

**TRƯỜNG ĐẠI HỌC CẦN THƠ  
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
KHOA CÔNG NGHỆ PHẦN MỀM**



**LUẬN VĂN TỐT NGHIỆP  
NGÀNH KỸ THUẬT PHẦN MỀM**

**Đề Tài  
PHÁT TRIỂN WEBSITE BÁN ĐỒ CHƠI CŨ SỬ DỤNG  
REACTJS VÀ NODEJS**

**Sinh Viên Thực Hiện:**

***Trần Lưu Vĩnh Khang***

***MSSV: B1906689***

***Khóa: 45***

Cần Thơ, 12/2023

**TRƯỜNG ĐẠI HỌC CẦN THƠ**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**KHOA CÔNG NGHỆ PHẦN MỀM**  
**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**  
**NGÀNH KỸ THUẬT PHẦN MỀM**

**Đề tài**  
**PHÁT TRIỂN WEBSITE BÁN ĐỒ CHƠI CŨ SỬ DỤNG**  
**REACTJS VÀ NODEJS**

**Giảng viên hướng dẫn :**  
*TS. Nguyễn Công Danh*

**Sinh viên thực hiện :**  
*Trần Lưu Vĩnh Khang*  
*MSSV: B1906689*  
*Khóa : 45*

Cần Thơ, 12/2023

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  
**KHOA CÔNG NGHỆ PHẦN MỀM**

**XÁC NHẬN CHỈNH SỬA LUẬN VĂN THEO YÊU CẦU CỦA**  
**HỘI ĐỒNG**

**Tên luận văn (tiếng Việt và tiếng Anh):** Phát triển website bán đồ chơi cũ sử dụng ReactJS và NodeJS (Developing an ecommerce website to sell second-hand toys using ReactJS and NodeJS)

**Họ tên sinh viên:** Trần Lưu Vĩnh Khang

**MASV:** B1906689

**Mã lớp:** DI1996A2

**Đã báo cáo tại hội đồng ngành:** Kỹ Thuật Phần Mềm

**Ngày báo cáo:** 08/12/2022

**Hội đồng báo cáo gồm:**

1. TS. Trương Minh Thái

Chủ tịch hội đồng

2. TS. Phan Phương Lan

Thành viên

3. TS. Nguyễn Công Danh

Thư ký

Luận văn đã được chỉnh sửa theo góp ý của Hội đồng.

Cần Thơ, ngày ..... tháng ..... năm 2022

**Giáo viên hướng dẫn**

(Ký và ghi họ tên)

## LỜI CẢM ƠN

Đầu tiên tôi xin cảm ơn thầy cô, gia đình và bạn bè đã tạo mọi điều kiện để giúp đỡ cho tôi vững bước trên con đường học tập. Xin cảm ơn cha mẹ và những người yêu thương nhất của tôi, những người đã tạo điều kiện tốt cho tôi để tôi có thể học tập, trau dồi kiến thức và là chỗ dựa vững chắc để tôi có thể vượt qua những khó khăn trong cuộc sống và hoàn thành tốt quá trình học tập của mình.

Để có thể hoàn thành xong bài luận văn tốt nghiệp này, tôi cũng xin gửi lời tri ân sâu sắc đến thầy Nguyễn Công Danh đã hết lòng giúp đỡ, góp ý và tận tình chỉ dạy cho tôi trong suốt khoảng thời gian hoàn thành luận văn tốt nghiệp của mình.

Tôi xin gửi lời cảm ơn đến tất cả các thầy cô Khoa Công Nghệ Thông Tin và Truyền Thông trường Đại học Cần Thơ, đặc biệt là những thầy cô thuộc bộ môn Công Nghệ Phần Mềm đã tận tình giảng dạy và truyền đạt những kiến thức bổ ích trong suốt quá trình học tập tại trường giúp tôi có được những kiến thức cần thiết để có thể hoàn thành bài luận văn cũng như những tri thức làm hành trang cho tôi vào đời. Bên cạnh đó, tôi cũng xin gửi lời cảm ơn đến tất cả những lời động viên và sự giúp đỡ nhiệt tình của bạn bè và tôi cũng cảm ơn hội đồng thầy Trương Minh Thái và cô Phan Phương Lan đến tham dự.

Mặc dù đã rất cố gắng tìm tòi, học hỏi và nỗ lực hết mình nhưng cũng không tránh khỏi những sai sót và hạn chế trong quá trình hoàn thành luận văn này. Chính vì vậy tôi rất mong nhận được sự giúp đỡ và chia sẻ của quý thầy cô để luận văn tốt nghiệp của tôi được hoàn thiện hơn. Cuối cùng tôi xin gửi đến thầy lời cảm ơn sâu sắc. Chúc thầy sức khỏe và thành công trong cuộc sống.

Cần Thơ, ngày ... tháng...năm 2023

Tác giả luận văn

**Trần Lưu Vĩnh Khang**

## This image shows a full page of primary-ruled paper. It features multiple horizontal rows of small black dots, evenly spaced across the entire page. These rows are designed to guide young learners as they practice writing letters and words. The background is white, and there are no margins or other markings present.

## Cán bộ phản biện

[illegible]

## Cán bộ phản biện

[illegible]

## Cán bộ phản biện

## MỤC LỤC

MỤC LỤC .....	i
DANH MỤC HÌNH ẢNH.....	iii
DANH MỤC BẢNG .....	v
KÝ HIỆU VÀ THUẬT NGỮ VIẾT TẮT.....	vii
TÓM TẮT.....	viii
ABSTRACT .....	ix
Phần 1: GIỚI THIỆU.....	1
1. Đặt vấn đề.....	1
2. Lịch sử giải quyết vấn đề.....	1
3. Mục tiêu đề tài .....	2
4. Đối tượng và phạm vi nghiên cứu .....	3
4.1. Đối tượng nghiên cứu.....	3
4.2. Phạm vi nghiên Cứu .....	3
4.3. Công cụ hỗ trợ phát triển hệ thống .....	3
4.4. Quy trình nghiên cứu.....	4
5. Những đóng góp chính của đề tài.....	4
Phần 2: NỘI DUNG .....	6
Chương 1: MÔ TẢ BÀI TOÁN .....	6
1.1. Mô tả hệ thống.....	6
1.2. Phân tích yêu cầu.....	7
1.3. Yêu cầu phi chức năng .....	26
1.4. Lựa chọn và đánh giá giải pháp.....	28
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	29
2.1. Tổng quan hệ thống .....	29
2.2. Giới thiệu về API và RESTful API .....	29
2.3. ReactJS .....	30
2.4. Javascript XML .....	31
2.5. Thư viện Redux toolkit.....	33



2.6. Thư viện Axios .....	34
2.7. Mongo.....	34
2.8. NodeJS.....	35
2.9. ExpressJS Framework .....	36
2.10. Mô hình MVC .....	38
2.11. Clouinary .....	39
CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP .....	40
3.1. Kiến trúc hệ thống .....	40
3.2. Tổng quan các lớp của mô hình MVC .....	42
3.3. Sơ đồ phân rã chức năng .....	45
3.4. Thiết kế dữ liệu.....	46
3.5. Thiết kế chức năng .....	51
Chương 4: KIỂM THỬ VÀ ĐÁNH GIÁ .....	79
4.1. Kiểm thử.....	79
4.2. Kế hoạch kiểm thử.....	79
4.3. Trách nhiệm và quyền hạn .....	80
4.4. Các rủi ro .....	80
4.5. Kịch bản kiểm thử .....	81
4.6. Đánh giá kết quả kiểm thử.....	81
4.7. Kết quả kiểm thử .....	84
4.8. Chức năng chưa kiểm thử.....	84
Phần 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	85
1.1. Kết quả đạt được.....	85
1.2. Hướng phát triển.....	85
PHỤ LỤC .....	86
PHỤ LỤC: HƯỚNG DẪN CÀI ĐẶT.....	86
TÀI LIỆU THAM KHẢO .....	89

## DANH MỤC HÌNH ẢNH

### Chương 1: MÔ TẢ BÀI TOÁN

Hình 1. 1 Use Case tổng quát .....	7
Hình 1. 2 Use Case Admin .....	8
Hình 1. 3 Use Case Khách hàng .....	9

### Chương 2: CƠ SỞ LÝ THUYẾT

Hình 2. 1 Phương thức RESTful API .....	30
Hình 2. 2 React .....	30
Hình 2. 3 Hình class components .....	32
Hình 2. 4 Hình Props và State .....	32
Hình 2. 5 Hình Redux .....	33
Hình 2. 6 Hình NodeJS .....	35
Hình 2. 7 Hình Express JS .....	36
Hình 2. 8 Hình Route parameters .....	37
Hình 2. 9 Hình mô hình MVC .....	38
Hình 2. 10 Hình Cloudinary .....	39

### Chương 3: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

Hình 3. 1 Hình sơ đồ thiết kế kiến trúc .....	40
Hình 3. 2 Sơ đồ kiến trúc MVC Web bán đồ chơi cũ .....	41
Hình 3. 3 Sơ đồ giao diện Admin và khách hàng .....	43
Hình 3. 4 Sơ đồ phân rã Admin .....	45
Hình 3. 5 Sơ đồ phân rã khách hàng .....	45
Hình 3. 6 Hình sơ đồ CDM mức quan niệm .....	46
Hình 3. 7 Hình thêm danh mục .....	51
Hình 3. 8 Sơ đồ xử lý Thêm danh mục .....	51
Hình 3. 9 Hình danh sách danh mục .....	52
Hình 3. 10 Hình sửa danh mục .....	52
Hình 3. 11 Hình thêm thương hiệu sản phẩm .....	53
Hình 3. 12 Sơ đồ xử lý Thêm thương hiệu .....	53
Hình 3. 13 Hình danh sách thương hiệu .....	54
Hình 3. 14 Hình sửa thương hiệu .....	54
Hình 3. 15 Hình thêm màu sản phẩm .....	54
Hình 3. 16 Sơ đồ xử lý thêm màu sản phẩm .....	55
Hình 3. 17 Hình danh mục màu .....	56
Hình 3. 18 Hình sửa màu .....	56
Hình 3. 19 Hình thêm sản phẩm .....	57

Hình 3. 20 Sơ đồ xử lý thêm sản phẩm .....	58
Hình 3. 21 Hình danh sách sản phẩm .....	59
Hình 3. 22 Hình sửa sản phẩm .....	60
Hình 3. 23 Hình quản lý đơn hàng .....	60
Hình 3. 24 Hình xem đơn hàng .....	61
Hình 3. 25 Hình thống kê thu nhập .....	62
Hình 3. 26 Thống kê đơn hàng .....	62
Hình 3. 27 Xem đơn hàng gần đây .....	63
Hình 3. 28 Quản lý xem tài khoản khách hàng .....	63
Hình 3. 29 Tạo tài khoản .....	64
Hình 3. 30 Sơ đồ xử lý đăng ký .....	65
Hình 3. 31 Hình đăng nhập .....	66
Hình 3. 32 Sơ đồ xử lý đăng nhập .....	67
Hình 3. 33 Đặt lại mật khẩu .....	68
Hình 3. 34 Sơ đồ xử lý quên mật khẩu .....	69
Hình 3. 35 Giao diện cập nhật thông tin .....	70
Hình 3. 36 Sơ đồ xử lý cập nhật thông tin .....	71
Hình 3. 37 Giao diện tìm kiếm .....	72
Hình 3. 38 Giao diện xem chi tiết sản phẩm .....	73
Hình 3. 39 Sơ đồ xử lý xem chi tiết đơn hàng .....	74
Hình 3. 40 Giao diện giỏ hàng .....	74
Hình 3. 41 Sơ đồ xử lý thêm giỏ hàng .....	75
Hình 3. 42 Giao diện nhập thông tin giao hàng .....	76
Hình 3. 43 Giao diện thanh toán .....	76
Hình 3. 44 Giao diện gmail cảm ơn khách hàng .....	77
Hình 3. 45 Sơ đồ xử lý từ thêm giỏ hàng thanh toán và gửi gmail .....	77
Hình 3. 46 Giao diện xem lại đơn hàng đã đặt của khách hàng .....	78

## **PHỤ LỤC**

Phụ lục 1: Hình website tải Visual Studio Code .....	86
Phụ lục 2: Hình thư mục chứa dự án .....	86
Phụ lục 3 Hình giao diện Visual Studio Code .....	87
Phụ lục 4 Hình mở terminal của Visual Studio Code .....	87
Phụ lục 5 Hình chạy code ở nhánh dev trong terminal .....	88
Phụ lục 6 Hình terminal khởi chạy dự án ở thư mục Font End .....	88

## DANH MỤC BẢNG

Bảng 1: Quy trình nghiên cứu .....	4
------------------------------------	---

### **Chương 1: MÔ TẢ BÀI TOÁN**

Bảng 1. 1 Bảng chức năng đăng ký .....	10
Bảng 1. 2 Bảng chức năng đăng nhập .....	11
Bảng 1. 3 Bảng chức năng quản lý thông tin cá nhân .....	12
Bảng 1. 4 Bảng chức năng Tìm kiếm sản phẩm.....	13
Bảng 1. 5 Bảng chức năng Lọc sản phẩm .....	14
Bảng 1. 6 Bảng chức năng xem chi tiết sản phẩm.....	15
Bảng 1. 7 Bảng chức năng xem giỏ hàng .....	16
Bảng 1. 8 Bảng chức năng đặt hàng .....	17
Bảng 1. 9 Bảng chức năng Thanh toán.....	18
Bảng 1. 10 Bảng chức năng xem thống kê .....	19
Bảng 1. 11 Bảng chức năng Quản lý sản phẩm.....	20
Bảng 1. 12 Bảng chức năng quản lý thương hiệu.....	21
Bảng 1. 13 Bảng chức năng quản lý danh mục .....	22
Bảng 1. 14 Bảng chức năng quản lý tình trạng sản phẩm .....	23
Bảng 1. 15 Bảng chức năng quản lý màu sản phẩm.....	24
Bảng 1. 16 Bảng chức năng Quản lý khách hàng.....	25
Bảng 1. 17 Bảng chức năng quản lý đơn hàng.....	26

### **Chương 2: CƠ SỞ LÝ THUYẾT**

Bảng 2. 1 Bảng phương thức HTTP .....	29
Bảng 2. 2 Bảng phương thức Response.....	38

### **Chương 3: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP**

Bảng 3. 1 Bảng ADMIN.....	47
Bảng 3. 2 Bảng PERSON .....	47
Bảng 3. 3 Bảng CUSTOMER .....	47
Bảng 3. 4 Bảng SHOPPING_CART .....	48
Bảng 3. 5 Bảng ITEM.....	48
Bảng 3. 6 Bảng ORDER.....	48
Bảng 3. 7 Bảng ORDER_STATUS.....	49
Bảng 3. 8 Bảng COMMENT .....	49
Bảng 3. 9 Bảng PRODUCT.....	49
Bảng 3. 10 Bảng IMAGES .....	50
Bảng 3. 11 Bảng VIDEO.....	50
Bảng 3. 12 Bảng thành phần đăng ký.....	64

Bảng 3. 13 Bảng thành phần cập nhật thông tin.....	70
<b>Chương 4: KIỂM THỬ VÀ ĐÁNH GIÁ</b>	
Bảng 4. 1 Trách nhiệm và quyền hạn .....	80
Bảng 4. 2 Các rủi ro.....	80
Bảng 4. 3 Kịch bản kiểm thử.....	81
Bảng 4. 4 Đánh giá kết quả kiểm thử .....	81
Bảng 4. 5 Chức năng đăng nhập.....	82
Bảng 4. 6 Chức năng đăng ký.....	82
Bảng 4. 7 Chức năng đặt hàng.....	83
Bảng 4. 8 Chức năng quản lý sản phẩm .....	83
Bảng 4. 9 Chức năng quản lý danh mục.....	83
Bảng 4. 10 Chức năng quản lý thương hiệu .....	84
Bảng 4. 11 Chức năng quản lý đơn hàng .....	84

## KÝ HIỆU VÀ THUẬT NGỮ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
1	Admin	Quản trị viên
2	CSS	Cascading Style Sheets
3	CSDL	Cơ sở dữ liệu
4	MVC	Model – View - Controller
5	Database	Cơ sở dữ liệu
6	Framwork	Các đoạn mã lệnh được viết sẵn, cấu thành nên bộ khung và các thư viện lập trình được đóng gói
7	Test Case	Trường hợp kiểm thử
8	UC/ UseCase	Trường hợp sử dụng
9	HTML	Hypertext Markup Language

## TÓM TẮT

**Bối cảnh:** Với sự bùng nổ của thương mại điện tử, việc xây phát triển các nền tảng trực tuyến để bán đồ chơi đã qua sử dụng trở nên ngày càng phổ biến. Tuy nhiên, việc phát triển website để bán lại đồ chơi đã qua sử dụng đặt ra những thách thức độc đáo, đặc biệt là khi phục vụ một đối tượng người dùng đa dạng và xử lý nhiều loại đồ chơi khác nhau.

**Mục tiêu:** Mục tiêu của luận văn của tôi là phát triển website thân thiện với người dùng, dành riêng cho việc bán lại đồ chơi đã qua sử dụng. Trong quá trình này, chúng tôi nhằm vượt qua những thách thức liên quan đến việc tạo ra một thị trường trực tuyến đảm bảo trải nghiệm mượt mà cho cả người bán và người mua. Ngoài ra, chúng tôi cố gắng tăng cường tính minh bạch của nền tảng bằng cách triển khai các biện pháp bảo mật mạnh mẽ và cung cấp quy trình giao dịch rõ ràng.

**Phương pháp:** Luận văn giới thiệu phương pháp của chúng tôi để đối mặt với những phức tạp của việc phát triển website để bán lại đồ chơi đã qua sử dụng. Chúng tôi nhấn mạnh sự quan trọng của trải nghiệm người dùng bằng cách sử dụng các nguyên tắc thiết kế trực quan và triển khai các tính năng tìm kiếm và phân loại hiệu quả. Biện pháp bảo mật, bao gồm các cổng thanh toán an toàn và xác thực người dùng, được tích hợp để phát triển một nền tảng đáng tin cậy. Phương pháp của chúng tôi bao gồm quá trình thử nghiệm và điều chỉnh lặp lại để đảm bảo môi trường trực tuyến mượt mà và đáng tin cậy. Hệ thống được phát triển trên hệ quản trị cơ sở dữ liệu MongoDB, và các công nghệ web như: ReactJS, NodeJS, Bootstrap,...

**Kết quả:** Kết quả sơ bộ từ giai đoạn bản mẫu và thử nghiệm chỉ ra những kết quả tích cực. Người dùng báo cáo trải nghiệm tích cực khi duyệt qua nền tảng, cho thấy rằng những xem xét về thiết kế của chúng tôi đã đóng góp vào một giao diện trực quan và thân thiện với người dùng. Ngoài ra, việc tích hợp phương thức thanh toán an toàn và các cơ chế xác thực đã chứng minh sự thành công trong việc tăng cường niềm tin của người dùng. Các đánh giá tiếp theo sẽ tiếp tục điều chỉnh và tối ưu hóa nền tảng.

## **ABSTRACT**

**Context:** With the explosion of e-commerce, establishing online platforms for reselling used toys has become increasingly prevalent. However, building a website for reselling used toys presents unique challenges, especially when catering to a diverse user base and dealing with various types of toys.

**Objective:** Our thesis aims to construct a user-friendly website specifically for the resale of used toys. Throughout this process, we aim to overcome challenges related to creating an online marketplace that ensures a smooth experience for both sellers and buyers. Additionally, we strive to enhance the platform's transparency by implementing robust security measures and providing clear transaction processes.

**Methodology:** The thesis introduces our methodology to tackle the complexities of building a website for reselling used toys. We emphasize the importance of user experience by employing principles of intuitive design and implementing efficient search and categorization features. Security measures, including secure payment gateways and user authentication, are integrated to build a trustworthy platform. Our approach involves iterative testing and adjustments to ensure a smooth and reliable online environment. The system is built on the MongoDB database management system, utilizing web technologies such as ReactJS, NodeJS, Bootstrap, etc.

**Results:** Preliminary results from the template phase and testing indicate positive outcomes. Users report a positive experience navigating the platform, indicating that our design considerations have contributed to an intuitive and user-friendly interface. Moreover, the integration of secure payment methods and verification mechanisms has proven successful in enhancing user trust. Further evaluations will continue to refine and optimize the platform.



## **Phần 1: GIỚI THIỆU**

### **1. Đặt vấn đề**

Ngày nay, với sự phát triển mạnh mẽ của khoa học – công nghệ, nó góp phần thúc đẩy sự tiến bộ của xã hội trong tất cả các lĩnh vực của đời sống xã hội. Đặc biệt với sự bùng nổ của công nghệ thông tin và internet thì tất cả các thông tin luôn được cập nhật, thay đổi một cách liên tục và đầy đủ thông qua mạng truyền thông và internet, mà phổ biến là thông qua các website.

Lợi ích của việc mua bán đồ chơi cũ: Thúc đẩy ý thức tái sử dụng và giúp tránh lãng phí tài nguyên. Những đồ chơi cũ thường được bỏ quên sau khi trẻ em không còn sử dụng, và việc bán lại chúng không chỉ giúp tạo thu nhập mà còn giúp giảm lượng đồ cần xử lý ở các bãi rác.

Khó khăn trong quá trình bán đồ chơi cũ: Việc vận chuyển, bảo quản và gửi tặng những món đồ này có thể là một thách thức. Một website giúp kết nối người bán và người mua có thể cải thiện tình hình này.

Góc nhìn về môi trường: Đồ chơi cũ, đặc biệt là những loại được làm từ các vật liệu không phân hủy, có thể gây ra ô nhiễm môi trường. Việc tìm cách tái sử dụng chúng có thể giúp giảm thiểu tác động tiêu cực này.

Tính tiện ích và cần thiết của website: Tạo ra một nền tảng mua bán đồ chơi cũ không chỉ giúp người dùng tiết kiệm tiền mà còn thời gian và nỗ lực trong việc tìm kiếm những món đồ cũ phù hợp.

Việc thiết kế website bán đồ chơi cũ có thể là một giải pháp để tận dụng và tái sử dụng những đồ chơi không còn sử dụng, đồng thời cung cấp một nền tảng thuận tiện cho việc trao đổi và mua bán giữa người dùng.

Do đó, tôi đã quyết định chọn đề tài “Website bán đồ chơi cũ” cho luận văn tốt nghiệp của mình với mục đích tạo ra một website bán đồ chơi cũ cho mọi người mua bán những món đồ chơi cũ với nhau, tránh gặp tình trạng kẻ gian lợi dụng gây thiệt hại về tài sản.

### **2. Lịch sử giải quyết vấn đề**

Trước Internet:

- Thương mại truyền thống: Trước khi Internet phổ biến, việc mua bán đồ cũ thường thông qua các cửa hàng địa phương, thị trường đồ rác, hoặc thông qua các quảng cáo trên báo, tạp chí. Sự ra đời của cộng đồng phi lợi nhuận: Các tổ chức phi lợi nhuận hoặc nhóm cộng đồng thường tổ chức các sự kiện trao đổi, buổi giao lưu để người dùng có thể trao đổi, tặng và mua bán đồ đạc cũ.

- Khi Internet phát triển:
- Sự xuất hiện của các trang web mua bán: Cùng với sự phát triển của Internet, nhiều trang web được tạo ra để hỗ trợ việc mua bán đồ dùng cũ như eBay, Craigslist, và các trang web phi lợi nhuận khác.
- Các nền tảng chuyên biệt: Nền tảng chuyên biệt dành riêng cho việc mua bán đồ chơi cũ hoặc đồ dùng gia đình cũ cũng đã xuất hiện để tối ưu hóa trải nghiệm người dùng trong việc mua bán và trao đổi.

Xu hướng hiện đại:

- Sự phát triển của ứng dụng di động: Việc có thể truy cập các nền tảng mua bán trên điện thoại di động đã tạo ra sự thuận tiện và nhanh chóng trong việc mua và bán hàng.
- Sự phát triển của nền tảng xã hội và cộng đồng: Các cộng đồng trực tuyến, nhóm Facebook, các diễn đàn chuyên ngành đã tạo ra cơ hội cho người dùng chia sẻ thông tin và tìm kiếm những món đồ cũ phù hợp.
- Tăng cường nhận thức về bảo vệ môi trường: Việc tái sử dụng và mua bán đồ cũ không chỉ là vấn đề về tiết kiệm, mà còn là vấn đề về bảo vệ môi trường. Điều này đã thúc đẩy sự phát triển của các nền tảng có mục tiêu xã hội, hướng tới việc giảm lượng đồ bỏ đi không cần thiết.
- Qua thời gian, việc giải quyết vấn đề này đã trải qua sự chuyển đổi từ các kênh truyền thống sang môi trường trực tuyến và cộng đồng, tạo ra nhiều cơ hội và thuận lợi hơn cho việc mua bán đồ dùng cũ và đồ chơi cũ.

### 3. Mục tiêu đề tài

Website bán đồ chơi cũ giúp chủ cửa hàng giới thiệu và bán sản phẩm nhanh chóng. Giúp người dùng có thể tiếp cận cửa hàng chỉ với thao tác trên thiết bị có kết nối internet mà không mất nhiều thời gian hay phải đến gặp trực tiếp của hàng.

- Mục tiêu chuyên môn: Sử dụng nền tảng NodeJS, framework ExpressJS và thư viện ReactJS để xây dựng ứng dụng.
- Sử dụng CSDL mongoDB và thư viện mongoose để lưu trữ dữ liệu

- Rèn luyện nâng cao kỹ năng quản lý dự án giúp quản lý thời gian một cách hiệu quả nhất
- Rèn luyện nâng cao kỹ năng viết tài liệu tư duy kỹ năng giải quyết vấn đề và lập trình.

