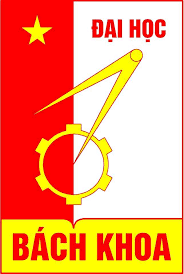
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**Phát triển phần mềm phân tán**

**Đề tài: Hệ thống cửa hàng bán sản phẩm (ReactJS, NodeJS, MongoDB, Docker)**

Họ và tên sinh viên:

* Nguyễn Bá Tùng Lâm - 20176035
* Phạm Trung Hiếu - 20175712
* Trần Văn Thành - 20176057

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | TS. Nguyễn Nhất Hải |
| **Bộ môn:** | Phát triển phần mềm phân tán |
| **Viện:** | Viện Công nghệ thông tin và truyền thông |
| **HÀ NỘI, 01/2022** | |

**MỤC LỤC**

[CHƯƠNG 1. Đặt vấn đề 2](#_Toc92063904)

[1.1 Phân tích yêu cầu đề bài 2](#_Toc92063905)

[1.1.1 Tên đề tài 2](#_Toc92063906)

[1.1.2 Mục đích 2](#_Toc92063907)

[1.2 Chức năng 2](#_Toc92063908)

[1.3 Quy mô, phạm vi 3](#_Toc92063909)

[1.3.1 Mặt hàng nhắm tới: 3](#_Toc92063910)

[1.3.2 Phạm vi phục vụ 3](#_Toc92063911)

[CHƯƠNG 2. Lý thuyết liên quan 4](#_Toc92063912)

[2.1 Cách viết HTML, CSS 4](#_Toc92063913)

[2.1.1 HTML 4](#_Toc92063914)

[2.1.2 CSS 4](#_Toc92063915)

[2.2 Ngôn ngữ JavaScript 4](#_Toc92063916)

[2.3 Các framework, library liên quan hỗ trợ cho việc thiết kế Front-end và Back-end 4](#_Toc92063917)

[2.3.1 Front-end 5](#_Toc92063918)

[2.3.2 Back-end 5](#_Toc92063919)

[2.4 Thiết kế cơ sở dữ liệu NoSQL 5](#_Toc92063920)

[2.5 Cài đặt docker 5](#_Toc92063921)

[2.5.1 Cơ bản cấu trúc docker 5](#_Toc92063922)

[2.5.2 Cấu hình Dockerfile 6](#_Toc92063923)

[2.5.3 Cấu hình Docker Compose file 7](#_Toc92063924)

[CHƯƠNG 3. Phân tích và thiết kế hệ thống 8](#_Toc92063925)

[3.1 Phân tích use case 8](#_Toc92063926)

[3.1.1 Use case tổng quan 8](#_Toc92063927)

[3.1.2 Phân rã use case 9](#_Toc92063928)

[3.2 Đặc tả use case 11](#_Toc92063929)

[3.2.1 Use case “Đăng nhập” 11](#_Toc92063930)

[3.2.2 Use case “Đăng ký” 13](#_Toc92063931)

[3.2.3 Use case “Quản lý thông tin cá nhân” 15](#_Toc92063932)

[3.2.4 Use case “Tìm kiếm” 18](#_Toc92063933)

[3.2.5 Use case “Quản lý đơn hàng cá nhân” 19](#_Toc92063934)

[3.2.6 Use case “Thanh toán” 22](#_Toc92063935)

# Đặt vấn đề

## Phân tích yêu cầu đề bài

### Tên đề tài

* Tổng quát: lập trình web sử dụng ReactJS, MongoDB.
* Cụ thể: thiết kế hệ thống quản lý bán giày dép online.

### Mục đích

Hệ thống quản lý bán giày dép online được thiết kế với mục đích đúng như tên gọi. Cụ thể, website giúp quản lý và bán hàng tập trung vào loại mặt hàng giày dép qua hình thức mua sắm trực tuyến cho 1 cửa hàng. Qua đó, chủ cửa hàng có thể tăng trải nghiệm mua hàng của các khách hàng gặp khó khăn trong quá trình di chuyển đến cửa hàng bằng việc cung cấp các hình ảnh trực quan về sản phẩm, giá cả công khai rõ ràng cùng phương thức thanh toán thuận tiện (COD).

Sử dụng website bán chuyên về 1 loại mặt hàng thay vì các trang thương mại điện tử lớn như Shopee, Lazada, Sendo, … góp phần giúp khách hàng giảm thiểu nỗi lo về mua hàng giả, hàng kém chất lượng từ vô số các shop.

Nói ngắn gọn, đây là một website chuyên nhập các đôi giày được nhiều người ưa chuộng (theo trend) từ các hãng phổ biến trên thế giới (Nike, Adidas, Jordan, …) để bán lại cho người dùng.

## Chức năng

Các chức năng chính hỗ trợ việc tương tác với hệ thống quản lý bán giày dép online và quản trị hệ thống.

#### Các chức năng tương tác với hệ thống:

1. Đăng ký thành viên:

* Để sử dụng website với các tác vụ nâng cao thì mỗi thành viên cần có tài khoản riêng.
* Tài khoản đó sau khi đăng kí thành công có thể đăng nhập vào hệ thống. Các tài khoản mặc định có các quyền cơ bản như nhau.

1. Chức năng đăng nhập/đăng xuất:

* Tài khoản đăng nhập hệ thống với đúng tài khoản và mật khẩu mà lúc đăng ký đã cung cấp.
* Tài khoản đăng nhập nếu không còn nhu cầu sử dụng hệ thống hoặc cần đăng nhập tài khoản khác có thể tiến hành đăng xuất.

1. Chức năng cập nhật thông tin cá nhân và mật khẩu:

* Hệ thống cho phép thành viên thay đổi thông tin cá nhân.
* Thành viên có thể thay đổi mật khẩu nếu muốn. Khi đổi mật khẩu sẽ phải xác nhận mật khẩu cũ, hệ thống sẽ gửi request về backend để kiểm tra và trả về response cho từng trường hợp thành công hay thất bại.

