**BỘ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT NAM**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**BÁO CÁO THỰC TẬP TỐT NGHIỆP ĐẠI HỌC**

**Đề tài: “”**

**Người hướng dẫn : ThS. LƯU NGUYỄN KỲ THƯ**

**Sinh viên thực hiện: Lê Thạnh**

**Mã số sinh viên : N14DCCN309**

**Lớp : D14CQCP01-N**

**Khóa : 2014**

**Hệ : ĐẠI HỌC CHINH QUY**

**TP.HCM, tháng 8 năm 2018**

**LỜI CẢM ƠN**

Lời đầu tiên em xin được gửi lời cảm ơn chân thành đến quý thầy cô giáo trong Học Viện Công Nghệ Bưu Chính Viễn Thông cơ sở Thành Phố Hồ Chí Minh – những người đã trực tiếp giảng dạy, truyền đạt những kiến thức bổ ích cho em trong suốt những năm học vừa qua. Đó chính là những nền tảng cơ bản, những hành trang vô cùng quý giá giúp em bước vào sự nghiệp sau này trong tương lai.

Bên cạnh đó, em cảm ơn Giám đốc Toàn, Giám đốc của công ty Horical. Qua hơn 6 tuần thực tập tại công ty Horical, em đã đúc kết được rất nhiều kinh nghiệm thực tế mà khi ngồi trên ghế nhà trường em chưa được biết. Hiểu được quy trình làm ra một ứng dụng di động sẽ có các bước nào, thực hiện ra làm sao. Trau dồi kĩ năng để hoàn thiện bạn thân hơn. Chúc Horical thành công hơn nữa.

Tiếp đến, em cũng gửi lời cám ơn chân thành đến thầy ThS.Lưu Nguyễn Kì Thứ và các anh chị tham gia training đã tận tình giúp đỡ và tạo điều kiện giúp em hoàn thành tốt quá trình thực tập.

Trong quá trình thực tập và làm báo cáo, do còn thiếu kinh nghiệm nên không thể tránh khỏi những sai sót. Em mong các thầy cô chỉ bảo thêm giúp em có thể hoàn thiện và đạt kết quả tốt hơn.

Kính chúc thầy cô luôn vui vẻ, dồi dào sức khỏe và thành công trên con đường dạy học của mình

Em xin chân thành cám ơn!

TP.HCM, ngày tháng năm 2018

Sinh viên thực hiện

**Lê Thạnh**

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

**Độc lập – Tự do – Hạnh Phúc**

TP. Hồ Chí Minh, ngày tháng năm 2017

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN THỰC TẬP TỐT NGHIỆP ĐAI HỌC**

**Tên đề tài:**

**Sinh viên: Lê Thạnh Lớp: D13CQCP01-N**

**Giáo viên hướng dẫn: ThS. Lưu Nguyễn Kỳ Thư**

**Nơi công tác:**

**NỘI DUNG NHẬN XÉT**

Đánh giá chung:

……………………………………………………………………………………………………………………………………………………………………

Đánh giá chi tiết:

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Nhận xét về tinh thần, thái độ làm việc:

……………………………………………………………………………………………………………………………………………………………………

Kết luận:

……………………………………………………………………………………………………………………………………………………………………

Điểm hướng dẫn ():

**GIẢNG VIÊN HƯỚNG DẪN**

(Ký, ghi rõ họ tên)

Ths. Lưu Nguyễn Kì Thư

**MỤC LỤC**

[LỜI MỞ ĐẦU 2](#_Toc499485411)

[CHƯƠNG 1: TỔNG QUAN HỆ ĐIỀU HÀNH ANDROID VÀ KIẾN TRÚC CỐT LÕI 2](#_Toc499485412)

[1.1 Giới thiệu 2](#_Toc499485413)

[1.2 Kiến trúc 2](#_Toc499485414)

[1.2.1 Linux Kernel 2](#_Toc499485415)

[1.2.2 Hardware Abstraction Layer (HAL) 2](#_Toc499485416)

[1.2.3 Native C/C++ Libraries 2](#_Toc499485417)

[1.2.4 Android Runtime 2](#_Toc499485418)

[1.2.5 Java API Framework 2](#_Toc499485419)

[1.2.6 System Apps 2](#_Toc499485420)

[CHƯƠNG 4: NGÔN NGỮ LẬP TRÌNH 2](#_Toc499485438)

[4.1 Ngôn ngữ lập trình Java 2](#_Toc499485439)

[4.1.1 Java là gì? 2](#_Toc499485440)

[4.1.2 Tại sao lại chọn ngôn ngữ lập trình Java 2](#_Toc499485441)

[4.2 Spring Framework 2](#_Toc499485442)

[4.2.1 Giới thiệu về Spring 2](#_Toc499485443)

[4.2.2 Kiến trúc Spring 2](#_Toc499485444)

[4.2.3 Lợi ích của việc sử dụng Spring Framework 2](#_Toc499485445)

[CHƯƠNG 5: GIỚI THIỆU VÀ THIẾT KẾ ỨNG DỤNG 2](#_Toc499485446)

[5.1 Giới thiệu 2](#_Toc499485447)

[5.2 Các chức năng chính 2](#_Toc499485448)

[5.3 Thiết kế ứng dụng 2](#_Toc499485449)

[5.3.1 Mục tiêu 2](#_Toc499485450)

[5.3.2 Application Achitechture 2](#_Toc499485451)

[5.3.3 Thiết kế cơ sở dữ liệu 2](#_Toc499485452)

[5.4 Chi tiết kỹ thuật 2](#_Toc499485453)

[5.4.1 Cài đặt môi trường phát triển 2](#_Toc499485454)

[5.4.2 Tiến trình xử lý 2](#_Toc499485455)

[- Đăng nhập, đăng ký tài khoản 2](#_Toc499485456)

[- Hồ sơ 2](#_Toc499485457)

[- Sổ y bạ (lịch sử bệnh của em bé) 2](#_Toc499485458)

[- Chẩn đoán bệnh và chấn thương ở trẻ em 2](#_Toc499485459)

[- Servie xử lý giọng nói của người dùng 2](#_Toc499485460)

[- Thuật toán tính tỉ lệ (%) giống nhau của hai chuỗi. 2](#_Toc499485461)

[- Hiển thị bệnh viện gần nhất 2](#_Toc499485462)

[KẾT LUẬN 2](#_Toc499485463)

[1 KẾT QUẢ ĐẠT ĐƯỢC 2](#_Toc499485439)

[1.1 Chương trình đã đạt được 2](#_Toc499485440)

[1.2 Khả năng ứng dụng 2](#_Toc499485441)

[1.3 Kiến thức đạt được 2](#_Toc499485442)

[2 HẠN CHẾ, KHÓ KHĂN 2](#_Toc499485443)

[TÀI LIỆU THAM KHẢO 2](#_Toc499485444)

# DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

# DANH MỤC CÁC BẢNG VẼ

# DANH MỤC CÁC HÌNH VẼ

# LỜI MỞ ĐẦU

# CHƯƠNG 1: TỔNG QUAN HỆ ĐIỀU HÀNH ANDROID VÀ KIẾN TRÚC CỐT LÕI

## 1.1 Giới thiệu

Android được phát triển bởi tập đoàn Google, phiên bản đầu tiên ra đời năm 2008.

