**TRƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ KINH TẾ SỐ**



**CHUYÊN ĐỀ THỰC TẬP**

Đề tài: “*Xây dựng ứng dụng bán cây cảnh trên nền tảng*

*hệ điều hành Android*”

Giảng viên hướng dẫn : ThS. Cao Thị Thu Hương

Họ và tên sinh viên : Nguyễn Công Thành

Mã sinh viên : 11194711

Lớp : Khoa học máy tính 61

Chuyên ngành : Khoa học máy tính

Hà Nội, tháng 4 năm 2023

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn sâu sắc tới ThS. Cao Thị Thu Hương, Viện Công nghệ thông tin và Kinh tế số, Trường Đại học Kinh tế Quốc dân – người đã dành nhiều thời gian tận tình chỉ bảo, hướng dẫn và giúp đỡ em trong suốt quá trình tìm hiểu nghiên cứu. Cô là người định hướng và đưa ra nhiều góp ý quý báu trong quá trình em thực hiện báo cáo này.

Em xin gửi lời cảm ơn chân thành tới các thầy, cô ở Viện Công nghệ thông tin và Kinh tế số - Trường Đại học Kinh tế Quốc dânđã trang bị những kiến thức và tạo điều kiện thuận lợi cho em trong suốt quá trình học tập tại trường.

Em cũng xin bày tỏ lòng biết ơn về sự giúp đỡ của các anh, chị, em, bạn bè đã cung cấp dữ liệu, tài liệu và cho em những lời khuyên quý báu và luôn tạo mọi điều kiện tốt nhất cho em.

Em xin chân thành cảm ơn!

# TÓM TẮT NỘI DUNG BÁO CÁO

Nội dung báo cáo gồm ba chương với nội dung lần lượt là:

**Chương I: Tổng quan về hệ thống**. Chương này sẽ trình bày hai phần chính. Phần một chịu trách nhiệm giới thiệu về đề tài. Phần này sẽ làm rõ các nội dung như lý do chọn đề tài, mục tiêu của đề tài hay phương pháp nghiên cứu để xây dựng đề tài…Nội dung thứ hai của chương này sẽ tóm tắt ngắn gọn khái niệm những công nghệ, cách thức sử dụng để xây dựng hệ thống: Java, Android và Firebase.

**Chương II: Phân tích và thiết kế hệ thống.** Chương này sẽ tóm tắt quá trình lên ý tưởng xây dựng hệ thống. Từ khâu phân tích, xác định các yêu cầu hệ thống (yêu cầu chức năng, yêu cầu phi chức năng) đến việc mô hình hóa các yêu cầu của hệ thống bằng các sơ đồ UML (use case, sequence diagram…), thiết kế logic và cuối cùng là thiết kế giao diện cho hệ thống.

**Chương III: Triển khai và kiểm thử hệ thống:** Chương III sẽ trình bày đến người đọc quá trình đưa những ý tưởng thiết kế từ chương II và áp dụng vào thực tế để xây dựng lên hệ thống bằng cách sử dụng những công cụ, tool hỗ trợ như git để làm việc nhóm, Jenkins để deploy hệ thống một cách nhanh chóng và hiệu quả hơn.

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc132542560)

[TÓM TẮT NỘI DUNG BÁO CÁO 2](#_Toc132542561)

[DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT 5](#_Toc132542562)

[DANH MỤC HÌNH VẼ 6](#_Toc132542563)

[DANH MỤC BẢNG BIỂU 7](#_Toc132542564)

[LỜI MỞ ĐẦU 8](#_Toc132542565)

[CHƯƠNG I: TỔNG QUAN VỀ HỆ THỐNG 9](#_Toc132542566)

[1.1 Giới thiệu chung 9](#_Toc132542567)

[1.1.1 Lý do chọn đề tài 9](#_Toc132542568)

[1.1.2 Mục tiêu đề tài 10](#_Toc132542569)

[1.1.3 Đối tượng và phạm vi nghiêm cứu 10](#_Toc132542570)

[1.1.4 Phương pháp nghiên cứu 10](#_Toc132542571)

[1.1.5 Ý nghĩa thực tiễn và khoa học của đề tài 11](#_Toc132542572)

[1.2 Công nghệ sử dụng 13](#_Toc132542573)

[1.2.1 Java 13](#_Toc132542574)

[1.2.2 Android 14](#_Toc132542575)

[1.2.3 Firebase 18](#_Toc132542576)

[CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 20](#_Toc132542577)

[2.1 Xác định yêu cầu bài toán 20](#_Toc132542578)

[2.1.1 Yêu cầu chức năng 20](#_Toc132542579)

[2.1.2 Yêu cầu phi chức năng 21](#_Toc132542580)

[2.2 Phân tích hệ thống 23](#_Toc132542581)

[2.2.1 Sơ đồ use case (Use case diagrams) 23](#_Toc132542582)

[2.2.2 Biểu đồ tuần tự (Sequence diagrams) 28](#_Toc132542583)

[2.2.3 Sơ quan hệ thực thể (ER diagram) 31](#_Toc132542584)

[2.2.4 Biểu đồ trạng thái (Stage diagrams) 33](#_Toc132542585)

[2.3 Thiết kế hệ thống 34](#_Toc132542586)

[2.3.1 Thiết kế logic hệ thống 34](#_Toc132542587)

[2.3.2 Thiết kế giao diện 39](#_Toc132542588)

[CHƯƠNG III: TRIỂN KHAI VÀ KIỂM THỬ HỆ THỐNG 48](#_Toc132542589)

[3.1 Triển khai xây dựng hệ thống 48](#_Toc132542590)

[3.2 Kiểm thử hệ thống 49](#_Toc132542591)

[KẾT LUẬN 55](#_Toc132542592)

[TÀI LIỆU THAM KHẢO 57](#_Toc132542593)

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Ký hiệu viết tắt | Chữ viết đầy đủ |
| 1 | TMĐT | Thương mại điện tử |
| 2 | JDK | Java development kit |
| 3 | JVM | Java Virtual Machine |
| 4 | SDK | Software development kit |
| 5 | MVP | Model – View – Presenter |
| 6 | MVC | Model – View – Controller |
| 7 | MVVM | Model – View - ViewModel |

# DANH MỤC HÌNH VẼ

[Hình 1: Vai trò của thương mại điện tử 12](#_Toc132541134)

[Hình 2: Yêu cầu phi chức năng 22](#_Toc132541135)

[Hình 3: Các tác nhân trong hệ thống 23](#_Toc132541136)

[Hình 4: Sơ đồ use case tổng quát của hệ thống 24](#_Toc132541137)

[Hình 5: Use case hiện thị sản phẩm 25](#_Toc132541138)

[Hình 6: Sơ đồ use case mua sản phẩm 26](#_Toc132541139)

[Hình 7: Use quản lý tài khoản cá nhân 27](#_Toc132541140)

[Hình 8: Quá trình người dùng duyệt sản phẩm 29](#_Toc132541141)

[Hình 9: Quá trình người dùng mua hàng và thanh toán 30](#_Toc132541142)

[Hình 10: Quy trình kiểm tra trạng thái đơn hàng 31](#_Toc132541143)

[Hình 11: Sơ đồ ER 32](#_Toc132541144)

[Hình 12: State diagram quá trình chuyển từ giỏ hàng thành đơn đặt hàng 33](#_Toc132541145)

[Hình 13: State diagram quản lý giỏ hàng 34](#_Toc132541146)

[Hình 14: Mô hình kiến trúc hệ thống 35](#_Toc132541147)

[Hình 15: Mô hình MVVM 36](#_Toc132541148)

[Hình 16: So sánh cách giao tiếp giữa client và server 38](#_Toc132541149)

[Hình 17: Màn hình đăng ký và đăng nhập 39](#_Toc132541150)

[Hình 18: Màn hình trang chủ 40](#_Toc132541151)

[Hình 19: Màn hình chi tiết sản phẩm 41](#_Toc132541152)

[Hình 20: Màn hình yêu thích và giỏ hàng 42](#_Toc132541153)

[Hình 21: Màn hình thanh toán 43](#_Toc132541154)

[Hình 22: Màn hình thông báo 44](#_Toc132541155)

[Hình 23: Màn hình cài đặt 45](#_Toc132541156)

[Hình 24: Màn hình trạng thái đơn hàng và cài đặt địa chỉ giao hàng 46](#_Toc132541157)

[Hình 25: Màn hình phương thức thanh toán và reviews 47](#_Toc132541158)

[Hình 26: Quy trình triển khai hệ thống 49](#_Toc132541159)

# DANH MỤC BẢNG BIỂU

[Bảng 1: Yêu cầu chức năng của hệ thống 44](#_Toc132537812)

[Bảng 2: Mô tả test case 44](#_Toc132537813)

[Bảng 3: Kết quả kiểm thử 48](#_Toc132537814)

# LỜI MỞ ĐẦU

Với sự phát triển nhanh chóng của công nghệ thông tin và Internet, thương mại điện tử đã được ứng dụng và phát triển mạnh mẽ trong những năm gần đây. Như chúng ta đã biết, mô hình ứng dụng và các công nghệ liên quan tới thương mại điện tử ngày càng hoàn thiện, có thể nói kỷ nguyên Thương mại điện tử đã và đang phát triển với tốc độ chóng mặt. Thương mại điện tử đã trở nên cần thiết cho các doanh nghiệp, đặc biệt là doanh nghiệp vừa và nhỏ. Các doanh nghiệp này đang tìm cách tối đa hóa lợi nhuận bằng cách sử dụng các phương pháp tiếp thị trực tuyến nhằm gia tăng hiệu quả bán hàng và tối thiểu hóa chi phí mà mình phải bỏ ra. Báo cáo này bao gồm các giới thiệu, xem xét phân tích, thiết kế và triển khai hệ thống thương mại điện tử. Vai trò của Thương mại điện tử trong doanh nghiệp, các vấn đề pháp lý và bảo mật tham gia phát triển hệ thống Thương mại điện tử, kiến trúc hệ thống Thương mại điện tử, thiết kế và chiến lược phát triển được đánh giá nghiêm túc và các phương pháp phù hợp nhất ứng dụng trong phát triển hệ thống Thương mại điện tử **SP.IO.**

# CHƯƠNG I: TỔNG QUAN VỀ HỆ THỐNG

## 1.1 Giới thiệu chung

### 1.1.1 Lý do chọn đề tài

Trong những năm trở lại đây, các doanh nghiệp trên thế giới nói chung và tại Việt Nam nói riêng đã trải qua những thay đổi đáng kể trong hệ thống thông tin kinh doanh của họ. Các khoản đầu tư lớn đã được thực hiện trong việc triển khai hệ thống hoạch định nguồn lực doanh nghiệp nhưng họ vẫn gặp khó khăn trong việc tiếp cận những thông tin kịp thời, cần thiết để đưa ra quyết định kinh doanh hiệu quả và đảm bảo sự phát triển liên tục cho doanh nghiệp. Đặt chữ "e" (electronic) trước bất kỳ quy trình, chức năng hoặc hệ thống nào dường như là liều thuốc thần kỳ cho câu chuyện thành công và lợi nhuận nhanh chóng của các doanh nghiệp: e-business (kinh doanh điện tử), e-procurement (mua sắm điện tử), e-sales (bán hàng điện tử), e-payment (thanh toán điện tử), e-banking (ngân hàng điện tử), e-CRM (CRM điện tử), e-CAD (CAD điện tử), e-delivery(giao hàng điện tử) chỉ là mội vài ví dụ nhỏ cho sự phát triển mạnh mẽ của electronic. Internet đã và đang trở thành một trong những phương tiện phổ biến nhất để gửi đi các dữ liệu khác nhau. Người dùng có thể tìm thấy bất kỳ loại thông tin nào mà họ mong muốn một cách hiệu quả và tốn ít thời gian hơn so với các phương pháp thông thường vốn tiêu tốn nhiều thời gian và công sức.

Sự xuất hiện của Internet trên toàn thế giới đã và đang đóng góp một phương tiện vô cùng hữu ích và đa dạng trong hoạt động kinh doanh cũng như lối sống của con người. Trên thực tế, Internet là điều kiện tiên quyết cần thiết cho sự tồn tại của Thương mại điện tử. Thương mại điện tử được định nghĩa là khả năng thực hiện các giao dịch liên quan đến trao đổi hàng hóa hoặc dịch vụ giữa hai hoặc nhiều bên sử dụng các công cụ và kỹ thuật điện tử (Yonah, 1997)[[1]](#footnote-2). Sự bùng nổ của Thương mại điện tử đã tạo ra những thay đổi mới, tích cực trong lối sống của chúng ta đặc biệt là trong hoạt động mua sắm. Người tiêu dùng có thể dễ dàng mua các sản phẩm hoặc dịch vụ như tạp chí và vé máy bay qua Internet mà không cần phải đến trực tiếp cửa hàng hoặc trung tâm mua sắm như trước đây.

