

Mục lục

LỜI MỞ ĐẦU.....	iii
CHƯƠNG 1. GIỚI THIỆU ANDROID	1
1.1. Khái niệm về Android	1
1.1.1. Android khác với các hệ điều hành chạy trên thiết bị di động khác.....	1
1.1.2. Đặc tính mở của Android.....	2
1.2. Kiến trúc của Android	2
1.2.1. Android Platform.....	3
1.2.2. Tầng Linux Kernel	3
1.2.3. Tầng Native Libraries	3
1.2.4. Tầng Runtime	4
1.2.5. Tầng Application Framework	4
CHƯƠNG 2. MÔI TRƯỜNG VÀ CÔNG CỤ LẬP TRÌNH	6
2.1. Giới thiệu công cụ.....	6
2.2. Cài đặt Android Studio	6
2.3. Tạo ứng dụng đầu tiên	9
2.4. Thành phần quan trọng trong một Android Project.....	15
2.5. Chu kỳ sống của ứng dụng Android.....	15
2.5.1. Chu kỳ sống	15
2.5.2. Activity Stack.....	15
2.5.3. Các trạng thái của chu kỳ sống	16
2.5.4. Các hàm thực thi.....	17
2.6. Các Layout trong Android.....	17
2.6.1. Frame Layout	17
2.6.2. LinearLayout	19
2.6.3. Relative Layout	20
2.6.4. TableLayout	22
2.6.5. GridLayout	23
CHƯƠNG 3. GIỚI THIỆU VỀ SAMSUNG KNOX	25
3.1. Tổng quan về Samsung Knox	25
3.2. Knox SDK	25

CHƯƠNG 4. MÔ TẢ VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	27
4.1. Mô tả và yêu cầu hệ thống	27
4.1.1. Mô tả hệ thống	27
4.1.2. Yêu cầu hệ thống.....	27
4.2. Phân tích thiết kế hệ thống.....	28
4.2.1. Biểu đồ use case tổng quát.....	28
4.2.2. Phân rã use case.....	29
4.2.3. Biểu đồ trạng thái	30
CHƯƠNG 5. XÂY DỰNG ỨNG DỤNG.....	34
5.1. Giao diện khởi chạy và xin quyền Admin	34
5.2. Giao diện đăng ký.....	36
5.3. Giao diện trang chủ	39
5.4. Giao diện thay đổi mật khẩu và thẻ NFC.....	40
5.5. Giao diện đặt thời gian phát video.....	41
5.6. Giao diện phát video.....	42
KẾT LUẬN.....	43
NHẬN XÉT.....	44
TÀI LIỆU THAM KHẢO.....	45

LỜI MỞ ĐẦU

Việc Samsung đang phát triển mạnh mẽ ở Việt Nam, các mẫu điện thoại được Samsung sản xuất ngày càng nhiều. Do đó việc trưng bày các mẫu điện thoại mới cho người dùng trải nghiệm là rất nhiều, việc quản lý và bảo vệ việc tránh đánh cắp điện thoại là một vấn đề đang được đặt ra.

Hệ thống an ninh được cài đặt để ngăn chặn trường hợp trộm cắp và sặc ố định các thiết bị được trưng bày trong cửa hàng. Hệ thống an ninh có thể rung chuông báo động cho nhân viên cửa hàng để ngăn chặn trường hợp trộm cắp và có yếu tố hình thức nhỏ hơn để ít ảnh hưởng đến trải nghiệm của khách hàng. Tuy nhiên, trong các cửa hàng vận hành và bán lẻ, hệ thống bảo mật hình thức công kênh và tốn kém chi phí được triển khai. Do đó, nó không chỉ cản trở trải nghiệm của khách hàng mà còn hiển thị nhất quán trong các cửa hàng và gây tốn kém chi phí. Hơn nữa, hệ thống an ninh hiện tại có sự phụ thuộc vào các nhà cung cấp bảo mật bằng hộp điều khiển.

Ứng dụng an ninh và phát video quảng cáo khi thiết bị ở chế độ rảnh là cần thiết và chúng nên được tích hợp vào làm một kèm theo đó là những điều chưa tối ưu như ở trên nên em chọn đề tài: *“Phát triển ứng dụng theo dõi và bảo vệ các mẫu điện thoại thông minh của Samsung tại các cửa hàng bán lẻ”*.

Để hoàn thành được đợt thực tập này, ngoài sự nỗ lực của bản thân, em xin chân thành gửi lời cảm ơn sâu sắc đến Trung tâm nghiên cứu và phát triển điện thoại di động Samsung tại Việt Nam – SVMC, cán bộ hướng dẫn tại Công ty cùng Ban lãnh đạo trường Đại học Công nghệ thông tin và Truyền thông Thái Nguyên và các thầy cô trong trường, đặc biệt là thầy **TS. Nguyễn Văn Núi** đã tận tình hướng dẫn, giúp đỡ và tạo điều kiện tốt nhất cho em kể từ khi bắt đầu nhận đề tài đến khi hoàn thành đề tài này.

Em xin chân thành cảm ơn!

CHƯƠNG 1. GIỚI THIỆU ANDROID

Như chúng ta biết, hiện tại đã có hơn nửa nhân loại sử dụng máy di động để thoại và giao tiếp qua các mạng không dây. Con số 3 tỷ người này sẽ còn tăng lên và máy di động càng ngày càng "thông minh" với nhiều chức năng và dịch vụ rất hấp dẫn, cho nên thị trường máy di động thông minh sẽ vượt xa máy vi tính trong một tương lai rất gần... Vì thế việc lập trình trên thiết bị di động ngày càng phổ biến và phát triển rất mạnh mẽ. Từ nền tảng mã nguồn mở, Google đã cho ra mắt Android chạy trên các thiết bị di động. Android có rất nhiều công cụ và dụng cụ miễn phí để nghiên cứu và phát triển phần mềm trên nền tảng của nó. Tài liệu này sẽ giúp chúng ta tìm hiểu về Android và cách viết một ứng dụng trên nền tảng này.

1.1. Khái niệm về Android

Trước hết Android là nền tảng phần mềm dựa trên mã nguồn mở của Linux OS (Kernel 2.6) cho máy di động và những phần mềm trung gian (middleware) để hỗ trợ các ứng dụng mà người sử dụng cần đến. Một cách định nghĩa không quá chuyên môn thì có thể coi Android là một nền tảng mở cho thiết bị di động của Google (gồm hệ điều hành, middleware và một số ứng dụng cơ bản). Android sẽ đương đầu với một số hệ điều hành (viết tắt là HDH) dành cho thiết bị di động khác đang hâm nóng thị trường như Windows Mobile, OS X (iPhone).

Có thể nói một cách nôm na rằng Android là một HDH chạy trên thiết bị di động, Cũng giống như Windows, Linux hay Mac chạy trên máy vi tính vậy.

1.1.1. Android khác với các hệ điều hành chạy trên thiết bị di động khác

Android đã thu hút được sự chú ý của giới công nghệ khác toàn cầu khi đưa con của Google sử dụng giấy phép mã nguồn mở. Đó là một sản phẩm kết tinh từ ý tưởng của Khối Liên minh thiết bị cầm tay do Google dẫn đầu, gồm 34 thành viên với các

công ty hàng đầu về công nghệ và di động toàn cầu như Qualcomm, Intel, Motorola, và LG Electronics, các nhà mạng như T-Mobile, Sprint Nextel à China Mobile.

Các nhà phát triển có thể sử dụng miễn phí bộ Kit Android Software Development để xây dựng các ứng dụng của mình.

1.1.2. Đặc tính mở của Android

Android được xây dựng để cho phép các nhà phát triển tạo ra các ứng dụng di động hấp dẫn, tận dụng tất cả tính năng một chiếc điện thoại đã cung cấp.

Android được xây dựng để mở. Ví dụ, một ứng dụng có thể gọi bất kỳ chức năng của lõi điện thoại như thực hiện cuộc gọi gửi tin nhắn văn bản, chụp ảnh... Cho phép các nhà phát triển tạo ra nhiều ứng dụng phong phú hơn cho người dùng (điều này hiện chưa có trên Windows Phone 7 của Microsoft). Android được xây dựng trên mã nguồn mở của Linux Kernel. Hơn nữa, nó sử dụng một máy ảo tùy chỉnh được thiết kế để tối ưu hóa bộ nhớ và tài nguyên phần cứng trong môi trường di động.

Android cung cấp truy cập đến một loạt các thư viện công cụ hữu ích và có thể được sử dụng để xây dựng các ứng dụng phong phú. Ví dụ, Android cho phép các thiết bị giao tiếp với nhau tạo điều kiện cho đồng đẳng rich-to-peer trong ứng dụng xã hội. Ngoài ra, Android bao gồm một tập hợp đầy đủ công cụ đã được xây dựng công phu, với việc cung cấp nền tảng phát triển, với năng suất cao và cái nhìn sâu vào các ứng dụng.

1.2. Kiến trúc của Android

“Understanding Android” là cách mà ta tiếp cận lập trình Android và thấu hiểu kiến trúc hệ thống của nó. Chúng ta có thể không cần hiểu rõ cấu trúc của một hệ điều hành nhưng chúng ta vẫn có thể lập trình một ứng dụng trên hệ điều hành đó, đây là một điều mà nhà sản xuất muốn khi release SDK với một framework có sẵn của họ. Như chúng ta biết điều này cũng có mặt tốt và xấu. Framework là một tầng cao cấp dành cho lập trình viên, nó đều có giới hạn của nó, chúng ta chỉ có thể lập

trình những ứng dụng phổ biến nhưng không nên tiến tới những ứng dụng cao cấp đi sâu vào hệ thống của hệ điều hành.

1.2.1. Android Platform

Bao gồm hệ điều hành Android đầy đủ tính năng, các ứng dụng và các tầng trung gian để developer có thể mở rộng, tùy chỉnh hoặc thêm vào các component của họ.

Có 4 tầng cơ bản trong hệ điều hành Android:

- Application Framework
- Android Runtime
- Native Libraries
- Linux Kernel

Mỗi tầng làm việc đều nhờ sự giúp đỡ của tầng bên dưới.

1.2.2. Tầng Linux Kernel

Đây là tầng nhân của hệ điều hành Android, mọi việc xử lý của hệ thống đều phải thông qua tầng này. Linux Kernel cung cấp các trình điều khiển thiết bị phần cứng (Driver) như camera, USB, Wifi, Bluetooth, Display, Power Management...

Android dựa trên Linux phiên bản 2.6 lựa chọn tính năng cốt lõi như bảo mật, quản lý bộ nhớ, quản lý tiến trình, mạng stack và các trình điều khiển phần cứng. Kernel hoạt động như một lớp trừu tượng giữa phần cứng và phần mềm còn lại của hệ thống

1.2.3. Tầng Native Libraries

- System C library: Có nguồn gốc từ hệ thống thư viện chuẩn C (libc), điều chỉnh các thiết bị nhúng trên Linux.

- Media Libraries: Mở rộng từ PacketVideo's OpenCORE. Là thư viện hỗ trợ playback và recording của nhiều định dạng video và image phổ biến như MPEG4, H.264, MP3, AAC, JPG và PNG...
- Surface Manager: Quản lý việc hiển thị và kết hợp đồ họa 2D và 3D.
- LibWebCore: Android dung lại Webkit Engine cho việc render trình duyệt mặc định của hệ điều hành Android browser và cho định dạng nhúng như HTML.
- SGL: 2D engine
- 3D libraries: Thư viện 3D dựa trên OpenGL ES 1.0 API, có nâng cấp tăng tốc "hardware 3D acceleration"
- FreeType: Render bitmap và vector font
- SQLite: Quản lý database của ứng dụng

1.2.4. Tầng Runtime

Một ứng dụng Android chạy trên một process riêng của Dalvik VM (Máy ảo). Dalvik được viết để chạy nhiều máy ảo cùng một lúc một cách hiệu quả trên cùng một thiết bị.

Máy ảo Dalvik thực thi các file mang định dạng `**.dex` (Dalvik Executable), định dạng này đã được tối ưu hóa chỉ chiếm một vùng nhớ nhỏ vừa đủ và nhỏ nhất có thể. Máy ảo chạy các class (đã được compile trước đó bởi 1 trình biên dịch ngôn ngữ Java), sở dĩ máy ảo chạy được các class này là nhờ chương trình DEX tool đã convert các class sang định dạng `**.dex`.

1.2.5. Tầng Application Framework

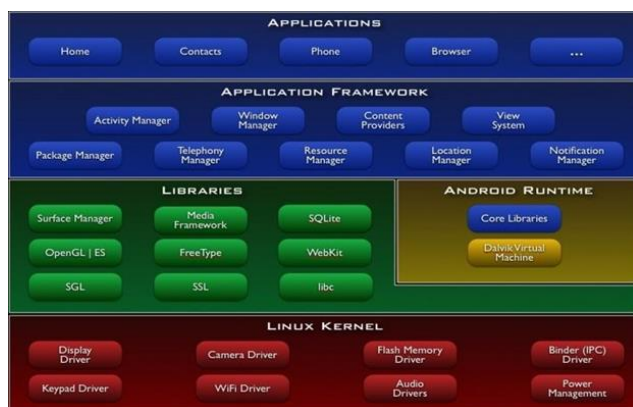
Đây là tầng mà Google xây dựng cho phép các Developer để phát triển các ứng dụng của họ trên Android, chỉ bằng cách gọi các API có sẵn mà Google đã viết để sử dụng các tính năng của phần cứng mà không cần hiểu cấu trúc bên dưới.