Được xây dựng trên một nền tảng mở, và một bộ thư viện đa năng, mạnh mẽ với nguyên lý mở, Android đã nhanh chóng được cộng đồng lập trình viên di động hưởng ứng mạnh mẽ.

Nền tảng Android tích hợp nhiều tính năng nổi bật:

- Android là một hệ điều hành nhân Linux, đảm bảo sự tương tác với các phần cứng, quản lý bộ nhớ, điều khiển các tiến trình tối ưu cho các thiết bị di động.

- Bộ ứng dụng khung cho phép sử dụng lại và thay thế các thành phần riêng lẻ.

- Máy ảo Dalvik được tối ưu cho các thiết bị di động, chạy các ứng dụng lập trình trên ngôn ngữ Java.

- Các thư viện cho phát triển ứng dụng mã nguồn mở bao gồm SQLite, WebKit,  
OpenGL và trình quản lý đa phương tiện.

- Hỗ trợ các chuẩn đa phương tiện phổ biến, thoại trên nền GSM, Bluetooth  
EDGE, 3G và Wifi

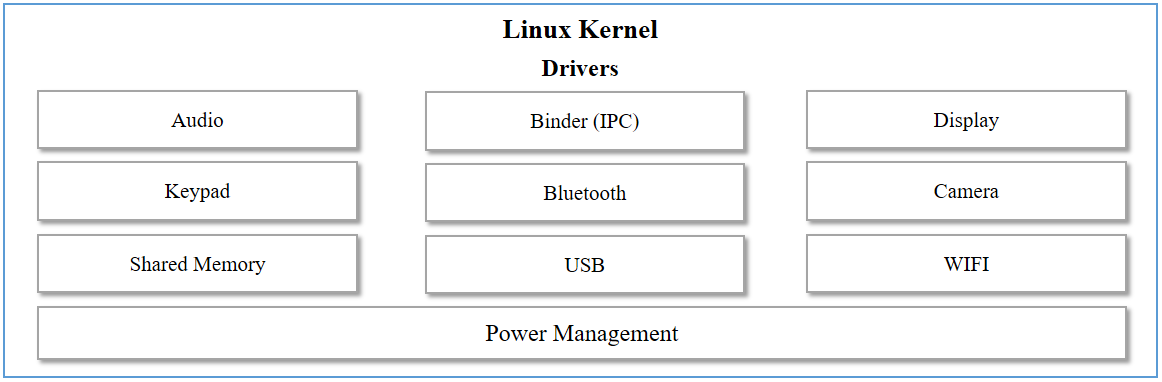
- Hỗ trợ Camera, GPS, la bàn, máy đo gia tốc…

Android cung cấp một tập hợp đầy đủ các phần mềm cho thiết bị di động bao gồm: hệ điều hành, các khung ứng dụng và các ứng dụng cơ bản.

## 1.2 Kiến trúc

Hệ điều hành Android là một ngăn xếp các thành phần phần mềm, được chia thành năm lớp và bao gồm sáu thành phần: Linux Kernel, Hardware Abstraction Layer (HAL), Native C/C++ Libraries, Android Runtime, Java API Framework, System applications.

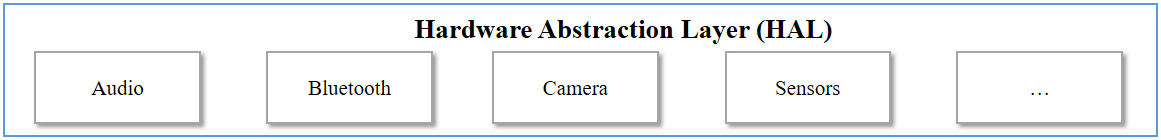
## 1.2.1 Linux Kernel



Hình 1.1 Kiến trúc nhân Linux trong Android

Từ phiên bản Android 4.0 trở về sau, Android sử dụng nhân Linux 3.x làm nhân cho các dịch vụ hệ thống như bảo mật, quản lý bộ nhớ, quản lý tiến trình (xử lý tiến trình, đa luồng), ngăn xếp mạng và trình điều khiển thiết bị (giao tiếp USB, giao tiếp hồng ngoại, không đây, v.v…). Nhân Linux này cũng có vai trò như một lớp trừu tượng giữa phần cứng và phần mềm.

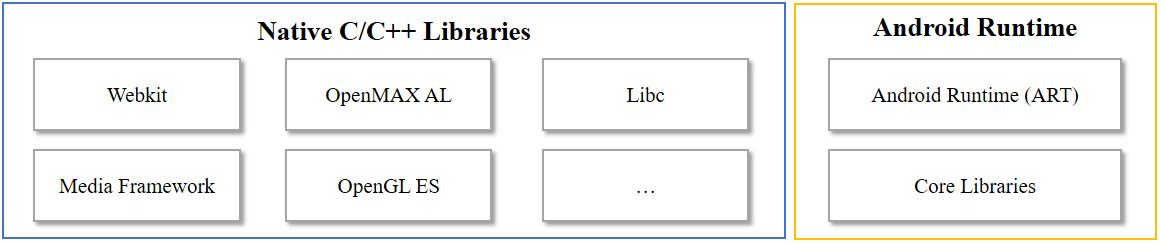
### 1.2.2 Hardware Abstraction Layer (HAL)



Hình 1.2 Tầng phần cứng trừu tượng

HAL cung cấp các interfaces chuẩn cho phép thiết bị phần cứng có khả năng truy cập đế các Java API framework. Bao gồm nhiều modules, mỗi modules implement một interfaces cho một loại phần cứng cụ thể, chẳng hạn như camera, bluetooth,..

### 1.2.3 Native C/C++ Libraries

****

Hình 1.3 Tầng Libraries và Android Runtime

Nhiều thành phần và dịch vụ cốt lõi của hệ thống Android, chẳng hạn như ART và HAL, được xây dựng từ mã nguồn yêu cầu các thư viện gốc được viết bằng C và C++. Android cung cấp các API Java Framework để sử dụng một số thư viện gốc này. Ví dụ, truy cập OpenGL ES thông qua Java OpenGL API của Android để hỗ trợ vẽ và thao tác đồ họa 2D và 3D.

Nếu bạn đang phát triển ứng dụng yêu cầu code C hoặc C++, bạn có thể sử dụng Android NDK (Android Native Development Kit) để truy cập vào một số thư viện được cũng cấp bởi Android.

