

Trường Đại học Bách Khoa Hà Nội
Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Ứng dụng sàn thương mại điện tử trên thiết bị Android

Vũ Xuân Trường

Hà Nội, 06/2018

Trường Đại học Bách Khoa Hà Nội
Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Ứng dụng sản thương mại điện tử trên thiết bị Android

Sinh viên thực hiện Vũ Xuân Trường

Người hướng dẫn ThS. Lê Đức Trung

Hà Nội, 06/2018

Lời cam kết

Họ và tên sinh viên: Vũ Xuân Trường

Điện thoại liên lạc: 0983 815 459

Email: truongvu0106@gmail.com

Lớp: CNTT&TT 2.04

Hệ đào tạo: Chính quy

Tôi – *Vũ Xuân Trường* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS. Lê Đức Trung*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày tháng năm

Tác giả ĐATN

Vũ Xuân Trường

Lời cảm ơn

« Lửa thử vàng gian nan thử thách »

Những năm tháng học tập tại Bách Khoa là khoảng thời gian mà chắc chắn ai đã từng trải qua một lần sẽ không thể nào quên. Đó không chỉ là những tháng ngày học tập miệt mài hay những giờ phút thi cử căng thẳng, Bách Khoa còn là khi ta mắc sai lầm, khi lần đầu biết nếm mùi thất bại và tự đứng dậy trên đôi chân của mình, để ta thấy được giá trị bản thân và sự trưởng thành qua từng ngày. Năm năm có lẽ là quãng thời gian không dài nhưng đủ cho mỗi chúng ta có một trải nghiệm về cái chất riêng mà nơi đây mang lại, để khi nghĩ về nó, ta càng thấm thía hơn câu nói : «Bách Khoa Hà Nội là nơi đào tạo những học sinh giỏi nhất để họ nhận ra rằng họ chưa phải là người giỏi nhất.»

Thành công của mỗi người, bên cạnh những nỗ lực và phấn đấu của bản thân, không thể thiếu đi bóng dáng của những người thầy tận tình chỉ dẫn. Một vài dòng ngắn ngủi có lẽ là chưa đủ, nhưng em xin được phép gửi lời cảm ơn chân thành nhất của mình đến các thầy cô thuộc bộ môn Công nghệ phần mềm, viện Công nghệ thông tin – Truyền thông, Đại học Bách Khoa Hà Nội, đặc biệt là thầy Lê Đức Trung, người đã luôn quan tâm, đồng hành cùng em trong suốt quá trình thực hiện đồ án. Em xin kính chúc các thầy cô luôn luôn mạnh khỏe và tràn đầy nhiệt huyết, là chỗ dựa vững chắc và là «ngọn hải đăng» dẫn đường cho những thế hệ sinh viên sau này.

Điều sau cùng, xin gửi lời cảm ơn đến các bạn lớp CNTT 2.04 K58. Sự đồng hành và giúp đỡ của các bạn trong suốt những năm tháng đại học là nguồn động viên không hề nhỏ giúp mình vượt qua khó khăn, thử thách. Chúc các bạn luôn tìm thấy niềm vui và thành công trong cuộc sống. Xin cảm ơn tất cả!

Tóm tắt

Công nghệ thông tin đã và đang thể hiện vai trò sâu rộng của mình trong mọi lĩnh vực đời sống, đặc biệt là trong các hoạt động thương mại. Các sàn thương mại điện tử lần lượt ra đời và phát triển không ngừng nhằm đáp ứng nhu cầu ngày một tăng cao về các hoạt động mua sắm và bán hàng trực tuyến. Khởi đầu của các sàn thương mại điện tử này là những website kinh doanh online. Đến tận thời điểm hiện tại, hình thức kinh doanh qua website này vẫn duy trì được vị trí của mình. Tuy nhiên, cùng với sự gia tăng nhanh chóng của các thiết bị di động, cách tiếp cận dựa trên nền tảng website lại phát sinh một số nhược điểm cố hữu như độ linh hoạt kém, không tương thích tốt với các màn hình nhỏ, kết nối chậm chạp... Đây là thời điểm mà các ứng dụng thương mại điện tử trên nền tảng điện thoại thông minh bắt đầu nở rộ và thể hiện được ưu thế vượt trội của mình: khả năng tương tác tốt, tối ưu cho các kích thước màn hình nhỏ, dễ dàng thực hiện các chức năng chỉ với các thao tác vuốt và chạm, đem lại trải nghiệm tốt cho người dùng... Một trong những đặc thù của hoạt động thương mại là: mang lại cho khách hàng ấn tượng tốt về cách giới thiệu sản phẩm và trải nghiệm sẽ làm tăng cơ hội bán hàng hơn nhiều lần. Đó cũng là những lý do chính mà em lựa chọn nền tảng di động cho việc phát triển một ứng dụng sàn thương mại điện tử. Để thực hiện đề tài này, công nghệ mà em lựa chọn là Android – một hệ điều hành mã nguồn mở với số lượng smartphone đang sử dụng chiếm hơn 80% tổng số smartphone trên toàn thế giới.

Về cơ bản, đồ án đã xây dựng thành công sản phẩm ứng dụng thương mại điện tử trên thiết bị Android. Sản phẩm áp dụng những quy trình đã được tiêu chuẩn hóa của các sàn thương mại điện tử hiện nay và được điều chỉnh lại ở một số mặt cho phù hợp với nhiều đối tượng người dùng khác nhau, đồng thời vận dụng mô hình MVP để xử lý những nghiệp vụ phức tạp, tạo ra ứng dụng có kiến trúc linh hoạt, dễ dàng mở rộng.

Abstract

Information technology has been playing an important role in every aspect of our daily lives, especially in commercial activities. E-commerce platforms constantly evolve and develop to meet the ever-increasing demand for online shopping. E-commerce platforms originated as relatively simple business websites, developing now to highly functional and trusted ways for consumers to learn about products and services. However, the rapid increase in mobile devices bring some inherent challenges and obstacles when using the web-based approach with examples including poor flexibility, incompatibility with smaller mobile screens, and slower web connections. It's time for e-commerce smartphone applications to begin to flourish and show their superiority: Good interoperability, optimization for small screen sizes, the ease of single-touch functionality. All focused towards giving users a good experience. One of the peculiarities of business is that: giving the customer a good first impression of their product and customer experience, increases the chances of a sale. That is also the main reason why I choose the mobile base to develop an e-commerce platform using a mobile application. The technology that I choose in this topic is Android – an open source operating system with more than 80% of all smartphones in use worldwide.

Basically, this project has successfully built an E-commerce application for Android devices. This application utilizes the standardized processes of current applications and is adjusted in some aspects to suit different end-users, utilizing the MVP model to handle complex operations to create an application with flexible and expandable architecture.

Mục lục

Lời cam kết.....	iii
Lời cảm ơn	iv
Tóm tắt	v
Abstract	vi
Mục lục	vii
Danh mục hình vẽ.....	xii
Danh mục bảng.....	xiv
Danh mục các từ viết tắt	xv
Danh mục thuật ngữ.....	xvi
Chương 1 Giới thiệu đề tài.....	1

1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài	3
1.3 Định hướng giải pháp	4
1.4 Bố cục đồ án	5
 Chương 2 Khảo sát và phân tích yêu cầu	 6
2.1 Khảo sát hiện trạng	6
2.2 Tổng quan chức năng.....	7
2.2.1 Biểu đồ use case tổng quan.....	7
2.2.2 Biểu đồ use case phân rã CRUD banner.....	9
2.2.3 Biểu đồ use case phân rã CRUD danh mục con	10
2.2.4 Biểu đồ use case phân rã CRUD danh mục cha	11
2.2.5 Biểu đồ use case phân rã UD đơn hàng	11
2.2.6 Biểu đồ use case phân rã CRUD sản phẩm	12
2.2.7 Biểu đồ use case phân rã Quản lý giỏ hàng	12
2.2.8 Quy trình nghiệp vụ	13
2.3 Đặc tả chức năng	15
2.3.1 Đặc tả use case Đặt hàng	15
2.3.2 Đặc tả use case Thêm sản phẩm mới.....	17

2.3.3	Đặc tả use case Cập nhật trạng thái đơn hàng	19
2.3.4	Đặc tả use case Đăng ký mở gian hàng	20
2.3.5	Đặc tả use case Đánh giá sản phẩm	23
2.4	Yêu cầu phi chức năng.....	24
Chương 3	Công nghệ sử dụng	26
3.1	Android.....	26
3.1.1	Tổng quan về Android	26
3.1.2	Lý do sử dụng	28
3.1.3	Kỹ thuật áp dụng.....	29
3.2	PHP	30
3.2.1	Tổng quan về PHP	30
3.2.2	Lý do sử dụng	32
3.2.3	Kỹ thuật áp dụng.....	32
Chương 4	Phát triển và triển khai ứng dụng	34
4.1	Thiết kế kiến trúc	34
4.1.1	Lựa chọn kiến trúc phần mềm	34
4.1.2	Thiết kế tổng quan	38
4.1.3	Thiết kế chi tiết gói	39

4.2 Thiết kế chi tiết	41
4.2.1 Thiết kế giao diện.....	41
4.2.2 Thiết kế lớp	44
4.2.3 Thiết kế cơ sở dữ liệu.....	45
4.3 Xây dựng ứng dụng.....	47
4.3.1 Thư viện và công cụ sử dụng.....	47
4.3.2 Kết quả đạt được	48
4.3.3 Minh hoạ các chức năng chính	49
4.4 Kiểm thử.....	53
4.4.1 Kiểm thử hộp đen.....	53
4.4.2 Kiểm thử hộp trắng	55
4.5 Triển khai.....	56
 Chương 5 Các giải pháp và đóng góp nổi bật	 58
 5.1 Áp dụng quy trình nghiệp vụ phức tạp vào giải quyết bài toán.....	 58
5.1.1 Đặt vấn đề	58
5.1.2 Giải pháp	59
 5.2 Hoàn thiện kiến trúc hệ thống theo mô hình MVP.....	 60

5.2.1 Đặt vấn đề	60
5.2.2 Giải pháp	60
5.3 Mô đun hóa các chức năng	61
5.3.1 Đặt vấn đề	61
5.3.2 Giải pháp	63
Chương 6 Kết luận và hướng phát triển.....	65
6.1 Kết luận	65
6.2 Hướng phát triển	66
Tài liệu tham khảo.....	67

Danh mục hình vẽ

Hình 1	Biểu đồ use case tổng quan	9
Hình 2	Biểu đồ use case phân rã CRUD banner	10
Hình 3	Biểu đồ use case phân rã CRUD danh mục con	10
Hình 4	Biểu đồ use case phân rã CRUD danh mục cha	11
Hình 5	Biểu đồ use case phân rã UD đơn hàng	11
Hình 6	Biểu đồ use case phân rã CRUD sản phẩm	12
Hình 7	Biểu đồ use case phân rã Quản lý giỏ hàng	13
Hình 8	Biểu đồ hoạt động quy trình mua hàng	13
Hình 9	Biểu đồ hoạt động quy trình xác nhận đặt hàng	14
Hình 10	Kiến trúc hệ điều hành Android	27
Hình 11	Các thành phần và luồng dữ liệu trong MVP	35
Hình 12	Thiết kế tổng quan ứng dụng	38
Hình 13	Thiết kế gói Giỏ hàng	39
Hình 14	Thiết kế gói chi tiết sản phẩm	40
Hình 15	Thiết kế gói Đặt hàng	41
Hình 16	Bảng phối màu của ứng dụng	42

Hình 17 Minh họa thiết kế ứng dụng.....	43
Hình 18 Thiết kế lớp chi tiết.....	44
Hình 19 Biểu đồ trình tự.....	45
Hình 20 Biểu đồ thực thể liên kết.....	46
Hình 21 Các bảng dữ liệu	47
Hình 22 Giao diện màn hình trang chủ	49
Hình 23 Giao diện màn hình danh mục sản phẩm	50
Hình 24 Giao diện màn hình danh sách sản phẩm	51
Hình 25 Giao diện màn hình thống kê	52

Danh mục bảng

Bảng 1 Thống kê một số ứng dụng thương mại điện tử	6
Bảng 2 Đặc tả use case Đặt hàng	15
Bảng 3 Đặc tả use case Thêm sản phẩm mới	17
Bảng 4 Đặc tả use case Cập nhật trạng thái đơn hàng	19
Bảng 5 Đặc tả use case Đăng ký mở gian hàng	20
Bảng 6 Đặc tả use case Đánh giá sản phẩm	23
Bảng 7 Danh sách công cụ và thư viện sử dụng	47
Bảng 8 Thống kê các thông tin ứng dụng	48
Bảng 9 Kiểm thử hộp đen chức năng “Thêm mới sản phẩm”	53
Bảng 10 Kiểm thử hộp đen chức năng “Đăng ký mở gian hàng”	54
Bảng 11 Thống kê kiểm thử hộp trắng	56
Bảng 12 Thống kê các thiết bị triển khai	56

Danh mục các từ viết tắt

IIS	Internet Information Services Dịch vụ thông tin internet
API	Application Programming Interface Giao diện lập trình ứng dụng
HAL	Hardware Abstraction Layer Lớp phần cứng trừu tượng trong kiến trúc Android
DEX	Dalvik Executable Dạng file thực thi được phát sinh sau khi biên dịch một chương trình Android
ART	Android runtime Một thành phần trong kiến trúc Android
SDK	Software Development Kit Bộ công cụ phát triển phần mềm
CMS	Content Management System Hệ thống quản trị nội dung

Danh mục thuật ngữ

E-commerce	Thương mại điện tử
E-tailing	Bán lẻ trực tuyến
Release	Phát hành sản phẩm
Cross-platform	Đa nền tảng
Cohesion	Kiểm dính giữa các thành phần trong một mô đun hoặc một lớp
Coupling	Ghép nối giữa hai hoặc nhiều mô đun/lớp

Chương 1 Giới thiệu đề tài

1.1 Đặt vấn đề

Trong những năm gần đây, không thể phủ nhận được sự bùng nổ mạnh mẽ của công nghệ thông tin, kèm theo đó là hàng loạt những ứng dụng của nó vào nhiều phương diện khác nhau của đời sống. Thương mại có lẽ là một trong những lĩnh vực chịu ảnh hưởng nhiều nhất. Hàng loạt những thuật ngữ và khái niệm như kinh doanh điện tử (e – business), mua hàng online, bán lẻ trực tuyến (e – tailing) hay “cửa hàng ảo” xuất hiện với tần suất ngày một dày đặc đủ cho chúng ta thấy đã và đang có sự chuyển biến rõ rệt trong tư duy cũng như cách thức vận hành của các hoạt động thương mại, từ hình thức truyền thống sang một hình thức mới hiệu quả hơn, dựa trên thành tựu của công nghệ thông tin – hình thức thương mại điện tử.

Vậy “Thương mại điện tử” là gì? Thương mại điện tử¹, hay còn gọi là e-commerce, e-comm hay EC, là sự mua bán sản phẩm hay dịch vụ trên các hệ thống điện tử như Internet và các mạng máy tính. Trong khoảng thời gian đầu tiên, thương mại điện tử thường gắn liền với máy tính, trình duyệt web và các website bán hàng online. Đến nay thời điểm hiện tại, những website này vẫn còn duy trì vai trò to lớn của mình. Tuy nhiên, với sự dịch chuyển không ngừng của dòng chảy công nghệ và nhu cầu thay đổi liên tục của người dùng, đặc biệt là sự bùng nổ của các thiết bị di động như điện thoại thông minh (Theo eMaketer, đến năm 2017, có khoảng 2,39 tỉ người trên thế giới đang sử dụng smartphone), hình thức kinh doanh qua website lại đang dần bộc lộ những nhược điểm như tính linh động không cao, người dùng phải tốn thêm

¹ https://vi.wikipedia.org/wiki/Thương_mại_điện_tử, lần truy cập cuối: 12/05/2018

các thao tác truy cập trình duyệt web và tìm kiếm, cơ hội tiếp cận đến khách hàng bởi thể cũng giảm đi. Đa số các website dù đã được phát triển theo hướng thân thiện với các thiết bị di động, nhưng trải nghiệm trong nhiều tình huống còn chưa thực sự tốt. Chính vì lẽ đó, khách hàng đang ngày càng ưa chuộng sử dụng các ứng dụng di động được phát triển sẵn thay vì phải truy cập các trình duyệt công kênh với các thao tác tìm kiếm rắc rối. Về phía người kinh doanh, việc sở hữu cho kênh bán hàng của mình một ứng dụng riêng sẽ là cách hiệu quả nhất để định hướng trực tiếp người dùng đến sản phẩm chỉ bằng một thao tác chạm tay. Tuy nhiên, chi phí để phát triển một ứng dụng di động là không hề nhỏ, và đối với những người bán hàng nhỏ lẻ, điều này là vượt quá khả năng của họ. Sẽ là tốt nhất nếu có một ứng dụng đảm nhiệm vai trò làm cầu nối giữa người bán và người mua. Ứng dụng đó tương tự như một khu chợ online, một sàn thương mại điện tử, nơi rất nhiều người bán có thể cùng quảng cáo, giới thiệu mặt hàng của mình và người mua kết nối được một cách nhanh chóng nhất với các thông tin và sản phẩm mà họ cần, thậm chí là kết nối trực tiếp đến doanh nghiệp kinh doanh. Chi phí xây dựng ứng dụng vì thế mà cũng được san sẻ và sẽ không còn là gánh nặng cho riêng cá nhân hay tổ chức nào. Có thể kể thêm rất nhiều lợi ích khác nữa từ một ứng dụng sàn thương mại điện tử. Về phía người bán hàng, nó giúp: (i) gây dựng lòng tin từ khách hàng; (ii) tăng cường giá trị thương hiệu, (iii) tăng khả năng truy cập; tạo điều kiện cho khách hàng tiếp cận sản phẩm; (iv) tăng khả năng bán hàng trực tuyến; (v) liên kết chặt chẽ với các khách hàng tiềm năng... Lợi nhuận vì thế cũng tăng lên rõ rệt. Khi Domino của Pizza tạo ra một ứng dụng di động để đặt giao hàng hoặc đến lấy tại cửa hàng của họ, họ đã nhìn thấy sự gia tăng về lợi nhuận tới 28% trước thuế chỉ tính riêng tại Anh. Trong thực tế, các thiết bị di động hiện nay chiếm khoảng 52% đơn đặt hàng trực tuyến của họ². Về phía người mua, họ có thể: (i) dễ dàng tiếp cận tới sản phẩm mọi lúc mọi nơi; (ii) nhận được những thông báo về những sự kiện đặc biệt hay khuyến mại; (iii) dễ dàng liên hệ; (iv) đặt hàng nhanh chóng và thuận tiện...

² Theo thống kê từ website <https://mtdcreativedesigns.com>

Xây dựng thành công một ứng dụng sản thương mại trên điện thoại thông minh tuy không phải là điều dễ dàng nhưng đang trở nên thực sự cần thiết. Lợi ích mà nó mang lại không chỉ riêng cho cá nhân hay tổ chức nào mà là một nhóm người, nhóm tổ chức. Quy trình mua bán hàng, mở rộng hơn là các quy trình thanh toán, vận chuyển trở nên đơn giản hơn nhiều lần.

1.2 Mục tiêu và phạm vi đề tài

Như đã nói ở mục 1.1, nhu cầu về các sản thương mại điện tử trên ứng dụng di động bắt nguồn chủ yếu từ việc gia tăng nhanh chóng số lượng thiết bị điện thoại thông minh. Ở Việt Nam, số người dùng smartphone tăng khoảng 20% chỉ riêng trong năm 2017, thuộc top 2 nước có tỉ lệ người dùng smartphone tăng lớn nhất (Theo thống kê từ eMarketer). Nhìn thấy tiềm năng rất lớn từ Việt Nam, nhiều ông lớn trong lĩnh vực thương mại điện tử đã nhanh chóng chen chân vào thị trường này. Bên cạnh website trực tuyến với số lượng truy cập cực kì lớn mỗi ngày, những ứng dụng tương ứng trên nền thiết bị di động của họ cũng luôn thuộc top phổ biến và có lượng tải xuống ở mức rất cao. Một vài cái tên có thể kể đến đầu tiên như Lazada, Shopee...hay các đại diện trong nước như Tiki, Sendo... Được phát triển bởi đội ngũ các kĩ thuật viên hàng đầu nên những sản phẩm này chắc chắn đã đáp ứng được phần lớn nhu cầu từ người dùng. Tuy nhiên, xét về tổng quan, những ứng dụng này vẫn tập trung chủ yếu vào các chức năng phía người dùng là người mua hàng. Còn về phía người bán hàng nhỏ lẻ, trải nghiệm trong nhiều trường hợp vẫn còn chưa thực sự tốt. Các chức năng như đăng bán sản phẩm, thống kê sản phẩm, thống kê đơn hàng... còn tương đối phức tạp. Ví dụ như với Lazada, để trở thành một người bán hàng trên nền tảng này, người bán cần đăng ký trực tiếp trên website và sử dụng một ứng dụng thứ hai để có thể thực hiện đầy đủ các chức năng mong muốn. Trải nghiệm vì thế mà bị gián đoạn.

Trên cơ sở đánh giá đó, ứng dụng mà em phát triển, bên cạnh những tính năng cần thiết phải có dành cho khách mua hàng phổ thông, sẽ tập trung hơn vào những chức năng đặc thù dành cho đối tượng người bán hàng. Một người dùng có thể đóng hai vai trò: người bán hàng và người mua hàng trên cùng một ứng dụng, việc chuyển đổi

giữa hai vai trò sẽ hoàn toàn dễ dàng với một vài thao tác chạm. Trải nghiệm người dùng được đảm bảo thống nhất. Lĩnh vực kinh doanh mà ứng dụng hướng tới sẽ tập trung chủ yếu vào các sản phẩm điện tử như điện thoại, laptop, đồ điện tử gia dụng...

