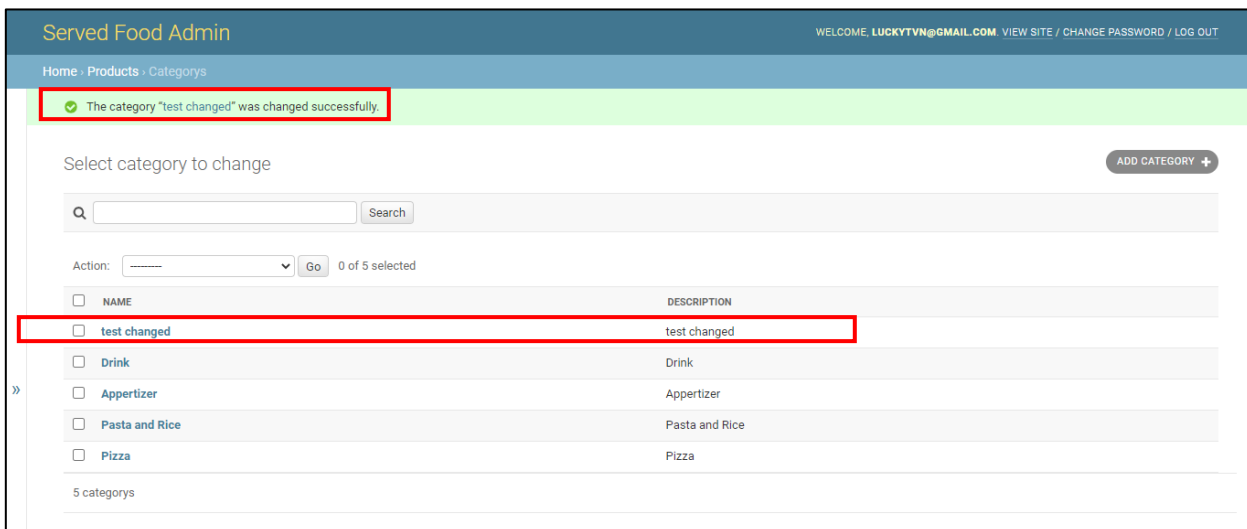
1.8.3. Giao diện chỉnh sửa Category

Để chỉnh sửa một bản ghi Category, người quản lý chọn một bản ghi và giao diện hiển thị như hình 70.



Hình 70 Giao diện chỉnh sửa loại sản phẩm

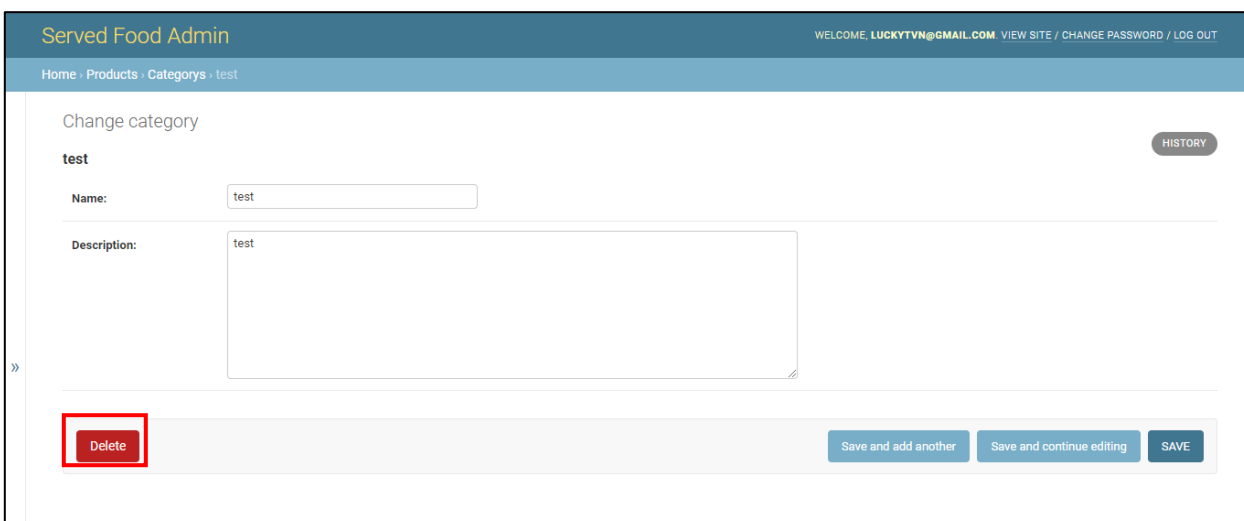
Sau khi chỉnh sửa thông tin, người quản lý chọn lưu và website gửi về kết quả như hình 71.



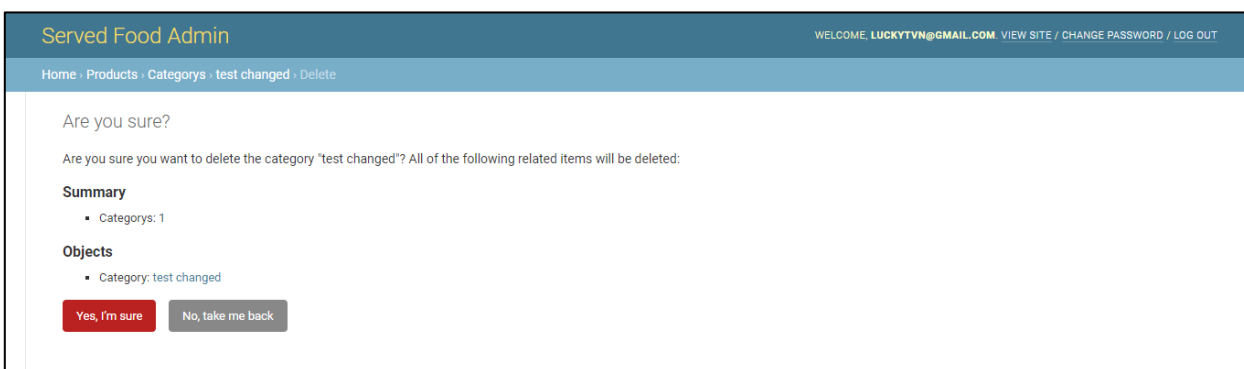
Hình 71 Giao diện chỉnh sửa loại sản phẩm thành công

1.8.4. Giao diện xóa Category

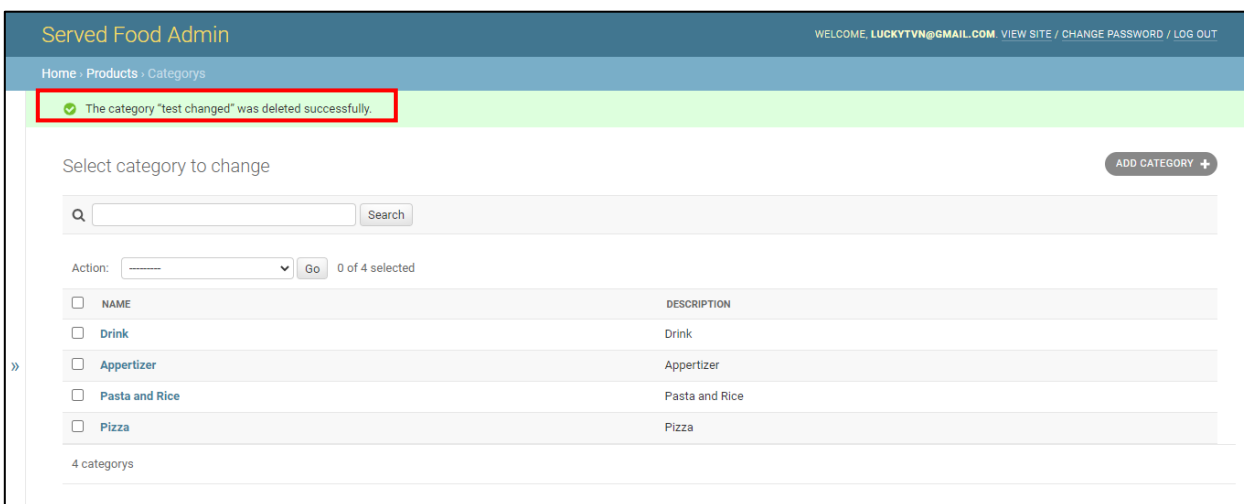
Để xóa một bản ghi Category, người quản lý chọn “Delete” như hình 72.



Hình 72 Giao diện xóa loại sản phẩm
Sau đó, website yêu cầu người quản lý xác nhận tác vụ xóa như hình 73.



Hình 73 Giao diện xác nhận xóa loại sản phẩm
Sau khi người quản lý xác nhận, hệ thống xóa bản ghi khỏi cơ sở dữ liệu và gửi về kết quả như hình 74.

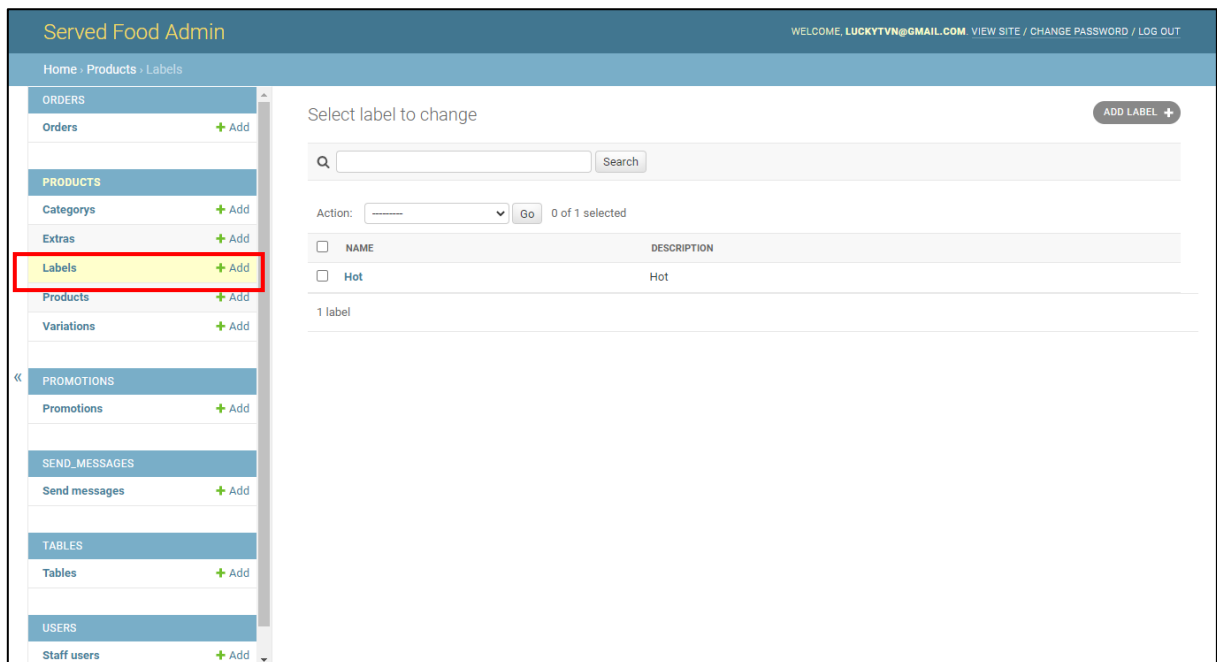


Hình 74 Giao diện xóa loại sản phẩm thành công

1.9. Giao diện Labels (Nhãn sản phẩm)

1.9.1. Giao diện chính

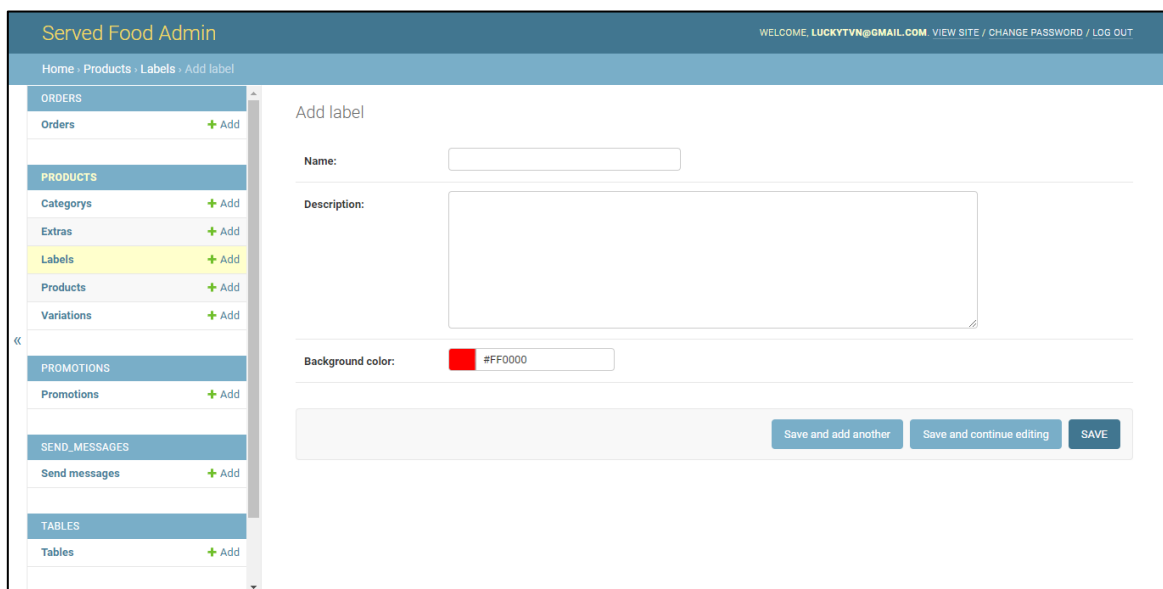
Người quản lý truy cập danh mục Labels bằng cách chọn danh mục Labels tại danh sách các danh mục bên trái. Giao diện như hình 75.



Hình 75 Giao diện nhãn sản phẩm

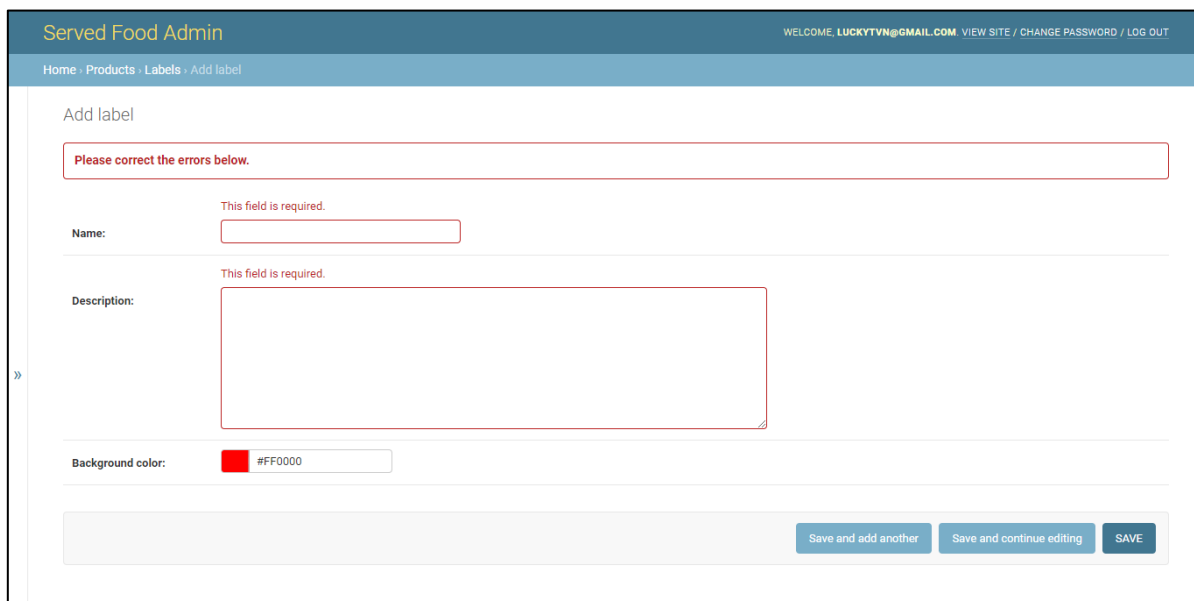
1.9.2. Giao diện thêm Label

Để thêm một bản ghi Label, người quản lý chọn “ADD LABEL”, giao diện hiển thị như hình 76.



Hình 76 Giao diện thêm mới nhãn sản phẩm

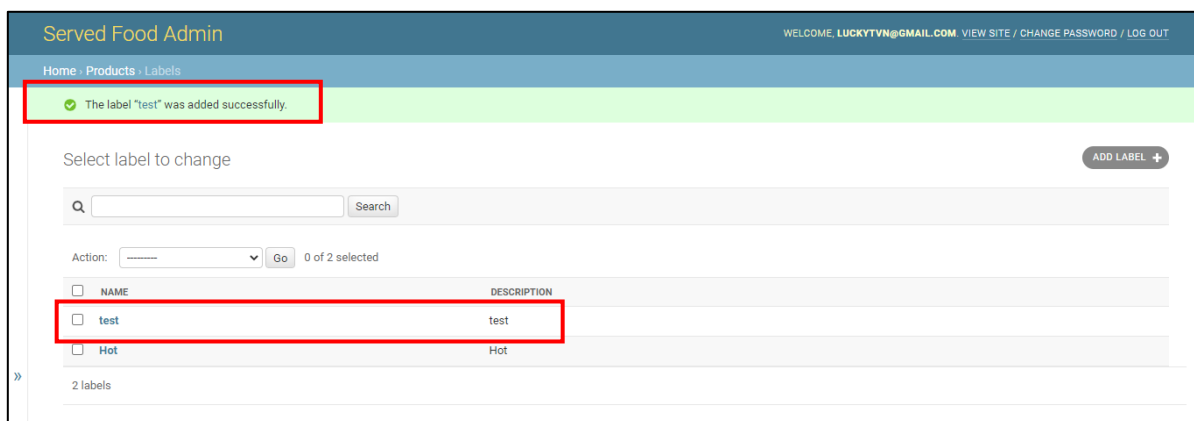
Trường hợp người quản lý nhập thiếu thông tin, website báo lỗi như hình 77.



The screenshot shows the 'Add label' form in the Served Food Admin interface. The form has three main sections: 'Name', 'Description', and 'Background color'. The 'Name' field is empty and has a red border with the error message 'This field is required.' above it. The 'Description' field is also empty and has a red border with the error message 'This field is required.' above it. The 'Background color' field is set to '#FF0000' with a red color swatch. At the bottom right, there are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'. A red box highlights the error messages.

Hình 77 Giao diện thêm mới nhãn sản phẩm lỗi

Khi người quản lý nhập đầy đủ thông tin và nhấn lưu, hệ thống lưu bản ghi vào cơ sở dữ liệu và gửi về kết quả như hình 78.



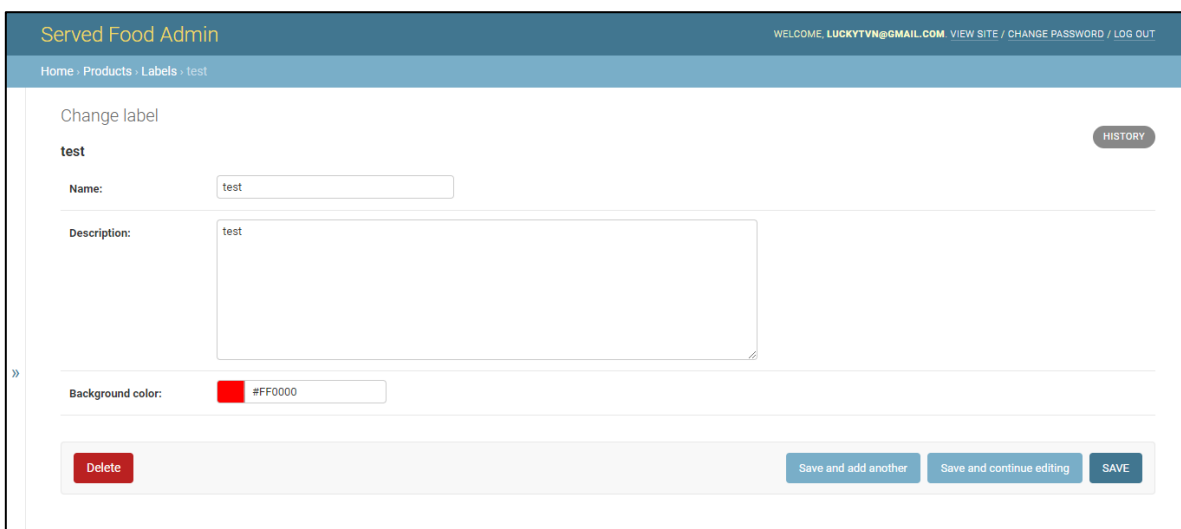
The screenshot shows the 'Labels' list in the Served Food Admin interface. A green success message is displayed at the top: 'The label "test" was added successfully.' Below this, there is a search bar and a table of labels. The table has two columns: 'NAME' and 'DESCRIPTION'. The first row is 'test' and the second row is 'Hot'. The 'test' row is highlighted with a red box. At the bottom, it says '2 labels'.

NAME	DESCRIPTION
test	test
Hot	Hot

Hình 78 Giao diện thêm mới nhãn sản phẩm thành công

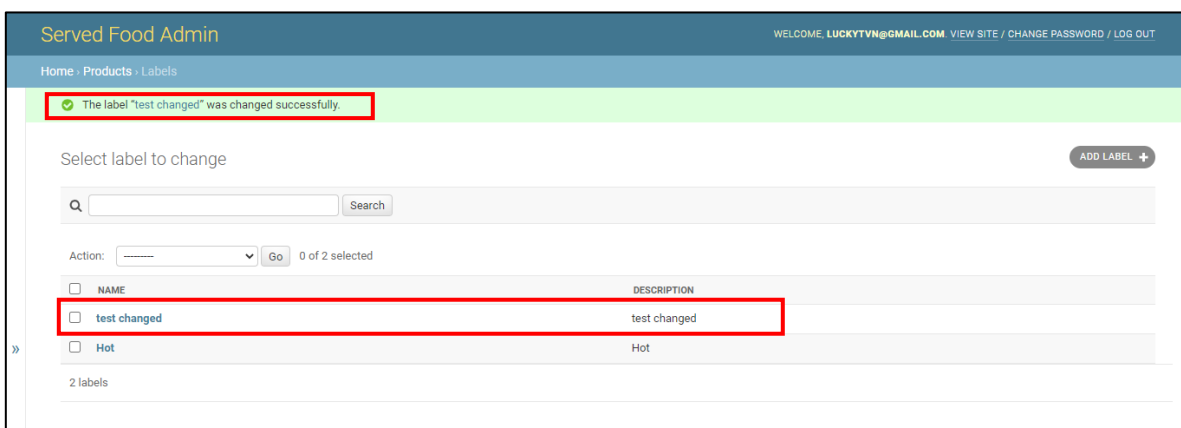
1.9.3. Giao diện chỉnh sửa Label

Để chỉnh sửa một bản ghi, người quản lý truy cập vào bản ghi. Giao diện như hình 79.



Hình 79 Giao diện chỉnh sửa nhãn sản phẩm

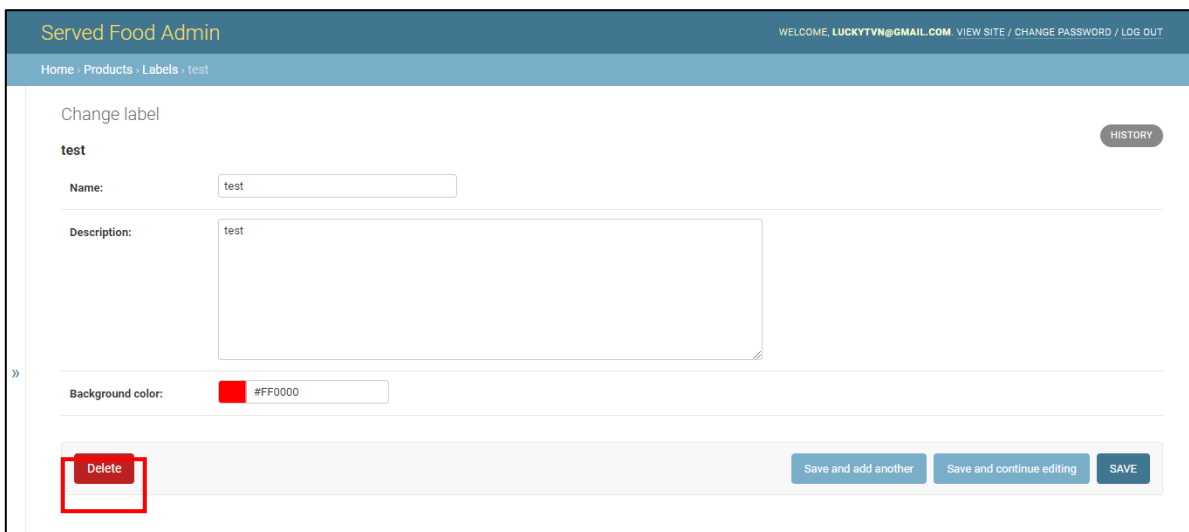
Người quản lý chỉnh sửa thông tin và nhấn lưu, kết quả gửi về thành công như hình 80.



Hình 80 Giao diện chỉnh sửa nhãn sản phẩm thành công

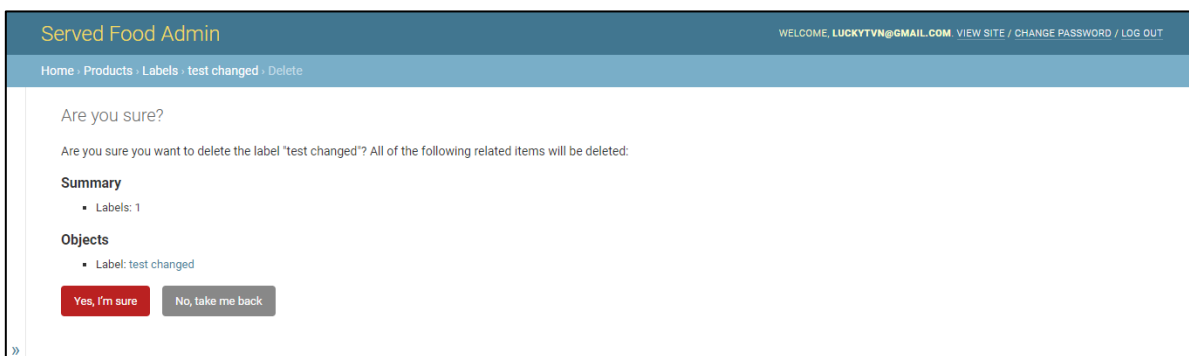
1.9.4. Giao diện xóa Label

Để xóa một bản ghi, người quản lý chọn “Delete” như hình 81.



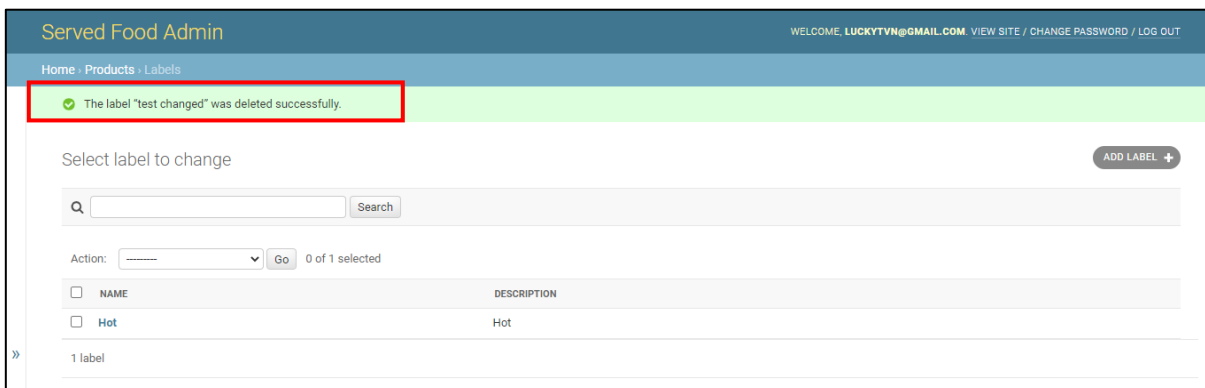
Hình 81 Giao diện xóa nhãn sản phẩm

Sau đó, hệ thống yêu cầu người quản lý xác minh tác vụ xóa như hình 82.



Hình 82 Giao diện xác nhận xóa nhãn sản phẩm

Sau khi người quản lý chọn “Yes, Im sure”, hệ thống xóa bản ghi khỏi cơ sở dữ liệu và trả về kết quả như hình 83.

