

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA CÔNG NGHỆ PHẦN MỀM**



**Nguyễn Hoàng Thái Dương      19521409**

**Nguyễn Âu Duy      19521423**

**ĐỒ ÁN 2**

**19**

**Tìm hiểu Flask và  
Xây dựng Web bán phim online kết hợp  
Khuyến nghị sản phẩm**

**GIẢNG VIÊN GIẢNG DẠY**

**TS. Nguyễn Hồng Thủy**

**TP. HỒ CHÍ MINH, 2022**

## **LỜI CẢM ƠN**

Nhóm xin gửi lời cảm ơn chân thành đến Cô Nguyễn Hồng Thủy đã tận tình hướng dẫn chúng em trong suốt thời gian vừa qua và các bạn học đã góp ý và giúp đỡ nhóm trong quá trình thực hiện đồ án này.

Do kiến thức và thời gian thực hiện hạn chế, đồ án của nhóm vẫn còn nhiều thiếu sót. Nhóm rất mong nhận được góp ý của Cô và các bạn để đồ án của nhóm được hoàn thiện.

## MỤC LỤC

|   |    |
|---|----|
| Chương 1. GIỚI THIỆU CHUNG .....                          | 1  |
| 1.1. Lý do chọn đề tài .....                              | 1  |
| 1.2. Tổng quan về chức năng .....                         | 2  |
| 1.2.1. Người dùng .....                                   | 2  |
| 1.2.2. Nhân viên .....                                    | 2  |
| Chương 2. CƠ SỞ LÝ THUYẾT .....                           | 3  |
| 2.1. ReactJs .....  | 3  |
| 2.1.1. Giới thiệu .....                                   | 3  |
| 2.1.2. Các khái niệm chính trong React (ReactJs) .....    | 3  |
| 2.1.3. Cách hoạt động .....                               | 10 |
| 2.1.4. Nhược điểm .....                                   | 10 |
| 2.1.5. Ưu điểm và Tại sao nên dùng ReactJs .....          | 11 |
| 2.1.6. Các framework-package liên quan đến ReactJs .....  | 12 |
| 2.2. Spring boot .....                                    | 21 |
| 2.2.1. Khái niệm .....                                    | 21 |
| 2.2.2. Nhược điểm của Spring Boot .....                   | 21 |
| 2.2.3. Ưu điểm của Spring Boot .....                      | 22 |
| 2.2.4. Các thư viện được sử dụng trong Spring Boot .....  | 22 |
| 2.2.5. Xây dựng back-end cho đồ án bằng Spring Boot ..... | 26 |
| 2.3. Flask .....  | 28 |
| 2.3.1. Tổng quan Python .....                             | 28 |

|  |    |
|--|----|
| 2.3.2. Độ phổ biến trong năm 2022 .....                | 29 |
| 2.3.3. Ứng dụng của Python .....                       | 29 |
| 2.3.4. Ưu điểm .....                                   | 32 |
| 2.3.5. Nhược điểm .....                                | 32 |
| 2.3.6. Flask framework .....                           | 32 |
| 2.3.7. Flask API .....                                 | 39 |
| 2.4. Machine Learning.....                             | 40 |
| 2.4.1. User mới .....                                  | 41 |
| 2.4.2. User đã có rates.....                           | 42 |
| 2.5. Frameworks và tools khác.....                     | 43 |
| 2.5.1. IntelliJ.....                                   | 43 |
| 2.5.2. Visual Studio Code.....                         | 43 |
| 2.5.3. Postman .....                                   | 44 |
| 2.5.4. Microsoft SQL Server .....                      | 44 |
| 2.5.5. Github .....                                    | 45 |
| 2.5.6. Ngôn ngữ Java .....                             | 45 |
| 2.5.7. Ngôn ngữ thiết kế Web HTML-CSS-Javascript ..... | 46 |
| 2.5.8. Bootstrap.....                                  | 47 |
| Chương 3. XÂY DỰNG HỆ THỐNG.....                       | 48 |
| 3.1. Xây dựng kiến trúc hệ thống .....                 | 48 |
| 3.1.1. Sơ đồ hệ thống .....                            | 48 |
| 3.1.2. Sơ đồ use case.....                             | 49 |
| 3.1.3. Sơ đồ lớp.....                                  | 51 |

|                                    |    |
|------------------------------------|----|
| 3.1.4. Sơ đồ tuần tự.....          | 52 |
| 3.1.5. Cơ sở dữ liệu.....          | 67 |
| 3.2. Thiết kế giao diện.....       | 71 |
| 3.2.1. Giao diện ứng dụng .....    | 71 |
| 3.2.2. Giao diện quản lý.....      | 79 |
| Chương 4. KẾT LUẬN.....            | 85 |
| 4.1. Kết quả đạt được .....        | 85 |
| 4.2. Ưu điểm.....                  | 85 |
| 4.3. Nhược điểm và giải pháp ..... | 85 |

## DANH MỤC HÌNH

|   |    |
|---|----|
| Hình 2-1 Mô hình DOM trong ứng dụng ReactJs   | 4  |
| Hình 2-2 Ví dụ về JSX   | 5  |
| Hình 2-3 Component  | 6  |
| Hình 2-4 Ví dụ props  | 7  |
| Hình 2-5 Component Lifecycle  | 8  |
| Hình 2-6 Dữ liệu số người dùng các Framework  | 12 |
| Hình 2-7 Ví dụ useState   | 13 |
| Hình 2-8 Ví dụ useEffect  | 13 |
| Hình 2-9 Thiết lập một bộ chuyển trang bằng React Router                                      | 15 |
| Hình 2-10 Ví dụ về cách dùng useNavigate  | 16 |
| Hình 2-11 Ví dụ cách dùng useLocation   | 16 |
| Hình 2-12 Ví dụ cách lấy params   | 17 |
| Hình 2-13 Cách dùng thẻ Link  | 18 |
| Hình 2-14 Mô hình Redux   | 19 |
| Hình 2-15 Các thành phần Slice, AsyncThunk trong ReduxToolkit                                 | 20 |
| Hình 2-16 Hibernate: các hàm tương ứng sẽ có các annotation query và câu lệnh query tương ứng | 23 |
| Hình 2-17 Ví dụ về việc sử dụng annotation  | 24 |
| Hình 2-18 Cấu trúc Backend đồ án.   | 26 |
| Hình 2-19 Python  | 28 |
| Hình 2-20 Python Google Trend   | 29 |
| Hình 2-21 Python TIOBE  | 29 |
| Hình 2-22 Thư viện Python hỗ trợ Web  | 30 |
| Hình 2-23 Tổng quan ứng dụng Python   | 30 |
| Hình 2-24 Python và Machine Learning  | 31 |
| Hình 2-25 Jinja2 template   | 34 |

|  |    |
|--|----|
| Hình 2-26 Cấu trúc Flask project               | 35 |
| Hình 2-27 Flask Routing                        | 36 |
| Hình 2-28 Flask static                         | 37 |
| Hình 2-29 Flask secret key                     | 37 |
| Hình 2- IntelliJ                               | 43 |
| Hình 2- Víual Studio Code                      | 43 |
| Hình 2- Postman                                | 44 |
| Hình 2- SQL Server                             | 44 |
| Hình 2- Github                                 | 45 |
| Hình 2- Java                                   | 45 |
| Hình 2- HTML-CSS-Javascript                    | 46 |
| Hình 2- Bootstrap                              | 47 |
| Hình 3-1 Sơ đồ kiến trúc hệ thống              | 48 |
| Hình 3-2 Sơ đồ use cases người dùng            | 49 |
| Hình 3-3 Sơ đồ use cases hệ thống              | 50 |
| Hình 3-4 Sơ đồ lớp                             | 51 |
| Hình 3-5 Sơ đồ tuần tự đăng nhập               | 52 |
| Hình 3-6 Sơ đồ tuần tự đăng ký                 | 53 |
| Hình 3-7 Sơ đồ tuần tự chỉnh sửa thông tin     | 54 |
| Hình 3-8 Sơ đồ tuần tự xóa tài khoản           | 55 |
| Hình 3-9 Sơ đồ tuần tự tạo tài khoản nhân viên | 56 |
| Hình 3-10 Sơ đồ tuần tự tạo phân loại          | 57 |
| Hình 3-11 Sơ đồ tuần tự sửa phân loại          | 58 |
| Hình 3-12 Sơ đồ tuần tự xóa phân loại          | 59 |
| Hình 3-13 Sơ đồ tuần tự tạo phim               | 60 |
| Hình 3-14 Sơ đồ tuần tự sửa thông tin phim     | 61 |
| Hình 3-15 Sơ đồ tuần tự xóa phim               | 62 |

|   |    |
|---|----|
| Hình 3-16 Sơ đồ tuần tự mua hàng                  | 63 |
| Hình 3-17 Sơ đồ tuần tự tình trạng đơn hàng       | 64 |
| Hình 3-18 Sơ đồ tuần tự thêm hàng vào giỏ         | 65 |
| Hình 3-19 Sơ đồ tuần tự xóa hàng khỏi giỏ         | 66 |
| Hình 3-20 Sơ đồ Database                          | 67 |
| Hình 3- Trang chủ và slider                       | 71 |
| Hình 3- Giao diện thanh đầu trang                 | 71 |
| Hình 3- Giao diện trang chủ với các phim bán chạy | 72 |
| Hình 3- Trang chủ và các phim phổ biến            | 72 |
| Hình 3- Giao diện danh sách phim                  | 73 |
| Hình 3- Giao diện chi tiết sản phẩm thông tin     | 74 |
| Hình 3- Giao diện chi tiết sản phẩm gọi           | 75 |
| Hình 3- Giao diện giỏ hàng                        | 76 |
| Hình 3- Giao diện thông tin khách                 | 76 |
| Hình 3- Giao diện đặt hàng                        | 77 |
| Hình 3- Giao diện đăng nhập                       | 78 |
| Hình 3- Giao diện đăng ký                         | 78 |
| Hình 3- Dashboard 1                               | 79 |
| Hình 3- Dashboard 2                               | 79 |
| Hình 3- Dashboard 3                               | 80 |
| Hình 3- Giao diện sản phẩm                        | 80 |
| Hình 3- Giao diện tạo mới sản phẩm phim           | 81 |
| Hình 3- Giao diện chỉnh sửa sản phẩm              | 81 |
| Hình 3- Giao diện phân loại                       | 82 |
| Hình 3- Giao diện đơn hàng                        | 82 |
| Hình 3- Giao diện chi tiết đơn hàng               | 83 |
| Hình 3- Giao diệ khách hàng                       | 83 |

Hình 3- Giao diện nhân viên \_\_\_\_\_ 84

Hình 3- Giao diện tạo nhân viên \_\_\_\_\_ 84

**DANH MỤC BẢNG**  
**DANH MỤC TỪ VIẾT TẮT**

|                   |    |
|-------------------|----|
| Bảng 3-1 Users    | 68 |
| Bảng 3-2 Roles    | 68 |
| Bảng 3-3 Cart     | 69 |
| Bảng 3-4 CartInfo | 69 |
| Bảng 3-5 Product  | 70 |
| Bảng 3-6 Category | 71 |

## TÓM TẮT ĐOÀN

Việc để tạo ra một trang Web bán hàng bắt mắt, tiện lợi cho người mua là một trong những vấn đề đang được quan tâm hàng đầu. Có rất nhiều cách để tạo ra một trang Web như thông qua các phần mềm thứ ba để tạo với những người thích thiết kế giao diện nhưng không có hiểu biết sâu rộng về lập trình hay lập trình với các ngôn ngữ, framework hỗ trợ phát triển ứng dụng Web : .NET, HTML-CSS, Javascript, Bootstrap, Vue.js, Angular... Trong đó nhóm đã chọn **ReactJs** cho phần thiết kế giao diện người dùng (front-end) và framework **Java SpringBoot** cho phần xử lý tác vụ logic, kết nối Database, để xây dựng một ứng dụng Web bán quần áo online.

Ngoài ra nhóm còn kết hợp ứng dụng Machine Learning cho website thông qua việc sử dụng **Framework Flask Python** hỗ trợ cho model **Recommender System** qua **API** về Recommender System sẽ giúp tăng cao trải nghiệm người dùng với những khuyến nghị sản phẩm liên quan (nổi bật, mua nhiều,...) dựa trên đánh giá của khách hàng. Từ đó việc mua hàng của người dùng sẽ dễ dàng hơn trong việc chọn lựa cũng như giúp cửa hàng có thể tăng số lượng bán hàng và làm hài lòng khách hàng.

# **Chương 1. GIỚI THIỆU CHUNG**

## **1.1. Lý do chọn đề tài**

Với sự phát triển nhảy vọt của công nghệ thông tin hiện nay, Internet ngày càng giữ vai trò quan trọng trong các lĩnh vực khoa học kĩ thuật và đời sống. Internet là tập hợp kết nối giữa các máy tính, là một mạng máy tính toàn cầu mà bất kì ai cũng có thể truy cập bằng laptop, PC, các thiết bị di động, thiết bị gia dụng thông minh,... Vì thế cùng với sự lớn mạnh của Internet thì hàng loạt các sản phẩm mang tính thương mại ra đời với quy mô từ nhỏ đến lớn ở các ngành nghề, lĩnh vực liên quan và phổ biến, thông dụng ở mọi lĩnh vực khác. Internet đã tạo ra một cuộc cách mạng trao đổi thông tin trong mọi lĩnh vực văn hóa, y tế, xã hội, giáo dục,... nhờ đó góp phần thúc đẩy sự phát triển cầu thế giới.

Trong đó nổi bật là các sản phẩm, thiết bị thương mại điện tử mà mọi người đều có thể sử dụng dễ dàng và hữu ích. Việc mua bán bây giờ không chỉ thông qua xem hàng – mua hàng trực tiếp mà còn có thể giao dịch qua các trang Web thương mại điện tử một cách tiện lợi nhưng vẫn đảm bảo được sự uy tín, chất lượng từ sản phẩm mà các doanh nghiệp, cửa hàng, siêu thị, cá nhân... cung cấp cho khách hàng cũng như sự hỗ trợ và chính sách bán hàng cần thiết.

Vì thế nhóm đã quyết định chọn đề tài để tạo nên một trang Web bán hàng để có thể đáp ứng nhu cầu người dùng cũng như người bán với các chức năng cần thiết nhất của một cửa hàng.

Thêm vào đó các công nghệ như AI, Blockchain, IoT cũng đang ngày càng phát triển cho hầu hết các lĩnh vực đời sống và tất nhiên việc bán hàng cũng là một thị trường tiềm năng. Bởi với lượng người dùng càng tăng và lượng dữ liệu có được sẽ là một tài nguyên cho việc nghiên cứu. áp dụng các công nghệ trên.

## **1.2. Tổng quan về chức năng**

### **1.2.1. Người dùng**

- Đăng nhập và đăng ký.
- Tìm kiếm sản phẩm và xem chi tiết sản phẩm.
- Đánh giá sản phẩm.
- Thêm vào giỏ hàng.
- Đặt hàng.
- Xem trailer sản phẩm.
- Xem, chỉnh sửa thông tin cá nhân.
- Xem lịch sử mua hàng.
- Xem gợi ý sản phẩm.

### **1.2.2. Nhân viên**

- Đăng nhập.
- Xem, thêm, xóa và chỉnh sửa phân loại sản phẩm theo quốc gia.
- Xem, thêm, xóa và chỉnh sửa sản phẩm.
- Tìm kiếm theo danh mục.
- Xem thông tin khách hàng.
- Xem chi tiết và chỉnh sửa trạng thái đơn hàng.
- Xem, xóa và tạo tài khoản nhân viên.
- Xem thống kê doanh thu, số lượng sản phẩm.

## **Chương 2. CƠ SỞ LÝ THUYẾT**

### **2.1. ReactJs**

#### **2.1.1. Giới thiệu**

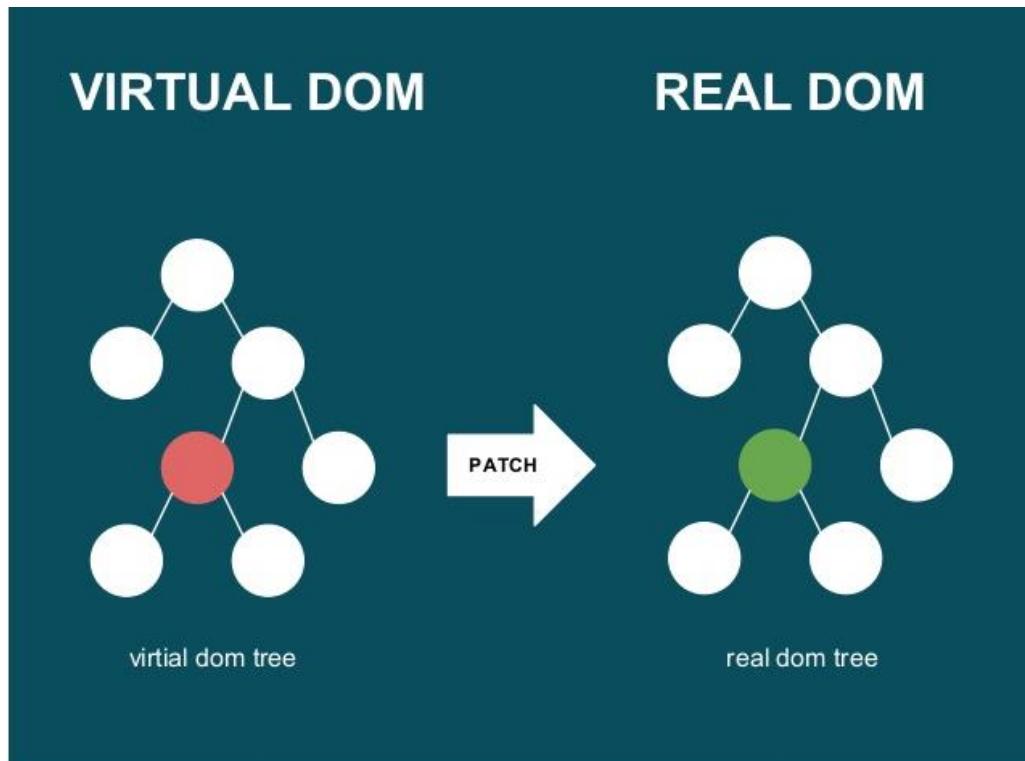
**React** (hay **ReactJs**, **React.js**) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi **Jordan Walker**, một kỹ sư phần mềm tại **Facebook**. Người bị ảnh hưởng bởi **XHP** ( Một nền tảng thành phần HTML cho PHP ). ReactJs lần đầu tiên được triển khai cho ứng dụng **Newsfeed** của **Facebook** năm 2011, sau đó được triển khai cho **Instagram.com** năm 2012. Nó được mở mã nguồn ( open-sourced) tại **JSCConf US** tháng 5 2013

ReactJs đang nổi lên trong những năm gần đây với xu hướng Single Page Application (**SPA**). Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code Javascript trong code HTML thông qua các attribute như ng-model, ng-repeat... thì với React là một library cho phép nhúng code HTML trong code Javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa Javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn.

#### **2.1.2. Các khái niệm chính trong React (ReactJs)**

##### **2.1.2.1. Virtual DOM**

Virtual DOM chỉ là một đại diện ảo của DOM, Virtual DOM ra đời đã tăng hiệu suất các ứng dụng ReactJS một cách đáng kể.



**Hình 2-1 Mô hình DOM trong ứng dụng ReactJs**

Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hóa việc re-render DOM tree thật.

⇒ Cách hoạt động cũng tựa như commit branch trên Github, Virtual DOM chỉ ghi nhận những thay đổi trên các component xảy ra từ đó chỉ thực việc render lại các component đó trên DOM thật.

### 2.1.2.2. JSX

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript.

Đặc điểm:

- Nhanh hơn: JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javascript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript.
- An toàn hơn. Ngược với Javascript, JSX là kiểu statically typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt.
- Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.

Như hình trên ta có thể thấy việc viết trực tiếp HTML vào function Javascript mà không có lỗi hay cần các dấu “” như xưa.

```
const CatalogNotFound = () => {
  return (
    <div>Không tìm thấy kết quả lọc </div>
  )
}
```

Hình 2-2 Ví dụ về JSX

### 2.1.2.3. Component

Component có cú pháp như một **function**.

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React

có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component.

```
const Layout = () => {
  return (
    <BrowserRouter>
      <Header />
      <div className='container'>
        <React.Fragment>
          <div className="main">
            <ARoutes />
          </div>
        </React.Fragment>
      </div>
      <Footer />
      <ProductViewModal />
      <LoginModal />
      <RegisterModal />
      <AlertMessage />
    </BrowserRouter>
  )
}

export default Layout
```

**Hình 2-3 Component**

Ta có thể trong hình thì component cha Layout chính là nơi chứa các component con khác. Và component Layout cũng chính là giao diện chính của Web. Việc thay đổi các component con chỉ ảnh hưởng đến các component con cần render.

Ví dụ như component LoginModal sẽ là Modal đăng nhập của trang Web và Modal này sẽ được hiện khi người dùng click đăng nhập. Điều này có nghĩa các component khác khi mà LoginModal xuất hiện hay ẩn đi đều không thay đổi vì chỉ có Login Modal là được thay đổi thuộc tính để render lại cho người dùng thấy.

#### 2.1.2.4. Props và State

- **Props:**

Viết tắt của **properties**, nhưng trong React lại là một khái niệm. Props là một đối tượng. Nó sẽ lưu trữ những thuộc tính, giá trị được truyền vào từ component cha hay trong hàm hay bất cứ khi nào Tag của component này được gọi.

```
function Welcome(props) {  
    return <h1>Hello, {props.name}</h1>;  
}  
  
const element = <Welcome name="ReactJS" />;  
ReactDOM.render(  
    element,  
    document.getElementById('root')  
);
```

**Hình 2-4 Ví dụ props**

Ta có thể thấy element là một biến được gán cho giá trị là một component Welcome và được truyền trong Tag với giá trị name="ReactJS".

Và trong component Welcome thì ta có thể thấy việc truy cập names thông qua props. Ở đây props được là một đối tượng nhận vào giá trị name như sau:

```
props = {  
    name: "ReactJS"  
}.
```

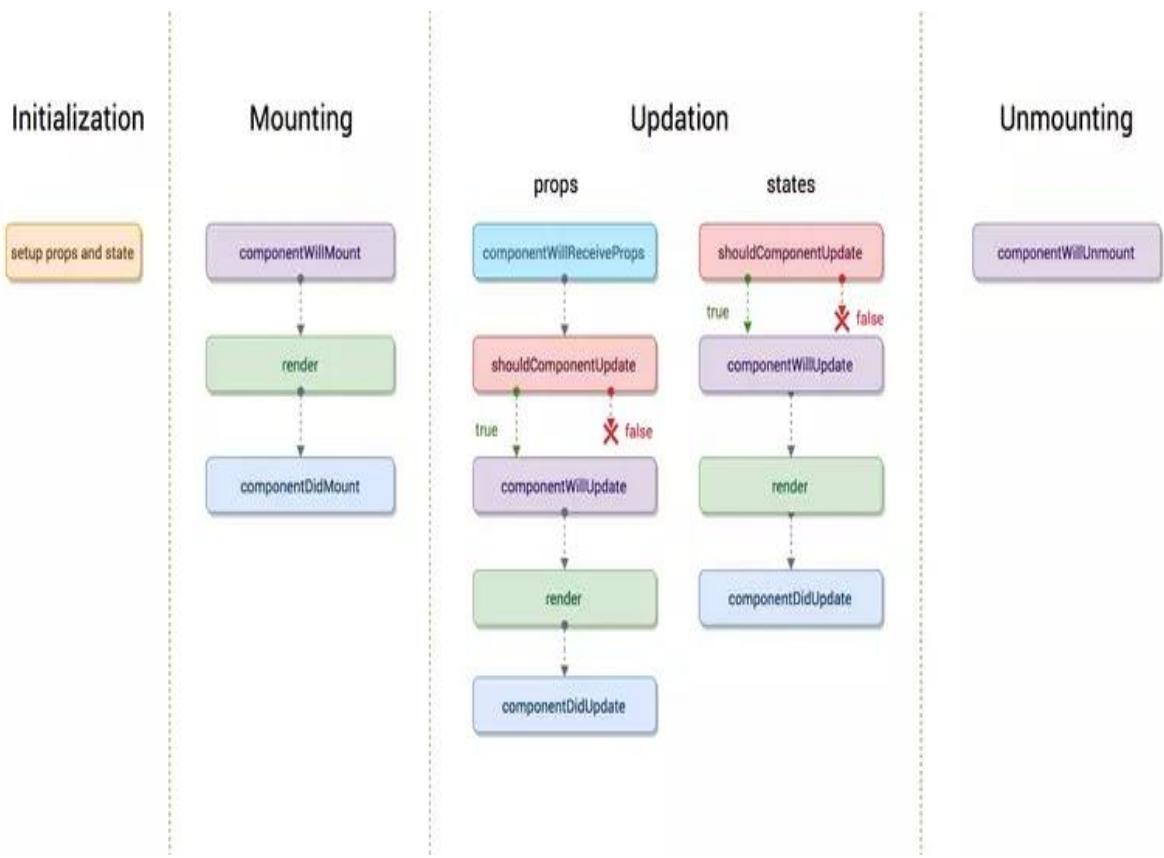
⇒ Lưu ý là chúng ta không nên thay đổi được props trong component con do ReactJS tuân theo quy tắc **pure function**: không làm thay đổi giá trị đầu vào

- **State:**

State cũng có vai trò như Props, nhưng là lưu trữ dữ liệu động của một component.

Do đó State thường được dung để theo dõi sự thay đổi bên trong component và render lại.

### 2.1.2.5. Component Lifecycle



**Hình 2-5 Component Lifecycle**

Trong React Component, components được khởi tạo (hiển thị ra DOM), update, và kết thúc (unmount), đó được gọi là một component life cycle.

React cho phép chúng ta tham gia vào các giai đoạn của mỗi component bằng cách sử dụng các phương thức được xây dựng sẵn trong mỗi giai đoạn đó. Khi một components được khởi chạy nó sẽ phải trải qua 4 giai đoạn chính:

- **Initialization:**

Đây là giai đoạn mà component sẽ bắt đầu khởi tạo state và props.

- **Mounting:**

Giai đoạn này thực hiện sau khi quá trình initialization (khởi tạo) được hoàn thành. Nó sẽ chuyển virtual DOM thành DOM và hiển thị lên người dùng. Component sẽ được render lần đầu tiên.

- **componentWillMount ()**

Được khởi chạy khi một component chuẩn bị được mount (tức là trước khi thực hiện render)

- **componentDidMount()**

Được gọi khi component đã được mount (render thành công).

- **Updating:**

Khi state hoặc props của component thay đổi, các hàm trong nhóm này sẽ được thực thi và quyết định xem có cần phải render lại UI hay không. Trường hợp UI không được re-render khi hàm shouldComponentUpdate () trả về giá trị false.

- **Unmounting:**

Đây là bước cuối cùng trong mỗi component, khi tất cả các tác vụ hoàn thành và bạn tiến hành unmount DOM.

#### **2.1.2.6. React Hook**

Trước khi react Hook ra đời thì chúng ta thường khai báo một component là class component và sử dụng các phương thức trong lifecycle một cách nhập nhằng. Trong khi đó functional component thì không thể dùng state và lifecycle dù cách trình bày dễ hiểu hơn class component.

Hooks là một bổ sung mới trong React 16.8.

Hooks là những hàm cho phép bạn “kết nối” React state và lifecycle vào các functional component.

Nhóm hầu như sử dụng React Hook cho đồ án vì nó dễ dùng và dễ hiểu hơn class component xong cũng không phủ nhận class component vẫn sẽ là đối tượng được nhóm nghiên cứu tổng tương lai vì trong một số trường hợp thì class component vẫn có thể mang lại hiệu quả.

### **2.1.3. Cách hoạt động**

ReactJs là một thư viện hỗ trợ cho việc phát triển các ứng dụng Web SPA – single page application. Điều này có nghĩa là ứng dụng chỉ có một trang ( thông thường là index.html ) cho cả ứng dụng và được chia thành các thành phần – component. Mỗi component sẽ chứa các props- được truyền từ component cha và state- lưu trữ các giá trị có trong component để render lại khi có thay đổi logic hay sự kiện từ phía người dùng.

Chính vì vậy ứng dụng sẽ không cần render lại mà chỉ component thay đổi render lại, từ đó giúp tăng đáng kể hiệu suất người dùng và giúp ích trong việc phát triển do có thể tái sử dụng ở nhiều nơi trong ứng dụng.

### **2.1.4. Nhược điểm**

Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax.

Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.

