

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

---



ISO 9001:2015

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Bùi Hoàng Anh**

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

**HẢI PHÒNG - 2018**

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

---

**XÂY DỰNG ỨNG DỤNG ANDROID**  
**ĐỌC BÁO MẠNG QUA DỊCH VỤ RSS**

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

**Sinh viên : Bùi Hoàng Anh**

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

**HẢI PHÒNG - 2018**

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

---

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

Sinh viên: Bùi Hoàng Anh

Mã SV: 1412101114

Lớp: CT1801

Ngành: Công nghệ thông tin

Tên đề tài: Xây dựng ứng dụng Android đọc báo mạng qua dịch vụ RSS

# MỤC LỤC

LỜI CẢM ƠN.....	6
CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID .....	7
1.1. Giới thiệu hệ điều hành Android .....	7
1.2. Lịch sử phát triển.....	8
1.3. Giao diện Android.....	9
1.4. Ứng dụng Android .....	11
1.5. Quản lý bộ nhớ Android .....	12
1.6. Nhân Linux.....	12
1.7. Lịch nâng cấp .....	13
1.8. Cộng đồng mã nguồn mở.....	14
1.9. Bảo mật và tính riêng tư của Android .....	15
CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO .....	17
2.1. Sơ lược về Android Studio .....	17
2.2. Cài đặt Android Studio .....	17
2.2.1. Yêu cầu phần cứng máy tính.....	17
2.2.2. Phần mềm Android Studio .....	19
2.2.3. Thiết bị ảo trong Android Studio .....	20
2.3. Cấu trúc dự án android studio .....	23
2.3.1. Tạo mới một Project .....	23
2.3.2. Cài đặt một Project .....	24
2.3.3. Màn hình làm việc của dự án Android Studio .....	26
2.4. Các thành phần trong một ứng dụng Android.....	29
2.5. Tạo giao diện chương trình trong Android Studio .....	32
2.5.1. Giới thiệu Android Layout .....	32
2.5.2. Giới thiệu một số Android View cơ bản.....	34
2.5.3. Bắt và xử lý sự kiện trên giao diện.....	35
CHƯƠNG 3: KỸ THUẬT ĐỌC LẤY NỘI DUNG CỦA TRANG WEB VỀ ĐIỆN THOẠI.....	37
3.1. Tìm hiểu về dịch vụ RSS .....	37
3.1.1. Khái niệm.....	37
3.1.2. Ví dụ.....	37
3.2. Kỹ thuật đọc lấy nội dung của trang web về điện thoại.....	38

3.2.1. Quy trình thực hiện.....	38
CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM.....	50
4.1. Phát biểu bài toán.....	50
4.2. Mô hình chức năng.....	50
4.3. Mô hình hoạt động.....	51
4.4. Giao diện ứng dụng.....	51
KẾT LUẬN.....	60
TÀI LIỆU THAM KHẢO.....	61

## LỜI CẢM ƠN

Để đề án này đạt kết quả tốt đẹp, em đã nhận được sự hỗ trợ, giúp đỡ của nhiều cơ quan, tổ chức, cá nhân. Với tình cảm sâu sắc, chân thành, cho phép em được bày tỏ lòng biết ơn sâu sắc đến tất cả các cá nhân và cơ quan đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu làm đề án. Trước hết em xin gửi tới các thầy cô khoa Công nghệ - Thông tin trường Đại học Dân Lập Hải Phòng lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay em đã có thể hoàn thành đề án: "Xây dựng ứng dụng Android đọc báo mạng qua dịch vụ RSS". Đặc biệt em xin gửi lời cảm ơn chân thành nhất tới thầy giáo - Ths. Phùng Anh Tuấn đã quan tâm giúp đỡ, hướng dẫn em hoàn thành tốt đề án này trong thời gian qua. Em xin bày tỏ lòng biết ơn đến lãnh đạo Trường Đại học Dân Lập Hải Phòng, Phòng Đào Tạo, các Khoa Phòng ban chức năng đã trực tiếp và gián tiếp giúp đỡ em trong suốt quá trình học tập tại trường. Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đề án này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công việc thực tế sau này.

Em xin chân thành cảm ơn!

Sinh viên

Bùi Hoàng Anh

# CHƯƠNG 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID

## 1.1. Giới thiệu hệ điều hành Android

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cảm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào tháng 10 năm 2008.

Android có mã nguồn mở và Google phát hành mã nguồn theo giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi. Vào tháng 10 năm 2012, có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play, cửa hàng ứng dụng chính của Android, ước tính khoảng 25 tỷ lượt.

Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới, vượt qua Symbian vào quý 4 năm 2010, và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu. Kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác. Bản chất mở của Android cũng khích lệ một đội ngũ đông đảo lập trình viên và những người đam mê sử dụng mã nguồn mở để tạo ra những dự án do cộng đồng quản lý. Những dự án này bổ sung các tính năng cao cấp cho những người dùng thích tìm tòi hoặc đưa Android vào các thiết bị ban đầu chạy hệ điều hành khác.

Android chiếm 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Sự thành công của hệ điều hành cũng khiến nó trở thành mục tiêu trong các vụ kiện liên quan đến bằng phát minh, góp mặt trong cái gọi là "cuộc chiến điện thoại thông minh" giữa các công ty công nghệ.<sup>[1]</sup>

## 1.2. Lịch sử phát triển

Tổng công ty Android (Android, Inc.) được thành lập tại Palo Alto, California vào tháng 10 năm 2003 bởi Andy Rubin (đồng sáng lập công ty Danger), Rich Miner (đồng sáng lập Tổng công ty Viễn thông Wildfire), Nick Sears (từng là Phó giám đốc T-Mobile), và Chris White (trưởng thiết kế và giao diện tại WebTV) để phát triển, theo lời của Rubin, "các thiết bị di động thông minh hơn có thể biết được vị trí và sở thích của người dùng". Dù những người thành lập và nhân viên đều là những người có tiếng tăm, Tổng công ty Android hoạt động một cách âm thầm, chỉ tiết lộ rằng họ đang làm phần mềm dành cho điện thoại di động. Trong năm đó, Rubin hết kinh phí. Steve Perlman, một người bạn thân của Rubin, mang cho ông 10.000 USD tiền mặt nhưng từ chối tham gia vào công ty.

Google mua lại Tổng công ty Android vào ngày 17 tháng 8 năm 2005, biến nó thành một bộ phận trực thuộc Google. Những nhân viên của chủ chốt của Tổng công ty Android, gồm Rubin, Miner và White, vẫn tiếp tục ở lại công ty làm việc sau thương vụ này. Vào thời điểm đó không có nhiều thông tin về Tổng công ty, nhưng nhiều người đồn đoán rằng Google dự tính tham gia thị trường điện thoại di động sau bước đi này. Tại Google, nhóm do Rubin đứng đầu đã phát triển một nền tảng thiết bị di động phát triển trên nền nhân Linux. Google quảng bá nền tảng này cho các nhà sản xuất điện thoại và các nhà mạng với lời hứa sẽ cung cấp một hệ thống uyển chuyển và có khả năng nâng cấp. Google đã liên hệ với hàng loạt hãng phần cứng cũng như đối tác phần mềm, bắt tin cho các nhà mạng rằng họ sẵn sàng hợp tác với các cấp độ khác nhau.

Ngày càng nhiều suy đoán rằng Google sẽ tham gia thị trường điện thoại di động xuất hiện trong tháng 12 năm 2006. Tin tức của BBC và Nhật báo phố Wall chú thích rằng Google muốn đưa công nghệ tìm kiếm và các ứng dụng của họ vào điện thoại di động và họ đang nỗ lực làm việc để thực hiện điều này. Các phương tiện truyền thông truyền thống lẫn online cũng viết về tin đồn rằng Google đang phát triển một thiết bị cầm tay mang thương hiệu Google. Một vài tờ báo còn nói rằng trong khi Google vẫn đang thực hiện những bản mô tả kỹ thuật chi tiết, họ đã trình diễn sản phẩm mẫu cho các nhà sản xuất điện thoại di động và nhà mạng. Tháng 9 năm 2007, InformationWeek đăng tải một nghiên cứu của Evalueserve cho biết Google đã nộp một số đơn xin cấp bằng sáng chế trong lĩnh vực điện thoại di động.

Ngày 5 tháng 11 năm 2007, Liên minh thiết bị cầm tay mở (Open Handset Alliance), một hiệp hội bao gồm nhiều công ty trong đó có Texas Instruments, Tập đoàn Broadcom, Google, HTC, Intel, LG, Tập đoàn Marvell Technology, Motorola, Nvidia,



Qualcomm, Samsung Electronics, Sprint Nextel và T-Mobile được thành lập với mục đích phát triển các tiêu chuẩn mở cho thiết bị di động. Cùng ngày, Android cũng được ra mắt với vai trò là sản phẩm đầu tiên của Liên minh, một nền tảng thiết bị di động được xây dựng trên nhân Linux phiên bản 2.6. Chiếc điện thoại chạy Android đầu tiên được bán ra là HTC Dream, phát hành ngày 22 tháng 10 năm 2008. Biểu trưng của hệ điều hành Android mới là một con robot màu xanh lá cây do hãng thiết kế Irina Blok tại California vẽ.

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

### **1.3. Giao diện Android**

Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo giãn và thu lại để xử lý các đối tượng trên màn hình. Sự phản ứng với tác động của người dùng diễn ra gần như ngay lập tức, nhằm tạo ra giao diện cảm ứng mượt mà, thường dùng tính năng rung của thiết bị để tạo phản hồi rung cho người dùng. Những thiết bị phần cứng bên trong như gia tốc kế, con quay hồi chuyển và cảm biến khoảng cách được một số ứng dụng sử dụng để phản hồi một số hành động khác của người dùng, ví dụ như điều chỉnh màn hình từ chế độ hiển thị dọc sang chế độ hiển thị ngang tùy theo vị trí của thiết bị, hoặc cho phép người dùng lái xe đua bằng xoay thiết bị, giống như đang điều khiển vô-lăng.

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop (bàn làm việc) trên máy tính để bàn. Màn hình chính Android thường gồm nhiều biểu tượng (icon) và tiện ích

(widget); biểu tượng ứng dụng sẽ mở ứng dụng tương ứng, còn tiện ích hiển thị những nội dung sống động, cập nhật tự động như dự báo thời tiết, hộp thư của người dùng, hoặc những mẩu tin thời sự ngay trên màn hình chính. Màn hình chính có thể gồm nhiều trang xem được bằng cách vuốt ra trước hoặc sau, mặc dù giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích. Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi "chủ đề" của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như Windows Phone chẳng hạn. Phần lớn những nhà sản xuất, và một số nhà mạng, thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh.

