BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC GIA ĐỊNH**

**KHOA: CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN**

**ỨNG DỤNG BÁN ĐIỆN THOẠI PHONE HORIZON**

**MÔN: LẬP TRÌNH CHO THIẾT BỊ DI ĐỘNG NÂNG CAO**

Ngành: **Công Nghệ Phần Mềm**

Giảng viên hướng dẫn: **LÊ HUỲNH PHƯỚC**

Sinh viên thực hiện: **Ngô Trung Nghĩa**

MSSV: **2104110019**

Lớp: **K15DCPM01**

TP. Hồ Chí Minh, ngày 16 tháng 04 năm 2024

**Khoa: Công nghệ thông tin**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

**TIỂU LUẬN MÔN: LẬP TRÌNH CHO THIẾT BỊ DI ĐỘNG NÂNG CAO**

1. **Họ và tên sinh viên:** Ngô Trung Nghĩa
2. **Tên đề tài:** Ứng dụng bán điện thoại Phone Horizone
3. **Nhận xét:**
4. **Kết quả đạt được:**

1. **Những hạn chế:**

1. **Điểm đánh giá:**

Sinh viên: Ngô Trung Nghĩa

Điểm số:……………….. Điểm chữ:…………………………………………….

TP.HCM, ngày… tháng… năm 20….

**Giảng viên chấm thi**

(Ký và ghi rõ họ tên)

# MỞ ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của khoa học công nghệ, Công nghệ Thông tin là một trong những ngành có vị thế dẫn đầu và có vai trò rất lớn trong sự phát triển chung đó. Các ứng dụng của công nghệ thông tin được áp dụng trong mọi lĩnh vực nghiên cứu khoa học cũng như trong mọi lĩnh vực của đời sống. Là một phần của Công nghệ Thông tin, Công nghệ web App đang có được sự phát triển mạnh mẽ và phổ biến rất nhanh bởi những lợi ích mà nó mang lại cho cộng đồng là rất lớn.

Bằng việc lựa chọn và thực hiện đề tài “***Xây dựng ứng dụng bán điện thoại Phone Horizone”,*** chúngem muốn tìm hiểu và đưa ra một giải pháp tốt nhằm giải quyết công việc bán hàng trực tuyến. Thông qua việc thực hiện xây dựng website này chúng em học hỏi nhiều kiến thức và hiểu được quy trình cơ bản để xây dựng một App Mobile hoàn chỉnh.

Ứng dụng bán điện thoại giúp cho khách hàng những lựa chọn linh hoạt và tiện lợi trong việc tìm mua sản phẩm thông qua chức năng tìm kiếm và giỏ hàng. Các thông tin về sản phẩm được hiển thị chi tiết với từng sản phẩm, từ đó khách hàng dễ dàng nhận biết và lựa chọn được thứ mình cần. Tuy chỉ là một đồ án nhỏ nhưng nó đã giúp chúng em học tập được nhiều kiến thức và công nghệ mới.

# LỜI CẢM ƠN

Trong suốt quá trình thực hiện đồ án môn lập trình cho thiết bị di động nâng cao, nhóm em đã được sự hỗ trợ hướng dẫn tận tình của thầy Lê Huỳnh Phước. Em xin trân trọng cảm ơn thầy đã luôn theo dõi quá trình tìm kiếm tài liệu và hướng dẫn nhóm thực hiện đồ án “***Xây dựng ứng dụng bán điện thoại Phone Horizone***”.

Do hạn chế về thời gian, chiều sâu về kiến thức, kinh nghiệm thực tế nên không tránh khỏi những thiếu xót trong quá trình thiết kế và thực thi ý tưởng. Chúng em rất mong nhận được sự đóng góp ý kiến, xem xét và chỉ dẫn thêm của thầy để đồ án này được hoàn thiện hơn.

Bên cạnh đó là sự chia sẻ kiến thức, kinh nghiêm của các anh chị khóa trước, bạn bè đã góp phần hỗ trợ hoàn thành ứng dụng này.

Nhóm xin chân thành cảm ơn!

# MỤC LỤC

[MỞ ĐẦU 2](#_Toc164229632)

[LỜI CẢM ƠN 3](#_Toc164229633)

[MỤC LỤC 4](#_Toc164229634)

[Chương I: Tổng quan 7](#_Toc164229635)

[1.1 Giới thiệu chung về lập trình ứng dụng di động: 7](#_Toc164229636)

[1.1.1 Khái niệm 7](#_Toc164229637)

[1.1.2 Các loại ứng dụng di động chính 7](#_Toc164229638)

[1.2 Giới thiệu đề tài 9](#_Toc164229639)

[1.3 Giới thiệu một số ngôn ngữ và frameword để xây dựng ứng dụng. 9](#_Toc164229640)

[1.3.1 Node.js 9](#_Toc164229641)

[1.3.2 MongoDB 10](#_Toc164229642)

[1.3.3 React Native 11](#_Toc164229643)

[1.4 Lý do chọn để tài 12](#_Toc164229644)

[1.5 Giới thiệu về project 13](#_Toc164229645)

[1.6 Tính cấp thiết của đề tài: 13](#_Toc164229646)

[1.6.1 Mục tiêu: 13](#_Toc164229647)

[1.6.2 Nội dung chính: 13](#_Toc164229648)

[1.6.3 Kết quả chính đạt được 14](#_Toc164229649)

[Chương II: Phân tích thiết kế hệ thống App 15](#_Toc164229650)

[1.7 Đặt tả yêu cầu về nghiệp vụ: 15](#_Toc164229651)

[1.7.1 Yêu cầu chức năng: 15](#_Toc164229652)

[1.7.2 Yêu cầu phi chức năng 15](#_Toc164229653)

[1.8 Sơ đồ use case : 17](#_Toc164229654)

[1.8.1 Use case tổng quát: 17](#_Toc164229655)

[1.8.2 Đặc tả use case: 19](#_Toc164229656)

[1.9 Sơ đồ Sequence 24](#_Toc164229657)

[1.9.1 Đăng nhập 24](#_Toc164229658)

[1.9.2 Đăng ký 25](#_Toc164229659)

[1.9.3 Thêm sản phẩm: 26](#_Toc164229660)

[1.9.4 Cập nhật sản phẩm: 27](#_Toc164229661)

[1.9.5 Xóa sản phẩm: 28](#_Toc164229662)