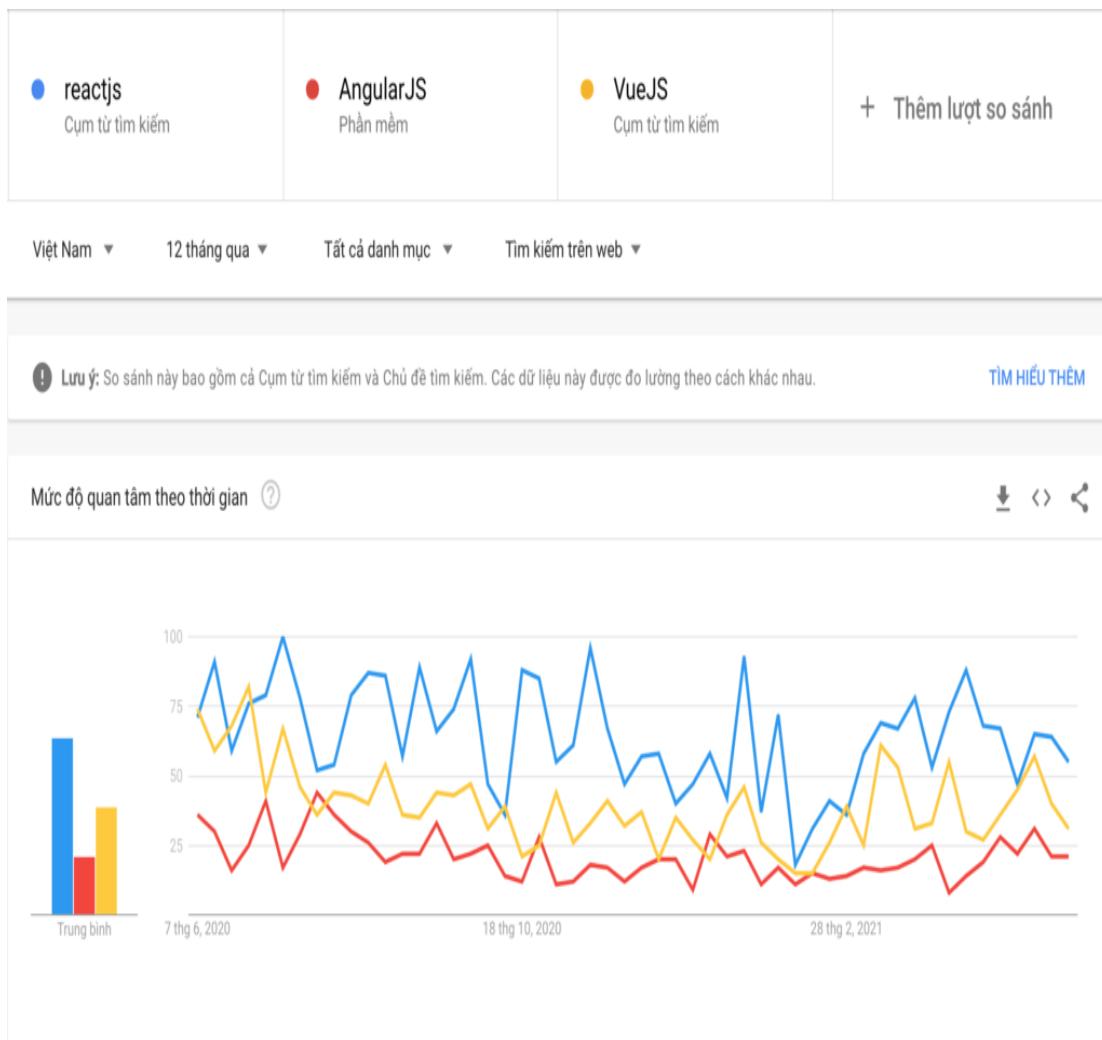
React khá nặng nề so với các framework khác React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.

Khó tiếp cận cho người mới học Web.

### 2.1.5. Ưu điểm và Tại sao nên dùng ReactJs

Ngoài việc hỗ trợ xây dựng giao diện nhanh, hạn chế lỗi trong quá trình code, cải thiện hiệu suất website thì những tính năng đặc biệt dưới đây có thể là lý do khiến bạn bắt đầu tìm hiểu nó từ bây giờ:

- Phù hợp với đa dạng thể loại website: ReactJS khiến cho việc khởi tạo website dễ dàng hơn bởi vì bạn không cần phải code nhiều như khi tạo trang web thuần chỉ dùng JavaScript, HTML.
- Tái sử dụng các Component.
- Có thể sử dụng cho cả Mobile application: React Native – một framework khác được phát triển cũng chính Facebook tương tự như ReactJs.
- Debug dễ dàng: Facebook đã phát hành một Chrome extension dùng trong việc debug trong quá trình phát triển ứng dụng.
- Cộng đồng người sử dụng lớn, các nguồn như Github đều có thể thấy các source liên quan về ReactJs. Tài liệu tham khảo phong phú và tích hợp với các Framework như Bootstrap cho ra đời React-Bootstrap.
- Hot khi tính tới 2021: Nếu bạn nhìn vào số liệu thống kê từ Google Trend ở Việt Nam ở hình bên dưới, dạo lướt qua các trang tuyển dụng hàng đầu ở Việt Nam như Topdev, Itviec,... bạn sẽ thấy số lượng tuyển dụng cho vị trí React Developer là cực kỳ lớn cùng với mức lương vô cùng hấp dẫn và độ phổ biến hiện tại của ReactJS trên thị trường Việt Nam là như thế nào.



**Hình 2-6 Dữ liệu số người dùng các Framework**

### 2.1.6. Các framework-package liên quan đến ReactJs

#### 2.1.6.1. React Hook

Khiến các component trở nên gọn nhẹ hơn.

Giảm đáng kể số lượng code, dễ tiếp cận.

Cho phép chúng ta sử dụng state ngay trong function component

Các Hooks:

### a. useState

Hàm này nhận đầu vào là giá trị khởi tạo của 1 state và trả ra 1 mảng gồm có 2 phần tử, phần tử đầu tiên là state hiện tại, phần tử thứ 2 là 1 function dùng để cập nhật state.

```
const [isLoading, setLoading] = useState(false);

onClick() {
  setLoading(true)
}
```

Hình 2-7 Ví dụ useState

### b. useEffect

Nó giúp chúng ta xử lý các side effects, useEffect sẽ tương đương với các hàm **componentDidMount**, **componentDidUpdate** và **componentWillUnmount** trong LifeCycle.

```
useEffect(
  () => {
    const subscription = props.source.subscribe();
    return () =>
      subscription.unsubscribe();
  },
  [props.source], // giá trị được subscribe
);
```

Hình 2-8 Ví dụ useEffect

useEffect có thể không cần code cleanup như hàm return.

Có 2 loại useEffect:

### c. useEffect with no dependencies

useEffect (**function ()**)

useEffect này sẽ luôn thực thi khi hàm render được gọi trong quá trình mà component tồn tại, một số trường hợp sẽ dẫn đến việc loop vô tận.

- **useEffect with dependencies**

Ở hình trên ta có thể useEffect nhận vào một dependencies là props.source điều này có nghĩa là useEffect sẽ thực thi mỗi khi props.source thay đổi giá trị bao gồm lúc component mount (props.source được khởi tạo). Nếu như ta truyền vào là một state thì useEffect sẽ thực thi mỗi khi setState.

Nếu ta không truyền bất cứ giá trị nào hay state nào thì useEffect sẽ chỉ thực thi mỗi khi component được gọi (mount). Vd:

`useEffect(function(),[])` đây là useEffect with dependencies trường hợp không truyền giá trị hay state.

`useEffect(function(),[state,props,A,B])` đây là use Effect with dependencies. Trường hợp truyền nhiều giá trị hay state và sẽ căn cứ xem giá trị.state nào thay đổi mà thực thi **function()** đã được định nghĩa.

#### d. **useRef**

Dùng để lưu trữ giá trị của một biến qua các lần render. Sau mỗi lần render, giá trị của một số biến trong component sẽ quay lại giá trị ban đầu, sử dụng useRef để lưu trữ các giá trị này.

Điều này giúp ta có thể quản lý một số tag mà không cần phải gọi `document.getelement` như trong Javascript thuần.

#### 2.1.6.2. React Router

React-Router là một thư viện định tuyến (routing) tiêu chuẩn trong React. Nó giữ cho giao diện của ứng dụng đồng bộ với URL trên trình duyệt. React-Router cho phép bạn định tuyến "luồng dữ liệu" (data flow) trong ứng dụng của bạn một cách rõ ràng. Nó tương đương với sự khảng định, nếu bạn có URL này, nó sẽ tương đương với Route này, và giao diện tương ứng.

Phiên bản hiện tại nhóm đang dùng là react-router v6.

## a. BrowerRouter, Routes và Rotue

Một bộ cục cơ bản của react-router trong việc điều trang trong ReactJs.

Mỗi một project Reactjs chỉ nên có một Brower Router bao bọc toàn bộ chương trình.

Có thể có nhiều Routes để phân chi việc điều hướng trang trong mỗi Component.

Route chỉ có thể được bọc bởi Routes và Routes chỉ chứa các children là Route. Mặc định khi vào chương trình thì ta sẽ được đưa tới Rouote có path là “/”. Thông thường sẽ page Home/ Trang chủ

Để điều trang ta sẽ dùng thẻ <Link/> trong thư viện react-router-dom

Để điều trang ta sẽ useNavigate trong thư viện react-router-dom

Truy cập pathname bằng useLocation hay useParams để truy cập URL

```
const ARoutes = () => {
  return (
    <BrowserRouter>
      <Routes>
        <Route path='/' element={<Home />}></Route>
        <Route path='/catalog/:slug' element={<Product />}></Route>
        <Route path='/catalog' element={<Catalog />}></Route>
        <Route path='/cart' element={<Cart />}></Route>
        <Route path='/accessories' element={<Accessories />}></Route>
        <Route path='/contact' element={<Contact />}></Route>
        <Route path='/order' element={<Order />}></Route>
        <Route path='/customer' element={<CustomerInfo />}></Route>
      </Routes>
    </BrowserRouter>
  )
}

export default ARoutes
```

Hình 2-9 Thiết lập một bộ chuyển trang bằng React Router

## b. useNavigate

```
// v6
import { useNavigate } from 'react-router-dom';

function MyButton() {
  let navigate = useNavigate();
  function handleClick() {
    navigate('/home');
  };
  return <button onClick={handleClick}>Submit</button>;
};
```

**Hình 2-10** Ví dụ về cách dùng useNavigate

Khai báo biến khởi tạo bằng useNavigate trong component để có thể điều trang thông qua các pathname đã khai báo trước đó trong route.

Ở các phiên bản react-router trước, useHistory sẽ đảm nhận vai trò này.

## c. useLocation

Cách khai báo tương tự như useNavigate.

```
import React from 'react';
import { useLocation } from 'react-router-dom';

const Example = props => {
  const location = useLocation();
  console.log(location);

  // ...
};
```

**Hình 2-11** Ví dụ cách dùng useLocation

useLocation sẽ chưa thông tin về path mà bạn đang ở. Nếu bạn đang ở trang admin và trước đó trong route bạn khai báo path của route chưa trang admin là path="/admin" thì location.pathname = "/admin".

#### d. useParams

```
1 import * as React from 'react';
2 import { Routes, Route, useParams } from 'react-router-dom';
3
4 function ProfilePage() {
5   // Get the userId param from the URL.
6   let { userId } = useParams();
7   // ...
8 }
9
10 function App() {
11   return (
12     <Routes>
13       <Route path="users">
14         <Route path=":userId" element={<ProfilePage />} />
15         <Route path="me" element={...} />
16       </Route>
17     </Routes>
18   );
19 }
```

**Hình 2-12 Ví dụ cách lấy params**

useParam cho phép bạn lấy các param trên url website.

Các param này được truyền vào bằng cách thêm vào path trong route theo sau các dấu :;?,....

#### e. Link

Thẻ <Link/> tương tự như thẻ <a/> trong HTML và chức năng như useNavigate nhưng <link/> sẽ được render ra như một component, element UI trên website để người dùng tương tác như các click, ...

```

1 import * as React from "react";
2 import { Link } from "react-router-dom";
3
4 function UsersIndexPage({ users }) {
5   return (
6     <div>
7       <h1>Users</h1>
8       <ul>
9         {users.map((user) => (
10           <li key={user.id}>
11             <Link to={user.id}>{user.name}</Link>
12           </li>
13         ))}
14       </ul>
15     </div>
16   );
17 }

```

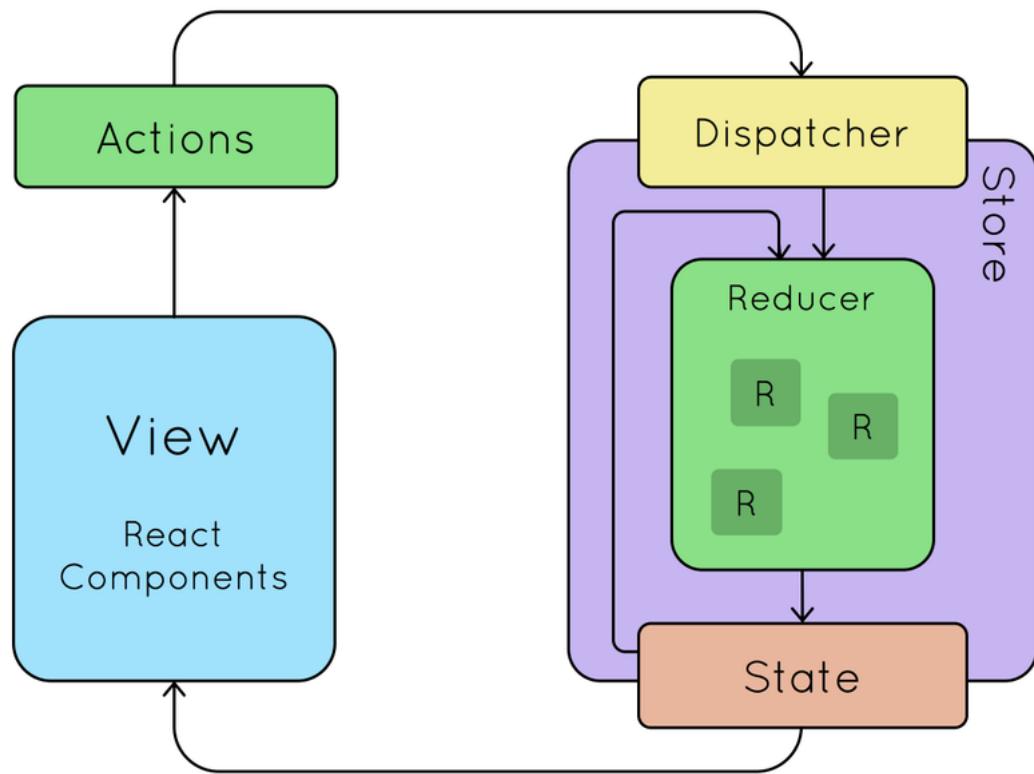
**Hình 2-13 Cách dùng thẻ Link**

### 2.1.6.3. Redux Toolkit

#### a. React Redux - Redux toolkit

- **Redux** là một thư viện Javascript giúp tạo ra thành một lớp quản lý trạng thái của ứng dụng. Tránh trường hợp phải truyền các props từ component cha sang component con và sâu hơn nữa. Vì có những props mà component con này không dùng tới nhưng vẫn phải nhận và truyền lại cho component con sau. Vì thế Redux ra đời để giải quyết vấn đề này.
- **Redux toolkit:** là một package được sinh ra để tiết kiệm thời gian trong việc cấu hình store, các file action hay reducer trong redux khá tách biệt và code lặp lại khá nhiều. Tuy nhiên, đến với redux toolkit, việc config store, action, reducer trở nên dễ dàng, nhanh chóng và ngắn gọn.

Các khái niệm: store, action, reducer và nguyên lý hoạt động



**Hình 2-14 Mô hình Redux**

- **Store** là nơi lưu trạng thái của ứng dụng và là duy nhất, có thể lưu trữ, truy xuất hoặc cập nhật giá trị state trong store thông qua các action.
- **Action**: là các event, các event này bao gồm type (để reducer biết đây là loại action gì, từ đó thực hiện các hành động cập nhật state thích hợp) và payload (chứa thông tin state mới), các payload này thông thường sẽ được cấu hình từ View hay UI tại nơi người dùng tạo ra các sự kiện như click, ...
- **Reducer**: sau khi action được gọi thông qua dispatch reducer là các pure function lấy state hiện tại, kết hợp với type, payload từ action để cập nhật giá trị state mới.

Các tính năng - tiện ích của redux được thay thế bởi redux toolkit:

```

import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";

import axios from "axios";
import { apiUrl } from "../../utils/constant";


export const getAllProduct = createAsyncThunk(
  'getAllProduct',
  async (data, { rejectWithValue }) => {
    const rs = await axios.get(` ${apiUrl}/product`)
    if (rs.status < 200 || rs.status >= 300) {
      return rejectWithValue(rs.data)
    }
    return rs.data
  }
)

export const productSlice = createSlice({
  name: 'productSlice',
  initialState: {
    value: []
  },
  reducers: {

  },
  extraReducers: (builder) => {
    builder.addCase(getAllProduct.fulfilled, (state, action) => {
      state.value = action.payload
    })
  }
})

export default productSlice.reducer

```

**Hình 2-15 Các thành phần Slice, AsyncThunk trong ReduxToolkit**

- **createSlice:** Với redux, khi định nghĩa action và reducer phải tách ra thành các file khác nhau, trong khi đó đối với slice, reducer và action được kết hợp lại trong cùng một file.
- **createAsyncThunk:** Thông thường các action, reducer không cho phép truyền vào , hay xử lý các hàm bất đồng bộ, vì

`createAsyncThunk` giúp cho việc này có thể diễn ra trong Redux. Và các xử lý này sẽ được gửi tới phần **extraReducers** trong slice để có thể cập nhật state.

- ⇒ Vì vậy nhóm đã sử dụng Redux Toolkit thay vì Redux và Redux Thunk mà vẫn đảm bảo với tính thích hợp với thiết đồ án.

## 2.2. Spring boot

### 2.2.1. Khái niệm

Spring Boot là một module của Spring Framework, cung cấp tinh năng RAD (Rapid Application Development - Phát triển ứng dụng nhanh). Spring Boot ra đời nhằm rút ngắn thời gian cài đặt và cấu hình Spring MVC Project. Giúp các lập trình viên tập trung hơn vào việc phát triển nghiệp vụ thay vì tốn nhiều thời gian cho việc cấu hình dự án. Spring Boot dễ dàng trong việc tích hợp với các hệ sinh thái của Spring như: Spring JDBC, Spring ORM, Spring Security,

### 2.2.2. Nhược điểm của Spring Boot

**Thiếu kiểm soát.** Do style cố định, Spring Boot tạo ra nhiều phụ thuộc không được sử dụng dẫn đến kích thước tệp triển khai lớn.

Quá trình chuyển đổi dự án Spring cũ hoặc hiện có thành các ứng dụng Spring Boot nhiều khó khăn và tốn thời gian.

**Không thích hợp cho các dự án quy mô lớn.** Hoạt động liên tục với các microservices, theo nhiều nhà phát triển, Spring Boot không phù hợp để xây dựng các ứng dụng nguyên khôi.

### **2.2.3. Ưu điểm của Spring Boot**

Spring Boot hỗ trợ nhúng trực tiếp các file Server như Tomcat, Jetty hoặc Undertow (Do đó, khi sử dụng Spring Boot không cần phải deploy ra file war). Điều này hỗ trợ cho lập trình viên rất nhiều trong việc phát triển ứng dụng nhanh và đặc biệt là tiết kiệm được rất nhiều thời gian trong việc deploy ứng dụng Spring.

Khác với Spring Framework đơn thuần, Spring Boot hỗ trợ việc tự động cấu hình Spring khi cần thiết. Và đặc biệt Spring Boot không sinh ra code cấu hình và cũng không cần cấu hình bằng XML. Những việc cấu hình thường rất tốn thời gian trong việc phát triển một ứng dụng. Với tiêu chí phát triển một ứng dụng nhanh,

Spring Boot đã giúp cho người dùng giảm thiểu lượng thời gian đáng kể cho việc cấu hình. Spring Boot còn được biết đến là chuẩn cho Microservices (Cloud support, giảm việc setup, config các thư viện hỗ trợ). Do đó, việc triển khai Microservices trên các ứng dụng Spring Boot là rất dễ dàng.

### **2.2.4. Các thư viện được sử dụng trong Spring Boot**

#### **2.2.4.1. Hibernate**

Khái niệm: Hibernate là một ORM (object-relationship-mapping: công nghệ cho phép chuyển đổi từ các object trong ngôn ngữ hướng đối tượng sang database quan hệ và ngược lại) framework mã nguồn mở giúp viết các ứng dụng Java có thể tham chiếu các đối tượng (entity) với bảng (table) trong hệ quản trị cơ sở dữ liệu.

Hibernate giúp thực hiện việc giao tiếp giữa tầng ứng dụng và tầng dữ liệu (kết nối, truy suất, lưu trữ...), cải thiện các vấn đề khi sử dụng JDBC như việc không cần phải lặp đi lặp lại các câu lệnh để truy xuất vào database, thay vào đó ta sẽ viết sẵn các kiểu query cần thiết và chỉ cần gọi đến khi cần truy xuất vào database.

```
@Query("SELECT new com.example.Backend.dto.ProductDto(p.id, p.title, p.id_cate, p.category$"
        "p.size, p.descriptions) FROM Product p")
List<ProductDto> getAll();

1 usage  ▾ Duy
@Query("SELECT new com.example.Backend.dto.ProductDto(p.id, p.title, p.id_cate, p.category$"
        "p.descriptions) FROM Product p WHERE p.id = ?1")
ProductDto getDetail_byID(String id);
```

### Hình 2-16 Hibernate: các hàm tương ứng sẽ có các annotation query và câu lệnh query tương ứng

Sử dụng Hibernate là độc lập với hệ quản trị cơ sở dữ liệu, nghĩa là ta không cần thay đổi câu lệnh HQL khi ta chuyển từ hệ quản trị CSDL MySQL sang Oracle, hay các hệ quản trị CSDL khác... Do đó rất dễ để ta thay đổi CSDL quan hệ, đơn giản bằng cách thay đổi thông tin cấu hình hệ quản trị CSDL trong file cấu hình.

Hibernate còn giúp cho việc mapping giữa Object Java và Table tương ứng trong database tương đối dễ dàng thông qua việc sử dụng các annotation



```
8      4 usages  ▾ Duy +1
9      @Entity
10     @Table(name = "Product")
11     public class Product {
12         3 usages
13         @Id
14         private String id;
15         3 usages
16         private String title;
17         3 usages
18         private String id_cate;
19         3 usages
20         private String categorySlug;
21         3 usages
```

**Hình 2-17** Ví dụ về việc sử dụng annotation

#### 2.2.4.2. Spring Data

Spring Data JPA là giảm thiểu việc thực hiện quá nhiều bước để có thể implement được JPA. Spring Data JPA là một phần của Spirng Data và nó hỗ trợ Hibernate 5, OpenJPA 2.4 và EclipseLink 2.6.1. Trong đồ án này, chúng em sử dụng JPA như một interface và Hibernate để thực hiện interface này.

-Cách hoạt động: Trong lập trình java, khi phải truy suất dữ liệu từ database, ta cần phải khởi tạo một class DAO (Data access object) ứng với object mà ta cần truy suất để lấy, lưu hoặc xóa dữ liệu. Vì vậy class DAO tồn rất nhiều thời gian để hoàn thiện, Spring data đơn giản hóa việc này bằng cách loại bỏ hoàn toàn việc triển khai class DAO mà chỉ sử dụng nó như một interface. Một interface DAO sẽ cần extends JPA specific Repository interface,

JpaRepository. Điều này cho phép Spring Data tìm thấy interface này và tự động tạo 1 implementation cho nó.

-Bằng cách extends interface, chúng ta nhận được các method CRUD (Create, Read, Update, Delete) phù hợp nhất để truy cập dữ liệu tiêu chuẩn có sẵn trong một DAO tiêu chuẩn.

#### **2.2.4.3. Spring Security**

-Khái niệm: Spring security là 1 framework thuộc hệ thống Spring, dành riêng cho việc thiết lập bảo mật của ứng dụng bao gồm authentication và authorization.

- Authentication: quá trình xác minh user, dựa vào thông tin đăng nhập mà user cung cấp. Ví dụ khi login, bạn nhập username và password, nó giúp hệ thống nhận ra bạn là ai.
- Authorization: Quá trình xác định xem user có quyền thực hiện những chức năng nào của hệ thống (đọc/sửa/xóa data), sau khi user đã authenticated thành công.

Hiểu nôm na thì authentication là cái công thứ nhất, xem user có thuộc hệ thống hay không, authorization là cái công thứ hai, xem user được phép làm những gì trong hệ thống đó.

#### **2.2.4.4. Swagger**

Open API Specification: là một định dạng mô tả API dành cho REST APIs. Một file OpenAPI cho phép bạn mô tả toàn bộ API bao gồm cả:

- Cho phép những endpoints (/users) và cách thức hoạt động của mỗi endpoint (GET /users, POST /users)
- Các tham số đầu vào & đầu ra của từng hoạt động
- Phương thức xác thực

- Thông tin liên lạc, chứng chỉ, điều khoản sử dụng và những thông tin khác

API specifications có thể được viết bằng YAML hoặc JSON. Định dạng này dễ đọc, dễ hiểu cho cả người dùng lẫn ngôn ngữ máy tính

Swagger là một bộ công cụ mã nguồn mở để xây dựng **OpenAPI specifications** giúp bạn có thể thiết kế, xây dựng tài liệu và sử dụng REST APIs.

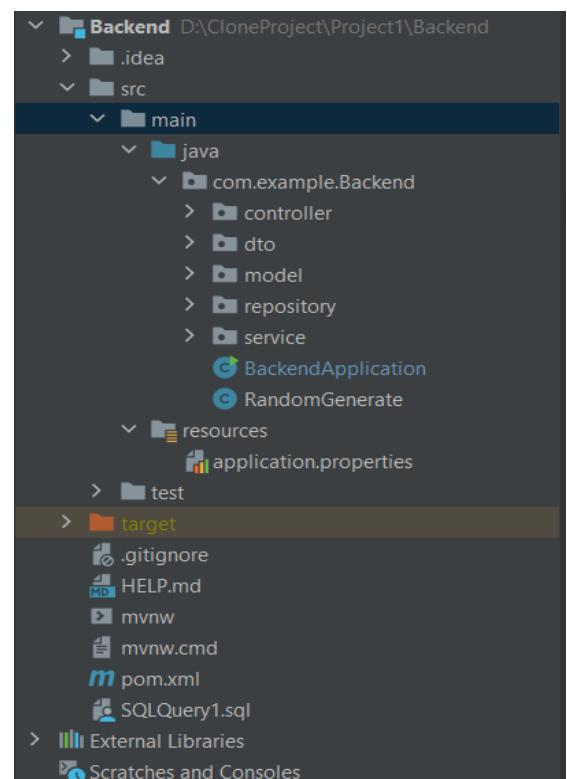
### 2.2.5. Xây dựng back-end cho đồ án bằng Spring Boot

Ở đồ án này, chúng em đã tìm hiểu và xây dựng một codebase để triển khai ứng dụng Spring Boot làm phần backend cho website của đồ án, codebase này tương đối hoàn chỉnh và đáp ứng được các nhu cầu như dễ đọc, dễ triển khai, bảo trì, nâng cấp, sửa chữa và hoàn toàn có thể được tái sử dụng cho các đồ án khác có liên quan.

Các file và tệp quan trọng của đồ án:

**Tệp model** chứa các model là các class object cần quản lý. Các object này đóng vai trò là entity có khả năng mapping đến các table bên dưới database

**Tệp dto** chứa các class dto (data transfer object) được sử dụng làm chỗ chứa tạm thời cho dữ liệu trong quá trình đọc, viết,



Hình 2-18 Cấu trúc Backend  
đồ án.

ghi, xóa dữ liệu từ database lên trên hoặc ngược lại.

**Tệp repository** chứa các repository của đồ án, các repository này kế thừa từ JPA repository

Tệp service chứa 2 phần:

- Các interface của service tương ứng với từng model mà service đó hướng đến.
- Các implement của các service interface.

**Tệp controller** chứa các class controller (các RestController trong Spring Boot) các controller này giúp điều hướng và mapping các request được gửi đến từ server

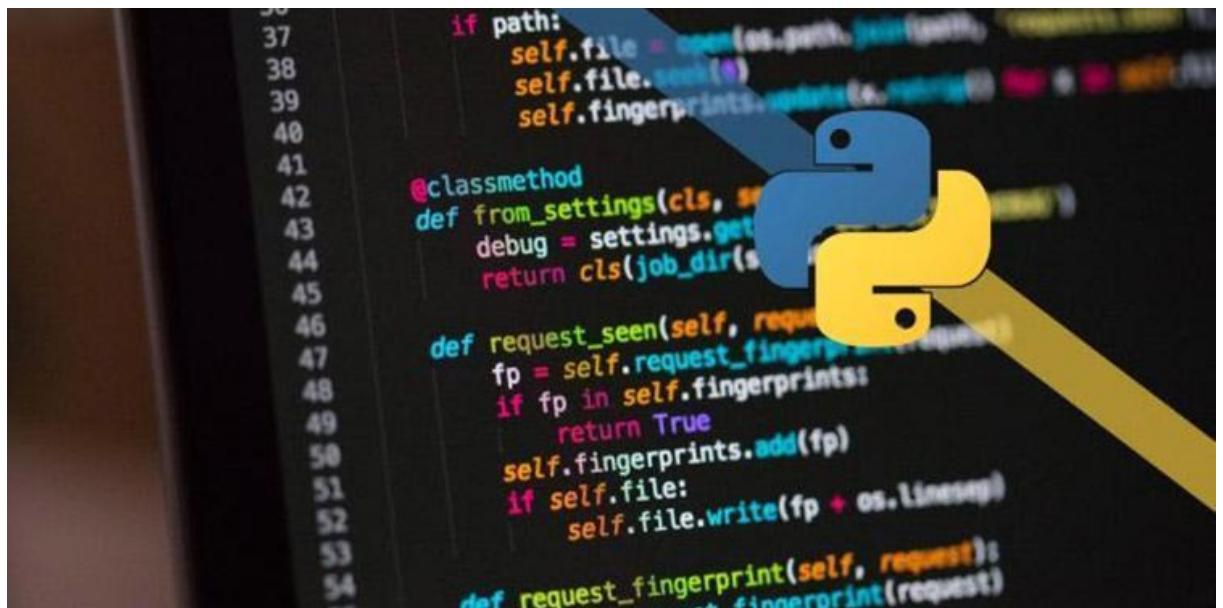
**File pom.xml** chứa các thông tin cần thiết để định nghĩa project và các plugin cũng như các dependency được khai báo và sử dụng trong project String Boot

**File BackendApplication** là file java chính để chạy project Spring Boot, file này còn có các hàm cấu hình security của project.