Bằng cách cung cấp một nền tảng phát triển mở, Android 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 thức, thêm các thông báo để các thanh trạng thái và nhiều hơn nữa.

Tất cả các ứng dụng thường gồm một bộ các dịch vụ và hệ thống cơ bản sau:

- View UI: dùng để xây dựng layout của ứng dụng bao gồm: listview, text field, button, dialog, from...
- Content Providers: cho phép các ứng dụng có thể truy cập dữ liệu từ các ứng dụng khác (như ứng dụng của ta có thể lấy thông tin Contacts của điện thoại Android), hoặc để chia sẻ dữ liệu của riêng ứng dụng.
- Resource Manager: cung cấp cách thức truy cập đến non-code resources như các asset, graphic, image, music, video...
- Notification Manager: cho phép tất cả các ứng dụng hiển thị thông báo của mình trên hệ điều hành.
- Activity Manager: quản lý vòng đời của các ứng dụng.

Ở góc nhìn của người dùng ta có thêm tầng Application (là tầng ứng dụng do chúng ta viết). Sau đây là sơ đồ tổng quát như hình dưới.



Hình 1. 1 Kiến trúc Android

CHƯƠNG 2. MÔI TRƯỜNG VÀ CÔNG CỤ LẬP TRÌNH

2.1. Giới thiệu công cụ

Trong chương này sẽ giới thiệu các công cụ lập trình cho Android (Android Development Tools). Chúng ta sẽ dần làm quen với Android Studio.

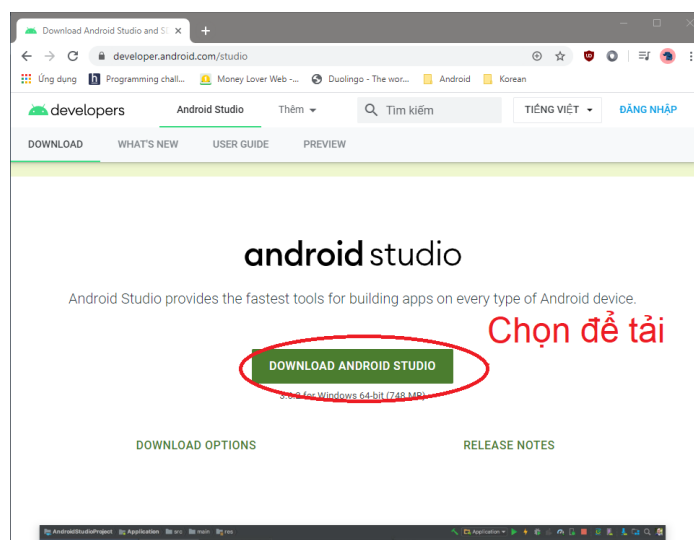
Android Studio là môi trường phát triển tích hợp IDE chính thức dành cho phát triển nền tảng Android.

Nó được ra mắt vào ngày 16 tháng 5 năm 2013, được phát hành miễn phí theo giấy phép Apache Licence 2.0.

Android Studio được phát triển dựa trên phần mềm IntelliJ IDEA của JetBrains, Android Studio được thiết kế đặc biệt để phát triển ứng dụng Android. Nó hỗ trợ các hệ điều hành Windows, Mac OS và Linux. Là IDE chính thức của Google để phát triển ứng dụng Android gốc để thay thế cho Android Development Tools dựa trên Eclipse.

2.2. Cài đặt Android Studio

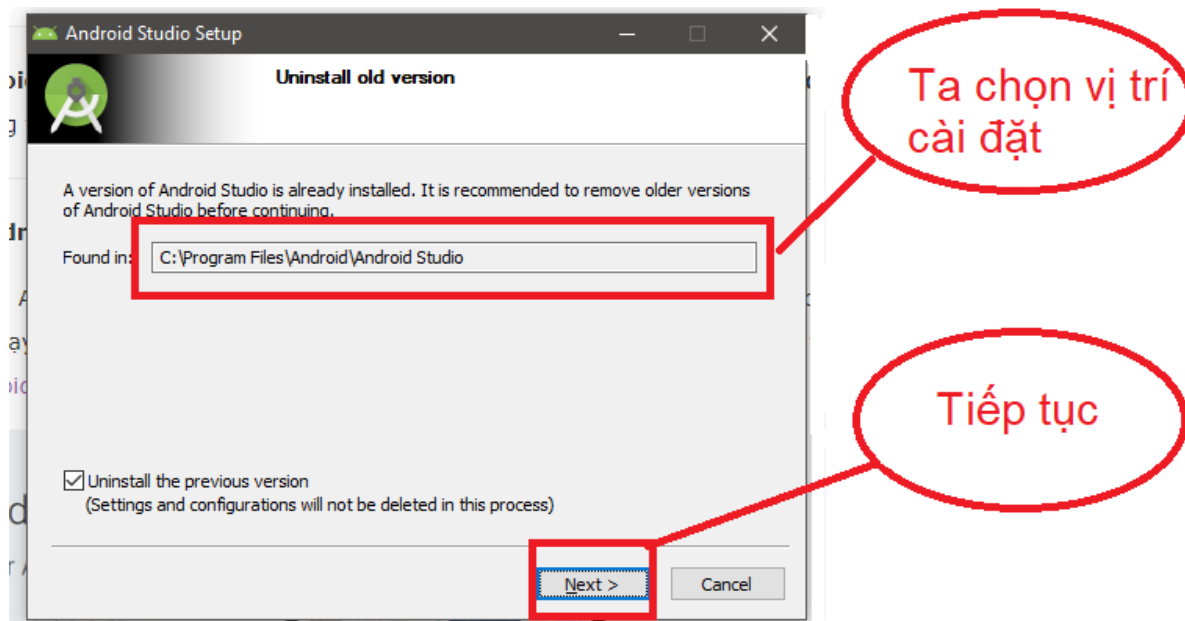
Bước 1: Chúng ta truy cập vào trang chủ của Android Studio để tải bản mới nhất <https://developer.android.com/studio>.



Hình 2. 1 Trang chủ tải Android Studio

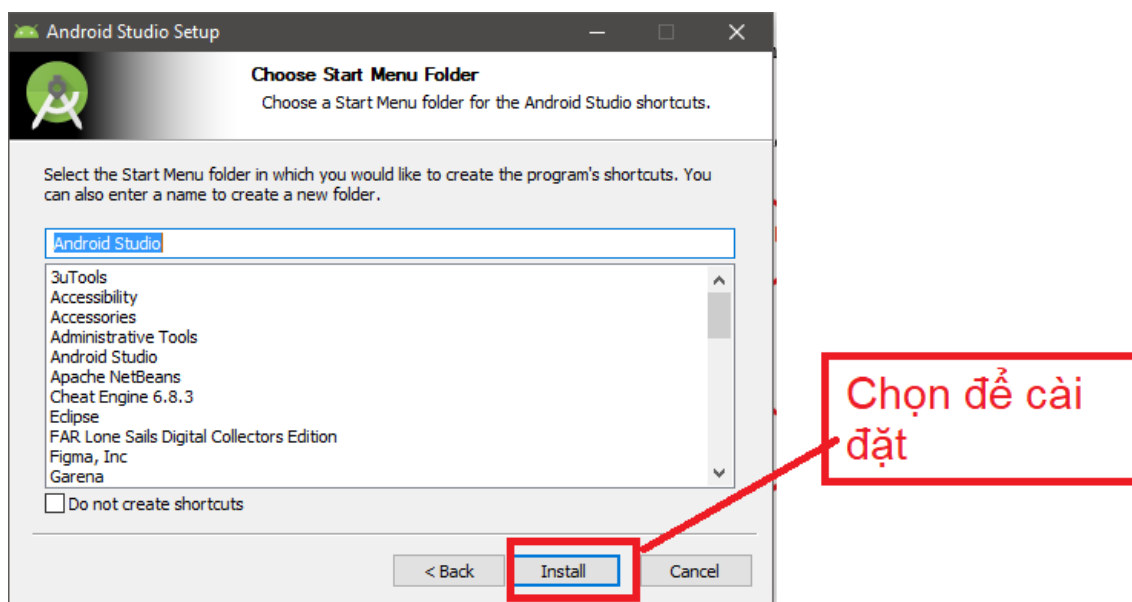
Bước 2: Tiến hành cài đặt trên Windows. Sau khi tải xuống sau tại thực hiện việc chạy file vừa tải xuống.

Ta có thể chọn vị trí lưu cài đặt Android Studio. Sau đó ấn **Next** để tiếp tục.



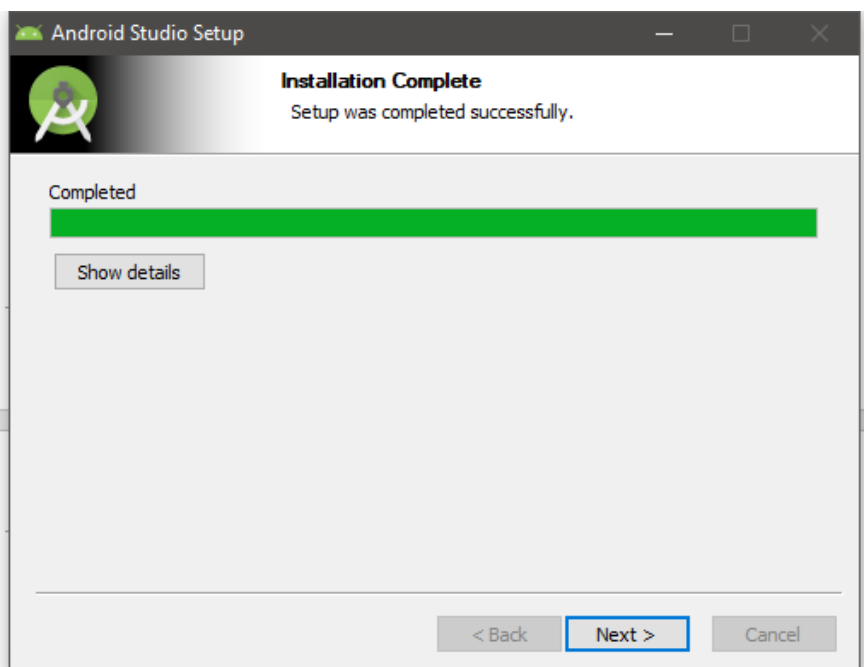
Hình 2. 2 Chọn thư mục cài đặt Android Studio

Bước 3: Sau đó ta chọn Install để cài đặt. Và đợi Android Studio tự cài đặt.



Hình 2. 3 Chọn Install để cài đặt Android Studio

Bước 4: Hoàn thành việc cài đặt. Ta chọn NEXT để hoàn thành khi Android Studio đã cài đặt xong.



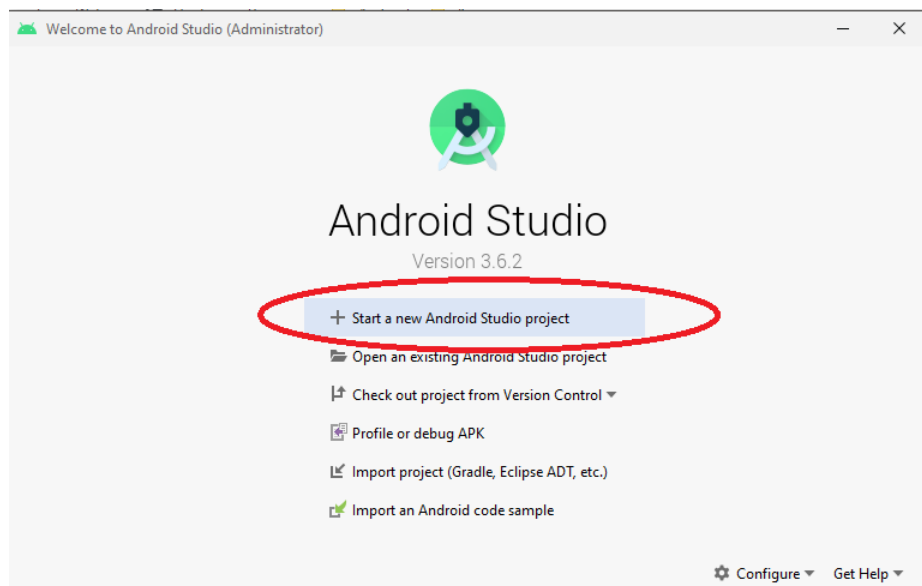
Hình 2. 4 Hoàn thành việc cài đặt Android Studio



Hình 2. 5 Đã hoàn thành việc cài đặt Android Studio

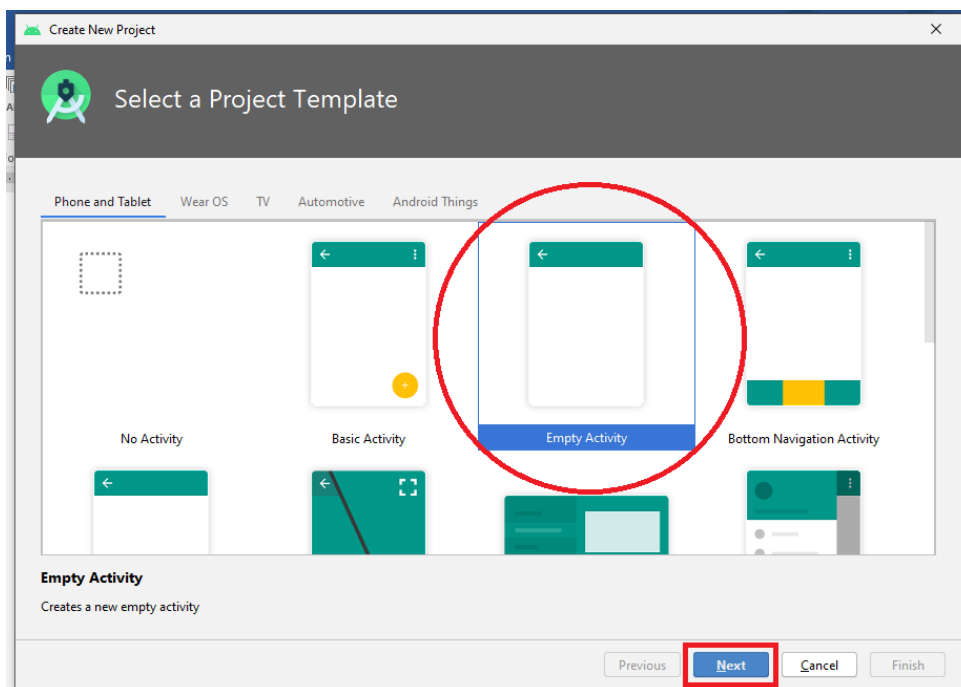
2.3. Tạo ứng dụng đầu tiên

Bước 1: Sau khi mở ứng dụng. Ta thấy giao diện bắt đầu của Android Studio. Ta chọn “Start a new Android Studio project”. Để bắt đầu tạo ứng dụng.



