

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 : Mai Đức Anh

Giảng viên hướng dẫn: Ths. Nguyễn Thị Xuân Hương

HẢI PHÒNG – 2020

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

-----

**XÂY DỰNG ỨNG DỤNG HỖ TRỢ LÊN THỰC ĐƠN  
KHOA HỌC THEO LƯỢNG CALO**

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

**Sinh viên : Mai Đức Anh**

**Giảng viên hướng dẫn: Ths. Nguyễn Thị Xuân Hương**

**HẢI PHÒNG – 2020**

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

---

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

**Sinh viên: Mai Đức Anh**

**Mã SV: 1412101038**

**Lớp** : CT1802

**Ngành** : Công nghệ Thông tin

**Tên đề tài:** Xây dựng ứng dụng hỗ trợ lên thực đơn khoa học theo lượng calo.

# 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 về hệ điều hành Android,
- Tìm hiểu về môi trường lập trình Android.
- Xây dựng ứng dụng hỗ trợ lên thực đơn khoa học theo lượng calo trong đó cho phép:
  - o Người dùng nhập thông tin
  - o Tạo danh mục thực phẩm
  - o Lên thực đơn của một ngày theo lượng calo
  - o Hiện thị thực đơn

### b. Các yêu cầu cần giải quyết

- Trình bày các nội dung tìm hiểu được về hệ điều hành Android, môi trường lập trình và phát triển ứng dụng Android.
- Cài đặt, cấu hình ứng dụng

## 2. Các tài liệu, số liệu cần thiết

- Tài liệu tham khảo về Hệ điều hành Android, lập trình phát triển ứng dụng trên Android.
- Tài liệu về các danh mục thực phẩm phổ biến cho người Việt Nam và lượng calo cần thiết cho cơ thể.

## 3. Địa điểm thực tập tốt nghiệp

- Trường Đại học Quản lý và Công nghệ Hải Phòng.

## CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

**Họ và tên** : Nguyễn Thị Xuân Hương

**Học hàm, học vị** : Thạc sỹ

**Cơ quan công tác** : Trường Đại học Quản lý và Công nghệ Hải Phòng

**Nội dung hướng dẫn:**

- Tìm hiểu về hệ điều hành Android
- Tìm hiểu về môi trường lập trình Android
- Xây dựng ứng dụng hỗ trợ lên thực đơn khoa học theo lượng calo.

Đề tài tốt nghiệp được giao ngày 30 tháng 03 năm 2020

Yêu cầu phải hoàn thành xong trước ngày 30 tháng 06 năm 2020

Đã nhận nhiệm vụ ĐTTN

*Sinh viên*

Đã giao nhiệm vụ ĐTTN

*Giảng viên 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 GIẢNG VIÊN HƯỚNG DẪN TỐT NGHIỆP**

Họ và tên giảng viên: Nguyễn Thị Xuân Hương

Đơn vị công tác: Khoa Công nghệ Thông tin – Trường Đại học quản lý và Công nghệ Hải Phòng.

Họ và tên sinh viên: Mai Đức Anh                      Ngành: Công nghệ thông tin

Nội dung hướng dẫn: Xây dựng ứng dụng hỗ trợ lên thực đơn khoa học theo lượng calo.

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

- Sinh viên chăm chỉ, chịu khó học hỏi, tự giác làm việc.
- Chủ động trong công việc và liên hệ với giảng viên.

### **2. Đánh giá chất lượng của đồ án/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...)**

- Sinh viên đã tìm hiểu về các kiến thức mới chưa được học trong chương trình về hệ điều hành Android và môi trường lập trình Android. Từ đó cài đặt hệ thống và xây dựng ứng dụng thử nghiệm.
- Sinh viên đã xây dựng ứng dụng hỗ trợ lên thực đơn khoa học theo lượng calo, trong đó cho phép người dùng nhập thông tin và lượng calo cần tiêu thụ một ngày, tạo danh mục thực phẩm, Lên thực đơn theo ngày và hiển thị thực đơn theo lượng calo cần. Chương trình ban đầu đã thực hiện được các yêu cầu cơ bản. Sinh viên đã nỗ lực tự tìm hiểu và tự học lập trình trên hệ điều hành Android và xây dựng được ứng dụng.
- Đồ án đáp ứng cơ bản những được yêu cầu đặt ra. Đề nghị cho phép sinh viên được bảo vệ trước hội đồng chấm tốt nghiệp ngành Công nghệ Thông tin.

### **3. Ý kiến của giảng viên hướng dẫn tốt nghiệp**

Đạt  Không đạt  Điểm:.....

*Hải Phòng, ngày 28 tháng 06 năm 2020*

**Giảng viên 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: Đỗ Văn Chiểu

Đơn vị công tác: Khoa Công nghệ thông tin, trường ĐH Quản lý và Công nghệ Hải Phòng

Họ và tên sinh viên: Mai Đức Anh      Ngành: Công nghệ thông tin

Đề tài tốt nghiệp: Xây dựng ứng dụng hỗ trợ lên thực đơn khoa học theo lượng calo

**1. Phần nhận xét của giảng viên chăm phản biện**

.....

.....

.....

.....

.....

.....

.....

.....

**2. Những mặt còn hạn chế**

.....

.....

.....

.....

.....

.....

.....

.....



### 3. Ý kiến của giảng viên chấm phản biện

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

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

**Giảng viên chấm phản biện**

*(Ký và ghi rõ họ tên)*

## LỜI MỞ ĐẦU

Với sự phát triển mạnh mẽ của công nghệ thông tin cùng các thiết bị di động nên phần lớn người dùng đều sử dụng máy vi tính hoặc điện thoại di động để làm việc và giải trí. Điều đó đã làm tăng năng suất, hiệu quả lao động và mang lại nhiều tiện ích cho người dùng trong cuộc sống hiện đại. Hầu hết, mỗi người dùng đều sử dụng một điện thoại thông minh, ngoài việc gọi điện hay nhắn tin thì họ còn sử dụng để tải các ứng dụng có sẵn phục vụ cho việc lưu trữ dữ liệu và xử lý các thông tin cá nhân khác. Do đó việc xây dựng các ứng dụng cho điện thoại di động đang là một ngành công nghiệp mới đầy tiềm năng và hứa hẹn nhiều sự phát triển vượt bậc của ngành khoa học kỹ thuật.

Phần mềm ứng dụng cho điện thoại di động hiện nay rất đa dạng và phong phú trên các hệ điều hành di động. Các hệ điều hành J2ME, Android, IOS, Hybrid, Web bases Mobile Application đã rất phát triển trên thị trường truyền thông di động.

Trong vài năm trở lại đây, hệ điều hành Android ra đời với sự kế thừa những ưu việt của các hệ điều hành ra đời trước và sự kết hợp của nhiều công nghệ tiên tiến nhất hiện nay. Android đã nhanh chóng là đối thủ cạnh tranh mạnh mẽ với các hệ điều hành trước đó và đang là hệ điều hành di động của tương lai và được nhiều người ưa chuộng nhất.

Trong cuộc sống hiện đại, người dùng ngày một quan tâm đến sức khỏe và vóc dáng bản thân hơn. Họ không chỉ quan tâm đến việc rèn luyện thể dục thể thao mà việc cân bằng lượng dinh dưỡng trong các bữa ăn hàng ngày cũng rất quan trọng. Vì vậy em chọn đề tài **“Xây dựng ứng dụng di động tính lượng calories cho thực đơn hàng ngày”** để giúp người dùng lựa chọn các thực phẩm theo lượng calo cần thiết đảm bảo cho việc chăm sóc sức khỏe và vóc dáng của người dùng,

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến quý thầy cô Trường Đại Học Dân Lập Hải Phòng, những người đã dìu dắt em tận tình, đã truyền đạt cho em những kiến thức và bài học quý báu 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 cô giáo Ths.Nguyễn Thị Xuân Hương, cô đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình làm tốt nghiệp. Với sự chỉ bảo của cô, em đã có những 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 tới tất cả bạn bè, đặc biệt là các bạn trong lớp CT1802 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 06 năm 2020*

Sinh viên

**Mai Đức Anh**

## MỤC LỤC

**LỜI MỞ ĐẦU**2

**LỜI CẢM ƠN**3

**MỤC LỤC**4

**CHƯƠNG 1: GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH ANDROID**6

- 1.1. Giới thiệu về hệ điều hành Android6
- 1.2. Kiến trúc cơ bản của hệ điều hành Android9
  - 1.2.1. Nhân Linux10
  - 1.2.2. Thư viện10
  - 1.2.3. Thực thi11
  - 1.2.4. Nền tảng Android11
  - 1.2.5. Tầng ứng dụng12

**CHƯƠNG 2: MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG ANDROID STUDIO, SQLITE**13

- 2.1. Giới thiệu ứng dụng Android Studio13
- 2.2. Cài đặt môi trường lập trình Android13
  - 2.2.1. Cài đặt JAVA JDK13
  - 2.2.2 Cài đặt Android Studio15
  - 2.2.3 Máy ảo Android Genymotion18
- 2.3. Thành phần trong một dự án ANDROID19
  - 2.3.1. Tập cấu hình Android19
  - 2.3.2. Thư mục Java22
  - 2.3.3. Thư mục Res22
  - 2.3.4. Tập Grade Scripts23
- 2.4. Thành phần giao diện24
  - 2.4.1. View group24
  - 2.4.2. View25
- 2.5. Vòng đời ứng dụng android27
- 2.6. Lớp Intent29
  - 2.6.1. Khái niệm29
  - 2.6.2. Các loại Intent29
- 2.7. Share preferences34
  - 2.7.1. Khái niệm34
  - 2.7.2. Cách sử dụng34

## 2.8. Hiệu ứng trong android36

### 2.8.1. Hiệu ứng cơ bản36

### 2.8.2. Cách sử dụng36

## 2.9. SQLite37

### 2.9.1. Giới thiệu SQLite37

### 2.9.2 Cấu hình SQLite37

## **CHƯƠNG 3: CHƯƠNG TRÌNH THỰC NGHIỆM45**

### 3.1. Giới thiệu45

### 3.2. Khái niệm về calories45

#### 3.2.1. Khái niệm45

#### 3.2.2. Chức năng của calo46

#### 3.2.3. Lượng calo cần thiết hàng ngày cho từng người:46

#### 3.2.4. Bảng dinh dưỡng của các thực phẩm.47

### 3.3. Bài toán lên thực đơn hàng ngày theo lượng calo một cách khoa học49

### 3.4. Phân tích bài toán49

#### 3.4.1. Ngữ cảnh của hệ thống49

#### 3.4.2. Mô hình chức năng50

#### 3.4.3. Cơ sở dữ liệu51

### 3.5 Cài đặt chương trình52

#### 3.5.1. Thiết bị và môi trường lập trình52

#### 3.5.2. Ứng dụng hỗ trợ lên thực đơn khoa học theo lượng calo52

### 3.6. Các yêu cầu đối với người dùng hệ thống.54

## **KẾT LUẬN56**

# CHƯƠNG 1: GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH ANDROID

## 1.1. Giới thiệu về hệ điều hành Android

Android 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 Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005.

Chính mã nguồn mở của Android cùng với tính không ràng buộc nhiều đã cho phép các nhà phát triển thiết bị di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị.

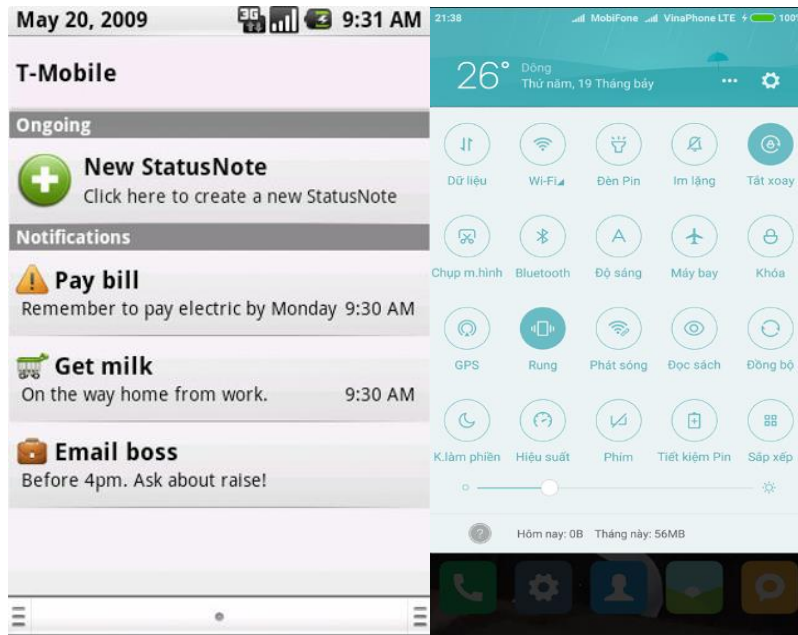
Nhờ yếu tố mở, dễ dàng tinh chỉnh cùng sự phát triển nhanh chóng đã khiến hệ điều hành này dần trở nên phổ biến, kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng nhưng giờ đây Android đã xuất hiện trên các smart TV, máy chơi game và một số thiết bị điện tử khác.

Android bắt đầu với bản beta đầu tiên vào tháng 11 năm 2007 và phiên bản thương mại đầu tiên, Android 1.0, được phát hành vào tháng 9 năm 2008. Kể từ tháng 4 năm 2009, phiên bản Android được phát triển, đặt tên theo chủ đề bánh kẹo và phát hành theo thứ tự bảng chữ cái: Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, Kitkat, Lollipop, Marshmallow, Nougat, và bây giờ là Oreo.

Kỷ nguyên của Android chính thức bắt đầu vào ngày 22 tháng 10 năm 2008, khi chiếc điện thoại T-Mobile G1 bắt đầu được bán ra tại Mỹ. Vào thời gian đầu, rất nhiều tính năng cơ bản bị thiếu sót như: bàn phím ảo, cảm ứng đa điểm và tính năng mua ứng dụng vẫn chưa xuất hiện. Tuy nhiên, một số tính năng cũng như giao diện đặc sản của hệ điều hành này đã khởi nguồn từ chiếc G1 và trở thành những yếu tố không thể thiếu trên Android sau này.

Sự phát triển của hệ điều hành Android:

- **Thanh thông báo vuốt từ trên xuống (Notification bar):** Ngay từ những ngày đầu tiên của Android, thanh thông báo này đã đánh dấu một bước quan trọng mà trước đây chưa hề có hệ điều hành nào làm được - đưa tất cả thông tin tin nhắn, tin thoại hoặc các cuộc gọi nhỡ chỉ với thao tác vuốt xuống.