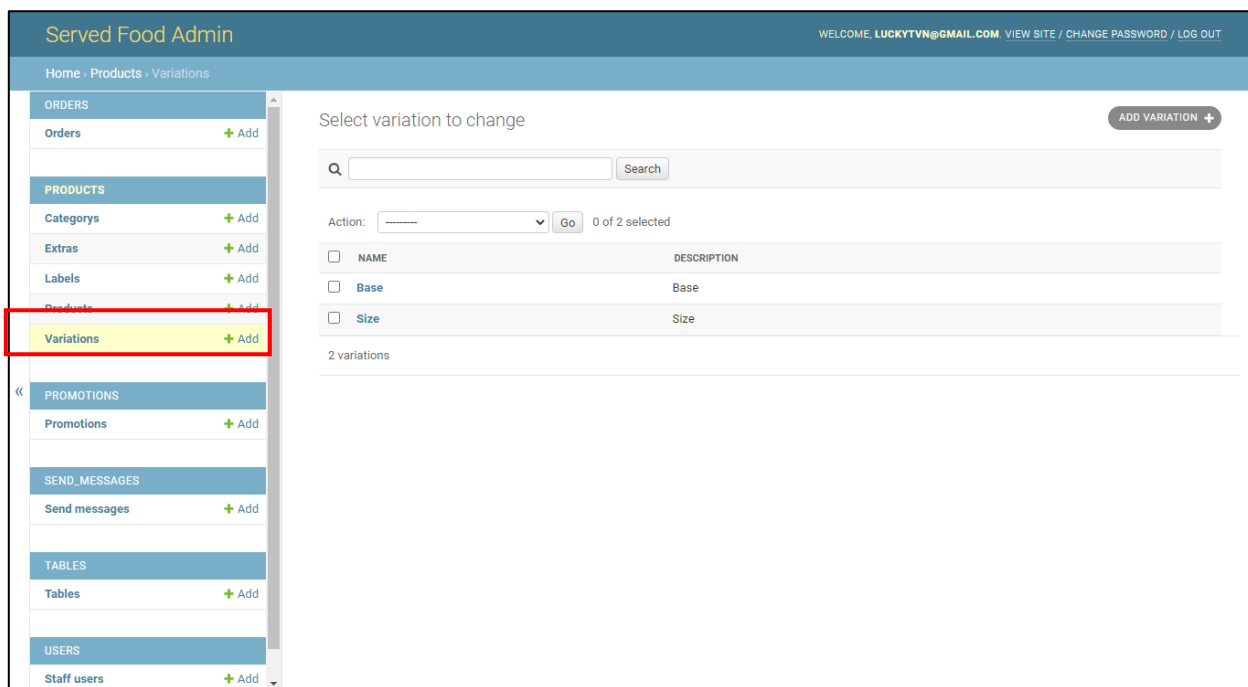


Hình 83 Giao diện xóa nhãn sản phẩm thành công

1.10. Giao diện Variations (Biến thể sản phẩm)

1.10.1. Giao diện chính Variations

Người quản lý truy cập vào danh mục Variations bằng cách chọn danh mục Variations tại danh sách các danh mục phía bên trái. Giao diện chính như hình 84.



Hình 84 Giao diện biến thể sản phẩm

1.10.2. Giao diện thêm mới Variation

Người quản lý thêm mới một bản ghi Variation bằng cách chọn “ADD VARIATION”, giao diện hiển thị như hình 85.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Variations > Add variation

Add variation

Name:

Description: 1

Price affected

VARIATION OPTIONS

NAME	DESCRIPTION	DELETE?
<input type="text"/>	<div style="border: 1px solid #ccc; height: 60px; width: 100%;"></div>	<input type="button" value="X"/>
<input type="text"/>	<div style="border: 1px solid #ccc; height: 60px; width: 100%;"></div>	<input type="button" value="X"/> 2
<input type="text"/>	<div style="border: 1px solid #ccc; height: 60px; width: 100%;"></div>	<input type="button" value="X"/>

Hình 85 Giao diện thêm mới biến thể sản phẩm

Giao diện hình 85 được chia thành 2 khối:

1. Thông tin chung: Tên của loại biến thể, mô tả loại biến thể
2. Các biến thể: Một loại biến thể có thể có một, nhiều hoặc không có biến thể. Mỗi biến thể bao gồm các thông tin: Tên biến thể, mô tả biến thể
 - Kí hiệu dấu X phía bên phải mỗi form biến thể dùng để xóa một form biến thể
 - “Add another Variation option”: dùng để thêm một form biến thể

Trường hợp người quản lý nhập không đầy đủ thông tin, website báo lỗi như hình 86.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Products > Variations > Add variation

Add variation

Please correct the errors below.

Name: This field is required.

Description:This field is required.

Price affected

VARIATION OPTIONS		
NAME	DESCRIPTION	DELETE?
<input type="text"/>	<div style="border: 1px solid #ccc; height: 60px;"></div>	✕
<input type="text"/>	<div style="border: 1px solid #ccc; height: 60px;"></div>	✕

[+ Add another Variation option](#)

Hình 86 Giao diện thêm mới biến thể sản phẩm lỗi

Một loại biến thể không bắt buộc phải có biến thể, nhưng trong trường hợp người quản lý nhập một biến thể nếu không đầy đủ thông tin website sẽ báo lỗi như hình 87.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Variations > Add variation

Add variation

Please correct the errors below.

Name:

Description:

Price affected

VARIATION OPTIONS

NAME	DESCRIPTION	DELETE?
<input type="text" value="test1"/>	This field is required.	<input type="button" value="✖"/>
<input type="text" value="test2"/>	This field is required.	<input type="button" value="✖"/>

[+ Add another Variation option](#)

Hình 87 Giao diện thêm mới biến thể sản phẩm lỗi 2

Sau khi nhập đầy đủ thông tin, hệ thống lưu các bản ghi vào cơ sở dữ liệu và gửi về kết quả như hình 88.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Variations

The variation "test" was added successfully.

Select variation to change

Action: 0 of 3 selected

<input type="checkbox"/>	NAME	DESCRIPTION
<input type="checkbox"/>	test	test
<input type="checkbox"/>	Base	Base
<input type="checkbox"/>	Size	Size

3 variations

Hình 88 Giao diện thêm mới biến thể sản phẩm thành công

1.10.3. Giao diện chỉnh sửa Variation

Để chỉnh sửa một bản ghi, người quản lý cần truy cập vào bản ghi. Giao diện như hình 89.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Variations > test

Change variation HISTORY

test

Name: test

Description: test

Price affected

VARIATION OPTIONS

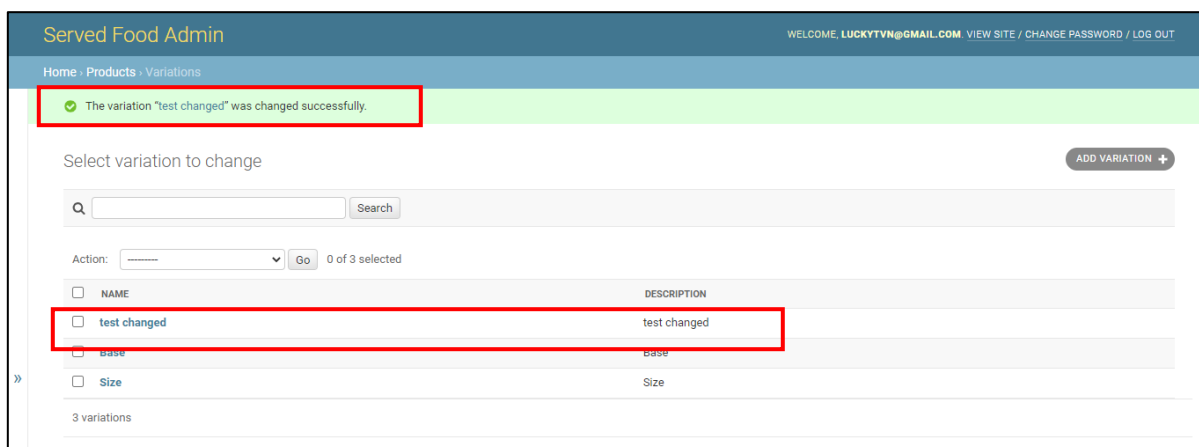
NAME	DESCRIPTION	DELETE?
test1	test1	<input type="checkbox"/>
test2	test2	<input type="checkbox"/>

+ Add another Variation option

Delete Save and add another Save and continue editing SAVE

Hình 89 Giao diện chỉnh sửa biến thể sản phẩm

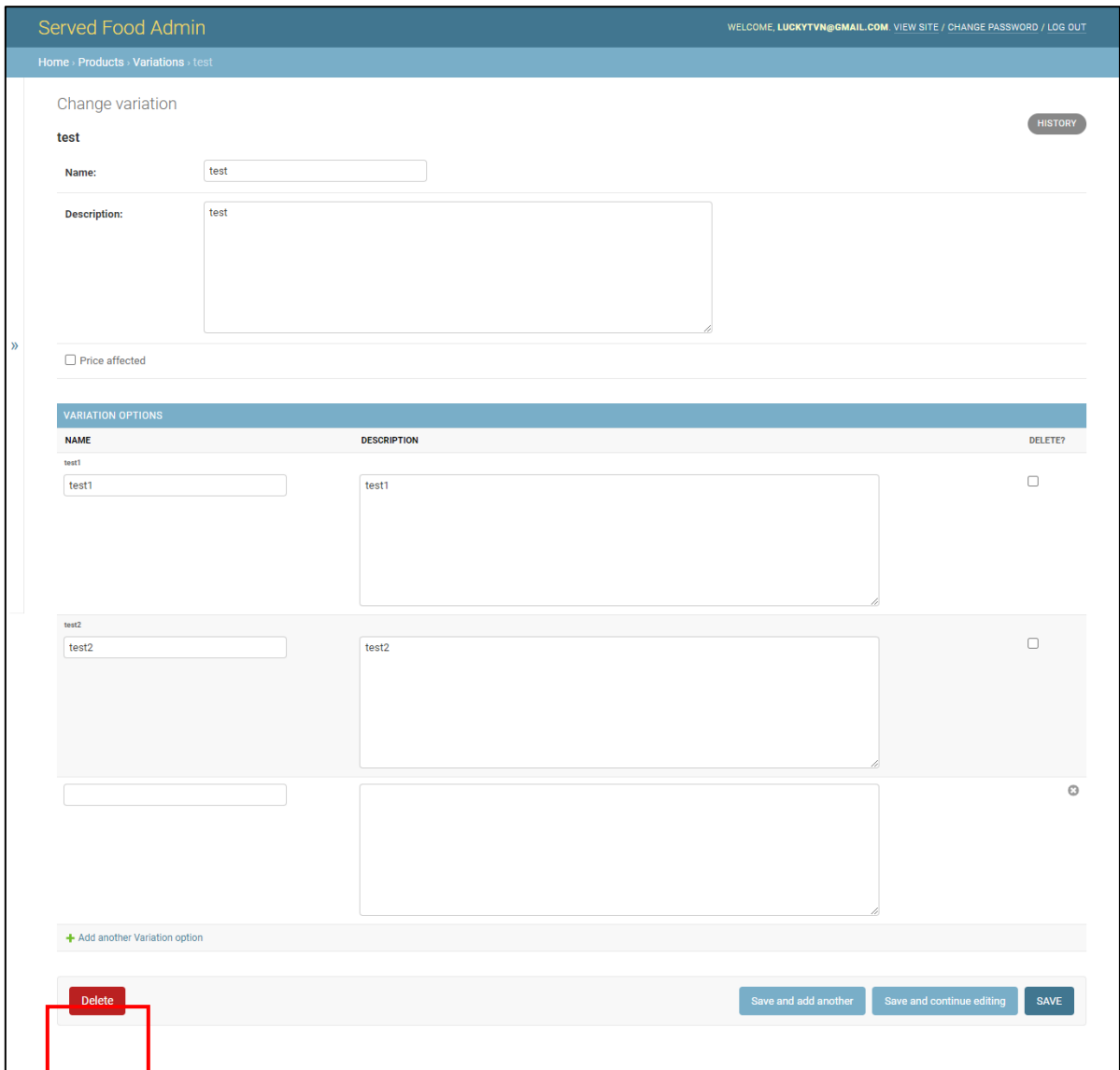
Sau khi điều chỉnh thông tin, người quản lý chọn lưu. Website gửi thông báo thành công như hình 90.



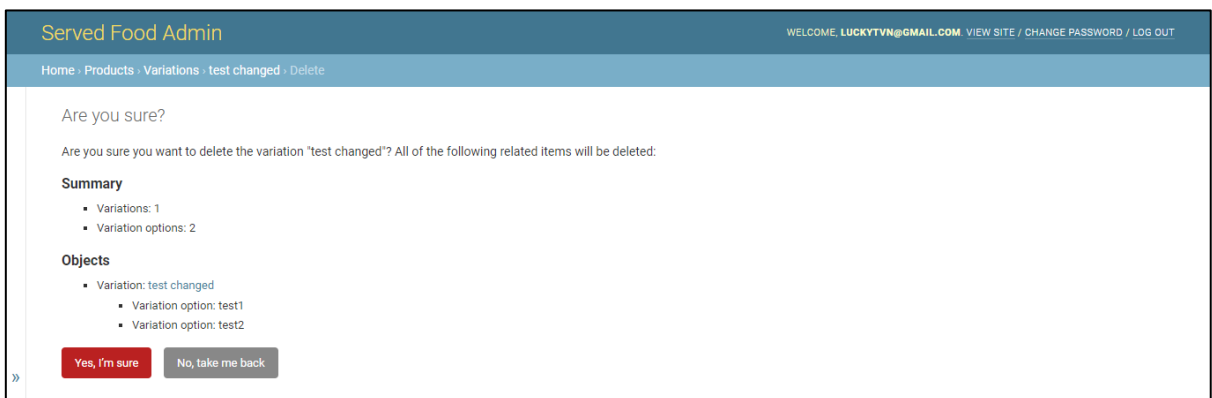
Hình 90 Giao diện thêm mới biến thể sản phẩm thành công

1.10.4. Giao diện xóa Variation

Để xóa một bản ghi, người quản lý cần truy cập bản ghi và chọn “Delete” như hình 91.

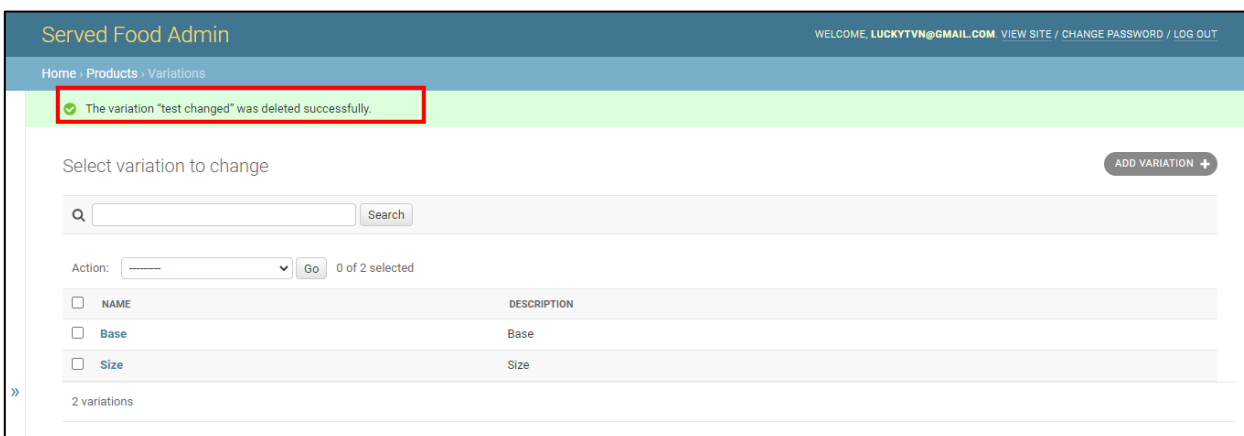


Hình 91 Giao diện xóa biến thể sản phẩm
Sau đó, hệ thống yêu cầu người quản lý xác minh tác vụ xóa như hình 92.



Hình 92 Giao diện xác nhận xóa biến thể sản phẩm

Sau khi người quản lý chọn “Yes, Im sure” để xác minh tác vụ, hệ thống xóa bản ghi khỏi cơ sở dữ liệu và gửi về kết quả như hình 93.

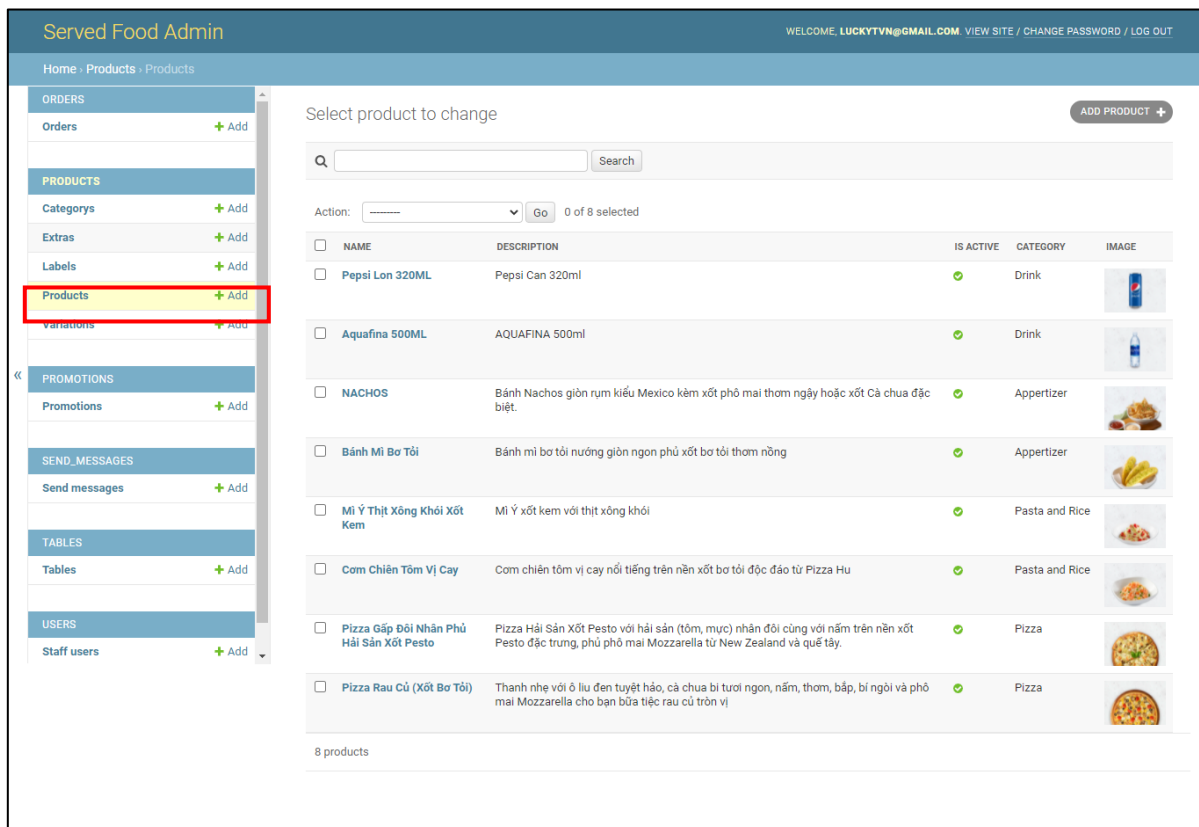


Hình 93 Giao diện xóa biến thể sản phẩm thành công

1.11. Giao diện Products (Sản phẩm)

1.11.1. Giao diện chính Products

Để truy cập giao diện danh mục Products, người quản lý cần chọn danh mục Products tại danh sách các danh mục phía bên trái. Giao diện danh mục Products như hình 94.



Hình 94 Giao diện sản phẩm

1.11.2. Giao diện thêm mới Product

Người quản chọn “ADD PRODUCT” để thêm một bản ghi product, giao diện như hình 95.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Products > Add product

Add product

Name:

Is active

Description:

Category: ✎ +

Labels: +
Hold down "Control", or "Command" on a Mac, to select more than one.

Variations: +
Hold down "Control", or "Command" on a Mac, to select more than one.

Image: Không có tệp nào được chọn

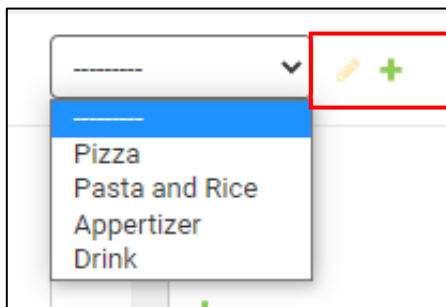
PRODUCT VARIATION OPTIONS

VARIATION OPTIONS	PRICE	DELETE?
<input type="text" value="Small"/> <input type="text" value="Medium"/> <input type="text" value="Large"/> <input type="text" value="Pan"/> <input type="text" value="Handtoss"/> <input type="text" value="Thin"/>	<input type="text"/>	<input type="button" value="✕"/>
<input type="text" value="Small"/> <input type="text" value="Medium"/> <input type="text" value="Large"/> <input type="text" value="Pan"/> <input type="text" value="Handtoss"/> <input type="text" value="Thin"/>	<input type="text"/>	<input type="button" value="✕"/>
<input type="text" value="Small"/> <input type="text" value="Medium"/> <input type="text" value="Large"/> <input type="text" value="Pan"/> <input type="text" value="Handtoss"/> <input type="text" value="Thin"/>	<input type="text"/>	<input type="button" value="✕"/>

+ Add another Product variation option

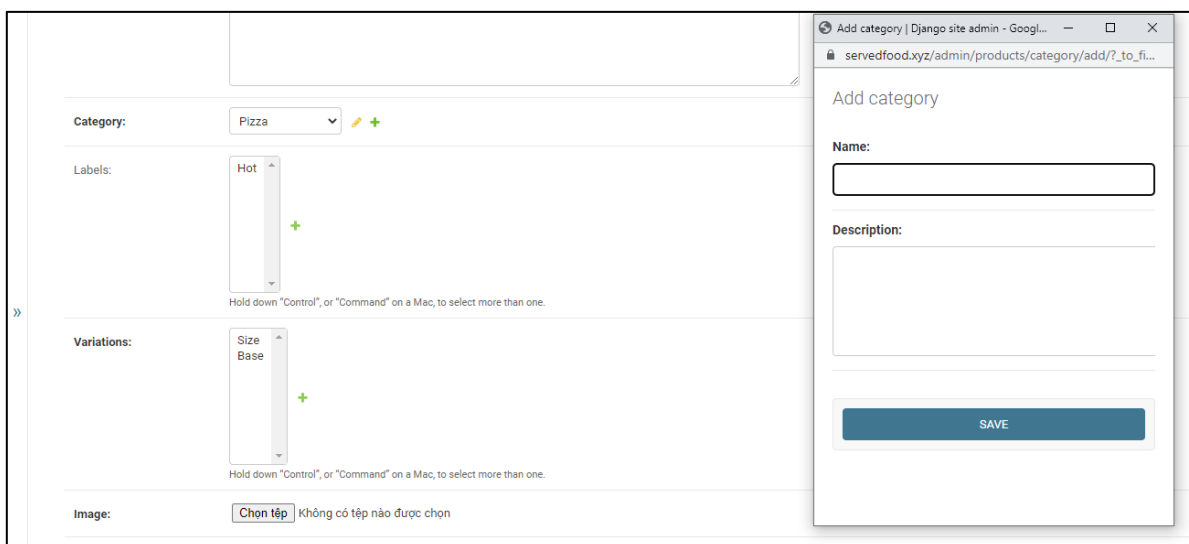
Hình 95 Giao diện thêm mới sản phẩm

- **Category:** Loại sản phẩm được nhập và thêm tại danh mục Categories, mỗi sản phẩm thuộc một loại sản phẩm. Sự lựa chọn được thể hiện như hình 96.



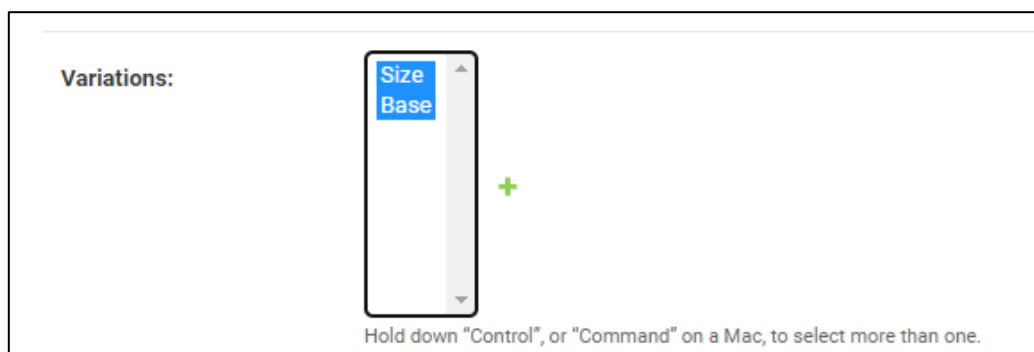
Hình 96 Giao diện lựa chọn loại sản phẩm

- Kí hiệu bút màu vàng: Cho phép chỉnh sửa bản ghi category đang được chọn ngay cửa sổ hiện tại như hình 97.
- Kí hiệu dấu cộng màu xanh: Cho phép thêm mới một bản ghi category ngay cửa sổ hiện tại như hình 97.



Hình 97 Giao diện thêm/chỉnh sửa loại sản phẩm trực tiếp

Đối với các trường thông tin có nhiều sự lựa chọn, người quản lý có thể nhấn giữ “Control” đối với máy hệ điều hành Window hay “Command” đối với máy Mac và chọn nhiều lựa chọn như hình 98.



Hình 98 Giao diện đa sự lựa chọn

PRODUCT VARIATION OPTIONS		
VARIATION OPTIONS	PRICE	DELETE?
<ul style="list-style-type: none"> Small Medium Large Pan Handtoss Thin 	<input type="text"/>	✕
<ul style="list-style-type: none"> Small Medium Large Pan Handtoss Thin 	<input type="text"/>	✕
<ul style="list-style-type: none"> Small Medium Large Pan Handtoss Thin 	<input type="text"/>	✕

Hình 99 Giao diện sản phẩm biến thể

Hình 99 thể hiện khối Product Variation Options – Sản phẩm biến thể. Một sản phẩm có thể có một hoặc nhiều biến thể, đối với mỗi biến thể có một giá riêng.

Trường hợp người quản lý nhập không đầy đủ thông tin, website báo lỗi như hình 100.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Products > Add product

Add product

Please correct the errors below.

Name: This field is required.

Is active

Description: This field is required.

Category: Pizza

Labels: Hot

Hold down "Control", or "Command" on a Mac, to select more than one.

Variations: Size Base

Hold down "Control", or "Command" on a Mac, to select more than one.

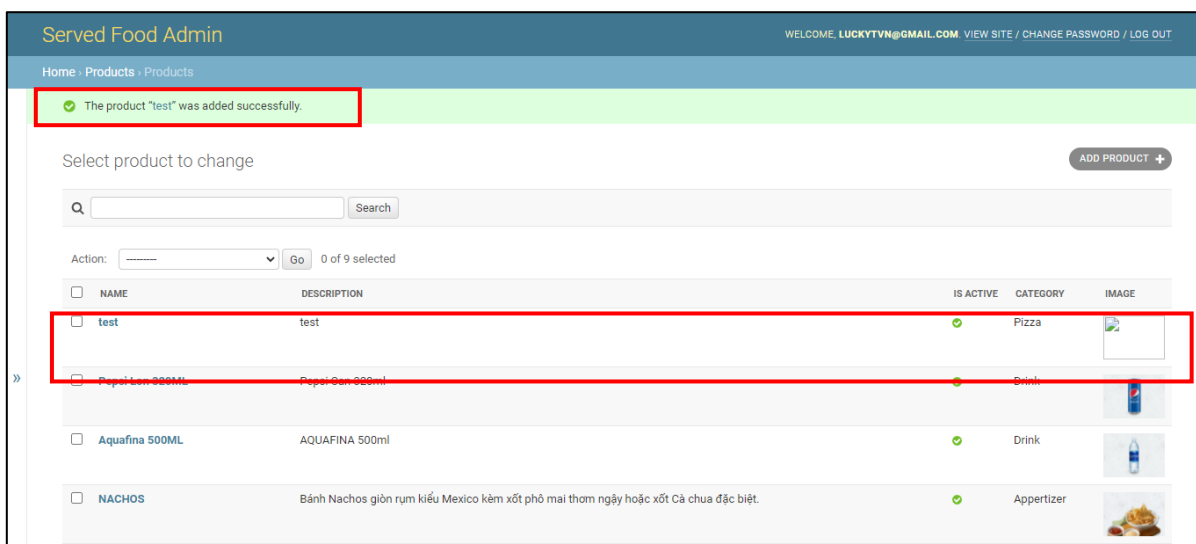
Image: không có tệp nào được chọn

PRODUCT VARIATION OPTIONS		
VARIATION OPTIONS	PRICE	DELETE?
<ul style="list-style-type: none"> Small Medium Large Pan Handtoss Thin 	<input type="text"/>	
<ul style="list-style-type: none"> Small Medium Large Pan Handtoss Thin 	<input type="text"/>	

[+ Add another Product variation option](#)

Hình 100 Giao diện thêm mới sản phẩm lỗi

Sau khi nhập đầy đủ thông tin, người quản lý chọn lưu. Hệ thống lưu bản ghi vào cơ sở hệ thống và gửi về kết quả như hình 101.