1.3 Định hướng giải pháp

Từ những yêu cầu và nhiệm vụ đã xác định ở mục 1.2, đề án mà em xây dựng sẽ được thực hiện theo mô hình tương tác giữa client – server. Ở phía client, ứng dụng được phát triển trên nền tảng thiết bị Android với nhiều ưu điểm nổi trội: (i) số lượng người dùng phổ thông lớn, (ii) cộng đồng phát triển đông đảo và sẵn sàng giúp đỡ, (iii) thư viện hỗ trợ đa dạng, (iv) thời gian release ngắn và nhờ đó nhanh chóng nhận được phản hồi từ phía người dùng. Ở phía server, công nghệ mà em sử dụng là PHP, vốn là một ngôn ngữ lập trình mã nguồn mở, được cài đặt phổ biến trên các WebServer thông dụng hiện nay như Apache, IIS... Từ phiên bản PHP 5, PHP đã có khả năng hỗ trợ hầu hết các đặc điểm nổi bật của lập trình hướng đối tượng như: Inheritance, Abstraction, Encapsulation, Polymorphism, Interface, Autoload... Hệ quản trị cơ sở dữ liệu mà em sử dụng là MySQL với khả năng bảo mật tốt và dễ dàng tích hợp, đặc biệt hiệu quả khi kết hợp cùng PHP. Ứng dụng trên thiết bị Android đóng vai trò là client, sẽ tương tác với server thông qua việc gửi các request với các phương thức GET và POST. Server tiếp nhận, xử lý yêu cầu và trả về response dưới định dạng dữ liệu JSON. Client xử lý chuỗi JSON nhận được, chuyển đổi thành các đối tượng thích hợp và hiển thị thông tin lên màn hình thiết bị Android.

Trên cơ sở những định hướng và giải pháp nêu trên, đề án mà em thực hiện đã kết hợp thành công các công nghệ phía client và server, tạo nên một ứng dụng hoàn chỉnh với đầy đủ các chức năng cơ bản của một sàn thương mại điện tử, Sản phẩm đặc biệt tập trung vào khả năng chuyển đổi dễ dàng giữa các vai trò người mua hàng và người bán hàng, chú trọng nhiều hơn vào các tính năng tiện ích cho người bán hàng như quản lý các đơn hàng, quản lý sản phẩm, thống kê... Đề án được xây dựng theo kiến trúc linh hoạt và hoàn toàn có khả năng mở rộng về sau.

1.4 Bố cục đề án

Phần còn lại của báo cáo đề án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về kết quả khảo sát hiện trạng và phân tích yêu cầu từ phía khách hàng. Trên cơ sở đó, đề ra các chức năng của sản phẩm, khái quát hóa các chức năng bằng biểu đồ ca sử dụng. Trong các chức năng được đặt ra, em lựa chọn 5 chức năng chính quan trọng nhất để đặc tả dưới dạng bảng. Bên cạnh đó, em cũng tiến hành mô tả và vẽ biểu đồ hoạt động minh họa hai quy trình nghiệp vụ chính là quy trình mua hàng và quy trình xác nhận đặt hàng.

Trong Chương 3, trên cơ sở các định hướng giải pháp đã nêu ở mục 1.3, em đi sâu vào giới thiệu chi tiết các công nghệ sử dụng, vai trò và tác dụng của từng công nghệ đối với các yêu cầu cụ thể.

Chương 4 là chương trình bày cụ thể về thiết kế cũng như quá trình triển khai của ứng dụng. Thiết kế được đề cập đến bao gồm cả thiết kế kiến trúc và thiết kế chi tiết. Trong thiết kế kiến trúc, em trình bày về lý thuyết cũng như áp dụng của mô hình MVP – một mô hình cải tiến từ MVC truyền thống với nhiều ưu điểm nổi trội hơn. Tiếp theo là phần trình bày và giải thích về các biểu đồ gói, chi tiết gói. Biểu đồ thực thể liên kết phục vụ cho việc thiết kế cơ sở dữ liệu cũng được nêu ra trong mục này. Phần cuối Chương 4 là các nội dung về xây dựng ứng dụng như thư viện và các công cụ được sử dụng, kết quả đạt được và minh họa một số chức năng chính.

Chương 5 trình bày về các giải pháp và đóng góp nổi bật của bản thân trong quá trình thực hiện đề án.

Chương 6 đưa ra kết luận và định hướng phát triển cho sản phẩm.

Chương 2 Khảo sát và phân tích yêu cầu

Chương 1 đã trình bày khái quát về mục tiêu và phạm vi của đề tài cùng các định hướng giải pháp. Chương 2 này sẽ là sự trình bày chi tiết về quá trình phân tích yêu cầu thông qua việc xây dựng biểu đồ ca sử dụng và đặc tả các chức năng cũng như những quy trình nghiệp vụ chính. Bên cạnh đó, một số yêu cầu phi chức năng cho sản phẩm cũng được nêu ra trong phần cuối chương.

2.1 Khảo sát hiện trạng

Hiện nay, các ứng dụng thương mại điện tử trên thiết bị di động không còn quá xa lạ. Hầu như mỗi sàn thương mại điện tử lớn đều cung cấp cho người dùng ứng dụng tương ứng trên cả hai nền tảng Android và iOS. Những ứng dụng này thường có số lượng lượt tải xuống, lượt đánh giá và bình luận thuộc hàng top, đủ cho thấy sự quan tâm rất lớn từ phía khách hàng. Bảng 1 dưới đây thống kê các thông tin liên quan đến một số ứng dụng thương mại điện tử hàng đầu ở Việt Nam hiện nay:

Bảng 1 Thống kê một số ứng dụng thương mại điện tử

Tiêu chí	Lazada	Shopee	Tiki	Sendo
Nền tảng	Android, iOS	Android, iOS	Android, iOS	Android, iOS
Số lượt tải	50 triệu	10 triệu	1 triệu	1 triệu

Danh mục hàng	Nhiều thể loại	Nhiều thể loại	Nhiều thể loại	Nhiều thể loại
Hỗ trợ mua và bán trên cùng một ứng dụng	Không	Có	Không	Không
Ứng dụng bán hàng riêng	Lazada Seller Center	Không có	Không có	Sendo Bán

Các ứng dụng này có đặc điểm chung là danh mục ngành hàng rộng, bao gồm nhiều mặt hàng khác nhau, tuy nhiên hầu hết đều không hỗ trợ mua và bán đồng thời trên cùng một ứng dụng. Điều đó có nghĩa là, khi muốn bán hàng trên các hệ thống này, người dùng bắt buộc phải sử dụng một ứng dụng thứ hai như Lazada (Lazada Seller Center) hay Sendo (Sendo bán). Từ đó, sản phẩm mà đồ án hướng tới xây dựng sẽ khắc phục những điểm này. Bên cạnh những chức năng chắc chắn phải có dành cho người mua hàng như: tìm kiếm sản phẩm, quản lý giỏ hàng, đặt hàng, quản lý tài khoản và lịch sử mua hàng... ứng dụng sẽ cung cấp những chức năng hữu ích cho người bán hàng như: đăng bán một sản phẩm trên cùng một ứng dụng, quản lý danh sách các sản phẩm đã bán ra, quản lý và cập nhật trạng thái các đơn hàng, thống kê... Việc xây dựng những chức năng này sẽ đảm bảo cung cấp cho người dùng một trải nghiệm thống nhất, đồng thời đơn giản hóa các thao tác xử lý, gắn kết chặt chẽ hơn vai trò của người bán và người mua.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quan

Hình 1 mô tả biểu đồ use case tổng quan của sản phẩm. Các tác nhân tham gia bao gồm: Khách hàng thành viên, Khách vắng lai, Người bán hàng và Admin. Tác nhân khách hàng thành viên ở đây đóng vai trò là người mua hàng, đã đăng ký một tài khoản thành viên trên hệ thống và đăng nhập thành công với tài khoản đó. Khách hàng thành viên là người trực tiếp tham gia vào toàn bộ quá trình mua hàng, từ khâu

tìm kiếm sản phẩm, xem chi tiết sản phẩm, thêm hàng vào giỏ, đặt hàng và thanh toán. Thông tin của khách hàng thành viên thu được khi đăng ký được lưu lại và có thể tùy ý truy xuất và cập nhật bất cứ lúc nào. Những thông tin này cũng được hệ thống sử dụng khi khách hàng thành viên muốn tạo đơn hàng hoặc thêm các đánh giá về một sản phẩm nào đó. Khách hàng thành viên có thể đăng ký mở gian hàng để trở thành một người bán hàng. Khách vắng lai cũng đóng vai trò là người mua hàng, tuy nhiên, khác với khách hàng thành viên, khách vắng lai là những người mua hàng chưa đăng ký tài khoản và đăng nhập vào hệ thống. Họ vẫn có thể tham gia vào các nghiệp vụ cơ bản như xem và tìm kiếm sản phẩm, đặt hàng và thanh toán. Tuy nhiên, họ sẽ bị hạn chế một số chức năng như: không thể đánh giá một sản phẩm, không theo dõi được đơn hàng, không theo dõi được gian hàng, không thêm được sản phẩm vào danh sách yêu thích... Người bán hàng là những người trực tiếp đưa sản phẩm lên hệ thống. Nghiệp vụ quan trọng nhất của tác nhân này là quy trình quản lý sản phẩm, bao gồm việc thêm, sửa, xóa sản phẩm. Bên cạnh đó, người bán hàng còn có thể thực hiện các chức năng khác như thống kê và quản lý các đơn hàng. Tác nhân cuối cùng là Admin hệ thống, người có khả năng thực hiện các chức năng chung liên quan đến toàn bộ hệ thống như xem thông tin người dùng, quản lý danh mục sản phẩm, quản lý banner quảng cáo, thêm mã giảm giá...

Một số use case quan trọng là “Quản lý giỏ hàng”, “Đặt hàng”, “CRUD sản phẩm”. Use case “Quản lý giỏ hàng” được thực hiện bởi khách hàng thành viên hoặc khách vắng lai. Sau khi xem danh sách hoặc tìm kiếm sản phẩm, người dùng mở xem chi tiết sản phẩm, sau đó chọn thêm vào giỏ hàng. Nếu sản phẩm đã tồn tại trong giỏ hàng sẽ không thêm được nữa. Người dùng có thể xem thông tin giỏ hàng bất cứ lúc nào và tùy chỉnh số lượng sản phẩm trong giỏ hàng. Use case “Đặt hàng” chỉ có thể được thực hiện khi trong giỏ hàng của khách hàng thành viên hoặc khách vắng lai tồn tại ít nhất một sản phẩm. Người dùng cần trải qua 4 bước để có thể tiến hành đặt hàng thành công: (i) xác nhận địa chỉ, thông tin liên hệ (đối với khách vắng lai, họ phải nhập lại bằng tay), (ii) chọn hình thức vận chuyển, (iii) chọn cách thức thanh toán, (iv) xác nhận đơn hàng. Use case “CRUD sản phẩm” bao gồm 3 use case nhỏ: thêm

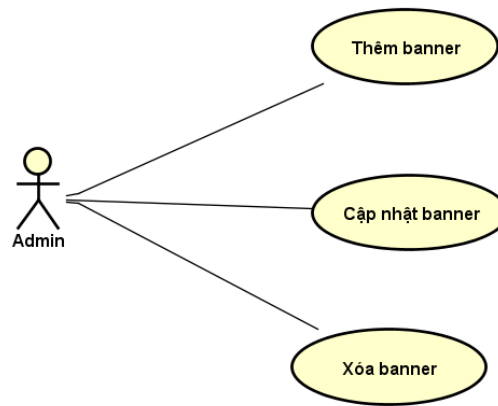
sản phẩm, chỉnh sửa sản phẩm, xóa sản phẩm. Use case này được thực hiện bởi tác nhân người bán hàng và sẽ được phân rã cụ thể trong mục 2.2.6.



Hình 1 Biểu đồ use case tổng quan

2.2.2 Biểu đồ use case phân rã CRUD banner

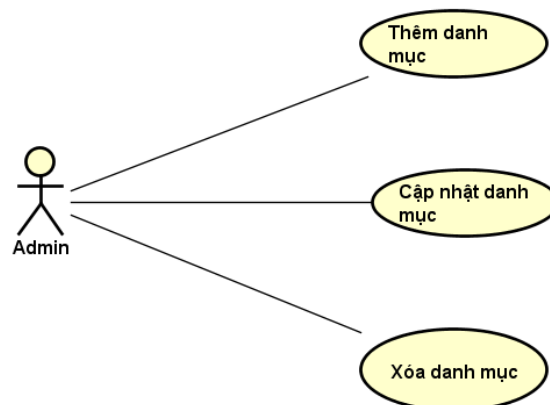
Hình 2 mô tả phân rã use case CRUD banner. Các use case phân rã được gồm: thêm banner, chỉnh sửa banner, xóa banner. Đây là các chức năng được thực hiện bởi admin. Admin có thể thêm mới một banner quảng cáo, chỉnh sửa các thông tin banner hoặc xóa banner. Những banner này được hiển thị trên màn hình trang chủ của ứng dụng.



Hình 2 Biểu đồ use case phân rã CRUD banner

2.2.3 Biểu đồ use case phân rã CRUD danh mục con

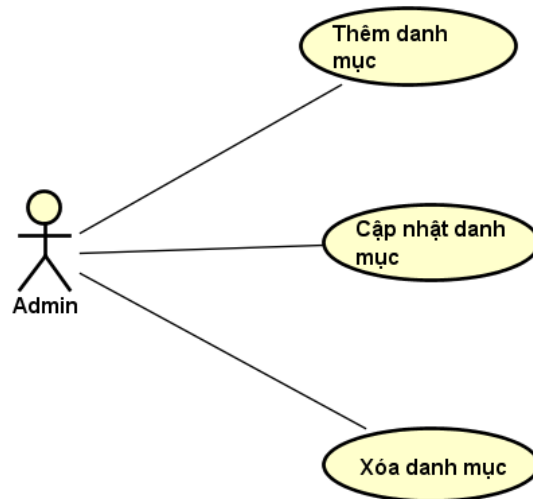
Hình 3 mô tả phân rã use case CRUD danh mục con. Chức năng này cũng được thực hiện bởi admin, bao gồm: thêm danh mục con, cập nhật thông tin danh mục (tên, hình ảnh, danh mục lớn) và xóa danh mục. Mỗi danh mục con này phải nằm trong một danh mục sản phẩm lớn được đề cập đến trong mục 2.2.4. Mỗi sản phẩm phải thuộc vào một danh mục con nhất định.



Hình 3 Biểu đồ use case phân rã CRUD danh mục con

2.2.4 Biểu đồ use case phân rã CRUD danh mục cha

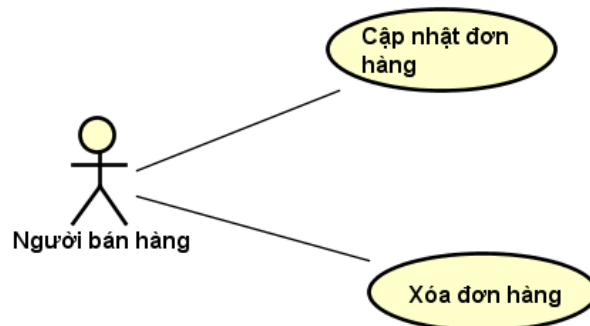
Hình 4 là biểu đồ use case phân rã CRUD danh mục cha. Sau khi phân rã được 3 use case: thêm danh mục, cập nhật danh mục, xóa danh mục. Danh mục cha bao gồm một hoặc nhiều danh mục sản phẩm con.



Hình 4 Biểu đồ use case phân rã CRUD danh mục cha

2.2.5 Biểu đồ use case phân rã UD đơn hàng

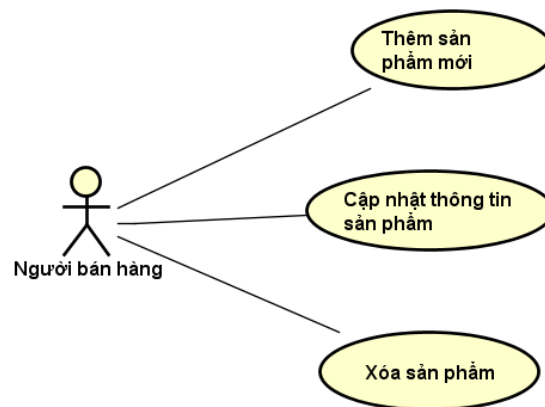
Hình 5 mô tả phân rã use case UD đơn hàng. Đây là use case được thực hiện bởi người bán hàng, bao gồm: cập nhật đơn hàng, xóa đơn hàng. Cập nhật đơn hàng là thao tác cập nhật trạng thái. Có 4 trạng thái của đơn hàng: đang đợi xử lý, đang giao hàng, đã giao hàng, đã hủy.



Hình 5 Biểu đồ use case phân rã UD đơn hàng

2.2.6 Biểu đồ use case phân rã CRUD sản phẩm

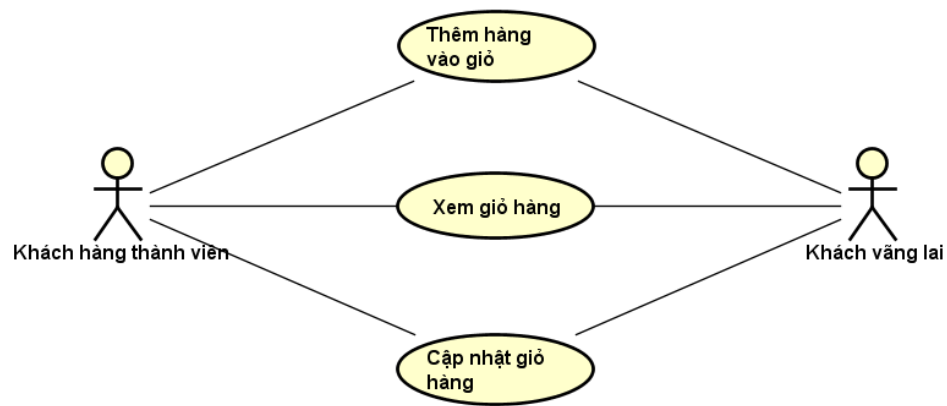
Hình 6 mô tả phân rã use case CRUD sản phẩm. Đây là một trong những use case quan trọng nhất của hệ thống, được thực hiện bởi tác nhân người bán hàng với điều kiện đã đăng ký mở một gian hàng thành công trên hệ thống. Sau khi phân rã được các use case sau: thêm sản phẩm, cập nhật thông tin sản phẩm, xóa sản phẩm. Trong số này, use case xóa sản phẩm cần được đặc biệt lưu tâm bởi có thể ảnh hưởng đến rất nhiều các thành phần khác. Chẳng hạn như xóa một sản phẩm đang tồn tại trong một đơn hàng, hoặc nằm trong danh sách ưa thích của một khách hàng thành viên nào đó...



Hình 6 Biểu đồ use case phân rã CRUD sản phẩm

2.2.7 Biểu đồ use case phân rã Quản lý giỏ hàng

Hình 7 mô tả phân rã use case Quản lý giỏ hàng. Các chức năng sau khi phân rã bao gồm: Thêm hàng vào giỏ, xem giỏ hàng, cập nhật giỏ hàng. Những chức năng này được thực hiện bởi cả khách hàng thành viên cũng như khách vãng lai mà không yêu cầu phải đăng nhập.

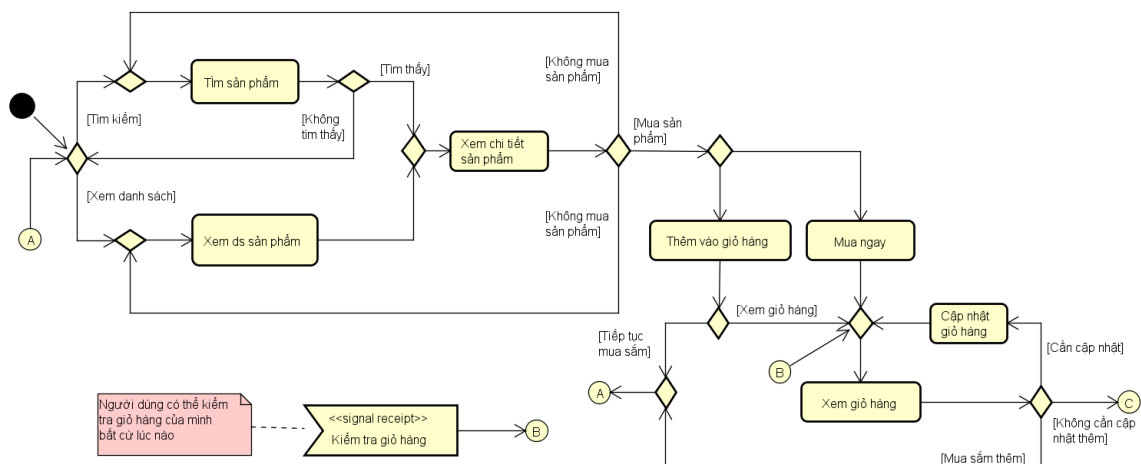


Hình 7 Biểu đồ use case phân rã Quản lý giỏ hàng

2.2.8 Quy trình nghiệp vụ

Sản phẩm cần xây dựng là một sàn thương mại điện tử, bởi vậy có hai quy trình nghiệp vụ quan trọng hàng đầu, đó là: quy trình mua hàng và quy trình thanh toán. Dưới đây là mô tả chi tiết về hai quy trình này.