Từ cơ sở thực tiễn đó, em đã chọn đề tài: “**Nghiên cứu xây dựng ứng dụng bán cây cảnh trên nền tảng hệ điều hành Android**”. Mục đích của để tài này là đi sâu tìm hiểu cơ chết hoạt hoạt động của một hệ thống thương mại điện tử cũng như cách xây dựng một ứng dụng trên nền tảng hệ điều hành Android.

### 1.1.2 Mục tiêu đề tài

Báo cáo tập trung thực hiện các mục tiêu sau:

* Hiểu được tầm quan trọng, ý nghĩa của thương mại điện tử trong thực tiễn hiện nay.
* Hiểu được cơ bản cơ chế về nguyên lý cũng như cách hoạt động của ngôn ngữ lập trình Java.
* Biết xây dựng và phát triển một ứng dụng trên nền tảng hệ điều hành Android.

Biết cách kết hợp dịch vụ Firebase với các ứng dụng trên hệ điều hành Android.

### 1.1.3 Đối tượng và phạm vi nghiêm cứu

***Đối tượng nghiên cứu:***

* Các vấn đề về Thương mại điện tử.
* Các vấn đề về cách xây dựng ứng dụng trên nền tảng hệ điều hành Android với ngôn ngữ lập trình Java.
* Các vấn đề về dịch vụ Firebase với hệ điều hành Android.
* Cách triển khai và xây dựng hệ thống trên nền tàng PaaS.

***Phạm vi nghiêm cứu:***

Nghiên cứu về cách xây dựng hệ thông thương mại điện tử dựa trên nền tảng hệ điều hành Android và Firebase.

Công cụ hỗ trợ xây dựng hệ thống là Android Studio, Firebase Console và Figma.

### 1.1.4 Phương pháp nghiên cứu

Báo cáo sử dụng phương pháp nghiên cứu kết hợp giữa nghiên cứu lý thuyết bằng cách tổng hợp tài liệu và nghiên cứu thực nghiệm.

Nghiên cứu lý thuyết: Tổng hợp các vấn đề lý thuyết liên quan từ các nguồn tài liệu như Giáo trình, Bài giảng, Internet, chuyên đề, luận văn, luận án và tìm hiểu thực tiễn cách thức hoạt động của các hệ thông thương mại điện tử lớn hiện nay như Shopee, Tiki, Lazada để phát biểu bài toán “**Xây dựng ứng dụng bán cây cảnh trên nền tảng hệ điều hành Android**”.

Nghiên cứu thực nghiệm: Nghiên cứu cài đặt, thực hành xây dựng ứng dụng trên hệ điều hành Android với ngôn ngữ lập trình Java. Nghiên cứu kết hợp dịch vụ Firebase làm server back-end cho ứng dụng.

### 1.1.5 Ý nghĩa thực tiễn và khoa học của đề tài

Hiện nay, việc phát triển các hệ thống thương mại điện tử đang là lựa chọn phổ biến nhất cho các doanh nghiệp lớn cũng như các nhà bán lẻ[[2]](#footnote-3). Nó cung cấp phạm vi giao dịch và bán hàng trực tuyến rộng lớn hơn rất nhiều so với những phương pháp truyền thống. Không giống như một hệ thống thông thường, thường được sử dụng để tìm kiếm và thu thập thông tin, các hệ thống thương mại điện tử cho phép người dùng có thể xem, tương tác và mua hàng hóa, dịch vụ mà không cần phải đến trực tiếp cửa hàng như trước.

***Tầm quan trọng của thương mại điện tử***

Có thể nói, kinh doanh trên nền tảng thương mại điện tử là lựa chọn tốt nhất cho chúng ta khi xây dựng một thế giới kinh doanh tốt hơn, hiệu quả hơn để đảm bảo thành công trong tương lai thay vì thực hiện một phương thức kinh doanh truyền thống. Đối với bất kỳ nhà kinh doanh nào, việc áp dụng thương mại điện tử vào hệ thống là một lợi thế vô cùng lơn cho doanh nghiệp của họ. Dưới đây là một số yếu tố để chứng minh cho tầm quan trọng của kinh doanh thương mại điện tử:

**Tính tiện lợi (Convenience)**

Đôi khi, câu hỏi đặt ra là tại sao chúng ta phải sử dụng các trang web, ứng dụng thương mại điện tử để mua sắm trực tuyến? Câu trả lời phổ biến nhất cho câu hỏi này là chính là sự tiện lợi. Chúng ta có thể trao đổi, mua sắm sản phẩm khi đang ở bất cứ nơi đâu và bất cứ khoảng thời gian nào.

**Dịch vụ 24/24 (Round the clock service)**

Thương mại điện tử cung cấp cho chúng ta các dịch vụ 24/24 mọi lúc, mọi nơi. Vì vậy, khách hàng không cần phải đến nơi bán trực tiếp nếu như họ cần thứ gì đó thậm trí là vào ban đêm. Đó là lựa chọn thuận tiện nhất cho những người thường bận rộn với lịch trình làm việc của họ. Vì vậy, thương mại điện tử giúp các doanh nghiệp luôn luôn sẵn sàng để phục vụ khách hàng 24/7.

**Nền tảng rộng (Wide platform)**

Thương mại điện tử mang lại nhiều khách hàng và thông tin trên toàn quốc hoặc toàn cầu cho các doanh nghiệp. Do đó, việc lựa chọn một nền tảng thương mại điện tử để vượt qua các rào cản địa lý và thời gian cho doanh nghiệp là một lựa chọn sáng suốt.

**Xúc tiến kinh doanh (Business promotion)**

Hiện nay, các trang web, ứng dụng thương mại điện tử thường xuyên liên kết trực tiếp với các chương trình quảng bá, khuyến mãi kinh doanh…Điều này là rất quan trọng đối với sự phát triển kinh doanh của doang nghiệp vì nó không những thu hút khách hàng, tăng doanh thu cho doanh nghiệp sở hữu mà nó còn góp phần quảng bá hình ảnh, thông tin của doanh nghiệp tới rất đông người tiêu dùng trên toàn thế giới.



Hình : Vai trò của thương mại điện tử

**Chi phí vận hành thấp hơn so với các phương pháp truyền thống (Lesser cost)**

Nếu như việc quản lý hàng hóa và dịch vụ tồn kho được tự động hóa thì đây là một giải pháp không chỉ giúp giảm chi phí mà còn giảm rủi ro cho các doanh nghiệp. Ngoài ra, việc kinh doanh thương mại điện tử sẽ tiết kiệm chi phí hơn nhiều so với cửa hàng truyền thống vì nó giúp các doanh ngiệp tiết kiệm các chi phí phụ như tiền thuê mặt bằng, nhân viên bán hàng ...

**Thiết lập dễ dàng (Easy setup)**

Chi phí để thiết lập kinh doanh thương mại điện tử cực kỳ thấp so với việc thiết lập một cửa hàng truyền thống. Hơn nữa, việc cấp phép và cho phép trang web thương mại điện tử khá dễ dàng so với một cửa hàng thực tế.

**Thúc đẩy nền kinh tế (Economy)**

Vì không có khoản đầu tư nào về cơ sở hạ tầng hoặc bảo hiểm kinh doanh trong thương mại điện tử, nên các doanh nghiệp có thể đầu tư nhiều tiền hơn vào sản phẩm, chiến lược và quảng cáo. Điều này sẽ thúc đẩy các chiến lược tiếp thị của doanh nghiệp và do đó cũng có thể tăng lưu lượng truy cập trên các trang web và ứng dụng thương mại điện tử của các doanh nghiệp đó.

**Dễ dàng mang thông tin sản phẩm tới tay người tiêu dùng (Better product information)**

Hầu hết các trang web hay ứng dụng thương mại điện tử đều có thông tin chi tiết về sản phẩm của họ cùng với hình ảnh và video để giới thiệu sản phẩm. Điều này giúp khách hàng có được một cái nhìn tổng quan và rõ ràng về sản phẩm khi quyết định đặt mua một sản phẩm nào đó.

## 1.2 Công nghệ sử dụng

### 1.2.1 Java

Java được phát triển bởi James Gosling tại Sun Microsystems Inc vào năm 1995, sau đó được Oracle Corporation mua lại. Nó là một trong những ngôn ngữ lập trình đơn giản nhất hiện nay. Java giúp cho việc viết, biên dịch và gỡ lỗi chương trình trở nên dễ dàng hơn rất nhiều so với các ngôn ngữ lập trình khác. Nó giúp tạo mã có thể tái sử dụng và các chương trình mô-đun. Java là ngôn ngữ lập trình hướng đối tượng, dựa trên lớp (class) và được thiết kế để có càng ít phụ thuộc triển khai càng tốt. Một ngôn ngữ lập trình có mục đích chung được tạo cho các nhà phát triển để viết một lần chạy ở bất kỳ đâu (write one run anywhere) mà mã Java được biên dịch có thể chạy trên tất cả các nền tảng hỗ trợ Java. Các ứng dụng Java được biên dịch thành mã byte có thể chạy trên bất kỳ Máy ảo Java nào. Cú pháp của Java tương tự như C/C++…[[3]](#footnote-4)

**Các thuật ngữ Java:**

***Máy ảo Java (JVM):*** thường được gọi là JVM. Để biên dịch một chương trình Java, chúng ta phải trải qua ba giai đoạn: viết mã nguồn, biên dịch và chạy chương trình.

Quá trình biên dịch được thực hiện bởi trình biên dịch JAVAC, trình biên dịch Java chính có trong bộ công cụ phát triển Java (JDK). Nó lấy chương trình Java làm đầu vào và tạo mã byte làm đầu ra.

Trong giai đoạn Chạy chương trình, JVM thực thi mã byte do trình biên dịch tạo ra.

Chúng ta có thể hiểu rằng chức năng của Máy ảo Java là thực thi mã byte do trình biên dịch tạo ra. Mỗi Hệ điều hành có một JVM khác nhau nhưng đầu ra mà chúng tạo ra sau khi thực thi mã byte là giống nhau trên tất cả các hệ điều hành. Đây là lý do tại sao Java được biết đến như một ngôn ngữ độc lập với nền tảng.

***Mã byte(Bytecode):*** Như chúng ta đã biết, trình biên dịch Java của JDK biên dịch mã nguồn java thành mã byte để JVM có thể thực thi mã này. Nó được trình biên dịch lưu dưới dạng tệp .class. Để xem bytecode, có thể sử dụng một trình dịch ngược như javap.

***Bộ công cụ phát triển Java (JDK):*** chúng ta đã bắt gặp thuật ngữ JDK rất nhiều khi tìm hiểu về bytecode và JVM. JDK là một bộ công cụ phát triển Java hoàn chỉnh bao gồm mọi thứ bao cần thiết cho một lập trình viên: trình biên dịch, môi trường thời gian chạy Java (JRE), trình gỡ lỗi java, tài liệu java, v.v. Để chương trình thực thi trong java, chúng ta cần cài đặt JDK trên máy tính của mình.

***Môi trường chạy Java (JRE)****:* JDK bao gồm JRE. Cài đặt JRE trên máy tính của chúng ta cho phép chương trình java chạy. Tuy nhiên, chúng ta không thể biên dịch nó. JRE bao gồm trình duyệt, JVM, hỗ trợ applet và plugin. Để chạy chương trình java, máy tính cần có JRE.

***Bộ dọn rác (Garbage Collector):*** Trong Java, lập trình viên không thể xóa các đối tượng. Để xóa hoặc thu hồi bộ nhớ đó, JVM có một chương trình gọi là Garbage Collector. Garbage Collector có thể tự động thu hồi lại các đối tượng không được tham chiếu. Vì vậy, Java làm cho công viejc của một lập trình viên trở nên dễ dàng hơn rất bằng cách tự động quản lý bộ nhớ. Tuy nhiên, các lập trình viên nên cẩn thận về mã của họ cho dù họ đang sử dụng các đối tượng đã được sử dụng trong một thời gian dài. Bởi vì Rác không thể khôi phục bộ nhớ của các đối tượng được tham chiếu.

### 1.2.2 Android

Hệ điều hành Android là hệ điều hành di động dựa trên nhân Linux và chủ yếu chạy trên điện thoại thông minh và máy tính bảng.