Hình 2. 6 Bắt đầu tạo project mới

Bước 2 : Ta chọn Empty Activity. Sau đó chọn Next.



Hình 2. 7 Bắt đầu tạo một Empty Activity

Bước 3: Như hình minh họa dưới.

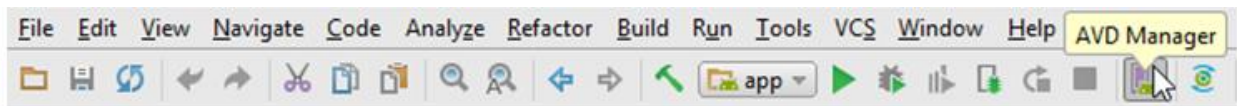
1. Là tên của dự án
2. Tên của gói dự án
3. Nơi lưu trữ dự án
4. Là ngôn ngữ, hiện tại thì sẽ có 2 ngôn ngữ lập trình cho Android là Java và Kotlin
5. Là phiên bản thấp nhất sử dụng SDK.

Sau khi chọn xong chúng ta chọn Next để tiếp tục.

The screenshot shows the 'Configure Your Project' dialog in Android Studio. The dialog has a title bar 'Create New Project' and a close button. The main area is titled 'Configure Your Project'. On the left, there is a preview of an 'Empty Activity' with a back arrow. On the right, there are five input fields, each highlighted with a red box and a number: 1. Name: 'My Application' 2. Package name: 'com.example.myapplication' 3. Save location: 'D:\Android\MyApplication' 4. Language: 'Java' 5. Minimum SDK: 'API 26: Android 8.0 (Oreo)' Below these fields, there is an information icon and text: 'Your app will run on approximately 66.6% of devices.' and a link 'Help me choose'. There is also a checkbox 'Use legacy android.support libraries' with a question mark. At the bottom right, there are four buttons: 'Previous', 'Next', 'Cancel', and 'Finish'. The 'Finish' button is highlighted with a red box.

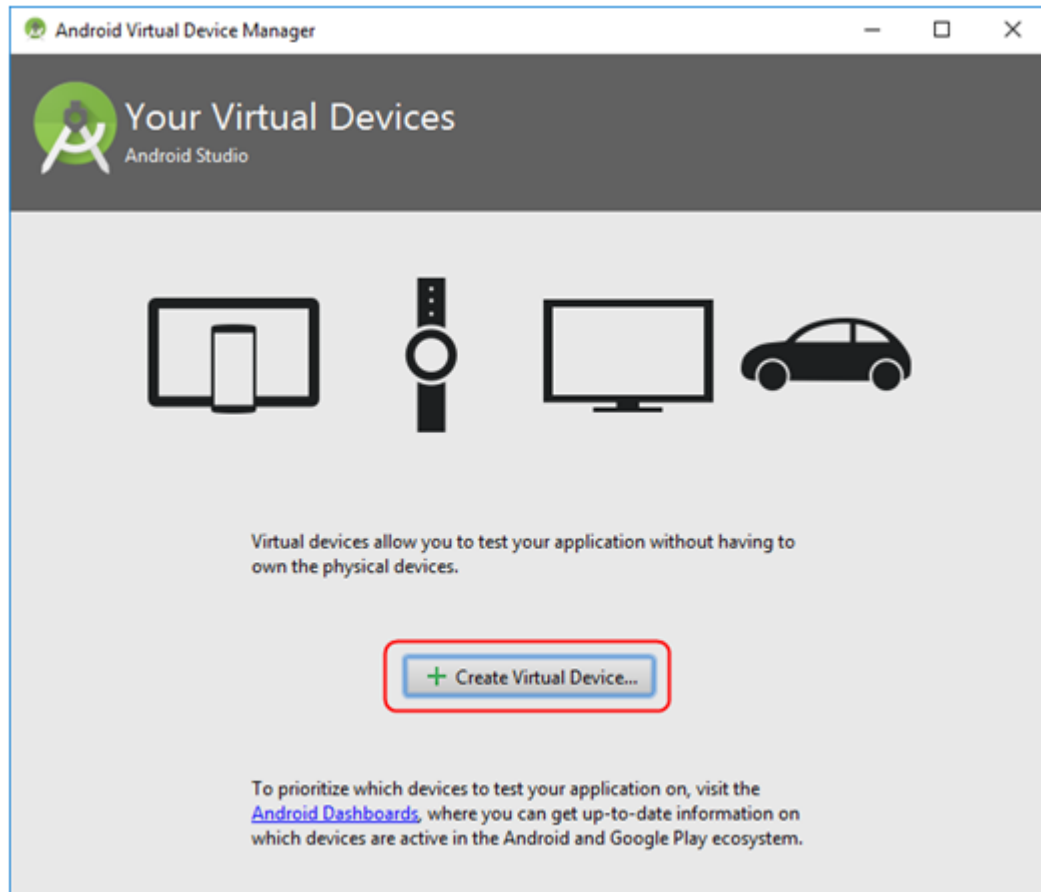
Hình 2. 8 Mô tả việc chọn phiên bản, ngôn ngữ lập trình và đặt tên dự án

Bước 4: Tạo máy ảo. Ta chọn vào biểu tượng AVD Manager trên thanh Toolbar.



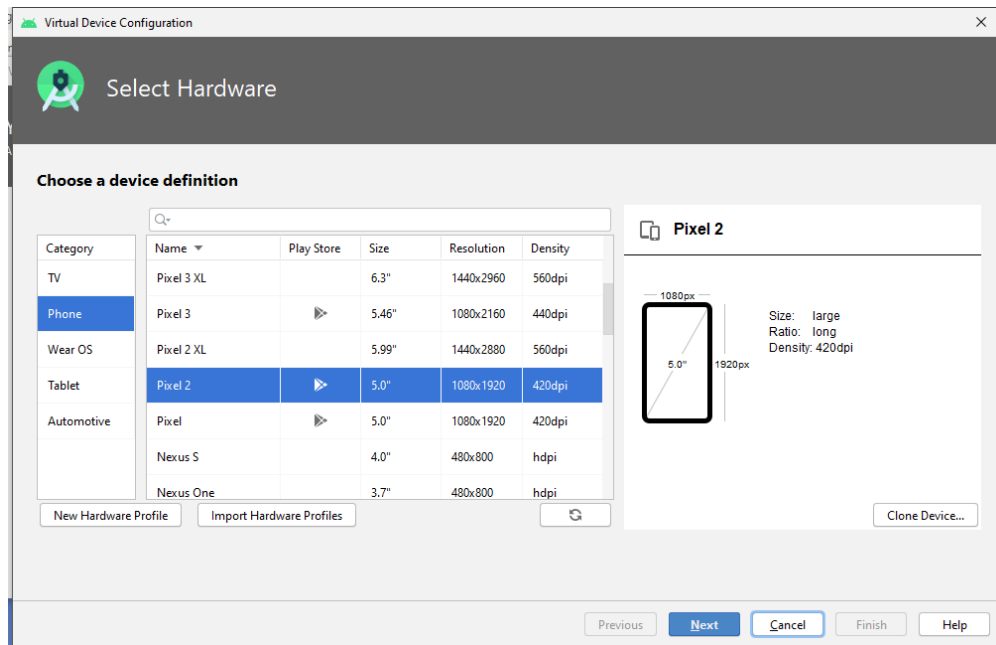
Hình 2. 9 Chọn Manager AVD

Bước 5: Chọn Create Virtual Device. Để bắt đầu tạo máy ảo.



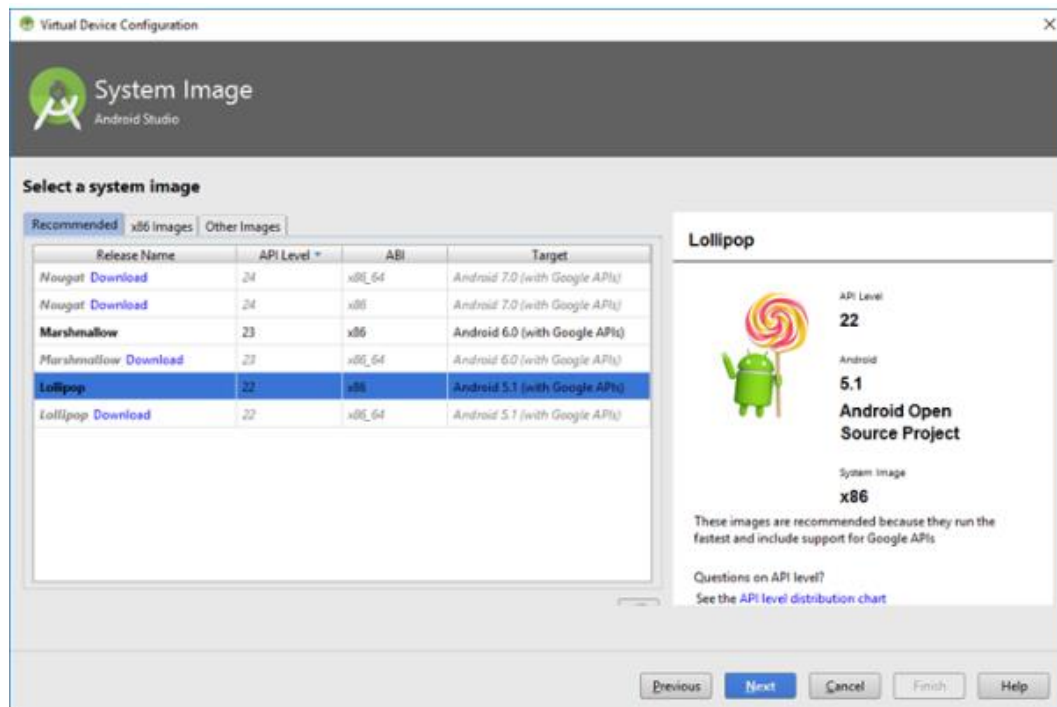
Hình 2. 10 Create Virtual Device

Bước 6: Chọn máy ảo muốn tạo. Sau đó ta ấn “Next”.




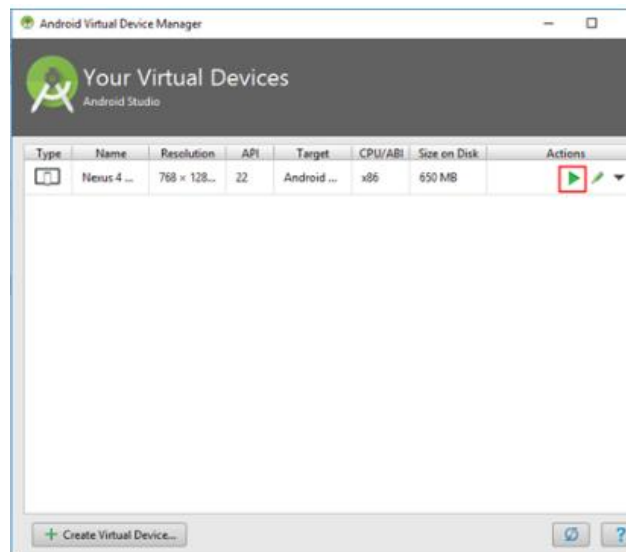
Hình 2. 11 Chọn máy ảo

Bước 7: Chọn hệ điều hành cho máy ảo.



Hình 2. 12 Chọn hệ điều hành cho máy ảo.


Bước 8: Chọn vào  để bắt đầu khởi động máy ảo.