Hình 8 mô tả biểu đồ hoạt động của quy trình mua hàng.

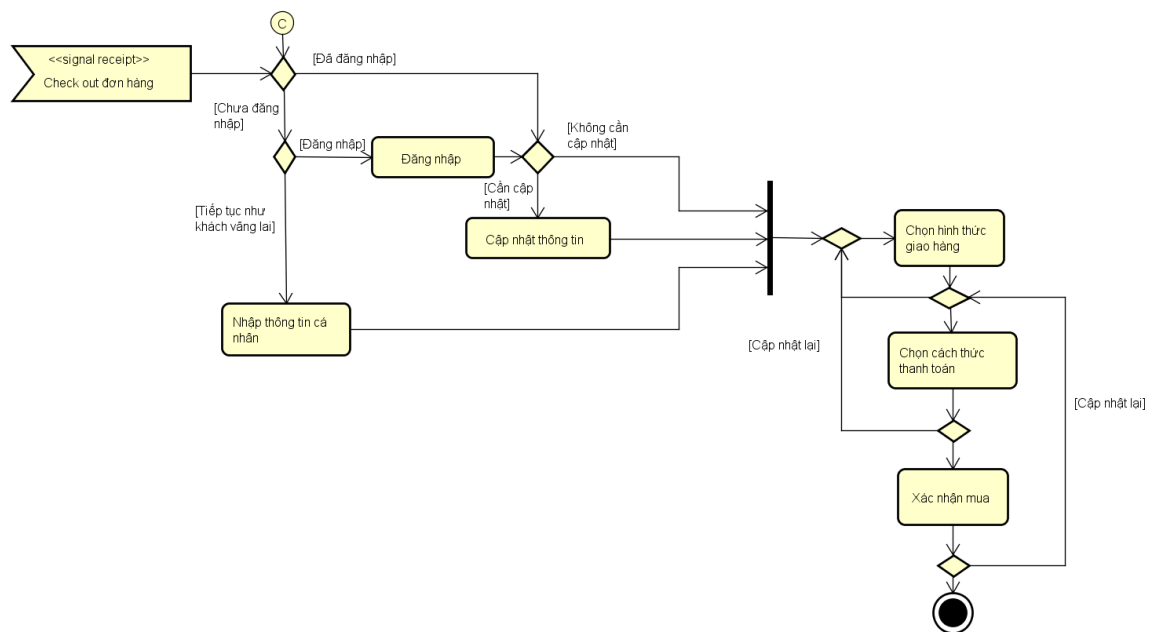


Hình 8 Biểu đồ hoạt động quy trình mua hàng

Đầu tiên, khách hàng thành viên hoặc khách vắng lai thực hiện thao tác xem danh sách sản phẩm hoặc tìm kiếm sản phẩm để truy xuất tới một nhóm các sản phẩm nhất định. Tiếp đó, người dùng chọn một sản phẩm mong muốn để xem chi tiết thông tin sản phẩm. Tại đây, sau khi đã xem xét và quyết định đặt mua, người dùng có hai lựa

chọn: “thêm vào giỏ hàng” hoặc “mua ngay”. Với lựa chọn “thêm vào giỏ hàng”, sản phẩm sẽ được thêm vào giỏ hàng và sau đó người dùng có thể tiếp tục chuyển sang thực hiện các thao tác khác. Với lựa chọn “mua ngay”, sản phẩm cũng được thêm vào giỏ hàng, đồng thời ngay lập tức chuyển sang giao diện giỏ hàng. Lưu ý rằng nếu sản phẩm đã tồn tại trong giỏ hàng thì người dùng không thể thêm vào nữa và có thể truy xuất đến thông tin giỏ hàng bất cứ lúc nào. Tại giỏ hàng, người dùng có thể cập nhật số lượng sản phẩm mong muốn hoặc xóa sản phẩm khỏi giỏ hàng. Sau các thao tác cập nhật giỏ hàng cần thiết, người dùng có thể chuyển đến bước tiếp theo – đặt hàng và thanh toán.

Hình 9 mô tả biểu đồ hoạt động của quy trình xác nhận đặt hàng.



Hình 9 Biểu đồ hoạt động quy trình xác nhận đặt hàng

Quy trình này là sự nối tiếp của quy trình mua hàng đã đề cập ở trên và chỉ có thể được thực hiện khi trong giỏ hàng của người dùng có ít nhất một mặt hàng. Bước đầu tiên, người dùng cần xác nhận lại các thông tin cá nhân sẽ xuất hiện trong đơn hàng, đó là: tên khách hàng, điện thoại, địa chỉ. Trong trường hợp là khách hàng thành viên đã đăng nhập, những thông tin đó sẽ được tự động cập nhật và hiển thị, trong trường hợp là khách vắng mặt, người dùng cần nhập lại các thông tin này bằng tay. Sau khi

đã hoàn thiện và xác nhận các thông tin cá nhân, người dùng chuyển sang bước tiếp theo, đó là lựa chọn hình thức vận chuyển hàng. Từ danh sách các hình thức vận chuyển, người dùng lựa chọn một hình thức nhất định và chuyển sang bước tiếp. Ở bước thứ ba này, tương tự như bước trước đó, người dùng cần lựa chọn một hình thức thanh toán từ danh sách trước khi chuyển sang bước cuối. Bước cuối là bước xác nhận, đơn hàng với đầy đủ thông tin tích hợp từ các bước trước đó được cập nhật và hiển thị, kèm với tổng số tiền cần thanh toán. Người dùng review lại một lượt các thông tin, nếu không cần thay đổi gì, người dùng chọn “Xác nhận”, đơn hàng sẽ được tạo và thêm vào cơ sở dữ liệu. Lưu ý rằng trong từng bước của quy trình xác nhận đặt hàng, người dùng đều có thể quay lại bước liền ngay trước đó để thay đổi thông tin nếu muốn.

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case Đặt hàng

Bảng 2 là đặc tả chi tiết use case “Đặt hàng”, đây là use case được thực hiện bởi tác nhân khách hàng thành viên hoặc khách vắng lai.

Bảng 2 Đặc tả use case Đặt hàng

Tên use case	Đặt hàng	
Tác nhân	Khách vắng lai hoặc khách hàng thành viên (Người dùng).	
Điều kiện trước	Trong giỏ hàng tồn tại ít nhất một sản phẩm.	
Luồng thực thi chính		
No.	Thực hiện	Hành động

1	Người dùng	Chọn “Tiến hành thanh toán”.
2	Hệ thống	Lấy thông tin người dùng hiện tại, hiển thị lên màn hình: tên người dùng, số điện thoại và các địa chỉ mà người dùng đã đăng ký để nhận hàng.
3	Người dùng	Chọn địa chỉ nhận hàng hoặc thêm địa chỉ mới.
4	Người dùng	Bấm nút “Bước tiếp: Vận chuyển”.
5	Người dùng	Chọn hình thức vận chuyển.
6	Người dùng	Chọn “Bước tiếp: Cách thức thanh toán”.
7	Người dùng	Chọn hình thức thanh toán.
8	Người dùng	Chọn “Bước tiếp: Xác nhận đặt hàng”.
9	Hệ thống	Tổng hợp và hiển thị thông tin đơn hàng.
10	Người dùng	Chọn “Xác nhận”.
11	Hệ thống	Thêm thông tin đơn hàng vào cơ sở dữ liệu.

Luồng thực thi mở rộng

No.	Thực hiện	Hành động
2a	Hệ thống	Yêu cầu người dùng nhập thông tin: tên, số điện thoại, địa chỉ (trường hợp là khách vắng lai).
3a	Người dùng	Nhập thông tin: tên, số điện thoại, địa chỉ. (trường hợp là khách vắng lai).

Điều kiện Hiển thị kết quả thêm đơn hàng thành công hay thất bại.
sau

2.3.2 Đặc tả use case Thêm sản phẩm mới

Bảng 3 đặc tả chi tiết use case “Thêm sản phẩm mới”, đây là use case được thực hiện bởi tác nhân Người bán hàng.

Bảng 3 Đặc tả use case Thêm sản phẩm mới

Tên use case	Thêm sản phẩm mới	
Tác nhân	Người bán hàng.	
Điều kiện trước	Người bán hàng đã đăng nhập và đã đăng ký mở cửa hàng thành công.	
Luồng thực thi chính		
No.	Thực hiện	Hành động
1	Người bán hàng	Tại giao diện Cá nhân, chọn “Cửa hàng của bạn”.
2	Người bán hàng	Chọn “Thêm sản phẩm mới”.
3	Hệ thống	Hiển thị giao diện thêm sản phẩm.
4	Người bán hàng	Nhập tên sản phẩm.
5	Người bán hàng	Chọn “Thêm ảnh”.
6	Người bán hàng	Chọn “Chụp ảnh”.

7	Người bán hàng	Chụp ảnh bằng thiết bị.
8	Người bán hàng	Chọn “Upload ảnh”.
9	Hệ thống	Upload ảnh lên server và trả về thông báo thành công.
10	Người bán hàng	Chọn “Thương hiệu” và “Danh mục”.
11	Người bán hàng	Nhập liệu: “Khối lượng”, “Mô tả”, “Số lượng”, “Giá tiền”.
12	Người bán hàng	Chọn “Thêm”.
13	Hệ thống	Thêm sản phẩm vào cơ sở dữ liệu.

Luồng thực thi mở rộng

No.	Thực hiện	Hành động
6a	Người bán hàng	Chọn “Chọn ảnh từ thiết bị”.
7a	Người bán hàng	Chọn ảnh từ thư viện ảnh của thiết bị.
9a	Hệ thống	Thông báo thêm ảnh thất bại. Trường hợp này có thể do 2 nguyên nhân chính: (i) Lỗi từ server, (ii) Ảnh đã chọn có kích thước quá lớn. Người dùng có thể quay lại thực hiện từ bước 5.

Điều kiện sau Hiển thị kết quả thêm sản phẩm thành công hay thất bại.

2.3.3 Đặc tả use case Cập nhật trạng thái đơn hàng

Bảng 4 đặc tả use case “Cập nhật trạng thái đơn hàng”, use case này cũng được thực hiện bởi người bán hàng.

Bảng 4 Đặc tả use case Cập nhật trạng thái đơn hàng

Tên use case	Cập nhật trạng thái đơn hàng	
Tác nhân	Người bán hàng.	
Điều kiện trước	Người bán hàng đã đăng nhập và đã đăng ký mở cửa hàng thành công. Cửa hàng đã có ít nhất một đơn đặt hàng.	
Luồng thực thi chính		
No.	Thực hiện	Hành động
1	Người bán hàng	Tại giao diện Cá nhân, chọn “Cửa hàng của bạn”.
2	Người bán hàng	Chọn “Danh sách đơn hàng”.
3	Hệ thống	Lấy thông tin danh sách đơn hàng của cửa hàng và hiển thị lên giao diện.
4	Người bán hàng	Chọn “Cập nhật trạng thái”.
5	Người bán hàng	Chọn trạng thái phù hợp.
6	Người bán hàng	Chọn “Cập nhật”.
7	Hệ thống	Cập nhật trạng thái đơn hàng trong cơ sở dữ liệu.
Luồng thực thi mở rộng		

No.	Thực hiện	Hành động
3a	Hệ thống	Hiển thị thông báo không có dữ liệu (Trường hợp cửa hàng chưa có đơn hàng).
Điều kiện sau	Hiển thị kết quả cập nhật trạng thái đơn hàng thành công hay thất bại.	

2.3.4 Đặc tả use case Đăng ký mở gian hàng

Bảng 5 đặc tả use case “Đăng ký mở gian hàng”. Use case này được thực hiện bởi người bán hàng. Trước khi có thể đăng bán sản phẩm, người bán hàng cần đăng ký mở gian hàng thành công.

Bảng 5 Đặc tả use case Đăng ký mở gian hàng

Tên use case Đăng ký mở gian hàng

Tác nhân Khách hàng thành viên.

Điều kiện trước Đã đăng nhập thành công.

Luồng thực thi chính

No.	Thực hiện	Hành động
1	Khách hàng thành viên	Tại giao diện Cá nhân, chọn “Cửa hàng của bạn”.
2	Hệ thống	Thông báo chưa tạo cửa hàng, yêu cầu người dùng chuyển đến trang đăng ký tạo cửa hàng.

3	Khách hàng thành viên	Chọn “Đăng ký cửa hàng”.
4	Hệ thống	Hiển thị giao diện đăng ký cửa hàng.
5	Khách hàng thành viên	Nhập liệu các thông tin: Tên cửa hàng (bắt buộc), slogan (bắt buộc), số điện thoại (bắt buộc), địa chỉ (bắt buộc), địa chỉ website (không bắt buộc).
6	Khách hàng thành viên	Tại mục “Ảnh đại diện”, chọn “Thêm ảnh”.
7	Khách hàng thành viên	Chọn “Chụp ảnh”.
8	Khách hàng thành viên	Chụp ảnh từ thiết bị
9	Khách hàng thành viên	Chọn “Upload ảnh”.
10	Hệ thống	Upload ảnh lên server và trả về thông báo thành công.
11	Khách hàng thành viên	Tại mục “Ảnh bìa”, chọn “Thêm ảnh”
12	Khách hàng thành viên	Tương tự bước 7.
13	Khách hàng thành viên	Tương tự bước 8.

14	Khách hàng thành viên	Tương tự bước 9.
15	Hệ thống	Tương tự bước 10.
16	Khách hàng thành viên	Chọn đăng ký.
17	Hệ thống	Thêm thông tin cửa hàng vào cơ sở dữ liệu.
Luồng thực thi mở rộng		
No.	Thực hiện	Hành động
7a	Khách hàng thành viên	Chọn “Chọn ảnh từ thiết bị”.
8a	Khách hàng thành viên	Chọn ảnh từ thư viện ảnh của thiết bị.
10a	Hệ thống	Thông báo thêm ảnh thất bại. Trường hợp này có thể do 2 nguyên nhân chính: (i) lỗi từ server, (ii) ảnh đã chọn có kích thước quá lớn. Người dùng có thể quay lại thực hiện từ bước 6.
12a	Khách hàng thành viên	Tương tự bước 7a.
13a	Khách hàng thành viên	Tương tự bước 8a.
15a	Khách hàng thành viên	Tương tự bước 10a. Người dùng có thể quay lại thực hiện từ bước 11.

Điều kiện Hiện thị kết quả cập nhật mở cửa hàng thành công hay thất bại.
sau

2.3.5 Đặc tả use case **Đánh giá sản phẩm**

Bảng 6 đặc tả use case “Đánh giá sản phẩm” được thực hiện sau khi người dùng đã đăng nhập thành công.

Bảng 6 Đặc tả use case Đánh giá sản phẩm

Tên use case		Đánh giá sản phẩm
Tác nhân		Khách hàng thành viên.
Điều kiện trước		Đã đăng nhập thành công.
Luồng thực thi chính		
No.	Thực hiện	Hành động
1	Khách hàng thành viên	Chọn sản phẩm muốn đánh giá.
2	Hệ thống	Lấy thông tin chi tiết sản phẩm và hiển thị.
3	Khách hàng thành viên	Chọn “Đánh giá sản phẩm”.
4	Khách hàng thành viên	Chọn số sao đánh giá.

5	Khách hàng thành viên	Nhập liệu các trường thông tin: Tiêu đề đánh giá (bắt buộc), Nội dung (không bắt buộc).
6	Khách hàng thành viên	Chọn “Đánh giá”.
7	Hệ thống	Thêm đánh giá vào cơ sở dữ liệu.

Luồng thực thi mở rộng

No.	Thực hiện	Hành động
2a	Hệ thống	Vô hiệu hóa lựa chọn đánh giá sản phẩm (Trường hợp người dùng đã đánh giá sản phẩm này).

Điều kiện sau Hiện thị kết quả đánh giá thành công hay thất bại. Trường hợp đánh giá thành công, ngay lập tức vô hiệu lựa chọn đánh giá sản phẩm này đối với người dùng hiện tại.

2.4 Yêu cầu phi chức năng

Bên cạnh yêu cầu chức năng đã được đặc tả trong các phần trên, sản phẩm cần đảm bảo một số yêu cầu phi chức năng khác. Đầu tiên, ứng dụng cần dễ thao tác, dễ sử dụng với người dùng phổ thông. Các biểu tượng, kí hiệu hay icon hình ảnh sử dụng cần mang tính phổ biến, không gây lạ lẫm hoặc làm rối mắt. Dữ liệu nhập vào từ người dùng cần được validate cẩn thận trước khi gửi lên server xử lý hoặc thực hiện các nghiệp vụ khác. Ứng dụng cần có khả năng xử lý và chịu lỗi. Với mỗi lỗi phát sinh từ hệ thống cần được lường trước và kiểm soát cẩn thận, khi cần thiết có thể xuất thông báo phù hợp cho người dùng. Tuyệt đối tránh trường hợp ứng dụng bị dừng đột ngột mà người dùng không biết nguyên nhân là gì. Bên cạnh đó cũng cần tính toán đến khả năng phục hồi và quản lý tài nguyên, đặc biệt là dữ liệu phía server. Dữ liệu này cần được backup thường xuyên. Trong thời gian đầu triển khai ứng dụng, vì

chưa có khả năng xây dựng một server backup, có thể đưa vào một số giải pháp thay thế như sử dụng dịch vụ lưu trữ đám mây của bên thứ ba. Ứng dụng cũng cần được xây dựng theo một mô hình linh hoạt, mô đun hóa được các chức năng nhằm dễ bảo trì và mở rộng sau này. Dữ liệu đảm bảo đạt chuẩn 3 – NF, thời gian truy xuất nhanh để đảm bảo tốc độ và hiệu năng ứng dụng.

Chương 2 đã trình bày đầy đủ các yêu cầu chức năng và phi chức năng của ứng dụng, Yêu cầu chức năng được thể hiện qua các biểu đồ use case tổng quan, use case phân rã và biểu đồ hoạt động mô tả hai quy trình nghiệp vụ chính. Nội dung Chương 2 còn bao gồm các bảng đặc tả use case quan trọng, nội dung được đặc tả cho mỗi use case bao gồm: tên, tác nhân thực hiện, điều kiện trước, luồng thực thi chính, luồng thực thi mở rộng, điều kiện sau. Vậy những công nghệ nào đã được áp dụng trong sản phẩm? Điều này sẽ được em trình bày trong chương tiếp theo – Chương 3.

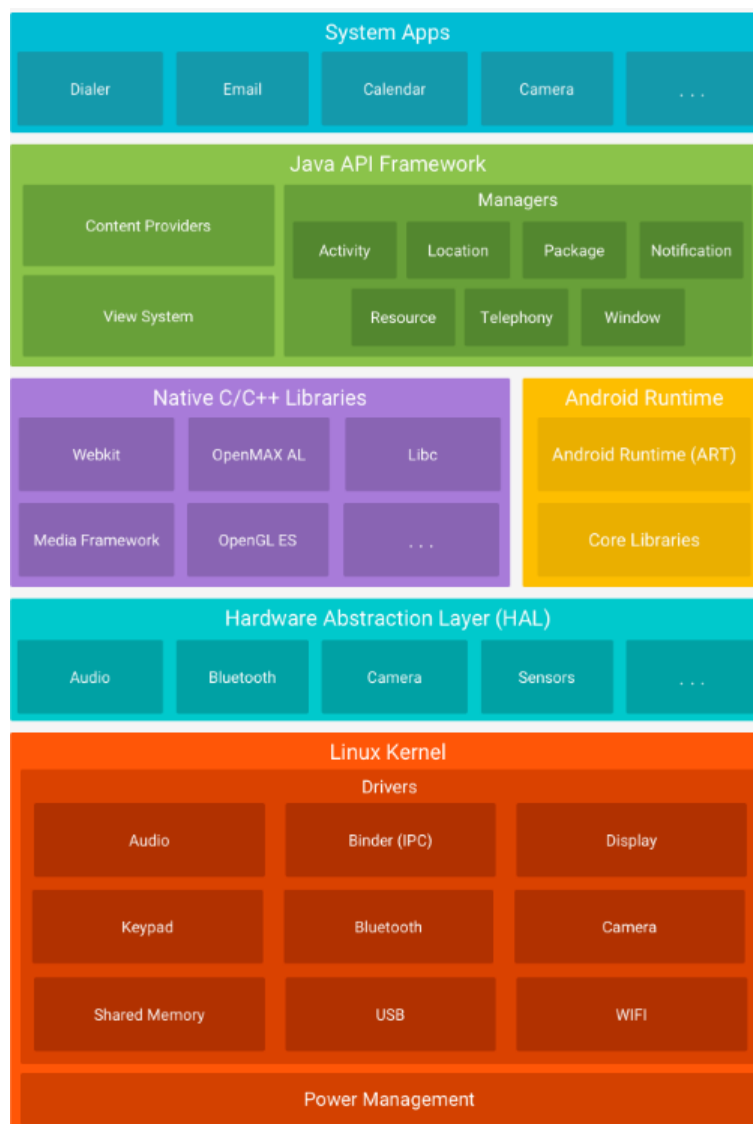
Chương 3 Công nghệ sử dụng

Từ những yêu cầu cụ thể đã được nêu ra trong Chương 2, Chương 3 này sẽ tập trung đi sâu vào công nghệ mà em sử dụng trong đồ án. Công nghệ ở đây bao gồm cả hai phía: phía ứng dụng client và phía server. Nội dung trình bày ở mỗi công nghệ bao gồm những kiến thức tổng quan về công nghệ, lý do sử dụng công nghệ đó để phát triển sản phẩm và những kỹ thuật chính được áp dụng. Hai nền tảng công nghệ tương ứng với hai phía của sản phẩm được đề cập đến là: Android và PHP.