Nền tảng Android bao gồm một hệ điều hành dựa trên nhân Linux, GUI, trình duyệt web và các ứng dụng người dùng cuối có thể tải xuống. Mặc dù những phiên bản đầu tiên của Android có một điện thoại thông minh QWERTY chung và màn hình VGA lớn, nhưng hệ điều hành này được viết để chạy trên các thiết bị cầm tay tương đối rẻ tiền với bàn phím số thông thường.[[4]](#footnote-5)

Android được phát hành theo giấy phép mã nguồn mở Apache v2, điều này cho phép nhiều biến thể của HĐH được phát triển cho các thiết bị khác, chẳng hạn như máy chơi game và máy ảnh kỹ thuật số. Android dựa trên phần mềm nguồn mở, nhưng hầu hết các thiết bị Android đều được cài đặt sẵn một bộ phần mềm độc quyền, chẳng hạn như Google Maps, YouTube, Google Chrome và Gmail.

**Lịch sử và phát triển**

Android bắt đầu hoạt động với tư cách là một công ty khởi nghiệp có trụ sở tại Palo Alto có tên là Android Inc vào năm 2003. Ban đầu, công ty bắt đầu phát triển một hệ điều hành cho máy ảnh kỹ thuật số, nhưng họ đã từ bỏ những nỗ lực đó thay vì tiếp cận thị trường rộng lớn hơn.[[5]](#footnote-6)

Google đã mua lại Android Inc và các nhân viên chủ chốt của nó vào năm 2005 với giá ít nhất là 50 triệu USD. Google đã tiếp thị nền tảng di động ban đầu cho các nhà sản xuất thiết bị cầm tay và nhà cung cấp dịch vụ di động với những lợi ích chính của nó là tính linh hoạt và khả năng nâng cấp.

Google đã bí mật phát triển hệ điều hành Android khi Apple phát hành iPhone vào năm 2007. Các nguyên mẫu trước đây của điện thoại Android gần giống với BlackBerry, với bàn phím vật lý và không có màn hình cảm ứng. Tuy nhiên, sự ra mắt của iPhone đã thay đổi đáng kể thị trường điện toán di động và buộc các nhà sáng tạo Android phải hỗ trợ màn hình cảm ứng nhiều hơn. Tuy nhiên, HTC Dream, điện thoại thông minh thương mại đầu tiên chạy hệ điều hành Android, có bàn phím QWERTY và đã nhận được một số phản hồi tích cực trong lần phát hành năm 2008.

Cuối năm 2007, Open Handset Alliance (OHA) tuyên bố thành lập. OHA là một liên minh gồm hơn 30 công ty phần cứng, phần mềm và viễn thông, bao gồm Google, Qualcomm, Broadcom, HTC, Intel, Samsung, Motorola, Sprint, Texas Instruments và các nhà mạng không dây Nhật Bản KDDI và NTT DoCoMo. Mục tiêu của liên minh là góp phần phát triển nền tảng nguồn mở đầu tiên cho thiết bị di động.

Google đã phát hành phiên bản beta công khai của Android 1.0 cho các nhà phát triển vào cùng thời điểm thông báo của liên minh, vào tháng 11 năm 2007. Mãi cho đến khi Google phát hành Android 1.5 vào tháng 4 năm 2009, Google mới giới thiệu sơ đồ đặt tên theo chủ đề món tráng miệng đặc trưng của Android; tên của Android 1.5 là "Cupcake." Vào khoảng thời gian phát hành Android 4.4 KitKat, Google đã đưa ra một tuyên bố chính thức để giải thích việc đặt tên: "Vì những thiết bị này khiến cuộc sống của chúng ta trở nên ngọt ngào nên mỗi phiên bản Android được đặt tên theo một món tráng miệng."

**Các tính năng của hệ điều hành Android**

Giao diện người dùng mặc định của Android dựa trên các đầu vào thao tác trực tiếp như chạm, vuốt và chụm. Thiết bị cung cấp phản hồi xúc giác cho người dùng thông qua các cảnh báo như rung để phản hồi các hành động. Ví dụ: nếu người dùng nhấn nút điều hướng, thiết bị sẽ rung.

Khi người dùng khởi động thiết bị, hệ điều hành Android sẽ hiển thị màn hình chính, đây là trung tâm điều hướng chính cho thiết bị Android và bao gồm các tiện ích và biểu tượng ứng dụng. Widget là màn hình thông tin tự động cập nhật nội dung như thời tiết hoặc tin tức. Màn hình chính có thể khác nhau tùy theo nhà sản xuất thiết bị đang chạy hệ điều hành nào. Người dùng cũng có thể chọn các chủ đề khác nhau cho màn hình chính thông qua các ứng dụng của bên thứ ba trên Google Play.

Thanh trạng thái ở đầu màn hình chính hiển thị thông tin về thiết bị và khả năng kết nối của thiết bị, chẳng hạn như mạng Wi-Fi mà thiết bị được kết nối hoặc cường độ tín hiệu. Người dùng có thể kéo thanh trạng thái xuống bằng cách vuốt ngón tay để xem màn hình thông báo.

Hệ điều hành Android cũng bao gồm các tính năng để tiết kiệm sử dụng pin. Hệ điều hành tạm dừng các ứng dụng không được sử dụng để tiết kiệm pin và mức sử dụng CPU. Android bao gồm các tính năng quản lý bộ nhớ tự động đóng các quy trình không hoạt động được lưu trữ trong bộ nhớ của nó.

Android chạy trên cả hai tiêu chuẩn di động được triển khai rộng rãi nhất là GSM/HSDPA và CDMA/EV-DO. Android cũng hỗ trợ:

* Bluetooth
* Edge
* Các giao thức truyền thông 3G, như EV-DO và HSDPA
* Wifi
* Tự động sửa lỗi
* Nhắn tin SMS và MMS
* Máy quay video/tĩnh kỹ thuật số
* GPS
* La bàn
* Tăng tốc đồ họa 3D
* Ứng dụng đa nhiệm

**Phần cứng (Hardware)**

Android sử dụng ARM cho nền tảng phần cứng của nó; các phiên bản mới hơn của hệ điều hành Android hỗ trợ kiến trúc x86 và x86-64. Bắt đầu từ năm 2012, các nhà sản xuất thiết bị đã phát hành điện thoại thông minh và máy tính bảng Android với bộ xử lý Intel.

Yêu cầu phần cứng tối thiểu của Android phụ thuộc vào kích thước màn hình, loại và mật độ CPU của thiết bị. Ban đầu, Google yêu cầu bộ xử lý 200 MHz, dung lượng lưu trữ 32 MB và RAM 32 MB.

Google phát hành tài liệu về các yêu cầu phần cứng mà các nhà sản xuất thiết bị gốc (OEM) phải đáp ứng để một thiết bị được "Được Google phê duyệt", có nghĩa là thiết bị đó sẽ xuất xưởng cùng với các ứng dụng chính thức của Google. Tuy nhiên, bản chất mã nguồn mở của Android có nghĩa là nó cũng có thể chạy trên phần cứng thấp hơn và ngược lại.

**So sánh với các hệ điều hành di động khác**

Ban đầu, những người tạo ra Android tin rằng hệ điều hành này sẽ cạnh tranh với các hệ điều hành di động khác như Symbian và Microsoft Windows Mobile.

Symbian là một hệ điều hành đóng với vi hạt nhân và giao diện người dùng cung cấp lớp vỏ đồ họa. Nhiều nhà sản xuất di động đã sử dụng hệ điều hành Symbian, bao gồm Nokia, Samsung và Motorola. Symbian là một hệ điều hành phổ biến trên toàn thế giới, nhưng nó không được phổ biến rộng rãi ở Bắc Mỹ. Tuy nhiên, thiết kế của Symbian không đơn giản như Android và iOS, và hệ điều hành này rất khó lập trình. Quá trình phát triển hệ điều hành Symbian đã bị ngừng vào năm 2014.

Windows Mobile có nguồn gốc từ Windows CE, một hệ điều hành nhúng và lần đầu tiên xuất hiện trên Pocket PC 2000. Microsoft đã tiếp thị hệ điều hành di động cho các doanh nghiệp. Sự cạnh tranh từ Android và iOS buộc Microsoft phải thay đổi; công ty đã thay thế Windows Mobile bằng Windows Phone vào năm 2010, nhằm vào thị trường tiêu dùng. Microsoft đã loại bỏ Windows Phone để chuyển sang Windows 10 Mobile, nhưng hệ điều hành đó cũng đã bị ngừng sản xuất; Microsoft đã tuyên bố hết hạn sử dụng vào ngày 14 tháng 1 năm 2020.

Đối thủ cạnh tranh chính của Android là Apple iOS. Cả hệ điều hành iOS và Android đều cung cấp các tính năng so sánh. Apple iOS là một hệ điều hành độc quyền với giao diện cố định, trong khi Android là một hệ điều hành mã nguồn mở cung cấp nhiều tính linh hoạt và tùy biến hơn.

Theo báo cáo của Statcount, thị phần toàn cầu của Android vào năm 2022 là 71,85%. Thị phần toàn cầu của Apple iOS là 27,5%. Tuy nhiên, tại Hoa Kỳ, Apple thống trị thị phần với 55,25%; Android chiếm 44,43%, tiếp theo là Samsung với 0,27% và Windows với 0,02%.

**Một vài sự chỉ trích với hệ điều hành Android**

Người dùng chỉ trích Android nhiều nhất là hệ điều hành này bị phân mảnh. Bản chất nguồn mở, linh hoạt của Android dẫn đến nhiều biến thể của phần cứng và phần mềm. Nhiều thiết bị chạy các phiên bản Android cũ hơn. Tính đến tháng 7 năm 2022, 29,63% người dùng Android chạy hệ điều hành phiên bản 11, 21,8% chạy phiên bản 10, 20,86% sử dụng phiên bản 12 và 10,74% sử dụng phiên bản 9, theo Statcount.

Phân mảnh thiết bị tạo ra thách thức cho các nhà phát triển vì rất khó để phát triển các ứng dụng hoạt động trên tất cả các loại và phiên bản thiết bị. Sự manh mún cũng là một vấn đề đối với các doanh nghiệp; Nhân viên CNTT không thể dễ dàng bảo mật và quản lý các thiết bị chạy trên nhiều loại phần cứng và phần mềm. Google đã khởi chạy Project Treble như một giải pháp tiềm năng cho vấn đề này. Sáng kiến này tách hệ điều hành Android khỏi các sửa đổi OEM và cho phép triển khai các bản cập nhật phần mềm nhanh hơn.

Một lời chỉ trích khác về hệ điều hành Android là các ứng dụng Android có thể dễ dàng bị vi phạm bản quyền. Tuy nhiên, với việc phát hành Android Jelly Bean, Google đã cung cấp cho các nhà phát triển khả năng mã hóa các ứng dụng trả phí.

### 1.2.3 Firebase

Google Firebase là phần mềm phát triển ứng dụng do Google hỗ trợ, cho phép các nhà phát triển phát triển ứng dụng iOS, Android và Web. Firebase cung cấp các công cụ để theo dõi phân tích, báo cáo và khắc phục sự cố ứng dụng, tạo thử nghiệm tiếp thị và sản phẩm.[[6]](#footnote-7)

**Firebase cung cấp rất nhiều dịch vụ, bao gồm:**

**Analytics (Phân tích)**: Google Analytics cho Firebase cung cấp báo cáo miễn phí, không giới hạn về tối đa 500 sự kiện riêng biệt. Analytics trình bày dữ liệu về hành vi của người dùng trong ứng dụng iOS và Android, giúp đưa ra quyết định tốt hơn về việc cải thiện hiệu suất và tiếp thị ứng dụng.

**Authentication (Xác thực):** Firebase giúp các nhà phát triển dễ dàng xây dựng các hệ thống xác thực an toàn và nâng cao trải nghiệm đăng nhập cũng như tích hợp cho người dùng. Tính năng này cung cấp giải pháp nhận dạng hoàn chỉnh, hỗ trợ tài khoản email và mật khẩu, xác thực điện thoại, cũng như đăng nhập Google, Facebook, GitHub, Twitter, v.v.

**Firebase Cloud Messaging (FCM)**: là một công cụ nhắn tin đa nền tảng cho phép các công ty nhận và gửi tin nhắn một cách đáng tin cậy trên iOS, Android và web mà không mất phí.

**Realtime database (Cơ sở dữ liệu thời gian thực)**: Cơ sở dữ liệu thời gian thực Firebase là cơ sở dữ liệu NoSQL được lưu trữ trên đám mây cho phép dữ liệu được lưu trữ và đồng bộ hóa giữa những người dùng trong thời gian thực. Dữ liệu được đồng bộ hóa trên tất cả các máy khách trong thời gian thực và vẫn khả dụng khi ứng dụng ngoại tuyến.