1. Chức năng xem mặt hàng và đặt đơn hàng:

* Khách hàng có thể xem các mặt hàng đang có và thêm vào giỏ hàng.
* Sau khi thêm vào giỏ thì có thể đặt đơn hàng. Người dùng cần đăng nhập và trong giỏ hàng phải có hàng nếu không thì hệ thống sẽ thông báo lỗi và redirect về trang xem sản phẩm.

1. Chức năng xem lịch sử mua hàng:

* Thành viên có thể xem lại lịch sử các đơn hàng đã mua, bao gồm giá tiền, trạng thái, link sản phẩm.

1. Chức năng theo dõi đơn hàng:

* Thành viên có thể kiểm tra tiến trình các đơn hàng mình vừa mới đặt thông qua tài khoản dùng để đặt đơn.

#### Các chức năng quản trị hệ thống

Các chức năng quản trị hệ thống chỉ có thể thực hiện bởi quản trị viên.

Cụ thể ở các chức năng như sau:

1. Quản lý thông tin thành viên:

* Quản lý thông tin khách hàng tồn tại trên hệ thống: xem lịch sử giao dịch, chỉnh sửa thông tin cá nhân thành viên, xóa thành viên.

1. Quản lý thông tin các sản phẩm:

* Tạo sản phẩm mới.
* Chỉnh sửa sản phẩm: chỉnh sửa tên, giá, nhãn hàng, số lượng sản phẩm và các thông tin liên quan như hình ảnh, giới thiệu sản phẩm, …
* Xóa sản phẩm.

1. Xem và duyệt đơn hàng:

* Xem trạng thái đơn hàng trên có trên hệ thống (thành công, hiện tại, thất bại)
* Chỉnh sửa các đơn hàng đang chạy: huỷ đơn hàng, chỉnh sửa thông tin đơn hàng, xác nhận đơn hàng.

## Quy mô, phạm vi

### Mặt hàng nhắm tới:

Website tập trung vào quản lý và bán mặt hàng thuộc nhóm ngành thời trang, cụ thể là: giày, dép.

Những đôi giày, dép này không đến từ thương hiệu chủ cửa hàng mà được order, nhập khẩu từ nhiều nguồn trên các nước, sau đó, bán lại cho khách hàng.

### Phạm vi phục vụ

Phạm vi hoạt động và phục vụ của website áp dụng cho 1 cửa hàng hoặc 1 chuỗi cửa hàng.

# Lý thuyết liên quan

Để giải quyết được vấn đề được nêu ra trong chương 1 cũng như hoàn thiện được sản phẩm cho project 2, nhưng kiến thức về thiết kế website là vô cùng cần thiết. Nói rõ hơn, em đã phải tìm hiểu và nắm được các phần kiến thức sau để tiến hành làm sản phẩm:

* Cách viết HTML, CSS
* Ngôn ngữ JavaScript
* Các framework, library liên quan hỗ trợ cho việc thiết kế Front-end và Back-end: ReacJS, NodeJS, Express, …
* Thiết kế cơ sở dữ liệu NoSQL: MongoDB

## Cách viết HTML, CSS

### HTML

Ở Project này, việc viết các đoạn HTML ít khi xảy ra vì đã áp dụng ReactJS trong quá trình thiết kế Front-end cho website. Tuy nhiên, nắm được bản chất và cách hoạt động của HTML vẫn là điều cần thiết nhằm phục vụ trong việc fix bug về giao diện sản phẩm.

Trong phần kiến thức này, em đã tìm hiểu và nắm được:

* Các thẻ cơ bản trong HTML
* Cách chia bố cục sử dụng các thẻ đặc biệt: div, span
* Cách xem và chỉnh sửa HTML trong developer tool của tình duyệt

### CSS

Tương tự với HTML, trong project này, em không thường xuyên phải viết CSS thuần để style cho website, thay vào đó, em sử dụng thư viện từ bên thứ 3 tên là Material UI. Nhưng, thư viện này cũng hỗ trợ lập trình viên custom các style có sẵn dựa trên cú pháp chuẩn của CSS nên các kiến thức sau vẫn cần nắm vững:

* Các thuộc tính cơ bản và nâng cao: text-align, font-size, flex-box, …
* Các selection trong CSS
* Thứ tự ưu tiên của các selection

## Ngôn ngữ JavaScript

Javascript có thể coi là ngôn ngữ cần phải biết khi thiết kế một website phong phú về mặt tính năng. Đây là ngôn ngữ được hỗ trợ compile và chạy trực tiếp trên các trình duyệt hiện nay.

Website được xây dựng chủ yếu sử dựng ReactJS, NodeJS, … , chúng đều là các framework/library của ngôn ngữ Javascript. Chính vì vậy, để hiểu và áp dụng được các framework/library trên, em cần tìm hiểu và áp dụng Javascript trước. Ngoài ra, Javascript cũng là ngôn ngữ được áp dụng trong việc xử lý dữ liệu ở phía Back-end. Sau đây là các kiến thức quan trọng của phần này:

* Nắm được cú pháp cơ bản trong việc: tạo biến, viết hàm, vòng lặp, if else,…
* Nắm được các dạng dữ liệu phổ biến: integer, string, double, array, object, …
* Lập trình bất đồng bộ với: promise, async, await
* Các gửi request, nhận response đến/từ API với: axios
* Import, Export được các module.

## Các framework, library liên quan hỗ trợ cho việc thiết kế Front-end và Back-end

### Front-end

ReactJS là thư viện chính được sử dụng trong quá trình thiết kế Front-end cho website. Các kiến thức cần hiểu để sử dụng được ReactJS:

* Cách tạo web app với NodeJS và ReacJS
* Cách ReactJS hoạt động:
  + Component
  + State
  + Prop
  + Hook
  + Life circle
* Các framework/library hỗ trợ:
  + MaterialUI: thiết kế giao diện dựa trên các component được tạo sẵn.
  + React Router: hỗ trợ chia route và điều hướng các phần của website.
  + Một vài framework/library hỗ trợ khác.