3.1 Android

3.1.1 Tổng quan về Android

Android là hệ điều hành mã nguồn mở, dựa trên nền tảng Linux, được tạo ra cho một lượng lớn các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Kiến trúc của Android giống như một ngăn xếp các thành phần phần mềm, được chia thành 6 phần: (i) Linux Kernel (ii) Hardware Abstraction Layer (iii) Native C/C++ Libraries, (iv) Android Runtime, (v) Java API Framework (vi) System Apps và 5 lớp chính. Hình 10 sau đây cho thấy cụ thể hơn về các thành phần chính trong kiến trúc của hệ điều hành Android.



Hình 10 Kiến trúc hệ điều hành Android³

Nền tảng cơ bản của Android là nhân Linux. Sử dụng hạt nhân Linux cho phép Android tận dụng các tính năng bảo mật quan trọng và cho phép các nhà sản xuất thiết bị phát triển các trình điều khiển phần cứng. Lớp phần cứng trừu tượng (HAL) cung cấp các giao diện chuẩn để các framework Java API ở mức cao hơn có thể tương tác. HAL bao gồm nhiều mô đun thư viện, mỗi mô đun thực hiện giao diện cho một loại thành phần phần cứng cụ thể, chẳng hạn như mô đun máy ảnh hoặc mô đun bluetooth. Đối với các thiết bị chạy Android phiên bản 5.0 (API cấp 21) hoặc cao

³ <https://developer.android.com/>, lần truy cập cuối: 21/05/2018

hơn, mỗi ứng dụng chạy trong quy trình riêng của nó và với phiên bản Android Runtime (ART) của riêng nó. ART được viết để chạy nhiều máy ảo trên các thiết bị bộ nhớ thấp bằng cách thực thi các tệp DEX. Nhiều thành phần và dịch vụ hệ thống cốt lõi của Android được xây dựng từ mã nguồn yêu cầu các thư viện gốc được viết bằng C và C++. Ở tầng trên cùng, Android cung cấp sẵn một tập những ứng dụng cốt lõi cho các yêu cầu thiết yếu nhất như email, nhắn tin SMS, lịch, duyệt web, danh bạ và cài đặt. Một ứng dụng của bên thứ ba có thể trở thành trình duyệt web mặc định của người dùng, tin nhắn SMS hoặc thậm chí là bàn phím mặc định (một số ngoại lệ được áp dụng, chẳng hạn như ứng dụng Cài đặt của hệ thống). Trong phạm vi của đề tài, sản phẩm sẽ chủ yếu được xây dựng trên tầng ứng dụng của kiến trúc Android.

Một ứng dụng Android có thể viết bằng các ngôn ngữ lập trình: Java, Kotlin và C++. Công cụ Android SDK biên dịch mã nguồn cùng các file dữ liệu và tài nguyên thành một tệp tin APK – một Android package. Thiết bị Android sẽ sử dụng tệp tin này để cài đặt ứng dụng. Một ứng dụng Android có 4 thành phần cơ bản: (i) Activities, (ii) Services, (iii) Broadcast receivers, (iv) Content providers. Mỗi loại phục vụ một mục đích riêng biệt và có một vòng đời riêng biệt xác định cách mà nó được tạo ra và bị phá hủy. Một Activity là một điểm đầu vào tương tác với người dùng. Nó biểu diễn một màn hình đơn với một giao diện người dùng. Service là một thành phần chạy dưới nền nhằm thực hiện các hoạt động dài hạn hoặc thực hiện các công việc cho các quá trình điều khiển từ xa. Một service không cung cấp một giao diện người dùng. Broadcast receivers cho phép hệ thống cung cấp sự kiện cho ứng dụng bên ngoài luồng người dùng thông thường, đồng thời cho phép ứng dụng phản hồi các thông báo phát sóng trên toàn hệ thống. Content provider quản lý một tập các dữ liệu được chia sẻ của ứng dụng. Dữ liệu này có thể được lưu trữ trong file hệ thống, trong cơ sở dữ liệu SQLite, trên web hoặc trong bất cứ vị trí lưu trữ liên tục nào đó mà ứng dụng có thể truy cập.

3.1.2 Lý do sử dụng

Android có nhiều ưu điểm vượt trội để trở thành sự lựa chọn hàng đầu cho việc phát triển một ứng dụng di động. Trước hết, về nền tảng phần cứng, số lượng thiết bị di

động sử dụng hệ điều hành Android hiện vẫn chiếm phần lớn so với các hệ điều hành khác. Điều này giúp ứng dụng phát triển trên nền tảng này có nhiều cơ hội tiếp cận với người dùng hơn. Số lượng các thiết bị đa dạng với nhiều kích thước màn hình khác nhau, nhiều phiên bản khác nhau cũng buộc người phát triển phải có những tính toán tốt nhất để tạo ra ứng dụng có độ tương thích cao và khả năng co giãn linh hoạt. Tiếp đó, về công cụ sử dụng, một ứng dụng Android được phát triển chủ yếu dựa trên mã nguồn Java – ngôn ngữ lập trình mã nguồn mở với lịch sử lâu dài. Java có cú pháp rõ ràng, dễ hiểu và thuần hướng đối tượng, bởi vậy việc áp dụng các kiến thức OOP (Object-Oriented Programming – lập trình hướng đối tượng) rất dễ dàng và hiệu quả, đặc biệt khi so sánh với một số ngôn ngữ khác cũng dùng để lập trình di động như Objective-C hay Swift trên iOS. Hiện nay, ứng dụng Android còn có thể được phát triển bằng một ngôn ngữ lập trình mới đầy hứa hẹn là Kotlin. Lập trình viên có thêm sự lựa chọn để tạo ra sản phẩm tốt nhất, thậm chí có thể kết hợp cả hai trong cùng một project. Mặc dù đã có rất nhiều các công nghệ cross-platform cho lập trình Android nói riêng và di động nói chung xuất hiện trong thời gian gần đây với ưu điểm nổi bật là tiết kiệm chi phí, công sức phát triển nhưng thực tế còn tồn tại nhiều hạn chế như: hiệu năng chưa thực sự ổn định và thiếu khả năng can thiệp sâu vào hệ thống. Sự lựa chọn phát triển một ứng dụng “native” vẫn được đặt lên hàng đầu. Cuối cùng là về quy trình phát triển. Quy trình phát triển Android là dễ tiếp cận, kể cả với người chưa có nhiều kinh nghiệm, được hỗ trợ bởi một cộng đồng rộng lớn cùng những bộ thư viện mã nguồn mở giúp đơn giản hóa rất nhiều những thao tác phức tạp. Sản phẩm sau khi hoàn thành có thể nhanh chóng đến được với người dùng, nhờ đó lập trình viên sớm nhận được phản hồi để kịp thời khắc phục, sửa chữa và cải thiện.

3.1.3 Kỹ thuật áp dụng

Ứng dụng được xây dựng có đặc trưng là tần suất tương tác tương đối cao với phía server. Hầu hết các chức năng đều cần lấy dữ liệu từ server và hiển thị lên giao diện người dùng. Điều này kéo theo một loạt các tác vụ bất đồng bộ cần được xử lý và một phương pháp quản lý đa luồng hiệu quả. Android cung cấp các cơ chế sau để giải

quyết vấn đề này, đó là: Thread, Handler và AsyncTask. Trong đó, AsyncTask là cơ chế đơn giản và hiệu quả hơn cả. Lớp này cho phép thực hiện các công việc dưới nền (background) và xuất các kết quả trên luồng giao diện người dùng (main UI thread) mà không cần thực hiện bất kỳ thao tác quản lý nào khác. Đây cũng là kỹ thuật xử lý đa tiến trình được áp dụng chủ yếu trong đề án. Bên cạnh việc gửi request liên tục lên server để lấy dữ liệu, một số thông tin cần được lưu trữ cục bộ tại thiết bị điện thoại để dễ dàng truy cập và thực hiện các nghiệp vụ cần thiết, chẳng hạn thông tin về giỏ hàng hay thông tin người dùng đã đăng nhập vào ứng dụng. Để quản lý dữ liệu được lưu trữ ngay trên thiết bị điện thoại, Android hỗ trợ nhiều cách khác nhau, hai trong số đó được em áp dụng là: SQLite và SharedPreferences. SQLite là một hệ quản trị cơ sở dữ liệu thu nhỏ được Android tích hợp sẵn trong các thiết bị của mình. Ưu điểm của SQLite là sự nhỏ gọn nhưng vẫn cung cấp đầy đủ các tính năng như một hệ quản trị cơ sở dữ liệu thông thường. SQLite được ứng dụng để lưu trữ các thông tin trong giỏ hàng và các dữ liệu về lịch sử tìm kiếm của khách hàng thành viên, khi người dùng đăng xuất hoặc gỡ bỏ ứng dụng, những dữ liệu đó cũng bị xóa theo. Nếu như SQLite lưu trữ theo các bảng dữ liệu quan hệ thì SharedPreferences lại lưu trữ theo từng cặp key – value. Tư tưởng cơ bản của SharedPreferences chỉ là làm cách nào để có thể ghi dữ liệu (Write to Shared Preferences) và đọc dữ liệu đã lưu (Read from Shared Preferences). Sử dụng SharedPreferences rất thuận tiện trong trường hợp ta muốn lưu trữ nhanh các thông tin đăng nhập của người dùng và lấy ra sử dụng trong một chức năng nào đó của ứng dụng.

3.2 PHP

3.2.1 Tổng quan về PHP

PHP viết tắt của cụm từ "Hypertext Preprocessor", là một ngôn ngữ lập trình kịch bản (scripting language) chủ yếu được dùng để phát triển các trang web động hoặc các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Khác với các ngôn ngữ client-side, như Javascript, mã lệnh PHP được thực thi ở phía server sau đó mã HTML được trả về cho trình duyệt (client). Cú pháp PHP được vay mượn

nhiều từ C và Java. PHP hiện đã trải qua nhiều phiên bản khác nhau, phiên bản mới nhất hiện này là PHP 7 với những bước nhảy vọt lớn như: cập nhật hệ thống Core Zend Engine hoàn toàn mới, tốc độ xử lý tăng lên gấp 2 lần, quản lý lỗi dễ dàng hơn, hỗ trợ các hệ thống Windows 64 bit, bổ sung thêm các toán tử và class⁴...

PHP chủ yếu tập trung vào kịch bản phía máy chủ, vì vậy nó có thể thực hiện rất nhiều các công việc như thu thập dữ liệu, tạo ra các trang web động, gửi và nhận cookie... Ba lĩnh vực chính mà mã PHP được sử dụng bao gồm: (i) Kịch bản phía máy chủ (Server-side scripting), (ii) Kịch bản cho Command line (Command line scripting) và (iii) Viết ứng dụng desktop. Trong đó, lĩnh vực đầu tiên là mục tiêu chính mà PHP hướng tới, cũng là lĩnh vực chủ yếu được áp dụng trong đồ án. Để thực hiện được kịch bản phía máy chủ cần sử dụng 3 công cụ sau: trình thông dịch PHP, web server và trình duyệt web. Một web server (đã được cài đặt PHP) được khởi chạy ban đầu, sau đó các file PHP được thực thi, và cuối cùng là tiếp nhận kết quả mà web server trả về trên trình duyệt. PHP có thể chạy được trên tất cả các hệ điều hành, bao gồm cả Linux, nhiều phiên bản của Unix (bao gồm cả HP-UX, Solaris và OpenBSD), Microsoft Windows, Mac OS X, RISC OS, và nhiều hệ điều hành khác. PHP cũng đã hỗ trợ cho hầu hết các máy chủ web ngày nay (bao gồm Apache, IIS và nhiều máy chủ web khác).

Một trong những tính năng mạnh nhất và quan trọng nhất của PHP là hỗ trợ một loạt các hệ quản trị cơ sở dữ liệu (CSDL), như MySQL, MSSQL, mSQL, PostgreSQL, SQLite... Và tương ứng với mỗi hệ quản trị đó, là một driver để kết nối PHP với chúng, ví dụ ODBC cho MSSQL, JDBC cho MySQL... PHP cũng đã hỗ trợ giao tiếp với các dịch vụ sử dụng các giao thức khác nhau như LDAP, IMAP, SNMP, NNTP, POP3, COM... và nhiều giao thức khác. PHP cũng hỗ trợ WDDX, giúp trao đổi dữ liệu giữa hầu hết các ngôn ngữ lập trình web.

⁴ <http://php.net/>, lần truy cập cuối 20/5/2018

3.2.2 Lý do sử dụng

Từ phân tổng quan đã nêu ở trên, có thể rút ra được nhiều lý do để lựa chọn PHP làm nền tảng công nghệ chính cho việc phát triển phía server. Điều đầu tiên, PHP được tối ưu tốt cho các ứng dụng web nên tốc độ thực thi nhanh và nhỏ gọn, thời gian xây dựng sản phẩm tương đối ngắn so với các ngôn ngữ lập trình khác. Do có cú pháp tương đồng với C và Java nên việc tiếp cận là rất dễ dàng. Từ PHP 5, OOP đã được hỗ trợ và lập trình viên có thể tùy ý lựa chọn sử dụng lập trình hướng thủ tục hoặc lập trình hướng đối tượng, hoặc hỗn hợp cả hai. Bên cạnh đó, PHP có rất nhiều các CMS mã nguồn mở như WordPress, OpenCart, NukeViet nên rất đa dạng và đáp ứng tốt hầu hết các công việc xử lý ở server, không chỉ phục vụ riêng cho phía client là mobile app mà còn cho các website kinh doanh trực tuyến, đặc biệt hữu ích khi muốn mở rộng hệ thống sau này. Về khả năng tương thích, như đã nói ở trên, PHP hỗ trợ hầu hết các hệ điều hành cũng như các hệ quản trị cơ sở dữ liệu phổ biến. Việc cài đặt, kết nối cũng như truy xuất dữ liệu được thực hiện nhanh chóng và dễ dàng chuyển đổi sang các công nghệ khác. Người phát triển có thể tự do lựa chọn một hệ điều hành, một máy chủ web và một cơ sở dữ liệu phù hợp cho hệ thống của mình. Cuối cùng là về cộng đồng hỗ trợ, PHP là ngôn ngữ lập trình mã nguồn mở được hỗ trợ bởi một cộng đồng đông đảo các lập trình viên nhiều kinh nghiệm. Số lượng thư viện và framework đa dạng cùng tài liệu rõ ràng cũng là một lý do quan trọng khiến PHP luôn dẫn đầu về độ phổ biến trong số các ngôn ngữ lập trình web hiện nay.

3.2.3 Kỹ thuật áp dụng

Như đã nói ở mục 2.1, phía server cần được xây dựng sao cho có khả năng tương tác linh hoạt với một hệ quản trị cơ sở dữ liệu ổn định, độ tin cậy tốt. Ở khía cạnh này, sự kết hợp giữa PHP và MySQL được đánh giá cao hơn cả bởi sự đơn giản trong thao tác kết nối và cấu hình cũng như hiệu năng ổn định, có thể đáp ứng được cho các sản phẩm quy mô vừa và nhỏ đến các hệ thống lớn. MySQL là một hệ quản trị cơ sở dữ liệu với đặc điểm là mã nguồn mở tương tự PHP, cú pháp truy vấn đơn giản, rõ ràng, dễ hiểu và khả năng bảo mật tốt. Để bắt đầu triển khai một server với PHP và MySQL,

công việc cần làm chỉ là tải và cài đặt XAMPP – một phần mềm được đóng gói từ Apache, MySQL, PHP và Pearl. Sau đó, ta có thể có đầy đủ các công cụ để bắt đầu xây dựng server side, từ ngôn ngữ lập trình, cơ sở dữ liệu, đến một máy chủ ảo ở localhost để kiểm tra kết quả thực thi. Sự đơn giản trong quá trình cài đặt cũng như dễ dàng triển khai cũng là một trong những yếu tố quan trọng để em quyết định lựa chọn PHP và MySQL là những nền tảng công nghệ chính phía server của sản phẩm.

Trong tương tác giữa client và server, định dạng dữ liệu mà phía server trả về cũng là một yếu tố cần quan tâm. Dữ liệu trả về cần được tổ chức một cách khoa học, dễ hiểu, đảm bảo thông tin đầy đủ, đáp ứng chính xác request mà phía ứng dụng client gửi lên. Tuy nhiên, dữ liệu cũng cần gọn nhẹ để thao tác truyền không tiêu tốn quá nhiều tài nguyên và có tốc độ đảm bảo. Hiện nay, hai định dạng dữ liệu được sử dụng phổ biến là: JSON (JavaScript Object Notation) và XML (Extensible Markup Language). JSON được xây dựng từ Javascript, ngôn ngữ được sử dụng nhiều, tương thích với cả front – end và back – end của cả web app và web service. Đây là một định dạng đơn giản với 2 thành phần: keys và values: key thể hiện thuộc tính của Object và value thể hiện giá trị của từng Key. Nếu như trong JSON các cặp dấu { } và [] được dùng để đánh dấu dữ liệu thì XML lại có cấu trúc tương tự như HTML, dùng thẻ để đánh dấu và được gọi là nodes. Về cơ bản, hai định dạng dữ liệu này đều có ưu điểm riêng và có nhiều thư viện hỗ trợ các thao tác đọc ghi. Tuy nhiên, JSON có nhiều đặc điểm nổi trội so với XML như cấu trúc đơn giản hơn, dễ dàng sử dụng và truy vấn hơn, không phải đọc theo từng node và đặt tên cho các thẻ. PHP có sẵn hàm hỗ trợ encode dữ liệu thành định dạng JSON và decode một chuỗi JSON bất kì thành mảng đối tượng mong muốn.

Như vậy, toàn bộ nội dung Chương 3 đã trình bày chi tiết về những công nghệ được sử dụng trong đồ án, bối cảnh cũng như bài toán cụ thể mà những công nghệ này được áp dụng. Về cơ bản những công nghệ này đều đã đáp ứng tốt những yêu cầu chức năng và phi chức năng mà sản phẩm cần đạt được. Trong Chương 4 tới, em sẽ tiếp tục trình bày về thiết kế của toàn bộ ứng dụng cũng như quá trình xây dựng ứng dụng.

Chương 4 Phát triển và triển khai ứng dụng

Chương 3 đã đề cập đến những công nghệ được vận dụng trong toàn bộ đồ án. Dựa trên nền tảng công nghệ đó, Chương 4 sẽ trình bày về thiết kế của ứng dụng, từ thiết kế tổng quan đến chi tiết. Bên cạnh đó, em sẽ đưa ra công cụ và thư viện cụ thể được áp dụng trong quá trình lập trình xây dựng sản phẩm. Chương 4 này cũng là phần trình bày về những kết quả đạt được của đồ án và minh họa một số chức năng chính của sản phẩm. Phần kết chương đưa ra một số nội dung về kiểm thử cũng như triển khai của sản phẩm trong thực tế.

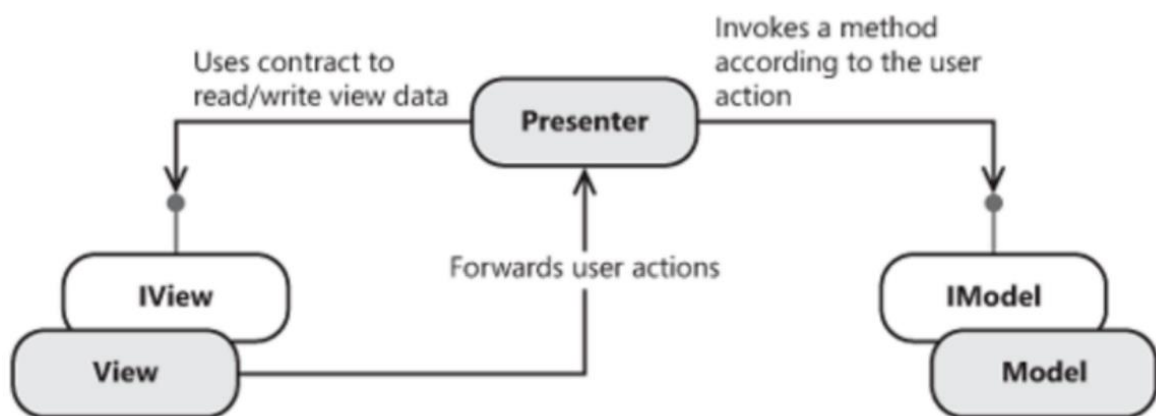
4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

Một hệ thống được xây dựng trên nền tảng kiến trúc tốt sẽ mang lại nhiều lợi ích: khả năng phát triển linh hoạt, dễ dàng bảo trì và mở rộng, hiệu năng đảm bảo... Việc lựa chọn được một kiến trúc phù hợp cũng giúp bản thân lập trình viên có cách tổ chức mã nguồn hợp lý, dễ hiểu, giảm công sức tìm kiếm và sửa chữa khi có lỗi phát sinh. Trong đồ án, kiến trúc mà em áp dụng là MVP – một kiến trúc đã và đang được áp dụng trong rất nhiều ứng dụng Android.