**Crashlytics (Báo cáo sự cố)**: là trình báo cáo sự cố theo thời gian thực giúp các nhà phát triển theo dõi, ưu tiên và khắc phục các sự cố về độ ổn định làm giảm chất lượng ứng dụng của họ. Với crashlytics, các nhà phát triển dành ít thời gian hơn cho việc tổ chức và khắc phục sự cố và có nhiều thời gian hơn để xây dựng các tính năng cho ứng dụng của họ.

**Performance (Hiệu suất):** Dịch vụ Giám sát hiệu suất Firebase cung cấp cho nhà phát triển cái nhìn sâu sắc về các đặc điểm hiệu suất của ứng dụng iOS và Android của họ để giúp họ xác định vị trí và thời điểm có thể cải thiện hiệu suất của ứng dụng.

**Test lab (Thử nghiệm)**: Phòng thử nghiệm Firebase là cơ sở hạ tầng thử nghiệm ứng dụng dựa trên đám mây. Với một thao tác, nhà phát triển có thể thử nghiệm ứng dụng iOS hoặc Android của họ trên nhiều loại thiết bị và cấu hình thiết bị. Họ có thể xem kết quả, bao gồm video, ảnh chụp màn hình và nhật ký, trong bảng điều khiển Firebase.

Các trường hợp sử dụng Firebase bao gồm:

**Create onboarding flows (Tạo quy trình giới thiệu)**: nhà phát triển có thể cung cấp cho người dùng quy trình đăng nhập nhanh chóng, trực quan bằng cách sử dụng Xác thực Firebase. Chúng cho phép người dùng đăng nhập vào ứng dụng của họ thông qua tài khoản Google, Twitter, Facebook hoặc GitHub trong vòng chưa đầy năm phút. Các nhà phát triển cũng có thể theo dõi từng bước trong quy trình giới thiệu của họ để nâng cao trải nghiệm người dùng. Ngoài ra, nhà phát triển có thể sử dụng Google Analytics cho Firebase để ghi nhật ký sự kiện ở mỗi bước trong quy trình giới thiệu của họ, tạo kênh để xác định nơi người dùng rời đi và sử dụng cấu hình từ xa để thực hiện các thay đổi đối với ứng dụng của họ nhằm xem những thay đổi đó ảnh hưởng như thế nào đến chuyển đổi.

# CHƯƠNG II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Phân tích, thiết kế hệ thống thông tin có vai trò rất quan trọng trong quy trình xây dựng phần mềm vì nó giúp các nhà phát triển có được cái nhìn đầy đủ, đúng đắn và chính xác về hệ thống thông tin mà họ sẽ xây dựng trong tương lai nhằm thuận lợi cho việc bảo trì, bổ sung và phát triển hệ thống khi có yêu cầu. Hơn thế nữa, việc phân tích và thiết kế hệ thống còn tránh được những sai lầm trong thiết kế, cài đặt.

## 2.1 Xác định yêu cầu bài toán

Để bắt đầu thiết kế một hệ thống thương mại điện tử, trước tiên chúng ta cần xác định từng nhóm bị ảnh hưởng bởi dự án đó và phải hiểu được nhu cầu của mọi người. Với thông tin đó những ý tưởng ban đầu về các yêu cầu chức năng và phi chức năng có thể được xác định.

### 2.1.1 Yêu cầu chức năng

Để chứng minh giá trị của một nền tảng thương mại điện tử và xác định sự hoàn thiện của các chức năng, ứng dụng hoặc trang web thương mại điện tử phải có tất cả các tính năng phổ biến của một cửa hàng truyền thống thông thường. Theo đó, người ta đã xem xét rằng bộ chức năng cần thiết nhất cho hệ thống này bao gồm những chức năng liên quan đến xem và mua sản phẩm, cũng như quản lý tài khoản khách hàng. Yêu cầu chi tiết dự kiến cho cửa hàng trực tuyến sẽ được mô tả bên dưới.

Trong trang chủ tất cả các sản phẩm được hiển thị sắp xếp theo mức độ phổ biến. Từ đây, người dùng có thể chọn một danh mục, sau đó tất cả các sản phẩm thuộc danh mục đó hoặc bất kỳ sản phẩm con nào sẽ được hiển thị. Bất cứ khi nào một bộ sản phẩm được liệt kê, những sản phẩm đang giảm giá đó sẽ được đánh dấu và người dùng sẽ được cung cấp tùy chọn sắp xếp và lọc giữa tất cả các sản phẩm. Việc sắp xếp có thể được thực hiện theo tên, giá hoặc lọc theo màu sắc. Mỗi hình thu nhỏ của sản phẩm bao gồm hình ảnh của sản phẩm, tên và giá của sản phẩm cũng như danh sách tất cả các biến thể màu hiện có. Nhấp vào hình thu nhỏ của sản phẩm sẽ chuyển hướng người dùng đến trang chi tiết sản phẩm, nơi bên cạnh tên, giá, mô tả và đánh giá của khác hàng về sản phẩm cũng sẽ được hiển thị. Tại đây, người dùng có thể chọn bất kỳ màu sắc và kích thước nào để hiển thị hình ảnh tương ứng. Bất cứ lúc nào người dùng có thể thêm sản phẩm đã chọn vào giỏ hàng, sau đó nội dung giỏ hàng được cập nhật và tổng giá sẽ được hiển thị. Truy cập vào chi tiết giỏ hàng, khách hàng cũng có thể thay đổi số lượng đơn vị của một mặt hàng hoặc xóa bất kỳ mặt hàng cụ thể nào khỏi giỏ hàng.

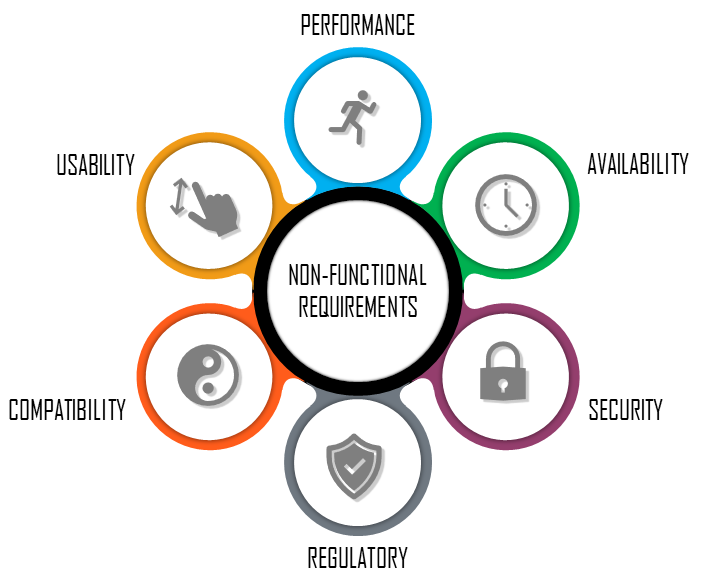
Từ đây, người dùng có thể chọn bắt đầu quy trình thanh toán, khi đó người mua được yêu cầu điền vào biểu mẫu có thông tin giao hàng (tức là địa chỉ và phương thức giao hàng) và thông tin thanh toán (tức là địa chỉ thanh toán và dữ liệu thanh toán). Trong quá trình thanh toán, tóm tắt đơn đặt hàng (tức là danh sách các mặt hàng đã mua và chi tiết giá cả) được hiển thị đồng thời và được cập nhật. Ngay trước khi kết thúc quá trình thanh toán, người dùng được thông báo về tất cả thông tin giao hàng và thanh toán được giới thiệu cũng như tóm tắt đơn hàng. Sau khi quá trình thanh toán kết thúc, một bản tóm tắt khác được hiển thị cùng với thông báo đặt hàng thành công.

Điều kiện đầu tiên để người dùng có thể mua hàng là phải đăng ký tài khoản trên hệ thống. Trong trường hợp đó, người mua phải cung cấp tên đầy đủ, địa chỉ email và mật khẩu của mình. Sau khi đăng ký, người dùng được chuyển hướng đến hồ sơ người dùng của mình, nơi khách hàng có thể cập nhật dữ liệu cá nhân, thay đổi mật khẩu, quản lý sổ địa chỉ hoặc xem lại chi tiết các đơn đặt hàng trước đó của mình. Sổ địa chỉ cho phép người dùng lưu trữ một tập hợp các địa chỉ gửi thư mà sau này có thể được chọn làm địa chỉ giao hàng hoặc thanh toán trong quá trình thanh toán. Người dùng được phép thêm địa chỉ mới vào sổ địa chỉ, cũng như cập nhật hoặc xóa bất kỳ địa chỉ đã lưu nào. Khi đăng nhập, người dùng có thể chọn đăng xuất để trở thành khách hàng ẩn danh. Bất cứ lúc nào, người dùng có thể đăng nhập lại bằng cách cung cấp địa chỉ email và mật khẩu của mình. Trong trường hợp người dùng quên mật khẩu, họ có thể yêu cầu khôi phục mật khẩu bằng cách nhập địa chỉ email của mình, trong trường hợp đó, một email sẽ được gửi đến địa chỉ được cung cấp có chứa liên kết web có thể được truy cập trong vòng một giờ tới, nơi người dùng có thể tạo một mật khẩu mới.

### 2.1.2 Yêu cầu phi chức năng

Việc xác định các yêu cầu phi chức năng là bước vô cùng quan trọng nhất để đảm bảo rằng các hệ thống có thể được vận hành một cách hiệu quả, ổn định, dễ dàng mở rộng và đáp ứng yêu cầu của khách hàng. Yêu cầu phi chức năng có thể tạo ra hoặc phá vỡ sự thành công của một sản phẩm hoặc hệ thống.

Mục tiêu cuối cùng của một dự án là cung cấp một sản phẩm chất lượng cao đúng như mong đợi của các bên liên quan. Các yêu cầu chức năng và phi chức năng là cách chính để thực hiện điều này.



Hình : Yêu cầu phi chức năng

Các yêu cầu phi chức năng của đề tài này bao gồm:

* Tính khả dụng (usability)
* Tính bảo mật (security)
* Hiệu suất (performance)
* Khả năng bảo trì (mantainability)
* Khả năng mở rộng (scalability)

## 2.2 Phân tích hệ thống

Các phương pháp Agile đề xuất rằng chỉ nên xây dựng tài liệu mô tả chi tiết hệ thống khi cần thiết khi mà không có bất kỳ thay đổi nào bắt buộc cho từng giai đoạn như các phương pháp truyền thống thường làm. Lý do cách viết theo tài liệu truyền thống không được dùng ở chương này là vì các yêu cầu dự kiến sẽ thay đổi liên tục trong quá trình phát triển, buộc phải cập nhật mọi sơ đồ và văn bản mỗi khi có yêu cầu thay đổi dẫn đến mất thời gian đáng lẽ có thể được sử dụng để phát triển sản phẩm.

Do đó, báo cáo này sẽ sử dụng các biểu đồ đơn giản (use case diagrams, sequence diagrams…) để phân thích và thiết kế hệ thống nhằm mục đích giúp người đọc hiểu rõ hơn về hệ thống.

### 2.2.1 Sơ đồ use case (Use case diagrams)

Diagram

Description automatically generated

Hình : Các tác nhân trong hệ thống

Hình 3 cho thấy có bốn tác nhân tương tác với hệ thống: admin, khách hàng, nền tảng thanh toán và nền tảng thương mại điện tử SP.IO. Khách hàng có thể là khách hàng ẩn danh hoặc khách hàng được xác định đã tồn tại trước đó trong nền tảng SP.IO (khách hàng đã đăng ký tài khoản trước đó). Do các chức năng bắt buộc của hệ thống hiện tại chủ yếu được thiết kế để thử nghiệm nền tảng SP.IO nên nền tảng này sẽ có mặt trong mọi use case đơn lẻ của hệ thống. Tuy nhiên để dễ dàng cho người đọc, nó sẽ không được đưa vào các sơ đồ use case.

Diagram

Description automatically generated

Hình : Sơ đồ use case tổng quát của hệ thống

Như đã đề cập trước đó, hệ thống có ba chức năng chính phù hợp với mọi use case bao gồm: hiển thị sản phẩm, mua sản phẩm và quản lý tài khoản khách hàng. Khách hàng có mặt trong tất cả các use case của hệ thống, trong khi nền tảng thanh toán và admin chỉ liên quan đến chức năng mua sản phẩm. Để hoàn tất quy trình đặt hàng, khách hàng sẽ cần nhập thông tin của mình vào form thanh toán (tên, địa chỉ, số điện thoại, phương thức thanh toán…). Nếu thông tin không chính xác hoặc không tồn tại, sẽ có một thông báo lỗi hiện lên. Ngược lại, tức là khách hàng đã nhập đúng và đủ thông tin yêu cầu, đơn hàng của người dùng sẽ được đưa vào trạng thái chờ duyệt. Lúc này, admin sẽ tham gia vào hệ thống, admin có thể duyệt hoặc hủy đơn hàng của người dùng nếu như đơn hàng mà người dùng đặt không còn trong kho.