Hình 101 Giao diện thêm sản phẩm thành công

1.11.3. Giao diện chỉnh sửa Product

Để chỉnh sửa một bản ghi Product, người quản lý cần truy cập vào một bản ghi. Giao diện như hình 102.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home · Products · Products · test

Change product

HISTORY

test

Name:

Is active

Description:

Category: Pizza ✎ +

Labels: Hot +

Hold down "Control", or "Command" on a Mac, to select more than one.

Variations: Size Base +

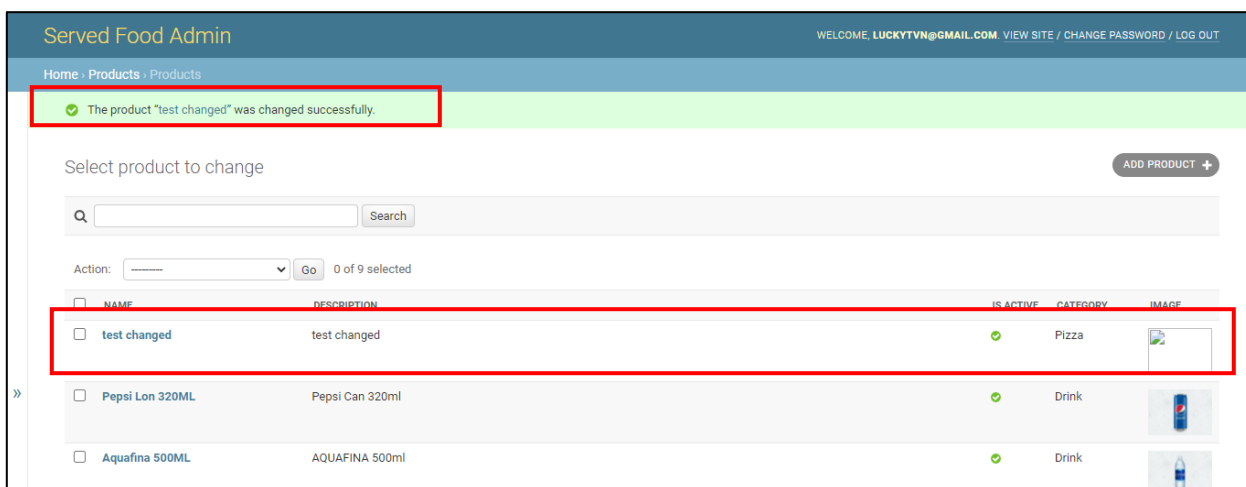
Hold down "Control", or "Command" on a Mac, to select more than one.

Image: Currently: 176080530_1226274554470624_5085427267940184655_n18yfi5j.jpg
Change: Không có tệp nào được chọn

PRODUCT VARIATION OPTIONS		
⊕ VARIATION OPTIONS	PRICE	DELETE?
<input type="text" value="Small"/> <input type="text" value="Medium"/> <input type="text" value="Large"/> <input type="text" value="Pan"/> <input type="text" value="Handtoss"/> <input type="text" value="Thin"/>	<input type="text"/>	<input type="button" value="✕"/>
+ Add another Product variation option		

Hình 102 Giao diện chỉnh sửa sản phẩm

Sau khi người quản lý thay đổi các thông tin và chọn lưu, kết quả trả về từ website như hình 103.



Hình 103 Giao diện chỉnh sửa sản phẩm thành công

1.11.4. Giao diện xóa Product

Để xóa một bản ghi product, người quản lý chọn “Delete” như hình 104.

Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home · Products · Products · test

Change product

test HISTORY

Name:

Is active

Description:

Category: Pizza ✎ +

Labels: Hot +
Hold down "Control", or "Command" on a Mac, to select more than one.

Variations: Size Base +
Hold down "Control", or "Command" on a Mac, to select more than one.

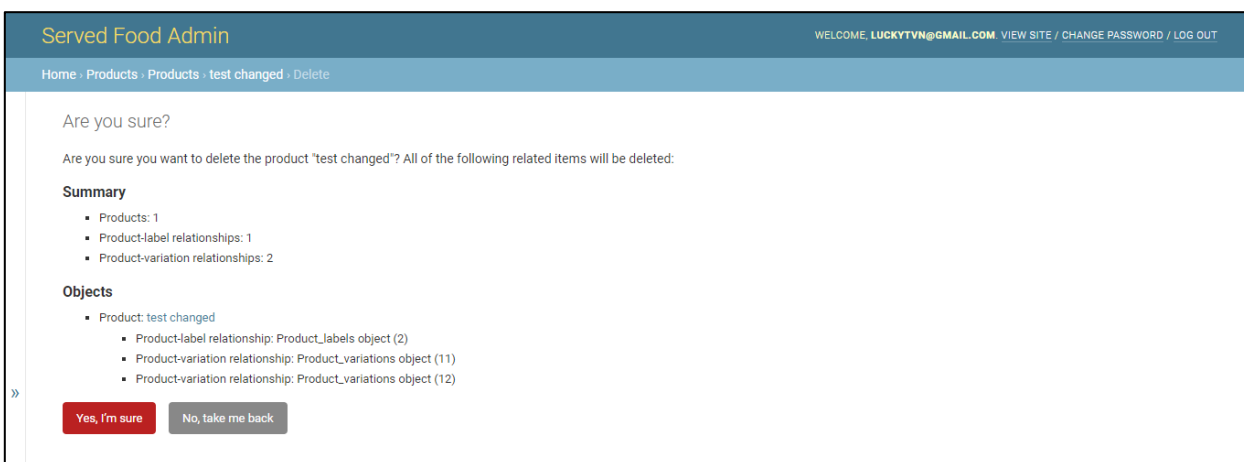
Image: Currently: 176080530_1226274554470624_5085427267940184655_n18yfi5j.jpg
Change: Chọn tệp Không có tệp nào được chọn

PRODUCT VARIATION OPTIONS		
VARIAION OPTIONS	PRICE	DELETE?
<ul style="list-style-type: none"> Small Medium Large Pan Handtoss Thin 	<input type="text"/>	✕
+ Add another Product variation option		

Delete Save and add another Save and continue editing SAVE

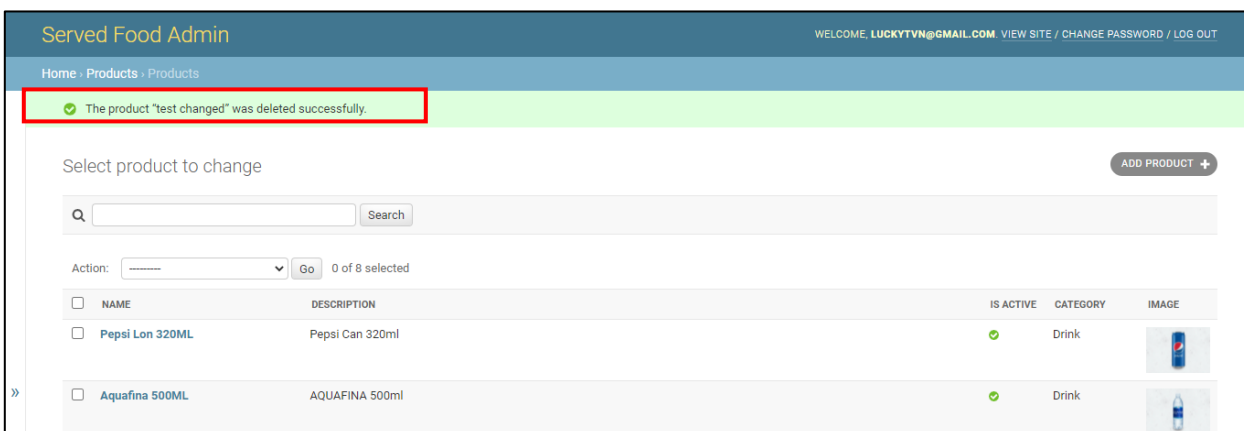
Hình 104 Giao diện xóa sản phẩm

Sau khi chọn “Delete”, hệ thống yêu cầu người quản lý xác minh tác vụ xóa. Giao diện như hình 105.



Hình 105 Giao diện xác nhận xóa sản phẩm

Sau khi quản lý xác nhận xóa, hệ thống xóa bản ghi khỏi cơ sở dữ liệu và gửi kết quả về như hình 106.

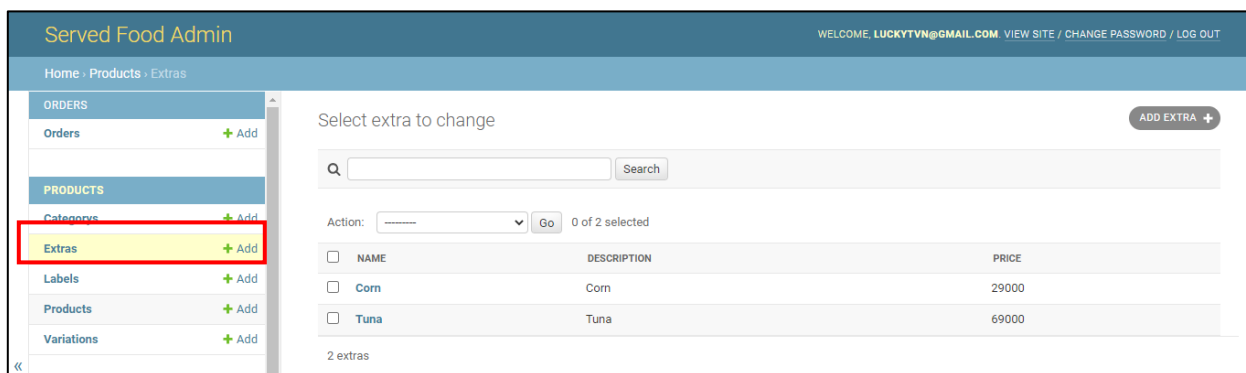


Hình 106 Giao diện xóa sản phẩm thành công

1.12. Giao diện Extras (Sản phẩm gọi thêm)

1.12.1. Giao diện chính Extras

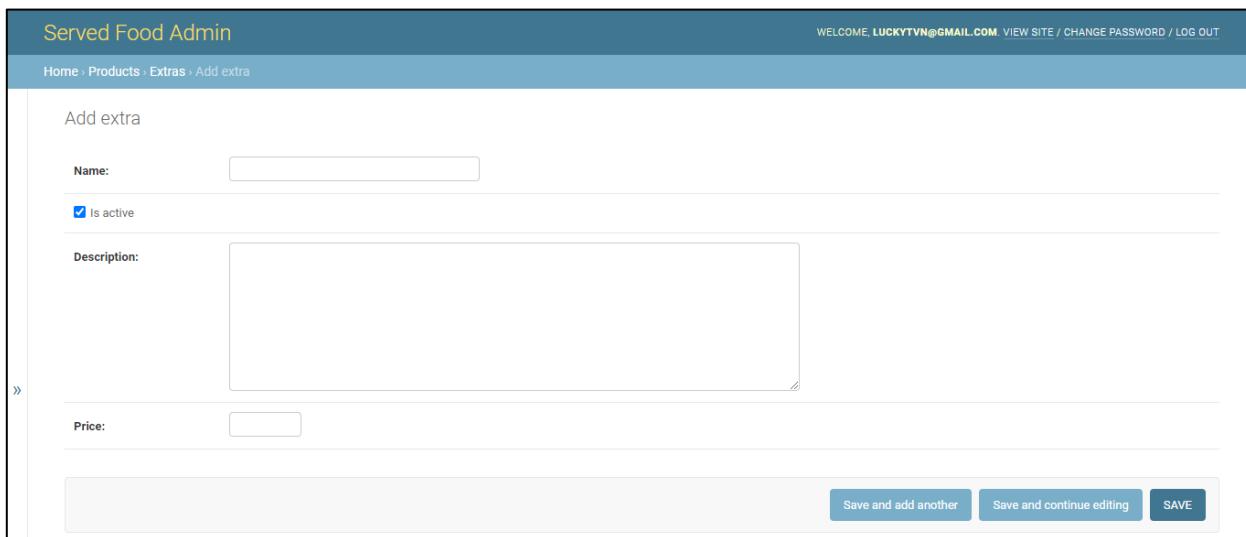
Để truy cập vào giao diện danh mục Extras, người quản lý chọn danh mục Extras. Giao diện hiển thị như hình 107.



Hình 107 Giao diện sản phẩm gọi thêm

1.12.2. Giao diện thêm mới Extra

Người quản lý chọn “ADD EXTRA” để thêm mới một bản ghi extra. Giao diện như hình 108.



Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Extras > Add extra

Add extra

Name:

Is active

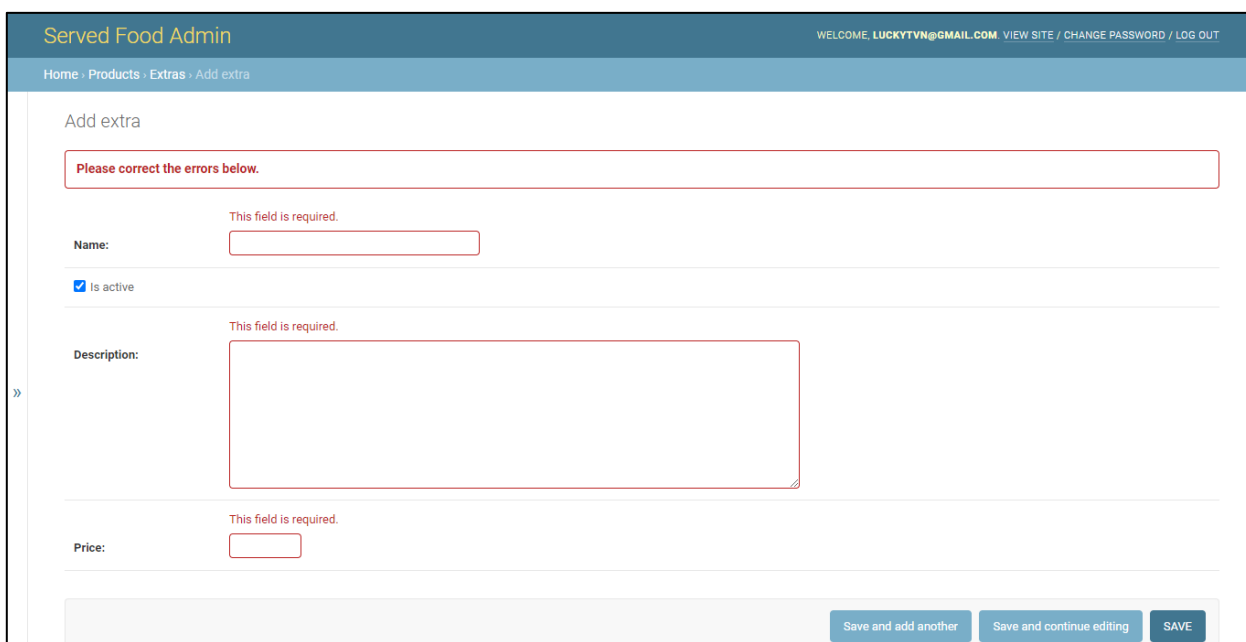
Description:

Price:

Save and add another Save and continue editing SAVE

Hình 108 Giao diện thêm mới sản phẩm gọi thêm

Trường hợp người quản lý nhập không đầy đủ các trường dữ liệu, website báo lỗi như hình 109.



Served Food Admin WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Extras > Add extra

Add extra

Please correct the errors below.

Name: This field is required.

Is active

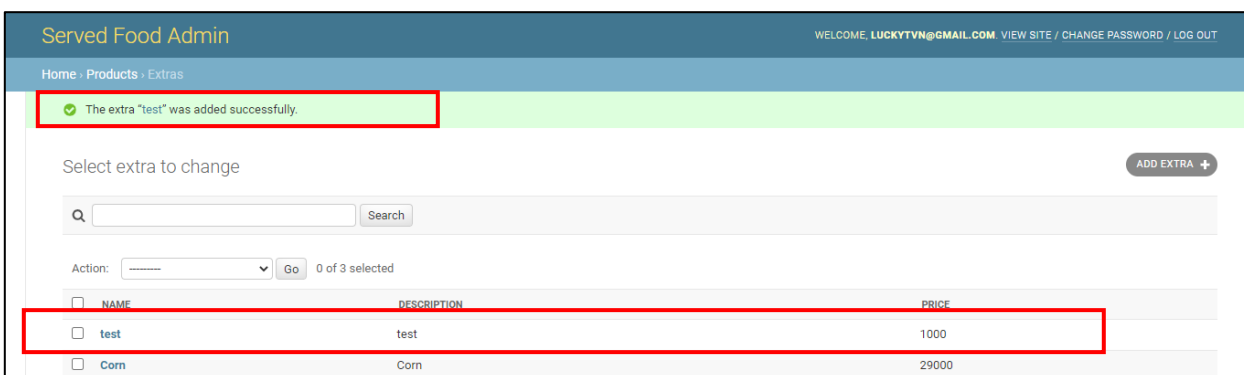
Description: This field is required.

Price: This field is required.

Save and add another Save and continue editing SAVE

Hình 109 Giao diện thêm mới sản phẩm gọi thêm lỗi

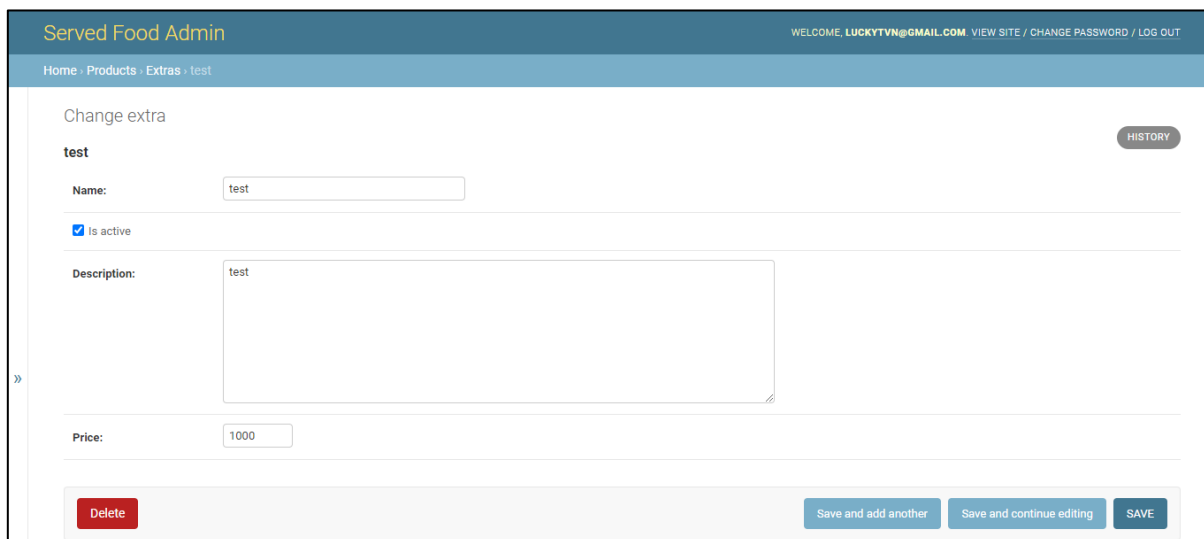
Sau khi nhập đầy đủ thông tin và chọn lưu. Website gửi kết quả thành công đến người quản lý như hình 110.



Hình 110 Giao diện thêm mới sản phẩm gọi thêm thành công

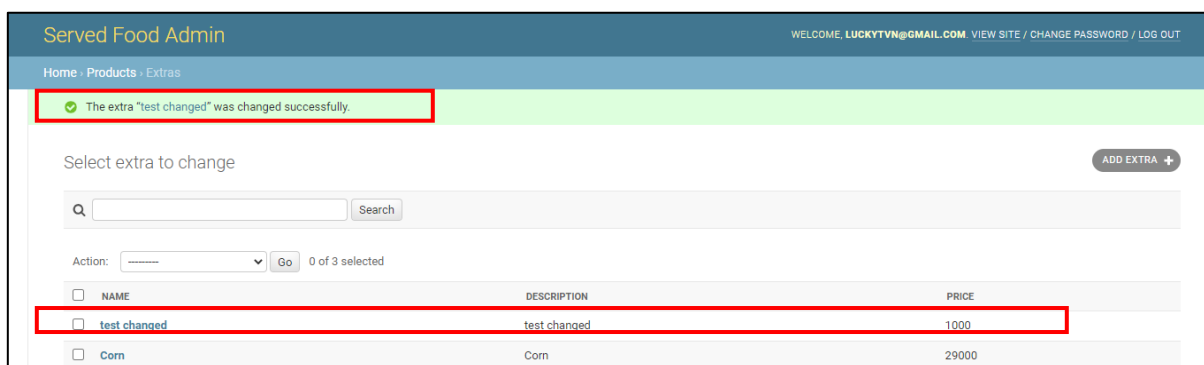
1.12.3. Giao diện chỉnh sửa Extra

Người quản lý truy cập một bản ghi để chỉnh sửa bản ghi. Giao diện như hình 111.



Hình 111 Giao diện chỉnh sửa sản phẩm gọi thêm

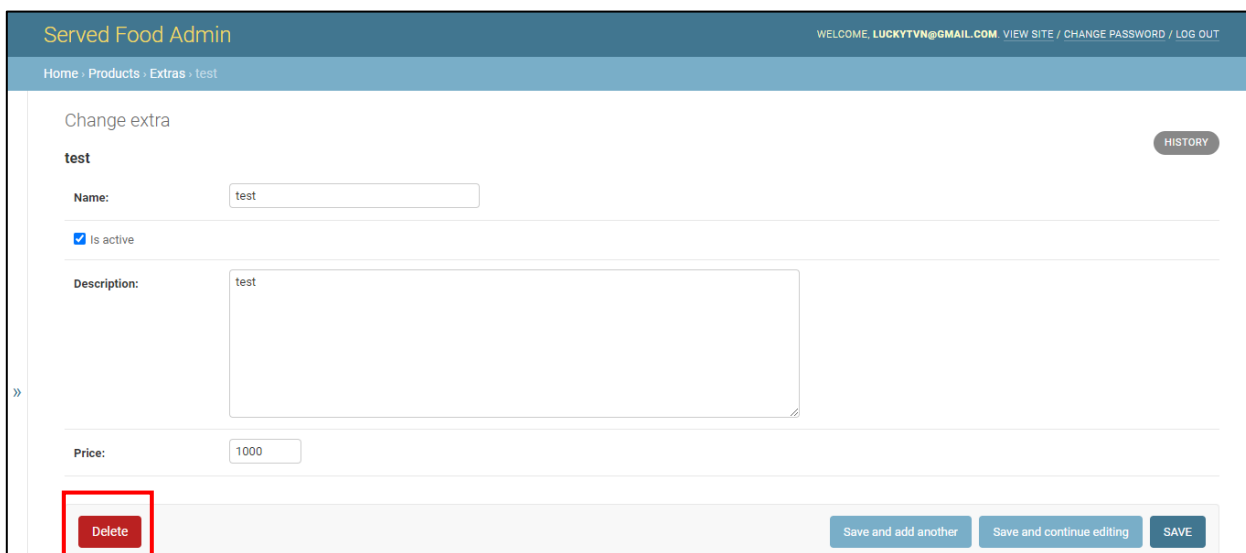
Sau khi thay đổi các trường thông tin, người quản lý chọn lưu. Hệ thống lưu trữ bản ghi với những thông tin mới và gửi về người quản lý thông báo thành công. Giao diện như hình 112.



Hình 112 Giao diện chỉnh sửa sản phẩm gọi thêm thành công

1.12.4. Giao diện xóa Extra

Người quản lý chọn “Delete” để xóa một bản ghi. Giao diện như hình 113.



Served Food Admin

WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home · Products · Extras · test

Change extra

test HISTORY

Name: test

Is active

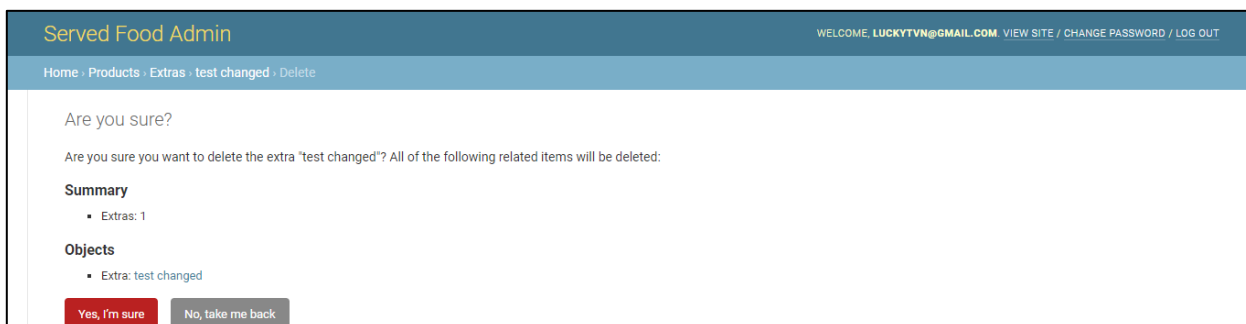
Description: test

Price: 1000

Delete Save and add another Save and continue editing SAVE

Hình 113 Giao diện xóa sản phẩm gọi thêm

Sau khi chọn “Delete”, hệ thống yêu cầu người quản lý xác nhận thao tác xóa, giao diện như hình 114.



Served Food Admin

WELCOME, LUCKYTVN@GMAIL.COM VIEW SITE / CHANGE PASSWORD / LOG OUT

Home · Products · Extras · test changed · Delete

Are you sure?

Are you sure you want to delete the extra "test changed"? All of the following related items will be deleted:

Summary

- Extras: 1

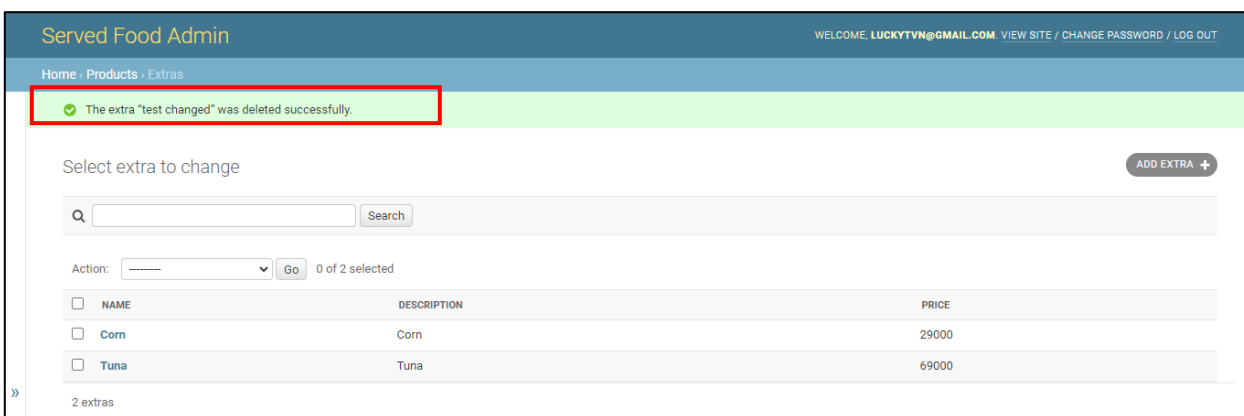
Objects

- Extra: test changed

Yes, I'm sure No, take me back

Hình 114 Giao diện xác nhận xóa sản phẩm gọi thêm

Quản lý chọn “Yes, Im sure” để xác nhận xóa, hệ thống xóa bản ghi khỏi cơ sở dữ liệu và gửi kết quả về như hình 118.

