TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG VIỆT HÀN

**Khoa Khoa Học Máy Tính**



BÁO CÁO ĐỒ ÁN CƠ SỞ 3

**PHẦN MỀM BÁN ĐIỆN THOẠI VÀ LAPTOP**

Sinh viên: **Bùi Văn Quyền** Mã: 20IT119

**Nguyễn Hà Hải Đăng** Mã:20IT091

Lớp: 20SE3

Giảng viên hướng dẫn: THS.ĐỖ CÔNG ĐỨC

Đà Nẵng, tháng 05 năm 2022

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG VIỆT HÀN

**Khoa Khoa Học Máy Tính**



BÁO CÁO ĐỒ ÁN CƠ SỞ 3

**PHẦN MỀM BÁN ĐIỆN THOẠI VÀ LAPTOP**

Sinh viên: **Bùi Văn Quyền** Mã sinh viên: 20IT119

**Nguyễn Hà Hải Đăng** Mã sinh viên: 20IT091

Lớp: 20SE3

Giảng viên hướng dẫn: THS.ĐỖ CÔNG ĐỨC

Đà Nẵng, tháng 05 năm 2022

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

....................................................................................................................................

Chữ ký của giảng viên hướng dẫn

LỜI CẢM ƠN

Để hoàn thành đồ án này đạt kết quả tốt đẹp, em đã nhận được sự hỗ trợ, giúp đỡ của thầy cô, bạn bè. Với tình cảm sâu sắc, chân thành, cho phép em được bày tỏ lòng biết ơn sâu sắc đến tất cả thầy cô và bạn bè đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu đề tài.

Trước hết em xin gửi tới các thầy cô Trường Đại học CNTT & TT Việt Hàn lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay em đã có thể hoàn thành luận văn, đề tài: " Phần mềm ôn tập trắc nghiệm ".

Đặc biệt tôi xin gửi lời cảm ơn chân thành nhất tới thầy giáo – ThS.Đỗ Công Đức quan tâm giúp đỡ, hướng dẫn em hoàn thành tốt đồ án này trong thời gian qua.

Em xin bày tỏ lòng biết ơn đến lãnh đạo Trường Đại học CNTT & TT Việt Hàn, các Khoa Phòng ban chức năng đã trực tiếp và gián tiếp giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một học viên, đồ án này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để tôi có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Em xin chân thành cảm ơn!

*Sinh viên*

MỤC LỤC

[Chương 1 MỞ ĐẦU 1](#_Toc104794228)

[1.1 Giới thiệu 1](#_Toc104794229)

[*1.2* Mục tiêu của đề tài 1](#_Toc104794230)

[1.3 Nội dung và kế hoạch thực hiện 2](#_Toc104794231)

[1.4 Bố cục báo cáo 2](#_Toc104794232)

[Chương 2 CỞ SỞ LÝ THUYẾT 3](#_Toc104794233)

[2.1 Ngôn ngữ Kotlin 3](#_Toc104794234)

[2.1.1 Giới Thiệu Kotlin 3](#_Toc104794235)

[2.1.2 Kotlin và Android 3](#_Toc104794236)

[2.1.3 Ưu và nhược điểm của Kotlin 4](#_Toc104794237)

[2.1.4 Đặc điểm 7](#_Toc104794238)

[2.2 Firebase Cloud Firestore 10](#_Toc104794239)

[2.2.1 Giới thiệu về Cloud Firesore 10](#_Toc104794240)

[2.2.2 Thêm Firestore vào ứng dụng 11](#_Toc104794241)

[2.2.3 Realtime database và Cloud firestore 11](#_Toc104794242)

[Chương 3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 13](#_Toc104794243)

[3.1 Giới thiệu về phầm mềm hệ thống trắc nghiệm 13](#_Toc104794244)

[3.2 Phân tích hệ thống 13](#_Toc104794245)

[3.2.1 Yêu cầu chức năng 13](#_Toc104794246)

[3.2.2 Yêu cầu phi chức năng 14](#_Toc104794247)

[3.2.3 Biểu đồ ca sử dụng 14](#_Toc104794248)

[3.2.4 Biểu đồ UseCase 16](#_Toc104794249)

[3.2.5 Đặc tả UseCase 18](#_Toc104794250)

[Chương 4 THIẾT KẾ VÀ ĐẶC TẢ GIAO DIỆN 21](#_Toc104794251)

[4.1 Thiết kế cơ sở dữ liệu 21](#_Toc104794252)

[4.2 Cơ sở dữ liệu 21](#_Toc104794253)

[4.2.1 USER 21](#_Toc104794254)

[4.2.2 Sản Phẩm 21](#_Toc104794255)

[4.2.3 Sản Phẩm mới 21](#_Toc104794256)

[4.2.4 Đơn Hàng 22](#_Toc104794257)

[4.2.5 Chi Tiết Đơn Hàng 22](#_Toc104794258)

[4.3 Tổng quan giao diện 22](#_Toc104794259)

[4.4 Thiết kế giao diện 23](#_Toc104794260)

[4.4.1 Giao diện đăng nhập 23](#_Toc104794261)

[4.4.2 Giao diện đăng kí 24](#_Toc104794262)

[4.4.3 Giao diện quên mật khẩu 25](#_Toc104794263)

[4.4.4 Giao diện menu list 26](#_Toc104794264)

[4.4.5 Giao diện trang chủ 27](#_Toc104794265)

[4.4.6 Giao diện trang điện thoại 28](#_Toc104794266)

[4.4.7 Giao diện trang laptop 29](#_Toc104794267)

[4.4.8 Giao diện thêm sửa sản phẩm 30](#_Toc104794268)

[4.4.9 Giao diện giỏ hàng 31](#_Toc104794269)

[4.4.10 Giao diện thanh toán 32](#_Toc104794270)

[4.4.11 Giao diện tìm kiếm 33](#_Toc104794271)

[KẾT LUẬN 34](#_Toc104794272)

[TÀI LIỆU THAM KHẢO 35](#_Toc104794273)

Danh Mục Bảng

[Bảng 3.1 UC đăng kí 18](#_Toc104793904)

[Bảng 3.2 UC hiển thị danh sách sản phẩm 18](#_Toc104793905)

[Bảng 3.3 UC đăng nhập 19](#_Toc104793906)

[Bảng 3.4 UC quản lí sản phẩm 19](#_Toc104793907)

[Bảng 3.5 UC xem đơn hàng đã đặt 20](#_Toc104793908)

[Bảng 4.1 SQL sanpham 21](#_Toc104793909)

[Bảng 4.2 SQL sanphammoi 21](#_Toc104793910)

[Bảng 4.3 SQL donhang 22](#_Toc104793911)

[Bảng 4.4 SQL chitietdonhang 22](#_Toc104793912)

DANH MỤC HÌNH ẢNH

[Hình 3.1 Biều đồ UC tổng quát 16](#_Toc104793889)

[Hình 3.2 Biểu đồ UseCase khách hàng 16](#_Toc104793890)

[Hình 3.3 Use-Case Diagram cho admin 17](#_Toc104793891)

[Hình 3.4 Use-Case Diagram admin quản lý sản phẩm 17](#_Toc104793892)

[Hình 4.1 Giao diện đăng nhập 23](#_Toc104793893)

[Hình 4.2 Giao diện đăng kí 24](#_Toc104793894)

[Hình 4.3 Giao diện quên mật khẩu 25](#_Toc104793895)

[Hình 4.4 Giao diện menu list 26](#_Toc104793896)

[Hình 4.5 Giao diện trang chủ 27](#_Toc104793897)

[Hình 4.6 Giao diện trang điện thoại 28](#_Toc104793898)

[Hình 4.7 Giao diện trang laptop 29](#_Toc104793899)

[Hình 4.8 Giao diện thêm và sửa sản phẩm 30](#_Toc104793900)

[Hình 4.9 Giao diện giỏ hàng 31](#_Toc104793901)

[Hình 4.10 Giao diện thanh toán 32](#_Toc104793902)