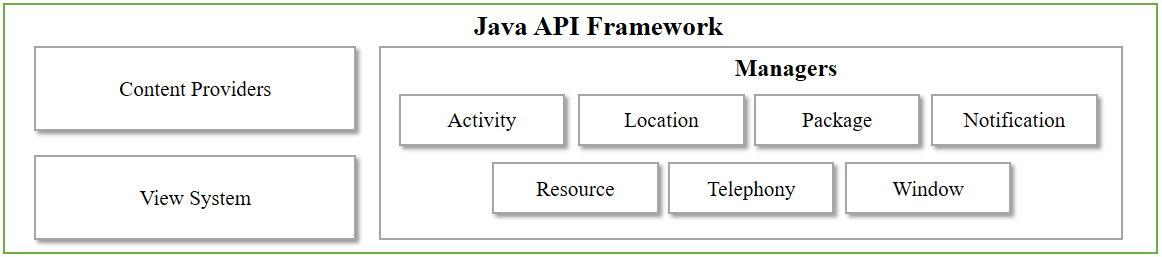
## 1.2.4 Android Runtime

Đối với các thiết bị chạy Android phiên bản 5.0 (API 21) trở về sau, mỗi ứng dụng chạy với một tiến trình riêng. ART được viết để chạy nhiều máy ảo trên các thiết bị bộ nhớ thấp bằng cách thực hiện các tệp DEX, một định dạng bytecode được thiết kế đặc biệt cho Android.

Một số tính năng chính của ART:

* Ahead-of-time (AOT) và just-in-time (JIT) compilation.
* Tối ưu hóa việc thu gom rác ( Garbage Collection GC).
* Hỗ trợ sửa lỗi (Debugging), Xử lý ngoại lệ (exceptions ),…

### 1.2.5 Java API Framework

****

Hình 1.4 Kiến trúc tầng Java API

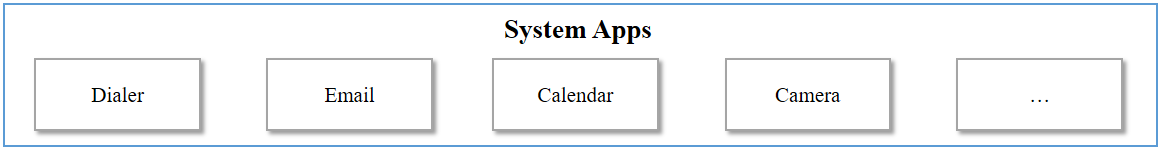
Kiến trúc của Android khuyến khích khái niệm Thành phần sử dụng lại, cho phép công bố và chia sẻ các Activity, Service, dữ liệu, với các ứng dụng khác với quyền truy cập được quản lý bởi khai báo.

Cơ chế đó cho phép người lập trình tạo ra một trình quản lý danh bạ hoặc trình quay số điện thoại mà có các thành phần người khác có thể tạo mới giao diện và mở rộng chức năng thay vì tạo lại chúng.

Những dịch vụ sau là những dịch vụ kiến trúc cơ bản nhất của tất cả các ứng dụng, cung cấp một framework cho mọi mọi phần mềm được xây dựng:

* View: Được sử dụng để tạo lập các giao diện người dùng cho các Activity như: Lists, Grids, Text boxes, Buttons, và nhúng một trang web và giao diện của android.
* [Resource Manage](https://developer.android.com/guide/topics/resources/overview.html): Hỗ trợ truy cập đế các thành phần không thuộc mã nguồn như là chuỗi ký tự, đồ họa được đặt bên ngoài.
* Notification Mamager: Cho phép tất cả các ứng dụng hiển thị thông báo tùy chỉnh trên thanh trạng thái của thiết bị.
* Actitvity Manager: Điều khiển vòng đời c ủa các Activity bao gồm cả quản lý các tầng Activity.
* Content Provider: Cho phép ứng dụng chia sẻ dữ liệu giữa các ứng dụng.

### 1.2.6 System Apps

****

Hình 1.5 Tầng ứng dụng

Đây là lớp trên cùng của kiến trúc nền tảng Android. Android sẽ hoạt động với một bộ các ứng dụng bao gồm ứng dụng thư điện tử, gửi tin nhắn, lịch, bản đồ, trình duyệt web, danh bạ v.v… Tất cả các ứng dụng được viết bằng ngôn ngữ Java. Các ứng dụng này có thể được cung cấp sẵn hoặc được phát triển bởi những lập trình viên.

# CHƯƠNG 4: NGÔN NGỮ LẬP TRÌNH

## 4.1 Ngôn ngữ lập trình Java

### 4.1.1 Java là gì?



Hình 4.1 Biểu tượng của Java

Java là một Ngôn ngữ lập trình và cũng là một Platform.

- **Ngôn ngữ lập trình:** Java là một ngôn ngữ lập trình có tính bảo mật cao, hướng đối tượng, bậc cao và mạnh mẽ.

- **Platform:** Bất cứ môi trường phần cứng hoặc phần mền nào mà trong đó một chương trình chạy, thì được biết đến như là một Platform. Với môi trường runtime riêng cho mình là JRE và API, Java được gọi là Platform.

### 4.1.2 Lịch sử và các nét đặc trưng của Java:

Ngôn ngữ Java được tạo ra bởi James Gosling, Mike Sheridan và Patrick Naughton khởi xướng dự án về ngôn ngữ Java trong tháng 6/1991. Họ được gọi là Green Team.

Đầu tiên Java được thiết kế dành cho các hệ thống nhỏ, có thể nhúng vào trong các thiết bị điện tử như set-top box và được gọi là Greentalk bởi James Gosling và file ngôn ngữ với đuôi là ".gt", sau đó được đổi tên là Oak.

Đến năm 1995, Oak được đổi tên thành Java bởi Jamas Gosling tại Sun Microsystems (bât giờ là công ty con của Oracle Coporation). Cũng trong năm này, tạp chí Time bình chọn Java là một trong 10 sản phẩm tốt nhất năm 1995.

Ngôn ngữ lập trình Java dựa trên nền tảng ngôn ngữ C++ nhưng đơn giản hơn do đã lược bỏ bớt những đặc điểm hiếm khi được sử dụng và rất khó để sử dụng như con trỏ tường minh, nạp chồng toán tử,... Và Java cũng có một bộ dọn rác tự động(Garbage Collection) để tự động dọn dẹp các đối tượng không được tham chiếu.

### 4.1.3 Tại sao lại chọn ngôn ngữ lập trình Java

* Là một ngôn ngữ lập trình hướng đối tượng
* Số lượng hàm dùng sẵn vô cùng phong phú
* Có các công cụ phát triển mạnh mẽ hỗ trợ như Esclipse, Netbeans,..
* Bộ sưu tập thư viện mã nguồn mở vô cùng đa dạng
* Hỗ trợ cộng đồng tuyệt vời
* Java là miễn phí
* Hỗ trợ tài liệu xuất sắc – Javadocs
* Là nền tảng độc lập
* Có mặt ở khắp mọi nơi
* Chạy được trên mọi nền tảng
* Là ngôn hỗ trợ viết ứng dụng trên Android

### 4.1.4 Android Studio IDE

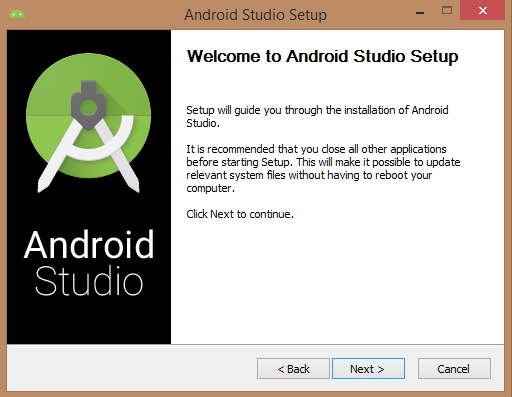
* 1. Android Studio là gì?

