

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH



CÁI HOÀNG HUYNH

APP MOBILE SÀN THƯƠNG MẠI ĐIỆN TỬ

ĐỒ ÁN NGÀNH
NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, 2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH



CÁI HOÀNG HUYNH

APP MOBILE SÀN THƯƠNG MẠI ĐIỆN TỬ

Mã số sinh viên: 2051010119

ĐỒ ÁN NGÀNH
NGÀNH KHOA HỌC MÁY TÍNH

Giảng viên hướng dẫn: ThS. Nguyễn Thị Mai Trang

TP. HỒ CHÍ MINH, 2023

LỜI CẢM ƠN

Lời mở đầu em xin được cảm ơn Cô Nguyễn Thị Mai Trang là giảng viên đã hướng dẫn tận tình chi tiết cũng như giải đáp thắc mắc chính cho em trong lần đồ án này, để hoàn thành tốt đồ án lần này em cũng phải cảm ơn toàn bộ giảng viên của trường Đại Học Mở Thành phố Hồ Chí Minh đã giúp em có những kiến thức quý báu, chia sẻ kinh nghiệm thực tế để em có thể vận dụng và bản thân em đã tự mình mài mò học hỏi những kiến thức mới trải dài theo đoạn đường em còn trên ghế nhà trường.

Mặc dù khoảng thời gian làm đồ án chỉ vỏn vẹn hai tháng gấp gút nhưng nhờ có Cô và mọi người đã giúp đỡ em rất nhiều để lại trong em nhiều cảm xúc và trong khoảng thời gian này em thấy mình còn nhiều điểm thiếu sót từ kiến thức cũng như kinh nghiệm làm việc nhóm xử lý tình huống, chưa được tự tin và còn hay nóng vội nhỡ vậy mà khi hoàn thành đồ án em đã cảm thấy mình tiến bộ hơn từng ngày và kể cả cho con đường đi sau này khi bước ra xã hội ôn ào náo nhiệt đầy cạnh tranh ngoài kia. Vì thế em mong nhận được những phản hồi quý báu của quý Thầy Cô để em có thể cải thiện sản phẩm app mobile sàn thương mại điện tử này hơn và cũng là kinh nghiệm cho em sau này, em xin cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....
.....

TÓM TẮT ĐỒ ÁN NGÀNH

Trong thời buổi công nghiệp hóa hiện đại hóa ngày nay thì giao thương buôn bán là một hoạt động không thể thiếu của nước ta, khi áp dụng công nghệ thông tin vào giao thương đã nâng hoạt động này lên một tầm cao mới được ví như hô mọc thêm cánh từ ý tưởng trên nên em đã nghĩ ra ý tưởng làm một sàn thương mại điện tử để cho mọi người dễ dàng mua bán trò chuyện trao đổi nhiều loại hàng hóa. Hiện nay trên thị trường có rất nhiều ngôn ngữ lập trình android và sau khi thu thập ý kiến hơn 100 bạn sinh viên trường mình và nhóm em đã cân nhắc kỹ cũng như để dễ tiếp cận sử dụng nên em chọn phát triển trên nền tảng app mobile và code bằng ngôn ngữ java và làm giao diện bằng thư viện material Ui android còn phần database quản lý cơ sở dữ liệu thì nhóm em dùng firebase.

ABSTRACT

MỤC LỤC

Chương 1. GIỚI THIỆU ĐỀ TÀI.....	8
1.1. Giới thiệu đề tài.....	9
1.2. Lý do chọn đề tài.....	9
1.3. Bố cục đề tài.....	9
1.4. Mục tiêu nghiên cứu.....	10
Chương 2. CƠ SỞ LÝ THUYẾT.....	11
2.1. Giới thiệu về android studio.....	11
2.2. Tìm hiểu về android.....	12
2.2.1. Giao diện.....	12
2.2.2. Gradle Build System:.....	12
2.2.3. Emulator và Device Testing:.....	13
2.2.4. Code Editor.....	14
2.2.5. Layout Editor.....	14
2.2.6. Community and Documentation:.....	15
2.3. Tạo ứng dụng đầu tiên với Android Studio.....	15
2.3.1. Cài đặt Android Studio.....	15
2.3.2. Tạo Empty project.....	16
2.3.3. Cấu trúc thư mục của Project.....	20
2.4. Cấu hình ứng dụng Android app.....	21
2.4.1. Xử lý hệ thống.....	21
2.4.2. Xử lý giao diện UI.....	22
2.5. Kết nối với Firebase.....	24
2.5.1. Tạo dự án Firebase.....	24
2.5.2. Thêm ứng dụng android.....	24
2.5.3. Kết nối Android với database.....	27
2.5.4. Khởi tạo Firebase trong ứng dụng.....	30
2.5.5. Sử dụng Firebase trong ứng dụng.....	31
2.6. Cấu trúc tổ chức file trong Android app.....	31
2.6.1. Các Class Activity.....	31
2.6.2. Các Class Adapter.....	35
2.6.3. Các Class hướng đối tượng.....	37
2.6.4. Các Class kết nối Database.....	38
Chương 3. XÂY DỰNG HỆ THỐNG APP MOBILE SÀN THƯƠNG MẠI.....	41
3.1. Giới thiệu hệ thống.....	41
3.2. Phân tích và thiết kế hệ thống.....	41
3.2.1.1. Xác định yêu cầu.....	41
3.2.2. Phân tích yêu cầu.....	41
3.2.3. Sơ đồ use case.....	42

3.2.4. Đặc tả use case.....	45
3.2.5. Activity Diagram.....	54
3.3. Thiết kế cơ sở dữ liệu.....	58
3.4. Các bảng trong CSDL quan hệ.....	59
3.4.1. Bảng Product.....	59
3.4.2. Bảng Follow.....	60
3.4.3. Bảng Comment.....	61
3.4.4. Bảng Chat.....	61
3.4.5. Bảng Category.....	62
3.4.6. Bảng Pay.....	62
3.4.7. Bảng User.....	63
3.5. Các giao diện của app mobile.....	64
Chương 4. KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN.....	71
4.1. Kết luận.....	71
4.2. Định hướng phát triển.....	72
4.2.1. Ưu điểm.....	72
4.2.2. Nhược điểm.....	72

MỞ ĐẦU

Chương 1. GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu đề tài

Với sự phát triển mạnh mẽ của giao thương hiện nay cùng với sự lan tỏa nhanh chóng của công nghệ thông tin và trí tuệ nhân tạo ra đời và mọi người hay có thói quen thích đi mua sắm trực tiếp để có thể tự mình trải nghiệm sản phẩm cũng như lựa được cho mình một sản phẩm như ý hợp với túi tiền và rất tiện lợi bên cạnh đó thì cũng có những bất cập lớn như phải từ mình di chuyển hàng hóa và đặc biệt là thời gian cuối năm 2019 thì covid-19 bùng nổ nó đã ảnh hưởng lớn đến toàn bộ khắp thế giới và ngành bị ảnh hưởng là giao thương buôn bán vì không thể được gặp trực tiếp và những ứng dụng buôn bán trên mạng không được các doanh nghiệp đầu tư, từ đây em đã áp ủ cho mình một ý tưởng làm app mobile sàn thương mại điện tử để giải quyết vấn đề trên ngoài ra sàn thương mại điện tử không chỉ có buôn bán các sản phẩm online mà mọi người có thể trò chuyện đánh giá sản phẩm một cách khách quan chỉ bằng vài cái chạm màn hình.

1.2. Lý do chọn đề tài

Khi mà Công nghệ thông tin nó trở thành một phần không thể thiếu trong cuộc sống của mỗi người dân đặc biệt là giới trẻ rất dễ bắt kịp xu hướng cộng với sự bất tiện mà mua hàng trực tiếp như đã nêu trên cũng như mong muốn của nhóm em tạo ra một sản phẩm phục vụ đời sống người dân một cách hoàn hảo và tiện nghi hơn, với đầy đủ các chức năng buôn bán và trao đổi thông tin giữa người bán người mua một cách đơn giản nhưng cực kỳ hiệu quả giảm thiểu những phi phí không cần thiết từ đó nên em quyết định chọn đề tài app mobile sàn thương mại điện tử này.

1.3. Bố cục đề tài

Chương 1: Giới thiệu đề tài

Chương 2: Cơ sở lý thuyết

Chương 3: Xây dựng hệ thống App mobile sàn thương mại điện tử

Chương 4: Kết luận và định hướng phát triển

1.4. Mục tiêu nghiên cứu

Sau khi cân nhắc tiếp thu ý kiến của mọi người và sự giúp đỡ tận tình của Giảng Viên hướng dẫn, nhóm chúng em đã phân tích kỹ hệ thống sàn thương mại điện tử đang có trên thị trường hiện nay như Shopee, Lazada, vv... từ đó học hỏi, tham khảo vào tạo ra một hệ thống sàn thương mại điện tử hoàn chỉnh đầy đủ tính năng và có thể đáp ứng được nhu cầu mua sắm rất lớn của người tiêu dùng.

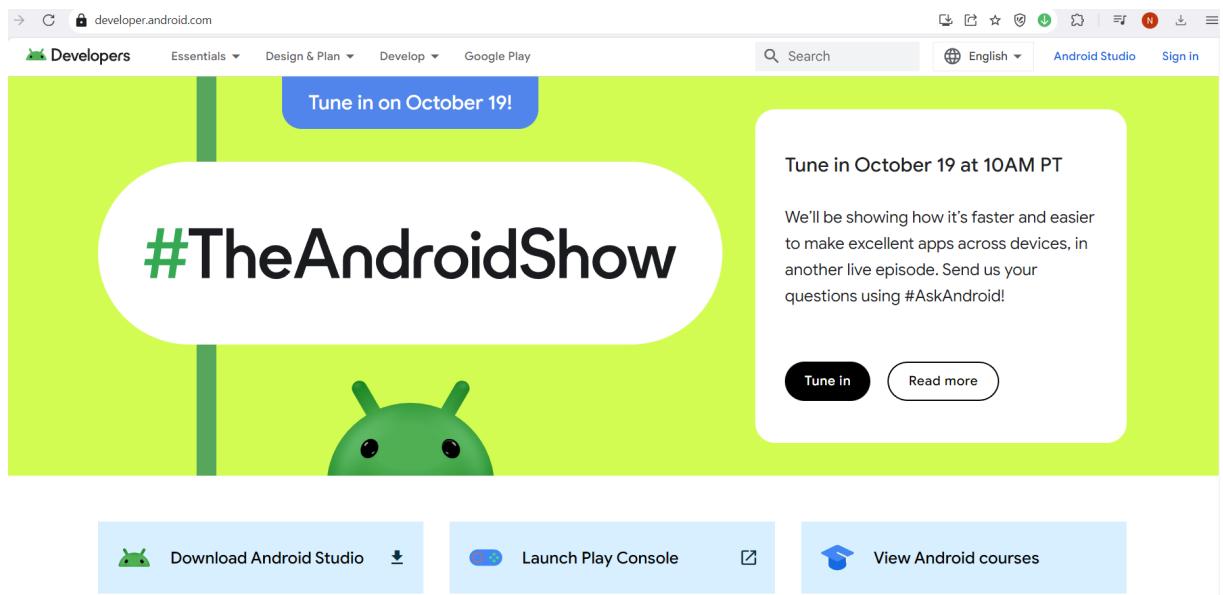
Do làm app mobile nên nhóm em lựa chọn ngôn ngữ lập trình app android phổ biến hiện nay là java với sự hỗ trợ lớn lao từ cộng đồng lập trình viên và thư viện vào giao diện app tuy đơn giản nhưng người dùng dễ tiếp cận và sử dụng và tiếp cận ở mọi lứa tuổi hoặc ít khi tiếp xúc với công nghệ đó là thư viện lớn của android do Google phát triển Material UI Android và phía back-end, database nhóm chúng em lựa chọn sử dụng firebase cụ thể là database-realtime cũng do ông lớn Google đứng sau và điểm mạnh của firebase là khả năng cập nhập dữ liệu thời gian thực khiến người dùng có một trải nghiệm tuyệt vời.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu về android studio

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Android Inc. với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005.

Từ năm 2011, Android đã là hệ điều hành bán chạy nhất trên toàn cầu trên điện thoại thông minh và từ năm 2013 trên máy tính bảng. Tính đến tháng 5 năm 2021, nó có hơn ba tỷ người dùng hàng tháng, là hệ điều hành có cài đặt nhiều nhất trên thế giới và tính đến tháng 1 năm 2021, Cửa hàng Google Play có hơn 3 triệu ứng dụng. Android 13, được phát hành vào ngày 15 tháng 8 năm 2022, là phiên bản mới nhất, và phiên bản Android 12.1/12L mới nhất bao gồm những cải tiến đặc biệt cho điện thoại gập, máy tính bảng, màn hình có kích thước lớn như máy tính để bàn 1080p và Chromebooks.



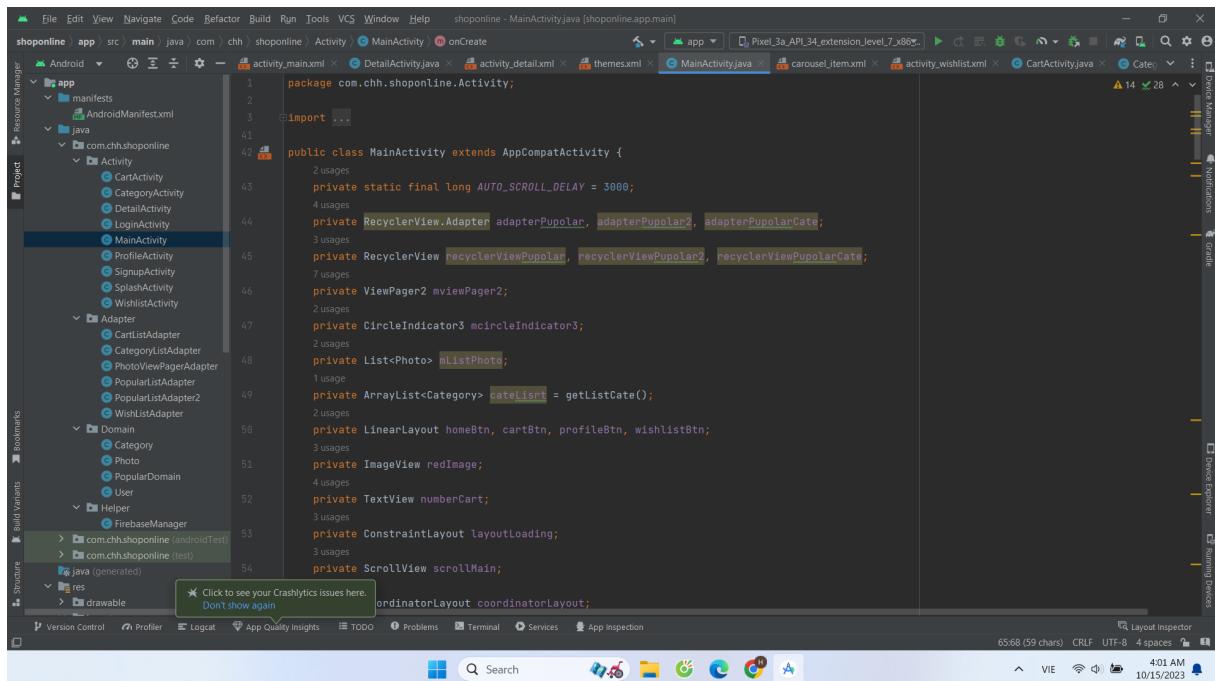
Hình 1. Giao diện web trang chủ Android studio

2.2. Tìm hiểu về android

Android Studio là một môi trường lập trình phát triển tích hợp (IDE) dành cho phát triển ứng dụng di động chạy trên hệ điều hành Android với hai ngôn ngữ thông dụng là java và kotlin. Đây là một công cụ cực kì mạnh mẽ được phát triển bởi ông lớn Google, cung cấp nhiều tính năng hữu ích cũng như giả lập để hỗ trợ việc phát triển ứng dụng Android một cách tốt nhất.

2.2.1. Giao diện

Android Studio có một giao diện thân thiện với người dùng dễ sử dụng với các nhiều chức năng được sắp xếp tiện lợi rõ ràng dễ dàng giúp các developer thao tác. Có các tab và cửa sổ chỉnh sửa mã nguồn, chạy debug, thiết kế giao diện và nhiều tính năng khác.



Hình 2. giao diện android studio

2.2.2. Gradle Build System:

Gradle Build System là một công cụ hệ thống xây dựng (build system) được sử dụng trong phát triển ứng dụng Android. Nó được tạo ra nhằm quản lý quá trình xây dựng app android, từ việc tải các thư viện cần thiết (implement) và xuất ra các file APK cuối cùng dành cho việc cài đặt trên thiết bị Android. Gradle Build system bao gồm các điểm nổi bật sau:

- Hỗ trợ nhiều ngôn ngữ (Multi-language Support): Gradle hỗ trợ nhiều ngôn ngữ lập trình, bao gồm cả Java, Kotlin và C++. Điều này cho phép nhà phát triển sử dụng ngôn ngữ phù hợp với sở thích và yêu cầu cụ thể của dự án.
- Quản lý phụ thuộc (Dependency Management): Gradle cho phép bạn quản lý các thư viện và phụ thuộc mà ứng dụng của bạn cần sử dụng. Nó có khả năng tải về và cập nhật tự động các thư viện từ các kho lưu trữ (repositories).
- Khả năng mở rộng (Extensibility): Gradle cho phép bạn mở rộng chức năng của nó thông qua các plugin. Có nhiều plugin hữu ích được cung cấp bởi cộng đồng phát triển.
- Cấu hình linh hoạt (Flexible Configuration): Gradle cho phép bạn định cấu hình quy trình xây dựng ứng dụng Android của mình theo cách tùy chỉnh, từ việc thiết lập biến môi trường đến việc định nghĩa các tác vụ tùy chỉnh.
- Hiệu suất tốt (Performance): Gradle được thiết kế để xử lý các dự án lớn và phức tạp một cách hiệu quả, đảm bảo thời gian xây dựng ngắn gọn. Ngoài ra Gradle hỗ trợ tự động hóa quá trình xây dựng ứng dụng. Bạn có thể thiết lập các tác vụ tự động chạy khi có sự thay đổi trong mã nguồn.

2.2.3. Emulator và Device Testing:

Emulator Testing và Device Testing là hai phương pháp chính để kiểm thử ứng dụng Android trước khi phát hành ra thị trường. Cụ thể là:

- Emulator testing là máy ảo giả lập chạy môi trường android trên thiết bị khác cụ thể là máy tính. Nó cho phép bạn chạy ứng dụng android mà mình đang code trên đó mà không cần phải dùng thiết bị điện thoại thật, Emulator hỗ trợ nhiều phiên bản giả lập android khác nhau giúp app của bạn có thể chạy trên đa nền tảng ngoài ra nó còn có thể mô phỏng tình trạng mạng khác nhau giúp bạn dễ dàng kiểm tra app của mình hơn.
- Device testing là việc kiểm tra ứng dụng trên thiết bị Android thực sự. Điều này đảm bảo rằng ứng dụng hoạt động chính xác trên các thiết bị thực tế. Kiểm tra thực tế nó giúp bạn có cái nhìn thực tế hơn về app mình trên phương diện hiệu xuất, cảm biến, hiệu năng, ngoài ra có một số tính

năng như Bluetooth, GPS và những thay đổi giao diện nhỏ nhất khi chạy trên môi trường máy ảo.

2.2.4. Code Editor

Code editor trong phát triển app Android là công cụ cho phép nhà phát triển viết, chỉnh sửa và quản lý mã nguồn của dự án. Đây là nơi bạn viết mã Java, Kotlin hoặc các ngôn ngữ lập trình khác được sử dụng để phát triển ứng dụng Android và Android Studio, là một IDE (Integrated Development Environment), cung cấp một trình soạn thảo mã nguồn tích hợp. Trình soạn thảo mã nguồn code editor hỗ trợ những chức năng sau:

- Code Completion (tự động hoàn thành mã nguồn): Gợi ý và hoàn thành các lỗi chính tả cho dev như lỗi từ vựng, tên biến tên hàm hoặc gợi ý các trường hợp code thường xuyên dùng.
- Kiểm tra lỗi và gỡ lỗi (code inspection và debugging): IDE kiểm tra lỗi về mặt cú pháp và gợi ý cách cách thay thế hoặc sửa lỗi và debugging sẽ giúp dev theo dõi luồng chạy dữ liệu để hạn chế gặp bug.
- Refactoring (tái cấu trúc mã nguồn): Cho phép thay đổi cùng lúc những tên biến giống nhau, ngoài ra IDE còn tích hợp tô màu cú pháp giúp dev có thể dễ dàng nhận dạng phân biệt và đánh dấu các phần trong source code của mình
- Integration with Version Control (tích hợp quản lý phiên bản): IDE có tích hợp tốt với các hệ thống quản lý phiên bản như Git, giúp dev có thể dễ dàng quản lý mã nguồn và làm việc nhóm hiệu quả và IDE còn hỗ trợ nhiều công cụ và plugin mở rộng để tăng cường tính năng của nó.