[Hình 4.11 Giao diện tìm kiếm 33](file:///C:\Users\admin\Downloads\LTDD\Bùi%20Văn%20Quyền_20IT119.docx#_Toc104793903)

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **VIẾT TẮT** | **NỘI DUNG** |
| UC | UseCase |
| SQL | Structured Query Language |
| TMĐT | Thương mại điện tử |

# MỞ ĐẦU

## Giới thiệu

Thương mại điện tử (TMĐT) là lĩnh vực tương đối mới ở Việt Nam rất được Chính phủ quan tâm thúc đẩy phát triển. Cho đến nay đã có rất nhiều lý do để các doanh nghiệp cũng như các tổ chức phải quan tâm đến ứng dụng TMĐT vào kinh doanh để cạnh tranh, tồn tại và phát triển trong thực trạng toàn cầu hóa hiện nay. Tuy nhiên, hiện ở Việt Nam chưa có nhiều doanh nghiệp hay tổ chức ứng dụng TMĐT một cách đúng nghĩa vào quá trình trao đổi mua bán ở trong nước và quốc tế. Hiện tại các phương tiện kỹ thuật và các công cụ lập trình ứng dụng vào TMĐT chưa được ứng dụng rộng rãi trong các doanh nghiệp, để có đủ kiến thức cơ bản về TMĐT xây dựng và vận hành một app bán hàng có hiệu quả, thực hiện marketing qua mạng, chọn lựa mô hình TMĐT phù hợp, xúc tiến thanh toán qua mạng, áp dụng các biện pháp an toàn mạng tối thiểu cần thiết v.v… Từ những lý do trên, chúng em quyết định chọn đề tài “Thiết kế app bán điện thoại di động” - xây dựng thử nghiệm cửa hàng điện thoại di động ảo theo mô hình thương mại điện tử B2C. Thông qua đó để chúng em hiểu biết, nắm rõ hơn về lĩnh vực TMĐT và các kỹ năng ứng dụng lập trình cho app thương mại điện tử một cách hiệu quả.

## Mục tiêu của đề tài

Nhận xét bài toán Bài toán thương mại điện tử là một bài toán lớn và phức tạp với nhiều module khách nhau như: tìm kiếm sản phẩm, đưa hàng vào giỏ hàng, đăng ký thành viên, bảo mật, các chức năng quản lý… Những khó khăn hiện tại của cửa hàng: cùng với sự xuất hiện đa dạng của các hãng điện thoại di động khác nhau và yêu cầu về mở rộng thị trường cũng như việc tiếp thị sản phẩm nhanh nhất đến khách hàng, cửa hàng phải đối mặt với những vấn đề khó giải quyết do cơ cấu cũ không đáp ứng được như:

Khách hàng phải tốn thời gian trực tiếp đến các cửa hàng bán lẻ để xem và mua sản phẩm.

Khả năng tiếp thị sản phẩm tới khách hàng thấp, không thu hút được nhiều khách hàng cho cửa hàng.

Việc quản lý tồn kho không chặt chẽ và thiếu linh động giữa công ty và dẫn đến khó khăn trong khâu quản lý và tình trạng thiếu thông tin cho khách hàng.

* Từ đó suy ra mục tiêu của đề tài là:
* Xây dựng được 1 app có đầy đủ các chức năng như: tìm kiếm, thêm vào giỏ hàng, thanh toán,..
* Có thể tạo tài khoản và đăng nhập vào app để mua hàng….

## Nội dung và kế hoạch thực hiện

* Nghiên cứu cấu trúc hệ thống bán hàng
* Nghiên cứu các sản phẩm
* Nghiên cứu bộ cách thanh toán giỏ hàng
* Nghiên cứu các công cụ hỗ trợ tiện ích cho ứng dụng
* Nghiên cứu bố cục bố trí các mục trong ứng dụng

## Bố cục báo cáo

Sau phần *Mở đầu*, báo cáo được trình bày trong ba chương, cụ thể như sau:

Chương 1. *Cơ sở lý thuyết*

Chương 2. *Phân tích và xây dựng hệ thống*

Chương 3. *Thiết kế và đặc tả giao diện*

Cuối cùng là *Kết luận*, *Tài liệu tham khảo* và *Phụ lục* liên quan đến đề tài.

# CỞ SỞ LÝ THUYẾT

## Ngôn ngữ Kotlin

### Giới Thiệu Kotlin

Kotlin là một ngôn ngữ ngữ dụng kiểu tĩnh dành cho Java Virtual Machine đã chính thức phát hành phiên bản 1.0. Nó được tạo ra bởi JetBrains, Kotlin cũng giống như nhiều ngôn ngữ lập trình không phải Java khác, tức là cũng sẽ chạy trên JVM và sử dụng các công cụ và thư viện hiện có của Java. Và ngược lại Java cũng có thể sử dụng các item được xây dựng trong Kotlin.

Theo website JetBrains, mục tiêu quan trọng của Kotlin là tính hữu dụng. Từ lúc giới thiệu vào năm 2011 cho đến khi phát hành phiên bản 1.0, JetBrains đã luôn chú trọng đến tính tương hợp với Java..Mặc dù Kotlin có một số tính năng hoàn toàn mới chẳng hạn như một type system được thiết kế để ngăn chặn các bug như các null pointer reference, nhưng quan trọng là nó làm việc cùng với code và cơ sở hạ tầng hiện có của Java. Cuối cùng, Kotlin không có trình quản lý gói và build system của riêng nó, do Java đã có sẵn.

### Kotlin và Android

Lập trình Android là một trong những lĩnh vực quan trọng mà JetBrains hướng đến với Kotlin. Ngôn ngữ mang đến tính tương thích ngược với Java 6 và 7, các phiên bản của Java hầu hết đều tương thích chặt chẽ với Android. JetBrains cũng hy vọng Kotlin sẽ được sử dụng trong các lĩnh vực khác chẳng hạn như các ứng dụng lớn và phức tạp, đề cao hiệu suất.

Các nhà phát triển không có lựa chọn thay thế cho việc phát triển ứng dụng Android bằng ngôn ngữ Java. Mặc dù được sử dụng rộng rãi, nhưng trong quá trình hoạt động ngôn ngữ Java sinh ra rất nhiều file rác. Java 8 đã giải quyết một số vấn đề ngôn ngữ và đặc biệt hơn là với Java 10. Để có được nhiều lợi ích từ việc chỉnh sửa trong hai phiên bản này, bạn phải đặt SDK tối thiểu sang Android 24 chỉ để sử dụng Java 8. Kotlin nhắm đến việc lấp đầy khoảng trống đó của một ngôn ngữ hiện đại đang thiếu cho nền tảng Android. Có một vài nguyên lý cốt lõi mà giúp Kotlin tồn tại đó là:

*Nhỏ gọn để giảm số lượng mã boilerplate cần viết. Mã code dễ đọc và dễ hiểu hơn. An toàn, tránh toàn bộ các lớp lỗi như các null pointer exceptions. Linh hoạt cho việc xây dựng ứng dụng phía máy chủ, ứng dụng Android hoặc mã lối vào đang chạy trong trình duyệt. Khả năng tương tác để tận dụng các khuôn khổ và thư viện hiện có của JVM với khả năng tương tác Java 100%.*

### Ưu và nhược điểm của Kotlin

#### Ưu điểm

Dù không thiếu các ngôn ngữ biên dịch sang Java bytecode, nhưng có một vài yếu tố làm cho Kotlin nổi bật hơn cả:

* Có thể thay thế cho Java

Một trong những thế mạnh lớn nhất của Kotlin như là một ứng viên để thay thế cho Java là khả năng tương tác rất tốt giữa Java và Kotlin, bạn có thể thậm chí có code Java và Kotlin tồn tại song song trong cùng dự án, và tất cả mọi thứ vẫn sẽ được biên dịch một cách hoàn hảo. Dưới đây, bạn có thể thấy một ví dụ về một dự án bao gồm một Activity Java và một Actitivy Kotlin.

