

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**TRẦN PHƯỚC SANG - 52100303
LÊ KHẮC THANH TÙNG - 52100943
NGUYỄN THANH TÚ - 52100349**

IFONEX

**BÁO CÁO CUỐI KỲ
PHÁT TRIỂN
ỨNG DỤNG WEB VỚI NODEJS**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**TRẦN PHƯỚC SANG - 52100303
LÊ KHẮC THANH TÙNG - 52100943
NGUYỄN THANH TÚ - 52100349**

IFONEX

**BÁO CÁO CUỐI KỲ
PHÁT TRIỂN
ỨNG DỤNG WEB VỚI NODEJS**

**Người hướng dẫn
GV. Vũ Đình Hồng**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn đến toàn thể giảng viên trường Đại học Tôn Đức Thắng nói chung và giảng viên ThS.NCS. Vũ Đình Hồng nói riêng vì đã tạo môi trường học tập, phát triển toàn diện cho chúng em. Ngoài ra, trong suốt quá trình học tập chúng em đã được dạy và tiếp thu nhiều kiến thức bổ ích, quan trọng đối với tương lai của chúng em.

Bằng kiến thức mà chúng em tiếp thu được trong suốt quá trình học tập tại trường để hoàn thiện bài báo cáo này. Tuy còn nhiều thiếu sót cần phải học hỏi và cải thiện trong bài báo cáo nên chúng em mong rằng sẽ được thầy nhận xét, đánh giá và cho ý kiến về bài báo cáo.

Chúng em rất mong có được sự góp ý đến từ thầy cô tại trường để biết được những thiếu sót của bản thân đồng thời cải thiện kỹ năng của chúng em. Cuối cùng, chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô trường Đại học Tôn Đức Thắng và thầy Vũ Đình Hồng.

TP. Hồ Chí Minh, ngày 1 tháng 12 năm 2023

Tác giả

(Ký tên và ghi rõ họ tên)

Trần Phước Sang

Lê Khắc Thanh Tùng

Nguyễn Thanh Tú

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng em xin cam đoan đây là đồ án của riêng nhóm chúng em và được sự hướng dẫn khoa học của ThS.NCS. Vũ Đình Hồng. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chúng em thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Báo cáo của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 1 tháng 12 năm 2023

Tác giả

(Ký tên và ghi rõ họ tên)

Trần Phước Sang

Lê Khắc Thanh Tùng

Nguyễn Thanh Tú

WEB SOCKET SERVER DÙNG EXPRESSJS

TÓM TẮT

Phần 1: Mở đầu và tổng quan đề tài

Phần 2: Cơ sở lý thuyết

Phần 3: Phân tích hệ thống

Phần 4: Thiết kế chức năng hệ thống

Phần 5: Tổng kết

MỤC LỤC

| | |
|--|-------------|
| DANH MỤC HÌNH VẼ | vi |
| DANH MỤC BẢNG BIỂU | vii |
| DANH MỤC CÁC CHỮ VIẾT TẮT..... | viii |
| CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI..... | 1 |
| 1. Lý do chọn đề tài | 1 |
| 2. Mục tiêu thực hiện đề tài | 1 |
| CHƯƠNG 2. CƠ SỞ LÝ THUYẾT..... | 2 |
| 2.1 Ngôn ngữ và nền tảng lập trình..... | 2 |
| 2.2 Thư viện và Framework | 3 |
| CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG..... | 4 |
| 3.1 Cơ sở dữ liệu | 4 |
| 3.2 Phân tích và thiết kế | 5 |
| 3.2.1 Các tác nhân trong hệ thống..... | 5 |
| 3.2.2 Sơ đồ Usecase tổng quát..... | 5 |
| 3.2.3 Đặc tả chức năng..... | 6 |
| 3.2.4 Sơ đồ Activity | 41 |
| 3.2.5 Sơ đồ Sequence | 43 |
| CHƯƠNG 4. XÂY DỰNG CHỨC NĂNG HỆ THỐNG | 45 |
| 4.1 Tạo tài khoản và gửi email người dùng | 45 |
| 4.2 Mở/ Khóa tài khoản..... | 46 |
| 4.3 Tạo hóa đơn sản phẩm và thanh toán..... | 47 |
| 4.4 Thống kê..... | 51 |

| | |
|---|-----------|
| 4.5 Kiểm tra quyền | 52 |
| CHƯƠNG 5. GIAO DIỆN CHỨC NĂNG HỆ THỐNG | 55 |
| 5.1 Chức năng tổng quát | 55 |
| 5.1.1 Thay đổi mật khẩu..... | 55 |
| 5.1.2 Xem thông tin hồ sơ cá nhân và cập nhật ảnh đại diện..... | 56 |
| 5.1.3 Xem danh sách sản phẩm..... | 57 |
| 5.1.4 Xem lịch sử mua hàng của khách hàng | 58 |
| 5.1.5 In hoá đơn | 58 |
| 5.2 Chức năng Admin | 59 |
| 5.2.1 Quản lý danh sách nhân viên..... | 59 |
| 5.2.2 Xem danh sách khách hàng..... | 60 |
| 5.2.3 Thống kê..... | 60 |
| 5.3 Chức năng salesperson | 61 |
| 5.3.1 Tạo đơn hàng, tìm khách hàng và thêm khách hàng mới | 61 |
| CHƯƠNG 6. TỔNG KẾT | 62 |
| 6.1 Kết quả đạt được | 62 |
| 6.2 Những mặt hạn chế | 62 |
| 6.3 Hướng phát triển trong tương lai | 62 |
| TÀI LIỆU THAM KHẢO | 63 |

DANH MỤC HÌNH VẼ

| | |
|---|----|
| Hình 3.1: Mối quan hệ giữa các collection..... | 4 |
| Hình 3.2.4.1: Sơ đồ Activity Tạo tài khoản cho nhân viên bán hàng..... | 41 |
| Hình 5.1.1.1: Giao diện thay đổi mật khẩu cho tài khoản mới đăng nhập lần đầu | 55 |
| Hình 5.1.1.2: Giao diện thay đổi mật khẩu cho tài khoản..... | 56 |
| Hình 5.1.2.1: Giao diện thay Xem thông tin hồ sơ cá nhân và cập nhật ảnh đại diện | 56 |
| Hình 5.1.3.1: Giao diện Xem danh sách sản phẩm của SalesPerson..... | 57 |
| Hình 5.1.3.2: Giao diện Xem danh sách sản phẩm của Admin | 57 |
| Hình 5.1.4.1: Giao diện Xem lịch sử mua hàng của khách hàng..... | 58 |
| Hình 5.1.5.1: Giao diện In hoá đơn | 58 |
| Hình 5.1.5.2: Giao diện In lại hoá đơn ở trang thông tin chi tiết đơn..... | 59 |
| Hình 5.2.1.1: Giao diện Quản lý danh sách nhân viên | 59 |
| Hình 5.2.2.1: Giao diện Xem danh sách khách hàng | 60 |
| Hình 5.2.3.1: Giao diện Thống kê | 60 |
| Hình 5.3.1.1: Giao diện Tạo đơn hàng | 61 |
| Hình 5.3.1.2: Giao diện tra cứu thông tin khách hàng qua số điện thoại | 61 |

DANH MỤC BẢNG BIỂU

| | |
|---|-----------|
| Bảng 2. 3: Bảng danh sách các tác nhân | 5 |
| Hình 3.2.1: Sơ đồ Usecase tổng quát | 5 |
| Bảng 3.2.2.1: Đặc tả Use case Tạo tài khoản cho nhân viên bán hàng | 7 |
| Bảng 3.2.2.2 Đặc tả Đăng nhập | 9 |
| Bảng 3.2.2.3: Đặc tả Đổi mật khẩu | 11 |
| Bảng 3.2.2.4: Xem danh sách nhân viên | 13 |
| Bảng 3.2.2.5: Xem chi tiết nhân viên | 15 |
| Bảng3.2.2.6: Gửi lại email đăng nhập | 17 |
| Bảng 3.2.2.7: Khóa/ Mở khóa tài khoản nhân viên bán hàng..... | 19 |
| Bảng 3.2.2.8: Xem trang thông tin chi tiết | 21 |
| Bảng 3.2.2.9: Xem chi tiết nhân viên | 23 |
| Bảng 3.2.2.10: Đặc tả Use case Tìm kiếm và xem lịch sử giao dịch sản phẩm .. | 25 |
| Bảng 3.2.2.11: Đặc tả Use case Tìm kiếm và xem lịch sử giao dịch sản phẩm .. | 27 |
| Bảng 3.2.2.12: Đặc tả Use case Tìm kiếm sản phẩm | 29 |
| Bảng 3.2.2.13: Đặc tả Use case Thêm sản phẩm..... | 31 |
| Bảng 3.2.2.14: Đặc tả Use case Thêm sản phẩm..... | 33 |
| Bảng 3.2.2.15: Đặc tả Use case Xóa sản phẩm..... | 35 |
| Bảng 3.2.2.16: Đặc tả Use case Thanh toán tại quầy | 38 |
| Bảng 3.2.2.17: Đặc tả Use case Thống kê | 40 |
| Hình 3.2.4.2: Sơ đồ Activity Đăng nhập..... | 42 |
| Hình 3.2.5.1: Sơ đồ Sequence Tạo tài khoản cho nhân viên bán hàng | 43 |
| Hình 3.2.5.2: Sơ đồ Sequence Đăng nhập | 44 |

DANH MỤC CÁC CHỮ VIẾT TẮT

Hbs

Handlebars

CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

1. Lý do chọn đề tài

Hiện nay các cửa hàng bán lẻ điện thoại và phụ kiện ngày càng nhiều, kèm theo đó là nhu cầu mua hàng của khách ngày càng gia tăng. Khách hàng thường lựa chọn các kênh mua hàng cung cấp một dịch vụ và trải nghiệm người dùng nhanh chóng và hiệu quả.

Để đáp ứng việc trên cần phải xây dựng và thiết kế hệ thống thanh toán và quản lý bán hàng một cách nhanh chóng, tối ưu và hiệu quả để đáp ứng được nhu cầu của khách hàng.

IfoneX là một Website dùng để tiến hành thanh toán và quản lý quá trình kinh doanh của một điểm bán hàng. Cung cấp các chức năng cơ bản như quản lý sản phẩm, người dùng và các chức năng thanh toán cho khách hàng, đồng thời có các chức năng thống kê quản lý.

2. Mục tiêu thực hiện đề tài

Xây dựng một trang web cung cấp chức năng Điểm bán hàng (Point of Sale) tại cửa hàng bán lẻ điện thoại và phụ kiện. Người dùng của ứng dụng web là các nhân viên bán hàng và quản trị viên tại cửa hàng điện thoại.

Ứng dụng web này cần cung cấp các chức năng như: giao dịch bán hàng, quản lý sản phẩm, quản lý nhân viên, xem báo cáo và thống kê.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Ngôn ngữ và nền tảng lập trình

HTML (HyperText Markup Language): Sử dụng để xây dựng cấu trúc nội dung của trang web, đặc biệt là thông qua các thẻ như `div` (để nhóm và tổ chức nội dung), `a` (tạo liên kết), `form` (cho nhập liệu và gửi dữ liệu), `table` (hiển thị dữ liệu theo dạng bảng), `p` (để định dạng đoạn văn), `ul` và `li` (tạo danh sách).

CSS (Cascading Style Sheets): Được sử dụng để trang trí và thiết kế giao diện cho trang web. Các thuộc tính như màu sắc, kích thước chữ, vị trí và độ trong suốt của các yếu tố trang web được sử dụng để tạo ra giao diện hấp dẫn và dễ đọc.