### Back-end

ExpressJS là framework chính trong việc lập trình Back-end cho website. Các kiến thức quan trọng để sử dụng được ExpressJS là:

* Require statements
* Middleware
* Routing
* App.listen()/ Starting the server

Ngoài ra còn một vài framework khác hỗ trợ quá trình xử lý ở phía Back-end như:

* Axios: gửi/nhận thông tin đến/từ phía Front-end.
* express-jwt: tạo token đăng nhập.
* mongoose: tương tác với MongoDB.
* passport, passport-local, crypto: xử lý mã hóa, xác nhận mật khẩu, đăng nhập.

## Thiết kế cơ sở dữ liệu NoSQL

Theo như yêu cầu đề tài, website sử dụng MongoDB để xây dựng và truy xuất cơ sở dữ liệu. Ở phần này, các kiến thức sau đã được tìm hiểu:

* Cách tạo các schema cho các tác nhân: khách hàng, sản phẩm.
* Cách khởi tạo dữ liệu và up dữ liệu lên MongoDB.
* Cách query, update dữ liệu từ MongoDB.

## Cài đặt docker

### Cơ bản cấu trúc docker

Với ứng dụng được hoạt động dựa trên 2 phần chính là front-end ReactJS và back-end NodeJS kết nối với database MongoDB, ta sẽ cố gắng container hoá hệ thống với 3 docker images cho front-end, back-end và database

Việc cài đặt docker sẽ giúp hệ thống hoạt động dễ dàng hơn, hiệu quả hơn và đảm bảo phiên bản, việc này bao gồm:

* Tạo Dockerfile cho phần backend và frontend
* Tạo Docker Compose config file ở định dạng YAML
* Cài đặt biến môi trường
* Khởi chạy hệ thống

### Cấu hình Dockerfile

Dockerfile sử dụng để khai báo các danh sách lệnh mà Docker sử dụng d dể cài đặt môi trường NodeJS cho cả backend và fronend. File này có ở cả front-end và back-end.

Trong file này có các điểm cần chú ý:

* FROM: Cài đặt phiên bản của NodeJS
* WORKDIR: Đường dẫn của ứng dụng.
* COPY: sao chép *package.json* file tới container, và sao chép các files khác trong thư mục của project.
* RUN: Thực thi câu lệnh trên terminal trong. Ví dụ như npm install để cài đặt thư viện cho tất cả các thư viện được mô tả trong package.json
* ENV: Cài đặt các biến môi trường cho ứng dụng sử dụng
* CMD: Chạy thực thi sau khi build thành công image.

Dưới đây là cấu hình dockerfile back-end, front-end tương tự

FROM node:lts-alpine

ENV NODE\_ENV=*production*

WORKDIR */usr/src/app*

COPY *[*"package.json"*,* "package-lock.json\*"*,* "npm-shrinkwrap.json\*"*,* "./"*]*

RUN *npm* *install* *--production* *--silent* *&&* *mv* *node\_modules* *../*

COPY *.* *.*

ENV PORT=*5000*

ENV DB\_NAME=*PROJECTX*

ENV DB\_HOST=*admin*

ENV DB\_KEY=*QcVC12L5pDXhieTK*

ENV DB\_PORT=*@cluster0.tizlc.mongodb.net*

ENV JWT\_SECRET="PROJECTX\_SECRETKEY"

EXPOSE *5000*

RUN *chown* *-R* *node* */usr/src/app*

USER *node*

CMD *npm* *start*

### Cấu hình Docker Compose file

Ở trong file ta sẽ cấu hình:

* version: Phiên bản Docker Compose sẽ được sử dụng.
* services: Các service riêng lẻ sẽ khởi tạo và chạy trong container của chúng ta. Ứng dụng có ba dịch vụ: frontend (ReactJS), backend (NodeJS) và mongodb (cơ sở dữ liệu MongoDB).
* services: Khởi tạo tên vùng chứa dữ

Chi tiết.

– **mongodb**:

* image: tên image của MongoDB chính thức trên Docker
* ports: cổng đầu ra từ trong container map ra ngoài
* volumes: chỉ tới thư mục chứa dữ liệu

– **backend**:

* image: tên image của backend
* build: các cấu hình được áp dụng tại thời điểm build mà ta cài đặt trong Dockerfile với đường dẫn tương đối
* environment: Các biến môi trường thêm mà biến trong ứng dụng sử dụng đầu ra
* ports: Cổng đầu ra chỉ tới từ container ra ngoài

– **frontend**:

* image: tên image của frontend
* build: các cấu hình được áp dụng tại thời điểm build mà ta cài đặt trong Dockerfile với đường dẫn tương đối
* environment: Các biến môi trường thêm mà biến trong ứng dụng sử dụng đầu ra
* ports: Cổng đầu ra chỉ tới từ container ra ngoài

# Phân tích và thiết kế hệ thống

Cơ sở dữ liệu

Ảnh có chứa văn bản

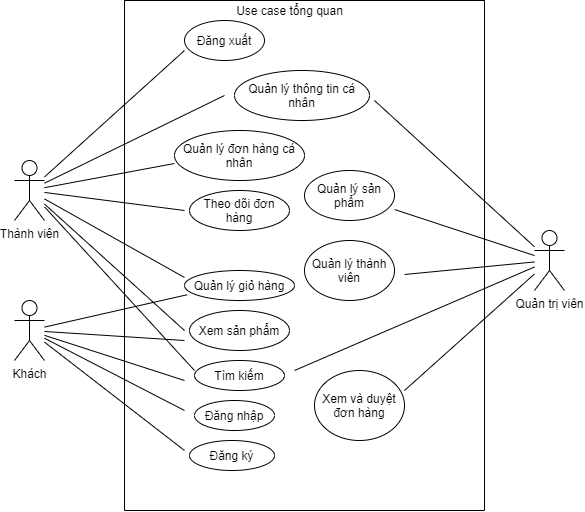
Mô tả được tạo tự động

Hình 1. Biểu đồ cơ sở dữ liệu

## Phân tích use case

### Use case tổng quan

Use case tổng quan biểu diễn chức năng mà hệ thống cung cấp cho các loại tác nhân khác nhau như Khách, Khách hàng hệ thống, Quản trị viên