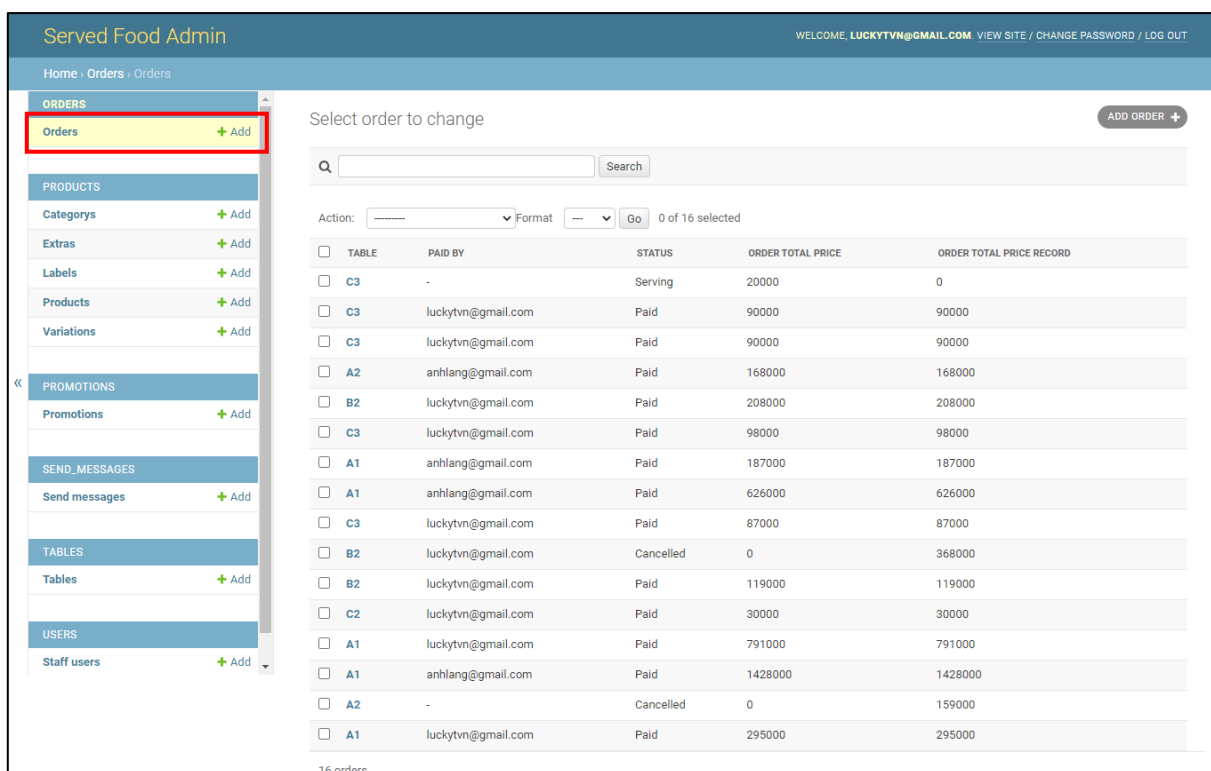


Hình 115 Giao diện xóa sản phẩm gọi thêm thành công

1.13. Giao diện Orders (Đơn hàng)

1.13.1. Giao diện chính Orders

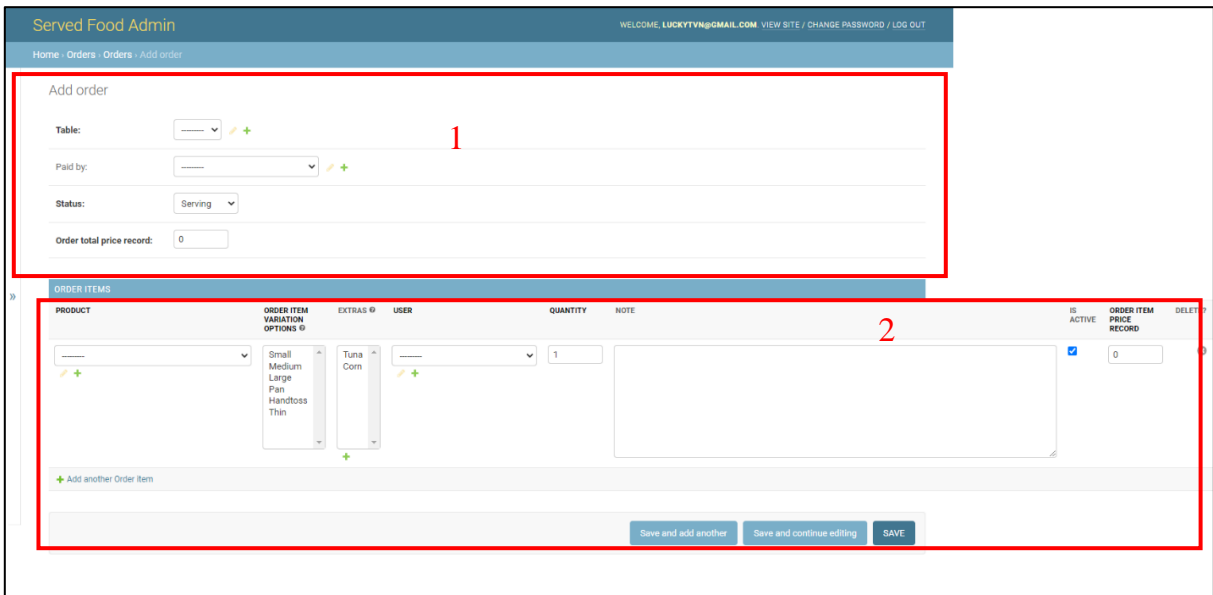
Người quản lý truy cập giao diện danh mục Orders. Giao diện như hình 116.



Hình 116 Giao diện đơn hàng

1.13.2. Giao diện thêm mới Order

Quản lý chọn “ADD ORDER” để tạo mới một order. Giao diện như hình 117.

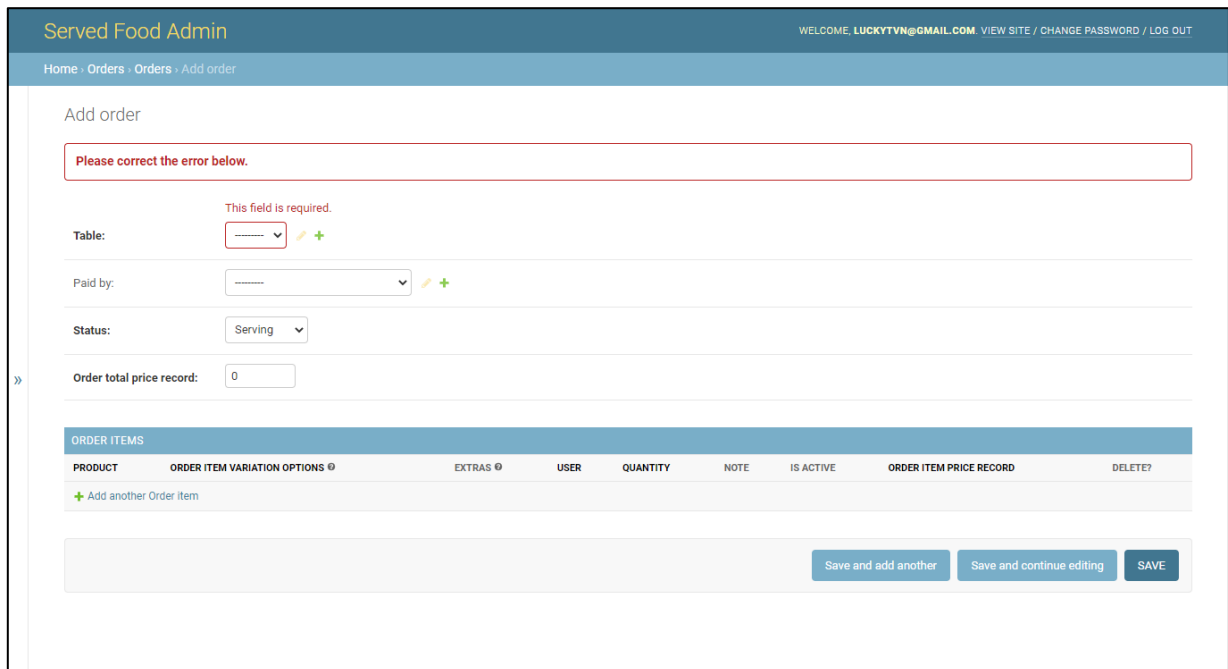


Hình 117 Giao diện thêm mới đơn hàng

Giao diện thêm mới Order được chia thành 2 khối như hình 120:

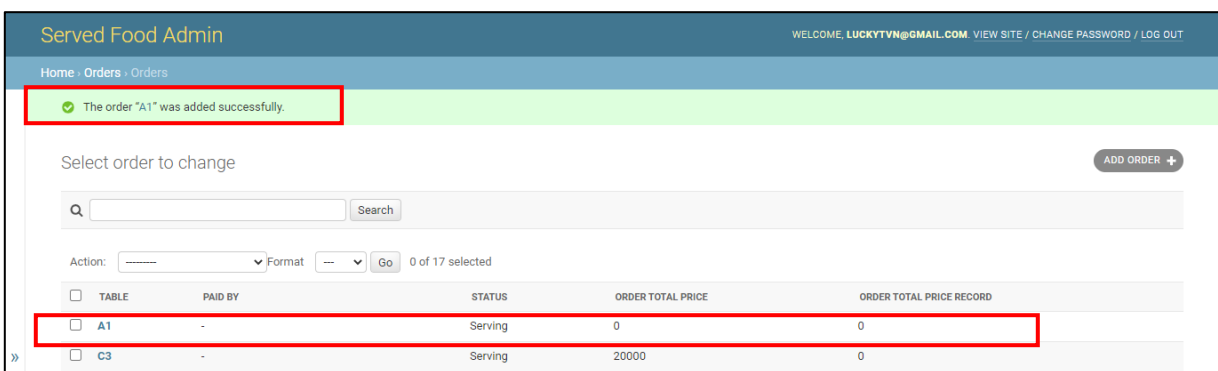
1. Thông tin chung: Thông tin chung của hóa đơn bao gồm bàn ăn, trạng thái, ...
2. Chi tiết hóa đơn: Các thông tin về chi tiết đơn hàng như sản phẩm, sản phẩm gọi thêm, số lượng.

Trường hợp người quản lý nhập sai hoặc thiếu thông tin, website báo lỗi như hình 118.



Hình 118 Giao diện thêm mới đơn hàng lỗi

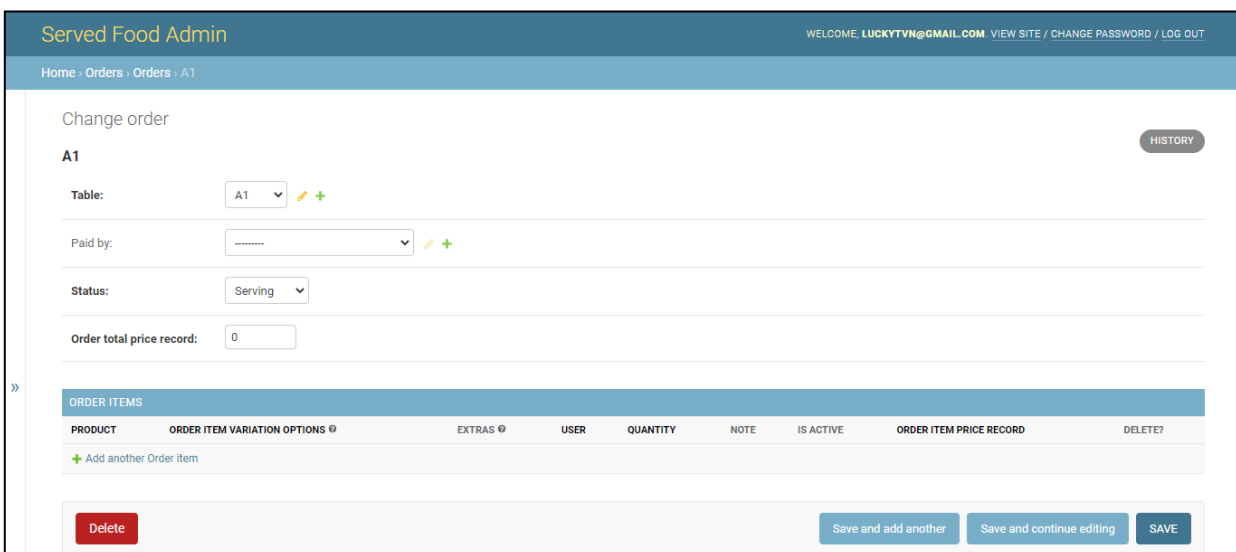
Giao diện thêm bản ghi order thành công như hình 119.



Hình 119 Giao diện thêm mới đơn hàng thành công

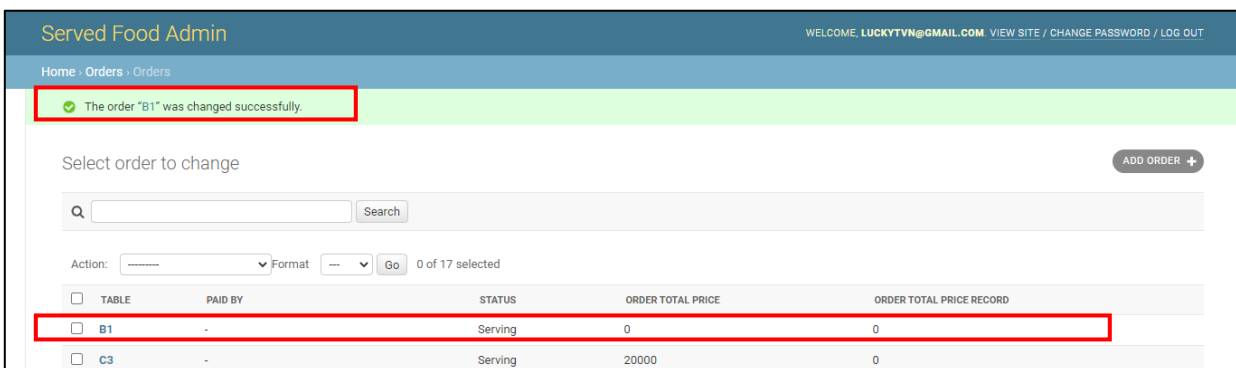
1.13.3. Giao diện chỉnh sửa Order

Người quản lý truy cập vào bản ghi để thực hiện chỉnh sửa. Giao diện như hình 120.



Hình 120 Giao diện chỉnh sửa đơn hàng

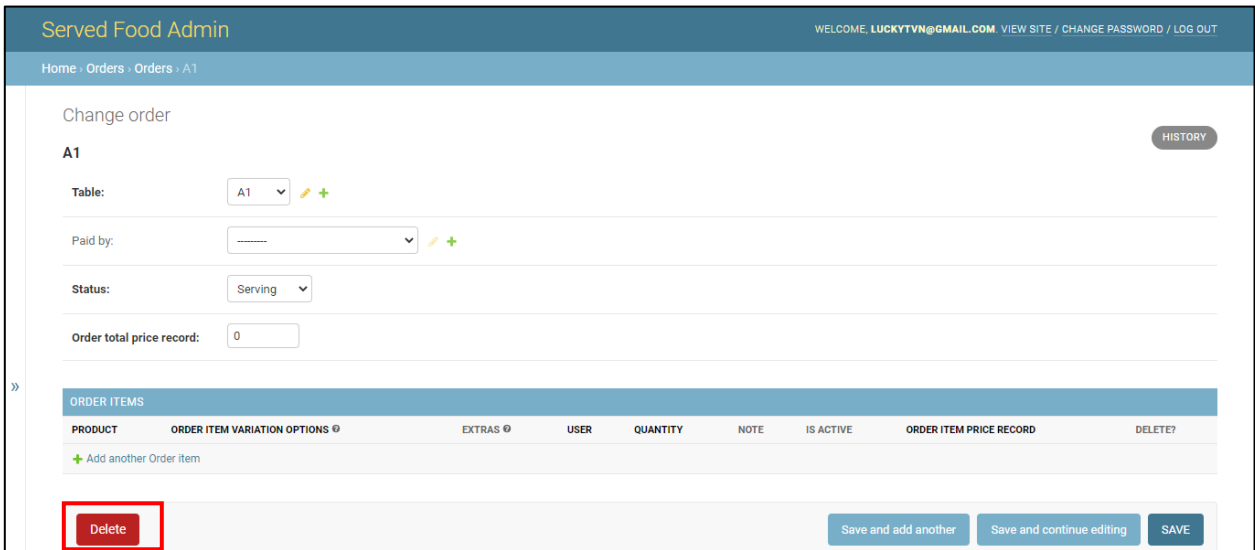
Sau khi thay đổi các trường thông tin và lưu. Hệ thống gửi về kết quả như hình 121.



Hình 121 Giao diện chỉnh sửa đơn hàng thành công

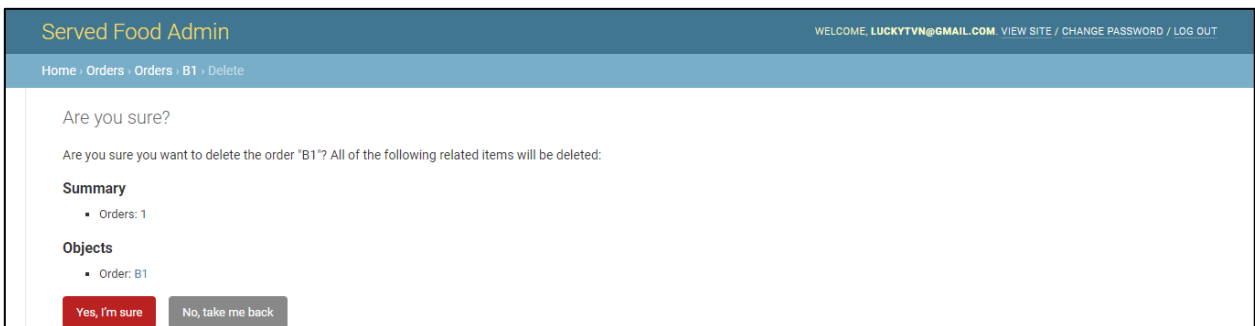
1.13.4. Giao diện xóa order

Người quản lý chọn “Delete” để xóa bản ghi order, giao diện như hình 122.



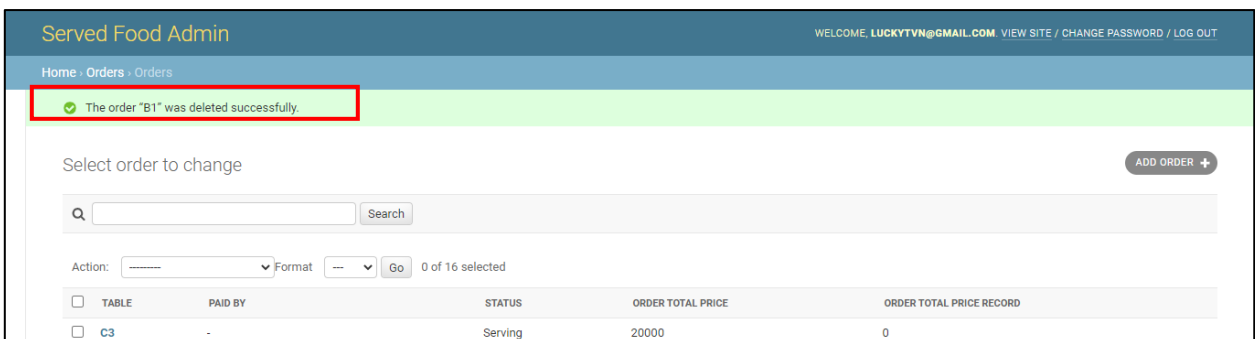
Hình 122 Giao diện xóa đơn hàng

Sau đó hệ thống yêu cầu người quản lý xác minh thao tác xóa, giao diện như hình 123.



Hình 123 Giao diện xác nhận xóa đơn hàng

Người quản lý xác minh thao tác xóa, hệ thống xóa bản ghi khỏi cơ sở dữ liệu và gửi về kết quả như hình 124.



Hình 124 Giao diện xóa đơn hàng thành công

1.14. Các giao diện khác

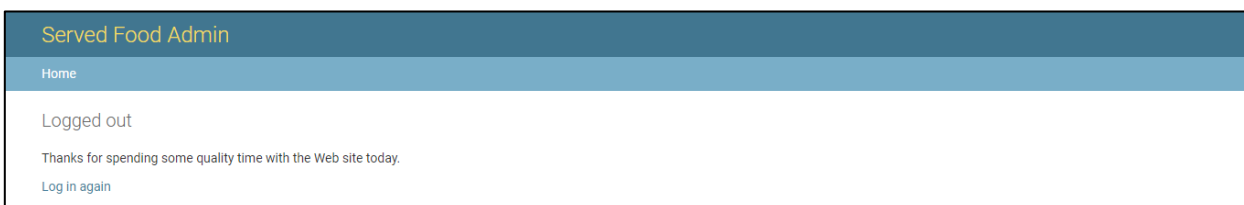
1.14.1. Giao diện đăng xuất

Để đăng xuất, người quản lý chọn “LOG OUT” trên thanh header như hình 125.



Hình 125 Giao diện header website

Sau khi chọn đăng xuất, website hiển thị giao diện như hình 126.



Hình 126 Giao diện đăng xuất

Trong đó “Log in again” sẽ hướng người quản lý đến giao diện đăng nhập.

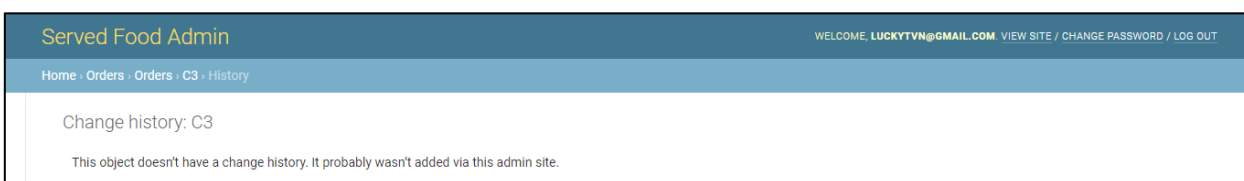
1.14.2. Giao diện lịch sử chỉnh sửa bản ghi

Khi truy cập vào mỗi bản ghi sẽ xuất hiện “HISTORY” – Lịch sử chỉnh sửa như hình 127.

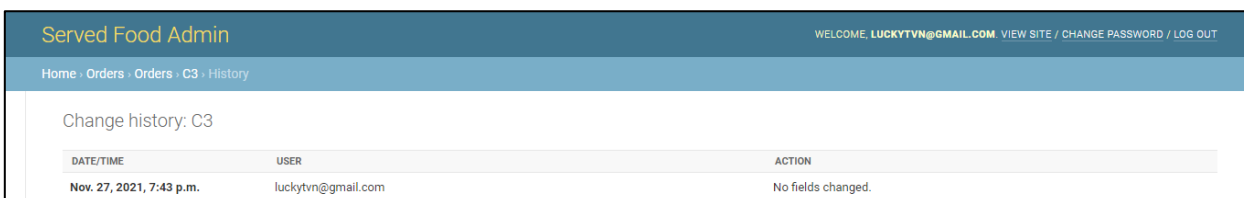


Hình 127 Giao diện nút "HISTORY"

Để xem lịch sử chỉnh sửa chọn “HISTORY”, các trường hợp lịch sử bản ghi có thể xảy ra như hình 128, 129.



Hình 128 Giao diện lịch sử trống



Hình 129 Giao diện lịch sử

2. Xây dựng ứng dụng di động

2.1. Tổng quan

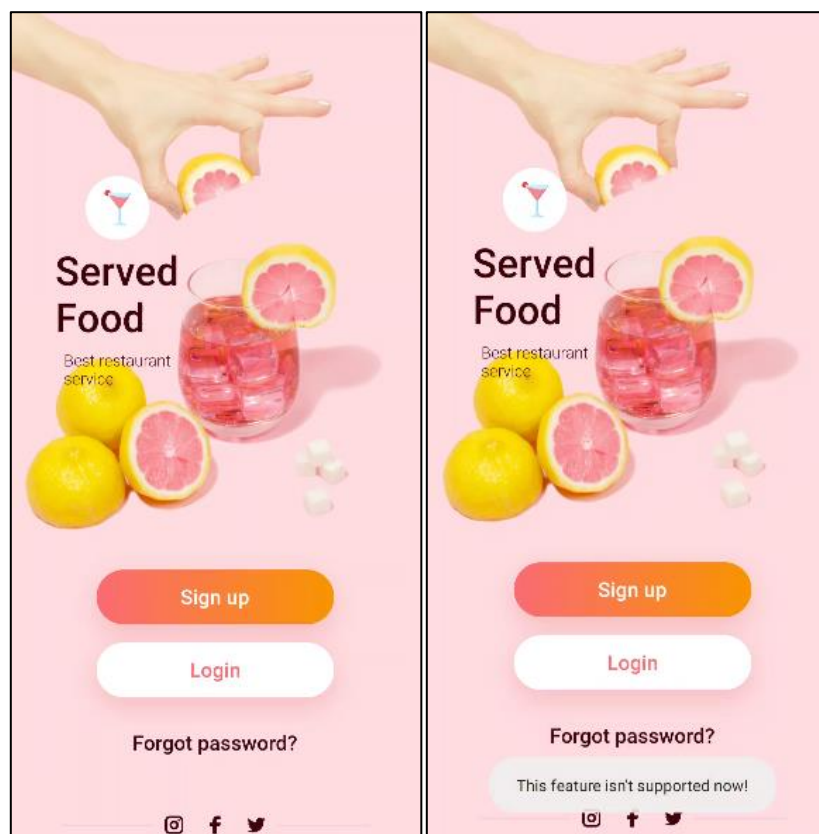
Ứng dụng di động của hệ thống được đặt tên “Served Food”, ứng dụng phát triển đa nền tảng bao gồm Android và IOS nhưng tập trung chủ yếu vào nền tảng Android. Icon đại diện của ứng dụng như hình 130.



Hình 130 Biểu tượng ứng dụng di động

2.2. Giao diện Welcome

Khi người dùng mở ứng dụng trên thiết bị di động. Ứng dụng hiển thị giao diện Welcome chào đón người dùng. Giao diện như hình 131a.



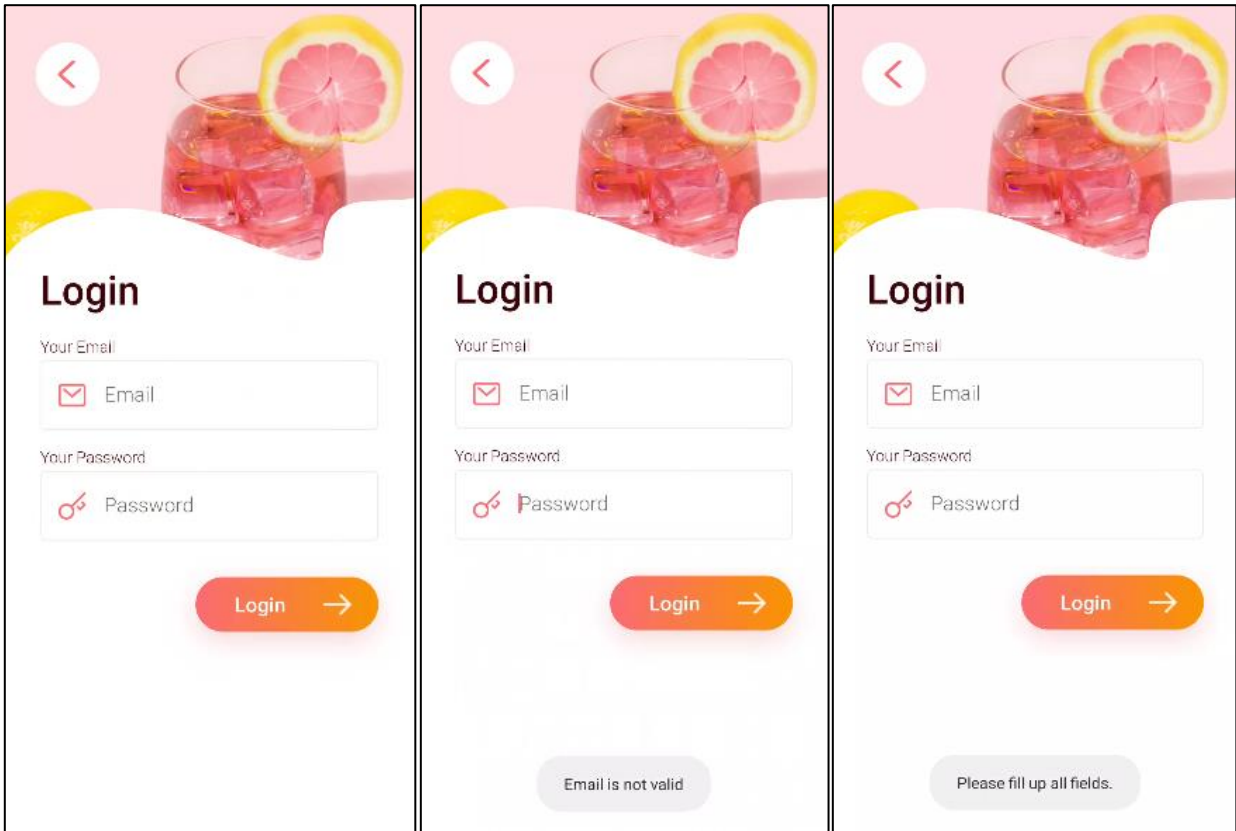
Hình 131 a,b Giao diện đăng nhập

Giao diện Welcome có các tính năng như: Sign Up – Đăng kí; Login – Đăng nhập; Forgot password ? – Quên mật khẩu; và các đường dẫn mạng xã hội. Trong đó “Đăng kí” chưa được hỗ trợ, khi người dùng nhấp vào giao diện hiển thị như hình 131b.

Các tính như “Đăng kí” và “Quên mật khẩu” không được phát triển để đáp ứng tình huống thực tế người quản lý sẽ cung cấp và quản lý thông tin của tài khoản người dùng.

2.3. Giao diện Login (Đăng nhập)

Để đăng nhập vào ứng dụng, người dùng chọn “Login” từ giao diện Welcome, ứng dụng điều hướng người dùng đến giao diện Login như hình 132a.