JavaScript: ngôn ngữ lập trình chủ yếu cho phía client, giúp tương tác động với người dùng. Thông qua thư viện quan trọng như jQuery hoặc framework như React hỗ trợ truy cập và thay đổi các thành phần HTML, xử lý sự kiện, và làm cho trang web trở nên linh hoạt.

NodeJS: là một nền tảng chạy mã JavaScript ở phía server, cho phép xây dựng ứng dụng web có khả năng mở rộng và hiệu suất cao. Nó thích hợp cho việc xây dựng các ứng dụng thời gian thực và xử lý nhiều kết nối đồng thời.

MongoDB: MongoDB là hệ quản trị cơ sở dữ liệu phi cấu trúc (NoSQL) sử dụng để lưu trữ dữ liệu. Đặc biệt, MongoDB thích hợp cho các ứng dụng có nhu cầu lưu trữ dữ liệu đa dạng, không cố định, và thường xuyên thay đổi, như làm việc với dữ liệu từ các form đặt hàng, thông tin khách hàng, và các bảng giá sản phẩm trong cửa hàng bán lẻ điện thoại.

2.2 Thư viện và Framework

ExpressJS: là một framework web cho NodeJS, giúp xây dựng và quản lý các ứng dụng web dễ dàng. ExpressJS cung cấp các chức năng middleware, routing, và quản lý HTTP requests, giúp việc xây dựng server-side logic trở nên thuận tiện.

Chart.js: là một thư viện JavaScript để tạo đồ thị và biểu đồ tương tác trực quan. Thư viện này thích hợp để hiển thị dữ liệu thống kê và báo cáo trong ứng dụng web của bạn, chẳng hạn như số lượng bán hàng, doanh thu, và xu hướng mua sắm.

Mongoose: là một thư viện ODM (Object Data Modeling) cho MongoDB và NodeJS. Nó giúp xây dựng và quản lý mô hình dữ liệu, cung cấp cách tiếp cận thuận tiện và an toàn khi tương tác với cơ sở dữ liệu MongoDB.

Handlebars: là một hệ thống template engine cho NodeJS, giúp tạo ra các template HTML động dựa trên dữ liệu từ server. Điều này làm cho việc hiển thị dữ liệu trên trang web trở nên linh hoạt và dễ dàng quản lý.

Multer: là một middleware cho NodeJS, được sử dụng để xử lý tải lên (upload) file từ client lên server. Điều này hữu ích khi cần lưu trữ hình ảnh sản phẩm hoặc các tệp tin khác liên quan trong ứng dụng của bạn.

Crypto: được sử dụng để thực hiện các chức năng mã hóa và giải mã dữ liệu. Nó thường được sử dụng để bảo vệ thông tin nhạy cảm, chẳng hạn như mật khẩu người dùng.

Easyinvoice: là một gói NPM cho phép bạn tạo hóa đơn PDF đẹp mắt một cách dễ dàng.

Bootstrap: là một framework cung cấp các thành phần giao diện sẵn có, grid system, và các stylesheet mặc định giúp tối ưu hóa quá trình thiết kế và làm đẹp giao diện của trang web.

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Cơ sở dữ liệu

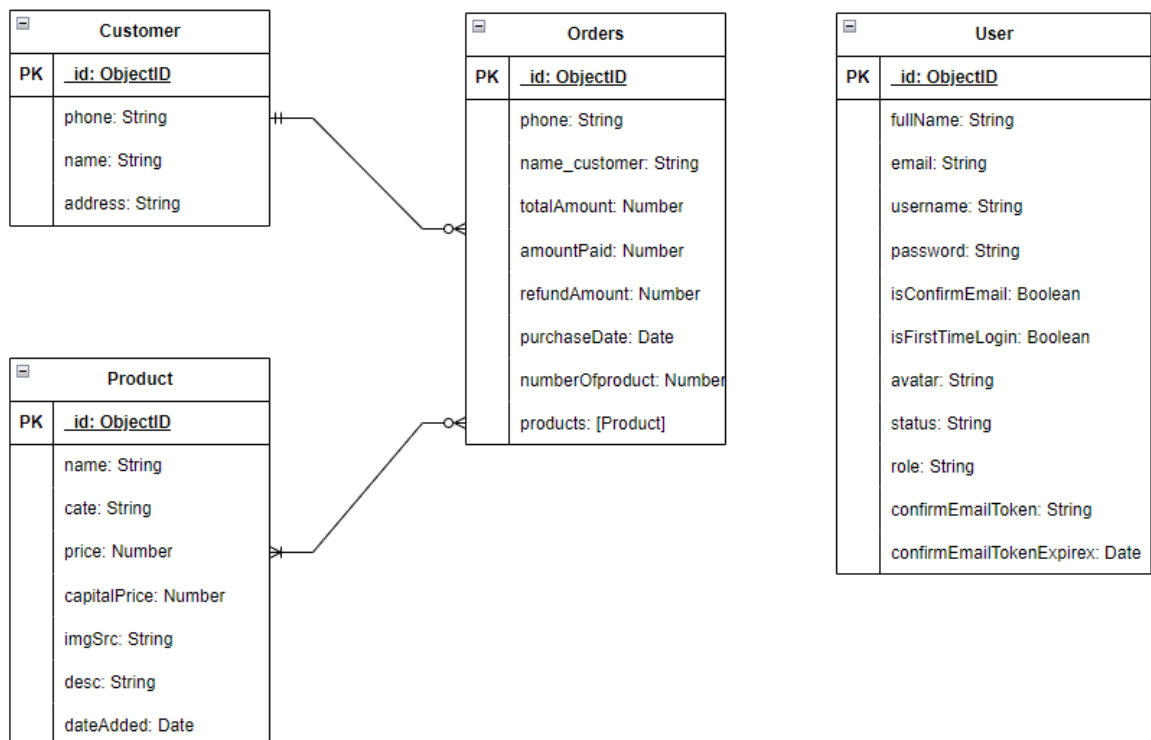
- **User:** Chứa các thông tin của nhân viên bao gồm các thông tin cơ bản như họ tên, email, tên đăng nhập, mật khẩu, trạng thái, quyền của tài khoản và ảnh đại diện của người dùng.

- **Customer:** Bao gồm thông tin của khách hàng như họ tên, số điện thoại, địa chỉ của khách hàng khi mua hàng tại hệ thống.

- **Order:** Chứa thông tin đơn hàng như thông tin khách hàng, danh sách sản phẩm và số lượng sản phẩm, tổng chi phí hoá đơn và ngày mua hàng.

- **Product:** Chứa thông tin của sản phẩm trong cửa hàng như tên sản phẩm, loại sản phẩm, chi phí, mô tả sản phẩm, hình ảnh sản phẩm, ngày thêm vào cửa hàng.

- Sơ đồ ERD:



Hình 3.1: Mối quan hệ giữa các collection

3.2 Phân tích và thiết kế

3.2.1 Các tác nhân trong hệ thống

| Tên actor | Diễn giải |
|--------------------|---|
| Admin | Là người có cấp bậc cao nhất hệ thống Có nhiệm vụ: quản lý nhân viên, xem thống kê,... |
| Nhân viên bán hàng | Là người dùng chủ yếu thêm sản phẩm vào hóa đơn, tạo hóa đơn và hỗ trợ khách hàng thanh toán đơn hàng |

Bảng 2. 3: Bảng danh sách các tác nhân

3.2.2 Sơ đồ Usecase tổng quát



Hình 3.2.1: Sơ đồ Usecase tổng quát

3.2.3 Đặc tả chức năng

3.2.3.1 Use case tạo tài khoản cho nhân viên bán hàng

| | |
|--------------------------|---|
| Use Case ID | UC01 |
| Use Case | Tạo tài khoản cho nhân viên bán hàng |
| Scenario | Admin muốn tạo tài khoản cho nhân viên bán hàng |
| Triggering Event | Admin nhấn vào nút “Thêm nhân viên” ở trang quản lý nhân viên |
| Description | Admin tạo tài tài khoản cho nhân viên bán hàng để nhân viên bán hàng có thể truy cập vào hệ thống |
| Actor(s) | Admin |
| Pre-Condition(s): | Nhân viên chưa có tài khoản. |

| | | |
|---------------------------|---|---|
| Post-Condition(s): | Mail sẽ được gửi đến email của nhân viên khi admin tạo thành công | |
| Flow of Event | Actor | System |
| | 1. Người dùng truy cập vào mục “Thêm nhân viên” | 1.1. Hệ thống hiển thị giao diện “Thêm nhân viên” |
| | 2. Người dùng nhập họ tên và email của nhân viên | |
| | 3. Người dùng nhấn nút “Thêm” | 3.1. Hệ thống lưu thông tin xuống cơ sở dữ liệu đồng thời gửi email đăng nhập đến cho nhân viên |
| Exception | Người dùng nhập email có định dạng không hợp lệ | |

Bảng 3.2.2.1: Đặc tả Use case Tạo tài khoản cho nhân viên bán hàng

3.2.3.2 Use case đăng nhập

| | |
|--------------------------|---|
| Use Case ID | UC02 |
| Use Case | Đăng nhập |
| Scenario | Admin, nhân viên bán hàng muốn đăng nhập vào hệ thống |
| Triggering Event | Admin, nhân viên bán hàng truy cập vào trang web khi chưa đăng nhập thì hệ thống sẽ điều hướng sang trang đăng nhập |
| Description | Admin, nhân viên bán hàng đăng nhập để có thể truy cập vào hệ thống |
| Actor(s) | Admin, nhân viên bán hàng |
| Pre-Condition(s): | Nhân viên bán hàng đã có tài khoản. |

| | | |
|---------------------------|---|---|
| Post-Condition(s): | Hệ thống sẽ điều hướng đến trang chủ | |
| Flow of Event | Actor | System |
| | 1. Người dùng truy cập vào trang “Đăng nhập” | 1.1. Hệ thống hiển thị giao diện “Đăng nhập” |
| | 2. Người dùng nhập tên đăng nhập và mật khẩu | |
| | 3. Người dùng nhấn nút “Đăng nhập” | 3.1. Hệ thống điều hướng đến trang chủ và lưu thông tin của người dùng trong phiên làm việc |
| Exception | Người dùng chưa đăng nhập lần đầu tiên qua email Người dùng chưa đổi mật khẩu lần đầu tiên khi đăng nhập | |

Bảng 3.2.2.2 Đặc tả Đăng nhập

3.2.3.3 Use case đổi mật khẩu

| | |
|---------------------------|--|
| Use Case ID | UC03 |
| Use Case | Đổi mật khẩu |
| Scenario | Admin, nhân viên bán hàng muốn đổi mật khẩu |
| Triggering Event | Admin, nhân viên bán hàng truy cập vào mục “Đổi mật khẩu” trên trang web |
| Description | Admin, nhân viên bán hàng muốn đổi mật khẩu của tài khoản của mình |
| Actor(s) | Admin, nhân viên bán hàng |
| Pre-Condition(s): | Người dùng đã đăng nhập trước đó. |
| Post-Condition(s): | Hệ thống sẽ điều hướng đến trang chủ |

| Flow of Event | Actor | System |
|------------------|--|--|
| | 1. Người dùng truy cập vào trang “Đổi mật khẩu” | 1.1. Hệ thống hiển thị giao diện “Đổi mật khẩu” |
| | 2. Người dùng nhập mật khẩu cũ, mật khẩu mới và mật khẩu xác nhận | |
| | 3. Người dùng nhấn nút “Cập nhật” | 3.1. Hệ thống điều hướng đến trang chủ và cập nhật thông tin xuống cơ sở dữ liệu |
| Exception | <p>Người dùng nhập mật khẩu cũ sai</p> <p>Người dùng nhập mật khẩu mới và mật khẩu xác nhận không khớp</p> | |