Trong thực tế, một khi dự án kết hợp Kotlin và Java được biên dịch, người dùng sẽ không thể biết những phần nào của dự án của bạn được viết bằng Java, và những phần nào được viết bằng Kotlin. Bởi vì các lớp Kotlin và Java có thể tồn tại song song trong cùng một dự án, nên có thể bắt đầu sử dụng Kotlin mà không cần phải làm bất cứ điều gì to tát cả giống như chuyển đổi toàn bộ dự án sang Kotlin hoặc bắt đầu một dự án mới để bạn có thể thử Kotlin.

Vì Kotlin là hoàn toàn tương thích với Java, nên cũng có thể sử dụng phần lớn các thư viện Java và các framework trong dự án Kotlin của bạn, thậm chí nâng cao các framework dựa vào chú thích xử lý.

* Dễ học

Kotlin nhằm mục đích là nâng cao hơn so vớiJava, chứ không phải hoàn toàn viết lại, rất nhiều các kỹ năng đã có trong việc code Java của bạn vẫn được áp dụng đối với dự án Kotlin.

Kotlin cũng được thiết kế để có thể dễ học cho các nhà phát triển Java. Những nhà phát triển Java sẽ cảm thấy rằng hầu hết cú pháp của Kotlin đều quen thuộc; ví dụ, các code được sử dụng để tạo ra một lớp mới trong Kotlin là rất giống với Java: class MainActivity : AppCompatActivity() {

Kotlin cũng được thiết kế trực quan và dễ đọc, do đó, ngay cả khi bạn gặp một số code khác biệt, thì bạn vẫn có thể để biết được ý nghĩa về những gì code này làm.

* Kết hợp những gì tốt nhất của lập trình hàm và thủ tục

Hiện đang có một số lý thuyết lập trình được sử dụng rộng rãi, nhưng khi nói đến các câu hỏi về "phương pháp nào là tốt nhất", thì không dễ để có câu trả lời. Mỗi kiểu lập trình có tập hợp điểm mạnh và điểm yếu của nó, do đó, mặc dù không thiếu các kịch bản mà lập trình hàm có một lợi thế, thì cũng có rất nhiều vấn đề nơi mà một cách tiếp cận lập trình thủ tục sẽ hiệu quả hơn.

Vậy tại sao bạn cần phải lựa chọn giữa hàm và thủ tục? Giống như nhiều ngôn ngữ lập trình hiện đại khác, Kotlin nhằm mục đích mang lại cho bạn những gì tốt nhất của cả hai bằng cách kết hợp các khái niệm và các yếu tố của lập trình thủ tục và hàm.

* Android Studio hỗ trợ

Kotlin được phát triển bởi JetBrains, công ty đứng sau IntelliJ IDE mà Android Studio dựa trên nó. Không có gì bất ngờ, rằng Android Studio hỗ trợ tốt cho Kotlin. Một khi bạn đã cài đặt plugin Kotlin, Android Studio làm cho việc cấu hình Kotlin trong dự án của bạn trở nên đơn giản giống như mở một vài menu.

Một khi bạn đã thiết lập plugin Kotlin cho Android Studio, IDE của bạn sẽ không gặp vấn đề gì việc hiểu, biên dịch và chạy code Kotlin. Android Studio cũng cung cấp việc gỡ lỗi, tự động hoàn tác, điều hướng code, unit testing, và tái cấu trúc cho Kotlin.

Một khi dự án Android Studio của bạn đã được cấu hình để hỗ trợ cho Kotlin, bạn thậm chí có thể chuyển đổi toàn bộ một tập tin mã nguồn Java thành một tập tin Kotlin, với chỉ một vài cú nhấp chuột.

* Code ngắn gọn hơn

Nếu bạn so sánh một lớp Java và một lớp Kotlin cho ra cùng một kết quả, thì cái được viết trong Kotlin thường sẽ gọn gàng hơn nhiều và nhỏ gọn hơn so với những gì được viết bằng Java. Và như mọi nhà phát triển đều biết, code ít hơn có nghĩa là ít lỗi hơn!

Kotlin đặc biệt tốt trong việc giảm số lượng code mà bạn cần phải viết, làm cho việc viết code trong Kotlin trở thành một trải nghiệm thú vị hơn nhiều, so với việc viết code trong nhiều ngôn ngữ khác như Java.

Đặc biệt, các extension của Kotlin Android (mà chúng ta sẽ khám phá trong phần hai) cho phép bạn nhập tham chiếu đến một View vào một tập tin Activity, từ đó, bạn có thể làm việc với giao diện như thể nó là một phần của Activity đó. Điều này có nghĩa là bạn không còn phải xác định mỗi View bằng cách sử dụng findViewById, mà có thể chuyển đổi code chẳng hạn như:

TextView text = (TextView) findViewById(R.id.myTextView); text.setText("Hello World"); Thành gọn gàng hơn nhiều: myTextView.setText("Hello World")

#### Nhược điểm

Không có ngôn ngữ lập trình nào là hoàn hảo cả, do đó, mặc dù Kotlin có rất nhiều thứ để cung cấp cho các nhà phát triển Android, nó vẫn có một số nhược điểm mà bạn cần lưu ý:

* Thêm thời gian chạy Runtime

Thư viện tiêu chuẩn của Kotlin và runtime sẽ làm tăng kích thước tập tin .apk của bạn. Mặc dù nó chỉ tương đương với khoảng 800KB, nhưng nếu ứng dụng của bạn đã lớn sẵn rồi thì 800KB phụ có thể làm nó phình to và khiến người dùng nghĩ lại trước khi tải về ứng dụng của bạn.

* Không hẳn là code dễ đọc đối với beginer

Mặc dù cú pháp ngắn gọn của Kotlin là một trong những thế mạnh lớn nhất của ngôn ngữ, nhưng bạn có thể thấy một số khó khăn ban đầu, đơn giản bởi vì có rất nhiều thứ đang được thực hiện trong một số lượng nhỏ code đó. Java có thể dài dòng hơn, nhưng ngược lại tất cả mọi thứ đều rõ ràng, có nghĩa là những người không quen code Java có xu hướng dễ dàng hơn để hiểu so Kotlin.

Ngoài ra, nếu sử dụng không hợp lý, quá tải toán tử của Kotlin có thể dẫn đến số code đó có thể khó khăn để đọc.

* Thiếu hỗ trợ chính thức

Kotlin có thể được trợ tốt trong Android Studio, nhưng cần ghi nhớ rằng Kotlin không được xác nhận chính thức là của Google.

Ngoài ra, tính năng tự động hoàn tác và biên dịch trong Android Studio có xu hướng chạy hơi chậm khi bạn làm việc với Kotlin so với một dự án thuần Java.

* Cộng đồng nhỏ hơn và ít có sẵn trợ giúp

Vì Kotlin là một ngôn ngữ tương đối mới, nên cộng đồng Kotlin vẫn còn khá nhỏ, đặc biệt là so với cộng đồng của ngôn ngữ khác như Java. Nếu bạn chuyển đổi sang Kotlin, thì bạn có thể không có được truy cập vào cùng một số hướng dẫn, bài đăng trên blog, và tài liệu hướng dẫn sử dụng, và có thể gặp ít hỗ trợ từ cộng đồng về những nơi như các diễn đàn và Stack Overflow. Tại thời điểm bài viết, tìm kiếm cho Kotlin trong Stack Overflow trả về các bài viết chỉ hơn 4.600 được dán nhãn Kotlin so với hơn 1.000.000 bài viết có chứa các từ khóa Java.