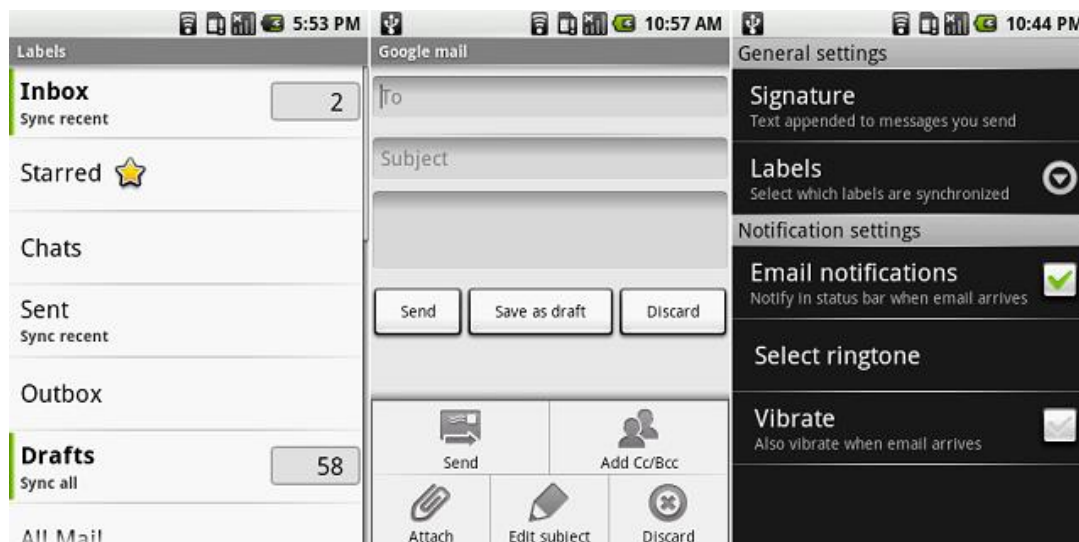
Hình 1.1.1. Thanh thông báo ở phiên bản thời kỳ đầu(trái) so với phiên bản Android 8.1(phải)

• Màn hình chính (Home Screen) và các widget: Một điểm khác biệt giữa Android so với các hệ điều hành khác là phần màn hình chính của mình. Bên cạnh việc thay đổi được hình nền, Android còn cho phép người dùng tùy biến màn hình chính của mình với nhiều widgets kèm theo, chẳng hạn như đồng hồ, lịch, trình nghe nhạc, đưa các icon ứng dụng ra ngoài hoặc thậm chí có thể can thiệp sâu hơn để thay đổi toàn bộ giao diện màn hình Home Screen này.



Hình 1.1.2. Màn hình chính của Android 1.0(trái) so với Android 8.1(phải)

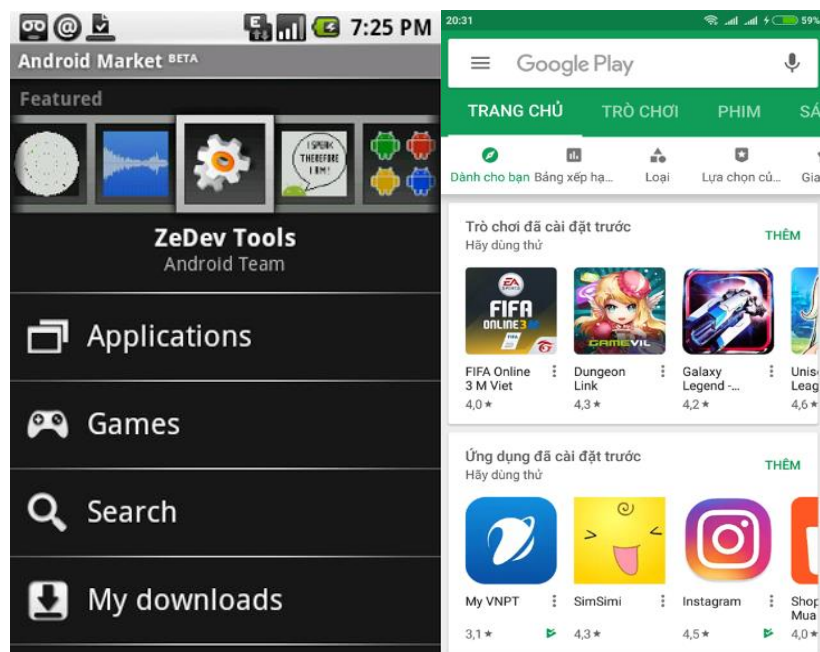
- Đồng bộ và tích hợp chặt chẽ với Gmail: Vào thời điểm điện thoại G1 được bán ra, Gmail đã hỗ trợ giao thức POP và IMAP để tích hợp với các trình email trên di động. Tuy nhiên, lúc bấy giờ không có bất kì sản phẩm nào có thể hỗ trợ được hoàn toàn những tính năng ưu việt này của Gmail. Mãi cho đến khi Android 1.0 xuất hiện, vấn đề này đã được khắc phục và G1 trở thành chiếc điện thoại mang lại trải nghiệm Gmail tốt nhất trên thị trường lúc bấy giờ.



*Hình 1.1.3. Gmail trên Android thời kỳ đầu*

- Kho ứng dụng Android Market: Thật khó có thể tưởng tượng một chiếc smartphone mà không hề có kho ứng dụng, nhưng vào thời điểm Android mới ra mắt, gần như không có bất kì điện thoại nào có kho ứng dụng nào được tích hợp và chính Android đã mở đầu cuộc cách mạng ứng dụng di động này. Android Market trên G1 thời bấy giờ có rất ít ứng dụng và giao diện cực kỳ đơn giản, hơn nữa tính năng mua ứng dụng trên phiên bản này vẫn chưa được xuất hiện mãi cho đến năm sau - những vấn đề này dễ hiểu vì thời điểm này Android chỉ mới được khai sinh nên mọi thứ còn khá thô sơ.



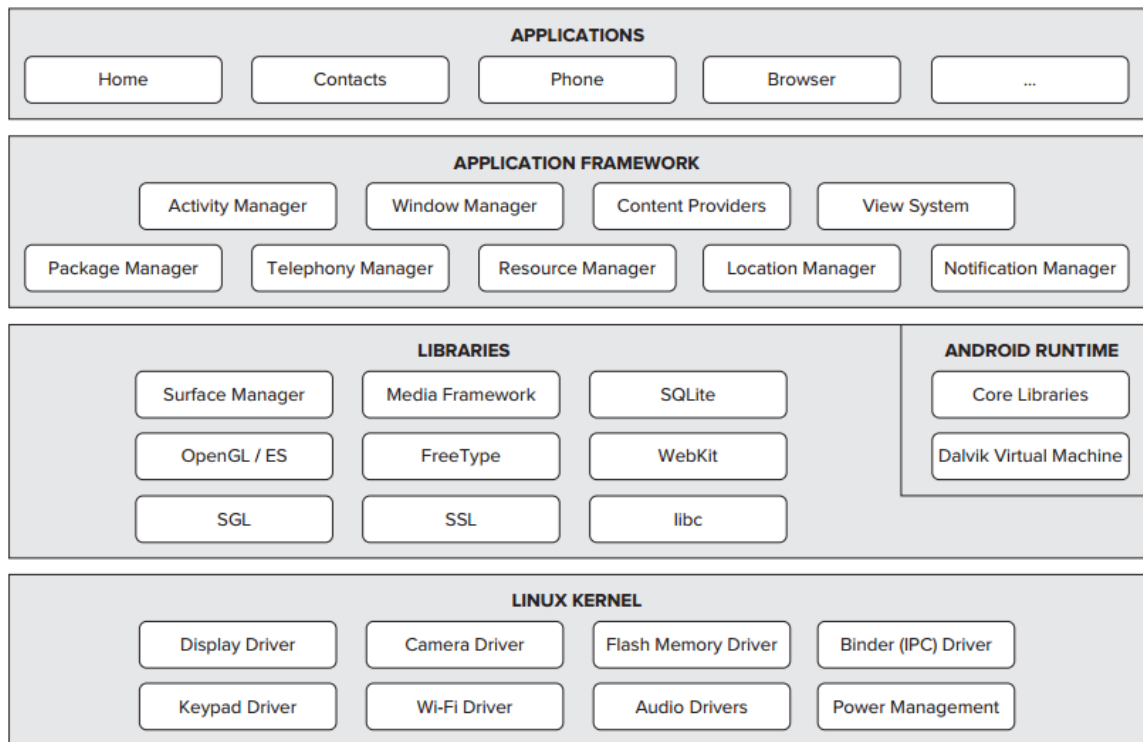


Hình 1.1.4. Hình ảnh Store của Android 1.0(trái) và Android 8.1(phải)

- Giao diện: Google đã phát triển giao diện Android phiên bản 1.0 với sự hỗ trợ từ TAT, viết tắt từ The Astonishing Tribe, một công ty thiết kế tương tác của Thụy Điển. Dấu ấn rõ ràng nhất mà TAT để lại trên phiên bản Android từ phiên bản 1.0 cho đến 2.2 chính là widget đồng hồ kim nằm ngoài Home Screen tuy đơn giản nhưng rất đẹp mắt. Công ty này sau đó ngừng hợp tác với Google và bị RIM mua lại để tập trung phát triển sản phẩm Blackberry cũng như nền tảng BBX sau này.

## 1.2. Kiến trúc cơ bản của hệ điều hành Android

Android gồm năm phần chính sau được chứa trong bốn lớp:



Hình 1.2.1. Kiến trúc hệ điều hành Android

### 1.2.1. Nhân Linux

Android dựa trên Linux phiên bản 2.6 cho hệ thống dịch vụ cốt lõi như security, memory management, process management, network stack, and driver model. Kernel Linux hoạt động như một lớp trừu tượng hóa giữa phần cứng và phần còn lại của phần mềm stack.

### 1.2.2. Thư viện

Android bao gồm một tập hợp các thư viện C/C++ được sử dụng bởi nhiều thành phần khác nhau trong hệ thống Android. Điều này được thể hiện thông qua nền tảng ứng dụng Android. Một số các thư viện cơ bản được liệt kê dưới đây:

- Hệ thống thư viện C: một BSD có nguồn gốc từ hệ thống thư viện tiêu chuẩn C (libc), điều chỉnh để nhúng vào các thiết bị dựa trên Linux.
- Thư viện Media - dựa trên PacketVideo's OpenCORE; các thư viện hỗ trợ phát lại và ghi âm của âm thanh phổ biến và các định dạng video, cũng như các tập tin hình ảnh tĩnh, bao gồm cả MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.
- Bề mặt quản lý - Quản lý việc truy xuất vào hệ thống hiển thị.

- LibWebCore - một công cụ trình duyệt web hiện đại mà quyền hạn cả hai trình duyệt web Android và xem web nhúng.
- SGL - Đồ họa 2D cơ bản của máy.
- Thư viện 3D - một thực hiện dựa vào OpenGL ES 1.0 APIs; các thư viện sử dụng phần cứng tăng tốc 3D (nếu có), tối ưu hóa cao rasterizer phần mềm 3D.
- FreeType - vẽ phông chữ bitmap và vector.

**SQLite** một công cụ cơ sở dữ liệu quan hệ mạnh mẽ và nhẹ có sẵn cho tất cả các ứng dụng.

### 1.2.3. Thực thi

Android bao gồm một tập hợp các thư viện cơ bản mà cung cấp hầu hết các chức năng có sẵn trong các thư viện lõi của ngôn ngữ lập trình Java. Tất cả các ứng dụng Android đều chạy trong tiến trình riêng. Máy ảo Dalvik đã được viết để cho một thiết bị có thể chạy nhiều máy ảo hiệu quả. Các VM Dalvik thực thi các tập tin thực thi Dalvik (dex). Định dạng được tối ưu hóa cho bộ nhớ tối thiểu. VM là dựa trên register-based, và chạy các lớp đã được biên dịch bởi một trình biên dịch Java để chuyển đổi thành các định dạng dex. Các VM Dalvik dựa vào nhân Linux cho các chức năng cơ bản như luồng và quản lý bộ nhớ thấp.

### 1.2.4. Nền tảng Android

Bằng cách cung cấp một nền tảng phát triển mở, Android cung cấp cho các nhà phát triển khả năng xây dựng các ứng dụng cực kỳ phong phú và sáng tạo. Nhà phát triển được tự do tận dụng các thiết bị phần cứng, thông tin địa điểm truy cập, các dịch vụ chạy nền, thiết lập hệ thống báo động, thêm các thông báo để các thanh trạng thái, và nhiều, nhiều hơn nữa. Nhà phát triển có thể truy cập vào các API cùng một khuôn khổ được sử dụng bởi các ứng dụng lõi. Các kiến trúc ứng dụng được thiết kế để đơn giản hóa việc sử dụng lại các thành phần; bất kỳ ứng dụng có thể xuất bản khả năng của và ứng dụng nào khác sau đó có thể sử dụng những khả năng (có thể hạn chế bảo mật được thực thi bởi khuôn khổ). Cơ chế này cho phép các thành phần tương tự sẽ được thay thế bởi người sử dụng.

Cơ bản tất cả các ứng dụng là một bộ các dịch vụ và các hệ thống, bao gồm:

- Một tập hợp rất nhiều các View có khả năng kế thừa lẫn nhau dùng để thiết kế phần giao diện ứng dụng như: gridview, tableview, linearlayout
- Một “Content Provider” cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác (chẳng hạn như Contacts) hoặc là chia sẻ dữ liệu giữa các ứng dụng đó.
- Một “Resource Manager” cung cấp truy xuất tới các tài nguyên không phải là mã nguồn, chẳng hạn như: localized strings, graphics, and layout files.
- Một “Notification Manager” cho phép tất cả các ứng dụng hiển thị các custom alerts trong status bar. Activity Manager được dùng để quản lý chu trình sống của ứng dụng và điều hướng các activity.

### **1.2.5. Tầng ứng dụng**

Tầng ứng dụng (Application) là tầng giao tiếp với người dùng với các thiết bị Android như Danh bạ, tin nhắn, trò chơi, tiện ích tính toán, trình duyệt... Mọi ứng dụng viết đều nằm trên tầng này.

## CHƯƠNG 2: MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG ANDROID STUDIO, SQLITE

### 2.1. Giới thiệu ứng dụng Android Studio

Android Studio là một nền tảng IDE<sup>1</sup> (integrated development environment) dùng để phát triển các ứng dụng android, được Google release vào khoảng đầu năm 2015 thay thế cho bản Eclipse cũ. Android Studio được phát triển dựa trên IntelliJ IDEA Community Edition - công cụ lập trình tốt nhất cho java, giúp cho các lập trình viên tạo ứng dụng, thực hiện các thay đổi một cách dễ dàng, bên cạnh đó có thể xem trước trong thời gian thực và thiết kế giao diện đẹp hơn trước. Tiếng Việt cũng đã được tích hợp trong Android Studio. Đặc biệt, Android Studio cho phép người dùng Import Project từ Eclipse sang và logic lập trình cũng tương tự.