## 2.3. Flask

### 2.3.1. Tổng quan Python

#### 2.3.1.1. Python là gì



Hình 2-19 Python

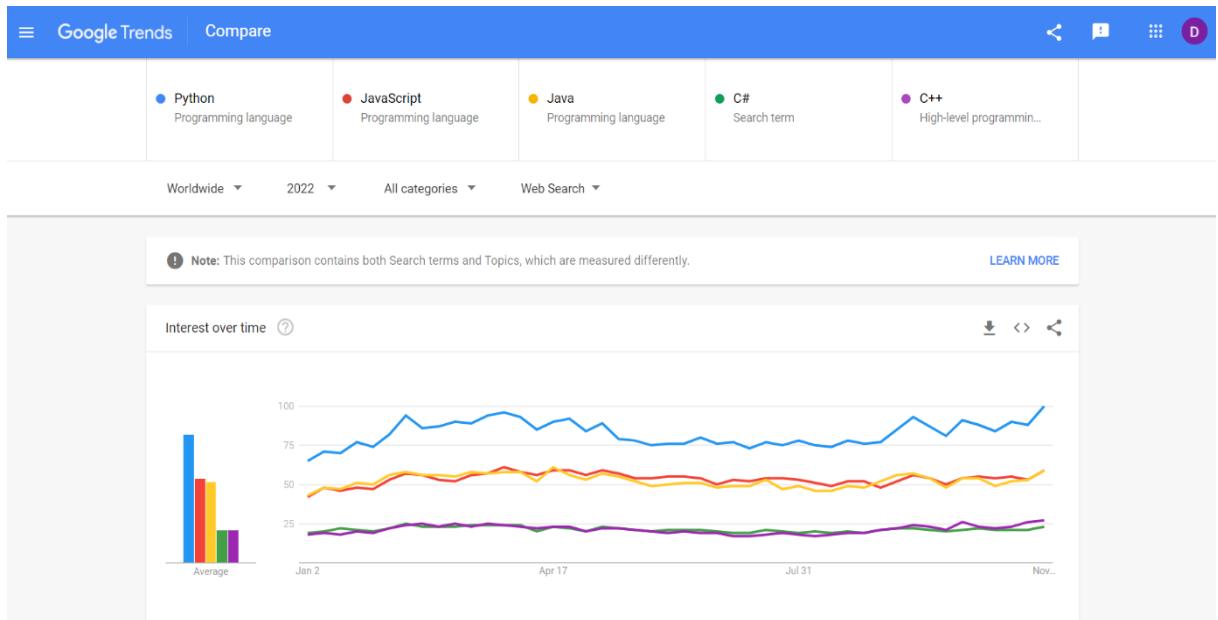
Python là một ngôn ngữ lập trình kịch bản (scripting language) do **Guido van Rossum** tạo ra năm 1990.

Là ngôn ngữ lập trình bậc cao, phục vụ cho các mục đích lập trình đa năng. Ưu điểm nổi bật nhất đó chính là dễ đọc, dễ nhớ, dễ học. Python là ngôn ngữ có cấu trúc tương đối rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của python còn cho phép người dùng sử dụng để viết mã lệnh với số lần gõ phím tối thiểu.

Ngôn ngữ python được thiết kế hướng tới đối tượng với cấu trúc hàng và cách xử lý dữ liệu đơn giản, dễ đọc. Nó sẽ giúp người dùng tạo ra những chương trình hay với số lượng dòng code ít nhất.

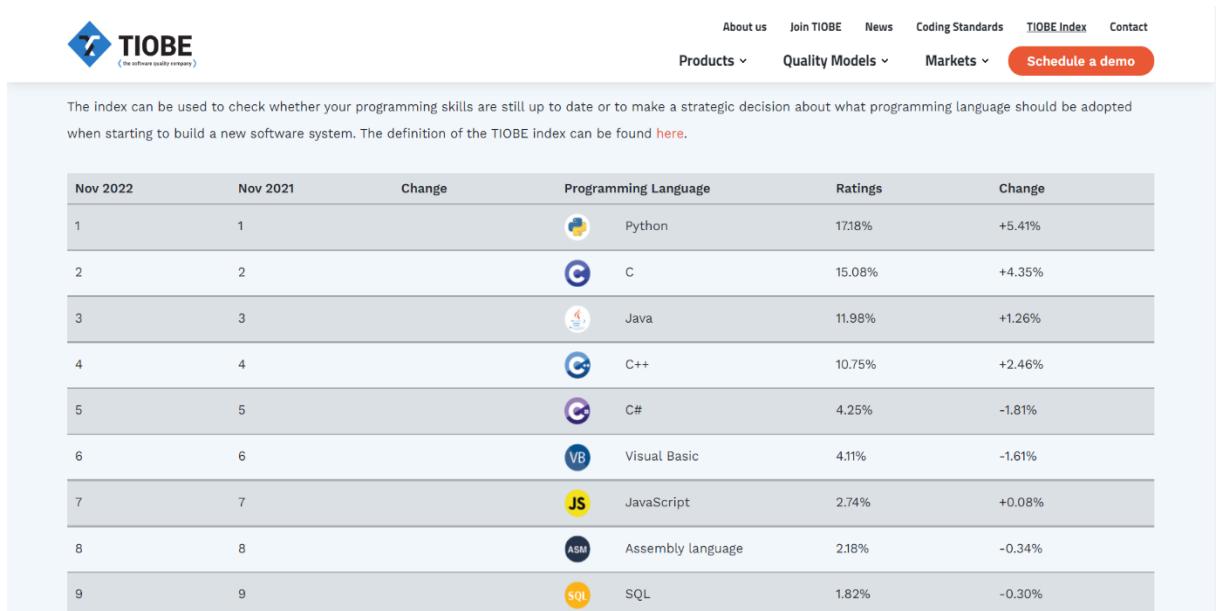
## 2.3.2. Độ phổ biến trong năm 2022

### 2.3.2.1. Google Trend



Hình 2-20 Python Google Trend

### 2.3.2.2. TIOBE



Hình 2-21 Python TIOBE

## 2.3.3. Ứng dụng của Python



Hình 2-23 Tổng quan ứng dụng Python

### 2.3.3.1. Phát triển Web



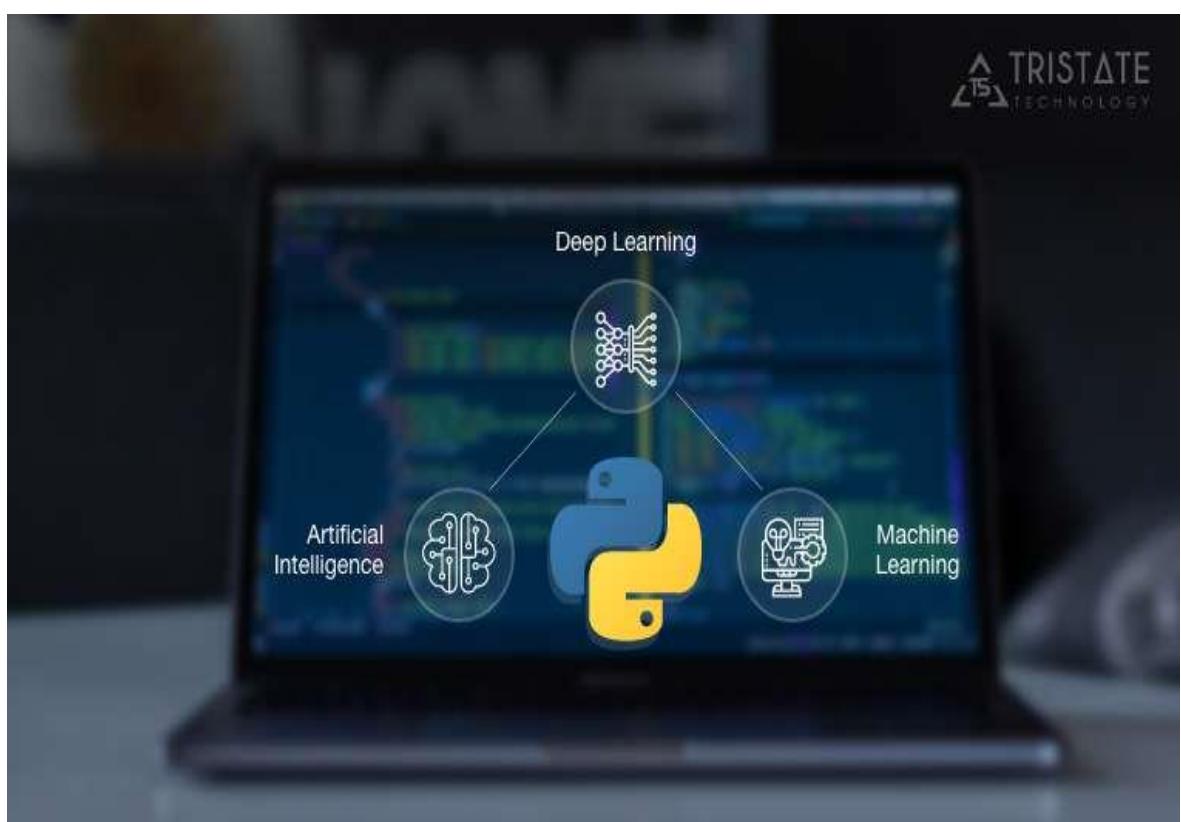
Hình 2-22 Thư viện Python hỗ trợ Web

Python được sử dụng để phát triển web. Nhiều trang web bạn đang sử dụng hàng ngày được xây dựng bằng Python. Một số framework nổi tiếng nhất

của Python là Django, Flask, Pyramid. Python được sử dụng để quản lý cơ sở dữ liệu, tài khoản người dùng trên các trang.

Trong đó Flask là một Web Framework rất nhẹ của Python, dễ dàng giúp người mới bắt đầu học Python có thể tạo ra website nhỏ. Flask cũng dễ mở rộng để xây dựng các ứng dụng web phức tạp. Trong đồ án này, Flask đóng vai trò là API cho việc chạy hệ thống khuyến nghị phim (recommendation system)

### 2.3.3.2. Machine / Deep Learning và AI



**Hình 2-24 Python và Machine Learning**

Các lập trình viên sử dụng python cho các dự Học máy và Trí tuệ nhân tạo. Ứng dụng của Python sở hữu các thư viện mạnh mẽ như scikit-learning và TensorFlow giúp việc triển khai các thuật toán Machine learning trở nên dễ dàng hơn. Đồng thời, Python sở hữu nhiều thư viện chuyên biệt giúp người học dễ dàng thực hiện nhiều tác vụ học máy khác nhau từ đơn giản đến phức tạp.

### 2.3.4. Ưu điểm

- ⊕ Đơn giản: Cú pháp đơn giản giúp cho người lập trình dễ dàng đọc và tìm hiểu.
- ⊕ Tốc độ: Python có tốc độ xử lý nhanh hơn so với ngôn ngữ PHP
- ⊕ Tương tác: Chế độ tương tác cho phép người lập trình thử nghiệm tương tác sửa lỗi của các đoạn mã.
- ⊕ Chất lượng: Thư viện có tiêu chuẩn cao, Python có khôi cơ sở dữ liệu khá lớn nhằm cung cấp giao diện cho tất cả các CSDL thương mại lớn.
- ⊕ Thuận tiện: Python được biên dịch và chạy trên tất cả các nền tảng lớn hiện nay.
- ⊕ Mở rộng: Với tính năng này, Python cho phép người lập trình có thể thêm hoặc tùy chỉnh các công cụ nhằm tối đa hiệu quả có thể đạt được trong công việc.
- ⊕ GUI Programming: Giúp cho việc thực hiện ảnh minh họa di động một cách tự nhiên và sống động.

### 2.3.5. Nhược điểm

Tất cả chúng ta đều biết không có một sản phẩm nào là hoàn hảo tuyệt đối 100%, nó luôn luôn tồn tại những nhược điểm và Python cũng không ngoại lệ. Mặc dù là một **ngôn ngữ lập trình** được yêu thích nhưng nó vẫn có những mặt hạn chế nhất định như sau:

- ⊕ Python không có các thuộc tính như: protected, private hay public, không có vòng lặp *do...while* và *switch.... case*.
- ⊕ Python mặc dù nhanh hơn so với PHP, nhưng lại không nhanh hơn so với C++, Java.

### 2.3.6. Flask framework

#### 2.3.6.1. Flask là gì?

Flask là một framework ứng dụng web được viết bằng Python. Nó được phát triển bởi Armin Ronacher, người dẫn đầu một nhóm những người đam mê Python quốc tế có tên là Pococco. Flask dựa trên bộ công cụ Werkzeug WSGI và template engine Jinja2.

## **a. Werkzeug và WSGI**

- **WSGI**

Giao diện cổng máy chủ web (Web Server Gateway Interface, WSGI) đã được sử dụng như một tiêu chuẩn để phát triển ứng dụng web Python. WSGI là đặc điểm kỹ thuật của giao diện chung giữa máy chủ web và ứng dụng web.

- **Werkzeug**

Werkzeug là một bộ công cụ WSGI thực hiện các yêu cầu, đối tượng phản hồi và các chức năng tiện ích. Điều này cho phép một web framework được xây dựng trên đó. Flask framework sử dụng Werkzeug làm một trong những cơ sở của nó.

## **b. Jinja2**

Jinja2 là một templates engine. Nó giống như là Razor view trong ASP.NET MVC hay Twig, Smarty trong PHP. Là một extensions hỗ trợ cho việc show dữ liệu từ app ra html. Jinja2 là 1 extension được kèm theo Flask nên sau khi cài đặt Flask, bạn có thể sử dụng luôn Jinja2 mà không cần phải cài thêm.

Ví dụ:

*Flask:*

```
# Khai báo 1 đường dẫn
@app.route("/")
def hello():
    # Function này sẽ render page sample.html và kèm theo là 2 dữ liệu foo, bar (2 dữ
    liệu này sẽ show ra html)
    return render_template("sample.html", foo=foo_string, bar=bar_array)
```

*HTML: sample với Foo là string và Bar là array*

```
<div>
    <!-- Sử dụng cặp dấu '{{' và '}}' để truy cập 1 biến đơn --&gt;
    Foo value:
    &lt;p&gt;
        {{ foo_string }}
    &lt;/p&gt;
    <!-- Sử dụng cặp dấu '{%' và '%}' để viết 1 đoạn code python --&gt;
    {% for bar in bar_array %}
        &lt;p&gt;
            0   {{ bar }}
            1   &lt;/p&gt;
            2   {% endfor %}
            3 &lt;/div&gt;</pre>
```

## Hình 2-25 Jinja2 template

### 2.3.6.2. Cấu trúc Flask project

Một ứng dụng sẽ được phân vào gói tương ứng. Trong Python, một thư mục con có chứa file `__init__.py` sẽ được xem như là một gói và có thể được import vào các chương trình khác. Khi bạn import một gói, file `__init__.py` sẽ được thực thi và định nghĩa các chức năng mà một gói cho phép các ứng dụng khác có thể sử dụng.

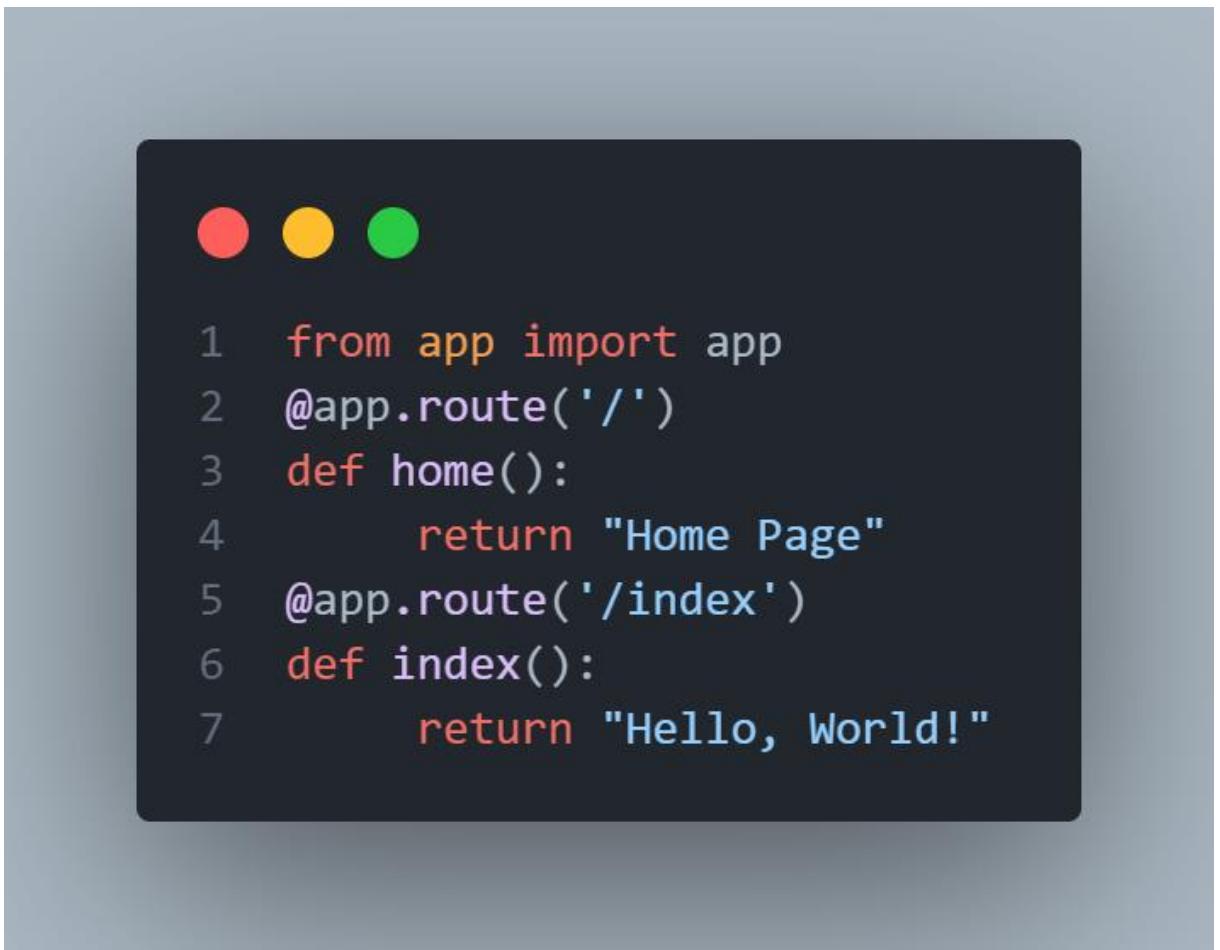
```
/home/user/Projects/flask-tutorial
├── flaskr/
│   ├── __init__.py
│   ├── db.py
│   ├── schema.sql
│   ├── auth.py
│   ├── blog.py
│   └── templates/
│       ├── base.html
│       ├── auth/
│       │   ├── login.html
│       │   └── register.html
│       └── blog/
│           ├── create.html
│           ├── index.html
│           └── update.html
└── static/
    └── style.css
tests/
├── conftest.py
├── data.sql
├── test_factory.py
├── test_db.py
├── test_auth.py
└── test_blog.py
venv/
setup.py
MANIFEST.in
```

**Hình 2-26 Cấu trúc Flask project**

- Flaskr: bao gồm code và files của project
- Tests: bao gồm các files test
- Venv: Môi trường và các dependencies
- .py: các files code python
- Templates: bao gồm các files HTML, thông thường Flask sẽ tự tìm folder mang tên này để thực thi khi dùng hàm render\_templates()
- Static: gồm các file CSS và Javascript

### 2.3.6.3. Routing

Được dùng để di chuyển tới các pages thông qua URL



The screenshot shows a presentation slide with a dark background. At the top left are three colored circles: red, yellow, and green. Below them is a code block in Python:

```
1 from app import app
2 @app.route('/')
3 def home():
4     return "Home Page"
5 @app.route('/index')
6 def index():
7     return "Hello, World!"
```

Hình 2-27 Flask Routing

### 2.3.6.4. Static

Các ứng dụng web động cũng cần các tệp tĩnh. Đó thường là nơi chứa các tệp CSS và JavaScript , ảnh,....

Chỉ cần tạo một thư mục có tên “Static” trong package.

Dạng “/static”

```
url_for('static', filename='style.css')
```

**Hình 2-28 Flask static**

Sử dụng hàm url\_for để truy cập đến file css/javascript nằm trong folder đó

### 2.3.6.5. SECRET\_KEY

Đối với bất kỳ ứng dụng nào ngoại trừ các ứng dụng rất đơn giản, chúng ta sẽ thấy rằng Flask (và có thể bao gồm các thư viện mở rộng của Flask mà bạn dùng) cho phép chúng ta tự do trong việc thực hiện ở mức độ nhất định.

Có một vài định dạng khác nhau trong việc cung cấp cấu hình cho ứng dụng. Giải pháp cơ bản nhất là định nghĩa các tham số cấu hình của bạn qua các khóa (key) trong *app.config*. File này dùng kiểu dictionary để mô tả các tham số.

```
index.html | form.py | config.py | style.css | routes.py  
app > __init__.py > ...  
1  from flask import Flask  
2  from config import Config  
3  
4  app = Flask(__name__)  
5  app.config.from_object(Config)  
6  from app import routes  
7  
8
```

**Hình 2-29 Flask secret key**

Nhưng thay vì đặt tất cả các thông số cấu hình vào trong mã khởi tạo của ứng dụng, chúng ta sẽ đặt nó vào một file cấu hình riêng rẽ.

Chúng ta sẽ theo quy ước là dùng một class để chứa tất cả các thông số cấu hình của ứng dụng. Để giữ cho cấu trúc chung của ứng dụng được gọn gàng, chúng ta sẽ tạo ra lớp cấu hình trong một module Python riêng biệt. Sau đây là các khai báo trong lớp cấu hình cho ứng dụng trong file **config.py** mà chúng ta sẽ lưu ở thư mục ngoài cùng của ứng dụng

Tham số cấu hình **SECRET\_KEY** trong lớp Config là một phần rất quan trọng trong các ứng dụng Flask. Flask và một số cá thư viện mở rộng dùng giá trị của biến này với vai trò là khóa mã hóa để tạo ra các chữ ký hoặc các token bảo mật.

Giá trị của khóa bí mật này là một biểu thức với hai mệnh đề được ghép lại với nhau qua toán tử OR. Mệnh đề thứ nhất sẽ tìm giá trị của một biến môi trường cũng gọi là **SECRET\_KEY**. Mệnh đề thứ hai là một chuỗi được định nghĩa sẵn. Chúng ta sẽ dùng cách này thường xuyên để thiết lập các tham số cấu hình xuyên suốt loạt bài này. Lý do là nó cho phép chúng ta có thể sử dụng giá trị của khóa từ biến môi trường hoặc một chuỗi định sẵn một cách linh hoạt. Trong môi trường phát triển, ít có nguy cơ bảo mật thì chúng ta có thể không cần khai báo biến môi trường và do đó, ứng dụng sẽ sử dụng giá trị của chuỗi định sẵn. Tuy nhiên, khi sử dụng chính thức thì chúng ta sẽ thiết lập một giá trị riêng cho khóa này trong biến môi trường. Giá trị này là duy nhất và khó có thể đoán ra bởi các phần mềm hack, vì thế chúng ta có thể bảo đảm rằng những dữ liệu nhạy cảm không bị lộ.

### 2.3.7. Flask API

#### 2.3.7.1. Giới thiệu

Flask đã quá quen thuộc đối với những ai làm web, app. Đây là một framework hỗ trợ xây dựng các API trên nền tảng python.

Flask là framework không quá nhanh. Nếu bạn làm việc nhiều với server, có nhiều ngôn ngữ thích hợp hơn mà các chuyên gia hay dùng đó là vuejs, golang, erlang. Nhưng trên ngôn ngữ python thì mình nghĩ Flask là đủ. Ngoài ra Flask cũng có ưu điểm đó là tính bảo mật của nó rất cao, code đẹp và dễ hiểu.

Về định nghĩa của API thì các bạn biết nó là gì rồi chứ? Nếu bạn chưa biết thì mình giải thích đơn giản như sau:

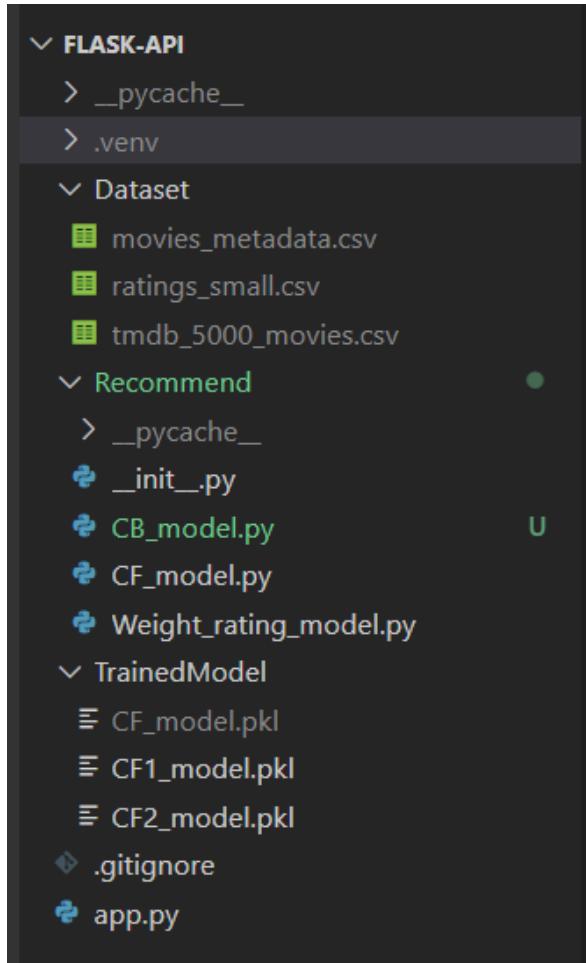
API được coi như chất liệu để làm nên một website. Khi chúng ta tương tác với một ứng dụng là tương tác với dữ liệu của ứng dụng đó được render trong một template html. Dữ liệu đó đến từ đâu? Nó được truyền đến thiết bị của bạn thông qua chính API. API là chiếc cầu nối dữ liệu qua lại giữa client và server và thay đổi những gì mà chúng ta nhìn thấy trên front end. Các phương thức tương tác dữ liệu trên API chính bao gồm: **GET, POST, PUT, PATCH, DELETE**. Đôi với project này chỉ cần hiểu **GET, POST** là đủ.

**GET:** Client nhận dữ liệu từ server.

**POST:** Client gửi dữ liệu lên server. Server xử lý dữ liệu và trả về một kết quả.

#### 2.3.7.2. Cấu trúc

Cấu trúc chung của Flask API:



File .venv chứa các thư viện và môi trường chạy python cho Flask

Thư mục Dataset chứa các dataset dùng cho việc training model

- Movies\_metadata.csv: file csv chứa dữ liệu của phim dưới dạng metadata
  - Rating\_small.csv: file csv chứa dữ liệu rating của khách hàng đối với các phim trong file Movies\_metadata
  - Tmdb\_5000\_movies: file csv chứa dữ liệu của phim dưới dạng bảng
- Thư mục Recommend chứa các model của recommendation system

### 2.3.7.3. Khởi tạo

## 2.4. Machine Learning

Hệ thống khuyến nghị sản phẩm (recommendation system) là một trong những ứng dụng phổ biến nhất của khoa học dữ liệu ngày nay. Chúng được sử

dụng để dự đoán "rating" hoặc "preference" mà người dùng sẽ dành cho một mặt hàng.

