

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-------------**



**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**Đề tài:** Xây dựng ứng dụng "giới thiệu bán hàng đồ gỗ Mỹ nghệ " trên android

**Giáo viên hướng dẫn :** Nguyễn Văn Tỉnh

**Người thực hiện :** Cao Văn Đương

Dương Văn Sơn

Hà Nội\_2018

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-------------**



**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**Đề tài:** Xây dựng ứng dụng "giới thiệu bán hàng đồ gỗ Mỹ nghệ " trên android

**Giáo viên hướng dẫn :** Nguyễn Văn Tỉnh

**Người thực hiện :** Cao Văn Đương

Dương Văn Sơn

Hà Nội\_2018

Mục Lục

[LỜI CẢM ƠN 4](#_Toc1807557)

[MỞ ĐẦU 5](#_Toc1807558)

[1. Tên đề tài 5](#_Toc1807559)

[2. Lý do chọn đề tài 5](#_Toc1807560)

[3. Mục đích 5](#_Toc1807561)

[4. Bố cục thực hiện 5](#_Toc1807562)

[5. Phương pháp 6](#_Toc1807563)

[6. Kế hoạch triển khai 6](#_Toc1807564)

[CHƯƠNG 1. TÌM HIỂU VỀ ANDROID, ANDROID STUDIO 7](#_Toc1807565)

[1.1.Giới thiệu về hệ điều hành android 7](#_Toc1807566)

[*1.1.1.Tổng quan về hệ điều hành Android* 7](#_Toc1807567)

[*1.1.2. Lịch sử phát triển của hệ điều hành android.* 7](#_Toc1807568)

[*1.1.3. Các phiên bản của hệ điều hành android.* 7](#_Toc1807569)

[*1.1.4.Kiến trúc cơ bản của hệ điều hành Android.* 8](#_Toc1807570)

[1.2. Công cụ xây dựng Android. 11](#_Toc1807571)

[*1.2.1. Giới thiệu công cụ xây dựng ứng dụng Android.* 11](#_Toc1807572)

[*1.2.2.Lập trình Android với Android Studio.* 11](#_Toc1807573)

[*1.2.3. Cách test ứng dụng Android* 14](#_Toc1807574)

[1.2.4. Mô hình MVP trong lập trình Android 16](#_Toc1807575)

[1.2.5. Các thành phần trong android 18](#_Toc1807576)

[1.2.6 Hướng dẫn cài đặt Android Studio 20](#_Toc1807577)

[CHƯƠNG 2. KHẢO SÁT, PHÂN TÍCH, THIẾT KẾ HỆ THỐNG 31](#_Toc1807578)

[2.1 Khảo sát hệ thống 31](#_Toc1807579)

[2.1.1 Mô tả hiện trạng 31](#_Toc1807580)

[2.1.2.Đánh giá hiện trạng 31](#_Toc1807581)

[*2.1.3*. Xác lập dự án mới 31](#_Toc1807582)

[2.2. Phân tích hệ thống 33](#_Toc1807583)

[2.2.1. Xây dựng mô hình use – case 33](#_Toc1807584)

[2.2.2. Mô hình lớp 43](#_Toc1807585)

[2.2.3. Hiện thực hóa các use case 44](#_Toc1807586)

[2.3. Thiết kế 50](#_Toc1807587)

[2.3.1. Thiết kế cơ sở dữ liệu 50](#_Toc1807588)

[*2.3.2. Thiết kế giao diện* 52](#_Toc1807589)

[CHƯƠNG 3. XÂY DỰNG VÀ CÀI ĐẶT PHẦN MỀM 54](#_Toc1807590)

[3.1 Xây dựng giao diện 54](#_Toc1807591)

# LỜI CẢM ƠN

Sau thời gian tìm hiểu đề tài “Xây dựng ứng dụng "giới thiệu bán hàng đồ gỗ Mỹ nghệ " trên android”, em đã hoàn thành đúng theo tiến độ dự kiến. Để đạt được kết quả này, em đã nỗ lực thực hiện và đồng thời nhận được nhiều sự giúp đỡ, quan tâm, ủng hộ của các thầy cô bạn bè và gia đình.

Em xin chân thành cảm ơn giáo viên hướng dẫn: thầy Nguyễn Văn Tỉnh – Bộ môn Cơ sở dữ liệu đa phương tiện – trường Đại học Công nghiệp Hà Nội đã tận tình giúp đỡ em hoàn thành thực tập chuyên ngành này.

Em xin chân thành cảm ơn các thầy cô giáo và ban lãnh đạo khoa Công nghệ thông tin – trường Đại học Công nghiệp Hà Nội đã trang bị cho em những kiến thức chuyên ngành rất hữu ích để em hoàn thành đề tài và phục vụ công việc của em sau này.

Sinh viên thực hiện.

***Cao Văn Đương***

***Dương Văn Sơn***

**MỞ ĐẦU**

1. **Tên đề tài**

Xây dựng ứng dụng "giới thiệu bán hàng đồ gỗ Mỹ nghệ " trên android

1. **Lý do chọn đề tài**

Trong kỷ nguyên công nghệ ngày này, smartphone không chỉ ngày càng phổ

biến mà còn mang đến cho con người hàng loạt khả năng mới trên mọi lĩnh vực như: trao đổi thông tin, làm việc di động, giải trí mọi lúc mọi nơi…

Điện thoại thực sự đã giúp thay đổi toàn diện cuộc sống theo hướng tích cực hơn.

Android là một hệ điều hành mã nguồn mở và phổ biến trong thế giới hiện nay và là một hệ điều hành chạy phổ biến trên các thiết bị di động.

Hiện nay việc quảng bá các sản phẩm đồ gỗ Mỹ nghệ đến người tiêu dung còn hạn chế, vì vậy con người sẽ khó mà tiếp cận các sản phẩm đồ gỗ đẹp, phong phú được vì vậy để hỗ trợ mọi người tiếp cận sản phẩm tốt, giá cả hợp lý là một điều rất cần thiết. Đây cũng chính là lý do mà em chọn đề tài này.

1. **Mục đích**

Tìm hiểu được hệ điều hành Android và cách lập trình một ứng dụng chạy trên nền tảng hệ điều hành Android.

Viết ra được ứng dụng giải quyết bài toàn quảng bá sản phẩm tới ngưới tiêu dung.

1. **Bố cục thực hiện**

*Nội dung đề tài được trình bày trong 4 chương:*

Chương 1: Tìm hiểu và công cụ Android Studio

Trình bày các thực sử dụng và hoạt động của phần mềm hỗ trợ Android Studio

Chương 2: Khảo sát, phân tích, thiết kế hệ thống

Phân tích bài toán đề ra, sau đó đưa ra mô hình dữ liệu hoạt động cho hệ thống

Chương 3: Xây dựng và cài đặt phần mềm

Trình bày quá trình xây dựng lên phần mềm

Chương 4: Kết quả thực nghiệm và đánh giá

Trình bày công việc đã hoàn thành và tóm tắt nhưng nhận định chính về phần mềm.

1. **Phương pháp**

* Đọc hiểu tài liệu
* Tìm kiếm thông tin trên mạng
* Cài đặt và cấu hình Android studio trên máy tính
* Xây dựng ứng dụng

1. **Kế hoạch triển khai**

|  |  |  |  |
| --- | --- | --- | --- |
| **Công việc** | | **Thời gian bắt đầu** | **Thời gian kết thúc** |
| Tìm hiểu và công cụ Android Studio | | **24/12** | **07/01** |
| Phân tích cơ sở dữ liệu | | **08/01** | **22/01** |
| Xây dựng ứng dụng phần mềm | **Design** | **23/01** | **30/01** |
| **Implement** | **31/01** | **28/02** |
| **Operate** | **01/03** | **03/03** |

## 

## CHƯƠNG 1. TÌM HIỂU VỀ ANDROID, ANDROID STUDIO

### 1.1.Giới thiệu về hệ điều hành android

### *1.1.1.Tổng quan về hệ điều hành Android*

Android hay hệ điều hành android là một hệ điều hành có mã nguồn mở, được phát triển bằng ngôn ngữ Linux. Hệ điều hành Android chỉ có trên thiết bị điện thoại cảm ứng, máy tính bảng, máy chơi game, smart TV…

Hiện nay, Android là hệ điều hành phổ biến nhất thế giới, chiếm 87.7% thị phần điện thoại thông minh trên toàn thế giới.

### *1.1.2. Lịch sử phát triển của hệ điều hành android.*

Ban đầu, Android được phát triển bởi tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính google mua lại vào năm 2005. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại di động đầu tiên chạy androidđược bán năm 2008.

### *1.1.3. Các phiên bản của hệ điều hành android.*

*Bảng 1: Các phiên bản của hệ điều hành Android*

|  |  |  |
| --- | --- | --- |
| 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 | 22/8/2018 (phát hành lần đầu) |
| Android 8.0 – 8.1 | Oreo | 21/8/2017 (phát hành lần đầu) |

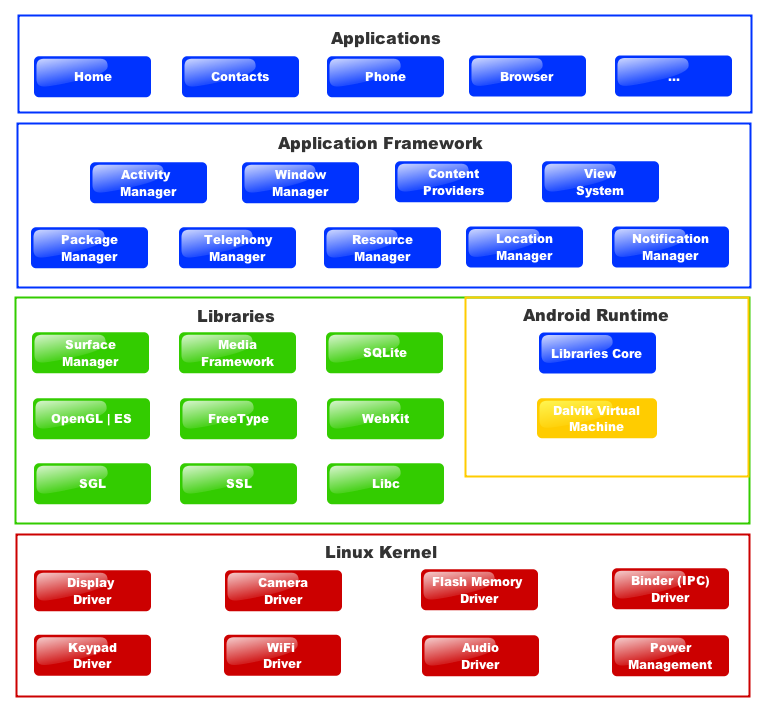
Phiên bản Android mới nhất, Android 8.1 Oreo

Bản phát hành đầu tiên của Android Oreo 8.0 là vào 21/8/2017, sau đó là Android 8.1 vào 5/12/2017

Oreo có một số cập nhật mới, nhiều bản cải tiến so với bản tiền nhiệm như sidaload (cài ứng dụng không có trên Store), giới hạn dữ liệu của ứng dụng nền, thông báo cho các ứng dụng chạy ở chế độ nền, Picture-in-piture, tự động điền mật khẩu trong ứng dụng, cải thiện thời lượng pin, Project Treble…

### *1.1.4.Kiến trúc cơ bản của hệ điều hành Android.*

Hệ điều hành Android là 1 ngăn xếp các thành phần phần mềm, được chia thành 5 phần và 4 lớp chính như trong hình bên dưới.



*Hình 1: Kiến trúc cơ bản của hệ điề hành Android*

**1.1.4.1. Linux Kernel**

Dưới cùng là lớp Linux - Linux 3.6 cùng với khoảng 115 bản vá. Lớp này cung cấp 1 cấp độ trừu tượng giữa phần cứng của thiết bị và các thành trình điều khiển phần cứng thiết yếu như máy ảnh, bàn phím, màn hình hiển thị... Đồng thời, hạt nhân (kernel) còn xử lý tất cả các thứ mà Linux có thể làm tốt như mạng kết nối và 1 chuỗi các trình điều khiển thiết bị, giúp cho giao tiếp với các thiết bị ngoại vi dễ dàng hơn.

**1.1.4.2. Các thư viện**

Ở trên lớp nhân Linux là tập các thư viện bao gồm WebKit - trình duyệt Web mã nguồn mở, được biết đến như thư viện lib, cơ sở dữ liệu SQLite - hữu dụng cho việc lưu trữ và chia sẻ dữ liệu ứng dụng, các thư viện chơi và ghi âm audio, video, hay các thư viện SSL chịu trách nhiệm bảo mật Internet...

**1.1.4.3. Các thư viện Android**

Đây là các thư viện dựa trên Java phục vụ cho việc phát triển Android. Ví dụ của các thư viện này bao gồm các thư viện ứng dụng dùng để xây dựng giao diện người dùng, vẽ đồ họa hay truy cập cơ sở dữ liệu. Một số thư viện chính của Android:

android.app - Cung cấp quyền truy cập vào ứng dụng và là nền tảng của tất cả ứng dụng Android.

android.content - Cung cấp quyền truy cập nội dung (content), truyền tải thông điệp giữa các ứng dụng hay các thành phần của ứng dụng.

android.database – Được sử dụng để truy cập dữ liệu của content provider và cơ sở dữ liệu SQLite.

android.opengl - giao diện các phương thức Java để sử dụng OpenGL ES

android.os - Cung cấp các ứng dụng với quyền truy cập vào các dịch vụ của hệ điều hành bao gồm thông điệp, các dịch vụ hệ thống và các giao tiếp nội tại.

android.text - Được sử dụng để hiển thị và điều chỉnh chữ trên màn hình thiết bị.

android.view - Các thành phần cơ bản trong việc xây dựng giao diện người dùng của ứng dụng.

android.widget - Tập các thành phần giao diện người dùng đã được xây dựng sẵn như các nút, các nhãn (label), list view,....

android.webkit - Tập các lớp cho phép xây dựng khả năng duyệt web.

**1.1.4.4. Android Runtime**

Đây là phần thứ 3 của kiến trúc và nằm ở lớp thứ 2 từ dưới lên. Phần này cung cấp một bộ phận quan trọng là Dalvik Vitual Machine - là 1 loại Java Virtual Machine được thiết kế đặc biệt để tối ưu cho Android.

Dalvik VM sử dụng các đặc trưng của nhân Linux như quản lý bộ nhớ và đa luồng, những thứ mà đã có sẵn trong Java. Dalvik VM giúp mọi ứng dụng Android chạy trong tiến trình riêng của nó, với các thể hiện (instance) riêng của Dalvik virtual Machine.

Android Runtime cũng cung cấp 1 tập các thư viện chính giúp các nhà phát triển ứng dụng Android có thể viết ứng dụng Android bằng Java.

**1.1.4.5. Application Framework**

Lớp Android Framework cung cấp các dịch vụ cấp độ cao hơn cho các ứng dụng dưới dạng các lớp Java. Các nhà phát triển ứng dụng được phép sử dụng các dịch vụ này trong ứng dụng của họ.

Android Framework bao gồm các dịch vụ chính sau:

Activitty Manager - Kiểm soát tất cả khía cạnh của vòng đời ứng dụng và ngăn xếp các Activity.

Content Providers - Cho phép các ứng dụng chia sẽ dữ liệu với các ứng dụng khác.

Resource Manager - Cung cấp quyền truy cập vào các tài nguyên như các chuỗi, màu sắc, các layout giao diện người dùng...

Notifications Manager - Cho phép các ứng dụng hiển thị cảnh báo và các thông báo cho người dùng.

View System - Tập các thành phần giao diện (view) được sử dụng để tạo giao diện người dùng.

**1.1.4.6. Application**

Lớp trên cùng của kiến trúc là Application. Các ứng dụng bạn tạo ra sẽ được cài đặt trên lớp này. Ví dụ như: Danh bạ, nhắn tin, trò chơi...