Bảng 3.2.2.3: Đặc tả Đổi mật khẩu

3.2.3.4 Use case xem danh sách nhân viên

| | |
|--------------------------|---|
| Use Case ID | UC04 |
| Use Case | Xem danh sách nhân viên |
| Scenario | Admin muốn xem danh sách nhân viên |
| Triggering Event | Admin truy cập vào mục “Quản lý nhân viên” trên trang web |
| Description | Admin muốn xem danh sách nhân viên |
| Actor(s) | Admin |
| Pre-Condition(s): | Người dùng đã đăng nhập trước đó. |

| | | |
|---------------------------|--|--|
| Post-Condition(s): | Hệ thống sẽ hiển thị giao diện trang “Quản lý nhân viên” | |
| Flow of Event | Actor | System |
| | 1. Người dùng truy cập vào trang “Quản lý nhân viên” | 1.1. Hệ thống hiển thị giao diện trang “Quản lý nhân viên” |
| Exception | | |

Bảng 3.2.2.4: Xem danh sách nhân viên

3.2.3.5 Use case xem chi tiết nhân viên

| | |
|--------------------------|---|
| Use Case ID | UC05 |
| Use Case | Xem chi tiết nhân viên |
| Scenario | Admin muốn xem thông tin chi tiết của một nhân viên bán hàng trong hệ thống |
| Triggering Event | Admin nhấn vào tên của một nhân viên bán hàng mà mình muốn xem thông tin chi tiết |
| Description | Admin muốn xem thông tin chi tiết của nhân viên |
| Actor(s) | Admin |
| Pre-Condition(s): | Người dùng đã đăng nhập trước đó. |

| | | |
|---------------------------|---|--|
| Post-Condition(s): | Hệ thống sẽ hiển thị giao diện trang thông tin chi tiết của nhân viên | |
| Flow of Event | Actor | System |
| | 1. Người dùng truy cập vào trang “Quản lý nhân viên” | 1.1. Hệ thống hiển thị giao diện trang “Quản lý nhân viên” |
| | 2. Người dùng nhấn vào tên của nhân viên muốn xem thông tin chi tiết | 2.1. Hệ thống hiển thị trang hiển thị thông tin chi tiết của nhân viên |
| Exception | | |

Bảng 3.2.2.5: Xem chi tiết nhân viên

3.2.3.6 Use case gửi lại email đăng nhập

| | |
|--------------------------|--|
| Use Case ID | UC06 |
| Use Case | Gửi lại email đăng nhập |
| Scenario | Nhân viên bán hàng hết hạn email đăng nhập nên liên hệ với admin để được gửi email mới để đăng nhập lại. |
| Triggering Event | Admin nhấn vào nút “Resend Login Email” của nhân viên mà mình muốn trên giao diện |
| Description | Admin muốn gửi lại email đăng nhập cho nhân viên |
| Actor(s) | Admin |
| Pre-Condition(s): | Người dùng đã đăng nhập trước đó. |

| | | |
|---------------------------|--|--|
| Post-Condition(s): | Hệ thống sẽ gửi lại email đăng nhập cho nhân viên | |
| Flow of Event | Actor | System |
| | 1. Người dùng truy cập vào trang “Quản lý nhân viên” | 1.1. Hệ thống hiển thị giao diện trang “Quản lý nhân viên” |
| | 2. Người dùng nhấn vào nút “Resend Login Email” | 2.1. Hệ thống gửi lại email đăng nhập cho người dùng |
| Exception | | |

Bảng3.2.2.6: Gửi lại email đăng nhập

3.2.3.7 Use case khóa/ mở khóa tài khoản nhân viên bán hàng

| | |
|--------------------------|--|
| Use Case ID | UC07 |
| Use Case | Khóa/ mở khóa tài khoản nhân viên bán hàng |
| Scenario | Admin muốn khóa hoặc mở khóa tài khoản của nhân viên bán hàng |
| Triggering Event | Admin nhấn vào nút “Lock” nếu tài khoản đang không bị khóa hoặc nhấn vào nút “Unlock” nếu tài khoản đang bị khóa của nhân viên |
| Description | Admin muốn khóa/ mở khóa tài khoản của nhân viên bán hàng |
| Actor(s) | Admin |
| Pre-Condition(s): | Người dùng đã đăng nhập trước đó. |

| | | |
|---------------------------|--|--|
| Post-Condition(s): | Hệ thống sẽ lưu thông tin vào cơ sở dữ liệu | |
| Flow of Event | Actor | System |
| | 1. Người dùng truy cập vào trang “Quản lý nhân viên” | 1.1. Hệ thống hiển thị giao diện trang “Quản lý nhân viên” |
| | 2. Người dùng nhấn vào “Lock” nếu tài khoản đang không bị khóa hoặc nhấn vào nút “Unlock” nếu tài khoản đang bị khóa của nhân viên | 2.1. Hệ thống cập nhật thông tin vào cơ sở dữ liệu |
| Exception | | |

Bảng 3.2.2.7: Khóa/ Mở khóa tài khoản nhân viên bán hàng

3.2.3.8 Use case xem trang thông tin chi tiết

| | |
|--------------------------|--|
| Use Case ID | UC08 |
| Use Case | Xem trang thông tin chi tiết nhân viên |
| Scenario | Admin, nhân viên muốn xem thông tin chi tiết mình trên hệ thống |
| Triggering Event | Admin, nhân viên bán hàng nhấn vào mục “View Profile” |
| Description | Admin, nhân viên bán hàng muốn xem thông tin chi tiết của mình trên hệ thống |
| Actor(s) | Admin, nhân viên bán hàng |
| Pre-Condition(s): | Người dùng đã đăng nhập trước đó. |

| | | |
|---------------------------|--|---|
| Post-Condition(s): | Hệ thống sẽ hiển thị giao diện trang thông tin chi tiết của người dùng | |
| Flow of Event | Actor | System |
| | 1. Người dùng truy cập vào mục “View Profile” | 1.1. Hệ thống hiển thị giao diện trang thông tin chi tiết của người dùng gồm ảnh đại diện, tên, quyền |
| Exception | | |

Bảng 3.2.2.8: Xem trang thông tin chi tiết

3.2.3.9 Use case cập nhật ảnh đại diện

| | |
|--------------------------|---|
| Use Case ID | UC09 |
| Use Case | Cập nhật ảnh đại diện |
| Scenario | Admin, nhân viên bán hàng muốn cập nhật ảnh đại diện trên hệ thống |
| Triggering Event | Admin, nhân viên bán hàng vào trang thông tin chi tiết và tải ảnh muốn cập nhật lên |
| Description | Admin, nhân viên bán hàng muốn cập nhật ảnh đại diện |
| Actor(s) | Admin, nhân viên |
| Pre-Condition(s): | Người dùng đã đăng nhập trước đó. |

| | | |
|---------------------------|---|---|
| Post-Condition(s): | Hệ thống sẽ cập nhật lại ảnh đại diện trên trang thông tin chi tiết | |
| Flow of Event | Actor | System |
| | 1. Người dùng truy cập vào mục “View Profile” | 1.1. Hệ thống hiển thị giao diện trang thông tin chi tiết của người dùng gồm ảnh đại diện, tên, quyền |
| | 2. Người dùng chọn ảnh đại diện muốn cập nhật | |
| | 3. Người dùng nhấn nút “Tải” | 3.1. Hệ thống hiển thị giao diện trang thông tin chi tiết của người dùng gồm ảnh đại diện, tên, quyền |
| Exception | | |

Bảng 3.2.2.9: Xem chi tiết nhân viên

3.2.3.10 Use case xem chi tiết đơn hàng

| | |
|---------------------------|---|
| Use Case ID | UC10 |
| Use Case | Tìm kiếm và xem lịch sử mua hàng của khách hàng |
| Scenario | Admin, Quản lý cửa hàng, Nhân viên bán hàng đang cần tìm kiếm thông tin cơ bản sản phẩm. |
| Triggering Event | Người dùng cần tìm thông tin của hóa đơn và nhấn khung tìm kiếm hoặc tìm kiếm qua danh mục. |
| Description | Admin, Quản lý cửa hàng, Nhân viên bán hàng tìm kiếm sản phẩm theo danh mục không biết chính xác mã đơn hàng hoặc nhập thông tin đơn hàng gồm mã và tên khách hàng và nhấn tìm kiếm để tìm ra sản phẩm. |
| Actor(s) | Admin, Quản lý cửa hàng, Nhân viên bán hàng |
| Pre-Condition(s): | <p>Thông tin hóa đơn đã được lưu trữ trong hệ thống quản lý cửa hàng.</p> <p>Người dùng phải nhập vào khung tìm kiếm hoặc chọn theo danh mục .</p> <p>Người dùng phải đăng nhập vào hệ thống</p> |
| Post-Condition(s): | Hiển thị các thông tin chi tiết của hóa đơn. |

| Flow of Event | Actor | System |
|---------------|--|--|
| | <p>1. Người dùng nhập thông tin hoặc chọn vào danh mục.</p> <p>2. Người dùng nhấn vào xem chi tiết đơn hàng.</p> | <p>1.1 Hệ thống tiến hành tìm kiếm .Nếu thông tin trùng khớp hệ thống hiển thị những sản phẩm liên quan tới thông tin nhận được. Hiển thị các thông tin cơ bản.</p> <p>2.1 Hệ thống hiển thị ra các thông tin chi tiết của đơn hàng.</p> |
| Exception | <p>1.1. a Hệ thống không tìm thấy đơn hàng trùng khớp và thông báo “Không tìm thấy sản phẩm”.</p> | |

Bảng 3.2.2.10: Đặc tả Use case Tìm kiếm và xem lịch sử giao dịch sản phẩm

3.2.3.11 Use case Tìm kiếm, xem lịch sử giao dịch của khách hàng

| | |
|---------------------------|---|
| Use Case ID | UC11 |
| Use Case | Tìm kiếm và xem lịch sử mua hàng của khách hàng |
| Scenario | Admin, Quản lý cửa hàng, Nhân viên bán hàng đang cần tìm kiếm thông tin cơ bản sản phẩm. |
| Triggering Event | Người dùng cần tìm thông tin của khách hàng và nhấn khung tìm kiếm hoặc tìm kiếm qua danh mục. |
| Description | Admin, Quản lý cửa hàng, Nhân viên bán hàng tìm kiếm khách theo danh mục không biết chính xác thông tin khách hàng hoặc nhập thông tin khách hàng và nhấn tìm kiếm để tìm ra sản phẩm. |
| Actor(s) | Admin, Quản lý cửa hàng, Nhân viên bán hàng |
| Pre-Condition(s): | <p>Thông tin sản phẩm đã được lưu trữ trong hệ thống quản lý cửa hàng.</p> <p>Người dùng phải nhập vào khung tìm kiếm hoặc chọn theo danh mục .</p> <p>Người dùng phải đăng nhập vào hệ thống</p> |
| Post-Condition(s): | Hiển thị các thông tin chi tiết của sản phẩm. |

| Flow of Event | Actor | System |
|---------------|--|--|
| | <p>1. Người dùng nhập thông tin hoặc chọn vào danh mục.</p> <p>2. Người dùng nhấn vào xem chi tiết sản phẩm.</p> | <p>1.1 Hệ thống tiến hành tìm kiếm .Nếu thông tin trùng khớp hệ thống hiển thị những sản phẩm liên quan tới thông tin nhận được. Hiển thị các thông tin cơ bản.</p> <p>2.1 Hệ thống hiển thị ra các thông tin chi tiết của sản phẩm.</p> |
| Exception | <p>1.1. a Hệ thống không tìm thấy sản phẩm trùng khớp và thông báo “Không tìm thấy sản phẩm”.</p> | |