### Đặc điểm

1.4.1. Khai báo biến

* Có 2 từ khóa khai báo biến trong Kotlin là var và val

val a: Int = 1 val b = 1 // `Int` type is inferred val c: Int // Type required when no initializer is provided c = 1 // definite assignment var x = 5 // `Int` type is inferred x += 1

* Từ khóa var sử dụng khi giá trị của biến thay đổi, val sử dụng khi giá trị của biến không thay đổi
* Từ khóa val giống như readonly trong C# hoặc final trong Java
* Biến val phải được khởi tạo lúc khai báo
* Từ khóa Void trong java hay C# sẽ được thay thế bằng Unit trong Kotlin

1.4.2. Null safety

Một trong những cạm bẫy phổ biến trong các ngôn ngữ lập trình, bao gồm cả Java là cho phép 1 thành phần nào đó được Null. Nếu không chắc chắn thành phần đó được phép Null hay không sẽ rất dễ xảy ra lỗi không lường trước được, gây nguy hiểm cho hệ thống của bạn. Cụ thể, trong java sẽ gây ra 1 exception là NullPoiterException hoặc viết ngắn gọi là NPE.

Kotlin nhằm mục đích xóa bỏ NullPoiterException trong code của chúng ta. Ngay khi khai báo biến với Kotlin, bạn đã phải chỉ rõ biến đó có được phép Null hay không. Có 2 trường hợp được phép đó là : không thể Null và có thể Null

Khai báo 1 biến cho phép Null

var b: String? = "abc" // có thêm dấu ? sau kiểu của biến b = null // compilation ok val l = b.length // not safe

* Khai báo 1 biến không được phép Null

var a: String = "abc" a = null // compilation error val l = a.length // safe

Chúng ta có thể thấy Kotlin đã khắc phục được expcetion NPE trong Java. Các biến cho phép Null hay không được phép Null đã được xác định ngay trong quá trình khai báo biến, các IDE sẽ giúp chúng ta phát hiện ra lỗi ngay khi compile. Việc phát hiện ra lỗi sớm khi compile sẽ tốt hơn so với khi runtime. Nó giúp hệ thống của chúng ta an toàn hơn.

#### Kế thừa Override

* Mặc định class trong Kotlin đều là final (tức là không được phép kế thừa)

|  |
| --- |
| open class Base { open fun v() {} fun nv() {}  } class Derived() : Base() { override fun v() {}  } |

Trong class final (tức là không có từ khóa “open”) thì việc khai báo các hàm, thuộc tính open bị cấm

* Ở 1 lớp con khác, hàm được đánh dấu là override thì chính nó là open (lại được kế thừa từ lớp khác), nếu muốn chống override thì lại sử dụng từ khóa final

open class AnotherDerived() : Base() { final override fun v() {}

}

* Trong interface, các member mặc định là “open”

interface B { fun f() { print("B") } // interface members are 'open' by default fun b() { print("b") }

}

* Kotlin implement interface :

|  |
| --- |
| class C() : A(), B {  // The compiler requires f() to be overridden:  override fun f() { super<A>.f() // call to A.f() super<B>.f() // call to B.f()  }  } |

#### Chỉ định truy cập

* Có 4 loại Visivility Modifiers trong Kotlin : private, protected, internal, và public
* Nếu không chỉ rõ thì mặc định là public
* Các chỉ định truy cập : private, protected và public giống cách sử dụng trọng java **-** Riêng internal ta có thể sử dụng trong cùng module

#### String template

String trong kotlin có thể chứa các biểu thức template, tức là những kết quả trả về hoặc biến có thể được nối vào trong 1 String. Một biểu thức template bắt đầu với ($) và tên :

|  |
| --- |
| val apples = 4 valbananas = 3 println(“I have $apples apples.”) println(“I have $apples apples and ” + (apples + bananas) + “ fruits.”)  println(“I have $apples apples and ${apples + bananas} “ fruits.”) |

## Firebase Cloud Firestore

### Giới thiệu về Cloud Firesore

Cloud Firestore là một Database linh hoạt và dễ mở rộng cho mobile, web và server được phát triển từ Firebase and Google Cloud Platform. Cũng giống như Firebase realtime database Cloud Firestore giúp cho việc đồng bộ dữ liệu giữa các ứng dụng phía client một các nhanh chóng (Realtime) và hộ trợ lưu offline data trong ứng dụng của bạn.

Cloud Firestore là một cloud-hosted, NoSQL database mà các ứng dụng phía client có thể trực tiếp truy cập thông qua native SDKs. Nó lưu dữ liệu theo mô hình dữ liệu NoSQL. Dữ liệu được lưu trữ trong các file tài liệu chứa các trường được ánh xạ vào các giá trị. Các file tài liệu này được lưu trữ trong các tập hợp chúng có thể sử dụng nó để tổ chức dữ liệu và truy vấn dữ liệu. Cloud Firestore hỗ trợ rất nhiều kiểu dữ liệu từ đơn giản như String, Integer hay những kiểu dữ liệu phức tạp như các nested object.

### Thêm Firestore vào ứng dụng

#### Tạo một cloud firestore project

* Để sử dụng Cloud Firestore bạn cần tạo một project từ tài khoản firebase<https://console.firebase.google.com/u/0/>
* Trong menu trái bạn chọn Database và chọn Try Firestore Beta, cuối cùng là click Enable.

#### Cài đặt môi trường

* Để sử dụng dịch các thử viện firebase bạn cần thêm firebase vào project: <https://firebase.google.com/docs/android/setup> **-** Thêm Cloud Firestore vào app/build.gradle

compile 'com.google.firebase:firebase-firestore:11.8.0'

#### Tạo một instance của Cloud Fristore

// Access a Cloud Firestore instance from your Activity

FirebaseFirestore db = FirebaseFirestore.getInstance();

#### Một số thao tác với dữ liệu với Cloud Firestore

Thêm dữ liệu Cloud Firestore lưu data trong Documents, Các Documents được lưu trữ trong các Collections. Khi bạn thêm dữ liệu vào trong một document lần đầu tiên thì Cloud Firestore sẽ tự động tạo các collections và documents bạn sẽ không cần tạo collections và documents một cách tường minh nữa.

### Realtime database và Cloud firestore

Cloud Firestore có thể được coi là phiên bản cải tiến của Realtime database, nó được cải tiến nhiều tính năng mới và tăng tốc độ truy vấn dữ liệu. Nếu bạn đang sử dụng Realtime Database thì việc chuyển sang sử dụng Cloud firestore khá dễ dàng. Dưới đây mình sẽ nêu một số những ưu điểm của Cloud Firestore so với Realtime database:

* Việc tổ chức lưu trữ dữ liệu phức tạp dễ dàng hơn so với Realtime database .
* Có thể kết hợp việc filter, sort, ... trong một query, điều này không thể làm được với Realtime database.
* Có thể truy vấn một collection hoặc một document mà không cần phải thông qua việc truy vấn toàn bộ collections giống như realtime database.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Giới thiệu về phầm mềm hệ thống trắc nghiệm

Đây là hệ thống phần mềm hỗ trợ áp dụng thương mại điện tử vào lĩnh vực kinh doanh bắt đầu bằng việc giới thiệu sản phẩm cho khách hàng qua app sau đó cho khách hàng đặt hàng, do việc thanh toán qua mạng dễ dàng và an toàn hơn nên chúng em cho ra đời app bán điện thoại online.