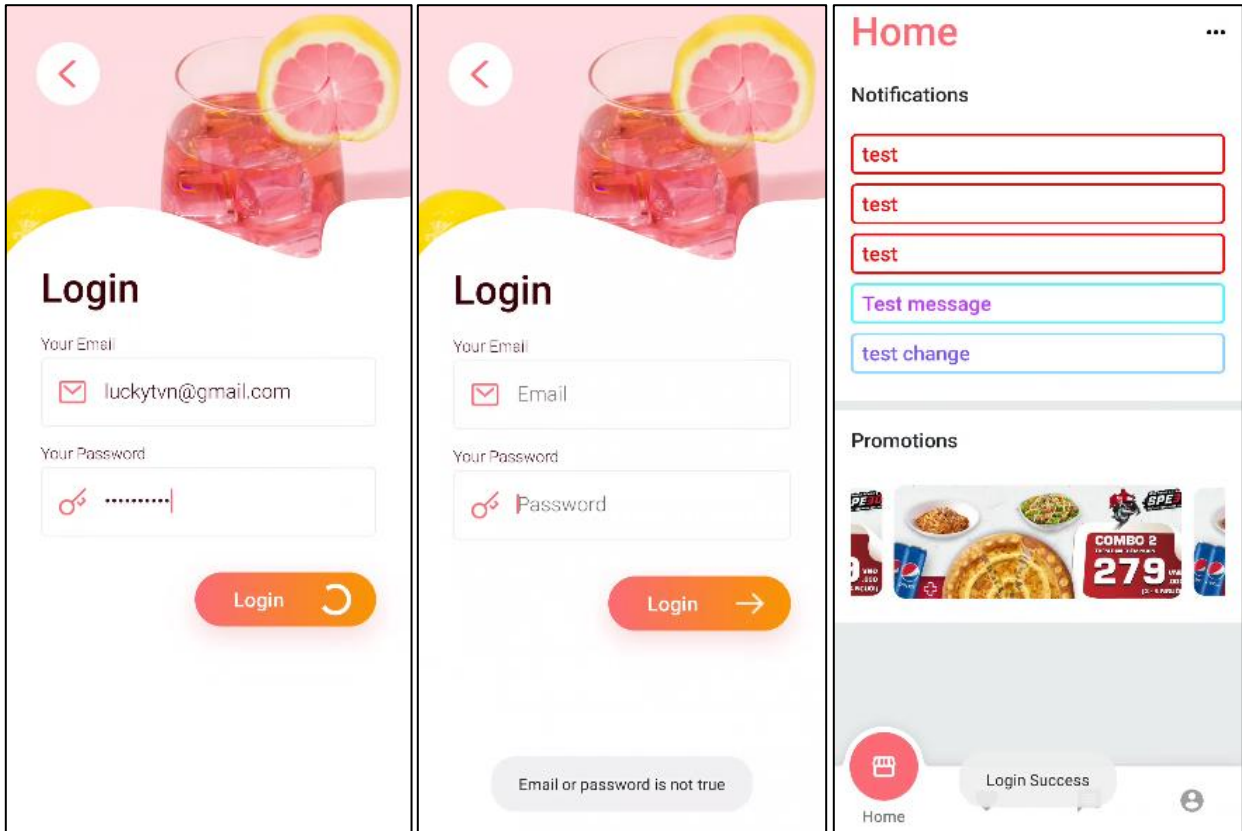


Hình 132 a,b,c Giao diện đăng nhập

Như hình 132a, ứng dụng yêu cầu người dùng nhập các thông tin “Email” và “Password” và chọn “Login” để đăng nhập. Trường hợp người dùng nhập thiếu thông tin, ứng dụng thông báo như hình 132b.

Nếu người dùng nhập đủ thông tin, ứng dụng sẽ kiểm tra định dạng Email nhập vào. Nếu không đúng định dạng, ứng dụng hiển thị giao diện như hình 132c.

Nếu người dùng nhập đầy đủ thông tin và “Email” đúng định dạng. Người dùng chọn “Login”, ứng dụng gửi thông tin “Email” và “Password” lên hệ thống để xác minh. Giao diện xử lý như hình 133a.



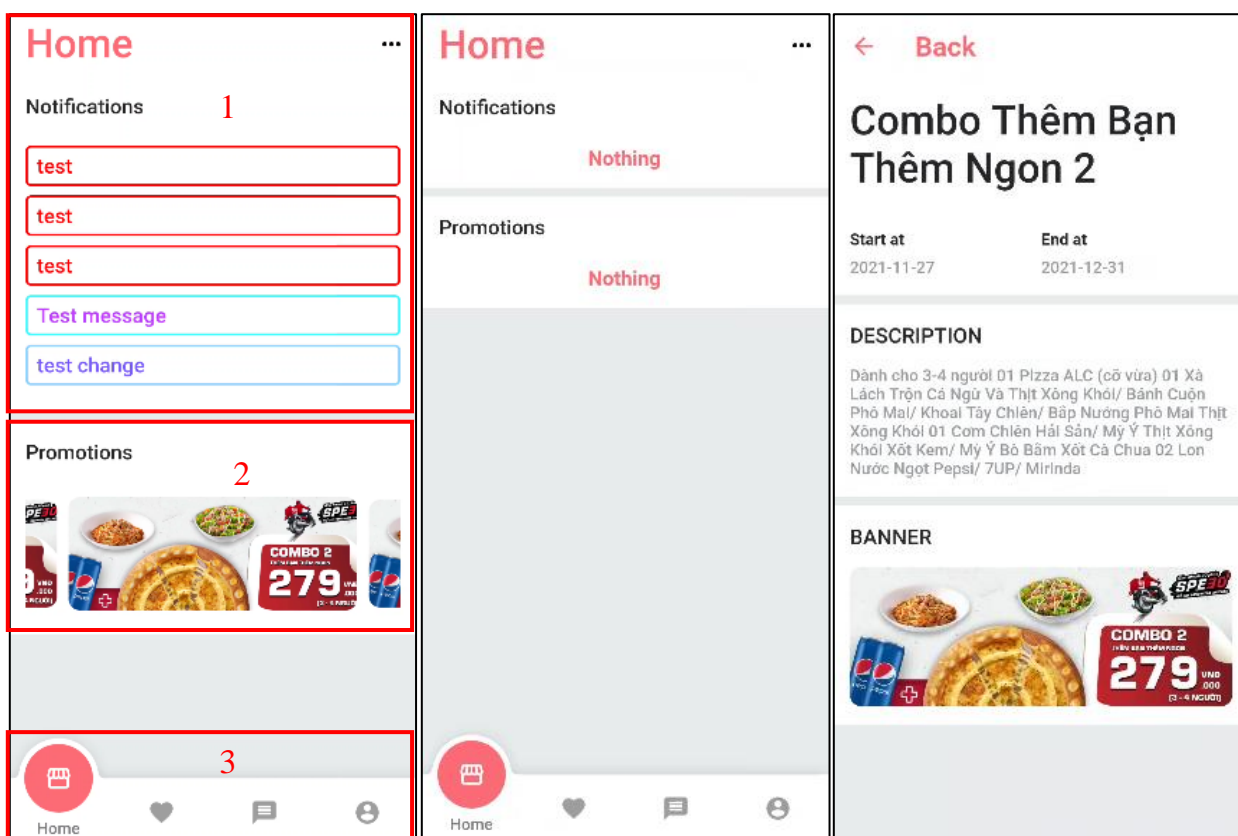
Hình 133 a,b,c Giao diện xử lý đăng nhập

Trường hợp sai toàn khoản hoặc mật khẩu, ứng dụng hiển thị kết quả như hình 133b.

Trường hợp tài khoản và mật khẩu chính xác, ứng dụng chuyển người dùng đến giao diện trang chủ và hiển thị kết quả đăng nhập thành công như hình 133c.

2.4. Giao diện Home (Trang chủ)

Sau khi đăng nhập thành công, ứng dụng hiển thị giao diện Home như hình 134a.



Hình 134 a,b,c Giao diện trang chủ

Giao diện Home được chia thành 3 khối như hình 142a:

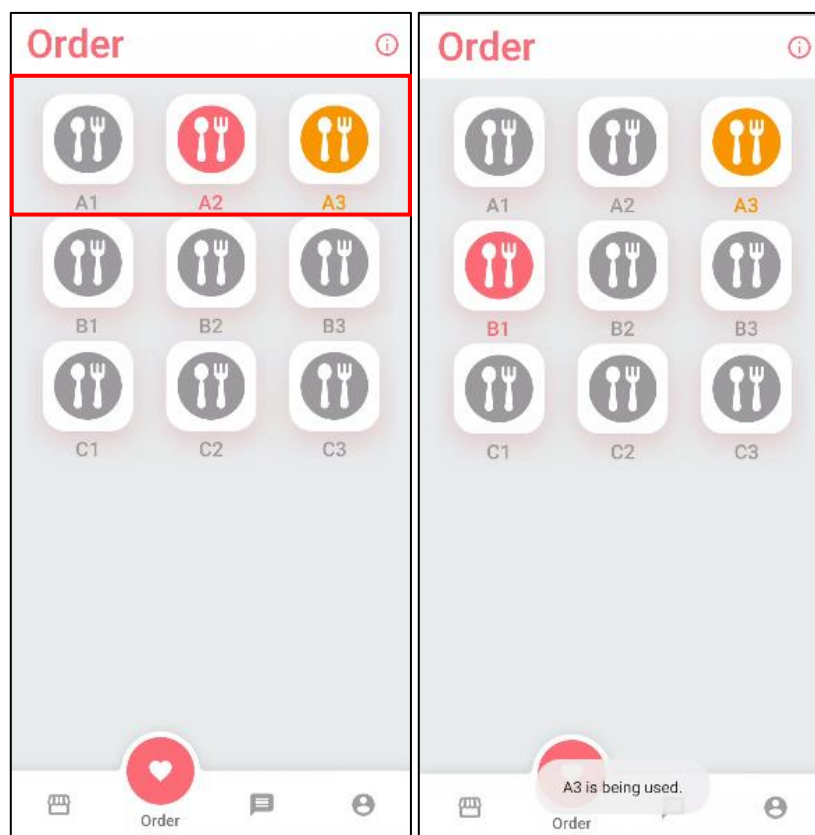
1. Notifications: Khối này hiển thị những lời nhắn từ danh mục Send Messages từ website quản trị. Những lời nhắn được hiển thị với màu chữ và màu viền theo 2 trường thông tin “Text Color” và “Border Color” của bản ghi Send Message. Trong trường hợp không có lời nhắn vào cả, ứng dụng hiển thị dòng chữ “Nothing” như hình 134b.
2. Promotions: Các chương trình khuyến mãi, dữ liệu được kéo về từ danh mục Promotions của website quản trị. Các chương trình khuyến mãi được thể hiện theo danh sách kiểu Slider tự chuyển theo vòng lặp, mỗi chương trình khuyến mãi hiển thị theo hình ảnh từ dữ liệu website như hình 134b.
3. Thanh meo: Thanh menu để chuyển các màn hình bao gồm: Home – Trang chủ với Icon gian hàng; Order – Bàn ăn với Icon trái tim; Chat – Nhắn tin với Icon tin nhắn; Profile – Hồ sơ với Icon hình người.

Tại khối (2) Promotions, người dùng có thể xem chi tiết chương trình khuyến mãi bằng cách nhấp chọn hình ảnh của chương trình đó, giao diện chi tiết hiển thị các thông tin bao gồm: tên chương trình, Start At- ngày có hiệu lực, End At – ngày hết hiệu lực, Description – Chi tiết/Mô tả, Banner- Hình ảnh chương trình, như hình 134c.

Để trở về giao diện trang chủ, người dùng có thể chọn “Back” như hình 134c.

2.5. Giao diện Order (Bàn ăn)

Người dùng chọn Icon trái tim tại thanh menu, ứng dụng chuyển đến giao diện Order như hình 135a.



Hình 135 a,b Giao diện bàn ăn

Mỗi bàn ăn được thể hiện với hình ảnh muỗng – nĩa và tên bàn phía dưới. Màu sắc hiển thị của bàn ăn tượng trưng cho trạng thái bàn ăn, ví dụ:

- Bàn A1: Màu xám thể hiện trạng thái có sẵn.
- Bàn A2: Màu hồng thể hiện trạng thái đã đặt món.
- Bàn A3: Màu vàng thể hiện trạng thái đang được sử dụng. Trong trạng thái này người dùng không thể truy cập bàn ăn, ứng dụng thông báo như hình 135b khi người dùng cố gắng truy cập bàn ăn.

2.6. Giao diện đơn hàng

Để truy cập vào giao diện đơn hàng, người dùng chọn một bàn ăn ở trạng thái có sẵn. Giao diện đơn hàng trông như hình 136.



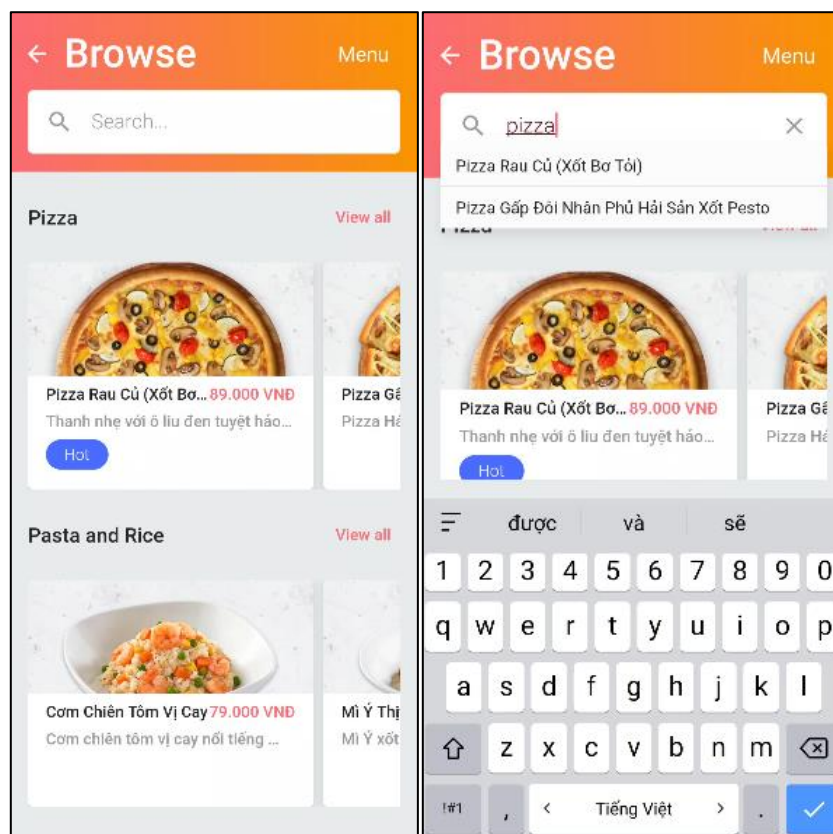
Hình 136 Giao diện đơn hàng trống

Giao diện được thành 4 khối như hình 136.

1. Header: Gồm tên bàn ăn, nút quay lại (kí tự ◀) và nút “Order” – Truy cập vào menu (Giao diện danh sách sản phẩm)
2. Thông tin đơn hàng: Gồm Subtotal – Giá tiền chưa giảm, Discount – % giảm giá và Total – Giá tiền sau khi giảm. Đơn vị tiền tệ hiện tại mặc định là VNĐ.
3. Chi tiết đơn hàng: Khi đơn hàng trống hiển thị “Nothing”, khi đơn hàng có món ăn giao diện được mô tả ở mục sau.
4. Footer: Gồm nút “Checkout” – Thanh toán và nút giỏ hàng (Icon giỏ hàng) về xem các món ăn đã thêm vào giỏ hàng.

2.7. Giao diện menu

Để truy cập giao diện menu, người dùng chọn “Order” tại khối (1) Header như hình 136. Menu thể hiện các món ăn được chia theo loại món ăn và danh sách món ăn hiển thị theo kiểu Slider như hình 137a.



Hình 137 a,b Giao diện menu

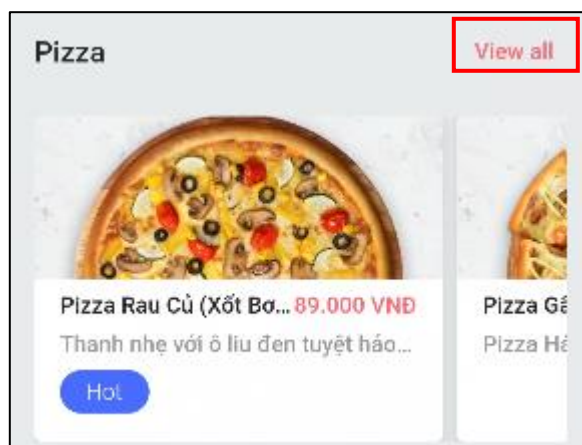
Giao diện chính của menu bao gồm phần header chứa nút quay về (←), nút “menu” – Chuyển hướng giao diện toàn bộ món ăn và thanh tìm kiếm. Chức năng tìm kiếm ăn món theo tên hiển thị như hình 137b.

Người dùng chọn một món ăn tại kết quả của thanh tìm kiếm, ứng dụng sẽ điều hướng người dùng đến giao diện chi tiết món ăn đó.



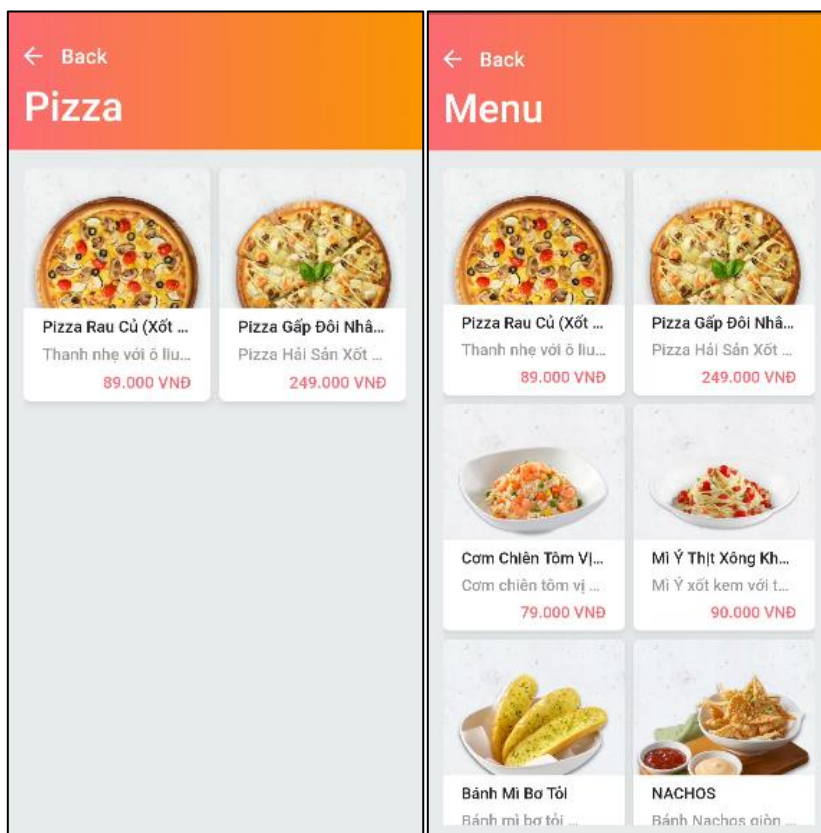
Hình 138 Giao diện món ăn

Mỗi món ăn được thể hiện thành một khối với hình ảnh món ăn, tên món ăn, giá món ăn (lấy giá thấp nhất), mô tả và nhãn của món ăn (Labels) như hình 150. Khi nhấp vào khối ứng dụng điều hướng người dùng đến giao diện chi tiết món ăn.



Hình 139 Giao diện "View all"

Mỗi mục loại món ăn được hiển thị tên loại món ăn, nút “View all” – Chuyển hướng đến giao diện tất cả món ăn thuộc loại và danh sách món ăn theo kiểu Slider như hình . Khi nhấn chọn “View all”, ứng dụng điều hướng giao diện như hình 140a.



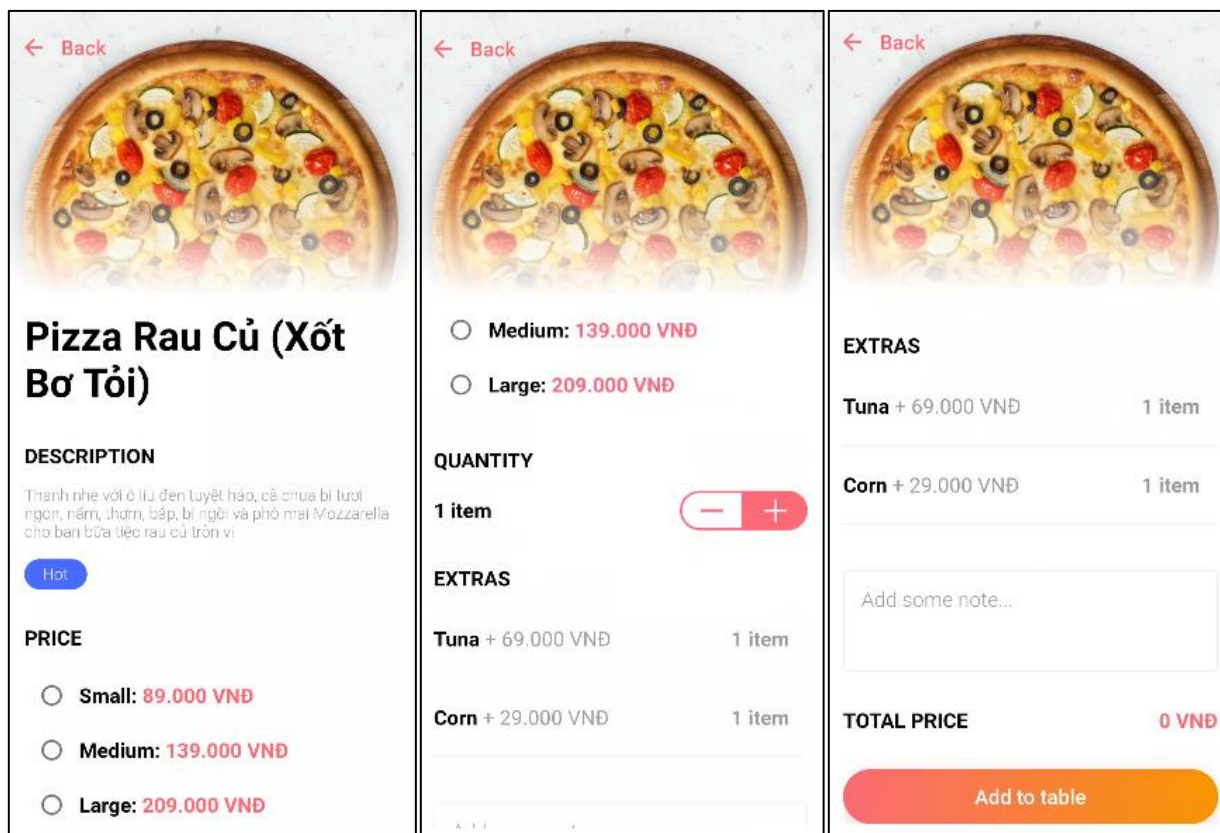
Hình 140 a,b Giao diện menu theo loại

Hình 140b hiển thị giao diện với header bao gồm tên loại món ăn và nút “Back”, các món ăn hiển thị theo danh sách Grid với 2 món ăn mỗi dòng.

Để hiển thị giao diện tất cả sản phẩm, người dùng chọn “Menu” như ở hình 137a, giao diện như hình 140b.

2.8. Giao diện chi tiết món ăn

Khi người dùng nhấp chọn một món ăn, giao diện chi tiết món ăn hiển thị khi cuộn xuống theo thứ tự hình 141a, 141b và 141c.



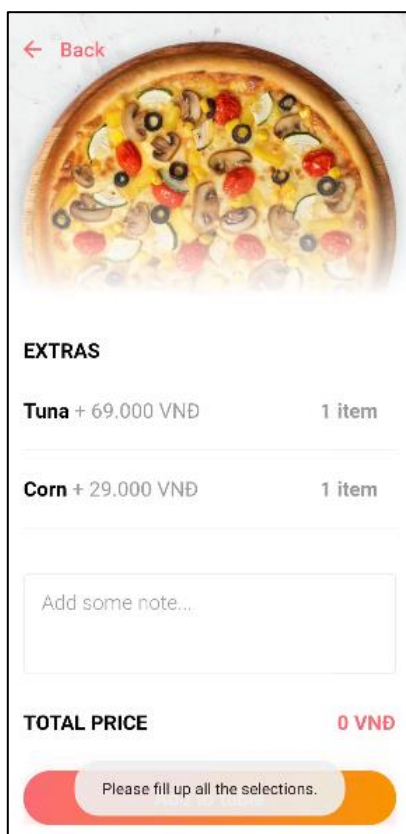
Hình 141 a,b,c Giao diện chi tiết món ăn

Hình 141a thể hiện hình ảnh món ăn kèm theo nút “Back”, tên món ăn, mô tả, nhãn món ăn và giá món ăn theo kích cỡ.

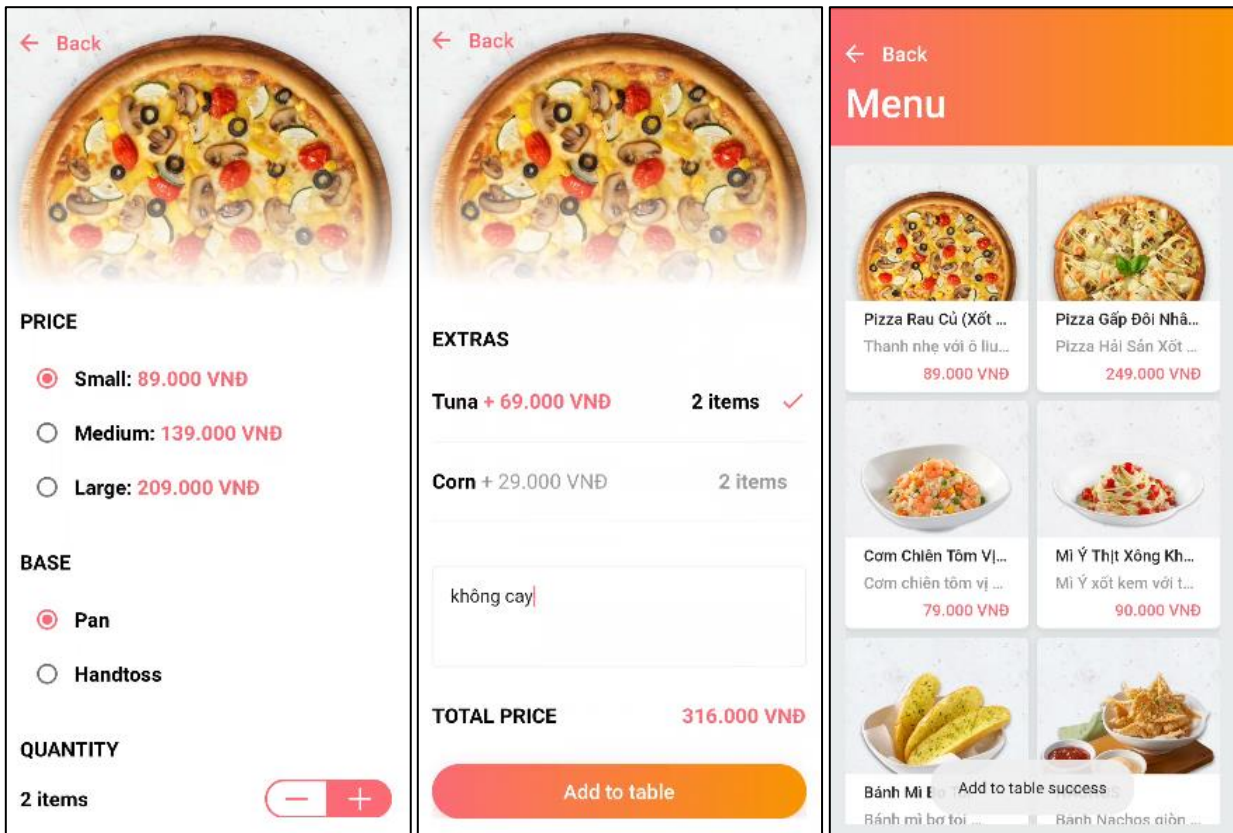
Hình 141b thể hiện số lượng, có thể tăng giảm số lượng chỉ chọn nút “+” hoặc “-“, số lượng mặc định là 1 và không thể về 0. Extras (Gọi thêm) được thể hiện gồm tên, giá và số lượng, số lượng đồng bộ với số lượng như ở trên.

Hình 141c thể hiện note – ghi chú: gửi một ghi chú đến bộ phận bếp làm món ăn, total price – tổng giá và nút “Add to table” để thêm món ăn vào giỏ hàng.

