# Description: khung doi

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**\_\_\_\_\_\_\_\_\_\_\_\_\_**🙠⬩🙢**\_\_\_\_\_\_\_\_\_\_\_\_\_**



**THỰC TẬP CƠ SỞ**

**ĐỀ TÀI: XÂY DỰNG APP BÁN ĐIỆN THOẠI, LAP TOP**

**Giảng viên hướng dẫn : Nguyễn Thế Vịnh**

**Sinh viên thực hiện : Nguyễn Phương Nam**

**Lớp: KTPM – K18A**

*Thái Nguyên, năm 2022*

# **LỜI MỞ ĐẦU**

Hiện nay với sự phát triển nhanh chóng của khoa học công nghệ, các thiết bị điện thoại di động ngày càng trở nên phổ biến. Cùng với sự phát triển của mạng 4G, 5G và sự ra đời của các dòng điện thoại cao cấp ngày càng nhiều. Chiếc điện thoại không chỉ được sử dụng với chức năng nghe gọi thông thường mà còn được sử dụng với rất nhiều mục đích khác nhau như nghe nhạc, chơi game, shopping online... giúp bạn thư giãn, mua bán theo như cầu của bạn.

Trong vài năm trở lại đây, sự phát triển của Android đã có những bước phát triển mạnh mẽ và dần chiếm cảm tình của người dùng. Lập trình ứng dụng mobile trên nền tảng Android ngày càng đa dạng và phổ biến, thu hút nhiều lập trình viên. Từ nhu cầu đó, đã có rất nhiều công ty lớn nhỏ đã cho ra đời hàng loạt các ứng dụng để cung cấp cho người sử dụng. Tuy vậy, những ứng dụng mua sắm online dường như vẫn chưa thể đáp ứng hết nhu cầu của người dùng. Việc xây dựng các ứng dụng trên thiết bị di động là rất cần cho việc phát triển của công nghệ di động.

Với những ứng dụng quen thuộc sẽ tiếp cận được với rất nhiều lứa tuổi như shopee, Lazada, Sen đỏ…. Em mong muốn có thể tự mình xây dựng một ứng dụng bán hàng phát triển kiến thức nên em đã lựa chọn xây dựng một ứng dụng bán điện thoại cho riêng mình, được lấy ý tưởng dựa trên các app bán hàng nổi tiếng trênmôi tường internet. Nhằm đáp ứng nhu cầu mua sắm cho mọi người, với những người bận rộn thì việc sủ dụng ứng dụng mua hàng online là một sự lựa chọn đúng đắn.

Đề tài giúp em củng cố thêm được nhiều kiến thức đã học: Ngôn ngữ lập trình Java, Lập trình trên thiết bị di động, Cơ sở dữ liệu, Phân tích thiết kế hệ thống,… và các kỹ năng khác. Những phần em tìm hiểu và đã làm được là nhờ quá trình học tập và tự tìm hiểu của bản thân. Hi vọng đề tài sẽ nhận được sự quan tâm của mọi người.

# **MỤC LỤC**

[1](#_Toc98957098)

[**LỜI MỞ ĐẦU** 2](#_Toc98957099)

[**MỤC LỤC** 1](#_Toc98957100)

[**CHƯƠNG I: TỔNG QUAN ĐỀ TÀI** 3](#_Toc98957101)

[**1.** **Lí do lựa chọn đề tài** 3](#_Toc98957102)

[**2.** **Mục tiêu đề tài** 3](#_Toc98957103)

[**3.** **Nội dung** 3](#_Toc98957104)

[**4.** **Phương pháp nghiên cứu** 3](#_Toc98957105)

[**5.** **Ý nghĩa lý luận và thực tiễn** 4](#_Toc98957106)

[**CHƯƠNG II: CƠ SỞ LÝ THUYẾT** 5](#_Toc98957107)

[**1.** **Giới thiệu về Android** 5](#_Toc98957108)

[**1.1** **Lịch sử hình thành** 5](#_Toc98957109)

[**1.2** **Ưu nhược điểm của Android** 7](#_Toc98957110)

[**1.3** **Kiến trúc cơ bản của hệ điều hành Android** 8](#_Toc98957111)

[**1.4** **Quá trình biên dịch và thực thi ứng dụng** 10](#_Toc98957112)

[**1.5** **Các thành phần chính của một ứng dụng Android** 11](#_Toc98957113)

[**2.** **Ngôn ngữ Java** 12](#_Toc98957114)

[**2.1** **Các đặc điểm của ngôn ngữ Java** 12](#_Toc98957115)

[**3.** **Giới thiệu môi trường phát triển Android Studio** 15](#_Toc98957116)

[**3.1** **Cấu trúc của một project** 16](#_Toc98957117)

[**3.2** **File manifest** 17](#_Toc98957118)

[**3.3** **File MainActivity.java** 19](#_Toc98957119)

[**3.4** **File strings.xml** 20](#_Toc98957120)

[**3.5** **Giao diện ứng dụng (Layout)** 20](#_Toc98957121)

[**CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 22](#_Toc98957122)

[**3.1** **Mô tả bài toán** 22](#_Toc98957123)

[**Nhu cầu thực tế** 22](#_Toc98957124)

[**3.2** **Phân tích hệ thống** 22](#_Toc98957125)

[***3.2.1 Phần dành cho khách hàng*** 22](#_Toc98957126)

[***3.2.2 Phần dành người quản lí*** 24](#_Toc98957127)

[**3.3** **Phân tích hệ thống** 25](#_Toc98957128)

[**3.3.1** **Biểu đồ Use Case tổng quát** 25](#_Toc98957129)

[**3.3.2** **Biểu đồ Use Case chức năng quản lí giỏ hàng** 25](#_Toc98957130)

[**3.3.3** **Biểu đồ Use Case chức năng quản lí sản phẩm** 26](#_Toc98957131)

[**3.3.4** **Biểu đồ Use Case chức năng quản thống kê** 26](#_Toc98957132)

[**3.3.5** **Biểu đồ hoạt động chúc năng đăng kí** 27](#_Toc98957133)

[**3.3.6** **Biểu đồ hoạt động chúc năng đăng nhập** 27](#_Toc98957134)

[**3.3.7** **Biểu đồ hoạt động chức năng thêm sản phẩm vào giỏ hàng** 28](#_Toc98957135)

[**3.3.8** **Biểu đồ hoạt động chức năng tìm kiếm** 28](#_Toc98957136)

[**3.3.9** **Biểu đồ hoạt động chức năng thêm sản phẩm** 28](#_Toc98957137)

[**3.3.10** **Biểu đồ hoạt động chức năng xóa sản phẩm** 29](#_Toc98957138)

[**3.3.11** **Biểu đồ hoạt động chức năng sửa sản phẩm** 29](#_Toc98957139)

[**3.3.12** **Biểu đồ trình tự chức năng đăng nhập** 30](#_Toc98957140)

[**3.3.13** **Biểu đồ trình tự chức năng đăng kí** 30](#_Toc98957141)

[**3.3.14** **Biểu đồ trình tự chức năng thêm sản phẩm** 31](#_Toc98957142)

[**3.3.15** **Biểu đồ trình tự chức năng sửa sản phẩm** 31](#_Toc98957143)

[**3.3.16** **Biểu đồ trình tự chức năng xóa sản phẩm** 32](#_Toc98957144)

[**3.3.17** **Biểu đồ trình tự chức năng đặt hàng** 32](#_Toc98957145)

[**3.3.18** **Biểu đồ lớp chi tiết** 33](#_Toc98957146)

[**3.3.19** **Cơ sở dữ liệu** 33](#_Toc98957147)

[**3.4** **Demo** 34](#_Toc98957148)

[**3.4.1** **Giao diện người dùng** 34](#_Toc98957149)

[**KẾT LUẬN** 34](#_Toc98957150)

[1. **Kết quả đạt được** 34](#_Toc98957151)

[2. **Hạn chế** 34](#_Toc98957152)

[3. **Hướng phát triển** 34](#_Toc98957153)

[4. **Kinh nghiệm thu được** 34](#_Toc98957154)

[**TÀI LIỆU THAM KHẢO** 35](#_Toc98957155)

# **CHƯƠNG I: TỔNG QUAN ĐỀ TÀI**

1. **Lí do lựa chọn đề tài**

Lập trình trên thiết bị di động đang là xu hướng của lập trình viên trên toàn thế giới vì vậy em muốn trau dồi kiến thức của mình để rèn luyện khả năng lập trình cũng như học hỏi được thêm nhiều kiến thức phục vụ cho công việc sau này. Trên thực tế, nhiều người đã và đang sử ụng các ứng dụng mua bán online, chính vì mong muốn được tìm hiểu và khám phá kiến thức về Lập trình android nên em đã lựa chọn đề tài xây dựng một ứng dụng bán điện thoại.

1. **Mục tiêu đề tài**

Tìm hiểu về Android và cách xây dựng 1 phần mềm nhằm thực hành những kiến thức mình đã được học. Từ phần mềm được xây dựng có thể cài đặt trên các thiết bị trên thực tế và nhiều loại thiết bị khác nhau.

1. **Nội dung** 
   1. *Đối tượng nghiên cứu*

Ngôn ngữ Java, các công cụ lập trình như: Android Studio, Eclipse.... sử dụng cơ sở dữ liệu để lưu trữ (SQL sever bằng công cụ xampp), được nghiên cứu đưa ứng dụng lên thiết bị di động (đóng gói ứng dụng).

* 1. *Phạm vi nghiên cứu*

Đề tài xây dựng ứng dụng câu hỏi trắc nghiệm mô phỏng theo chương trình trên truyền hình. 

Dựa theo kiến thức Java, Cơ sở dữ liệu,.. lựa chọn một môi trường xây dựng ứng dụng hợp lý.