2.2.5. Layout Editor

Layout Editor trong Android Studio là một công cụ cho phép lập trình viên thiết kế giao diện người dùng của ứng dụng Android một cách trực quan và dễ dàng. Đây là nơi bạn có thể tạo, chỉnh sửa, xóa và xem trước cấu trúc cũng như bối cảnh từng trang giao diện của các màn hình ứng dụng của mình. Một số tính năng quan trọng của Layout Editor:

- Trình điều khiển kéo thả (design): Lập trình viên có thể kéo hoặc thả các thành phần mình muốn như button, textview, linearlayout, imageview, vv..., vào màn hình thiết kế.
- Hiển thị cấu trúc XML (show xml): Bạn có thể chuyên đổi giữa chế độ chỉnh sửa mã XML để tùy chỉnh giao diện theo ý của mình bởi có một số thành phần trên design kéo thả không hiệu quả hoặc khó thao tác.
- Tự động tạo code (Generate Code): Layout Editor có thể tự động tạo mã XML tương tự với giao diện bạn thiết kế ngoài ra bạn có thể thay đổi thuộc tính của từng phần tử trong giao diện như: kích thước, màu sắc chữ, background, vị trí của phần tử đó.

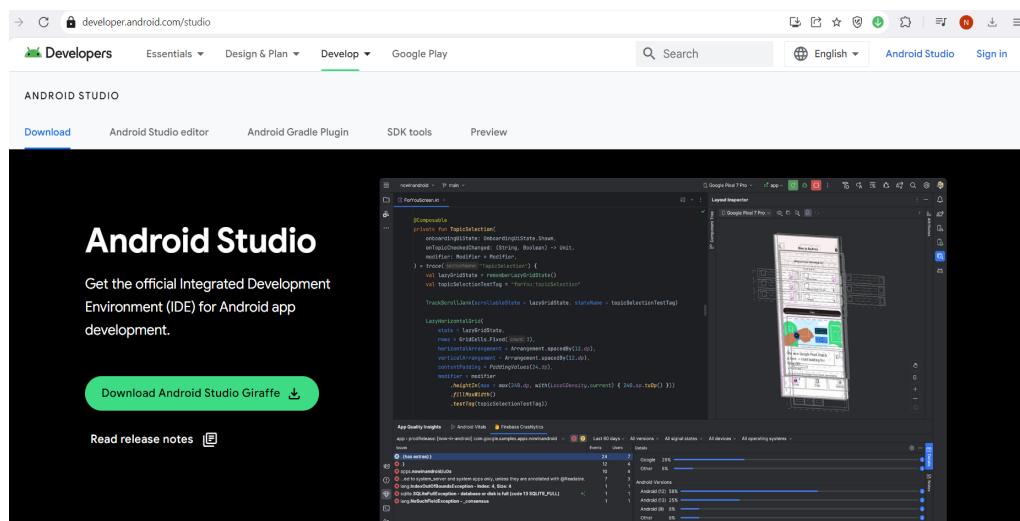
2.2.6. Community and Documentation:

Nhắc đến một công cụ mạnh như Android Studio thì không thể thiếu đó là sự hỗ trợ và phát triển của ông lớn Google vì thế cộng đồng của android cực kỳ lớn mạnh và nguồn tài liệu phong phú và dễ tìm thấy. và đặc biệt nhất là android hỗ trợ đa ngôn ngữ như java, kotlin, C++, vv..., vì thế các lập trình viên rất thực hiện cung nhu tiếp cận nó.

2.3. Tạo ứng dụng đầu tiên với Android Studio

2.3.1. Cài đặt Android Studio

- Đầu tiên chúng ta cần tải bộ cài đặt của Android Studio từ trang web chính thức, Đường dẫn tới trang tải xuống là: <https://developer.android.com/studio>, sau khi click vào link thì giao diện web của Android Studio hiện lên và chọn vào nút màu xanh lá cây nằm giữa bên tay trái có tên là “Download Android Studio Giraffe”

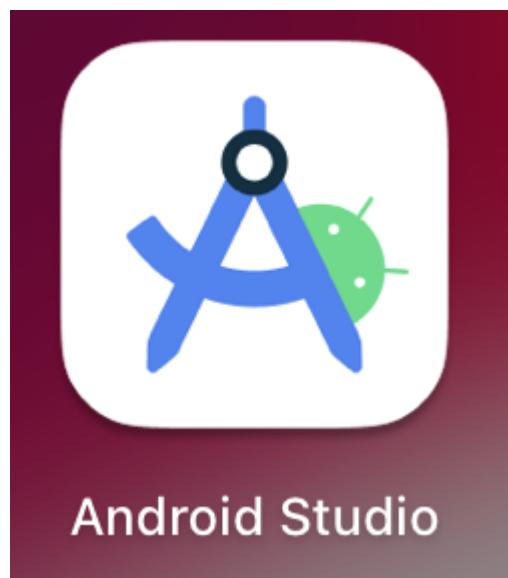


Hình 3. Giao diện web tải xuống của Android Studio

- Đợi khi tải xong về máy ta double click vào biểu tượng Android Studio để bắt đầu chạy bộ cài đặt.
- Khi lần đầu mở lên bạn sẽ thấy một cửa sổ “Welcome to Android Studio”. Nhấn “Next” để tiếp tục.
- Sau đó ta chọn thư mục mà mình muốn lưu trữ rồi nhấn “Next”. Tiếp đến Android Studio sẽ đề xuất cài đặt một số thành phần cần thiết. Để bạn chọn các component mà bạn cần để lập trình app, sau đó nhấn “Next”.
- Tiếp theo Android Studio hỏi bạn chọn thư mục lưu trữ các file SDK của Android. bạn có thể để mặc định rồi nhấn “Next”. Sau đây là chờ quá trình tải xuống hoàn tất, nhấn “Finish” để thoát khỏi trình cài đặt.

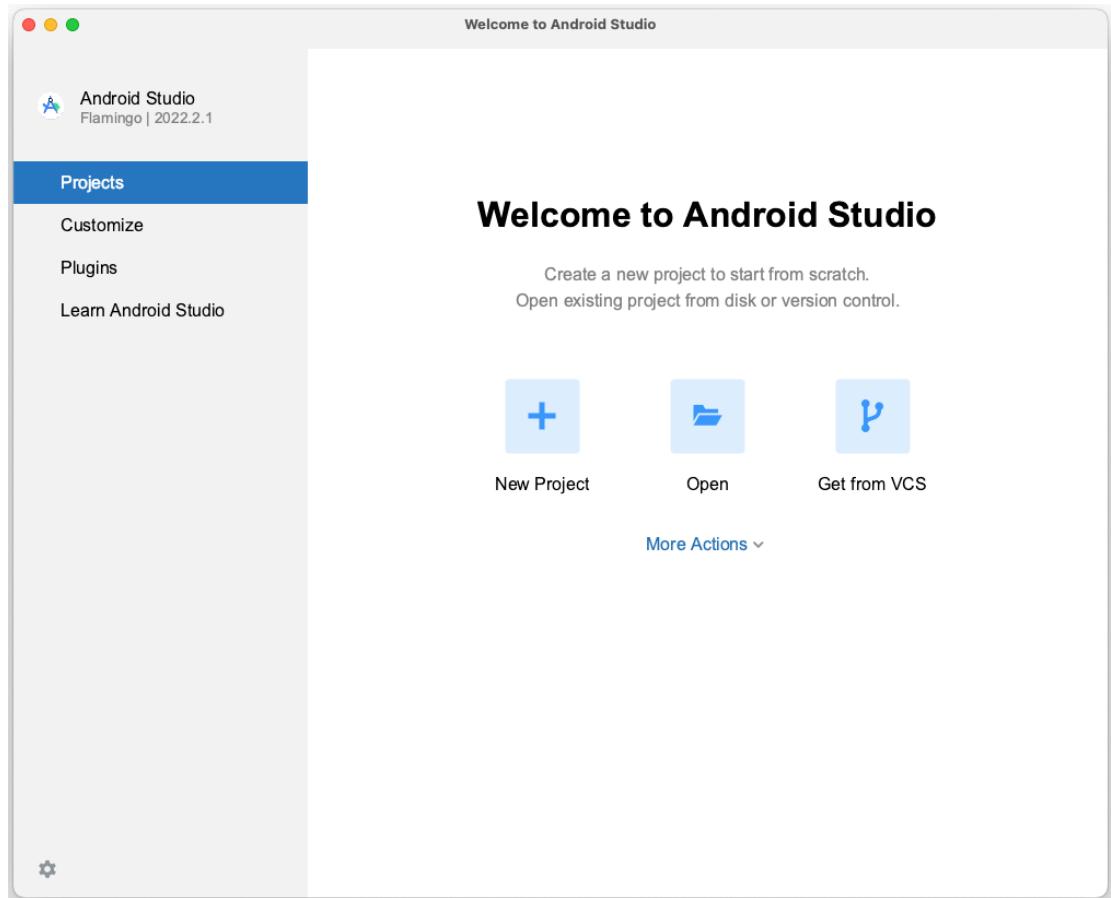
2.3.2. Tạo Empty project

- Double click vào biểu tượng Android Studio trên màn hình máy tính



Hình 4. biểu tượng Android Studio

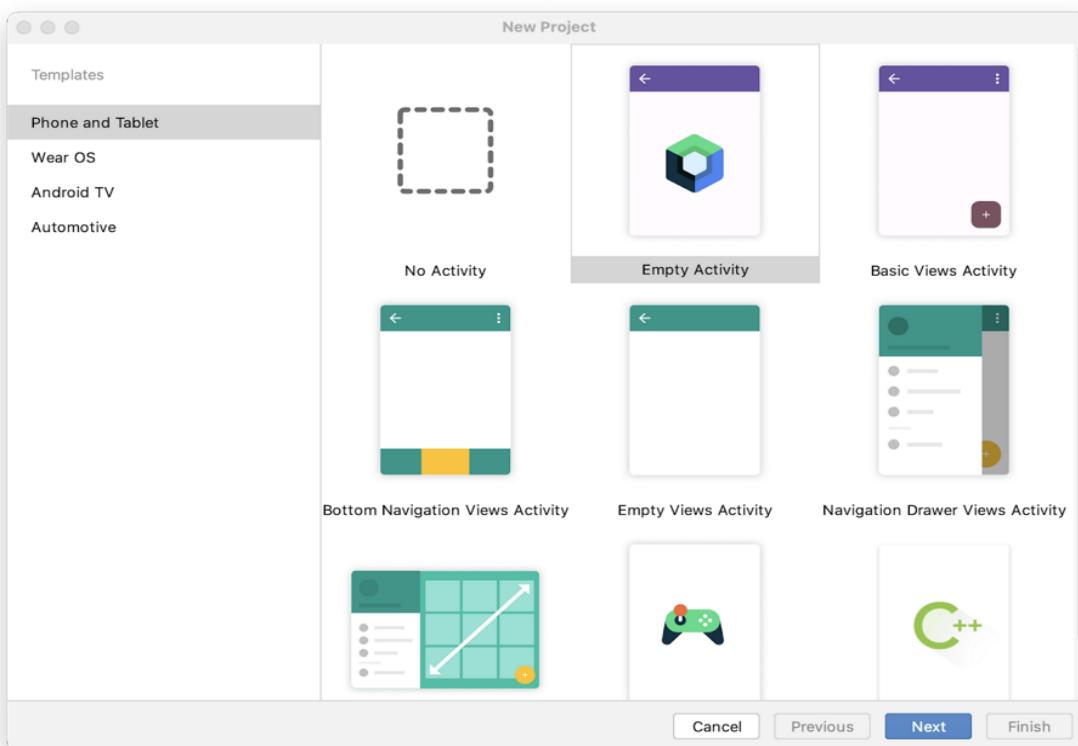
- Trong cửa sổ được mở lên “Welcome to Android Studio” chúng ta chọn



“new project”.

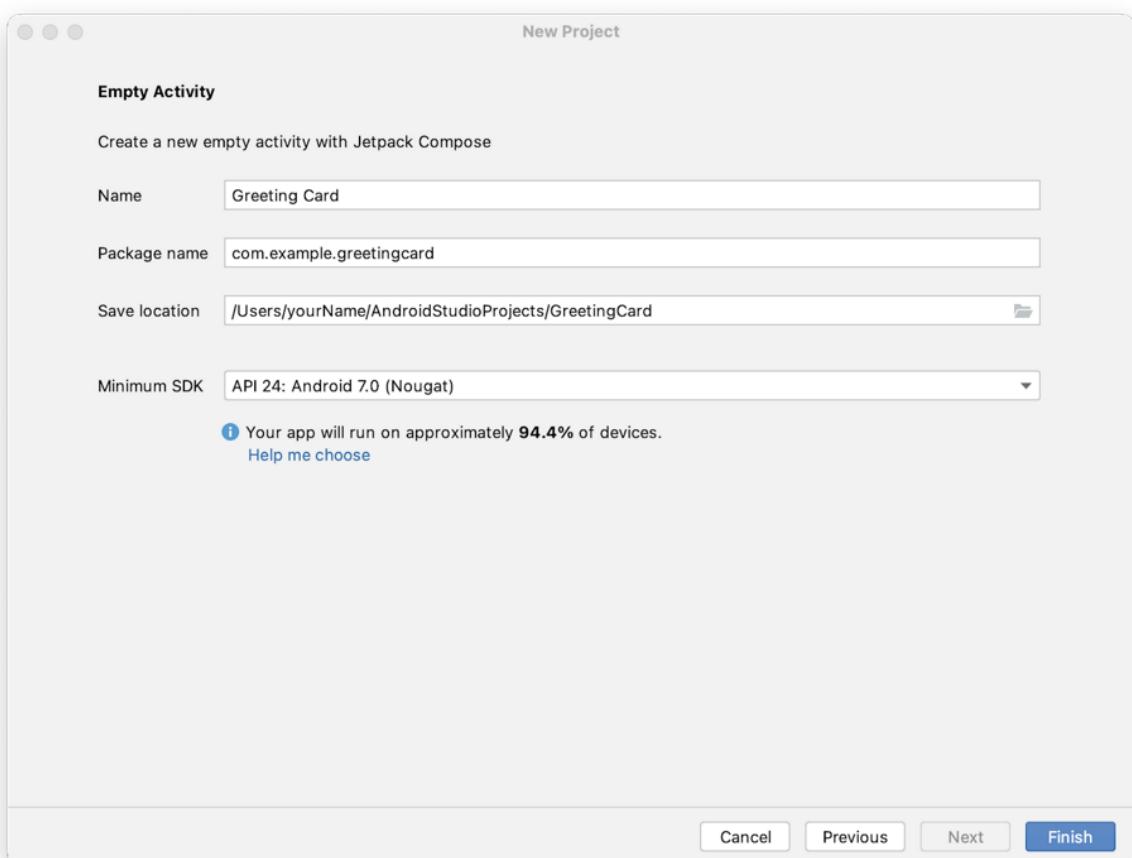
Hình 5. Cửa sổ Welcome to Android Studio

- Sau khi cửa sổ New Project hiện lên chúng ta sẽ thấy một list các templates được Android Studio cung cấp sẵn.



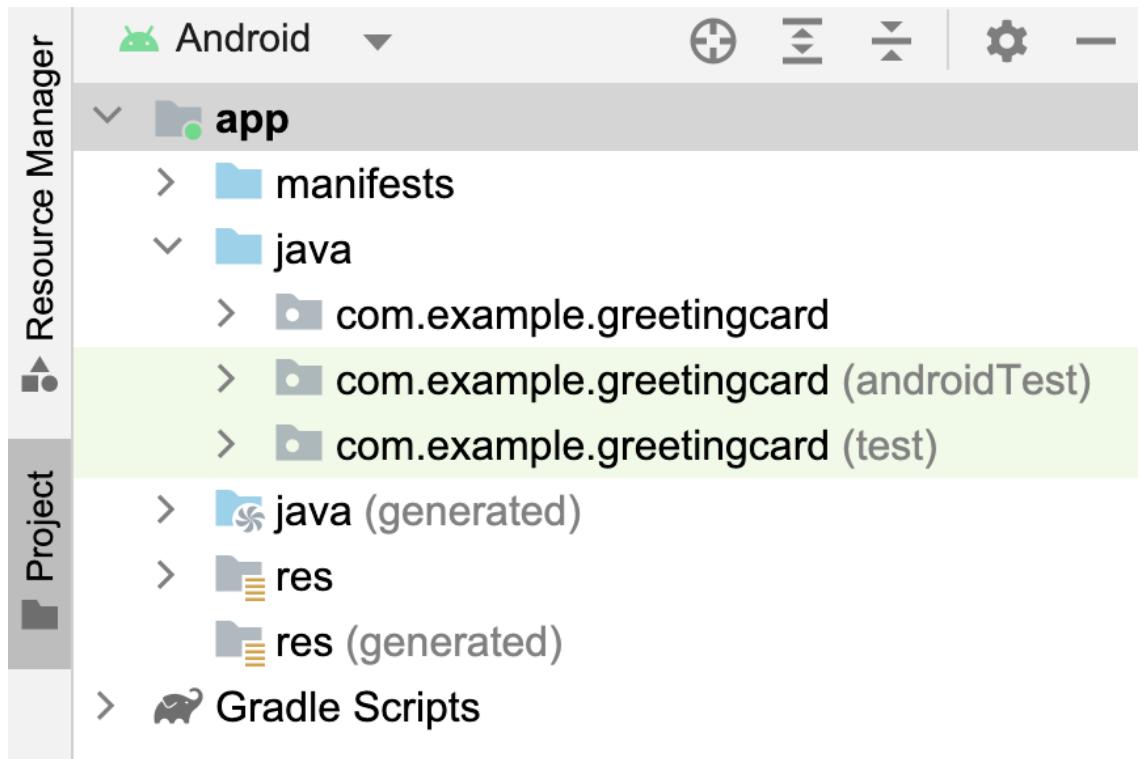
Hình 6. Cửa sổ New Project

- Ở đây chúng ta chọn Empty Activity sau đó nhấn “Next” thì có một hộp thoại được mở lên để chúng ta có thể cấu hình project của mình và có 4 trường chín:
 - + Trường “Name” là để người dùng nhập tên app của mình
 - + Trường “Package Name” Đây là trường để tạo cấu trúc các tệp trong project của mình, ví dụ: com.chh.shoponline
 - + Trường “Save location” Đây là nơi lưu trữ các tập tin liên quan đến project của bạn. Bạn có thể thay đổi đến nơi mình mong muốn.
 - + Trường “Minimum SDK” cho biết phiên bản tối thiểu mà app android của bạn có thể chạy trên nó, như trong ảnh thì app android của chúng ta chỉ hoạt động từ android 7 trở lên.
- Sau đó chúng ta chọn “Finish” và chờ Android Studio tạo project cho chúng ta.



Hình 7. Hộp thoại New Project

2.3.3. Cấu trúc thư mục của Project



Hình 8. Cấu trúc cơ bản của project android app

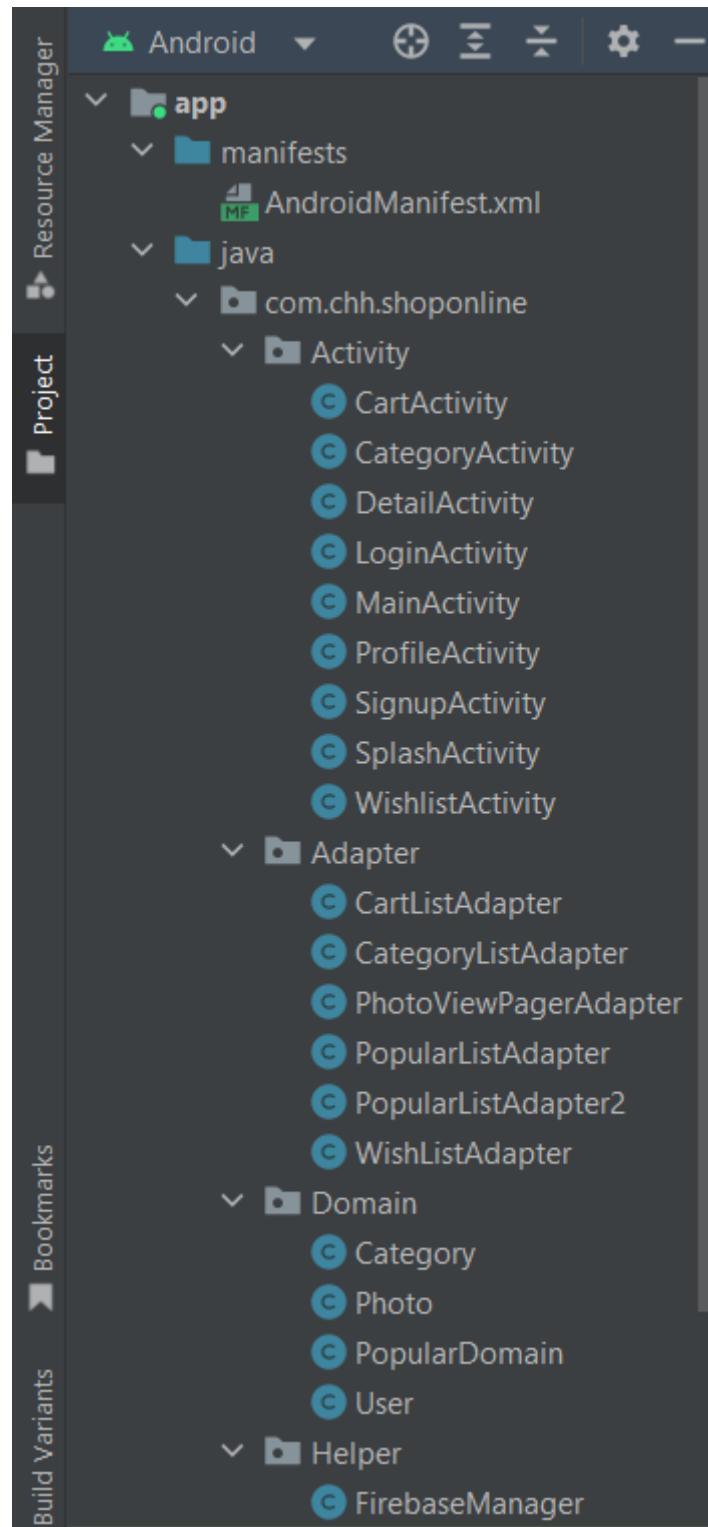
- Trong thư mục app ta có:
 - + Thư mục manifests: Tệp này chứa thông tin về ứng dụng của bạn, bao gồm quyền truy cập, khai báo hoạt động, dịch vụ.
 - + Thư mục java/ com.example.myapp/ MainActivity.java: Đây là tệp mã nguồn java cho hoạt động chính của ứng dụng. nơi bạn viết code logic ở đây cũng như là phần đầu tiên chạy khi app của bạn được khởi động.
 - + Thư mục res/ drawable: Đây là nơi chứa các hình ảnh và biểu tượng sẽ được bạn sử dụng trong ứng dụng của mình ngoài ra chúng ta cũng có thể tự tạo hình ảnh theo ý mình.
 - + Thư mục Gradle Scripts/ build.gradle(Module: app): Tệp này chứa các cài đặt cấu hình và phụ thuộc (implement) của dự án.

