

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**

---



ISO 9001:2015

# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH : CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Nguyễn Bùi Việt Đức**

**Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn**

**HẢI PHÒNG - 2020**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**

-----

**XÂY DỰNG ỨNG DỤNG ANDROID  
XEM TRUYỆN TRANH ONLINE**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY  
NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Nguyễn Bùi Việt Đức  
Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn**

**HẢI PHÒNG – 2020**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG**

---

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

Sinh viên: Nguyễn Bùi Việt Đức      Mã SV: 1512112001

Lớp: CT1901C      Ngành: Công Nghệ Thông Tin

Tên đề tài: Xây dựng ứng dụng android xem truyện tranh online

## NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp
  - a. Nội dung
    - Tìm hiểu hệ điều hành android.
    - Tìm hiểu môi trường lập trình Android Studio.
    - Tìm hiểu lập trình web-service bằng ngôn ngữ PHP
    - Tìm hiểu các cơ sở dữ liệu trên Host
    - Ứng dụng các kỹ thuật đã tìm hiểu xây dựng chương trình ứng dụng thực nghiệm
  - b. Các yêu cầu cần giải quyết
    - Sử dụng công cụ Android Studio để viết chương trình chạy trên điện thoại android
    - Xây dựng giao diện chương trình
2. Các số liệu cần thiết để thiết kế, tính toán.
  - Sử dụng các số liệu giả lập
3. Địa điểm thực tập tốt nghiệp.
  - Trường Đại Học Dân Lập Hải Phòng

## CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

### **Người hướng dẫn thứ nhất:**

Họ và tên: Phùng Anh Tuấn

Học hàm, học vị: Thạc Sĩ

Cơ quan công tác: Trường Đại Học Dân Lập Hải Phòng

Nội dung hướng dẫn:

- Tìm hiểu hệ điều hành android.
- Tìm hiểu môi trường lập trình Android Studio.
- Tìm hiểu lập trình web-service bằng ngôn ngữ PHP
- Tìm hiểu các cơ sở dữ liệu trên Host
- Ứng dụng các kỹ thuật đã tìm hiểu xây dựng chương trình ứng dụng thực nghiệm

### **Người hướng dẫn thứ hai:**

Họ và tên:

Học hàm, học vị:

Cơ quan công tác:

Nội dung hướng dẫn:

Đề tài tốt nghiệp được giao ngày 03 tháng 08 năm 2020

Yêu cầu phải hoàn thành xong trước ngày 20 tháng 10 năm 2020

Đã nhận nhiệm vụ ĐTTN

*Sinh viên*

Đã giao nhiệm vụ ĐTTN

*Người hướng dẫn*

*Hải Phòng, ngày ..... tháng.....năm 2020*

**Hiệu trưởng**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

**PHIẾU NHẬN XÉT CỦA GIÁNG VIÊN HƯỚNG DẪN TỐT NGHIỆP**

Họ và tên giảng viên: Phùng Anh Tuấn.

Đơn vị công tác: Trường Đại học Dân lập Hải Phòng.

Họ và tên sinh viên: Nguyễn Bùi Việt Đức - Ngành: Công nghệ Thông tin

Nội dung hướng dẫn:

- Tìm hiểu hệ điều hành android.
- Tìm hiểu môi trường lập trình Android.
- Tìm hiểu lập trình web-service bằng ngôn ngữ PHP
- Tìm hiểu các cơ sở dữ liệu trên Host
- Ứng dụng các kỹ thuật đã tìm hiểu xây dựng chương trình ứng dụng thực nghiệm

**1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:**

- .....  
.....  
.....

**2. Đánh giá chất lượng của khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T. T.N trên các mặt lý luận, thực tiễn, tính toán số liệu...):**

- .....  
.....  
.....

**3. Cho điểm của cán bộ hướng dẫn (ghi bằng cả số và chữ):**

Đạt  Không đạt  Điểm:.....

Hải Phòng, ngày ... tháng ... năm 2020

Cán bộ hướng dẫn  
(Ký và ghi rõ họ tên)

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

Độc lập - Tự do - Hạnh phúc

**PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHĂM PHẢN BIỆN**

Họ và tên giảng viên:

Đơn vị công tác:

Họ và tên sinh viên: ..... Ngành: .....

Đề tài tốt nghiệp:

1. Phần nhận xét của giảng viên chăm phản biện

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

2. Những mặt còn hạn chế

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

3. Ý kiến của giảng viên chăm phản biện

Được bảo vệ  Không được bảo vệ:  Điểm: .....

Hải Phòng, ngày ..... tháng..... năm 2020

Giảng viên chăm phản biện

(Ký và ghi rõ họ tên)

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến quý thầy cô Trường Đại Học Quản Lý Và Công Nghệ Hải Phòng, những người đã dìu dắt em tận tình, đã truyền đạt cho em những kiến thức và bài học quý giá trong suốt thời gian em theo học tại trường.

Em xin trân trọng gửi lời cảm ơn đến tất cả các thầy cô trong khoa Công Nghệ Thông Tin, đặc biệt là thầy giáo ThS. Phùng Anh Tuấn, thầy đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình làm tốt nghiệp. Với sự chỉ bảo của thầy, em đã có định hướng tốt trong việc triển khai và thực hiện các yêu cầu trong quá trình làm đồ án tốt nghiệp.

Em xin cảm ơn những người thân và gia đình đã quan tâm, động viên và luôn tạo cho em những điều kiện tốt nhất trong suốt quá trình học tập và làm tốt nghiệp.

Ngoài ra em cũng xin gửi lời cảm ơn đến tất cả bạn bè, đặc biệt là các bạn trong lớp CT1901C đã luôn gắn bó, cùng học tập và giúp đỡ em trong những năm qua và trong suốt quá trình thực hiện đồ án này.

Em xin chân thành cảm ơn!

*Hải Phòng, ngày ..... tháng ..... năm 2020*

**Sinh viên**

Nguyễn Bùi Việt Đức



## MỤC LỤC

<b>CHƯƠNG 1 : GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH ANDROID.....</b>	<b>16</b>
<b>1.1. Lịch sử ra đời . . . . .</b>	<b>16</b>
1.1.1. Biểu tượng Android . . . . .	17
1.1.2. Những phiên bản của Android . . . . .	17
<b>1.2. Giao diện . . . . .</b>	<b>23</b>
<b>1.3. Ứng dụng . . . . .</b>	<b>25</b>
<b>1.4. Kiến trúc hệ điều hành Android . . . . .</b>	<b>25</b>
1.4.1. Tầng hạt nhân Linux . . . . .	26
1.4.2. Tầng thư viện . . . . .	27
1.4.3. Tầng thực thi các ứng dụng trên hệ điều hành Android . . . . .	28
1.4.4. Tầng khung ứng dụng . . . . .	28
1.4.5. Tầng ứng dụng . . . . .	29
<b>1.5. Quản lý bộ nhớ . . . . .</b>	<b>30</b>
<b>1.6. Cập nhật . . . . .</b>	<b>31</b>
<b>1.7. Bảo mật và riêng tư . . . . .</b>	<b>32</b>
<b>1.8. Ưu Nhược điểm . . . . .</b>	<b>32</b>
1.8.1. Ưu điểm . . . . .	32
1.8.2. Nhược điểm . . . . .	32
<b>1.9. Thiết bị . . . . .</b>	<b>33</b>
<b>1.10 Kết luận . . . . .</b>	<b>33</b>
<b>CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO.....</b>	<b>34</b>
<b>1.1. Thiết lập môi trường . . . . .</b>	<b>34</b>
2.1.1. Cài đặt Java . . . . .	34
2.1.2. Cài đặt Android Studio . . . . .	35
<b>2.2. Các thành phần trong Android Studio . . . . .</b>	<b>43</b>
2.2.1. File manifests . . . . .	43
2.2.2. File Java . . . . .	44
2.2.3. File tài nguyên . . . . .	44
2.2.4. File Grandle . . . . .	45
<b>2.3.View . . . . .</b>	<b>45</b>
2.3.1. TextView . . . . .	45
2.3.2. Button . . . . .	46
2.3.3. ImageButton . . . . .	47
2.3.4. ImageView . . . . .	47

2.3.5. EditText	47
2.3.6. ListView	48
2.3.7. GridView	49
<b>2.4. ViewGroup</b>	<b>49</b>
2.4.1. LinearLayout	49
2.4.2. RelativeLayout	50
2.4.3. Framelayout	50
2.4.4. Tablelayout	51
<b>2.5. Activity&amp;Intent</b>	<b>51</b>
2.5.1. Activity	51
2.5.2. Intent	51
<b>CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....</b>	<b>52</b>
<b>3.1. Phát biểu bài toán</b>	<b>52</b>
<b>3.2. Phân tích thiết kế</b>	<b>53</b>
3.2.1. Phân tích dữ liệu	53
3.2.2. Bảng thiết kế dữ liệu	53
3.2.3. Bảng cơ sở dữ liệu vật lý	53
<b>3.3. Phân tích thiết kế chức năng</b>	<b>54</b>
3.3.1. Sơ đồ phân rã chức năng	54
3.3.2. Chức năng Danh sách truyện	54
3.3.3. Chức năng Thống kê like	58
3.3.4. Chức năng giới thiệu	59
3.3.5. Chức năng Thoát ứng dụng	60
<b>CHƯƠNG 4 : CHƯƠNG TRÌNH ỨNG DỤNG.....</b>	<b>61</b>
<b>4.1. Kết quả đạt được</b>	<b>61</b>
<b>4.2. Một số giao diện chính</b>	<b>61</b>
4.2.1. Giao diện Chính	61
4.2.2. Giao diện Danh sách truyện	62
.....	62
4.2.3. Giao diện Thống kê	67
4.2.4. Giao diện Giới thiệu	69
.....	70
4.2.5. Giao diện Thoát	71
<b>4.3. Những hạn chế và hướng phát triển trong tương lai</b>	<b>72</b>
4.3.1. Hạn chế	72
4.3.2. Hướng phát triển	72

<b>KẾT LUẬN .....</b>	<b>73</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>74</b>

## MỞ ĐẦU

### 1. Lý do chọn đề tài

Ngày nay với sự cùng với sự phát triển như vũ bão của công nghệ thông tin, các hệ điều hành trên thiết bị di động đang trở nên rất được quan tâm. Các hệ điều hành tiêu biểu như: windows phone của Microsoft, IOS của Apple.... Và hệ điều hành Android. Android chiếm 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm hiện tại. Sự thành công của hệ điều hành cũng khiến nó trở thành mục tiêu trong các vụ kiện liên quan đến bằng phát minh, góp mặt trong cái gọi là "cuộc chiến điện thoại thông minh" giữa các công ty công nghệ.

Số liệu ấy đã chỉ ra cho chúng ta thấy sự phát triển mạnh mẽ của ngôn ngữ lập trình thiết bị như thế nào. Nếu như chỉ vài năm trước, các ứng dụng như: Internet, Email, game online, ứng dụng văn phòng... thường chỉ xuất hiện trên máy tính bàn hoặc máy tính xách tay thì giờ đây, những ứng dụng đó ngày càng phát triển rộng rãi trên các thiết bị như smartphone, máy tính bảng. Các công ty phần mềm sản xuất ứng dụng di động mới liên tục được mở ra, nhiều công ty phần mềm chuyển hướng hoặc mở rộng sang mảng sản xuất ứng dụng trên thiết bị di động cùng với đó là một hướng đi mới trong việc lựa chọn nghề nghiệp, tương lai cho bản thân.

Đây cũng chính là lý do khiến em chọn đề tài: “ Xây dựng ứng dụng android xem truyện tranh online”, cụ thể là lập trình cho thiết bị Android để làm sáng rõ sức mạnh của hệ điều hành Android như thế nào và cũng như đáp ứng nhu cầu cho các độc giả yêu thích truyện tranh có cơ hội được tiếp cận với nhiều nguồn truyện mới cũng như tạo ra một môi trường giải trí lành mạnh cho giới trẻ hiện nay , việc đọc truyện qua điện thoại sẽ giúp chúng ta tiết kiệm được chi phí thời gian rảnh

### 2. Ý tưởng đề tài

Ứng dụng đọc truyện online là một ứng dụng trên nền tảng android. Ứng dụng sẽ có chức năng tìm kiếm để giúp cho độc giả có thể dễ dàng tìm được bộ truyện mình cần tìm, truyện sẽ hiển thị dưới dạng hình ảnh , chúng ta sẽ dùng thao tác trượt từ dưới lên để đọc.

### **3. Mục tiêu của đề án**

- Tìm hiểu về hệ điều hành android,
- Nắm bắt được cách thức xây dựng một ứng dụng trên hệ điều hành Android.
- Phân tích thiết kế ứng dụng đọc truyện.
- Xây dựng ứng dụng đọc truyện

### **4. Nội dung đề tài**

Đề án nghiên cứu về tổng quan về cơ sở lý thuyết, đánh giá những đặc điểm nổi bật, hạn chế của những ứng dụng đã xây dựng được và phân tích thiết kế hệ thống. Đề tài gồm 4 chương:

- Chương 1 : Giới thiệu về hệ điều hành android
- Chương 2 : Môi trường lập trình android
- Chương 3 : Phân tích thiết kế hệ thống.
- Chương 4 : Chương trình thực nghiệm

## DANH MỤC HÌNH VẼ

- Hình 1.1.1.1 : Biểu tượng của Android
- Hình 1.2.1: Một vài giao diện người dùng tùy biến của Android
- Hình 1.3.1: Kho ứng dụng
- Hình 1.4.1: Kiến trúc hệ điều hành Android
- Hình 1.4.1.1: Tầng Linux
- Hình 1.4.2.1: Tầng thư viện
- Hình 1.4.3.1: Tầng thực thi các ứng dụng trong Android
- Hình 1.4.4.1: Tầng khung ứng dụng
- Hình 1.4.5.1: Tầng ứng dụng
- Hình 2.1.1.1: Cài đặt JDK
- Hình 2.1.2.2.1: Trang download Android Studio
- Hình 2.1.2.2.2: Cài đặt Android Studio bằng bộ cài vừa tải về
- Hình 2.1.2.2.3 : Giao diện cài đặt SDK và AVD
- Hình 2.1.2.2.4: Cài đặt hoàn tất
- Hình 2.1.2.3.1: Giao diện download máy ảo Genymotion
- Hình 2.2.1.1: File Manifests
- Hình 2.2.2.1: File Java
- Hình 2.3 : View & ViewGroup
- Hình 2.3.1.1: TextView trong xml
- Hình 2.3.1.2: Hàm TextView trong java
- Hình 2.3.2.1: Button trong xml
- Hình 2.3.2.2: Hàm Button trong java
- Hình 2.3.3.1: ImageButton trong xml
- Hình 2.3.4.1: ImageView trong Xml
- Hình 2.3.5.1: Hàm EditText
- Hình 2.3.6.1: Hình minh họa Listview
- Hình 2.3.6.2: Listview trong xml
- Hình 2.3.6.3: Gán Listview
- Hình 2.3.7.1: Hàm Gridview
- Hình 2.4.1.1: Linearlayout
- Hình 2.4.2.1: Relativelayout
- Hình 2.5.2.2 : Các thuộc tính của một đối tượng Intent
- Hình 2.5.2.3 : Các Action được định nghĩa sẵn
- Hình 3.2.2.1 : Bảng thiết kế logic

Hình 3.2.3.1 : Bảng dữ liệu vật lý

Hình 3.3.1.1 : Sơ đồ phân rã chức năng

Hình 3.3.2.1 : Đọc truyện trong xml

Hình 3.3.2.2 : Hàm đọc truyện trong .java

Hình 3.3.2.3 : Thiết kế nút tìm kiếm trong xml

Hình 3.3.2.4 : Hàm tìm kiếm trong java

Hình 3.3.2.5 : Thiết kế nút quay lại trong xml

Hình 3.3.2.6 : Hàm quay lại trong java

Hình 3.3.2.7 : Thiết kế nút tải lại trang trong xml

Hình 3.3.2.8 : Hàm tải lại trang trong java

Hình 3.3.2.9 : Thiết kế nút đếm like trong xml

Hình 3.3.2.10 : Hàm đếm like trong java

Hình 3.3.3.1 : Thiết kế nút thông kê like trong xml

Hình 3.3.3.2 : Hàm thông kê like trong java

Hình 3.3.4.1 : Thiết kế nút giới thiệu trong xml

Hình 3.3.4.2 : Thiết kế nút đóng trong xml

Hình 3.3.4.3 : Hàm đóng trong java

Hình 3.3.5.1 : Thiết kế nút thoát màn hình chính trong xml

Hình 3.3.5.2 : Hàm thoát màn hình chính trong java

Hình 4.2.1.1 : Giao diện Chính

Hình 4.2.2.1 : Giao diện Danh sách truyện

Hình 4.2.2.2 : Thoát màn hình chính

Hình 4.2.2.3 : Tìm kiếm truyện cần tìm

Hình 4.2.2.4 : Tải lại trang

Hình 4.2.2.5 : Màn hình chi tiết

Hình 4.2.2.6 : Quay lại màn hình chính

Hình 4.2.2.7 : Tải lại màn hình chi tiết

Hình 4.2.2.7 : Like truyện

Hình 4.2.3.1 : Giao diện Thông kê like

Hình 4.2.4.1 : Giao diện Giới thiệu

Hình 4.2.5.1 : Giao diện Thoát

## CHƯƠNG 1 : GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH ANDROID

### 1.1. Lịch sử ra đời

Android[1] là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Android, Inc. Với sự hỗ trợ tài chính của Google và sau này được chính Google mua lại vào năm 2005 và hệ điều hành Android đã ra mắt vào năm 2007. Chiếc điện thoại đầu tiên chạy Android là HTC Dream được bán vào ngày 22 tháng 10 năm 2008. Biểu trưng của hệ điều hành Android mới là một con robot màu xanh lá cây do hãng thiết kế Irina Blok tại California vẽ.

Chính mã nguồn mở cùng với giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới. Android chiếm 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Tháng 10 năm 2012, đã có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play (cửa hàng ứng dụng chính của Android) ước tính khoảng 25 tỉ lượt. Hiện nay con số này đã giảm xuống do sự ảnh hưởng lớn của iOS từ Apple và một phần nhỏ của Windows Phone, tuy nhiên Android vẫn dẫn đầu thị phần.

Từ năm 2008, Android đã trải qua nhiều lần cập nhật để dần dần cải tiến hệ điều hành, bổ sung các tính năng mới và sửa các lỗi trong những lần phát hành trước. Mỗi bản nâng cấp được đặt tên lần lượt theo thứ tự bảng chữ cái, theo tên của một món ăn tráng miệng; ví dụ như phiên bản 1.5 Cupcake (bánh bông lan nhỏ có kem) tiếp nối bằng phiên bản 1.6. Phiên bản mới nhất hiện nay là 9.0 Pie ra mắt vào tháng 8 năm 2018. Vào năm 2010, Google ra mắt loạt thiết bị Nexus một dòng sản phẩm bao gồm điện thoại thông minh và máy tính bảng chạy hệ điều hành Android, do các đối tác phần cứng sản xuất. HTC đã hợp tác với Google trong chiếc điện thoại thông minh Nexus đầu tiên, Nexus One. Kể từ đó nhiều thiết bị mới hơn đã gia nhập vào dòng sản phẩm này, như điện thoại Nexus 4 và máy tính bảng Nexus 10, lần lượt do LG và Samsung sản xuất. Google xem điện thoại và máy tính bảng Nexus là những thiết bị Android chủ lực của mình, với những tính năng phần cứng và phần mềm mới nhất của Android.[1]



### 1.1.1. Biểu tượng Android



Hình 1.1.1.1. Biểu tượng android

Hình ảnh quen thuộc hiện nay cho hệ điều hành Android giống như sự kết hợp của một con robot và một lõi màu xanh lá cây do Irina Blok tạo ra. Theo Blok, thiết kế cuối được lấy cảm hứng từ việc nhìn vào biểu tượng nhà vệ sinh quen thuộc đại diện cho “đàn ông” và “phụ nữ”

### 1.1.2. Những phiên bản của Android

#### 1. Android 1.5 Cupcake



Ra mắt: ngày 30-4-2009.