### 2.2. Cài đặt môi trường lập trình Android

#### 2.2.1. Cài đặt JAVA JDK

- Bước 1: Tải file cài đặt tại đường link

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

**Lưu ý:** Chọn phiên bản tương ứng với hệ điều hành đúng với máy đang sử dụng.

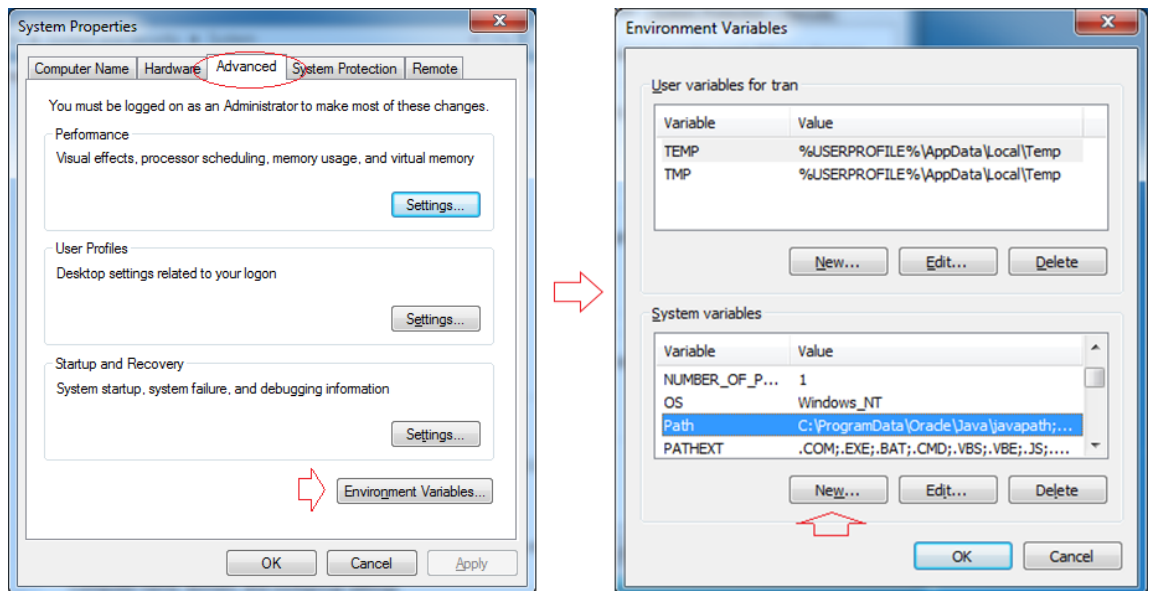
- Bước 2: Mở file cài đặt “jdk-\*.exe” để tiến hành cài đặt
- Bước 3: Cấu hình biến môi trường cho Java

Việc này không bắt buộc, nhưng nếu trên máy tính của cài đặt nhiều phiên bản Java, thì việc cấu hình là cần thiết để xác định phiên bản java nào mặc định được sử dụng.

- Trên Desktop, nhấn phải chuột vào Computer, chọn **Properties > Advanced system settings**.

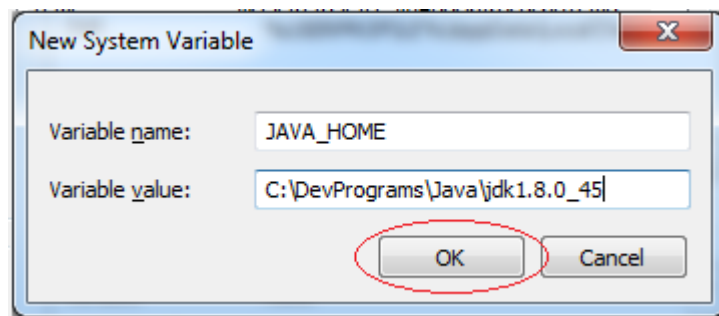
---

<sup>1</sup> Môi trường lập trình phát triển tích hợp



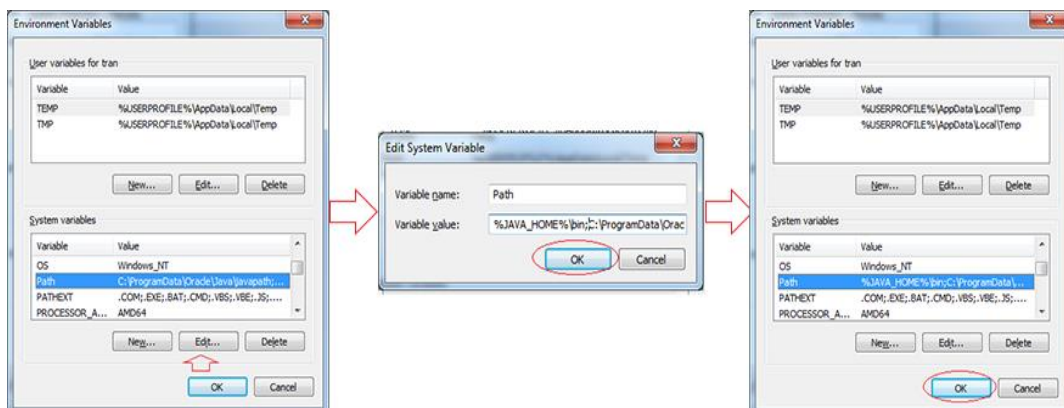
Hình 2.2.1.1. Thêm đường dẫn mới tới thư mục JDK

- Nhập vào đường dẫn tới thư mục JDK.



Hình 2.2.1.2. Nhập vào đường dẫn tới thư mục JDK

- Tiếp theo đổi tên path là đã cài đặt xong Java JDK



Hình 2.2.1.3. Đổi tên path

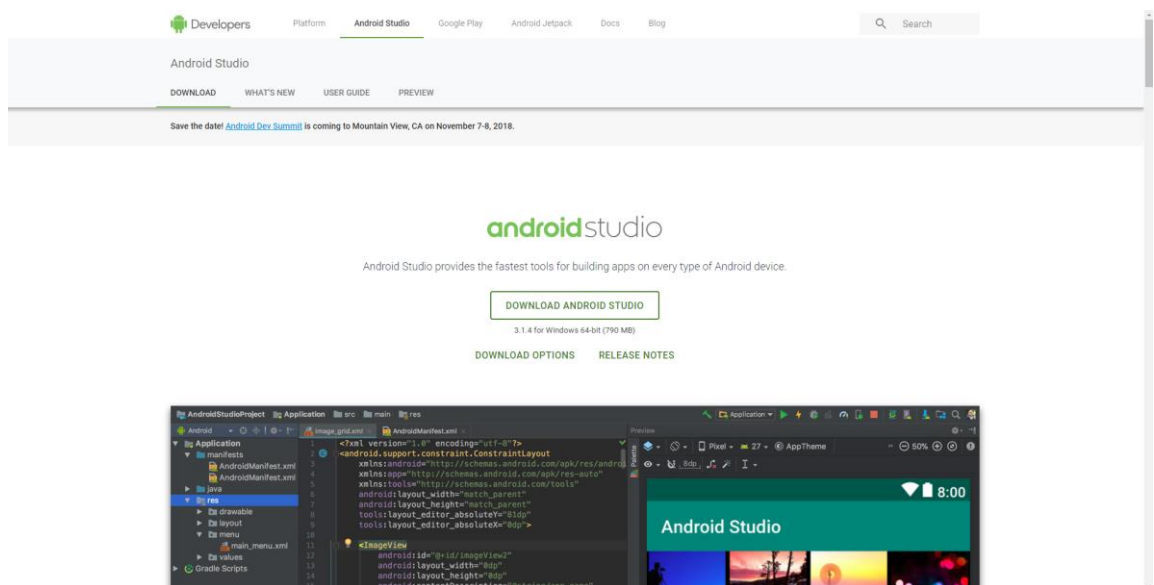
## 2.2.2 Cài đặt Android Studio

a. Cấu hình yêu cầu:

- Microsoft® Windows® 10/8/7/Vista (32 or 64-bit)
- Tối thiểu 4 GB RAM, cấu hình đề nghị: 8 GB RAM
- Ổ cứng trống ít nhất : 20Gb
- Độ phân giải tối thiểu 1280 x 800
- Java Development Kit (JDK) 7 trở lên
- Lựa chọn thêm cho accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

b. Tải file cài đặt

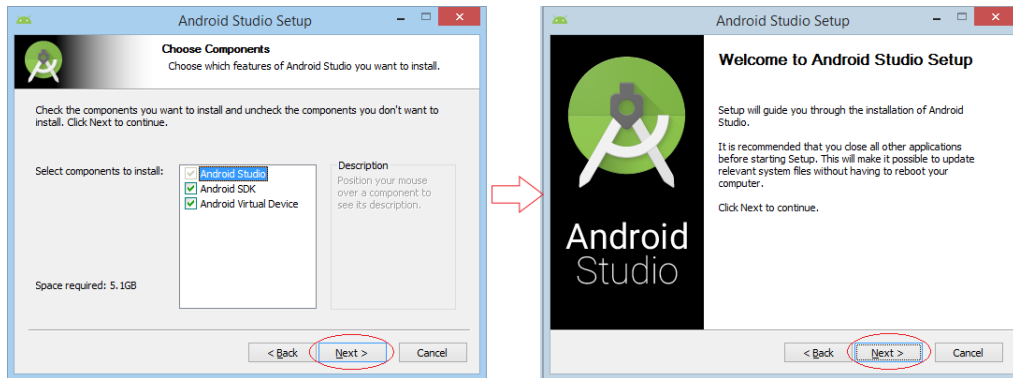
**Download file cài đặt: <https://developer.android.com/studio>**



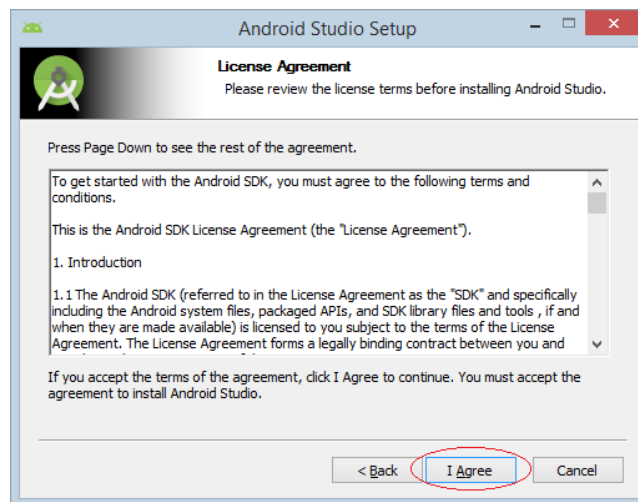
Hình 2.2.2.1. Giao diện trang web để tải file cài đặt

### c. Cách cài đặt

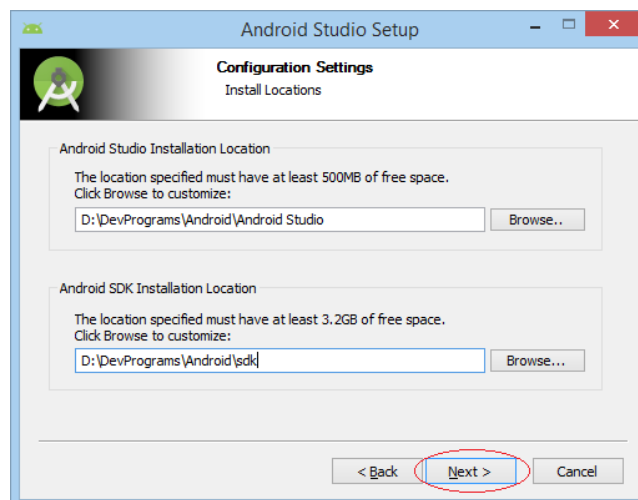
Bước 1: Mở file cài đặt và ấn Next



Bước 2: Chọn “I agree” để xác nhận, đồng ý với các điều khoản và tiếp tục.

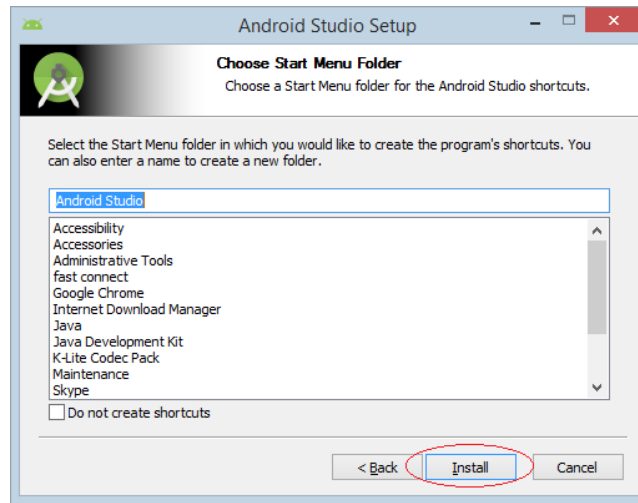


Bước 3: Chọn nơi cài đặt Android Studio và Android SDK

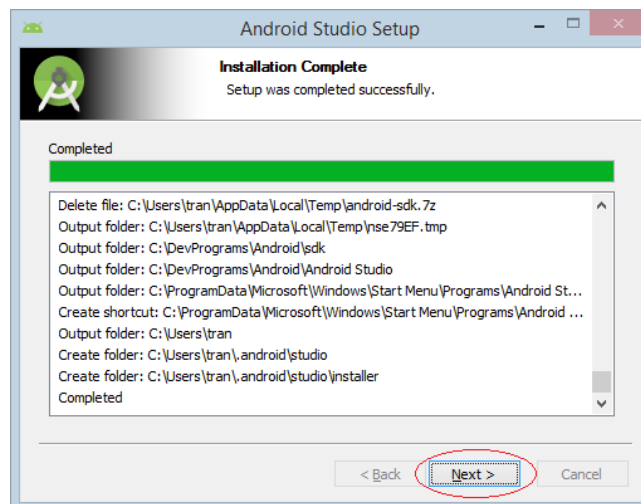




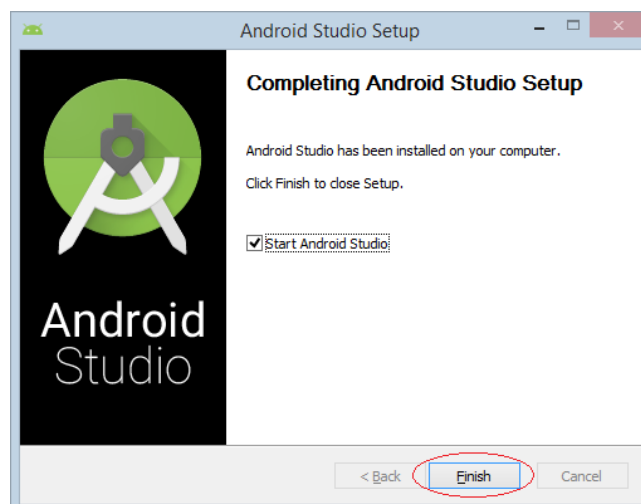
Bước 4: Chọn Start Menu Folder. Ở bước này, có thể tích tạo short cut cho ứng dụng hoặc không. Bấm Install để bắt đầu cài đặt.



Bước 5: Chờ hệ thống cài đặt các package cần thiết cho tới khi hoàn thành và bấm Next để tiếp tục.



Bước 6: Bấm Finish và click Start Android Studio để khởi động phần mềm.



## 2.2.3 Máy ảo Android Genymotion

### a. Yêu cầu:

- Card đồ họa OpenGL 2.0.
- CPU hỗ trợ VT-x hoặc AMD-V và kích hoạt thiết lập BIOS.
- RAM: Tối thiểu 1 GB.
- Dung lượng trống của ổ cứng: Ít nhất là 2GB để cài đặt Genymotion và các máy ảo chạy Genymotion (đây chỉ là mức yêu cầu tối thiểu, bởi nếu sử dụng nhiều máy ảo cùng lúc và có nhiều ứng dụng, phần mềm cài đặt thì dung lượng trống có thể được yêu cầu nhiều hơn gấp 4 lần).
- Đảm bảo có kết nối Internet
- Độ phân giải màn hình: ít nhất 1024 x 768 pixel
- Oracle **VirtualBox 4.1** trở lên.
- Ngoài ra, người dùng cần có một tài khoản Genymotion để có thể sử dụng.