Biểu đồ use case tổng quan

### Phân rã use case

**Diagram

Description automatically generated**

Phân rã use case “Quản lý giỏ hàng”

**Diagram

Description automatically generated**

Phân rã use case “Quản lý đơn hàng cá nhân”

**Diagram

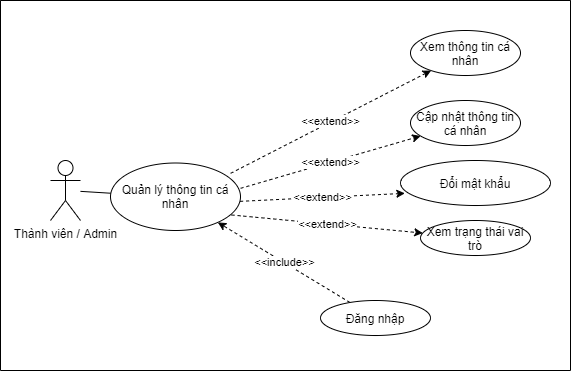
Description automatically generated**

Phân rã use case “Thanh toán”

**Diagram

Description automatically generated**

Phân rã use case “Tìm kiếm”



Phân rã use case “Quản lý thông tin cá nhân”

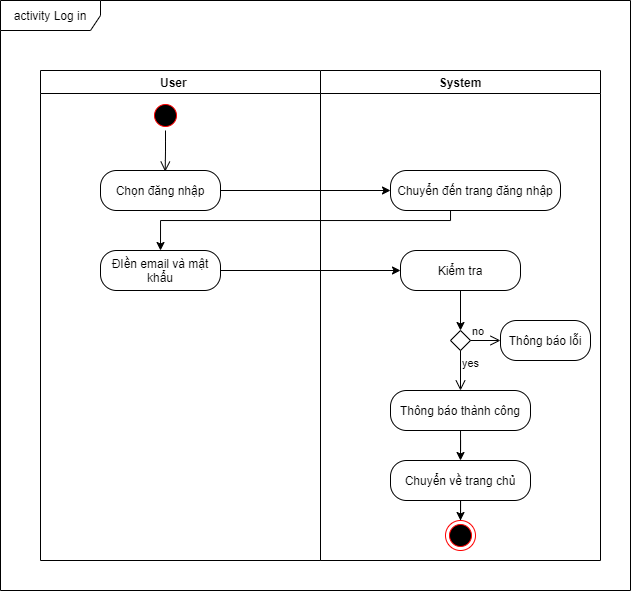
## Đặc tả use case

Chi tiết về các use case được đưa ra trong phần 3.2.2 và một số use case khác được đặc tả dưới đây.

### Use case “Đăng nhập”

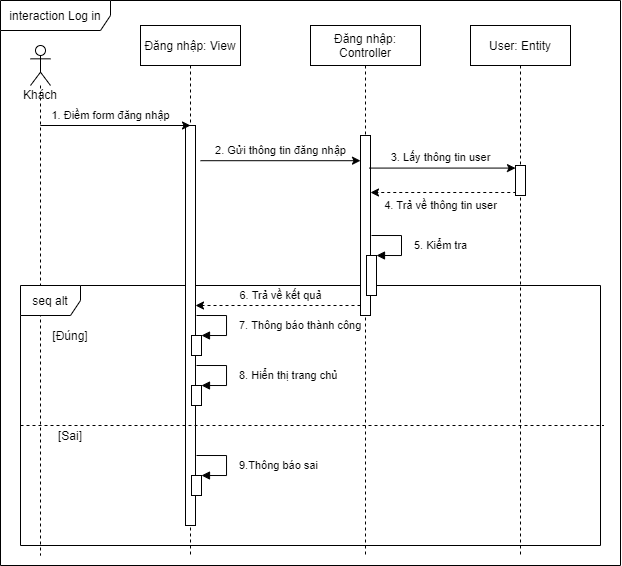
|  |  |
| --- | --- |
| **Tên use case** | Đăng nhập |
| **Tác nhân** | Khách |
| **Tiền điều kiện** | Không |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | |  | Khách | Chọn chức năng Đăng nhập | |  | Hệ thống | Chuyển hướng đến trang đăng nhập | |  | Khách | Nhập email và mật khẩu | |  | Khách | Yêu cầu đăng nhập | |  | Hệ thống | Kiểm tra xem khách đã nhập các trường bắt buộc nhập hay chưa | |  | Hệ thống | Kiểm tra email và mật khẩu có hợp lệ do khách nhập trong hệ thống hay không | |  | Hệ thống | Trở về trang chủ hệ thống | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 6a. | Hệ thống | Thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu khách nhập thiếu | | 7a. | Hệ thống | Thông báo lỗi: Email và/hoặc mật khẩu chưa đúng nếu không tìm thấy email và mật khẩu kèm theo trường hệ thống | |
| **Hậu điều kiện** | Không |