Hình 2. 13 Danh sách máy ảo



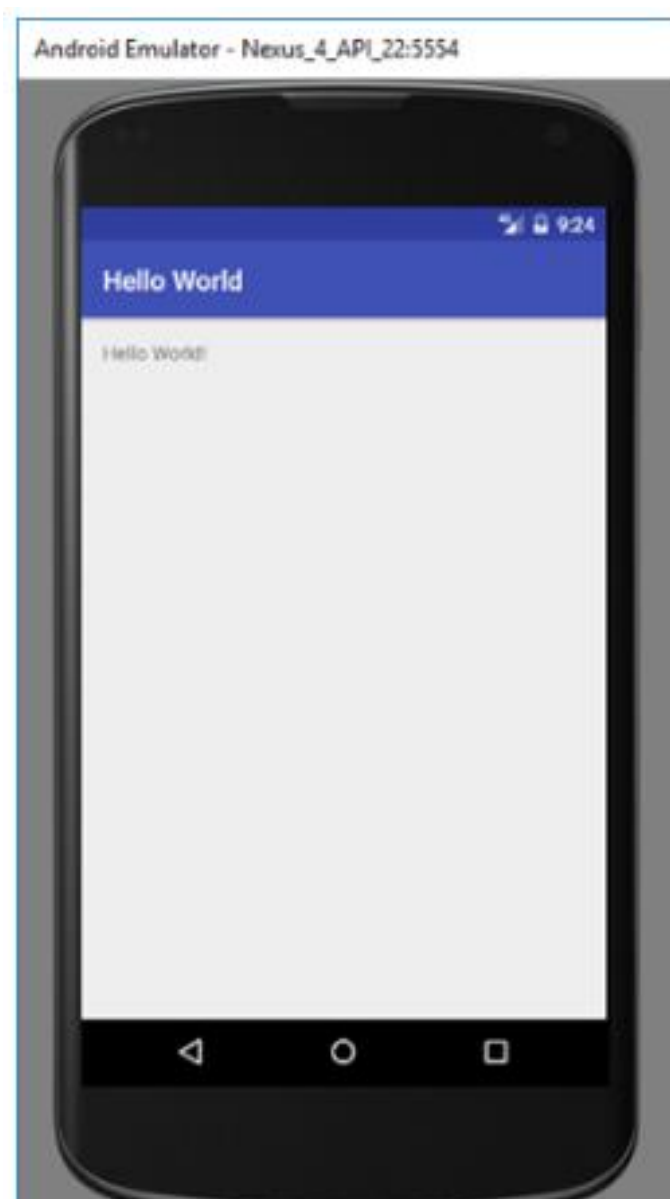
Hình 2. 14 Máy ảo đã khởi động xong

Bước 9: Giờ ta có thể tiến hành Build app bằng cách chọn vào biểu tượng  để bắt đầu Build.



Hình 2. 15 Toolbar Build

Sau khi Build và cài đặt xong ta sẽ có kết quả như hình bên dưới.



Hình 2. 16 Chương trình đầu tiên

2.4. Thành phần quan trọng trong một Android Project

- Activity: Đây là một lớp khởi tạo giao diện ứng dụng nội bộ trên Android.
- Service: Cung cấp các dịch vụ liên quan đến client/service. Một Service sẽ chạy ngầm bên dưới.
- Broadcast receiver: đây là một ứng dụng chạy ngầm dùng để đọc và nhận thông tin trên UI, ví như cập nhật pin, ngày giờ...
- Content Provider: Cung cấp chức năng truy vấn dữ liệu giữa các ứng dụng của Android
- Intent: Là nền tảng để truyền tải các thông báo. Intent được sử dụng để gửi các thông báo đi nhằm khởi tạo 1 Activity hay Service để thực hiện công việc mà chúng mong muốn.

2.5. Chu kỳ số của ứng dụng Android

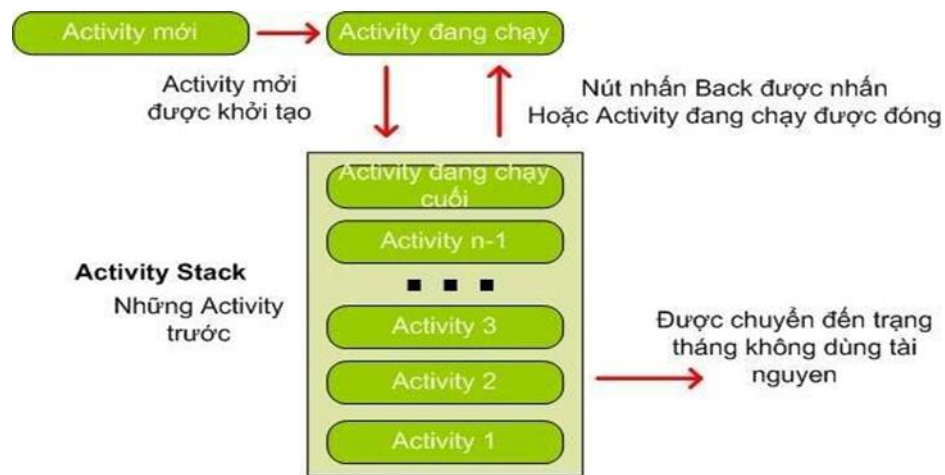
2.5.1. Chu kỳ sống

Các thành phần ứng dụng có một chu kỳ sống từ lúc khởi tạo và đến thời điểm kết thúc.

2.5.2. Activity Stack

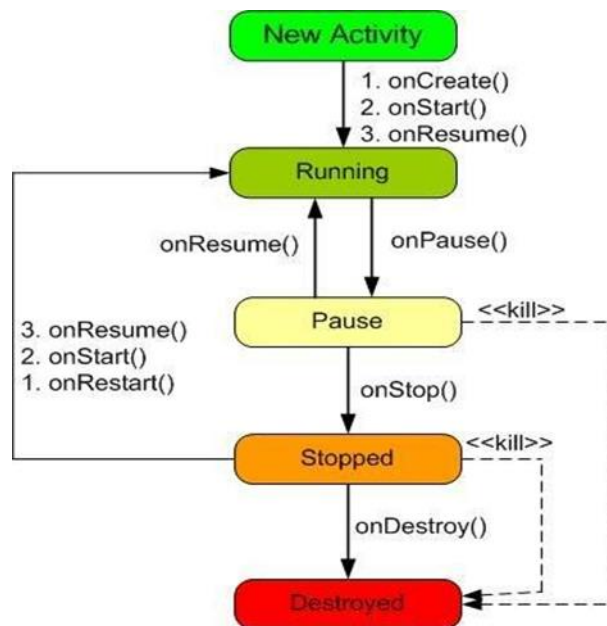
Bên trong hệ thống các Activity được quản lý như một activity stack. Khi một activity mới được chạy nó được đặt ở đỉnh của stack và trở thành activity đang chạy. Activity trước sẽ ở bên dưới activity mới và sẽ không thấy trong suốt quá trình activity mới tồn tại

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



Hình 2. 17 Activity Stack

2.5.3. Các trạng thái của chu kỳ sống



Hình 2. 18 Chu kỳ sống của Activity

Một Activity chủ yếu có 4 chu kỳ chính là:

- Active hoặc running: Khi Active là được chạy trên màn hình. Activity này tập trung vào những thao tác của người dùng trên ứng dụng.

- Paused: là Activity được tạm dừng khi mất focus nhưng người dùng vẫn trông thấy.
- Stopped: Nếu nó hoàn toàn bị bao phủ bởi một Activity khác. Nó vẫn có trạng thái và thông tin thành viên trong nó. Người dùng không thấy được nó và thường bị loại bỏ trong trường hợp hệ thống cần vùng nhớ cho tác vụ khác.
- Killed: Khi hệ thống thiếu vùng nhớ nó sẽ giải phóng các tiến trình theo nguyên tắc ưu tiên. Các Activity ở trạng thái Stop hay Pause cũng có thể bị giải phóng.

2.5.4. Các hàm thực thi

- onCreate(...): hàm này được gọi khi lớp Activity được khởi tạo, dùng để thiết lập giao diện ứng dụng và thực thi những thao tác cơ bản.
- onStart(): hàm này được gọi khi lớp ứng dụng xuất hiện trên màn hình.
- onResume(): hàm được gọi ngay sau onStart hoặc khi người dùng focus ứng dụng, hàm này sẽ đưa ứng dụng lên top màn hình.
- onPause(): hàm được gọi khi hệ thống đang focus đến 1 activity trước đó.
- onStop(): hàm được gọi khi một activity khác được khởi động và focus.
- onRestart(): được gọi khi ứng dụng chuyển sang onStop(), nhưng muốn khởi động lại bằng onStart().

2.6. Các Layout trong Android

2.6.1. Frame Layout

Frame Layout là một dạng Layout cơ bản nhất khi gắn các View lên layout này thì nó sẽ luôn giữ các View này ở phía góc trái màn hình và không cho chúng thay đổi vị trí của chúng. Các view sau đưa vào sau sẽ đè lên view ở trước trừ khi bạn thiết lập “transparent” cho view sau đó.

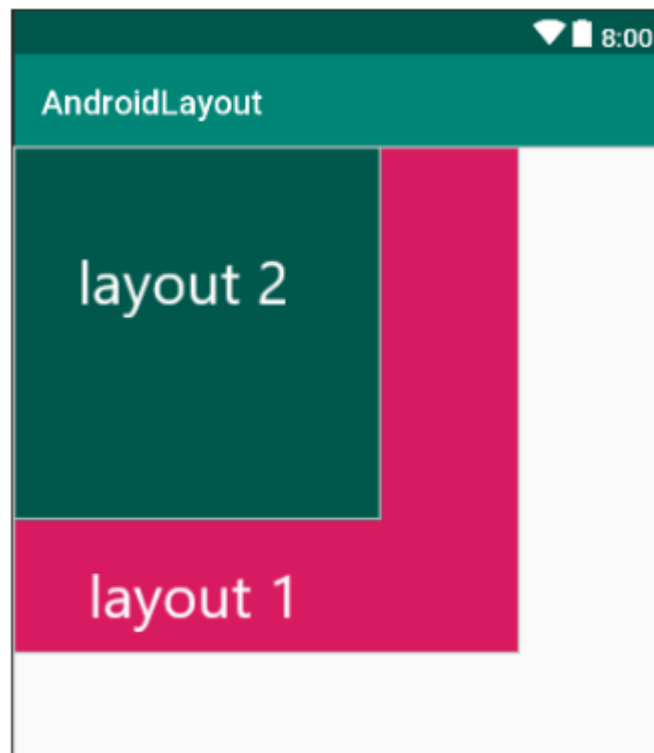
```

<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:background="@color/colorAccent"
        android:text="TextView" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="218dp"
        android:layout_height="221dp"
        android:background="@color/colorPrimaryDark" />
</FrameLayout>

```



Hình 2. 19 Minh họa FrameLayout

2.6.2. *LinearLayout*

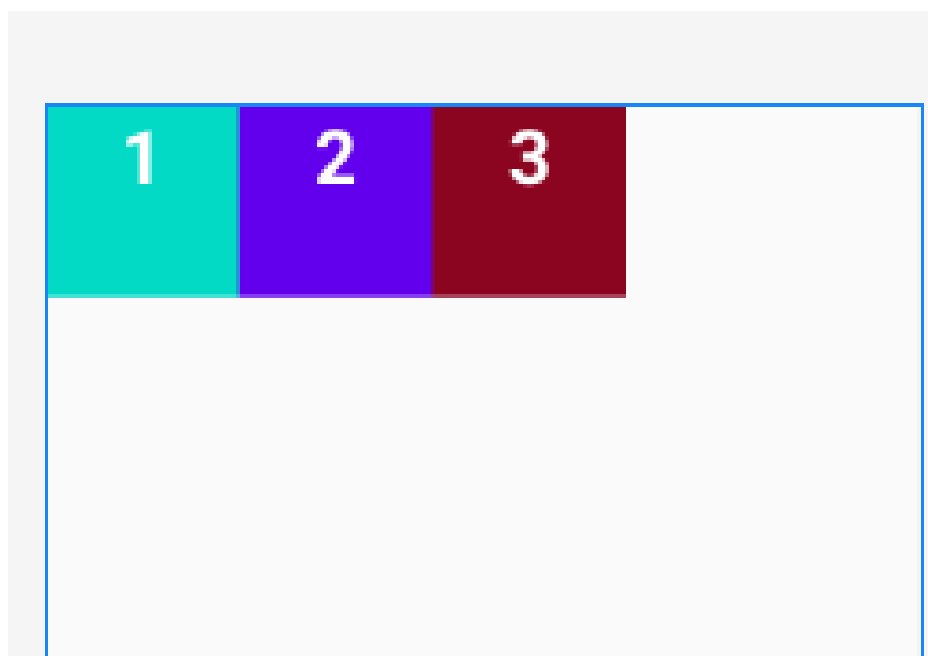
LinearLayout là loại layout thường được sử dụng. Được bố trí theo dạng khối và không đè lên nhau. LinearLayout được bố trí theo hai bố cục “*Vertical Orientation*” bố trí theo chiều dọc và “*Horizontal Orientation*” bố trí theo chiều ngang.