VD: Netflix sử dụng gợi ý phim cho người dùng dựa trên các đánh giá và xu hướng coi phim của người dùng trong một khu vực lớn.

hệ thống Recommender System có thể được chia thành 4 loại chính:

- Simple Recommenders: hệ thống này đưa ra các đề xuất tổng quát cho người dùng, dựa trên một số tiêu chí đánh giá của mặt hàng đó. Đối với phim ảnh, Simple Recommenders dựa trên các yếu tố như độ phổ biến, thể loại để đưa ra gợi ý cho người dùng
- Content-based Recommenders: hệ thống này đề xuất các mặt hàng tương tự dựa trên một mặt hàng cụ thể. Hệ thống này sử dụng siêu dữ liệu mục, chẳng hạn như thể loại, đạo diễn, mô tả, diễn viên, ... cho phim, để đưa ra các đề xuất này. Ý tưởng chung rằng sau các hệ thống giới thiệu này là nếu một người thích một mặt hàng cụ thể, họ cũng sẽ thích một mặt hàng tương tự với nó. Và để khuyến nghị điều đó, nó sẽ sử dụng siêu dữ liệu mục trước đây của người dùng. Một ví dụ điển hình có thể là YouTube, nơi dựa trên lịch sử của bạn, nó gợi ý cho bạn những video mới mà bạn có thể xem.
- Collaborative filtering Recommenders: hệ thống này được sử dụng rộng rãi và chúng cố gắng dự đoán "ratings" hoặc "preference" mà người dùng sẽ đưa ra một mặt hàng dựa trên xếp hạng trước đây và sở thích của những người dùng khác. Bộ lọc cộng tác không yêu cầu siêu dữ liệu mục giống như các bộ lọc dựa trên nội dung của nó.
- Hybrid Recommenders: Hybrid Filtering là sự kết hợp của hai giải thuật Content-based Filtering và Collaborative Filtering: Hybrid Filtering được sử dụng mềm dẻo khi hệ thống Collaborative Filtering không có các hành vi (ratings), khi đó hệ thống sẽ sử dụng Content-based Filtering và ngược lại, khi Content-based Filtering không có các feature cần thiết trong việc đánh giá thì hệ thống sẽ sử dụng Collaborative Filtering để thay thế.

#### 2.4.1. User mới

Đối với khách hàng mới, họ chưa có lịch sử mua hàng cũng như đánh giá của họ với nhiều bộ phim, vì vậy việc áp dụng thuật toán content-based, và collaboration filtering bị loại bỏ do thiếu dữ liệu và việc áp dụng 2 thuật toán

trên làm tăng thời gian xử lý của hệ thống một cách thừa thãi. Vì vậy ta sẽ áp dụng simple recommender cho trường hợp này:

Để xây dựng hệ thống này, ta cần:

- Thước đo tổng quát để tính toán dữ liệu đánh giá phim của người dùng
- Tính toán mức độ đánh giá cho tất cả các phim hiện có
- Sắp xếp và khuyến nghị các phim thuộc top đầu dựa theo đánh giá đã tính toán

Sử dụng thuật toán weighted rating làm thước đo đánh giá cho từng bộ phim:

$$\text{Weighted Rating (WR)} = \left( \frac{v}{v+m} \cdot R \right) + \left( \frac{m}{v+m} \cdot C \right)$$

Trong đó:

- v là số lượng vote của phim
- m là số lượng vote tối thiểu để được đưa vào đồ thị, m được tính toán dựa vào việc 1 bộ phim muốn được đưa vào đồ thị cần phải có lượt vote lớn hơn 90% các bộ phim còn lại
- R đánh giá trung bình của phim
- C là đánh giá trung bình trên toàn bộ dữ liệu

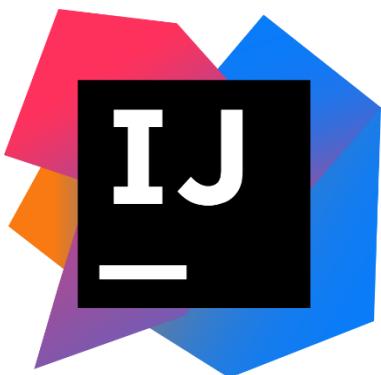
## 2.4.2. User đã có rates

Ở loại user này hướng tiếp cận cho model recommendation sẽ là 2 model Content-based và Collaborative Filtering.

### 2.4.2.1. Content – based

## 2.5. Frameworks và tools khác

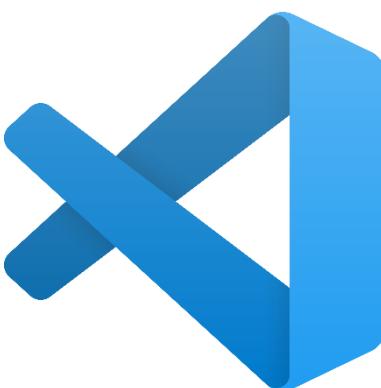
### 2.5.1. IntelliJ



Hình 2-30 IntelliJ

Được phát hành lần đầu tiên vào năm 2001, IntelliJ IDEA là một môi trường phát triển tích hợp cho Windows, macOS và Linux. Nó chủ yếu được sử dụng để phát triển phần mềm bằng ngôn ngữ lập trình Java, nhưng cũng hỗ trợ các ngôn ngữ khác hoặc tự nhiên hoặc sử dụng một plugin. Nó cũng là cơ sở cho Android Studio của Google, môi trường phát triển chính thức để tạo các ứng dụng Android. Nhóm sử dụng cho việc code Back-end.

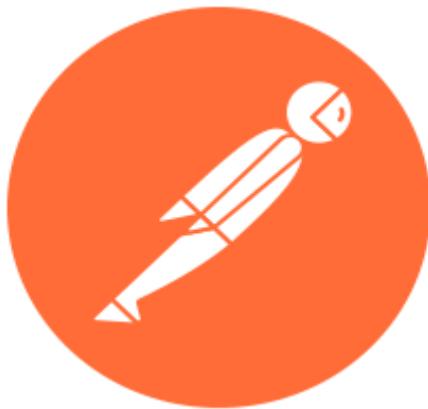
### 2.5.2. Visual Studio Code



Hình 2-31 Víual  
Studio Code

Visual Studio Code là một trình soạn thảo mã nguồn được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có chức năng nổi bật cú pháp (syntax highlighting), smart code, snippets, và cài tiến mã nguồn. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và các tùy chọn khác. Nó miễn phí và là phần mềm mã nguồn mở theo giấy phép MIT. Nhóm sử dụng cho việc code Front-end.

### 2.5.3. Postman



Hình 2-32 Postman

Postman là một công cụ cho phép chúng ta thao tác với API, phổ biến nhất là REST.

Postman hiện là một trong những công cụ phổ biến nhất được sử dụng trong thử nghiệm các API. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào.

Postman hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE, ...). Bên

cạnh đó, nó còn cho phép lưu lại lịch sử các lần request, rất tiện cho việc sử dụng lại khi cần.

Nhóm sử dụng Postman cho việc test API.

### 2.5.4. Microsoft SQL Server



Hình 2-33 SQL Server

SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, database engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

Các thành phần cơ bản trong SQL Server gồm có: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service, ... Tất cả kết hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.

Nhóm sử dụng CSDL SQL Server làm Database cho ứng dụng.

### **2.5.5. Github**



**Hình 2-34 Github**

GitHub là một hệ thống quản lý dự án và phiên bản code, hoạt động giống như một mạng xã hội cho lập trình viên. Các lập trình viên có thể clone lại mã nguồn từ một repository và Github chính là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các kho chứa của riêng mình để có thể làm việc.

GitHub được coi là một mạng xã hội dành cho lập trình viên lớn nhất và dễ dùng nhất với các tính năng cốt lõi như:

Wiki, issue, thống kê, đổi tên project, project được đặt vào namespace là user.

Watch project: theo dõi hoạt động của project của người khác. Xem quá trình người ta phát triển phần mềm thế nào, project phát triển ra sao

Follow user: theo dõi hoạt động của người khác.

Github giúp ta quản lý source code dễ dàng, tracking sự thay đổi version.

### **2.5.6. Ngôn ngữ Java**



Java là ngôn ngữ lập trình hướng đối tượng được sử dụng các cú pháp C và C++, thường được dùng trong phát triển phần mềm, trang web, game hay ứng dụng

**Hình 2-35 Java**

Java được phát triển bởi James Gosling và đồng nghiệp ở Sun MicroSystem năm 1991 và được phát hành 1994 đến năm 2010 được Oracle mua lại.

Các phiên bản của Java:

- Java SE: Là nền tảng cơ bản phát triển giao diện ứng dụng Winform.
- Java EE: Dựa trên SE nhưng dùng để phát triển web.
- Java ME: Phát triển dành cho mobile.

### 2.5.7. Ngôn ngữ thiết kế Web HTML-CSS-Javascript

Đây là bộ ba ngôn ngữ nền tảng cơ bản cho việc thiết kế Web từ lâu đời, ReactJs chỉ là thư viện được mở rộng và đóng gói các dòng lệnh của các ngôn ngữ trên. Trong đó:



Hình 2-36 HTML-CSS-Javascript

- **HTML:** (*HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản*) là một ngôn ngữ đánh dấu được thiết kế để tạo nên các trang web với các mẫu thông tin được trình bày trên WWW. Cùng với CSS, Javascript, HTML tạo ra bộ ba nền tảng kỹ thuật cho các Website.

HTML là cốt lõi của mọi trang web. Bất kể sự phức tạp của một trang web hoặc số lượng công nghệ liên quan. Đó là một kỹ năng thiết yếu cho bất kỳ chuyên gia web. Đó là điểm khởi đầu cho bất cứ ai học cách tạo nội dung cho web. Và thật may mắn cho những bạn mới bắt đầu là HTML rất dễ học.

- **CSS:** (*Cascading Style Sheets*). Ngôn ngữ lập trình này chỉ ra cách các yếu tố HTML của trang web thực sự sẽ xuất hiện trên giao diện của trang. Nếu HTML cung cấp các công cụ thô cần thiết để cấu trúc nội dung trên một trang web thì CSS sẽ giúp định hình kiểu nội dung này để trang web xuất hiện trước người dùng theo một cách đẹp hơn.
- **Javascript:** JavaScript là ngôn ngữ lập trình dựa trên logic. Nó có thể được sử dụng để sửa đổi nội dung trang web. JavaScript là ngôn ngữ lập trình cho phép các nhà phát triển web thiết kế các trang web tương tác.

Nhóm sử dụng để code cho Front-end bởi vì chủ yếu ReactJS sẽ dùng các cú pháp ES6 (thuộc Javascript) và các dòng HTML trong render component. Và SCSS ( là tập CSS đã được biên dịch).

### 2.5.8. Bootstrap



Hình 2-37 Bootstrap

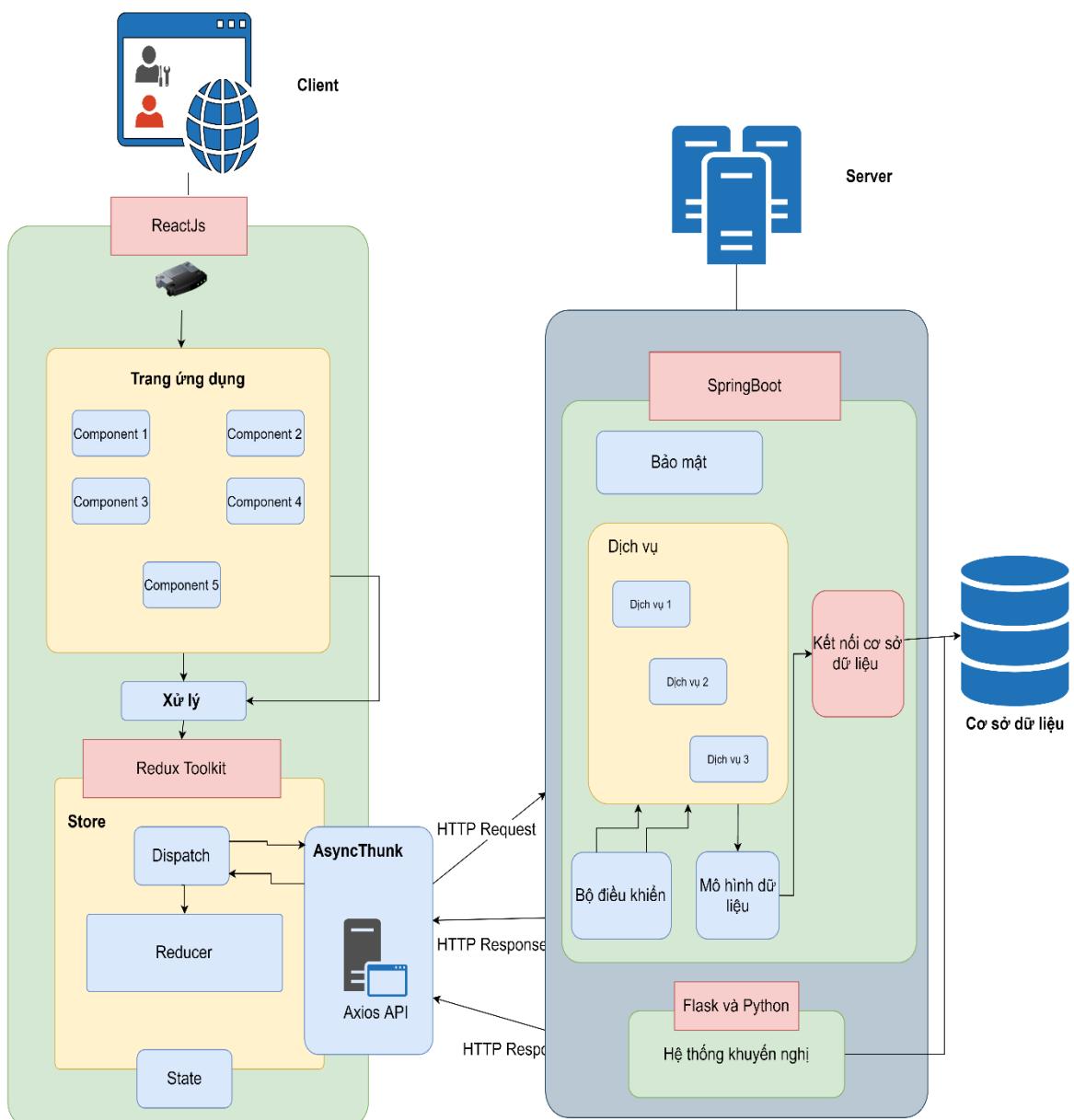
Bootstrap là một framework bao gồm các HTML, CSS và JavaScript template dùng để phát triển website chuẩn **Responsive-tương thích với mọi phiên bản nền tảng (mobile desktop hay tablet, ...)**.

Nhóm sử dụng Bootstrap 5 là một phiên bản mới của Bootstrap giúp thiết kế ứng dụng trông đẹp mắt hơn.