### b. Tải file cài đặt:

**Download:** <https://www.genymotion.com/fun-zone/>

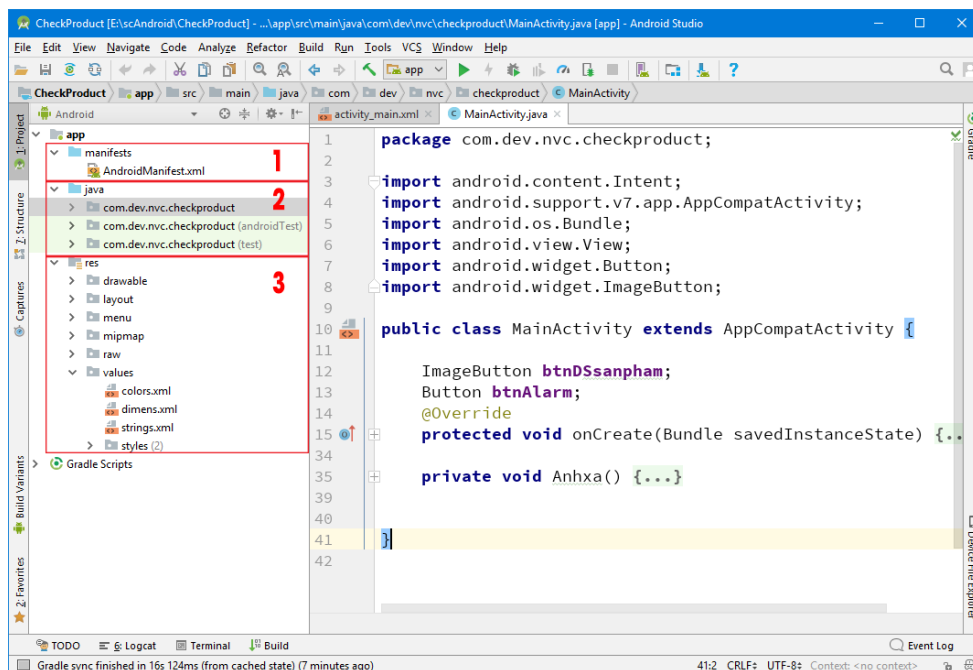
### c. Cách cài đặt:

- Bước 1: Tìm tới vị trí lưu file vừa tải về đầu tiên, sau đó click đúp chuột trái vào file đó để bắt đầu tiến hành. Giao diện đầu tiên sẽ cho lựa chọn ngôn ngữ để sử dụng.
- Bước 2: Tùy theo ý muốn người dùng mà có thể **Next** ngay và cài đặt Genymotion ngay tại vị trí được gợi ý hoặc chọn **Browse** để tìm vị trí khác.
- Bước 3: Tiếp tục **Next**.
- Bước 4: Chọn mục **Create a desktop shortcut** để tạo biểu tượng cho phần mềm này trên máy tính, sau đó **Next** tiếp. Chọn vào **Install** để thực hiện cài đặt.
- Bước 5: Rất có thể Genymotion sẽ tự động cài thêm một số phần mềm hỗ trợ, đồng thời hỏi có muốn cài đặt VirtualBox hay không? Nếu máy tính của đã có sẵn phần mềm VirtualBox mới nhất rồi thì thôi, còn nếu không, chọn **Yes** để cài đặt tiếp.

- Bước 6: Tới đây, ngoài cửa sổ cài đặt Genymotion (sẽ tạm thời bị dừng lại), sẽ thấy có thêm một giao diện khác để tiến hành cài đặt VirtualBox.
- Bước 7: Sẽ không có thêm bất cứ phần mềm hay thao tác thiết lập nào khác, nên các có thể yên tâm **Next** tới khi kết thúc > **Next** tiếp nào > Click chọn **Yes**.
- Bước 8: **Install**.
- Bước 9: Bỏ dấu tick ở **Start Oracle WM VirtualBox** và chọn vào **Finish**.
- Bước 10: Lúc này cửa sổ cài đặt cuối cùng của Genymotion xuất hiện, **Finish** để kết thúc quá trình.

## 2.3. Thành phần trong một dự án ANDROID

Bao bọc một Project android là thư mục app và phía dưới là ba thành phần chính của một project trong Android Studio:



Hình 2.3.1. Cấu trúc một project trong Android Studio

### 2.3.1. Tập cấu hình Android<sup>2</sup>

#### a. Giới thiệu

Trong bất kì một project Android nào khi tạo ra đều có một file AndroidManifest.xml, file này được dùng để định nghĩa các screen sử dụng, các permission cũng như các

<sup>2</sup> AndroidManifest

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. Trong file manifest bao giờ cũng có 3 thành phần chính đó là: application, permission và version. Hay nói cách khác đây là file dùng để config những thuộc tính cho ứng dụng của mà khi ứng dụng khởi chạy hệ điều hành có thể hiểu được và xử lí.

#### *b. Tác dụng của AndroidManifest*

- Nó đặt tên gói Java cho ứng dụng. Tên gói đóng vai trò như một mã nhận diện duy nhất cho ứng dụng.
- Nó mô tả các thành phần của ứng dụng - hoạt động, dịch vụ, hàm nhận quảng bá, và trình cung cấp nội dung mà ứng dụng được soạn bởi. Nó đặt tên các lớp triển khai từng thành phần và công bố các khả năng của chúng (ví dụ, những tin nhắn Intent mà chúng có thể xử lý). Những khai báo này cho phép hệ thống Android biết các thành phần là gì và chúng có thể được khởi chạy trong những điều kiện nào.
- Nó xác định những tiến trình nào sẽ lưu trữ các thành phần ứng dụng.
- Nó khai báo các quyền mà ứng dụng phải có để truy cập các phần được bảo vệ của API và tương tác với các ứng dụng khác.
- Nó cũng khai báo các quyền mà ứng dụng khác phải có để tương tác với các thành phần của ứng dụng.
- Nó liệt kê các lớp Instrumentation cung cấp tính năng tạo hồ sơ và các thông tin khác khi ứng dụng đang chạy. Những khai báo này chỉ xuất hiện trong bản kê khai khi ứng dụng đang được phát triển và thử nghiệm; chúng bị loại bỏ trước khi ứng dụng được công bố.
- Nó khai báo mức tối thiểu của API Android mà ứng dụng yêu cầu.
- Nó liệt kê các thư viện mà ứng dụng phải được liên kết với.

## Cấu trúc tệp AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest>
  <uses-permission /> <permission /> <permission-tree
/> <permission-group />
  <instrumentation />
  <uses-sdk /> <uses-configuration /> <uses-feature />
  <supports-screens />
  <compatible-screens />
  <supports-gl-texture />
  <application>
    <activity>
      <intent-filter> <action /> <category /> <data /> </intent-
filter>
      <meta-data />
    </activity>
    <activity-alias>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </activity-alias>
    <service>
      <intent-filter> . . . </intent-filter>
      <meta-data/>
    </service>
    <receiver>
      <intent-filter> . . . </intent-filter>
      <meta-data />
    </receiver>
    <provider>
      <grant-uri-permission />
      <meta-data />
      <path-permission />
    </provider>
    <uses-library />
  </application>
</manifest>
```

### c. Một số thẻ trong *AndroidManifest.xml*

`<uses-permission>...</uses-permission/>`: đây là cú pháp lệnh để xin cấp một quyền gì đó khi ứng dụng tương tác với dữ liệu nào đó mà google không cho phép dùng tùy ý.

`<activity>...</activity>`: đây là thẻ để khai báo các activity trong ứng dụng và các thuộc tính của activity đó, nếu như không khai báo thì khi khởi chạy ứng dụng sẽ lỗi ngay.

`<service>...</service>`: đây là một dịch vụ chạy ngầm trên ứng dụng của kể cả khi ứng dụng đã tắt đi và khi sử dụng phải khai báo chúng trong *AndroidManifest*.

`<receiver>...</receiver>`: đây là khai báo khi sử dụng Broadcast Receiver<sup>3</sup>, cái này là để lắng nghe các sự kiện thay đổi của hệ thống như khi bắt wifi, khi sạc pin...

`<uses-sdk>...</uses-sdk>`: Thẻ xác định phiên bản SDK

### 2.3.2. Thư mục Java

Đây chính là nơi chứa các gói<sup>4</sup> của dự án, có thể tạo các gói ở đây và bên trong là các class.

### 2.3.3. Thư mục Res

**Gói Drawable:** Đây chính là thư mục chứa các file hình ảnh, config xml... trong dự án android. Ví dụ như muốn ứng dụng sử dụng một hình ảnh nào đó là background thì ảnh đó sẽ bỏ vào thư mục này. Hoặc muốn điều chỉnh một nút button khi click vào màu xanh còn khi không click vào màu trắng thì sẽ config trong file xml và lưu vào trong này.

**Gói Layout:** Đây chính là thư mục lưu các file xml về giao diện của các màn hình ứng dụng của. Ở trên phần số một có các package lưu các class, các class này sẽ

---

<sup>3</sup> Một trong các lớp của Android

<sup>4</sup> Package

kết nối với các file xml trong thư mục layout nào để tạo nên một màn hình có giao diện cho người dùng thao tác.

**Gói Mipmap:** Đây là thư mục mà sẽ chứa ảnh logo ứng dụng chúng ta, lúc này có nói là các file hình ảnh sẽ được chứa trong thư mục drawable nhưng ngoại lệ ảnh logo thì chứa trong thư mục *mipmap* này cho chuẩn.

**Gói Values:** Sẽ có rất nhiều file ở bên trong như sau:

- **color.xml:** đây là file định nghĩa các mã màu trong dự án android, khi sử dụng màu nào chỉ cần gọi tên mã màu đã định nghĩa trong đây ra là xong.
- **dimens.xml:** đây là file mà sẽ định nghĩa ra các kích thước như cỡ chữ, chiều cao, chiều rộng các view.
- **strings.xml:** đây là file định nghĩa các đoạn văn bản trong ứng dụng Android của ví dụ như có một đoạn văn bản mà sử dụng đi sử dụng lại trong các màn hình khác nhau, khi set cứng ở nhiều nơi thì khi cần chỉnh sửa thì phải tìm hết tất cả và sửa lại. Bây giờ định nghĩa đoạn văn bản đó trong đây và khi dùng thì gọi ra sử dụng và sau này chỉnh sửa chỉ cần sửa trong đây là xong, nó sẽ apply tất cả mọi nơi.
- **styles.xml:** đây chính là nơi định nghĩa các giao diện của các file layout trong thư mục layout đã nói phía trên. Kiểu như thế này nhé, muốn chỉnh một nút Button chiều cao 10dp, chiều rộng 10dp, màu xanh... và lại sử dụng kiểu thiết kế này ở năm màn hình khác nhau. Không thể mỗi màn hình lại định nghĩa lại như thế sẽ làm duplicate code (lặp lại) và sẽ không tối ưu tí nào cả. Thay vào đó chỉ cần định nghĩa một file giao diện như trên và ở mỗi màn hình chỉ cần gọi là xong.

#### 2.3.4. Tệp Grade Scripts

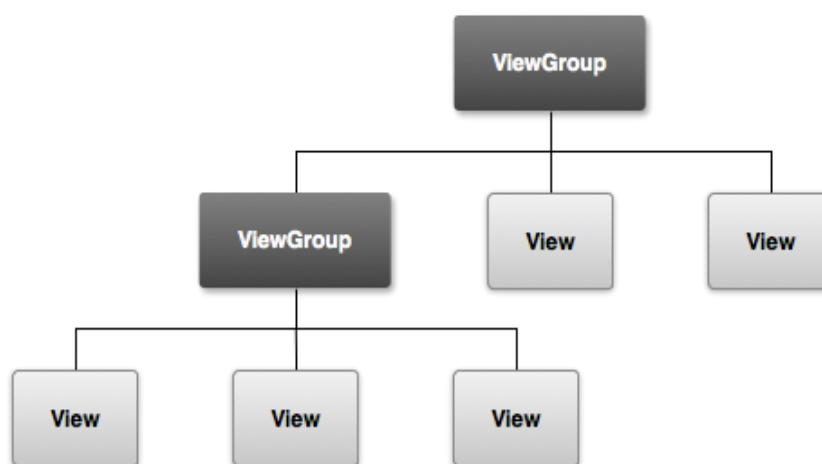
**Build.grade** là file để thiết lập các thuộc tính cho dự án android như: phiên bản SDK, Version ứng dụng, package, thêm thư viện ngoài...

## 2.4. Thành phần giao diện

### 2.4.1. View group

#### a. Khái niệm

Một ViewGroup là một đối tượng được 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 (xem bên dưới) do đó có thể tạo các layout phức tạp hơn.



Hình 2.4.1.1. Sơ đồ phân cấp các thành phần giao diện

#### b. Một số view group cơ bản

**LinearLayout:** Tồn tại để hiển thị các phần tử theo một thứ tự xếp chồng lên nhau theo chiều ngang hoặc chiều dọc. LinearLayout cũng có thể được sử dụng để gán weight cho các phần tử View con để các phần tử được cách khoảng trên màn hình theo tỉ lệ tương ứng với nhau.

**RelativeLayout:** Lớp con này của ViewGroup cho phép hiển thị các phần tử trên màn hình tương đối với nhau, cung cấp nhiều tính linh hoạt hơn và tự do trong cách layout của xuất hiện so với LinearLayout.

**FrameLayout:** Được thiết kế để hiển thị một View con tại một thời điểm, FrameLayout vẽ các phần tử trong một ngăn xếp và cung cấp một cách đơn giản để hiển thị một phần tử trên các kích cỡ màn hình khác nhau.



**ScrollView:** Một lớp mở rộng của `FrameLayout`, lớp `ScrollView` xử lý việc cuộn các đối tượng con của nó trên màn hình.

**RecyclerView:** Lớp `RecyclerView` là một lớp con của `ViewGroup`, nó liên quan đến các lớp `ListView` và `GridView` và nó được cung cấp bởi Google thông qua thư viện hỗ trợ `RecyclerView` cho các phiên bản Android cũ hơn. Lớp `RecyclerView` đòi hỏi việc sử dụng các mẫu thiết kế view holder để tái sử dụng phần tử một cách có hiệu quả và nó hỗ trợ việc sử dụng một `LayoutManager`, một thành phần trang trí, và một phần tử động để làm cho thành phần này vô cùng linh hoạt và đơn giản.

**CoordinatorLayout:** Được thêm gần đây vào thư viện hỗ trợ thiết kế, lớp `CoordinatorLayout` sử dụng một đối tượng `Behavior` để xác định cách các phần tử `View` con sẽ được sắp xếp và di chuyển khi người dùng tương tác với ứng dụng của .

## 2.4.2. View

### a. Khái niệm

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`. 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, background, kích thước, lề,... Tất cả những thuộc tính chung này được thể hiện hết ở trong đối tượng `View`. Trong Android Platform, các screen luôn được bố trí theo một kiểu cấu trúc. Một screen là một tập hợp các `Layout` và các widget được bố trí có thứ tự.

### b. Một số view cơ bản trong android

Tên widget	Chức năng
<b>TextView</b>	Cho phép người dùng hiển thị một đoạn văn bản lên màn hình mà không cho phép người dùng sửa nó.
<b>EditText</b>	Cho phép người dùng nhập, xóa, sửa một đoạn văn bản vào trong đó.  Có nhiều dạng <code>EditText</code> khác nhau như: <code>Plaintext</code> , <code>Person</code>

<b>Tên widget</b>	<b>Chức năng</b>
	Name, Password, Email, Phone,...
<b>Button</b>	Dùng để thiết lập các sự kiện khi người dùng thao tác với nó.
<b>ImageButton</b>	Là dạng nút bấm nhưng có thể chèn thêm hình ảnh vào để giao diện thêm sinh động, trực quan hơn.
<b>CheckBox</b>	Một dạng nút bấm đặc biệt chỉ có hai trạng thái là check và uncheck.
<b>ToggleButton</b>	Có thể xem nó như một Checkbox và có kèm theo hiệu ứng ánh sáng bật/tắt thể hiện trạng thái Check/Uncheck.
<b>RadioButton</b>	Chọn một trong các Radio
<b>Spinner</b>	Là view thể hiện quá trình xử lý diễn ra bên dưới ứng dụng, tạo cảm giác thực cho người dùng. Có hai dạng progressBar là Large và Horizontal
<b>ListView</b>	Chỉ cho phép chọn một trong nhóm lựa chọn.
<b>GridView</b>	Là view được hiển thị dưới dạng lưới gồm nhiều item con bên trong và ta có thể tùy chỉnh nội dung cũng như các đối tượng nằm bên trong item một cách tùy ý.
<b>ViewFlipper</b>	Cho phép định nghĩa một tập hợp nhiều View nhưng chỉ có một View hiển thị tại một thời điểm và hỗ trợ hiệu ứng chuyển đổi giữa các View
<b>QuickContactBage</b>	Hiển thị hình ảnh gắn liền với một đối tượng bao gồm số điện thoại, tên, email đồng thời hỗ trợ việc gọi, nhắn tin sms, email hay tin nhắn tức thời (IM - instant message).
<b>ImageView</b>	Hiển thị hình ảnh
<b>Switch</b>	Tồn tại hai trạng thái đóng mở
<b>ProgressBar</b>	Thể hiện tiến trình, mức độ

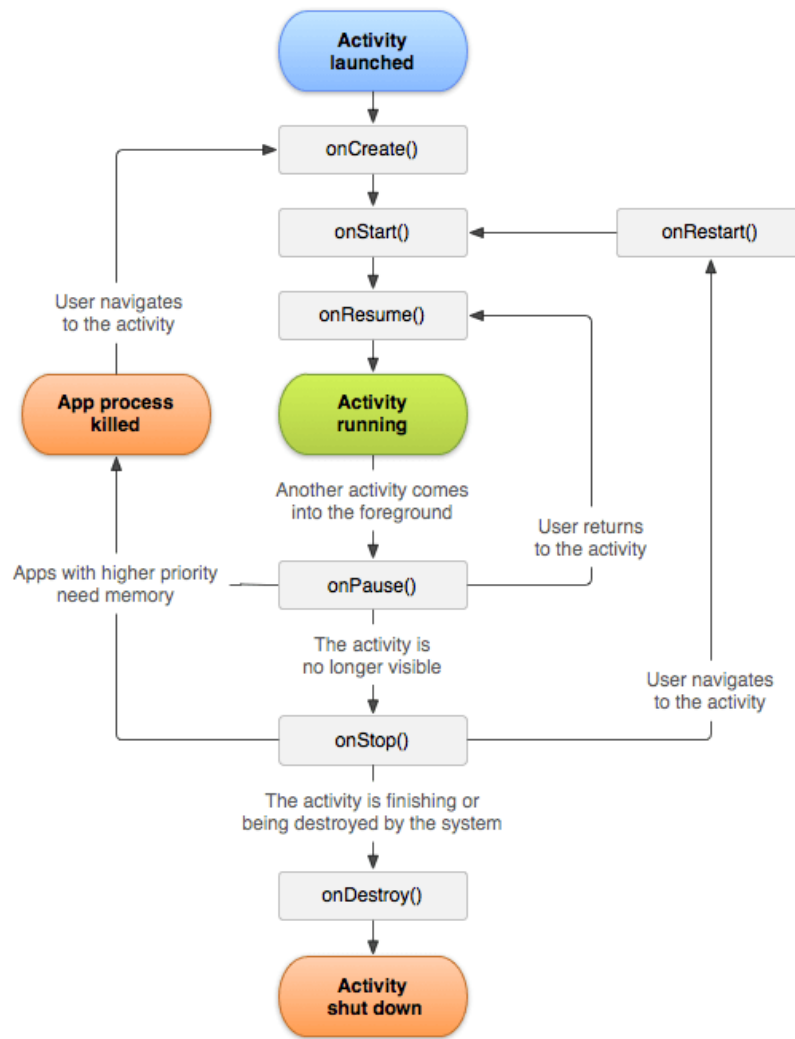
c. Thuộc tính cơ bản widget

Tên thuộc tính	Công dụng
<b>layout_width</b>	Chiều rộng của control (Tính theo trục Ox từ góc trên bên trái qua phải). Có ba thuộc tính: wrap_content (Co dẫn theo nội dung control), match_parent (kích thước chiều ngang bằng kích thước đối tượng chứa nó)
<b>layout_height</b>	Chiều cao của control (Tính theo trục Oy từ góc trên bên trái xuống dưới). Có ba thuộc tính như <i>layout:width</i>
<b>text</b>	Nội dung hiển thị
<b>textcolor</b>	Màu chữ
<b>background</b>	Màu nền của view
<b>id</b>	Định danh của view
<b>gravity</b>	Căn lề cho nội dung của view
<b>textStyle</b>	Thiết lập kiểu dáng chữ: đậm, in nghiêng, gạch chân
<b>textSize</b>	Thiết lập cỡ chữ
<b>fontFamily</b>	Thiết lập họ phong chữ
<b>inputType</b>	Thiết lập kiểu nhập dữ liệu
<b>hint</b>	Đoạn văn bản gợi ý cho người dùng biết về chức năng hay ràng buộc gì đó...

## 2.5. Vòng đời ứng dụng android

- Các Activity trong hệ thống được quản lý như một ngăn xếp activity (activity stack). Khi một activity mới bắt đầu nó được đặt lên đầu của ngăn xếp và trở thành Running Activity (activity đang chạy), đồng thời activity trước đó sẽ nằm ngay phía dưới trong ngăn xếp đó, và sẽ không trở nên visible (nhìn thấy) cho đến khi activity ở trên thoát ra khỏi ngăn xếp.

- Một Activity gồm bốn trạng thái chính:
  - Nếu activity ở phía trên của màn hình (hay ở trên cùng của ngăn xếp), thì nó đang ở trạng thái active (hoạt động) / running (đang chạy). Ví dụ khi ta cần gọi điện thì activity bấm số đó đang ở trạng thái active.
  - Nếu activity không thể tương tác nhưng vẫn nhìn thấy (khi mà bị che bởi một activity khác nhưng người dùng vẫn có thể nhìn thấy nó ở phía sau) thì activity này đang ở trạng thái paused (tạm dừng). Khi ở trạng thái này activity có thể bị xóa bỏ bởi hệ thống khi thiết bị thiếu bộ nhớ. Ví dụ khi có một activity khác dạng dialog hiện lên chỉ che đi một phần của activity hiện tại thì activity vào trạng thái paused.
  - Nếu activity hoàn toàn bị che khuất bởi activity khác thì nó đang ở trạng thái stopped (đã dừng). Activity này vẫn giữ được tất cả trạng thái và thông tin, nhưng không còn hiển thị với người dùng và thường xuyên bị xóa bỏ bởi hệ thống khi thiếu bộ nhớ. Ví dụ khi ta tắt màn hình thì khi đó activity vào trạng thái stopped.
  - Nếu activity ở trạng thái paused (tạm dừng) hay stopped (đã dừng), hệ thống có thể xóa bỏ activity đó khỏi bộ nhớ bằng cách yêu cầu nó tự kết thúc hoặc xóa bỏ tiến trình của nó. Khi activity đó hiển thị lại với người dùng thì sẽ được khởi tạo lại và khôi phục lại trạng thái trước đó.



Hình 2.5.1. Vòng đời của một Activity

## 2.6. Lớp Intent

### 2.6.1. Khái niệm

**Intent** là một thành phần quan trọng trong android. **Intent** cho phép các thành phần ứng dụng có thể yêu cầu các hàm từ các thành phần ứng dụng android khác.

**Intent** là đối tượng của lớp **android.content.Intent**. Mã của nó có thể gửi **Intent** vào hệ thống **Android** với chỉ định thành phần mục tiêu gửi đến.

Một đối tượng **Intent** có thể chứa dữ liệu thông qua một đối tượng của lớp **Bundle**. Dữ liệu này có thể được sử dụng bởi các thành phần tiếp nhận.

### 2.6.2. Các loại Intent

Android hỗ trợ hai loại Intent là Intent tường minh (explicit) và Intent không tường minh (implicit).

Một ứng dụng có thể xác định thành phần mục tiêu một cách trực tiếp vào Intent (mục tiêu được yêu cầu là rõ ràng) hoặc yêu cầu hệ thống Android đánh giá các thành phần đã đăng ký trên dữ liệu đích để chọn ra một cái để gửi yêu cầu đến ( Intent không tường minh).

**Intent tường minh (Explicit intents):** Là những ý định (intent) chỉ định rõ ràng tên của các thành phần mục tiêu để xử lý; trong đó, trường mục tiêu (tùy chọn) được sét một giá trị cụ thể thông qua các phương thức *setComponent()* hoặc *setClass()*.