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:text="1"
        android:textColor="#fff"
        android:textSize="15pt"
        android:textAlignment="center"
        android:textStyle="bold"
        android:background="@color/colorAccent" />

    <TextView
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:text="2"
        android:textColor="#fff"
        android:textSize="15pt"
        android:textAlignment="center"
        android:textStyle="bold"
        android:background="@color/colorPrimary" />

    <TextView
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:text="3"
        android:textColor="#fff"
        android:textSize="15pt"
        android:textAlignment="center"
        android:textStyle="bold"
        android:background="#8c0520" />
</LinearLayout>
```



Hình 2. 20 Minh họa LinearLayout

2.6.3. Relative Layout

RelativeLayout là một loại Layout mà trong đó vị trí của mỗi View con sẽ được xác định so với view khác hoặc so với thành phần cha của chúng thông qua ID.

Sẽ có 4 tính cơ bản để sử dụng

- Layout_below: Nằm bên dưới.
- Layout_above: Nằm bên trên
- Layout_toRightOf: Nằm bên phải.
- Layout_toLeftOf: nằm bên trái.



Hình 2. 21 Minh họa RelativeLayout

```

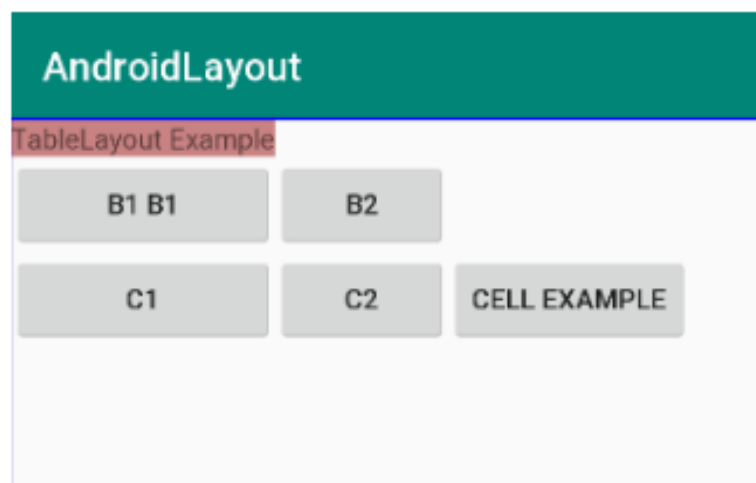
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/parent"
    tools:context=".MainActivity">
    <TextView
        android:background="@color/colorAccent"
        android:textAlignment="center"
        android:textSize="50sp"
        android:text="1"
        android:id="@+id/tview1"
        android:layout_width="100dp"
        android:layout_height="100dp"
        tools:ignore="NotSibling" />
    <TextView
        android:layout_below="@+id/tview1"
        android:background="@color/colorPrimary"
        android:textAlignment="center"
        android:textSize="50sp"
        android:text="2"
        android:id="@+id/tview2"
        android:layout_width="100dp"
        android:layout_height="100dp"
        tools:ignore="NotSibling" />
    <TextView
        android:layout_above=""
        android:layout_toRightOf="@+id/tview1"
        android:background="@android:color/holo_red_dark"
        android:textAlignment="center"
        android:textSize="50sp"
        android:text="3"
        android:id="@+id/tview3"
        android:layout_width="100dp"
        android:layout_height="100dp"
        tools:ignore="NotSibling" />
</RelativeLayout>

```


2.6.4. *TableLayout*

TableLayout sẽ sắp xếp các View con bên trong thành dạng bảng. Mỗi hàng là một đối tượng View *TableRow*. Bên trong *TableRow* chứa các View con, mỗi View con này nằm ở một ô.

```
<TableLayout android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <TableRow>
        <TextView
            android:text="TableLayout Example"
            android:background="#c98282"
            android:gravity="center"/>
    </TableRow>
    <TableRow>
        <Button android:text="B1 B1" />
        <Button android:text="B2"/>
    </TableRow>
    <TableRow>
        <Button android:text="C1" />
        <Button android:text="C2" />
        <Button android:text="Cell example" />
    </TableRow>
</TableLayout>
```



Hình 2. 22 Minh họa *TableLayout*

2.6.5. GridLayout

GridLayout là một dạng lưới và ta có thể chia các cột và dòng cho các lưới đó, các view sẽ được đặt vào các ô trong các lưới này.

```
<GridLayout
    xmlns:android1="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="2"
    android:rowCount="2">
    <TextView
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:text="1"
        android:textColor="#fff"
        android:textSize="15pt"
        android:textAlignment="center"
        android:background="@color/colorAccent" />
    <TextView
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:text="2"
        android:textColor="#fff"
        android:textSize="15pt"
        android:textAlignment="center"
        android:background="@color/colorPrimary" />
    <TextView
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:text="3"
        android:textColor="#fff"
        android:textSize="15pt"
        android:textAlignment="center"
        android:background="#8c0520" />
    <TextView
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:background="#efcd21"
        android:text="4"
        android:textAlignment="center"
        android:textColor="#fff"
        android:textSize="15pt" />
</GridLayout>
```



Hình 2. 23 Minh họa GridLayout

GridLayout được quy định bằng hai thuộc tính

`android:columnCount="2"` // số hàng

`android:rowCount="2"` // số cột

CHƯƠNG 3. GIỚI THIỆU VỀ SAMSUNG KNOX

3.1. Tổng quan về Samsung Knox

Samsung Knox là một giải pháp bảo mật di động hàng đầu cung cấp một môi trường an toàn cho dữ liệu và ứng dụng của công ty cho tất cả các thiết bị Galaxy. Nó bảo vệ doanh nghiệp và quyền riêng tư cá nhân của bạn khỏi một thiết bị mà không cần bảo vệ CNTT của bên thứ ba.

Vào ngày 5 tháng 3 năm 2018, Samsung đã công bố các thiết bị chạy Knox 3.0 trở lên tích hợp hoàn hảo với các tính năng tương tự của Android Enterprise.

Samsung Knox cung cấp danh sách các tính năng bảo mật cho cả phần cứng và phần mềm. Cho phép các nội dung của cá nhân và doanh nghiệp được bảo mật an toàn trên thiết bị có Knox.

Samsung Knox còn cung cấp các SDK giúp các nhà phát triển tiếp cận dễ dàng hơn.

Knox là nền tảng bảo mật cao cấp của Samsung và là giải pháp an ninh giúp bạn tự do làm việc và vui chơi mọi lúc, mọi nơi. Samsung Knox bao gồm nền tảng được tích hợp ngay trong thiết bị Samsung và đi kèm một bộ giải pháp tận dụng nền tảng này.

3.2. Knox SDK

Thông qua SDK Knox, bạn có thể quản lý một bộ tính năng toàn diện trên thiết bị di động Samsung Android. Bạn có thể kiểm soát tài khoản, ứng dụng, kết nối, tính năng tùy chỉnh, cài đặt thiết bị, bảo mật, cài đặt VPN, v.v. Thông qua hàng trăm phương thức API, SDK cung cấp bộ công cụ mạnh mẽ cho các nhà phát triển để tạo ra các giải pháp hấp dẫn và khác biệt. SDK đặc biệt phù hợp với các doanh nghiệp

muốn kiểm soát hoàn toàn thiết lập thiết bị của họ và khả năng đáp ứng các yêu cầu khắt khe nhất của công ty và ngành về bảo mật.

Knox SDK mở rộng các chức năng tiêu chuẩn của SDK Android, nhằm cung cấp quyền truy cập chi tiết vào các tính năng của thiết bị, tùy chọn bảo mật, cài đặt, ...

Ví dụ: Một đoạn sử dụng API ngăn quyền mở máy ảnh.

```
EnterpriseDeviceManager edm = EnterpriseDeviceManager.getInstance(context);
RestrictionPolicy restrictionPolicy = edm.getRestrictionPolicy();

try {
    boolean result = restrictionPolicy.setCameraState(false);