Android Studio là IDE chính thức để phát triển ứng dụng Android. Nó được tạo ra dựa trên nền tảng IntelliJ IDEA. Để sử dụng được Android Studio trên Windows hoặc Linux bạn phải cài đặt JDK, trên Mac OS thì Android Studio đã được tích hợp sẵn JDK.

* 1. Cài đặt Android Studio

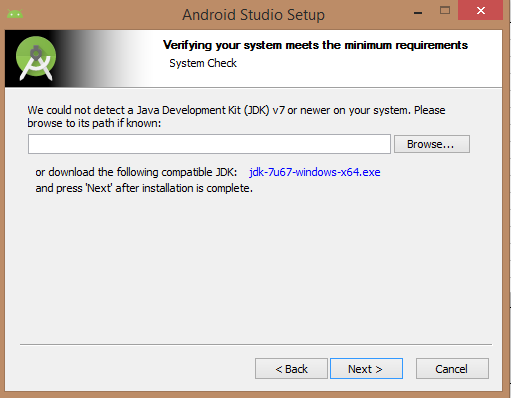
Bước 1: Tải, cài đặt JDK và thiết lập các biến môi trường.

Bước 2: Khởi chạy file cài đặt Android Studio.



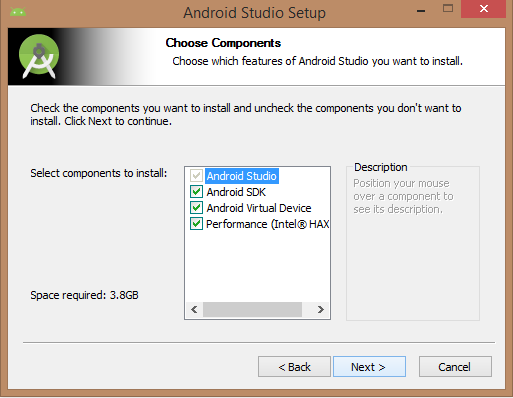
*Hình 1.1. Cài đặt Android Studio bước 2*

Bước 3: Nếu Android Studio không tìm được đường dẫn JDK đã được cài đặt trước đó thì sẽ hiển thị màn hình để người dùng tự chọn đường dẫn.



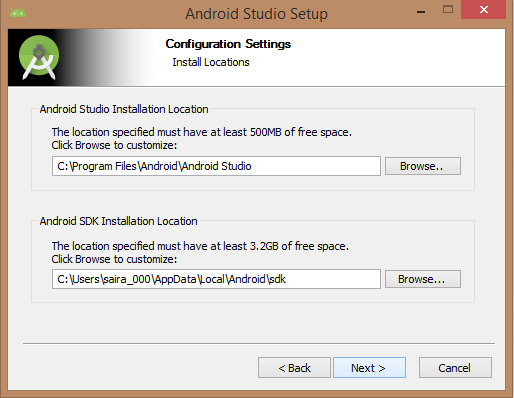
*Hình 1.2. Cài đặt Android Studio bước 3*

Bước 4: Chọn các thành phần cần thiết để cài đặt.



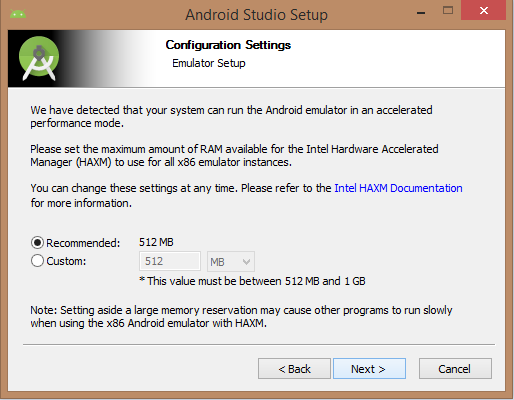
*Hình 1.3. Cài đặt Android Studio bước 4*

Bước 5: Chọn vị trí cài đặt Android Studio và nơi cài đặt Android SDK.



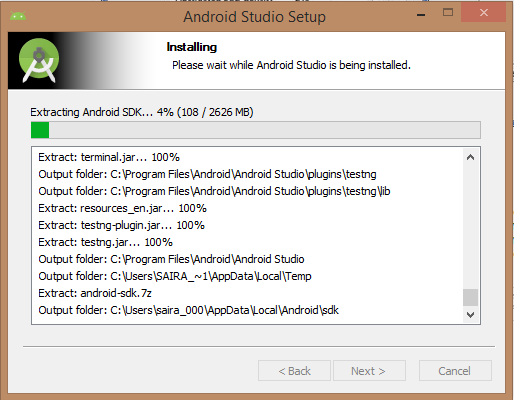
*Hình 1.4. Cài đặt Android Studio bước 5*

Bước 6: Xác định bộ nhớ RAM cho Android Emulator và nhấn Next.



*Hình 1.5. Cài đặt Android Studio bước 6*

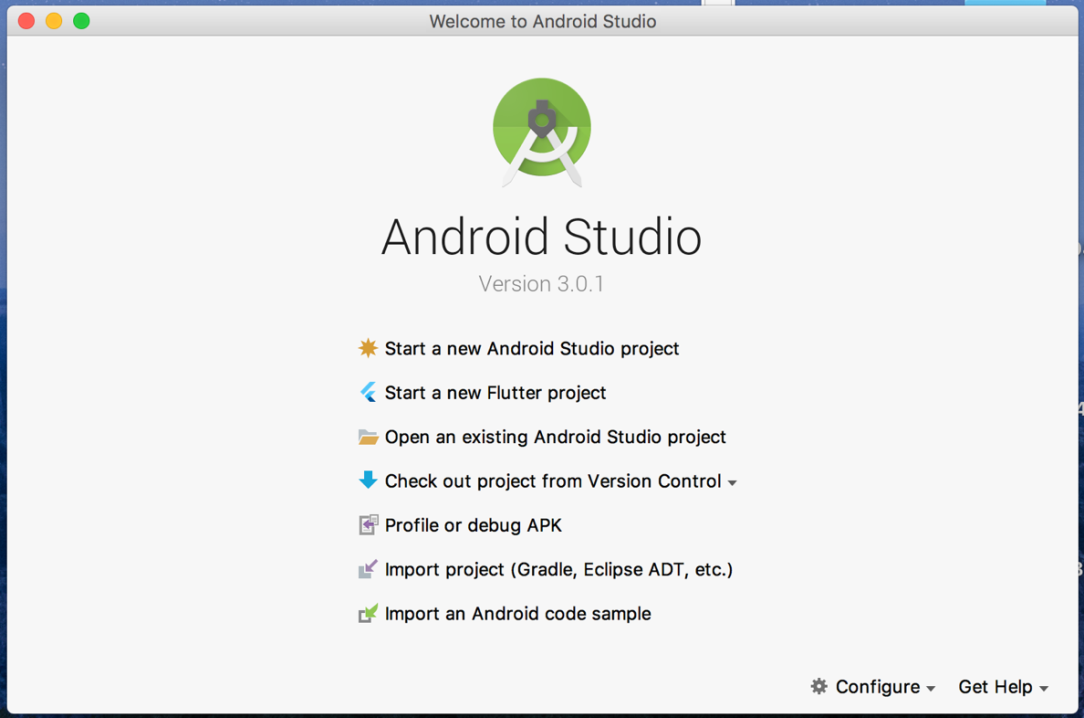
Bước 7: Quá trình cài đặt diễn ra và hoàn tất.