```
//Màn hình 1:
//cách 1:
//Sử dụng Bundle
Intent intent1 = new Intent(this, ManHinh2Activity.class);
Bundle bundle = intent1.getExtras();
bundle.putString("khoa1", "Giá trị 1");
bundle.putString("khoa1", "Giá trị 2");
//cách 2:
//Sử dụng Bundle
Intent intent2 = new Intent(this, ManHinh2Activity.class);
Bundle bundle2 = new Bundle();
bundle2.putString("khoa1", "Giá trị 1");
bundle2.putString("khoa2", "Giá trị 2");
intent2.putExtra(bundle2);
//-----
//màn hình 2:
Intent intent = this.getIntent();
String str_giatri1 = intent.getStringExtra("khoa1");
String str_giatri2 = intent.getStringExtra("khoa2");
```

### Ví dụ tạo Intents tường minh:

**Intent không tường minh (Implicit Intents):** Là những ý định (intent) không chỉ định rõ một mục tiêu thành phần, nhưng bao gồm đầy đủ thông tin cho hệ thống để xác định các thành phần có sẵn là tốt nhất để chạy cho mục đích đó. Hãy xem xét một ứng dụng liệt kê các nhà hàng có sẵn ở gần . Khi bấm vào một tùy chọn nhà hàng cụ thể, ứng dụng sẽ hỏi một ứng dụng khác để hiển thị các tuyến đường đến nhà hàng đó. Để đạt được điều này, nó có thể gửi một ý định rõ ràng trực tiếp đến các ứng dụng

**Google Maps**, hoặc gửi ý định ngầm, ý định sẽ được chuyển giao cho bất kỳ ứng dụng nào cung cấp các tính năng bản đồ (map) (chẳng hạn, **Yahoo Maps**).

### Ví dụ Intents không tường minh:

```
//Một Intent không tường minh, yêu cầu xem một URL
//Một đường dẫn URL vốn là một thứ được xem trên trình duyệt
Intent intent = new Intent(Intent.ACTION_VIEW,
Uri.parse("//google.com"));
startActivity(intent);
```

#### c. Truyền dữ liệu giữa các activities

Truyền dữ liệu đến thành phần đích

Để truyền dữ liệu cho intent, ta dùng phương thức **putExtras()**. Extras là **một cặp key/value**. key luôn luôn là kiểu string. value có thể sử dụng kiểu dữ liệu nguyên thủy hoặc đối tượng của String, Bundle, ....

Thành phần tiếp nhận có thể lấy lại được đối tượng intent thông qua hàm **getIntent()**. Để lấy ra được dữ liệu, tùy thuộc vào kiểu dữ liệu truyền đi, sử dụng các phương thức **getStringExtra()**, **getIntExtra()**;

Ví dụ:

```
Intent intent = new Intent(MainActivity.this,
Main2Activity.class);
Intent.putExtra("name", "My name");
Intent intent = getIntent();
String hoTen = intent.getStringExtra("name");
```

Ta có thể sử dụng đối tượng **Bundle**. Đóng gói tất cả dữ liệu vào trong Bundle xong truyền Bundle cho Intent, ví dụ:

## – Sử dụng Intent chia sẻ

```
Intent mIntent = new Intent(MainActivity.this,
Main2Activity.class);
Bundle mBundle = new Bundle();
mBundle.putString(key, value);
mIntent.putExtras(mBundle);
-----
String value = getIntent().getExtras().getString(key);
```

Có rất nhiều ứng dụng android cho phép chia sẻ dữ liệu với những người khác, ví dụ: facebook, G+, Gmail... có thể gửi dữ liệu tới một vài thành phần nào đó. Ví dụ

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.setType("text/plain");
intent.putExtra(android.content.Intent.EXTRA_TEXT, "News for
you!");
startActivity(intent);
```

dưới đây sẽ mô tả việc sử dụng intent như vậy.

Lấy lại kết quả sử dụng từ sub-activity

Một activity có thể được đóng lại thông qua button back trên điện thoại. Trong trường hợp đó, hàm **finish()** sẽ được thực thi. Nếu activity đã được khởi chạy cùng với hàm **startActivity(intent)**, người gọi không cần thiết phải có kết quả hoặc phản hồi từ activity mà có thể close ngay lập tức.

Nếu start một activity cùng với hàm **startActivityForResult()**, như vậy là mong muốn có phản hồi từ **sub-activity**. Khi một sub-activity kết thúc, hàm **onActivityResult()** trên sub-activity sẽ được gọi và có thể thực hiện hành động dựa trên kết quả.

Trong hàm **startActivityForResult()**, có thể xác định hành động sẽ khởi chạy. Hành động này sẽ mong muốn nhận về kết quả. Một hành động đã được khởi chạy



cũng có thể thiết lập một đoạn mã code mà người gọi có thể sử dụng để xác định hành động sẽ được cancel hoặc not.