Bảng 3.2.2.11: Đặc tả Use case Tìm kiếm và xem lịch sử giao dịch sản phẩm

3.2.3.12 Use case Tìm kiếm và xem chi tiết sản phẩm

| | |
|---------------------------|---|
| Use Case ID | UC12 |
| Use Case | Tìm kiếm và xem chi tiết sản phẩm |
| Scenario | Admin, Quản lý cửa hàng, Nhân viên bán hàng đang cần tìm kiếm thông tin cơ bản sản phẩm. |
| Triggering Event | Người dùng cần tìm thông tin của sản phẩm và nhấn khung tìm kiếm hoặc tìm kiếm qua danh mục. |
| Description | Admin, Quản lý cửa hàng, Nhân viên bán hàng tìm kiếm sản phẩm theo danh mục không biết chính xác tên sản phẩm hoặc nhập thông tin sản phẩm và nhấn tìm kiếm để tìm ra sản phẩm. |
| Actor(s) | Admin, Quản lý cửa hàng, Nhân viên bán hàng |
| Pre-Condition(s): | <p>Thông tin sản phẩm đã được lưu trữ trong hệ thống quản lý cửa hàng.</p> <p>Người dùng phải nhập vào khung tìm kiếm hoặc chọn theo danh mục .</p> <p>Người dùng phải đăng nhập vào hệ thống</p> |
| Post-Condition(s): | Hiển thị các thông tin chi tiết của sản phẩm. |

| Flow of Event | Actor | System |
|---------------|--|--|
| | <p>1. Người dùng nhập thông tin hoặc chọn vào danh mục.</p> <p>2. Người dùng nhấn vào xem chi tiết sản phẩm.</p> | <p>1.1 Hệ thống tiến hành tìm kiếm .Nếu thông tin trùng khớp hệ thống hiển thị những sản phẩm liên quan tới thông tin nhận được. Hiển thị các thông tin cơ bản.</p> <p>2.1 Hệ thống hiển thị ra các thông tin chi tiết của sản phẩm.</p> |
| Exception | <p>1.1. a Hệ thống không tìm thấy sản phẩm trùng khớp và thông báo “Không tìm thấy sản phẩm”.</p> | |

Bảng 3.2.2.12: Đặc tả Use case Tìm kiếm sản phẩm

3.2.3.13 Use case Thêm sản phẩm

| | | |
|---------------------------|---|---------------|
| Use Case ID | UC13 | |
| Use Case | Thêm sản phẩm | |
| Scenario | Người quản trị muốn thêm một sản phẩm mới vào hệ thống quản lý chuỗi. | |
| Triggering Event | Admin cần thêm sản phẩm mới và hệ thống. | |
| Description | Chức năng thêm sản phẩm cho phép người quản trị hệ thống thêm một sản phẩm mới vào hệ thống quản lý khi cần kinh doanh thêm một sản phẩm mới. | |
| Actor(s) | Admin | |
| Pre-Condition(s): | Admin đã đăng nhập vào hệ thống quản lý cửa hàng. Sản phẩm chưa tồn tại trong hệ thống. | |
| Post-Condition(s): | Sản phẩm mới được thêm vào hệ thống. Thông tin của sản phẩm mới được lưu trữ trong cơ sở dữ liệu của hệ thống. | |
| Flow of Event | Actor | System |

| | | |
|------------------|---|--|
| | <p>1. Admin chọn tính năng thêm sản phẩm trong giao diện quản lý chuỗi.</p> <p>2. Admin nhập thông tin sản phẩm mới vào các trường thông tin cần thiết.</p> <p>3. Admin nhấn nút "Lưu" để lưu thông tin của sản phẩm mới.</p> <p>4. Admin hoàn tất việc thêm nhân viên mới và quay lại giao diện quản lý chuỗi.</p> | <p>1.1 Hệ thống hiển thị một form nhập thông tin sản phẩm mới.</p> <p>2.1 Hệ thống kiểm tra tính hợp lệ của thông tin sản phẩm mới và thông báo kết quả cho người nhập.</p> <p>3.1 Nếu thông tin hợp lệ, hệ thống lưu thông tin sản phẩm mới vào cơ sở dữ liệu, thông báo cho người quản trị hệ thống.</p> |
| Exception | 3.2.a Thông tin không hợp lệ . Hệ thống yêu cầu nhập lại thông tin | |

Bảng 3.2.2.13: Đặc tả Use case Thêm sản phẩm

3.2.3.14 Use case Sửa thông tin sản phẩm

| | | |
|---------------------------|---|---------------|
| Use Case ID | UC14 | |
| Use Case | Sửa thông tin sản phẩm | |
| Scenario | Người quản trị muốn chỉnh sửa sửa thông tin sản phẩm vào hệ thống quản lý. | |
| Triggering Event | Admin cần sửa thông tin sản phẩm trong hệ thống. | |
| Description | Chức năng sửa thông tin sản phẩm cho phép người quản trị hệ thống cập nhật thông tin sản phẩm vào hệ thống quản lý. | |
| Actor(s) | Admin | |
| Pre-Condition(s): | Admin đã đăng nhập vào hệ thống quản lý cửa hàng. Sản phẩm đã tồn tại trong hệ thống. | |
| Post-Condition(s): | Sản phẩm được cập nhật vào hệ thống. Thông tin của sản phẩm được lưu trữ trong cơ sở dữ liệu của hệ thống. | |
| Flow of Event | Actor | System |

| | | |
|------------------|--|--|
| | <p>1. Admin chọn tính năng sửa sản phẩm trong giao diện quản lý.</p> <p>2. Admin nhập thông tin sản phẩm mới vào các trường thông tin cần thiết.</p> <p>3. Admin nhấn nút "Lưu" để lưu thông tin của sản phẩm.</p> <p>4. Admin hoàn tất cập nhật thông tin sản phẩm và quay lại giao diện quản lý.</p> | <p>1.1 Hệ thống hiển thị một form nhập thông tin sản phẩm.</p> <p>2.1 Hệ thống kiểm tra tính hợp lệ của thông tin sản phẩm và thông báo kết quả cho người nhập.</p> <p>3.1 Nếu thông tin hợp lệ, hệ thống lưu thông tin sản phẩm vào cơ sở dữ liệu, thông báo cho người quản trị hệ thống.</p> |
| Exception | 3.2.a Thông tin không hợp lệ . Hệ thống yêu cầu nhập lại thông tin | |

Bảng 3.2.2.14: Đặc tả Use case Thêm sản phẩm

3.2.3.15 Use case Xóa sản phẩm

| | | |
|---------------------------|--|---------------|
| Use Case ID | UC15 | |
| Use Case | Xóa sản phẩm | |
| Scenario | Admin muốn xóa một sản phẩm trong hệ thống quản lý cửa hàng. | |
| Triggering Event | Admin cần xóa sản phẩm. Do sản phẩm không còn được kinh doanh. Nhấn vào chức năng xóa sản phẩm. | |
| Description | Admin cho phép xóa sản phẩm đã không còn kinh doanh tại cửa hàng. Khi đó toàn bộ thông tin của sản phẩm sẽ bị xóa khỏi hệ thống. | |
| Actor(s) | Admin | |
| Pre-Condition(s): | Admin đã đăng nhập vào hệ thống quản lý cửa hàng. Thông tin sản phẩm đã được lưu trữ trong hệ thống quản lý cửa hàng. | |
| Post-Condition(s): | Sản phẩm được xóa khỏi hệ thống quản lý cửa hàng. Thông tin liên quan của sản phẩm đã bị xóa khỏi hệ thống. | |
| Flow of Event | Actor | System |

| | | |
|------------------|--|---|
| | <p>1. Admin mở giao diện chứa chức năng xóa thông tin sản phẩm trong hệ thống quản lý cửa hàng.</p> <p>2 . Admin chọn sản phẩm cần xóa thông tin.</p> <p>3. Admin xác nhận xóa sản phẩm.</p> | <p>1.1 Hệ thống hiển thị giao diện xóa thông tin sản phẩm.</p> <p>2. Hệ thống thông báo xác nhận xóa sản phẩm.</p> <p>3. Hệ thống xóa thông tin sản phẩm khỏi hệ thống.</p> |
| Exception | <p>3.a Admin hoặc Quản lý cửa hàng không xác nhận xóa sản phẩm.</p> <p>4.1.a Hệ thống không xóa sản phẩm và hệ thống quay lại giao diện hiển thị sản phẩm.</p> | |

Bảng 3.2.2.15: Đặc tả Use case Xóa sản phẩm

3.2.3.16 Use case Thanh toán tại quầy

| | |
|---------------------------|--|
| Use Case ID | UC16 |
| Use Case | Thanh toán |
| Scenario | Người dùng muốn thanh toán các sản phẩm đã mua. |
| Triggering Event | Khách hàng muốn thanh toán tiền tại quầy. |
| Description | Thanh toán tiền mà người dùng mua sản phẩm. Người dùng mua trực tiếp tại thanh toán qua nhân viên bán hàng. |
| Actor(s) | Quản lý cửa hàng Nhân viên bán hàng |
| Pre-Condition(s): | Nhân viên bán hàng đã đăng nhập vào hệ thống. Nhân viên có thể nhập được mã sản phẩm và số lượng sản phẩm. |
| Post-Condition(s): | Khách hàng được thanh toán thành công. Số lượng sản phẩm trên quầy bị giảm theo số sản phẩm được mua. Thông tin hóa đơn được lưu vào hệ thống. |

| Flow of Event | Actor | System |
|---------------|--|--|
| | <p>1. Người dùng truy cập vào hệ thống thanh toán.</p> <p>2. Người dùng nhập mã sản phẩm bằng máy quét QR hoặc nhập bằng tay và nhập vào số lượng sản phẩm.</p> <p>3. Người dùng nhấn vào thanh toán</p> <p>4. Người dùng nhập thông tin khách hàng.</p> <p>5. Yêu cầu khách hàng thanh toán. Người dùng nhập số tiền khách hàng đưa</p> <p>6. Người dùng nhấn vào In hóa đơn.</p> | <p>1.1 Hiện thị giao diện thanh toán.</p> <p>2.1 Tính toán và hiển thị số tiền cần thanh toán.</p> <p>3.1 Hệ thống yêu cầu nhập thông tin khách hàng.</p> <p>4.1 Hệ thống kiểm tra số điện thoại.</p> <p>4.1.a Nếu số điện thoại đã tồn tại thì hiển thị thông tin khách hàng và yêu cầu nhập số tiền khách hàng đưa.</p> <p>4.1.b Nếu số điện thoại không tồn tại thì yêu cầu người dùng nhập tên và địa chỉ để tạo khách hàng mới rồi hiển thị lại thông tin khách hàng.</p> <p>5.1 Hệ thống kiểm tra số tiền.</p> <p>5.1.a Số tiền không đủ thì yêu cầu nhập lại.</p> |

| | | |
|------------------|--|---|
| | | <p>5.1.b Số tiền đủ thì cho phép in hóa đơn.</p> <p>6.1 Hệ thống hiển thị thanh toán thành công và tự động trừ số lượng hàng hóa trên quầy, số tiền từ hóa đơn được cập nhật lên hệ thống cộng vào tổng doanh thu. Thông tin hóa đơn được lưu lại . Hệ thống xuất hóa đơn cho khách hàng.</p> |
| Exception | <p>1.1 a Người dùng đăng nhập không thành công.</p> <p>4.1 a Hệ thống lỗi không xuất được hóa đơn.</p> | |