2.4. Cấu hình ứng dụng Android app

2.4.1. Xử lý hệ thống

Bao gồm tất cả các class nằm trong thư mục java/com.chh.shoponline và được chứa trong các thư mục:

- Thư mục Activity: Chứa toàn bộ các Class Activity như: MainActivity.class, CategoryActivity, UserActivity.class, vv..., các class này đóng vai trò như xương sống của project của chúng ta, tất cả các luồng dữ liệu xử lý giữa người dùng với hệ thống và database đều phải thông qua các class này nó đóng vai trò chung gian kết nối các phần khác lại với nhau.
- Thư mục Adapter: Chứa toàn bộ các Class Adapter như: CartListAdapter.class, WishListAdapter.class, vv..., các class này đóng vai trò cực kỳ quan trọng là nơi ánh xạ từ giao diện xuống cho các class Activity xử lý để tra ra kết quả và thông qua Adapter để có thể hiển thị ra màn hình giao diện, ngoài ra Android hỗ trợ rất mạnh mẽ trong việc render ra dữ liệu dưới dạng danh sách đối tượng gồm nhiều thuộc tính cũng như thay đổi được bối cảnh của dữ liệu.
- Thư mục Domain: Tên thư mục có thể tự do đặt tên nhưng nó mang hàm nghĩa đây là thư mục chứa những class tạo đối tượng mà ta cần hiện thực trong thực tế như: User.class, Product.class, Category.class, vv..., Những class này giúp ta có thể hiện thực hóa những đối tượng ngoài đời thực kết với với Activity để hiện thực hóa những hành động thực tế như: mua hàng hóa, bình luận, theo dõi (like, follow), thanh toán, vv...
- Thư mục Helper: Đây là nơi lưu trữ những class có chức năng liên kết với database trong app Sàn thương mại điện tử là firebase, những class này chuyên thực hiện các kết nối dữ liệu xuống database để đọc, tạo, sửa và xóa dữ liệu ngoài ra còn truy vấn dữ liệu khi được nhận yêu cầu từ Activity.

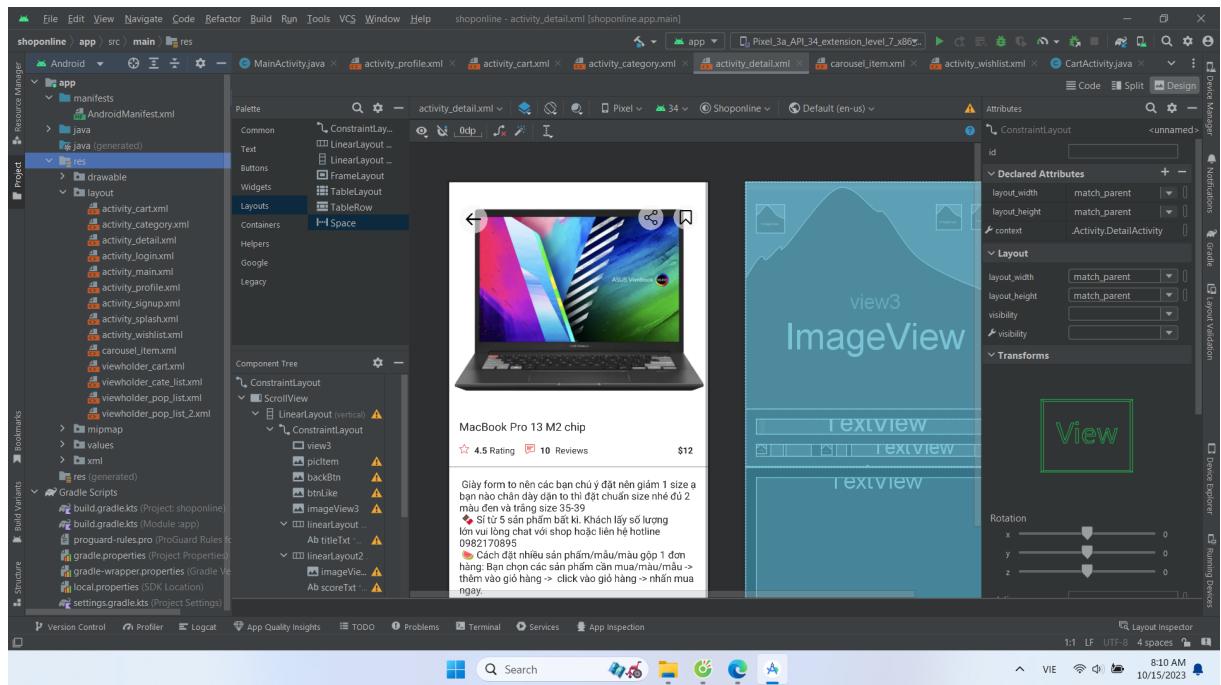


Hình 9. Cấu hình file project xử lý hệ thống

2.4.2. Xử lý giao diện UI

Bao gồm các thư mục nằm trong thư mục “res/”:

- Thư mục drawable: Thư mục chuyên lưu trữ hình ảnh của app ngoài bạn còn có thể tự tạo ra hình ảnh đơn giản mình muốn thương là hình tròn, vuông, oval và có thể thiết lập màu tùy ý.
- Thư mục layout: Đây là thư mục quan trọng nhất vì nó chứa toàn bộ giao diện của app chúng ta được android hỗ trợ rất mạnh và nhiều thư viện về giao diện trong app Sàn thương mại điện tử thì ta sử dụng Material UI Component ngoài ra khi thiết kế giao diện có 2 cách là kéo thả và code trực tiếp, file giao diện android có đuôi là .xml, Android còn hỗ trợ thiết kế giao diện với các component như: TextView, Button, RecyclerView, ConstraintLayout, vv..., Đặc biệt ta có thể xét nhiều thuộc tính cho các phần tử trực tiếp một cách dễ dàng.
- Các thư mục còn lại chủ yếu hỗ trợ và cấu hình màu sắc, font chữ vv...

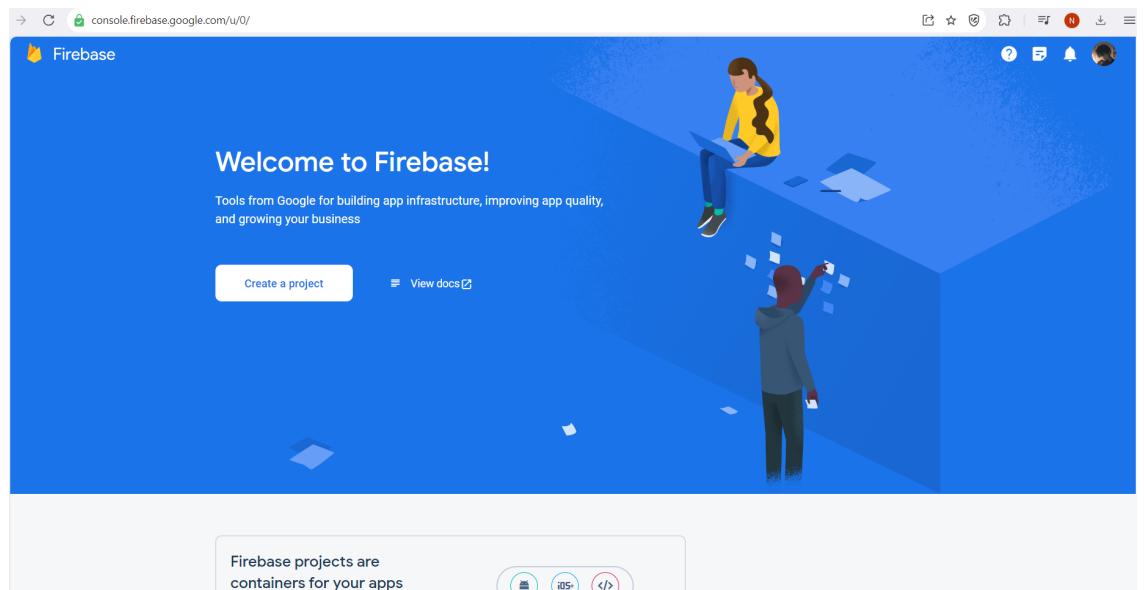


Hình 10. xử lý giao diện app Android

2.5. Kết nối với Firebase

2.5.1. Tạo dự án Firebase

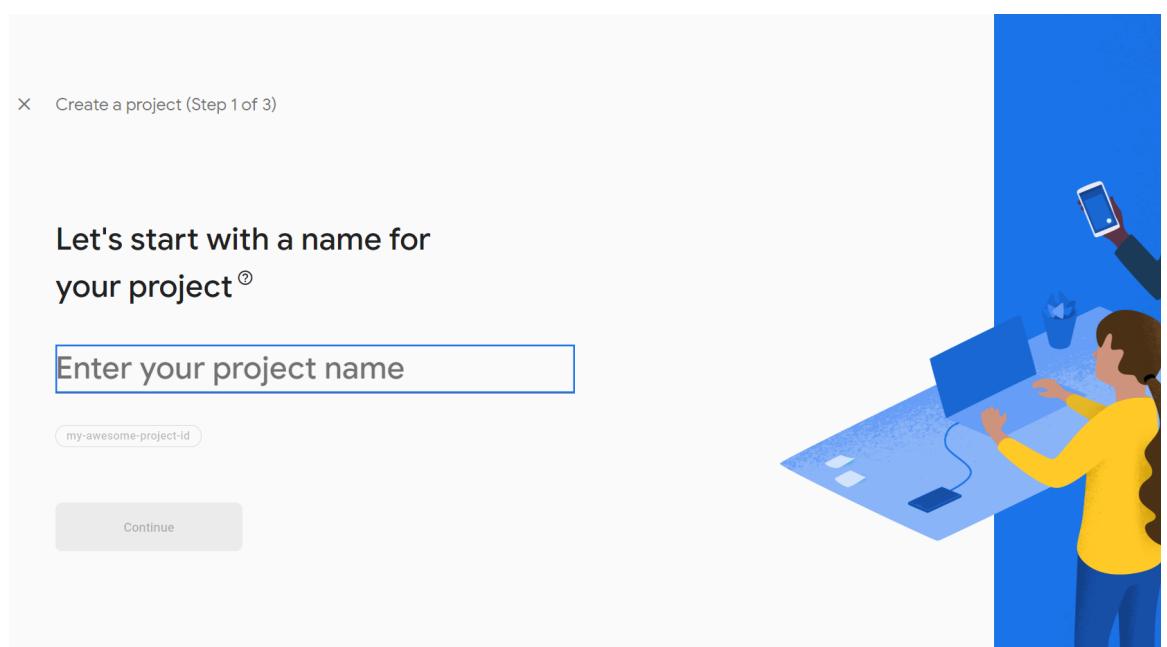
- Truy cập Firebase Console (<https://console.firebaseio.google.com>), đăng nhập bằng tài khoản Google của bạn, sau đó nhấn “Create a project” để tạo dự án mới.



Hình 11. Trang Console của firebase

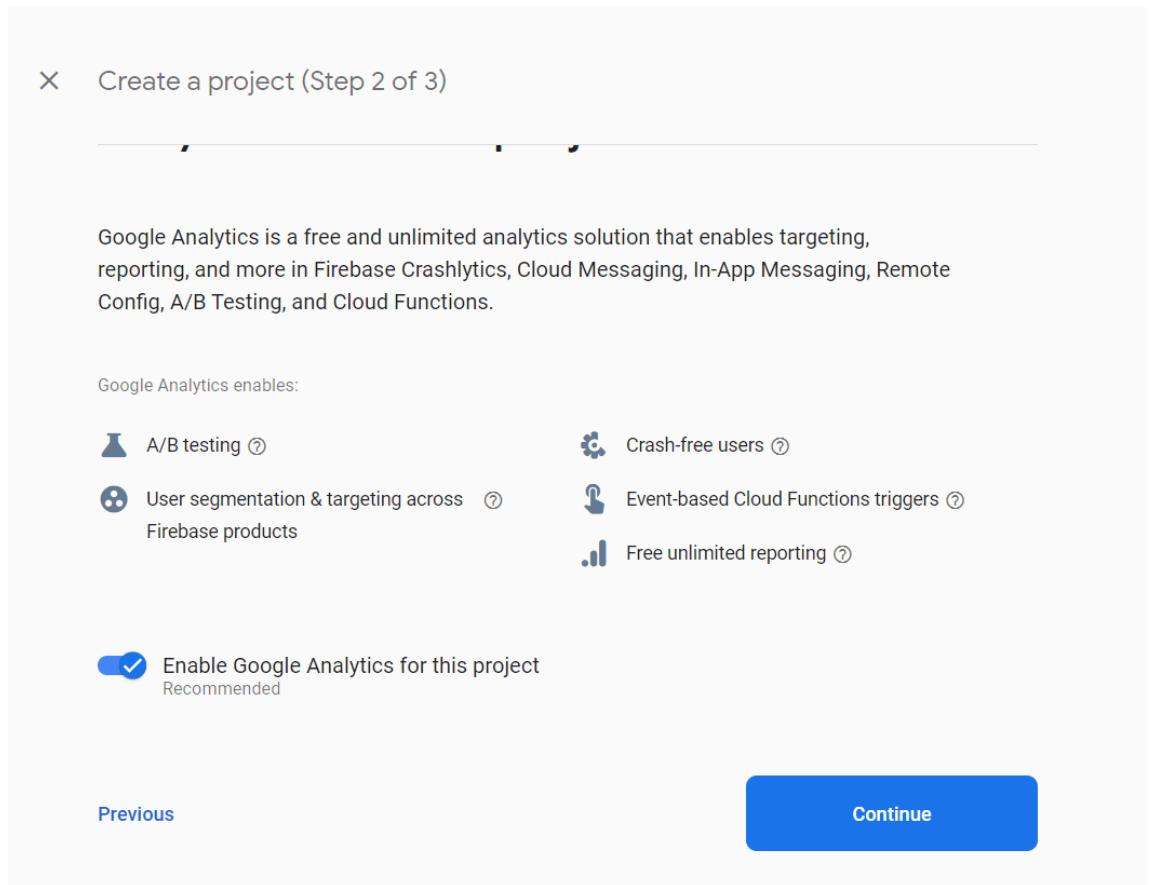
2.5.2. Thêm ứng dụng android

- Hộp thoại mới được hiện lên yêu cầu bạn đặt tên cho project của mình sau đó nhấn vào nút “Continue”.



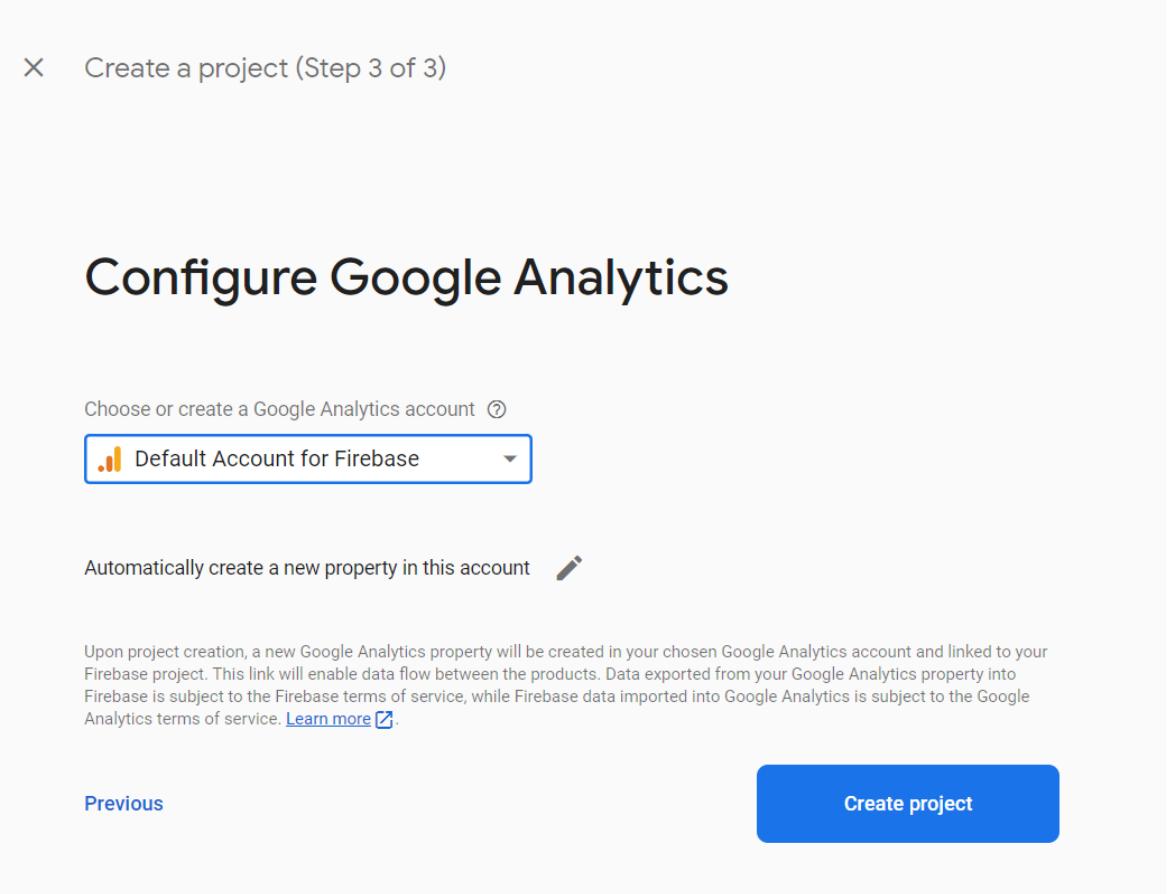
Hình 12. Đặt tên cho project android app

- Kế tiếp Firebase hỏi chúng ta về các dụng vụ hỗ trợ của Google Analytics, ta chọn “Continue”.



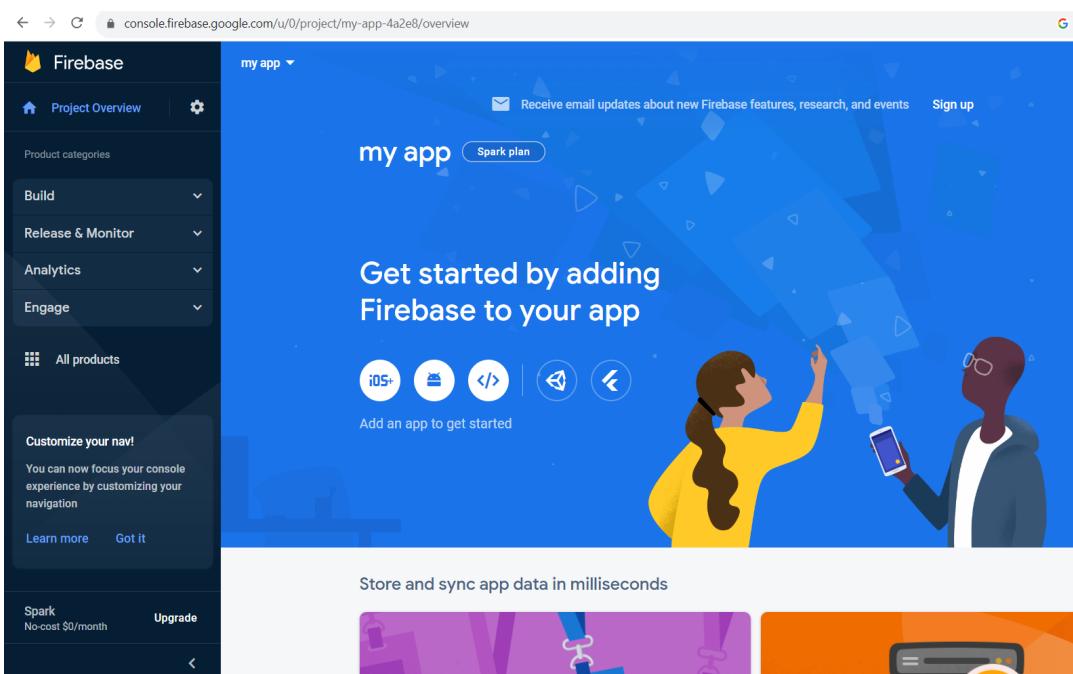
Hình 13. Các dịch vụ của Google Analytics

- Tiếp theo Firebase yêu cầu ta Configure Google Analytics, lúc này ta sẽ chọn “Default Account for Firebase” sau đó chọn “Create project” và chờ một lát để Firebase tạo project cho bạn.