Khi người dùng để trống những trường thông tin yêu cầu và chọn “Add to table”, ứng dụng thông báo đến người dùng như hình 142.



Hình 142 Giao diện thêm vào giỏ hàng thất bại
Khi người dùng điền và chọn đầy đủ, giao diện có thể như hình 143a và 143b.

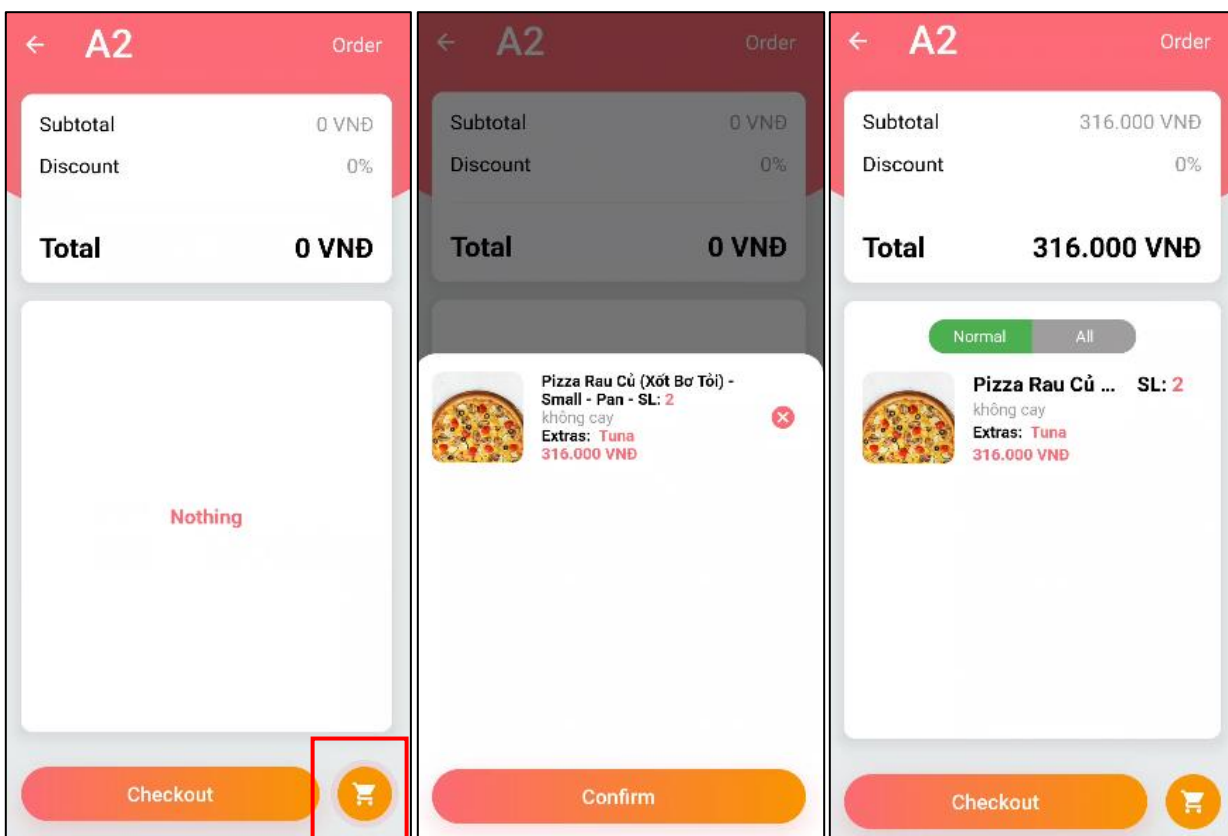


Hình 143 a,b,c Giao diện thêm vào giỏ hàng

Khi thêm món ăn vào giỏ hàng thành công, ứng dụng chuyển người dùng về giao diện trước đó và hiện thông báo như hình 143c.

2.9. Giao diện giỏ hàng

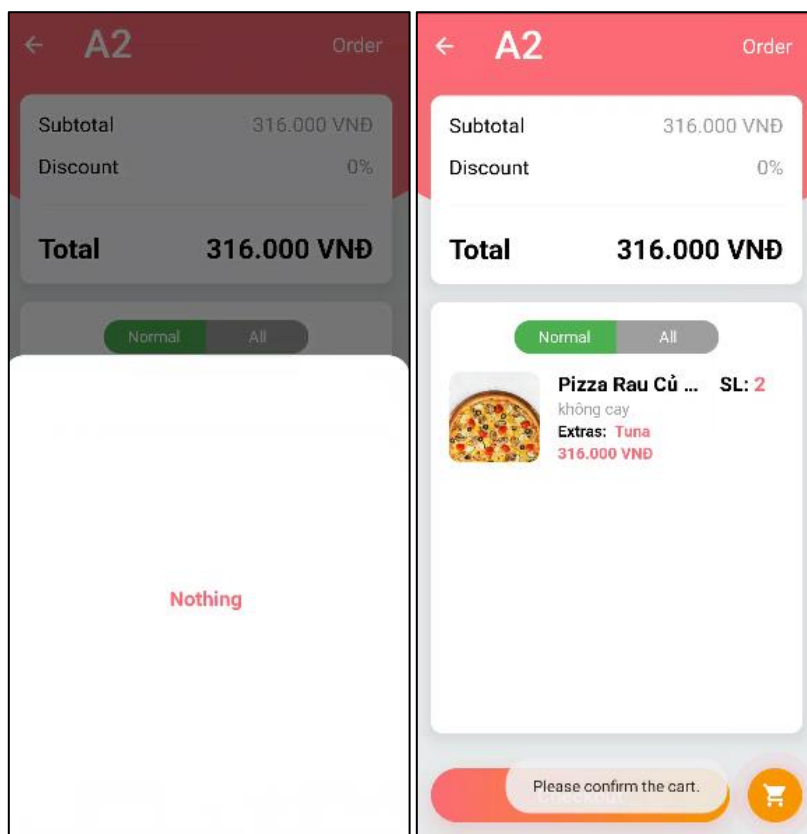
Để truy cập vào giỏ hàng, người dùng chọn nút giỏ hàng như hình 144a.



Hình 144 a,b,c Giao diện giỏ hàng

Giỏ hàng bao gồm danh sách món ăn đã đặt, mỗi món ăn hiển thị thông tin bao gồm hình ảnh, tiêu đề (Kết hợp giữa tên món ăn, biến thể và số lượng - SL), ghi chú, extras (gọi thêm) và tổng giá. Bên cạnh mỗi món ăn đều có nút xóa khỏi giỏ hàng biểu tượng dấu X và nút “Confirm” để xác nhận đơn hàng như hình 144b.

Sau khi xác nhận đơn hàng, ứng dụng hiển thị như hình 144c. Sau đó giỏ hàng trở về trống như hình 145a.

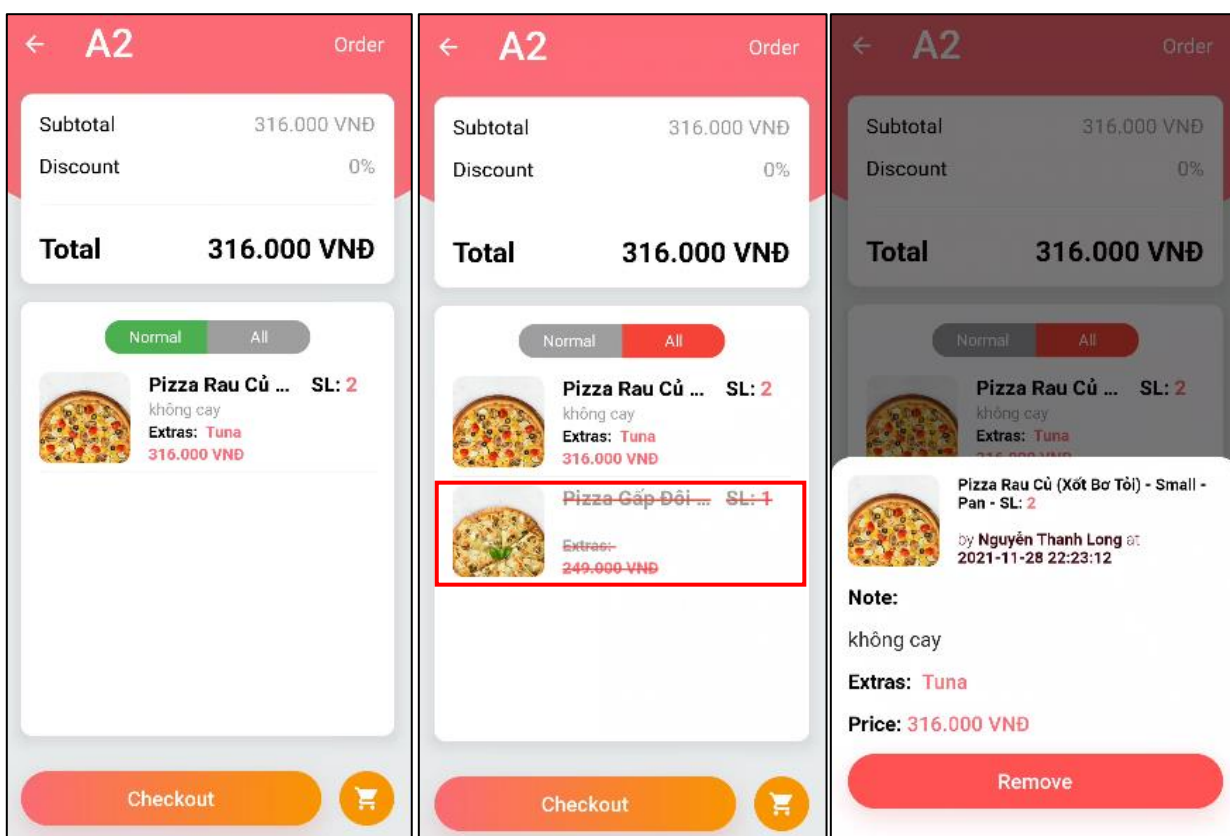


Hình 145 a,b Giao diện giỏ hàng

Trong trường hợp người dùng muốn thoát khỏi giao diện đơn hàng trở về giao diện Order (Bàn ăn). Ứng dụng thông báo như hình 145b.

2.10. Giao diện chi tiết đơn hàng

Giao diện đơn hàng khi có món ăn được thể hiện như hình 146a.



Hình 146 a,b,c Giao diện chi tiết đơn hàng

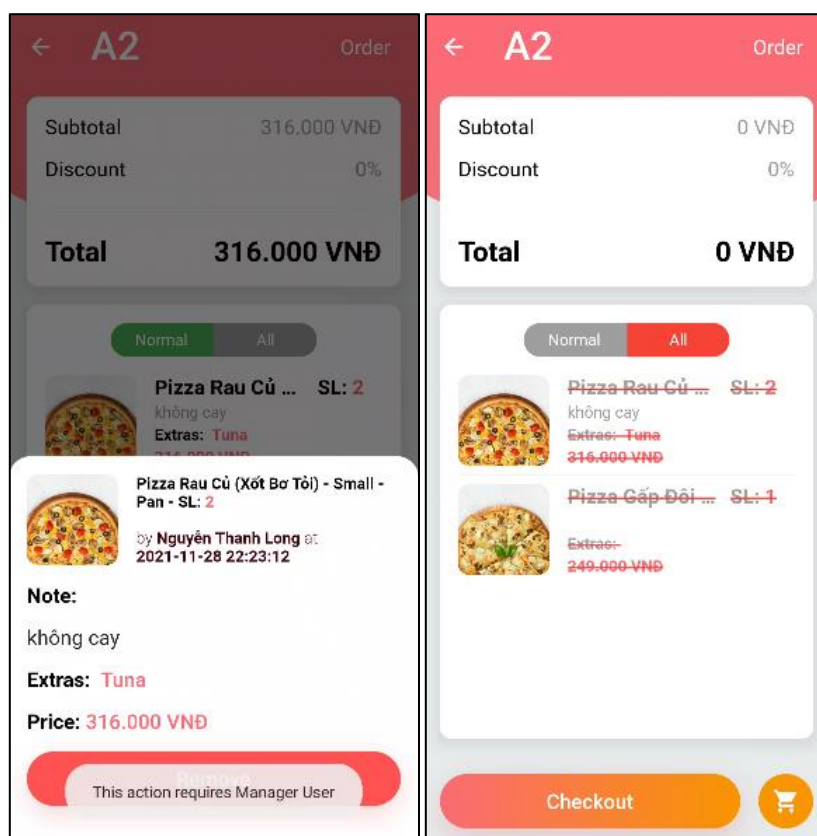
Giao diện hình 146a thể hiện danh sách món ăn đã đặt gồm các thông tin như tiêu đề, số lượng, ghi chú, gọi thêm và giá tiền. Có hai trạng thái hiển thị là “Normal” và “All”, với “Normal” người dùng chỉ thấy những món ăn đang được phục vụ, với “All” người dùng có thể thấy được tất cả món ăn bao gồm các món đã hủy.

Hình 146b thể hiện món ăn bị hủy với những đường gạch đỏ.

Để xem chi tiết thông tin của một món ăn được đặt, người dùng nhấp chọn món ăn đó, giao diện hiển thị như hình 146c bao gồm chi tiết một món ăn được đặt với các thông tin chi tiết như người dùng đặt món và thời gian đặt. Ở đây người dùng có thể chọn “Remove” để hủy món.

2.11. Giao diện hủy món

Để hủy món, người dùng cần có quyền quản lý, nếu không ứng dụng thông báo như hình 147a.

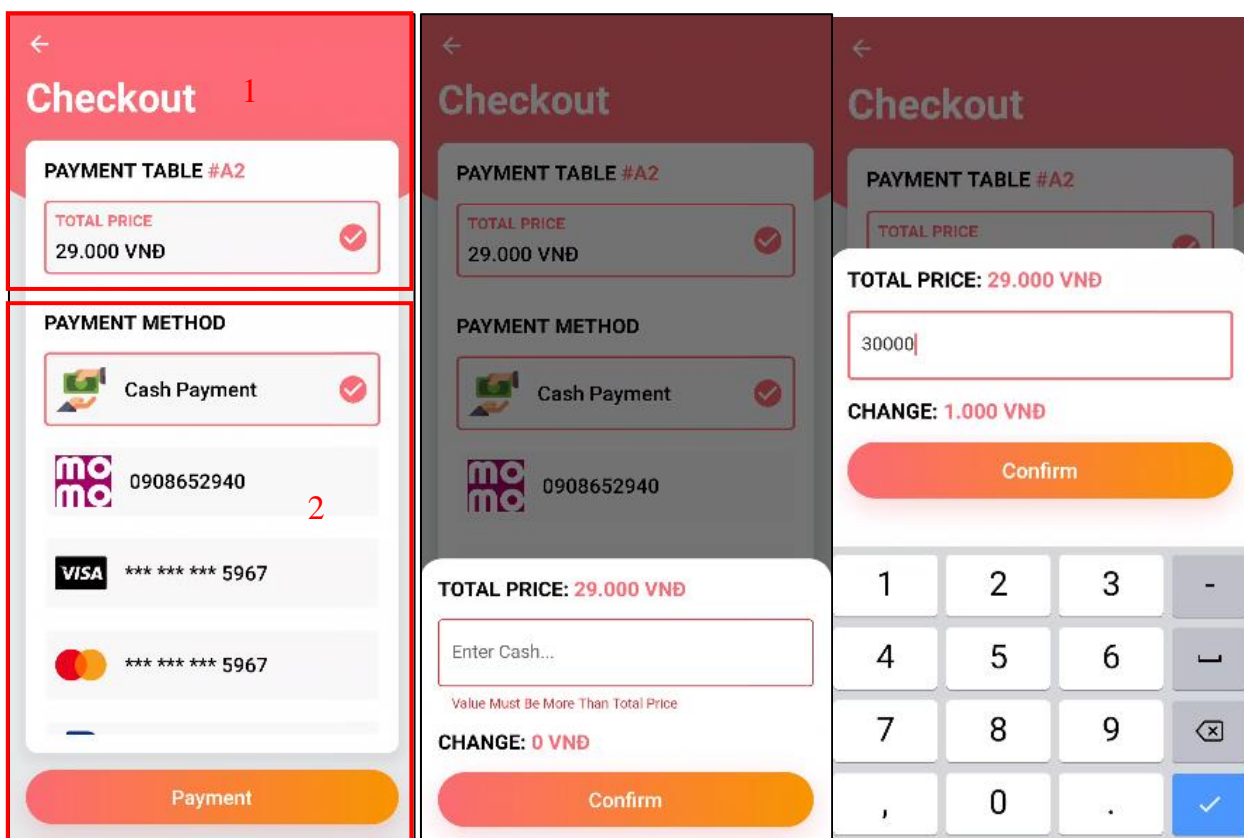


Hình 147 a,b Giao diện hủy món ăn

Nếu có quyền quản lý, người dùng có thể hủy món và kiểm tra lại đơn hàng như hình 147b

2.12. Giao diện thanh toán

Để thực hiện thanh toán, người dùng chọn “Checkout”, giao diện hiển thị như hình 148a.



Hình 148 a,b,c Giao diện thanh toán

Giao diện thanh toán được chia thành 2 khối và nút “Payment”:

1. Tên bàn ăn và tổng tiền.
2. Các phương thức thanh toán.

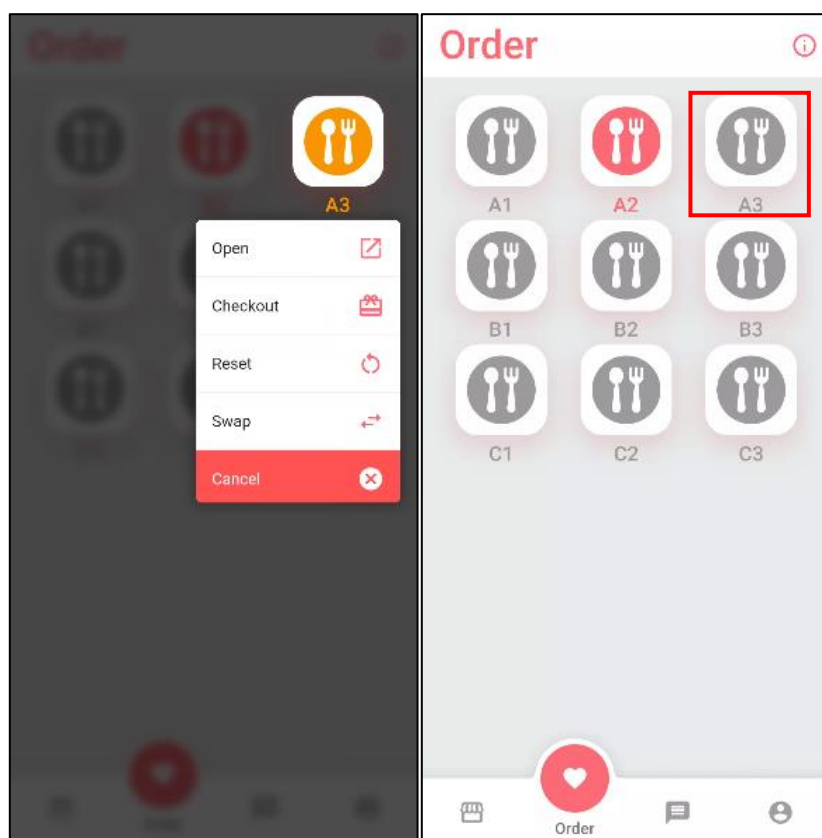
Ứng dụng hiện tại phát triển hình thức thanh toán tiền mặt và các phương thức khác sẽ được phát triển trong tương lai, khi chọn phương thức “Cash Payment” – Thanh toán bằng tiền mặt và chọn “Payment”, giao diện như hình 172.

Hình 148b thể hiện giao diện thanh toán tiền mặt. Hình thức thanh toán này yêu cầu nhập số tiền lớn hơn hoặc bằng tổng tiền và hiển thị số tiền thối (Change) và nút “Confirm” xác nhận thanh toán.

Hình 148c thể hiện giao diện khi người dùng nhập số tiền hợp lệ và trả về số tiền thối. Sau khi người dùng chọn “Confirm” đơn hàng chuyển trạng thái “Paid” và chuyển trạng thái bàn về “Ready”.

2.13. Các giao diện chức năng xử lý đơn hàng khác

Tại giao diện bàn ăn, để thực hiện các tác vụ khác xử lý đơn hàng, người dùng nhấp và giữ một bàn ăn. Giao diện hiển thị các tác vụ như hình 149a.

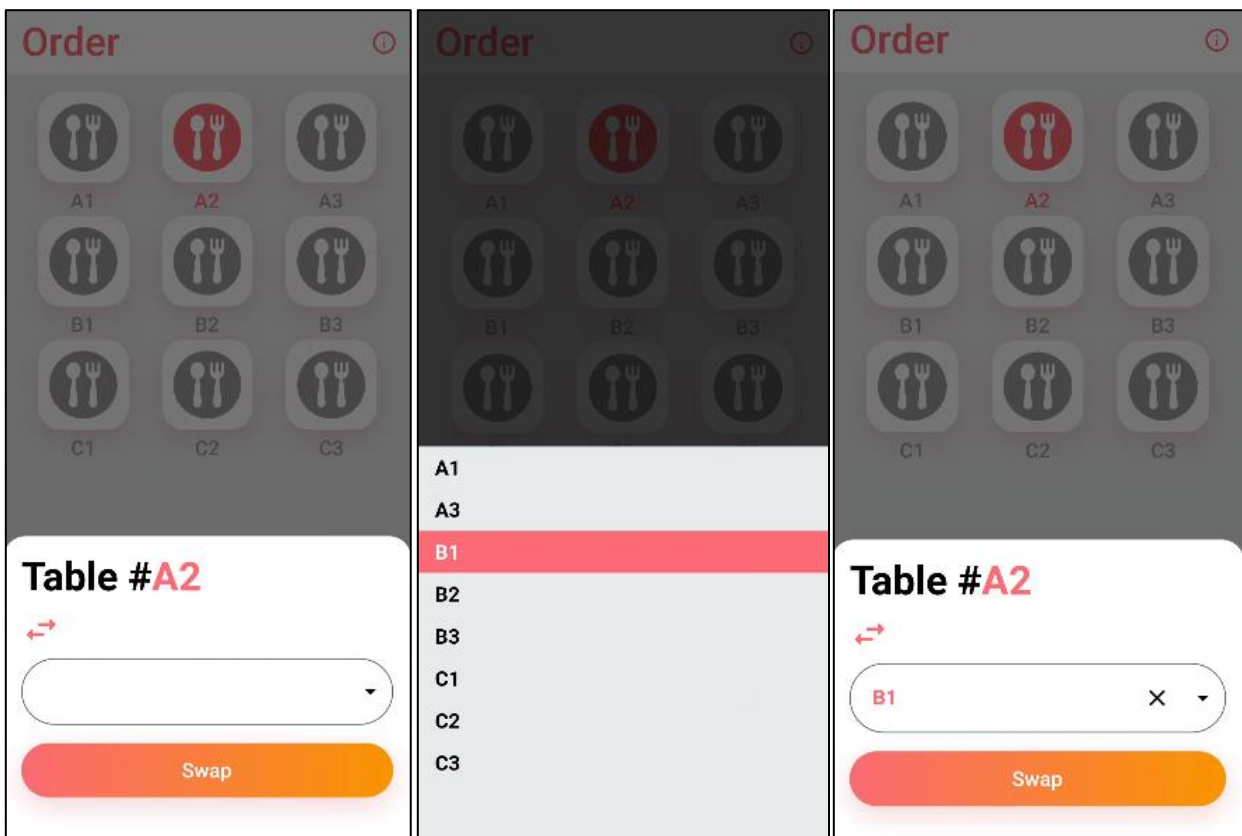


Hình 149 a,b Giao diện các thao tác bàn ăn

- Open: Truy cập đơn hàng.
- Checkout: Thanh toán.
- Reset: Đặt bàn ăn từ trạng thái đang sử dụng về trạng thái bình thường.
- Swap: Chuyển bàn ăn.
- Cancel: Hủy đơn hàng

Open và Checkout giao diện sẽ như các giao diện đơn hàng và thanh toán. Chức năng Reset yêu cầu quyền quản lý, sau khi chọn reset trạng thái bàn ăn trở về như cũ, giao diện như hình 149b.

Giao diện Swap như hình 150a.

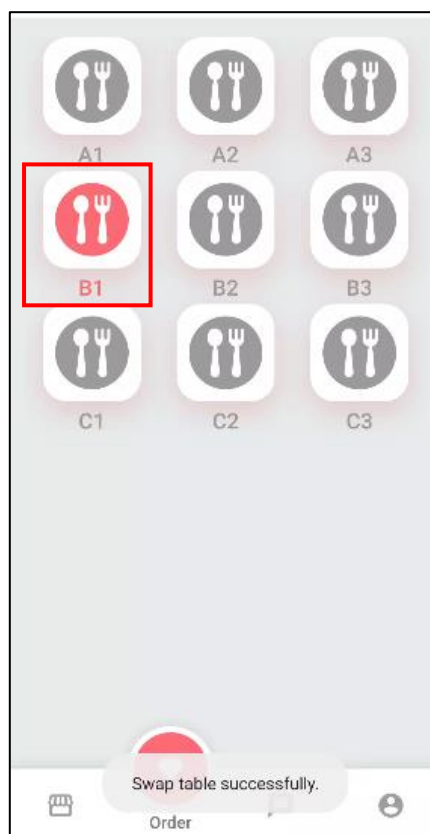


Hình 150 a,b,c Giao diện chuyển bàn ăn

Giao diện Swap thể hiện tên bàn ăn và thanh Dropdown bao gồm những bàn ăn còn trống, giao diện như hình 150b.

Sau khi chọn một bàn ăn đích giao diện hiển thị như hình 150c.

Người dùng có thể chọn “X” để hủy sự lựa chọn B1. Để thực hiện chuyển bàn, người dùng chọn Swap, kết quả hiển thị như hình 151.

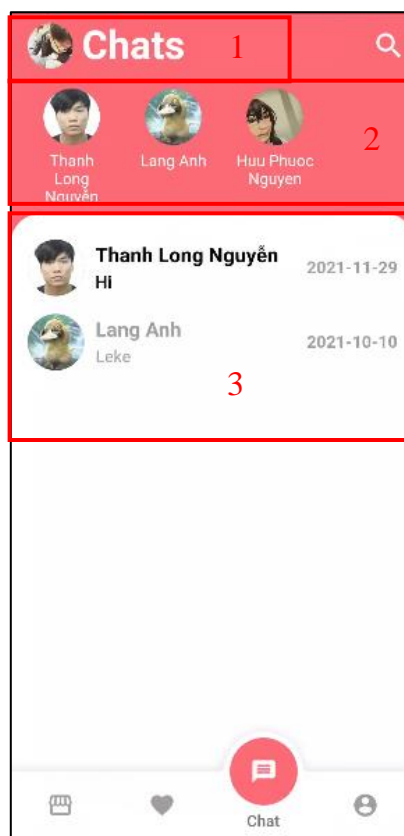


Hình 151 Giao diện chuyển bàn ăn thành công

Chức năng Cancel – Hủy đơn hàng yêu cầu quyền quản lý. Khi người dùng chọn Cancel hệ thống sẽ đưa trạng thái đơn hàng về Cancelled và trạng thái bàn ăn về Ready.

2.14. Giao diện Chat (Nhắn tin)

Giao diện Chat hiển thị khi người dùng chuyển tab “Chat” có icon chat. Giao diện chat như hình 152.



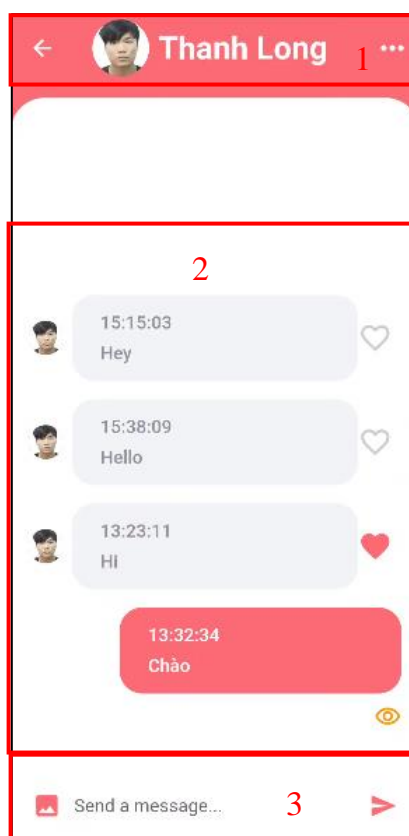
Hình 152 Giao diện nhắn tin

Giao diện chat được chia thành 3 khối như hình 152:

1. Ảnh đại diện người dùng.
2. Các người dùng khác hiển thị với tên và hình ảnh.
3. Giao diện các người dùng vừa nhắn tin với hình ảnh, tên, tin nhắn và ngày tháng nhắn tin. Đối với cách hiển thị tên và tin nhắn được in đậm là những tin nhắn mà người dùng chưa xem.