[1.9.6 Quản lý đơn đặt hàng 29](#_Toc164229663)

[1.9.7 Xem danh sách các sản phẩm bày bán 30](#_Toc164229664)

[1.9.8 Đặt hàng 32](#_Toc164229665)

[1.9.9 Quản lý giỏ hàng: 33](#_Toc164229666)

[Chương III: Thiết kế và xây dựng ứng dụng: 35](#_Toc164229667)

[1.10 Thiết kế cơ sở dữ liệu: 35](#_Toc164229668)

[1.11 Thiết kế giao diện 36](#_Toc164229669)

[1.11.1 Màn hình Trang chủ: 36](#_Toc164229670)

[1.11.2 Màn hình đăng nhập 37](#_Toc164229671)

[1.11.3 Màn hình đăng kí 38](#_Toc164229672)

[1.11.4 Màn hình xem chi tiết sản phẩm: 39](#_Toc164229673)

[1.11.5 Màn hình chi tiết giỏ hàng và tiết hành checkout: 40](#_Toc164229674)

[1.11.6 Màn hình thêm sản phẩm mới 41](#_Toc164229675)

[1.11.7 Màn hình quản lý sản phẩm 42](#_Toc164229676)

[1.11.8 Màn hình cập nhật sản phẩm 43](#_Toc164229677)

[Chương IV: Tổng kết: 44](#_Toc164229678)

[1.12 Kết quả đạt được: 44](#_Toc164229679)

[1.13 Hướng phát triển của ứng dụng: 44](#_Toc164229680)

[1.14 Hạn chế: 44](#_Toc164229681)

[1.15 Phân công công việc: 44](#_Toc164229682)

[1.16 Tài liệu tham khảo: 45](#_Toc164229683)

# Chương I: Tổng quan

## Giới thiệu chung về lập trình ứng dụng di động:

### Khái niệm

Lập trình ứng dụng di động là quá trình tạo ra các ứng dụng di động chạy trên các thiết bị thông minh như điện thoại thông minh và máy tính bảng. Các ứng dụng di động có thể được sử dụng cho nhiều mục đích khác nhau, chẳng hạn như chơi game, nghe nhạc, xem phim, đọc sách, mua sắm, quản lý tài chính, v.v.

Một số ngôn ngữ phổ biến nhất bao gồm: Java: Đây là ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng di động Android. Swift: Đây là ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng di động IOS. Kotlin: Đây là một ngôn ngữ lập trình đa nền tảng có thể được sử dụng để phát triển ứng dụng di động Android, iOS và Windows. React Native: Đây là một nền tảng phát triển ứng dụng di động đa nền tảng sử dụng JavaScript để viết mã. Flutter: Đây là một nền tảng phát triển ứng dụng di động đa nền tảng sử dụng Dart để viết mã.

### Các loại ứng dụng di động chính

Có hai loại ứng dụng di động chính:

Ứng dụng di động gốc: Được viết bằng ngôn ngữ lập trình cụ thể cho từng nền tảng, chẳng hạn như Java cho Android và Swift cho iOS. Ứng dụng di động gốc thường có hiệu suất tốt hơn và có thể truy cập nhiều tính năng hơn của thiết bị.

Ứng dụng di động đa nền tảng: Được viết bằng một ngôn ngữ lập trình duy nhất có thể được biên dịch thành mã cho nhiều nền tảng khác nhau. Ứng dụng di động đa nền tảng dễ phát triển hơn nhưng có thể không có hiệu suất tốt hoặc không thể truy cập tất cả các tính năng của thiết bị. Các ngôn ngữ lập trình phổ biến để lập trình app Có nhiều ngôn ngữ lập trình khác nhau có thể được sử dụng để lập trình app.

 Thành phần cụ thể của một ứng dụng di động có thể khác nhau tùy thuộc vào chức năng, độ phức tạp và nền tảng phát triển. Tuy nhiên, nhìn chung, các ứng dụng di động thường bao gồm các thành phần sau:

1. Giao diện người dùng (UI): UI là phần giao diện mà người dùng tương tác với. Nó bao gồm các yếu tố như màn hình, nút, văn bản, hình ảnh và video. UI cần được thiết kế đẹp mắt, dễ sử dụng và phù hợp với đối tượng mục tiêu của ứng dụng.

2. Logic ứng dụng: Logic ứng dụng là phần xử lý dữ liệu và thực hiện các chức năng của ứng dụng. Nó được viết bằng ngôn ngữ lập trình và bao gồm các lớp, hàm và thuật toán. Logic ứng dụng cần được viết hiệu quả, bảo mật và dễ dàng bảo trì.

3. Dữ liệu: Dữ liệu là thông tin được sử dụng bởi ứng dụng. Nó có thể bao gồm dữ liệu người dùng, cài đặt ứng dụng, nội dung ứng dụng, v.v. Dữ liệu có thể được lưu trữ cục bộ trên thiết bị hoặc trên máy chủ. Dữ liệu cần được bảo mật và quản lý hiệu quả.

4. API: API là giao diện lập trình ứng dụng cho phép các phần khác nhau của ứng dụng hoặc các ứng dụng khác tương tác với nhau. API thường được sử dụng để truy cập dữ liệu, thực hiện các chức năng hoặc gửi thông báo. API cần được thiết kế rõ ràng, dễ sử dụng và bảo mật.

5. Thư viện và khuôn khổ: Thư viện và khuôn khổ là các công cụ giúp đơn giản hóa việc phát triển ứng dụng. Chúng cung cấp các chức năng chung như quản lý giao diện người dùng, kết nối mạng, xử lý dữ liệu, v.v. Việc sử dụng thư viện và khuôn khổ có thể giúp tiết kiệm thời gian và cải thiện chất lượng ứng dụng.

6. Hệ thống back-end: Hệ thống back-end là phần máy chủ cung cấp các dịch vụ cho ứng dụng di động. Nó có thể bao gồm cơ sở dữ liệu, máy chủ web, dịch vụ API và các dịch vụ khác. Hệ thống back-end cần được thiết kế có khả năng mở rộng, hiệu quả và bảo mật.

7. Phân tích ứng dụng: Phân tích ứng dụng giúp thu thập dữ liệu về cách người dùng sử dụng ứng dụng. Dữ liệu này có thể được sử dụng để cải thiện trải nghiệm người dùng, sửa lỗi và thêm các tính năng mới. Các công cụ phân tích ứng dụng phổ biến bao gồm Google Analytics và Firebase Analytics.