## **4. Đối tượng và phạm vi nghiên cứu**

### **4.1. Đối tượng nghiên cứu**

- Đối tượng nghiên cứu về công nghệ Thư viện ReactJS
- Nền tảng NodeJS, framework Express JS Các ngôn ngữ như: JavaScript, CSS, HTML Mô hình REST API
- Cơ sở dữ liệu MongoDB và thư viện Mongoose
- Đối tượng nghiên cứu về sản phẩm:
- Các website kinh doanh chuyên về đồ chơi cũ

### **4.2. Phạm vi nghiên Cứu**

- Về mặt lý thuyết: Tìm hiểu về ngôn ngữ Javascript, nền tảng NodeJS, Framework ExpressJS thư viện React kết hợp với CSDL MongoDB
- Tìm hiểu cách tích hợp thanh toán trực tuyến thông qua cổng thanh toán Stripe.
- Về mặt kỹ thuật:
- Back-end: sử dụng nền tảng nodeJS với framework ExpressJS để xử lý dữ liệu từ client
- Front End: sử dụng: HTML, CSS, và thư viện ReactJS để xây dựng giao diện Database: sử dụng MongoDB để lưu trữ dữ liệu.

### **4.3. Công cụ hỗ trợ phát triển hệ thống**

- Công cụ xây dựng và quản lý CSDL: MongoDB Compass,
- Công cụ hỗ trợ lập trình: Visual Studio Code,
- Công cụ hỗ trợ thực thi: Google Chrome,
- Công cụ kiểm thử API Back-End: Postman
- Công cụ hỗ trợ viết tài liệu: Microsoft Word,
- Công cụ hỗ trợ vẽ sơ đồ: Draw.io.

#### 4.4. Quy trình nghiên cứu

STT	Công việc	Tuần															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Nghiên cứu đề tài																
2	Chuẩn bị, tìm hiểu công nghệ																
3	Mô tả chi tiết hệ thống																
4	Viết tài liệu đặc tả																
5	Viết sơ bộ tài liệu thiết kế																
6	Viết chương trình demo																
7	Hoàn thành tài liệu thiết kế																
8	Kiểm thử và sửa lỗi																
9	Hoàn thành chương trình demo																
10	Hoàn thành quyền báo cáo																

Bảng 1: Quy trình nghiên cứu

#### 5. Những đóng góp chính của đề tài

Phát triển một website thương mại điện tử ở Việt Nam chuyên về các loại đồ chơi cũ giúp cho người dùng có thể tìm kiếm và lựa chọn cho mình sản phẩm phù hợp ở bất kì đâu chỉ cần một thiết bị có kết nối internet. Quá trình thanh toán dễ dàng tiếp cận với người dùng.

Tiết kiệm chi phí hoạt động, sản xuất kinh doanh, không cần bỏ số tiền quá lớn để thuê mặt bằng cho cửa hàng, doanh nghiệp tiết kiệm thời gian của việc mua và bán giữa khách hàng và cửa hàng.

Tăng tính trải nghiệm của khách hàng khi mà họ có thể xem hình ảnh chi tiết sản phẩm dưới dạng zoom ảnh với xem video về món hàng nâng cao lợi nhuận cho người bán và giúp người mua có được những món đồ chơi cũ dễ dàng và tiện lợi.

## **Phần 2: NỘI DUNG**

### **Chương 1: MÔ TẢ BÀI TOÁN**

Trong chương này trình bày các yêu cầu của website, mô tả chi tiết bài toán, đặt vấn đề, các yêu cầu giao tiếp bên ngoài, các yêu cầu phi chức năng, chức năng chính của hệ thống và cuối cùng là sơ đồ usecase của hệ thống.

#### **1.1. Mô tả hệ thống**

Website tạo ra với mục đích giúp cho cửa hàng bán đồ chơi cũ và khách hàng có thể thực hiện mua bán một cách thuận tiện nhanh gọn hơn. Cung cấp một website nơi mà khách hàng có thể thoải mái chọn lựa sản phẩm và đặt mua một cách dễ dàng.

Website sẽ gồm 2 phần là giao diện quản lý và giao diện người dùng. Có 3 nhóm người dùng là khách vãng lai, khách hàng (là người đã có tài khoản) và người quản lý. Giao diện quản lý được dành riêng cho người quản lý, giao diện người dùng dành cho khách vãng lai và khách hàng. Khách vãng lai sau khi đăng nhập sẽ trở thành khách hàng, khách hàng có thể thực hiện tất cả các chức năng khách vãng lai.

Khách vãng lai là những khách hàng chưa đăng ký tài khoản trên website, khách vãng lai có thể đăng ký tài khoản bằng cách nhập các thông tin cần thiết. Khách vãng lai có thể thực hiện các chức năng xem sản phẩm theo danh mục, tìm kiếm sản phẩm theo tên, lọc sản phẩm. Chức năng xem sản phẩm theo danh mục cho phép xem tất cả sản phẩm thuộc danh mục đó. chức năng xem chi tiết sản phẩm có thể xem được thông tin chi tiết của sản phẩm như nhóm sản phẩm, số lượng, thương hiệu, ảnh sản phẩm, mô tả chi tiết sản phẩm và danh sách đánh giá sản phẩm nhưng không thể thực hiện chức năng đánh giá. Ngoài ra người dùng có thể tìm kiếm sản phẩm theo tên.

Khách hàng là người đã có tài khoản trên website, khách hàng có thể đăng nhập thông qua email và mật khẩu. Khách hàng có thể thực hiện tất cả các chức năng của khách vãng lai. Ngoài ra khách hàng có thể thực hiện các chức năng khác như đánh giá sản phẩm, thanh toán, quản lý tài khoản, xem lịch sử đơn hàng,.. Chức năng đánh giá sản phẩm cho phép người dùng chấm sao cho sản phẩm ( sao từ 1 đến 5) và góp ý về sản phẩm. Chức năng thanh toán trong quá trình thanh toán người dùng cần nhập thông tin nhận hàng như số điện thoại, địa chỉ nhận hàng vào biểu mẫu. Có 2 hình thức thanh toán là thanh toán trực tiếp. Khách hàng có xem và thay đổi thông tin cá nhân khi chọn quản lý tài khoản, ngoài ra khách hàng có thể thay đổi mật khẩu.

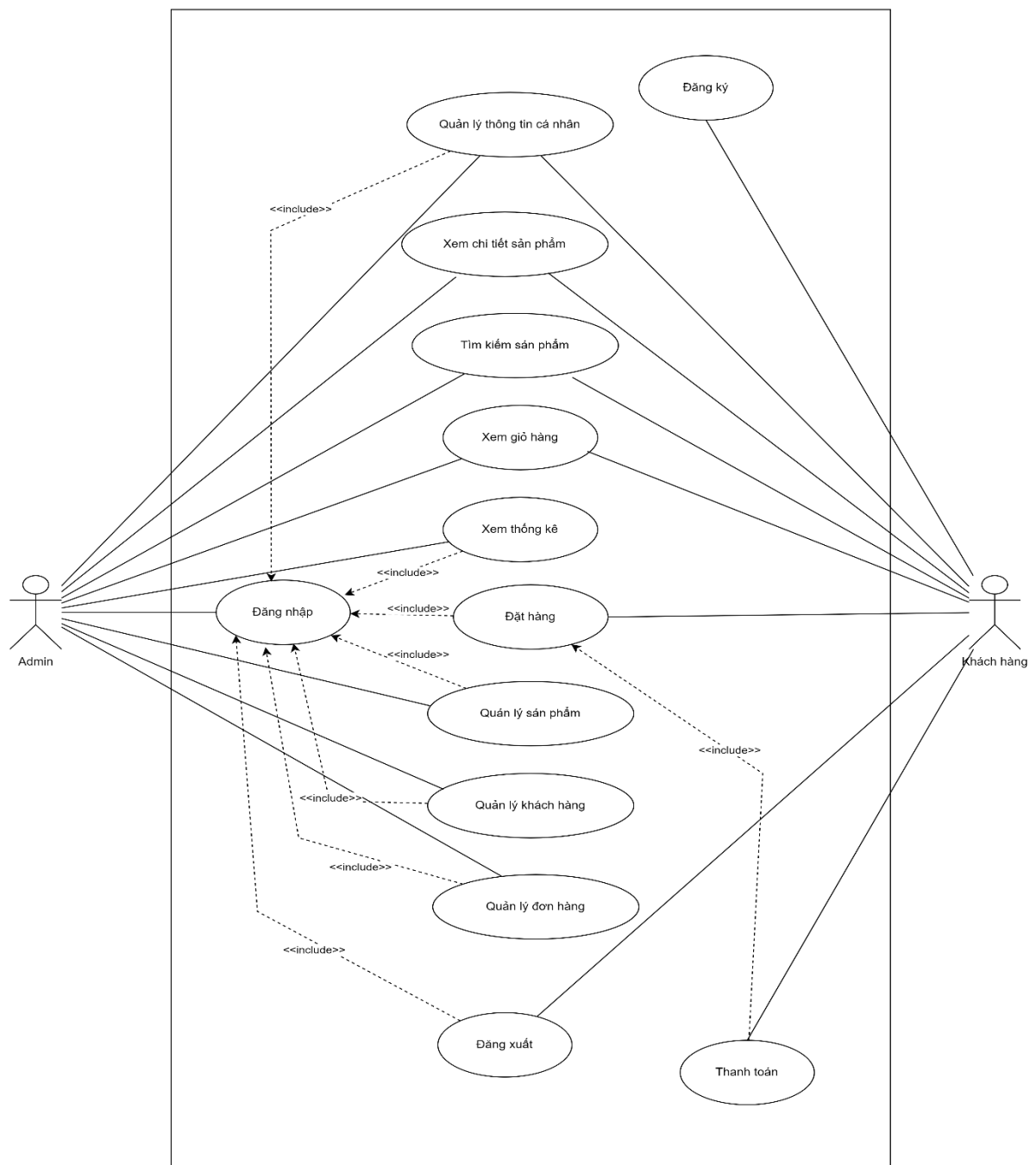
Người quản lý đóng vai trò như chủ cửa hàng được cấp tài khoản cho trước. Người quản lý đăng nhập vào giao diện quản lý bằng tài khoản đã cấp, giao diện trang chủ quản lý sẽ xuất hiện các phần như danh mục các chức năng, các bảng thống kê như đơn hàng, tổng doanh thu. Người quản lý được cung cấp các chức năng quản lý như quản lý

tài khoản người dùng, quản lý sản phẩm.

## 1.2. Phân tích yêu cầu

### 1.2.1. Sơ đồ Use Case

Hình 1:3 mô tả sơ đồ usecase của hệ thống với 2 tác nhân chính là admin (người quản trị) và khách hàng

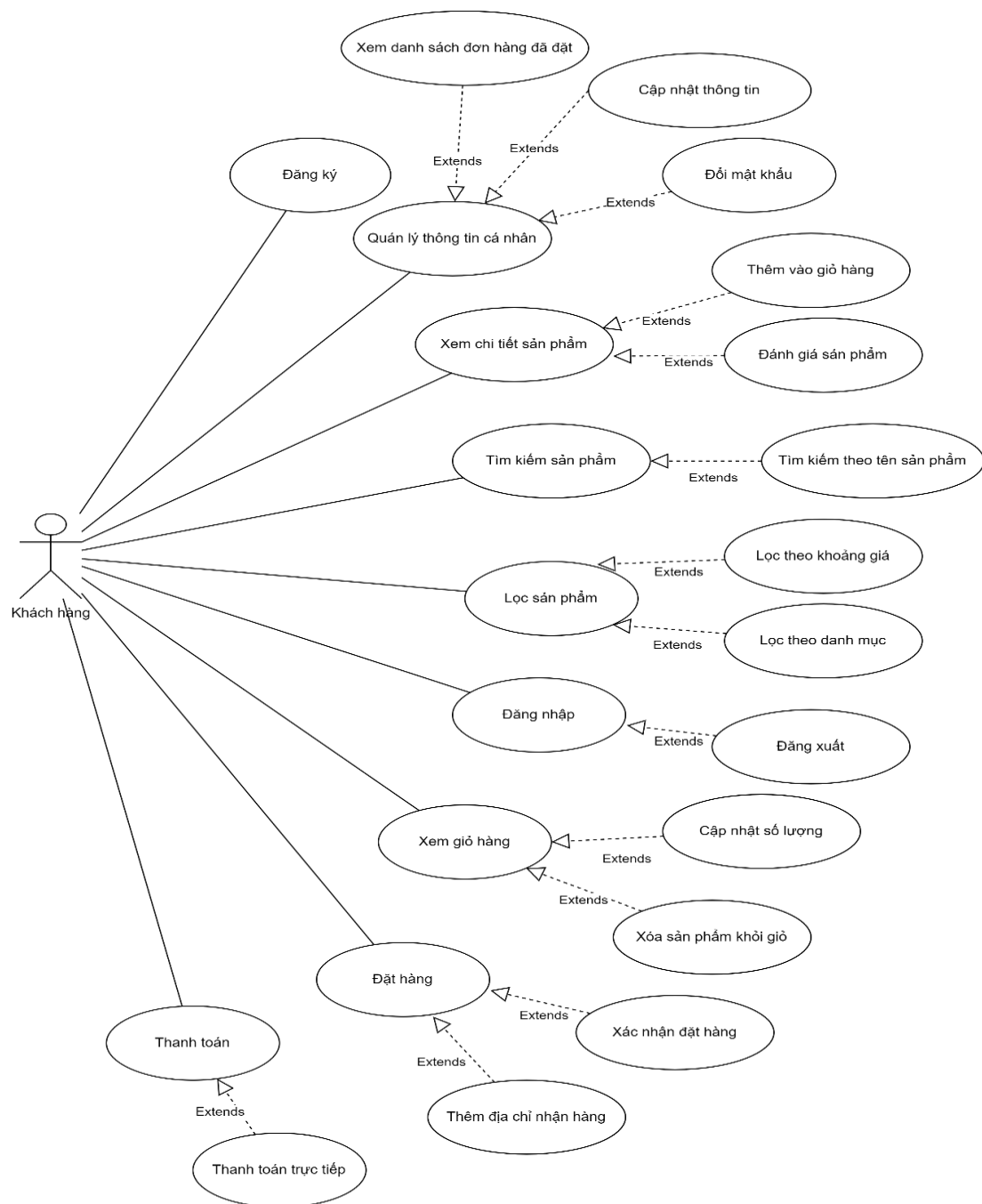


Hình 1. 1 Use Case tổng quát





Hình 1: 4 mô tả sơ đồ use case [1] với tác nhân chính là khách hàng, với các chức năng: đăng nhập, đăng ký, quản lý thông tin cá nhân, xem chi tiết sản phẩm, tìm kiếm sản phẩm, lọc sản phẩm, xem giỏ hàng, đặt hàng, thanh toán.



Hình 1. 3 Use Case Khách hàng

### 1.2.2. Yêu cầu chức năng

#### - Chức năng đăng ký

*Bảng 1. 1 Bảng chức năng đăng ký*

Usecase	UC_001_Đăng ký
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Bắt buộc
Các thành phần tham gia và mối quan tâm:	Người dùng muốn đăng ký vào hệ thống
Mô tả tóm tắt:	Sử dụng thông tin cá nhân để tạo tài khoản người dùng
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"><li>+ Association (kết hợp): Khách hàng</li><li>+ Include (bao gồm):</li><li>+ Extend (mở rộng):</li><li>+ Generalization (tổng quát hóa):</li></ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"><li>1. Truy cập vào trang chủ của website</li><li>2. Nhấn vào biểu tượng tài khoản ở góc phải</li><li>3. Điền thông tin cần thiết</li><li>4. Nhấn nút đăng ký</li></ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

## - Chức năng đăng nhập

*Bảng 1. 2 Bảng chức năng đăng nhập*

Usecase	UC_002_Đăng nhập
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Bắt buộc
Các thành phần tham gia và mối quan tâm:	Admin và người dùng muốn đăng nhập vào hệ thống
Mô tả tóm tắt:	Sử dụng tài khoản cá nhân để đăng nhập vào hệ thống
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Admin, Khách hàng</li> <li>+ Include (bao gồm): Đánh giá giá sản phẩm, đặt hàng, xem thống kê, quản lý ( sản phẩm, thương hiệu, danh mục, màu, khách hàng, tình trạng, đơn hàng)</li> <li>+ Extend (mở rộng): Đăng xuất</li> <li>+ Generalization (tổng quát hóa):</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Nhấp vào biểu tượng đăng nhập ở trang chủ</li> <li>2. Nhập tin gồm email và mật khẩu</li> <li>3. Nhấn nút đăng nhập</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

- **Chức năng Quản lý thông tin cá nhân**

*Bảng 1. 3 Bảng chức năng quản lý thông tin cá nhân*

Usecase	UC_003_Quản lý thông tin cá nhân
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Bắt buộc
Các thành phần tham gia và mối quan tâm:	Người dùng muốn quản lý thông tin cá nhân
Mô tả tóm tắt:	Người dùng có thể cập nhật thông tin cá nhân và xem danh sách đơn hàng
Trigger kiểu sự kiện:	External
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Khách hàng</li> <li>+ Include (bao gồm):</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Đổi mật khẩu, cập nhật thông tin. Xem danh sách đơn hàng.</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Người dùng đăng nhập vào hệ thống</li> <li>2. Nhấn Người dùng nhấp vào tên của mình ở góc trái quản lý tài khoản</li> <li>3. Danh mục quản lý hiện ra</li> <li>4. Người dùng nhấp vào biểu tượng quản lý thông tin người dùng tương ứng</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

- **Chức năng tìm kiếm sản phẩm**

*Bảng 1. 4 Bảng chức năng Tìm kiếm sản phẩm*

Usecase	UC_004_Tìm kiếm sản phẩm
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Trung bình
Các thành phần tham gia và mối quan tâm:	Người dùng muốn tìm kiếm sản phẩm
Mô tả tóm tắt:	Người dùng có thể tìm kiếm theo tên sản phẩm
Trigger kiểu sự kiện:	External
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Khách hàng</li> <li>+ Include (bao gồm):</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Tìm kiếm theo tên sản phẩm</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Người dùng truy cập vào website</li> <li>2. Nhấn vào biểu tượng tìm kiếm trên web</li> <li>3. Nhập thông tin cần tìm kiếm</li> <li>4. Nhấn nút tìm</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

## - Chức năng Lọc sản phẩm

*Bảng 1. 5 Bảng chức năng Lọc sản phẩm*

Usecase	UC_005_Lọc sản phẩm
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Trung bình
Các thành phần tham gia và mối quan tâm:	Người dùng muốn lọc sản phẩm
Mô tả tóm tắt:	Người dùng có thể lọc sản phẩm theo thương hiệu, kích cỡ khoảng giá
Trigger kiểu sự kiện:	External
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Khách hàng</li> <li>+ Include (bao gồm):</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Lọc theo thương hiệu, lọc theo khoảng giá.</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Người dùng truy cập vào website</li> <li>2. Danh mục sản phẩm</li> <li>3. Nhấn chọn vào danh mục muốn lọc</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

## - Chức năng Xem chi tiết sản phẩm

*Bảng 1. 6 Bảng chức năng xem chi tiết sản phẩm*

Usecase	UC_006_Xem chi tiết sản phẩm
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Cao
Các thành phần tham gia và mối quan tâm:	Khách hàng muốn xem thông tin chi tiết sản phẩm
Mô tả tóm tắt:	Người dùng có thể xem thông tin sản phẩm một cách chi tiết
Trigger kiểu sự kiện:	External
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Khách hàng</li> <li>+ Include (bao gồm):</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Thêm vào giỏ hàng, đánh giá sản phẩm</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Khách hàng truy cập vào trang web</li> <li>2. Khách hàng tìm kiếm sản phẩm hoặc nhấn vào con mắt để xem chi tiết sản phẩm</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

- **Chức năng Xem giỏ hàng**

*Bảng 1. 7 Bảng chức năng xem giỏ hàng*

Usecase	UC_007_Xem giỏ hàng
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Cao
Các thành phần tham gia và mối quan tâm:	Khách hàng muốn xem sản phẩm trong giỏ hàng
Mô tả tóm tắt:	Khách hàng có thể truy cập và quản lý thông tin đơn hàng trong giỏ hàng
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Khách hàng</li> <li>+ Include (bao gồm):</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Cập nhật số lượng, xóa sản phẩm, đặt hàng</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Người dùng truy cập vào website</li> <li>2. Người dùng nhấp vào biểu tượng giỏ hàng bên trên góc phải</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	



## - Chức năng đặt hàng

*Bảng 1. 8 Bảng chức năng đặt hàng*

Usecase	UC_008_Đặt hàng
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Cao
Các thành phần tham gia và mối quan tâm:	Khách hàng muốn đặt hàng
Mô tả tóm tắt:	Khách hàng có thể đặt hàng
Trigger kiểu sự kiện:	External
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Khách hàng</li> <li>+ Include (bao gồm): Đăng nhập</li> <li>+ Extend (mở rộng): Xem giỏ hàng</li> <li>+ Generalization (tổng quát hóa): Thêm địa chỉ, xác nhận đơn đặt hàng</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Khách hàng đăng nhập vào website</li> <li>2. Khách hàng bấm vào biểu tượng giỏ hàng</li> <li>3. Khách hàng nhấn nút đặt hàng</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

**- Chức năng thanh toán**

*Bảng 1. 9 Bảng chức năng Thanh toán*

Usecase	UC_009_Thanh toán
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Bắt buộc
Các thành phần tham gia và mối quan tâm:	Khách hàng muốn thanh toán sản phẩm trong giỏ hàng
Mô tả tóm tắt:	Khách hàng có thể thanh toán qua phương thức
Trigger kiểu sự kiện:	External
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Khách hàng</li> <li>+ Include (bao gồm): Đặt hàng</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Thanh toán khi nhận hàng.</li> </ul>
Luồng xử lý bình thường của sự kiện	Khách hàng đăng nhập vào web site Khách hàng bấm vào giỏ hàng Khách hàng bấm vào nút đặt hàng Khách hàng chọn phương thức thanh toán
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

- **Chức năng xem thống kê**

*Bảng 1. 10 Bảng chức năng xem thống kê*

Usecase	UC_010_Xem thống kê
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Bắt buộc
Các thành phần tham gia và mối quan tâm:	Admin muốn xem thống kê sản phẩm
Mô tả tóm tắt:	Admin muốn xem thống kê sản phẩm và số đơn đã bán
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Admin</li> <li>+ Include (bao gồm): Đăng nhập</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Thống kê theo thời gian</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Admin đăng nhập vào hệ thống bằng tài khoản được cấp</li> <li>2. Admin xem thống kê tại giao diện trang chủ</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

## - Chức năng Quản lý sản phẩm

*Bảng 1. 11 Bảng chức năng Quản lý sản phẩm*

Usecase	UC_011_Quản lý sản phẩm
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Bắt buộc
Các thành phần tham gia và mối quan tâm:	Admin muốn quản lý sản phẩm
Mô tả tóm tắt:	Admin có thể quản lý chi tiết từng sản phẩm
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Admin</li> <li>+ Include (bao gồm): Đăng nhập</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Xem danh sách sản phẩm, xóa sản phẩm, cập nhật sản phẩm, thêm sản phẩm.</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Admin đăng nhập vào website</li> <li>2. Admin nhấn vào danh mục quản lý sản phẩm bên trái</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

- **Chức năng quản lý thương hiệu**

*Bảng 1. 12 Bảng chức năng quản lý thương hiệu*

Usecase	UC_012_Quản lý thương hiệu
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Bắt buộc
Các thành phần tham gia và mối quan tâm:	Admin muốn quản lý thương hiệu
Mô tả tóm tắt:	Admin có thể quản lý chi tiết từng thương hiệu
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Admin</li> <li>+ Include (bao gồm): Đăng nhập</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Xem danh sách thương hiệu, xóa thương hiệu, cập nhật thương hiệu, thêm thương hiệu.</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Admin đăng nhập vào website</li> <li>2. Admin nhấn vào danh mục quản lý thương hiệu bên trái</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

- **Chức năng quản lý danh mục**

*Bảng 1. 13 Bảng chức năng quản lý danh mục*

Usecase	UC_013_Quản lý danh mục
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Bắt buộc
Các thành phần tham gia và mối quan tâm:	Admin muốn quản lý danh mục
Mô tả tóm tắt:	Admin có thể quản lý chi tiết từng danh mục
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Admin</li> <li>+ Include (bao gồm): Đăng nhập</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Xem danh sách danh mục, xóa danh mục, cập nhật danh mục, thêm danh mục.</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Admin đăng nhập vào website</li> <li>2. Admin nhấn vào danh mục quản lý danh mục bên trái</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

- **Chức năng quản lý tình trạng sản phẩm**

*Bảng 1. 14 Bảng chức năng quản lý tình trạng sản phẩm*

Usecase	UC_014_Quản lý tình trạng sản phẩm
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Cao
Các thành phần tham gia và mối quan tâm:	Admin muốn quản lý tình trạng sản phẩm
Mô tả tóm tắt:	Admin có thể quản lý chi tiết tình trạng sản phẩm
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Admin</li> <li>+ Include (bao gồm): Đăng nhập</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Xem danh sách tình trạng sản phẩm, xóa tình trạng sản phẩm, cập nhật tình trạng sản phẩm, thêm tình trạng sản phẩm.</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Admin đăng nhập vào website</li> <li>2. Admin nhấn vào danh mục quản lý tình trạng sản phẩm bên trái</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

**- Chức năng quản lý màu sản phẩm**

*Bảng 1. 15 Bảng chức năng quản lý màu sản phẩm*

Usecase	UC_015_Quản lý màu sản phẩm
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Cao
Các thành phần tham gia và mối quan tâm:	Admin muốn quản lý màu sản phẩm
Mô tả tóm tắt:	Admin có thể quản lý chi tiết màu sản phẩm
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Admin</li> <li>+ Include (bao gồm): Đăng nhập</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Xem danh sách màu, xóa màu, cập nhật màu, thêm màu.</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Admin đăng nhập vào website</li> <li>2. Admin nhấn vào danh mục quản lý màu bên trái</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	



- **Chức năng Quản lý khách hàng**

*Bảng 1. 16 Bảng chức năng Quản lý khách hàng*

Usecase	UC_015_Quản lý khách hàng
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Bắt buộc
Các thành phần tham gia và mối quan tâm:	Admin muốn quản lý khách hàng đã đăng nhập vào hệ thống
Mô tả tóm tắt:	Admin có thể quản lý chi tiết từng tài khoản khách hàng đã đăng nhập vào hệ thống
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Admin</li> <li>+ Include (bao gồm): Đăng nhập</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Xem danh sách khách hàng</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>3. Admin đăng nhập vào website</li> <li>4. Admin nhấn vào danh mục quản lý khách hàng bên trái</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

## - Chức năng Quản lý đơn hàng

*Bảng 1. 17 Bảng chức năng quản lý đơn hàng*

Usecase	UC_016_Quản lý đơn hàng
Tên tác nhân: Người dùng (Khách hàng )	Mức độ cần thiết: Bắt buộc
Các thành phần tham gia và mối quan tâm:	Admin muốn quản lý đơn hàng mà khách hàng đã đặt
Mô tả tóm tắt:	Admin có thể quản lý chi tiết từng đơn hàng mà khách hàng đã đặt
Trigger kiểu sự kiện:	external
Các mối quan hệ	<ul style="list-style-type: none"> <li>+ Association (kết hợp): Admin</li> <li>+ Include (bao gồm): Đăng nhập</li> <li>+ Extend (mở rộng):</li> <li>+ Generalization (tổng quát hóa): Xem danh sách đơn hàng, cập nhật trạng thái.</li> </ul>
Luồng xử lý bình thường của sự kiện	<ol style="list-style-type: none"> <li>1. Admin đăng nhập vào website</li> <li>2. Admin nhấn vào danh mục quản lý đơn hàng bên trái</li> </ol>
Các luồng sự kiện con (subflows):	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

### 1.3. Yêu cầu phi chức năng

#### 1.3.1. Yêu cầu thực thi

- Máy tính cần có kết nối Internet và phải được cài đặt đầy đủ các phần mềm hỗ trợ chạy trang web.
- Khi khách hàng thực thi các chức năng trên website thì hệ thống phải thực hiện đúng yêu cầu của người dùng và trả về kết quả thực thi hoặc thông báo.
- Thông tin, dữ liệu cũng như bố cục trang web phải được sắp xếp một cách trực quan, khoa học giúp người dùng dễ sử dụng.