# Chương 3. XÂY DỰNG HỆ THỐNG

## 3.1. Xây dựng kiến trúc hệ thống

### 3.1.1. Sơ đồ kiến trúc



Hình 3-1 Sơ đồ kiến trúc hệ thống

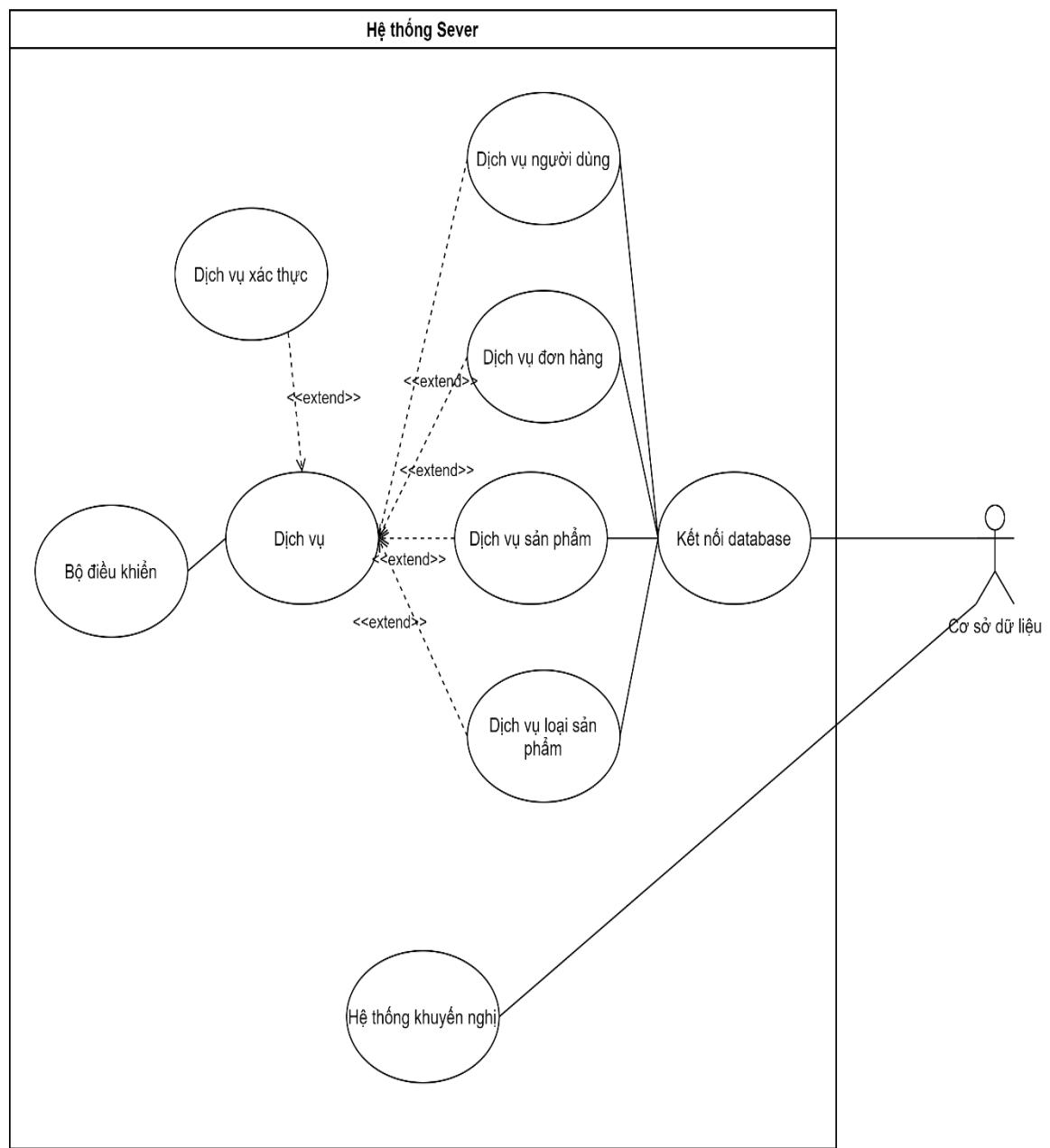
### 3.1.2. Sơ đồ use case

#### 3.1.2.1. Người dùng/ Khách hàng



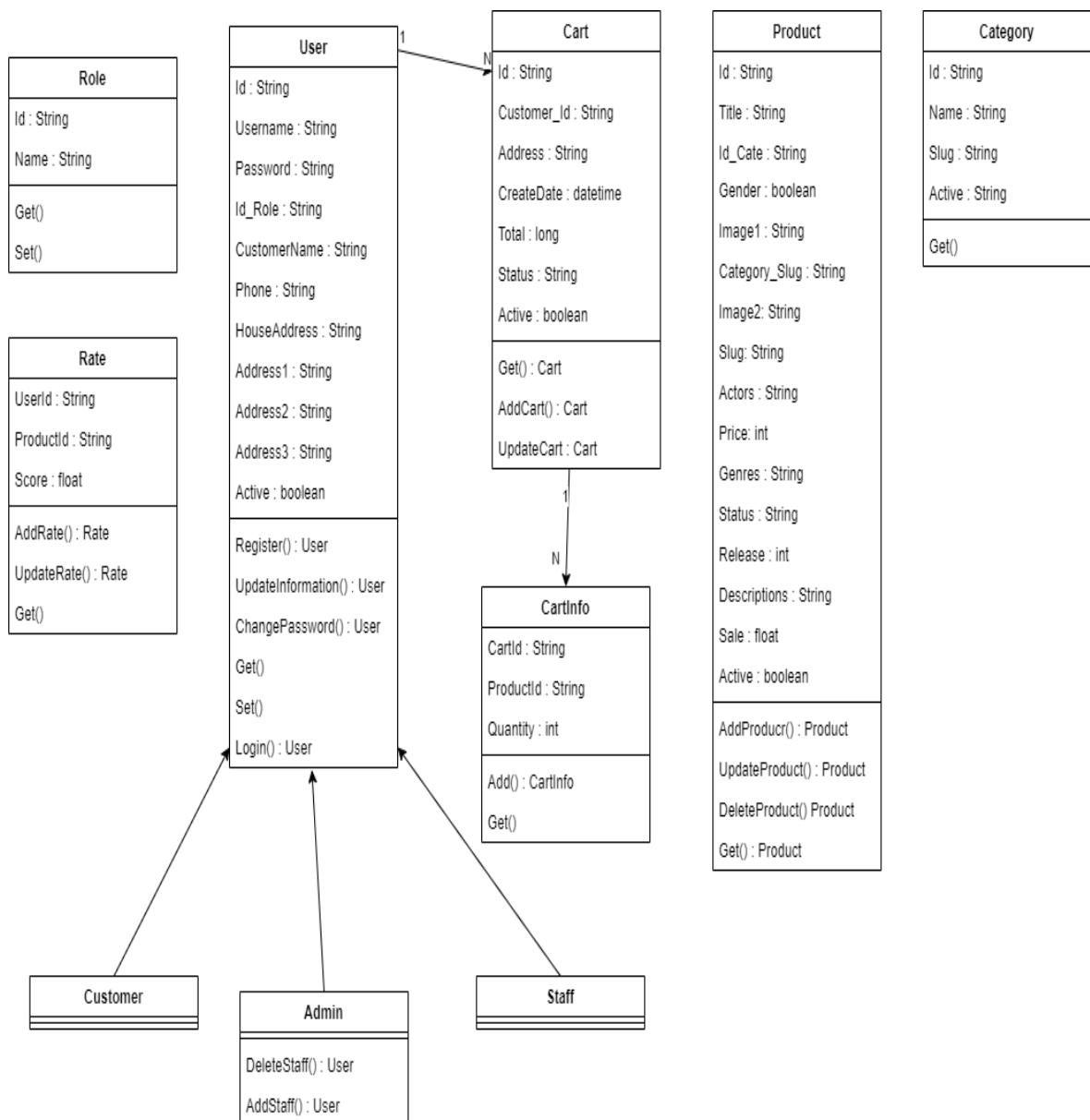
**Hình 3-2 Sơ đồ use cases người dùng**

### 3.1.2.2. Hệ thống



**Hình 3-3 Sơ đồ use cases hệ thống**

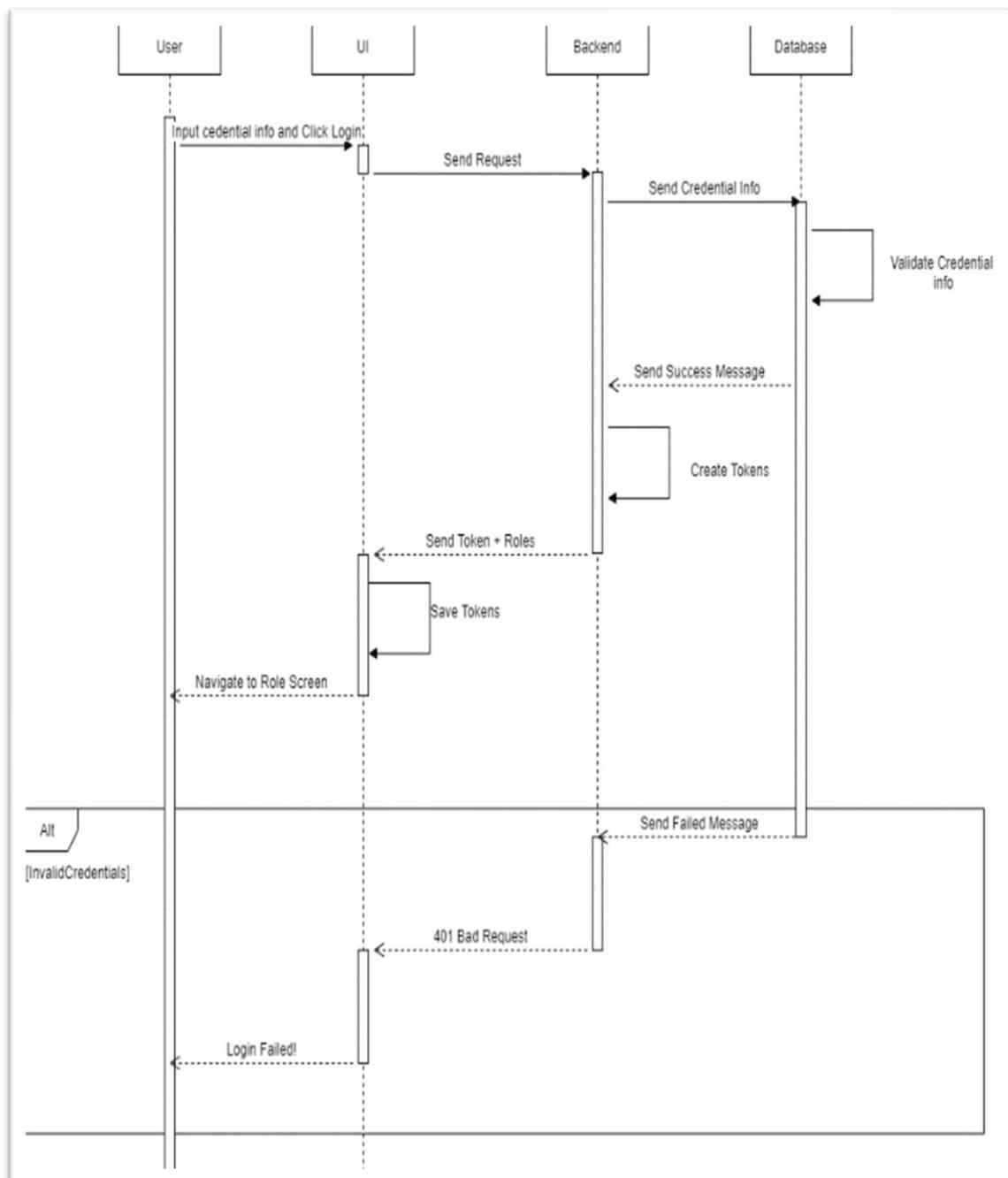
### 3.1.3. Sơ đồ lớp



Hình 3-4 Sơ đồ lớp

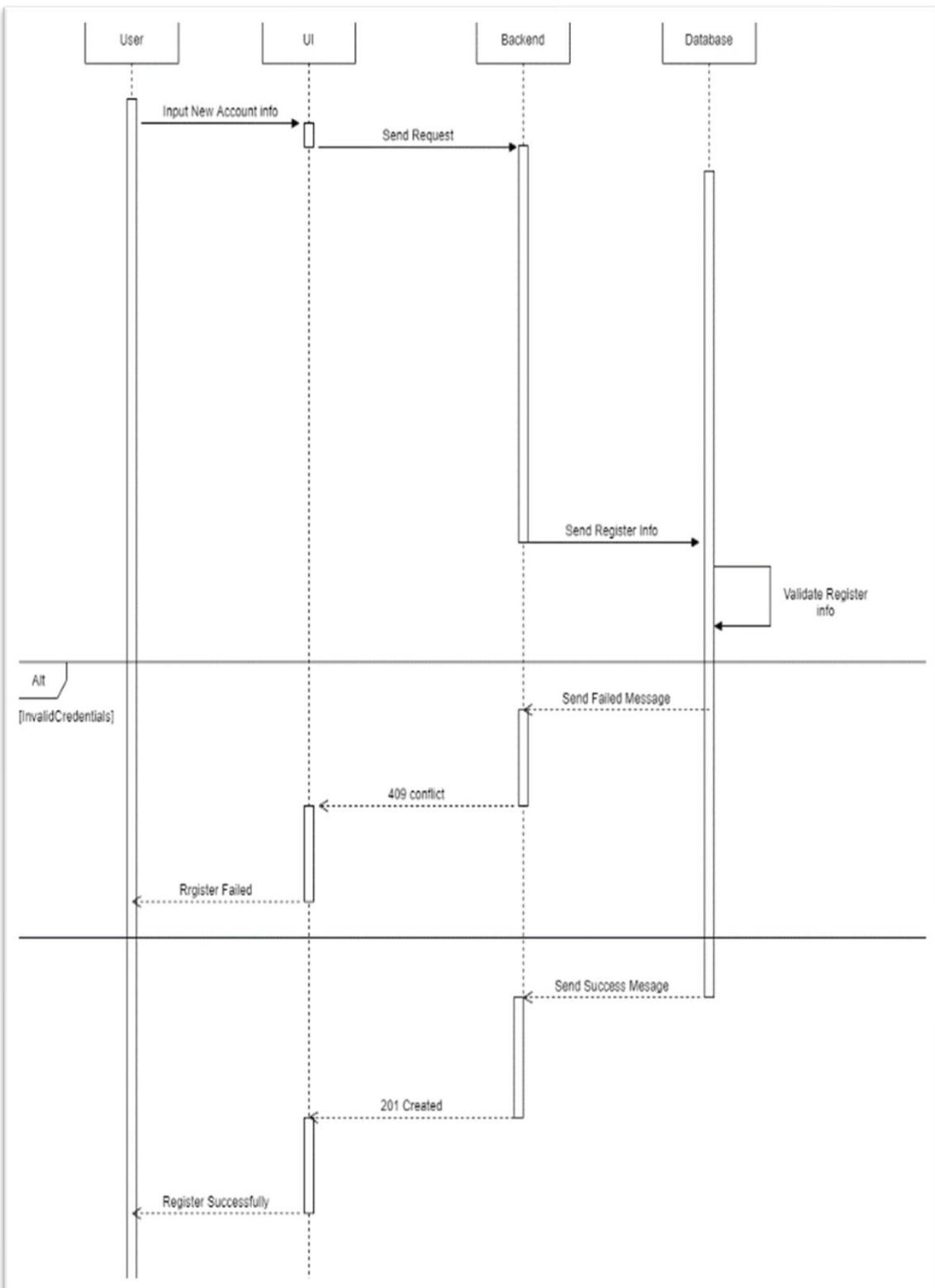
### 3.1.4. Sơ đồ tuần tự

#### 3.1.4.1. Đăng nhập



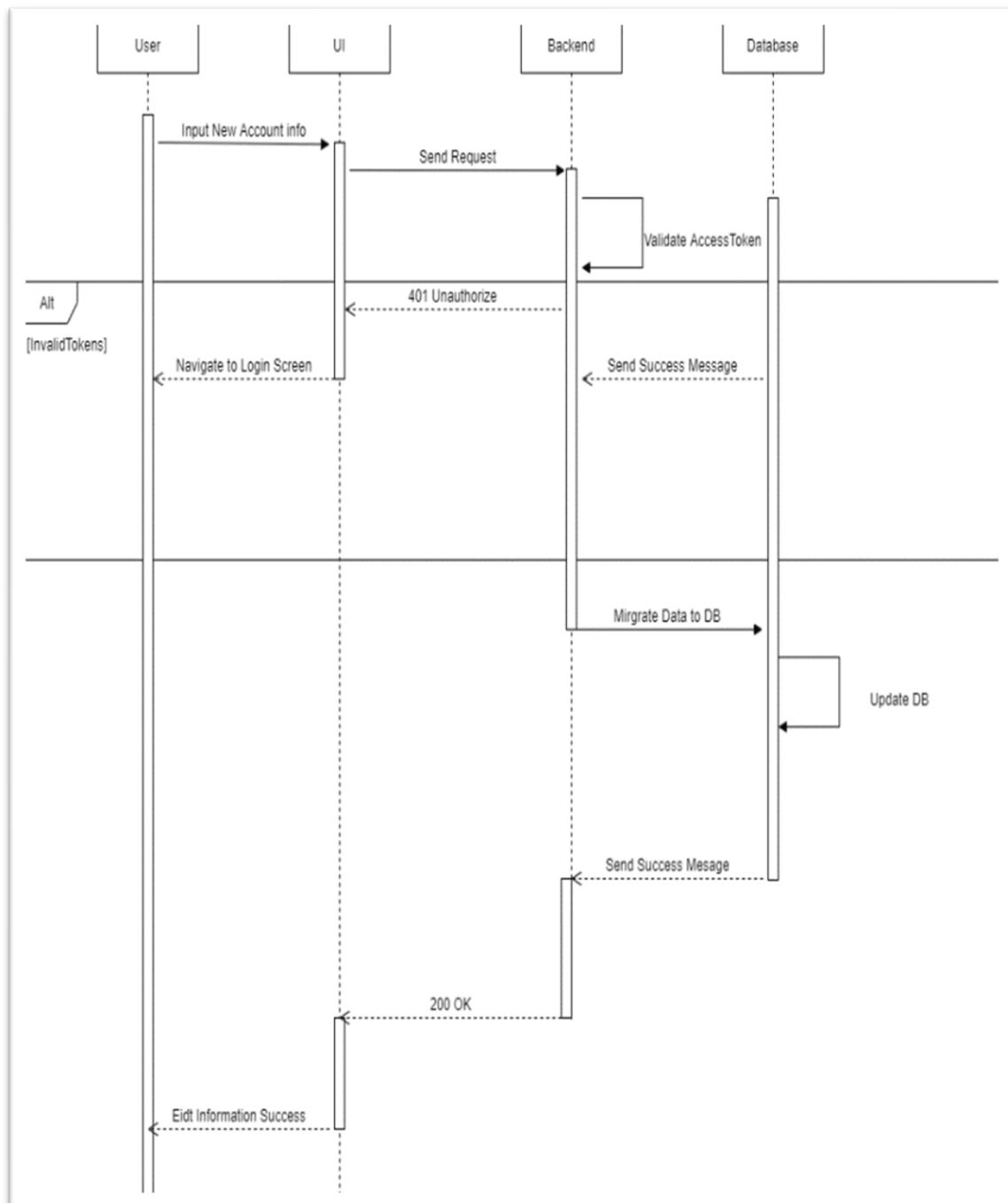
Hình 3-5 Sơ đồ tuần tự đăng nhập

### 3.1.4.2. Đăng ký



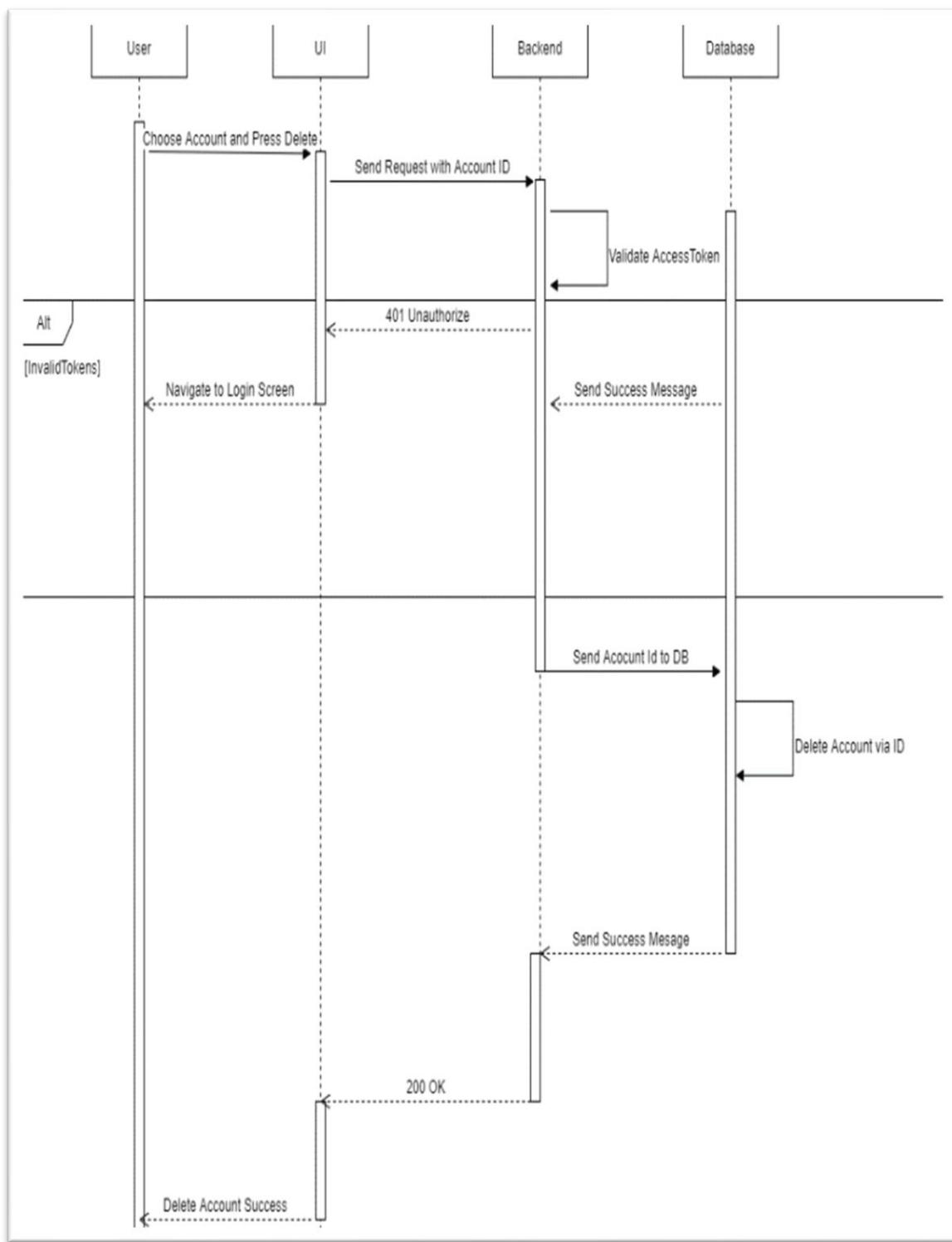
Hình 3-6 Sơ đồ tuần tự đăng ký

### 3.1.4.3. Chính sửa thông tin cá nhân



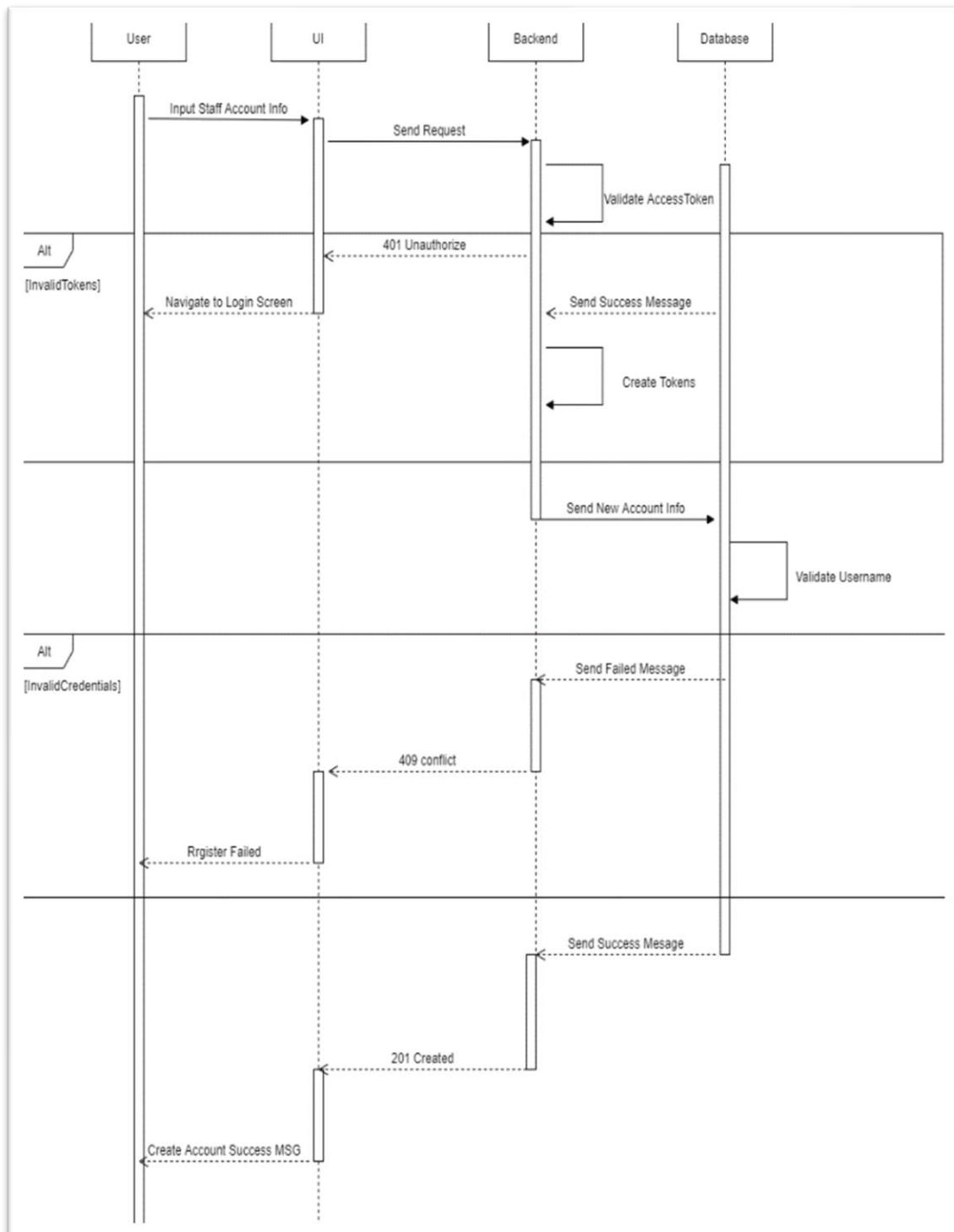
Hình 3-7 Sơ đồ tuần tự chỉnh sửa thông tin

### 3.1.4.4. Xóa tài khoản



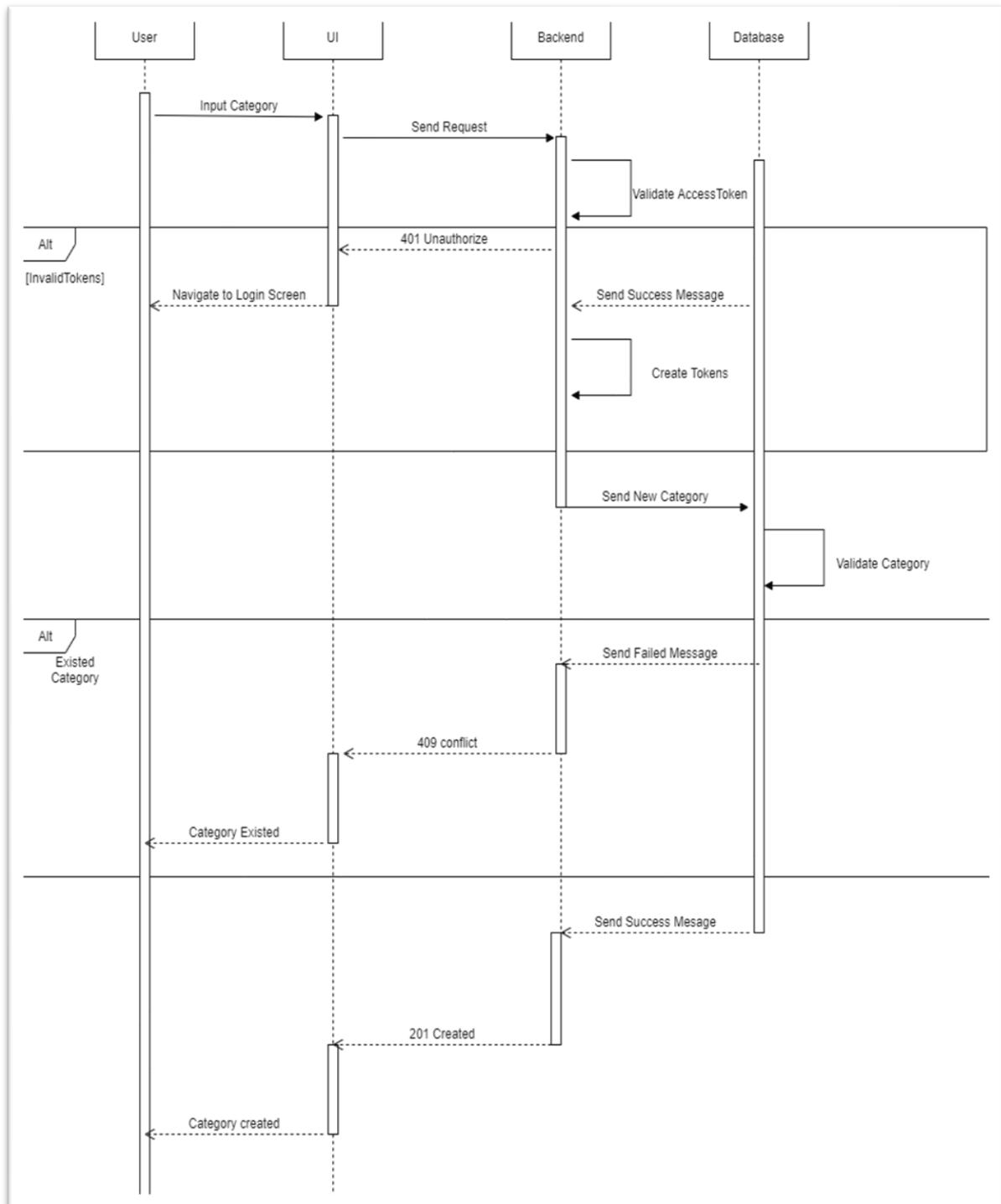
**Hình 3-8 Sơ đồ tuần tự xóa tài khoản**

### 3.1.4.5. Tạo tài khoản nhân viên



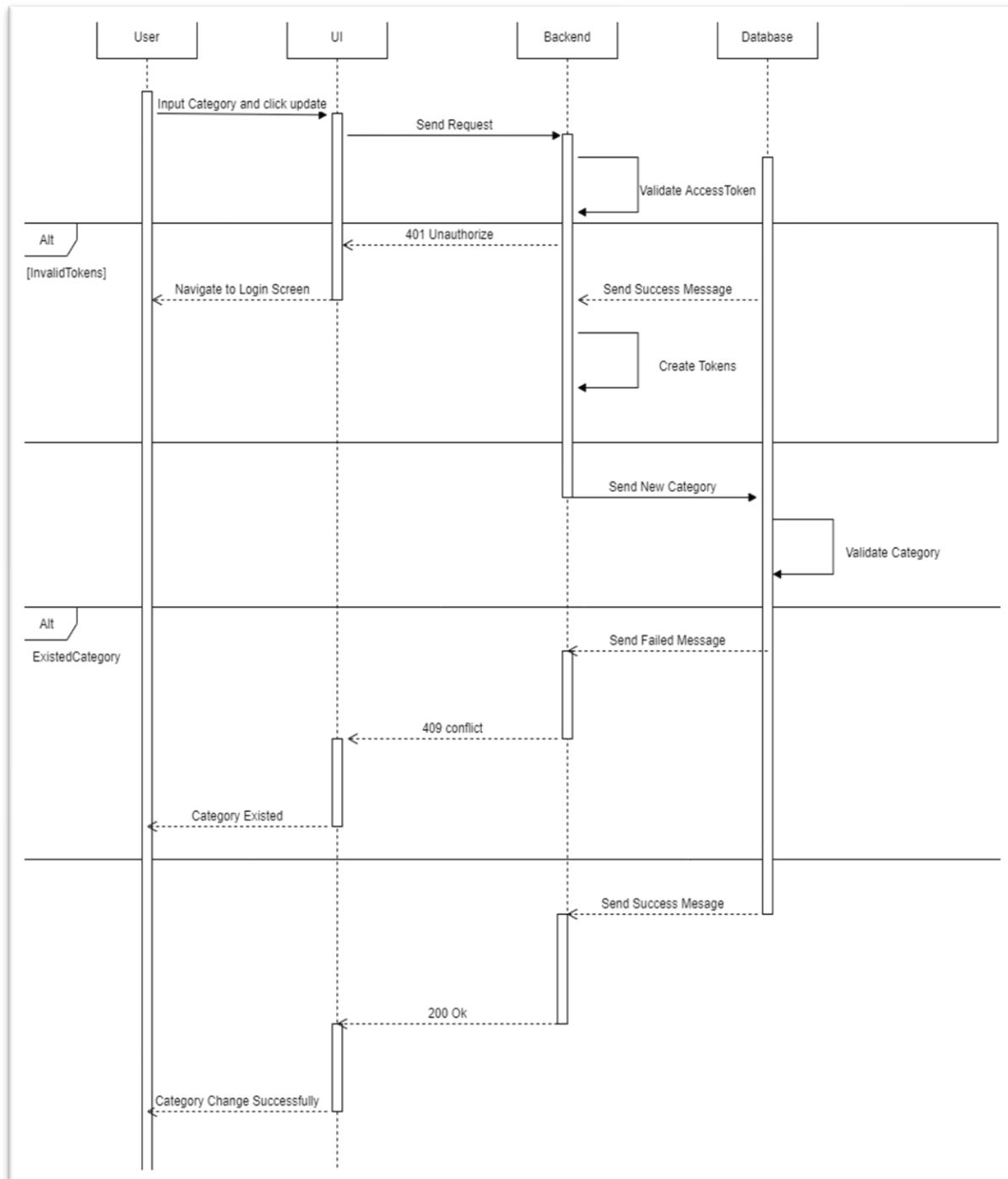
Hình 3-9 Sơ đồ tuần tự tạo tài khoản nhân viên