Diagram, schematic

Description automatically generated

Hình : Use case hiện thị sản phẩm

Các use case để hiển thị sản phẩm được hiển thị bên trên (Hình 5). Khách hàng có thể chọn hoặc liệt kê một bộ sản phẩm hoặc hiển thị một sản phẩm cụ thể. Các chức năng bổ sung khác có thể được áp dụng cho danh sách sản phẩm, riêng lẻ hoặc kết hợp với nhau, để thay đổi chính danh sách đó (tức là lọc) hoặc cách các sản phẩm được liệt kê (tức là sắp xếp và phân trang).

Hình 6 chỉ ra các use case liên quan đến việc mua sản phẩm. Chúng có thể được chia rõ ràng thành hai chủ đề khác nhau: một mặt là tất cả các use case để quản lý giỏ hàng (tức là thêm, cập nhật và xóa các mục), mặt khác là những mục liên quan đến việc đặt và liệt kê đơn đặt hàng. Khi đặt hàng, khách hàng có thể được yêu cầu thanh toán trực tuyến, trong trường hợp đó, nền tảng thanh toán sẽ cung cấp các phương tiện cần thiết. Khách hàng ẩn danh sẽ chỉ có thể xem các mặt hàng mà không thể thêm chúng vào mục yêu thích hay giỏ hàng, nhưng chỉ những khách hàng đã được xác định mới có thể liệt kê đơn đặt hàng của riêng mình, nếu không họ sẽ được yêu cầu xác định danh tính của mình.

Diagram, schematic

Description automatically generated

Hình : Sơ đồ use case mua sản phẩm

Cuối cùng, đối với các trường hợp sử dụng liên quan đến quản lý tài khoản (Hình 6), một khách hàng đã đăng ký tài khoản có thể quản lý thông tin cá nhân của mình (tức là thêm, cập nhật hoặc xóa thông tin) hoặc cập nhật địa chỉ của tài khoản (tức là thay đổi dữ liệu cá nhân hoặc mật khẩu của mình). Người dùng khi đã đăng ký cũng có thể quyết định đăng xuất khỏi hệ thống và trở thành khách hàng ẩn danh. Nếu như là một khách hàng ẩn danh, người dùng có thể đăng ký một tài khoản mới hoặc đăng nhập bằng tài khoản hiện có. Trong trường hợp người dùng không nhớ thông tin đăng nhập của mình, người dùng đó sẽ được cung cấp tùy chọn để khôi phục mật khẩu của mình.

Diagram, schematic

Description automatically generated

Hình : Use quản lý tài khoản cá nhân

Các use case được giải thích phía trên là hữu ích để xác định phạm vi của dự án và hiểu chức năng của nó. Các use case này có thể hữu ích để ước tính các tác vụ và xây dựng kế hoạch phát triển, cũng như hướng dẫn để xác định chức năng cần thiết để viết các test case cần thiết cho hệ thống. Nhưng các use case này không quá chi tiết cho các mục đích khác, chẳng hạn như xác định kiểm tra acceptance hoặc mô tả trình tự tương tác của người dùng với hệ thống. Những nhiệm vụ này yêu cầu mức độ trừu tượng cao hơn của các use case, tập trung vào mục tiêu của người dùng thay vì các chức năng, đôi khi được gọi là các use case cấp cao nhất. Use case cấp cao nhất mô tả một quy trình kinh doanh cơ bản duy nhất cho phép một người dùng có thể thực hiện một mục tiêu nào đó. Trong hệ thống này, chủ yếu có ba mục tiêu mà khách hàng có thể muốn đạt được khi sử dụng cửa hàng trực tuyến này. Trường hợp tiên là việc người dùng duyệt qua các danh mục sản phẩm và chọn những sản phẩm quan tâm. Tại một số thời điểm, người dùng có thể quyết định xem xét các mặt hàng đã chọn và cuối cùng mua chúng, đó là trường hợp thứ hai. Cuối cùng, là trường hợp liên quan đến việc kiểm tra tình trạng thanh toán hoặc giao hàng của đơn đặt hàng hoặc bất kỳ thông tin bổ sung nào có liên quan thông tin.

Tất cả các use case cấp thấp được xác định trước đó đang thực sự cung cấp các chức năng để đáp ứng những ba mục tiêu kể trên.

### 2.2.2 Biểu đồ tuần tự (Sequence diagrams)

Hầu như tất cả các use case cấp thấp (Phần 2.2.1) của đề tài này chỉ bao gồm một tương tác giữa người dùng và hệ thống. Điều này có thể không hữu ích cho các đề tài yêu cầu thông tin rất chi tiết về hệ thống sẽ được phát triển, có thể bởi vì hành vi của nó rất cụ thể và cần sự chi tiết.

Nhưng đây không phải là trường hợp của đề tài này, các use case được xác định ở đây chính xác là rất phổ biến giữa các cửa hàng trực tuyến, do đó, bất kỳ hoạt động nào được cung cấp bởi hệ thống này đều cần được giải thích chi tiết.

Như đã đề cập ở phần trước, các use case cấp cao nhất ở đây phù hợp hơn để mô tả người dùng giao tiếp với hệ thống. Điều này là do chúng cung cấp thông tin không chỉ về hành vi của hệ thống, mà còn về trình tự các tương tác mà khách hàng thường thực hiện để đạt được mục tiêu.

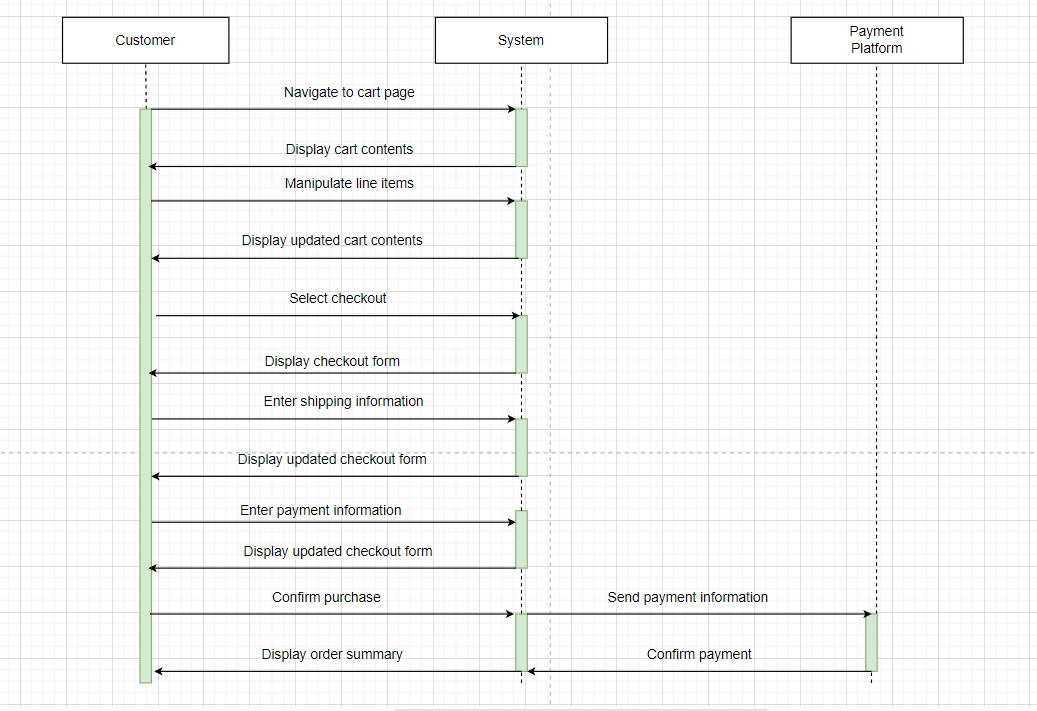
Hình 8 biểu thị sơ đồ tuần tự cho use case cấp cao nhất khi người dùng vào ứng dụng và chọn xem sản phẩm - một trong những kịch bản thành công có thể xảy ra. Trong trường hợp này, người dùng thường sẽ truy cập trang chủ, chọn một danh mục và sau đó lọc hoặc sắp xếp các sản phẩm cho đến khi tìm thấy một sản phẩm ưng ý. Sau đó người dùng có thể sẽ hỏi chi tiết về sản phẩm và tiếp theo anh ta sẽ thêm nó vào giỏ hàng.

Table

Description automatically generated

Hình : Quá trình người dùng duyệt sản phẩm

Use case cấp cao nhất của thanh toán được thể hiện trong Hình 9 phía dưới. Sau khi khách hàng đã thêm một số đơn hàng vào trong giỏ hàng của mình, bước tiếp theo là điều hướng đến trang giỏ hàng. Tại đây người dùng có thể xóa hoặc sửa đổi chi tiết đơn hàng của mình cho đến khi người dùng sẵn sàng bắt đầu quy trình thanh toán. Ở đó, sau khi nhập tất cả thông tin vận chuyển và thanh toán, khách hàng sẽ xác nhận việc mua hàng và hệ thống sẽ yêu cầu nền tảng thanh toán xử lý thanh toán, hiển thị chi tiết đơn đặt hàng theo phản hồi đến khách hàng.



Hình : Quá trình người dùng mua hàng và thanh toán

Biểu đồ trình tự cuối cùng thể hiện các tương tác mà khách hàng phải thực hiện theo thứ tự để kiểm tra trạng thái của một đơn đặt hàng (Hình 10). Kịch bản này yêu cầu khách hàng trước đây đăng ký vào hệ thống và mua một số mặt hàng như một khách hàng đã đăng ký. Rồi bất cứ lúc nào người dùng có thể truy cập trang đăng nhập và nhập thông tin đăng nhập để truy cập hồ sơ khách hàng của mình. Ở đó, người dùng có thể chọn để liệt kê tất cả các đơn đặt hàng của mình và chọn đơn hàng mà khách hàng đó muốn xem chi tiết.

Table

Description automatically generated with low confidence

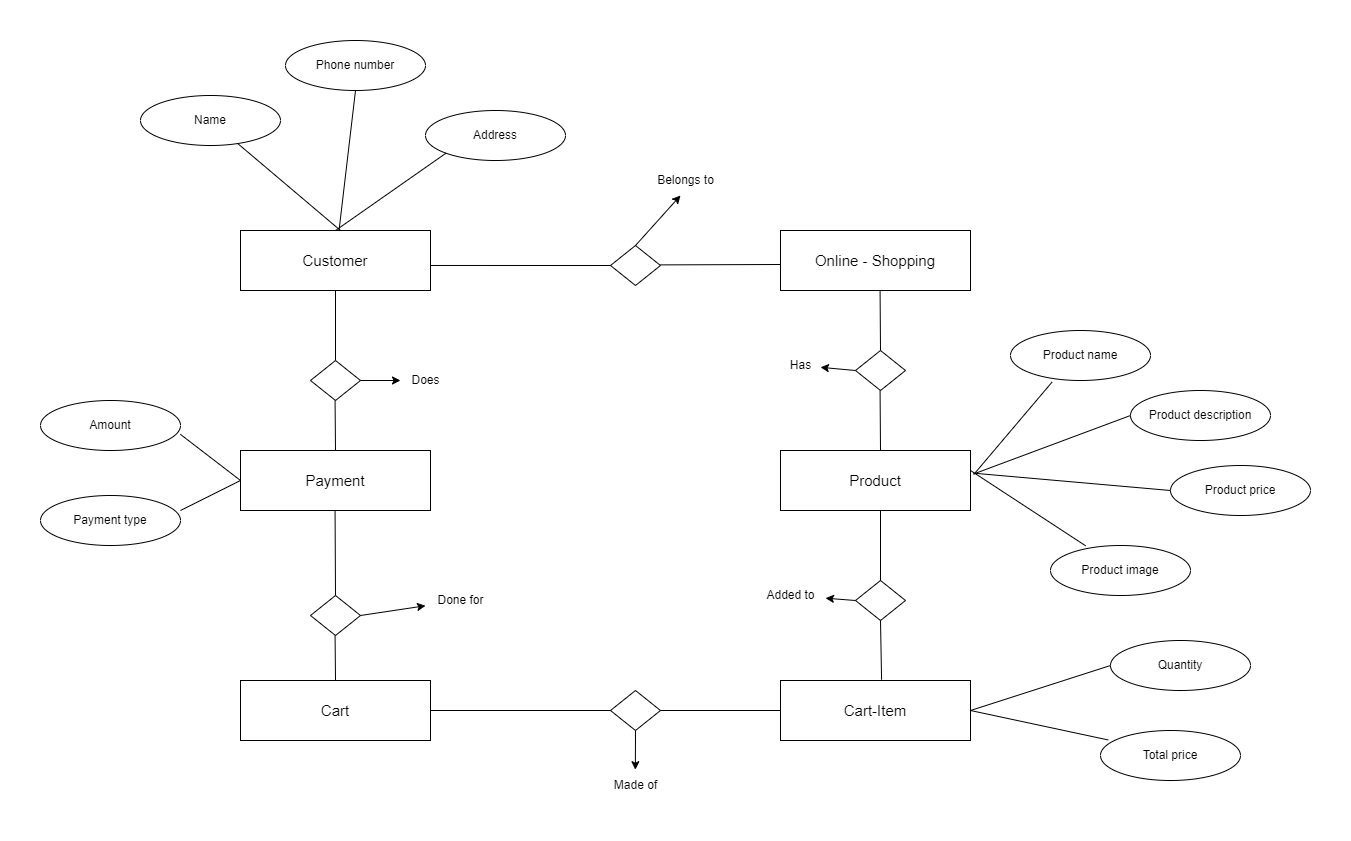
Hình : Quy trình kiểm tra trạng thái đơn hàng