MVP (Model View Presenter) là một mô hình kiến trúc hướng giao diện người dùng. Bản chất MVP được dẫn xuất từ mô hình MVC (Model View Controller) truyền thống, tuy nhiên có nhiều điểm cải tiến hơn như được thiết kế để tạo thuận lợi cho

việc kiểm thử đơn vị (unit testing) và tăng tính tách biệt giữa tầng dữ liệu và tầng hiển thị dữ liệu trong mô hình MVC. Kiến trúc MVP cho phép tách tầng trình diễn (Presenter) ra khỏi tầng dữ liệu (Model), vì vậy tương tác với giao diện được tách biệt với cách chúng ta biểu diễn nó trên màn hình (View), hay nói cách khác, tất cả logic khi người dùng tương tác được tách ra và đưa vào tầng trình diễn. Thiết kế lý tưởng nhất là với cùng một logic được áp dụng cho nhiều View khác nhau và hoán đổi được cho nhau. Hình 11 dưới đây mô tả các thành phần cũng như luồng dữ liệu trong MVP.



Hình 11 Các thành phần và luồng dữ liệu trong MVP⁵

Trong kiến trúc MVP: (i) Model là một interface xác định cách mà dữ liệu được hiển thị trong giao diện người dùng. (ii) View là một giao diện người dùng thụ động hiển thị dữ liệu (Model) và tiếp nhận tương tác người dùng và truyền tới để Presenter xử lý tương tác. (iii) Presenter được ví như middle-man. Khi người dùng tương tác với View, Presenter tiếp nhận tương tác người dùng và update Model. Khi Model được update hay có thay đổi, Presenter lấy dữ liệu từ Model, định dạng và đưa tới View để hiển thị. Điểm khác biệt dễ thấy nhất ở đây khi so sánh mô hình MVP với mô hình MVC chính là vị trí cũng như chức năng của tầng Presenter và View so với tầng Controller. Trong mô hình MVP, tầng View là tầng duy nhất tiếp nhận tương tác người dùng thay vì cả 2 tầng View và tầng Controller trong mô hình MVC. Logic xử

⁵ <https://viblo.asia/p/mvp-pattern-for-android/>, lần truy cập cuối 21/05/2018

lý tương tác cũng như logic xử lý dữ liệu hiển thị được tách ra trong tầng Presenter thay vì được gộp chung với tiếp nhận tương tác trong Controller.

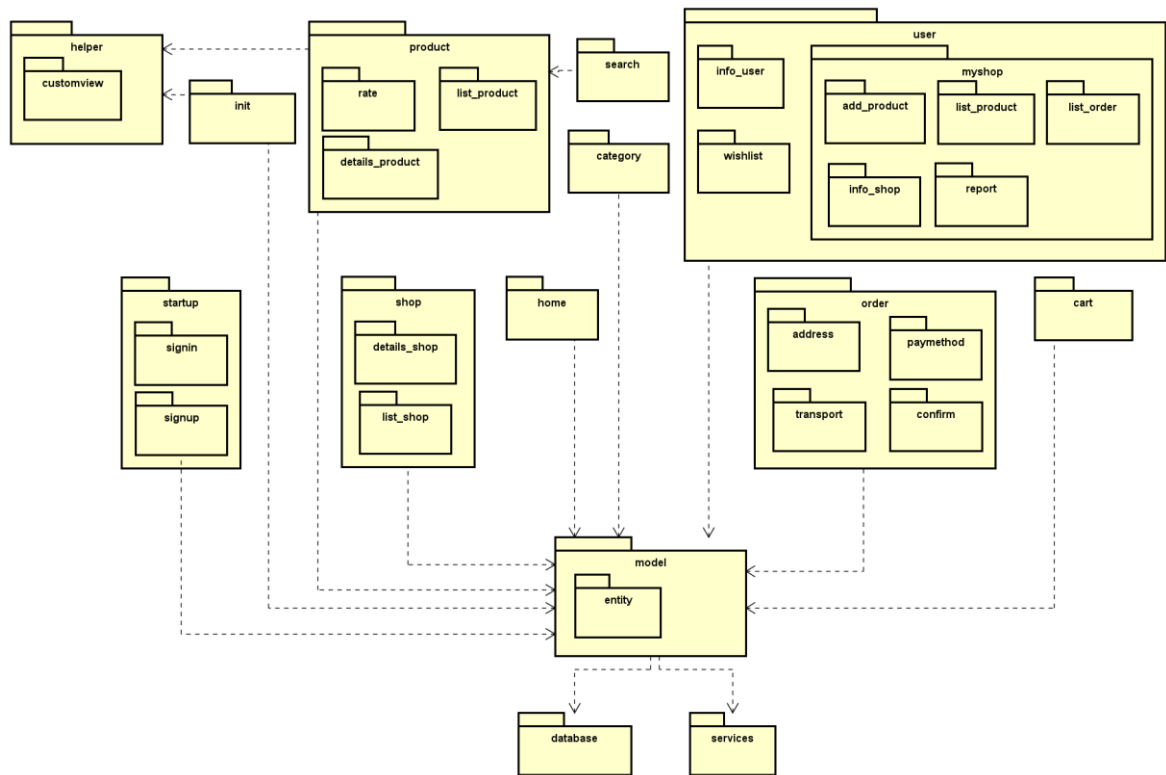
Trong Android, có một vấn đề phát sinh từ thực tế là các cơ chế xử lý tương tác trong Android được kết hợp chặt chẽ giữa giao diện người dùng và xử lý, truy cập dữ liệu. Việc kết hợp này có thể giúp giảm thiểu lượng code trong ứng dụng cũng như gộp các xử lý, logic trong ứng dụng vào cùng một nơi. Nhưng đối với một ứng dụng liên tục phát triển, hay một ứng dụng lớn, việc này khiến cho lượng code trên mỗi logic trở nên rất lớn, các logic xen lẫn, chồng chéo lên nhau, rất khó cho việc đọc hiểu, bảo trì cũng như mở rộng. Phân tầng, phân lớp ứng dụng có thể giảm đi hiệu năng do ứng dụng cần thêm nhiều tài nguyên, nhưng tính khả chuyển của ứng dụng cũng tăng gấp nhiều lần., không những thế còn mở rộng khả năng cho phép kiểm thử từng phần của ứng dụng trở nên dễ dàng hơn. Chính vì thế cân nhắc giữa hiệu năng và tính dễ dàng mở rộng, bảo trì là hết sức quan trọng. Bên cạnh đó, trong Android, kiểm thử là một vấn đề khó khăn vì mối liên kết chặt chẽ giữa giao diện, logic và dữ liệu. Với việc áp dụng kiến trúc MVP, ta có thể giải quyết được hầu hết những vấn đề trên như tách biệt giao diện khỏi dữ liệu, chia ứng dụng ra thành ít nhất ba lớp khác nhau, từ đó có thể kiểm thử đo đạc một cách tối đa các logic của ứng dụng.

Trong đồ án, về cơ bản, MVP được áp dụng mà vẫn giữ nguyên những đặc điểm ban đầu về kiến trúc, tuy nhiên được điều chỉnh lại ở một vài khía cạnh nhằm thích hợp hơn với bài toán. Các thành phần Model, View, Presenter vẫn giữ nguyên vai trò và chức năng như trong kiến trúc đã định nghĩa. Thành phần Model bao gồm các lớp tương tác với dữ liệu. Dữ liệu ở đây có thể là dữ liệu được request từ server hoặc dữ liệu lấy ở database local trên thiết bị. Model thực hiện tất cả các thao tác xử lý liên quan đến việc truy vấn các dữ liệu này hoặc chuyển đổi định dạng dữ liệu cho phù hợp và gửi lại kết quả cho Presenter. Presenter được tách thành 2 thành phần con: Một interface IPresenter và một class PresenterImp có nhiệm vụ thực thi (implement) interface đó. Việc phân tách ra hai thành phần ở Presenter giúp tăng tính khả chuyển và khả năng tái sử dụng. Những logic ở Presenter đôi khi được sử dụng để xử lý nhiều view khác nhau và khi một presenter nào đó muốn sử dụng lại logic ấy, ta chỉ việc

implement từ IPresenter đã tạo trước đó. Lớp PresenterImp cần có một phương thức khởi tạo truyền vào một View (sẽ đề cập tới trong phần sau), đồng thời chứa một instance của thành phần Model. Dựa trên những dữ liệu nhận được từ Model, Presenter thực hiện những thao tác xử lý cần thiết và gọi đến View để hiển thị thông tin mong muốn lên giao diện người dùng. View cũng được tách biệt thành hai thành phần con: Một interface IView chứa các hàm chức năng và một class giao diện đặc trưng của Android (một Activity hoặc một Fragment) thực thi interface đó. Lớp giao diện chứa một instance của PresenterImp, với mỗi thao tác người dùng thực hiện trên ứng dụng đòi hỏi cập nhật lại giao diện, ta có thể sử dụng instance đó gọi đến các hàm xử lý đã định nghĩa từ trước của lớp PresenterImp tương ứng để tương tác với dữ liệu từ Model và cập nhật lại View. Ví dụ cụ thể về kiến trúc MVP được áp dụng trong chức năng hiển thị danh sách sản phẩm mới nhất: Thành phần Model là lớp ProductModel thực hiện việc request dữ liệu từ server về thông tin sản phẩm, thành phần Presenter gồm interface ProductPresenter và lớp ProductPresenterImp implement interface đó, thành View gồm interface ListProductView và một fragment ListProductFragment implement interface. Dựa theo kiến trúc được trình bày ở trên, trình tự luồng sự kiện của chức năng này sẽ diễn ra như sau: Khi người dùng thực hiện thao tác mở fragment, instance của lớp ProductPresenterImp trong class ListProductFragment sẽ gọi đến hàm load dữ liệu đã được định nghĩa trước đó trong lớp ProductPresenterImp. Hàm load dữ liệu này sử dụng một instance của ProductModel để request dữ liệu về danh sách các sản phẩm mới nhất. Dựa trên dữ liệu trả về. nếu dữ liệu rỗng, sử dụng instance của View trong ProductPresenterImp để gọi hàm hiển thị thông báo không có dữ liệu cho người dùng; ngược lại, sử dụng instance của View để gọi hàm hiển thị danh sách sản phẩm. Cách thức hiển thị cụ thể như thế nào sẽ được thực hiện trong ListProductFragment.

4.1.2 Thiết kế tổng quan

Hình 12 dưới đây mô tả thiết kế tổng quan của ứng dụng:



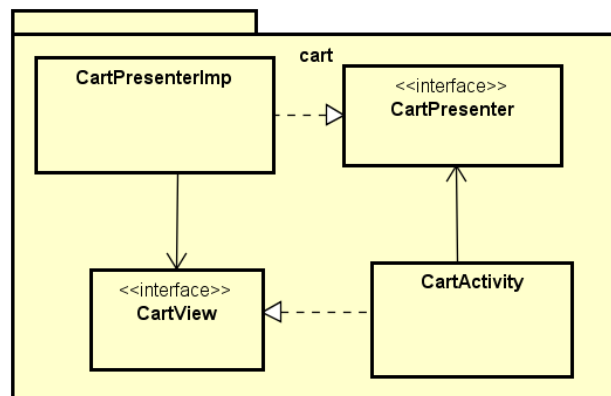
Hình 12 Thiết kế tổng quan ứng dụng

Ứng dụng được thiết kế phân chia thành 3 tầng. Tầng trên cùng chứa các package theo từng chức năng, thực hiện các nghiệp vụ xử lý chính của toàn bộ ứng dụng. Tầng thứ hai gồm các gói và lớp liên quan đến việc xử lý logic và tương tác với dữ liệu. Tầng dưới cùng là tầng cơ sở dữ liệu, nơi quản lý dữ liệu của hệ thống. Thiết kế chi tiết các gói của từng tầng sẽ được mô tả kỹ hơn trong phần tiếp theo.

4.1.3 Thiết kế chi tiết gói

Phần này sẽ trình bày thiết kế chi tiết của một số gói quan trọng trong kiến trúc. Đây là các gói có thiết kế đặc trưng, mang tính tổng quát, có thể làm đại diện cho các gói còn lại, hoặc các gói thực hiện những chức năng quan trọng của sản phẩm. Ba gói mà em lựa chọn để trình bày là: gói Giỏ hàng (cart), gói Chi tiết sản phẩm (details_product) và gói Đặt hàng (order).

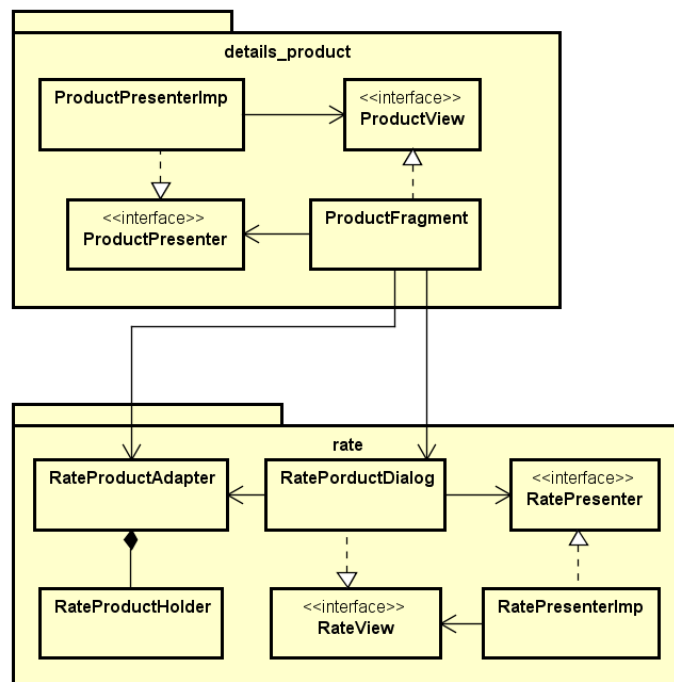
Hình 13 mô tả thiết kế chi tiết gói Giỏ hàng (cart):



Hình 13 Thiết kế gói Giỏ hàng

Gói gồm 4 thành phần chính được phân tách theo mô hình MVP đã được trình bày trong mục 4.1.1 gồm: 1 interface **CartPresenter**, 1 class **CartPresenterImp** thực thi interface đó, class này chứa một instance của interface **CartView**. Cuối cùng là lớp **CartActivity** thực thi interface **CartView** và chứa một instance của **CartPresenter**. Mỗi thao tác của người dùng được thực hiện trên giao diện quản lý bởi **CartActivity** sẽ được **CartPresenter** tiếp nhận và xử lý, dữ liệu được lấy từ tầng Model và được cập nhật hiển thị thông qua một đối tượng **CartView** chứa trong **CartPresenterImp**. Đây là một thiết kế đặc trưng được áp dụng trong rất nhiều gói của đồ án.

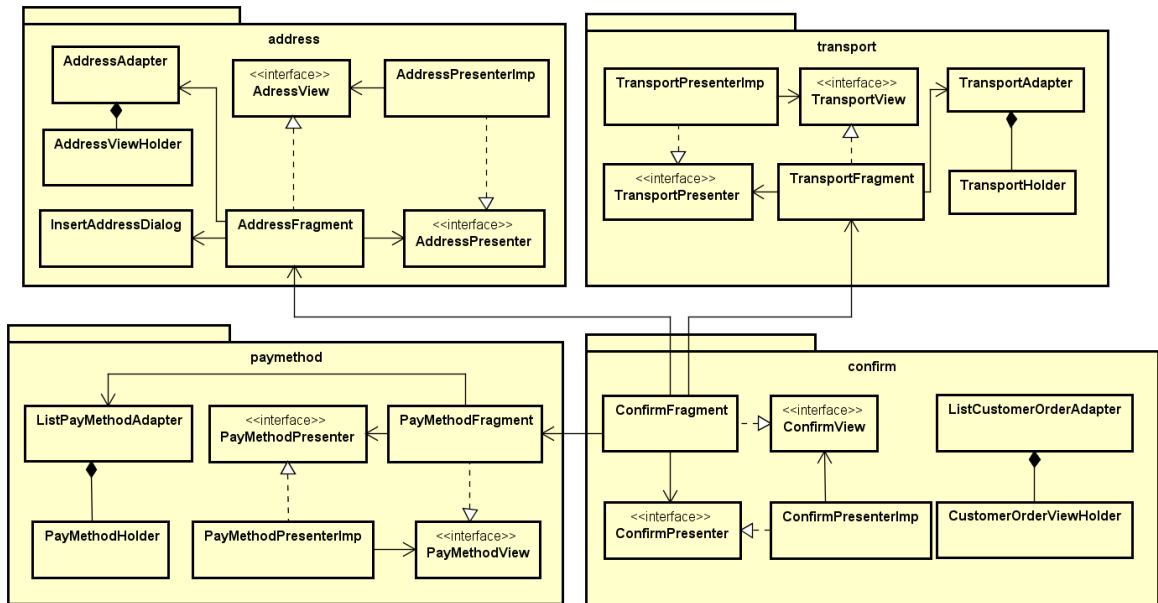
Hình 14 mô tả thiết kế của gói Chi tiết sản phẩm (details_product).



Hình 14 Thiết kế gói chi tiết sản phẩm

Gói này được kết hợp cùng gói Đánh giá (rate). Khi xem thông tin chi tiết của một sản phẩm, người dùng đồng thời có thể để lại đánh giá trong trang chi tiết đó. Gói Đánh giá thực hiện hai nhiệm vụ: (i) Kiểm tra khách hàng đã đánh giá sản phẩm đó chưa, nếu đã tồn tại đánh giá của khách hàng, thông báo cho gói Chi tiết sản phẩm vô hiệu hóa chức năng đánh giá. (ii) Nếu người dùng chưa đánh giá, thực hiện chức năng đánh giá sản phẩm. Trong mỗi gói, thiết kế cũng được thực hiện theo mô hình MVP đã đề cập ở trên. Lớp **ProductFragment** nhận vào các thành phần **RateProductAdapter** và **RateProductDialog** để thực hiện chức năng đánh giá trên giao diện.

Tiếp theo là thiết kế chi tiết của gói Đặt hàng (order) được mô tả lại trong Hình 15.



Hình 15 Thiết kế gói Đặt hàng

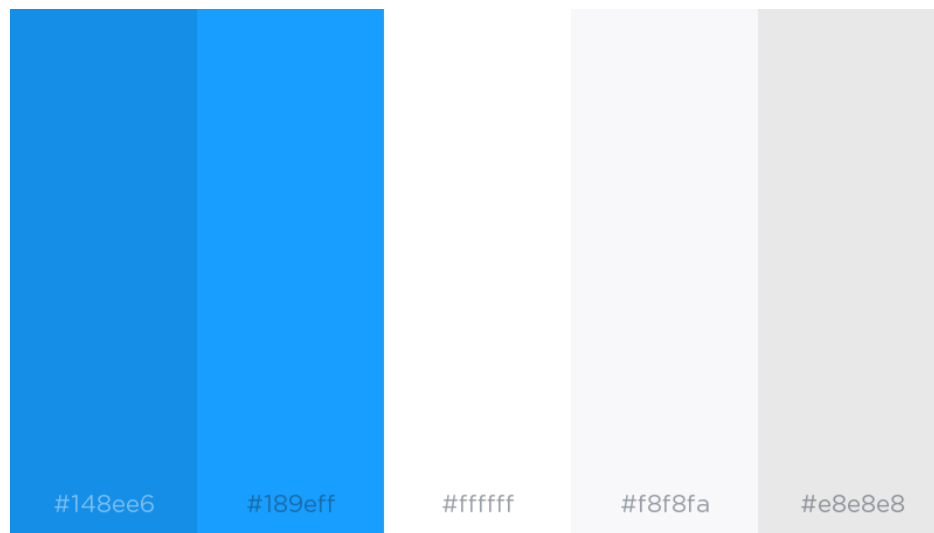
Gói này có sự tham gia của nhiều thành phần tương ứng với từng bước trong quy trình đặt hàng, từ bước xác nhận địa chỉ, chọn hình thức vận chuyển, chọn cách thức thanh toán và xác nhận đặt hàng. Mỗi gói con đều tuân theo mô hình MVP. Ở gói cuối – gói xác nhận (confirm) nhận vào một số tham số mà người dùng đã lựa chọn ở những bước trước để tạo và hiển thị đơn hàng hoàn chỉnh với tổng số tiền cần thanh toán.

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Ứng dụng được thiết kế phù hợp với các thiết bị Android có kích thước màn hình từ 4” (inch) đến 6”. Độ phân giải màn hình hỗ trợ: 480 x 800, 720 x 1280 (HD), 1080 x 1920 (full HD) và 1440 x 2560 (2k). đặc biệt hiển thị tốt trên màn hình 5,5” , độ phân giải full HD (1080 x 1920).

Trong thiết kế ứng dụng nói riêng và thiết kế sản phẩm nói chung, màu sắc đóng một vai trò đặc biệt quan trọng. Không giống với bất kì một phương tiện giao tiếp nào khác, màu sắc có thể ảnh hưởng trực tiếp đến cảm xúc và là cách thức truyền tải thông điệp nhanh chóng. Một ứng dụng về thương mại cần lựa chọn màu sắc tạo được cảm giác an toàn và tin cậy cho người dùng. Đó là lý do em lựa chọn xanh dương làm màu chủ đạo cho sản phẩm. Màu xanh gây cho người ta cảm nhận về sự tin tưởng, sự trông cậy, có trách nhiệm về tài chính và tính bảo đảm⁶. Bên cạnh đó, với đặc thù là một ứng dụng mua bán hàng, số thành phần giao diện là tương đối nhiều và dễ gây rối mắt, bởi vậy màu sắc cần có sự phối hợp hài hòa và nhẹ nhàng. Hình 16 dưới đây là bảng phối màu mà em áp dụng cho sản phẩm của mình.