### 1.2. Công cụ xây dựng Android.

### *1.2.1. Giới thiệu công cụ xây dựng ứng dụng Android.*

Hệ điều hành Android đặc biệt là có nhiều API hữu ích cho các nhà phát triển phần mềm, nhà phát triển ROM có thể phát triển nên nhiều ứng dụng, hệ thống khác nhau. Mặt khác Android còn là hệ điều hành mã nguồn mở, nên hoàn toàn miễn phí nên các nhà phát triển ứng dụng trên android mà không phải lo lắng về vấn đề bản quyền(những điều khoản ràng buộc chặt chẽ, hạn chế một số quyền, thủ tục đăng ký phức tạp,…) như trong một số hệ điều hành khác như là IOS của Apple, Symbian của Nokia,…Ngoài ra việc hệ điều hành Android dựa trên nền tảng Linux và được sử dụng ngày càng phổ biến( theo thống kê thì Android đang là hệ điều hành được ưa chuộng nhất tại Mỹ trên các thiết bị di động) đồng nghĩa với việc giới lập trình viên có thể tin tưởng là các ứng dụng được viết ra có thể chạy được ngay trên nhiều dòng di động của các hãng khác nhau. Họ ít phải quan tâm là đang phát triển cho loại ứng dụng di động nào, phiên bản bao nhiêu vì nền tảng Android là chung cho nhiều dòng máy, máy ảo Java đã chịu trách nhiệm thực thi những ứng dụng phù hợp với mỗi dòng di động mà nó đang chạy.

Tất cả các chương trình ứng dụng được viết bằng ngôn ngữ java kết hợp với XML nên có khả năng khả chuyển cao. Vì vậy có khá nhiều công cụ phát triển ứng dụng Android như: Eclipse, netbean, Android Studio…

### *1.2.2.Lập trình Android với Android Studio.*

**1.2.2.1. Giới thiệu tổng quát về Android Studio**

Google cung cấp cho lập trình viên 1 IDE gọi là Android Studio để phát triển ứng dụng Android và đây là một IDE được các lập trình viên ưa chuộng nhất.

Ứng dụng Android yêu cầu một số file cấu hình. Ứng dụng sẽ được viết chủ yếu bằng ngôn ngữ lập trình Java. Android Studio sẽ sử dụng Android SDK( Android software development Kit) và Gradle tooling để convert những file này thành ứng dụng Android. Nó cho phép biên dịch, đóng gói và deploy và khởi chạy ứng dụng sau khi đã build xong.

Android SDK bao gồm cả ADB (Android Debug bridge) là một công cụ cho phép ta kết nối tới các thiết bị máy ảo và máy thật để build ứng dụng. Nó cho phép ta quản lý các thiết bị và hỗ trợ ta debug ứng dụng.

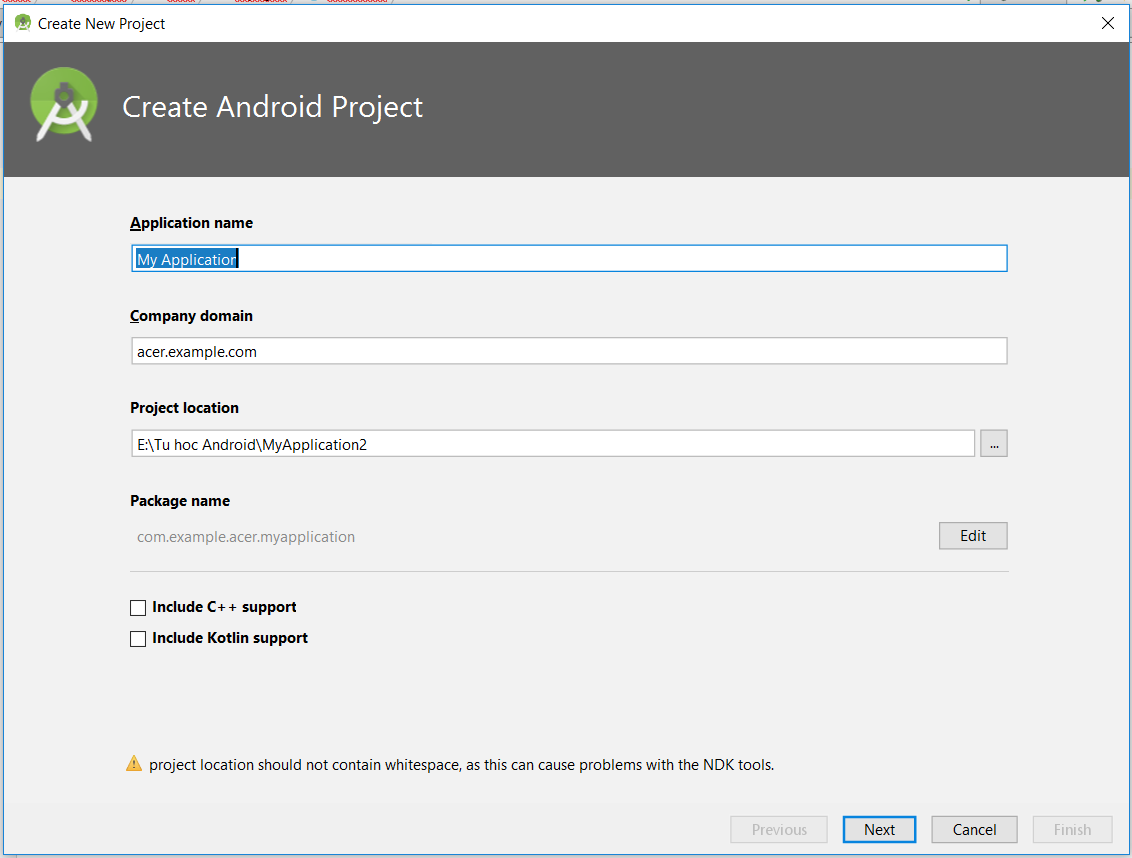
Hầu hết các file cấu hình trong Android đều dưới dạng XML. Android Studio cung cấp các trình sửa file cho từng file riêng biệt (tùy theo loại file mà android sẽ hiển thị cách mà chúng ta sửa file khác nhau).

Để mang lại hiệu quả cao nhất khi ta làm việc thì việc mà ta trang bị cấu hình càng cao càng tốt. Nhưng bao nhiêu là phù hợp, vừa đủ? Xung nhịp ít nhất cho CPU:2.5GHz với ít nhất 8GB RAM và một ổ cứng SSD để mang lại hiệu năng tốt khi ta làm việc.

**1.2.2.2. Các bước khởi tạo một dự án trên Android Studio**

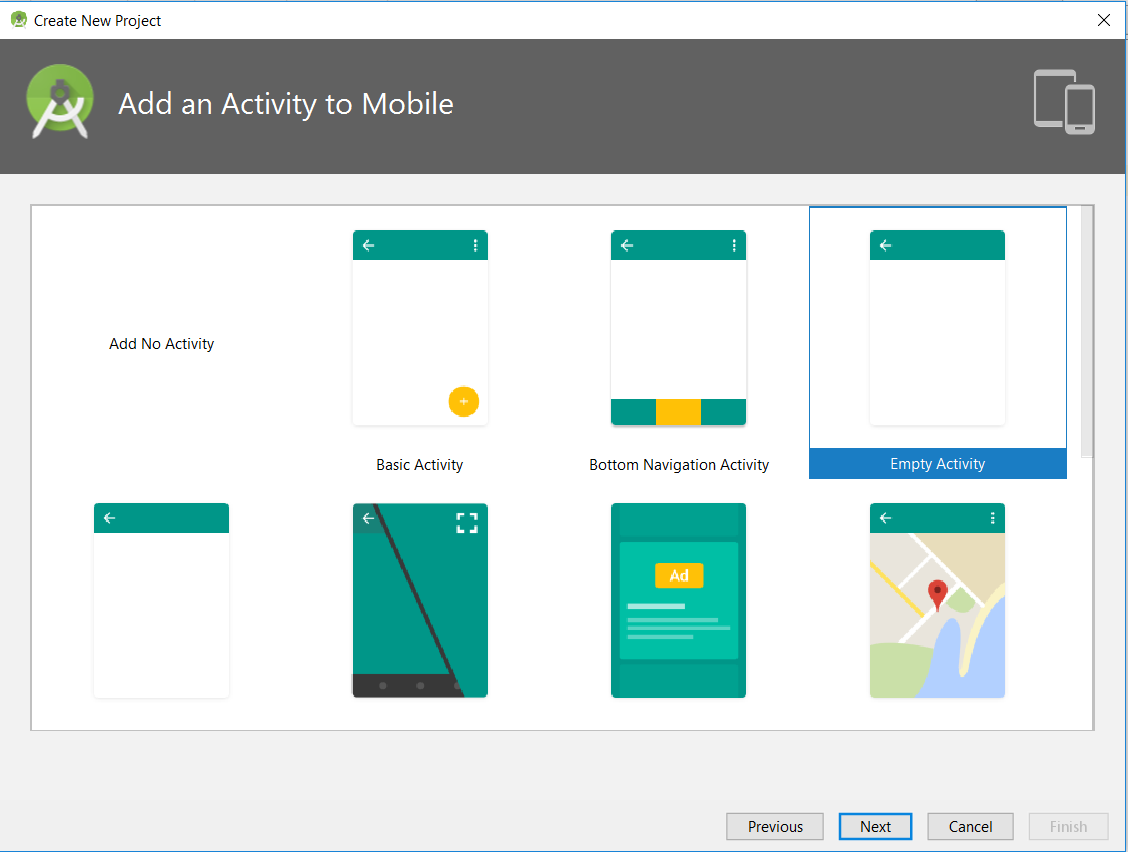
Sau khi cài đặt Android Studio xong, để khởi tạo một dự án cần thực hiện các bước sau:

Mở **Android Studio** và chọn **New Project** hoặc truy cập **File -> New -> New Project** từ IDE.



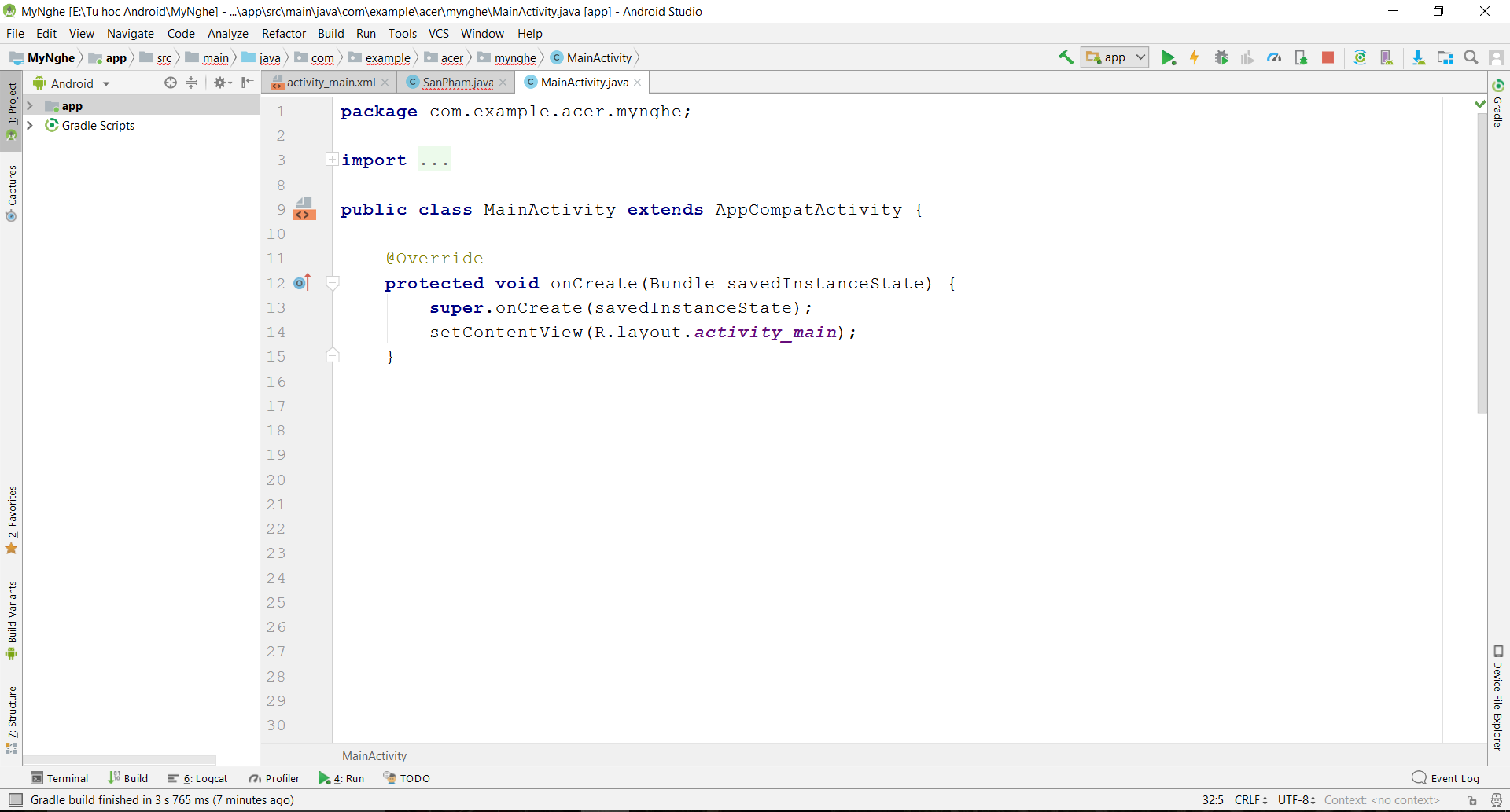
*Hình 2: Khởi tạo Android Studio*

Sau đó, chọn loại hoạt động (Activity) bạn muốn. Activity là “màn hình” thực sự của một ứng dụng. Trong một số trường hợp, nó sẽ là toàn bộ ứng dụng hoặc nằm trong những thành phần khác. Ứng dụng của bạn có thể chuyển từ màn hình này sang màn hình bên cạnh. Bạn thoải mái bắt đầu dự án mới mà không cần activity (chọn**Add No Activity**) nhưng chắc chắn bạn sẽ luôn muốn nó vì dễ dàng bắt đầu công việc hơn với một mẫu ứng dụng trống.



*Hình 3: Hình ảnh chọn Activity trong Android Studio*

Sau khi chọn Activity xong, ta sẽ có một màn hình để thực hiện việc viết code.



*Hình 4:Hình ảnh project Android sau khi khởi tạo*

Chúng ta đều biết ứng dụng Android được hợp thành bởi nhiều file khác nhau. Nhiệm vụ của Android Studio là lưu giữ tất cả chúng ở cùng một vị trí. Cửa sổ chính bên phải màn hình hiển thị từng scrip và file, các tab dọc theo phía trên cho phép chuyển đổi cửa sổ đang mở.

Nếu muốn mở hoạt động mới, hãy di chuyển tới khu vực phân cấp file bên trái. Tại đây, bạn sẽ thấy tất cả thư mục và các thư mục bên trong chúng. File Java nằm trong java, tiếp đến là tên gói ứng dụng. Nhấp đúp vào **MainActivity.Java**, nó sẽ hiện ở phía trên cửa sổ bên phải.

### *1.2.3. Cách test ứng dụng Android*

**1.2.3.1. Sử dụng máy ảo Genymotion**

Sử dụng máy ảo để lập trình các ứng dụng Android là một trong những vấn đề được các lập trình viên khá quan tâm và lo ngại từ trước tới nay. Trong khi sự cải tiến máy ảo Android của Google vẫn chưa thực sự thuyết phục thì Genymotion đã ra đời để giải quyết bài toán nan giải này cho các lập trình viên Android.

Có lẽ nhiều bạn đã khá quen thuộc với cái tên Genymotion rồi, vì hiện nay nó đang là loại máy ảo dành cho Android được sử dụng rất phổ biến hiện nay vì nhiều lí do:

Tốc độ khởi động hay tắt đi rất rất nhanh.