## Giới thiệu đề tài

Hiện nay, với sự phát triển nhanh chóng của khoa học và công nghệ, hầu hết các cửa hàng đều mong muốn đưa cửa hàng mình lên App để việc mua hàng được giải quyết một cách nhanh hơn, thuận tiện hơn, dẫn đến việc ứng dụng công nghệ thông tin trong công tác quản lý ngày càng tăng. Vì vậy việc xây dựng App không chỉ là nhu cầu thiết yếu hiện nay giúp cho việc mua hàng dễ dàng hơn và quản lý công việc kinh doanh tại các tổ chức, doanh nghiệp ngày càng được các nhà quản lý, các chủ cửa hàng.

Cùng với sự phát triển của khoa học và công nghệ, các cửa hàng ngày nay cũng dần đưa các sản phẩm của mình lên App để dễ dàng chọn lựa và mua hàng chứ không cần phải đến trực tiếp cửa hàng. Vì thế ngày càng nhiều các cửa hàng có App riêng ngày càng nhiều dẫn đến nhu cầu làm một App đẹp và gây ấn tượng cho người dùng ngày càng lớn.

Để giải quyết vấn đề trên, cần một giải pháp mang tính thực tiễn, cần xây dựng nên một “App bán hàng” cho một cửa hàng bán hàng bán sách. Việc App bán hàng không chỉ đơn thuần là một nghiệp vụ đơn lẻ mà là một tập hợp của nhiều nghiệp vụ khác nhau khác, như: Quản lý sản phẩm, quản lí người dùng,…. Người chủ cửa hàng đưa các sản phẩm đó lên App của mình và quản lí bằng App đó. Khách hàng có thể đặt mua hàng trên App mà không cần đến cửa hàng.

## Giới thiệu một số ngôn ngữ và frameword để xây dựng ứng dụng.

### Node.js

* Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.
* Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ. Dù sao thì chúng ta cũng nên biết qua một chút chút lịch sử của thứ mà chúng ta đang học một chút chứ nhỉ?
* Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.
* Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.
* Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.
* Nodejs chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.
* Có thể chạy ứng dụng Nodejs ở bất kỳ đâu trên máy Mac – Window – Linux, hơn nữa cộng đồng Nodejs rất lớn và hoàn toàn miễn phí, các package đều hoàn toàn miễn phí: https://www.npmjs.com/
* Các ứng dụng NodeJS đáp ứng tốt thời gian thực và chạy đa nền tảng, đa thiết bị.

### MongoDB

* MongoDB là gì ?
* MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.
* MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.
* Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng
* So với RDBMS thì trong MongoDB collection ứng với table, còn document sẽ ứng với row , MongoDB sẽ dùng các document thay cho row trong RDBMS.
* Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.
* Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB
* Ưu điểm của MongoDB
* Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái.
* Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.
* MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster:
* Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
* Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.
* Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.
* Nhược điểm của MongoDB
* Một ưu điểm của MongoDB cũng chính là nhược điểm của nó. MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.
* Tốn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dữ thừa dữ liệu.
* Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điêù này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện...

### React Native

* React Native là một framework mã nguồn mở được tạo ra bởi Facebook cho phép lập trình viên phát triển ứng dụng di động đa nền tảng (Android, iOS) bằng JavaScript. Nó sử dụng các thành phần giao diện người dùng (UI) gốc được viết bằng ngôn ngữ lập trình React, giúp tạo ra các ứng dụng di động hiệu quả và mượt mà.
* Ưu điểm của React Native: Học tập dễ dàng: React Native sử dụng JavaScript, một ngôn ngữ lập trình phổ biến và dễ học, giúp các lập trình viên web dễ dàng chuyển sang phát triển ứng dụng di động. Mã tái sử dụng: React Native cho phép tái sử dụng mã giữa các nền tảng Android và iOS, giúp tiết kiệm thời gian và công sức phát triển.
* Hiệu suất cao: React Native sử dụng các thành phần UI gốc, mang lại hiệu suất cao và trải nghiệm người dùng mượt mà.
* Cộng đồng lớn: React Native có một cộng đồng lập trình viên lớn và phát triển nhanh chóng, cung cấp nhiều tài nguyên và hỗ trợ cho các nhà phát triển.
* Nhược điểm của React Native: Khó khăn trong việc tối ưu hóa: Việc tối ưu hóa hiệu suất ứng dụng React Native có thể khó khăn hơn so với các ứng dụng di động gốc.
* Ứng dụng React Native thường có kích thước lớn hơn so với ứng dụng di động gốc.
* Một số tính năng API giới hạn: React Native không hỗ trợ tất cả các tính năng API của Android và iOS.Tương thích với trình duyệt: Nó tương thích với tất cả các trình duyệt (Chrome, Firefox, Internet Explorer, Safari, and Opera).

## Lý do chọn để tài

Bằng việc lựa chọn và thực hiện đề tài “Xây dựng App bán sách điện tử ”, em muốn tìm hiểu và đưa ra một giải pháp tốt nhằm giải quyết công việc bán hàng trực tuyến cho công ty. App mang lại cho công ty rất nhiều lợi ích như: khả năng quảng cáo, phổ biến tên tuổi công ty rất nhanh trên môi trường mạng toàn cầu, việc kinh doanh sẽ mở 24/24, giảm thiểu chi phí nhân viên cho tiếp thị, quảng cáo, dễ dàng nhận phải hồi từ phía khách hàng, cơ hôi mở rộng liên kết và hợp tác ở phạm vi quốc tế...v..v.

App bán hàng giúp cho công ty cung cấp cho khách hàng những lựa chọn linh hoạt và tiện lợi trong việc tìm mua sản phẩm thông qua chức năng tìm kiếm và giỏ hàng. Các thông tin về sản phẩm được hiển thị chi tiết với từng sản phẩm, từ đó khách hàng dễ dàng nhận biết và lựa chọn được thứ mình cần. Về phía Công ty, hệ thống mang lại sự tiện dụng trong việc cập nhập và quản lý các thông tin cho website, tạo khả năng liên lạc nhanh và dễ dang với khách hàng.

## Giới thiệu về project

* Project xây dựng App bán sách ReaBok là app được xây dựng trên ngôn ngữ chính đó là javascript bên cạnh đó sử dụng các thư viện: nodejs, reactNative và cơ sở dữ liệu Mongodb để kết nối.