1. **Phương pháp nghiên cứu**

Để đạt được mục tiêu đặt ra của đề tài, em đã sử dụng các phương pháp sau:

* Thứ nhất, sử dụng một số phương pháp thống kê như: phân tích, tổng hợp, so sánh áp dụng các với các phần mềm đã có.
* Thứ hai, sử dụng các phương pháp chuyên dụng trong việc phát triển hệ thống thông tin như: phương pháp phân tích, thiết kế hướng chức năng, hướng sự kiện, phương pháp lập trình hướng đối tượng...

1. **Ý nghĩa lý luận và thực tiễn**

* Về mặt lý luận: Đề tài đã cung cấp các cơ sở lý thuyết về ngôn ngữ lập trình Android cho học tập và nghiên cứu.
* Về mặt thực tiễn: Trên cơ sở kết quả của việc nghiên cứu tìm hiểu, từ đó viết ra 1 ứng dụng cụ thể mang tính thực tiễn cao.

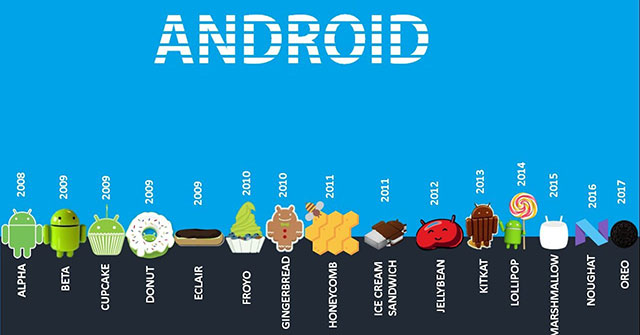
# **CHƯƠNG II: CƠ SỞ LÝ THUYẾT**

1. **Giới thiệu về Android**
2. **Lịch sử hình thành**

[Như các bạn đã biết Android là hệ điều hành di động phổ biế](https://text.123docz.net/trang-chu.htm)n nhất hiện tại và cũng là hệ điều hành có sự phát triển nhanh nhất.

Lịch sử của Android ra đời vào vào năm 2005, là một phần của chiến lược không gian mobile. Nó là hệ điều hành nhân Linux và sau đó được Goolge mua lại và họ đã đưa Android trở thành một hệ điều hành mã nguồn mở. Điều này có nghĩa là bạn có thể download mã nguồn Android và phục vụ cho những mục đích riêng của mình. Hiện tại thì Android xuất hiện hầu như trong các sản phẩm của các nhà sản xuất lớn Samsung, Sony, Htc…từ điện thoại thông minh - smart phone cho tới máy tính bảng, ti vi…Và có thể nói đây là một trong những nguyên nhân để nó phát triển mạnh tới vậy cũng là tiềm năng lớn cho các lập trình viên với hệ điều hành này.

Hệ điều hành Android phát triển ngày càng lớn mạnh và kể từ khi nó ra đời tới giờ nó đã có các phiên bản sau.



Android có thể khiến nhiều người bối rối bởi có rất nhiều phiên bản và rất nhiều trong số chúng vẫn đang chạy ngày nay. Và việc luôn chạy phiên bản mới nhất có thể là một thử thách không nhỏ.

Thông thường, các bản Android chính được phát hành mỗi năm 1 lần (dù không phải khi nào cũng vậy) cùng với đó là các bản cập nhật bảo mật giữa 2 lần phát hành. Thi thoảng Google cũng phát hành các bản cập nhật (.1, .2…) không thường xuyên. Họ cũng phát hành các bản cập nhật lớn, dù không lớn như bản đầy đủ, như cập nhật từ Android 8.0 lên Android 8.1 chẳng hạn.

Dưới đây là danh sách tóm tắt các bản Android cùng tên và ngày phát hành.

|  |  |  |
| --- | --- | --- |
| **Phiên bản** | **Tên** | **Ngày phát hành** |
| Android 1.5 | Cupcake | 27/4/2009 |
| Android 1.6 | Donut | 15/9/2009 |
| Android 2.0 - 2.1 | Eclair | 26/9/2009 (phát hành lần đầu) |
| Android 2.2 - 2.2.3 | Froyo | 20/5/2010 (phát hành lần đầu) |
| Android 2.3 - 2.3.7 | Gingerbread | 6/12/2010 (phát hành lần đầu) |
| Android 3.0 - 3.2.6 | Honeycomb | 22/2/2011 (phát hành lần đầu) |
| Android 4.0 - 4.0.4 | Ice Cream Sandwich | 18/10/2011 (phát hành lần đầu) |
| Android 4.1 - 4.3.1 | Jelly Bean | 9/7/2012 (phát hành lần đầu) |
| Android 4.4 - 4.4.4 | KitKat | 31/10/2013 (phát hành lần đầu) |
| Android 5.0 - 5.1.1 | Lollipop | 12/11/2014 (phát hành lần đầu) |
| Android 6.0 - 6.0.1 | Marshmallow | 5/10/2015 (phát hành lần đầu) |
| Android 7.0 - 7.1.2 | Nougat | 22/8/2016 (phát hành lần đầu) |
| Android 8.0 - 8.1 | Oreo | 21/8/2017 (phát hành lần đầu) |

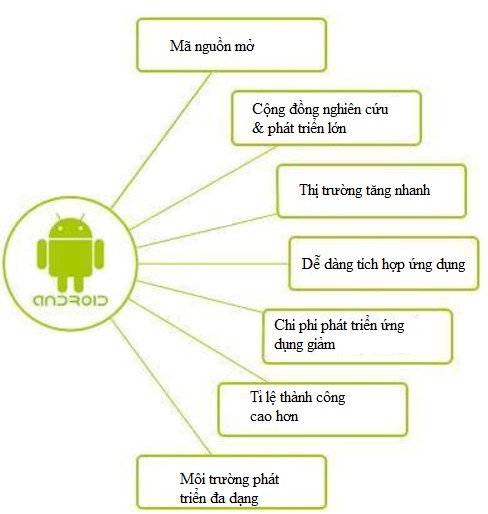
Nhìn vào bảng trên, có thể thấy thời gian cập nhật hệ thống của Android không theo 1 lịch trình cụ thể nào. Nhưng kể từ Ice Cream Sandwich thì OS được cập nhật theo năm.

1. **Ưu nhược điểm của Android**

Android là một hệ điều hành nguồn mở trong đó người dùng có thể sửa đổi, cải tiến, phát triển và nâng cấp theo một số nguyên tắc đã được quy định trước, nó cung cấp một cách tiếp cận thống nhất để phát triển ứng dụng. Phần mềm mã nguồn mở sẽ gần gũi với người dùng hơn bởi chính những người sử dụng là người tạo ra phần mềm đó.

Tính tuỳ biến: mã nguồn mở còn đa dạng trong tuỳ biến nguồn dữ liệu. Những đoạn mã trong chương trình được công khai, nên người dùng có thể thêm các chức năng mà người dùng muốn có.

Android được phát triển và nâng cấp rất nhiều. Chính vì công nghệ mở nên cộng đồng những nhà phát triển nền tảng và ứng dụng Android có quyền truy cập, điều chỉnh, nâng cấp và hoàn thiện hơn về tính năng và hiệu quả của nó.



Android miễn phí cho sử dụng thương mại. Android có một công ty lớn nhất, mạnh nhất và sáng tạo nhất đứng đằng sau nó: Google. Thêm vào đó, còn có các tập đoàn lớn khác trong hiệp hội OHA (Open Handset Alliance – Hiệp hội những nhà phát triển các thiết bị cầm tay mở) đứng sau đầu tư hàng tỷ đôla hỗ trợ cho việc nâng cấp và phát triển nền tảng Android. Do đó, việc phát triển Android không những không làm tốn tiền mà còn mở ra rất nhiều cơ hội cho các nhà phát triển phần mềm di động.

Phát triển và quảng bá ứng dụng Android dễ dàng: Chỉ cần cài đặt môi trường Android và nắm vững công nghệ của nó, bạn có thể tạo ra những ứng dụng tối ưu cho cộng đồng, Android cung cấp “chợ ứng dụng” Android Market giúp cho việc quảng bá ứng dụng của nhà phát triển với cộng đồng người sử dụng hoàn toàn dễ dàng với chi phí thấp.

Đối với người sử dụng thiết bị, Android mang lại thuận lợi. Android có kho ứng dụng khổng lồ: Người dùng có thể tìm ra rất nhiều ứng dụng miễn phí phù hợp với mục đích của mình. Android hỗ trợ đa nhiệm: Hệ điều hành Android trên Olive Pad hỗ trợ tính năng đa nhiệm, cho phép chạy nhiều ứng dụng cùng một lúc. Ví dụ, trong khi nghe nhạc, người dùng có thể lướt web, chơi game, đọc tài liệu ...

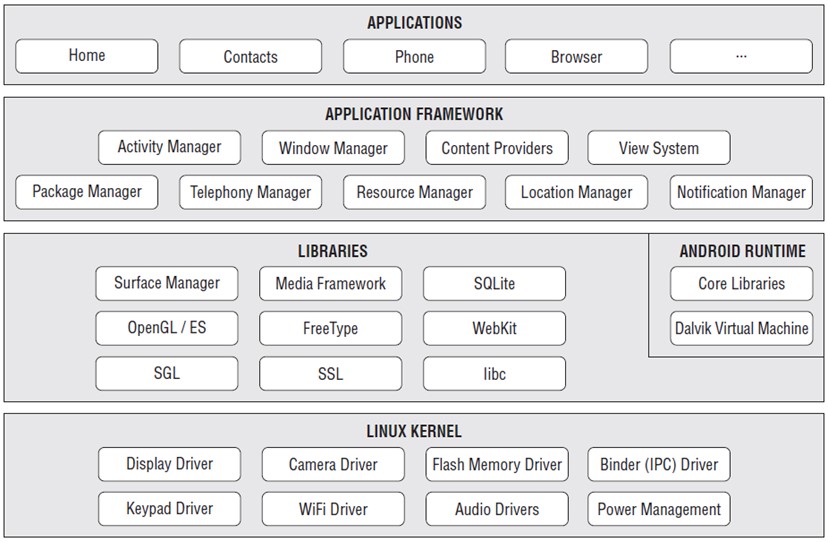
Bên cạnh đó Android cũng còn có một số hạn chế riêng so với những hệ điều hành di động khác như:

Bảo mật: Vì là hệ điều hành nguồn mở nên tất cả những thông tin về hệ thống bất kì ai cũng có thể nắm được. Đây là ưu điểm nhưng cũng là nhược điểm, bởi các hacker có thể tìm kiếm những lỗ hổng hệ thống và tìm cách tấn công các thiết bị.