Hình 14. Configure Google Analytics

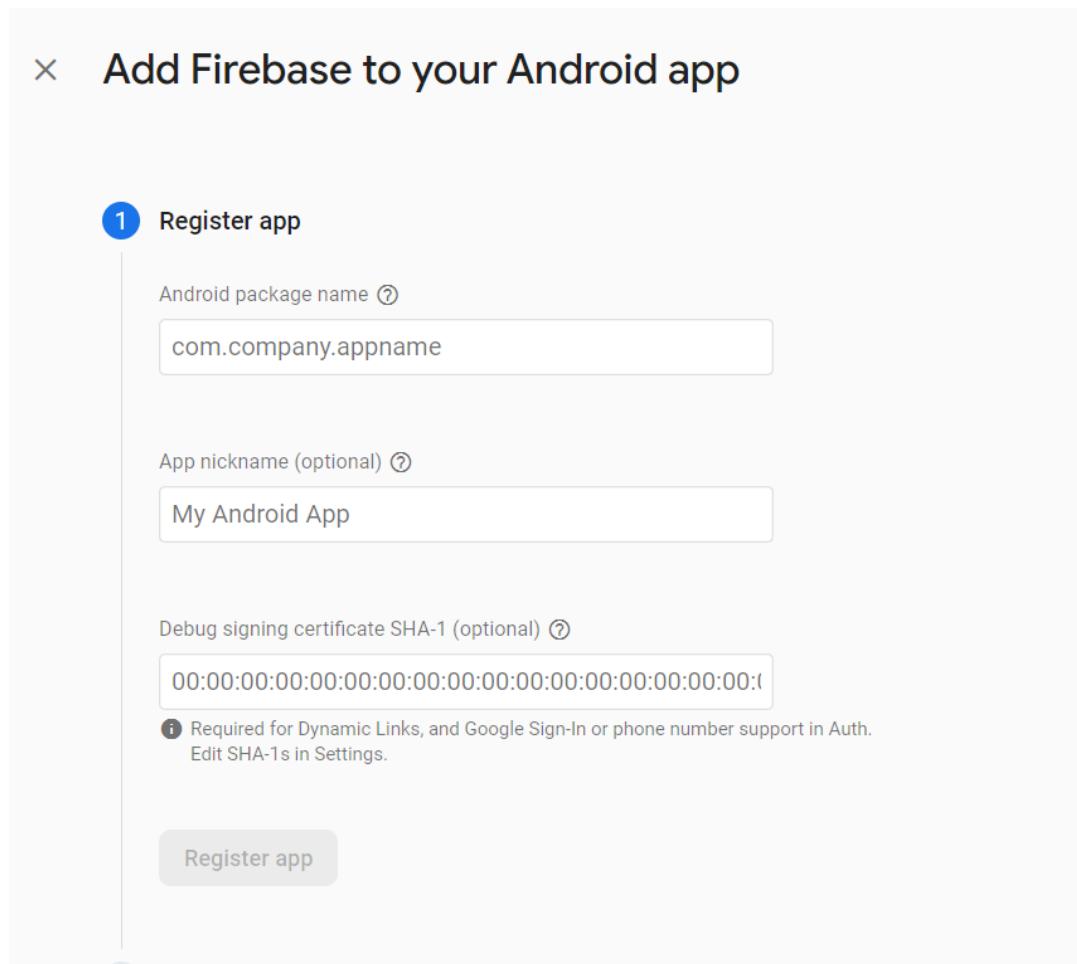
- Sau khi tạo xong project Firebase sẽ chuyển hướng bạn đến trang Console, tiếp đến bạn chọn vào biểu tượng Android nằm giữa màn hình để tiến hành kết nối đến Android app của mình.



Hình 15. Giao diện trang chủ console.firebaseio.google.com

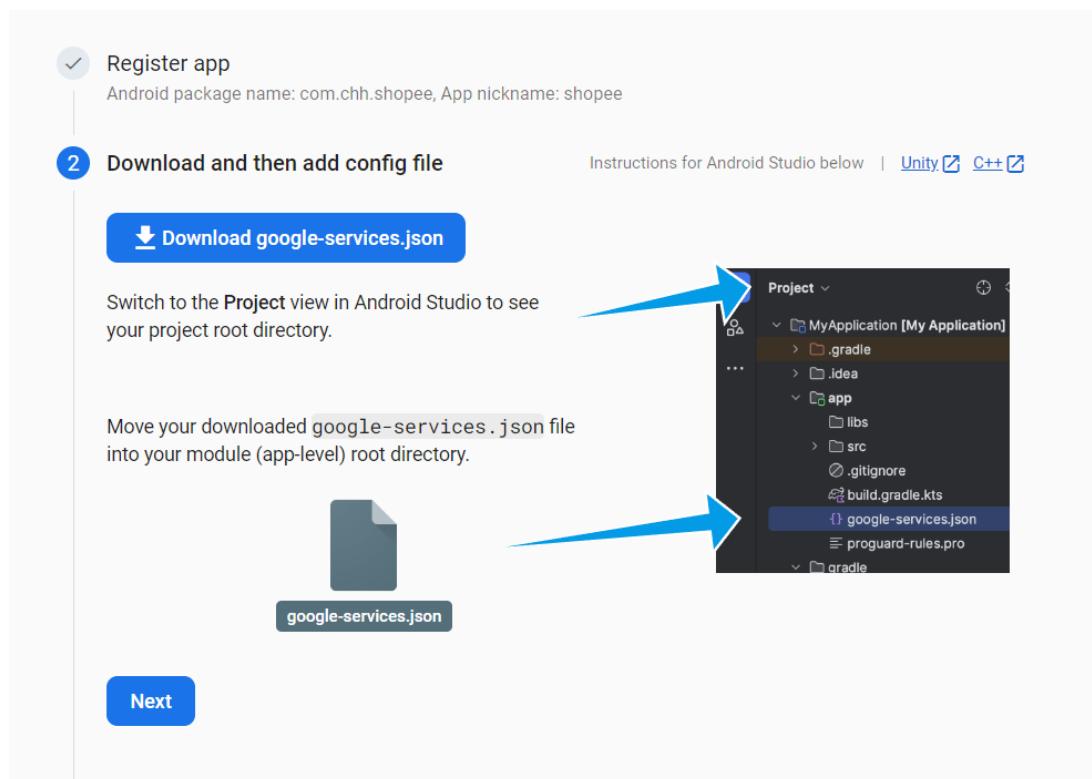
2.5.3. Kết nối Android với database

- Một cửa sổ mới được hiện lên “Add Firebase to your Android app” yêu cầu chúng ta thực hiện 4 bước:
 - + Bước 1: Bao gồm điền thông tin “Android package name” (bắt buộc) ở mục này chúng ta cần điền chính xác thư mục app android của mình thường là com.example.<tên app của bạn> trong app sàn thương mại điện tử của em là com.chh.shoponline, ở mục “App nickname” (không bắt buộc) thì bạn đặt tên cho app của mình và mục cuối cùng “Debug signing certificate SHA-1” (không bắt buộc) mã SHA-1 bạn có thể lấy ở phần gradle trong app android của mình giúp bạn debug hiệu quả hơn. Sau đó nhấn “Register app”.



Hình 16. Hình Add Firebase to your Android app

- + Bước 2: Click vào Download google-services.json về máy tính, sau đó chỉnh cấu trúc project nằm ở góc trên bên trái màn hình từ dạng android sang dạng project rồi copy file .json vừa tải xuống lưu vào thư mục src, sau đó click “Next” ở bên trang web của firebase để sang bước 3.



Hình 17. Tải và cấu hình file google-services.json

- + Bước 3: Sau khi đã cài xong file google-services.json vào project thì chúng ta cần khai báo cho chương trình biết và phụ thuộc vào nó bằng cách vào thư mục build.gradle.kts sẽ có 2 file cùng tên chúng ta chọn file dạng <project>/build.gradle.kts và thêm dòng code sau: “id("com.google.gms.google-services") version "4.4.0" apply false” nó được đặt trong hàm plugins.

```

plugins {
    // ...

    // Add the dependency for the Google services Gradle plugin
    id("com.google.gms.google-services") version "4.4.0" apply false
}

```

Hình 18. vị trí đặt dòng code khai báo google-services

Trong file <app-module>/build.gradle.kts tương tự ta thêm những dòng code sau để chương trình nhận biết sau đó ta click vào “Sync now” để chương trình tải thêm những phần liên quan cần thiết đến cấu hình firebase SDK.

```

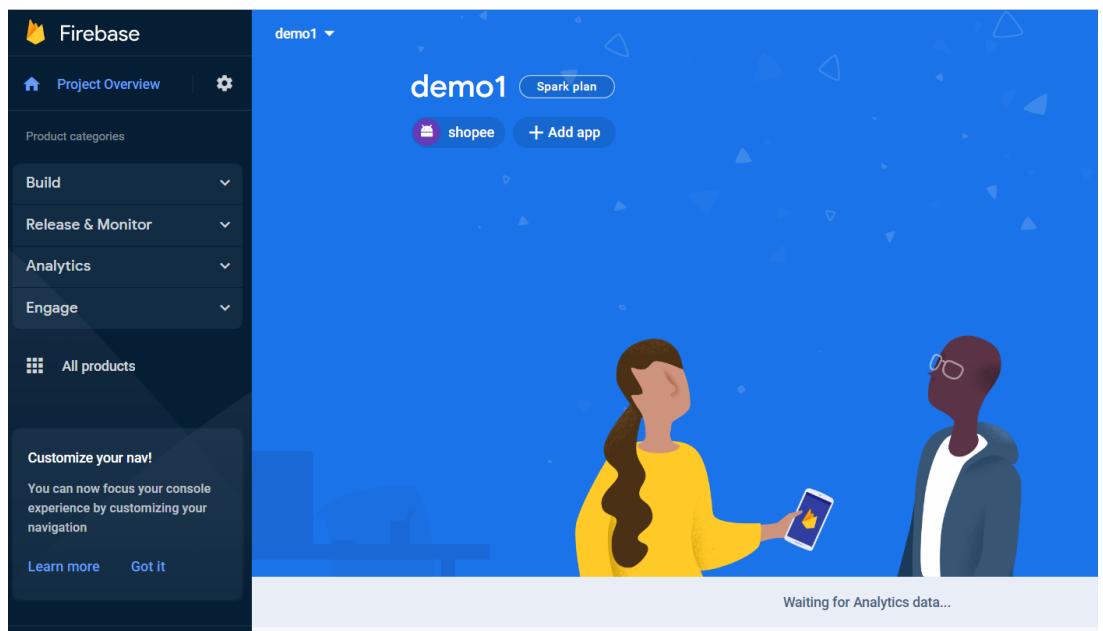
plugins {
    id("com.android.application")
    // Add the Google services Gradle plugin
    id("com.google.gms.google-services")
    ...
}

dependencies {
    // Import the Firebase BoM
    implementation(platform("com.google.firebase:firebase-bom:32.3.1"))
}

```

Hình 19. Cấu hình firebase SDK trong module gradle.build

- + Bước 4: Firebase muốn hỏi bạn có muốn xem thêm thông tin hoặc ví dụ để hiểu hơn về firebase nếu không quan tâm bạn có thể nhấn “Continue to Console”.



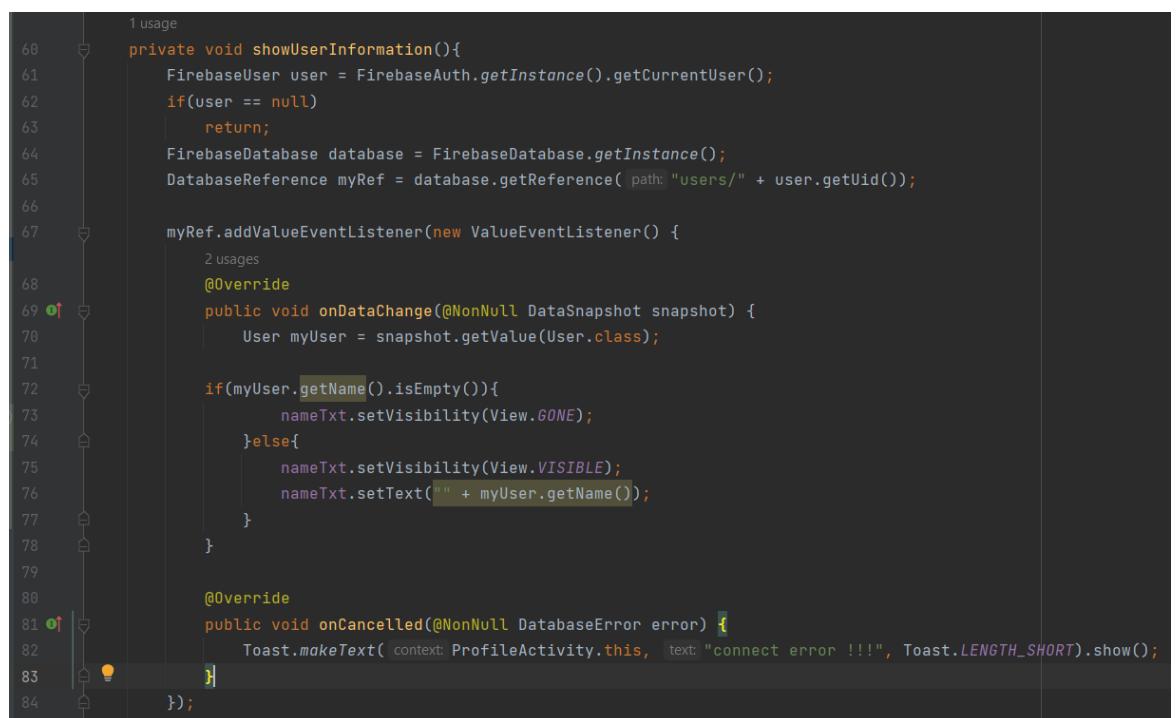
Hình 19. Giao diện console khi bạn kết nối với firebase

2.5.4. Khởi tạo Firebase trong ứng dụng

Trong một class bất kỳ trong thư mục Activity ví dụ như: MainActivity.class thì ta chỉ cần “import com.google.firebaseio.FirebaseApp”, đồng thời trong phương thức onCreate() của MainActivity ta thêm dòng lệnh sau “FirebaseApp.initializeApp(this); // khởi tạo firebase”

2.5.5. Sử dụng Firebase trong ứng dụng

Bây giờ bạn đã kết nối Firebase với ứng dụng Android của mình việc còn lại là bạn sử dụng các dịch vụ mà Firebase hỗ trợ như: Firebase Realtime database, Firebase Authentication, Firebase Cloud Messaging, vv..., ví dụ bạn có thể dùng Firebase Authentication thì thêm code như sau:



```
1 usage
60     private void showUserInformation(){
61         FirebaseAuth user = FirebaseAuth.getInstance().getCurrentUser();
62         if(user == null)
63             return;
64         FirebaseDatabase database = FirebaseDatabase.getInstance();
65         DatabaseReference myRef = database.getReference( path: "users/" + user.getUid());
66
67         myRef.addValueEventListener(new ValueEventListener() {
68             2 usages
69             @Override
70             public void onDataChange(@NonNull DataSnapshot snapshot) {
71                 User myUser = snapshot.getValue(User.class);
72
73                 if(myUser.getName().isEmpty()){
74                     nameTxt.setVisibility(View.GONE);
75                 }else{
76                     nameTxt.setVisibility(View.VISIBLE);
77                     nameTxt.setText(" " + myUser.getName());
78                 }
79
80             @Override
81             public void onCancelled(@NonNull DatabaseError error) {
82                 Toast.makeText( context: ProfileActivity.this, text: "connect error !!!", Toast.LENGTH_SHORT).show();
83             }
84         });
85     }
```

Hình 20. Hàm sử dụng FirebaseAuth

2.6. Cấu trúc tổ chức file trong Android app

2.6.1. Các Class Activity

Trong Android, mỗi màn hình giao diện mà người dùng tương tác được đại diện bởi một hoạt động của họ được gọi “Activity”, trong trường hợp này em xin được ví dụ MainActivity.class:

- + Các Activity đều kế thừa AppCompatActivity bởi vì AppCompatActivity là một Class trong AndroidX, nó cung cấp cho app của chúng ta nhiều chức năng tiện ích , hỗ trợ tính năng hành vi tương thích với phiên bản Android cũ hơn, tùy chỉnh thanh điều hướng, thanh trạng thái hoạt động, vv...

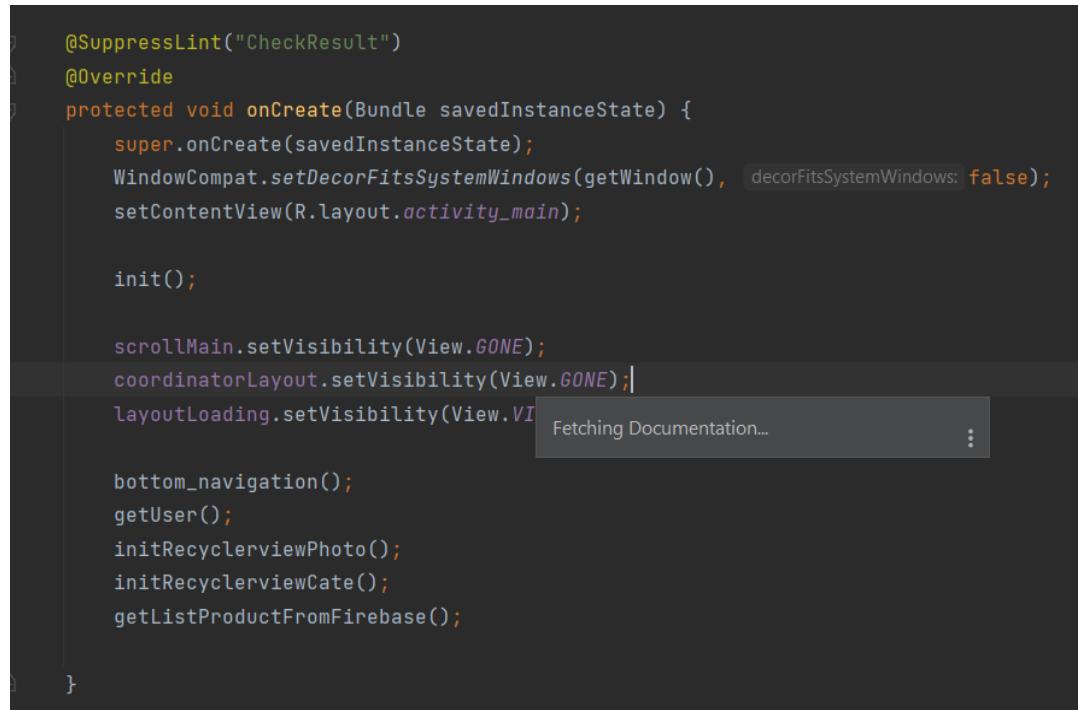
```
package com.chh.shoponline.Activity;

import ...

public class MainActivity extends AppCompatActivity {
    2 usages
    private static final long AUTO_SCROLL_DELAY = 3000;
    4 usages
    private RecyclerView.Adapter adapterPopular, adapterPopular2, adapterPopularCategory;
    3 usages
    private RecyclerView recyclerViewPopular, recyclerViewPopular2, recyclerViewPopularCategory;
    7 usages
    private ViewPager2 mViewPager2;
    2 usages
    private CircleIndicator3 mCircleIndicator3;
    2 usages
    private List<Photo> mListPhoto;
    1 usage
    private ArrayList<Category> cateList = getListCategory();
    2 usages
    private LinearLayout homeBtn, cartBtn, profileBtn, wishlistBtn;
    3 usages
    private ImageView redImage;
    4 usages
    private TextView numberCart;
    3 usages
    private ConstraintLayout layoutLoading;
    3 usages
    private ScrollView scrollMain;
    3 usages
```

Hình 21. MainActivity kế thừa AppCompatActivity

- + Trong phương thức onCreate() có dòng lệnh
“WindowCompat.setDecorFitsSystemWindows(getWindow(), false)” nó giúp cho thông tin hiển thị tràn màn hình để tăng diện tích cung cấp thông tin cho người dùng.



```
@SuppressLint("CheckResult")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    WindowCompat.setDecorFitsSystemWindows(getWindow(), decorFitsSystemWindows: false);
    setContentView(R.layout.activity_main);

    init();

    scrollMain.setVisibility(View.GONE);
    coordinatorLayout.setVisibility(View.GONE);
    layoutLoading.setVisibility(View.VISIBLE); // Tooltip: Fetching Documentation...
    bottom_navigation();
    getUser();
    initRecyclerviewPhoto();
    initRecyclerviewCart();
    getListProductFromFirebase();
}

}
```

Hình 22. Phương thức onCreate()

- + Hàm init() hàm này được gọi khi MainActivity bắt đầu chạy nó sẽ thực hiện nhiệm vụ ánh xạ lên những phần tử trên giao diện như: Button, TextView, RecyclerView, vv...,



```
1 usage
private void init(){
    mViewPager2 = findViewById(R.id.viewPager);
    mCircleIndicator3 = findViewById(R.id.circle_indicator);
    homeBtn = findViewById(R.id.homeBtn);
    cartBtn = findViewById(R.id.cartBtn);
    profileBtn = findViewById(R.id.profile);
    redImage = findViewById(R.id.redImage);
    numberCart = findViewById(R.id.numberCart);
    layoutLoading = findViewById(R.id.layoutLoading);
    scrollMain = findViewById(R.id.scrollMain);
    coordinatorLayout = findViewById(R.id.coordinatorLayout);
    wishlistBtn = findViewById(R.id.wishlistBtn);
}
```