#### 1.3.2. Yêu cầu về hiệu suất

- Hệ thống đảm bảo hoạt động tốt với nhiều lượt truy cập cùng lúc, thời gian chờ tối đa không quá 5s.
- Có khả năng hoạt động tốt trên nhiều thiết bị khác nhau cũng như nhiều hệ điều hành khác nhau.

### **1.3.3. Yêu cầu về an toàn**

- Đảm bảo về dữ liệu khi có yêu cầu lưu trữ một lượng lớn dữ liệu.
- Đảm bảo về phục hồi dữ liệu khi bị mất tập tin lưu trữ.
- Đảm bảo về thông tin, dữ liệu của trang web được sao lưu khi được nâng cấp và bảo trì thường xuyên.

### **1.3.4. Yêu cầu về tính bảo mật**

- Hệ thống được bảo vệ bởi một hệ thống mã hóa mật khẩu tiên tiến phải thường xuyên được cập nhật.
- Các tài khoản chỉ được dùng đúng chức năng được cho phép.
- Xây dựng cơ chế bảo mật bằng mật khẩu để quản lý được số lượng truy cập vào trang web.

### **1.3.5. Yêu cầu giao tiếp bên ngoài**

#### **Giao tiếp phần cứng:**

- Về cơ sở dữ liệu: Có thể chứa lượng lớn dữ liệu của trang web, cho phép kết nối nhanh chóng, phản hồi trực tiếp đến khách hàng, có thể mở rộng và cải tiến thêm CSDL trong tương lai khi hệ thống có nhu cầu phát triển thêm chức năng mới và trang web có thể tương tác với người dùng bằng chuột, bàn phím và màn hình.

- Về máy tính sử dụng: Dùng để hỗ trợ cho Admin và khách hàng có đầy đủ các chức năng, phần mềm hỗ trợ chạy trang web.
- Giao tiếp phần mềm:
- Trang web phải hoạt động tốt trên các trình duyệt web như: Firefox, chrome,... dữ liệu sử dụng từ MongoDB.
- Hệ thống vận hành ổn định trên máy tính sử dụng win 7, 8, 10, MacOS, Linux, điện thoại Android và máy tính bảng.
- Giao tiếp truyền thông: Trang web sử dụng các giao thức truyền HTTP, sử dụng giao thức GET, POST để gửi dữ liệu, các thông báo và các Form đến người dùng.

#### **1.4. Lựa chọn và đánh giá giải pháp**

Giải pháp để xây dựng trang web gồm các bước sau:

- Chọn lựa các ngôn ngữ xây dựng trang web: HTML, CSS, JavaScript và thư viện ReactJS làm nền tảng để xây dựng website.
- Sử dụng các thư viện Framework để xây dựng thư viện: Antd Design, Antd Design Chart
- Sử dụng CSDL MongoDB để tổ chức, xây dựng, lưu trữ và quản lý dữ liệu.

Đánh giá giải pháp:

- Do sử dụng thư viện ReactJS nên tiết kiệm được thời gian và công sức bởi tốc độ xử lý nhanh và dễ sử dụng.
- Sử dụng thư viện Antd Design đem đến cho trang web một giao diện hoàn toàn mới lạ
- CSDL MongoDB miễn phí và có thể chứa một nguồn dữ liệu lớn, phù hợp cho mở rộng dữ liệu cho tương lai khi nâng cấp trang web.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Trong chương này sẽ trình bày tổng quan về trang web, phần tiếp theo là cơ sở lý thuyết các kiến thức về API, thư viện ReactJS, Axios, ExpressJS, NodeJS và MongoDB được áp dụng dung để thiết kế và xây dựng hệ thống.

### 2.1. Tổng quan hệ thống

Website ra đời nhằm hỗ trợ người dùng có một nơi để mua bán trở nên thuận lợi hơn, tiết kiệm và hiệu quả hơn. Tạo ra một môi trường mua sắm hiện đại, khách quan để có thể cùng nhau hội nhập công nghệ. Người dùng trong hệ thống gồm 2 nhóm chính: Khách hàng và Quản trị viên mỗi người chỉ được sử dụng chức năng tương ứng với quyền hạn của mình. Các chức năng chính của hệ thống đã được trình bày cụ thể trong chương một của tài liệu này.

### 2.2. Giới thiệu về API và RESTful API

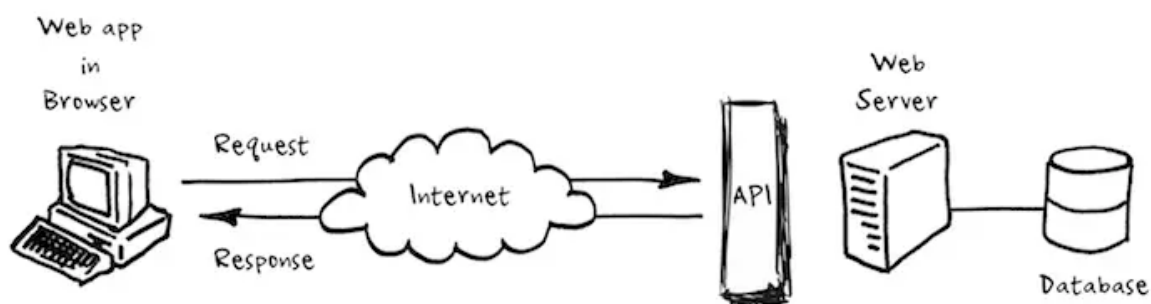
API (Application programming Interface) là những quy tắc tương tác để các ứng dụng và thành phần của ứng dụng có thể giao tiếp được với nhau. Mục đích của API là đơn giản hóa việc lập trình bằng cách trừu tượng hóa đi những cài đặt phức tạp và chi phối bày ra cách thức để thực hiện hành động mà lập trình viên cần. Trong ngữ cảnh lập trình web API là cách thức để trao đổi dữ liệu với một dịch vụ trực tuyến. API sẽ cung cấp các chức năng dịch vụ thông qua đường dẫn URL (Uniform Resource Locator) với định nghĩa thông số kỹ thuật như gửi request bằng HTTP (Hypertext Transfer Protocol) và trả về response message theo định dạng XML (Extensible Markup Language) hoặc JSON (JavaScript Object Notation).

REST (Representational State Transfer) là một kiến trúc phần mềm bao gồm các quy tắc để tạo ra dịch vụ web (web service). Một webservice tuân thủ theo kiến trúc REST thì được gọi là RESTful webservice. Webservice này sử dụng giao thức HTTP để triển khai cấu trúc web. Như vậy, RESTful API chính là kiến trúc thiết kế API tuân thủ theo kiến trúc REST thông qua các phương thức của HTTP (GET, POST, PUT, DELETE, ...) Tương ứng với mỗi phương thức HTTP sẽ thực hiện những tác vụ tương ứng:

*Bảng 2. 1 Bảng phương thức HTTP*

Phương thức HTTP	Tác vụ
GET	Lấy dữ liệu
POST	Tạo mới dữ liệu
PUT	Cập nhật dữ liệu
DELETE	Xóa dữ liệu

Các tác vụ đọc, tạo, cập nhật, xóa được gọi là CRUD service (Create, Read, Update, Delete). Mỗi tác vụ trên phải được gọi thông qua địa chỉ URL (Uniform Resource Identifier) kèm theo phương thức và payload (có thể có hoặc không, thường là định dạng XML hoặc JSON).



Hình 2. 1 Phương thức RESTful API

RESTful API sử dụng giao thức stateless (là một giao thức truyền thông không sử dụng session) và theo tiêu chuẩn nên hệ thống sẽ nhanh, đáng tin cậy và có thể mở rộng dễ dàng. Thông thường, RESTful API sẽ xác thực người dùng khi gửi yêu cầu đối với những tác vụ nguy hiểm như cập nhật hoặc xóa dữ liệu chỉ cho phép đối với người quản trị.

### 2.3. ReactJS



Hình 2. 2 React

ReactJS hay (React, React.js) là một thư viện mã nguồn mở (MIT License) dùng để xây dựng giao diện người dùng được viết bằng JavaScript. React được phát hành lần đầu tiên vào năm 2013 bởi FaceBook. React hiện nay được duy trì bởi Facebook và cộng đồng lập trình viên. React được sử dụng bởi những doanh nghiệp công nghệ hàng đầu như Facebook, Twitter, Instagram.

Sức mạnh của React nằm ở việc tự động cập nhật UI (User Interfaces) khi dữ liệu thay đổi với chi phí ít nhất mà không cần phải tải lại toàn bộ trang web. Điều này làm cho ứng dụng phản hồi lại nhanh hơn, tiết kiệm băng thông, tăng tốc ứng

dụng và trải nghiệm người dùng. Trong react sẽ có một số khái niệm cơ bản về JSX, components, props, state và hook [.

## **2.4. Javascript XML**

JSX (JavaScript XML) là một cú pháp mở rộng của JavaScript cho phép định nghĩa các thành phần HTML trong React. Nói một cách đơn giản cú pháp này cho phép ta viết các HTML tag trực tiếp trong JavaScript

Trên thực tế, các trình duyệt ngay cả các trình duyệt mới nhất cũng không hỗ trợ cú pháp của JSX. Do đó mã nguồn sử dụng JSX cần được chuyển về JavaScript thông qua một thư viện mang tên Bable (một Javascript compiler).

### **2.4.1. Components**

Components là những thành phần UI được chia nhỏ ra, độc lập và có thể tái sử dụng. Components có thể là những function (stateless) hoặc class (stateful) trong JS.

Functional component: là một hàm Javascript hoặc (ES6) trả về một React element. Theo tài liệu chính thức của React, hàm dưới đây là một component hợp lệ.


Function này là một component React hợp lệ vì nó nhận một “props” làm tham số trả về 1 React element.

Function component cũng được biết tới với cái tên là stateless components. Bởi vì chúng không thể làm nhiều thứ phức tạp như quản lý React state (data) hoặc xử lý vấn đề liên quan tới life-cycle trong functional components [7]. Tuy nhiên, từ phiên bản React 16.8 nhà phát hành giới thiệu tính năng React Hooks. Với hooks, chúng ta có thể sử dụng state và những features khác trong functional components.

Class components: Các class components là những ES6. Chúng phức tạp hơn functional components.

Class components còn có:

- + Phương thức khởi tạo, có hàm và vòng đời component, hàm render().
- + State (dữ liệu ứng dụng).



```

import { Component } from "react";

/**
 * @augments {Component<a: number>, State>}
 */
class MyComponent extends Component {
  render() {
    this.props.b; // Error: b does not exist on {a:number}
  }
}

```

Hình 2. 3 Hình class components

Tóm lại, một class components là một class kế thừa từ React.Component, có thể nhận props (trong hàm khởi tạo) nếu cần.

#### 2.4.2. Props và State

Props là những thuộc tính được truyền vào một component và chỉ có thể đọc. Ví dụ như thẻ Post có thuộc tính được truyền vào title. Truy xuất bằng cú pháp props.title sẽ cho giá trị “My post Title”.



```

2
3 // ES6 class
4 class CatComponent extends React.Component {
5   constructor(props) { }
6
7   render() {
8     return <div>{this.props.catName} Cat</div>
9   }
10 }
11
12 // Functional component
13 function CatComponent(props) {
14   return <div>{props.catName} Cat</div>
15 }
16

```

Hình 2. 4 Hình Props và State

State là trạng thái thuộc về chính component đó, được quản lý bởi chính nó và không được truy xuất ra bên ngoài. Chỉ có thể sử dụng state khi dùng stateful component.

#### 2.4.3. Hook

Hook là tính năng được thêm vào React ở phiên bản 16.8. Cho phép người sử dụng một số tính năng chỉ có ở stateful component (class) khi sử dụng stateless component (function) như state (useState), life cycle (useEffect), ...



## 2.5. Thư viện Redux toolkit



Hình 2. 5 Hình Redux

Redux Toolkit là một thư viện giúp đơn giản hóa việc sử dụng Redux. Nó cung cấp một số tính năng hữu ích, bao gồm:

**Slices:** Slices là một cách để nhóm các reducer và action creators lại với nhau. Điều này giúp cho mã của bạn trở nên dễ quản lý hơn.

**Action creators:** Action creators là một cách để tạo ra các action. Redux Toolkit cung cấp một số action creators hữu ích, chẳng hạn như action creators để tạo ra các action với các đối số.

**Reducers:** Reducers là một cách để cập nhật trạng thái của ứng dụng. Redux Toolkit cung cấp một số reducers hữu ích, chẳng hạn như reducers để cập nhật trạng thái của một đối tượng.

**Tạo store:** Redux Toolkit cung cấp một phương thức đơn giản để tạo store.

### Slices

- Slices là một cách để nhóm các reducer và action creators lại với nhau. Điều này giúp cho mã của bạn trở nên dễ quản lý hơn. Để tạo một slice, bạn sử dụng hàm `createSlice()`. Hàm này yêu cầu hai đối số:
- Tên của slice: Tên của slice sẽ được sử dụng để tham chiếu đến nó trong ứng dụng của bạn.
- Một đối tượng chứa các reducer và action creators: Đối tượng này sẽ chứa các reducer và action creators cho slice.

### Action creators

- Action creators là một cách để tạo ra các action. Redux Toolkit cung cấp một số action creators hữu ích, chẳng hạn như action creators để tạo ra các action với các đối số.

### Reducers

- Reducers là một cách để cập nhật trạng thái của ứng dụng. Redux Toolkit cung cấp một số reducers hữu ích, chẳng hạn như reducers để cập nhật trạng thái của một đối tượng.

### Tạo store

- Redux Toolkit cung cấp một phương thức đơn giản để tạo store. Để tạo store, bạn sử dụng hàm createStore(). Hàm này yêu cầu một reducer hoặc một danh sách các reducer.

Redux Toolkit là một thư viện hữu ích giúp đơn giản hóa việc sử dụng Redux. Nó cung cấp một số tính năng hữu ích, bao gồm slices, action creators, reducers và tạo store.

## 2.6. Thư viện Axios

Axios là một thư viện HTTP client dựa trên Promise dùng để gửi các request HTTP bất đồng bộ đến REST endpoint để sử dụng các dịch vụ CRUD [9].

Promise là một cải tiến để loại bỏ try catch, callback rườm rà trong xử lý bất đồng bộ, thay vào đó là then - catch. Hiểu đơn giản là khi thực hiện một tác vụ bất đồng bộ, sau khi thực hiện thành công thì “then” sẽ được gọi, ngược lại khi có lỗi phát sinh thì “catch” sẽ được gọi. Tương tự như promise, axios cũng có thể sử dụng theo kiểu gửi request lồng nhau.

Ngoài ra, axios còn hỗ trợ interceptor dùng để thực hiện một số công việc khác trước khi gửi request hoặc ngay sau khi nhận được response thông báo lỗi “401-Unauthorized”.

## 2.7. Mongo

MongoDB là một database hướng tài liệu (document) và là CSDL thuộc NoSQL hàng đầu, được hàng triệu người sử dụng. MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và document khác nhau. Các dữ liệu được lưu trữ trong document kiểu Json nên truy vấn sẽ rất nhanh.

NoSQL là 1 dạng CSDL mã nguồn mở không sử dụng transect-SQL, hay có nơi thường gọi là Not-Only SQL. CSDL này phát triển trên JavaScript framework với kiểu dữ liệu JSON (cú pháp của JSON là “key.value”). NoSQL ra đời như là một mảnh vá cho những khuyết điểm và thiếu sót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS về tốc độ, tính năng khả năng mở rộng, memory cache,...

Các thuật ngữ trong MongoDB

- `_id`: Là trường bắt buộc có trong mỗi document. Trường `_id` đại diện cho một giá trị duy nhất trong Document MongoDB. Trường `_id` cũng có thể được hiểu là khóa chính trong document.
- Collection: Là nhóm của nhiều document trong MongoDB. Collection có thể được hiểu là một bảng tương ứng trong CSDL RDBMS (Relational Database Management System). Collection nằm trong một CSDL duy nhất. Các collection không phải định nghĩa các cột, các hàng hay kiểu dữ liệu trước.
- Cursor: Đây là một con trỏ đến tập kết quả của một truy vấn. Máy

khách có thể lặp qua một con trỏ để lấy kết quả.

- Database: Nơi chứa các Collection, giống với CSDL quan hệ, chúng chứa các bảng. Mỗi database có một tập tin riêng lưu trữ trên bộ nhớ vật lý. Một máy chủ MongoDB có thể chứa nhiều Database.

### 2.7.1. Mongoose

Mongoose là một thư viện mô hình hóa đối tượng (Object Data Model - ODM) cho MongoDB và Node.js. Nó quản lý mối quan hệ giữa dữ liệu, cung cấp sự xác nhận gián đồ và được sử dụng để dịch giữa các đối tượng trong mã và biểu diễn các đối tượng trong MongoDB.

Mongoose cung cấp một số lượng đáng kinh ngạc các chức năng cho việc tạo ra và làm việc với các schema. Mongoose hiện có 8 Schema Types gồm: String, Number, Date, Buffer, Boolean, Mixed, ObjectId, Array.

## 2.8. NodeJS



Hình 2. 6 Hình NodeJS

NodeJS Hình 2: 8 là một nền tảng được xây dựng trên V8 JavaScript Engine - trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng [8].

NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy được trên hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.

### 2.8.1. Module

Node.js sử dụng kiến trúc Module để đơn giản hóa việc tạo ra các ứng dụng phức tạp. Module là giống như các thư viện trong C, C#, Java,... Mỗi Module chứa một tập các hàm chức năng có liên quan đến một đối tượng của module.

Các module được giữ tách riêng biệt với nhau, tách riêng các code base, khi nào cần sử dụng những cái nào thì gọi chúng ra và kết hợp lại với nhau tùy logic xử lý của bạn.

### 2.8.2. NPM – Node Package Manager

NPM viết tắt của Node Package Manager là một công cụ tạo và quản lý các thư viện lập trình JavaScript cho NodeJS. Trong cộng đồng JavaScript, các lập trình viên chia sẻ hàng trăm nghìn các thư viện với các đoạn code đã thực hiện sẵn một chức năng nào đó. Nó giúp cho các dự án mới tránh phải việc lập lại các thành phần cơ bản, Các thư viện lập trình hay thậm chí các Framework.

Các loại Package

Dựa theo chức năng mà ta chia package ra làm 2 loại, đó là Simple dependencies và Development Dependencies.

Simple dependencies là những package bắt buộc phải có trong quá trình chạy sản phẩm. Khi cài đặt Simple dependencies, Npm sẽ tự động cài đặt tất cả các dependencies cần thiết.

Development dependencies là những package bắt buộc phát triển cũng như phát hàng sản phẩm. Khi cài đặt Development dependencies, Npm sẽ chỉ cài đặt các dependencies cần thiết.

### 2.9. ExpressJS Framework



Hình 2. 7 Hình Express JS

ExpressJS là một framework được xây dựng trên nền tảng của NodeJS. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

Định tuyến (Routing)

Routing trong Node.js là một khái niệm nói đến việc xác định ứng dụng sẽ đáp ứng như thế nào khi người dùng tạo một request đến một endpoint (điểm cuối) cụ thể nào đó. Điểm cuối đó thường là một URI hoặc một đường dẫn (Path) với một request method (POST, PUT, GET,...) cụ thể.

Cấu trúc định tuyến cơ bản: app.METHOD (Path, Handler...) Trong đó:

- app: là một instance của express
- Method: là một HTTP Method

- Express hỗ trợ rất nhiều loại HTTP methods khác nhau, bao gồm:
- get, put, head delete, options, trace, copy, lock, mkcol, move, purge, unlock, report, mkactivity, checkout, merge, m-search, notify, subscribe, patch.
- Path: là một đường dẫn trên máy chủ Route path *Hình 2: 10* có thể là một chuỗi thông thường (String) hoặc là một chuỗi có ký hiệu biểu thức chính quy (String patterns) hoặc là một biểu thức chính quy (regular expressions)
- Handler: là một function sẽ thực thi khi một route được trùng khớp

Đơn giản là một hoặc nhiều function sẽ được gọi khi một route trùng khớp để đáp ứng yêu cầu nào đó. Lưu ý các handler sẽ được gọi đúng theo thứ tự truyền vào.

#### Route parameters

```
// → routes/posts.routes.js

const express = require('express')
const router = express.Router()
const posts = require('../controllers/post.controller')

router.get('/', posts.findAll)
router.post('/', posts.create)
router.get('/:postId', posts.find)
router.put('/:postId', posts.update)
router.delete('/:postId', posts.delete)

module.exports = router
```

*Hình 2. 8 Hình Route parameters*

Route parameters là những vị trí trên URL được đánh dấu bằng cách đặt tên, mục đích chính là để lấy ra các giá trị tương ứng. Tất cả các giá trị đối số sẽ được đặt vào đối tượng req trong thuộc tính params. Với thuộc tính trùng khớp với từ khóa được xác định trên URL.

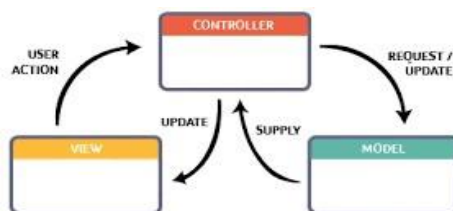
#### Response methods

Sau việc tiếp nhận và xử lý, thì việc tiếp theo là đáp ứng (Response). Trong express định nghĩa sẵn một phương hỗ trợ hay dùng nhất là

Bảng 2. 2 Bảng phương thức Response

Tên phương thức	Ý nghĩa
res.json()	Trả về dữ liệu dạng JSON
res.redirect()	Chuyển hướng đến một đường dẫn nào đó
res.render()	Trả về một view template
res.send()	Gửi dữ liệu dạng text

## 2.10. Mô hình MVC



Hình 2. 9 Hình mô hình MVC

- **Model:** có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu và controller sẽ thông qua các hàm, phương thức đó để lấy dữ liệu rồi gửi qua View.
- **View:** có nhiệm vụ tiếp nhận dữ liệu từ controller và hiển thị nội dung sang các đoạn mã HTML, bạn có thể hiểu nôm na đây là người ta còn gọi là phần giao diện [5]
- **Controller:** đóng vai trò trung gian giữa Model và View. Nó có nhiệm vụ tiếp nhận yêu cầu từ client sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về client [10].