## Phân tích hệ thống

### Yêu cầu chức năng

Sau khi hoàn thiện phần mềm đáp ứng đầy đủ các chức năng cơ bản của ứng dụng bán điện thoại:

* Đối với người quản trị:
* Điều hành mọi hoạt động của app.
* Thêm xóa, sửa sản phẩm trong catalog chính của cửa hàng.
* Tiếp nhận đơn đặt hàng của khách hàng.
* Xử lý đơn đặt hàng.
* Thống kê số lượng danh mục, mặt hàng, tin tức và đơn đặt hàng.
* Đối với khách hàng:
* Tham quan cửa hàng.
* Đăng ký thành viên.
* Xem, lựa chọn và tìm kiếm các sản phẩm .
* Có thể đặt hàng qua mạng.
* Xử lý:
* Đặt hàng tự động.
* Tìm kiếm thông tin ngay trên app.
* Không hỗ trợ forum.
* Giao diện: Hệ thống hoạt động trên giao diện điện thoại application.
* Phạm vi:
* Dữ liệu cập nhật hàng ngày.
* Đáp ứng lưu trữ với khối lượng lớn sản phẩm.
* Chứa hình ảnh ở mức trung bình.
* Về mạng: Hệ thống hỗ trợ hoạt động trên Internet (bán hàng qua mạng)

### Yêu cầu phi chức năng

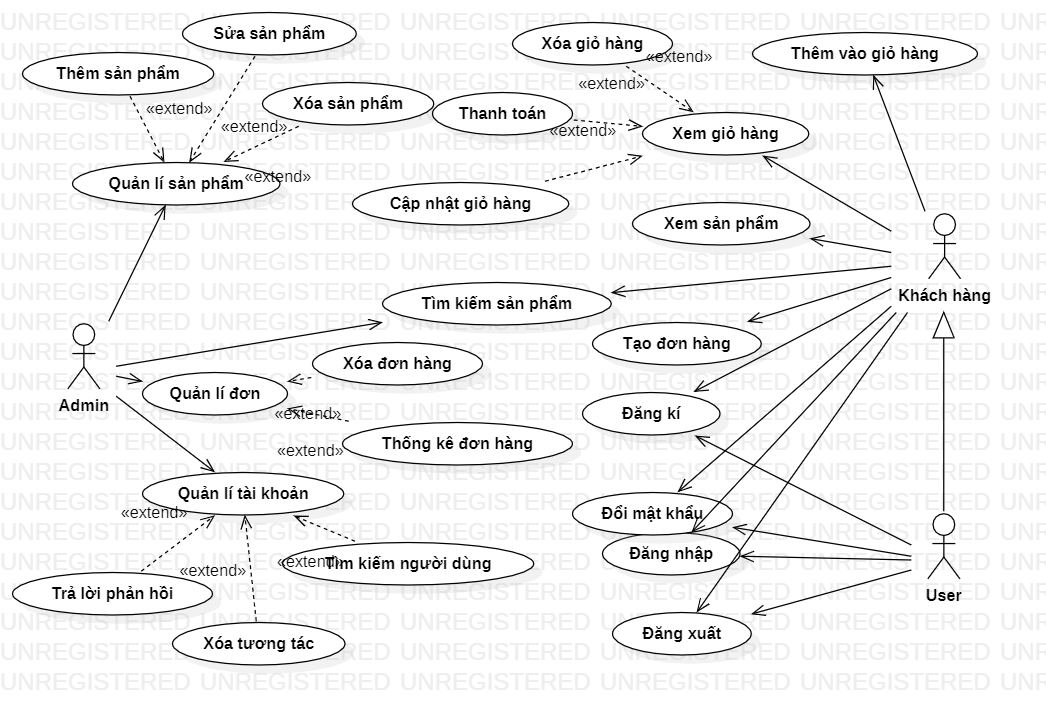
* Giao diện hệ thống dễ dùng, dễ nhìn
* Hệ thống có thể chạy trên các thiết bị và hệ điều hành có kết nối internet
* Hệ thống hoạt động 24/7, 24/24 kể cả ngày lễ
* Chuyển trang mượt, thời gian tương tác giữa hệ thống và người dùng phải nhỏ hơn 3s
* Bảo mật thông tin người dùng

### Biểu đồ ca sử dụng

* Các tác nhân: Admin, người dùng.
* Các ca sử dụng:
* Chức năng đối với khách hàng:
* Chức năng đăng ký tài khoản khách hàng
* Chức năng đăng nhập hệ thống
* Chức năng thoát khỏi hệ thống
* Chức năng tìm kiếm sản phẩm
* Chức năng thêm hàng vào giỏ hàng
* Chức năng xem giỏ hàng:
* Chức năng xóa sản phẩm trong giỏ hàng
* Chức năng cập nhật giỏ hàng
* Chức năng tạo đơn hàng
* Chức năng xem sản phẩm
* Chức năng xem chi tiết sản phẩm
* Chức năng xem tin tức
* Chức năng đối với người quản trị hệ thống:
* Chức năng đăng nhập hệ thống
* Chức năng thoát khỏi hệ thống
* Chức năng quản lý danh mục sản phẩm:
* Thêm danh mục
* Xoá danh mục
* Cập nhật danh mục
* Chức năng quản lý sản phẩm:
* Thêm sản phẩm
* Xoá sản phẩm
* Cập nhật sản phẩm
* Tìm kiếm sản phẩm
* Chức năng quản lý đơn đặt hàng:
* Duyệt đơn đặt hàng
* Xoá đơn đặt hàng
* Xem chi tiết đơn đặt hàng
* Tìm kiếm đơn đặt hàng
* Chức năng quản lý tin tức:
* Tạo mới tin tức
* Xoá tin tức
* Cập nhật tin tức
* Chức năng xem sản phẩm
* Chức năng quản lý nhà cung cấp:
* Thêm nhà cung cấp
* Xoá nhà cung cấp
* Cập nhật nhà cung cấp
* Chức năng xem chi tiết sản phẩm
* Chức năng xem tin tức

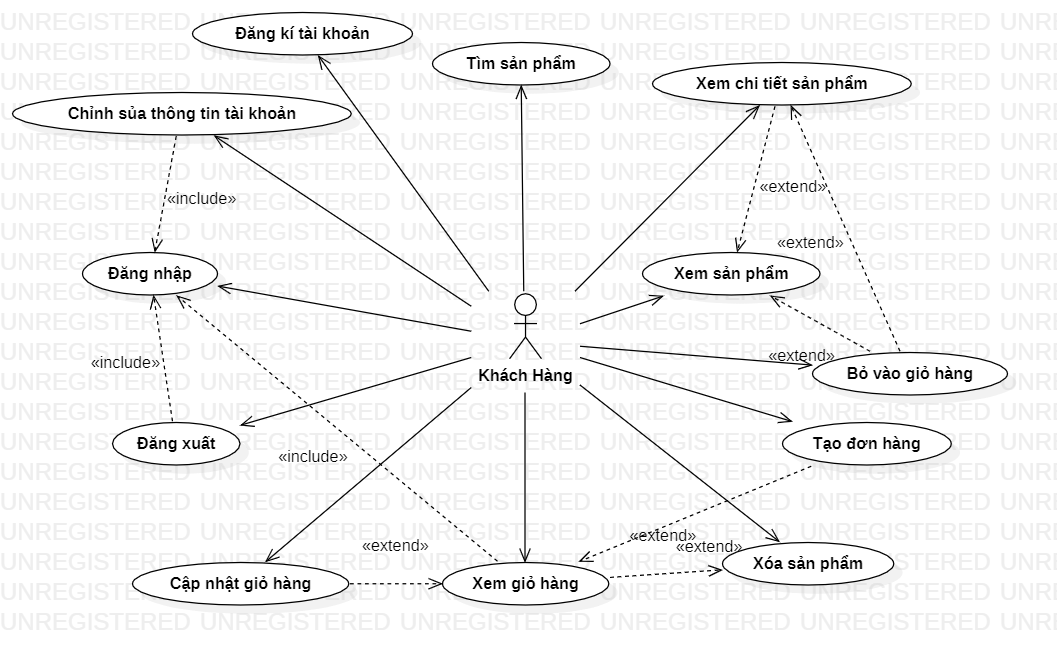
### Biểu đồ UseCase

#### Biểu đồ UseCase tổng quát



Hình 3.1 Biều đồ UC tổng quát

#### Biểu đồ UseCase khách hàng



Hình 3.2 Biểu đồ UseCase khách hàng

#### Biểu đồ UseCase admin



Hình 3.3 Use-Case Diagram cho admin

#### Biểu đồ UseCase admin quản lý sản phẩm



Hình 3.4 Use-Case Diagram admin quản lý sản phẩm

### Đặc tả UseCase

#### UC đăng ký

Bảng 3.1 UC đăng kí

|  |  |  |
| --- | --- | --- |
| Tác nhân | - | Người dùng |
| Mô tả | - | Cho phép người dùng đăng ký để trở thành thành viên của hệ thống |
| Tiền điều kiện | - | Người dùng vào hệ thống |
| Luồng sự kiện chính | -  - | Người dùng chọn mục đăng ký thành viên Form đăng ký thành viên xuất hiện |
|  | - | Người dùng xem và điền thông tin vào form đăng ký của hệ thống |
|  | - | Nhấn nút đăng ký |
|  | - | Thông tin khách hàng đầy đủ, không lặp thông tin của người dùng trước, chính xác thì hệ thống sẽ xác nhận tài khoản của khách hàng. |
|  | - | UC kết thúc |
| Hậu điều kiện | - | Người dùng sẽ trở thành thành viên của hệ thống |