### 2.2.3 Sơ quan hệ thực thể (ER diagram)

Sơ đồ quan hệ thực thể (ERD) cung cấp một cái nhìn trực quan cho thiết kế cơ sở dữ liệu và nó cũng có thể được sử dụng để giúp xác định các yêu cầu của hệ thống thông tin. Sau khi triển khai cơ sở dữ liệu quan hệ, ERD vẫn có thể đóng vai trò là điểm tham chiếu, nếu cần bất kỳ quá trình gỡ lỗi hoặc tái cấu trúc quy trình hệ thống nào sau này.

Tuy nhiên, mặc dù ERD có thể hữu ích để tổ chức dữ liệu có thể được biểu thị bằng cấu trúc quan hệ, nhưng nó không thể biểu thị đầy đủ dữ liệu bán cấu trúc hoặc không cấu trúc. Bản thân nó cũng không hữu ích trong việc tích hợp dữ liệu vào một hệ thống thông tin có sẵn.

Trong đề tài này, sơ đồ ERD sẽ lấy user làm đối tượng trung tâm. Thông tin user sẽ bao gồm những thông tin cơ bản như tên, địa chỉ và số điện thoại. Mỗi user sẽ thuộc về một cửa hàng trực tuyến. Trong cửa hàng trực tuyến sẽ có rất nhiều sản phẩm khác nhau. Thông tin sản phẩm bao gồm tên, mô tả, giá và hình ảnh. Khi người dùng muốn mua sản phẩm họ sẽ lựa chọn trong danh sách các product đã được hiện thị trên màn hình và thêm chúng và giỏ hàng (cart - item). Giỏ hàng sẽ bao gồm số lượng mà người dùng đã thêm và tổng giá mà người dùng phải trả. Thanh toán sẽ là bước cuối cùng để người dùng hoàn tất quá trình đặt hàng. Cụ thể sẽ được mô tả như hình bên dưới:



Hình : Sơ đồ ER

### 2.2.4 Biểu đồ trạng thái (Stage diagrams)

Có hai sơ đồ trạng thái của hệ thống này, cả hai đều liên quan đến yếu tố giỏ hàng. Sơ đồ đầu tiên (Hình 12) mô tả cách một thực thể giỏ hàng thay đổi cho đến khi nó trở thành một đơn đặt hàng hoàn chỉnh. Như sơ đồ bên dưới cho thấy, giỏ hàng hiện tại là trạng thái ban đầu, cho phép thay đổi nội dung của nó theo nhiều cách, chẳng hạn như thêm hoặc xóa chi tiết đơn hàng hoặc chọn địa chỉ giao hàng. Sau khi thanh toán xong, giỏ hàng trở thành đơn đặt hàng, đây là một thay đổi không thể đảo ngược. Từ giờ trở đi, lệnh chỉ có thể thay đổi từ trạng thái mở sang trạng thái hoàn thành và ngược lại.

Diagram

Description automatically generated

Hình : State diagram quá trình chuyển từ giỏ hàng thành đơn đặt hàng

Sơ đồ thứ hai (hình 13) mô tả toàn bộ quá trình quản lý giỏ hàng và cuối cùng mua những sản phẩm này trong quá trình thanh toán. Sơ đồ này sẽ trở thành đặc biệt hữu ích khi thiết kế giao diện thanh toán, vì nó hiển thị rõ ràng các yêu cầu của từng bước trong quy trình thanh toán.

Diagram

Description automatically generated

Hình : State diagram quản lý giỏ hàng

yêu cầu một phương thức giao hàng để hiển thị các tùy chọn thanh toán. Tất nhiên trình tự này có thể được bỏ qua nếu giỏ hàng đã có những yêu cầu này.

Khi người dùng cung cấp thông tin thanh toán và hoàn tất thanh toán, hệ thống sẽ tính phí khách hàng. Sau đó, đơn hàng được tạo sau khi nền tảng thanh toán xác nhận rằng khoản thanh toán đã thành công. Thời điểm giỏ hàng trước đó trở thành đơn đặt hàng, một giỏ hàng mới được tạo cho khách hàng để bắt đầu quá trình một lần nữa.

## 2.3 Thiết kế hệ thống

Thiết kế hệ thống là việc mô tả các chi tiết cuối cùng của một hệ thống trước khi nó được triển khai. Trong quá trình thiết kế, các quyết định được đưa ra để đáp ứng các yêu cầu chức năng và phi chức năng đã thu thập, các quyết định sau đó được áp dụng cho hệ thống được xác định trong phần Thông số kỹ thuật. Cả thiết kế vật lý và logic của hệ thống được sẽ được mô tả chi tiết trong phần này. Lựa chọn công nghệ, kiến trúc dự án là một quá trình mang tính quyết định nhằm đạt được kết quả tối ưu của dự án. Một quyết định thiếu khôn ngoan đôi khi có thể ảnh hưởng nghiêm trọng đến tổng nguồn lực cần thiết hoặc việc thực hiện thành công các mục tiêu đã đề ra. Điều quan trọng nữa là thiết kế chính xác cấu trúc của hệ thống, ví dụ như xác định và áp dụng các mẫu phần mềm có thể giải quyết các vấn đề hiện có trong dự án cụ thể này.

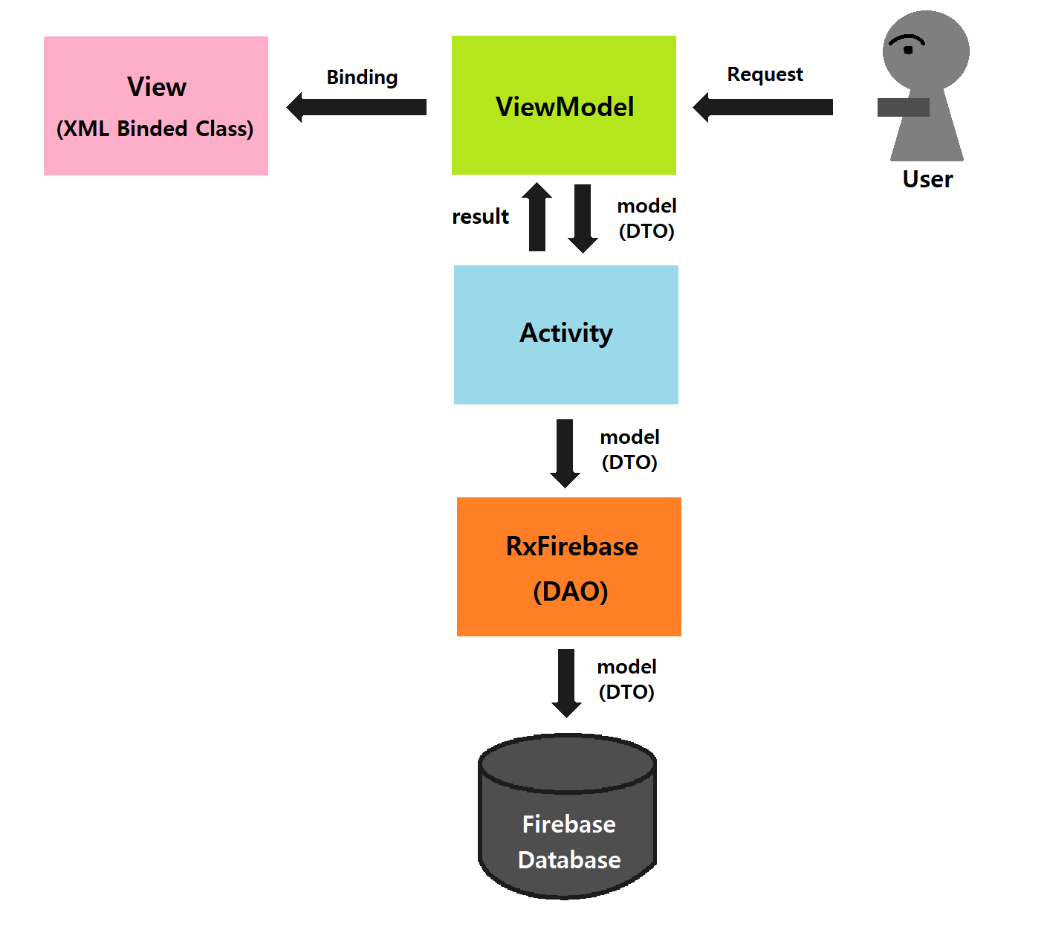
### 2.3.1 Thiết kế logic hệ thống

Kiến trúc logic của hệ thống được thiết kế theo mẫu kiến trúc client – server với client là những ứng dụng chạy hệ điều hành android và server dựa trên nên tảng firebase của google. Việc sử dụng mô hình kiến trúc MVVM trong đề tài này là không bắt buộc, tuy nhiên việc áp dụng mô hình MVVM trong phát triển một dự án android là hết sức cần thiết vì những lợi ích mà nó mang lại.

Như tên (MVVM) gợi ý, logic phía client được chia thành ba thành phần: Model, View và ViewModel. Theo định nghĩa chung, Model quản lý logic nghiệp vụ (business logic) và dữ liệu (data), View chịu trách nhiệm hiển thị thông tin và ViewModel chịu trách nhiệm thay đổi Model và View cho phù hợp với đầu vào của người dùng.

Thiết kế MVVM cụ thể của hệ thống hiện tại được thể hiện trong bên dưới. Một trong những điểm đặc biệt của thiết kế này là ViewModel - thành phần chính của Mô hình, vì nó kiểm soát tất cả dữ liệu của ứng dụng cũng như hầu hết logic nghiệp vụ.

Chỉ một số quy tắc nghiệp vụ (business rules) được thêm vào Mô hình để xác thực đầu vào biểu mẫu đến từ người dùng, trước khi gửi dữ liệu này tới SP.IO, cũng như một số chức năng bên ngoài như gửi email và thanh toán trực tuyến.



Hình : Mô hình kiến trúc hệ thống

Logic hệ thống được chia làm 2 phía, một là logic phía client và hai là logic phía backend.

**Logic phía client:**

Những nhà phát triển phần mềm luôn muốn tạo ra những mã nguồn dễ viết, dễ đọc và dễ bảo trì cho cấu trúc của các dự án. Việc tổ chức mã nguồn theo một mẫu thiết kế giúp bảo trì và tái sử dụng code trở lên dễ dàng và thuận tiện hơn. Hơn nữa, áp dụng một kiến trúc phần mềm nhất định trong thiết kế cũng đảm bảo rằng tất cả các mã nguồn được dễ dàng triển khai khi viết unit test mà không có sự phụ thuộc logic giữa các mã nguồn với nhau.