Đột phá ý tưởng: Mặc dù, Android đang trên đà phát triển nhanh với nhiều cải tiến mới, hoàn thiện hơn. Nhưng ở góc độ nào đó Android vẫn là người chạy theo những ý tưởng của người khổng lồ iPhone. Những người phát triển ứng dụng cho Android hầu như đều lấy ý tưởng từ iPhone. Đây chính là thách thức cho những nhà phát triển hệ điều hành cũng như những phát triển ứng dụng cho Android.

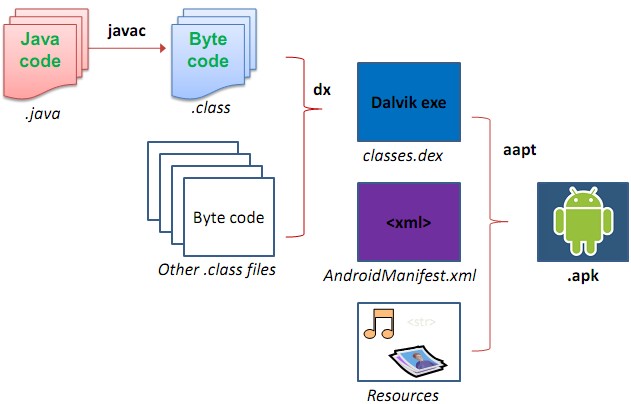
1. **Kiến trúc cơ bản của hệ điều hành Android**

Android là một môi trường phân tầng, xây dựng trên nền của hệ điều hành nhân Linux, và nó bao hàm nhiều chức năng phong phú. Hệ thống con giao diện người sử dụng gồm có: cửa sổ, khung nhìn, các tiểu trình để hiển thị các phần tử phổ biến như các hộp biên soạn, danh sách, danh sách thả xuống. Hình sau cho thấy một khung nhìn đơn giản hoá về các tầng phần mềm Android.



1. **Nhân Linux (Linux kernel)** là tầng thấp nhất, chính là phần lõi (core) của hệ điều hành Linux hoạt động trên bộ vi xử lý ARM (Acom RLSC Machine) hoặc Intel, sử dụng các trình điều khiển thiết bị để hoạt động với hệ thống phần cứng (màn hình, camera, bộ nhớ flash, card mạng, bàn phím, âm thanh, nguồn), quản lý bộ nhớ, điều khiển các tiến trình, hỗ trợ mạng v.v..
2. **Môi trường thực thi (Android runtime**) được xây dựng trên nhân Linux bao gồm hai thành phần: bộ thư viện lõi (core libraries), và máy ảo Dalvik. Môi trường thực thi chịu trách nhiệm khởi tạo và thực thi ứng dụng Android. Mỗi chương trình ứng dụng Android chạy trong một tiến trình riêng với một máy ảo Dalvik dành riêng cho ứng dụng đó. Máy ảo Dalvik là một phiên bản của máy ảo Java do Google phát triển có nhiệm vụ đọc bytecode (loại mã trung gian khi biên dịch mã nguồn Java) của chương trình, lưu trên máy Android dưới định dạng DEX, và tương tác với bộ thư viện lõi. Bộ thư viện lõi là một bộ phận của bộ thư viện lớp trong Java SE và cung cấp những chức năng cơ bản cho các chương trình Java. Dalvik có cơ chế thu dọn dữ liệu rác (Garbage Collector) cho phép tự động quản lý và thu hồi bộ nhớ từ các biến không sử dụng.
3. **Bộ thư viện hệ thống (System Libraries)** là bộ thư viện viết trên C/C++ và được tầng trên gọi thông qua giao diện Java. Bộ thư viện này chịu trách nhiệm cho những nhiệm vụ tính toán phức tạp (như đồ họa, phát âm thanh, truy cập cơ sở dữ liệu v.v..) và không phù hợp với thực thi trong máy ảo Dalvik. Phần này gồm thư viện Surface Manager (tạo các cửa sổ giao diện), OpenGL ES (hỗ trợ xây dựng ứng dụng đồ họa 2D và 3D), Media Framework (hỗ trợ xây dựng các ứng dụng về âm thanh, hình ảnh), SSL (cung cấp chức năng bảo mật thiết bị), SqLite (cơ sở dữ liệu quan hệ mã nguồn mở được nhúng trong thiết bị), Webkit (hỗ trợ hiền thị nội dung website), v.v.. Môi trường thực thi và bộ thư viện hệ thống có thể coi là hai tầng con của một tầng lớn gọi là Android Framework (để phân biệt với tầng Application trên cùng, tầng Application Framework, tầng Android Framework, và tầng lõi Linux).
4. **Khung ứng dụng (Application Framework)** chứa các thư viện Java hỗ trợ người dùng giao tiếp với tầng Android framework. Một phần của tầng này do Google cung cấp sẵn, một phần do người lập trình tạo ra. Tầng này cung cấp tất cả các loại dịch vụ sử dụng trong các ứng dụng ở tầng trên cùng (tầng Application). Quá trình phát triển ứng dụng cho Android luôn cần đến các dịch vụ của tầng này: Activity Manager (quản lý chu kì sống của các Activity trong ứng dụng Android), Telephony Manager (cung cấp thư viện để truy xuất đến các dịch vụ điện thoại cũng như là thông tin thuê bao), View system (xử lý giao diện trong ứng dụng Android), Location Manager (cung cấp thư viện hỗ trợ người dùng định vị vị trí của thiết bị).
5. **Ứng dụng (Aplication)** Chúng ta sẽ tìm thấy tất cả ứng dụng Android ở lớp trên cùng. Chúng ta sẽ viết ứng dụng của bạn để được cài đặt trên lớp này chỉ. Ví dụ về các ứng dụng như Sách, Trình duyệt, Trò chơi, v.v.
6. **Quá trình biên dịch và thực thi ứng dụng**

Trong nền tảng Android, mã nguồn Java được biên dịch thành các tập tin .class. Bộ SDK của Android chứa một chương trình đặc biệt gọi là DX (viết tắt của Dalvik Excecutable) để chuyển các tập tin .class về định dạng DEX. Một tập tin .class chứa thông tin một lớp, còn một tập tin .dex chứa tổng hợp của nhiều lớp. Tập tin .dex này sẽ được dùng để chạy trên máy ảo Dalvik tương tự như file .exe trên Windows. Để phục vụ việc phân phối ứng dụng, các tập tin .dex một lần nữa được đóng gói vào một tập tin .apk nhờ một chương trình đặc biệt trong SDK gọi là AAPT. Tập tin .apk này có thể tải lên kho ứng dụng hoặc copy vào bộ nhớ của thiết bị Android. Trong quá trình cài đặt, các tập tin .dex và các tài nguyên của chương trình sẽ được giải nén khỏi file .apk. Quy trình biên dịch và đóng gói chương trình trong Android được trình bày trong hình sau:



Khi thực thi một ứng dụng, Android sẽ tạo ra một máy ảo Dalvik cho riêng ứng dụng đó và chạy trong một tiến trình riêng biệt. Mỗi tiến trình này có vai trò tương đương như một người dùng trong hệ thống Linux. Một ứng dụng mặc định không có bất cứ quyền hạn gì tác động đến hệ điêu hành, người dùng hoặc ứng dụng khác. Đây là môi trường hoàn toàn cô lập bởi vì nó không có quyền xâm nhập vào ứng dụng hoặc tiến trình khác cũng như tiến trình khác cũng không có quyền xâm nhập vào nó. Do vậy, việc cho phép trao đổi thông tin và tương tác qua lại giữa các tiến trình với ứng dụng trong Android phải được định nghĩa trước trong ứng dụng đó để khi cài đặt hệ điều hành Android sẽ nhận diện được.

1. **Các thành phần chính của một ứng dụng Android**

Một ứng dụng Android bao gồm các thành phần chính sau đây:

* ***Activities:***Là khối cho phép xây dựng lên giao diện người dùng, tương tự như một cửa sổ hoặc một hộp thoại
* ***Content providers:***Cho phép trìu tượng hóa dữ liệu được lưu trữ trên thiết bị khi được truy cập bởi nhiều ứng dụng.
* ***Services:***Các Activities và content providers có thời gian tồn tại ngắn (thời gian sống) và có thể bị tắt bất cứ lúc nào. Còn các service được thiết kế để chạy liên tục và độc lập với các Activity, các service có thể kiểm tra việc cập nhật các RSS, play các bản nhạc, ...
* ***Intents:***là các message (thông điệp) hệ thống, chạy liên tục bên trong thiết bị để cảnh báo các ứng dụng về sự thay đổi các sự kiện hoặc trạng thái các thiết bị phần cứng (thẻ nhớ, ...), dữ liệu tới, các sự kiện ứng dụng khác. Người lập trình không thể phản hồi lại các intents nhưng có thể tạo ra các intents cho ứng dụng của mình.