Hình 23. Phương thức init()

- + Hàm bottom_navigation() có chức năng điều hướng từ MainActivity đến các Class Activity khác tùy theo nhu cầu người sử dụng.

```
1 usage
private void bottom_navigation() {
    homeBtn.setOnClickListener(view -> startActivity(new Intent(packageContext: MainActivity.this, MainActivity.class)));
    cartBtn.setOnClickListener(view -> {
        Intent intent = new Intent(packageContext: MainActivity.this, CartActivity.class);
        startActivity(intent);
    });
    profileBtn.setOnClickListener(view -> startActivity(new Intent(packageContext: MainActivity.this, ProfileActivity.class)));
    wishlistBtn.setOnClickListener(view -> startActivity(new Intent(packageContext: MainActivity.this, WishlistActivity.class)));
}
```

Hình 24. Phương thức bottom_navigation()

- + Hàm initRecyclerview() có chức năng nhận vào một list đối tượng như listProduct, listUser, và render (trình chiếu) ra giao diện cũng như lắng nghe sự kiện thay đổi trên list đối tượng liền cập nhật ra giao diện người dùng.

```
2 usages
private void initRecyclerview() {
    recyclerViewPopular = findViewById(R.id.view1);
    recyclerViewPopular.setLayoutManager(new LinearLayoutManager(
        context: this, LinearLayoutManager.HORIZONTAL, reverseLayout: false));
    adapterPopular = new PopularListAdapter(items);
    recyclerViewPopular.setAdapter(adapterPopular);

    if (myUser == null || myUser.getNumCart() == 0) {
        numberCart.setVisibility(View.GONE);
        redImage.setVisibility(View.GONE);
    } else {
        numberCart.setText(""+ myUser.getNumCart());
        numberCart.setVisibility(View.VISIBLE);
        redImage.setVisibility(View.VISIBLE);
    }
}
```

Hình 25. Phương thức initRecyclerview()

- + Hàm getUser() và getListProductFromFirebase() dùng để lấy dữ liệu từ Firebase cũng như lắng nghe sự thay đổi phía database rồi thông báo cho

app để điều chỉnh kết hợp với hàm initRecyclerview() cập nhật lại giao diện người dùng.

```
↑ usage
private void getListProductFromFirebase(){
    firebase.addObserver(new Observer<ArrayList<PopularDomain>>() {
        @Override
        public void onSubscribe(@io.reactivex.rxjava3.annotations.NonNull Disposable d) {}

        @Override
        public void onNext(ArrayList<PopularDomain> productList) {
            // lay list product
            items = productList;

            //cap nhat giao dien
            initRecyclerview();
            initRecyclerview2();
            adapterPupolar.notifyDataSetChanged();
        }

        @Override
        public void onError(Throwable e) {
            // Xử lý lỗi
        }

        @Override
        public void onComplete() {
            ...
        }
    });

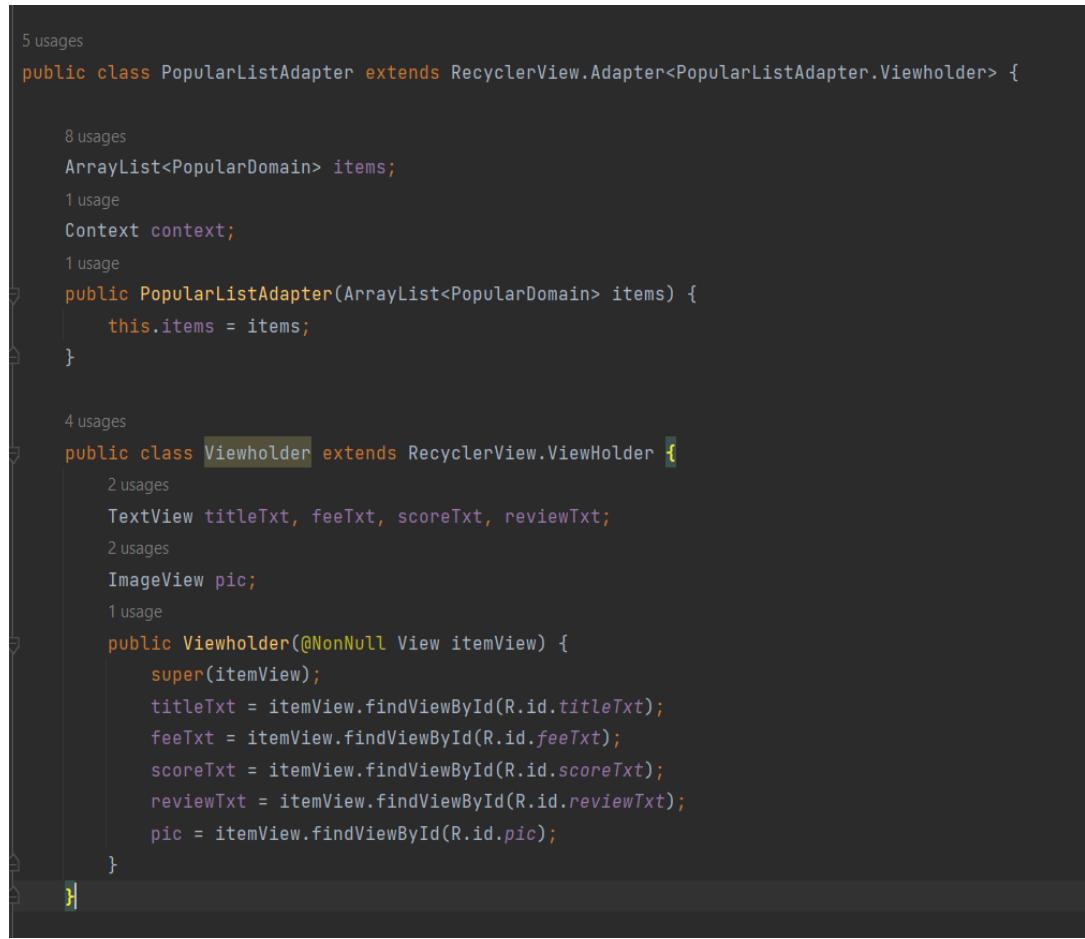
    firebase.fetchProductListFromFirebase().subscribe();
}
```

Hình 26. Phương thức getListProductFromFirebase()

2.6.2. Các Class Adapter

Ví dụ sử dụng PopularListAdapter.class ,Class này được dùng để kết nối dữ liệu với RecyclerView. Nó cung cấp các phương thức bao gồm: để tạo ViewHolder, gắn dữ liệu cho ViewHolder và cung cấp thông tin về số lượng phần tử của danh sách đối tượng. Để sử dụng ta cần triển khai các phương thức cần thiết như onCreateViewHolder() và onBindViewHolder() để có thể cấu hình lại cách dữ liệu của chúng ta muốn hiển thị ra giao diện người dùng.

- + Các class Adapter kế thừa RecyclerView.Adapter<T> nó giúp ta định nghĩa các phương thức cần thiết để quản lý dữ liệu và tạo các viewHolder. viewHolder là lớp trong PopularListAdapter.class dùng để lưu trữ các tham chiếu đến các phần tử giao diện người dùng mà RecyclerView sẽ hiển thị. Các ViewHolder này đều phải kế thừa RecyclerView.ViewHolder.



```
5 usages
public class PopularListAdapter extends RecyclerView.Adapter<PopularListAdapter.ViewHolder> {

    8 usages
    ArrayList<PopularDomain> items;
    1 usage
    Context context;
    1 usage
    public PopularListAdapter(ArrayList<PopularDomain> items) {
        this.items = items;
    }

    4 usages
    public class ViewHolder extends RecyclerView.ViewHolder {
        2 usages
        TextView titleTxt, feeTxt, scoreTxt, reviewTxt;
        2 usages
        ImageView pic;
        1 usage
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            titleTxt = itemView.findViewById(R.id.titleTxt);
            feeTxt = itemView.findViewById(R.id.feeTxt);
            scoreTxt = itemView.findViewById(R.id.scoreTxt);
            reviewTxt = itemView.findViewById(R.id.reviewTxt);
            pic = itemView.findViewById(R.id.pic);
        }
    }
}
```

Hình 27. Hình ảnh PopularListAdapter.class

- + Phương thức onCreateViewHolder() được gọi khi RecyclerView tạo một ViewHolder mới dựa trên layout đã được thiết kế trước như “viewholder_pop_list.xml” (đây là giao diện được thiết kế cho một phần tử như: một danh mục, một user hoặc một sản phẩm).
- + Phương thức onBindViewHolder() được gọi khi RecyclerView cần gắn dữ liệu lên ViewHolder. Nó cập nhật các view trong ViewHolder với dữ liệu từng mục cụ thể. Ngoài ra còn có thể lắng nghe sự kiện click vào từng item trong trường hợp của PopularListAdapter.class thì phương thức này lắng nghe sự kiện click vào từng item sản phẩm thì chuyển sang trang DetailActivity.class và truyền dữ liệu của item đó cho DetailActivity.class nhận và xuất ra giao diện.

- + Phương thức getItemCount() trả về số lượng đối tượng có trong mảng để cho RecyclerView sử dụng giá trị này để biết có bao nhiêu mục cần được hiển thị.

```

@NonNull
@Override
public PopularListAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View inflate = LayoutInflater.from(parent.getContext()).inflate(R.layout.viewholder_pop_list, parent, attachToRoot: false);
    context = parent.getContext();

    return new ViewHolder(inflate);
}

@Override
public void onBindViewHolder(@NonNull PopularListAdapter.ViewHolder holder, int position) {
    holder.titleTxt.setText(items.get(position).getTitle());
    holder.priceTxt.setText("$" + items.get(position).getPrice());
    holder.scoreTxt.setText(" " + items.get(position).getScore());
    holder.reviewTxt.setText(" " + items.get(position).getReview());

    Glide.with(holder.itemView.getContext()).load(items.get(position).getPicUrl()).error(R.drawable.pic1).into(holder.pic);

    holder.itemView.setOnClickListener(view -> {
        Intent intent = new Intent(holder.itemView.getContext(), DetailActivity.class);
        intent.putExtra("object", items.get(position));
        holder.itemView.getContext().startActivity(intent);
    });
}

@Override
public int getItemCount() {
    return items.size();
}
}

```

Hình 28. Các phương thức ghi đè của PopularListAdapter

- + Trường hợp đặt biệt đối tượng trong list danh sách có thuộc tính hình ảnh thì ta sẽ dùng thư viện Glide thư viện này có thể load được ảnh từ link Url kiểu String hoặc chuyển đổi ảnh thành số kiểu Integer.

2.6.3. Các Class hướng đối tượng

Các Class này giúp ta hiện thực hóa đối tượng ngoài đời thực như User, Product, bao gồm các thuộc tính như: int id, String name, Double price, vv..., phương thức khởi tạo không tham số và có tham số để khởi tạo đối tượng và cuối cùng là getter, setter nhằm mục đích đọc dữ liệu.

```

22 usages
public class User {
    3 usages
    private String id;
    3 usages
    private String address;
    3 usages
    private String credit;
    3 usages
    private String name;
    3 usages
    private String role;
    2 usages
    private int numCart = 0;
    2 usages
    public User(){
    }

    1 usage
    public User(String id, String address, String credit, String name, String role) {
        this.id = id;
        this.address = address;
        this.credit = credit;
        this.name = name;
        this.role = role;
    }

    public String getId() { return id; }

    public void setId(String id) { this.id = id; }
}

```

Hình 29. Hình ảnh cấu trúc Class User

2.6.4. Các Class kết nối Database

Trong đồ án này thì sử dụng công nghệ của Firebase cụ thể là để kết nối với database là Class FirebaseManager.class bao gồm các thuộc tính khởi tạo FirebaseDatabase giúp khởi tạo kết nối xuống firebase, FirebaseAuth lấy người dùng hiện tại đang đăng nhập, DatabaseReference tham chiếu đến bảng dữ liệu mà ta cần.

- + Firebase Realtime Database: do đây là dữ liệu thời gian thực nên khi chạy chương trình cần một khoảng thời gian delay để load database về để xử lý vấn đề bất đồng bộ này em đã dùng thư viện RxJava là Observable có công việc tạo kết nối xuống database và load dữ liệu, đăng ký một tuyến đường và Observer sẽ nhận lấy tuyến đường ấy và chờ tính hiệu từ Observable

```
//lay list product cua cart
1 usage
public Observable<ArrayList<PopularDomain>> fetchCartListFromFirebase() {
    return Observable.create(emitter -> {
        DatabaseReference myRef = database.getReference( path: "users/" + auth.getUid() + "/carts");
        myRef.addValueEventListener(new ValueEventListener() {
            2 usages
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                ArrayList<PopularDomain> productList = new ArrayList<>();
                for (DataSnapshot dataSnapshot : snapshot.getChildren()){
                    PopularDomain popularDomain = dataSnapshot.getValue(PopularDomain.class);
                    productList.add(popularDomain);
                }
                notifyObservers(productList);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                }
            });
        });
    }
}

@NonNull
DatabaseError error

error – A description of the error that occurred
shoponline.app.main
:
```

Hình 30. Sử dụng Observable để load database

- + Khi đã Observable load xong sẽ gửi một tín hiệu thông báo cho Observer biết và Observer sẽ ngắt kết nối để tránh rò rỉ bộ nhớ và thông báo cho chương trình đã load xong data và xuất ra giao diện người dùng. Ngoài ra Observable còn hỗ trợ nhiều phương thức khác rất rõ ràng theo từng giai đoạn của data như: trước khi tải, trong khi tải, sau khi tải, nếu gặp trực trặc lỗi để lập trình viên có thể dễ xử lý.

```

1 usage
private void getListProductFromFirebase(){
    firebase.addObserver(new Observer<ArrayList<PopularDomain>>() {
        @Override
        public void onSubscribe(@io.reactivex.rxjava3.annotations.NonNull Disposable d) {}

        @Override
        public void onNext(ArrayList<PopularDomain> productList) {
            // lay list product
            items = productList;

            //cap nhat giao dien
            initRecyclerview();
            initRecyclerview2();
            adapterPupolar.notifyDataSetChanged();
        }

        @Override
        public void onError(Throwable e) {
            // Xử lý lỗi
        }

        @Override
        public void onComplete() {
        }
    });

    firebase.fetchProductListFromFirebase().subscribe();
}

```

Hình 31. Dùng Observer để nhận data từ Observable

Chương 3. XÂY DỰNG HỆ THỐNG APP MOBILE SÀN THƯƠNG MẠI

3.1. Giới thiệu hệ thống

Đây là một hệ thống app android sàn thương mại điện tử, hỗ trợ kết nối giữa các doanh nghiệp và người tiêu dùng lại với nhau thông qua mạng Internet. Cụ thể hơn app giúp các doanh nghiệp có thể tiếp cận được nhiều khách hàng cũng như quảng bá sản phẩm của mình, còn người dùng có thể dễ dàng mua sắm và tìm kiếm những đồ dùng mà mình yêu thích chỉ bằng vài cú click cũng như tối ưu chi phí khi mua sắm và cải thiện chất lượng sản phẩm.

3.2. Phân tích và thiết kế hệ thống

3.2.1.1. Xác định yêu cầu

Để thiết kế một app sàn thương mại điện tử nhóm em đã phân tích được hệ thống cần những yêu cầu sau:

- + Hoạt động xoay quanh 3 đối tượng: người dùng, Admin, doanh nghiệp.

- + Đăng nhập/đăng ký (có upload avatar).
- + Tìm kiếm sản phẩm (tên/giá/danh mục/vv...).
- + Thanh toán (tiền mặt/online).
- + Giỏ hàng giúp cho người dùng có thể lưu những sản phẩm mà mình đã đặt hàng.
- + Trò chuyện trực tuyến realtime.
- + Thêm sản phẩm yêu thích.
- + Bình luận, đăng ảnh và đánh giá sản phẩm đã mua.
- + Doanh nghiệp có thể đăng sản phẩm muốn bán.
- + Lấy vị trí địa lý hiện tại.
- + Admin quản lý người dùng và xét duyệt đăng ký doanh nghiệp.

3.2.2. Phân tích yêu cầu

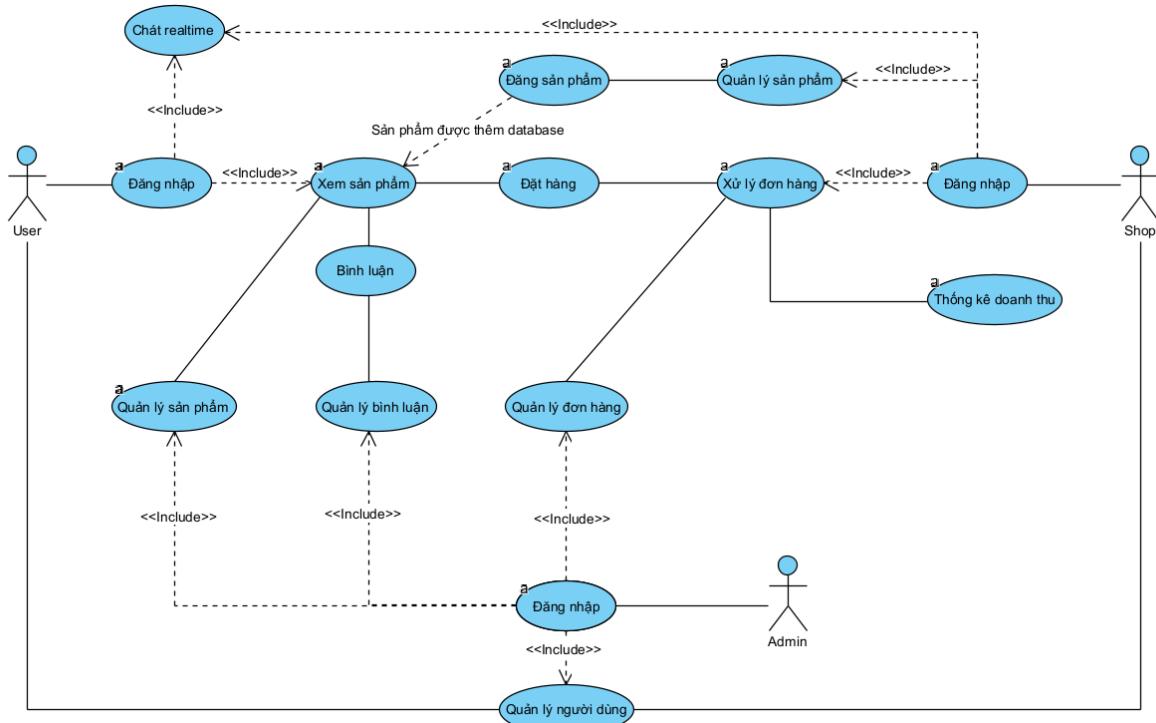
Để thực hiện hết những yêu cầu trên nhóm em đã tổng hợp lại những chức năng sau:

- + Khi đăng nhập sẽ xác định role của đối tượng đang đăng nhập tương đương họ sẽ có những quyền và nghĩa vụ riêng cũng như giao diện cũng khác nhau.
- + Khi muốn mua sản phẩm hoặc các chức năng của app thì phải đăng nhập.
- + Tìm kiếm sản phẩm (tên/giá/danh mục/vv...).
- + Thanh toán (tiền mặt/online).
- + Giỏ hàng giúp cho người dùng có thể lưu những sản phẩm mà mình đã đặt hàng ngoài ra còn có thể mua nhiều sản phẩm của nhiều cửa hàng khác nhau.
- + Trò chuyện trực tuyến giữa shop và người dùng để giải đáp thắc mắc và trao đổi thông tin sản phẩm.
- + Thêm sản phẩm yêu thích vào wishlist của mình.

- + Bình luận có thể đăng ảnh và kèm đánh giá sao từ 1 đến 5 sao sản phẩm đã mua.
- + Doanh nghiệp có thể đăng sản phẩm muốn bán bao gồm thông tin, giá cả, hình ảnh, giá, vv..., theo dõi đơn hàng khách đã đặt và thống kê doanh số sản phẩm.
- + Lấy vị trí địa lý hiện tại dựa theo google map.
- + Admin quản lý người dùng, thêm sửa xóa thông tin của họ, quản lý doanh nghiệp, xét duyệt cho đăng ký doanh nghiệp, thêm sửa xóa thông tin của họ, quản lý đơn hàng.
- + Người dùng ngoài những chức năng trên họ có thể thay đổi thông tin cá nhân.