Luồng xử lý

## 2.11. Cloudinary



*Hình 2. 10 Hình Cloudinary*

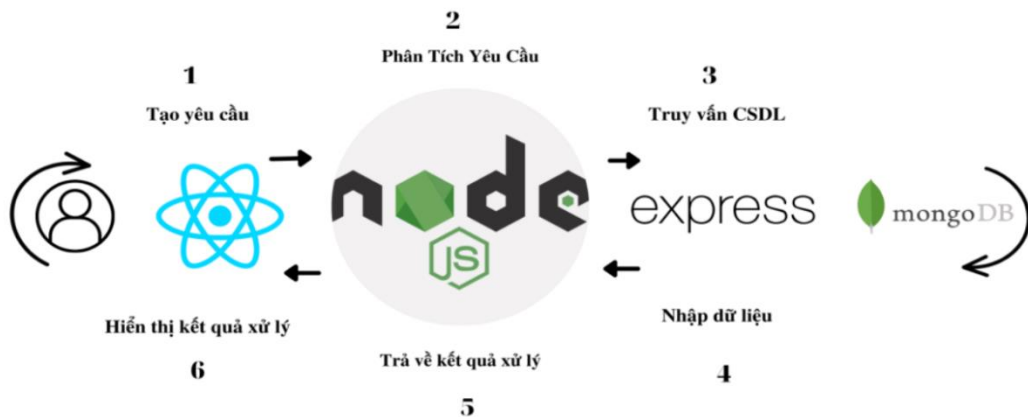
Cloudinary là một dịch vụ dựa trên đám mây, cung cấp giải pháp quản lý hình ảnh bao gồm upload, lưu trữ, thao tác, tối ưu hóa và delivery. Với Cloudinary, bạn có thể dễ dàng upload ảnh lên đám mây, tự động thực thi các thao tác với ảnh một cách thông minh mà không cần phải cài đặt bất kỳ một phần mềm phức tạp nào khác. Cloudinary cung cấp các APIs toàn diện và màn hình quản lý giúp chúng ta dễ dàng tích hợp vào các trang web và ứng dụng di động.

## CHƯƠNG 3: THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

Phần đầu chương 3 thể hiện kiến trúc của hệ thống, phần tiếp theo giải thích các lớp trong mô hình MVC được dùng trong website, các sơ đồ, biểu đồ mô tả cách thức hoạt động của trang web, cuối cùng là phần kết quả thiết kế giao diện và thiết kế chức năng của website.

### 3.1. Kiến trúc hệ thống

#### 3.1.1. Thiết kế kiến trúc



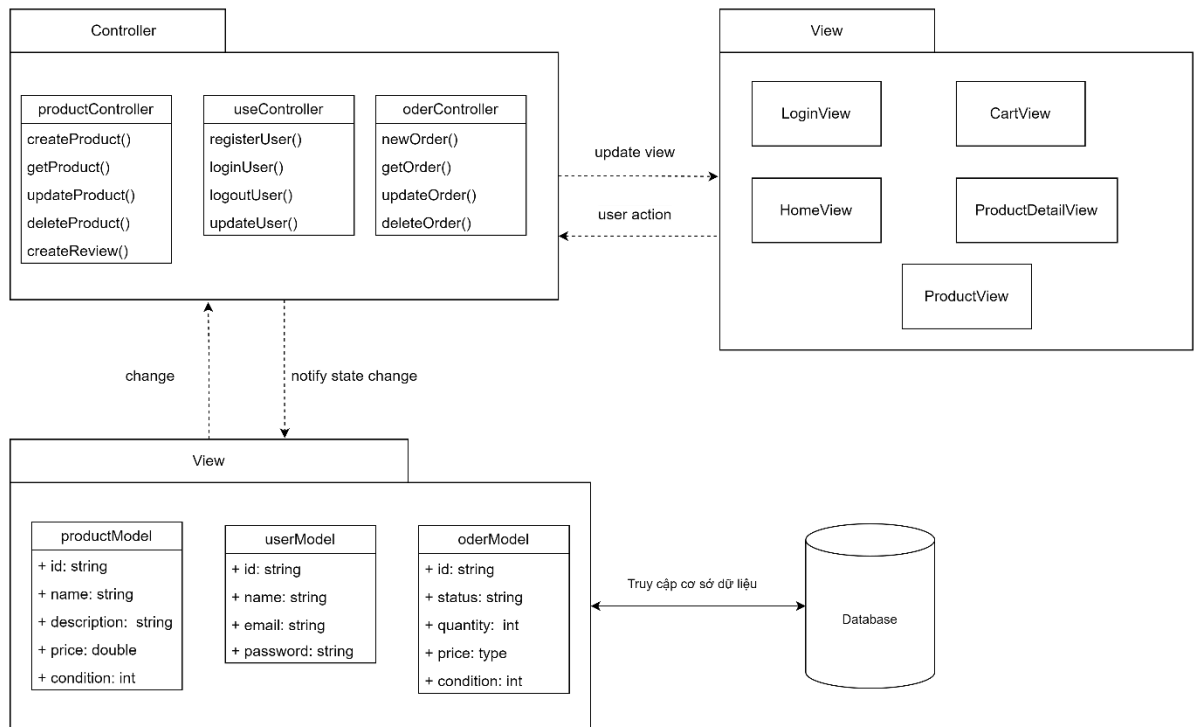
Hình 3. 1 Hình sơ đồ thiết kế kiến trúc

Thành phần trong thiết kế kiến trúc gồm:

- **Khách hàng:** là nhóm người dùng phổ biến truy cập đến địa chỉ của website để xem và mua sắm sản phẩm.
- **ReactJS:** là một thư viện giúp tạo giao diện người dùng một cách dễ dàng, là một thư viện giúp thiết kế, phát triển và hiển thị các thành phần giao diện người dùng, kiểm soát trạng thái của các sự kiện khi có một lượng lớn dữ liệu động cần được cập nhật.
- **Node.js:** là Java Runtime Environment (JRE), được sử dụng để tạo ra web server
- **Express.js:** là một web framework được xây dựng trên Node.js. Nó được sử dụng để tạo API và phát triển ứng dụng web.
- **MongoDB:** là một hệ quản trị cơ sở dữ liệu NoSQL mã nguồn mở đa nền tảng viết bằng C++. Bản ghi trong MongoDB được lưu trữ dạng một dữ liệu văn bản, là nơi tổ chức và lưu trữ dữ liệu của website.

#### 3.1.2. Mô hình MVC mô tả thiết kế kiến trúc của hệ thống





Hình 3. 2 Sơ đồ kiến trúc MVC Web bán đồ chơi cũ

Kiến trúc hệ thống [4] của “Website đồ chơi cũ” được xây dựng theo kiến trúc MVC (Model - View - Controller) là mô hình chuẩn cho ứng dụng website được sử dụng phổ biến hiện nay. Trong đó:

- **Controller:** là nơi tiếp nhận các yêu cầu xử lý từ phía khách hàng, có nhiệm vụ xử lý giữa, Model và View tương tác và kết nối với nhau. Controller nhận các yêu cầu (request) từ phía khách hàng, sau đó gọi các phương thức để thực hiện các hoạt động được yêu cầu xử lý và gửi ra View. Lúc này View sẽ chịu trách nhiệm format lại các dữ liệu (data) từ phía Controller gửi ra và trình bày các dữ liệu đó theo một định dạng chuẩn ra HTML.
- **Model:** là nơi chịu trách nhiệm quản lý tất cả các dữ liệu, là nơi lưu trữ và truy xuất các thực thể từ CSDL MongoDB và chứa các logic được thực thi bởi ứng dụng.
- **View:** là nơi chịu trách nhiệm hiển thị ra tất cả dữ liệu và giúp tương tác với hệ thống đã được truy xuất từ Model theo chuẩn đã được quy ước từ các lập trình viên.
- **MongoDB:** là nơi tổ chức và lưu trữ dữ liệu của website.

Khách hàng truy cập vào website, khi gửi một yêu cầu (request) thì đều sẽ phải đi qua routing, các dữ liệu (data) được truyền xuống Controller để xử lý. Nếu cần dữ liệu thì phải lấy từ Model lên hoặc cập nhật dữ liệu xuống Model, lúc này các kết quả sẽ được gửi ra View cho phía khách hàng.

## **3.2. Tổng quan các lớp của mô hình MVC**

### **3.2.1. Lớp Controller**

Controller là thành phần xử lý các yêu cầu từ người dùng. Nó nhận các sự kiện từ giao diện người dùng thông qua View, xử lý logic điều khiển và gửi yêu cầu đến Model để thay đổi dữ liệu. Controller sau đó cập nhật View dựa trên dữ liệu mới từ Model.

### **3.2.2. Lớp Model**

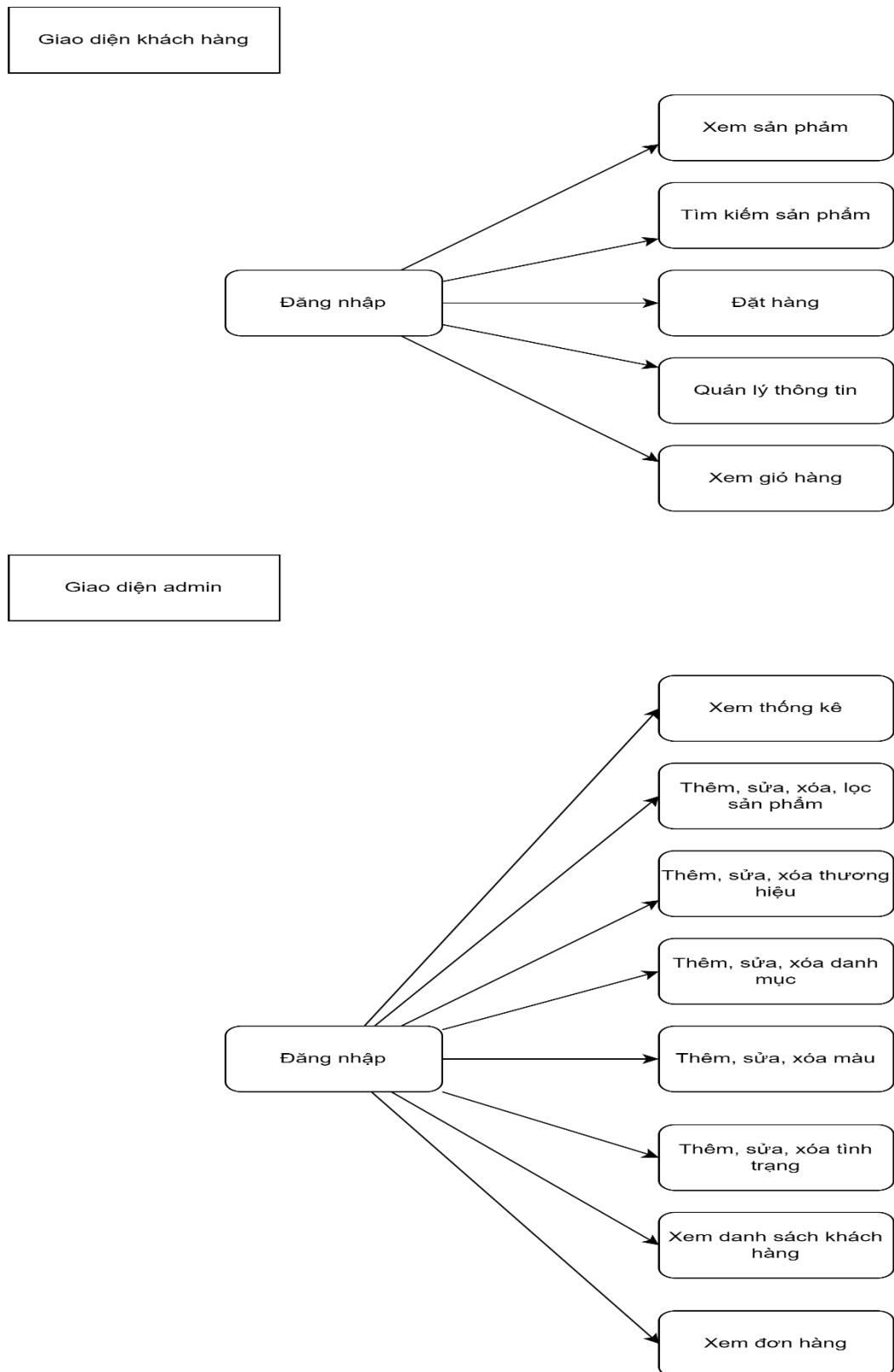
Đây là thành phần chịu trách nhiệm cho việc xử lý dữ liệu của ứng dụng. Nó đại diện cho các đối tượng dữ liệu và logic xử lý dữ liệu như lưu trữ, truy xuất, cập nhật dữ liệu. Model không biết gì về giao diện người dùng hoặc controller.

### **3.2.3. Lớp View**

View là lớp thuộc về giao diện nơi tương tác giữa người dùng với phần phía sau của trang web nó cũng là nơi hiển thị kết quả trả về cho người dùng. Để thực hiện các chức năng mà người dùng mong muốn, họ phải tương tác với website qua lớp View, với chức năng chính là hiển thị giao diện cho người dùng nên có thể nói lớp View là bộ mặt của cả trang web cũng không sai.

Đối với website bán đồ chơi cũ các đối tượng khách hàng, người dùng được chia làm hai nhóm chính là Người quản trị (Admin) và khách hàng (người dùng đã đăng ký và chưa đăng ký tài khoản đăng nhập), theo đó từng nhóm đối tượng khi truy cập vào trang web sẽ hiển thị từng giao diện khác nhau cho từng đối tượng.

Sơ đồ mô tả giao diện chức năng của từng nhóm đối tượng người dùng bao gồm người quản trị (admin) và khách hàng (khách hàng đã đăng ký tài khoản và khách hàng chưa đăng ký tài khoản).

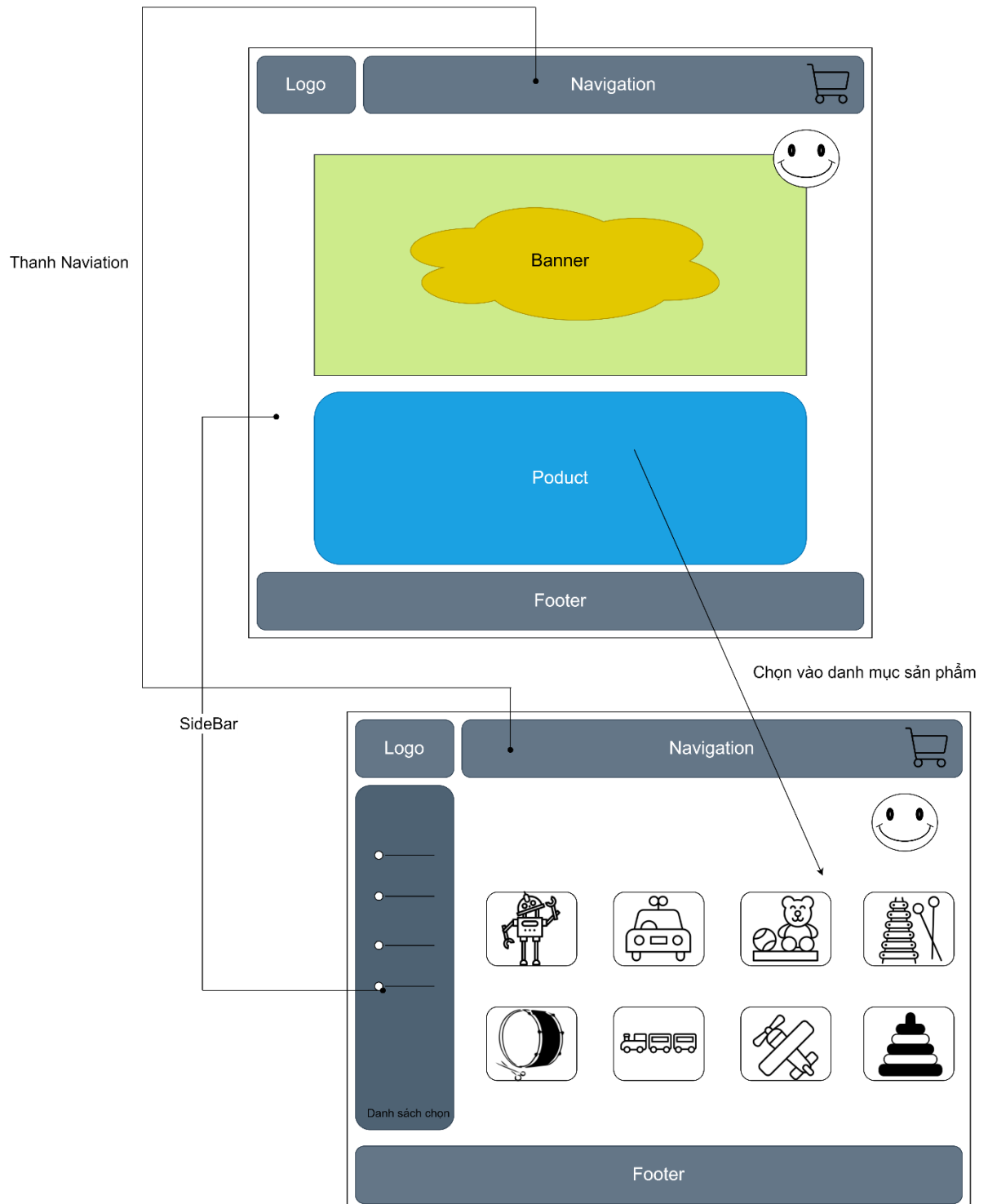


Hình 3. 3 Sơ đồ giao diện Admin và khách hàng

Một ví dụ khi khách hàng đăng nhập từ giao diện trang chủ người dùng Hình 3: 7, tương tác, nhận thông tin trả về và hiển thị ở View. Từ hình minh họa bên dưới để thấy được giao diện khi khách hàng chuyển từ giao diện trang chủ sang giao diện trang danh

mục sản phẩm

Ví dụ: giao diện View thay đổi khi khách hàng thực hiện thao tác chọn vào danh mục sản phẩm từ giao diện trang chủ



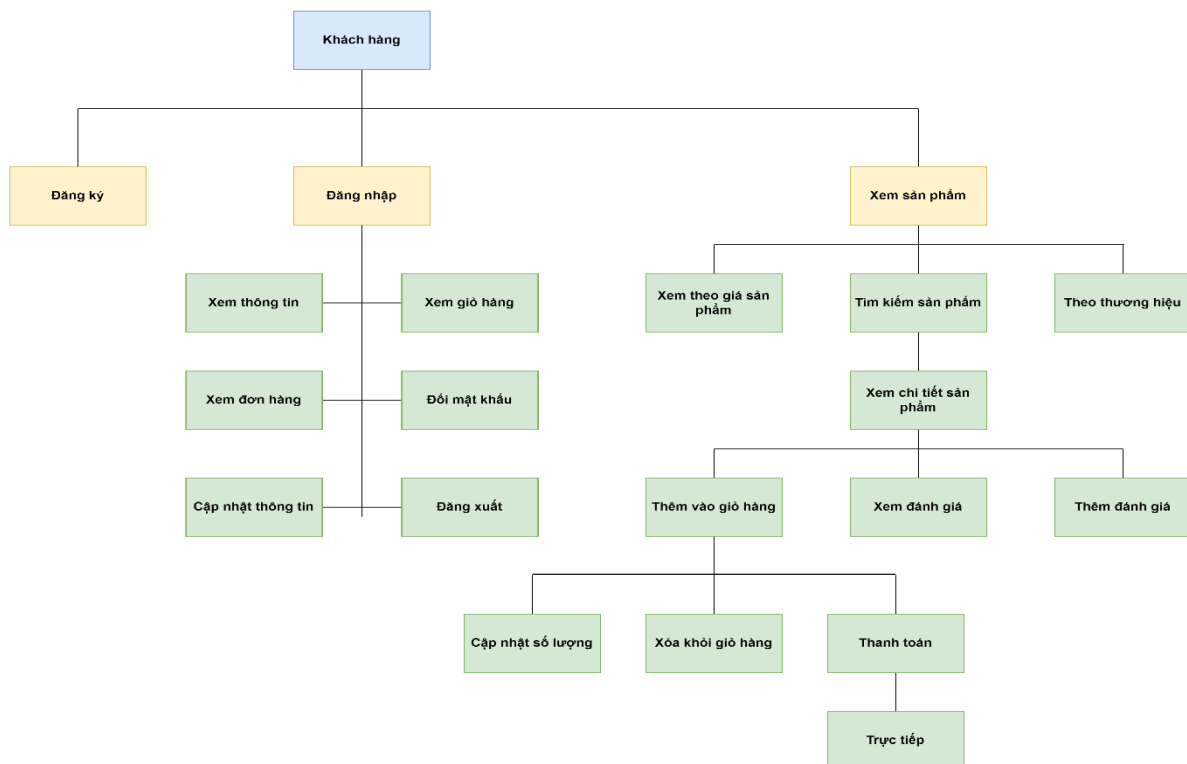
### 3.3. Sơ đồ phân rã chức năng

#### 3.3.1. Sơ đồ phân rã chức năng cho đối tượng Admin ( Người quản trị )



Hình 3. 4 Sơ đồ phân rã Admin

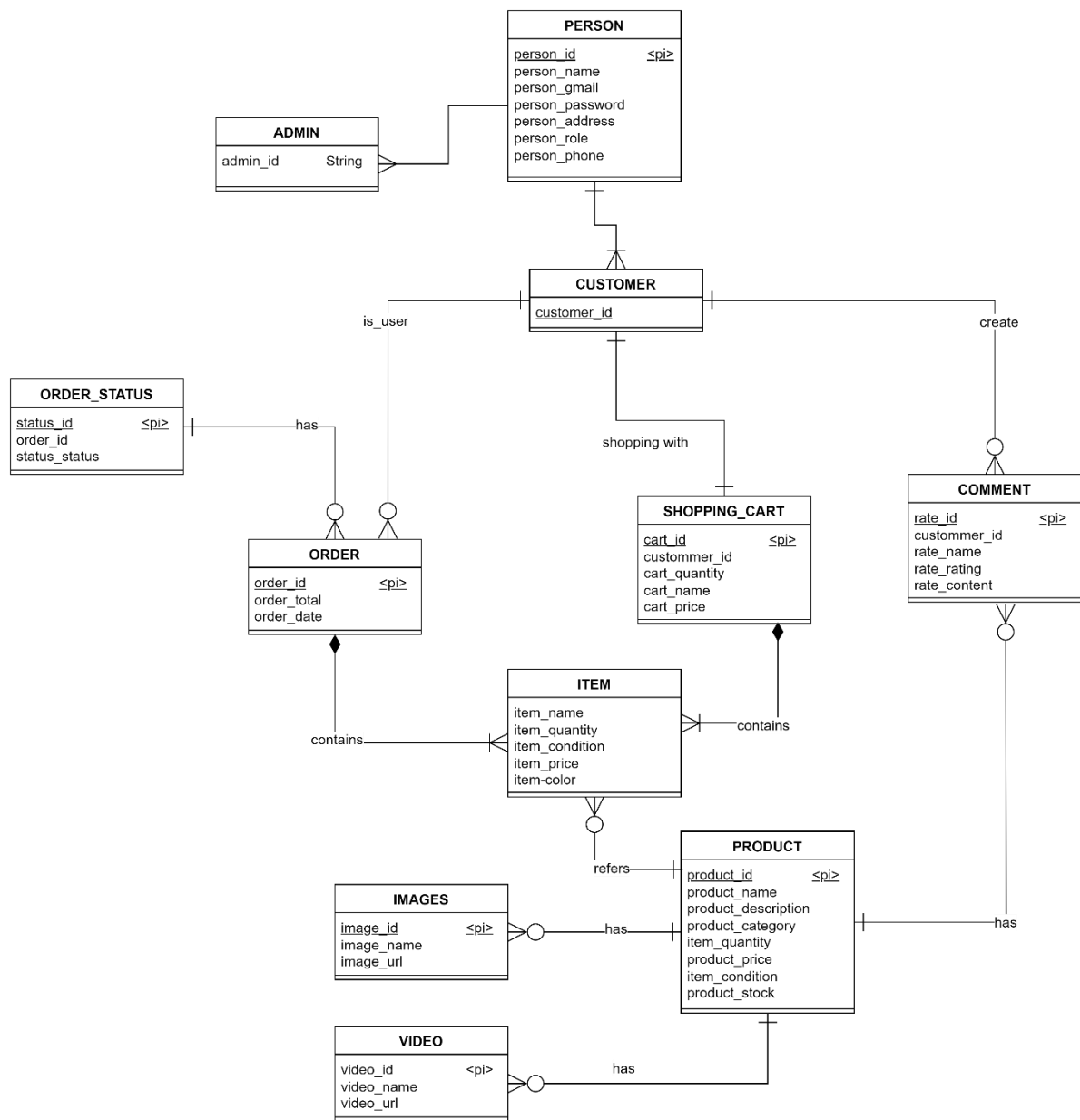
#### 3.3.2. Sơ đồ phân rã chức năng cho đối tượng khách hàng



Hình 3. 5 Sơ đồ phân rã khách hàng

### 3.4. Thiết kế dữ liệu

#### 3.4.1. Mô hình dữ liệu mức quan niệm CDM (Conceptual Data Model)



Hình 3. 6 Hình sơ đồ CDM mức quan niệm

### 3.4.2. Từ điển dữ liệu

#### Bảng ADMIN

Dùng để lưu trữ dữ liệu admin\_id

*Bảng 3. 1 Bảng ADMIN*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	admin_id	Interger	Mã quản trị

#### Bảng PERSON

Dùng để lưu trữ người dùng

*Bảng 3. 2 Bảng PERSON*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	person_id	Interger	Mã người dùng
2	person_name	Var_char	Tên người dùng
3	person_gmail	Var_char	Gmail người dùng
4	person_password	Var_char	Mật khẩu người dùng
5	person_role	Var_char	Quyền người dùng
6	person_phone	Number	Số điện thoại người dùng