1. **Ngôn ngữ Java**

Java là một ngôn ngữ lập trình, được phát triển bởi **Sun Microsystem** vào năm 1995, là ngôn ngữ kế thừa trực tiếp từ C/C++ và là một ngôn ngữ lập trình hướng đối tượng.

Vì sao ngôn ngữ này lại được đặt tên là Java? Java là tên một hòn đảo ở Indonesia - hòn đảo nổi tiếng với loại coffee Peet và cũng là loại nước uống phổ biến của các kỹ sư Sun. Ban đầu Ngôn ngữ này được đặt tên là "*Oak*" (*có nghĩa là "Cây sồi" - 1991*), nhưng các luật sư của Sun xác định rằng tên đó đã được đăng ký nhãn hiệu nên các nhà phát triển đã phải thay thế bằng một tên mới -  và cũng vì lý do trên mà cái tên Java đã ra đời và trở thành tên gọi chính thức của Ngôn ngữ này - Ngôn ngữ Lập trình Java.

1. **Các đặc điểm của ngôn ngữ Java**
2. *Đơn giản*

Java phát triển trên một ngôn ngữ dễ học và quen thuộc với đa số người lập trình là C và Java loại bỏ các phức tạp của C và C++ nhớ: con trỏ, đa kế thừa, định nghĩa chồng toán tử , không sử dụng lệnh “goto” cũng nhớ file header (.h), loại bỏ cấu trúc “struct” và “union”.

1. *Hướng đối tượng*

Java là ngôn ngữ lập trình hoàn toàn hướng đối tượng:

* Mọi thực thể trong hệ thống đều được coi là một đối tượng, tức là một thể hiện cụ thể của một lớp xác định.
* Tất cả các chương trình đều phải nằm trong một class nhất định.
* Không thể dùng Java để viết một chức năng mà không thuộc vào bất kì một lớp nào. Tức là Java không cho phép định nghĩa dữ liệu và hàm tự do trong chương trình.

1. *Độc lập về phần cứng hệ điều hành*

Đối với các ngôn ngữ lập trình truyền thống như C/C++, phương pháp biên dịch với một nền phần cứng khác nhau, có một trình biên dịch khác nhau để biên dịch mã nguồn chương trình cho phù hợp với nền phần cứng ấy. Do vậy, khi chạy trên một nền phần cứng khác, bắt buộc phải biên dịch lại mà nguồn.

Đối với chương trình viết bằng Java, thì cần cài máy ảo Java là có thể chạy được.

Một chương trình viết bằng ngôn ngữ Java có thể chạy tốt ở nhiều môi trường khác nhau. Gọi là khả năng “cross-platform”. Khả năng độc lập phần cứng và hệ điều hành được thể hiện ở 2 cấp độ là cấp độ mã nguồn và cấp độ nhị phân.

Ở cấp độ mã nguồn: Kiểu dữ liệu trong Java nhất quán cho tất cả các hệ điều hành và phần cứng khác nhau. Java có riêng một bộ thư viện để hỗ trợ vấn đề này. Chương trình viết bằng ngôn ngữ Java có thể biên dịch trên nhiều loại máy khác nhau mà không gặp lỗi.

Ở cấp độ nhị phân: Một mã biên dịch có thể chạy trên nhiều nền tảng khác nhau mà không cần dịch lại mã nguồn. Tuy nhiên cần có Java Virtual Machine để thông dịch đoạn mã này.

**Máy ảo Java (Java Virtual Machine)**

* Là một phần mềm dựa trên cơ sở máy tính ảo
* Là tập hợp các lệnh logic để xác định hoạt động của máy tính
* Được xem như là một hệ điều hành thu nhỏ
* Nó thiết lập lớp trừu tượng cho: phần cứng cho người bên dưới, hệ điều hành, mã đã biên dịch.

1. *Mạnh mẽ*

Java là ngôn ngữ yêu cầu chặt chẽ về kiểu dữ liệu:

* Kiểu dữ liệu phải được khai báo tường minh.
* Java không sử dụng con trỏ và các phép toán con trỏ.
* Java kiểm tra việc truy nhập đến mảng, chuỗi khi thực thi để đảm bảo rằng các truy nhập đó không ra ngoài giới hạn kích thước mảng.
* Quá trình cấp phát, giải phóng bộ nhớ cho biến được thực hiện tự động, nhờ dịch vụ thu nhặt những đối tượng không còn sử dụng nữa (garbage collection).
* Cơ chế bẫy lỗi của Java giúp đơn giản hóa qúa trình xử lý lỗi và hồi phục sau lỗi.

1. *Bảo mật*

Java cung cấp một môi trường quản lý thực thi chương trình với nhiều mức để kiểm soát tính an toàn:

* Ở mức thứ nhất, dữ liệu và các phương thức được đóng gói bên trong lớp. Chúng chỉ được truy xuất thông qua các giao diện mà lớp cung cấp.
* Ở mức thứ hai, trình biên dịch kiểm soát để đảm bảo mã là an toàn, và tuân theo các nguyên tắc của Java.
* Mức thứ ba được đảm bảo bởi trình thông dịch. Chúng kiểm tra xem bytecode có đảm bảo các qui tắc an toàn trước khi thực thi.
* Mức thứ tư kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.

1. *Phân tán*

Java được thiết kế để hỗ trợ các ứng dụng chạy trên mạng bằng các lớp Mạng (java.net). Hơn nữa, Java hỗ trợ nhiều nền chạy khác nhau.

1. *Thông dịch*

Ngôn ngữ lập trình thường được chia ra làm 2 loại (tùy theo các hiện thực hóa ngôn ngữ đó) là ngôn ngữ thông dịch và ngôn ngữ biên dịch.

* Thông dịch (Interpreter) : Nó dịch từng lệnh rồi chạy từng lệnh, lần sau muốn chạy lại thì phải dịch lại.
* Biên dịch (Compiler): Code sau khi được biên dịch sẽ tạo ra 1 file thường là .exe, và file .exe này có thể đem sử dụng lại không cần biên dịch nữa.

Ngôn ngữ lập trình Java thuộc loại ngôn ngữ thông dịch. Chính xác hơn, Java là loại ngôn ngữ vừa biên dịch vừa thông dịch. Cụ thể như sau:

Khi viết mã, hệ thống tạo ra một tệp .java. Khi biên dịch mã nguồn của chương trình sẽ được biên dịch ra mã byte code. Máy ảo Java (Java Virtual Machine) sẽ thông dịch mã byte code này thành machine code  (hay native code) khi nhận được yêu cầu chạy chương trình.

* Ưu điểm: Phương pháp này giúp các đoạn mã viết bằng Java có thể chạy được trên nhiều nền tảng khác nhau. Với điều kiện là JVM có hỗ trợ chạy trên nền tảng này.
* Nhược điểm : Cũng như các ngôn ngữ thông dịch khác, quá trình chạy các đoạn mã Java là chậm hơn các ngôn ngữ biên dịch khác (tuy nhiên vẫn ở trong một mức chấp nhận được).

1. *Đa luồng*

Java hỗ trợ lập trình đa tiến trình (multithread) để thực thi các công việc đồng thời. Đồng thời cũng cung cấp giải pháp đồng bộ giữa các tiến trình (giải pháp sử dụng priority…).

1. **Giới thiệu môi trường phát triển Android Studio**

Tháng 5/2013, Google cung cấp một công cụ phát triển ứng dụng Android trên Website chính thức dựa trên nền tảng IntelliJ IDEA gọi là Android Studio. Bộ công cụ Android Studio cung cấp những trình soạn thảo riêng biệt tương ứng với hầu hết các file cấu hình và Layout của ứng dụng Android với định dạng XML. Với những file Layout giao diện, Android Studio cho phép các lập trình viên dễ dàng chuyển đổi chế độ chỉnh sửa giữa trình biên soạn nội dung XML hoặc trình biên soạn dưới dạng giao diện viết tắt là GUI dễ dàng thao tác và chỉnh sửa.

Ngoài ra , bộ phát triển Android Studio còn tích hợp bên trong những tiện ích hỗ trợ phát triển ứng dụng như sau:

* Bộ xây dựng ứng dụng Gradle với nhiều cấu hình linh động. Cho phép xây dựng ứng dụng tùy biến và tự động tạo ra file .apk tương thích với thông tin tùy biến. Bộ code mẫu giúp chúng ta xây dựng các chức năng phổ biến của các ứng dụng.
* Trình biên soạn Layout GUI cho ứng dụng Android phong phú và tiện lợi, cho phép người dùng dễ dàng tạo ra giao diện màn hình bằng cách kéo thả các Component mẫu có sẵn và chỉnh sửa giao diện themes (kích thướt, màu sắc, …) tùy ý.
* Tích hợp lint – Một ứng dụng tích hợp giúp các developer kiểm soát về hiệu suất (performance), tính khả dụng (usability), khả năng tương thích của các phiên bản API sử dụng, và những vấn đề tiềm ẩn bên trong có thể xảy ra lúc Runtime.
* [ProGuard (tiện ích tối ưu và mã hóa code khi build ứng dụng](https://text.123docz.net/trang-chu.htm)) và Android appsigning.
* Bộ tích hợp hỗ trợ phát triển ứng dụng Android dễ dàng với các dịch vụ của nền tảng đám mây của Google (Google Cloud Platform)

1. **Cấu trúc của một project**

* Một project Android được tổ chức theo cấu trúc cây thư mục gồm các thành phần sau:
* androidmanifest.xml: Là file mô tả các thành phần ứng dụng như các services và các activities, ...
* build.xml: Là một file kịch bản phục vụ cho việc biên dịch và cài đặt ứng dụng trên thiết bị thật.
* project.properties: Được sử dụng bởi file kịch bản Ant.
* assets/: Là thư mục chứa các file tĩnh mà người lập trình muốn đóng gói để triển khai trên thiết bị.
* bin/: Chứa các file của ứng dụng khi nó đã được biên dịch
* gen/: Chứa các mã nguồn được sinh ra khi build các tool của Android.
* lib/: Chứa đựng các file thư viện mà ứng dụng sử dụng
* src/: Chứa các file mã nguồn java
* res/: Chứa các tài nguyên như các icon và các GUI layout.
* res/drawable: Chứa các ảnh (jpeg, png,...)
* res/menu: Mô tả XML cho các menu cụ thể
* res/layout: Mô tả XML cho các giao diện người dùng cụ thể
* res/raw: Các file dùng cho mục đích chung như(như file CSV chứa thông tin tài khoản).
* res/value: Chứa các string hoặc các dimensions
* res/xml: Chứa các file xml dùng chung.