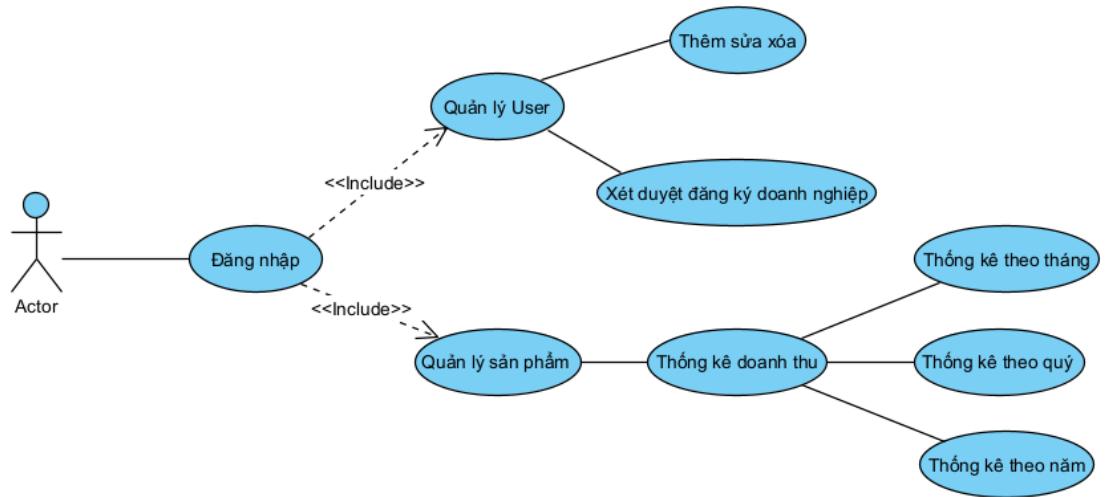
3.2.3. Sơ đồ use case

Use case Tổng quan



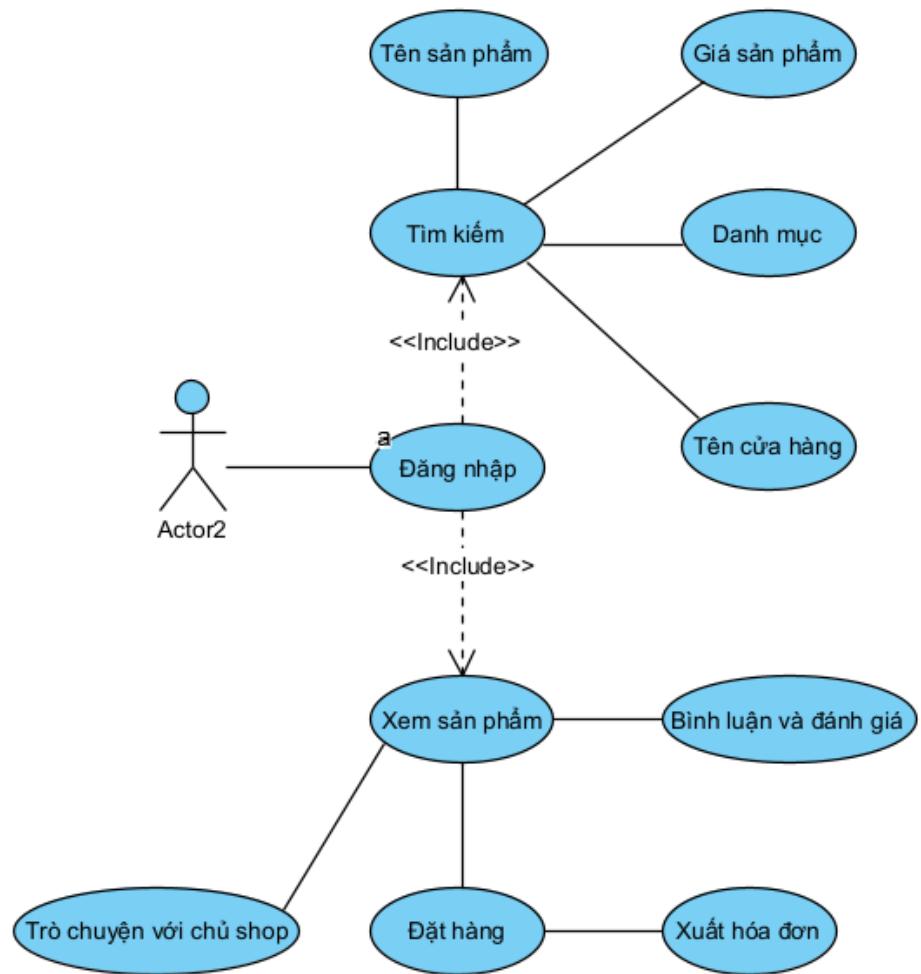
Hình 32. Sơ đồ use case tổng quan

Use case admin



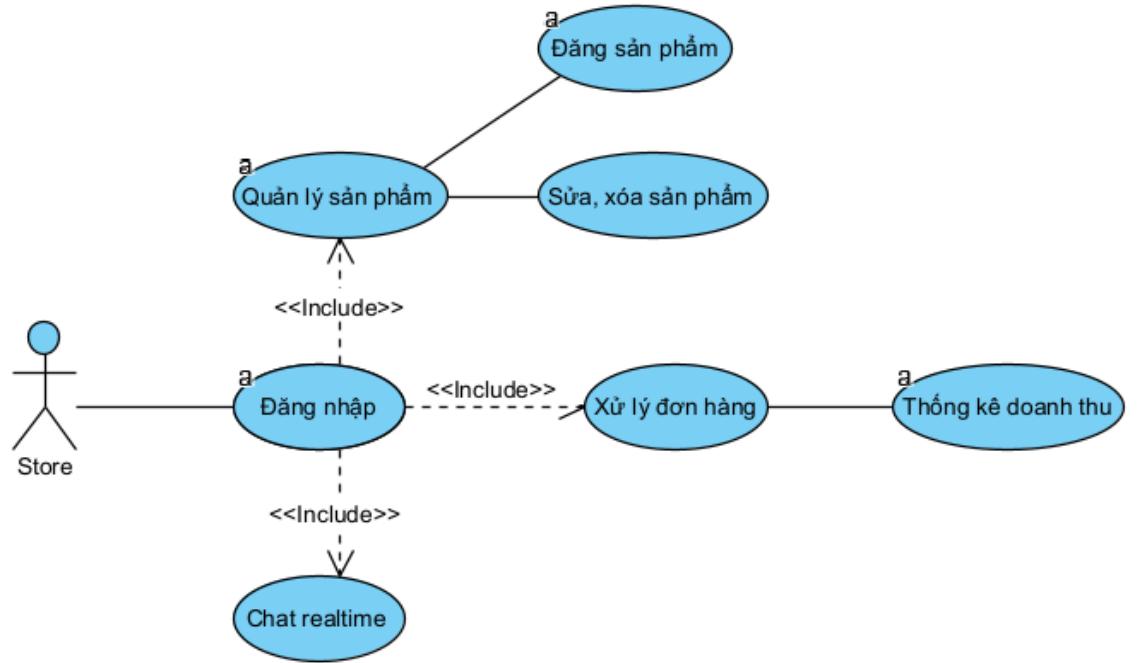
Hình 33. Sơ đồ use case admin

Use case khách hàng



Hình 34. sơ đồ use case cửa hàng

Use case cửa hàng



Hình 35. Sơ đồ use case cửa hàng

3.2.4. Đặc tả use case

Use case quản lý user

Mô tả	Ở Use case này admin thêm, xóa, sửa thông tin tài khoản người dùng
Actor chính	Admin
Actor phụ	Không có
Trước điều kiện	Đăng nhập thành công với tài khoản có role admin
Hậu điều kiện	Không có
Luồng hoạt động	1) Người dùng đăng nhập vào hệ thống với quyền admin 2) Hệ thống kiểm tra thông tin đăng

	nhập 3) Sau khi đăng nhập chuyển đến giao diện trang chủ 4) Sau đó chọn chức năng xem danh sách user 5) Người dùng thực hiện chức năng thêm, xóa, sửa user tùy theo yêu cầu 6) Người dùng duyệt và cấp quyền cho những user đăng ký kinh doanh
Luồng thay thế	Ở bước 4 nếu người dùng đã đăng nhập không phải role admin thì không cho truy cập
Luồng ngoại lệ	Không có

Use case đăng ký người dùng

Mô tả	Use case này cho phép người dùng đăng ký tài khoản mới để sử dụng các dịch vụ và chức năng của app
Actor chính	Người dùng (khách hàng)
Actor phụ	Không có
Tiền điều kiện	Không có
Hậu điều kiện	Không có
Luồng hoạt động	1) Khách hàng truy cập vào hệ thống 2) Sau đó chọn chức năng đăng ký

	<p>người dùng</p> <p>3) Người dùng nhập thông tin đăng ký</p> <p>4) Người dùng chọn Đăng ký</p>
Luồng thay thẻ	Ở bước 3 nếu người dùng nhập thiếu hoặc không hợp lệ chương trình sẽ bắt nhập lại
Luồng ngoại lệ	Không có

Use case thống kê và báo cáo doanh thu

Mô tả	Use case này cho phép admin có thể xem các số liệu báo cáo và thống kê doanh thu từ các shop
Actor chính	Admin, cửa hàng(shop)
Actor phụ	Không có
Tiền điều kiện	Người dùng đã đăng nhập thành công với quyền admin hoặc shop
Hậu điều kiện	Không có
Luồng hoạt động	<p>1) Người dùng đăng nhập vào hệ thống với quyền admin hoặc shop</p> <p>2) Hệ thống kiểm tra thông tin đăng nhập</p> <p>3) Sau khi đăng nhập chuyển đến giao diện trang chủ</p> <p>4) Người dùng chọn chức năng xem</p>

	thống kê báo cáo
Luồng thay thế	Ở bước 4 nếu người dùng đã đăng nhập với vai trò không phải là admin hoặc shop thì báo lỗi và không cho truy cập
Luồng ngoại lệ	Không có

Use case tìm kiếm

Mô tả	Use case này cho phép khách hàng có thể tìm kiếm các sản phẩm mà mình cần dựa trên các tên, giá, danh mục, vv...
Actor chính	Khách hàng, shop, admin
Actor phụ	Không có
Trước điều kiện	Không có
Hậu điều kiện	Không có
Luồng hoạt động	1) Người dùng truy cập vào hệ thống 2) Người dùng nhập thông tin cần tìm kiếm vào ô tìm kiếm 3) Hệ thống lọc ra kết quả và trả về cho người dùng những sản phẩm đúng nhất và sản phẩm tương tự
Luồng thay thế	Ở bước 2 nếu khách hàng nhập vào thông tin không hợp lệ thì sẽ không

	tìm kiếm nếu không tìm thấy sản phẩm thì báo trống
Luồng ngoại lệ	Không có

Use case bình luận và đánh giá

Mô tả	Use case này cho phép khách hàng đánh giá từ 1 đến 5 sao, bình luận sản phẩm của shop
Actor chính	Khách hàng
Actor phụ	Không có
Trước điều kiện	Người dùng đã đăng nhập thành công với quyền user
Hậu điều kiện	Không có
Luồng hoạt động	1) Người dùng truy cập vào hệ thống 2) Người dùng chọn xem chi tiết shop hoặc sản phẩm yêu thích 3) Người dùng tiến hành nhập bình luận, số sao và hình ảnh rồi nhấn gửi
Luồng thay thế	Ở bước 3 nếu người dùng chưa đăng nhập thì ở giao diện Bình luận sẽ trở thành giao diện Yêu cầu đăng nhập
Luồng ngoại lệ	Không có

Use case đặt hàng

Mô tả	Use case này cho phép khách hàng có thể đặt hàng bằng cách chọn lựa các sản phẩm của mình cho vào trong giỏ hàng
Actor chính	Khách hàng
Actor phụ	Không có
Trước điều kiện	Người dùng đã đăng nhập thành công với quyền user
Hậu điều kiện	Sau khi đặt hàng thành công hệ thống sẽ lưu xuống firebase
Luồng hoạt động	<p>1) Người dùng đăng nhập vào hệ thống với quyền user</p> <p>2) Người dùng chọn lựa các sản phẩm mà mình muốn</p> <p>3) Người dùng nhấn vào “add to cart”</p> <p>4) Người dùng có thể tùy chọn số lượng hoặc xóa các sản phẩm mà mình đã thêm</p> <p>5) Người dùng nhấn nút “Buy now”</p>
Luồng thay thế	Ở bước 3 nếu người dùng chưa đăng nhập thì ở giao diện “Add to cart” sẽ trở thành giao diện Yêu

	cầu đăng nhập
Luồng ngoại lệ	Ở bước 5 nếu người dùng không ấn đặt hàng thì giỏ hàng sẽ vẫn lưu các sản phẩm trong giỏ

Use case chat-realtime

Mô tả	Khách hàng và cửa hàng có thể trò chuyện trực tiếp
Actor chính	Khách hàng, người dùng cửa hàng
Actor phụ	Firebase
Trước điều kiện	Người dùng đã đăng nhập thành công với quyền shop hoặc user
Hậu điều kiện	Không có
Luồng hoạt động	1) Người dùng đăng nhập vào hệ thống với quyền shop hoặc user 2) Người dùng chọn chức năng trò chuyện 3) Người dùng nhập nội dung tin nhắn 4) Sau đó nhấn nút gửi
Luồng thay thế	Ở bước 1 nếu người dùng chưa đăng nhập thì bắt đăng nhập rồi mới được gửi tin nhắn
Luồng ngoại lệ	Không có

Use case thanh toán hóa đơn

Mô tả	Use case này cho phép người dùng được phép thanh toán hóa đơn sau khi đặt hàng
Actor chính	Admin
Actor phụ	Không có
Trước điều kiện	Người dùng đã đăng nhập thành công với quyền của khách hàng
Hậu điều kiện	Sau khi thanh toán thành công hệ thống sẽ gửi thông báo cho người dùng
Luồng hoạt động	<ul style="list-style-type: none"> 1) Người dùng đăng nhập vào hệ thống với quyền user 2) Người dùng chọn chức năng hóa đơn 3) Người dùng có thể chọn thanh toán trực tuyến hoặc thanh toán online 4) Hệ thống yêu cầu người dùng đăng nhập vào cổng thanh toán online 5) Người dùng đăng nhập vào cổng thanh toán online 6) Người dùng xác nhận hóa đơn cần thanh toán 7) Hệ thống gửi thông báo tương

	Ứng nếu thanh toán thành công hoặc thất bại
Luồng thay thế	Ở bước 2 nếu người dùng chưa đăng nhập thì sẽ không hiển thị
Luồng ngoại lệ	Không có

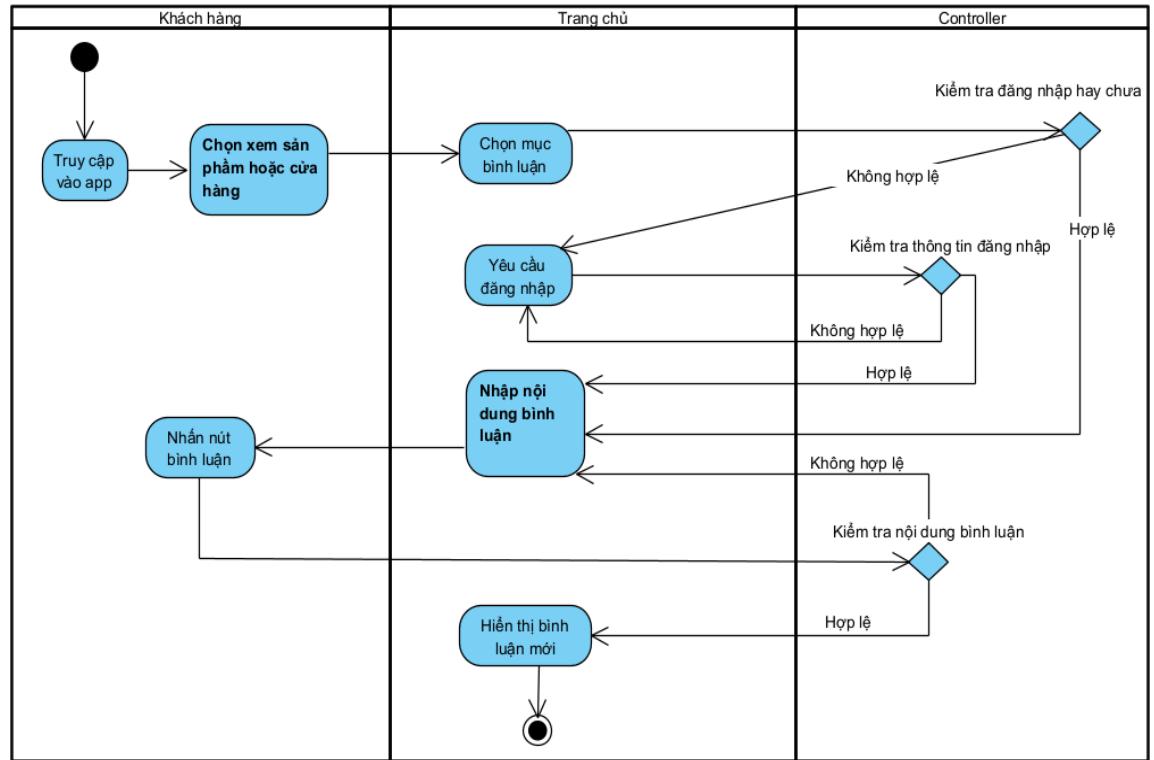
Use case đăng bán sản phẩm

Mô tả	Use case này cho phép shop đăng sản phẩm mới lên hệ thống
Actor chính	Chủ shop
Actor phụ	Không có
Trước điều kiện	Người dùng phải đăng nhập vào hệ thống với quyền shop
Hậu điều kiện	Không có
Luồng hoạt động	1) Người dùng đăng nhập vào hệ thống với quyền shop 2) Người dùng truy cập vào tính năng quản lý sản phẩm 3) Người dùng nhập thông tin sản phẩm cần đăng 4) Người dùng ấn Thêm sản phẩm
Luồng thay thế	Ở bước 3 nếu người dùng nhập sai hoặc thiếu thông tin sản phẩm thì hệ thống sẽ báo lỗi

Luồng ngoại lệ	Không có
----------------	----------

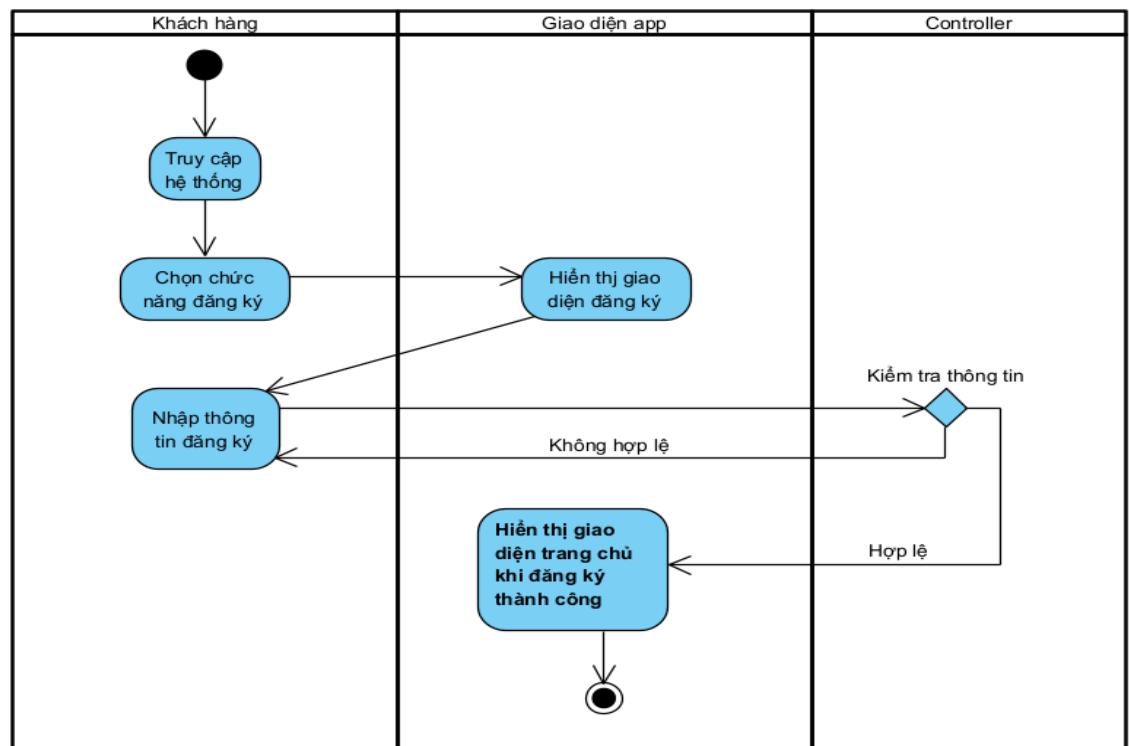
3.2.5. Activity Diagram

- Tính năng bình luận



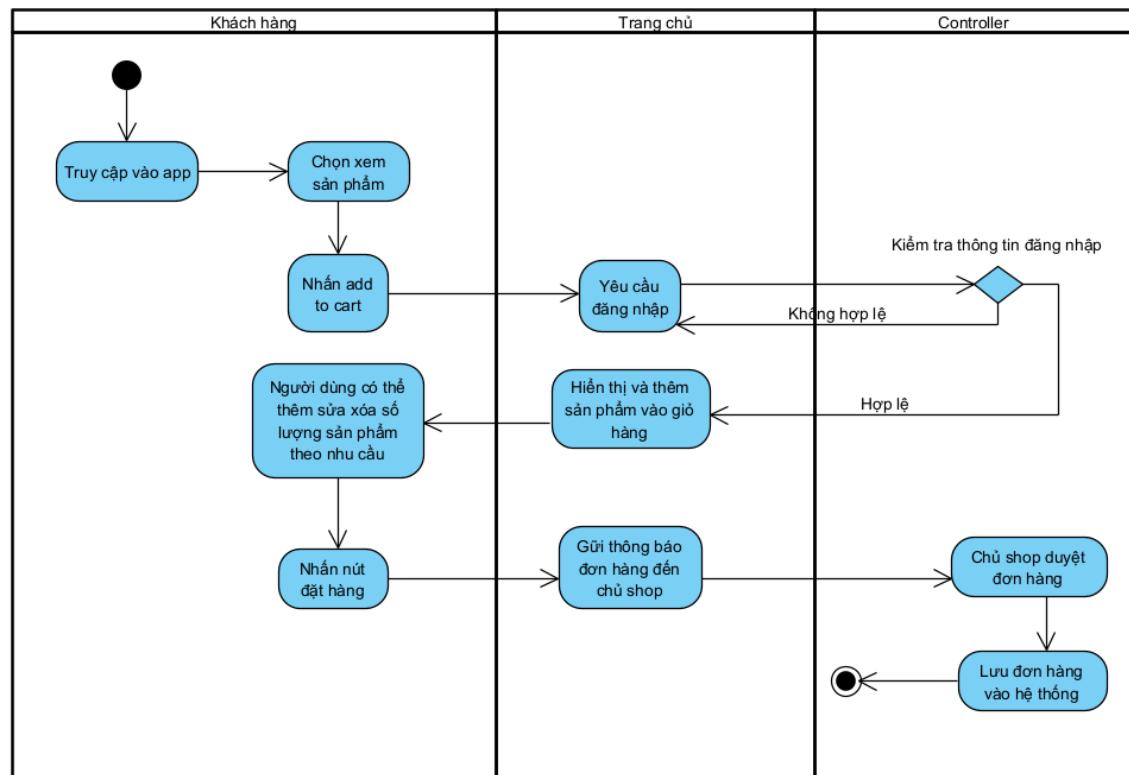
Hình 36. Mô tả tính năng bình luận bằng Activity Diagram