#### Biểu đồ hoạt động



Hình 12. Biểu đồ hoạt động Đăng nhập

#### Biểu đồ tuần tự

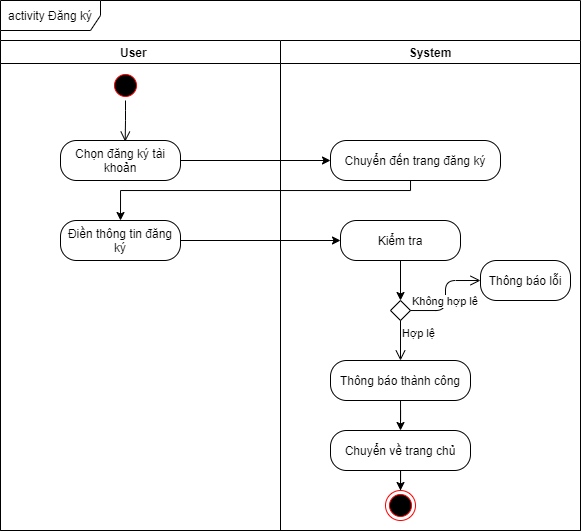


Hình 13. Biểu đồ tuần tự cho hoạt động Đăng nhập

### Use case “Đăng ký”

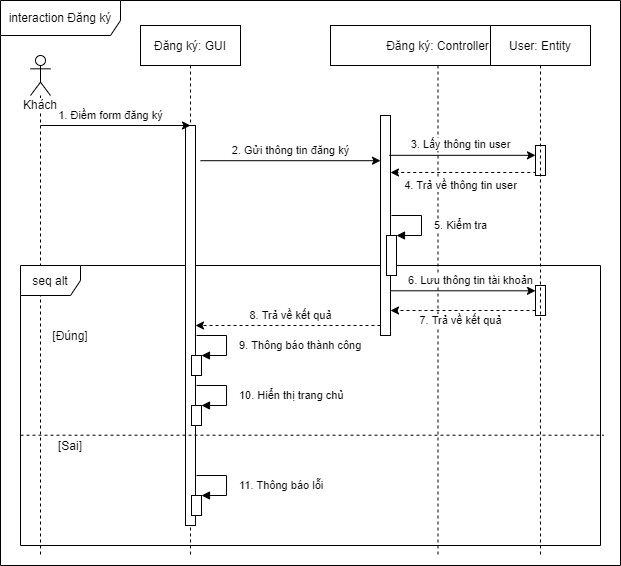
|  |  |
| --- | --- |
| **Tên Use case** | Đăng ký |
| **Tác nhân** | Khách |
| **Tiền điều kiện** | Không |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Khách | Chọn chức năng Đăng ký | | 2 | Hệ thống | Chuyển hướng đến trang đăng ký | | 3 | Khách | Điền thông tin cá nhân | | 4 | Khách | Yêu cầu đăng ký | | 5 | Hệ thống | Kiểm tra xem khách đã nhập các trường bắt buộc nhập hay chưa | | 6 | Hệ thống | Kiểm tra địa chỉ email, vai trò của khách có hợp lệ không | | 7 | Hệ thống | Kiểm tra mật khẩu nhập lại và mật khẩu có trùng nhau hay không | | 8 | Hệ thống | Kiểm tra mật khẩu có đủ mức độ an toàn hay không | | 9 | Hệ thống | Lưu thông tin tài khoản và thông báo đăng ký thành công | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 6a | Hệ thống | Thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu khách nhập thiếu | | 7a | Hệ thống | Thông báo lỗi: Địa chỉ email không hợp lệ nếu địa chỉ email không hợp lệ | | 8a | Hệ thống | Thông báo lỗi: Mật  khẩu xác nhận không trùng với Mật khẩu nếu hai mật khẩu không trùng nhau | | 9a | Hệ thống | Thông báo lỗi: Mật khẩu cần đảm bảo độ an toàn  nếu mật khẩu không đảm bảo độ an toàn được quy định bởi hệ thống (có ít nhất 8 ký tự, trong đó có cả chữ cái thường và chữ cái hoa, ít nhất 1 chữ số hoặc 1 ký tự đặc biệt) | |
| **Hậu điều kiện** | Không |

#### Biểu đồ hoạt động



Hình 14. Biểu đồ hoạt động “Đăng ký”

#### Biểu đồ tuần tự

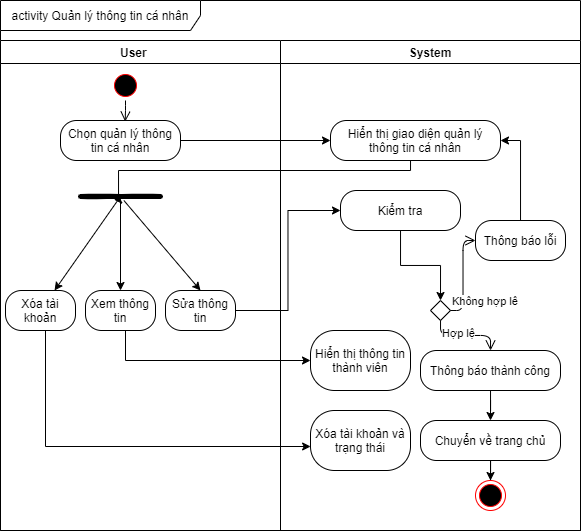


Hình 15. Biểu đồ tuần tự hành động “Đăng ký”