### 3.1.4.6. Tạo phân loại



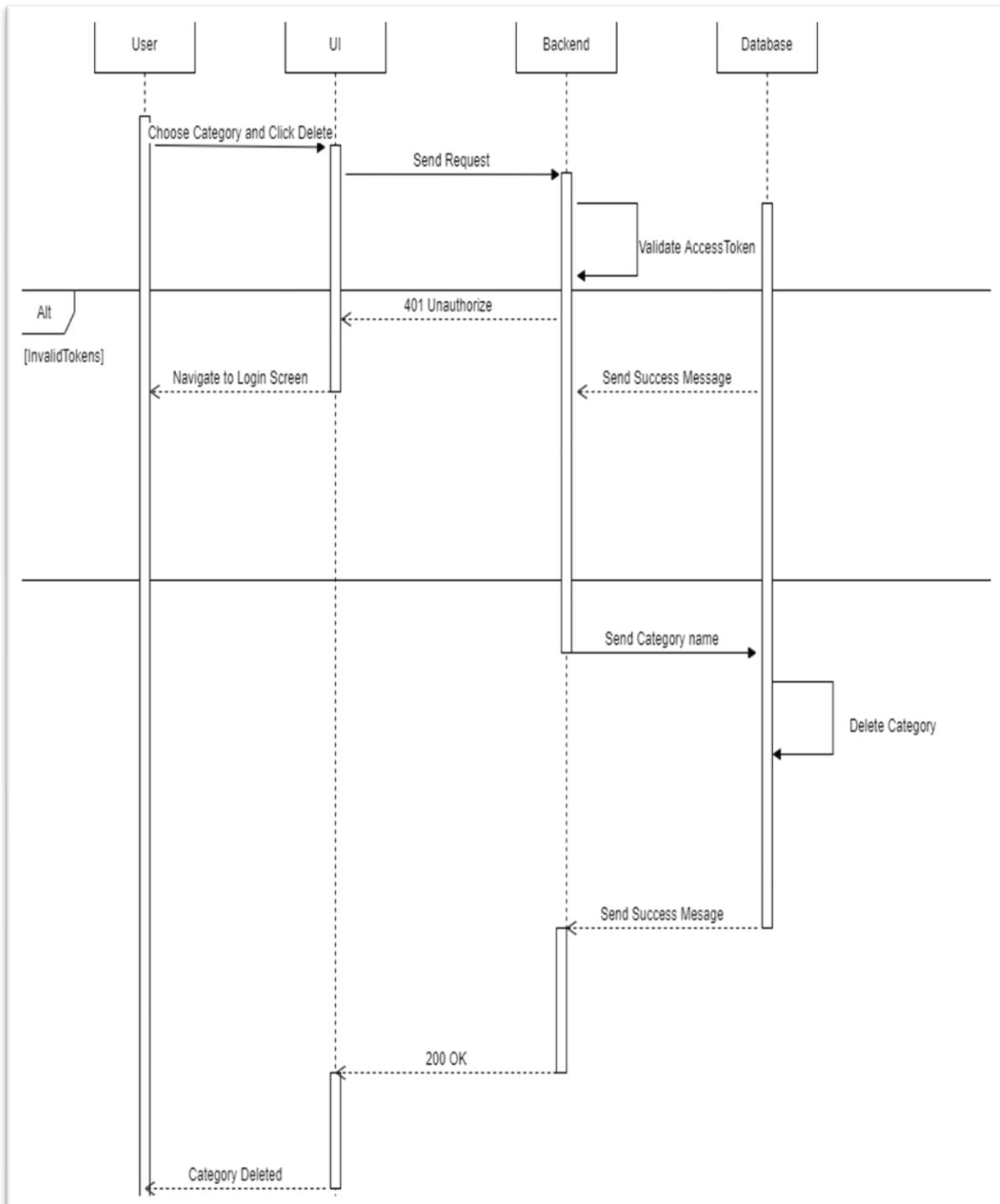
Hình 3-10 Sơ đồ tuần tự tạo phân loại

### 3.1.4.7. Sửa phân loại



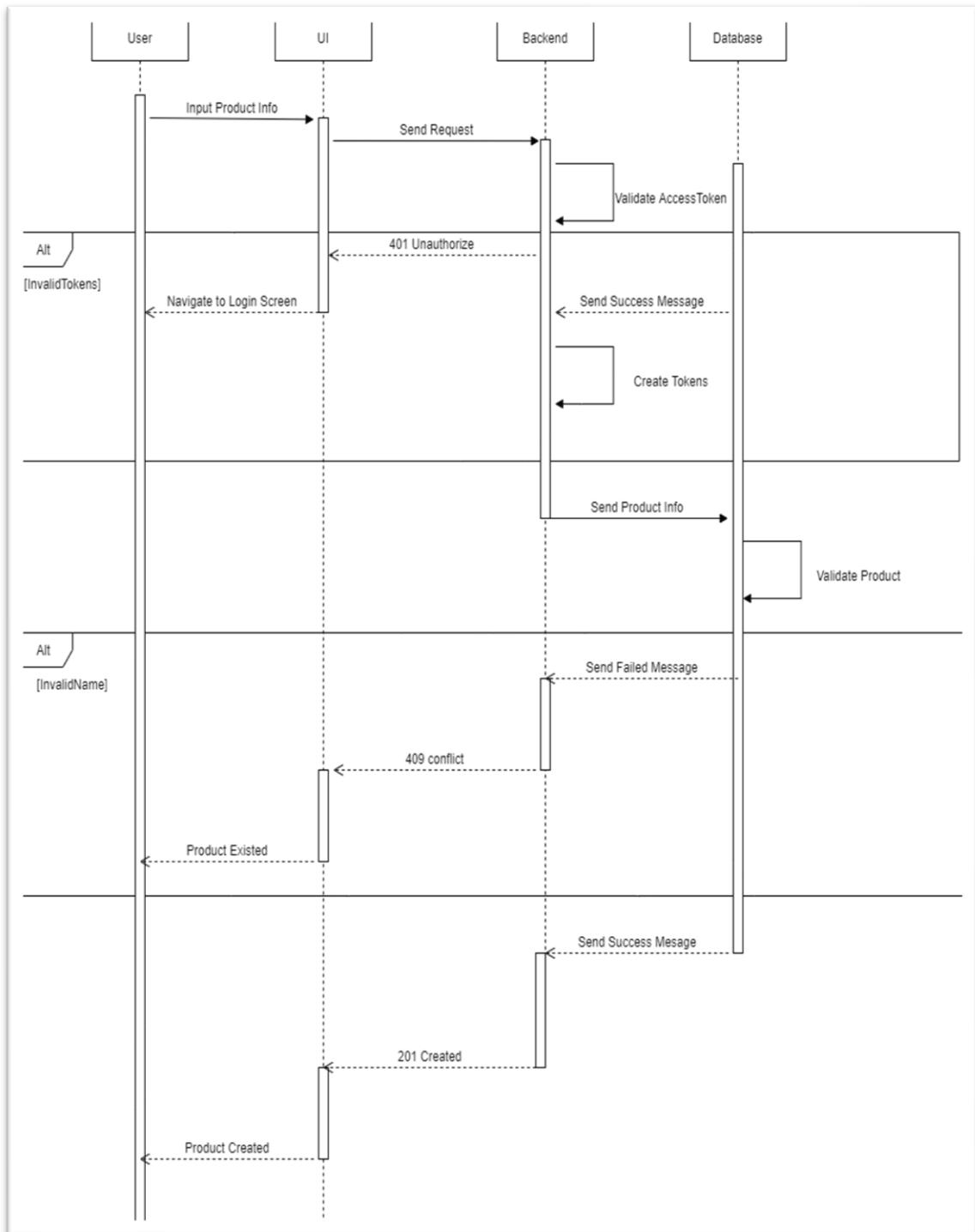
Hình 3-11 Sơ đồ tuần tự sửa phân loại

### 3.1.4.8. Xóa phân loại



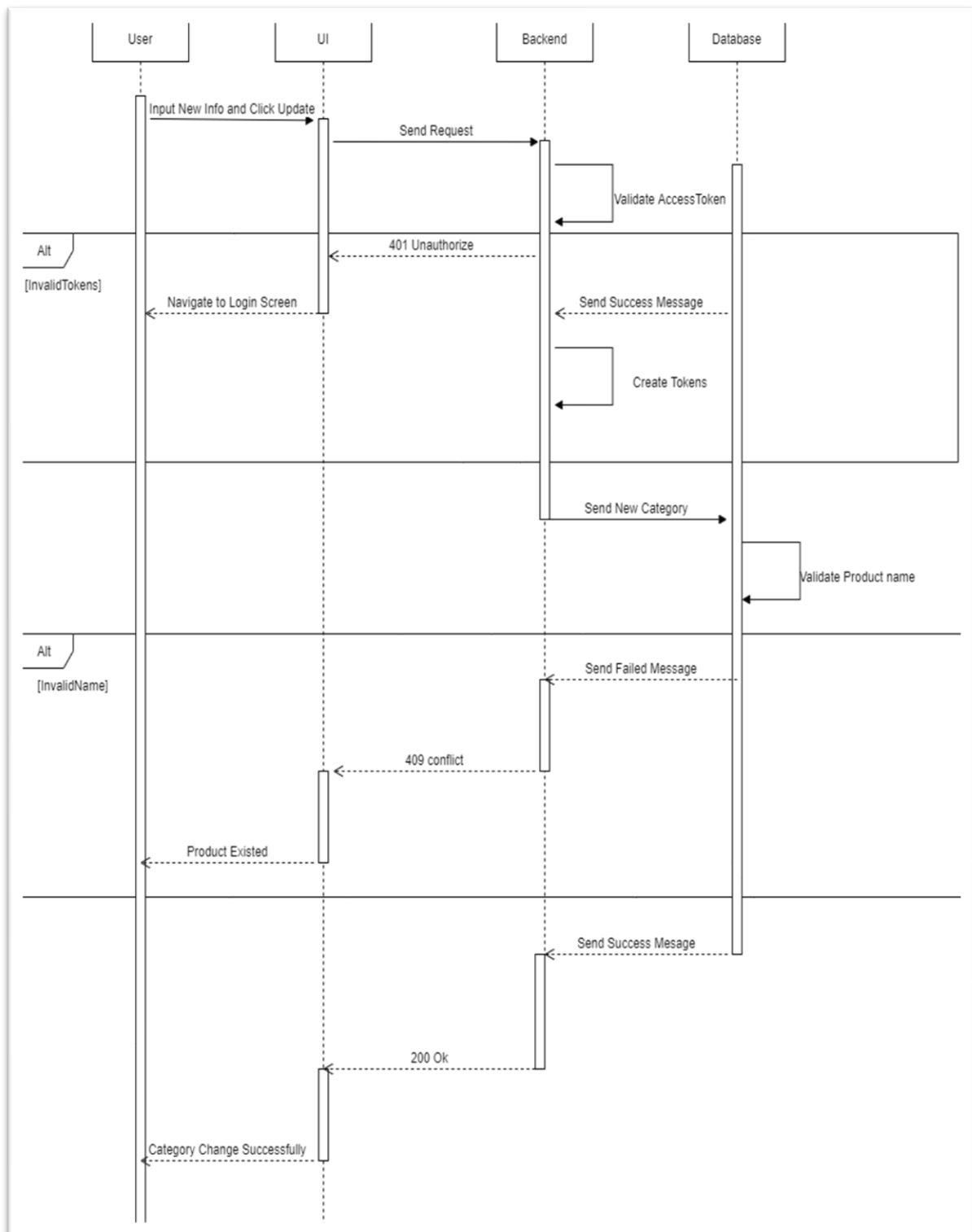
Hình 3-12 Sơ đồ tuần tự xóa phân loại

### 3.1.4.9. Tạo phim



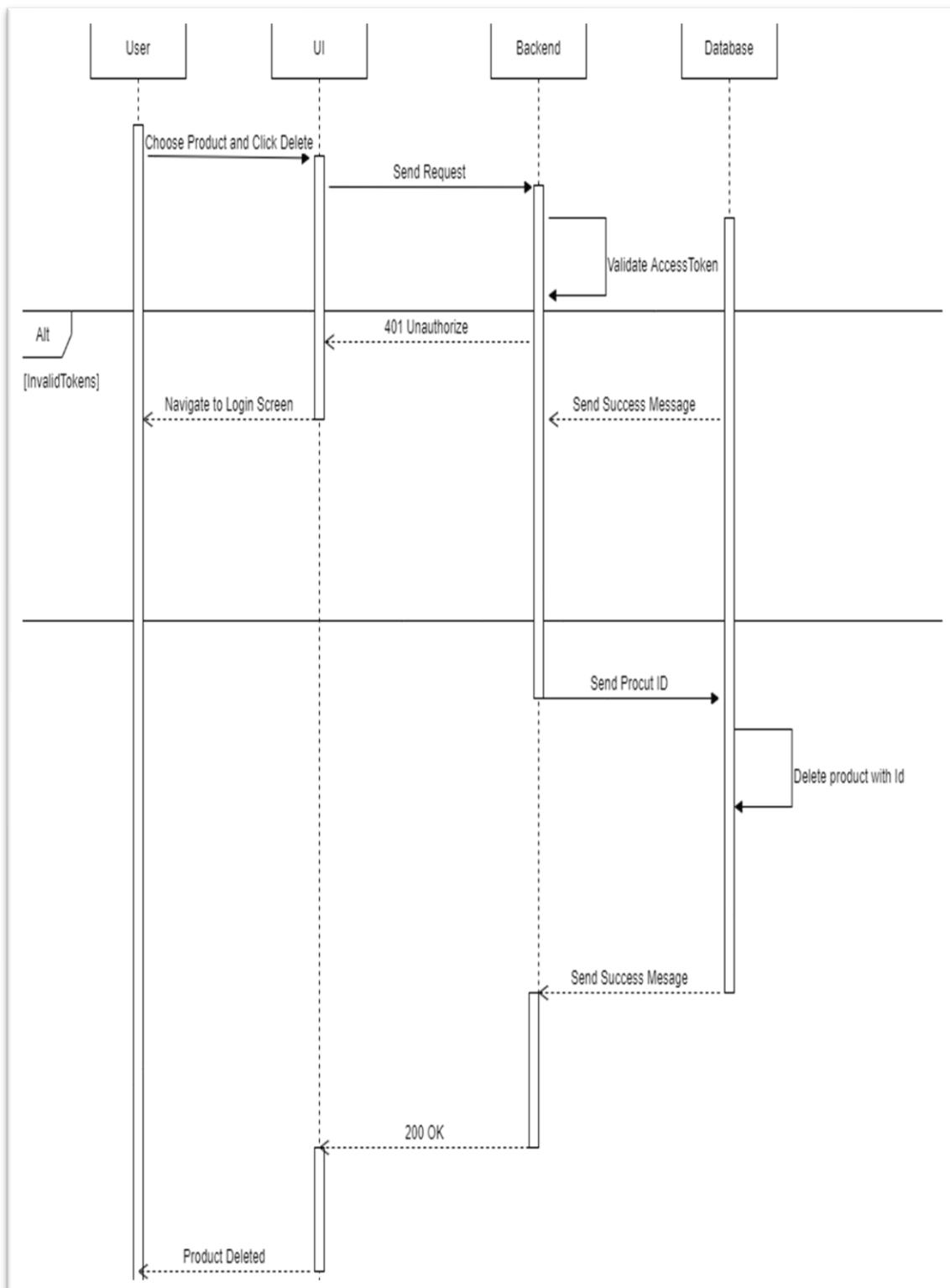
Hình 3-13 Sơ đồ tuần tự tạo phim

### 3.1.4.10. Sửa thông tin phim



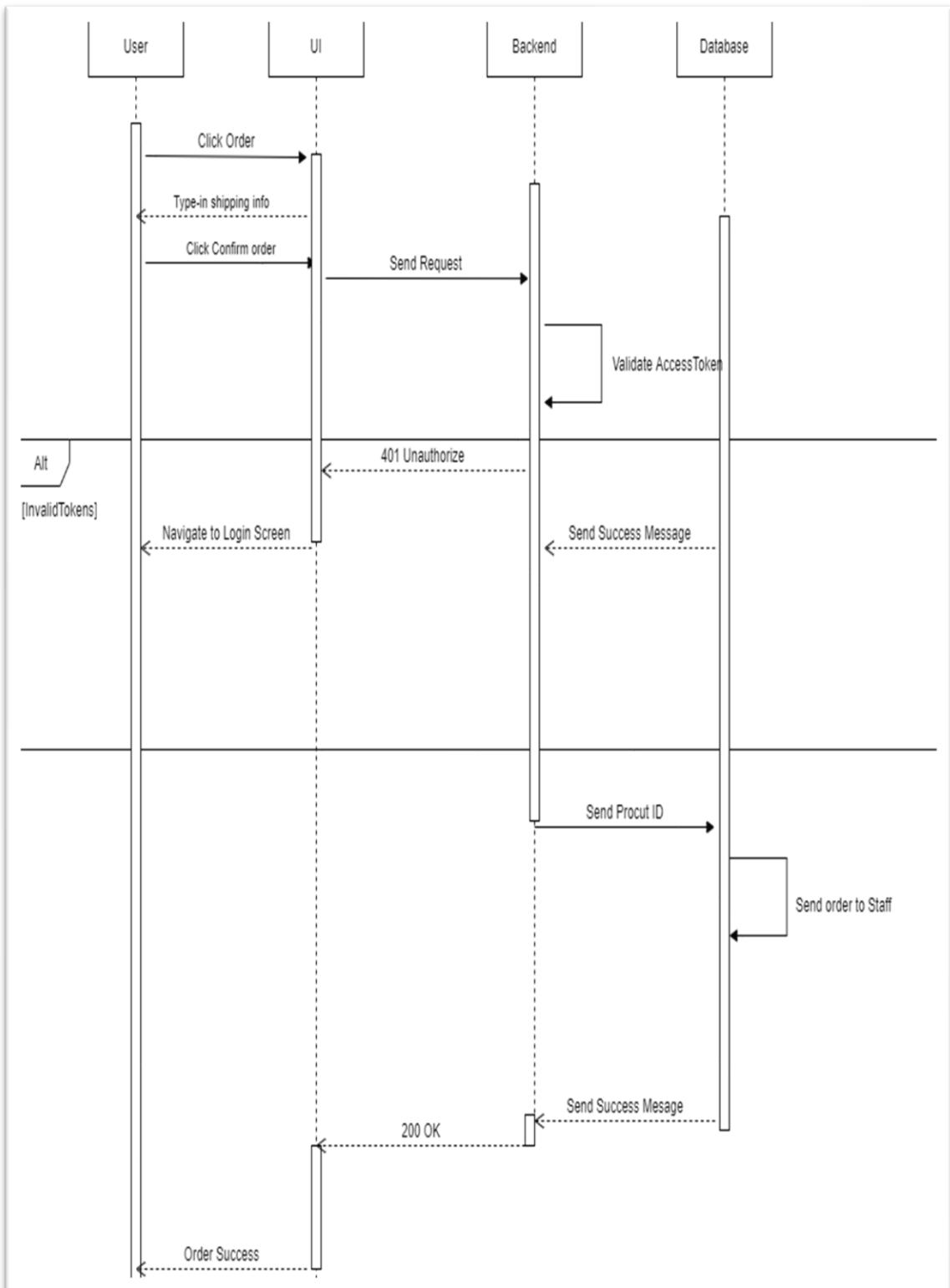
Hình 3-14 Sơ đồ tuần tự sửa thông tin phim

### 3.1.4.11. Xóa phim



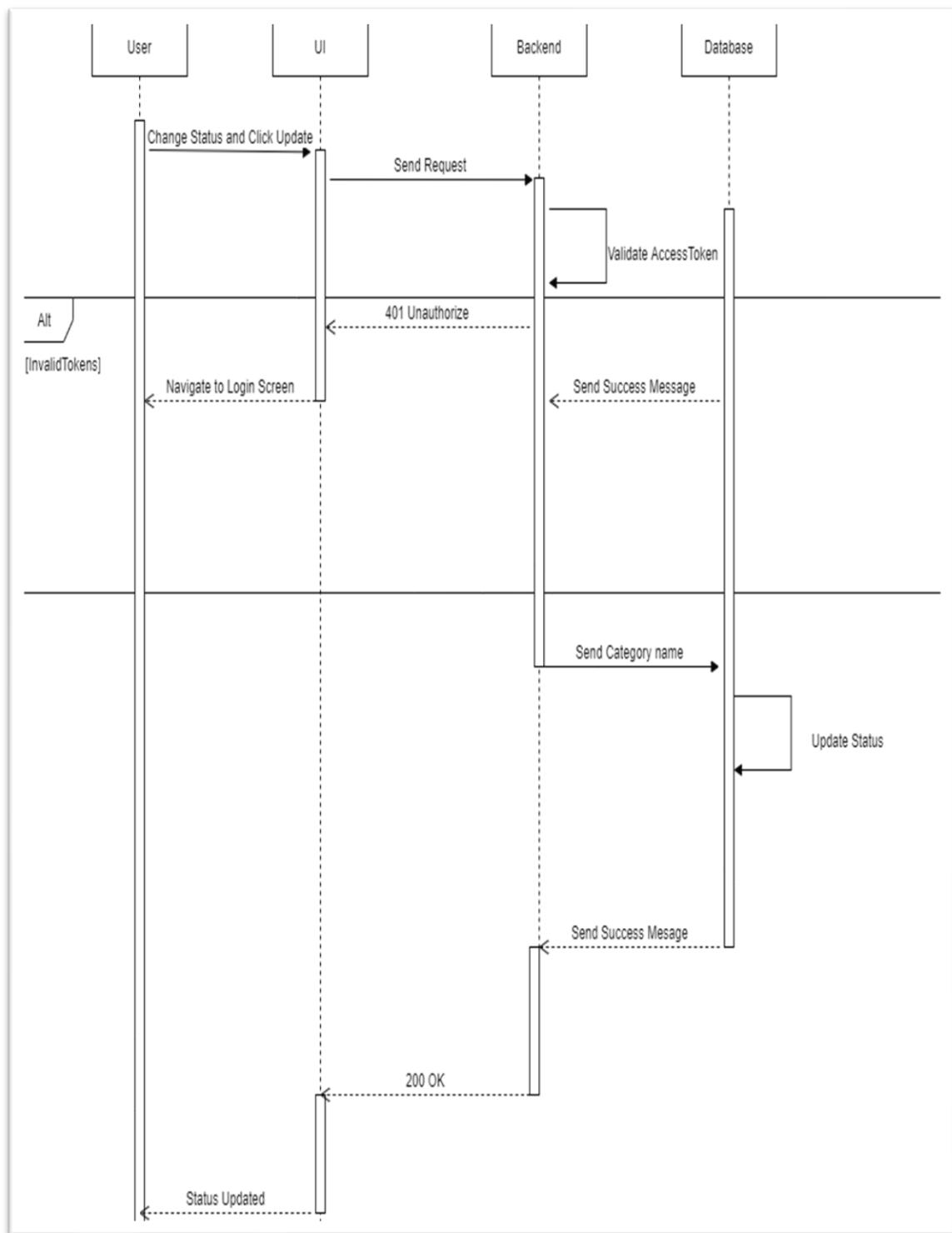
Hình 3-15 Sơ đồ tuần tự xóa phim

### 3.1.4.12. Mua hàng



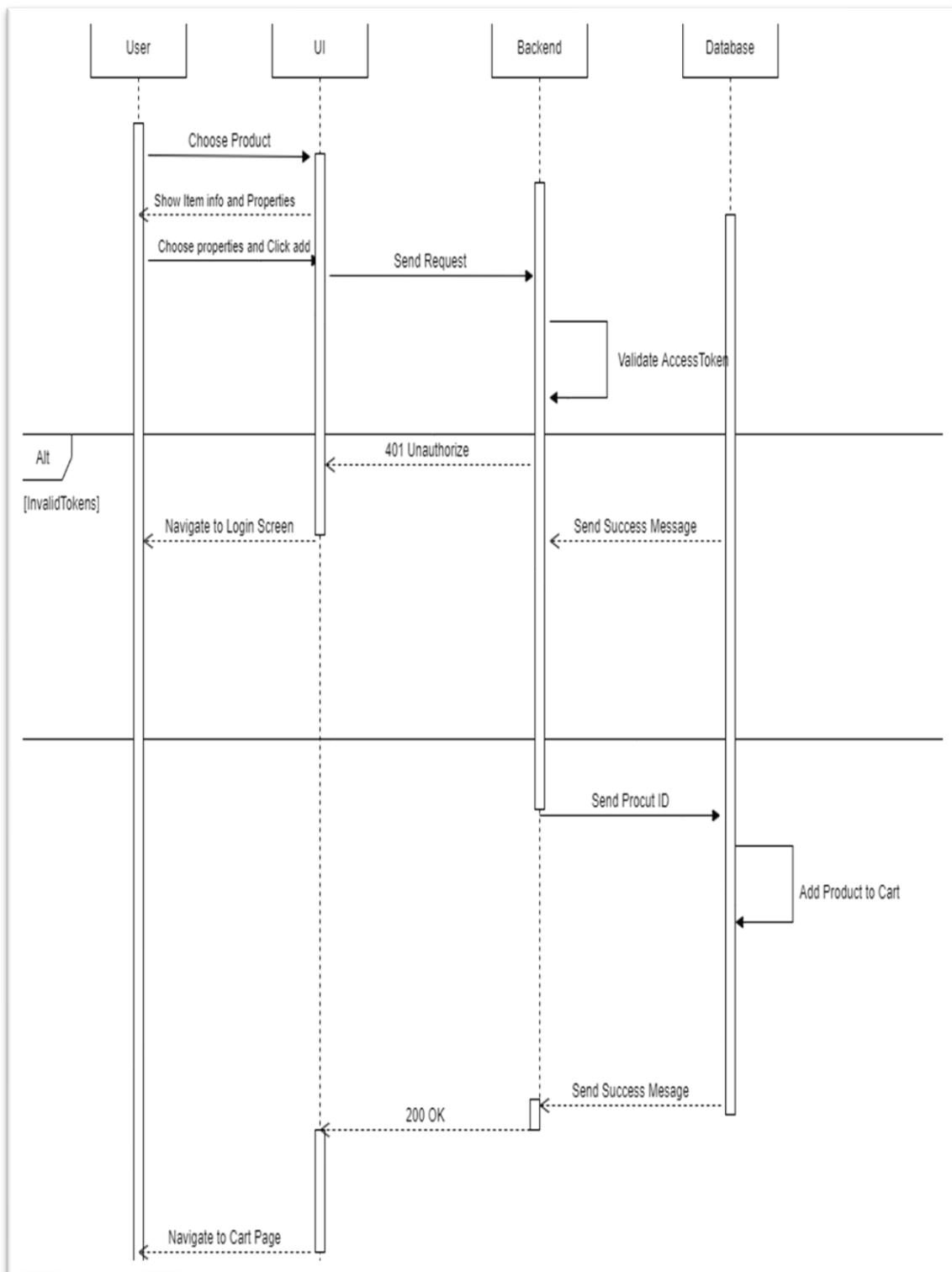
Hình 3-16 Sơ đồ tuần tự mua hàng

### 3.1.4.13. Thay đổi tình trạng đơn hàng



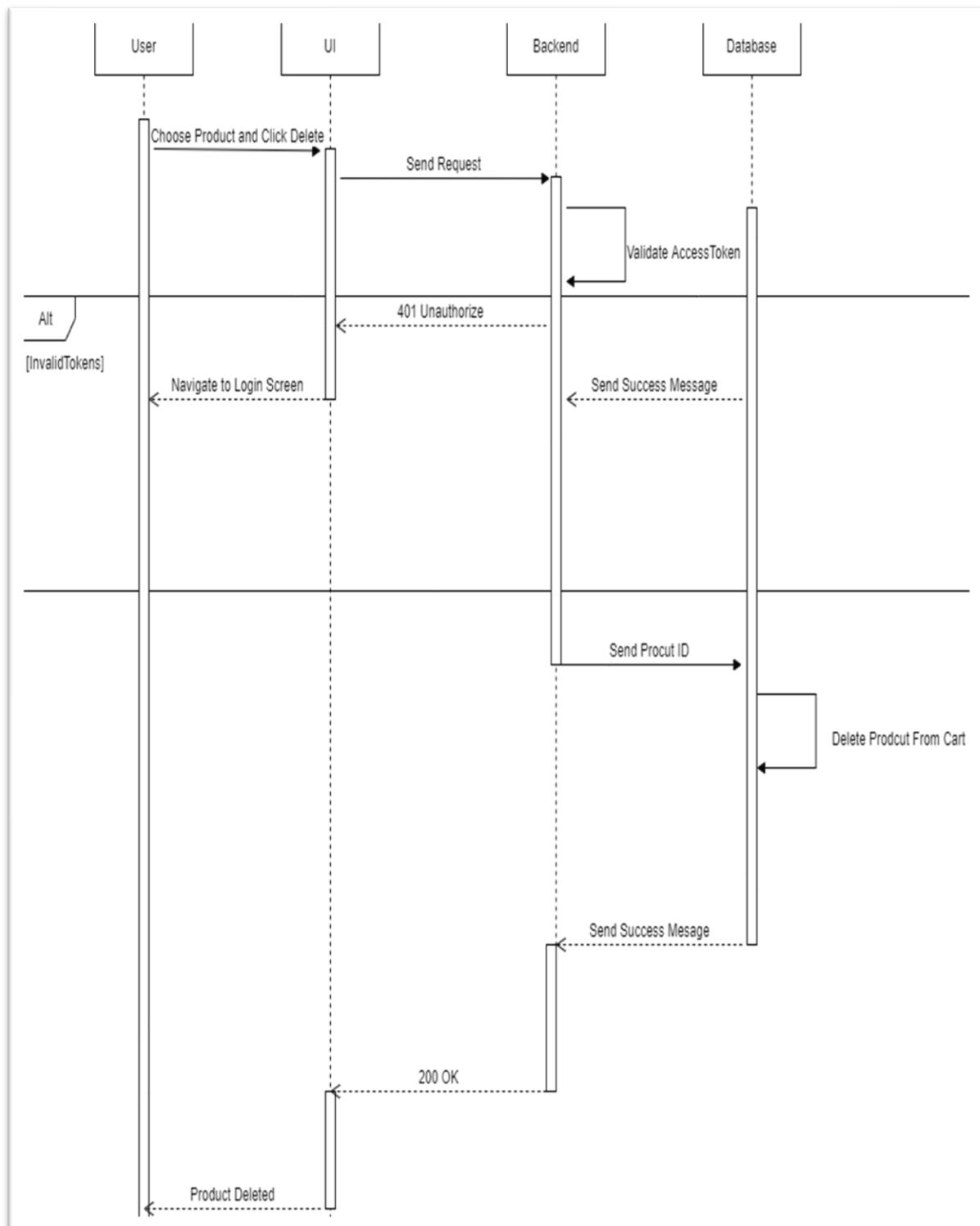
Hình 3-17 Sơ đồ tuần tự tình trạng đơn hàng

### 3.1.4.14. Thêm hàng vào giỏ



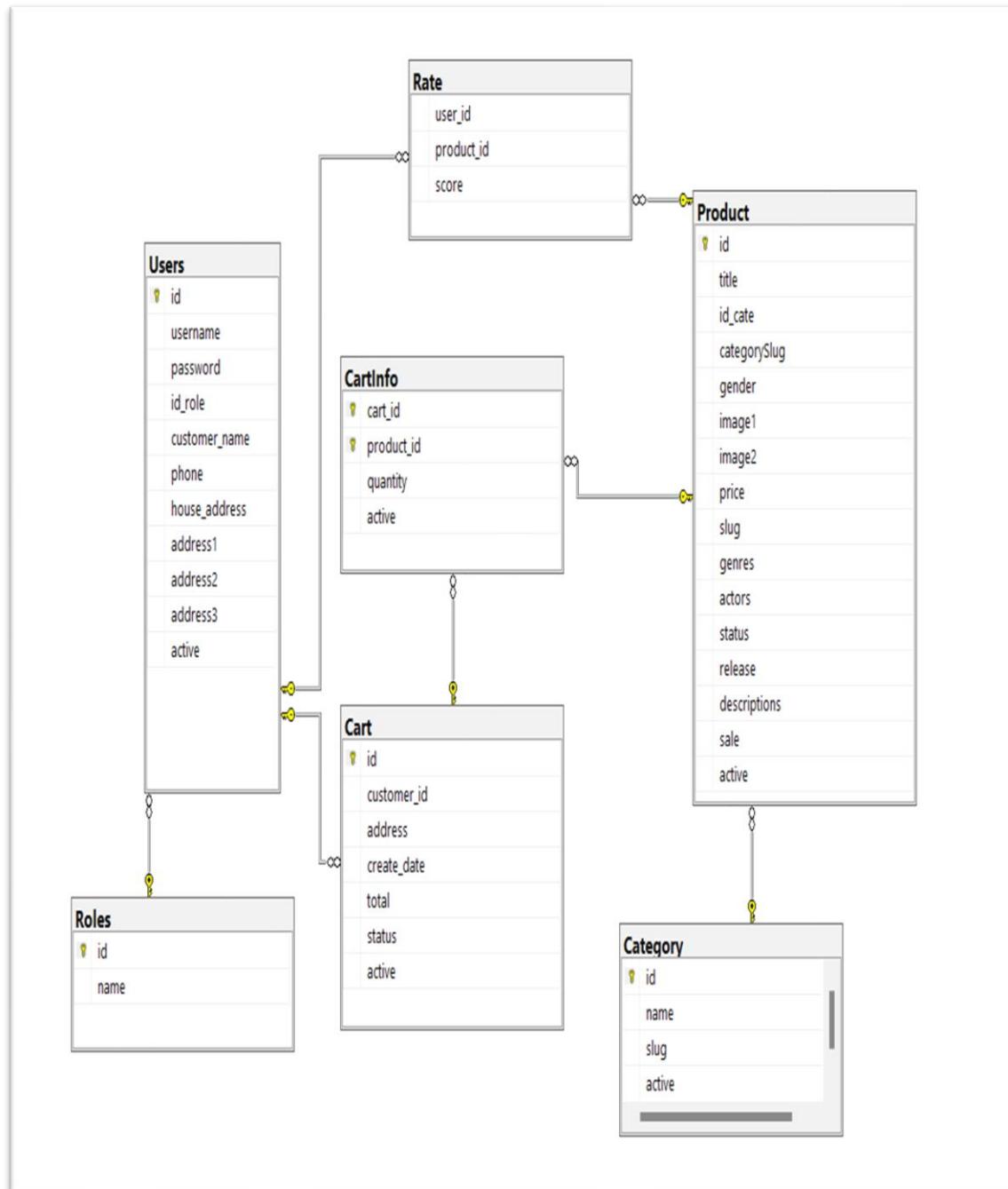
Hình 3-18 Sơ đồ tuần tự thêm hàng vào giỏ

### 3.1.4.15. Xóa hàng khỏi giỏ



Hình 3-19 Sơ đồ tuần tự xóa hàng khỏi giỏ

### 3.1.5. Cơ sở dữ liệu



Hình 3-20 Sơ đồ Database

### **3.1.5.1. Sơ đồ Database**

### **3.1.5.2. Cấu trúc bảng**

#### **a. Users**

**Bảng 3-1 Users**

| Tên cột        | Kiểu dữ liệu | Kích cỡ | Mô tả                       |
|----------------|--------------|---------|-----------------------------|
| <u>id</u>      | varchar      | 5       | Id của user                 |
| username       | varchar      | 50      | Email được đặt là tài khoản |
| password       | varchar      | max     | Mật khẩu                    |
| <i>id_role</i> | varchar      | 100     | Id của role                 |
| customer_name  | varchar      | 50      | Tên user                    |
| phone          | varchar      | 20      | Số điện thoại user          |
| house_address  | varchar      | 100     | Địa chỉ nhà                 |
| address1       | varchar      | 100     | Địa chỉ phường              |
| address2       | varchar      | 100     | Địa chỉ quận                |
| address3       | varchar      | 100     | Địa chỉ thành phố           |
| active         | bit          |         | Cờ hiệu dành cho xóa        |

#### **b. Roles**

**Bảng 3-2 Roles**

| Tên cột   | Kiểu dữ liệu | Kích cỡ | Mô tả        |
|-----------|--------------|---------|--------------|
| <u>id</u> | varchar      | 5       | Id của role  |
| name      | varchar      | 50      | Tên của role |

### c. Cart

Bảng 3-3 Cart

| Tên cột     | Kiểu dữ liệu  | Kích cỡ | Mô tả  |
|-------------|---------------|---------|--|
| <u>id</u>   | varchar       | 9       | Id của cart dùng để tham chiếu đến các cart info |
| Customer_id | varchar       | 5       | Id của customer                                  |
| Adress      | nvarchar      | Max     | Địa chỉ giao hàng của cart                       |
| Create_date | Smalldatetime |         | Ngày tạo đơn                                     |
| Total       | Int           |         | Tổng tiền đơn                                    |
| Status      | nvarchar      | 50      | Trạng thái đơn                                   |
| Active      | Bit           |         | Cờ hiệu dành cho xóa                             |

### d. CartInfo

Bảng 3-4 CartInfo

| Tên cột        | Kiểu dữ liệu | Kích cỡ | Mô tả                |
|----------------|--------------|---------|----------------------|
| <u>cart_id</u> | varchar      | 9       | Id của cart          |
| product_id     | varchar      | 10      | Id của product       |
| quantity       | int          |         | Số lượng product     |
| active         | bit          |         | Cờ hiệu dành cho xóa |