- Tính năng đăng ký



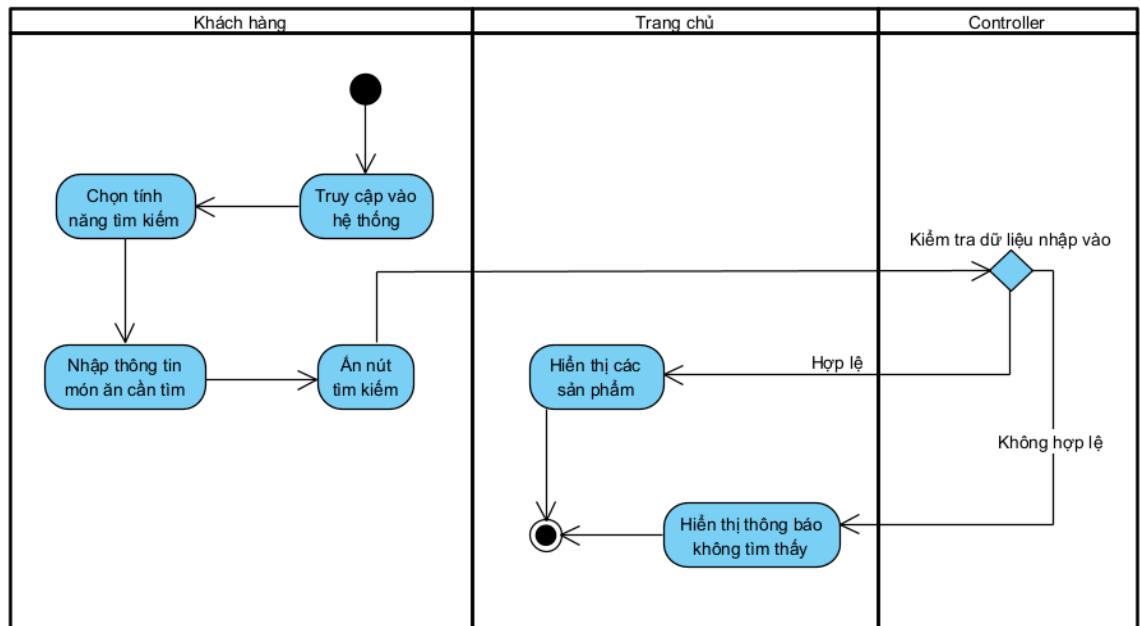
Hình 37. Mô tả tính năng đăng ký bằng Activity Diagram

- Tính năng đặt hàng



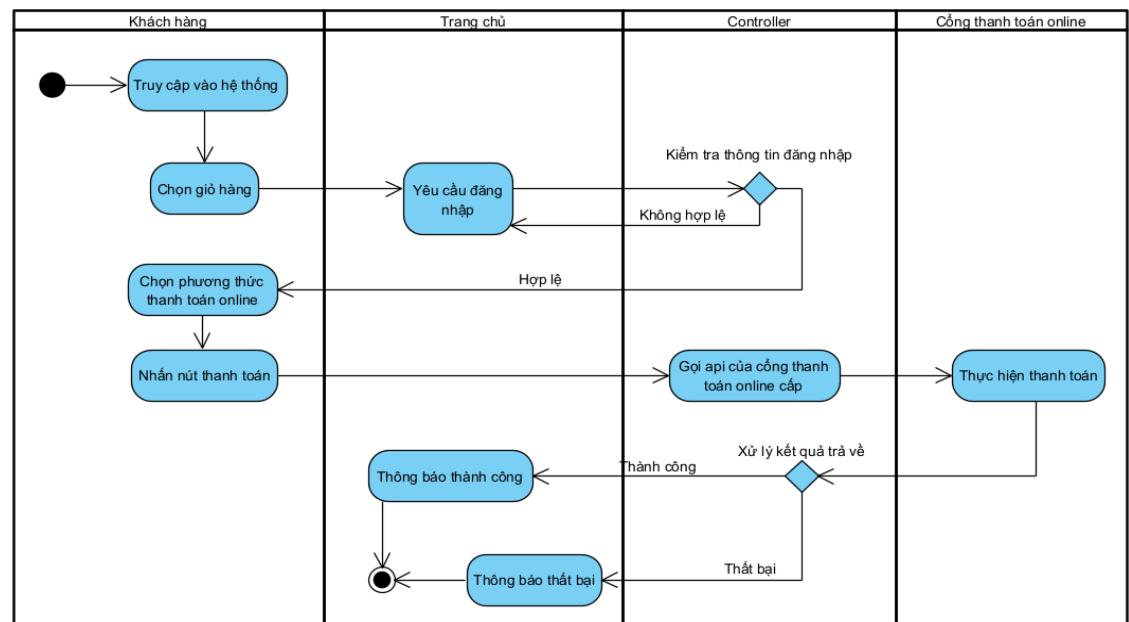
Hình 38. Mô tả tính năng đặt hàng bằng Activity Diagram

- Tính năng Tìm kiếm



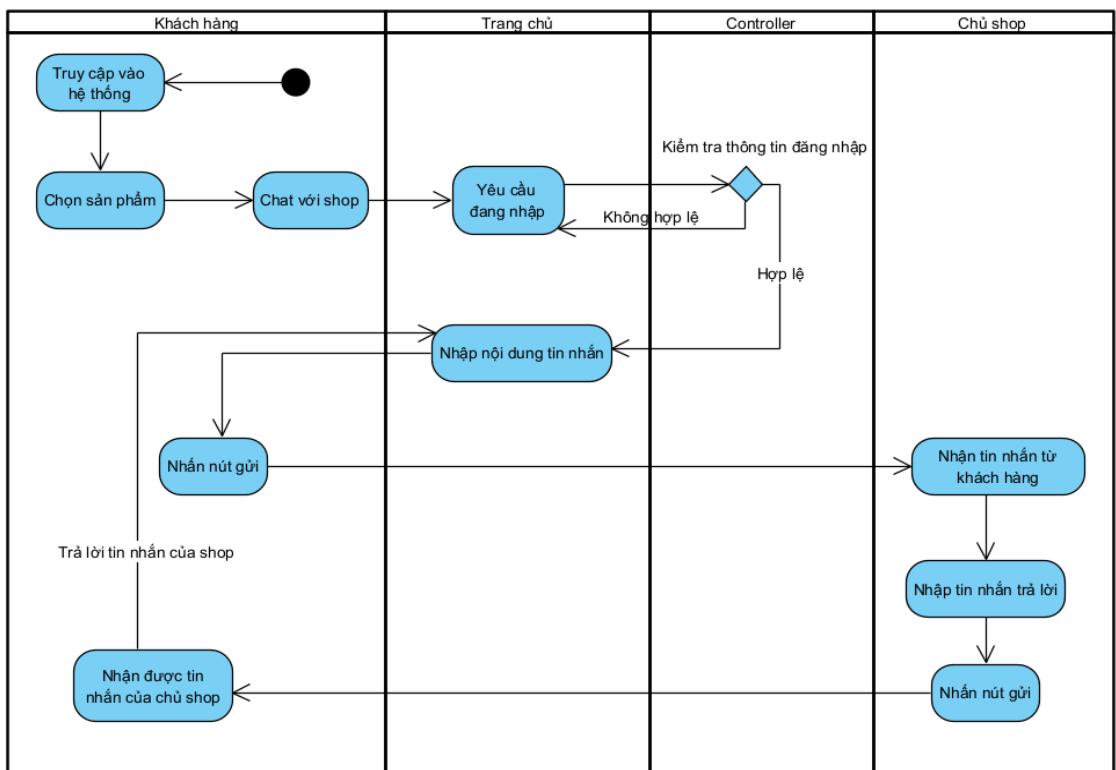
Hình 39. Mô tả tính năng tìm kiếm bằng Activity Diagram

- Tính năng thanh toán



Hình 40. Mô tả tính năng thanh toán bằng Activity Diagram

- Tính năng Chat realtime

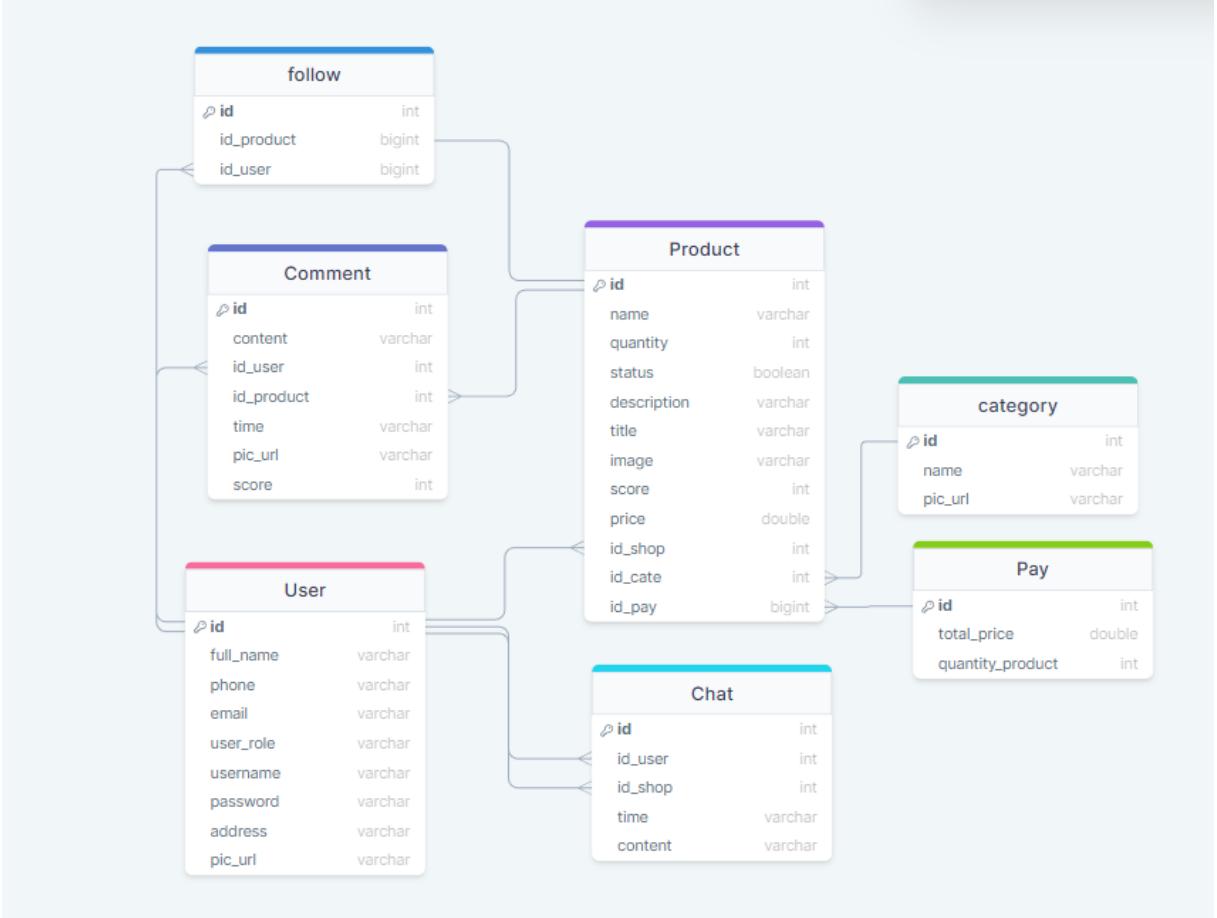


Hình 41. Mô tả tính năng chat realtime bằng Activity Diagram

3.3. Thiết kế cơ sở dữ liệu

Nhóm chúng tôi sử dụng Firebase để tạo cơ sở dữ liệu cụ thể là Firebase real-time database nó là một thông tin cơ sở thời gian thực do Google phát triển và nó là một dạng cơ sở dữ liệu noSQL, nghĩa là nó không lưu trữ dữ liệu bằng các bảng và mối quan hệ như truyền thống mà thay vào đó Firebase lưu cơ sở dữ liệu bằng cấu trúc JSON.

Cho nên nhóm chúng tôi dùng vẽ bảng để có thể nhìn một cách trực quan, hệ thống sàn thương mại gồm các bảng sau:



Hình 42. Lược đồ cơ sở dữ liệu của app sàn thương mại điện tử

3.4. Các bảng trong CSDL quan hệ

3.4.1. Bảng Product

Tên trường	Khóa chính	Kiểu dữ liệu	Mô tả	Thông tin thêm
id	có	INT	Trường id của Product	Not null, auto increase
id_product		INT	Số lượng Product	0
status		Boolean	Trạng thái Product	True

description		VARCHAR(255)	Mô tả sản phẩm	
title		VARCHAR(255)	Tiêu đề sản phẩm	Not null
image		VARCHAR(255)	Link ảnh sản phẩm	Not null
score		INT	Số sao đánh giá	Not null
price		Double	Giá của sản phẩm	Not null
id_shop		INT		Not null
id_cate		INT		Not null
id pay		INT		

3.4.2. Bảng Follow

Tên trường	Khóa chính	Kiểu dữ liệu	Mô tả	Thông tin thêm
id	có	INT	Trường id của Follow	Not null, auto increase
id_user		INT	Id của người follow	
id_product		INT	Id của sản phẩm cần follow	

3.4.3. Bảng Comment

Tên trường	Khóa chính	Kiểu dữ liệu	Mô tả	Thông tin thêm
id	có	INT	Trường id của Comment	Not null, auto increase
id_user		INT	Id của người comment	
id_product		INT	Id của sản phẩm được comment	
content		VARCHAR(255)	Nội dung comment	
time		VARCHAR(255)	Thời gian comment	
pic_url		VARCHAR(255)	Link ảnh comment	
score		INT	Số sao đánh giá	

3.4.4. Bảng Chat

Tên trường	Khóa chính	Kiểu dữ liệu	Mô tả	Thông tin thêm
id	có	INT	Trường id của Chat	Not null, auto increase
id_user		INT	Id của người chat	
id_shop		INT	Id của shop bị	

			nhắn	
content		VARCHAR(25 5)	Nội dung chat	
time		VARCHAR(25 5)	Thời gian Chat	

3.4.5. Bảng Category

Tên trường	Khóa chính	Kiểu dữ liệu	Mô tả	Thông tin thêm
id	có	INT	Trường id của Category	Not null, auto increase
name		VARCHAR(25 5)	Tên danh mục	
pic_url		VARCHAR(25 5)	Link hình ảnh danh mục	

3.4.6. Bảng Pay

Tên trường	Khóa chính	Kiểu dữ liệu	Mô tả	Thông tin thêm
id	có	INT	Trường id của Pay	Not null, auto increase
total_price		Double	Tổng giá của tất cả sản phẩm mua	
quantity_pro		INT	Tổng số lượng	

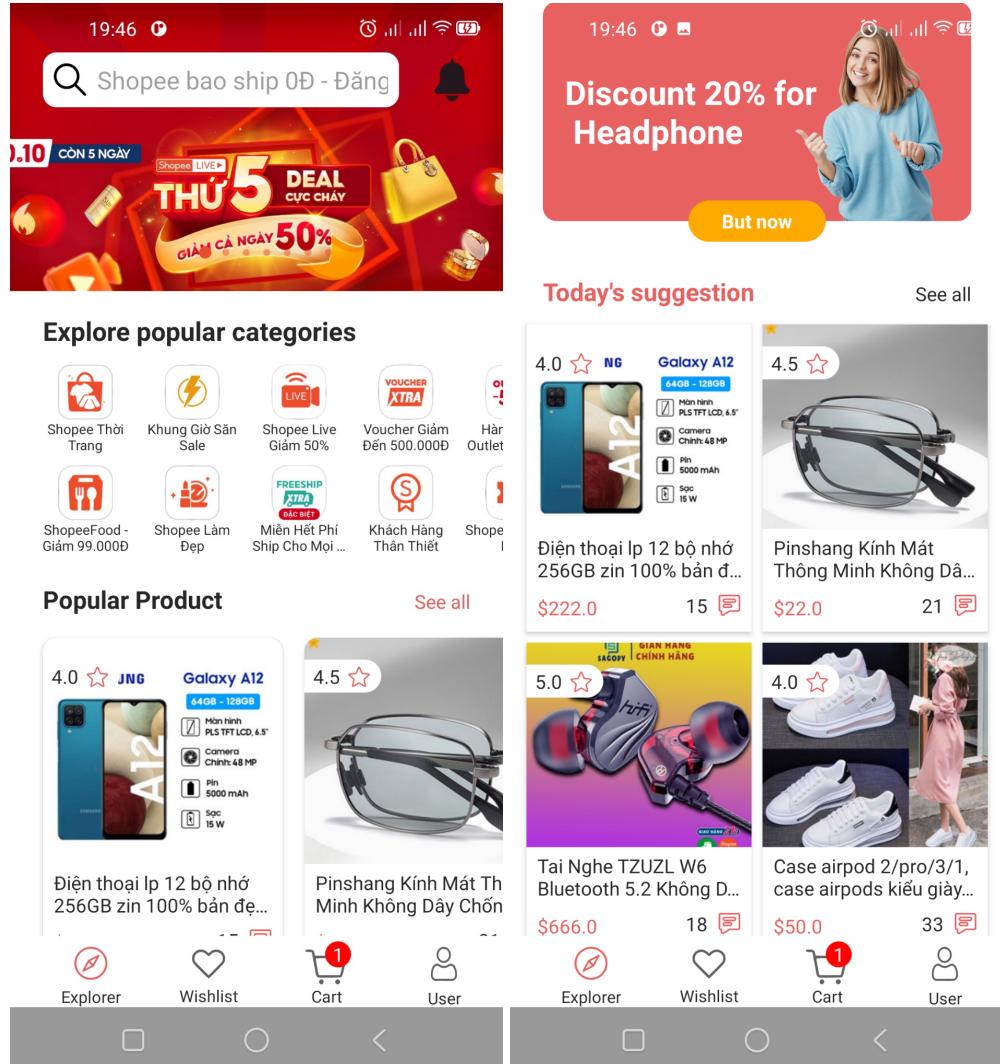
duct			sản phẩm mua	
------	--	--	--------------	--

3.4.7. Bảng User

Tên trường	Khóa chính	Kiểu dữ liệu	Mô tả	Thông tin thêm
id	có	INT	Trường id của User	Not null, auto increase
full_name		VARCHAR(255)	Tên đầy đủ của người dùng	
phone		VARCHAR(255)	Số điện thoại	Not null
email		VARCHAR(255)	Email	Not null
user_role		VARCHAR(255)	Quyền của user	
username		VARCHAR(255)	Tên đăng nhập	Not null
password		VARCHAR(255)	Mật khẩu	Not null
address		VARCHAR(255)	Địa chỉ	
pic_url		VARCHAR(255)	Link ảnh của user	