    if (true == result) {
        // Camera is disabled and cannot be enabled by user.
    }
} catch (SecurityException e) {    Log.w(TAG, "SecurityException: " + e); }
```

CHƯƠNG 4. MÔ TẢ VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG

4.1. Mô tả và yêu cầu hệ thống

4.1.1. Mô tả hệ thống

Tích hợp hai ứng dụng bảo vệ điện thoại và phát video giới thiệu, quảng cáo sản phẩm, nhằm tiết kiệm chi phí và giảm thiểu cài nhiều ứng dụng trên máy.

Tiện lợi cho người dùng.

Khi điện thoại bị rút khỏi cáp kết nối, một hệ thống chống trộm được kích hoạt.

Khi hệ thống được kích hoạt, thiết bị đồ chuông báo động và màn hình và các phím mềm bị khóa, hệ thống bảo vệ sẽ ngăn chặn việc người dùng buộc dừng, xóa ứng dụng hay tắt hoặc khởi động lại máy.

Khi hệ thống được kích hoạt vì báo động sai hoặc trục trặc, có thể xử lý bằng cách gắn thẻ NFC. Trong trường hợp thiết bị không hỗ trợ mô-đun NFC, cách mới sẽ được áp dụng để hoàn thành chế độ trộm.

Hệ thống cần phải đảm bảo được chức năng và giao diện thân thiện với người sử dụng, Xây dựng ứng dụng UI dựa trên One UI của samsung.

Hiệu năng của app chạy đảm bảo không ảnh hưởng đến hệ thống.

Video phát khi điện thoại ở chế độ rảnh cần đồng bộ với nhau ở mọi thời điểm.

4.1.2. Yêu cầu hệ thống

Hệ thống cần phải đáp ứng được tính bảo vệ các mẫu điện thoại của Samsung. Việc phát video và dừng video khi của hàng đóng cửa.

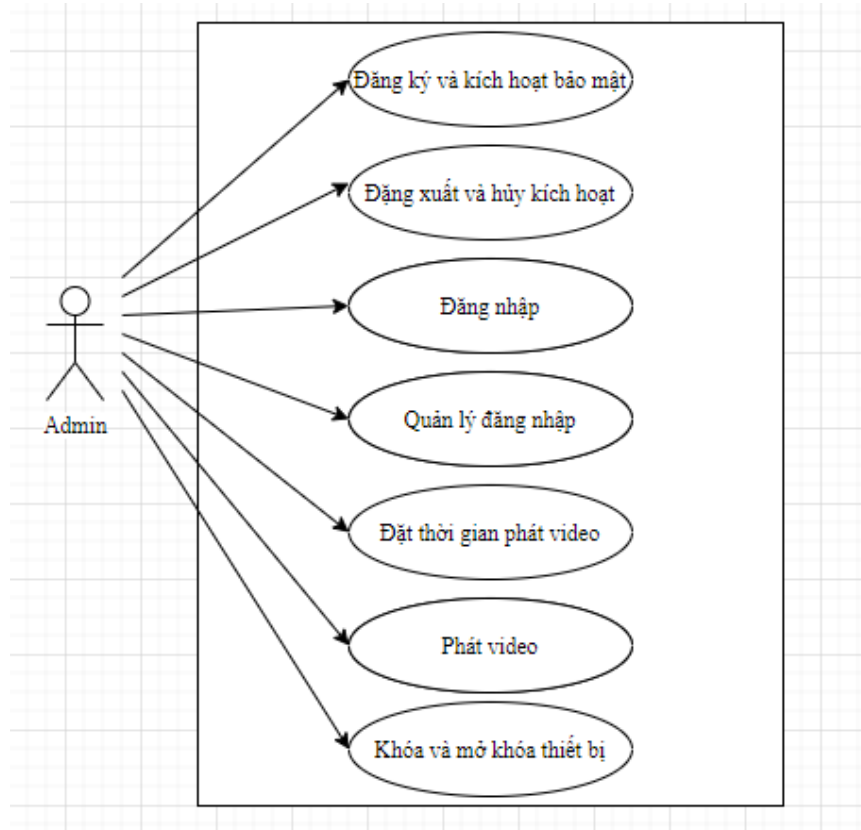
Các chức năng của ứng dụng:

- Đăng nhập
- Đăng ký
- Đăng xuất

- Khóa và cảnh báo khi thiết bị bị đánh cắp
- Phát video dưới nền khi điện thoại ở chế độ rảnh
- Quản lý đăng nhập
- Đặt thời gian phát video

4.2. Phân tích thiết kế hệ thống

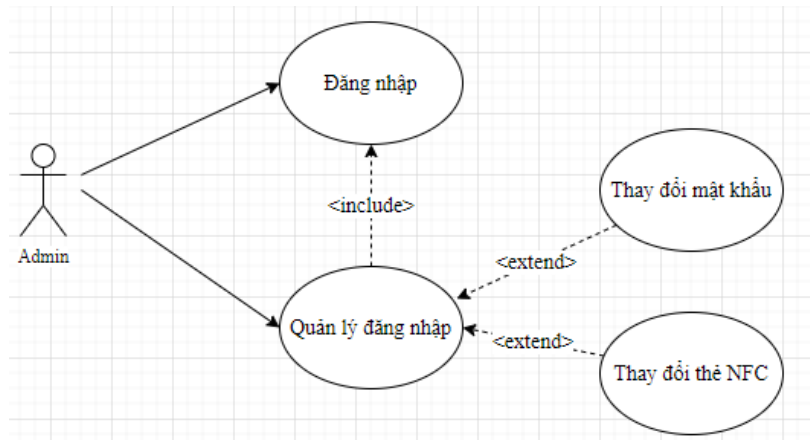
4.2.1. Biểu đồ use case tổng quát



Hình 4. 1 Biểu đồ use case tổng quát

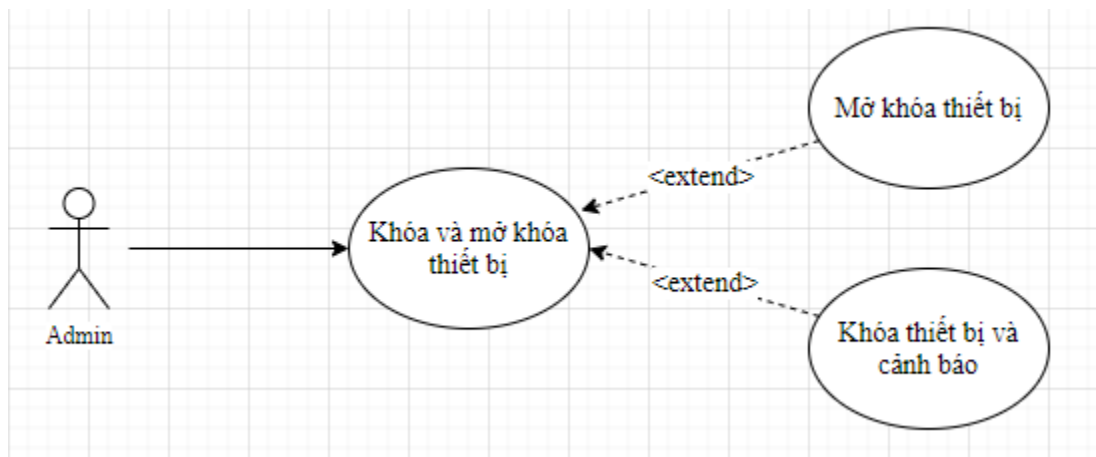
4.2.2. Phân rã use case

- Phân rã chức năng “Quản lý đăng nhập”



Hình 4. 2 Biểu đồ phân rã chức năng đăng nhập

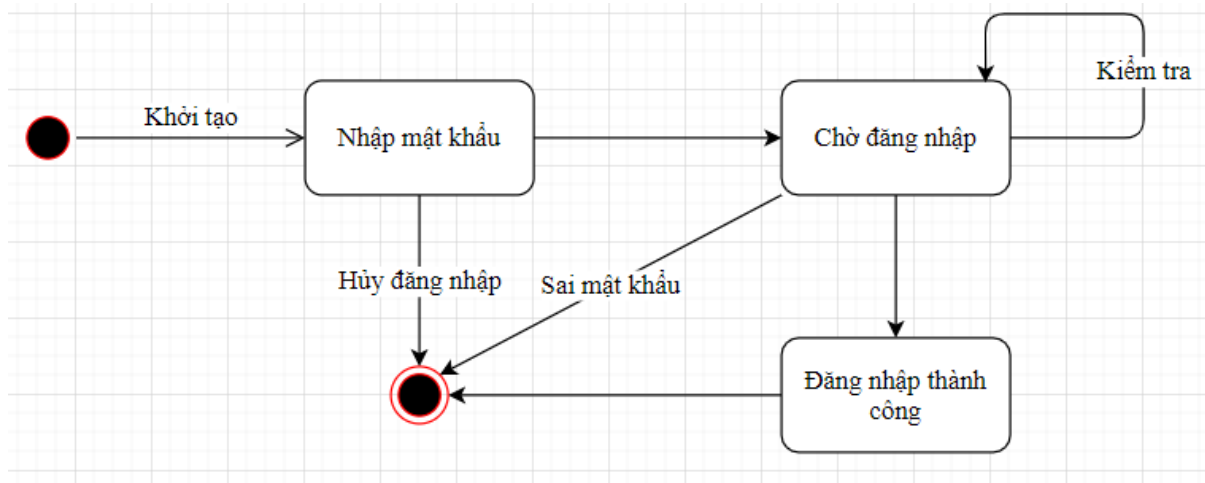
- Phân rã chức năng “Khóa và mở khóa thiết bị”



Hình 4. 3 Biểu đồ phân rã chức năng khóa và mở khóa thiết bị

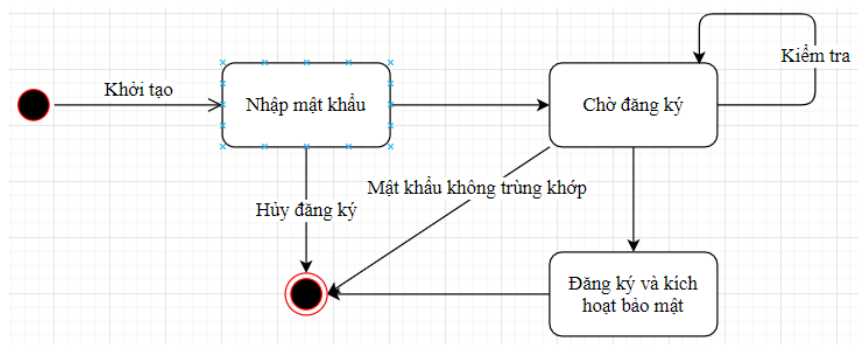
4.2.3. Biểu đồ trạng thái

- Biểu đồ trạng thái “Đăng nhập”



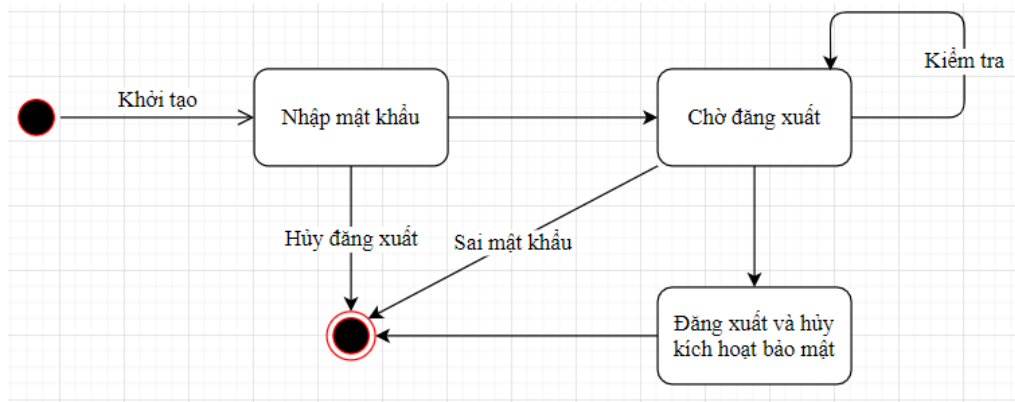
Hình 4. 4 Biểu đồ trạng thái “Đăng nhập”

- Biểu đồ trạng thái “Đăng ký và kích hoạt bảo mật”



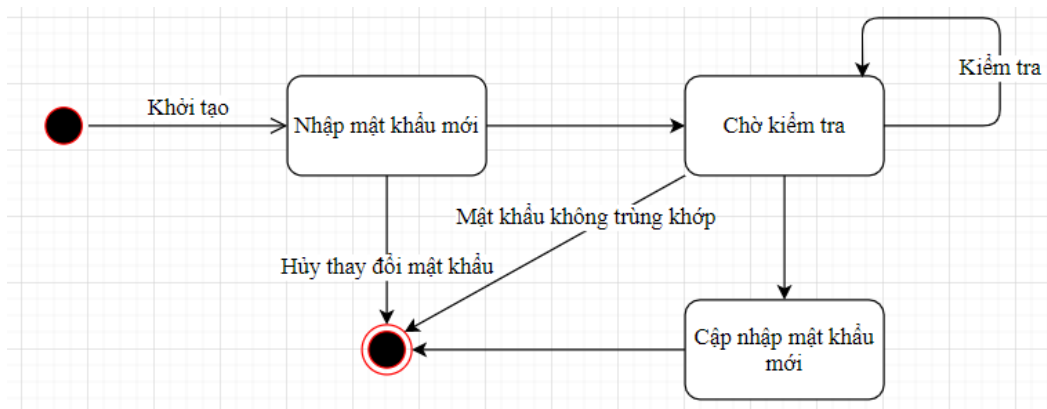
Hình 4. 5 Biểu đồ trạng thái “Đăng ký và kích hoạt bảo mật”

- Biểu đồ trạng thái “Đăng xuất và hủy kích hoạt”



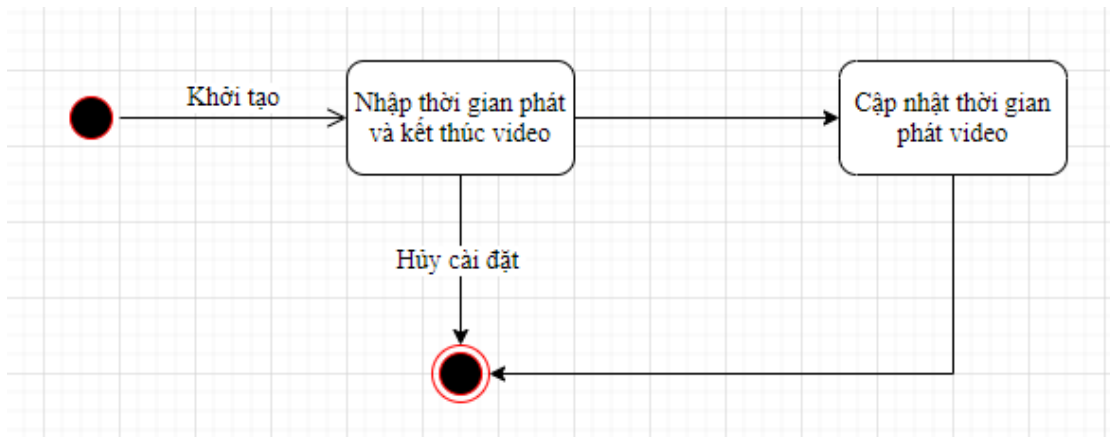
Hình 4. 6 Biểu đồ trạng thái “Đăng xuất và hủy kích hoạt”

- Biểu đồ trạng thái “Thay đổi mật khẩu”



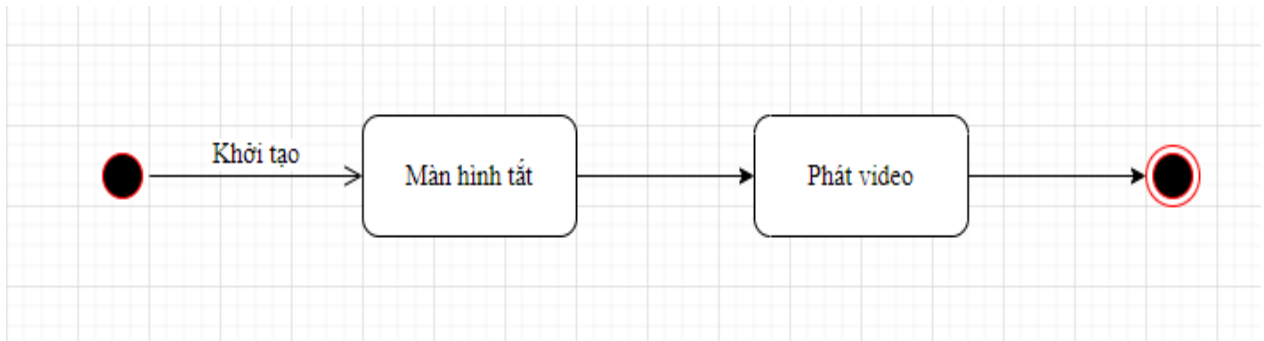
Hình 4. 7 Biểu đồ trạng thái “Thay đổi mật khẩu”

- Biểu đồ trạng thái “Đặt thời gian phát video”



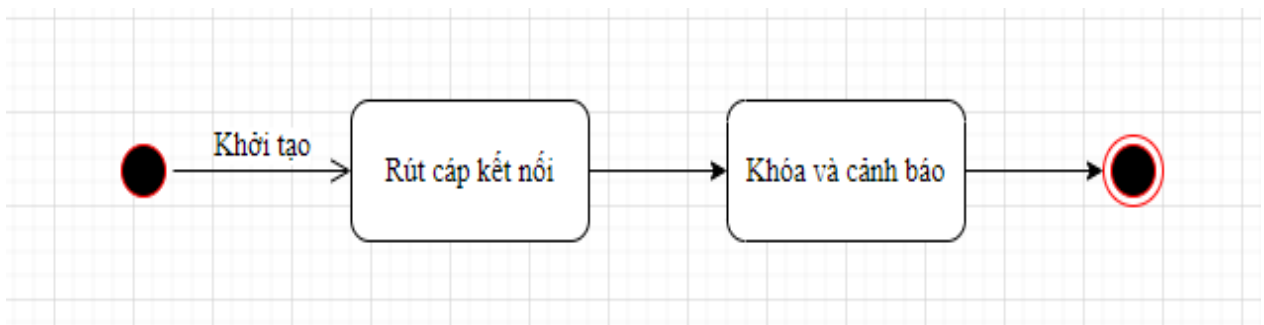