Bảng 3.2.2.16: Đặc tả Use case Thanh toán tại quầy

3.2.3.17 Use case Thống kê

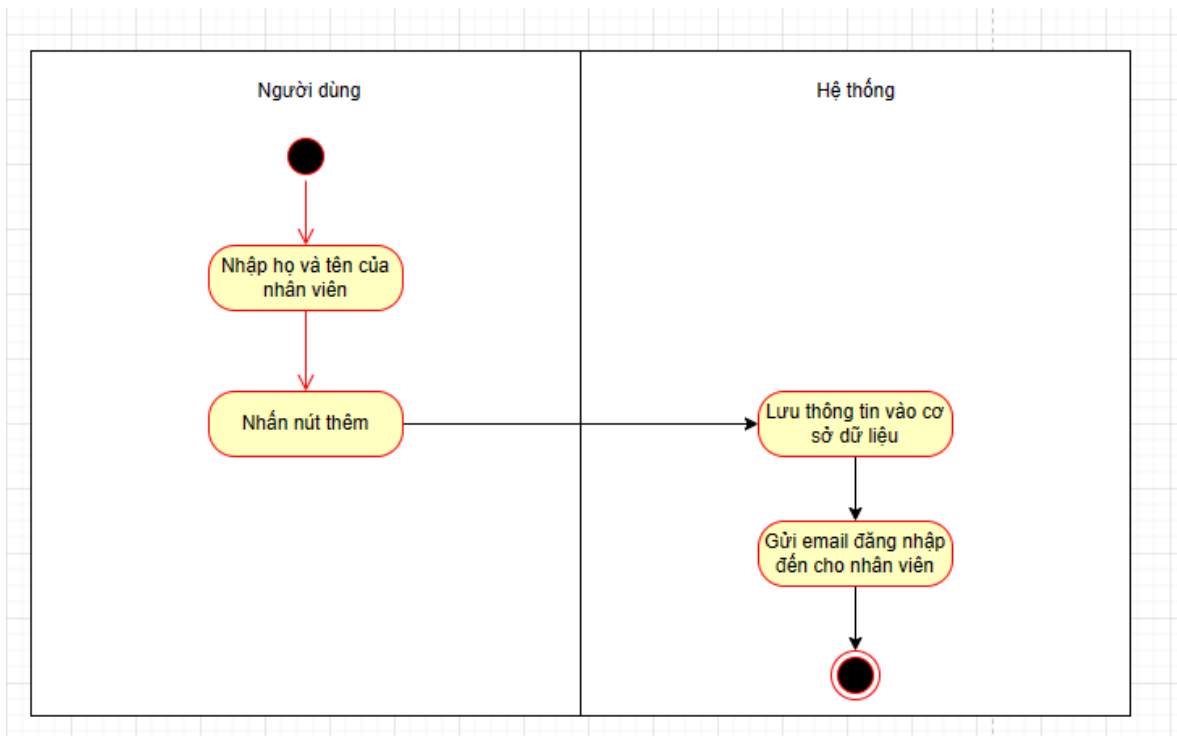
| | | |
|---------------------------|---|---------------|
| Use Case ID | UC17 | |
| Use Case | Thống kê | |
| Scenario | Admin muốn thống kê của cửa hàng. | |
| Triggering Event | Admin tiến hành thống kê. | |
| Description | Thống kê doanh thu, đơn hàng, sản phẩm được bán theo các mốc thời gian hoặc khoảng thời gian. | |
| Actor(s) | Admin | |
| Pre-Condition(s): | Admin đã đăng nhập vào hệ thống. | |
| Post-Condition(s): | Thông tin doanh thu, đơn hàng, người dùng mua. | |
| Flow of Event | Actor | System |

| | | |
|------------------|--|--|
| | <p>1. Admin truy cập vào hệ thống quản lý.</p> <p>2. Admin nhấn chọn thống kê trên thanh điều hướng.</p> <p>3. Admin lựa chọn thời gian thống kê và nhấn nút thống kê.</p> | <p>1.1 Hiện thị giao diện quản lý.</p> <p>2.1 Hệ thống hiển thị giao diện thống kê kèm các tùy chọn ngày thống kê.</p> <p>3.1 Hệ thống hiển thị thống kê doanh thu, số đơn hàng đã bán kèm thông tin của đơn hàng và khách hàng đã mua theo khoảng thời gian được Admin muốn thống kê.</p> |
| Exception | <p>1.1 a Admin đăng nhập không thành công.</p> <p>4.1 a Hệ thống lỗi không thể lấy dữ liệu.</p> | |