### Use case “Quản lý thông tin cá nhân”

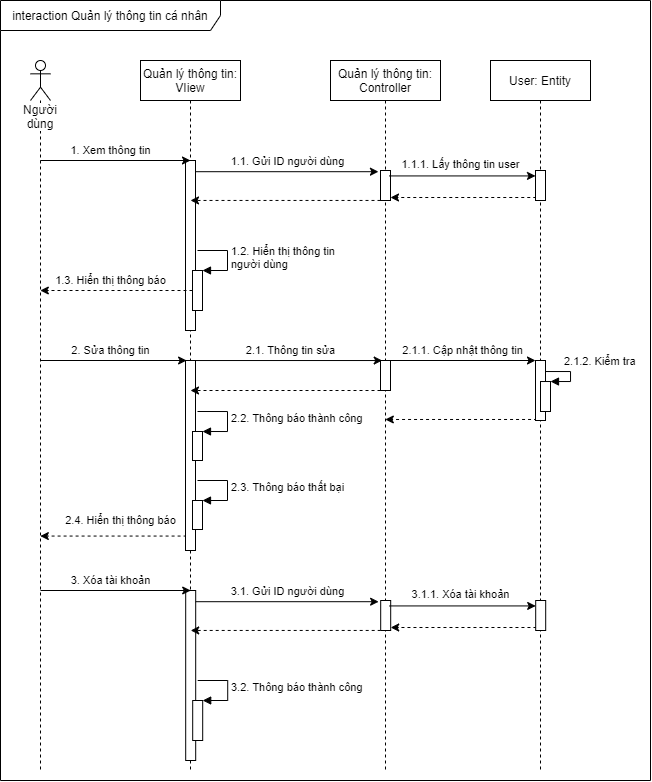
|  |  |
| --- | --- |
| **Tên Use case** | Quản lý thông tin cá nhân |
| **Tác nhân** | Quản trị viên, Thành viên |
| **Tiền điều kiện** | Khách đăng nhập thành công |
| |  |  |  |  | | --- | --- | --- | --- | | **Xem** | | | | | **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Người dùng | yêu cầu Xem thông tin cá nhân thành viên | | 2 | Hệ thống | lấy thông tin của thành viên tương ứng | | 3 | Hệ thống | hiển thị giao diện Xem thông tin thành viên gồm:   * Xem chi tiết thành viên * Vai trò hiện tại và trạng thái * Các nút để chỉnh sửa thông tin |  |  |  |  |  | | --- | --- | --- | --- | | **Sửa** | | | | | **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Thành viên | chọn thông tin cá nhân và yêu cầu chỉnh sửa | | 2 | Thành viên | chỉnh sửa các thông tin cá nhân và yêu cầu lưu. | | 3 | Hệ thống | kiểm tra các trường bắt buộc nhập | | 4 | Hệ thống | kiểm tra các trường cần đảm bảo duy nhất | | 5 | Hệ thống | cập nhật các thông tin cần chỉnh sửa và thông báo chỉnh sửa thành công. | | **Luồng sự kiện thay thế** | 3a | Hệ thống | thông báo lỗi: chưa nhập đủ các trường cần thiết nếu quản trị viên nhập thiếu trường | | 4a | Hệ thống | thông báo lỗi: thông báo cụ thể trường nào không duy nhất nếu quản trị viên nhập trùng |  |  |  |  |  | | --- | --- | --- | --- | | **Xoá** | | | | | **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Thành viên | chọn một thông tin và yêu cầu xoá | | 2 | Hệ thống | hiển thị thông báo yêu cầu người dùng xác nhận việc xoá | | 3 | Thành viên | xác nhận xoá tài khoản | | 4 | Hệ thống | xoá thành viên, vai trò và thông báo xoá thành công | | **Luồng sự kiện thay thế** | 4a | Hệ thống | Giữ nguyên use case nếu người dùng xác nhận không xoá | | |
| **Hậu điều kiện** | Không |

#### Biểu đồ hoạt động



Hình 16. Biểu đồ hoạt động “Quản lý thông tin cá nhân”

#### Biểu đồ tuần tự

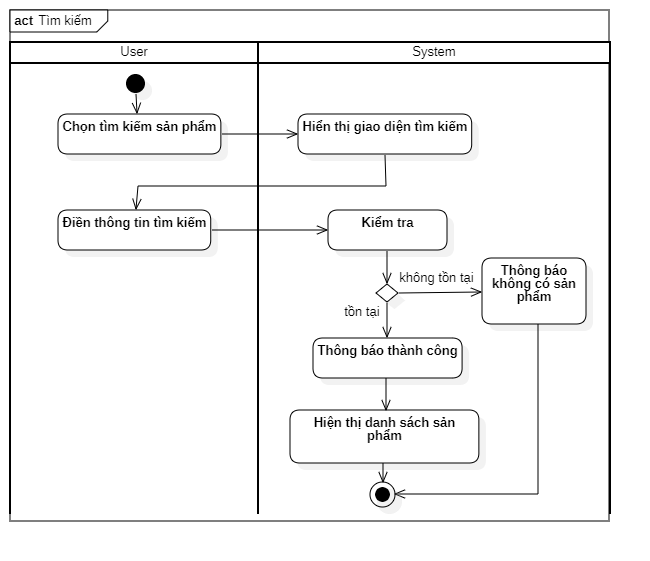


Hình 17. Biểu đồ tuần tự “Quản lý thông tin cá nhân”

### Use case “Tìm kiếm”

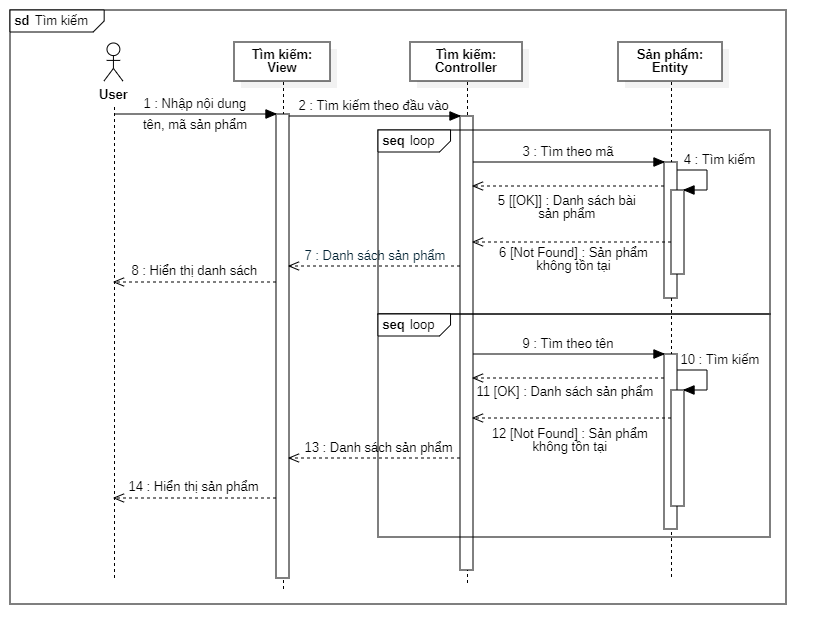
|  |  |
| --- | --- |
| **Tên Use case** | Tìm kiếm hàng hóa |
| **Tác nhân** | Quản trị viên, Khách, Thành viên |
| **Tiền điều kiện** | Không |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | |  | Tác nhân | Chọn chức năng Tìm kiếm hàng hoá | |  | Hệ thống | Lấy danh sách hàng hoá và hiển thị giao diện tìm kiếm hàng hoá gồm những thông tin xung quanh | |  | Tác nhân | Nhập từ khoá loại hàng hoá muốn tìm kiếm (mô tả phía dưới \*) | |  | Tác nhân | Yêu cầu tìm kiếm | |  | Hệ thống | Kiểm tra xem khách đã nhập từ khoá tìm kiếm chưa | |  | Hệ thống | Tìm và lấy về thông tin những người dùng thoả mãn các từ khoá tìm kiếm bao gồm tên sản phẩm, mã sản phẩm | |  | Hệ thống | Hiển thị danh sách những hàng hoá thoả mãn từ khoá tìm kiếm nếu có ít nhất một hàng hoá được tìm thấy | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 6a. | Hệ thống | Nếu mà không nhập bất kỳ từ khoá nào thì hiển thị tất cả hàng hoá | | 7a. | Hệ thống | Thông báo: Không tìm thấy thông tin hàng hoá nào thỏa mãn từ khóa tìm kiếm nếu trả về danh sách rỗng | |
| **Hậu điều kiện** | Không |