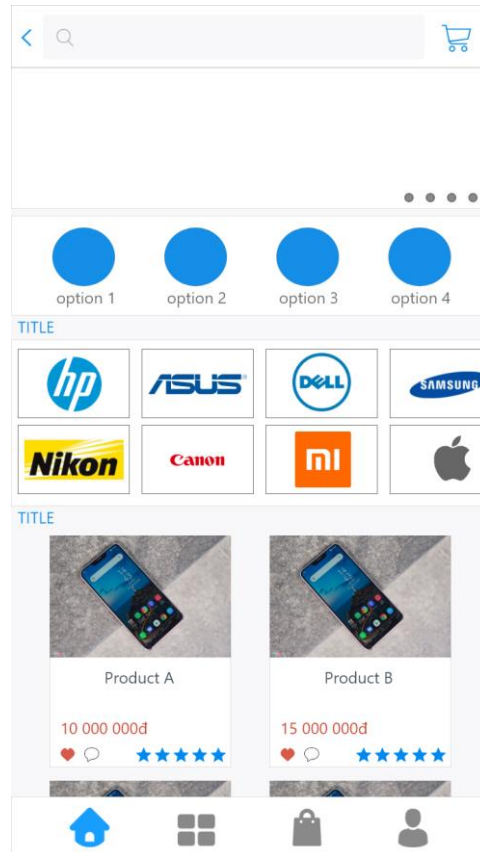
Hình 16 Bảng phối màu của ứng dụng

Ứng dụng tuân theo 3 nguyên tắc thiết kế lớn⁷ của Material Design: (i) Có thể đoán trước (Predictable), (ii) Thích hợp (Consistent) và (iii) Đáp ứng (Responsive). Các phần tử xuất hiện trong một màn hình có bố cục rõ ràng, không chồng chéo lên nhau và dễ dàng để người dùng nhận biết. Giao diện dạng lưới được áp dụng tối đa, đặc biệt là các màn hình hiển thị danh sách sản phẩm. Kích thước padding và margin

⁶ <https://greenacademy.edu.vn/>, lần truy cập cuối 15/5/2018

⁷ <https://material.io/design/>, lần truy cập cuối 20/5/2018

được thống nhất trong toàn ứng dụng. Giao diện có khả năng co giãn và tương thích với nhiều kích thước màn hình khác nhau nhờ sử dụng các kỹ thuật phân chia theo tỷ lệ, đồng thời hạn chế tối đa việc xét cứng các kích thước. Hình 17 dưới đây minh họa thiết kế điển hình của màn hình trang chủ.



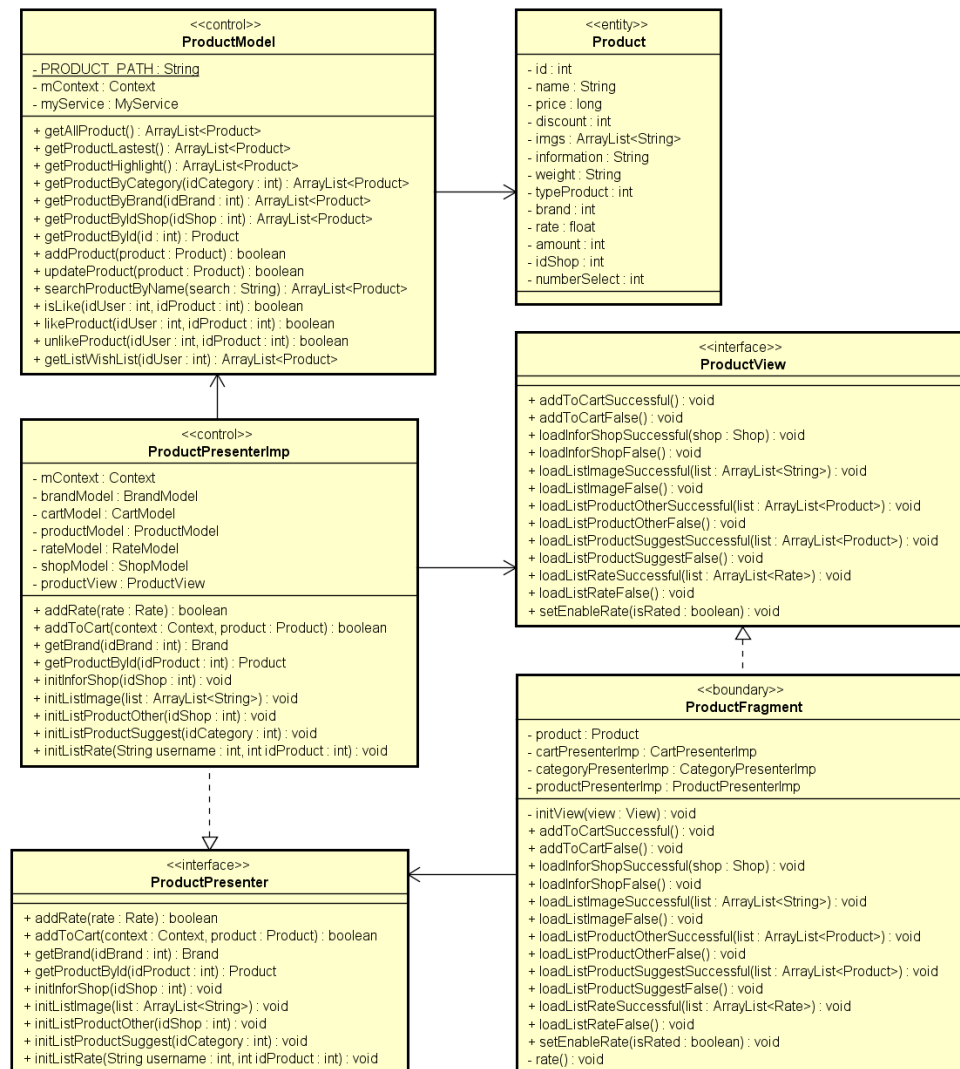
Hình 17 Minh họa thiết kế ứng dụng

Một số chuẩn hóa khác khi thiết kế giao diện: Nút bấm được bo viền, màu nền là xanh lục hoặc đỏ, tùy theo mục đích hiển thị hoặc ý nghĩa hành động (ví dụ với các thao tác xóa dữ liệu, nút bấm có màn đỏ), text hiển thị trong nút màu trắng, được căn ở vị trí chính giữa. Các thông điệp của hệ thống (thông báo thành công, thất bại, lỗi phát sinh...) được hiển thị chính giữa, bên dưới màn hình. Nội dung thông điệp là các đoạn text màu trắng trên nền xám, xuất hiện trong thời gian từ 1 – 3 giây. Dialog của ứng dụng được hiển thị chính giữa màn hình, nền trắng, bo góc. Về điều khiển, ứng dụng dùng thanh công cụ đặt phía dưới cùng của màn hình để điều hướng đến 4 màn hình chính khác nhau: (i) màn hình trang chủ, (ii) màn hình danh mục sản phẩm,

(iii) màn hình market, (iv) màn hình profile. Đây là 4 màn hình cơ sở được khởi tạo khi người dùng bắt đầu mở ứng dụng, từ các màn hình này có thể lần lượt truy xuất đến tất cả các màn hình chức năng còn lại của hệ thống. Từ một màn hình bất kì của ứng dụng, để quay lại màn hình trước đó, người dùng có thể sử dụng nút back trên thiết bị Android hoặc phím mũi tên quay lại nằm ở góc trên cùng bên trái màn hình.

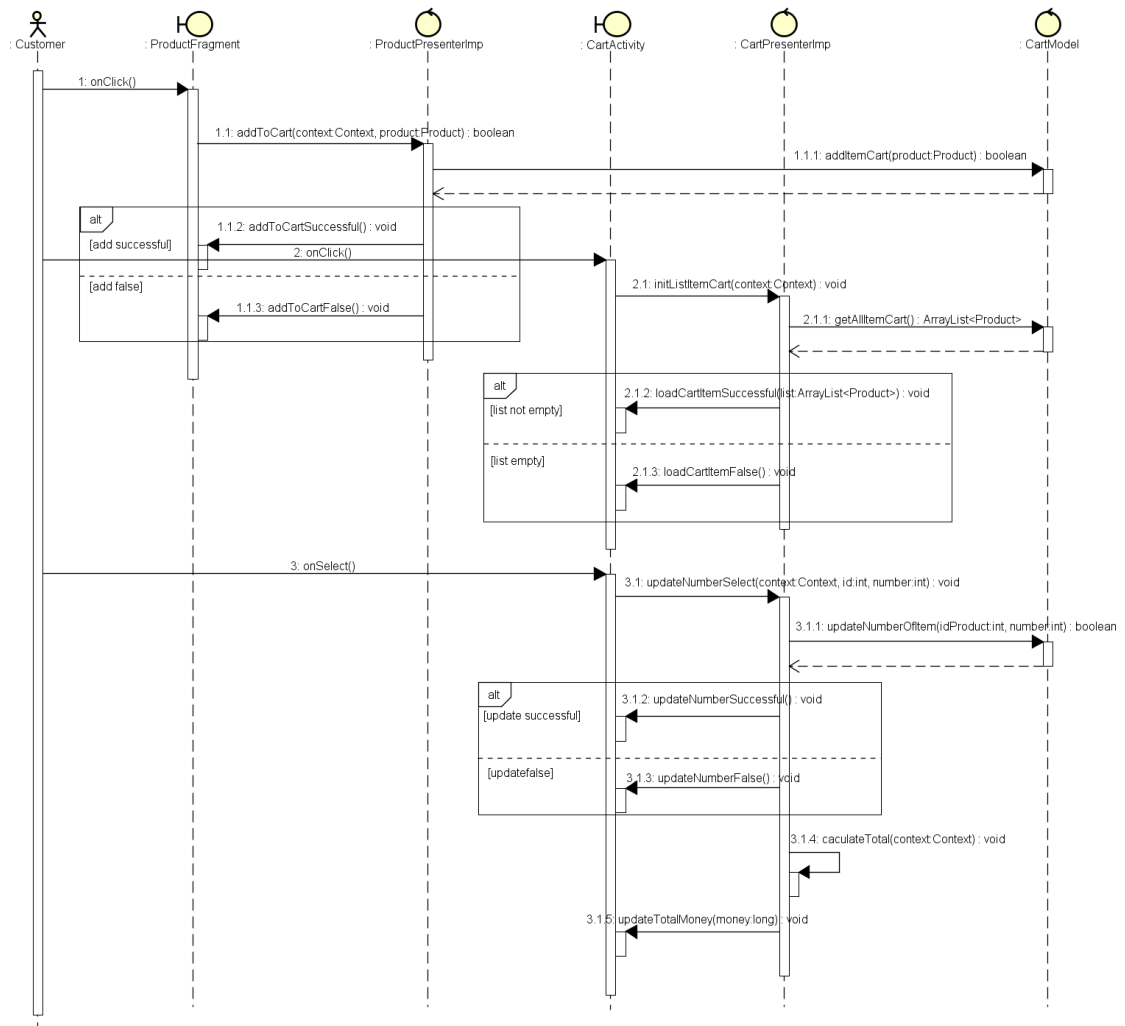
4.2.2 Thiết kế lớp

Hình 18 mô tả thiết kế chi tiết của các lớp liên quan đến đối tượng sản phẩm – một thành phần quan trọng không thể thiếu trong ứng dụng.



Hình 18 Thiết kế lớp chi tiết

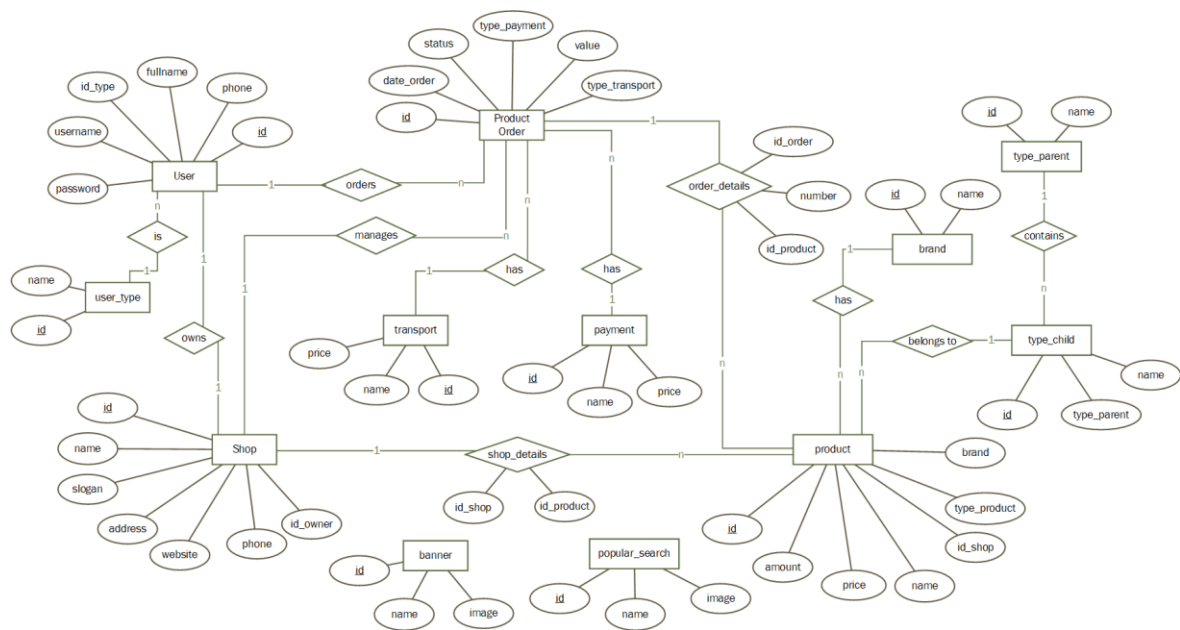
Và để minh họa rõ hơn cho thiết kế lớp, Hình 19 dưới đây là biểu đồ trình tự mô tả luồng truyền thông điệp giữa các đối tượng tham gia cho trong 2 use case: Thêm hàng vào giỏ và cập nhật thông tin giỏ hàng.



Hình 19 Biểu đồ trình tự

4.2.3 Thiết kế cơ sở dữ liệu

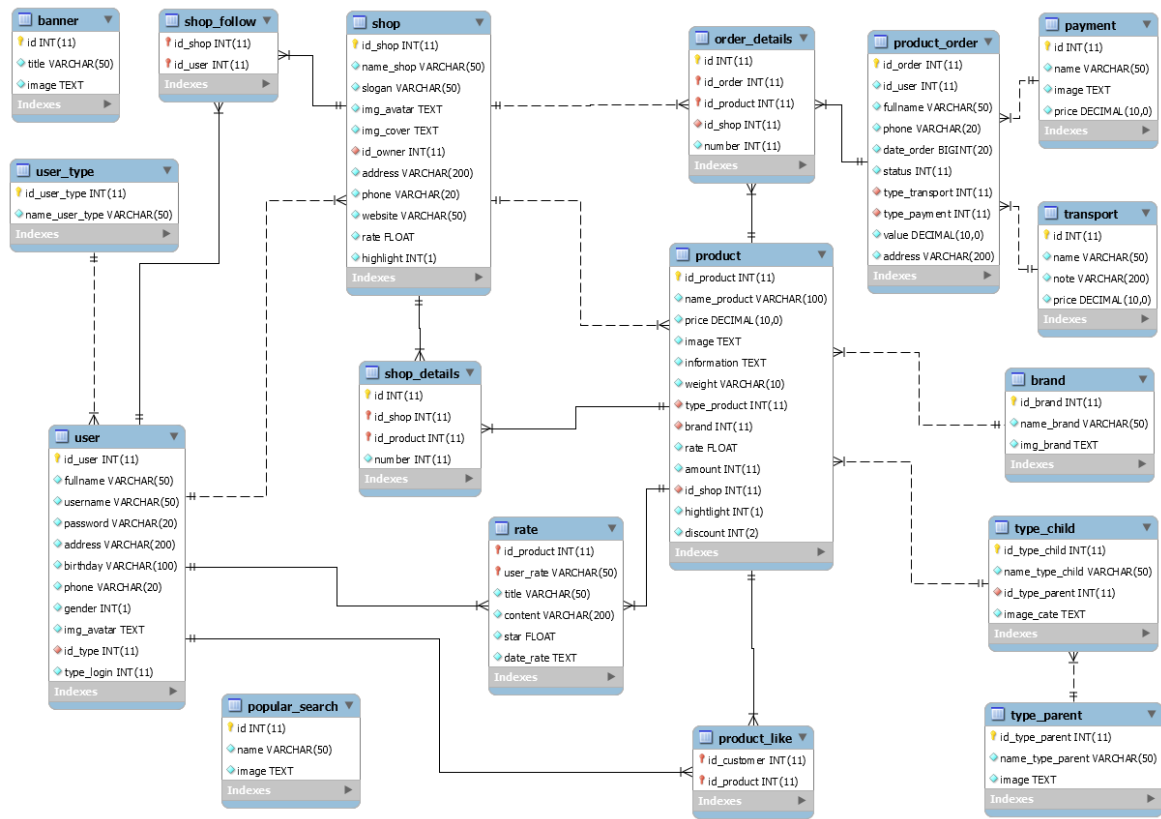
Hình 20 dưới đây là biểu đồ thực thể liên kết, mô tả các thuộc tính chính và mối quan hệ giữa các thực thể quan trọng trong hệ thống.



Hình 20 Biểu đồ thực thể liên kết

Ta có thể giải thích sơ bộ về biểu đồ trên như sau: Mỗi đối tượng thực thể user (người dùng) liên kết với một thực thể user_type, xác định đối tượng user đó có vai trò gì trong hệ thống (người khách mua hàng, người bán hàng hay admin), liên kết ở đây là 1 – n, một user_type có thể có nhiều user khác nhau. Mỗi user có thể sở hữu 1 cửa hàng (shop), liên kết là 1 – 1, đồng thời có thể liên kết tới nhiều đơn hàng (product_order) khác nhau. Mỗi đơn hàng có một hình thức thanh toán hoặc vận chuyển nhất định. Liên kết giữa thực thể đơn hàng với các thực thể vận chuyển (transport) và thanh toán (payment) đều là liên kết 1 – n. Thực thể đơn hàng (product_order) và sản phẩm (product) liên kết với nhau qua một thực thể trung gian là chi tiết đơn hàng (order_details), mỗi đơn hàng có thể bao gồm nhiều sản phẩm khác nhau. Tương tự như vậy, liên kết giữa cửa hàng (shop) và sản phẩm (product) cũng được thực hiện thông qua một thực thể trung gian là chi tiết sản phẩm (shop_details). Thực thể sản phẩm lần lượt liên kết với thực thể thương hiệu (brand) và thực thể loại sản phẩm con (type_child) bằng các liên kết 1 – n, mỗi sản phẩm thuộc một thương hiệu và một danh mục sản phẩm nhất định. Thực thể loại sản phẩm con nằm trong một loại sản phẩm cha (type_parent) duy nhất, liên kết là 1 – n.

Từ biểu đồ thực thể - liên kết trên, cơ sở dữ liệu của sản phẩm được xây dựng trên hệ quản trị cơ sở dữ liệu MySQL. Hình 21 dưới đây là chi tiết các bảng dữ liệu.



Hình 21 Các bảng dữ liệu

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Danh sách các IDE, công cụ, thư viện và ngôn ngữ lập trình được sử dụng trong đồ án được liệt kê trong Bảng 7 dưới đây:

Bảng 7 Danh sách công cụ và thư viện sử dụng

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Android Studio 3.0.1	https://developer.android.com/

IDE lập trình	Sublime text 3, build 3143	https://www.sublimetext.com/
Công cụ tạo và quản lý server	XAMPP 7.1	https://www.apachefriends.org/
Ngôn ngữ lập trình	Java 8 update 131	http://www.oracle.com/
Ngôn ngữ lập trình	PHP 7.1.6	http://php.net/
Thư viện	Volley 1.1.0	https://github.com/google/volley/
Thư viện kiểm thử	JUnit	https://junit.org/
Thư viện kiểm thử	Mockito	http://site.mockito.org/

4.3.2 Kết quả đạt được

Đồ án đã thực hiện được đầy đủ các use case chính đề cập trong mục 2.2. Các use case lớn có nghiệp vụ phức tạp bao gồm: Quản lý giỏ hàng, Đặt hàng, CRUD sản phẩm, UD đơn hàng, Đăng ký mở gian hàng.

Sản phẩm được đóng gói thành các phần: gói server, gói ứng dụng client và gói database. Phía server, gói bao gồm mã nguồn, hình ảnh và các thư viện cần thiết, thực hiện các thao tác xử lý phía server (tiếp nhận request từ client, xử lý và trả về response...). Phía ứng dụng client, gói cũng bao gồm mã nguồn, danh sách các hình ảnh và file cài đặt apk của ứng dụng. Gói database gồm file dữ liệu định dạng sql, chứa toàn bộ dữ liệu được sử dụng trong sản phẩm.