Bảng 3.2.2.17: Đặc tả Use case Thống kê

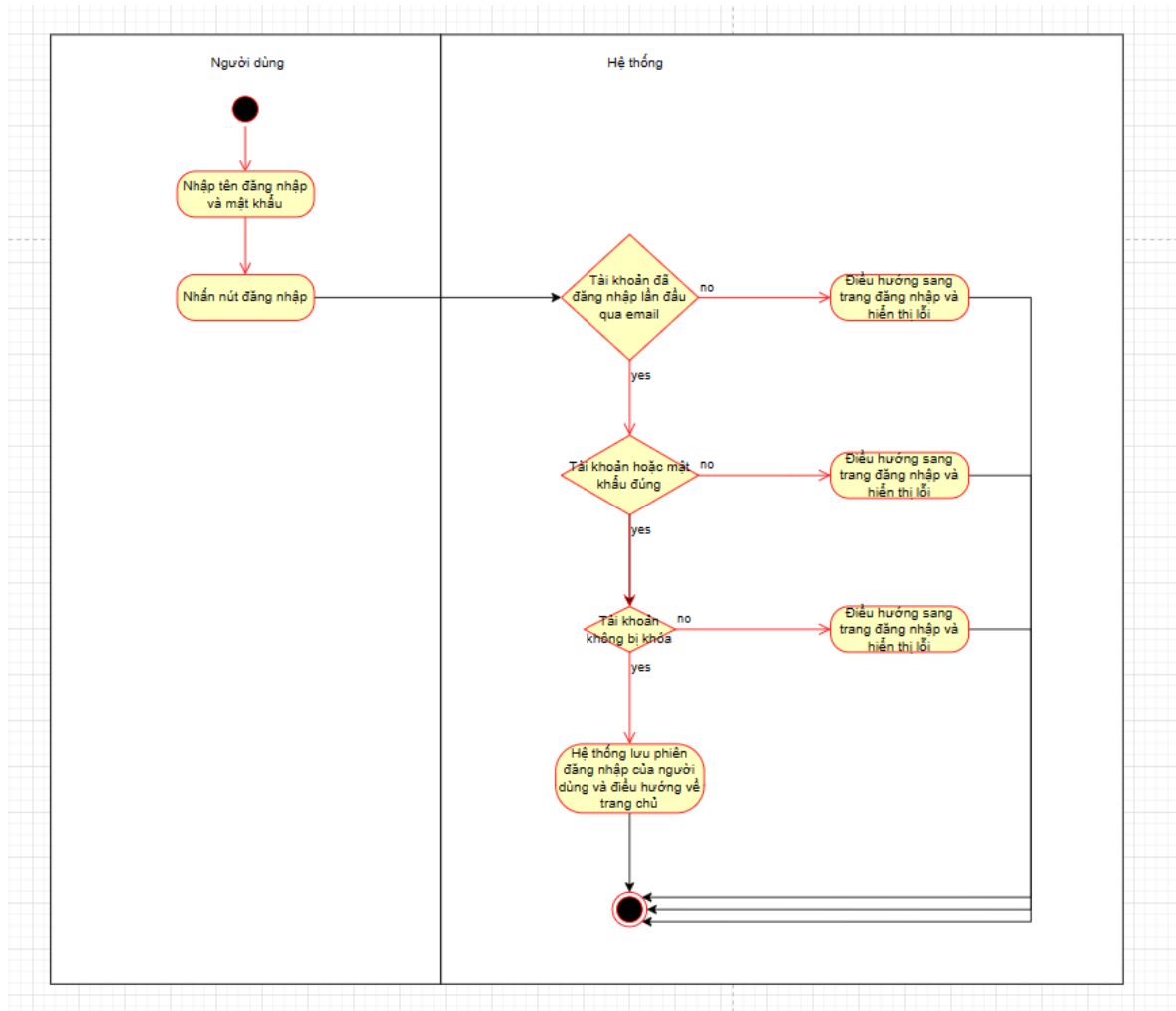
3.2.4 Sơ đồ Activity

3.2.4.1 Tạo tài khoản cho nhân viên bán hàng



Hình 3.2.4.1: Sơ đồ Activity Tạo tài khoản cho nhân viên bán hàng

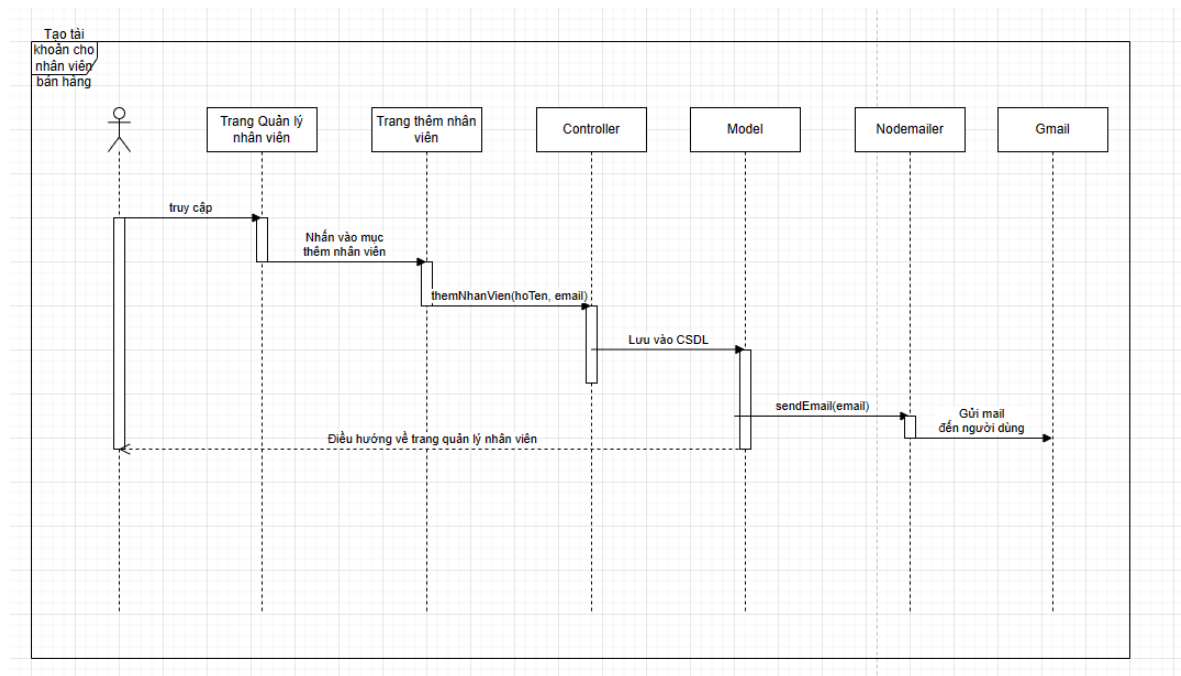
3.2.4.2 Đăng nhập



Hình 3.2.4.2: Sơ đồ Activity Đăng nhập

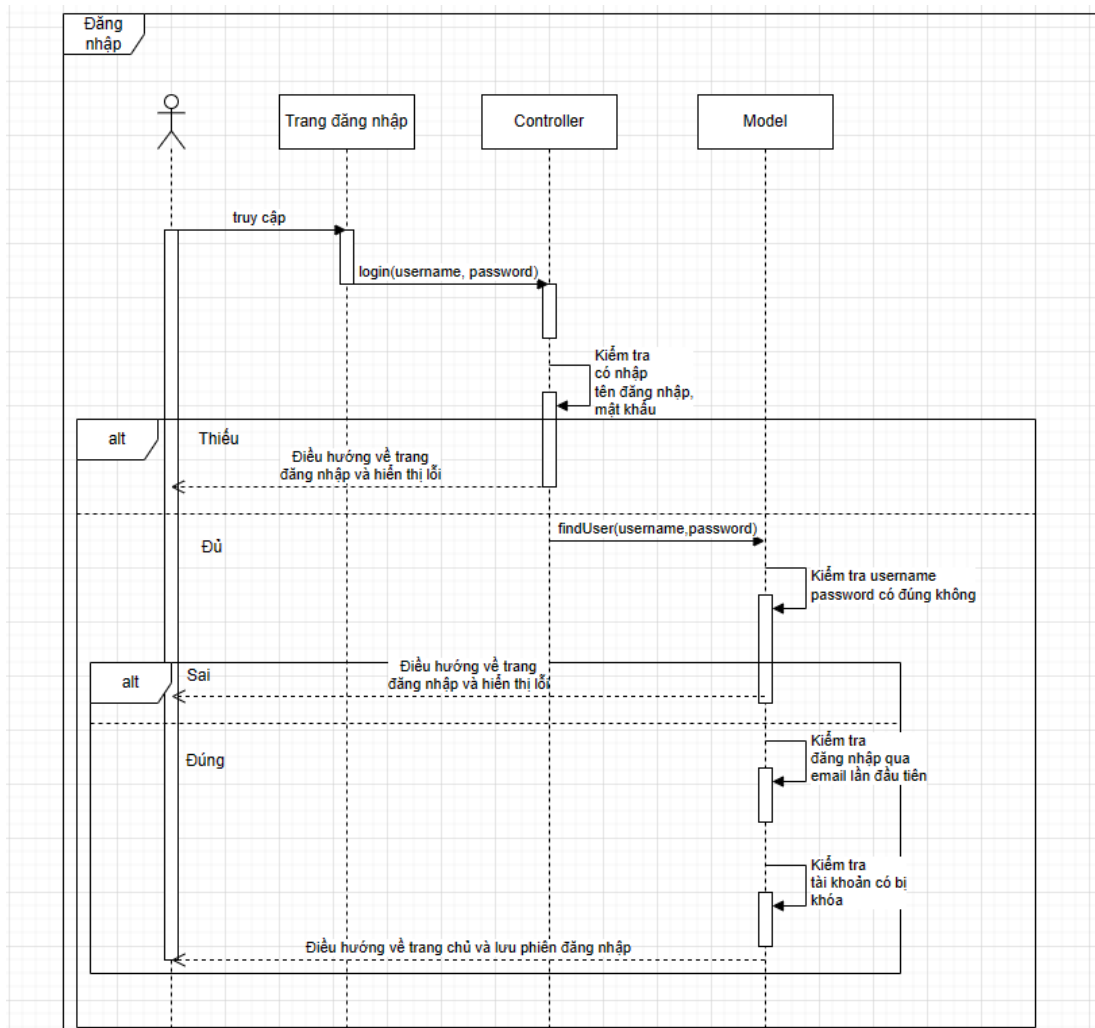
3.2.5 Sơ đồ Sequence

3.2.5.1 Tạo tài khoản cho nhân viên bán hàng



Hình 3.2.5.1: Sơ đồ Sequence Tạo tài khoản cho nhân viên bán hàng

3.2.5.2 Đăng nhập



Hình 3.2.5.2: Sơ đồ Sequence Đăng nhập

CHƯƠNG 4. XÂY DỰNG CHỨC NĂNG HỆ THỐNG

4.1 Tạo tài khoản và gửi email người dùng

```
exports.createAccount = async (req, res) => {  
  const { fullName, email } = req.body  
  const newUser = await User.create({  
    fullName,  
    email,  
  })  
  const token = newUser.generateConfirmEmailToken()  
  newUser.save()  
  sendEmail(  
    email,  
    'Your account has just been created',  
    `Click this link to confirm  
    ${req.protocol}://localhost:${process.env.PORT}/accounts/${token}`  
  )  
  res.redirect('/dashboard/users')  
}
```

Khi admin truy cập vào form tạo tài khoản cho salesperson và nhập thông tin cần thiết là họ tên và tài khoản email thì server sẽ lưu xuống cơ sở dữ liệu thông tin của user này sau đó server sẽ tạo token để xác thực email và gửi về cho salesperson.

Hàm generateConfirmEmailToken được triển khai trong userSchema

```

userSchema.methods.generateConfirmEmailToken = function () {
  const token = crypto.randomBytes(32).toString('hex')
  this.confirmEmailToken = crypto
    .createHash('sha256')
    .update(token)
    .digest('hex')
  this.confirmEmailTokenExpires = Date.now() + 60 * 1000
  return token
}

```

Sử dụng thư viện crypto để tạo ra token đồng thời thiết lập thời gian hiệu lực của token là 1 phút.

4.2 Mở/ Khóa tài khoản

```

exports.changeUserStatus = async (req, res) => {
  const id = req.body.id
  const userFound = await User.findOne({ _id: id })

  userFound.changeStatus()
  userFound.save()

  res.status(200).json({
    status: 'success',
  })
}

```


Tìm user theo id sau đó gọi hàm changeStatus

```
userSchema.methods.changeStatus = function() {
    this.status = this.status == "inactive" ? "active" :
    "inactive"
}
```

Hàm này sẽ thay đổi status của document thành “active” nếu đang là “inactive” và ngược lại.

4.3 Tạo hóa đơn sản phẩm và thanh toán

Người dùng nhập các sản phẩm nhập các thông tin sản phẩm khách hàng mua để hệ thống tính toán. Khi nhấn thanh toán thì sẽ chuyển giao diện để nhập số điện thoại khách hàng và thông tin sản phẩm vừa mới nhập.

```
exports.getCheckNp= async (req,res)=>{
    const billData = req.query.bill;
    const bill = JSON.parse(billData);
    for (let i = 0; i < bill.items.length; i++) {
        const itemId = bill.items[i].id;
        const foundProduct = await products.findOne({ _id:new mongoose.Types.ObjectId(itemId)});
        if (foundProduct) {
            bill.items[i].imgSrc = foundProduct.imgSrc;
        }
    }
    res.render('pay/checkNumberPhone', { bill: bill });
}
```

Người dùng nhập số điện thoại khách hàng vào được gửi về sever kiểm tra. Nếu số điện thoại đã tồn tại thì hiển thị thông tin khách hàng và yêu cầu nhập tiền khách đưa. Nếu không tồn tại thì hiện form để nhập thông tin khách hàng và lưu khách hàng lại.

```

exports.checkPhoneNumber = (req, res, next) => {
  phoneNumber = req.body.phone;
  customers.findOne({ phone: phoneNumber })
    .then(customer => {
      if (customer) {
        res.send(customer.toObject());
      } else {
        res.send({message: 'New customer'});
      }
    })
    .catch(err => {
      console.log(err);
    });
};

```

Người dùng sẽ nhập số tiền khách đưa và hệ thống sẽ kiểm tra . Số tiền khách đưa phải lớn hơn tổng tiền sản phẩm.

```

exports.checkMoney=(req,res)=>{
  let moneyOfCustomer= req.body.moneyOfCustomer;
  let total_money=req.body.total_money;
  if(moneyOfCustomer<total_money){
    res.send({error:'Số tiền không đủ hãy nhập lại'})
  }else{
    let excess_money=moneyOfCustomer-total_money;
    res.send({excess_money:excess_money})
  }
}

```

Người dùng nhấn in hóa đơn thì hóa đơn được lưu vào database và đồng thời hệ thống sẽ xuất hóa đơn cho khách hàng ở dạng pdf.

```
exports.saveOrder= async (req,res)=>{
  const databill=req.body.detailBill;
  const phone= databill.phone;
  const name_customer= databill.name_customer;
  const totalAmount= databill.totalAmount
  const amountPaid=databill.amountPaid
  const refundAmount=databill.refundAmount
  const numberOfproduct=databill.numberOfproduct
  const products=databill.products
  const newOrder = new order({
    phone,
    name_customer,
    totalAmount,
    amountPaid,
    refundAmount,
    numberOfproduct,
    products,
  });
};
```

```

    await newOrder.save()
    .then(savedOrder => {
      printBill(savedOrder)
      res.send({success:"Đã lưu hóa đơn thành công"})
    })
    .catch(error => {
      res.status(500).json({ error: 'Internal Server Error' });
    });
  }
}

```

In hóa đơn được sẽ sử dụng thư viện easyinvoice để tạo ra hóa đơn dạng pdf. Đầu tiên tạo một data chứa các thông tin cần thiết trong hóa đơn.

```

var data = {
  "customize": {
  },
  "taxNotation": '',
  "images": {
    "logo": "https://img.upanh.tv/2023/12/10/Web_Photo_Editor.jpg",
    "background": "https://img.upanh.tv/2023/12/10/Invoice.png"
  },
  "sender": {
    "company": "IPoneX",
    "address": "19, Nguyễn Hữu Thọ",
    "zip": "Quận 7",
    "city": "HCMC"
  },
  "client": {
    "company": `${detailBill.name_customer}`,
    "address": `0${detailBill.phone}`,
  },
  "information": {
    "date": vietnamTime
  },

  "products": products,
  "bottom-notice": "Cảm ơn đã mua hàng",
  "settings": {
    "currency": "VND",
    "format": "A4",
  },
  // Translate your invoice to your preferred language
  "translate": {

  },
};
return data;

```

File pdf sẽ được tải về sau khi lưu thành công hóa đơn

```
easyinvoice.createInvoice(data.pdf, function (result) {
  easyinvoice.download('Invoice.pdf', result.pdf);
});
```

4.4 Thống kê

Thống kê doanh thu của cửa hàng theo các mốc thời gian như 7 ngày, 1 tháng hoặc khoản thời gian do người dùng chọn.