#### Biểu đồ hoạt động



Hình 18. Biểu đồ hoạt động “Tìm kiếm”

#### Biểu đồ tuần tự



Hình 19. Biểu đồ tuần tự “Tìm kiếm”

### Use case “Quản lý đơn hàng cá nhân”

|  |  |
| --- | --- |
| **Tên Use case** | Quản lý đơn hàng cá nhân |
| **Tác nhân** | Khách hàng hệ thống |
| **Tiền điều kiện** | Khách đăng nhập thành công với vai trò khách hàng hệ thống |
| |  |  |  |  | | --- | --- | --- | --- | | **Xem** | | | | | **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Khách hàng | Yêu cầu Xem danh sách đơn hàng | | 2 | Hệ thống | Lấy danh sách tất cả đơn hàng | | 3 | Hệ thống | Hiển thị giao diện Xem danh sách đơn hàng gồm:   * Phía trên: Danh sách nhóm đơn hàng để người dùng có thể lọc theo thời gian, trạng thái. * Trung tâm: Danh sách đơn hàng của người dùng (mô tả phía dưới \*) |  |  |  |  |  | | --- | --- | --- | --- | | **Sửa** | | | | | **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Khách hàng | Chọn một đơn hàng đang không ở trạng thái đã hoàn thành | | 2 | Hệ thống | Lấy thông tin chi tiết của đơn hàng và hiển thị thông tin đơn hàng trên giao diện sửa đơn hàng | | 3 | Khách hàng | Chỉnh sửa các thông tin đơn hàng và yêu cầu sửa | | 4 | Hệ thống | Kiểm tra các trường bắt buộc nhập | | 5 | Hệ thống | Kiểm tra các trường cần đảm bảo duy nhất | | 6 | Hệ thống | Cập nhật các thông tin cần chỉnh sửa và thông báo chỉnh sửa thành công. | | **Luồng sự kiện thay thế** | 5a | Hệ thống | Thông báo lỗi: chưa nhập đủ các trường cần thiết nếu khách hàng nhập thiếu trường | | 6a | Hệ thống | Thông báo lỗi: thông báo cụ thể trường nào không duy nhất nếu khách hàng nhập trùng |  |  |  |  |  | | --- | --- | --- | --- | | **Hủy đơn** | | | | | **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Khách hàng | Chọn một đơn hàng và yêu cầu huỷ đơn | | 2 | Hệ thống | Hiển thị thông báo yêu cầu người dùng xác nhận việc huỷ đơn | | 3 | Khách hàng | Xác nhận huỷ hàng hoá | | 4 | Hệ thống | Huỷ đơn hàng và thông báo huỷ thành công | | **Luồng sự kiện thay thế** | 4a | Hệ thống | Giữ nguyên use case nếu người dùng xác nhận không huỷ |  |  |  |  |  | | --- | --- | --- | --- | | **Tạo đơn** | | | | | **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Khách hàng | Yêu cầu tạo đơn hàng hoá mới | | 2 | Hệ thống | Lấy giỏ hàng hiện tại và hiển thị giao diện nhập thông tin giao hàng | | 3 | Khách hàng | Nhập các thông tin đơn hàng và yêu cầu tạo đơn | | 4 | Hệ thống | Kiểm tra các trường bắt buộc nhập | | 5 | Hệ thống | Kiểm tra các trường cần đảm bảo duy nhất | | 6 | Hệ thống | Cập nhật các thông tin cần tạo đơn và thông báo tạo đơn thành công. | | **Luồng sự kiện thay thế** | 5a | Hệ thống | Thông báo lỗi: chưa nhập đủ các trường cần thiết nếu khách hàng nhập thiếu trường | | 6a | Hệ thống | Thông báo lỗi: thông báo cụ thể trường nào không duy nhất nếu khách hàng trùng | | |
| **Hậu điều kiện** | Không |