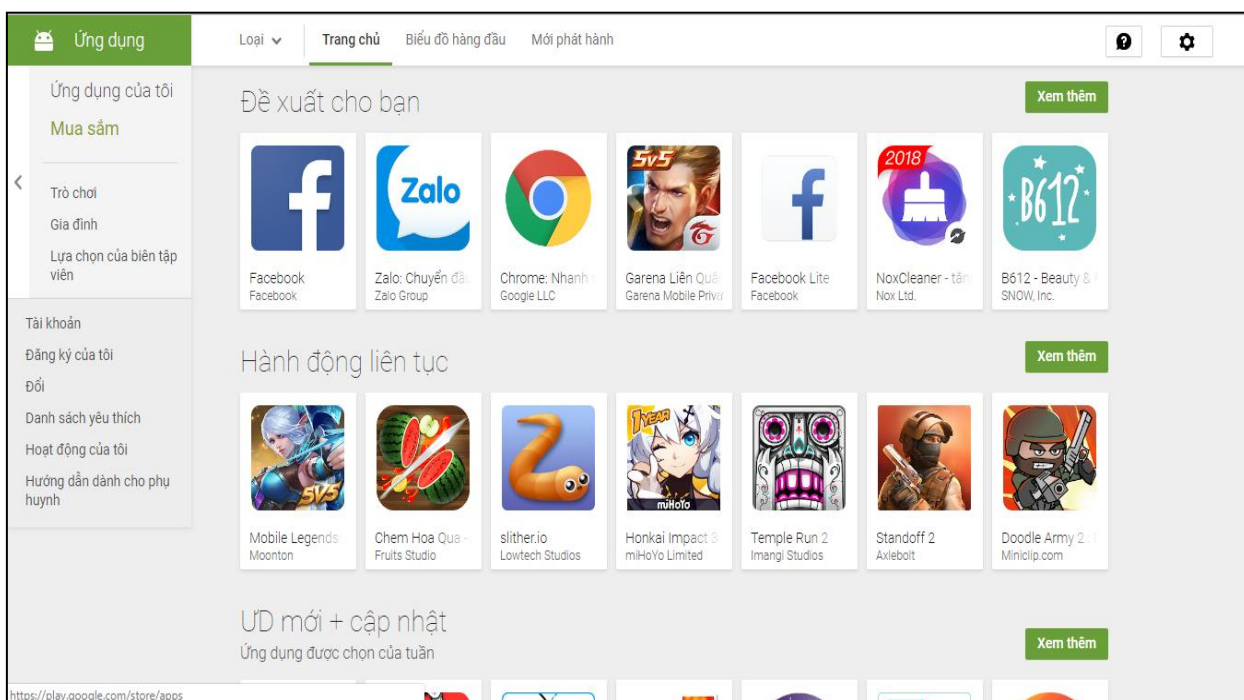


Hình 1.3.1: Giao diện màn hình điện thoại trên hệ điều hành Android 8.0

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

## 1.4. Ứng dụng Android

Android có lượng ứng dụng của bên thứ ba ngày càng nhiều, được chọn lọc và đặt trên một cửa hàng ứng dụng như Google Play hay Amazon Appstore để người dùng lấy về, hoặc bằng cách tải xuống rồi cài đặt tập tin APK từ trang web khác. Các ứng dụng trên Cửa hàng Play cho phép người dùng duyệt, tải về và cập nhật các ứng dụng do Google và các nhà phát triển thứ ba phát hành. Cửa hàng Play được cài đặt sẵn trên các thiết bị thỏa mãn điều kiện tương thích của Google. Ứng dụng sẽ tự động lọc ra một danh sách các ứng dụng tương thích với thiết bị của người dùng, và nhà phát triển có thể giới hạn ứng dụng của họ chỉ dành cho những nhà mạng cố định hoặc những quốc gia cố định vì lý do kinh doanh. Nếu người dùng mua một ứng dụng mà họ cảm thấy không thích, họ được hoàn trả tiền sau 15 phút kể từ lúc tải về, và một vài nhà mạng còn có khả năng mua giúp các ứng dụng trên Google Play, sau đó tính tiền vào trong hóa đơn sử dụng hàng tháng của người dùng. Đến năm 2017, có hơn 3,6 triệu ứng dụng dành cho Android, và số lượng ứng dụng tải về từ Cửa hàng Play ước tính đạt 64 tỷ lượt.



Hình 1.4.1: Giao diện trang chủ Google Play

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng bộ phát triển phần mềm Android (SDK). SDK bao gồm một bộ đầy đủ các công cụ dùng để phát triển, gồm có công cụ gỡ lỗi, thư viện phần mềm, bộ giả lập điện thoại dựa trên QEMU, tài liệu hướng dẫn, mã nguồn mẫu, và hướng dẫn từng bước. Môi trường phát triển tích hợp (IDE) được hỗ trợ chính thức là Eclipse sử dụng phần bổ sung Android Development Tools (ADT). Các công cụ phát triển khác cũng có sẵn, gồm có Bộ phát triển gốc dành

cho các ứng dụng hoặc phần mở rộng viết bằng C hoặc C++, Google App Inventor, một môi trường đồ họa cho những nhà lập trình mới bắt đầu, và nhiều nền tảng ứng dụng web di động đa nền tảng phong phú.

### **1.5. Quản lý bộ nhớ Android**

Vì các thiết bị Android được thiết kế để quản lý bộ nhớ (RAM) để giảm tối đa mức tiêu thụ điện năng, trái với hệ điều hành máy tính để bàn luôn cho rằng máy tính sẽ có nguồn điện không giới hạn. Khi một ứng dụng Android không còn được sử dụng, hệ thống sẽ tự động ngưng nó trong bộ nhớ-trong khi ứng dụng về mặt kỹ thuật vẫn "mở", những ứng dụng này sẽ không tiêu thụ bất cứ tài nguyên nào (như năng lượng pin hay năng lượng xử lý) và nằm đó cho đến khi nó được cần đến. Cách làm như vậy có lợi kép là vừa làm tăng khả năng phản hồi nói chung của thiết bị Android, vì ứng dụng không nhất phải đóng rồi mở lại từ đầu, vừa đảm bảo các ứng dụng nền không làm tiêu hao năng lượng một cách không cần thiết.

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

### **1.6. Nhân Linux**

Android có một hạt nhân dựa trên nhân Linux phiên bản 2.6, kể Android 4.0 Ice Cream Sandwich (bánh ngọt kẹp kem) trở về sau, là phiên bản 3.x, với middleware, thư viện và API viết bằng C, còn phần mềm ứng dụng chạy trên một nền tảng ứng dụng gồm các thư viện tương thích với Java dựa trên Apache Harmony. Android sử dụng máy ảo Dalvik với một trình biên dịch động để chạy 'mã dex' (Dalvik Executable) của Dalvik, thường được biên dịch sang Java bytecode. Nền tảng phần cứng chính của Android là kiến trúc ARM. Người ta cũng hỗ trợ x86 thông qua dự án Android x86, và Google TV cũng sử dụng một phiên bản x86 đặc biệt của Android.

Một số tính năng cũng được Google đóng góp ngược vào nhân Linux, đáng chú ý là tính năng quản lý nguồn điện có tên wakelock, nhưng bị những người lập trình chính cho nhân từ chối vì họ cảm thấy Google không có định sẽ tiếp tục bảo trì đoạn mã do họ viết.

Google thông báo vào tháng 4 năm 2010 rằng họ sẽ thuê hai nhân viên để làm việc với cộng đồng nhân Linux, nhưng Greg Kroah-Hartman, người bảo trì nhân Linux hiện tại của nhánh ổn định, đã nói vào tháng 12 năm 2010 rằng ông ta lo ngại rằng Google không còn muốn đưa những thay đổi của mình vào Linux dòng chính nữa. Một số lập trình viên Android của Google tỏ ý rằng "nhóm Android thấy chán với quy trình đó" vì nhóm họ không có nhiều người và có nhiều việc khẩn cấp cần làm với Android hơn.

Vào tháng 8 năm 2011, Linus Torvalds rằng "rốt cuộc thì Android và Linux cũng sẽ trở lại với một bộ nhân chung, nhưng điều đó có thể sẽ không xảy ra trong 4 hoặc 5 năm nữa". Vào tháng 12 năm 2011, Greg Kroah-Hartman thông báo kích hoạt Dự án Dòng chính Android, nhằm tới việc đưa một số driver, bản vá và tính năng của Android ngược vào nhân Linux, bắt đầu từ Linux 3.3. Linux cũng đưa tính năng autosleep (tự nghỉ hoạt động) và wakelocks vào nhân 3.5, sau nhiều nỗ lực phối trộn trước đó. Tương tác thì vẫn vậy nhưng bản hiện thực trên Linux dòng chính cho phép hai chế độ nghỉ: bộ nhớ (dạng nghỉ truyền thống mà Android sử dụng), và đĩa (là ngủ đông trên máy tính để bàn). Việc trộn sẽ hoàn tất kể từ nhân 3.8, Google đã công khai kho mã nguồn trong đó có những đoạn thử nghiệm đưa Android về lại nhân 3.8.

Bộ lưu trữ flash trên các thiết bị Android được chia thành nhiều phân vùng, như "system" dành cho hệ điều hành và "/data" dành cho dữ liệu người dùng và cài đặt ứng dụng. Khác với các bản phân phối Linux cho máy tính để bàn, người sở hữu thiết bị Android không được trao quyền truy cập root vào hệ điều hành và các phân vùng nhạy cảm như /system được thiết lập chỉ đọc. Tuy nhiên, quyền truy cập root có thể chiếm được bằng cách tận dụng những lỗ hổng bảo mật trong Android, điều mà cộng đồng mã nguồn mở thường xuyên sử dụng để nâng cao tính năng thiết bị của họ, kể cả bị những người ác ý sử dụng để cài virus và phần mềm ác ý.

Việc Android có được xem là một bản phân phối Linux hay không vẫn còn là vấn đề gây tranh cãi, tuy được Linux Foundation và Chris DiBona, trưởng nhóm mã nguồn mở Google, ủng hộ. Một số khác, như linux-magazine.com thì không đồng ý, do Android không hỗ trợ nhiều công cụ GNU, trong đó có glibc.

### **1.7. Lịch nâng cấp**

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

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

Việc thiếu các hỗ trợ hậu mãi của nhà sản xuất và nhà mạng đã bị những nhóm người dùng và các trang tin công nghệ chỉ trích rất nhiều. Một số người viết còn nói rằng giới công nghiệp do cái lợi về tài chính đã cố tình không nâng cấp thiết bị, vì nếu thiết bị hiện tại không cập nhật sẽ thúc đẩy việc mua thiết bị mới, một thái độ được coi là "xúc phạm". The Guardian đã than phiền rằng phương cách phân phối bản nâng cấp trở nên phức tạp chính vì những nhà sản xuất và nhà mạng đã cố tình làm nó như thế. Vào năm 2011, Google đã hợp tác cùng một số hãng công nghiệp và ra mắt "Liên minh nâng cấp Android", với lời hứa sẽ nâng cấp thường xuyên cho các thiết bị trong vòng 18 tháng sau khi ra mắt. Tính đến năm 2012, người ta không còn nghe nhắc đến liên minh này nữa.

### **1.8. Cộng đồng mã nguồn mở**

Android có một cộng đồng các lập trình viên và những người đam mê rất năng động. Họ sử dụng mã nguồn Android để phát triển và phân phối những phiên bản chỉnh sửa của hệ điều hành. Các bản Android do cộng đồng phát triển thường đem những tính năng và cập nhật mới vào nhanh hơn các kênh chính thức của nhà sản xuất/nhà mạng, tuy không được kiểm thử kỹ lưỡng cũng như không có đảm bảo chất lượng; cung cấp sự hỗ trợ liên tục cho các thiết bị cũ không còn nhận được bản cập nhật chính thức; hoặc mang Android vào những thiết bị ban đầu chạy một hệ điều hành khác, như HP Touchpad. Các

bản Android của cộng đồng thường được root sẵn và có những điều chỉnh không phù hợp với những người dùng không rành rẽ, như khả năng ép xung hoặc tăng/giảm áp bộ xử lý của thiết bị. CyanogenMod là firmware của cộng đồng được sử dụng phổ biến nhất, và hoạt động như một tổ chức của số đông khác.

Trước đây, nhà sản xuất thiết bị và nhà mạng tỏ ra thiếu thiện chí với việc phát triển firmware của bên thứ ba. Những nhà sản xuất còn thể hiện lo ngại rằng các thiết bị chạy phần mềm không chính thức sẽ hoạt động không tốt và dẫn đến tốn tiền hỗ trợ. Hơn nữa, các firmware đã thay đổi như CyanogenMod đôi khi còn cung cấp những tính năng, như truyền tải mạng (tethering), mà người dùng bình thường phải trả tiền nhà mạng mới được sử dụng. Kết quả là nhiều thiết bị bắt đầu đặt ra hàng rào kỹ thuật như khóa bootloader hay hạn chế quyền truy cập root. Tuy nhiên, khi phần mềm do cộng đồng phát triển ngày càng trở nên phổ biến, và sau một thông cáo của Thư viện Quốc hội Hoa Kỳ cho phép “jailbreak” (vượt ngục) thiết bị di động, các nhà sản xuất và nhà mạng đã tỏ ra mềm mỏng hơn với các nhà phát triển thứ ba, thậm chí một số hãng như HTC, Motorola, Samsung và Sony, còn hỗ trợ và khuyến khích phát triển. Kết quả của việc này là dần dần nhu cầu tìm ra các hạn chế phần cứng để cài đặt được firmware không chính thức đã bớt đi do ngày càng nhiều thiết bị được phát hành với bootloader đã mở khóa sẵn hoặc có thể mở khóa, tương tự như điện thoại dòng Nexus, tuy rằng thông thường họ sẽ yêu cầu người dùng từ bỏ chế độ bảo hành nếu họ làm như vậy. Tuy nhiên, tuy được sự chấp thuận của nhà sản xuất, một số nhà mạng tại Mỹ vẫn bắt buộc điện thoại phải bị khóa.