Để nhắn tin với người dùng khác, người dùng có thể chọn vào người dùng khác tại khối (2) hoặc (3).

Giao diện chi tiết tin nhắn như hình 153



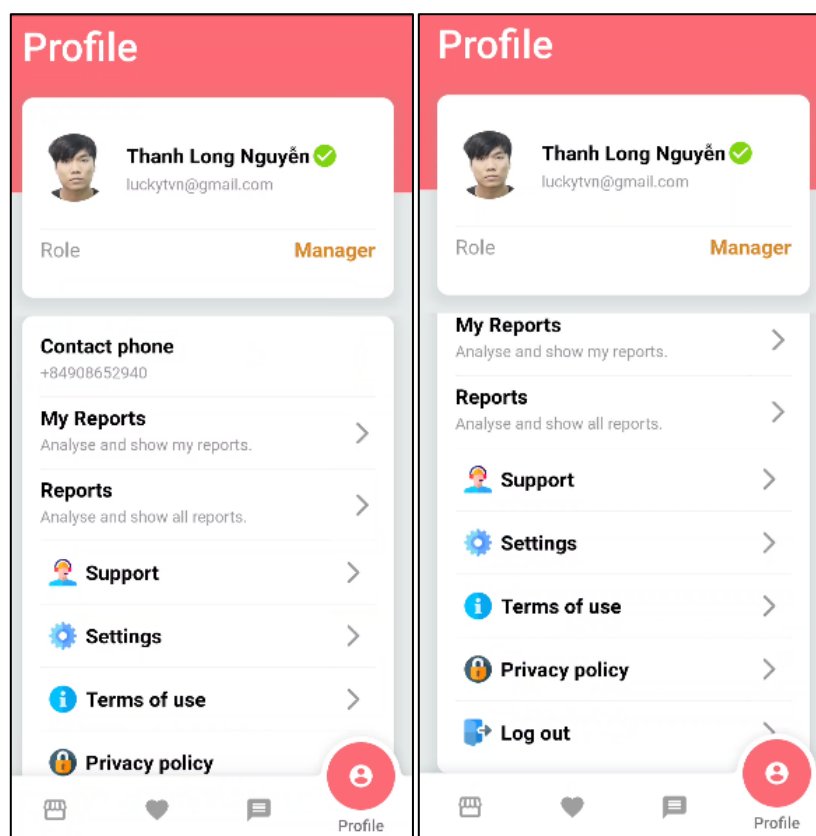
Hình 153 Giao diện chi tiết tin nhắn

Giao diện hình 181 được chia thành 3 khối:

1. Bao gồm nút quay lại ←, tên người dùng và hình đại diện.
2. Tin nhắn được hiển thị theo danh sách thời gian sớm nhất từ dưới đi lên. Tin nhắn gửi được hiển thị phía bên phải với nền màu hồng, tin nhắn nhận hiển thị bên trái với màu nền trắng kèm theo ảnh đại diện người gửi. Bên cạnh những tin nhắn nhận có nút trái tim, người dùng có thể nhấn để đánh dấu “Thích” tin nhắn đó. Trái tim có nền màu hồng được đánh dấu là đã thích. Tin nhắn được gửi khi có icon con mắt 👁 nghĩa là người dùng nhận tin nhắn đã đọc tin nhắn.
3. Khung nhập tin nhắn và phím gửi với icon ➤

2.15. Giao diện Profile

Người dùng chọn tab “Profile” để chuyển hướng đến giao diện Profile như hình 154a và 154b.

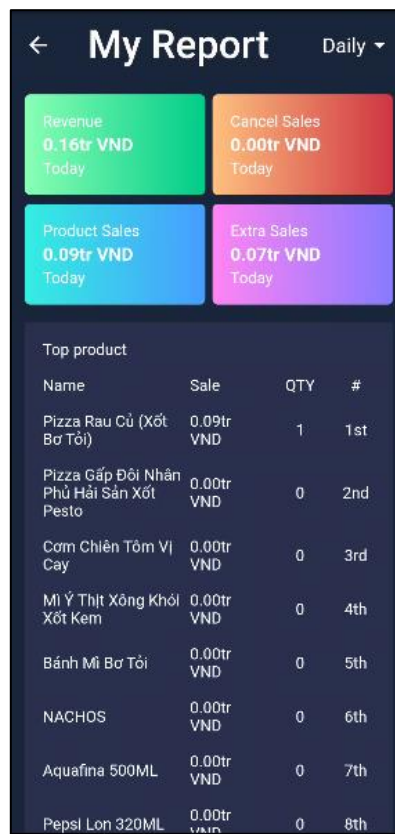


Hình 154 a,b Giao diện Profile

Như hình 183 giao diện Profile sẽ hiển thị các thông tin như: ảnh đại diện, tên người dùng, email, quyền người dùng (Quyền Manager hoặc Staff – Quản lý hoặc nhân viên), số điện thoại và các chức năng như “My Reports”, “Reports” và “Log out”.

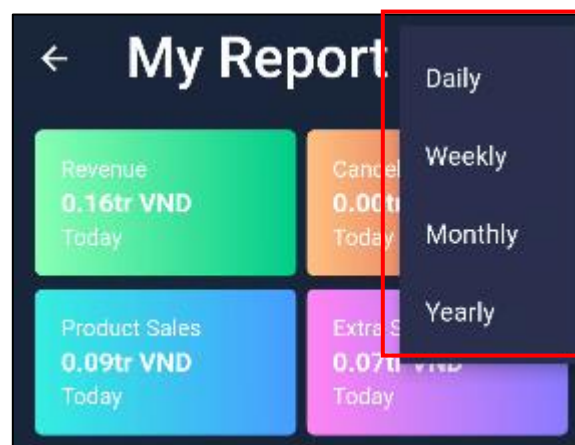
2.16. Giao diện My Reports

Tại giao diện Profile người dùng chọn “My Reports” để hiển thị giao diện như hình 155.



Hình 155 Giao diện báo cáo cá nhân

Như hình 155, giao diện hiển thị bao gồm Header với nút quay lại ←, mốc thời gian bao gồm: Daily (Ngày hôm nay), Weekly (Tuần này), Monthly (Tháng này) và Yearly (Năm này) như hình 156, các thông tin chung của doanh thu như Revenue (Tổng doanh thu), Cancel Sales (Tổng tiền đơn hàng hủy), Product Sales (Tổng doanh thu bán món ăn) và Extra Sales (Tổng doanh thu bán món gọi thêm) và danh sách bảng xếp hạng các món ăn hiển thị theo tên, tổng tiền bán được, tổng số lượng và xếp hạng. Tất cả những thông tin trên là báo cáo doanh thu của người dùng.



Hình 156 Giao diện mốc thời gian

2.17. Giao diện Reports

Chức năng Reports tương tự như My Reports nhưng yêu cầu quyền quản lý, giao diện tương tự My Reports nhưng được bổ sung thêm một số thông tin như hình 157a và 157b.



Hình 157 a,b Giao diện báo cáo

Như hình 157a và 157b, giao diện Reports có thêm hai chức năng thống kê doanh thu theo biểu đồ và thống kê bảng xếp hạng người dùng bán hàng (Top Seller). Top Seller hiển thị theo danh sách gồm các thông tin tên, hình ảnh, tổng sale và xếp hạng.

Biểu đồ hiển thị gồm trục X là thông tin 6 tháng gần nhất, trục Y là các mốc doanh thu và đường biểu đồ hiển thị sự chuyển biến. Có thể xem biểu đồ ở chế độ xem trung bình doanh thu bằng cách nhấp vào “avg” trên biểu đồ như hình 158.



Hình 158 Giao diện biểu đồ

CHƯƠNG VI: KẾT LUẬN

1. Kết quả đạt được

- ✓ Vận dụng kiến thức về ngôn ngữ Python vào dự án thực tế.
- ✓ Thực hành và nghiên cứu sâu vào framework Django.
- ✓ Nắm được quy tắc hoạt động của Restful API.
- ✓ Hiểu được cách thức điều hành một Server.
- ✓ Vận dụng kiến thức về lập trình ứng dụng di động vào dự án.
- ✓ Thực hành và nghiên cứu sâu vào đa nền tảng.
- ✓ Hiểu rõ phương thức lập trình Flutter.
- ✓ Phân tích, thiết kế hệ thống hoàn thiện chi tiết áp dụng công nghệ trong quá trình phục vụ món ăn tại nhà hàng.
- ✓ Xây dựng website và ứng dụng di động đầy đủ thông tin, dễ thao tác truy cập.
- ✓ Xây dựng và chuẩn hóa dữ liệu đảm bảo tính đúng đắn.
- ✓ Giao diện quản lý dễ nhìn, rõ ràng dễ sử dụng.

2. Hướng phát triển

- ✓ Xây dựng website giới thiệu cho hệ thống.
- ✓ Tối ưu hóa cách hiển thị và phương thức nhập liệu của website quản trị.
- ✓ Nâng cấp giao diện thân thiện, dễ sử dụng cho người dùng.
- ✓ Hỗ trợ thanh toán bằng các phương thức online và thẻ tín dụng.
- ✓ Hỗ trợ tính năng chat theo nhóm.

Do kiến thức hạn hẹp và kinh nghiệm còn nhiều hạn chế nên bài báo cáo này không tránh khỏi những sai sót. Rất mong nhận được những đóng góp ý kiến quý báu của thầy cô trường Đại học Bà Rịa – Vũng Tàu.

Một lần nữa em xin chân thành cảm ơn quý thầy cô trong khoa Công nghệ kỹ thuật- Nông nghiệp công nghệ cao và đặc biệt em xin gửi đến thầy Phan Ngọc Hoàng, người đã tận tình hướng dẫn, giúp đỡ em hoàn thành đề tài đồ án tốt nghiệp lời cảm ơn sâu sắc nhất.

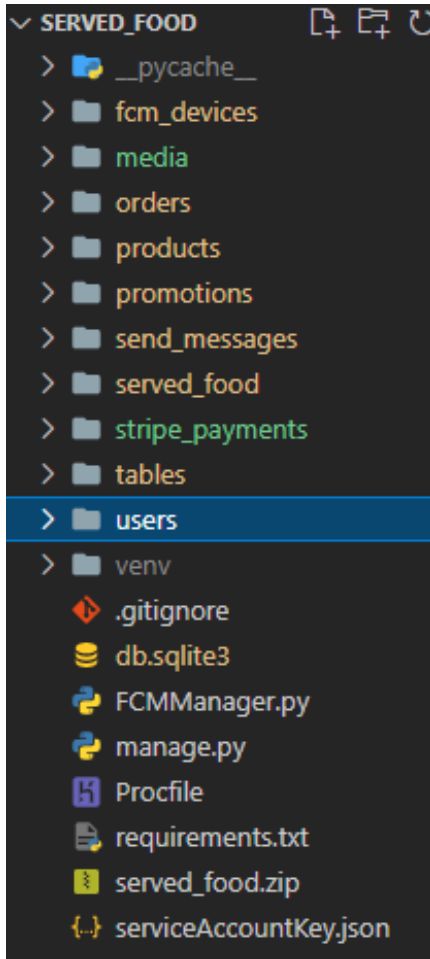
TÀI LIỆU THAM KHẢO

- [1] Vũ Việt Anh - “Tìm hiểu về ngôn ngữ Dart - Phần I” , 21/04/2019 lấy từ URL:
<https://viblo.asia/p/tim-hieu-ve-ngon-ngu-dart-phan-i-bJzKmykwK9N>
- [2] VU NGOC TUAN – “Giới thiệu về Flutter”, 20/03/2018 lấy từ URL:
<https://viblo.asia/p/gioi-thieu-ve-flutter-bWrZnNxrZxw>
- [3] Poman (2020) - ‘HọcPython Để Làm Gì Cho Đời?’, 15/12/2020 Lấy từ URL:
<https://codelearn.io/sharing/hoc-python-de-lam-gi-cho-doi>
- [4] Thủy Nguyễn - “Django là gì và lý do vì sao nên sử dụng Django trong thiết kế web”, 29/03/2021 lấy từ URL:
<https://bizfly.vn/techblog/django-la-gi.html>
- [5] Admin – “Bài 1: SQLite là gì? Ưu và nhược điểm của SQLite - Học SQLite cơ bản”, lấy từ URL:
<https://vncoder.vn/bai-hoc/sqlite-la-gi-uu-va-nhuoc-diem-cua-sqlite-182>
- [6] Mát Bão – “Firebase là gì? Giải pháp lập trình không cần Backend từ Google”, 25/02/2021 lấy từ URL:
<https://wiki.matbao.net/firebase-la-gi-giai-phap-lap-trinh-khong-can-backend-tu-google/#firebase-la-gi>
- [7] Nghi Can – “Cùng tìm hiểu về Firebase Cloud Firestore”, 27/12/2017 lấy từ URL:
<https://viblo.asia/p/cung-tim-hieu-ve-firebase-cloud-firestore-ByEZk32YZQ0>

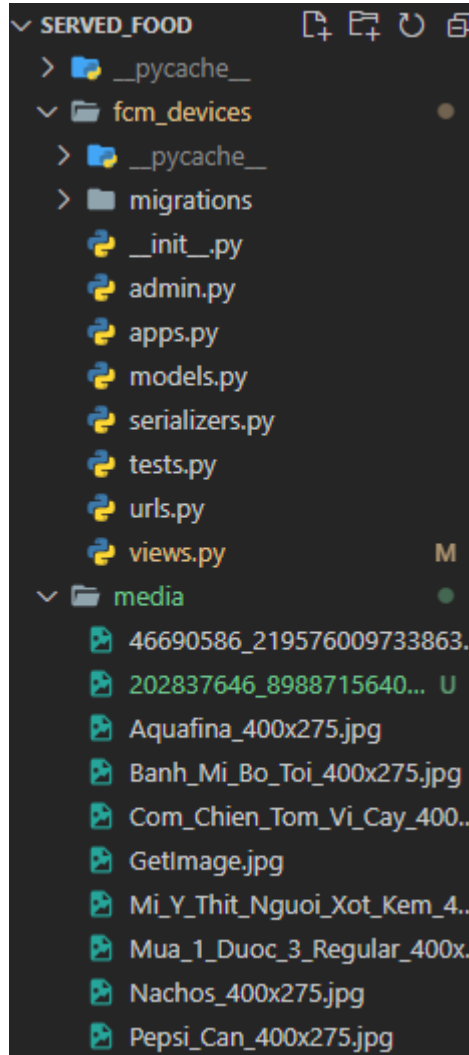
PHỤ LỤC

1. Sơ đồ cấu trúc thư mục

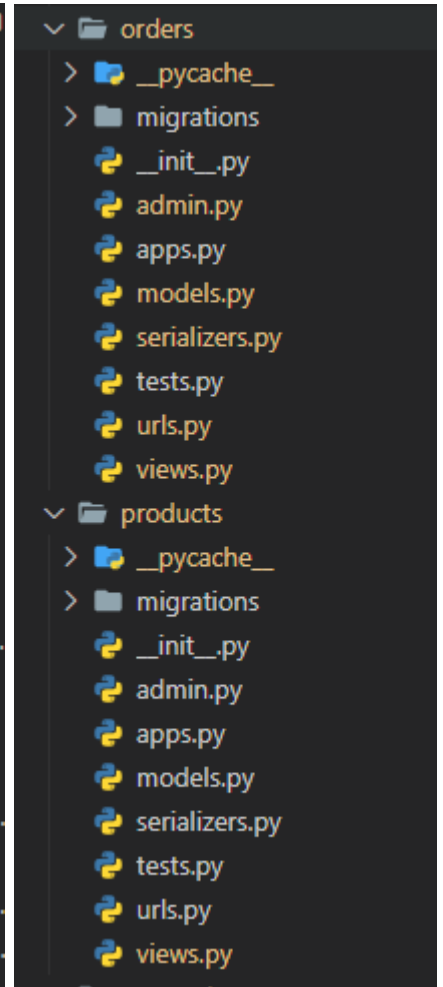
1.1. Website quản trị



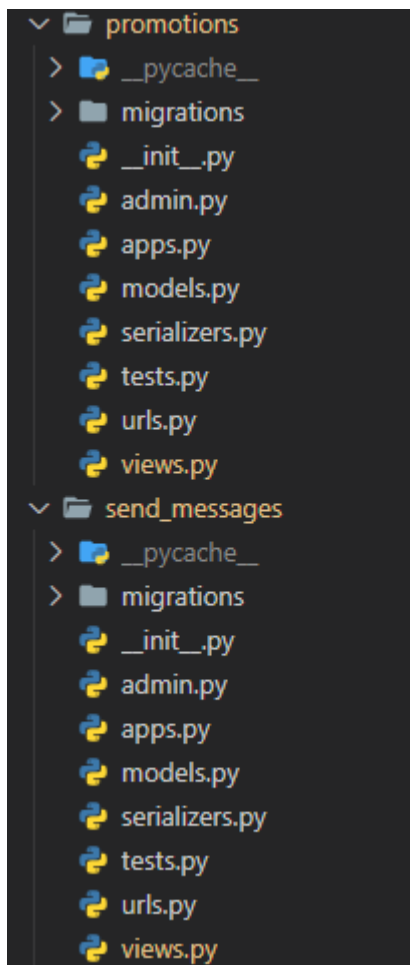
Hình 161 Cấu trúc thư mục website quản trị a



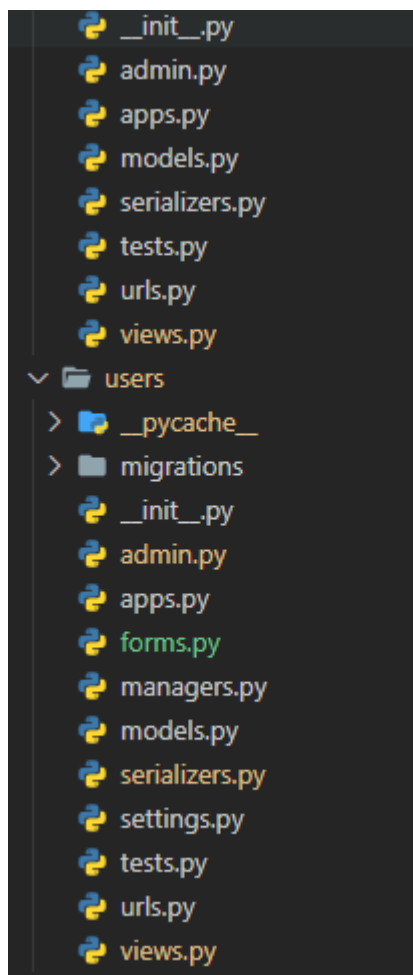
Hình 160 Cấu trúc thư mục website bán trị b



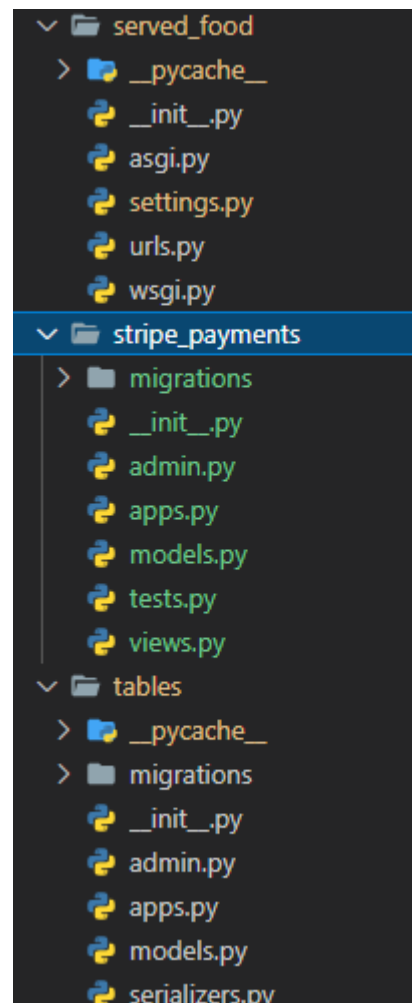
Hình 159 Cấu trúc thư mục website quản trị c



Hình 162 Cấu trúc thư mục website quản trị d



Hình 163 Cấu trúc thư mục website quản trị e

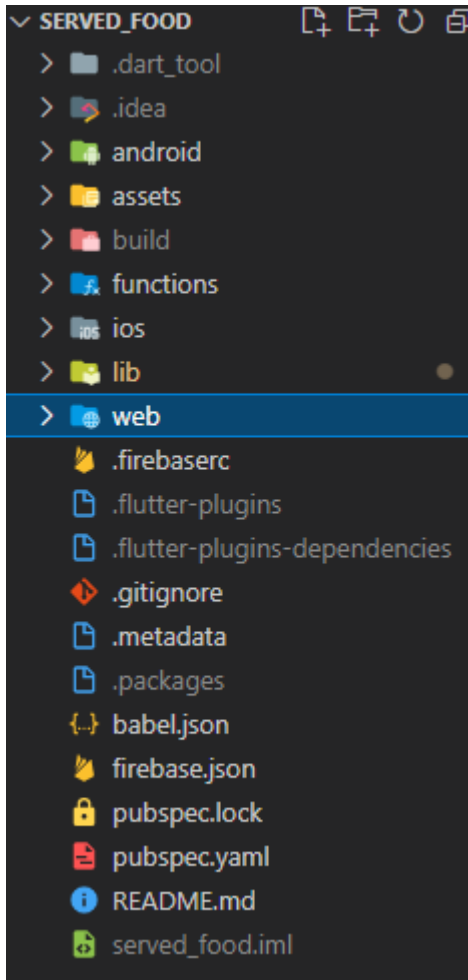


Hình 164 Cấu trúc thư mục website quản trị f

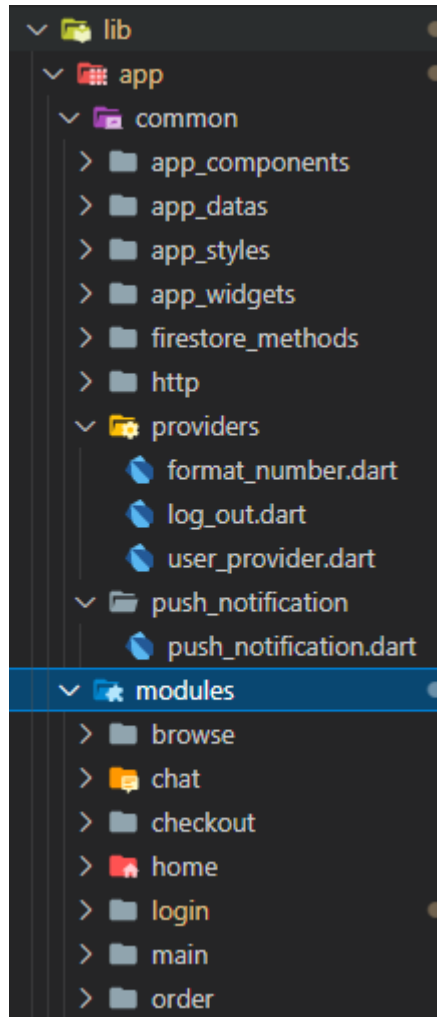
Bảng 39 Giải thích cấu trúc thư mục website quản trị

STT	Tên thư mục	Ý nghĩa
1	media	Thư mục chứa hình ảnh
2	venv	Thư mục chứa môi trường, các thư viện
3	migrations	Thư mục chứa các file chuyên đổi dữ liệu

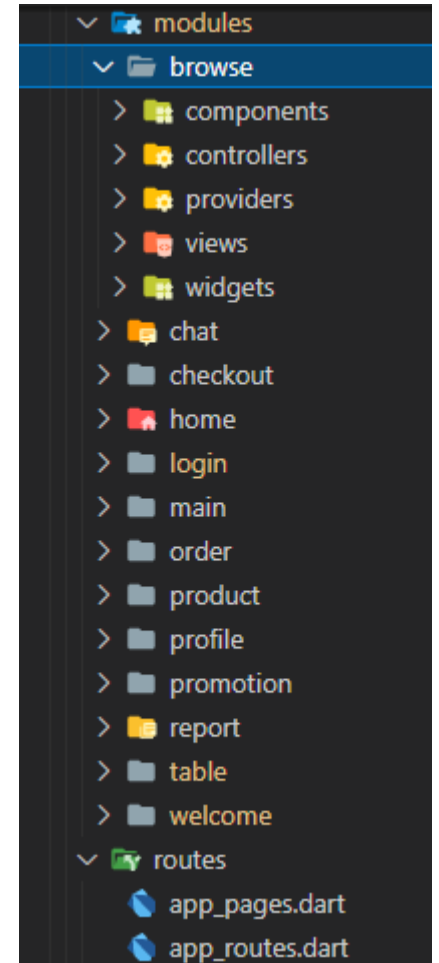
1.2. Ứng dụng di động



Hình 167 Cấu trúc thư mục ứng dụng di động a



Hình 166 Cấu trúc thư mục ứng dụng di động b



Hình 165 Cấu trúc thư mục ứng dụng di động c

Bảng 40 Giải thích cấu trúc thư mục ứng dụng di động

STT	Tên thư mục	Ý nghĩa
1	android	Thư mục chứa các file cấu hình android
2	ios	Thư mục chứa các file cấu hình ios
3	build	Thư mục chứa các file xây dựng apk
4	lib	Thư mục chứa các file code ứng dụng
5	common	Thư mục chứa các file code được sử dụng chung
6	modules	Thư mục chứa các file code theo màn hình

1. Code xử lý

1.1. Website quản trị

1.1.1. File models.py

```
from django.core.checks.messages import Error
from fcm_devices.models import FCMDevice
from django.db import models
from django.db.models.signals import post_delete, post_save, pre_save
from django.dispatch import receiver
from tables.models import Table
from products.models import Extra, Product, ProductVariationOption, VariationOption
import FCMManager as fcm
import logging
from django.contrib.auth import get_user_model

User = get_user_model()

logger = logging.getLogger(__name__)

class Order(models.Model):
    table = models.ForeignKey(Table, on_delete=models.CASCADE, related_name='table')
    paid_by = models.ForeignKey(User, on_delete=models.CASCADE, related_name='user',
blank=True, null=True)
    status = models.CharField(max_length=100, choices=[('serving', 'Serving'), ('paid', 'Paid'),
('cancelled', 'Cancelled')], default='serving')
    order_total_price = models.PositiveIntegerField(default=0)
    order_total_price_record = models.PositiveIntegerField(default=0)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now_add=True)

    def _order_total_price(self):
        order_items = OrderItem.objects.filter(order = self)
        total_price = 0
        for item in order_items:
            total_price += item.order_item_price
        return total_price

    order_total_price = property(_order_total_price)

    def __str__(self):
        return self.table.name
```

```

@receiver(pre_save, sender=Order)
def on_change(sender, instance, **kwargs):
    if instance.id is None:
        table = Table.objects.get(id = instance.table.id)
        table.status = 'ordered'
        table.save()
    else:
        try:
            fcm_devices = FCMDevice.objects.all()
            tokens = []
            previous = Order.objects.get(id = instance.id)

            for item in fcm_devices:
                tokens.append(item.registration_token)
            if previous is not None:
                if previous.table != instance.table:
                    previous_table = Table.objects.get(id = previous.table.id)
                    previous_table.status = 'ready'
                    previous_table.save()
                    instance_table = Table.objects.get(id = instance.table.id)
                    instance_table.status = 'ordered'
                    instance_table.save()
                    logger.error('In table change')
                    fcm.sendPush('Chuyển bàn', 'Đã chuyển bàn "' + str(previous.table) + '" thành
''' + str(instance.table) + '''', tokens)

                if previous.status != instance.status:
                    instance.order_total_price_record = instance.order_total_price
                    table = Table.objects.get(id = instance.table.id)
                    table.status = 'ready'
                    table.is_in_use = False
                    table.save()
                    logger.error(table.status)
                    order_items = OrderItem.objects.filter(order = instance)
                    for order_item in order_items:
                        extras_total_price = 0
                        if order_item.extras.count() != 0:
                            for extra in order_item.extras.all():
                                extras_total_price += extra.price
                                order_item.order_extra_total_price_record = extras_total_price *
order_item.quantity
                                order_item.order_product_total_price_record = order_item.product_variation_option.price*
order_item.quantity
                                order_item.order_item_price_record = order_item.order_item_price
                                order_item.save()