#### Bảng CUSTOMER

Dùng để lưu trữ dữ liệu ID

*Bảng 3. 3 Bảng CUSTOMER*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	customer_id	Interger	Định danh người dùng

## Bảng SHOPPING\_CART

Dùng để lưu thông tin giỏ hàng

*Bảng 3. 4 Bảng SHOPPING\_CART*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	cart_id	Interger	Định danh giỏ hàng
2	customer_id	Interger	Định danh khách hàng
3	cart_quantity	Var_char	Số lượng sản phẩm trong giỏ hàng
4	cart_name	Var_char	Tên sản phẩm trong giỏ hàng
5	cart_price	Double	Giá sản phẩm trong giỏ hàng

## Bảng ITEM

Dùng để item trong giỏ hàng của khách hàng

*Bảng 3. 5 Bảng ITEM*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	item_name	Interger	Tên sản phẩm
2	item_quantity	Number	Số lượng khách chọn
3	item_condition	Var_char	Tình trạng sản phẩm
4	item_price	Double	Giá sản phẩm
5	item_color	Var_char	Màu sản phẩm

## Bảng ORDER

Dùng để lưu đơn hàng

*Bảng 3. 6 Bảng ORDER*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	order_id	Interger	Định danh đơn hàng
2	order_total	Var_char	Tổng giá trị đơn hàng
3	order_date	Date	Ngày đặt hàng



## Bảng ORDER\_STATUS

Dùng để lưu trạng thái đơn hàng

*Bảng 3. 7 Bảng ORDER\_STATUS*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	status_id	Interger	Định danh giỏ hàng
2	order_id	Interger	Số lượng sản phẩm
3	status_status	Interger	Trạng thái đơn hàng

## Bảng COMMENT

Dùng để comment của khách hàng

*Bảng 3. 8 Bảng COMMENT*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	rate_id	Interger	Định danh bình luận
2	customer_id	Interger	Định danh người dùng
3	rate_name	String	Tên đánh giá
4	rate_rating	Interger	Đánh giá
5	rate_content	String	Nội dung đánh giá

## Bảng PRODUCT

Dùng để lưu sản phẩm

*Bảng 3. 9 Bảng PRODUCT*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	product_id	Interger	Định danh chi tiết sản phẩm
2	product_name	String	Tên sản phẩm
3	product_description	String	Mô tả sản phẩm
4	product_category	String	Danh mục sản phẩm
5	item_quantity	Int	Chi tiết sản phẩm
6	product_price	Double	Giá sản phẩm
7	item_condition	String	Tình trạng sản phẩm
8	product_stock	Number	Tình trạng sản phẩm trong kho

## Bảng IMAGES

Dùng để lưu hình ảnh

*Bảng 3. 10 Bảng IMAGES*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	image_id	String	Định danh ảnh
2	image_name	String	Tên ảnh
3	image_url	String	Đường dẫn ảnh

## Bảng VIDEO

Dùng để lưu video

*Bảng 3. 11 Bảng VIDEO*

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	video_id	String	Định danh video
2	video_name	String	Tên video
3	video_url	String	Đường dẫn video

### 3.4.3. Các yêu cầu ràng buộc cần thiết khi xử lý dữ liệu

Bảng COMMENT ( Bình luận )

- Thuộc tính rate số lượng phải lớn hơn 3 và nhỏ hơn 5

Bảng CUSTOMER ( Khách hàng )

- Thuộc tính password: bắt buộc phải khác rỗng
- Thuộc tính email: bắt buộc phải khác rỗng, và đúng định dạng
- Thuộc tính phone\_number : Phải đủ 10 số

Bảng ORDER ( Đặt hàng )

- Thuộc tính total phải lớn hơn 0

Bảng PRODUCT ( Sản phẩm )

- Thuộc tính price phải lớn hơn 0
- Thuộc tính quantity không quá 4 chữ số

Bảng STATUS ( Trạng thái )

- Thuộc tính name thuộc các giá trị: “Đã đặt hàng, Đang xử lý, Đã vận chuyển, Giao hàng, Đã giao hàng”

### 3.5. Thiết kế chức năng

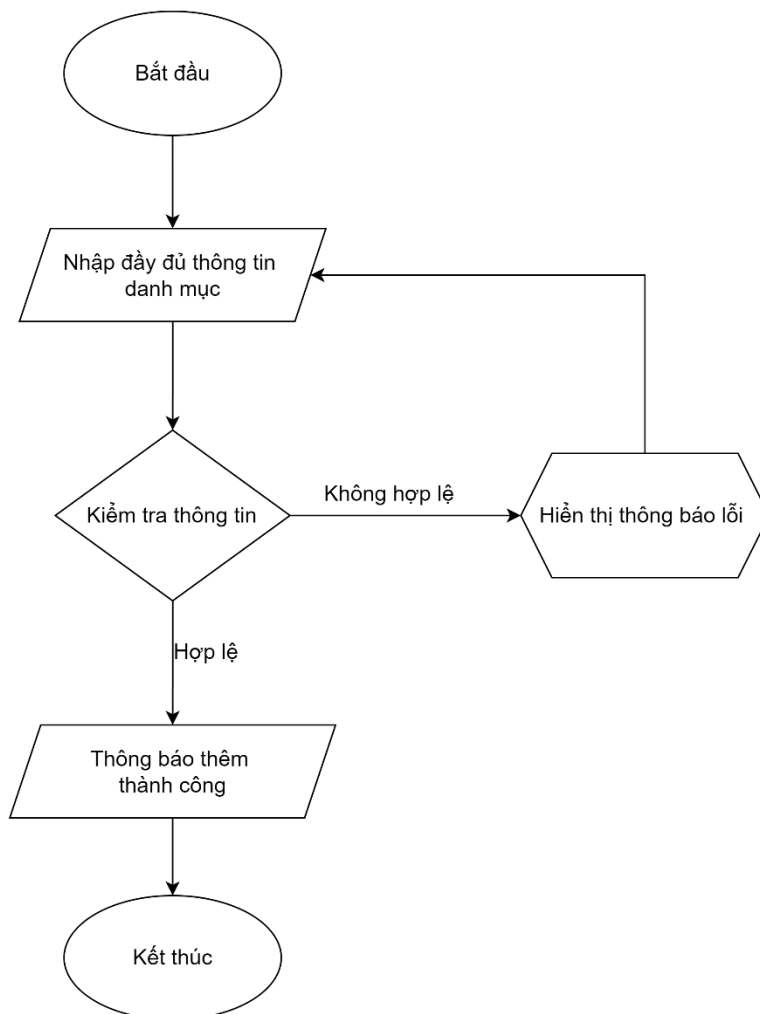
#### 3.5.1. Chức năng quản lý danh mục

- Giao diện thêm danh mục:
  - + Ở giao diện này để thêm một danh mục mới ta nhập tên danh mục. Sau đó nhấp chọn vào nút thêm.



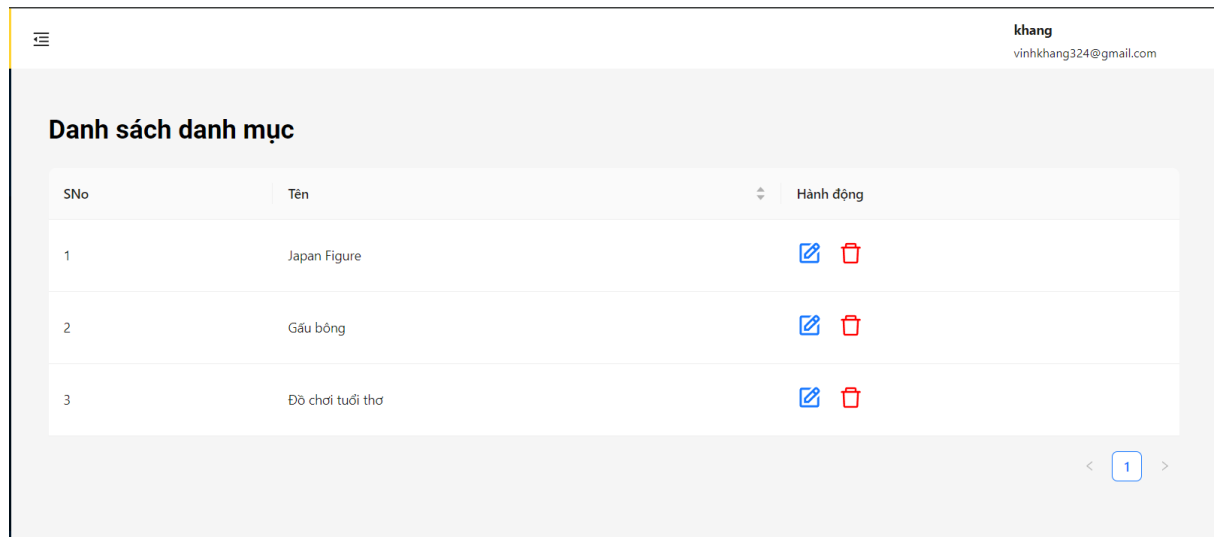
Hình 3. 7 Hình thêm danh mục

- Cách xử lý:



Hình 3. 8 Sơ đồ xử lý Thêm danh mục

- Giao diện quản lý chung
  - + Ở giao diện này chúng ta nhấp để sửa danh mục đó.
  - + Nhấp thùng rác để xóa danh mục



Hình 3. 9 Hình danh sách danh mục

- Giao diện chỉnh sửa danh mục:
  - + Để chỉnh sửa danh mục ta nhập thông tin cần chỉnh sửa và chọn nút “Sửa danh mục”.



Hình 3. 10 Hình sửa danh mục

### 3.5.2. Chức năng quản lý thương hiệu

- Giao diện thêm thương hiệu:
  - + Ở giao diện này để thêm một thương hiệu mới ta nhập tên thương hiệu. Sau đó nhấp chọn vào nút thêm.

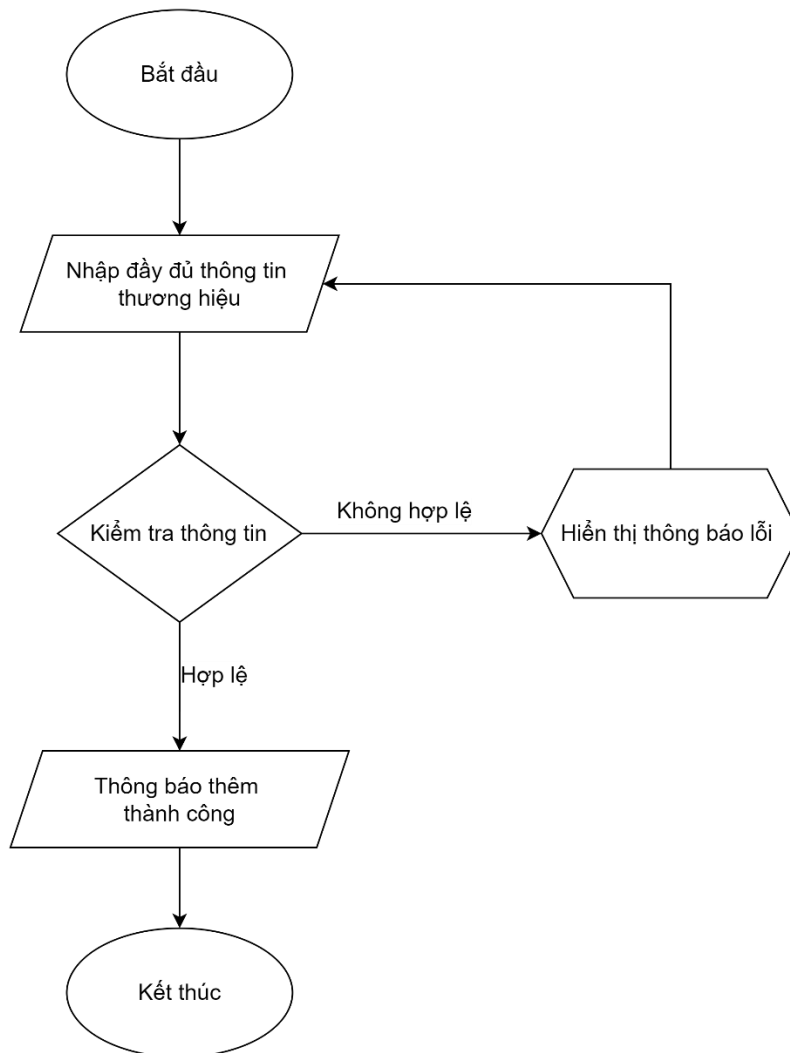
Thêm Thương hiệu

Thêm tên thương hiệu

Thêm Thương hiệu

Hình 3. 11 Hình thêm thương hiệu sản phẩm













- Cách xử lý



Hình 3. 12 Sơ đồ xử lý Thêm thương hiệu

- Giao diện quản lý chung

- + Ở giao diện này chúng ta nhấp để sửa thương hiệu đó.
- + Nhấp thùng rác để xóa thương hiệu

Danh sách thương hiệu		
SNo	Tên	Hành động
1	Aniplex	 
2	Figure	 
3	Gấu bông	 
4	Máy bay	 
5	MP Oto	 
6	22	 

Hình 3. 13 Hình danh sách thương hiệu

- Giao diện chỉnh sửa thương hiệu:
  - + Để chỉnh sửa thương hiệu ta nhập thông tin cần chỉnh sửa và chọn nút “Sửa thương hiệu”

Sửa Thương hiệu

Máy bay

Sửa Thương hiệu

Hình 3. 14 Hình sửa thương hiệu

### 3.5.3. Chức năng quản lý màu

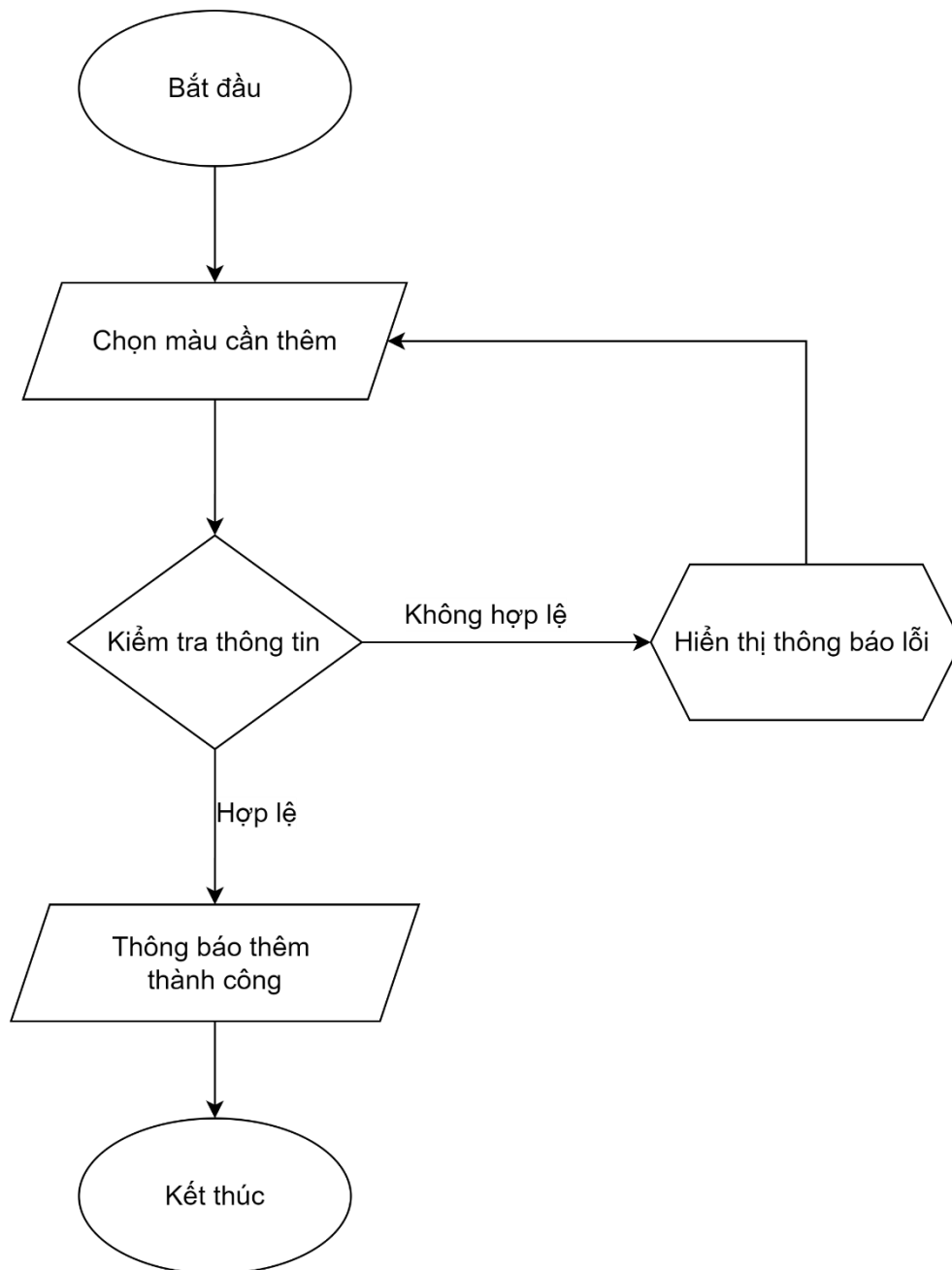
- Giao diện thêm màu:
  - + Ở giao diện này để thêm một màu mới ta chọn màu để thêm. Sau đó nhấp chọn vào nút thêm.

Thêm Màu

Thêm Màu

Hình 3. 15 Hình thêm màu sản phẩm









- Cách xử lý





*Hình 3. 16 Sơ đồ xử lý thêm màu sản phẩm*

- Giao diện quản lý chung

- + Ở giao diện này chúng ta nhấp để sửa màu đó.
- + Nhấp thùng rác để xóa màu

Danh mục màu		
SNo	Tên	Hành động
1	#274bdd	 
2	#dd5f9c	 
3	#1fefcd	 
4	#1e6b47	 


1


Hình 3. 17 Hình danh mục màu

- Giao diện chỉnh sửa màu:
  - + Để chỉnh sửa màu ta nhập thông tin cần chỉnh sửa và chọn nút “Sửa màu”

### Sửa Màu

Sửa Màu

Hình 3. 18 Hình sửa màu





#### 3.5.4. Chức năng quản lý sản phẩm

- Giao diện thêm mới sản phẩm
  - + Để thêm một sản phẩm mới vào hệ thống ta nhập tên sản phẩm, giá, mô tả sản phẩm, giá tiền, chọn danh mục, chọn thương hiệu, thẻ, màu, nhập số lượng sản phẩm, hình ảnh và video. Sau đó nhấn nút “Thêm”



### Thêm Sản Phẩm

Normal

**B** *I* U    

Chọn tình trạng

▼

Chọn thương hiệu

▼

Chọn danh mục

▼

Thẻ

▼

Chọn màu

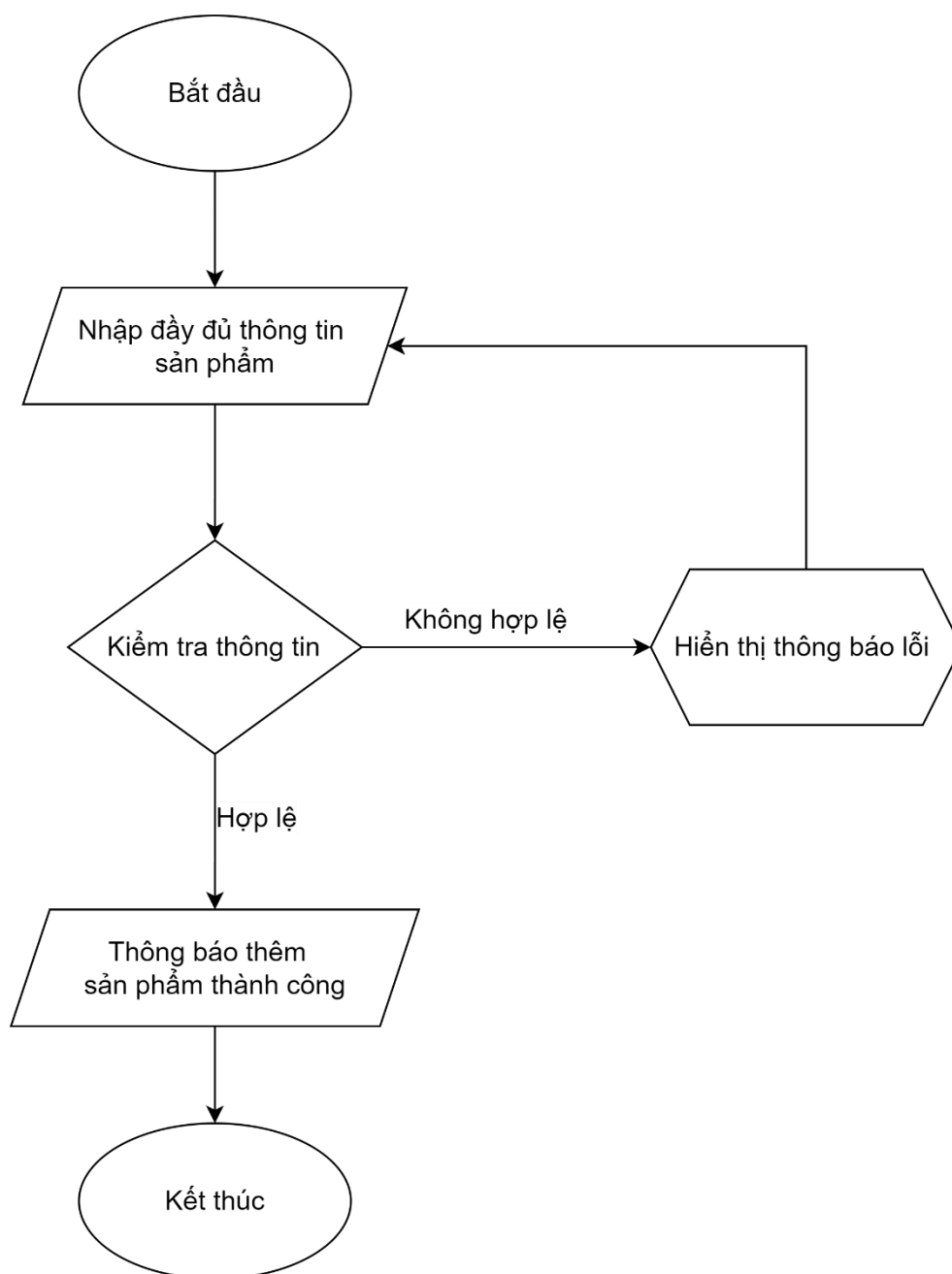
▼

Kéo 'n' thả một số file vào đây hoặc click để chọn file

Thêm Sản phẩm

Hình 3. 19 Hình thêm sản phẩm







- Cách xử lý



Hình 3. 20 Sơ đồ xử lý thêm sản phẩm

- Đây là giao diện chung tất cả sản phẩm

**Danh sách sản phẩm**

SNo	Tiêu đề	Thương hiệu	Danh mục	Màu	Tình trạng	Giá tiền	Hành động
1	Cầu nhồi bóng	Gấu bông	Gấu bông	656ae9b90f4e36a12aabf44a	656ae7d70f4e36a12aabf397	100000	 
2	Máy bay chiến đấu bằng nhựa xua tuổi thơ thế hệ 8x, 9x	Máy bay	Đồ chơi tuổi thơ	656ae9b90f4e36a12aabf44a		25000	 
3	Lô 02 ô tô nhựa tuổi thơ thế hệ 8x, 9x	MP Oto	Đồ chơi tuổi thơ	656ae9b90f4e36a12aabf44a	656ae7d70f4e36a12aabf397	25000	 

< 1 >

*Hình 3. 21 Hình danh sách sản phẩm*

- + Ở giao diện chung tất cả sản phẩm chúng ta nhấp vào hình cây bút để sửa sản phẩm và nhấp biểu tượng thùng rác để xóa sản phẩm
- Để sửa một sản phẩm thì  
 Bước 1: Nhấp biểu tượng cây bút tại ô sản phẩm muốn sửa thì giao diện sẽ hiển thị.

Sửa Sản Phẩm

Cửu nhối bông

Normal

B

I

U

Hàng tăng  
Hình dáng con cửu

95%

100000

Gấu bông


Gấu bông

Special

#dd5f9c

1

Kéo 'n' thả một số file vào đây hoặc click để chọn file



Sửa Sản phẩm

Hình 3. 22 Hình sửa sản phẩm

Bước 2: Nhập thông tin cần thay đổi

Bước 3: Bấm vào nút “Sửa sản phẩm”

Để xóa sản phẩm: ta chọn biểu tượng thùng rác để xóa trong danh sách sản phẩm.