Việc mở khóa và “hack” điện thoại thông minh và máy tính bảng vẫn còn là tác nhân gây căng thẳng giữa cộng đồng và công nghiệp. Cộng đồng luôn biện hộ rằng sự hỗ trợ không chính thức ngày càng trở nên quan trọng trước việc nền công nghiệp không cung cấp các bản cập nhật thường xuyên và/hoặc ngưng hỗ trợ cho chính các thiết bị của họ.

### **1.9. Bảo mật và tính riêng tư của Android**

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

này, người dùng có thể chọn đồng ý hoặc từ chối chúng, ứng dụng chỉ được cài đặt khi người dùng đồng ý.

Hệ thống hộp cát và hỏi quyền làm giảm bớt ảnh hưởng của lỗi bảo mật hoặc lỗi chương trình có trong ứng dụng, nhưng sự bối rối của lập trình viên và tài liệu hướng dẫn còn hạn chế đã dẫn tới những ứng dụng hay đòi hỏi những quyền không cần thiết, do đó làm giảm đi hiệu quả của hệ thống này. Một số công ty bảo mật, như Lookout Mobile Security, AVG Technologies, và McAfee, đã phát hành những phần mềm diệt virus cho các thiết bị Android. Phần mềm này không có hiệu quả vì cơ chế hộp cát vẫn áp dụng vào các ứng dụng này, do vậy làm hạn chế khả năng quét sâu vào hệ thống để tìm nguy cơ.

Một nghiên cứu của công ty bảo mật Trend Micro đã liệt kê tình trạng lạm dụng dịch vụ trả tiền là hình thức phần mềm ác ý phổ biến nhất trên Android, trong đó tin nhắn SMS sẽ bị gửi đi từ điện thoại bị nhiễm đến một số điện thoại trả tiền mà người dùng không hề hay biết. Loại phần mềm ác ý khác hiển thị những quảng cáo không mong muốn và gây khó chịu trên thiết bị, hoặc gửi thông tin cá nhân đến bên thứ ba khi chưa được phép. Đe dọa bảo mật trên Android được cho là tăng rất nhanh theo cấp số mũ; tuy nhiên, các kỹ sư Google phản bác rằng hiểm họa từ phần mềm ác ý và virus đã bị thổi phồng bởi các công ty bảo mật nhằm mục đích thương mại, và buộc tội ngành công nghiệp bảo mật đang lợi dụng sự sợ hãi để bán phần mềm diệt virus cho người dùng. Google vẫn giữ quan điểm rằng phần mềm ác ý thật sự nguy hiểm là cực kỳ hiếm, và một cuộc điều tra do F-Secure thực hiện cho thấy chỉ có 0,5% số phần mềm ác ý Android là len vào được cửa hàng Google Play.

Google hiện đang sử dụng bộ quét phần mềm ác ý Google Bouncer để theo dõi và quét các ứng dụng trên Cửa hàng Google Play. Nó sẽ đánh dấu các phần mềm bị nghi ngờ và cảnh báo người dùng về những vấn đề có thể xảy ra trước khi họ tải nó về máy. Android phiên bản 4.2 Jelly Bean được phát hành vào năm 2012 cùng với các tính năng bảo mật được cải thiện, bao gồm một bộ quét phần mềm ác ý được cài sẵn trong hệ thống, hoạt động cùng với Google Play nhưng cũng có thể quét các ứng dụng được cài đặt từ nguồn thứ ba, và một hệ thống cảnh báo sẽ thông báo cho người dùng khi một ứng dụng cố gắng gửi một tin nhắn vào số tính tiền, chặn tin nhắn đó lại trừ khi người dùng công khai cho phép nó.

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



Foursquare, Google Latitude, Facebook Places, và gửi những đoạn quảng cáo dựa trên vị trí. Phần mềm theo dõi của bên thứ ba như TaintDroid, một dự án nghiên cứu trong trường đại học, đôi khi có thể biết được khi nào thông tin cá nhân bị gửi đi từ ứng dụng đến các máy chủ đặt ở xa.

Bản chất mã nguồn mở của Android cho phép những nhà thầu bảo mật lấy những thiết bị sẵn có rồi điều chỉnh để sử dụng ở mức độ bảo mật cao hơn. Ví dụ như Samsung đã cộng tác với General Dynamics sau khi họ thu tóm Open Kernel Labs để xây dựng lại Jellybean trên nền bộ vi kiểm soát dành cho dự án "Knox".

## **CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH ANDROID STUDIO**

### **2.1. Sơ lược về Android Studio**

Google cung cấp một công cụ phát triển ứng dụng Android trên Website chính thức dựa trên nền tảng IntelliJ IDEA gọi là Android Studio. Android studio dựa vào IntelliJ IDEA, là một IDE tốt cho nhất hiện nay. Do đó Android Studio sẽ là môi trường phát triển ứng dụng tốt nhất cho Android.

### **2.2. Cài đặt Android Studio**

#### **2.2.1. Yêu cầu phần cứng máy tính**

##### **a) Hệ điều hành Windows**

- Microsoft® Windows® 7/8/10 (32-bit hoặc 64-bit).
- RAM tối thiểu 3 GB, khuyến nghị sử dụng RAM 8 GB; cộng 1 GB cho trình giả lập Android.
- Tối thiểu 2 GB dung lượng đĩa trống, 4 GB được đề xuất (500 MB cho IDE + 1.5 GB cho Android SDK và hình ảnh hệ thống mô phỏng).
- Độ phân giải màn hình tối thiểu 1280 x 800.

##### **b) Hệ điều hành MacOS**

- Mac® OS X® 10.10 (Yosemite) hoặc cao hơn, lên đến 10.13 (macOS High Sierra).
- RAM tối thiểu 3 GB, khuyến nghị sử dụng RAM 8 GB; cộng 1 GB cho trình giả lập Android.
- Tối thiểu 2 GB dung lượng đĩa trống, 4 GB được đề xuất (500 MB cho IDE + 1.5 GB cho Android SDK và hình ảnh hệ thống mô phỏng).

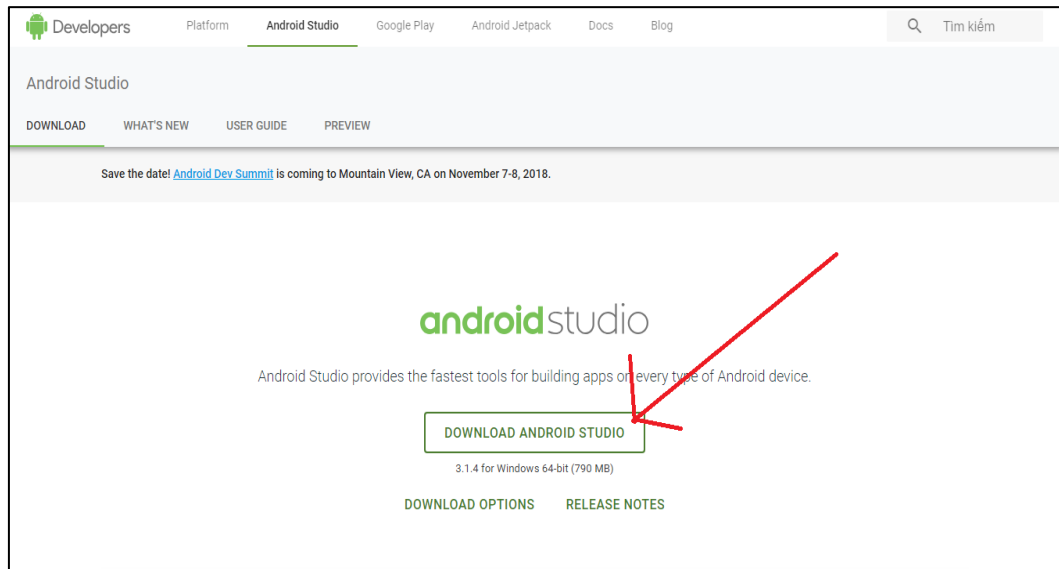
- Độ phân giải màn hình tối thiểu 1280 x 800.

**c) Hệ điều hành Linux**

- Thư viện GNU C (glibc) 2.19 hoặc mới hơn.
- RAM tối thiểu 3 GB, khuyến nghị sử dụng RAM 8 GB; cộng 1 GB cho Trình giả lập Android.
- Tối thiểu 2 GB dung lượng đĩa trống, 4 GB được đề xuất (500 MB cho IDE + 1.5 GB cho Android SDK và hình ảnh hệ thống mô phỏng).
- Độ phân giải màn hình tối thiểu 1280 x 800.

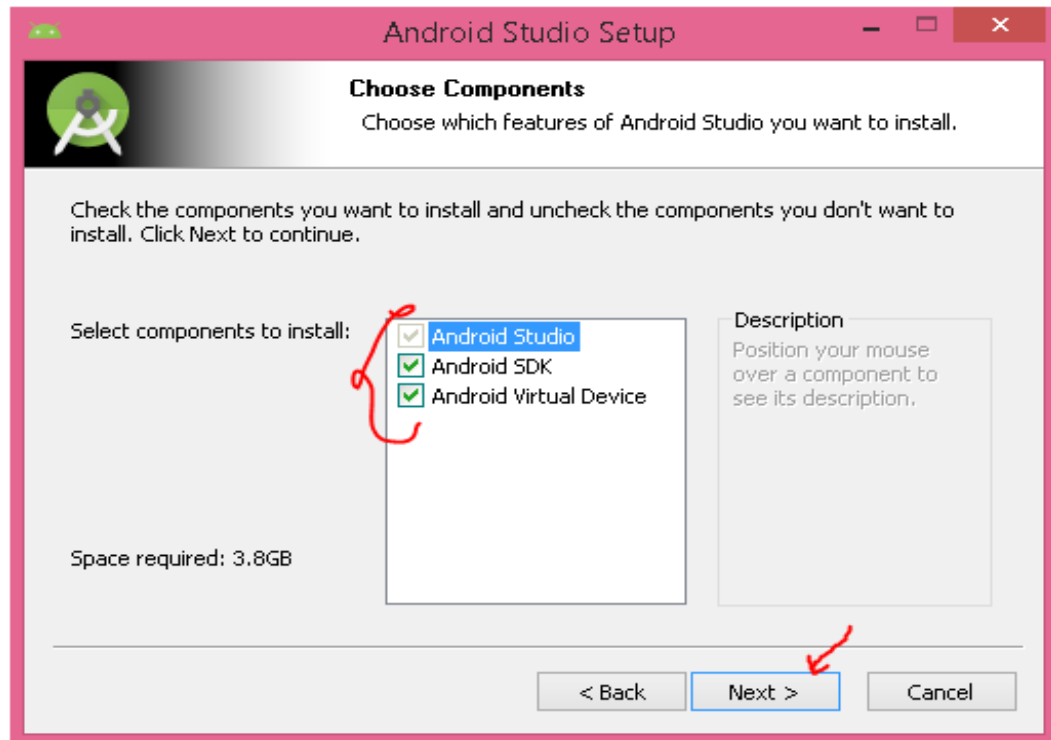
## 2.2.2. Phần mềm Android Studio

- Vào đường dẫn: “<https://developer.android.com/studio/>” để download bản mới nhất và tiến hành cài đặt như hình:



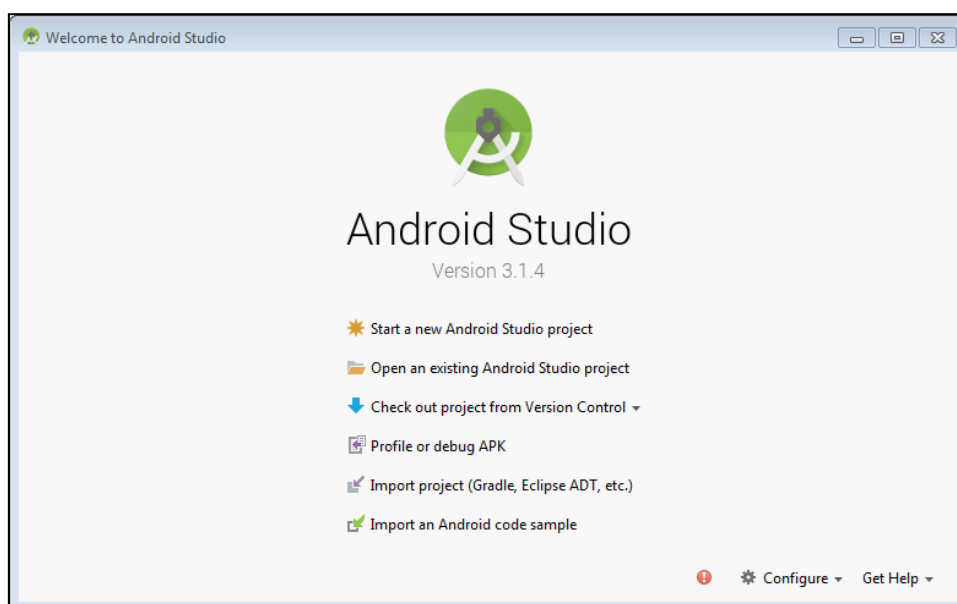
Hình 2.2.1: Trang download Android Studio

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



Hình 2.2.2: Giao diện cài đặt SDK và AVD

- Tiếp tục chọn next và agree cho đến khi hoàn tất.
- Khi việc cài đặt hoàn tất, chúng ta sẽ được như hình sau:

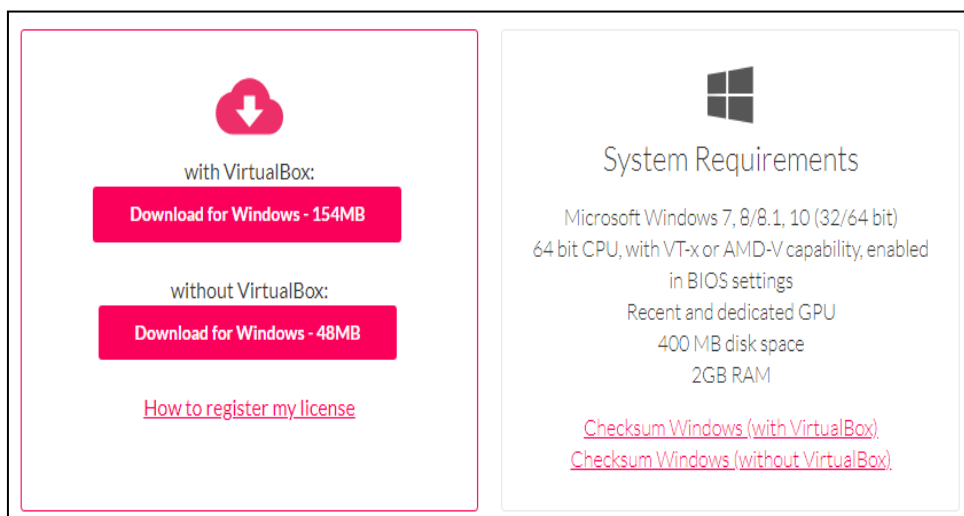


Hình 2.2.3: Giao diện cửa sổ Android Studio khi cài đặt hoàn tất

### 2.2.3. Thiết bị ảo trong Android Studio

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

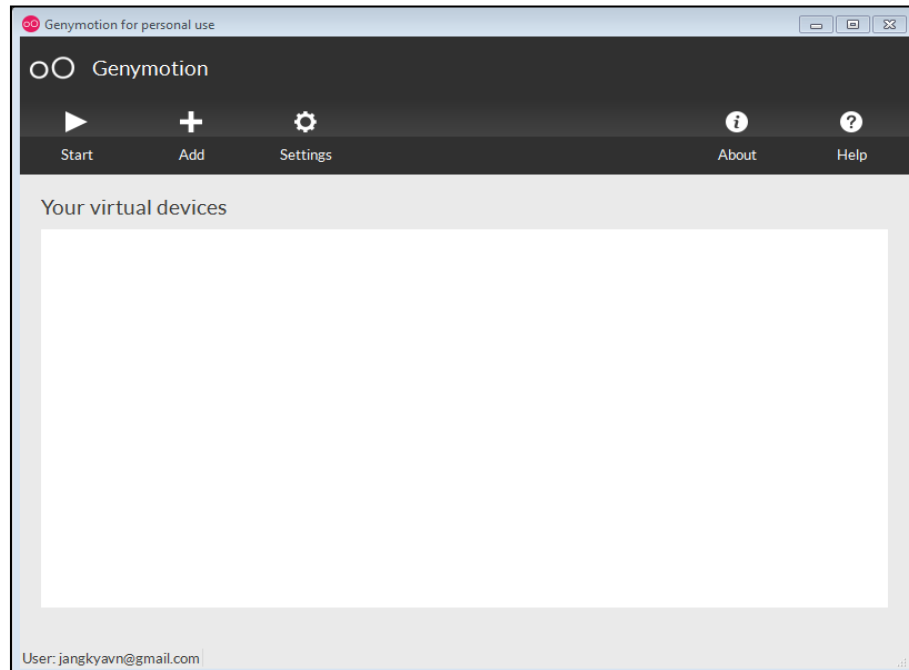
- Để cài đặt máy ảo Genymotion ta truy cập vào đường dẫn:  
<https://www.genymotion.com/download/>
- Ta phải tạo một tài khoản rồi đăng nhập vào mới thấy được mục này:



Hình 2.2.4: Giao diện download máy ảo Genymotion

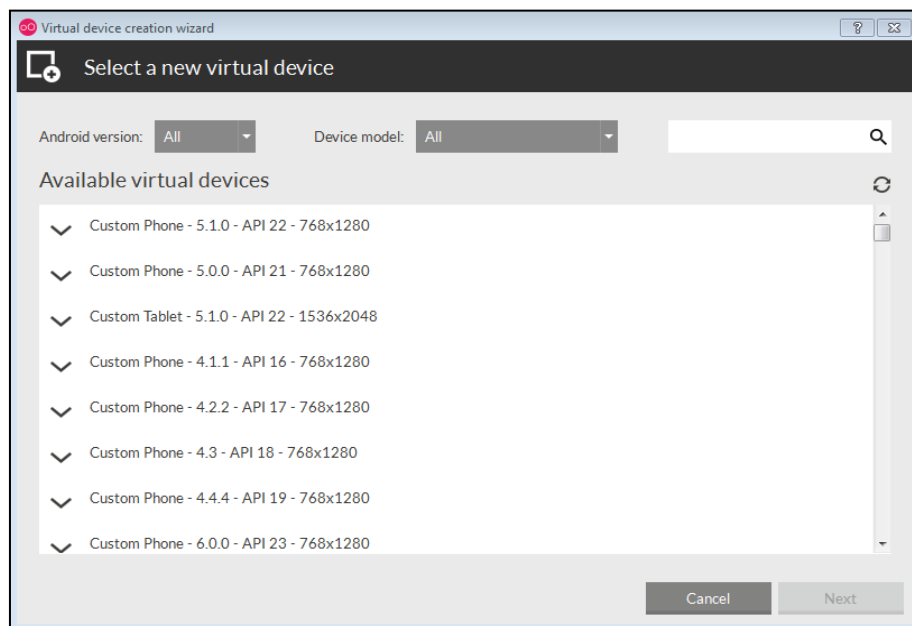
- Ở đây ta nên tải phiên bản with VirtualBox, vì chương trình tích hợp sẵn VirtualBox cho chúng ta vì máy ảo này phải có VirtualBox mới chạy được.

- Sau khi cài xong sẽ có giao diện như sau:



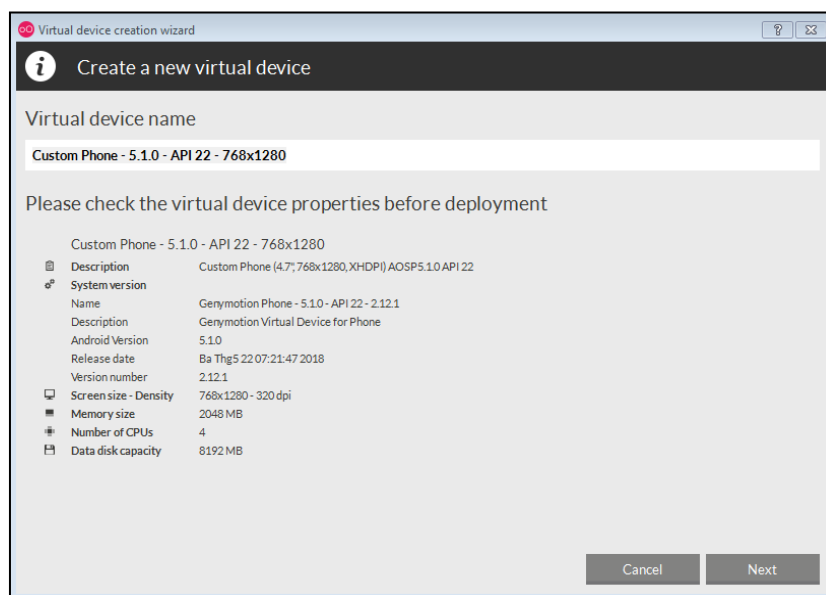
Hình 2.2.5: Giao diện chính của Genymotion

- Ta cần đăng nhập vào tạo một cái máy ảo
- Các phím chức năng của máy ảo Genymotion
  - **Start:** Bắt đầu khởi động máy ảo
  - **Add:** Tạo máy ảo
  - **Setting:** Cài đặt
- Nhấn vào Add để tạo máy ảo.



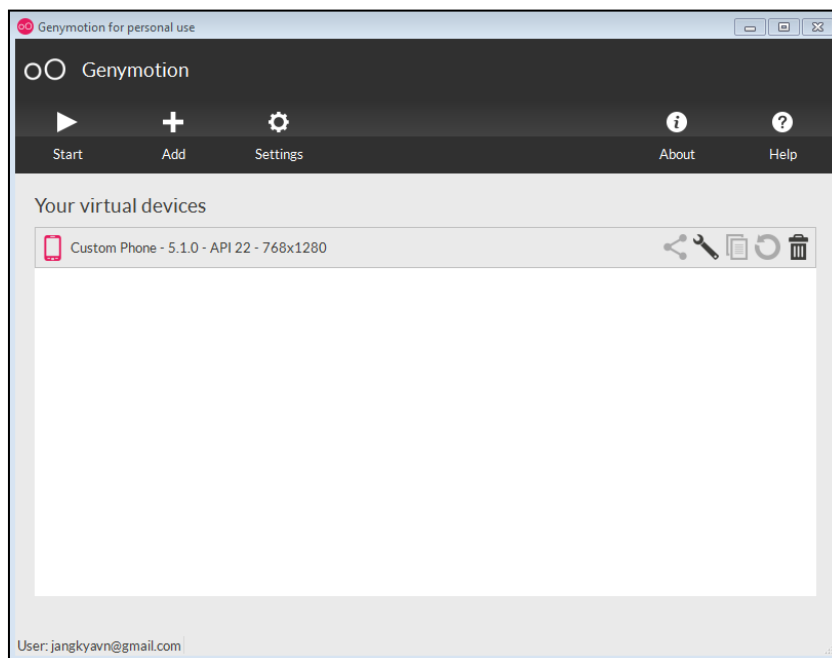
Hình 2.2.6: Giao diện chọn máy ảo

- Sau đó chọn một hoặc nhiều thiết bị theo ý muốn (như ở đây tôi chọn Custom Phone - 5.1.0 - API 22 - 768x1280 ) :
  - **Custom Phone**: là tên thiết bị máy ảo
  - **5.1.0**: là phiên bản hệ điều hành Android
  - **API**: Application Programming Interface (giao diện lập trình ứng dụng)
  - **768x1280**: Độ phân giải màn hình
- Bạn chọn thiết bị sau đó nhấn Next sẽ có giao diện thông tin thiết