Bảng 8 thống kê chi tiết về các thông tin của ứng dụng:

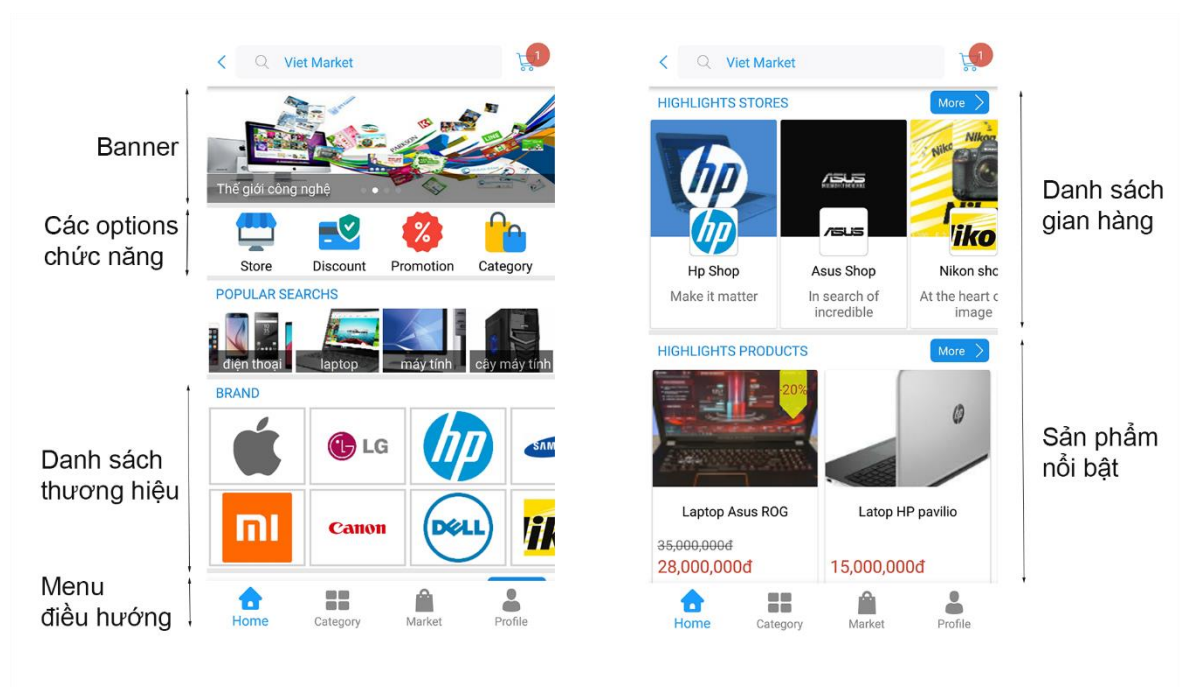
Bảng 8 Thống kê các thông tin ứng dụng

Tiêu chí	Thông tin
Số dòng code	13298
Số lớp	177

Số gói	36
Dung lượng toàn bộ mã nguồn	4,35 MB
Dung lượng gói ứng dụng client	2,83 MB
Dung lượng gói server	1,52 MB
Dung lượng gói database	26 KB

4.3.3 Minh hoạ các chức năng chính

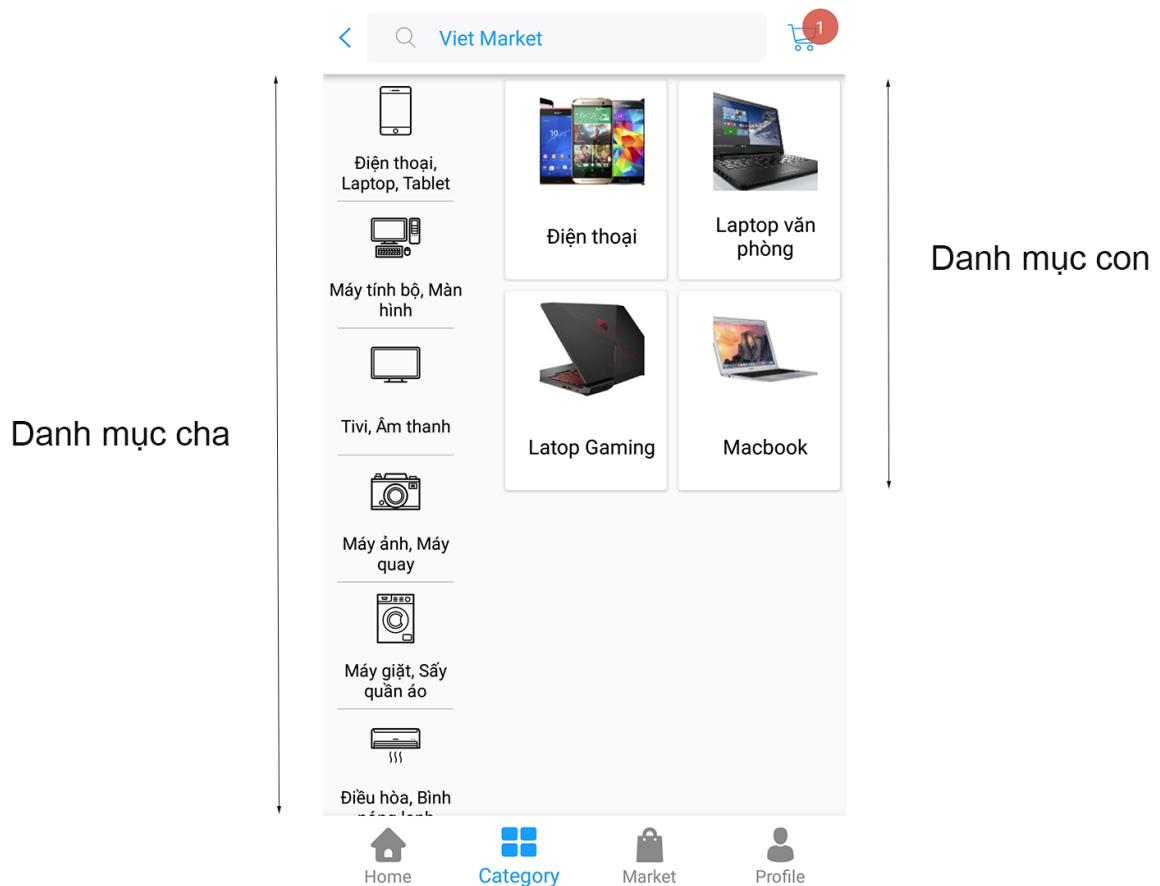
Một số chức năng chính và quan trọng của ứng dụng được thể hiện qua các giao diện màn hình dưới đây.



Hình 22 Giao diện màn hình trang chủ

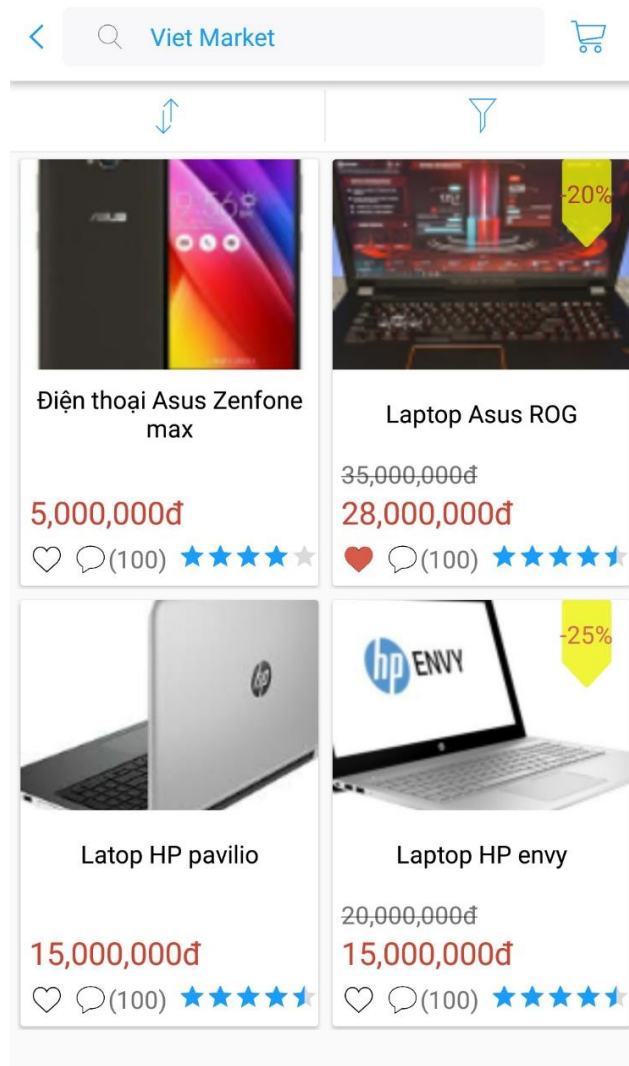
Hình 22 mô tả giao diện màn hình trang chủ, nơi hiển thị đầu tiên mỗi khi người dùng truy cập vào ứng dụng. Màn hình này là nơi tập trung nhiều thông tin nhất trong toàn bộ ứng dụng. Phía trên là vị trí đặt các banner quảng cáo, danh mục những từ khóa tìm kiếm phổ biến, danh sách các thương hiệu. Phần phía dưới đưa ra danh sách các cửa hàng nổi bật. Danh sách các sản phẩm được sắp xếp theo nhiều tiêu chí khác

nhau: sản phẩm mới nhất, sản phẩm nổi bật, sản phẩm gợi ý. Từ màn hình trang chủ, người dùng có thể truy xuất sang các màn hình chức năng quan trọng khác như: màn hình tìm kiếm, màn hình danh mục sản phẩm, danh sách cửa hàng hoặc danh sách các sản phẩm.



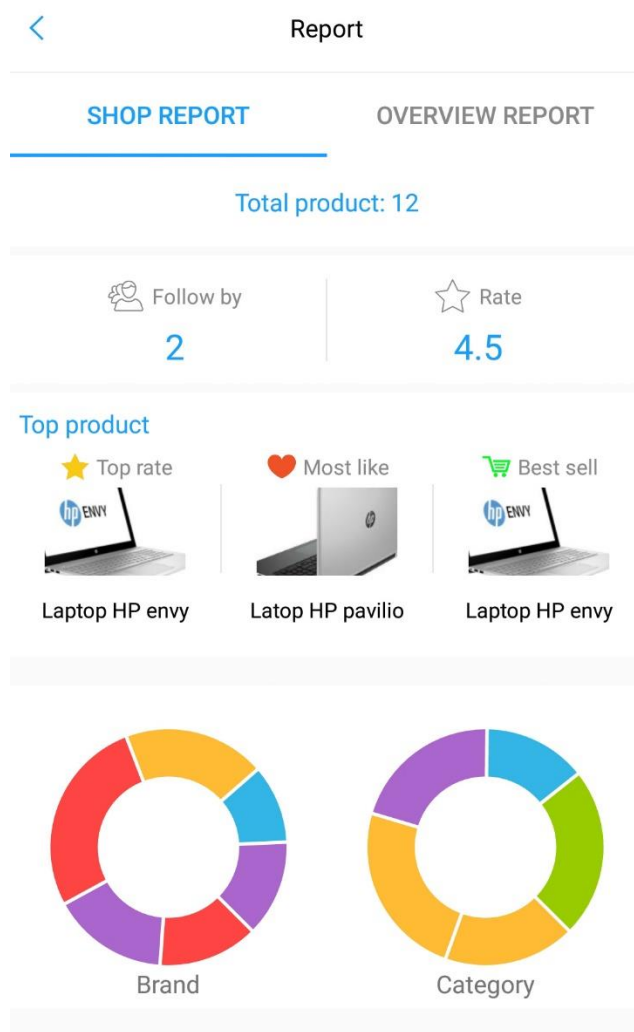
Hình 23 Giao diện màn hình danh mục sản phẩm

Hình 23 là giao diện danh mục các sản phẩm. Theo thiết kế, một danh mục sản phẩm lớn sẽ gồm nhiều danh mục sản phẩm con, bởi vậy màn hình cũng được xây dựng sao cho thể hiện đồng thời hai thông tin này một cách trực quan. Bên cạnh trái là danh sách các danh mục lớn, khi chọn vào một danh mục lớn bất kỳ, danh sách các danh mục con tương ứng với danh mục đó sẽ hiển thị bên phải. Từ danh mục con có thể truy xuất tới danh sách các sản phẩm thuộc danh mục con đó. Các phần tử thuộc từng loại danh sách được thiết kế theo chuẩn khác nhau giúp người dùng dễ phân biệt.



Hình 24 Giao diện màn hình danh sách sản phẩm

Hình 24 là giao diện màn hình danh sách các sản phẩm. Danh sách hiển thị dưới dạng grid, kích thước 2 x n, phía trên là một thanh công cụ tùy biến cho phép thực hiện một số chức năng tiện ích như sắp xếp sản phẩm hoặc lọc sản phẩm. Các tiêu chí sắp xếp sản phẩm bao gồm: Sắp xếp sản phẩm theo thời gian cũ/mới nhất và sắp xếp theo mức giá (từ đắt nhất – rẻ nhất hoặc từ rẻ nhất – đắt nhất). Các tiêu chí lọc sản phẩm bao gồm: lọc theo khoảng giá, lọc theo thương hiệu, lọc theo danh mục sản phẩm.



Hình 25 Giao diện màn hình thống kê

Hình 25 mô tả giao diện màn hình thống kê của người bán hàng. Từ màn hình này, người bán hàng có thể biết được các thông tin về gian hàng mà mình đang quản lý như: Tổng số sản phẩm trong kho, sản phẩm được yêu thích nhất, sản phẩm được đánh giá cao nhất hay được bán nhiều nhất. Bên cạnh đó, màn hình còn được tích hợp một số biểu đồ thống kê về tỉ lệ các thương hiệu và danh mục hàng mà gian hàng đang kinh doanh hay biểu đồ thống kê doanh số theo các tháng giúp thông tin được hiển thị trực quan và dễ hiểu nắm bắt hơn.

4.4 Kiểm thử

4.4.1 Kiểm thử hộp đen

Bảng 9 và Bảng 10 dưới đây mô tả các trường hợp kiểm thử hộp đen cho 2 chức năng: Thêm mới sản phẩm và đăng ký mở gian hàng.

Bảng 9 Kiểm thử hộp đen chức năng “Thêm mới sản phẩm”

Test case ID	Mô tả	Các bước thực hiện	Kết quả mong muốn	Trạng thái
TC001	Nhập đúng và đầy đủ các trường thông tin	(i) Nhập tên: “Điện thoại A”. (ii) Chọn “Thêm ảnh” → Chọn “Thư viện ảnh” → Chọn ảnh bất kỳ trong thiết bị → Chọn “Tải ảnh lên”. (iii) Chọn thương hiệu: “Apple”. (iv) Chọn danh mục “Điện thoại”. (v) Nhập cân nặng: 120 → Chọn đơn vị “gam”. (vi) Nhập mô tả: “Điện thoại của hãng Apple”. (vii) Nhập số lượng: 10. (viii) Nhập giá: 5000 000. (ix) Chọn “Thêm”.	(ii) Hiện thị thông báo “Ảnh được tải lên thành công”. (ix) Hiện thị thông báo “Sản phẩm được thêm thành công”.	Passed
TC002	Không chọn hình ảnh	(i) Nhập tên: “Điện thoại B”. (ii) Chọn thương hiệu: “Apple”. (iii) Chọn danh mục “Điện thoại”.		Passed

(iv) Nhập cân nặng: 120 ➔
Chọn đơn vị “gam”.

(v) Nhập mô tả: “Điện thoại
của hãng Apple”.

(vi) Nhập số lượng: 10.

(vii) Nhập giá: 5000 000.

(viii) Chọn “Thêm”.

TC003	Nhập số lượng hàng bằng 0	(i) Nhập tên: “Điện thoại B”. (ii) Chọn “Thêm ảnh” ➔ Chọn “Thư viện ảnh” ➔ Chọn ảnh bất kỳ trong thiết bị ➔ Chọn “Tải ảnh lên”. (iii) Chọn thương hiệu: “Apple”. (iv) Chọn danh mục “Điện thoại”. (v) Nhập cân nặng: 120 ➔ Chọn đơn vị “gam”. (vi) Nhập mô tả: “Điện thoại của hãng Apple”. (vii) Nhập số lượng: 0. (viii) Nhập giá: 5000 000. (ix) Chọn “Thêm”.	(ii) Hiện thị thông báo “Ảnh được tải lên thành công”. (ix) Hiện thị thông báo “Số lượng sản phẩm phải lớn hơn 0”.	Passed
-------	---------------------------------	---	---	--------

Bảng 10 Kiểm thử hộp đen chức năng “Đăng ký mở gian hàng”

Test case ID	Mô tả	Các bước thực hiện	Kết quả mong muốn	Trạng thái
TC004	Nhập đúng và đầy đủ	(i) Nhập tên: “Cửa hàng A”.		Passed

các trường
thông tin

(ii) Nhập slogan: “Chất lượng hàng đầu”.

(iii) Chọn “Ảnh đại diện” →
Chọn “Thư viện ảnh” →
Chọn ảnh bất kỳ trong thiết bị.

(iv) Chọn “Ảnh bìa” → Chọn
“Thư viện ảnh” → Chọn ảnh
bất kỳ trong thiết bị.

(v) Chọn “Tải ảnh lên”.

(vi) Nhập số điện thoại:
012345678.

(vii) Nhập địa chỉ: “Hà Nội”.

(viii) Nhập địa chỉ website:
“www.abc.com”.

(ix) Chọn “Đăng ký”.

(v) Hiện thị
thông báo “Ảnh
được tải lên
thành công”.

(ix) Hiện thị
thông báo:
“Đăng ký thành
công”.

TC005	Không chọn hình ảnh	(i) Nhập tên: “Cửa hàng B”. (ii) Nhập slogan: “Chất lượng hàng đầu”. (vi) Nhập số điện thoại: 012345678. (vii) Nhập địa chỉ: “Hà Nội”. (viii) Nhập địa chỉ website: “www.abc.com”. (ix) Chọn “Đăng ký”.	Passed
-------	---------------------	--	--------

(ix) Hiện thị
thông báo:
“Chưa nhập đủ
thông tin”.

4.4.2 Kiểm thử hộp trắng

Kiểm thử hộp trắng được thực hiện với hai chức năng trên màn hình trang chủ: Hiện thị danh sách thương hiệu và hiện thị danh sách cửa hàng. Để thực hiện kiểm thử hộp trắng, em sử dụng các thư viện JUnit và Mockito. Hai thư viện này giúp đơn giản hóa việc thực hiện các thao tác kiểm thử đơn vị nội bộ (local unit test), ta có thể tùy ý tạo

dữ liệu mẫu và thử nghiệm trực tiếp bằng cách viết các hàm test. Bảng 11 dưới đây thống kê chi tiết về các trường hợp kiểm thử hộp trắng đã thực hiện.

Bảng 11 Thống kê kiểm thử hộp trắng

Chức năng	Lớp kiểm thử	Thời gian chạy	Bộ dữ liệu kiểm thử	Kết quả
Hiển thị danh sách thương hiệu	ListBrandTest.java	13ms	Danh sách 10 thương hiệu được khởi tạo local	Passed
Hiển thị danh sách cửa hàng	ListShopTest.java	34ms	Danh sách 10 cửa hàng được khởi tạo local	Passed

Tổng kết lại, quá trình kiểm thử được thực hiện với tổng số 7 trường hợp kiểm thử, trong đó, 5 trường hợp kiểm thử hộp đen và 2 trường hợp kiểm thử hộp trắng. Các trường hợp kiểm thử đều trả về kết quả đạt.

4.5 Triển khai

Sản phẩm được chia thành hai phần: Phần server và phần ứng dụng client. Phần ứng dụng client được triển khai trên 3 thiết bị Android khác nhau, chi tiết có thể xem trong Bảng 12 dưới đây.

Bảng 12 Thống kê các thiết bị triển khai

Tên thiết bị	Phiên bản Android	Dung lượng RAM	Độ phân giải màn hình	Loại máy
Nexus S	5.1	343 MB	480 x 800	Máy ảo
Motorola Moto X	4.4.4	2GB	720 x 1280	Máy ảo

Asus Zenfone 3 Max	7.1.1	3GB	1080 x 1920	Máy thật
--------------------------	-------	-----	-------------	----------

Phần server được triển khai trên localhost với máy chủ Apache được tích hợp sẵn trong XAMPP.

Như vậy, Chương 4 trên đây đã trình bày toàn bộ về thiết kế các thành phần trong ứng dụng và quá trình xây dựng sản phẩm, những công cụ sử dụng và kết quả đạt được. Nội dung của Chương 5 tới đây sẽ là những giải pháp và đóng góp nổi bật được thể hiện trong đề án.

Chương 5 Các giải pháp và đóng góp nổi bật

Nếu như Chương 4 cho ta cái nhìn toàn diện về thiết kế và quá trình phát triển ứng dụng thì Chương 5 này sẽ giúp hiểu sâu hơn về những giải pháp và đóng góp nổi bật trong thời gian thực hiện đồ án. Những giải pháp và đóng góp đó được đúc kết từ quá trình tìm hiểu, nghiên cứu nhất định và là tiền đề quan trọng để em có thể hoàn thiện tốt được sản phẩm.

5.1 Áp dụng quy trình nghiệp vụ phức tạp vào giải quyết bài toán

5.1.1 Đặt vấn đề

Với bất kỳ ứng dụng hay hệ thống nào, khi bắt tay vào quá trình xây dựng, quy trình nghiệp vụ cũng là thứ đầu tiên cần làm rõ và được đặt lên hàng đầu. Một quy trình nghiệp vụ nhập nhằng, không rõ ràng và có nhiều lỗ hổng, bất hợp lý sẽ dẫn đến thiết kế tồi và kết quả sau cùng cho ra một sản phẩm đáp ứng không đầy đủ yêu cầu từ người dùng hoặc thực hiện sai những yêu cầu đó. Với đặc thù một sản phẩm liên quan đến thương mại điện tử, quy trình nghiệp vụ lại càng trở nên quan trọng hơn bởi nó trực tiếp ảnh hưởng đến lợi ích của rất nhiều đối tượng tham gia vào hệ thống. Tuy nhiên, quy trình nghiệp vụ của các sản phẩm thương mại điện tử thường không dễ dàng nắm bắt, đặc biệt là với những người chưa có nhiều kinh nghiệm. Không chỉ gồm các hoạt động như tìm kiếm hay đặt hàng, một sản thương mại điện tử còn bao

hàm các nghiệp vụ khác như vận chuyển, thanh toán, quản lý sản phẩm, phân tích, thống kê doanh số... đòi hỏi người phát triển muốn có được ứng dụng tốt cần có thời gian nghiên cứu và tìm hiểu nhất định, cả trong tài liệu cũng như trải nghiệm thực tế từ những hệ thống có sẵn. Khi bắt đầu vào xây dựng đồ án, em cũng đã gặp không ít khó khăn khi tiếp cận và giải quyết những vấn đề đòi hỏi am hiểu sâu về nghiệp vụ.