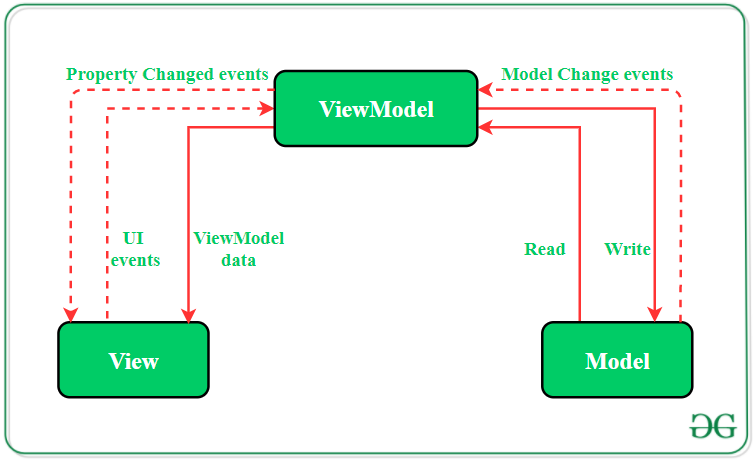
Model - View - ViewModel (MVVM) là mẫu kiến trúc phần mềm được nhiều lập trình viên trên thế giới khuyến khích sử dụng, nó khắc phục được mọi nhược điểm của các kiến trúc thiết kế MVP và MVC. MVVM đề xuất tách logic trình bày dữ liệu – presentation logic (View hoặc UI) khỏi phần logic nghiệp vụ (business logic) cốt lõi của ứng dụng.

**Mô hình MVVM được chia làm 3 lớp:**

**Model**: Lớp này chịu trách nhiệm trừu tượng hóa các nguồn dữ liệu. Model và ViewModel phối hợp với nhau để lấy và lưu dữ liệu.

**View**: Mục đích của lớp này là thông báo cho ViewModel về hành động của người dùng. Lớp này quan sát (observe) ViewModel và không chứa bất kỳ loại logic ứng dụng nào.

**ViewModel**: Nó hiển thị các luồng dữ liệu có liên quan đến View. Hơn nữa, nó phục vụ như một liên kết giữa Model và View.



Hình : Mô hình MVVM

Kiến trúc MVVM có một số điểm tương đồng với MVP (Model - View - Presenter) do ViewModel đóng vai trò Presenter. Tuy nhiên, nhược điểm của mẫu MVP đã được MVVM giải quyết bằng các cách sau:

* ViewModel không nắm giữ bất kỳ tham chiếu nào đến View.
* Quan hệ giữa Viewmodel và View là mốt quan hệ 1 - N.
* Không cần phải triển khai bất kỳ phương thức kích hoạt nào để cập nhật View.

**Các cách để triển khai MVVM trong đề tài này:**

* Sử dụng thư viện DataBinding do Google phát hành.
* Sử dụng RxJava cho DataBinding.

**Data Binding:**

Google phát hành Thư viện Data Binding Library cho Android cho phép các nhà phát triển liên kết các thành phần giao diện người dùng trong bố cục XML với kho lưu trữ dữ liệu của ứng dụng. Điều này giúp giảm thiểu mã logic ứng dụng cốt lõi liên kết với View. Hơn nữa, Liên kết dữ liệu hai chiều được thực hiện để liên kết các đối tượng với bố cục XML sao cho cả đối tượng và bố cục đó đều có thể gửi dữ liệu cho nhau.

**Ưu điểm của kiến trúc MVVM**

* Tăng cường khả năng tái sử dụng mã nguồn.
* Tất cả các mô-đun đều độc lập giúp cải thiện khả năng kiểm tra của từng class.
* Làm cho các dự án có thể duy trì và dễ dàng thực hiện các thay đổi.

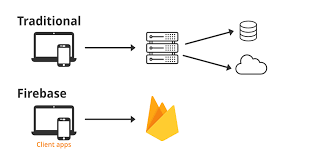
**Nhược điểm của kiến trúc MVVM**

* Kiến trúc này không phù hợp cho các dự án nhỏ.
* Nếu logic databinding quá phức tạp, việc gỡ lỗi ứng dụng sẽ khó hơn một chút.

**Logic phía server:**

Như đã đề cập ở phần trước, Firebase là một sản phẩm của Google giúp các nhà phát triển xây dựng, quản lý và phát triển ứng dụng của họ một cách dễ dàng. Nó giúp các nhà phát triển xây dựng ứng dụng của họ nhanh hơn và theo cách an toàn hơn. Không yêu cầu lập trình ở phía firebase, điều này giúp dễ dàng sử dụng các tính năng của nó hiệu quả hơn. Nó cung cấp dịch vụ cho android, ios, web và unity. Firebase cũng cung cấp lưu trữ trên cloud và sử dụng NoSQL cho cơ sở dữ liệu để lưu trữ dữ liệu.

Để giao tiếp giữa client và server (Firebase) ta chỉ cần tích hợp Firebase SDK vào ứng dụng. Tất cả giao tiếp giữa SDK Firebase và server đều được mã hóa an toàn, bất kể loại kênh được sử dụng. Điều này đúng trên bất kỳ nền tảng nào cung cấp SDK. Để xác thực, chúng ta có thể sử dụng JWT.



Hình : So sánh cách giao tiếp giữa client và server

Firebase đã phát triển từ Mobile-Backend-as-a-Service (MBaaS) thành một giải pháp back-end hiệu quả để phát triển ứng dụng web và di động. Firebase bao gồm nhiều công cụ và SDK khác nhau, cho phép tạo, quản lý, cải thiện và mở rộng ứng dụng của mình trên nhiều nền tảng khác nhau - web, iOS và Android.

Google Firebase cung cấp dịch vụ lưu trữ file, phân tích, xác thực người dùng, cơ sở dữ liệu thời gian thực, phân tích sự cố, lưu trữ đám mây và nhắn tin đẩy.

Firebase SDK hỗ trợ nhiều ngôn ngữ lập trình như C++, Java, JavaScript, Swift và Node.Js. Ngoài ra, nó mở rộng hỗ trợ cho React, Angular và Backbone thông qua liên kết cơ sở dữ liệu.

Một tính năng khác của Firebase – Cấu hình từ xa cho phép nhà phát triển thực hiện các thay đổi cần thiết đối với giao diện hoặc hành vi trực quan của ứng dụng mà không cần tải xuống phiên bản mới.

Tóm lại, ta có thể coi Firebase là Nền tảng là Dịch vụ - Platform as a Service (PaaS) và Phần cuối là Dịch vụ - Backend as a Service (BaaS).

### 2.3.2 Thiết kế giao diện

Đối với đề tài này, em đã quyết định triển khai cấu trúc bố cục và tương tác lấy cảm hứng từ các ứng dụng thương mai điện tử khác nhau với thiết kế sáng tạo, luôn tập trung vào việc cung cấp cho người dùng sự tương tác mượt mà với hệ thống. Mỗi màn hình sẽ được trình bày và mô tả bằng các hình ảnh, sau đó các yêu cầu và minh họa các kết nối giữa các màn hình đó.

Graphical user interface, application

Description automatically generated

Hình : Màn hình đăng ký và đăng nhập

Hình 17 mô tả màn hình đăng ký và đăng nhập. Với màn hình đăng ký, người dùng cần nhập tên, email và mật khẩu. Sau khi nhập đầy đủ thông tin, người dùng bấm nút “Sign up” để hoàn tất quá trình đăng ký. Với màn hình đăng nhập, người dùng nhập email và mật khẩu đã đăng ký và bấm “Log in” để chuyển hướng tới màn hình trang chủ.

Graphical user interface

Description automatically generated

Hình 18 trình bày màn hình trang chủ, tại đây người dùng có thể duyệt qua tất cả các sản phẩm. Các sản phẩm được chia theo thể loại nhằm mục đích giúp người dùng có thể dễ dàng tìm kiếm.

Hình : Màn hình trang chủ

Graphical user interface, text

Description automatically generated

Hình 19 trình bày màn hình chi tiết sản phẩm sau khi người dùng click vào sản phẩm hiển thị trên màn hình trang chủ. Những thông tin sẽ được hiện thị bao gồm: Hình ảnh, tên sản phẩm, giá tiền, đánh giá và mô tả chi tiết sản phẩm…

Để thêm sản phẩm vào giỏ hàng hoặc thêm vào mục yêu thích, người dùng có thể thao tác trực tiếp với các nút bấm trên màn hình

Hình : Màn hình chi tiết sản phẩm

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình : Màn hình yêu thích và giỏ hàng

Hình 20 trình bày những sản phẩm trong mục yêu thích và giỏ hàng của người dùng. Tại màn hình yêu thích người dùng có thể xóa cũng như thêm toàn bộ sản phẩm vào giỏ hàng. Tại màn hình giỏ hàng, người dùng có thể lựa chọn một hoặc toàn bộ sản phẩm để tiến hành thanh toán.

Graphical user interface, application

Description automatically generated

Hình : Màn hình thanh toán

Hình 21 mô tả quá trình thanh toán, tại đây, người dùng cần nhập thông tin về địa chỉ nhận hàng, phương thức thanh toán và mã giảm giả (nếu có).

Graphical user interface, application

Description automatically generated

Hình 22 là màn hình thông báo, tại đây, người dùng có thể theo dõi thông tin đơn hàng mà mình đã đặt.

Hình : Màn hình thông báo

Graphical user interface, application

Description automatically generated

Hình 23 là màn hình cài đặt của người dùng. Tại đây người dùng có thể nắm bắt được những thông tin: đơn hàng đã đặt, những reviews đã đánh giá. Người dùng cũng có thể thay đổi địa chỉ đặt hàng và phương thức thanh toán.

Hình : Màn hình cài đặt

Graphical user interface, application

Description automatically generated

Hình : Màn hình trạng thái đơn hàng và cài đặt địa chỉ giao hàng

Graphical user interface, application, website

Description automatically generated

Hình : Màn hình phương thức thanh toán và reviews

# CHƯƠNG III: TRIỂN KHAI VÀ KIỂM THỬ HỆ THỐNG

Phần triển khai và kiểm thử hệ thống mô tả quy trình thực tế được tuân theo để phát triển ứng dụng cửa hàng trực tuyến được trình bày trong các chương trước. Một mặt được giải thích về quy trình phát triển(phần Triển khai xây dựng hệ thống 3.1), chẳng hạn như phần mềm hỗ trợ triển khai và các giai đoạn của quy trình phát triển. Mặt khác, là quá trình kiểm thử hệ thống để đảm bảo không gây ra bug trong quá trình release sản phẩm (phần kiểm thử hệ thống 3.2)

## 3.1 Triển khai xây dựng hệ thống

Quy trình phát triển dự án này bao gồm ba giai đoạn: development, staging và production. Môi trường phát triển là một máy di động với hệ điều hành android 5.0 trở lên. Mã nguồn được phát triển với sự hỗ trợ của Java IDE Intellij IDEA Community Edition và android studio chủ yếu được sử dụng cho các tính năng sửa lỗi và biên dịch mã nguồn. Hệ thống backend dựa trên nền tảng của Firebase.

Git được sử dụng như một hệ thống kiểm soát sửa đổi. Đặc điểm đáng chú ý nhất của Git là hệ thống phân tán của nó, trong đó mỗi người dùng có kho lưu trữ cục bộ của riêng mình, nơi các thay đổi được thực hiện. Chỉ khi nhà phát triển cho là thuận tiện, các thay đổi cục bộ sau đó mới được đồng bộ hóa với kho lưu trữ từ xa, do đó làm cho cả nhóm có thể truy cập chúng. Kho lưu trữ từ xa được lưu trữ bởi GitHub, với chức năng mạng xã hội rất thú vị hữu ích cho sự hợp tác trong tương lai với cộng đồng nhà phát triển.

Trong quá trình phát triển của đề tài này, kỹ thuật triển khai liên tục (Continuous Deployment) được áp dụng. Jenkins được sử dụng trong môi trường dàn dựng như một công cụ triển khai liên tục, kích hoạt quy trình triển khai hệ thống mỗi khi các thay đổi được đẩy đến kho lưu trữ từ xa. Cái này quy trình bao gồm xây dựng và thử nghiệm hệ thống, chạy thử nghiệm chấp nhận tự động và triển khai dự án vào sản xuất sau khi dàn dựng ổn định và sẵn sàng. Bất cứ khi nào các bước này không thành công tại một số điểm, quy trình sẽ dừng lại và phản hồi được đăng ký để giải quyết vấn đề.

Để điều này hoạt động, mọi tính năng mới được phát triển cho hệ thống phải luôn đi cùng với các thử nghiệm xác thực tính năng đó. Tự động hóa các thử nghiệm này trên một hệ thống dàn dựng cho phép hợp nhất hoàn hảo các đoạn mã nhỏ với dòng chính của dự án với tốc độ nhanh chóng. Việc hợp nhất mã cũng kích hoạt quy trình xem xét với tất cả các nhà phát triển tham gia, dẫn đến chất lượng mã nguồn cao hơn. Ngoài ra, các bài kiểm tra chấp nhận xác minh logic nghiệp vụ có thể được chạy mỗi lần để đảm bảo rằng các yêu cầu của dự án được đáp ứng.