Admin sẽ gửi một request đến server bao gồm khoản thời gian thống kê

Truy vấn dữ liệu thống kê doanh thu:

```
// Doanh thu
const totalAmountDate = await Order.aggregate([
  {
    $match: {
      purchaseDate: { $gte: dateStart, $lt: currentDate }
    },
  },
  {
    $group: {
      _id: { $dayOfWeek: "$purchaseDate" },
      totalAmount: { $sum: "$totalAmount" }
    },
  },
  {
    $sort: { _id: 1 }
  }
]);
```

Truy vấn dữ liệu thống kê sản phẩm được bán:

```
const idProducts = []
const productsByOrder = await
Order.find({}).populate('products')
for (let i = 0; i < productsByOrder.length; i++) {
```

```

    for (let j = 0; j < productsByOrder[i].products.length;
j++) {
        idProducts.push(productsByOrder[i].products[j].id)
    }
}

```

```

const countProducts = await Product.aggregate([
  $match: {
    _id: { $in: idProducts.map(
      id => new mongoose.Types.ObjectId(id))
    }},
  $group: {
    _id: '$cate',
    count: { $sum: 1 }
  }
])

```

4.5 Kiểm tra quyền

```

exports.checkAuthorization = (...roles) => {
  return async (req, res, next) => {
    const userId = req.session.userId
    const userFound = await User.findOne({ _id: userId })
    if (roles.includes(userFound.role)) {
      next()
    } else {
      return next(
        new Error(
          'You do not have any privilege to access
this resource'

```

```
)  
  })  
}}
```

Ví dụ:

```
router.route('/users').get(checkAuthorization('admin'),  
  getAllUserExceptAdmin)
```

Chỉ có admin mới có thể lấy được danh sách các salesperson trong hệ thống. Nếu người dùng không có quyền truy cập vào tài nguyên này thì hệ thống sẽ điều hướng đến trang thông báo lỗi 500 với lỗi 'You do not have any privilege to access this resource'

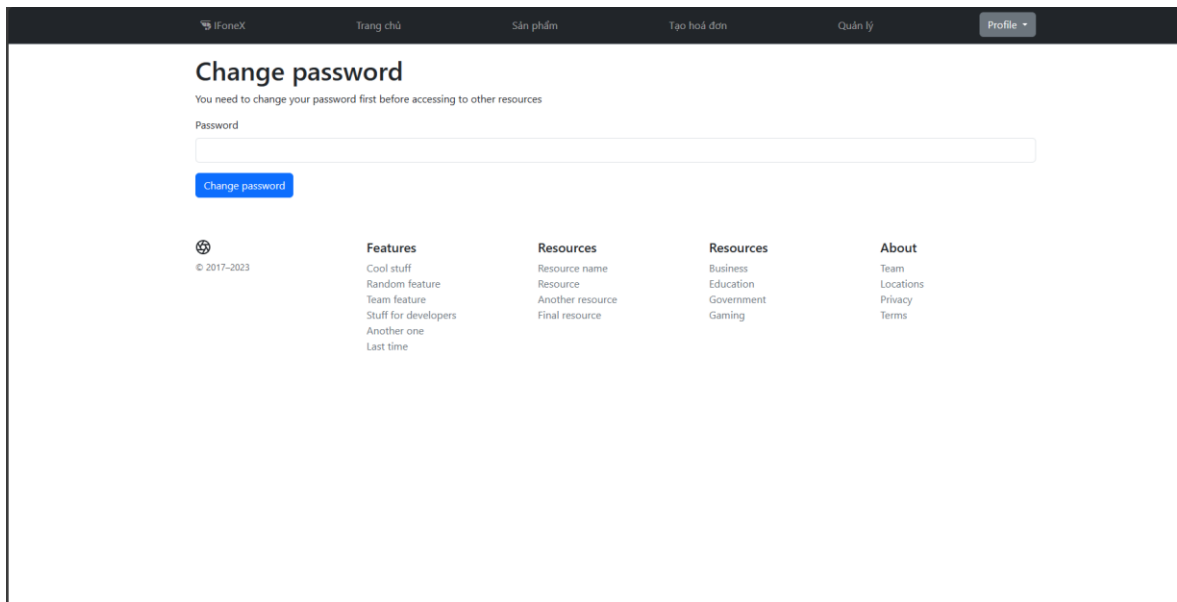
CHƯƠNG 5. GIAO DIỆN CHỨC NĂNG HỆ THỐNG

Code giao diện được tham khảo từ <https://startbootstrap.com/theme/sb-admin-2> và tùy chỉnh lại để phù hợp với dự án.

5.1 Chức năng tổng quát

5.1.1 Thay đổi mật khẩu

Đây là giao diện thay đổi mật khẩu cho các tài khoản mới đăng nhập lần đầu (mới được tạo). Người dùng chỉ cần nhập mật khẩu mới.



Hình 5.1.1.1: Giao diện thay đổi mật khẩu cho tài khoản mới đăng nhập lần đầu

Đây là giao diện thay đổi mật khẩu. Người dùng phải nhập mật khẩu cũ, mật khẩu mới và mật khẩu xác nhận để có thể đổi mật khẩu.

Hình 5.1.1.2: Giao diện thay đổi mật khẩu cho tài khoản

5.1.2 Xem thông tin hồ sơ cá nhân và cập nhật ảnh đại diện

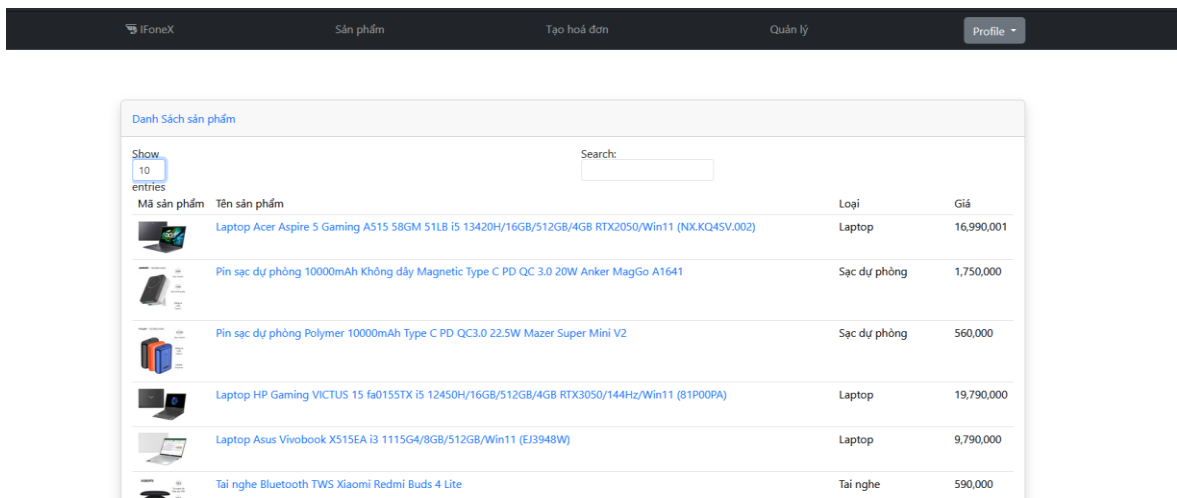
Hình 5.1.2.1: Giao diện thay Xem thông tin hồ sơ cá nhân và cập nhật ảnh đại diện

Tại trang này, người dùng có thể xem thông tin cá nhân của mình như avatar, họ tên và quyền của mình.

Người dùng có thể chọn ảnh khác để cập nhật lại avatar của mình. Sau khi chọn xong người dùng nhấn nút “Submit” thì ảnh sẽ được lưu trên server và thay đổi trên giao diện

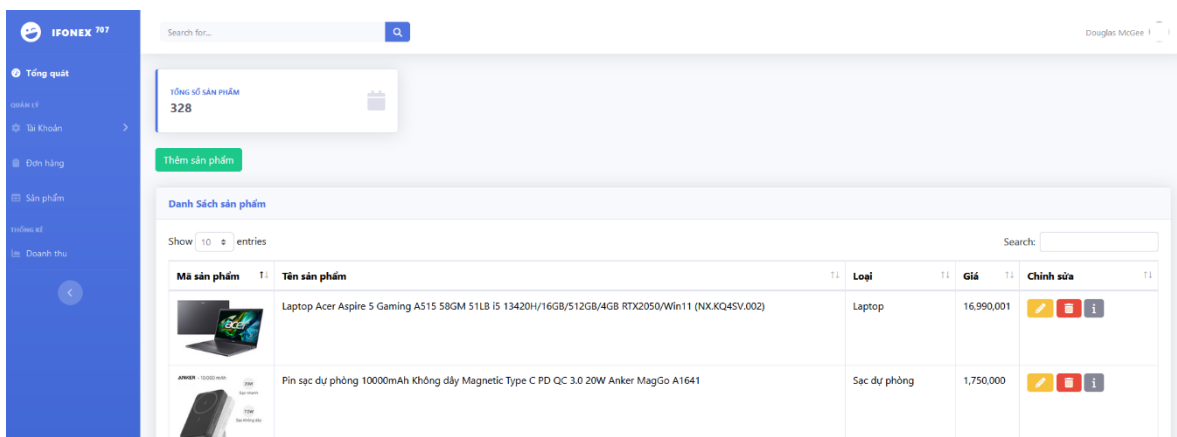
5.1.3 Xem danh sách sản phẩm

Admin hoặc SalesPerson có thể xem danh sách các sản phẩm có trong hệ thống.



Hình 5.1.3.1: Giao diện Xem danh sách sản phẩm của SalesPerson

Admin có một giao diện hiển thị danh sách sản phẩm kèm theo các thao tác chỉnh sửa, cập nhật sản phẩm.



Hình 5.1.3.2: Giao diện Xem danh sách sản phẩm của Admin

5.1.4 Xem lịch sử mua hàng của khách hàng

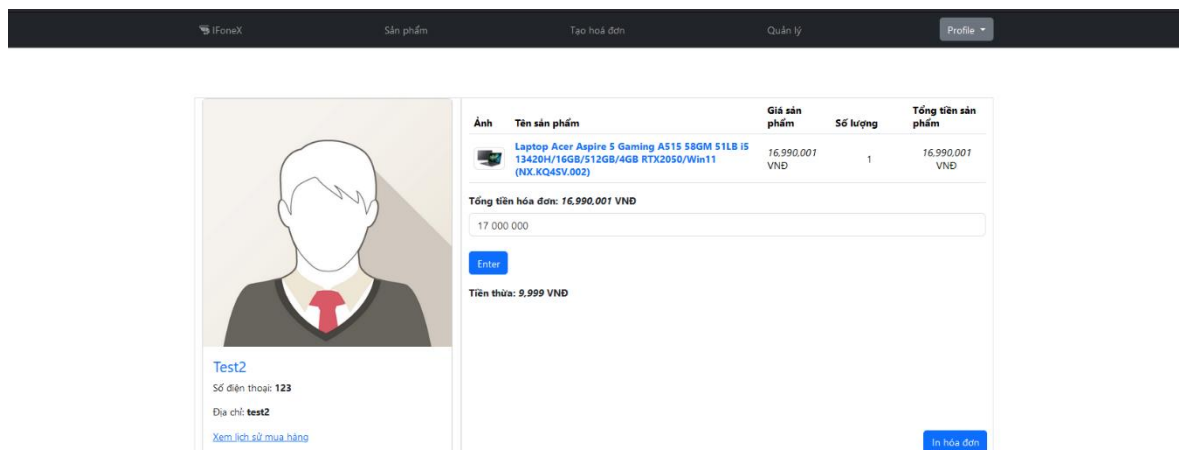
Giao diện hiển thị thông tin khách hàng kèm theo lịch sử mua hàng của khách, người dùng có thể nhấn chọn “Xem chi tiết” để xem thông tin chi tiết của hoá đơn.



Hình 5.1.4.1: Giao diện Xem lịch sử mua hàng của khách hàng

5.1.5 In hoá đơn

Nhân viên có thể in hoá đơn trong khi tạo đơn hàng thông qua nút “In hoá đơn”



Hình 5.1.5.1: Giao diện In hoá đơn

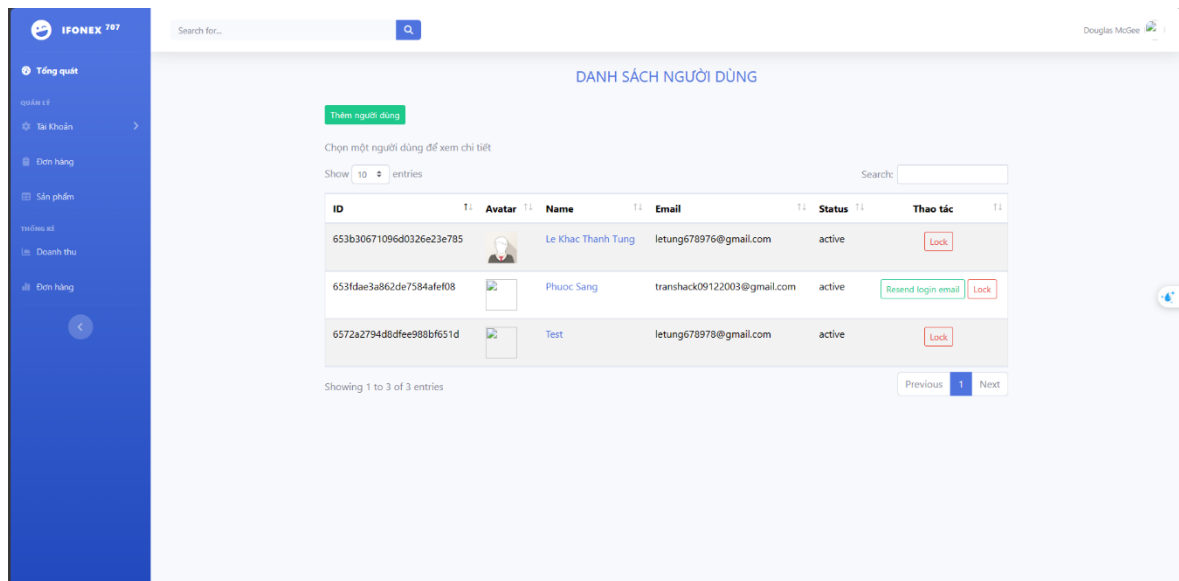
Admin có thể tiến hành in lại hoá đơn của khách trong quản lý danh sách các đơn hàng bằng thao tác nhấn nút “In hoá đơn” trong giao diện xem chi tiết đơn hàng.