1. **File manifest**

Mỗi một project Android có một phần từ khóa là file manifest (Androidmanifest.xml), file này chứa bảng nội dung của project, liệt kê tất cả các thành phần chính của ứng dụng, quyền, ... File này được sử dụng bởi Android runtime để ràng buộc ứng dụng vào hệ điều hành. File Androidmanifest.xml sẽ tự động được sinh ra khi ứng dụng dụng được tạo mới. *File manifest của các ứng dụng nhỏ, đơn giản thì làm việc rất tốt. Một file manifest phù hợp chỉ khoảng hơn 1000 dòng, trong trường hợp kích thước file manifest quá lớn thì ứng dụng có thể bị đơ khi triển khai.*

Nội dung của file manifest được đặt trong cặp thẻ manifest:

<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.commonsware.android.search">

...

</manifest>

//xmlns:android: chỉ namespace

//package: chỉ gói java mà ứng dụng sử dụng.

Các thẻ Permissions, Instrumentations, and Applications:

* Thẻ uses-permission: Xác định quyền cho các chức năng trong ứng dụng.
* Thẻ permission: xác định quyền các các activities và các servives được yêu cầu bởi các ứng dụng khác khi truy cập dữ liệu.
* Thẻ Instrumentation: Xác định phần code sẽ được thực thi khi các sự kiện phím hệ thống được nhấn phục vụ cho mục đích giám sát.
* Thẻ uses-library: là thẻ tùy chọn, phục vụ cho mục đích mapping các services.
* Thẻ uses-sdk: Xác định phiên bản SDK của Android sẽ được sử dụng để biên dịch.
* Thẻ Application: Định nghĩa nội dung chính cho ứng dụng.

Thí dụ file Androidmanifest.xml có nội dụng như sau:

<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.commonsware.android">

<uses-permission android:name="android.permission.ACCESS\_LOCATION" />

<uses-permission android:name="android.permission.ACCESS\_GPS" />

<uses-permission android:name="android.permission.ACCESS\_ASSISTED\_GPS" />

<uses-permission android:name="android.permission.ACCESS\_CELL\_ID" />

<application> ...

</application>

</manifest>

Phần tử application là nội dung chính của file manifest, khi tạo ứng dụng Android thì thẻ này có nội dung mặc định như sau:

<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.commonsware.android.skeleton"> <application>

<activity android:name=".Now" android:label="Now"> <intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" /> </intent-filter>

</activity>

</application>

</manifest>

Thẻ uses-sdk: sẽ cho phép chỉ định phiên bản SDK Android cũ nhất sẽ được sử dụng cho ứng dụng. Thí dụ:

<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.commonsware.android.search"> <uses-sdk minSdkVersion="2" /> ...

</manifest>

//Không cần chỉ định phiên bản sdk sau cùng.

1. **File MainActivity.java**

Main Activity code là một Java file với tên MainActivity.java. Đây là một file ứng dụng thực sự mà cuối cùng được chuyển đổi thành một Dalvik có thể thực thi và chạy ứng dụng của bạn. Sau đây là phần code mặc định được tạo cho ứng dụng HelloWorld (hiển thị dòng chữ “HelloWorld” ra màn hình). Ở đây, R.layout.activity\_main tham chiếu tới file activity\_main.xml được đặt trong thư mục res/layout. Phương thức onCreate() là một trong nhiều phương thức mà được tính toán khi một Activity được tải. Dưới đây là mã mặc định được tạo cho ứng dụng Hello World.

package com.example.helloworld; import android.os.Bundle; import android.app.Activity;

public class MainActivity extends Activity {

@Override public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

}

}

1. **File strings.xml**

File strings.xml nằm trong thư mục res / values và nó chứa tất cả các văn bản mà ứng dụng của bạn sử dụng. Ví dụ, tên của button, nhãn, văn bản mặc định, tên của ứng dụng ...

<resources>

<!--Tên của ứng dụng khi hiển thị trong điện thoại của bạn-->

<string name="app\_name"> Ứng dụng HelloWorld</string>

<!--Chuỗi văn bản hiển thị trên màn hình khi ứng dụng của bạn được mở-->

<string name="hello\_world">Hello World!\n Đây là ứng dụng Android đầu tiên của tôi</string> </resources>

1. **Giao diện ứng dụng (Layout)**

Layout (Giao diện ứng dụng): ở đây là một ví dụ đơn giản RelativeLayout (chúng ta sẽ nghiên cứu trong các chương 2). Các TextView là một thành phần giúp hiển thị các chuỗi trong ứng dụng Android và nó có các thuộc tính như android: layout\_width (dùng để thiết lập chiều rộng của TextView), android: layout\_height (dùng để thiết lập chiều cao), string (dùng để gọi đến file strings.xml nằm trong thư mục res / values). Do đó, string/hello\_world gọi đến chuỗi hello\_world định nghĩa trong file strings.xml để hiển thị. Nội dung file activity\_main.xml cảu ví dụ HelloWorld như sau:

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context="dev4u.com.helloworld.MainActivity">

<TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_centerInParent="true" android:text="@string/hello\_world" />

<!--Dòng android:layout\_centerInParent="true" dùng đểcăn giữa chiểu cao và chiều rộng màn hình, nhưng nó chỉ áp dụng cho RelativeLayout--> </RelativeLayout>

# **CHƯƠNG III: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

* 1. **Mô tả bài toán**

**Nhu cầu thực tế**

Cửa hàng bán điện thoại di động “**DiDongVietNam**” chuyên bán điện thoại của các hãng như: Apple, xiaomi, Samsung, oppo…Tiêu chí hoạt động của cửa hàng là làm sao để sản phẩm đến tay người dùng với giá hợp lí nhất thị trường và chất lượng cũng phải đảm bảo. giữ uy tín với khách hàng luôn là phương châm hoạt động của cửa hàng. Hiện tại cửa hàng chỉ phục vụ cho những khách hàng đến mua hàng trực tiếp tại cửa hàng

Nhưng với sự phát triển của công nghệ để không bị tụt lùi phía sau và cũng là nhu cầu mở rộng quy mô của cửa hàng. shop đã mong muốn được phục vụ tất cả mọi người ở xa mà không có thời gian đến mua trực tiếp tại cửa hàng.

* 1. **Phân tích hệ thống**

Ứng dụng được xây dựng sẽ phải thỏa mãn hai phần:

***3.2.1 Phần dành cho khách hàng***

Khách hàng là những người có nhu cầu mua các thiết bị điện tử được bán trong trong ứng dụng. Vì vậy ứng dụng phải thỏa mãn những yêu cầu sau:

* **Đăng Nhập:**

Để có thể tham gia mua hàng, xem sản phẩm, tìm kiếm… khách hàng cần phải đăng kí tài khoản (lưu ý những thông tin khách hàng đăng kí sẽ là thông tin được sử dụng để giao hàng khi khách hàng phát sinh việc mua hàng) sau đó đăng nhập để có thể thực hiện các chức năng bên trong ứng dụng.

* Đầu vào: Thông tin tài khoản khoản khách hàng.
* Xử lí: hệ thống sẽ đối chiếu với cơ sở dữ liệu, nếu thông tin đăng nhập đúng hệ thốn sẽ chuyển người dùng đến giao diện trang chủ sau khi click nút đăng nhập. Ngược lại nếu thông tin đăng nhập không đúng hệ thống sẽ hiện thị thông báo “thông tin tài khoản hoặc mật khẩu không chính xác”.
* Đầu ra: giao diện phù hợp.
* **Xem sản phẩm:**

Khách hàng có thể xem ảnh sản phẩm, và giá của sản phẩm ở dưới dạng một cardview nhỏ. Khi khách hàng bấm vào cardview đó, ứng dụng sẽ đưa khách hàng đến giao diện có thông tin chi tiết của sản phẩm.

Đọc giả truy cập vào địa chỉ website trên thanh url, chọn vào các trang trên website để xem tin tức.

* Đầu vào: Thông tin yêu cầu của đọc giả.
* Xử lý: Đọc giả chọn một tin tức bất kì trên website, thông tin của đọc giả sẽ được hệ thống xử lý. Hệ thống sẽ lấy dữ liệu từ cơ sở dữ liệu và phản hồi lại cho đọc giả.
* Đầu ra: Thông tin chi tiết về tin tức.
* **Tìm kiếm:**

Để không mất thời gian duyệt từng tin tức trong ứng dụng. Khách hàng có thể nhập thông tin về tin tức mình muốn đọc vào ô tìm kiếm

* Đầu vào: Tên sản phẩm mà khách hàng muốn tìm kiếm.
* Xử lý: Sau nhập keyword hệ thống sẽ so sánh các tên sản phẩm của đọc giả nhập với tên các sản phẩm có trong app, danh mục trong cơ sở dữ liệu. Nếu nhận được kết quả thì hệ thống sẽ trả về kết quả tìm kiếm cho khách hàng. Nếu không tìm thấy hệ thống sẽ thông báo không có kết quả nào.
* Đầu ra: Thông báo, kết quả từ hệ thống.
* **Mua Hàng:**

Khi khách hàng đã lựa chọn được một sản phẩm ưng ý, khách hàng có thể bấm nút thêm vào giỏ hàng, sau đó khách hàng bấm vào biểu tưởng giỏ hàng ở ở thanh toolbar để xem sản phẩm trong giỏ hàng. Tại giỏ hàng, khách hàng có thể tăng hoặc giảm số lượng tùy ý.

* Đầu vào: Sản phẩm khách hàng muốn mua.
* Xử Lí: Khi khách hàng đã lựa chọn được sản phẩm và tiến hành chỉnh số lượng, khi khách hàng bấm đặt hàng, ứng dụng sẽ chuyển khách hàng đến giao diện nhập địa chỉ đặt hàng. Nếu đặt hàng thành công hệ thống sẽ thông báo thành công, ngược lại nếu không thành công hệ thống sẽ đưa ra lỗi.
* Đầu ra: Thông báo của hệ thống.
* **Xem dơn hàng đã mua:**

Khách hàng có thế xem lại đơn hàng mình đã mua.

* Đầu vào: yêu cầu xem đơn hàng khách hàng
* Xử lí: hệ thống sẽ truy xuất thông tin từ cơ sở dự liệu để hiện thị cho khách hàng xem.
* Đầu ra: thông tin lịch sử mua hàng.
* **Liên hệ:**

Khi khách hàng có thắc mắc hoặc yêu cầu cần được giải đáp, khách hàng có thể liên hệ tới link facebook của chủ shop trong phần liên hệ.

***3.2.2 Phần dành người quản lí***

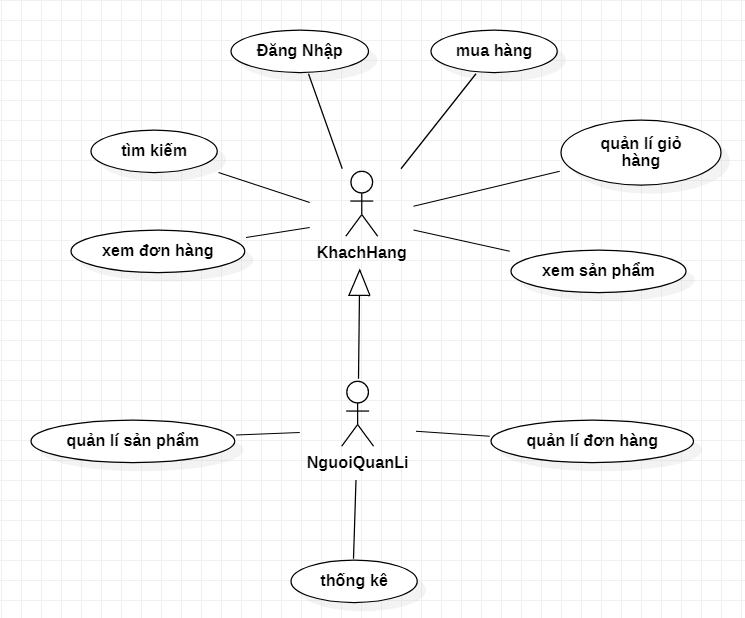
Người quản lí có các chức năng:

Quản lí sản phẩm(thêm, sửa, xóa), quản lí đơn, thống kê và tất cả các chức năng mà người dùng có.

* **Thêm Sản Phẩm:**

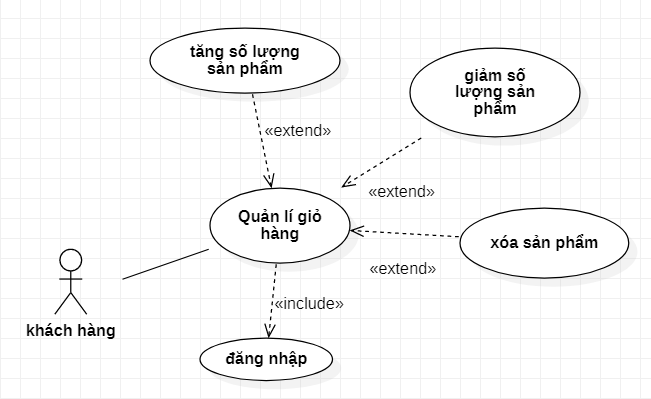
Khi người quản lí chọn chức năng thêm thêm sản phẩm ở trong danh mục quản lí sản phẩm, hệ thống sẽ đưa người quản lí đến giao diện thêm thông tin về sản phẩm mới. Sau khi người quản lí sẽ bấm vào button thêm sản phẩm, hệ thống sẽ phản hồi lại bằng thông báo.

* Đầu vào: thông tin của sản phẩm mới.
* Xử lí: hệ thống sẽ kiểm tra xem tên của sản phẩm đó đã có hay chưa? Nếu chưa có hệ thống sẽ cho phép thêm thông tin sản phẩm mới vào cơ sở dữ liệu, khi Người quản lí điền đầy đủ thông tin của sản phẩm đó.
* Đầu ra: hệ thống sẽ trả về thông báo phù hợp
  1. **Phân tích hệ thống**
     1. **Biểu đồ Use Case tổng quát**



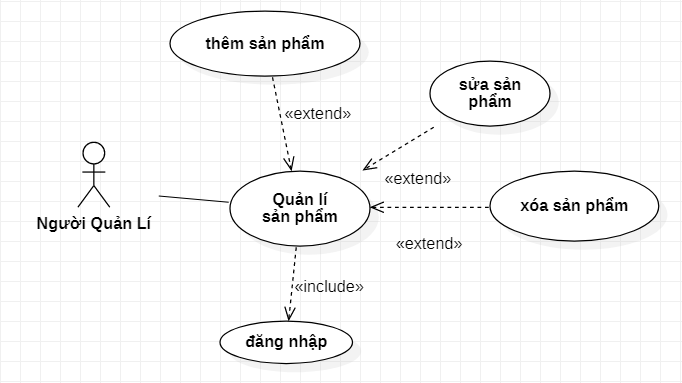
*Hình ảnh: Biểu đồ usecase tổng quát*

* + 1. **Biểu đồ Use Case chức năng quản lí giỏ hàng**



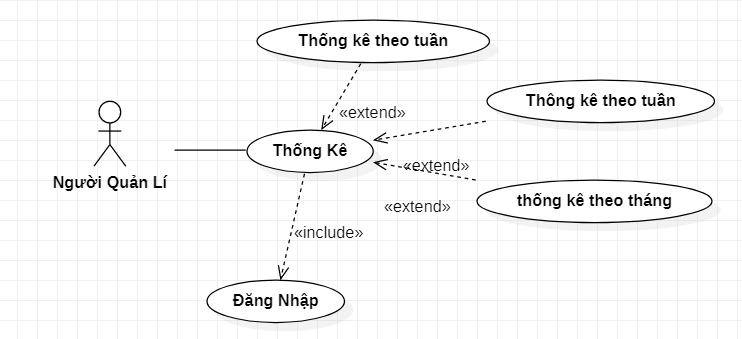
*Hình ảnh: biểu đồ usecase chức năng quản lí giỏ hàng.*

* + 1. **Biểu đồ Use Case chức năng quản lí sản phẩm**



*Hình ảnh: biểu đồ usecase chức năng quản lí sản phẩm*

* + 1. **Biểu đồ Use Case chức năng quản thống kê**

**

*Hình ảnh: biểu đồ usecase chức năng thông kê*

* + 1. **Biểu đồ hoạt động chúc năng đăng kí**

Diagram

Description automatically generated

*Hình ảnh: biểu đồ hoạt động chức năng đăng kí.*

* + 1. **Biểu đồ hoạt động chúc năng đăng nhập**

Diagram

Description automatically generated

*Hình ảnh: biểu đồ trình tự chức năng đăng nhập*

* + 1. **Biểu đồ hoạt động chức năng thêm sản phẩm vào giỏ hàng**

Diagram

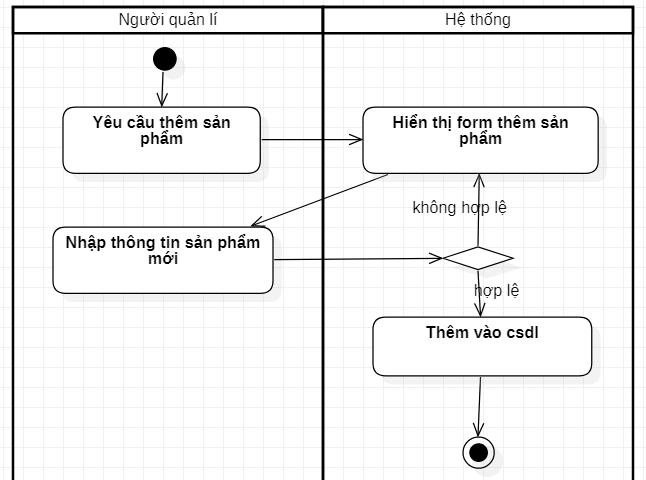
Description automatically generated with low confidence

* + 1. **Biểu đồ hoạt động chức năng tìm kiếm**

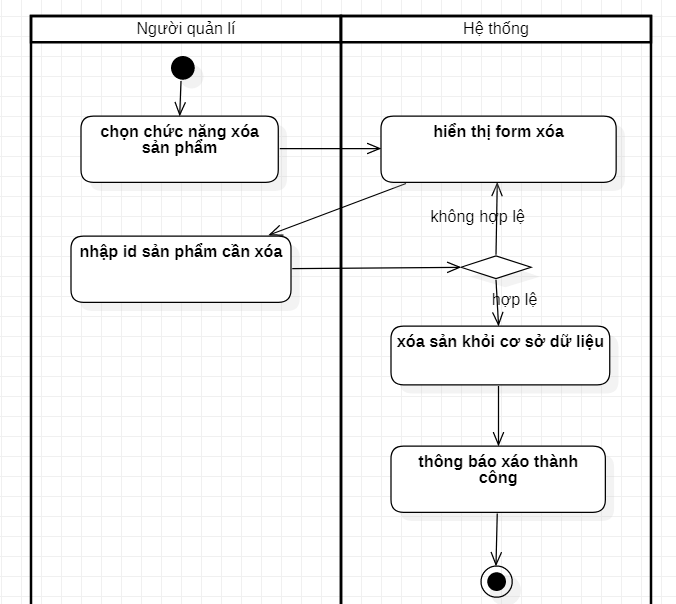
Diagram

Description automatically generated

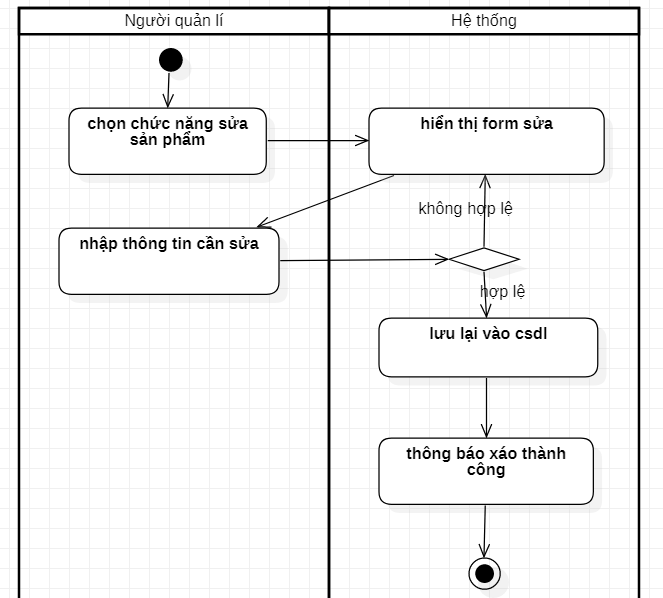
* + 1. **Biểu đồ hoạt động chức năng thêm sản phẩm**



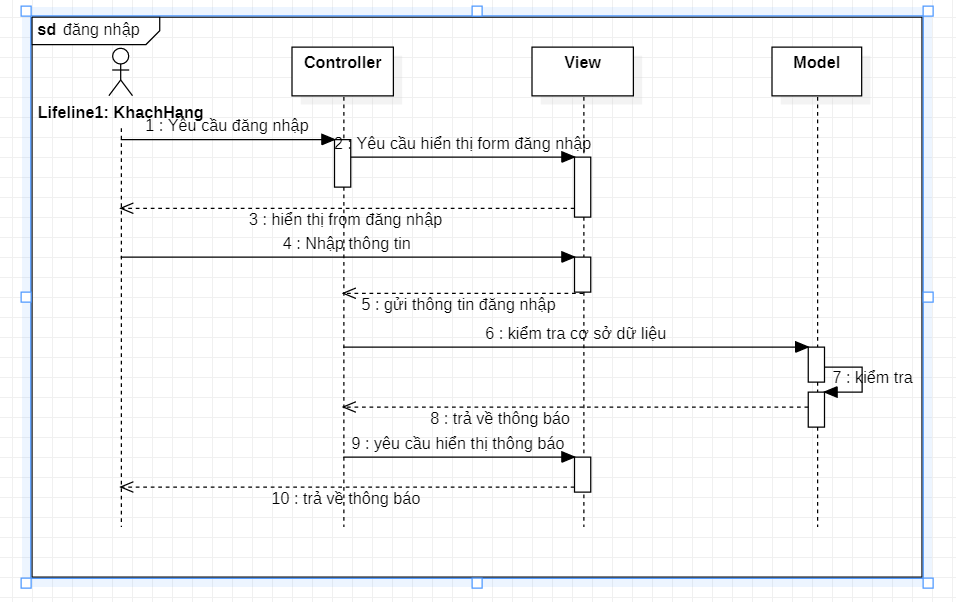
* + 1. **Biểu đồ hoạt động chức năng xóa sản phẩm**



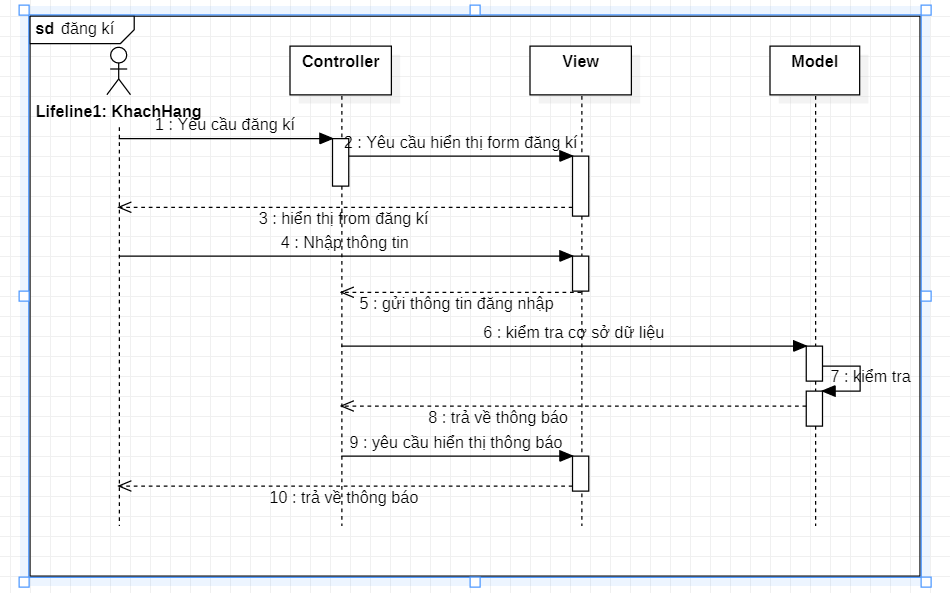
* + 1. **Biểu đồ hoạt động chức năng sửa sản phẩm**



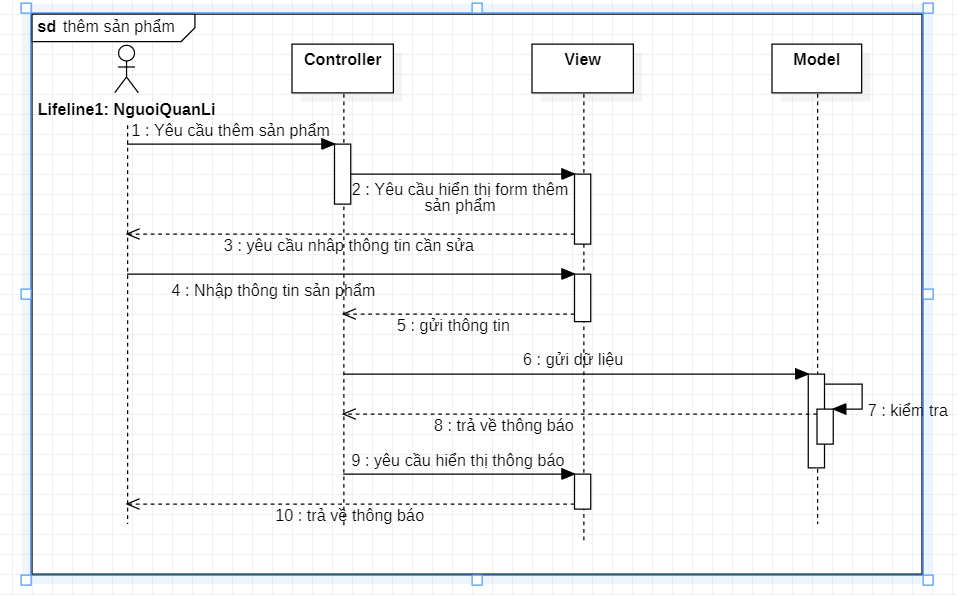
* + 1. **Biểu đồ trình tự chức năng đăng nhập**



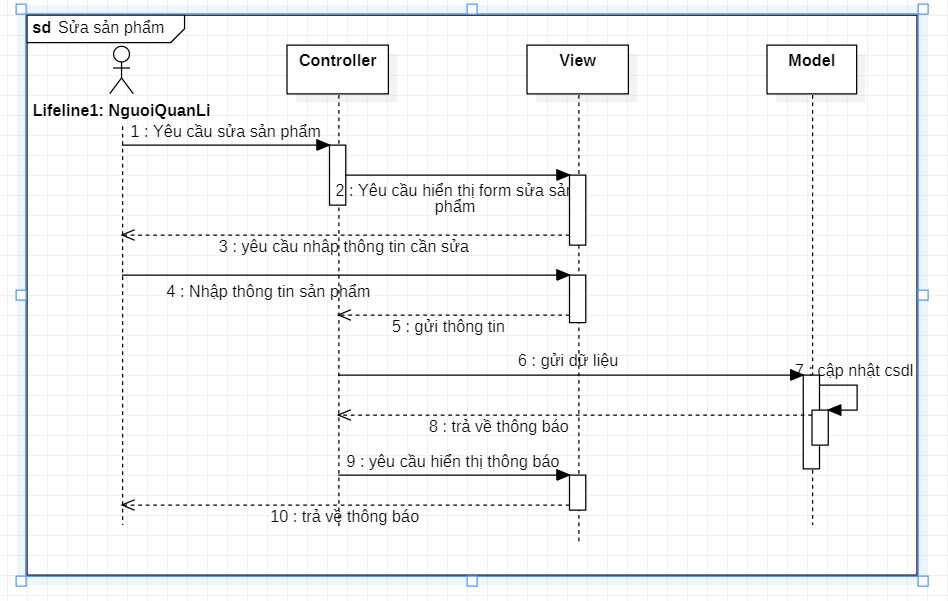
* + 1. **Biểu đồ trình tự chức năng đăng kí**



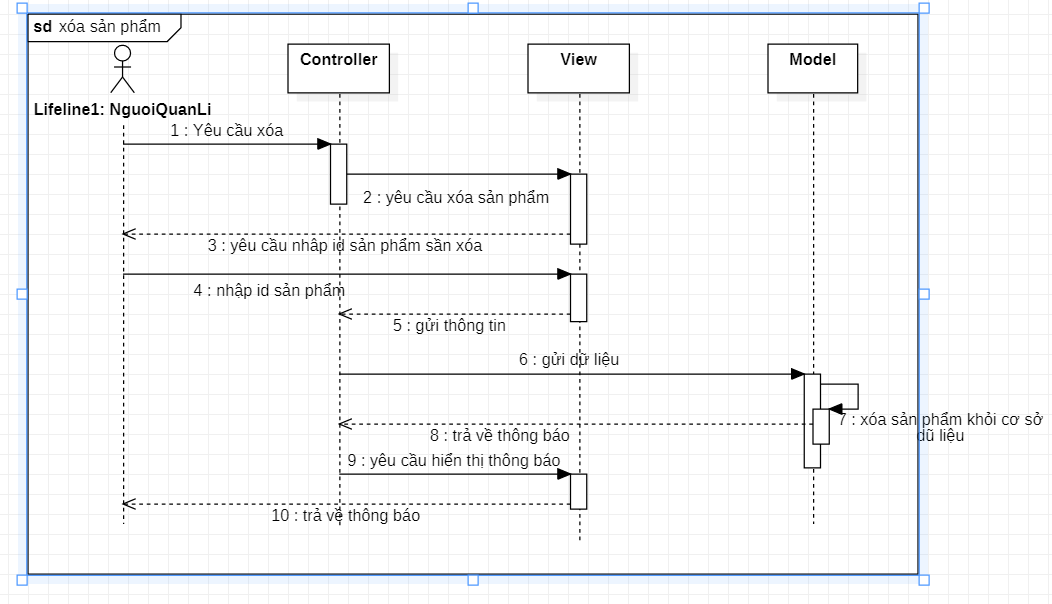
* + 1. **Biểu đồ trình tự chức năng thêm sản phẩm**



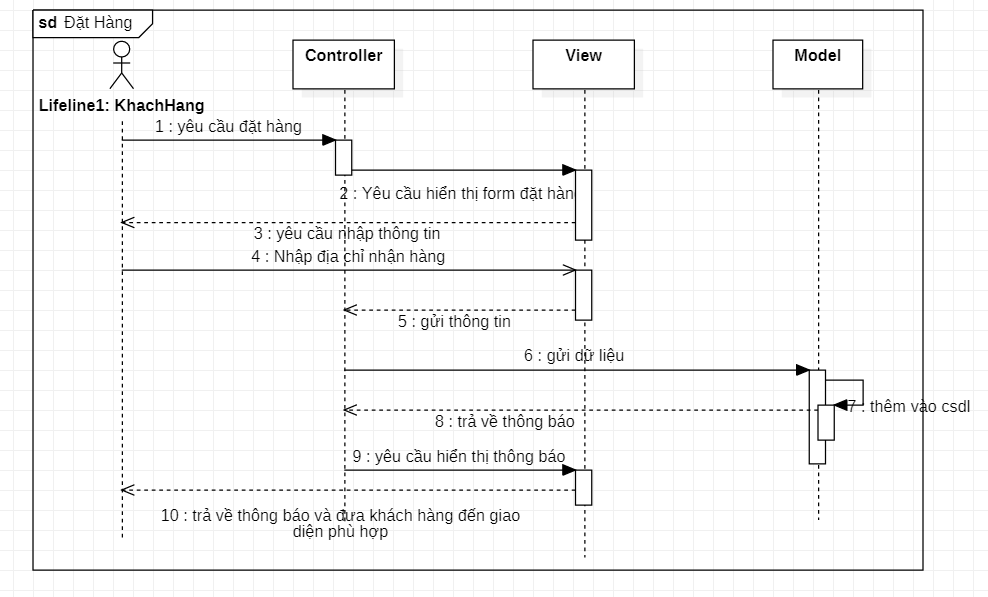
* + 1. **Biểu đồ trình tự chức năng sửa sản phẩm**



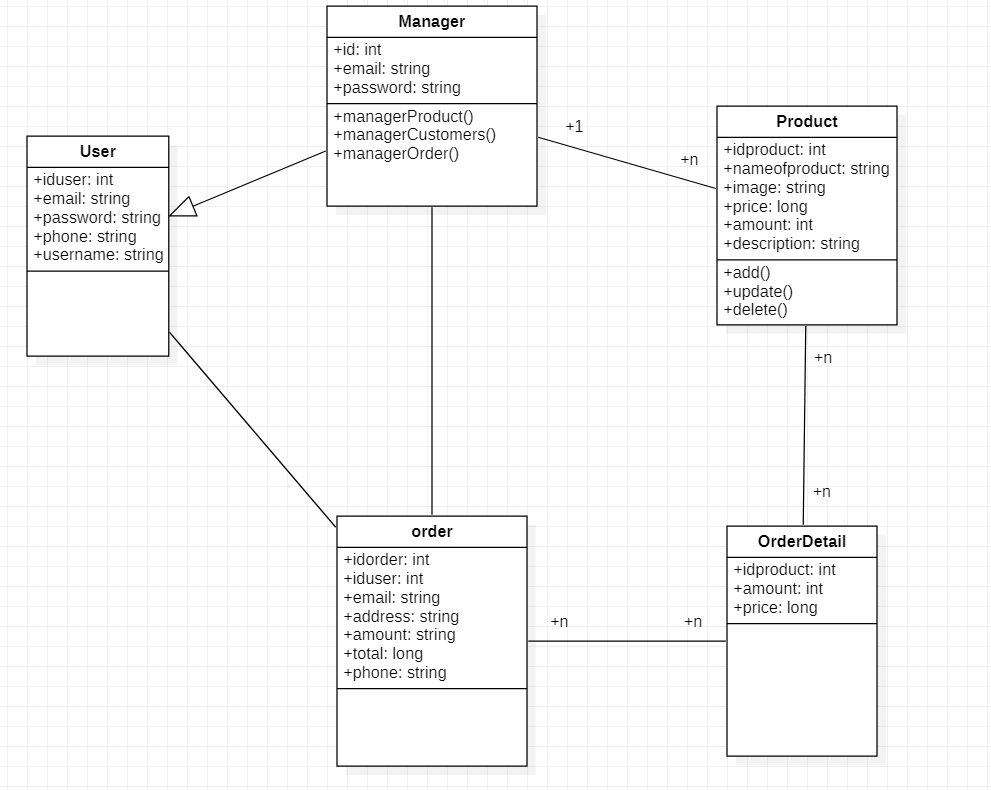
* + 1. **Biểu đồ trình tự chức năng xóa sản phẩm**



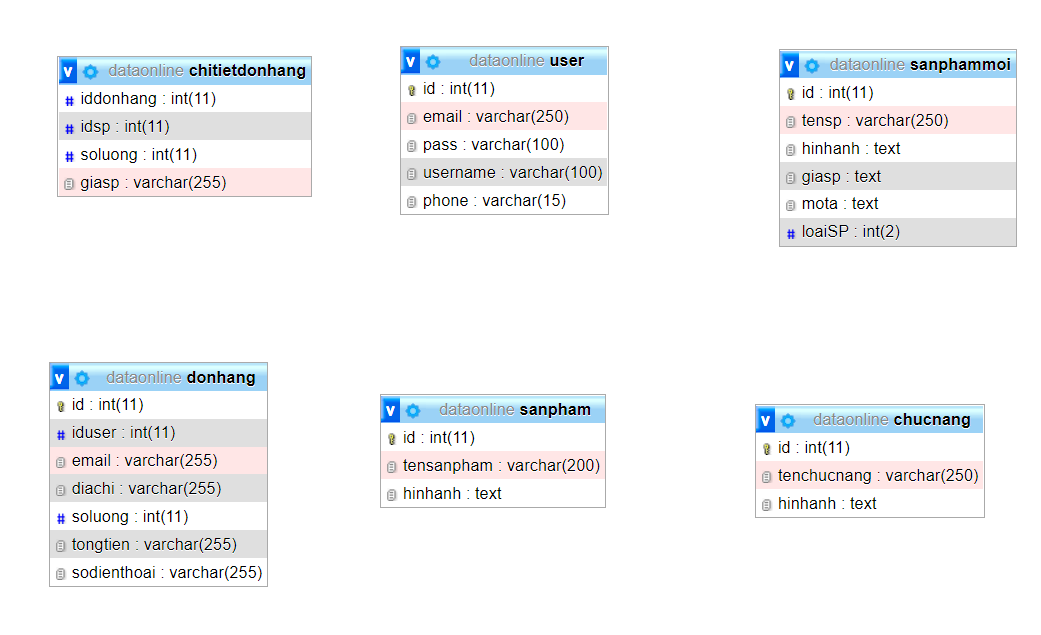
* + 1. **Biểu đồ trình tự chức năng đặt hàng**



* + 1. **Biểu đồ lớp chi tiết**



* + 1. **Cơ sở dữ liệu**



* 1. **Demo**
     1. **Giao diện người dùng**

# **KẾT LUẬN**

1. **Kết quả đạt được**
2. **Hạn chế**
3. **Hướng phát triển**
4. **Kinh nghiệm thu được**

**TÀI LIỆU THAM KHẢO**

[1]. Giáo trình lập trình ứng dụng Android - Trường Đại Học CNNT & Truyền Thông Thái Nguyên.

[2]. Giáo trình lập trình hướng đối tượng với Java - Trường Đại Học CNNT & Truyền Thông Thái Nguyên.

[3]. https://developer.android.com/studio.