## Tính cấp thiết của đề tài:

* Cùng với sự phát triển mạnh mẽ của Internet, ngày nay, việc sở hữu một App không còn là điều xa lạ, thậm chí trong một số trường hợp còn là tiêu chuẩn bắt buộc đối với doanh nghiệp – công ty trong thời điểm cạnh

tranh mang tính toàn cầu như hiện nay.

* Cùng với đó các công ty thương mại kinh doanh các mặt hàng điện tử đang rất cần một App để có thể giới thiệu sản phẩm, thực hiện bán hàng trực tuyến.

### Mục tiêu:

* Xây dựng các chức năng cơ bản của một App bán hàng sách. App có khả năng tự tương thích, hiển thị được trên tất cả các thiết bị hiện tại và có thể nâng cấp trong tương lai.
* App hiển thị sản phẩm đẹp, thu hút người dùng.
* Hỗ trợ khách hàng một cách nhanh nhất khi nhận được yêu cầu liên hệ.
* Quản lý các mặt hàng sản phẩm, cấu hình app.
* Quản lý thống kê doanh thu, các đơn hàng sản phẩm, số lượng sản phẩm tồn đọng.
* Nắm bắt được công nghệ thiết kế App resful Node.js, MongoDB, React.
* Ứng dụng mô hình MVC vào xây dựng trang web.

### Nội dung chính:

* Tìm hiều, ứng dụng các ngôn ngữ lập trình, platform, hệ quản trị cơ sở dữ liệu vào thiết kế website (Node.js ,MongoDB).
* Thiết kế đặc tả hệ thống.
* Xây dưng cơ sở dữ liệu trên MongoDB.
* Xây dựng các module quản lý.
* Tìm hiểu về Node.js, MongoDB phục vụ trong quá trình xây dựng trang App.
* App nhanh và hiệu quả hơn trong việc quản lý, bán hàng.

### Kết quả chính đạt được

* Xây dựng thành công App bán sách điện tử.
* Tìm hiểu nắm bắt được các kiến thức về Node.js ,MongoDB,…
* Ứng dụng , phát triển được công nghệ thiết kế App bằng Node.js.
* Với quy mô trang app vừa và nhỏ, cộng thêm thời gian không cho phép nên đề tài “Xây dựng website bán hàng sách Reabok” có thể chưa có đầy đủ các chức năng hoàn thiện của một trang app bán hàng. Trong quá trình làm còn nhều sai sót, do kiến thức có hạn lên app còn ít nhiều thiếu sót về việc hiển thị cũng như tối ưu thuật toán, thời gian thực hiện. Trong tương lai nhóm chúng em sẽ cố gắng khắc phục những sai sót này.

# Chương II: Phân tích thiết kế hệ thống App

## Đặt tả yêu cầu về nghiệp vụ:

### Yêu cầu chức năng:

1. Yêu cầu lưu trữ:

Tính năng lưu trữ là yêu cầu quan trọng nhất của hệ thống, là cơ sở để thực hiện phần lớn các tính năng của hệ thống, để đảm bảo việc quản lý việc bán hàng có hiệu quả, hệ thống cần lưu trữ những thông tin sau:

* Thông tin về sản phẩm: tên sản phẩm, loại sản phẩm, giá, mô tả.
* Thông tin về khách hang: tên khách hàng, tuổi, địa chỉ, các hóa đơn mua hang của khách hang đó.
* Thông tin về hóa đơn: khách hang, ngày lập hóa đơn, các sản phẩm trong hóa đơn, thành tiền.
* Thông tin về nhân viên: tên nhân viên, tuổi, địa chỉ.

1. Yêu cầu về tính năng

Ngoài chức năng lưu trữ, hệ thống còn phải đáp ứng được những yêu cầu sau:

* Chức năng truy vấn: hệ thống phải luôn đáp ứng các yêu cầu truy vấn từ người dung như: tìm kiếm sản phẩm, tìm kiếm loại sản phẩm,..
* Chức năng thống kê, báo cáo, giúp nhân viên quản lý theo dõi tình hình của shop.
* Chức năng cập nhật: hệt thống phải đảm bảo chức năng cập nhật được thuận tiện và tránh tối đa lỗi do sơ xuất của người sử dụng.
* Các chức năng tự động: giúp người dung thuận tiện hơn trong thao tác như tự động tính tổng tiền hóa đơn, tự động tính doanh thu theo từng tháng, năm,…

### Yêu cầu phi chức năng

* Yêu cầu về giao diện: Giao diện quản lý gần gũi, đơn giản. Giao diện dùng cho khách hang than thiện, sinh động, tương tác cao.
* Yêu cầu chất lượng:
* Tính tiến hóa: Dễ dàng trong lúc nâng cấp hoặc tăng các module tiện ích.
* Tính tiện dụng: Hệ thống có giao diện trực quan, thân thiện và dễ sử dụng. Các đặc tả và hướng dẫn sử dụng rõ ràng.
* Tính hiệu quả: App thể hiện tốt các thông tin được yêu cầu. App hoạt động ổn và đáng tin cậy.
* Tính tương thích: Hoạt động tốt với tất cả các trình duyệt App phổ biến hiện nay.
* Yêu cầu an toàn:
* Thông báo và chờ xác nhận của người dùng trước khi xóa một dòng dữ liệu.
* Cho phép hủy thật sự những dòng dữ liệu bị lỗi theo ý của người dùng.
* Yêu cầu về bảo mật:
* Ghi nhận và thực hiện đúng theo quyền hạn đã định trước.
* Tất cả các thông tin lưu trữ phải được bảo mật, chỉ những người có quyền hạn mới được xem, tác động.
* Yêu cầu về công nghệ:
* Dễ sửa đổi: xác định lỗi trung bình trong 13 phút.
* Dễ bảo trì: cho phép thêm chức năng mới, nhanh.
* Tái sử dụng: các module hiện tại có thể sử dụng lại cho các yêu cầu sau này.
* Dễ mang chuyển: đổi sang hệ quản trị cơ sở dữ liệu mới tối đa 3 ngày.
* Khi sửa một lỗi chức năng không ảnh hưởng đến chức năng khác.