Hình 4. 8 Biểu đồ trạng thái “Đặt thời gian phát video”

- Biểu đồ trạng thái “Phát video”



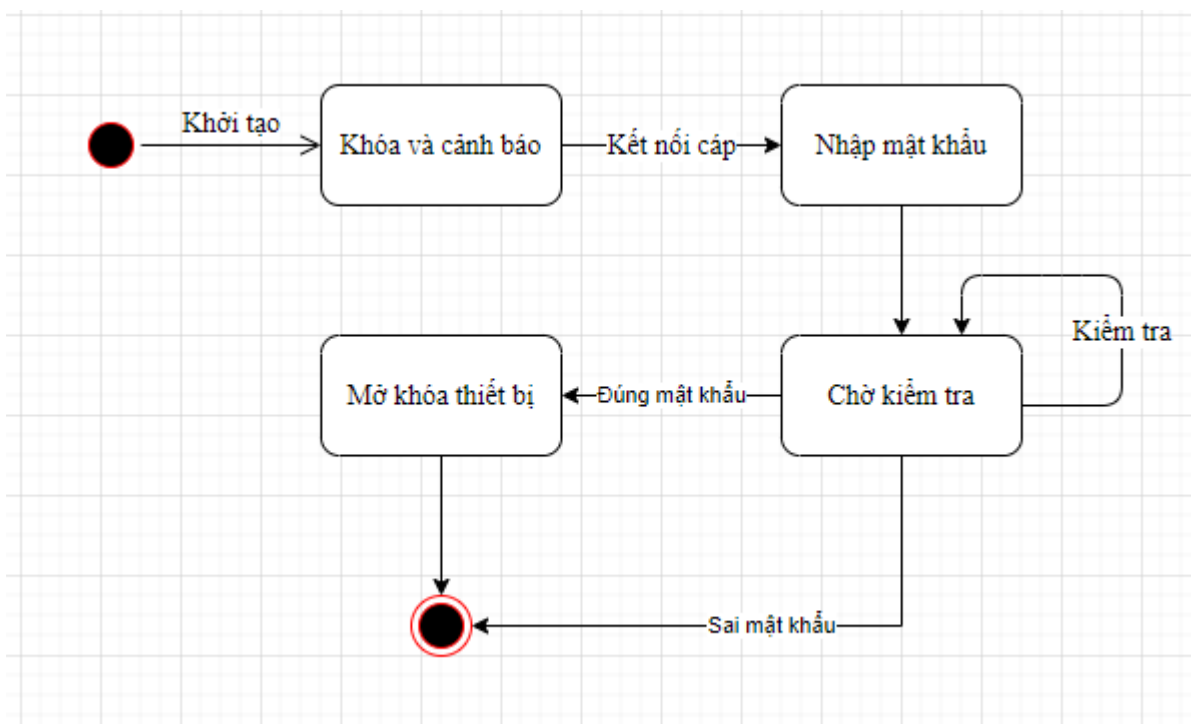
Hình 4. 9 Biểu đồ trạng thái “Phát video”

- Biểu đồ trạng thái “Khóa và cảnh báo”



Hình 4. 10 Biểu đồ trạng thái “Khóa và cảnh báo”

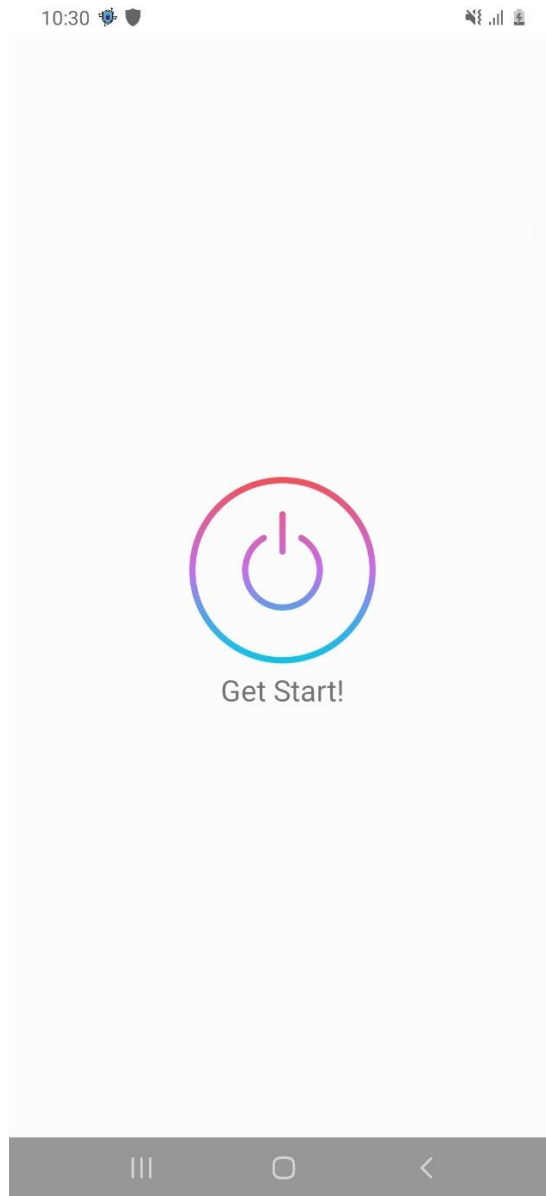
- Biểu đồ trạng thái “Mở khóa thiết bị”



Hình 4. 11 Biểu đồ “Mở khóa thiết bị”

CHƯƠNG 5. XÂY DỰNG ỨNG DỤNG

5.1. Giao diện khởi chạy và xin quyền Admin



Hình 5. 1 Giao diện khởi chạy ứng dụng

Giao diện khởi chạy là giao diện bắt đầu được chạy ở lần đầu tiên. Trên giao diện có một nút bấm. Khi Admin ấn vào, hệ thống tiếp nhận và bắt đầu khởi chạy hệ thống. Nếu hệ thống chưa được cấp quyền. Màn hình thông báo xin cấp quyền ứng dụng hiển thị lên.

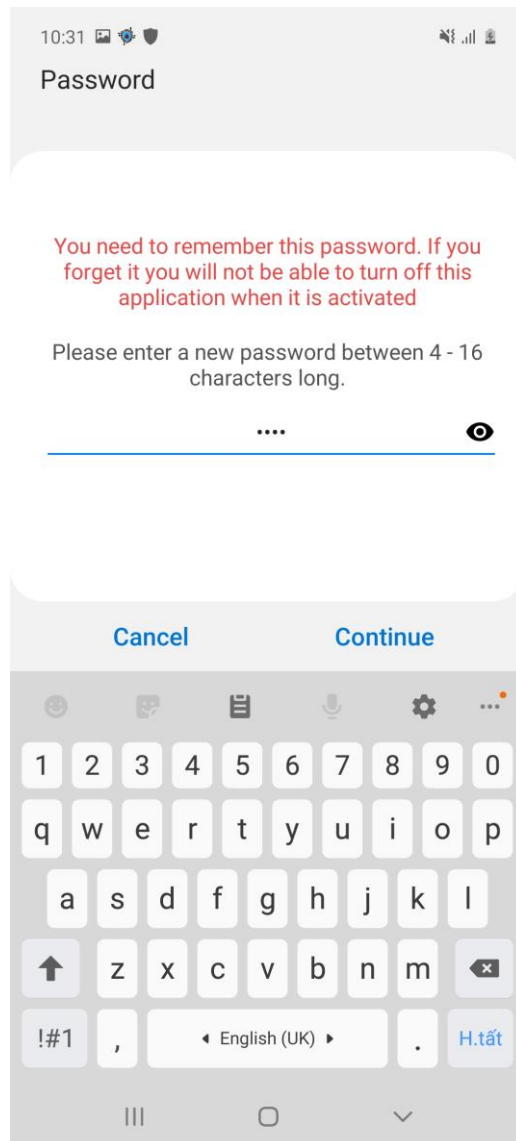


Hình 5. 2 Giao diện xin cấp quyền Admin

Khi ứng dụng chưa được cấp quyền Admin. Giao diện cấp quyền hiện được hiện lên có ghi danh sách các sẽ được cấp nếu người sử dụng dùng ấn vào “Bật”.

Nếu người sử dụng ấn vào “Thoát” thì ứng dụng sẽ kết thúc và sẽ bị buộc dừng. Khi người sử dụng ấn vào gỡ cài đặt. Việc gỡ ứng dụng sẽ được thực hiện.

5.2. Giao diện đăng ký

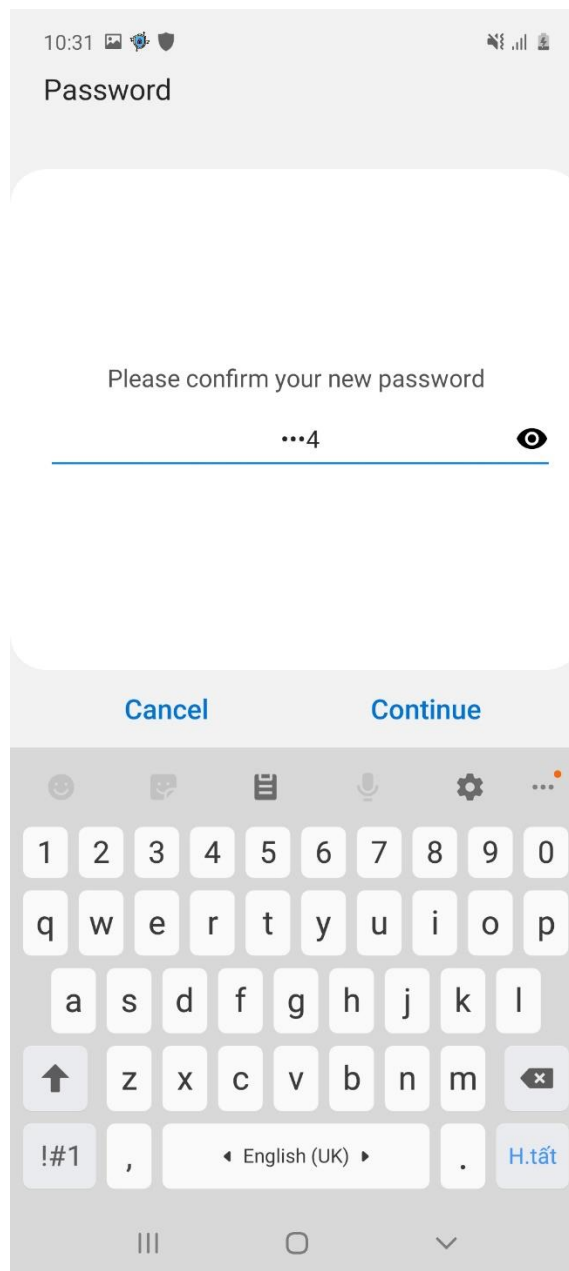


Hình 5. 3 Giao diện đăng ký

Xin các quyền cần thiết xong. Màn hình đăng ký sẽ được hiển thị nếu người sử dụng chưa kích hoạt ứng dụng.

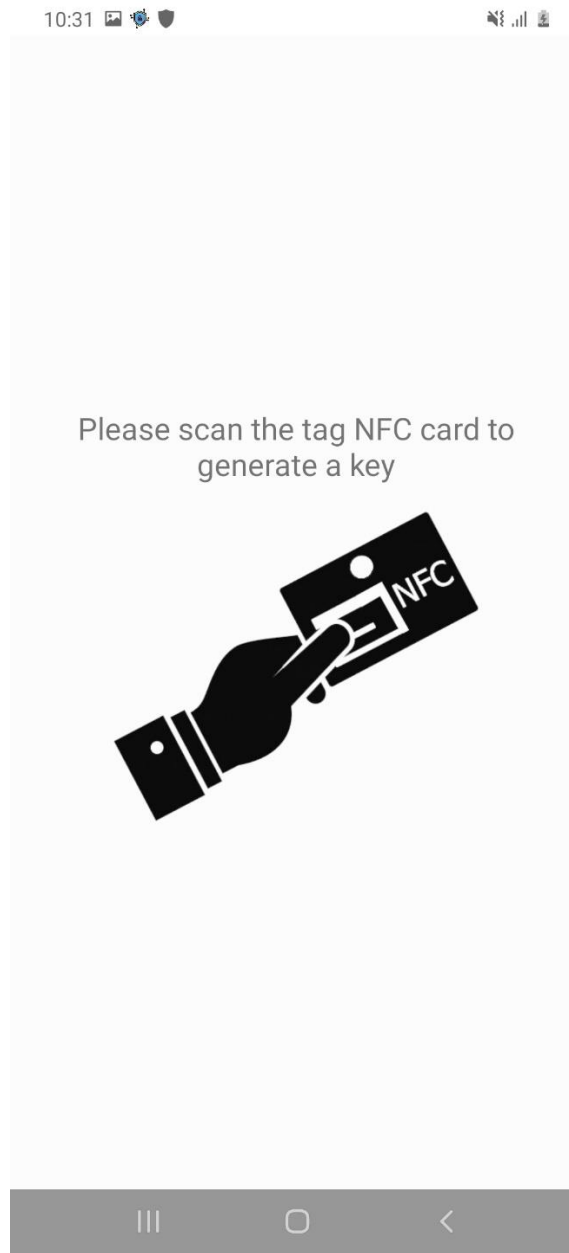
Khi người sử dụng nhập đủ độ dài của mật khẩu, Thì nút “Continue” mới được hiển thị lên, cho phép người sử dụng đến bước tiếp theo.

Ngoài ra còn nút hiển thị mật khẩu và ẩn mật khẩu trong trường hợp khi người dùng muốn xem mật khẩu mình nhập đã chính xác.



Hình 5. 4 Giao diện xác nhận mật khẩu

Khi người sử dụng nhập mật khẩu xong. Giao diện nhập lại mật khẩu được hiển thị nhằm xác nhận lại mật khẩu người sử dụng vừa nhập, giúp người sử dụng không nhập sai mật khẩu nhằm tránh quên mật khẩu và nhập mật khẩu không chính xác.

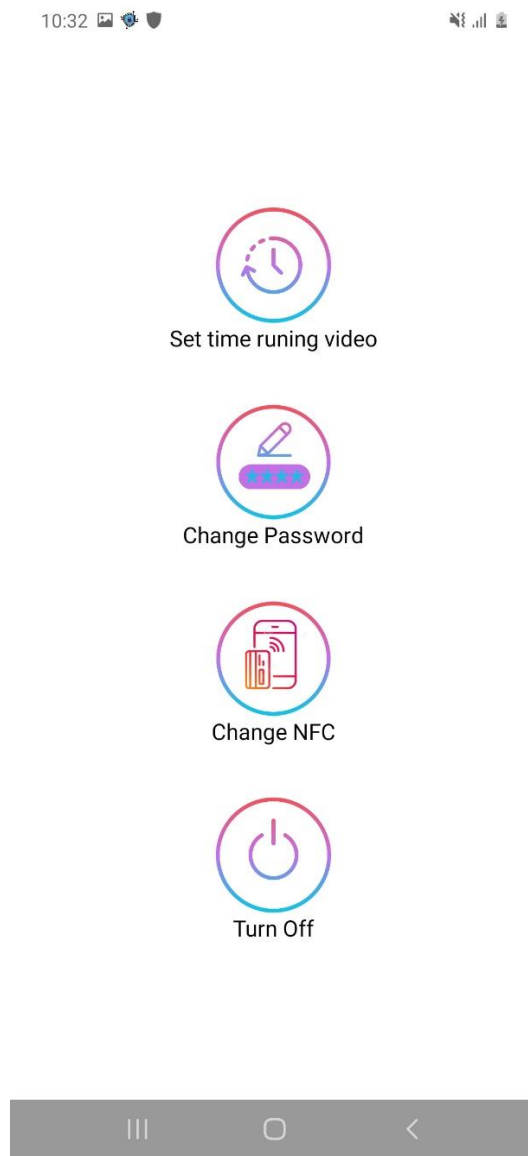


Hình 5. 5 Giao diện đăng ký thẻ NFC

Giao diện đăng ký thẻ NFC cho phép người sử dụng tạo thêm một lớp khóa bảo mật bằng thẻ NFC. Khi người dùng quét thẻ NFC hệ thống sẽ lưu lại.

Nếu người dùng không đăng ký thẻ hoặc thoát ứng dụng. Hệ thống sẽ không lưu lại mật khẩu.

5.3. Giao diện trang chủ



Hình 5. 6 Giao diện trang chủ

Khi việc đăng ký hoặc đăng nhập được hoàn tất. Giao diện trang chủ được hiện thị, ứng dụng sẽ kích hoạt tính năng bảo vệ.

Trên giao diện trang chủ có các nút ấn tương ứng với các chức năng của ứng dụng gồm: Cài đặt thời gian phát video, thay đổi mật khẩu, thay đổi thẻ NFC và hủy kích hoạt ứng dụng.

5.4. Giao diện thay đổi mật khẩu và thẻ NFC

10:32

Password

You need to remember this password. If you forget it you will not be able to turn off this application when it is activated

Password length is from 4 to 16 characters

Old password