*Hình 1.6. Cài đặt Android Studio bước 7*

* 1. Tạo dự án mới bằng Android Studio

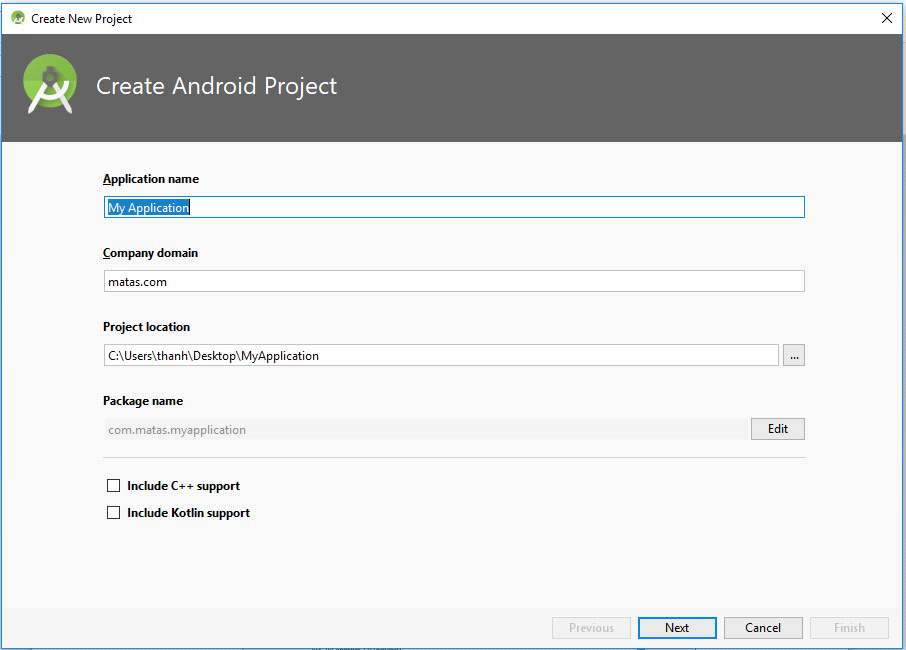
Khi khởi động Android Studio lên, giao diện hiện ra cho phép bạn tạo một project mới, hoặc mở lại các project gần đây.



*Hình 1.7. Giao diện Welcome*

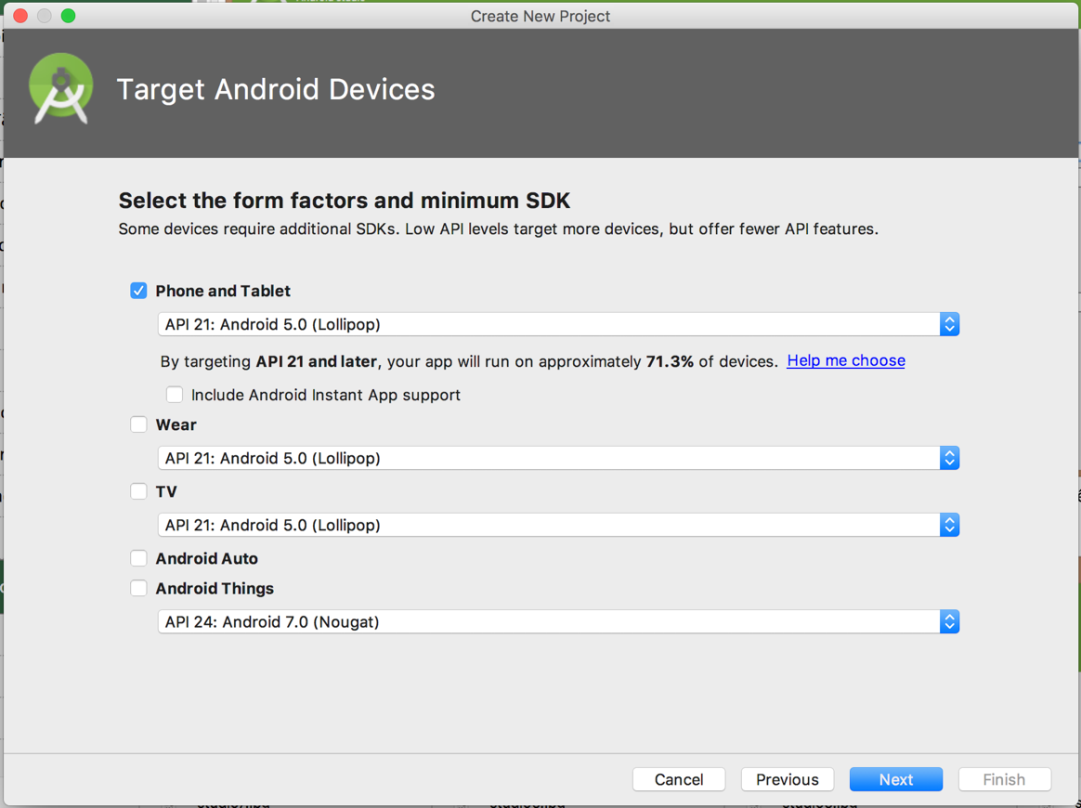
Tại giao diện Welcome, bạn có thể tạo mới một project bằng cách chọn "Start a new Android Studio project".

Sau khi chọn "Start a new Android Studio project", Android Studio sẽ yêu cầu bạn nhập vào một số thông tin của project như: Application name, Company domain, Project location, có bao gồm Kotlin hay C++ hay không.