#### UC hiển thị danh sách sản phẩm

Bảng 3.2 UC hiển thị danh sách sản phẩm

|  |  |  |
| --- | --- | --- |
| Tác nhân | **-** | Admin, người dùng |
| Mô tả | **-** | UC cho phép người dùng và admin xem danh sách thông tin các sản phẩm có trên hệ thống |
| Tiền điều kiện |  |  |
| Luồng sự kiện chính | **-**  **-** | Admin, người dùng chọn sản phẩm  Form hiển thị danh sách sẽ xuất hiện. hệ thống sẽ hiển thị các sản phẩm |
|  | **-** | Admin, người dùng xem thông tin chi tiết về tất cả sản phẩm |
|  | **-** | UC kết thúc |

#### UC đăng nhập

Bảng 3.3 UC đăng nhập

|  |  |  |
| --- | --- | --- |
| Tác nhân | **-** | Admin, người dùng |
| Mô tả | **-** | UC cho phép người dùng và admin đăng nhập vào hệ thống |
| Tiền điều kiện |  |  |
| Luồng sự kiện chính | **-**  **-** | Admin, người dùng form đăng nhập  Form hiển thị đăng nhập xuất hiện người dùng và admin nhập thông tin yêu cầu và xác nhận đăng nhập. |
|  | **-** | UC kết thúc |

#### UC quản lí sản phẩm , đơn hàng

Bảng 3.4 UC quản lí sản phẩm

|  |  |  |
| --- | --- | --- |
| Tác nhân | - | Admin. |
| Mô tả | - | UC cho phép admin sau khi đăng nhập có thể quản lý (thêm, sửa, xóa) sản phẩm. |
| Tiền điều kiện | - | Người quản trị phải đăng nhập vào hệ thống admin. |
| Luồng sự kiện chính | - | Chức năng này bắt đầu khi người quản trị lựa chọn quản lý sản phẩm. |
|  | - | Người quản lý chọn đối tượng cụ thể. |
|  | - | Người quản trị thực hiện các thao tác thêm mới hay sửa,xóa và có lưu hoặc hủy thay đổi |
|  | - | Hệ thống kiểm tra tính hợp lệ của dữ liệu và thực hiện lưu thay đổi. |
| Luồng sự các phụ | - | Nếu các sự kiện người quản trị cung cấp những dữ liệu không hợp lệ(thêm, sửa) hệ thống sẽ xuất hiện thông báo nhắc nhở. |
|  | - | Người quản trị có thể tiếp tục thực hiện các thao tác theo dòng sự kiện chính hoặc hủy bỏ chức năng này. |
|  | - | Các thay đổi về sản phẩm được thực thi. |

#### UC xem đơn hàng

Bảng 3.5 UC xem đơn hàng đã đặt

|  |  |  |
| --- | --- | --- |
| Tác nhân | - | Người dùng. |
| Mô tả | - | UC cho phép người dùng xem đơn hàng của mình. |
| Tiền điều kiện | - | Người dùng đăng nhập vào hệ thống. |
| Luồng sự kiện chính | -  - | Người dùng chọn sản phẩm.  Người dùng hoàn thành đơn đặt hàng hệ thống sẽ gửi thông tin đơn hàng về đơn hàng của người dùng. |
|  | - | UC kết thúc. |

# THIẾT KẾ VÀ ĐẶC TẢ GIAO DIỆN

## Thiết kế cơ sở dữ liệu

## Cơ sở dữ liệu

### USER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | Tên | Kiểu | Bảng mã đối chiếu | Null | Mặc định | Ghi chú | Thêm |
| 1 | id | *Int(11)* |  | *Không* | *Không* |  | *AUTO\_INCREMENT* |
| 2 | email | *Varchar(250)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 3 | pass | *Varchar(250)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 4 | Username | *Varchar(100)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 5 | Mobile | *Varchar(15)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |

#### Bảng 3.1: SQL USER

### Sản Phẩm

Bảng 4.1 SQL sanpham

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | Tên | Kiểu | Bảng mã đối chiếu | Null | Mặc định | Ghi chú | Thêm |
| 1 | id | *Int(11)* |  | *Không* | *Không* |  | *AUTO* |
| 2 | tensanpham | *Varchar(1000)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 3 | hinhanh | *text* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |

### Sản Phẩm mới

Bảng 4.2 SQL sanphammoi

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | Tên | Kiểu | Bảng mã đối chiếu | Null | Mặc định | Ghi chú | Thêm |
| 1 | id | *Int(10)* |  | *Không* | *Không* |  | *AUTO* |
| 2 | tensp | *Varchar(250)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 3 | giasp | *Varchar(250)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 4 | hinhanh | *text* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 5 | mota | *text* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 6 | loai | *Int(2)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |

### Đơn Hàng

Bảng 4.3 SQL donhang

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | Tên | Kiểu | Bảng mã đối chiếu | Null | Mặc định | Ghi chú | Thêm |
| 1 | id | *int(11)* |  | *Không* | *Không* |  | *AUTO* |
| 2 | iduser | *int(11)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 3 | email | *Int(10)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 4 | diachi | *text* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 5 | sodienthoai | *Varchar(250)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |
| 6 | soluong | *int(11)* |  | *Không* | *Không* |  |  |
| 7 | tongtien | *Varchar(250)* | *utf8mb4\_general\_ci* | *Không* | *Không* |  |  |

### Chi Tiết Đơn Hàng

Bảng 4.4 SQL chitietdonhang

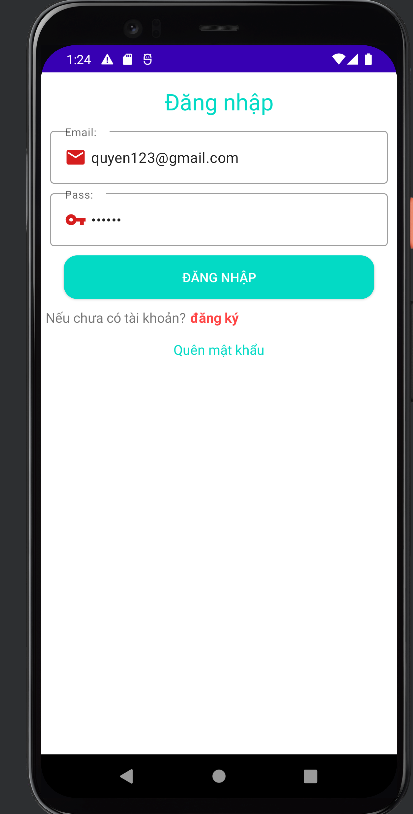
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| # | Tên | Kiểu | Bảng mã đối chiếu | Null | Mặc định | Ghi chú | Thêm |
| 1 | Iddonhang | *int(11)* |  | *Không* | *Không* |  |  |
| 2 | Idsanpham | *int(11)* |  | *Không* | *Không* |  |  |
| 3 | soluong | *int(11))* |  | *Không* | *Không* |  |  |
| 4 | Gia | *Varchar(250)* | *utf8\_unicode\_ci* | *Không* | *Không* |  |  |

## Tổng quan giao diện

Giao diện được thiết kế phù hợp với nhu cầu sử dụng của người dùng. các chức năng được hiển thị ở menu giúp người dùng có thể thao tác nhanh chóng giữa các trang. Nội dung sản phẩm được chia làm 2 trang là điện thoại và laptop. App có trang xem lại đơn hàng đã đặt của khách hàng trước đó.