### 3.5.5. Chức năng quản lý đơn hàng

- Giao diện đơn hàng:

khánh  
vinhkhong324@gmail.com

Đơn hàng

SNo	Tên	Sản phẩm	Giá tiền	Ngày mua	Hành động
1	khánh22	<a href="#">Xem đơn hàng</a>	25000	19:41:17 4/12/2023	<div>Đang xử lý</div>

<

1

>

Hình 3. 23 Hình quản lý đơn hàng

- Đơn hàng có 5 trạng thái khác nhau:
  - + Đã đặt hàng
  - + Đang xử lý
  - + Đã vận chuyển
  - + Giao hàng
  - + Đã giao hàng
- Tùy theo các trạng thái khác nhau mà người quản trị có tác động khác nhau
- Tiến trình của đơn hàng là Đang xử lý -> Đã vận chuyển -> Chuẩn bị giao hàng -> Đã giao hàng
- Dòng chữ xem đơn hàng nhấp vào để xem chi tiết đơn hàng
- Giao diện chi tiết đơn hàng

**Xem đơn hàng**

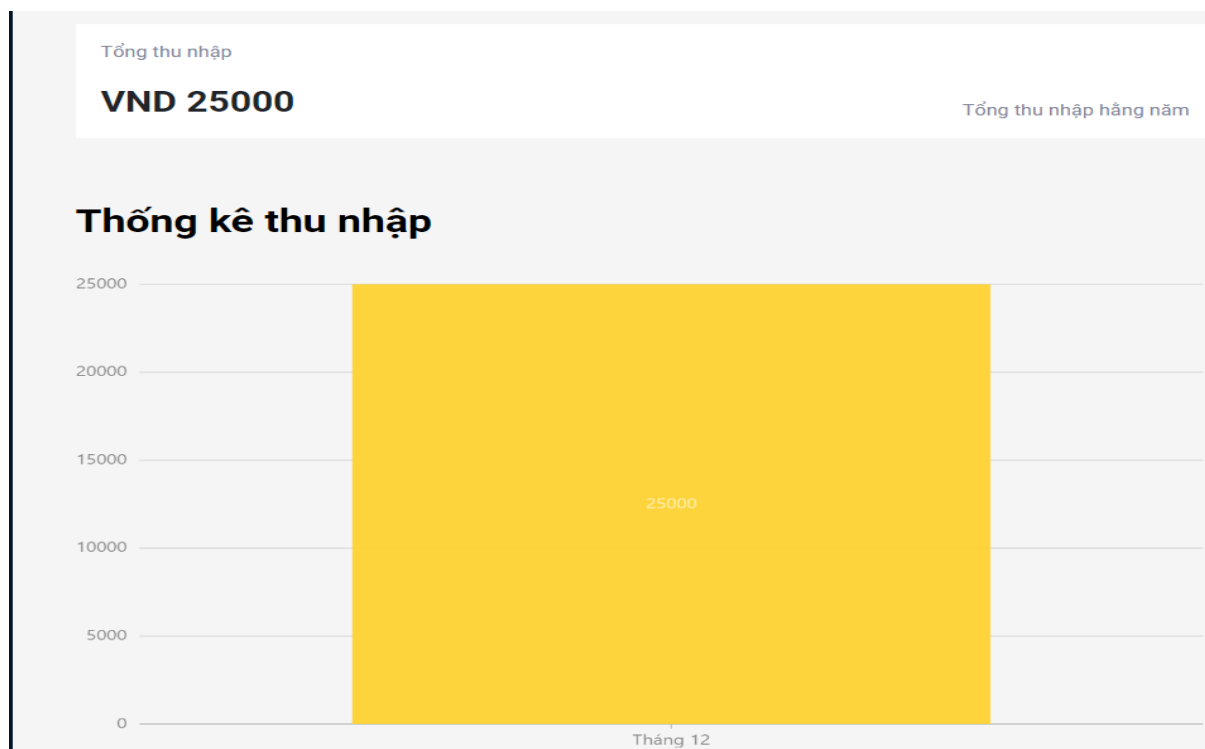
SNo	Tên sản phẩm	Thương hiệu	Số lượng	Phương thức thanh toán	Thông tin giao hàng	Màu	Giá tiền
1	Lô 02 ô tô nhựa tuổi thơ hệ 8x, 9x	MP Oto	1	Cash On Delivery	Cần Thơ	#dd5f9c	25000

< 1 >

*Hình 3. 24 Hình xem đơn hàng*

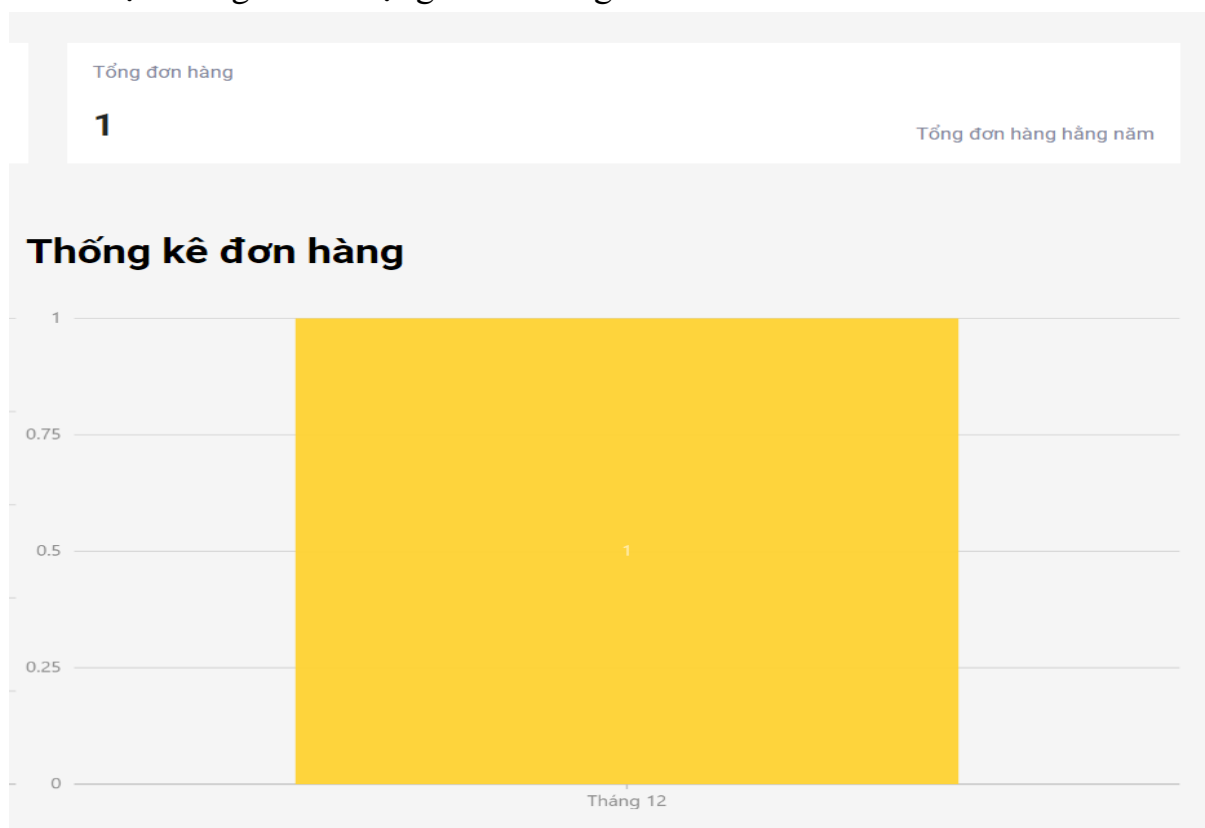
### **3.5.6. Chức năng thống kê**

- Giao diện thống kê theo tiền mỗi tháng



Hình 3. 25 Hình thống kê thu nhập

### Giao diện thống kê số lượng đơn / tháng



Hình 3. 26 Thống kê đơn hàng

### 3.5.7. Giao diện xem đơn hàng đặt vừa qua

Đơn hàng gần đây					
SNo	Tên	Số lượng sản phẩm	Tổng giá	Tổng giá giảm	Trạng thái
0	khang22tran	1	25000	25000	Đang xử lý

Hình 3. 27 Xem đơn hàng gần đây

### 3.5.8. Giao diện quản lý xem tài khoản khách hàng

- Giao diện liệt kê các tài khoản của khách hàng

Khách hàng				
SNo	Tên	Email	Số điện thoại	
2	khang22 tran	vinhkhong325@gmail.com	1234567890	
3	Khang Trần	khong123@gmail.com	1234456346534645	
4	Khang Trần	khong33@gmail.com	12345657890	
5	Trần Khang	khong11@gmail.com	0876543219	

Hình 3. 28 Quản lý xem tài khoản khách hàng

### 3.5.9. Chức năng đăng ký khách hàng

- Giao diện đăng ký:
- Để đăng ký ta cần phải nhập đầy đủ thông tin trong giao diện. Rồi nhấn nút “Tạo”.

Hình 3. 29 Tạo tài khoản

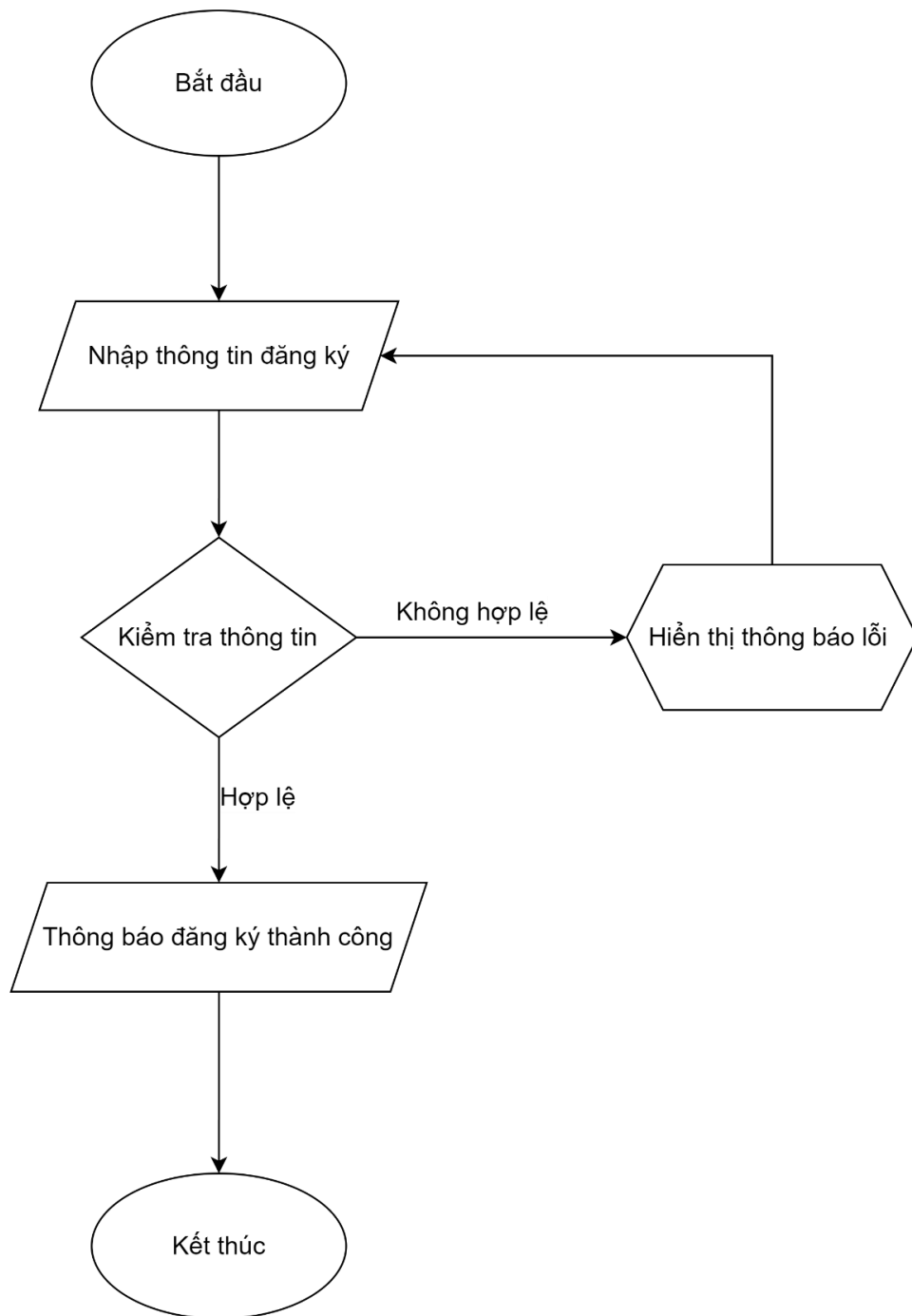
Các thành phần bên trong giao diện:

Bảng 3. 12 Bảng thành phần đăng ký

STT	Loại điều khiển	Nội dung	Lưu ý
1	Input	Nhập họ	
2	Input	Nhập tên	
3	Input	Nhập Email	
4	Input	Nhập SDT	
5	Input	Nhập mật khẩu	

- Cách xử lý





Hình 3. 30 Sơ đồ xử lý đăng ký

#### 3.5.10. Chức năng đăng nhập

- Mục đích: Để người dùng đăng nhập vào hệ thống, xác minh vai trò, quyền hạn của người dùng.
- Giao diện:

The image shows a login form titled "Đăng nhập" (Login). It contains two input fields: "Email" and "Mật khẩu" (Password). Below the password field is a link "Quên mật khẩu?" (Forgot password?). At the bottom are two buttons: "Đăng nhập" (Login) in a dark blue rounded rectangle and "Đăng ký" (Register) in an orange rounded rectangle.

Đăng nhập

Email

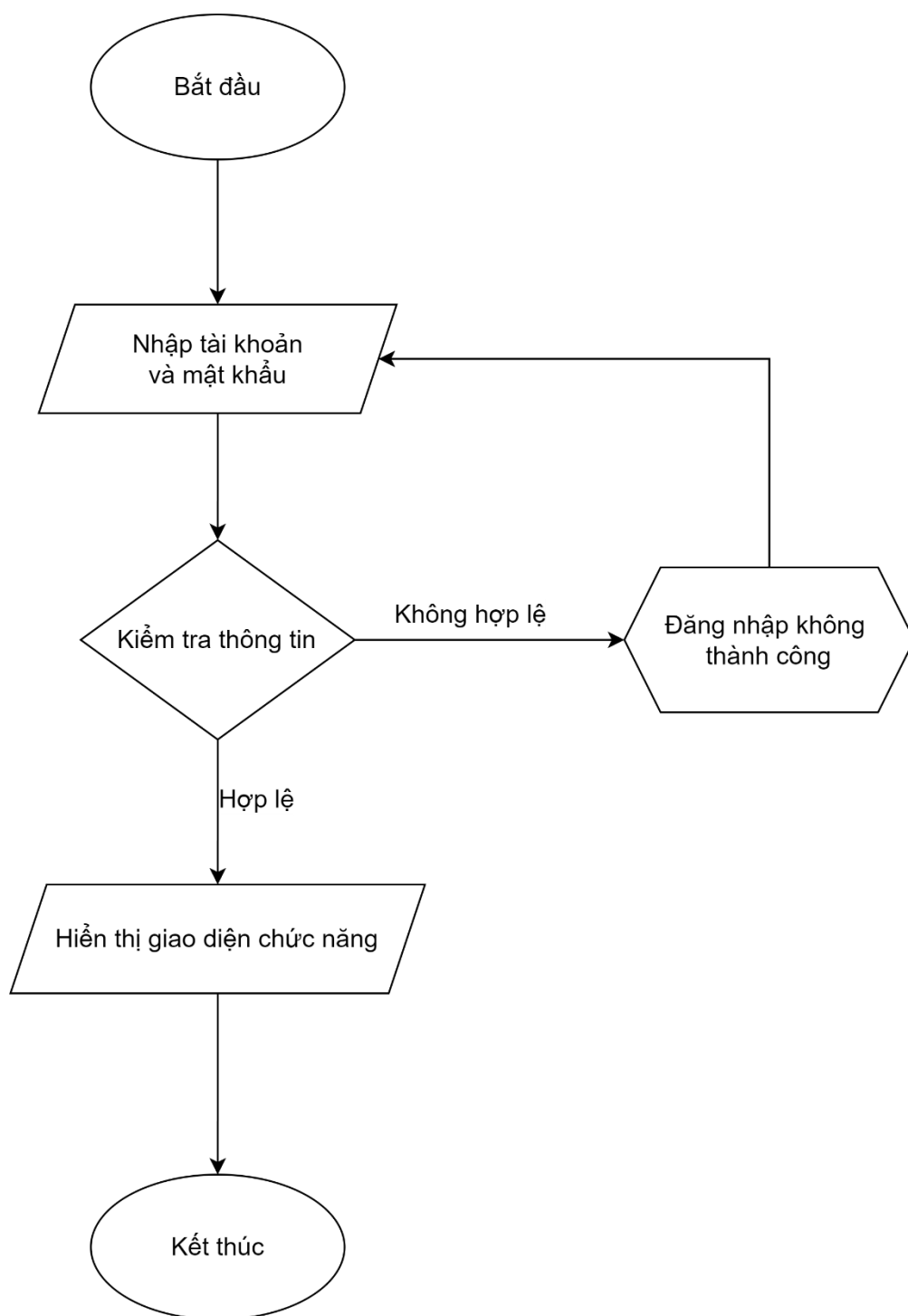
Mật khẩu

[Quên mật khẩu?](#)

Đăng nhập Đăng ký

*Hình 3. 31 Hình đăng nhập*

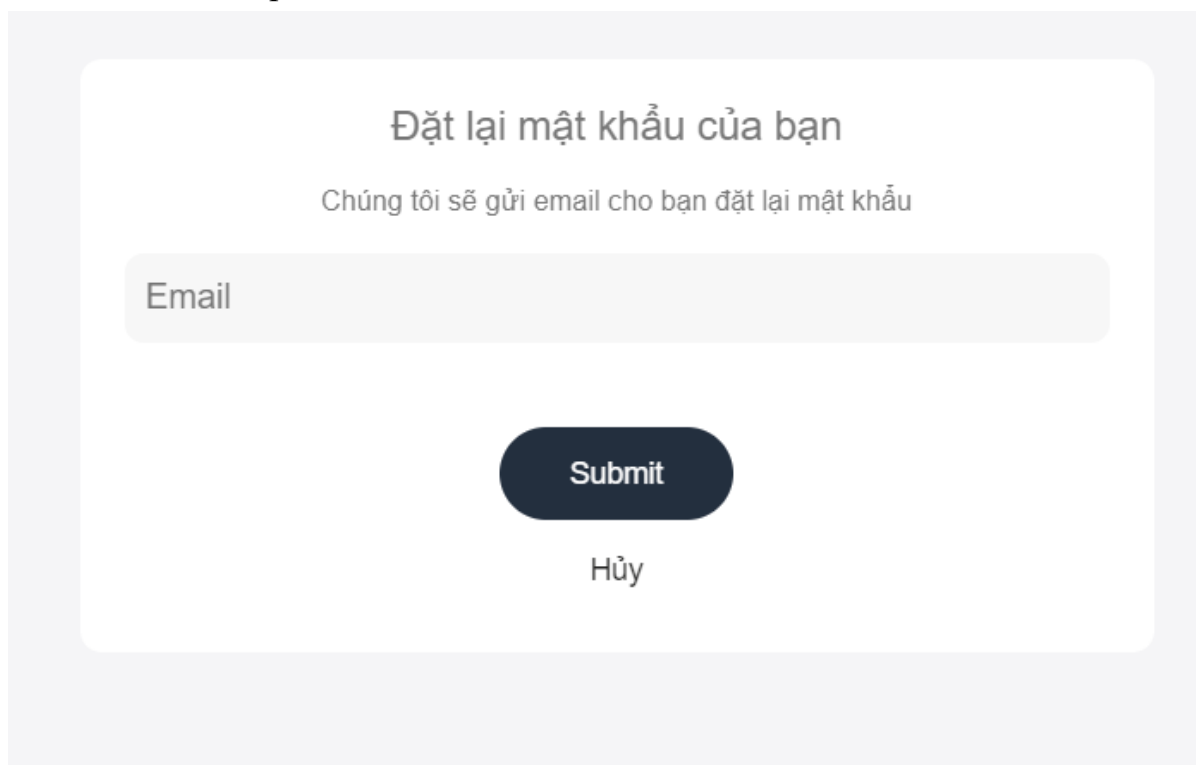
- Cách xử lý



Hình 3. 32 Sơ đồ xử lý đăng nhập

### 3.5.11. Chức năng quên mật khẩu

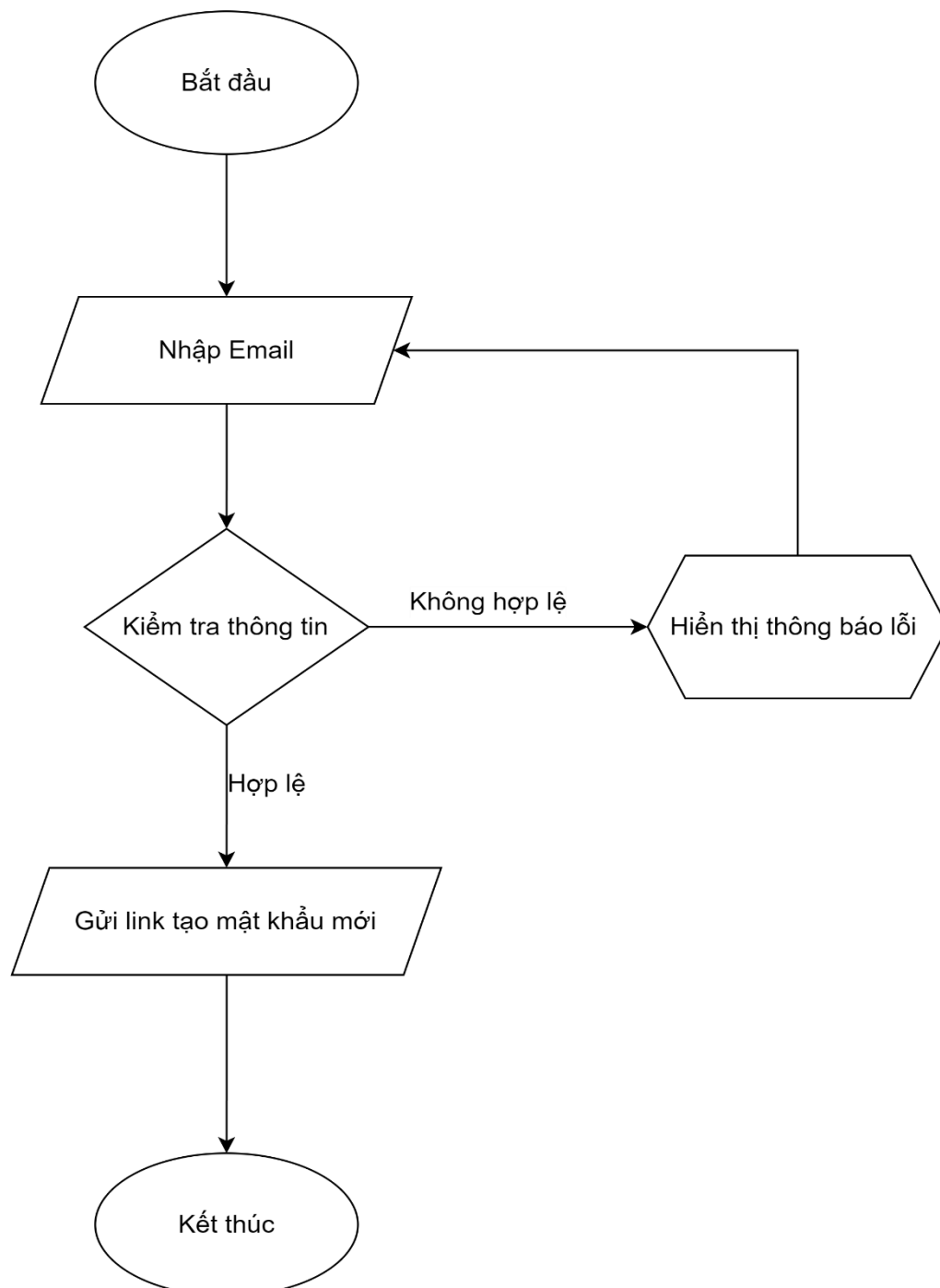
- Giao diện quên mật khẩu



The image shows a web interface for resetting a password. It features a light gray background with a white rounded rectangle in the center. Inside the rectangle, the text "Đặt lại mật khẩu của bạn" (Reset your password) is displayed in a bold, dark gray font. Below this, a smaller line of text reads "Chúng tôi sẽ gửi email cho bạn đặt lại mật khẩu" (We will send you an email to reset your password). There is a light gray input field with the placeholder text "Email". Below the input field, there is a dark blue rounded button with the text "Submit" in white. At the bottom of the white rectangle, there is a link labeled "Hủy" (Cancel) in a dark gray font.