Cupcake, tên mã đầu tiên áp dụng cho phiên bản Android . Cupcake mang nhiều tính năng mới như bàn phím ảo có khả năng dự đoán từ đang gõ, từ điển

từ ngữ do người dùng đặt ra, hỗ trợ widget trên giao diện chủ, quay phim và phát lại video clip, lược sử thời gian cuộc gọi, chế độ tự động xoay màn hình theo hướng sử dụng (screen rotation). Trình duyệt web trong Cupcake có thêm khả năng sao chép/ dán (copy/paste).[2]

Ngoài ra, phiên bản này cho phép người dùng hiển thị hình ảnh trong danh bạ, một điểm thú vị mà hầu hết người dùng điện thoại di động muốn có. Màn hình chuyên đổi và hình ảnh khi khởi động máy được làm mới [2]

## 2. Android 1.6 Donut



Ra mắt: ngày 30-9-2009.

Donut khắc phục các chức năng "lỏng lẻo" trong Cupcake, mở rộng chức năng tìm kiếm bằng giọng nói và ký tự đến bookmark và danh bạ. Android Market trở thành "chợ đầu mối" để người dùng tìm kiếm và xem các ứng dụng Android. Ứng dụng chụp ảnh và quay phim trong Donut nhanh hơn. Hệ điều hành hỗ trợ màn hình có độ phân giải lớn hơn, hướng đến các thế hệ smartphone màn hình lớn. [2]

### 3. Android 2.0-2.1 Éclair



Ra mắt: ngày 26-10-2009.

Chỉ sau gần một tháng ra mắt Donut (Android 1.6), Google tung ra Eclair, phiên bản được nhận định là "bước đi lớn" của hệ điều hành này.[2]

\* Nhịp Sống Số: Android lên ba và bước nhảy xa ngoạn mục

Eclair cải tiến rất nhiều, từ giao diện đến ứng dụng bên trong hệ thống. Ứng dụng chụp ảnh tăng cường thêm chức năng zoom số (phóng to), cân bằng trắng, hỗ trợ đèn flash và các hiệu ứng màu sắc.[2]

Hệ thống hoạt động ổn định hơn, cải thiện khả năng xử lý, hỗ trợ kết nối Bluetooth tốt hơn, đặc biệt tùy chọn đồng bộ nhiều tài khoản. Một điểm thuận tiện được đánh giá cao lúc bấy giờ là giao diện danh bạ cho phép nhấn chọn vào một ảnh danh bạ để gọi, nhắn tin hay email đến họ. Giao diện ứng dụng lịch biểu (Calendar) cũng thay đổi.[2]

Eclair là phiên bản Android đầu tiên hỗ trợ ảnh nền động (live wallpaper) dù tùy chọn này tiêu tốn khá nhiều pin.[2]

### 4. Android 2.2 Froyo



Ra mắt: ngày 20-5-2010.

Từ phiên bản 2.0 trở đi, Android dần hoàn thiện hơn. Phiên bản 2.2 (Froyo) mang Adobe Flash đến Android, kéo theo hàng loạt ứng dụng và game trên nền Flash. Người dùng cũng có thể xem video clip nền Flash như YouTube và "ra lệnh" thực hiện cuộc gọi qua Bluetooth.[2]

Một chức năng mới trong Froyo được nhóm người dùng lưu động yêu thích là USB Tethering và Wi-Fi Hotspot, biến chiếc smartphone Android thành thiết bị phát sóng Wi-Fi từ kết nối 3G. Tính năng này được sử dụng rất phổ biến đến ngày nay.[2]

Lần đầu tiên Android cho phép cài đặt ứng dụng (app) lên thẻ nhớ SD thay vì mặc định cài ngay vào bộ nhớ trong của thiết bị. Điểm "đầu tiên" nữa trong Froyo bao gồm mật khẩu đã hỗ trợ số và chữ số. Thiết bị đầu tiên mang nhãn Froyo ra mắt thị trường là HTC Nexus One.[2]

## 5. Android 2.3 Gingerbread



Ra mắt: ngày 6-12-2010

Đến cuối năm 2012, Gingerbread vẫn đang "phủ sóng" trên rất nhiều thiết bị dùng Android, chiếm đến hơn phân nửa (54%). Google hợp tác Samsung trình làng dòng smartphone đầu tiên sử dụng Gingerbread mang tên Nexus S, hỗ trợ công nghệ giao tiếp tầm gần NFC.[2]

Gingerbread đưa vào hệ thống một công cụ quản lý tải tập tin, cho phép theo dõi và truy xuất đến các tập tin đã tải về máy. Hệ thống này hỗ trợ nhiều camera cho các thiết bị có camera mặt sau và trước, quản lý nguồn pin hiệu quả hơn, tiết kiệm thời lượng pin. Phiên bản này khắc phục khá nhiều lỗi từ Froyo, kèm theo một số điều chỉnh trong giao diện người dùng (UI).[2]

## 6. Android 3.0 Honeycomb



Ra mắt: ngày 22-2-2011.

Đây không chỉ là một phiên bản, mà có thể xem là một thế hệ Android đầu tiên dành riêng cho máy tính bảng (tablet), ra mắt cùng tablet Motorola XOOM.[2]

Mang những tính năng từ thế hệ Android 2.x, Android 3.0 cải tiến giao diện phù hợp với cách sử dụng máy tính bảng, bàn phím ảo thân thiện hơn, hỗ trợ xử lý đa tác vụ (multi-tasking), cho phép chuyển đổi qua lại các ứng dụng đang cùng chạy. Không chỉ có bề mặt được trau chuốt, phần lõi hệ thống có các cải tiến tương thích với phần cứng như hỗ trợ chip xử lý (CPU) đa lõi, tăng tốc phần cứng...[2]

Android 3.0 đặt nền móng quan trọng cho thế hệ Android 4.x hợp nhất, khắc phục sự phân mảnh của Android (có các phiên bản riêng dành cho smartphone và tablet)[2]

## 7. Android 4.0 Ice Cream Sandwich



Ra mắt: ngày 19-10-2011.

"Bánh kem sandwich" (ICS) là thế hệ Android được mong đợi nhất đến nay, ra đời cùng dòng smartphone "bom tấn" Samsung Galaxy Nexus, thế hệ smartphone đầu tiên trang bị ICS.[2]

Android 4.0 đưa chức năng truy xuất nhanh các ứng dụng thường dùng vào phần bên dưới giao diện chủ, tùy biến widget, dễ sắp xếp và duyệt danh sách ứng dụng

hơn. Các ứng dụng đã có thể truy xuất nhanh từ màn hình khóa thiết bị (Lock screen), hiện các hãng sản xuất thiết bị chỉ mới cho phép Camera có thể chọn nhanh từ Lock screen.[2]

Ice Cream Sandwich hoạt động mượt mà, nhanh và đẹp hơn.

## 8. Android 4.1 Jelly Bean



Ra mắt: 9-7-2012.

Máy tính bảng Nexus 7, sản phẩm hợp tác giữa Google và Asus, là thiết bị dùng Jelly Bean đầu tiên ra mắt. Android 4.1 nâng tầm hoạt động cho hệ điều hành của Google, trở thành hệ điều hành cho thiết bị di động hàng đầu hiện nay, đe dọa cả "ông lớn" Windows.[2]

Khả năng sắp xếp giao diện chủ và widget trong Jelly Bean rất tùy biến và linh hoạt. Hệ thống hỗ trợ dịch vụ ví điện tử Google Wallet, đặc biệt trình duyệt web mặc định trong Android được thay thế bởi đại diện tên tuổi: Chrome, với khả năng đồng bộ dữ liệu theo tài khoản với bản Chrome trên máy tính [2]

### 1.2. Giao diện

Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo giãn và thu lại để xử lý các đối tượng trên màn hình.



Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop trên máy tính để bàn. Màn hình chính Android thường gồm nhiều biểu tượng(icon) và tiện ích(widget). Giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích.

Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi “chủ đề” của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như Windows Phone hay iOS. Phần lớn những nhà sản xuất và một số nhà mạng, thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh.



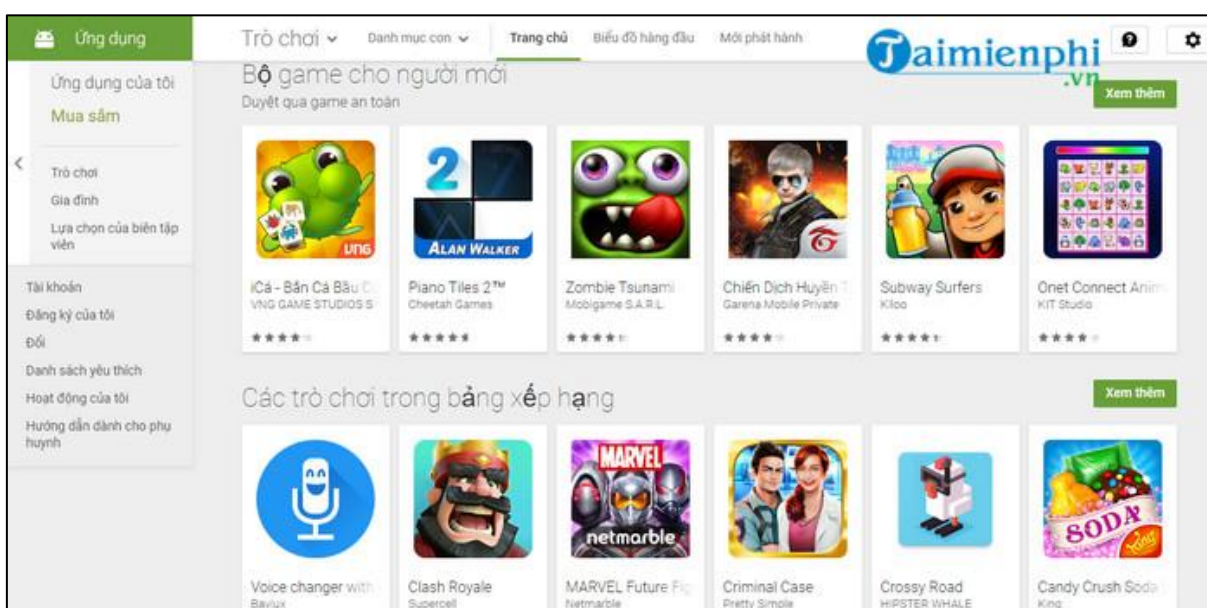
Hình 1.2.1: Một vài giao diện người dùng tùy biến của Android

Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Thanh trạng thái này có thể “kéo” xuống để xem màn hình thông báo gồm thông tin quan trọng hoặc cập nhật của các ứng dụng, như email hay tin nhắn SMS mới nhận, mà không làm gián đoạn hoặc khiến người dùng cảm thấy bất tiện. Trong các phiên bản đời đầu, người dùng có thể nhấn vào thông báo để mở ra ứng dụng tương ứng, về sau này các thông tin cập nhật được bổ sung thêm tính năng, như có khả năng lập tức gọi ngược lại khi có cuộc gọi nhỡ mà không cần mở ứng dụng gọi điện. Thông báo sẽ luôn nằm trong đó cho đến khi người dùng đã đọc hoặc xóa nó đi[1].



### 1.3. Ứng dụng

Android có lượng ứng dụng của bên thứ ba ngày càng nhiều, được chọn lọc và đặt tên một cửa hàng ứng dụng như Google Play hay Amazon Appstore để người dùng lấy về, hoặc bằng cách tải xuống rồi cài đặt tập tin APK từ trang web khác. Các ứng dụng trên Play Store cho phép người dùng duyệt, tải về và cập nhật các ứng dụng do Google và các nhà phát triển thứ ba phát hành. Play Store được cài đặt sẵn trên các thiết bị thỏa mãn điều kiện tương thích của Google. Ứng dụng sẽ tự động lọc ra một danh sách các ứng dụng tương thích với thiết bị của người dùng, và nhà phát triển có thể giới hạn ứng dụng của họ chỉ dành cho những nhà mạng cố định hoặc những quốc gia cố định vì lý do kinh doanh

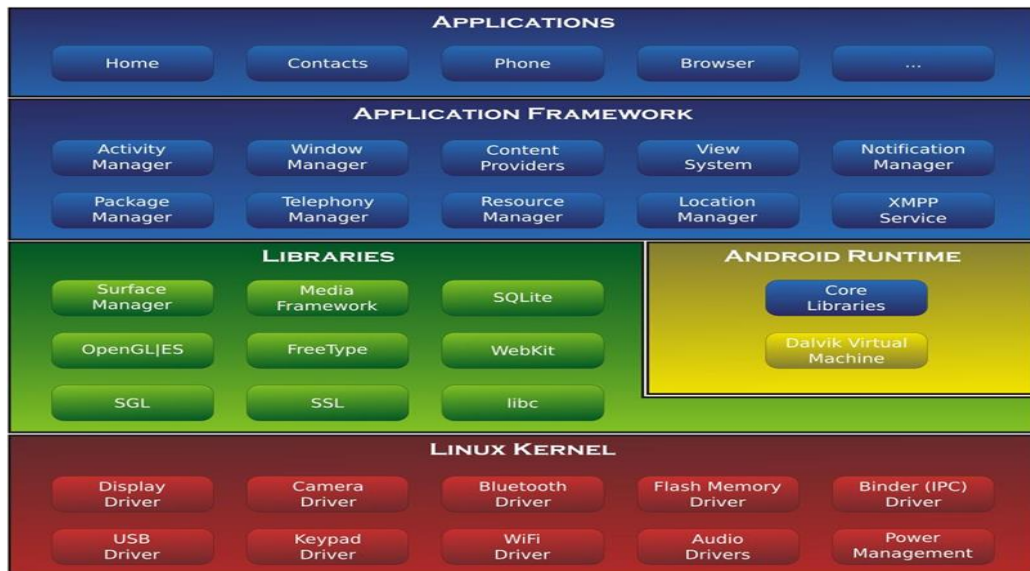


Hình 1.3.1: Kho ứng dụng

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng bộ phát triển phần mềm Android (SDK). SDK bao gồm một bộ đầy đủ các công cụ dùng để phát triển, gồm có công cụ gỡ lỗi, thư viện phần mềm, bộ giả lập điện thoại dựa trên QEMU, tài liệu hướng dẫn, mã nguồn mẫu, và hướng dẫn từng bước. Môi trường phát triển tích hợp được hỗ trợ chính thức là Android Studio, một môi trường đồ họa cho những nhà lập trình mới bắt đầu, và nhiều nền tảng ứng dụng di động đa nền tảng phong phú[1]

### 1.4. Kiến trúc hệ điều hành Android

Hệ điều hành Android có 4 tầng từ dưới lên trên là tầng hạt nhân Linux (phiên bản 2.6), tầng thư viện & tầng thực thi các ứng dụng trên hệ điều hành Android, tầng khung ứng dụng và trên cùng là tầng ứng dụng



Hình 1.4.1: Kiến trúc hệ điều hành Android

### 1.4.1. Tầng hạt nhân Linux

Hệ điều hành Android được phát triển dựa trên hạt nhân Linux, cụ thể là hạt nhân Linux phiên bản 2.6, điều đó được thể hiện ở lớp dưới cùng này. Tất cả mọi hoạt động của điện thoại muốn thi hành được thì đều được thực hiện ở mức cấp thấp ở lớp này bao gồm quản lý bộ nhớ (memory management), giao tiếp với phần cứng (driver model), thực hiện bảo mật (security), quản lý tiến trình (process).

Tuy được phát triển dựa vào nhân Linux nhưng thực ra nhân Linux đã được nâng cấp và sửa đổi rất nhiều để phù hợp với tính chất của những thiết bị cầm tay như hạn chế về bộ vi xử lý, dung lượng bộ nhớ, kích thước màn hình, nhu cầu kết nối mạng không dây...

Tầng này có các thành phần chủ yếu:

- Display Driver: Điều khiển việc hiển thị lên màn hình cũng như thu thập những điều khiển của người dùng lên màn hình (di chuyển, cảm ứng)
- Camera Driver: Điều khiển hoạt động camera, nhận luồng dữ liệu từ camera trả về.
- Bluetooth Driver: Điều khiển thiết bị phát và thu sóng Bluetooth.
- USB Driver: Quản lý hoạt động của các cổng giao tiếp USB.
- Keypad Driver: Điều khiển bàn phím.
- Wi-Fi Driver: Chịu trách nhiệm về việc thu phát sóng Wi-Fi.
- Audio Driver: Điều khiển các bộ thu phát âm thanh, giải mã các tín hiệu dạng audio thành tín hiệu số và ngược lại.

- Binder IPC Driver: Chịu trách nhiệm về việc kết nối và liên lạc với mạng vô tuyến như CDMA, GSM, 3G, 4G, E để đảm bảo những chức năng truyền thông được thực hiện.
- M-system Driver: Quản lý việc đọc ghi... lên các thiết bị nhớ như thẻ SD, flash.
- Power Management: Giám sát việc tiêu thụ điện năng



Hình 1.4.1.1. Tầng Linux

### 1.4.2. Tầng thư viện

Phần này có nhiều thư viện được viết bằng C/C++ để các phần mềm có thể sử dụng, các thư viện đó được tập hợp thành một số nhóm như:

- Thư viện hệ thống (System C library): thư viện dựa trên chuẩn C, được sử dụng chỉ bởi hệ điều hành.
- Thư viện Media (Media Libraries): Có nhiều code để hỗ trợ việc phát và ghi các loại định dạng âm thanh, hình ảnh, video thông dụng.
- Thư viện web (LibWebCore): Đây là thành phần để xem nội dung trên web, được sử dụng để xây dựng phần mềm duyệt web (Android Browse) cũng như để các ứng dụng khác có thể nhúng vào. Nó cực kỳ mạnh, hỗ trợ được nhiều công nghệ mạnh mẽ như HTML5, JavaScript, CSS, DOM, AJAX.
- Thư viện SQLite: Hệ cơ sở dữ liệu để các ứng dụng có thể sử dụng.



Hình 1.4.2.1. Tầng thư viện

### 1.4.3. Tầng thực thi các ứng dụng trên hệ điều hành Android

Phần này chứa các thư viện mà một chương trình viết bằng ngôn ngữ Java có thể hoạt động. Phần này có 2 bộ phận tương tự như mô hình chạy Java trên máy tính thường.

- Thứ nhất các thư viện lõi (Core Library), chứa các lớp như JAVA IO, collections, File Access.
- Thứ hai là một máy ảo java (Dalvik Virtual Machine).

Mặc dù cũng được viết từ ngôn ngữ Java nhưng một ứng dụng Java của hệ điều hành Android không được chạy bằng JRE của Sun (nay là Oracle) (JVM) mà là chạy bằng máy ảo Dalvik do Google phát triển.



Hình 1.4.3.1. Tầng thực thi các ứng dụng trên hệ điều hành

### 1.4.4. Tầng khung ứng dụng

Tầng này xây dựng bộ công cụ - các phần tử ở mức cao để các lập trình viên có thể nhanh chóng xây dựng ứng dụng. Nó được viết bằng Java, có khả năng sử dụng chung để tiết kiệm tài nguyên.

Đây là một nền tảng mở, điều đó có 2 điều lợi:

- Với các hãng sản xuất điện thoại: Có thể tùy biến để phù hợp với cấu hình điện thoại mà họ sản xuất cũng như để có nhiều mẫu mã, style hợp thị hiếu người dùng. Vì thế nên tuy cùng chung nền tảng android mà điện thoại của Google có thể khác hẳn với Motorola, HTC, T-Mobile, Samsung.
- Với lập trình viên: Cho phép lập trình viên có thể sử dụng các API ở tầng trên mà không cần phải hiểu rõ cấu trúc bên dưới, tạo điều kiện cho lập trình viên tự do sáng tạo bởi vì chỉ cần quan tâm đến nội dung mà ứng dụng họ làm việc.

Một tập hợp API rất hữu ích được xây dựng sẵn như hệ thống định vị, các dịch vụ chạy nền, liên lạc giữa các ứng dụng, các thành phần giao diện cấp cao.

Giới thiệu một số thành phần của phần này:

- Activity Manager: Quản lý các chu kỳ sống của một ứng dụng cũng như cung cấp công cụ điều khiển các Activity.
- Telephony Manager: Cung cấp công cụ để thực hiện việc liên lạc như gọi điện thoại.
- XMPP Service: Cung cấp công cụ để liên lạc trong thời gian thực.
- Location Manager: Cho phép xác định vị trí của điện thoại dựa vào hệ thống định vị toàn cầu GPS và Google Maps.
- Window Manager: Quản lý việc xây dựng và hiển thị các giao diện người dùng cũng như tổ chức quản lý các giao diện giữa các ứng dụng.
- Notification Manager: Quản lý việc hiển thị các thông báo (như báo có tin nhắn, có email mới).
- Resource Manager: Quản lý tài nguyên tĩnh của các ứng dụng bao gồm các file hình ảnh, âm thanh, layout, string.



Hình 1.4.4.1. Tầng khung ứng dụng

#### 1.4.5. Tầng ứng dụng

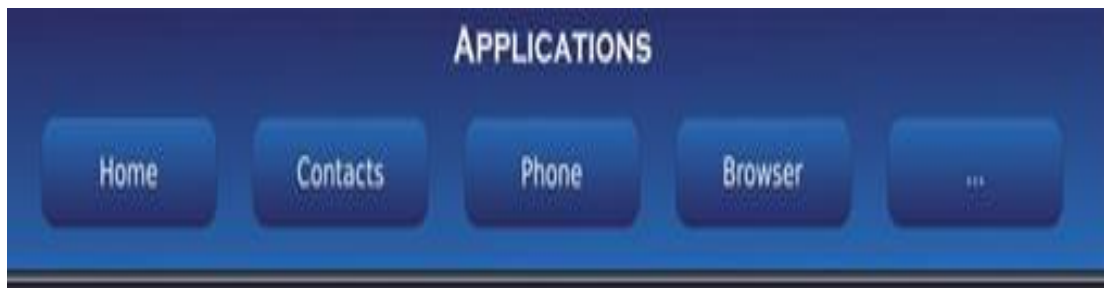
Đây là lớp ứng dụng giao tiếp với người dùng, bao gồm các ứng dụng như:

- Các ứng dụng cơ bản, được cài đặt đi liền với hệ điều hành là gọi điện (Phone), quản lý danh bạ (Contacts), duyệt web (Browser), nhắn tin (SMS), lịch làm việc (Calendar), đọc e-mail (Email-Client), bản đồ (Map), quay phim chụp ảnh (Camera)...
- Các ứng dụng được cài thêm như các phần mềm chứng khoán (Stock), các trò chơi (Game), từ điển...

Các chương trình có các đặc điểm là:



- Viết bằng Java, phần mở rộng là apk.
- Khi mỗi ứng dụng chạy, nó có một phiên bản Virtual Machine được dựng lên để phục vụ cho nó. Nó có thể là một Active Program: Chương trình có giao diện với người sử dụng hoặc là một background: chương trình chạy nền hay là dịch vụ.
- Android là hệ điều hành đa nhiệm, điều đó có nghĩa là trong cùng một thời điểm, có thể có nhiều chương trình cùng chạy một lúc, tuy nhiên với mỗi ứng dụng thì có duy nhất một thực thể (instance) được phép chạy mà thôi. Điều đó có tác dụng hạn chế sự lạm dụng tài nguyên, giúp hệ thống hoạt động tốt hơn.
- Các ứng dụng được gán số ID của người sử dụng nhằm phân định quyền hạn khi sử dụng tài nguyên, cấu hình phần cứng và hệ thống.
- Android là một hệ điều hành có tính mở, khác với nhiều hệ điều hành di động khác, Android cho phép một ứng dụng của bên thứ ba được phép chạy nền. Các ứng dụng đó chỉ có một hạn chế nhỏ đó là nó không được phép sử dụng quá 5~10% công suất CPU, điều đó nhằm để tránh độc quyền trong việc sử dụng CPU.
- Ứng dụng không có điểm vào cố định, không có phương thức main để bắt đầu [1].



Hình 1.4.5.1. Tầng ứng dụng

### 1.5. Quản lý bộ nhớ

Android được thiết kế để quản lý bộ nhớ (RAM) để giảm tối đa tiêu thụ điện năng, trái với hệ điều hành máy tính để bàn luôn cho rằng máy tính sẽ có nguồn điện không giới hạn. Khi một ứng dụng Android không còn được sử dụng, hệ thống sẽ tự động ngưng nó trong bộ nhớ - trong khi ứng dụng về mặt kỹ thuật vẫn “mở”, những ứng dụng này sẽ không tiêu thụ bất cứ tài nguyên nào (như năng lượng pin hay năng lượng xử lý) và nằm đó cho đến khi nó được cần đến. Cách làm như vậy có lợi kép là

vừa làm tăng khả năng phản hồi nói chung của thiết bị Android, vì ứng dụng không nhất thiết phải đóng rồi mở lại từ đầu, vừa đảm bảo các ứng dụng nền không làm tiêu hao năng lượng một cách không cần thiết.

Android quản lý các ứng dụng trong bộ nhớ một cách tự động: khi bộ nhớ thấp, hệ thống sẽ bắt đầu diệt ứng dụng và tiến trình không hoạt động được một thời gian, sắp theo thời điểm cuối mà chúng được sử dụng (tức là cũ nhất sẽ bị tắt trước). Tiến trình này được thiết kế ẩn đi với người dùng, để người dùng không cần phải quản lý bộ nhớ hoặc tự tay tắt các ứng dụng. Tuy nhiên, sự che giấu này của hệ thống quản lý bộ nhớ Android đã dẫn đến sự thịnh hành của các ứng dụng tắt chương trình của bên thứ ba trên của hàng Google Play; những ứng dụng kiểu như vậy được cho là có hại nhiều hơn có lợi. [1]

## **1.6. Cập nhật**

Google đưa ra các bản cập nhật lớn cho Android theo chu kỳ từ 6 đến 9 tháng, mà phần lớn thiết bị đều có thể nhận được qua sóng không dây. Bản cập nhật lớn mới nhất là Android 9.0 Pie.

So với các hệ điều hành cạnh tranh khác, Như iOS, các bản cập nhật Android thường mất thời gian lâu hơn để đến với các thiết bị, Với những thiết bị không thuộc dòng Nexus và Pixel, các bản cập nhật thường đến sau vài tháng kể từ khi phiên bản được chính thức phát hành. Nguyên nhân của việc này một phần do sự phong phú về phần cứng của các thiết bị Android, nên người ta phải mất thời gian điều chỉnh bản cập nhật cho phù hợp, vì mã nguồn chính thức của Google chỉ chạy được trên những thiết bị Nexus chủ lực của họ. Chuyển Android sang những phần cứng cụ thể là một quy trình tốn thời gian và công sức của các nhà sản xuất thiết bị, những người luôn ưu tiên các thiết bị mới nhất và thường bỏ rơi các thiết bị cũ hơn. Do đó, những chiếc điện thoại thông minh thế hệ cũ thường không được cập nhật nếu nhà sản xuất quyết định rằng nó không đáng để bỏ thời gian, bất kể chiếc điện thoại đó có khả năng chạy bản cập nhật hay không. Vấn đề này còn trầm trọng hơn khi những nhà sản xuất điều chỉnh Android để đưa giao diện và ứng dụng của họ vào, những thứ này cũng sẽ phải làm lại cho mỗi bản cập nhật. Sự chậm trễ còn được đóng góp bởi nhà mạng, sau khi nhận được bản cập nhật từ nhà sản xuất, họ còn điều chỉnh thêm cho phù hợp với nhu cầu rồi thử nghiệm kỹ lưỡng trên hệ thống mạng của họ trước khi chuyển nó đến người dùng [1]

## **1.7. Bảo mật và riêng tư**

Các ứng dụng Android chạy trong một “sandbox”, là một vực riêng rẽ với hệ thống và không được tiếp cận đến phần còn lại của tài nguyên hệ thống, trừ khi nó được người dùng trao quyền truy cập một cách công khai khi cài đặt. Trước khi cài đặt ứng dụng, cửa hàng Play sẽ hiển thị tất cả các quyền mà ứng dụng đòi hỏi: ví dụ như một trò chơi cần phải kích hoạt bộ rung hoặc dữ liệu vào thẻ nhớ SD, nhưng nó không nên cần quyền đọc tin nhắn SMS hoặc tiếp cận danh bạ điện thoại. Sau khi xem xét các quyền này, người dùng có thể chọn đồng ý hoặc từ chối chúng, ứng dụng chỉ được cài đặt khi người dùng đồng ý.

Điện thoại thông minh Android có khả năng báo cáo vị trí của điểm truy cập Wi-Fi, phát hiện ra việc di chuyển của người dùng điện thoại, để xây dựng những cơ sở dữ liệu có chứa vị trí của hàng trăm triệu điểm truy cập. Những cơ sở dữ liệu này tạo nên một bản đồ điện tử để tìm vị trí điện thoại thông minh, cho phép chúng chạy các ứng dụng như Google Latitude, Facebook Places, và gửi những đoạn quảng cáo dựa trên vị trí [1]

## **1.8. Ưu Nhược điểm**

### **1.8.1. Ưu điểm**

Là hệ điều hành có mã nguồn mở nên khả năng tùy biến cao, có thể tùy ý chỉnh sửa mà không có sự can thiệp hay cấm cản từ Google.

Đa dạng sản phẩm, rất nhiều hãng điện thoại, thiết bị công nghệ đã ưu ái chọn Android cho thiết bị của họ, giá cả thì hợp lý từ bình dân đến cao cấp.

Kho ứng dụng Google Play Store đồ sộ.

Thân thiện và dễ sử dụng.

Khả năng đa nhiệm, chạy cùng lúc nhiều ứng dụng cao.

### **1.8.2. Nhược điểm**

Dễ nhiễm phần mềm độc hại và virus. Do tính chất mã nguồn mở, nhiều phần mềm không được kiểm soát có chất lượng không tốt hoặc lỗi bảo mật vẫn được sử dụng.

Kho ứng dụng quá nhiều dẫn đến khó kiểm soát chất lượng, thiếu các ứng dụng thật sự tốt.



Sự phân mảnh lớn. Trong khi một số thiết bị Android xuất sắc đã trình làng như Galaxy S5, Galaxy Note 4, Xperia Z3..., vẫn còn rất nhiều sản phẩm giá rẻ bình thường khác.

Cập nhật không tự động với tất cả thiết bị. Khi một phiên bản hệ điều hành mới ra mắt, không phải tất cả sản phẩm đều được cập nhật, thậm chí nếu muốn trải nghiệm bạn thường xuyên phải mua mới thiết bị.

## **1.9. Thiết bị**

Ngoài điện thoại và máy tính bảng, do bản chất mở và cho phép thay đổi của Android giúp nó xuất hiện trên các thiết bị điện tử khác, như laptop, netbook, smartbook, TV thông minh và máy ảnh (Galaxy Camera). Hơn thế nữa, hệ điều hành Android còn được ứng dụng trong kính mắt thông minh (Project Glass), đồng hồ đeo tay, tai nghe, đầu CD và DVD cho xe hơi, máy nghe nhạc bỏ túi và máy trò chơi điện tử chạy Android.[1]

## **1.10 Kết luận**

Mới bước chân vào làng điện thoại di động nhưng Android đã lần lượt hạ gục những đối thủ tầm cỡ và trở thành hệ điều hành tăng trưởng nhanh nhất trên thế giới. Mỗi phiên bản Android luôn thu hút mỗi quan tâm lớn của giới công nghệ, các nhà sản xuất và người sử dụng. Thị phần điện thoại Android trên thị trường ngày càng tăng mạnh mẽ, phần nào đó chứng tỏ những ưu thế và tính năng vượt trội của nền tảng này, đồng thời khẳng định những nỗ lực của Android trong việc mạng đến cho người tiêu dùng những sản phẩm tốt nhất

## CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO

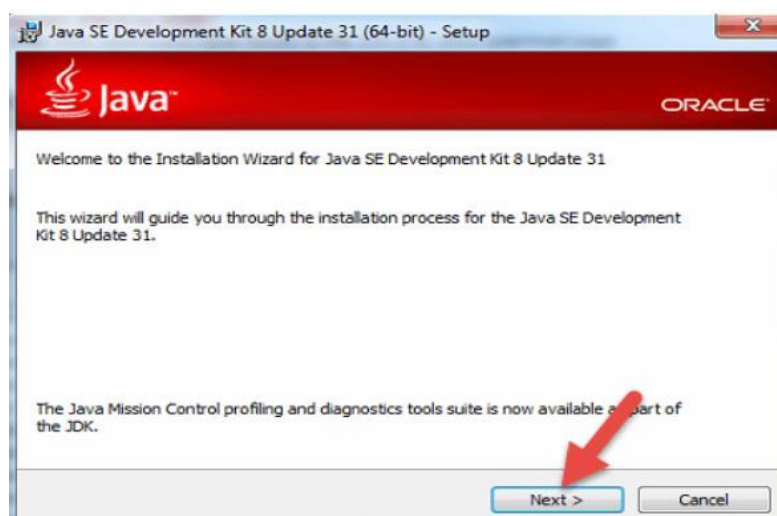
### 4.1. Thiết lập môi trường

Hai thành phần cơ bản nhất mà chúng ta cần phải có để lập trình Android là bộ JDK(Java Development Kit) và Android SDK (Software Development Kit). JDK dùng để tạo ra môi trường thực thi máy ảo cho hệ điều hành mà chúng ta sử dụng. Android SDK chứa các phiên bản Android, các hàm API cần thiết, source code minh họa cũng như các công cụ hỗ trợ lập trình khác. Mỗi khi Google ra phiên bản Android mới thì Android SDK cũng được cập nhật tương ứng.

#### 2.1.1. Cài đặt Java

Để cài đặt JDK, cần phải truy cập vào trang của Oracle JDK.

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>



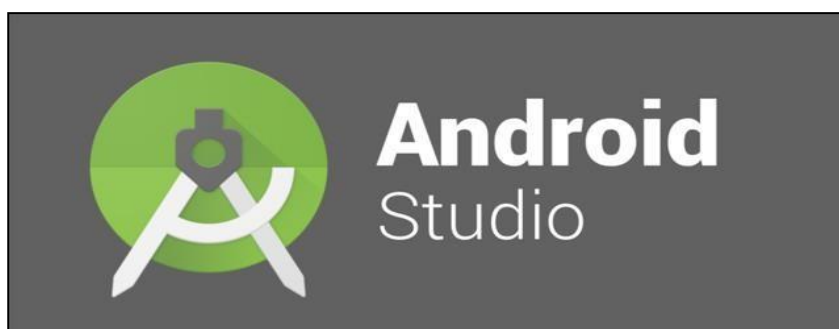
Hình 2.1.1.1: Cài đặt JDK

## 2.1.2. Cài đặt Android Studio

### 2.1.2.1. Phần mềm Android Studio

Có nhiều công cụ để phát triển Android nhưng đến nay công cụ chính thức và mạnh mẽ nhất là Android Studio. Đây là IDE (Môi trường phát triển tích hợp) chính thức cho nền tảng Android, được phát triển bởi Google và được sử dụng để tạo phần lớn các ứng dụng mà chúng ta có thể sử dụng hàng ngày. [3]

Android Studio lần đầu tiên được công bố tại hội nghị Google I/O vào năm 2013 và được phát hành cho công chúng vào năm 2014 sau nhiều phiên bản beta khác nhau. Trước khi được phát hành, các nhà phát triển Android thường sử dụng các công cụ như Eclipse IDE, một IDE Java chung cũng hỗ trợ nhiều ngôn ngữ lập trình khác [3]



Chức năng của Android Studio là cung cấp giao diện để tạo các ứng dụng và xử lý phần lớn các công cụ quản lý file phức tạp đằng sau hậu trường. Ngôn ngữ lập trình được sử dụng ở đây là Java và được cài đặt riêng trên thiết bị của chúng ta. Android Studio rất đơn giản, chúng ta chỉ cần viết, chỉnh sửa và lưu các dự án của mình và các file trong dự án đó. Đồng thời, Android Studio sẽ cấp quyền truy cập vào Android SDK. [3]

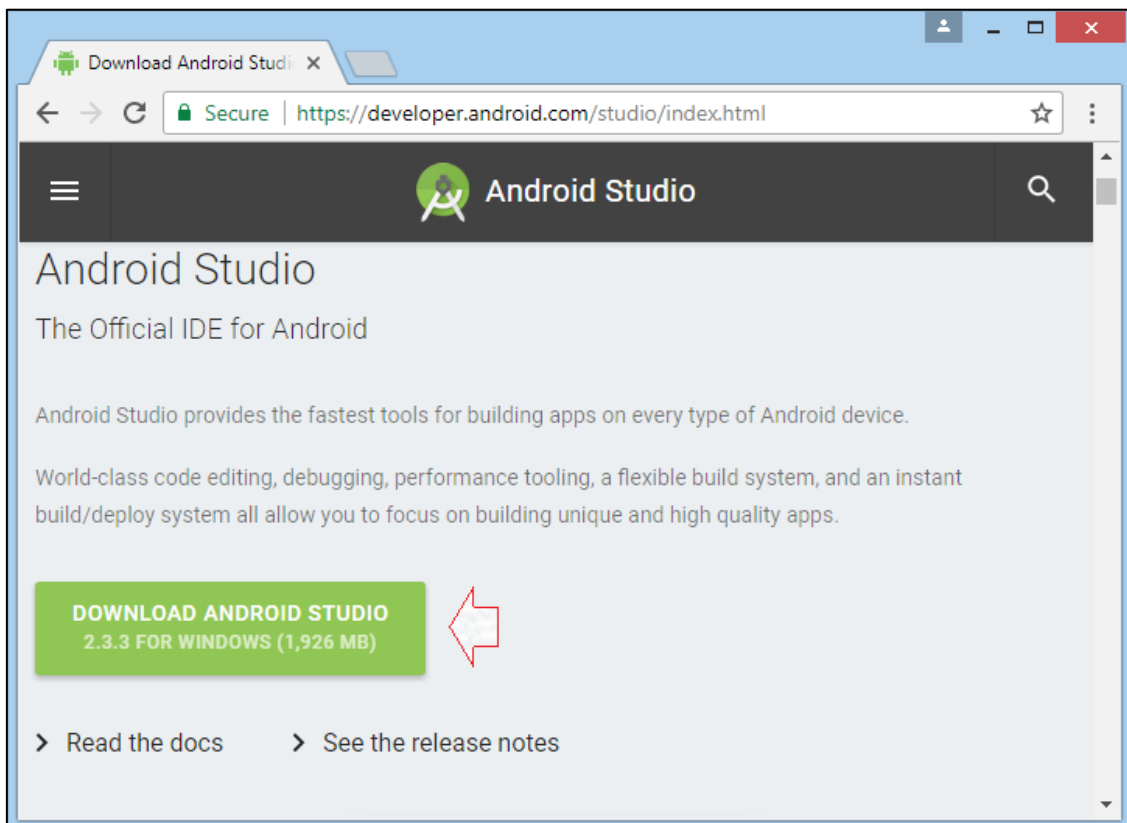
Hãy coi đây là đuôi cho code Java cho phép nó chạy trơn tru trên các thiết bị Android và tận dụng lợi thế của phần cứng gốc. Chúng ta cần sử dụng ngôn ngữ lập trình Java để viết các chương trình, Android SDK có nhiệm vụ kết nối các phần này lại với nhau. Cùng lúc đó Android Studio kích hoạt để chạy code, thông qua trình giả lập hoặc qua một phần cứng kết nối với thiết bị. Sau đó, chúng ta cũng có thể “gỡ rối” chương trình khi nó chạy và nhận phản hồi giải thích sự cố, v.v... để chúng ta có thể nhanh chóng giải quyết vấn đề.[3]

Google đã nỗ lực rất nhiều để làm cho Android Studio trở nên mạnh mẽ và hữu ích nhất có thể. Nó cung cấp những gợi ý trực tiếp trong khi viết code và thường đề

xuất những thay đổi cần thiết để sửa lỗi hoặc làm code hiệu quả hơn. Ví dụ, nếu không sử dụng biến, biến đó sẽ được tô đậm bằng màu xám. Và khi bắt đầu gõ một dòng code, Android Studio sẽ cung cấp danh sách gợi ý tự hoàn thành để giúp bạn hoàn thiện dòng code đó. Chức năng này rất hữu ích khi chúng ta không nhớ được chính xác cú pháp hoặc để tiết kiệm thời gian.[3]

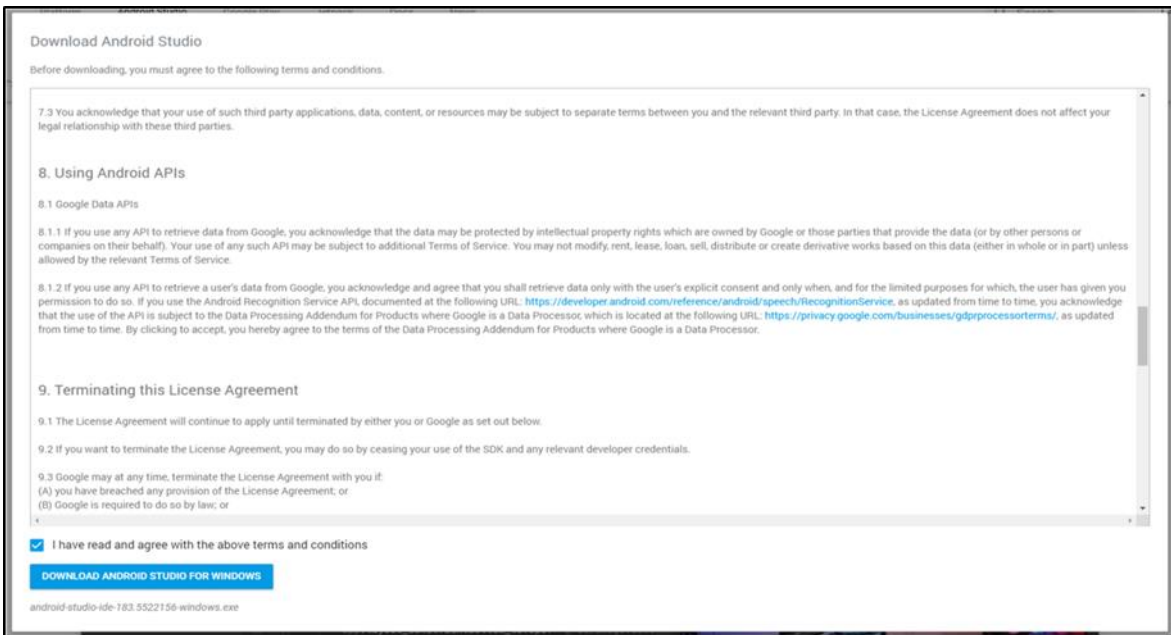
### 2.1.2.2. Hướng dẫn cài đặt Android Studio

Truy cập trang web <https://developer.android.com/studio> và nhấn nút **Download Android Studio**



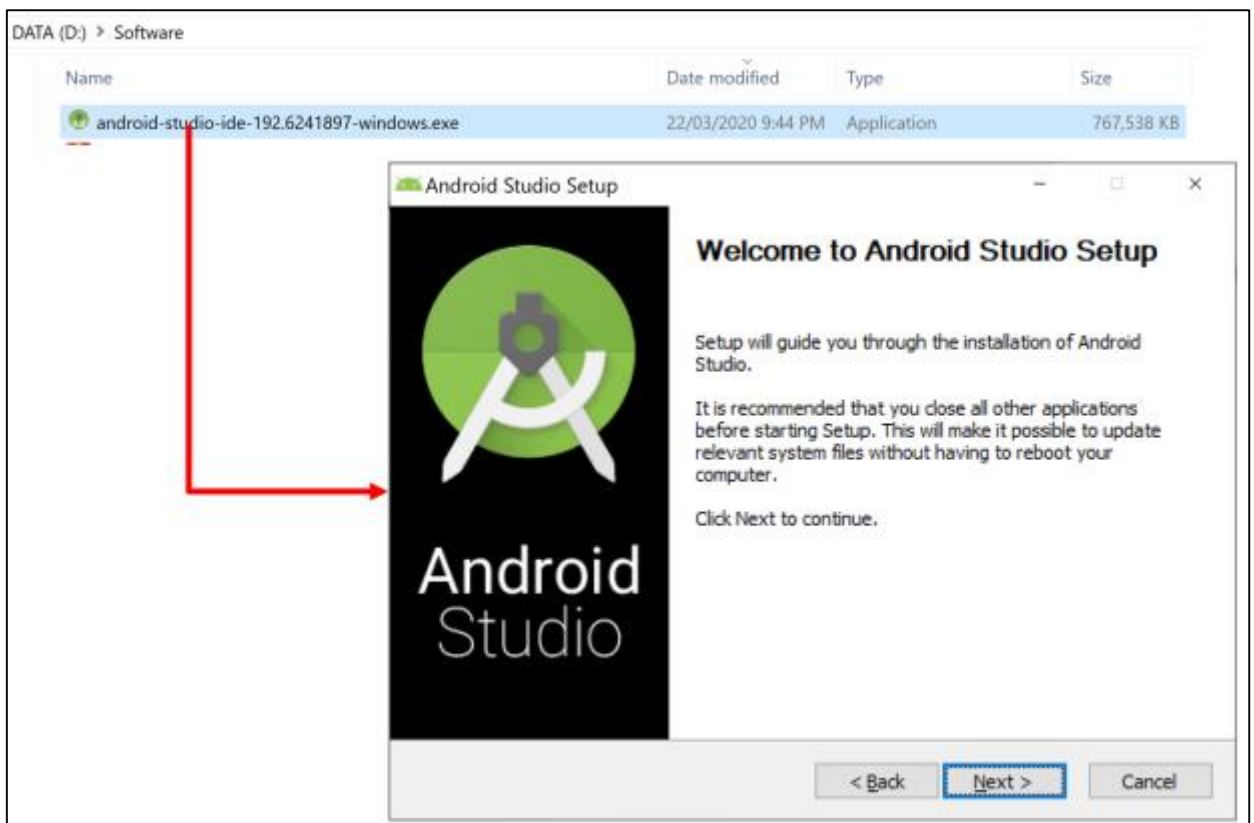
Hình 2.1.2.2.1: Trang download Android Studio

Tick vào ô “I have read and agree with the above terms and conditions” và nhấn nút **Download Android Studio for Windows** nếu có thông báo xuất hiện



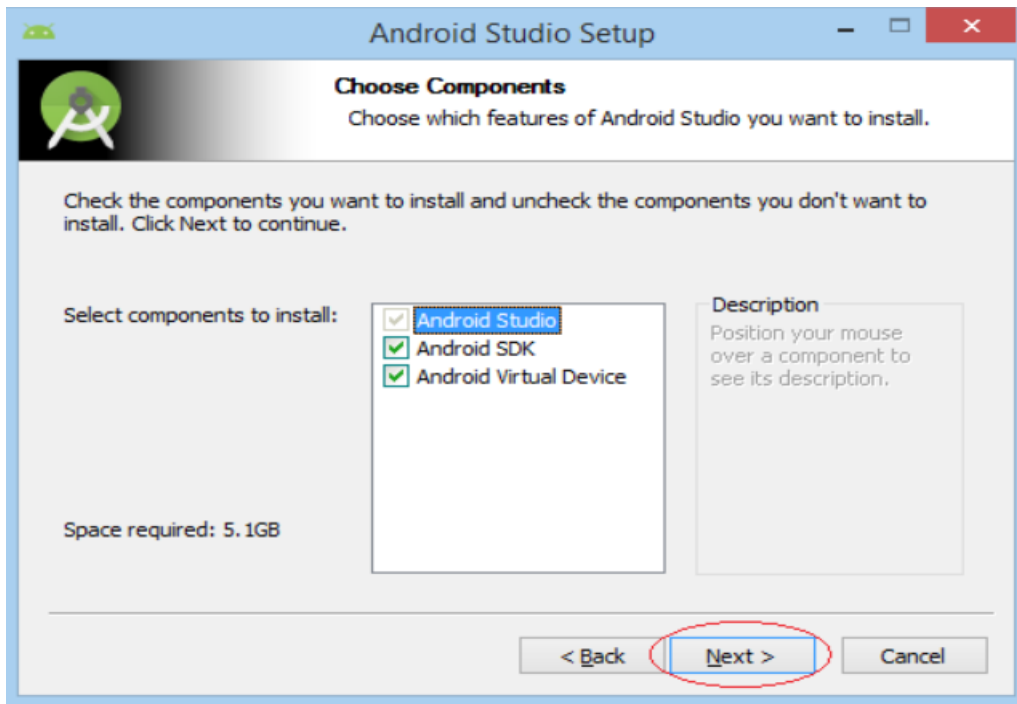
Hình 2.1.2.2.2. Xác nhận điều khoản sử dụng để có thể tải về

Cài đặt Android Studio bằng bộ cài vừa tải về



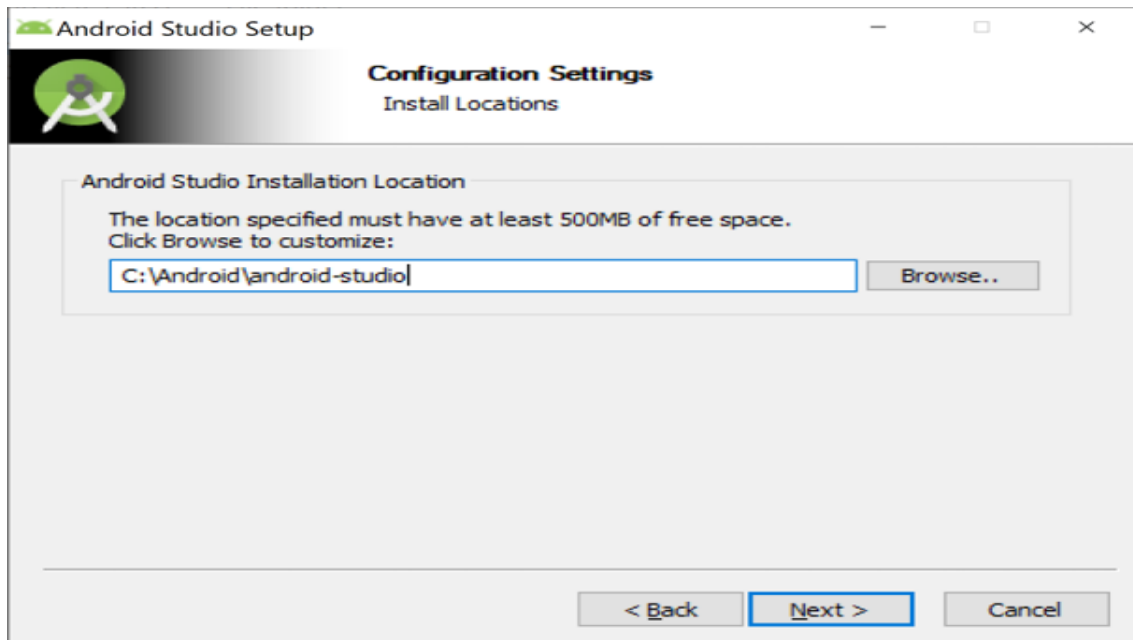
Hình 2.1.2.2.3: Cài đặt Android Studio bằng bộ cài vừa tải về

Khi cài đặt chú ý chọn cả SDK và trình giả lập thiết bị android ảo ADV như hình:

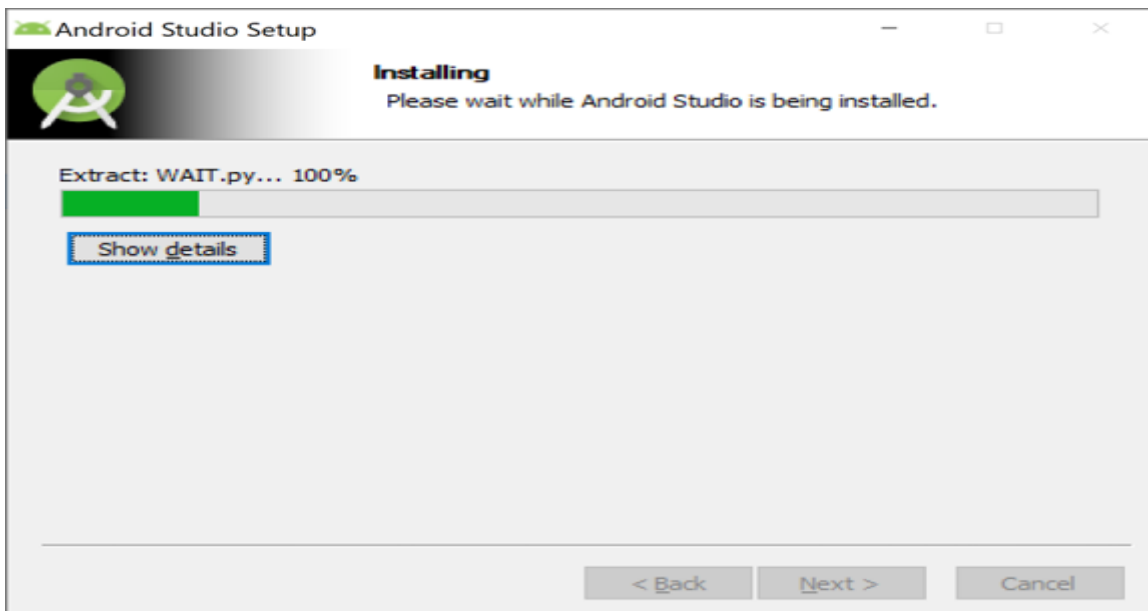
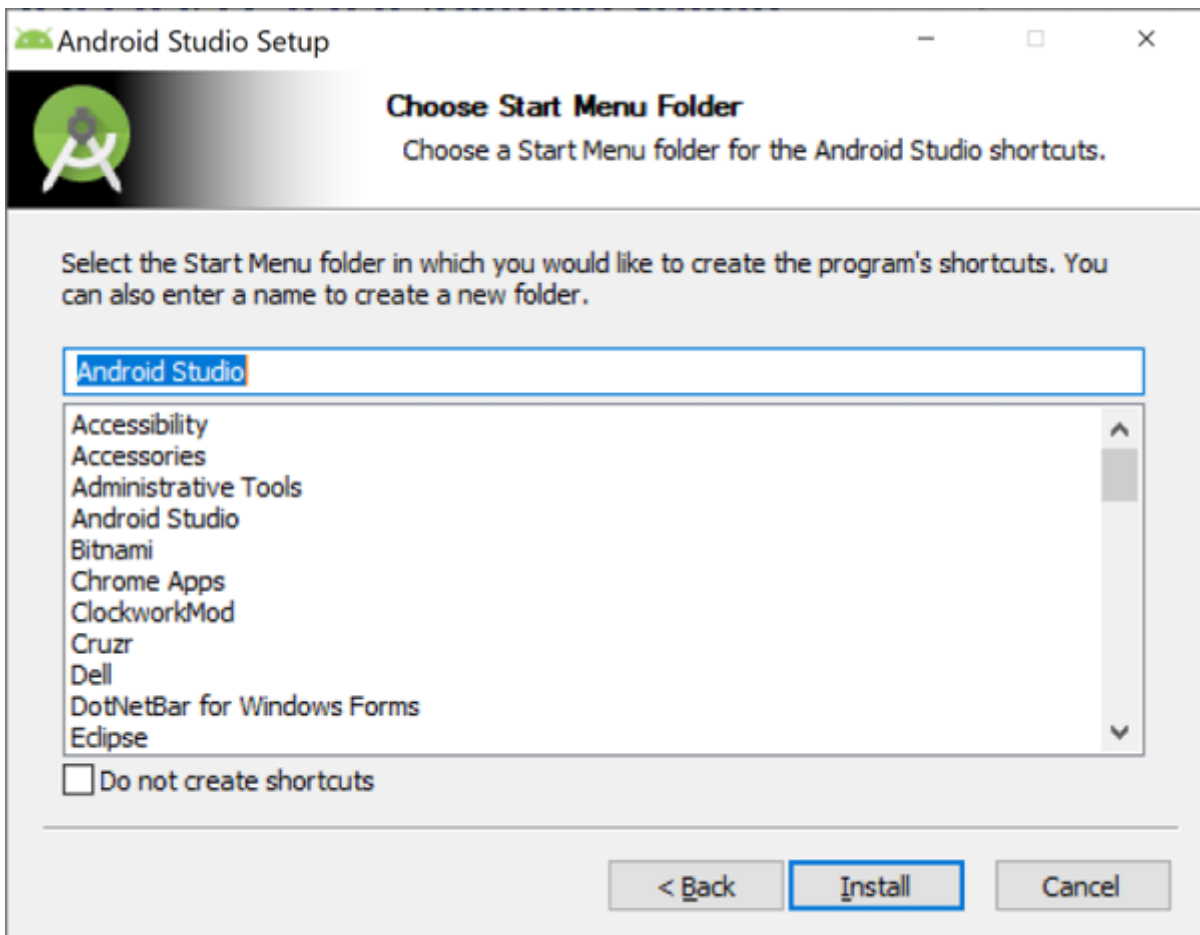


Hình 2.1.2.2.4 : Giao diện cài đặt SDK và AVD

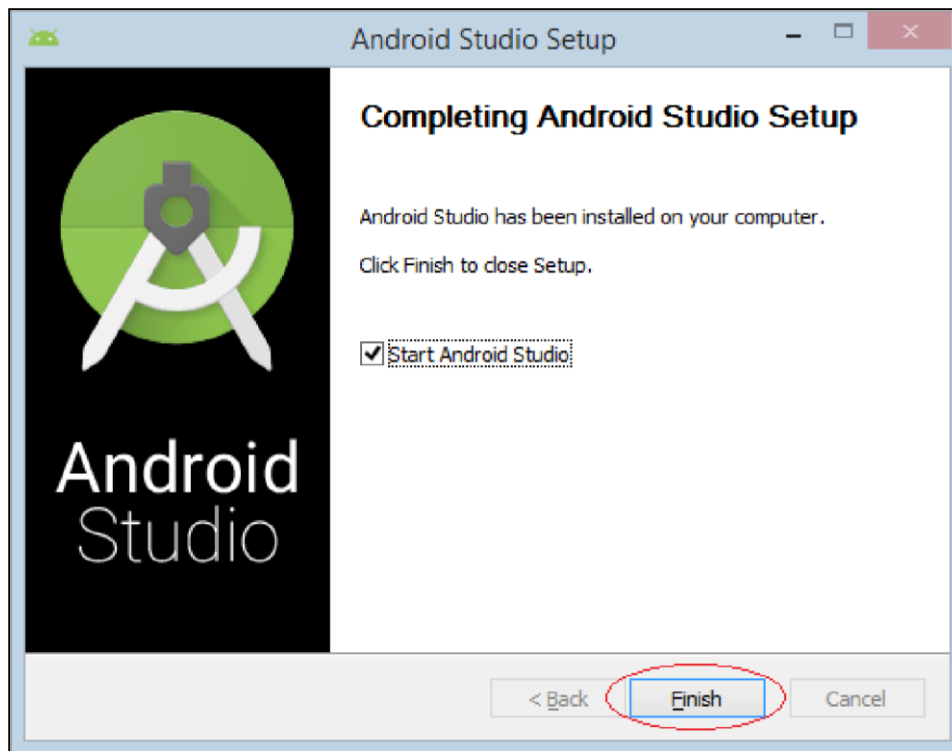
Ở màn hình trên lưu ý cài vào thư mục C:\Android\android-studio  
Sau đó nhấn Next để tiếp tục:



Nhấn “Install” để bắt đầu cài đặt



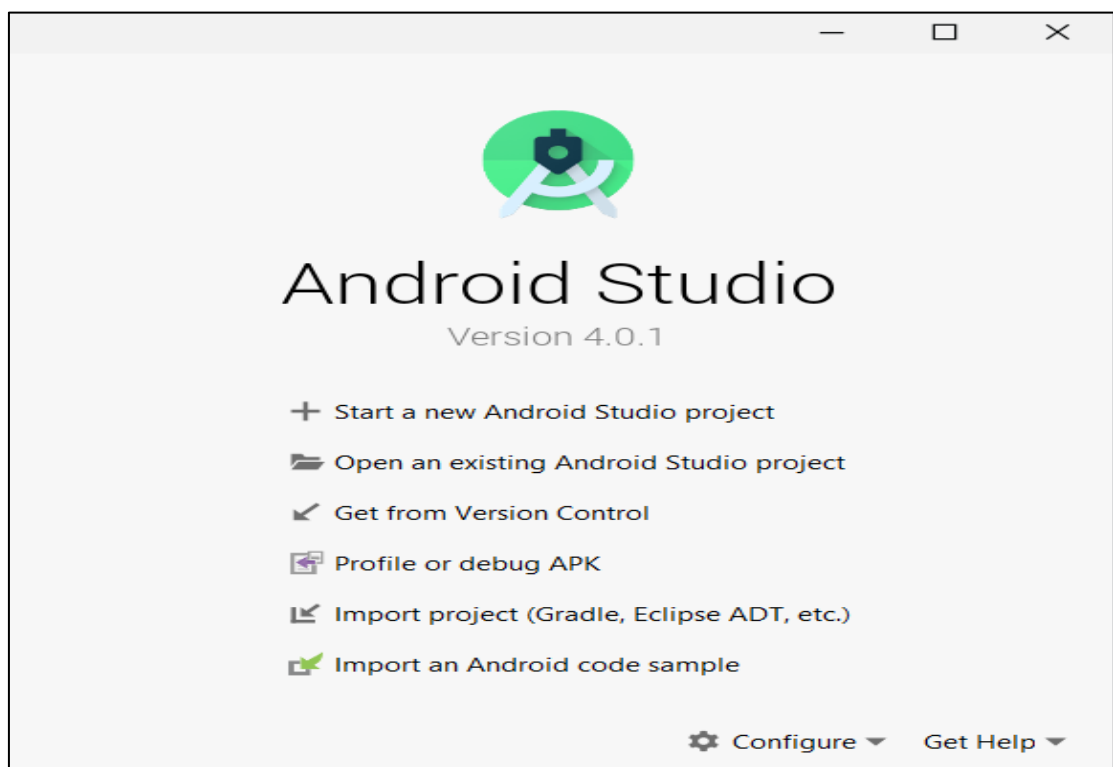
Khi việc cài đặt hoàn tất, chúng ta sẽ được như hình sau:



Hình 2.1.2.2.5. Cài đặt hoàn tất

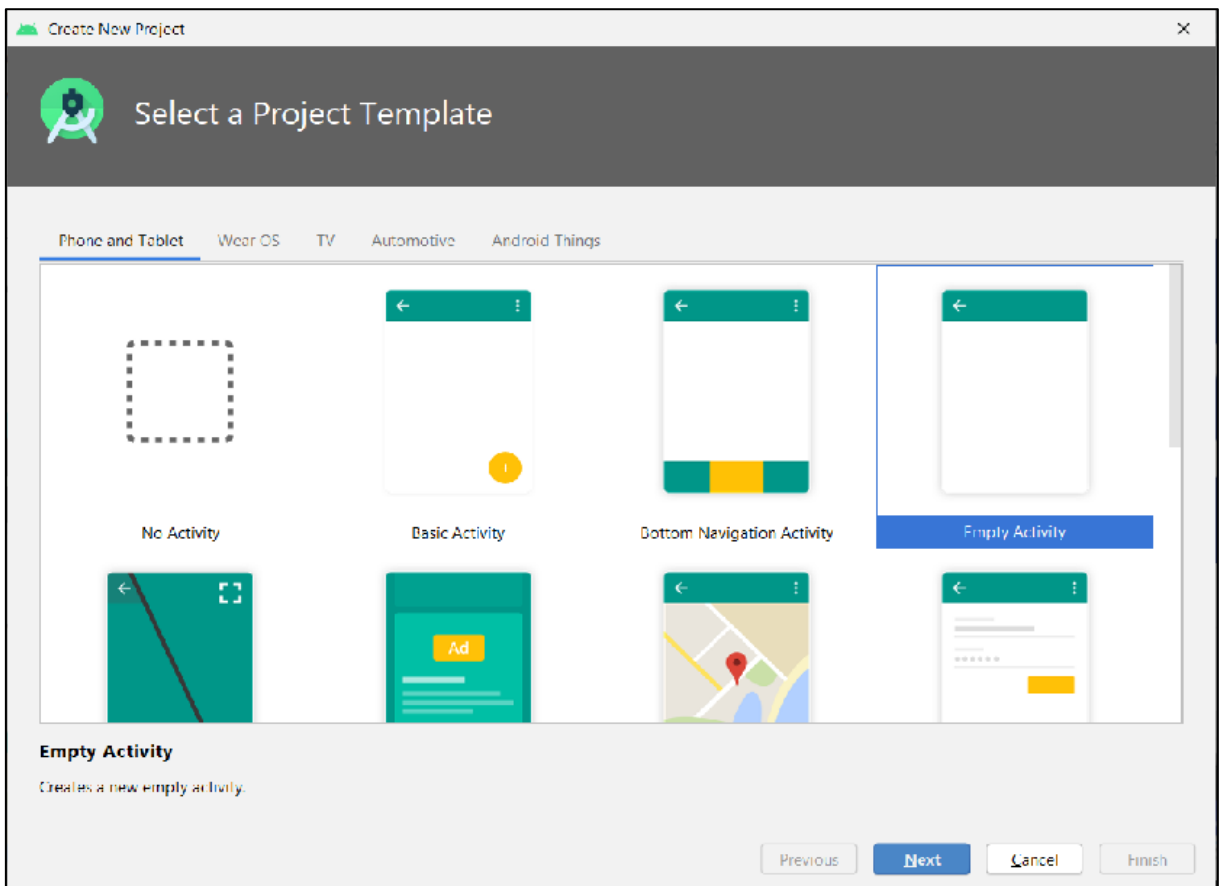
### 2.1.2.3. Tạo project trong Android Studio

Bước 1: Khởi động Android Studio → Chọn Start a new Android Studio project

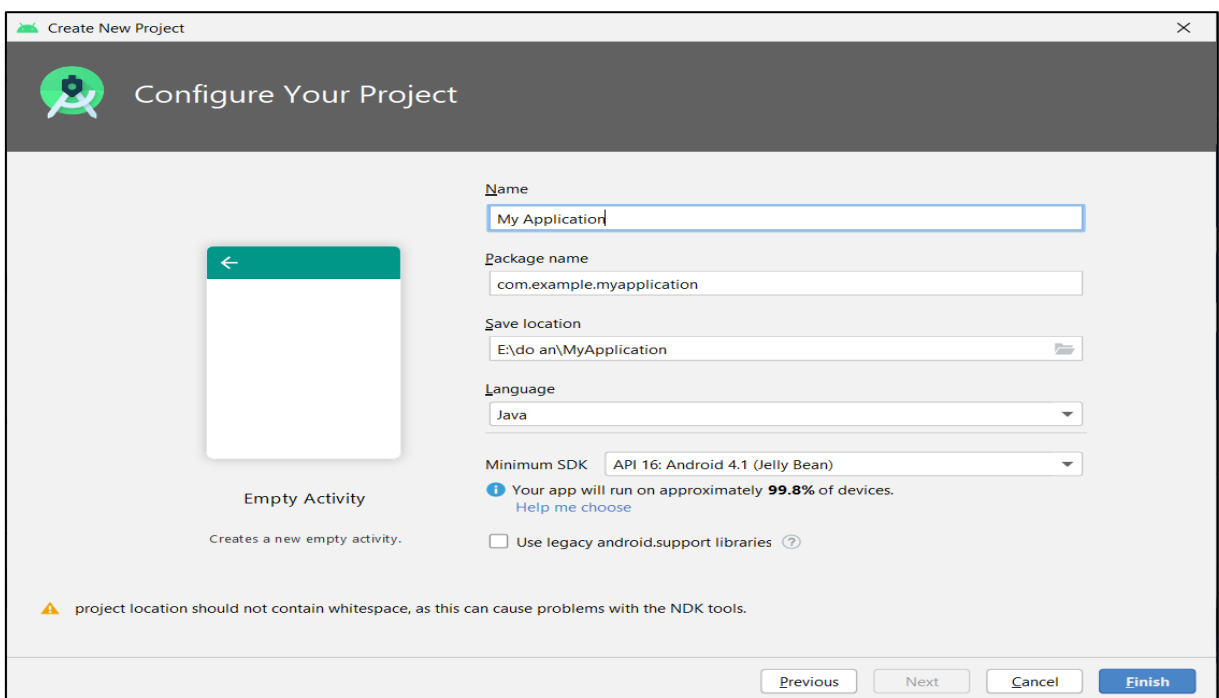


Bước 2: Trong Phone and Tablet → chọn **Empty Activity** → chọn **Next**

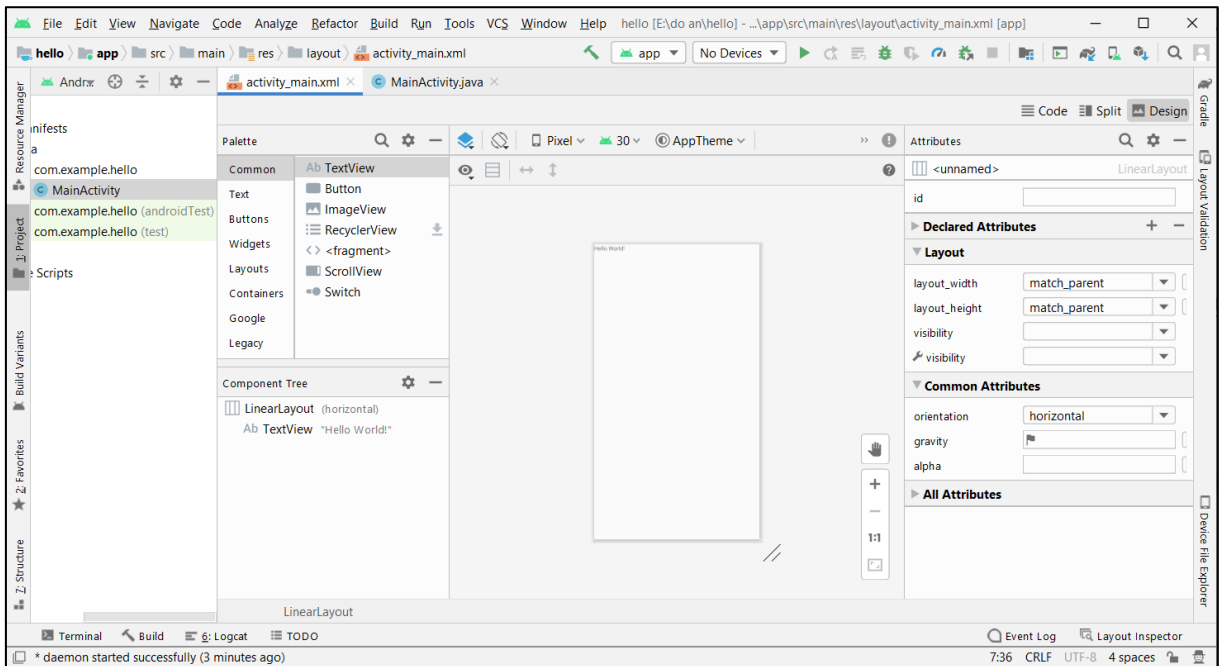




Bước 3 : nhập tên Activity tại Name → chọn Finish



Bước 4: Sau khi tạo thành công project, chúng ta sẽ nhìn thấy như hình bên dưới

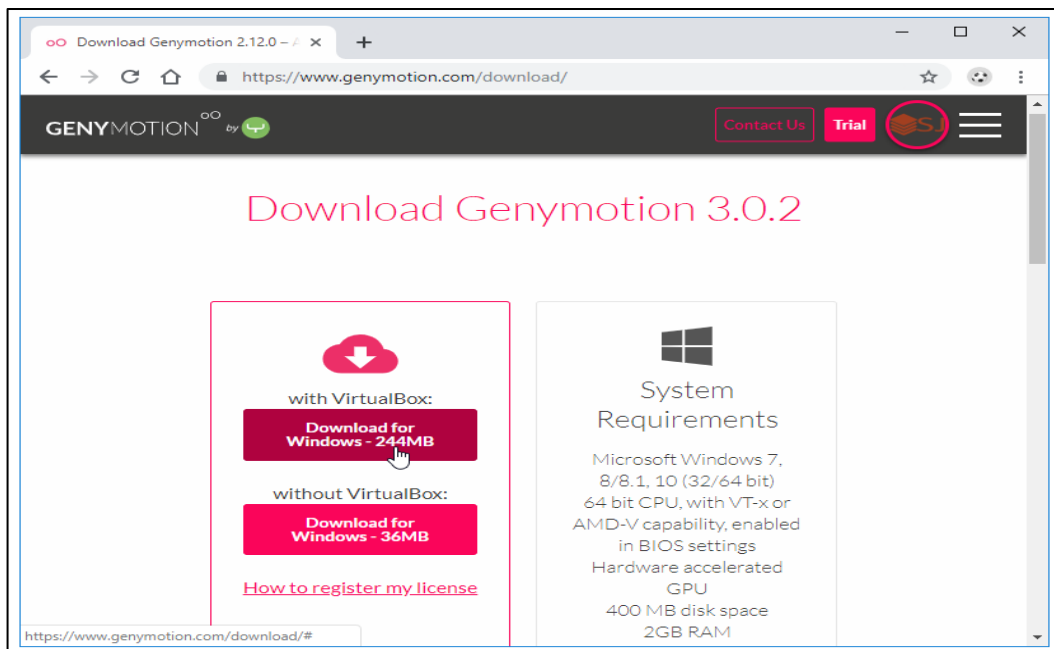


### 2.1.2.4. Thiết bị ảo trong Android Studio

Máy ảo Android là một phần không thể thiếu khi chúng ta lập trình ứng dụng cho hệ điều hành Android, nó giúp chúng ta chạy thử ứng dụng ngay trên máy tính. Và công cụ máy ảo tiện dụng hiện giờ là Genymotion.

Để cài đặt máy ảo Genymotion ta truy cập vào đường dẫn: <https://www.genymotion.com/download/>

Ta phải tạo một tài khoản rồi đăng nhập vào mới thấy được mục này:



Hình 2.1.2.3.1: Giao diện download máy ảo Genymotion

## 2.2. Các thành phần trong Android Studio

### 2.2.1. File manifests

Trong bất kỳ một dự án Android nào khi tạo ra đều có một file manifest, file này được dùng để đăng ký các màn hình sử dụng trong ứng dụng Android, các permission cũng như các theme cho ứng dụng. Đồng thời nó cũng chứa thông tin về phiên bản SDK cũng như main activity sẽ chạy đầu tiên.

File này được tự động sinh ra khi tạo một Android project.

Dưới đây là nội dung của một file AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.tong2so">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Tong 2 so"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Hình 2.2.1.1 : File Manifests

#### 2.2.1.1. Thành phần ứng dụng Application

Thẻ <application>, bên trong thẻ này chứa các thuộc tính được định nghĩa cho ứng dụng Android như:

- android:icon = "drawable resource". Ở đây đặt đường dẫn đến file icon của ứng dụng khi cài đặt. VD: android:icon = "@mipmap/ic\_launcher".
- android:name = "string". Thuộc tính này để đặt tên cho ứng dụng Android. Tên này sẽ được hiển thị lên màn hình sau khi cài đặt ứng dụng.
- android:theme = "drawable theme". Thuộc tính này để đặt theme cho ứng dụng. Các theme là các cách để hiển thị giao diện ứng dụng.
- Ngoài ra còn nhiều thuộc tính khác....

## 2.2.2. File Java

File java là một file tự động sinh ra ngay khi tạo ứng dụng, file này được sử dụng để quản lý thuộc tính được khai báo trong file XML của ứng dụng và các tài nguyên hình ảnh. Mã nguồn của file java được tự động sinh ra khi có bất kỳ một sự kiện này xảy ra làm thay đổi các thuộc tính trong ứng dụng.

Chẳng hạn như, kéo và thả một file hình ảnh từ bên ngoài vào project thì ngay lập tức thuộc tính đường dẫn đến file đó cũng sẽ được hình thành trong file java hoặc xóa một file hình ảnh thì đường dẫn tương ứng đến hình ảnh đó cũng tự động bị xóa

Dưới đây là ví dụ về nội dung của một file java:

```
package com.example.tong2so;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    EditText edt_s01, edt_s02, edt_ketqua;
    Button btn_tinh, btn_xoa, btn_thoat;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //bac cau cho bien
        edt_s01 = findViewById(R.id.edt_s01);
        edt_s02 = findViewById(R.id.edt_s02);
        edt_ketqua = findViewById(R.id.edt_ketqua);
        btn_tinh = findViewById(R.id.btn_tinh);
        btn_xoa = findViewById(R.id.btn_xoa);
        btn_thoat = findViewById(R.id.btn_thoat);

        //viet code bat su kien
        btn_tinh.setOnClickListener((view) -> {
            int s01, s02, s03, ketqua;
            s01 = Integer.parseInt(edt_s01.getText().toString());
            s02 = Integer.parseInt(edt_s02.getText().toString());
            ketqua = s01+s02;
            edt_ketqua.setText(ketqua+"");
        });
    }
}
```

Hình 2.2.2.1 : File Java

## 2.2.3. File tài nguyên

Chứa các tài nguyên mà ứng dụng sẽ sử dụng đến, nó tổ chức thành các thư mục con như:

- drawable/ : ở đây cơ bản lưu các đối tượng đồ họa như các ảnh dạng png, các ảnh dạng xml...
- layout/: lưu trữ các file xml biểu diễn về thành phần, bố cục của các thành phần hiển thị được trên màn hình.
- mipmap/: cũng để lưu các đối tượng hình ảnh, ví dụ icon ứng dụng ic\_launcher đặt ở đây.

- values/: chứa các file như color.xml, dimens.xml, string.xml, style.xml, đây là các file xml định nghĩa các giá trị có thể sử dụng trong ứng dụng như màu sắc, kích thước, các chuỗi, các theme ..

#### 2.2.4. File Grandle

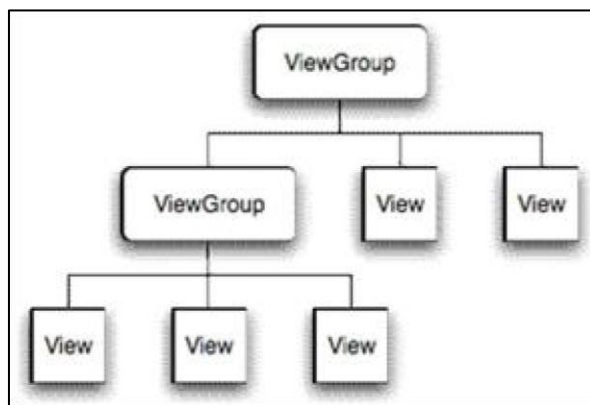
Gradle Scrips: Chứa nhiều nhánh con như build.gradle, local.properties, ... là nơi bạn thiết lập các thông số để Gradle build ứng dụng. Lưu ý Gradle là một công cụ tích hợp vào Android Studio, chức năng của nó build mã nguồn, kết hợp tài nguyên, phân tích xml ... rồi kết hợp chúng lại với nhau tạo ra ứng dụng chạy trên JVM.

#### 2.3.View

Trong một ứng dụng Android, giao diện người dùng được xây dựng từ các đối tượng View và ViewGroup. Có nhiều kiểu View và ViewGroup. Mỗi kiểu lại được kế thừa từ lớp View và tất cả các kiểu đó được gọi là các Widget.

Tất cả mọi widget đều có chung các thuộc tính cơ bản như là cách trình bày vị trí, nền, kích thước, lề,... Tất cả những thuộc tính chung này được thể hiện hết ở trong đối tượng View. Trong Android Platform, các màn hình (screen) luôn được bố trí theo một kiểu cấu trúc phân cấp như hình dưới.

Một màn hình là một tập hợp các Layout và các widget được bố trí có thứ tự. Để thể hiện một màn hình thì trong hàm onCreate của mỗi Activity cần phải được gọi một hàm là setContentView(R.layout.main); hàm này sẽ load giao diện từ file XML lên để phân tích thành mã bytecode



Hình 2.3 : View & ViewGroup

##### 2.3.1. TextView

TextView là một View cho phép hiển thị các dòng chữ (text) trên màn hình, nó có nhiều thuộc tính tùy mục đích sử dụng mà áp dụng, như thiết lập cỡ chữ, font chữ, màu chữ.

Khai báo TextView trong XML:

```
<TextView
    android:text="Nhập số thứ 2 : "
    android:layout_width="100dp"
    android:layout_height="wrap_content"/>
```

Hình 2.3.1.1. Hàm TextView trong xml

Lấy TextView trong Java:

```
TextView textView;
txt01 = findViewById(R.id.txt01);
txt02 = findViewById(R.id.txt02);
```

Hình 2.3.1.2.Hàm TextView trong java

## 2.3.2. Button

Button là một loại View, nó hiển thị nút bấm để chờ người dùng bấm vào. Button kế thừa từ TextView nên các thuộc tính, thiết lập cho TextView là có hiệu quả như đối với Button.

Khai báo Button trong XML:

```
<Button
    android:id="@+id/btn_tinh"
    android:text="Tính"
    android:layout_width="100dp"
    android:layout_height="wrap_content"/>
```

Hình 2.3.2.1. Hàm Button trong xml

Lấy button và bắt sự kiện java

```
btn_tinh = findViewById(R.id.btn_tinh);
//viet code bat su kien
btn_tinh.setOnClickListener((view) -> {
    int s01,s02,s03,ketqua;
    s01 = Integer.parseInt(edt_s01.getText().toString());
    s02 = Integer.parseInt(edt_s02.getText().toString());
    ketqua = s01+s02;
    edt_ketqua.setText(ketqua+"");
});
```

Hình 2.3.2.2. Hàm Button trong java

### 2.3.3. ImageButton

Cũng tương tự như Button, ImageButton chỉ có thêm thuộc tính `android:src = "@drawable/ic_launcher_background"` để thêm hình ảnh vào và không có thẻ text.

Khai báo ImageButton trong XML

```
<ImageButton
    android:id="@+id/imb_click"
    android:src="@drawable/ic_launcher_background"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

Hình 2.3.3.1. Hàm ImageButton trong xml

### 2.3.4. ImageView

Được dùng để hiển thị tài nguyên hình ảnh như các ảnh Bitmap, cá ảnh Drawable. Nó cung cấp các chức năng tùy biến khác nhau như co kéo/cắt ảnh khi hiển thị trên View.

Khai báo trong ImageView trong XML

```
<ImageView
    android:id="@+id/Img_anh"
    android:src="@mipmap/ic_launcher"
    android:layout_width="70dp"
    android:layout_height="100dp"/>
```

Hình 2.3.4.1. Hàm ImageView trong Xml

### 2.3.5. EditText

EditText là loại View hiển thị một hộp (chữ nhật) cho phép người dùng nhập dữ liệu (chữ, số ... có thể khống chế nhập dữ liệu là text, số, phone, ngày tháng ...).

Do EditText mở rộng chức năng từ TextView, nên các tùy chọn thiết lập trình bày ở TextView vẫn đúng cho EditText

```
<EditText
    android:id="@+id/edt_s01"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
```

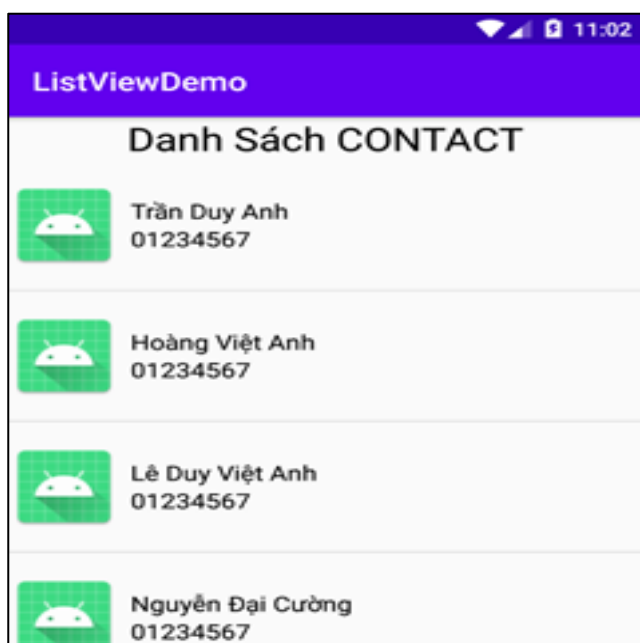
Hình 2.3.5.1. EditText

### 2.3.6. ListView

Được sử dụng để thể hiện một danh sách các thông tin theo từng hàng. Một hàng thông thường được load lên từ một file XML đã được cố định trên đó số lượng thông tin và loại thông tin cần được thể hiện.

Để thể hiện được một list thông tin trên một screen thì cần phải có 3 yếu tố chính:

- Data Source: Data Source có thể là một ArrayList hoặc bất kỳ một cấu trúc dữ liệu kiểu danh sách nào.
- Adapter: Adapter là một class trung gian giúp ánh xạ dữ liệu trong Data Source vào đúng vị trí hiển thị trong ListView. Chẳng hạn, trong Data Source có một trường name và trong ListView cũng có một TextView để thể hiện trường name này. Tuy nhiên ListView sẽ không thể hiển thị dữ liệu trong Data Source lên được nếu như Adapter không gán dữ liệu vào cho đối tượng hiển thị.
- ListView: ListView là đối tượng để hiển thị các thông tin trong Data Source ra một cách trực quan và người dùng có thể thao tác trực tiếp trên đó.



Hình 2.3.6.1: Hình minh họa Listview

Khai báo ListView trong XML:

```
<ListView
    android:id="@+id/lv_contact"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

Hình 2.3.6.2. Listview trong xml



Gán cho ListView một Adapter là nguồn cấp dữ liệu cho nó

```
listView = findViewById(R.id.listView);  
listView.setAdapter(ListViewAdapter);
```

Hình 2.3.6.3. Gán Listview

### 2.3.7. GridView

GridView cũng tương tự như ListView, GridView cũng dựa vào DataSource, và Adapter.

Điểm khác nhau là GridView có thể thiết lập số cột. Dữ liệu luôn đưa vào dưới dạng mảng, list một chiều, nhưng dựa vào số cột ta thiết lập nó tự động ngắt hàng.

Khai báo GridView trong XML.

```
<GridView  
    android:id="@+id/gdvDSTruyen"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:numColumns="3">  
</GridView>
```

Hình 2.3.7.1. Gridview trong xml

Thiết lập android:numColumns="3", tức là GridView sẽ ngắt dòng khi đủ 3 phần tử, nó chỉ khác chỗ này, còn việc đưa dữ liệu lên giống hệt như làm với ListView

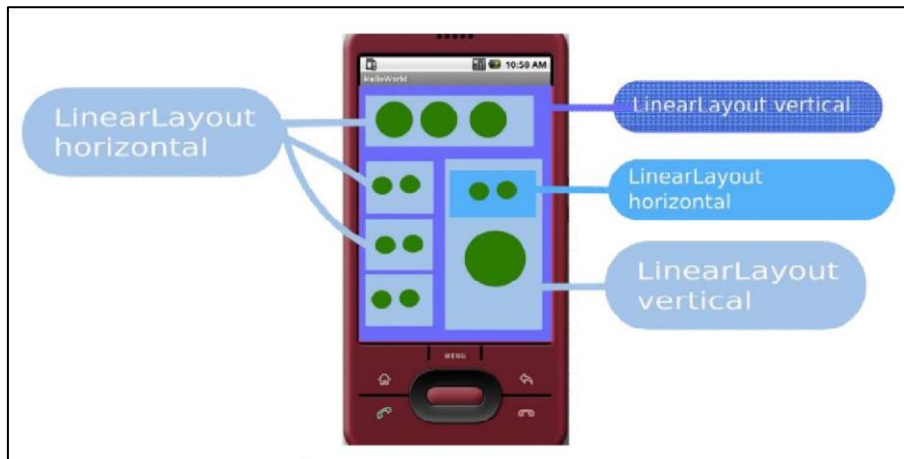
## 2.4. ViewGroup

Một ViewGroup là một đối tượng sử dụng để chứa các đối tượng View và ViewGroup khác để tổ chức và kiểm soát layout của một màn hình. Các đối tượng ViewGroup được sử dụng cho việc tạo ra một hệ thống phân cấp của các đối tượng View do đó có thể tạo các layout phức tạp hơn.

### 2.4.1. LinearLayout

LinearLayout được dùng để bố trí các thành phần giao diện theo chiều ngang hoặc chiều dọc nhưng trên một line duy nhất mà không có xuống dòng.

LinearLayout làm cho các thành phần trong nó không bị phụ thuộc vào kích thước của màn hình. Các thành phần trong LinearLayout được dàn theo những tỉ lệ cân xứng dựa vào các ràng buộc giữa các thành phần.



Hình 2.4.1.1. Linearlayout

## 2.4.2. RelativeLayout

RelativeLayout là một layout mà nó thực hiện các view con nó chứa ở các vị trí trong mối liên hệ của chúng với nhau (như View con này nằm dưới một View con khác View con này nằm bên trái một View con khác), kể cả mối liên hệ của chúng với chính phần tử cha RelativeLayout (như căn thẳng theo cạnh đáy của phần tử cha, nằm giữa phần tử cha, nằm bên trái phần tử cha)



Hình 2.4.2.1. RelativeLayout

## 2.4.3. Framelayout

FrameLayout là loại View cơ sở, nó là loại Layout đơn giản nhất. Mặc dù nó có thể chứa nhiều View con bên trong, nhưng mục đích chính thiết kế ra nó để chứa một View, từ đó nó trở thành cơ sở để tạo ra các View khác phức tạp hơn. Khi thiết kế Layout chứa nhiều View thì không nên sử dụng layout này, vì nó quá đơn giản việc bố cục các View con trong nó rất khó khăn (nó không có các tính năng điều khiển vị trí View con sao cho việc độc lập về màn hình được đảm bảo)

#### 2.4.4. Table layout

Layout này thường được sử dụng khi cần thiết để một table chứa dữ liệu hoặc cần bố trí các widget theo các hàng và cột. Chẳng hạn như, giao diện của một chiếc máy tính đơn giản.

### 2.5. Activity & Intent

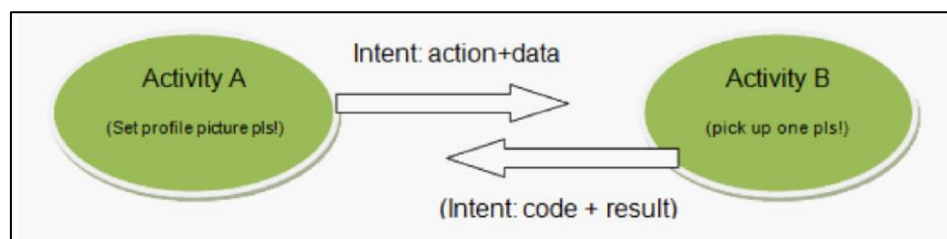
#### 2.5.1. Activity

Activity là một thành phần của ứng dụng Android. Android Activity là nơi để ứng dụng tương tác trực tiếp với người dùng thông qua giao diện. Một ứng dụng có thể sẽ có nhiều màn hình và mỗi màn hình có thể là một Activity (nếu không sử dụng Fragment).

#### 2.5.2. Intent

Là một cấu trúc dữ liệu mô tả cách thức, đối tượng thực hiện của một Activity.

Là cầu nối giữa các Activity : ứng dụng Android thường bao gồm nhiều Activity, mỗi Activity hoạt động độc lập với nhau và thực hiện những công việc khác nhau. Intent chính là người đưa thư, giúp các Activity có thể triệu gọi cũng như truyền các dữ liệu cần thiết tới một Activity khác



Hình 2.5.2.1.Intent

Dữ liệu của Intent

Các thuộc tính của một đối tượng Intent:

Thuộc tính chính	Thuộc tính phụ
<b>action</b> -tên (string) của action mà Intent sẽ yêu cầu thực hiện -có thể là action được Android định nghĩa sẵn (built-in standard action) hoặc do người lập trình tự định nghĩa	<b>category</b> -thông tin về nhóm của action <b>type</b> -định dạng kiểu dữ liệu (chuẩn MIME) -thường được tự động xác định
<b>data</b> -dữ liệu mà Activity được gọi sẽ xử lý -định dạng Uri (thông qua hàm Uri.parse(data))	<b>component</b> -chỉ định cụ thể lớp sẽ thực thi Activity -khi được xác định, các thuộc tính khác trở thành không bắt buộc (optional)
	<b>extras</b> -chứa tất cả các cặp (key,value) do ứng dụng thêm vào để truyền qua Intent (cấu trúc Bundle)
<a href="http://developer.android.com/reference/android/content/Intent.html">http://developer.android.com/reference/android/content/Intent.html</a>	

Hình 2.5.2.2. . Các thuộc tính của một đối tượng Intent

Các Action được định nghĩa sẵn:

Built-in Standard Actions	
ACTION_MAIN ACTION_VIEW ACTION_ATTACH_DATA ACTION_EDIT ACTION_PICK ACTION_CHOOSER ACTION_GET_CONTENT ACTION_DIAL ACTION_CALL ACTION_SEND	ACTION_ANSWER ACTION_INSERT ACTION_DELETE ACTION_RUN ACTION_SYNC ACTION_PICK_ACTIVITY ACTION_SEARCH ACTION_WEB_SEARCH ACTION_FACTORY_TEST ACTION_SENDTO
Built-in Standard Broadcast Actions	
ACTION_TIME_TICK ACTION_TIME_CHANGED ACTION_TIMEZONE_CHANGED ACTION_BOOT_COMPLETED ACTION_PACKAGE_ADDED ACTION_PACKAGE_CHANGED ACTION_PACKAGE_REMOVED	ACTION_PACKAGE_RESTARTED ACTION_PACKAGE_DATA_CLEARED ACTION_UID_REMOVED ACTION_BATTERY_CHANGED ACTION_POWER_CONNECTED ACTION_POWER_DISCONNECTED ACTION_SHUTDOWN

Hình 2.5.2.3. Các Action được định nghĩa sẵn

## CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 3.1. Phát biểu bài toán

Truyện Tranh hiện nay đang là một loại hình giải trí lành mạnh đối với giới trẻ hiện nay , việc đọc truyện đang được phổ biến tại Việt nam và hầu hết chúng được xuất bản dưới dạng sách , cuốn việc này khá là bất tiện trong việc mang theo.

Ứng dụng android xem truyện tranh online sẽ giải quyết vấn đề đang được lưu tâm đến của giới trẻ , việc đọc truyện tranh qua điện thoại sẽ giúp chúng ta tiết kiệm chi phí và thuận tiện hơn

Ứng dụng android xem truyện tranh online giúp người dùng dễ dàng tìm kiếm và đọc truyện mọi lúc mọi nơi . Người đọc có thể phản hồi sự yêu thích quyển truyện đó thông qua thống kê like từ đó giúp nâng cao cải thiện ứng dụng

## 3.2. Phân tích thiết kế

### 3.2.1. Phân tích dữ liệu

Xuất phát từ những yêu cầu của bài toán từ đó xây dựng dựa trên những thông tin cơ bản sau: Id, Tên truyện, Số chương, Ảnh truyện, Link truyện và Đếm like. Đó đó, ta sẽ xây dựng một bảng thiết kế các dữ liệu đã nêu trên

### 3.2.2. Bảng thiết kế dữ liệu

Đây là các dữ liệu truyện mà ta phải đưa lên host bao gồm: Tên truyện, Số chương, Ảnh truyện, Link truyện và Đếm like

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Mô tả	Ghi chú
1	id	int	11	Định danh	Khóa chính
2	Tentruyen	Varchar	255	Tên truyện	
3	Sochuong	Int	11	Số chương	
4	Anhtruyen	Varchar	255	Ảnh truyện	
5	linktruyen	Varchar	255	Link truyện	
6	Demlike	Int	11	Đếm like	

Hình 3.2.2.1. Bảng thiết kế logic

### 3.2.3. Bảng cơ sở dữ liệu vật lý

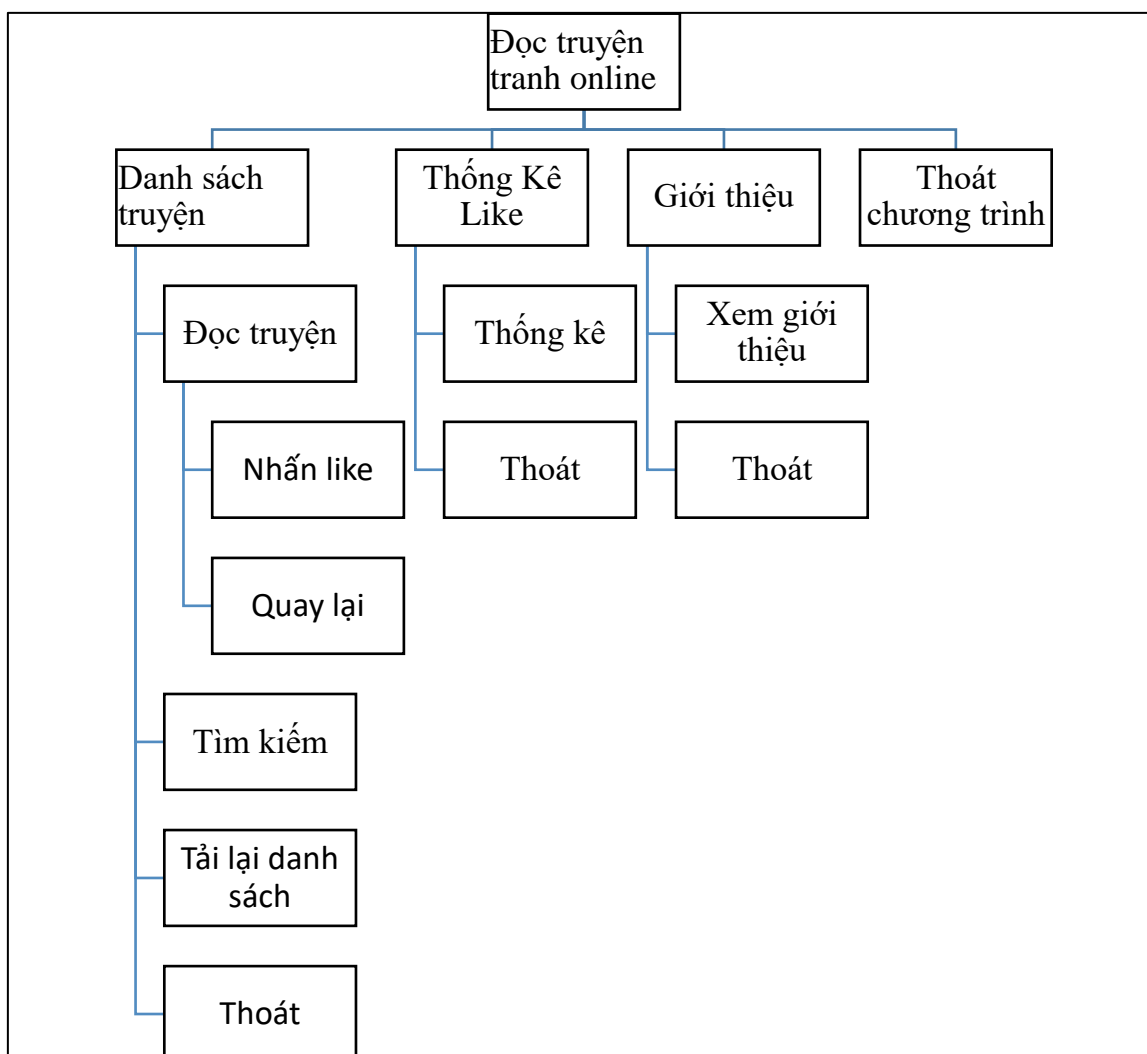
Dữ liệu vật lý hoàn chỉnh sau khi ta tạo xong ở trên hệ quản trị cơ sở dữ liệu

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/>	2 tentruyen	varchar(255)	utf8_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	3 sochuong	int(11)			No	None			Change  Drop  More
<input type="checkbox"/>	4 anhtruyen	varchar(255)	utf8_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	5 linktruyen	varchar(255)	utf8_unicode_ci		No	None			Change  Drop  More
<input type="checkbox"/>	6 demlike	int(11)			No	None			Change  Drop  More

Hình 3.2.3.1. Bảng dữ liệu vật lý

### 3.3. Phân tích thiết kế chức năng

#### 3.3.1. Sơ đồ phân rã chức năng



Hình 3.3.1.1. Sơ đồ phân rã chức năng

#### 3.3.2. Chức năng Danh sách truyện

Chức năng này sẽ hiển thị tất cả danh sách thông tin của truyện khi chạy ứng dụng mỗi truyện sẽ được liệt kê trong một danh sách bao gồm: Tên truyện, Số chương, Ảnh của truyện đó

- Khi người dùng muốn đọc truyện thì ấn vào quyền truyện muốn đọc trong danh sách truyện

Khai báo trong .xml :

```

<WebView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/webview">
</WebView>

```

Hình 3.3.2.1. Đọc truyện trong xml

Khai báo trong .java :

```

private void doctruyen(){
    // doc truyện theo link lay duoc
    WebView webview = findViewById(R.id.webview);
    webview.loadUrl(linktruyen);
}

```

Hình 3.3.2.2.Hàm đọc truyện trong .java

- Để người dùng có thể dễ dàng tìm ra những quyển truyện có nội dung đúng với từ khóa

Khai báo trong XML :

```

<EditText
    android:id="@+id/edtTimKiem"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="80dp"
    android:layout_marginTop="5dp"
    android:layout_marginRight="80dp"
    android:layout_marginBottom="5dp"
    android:background="@drawable/bg_edt"
    android:gravity="center"
    android:hint="Nhập tên truyện cần tìm"
    android:textColor="#fff"
    android:textColorHint="#B3ADAD"
    android:textSize="20sp" />

```

Hình 3.3.2.3. Thiết kế nút tìm kiếm trong xml

Khai báo trong .java

```

edtTimKiem.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

    }

    @Override
    public void afterTextChanged(Editable editable) {
        String s = edtTimKiem.getText().toString();
        if(s.length() > 0) {
            adapter.sortTruyen(s);
        } else {
            layalltruyentranh();
        }
    }
});

```

Hình 3.3.2.4. Hàm tìm kiếm trong java

- Nếu người dùng không muốn dùng ứng dụng thì nhấp vào nút thoát
- Khai báo trong xml

```

<ImageButton
    android:id="@+id/imb_thoat"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentBottom="true"
    android:layout_marginLeft="17dp"
    android:layout_marginTop="16dp"
    android:layout_marginBottom="12dp"
    android:background="@null"
    android:src="@drawable/ic_thoat" />

```

Hình 3.3.2.5. Thiết kế nút quay lại trong xml

Khai báo trong java



```

imb_thoat.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        finish();
    }
});

```

Hình 3.3.2.6. Hàm quay lại trong java

- Khi ứng dụng đang chạy mà bị chậm hoặc kết nối internet bị ngắt thì người dùng có thể nhấn nút tải lại trang

Khai báo trong .xml

```

<ImageButton
    android:id="@+id/imb_tailai"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginRight="15dp"
    android:layout_marginBottom="12dp"
    android:background="@null"
    android:src="@drawable/ic_tailai" />

```

Hình 3.3.2.7. Thiết kế nút tải lại trang trong xml

Khai báo trong .java :

```

imb_tailai.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        init();
    }
});

```

Hình 3.3.2.8. Hàm tải lại trang trong java

- Ở màn hình chi tiết nếu người dùng muốn thích truyện thì ấn vào nút like bên phải

Khai báo trong xml

```

<ImageButton
    android:id="@+id/imb_like"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:scaleType="centerInside"
    android:src="@drawable/ic_love16"/>

```

Hình 3.3.2.9. Thiết kế nút đếm like trong xml

Khai báo trong java

```

imb_like.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        demlike = demlike+1;
        txt_demlike.setText(demlike+"");
    }
});

```

Hình 3.3.2.10. Hàm đếm like trong java

### 3.3.3. Chức năng Thống kê like

Chức năng này có tác dụng để thống kê những quyền truyện có lượt like nhiều nhất

Khai báo trong xml

```

<Button
    android:id="@+id/btn_thongkelike"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Thống Kê Like"
    android:textAllCaps="false" />

```

Hình 3.3.3.1. Thiết kế nút thống kê like trong xml

Khai báo trong jav

```

btn_thongkelike.setOnClickListener((view) -> {
    Intent intent_tklike = new Intent(getApplicationContext(), ThongKeLike.class);
    intent_tklike.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(intent_tklike);
});

```

Hình 3.3.3.2. Hàm thông kê like trong java

### 3.3.4. Chức năng giới thiệu

Chức năng này có tác dụng hiển thị nội dung của ứng dụng cho người dùng nhìn thấy

- Để có thể hiển thị nội dung ta phải dùng TextView để làm

Khai báo trong xml

```

<TextView
    android:text="Ứng dụng đọc truyện online cho điện thoại phiên bản 1.0"
    android:textColor="#000"
    android:textSize="18sp"
    android:gravity="center_vertical"
    android:layout_width="match_parent"
    android:layout_margin="5dp"
    android:layout_height="0dp"
    android:layout_weight="9"/>

```

Hình 3.3.4.1 : Thiết kế nút giới thiệu trong xml

- Ở màn hình giới thiệu nếu người dùng muốn quay lại màn hình chính thì nhấn nút ‘Đóng’

Khai báo trong xml

```

<Button
    android:id="@+id/btn_dong"
    android:text="Đóng"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

```

Hình 3.3.4.2.Thiết kế nút đóng trong xml

Khai báo trong java

```
btn_dong.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        finish();  
    }  
});
```

Hình 3.3.4.3. Hàm đóng trong java

### 3.3.5. Chức năng Thoát ứng dụng

Chức năng này có tác dụng nếu người dùng không muốn dùng ứng dụng thì nhấp vào nút thoát

Khai báo trong xml

```
<Button  
    android:id="@+id/btn_thoat"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Thoát"  
    android:textAllCaps="false" />
```

Hình 3.3.5.1 : Thiết kế nút thoát màn hình chính trong xml

Khai báo trong java

```
btn_thoat.setOnClickListener((view) -> {  
    AlertDialog.Builder builder = new AlertDialog.Builder( context: MainActivity.this, android.R.style.Theme_DeviceDefault_Dialog);  
    builder.setTitle("Bạn có chắc muốn thoát khỏi app");  
    builder.setMessage("Hãy lựa chọn bên dưới để xác nhận");  
    builder.setIcon(android.R.drawable.ic_dialog_alert);  
    builder.setPositiveButton( text: "Có", (dialogInterface, i) -> {  
        onBackPressed();  
    });  
    builder.setNegativeButton( text: "Không", new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialogInterface, int i) {  
        }  
    });  
    builder.show();  
});
```

Hình 3.3.5.2.Hàm thoát màn hình chính trong java

## CHƯƠNG 4 : CHƯƠNG TRÌNH ỨNG DỤNG

### 4.1. Kết quả đạt được

Tìm hiểu lịch sử và kiến trúc của Hệ điều hành Android.

Tìm hiểu được các kỹ thuật để xây dựng một ứng dụng hoàn chỉnh trên thiết bị thật

Xây dựng thành công ứng dụng đọc truyện tranh online

Bước đầu xây dựng thành công ứng dụng đọc truyện tranh online với các chức năng cơ bản.

### 4.2. Một số giao diện chính

#### 4.2.1. Giao diện Chính

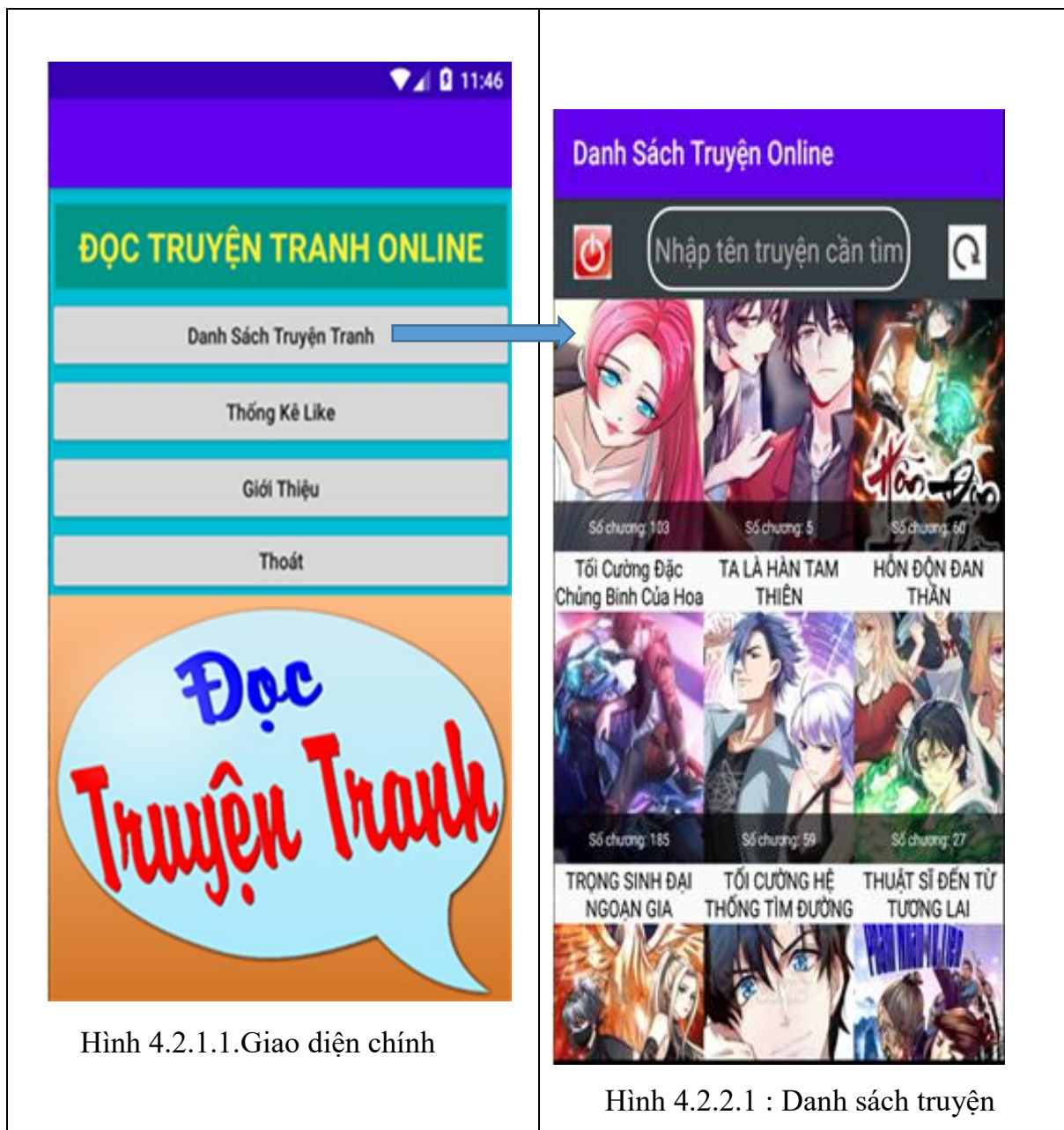
Sau khi khởi động ứng dụng ta sẽ mở một màn hình như hình dưới đây gồm các chức năng sau :



Hình 4.2.1.1. Giao diện chính

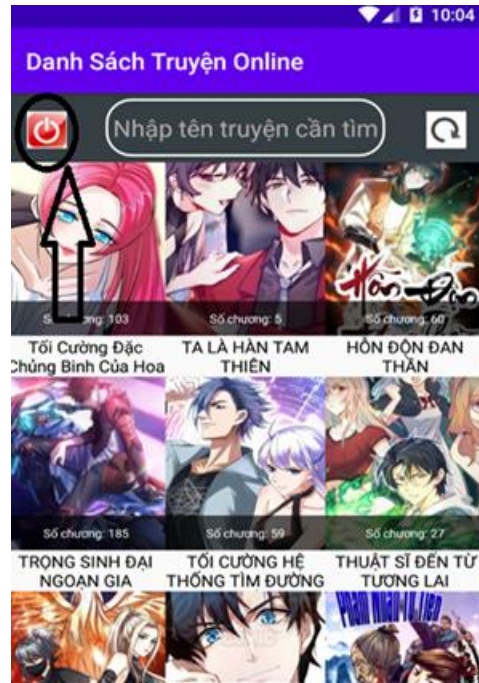
#### 4.2.2. Giao diện Danh sách truyện

Ở màn hình chính của ứng dụng , người dùng nhấn vào nút Danh Sách Truyện thì một màn hình mới mở ra



Chức năng Danh sách truyện có tác dụng hiển thị tất cả danh sách thông tin của truyện khi chạy ứng dụng

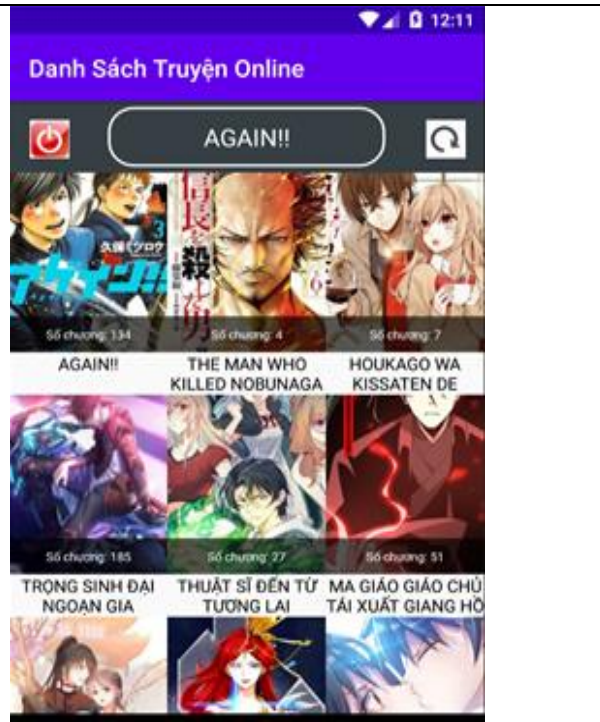
Ở màn hình danh sách truyện nếu người dùng muốn quay lại màn hình chính



Hình 4.2.2.2: Thoát về màn hình chính

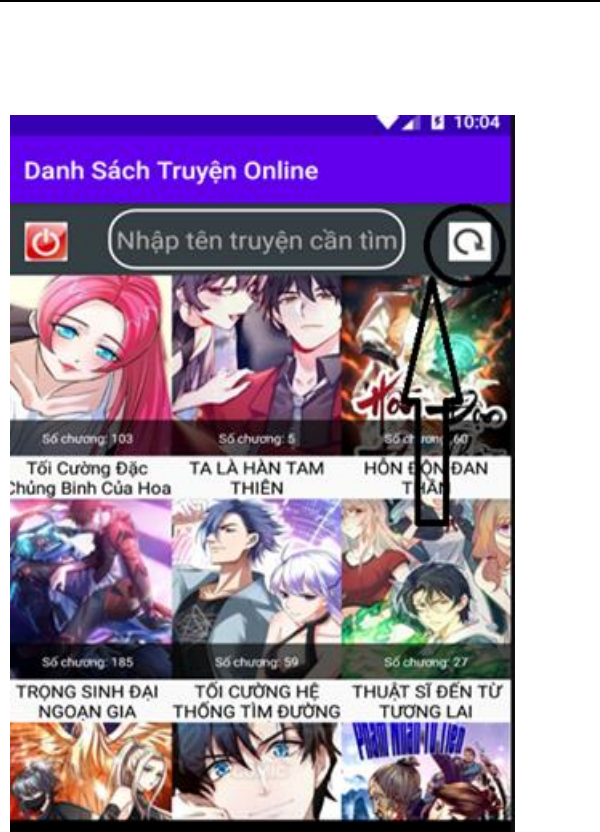
Màn hình tìm kiếm : Hỗ trợ người dùng có thể dễ dàng tìm kiếm các bộ truyện tranh cần tìm , khi nhập từ khóa hệ thống sẽ lọc ra các bộ truyện theo yêu cầu của người sử dụng





Hình 4.2.2.3: Tìm kiếm truyện cần tìm

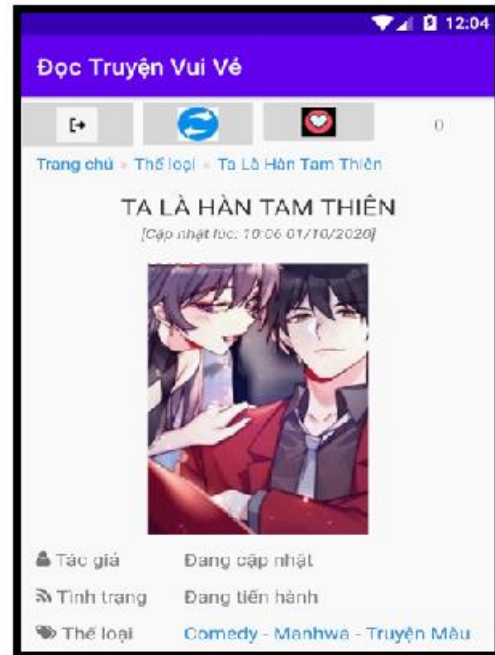
Nếu danh sách truyện hiển thị ra màn hình chính bị mất kết nối thì người dùng cần nhấp vào nút tải lại bên phải



Hình 4.2.2.4: Tải lại trang

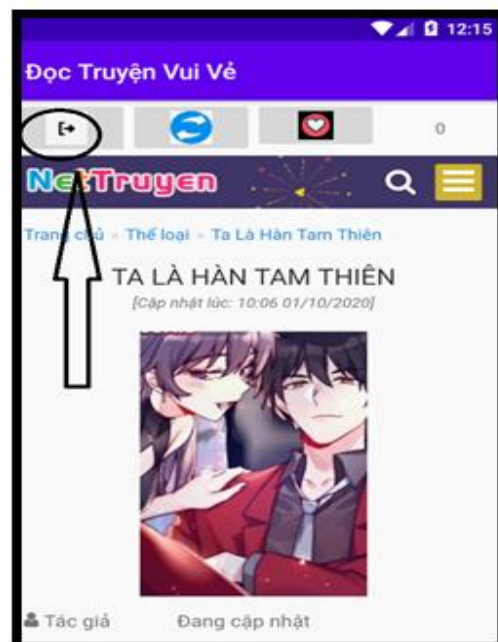


Màn hình chi tiết : hiển thị chi tiết thông tin và các chương của bộ truyện



Hình 4.2.2.5: Giao diện màn hình chi tiết

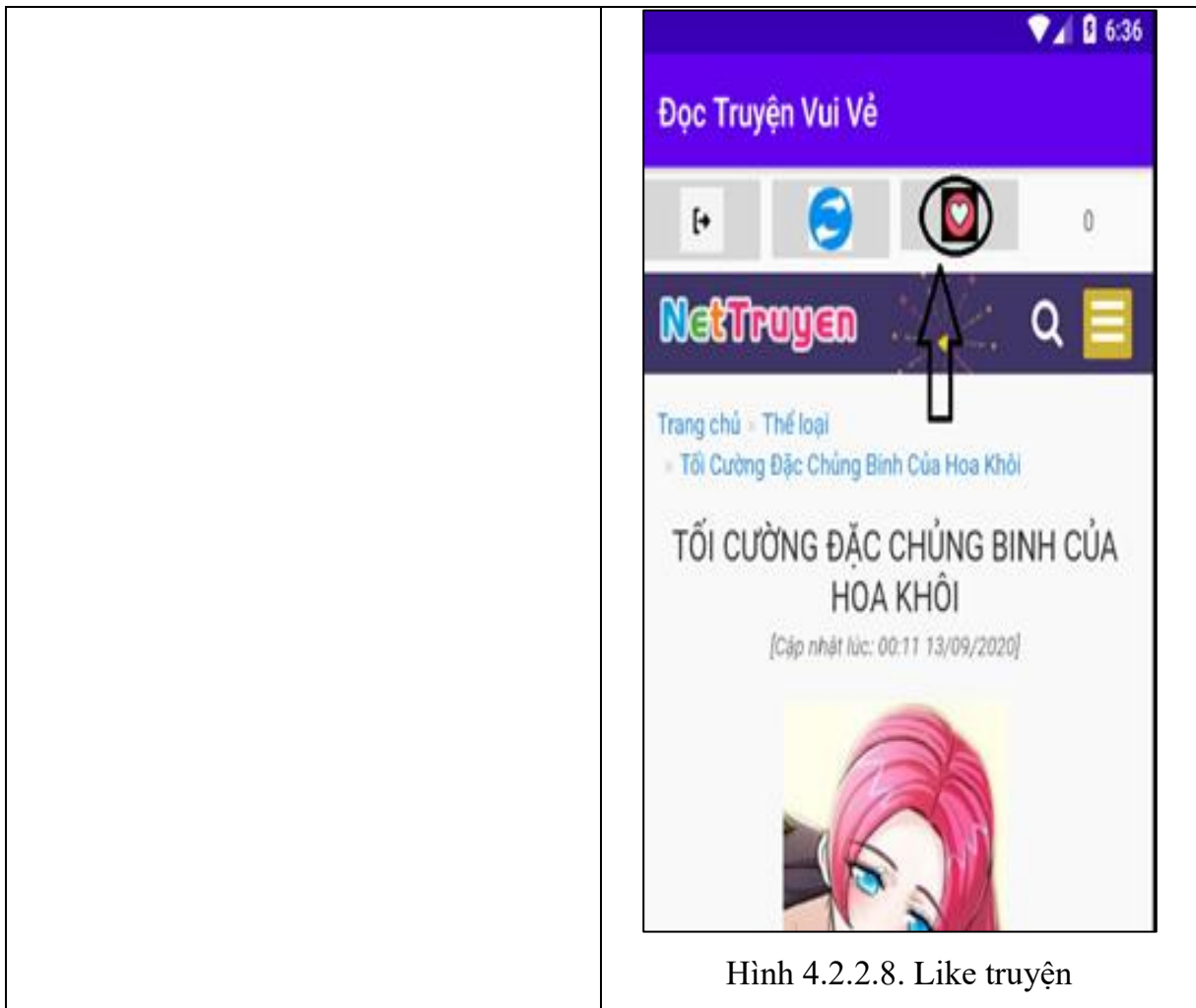
Khi người dùng đang ở màn hình chi tiết truyện mà nếu người dùng muốn quay trở lại màn hình danh sách truyện thì nhấp vào nút quay lại bên trái



Hình 4.2.2.6 : Quay lại danh sách

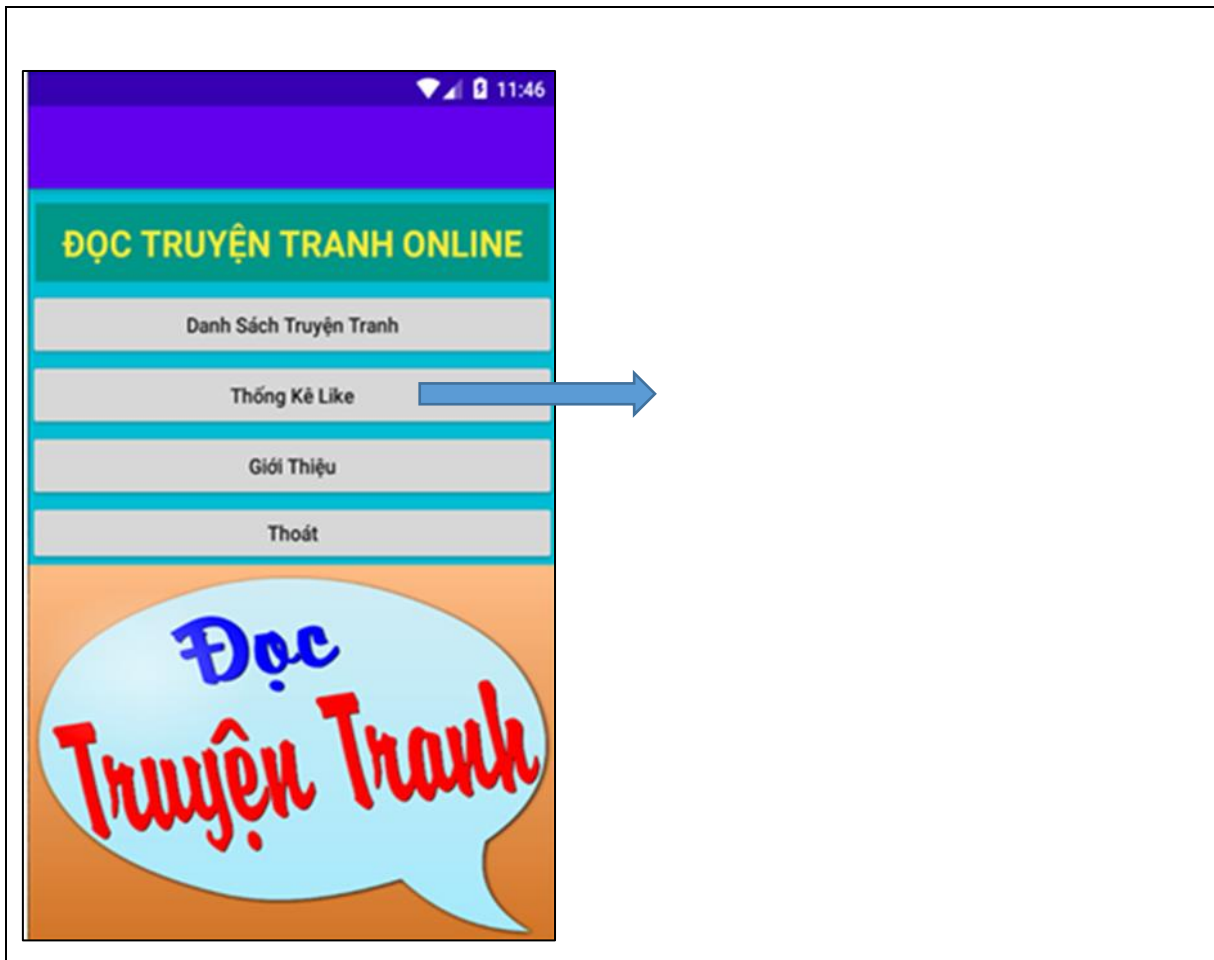
	truyện
<p>Nếu màn hình chi tiết bị mất kết nối thì người dùng cần nhấp vào nút tải lại ở giữa</p>	 <p>Hình 4.2.2.7. Tải lại màn hình chi tiết</p>

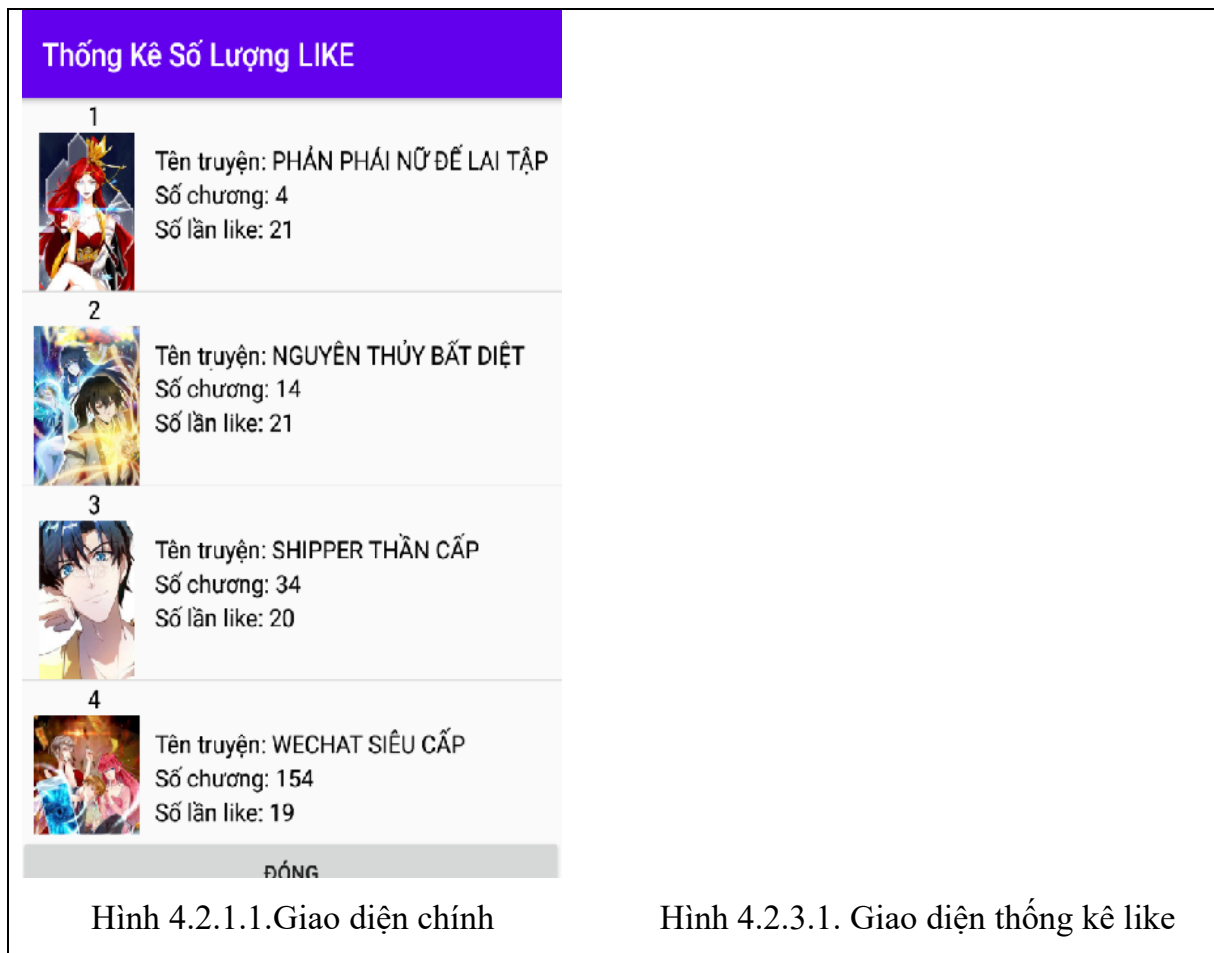
<p>Khi người dùng đang ở màn hình đọc truyện mà nếu người dùng muốn like quyển truyện đang đọc hoặc vừa đọc thì nhấp vào nút like bên phải</p>	
--	--



### 4.2.3. Giao diện Thống kê

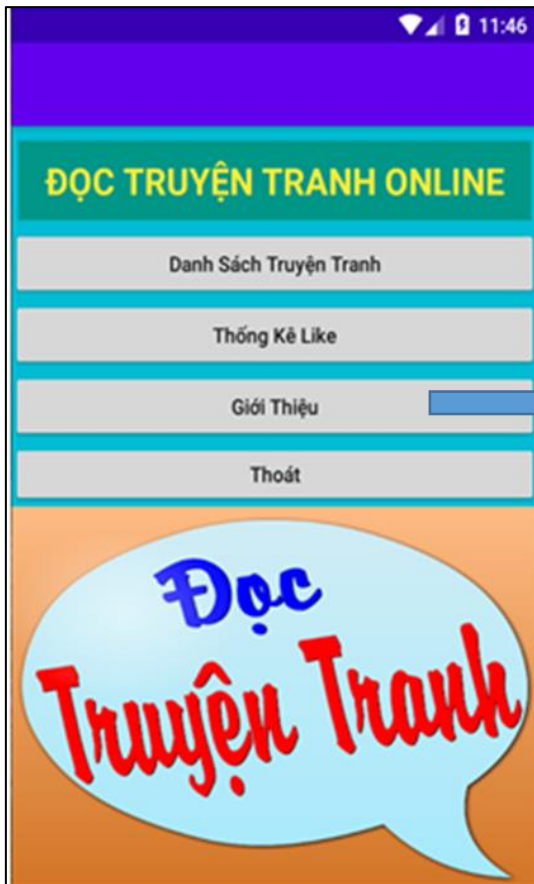
Ở màn hình giao diện chính khi người dùng ấn vào nút Thống kê like thì một màn hình mới hiển thị những quyển truyện có lượt like nhiều nhất và những quyển truyện có lượt like ít nhất như hình 4.2.3.1 và khi muốn quay lại thì nhấn nút “Đóng” bên dưới





#### 4.2.4. Giao diện Giới thiệu

Ở màn hình giao diện chính khi người dùng ấn vào nút Giới thiệu thì một màn hình mới hiển thị thông tin giới thiệu của ứng dụng như hình 4.2.4.1 và khi muốn quay lại màn hình chính thì nhấn nút “Đóng” bên dưới để quay lại.



Hình 4.2.1.1. Giao diện chính

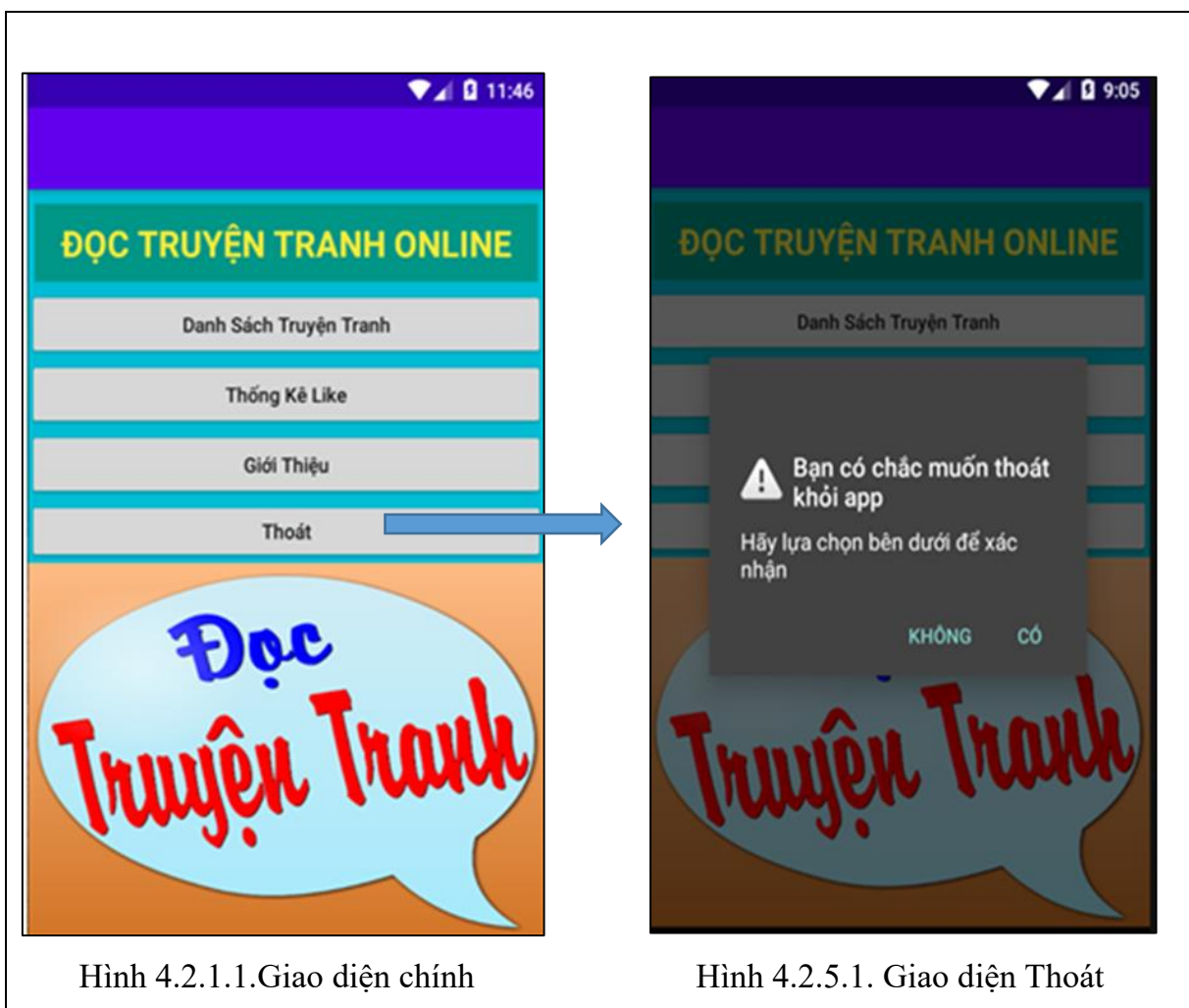
Hình 4.2.4.1. Giao diện giới thiệu

#### 4.2.5. Giao diện Thoát

Ở màn hình khi người dùng ấn vào nút “Thoát” thì sẽ hiện ra bảng thông báo muốn chúng ta xác nhận có muốn thoát chương trình hay không như hình 4.2.5.1

Nếu người dùng ấn Có thì thoát ứng dụng

Còn nếu ấn Không thì sẽ ở lại màn hình chính



### **4.3. Những hạn chế và hướng phát triển trong tương lai**

#### **4.3.1. Hạn chế**

- Chưa hỗ trợ nền tảng IOS
- Còn nhiều chức năng bị thiếu sót
- Chưa mang lại tính giải trí cao
- Chưa mang lại thu nhập cho người quản lý

#### **4.3.2. Hướng phát triển**

- Tạo thêm nhiều chức năng mới
- Có thể import nhiều bộ truyện tranh
- Xây dựng được ứng dụng độc đáo
- Chạy quảng cáo cho google mở các mini game
- Tích hợp các câu hỏi đố vui
- Tạo ra một môi trường giải trí tốt nhất
- Mở rộng phát triển ứng dụng trên nền tảng IOS
-



## KẾT LUẬN

Sau một thời gian tìm hiểu đề tài “Xây dựng ứng dụng android xem truyện tranh online” em đã thực hiện được nội dung đề tài của em theo yêu cầu đặt ra .

Đề tài “Xây dựng ứng dụng xem truyện tranh online ” là do sở thích đọc truyện cũng như mong muốn tạo ra một môi trường giải trí lành mạnh dành cho giới trẻ hiện nay

Mặc dù em đã có nhiều cố gắng , tìm hiểu các kiến thức đã học , kết hợp tra cứu các tài liệu trên mạng nhưng do hạn chế về thời gian , khả năng và kinh nghiệm nên không tránh khỏi những thiếu sót nhất định nên đồ án tốt nghiệp của em đã đạt được một số kết quả sau:.

- Tạo được một Host free
- Tạo được cơ sở dữ liệu web trên Hosting
- Xây dựng được Web-service để đẩy dữ liệu từ ứng dụng lên cơ sở dữ liệu web.
- Tìm hiểu lịch sử và kiến trúc của Hệ điều hành Android.
- Tìm hiểu được các kỹ thuật để xây dựng một ứng dụng cho trên thiết bị di động hoàn chỉnh trên thiết bị thật
- Bước đầu xây dựng thành công ứng dụng đọc truyện tranh online với các chức năng cơ bản.

## TÀI LIỆU THAM KHẢO

- [1].[https://vi.wikipedia.org/wiki/Android\\_\(h%E1%BB%87\\_%C4%91i%E1%BB%81u\\_h%C3%A0nh\)](https://vi.wikipedia.org/wiki/Android_(h%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh))
- [2].<http://3i.com.vn/index.php/cong-ngh/c%C3%B4ng-ngh%E1%BB%87-%C4%91i%E1%BB%87n-tho%E1%BA%A1i.html>
- [3].<https://quantrimang.com/android-studio-la-gi-149713>