Sub-activity sử dụng hàm **finish()** để khởi tạo một intent mới và truyền dữ liệu cho nó. Nó cũng thiết lập kết quả thông qua hàm **setResult**

```
public void onClick(View view) {
    Intent i = new Intent(this,ActivityTow.class);
    i.putExtra("value1","This value one for activityTow");
    i.putExtra("Value2", "This value two ActivityTwo");
    // set the request code to any code you like,
    // you can identify the callback via this code
    startActivityForResult(i, REQUEST_CODE);
}
```

Nếu sử dụng hàm **startActivityForResult()**, thì sau khi start activity sẽ gọi đến sub-activity

Nếu hàm sub-activity kết thúc, nó sẽ gửi dữ liệu đến cái đã gọi nó thông qua intent. điều này được xử lý trên hàm **finish()**

```
@Override
public void finish() {
    // Prepare data intent
    Intent data = new Intent();
    data.putExtra("returnKey1", "Swinging on a star. ");
    data.putExtra("returnKey2", "You could be better then you
are. ");
    // Activity finished ok, return the data
    setResult(RESULT_OK, data);
    super.finish();
}
```

Một sub-activity kết thúc, hàm onActivityResult() trong activity cha sẽ được gọi

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    if (resultCode == RESULT_OK && requestCode == REQUEST_CODE)
    {
        if (data.hasExtra("returnKey1")) {
            Toast.makeText(this,
            data.getExtras().getString("returnKey1"),
            Toast.LENGTH_SHORT).show();
        }
    }
}
```

## 2.7. Share preferences

### 2.7.1. Khái niệm

Đây là một class **Interface** cho phép lưu trữ và đọc dữ liệu với bằng các cặp key và value và nó được lưu dưới dạng một file xml, dữ liệu nó có thể lưu là ở dạng nguyên thủy như: int, float, string, boolean, long. Dữ liệu của Shared Preferences sẽ được lưu ở trong ứng dụng android luôn chính vì thế nếu các xoá ứng dụng đi hoặc là xoá dữ liệu app thì dữ liệu này sẽ hoàn toàn bị biến mất.

### 2.7.2. Cách sử dụng

#### a. Lưu dữ liệu xuống Shared Preferences

Đầu tiên, để lưu dữ liệu xuống thì sẽ khởi tạo một đối tượng (biến) thuộc

```
SharedPreferences sharedPreferences = getSharedPreferences
(SHARED_PREFERENCES_NAME, Context.MODE_PRIVATE );
```

kiểu **SharedPreferences**, với cú pháp như sau:

**SHARED\_PREFERENCES\_NAME**: là tên của file **Shared Preferences**.

**Context.MODE\_PRIVATE**: đây là một chế độ bảo mật dữ liệu trong android, khi để như vậy có nghĩa là chỉ cho ứng dụng hiện tại truy cập vào file Shared Preferences này thôi mà không một ứng dụng nào có quyền truy cập vào được.

Tạo đối tượng Editor từ biến sharedPreferences đã tạo ở trên, mục đích của thặng này là để có thể mở file **shared\_preferences\_name.xml** ra và đưa dữ liệu vào, cú

```
SharedPreferences.Editor editor = sharedPreferences.edit();
```

```
editor.putX(String key,value);
```

pháp như sau:

Đưa dữ liệu vào:

Sau khi đã đặt dữ liệu xong thì sẽ gọi hàm **commit()** hoặc **apply()** để xác nhận những thay đổi.

*b. Cách lấy dữ liệu từ Shared Preferences*

```
SharedPreferences sharedPreferences =
getSharedPreferences (SHARED_PREFERENCES_NAME,
Context.MODE_PRIVATE);

boolean isFirtsLauncher =
sharedPreferences.getBoolean ("IS_FIRTS_LAUNCHER",true);

int highScore = sharedPreferences.getInt ("HIGH_SCORE",0);

float mark = sharedPreferences.getFloat ("MARK",0.0f);

long downloadProgress =
sharedPreferences.getLong ("DOWNLOAD_PROGRESS",0);

String name = sharedPreferences.getString ("NAME","");
```

Đây là cách lấy ra tất cả các dữ liệu mà ở trên đã lưu:

### c. Cách xóa dữ liệu đã lưu xuống Shared Preferences

```
editor.remove(String key);
```

Xoá đi dữ liệu nào đó đã lưu trong Shared Preferences thì chỉ cần gọi lệnh:

Xoá đi hết tất cả giá trị đã lưu thì bạn dùng cú pháp:

```
editor.clear();
```

Gọi `editor.apply()` hoặc `commit()` để lưu thay đổi.

## 2.8. Hiệu ứng trong android

### 2.8.1. Hiệu ứng cơ bản

- a. *Fade in*: Là hiệu ứng làm mờ đối tượng.
- b. *Fade out*: Là hiệu ứng làm mờ dần đối tượng.
- c. *Blink*: Là hiệu ứng làm nhấp nháy đối tượng.
- d. *Zoom in*: Là hiệu ứng phóng to đối tượng.
- e. *Zoom out*: Là hiệu ứng thu nhỏ đối tượng.
- f. *Rotate*: Là hiệu ứng xoay đối tượng.
- g. *Move*: Là hiệu ứng dịch chuyển đối tượng.
- h. *Side up*: Là hiệu ứng trượt đối tượng lên trên.
- i. *Side down*: Là hiệu ứng trượt đối tượng xuống dưới.
- j. *Sequential animation*: Lặp lại một trong các hiệu ứng trên.
- k. *Together animation*: Xây ra đồng thời hai hoặc nhiều trong các hiệu ứng trên.

### 2.8.2. Cách sử dụng

- Bước 1: Tạo tập tin xml định nghĩa hiệu ứng  
(Tập tin này được đặt trong thư mục **anim** dưới thư mục **res** (res ⇒ anim ⇒ xml))
- Bước 2: Nạp hiệu ứng

Tại activity tạo một đối tượng của lớp Animation và nạp tập tin xml sử dụng phương thức **loadAnimation()** của lớp **AnimationUtils**

- Bước 3: Thiết lập sự kiện (Tuỳ chọn)

Nếu bạn muốn xử lý các sự kiện như start, end và repeat, bạn phải cài đặt giao diện `AnimationListener` cho activity của bạn. Lưu ý bước này là tuỳ chọn, bạn có thể bỏ qua nếu không cần thiết.

- Bước 4: Bắt đầu hiệu ứng

Chúng ta có thể bắt đầu một animation bất cứ khi nào mình muốn bằng cách gọi phương thức `startAnimation()` trên bất kỳ thành phần UI nào mà chúng ta muốn thiết lập animation.

## 2.9. SQLite

### 2.9.1. Giới thiệu SQLite

SQLite là một cơ sở dữ liệu quan hệ, mã nguồn mở và được tích hợp sẵn trên Android.

SQLite thường được sử dụng trong các ứng dụng ở Local, như các ứng dụng Danh bạ, Tin nhắn, Ghi chú, Quản lý thông tin cá nhân, Các tùy chọn thiết lập (Setting) trong phần mềm,...etc...

### 2.9.2 Cấu hình SQLite

NHÓM TRUY VẤN KHÔNG TRẢ VỀ DỮ LIỆU:

- Tạo một bảng: `Create Ten_bang (Danh_sach_cac_cot Kieu_du_lieu(Kich_thuoc))`.

- Xóa một bảng: `Drop Table Ten_bang`.

- Thêm một dòng vào bảng: `Insert Into Ten_bang(Cac_cot) Values (Gia_tri_tuong_ung)`.

- Sửa một dòng trong bảng: `Update Ten_bang Set Gia_tri Where Dieu_kien`.

- Xóa một dòng trong bảng: `Delete From Ten_bang Where Dieu_kien`.

NHÓM TRUY VẤN TRẢ VỀ DỮ LIỆU:

- Lấy ra các dòng trong bảng: `Select Ten_cot From Ten_bang Where Dieu_kien`.

- Lấy ra tất cả dòng trong bảng: `Select * From Ten_bang`.

Tạo một project mới trong Android Studio, tạo thêm một Java Class mới và đặt tên là DataSqlite. Sau đó cho nó extends từ SQLiteOpenHelper. Lúc này Class sẽ báo lỗi do thiếu một vài phương thức, lần lượt tạo một Constructor và 2 phương thức onCreate và onUpgrade cho Class để hết lỗi.

VD: Để lấy dữ liệu từ bảng thucPhamFinal.sqlite đã có sẵn thông tin về tên thực

```
class DataSqlite extends SQLiteOpenHelper {

    DataSqlite(Context context, String name, SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
    }

    //Hàm phục vụ cho việc truy vấn không trả về dữ liệu:
    void TruyVanKhongTraVe(String sql)
    {
        SQLiteDatabase db=getWritableDatabase();
        db.execSQL(sql);
    }

    //Hàm phục vụ cho việc truy vấn trả về dữ liệu:
    Cursor TruyVanTraVe(String sql)
    {
        SQLiteDatabase db=getWritableDatabase();
        return db.rawQuery(sql, null);
    }

}
```

phẩm, lượng calories để show ra cho người dùng chọn

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="20dp">
    <TextView
        android:id="@+id/foodid"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="1" />
    <TextView
        android:id="@+id/foodname"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="FoodName"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/calories"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="right"
        android:text="1000" />
</LinearLayout>

```

## 2.10. Fragment

### 2.10.1. Tổng quan

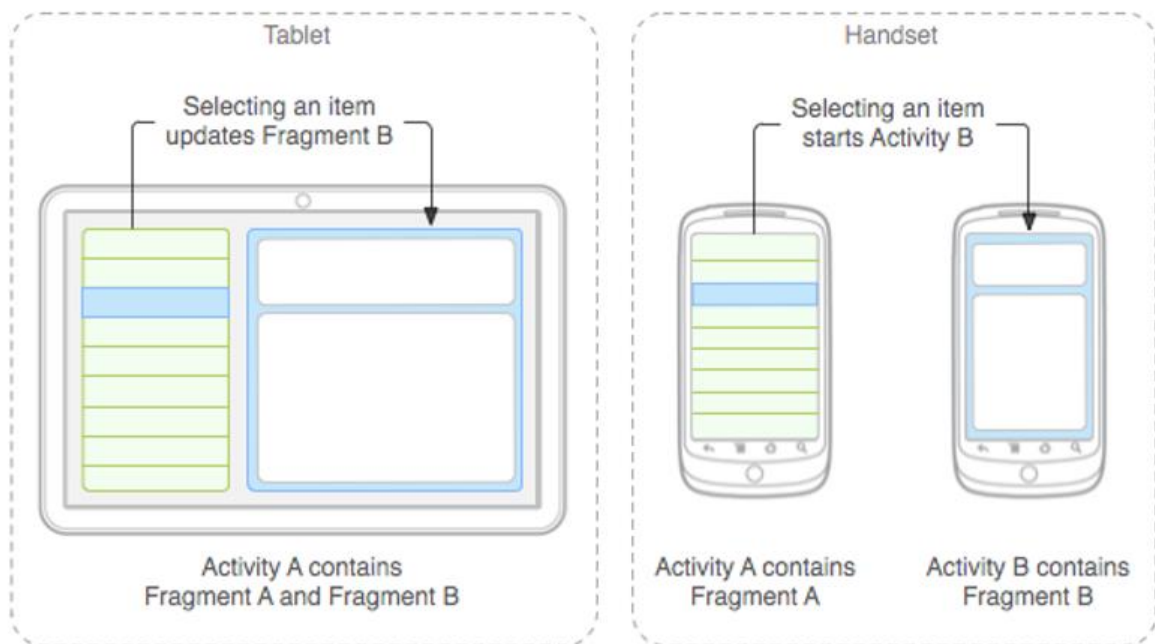
#### a. Lý do ra đời của fragment

Lý do cần phải tạo ra fragment?

- Trước đây, trên Honeycomb, Android giới thiệu khái niệm Fragments. ta có thể xem nó như là các khối được xây dựng riêng biệt với vòng đời

riêng trong một Activity. Nó hỗ trợ rất nhiều trong việc **tối ưu cho các loại màn hình**, đồng thời dễ dàng được quản lý bởi activity cha, có thể **sử dụng lại, kết hợp và bố trí theo ý muốn**.

- Việc chạy từng activity riêng cho mỗi màn hình ứng dụng sẽ có hiệu quả rất tệ khi hệ thống phải cố lưu trữ chúng trong bộ nhớ lâu hết mức có thể. Tắt một cái trong số đó cũng không giải phóng các tài nguyên được sử dụng bởi những cái còn lại.



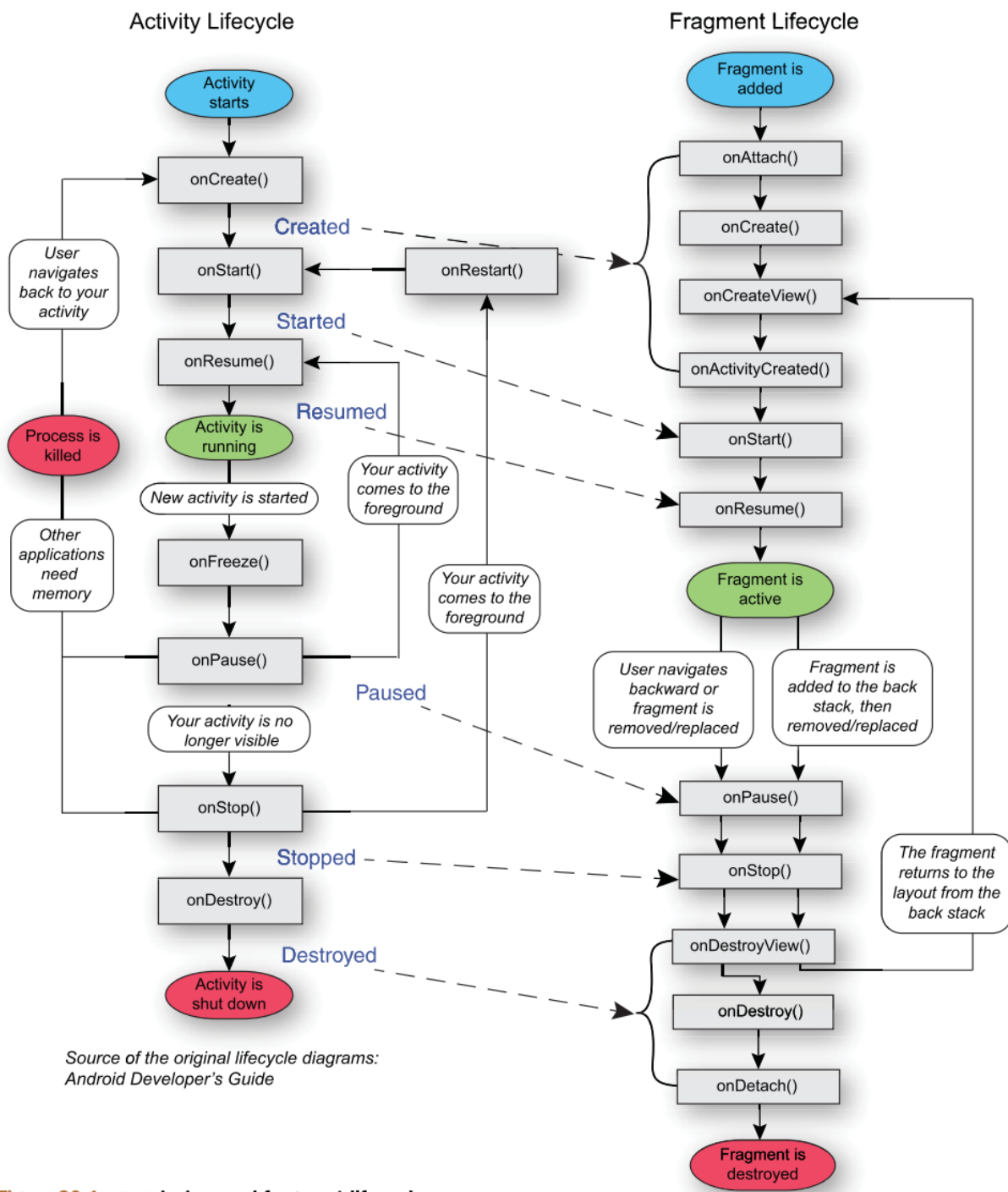
## b. Vòng đời của fragment

- Fragment là một thành phần android độc lập có vòng đời và giao diện riêng được quản lý bởi một activity và hoạt động giống như một sub-activity
- Vòng đời của fragment bị ảnh hưởng trực tiếp bởi vòng đời của activity chứa nó. Tức là, khi activity bị tạm dừng thì tất cả các fragment được chứa bởi activity đó cũng tạm dừng, và khi activity bị hủy thì tất cả các fragment bên trong cũng bị hủy theo.



- Khi Fragment được gắn vào Activity, các callback **onAttach()**, **onCreate()**, **onCreateView()**, **onActivityCreated()**, **onStart()**, **onResume()** lần lượt được gọi.
- Sau khi các callback trên được gọi, fragment lúc đó mới chính thức được xem là đang chạy.

Sau đó, nếu người dùng bấm nút **Back** hay có bất kì thao tác gỡ/ thay thế fragment ra khỏi activity nào thì các callback **onPause()**, **onStop()**, **onDestroyView()**, **onDestroy()**, **onDetach()** sẽ được gọi. (Đây là trường hợp fragment chưa được thêm vào back stack. Ở phía dưới mình sẽ nói về trường hợp fragment được thêm vào back stack sau).



**Figure 20.1** Activity and fragment lifecycles

Hình 29.2. Vòng đời của Activity và Fragment

Như vậy, việc quản lý vòng đời của một fragment rất giống với quản lý vòng đời của một activity. Giống như activity, fragment có thể tồn tại ở **3 trạng thái**:

**Hoạt động (Resume):** Khi fragment được gắn vào activity, có thể nhìn thấy và có thể tương tác được.

**Tạm dừng (Pause):** Nếu activity chứa fragment bị che lấp bởi 1 activity khác nhưng không bị che hoàn toàn, người dùng vẫn nhìn thấy được activity bị che lấp, chỉ là không tương tác được thì cả activity và fragment đều đi vào trạng thái tạm dừng.

**Dừng (Stop):** Cũng giống như activity, fragment bị dừng khi bị thành phần nào đó che mất hoàn toàn. Ở trạng giai đoạn, các trạng thái của của fragment vẫn được giữ lại phòng trường hợp fragment được hiển thị trở lại. Và nếu nó không còn được hiển thị với người dùng thì fragment sẽ bị gỡ bỏ nếu activity bị hủy.

## 2.10.2. Sử dụng

### 1. Tạo và hiển thị fragment

#### a. *Hiển thị kiểu tĩnh*

Đây là cách thực hiện rất nhanh chóng. Người lập trình chỉ cần sử dụng một layout có tên là fragment để hiển thị một fragment mà bạn mong muốn. Layout fragment này cũng cần bạn chỉ định các thuộc tính android:layout\_width và android:layout\_height như các layout khác. Chính vì vậy bạn có thể thiết kế bao nhiêu fragment vào trong giao diện của Activity đều được, và đặt chúng vào vào bất cứ vị trí nào bạn muốn. Chính thuộc tính android:name của thẻ fragment này sẽ giúp bạn chỉ định fragment nào cần hiển thị.

#### b. *Hiển thị theo kiểu động*

Nếu như với cách hiển thị tĩnh trên kia, bạn phải chỉ định thẻ fragment nào sẽ chứa đựng Fragment nào một cách cố định. Thì với cách hiển thị động này, bạn chỉ cần khai báo một vùng không gian nào đó sẽ chứa đựng fragment, vùng không gian đó được khai báo bằng một FrameLayout.

## 2.10.3. Tổng kết

Fragment là một thành phần android độc lập, được sử dụng bởi một activity, giống như một sub-activity.

Fragment có vòng đời và giao diện riêng.

Các Fragment thường có một file java đi kèm với file giao diện xml. Các fragment không có file giao diện xml thường được gọi là headless fragments.

Vòng đời của fragment bị ảnh hưởng trực tiếp bởi vòng đời của activity chủ. Ví dụ, khi hoạt động bị tạm dừng, tất cả phân đoạn trong nó cũng vậy, và khi hoạt động bị hủy, tất cả phân đoạn cũng vậy.

Một Fragment có thể được sử dụng trong nhiều Activity.

Fragment được thêm vào API 11 trở lên.

Fragment sử dụng phương thức `getActivity()` để lấy ra Activity cha

Fragment được định nghĩa trong file xml của activity (static definition) hoặc có thể sửa đổi fragment khi đang chạy (dynamic definition)

## CHƯƠNG 3: CHƯƠNG TRÌNH THỰC NGHIỆM

### 3.1. Giới thiệu

Ngày nay, với sự phát triển không ngừng của xã hội, con người ngày một quan tâm đến sức khỏe và vóc dáng bản thân hơn. Bên cạnh việc tập luyện thể dục thể thao thì việc dung nạp lượng calo thông qua thực phẩm hàng ngày cũng rất quan trọng, có thể chiếm đến 70% hiệu quả tác động đến cơ thể. Chính vì lẽ đó, việc lựa chọn các thực phẩm để lên thực đơn hàng ngày là rất cần thiết để cân bằng lượng dinh dưỡng sao cho không vượt quá nhu cầu calo cần thiết cho cơ thể hàng ngày.

Trên thực tế, đã có một số ứng dụng trên thiết bị di động để lên thực đơn cho người dùng nhưng hầu hết đều bằng tiếng Anh và các thực phẩm không phù hợp với người Việt Nam, vì thế việc xây dựng một ứng dụng phù hợp với người Việt là cần thiết.

Bên cạnh đó, các thiết bị di động và đặc biệt là điện thoại thông minh đang ngày càng trở nên phổ biến. Hầu hết mỗi người dùng đều sở hữu một điện thoại cầm tay phục vụ cho việc liên lạc và các tiện ích cá nhân. Do đó, việc phát triển các ứng dụng trên điện thoại di động giúp người dùng có thể lưu trữ, xử lý các công việc một cách thuận tiện và đảm bảo tính riêng tư. Đây chính là lý do để em lựa chọn đề tài xây dựng ứng dụng trên điện thoại di động “Lên thực đơn theo lượng calo một cách khoa học” nhằm đáp ứng những nhu cầu thiết thực trong cuộc sống hiện đại ngày nay.

### 3.2. Khái niệm về calories

#### 3.2.1. Khái niệm

Calories (hay còn được gọi là calo) là một đơn vị đo năng lượng, để tính hàm lượng dinh dưỡng trong các loại thức ăn được đưa vào cơ thể. Khi mà cơ thể nạp vào thức ăn và biến nó thành calo (năng lượng) để duy trì sự sống và thực hiện tất cả các hoạt động.

Tất cả các loại thức ăn, nước uống (trừ nước lọc) đều chứa một lượng calo nhất định, tùy thuộc vào các chất:

- Tinh bột
- Đạm
- Béo

- Xơ

### 3.2.2. Chức năng của calo

Số lượng calo trong thực phẩm là thước đo thực phẩm sở hữu bao nhiêu năng lượng tiềm năng như đã nói ở trên. Thực phẩm là sự sắp xếp kết hợp của ba khối hợp nhất này. Vì vậy, nếu biết có bao nhiêu Carbohydrate, chất béo và protein trong thực phẩm bất kỳ nào đó, bạn sẽ biết thực phẩm chứa bao nhiêu calo hoặc bao nhiêu năng lượng.

Ví dụ như trong 1 gói yến mạch, ta thường thấy ghi rằng là nó có 160 calo. Điều này nghĩa là chúng ta phải đổ bột yến mạch này có một cái bát, đun trong lửa và làm nó cháy hoàn toàn, phản ứng sẽ tạo ra 160 kiloCalo. Lưu ý calo được in trên thực phẩm là kiloCalo – đủ năng lượng để tăng nhiệt độ 160 kg nước lên 1 độ C.

Cơ thể phải “đốt cháy” calo trong bột yến mạch qua quá trình trao đổi chất, các enzyme phá vỡ các Carbohydrate thành glucose và các loại đường khác, chất béo thành glycerol và axit béo, protein thành các axit amin.

Những phân tử này sau đó được vận chuyển bằng đường mạch máu tới các tế bào. Ở các tế bào, chúng lựa chọn:

- Được hấp thụ để cơ thể sử dụng ngay
- Được đưa thẳng vào giai đoạn cuối của quá trình trao đổi chất và phản ứng hóa học với oxy để giải phóng năng lượng lưu trữ.

### 3.2.3. Lượng calo cần thiết hàng ngày cho từng người:

Trẻ em từ 2 – 3 tuổi: 1.000 calo

Trẻ em từ 4 – 8 tuổi: 1.200 – 1.400 calo

Bé gái 9 – 13 tuổi: 1.600 calo

Bé trai 9 – 13 tuổi: 1.800 calo

Bé gái từ 14 – 18 tuổi: 1.800 calo

Bé trai từ 14 – 18 tuổi: 2.200 calo

Nữ từ 19 – 30 tuổi: 2.000 calo

Nam từ 19 – 30 tuổi: 2.400 calo

Nữ từ 31 – 50 tuổi: 1.800 calo

Nam từ 31 – 50 tuổi: 2.200 calo

Nữ trên 51 tuổi: 1.600 calo

Nam trên 51 tuổi: 2.000 calo.

Trên là hàm lượng calo trong các loại thực phẩm hàng ngày được sử dụng khá phổ biến trong các món ăn của người Việt.

Biết được hàm lượng calo trong các loại thực phẩm để bạn có thể lập bảng tính lượng calo phù hợp với bản thân.

### 3.2.4. Bảng dinh dưỡng của các thực phẩm.

STT	Loại	Tên thực phẩm	Năng lượng (kcal)/100g
1	Cháo, phở, miến, mì ăn liền	Bún ăn liền	348
2	Cháo, phở, miến, mì ăn liền	Cháo ăn liền	346
3	Cháo, phở, miến, mì ăn liền	Mì ăn liền	435
4	Củ giàu tinh bột	Củ sắn	152
5	Củ giàu tinh bột	Khoai lang	119
6	Củ giàu tinh bột	Khoai môn	109
7	Dầu, mỡ, bơ	Bơ	756
8	Dầu, mỡ, bơ	Dầu thực vật	897
9	Đồ hộp	Cá thu hộp	207
10	Đồ hộp	Cá trích hộp	233
11	Đồ ngọt	Bánh in chay	376
12	Đồ ngọt	Mật ong	327
13	Gia vị, nước chấm	Cari bột	283
14	Gia vị, nước chấm	Gừng tươi	25
15	Gia vị, nước chấm	Mắm tôm đặc	73
16	Hạt giàu đạm và chất béo	Đậu Hà lan (hạt)	342
17	Hạt giàu đạm và chất béo	Đậu phộng	573

<b>18</b>	Hạt giàu đạm và chất béo	Đậu phụ	95
<b>19</b>	Hạt giàu đạm và chất béo	Đậu tương (đậu nành)	400
<b>20</b>	Ngũ cốc	Bắp tươi	196
<b>21</b>	Ngũ cốc	Bún	110
<b>22</b>	Ngũ cốc	Gạo nếp cái	346
<b>23</b>	Quả chín	Chuối tiêu	97
<b>24</b>	Quả chín	Đu đủ chín	35
<b>25</b>	Quả chín	Dưa hấu	16
<b>26</b>	Thịt	Chân giò lợn (bỏ xương)	230
<b>27</b>	Thịt	Sườn heo bỏ xương	187
<b>28</b>	Thịt	Thịt bê nạc	85
<b>29</b>	Thịt	Thịt gà ta	199
<b>30</b>	Thịt	Thịt gà tây	218
<b>31</b>	Thủy hải sản	Cá giếc	87
<b>32</b>	Thủy hải sản	Cá hồi	136
<b>33</b>	Thủy hải sản	Cá khô	208
<b>34</b>	Thủy hải sản	Cá lóc	97
<b>35</b>	Thủy hải sản	Cá mè	144
<b>36</b>	Thủy hải sản	Tôm biển	82
<b>37</b>	Trứng	Trứng gà	166
<b>38</b>	Trứng	Trứng vịt	184
...	...	..	...



### 3.3. Bài toán lên thực đơn hàng ngày theo lượng calo một cách khoa học

#### Phát biểu bài toán:

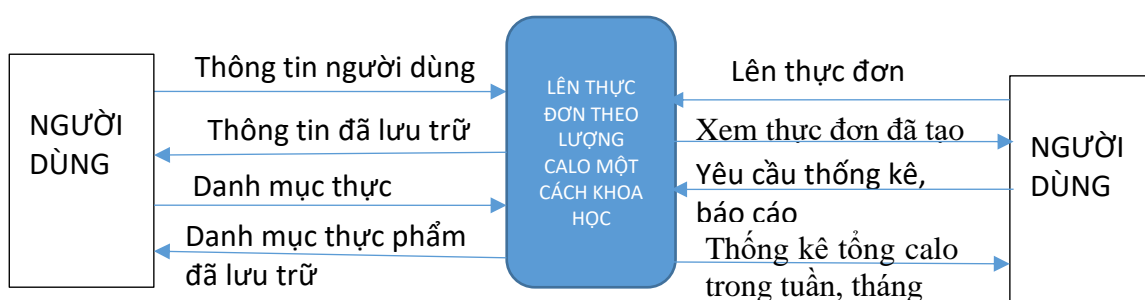
- Dựa trên bảng danh mục các thực phẩm thông dụng của người Việt tính theo lượng calo cho mỗi 100 gram thực phẩm và lượng calo cần thiết của người dùng trong một ngày, hệ thống sẽ cho phép người dùng lên thực đơn cho một ngày bằng cách chọn một số thực phẩm trong số danh mục đó và tính toán lượng calo tương ứng, hệ thống thông báo cho người dùng xem lượng calo còn thiếu để thêm là bao nhiêu để người dùng lựa chọn cho phù hợp, và hiển thị thực đơn gồm các thực phẩm đã chọn.
- Hệ thống hỗ trợ lưu trữ các thực đơn đã chọn để giúp tìm kiếm và thống kê các thực đơn đã tạo để người dùng có cái nhìn tổng quan và so sánh.

Trên đó cho phép thực hiện:

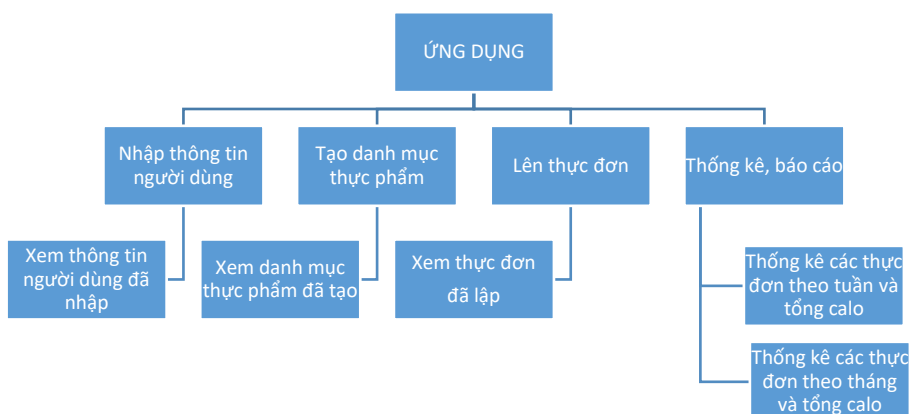
- Người dùng nhập thông tin của họ gồm: Họ tên, tuổi, giới tính, lượng calo cần thiết của cơ thể,
- Lưu trữ thông tin của danh mục các loại thực phẩm phổ biến tại Việt Nam, trong đó gồm: mã thực phẩm, tên thực phẩm, lượng calo của thực phẩm
- Người dùng lên thực đơn cho mỗi ngày để đi mua từ các thực phẩm đã lưu trữ, dựa vào số calo cần thiết, để chọn loại thực phẩm cho ngày hôm đó phù hợp với lượng calo cần thiết.
- Người dùng có thể xem thực đơn gồm các thực phẩm đã lựa chọn cho một ngày.
- Lập báo cáo thống kê các thực đơn sử dụng trong tuần/ tháng và tổng lượng calo đã sử dụng.

### 3.4. Phân tích bài toán

#### 3.4.1. Ngữ cảnh của hệ thống



### 3.4.2. Mô hình chức năng



- *Nhập thông tin người dùng*: Ứng dụng cho phép người dùng nhập thông tin cơ bản như họ tên, giới tính, tuổi, lượng calo cần thiết. Sau khi nhập thông tin người dùng, người dùng có thể xem lại thông tin đã nhập
- *Tạo danh mục thực phẩm*: người dùng nhập danh mục các thực phẩm gồm: mã thực phẩm, tên thực phẩm, lượng calo của thực phẩm. Sau khi tạo danh mục thực phẩm, người dùng có thể xem danh mục thực phẩm đã nhập.
- *Lên thực đơn*: Người dùng chọn thực đơn trong ngày theo danh mục thực phẩm đã lưu trữ và trọng lượng, hệ thống cho biết tổng số lượng calo tương ứng với mỗi thực phẩm đã chọn và trước khi thực hiện thêm vào thực đơn sẽ thông báo cho người dùng biết còn được thêm bao nhiêu lượng calo nữa.

Sau khi lên thực đơn, người dùng có thể Xem thực đơn người dùng đã tạo trong ngày.

- *Thống kê, báo cáo*:
  - o Thống kê các thực đơn sử dụng trong một tuần và tổng calo đã sử dụng
  - o Thống kê các thực đơn sử dụng trong tháng và tổng lượng calo đã sử dụng.

### 3.4.3. Cơ sở dữ liệu

Bảng Người dùng: chứa thông tin người dùng nhập vào

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Mô tả	Ghi chú
1	Id	int		Định danh	Khóa chính
2	hoten	Varchar	50	Họ tên	
3	tuoi	int		Tuổi	
4	gioitinh	Varchar	3	Giới tính	
5	calo	int		Lượng calo	

- Bảng Thực phẩm: chứa danh mục các thực phẩm lưu trữ khoảng 258 thực phẩm thường dùng của người Việt, theo viện dinh dưỡng Việt Nam, gồm các thông tin sau: Mã thực phẩm, tên thực phẩm, lượng calo thực phẩm.

Tên trường	Kiểu	Kích thước	Mô tả	Ghi chú
idthucpham	Integer		Mã thực phẩm	primarykey
tenthucpham	Varchar	50	Tên thực phẩm	
calothucpham	Integer		Calo thực phẩm	

- Bảng Thực đơn: chứa các thực phẩm được chọn trong một thực đơn

Tên trường	Kiểu	Kích thước	Mô tả	Ghi chú
ngaydon	Date		Ngày lấy thực đơn đơn	primarykey
idthucpham	Integer		Mã thực phẩm	primarykey
trongluong	Integer		Trọng lượng	

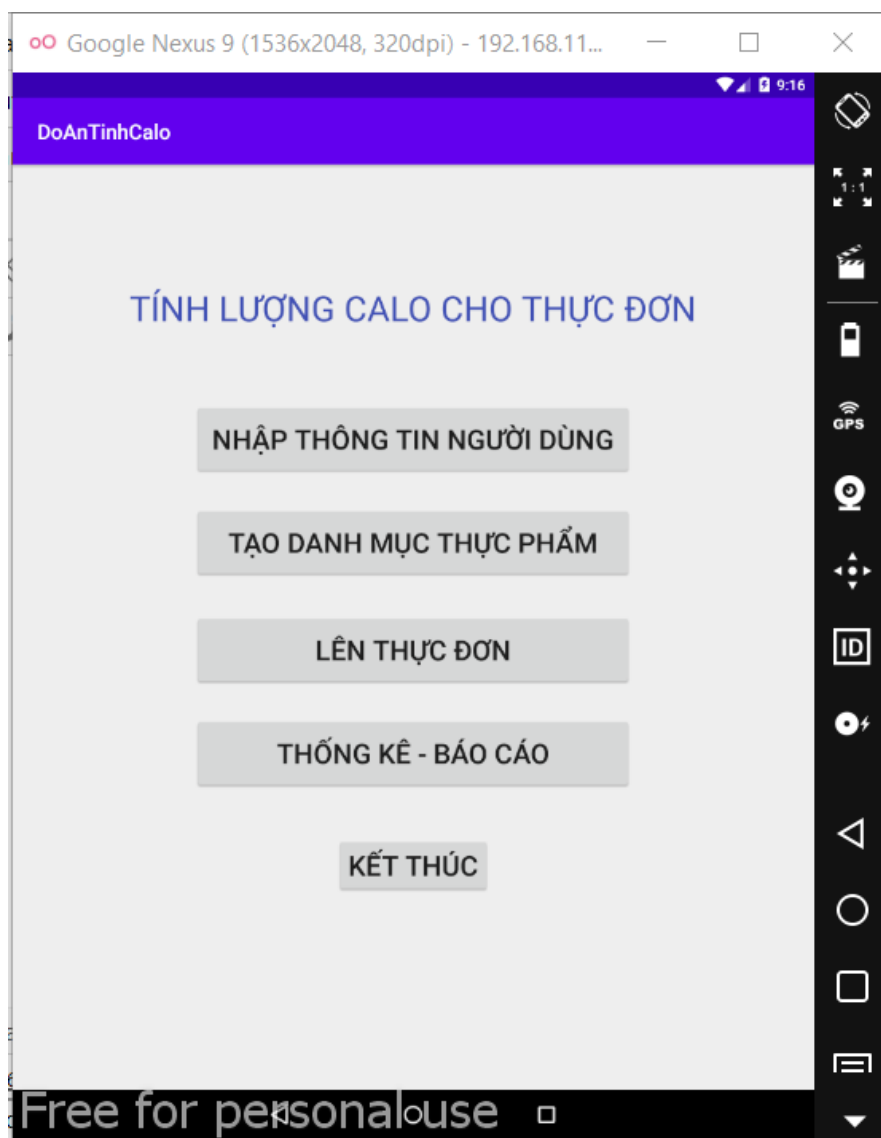
### 3.5 Cài đặt chương trình

#### 3.5.1. Thiết bị và môi trường lập trình

- Máy tính: Core i5, 2.3 Ghz, 8GB Ram, ổ cứng 500 GB.
- Điện thoại di động hệ điều hành android.
- Môi trường lập trình:     Hệ điều hành windows 10,  
                                  Ngôn ngữ lập trình Java, SDK 8.0  
                                  Android Studio 4.0

#### 3.5.2. Ứng dụng hỗ trợ lên thực đơn khoa học theo lượng calo

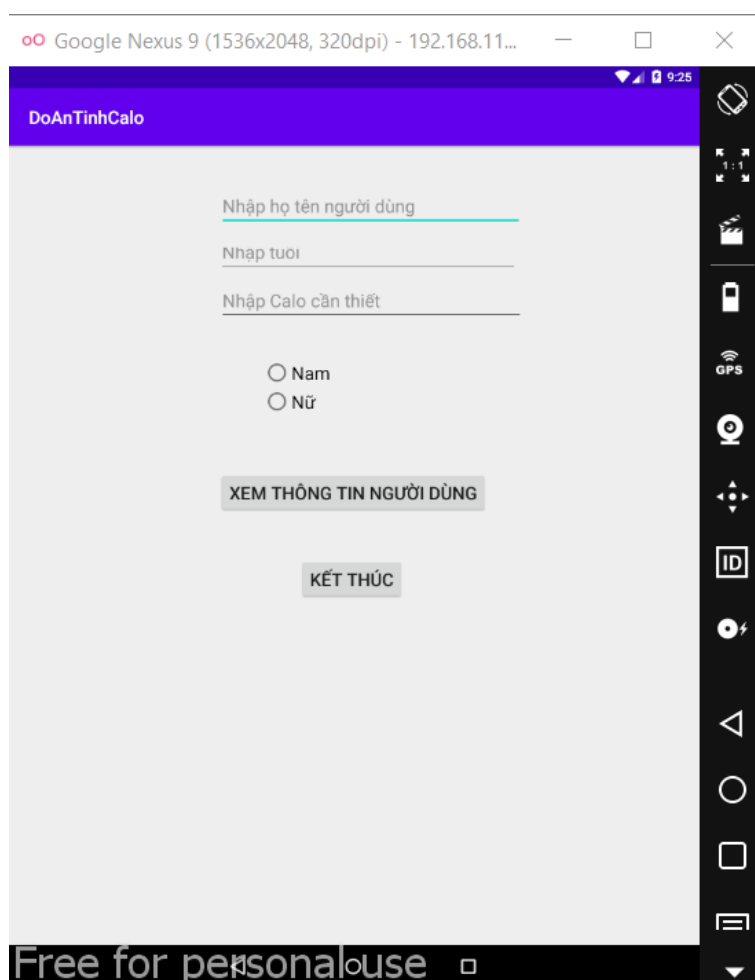
##### 3.5.2.1. Giao diện chương trình



Trong đó:

- Chức năng nhập thông tin người dùng cho phép người dùng nhập các thông tin cần thiết: Họ tên, tuổi, giới tính, lượng calo cần thiết và thực hiện xem lại thông tin đã nhập và nhập lại nếu cần thiết và kết thúc.
- Chức năng tạo Danh mục thực phẩm: tạo danh mục các thực phẩm thông dụng của người Việt Nam, theo Viện dinh dưỡng Việt Nam thì có khoảng 258 thực phẩm thông dụng, trong đó cho phép xem danh mục thực phẩm đã nhập.
- Chức năng lên thực đơn: người dùng chọn nhập thực phẩm từ danh mục thực phẩm đã tạo và trọng lượng tương ứng theo lượng calo cần thiết, và hiển thị thực đơn đã tạo.
- Chức năng thống kê báo cáo: Cho phép thực hiện hai mức thống kê: thống kê các thực đơn và lượng calo đã sử dụng theo tuần và thống kê các thực đơn và lượng calo đã sử dụng theo tháng.

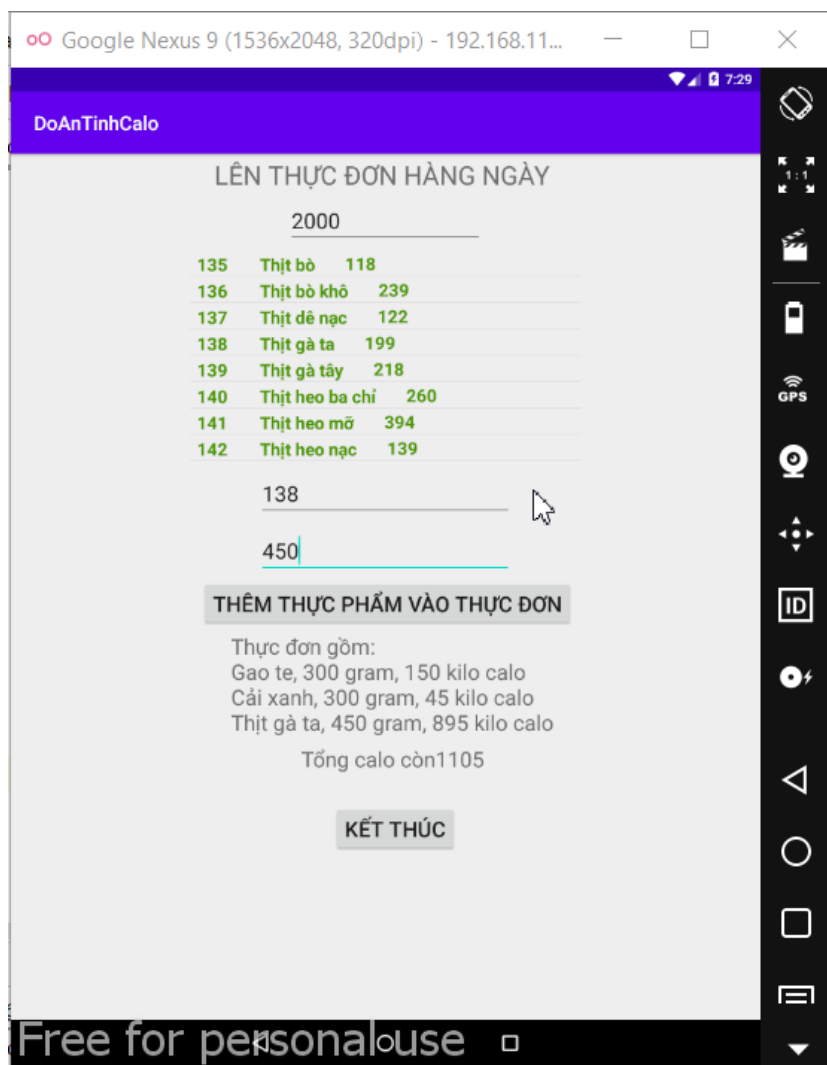
### 3.5.2.2. Nhập thông tin người dùng



The screenshot shows the 'DoAnTinhCalo' app interface on a mobile device. The app title is 'DoAnTinhCalo'. The screen displays a form for entering user information. The form includes three text input fields: 'Nhập họ tên người dùng', 'Nhập tuổi', and 'Nhập Calo cần thiết'. Below these fields are two radio button options for gender: 'Nam' and 'Nữ'. At the bottom of the form, there are two buttons: 'XEM THÔNG TIN NGƯỜI DÙNG' and 'KẾT THÚC'. The app is running on a Google Nexus 9 device, as indicated by the status bar at the top. The time is 9:25. The bottom of the screen shows a watermark 'Free for personal use'.

### 3.5.2.3. Lên thực đơn

Người dùng chọn thực phẩm vào thực đơn dựa trên danh mục thực phẩm đã được tạo phù hợp với lượng calo ban đầu đã nhập.



### 3.6. Thiết kế cơ sở dữ liệu

Sử dụng SQLite tạo CSL với hai bảng dữ liệu là Thực phẩm và Thực đơn.

### 3.7. Các yêu cầu đối với người dùng hệ thống.

- Người sử dụng phải là Cán bộ, giảng viên, nhân viên và sinh viên trong trường
- Người dùng phải có thiết bị di động sử dụng hệ điều hành Android, ví dụ:
  - + Điện thoại : Sam sung , Oppo , Xiaomi , Vinsmart , Nexus,...

- + Máy tính bảng : Samsung Galaxy Tab , Masstel Tab , Huawei MediaPad Lenovo Yoga Tab ...
  - + Smart TV chạy android của các hãng như : Samsung , LG...
  - + TV box : FPT Play Box , Xiami Mibox , VNPT Smart Box .....
- Và một số thiết bị khác

## KẾT LUẬN

- Kết quả đạt được của đồ án
  - Trong thời gian thực hiện đề tài, em đã tìm hiểu cài đặt và học lập trình Android để có thể Xây dựng ứng dụng hỗ trợ lên thực đơn khoa học theo lượng calo.
  - Trong ứng dụng này em đã thực hiện được một số nhiệm vụ cơ bản như: nhập thông tin của người dùng, Tạo danh mục thực phẩm, Lên thực đơn theo lượng calo đã nhập và hiển thị thực đơn.
  - Với danh mục thực phẩm phổ biến của người Việt nam theo lượng calo được công bố từ Viện dinh dưỡng Việt Nam, người dùng có thể chọn được các thực phẩm phù ợp và xác định được lượng calo tương ứng phù hợp với bản thân, nhờ đó họ toote chức được các bữa ăn phong phú và đảm bảo lượng dinh dưỡng cần thiết và giữ gìn sức khỏe, vóc dáng.
- Những hạn chế
  - Trong khoảng thời gian ngắn để thực đề tài, em mới chỉ xây dựng được ứng dụng với các chức năng cơ bản theo yêu cầu của hệ thống, chưa lưu trữ được toàn bộ các thực đơn đã tạo để so sánh đối chiếu.
  - Ứng dụng hạn chế về giao diện và các thông điệp trao đổi khi người dùng tương tác
- Hướng phát triển tiếp theo: Trong thời gian tới, em sẽ chỉnh sửa lại giao diện cho gần gũi dễ sử dụng,, thêm các chức năng chỉnh sửa dữ liệu và thực hiện các thống kê báo cáo.