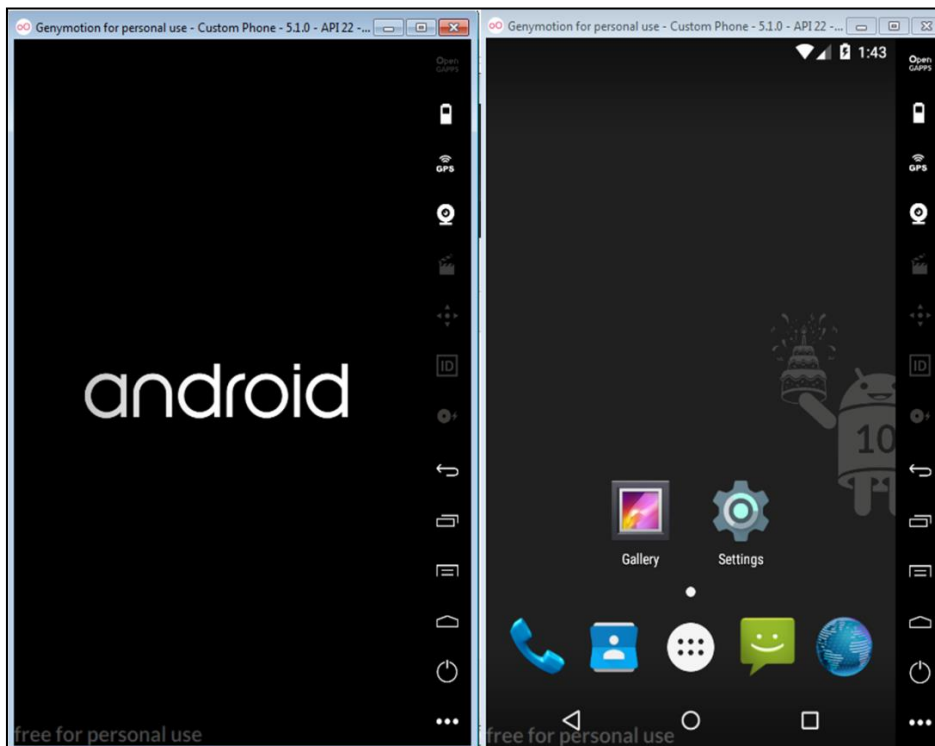
Hình 2.2.7: Giao diện xem thông tin máy ảo đang chọn

- Sau đó tiếp tục nhấn Next và Finish để hoàn thành cài máy ảo



Hình 2.2.8: Giao diện hiển thị thông tin máy ảo sau khi được tạo

- Ta tiếp tục nhấn vào thiết bị và nhấn Start để khởi động máy ảo

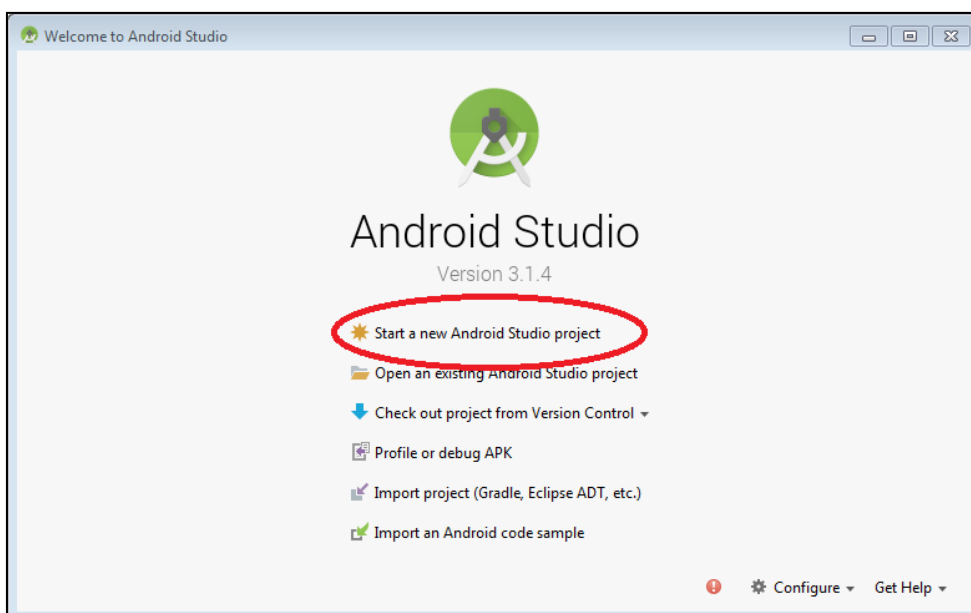


Hình 2.2.9: Giao diện màn hình đang khởi động và khi khởi động xong của máy ảo Genymotion

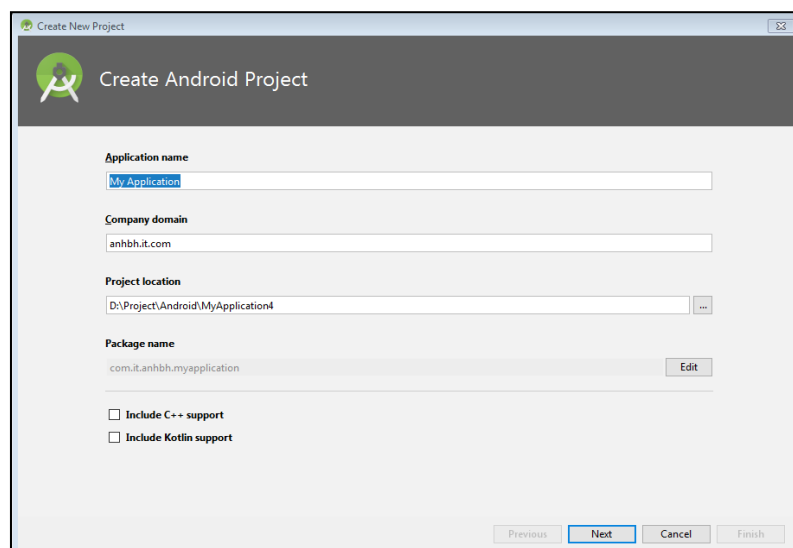
## 2.3. Cấu trúc dự án android studio

### 2.3.1. Tạo mới một Project

- Click vào dòng “Start a new Android Studio project” như hình sau:



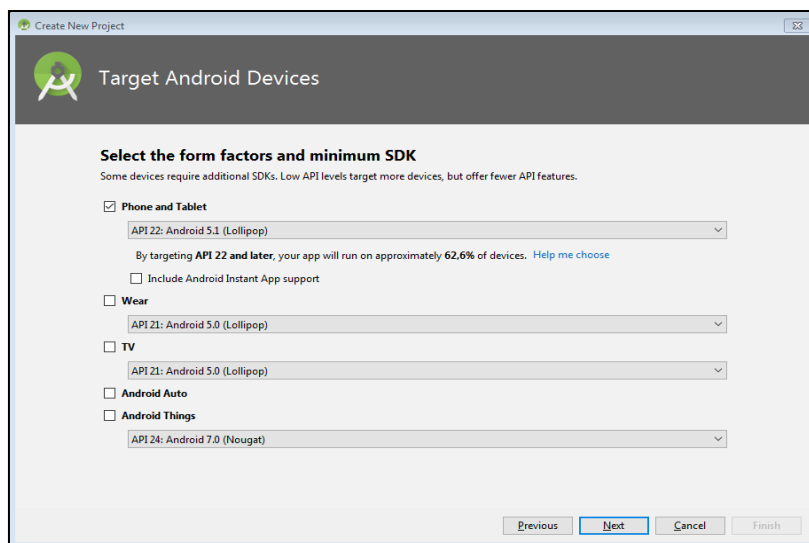
Hình 2.3.1: Giao diện cửa sổ đầu tiên khi chọn Android Studio



Hình 2.3.2: Giao diện tạo project Android

- Khi đó xuất hiện màn hình tạo project Android gồm các thành phần sau:
  - **Application Name:** Tên ứng dụng muốn đặt
  - **Company Domain:** Tên domain công ty, thường được dùng để kết hợp với tên Application để tạo thành Package (chú ý viết thường hết và có ít nhất 1 dấu chấm).
  - **Package name:** Nó sẽ tự động nối ngược Company Domain với Application name
  - **Project location:** Là nơi lưu trữ ứng dụng
- Sau đó nhấp Next để tiếp tục.

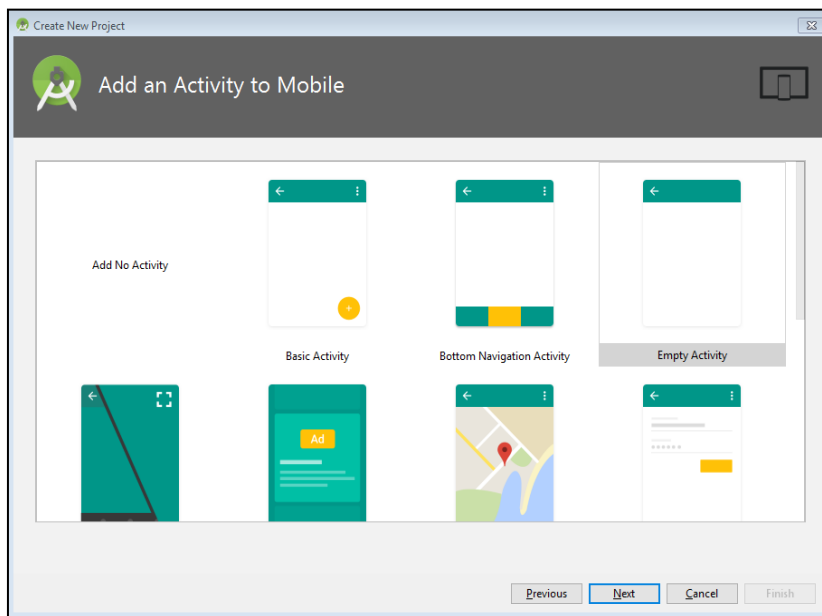
### 2.3.2. Cài đặt một Project



Hình 2.3.3: Giao diện chọn thiết bị và phiên bản hệ điều hành

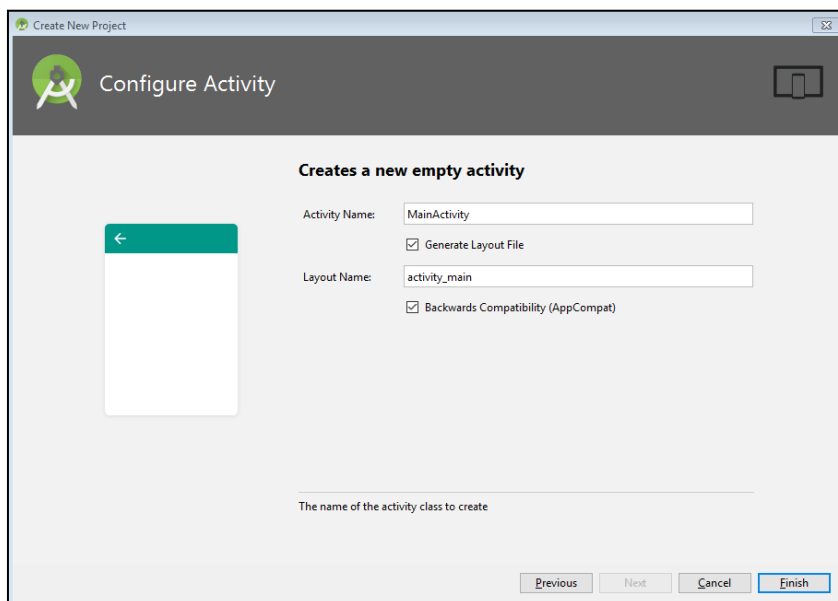


- Ở hộp thoại trên cho phép ta lựa chọn là ứng dụng sẽ được viết cho những thiết bị nào (Phone and Tablet, TV, Wear)
- Chọn phiên bản Android phù hợp rồi nhấn Next