5.1.2 Giải pháp

Để khắc phục những vấn đề trên, em đã dành thời gian đầu của quá trình làm đồ án tiến hành tham khảo tài liệu về quy chế hoạt động của một số sàn thương mại điện tử đang hoạt động ở Việt Nam hiện nay được công khai trên Internet như: tuvanmuaxe.vn, vietnamworks.com, hoczita.com... Những tài liệu này về cơ bản cung cấp cho em cái nhìn tổng quan về cơ chế trao đổi cũng như vai trò cụ thể của các đối tượng tham gia. Nếu xét về vai trò người dùng là khách mua hàng, các nghiệp vụ liên quan đến đối tượng này như mua hàng và thanh toán là tương đối rõ ràng bởi ta hoàn toàn có thể tự mình trải nghiệm thông qua các ứng dụng thương mại điện tử đã có. Tuy nhiên, ứng dụng mà em hướng đến còn có vai trò của người bán hàng và quản trị viên, những vị trí này không dễ tiếp cận trong thực tế. Để xây dựng nghiệp vụ cho những đối tượng đó, em đã xuất phát từ vai trò người khách hàng, đề ra những quy trình cần có để kết nối hoạt động mua hàng với hệ thống, từ đó vạch ra chức năng và vai trò mà người quản lý hay nhân viên bán hàng cần đảm nhiệm. Chẳng hạn như với một chức năng đặt hàng được thực hiện bởi người mua sẽ kéo theo một loạt các thông tin khác cần quản lý như: đơn hàng sẽ được tạo như thế nào? Gồm những mặt hàng gì? Thông tin cần có trong đơn hàng là gì? Lấy ở đâu? Theo dõi và quản lý đơn hàng bằng cách nào?... Trả lời được tất cả các câu hỏi đó sẽ trả về nghiệp vụ quản lý và theo dõi đơn hàng của người bán.

5.2 Hoàn thiện kiến trúc hệ thống theo mô hình MVP

5.2.1 Đặt vấn đề

Nếu như quy trình nghiệp vụ đã nói ở mục 5.1 là khởi đầu cho việc xây dựng một hệ thống thì khâu thiết kế chính là bước áp dụng những kiến thức đã tổng hợp được để tạo nên một khung kiến trúc cho toàn bộ quá trình phát triển sau này. Hệ thống không được thiết kế một cách cẩn thận và phù hợp sẽ dẫn đến những khó khăn trong quá trình xây dựng sản phẩm, đặc biệt là ở giai đoạn sau, khi số lượng mã nguồn tăng lên nhanh chóng, hệ thống sẽ thiếu đi sự linh hoạt và khả năng mở rộng, các lỗi phát sinh sẽ tốn nhiều thời gian và công sức hơn để sửa chữa, việc bổ sung thêm chức năng mới cũng gặp nhiều khó khăn. Bởi vậy, có thể nói thiết kế quyết định tới một nửa thành công của dự án. Một thiết kế cụ thể thường được xây dựng dựa trên mô hình đã có sẵn từ trước. Những mô hình này có thể được tạo nên bởi các cá nhân hoặc tổ chức có uy tín, trải qua quá trình thay đổi và thử nghiệm lâu dài để nhận được sự chấp nhận chung từ cộng đồng phát triển rồi mới được chính thức công bố và đưa vào áp dụng trong thực tế. Tuy nhiên, bản chất của các mô hình thường chỉ dừng lại ở mức khái quát, trừu tượng, mang tính tư tưởng và thường không có chuẩn chung. Tùy vào từng trường hợp cụ thể sẽ có cách áp dụng và xử lý khác nhau. Bởi vậy, việc hiểu rõ một mô hình là rất khó, và vận dụng được mô hình ấy để giải quyết vấn đề cũng không hề đơn giản. Đặt trong bài toán xây dựng ứng dụng sàn thương mại điện tử trên thiết bị Android, yêu cầu đặt ra là cần sử dụng một mô hình phù hợp sao cho thuận lợi nhất với quy trình phát triển ứng dụng di động.

5.2.2 Giải pháp

Từ những vấn đề đã đặt ra ở trên và qua quá trình tìm hiểu các thiết kế phổ biến, đặc biệt là các thiết kế chuyên biệt cho ứng dụng Android, em đã quyết định lựa chọn mô hình MVP – một cải tiến từ mô hình MVC làm hình mẫu cho việc thiết kế kiến trúc

của hệ thống. Chi tiết về cấu trúc, luồng hoạt động cũng như lợi ích của mô hình đã được trình bày ở mục 4.1.1. Trên thực tế, từng chức năng liên quan đến giao diện người dùng trong sản phẩm của đồ án đều được xây dựng theo mô hình MVP. Mỗi chức năng bao gồm sự tham gia của cả ba thành phần Model, View và Presenter trong kiến trúc. Bên cạnh lợi ích nổi bật trong phân tách hai tầng View - Model cũng như thuận lợi cho quá trình viết và thực thi các unit test như đã đề cập từ trước, việc áp dụng MVP còn giúp cho cấu trúc của hệ thống trở nên linh hoạt hơn rất nhiều. Nếu như theo cách truyền thống, các logic xử lý sẽ được thực hiện trực tiếp trên các lớp quản lý View (Activity hoặc Fragment), nếu View đó bao gồm nhiều thành phần yêu cầu phải xử lý phức tạp thì một điều không thể tránh khỏi là lượng code được viết ra sẽ rất nhiều và rắc rối. Tuy nhiên, khi áp dụng mô hình MVP, lượng logic xử lý ở các lớp View này sẽ trở nên cực kỳ nhẹ nhàng. Chúng sẽ chỉ đơn giản là tiếp nhận các sự kiện từ người dùng sau đó gửi dữ liệu cho Presenter và nhận lại phản hồi thông qua các interface để hiển thị thông tin cho người dùng. Thậm chí, với việc áp dụng MVP, đặc biệt là sự xuất hiện của thành phần Presenter, ta hoàn toàn có thể đẩy một phần các xử lý từ thành phần Model sang cho Presenter, trong trường hợp Model quá cồng kềnh. Sự trao đổi giữa các phần tử trong một mô đun chức năng nhờ thế mà trở nên linh hoạt hơn rất nhiều. Các xử lý cũng được phân chia rành mạch: xử lý dữ liệu, xử lý view, xử lý business logic.

5.3 Mô đun hóa các chức năng

5.3.1 Đặt vấn đề

Xét một cách tổng quan trong quá trình phát triển phần mềm, mô đun hóa luôn là một yêu cầu quan trọng. Một hệ thống được mô đun hóa tốt, tức là các chức năng được phân tách tương đối độc lập theo từng mô đun, và một mô đun có thể được phát triển riêng rẽ cũng như có khả năng tái sử dụng ở các vị trí khác nhau trong hệ thống hoặc ở một hệ thống khác. Mô đun hóa tốt giúp quá trình phát triển sản phẩm và ứng dụng trở nên dễ dàng và linh hoạt. Một nhóm các lập trình viên có thể tập trung phát triển một mô đun A trong khi một nhóm khác cải tiến mô đun B mà không cần phụ thuộc

hoặc chờ đợi nhau. Kết quả sau cùng có thể được ghép nối lại dễ dàng thông qua các giao diện được định nghĩa từ trước. Mô đun hóa còn giúp quá trình sửa chữa những chức năng sẵn có hay thêm mới một chức năng dễ dàng hơn do không bị phụ thuộc vào các thành phần khác của hệ thống. Tuy nhiên, việc mô đun hóa thế nào để vừa đảm bảo các bộ phận thành phần trong cùng một mô đun phải có một quan hệ logic chặt chẽ (high cohesion), vừa đảm bảo các mô đun ghép nối với nhau một cách lỏng lẻo (low coupling) không phải là điều dễ dàng. Cohesion hay sự kết dính của một mô đun là độ đo về tính khớp lại với nhau của các phần trong mô đun đó. Nếu một mô đun chỉ thực hiện một chức năng logic hoặc là một thực thể logic, tức là tất cả các bộ phận của mô đun đó đều tham gia vào việc thực hiện một công việc thì độ kết dính là cao. Nếu một hoặc nhiều bộ phận không tham gia trực tiếp vào việc chức năng logic đó thì mức độ kết dính của nó là thấp. Thiết kế là tốt khi độ kết dính cao. Khi đó sẽ dễ dàng hiểu được từng mô đun và việc sửa chữa một mô đun sẽ không (ít) ảnh hưởng tới các mô đun khác. Constantine và Yourdon định ra 7 mức kết dính theo thứ tự tăng dần sau đây⁸: (i) Kết dính gom góp: các công việc không liên quan với nhau, song lại bị bó vào một mô đun. (ii) Kết dính logic: các thành phần cùng thực hiện các chức năng tương tự về logic chẳng hạn như vào/ra, xử lý lỗi,... được đặt vào cùng một mô đun. (iii) Kết dính thời điểm: tất cả các thành phần cùng hoạt hóa một lúc, chẳng hạn như các thao tác khởi tạo được bó lại với nhau. (iv) Kết dính thủ tục: các phần tử trong mô đun được ghép lại trong một dãy điều khiển. (v) Kết dính truyền thông: tất cả các phần tử của mô đun cùng thao tác trên một dữ liệu vào và đưa ra cùng một dữ liệu ra. (vi) Kết dính tuần tự: trong một mô đun, đầu ra của phần tử này là đầu vào của phần tử khác. (vii) Kết dính chức năng: Mỗi phần của mô đun đều là cần thiết để thi hành cùng một chức năng nào đó. Các lớp kết dính này không được định nghĩa chặt chẽ và cũng không phải luôn luôn xác định được. Một đối tượng kết dính nếu nó thể hiện như một thực thể đơn: tất cả các phép toán trên thực thể đó đều nằm trong thực thể đó. Từ đó phát sinh thêm một mức mới: (viii) Kết dính đối tượng: mỗi phép toán đều liên quan đến thay đổi, kiểm tra và sử dụng thuộc tính của một đối tượng, là

⁸ <https://voer.edu.vn/>, lần truy cập cuối 20/5/2018

cơ sở cung cấp các dịch vụ của đối tượng. Coupling hay ghép nối là độ đo sự nối ghép với nhau giữa các đơn vị (mô đun) của hệ thống. Hệ thống có nối ghép cao thì các mô đun phụ thuộc lẫn nhau lớn. Hệ thống nối ghép lỏng lẻo thì các mô đun là độc lập hoặc là tương đối độc lập với nhau và chúng ta sẽ dễ bảo trì nó. Các mô đun được ghép nối chặt chẽ nếu chúng dùng các biến chung và nếu chúng trao đổi các thông tin điều khiển (ghép nối chung nhau và ghép nối điều khiển). Ghép nối lỏng lẻo đạt được khi bảo đảm rằng các thông tin cục bộ được che dấu trong các mô đun và các mô đun trao đổi thông tin thông qua danh sách tham số (giao diện) xác định. Có thể chia ghép nối thành các mức từ chặt chẽ đến lỏng lẻo như sau: (i) Ghép nối nội dung: hai hay nhiều mô đun dùng lẫn dữ liệu của nhau, đây là mức xấu nhất, thường xảy ra đối với các ngôn ngữ mức thấp dùng các dữ liệu toàn cục hay lạm dụng lệnh GOTO. (ii) Ghép nối chung: một số mô đun dùng các biến chung, nếu xảy ra lỗi thao tác dữ liệu, sẽ khó xác định được lỗi đó do mô đun nào gây ra. (iii) Ghép nối điều khiển: một mô đun truyền các thông tin điều khiển để điều khiển hoạt động của một mô đun khác. (iv) Ghép nối dư thừa: mô đun nhận thông tin thừa không liên quan trực tiếp đến chức năng của nó, điều này sẽ làm giảm khả năng thích nghi của mô đun đó. (v) Ghép nối dữ liệu: Các mô đun trao đổi thông tin thông qua tham số và giá trị trả lại. (vi) Ghép nối không có trao đổi thông tin: mô đun thực hiện một chức năng độc lập và hoàn toàn không nhận tham số và không có giá trị trả lại.

Đặt vào bài toán xây dựng sản thương mại điện tử, việc mô đun hóa là đặc biệt quan trọng bởi lượng chức năng cần hoàn thiện là tương đối nhiều và cho nhiều đối tượng người dùng khác nhau. Ví dụ như các chức năng của người mua hàng cần được tách biệt với chức năng của người bán hàng. Xét riêng trong đối tượng người bán hàng thì nhóm các chức năng liên quan đến quản lý sản phẩm cũng cần tách biệt với nhóm chức năng quản lý đơn hàng.

5.3.2 Giải pháp

Có nhiều cách để thực hiện mô đun hóa một hệ thống sau khi đã có bản thiết kế tổng quan. Tuy nhiên, như đã nói ở trên, mô đun hóa không được tùy tiện mà cần đảm bảo nguyên tắc: high cohesion, low coupling. Trong hệ thống mà em xây dựng, việc mô

đơn hóa đã được thực hiện tương đối thành công. Đầu tiên, về sự kết dính giữa các thành phần trong một mô đun, các thành phần này được kết dính với nhau đạt tới mức 7 – Mức kết dính chức năng. Đây là mức cao nhất theo thang đo của Constantine và Yourdon đã đề cập ở trên, nghĩa là mỗi phần của mô đun đều là cần thiết để thi hành cùng một chức năng nào đó. Thật vậy, với từng chức năng trong sản phẩm đều được thiết kế theo mô hình MVP, và chúng cùng tham gia vào thực hiện chức năng ấy với những vai trò khác nhau. Sự thay đổi ở một thành phần có thể kéo theo sự thay đổi của một loạt các thành phần khác trong cùng mô đun chức năng đó. Chẳng hạn như một tham số nào đó mà View nhận vào bị thay đổi kiểu dữ liệu, sẽ buộc Presenter tương ứng với View đó phải điều chỉnh lại cách tiếp nhận, kéo theo thành phần Model sẽ thay đổi cách thức tương tác với dữ liệu và cách mà Presenter cập nhật lại View cũng sẽ khác đi. Một đặc trưng khác trong mô hình MVP là mối quan hệ 1 – 1 giữa Presenter và View, nghĩa là một Presenter chỉ tham gia điều khiển một View, điều này càng cho thấy độ kết dính cao giữa các thành phần trong mô đun. Về độ ghép nối, hầu hết giữa các mô đun trong hệ thống có sự ghép nối dữ liệu: Các mô đun trao đổi thông tin thông qua tham số và giá trị trả lại. Với đặc trưng là ứng dụng trên thiết bị di động nên các thao tác chuyển màn hình là tương đối nhiều, mỗi màn hình thường tương ứng với nhiều chức năng, và để hiển thị đúng thông tin cũng như thực hiện đúng các chức năng, cần có sự trao đổi dữ liệu qua lại giữa các màn hình này. Bởi vậy, sự ghép nối dữ liệu giữa các mô đun là lựa chọn phù hợp nhất. Lấy ví dụ với 2 mô đun chức năng: hiển thị danh sách sản phẩm và hiển thị chi tiết sản phẩm. Chức năng hiển thị chi tiết sản phẩm cần nhận vào tham số là một đối tượng sản phẩm hoàn chỉnh hoặc một mã sản phẩm. Tham số này được truyền từ màn hình chức năng danh sách sản phẩm khi người dùng thực hiện thao tác chọn. Ghép nối dữ liệu cũng là mức ghép nối tương đối thấp trong thang đo đã nêu, và do đó đảm bảo nguyên tắc low coupling giữa các mô đun của hệ thống.

Như vậy, Chương 5 đã trình bày về những đóng góp và giải pháp nổi bật trong đề án. Chương 6 tiếp đây là phần nội dung kết thúc, nơi em đưa ra những kết quả đạt được sau cùng và những kinh nghiệm, bài học thu được, bên cạnh đó là hướng phát triển sau này của sản phẩm.

Chương 6 Kết luận và hướng phát triển

6.1 Kết luận

Trong suốt quá trình thực hiện đồ án, em đã học hỏi thêm được rất nhiều kiến thức và kỹ năng, không chỉ riêng trong việc lập trình, mà còn trong cách tiếp cận, giải quyết bài toán và phân tích, thiết kế hệ thống. Tuy nhiên, vì đề tài mà đồ án hướng đến là tương đối rộng, đòi hỏi quá trình tìm hiểu và thực hiện lâu dài nên bản thân em đã gặp phải không ít những khó khăn. Bởi vậy dẫn đến sản phẩm thực hiện được còn một số điểm hạn chế như hiệu năng chưa thực sự tốt, một vài chức năng vẫn còn mang tính thử nghiệm và cần phát triển thêm nếu muốn đưa vào thực tế. Để so sánh với những sản phẩm thương mại điện tử hiện đã xuất hiện trên thị trường, ứng dụng mà em xây dựng có thể vẫn chưa thực sự nổi trội hơn nhiều do những thiếu sót về kinh nghiệm thực tế hoặc hạn chế về kiến thức cũng như kỹ năng. Tuy nhiên, ứng dụng phần nào đó cũng đã đưa ra được những giải pháp công nghệ tốt cho một bài toán có quy mô rộng với nghiệp vụ phức tạp, đồng thời khắc phục một số điểm bất tiện trong các ứng dụng tương tự, cung cấp thêm cho người dùng những chức năng tiện ích mới.

Đóng góp nổi bật nhất của bản thân là việc em đã vận dụng kết hợp được những kiến thức được học trên ghế nhà trường và các công nghệ mới vào một vấn đề cụ thể để tạo ra được sản phẩm hoàn chỉnh có tính ứng dụng tốt. Từ những thành công và cả những sai sót gặp phải trong quá trình thực hiện đồ án, em đã rút ra được nhiều bài học kinh nghiệm đáng quý. Đó là với bất kỳ sản phẩm nào, việc tìm hiểu trước yêu

cầu trực tiếp từ khách hàng là rất quan trọng, khách hàng thật sự cần gì, muốn gì, đó mới chính là cái đích mà sản phẩm hướng đến sau cùng. Một chức năng được tạo ra không vì lợi ích người dùng sẽ chỉ là một chức năng dư thừa cần loại bỏ. Bên cạnh đó, xét trong lĩnh vực phát triển phần mềm, cần tuân thủ chặt chẽ thứ tự các bước thực hiện, từ phân tích yêu cầu đến thiết kế hệ thống, thiết kế chi tiết, xây dựng, kiểm thử và triển khai. Được trực tiếp thực hiện từng bước trong một vòng đời phát triển phần mềm càng giúp em hiểu rõ thêm tầm quan trọng của chúng, việc bỏ sót hoặc thực hiện không đúng bất kỳ giai đoạn nào sẽ dẫn đến những khó khăn và lỗi phát sinh không lường trước, có thể dẫn đến sụp đổ của cả một dự án.

6.2 Hướng phát triển

Mục đích sản phẩm hướng đến là một sàn thương mại điện tử hoàn chỉnh được sử dụng bởi nhiều đối tượng khác nhau. Và để thực hiện được mục tiêu đó, ứng dụng cần hoàn thiện thêm một số mô đun chức năng như liên kết thanh toán với tài khoản ngân hàng hoặc chức năng dành cho tác nhân người giao hàng (hiện tác nhân này chưa được bao hàm trong hệ thống). Bên cạnh đó cần bổ sung một số chức năng như cho phép bình luận và phản hồi bình luận trong sản phẩm, tính năng chat giữa chủ shop và khách hàng, tính năng thông báo của hệ thống. Những chức năng này khi được hoàn thiện sẽ giúp tăng tương tác tối đa giữa người bán và người mua. Người bán dễ dàng nắm bắt nhu cầu của khách hàng hơn và người mua có thể chủ động tìm kiếm được sản phẩm mình mong muốn.

Về hướng đi trong tương lai, sản phẩm có thể được tái cấu trúc cho phép mang lại hiệu năng và trải nghiệm tốt hơn, đặc biệt là khi số lượng người dùng tăng lên. Việc tái cấu trúc ở đây có thể là sử dụng kết hợp mô hình MVP với các thư viện Dagger và RxAndroid giúp giảm số lượng các method, đơn giản hóa mã nguồn cũng như kiểm soát tốt hơn các thao tác xử lý bất đồng bộ. Một hướng đi nữa là kết hợp ứng dụng với một website thương mại điện tử để tạo thành một hệ thống thống nhất. Điều này là hoàn toàn khả thi bởi API được xây dựng phía server có thể được sử dụng bởi bất cứ nền tảng hoặc công nghệ nào.

Tài liệu tham khảo

- [1] Greg Nudelman, Android Design Patterns: Interaction Design Solutions for Developers, 2013.
- [2] Ian Darwin, Android Cookbook: Problems and Solutions for Android Developers, 2012.
- [3] David Sklar, Adam Trachtenberg, PHP Cookbook, 2003.
- [4] <https://developer.android.com/>, last visited May 2018.
- [5] <http://php.net/docs.php>, last visited May 2018.
- [6] <https://www.uml-diagrams.org/>, last visited May 2018.