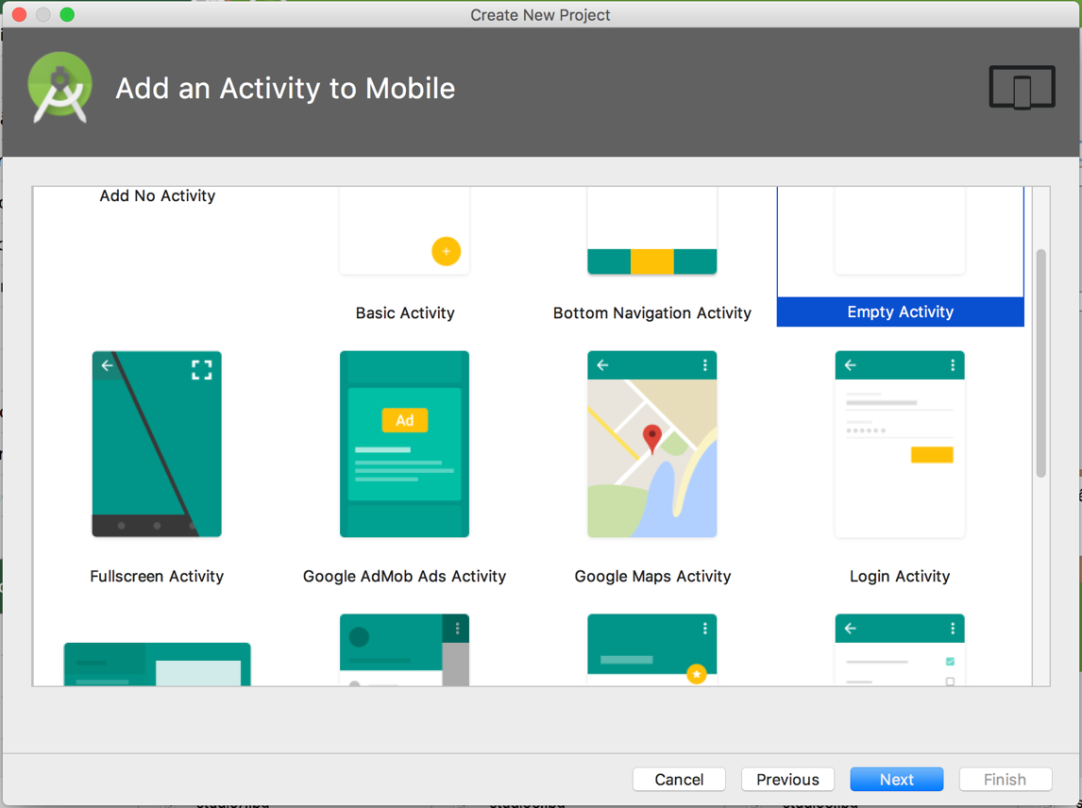
*Hình 1.8. Tạo project mới bước 1*

Tiếp theo đó bạn chọn phiên bản Android và những thiết bị được hỗ trợ trong Project.



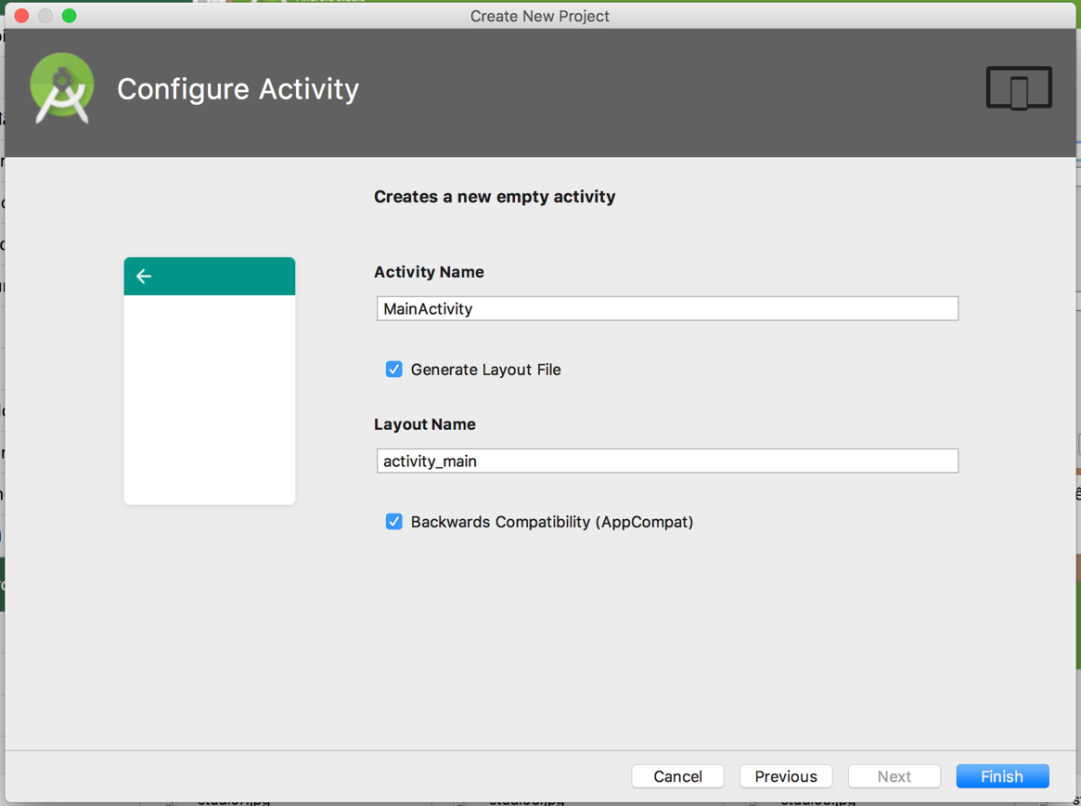
*Hình 1.9. Tạo project mới bước 2*

Tiếp theo bạn chọn Activity muốn khởi tạo cho Project mới.



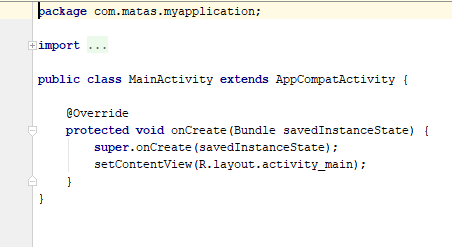
*Hình 1.10. Tạo project mới bước 3*

Đặt tên cho Activity và như vậy bạn đã tạo xong một Project mới.



*Hình 1.11. Tạo project mới bước 4*

* 1. Chương trình đầu tiên bằng Java trên Android: "Hello World"

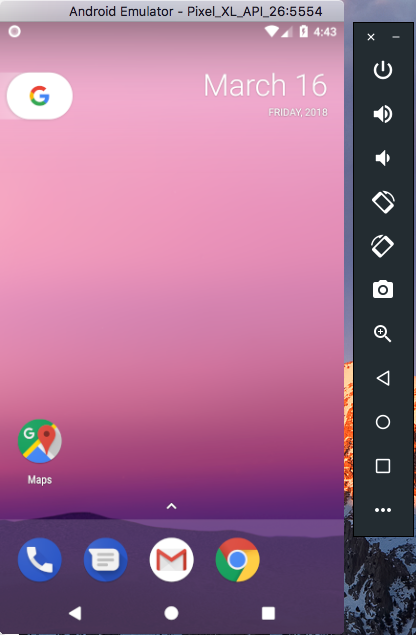


* 1. Android Emulator

Android Emulator là một phần trong bộ công cụ kèm theo của phần mềm Android Studio. Android Emulator chứa Android SDK cho phép bạn giả lập môi trường Android nhằm phục vụ cho việc kiểm thử ứng dụng được viết ra trước khi kiểm thử ứng dụng trên thiết bị thật.