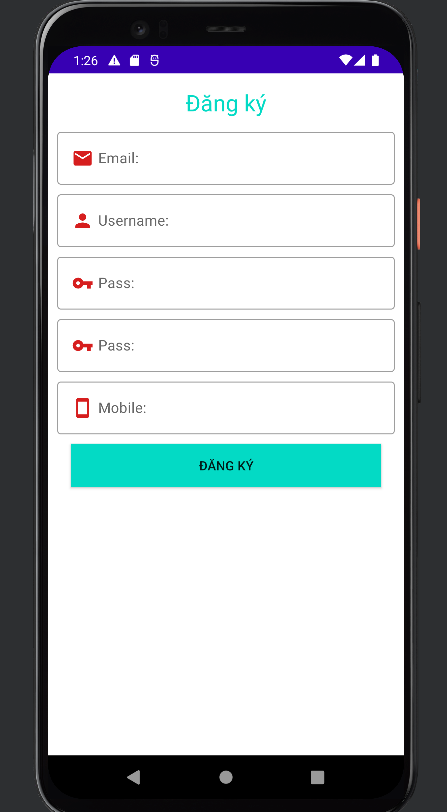
## Thiết kế giao diện

### Giao diện đăng nhập



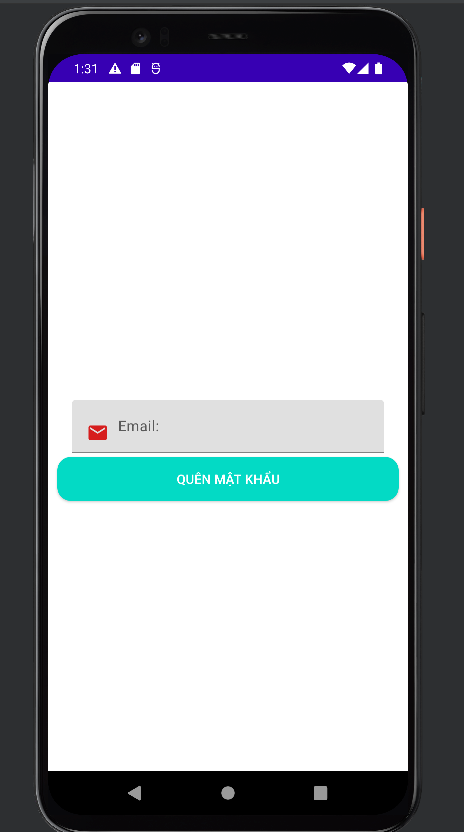
Hình 4.1 Giao diện đăng nhập

### Giao diện đăng kí



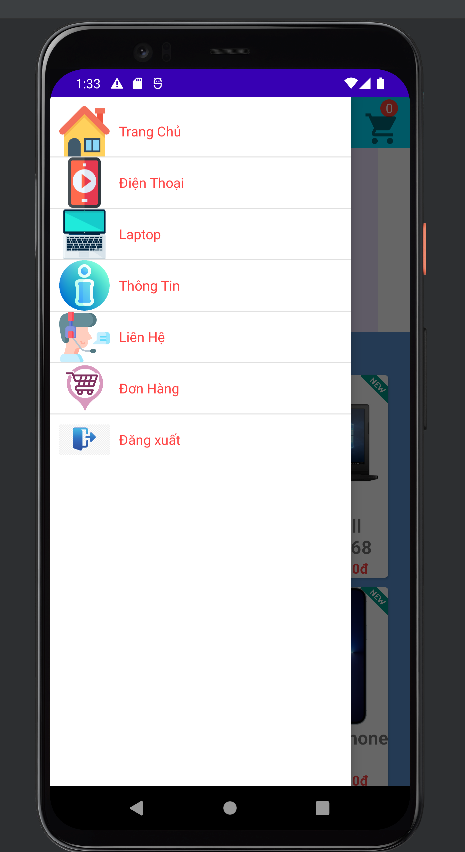
Hình 4.2 Giao diện đăng kí

### Giao diện quên mật khẩu



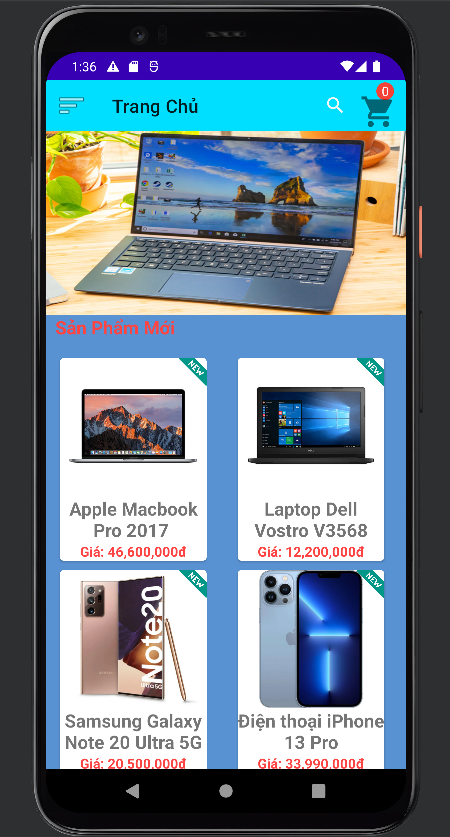
Hình 4.3 Giao diện quên mật khẩu

### Giao diện menu list



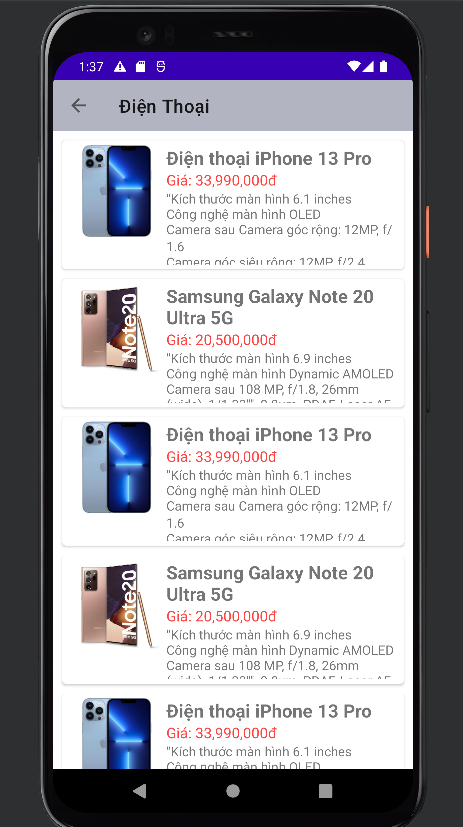
Hình 4.4 Giao diện menu list

### Giao diện trang chủ



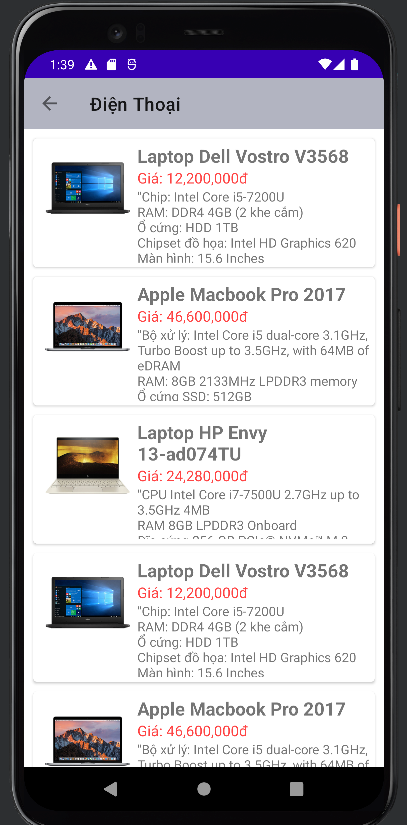
Hình 4.5 Giao diện trang chủ

### Giao diện trang điện thoại



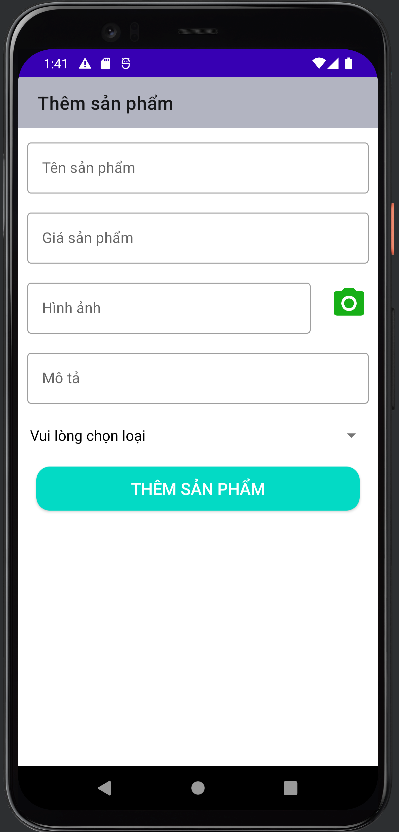
Hình 4.6 Giao diện trang điện thoại

### Giao diện trang laptop



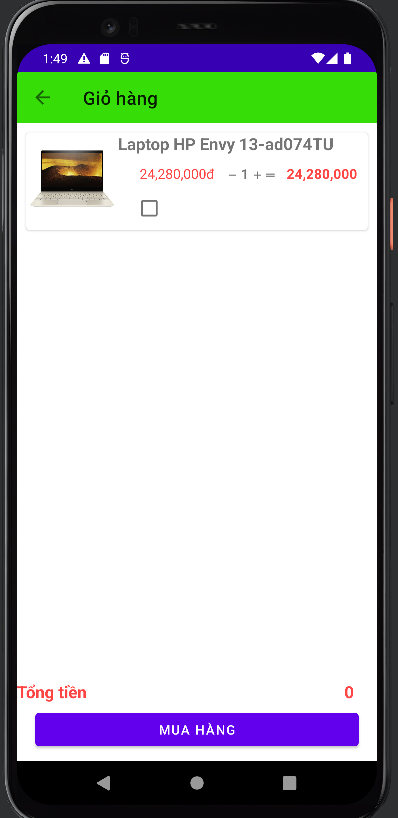
Hình 4.7 Giao diện trang laptop

### Giao diện thêm sửa sản phẩm



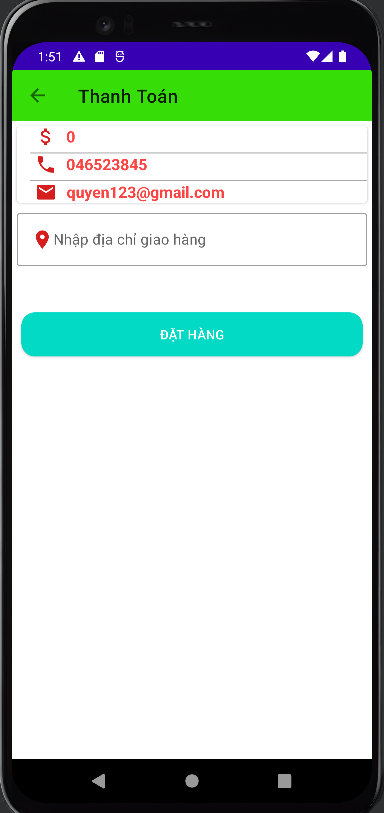
Hình 4.8 Giao diện thêm và sửa sản phẩm

### Giao diện giỏ hàng



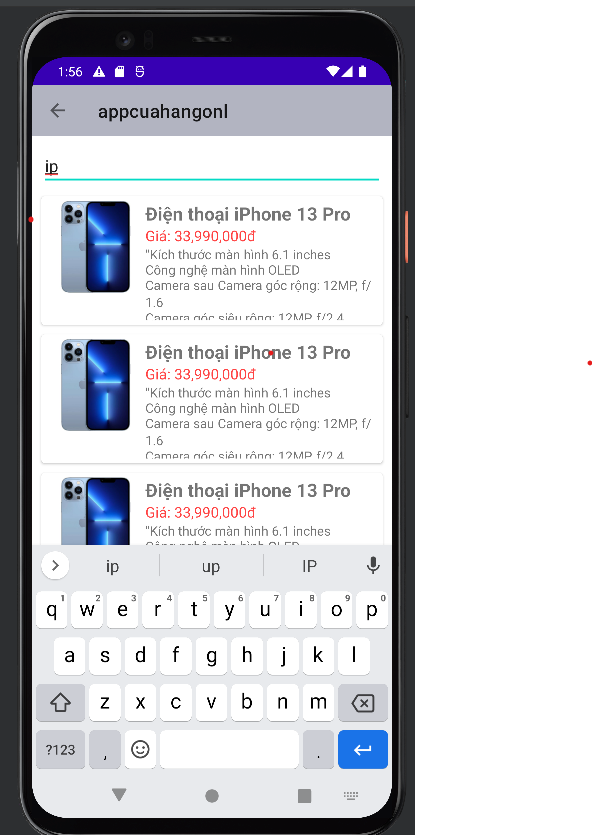
Hình 4.9 Giao diện giỏ hàng

### Giao diện thanh toán



Hình 4.10 Giao diện thanh toán

### Giao diện tìm kiếm



Hình 4.11 Giao diện tìm kiếm

# KẾT LUẬN

Sau một khoảng thời gian học tập và thực hành nghiêm túc và hiệu quả, đến nay đề tài “app bán hàng” đã hoàn thành cơ bản theo đúng yêu cầu về nội dung và thời gian đã định. Trong quá trình nghiên cứu và thực hiện đề tài với quyết tâm cao nhưng do hạn chế về kinh nghiệm và kiến thức nên đề tài chắc chắn sẽ không thể tránh khỏi những thiếu sót. Rất mong nhận được ý kiến đóng góp từ quý thầy cô và các bạn đề đề tài được hoàn thiện hơn.