New password

Confirm Password

Cancel Done

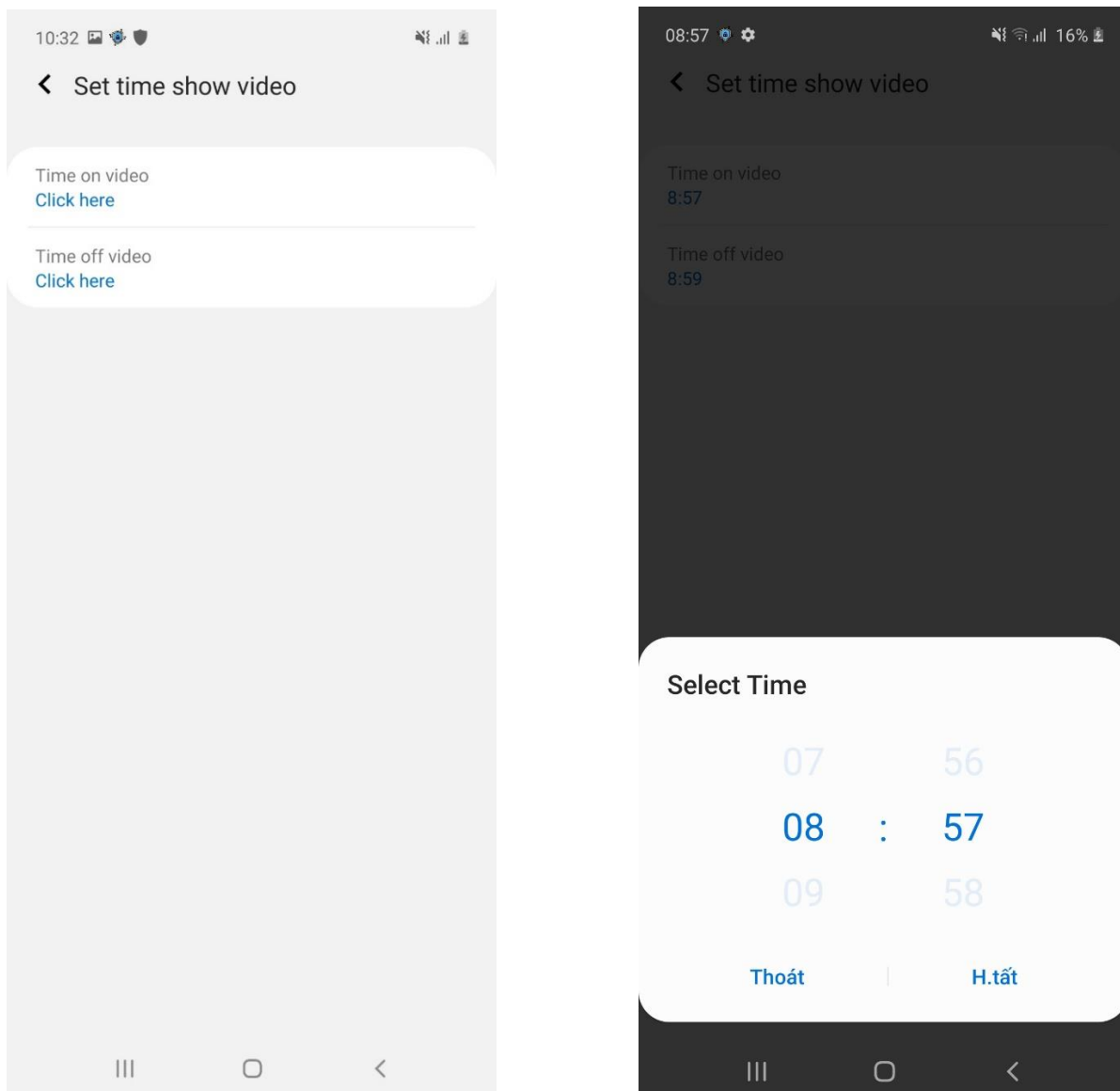
Hình 5. 7 Giao diện thay đổi mật khẩu

Khi độ dài mật khẩu đáp ứng đúng yêu cầu, thì nút “Done” sẽ hiển thị.

Giao diện thay đổi gồm có ba trường nhập dữ liệu. Với trường đầu tiên là mật khẩu cũ. Trường thứ hai là mật khẩu mới và trường thứ ba là nhập lại mật khẩu mới. Nếu các trường nhập mật khẩu hợp lệ thì cập nhập mật khẩu thành công.

Nếu người dùng không muốn thay đổi mật khẩu thì có thể ấn “Cancel” để hủy.

5.5. Giao diện đặt thời gian phát video



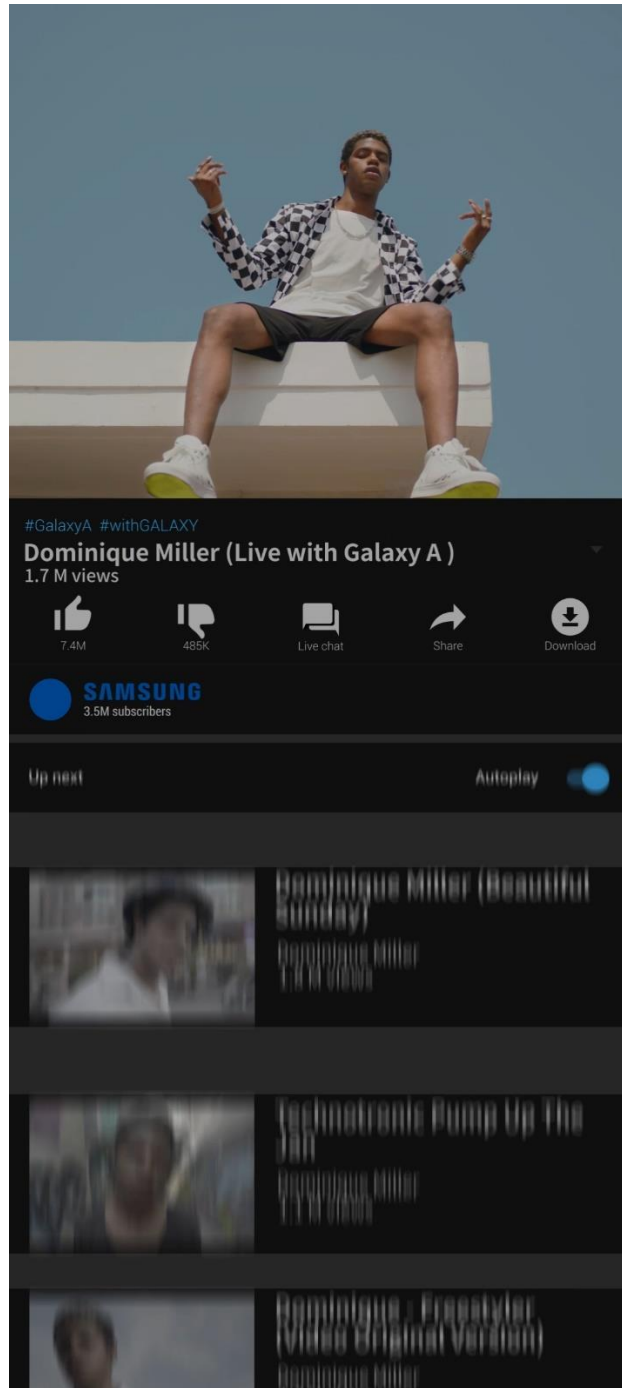
Hình 5. 8 Giao diện đặt thời gian phát video

Người sử dụng muốn đặt thời gian bắt đầu phát video và dừng phát video quảng cáo.

Với giao diện trên, dòng đầu tiên là thời gian bắt đầu phát video. Được tính theo giờ và phút.

Dòng thứ hai là thời gian dừng phát video được tính theo giờ và phút.

5.6. Giao diện phát video



Hình 5. 9 Giao diện phát video

Khi điện thoại tắt màn hình hoặc không sử dụng, một video được chạy quảng cáo sẽ được phát.

Phát video được đồng bộ với các thiết bị cùng loại vào cùng thời điểm.

KẾT LUẬN

Hệ thống an ninh được cài đặt để ngăn chặn trường hợp trộm cắp và sặc ỏn định cho các thiết bị được trưng bày trong cửa hàng, chạy các video quảng cáo với chi phí rẻ, hệ thống bớt công kênh.

Nên việc tạo ra một ứng dụng nhằm hỗ trợ việc bảo vệ, quảng cáo sản phẩm là việc hết sức cần thiết. Việc cho ra đời một ứng dụng nhằm hỗ trợ, bảo vệ và quảng cáo điện thoại đã mang lại những lợi ích lớn cho việc quản lý vào bảo vệ.

Những công việc đã làm được:

- Tìm hiểu các kiến thức về Android, nắm được kiến thức cơ bản về các phiên bản Android.
- Tìm hiểu và sử dụng được Samsung Knox.
- Tìm hiểu và sử dụng được Android Security.
- Hoàn thành tốt app với những chức năng đã đặt ra.

Hướng phát triển:

- Triển khai Server: hướng đến triển khai và lưu trữ dữ liệu trên Server, sử dụng Server để điều khiển và đồng bộ giữa các điện thoại. Thực hiện khóa, mở khóa và thay đổi video hoặc đặt thời gian phát video bằng trên Server.
- Thêm các tính năng mới như thông báo tình trạng pin, cảnh báo khi hết pin.
- Thống kê được phiên bản phần mềm của điện thoại ở các cửa hàng.
- Người dùng có thể phản hồi các ý kiến trải nghiệm điện thoại về cho server, để thống kê và cải thiện khi cần thiết.

NHẬN XÉT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, Ngày Tháng Năm 2020

Giáo viên hướng dẫn

TÀI LIỆU THAM KHẢO

Website

- [1] [https://vi.wikipedia.org/wiki/Android_\(hệ_điều_hành\)](https://vi.wikipedia.org/wiki/Android_(hệ_điều_hành))
- [2] <https://docs.samsungknox.com/dev/knox-sdk/index.htm>
- [3] <https://docs.samsungknox.com/devref/knox-sdk/reference/packages.html>
- [4] <https://developer.android.com/reference>
- [5] <https://stackoverflow.com/>
- [6] <https://www.tutorialspoint.com/index.htm>