Android Emulator cho phép bạn cài đặt nhiều thiết bị Android khác nhau như với nhiều phiên bản Android và kích thước màn hình khác nhau.

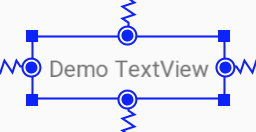
Với Android Emulator, bạn có thể kiểm thử ứng dụng của bạn về thiết kế giao diện, về tính năng của ứng dụng, từ đó có thể khắc phục các lỗi phát sinh, tối ưu hóa ứng dụng trước khi bạn đem ứng dụng lên thiết bị thật.



*Hình 1.12. Giao diện Android Emulator cho Pixel XL*

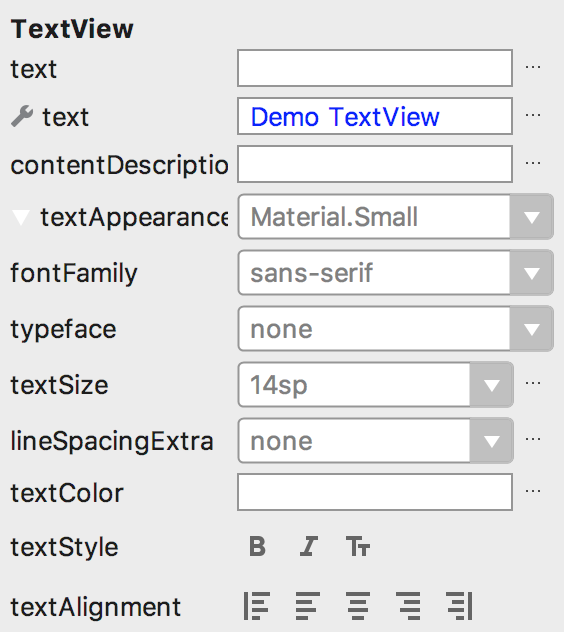
### 4.1.5 Một số đối tượng cơ bản

* 1. TextView – Button – EditText
* TextView: dùng để hiển thị một đoạn text có tính cố định ít thay đổi.



*Hình 2.1. UILabel*

TextView cho phép người dùng thay đổi một số thuộc tính để hiển thị nội dung ra giao diện cho phù hợp như màu chữ, font và kích thước, ...



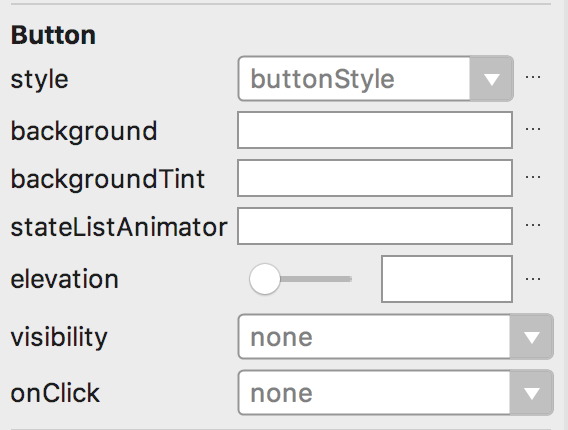
*Hình 2.2. Tùy chỉnh TextView*

* Button: để người dùng tương tác với ứng dụng nhằm tạo ra một sự kiện nào đó.



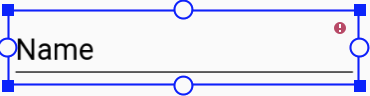
*Hình 2.3. Button*

Với button, bạn có thể lựa chọn một số định dạng button có sẵn, hoặc tùy chọn một dạng button khác theo ý bạn. Bạn còn có thể tùy chỉnh font chữ, màu chữ, màu nền của button hoặc thay thế button theo mộ hình ảnh button được thiết kế trước bằng các thuộc tính như của TextView.



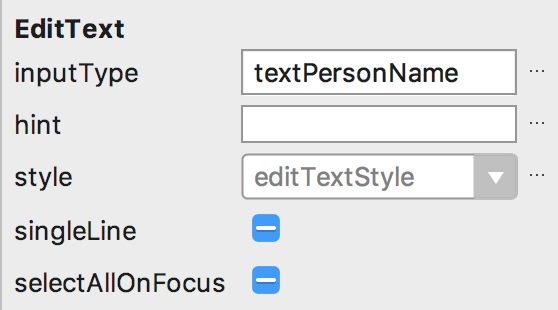
*Hình 2.4. Tùy chỉnh button*

* EditText: được sử dụng để người dùng nhập dữ liệu đầu vào cho ứng dụng.



*Hình 2.5. EditText*

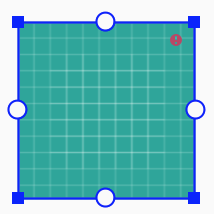
EditText cho phép tùy chỉnh chữ như TextView. Hơn nữa, bạn có thể tùy chỉnh định dạng nội dung gợi ý cho người dùng (hint).



*Hình 2.6. Tùy chỉnh EditText*

* 1. ImageView

ImageView dùng để hiển thị hình ảnh, có thể sử dụng các hiệu ứng animation để hiển thị hình ảnh.



*Hình 2.7. Đối tượng ImageView*

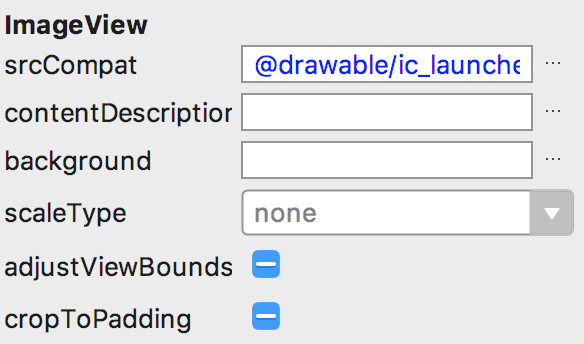
* Hiển thị ảnh có sẵn trên máy:

Bước 1: Add ImageView vào file layout.

* + Mở file layout
  + Kéo thả đối tượng ImageView vào layout

Bước 2: Load file hình vào ImageView

* Cách 1: Chọn hình từ menu tùy chỉnh của “Image View”



*Hình 2.8. Chọn hình từ hộp thoại Image View*

* Cách 2: Gọi phương thức sau để load file hình vào ImageView.



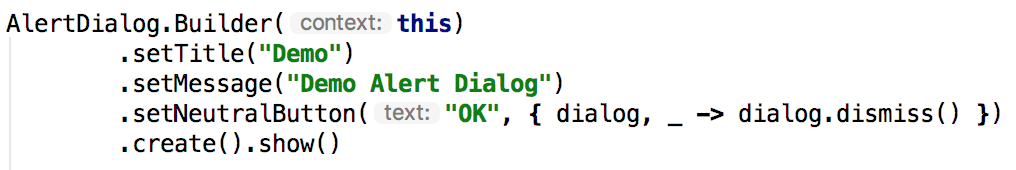
* Hiển thị ảnh từ một URL:

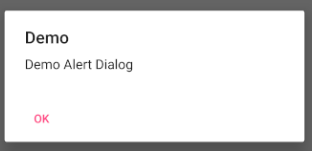


* 1. AlertDialog và Dialog

AlertDialog là đối tượng được sử dụng để hiển thị nội dung thông báo cho người dùng, Dialog là đối tượng dialog có thể tùy chỉnh layout.

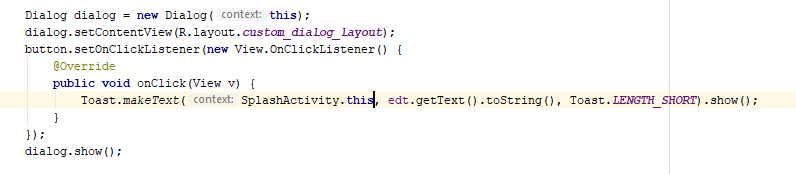
* AlertDialog

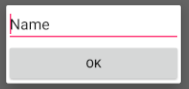




*Hình 2.9. Hiển thị thông báo bằng AlertDialog*

* Dialog





*Hình 2.10. Hiển thị màn hình custom bằng Dialog*

* 1. RecyclerView

Về mặt lý thuyết, RecyclerView là một viewGroup mới vừa được giới thiệu trong phiên bản android API 21, nó là sự kế thừa và nâng cao từ ListView và GridView.

Một trong những ưu điểm nổi trội của RecyclerView là nó có khả năng mở rộng tốt hơn, nó có thể hiển thị dữ liệu theo cả chiều ngang và chiều dọc.

Các thành phần của RecyclerView bao gồm:

* *RecyclerView.Adapter:* dùng để thu thập và hiển thị dữ liệu lên view.
* *RecyclerView.ViewHolder:* dùng để lưu trữ các tham chiếu của view.
* *LayoutManager:* để chỉ định chiều hiển thị của 1 item là ngang hay dọc.
* *ItemAnimator:* dùng để tạo ra hiệu ứng chuyển động cho view.

Những điều khiến RecyclerView khác biệt hơn so với các view tiền nhiệm:

* Bắt buộc phải sử dụng *RecyclerView.ViewHolder* để lưu trữ lại các tham chiếu của view, giúp cho view có thể tái sử dụng lại các tham chiếu đó mà không phải sử dụng lại các view cũ, giúp giải phóng bộ nhớ tốt hơn*.*
* Chỉ cần sử dụng *RecyclerView.LayoutManager* để chỉ định cách cuộn dữ liệu theo chiều dọc hoặc ngang thay vì phải sử dụng một custom view.
* Thiết lập animations cho view dễ dàng hơn bằng *RecyclerView.ItemAnimator* thay vì sử dụng một animation bổ sung.
* Adapter:

Đây là thành phần xử lý data collection và gắn những dữ liệu này vào các Item của RecyclerView.

Khi khởi tạo custom adapter thì phải override hai phương thức:

* onCreateViewHolder: phương thức này được gọi ra khi ViewHolder được khởi tạo, dùng để tạo view mới cho RecyclerView. Nếu như view đã được tạo thì phương thức này sẽ không được gọi lại.
* onBindViewHolder: phương thức này dùng để gắn data trong Viewholder vào RecyclerView.
* LayoutManager:

Chức năng chính dùng để sắp xếp các item trong RecyclerView.

Các item cuộn dọc hay ngang phụ thuộc vào việc chúng ta cài đặt.

LayoutManager bao gồm các class con:

* LinearLayoutManager: cuộn các item theo chiều ngang hay dọc.
* GridLayoutManager: sắp xếp các item giống như GridView.
* StaggeredGridLayoutManager: sắp xếp các item trong view dạng grid so lẻ.

# KẾT LUẬN

## KẾT QUẢ ĐẠT ĐƯỢC

### 1.1 Chương trình đã đạt được

Có khả năng đáp ứng các nhu cầu sau:

Quản lý hồ sơ của bệnh nhân bao gồm quản lý thông tin của bệnh nhân cũng như bệnh án của một bệnh nhân, lưu dữ liệu cho việc tính toán viện phí.

Quản lý khoa: bao gồm các thông tin về khoa.

Quản lý nhân viên: bao gồm quản lý các thông tin cá nhân của nhân viên.

Quản lý phòng – giường bệnh: bao gồm thông tin quản lý về loại phòng, giường, giá từng loại phòng.

Quản lý Danh mục Thuốc: bao gồm quản lý thông tin về thuốc.

Quản lý Danh mục Dịch Vụ: bao gồm quản lý thông tin về dịch vụ cho bệnh nhân.

Có chức năng khám bệnh, lập toa thuốc, chỉ định và thực hiện dịch vụ hỗ trợ cho mỗi lần khám bệnh của bác sỹ.

Giao diện dễ sử dụng và thân thiện với người dùng.

### 1.2 Khả năng ứng dụng

Có khả năng hỗ trợ trong việc quản lý bệnh nhân điều trị nội trú trong bệnh viện. Áp dụng vào tình hình thực tế của bệnh viện.

### 1.3 Kiến thức đạt được

Trong quá trình thực hiện đề tài, em đã củng cố được rất nhiều kiến thức về phân tích thiết kế hệ thống thông tin, thiết kế và ứng dụng kỹ năng viết phần mềm, kiến thức về cơ sở dữ liệu, hệ quản trị cơ sở dữ liệu,..

Tăng cường khả năng tiếp cận thực tế, khả năng tự tìm tòi tài liệu và học hỏi cũng như trao đổi kiến thức chuyên ngành và thực tế đối với giáo viên hướng dẫn và bạn bè.

Rèn luyện được tính cẩn thận, tư duy khi giải quyết vấn đề.

## 2. HẠN CHẾ, KHÓ KHĂN

* Chương trình chưa có thời gian để kiểm chứng thực tế.
* Khó tiếp cận với dữ liệu thực tế của các bệnh viện
* Do thời gian có hạn nên nội dung báo cáo thực tập tốt nghiệp không tránh khỏi những sai sót, em rất mong nhận được sự góp ý của quý thầy cô.

# TÀI LIỆU THAM KHẢO

**Tiếng việt:**

Bài giảng Phân tích thiết kế hệ thống thông tin, biên soạn bởi ThS.Lương Thị Ngọc Khánh, khoa Công nghệ thông tin, đại học Tôn Đức Thắng.

**Danh mục các website tham khảo:**

1. <http://laptrinhlibrary.blogspot.com/2016/12/share-bunifu-net-framework-crack-mien.html>
2. <http://doc.edu.vn/tai-lieu/giao-trinh-ve-sql-server-58929/>
3. <https://drive.google.com/file/d/0B5_uFw-fH1KFdTBFMHRiMzB2SUk/view>
4. <https://www.youtube.com/>