A picture containing diagram

Description automatically generated

Hình : Quy trình triển khai hệ thống

## 3.2 Kiểm thử hệ thống

Phần này liệt kê các yêu cầu chức năng được sử dụng để tạo bảng test case, các test case được sử dụng để xác minh bảng yêu cầu và kết quả cho bảng test case**.**

Bảng 1 liệt kê các yêu cầu chức năng cho được xây dựng cho ứng dụng giỏ hàng trực tuyến, cùng với mô tả ngắn gọn về từng yêu cầu:

|  |  |
| --- | --- |
| STT | Yêu cầu chức năng |
| FR01 | Hệ thống phải có hai loại xác thực: xác thực cho người dùng và xác thực cho admin. |
| FR02 | Người dùng (cả người dùng ẩn danh và người dùng đã đăng ký tài khoản) có thể duyệt và xem sản phẩm. |
| FR03 | Người dùng (đã đăng ký tài khoản) có thể xem sản phẩm và thêm vào giỏ hàng. |
| FR04 | Admin có thể tạo mới, sửa, xóa sản phẩm cũng như là thể loại. |
| FR05 | Admin có thể quản lý tất cả người dùng đã đăng ký tài khoản. |
| FR06 | Admin có thể xem thông tin chi tiết (tên, địa chỉ, số điện thoại…) những người đã gửi yêu cầu đặt hàng. |
| FR07 | Người dùng (đã đăng ký tài khoản) không thể chuyển sang màn check out khi không có sản phẩm nào trong giỏ hàng |
| FR08 | Người dùng không thể đặt hàng nếu như không cung cấp đủ thông tin mà form thanh toán yêu cầu. |
| FR09 | Người dùng sẽ không thể đặt hàng nếu như cung cấp thông tin (địa chỉ, số điện thoại…) không chính xác. |

Bảng : Yêu cầu chức năng của hệ thống

**Test case:**

Bảng 2 thể hiện những test case được viết cho nhưng yêu cầu chức năng ở phía trên. Mỗi test case sẽ có số thứ tự và một mô tả ngắn.

|  |  |  |
| --- | --- | --- |
| FR No. | Test case No. | Mô tả |
| FR01  FR02 | TC01 | Dùng để test màn login và authentication cho admin. |
| TC02 | Dùng để test màn login và authentication cho người dùng. |
| FR03 | TC03 | Dùng để test những sản phẩm mà người dùng đã thêm vào giỏ hàng. |
| FR04  FR05 | TC04 | Dùng để test khi admin thêm, sửa, xóa thể loại mới. |
| TC05 | Dùng để test khi admin thêm, sửa, xóa sản phẩm mới. |
| FR06 | TC06 | Dùng để test khi admin yêu cầu xem thông tin tất cả người dùng đã đăng ký tài khoản. |
| FR07 | TC07 | Dùng để test khi admin yêu cầu xem thông tin tất cả người dùng đã đặt hàng thành công. |
| FR08 | TC08 | Dùng để test khi người dùng check out khi chưa có sản phẩm nào trong giỏ hàng. |
| FR9 | TC09 | Dùng để test khi người dùng submit đơn đặt hàng khi thông tin điền vào không tồn tại. |

Bảng : Mô tả test case

Danh sách sau đây bao gồm các bước mà người dùng nên thực hiện, các điều kiện cần đáp ứng để thực hiện thành công các test case và kết quả cuối cùng cần đáp ứng để có thể vượt qua các test case trên.

**TC01: Test Login và Authentiacation interface**

Input: Usename và Password

Output: Hiện thị thông báo “đăng nhập thành công” và chuyển hướng đến màn hình trang chủ

Chi tiết:

* Nếu (User == Valid User) 🡺 Hiển thị thông báo đăng nhập thành công và chuyển hướng đến màn hình trang chủ.
* Nếu (User != Valid User) 🡺 Hiện thị thông báo lỗi và cho người dùng nhập lại thông tin tài khoản và mật khẩu.

**TC02: Test khi người dùng xem những sản phẩm mà họ đã thêm vào giỏ hàng**

Mô tả: Hệ thống sẽ đưa ra tất cả những sản phẩm mà người dùng đã thêm vào giỏ hàng, tại đây người dùng có thể lựa chon một hoặc tất cả sản phẩm để tiến hàng thanh toán.

Input: Người dùng thêm sản phẩm vào giỏ hàng

Output: Chuyển hướng đến màn hình giỏ hàng và hiển thị những sản phẩm đã được thêm.

Chi tiết:

* Nếu (Tồn tại sản phẩm trong giỏ hàng): Người dùng có thể thêm mới, xóa hoặc lựa chọn bất kỳ sản phẩm nào để thanh toán.
* Nếu (Giỏ hàng rỗng – không có sản phẩm nào): Hiện thị một hình ảnh biểu thị giỏ hàng đang chưa có sản phẩm nào và ẩn nút check out.

**TC03: Test khi admin thêm hoặc sửa một thể loại nào đó.**

Mô tả: Admin có thể thêm hoặc sửa các danh mục sản phẩm (điều chỉnh tên, hình ảnh)

Input:

* User = Admin
* Selection = Categoroes

Output: Danh mục sản phẩm được thêm mới hoặc chỉnh sửa thành công

Chi tiết:

* Nếu (user == Admin và selection == categories) thay đổi như yêu cầu của admin và cập nhật thông tin nên hệ thống

**TC04: Test khi admin xem thông tin của tất cả người dùng đã đăng ký hệ thống**

Mô tả: Admin có thể xem thông tin của những người dùng đã đăng ký tài khoản trên hệ thống.

Input:

* User = Admin
* Selection = View Inforamation

Output: Thông tin người dùng

Chi tiết:

* Nếu người dùng tồn tại trong hệ thống: Hiện thị thông tin người dùng. Ngược lại, đưa ra danh sách trống.

**TC05: Test khi admin xem thông tin của những người đã đặt hàng thành công**

Mô tả: Có thể xem thông tin của tất cả người dùng đã đặt hàng thành công

Input:

* User = Admin
* Selection = Check information user

Output: Thông tin người dùng.

Chi tiết:

* Nếu (User == Admin và Selection = Check information user): Hiện thị thông tin người dùng. Ngược lại hiện thị thông tin không tồn tại.

**TC06: Test khi người dùng check out mà trong giỏ hàng không có sản phẩm nào**

Mô tả: Nếu trong giỏ hàng không có sản phẩm nào, nút check out sẽ bị ẩn, người dùng không thể click và nút check out nếu giỏ hàng của người dùng trống.

Input:

* User = User
* Selection = View Cart 🡺 Không có sản phẩm nào trong giỏ hàng

Output: Disable nút checkout

Chi tiết

* Nếu giỏ hàng của người dùng rỗng: disable nút checkout
* Nếu giỏ hàng của người dùng không rỗng: enable nút checkout

**TC07: Test khi người dùng submit form checkout khi thông tin không đầy đủ hoặc không chính xác**

Mô tả: Người dùng không thể submit form checkout nếu như thông tin người dùng nhập vào không đúng hoặc không đầy đủ.

Input:

* User = User
* Selection = Order form

Output: Người dùng đặt hàng thành công hoặc không thành công

Chi tiết:

* Nếu thông tin người dùng nhập vào không đầy đủ hoặc không chính xác: Hiển thị lỗi sau khi người dùng click nút order.
* Nếu thông tin người dùng nhập đã đầy đủ và chính xác: Thông báo đặt hàng thành công

**Kết quả:**

Phần này liệt kê các kết quả được tạo ra bằng cách chạy các test case. Bảng 3 liệt kê các test case đã được sử dụng trong khi test giao diện cùng với kết quả mong đợi và kết quả thực tế cho từng test case:

|  |  |  |
| --- | --- | --- |
| Test case number | Expected result | Actual result |
| TC01 | Pass | Pass |
| TC02 | Pass | Pass |
| TC03 | Pass | Pass |
| TC04 | Pass | Pass |
| TC05 | Pass | Pass |
| TC06 | Pass | Pass |
| TC07 | Pass | Pass |

Bảng : Kết quả kiểm thử

# KẾT LUẬN

Sau khi giải thích em đã lên ý tưởng và xây dựng như thế nào để biến dự án này thành hiện thực, em không muốn kết thúc báo cáo này mà không bày tỏ sự biết ơn khi dự án này đã đóng góp rất nhiều trong quá trình học tập và làm việc thực tế của em . Đây là một kỳ học rất đặc biệt khi em có cơ hội làm việc trong một môi trường rất năng động và đổi mới, xung quanh là những con người cực kỳ tài năng và tuyệt vời.

Ở đó, em được khuyến khích thử các phương pháp và công nghệ mới để giải quyết các vấn đề về thiết kế và triển khai. Trên thực tế, chính kiến thức công nghệ thu được mà tôi đánh giá cao nhất trong quá trình học tập của mình, đặc biệt là khi so sánh với kinh nghiệm trước đây của em, về cơ bản bao gồm các công nghệ truyền thống và lỗi thời. Kiến thức thu được này phần lớn bao gồm kiểm tra chức năng hoàn chỉnh của hệ thống thương mại điện tử, một chủ đề đang chờ xử lý trong chuyên môn của em và đời sống học thuật.

Làm việc với một môi trường phát triển phù hợp cũng là một trải nghiệm thú vị mà em chưa bao giờ có cơ hội thực hành trước đây. Em cũng rất thích tìm hiểu các phương pháp triển khai thực tế từ bên trong thay vì học lý thuyết từ một cuốn sách. Và không chỉ phương pháp, mà còn là triết lý làm việc của công ty, có thể tóm tắt đơn giản bằng cảm giác gia nhập một đội ngũ chuyên nghiệp quan tâm đến việc thực hiện mọi việc ngay từ đầu, coi trọng chất lượng hơn số lượng.

**Hạn chế và hướng phát triển**

Hiện tại, hệ thống có một vài vấn đề mở cần được khắc phục và cải thiện, cũng như các chức năng mới cần được triển khai. Trước tiên tất cả các thử nghiệm đơn vị và tích hợp (unit and integration tests) cần phải được hoàn thành. Các bài kiểm tra chấp nhận (acceptance tests) còn thiếu cũng phải được triển khai và hoàn thành sớm nhất có thể.

Các cải thiện khác cho đề tài bao gồm các yếu tố để cải thiện trải nghiệm người dùng, chẳng hạn như lưu ảnh tải cho tất cả các thành phần được tải trong background. Cũng như lưu dữ liệu vào bộ nhớ cache của thiết bị để người dùng vẫn có thể sử dụng ứng dụng khi mạng kém hoặc khi không có mạng.

Ngoài ra, sẽ rất tốt nếu cải thiện hơn nữa hiệu suất, đặc biệt là phân tích lý do khiến ứng dụng hết bộ nhớ khi liệt kê sản phẩm. Và tất nhiên, dự án sẽ tiếp tục tích hợp các tính năng khi SP.IO phát triển, chẳng hạn như các danh mục khác nhau, hỗ trợ đa ngôn ngữ, danh sách mong muốn và bất kỳ chức năng mới nào khác có thể thu hút người dùng trong tương lai.

# TÀI LIỆU THAM KHẢO

[1] Nguyễn Văn Ba, *Phát triển hệ thống hướng đối tượng với UML 2.0 và C++*, NXB Đại học Quốc gia Hà nội, 2015.

[2] A. Dennis B. H. Wixom and David Tegarden, *System Analysis and Design with UML version 2.0: An Object-Oriented Approach*, Second Edition, John Wiley & Sons 2015

[3] Brett Spell, *Pro Java Programming*, Second Edition, Apress 2006 S. Schach, *Object-oriented and classical software engineering*, Sixth Edition, McGrawHill, 2016.

[4] Brett Spell, Pro Java Programming, Second Edition, Apress 2006

**Kết quả Turnitin**

Graphical user interface, text, application

Description automatically generated

1. Theo wikipedia [↑](#footnote-ref-2)
2. Theo báo tuổi trẻ, ngày 15 tháng 3 năm 2023 [↑](#footnote-ref-3)
3. Java for beginer, trang 22 - 23 [↑](#footnote-ref-4)
4. Theo tài liệu hướng dẫn lập trình android, trang 16. [↑](#footnote-ref-5)
5. Theo <https://www.britannica.com/technology/Android-operating-system>, truy cập ngày 16 tháng 3 năm 2023 [↑](#footnote-ref-6)
6. https://firebase.google.com/docs/ [↑](#footnote-ref-7)