Hình 2.3.4: Chọn kiểu màn hình cho ứng dụng

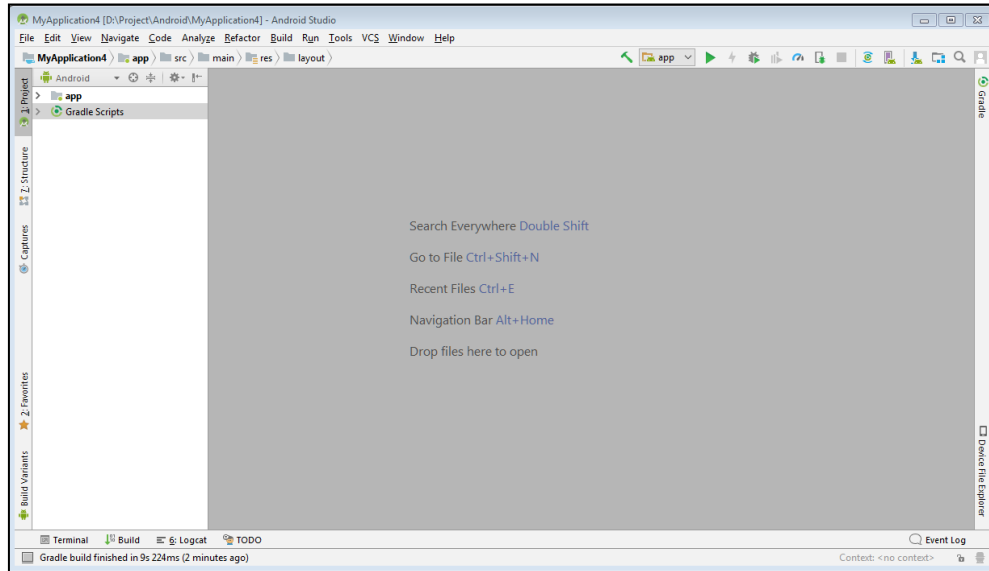
- Chọn **Empty Activity** rồi bấm Next



Hình 2.3.5: Đặt tên cho màn hình

- **Activity Name:** Tên class Activity (java) để ta viết mã lệnh
  - **Layout Name:** Tên file XML làm giao diện cho Activity Name.
- Sau khi cấu hình xong, bấm Finish

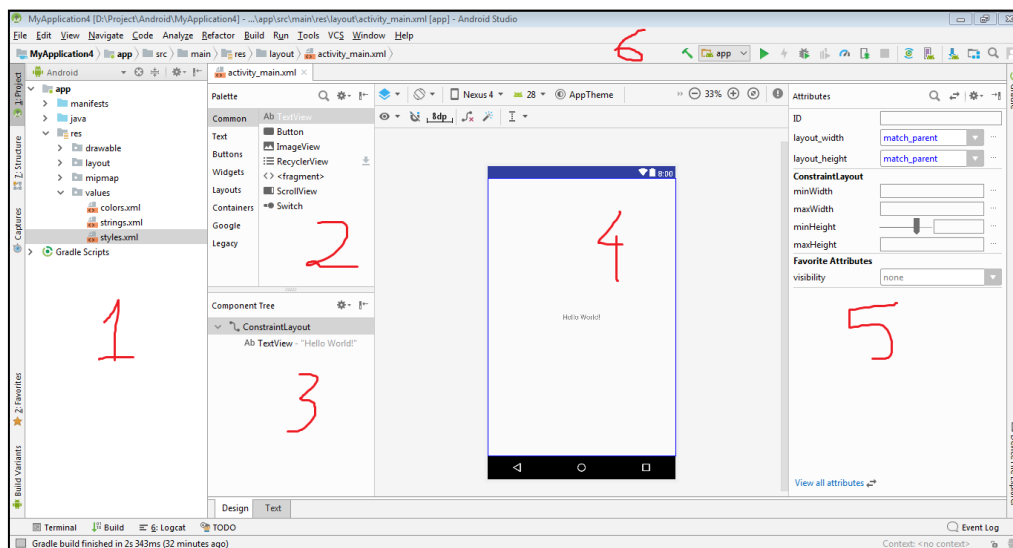
- Khi build xong mặc định có màn hình dưới đây:



Hình 2.3.6: Giao diện sau khi ra tạo thành công project

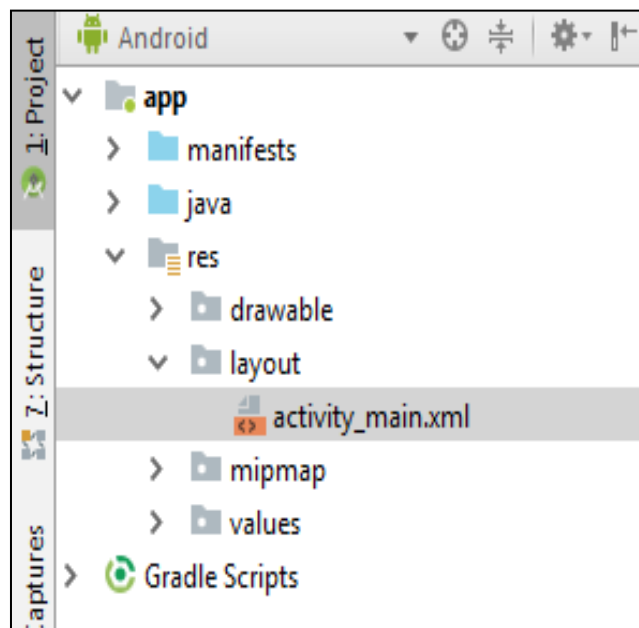
### 2.3.3. Màn hình làm việc của dự án Android Studio

- Theo mặc định Android Studio hiển thị các files trong project theo góc nhìn Android. Góc nhìn này Android Studio sẽ tổ chức các files theo 3 module:
  - **manifests**: chứa file AndroidManifest.xml.
  - **java**: chứa các file mã nguồn Java.
  - **res**: chứa tất cả các file layout, xml, giao diện người dùng (UI), ảnh
- Mặc định **activity\_main.xml** sẽ cho ta có màn hình như sau:



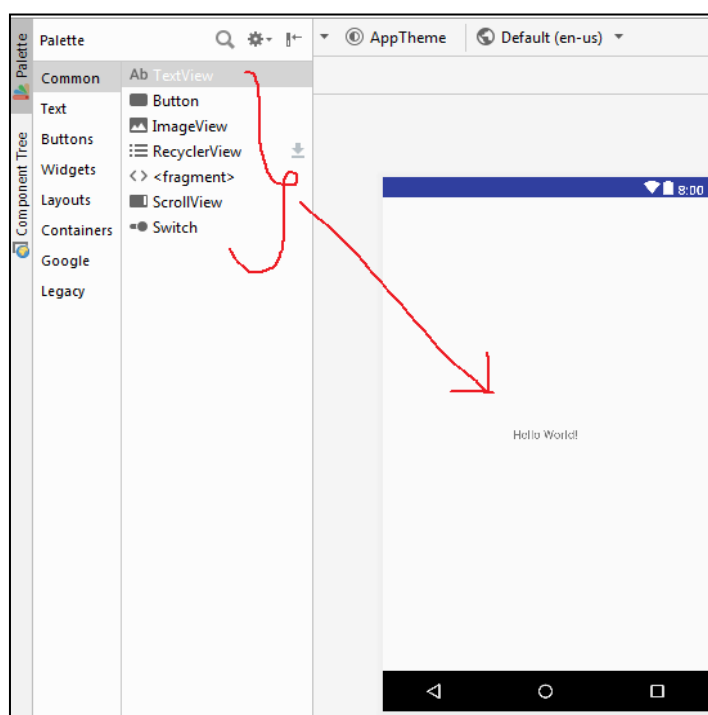
Hình 2.3.7: Các vùng làm việc trong Android Studio

- Ở hình 2.3.7 tạm thời chia làm 6 vùng làm việc mà lập trình viên thường tương tác :
  - **Vùng 1**: Là nơi cấu trúc hệ thống thông tin của Ứng dụng, Ta có thể thay đổi cấu trúc hiển thị (thường để mặc định là Android).



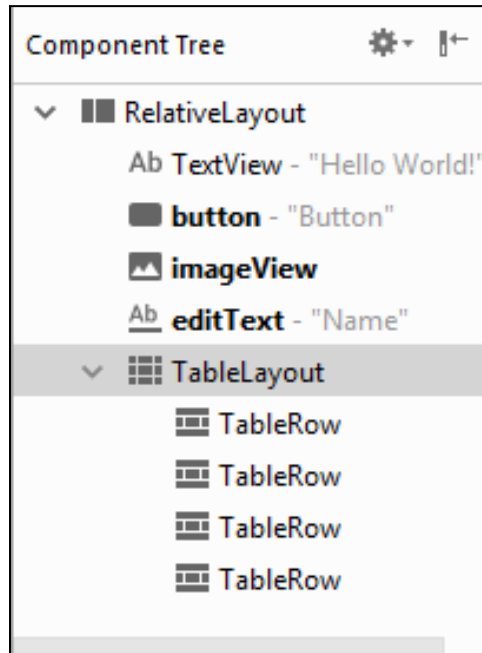
Hình 2.3.8: Cấu trúc thư mục dưới dạng Android

- **Vùng 2** : Là vùng khá quan trọng cho những bạn mới bắt đầu lập trình, nó là nơi hiển thị các Control mà Android hỗ trợ, cho phép bạn kéo thả trực tiếp vào vùng 4 để thiết kế.



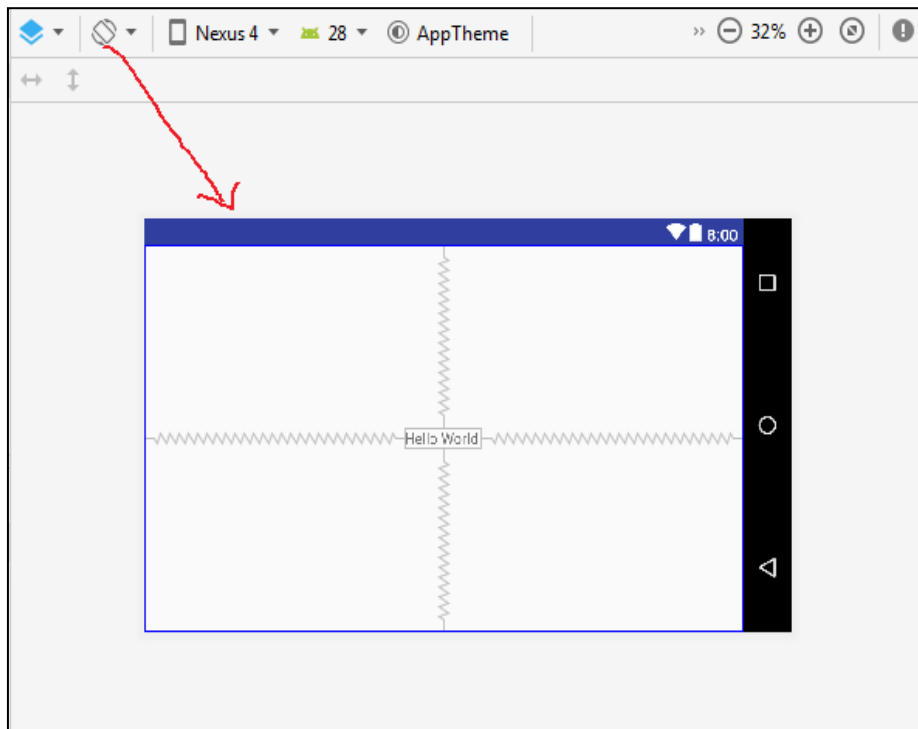
Hình 2.3.9: Các control để kéo thả vào màn hình

- **Vùng 3:** Khi màn hình có nhiều control thì vùng 4 này trở lên hữu ích, nó cho phép hiển thị giao diện theo dạng cấu trúc cây, nên ta dễ dàng quan sát và lựa chọn control khi chúng bị chồng lấp trên giao diện (vùng 4).



Hình 2.3.10: Cấu trúc giao diện theo dạng cây

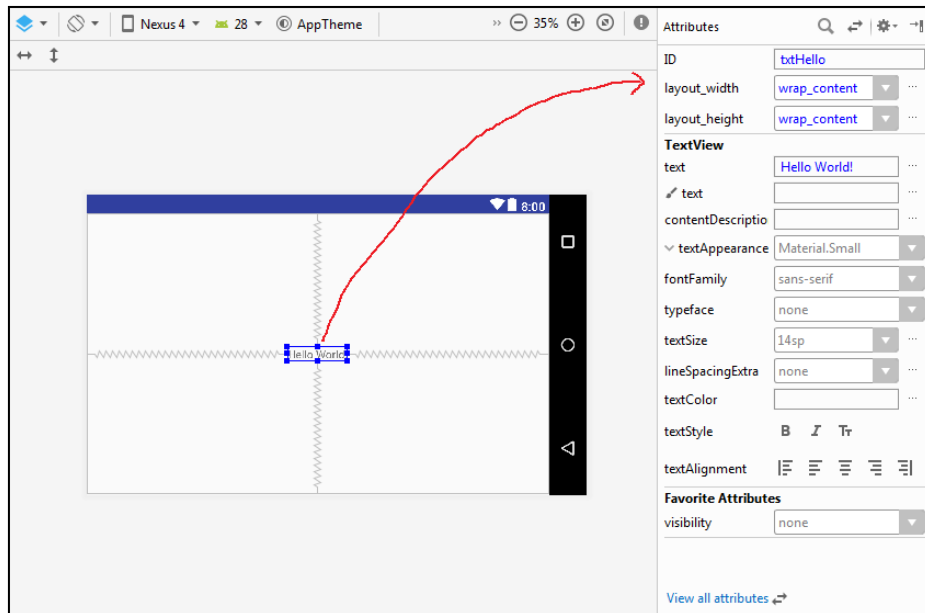
- **Vùng 4:** Là vùng giao diện thiết bị, cho phép các Control kéo thả vào đây và đồng thời cho ta hiệu chỉnh control.