Máy ảo hỗ trợ khá nhiều chức năng với các công cụ tuỳ biến dễ sử dụng và được tích hợp trực tiếp trên máy ảo.

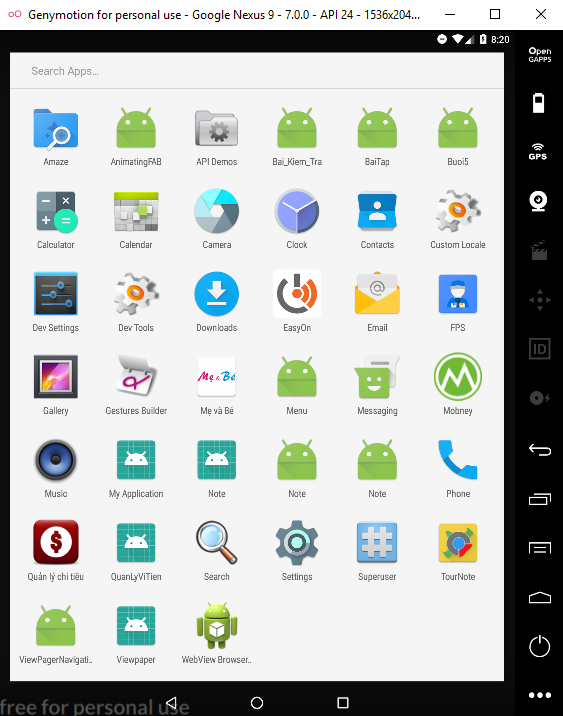
Kích thước máy ảo của Genymotion nhỏ hơn rất nhiều so với máy ảo Android Emulator của Google.

Bộ tài liệu hướng dẫn khá đầy đủ, chi tiết và có ví dụ minh hoạ.

Hỗ trợ người sử dụng rất nhanh và nhiệt tình.

Tuy nhiên, có một điều hơi bất cập một chút là ở việc trả phí. Ở mức độ sử dụng miễn phí thì cơ bản dùng được một vài chức năng liên quan tới phần cứng, ví dụ như camera...còn nếu muốn sử dụng các chức năng phức tạp hơn như nhắn tin, gọi điện, GPS...thì các lập trình viên phải đăng kí trả phí sử dụng hàng tháng.

Nhưng nói chung về cơ bản là đủ dùng để phát triển và thử nghiệm các ứng dụng Android rồi. Với Genymotion chúng ta có thể cài đặt các loại máy ảo sử dụng hệ điều hành Android với nhiều version khác nhau để có thể test được nhiều ở các phiên bản khác nhau. Dưới đây là hình ảnh của ảo genymotion sau khi cài đặt:



*Hình 5: Máy ảo genymotion*

**1.2.3.2. Sử dụng máy thật**

Để có thể sử dụng máy thật để test các ứng dụng Android thì thiết bị di động của bạn phải có hệ điều hành Android 4.2 trở lên và thực hiện cấu hình thiết bị. Tất cả các thiết lập dành cho nhà phát triển ứng dụng sẽ được ẩn theo mặc định. Để hiển thị các cài đặt này, mở ứng dụng Settings trên thiết bị của bạn và điều hướng đến màn hình About phone. Tiếp theo, cuộn xuống đến phần Build number và bấm vào đó 7 lần.

Khi bạn làm điều này, bạn sẽ có thể để nhìn thấy tuỳ chọn Developer. Mở nó và đảm bảo rằng tuỳ chọn USB debugging đã được chọn.

Ngoài ra, tôi đề nghị bạn nên kiểm tra kỹ Strict mode enabled và Show CPU usage cũng đã được lựa chọn. Với các tùy chọn này đã được kích hoạt, sẽ cho bạn biết khi nào bạn đi chệch hướng khỏi những cách viết code được khuyến nghị.

Tại thời điểm này, điện thoại của bạn có thể được sử dụng cho việc phát triển ứng dụng. Sử dụng cáp USB của nó để kết nối nó với máy tính của bạn.

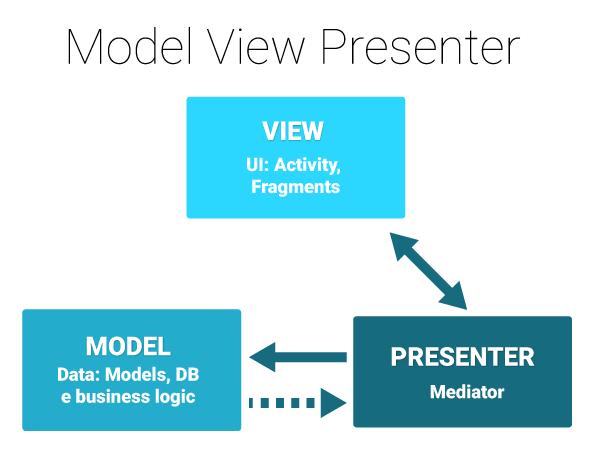
Máy tính của bạn cần cài đặt phần mềm hỗ trợ việc kết nối để có thể test code trên android Studio như Impactor…Sau đó sử dụng cáp USB kết nối giữa máy tính với điện thoại và thực hiện build ứng dụng.

### 1.2.4. Mô hình MVP trong lập trình Android

**1.2.4.1. Tìm hiểu tổng quát về mô hình MVP**

MVP là một mô hình kiến trúc hướng giao diện người dùng, 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.

Mô hình 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 6: Mô hình MVP*

Trong đó:

M-Model: Lớp xử lý dữ liệu. Lớp này sẽ chịu trách nhiệm lấy dữ liệu từ database hoặc network một cách bất đồng bộ. Sau đó sẽ trả về dữ liệu cho presenter thông qua hàm callback.

V-View: Lớp xử lý view. Lớp này chịu trách nhiệm tìm view( bindview), đưa dữ liệu vào view, animation, kiểm soát các input event của user và gửi cho present các event.

P – Presenter: Lớp này sẽ là lớp xử lý các bussiness logic. Đây sẽ là lớp trung gian có thể giao tiếp được với hai lớp M & V ( chú ý M – V không giao tiếp trực tiếp được với nhau). Khi lớp View nhận một input event sẽ gửi sự kiện xuống lớp P, lớp P sẽ lấy dữ liệu từ lớp Model và gửi lại cho lớp V và hướng dẫn lớp V cách hiển thị.

**1.2.4.2. Lợi ích của việc sử dụng mô hình MVP**

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.

Một ví dụ điển hình là CursorAdapter, đây là sự kết hợp của việc định dạng dữ liệu với xử lý tương tác giao diện. Đôi khi trong đó còn bao gồm cả xử lý dữ liệu ở mức sâu hơn (như tương tác với CSDL) thông qua Cursor.

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 lần, 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.

Trong Android, kiểm thử là một vấn đề khó khăn vì mối liên kết chặt chẽ giữa giao diện, logic và dữ liệu. MVP tách biệt giao diện khỏi dữ liệu, chia ứng dụng ra thành ít nhất ba lớp khác nhau, từ đó có thể kiểm thử một các độc lập. Với MVP ta có thể kiểm thử đo đạc một cách tối đa các logic của ứng dụng.

**1.2.4.3. Triển khai MVP trên Android**

Có rất nhiều biến thể cũng như phương pháp triển khai MVP, tất cả mọi người có thể điều chỉnh mô hình này tùy theo nhu cầu và cách họ cảm thấy thoải mái hơn. Các mô hình này, về cơ bản khác nhau ở số lượng chức năng mà tầng *Presenter* đảm nhận.

Tầng trình diễn - Presenter

Tầng trình diễn có trách nhiệm như một middle-man giữa *View* và *Model.* Nó lấy dữ liệu từ *Model*, định dạng và trả về cho *View*. Nhưng không giống như MVC, nó cũng quyết định những gì sẽ xảy ra khi người dùng tương tác với *View*, hay nói cách khác nó hàm chứa logic ứng dụng.

Tầng logic dữ liệu – Model

Trong một ứng dụng với thiết kế kiến trúc tốt, mô hình này sẽ chỉ là một *gateway* giữa tầng *domain* và tầng *business* logic. Trong mô hình *Clean Architecture* của *Uncle Bob*, *model* sẽ là một interactor thực thi một use case. Để đơn giản, ở đây model đơn thuần được nhìn nhận như một data source – cung cấp dữ liệu chúng ta muốn hiển thị một giao diện ứng dụng.

Tầng giao diện – View

View thường được *implement* bởi một Activity( hoặc có thể là một Fragment, một View … tùy thuộc vào cấu trúc ứng dụng), Activity này sẽ chứa một thuộc tính là một lớp *Presenter*. Lý tưởng nhất *Presenter* nên được cung cấp bởi một Dependency Injection framework như Dagger, nhưng trong trường hợp ứng dụng không sử dụng một thư viện hay framework như vậy ta hoàn toàn có thể tạo ra các đối tượng *Presenter* này.

### 1.2.5. Các thành phần trong android

**1.2.5.1. Các thành phần chính**

Trong Android gồm 4 thành phần chính là: Activities , Services , Broadcast Receivers và Content Providers.

* Activities điều khiển UI và xử lý tương tác người dùng trên màn hình smartphone.

Một Activity biểu diễn một màn hình với một giao diện UI. Nói đơn giản, Activity thực hiện các hành động trên màn hình. Ví dụ, một ứng dụng email có thể thực hiện một activity mà hiển thị một danh sách các email mới, activity khác để soạn thảo một email, và activity khác để đọc email. Nếu một ứng dụng có nhiều hơn một activity, thì một trong số chúng sẽ được đánh dấu như là activity mà sẽ được hiển thị khi ứng dụng chạy.

* Services xử lý các tiến trình background được gắn kết với một ứng dụng

Một service là một Component mà chạy trong Background để thực hiện các hoạt động. Ví dụ, một service có thể chơi nhạc nền trong khi người dùng là trong một ứng dụng khác, hoặc nó có thể lấy dữ liệu qua mạng mà không cần chặn tương tác người dùng với một activity.

* Broadcast Receivers xử lý giao tiếp giữa Android OS và các ứng dụng

Về cơ bản, Broadcast Receivers phản hồi các thông điệp từ ứng dụng khác hoặc từ hệ thống. Ví dụ, các ứng dụng cũng có thể khởi tạo các broadcast để báo cho các ứng dụng khác biết rằng có một số dữ liệu đã được tải về thiết bị và nó là có sẵn cho các ứng dụng này để sử dụng chúng. Vì thế, Broadcast Receiver sẽ thông dịch thông tin này và sẽ khởi tạo hành động thích hợp.

Một Broadcast Receiver được triển khai như là một lớp con của BroadcastReceiver và mỗi thông điệp được truyền như là một đối tượng Intent.

* Content Providers xử lý dữ liệu và quản lý cơ sở dữ liệu

Thành phần Content Provider cung cấp dữ liệu từ một ứng dụng tới các ứng dụng khác tùy theo yêu cầu. Những yêu cầu này được xử lý bởi các phương thức của lớp ContentResolver. Dữ liệu có thể được lưu trữ trong hệ thống file, cơ sở dữ liệu hoặc bất cứ đâu.

Một Content Provider được triển khai như là một lớp con của lớp ContentProvider và phải triển khai một APIs chuẩn mà cho phép các ứng dụng khác thực hiện các transaction.

Ngoài ra còn có một số thành phần phụ là Fragments, Views, Layouts, Intents,

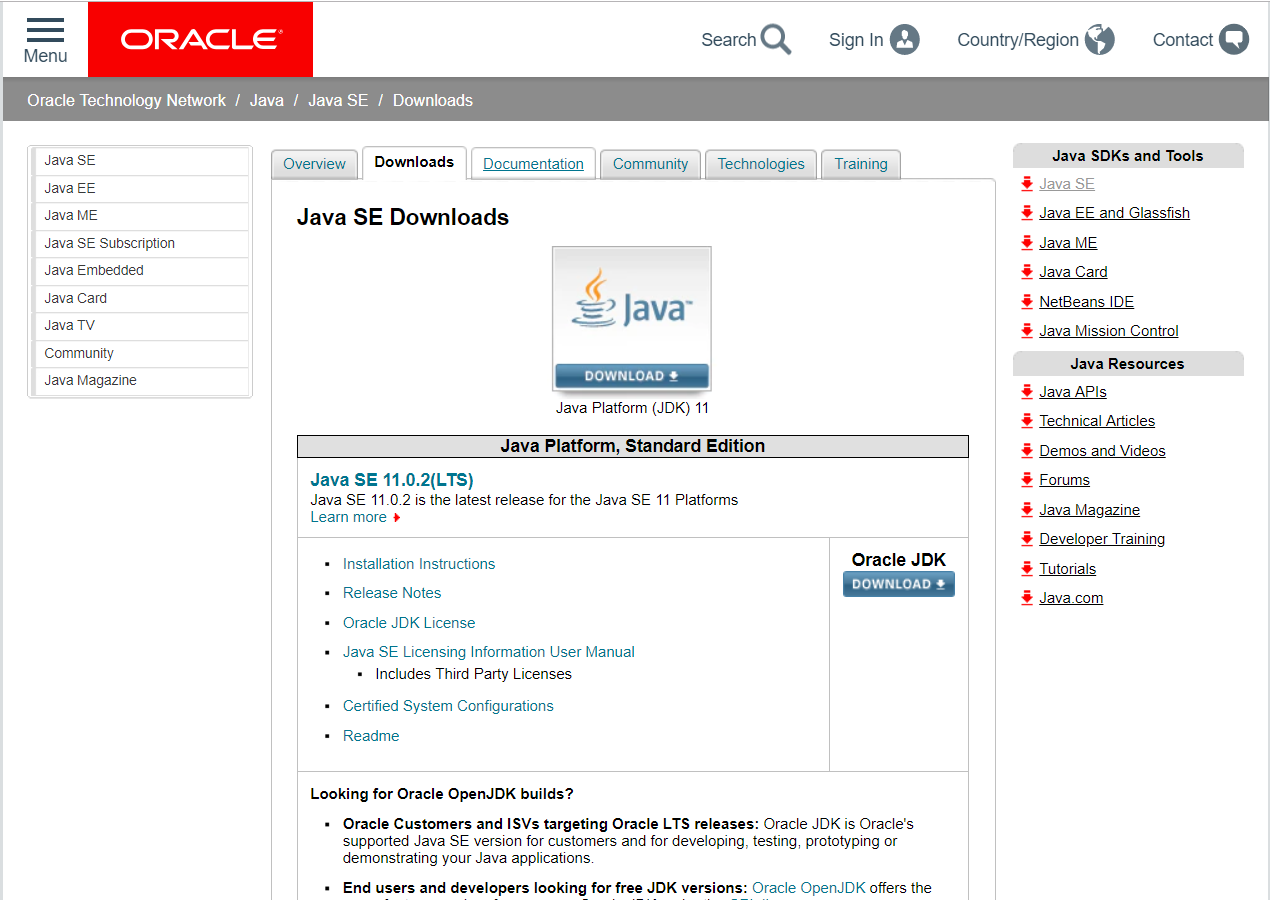
Resources và Manifest.

* + Fragments biểu diễn một phần của giao diện UI trong một Activity
  + Views là các phần tử giao diện UI mà được vẽ trên màn hình bao gồm button, list, form, …
  + Layouts thể hiện cấu trúc thứ bậc của view mà điều khiển định dạng màn hình và bề mặt của các view.
  + Intents các thành phần kết nối các thông báo với nhau
  + Resources các phần tử ngoại vi, như các chuỗi, hằng, và hình ảnh có thể vẽ
  + Manifest chính là Configuration file cho ứng dụng.

### 1.2.6 Hướng dẫn cài đặt Android Studio

Bước 1: Tải cài cấu hình môi trường JDK

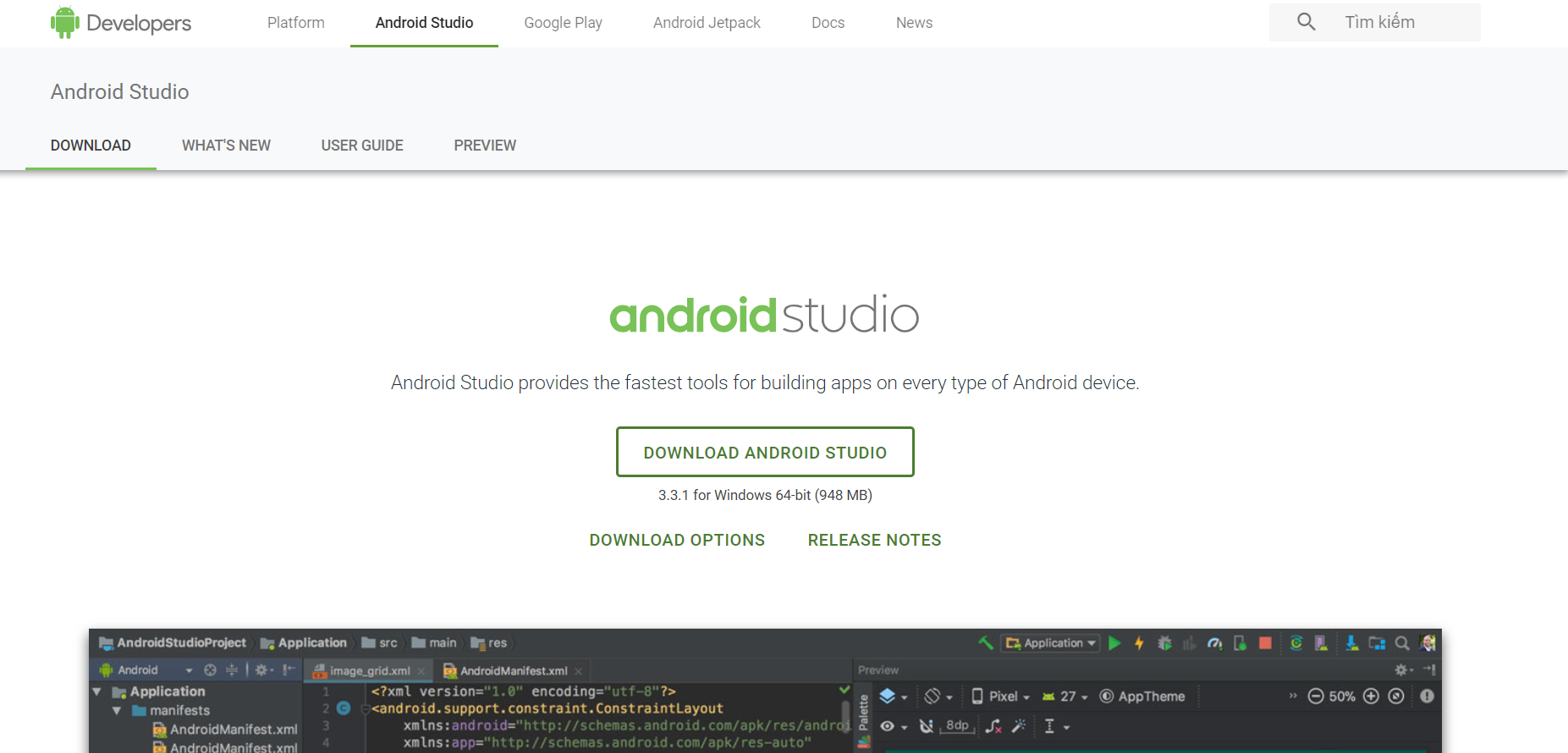
[*https://www.oracle.com/technetwork/java/javase/downloads/index.html*](https://www.oracle.com/technetwork/java/javase/downloads/index.html)



*Hình 7 Tải JDK*

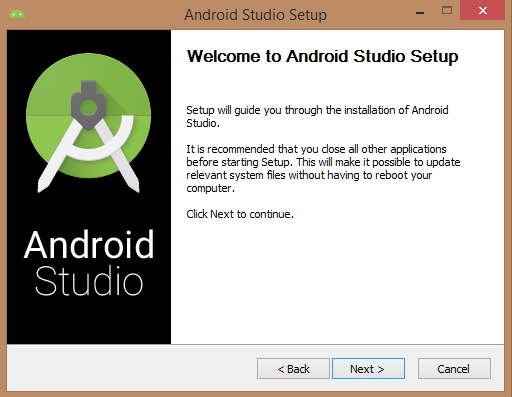
*Sau đó chạy và cấu hình JDK mặc định.*

Bước 2*: Tải và cấu hình Android Studio* [*https://developer.android.com/studio/*](https://developer.android.com/studio/)

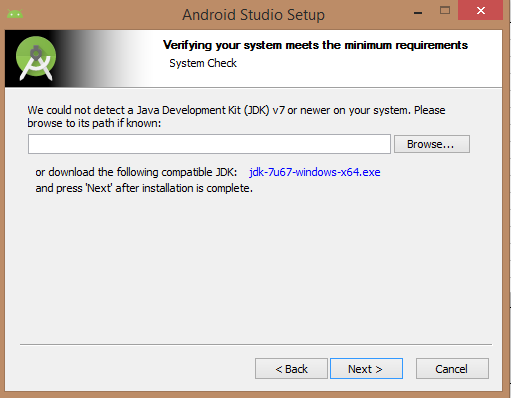


*Hình 8 Tải Android studio*

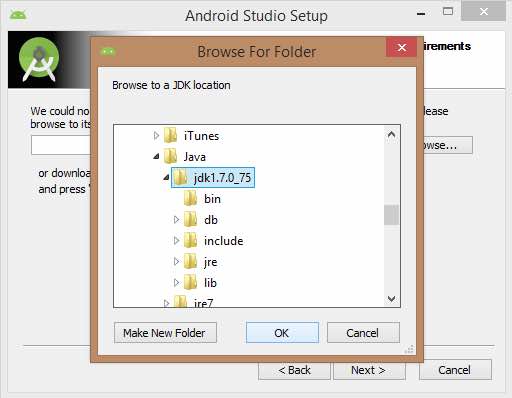
Chạy AndroidStudio.exe, đảm bảo trước khi chạy Android Studio, thiết bị cần thiết Java JDK đã được cài đặt.



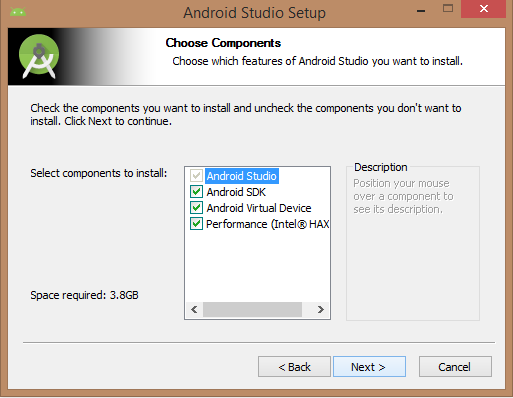
Khi bạn đã chạy Android Studio, lúc này, điều cần nói đến là JDK5 Path hoặc phiên bản trong Installer.



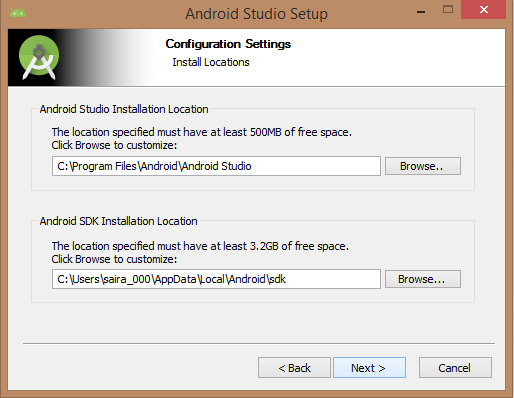
Hình ảnh dưới khởi tạo JDK cho Android SDK.



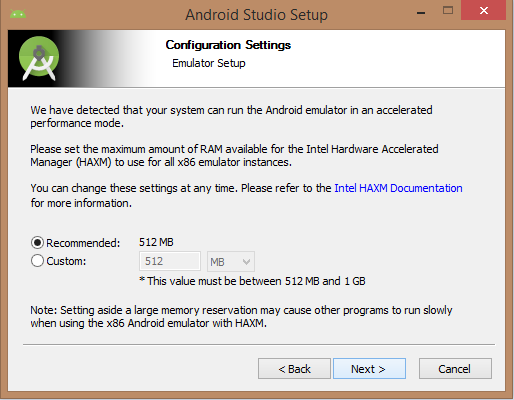
Bạn cần kiểm tra các thành phần, mà là cần thiết để tạo các ứng dụng, hình ảnh dưới đã chọn:



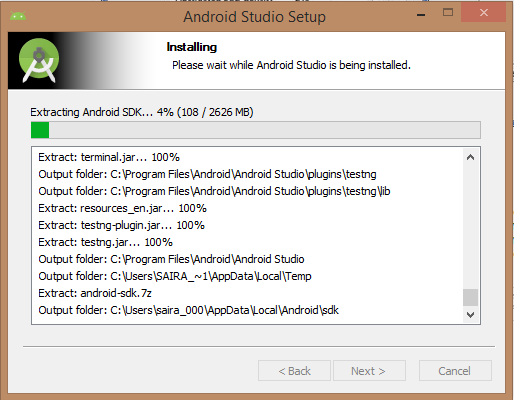
Bạn cần xác định vị trí của local path cho Android Studio và Android SDK. Ảnh dưới lấy vị trí mặc định của Windows 8.1 x64 bit.



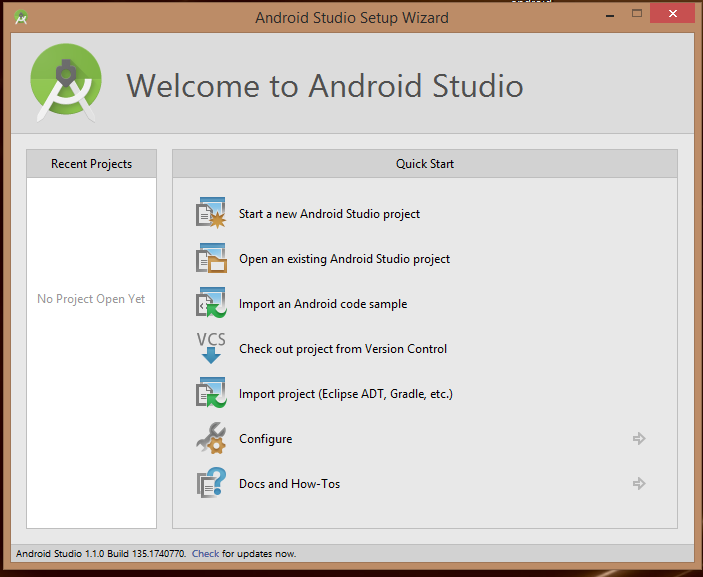
Tiếp đó, bạn xác định RAM space cho Android Emulator, theo mặc định nó sẽ là 512 MB.



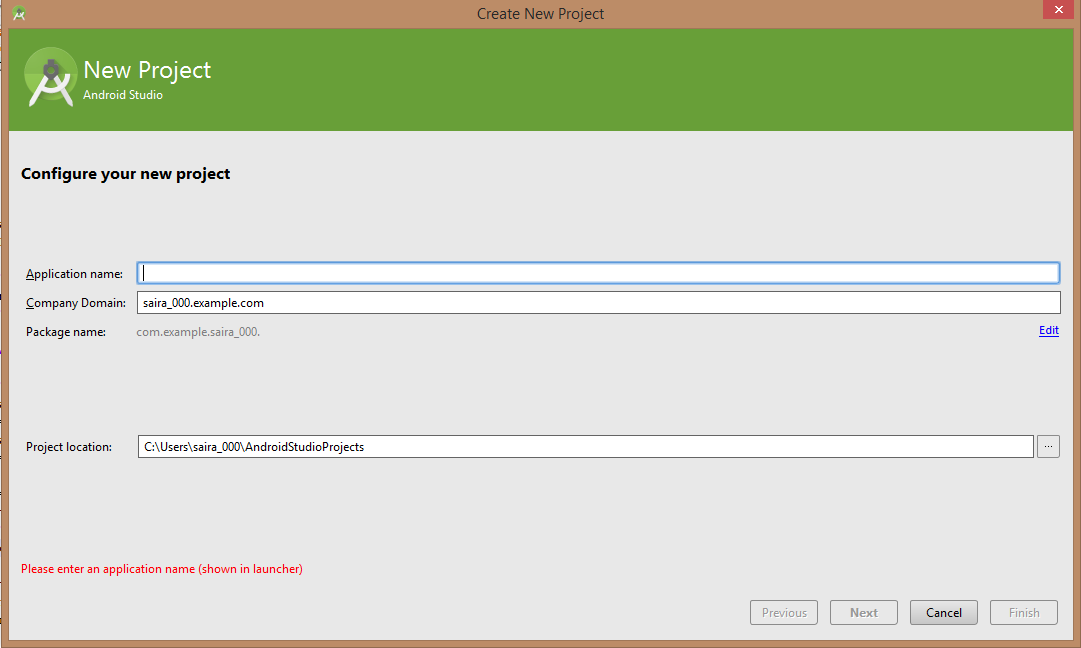
Cuối cùng, nó sẽ extract SDK package vào trong local machine. Bạn sẽ cần đợi trong chốc lát để hoàn thành tác vụ này, và nó sẽ nhận 2626 MB của Hark Disk.



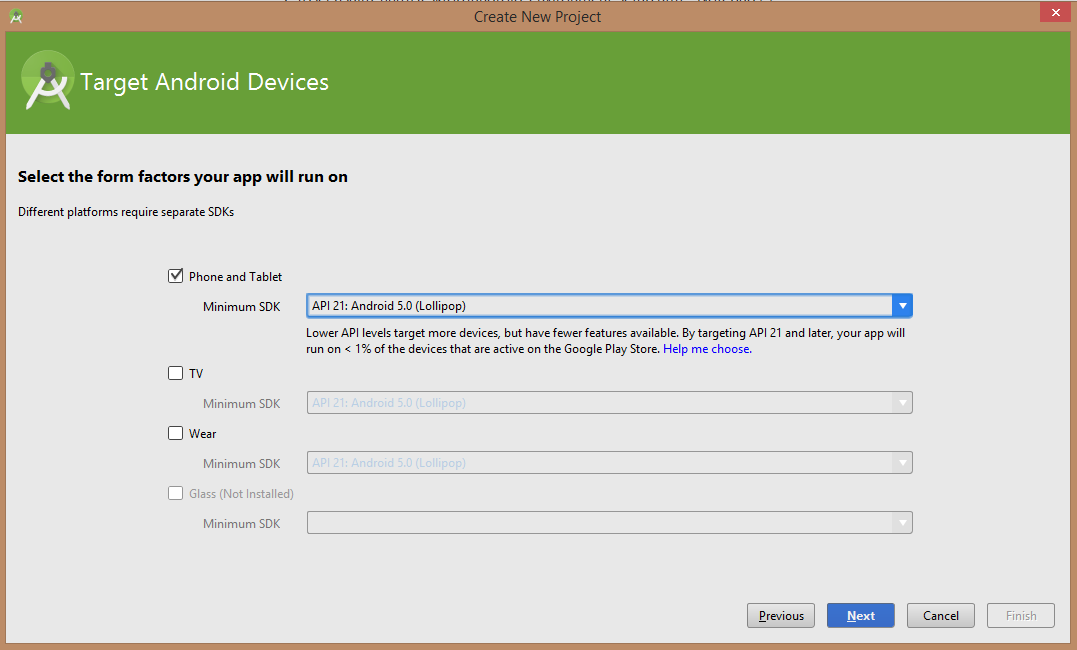
Sau khi thực hiện tất cả bước trên, bạn phải nhất nút Finish và nó sẽ mở Android Studio với thông báo Welcome to Android Studio, như hình dưới.



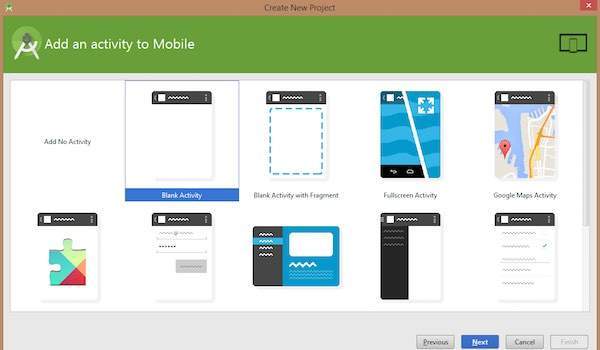
Bây giờ, bạn có thể bắt đầu phát triển ứng dụng của mình với Project mới. Trong khung cài đặt, nó sẽ hỏi Application name, thông tin về Package và vị trí của Project.



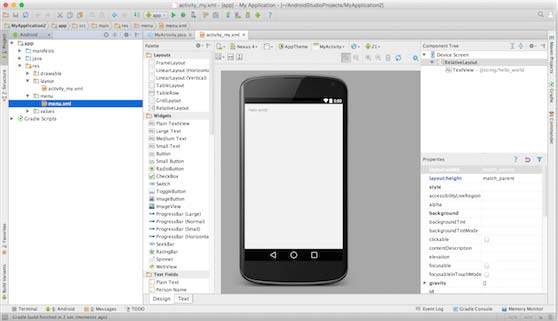
Sau khi nhập tên ứng dụng, bạn cần lựa chọn factor mà ứng dụng của bạn sẽ chạy trên đó. Ở đây, cần thiết để xác định Minimum SDK, trong bài này tôi đã khai báo là API21: Android 5.0 (Lollipop).



Trong bước tiếp theo, bạn sẽ chọn một activity to Mobile, nó sẽ xác định layout mặc định cho các ứng dụng.

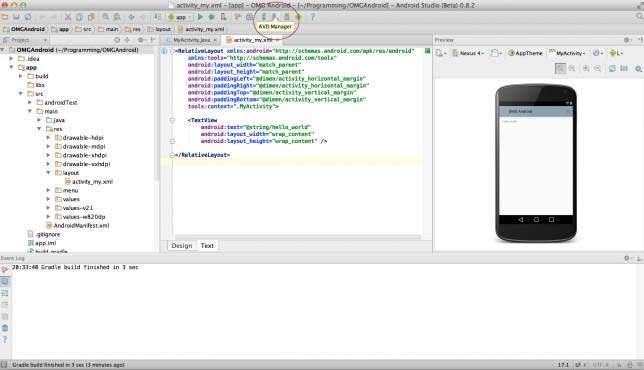


Cuối cùng, nó sẽ mở Development Tool để viết code cho ứng dụng:



Bước 3: Tạo Android Virtual Device

Để kiểm tra các ứng dụng Android của mình, bạn sẽ cần một thiết bị Android ảo. Do đó, trước khi bắt đầu viết code, chúng ta tạo một Android Virtual Device. Chạy Android AVD Manager bằng việc nhấn chuột vào biểu tượng AVD\_Manager, như dưới:



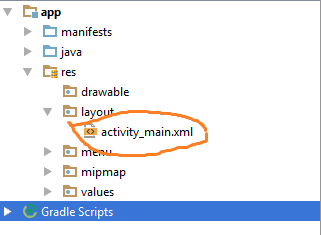
Nó sẽ hiển thị các thiết bị ảo mặc định có mặt trên SDK của bạn, hoặc bạn có thể cần tạo một thiết bị ảo bằng việc nhấn chuột vào nút Create new Virtual device.



Nếu AVD được tạo thành công, nghĩa là môi trường của bạn đã sẵn sàng để phát triển ứng dụng Android. Nếu bạn thích, bạn có thể đóng cửa sổ này bởi sử dụng nút X ở trên cùng bên phải. Dưới đây chúng ta sẽ xem một số khái niệm quan trọng liên quan tới phát triển ứng dụng Android.

Tạo ứng dụng Hello Word trong Android

Trước khi viết code để hiển thị Hello World, bạn phải biết về các thẻ trong XML. Để viết chương trình này, bạn nên redirec tới **App>res>layout>Activity\_main.xml**



Để hiển thị Hello World, chúng ta cần gọi Text View với Layout. (bạn có thể tham khảo tại [**Relative Layout**](https://vietjack.com/android/relative_layout_trong_android.jsp) và [**Text View**](https://vietjack.com/android/textview_trong_android.jsp) ).

<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:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin" tools:context=".MainActivity">

<TextView android:text="@string/hello\_world"

android:layout\_width="550dp"

android:layout\_height="wrap\_content" />

</RelativeLayout>

Để chạy chương trình bằng việc nhấn chuột vào **Run>Run App** hoặc tổ hợp phím **shift+f10**. Cuối cùng, kết quả sẽ được đặt tại Virtual Device, như sau:



## CHƯƠNG 2. KHẢO SÁT, PHÂN TÍCH, THIẾT KẾ HỆ THỐNG

### 2.1 Khảo sát hệ thống

### 2.1.1 Mô tả hiện trạng

Ứng dụng ***“Đồ gỗ Mỹ nghệ”:***

Ứng dụng là cầu nối giữa các sản phẩm mặt hàng đồ gỗ Mỹ nghệ đến với mọi người, đặc biệt là đến với những người đang có ý định tìm hiểu và mua sắm cho mình.

Người dùng sẽ tiếp cận được các sản phẩm đồ gỗ đẹp, chuẩn mẫu thiết kế, biết được giá cả, chất liệu cho sản phẩm cũng như là địa chỉ mua sản phẩm.

Trong lần đầu tiên sử dụng ứng dụng bạn có thể đăng ký tài khoản hoặc có thế biết được thông tin người bán sản phẩm để thực hiện việc trao đổi trực tiếp.

- App gồm 3 mục chính:

* Giới thiệu: Là nơi khách hàng có thể nhìn thấy tất cả sản phẩm đang có trong hệ thống.
* Tài khoản: Là nơi quản lý các thông tin của người dùng.
* Lịch sử mua hàng: Sẽ cho bạn biết rõ hơn về tình hình những sản phẩm đã mua.

### 2.1.2.Đánh giá hiện trạng

***\* Ưu điểm:***

- Dễ sử dụng và rút ngắn khoảng cách giữa sản phẩm và người mua hàng.

- Quản lý lượng sản phẩm vào ra dễ dàng.

***\*Nhược điểm:***

- Khó xác thực được thông tin khách hàng.

### *2.1.3*. Xác lập dự án mới

**2.1.3.1. Mục đích, phạm vi**

*a.Mục đích:*

- Mục đích xây dựng ứng dụng:Giúp người tiêu dùng dễ dàng tiếp cận sản phẩm ưa thích. Giải quyết bài toán tồn hàng.

- Các mục tiêu cụ thể:

***Về mặt nghiệp vụ:*** Xây dựng ứng dụng quản quảng bá đồ gỗ một cách hợp lí, rõ ràng và đạt hiệu quả cao.

***Về mặt kinh tế:*** Việc áp dụng công nghệ thông tin vào việc quảng bá về sản phẩm tạo được số lượng người mua tiếp cận dễ dàng nâng cao thu nhập.

***Về mặt sử dụng:*** Những người sử dụng ứng dụng này chỉ cần tải ứng dụng về chiếc điện thoại của mình là có thể sử dụng được.

*b. Phạm vi:*

Ứng dụng Đồ gỗ Mỹ nghệ áp dụng đối với những người sử dụng điện thoại thông minh có hệ điều hành Android.

**2.1.3.2.Yêu cầu đối với ứng dụng**

- Mô tả về môi trường hoạt động: Ứng dụng này sẽ hoạt động trên các chiếc điện thoại thông minh chạy trên nền tàng hệ điều hành Android. Mọi người ai sở hữu chiếc điện thoại như vậy đều có thể cài đặt và sử dụng.

- Giao diện đẹp, dễ sử dụng, thân thiện người dùng.

- Thời gian build ứng dụng càng nhanh càng tốt.

- Xây dựng ứng dụng cần chính xác, tránh sai xót.

- Dung lượng ứng dụng càng bé càng tốt.

**2.1.3.3. Yêu cầu chức năng**

\*Chức năng chính

- Tìm kiếm sản phẩm

- Xem thông tin sản phẩm

- Quản lý người dùng

- Đặt mua sản phẩm

\*Chức năng người dùng

- Mỗi người dùng có thể đăng ký tài khoản để truy cập vào hệ thống với các lợi ích sau:

1. Lưu lại được thông tin cá nhân của chính người dùng

2. Có thể xem lại lịch sử mua hàng

- Đối với người quản lý (Admin) đăng nhập vào hệ thống bằng tài khoản được cung cấp trước nhầm.

1. Thêm sửa xóa sản phẩm.

2. Quản lý danh sách người dùng

3. Quản lý danh sách người mua hàng để chuyển hàng

\*Chức năng đặt mua sản phẩm

- Đối với người dùng đã có tài khoản trong hệ thống thì thông tin sẽ được lấy trực tiếp trong tài khoản người dùng đó.

- Đối với người mua hàng chưa có tài khoản thì sẽ có 1 phiếu điền thông tin cá nhân phục vụ cho việc chuyển hàng.

### 2.2. Phân tích hệ thống

### 2.2.1. Xây dựng mô hình use – case

**2.2.1.1. Xác định các tác nhân**

*Bảng 2: Các tác nhân của ứng dụng*

|  |  |
| --- | --- |
| **Tác nhân** | **Mô tả** |
| Khách hàng | Khách hàng sau khi hoàn thành việc cài đặt ứng dụng, thì có thể xem được các thông tin về sản phẩm cùng với chức năng tìm kiếm sản phẩm. Để có thể mua hàng thì khách hàng phải đăng ký tài khoản sử dụng hoặc đăng nhập nếu đã có tài khoản với các thông tin cá nhân của chính khách hàng. Khách hàng cũng có thế không cần tài khoản và thực hiện việc mua hàng trực tiếp với chủ sản phẩm. |
| Admin | Thực hiện các công việc quản lý sản phẩm, quản lý người dùng |

**2.2.1.2.Xác định các UC**

***a. Các use - case chung.***

*Bảng 3:Use case chung của ứng dụng*

|  |  |  |
| --- | --- | --- |
| **Ca sử dụng** | **Tác nhân** | **Thực hiện** |
| Đăng ký | Khách hàng | Trước khi người dùng sử dụng muốn đăng nhập vào hệ thống để thực hiện chức năng đặt mua sản phẩm. |
| Đăng nhập | Khách hàng | Trước khi người dùng xác lập dặt mua hàng. |
| Quản lý tài khoản | Khách hàng | Người dùng sẽ thực hiện đăng ký tài khoản, đăng nhập, khi quên mật khẩu thì có thể yêu cầu hệ thống để cài đặt lại mật khẩu, thực hiện đăng xuất hoặc xóa tài khoản của mình trên hệ thống, có thể chỉnh sửa các thông tin cá nhân của mình trên hệ thống. |
|  |  |  |
| Quản lý lịch sử mua hàng | Khách hàng | Người dùng có thể xem lại những thông tin của các sản phẩm đã mua và cũng có thể thực hiện việc xóa lịch sử mua hàng. |

***b. Chi tiết các use – case:***

**\* Quản lý tài khoản**:

*Bảng 4: Chi tiết các use case con của use case “Quan ly tai khoan”*

|  |  |  |
| --- | --- | --- |
| **Ca sử dụng** | **Tác nhân** | **Thực hiện** |
| Quên mật khẩu | Khách hàng | Khi người dùng muốn sử dụng ứng dụng nhưng lại quên mât khẩu, người dùng có thể thông báo và ứng dụng sẽ cho phép người dùng cài đặt lại mật khẩu ứng dụng của mình. |
| Sửa tài khoản | Khách hàng | Người dùng có quyền thêm các thông tin còn thiếu, sửa đổi các thông tin sai, xóa các thông tin không cần thiết trong các thông tin của tài khoản cá nhân người dùng. |
| Đăng xuất | Khách hàng | Khi người dùng muốn thoát tài khoản của mình ra khỏi thiết bị hoặc muốn chuyển đổi tài khoản sử dụng trên thiết bị, người dùng thực hiện đăng xuất để thoát tài khoản của mình ra khỏi thiết bị đang sử dụng ứng dụng hiện tại. |
| Xóa tài khoản | Khách hàng | Khi người dùng không có nhu cầu sử dụng ứng dụng nữa, người dùng có thể xóa tài khoản của mình khỏi hệ thống. |

**2.2.1.3 Đặc tả các use – case**

* **Đặc tả use case “đăng ký”**

- *Tên ca sử dụng:* Dang ky

- *Mục đích*: Để có thể sử dụng được các chức năng có trong ứng dụng thì người dùng cần phải có tài khoản trong ứng dụng, vì vậy người dùng phải thực hiện đăng ký tài khoản sử dụng trong ứng dụng.

- *Tác nhân:* Khách hàng

- *Luồng sự kiện:*

* Luồng sự kiện chính: Người dùng sẽ thực hiện đăng ký tài khoản có thể bằng gmail, facebook, hoặc google, nếu người dùng đăng ký thành công thì cơ sở dữ liệu của ứng dụng sẽ lưu thông tin mà người dùng đã đăng ký.
* Luồng sự kiện rẽ nhánh: Người dùng thực hiện đăng ký tài khoản để sử dụng ứng dụng, nếu đăng ký bị lỗi thì hệ thống sẽ đưa ra thông báo lỗi và yêu cầu người dùng đăng ký lại tài khoản:

*\* TH 1*: Người dùng đăng ký bằng tài khoản email:

Nếu tài khoản email này đã được đăng ký thì hệ thống sẽ thông đưa ra thông báo “email này đã tồn tại” và yêu cầu người dùng sử dụng một tài khoản email khác.

Nếu mật khẩu ngắn hoặc không đảm bảo mức độ phức tạp mà ứng dụng yêu cầu thì sẽ có thông báo “yêu cầu nhập lại mật khẩu” để đảm bảo an toàn cho tài khoản.

Trong textbox nhập lại mật khẩu: Người dùng cần nhập mật khẩu giống với mật khẩu ở trên, nếu người dùng nhập không giống thì ứng dụng sẽ đưa ra thông báo và yêu cầu người dùng nhập lại cho đúng mật khẩu ở trên.

Sau khi nhấn nút đăng ký nếu có lỗi đăng ký thì hệ thống sẽ đưa ra thông báo lỗi.

\* *TH 2:* Người dùng đăng ký bằng tài khoản facebook:

Khi người dùng chọn cách đăng ký bằng tài khoản facebook thì hệ thống sẽ yêu cầu người dùng nhập tài khoản facebook của mình để có thể sử dụng hệ thống.

Nếu tài khoản facebook này đã được đăng ký hoặc tài khoản facebook không tồn tại thì hệ thống sẽ đưa ra thông báo lỗi và yêu cầu người dùng sử dụng tài khoản khác để đăng ký hoặc có thể đăng ký theo cách khác.

\* *TH 3:* Người dùng đăng ký bằng tài khoản google:

Khi người dùng chọn cách đăng ký bằng tài khoản google thì hệ thống sẽ yêu cầu người dùng nhập tài khoản google của mình để đăng ký và sử dụng ứng dụng.

Nếu tài khoản google này đã được đăng ký hoặc tài khoản google không tồn tại thì hệ thống sẽ đưa ra thông báo lỗi và yêu cầu người dùng sử dụng tài khoản khác để đăng ký hoặc có thể đăng ký bằng cách khác.

*- Hậu điều kiện:* Nếu người dùng đăng ký thành công thì cơ sở dữ liệu sẽ lưu thêm tài khoản người dùng, nếu đăng ký không thành công thì dữ liệu không có gì thay đổi.

* **Đặc tả use-case ‘đăng nhập’**

- *Tên ca sử dụng:* Dang nhap

- *Mục đích:* Để sử dụng các chức năng trong ứng dụng thì đòi hỏi người dùng phải thực hiện đăng nhập bằng tài khoản của mình

- *Tác nhân:* người dùng

- *Tiền điều kiện:* Người dùng phải có tài khoản để sử dụng ứng dụng

- *Luồng sự kiện:*

* Luồng sự kiện chính: Người dùng sẽ nhập tên tài khoản và mật khẩu của mình, nếu không có sai xót gì thì người dùng sẽ đăng nhập được vào ứng dụng.
* Luồng sự kiện rẽ nhánh: Người dùng thực hiện đăng nhập bằng tài khoản mà mình đã đăng ký để sử dụng ứng dụng:

Nếu tên người dùng hoặc mật khẩu của bạn nhập vào bị sai so với dữ liệu được lưu trữ thì ứng dụng sẽ đưa ra thông báo lỗi và yêu cầu người dùng đăng nhập lại.

Nếu người dùng quên mật khẩu không đăng nhập được vào ứng dụng thì có thể lựa chọn textview “ quên mật khẩu” trên màn hình. Sau đó bạn sẽ nhập địa chỉ email mà bạn dùng để đăng ký tài khoản và hệ thống sẽ gửi mã xác nhận vào địa chỉ email của bạn. Sau đó bạn nhập lại mã xác nhận vào hệ thống:

\* TH 1: Nếu người dùng nhập sai mã xác nhận thì hệ thống sẽ thông báo lỗi và người dùng không được quyền đổi mới mật khẩu.

\* TH2: Nếu người dùng nhập đúng mã xác nhận thì sẽ di chuyển đến màn hình cho phép người dùng đổi mới mật khẩu.

*- Hậu điều kiện:* Sau khi người dùng đăng nhập thành công vào ứng dụng thì sẽ được quyền sử dụng các chức năng có trong ứng dụng. Trong trường hợp người dùng đăng ký không thành công vào ứng dụng thì không có gì xảy ra.

* **Đặc tả use – case “Quản lý tài khoản”**

*-* *Tên use - case:* “Quan ly tai khoan”

*- Mục đích:* Để quản ký thông tin tài khoản của người dùng như các thông tin cá nhân…

*- Tác nhân:* người dùng

- *Tiền điều kiện:* Người dùng đã đăng nhập vào ứng dụng.

*- Luồng sự kiện:*

\* Luồng sự kiện chính: Người dùng có thể xem lại thông tin cá nhân mà mình đã đăng ký, đồng thời người dùng có quyền sửa các thông tin cá nhân trong tài khoản và thực hiện đăng xuất hoặc xóa tài khoản.

\* Luồng sự kiện rẽ nhánh:

Khi người dùng muốn sửa thông tin của mình trên ứng dụng, việc sửa đổi có thể không thành công. Lúc đó sẽ có thông báo lỗi được đưa ra và yêu cầu người dùng thực hiện lại thao tác sửa.

Khi người dùng không muốn sử dụng ứng dụng nữa người dùng có thể xóa tài khoản của mình khỏi ứng dụng. Nếu xóa không thành công thì sẽ có thông báo “xóa không thành công” đến người dùng.

- *Hậu điều kiện:* Khi việc sửa đổi tài khoản hoặc việc xóa tài khoản thành công thì trong cơ sở dữ liệu sẽ có sự thay đổi. Nếu những hành động trên không thực hiện hoặc thực hiện không thành công thì cơ sở dữ liệu sẽ không có thay đổi gì.

* **Đặc tả use – case “Xem thông tin sản phẩm”**

*-* *Tên use - case:* “Xem thông tin sản phẩm”

*- Mục đích:* Để người dùng có thế nắm bắt được sản phẩm hình dáng như nào, chất lượng như nào, giá cả ra làm sao …..

*- Tác nhân:* người dùng

- *Tiền điều kiện:* hoàn thành việc cài đặt ứng dụng trên smartphone

*- Luồng sự kiện:*

\* Luồng sự kiện chính: Sau khi hoàn thành việc cài đặt ứng dụng trên thiết bị di động Người dùng có thể xem thông tin sản phẩm mà minhg quan tâm cũng như các sản phẩm khác.

* **Đặc tả use – case “Quản lý lịch sử mua hàng”**

*-* *Tên use - case:* “Quản lý lịch sử mua hàng”

*- Mục đích:* Để người dùng có thể xem lại các sản phẩm đã mua cùng với các thông tin liên quan đến sản phẩm đó.

*- Tác nhân:* người dùng

- *Tiền điều kiện:* Người dùng đã đăng nhập vào ứng dụng.

*- Luồng sự kiện:*

\* Luồng sự kiện chính: Người dùng có thể xem lại thông tin sản phẩm mà mình đã mua từ trước, đồng thời người dùng có quyền xóa các thông tin sản phẩm trong tài khoản.

- *Hậu điều kiện:* Khi việc xóa lịch sử thành công thì trong cơ sở dữ liệu sẽ có sự thay đổi. Nếu những hành động trên không thực hiện hoặc thực hiện không thành công thì cơ sở dữ liệu sẽ không có thay đổi gì.

* **Đặc tả use – case “Đặt mua”**

*-* *Tên use - case:* “Đặt mua”

*- Mục đích:* Để người dùng có thể đặt mua sản phẩm cho chính bản thân người dùng và thực hiện các thông tin xác nhận để nhận sẩn phẩm.

*- Tác nhân:* người dùng

- *Tiền điều kiện:* Người dùng đã đăng nhập vào ứng dụng.

*- Luồng sự kiện:*

\* Luồng sự kiện chính: Người dùng sau quá trình tìm hiểu về sản phẩm và có ý định mua sản phẩm cần. Sẽ thực hiện việc đặt mua để chủ sản phẩm biết và thực hiện hoạt động bàn giao sản phẩm.

- *Hậu điều kiện:*Sau khi nhận hàng Người dùng phải thực hiện việc thanh toán cho sản phẩm mua theo giá cả đã có từ trước.

* **Đặc tả use case “Quản lý sản phẩm”**

*-Tên use - case:* “Quản lý sản phẩm”

*- Mục đích:* Để admin có thể quản lý 1 lượng sản phẩm lớn 1 cách dễ dàng nhanh chóng và chính xác.

*- Tác nhân:* admin

- *Tiền điều kiện:* Admin đăng nhập vào trang quản lý cho trước

*- Luồng sự kiện:*

\* Luồng sự kiện chính: Admin sau khi truy nhập thành công vào trang quản lý sẽ chọn chức năng quản lý sản phầm. Tại đây admin có thế thêm, sửa, xóa sản phẩm và thống kê lượng sản phẩm còn lại trong hệ thống.

- *Hậu điều kiện:*Sau khi admin có sử điều chỉnh thành đổi thì phải xác nhận lại thay đổi đó.

* **Đặc tả use case “Quản lý đơn hàng”**

*-Tên use - case:* “Quản lý đơn hàng”

*- Mục đích:* Để quản lý các sản phẩm cần giao 1 cách tiện lợi và chính xác

*- Tác nhân:* admin

- *Tiền điều kiện:* Admin đăng nhập vào trang quản lý cho trước

*- Luồng sự kiện:*

\* Luồng sự kiện chính: Admin sau khi truy nhập thành công vào trang quản lý sẽ chọn chức năng quản lý đơn hàng. Tại đây admin có thế xem lại các đơn đặt hàng của khách hàng để xác định lại và thực hiện chuyển hàng.

- *Hậu điều kiện:*

* **Đặc tả use case “Quản lý tài khoản khách hàng”**

*-Tên use - case:* “Quản lý tài khoản khách hàng”

*- Mục đích:* Để nắm bắt được lượng khách hàng đã tham gia hệ thống, quản lý thống tin cũng như đánh giá nhìn nhận về khách hàng.

*- Tác nhân:* admin

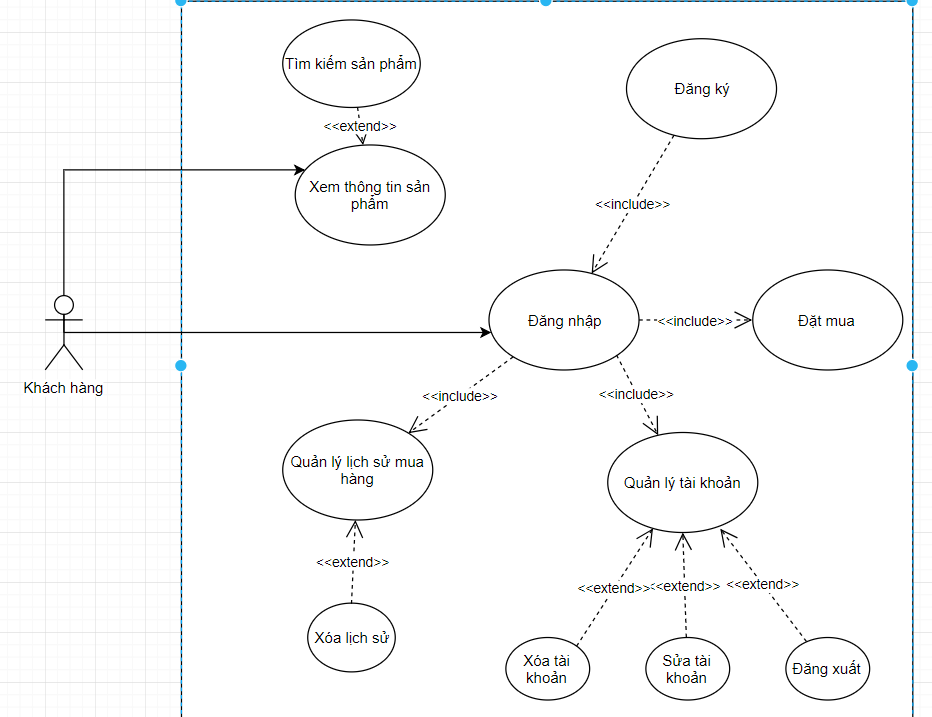
- *Tiền điều kiện:* Admin đăng nhập vào trang quản lý cho trước

*- Luồng sự kiện:*

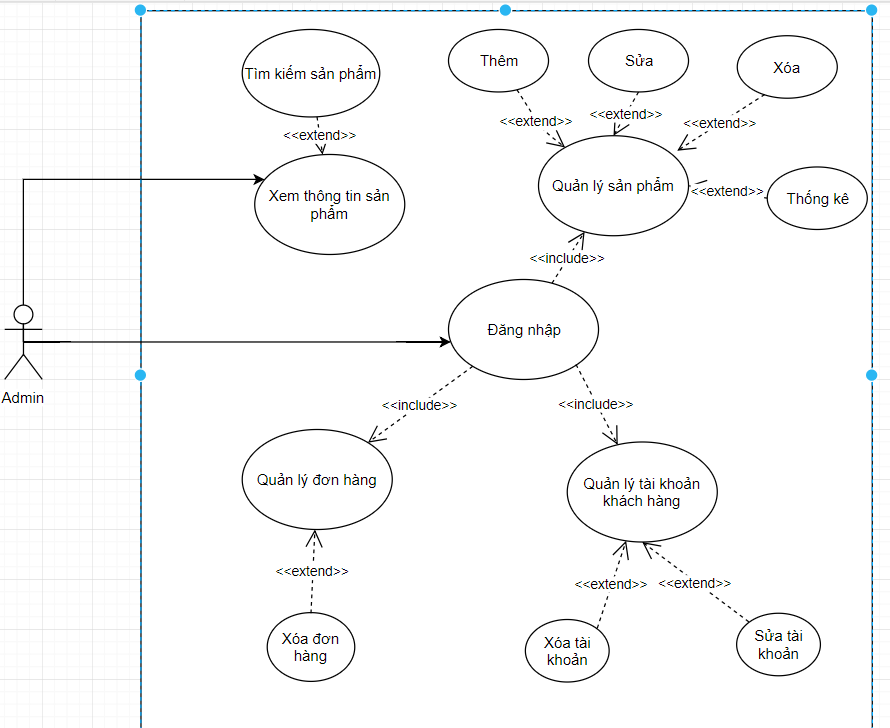
\* Luồng sự kiện chính: Admin sau khi truy nhập thành công vào trang quản lý sẽ chọn chức năng quản lý tài khoản khách hàng. Tại đây admin có thế sửa, xóa thông tin khách hàng và đánh giá mức độ về khách hàng.

- *Hậu điều kiện:*

**2.2.1.4. Biểu đồ UC**



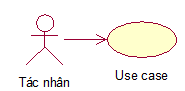
*Hình 7: Biểu đồ use case của khách hàng*

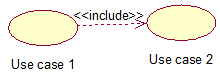


*Hình 8. Biểu đồ Use Case của Admin*

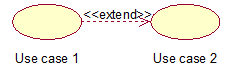
Chú giải:



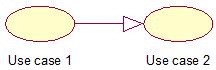
Association: thường được dùng để mô tả mối quan hệ giữa Actor và Use Case và giữa các Use Case với nhau.



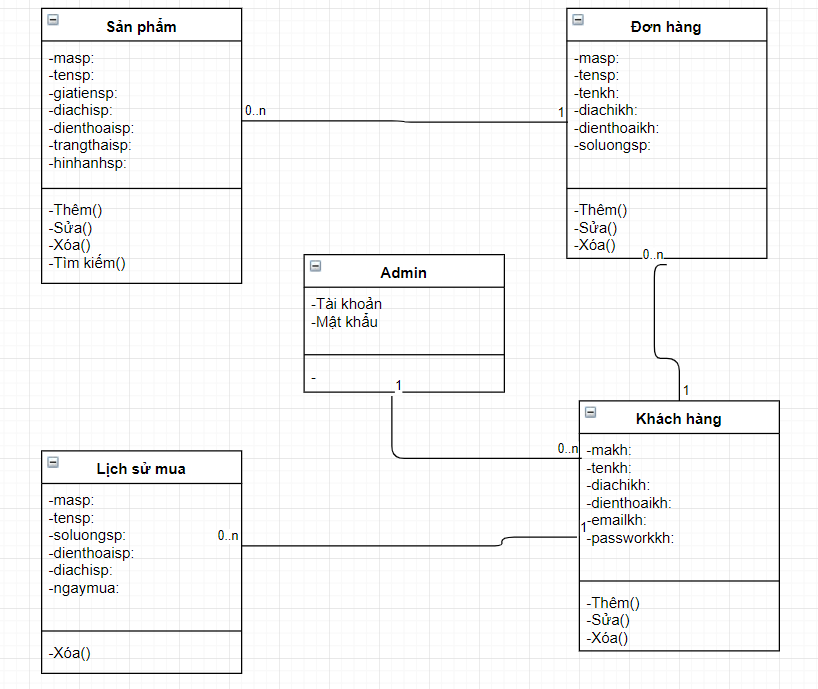
Include: là quan hệ giữa các Use Case với nhau, nó mô tả việc một Use Case lớn được chia ra thành các Use Case nhỏ để dễ cài đặt (module hóa) hoặc thể hiện sự dùng lại



Extend: Extend dùng để mô tả quan hệ giữa 2 Use Case. Quan hệ Extend được sử dụng khi có một Use Case được tạo ra để bổ sung chức năng cho một Use Case có sẵn và được sử dụng trong một điều kiện nhất định nào đó.

Generalization: được sử dụng để thể hiện quan hệ thừa kế giữa các Actor hoặc giữa các Use Case với nhau.

### 2.2.2. Mô hình lớp

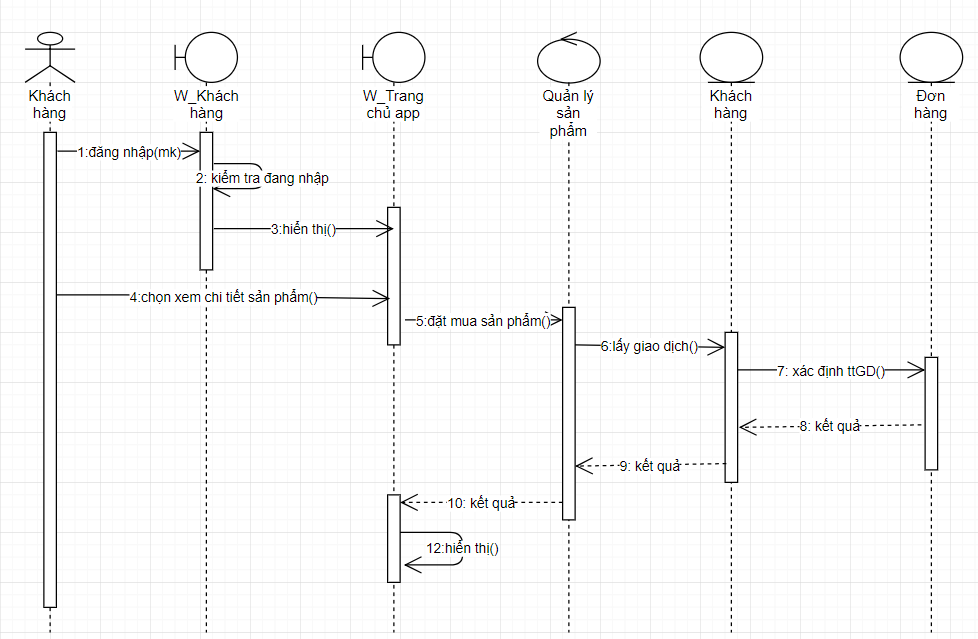


*Hình 9. Biểu đồ lớp*

### 2.2.3. Hiện thực hóa các use case

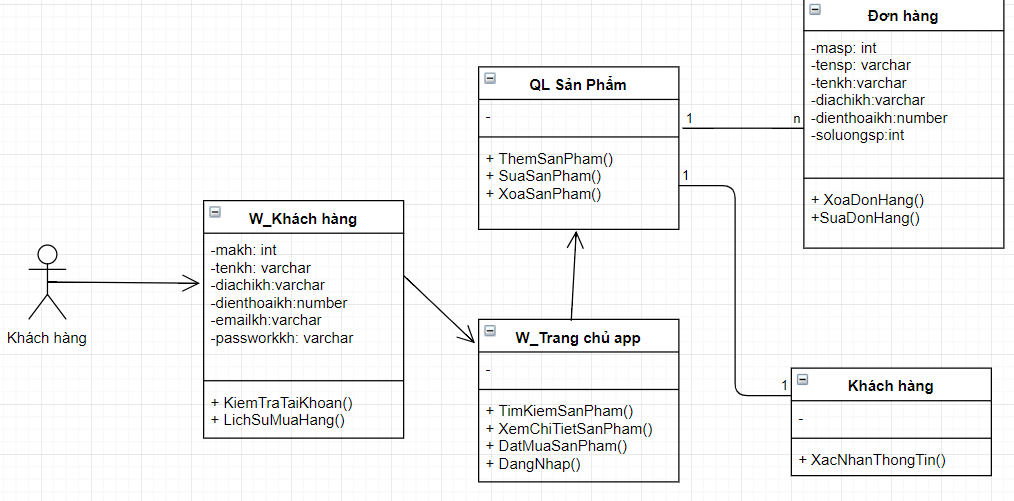
**2.2.3.1 Hiện thực hóa UC Đặt mua**

* Biểu đồ trình tự



*Hình 10. Biểu đồ trình tự đặt mua sản phẩm*

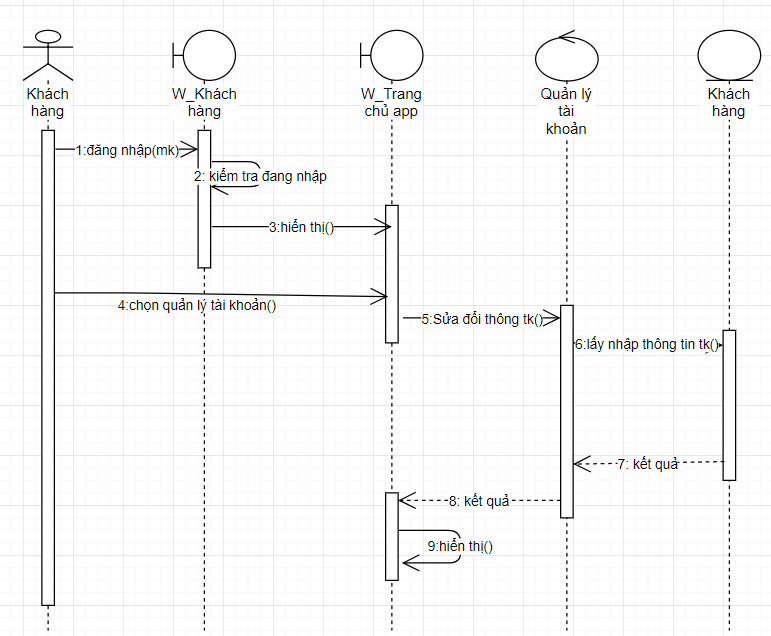
* *Biểu đồ các lớp tham gia*



*Hình 11. Biểu đồ các lớp tham gia UC Đặt mua*

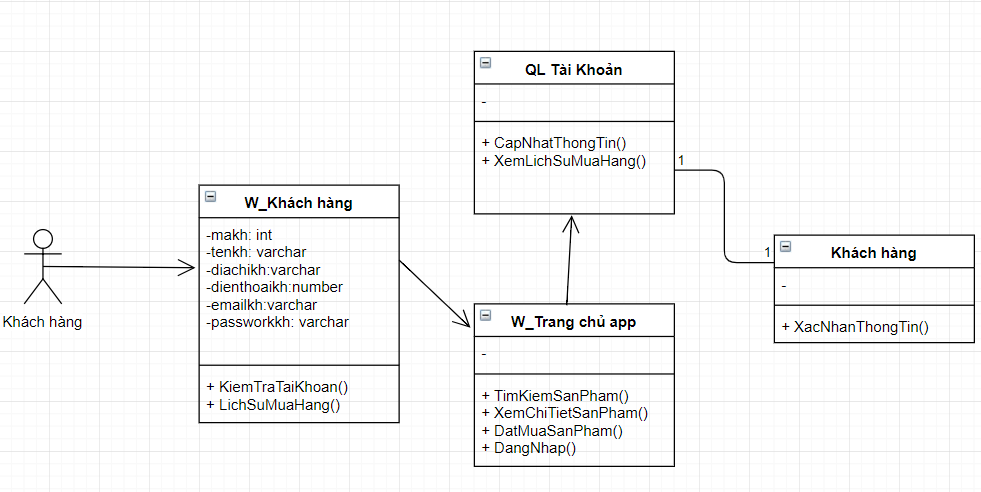
**2.2.3.2. Hiện thực hóa UC Quản lý tài khoản**

* Biểu đồ tuần tự



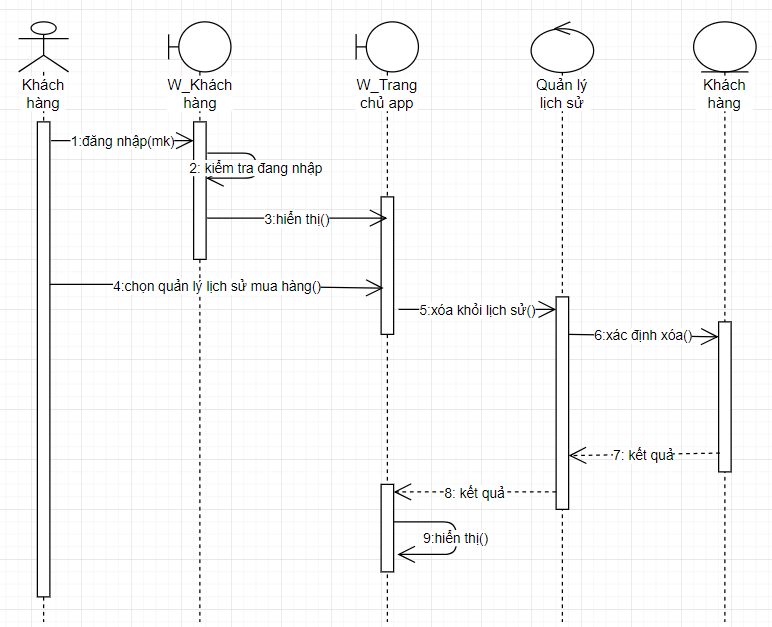
*Hình 12. Biểu đồ trình tự quản lý tài khoản*

* *Lớp tham gia*

 *Hình 13. Lớp tham gia UC Quản lý tài khoản*

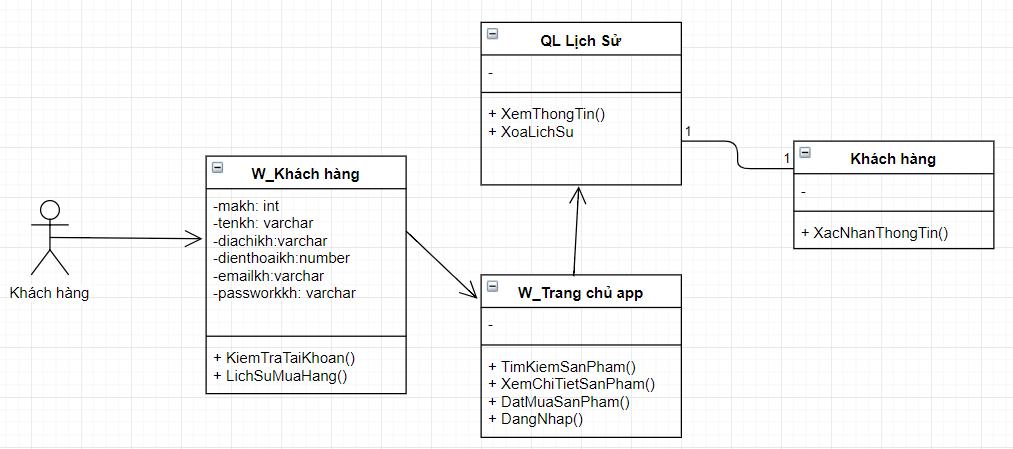
**2.2.3.3. Hiện thực hóa UC Quản lý lịch sử mua hàng**

* Biểu đồ tuần tự



*Hình 14. Biểu đồ trình tự quản lý lịch sử mua hàng*

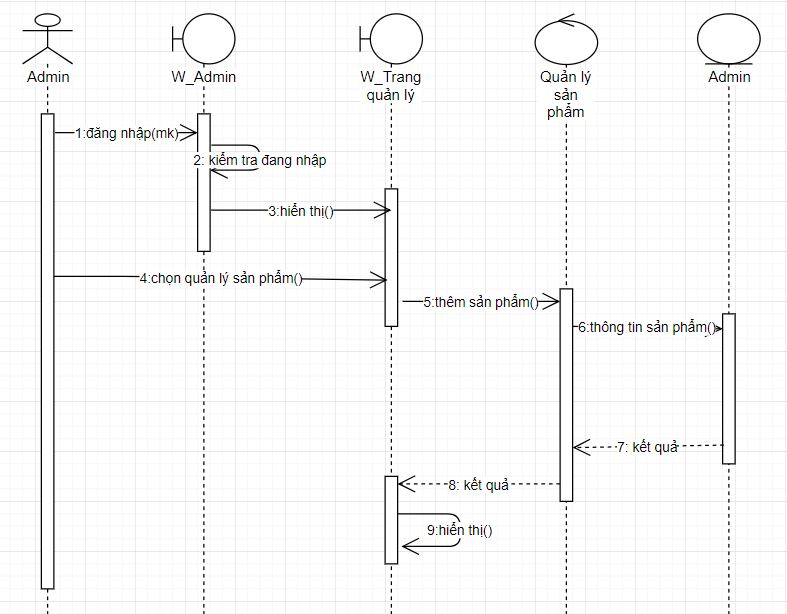
* *Biểu đồ các lớp tham gia*



*Hình 15. Biểu đồ các lớp tham gia UC Quản lý lịch sử mua hàng*

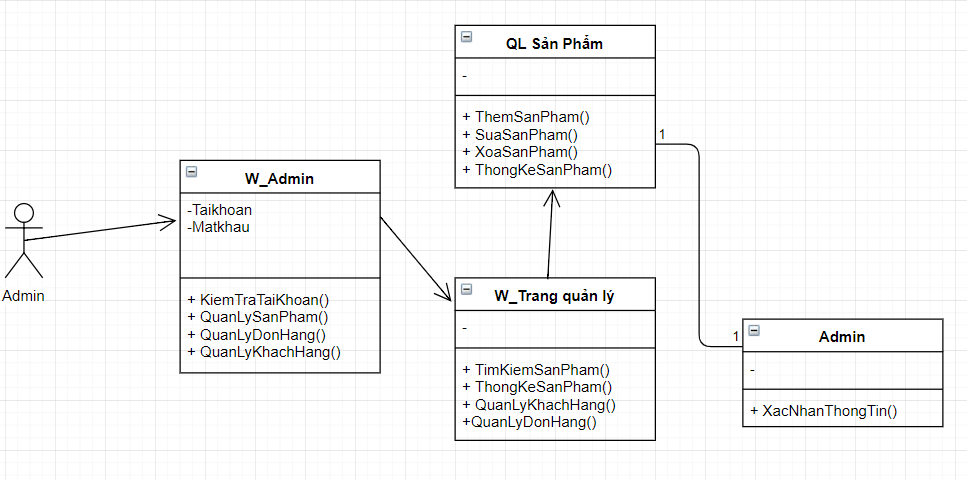
***2.2.3.4. Hiện thực hóa UC Quản lý sản phẩm***

* Biểu đồ trình tự



*Hình 16. Biểu đồ trình tự quản lý sản phẩm*

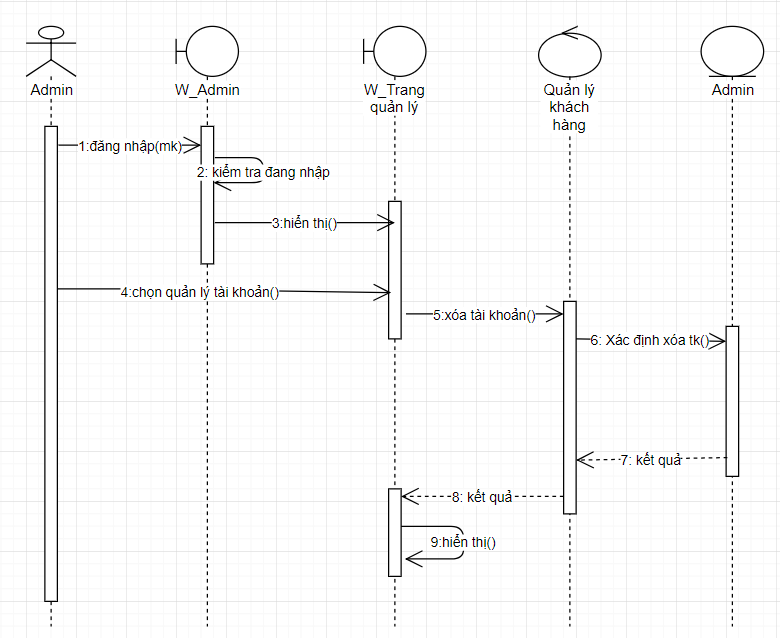
* *Biểu đồ các lớp tham gia*



*Hình 17. Biểu đồ các lớp tham gia UC Quản lý sản phẩm*

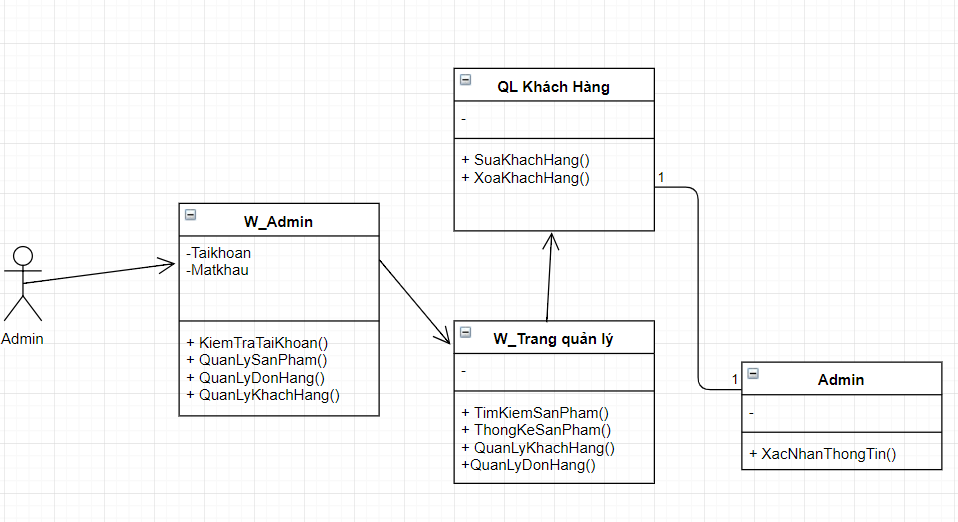
***2.2.3.5. Hiện thực hóa UC Quản lý khách hàng***

* Biểu đồ trình tự



*Hình 18. Biểu đồ trình tự quản lý khách hàng*

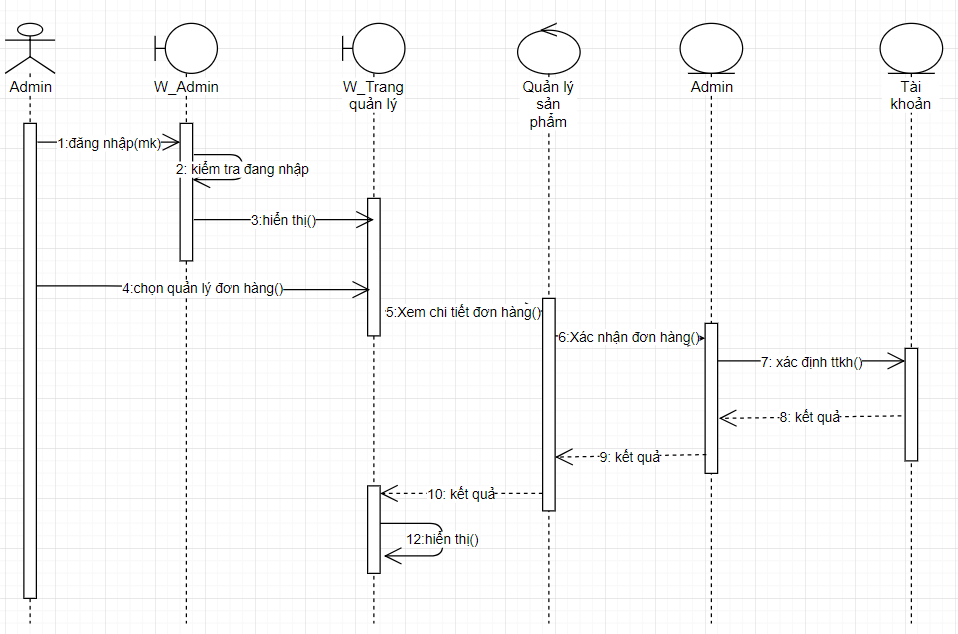
* *Biểu đồ các lớp tham gia*



*Hình 19. Biểu đồ các lớp tham gia UC Quản lý khách hàng*

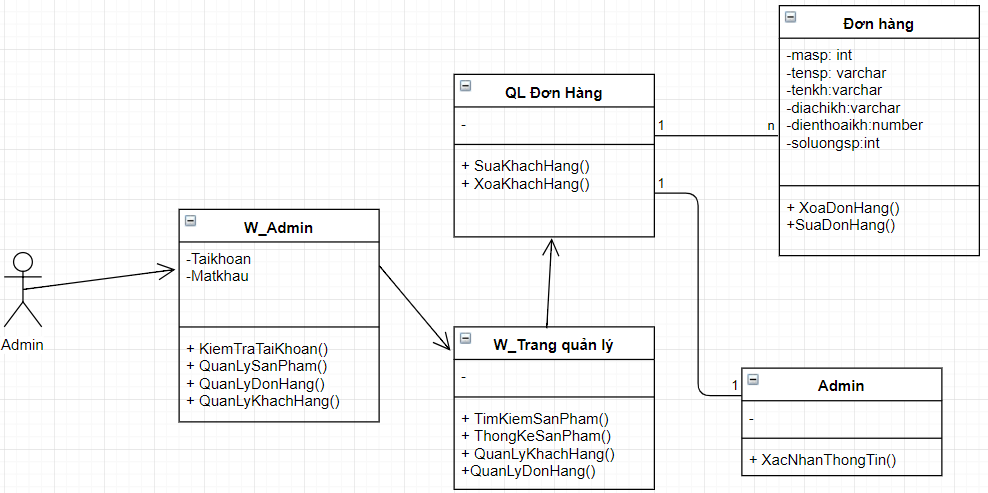
***2.2.3.6. Hiện thực hóa UC Quản lý đơn hàng***

* *Biểu đồ trình tự*



*Hình 20. Biểu đồ trình tự UC Quản lý đơn hàng*

* *Biểu đồ các lớp tham gia*



*Hình 21. Biểu đồ các lớp tham gia UC Quản lý đơn hàng*

### 2.3. Thiết kế

### 2.3.1. Thiết kế cơ sở dữ liệu

*Bảng 5: Cơ sở dữ liệu sản phẩm*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Điều kiện** | **Diễn giải** |
| MaSP | Int (Autoincrement) | PRIMARY KEY | Mã sản phẩm |
| TenSP | Varchar(100) |  | Tên sản phẩm |
| GiaTienSP | Money |  | Giá tiền sản phẩm |
| DiaChiSP | Varchar(150) |  | Địa chỉ bán sản phẩm |
| PhoneSP | Number |  | Số điện thoại chủ sản phẩm |
| TrangThaiSP | Int |  | Lưu lại trạng thái: nếu là 0 thì là hết hàng, nếu bằng 1 là còn hàng |
| HinhAnhSP |  |  | Hinh anh minh hoa |

*Bảng 6: Cơ sở dữ liệu người dùng*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Điều kiện** | **Diễn giải** |
| MaND | Int (Autoincrement) | Primary key | Mã người dùng |
| TenND | Varchar(30) |  | Tên người dùng |
| DiaChiND | Varchar(150) |  | Địa chỉ người dùng |
| PhoneND | Number |  | Số điện thoại người dùng |
| EmailND | Varchar(30) |  | Hộp thư điện tử |
| PasswordND | Varchar(30) |  | Mật khẩu người dùng |

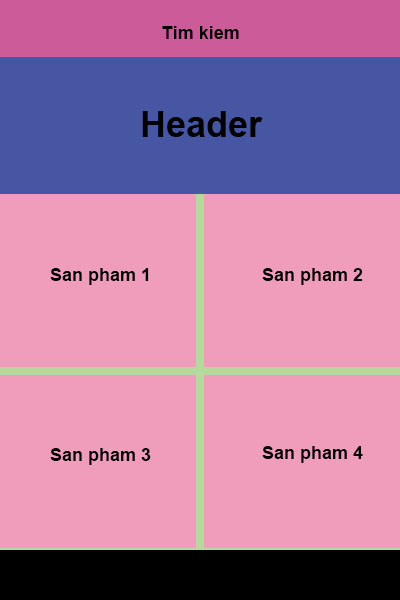
*Bảng 7: Cơ sở dữ liệu đặt mua*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Điều kiện** | **Diễn giải** |
| MaSP | Int | Primary key | Mã sản phẩm |
| TenSP | Varchar(100) |  | Tên sản phẩm |
| DiaChiND | Varchar(150) |  | Địa chỉ người dùng |
| PhoneND | Number |  | Số điện thoại người dùng |
| SoLuongSP | Int |  | Số lượng sản phẩm |

*Bảng 8: Cơ sở dữ liệu lịch sử mua hàng*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Điều kiện** | **Diễn giải** |
| MaSP | Int | Primary key | Mã sản phẩm |
| TenSP | Varchar(100) |  | Tên sản phẩm |
| SoLuongSP | Int |  | Số lượng sản phẩm |
| PhoneSP | Number |  | Số điện thoại chủ sản phẩm |
| DiaChiSP | Varchar(150) |  | Địa chỉ sản phẩm |
| NgayMua | Date |  | Ngày mua sản phẩm |

### *2.3.2. Thiết kế giao diện*

******

*Hình 22. Giao dien trang chủ*

**

*Hình 23. Giao diện chi tiết sản phẩm*

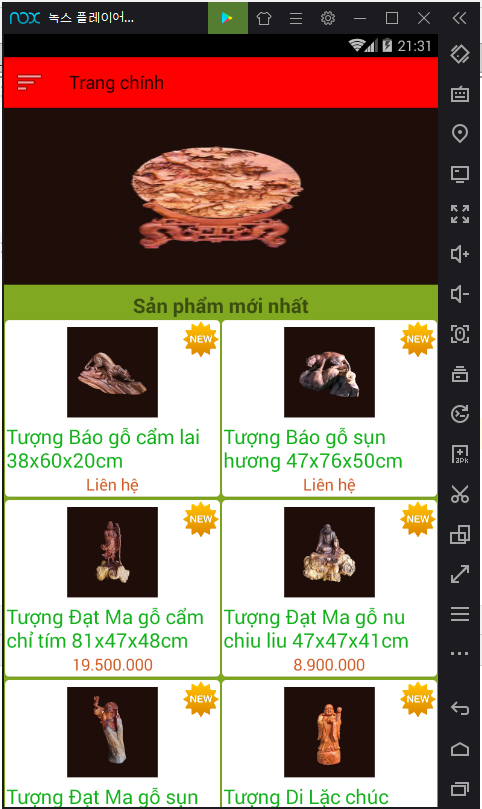
**

*Hình 24. Giao diện trang thông tin khách hàng*

## CHƯƠNG 3. XÂY DỰNG VÀ CÀI ĐẶT PHẦN MỀM

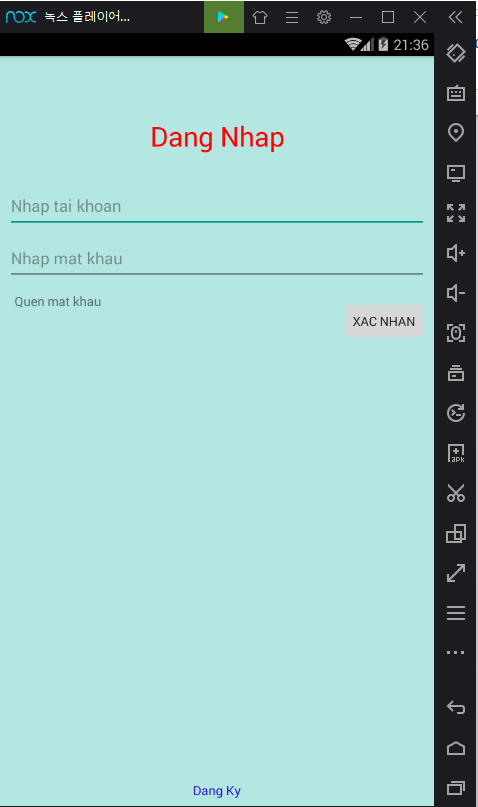
## 3.1 Xây dựng giao diện

* **Trang chủ**



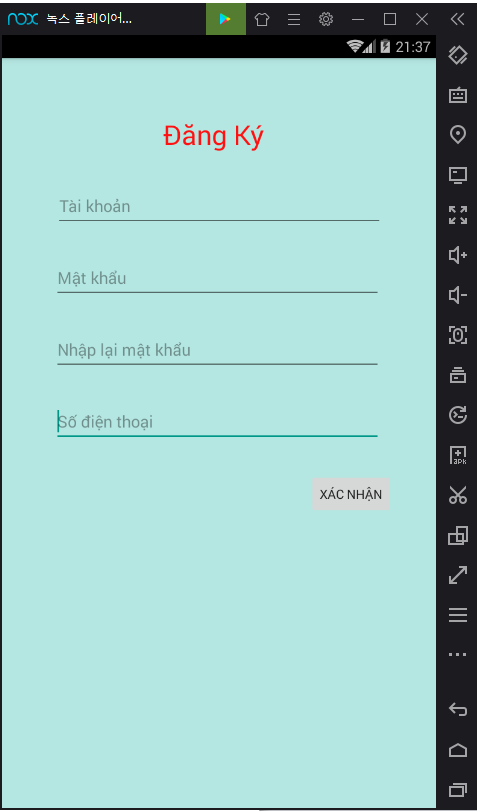
*Hình 25. Giao diện trang chủ*

* **Đăng nhập**



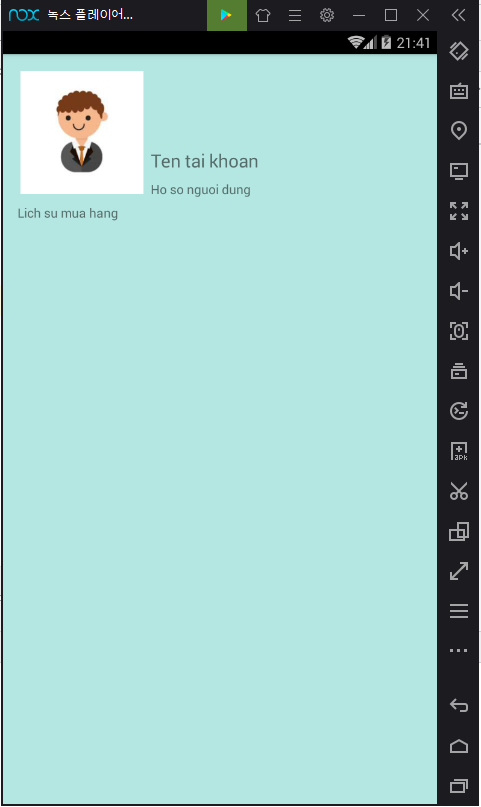
*Hình 26. Giao diện đăng nhập*

* **Đăng ký**



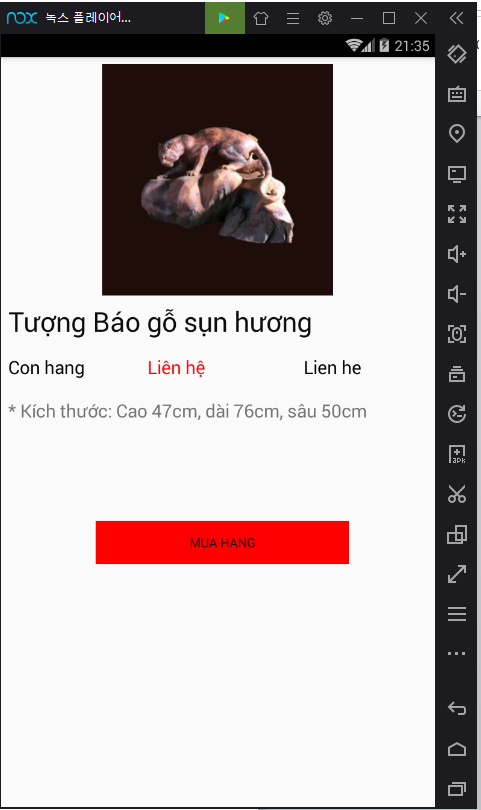
*Hình 27. Giao diện đăng ký*

* **User**



*Hình 28. Giao diện trang cá nhân người dùng*

* **Chi tiết sản phẩm**



*Hình 29. Giao diện chi tiết sản phẩm*