##### Kết quả đạt được

* Xây dựng được phần mềm có thể ứng dụng được thực tế.
* Hiểu và lập trình thành công ngôn ngữ Kotlin, Java.
* Hiểu và sử dụng được Firebase Cloud Firesote.

Bên cạnh những kết quả đạt được là những hạn chế của đề tài:

Còn chưa đạt được nét thẩm mỹ và các chức năng còn thiếu sót nhiều..

##### Hướng nghiên cứu

Với những hạn chế và tồn tại nêu trên, hướng nghiên cứu của đề tài dự kiến như sau:

* Tìm hiểu thêm một số chức năng nâng cao cần thiết cho phần mềm
* Khai thác thêm tài nguyên có sẵn để giúp phần mềm hoàn thiện và đa dạng

# TÀI LIỆU THAM KHẢO

1. [Kotlin Programming Language (kotlinlang.org)](https://kotlinlang.org/)
2. [Best trivia apps of 2021 | Tom's Guide (tomsguide.com)](https://www.tomsguide.com/round-up/best-trivia-apps)
3. [Complete Android Quiz App Using Android Studio |(campcodes.com)](https://www.campcodes.com/mobile-app/android-studio/complete-android-quiz-app-using-android-studio/)
4. [Build a Complete Quiz App for Android from Scratch Using Kotlin and Android Studio - YouTube](https://www.youtube.com/watch?v=b21fiIyOW4A)
5. [Create Quiz App for Android - Make Quiz Mobile Game (appsgeyser.com)](https://appsgeyser.com/blog/create-quiz-app-for-android/)
6. [Build a Complete Quiz App for Android from Scratch Using Kotlin and Android Studio (morioh.com)](https://morioh.com/p/6046cddc2a09)