Hình 3. 33 Đặt lại mật khẩu

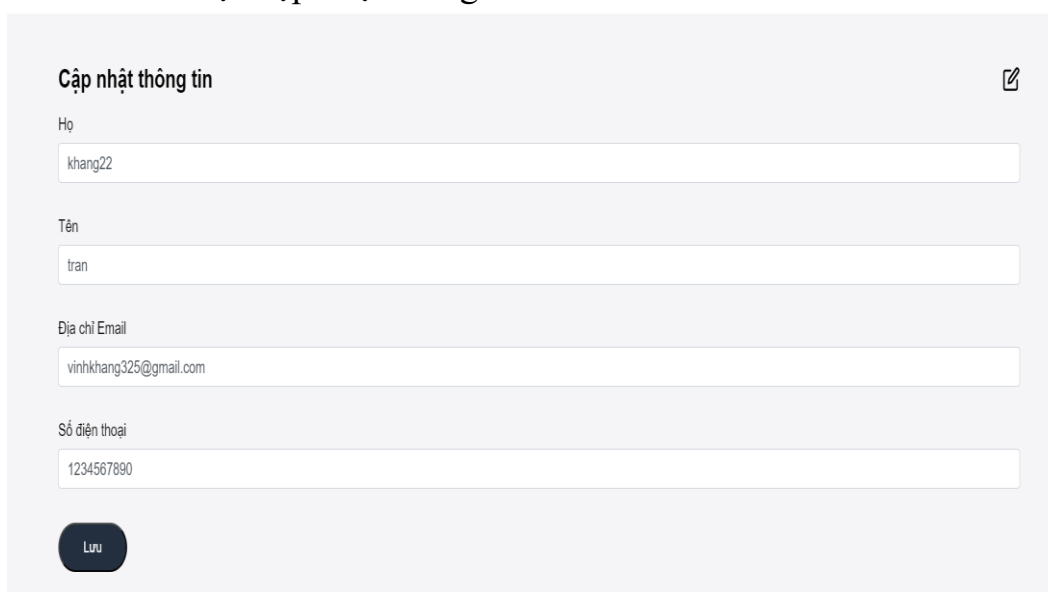
- Cách xử lý




Hình 3. 34 Sơ đồ xử lý quên mật khẩu

### 3.5.12. Chức năng cập nhật thông tin

#### - Giao diện cập nhật thông tin



**Cập nhật thông tin** 

Họ

Tên

Địa chỉ Email

Số điện thoại

**Lưu**

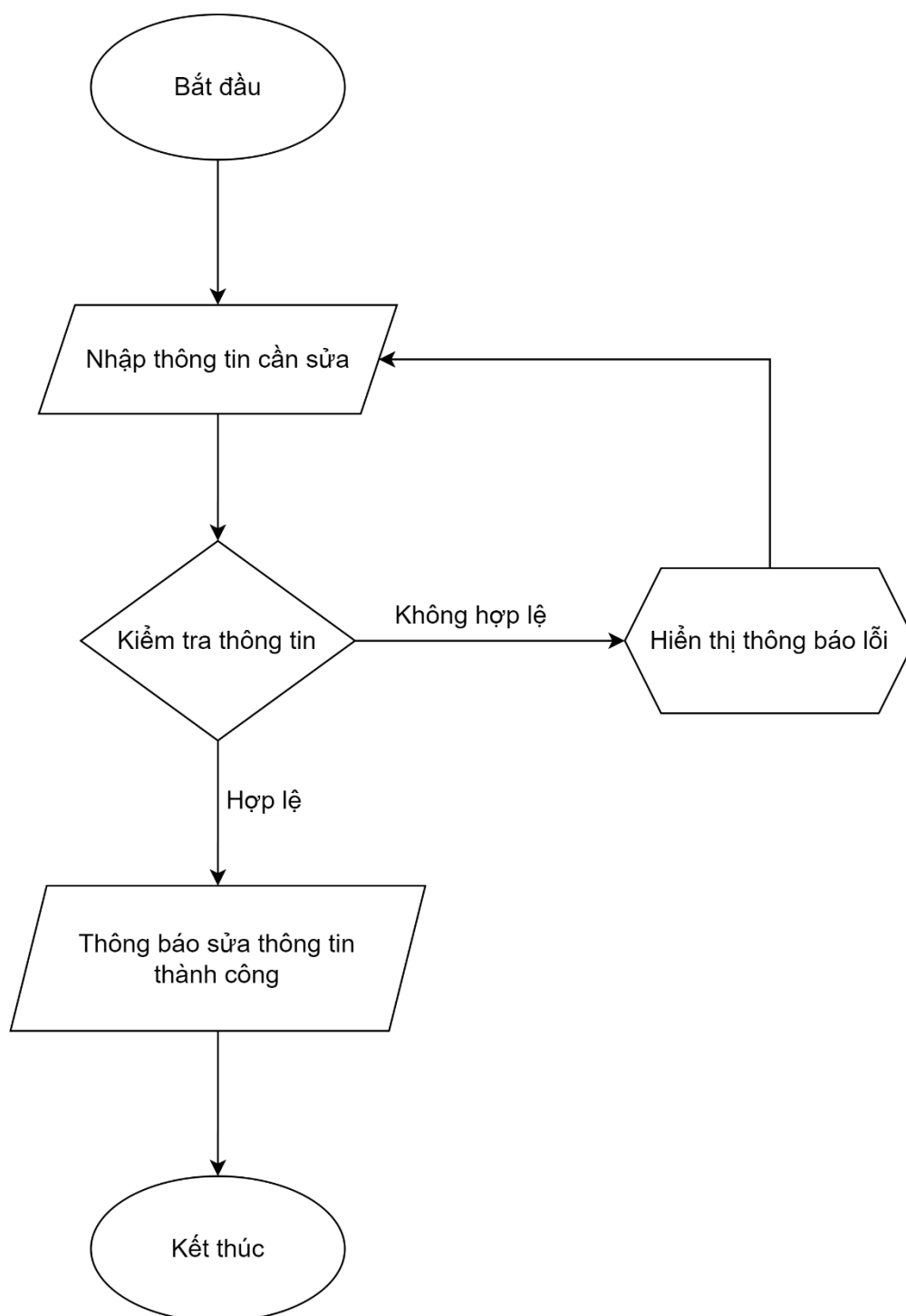
Hình 3. 35 Giao diện cập nhật thông tin

#### - Các thành phần trong giao diện

Bảng 3. 13 Bảng thành phần cập nhật thông tin

STT	Loại điều khiển	Nội dung	Lưu ý
1	Input	Nhập họ	
2	Input	Nhập tên	
3	Input	Nhập Email	
4	Input	Nhập SDT	

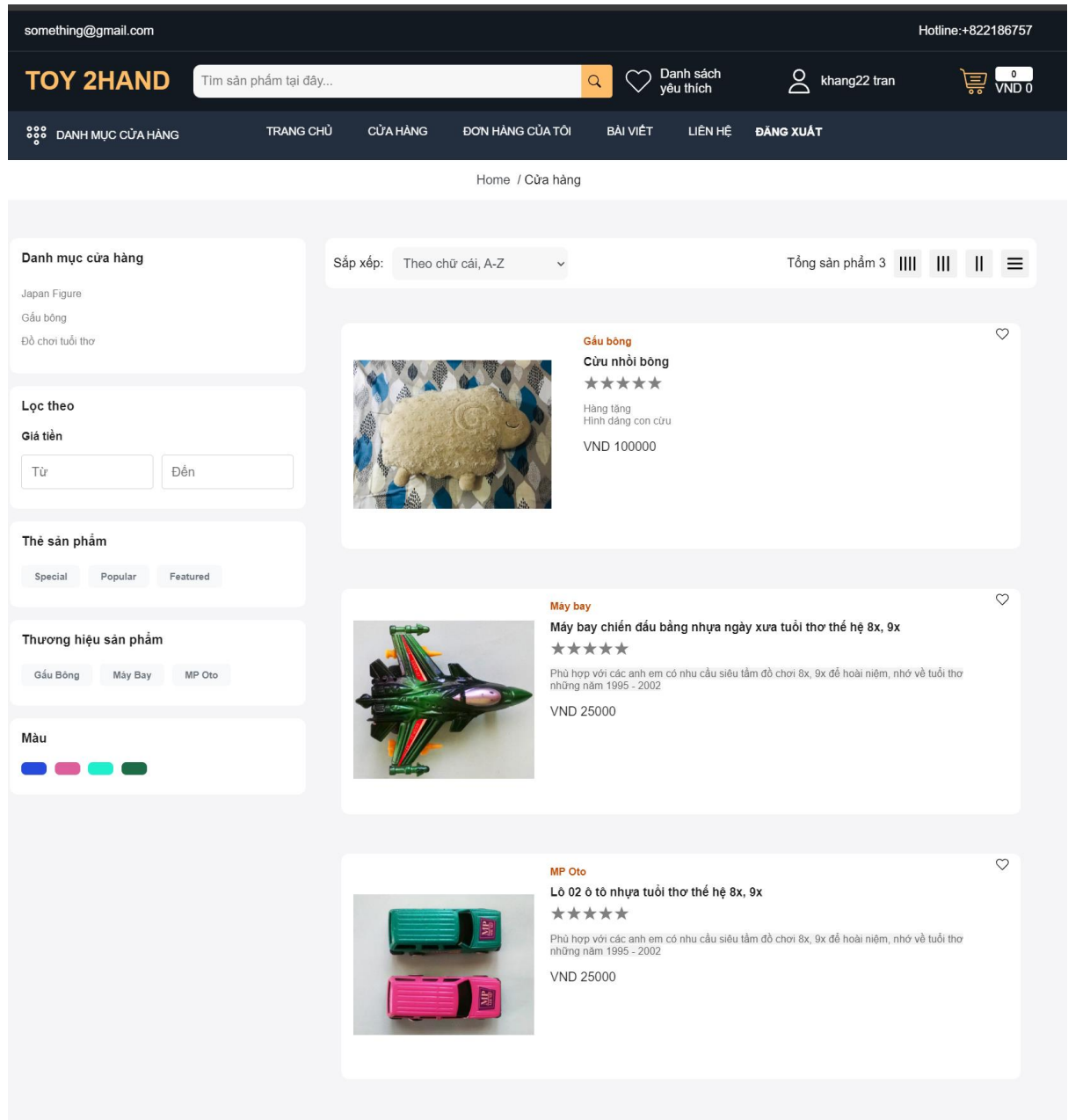
- Cách xử lý



Hình 3. 36 Sơ đồ xử lý cập nhật thông tin

### 3.5.13. Chức năng tìm kiếm sản phẩm

#### - Giao diện tìm kiếm



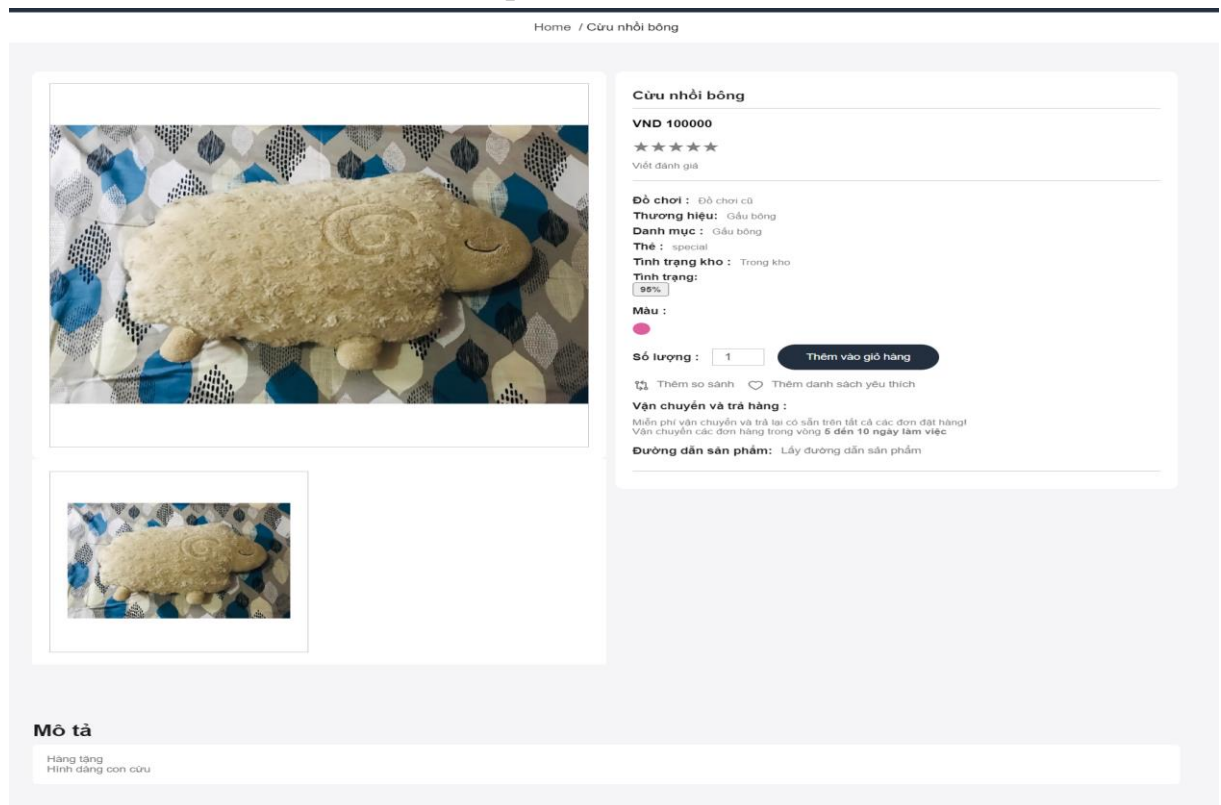
Hình 3. 37 Giao diện tìm kiếm

- Có thể lọc sản phẩm theo giá, theo danh mục sản phẩm, theo thương hiệu hoặc thẻ sản phẩm đang hiển thị.
- Cách tìm kiếm: nhập từ khóa vào ô tìm kiếm rồi nhấn nút tìm kiếm hệ thống sẽ về sản phẩm cần tìm



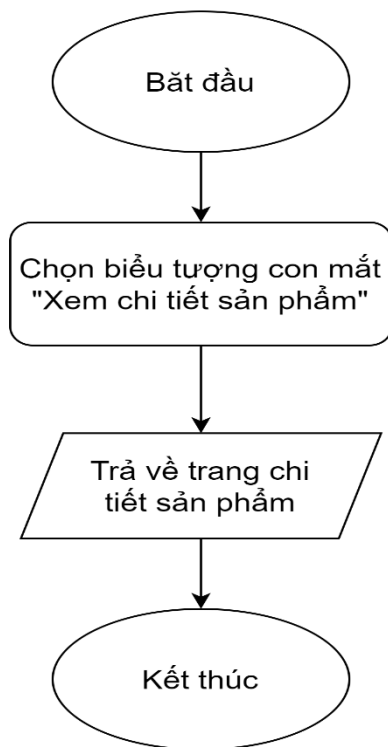
### 3.5.14. Chức năng xem sản phẩm

- Giao diện xem chi tiết sản phẩm



Hình 3. 38 Giao diện xem chi tiết sản phẩm

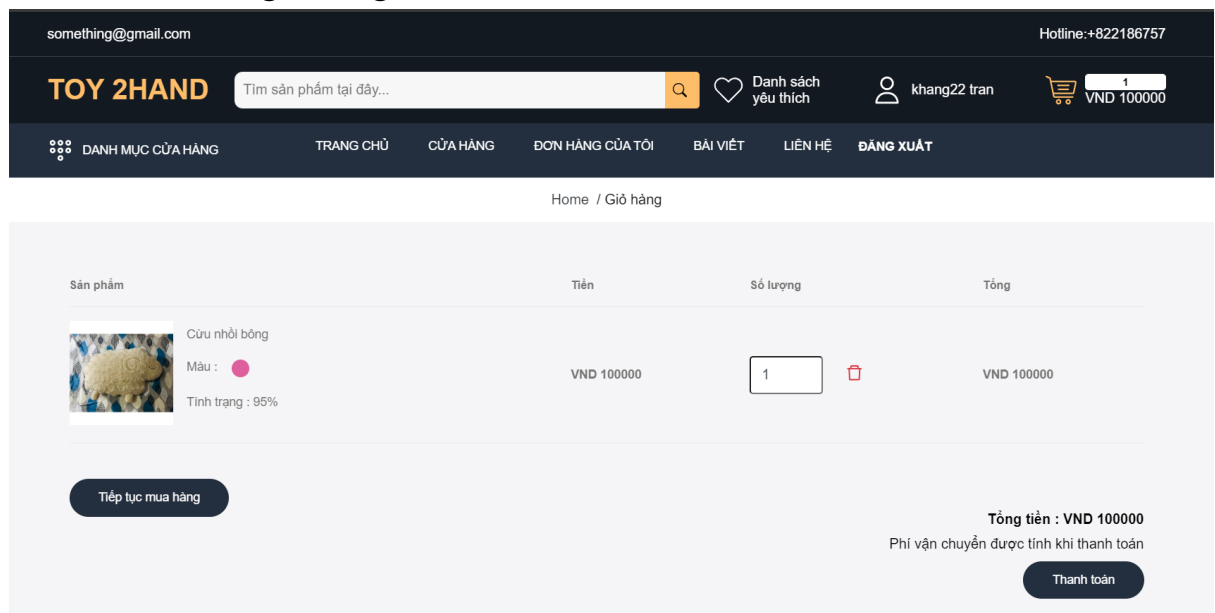
- Trang chi tiết sản phẩm này có thể thêm vào giỏ hàng và thay đổi số lượng sản phẩm mua
- Cách xử lý



Hình 3. 39 Sơ đồ xử lý xem chi tiết đơn hàng

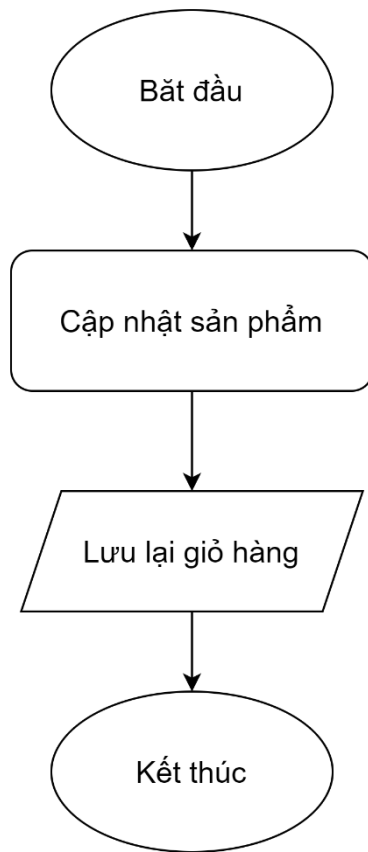
### 3.5.15. Chức năng quản lý giỏ hàng

#### - Giao diện giỏ hàng



Hình 3. 40 Giao diện giỏ hàng

- Cách xử lý



*Hình 3. 41 Sơ đồ xử lý thêm giỏ hàng*

**3.5.16. Chức năng quản lý đặt hàng**

- Sau khi nhấn nút “Đặt hàng” bên trang giỏ hàng hệ thống sẽ chuyển đến trang xác nhận đơn hàng. Nếu nhập đúng thông tin thì nhấn nút thanh toán

Home / Thanh toán

**TOY 2HAND**

Giỏ hàng > Thông tin > Giao hàng > Thanh toán

**Địa chỉ nhận hàng**

Chọn quốc gia

Họ Tên


Địa chỉ

Số nhà

Thành phố Email ZipCode

State

← Trở lại giỏ hàng Tiếp tục mua hàng Thanh toán



Cừu nhồi bông  
#dd5f9c

**\$100000**

Tổng tiền	VND 100000
Tiền ship	VND 30000
<b>Tổng</b>	<b>VND 130000</b>

Hình 3. 42 Giao diện nhập thông tin giao hàng

- Khi kiểm tra chính xác thông tin mình cần đặt thì chúng ta nhấn nút thanh toán

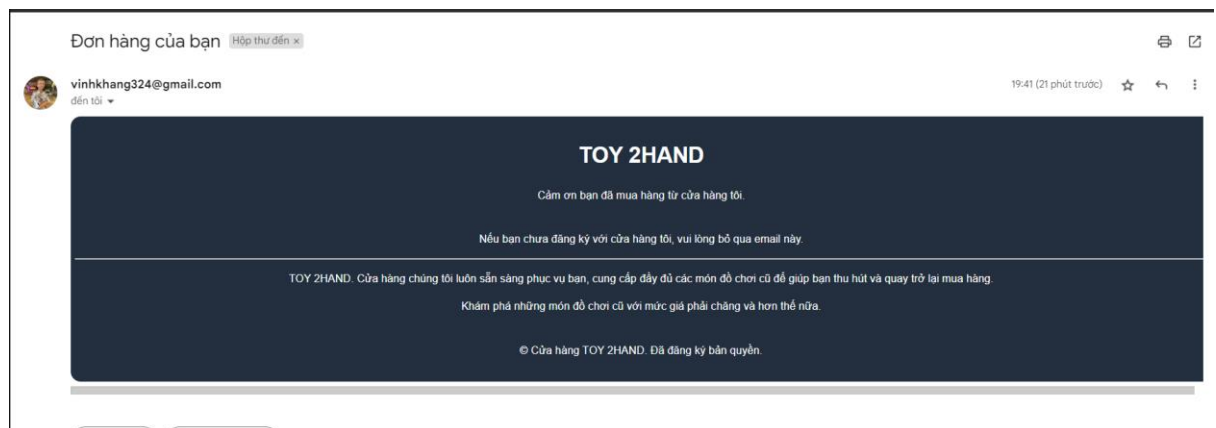
**Chọn hình thức thanh toán**

Thanh toán khi nhận hàng

Đóng

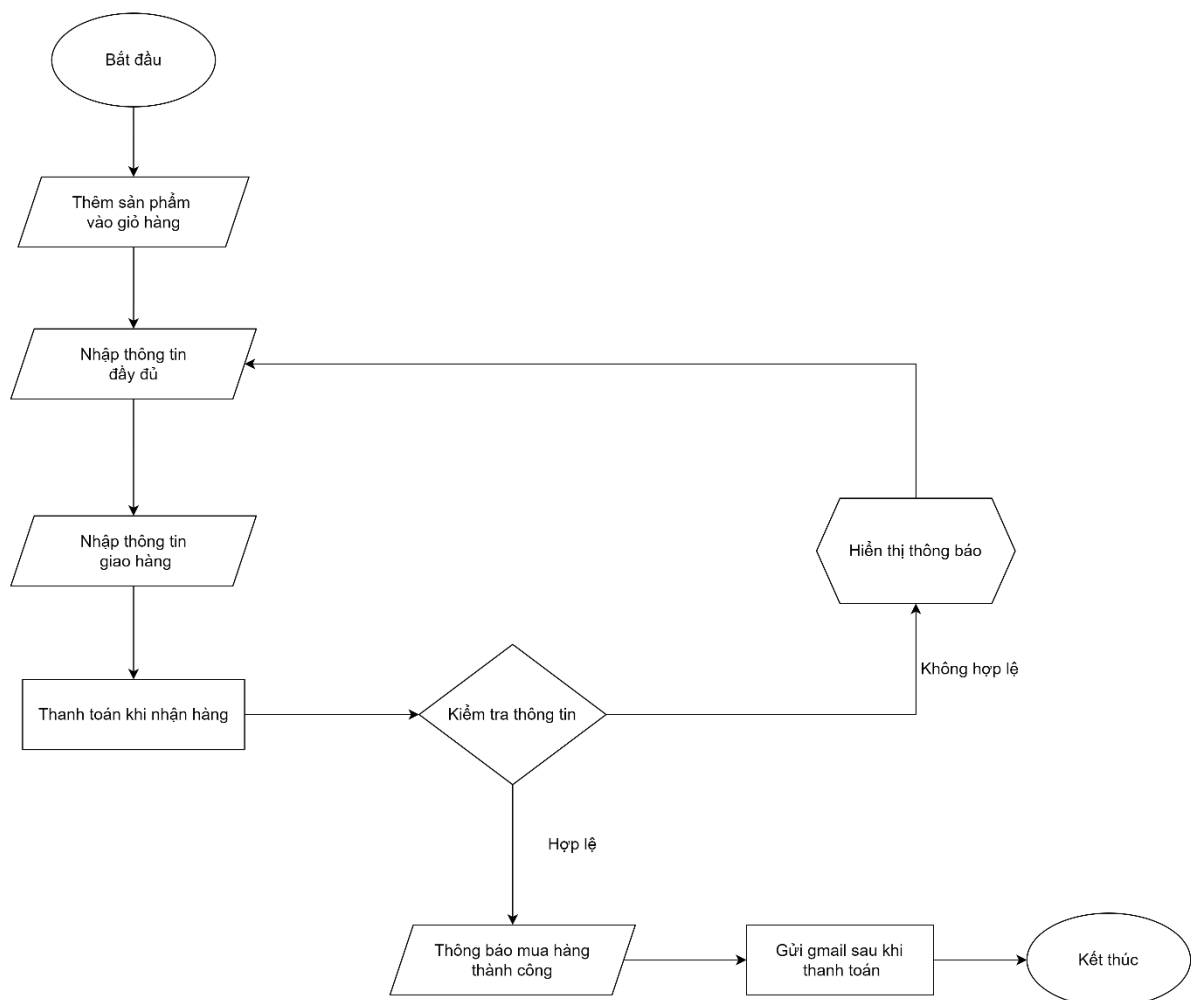
Hình 3. 43 Giao diện thanh toán

- Chọn hình thức thanh toán:
  - + Thanh toán khi nhận hàng
- Kết quả thông báo đặt hàng thành công và được trả gmail cảm ơn mua hàng
-



Hình 3. 44 Giao diện gmail cảm ơn khách hàng

### - Cách xử lý:



Hình 3. 45 Sơ đồ xử lý từ thêm giỏ hàng thanh toán và gửi gmail

### 3.5.17. Chức năng quản lý đơn hàng của khách hàng

- Giao diện quản lý xem lại đơn hàng của khách hàng

Home / Đơn hàng

Mã đơn hàng	Tổng tiền	Tổng tiền giảm giá	Trạng thái	Thông tin nhận hàng
656dc8ed3503374e958efe49	25000	25000	Đang xử lý	Cần Thơ
Tên sản phẩm	Số lượng	Giá tiền	Màu	
Lô 02 ô tô nhựa tuổi thơ thể hệ 8x, 9x	1	25000		

Hình 3. 46 Giao diện xem lại đơn hàng đã đặt của khách hàng

## **Chương 4: KIỂM THỬ VÀ ĐÁNH GIÁ**

Trong chương này sẽ trình bày một cách chi tiết về quá trình kiểm thử, các chức năng được kiểm thử, chi tiết trường hợp kiểm thử của các chức năng, đánh giá sản phẩm nhằm cung cấp tài liệu cho bảo trì và cập nhật sau này.