Hình 2.3.11: Vùng giao diện để các control kéo thả vào

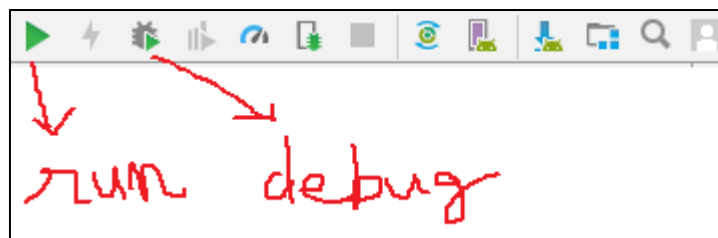
Vùng 4 ta có thể chọn cách hiển thị theo nằm ngang nằm đứng, phóng to thu nhỏ, căn chỉ control, lựa chọn loại thiết bị hiển thị....

- **Vùng 5** : Vùng này rất quan trọng, đây là vùng cho phép thiết lập trạng thái hay thuộc tính cho các Control trên giao diện



Hình 2.3.12: Vùng hiển thị thuộc tính của control nằm trên màn hình

- **Vùng 6** : Là vùng các chức năng quan trọng thường dùng trong Android Studio.



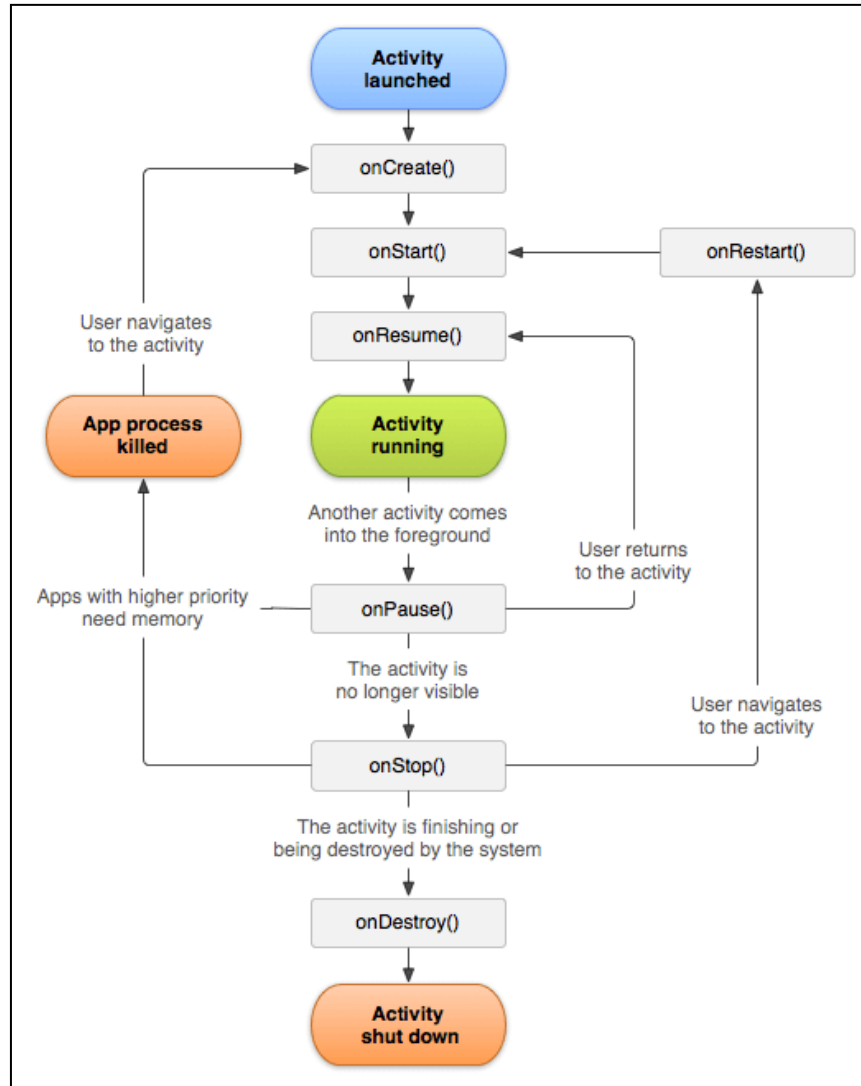
Hình 2.3.13: Các nút quan trọng trên Android Studio

## 2.4. Các thành phần trong một ứng dụng Android

Phần này tìm hiểu các thành phần cơ bản nhất tạo nên một ứng dụng Android. Hiểu về các thành phần này giúp cho có một cái nhìn tổng quan về cấu trúc ứng dụng.

- **Activity** : là một thành phần của ứng dụng Android nó sẽ biểu diễn giao diện của một màn hình nào đó trong ứng dụng của bạn. Một ứng dụng có thể sẽ có nhiều màn hình và mỗi màn hình có thể là một Activity.<sup>[2]</sup> Class Activity định nghĩa nhiều phương thức cho các sự kiện, bạn không cần phải thực thi tất cả các phương

thức này, tuy nhiên việc hiểu rõ và áp dụng chúng vào ứng dụng của bạn khá quan trọng, giúp cho ứng dụng của bạn có các hành vi như mong đợi của người dùng.



Hình 2.4.1: Vòng đời của Activity

- **onCreate()** : Nó là phương thức đầu tiên được gọi dùng để tạo một **activity** vào lần đầu tiên **activity** được gọi.
- **onStart()**: Sẽ được gọi khi nó hiện hữu với người dùng.
- **onResume()**: Sẽ được gọi khi người dùng tương tác với các ứng dụng.
- **onPause()**: Tạm dừng một **activity**, không nhận dữ liệu do người dùng nhập vào và không thể thực thi lệnh nào. Phương thức này được gọi khi **activity** hiện tại đang được tạm dừng, và **activity** trước đó đang được tiếp tục
- **onStop()**: Được gọi khi một activity đã không được nhìn thấy trong thời gian dài.
- **onDestroy()**: Được gọi trước khi hệ thống hủy **activity**.

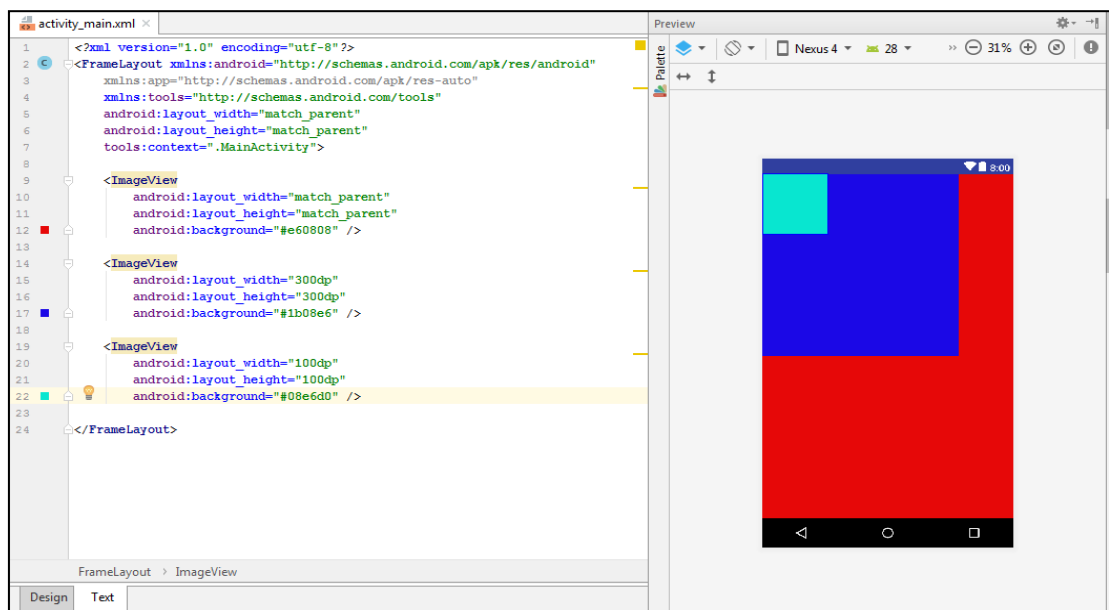
- **onRestart():** Được gọi khi activity cần được dùng trở lại sau khi bị gọi **onStop()**.<sup>[3]</sup>
- **Service:** thành phần chạy ẩn trong Android từ lúc chạy cho tới khi thiết bị cầm tay tắt đi. Service sử dụng để update dữ liệu, đưa ra các cảnh báo (Notification) và không bao giờ hiển thị cho người dùng thấy.  
VD: một chiếc MP3 chơi các bài nhạc được lưu trong hàng đợi, ngay cả khi người sử dụng chạy vào ứng dụng khác thì ứng dụng của bạn vẫn phải cần thực hiện các Services để trình diễn các nhiệm vụ mà không có một giao diện người dùng cụ thể nào cả.
- **Broadcast Receiver:** thành phần thu nhận các Intent bên ngoài gửi tới.  
VD: bạn viết 1 chương trình thay thế cho phần gọi điện mặc định của Android, khi đó bạn cần 1 BR để nhận biết các Intent là các cuộc gọi tới.
- **Content Provider:** Là một kho dữ liệu chia sẻ, nó cung cấp cách truy cập dữ liệu và giúp chia sẻ dữ liệu giữa các ứng dụng.
- **Intent:** là thành phần tối quan trọng trong một ứng dụng android, nó là một định dạng chuẩn dùng để đưa dữ liệu lên đó và khởi tạo một Activity hay truyền dữ liệu giữa các Activity, thậm chí là truyền dữ liệu giữa các ứng dụng đang thực thi, hoặc nó có thể được dùng để khởi tạo một Service.  
Ví dụ: Khi bạn mở một trang web, thì phải có một thành phần nào đó gửi dữ liệu đi để tạo một activity hiển thị trang web, đó chính là intent.
- **Notification:** đưa ra các thông báo khi ứng dụng android đang được thực thi và không làm ảnh hưởng gì đến ứng dụng thực thi. Thông thường các lập trình viên hay dùng dạng thông báo là Toast và AlertDialog.
- **View:** View là thành phần thiết yếu của giao diện người dùng, nó bao gồm các control như button, radio button, textview, checkbox, các layout,.... trong java các thành phần này được gọi là components chứ không phải view.  
Khi một view được tạo các lập trình viên thường quan tâm tới các thông số cơ bản sau:
  - Các thuộc tính cần thiết về kích thước, màu sắc,...
  - Cho phép view này được nhìn thấy hoặc bị ẩn
  - Xử lý sự kiện khi người dùng thao tác với view này (listener)
- ❖ **Lưu ý:** Activity, Service, Broadcast Receiver và Content Provider là những thành phần chính tạo nên ứng dụng Android do đó chúng phải được khai báo trong

## 2.5. Tạo giao diện chương trình trong Android Studio

### 2.5.1. Giới thiệu Android Layout

Layout là nơi chứa các control lên giao diện và mỗi layout có một cách sắp xếp các control khác nhau, vì vậy với mỗi cấu trúc giao diện khác nhau ta nên chọn layout cho phù hợp. Sau đây là một số layout cơ bản cho để ta thiết kế giao diện.

- **FrameLayout:** Là loại layout cơ bản nhất, nó sẽ được dùng nhiều khi ta sử dụng vẽ giao diện nâng cao sau này. Khi ta kéo các control vào thì mặc định các control sẽ nằm ở vị trí trên cùng bên trái. Các control khi được kéo vào framelayout sẽ bị đè lên nhau, control sau sẽ đè lên control trước. Cách duy nhất để căn các control vào giữa là sử dụng thuộc tính **android:layout\_gravity="center"**. Ta có thể tham khảo đoạn XML sau để hiểu thêm về framelayout.