## e. Product

**Bảng 3-5 Product**

| Tên cột        | Kiểu dữ liệu | Kích cỡ | Mô tả                   |
|----------------|--------------|---------|-------------------------|
| <u>id</u>      | varchar      | 10      | Id của product          |
| title          | nvarchar     | 100     | Tên của product         |
| <u>id_cate</u> | varchar      | 10      | Id của category         |
| categoryslug   | varchar      | max     | Phân loại của category  |
| gender         | nvarchar     | 10      | Giới tính sản phẩm      |
| image1         | varchar      | max     | Link ảnh 1              |
| image2         | varchar      | max     | Link ảnh 2              |
| price          | int          |         | Giá product             |
| slug           | varchar      | 100     | Phân loại product       |
| genres         | varchar      | 100     | Thể loại                |
| actors         | varchar      | 50      | Diễn viên               |
| status         | varchar      | 50      | Tính trạng              |
| release        | int          |         | Năm phát hành           |
| descriptions   | nvarchar     | max     | Mô tả sản phẩm          |
| sale           | int          |         | Phần trăm sale sản phẩm |

|        |     |  |                      |
|--------|-----|--|----------------------|
| active | bit |  | Cờ hiệu dành cho xóa |
|--------|-----|--|----------------------|

### f. Category

Bảng 3-6 Category

| Tên cột   | Kiểu dữ liệu | Kích cỡ | Mô tả                |
|-----------|--------------|---------|----------------------|
| <u>id</u> | varchar      | 10      | Id của category      |
| active    | bit          |         | Cờ hiệu dành cho xóa |
| display   | varchar      | 50      |                      |

## 3.2. Thiết kế giao diện

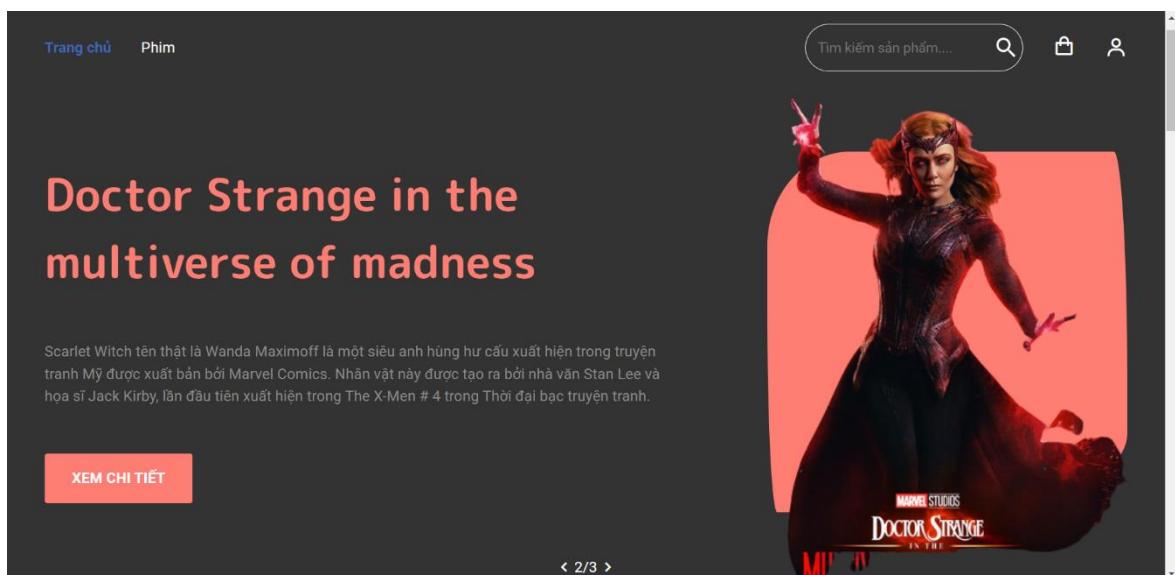
### 3.2.1. Giao diện ứng dụng

#### 3.2.1.1. Giao diện thanh đầu trang

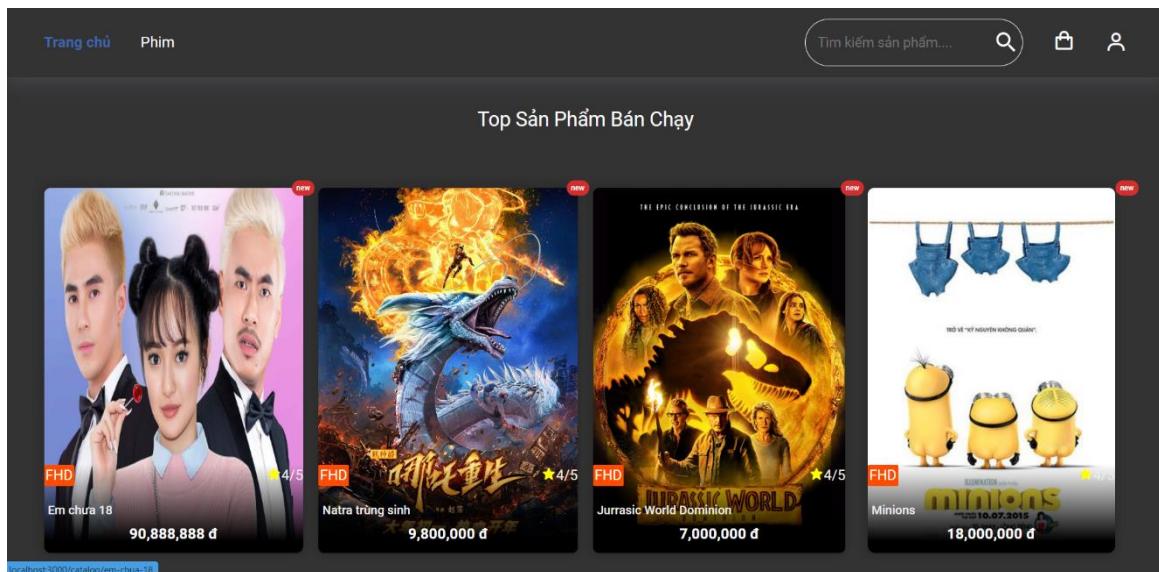


Hình 3-22 Giao diện thanh đầu trang

#### 3.2.1.2. .Trang chủ



Hình 3-21 Trang chủ và slider



**Hình 3-23 Giao diện trang chủ với các phim bán chạy**

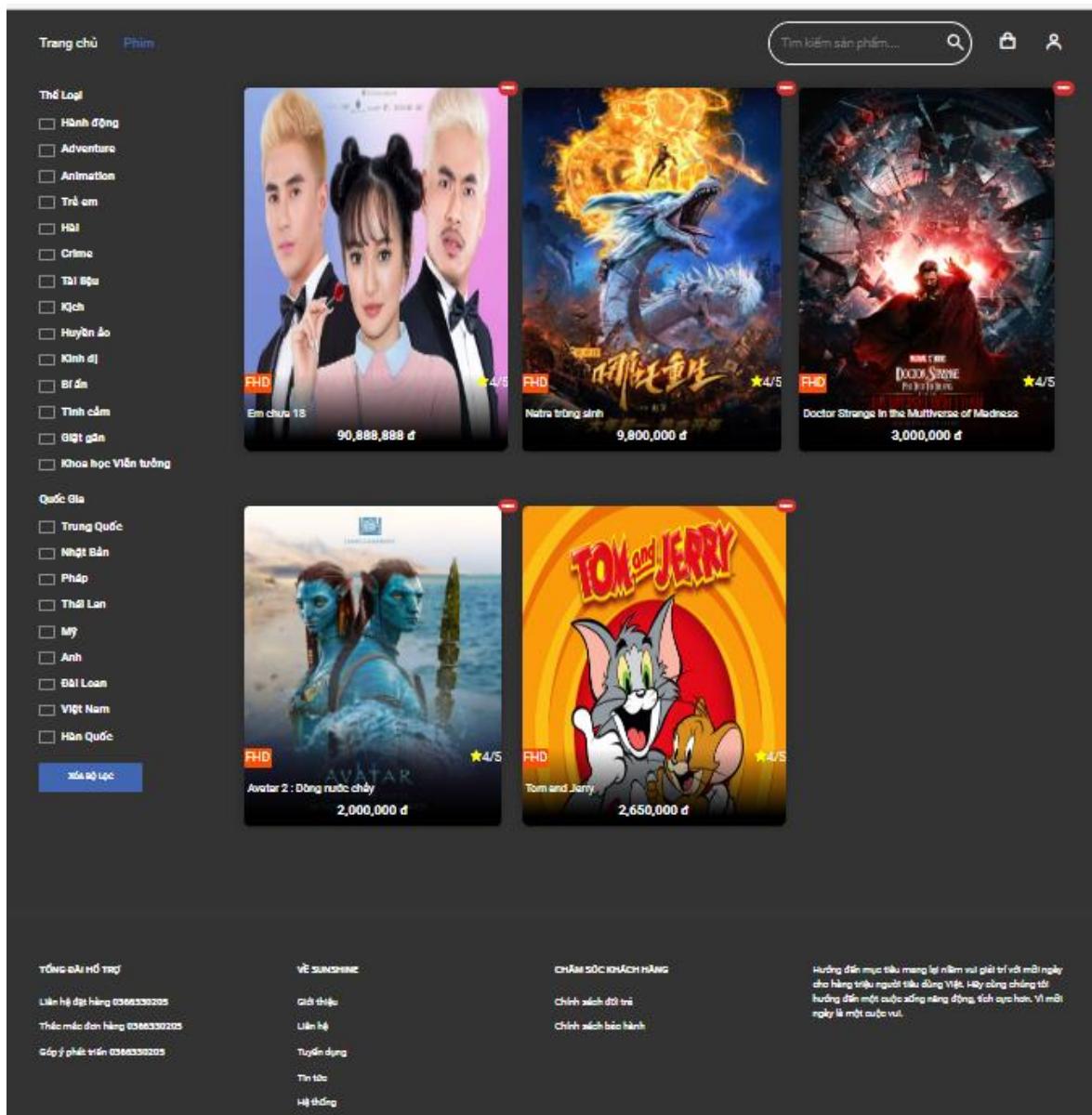
Giao diện trang chủ với phần các bộ phim bán chạy.

| TỔNG ĐÀI HỖ TRỢ              | VỀ SUNSHINE | CHĂM SÓC KHÁCH HÀNG |  |
|------------------------------|-------------|---------------------|--|
| Liên hệ đặt hàng 0366330205  | Giới thiệu  | Chính sách đổi trả  | Hướng đến mục tiêu mang lại niềm vui giải trí với mỗi ngày |
| Thắc mắc đơn hàng 0366330205 | Liên hệ     | Chính sách bảo hành | cho hàng triệu người tiêu dùng Việt. Hãy cùng chúng tôi    |
| Góp ý phát triển 0366330205  | Tuyển dụng  |                     | hướng đến một cuộc sống năng động, tích cực hơn. Vì mỗi    |
|                              | Tin tức     |                     | ngày là một cuộc vui.                                      |

**Hình 3-24 Trang chủ và các phim phổ biến**

Giao diện trang chủ với phần các bộ phim phổ biến.

### 3.2.1.3. Trang danh sách các sản phẩm phim



Hình 3-25 Giao diện danh sách phim

Giao diện hiển thị tất cả các bộ phim và có thể lọc tìm kiếm theo bộ lọc.

### 3.2.1.4. Giao diện chi tiết sản phẩm

#### a. Giao diện chi tiết sản phẩm thông tin

The screenshot shows a product detail page for the movie 'Avatar 2: The Way of Water'. At the top left are navigation links 'Trang chủ' and 'Phim'. A search bar with placeholder text 'Tim kiem san pham....' and a magnifying glass icon is at the top right. Below the search bar is a large image of the movie poster featuring Neytiri and Jake Sully. To the left of the main image are two smaller thumbnail images of the movie. To the right of the main image, the product title is 'Avatar 2 : Dòng nước chảy', the price is '2,000,000 đ', and the country of origin is 'Mỹ'. Below this, sections include 'Thể loại', 'Đạo diễn', 'Diễn viên', 'Danh giá', and a rating of 'Good'. There is also a 'Số lượng' section with a quantity input set to '1'. At the bottom right are two buttons: 'THÊM VÀO GIỎ' and 'MUA NGAY'. Below the main image, there is a section titled 'Chi tiết sản phẩm' containing descriptive text about the movie's development and cast.

Chi tiết sản phẩm

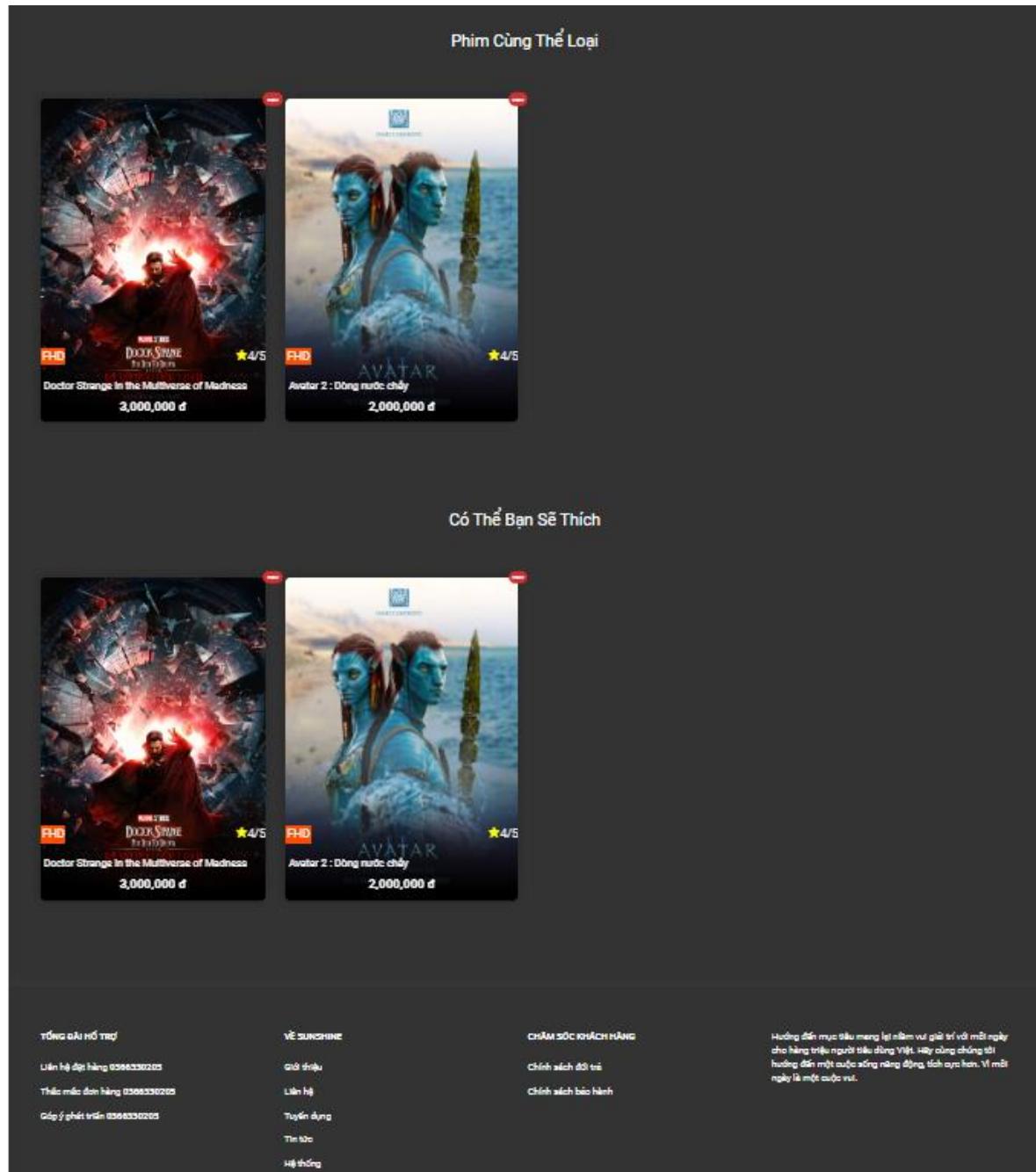
Đạo diễn James Cameron và nhà làm phim Jon Landau cùng xuất hiện tại sự kiện CinemaCon để chia sẻ về các thông tin mới nhất của dự án Avatar 2, đồng thời những hình ảnh đầu tiên của dự án được hé lộ nhưng hiện tại chưa được công bố trên các mặt báo phương Tây. Phần tiếp theo của Avatar có tên chính thức là Avatar: The Way of Water, phim do James Cameron chỉ đạo, đồng biên kịch với Josh Friedman và đồng sản xuất với Jon Landau. Theo như các nguồn tin của Entertainment Weekly, The Verge, phần thứ 2 này tập trung vào Jake Sully (Sam Worthington đóng) và Neytiri (Zoe Saldana đóng), nay họ đã có con. Tuy nhiên, họ lại tiếp tục đối đầu hiểm nguy và cả 2 phải đứng lên chiến đấu để bảo vệ cộng đồng của mình. Các sự kiện trong phim diễn ra sau các diễn biến của phần đầu. Cùng với 2 diễn viên trên, dự án còn quy tụ dàn nghệ sĩ như Edie Falco, Kate Winslet, Dương Tú Quỳnh và nhiều gương mặt mới. Họ là thế hệ tiếp theo của tộc Na'vi.

XEM THÊM

Hình 3-26 Giao diện chi tiết sản phẩm thông tin

Giao diện hiển thị chi tiết các thông tin về phim và các nút để có thể thêm phim vào giỏ hàng

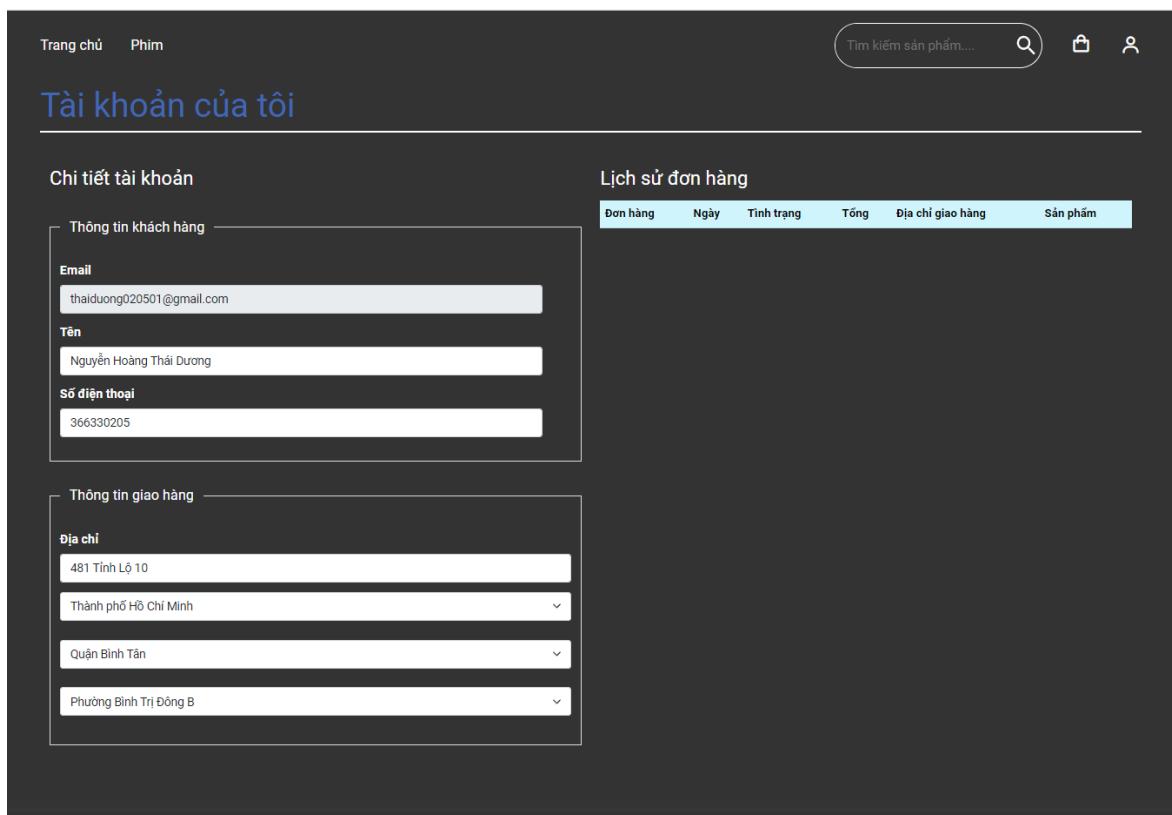
#### b. Giao diện chi tiết sản phẩm gợi ý



**Hình 3-27 Giao diện chi tiết sản phẩm gợi**

Gồm các bộ phim được gợi ý thông qua API đến mô hình Recommendation khi tải trang.

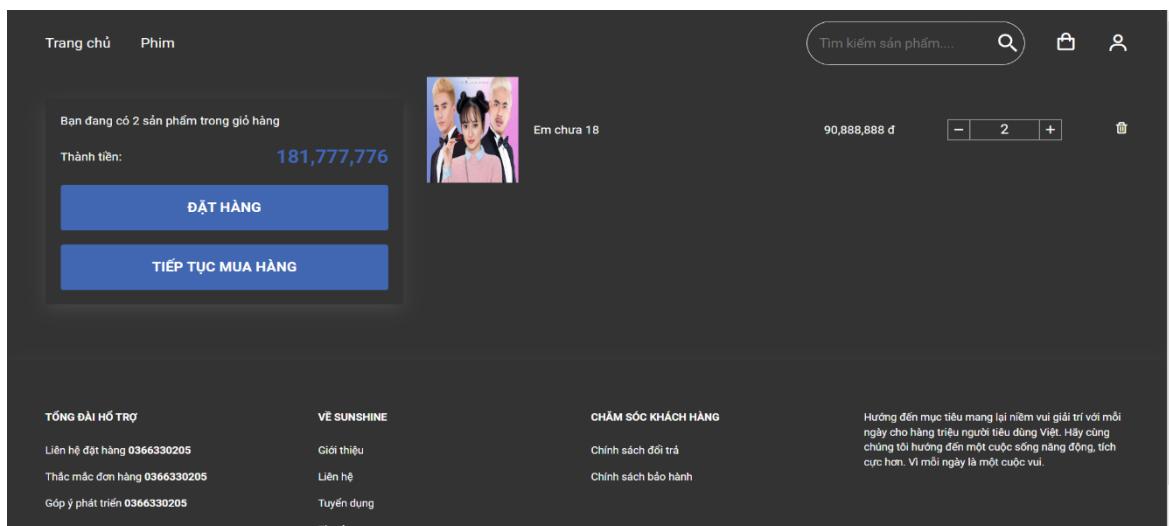
### 3.2.1.5. Giao diện trang thông tin cá nhân



**Hình 3-29 Giao diện thông tin khách**

Thể hiện tất cả lịch sử đơn hàng đã đặt và thông tin cá nhân cũng như địa chỉ để đặt hàng.

### 3.2.1.6. Giao diện giỏ hàng

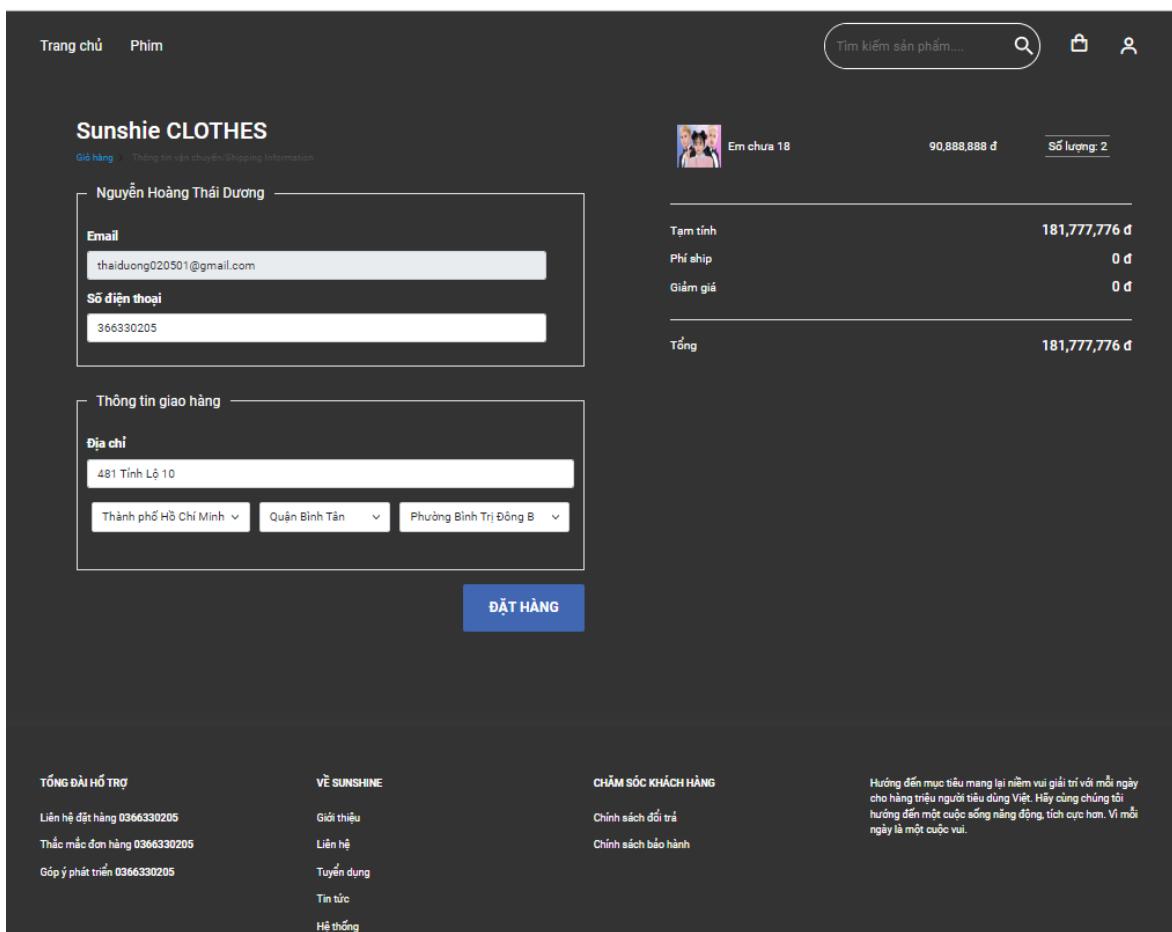


**Hình 3-28 Giao diện giỏ hàng**

Thể hiện tất cả thông tin phim đã được chọn và có thể tùy chỉnh số lượng.

Tính tạm giá tất cả các bộ phim có trong giỏ hàng để cho khách hàng có thể xem xét trước khi đặt hàng.

### 3.2.1.7. Giao diện đặt hàng



Hình 3-30 Giao diện đặt hàng

Thể hiện tất cả thông tin phim đã được chọn và không thể tùy chỉnh số lượng.

Tính tất cả các giá gồm phí giao hàng và giảm giá của sản phẩm nếu có.

Gồm các trường thông tin giao hàng như mail và số điện thoại.

Các thông tin như địa chỉ sẽ mặc định theo thông tin cá nhân khách hàng và có thể sửa tùy ý cho lần đặt hàng này.

### 3.2.1.8. Đăng nhập

The screenshot shows a login form titled "Đăng nhập". It includes fields for "Email" and "Mật khẩu" (Password), a blue "ĐĂNG NHẬP" (Login) button, and a link "Chưa có tài khoản? Đăng ký" (Don't have an account? Register). To the right, a dark sidebar displays the text "This beautiful theme yours!" and a quote from Admin: "Chúc quý khách hàng sẽ có những trải nghiệm tuyệt vời khi mua sắm ở đây."

Hình 3-31 Giao diện đăng nhập

Đăng nhập khách hàng và cả admin.

### 3.2.1.9. Đăng ký

The screenshot shows a registration form titled "Đăng ký". It includes fields for "Họ và tên" (Name), "Email", "Mật khẩu" (Password), and "Lặp lại Mật khẩu" (Re-enter Password), a blue "XÁC NHẬN ĐĂNG KÝ" (Confirm Registration) button, and a link "Đã có tài khoản? Đăng nhập" (Already have an account? Login). To the right, a dark sidebar displays the text "This beautiful theme yours!" and a quote from Admin: "Chúc quý khách hàng sẽ có những trải nghiệm tuyệt vời khi mua sắm ở đây."