## Sơ đồ use case :

### Use case tổng quát: A diagram of a diagram Description automatically generated with medium confidence

Danh sách tác nhân của hệ thống:

|  |  |  |
| --- | --- | --- |
| STT | Tác nhân | Ý nghĩa |
| 1 | Người quản trị (Admin) | Người quản trị hệ thống. |
| 2 | Người dùng | Khách hàng có nhu cầu mua hàng truy cập vào hệ thống. |

|  |  |  |
| --- | --- | --- |
| STT | Use case | Ý nghĩa |
| 1 | Đăng nhập | Cho phép người truy cập đăng nhập vào hệ thống. Dựa vào vai trò của user đăng nhập mà có thể giới hạn các thao tác với hệ thống. |
| 2 | Đăng ký | Cho phép người truy cập tạo tài khoản và có thể đăng nhập vào hệ thống với thông tin đăng nhập lúc đăng ký. |
| 3 | Quản lý sản phẩm | Thực hiện thêm/xoá/cập nhật/xem các sản phẩm của hệ thống. |
| 4 | Quản lý đơn đặt hàng | Cho phép người dùng xem các đơn đặt hàng sau khi đăng nhập vơi vai trò admin. |
| 5 | Xem danh sách các sản phẩm bày bán | Cho phép người dùng xem danh sách các sản phẩm phù hợp theo loại. |
| 6 | Quản lý giỏ hàng | Thực hiện thêm/xoá/cập nhật/xem các sản phẩm trong giỏ hàng của hệ thống. |

### Đặc tả use case:

#### a. Use case đăng nhập:

**sự kiện kích hoạt**: Usecase bắt đầu khi có người dùng muốn đăng nhập vào hệ thống.

**Dòng sự kiện chính:**

1. Hệ thống hiển thị trang đăng nhập.
2. Người dùng nhập thông tin tài khoản và mật khẩu để đăng nhập vào hệ thống.
3. Hệ thống xác thực thông tin đăng nhập.
4. Trường hợp xác thực thành công, hệ thống cho phép người dùng thao tác dựa trên quyền của tài khoản đăng nhập. Ngược lại, từ chối đăng nhâp.

**Dòng sự kiện phụ**:

1. Nhập thiếu thông tin: Hệ thống kiểm tra các trường bắt buộc và yêu cầu người dùng nhập lại đầy đủ thông tin đăng nhập
2. Tài khoản đăng nhập không đúng: Hệ thống thông báo tài khoản đăng nhập không đúng và yêu cầu người dùng nhập lại.
3. Đăng nhập thất bại: Hệ thống trả về lỗi đăng nhập

**Tiền điều kiện**: Người dùng đã có tài khoản đăng nhập vào hệ thống

**Hậu điều kiện**: Hệ thống chuyển sang trang giao diện chính của chương trình

#### b. Use case đăng ký:

**sự kiện kích hoạt**: Use case bắt đầu khi có người dùng muốn đăng ký tài khoản.

**Dòng sự kiện chính:**

1. Hệ thống hiển thị trang đăng ký.
2. Người dùng nhập thông tin cá nhân tài khoản và mật khẩu.
3. Hệ thống xác thực thông tin đăng ký.
4. Hệ thống lưu trữ thông tin của tài khoản đăng kí xuống CSDL.
5. Hệ thống xuất kết quả lưu trữ, cập nhật ra màn hình.
6. Trường hợp xác thực thành công, hệ thống thông báo đăng ký thành công. Ngược lại từ chối đăng ký.

**Dòng sự kiện phụ**:

1. Nhập thiếu thông tin: Hệ thống kiểm tra các trường bắt buộc và yêu cầu người dùng nhập lại đầy đủ thông tin để có thể đăng ký.
2. Tài khoản đăng ký bị trùng, hoặc không đúng: Hệ thống thông báo tài khoản đăng ký không đúng và yêu cầu người dùng nhập lại.
3. Đăng ký thất bại: Hệ thống trả về lỗi đăng ký.

**Tiền điều kiện**: Người dùng đã có tài khoản mail.

**Hậu điều kiện**: Hệ thống chuyển sang trang giao diện diện đăng nhập.

#### c. Use case Quản lý sản phẩm:

**sự kiện kích hoạt**: Usecase bắt đầu khi người dùng chọn thực hiện quản lý sản phẩm từ trang giao diện chính. Hệ thống sẽ xử lý yêu cầu của người dùng.

**Dòng sự kiện chính:**

1. Hệ thống lấy dữ liệu danh sách các sản phẩm từ CSDL và hiển thị ra màn hình
2. Người dùng chọn chức năng cần thực hiện (Tạo mới hoặc Cập nhật sản phẩm)
3. Hệ thống hiển thị giao diện Tạo mới hoặc Cập nhật tuỳ theo lựa chọn ở bước 2
4. Hệ thống lưu trữ/ cập nhật xuống CSDL
5. Hệ thống xuất kết quả lưu trữ, cập nhật ra màn hình

**Dòng sự kiện phụ:**

1. Người dùng nhập thiếu thông tin: Hệ thống kiểm tra các trường thông tin bắt buộc và yêu cầu người dùng nhập lại đầy đủ.
2. Tạo mới sản phẩm thất bại: Hệ thống trả về lỗi.
3. Cập nhật sản phẩm thất bại: Hệ thống trả về lỗi.

**Tiền điều kiện:** Người dùng đã đăng nhập thành công vào hệ thống với vai trò là admin.

#### f. Use case Quản lý đơn đặt hàng:

**sự kiện kích hoạt**: Use case bắt đầu khi người dùng chọn thực hiện quản lý đơn đặt hàng từ trang giao diện quản lý của admin. Hệ thống sẽ xử lý yêu cầu của người dùng.

**Dòng sự kiện chính:**

1. Hệ thống lấy dữ liệu danh sách các thông tin của đơn đặt hàng từ CSDL và hiển thị ra màn hình
2. Người dùng chọn chức năng cần thực hiện (xem đơn đặt hàng).
3. Người dùng chọn thời gian phù hợp để xem những đơn hàng đã được đặt trong thời gian đó.
4. Hệ thống hiển thị giao diện danh sách đơn đặt hàng.