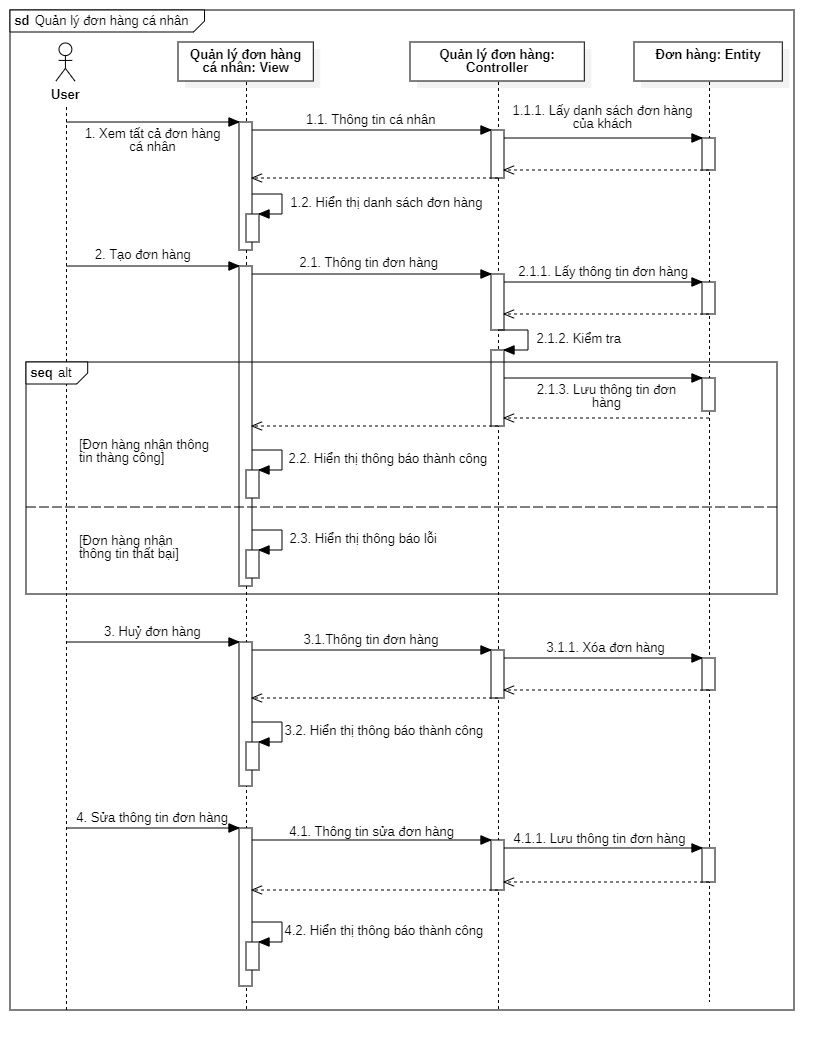
#### Biểu đồ hoạt động

Ảnh có chứa văn bản, đồng hồ đo, bãi đậu xe, đen

Mô tả được tạo tự động

Hình 24. Biểu đồ hoạt động “Quản lý đơn hàng cá nhân”

#### Biểu đồ tuần tự



Hình 25. Biểu đồ tuần tự “Quản lý đơn hàng cá nhân”

### Use case “Thanh toán”

|  |  |
| --- | --- |
| **Tên Use case** | Thanh toán |
| **Tác nhân** | Khách hàng hệ thống |
| **Tiền điều kiện** | Khách đăng nhập thành công với vai trò khách hàng hệ thống, tạo đơn hàng thành công và tiến đến thanh toán |
| |  |  |  |  | | --- | --- | --- | --- | | **Luồng sự kiện chính** | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Khách hàng | Nhấn thanh toán để tạo đơn hàng | | 2 | Hệ thống | Lấy thông tin đơn hàng và tạo thông tin mặc định | | 3 | Hệ thống | Hiển thị giao diện khuôn thanh toán   * Phía trên: hiển thị thông tin cơ bản đơn hàng, giá tiền, số lượng sản phẩm * Trung tâm: danh sách sản phẩm chi tiết, thông tin mặc định của sản phẩm và đơn | | 4 | Khách hàng | Nhập tiếp các thông tin: họ tên, số tài khoản, phương thức thanh toán, địa chỉ giao hàng, thời gian giao hàng, địa chỉ email để hoàn thành đơn hàng | | 5 | Khách hàng | Chọn hoàn tất thông tin đơn hàng, tiến hàng chọn hình thức thanh toán | | 6 | Khách hàng | Chọn hình thức thanh toán giữa COD (tiến tới 6a) và thanh toán qua thẻ (tiến tới 6b) | | 7 | Hệ thống | Xác nhận thông tin thanh toán | | 8 | Hệ thống | Thông báo giao dịch thành công | | **Luồng sự kiện thay thế** | 5a | Hệ thống | Thông báo lỗi: chưa nhập đủ các trường cần thiết nếu khách hàng nhập thiếu trường | | 5a.1 | Khách hàng | Nhập lại thông tin và trở lại sử kiện chính hoặc nếu huỷ đơn sẽ tiến đến 5a.2 | | 5b.2.1 | Khách hàng | Ấn huỷ bỏ đơn hàng | | 5b.2.2 | Hệ thống | Hiện thị các sản phẩm để khách hàng thực hiện giao dịch mới | | 6a.1 | Hệ thống | Gửi đơn hàng lên hệ thống chung chờ quản trị viên xác nhận đơn. (nếu quản trị viên từ chối sẽ tiến tới 6a.1.1 | | 6a.2 | Hệ thống | Lưu lại đơn hàng và tiếp tục luồng sự kiện chính | | 6a.1.1 | Hệ thống | Thông báo đơn hàng không được chuấp thuận, hiển thị các sản phẩm để khách hàng thực hiện giao dịch mới | | 6b | Hệ thống và đơn vị thẻ | Kiểm tra số tài khoản của khách hàng và phương thức thanh toán. Nếu hợp lệ và đã hoàn thành thì trở lại luồng sự k­­­­iện chính, nếu sai thì sẽ tiến tới luồng 6b.1 | | 6b.1.1 | Hệ thống | Thông báo phương thức thanh toán và tài khoản không hợp lệ | | 6b.1.2 | Khách hàng | Nhập lại thông tin và trở lại sự kiện chính, nếu chọn muốn ngưng giao dịch thì tiến tới 6b.1.3 | | 6b.1.3 | Hệ thống | Hiện thị các sản phẩm để khách hàng thực hiện giao dịch mới | | |
| **Hậu điều kiện** | Không |