Hình 3-32 Giao diện đăng ký

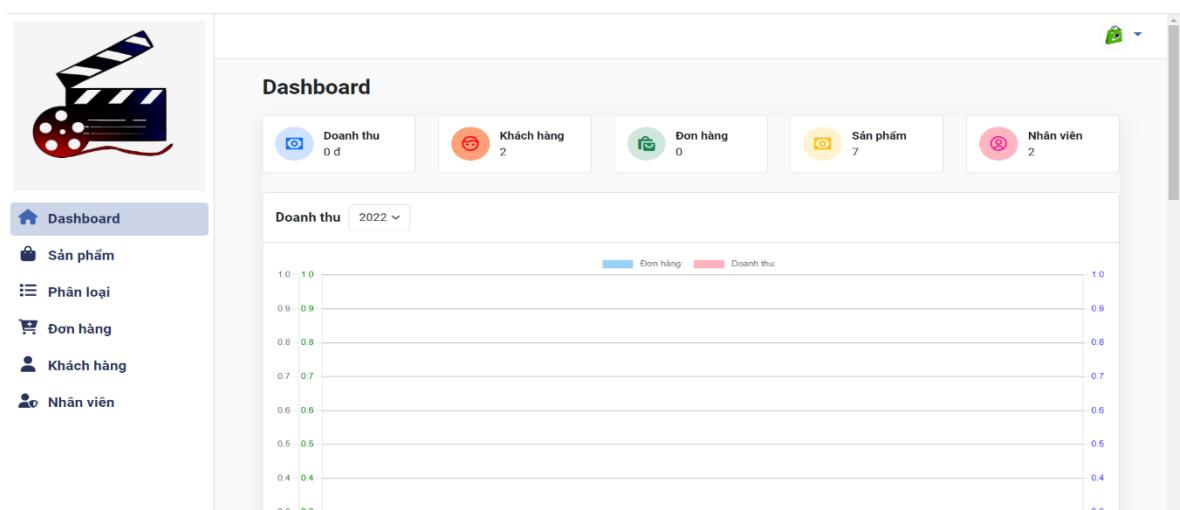
Chỉ được đăng ký khách hàng

### 3.2.2. Giao diện quản lý

#### 3.2.2.1. Dashboard

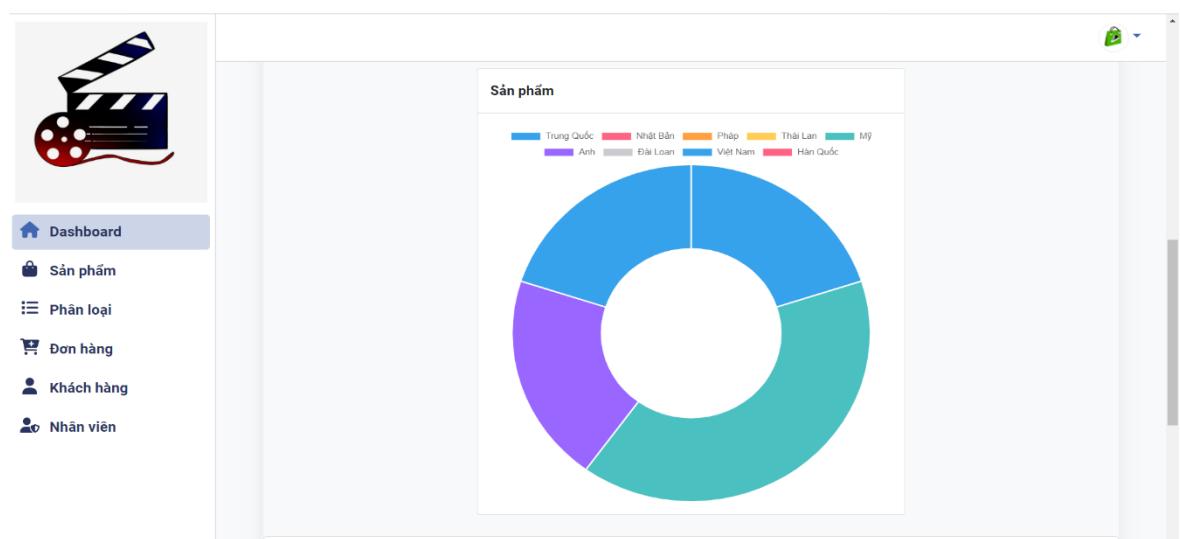
Gồm các thông kê cơ bản dữ liệu của cửa hàng và các đơn hàng mới nhất được đặt

##### a. Dashboard 1



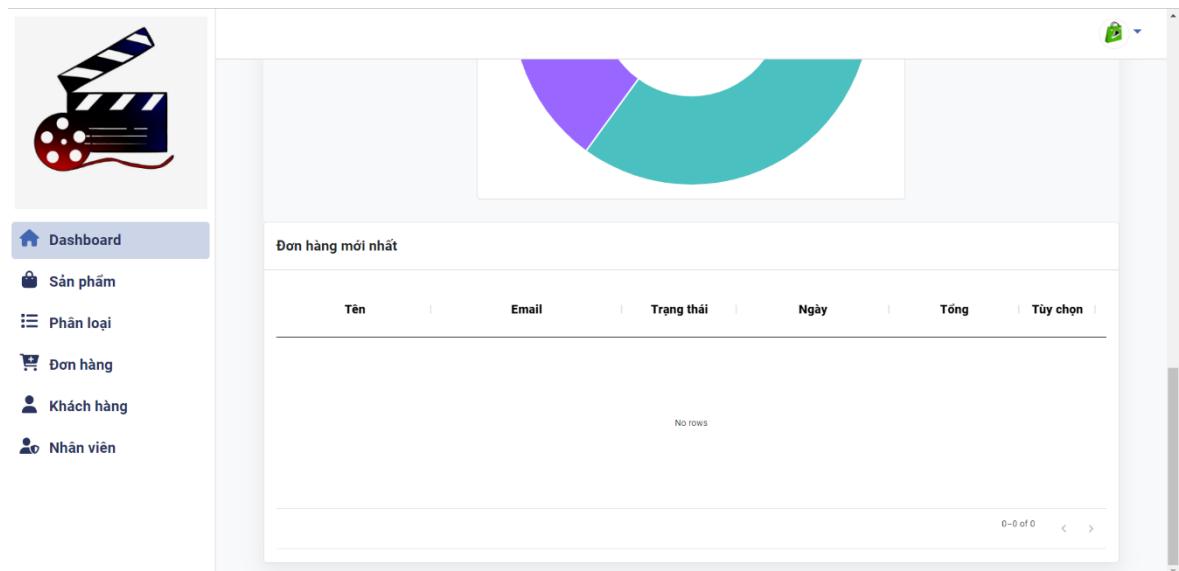
Hình 3-33 Dashboard 1

##### b. Dashboard 2



Hình 3-34 Dashboard 2

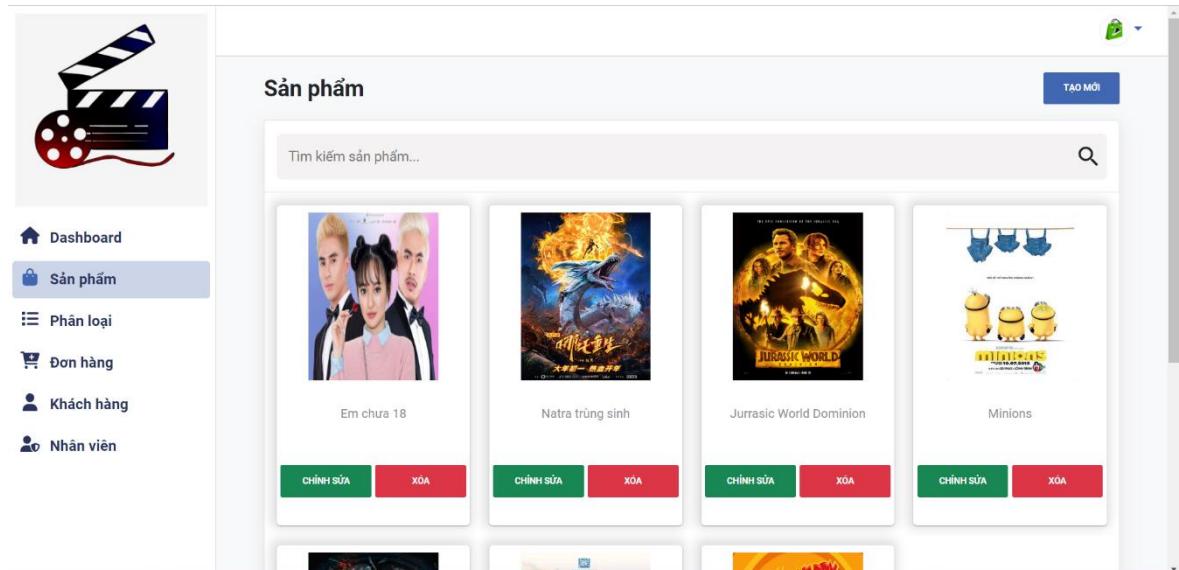
### c. Dashboard 3



Hình 3-35 Dashboard 3

#### 3.2.2.2. Giao diện sản phẩm

##### a. Giao diện sản phẩm

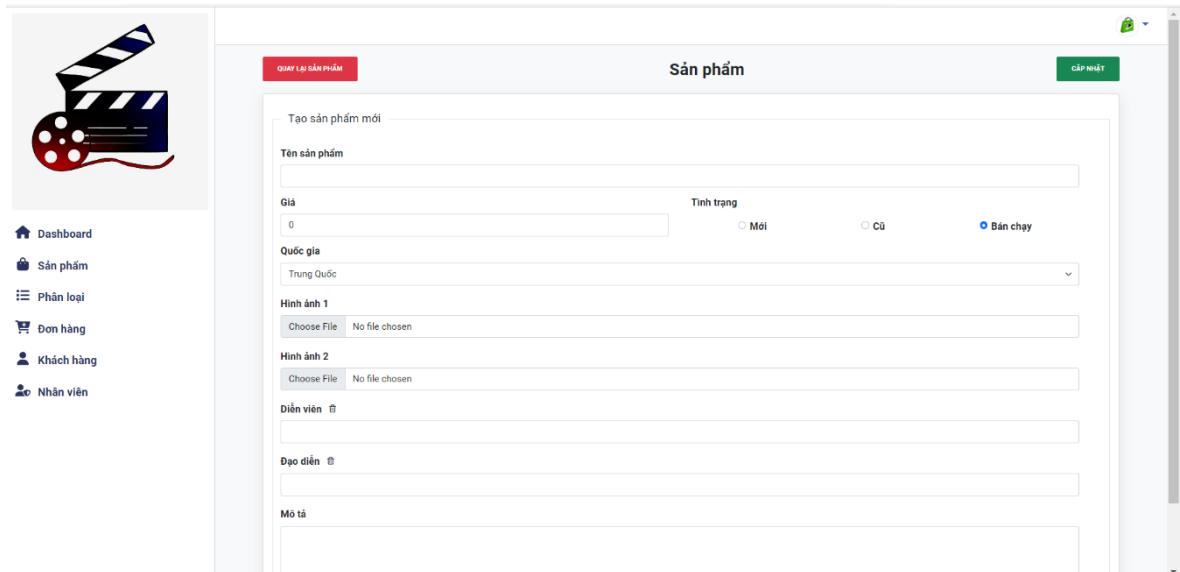


Hình 3-36 Giao diện sản phẩm

Tìm kiếm các bộ phim sản phẩm trong cửa hàng theo tên.

Có thể đi đến trang tạo mới sản phẩm hoặc sửa thông tin phim đó hoặc xóa phim đó.

## b. Giao diện tạo mới sản phẩm

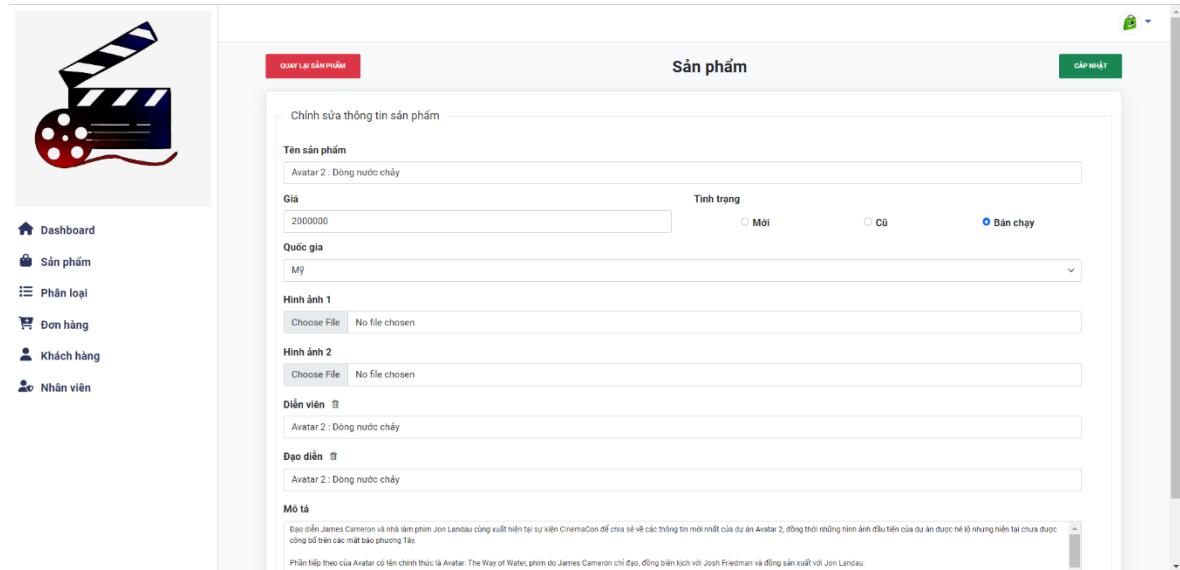


The screenshot shows the 'Tạo sản phẩm mới' (Create New Product) form. It includes fields for 'Tên sản phẩm' (Product Name), 'Giá' (Price) set to 0, 'Tình trạng' (Status) with radio buttons for 'Mới' (New), 'Cũ' (Used), and 'Bán chạy' (Best Seller) which is selected. There are also fields for 'Quốc gia' (Country) set to 'Trung Quốc' (China), 'Hình ảnh 1' (Image 1) and 'Hình ảnh 2' (Image 2) both showing 'Choose File No file chosen', 'Diễn viên' (Actor) with a placeholder 'Avatar 2 : Dòng nước chảy', 'Đạo diễn' (Director) with a placeholder 'Avatar 2 : Dòng nước chảy', and a large 'Mô tả' (Description) field containing text about James Cameron and Jon Landau.

Hình 3-37 Giao diện tạo mới sản phẩm phim

Giao diện này sẽ giống với chỉnh sửa về các trường nhưng khi mở lên sẽ có tiêu đề là “tạo mới” và các trường thông tin là trống.

## c. Giao diện sửa thông tin sản phẩm



The screenshot shows the 'Chỉnh sửa thông tin sản phẩm' (Edit Product Information) form. It includes fields for 'Tên sản phẩm' (Product Name) set to 'Avatar 2 : Dòng nước chảy', 'Giá' (Price) set to 2000000, 'Tình trạng' (Status) with radio buttons for 'Mới' (New), 'Cũ' (Used), and 'Bán chạy' (Best Seller) which is selected. There are also fields for 'Quốc gia' (Country) set to 'Mỹ' (USA), 'Hình ảnh 1' (Image 1) and 'Hình ảnh 2' (Image 2) both showing 'Choose File No file chosen', 'Diễn viên' (Actor) with a placeholder 'Avatar 2 : Dòng nước chảy', 'Đạo diễn' (Director) with a placeholder 'Avatar 2 : Dòng nước chảy', and a large 'Mô tả' (Description) field containing text about James Cameron and Jon Landau. A note at the bottom states: 'Đạo diễn James Cameron và nhà làm phim Jon Landau cũng xuất hiện tại sự kiện CinemaCon để chia sẻ về các thông tin mới nhất của dự án Avatar 2, đồng thời những hình ảnh đầu tiên của dự án được hé lộ nhưng hiện tại chưa được công bố trên các mặt báo phương Tây.' and 'Phản hồi thực của Avatar có tên chính thức là Avatar: The Way of Water; phim do James Cameron chỉ đạo, đồng biên kịch với Josh Friedman và đồng sản xuất với Jon Landau.'

Hình 3-38 Giao diện chỉnh sửa sản phẩm

Có thông tin và tiêu đề là “chỉnh sửa”

### 3.2.2.3. Giao diện phân loại

The screenshot shows a sidebar menu with icons for Dashboard, Sản phẩm, Phân loại (selected), Đơn hàng, Khách hàng, and Nhân viên. The main area is titled 'Phân loại' and contains a table with columns 'Tên' and 'Tùy chọn'. The table lists categories: Trung Quốc, Nhật Bản, Pháp, Thái Lan, and Mỹ. A search bar labeled 'Quốc gia' and a green 'CẤP NHẬT' button are also present.

Hình 3-39 Giao diện phân loại

Có thẻ xóa và chỉnh sửa sản phẩm bằng cách chọn các dòng trong bảng và thực hiện thao tác ấn nút thích hợp.

Tạo mới bằng cách để trống trường và nhập tên phân loại quốc gia chưa có hệ thống sẽ ghi nhận là tạo mới.

### 3.2.2.4. Giao diện đơn hàng

#### a. Giao diện đơn hàng

The screenshot shows a sidebar menu with icons for Dashboard, Sản phẩm, Phân loại, Đơn hàng (selected), Khách hàng, and Nhân viên. The main area is titled 'Đơn hàng' and displays a table with columns: Tên, Email, Trạng thái, Ngày, Tổng, and Tùy chọn. One row is visible showing Nguyễn Hoàng Thái Dương, thaiduong020501@gmail.com, đang chờ xử lý, 27-12-2022, 191,577,776, and a green eye icon.

Hình 3-40 Giao diện đơn hàng

Gồm danh sách các đơn hàng.

## b. Giao diện chi tiết đơn hàng

Chi tiết đơn hàng

QUAY LẠI ĐƠN HÀNG

27-12-2022  
Mã đơn hàng: cgfzglase

đang chờ xử lý

Khách hàng  
Nguyễn Hoàng Thái Dương  
thaiduong020501@gmail.com

Thông tin đơn  
Giao hàng nhanh  
Thanh toán: tiền mặt

Vận chuyển  
Địa chỉ: 481 Tỉnh Lộ 10, Phường Bình Trị Đông B, Quận Bình Tân, Thành phố Hồ Chí Minh

| Tạm tính | 0 đ      |
|----------|----------|
| Phi ship | 25,000 đ |
| Giảm giá | 0 đ      |
| Tổng     | 25,000 đ |

Hình 3-41 Giao diện chi tiết đơn hàng

Thao tác cập nhật trạng thái đơn hàng.

Thể hiện thông tin chi tiết đơn hàng.

### 3.2.2.5. Giao diện khách hàng

Khách hàng

Tìm kiếm khách hàng...

Tên  E-mail

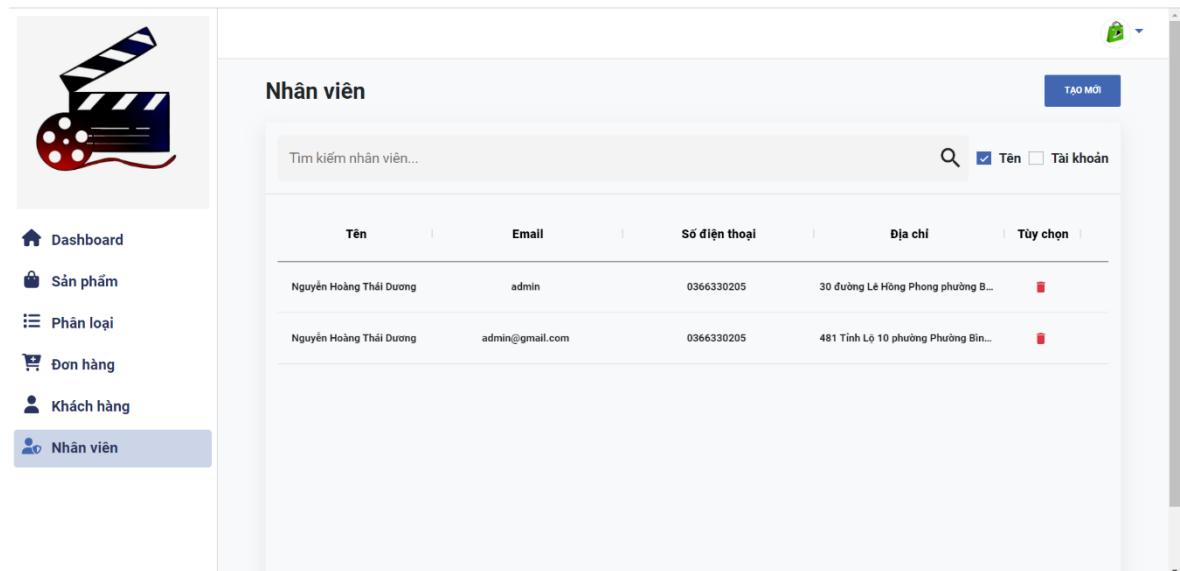
| Tên                     | Email                     | Số điện thoại | Địa chỉ                             | Tổng số đơn... | Tổng |
|-------------------------|---------------------------|---------------|-------------------------------------|----------------|------|
| Nguyễn Hoàng Thái Dương | thaiduong020502@gmail.com |               |                                     |                |      |
| Nguyễn Hoàng Thái Dương | thaiduong020501@gmail.com | 366330205     | 481 Tỉnh Lộ 10 phường Phường Bìn... |                |      |

Hình 3-42 Giao diện khách hàng

Thể hiện danh sách khách hàng và thông tin tổng số lượng đơn hàng và tổng tiền khách hàng đó

### 3.2.2.6. Giao diện nhân viên

#### a. Giao diện nhân viên



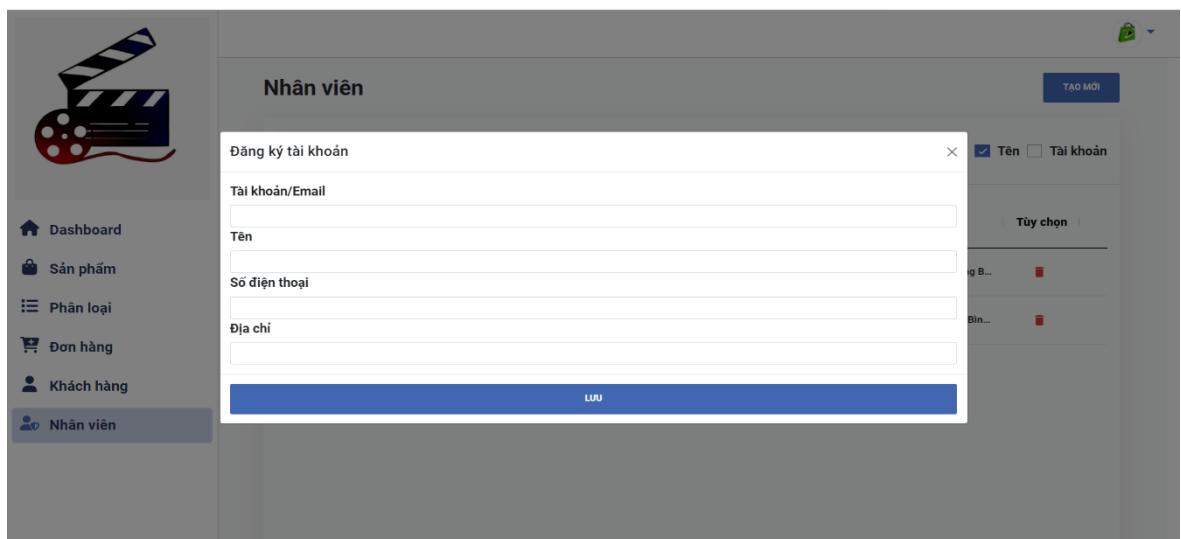
The screenshot shows a web-based application interface for managing staff. On the left is a sidebar with icons for Dashboard, Sản phẩm (Products), Phân loại (Category), Đơn hàng (Order), Khách hàng (Customer), and Nhân viên (Staff). The 'Nhân viên' icon is highlighted with a blue background. The main area is titled 'Nhân viên' and contains a search bar with placeholder text 'Tim kiếm nhân viên...' and filter checkboxes for 'Tên' (Name) and 'Tài khoản' (Account). Below the search is a table with columns: Tên (Name), Email, Số điện thoại (Phone Number), Địa chỉ (Address), and Tùy chọn (Options). Two rows of data are listed:

| Tên                     | Email           | Số điện thoại | Địa chỉ                              | Tùy chọn |
|-------------------------|-----------------|---------------|--------------------------------------|----------|
| Nguyễn Hoàng Thái Dương | admin           | 0366330205    | 30 đường Lê Hồng Phong phường B...   |          |
| Nguyễn Hoàng Thái Dương | admin@gmail.com | 0366330205    | 481 Tỉnh Lộ 10 phường Phường Bình... |          |

Hình 3-43 Giao diện nhân viên

Thể hiện danh sách nhân viên và Xóa nhân viên được chọn.

#### b. Giao diện đăng ký nhân viên



The screenshot shows the staff registration form. A modal window titled 'Nhân viên' is open, prompting for 'Đăng ký tài khoản'. The form fields include 'Tài khoản/Email' (Email), 'Tên' (Name), 'Số điện thoại' (Phone Number), and 'Địa chỉ' (Address). At the bottom of the form is a large blue 'LƯU' (Save) button. In the background, the staff list from the previous screenshot is visible.

Hình 3-44 Giao diện tạo nhân viên

Đăng ký thông tin nhân viên và mật khẩu được cung cấp thông qua bảo mật của cửa hàng.

## **Chương 4. KẾT LUẬN**

### **4.1. Kết quả đạt được**

Qua bài báo cáo trên, nhóm đã thực hiện được việc áp dụng các kiến thức mà mình tìm hiểu được về ReactJs – SpringBoot – Machine Learning - Flask để thiết kế một website bán hàng cơ bản mà vẫn đáp ứng được các chức năng thiết yếu.

Quá trình thiết kế cũng như tìm hiểu đã được ghi nhận lại cho các lập trình viên hay các bạn sinh viên muốn sử dụng tài liệu tham khảo cho các chủ đề liên quan.

### **4.2. Ưu điểm**

Tài liệu tham khảo về các Frameworks như ReactJs, Springboot và Flask phong phú giúp nhóm dễ dàng tiếp cận hơn cho việc xây dựng Đồ Án.

Website bán hàng là một trong những web có nhiều người sử dụng cũng như thiết kế nên dễ tham khảo.

Website có giao diện đơn giản thân thiện với người dùng cũng như Responsive trên đa thiết bị.

Sử dụng mô hình kiến trúc Client-Sever thông qua API nên phần thiết kế cũng như công việc các thành viên Front – Back được tách biệt và dễ dàng xây dựng.

### **4.3. Nhược điểm và giải pháp**

- Nhược điểm:

Machine Learning đang là lĩnh vực phát triển nhưng cũng đòi hỏi yêu cầu học thuật cao nên sẽ gặp khó khăn hơn trong việc tiếp cận

Việc sử dụng các Recommender System yêu cầu các tác vụ hiệu năng của thiết bị cao hơn so với bình thường.

Thiết kế database và dataset chưa có thể đồng bộ.

Chưa nắm sâu được các thuật toán mà mô hình Recommendation sử dụng do chỉ áp dụng các thuật toán từ các thư viện có sẵn.

- Giải pháp:

Tìm hiểu sâu nhiều hơn về các thuật toán cũng như các diễn đang về Machine Learning để có tối ưu hóa việc dự đoán.

Xử lý các nguồn dữ liệu và thử thêm vào cơ sở dữ liệu để có đồng bộ với nhau và chỉ thực hiện thao tác API thông qua một loại framework.

## TÀI LIỆU THAM KHẢO

[1]: Tổng quan về HTML-CSS-Javascript: [Lập trình HTML, CSS và JavaScript trong việc xây dựng website \(codegym.vn\)](#).

[2]: Redux: [https://www.youtube.com/watch?v=g\\_K1w8e0lLo&list=WL&index=3&t=2197s](https://www.youtube.com/watch?v=g_K1w8e0lLo&list=WL&index=3&t=2197s)

[3]: Lập trình ReactJs, HTML-CSS-Javascript: <https://fullstack.edu.vn/>

[4]: Tài liệu ReactJs:

**Giới thiệu ReactJs** <https://viblo.asia/p/gioi-thieu-ve-reactjs-phan-i-cac-khai-niem-co-ban-V3m5WzjblO7>

**Props và State:** <https://viblo.asia/p/su-khac-nhau-giua-props-va-state-trong-reactjs-OeVKBvrJKkW>

**ReactJs Docs:** <https://reactjs.org/docs/getting-started.html>

**Component Lifecycle:** <https://viblo.asia/p/lifecycle-component-trong-reactjs-gGJ59jzxKX2>

**ReactHook:** <https://viblo.asia/p/cung-tim-hieu-vecac-hook-trong-react-hooks-Ljy5VYgjlra>

[5]: React-Router: <https://reactrouter.com/docs/en/v6>

[6] Tài liệu Spring Boot:

**Trang tài liệu Spring :** <https://spring.io/projects/spring-boot>

**Video về Spring:**

[https://www.youtube.com/watch?v=O\\_XL9oQ1\\_To&t=947s](https://www.youtube.com/watch?v=O_XL9oQ1_To&t=947s)

**Hibernate:**<https://topdev.vn/blog/hibernate-la-gi-sao-phai-dung-no-thay-jdbc/>

**Spring Data:** [Giới thiệu về Spring Data JPA - Tai sao cần sử dụng nó? - Học Spring Boot \(hocspringboot.net\)](#)

[7] Tài liệu Flask:

**Flask Homepage:** [Welcome to Flask — Flask Documentation \(2.2.x\) \(palletsprojects.com\)](#)

**Flask, Flask API tham khảo thêm :** [Hướng dẫn lập trình Flask – Phần 1: Hello World! – Thai's Tech Blog \(thaitpham.com\)](#)

[8] Tài liệu Machine Learning:

**Recommendation system:** <https://viblo.asia/p/tong-quan-ve-recommender-system-recommender-system-co-ban-phan-1-924IJGBb5PM>

**Recommendation system:** <https://www.kaggle.com/code/ibtesama/getting-started-with-a-movie-recommendation-system>