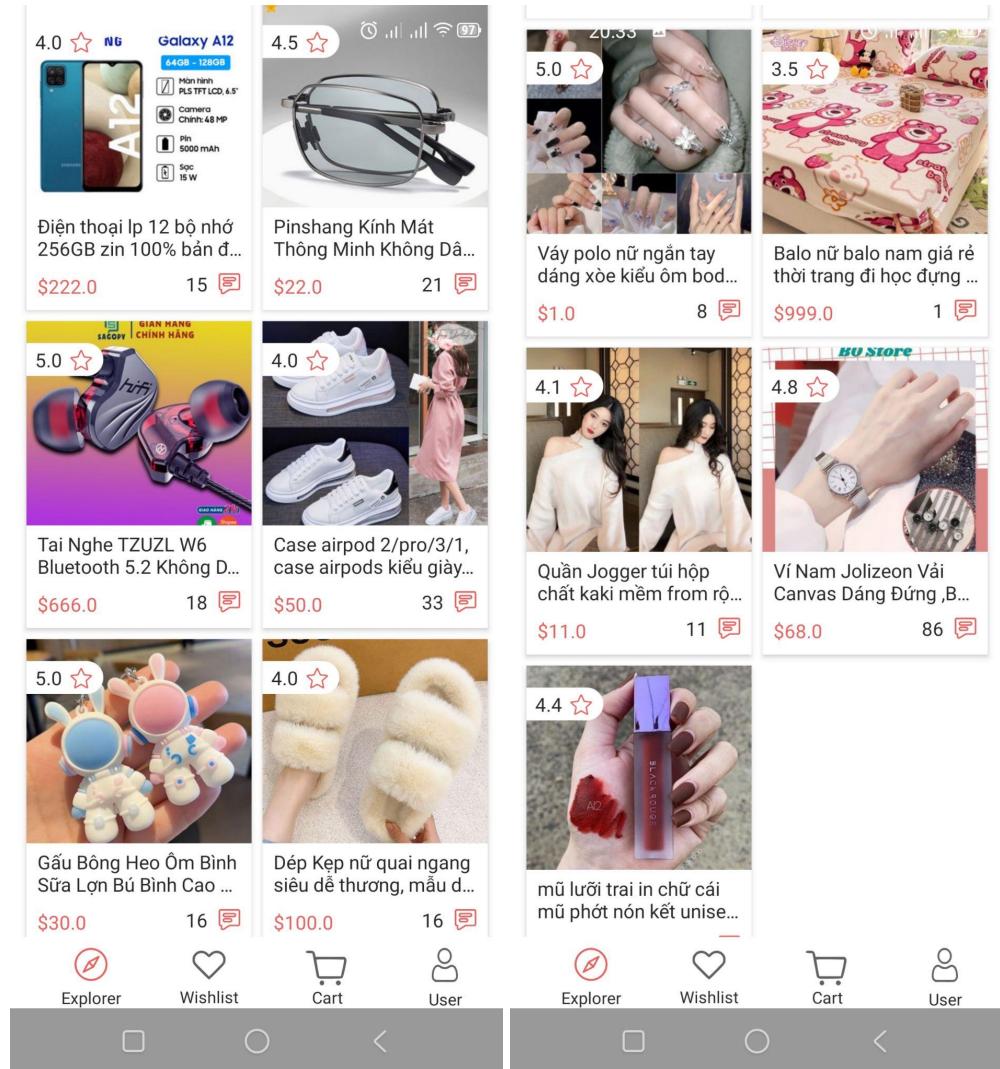
3.5. Các giao diện của app mobile

3.5.1. Giao diện trang chủ



Hình 43. Giao diện của trang chủ

3.5.2. Giao diện sản phẩm



Hình 44. Giao diện của trang sản phẩm

3.5.3. Giao diện chi tiết sản phẩm

19:50

20:39

Buy Now

Gating and Reviews See all

☆ 5.0 Rating 1 Reviews

cai hoang huynh
☆ 1 Rating 9:9 9/9/2023

quá đẹp

Leave a review

Today's suggestion See all

4.0 ☆ NG Galaxy A12 64GB - 128GB Màn hình PLST TFT LCD, 6.5" Camera Chính: 48 MP Pin 5000 mAh Sạc 15 W \$222.0 15	4.5 ☆ Pinshang Kính Mát Thông Minh Không Dây... \$22.0 21
Điện thoại Iphone 12 bộ nhớ 256GB zin 100% bản đ... \$222.0 15	
5.0 ☆ 4.0 ☆	4.0 ☆

44444 Giày form to nên các bạn chú ý đặt nên
giảm 1 size ạ bạn nào chân dày dặn to thi đặt
chuẩn size nhé đủ 2 màu đen và trắng size
35-39 ❤ Một số lưu ý khi đặt hàng ❤"

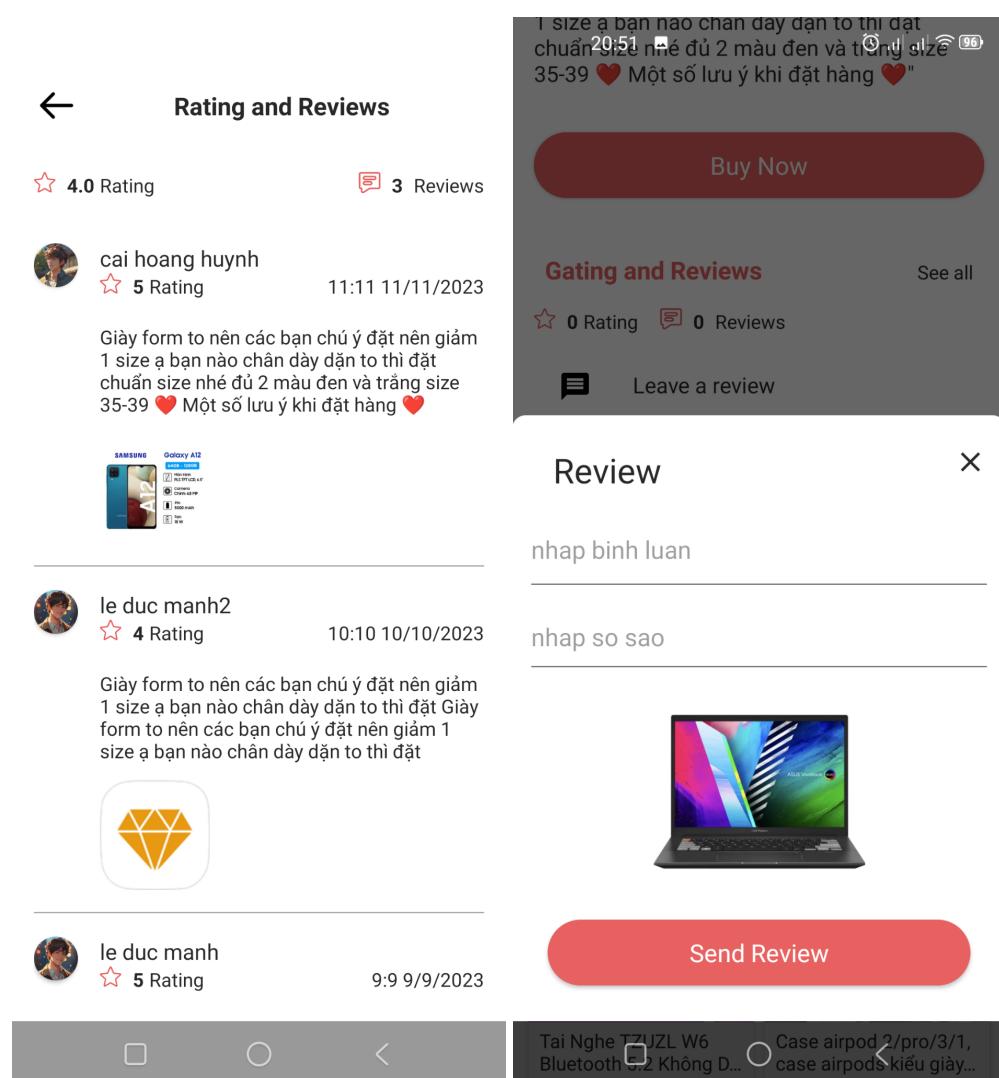
Buy Now

Gating and Reviews See all

☆ 4.0 Rating 1 Reviews

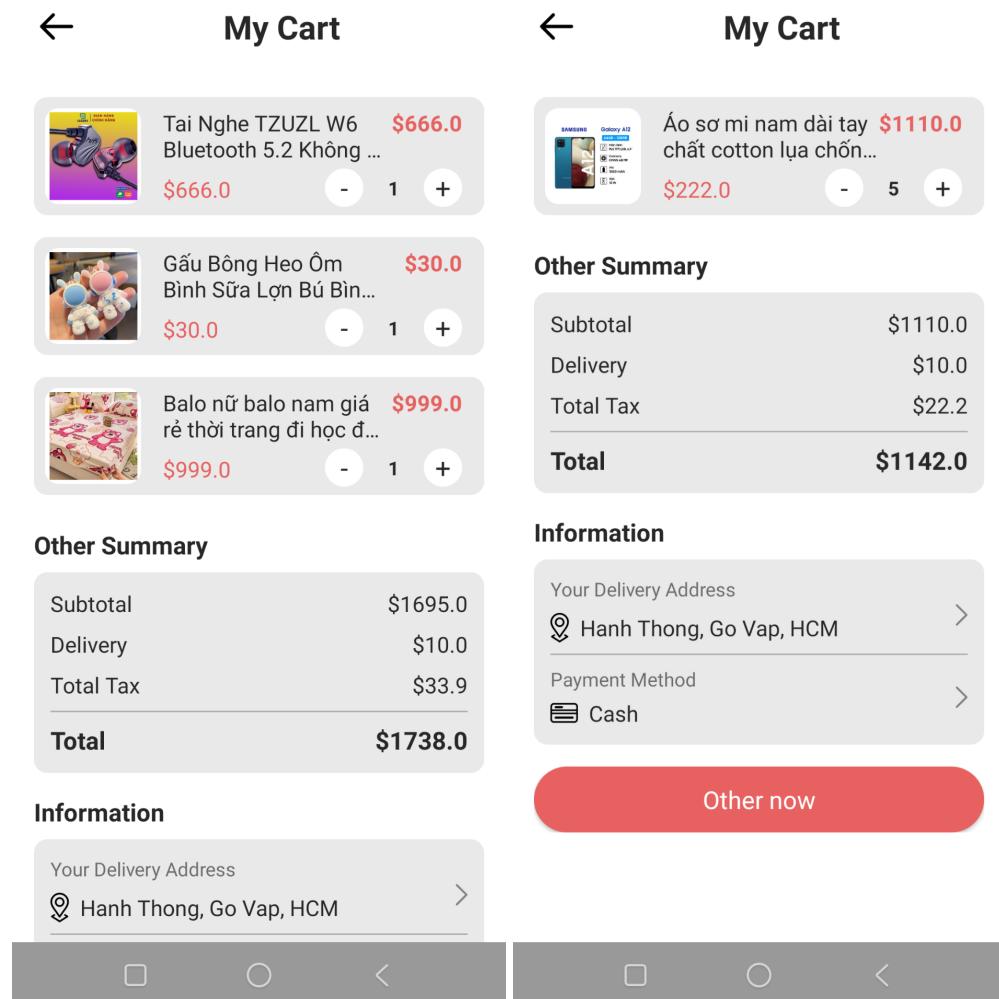
Hình 45. Giao diện chi tiết sản phẩm

3.5.4. Giao diện bình luận và xem bình luận



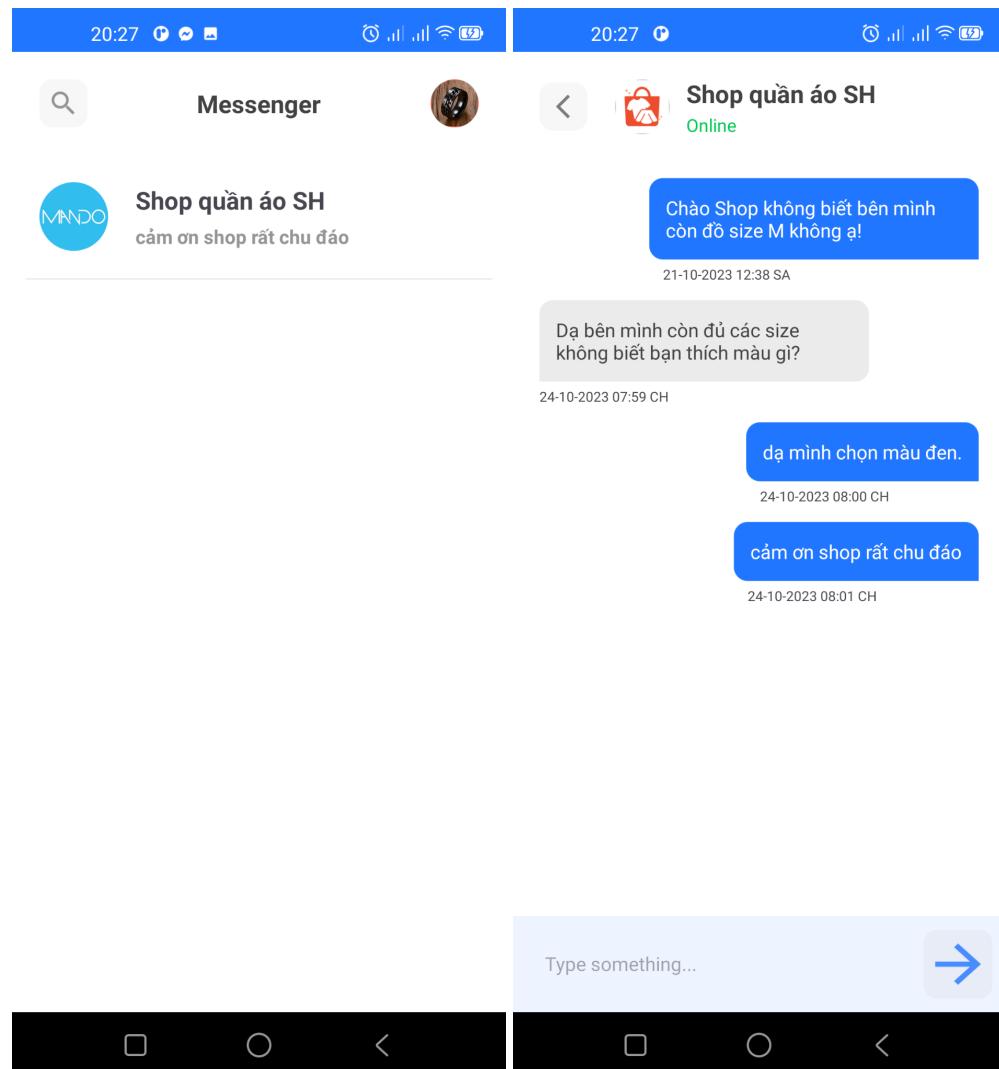
Hình 46. Giao diện bình luận và xem tất cả bình luận

3.5.5. Giao diện giỏ hàng và thanh toán



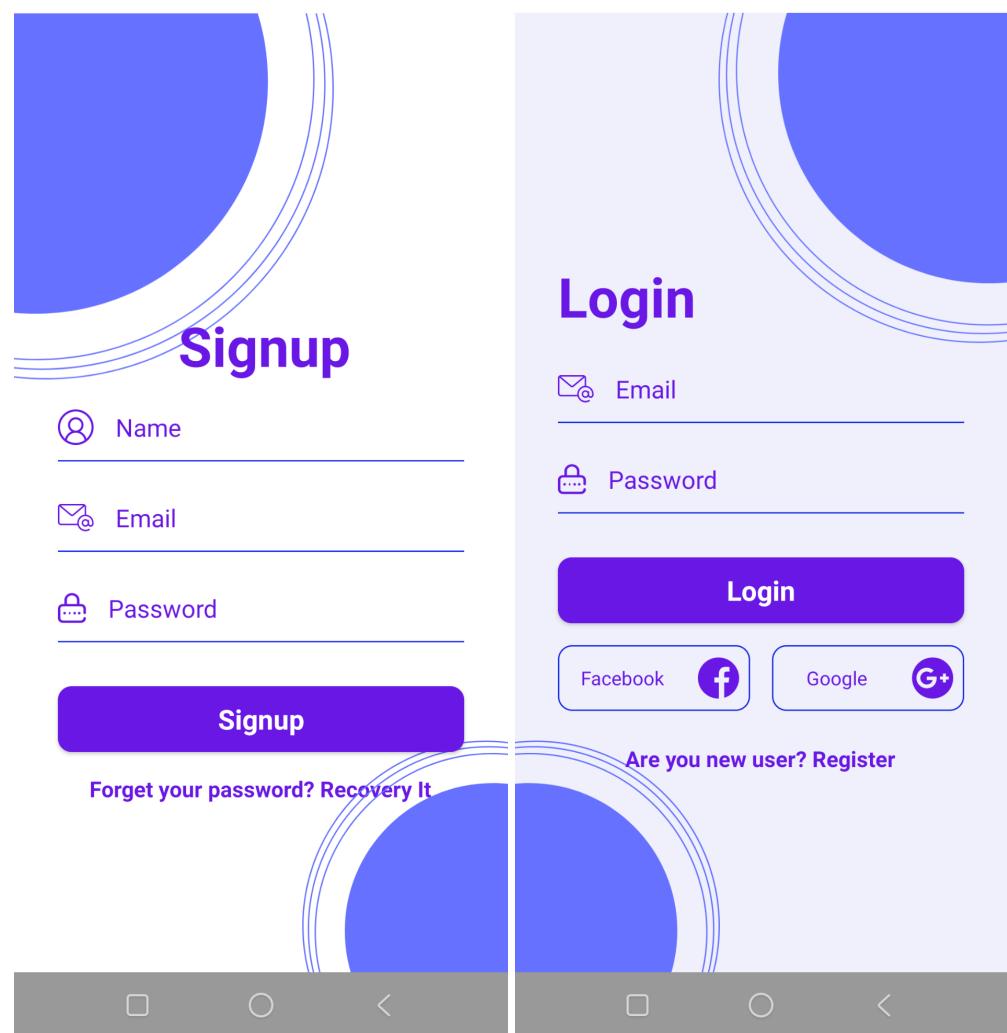
Hình 47. Giao diện trang giỏ hàng và thanh toán

3.5.6. Giao diện trang chat realtime



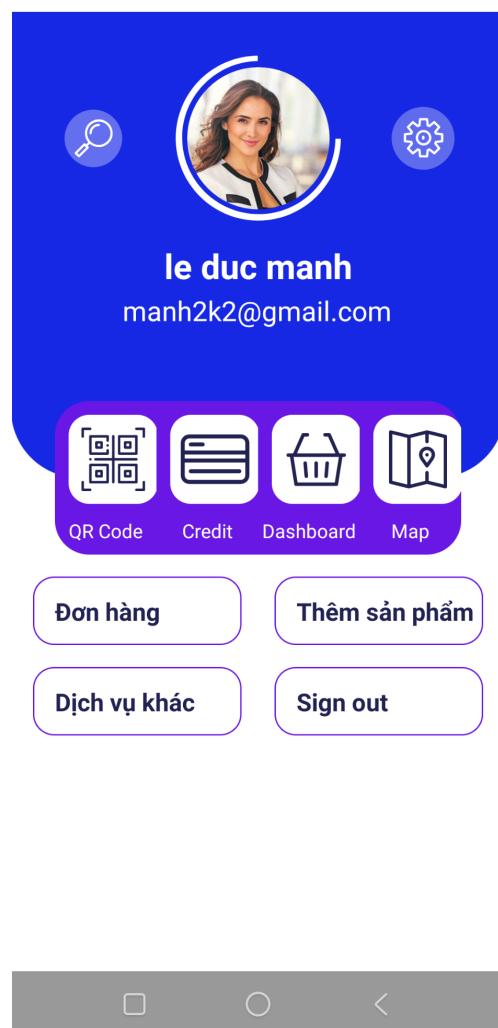
Hình 48. Giao diện chat realtime

3.5.7. Giao diện đăng nhập đăng ký



Hình 49. Giao diện đăng nhập và đăng ký

3.5.8. Giao diện profile của người dùng



Hình 50. Giao diện trang profile của người dùng

Chương 4. KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

4.1. Kết luận

- Sau hơn 2 tháng thực hiện đồ án này nó đã giúp em tích lũy được rất nhiều kinh nghiệm, giúp em hiểu hơn về quy trình hệ thống bán hàng thực tế và khi áp dụng công nghệ thông tin vào thực tiễn với những chức năng cơ bản phục vụ yêu cầu thiết yếu hiện nay như: đặt hàng, bán hàng đăng sản phẩm, trò chuyện trực tuyến, đánh giá sản phẩm, theo dõi đơn hàng, thanh toán online, tìm kiếm sản phẩm, thông báo tiêu dùng, gợi ý sản phẩm vv...

- Tuy nhiên do lần đầu làm đồ án nên không tránh khỏi những thiếu sót và những chức năng vẫn chưa kịp hoàn thiện như: thống kê doanh số, tần suất bán hàng cho chủ shop, admin hoặc chức năng thanh toán online kết hợp với momo, banking và tính năng google map.

4.2. Định hướng phát triển

4.2.1. Ưu điểm

Tổng quan App sàn thương mại điện tử của em đã đạt được một số chức năng như sau:

- + Đăng nhập đăng ký, phân quyền rõ ràng bảo mật cao.
- + Khả năng cập nhật dữ liệu thời gian thực tăng cao khả năng trải nghiệm người dùng.
- + Chức năng bình luận có cả đăng hình khách quan giúp người dùng dễ dàng hơn mua sắm.
- + Giao diện đẹp đơn giản dễ sử dụng và tương tác.
- + Đáp ứng được những chức năng mua bán đầy đủ.
- + Trò chuyện trực tuyến giúp thúc đẩy kết nối giữa người bán và mua.
- + Hỗ trợ thanh toán online.

4.2.2. Nhược điểm

Bên cạnh những ưu điểm nêu trên, mà còn những khuyết điểm mà em cần phải khắc phục để làm kinh nghiệm cho mình sau này.

- + do vấn đề Database Realtime nên không tránh khỏi tình trạng load app chậm, dữ liệu không ổn định dẫn tới dễ bị mất data.
- + một số chức năng vẫn chưa kịp hoàn thiện như: tích hợp google map lấy vị trí hiện tại và tích hợp ngân hàng hoặc các cổng thanh toán online để thanh toán sản phẩm.

TÀI LIỆU THAM KHẢO

(Theo chuẩn IEEE – bỏ dòng này khi viết báo cáo)

[1] Duong Huu Thanh. Lap Trinh Java. NXB Thong tin & Truyen Thong. 2019

PHỤ LỤC