```



```

        if instance.status == 'paid':
            fcm.sendPush(str('Quan ly'), 'Đã thanh toán bàn '+ str(previous.table)
+ '' , tokens)
        elif instance.status == 'cancelled':
            order_items = OrderItem.objects.filter(order = instance)
            for order_item in order_items:
                order_item.is_active = False;
                order_item.save()
            fcm.sendPush('Quản lý', 'Đã hủy bàn '+ str(previous.table) + '' ,
tokens)

    except NameError:
        pass

class OrderItem(models.Model):
    order = models.ForeignKey(Order, on_delete=models.CASCADE, related_name='order_items')
    product = models.ForeignKey(Product, on_delete=models.CASCADE, null=True,
related_name='product')
    product_variation_option = models.ForeignKey(ProductVariationOption,
on_delete=models.CASCADE, related_name='product_variation_option')
    order_item_variation_options = models.ManyToManyField(VariationOption,
related_name='order_item_variation_options')
    extras = models.ManyToManyField(Extra, related_name='extras', null=True, blank=True)
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    quantity = models.PositiveIntegerField(default=1)
    note = models.TextField(null=True, blank=True)
    is_active = models.BooleanField(default=True)
    order_item_price = models.PositiveIntegerField(default=0)
    order_product_total_price_record = models.PositiveIntegerField(default=0)
    order_extra_total_price_record = models.PositiveIntegerField(default=0)
    order_item_price_record = models.PositiveIntegerField(default=0)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now_add=True)

    def _get_order_item_price(self):
        if self.is_active is True:
            extras_total_price = 0
            if self.extras.count() != 0:
                for extra in self.extras.all():
                    extras_total_price += extra.price
            return (self.product_variation_option.price + extras_total_price) * self.quantity
        else:
            return 0

```

```

order_item_price = property(_get_order_item_price)

def __str__(self):
    return self.product.name

@receiver(pre_save, sender=OrderItem)
def on_change(sender, instance, **kwargs):
    order = Order.objects.get(id = instance.order.id)

    fcm_devices = FCMDevice.objects.all()
    tokens = []
    if order is not None:

        for item in fcm_devices:
            if item.is_active:
                tokens.append(item.registration_token)
    if instance.id is None:
        try:
            fcm.sendPush(instance.user.first_name + ' ' + instance.user.last_name, 'Đặt món "' +
str(instance.product_variation_option) + '" cho bàn "' + str(order.table) + '"', tokens)
        except NameError:
            logger.error(NameError)
    else:
        previous = OrderItem.objects.get(id = instance.id)
        if previous is not None:
            try:
                if not instance.is_active:
                    extras_total_price = 0
                    if instance.extras.count() != 0:
                        for extra in instance.extras.all():
                            extras_total_price += extra.price

                instance.order_product_total_price_record = extras_total_price *
instance.quantity
                instance.order_product_total_price_record =
instance.product_variation_option.price* instance.quantity
                instance.order_item_price_record = previous.order_item_price
                if order.status == "serving":
                    fcm.sendPush("Quản lý", 'Hủy món "' +
+ str(previous.product_variation_option) + '" cho bàn "' + str(order.table) + '"', tokens)

            except NameError:
                logger.error(NameError)

```

1.1.2. File admin.py

```
from django.contrib import admin
from orders.models import Order, OrderItem
from import_export.admin import ExportActionMixin
class OrderItemInline(ExportActionMixin, admin.TabularInline):
    exclude = (
        ('product_variation_option', 'order_product_total_price_record', 'order_extra_total_price_record',
        'order_extra_total_price_record')
    )
    model = OrderItem

class OrderAdmin(ExportActionMixin, admin.ModelAdmin):
    list_display = ('table', 'paid_by',
    'status', 'order_total_price', 'order_total_price_record']
    search_fields = ['table']
    inlines = [OrderItemInline]

admin.site.register(Order, OrderAdmin)
```

1.1.3. File serializers.py

```
from django.db.models import fields
from django_filters import filters
from rest_framework import serializers
from orders import models
from tables.serializers import TableSerializer
from users.serializers import UserSerializer
from products.serializers import ExtraSerializer, ProductVariationOptionSerializer,
ProductSerializer, VariationOptionSerializer

class OrderItemSerializer(serializers.ModelSerializer):
    product_variation_option = ProductVariationOptionSerializer(many=False)
    user = UserSerializer(many=False)
    order_item_variation_options = VariationOptionSerializer(many=True)
    product = ProductSerializer(many=False)
    extras = ExtraSerializer(many=True)
    class Meta:
        model = models.OrderItem
        fields = (
            ['id', 'order', 'user', 'product', 'product_variation_option', 'order_item_variation_options', 'extras',
            'quantity', 'note', 'order_item_price', 'order_item_price_record', 'order_product_total_price_record',
            'order extra total price record', 'is active', 'created at', 'updated at']
```

```

class OrderSerializer(serializers.ModelSerializer):
    order_items = OrderItemSerializer(many=True)
    class Meta:
        model = models.Order
        fields = ['id', 'table', 'status', 'order_items',
, 'order_total_price', 'order_total_price_record', 'created_at', 'updated_at']

class OrderItemCreateSerializer(serializers.ModelSerializer):
    class Meta:
        model = models.OrderItem
        fields =
['id', 'order', 'user', 'product', 'product_variation_option', 'order_item_variation_options', 'extr
as', 'quantity', 'note', 'order_item_price', 'order_item_price_record', 'order_product_total_price
record', 'order_extra_total_price_record', 'is_active', 'created_at', 'updated_at']

class OrderCreateSerializer(serializers.ModelSerializer):

    class Meta:
        model = models.Order
        fields = ['id', 'table', 'status', 'order_items', 'created_at', 'updated_at']

class OrderRecordSerializer(serializers.ModelSerializer):

    class Meta:
        model = models.Order
        fields = ['id', 'table', 'status', 'order_items', 'order_total_price_record', 'created_at',
'updated_at']

class OrderItemRecordSerializer(serializers.ModelSerializer):

    class Meta:
        model = models.OrderItem
        fields = ['id', 'is_active', 'product', 'user',
, 'order_product_total_price_record', 'order_extra_total_price_record', 'order_item_price_record',
'quantity', 'updated_at']

```

1.1.4. File views.py

```
from logging import Logger
import logging
from django.shortcuts import render
from django_filters import filters
from rest_framework import generics
from rest_framework import pagination
from orders import models
from orders import serializers
from rest_framework.filters import SearchFilter, OrderingFilter
from django_filters.rest_framework.backends import DjangoFilterBackend
from rest_framework.response import Response
from rest_framework.authentication import SessionAuthentication, BasicAuthentication
from rest_framework.permissions import IsAuthenticated
from rest_framework.response import Response
logger = logging.getLogger(__name__)
class OrderList(generics.ListCreateAPIView):
    permission_classes = [IsAuthenticated]
    queryset = models.Order.objects.all()
    serializer_class = serializers.OrderSerializer
    filter_backends = [DjangoFilterBackend]
    filter_fields = ['table', 'status']

class OrderCreateList(generics.ListCreateAPIView):
    permission_classes = [IsAuthenticated]
    queryset = models.Order.objects.all()
    serializer_class = serializers.OrderCreateSerializer

class OrderDetail(generics.RetrieveUpdateDestroyAPIView):
    permission_classes = [IsAuthenticated]
    queryset = models.Order.objects.all()
    serializer_class = serializers.OrderSerializer

class OrderCreateDetail(generics.RetrieveUpdateDestroyAPIView):
    permission_classes = [IsAuthenticated]
    queryset = models.Order.objects.all()
    serializer_class = serializers.OrderCreateSerializer

class OrderItemList(generics.ListCreateAPIView):
    permission_classes = [IsAuthenticated]
    queryset = models.OrderItem.objects.all()
    serializer_class = serializers.OrderItemSerializer
    filter_backends = [DjangoFilterBackend, OrderingFilter]
```

```

class OrderItemCreateList(generics.ListCreateAPIView):
    permission_classes = [IsAuthenticated]
    queryset = models.OrderItem.objects.all()
    serializer_class = serializers.OrderItemCreateSerializer

class OrderItemDetail(generics.RetrieveUpdateDestroyAPIView):
    permission_classes = [IsAuthenticated]
    queryset = models.OrderItem.objects.all()
    serializer_class = serializers.OrderItemSerializer

class OrderItemCreateDetail(generics.RetrieveUpdateDestroyAPIView):
    permission_classes = [IsAuthenticated]
    queryset = models.OrderItem.objects.all()
    serializer_class = serializers.OrderItemCreateSerializer

class OrderRecordPagination(pagination.PageNumberPagination):
    def get_paginated_response(self, data):
        total_sales = 0
        total_cancelled_sales = 0
        orders = models.Order.objects.all()
        cancelled_count = 0
        paid_count = 0
        for order in data:
            if order['status'] == 'paid':
                total_sales += order['order_total_price_record']
                paid_count += 1
            elif order['status'] == 'cancelled':
                total_cancelled_sales += order['order_total_price_record']
                cancelled_count += 1
        return Response({
            'next': self.get_next_link(),
            'previous': self.get_previous_link(),
            'count': self.page.paginator.count,
            'cancelled_count': cancelled_count,
            'paid_count': paid_count,
            "total_sales": total_sales,
            'total_cancel_sales': total_cancelled_sales,
            'results': data,
        })

class OrderRecord(generics.ListCreateAPIView):
    permission_classes = [IsAuthenticated]
    queryset = models.Order.objects.all()
    serializer_class = serializers.OrderRecordSerializer

```

```

pagination_class = OrderRecordPagination
    filter_backends = [DjangoFilterBackend]
    filterset_fields = {
        'updated_at':['gte', 'lte'], 'status':['exact'], 'table':['exact']
    }
class OrderItemRecordPagination(pagination.PageNumberPagination):
    def get_paginated_response(self, data):
        total_sales = 0
        total_cancelled_sales = 0
        cancelled_count = 0
        paid_count = 0
        total_product_sales = 0
        total_product_cancelled_sales = 0
        total_extra_sales = 0
        total_extra_cancelled_sales = 0
        for orderItem in data:
            if orderItem['is_active'] == True:
                total_product_sales += orderItem['order_product_total_price_record']
                total_sales += orderItem['order_item_price_record']
                total_extra_sales += orderItem['order_extra_total_price_record']
                paid_count += 1
            else:
                total_cancelled_sales += orderItem['order_item_price_record']
                total_product_cancelled_sales += orderItem['order_product_total_price_record']
                total_extra_cancelled_sales += orderItem['order_extra_total_price_record']
                cancelled_count += 1
        return Response({
            'next': self.get_next_link(),
            'previous': self.get_previous_link(),
            'count': self.page.paginator.count,
            'cancelled_count': cancelled_count,
            'paid_count': paid_count,
            "total_sales": total_sales,
            'total_cancelled_sales': total_cancelled_sales,
            'total_product_sales': total_product_sales,
            'total_product_cancelled_sales': total_product_cancelled_sales,
            'total_extra_sales': total_extra_sales,
            'total_extra_cancelled_sales': total_extra_cancelled_sales,
            'results': data,
        })

```

2.1.5 File urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.OrderList.as_view()),
    path('<int:pk>', views.OrderDetail.as_view()),
    path('order_items', views.OrderItemList.as_view()),
    path('order_items/<int:pk>', views.OrderItemDetail.as_view()),
    path('create_order', views.OrderCreateList.as_view()),
    path('create_order/<int:pk>', views.OrderCreateDetail.as_view()),
    path('create_order_item', views.OrderItemCreatelist.as_view()),
    path('create_order_item/<int:pk>', views.OrderItemCreateDetail.as_view()),
    path('order_records', views.OrderRecord.as_view()),
    path('order_item_records', views.OrderItemRecord.as_view())
]
```

1.2. Ứng dụng di động

1.2.1. File models.dart

```
import 'dart:convert';

OrderModel orderModelFromJson(String str) =>
    OrderModel.fromJson(json.decode(str));

String orderModelToJson(OrderModel data) => json.encode(data.toJson());

class OrderModel {
  OrderModel({
    this.id,
    this.table,
    this.status,
    this.orderItems,
    this.orderTotalPrice,
    this.orderTotalPriceRecord,
    this.createdAt,
    this.updatedAt,
  });

  int id;
  int table;
  String status;
  List<OrderItem> orderItems;
```



```

int orderTotalPrice;
int orderTotalPriceRecord;
DateTime createdAt;
DateTime updatedAt;

factory OrderModel.fromJson(Map<String, dynamic> json) => OrderModel(
  id: json["id"],
  table: json["table"],
  status: json["status"],
  orderItems: List<OrderItem>.from(
    json["order_items"].map((x) => OrderItem.fromJson(x))),
  orderTotalPrice: json["order_total_price"],
  orderTotalPriceRecord: json["order_total_price_record"],
  createdAt: DateTime.parse(json["created_at"]),
  updatedAt: DateTime.parse(json["updated_at"]),
);

Map<String, dynamic> toJson() => {
  "id": id,
  "table": table,
  "status": status,
  "order_items": List<dynamic>.from(orderItems.map((x) => x.toJson())),
  "order_total_price": orderTotalPrice,
  "order_total_price_record": orderTotalPriceRecord,
  "created_at": createdAt.toIso8601String(),
  "updated_at": updatedAt.toIso8601String(),
};
}

class OrderItem {
OrderItem({
  this.id,
  this.order,
  this.user,
  this.product,
  this.productVariationOption,
  this.orderItemVariationOptions,
  this.extras,
  this.quantity,
  this.note,
  this.orderItemPrice,
  this.orderItemPriceRecord,
  this.orderProductTotalPriceRecord,

```

```

this.orderExtraTotalPriceRecord,
    this.isActive,
    this.createdAt,
    this.updatedAt,
});

int id;
int order;
User user;
Product product;
ProductVariationOption productVariationOption;
List<Extra> orderItemVariationOptions;
List<Extra> extras;
int quantity;
String note;
int orderItemPrice;
int orderItemPriceRecord;
int orderProductTotalPriceRecord;
int orderExtraTotalPriceRecord;
bool isActive;
DateTime createdAt;
DateTime updatedAt;

factory OrderItem.fromJson(Map<String, dynamic> json) => OrderItem(
    id: json["id"],
    order: json["order"],
    user: User.fromJson(json["user"]),
    product: Product.fromJson(json["product"]),
    productVariationOption:
        ProductVariationOption.fromJson(json["product_variation_option"]),
    orderItemVariationOptions: List<Extra>.from(
        json["order_item_variation_options"].map((x) => Extra.fromJson(x))),
    extras: List<Extra>.from(json["extras"].map((x) => Extra.fromJson(x))),
    quantity: json["quantity"],
    note: json["note"],
    orderItemPrice: json["order_item_price"],
    orderItemPriceRecord: json["order_item_price_record"],
    orderProductTotalPriceRecord: json["order_product_total_price_record"],
    orderExtraTotalPriceRecord: json["order_extra_total_price_record"],
    isActive: json["is_active"],
    createdAt: DateTime.parse(json["created_at"]),
    updatedAt: DateTime.parse(json["updated_at"]),
);

```

```

Map<String, dynamic> toJson() => {
    "id": id,
    "name": name,
    "description": description,
    "category": category.toJson(),
    "labels": List<dynamic>.from(labels.map((x) => x)),
    "variations": List<dynamic>.from(variations.map((x) => x.toJson())),
    "image": image,
    "product_variation_options":
        List<dynamic>.from(productVariationOptions.map((x) => x.toJson())),
    "created_at": createdAt.toIso8601String(),
    "updated_at": updatedAt.toIso8601String(),
};
}

class ProductVariationOption {
    ProductVariationOption({
        this.id,
        this.price,
        this.variationOptions,
        this.createdAt,
        this.updatedAt,
    });

    int id;
    int price;
    List<Extra> variationOptions;
    DateTime createdAt;
    DateTime updatedAt;

    factory ProductVariationOption.fromJson(Map<String, dynamic> json) =>
        ProductVariationOption(
            id: json["id"],
            price: json["price"],
            variationOptions: List<Extra>.from(
                json["variation_options"].map((x) => Extra.fromJson(x))),
            createdAt: DateTime.parse(json["created_at"]),
            updatedAt: DateTime.parse(json["updated_at"]),
        );

    Map<String, dynamic> toJson() => {
        "id": id,
        "price": price,
        "variation_options":

```

```

        List<dynamic>.from(variationOptions.map((x) => x.toJson())),
        "created_at": createdAt.toIso8601String(),
        "updated_at": updatedAt.toIso8601String(),
    });
}

class Variation {
    Variation({
        this.id,
        this.name,
        this.description,
        this.variationOptions,
    });

    int id;
    String name;
    String description;
    List<Extra> variationOptions;

    factory Variation.fromJson(Map<String, dynamic> json) => Variation(
        id: json["id"],
        name: json["name"],
        description: json["description"],
        variationOptions: List<Extra>.from(
            json["variation_options"].map((x) => Extra.fromJson(x))),
    );

    Map<String, dynamic> toJson() => {
        "id": id,
        "name": name,
        "description": description,
        "variation_options":
            List<dynamic>.from(variationOptions.map((x) => x.toJson())),
    };
}

class User {
    User({
        this.id,
        this.password,
        this.lastLogin,
        this.email,
        this.firstName,

```

```

this.firstName,
  this.lastName,
  this.image,
  this.contactPhone,
  this.isSuperuser,
  this.isStaff,
  this.isActive,
  this.dateJoined,
  this.table,
  this.groups,
  this.userPermissions,
});

int id;
String password;
DateTime lastLogin;
String email;
String firstName;
String lastName;
String image;
String contactPhone;
bool isSuperuser;
bool isStaff;
bool isActive;
DateTime dateJoined;
dynamic table;
List<dynamic> groups;
List<dynamic> userPermissions;

factory User.fromJson(Map<String, dynamic> json) => User(
  id: json["id"],
  password: json["password"],
  lastLogin: DateTime.parse(json["last_login"]),
  email: json["email"],
  firstName: json["first_name"],
  lastName: json["last_name"],
  image: json["image"],
  contactPhone: json["contact_phone"],
  isSuperuser: json["is_superuser"],
  isStaff: json["is_staff"],
  isActive: json["is_active"],
  dateJoined: DateTime.parse(json["date_joined"]),
  table: json["table"],

```

```

groups: List<dynamic>.from(json["groups"].map((x) => x)),
  userPermissions:
    List<dynamic>.from(json["user_permissions"].map((x) => x)),
);

Map<String, dynamic> toJson() => {
  "id": id,
  "password": password,
  "last_login": lastLogin.toIso8601String(),
  "email": email,
  "first_name": firstName,
  "last_name": lastName,
  "image": image,
  "contact_phone": contactPhone,
  "is_superuser": isSuperuser,
  "is_staff": isStaff,
  "is_active": isActive,
  "date_joined": dateJoined.toIso8601String(),
  "table": table,
  "groups": List<dynamic>.from(groups.map((x) => x)),
  "user_permissions": List<dynamic>.from(userPermissions.map((x) => x)),
};
}

```

1.2.2. File controllers.dart

```

import 'dart:convert';

import 'package:fluttertoast/fluttertoast.dart';
import 'package:get/get.dart';
import 'package:served_food/app/modules/order/models/order_model.dart';
import 'package:served_food/app/modules/order/providers/order_provider.dart';
import 'package:served_food/app/modules/table/providers/table_provider.dart';
import 'package:shared_preferences/shared_preferences.dart';

class OrderController extends GetxController {
  var isDataProcessing = false.obs;
  var isDataError = false.obs;
  var dataError = ''.obs;
  var lstOrder = List<dynamic>.empty(growable: true).obs;
  var tableID = ''.obs;
  var order = new OrderModel().obs;
  @override
  void onInit() {
    super.onInit();
  }
}

```

```

import 'dart:convert';

import 'package:fluttertoast/fluttertoast.dart';
import 'package:get/get.dart';
import 'package:served_food/app/modules/order/models/order_model.dart';
import 'package:served_food/app/modules/order/providers/order_provider.dart';
import 'package:served_food/app/modules/table/providers/table_provider.dart';
import 'package:shared_preferences/shared_preferences.dart';

class OrderController extends GetxController {
  var isDataProcessing = false.obs;
  var isDataError = false.obs;
  var dataError = ''.obs;
  var lstOrder = List<dynamic>.empty(growable: true).obs;
  var tableID = ''.obs;
  var order = new OrderModel().obs;
  @override
  void onInit() {
    super.onInit();
    updateTableID(Get.arguments['id'].toString());
  }

  void backToTable(var arguments) async {
    SharedPreferences pref = await SharedPreferences.getInstance();
    if (pref.containsKey('carts')) {
      final String cartItemsString = await pref.getString('carts');
      if (cartItemsString == '[]') {
        updateTable(arguments);
      } else {
        Fluttertoast.showToast(msg: 'Please confirm the cart.');
      }
    } else {
      updateTable(arguments);
    }
  }

  void updateTable(var arguments) {
    try {
      dynamic table = jsonEncode({
        'id': arguments['id'],
        'name': arguments['name'],
        'is_active': arguments['is_active'],
        'is_in_use': false,
        'status': arguments['status']
      });
    }
  }
}

```

```

});
    TableProvider().updateTable(arguments['id'].toString(), table).then(
      (response) {
        Get.back();
      }, onError: (e) {
        if (e == 'Not Found') {
          Fluttertoast.showToast(msg: 'Table not found');
        } else {
          Fluttertoast.showToast(msg: e);
        }
      });
} catch (e) {
  print(e);
}
}

void updateTableID(var id) {
  tableID.value = id;
  getOrderDetail();
}

void getOrderDetail() {
  try {
    isDataProcessing(true);
    OrderProvider().getOrderDetail(tableID.value).then((response) {
      isDataProcessing(false);
      if (response.length != 0) {
        order.value = OrderModel.fromJson(response[0]);
      } else {
        order.value = null;
      }
      isDataError(false);
    }, onError: (error) {
      isDataProcessing(false);
      isDataError(true);
      dataError(error);
    });
  } catch (e) {
    isDataProcessing(false);
    isDataError(true);
    dataError(e);
  }
}
}
}

```


1.2.3. File providers.dart

```
import 'package:get/get_connect.dart';
import 'package:served_food/app/common/app_datas/get_request_url.dart';
import 'package:served_food/app/common/app_datas/request_headers.dart';

class OrderProvider extends GetConnect {
  Future<List<dynamic>> getOrderItems(String id) async {
    try {
      final response = await get(GetRequestUrl.ORDER_ITEMS + id,
        headers: await RequestHeaders().getRequestHeader());
      if (response.status.hasError) {
        return Future.error(response.statusText);
      } else {
        return response.body;
      }
    } catch (e) {
      return Future.error(e.toString());
    }
  }

  Future<dynamic> getOrderDetail(String id) async {
    try {
      final response = await get(GetRequestUrl.ORDER_DETAIL + id,
        headers: await RequestHeaders().getRequestHeader());
      if (response.status.hasError) {
        return Future.error(response.statusText);
      } else {
        return response.body['results'];
      }
    } catch (e) {
      return Future.error(e.toString());
    }
  }

  Future<dynamic> createOrder(dynamic body) async {
    try {
      final response = await post(GetRequestUrl.CREATE_ORDER, body,
        headers: await RequestHeaders().getRequestHeader());
      if (response.status.hasError) {
        return Future.error(response.statusText);
      } else {
        return response.body;
      }
    } catch (e) {
      return Future.error(e.toString());
    }
  }
}
```

```

} catch (e) {
    return Future.error(e.toString());
}
}

Future<dynamic> updateOrder(String id, dynamic body) async {
    try {
        final response = await put(GetRequestUrl.CREATE_ORDER_DETAIL + id, body,
            headers: await RequestHeaders().getRequestHeader());
        if (response.status.hasError) {
            return Future.error(response.statusText);
        } else {
            return response.body;
        }
    } catch (e) {
        return Future.error(e.toString());
    }
}
}
}

```

1.2.4. File views.dart

```

import 'package:avatar_glow/avatar_glow.dart';
import 'package:bouncing_widget/bouncing_widget.dart';
import 'package:flutter/material.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:get/get.dart';

import 'package:served_food/app/common/app_styles/index.dart';
import 'package:served_food/app/common/app_widgets/btn_text_white_widget.dart';
import 'package:served_food/app/common/app_widgets/gradient_btn_widget.dart';
import 'package:served_food/app/common/providers/format_number.dart';
import 'package:served_food/app/modules/order/components/order_item_component.dart';
import 'package:served_food/app/modules/order/components/order_price_component.dart';
import 'package:served_food/app/modules/order/controllers/order_controller.dart';
import 'package:served_food/app/modules/order/controllers/shopping_cart_controller.dart';
import 'package:served_food/app/modules/order/widgets/cart_item.dart';
import 'package:served_food/app/modules/order/widgets/order_clipper.dart';
import 'package:served_food/app/routes/app_routes.dart';

class OrderView extends GetView<OrderController> {
    @override

```

```

class OrderView extends GetView<OrderController> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        leading: BackButton(
          onPressed: () {
            controller.backToTable(Get.arguments);
          },
        ),
        title: Text(
          Get.arguments['name'],
          style: kBodyTextStyle.copyWith(
            fontSize: kTitleTextSize,
            color: Colors.white,
            fontWeight: FontWeight.w500),
        ),
        actions: [
          TextButton(
            onPressed: () {
              Get.toNamed(AppRoutes.BROWSE);
            },
            child: Padding(
              padding: const EdgeInsets.only(right: kPadding),
              child: Text(
                'Order',
                style: kBodyTextStyle.copyWith(
                  fontSize: kSubtitleTextSize,
                  color: Colors.white,
                  fontWeight: FontWeight.w400),
              ),
            ),
          ),
        ],
        backgroundColor: kBtnColorStart,
      ),
      body: Stack(
        children: [
          ClipPath(
            clipper: OrderClipper(),
            child: Container(
              color: kBtnColorStart,
            ),
          ),
        ],
      ),
      Positioned(

```

```

Positioned(
  top: kPadding,
  child: Column(
    children: [
      OrderPriceComponent(controller: controller),
      SizedBox(
        height: kPadding,
      ),
      OrderItemComponent(controller: controller)
    ],
  ),
),
Positioned(
  bottom: 0,
  child: Row(
    children: [
      Container(
        width: Get.width - 48 - kPadding * 2,
        padding: const EdgeInsets.only(
          top: kPadding, bottom: kPadding, left: kPadding),
        child: GradientBtnWidget(
          child: BtnTextWhiteWidget(
            text: 'Checkout',
          ),
          onTap: () {
            if (controller.order.value != null) {
              Get.toNamed(AppRoutes.CHECKOUT, arguments: [
                Get.arguments['name'],
                controller.order.value
              ]);
            } else {
              Fluttertoast.showToast(msg: 'No item to check out');
            }
          },
        ),
      ),
      BouncingWidget(
        duration: Duration(milliseconds: 100),
        scaleFactor: 1.5,
        onPressed: () {
          Get.bottomSheet(ShoppingCartView(
            arguments: Get.arguments,
          ));
        },
        child: AvatarGlow(

```

```

        child: AvatarGlow(
          glowColor: kBtnColorStart,
          endRadius: 40.0,
          duration: Duration(milliseconds: 1000),
          repeat: true,
          showTwoGlows: true,
          repeatPauseDuration: Duration(milliseconds: 100),
          child: Container(
            width: 48,
            height: 48,
            decoration: BoxDecoration(
              color: kBtnColorEnd,
              borderRadius:
                BorderRadius.all(Radius.circular(48)),
            ),
            child: Icon(
              Icons.shopping_cart,
              color: Colors.white,
            ),
          ),
        ),
      ],
    )),
  ],
),
);
}
}

class ShoppingCartView extends StatelessWidget {
  final dynamic arguments;
  const ShoppingCartView({Key key, this.arguments}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    ShoppingCartController controller = Get.put(ShoppingCartController());
    return Container(
      height: Get.height * 0.7,
      padding: EdgeInsets.all(kPadding),
      decoration: BoxDecoration(
        borderRadius: BorderRadius.only(
          topLeft: Radius.circular(kBorderRadius * 2),
          topRight: Radius.circular(kBorderRadius * 2)),
        color: Colors.white,
      ),
      child: Obx(() {

```

```

child: Obx(() {
  if (controller.isDataProcessing.value) {
    return Center(
      child: SpinKitFadingFour(
        color: kBtnColorStart,
        size: 50,
      ),
    );
  } else {
    return controller.cartItems.length != 0
      ? Stack(children: [
        ListView.separated(
          itemBuilder: (context, index) {
            String extras = '';
            for (var item in controller.cartItems[index].extras) {
              extras = extras + ' $item';
            }
            return CartItem(
              controller: controller,
              index: index,
              title: controller.cartItems[index].title,
              note: controller.cartItems[index].note,
              image: controller.cartItems[index].image,
              quantity: controller.cartItems[index].quantity
                .toString(),
              price: formatNumber(
                controller.cartItems[index].orderItemPrice),
              extras: extras);
          },
          separatorBuilder: (context, index) {
            return Divider();
          },
          itemCount: controller.cartItems.length),
        Positioned(
          bottom: 0,
          width: Get.width - kPadding * 2,
          child: GradientBtnWidget(onTap: () {
            controller.addOrder(arguments);
          }, child: Obx(() {
            if (controller.isAddOrderProcessing.value) {
              return Container(
                height: kIconSize,
                width: kIconSize,

```

```

        return Container(
          height: kIconSize,
          width: kIconSize,
          child: CircularProgressIndicator(
            color: Colors.white,
          ),
        );
      } else {
        return BtnTextWhiteWidget(
          text: 'Confirm',
        );
      }
    })))
  ])
  : Center(
    child: Text(
      'Nothing',
      style: kBodyTextStyle.copyWith(
        color: kBtnColorStart,
        fontSize: kSubtitleTextSize,
        fontWeight: FontWeight.bold),
    ),
  );
}
}));
}
}

```