**Dòng sự kiện phụ:**

1. Lấy dữ liệu thất bại hệ thống trả về lỗi.

**Tiền điều kiện:** Người dùng đã đăng nhập thành công vào hệ thống với vai trò là admin.

**Hậu điều kiện:** Hệ thống lưu trữ load danh sách đặt hàng trong thời gian cụ thể mà người dùng muốn xem.

#### g. Use case Xem danh sách sản phẩm bày bán:

**sự kiện kích hoạt**: Use case thực khi người dùng truy cập vào App.

1. Dòng sự kiện chính: Hệ thống lấy dữ liệu danh sách thông tin của sản phẩm từ CSDL. Và hiển thị ra màn hình.
2. Người dùng tiến hành chọn loại sản phẩm mà mình muốn xem. Hể thống sẽ điều hướng đến trang load danh sách các sản phẩm thuộc loại đó/
3. Người dùng click vào sản phẩm và tiến hành xem sản phẩm. Nếu click vào xem chi tiết sản phẩm trang web sẽ điều hướng đến trang xem chi tiết sản phẩm.

**Dòng sự kiện phụ:**

1. Lấy dữ liệu thất bại hệ thống trả về lỗi.

**Tiền điều kiện:** Người dùng đã truy cập vào App.

**Hậu điều kiện:** Hệ thống hiển thị hiển thị danh sách các sản phẩm được bày bán theo thể loại.

#### i. Use case đặt hàng:

**sự kiện kích hoạt**: Use case thực hiện người dùng tiến hành lựa chọn sản phẩm đặt hàng.

**Dòng sự kiện chính:**

1. Hệ thống lấy dữ liệu danh sách các sản phẩm
2. Người dùng tiến hành xem sản phẩm và tiến hành thêm sản phẩm vào giỏ hàng.
3. Sau đó nhập thông tin cần thiết để tiến hành thanh toán.
4. Hệ thống kiểm tra các thông tin cần thiết cập nhật lưu trữ thông tin đặt hàng của khách hàng xuống CSDL.
5. Hệ thống xuất kết quản ra màn hình.

**Dòng sự kiện phụ:**

Lấy dữ liệu thất bại hệ thống trả về lỗi.

1. Người dùng nhập thiếu thông tin: Hệ thống kiểm tra các trường thông tin bắt buộc và yêu cầu người dùng nhập lại đầy đủ.
2. Lưu dữ liệu thất bại: Hệ thống trả về lỗi
3. Cập nhật dữ liệu thất bại: Hệ thống trả về lỗi

**Tiền điều kiện:** Người dùng đã đăng nhập thành công vào hệ thống với vai trò là admin.

**Hậu điều kiện:** Hệ thống lưu trữ và cập nhật thông tin xuống cơ sở dữ liệu.

#### j. Use case Quản lý giỏ hàng:

**sự kiện kích hoạt**: Usecase bắt đầu khi người dùng click vào giỏ hàng . Chức năng này cho phép người dùng cập nhật giỏ hàng và xem lại các thông tin của sản phẩm đã đặt mua.

**Dòng sự kiện chính:**

1. Hệ thống lấy dữ liệu về danh sách các sản phẩm có trong giỏ hàng.
2. Người dùng sau khi chỉnh sửa số lượng sản phẩm trong giỏ hàng sẽ chọn update giỏ hàng
3. Hệ thống lưu trữ và cập nhật xuống cơ sở dữ liệu
4. Hệ thống xuất kết quả thực hiện ra màn hình

**Dòng sự kiện phụ:**

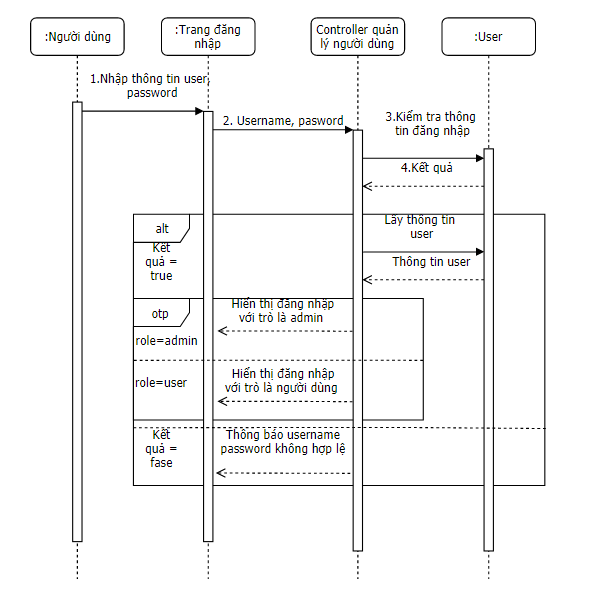
1. Lấy dữ liệu thất bại: Hệ thống trả về lỗi
2. Lưu dữ liệu thất bại: Hệ thống trả về lỗi
3. Cập nhật dữ liệu thất bại: Hệ thống trả về lỗi

**Tiền điều kiện:** Người dùng đã đăng nhập thành công vào hệ thống

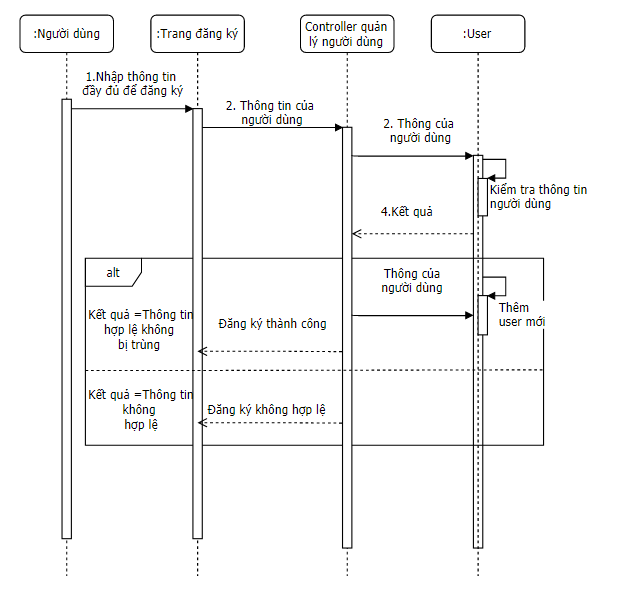
**Hậu điều kiện:** Hệ thống lưu trữ, cập nhật thông tin xuống cơ sở dữ liệu

## Sơ đồ Sequence

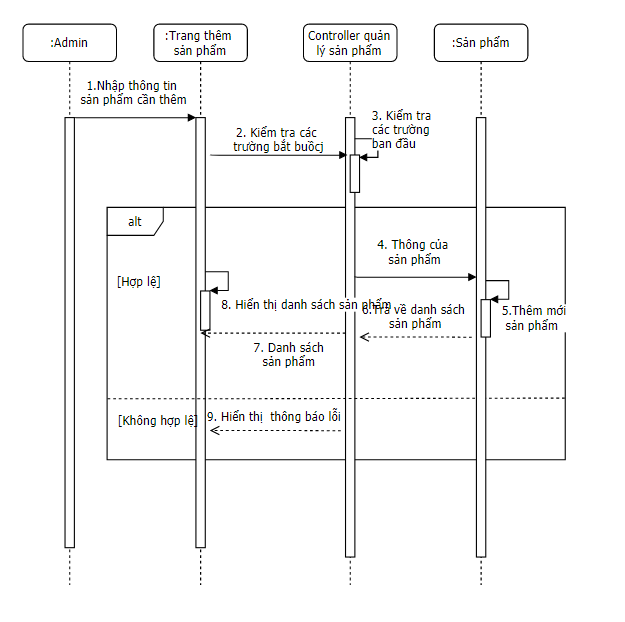
### Đăng nhập



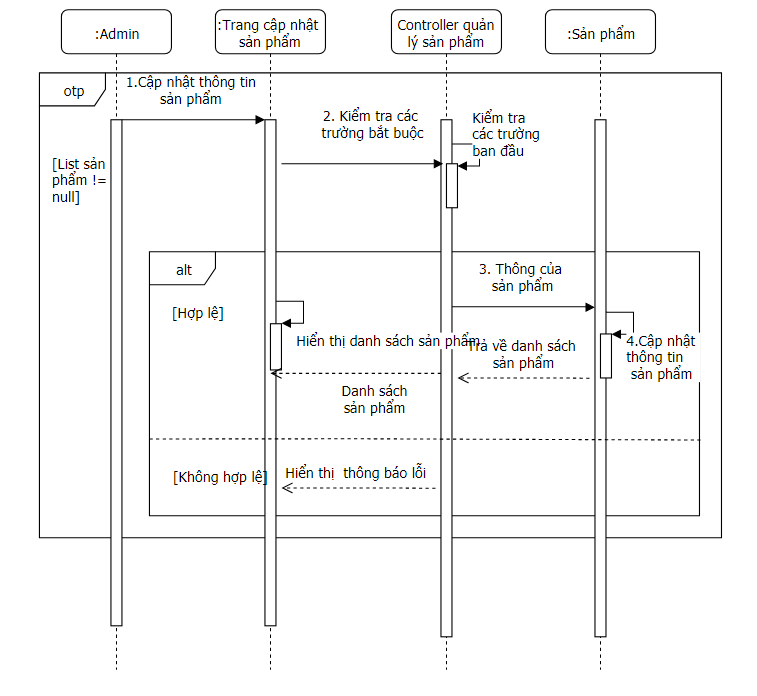
### Đăng ký



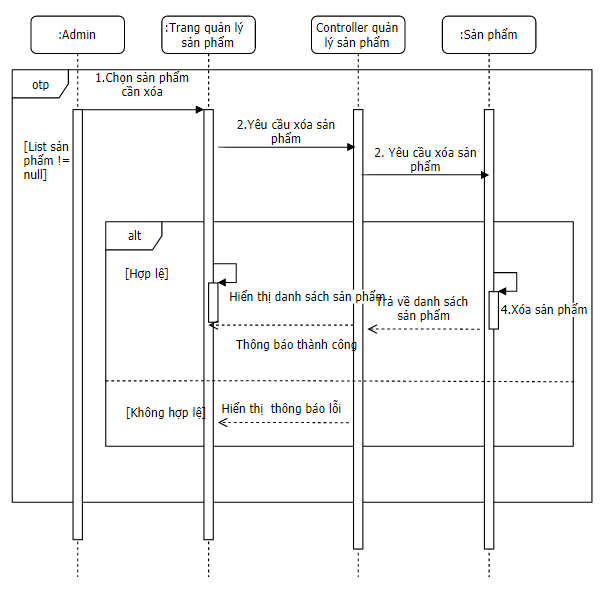
### Thêm sản phẩm:



### Cập nhật sản phẩm:

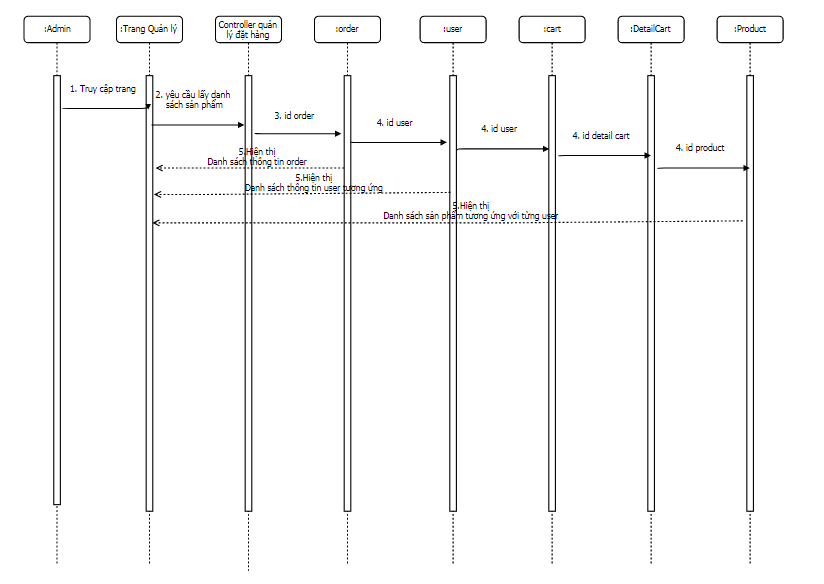


### Xóa sản phẩm:

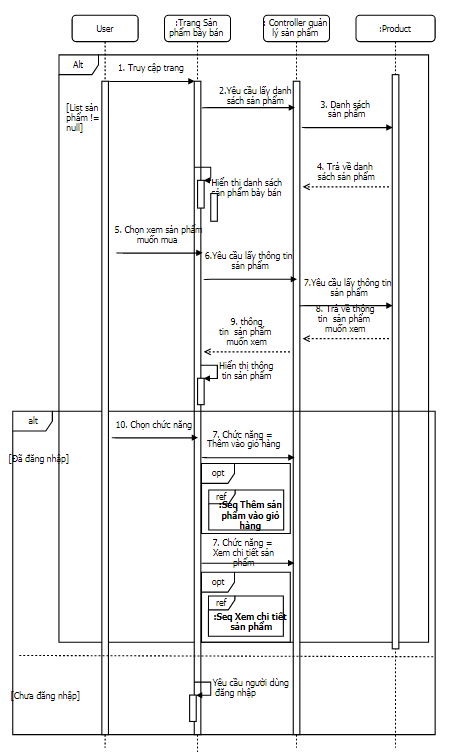


### Quản lý đơn đặt hàng

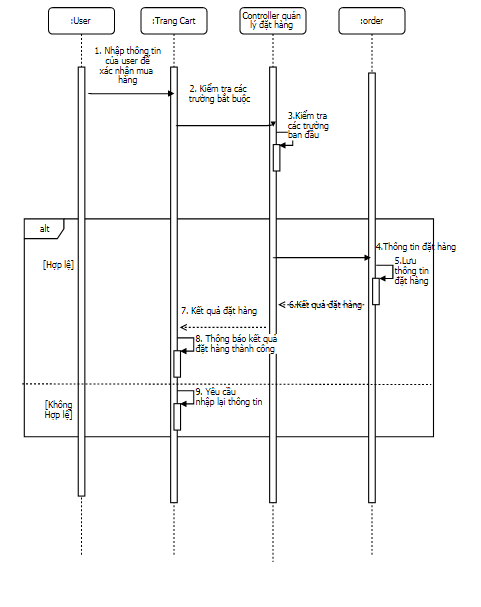
Xem danh sách đơn đặt hàng:



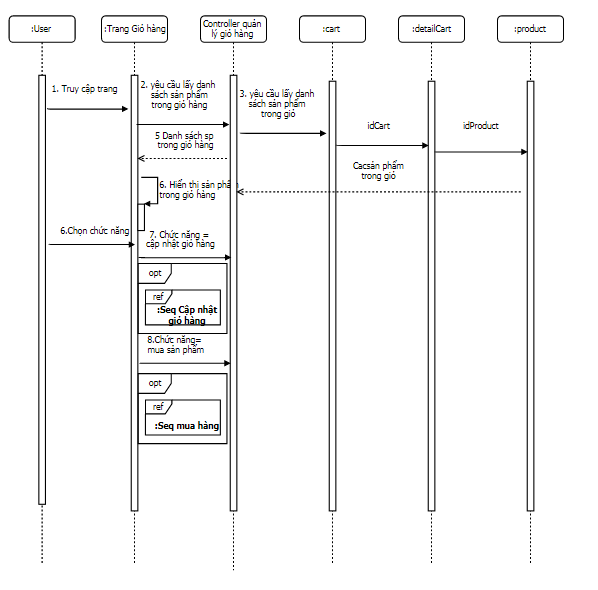
### Xem danh sách các sản phẩm bày bán



### Đặt hàng



### Quản lý giỏ hàng:



# Chương III: Thiết kế và xây dựng ứng dụng:

## Thiết kế cơ sở dữ liệu:

A screenshot of a computer

Description automatically generated

## Thiết kế giao diện

### Màn hình Trang chủ:

A screenshot of a cell phone

Description automatically generated

### Màn hình đăng nhập

A screenshot of a login screen

Description automatically generated

### Màn hình đăng kí

A screenshot of a phone login

Description automatically generated

### Màn hình xem chi tiết sản phẩm:

A cell phone with a picture of a person jumping on the edge of a screen

Description automatically generated

### Màn hình chi tiết giỏ hàng và tiết hành checkout:

A screenshot of a cell phone

Description automatically generated

### Màn hình thêm sản phẩm mới

A screenshot of a phone

Description automatically generated

### Màn hình quản lý sản phẩm

A screenshot of a phone

Description automatically generated

### Màn hình cập nhật sản phẩm

A screenshot of a phone

Description automatically generated

# Chương IV: Tổng kết:

## Kết quả đạt được:

* **Xây dựng ứng dụng bán hàng** không phải là đề tài mới, tuy nhiên vẫn còn một vài cơ sở quản lý bằng phương pháp thủ công. Nhận thấy đây là một đề tài hay, có thiết thực và có thể áp dụng trong thực tế, chính vì vậy nhóm chúng em đã xây ứng dụng bán điện thoại để giúp ích cho việc quản lý đơn giản, tiện ích hơn công.

Dưới đây là một số chức năng chính mà nhóm đã hoàn thành:

Thực hiện lập hoá đơn thanh toán khi đặt hàng

* Quản lý đặt hàng
* Quản lý sản phẩm
* Đặt hàng
* Quản lý giỏ hàng
* Xem sản phẩm và chi tiết sản phẩm.

## Hướng phát triển của ứng dụng:

* Hướng phát triển của sản phẩm trong tương lai sẽ cập thêm chức năng chia sẻ, và thực hiện thanh toán qua ví điện tử. Để người sử dụng có thể mua hàng nhanh chóng và tiện dụng hơn. Tra cứu thông tin sản phẩm. Có thể xuất báo cáo thống kê file excel. Cập nhật tình trạng mua hàng từ khách hàng sau khi đặt hàng. Đồng thời cải thiện về mặt thiết kế để cho ứng dụng trở nên đẹp mắt và thân thiện với người dùng hơn.

## Hạn chế:

* Sản phẩm vẫn còn hạn chế về mặt thiết kế cũng như những tính năng mới. Thời gian load các tính năng chưa được nhanh và mượt lắm.

## Phân công công việc:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thành viên | Công việc | Đánh giá hoàn thành |
| 1 | Nguyễn Viết Long Sơn  2104110006 | Xây dựng server với các chức năng:   * Login * logout * Quản lý sản phẩm   +Viết báo cáo | 90% |
| 2 | Nguyễn Trần Anh Khoa  2104110042 | Xây dựng server với các chức năng:   * Đặt hàng * Quản lý đặt hàng * Quản lý giỏ hàng * Xem sản phẩm và chi tiết sản phẩm.   + Viết báo cáo | 90% |
| 3 | Ngô Trung Nghĩa  2104110019 | Thiết kế ứng dụng (front-end)  + Viết báo cáo | 90% |

## Tài liệu tham khảo:

[1]: <https://scotch.io/tutorials/build-and-understand-a-simple-nodejs-website-with-user-authentication>

[2]: <https://hackernoon.com/how-to-build-a-website-for-your-band-using-node-js-a7c856266c>

[3]: <https://www.tutorialspoint.com/mongodb/index.htm>