### **4.1. Kiểm thử**

#### **4.1.1. Mục tiêu**

Mục tiêu chính của kiểm thử là phát hiện lỗi, cũng như kiểm tra chương trình có thỏa được yêu cầu đề ra không

- Phát hiện lỗi và kiểm tra hệ thống có hoạt động đúng với yêu cầu đã nêu ra theo đặc tả chưa.
- Liệt kê kết quả có được sau khi kiểm thử.
- Làm tài liệu cho quá trình cập nhật, chỉnh sửa đề tài.

#### **4.1.2. Phạm vi kiểm thử**

Những trường hợp kiểm thử sẽ tương ứng với các chức năng được mô tả trong tài liệu

### **4.2. Kế hoạch kiểm thử**

#### **4.2.1. Các chức năng được kiểm thử**

- Đăng nhập
- Đăng ký
- Đặt hàng
- Quản lý sản phẩm
- Quản lý danh mục
- Quản lý thương hiệu
- Quản lý đơn hàng

#### **4.2.2. Tiêu chí đình chỉ và yêu cầu đánh giá lại**

Nếu trong quá trình kiểm thử có 45% trường hợp kiểm thử không thành công thì quá trình kiểm thử sẽ được tạm dừng cho đến khi phần mềm khắc phục được tất cả các trường hợp không thành công thì quá trình kiểm thử sẽ được tiếp tục.

Quá trình kiểm thử sẽ được dừng lại khi hầu hết các mục tiêu sau đây được hoàn thành:

- Tất cả các trường hợp kiểm thử được thực hiện ít nhất một lần
- Xác định mức độ hoàn thiện của tất cả các Test Case.
- Xác định số lỗi có đạt được như dự kiến không.

- Tất cả các Fault đều được kiểm tra và khắc phục.
- Cạn kiệt ngân sách dự đoán.
- Thời gian cho kiểm thử đã hết.
- Các tài liệu kiểm thử được xem xét và lưu hành.

#### 4.2.3. Môi trường kiểm thử

Phần cứng:

- CPU: Intel Core I5 12400F
- Ram: 32Gb
- Ổ cứng: 2,5TB
- Card màn hình: RTX 3070 8gb
- Cấu hình mạng: có kết nối Internet

Phần mềm:

- Hệ điều hành: Windows 11 Pro
- Trình duyệt: Google Chrome

#### 4.3. Trách nhiệm và quyền hạn

*Bảng 4. 1 Trách nhiệm và quyền hạn*

Họ và tên	Trách nhiệm					
	Quản lý	Thiết kế	Chuẩn bị	Thực hiện	Chứng kiến	Kiểm tra
Trần Lưu Vĩnh Khang	✓	✓	✓	✓	✓	✓

#### 4.4. Các rủi ro

*Bảng 4. 2 Các rủi ro*

Tên rủi ro	Mức độ	Kế hoạch
Thiếu nhân sự kiểm thử	Cao	Tăng số lượng kiểm thử
Kiểm thử không đúng tiến độ	Thấp	Đẩy nhanh tiến độ kiểm thử
Kiểm thử không hiệu quả	Trung bình	Tham khảo các nguồn tài liệu kiểm thử



#### 4.5. Kịch bản kiểm thử

Bảng 4. 3 Kịch bản kiểm thử

Mã kịch bản	Tên kịch bản kiểm thử	Số trường hợp
KB01	Kiểm thử chức năng đăng nhập	4
KB02	Kiểm thử chức năng đăng ký	3
KB03	Kiểm thử chức năng đặt hàng	3
KB04	Kiểm thử chức quản lý sản phẩm	2
KB05	Kiểm thử chức năng quản lý danh mục	2
KB06	Kiểm thử chức năng quản lý thương hiệu	2
KB07	Kiểm thử chức năng quản lý đơn hàng	5

#### 4.6. Đánh giá kết quả kiểm thử

Bảng 4. 4 Đánh giá kết quả kiểm thử

Mã kịch bản	Kịch bản kiểm thử	Số trường hợp kiểm thử	Số trường hợp thành công	Số trường hợp thất bại
KB01	Kiểm thử chức năng đăng nhập	4	4	0
KB02	Kiểm thử chức năng đăng ký	3	3	0
KB03	Kiểm thử chức năng đặt hàng	3	3	0
KB04	Kiểm thử chức quản lý sản phẩm	2	2	0
KB05	Kiểm thử chức năng quản lý danh mục	2	2	0
KB06	Kiểm thử chức năng quản lý thương hiệu	2	2	0
KB07	Kiểm thử chức năng quản lý đơn hàng	5	5	0

Sau khi tiến hành đánh giá kiểm thử một số chức năng của hệ thống ta sẽ thu được các kết quả sau đây.

#### 4.6.1. Chức năng đăng nhập (KB01)

Bảng 4. 5 Chức năng đăng nhập

Trường hợp kiểm thử	Test case	Kết quả mong đợi	Kết quả thực tế	Thành công / Thất bại
Đăng nhập thông tin hợp lệ	Tài khoản: vinhkhong324@gmail.com Mật khẩu: “khong240800”	Đăng nhập thành công	Đăng nhập thành công	Thành công
Đăng nhập với 2 trường bị bỏ trống	Tài khoản: Null Mật khẩu: Null	Đăng nhập thất bại	Đăng nhập thất bại	Thành công
Đăng nhập với tài khoản sai	Tài khoản: <a href="mailto:vinhkhong325@gmail.com">vinhkhong325@gmail.com</a> Mật khẩu: “khong240800”	Đăng nhập thất bại	Đăng nhập thất bại	Thành công
Đăng nhập với 1 trường bị bỏ trống	Tài khoản: Null Mật khẩu: “240800”	Đăng nhập thất bại	Đăng nhập thất bại	Thành công

#### 4.6.2. Chức năng đăng ký (KB02)

Bảng 4. 6 Chức năng đăng ký

Trường hợp kiểm thử	Test case	Kết quả mong đợi	Kết quả thực tế	Thành công / Thất bại
Nhập thông tin không đúng yêu cầu		Báo lỗi	Báo lỗi	Thành công
Nhập thông tin đúng yêu cầu nhưng email tồn tại		Báo lỗi	Báo lỗi	Thành công
Nhập đúng thông tin		Thành công	Thành công	Thành công

#### 4.6.3. Chức năng đặt hàng (KB03)

Bảng 4. 7 Chức năng đặt hàng

Trường hợp kiểm thử	Test case	Kết quả mong đợi	Kết quả thực tế	Thành công / Thất bại
Bỏ trống không ghi địa chỉ		Báo lỗi	Báo lỗi	Thành công
Không chọn hình thức thanh toán		Báo lỗi	Báo lỗi	Thành công
Nhập đúng thông tin		Thành công	Thành công	Thành công

#### 4.6.4. Chức năng quản lý sản phẩm (KB04)

Bảng 4. 8 Chức năng quản lý sản phẩm

Trường hợp kiểm thử	Test case	Kết quả mong đợi	Kết quả thực tế	Thành công / Thất bại
Nhập thông tin không đúng yêu cầu		Báo lỗi	Báo lỗi	Thành công
Nhập đúng thông tin		Thành công	Thành công	Thành công

#### 4.6.5. Chức năng quản lý danh mục (KB05)

Bảng 4. 9 Chức năng quản lý danh mục

Trường hợp kiểm thử	Test case	Kết quả mong đợi	Kết quả thực tế	Thành công / Thất bại
Nhập thông tin không đúng yêu cầu		Báo lỗi	Báo lỗi	Thành công
Nhập đúng thông tin		Thành công	Thành công	Thành công

#### 4.6.6. Chức năng quản lý thương hiệu (KB06)

Bảng 4. 10 Chức năng quản lý thương hiệu

Trường hợp kiểm thử	Test case	Kết quả mong đợi	Kết quả thực tế	Thành công / Thất bại
Nhập đúng thông tin		Thành công	Thành công	Thành công

#### 4.6.7. Chức năng quản lý đơn hàng (KB007)

Bảng 4. 11 Chức năng quản lý đơn hàng

Trường hợp kiểm thử	Test case	Kết quả mong đợi	Kết quả thực tế	Thành công / Thất bại
Đã đặt hàng		Đổi trạng thái	Như mong đợi	Thành công
Duyệt đơn hàng		Đổi trạng thái	Như mong đợi	Thành công
Giao đơn hàng		Đổi trạng thái	Như mong đợi	Thành công
Đang xử lý		Đổi trạng thái	Như mong đợi	Thành công
Đã giao hàng		Đổi trạng thái	Như mong đợi	Thành công

#### 4.7. Kết quả kiểm thử

- Kết quả kiểm thử dựa trên 7 kịch bản và 21 trường hợp kiểm thử. Số trường hợp thành công là 21/21, số trường hợp thất bại 0/21. Đạt tỉ lệ kiểm thử thành công 100%.
- Chức năng theo kế hoạch và kịch bản kiểm thử hoạt động đúng như mong đợi
- Không có chức năng nào phải đình chỉ và bắt đầu lại

#### 4.8. Chức năng chưa kiểm thử

- Đăng xuất
- Đăng nhập tài khoản khách hàng
- Cập nhật thông tin khách hàng
- Đánh giá sản phẩm

## **Phần 3: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **1.1. Kết quả đạt được**

#### **1.1.1. Về kiến thức**

Giúp bản thân hiểu thêm về quy trình, cách vận hành thực tế của một hệ thống thương mại điện tử, biết thêm về thư viện ReactJS cũng như cách sử dụng mô hình MVC. Giúp bản thân biết thêm được một số thư viện hỗ trợ thiết kế giao diện như Antd Design, HTML, CSS.

Có khả năng phân tích, thiết kế, lập trình với ReactJS. Sử dụng tốt các công cụ hỗ trợ cho việc phác thảo và lập trình. Hiểu được phương pháp kiểm thử phần mềm, có thể áp dụng vào sản phẩm của mình.

Nâng cao kinh nghiệm lập trình, tư duy sáng tạo, khả năng phân tích và giải quyết vấn đề, phân tích hệ thống, tìm hiểu và học hỏi thêm các công nghệ mới.

#### **1.1.2. Về kỹ năng**

Về cơ bản “Website bán đồ chơi cũ” được phát triển tương đối hoàn chỉnh, hoàn thành tốt các yêu cầu đặt ra, xây dựng được đầy đủ chức năng cho từng người dùng khác nhau.

Phát triển website với giao diện thu hút, bắt mắt và khả năng tương thích tốt với nhiều thiết bị giúp khách hàng có thể dễ dàng theo dõi và mua sắm trên trang web.

Trang web cho phép người dùng thanh toán trực tiếp thuận lợi. Đặc biệt website còn được tích hợp Zoom Ảnh và đăng video nhằm nâng cao trải nghiệm khách hàng lên một tầm cao mới, giúp người dùng có thể trải nghiệm sản phẩm chân thực hơn khi mua sắm online.

#### **1.1.3. Hạn chế**

Hệ thống còn hạn chế về mặt giao diện, chưa thể chia sẻ các sản phẩm qua các mạng xã hội.

#### **1.1.4. Khả năng ứng dụng thực tiễn**

### **1.2. Hướng phát triển**

Tích hợp chức năng đăng nhập từ tài khoản mạng xã hội Facebook, zalo.

Phát triển hệ thống trên nền tảng Android và IOS, khách hàng không cần truy cập vào website, mà chỉ cần truy cập vào ứng dụng di động, giúp khách hàng dễ dàng mua sắm mọi lúc mọi nơi.

Xây dựng thêm các chức năng cho bên phía quản trị viên để họ có thể hỗ trợ tốt hơn việc quản lý trang web cũng như cửa hàng kinh doanh.

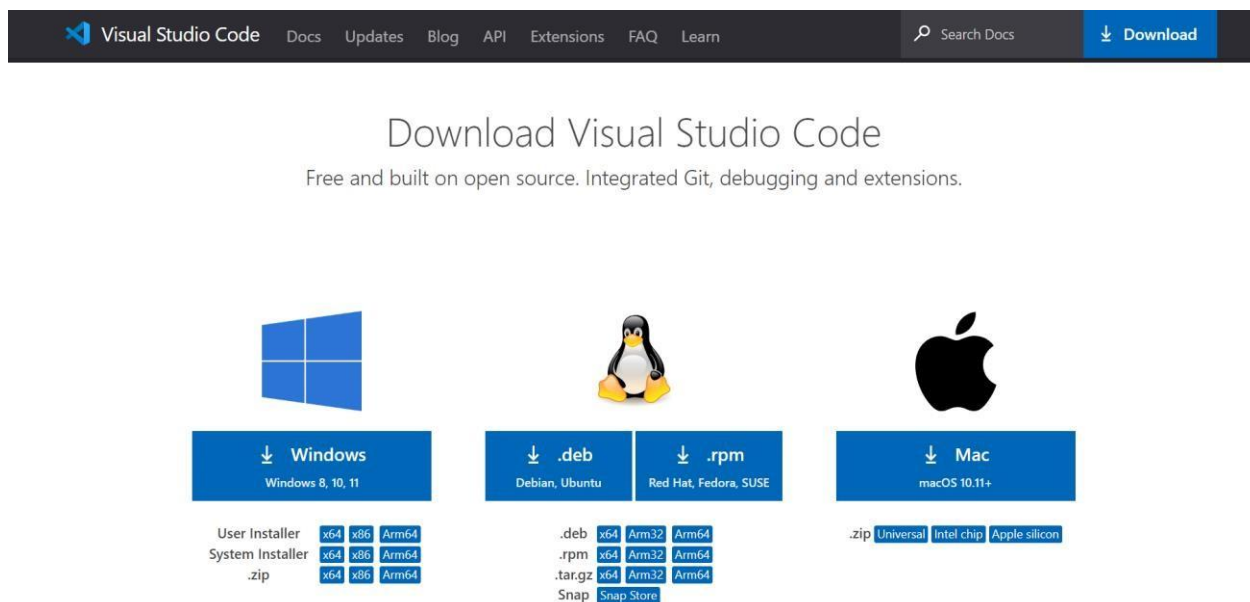
## PHỤ LỤC

### PHỤ LỤC: HƯỚNG DẪN CÀI ĐẶT

Cài đặt Studio Code và môi trường chạy Cài đặt Visual Studio Code:

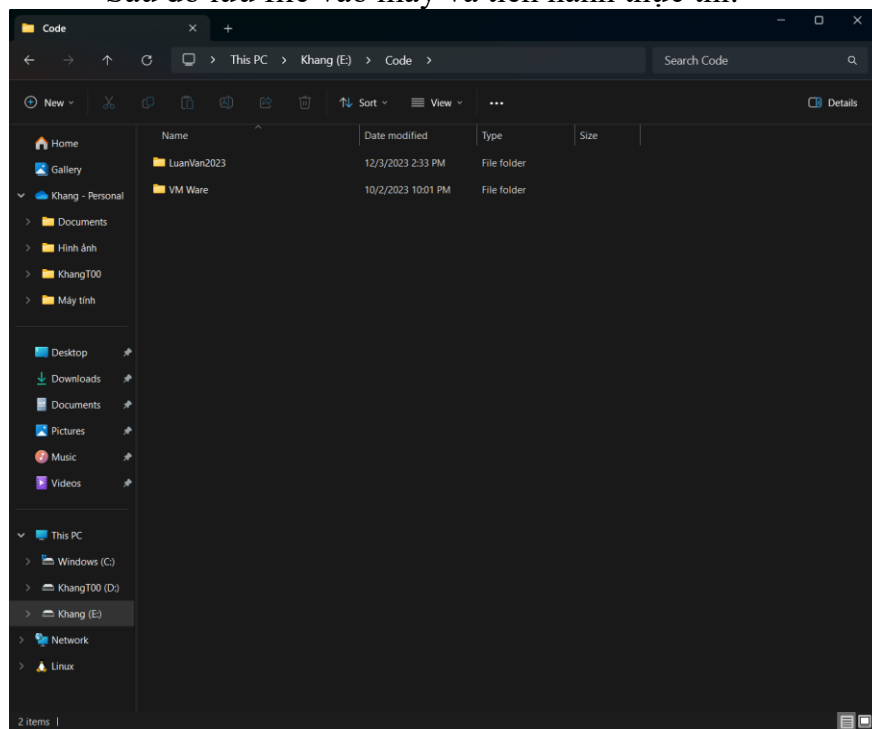
Đầu tiên truy cập vào trang chủ để tải về Visual Studio Code tương thích với phiên bản hệ thống 32 bit hay 64 bit:

<https://code.visualstudio.com/download>



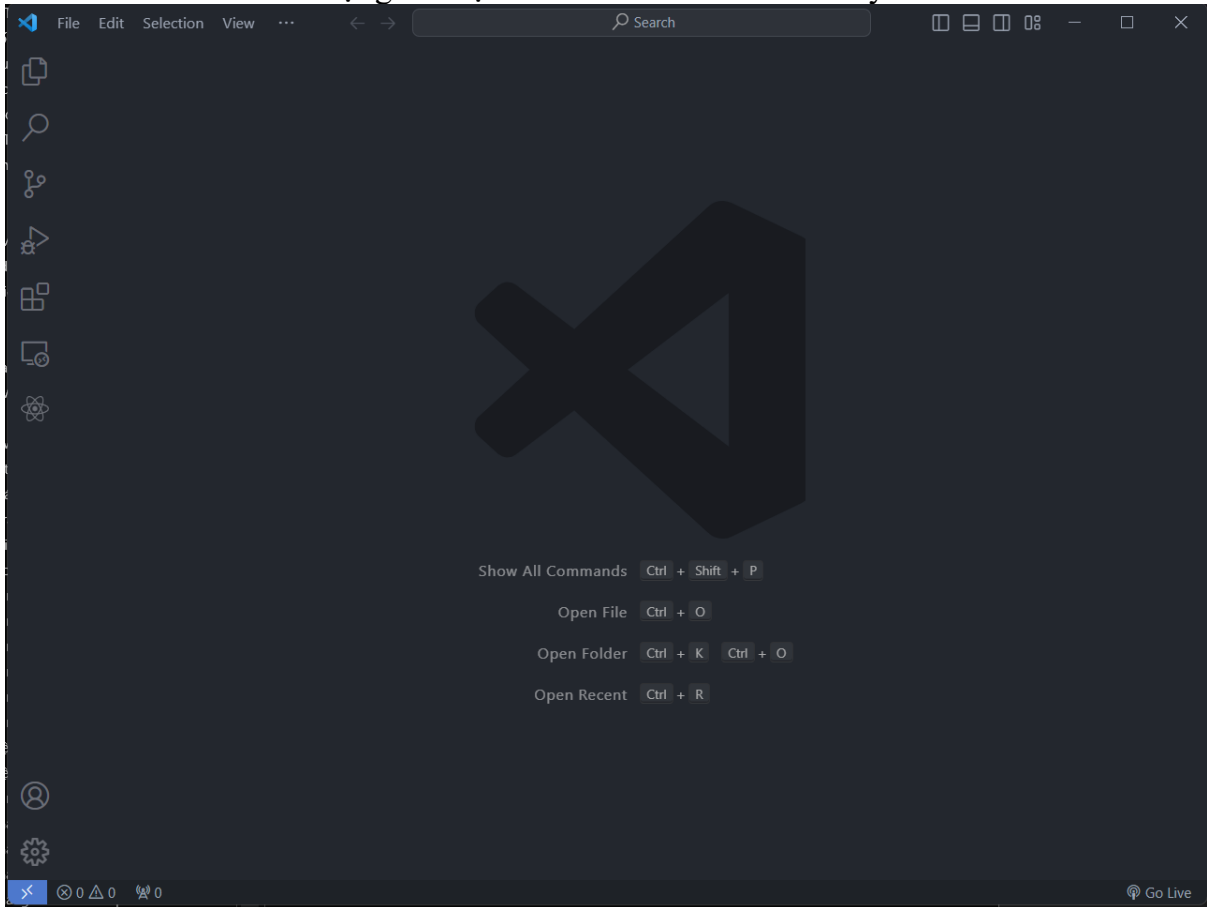
Phụ lục 1: Hình website tải Visual Studio Code

- Sau đó lưu file vào máy và tiến hành thực thi.



Phụ lục 2: Hình thư mục chứa dự án

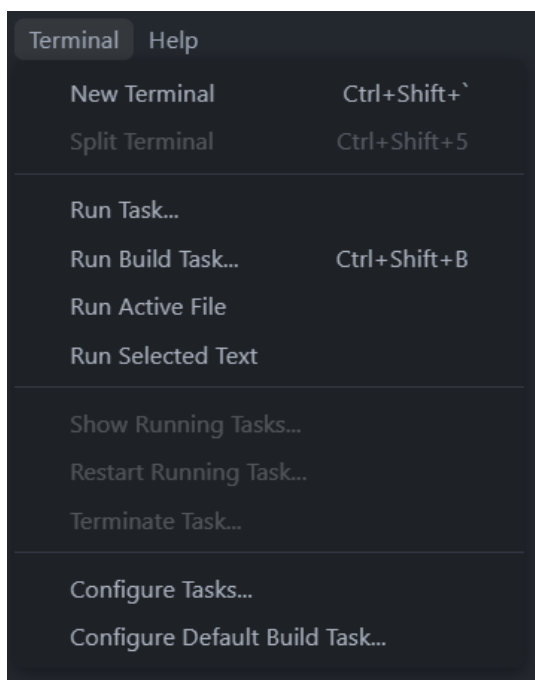
- Sau khi cài đặt giao diện Visual code sẽ như thế này:



Phụ lục 3: Hình giao diện Visual Studio Code

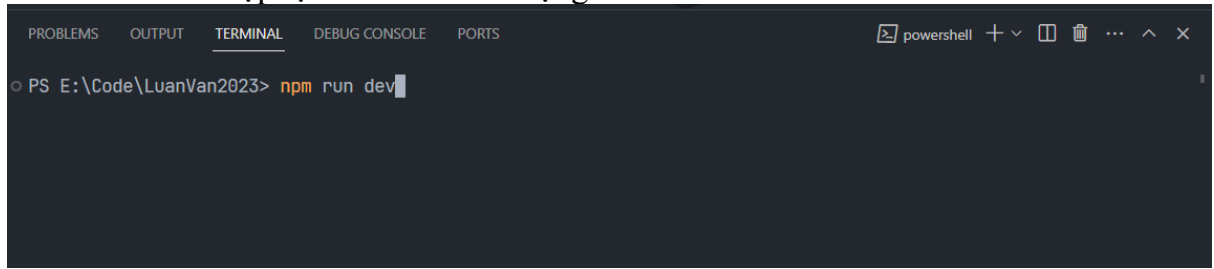
### Cài đặt môi trường

Khi vào giao diện Visual Code chọn, chọn Terminal để cài đặt biến môi trường:



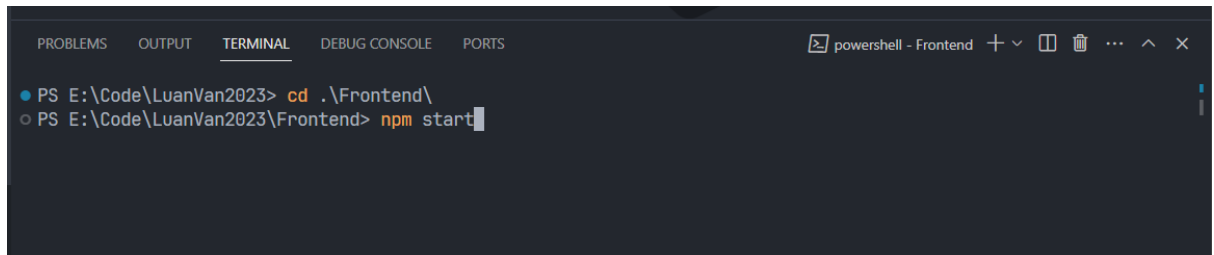
Phụ lục 4: Hình mở terminal của Visual Studio Code

- Sau đó nhập lệnh sau để khởi động server:



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS powershell + - [ ] [ ] ... ^ X
PS E:\Code\LuanVan2023> npm run dev
```

*Phụ lục 5 Hình chạy code ở nhánh dev trong terminal*



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS powershell - Frontend + - [ ] [ ] ... ^ X
PS E:\Code\LuanVan2023> cd .\Frontend\
PS E:\Code\LuanVan2023\Frontend> npm start
```

*Phụ lục 6: Hình terminal khởi chạy dự án ở thư mục Font End*

Sau đó truy cập đường dẫn: <http://172.0.0.1:3000> để vào website.



## TÀI LIỆU THAM KHẢO

- [1] Nguyễn Công Danh, Trần Cao Đệ. Giáo trình đảm bảo chất lượng phần mềm, Khoa CNTT & TT – Đại học Cần Thơ
- [2] Huỳnh Xuân Hiệp và Phan Phương Lan (2011) – Giáo trình nhập môn Công Nghệ Phần Mềm, Đại học Cần Thơ 2011
- [3] Huỳnh Xuân Hiệp, Võ Huỳnh Trâm, Phan Phương Lan, Huỳnh Quang Nghi. Giáo trình Kiến trúc và Thiết kế phần mềm. Nhà Xuất bản Đại Học Cần Thơ, 2015.
- [4] Tìm hiểu về Reactjs: <https://react.dev/learn>
- [5] Tìm hiểu về Cloudinary: <https://cloudinary.com/documentation>
- [6] Tìm hiểu Axios: <https://axios-http.com/vi/docs/intro>
- [7] Tìm hiểu Nodejs: <https://nodejs.org/en/learn>
- [8] Tìm hiểu mô hình MVC: <https://viblo.asia/p/tat-tan-tat-ve-mo-hinh-mvc-Rk74avjAJeO>
- [9] Tìm hiểu về Ant Design: <https://viblo.asia/p/front-end-cung-tim-hieu-ve-ant-design-mot-thu-vien-dac-luc-cua-front-end-1VgZv00M5Aw>