Hình 5.1.5.2: Giao diện In lại hoá đơn ở trang thông tin chi tiết đơn

5.2 Chức năng Admin

5.2.1 Quản lý danh sách nhân viên



Hình 5.2.1.1: Giao diện Quản lý danh sách nhân viên

Ở trang này admin có thể thêm nhân viên, xem danh sách nhân viên có trong hệ thống cũng như xem thông tin chi tiết của một nhân viên bằng cách nhấp vào tên.

Ngoài ra admin có thể gửi lại email đăng nhập cho người dùng chưa đăng nhập lần đầu bằng email cũng như có thể mở khóa/ khóa tài khoản.

5.2.2 Xem danh sách khách hàng

IFONEX 707

Search for...

DANH SÁCH KHÁCH HÀNG

Chọn một khách hàng để xem chi tiết

Show 10 entries

| ID | Name | Phone | Address | Thao tác |
|--------------------------|-------------------|-------------|--|---|
| 653a47ed9e9c6fc57a697934 | Nguyen Thanh Tu | 0326282965 | 1135 Huỳnh Tấn Phát | Update Delete |
| 653b818aeea37a77d8724466 | tutthanh | 0907014202 | 320 cầu trâm | Update Delete |
| 6555ec58ee511f4d34e1c675 | Anh Sang Đà | 0964038277 | 7 núi An Giang | Update Delete |
| 6555ecbada7593860f448e56 | Khải 7 Sang | 113 | Núi Sam | Update Delete |
| 65683d243edafa9df9399bc9 | Anh Khôi Chà Đình | 0373737981 | Vườn dừa sập Trà Vinh | Update Delete |
| 656ec396b29a84f2d83ba359 | Nguyễn Thu Trang | 03391234567 | Trần Xuân Soạn | Update Delete |
| 6571705edc681758384e87eb | admin | 0347763291 | admin | Update Delete |
| 6572b04e5e706713bd25d99d | Anh Be Thanh Huy | 0123456789 | 1135/62 Huỳnh Tấn Phát ,Phủ Thuận ,Quận 7 ,TpHCM | Update Delete |

Showing 1 to 8 of 8 entries

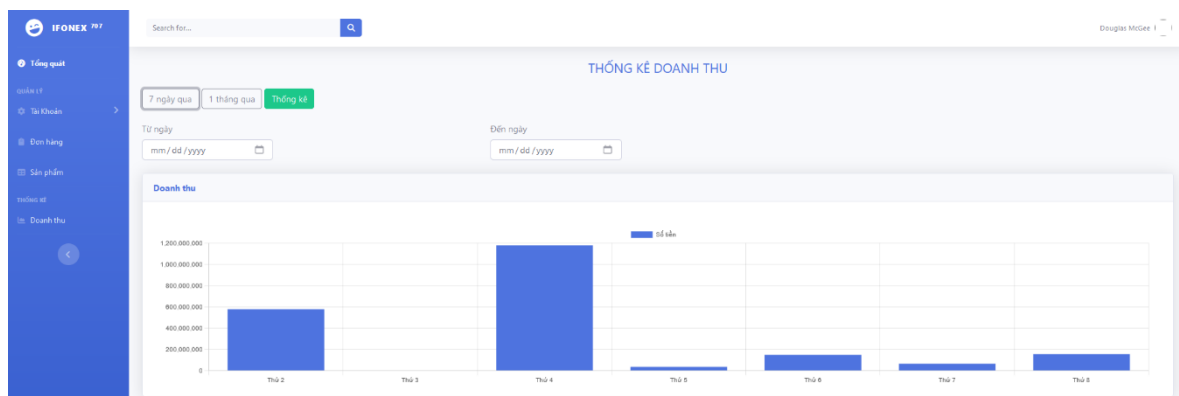
Previous 1 Next

Hình 5.2.2.1: Giao diện Xem danh sách khách hàng

Ở trang này admin có thể xem danh sách khách hàng có trong hệ thống cũng như xem thông tin chi tiết của một khách hàng bằng cách nhấp vào tên. Ngoài ra admin còn có thể cập nhật thông tin của khách hàng, xóa khách hàng khỏi cơ sở dữ liệu

5.2.3 Thống kê

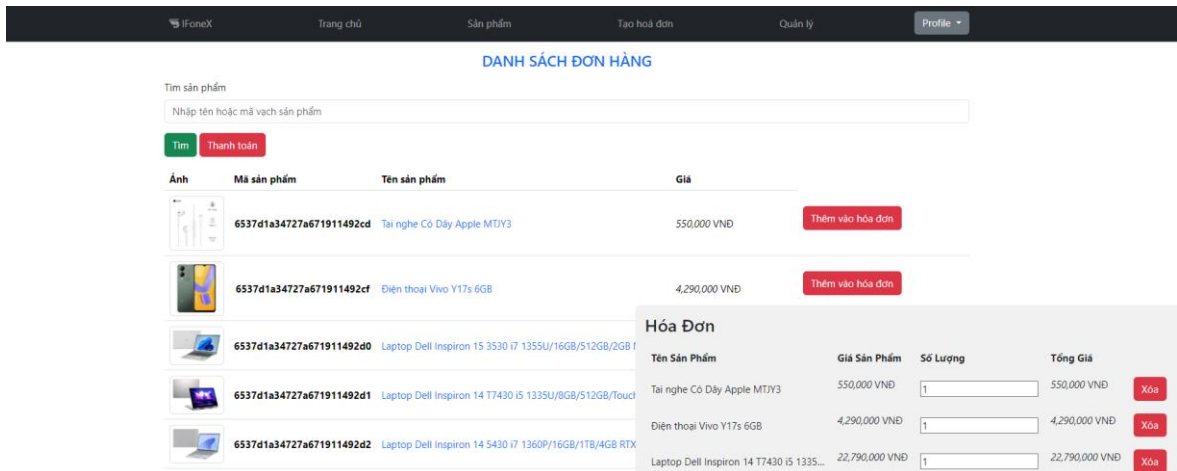
Khi admin truy cập vào trang thống kê, hệ thống sẽ hiển thị các thao tác để chọn khoảng thời gian thống kê.



Hình 5.2.3.1: Giao diện Thống kê

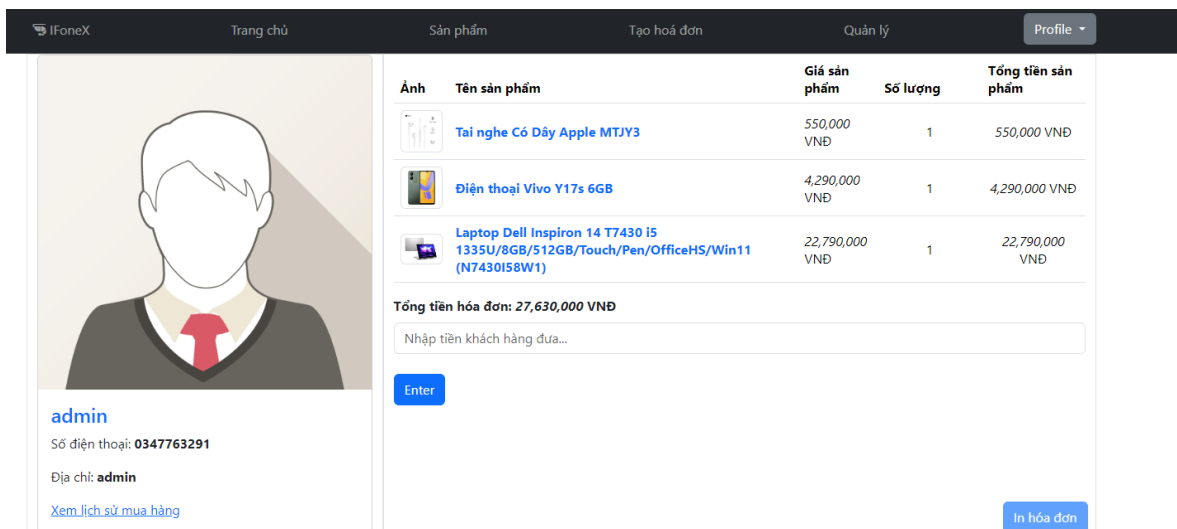
5.3 Chức năng salesperson

5.3.1 Tạo đơn hàng, tìm khách hàng và thêm khách hàng mới



Hình 5.3.1.1: Giao diện Tạo đơn hàng

Trong quá trình tạo đơn hàng, nhân viên sẽ yêu cầu khách hàng cung cấp thông tin như số điện thoại. Hệ thống sẽ tiến hành tìm kiếm khách hàng trong cơ sở dữ liệu thông qua số điện thoại của khách. Nếu là khách hàng lần đầu thì hệ thống sẽ thêm dữ liệu của khách hàng mới vào hệ thống.



Hình 5.3.1.2: Giao diện tra cứu thông tin khách hàng qua số điện thoại

CHƯƠNG 6. TỔNG KẾT

6.1 Kết quả đạt được

Trong quá trình hoàn thành đồ án này, nhóm thực hiện tất cả các giai đoạn để xây dựng một hệ thống hoàn chỉnh, mỗi giai đoạn bao gồm nhiều công việc khác nhau. Qua bài báo cáo môn học này, nhóm em đã được trao đổi cũng như tiếp thu thêm nhiều kiến thức quý báu, được thực hành những điều đã học về môn “Phát triển ứng dụng web với NodeJS”.

Qua đồ án báo cáo môn học nhóm em đã được củng cố các kiến thức lý thuyết về môn học “Phát triển ứng dụng web với NodeJS”, các kiến thức về phân tích và thiết kế yêu cầu, kiến thức và kỹ năng xây dựng cơ sở dữ liệu và xây dựng website để thanh toán và quản lý quá trình kinh doanh của một điểm bán hàng dựa trên HTML, CSS, JavaScript, NodeJS,... Nhóm em nghĩ rằng điều này sẽ giúp ích rất nhiều cho công việc sau này của chúng em.

6.2 Những mặt hạn chế

- Tổ chức source code chưa được tốt
- Chưa validate kỹ các dữ liệu do người dùng gửi lên

6.3 Hướng phát triển trong tương lai

- Validate dữ liệu chặt chẽ hơn
- Nhấn tin với admin
- Deploy trang web lên server để có thể sử dụng thực tế

TÀI LIỆU THAM KHẢO

Tiếng Việt:

Tiếng Anh: