

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA CÔNG NGHỆ PHẦN MỀM



**BÁO CÁO**  
**ĐỒ ÁN CHUYÊN NGÀNH**

**ĐỀ TÀI**  
**HỆ THỐNG QUẢN GIA ĐIỆN TỬ**  
**TRÊN THIẾT BỊ DI ĐỘNG**

**Giảng viên hướng dẫn: Th.S Trần Anh Dũng**

**Sinh viên thực hiện: 12520026 – Phan Y Biển**

**12520328 – Võ Hoài Phương**

**TP Hồ Chí Minh, Tháng 6 năm 2016**

## MỤC LỤC

<b>LỜI CẢM ƠN.....</b>	<b>1</b>
<b>LỜI NÓI ĐẦU .....</b>	<b>2</b>
<b>CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI .....</b>	<b>3</b>
I.1. Tổng quan hiện trạng.....	3
I.2. Mục tiêu .....	4
I.3. Đối tượng và nội dung nghiên cứu.....	4
I.4. Phạm vi nghiên cứu .....	5
<b>CHƯƠNG II: CƠ SỞ LÝ THUYẾT.....</b>	<b>6</b>
II.1. Giới thiệu về Android.....	6
II.1.1. Lịch sử .....	6
II.1.2. Kiến trúc .....	8
II.2. Android Studio .....	11
II.2.1. Tổng quan .....	11
II.2.2. Cấu trúc ứng dụng Android .....	19
II.2.3. Chu kỳ ứng dụng Android .....	20
II.2.4. Các thành phần giao diện trong Android.....	25
II.3. Biểu đồ trong Android.....	31
II.3.1. Tổng quan .....	31
II.3.2. Thư viện AchartEngine.....	32
<b>CHƯƠNG III: TÌM HIỂU CÔNG NGHỆ HỒNG NGOẠI IR.....</b>	<b>35</b>
III.1. Giới thiệu.....	35
III.2. Mô tả giao thức hoạt động.....	35
III.3. Data Format .....	39
III.3.1. Giao thức truyền tải Philips RC5 .....	39
III.3.2. Giao thức truyền tải NEC .....	41
<b>CHƯƠNG IV: QUẢN GIA ĐIỆN TỬ TRÊN THIẾT BỊ DI ĐỘNG ....</b>	<b>44</b>
IV.1. Đặc tả Use Case.....	44
IV.1.1. Thêm thiết bị .....	44

IV.1.2.	Cập nhật tên thiết bị.....	45
IV.1.3.	Xóa thiết bị .....	45
IV.1.4.	Hiển thị danh sách loại thiết bị.....	46
IV.1.5.	Tìm kiếm loại thiết bị .....	47
IV.1.6.	Hiển thị danh sách hãng thiết bị .....	47
IV.1.7.	Tìm kiếm hãng thiết bị .....	48
IV.1.8.	Hiển thị danh sách mẫu thiết bị.....	49
IV.1.9.	Tìm kiếm mẫu thiết bị .....	49
IV.1.10.	Cập nhật cơ sở dữ liệu .....	50
IV.1.11.	Gửi phản hồi .....	51
IV.1.12.	Thay đổi thông số cài đặt .....	52
IV.1.13.	Bật/Tắt chức năng cảnh báo sử dụng thiết bị .....	52
IV.1.14.	Cài đặt chế độ cảnh báo.....	53
IV.1.15.	Lưu lịch sử hoạt động của thiết bị .....	54
IV.1.16.	Thống kê thời gian hoạt động của thiết bị.....	55
IV.1.17.	Cảnh báo thời gian hoạt động của thiết bị.....	56
IV.1.18.	Đánh giá tuổi thọ của thiết bị .....	57
IV.2.	Biểu đồ hoạt động.....	58
IV.2.1.	Thêm thiết bị .....	58
IV.2.2.	Cập nhật tên thiết bị.....	58
IV.2.3.	Xóa thiết bị .....	59
IV.2.4.	Hiển thị danh sách loại thiết bị.....	59
IV.2.5.	Tìm kiếm loại thiết bị .....	60
IV.2.6.	Hiển thị danh sách hãng thiết bị .....	60
IV.2.7.	Tìm kiếm hãng thiết bị .....	61
IV.2.8.	Hiển thị danh sách mẫu thiết bị.....	61
IV.2.9.	Tìm kiếm mẫu thiết bị .....	62
IV.2.10.	Cập nhật cơ sở dữ liệu .....	62
IV.2.11.	Gửi phản hồi .....	63
IV.2.12.	Thay đổi thông số cài đặt .....	63
IV.2.13.	Bật/Tắt chức năng cảnh báo sử dụng thiết bị .....	64

IV.2.14.	Cài đặt chế độ cảnh báo.....	65
IV.2.15.	Lưu lịch sử hoạt động của thiết bị.....	66
IV.2.16.	Thông kê thời gian hoạt động của thiết bị.....	67
IV.2.17.	Cảnh báo thời gian hoạt động của thiết bị.....	67
IV.3.	Thiết kế dữ liệu.....	68
IV.3.1.	MAUTHIETBI .....	68
IV.3.2.	LOATHIETBI.....	68
IV.3.3.	HANGTHIETBI.....	69
IV.3.4.	THIETBI.....	69
IV.3.5.	CHUCNANG .....	69
IV.3.6.	DIEUKHIEN .....	69
IV.3.7.	HOATDONG .....	70
IV.3.8.	THONGKE.....	70
IV.3.9.	THAMSO .....	70
IV.4.	Thiết kế giao diện .....	71
IV.4.1.	Danh sách các màn hình.....	71
IV.4.2.	Mô tả chi tiết.....	72
IV.5.	Cài đặt và thử nghiệm .....	82
IV.5.1.	Môi trường phát triển ứng dụng .....	82
IV.5.2.	Môi trường triển khai ứng dụng .....	82
<b>CHƯƠNG V:</b>	<b>TỔNG KẾT.....</b>	<b>83</b>
V.1.	Nhận xét.....	83
V.1.1.	Ưu điểm .....	83
V.1.2.	Nhược điểm.....	83
V.2.	Hướng phát triển.....	83
V.3.	Kiến thức đạt được .....	83
V.4.	Kết luận .....	84
<b>TÀI LIỆU THAM KHẢO .....</b>		<b>85</b>

## LỜI CẢM ƠN

Trong quá trình thực hiện báo, nhóm xin được gửi lời cảm ơn đến thầy Trần Anh Dũng đã tạo điều kiện và tận tình hướng dẫn cho nhóm các bước tìm hiểu, nghiên cứu đề tài này. Bên cạnh đó, nhóm còn được trau dồi thêm nhiều kiến thức, kinh nghiệm có thể áp dụng vào các dự án thực tiễn.

Ngoài những kết quả đạt được, nhóm vẫn còn những thiếu sót cần cải thiện trong quá trình thực hiện đồ án. Thông qua môn học này, nhóm mong rằng có thể học hỏi thêm nhiều kinh nghiệm hơn nữa và áp dụng những kinh nghiệm đó cho các dự án sắp tới.

Nhóm xin chân thành cảm ơn!

Sinh viên thực hiện

Phan Y Biễn

Võ Hoài Phương

## LỜI NÓI ĐẦU

Đời sống xã hội hiện nay ngày càng phát triển, các nhu cầu thường ngày của con người như công việc, học tập, giải trí... ngày càng được nâng cao cùng với sự phát triển nhanh chóng của công nghệ. Trong lúc đó, các thiết bị điện tử phục vụ các nhu cầu thường ngày của con người được sử dụng rộng rãi trong gia đình, trường học ... cũng có sự phát triển nhanh chóng về cả số lượng, chất lượng, công nghệ hiện đại giúp người sử dụng dễ dàng điều khiển, quản lý thông qua các công nghệ điều khiển từ xa.

Sự phát triển của các thiết bị điện tử cùng với công nghệ điều khiển từ xa đem lại nhiều tiện ích hơn cho người sử dụng như có thể điều khiển thiết bị ở bất kỳ đâu, quản lý được hiệu suất làm việc của thiết bị... Nhưng điều đó cũng mang lại một số bất lợi trong việc điều khiển nhiều thiết bị khác nhau bằng các bộ điều khiển khác nhau cùng một lúc dẫn đến việc khó khăn trong quản lý các thiết bị điều khiển, không thể sử dụng thay thế giữa các bộ điều khiển với nhau khi bị thất thoát, ngoài ra các thiết bị điều khiển còn có thể bị hỏng hóc sau một thời gian sử dụng.

Chính vì lý do đó, ứng dụng **Quản gia điện tử trên thiết bị di động** được xây dựng nhằm mục đích đem lại sự tiện ích, tối ưu hóa việc điều khiển, quản lý các thiết bị điện tử gia dụng cho người dùng trên chính thiết bị di động của mình.

## CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

### I.1. Tổng quan hiện trạng

Tại Việt Nam, nhu cầu sử dụng các thiết bị di động nói chung và điện thoại thông minh nói riêng đang có sự phát triển nhanh chóng. Các thiết bị di động phổ biến hiện nay đều chạy trên nhiều hệ điều hành khác nhau nhưng theo như thống kê thì Android là hệ điều hành được sử dụng nhiều nhất, tiếp đến là hệ điều hành iOS, phần còn lại chiếm một tỷ lệ nhỏ là Windows Phone và các hệ điều hành khác. Qua đó, ta có thể thấy việc phát triển ứng dụng trên Android hoặc iOS có thể đem lại hiệu quả cao, quảng bá sản phẩm đến nhiều người dùng một cách tốt nhất.

Hiện nay, có khá nhiều ứng dụng điều khiển các thiết bị điện tử như TV, điều hòa, máy chiếu, máy ảnh kỹ thuật số ... trên thị trường. Nhưng hầu hết các ứng dụng này chỉ hỗ trợ điều khiển các thiết bị thông qua công nghệ điều khiển bằng hồng ngoại hoặc mạng wifi, tiêu biểu như một số ứng dụng: ASmart Remote IR, Zararemove, IR Universal Remote... Tuy nhiên, việc quản lý hiệu suất hoạt động, cảnh báo thời gian hoạt động cho các thiết bị vẫn chưa được thực hiện trong các ứng dụng hiện có.

Thông tin và chức năng của một số ứng dụng điều khiển các thiết bị điện tử đã có trên thị trường:

- ASmart Remote IR:
  - ASmart Remote IR là một phần mềm điều khiển từ xa thông minh bằng hồng ngoại (IR) trên nền tảng Android. Phần mềm có thể sử dụng trên các dòng điện thoại như Samsung Galaxy S4, S5, Note 3, Galaxy Tab/ HTC ONE có hỗ trợ IR (Infra Red) Remote. Bạn có thể sử dụng điện thoại để điều khiển TV, máy điều hòa, máy ảnh kỹ thuật số, đầu đĩa DVD/Bluray, máy chiếu... Phần mềm hỗ trợ hầu hết các thương hiệu, mẫu mã của các thiết bị điện tử.
  - ASmart Remote IR đang được phát hành miễn phí trên CH Play.
- Zazaremove:

- Zazaremote là phần mềm điều khiển từ xa các thiết bị điện tử thông qua hồng ngoại với giao diện ngắn gọn, dễ sử dụng. Với Zazaremote, bạn có thể điều khiển tất cả các loại thiết bị hồng ngoại. Cơ sở dữ liệu của phần mềm có thể hỗ trợ hơn 6.000 nhãn hiệu, 250.000 mẫu mã và hơn 8.000 chức năng điều khiển khác nhau. Ngoài ra, Zazaremote còn có thể sử dụng khi không có kết nối mạng. Các loại thiết bị mà phần mềm có thể điều khiển được: TV, điều hòa, máy chiếu, máy ảnh kỹ thuật số, DVD, quạt điện, bóng đèn...
- Zazaremote đang được phát hành miễn phí trên CH Play.
- IR Universal Remote™ + WiFi:
  - IR Universal Remote™ được sử dụng để biến điện thoại của bạn thành một bộ điều khiển từ xa được xây dựng dựa trên IR Blaster điều khiển các thiết bị điện tử trong nhà. Hơn 1.000.000 thiết bị hỗ trợ và phát triển được thêm vào hàng ngày. IR Universal Remote™ cho phép người dùng tạo ra các chủ đề điều khiển riêng và có thể kiểm soát nhiều thiết bị cùng một lúc. Các thiết bị mà ứng dụng có thể điều khiển: TV, DVD, điều hòa...
  - Ngoài ra, còn có một số hệ thống khác có cùng chức năng tương tự như: Smart TV Remote, Quick Remote...

## **I.2. Mục tiêu**

- Tìm hiểu kiến thức về nền tảng Android và lập trình Java nhằm hỗ trợ tốt cho việc xây dựng ứng dụng.
- Tìm hiểu công nghệ hồng ngoại (IR) để sử dụng xử lý tín hiệu trong ứng dụng.
- Tích lũy, rèn luyện và trau dồi thêm các kỹ năng mềm: kỹ năng làm việc nhóm, lên kế hoạch làm việc, quản lý thời gian, tìm kiếm tài liệu ...

## **I.3. Đối tượng và nội dung nghiên cứu**

- Tìm hiểu cách thức hoạt động và cách sử dụng các bộ điều khiển của các thiết bị điện tử gia dụng cần đưa vào sử dụng trong ứng dụng.
- Tìm hiểu nền tảng Android, cách xây dựng một ứng dụng Android, sử dụng các thư viện cần thiết để xử lý tín hiệu hồng ngoại.



- Tìm hiểu công nghệ IR và cách áp dụng vào một ứng dụng Android.
- Phân tích, thiết kế và xây dựng ứng dụng.
- Cài đặt và kiểm thử.
- Triển khai ứng dụng và nhận phản hồi từ phía người dùng.

#### **I.4. Phạm vi nghiên cứu**

Trong quá trình thực hiện đồ án, có nhiều nền tảng phát triển trên di động và đồng thời cũng có nhiều công nghệ điều khiển từ xa có thể áp dụng cho ứng dụng. Nhưng nhóm chỉ tập trung nghiên cứu phát triển ứng dụng trên nền tảng Android và sử dụng công nghệ hồng ngoại (IR) để xây dựng ứng dụng.

Ứng dụng quản gia điện tử trên thiết bị di động nhóm đang xây dựng bao gồm các chức năng sau:

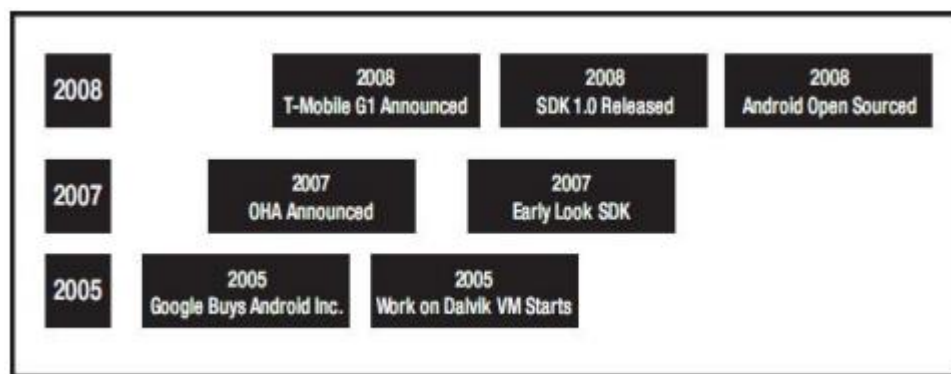
- Chức năng điều khiển hoạt động các thiết bị được hỗ trợ sẵn.
- Chức năng thêm mới các mẫu thiết bị theo nhu cầu người dùng.
- Chức năng quản lý hiệu suất hoạt động của thiết bị.

## CHƯƠNG II: CƠ SỞ LÝ THUYẾT

### II.1. Giới thiệu về Android

#### II.1.1. Lịch sử

Ban đầu, Android là hệ điều hành cho các thiết bị cầm tay dựa trên lõi Linux do công ty Android Inc. (California, Mỹ) thiết kế. Công ty này sau đó được Google mua lại vào năm 2005 và bắt đầu xây dựng Android Platform. Các thành viên chủ chốt tại Android Inc. gồm có: Andy Rubin, Rich Miner, Nick Sears, and Chris White.



Và sau tiếp, vào cuối năm 2007, thuộc về Liên minh Thiết bị Cầm tay Mã Nguồn mở (Open Handset Alliance) gồm các thành viên nổi bật trong ngành viễn thông và thiết bị cầm tay như:

Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, and Vodafone Group,...

Mục tiêu của Liên minh này là nhanh chóng đổi mới để đáp ứng tốt hơn cho nhu cầu người tiêu dùng và kết quả đầu tiên của nó chính là nền tảng Android. Android được thiết kế để phục vụ nhu cầu của các nhà sản xuất thiết, các nhà khai thác và các lập trình viên thiết bị cầm tay.

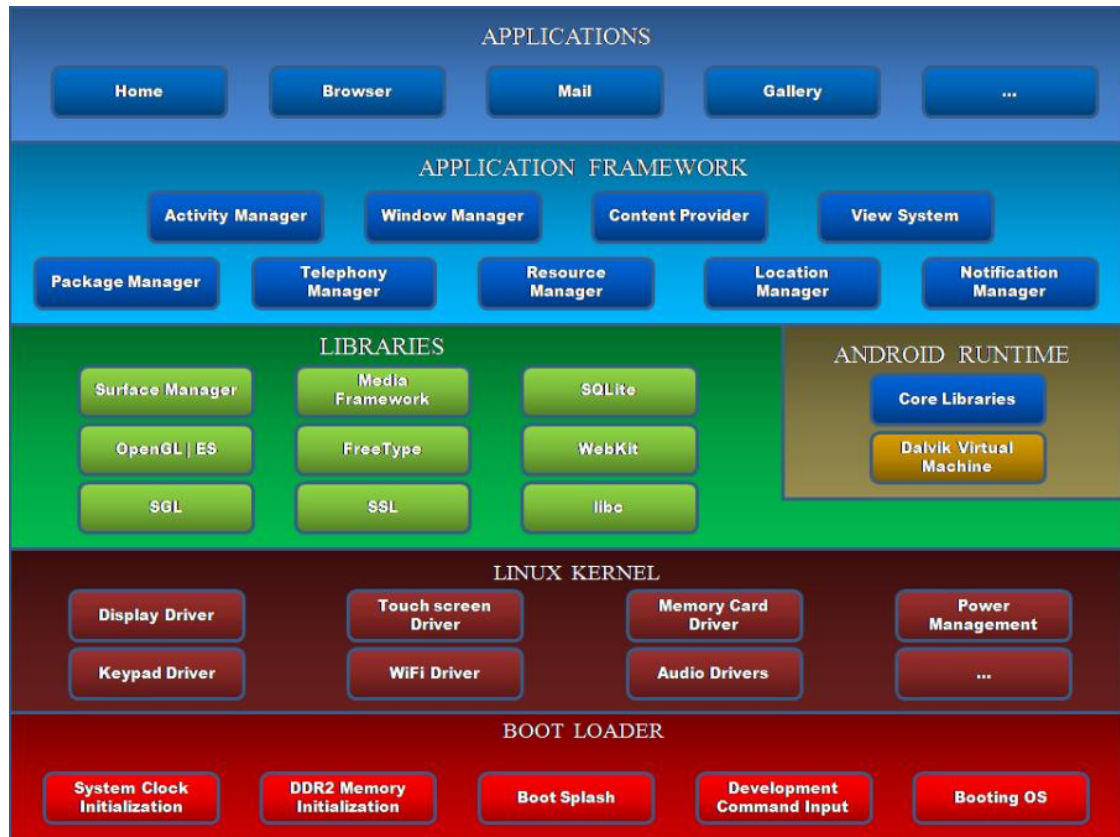
Phiên bản SDK lần đầu tiên phát hành vào tháng 11 năm 2007, hãng T-Mobile cũng công bố chiếc điện thoại Android đầu tiên đó là chiếc T-Mobile G1, chiếc smartphone đầu tiên dựa trên nền tảng Android. Một vài ngày sau đó, Google lại tiếp tục công bố sự ra mắt phiên bản Android SDK release Candidate 1.0. Trong tháng 10 năm 2008, Google được cấp giấy phép mã nguồn mở cho Android Platform.

Khi Android được phát hành thì một trong số các mục tiêu trong kiến trúc của nó là cho phép các ứng dụng có thể tương tác được với nhau và có thể sử dụng lại các thành phần từ những ứng dụng khác. Việc tái sử dụng không chỉ được áp dụng cho các dịch vụ mà nó còn được áp dụng cho cả các thành phần dữ liệu và giao diện người dùng.

Vào cuối năm 2008, Google cho phát hành một thiết bị cầm tay được gọi là Android Dev Phone 1 có thể chạy được các ứng dụng Android mà không bị ràng buộc vào các nhà cung cấp mạng điện thoại di động. Mục tiêu của thiết bị này là cho phép các nhà phát triển thực hiện các cuộc thí nghiệm trên một thiết bị thực có thể chạy hệ điều hành Android mà không phải ký một bản hợp đồng nào. Vào khoảng cùng thời gian đó thì Google cũng cho phát hành một phiên bản vá lỗi 1.1 của hệ điều hành này. Ở cả hai phiên bản 1.0 và 1.1 Android chưa hỗ trợ soft-keyboard mà đòi hỏi các thiết bị phải sử dụng bàn phím vật lý. Android cố định vấn đề này bằng cách phát hành SDK 1.5 vào tháng Tư năm 2009, cùng với một số tính năng khác. Chẳng hạn như nâng cao khả năng ghi âm truyền thông, vật dụng, và các live folder.

### II.1.2. Kiến trúc

Mô hình sau thể hiện một cách tổng quát các thành phần của hệ điều hành Android. Mỗi một phần sẽ được đặc tả một cách chi tiết.



- Tồn ứng dụng

Android được tích hợp sẵn một số ứng dụng cần thiết cơ bản như: contacts, browser, camera, phone,... Tất cả các ứng dụng chạy trên hệ điều hành Android đều được viết bằng Java.

- Application framework

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 lỗi. 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 mình 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.

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.

- Library

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:

- **System C library** – a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices

- **Media Libraries** – based on PacketVideo’s OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG
- **Surface Manager** – manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications
- **LibWebCore** – a modern web browser engine which powers both the Android browser and an embeddable web view
- **SGL** – the underlying 2D graphics engine
- **3D libraries** – an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer
- **FreeType** – bitmap and vector font rendering
- **SQLite** – a powerful and lightweight relational database engine available to all applications
- Android Runtime

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.

- Linux kernel

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.

## II.2. Android Studio

### II.2.1. Tổng quan

#### II.2.1.1. Giới thiệu

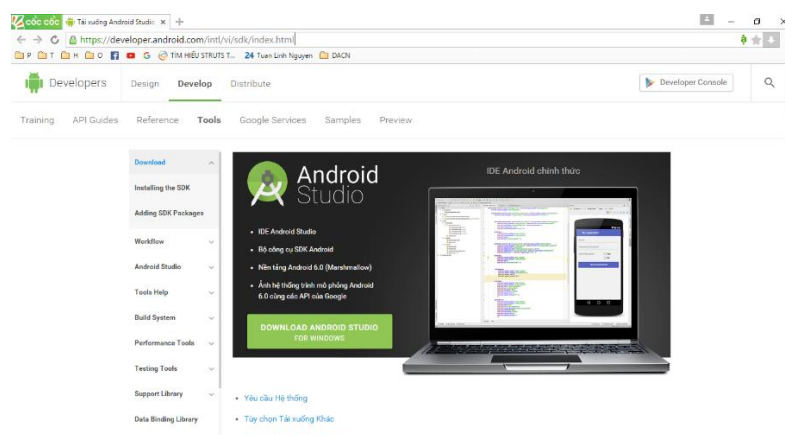
Android Studio là môi trường lập trình phát triển ứng dụng mới vừa được giới thiệu tại Google I/O 2013. Dựa trên “IntelliJ IDEA Community Edition”, công cụ này hoạt động giống WYSIWYG, cho phép lập trình viên tạo ứng dụng, dễ dàng thực hiện các thay đổi và xem trước trong thời gian thực, đồng thời cũng có khả năng tăng tốc sản phẩm, thiết kế giao diện đẹp hơn trước. Đặc biệt là tiếng Việt cũng được hỗ trợ trong Android Studio.

Android Studio hỗ trợ một loạt các giả lập để xem trước ứng dụng, vì vậy ngay cả khi bạn không có thiết bị thử nghiệm, bạn vẫn có thể chắc chắn rằng mọi thứ đều hoạt động trơn tru. Bên cạnh đó, loạt công cụ như lời khuyên tối ưu hóa, đồ thị doanh số bán hàng, và số liệu lấy từ phân tích sẽ giúp các nhà phát triển quản lý ứng dụng đang bán của mình và tìm ra hướng đi cụ thể với từng thiết bị Android.

#### II.2.1.2. Cài đặt

- Tải xuống

<https://developer.android.com/intl/vi/sdk/index.html>



- Yêu cầu trước khi cài đặt

Trước khi cài đặt Android Studio, yêu cầu máy tính đã được cài đặt Java Platform.

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

- Cài đặt

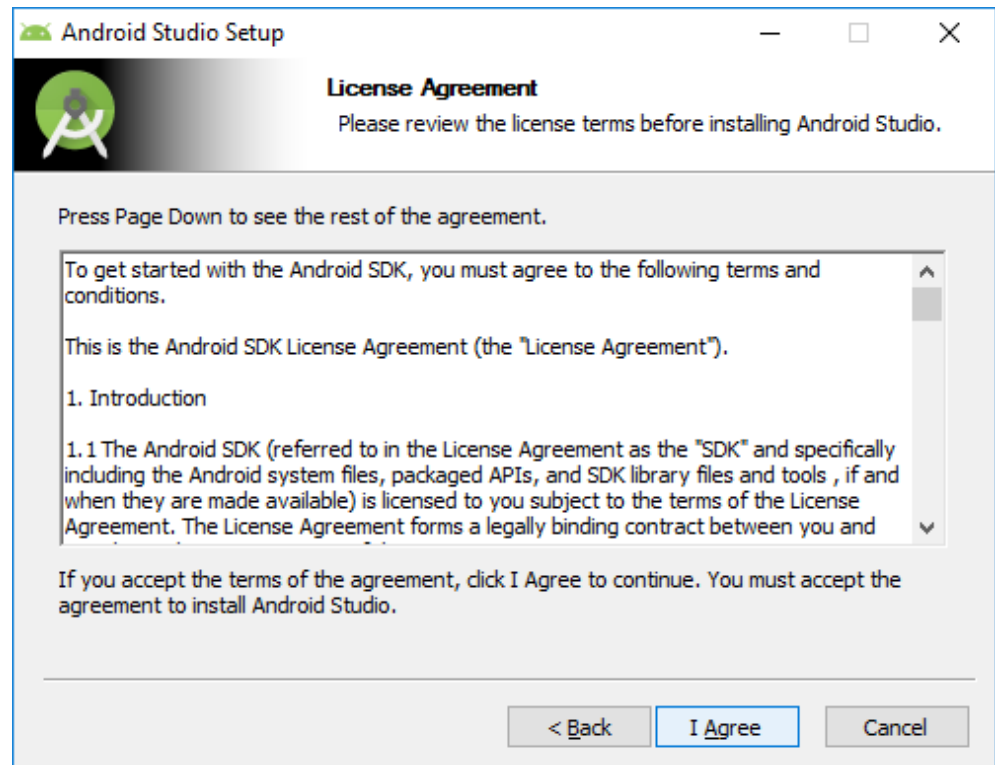


- Các Option:

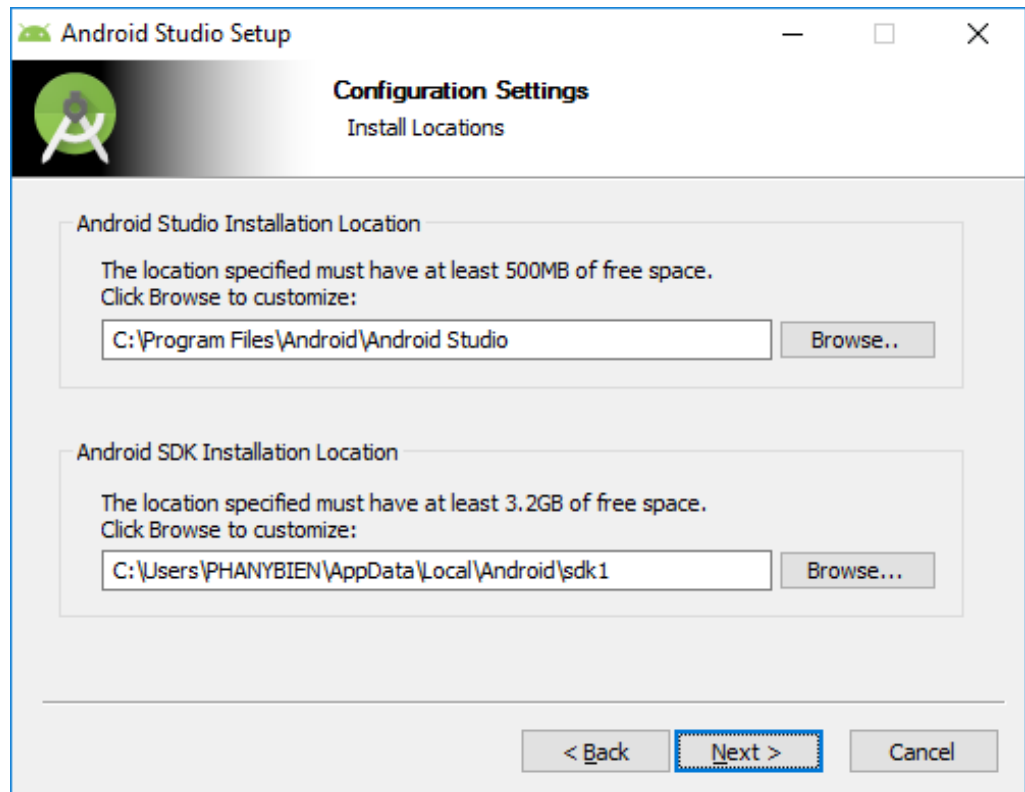
- + **Android SDK** là một tập hợp các công cụ được sử dụng để phát triển ứng dụng cho Android. Android SDK bao gồm:
  - Các thư viện đòi hỏi
  - Bộ dò lỗi (Debugger)
  - Thiết bị mô phỏng (emulator)
  - Các tài liệu liên quan cho Android API.
  - Các đoạn code mẫu.
  - Các hướng dẫn cho hệ điều hành Android.
- + **Android Virtual Device (AVD)** là một cấu hình thiết bị, nó chạy với bộ giả lập Android (Android emulator). Nó làm việc với bộ giả lập để cung cấp một môi trường thiết bị ảo cụ thể, để cài đặt và chạy ứng dụng Android.



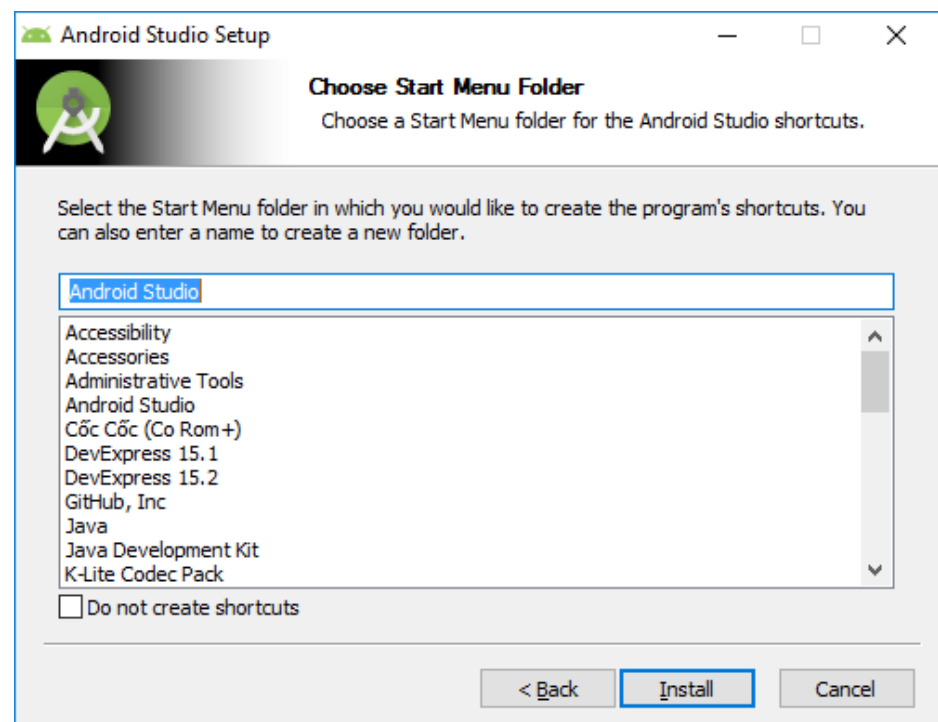
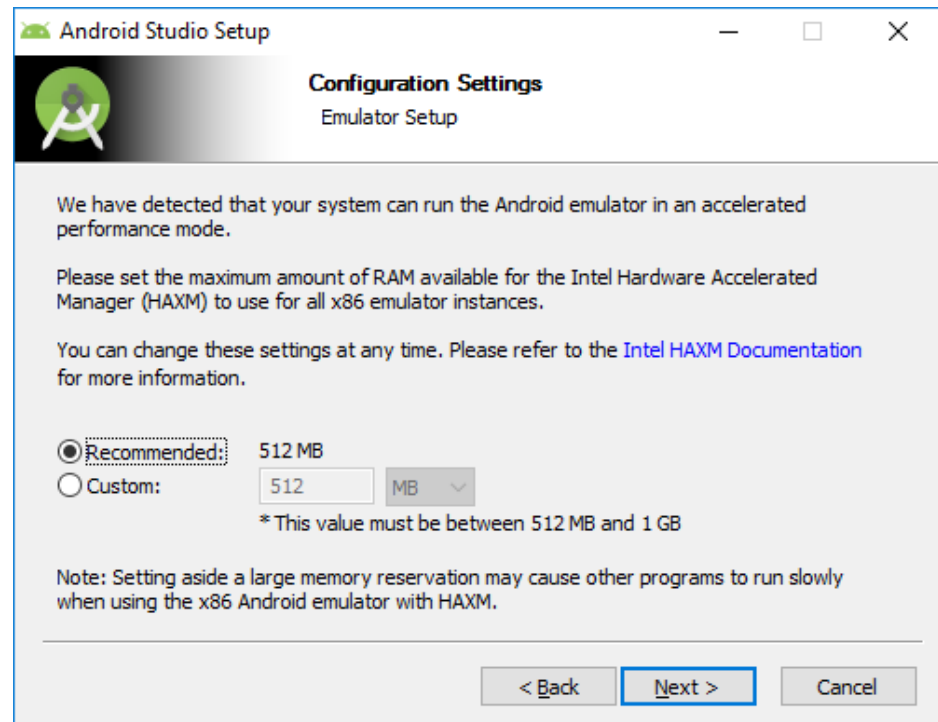
- + **Intel Hardware Accelerated Execution Manager (Intel® HAXM)** là một phần cứng hỗ trợ ảo hóa (hypervisor) có sử dụng công nghệ Intel Virtualization Technology (Intel® VT) để tăng tốc độ ứng dụng Android trên máy chạy phần mềm giả lập Android.

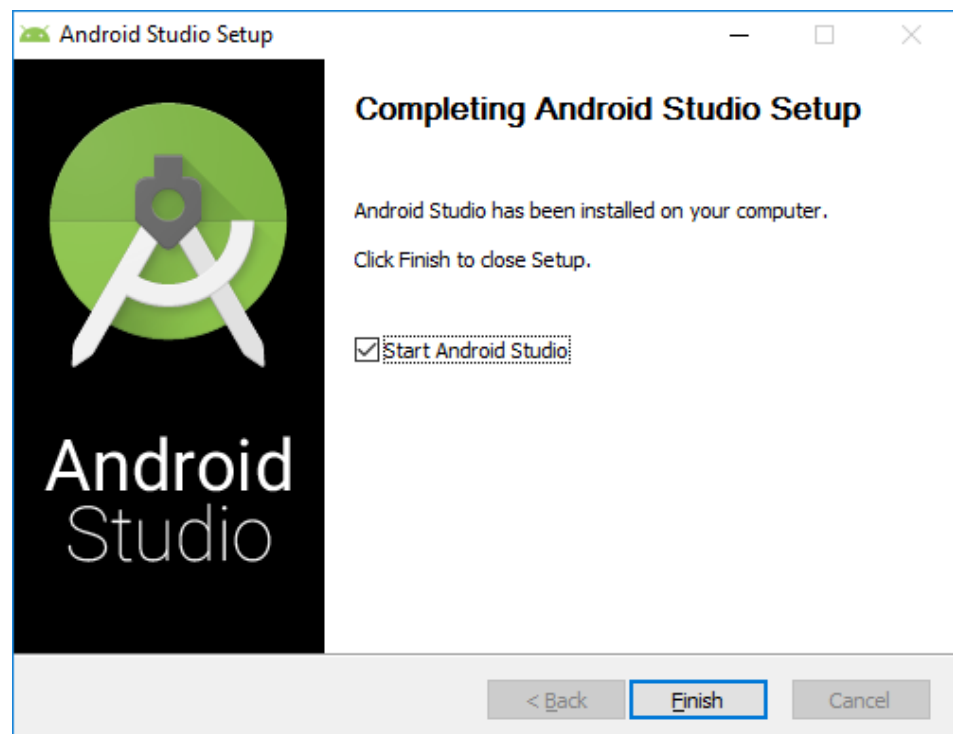
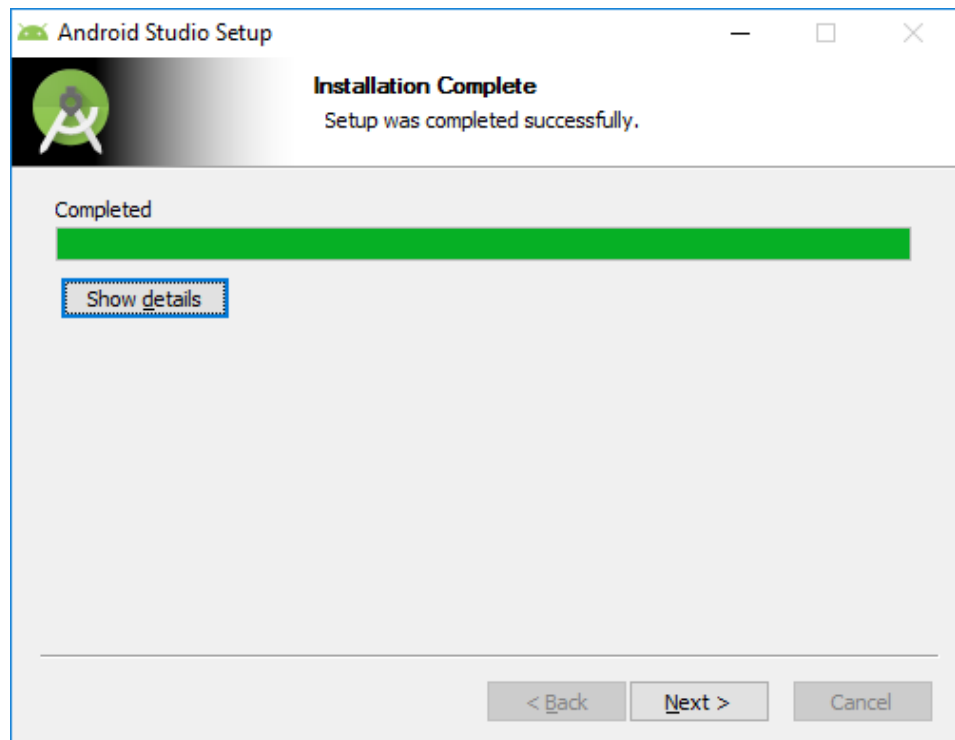


- Chọn thư mục cài đặt



- Nếu máy tính của bạn được trang bị phần cứng tốt, bộ giả lập Android (Android Emulator) có thể chạy được trong chế độ tăng tốc (Accelerated performance mode). Bạn có thể cấu hình chỉ định số lượng RAM tối đa dành cho bộ quản lý tăng tốc phần cứng (Intel Hardware Accelerated Manager - HAXM)

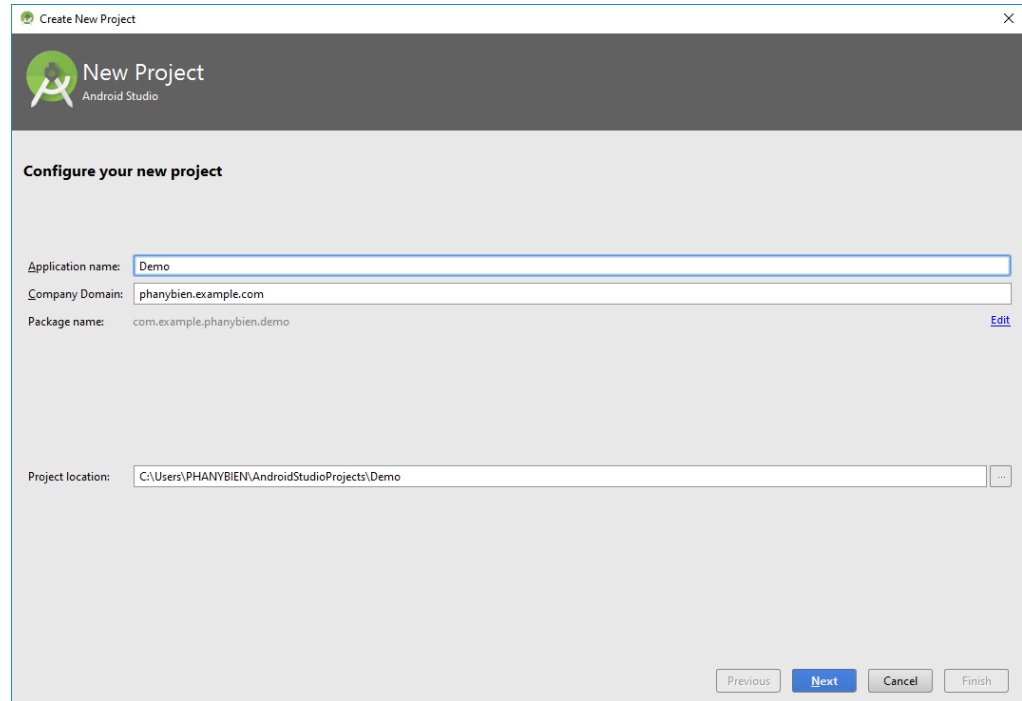




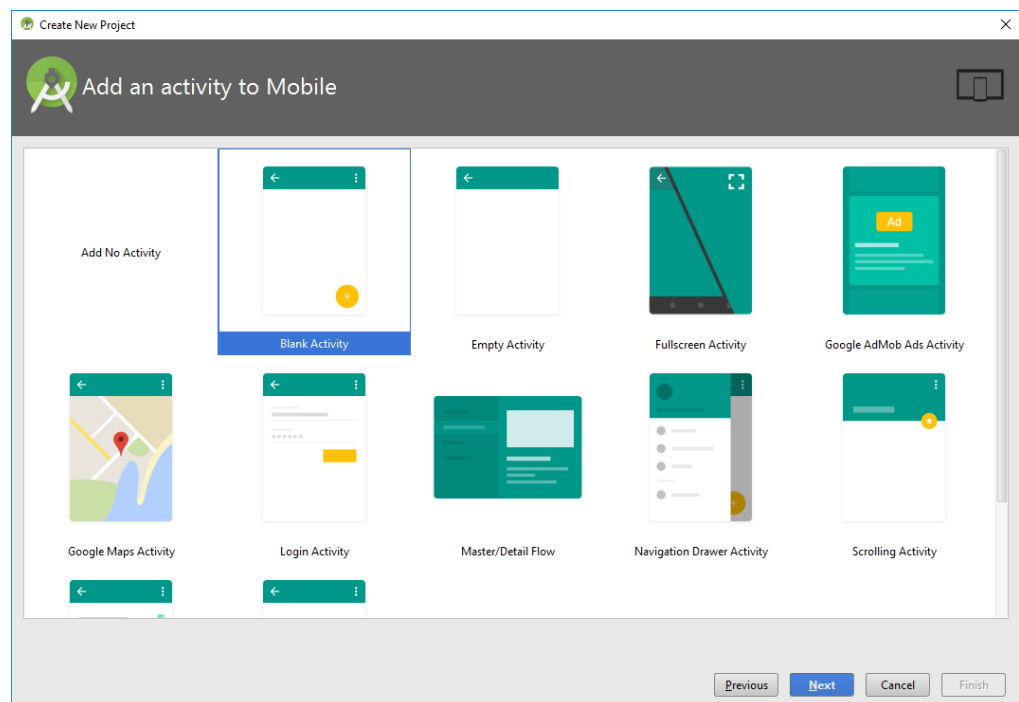
- Cài đặt thành công Android Studio.

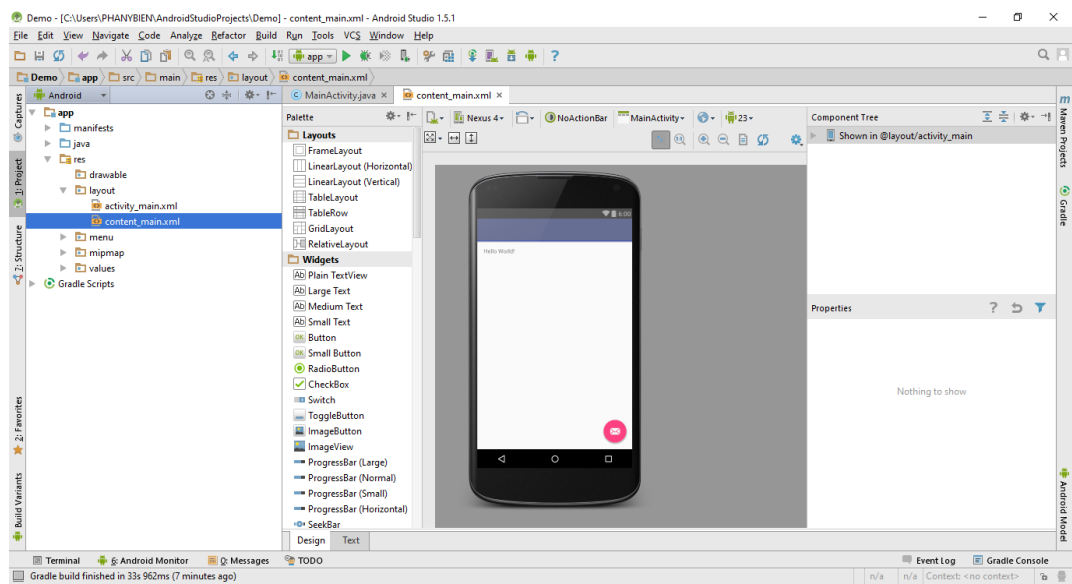
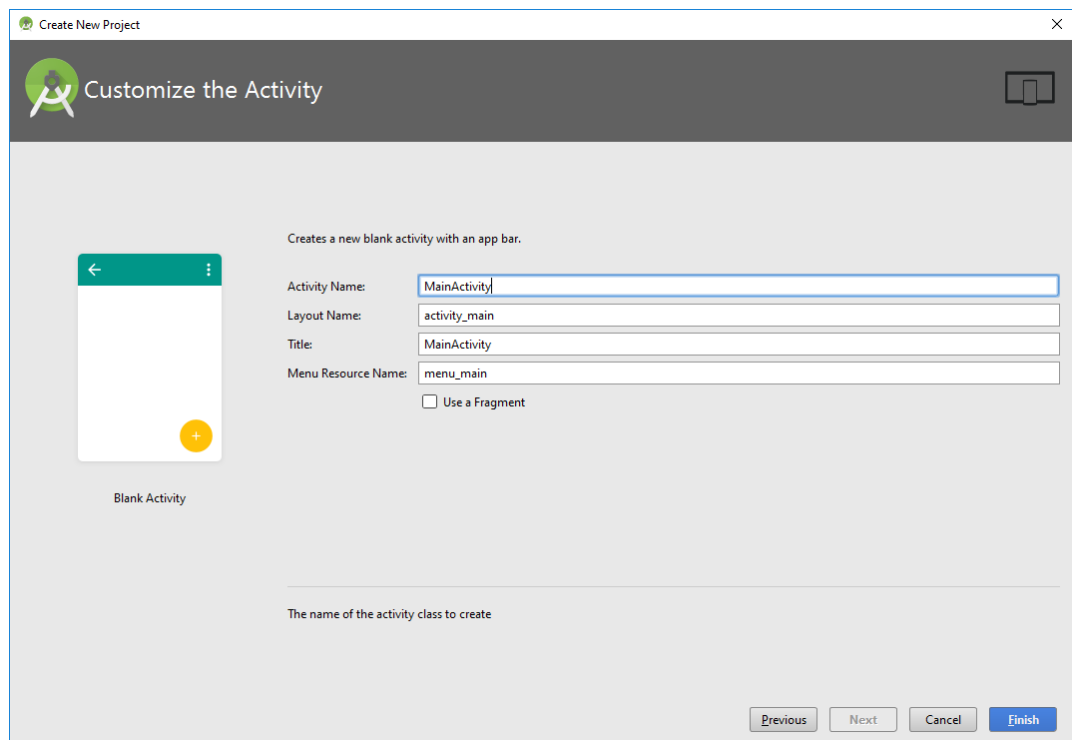
### II.2.1.3. Tạo mới

- File → New → New Project...



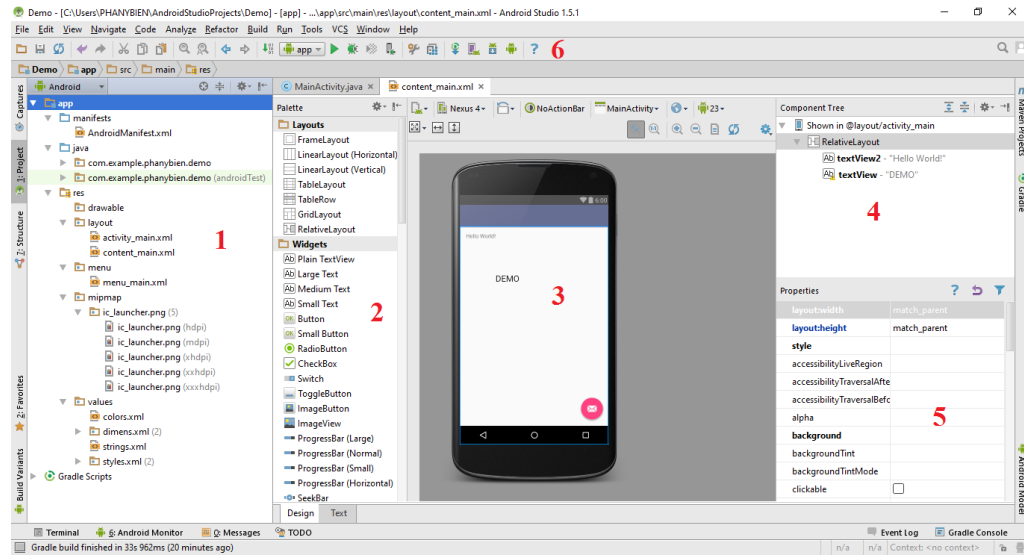
- Chọn Blank Activity.





- Project mới đã được tạo

## II.2.2. Cấu trúc ứng dụng Android



Một Project được chia làm các khu vực như trên hình.

- Khu vực 1: Nơi cấu trúc hệ thống thông tin của Ứng dụng, Ta có thể thay đổi cấu trúc hiển thị

Thư mục	Mô tả
AndroidManifest.xml	Đây là file manifest mô tả các đặc điểm cơ bản của ứng dụng và xác định từng thành phần của nó.
java	Thư mục này có chứa các file nguồn java cho dự án của bạn. Theo mặc định, nó bao gồm một tập tin nguồn MainActivity.java một lớp hoạt động (activity) chạy khi ứng dụng của bạn được khởi động.
res/drawable	Các phiên bản Android trước đây sử dụng thư mục này để chứa ảnh, các phiên bản hiện tại sử dụng thư mục mipmap thay thế làm nơi chứa ảnh. Thư mục này gần như không còn sử dụng.
res/layout	Thư mục này chứa các file định nghĩa giao diện người dùng.

res/menu	Thư mục này chứa các file xml, định nghĩa các menu sẽ hiển thị trên Action Bar.
res/mipmap	Chứa các ảnh 'mipmap'.
res/values	Đây là một thư mục cho các tập tin XML khác nhau có chứa một tập hợp các nguồn, chẳng hạn như các chuỗi và các định nghĩa màu sắc.

- Khu vực 2: Nơi hiển thị các Control mà Android hỗ trợ, cho phép bạn kéo thả trực tiếp vào Giao Diện Thiết Bị để thiết kế.
- Khu vực 3: Vùng giao diện thiết bị, cho phép các Control kéo thả vào đây và đồng thời cho chúng ta hiểu chính control. Vùng giao diện thiết bị có thể hiển thị theo nằm ngang nằm đứng, phóng to thu nhỏ, căn chỉnh control, lựa chọn loại thiết bị hiển thị....
- Khu vực 4: Vùng cho phép hiển thị giao diện theo dạng cấu trúc Cây, dễ dàng quan sát và lựa chọn control khi chúng bị chồng lặp trên giao diện
- Khu vực 5: Vùng cho phép thiết lập trạng thái hay thuộc tính cho các Control trên giao diện.
- Khu vực 6: Thanh công cụ giúp thực hiện các thao tác một cách nhanh chóng.

### II.2.3. Chu kỳ ứng dụng Android

#### II.2.3.1. Chu kỳ sống thành phần

Các thành phần ứng dụng có một chu kỳ sống, tức là mỗi thành phần từ lúc bắt đầu khởi tạo và đến thời điểm kết thúc. Giữa đó, đôi lúc chúng có thể là active hoặc inactive, hoặc là trong trường hợp activities nó có thể visible hoặc invisible.

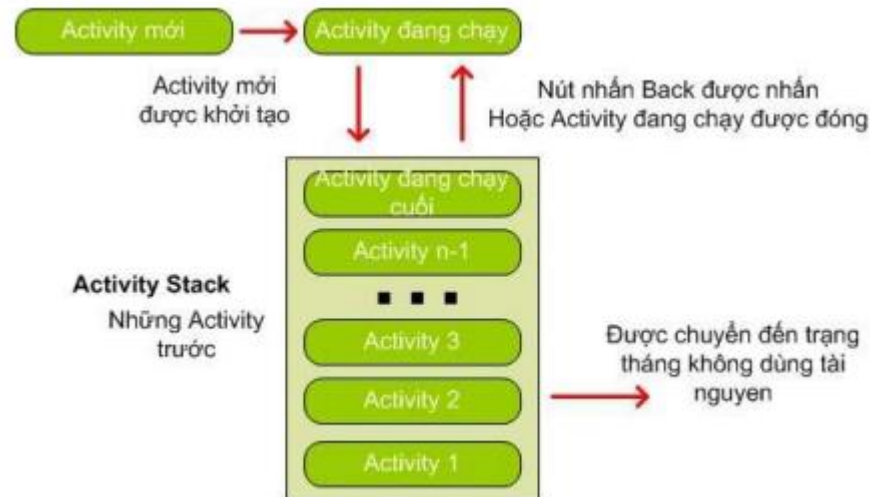
#### II.2.3.2. Activity Stack

Bên trong hệ thống các activity được quản lý như một activity stack. Khi một activity mới được start, nó được đặt ở đỉnh của stack và trở thành



activity đang chạy, activity trước sẽ ở bên dưới activity mới và sẽ không thấy trong suốt quá trình activity mới tồn tại.

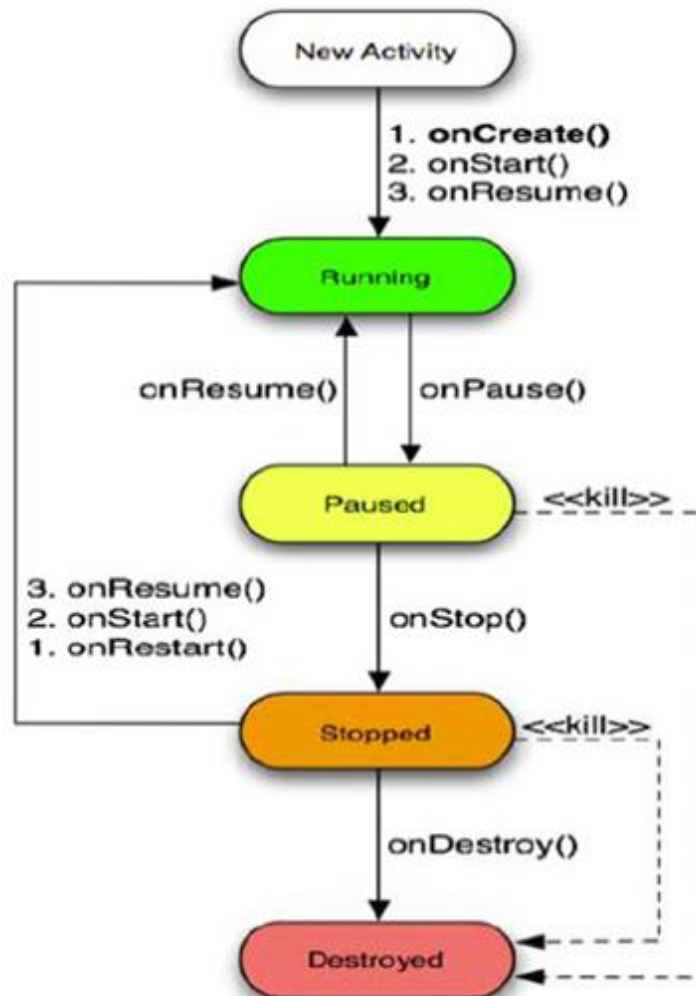
Nếu người dùng nhấn nút Back thì activity kết tiếp của stack sẽ di chuyển lên và trở thành active.



### II.2.3.3. Chu kỳ sống của ứng dụng

Trong một ứng dụng Android có chứa nhiều thành phần và mỗi thành phần và mỗi thành phần đều có một chu trình sống riêng. Và ứng dụng chỉ được gọi là kết thúc khi tất cả các thành phần trong ứng dụng kết thúc. Activity là một thành phần cho phép người dùng giao tiếp với ứng dụng. Tuy nhiên, khi tất cả các Activity kết thúc và người dùng không còn giao tiếp được với ứng dụng nữa nhưng không có nghĩa là ứng dụng đã kết thúc. Bởi vì ngoài Activity là thành phần có khả năng tương tác người dùng thì còn có các thành phần không có khả năng tương tác với người dùng như là Service, Broadcast receiver. Có nghĩa là những thành phần không tương tác người dùng có thể chạy background dưới sự giám sát của hệ điều hành cho đến khi người dùng tự tắt chúng.

#### II.2.3.4. Các trạng thái của chu kỳ sống



Một Activity chủ yếu có 3 chu kỳ chính sau:

- **Active hoặc Running:** Khi Activity được chạy trên màn hình. Activity này tập trung vào những thao tác của người dùng trên ứng dụng.
- **Paused:** Activity tạm dừng khi mất focus nhưng người dùng vẫn trông thấy. Có nghĩa là một Activity mới ở trên nó nhưng không bao phủ đầy màn hình. Một Activity tạm dừng vẫn còn hoạt động nhưng có thể bị kết thúc bởi hệ thống trong trường hợp thiếu vùng nhớ.
- **Stopped:** Nếu nó hoàn toàn bao phủ bởi Activity khác. Nó vẫn còn trạng thái và thông tin thành viên trong nó. Người dùng không thấy nó và

thường bị loại bỏ trong trường hợp hệ thống cần vùng nhớ cho tác vụ khác.

#### **II.2.3.5. Các sự kiện trong chu kỳ sống**

Nếu một Activity được tạm dừng hoặc dừng hẳn, hệ thống có thể bỏ thông tin khác của nó từ vùng nhớ bởi việc finish (gọi hàm finish()), hoặc đơn giản hủy tiến trình của nó. Khi nó được hiển thị lần nữa với người dùng, nó phải được hoàn toàn restart và phục hồi lại trạng thái trước. Khi một Activity chuyển qua chuyển lại giữa các trạng thái, nó phải báo việc chuyển của nó bằng việc gọi hàm transition.

Tất cả các phương thức là những móc nối mà bạn có thể override để tương thích công việc trong ứng dụng khi thay đổi trạng thái. Tất cả các Activity bắt buộc phải có onCreate() để khởi tạo ứng dụng. Nhiều Activity cũng sẽ thực hiện onPause() để xác nhận việc thay đổi dữ liệu và mặt khác chuẩn bị dừng hoạt động với người dùng.

#### **II.2.3.6. Các phương thức của chu kỳ sống**

- onCreate():
  - Được gọi khi Activity lần đầu tiên được tạo.
  - Ở đây bạn thực hiện tất cả các cài đặt tĩnh – tạo các view, kết nối dữ liệu đến list,...
  - Phương thức này gửi qua một đối tượng Bundle chứa đựng từ trạng thái trước của Activity.
  - Luôn theo sau bởi onStart().
- onRestart():
  - Được gọi sau khi Activity đã được dừng, chỉ một khoảng đang khởi động lần nữa.
  - Luôn theo sau bởi onStart().
- onStart():

- Được gọi trước khi một Activity hiển thị với người dùng.
- Theo sau bởi onResume() nếu Activity đến trạng thái foreground hoặc onStop() nếu nó trở nên ẩn.
- onResume():
  - Được gọi trước khi Activity bắt đầu tương tác với người dùng.
  - Tại thời điểm này Activity ở trên đỉnh của stack Activity.
  - Luôn theo sau bởi onPause().
- onPause():
  - Được gọi khi hệ thống đang resuming Activity khác.
  - Phương thức này giữ lại việc không thay đổi dữ liệu.
  - Nó nên được diễn ra một cách nhanh chóng bởi vì Activity kế tiếp sẽ không được resumed ngay cho đến khi nó trở lại.
  - Theo sau bởi onResume() nếu Activity trở về từ ở trước, hoặc bởi onStop() nếu nó được hiển thị với người dùng.
  - Trạng thái của Activity có thể bị hủy bởi hệ thống.
- onStop():
  - Được gọi khi Activity không thuộc tầm nhìn của người dùng.
  - Nó có thể diễn ra bởi vì nó đang bị hủy, hoặc bởi vì Activity khác vừa được resumed và bao phủ nó.
  - Được theo sau bởi onRestart() nếu Activity đang ở lại để tương tác với người dùng, hoặc onDestroy() nếu Activity đang bị bỏ.
  - Trạng thái của Activity có thể bị hủy bởi hệ thống.
- onDestroy():
  - Được gọi trước khi Activity bị hủy.
  - Đó là lần gọi cuối cùng mà Activity này được nhận.
  - Nó được gọi khi Activity đang hoàn thành, hoặc bởi vì hệ thống tạm thời bị hủy để tiết kiệm vùng nhớ.
  - Trạng thái của Activity có thể bị hủy bởi hệ thống.

### **II.2.3.7. Thời gian sống của ứng dụng**

Bảy phương thức chuyển tiếp định nghĩa trong chu kỳ sống của một Activity. Thời gian sống của một Activity diễn ra giữa lần đầu tiên gọi onCreate() đến trạng thái cuối cùng gọi onDestroy(). Một Activity khởi tạo toàn bộ trạng thái toàn cục trong onCreate(), và giải phóng các tài nguyên đang tồn tại trong onDestroy().

### **II.2.3.8. Thời gian hiển thị của Activity**

Visible lifetime của một Activity diễn ra giữa lần gọi một onStart() cho đến khi gọi onStop(). Trong suốt khoảng thời gian này người dùng có thể thấy Activity trên màn hình, có nghĩa là nó không bị foreground hoặc đang tương tác với người dùng. Giữa 2 phương thức người dùng có thể duy trì tài nguyên để hiển thị Activity đến người dùng.

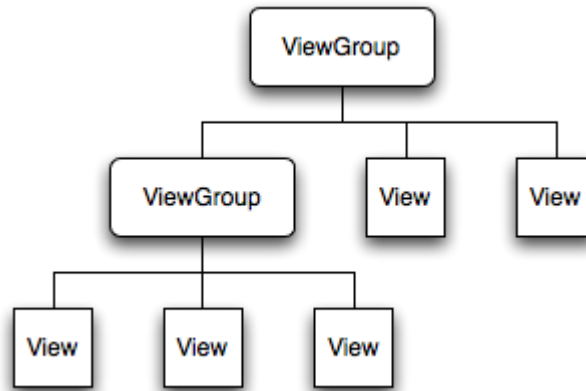
## **II.2.4. Các thành phần giao diện trong Android**

### **II.2.4.1. View**

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

Tất cả mọi widget đều có chung các thuộc tính cơ bản như là cách trình bày vị trí, 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 phân cấp như hình dưới. Một screen là một tập hợp các Layout và các widget được bố trí có thứ tự. Để thể hiện một screen thì trong hàm onCreate() của mỗi Activity cần phải được gọi một hàm là setContentView(R.layout.main), hàm này sẽ load giao diện từ file XML lên để phân tích thành mã bytecode.



#### II.2.4.2. ViewGroup

ViewGroup thực ra chính là View hay nói đúng hơn thì ViewGroup chính là các widget Layout được dùng để bố trí các đối tượng khác nhau trong một screen. Có một số loại ViewGroup như sau:

- **LinearLayout:**

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

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

- **FrameLayout:**

FrameLayout được dùng để bố trí các đối tượng theo kiểu giống như là các Layer trong Photoshop. Những đối tượng nào thuộc Layer bên dưới thì sẽ bị che khuất bởi các đối tượng thuộc Layer nằm trên. FrameLayer thường được sử dụng khi muốn tạo ra các đối tượng có khung hình bên ngoài chẳng hạn như contact image button.

- **AbsoluteLayout:**

Layout này được sử dụng để bố trí các widget vào một vị trí bất kì trong layout dựa vào 2 thuộc tính tọa độ x,y. Tuy nhiên, kiểu layout này rất ít khi được dùng bởi vì tọa độ của các đối tượng luôn cố định và sẽ không tự điều chỉnh được tỷ lệ khoảng cách giữa các đối tượng. Khi chuyển ứng dụng sang một màn hình có kích thước với màn hình thiết kế ban đầu thì vị trí của các đối tượng sẽ không còn được chính xác như ban đầu.

- **RelativeLayout:**

Layout này cho phép bố trí các widget theo một trục đối xứng ngang hoặc dọc. Để đặt đúng vị trí thì các widget cần được xác định một mối ràng buộc nào đó với các widget khác. Các ràng buộc này là các ràng buộc trái, phải, trên, dưới so với một widget hoặc so với layout parent. Dựa vào những mối ràng buộc đó mà RelativeLayout cũng không phụ thuộc vào kích thước của screen thiết bị. Ngoài ra, nó còn có ưu điểm là giúp tiết kiệm layout sử dụng nhằm mục đích giảm lượng tài nguyên sử dụng khi load đồng thời đẩy nhanh quá trình xử lý.

- **TableLayout:**

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

### **II.2.4.3. Các thành phần khác**

- **Button:**

Button là một widget thường được sử dụng nhiều nhất trong hầu hết các ứng dụng Android. Button là một nút được dùng để xử lý các xử kiện khi nhấn vào.

- **ImageButton:**

Tương tự như Button và có thêm một thuộc tính android:src để thêm hình ảnh vào và không có thẻ text.

- **ImageView:**

Được dùng để thể hiện một hình ảnh. Nó gần giống như ImageButton, chỉ khác không có hình dạng như một button.

- **ListView:**

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

Để thể hiện được một list thông tin lên một màn hình cần phải có 3 yếu tố chính:

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

- **TextView**

TextView ngoài tác dụng là để hiển thị văn bản, nó còn cho phép định dạng nội dung bằng thẻ html.

Nội dung TextView cũng có thể được định dạng bằng thẻ html ngay trong XML.

- **EditText**

Trong Android, đối tượng EditText được sử dụng như một TextField hoặc một TextBox dựa vào thuộc tính android:singleLine (android:singleLine = “true” EditText của bạn sẽ trở thành một TextField, ngược lại sẽ là TextBox).



- CheckBox

Nhận 2 giá trị true hoặc false. Đối tượng CheckBox cho phép chọn nhiều item cùng một lúc.

- MenuOptions

Hiển thị danh sách menu.

- ContextMenu

ContextMenu được sử dụng để hiển thị các tùy chọn khi người dùng nhấn dài vào một dòng nào đó trong ListView.

- Quick Search Box

Một trong những tính năng mới trong phiên bản Android 1.6 là Quick Search Box. Đây là khuôn khổ tìm kiếm mới trên hệ thống Android, điều này làm cho người dùng có thể nhanh chóng tìm kiếm bất cứ thứ gì có trên chiếc điện thoại Android của họ và cả các tài nguyên trên web khi họ đang online. Nó tìm kiếm và hiển thị kết quả tìm kiếm ngay khi bạn đang gõ. Nó cũng cung cấp các kết quả từ các gợi ý tìm kiếm web, danh sách doanh nghiệp địa phương, và thông tin khác từ Google, chẳng hạn như báo giá cổ phiếu, thời tiết, và tình trạng chuyến bay, thời tiết, và tình trạng chuyến bay. Tất cả điều này có sẵn ngay từ màn hình chủ, bằng cách khai thác trên Quick Search Box .

- Activity & Intent

- Activity:

Activity là một thành phần chính của một ứng dụng Android, được dùng để hiển thị một màn hình và nắm bắt các hoạt động xảy ra trên màn hình đó. Khi làm việc với Activity cần nắm bắt được một số kiến thức cơ bản sau:

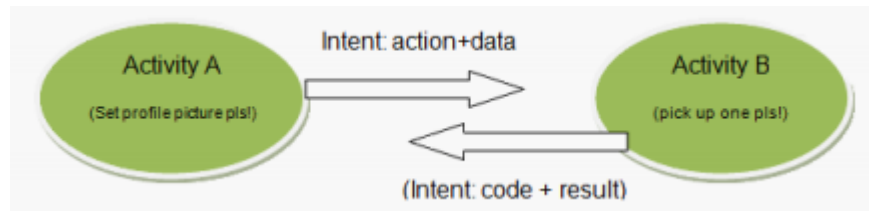
- + Chu kỳ sống của một Activity
- + Tạo menu và dialog
- + Khởi động một Activity

- Intent:

+ Khái niệm:

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

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



+ Dữ liệu của Intent:

Intent về cơ bản là một cấu trúc dữ liệu, được mô tả trong lớp `android.content.Intent`

Các thuộc tính của một đối tượng Intent:

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

+ Các Action được định nghĩa sẵn:

Dưới đây là những hằng String được định nghĩa sẵn trong lớp Intent. Đi kèm với nó là các Activity hay Application được xây dựng

sẵn sẽ được triệu gọi mỗi khi Intent tương ứng được gửi (tất nhiên khi được cung cấp đúng data).

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

## II.3. Biểu đồ trong Android

### II.3.1. Tổng quan

Trong mỗi ứng dụng có rất nhiều giá trị, và chúng ta phải đánh giá xem kết quả các giá trị đó khác nhau, chênh lệch, thay đổi như thế nào. Và cách đem lại hiểu quả tốt nhất là thể hiện nó qua biểu đồ.

Biểu đồ là hình vẽ biểu diễn các số liệu, thường là các số liệu địa lý, dùng để so sánh, nhận ra sự khác biệt hoặc quá trình tăng trưởng.

Một biểu đồ thể hiện giá trị là một chuỗi những điểm giá trị trên một khung thời gian nhất định.

Các loại biểu đồ:

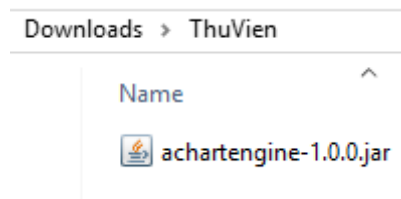
- Biểu đồ cột
- Biểu đồ đồ thị
- Biểu đồ tròn
- Biểu đồ miền

### II.3.2. Thư viện AchartEngine

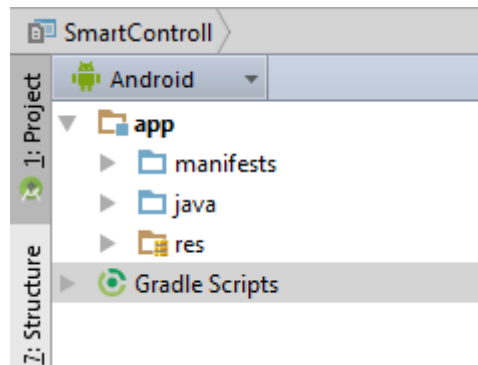
Android có nhiều thư viện mã nguồn mở trong đó AChart "achartengine-1.0.0" là một thư viện hỗ trợ cho việc xây dựng biểu đồ.

- Các bước cài đặt thư viện vào trong dự án Android:
  - Tải thư viện về theo đường link:

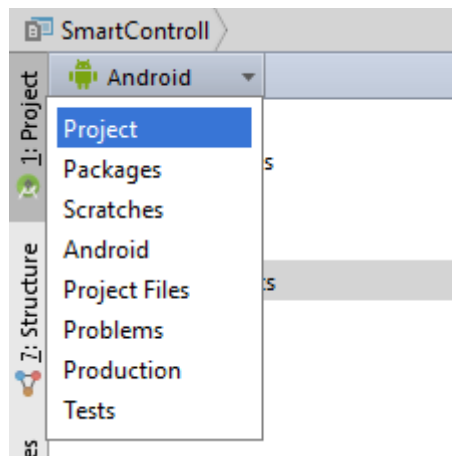
<http://www.mediafire.com/download/99ojp0qrhd3coqb/ThuVien.rar>



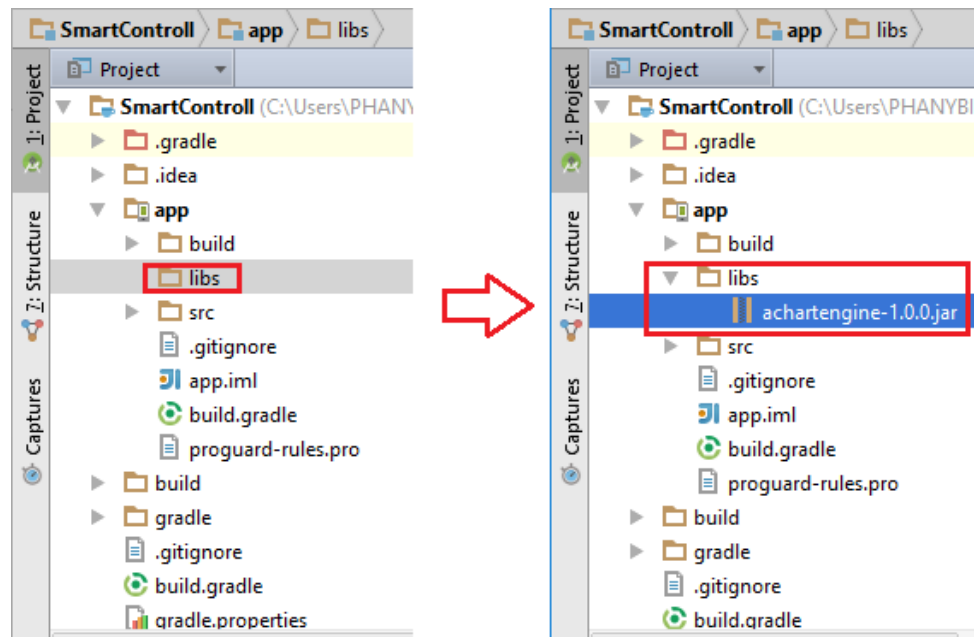
- Mở Project cần tạo biểu đồ.



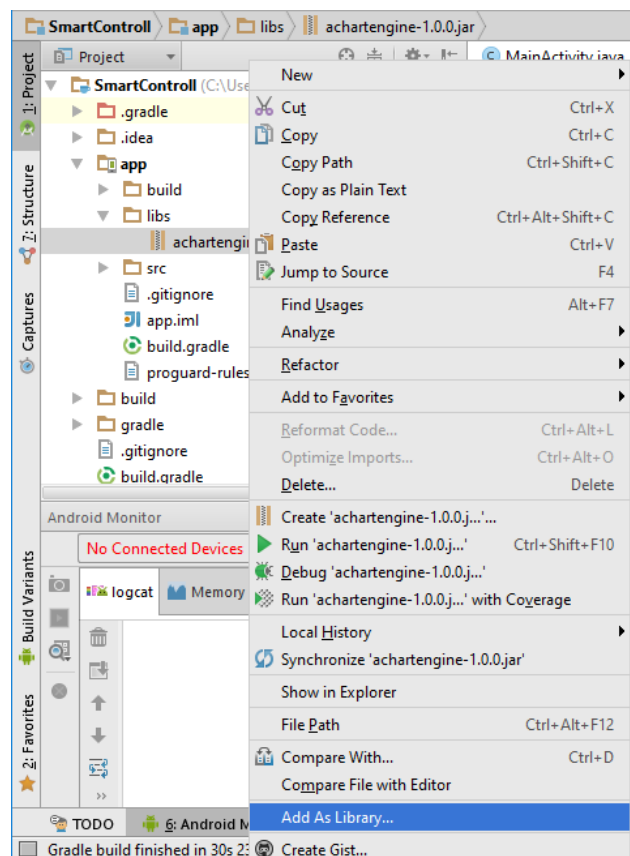
- Chuyển sang tab Project.



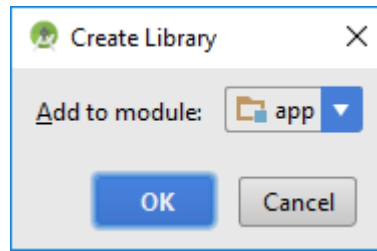
- Copy file achartengine-1.0.0.jar và paste vào thư mục **libs**.



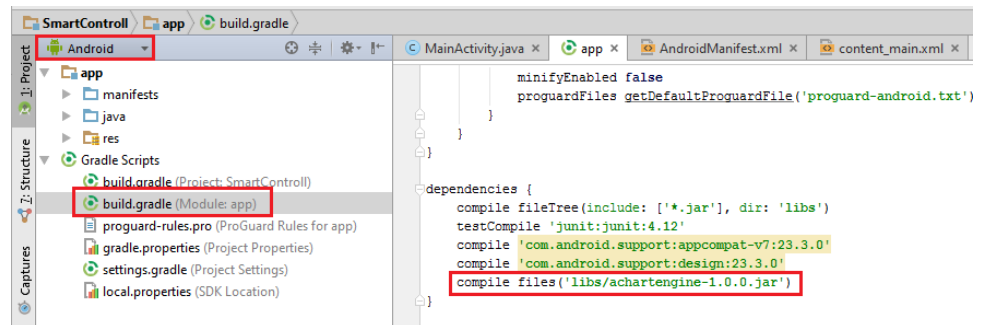
- Nhấn phải chuột vào file .jar, chọn **Add as Library...**



- Nhấn OK.



- Trở lại Tab Android, bạn có thể thấy thư viện của bạn đã được khai báo trong **build.gradle (Module: app)**.



- Lúc này đã có thể sử dụng thư viện để xây dựng biểu đồ.

```
import org.achartengine.ChartFactory;
import org.achartengine.GraphicalView;
import org.achartengine.chart.PointStyle;
import org.achartengine.model.SeriesSelection;
import org.achartengine.model.TimeSeries;
import org.achartengine.model.XYMultipleSeriesDataset;
import org.achartengine.renderer.XYMultipleSeriesRenderer;
import org.achartengine.renderer.XYSeriesRenderer;
```

## **CHƯƠNG III: TÌM HIỂU CÔNG NGHỆ HỒNG NGOẠI IR**

### **III.1. Giới thiệu**

Consumer IR hoặc CIR, đề cập đến một loạt các thiết bị sử dụng các tia hồng ngoại quang phổ điện từ để truyền tin hiệu không dây. Thường được tìm thấy trong truyền hình điều khiển từ xa, cổng hồng ngoại là như nhau trong mọi thiết bị điện tử, chẳng hạn như máy PDA, máy tính xách tay, và máy tính. Các chức năng của CIR là chuyển tải dòng lệnh để các thiết bị điện tử thực hiện nó. Ví dụ, một điều khiển ti vi từ xa có thể chuyển tải một "kênh lên" để ra lệnh cho ti vi chuyển đến kênh tiếp theo, trong khi một máy tính có thể có thể lướt Internet chỉ qua CIR. Cách phân loại, tốc độ, băng thông và sức mạnh của tín hiệu thông tin được truyền đi phụ thuộc đặc biệt vào giao thức CIR sử dụng.

CIR là loại phổ biến nhất của truyền tín hiệu quang học trong không gian tự do.

### **III.2. Mô tả giao thức hoạt động**

Kể từ khi các giao thức Consumer IR được sử dụng cho hầu hết cho các phần không được chuẩn hóa, máy tính và điều khiển từ xa thường ghi nhớ một dòng bit, có thể nén, không xác định tỉ lệ bit thực tế, và thực hiện nó trở lại. Sự giống nhau giữa điều khiển từ xa thường phần lớn là sự lựa chọn kết quả ngẫu nhiên của chip mã hóa / giải mã hồng ngoại (mặc dù hiện nay vi điều khiển cũng được sử dụng) và IR module nhận hoặc sự bất chước của các chip cũ hơn là do khi thiết kế. Các nhà sản xuất thiết bị tiêu dùng thường xuyên tái sử dụng cùng một giao thức trên nhiều thiết bị tương tự, mặc dù đối với mỗi nhà sản xuất và loại thiết bị có nhiều giao thức thường được sử dụng.

Với sự sẵn sàng của các chip vi điều khiển rẻ tiền, nhiều điều khiển từ xa có thể dựa trên các chip như hiện nay chứ không phải chip mã hóa điều khiển từ xa chuyên dụng. Điều này làm cho nó dễ dàng hơn để giữ các mã tương tự khi di chuyển các nút trên điều khiển từ xa.

Ngoài ra, các chức năng giải mã sẽ thường được tích hợp vào một vi điều khiển để điều khiển phức tạp hơn các thiết bị A/V, loại bỏ sự cần thiết của các chip riêng biệt. Trong trường hợp không có một tiêu chuẩn khả thi, các vi điều khiển có thể được sử dụng để mô phỏng các giao thức mở hồ được sử dụng bởi các chip chuyên dụng mã hóa / giải mã cũ và nó xuất hiện rằng đây là trường hợp thường. Thậm chí việc phân tích 4-bit vi điều khiển được thiết kế chỉ để sử dụng điều khiển từ xa (như  $\mu$ PD612xA NEC (ngưng sử dụng),  $\mu$ PD613x,  $\mu$ PD1724x,  $\mu$ PD6x, và  $\mu$ PD17932x (8-bit) dùng trong gia đình). Việc phát triển phục vụ việc sử dụng bàn phím không dây, chế độ chờ với điện năng thấp, và mã điều khiển mẫu mặc dù các tính năng tương tự có mặt trên đa số vi điều khiển PIC hoặc Atmel AVR.

- **Tiêu chuẩn:**

Sony sản xuất một số thiết bị tiêu dùng thuộc loại khác nhau nhưng chia sẻ chung giao thức độc quyền, được gọi là S-link. Một jack cắm trên mỗi thiết bị cho phép các tín hiệu điều khiển từ xa được kết nối với nhau giữa các thiết bị. Các giao thức bao gồm các tính năng hữu ích nhưng bình thường không hỗ trợ nhiều hơn một trong cùng một loại thiết bị (như nhiều lần đổi đĩa CD). Một số thành phần A/V có thể tạo ra mã trạng thái thông tin mà có thể được sử dụng để làm những việc như tự động dừng băng của bạn khi đĩa CD bạn đã được dừng ghi. Phần mềm chạy trên một máy tính với một giao diện phù hợp cũng có thể kiểm soát các thành phần A/V và giám sát hoạt động của họ; Ví dụ, máy tính của bạn có thể cho biết thông tin và ca khúc được chơi trong đĩa CD của bạn và tìm kiếm các tiêu đề tại một trong những cơ sở dữ liệu CD internet. Sony yêu cầu phí 5000 USD để truy cập vào các tài liệu S-Link. Sony sử dụng giao thức SIRC cho điều khiển từ xa. SIRC được phát triển trong ba phiên bản khác nhau: 12 bit, 15 bit và 20 bit.

Các REC-80 và RC-5 mã được phát triển bởi Philips đã tình cờ gọi là tiêu chuẩn quốc tế. Tuy nhiên, giao thức REC-80 là dễ bị can thiệp và đã nhanh chóng được thay thế bằng giao thức RC-5. Mặc dù nó xuất hiện là giao thức độc quyền được phát triển bởi Philips nhưng nó cũng đã được thông qua bởi các nhà sản xuất khác nhau, cụ thể



European- based và US-based. Điều này cho phép khả năng tương tác giữa các thiết bị từ xa của các thương hiệu khác nhau. Các mã RC-5 đã và vẫn còn được sử dụng bởi nhiều người Mỹ và nhà sản xuất châu Âu dựa trên thiết bị A/V. Thật không may, tài liệu của các lệnh chuẩn bị không được phân phối rộng rãi. Vì vậy, có một số thương hiệu của các thiết bị sử dụng lệnh phi tiêu chuẩn, gây nhiễu với các thiết bị khác cũng sử dụng giao thức RC-5.

Lệnh RC-5 thiết lập được xác định vào cuối năm 1980 và mở rộng để tăng số lượng các lệnh trong đầu những năm 1990 (đôi khi được gọi là RC-5x). Tuy nhiên, các yêu cầu phát triển nhanh chóng cho các loại mới hơn của các sản phẩm điện tử kể từ thời điểm đó (ví dụ, đầu DVD, hộp cáp, DVR...) đã dẫn Philips thay thế giao thức RC-5 với các phiên bản mới hơn RC-6, giao thức mà có cả một thiết lập mở rộng của các thiết bị (256 so với 32) và lệnh cho mỗi thiết bị (256 so với 64 trong RC-5 và 128 trong RC-5x). Một lần nữa, thông tin về các giao thức RC-6 là không có sẵn từ Philips.

Ngược lại, người sản xuất điện tử lớn ở Nhật Bản dường như đã thông qua một giao thức được phát triển và quản lý bởi NEC . Trong giao thức NEC, mỗi nhà sản xuất được gán một mã số duy

nhất được chứa trong các lệnh truyền, tránh khả năng kích hoạt sai bằng thiết bị điều khiển từ xa khác.

REC-80 sử dụng điều chế vị trí xung và RC-5 sử dụng hai giai đoạn. Đầu chip được cung cấp bởi Philips Semiconductors cho phép sử dụng dễ dàng của giao thức REC-80 và RC-5. Các SAA3004, SAA3007, và chip encoder SAA3008 sử dụng REC-80, và các chip mã hóa SAA3006 và SAA3010 sử dụng RC-5. Chip SAA3049A giải mã được một trong hai loại giải mã. (Lưu ý rằng sự phân chia Philips Semiconductors là NXP). Tất cả những con chip đã bị ngưng sử dụng. Tuy nhiên, các giao thức truyền tải dễ dàng tạo ra, hay giải mã với mục đích chung bằng vi điều khiển 8-bit, chẳng hạn như những thứ được cung cấp bởi Microchip Technology và Atmel .

Truyền tải các lệnh IR chỉ đòi hỏi một vi điều khiển và một hồng ngoại LED, có sẵn từ nhiều nguồn khác nhau. Tiếp nhận các lệnh điều chế cho các giao thức NEC RC-5, RC-6, và có thể dễ dàng thực hiện với máy thu IR chuyên dụng, có sẵn nhất từ Sharp và Vishay Intertechnology. Những thiết bị nhận bao gồm một hình ảnh diode, một điều khiển tự động mạch (AGC), và một bộ giải điều chế. Sau đó các tín hiệu giải điều chế được giải mã với một vi điều khiển.

CEA-931-B định nghĩa một phương pháp đóng gói mã điều khiển từ xa qua IP và CEA-931-A định nghĩa một phương pháp đóng gói mã điều khiển từ xa qua IEEE-1394.

- **Hạn chế giao thức:**

Sự thiếu chuẩn hóa tạo ra nhiều vấn đề cho người sử dụng: cần mua điều khiển từ xa đa năng bởi vì ban đầu không thể kiểm soát các chức năng liên quan trên các thiết bị kết nối với nhau và nâng cấp chúng khi mua một thiết bị mới, điều khiển từ xa đa năng không kiểm soát đầy đủ các thiết bị, không có khả năng kiểm soát nhiều hơn một trong những đơn vị cùng loại, không có khả năng trong hầu hết các thiết lập của người tiêu dùng để dừng ghi âm băng khi CD kết thúc, sự bất lực của VCR để kiểm soát các kênh truyền hình cáp, và những thứ khác. Điều này đồng nghĩa việc có 5 + điều khiển từ xa khác nhau trên bàn cà phê.

- **Thông tin kỹ thuật:**

Bước sóng hồng ngoại. Khoảng 870 nm và 930-950 nm.

Tần số sóng mang : Thường cố định tần số sóng mang, thường ở khoảng giữa 33 và 40 kHz hoặc 50-60 kHz. Các giao thức thường được sử dụng nhất là giao thức NEC, trong đó xác định một tần số sóng mang của 38 kHz. Các giao thức NEC được sử dụng bởi phần lớn các thiết bị điện tử tiêu dùng do Nhật Bản sản xuất. Giao thức Philips RC-5 và RC-6 định một tần số sóng mang là 36 kHz. Tuy nhiên, RC-5 chip mã hóa đầu chia tần số tổng thể của vi điều khiển 4-bit cho 12. Điều này đòi hỏi một bộ cộng hưởng thạch anh 432 kHz để đạt được một tần số sóng mang 36 kHz. Vì vậy,

hiều công ty sử dụng một bộ cộng hưởng thạch anh 455 kHz, đó là điều bình thường do đó tần số được sử dụng trong các tần số trung gian các giai đoạn phát sóng AM radio, kết quả trong một tần số sóng mang của 37,92 kHz (thực chất là 38 kHz). Ngay cả tài liệu hướng dẫn cho các chip điều khiển riêng của Philips đề nghị một bộ cộng hưởng thạch anh 429 kHz để dễ dàng hơn cho ra một tần số sóng mang của 35,75 kHz. Máy phát hồng ngoại hiện đại thường sử dụng vi điều khiển 8-bit với 4 MHz tần số xung nhịp, cho phép lựa chọn gần như độc đoán của các tần số sóng mang.

Đề án điều chế: thường 100% amplitude-shift keying (ASK). Cũng có thể liên quan đến việc điều chế xung vị trí, mã hóa biphas/manchester, vv của các xung truyền đi. Hầu hết các điều khiển từ xa sử dụng độ dài của không gian giữa các xung để mã hóa dữ liệu.

Tốc độ dữ liệu: thường thấp hơn so với tần số sóng mang đáng kể. Hầu hết các giao thức dường như dao động từ 120 bit/giây và 4 bit/giây. Tốc độ dữ liệu có thể mã hóa bằng một số chương trình mã hóa bit thông thường, thay đổi về thời gian giữa các xung để phân biệt giữa 1 và 0.

Encoding: thay đổi dựa trên các chip bộ mã hóa/giải mã sử dụng. Thường bao gồm một số dự phòng để phát hiện lỗi hoặc sửa chữa. Ví dụ, một số chip NEC gửi mã cùng bốn lần (đảo ngược thời gian thứ hai và thứ tư).

Chìa khóa để lập bản đồ mã: thay đổi từ điều khiển từ xa để điều khiển từ xa.

### III.3. Data Format

#### III.3.1. Giao thức truyền tải Philips RC5

Các giao thức truyền Philips RC5 IR sử dụng mã hóa Manchester của các bit thông điệp. Mỗi xung nhịp có chiều dài là 889us, ở một tần số sóng mang 36kHz (27.7us). Bit logic được truyền như sau:

Bit '0' - một xung nhịp 889us tiếp theo là một khoảng trống 889us, với tổng thời gian truyền của 1.778ms

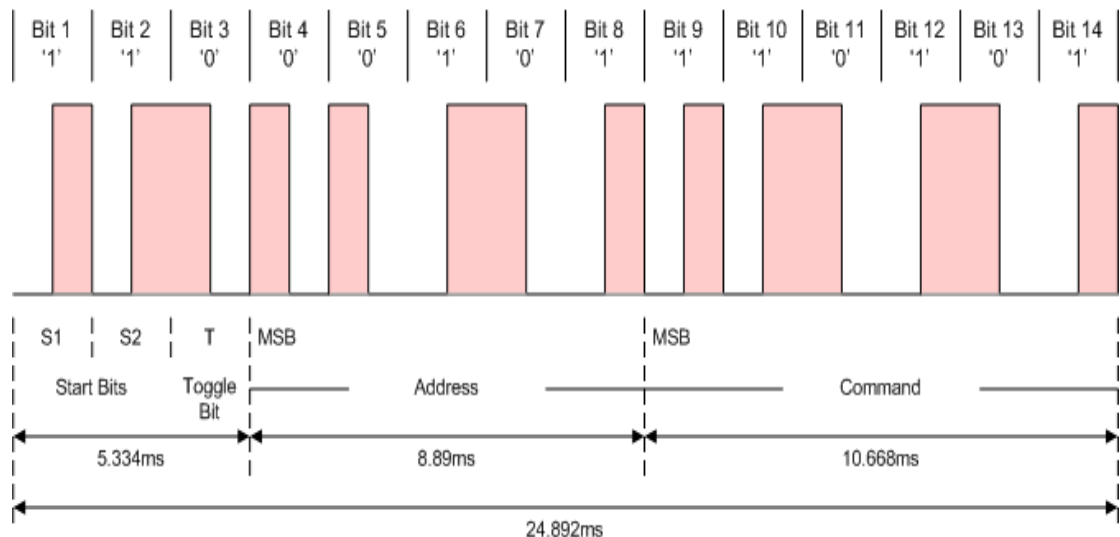
Bit '1' - một khoảng trống 889us tiếp theo là một xung nhịp 889us, với tổng thời gian truyền của 1.778ms

Tỷ lệ xung nhịp/ tạm dừng các tần số sóng mang 36 kHz là 1/3 hoặc 1/4, làm giảm điện năng tiêu thụ.

Khi một phím được nhấn trên bộ điều khiển từ xa, dữ liệu được truyền gồm 14 bit sau, theo thứ tự:

- Hai bit Start (S1 và S2), cả hai đều là bit '1'.
- 1 bit Toggle (T). Bit này được đảo ngược mỗi khi một phím được thả ra và nhấn lại.
- 5-bit địa chỉ cho các thiết bị nhận
- 6-bit dòng lệnh.

Các bit địa chỉ và dòng lệnh sẽ gửi các bit quan trọng nhất. Hình 1 minh họa các định dạng của một khung truyền Philips RC5 IR, cho một địa chỉ 05h (00101b) và một dòng lệnh 35h (110101b).



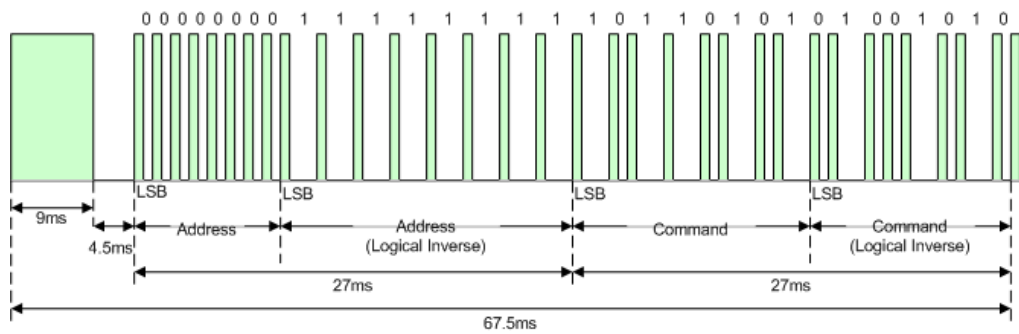
Ví dụ bằng cách sử dụng giao thức truyền tải Philips RC5 IR.

Từ hình 1 ta có thể thấy rằng nó cần:

- 5.334ms để truyền tải các bit bắt đầu (Start) và Toggle (S1, S2 và T). Chú ý rằng, một nửa bit đầu tiên của S1 là một khoảng trống, người nhận sẽ chỉ nhận thấy sự khởi đầu thực sự của dữ liệu truyền tải sau 889us.
- 8.89ms để truyền tải 5 bit cho địa chỉ
- 10.668ms để truyền 6 bit cho dòng lệnh
- 24.892ms để truyền tải đầy đủ các dữ liệu thực tế.

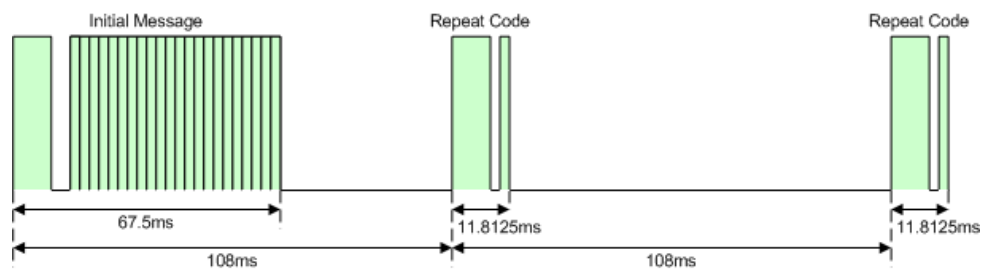
### III.3.2. Giao thức truyền tải NEC

- Các giao thức truyền tải NEC sử dụng mã hóa khoảng cách xung của các bit thông điệp. Mỗi xung nhịp có chiều dài là  $562.5\mu\text{s}$ , ở một tần số sóng mang của 38 kHz ( $26.3\mu\text{s}$ ). Bit logic được truyền như sau:
  - Bit '0' - một xung nhịp  $562.5\mu\text{s}$  theo sau là một khoảng trống  $562.5\mu\text{s}$ , với tổng thời gian truyền của 1.125ms
  - Bit '1' - một xung nhịp  $562.5\mu\text{s}$  xung theo sau là một khoảng trống 1.6875ms, với tổng thời gian truyền của 2.25ms.
- Khi truyền tải hoặc nhận mã điều khiển từ xa bằng cách sử dụng giao thức truyền dẫn NEC IR, WB\_IRRC chạy được tối ưu khi tần số sóng mang (dùng để điều chế / giải điều chế) được thiết lập để 38.222kHz.
- Khi một phím được nhấn trên bộ điều khiển từ xa, thông điệp được truyền gồm những điều sau đây, theo thứ tự:
  - Một xung nhịp đầu 9ms (16 lần độ dài xung nhịp sử dụng cho một bit dữ liệu logic)
  - Một khoảng trống 4.5ms
  - 8-bit địa chỉ cho các thiết bị nhận
  - 8-bit nghịch đảo địa chỉ
  - 8-bit dòng lệnh
  - 8-bit nghịch đảo dòng lệnh
  - Một xung nhịp  $562.5\mu\text{s}$  để biểu thị kết thúc truyền tin.



Hình minh họa các định dạng của một khung truyền NEC IR, cho một địa chỉ 00h (00000000b) và một dòng lệnh ADh (10101101b).

- Chú ý trong hình trên cần:
  - 27ms để truyền cả 16 bit cho địa chỉ (địa chỉ + nghịch đảo) và 16 bit cho các lệnh (dòng lệnh + nghịch đảo). Điều này xuất phát từ mỗi của các khối 16 bit cuối cùng chứa tám bit ‘0’ và tám bit ‘1’ - cho  $(8 * 1.125ms) + (8 * 2.25ms)$ .
  - 67.5ms để truyền tải đầy đủ các thông điệp.
- Lặp lại mã:
  - Nếu phím trên bộ điều khiển từ xa được giữ lâu, một mã lặp lại sẽ được ban hành, thường là khoảng 40ms sau khi có xung nhịp biểu thị sự kết thúc của thông điệp. Một mã lặp lại sẽ tiếp tục được gửi khoảng thời gian là 108ms, cho đến khi khóa chính cuối cùng đã được phát hành. Mã lặp lại bao gồm những điều sau đây, theo thứ tự:
    - Một xung nhịp 9ms
    - Một khoảng trống 2.25ms
    - Một xung nhịp 562.5μs để đánh dấu sự kết thúc của khoảng trống (kết thúc của mã lặp lại truyền).



Hình minh họa sự lặp truyền của hai mã lặp lại sau một khung thông báo ban đầu được gửi đi.

## CHƯƠNG IV: QUẢN GIA ĐIỆN TỬ TRÊN THIẾT BỊ DI ĐỘNG

### IV.1. Đặc tả Use Case

#### IV.1.1. Thêm thiết bị

Tác nhân	Người dùng
Mục đích	Chọn thiết bị điều khiển và thêm vào danh sách thiết bị sử dụng.
Dòng sự kiện chính	<ol style="list-style-type: none"> <li>1. Ứng dụng hiển thị màn hình thêm thiết bị. Người dùng nhấn nút <b>THÊM THIẾT BỊ</b></li> <li>2. Ứng dụng hiển thị màn hình chọn loại thiết bị. Người dùng chọn loại thiết bị cần thêm.</li> <li>3. Ứng dụng hiển thị màn hình chọn hãng thiết bị. Người dùng chọn hãng thiết bị tương ứng.</li> <li>4. Ứng dụng hiển thị màn hình chọn mẫu thiết bị. Người dùng chọn mẫu thiết bị tương ứng.</li> </ol>
Dòng sự kiện khác	Không có.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng hiển thị màn hình thêm thiết bị.
Trạng thái hệ thống sau khi thực hiện Use-case	Ứng dụng hiển thị màn hình điều khiển và thiết bị được thêm vào danh sách sử dụng.
Điểm mở rộng	Không có.



#### IV.1.2. Cập nhật tên thiết bị

a

Tác nhân	Người dùng
Mục đích	Thay đổi tên thiết bị đã chọn.
Dòng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng vào màn hình điều khiển thiết bị cần đổi tên. Chọn menu trên ActionBar, nhấn vào nút Cập nhật.</li> <li>2. Người dùng thực hiện thay đổi tên thiết bị sau đó nhấn nút OK.</li> <li>3. Ứng dụng thông báo cập nhật tên thiết bị thành công.</li> </ol>
Dòng sự kiện khác	Không có.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình điều khiển thiết bị.
Trạng thái hệ thống sau khi thực hiện Use-case	Tên thiết bị được cập nhật.
Điểm mở rộng	Không có.

#### IV.1.3. Xóa thiết bị

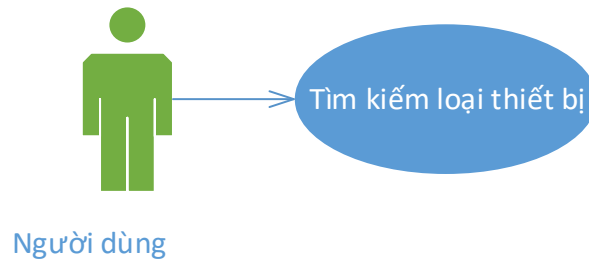
Tác nhân	Người dùng
Mục đích	Xóa thiết bị đã chọn khỏi danh sách sử dụng.
Dòng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng vào màn hình điều khiển thiết bị cần xóa. Chọn menu trên ActionBar, nhấn vào nút Xóa.</li> <li>2. Người dùng nhấn nút OK xác nhận xóa thiết bị được chọn khỏi danh sách.</li> </ol>

	3. Ứng dụng thông báo xóa thành công.
Dòng sự kiện khác	Nếu danh sách sử dụng không còn thiết bị nào thì ứng dụng hiển thị màn hình chọn loại thiết bị.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình điều khiển thiết bị.
Trạng thái hệ thống sau khi thực hiện Use-case	Thiết bị được chọn bị xóa khỏi danh sách.
Điểm mở rộng	Không có.

#### IV.1.4. Hiển thị danh sách loại thiết bị

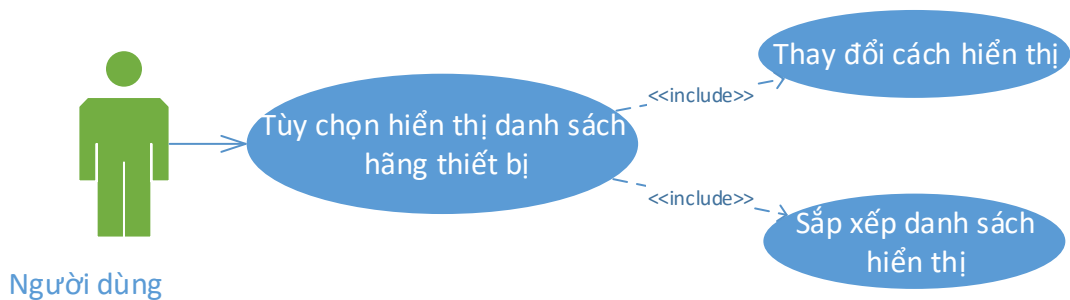
Tác nhân	Người dùng
Mục đích	Chọn loại hiển thị và sắp xếp danh sách loại thiết bị.
Dòng sự kiện chính	Người dùng vào màn hình chọn loại thiết bị. Chọn menu trên ActionBar, thay đổi cách hiển thị danh sách hoặc sắp xếp danh sách theo yêu cầu.
Dòng sự kiện khác	Không có.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình chọn loại thiết bị.
Trạng thái hệ thống sau khi thực hiện Use-case	Danh sách loại thiết bị được thay đổi theo yêu cầu của người dùng.
Điểm mở rộng	Không có.

#### IV.1.5. Tìm kiếm loại thiết bị



Tác nhân	Người dùng
Mục đích	Tìm kiếm loại thiết bị.
Dòng sự kiện chính	Người dùng vào màn hình chọn loại thiết bị. Nhập loại thiết bị hoặc những từ gợi ý đến loại thiết bị cần tìm vào ô Search
Dòng sự kiện khác	Không có.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình chọn loại thiết bị.
Trạng thái hệ thống sau khi thực hiện Use-case	Hiển thị danh sách chứa tên loại thiết bị cần tìm kiếm.
Điểm mở rộng	Không có.

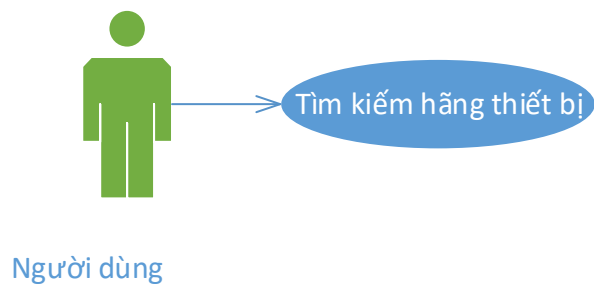
#### IV.1.6. Hiển thị danh sách hãng thiết bị



Tác nhân	Người dùng
Mục đích	Chọn loại hiển thị và sắp xếp danh sách hãng thiết bị.

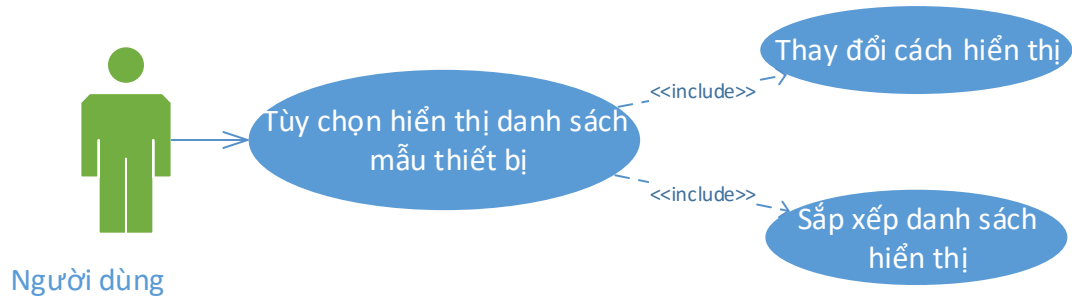
Dòng sự kiện chính	Người dùng vào màn hình chọn hãng thiết bị. Chọn menu trên ActionBar, thay đổi cách hiển thị danh sách hoặc sắp xếp danh sách theo yêu cầu.
Dòng sự kiện khác	Không có.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình chọn hãng thiết bị.
Trạng thái hệ thống sau khi thực hiện Use-case	Danh sách hãng thiết bị được thay đổi theo yêu cầu của người dùng.
Điểm mở rộng	Không có.

#### IV.1.7. Tìm kiếm hãng thiết bị



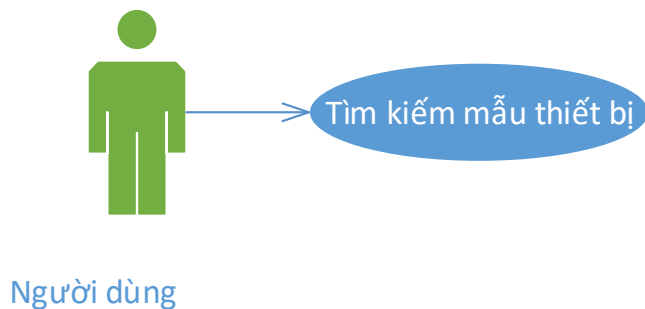
Tác nhân	Người dùng
Mục đích	Tìm kiếm hãng thiết bị.
Dòng sự kiện chính	Người dùng vào màn hình chọn hãng thiết bị. Nhập hãng thiết bị hoặc những từ gợi ý đến hãng thiết bị cần tìm vào ô Search
Dòng sự kiện khác	Không có.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình chọn hãng thiết bị.
Trạng thái hệ thống sau khi thực hiện Use-case	Hiển thị danh sách chứa tên hãng thiết bị cần tìm kiếm.
Điểm mở rộng	Không có.

#### IV.1.8. Hiển thị danh sách mẫu thiết bị



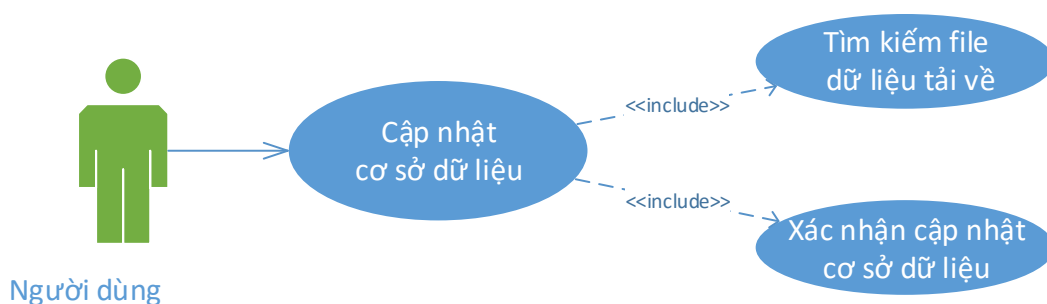
Tác nhân	Người dùng
Mục đích	Chọn loại hiển thị và sắp xếp danh sách mẫu thiết bị.
Dòng sự kiện chính	Người dùng vào màn hình chọn mẫu thiết bị. Chọn menu trên ActionBar, thay đổi cách hiển thị danh sách hoặc sắp xếp danh sách theo yêu cầu.
Dòng sự kiện khác	Không có.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình chọn mẫu thiết bị.
Trạng thái hệ thống sau khi thực hiện Use-case	Danh sách mẫu thiết bị được thay đổi theo yêu cầu của người dùng.
Điểm mở rộng	Không có.

#### IV.1.9. Tìm kiếm mẫu thiết bị



Tác nhân	Người dùng
Mục đích	Tìm kiếm mẫu thiết bị.
Dòng sự kiện chính	Người dùng vào màn hình chọn mẫu thiết bị. Nhập mẫu thiết bị hoặc những từ gợi ý đến mẫu thiết bị cần tìm vào ô Search
Dòng sự kiện khác	Không có.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình chọn mẫu thiết bị.
Trạng thái hệ thống sau khi thực hiện Use-case	Hiển thị danh sách chứa tên mẫu thiết bị cần tìm kiếm.
Điểm mở rộng	Không có.

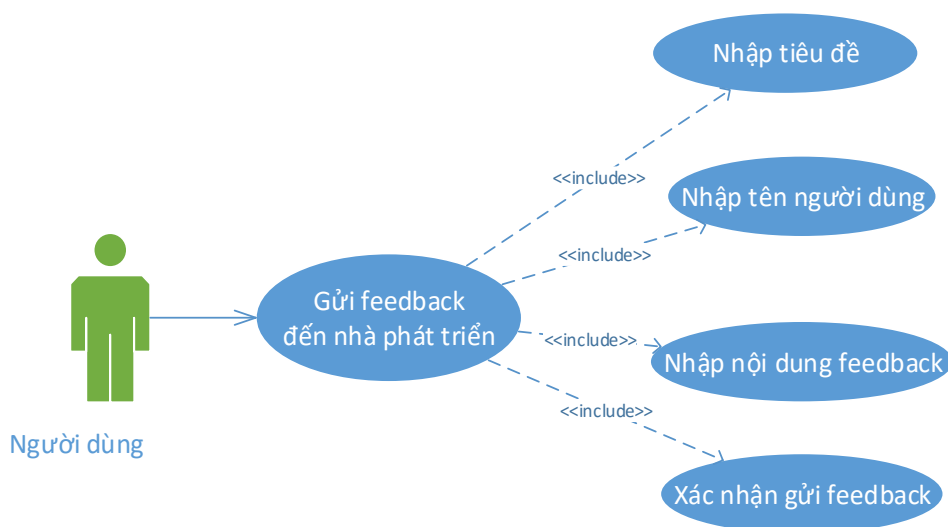
#### IV.1.10. Cập nhật cơ sở dữ liệu



Tác nhân	Người dùng
Mục đích	Cập nhật cơ sở dữ liệu mới
Dòng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng vào màn hình cài đặt chọn cập nhật cơ sở dữ liệu.</li> <li>2. Tìm kiếm file dữ liệu đã tải về.</li> <li>3. Chọn OK để cập nhật cơ sở dữ liệu mới.</li> </ol>
Dòng sự kiện khác	Kiểm tra file dữ liệu người chọn có hợp lệ hay không? Nếu không hợp lệ thông báo lỗi cho người dùng.
Các yêu cầu đặc biệt	Không có.

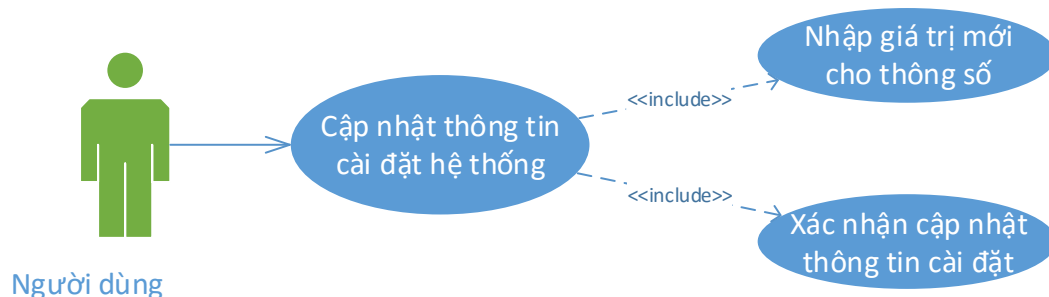
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình cài đặt
Trạng thái hệ thống sau khi thực hiện Use-case	Thông báo kết quả, yêu cầu khởi động lại ứng dụng và trở về màn hình chính
Điểm mở rộng	Không có.

#### IV.1.11. Gửi phản hồi



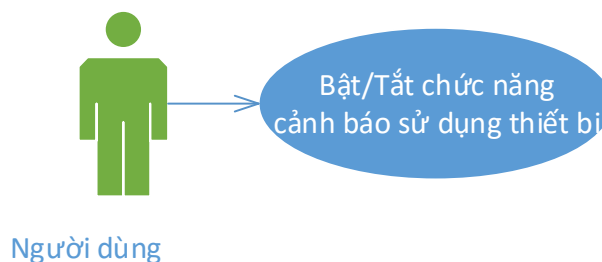
Tác nhân	Người dùng
Mục đích	Gửi phản hồi đến email nhà phát triển
Dòng sự kiện chính	Người dùng vào màn hình cài đặt chọn gửi feedback. Nhập tiêu đề, tên người dùng, và nội dung feedback. Chọn OK để gửi nội dung feedback đến email nhà phát triển.
Dòng sự kiện khác	Không có
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình cài đặt
Trạng thái hệ thống sau khi thực hiện Use-case	Thông báo kết quả và trở về màn hình chính
Điểm mở rộng	Không có.

#### IV.1.12. Thay đổi thông số cài đặt



Tác nhân	Người dùng
Mục đích	Cập nhật thông tin cài đặt hệ thống
Dòng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng vào màn hình cài đặt chọn cập nhật thông số cài đặt</li> <li>2. Nhập giá trị mới cho thông số</li> <li>3. Chọn OK để cập nhật các thông số cài đặt</li> </ol>
Dòng sự kiện khác	Kiểm tra file dữ liệu người chọn có hợp lệ hay không? Nếu không hợp lệ thông báo lỗi cho người dùng.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình cài đặt
Trạng thái hệ thống sau khi thực hiện Use-case	Thông báo kết quả, yêu cầu khởi động lại ứng dụng và trở về màn hình chính
Điểm mở rộng	Không có.

#### IV.1.13. Bật/Tắt chức năng cảnh báo sử dụng thiết bị





Tác nhân	Người dùng
Mục đích	Bật tắt chức năng cảnh báo sử dụng thiết bị
Dòng sự kiện chính	Người dùng vào màn hình cài đặt người dùng tích chọn hay bỏ chọn checkbox chức năng cảnh báo sử dụng thiết bị để bật tắt chức năng.
Dòng sự kiện khác	Không có
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình cài đặt
Trạng thái hệ thống sau khi thực hiện Use-case	Thông báo kết quả và trở về màn hình chính
Điểm mở rộng	Không có.

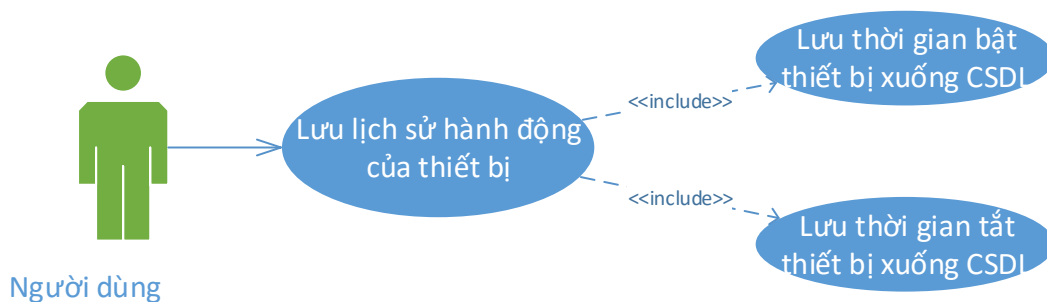
#### IV.1.14. Cài đặt chế độ cảnh báo



Tác nhân	Người dùng
Mục đích	Cài đặt chế độ cảnh báo: Rung, âm thanh
Dòng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng vào màn hình cài đặt chọn cài đặt chế độ cảnh báo, người dùng tích chọn hay bỏ chọn chế độ rung, chọn âm thanh cần thay đổi.</li> <li>2. Nhấn OK để hoàn thành thao tác.</li> </ol>

Dòng sự kiện khác	Kiểm tra file dữ liệu người chọn có hợp lệ hay không? Nếu không hợp lệ thông báo lỗi cho người dùng.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang hiển thị màn hình cài đặt
Trạng thái hệ thống sau khi thực hiện Use-case	Thông báo kết quả và trở về màn hình chính
Điểm mở rộng	Không có.

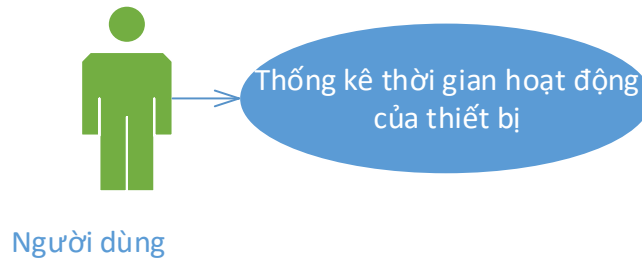
#### IV.1.15. Lưu lịch sử hoạt động của thiết bị



Tác nhân	Người dùng
Mục đích	Lưu lịch sử hành động bật tắt của thiết bị
Dòng sự kiện chính	Người dùng thực hiện chức năng bật tắt một thiết bị tại màn hình điều khiển, xác nhận thiết bị đã được bật hoặc tắt, lưu thời gian bật/tắt của thiết bị xuống cơ sở dữ liệu.
Dòng sự kiện khác	Kiểm tra nếu tivi đang bật/tắt và người dùng vẫn thực hiện thao tác bật/tắt thì không lưu thời gian khi thực hiện thao tác này xuống cơ sở dữ liệu
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng đang ở màn hình điều khiển thiết bị

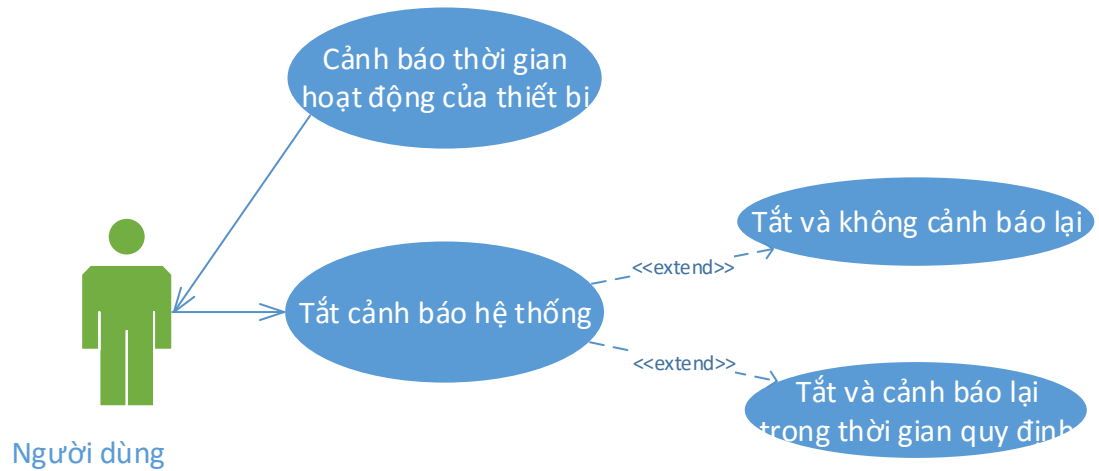
Trạng thái hệ thống sau khi thực hiện Use-case	Ứng dụng đang ở màn hình điều khiển thiết bị
Điểm mở rộng	Không có.

#### IV.1.16. Thống kê thời gian hoạt động của thiết bị



Tác nhân	Người dùng
Mục đích	Thống kê thời gian hoạt động của thiết bị đang được sử dụng theo ngày, biểu đồ
Dòng sự kiện chính	Người dùng vào màn hình thống kê của thiết bị, hiển thị số liệu thống kê thời gian hoạt động của thiết bị theo ngày bằng biểu đồ. Hiển thị ngày sử dụng cao nhất.
Dòng sự kiện khác	Không có
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Hệ thống đang ở màn điều khiển thiết bị.
Trạng thái hệ thống sau khi thực hiện Use-case	Hiển thị màn hình thống kê của thiết bị với kết quả thống kê của thiết bị
Điểm mở rộng	Không có.

#### IV.1.17. Cảnh báo thời gian hoạt động của thiết bị



Tác nhân	Người dùng
Mục đích	Cảnh báo thời gian hoạt động của thiết bị khi qua thời gian tối đa hoạt động của thiết bị được cấu hình.
Dòng sự kiện chính	Hệ thống cảnh báo thời gian hoạt động của thiết bị nếu thiết bị hoạt động quá thời gian hoạt động tối đa được cấu hình sẵn.
Dòng sự kiện khác	Người dùng chọn tắt thiết bị hệ thống thực hiện tắt thiết bị. Người dùng chọn tắt cảnh báo hệ thống thực hiện tắt cảnh báo, thực hiện cảnh báo lại trong 5p. Người dùng chọn cập nhật thời gian tắt bật thiết bị. Hệ thống lưu thời gian tắt là thời gian hiện tại, và thời gian bật lần mới.
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Bất kỳ màn hình nào của ứng dụng.
Trạng thái hệ thống sau khi thực hiện Use-case	Thông báo kết quả và trở về màn hình đang hiển thị
Điểm mở rộng	Không có.

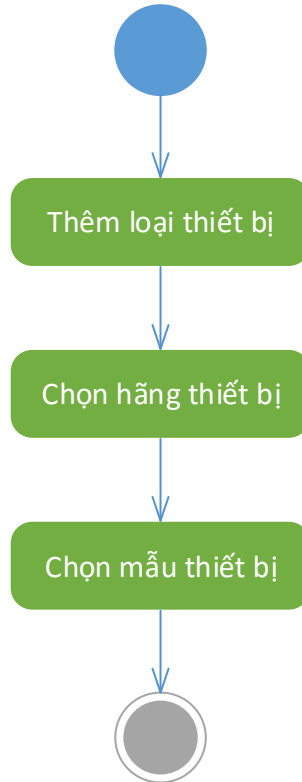
#### IV.1.18. Đánh giá tuổi thọ của thiết bị



Tác nhân	Người dùng
Mục đích	Đánh giá tuổi thọ hoạt động của thiết bị
Dòng sự kiện chính	Dựa vào tổng thời gian hoạt động của thiết bị thực hiện tính tổng thời gian hoạt động trung bình của thiết bị, đánh giá tuổi thọ của thiết bị và hiển thị cho người dùng xem ở màn hình thống kê.
Dòng sự kiện khác	Không có
Các yêu cầu đặc biệt	Không có.
Trạng thái hệ thống khi bắt đầu thực hiện Use-case	Ứng dụng ở màn hình điều khiển
Trạng thái hệ thống sau khi thực hiện Use-case	Hiển thị kết quả lên màn hình thống kê
Điểm mở rộng	Không có.

## IV.2. Biểu đồ hoạt động

### IV.2.1. Thêm thiết bị



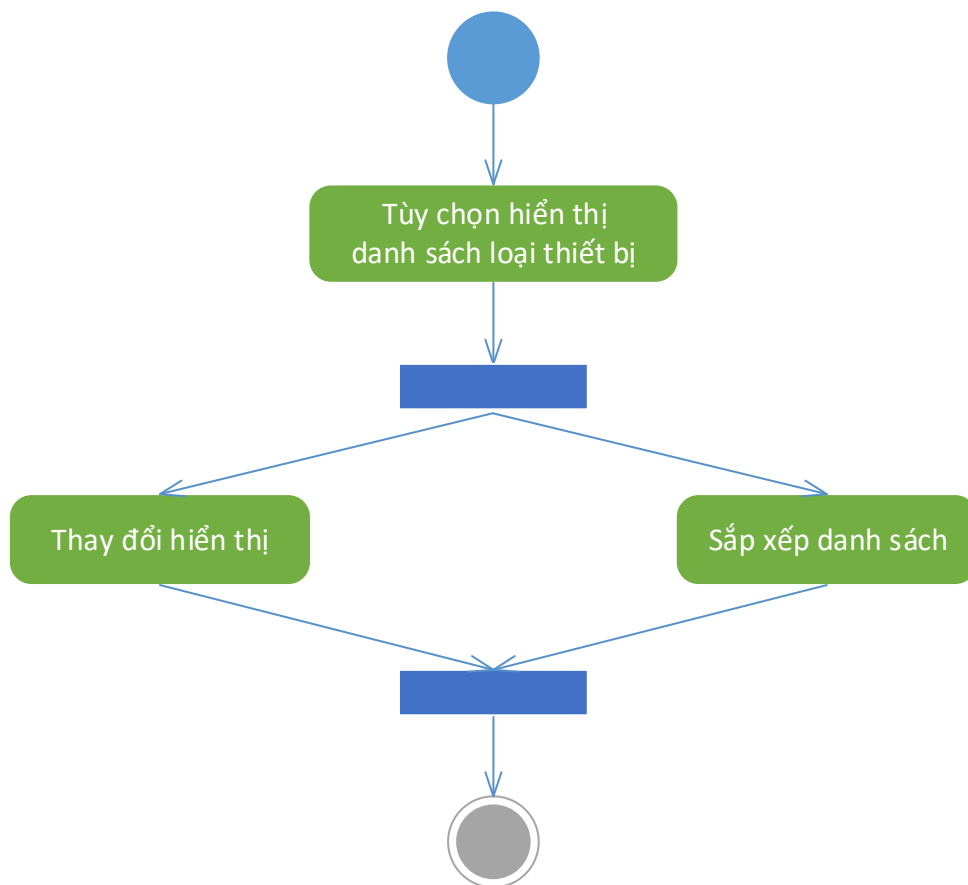
### IV.2.2. Cập nhật tên thiết bị



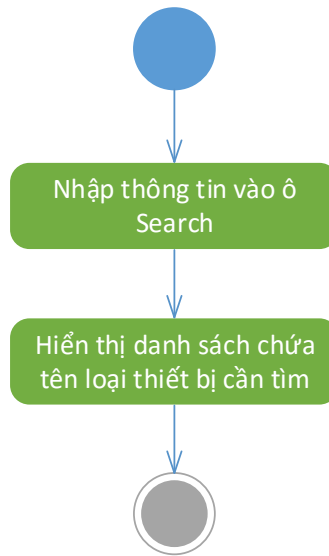
#### IV.2.3. Xóa thiết bị



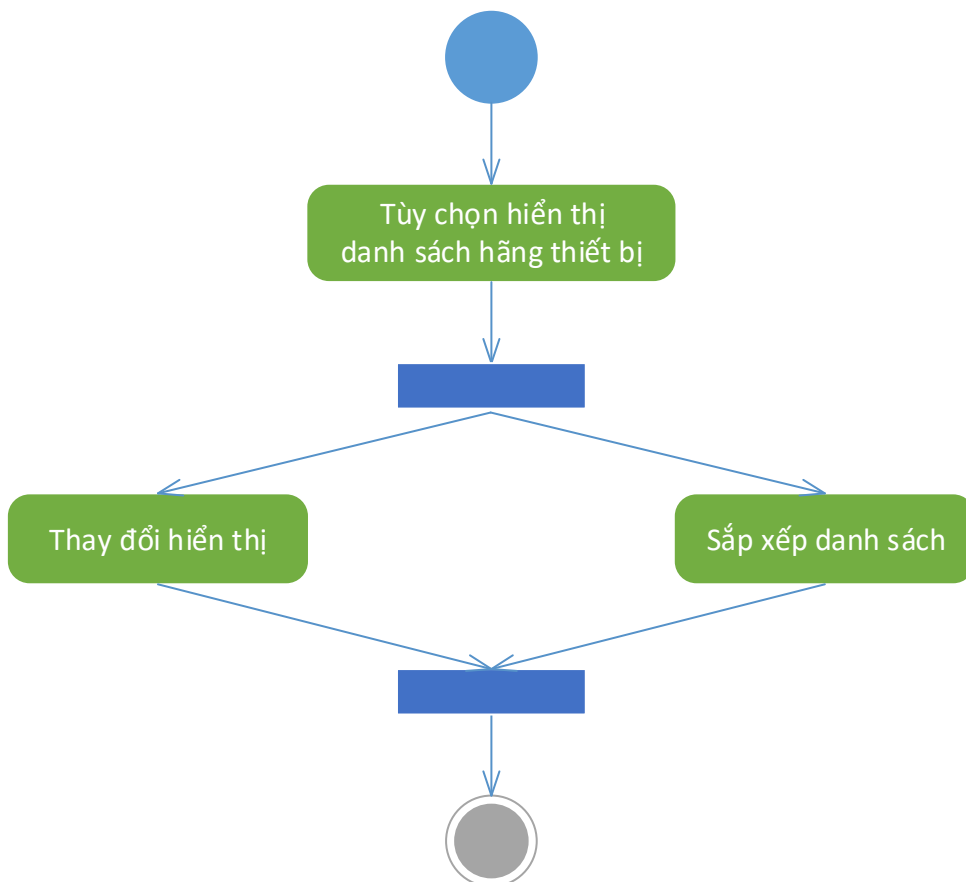
#### IV.2.4. Hiện thị danh sách loại thiết bị



#### IV.2.5. Tìm kiếm loại thiết bị



#### IV.2.6. Hiển thị danh sách hãng thiết bị

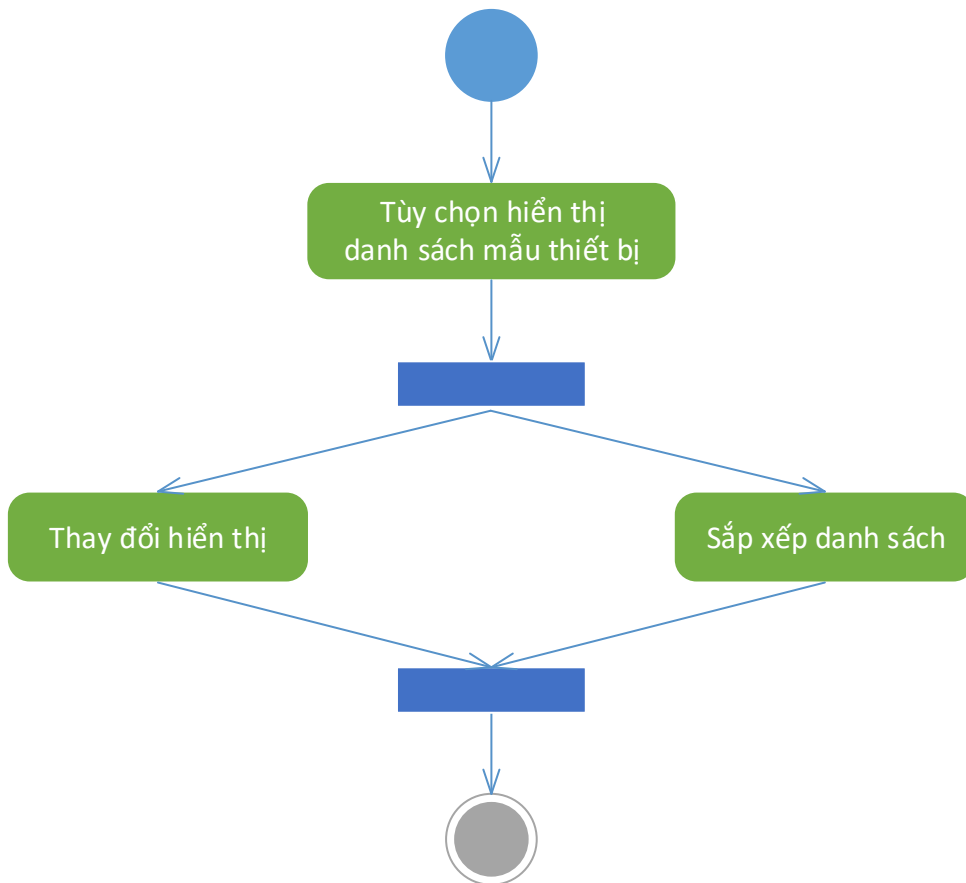




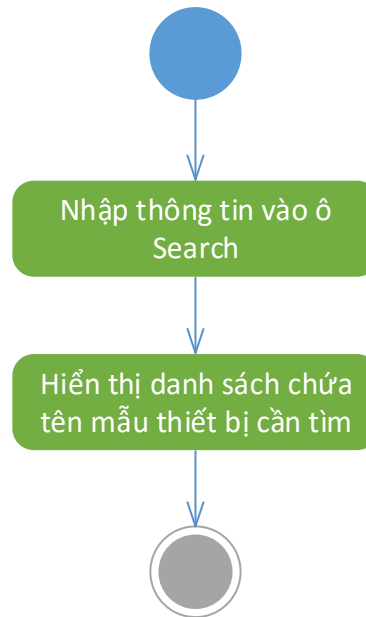
#### IV.2.7. Tìm kiếm hãng thiết bị



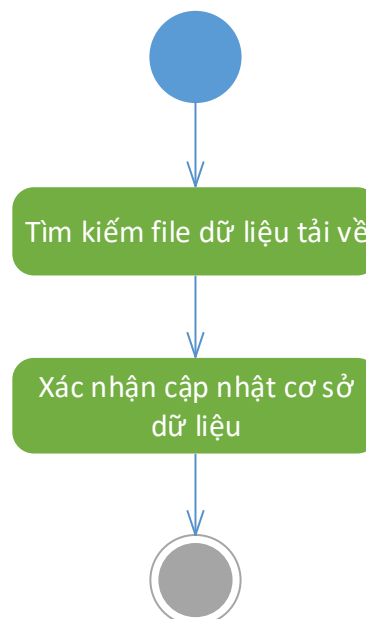
#### IV.2.8. Hiện thị danh sách mẫu thiết bị



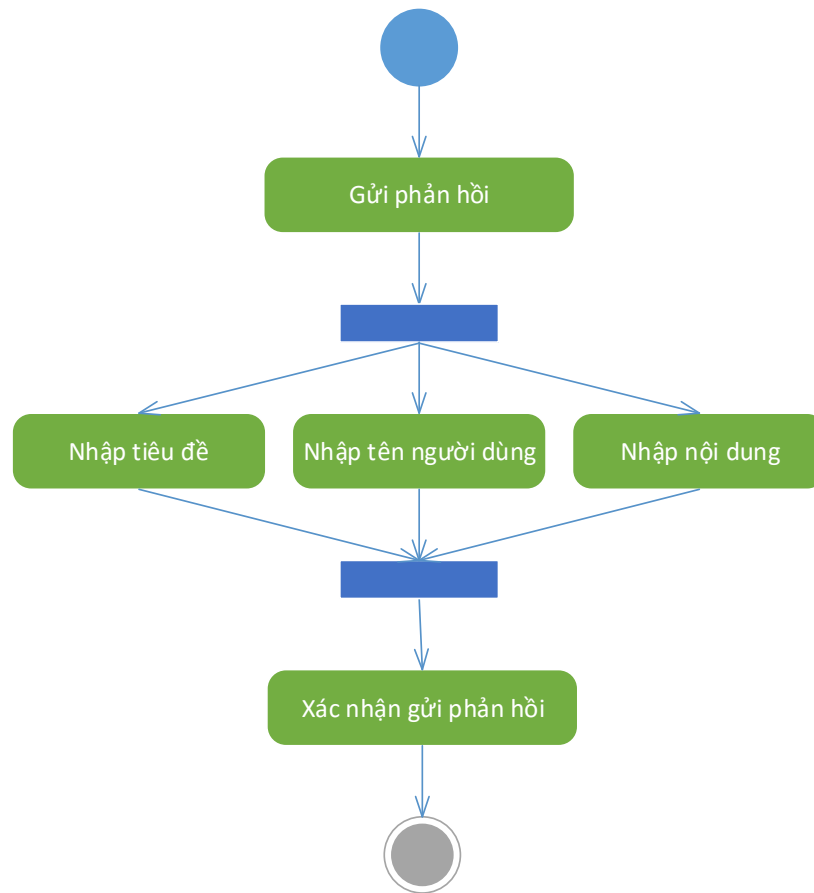
#### IV.2.9. Tìm kiếm mẫu thiết bị



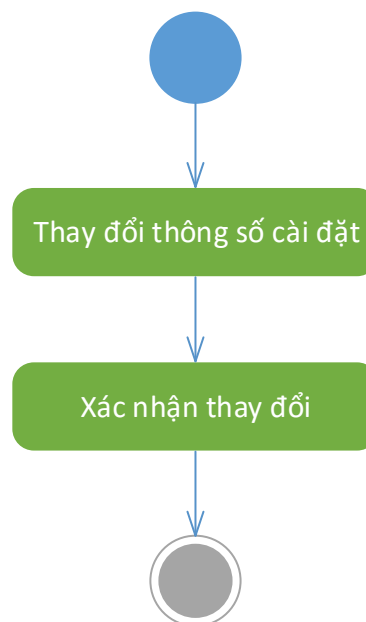
#### IV.2.10. Cập nhật cơ sở dữ liệu

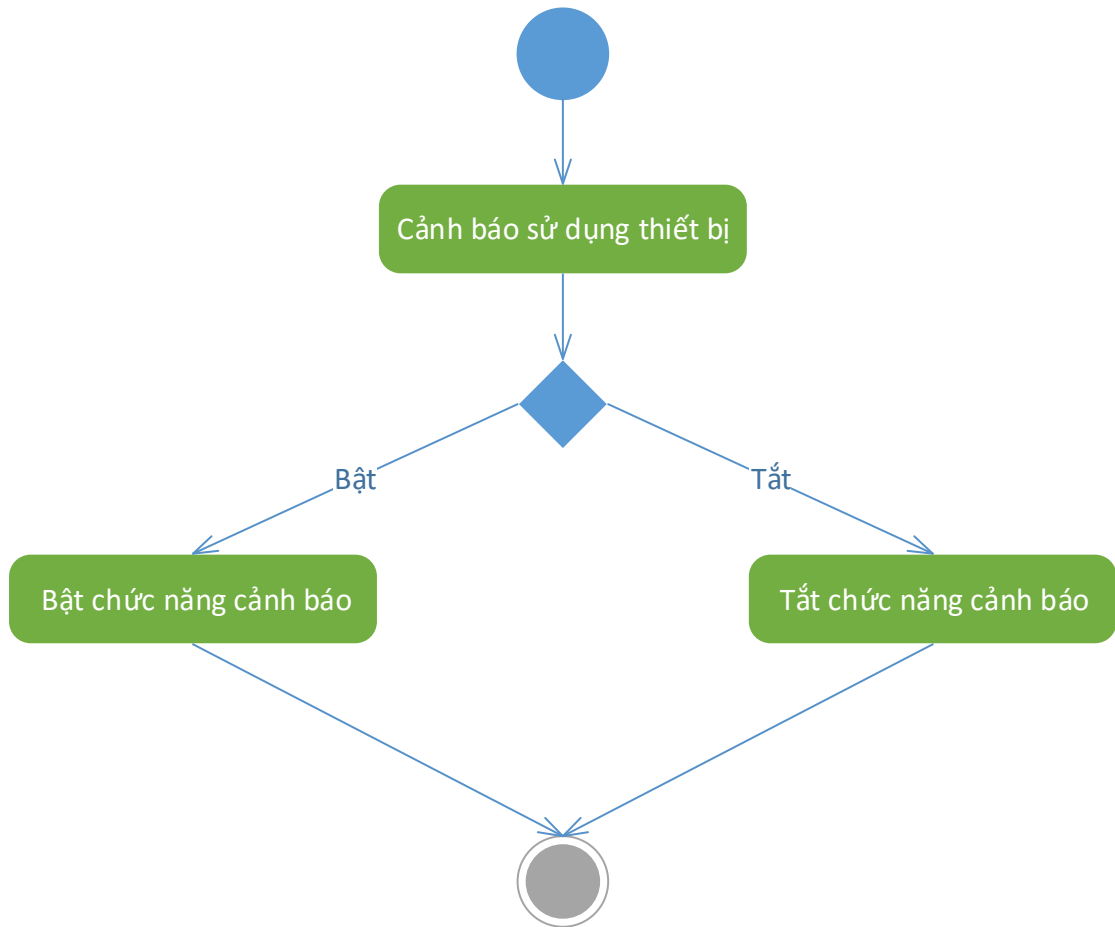


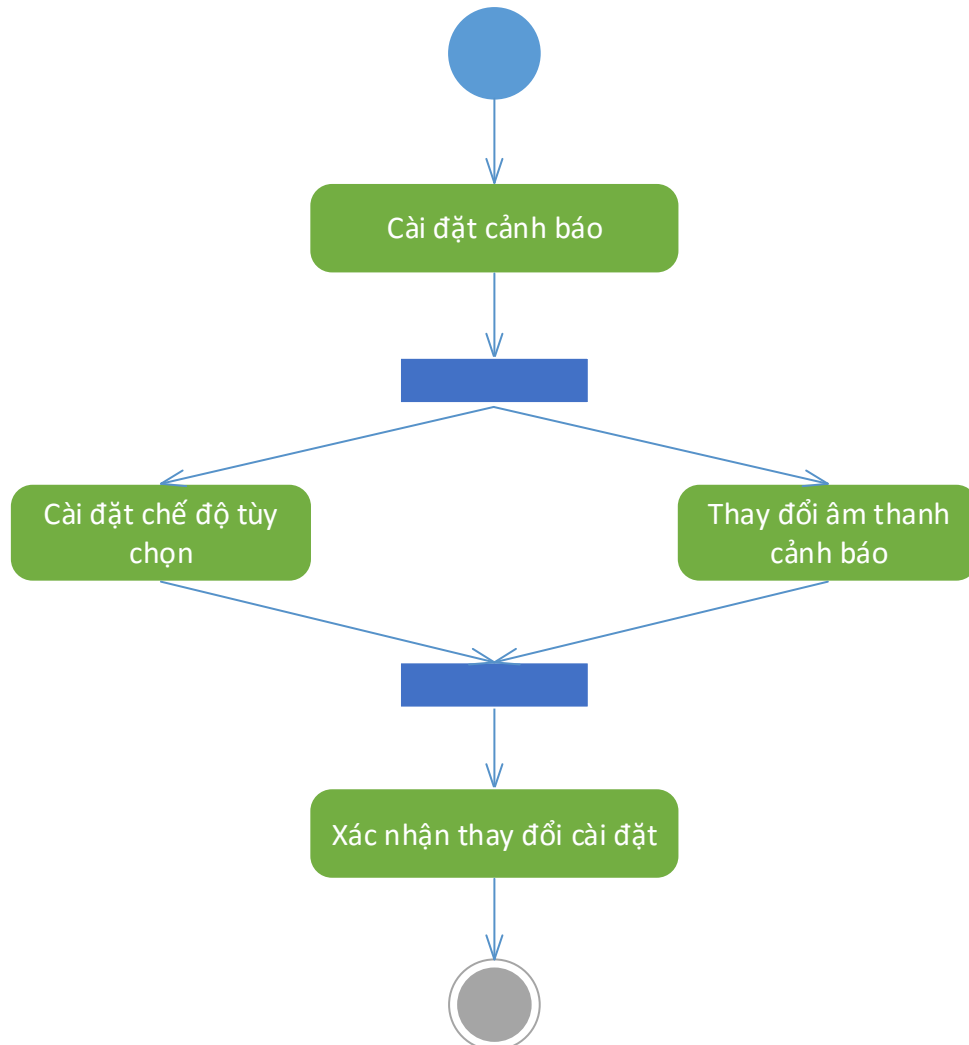
#### IV.2.11. Gửi phản hồi



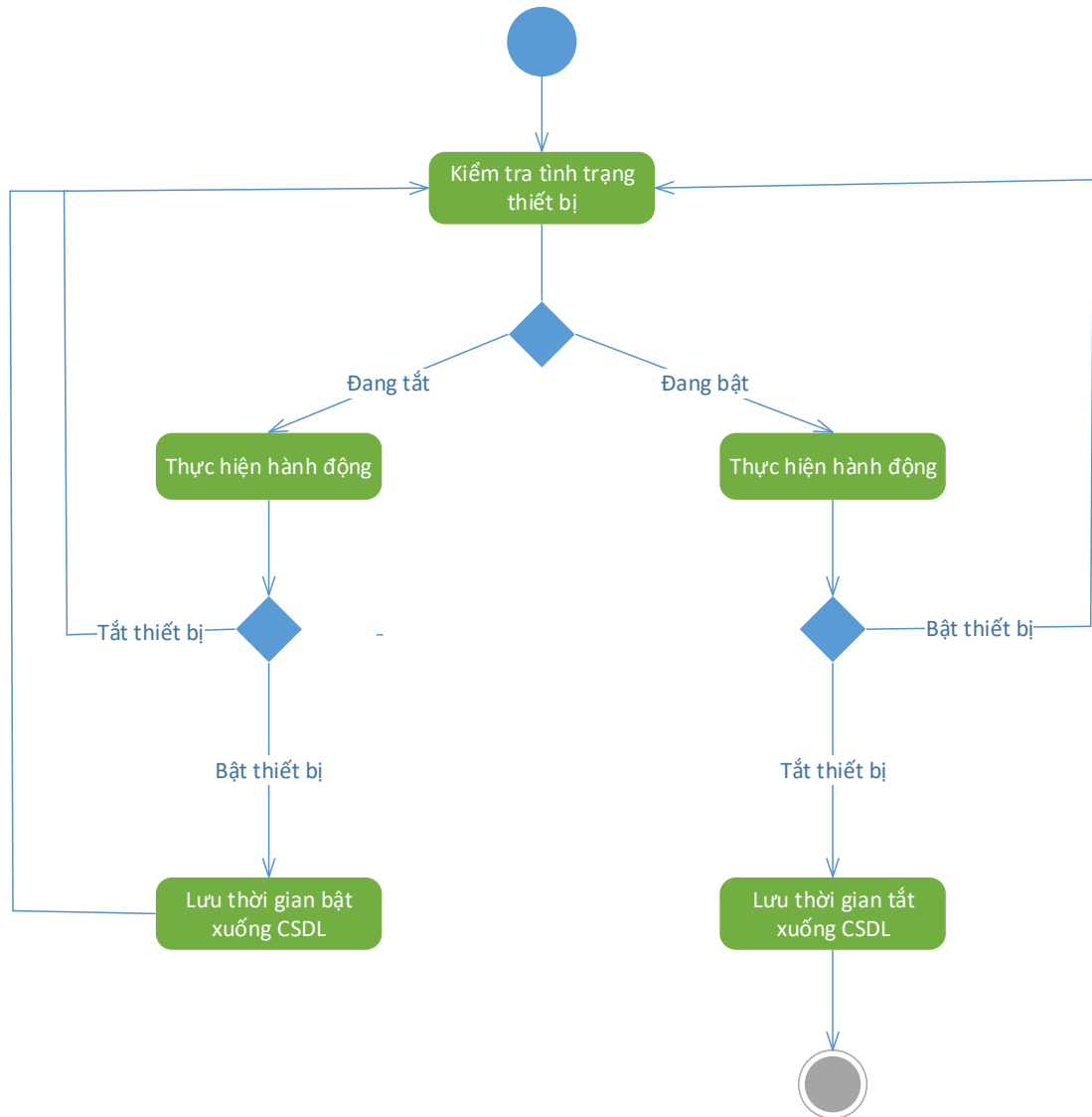
#### IV.2.12. Thay đổi thông số cài đặt



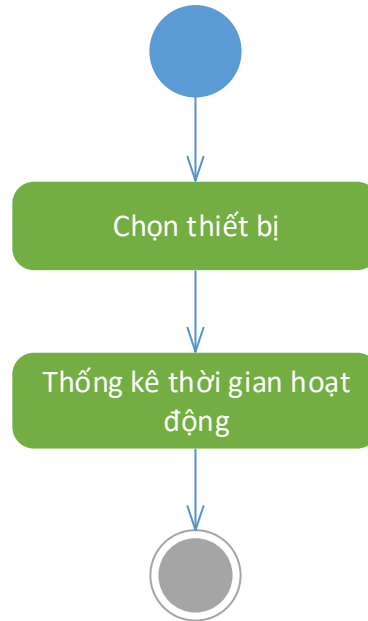
**IV.2.13. Bật/Tắt chức năng cảnh báo sử dụng thiết bị**

**IV.2.14. Cài đặt chế độ cảnh báo**

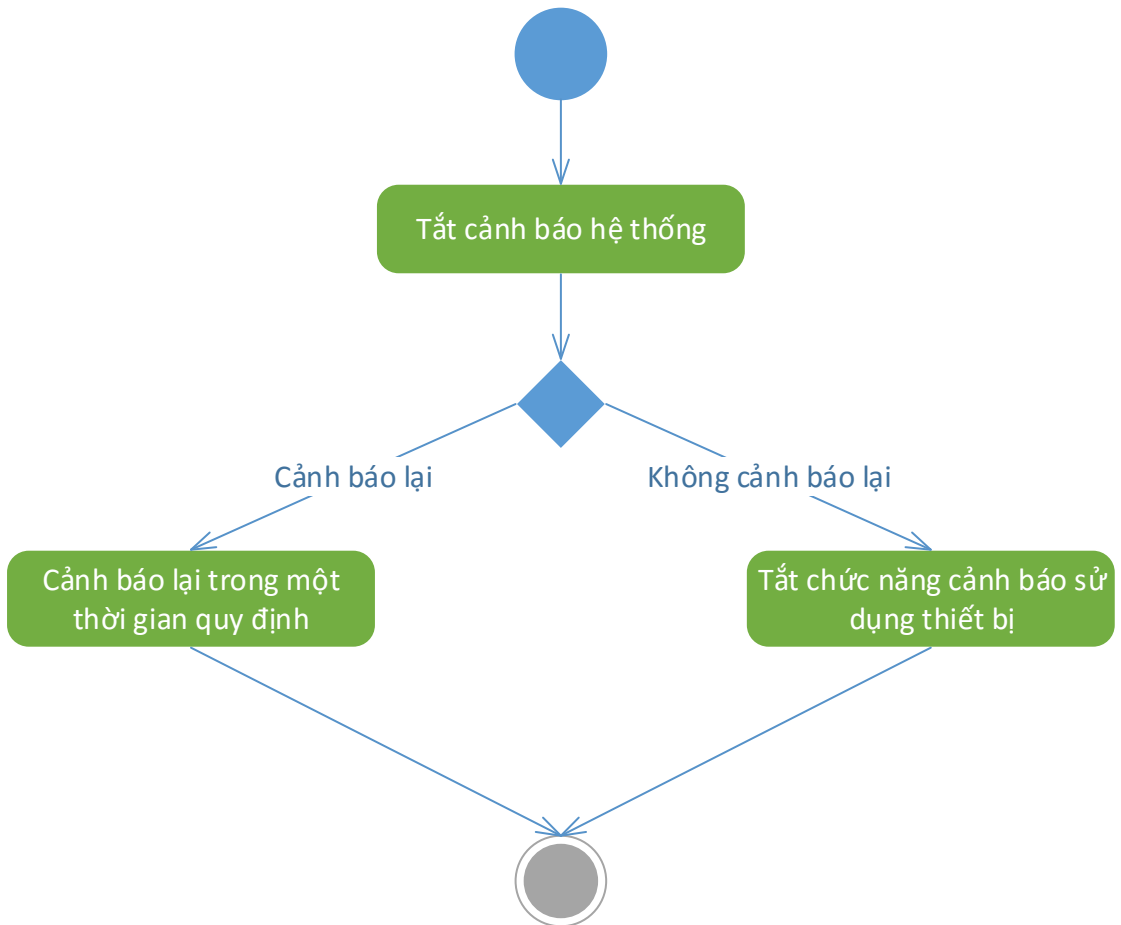
#### IV.2.15. Lưu lịch sử hoạt động của thiết bị



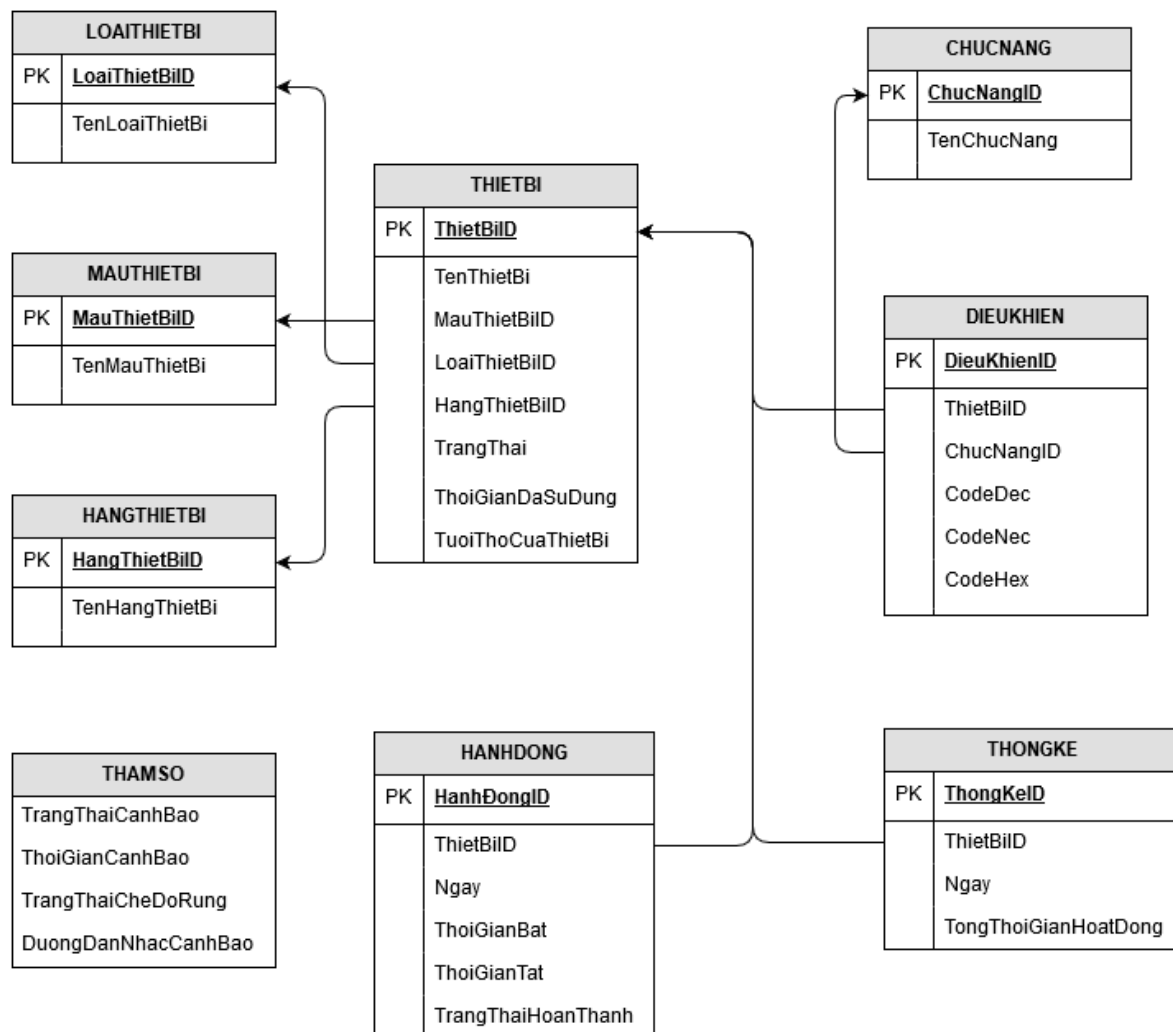
#### IV.2.16. Thống kê thời gian hoạt động của thiết bị



#### IV.2.17. Cảnh báo thời gian hoạt động của thiết bị



### IV.3. Thiết kế dữ liệu



#### IV.3.1. MAUTHIETBI

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	MauThietBiID	INT	Khóa chính
2	TenMaThietBi	TEXT	

#### IV.3.2. LOAITHIETBI

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	LoaiThietBiID	INT	Khóa chính
2	TenLoaiThietBi	TEXT	



**IV.3.3. HANGTHIETBI**

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	HangThietBiID	INT	Khóa chính
2	TenHangThietBi	TEXT	

**IV.3.4. THIETBI**

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	ThietBiID	INT	Khóa chính
2	TenThietBi	TEXT	
3	MaThietBiID	INT	Khóa ngoại
4	LoaiThietBiID	INT	Khóa ngoại
5	HangThietBiID	INT	Khóa ngoại
6	TrangThai	BOOLEAN	

**IV.3.5. CHUCNANG**

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	ChucNangID	INT	Khóa chính
2	TenChucNang	TEXT	

**IV.3.6. DIEUKHIEN**

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	DieuKhenID	INT	Khóa chính
2	ThietBiID	INT	Khóa ngoại
3	ChucNangID	INT	Khóa ngoại
4	CodeDec	NVARCHAR(MAX)	
5	CodeNec	NVARCHAR(MAX)	
6	CodeHex	NVARCHAR(MAX)	

**IV.3.7. HOATDONG**

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	HoatDongID	INT	Khóa chính
2	ThietBiID	INT	Khóa ngoại
3	Ngay	DATETIME	
4	ThoiGianBat	TIME	
5	ThoiGianTat	TIME	
6	TrangThaiHoanThanh	BOOLEAN	

**IV.3.8. THONGKE**

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	ThongKeID	INT	Khóa chính
2	ThietBiID	INT	Khóa ngoại
3	Ngay	DATETIME	
4	TongThoiGian	DOUBLE	

**IV.3.9. THAMSO**

STT	Thuộc tính	Kiểu dữ liệu	Ràng buộc
1	TrangThaiCanhBao	BOOLEAN	
2	ThoiGianCanhBao	TEXT	
3	TrangThaiCheDoRung	BOOLEAN	
4	DuongDanNhanCanhBao	TEXT	

#### IV.4. Thiết kế giao diện

##### IV.4.1. Danh sách các màn hình

STT	Tên màn hình	Ý nghĩa
1	Màn hình chính	Màn hình xuất hiện khi bắt đầu ứng dụng.
2	Màn hình chọn thiết bị	Hiện thị danh sách thiết bị.
3	Màn hình chọn hãng	Hiện thị danh sách hãng tương ứng với thiết bị đã chọn.
4	Màn hình chọn mẫu	Hiện thị danh sách mẫu tương ứng với thiết bị và hãng đã chọn.
5	Màn hình điều khiển	Hiện thị các nút điều khiển.
6	Màn hình quản lý hiệu suất	Hiện thị các biểu đồ, thông báo về hiệu suất của thiết bị.

## IV.4.2. Mô tả chi tiết

### IV.4.2.1. Màn hình chính



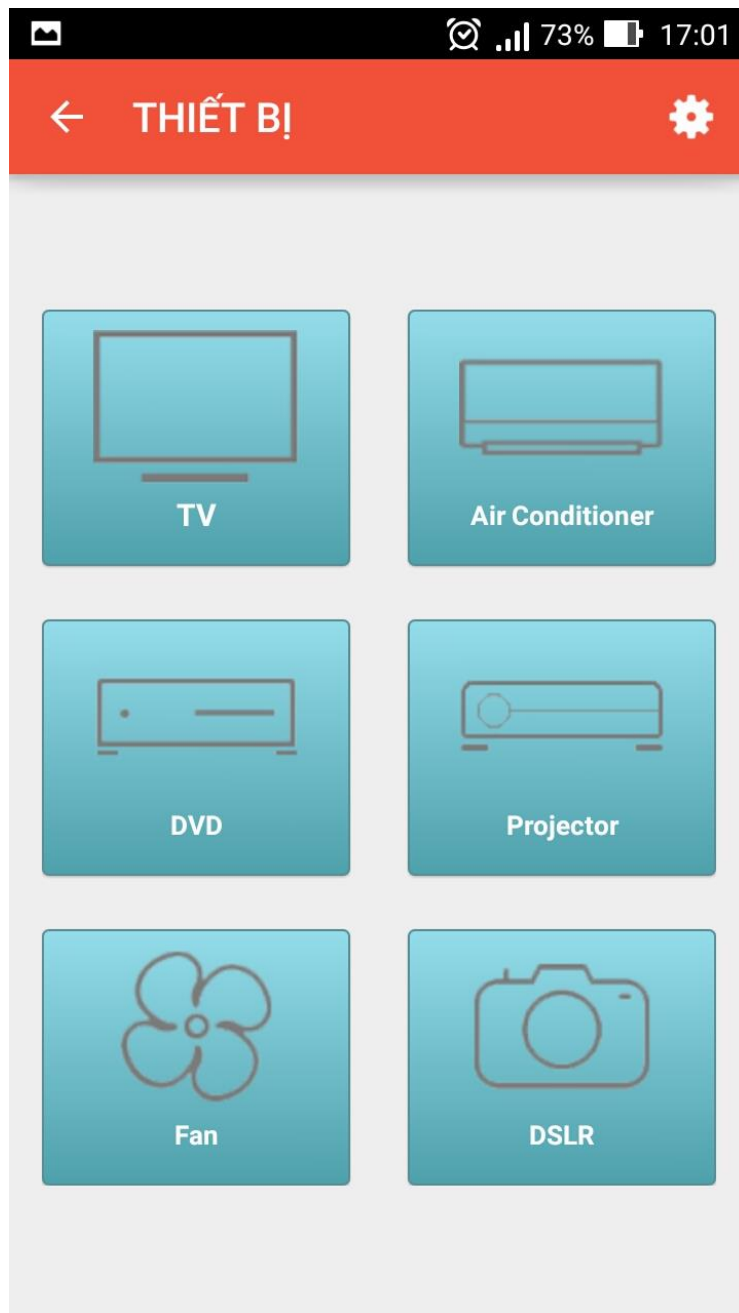
- Các đối tượng

STT	Tên đối tượng	Kiểu	Chức năng	Ghi chú
1	logoImage	ImageView	Hiển thị logo của ứng dụng	
2	btn_AddDevice	Button	Thêm thiết bị	

- Các biến cố

STT	Biến cố	Xử lý
1	Click button THÊM THIẾT BỊ	Chuyển đến màn hình chọn thiết bị

#### IV.4.2.2. Màn hình chọn thiết bị



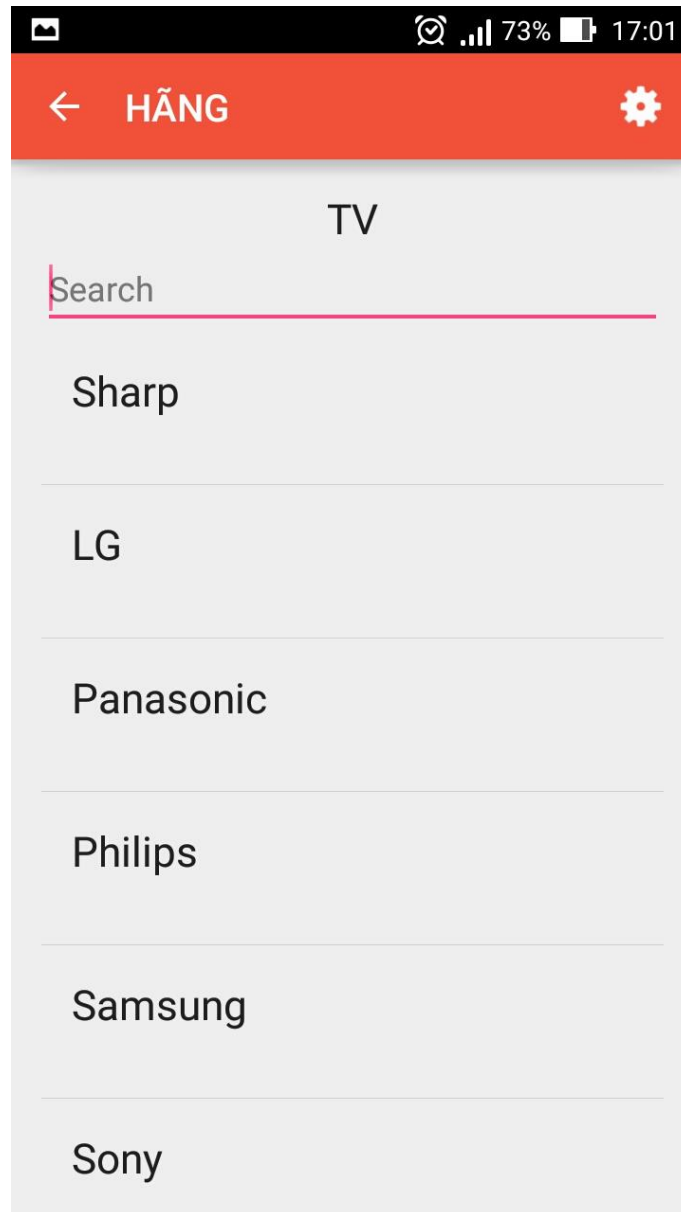
- Các đối tượng

STT	Tên đối tượng	Kiểu	Chức năng	Ghi chú
1	btn_TV	Button	Chọn thiết bị TV	
2	btn_AC	Button	Chọn thiết bị máy điều hòa	
3	btn_DVD	Button	Chọn thiết bị DVD	Chưa hỗ trợ
4	btn_Projector	Button	Chọn thiết bị máy chiếu	
5	btn_Fan	Button	Chọn thiết bị quạt	Chưa hỗ trợ
6	btn_DSLR	Button	Chọn thiết bị máy ảnh kỹ thuật số	Chưa hỗ trợ

- Các biến cố

STT	Biến cố	Xử lý
1	Click button TV	Chuyển qua màn hình chọn hãng cho thiết bị TV.
2	Click button Air Conditioner	Chuyển qua màn hình chọn hãng cho thiết bị máy điều hòa.
3	Click buttton DVD	Hiện thị bảng thông báo thiết bị chưa được hỗ trợ.
4	Click button Projector	Chuyển qua màn hình chọn hãng cho thiết bị máy chiếu.
5	Click button Fan	Hiện thị bảng thông báo thiết bị chưa được hỗ trợ.
6	Click button DSLR	Hiện thị bảng thông báo thiết bị chưa được hỗ trợ.

#### IV.4.2.3. Màn hình chọn hãng



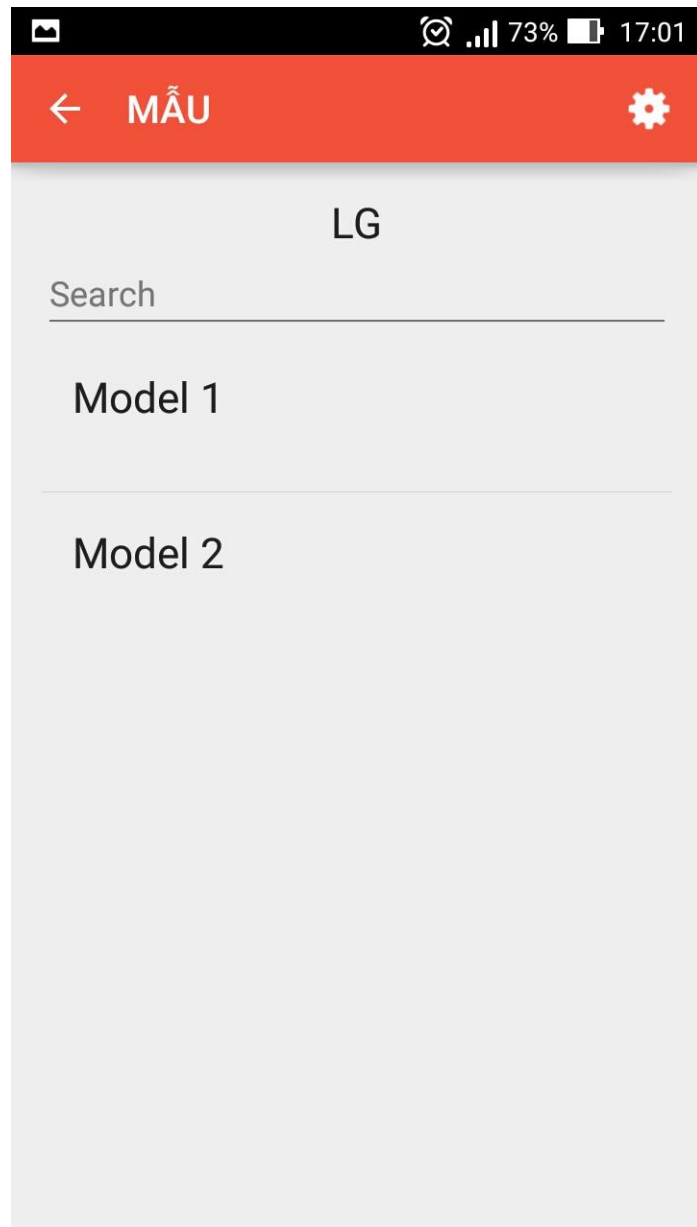
- Các đối tượng

STT	Tên đối tượng	Kiểu	Chức năng	Ghi chú
1	txt_Device	TextView	Hiển thị tên thiết bị được chọn	
2	txt_SearchBrand	EditText	Tìm kiếm hãng	
3	lv_Brand	ListView	Hiển thị danh sách hãng	

- Các biến cố

STT	Biến cố	Xử lý
1	Nhập kí tự vào ô Search	Lọc ra danh sách hãng có chứa kí tự được nhập.
2	Click hãng trên listView	Chuyển qua màn hình chọn mẫu tương ứng.

#### IV.4.2.4. Màn hình chọn mẫu





- Các đối tượng

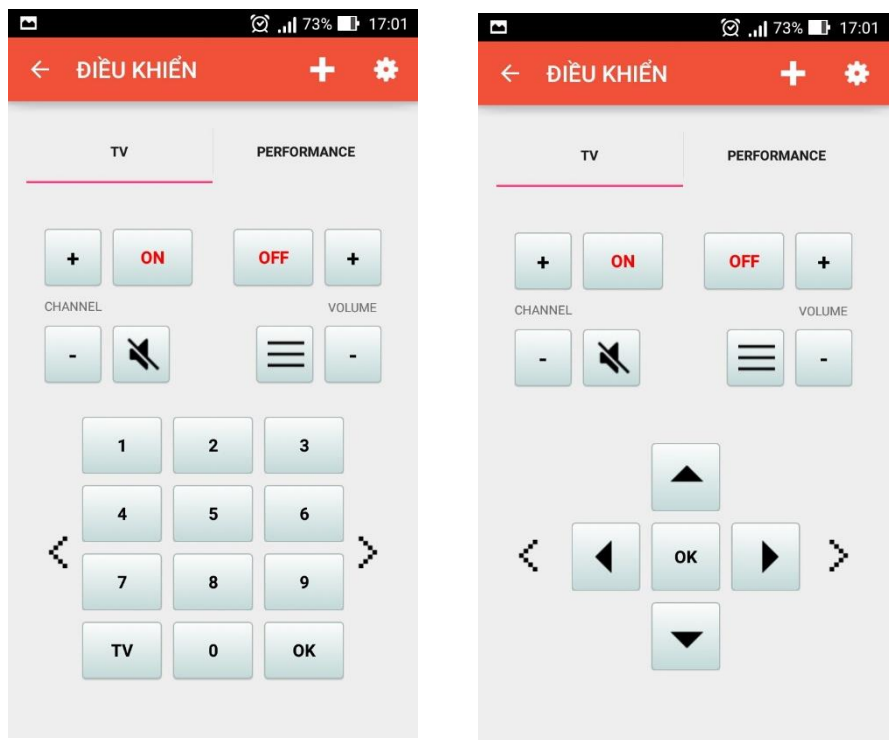
STT	Tên đối tượng	Kiểu	Chức năng	Ghi chú
1	txt_Model	TextView	Hiển thị tên hãng được chọn	
2	txt_SearchModel	EditText	Tìm kiếm mẫu	
3	lv_Model	ListView	Hiển thị danh sách mẫu	

- Các biến cố

STT	Biến cố	Xử lý
1	Nhập kí tự vào ô Search	Lọc ra danh sách mẫu có chứa kí tự được nhập.
2	Click mẫu trên listView	Chuyển qua màn hình điều khiển tương ứng.

IV.4.2.5. Màn hình điều khiển

❖ TV

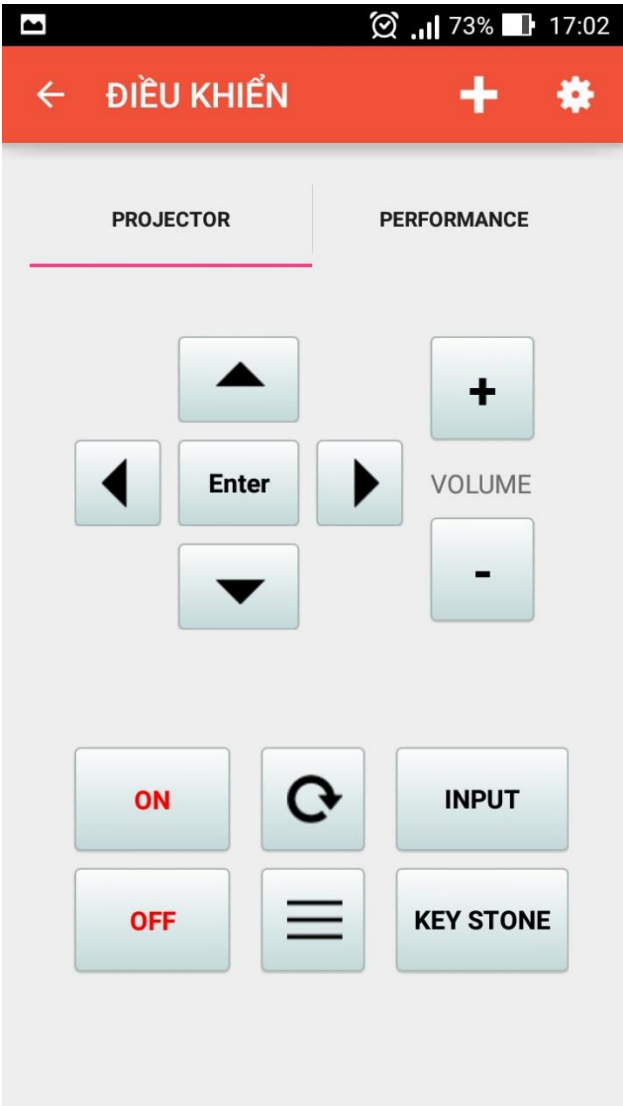


• Các đối tượng

STT	Tên đối tượng	Kiểu	Chức năng	Ghi chú
1	btn_tv_ChannelUp	Button	Tăng kênh	
2	btn_tv_ChannelDown	Button	Giảm kênh	
3	btn_tv_VolumeUp	Button	Tăng âm lượng	
4	btn_tv_VolumeDown	Button	Giảm âm lượng	
5	btn_tv_PowerOn	Button	Mở TV	
6	btn_tv_PowerOff	Button	Tắt TV	
7	btn_tv_Mute	Button	Tắt âm thanh	
8	btn_tv_Menu	Button	Hiển thị menu	

9	btn_tv_ChangeLayout1	Button	Chuyển đổi layout	
10	btn_tv_ChangeLayout2	Button	Chuyển đổi layout	
11	btn_tv_1	Button	Phím 1	
12	btn_tv_2	Button	Phím 2	
13	btn_tv_3	Button	Phím 3	
14	btn_tv_4	Button	Phím 4	
15	btn_tv_5	Button	Phím 5	
16	btn_tv_6	Button	Phím 6	
17	btn_tv_7	Button	Phím 7	
18	btn_tv_8	Button	Phím 8	
19	btn_tv_9	Button	Phím 9	
20	btn_tv_TV	Button	Hiện thông tin	
21	btn_tv_0	Button	Phím 0	
22	btn_tv_OK	Button	Chấp nhận lựa chọn	
23	btn_tv_UpArrow	Button	Di chuyển lên	
24	btn_tv_LeftArrow	Button	Di chuyển trái	
25	btn_tv_RightArrow	Button	Di chuyển phải	
26	btn_tv_DownArrow	Button	Di chuyển xuống	

❖ Projector

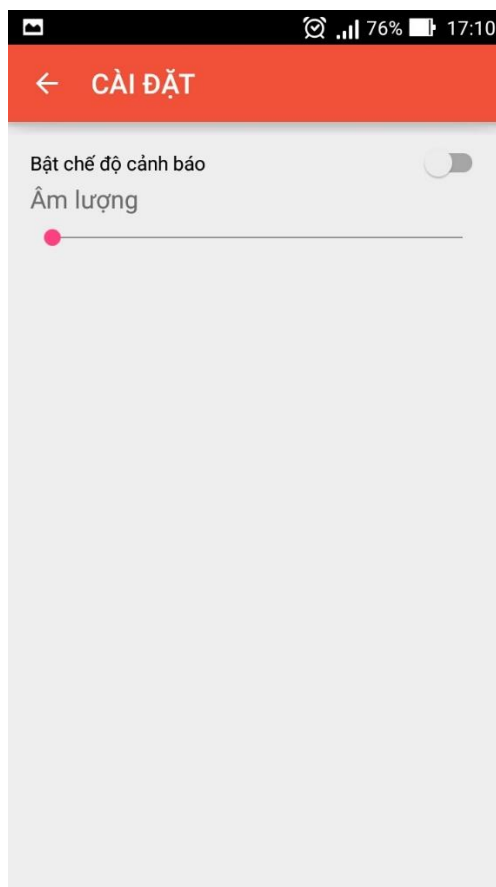


• Các đối tượng

STT	Tên đối tượng	Kiểu	Chức năng	Ghi chú
1	btn_pro_PowerOn	Button	Mở máy chiếu	
2	btn_pro_PowerOff	Button	Tắt máy chiếu	
3	btn_pro_VolumeUp	Button	Tăng âm lượng	

4	btn_pro_VolumeDown	Button	Giảm âm lượng	
5	btn_pro_Reset	Button		
6	btn_pro_Menu	Button		
7	btn_pro_Input	Button		
8	btn_pro_KeyStone	Button		
9	btn_pro_UpArrow	Button		
10	btn_pro_LeftArrow	Button		
11	btn_pro_RightArrow	Button		
12	btn_pro_DownArrow	Button		
13	btn_pro_Enter	Button		

#### IV.4.2.6. Màn hình cài đặt



## **IV.5. Cài đặt và thử nghiệm**

### **IV.5.1. Môi trường phát triển ứng dụng**

- Công cụ lập trình: Android Studio, Notepad++.
- Giả lập Android: Genymotion, Droid4x.
- Cơ sở dữ liệu: SQLite
- Quản lý cấu hình: Github.
- Môi trường Java.

### **IV.5.2. Môi trường triển khai ứng dụng**

- Hệ điều hành: Android.
- Thiết bị triển khai ứng dụng được hỗ trợ cổng hồng ngoại.

## **CHƯƠNG V: TỔNG KẾT**

### **V.1. Nhận xét**

#### **V.1.1. Ưu điểm**

- Giao diện đơn giản, dễ sử dụng.
- Các thao tác xử lý nhanh chóng và hiệu quả.
- Sử dụng SQLite cho kết nối cơ sở dữ liệu giúp truy xuất và cập nhật dữ liệu dễ dàng, linh hoạt hơn.
- Ứng dụng dễ quản lý, phát triển và bảo trì.
- Các chức năng đã hoàn thành hoạt động với hiệu quả tốt nhất.

#### **V.1.2. Nhược điểm**

- Chưa hỗ trợ một số chức năng cảnh báo và thống kê hoạt động của thiết bị.
- Số lượng thiết bị được hỗ trợ còn hạn chế.
- Giao diện chưa được thiết kế một cách tối ưu.

### **V.2. Hướng phát triển**

- Xây dựng thêm các chức năng cảnh báo, thống kê hoạt động của thiết bị.
- Cập nhật hỗ trợ nhiều loại thiết bị.
- Phát triển ứng dụng dựa trên nhiều công nghệ khác nhau.
- Triển khai ứng dụng thực tế và nhận phản hồi từ người dùng.

### **V.3. Kiến thức đạt được**

- Hiểu rõ cách thức hoạt động của các thiết bị điều khiển.
- Bổ sung kiến thức về nền tảng Android và cách xây dựng một ứng dụng Android.
- Trau dồi kiến thức ngôn ngữ lập trình Java hỗ trợ xây dựng ứng dụng.
- Biết thêm kiến thức về công nghệ IR áp dụng vào ứng dụng.
- Nâng cao các kỹ năng mềm: làm việc nhóm, quản lý thời gian,...

#### **V.4. Kết luận**

- Giao diện: Đã hoàn thành các giao diện chính, chưa được trau chuốt kỹ lưỡng nhưng đơn giản, dễ sử dụng.
- Chức năng: Đã hoàn thành các chức năng điều khiển và hỗ trợ một số thiết bị như TV, máy chiếu. Số lượng thiết bị được hỗ trợ còn hạn chế và một số chức năng cảnh báo, thống kê chưa được hoàn thành.
- Lưu trữ: Lưu trữ đầy đủ các thông tin điều khiển, quản lý hoạt động cần thiết cho ứng dụng.



## TÀI LIỆU THAM KHẢO

- [1] Beginning Android 4 Application Development, Wei-Meng Lee, 03/2012
- [2] Android™ A programmer's Guide, J.F. DiMarzio, 2008
- [3] <http://techdocs.altium.com/display/FPGA/Infrared+Communication+Concepts>
- [4] <http://www.vishay.com/ir-emitting-diodes/>
- [5] <https://irdb.globalcache.com/>
- [6] <https://github.com/fishjord/IRDARemote>
- [7] [https://en.wikipedia.org/wiki/Consumer\\_IR](https://en.wikipedia.org/wiki/Consumer_IR)
- [8] <https://developer.android.com/reference/android/hardware/ConsumerIrManager.html>
- [9] <https://developer.sony.com/develop/smartphones-and-tablets/camera-add-on-small-apps-or-ir-remote-apis/about/ir-remote-api/>
- [10] <https://github.com/timnew/AndroidInfrared>