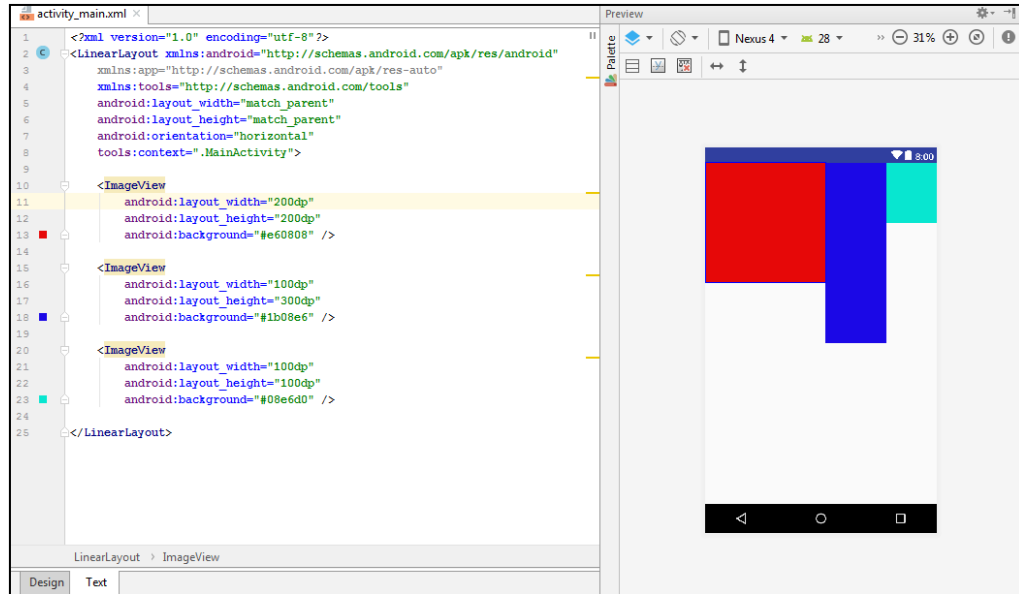
Hình 2.5.1: Giao diện sử dụng FrameLayout

- **LinearLayout:** Layout này cho phép ta vẽ giao diện theo 2 hướng, từ trái qua phải hoặc từ trên xuống dưới. Để xét chiều cho các control trong layout ta sử dụng thuộc tính **orientation**.
  - **Android:orientation="horizontal"** : Xếp các control từ trái sang phải (theo hàng).
  - **Android:orientation="vertical"** : Xếp các control từ trên xuống dưới (theo cột)



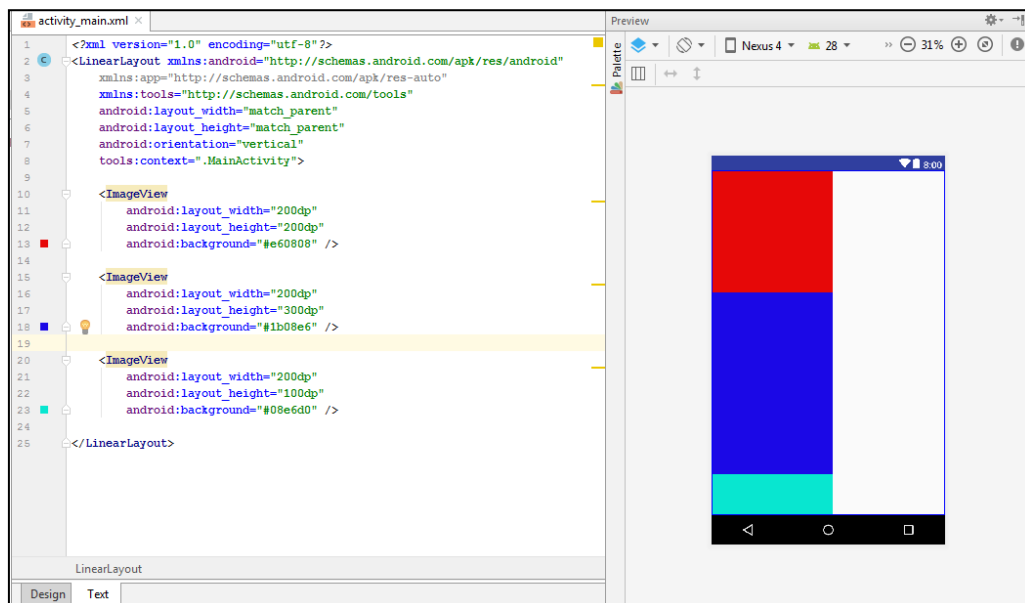
Với những giao diện có độ phức tạp vừa phải thì dùng LinearLayout là rất hiệu quả, rất thuận tiện trong thiết kế và đi bảo trì ứng dụng sau này. Sau đây là đoạn XML demo cách sử dụng layout này:

- Theo hàng



Hình 2.5.2: Giao diện LinearLayout theo chiều ngang

- Theo cột



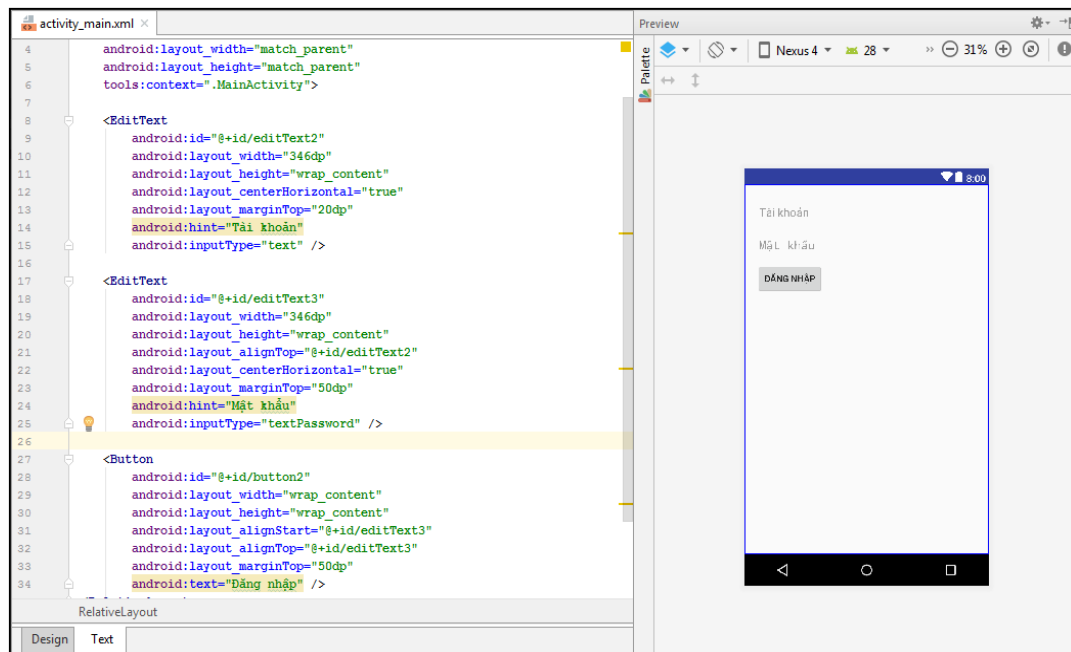
Hình 2.5.3: Giao diện LinearLayout theo chiều dọc

- **RelativeLayout:** Layout này cho phép ta sắp xếp các control theo vị trí tương đối giữa các control khác kể cả control chứa nó. Khi gặp những layout có độ phức tạp cao, có nhiều giao diện nhỏ thì sử dụng RelativeLayout là lựa chọn tốt nhất. Một

vài chú ý khi sử dụng layout này:

- Các control đều có id riêng, việc đặt tên id phải rõ ràng để hiểu.
- Các control được sắp xếp dựa vào id của các control khác.
- Các control có sự ràng buộc và tương tác với nhau nên khi thay đổi một control sẽ làm thay đổi vị trí của mọi control khác. Vì vậy rất khó trong việc bảo trì nếu giao diện quá phức tạp.

Ta có thể tham khảo đoạn XML demo sau để hình dung dễ hơn:



Hình 2.5.4: Giao diện RelativeLayout

### 2.5.2 Giới thiệu một số Android View cơ bản

- **TextView:** là view sử dụng để hiển thị text màn hình. TextView được định nghĩa bởi thẻ trong xml.
- **EditText:** là view dùng để lấy giá trị từ người dùng nhập vào. EditText được định nghĩa bởi thẻ trong xml.
- **ImageView:** là một view sử dụng rất nhiều trong ứng dụng android, ImageView sử dụng để hiển thị hình ảnh.
- **Button:** là view được sử dụng khá nhiều trong android, hầu như sử dụng ở mọi nơi cùng với EditText, TextView. Button có chức năng là làm nhiệm vụ nào đó khi mà người dùng click trong phương thức onClick.
- **ListView:** được tạo từ một danh sách các ListItem. ListItem là một dòng (row) riêng lẻ trong listview nơi mà dữ liệu sẽ được hiển thị. Bất kỳ dữ liệu nào trong

listview chỉ được hiển thị thông qua listItem. Có thể coi listview như là một nhóm cuộn của các ListItem.

### 2.5.3. Bắt và xử lý sự kiện trên giao diện

Sự kiện là một cách hữu ích để thu thập dữ liệu về sự tương tác của người dùng với các thành phần tương tác của ứng dụng. Giống như bấm vào một nút hoặc chạm vào màn hình cảm ứng, vv. Ta có thể nắm bắt những sự kiện trong chương trình và có những xử lý thích hợp theo yêu cầu.<sup>[5]</sup>

#### Có hai khái niệm liên quan đến quản lý sự kiện Android

- **Event Listeners** là một interface. Event Listeners được sử dụng để đăng ký sự kiện cho các thành phần trong UI. (Đăng ký sự kiện)
- **Event Handlers** – Là phương thức xử lý khi phát sinh sự kiện. (Xử lý sự kiện)

#### Một số sự kiện thường gặp trong Android

Event Handler	Event Listener Name
onClick()	<b>OnClickListener()</b> Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focuse trên bất kỳ widget như button, text, image vv. Chúng ta sẽ sử dụng phương onClick() để xử lý sự kiện.
onLongClick()	<b>OnLongClickListener()</b> Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focus trên bất kỳ widget như button, text, image vv. trong một hoặc nhiều giây. Chúng ta sẽ sử dụng phương onLongClick() để xử lý sự kiện.
onFocusChange()	<b>OnFocusChangeListener()</b> Sự kiện phát sinh khi widget mất focus.
onKey()	<b>OnFocusChangeListener()</b> Sự kiện phát sinh khi người dùng focus trên widget và nhấn (presse) hoặc thả (release) một phím trên thiết bị.
onTouch()	<b>OnTouchListener()</b>

	Sự kiện phát sinh khi người dùng nhấn phím, nhả phím, hoặc bất kỳ cử chỉ chuyển động trên màn hình.
onMenuItemClick()	<b>OnMenuItemClickListener()</b> Sự kiện phát sinh khi người dùng chọn một mục trong menu.
onCreateContextMenu()	<b>OnCreateContextMenuListener()</b> Sự kiện phát sinh khi người dùng chọn một mục trong menu ngữ cảnh (Context Menu)

# CHƯƠNG 3: KỸ THUẬT ĐỌC LẤY NỘI DUNG CỦA TRANG WEB VỀ ĐIỆN THOẠI

## 3.1. Tìm hiểu về dịch vụ RSS

### 3.1.1. Khái niệm

RSS (Really Simple Syndication) là dịch vụ cung cấp thông tin cực kì đơn giản, giúp người dùng có thể đọc được những tin tức cập nhật mới nhất từ một hoặc nhiều website cùng lúc mà không cần vào trực tiếp trang web đó. Theo đó, RSS không những đem lại lợi ích thiết thực cho người dùng mà còn là phương tiện để quảng bá website tiết kiệm nhất, không cần dùng đến những chiến lược quảng cáo rầm rộ tốn kém.<sup>[4]</sup>

### 3.1.2. Ví dụ

Trang web 24h.com.vn hiện tại cung cấp các nguồn kênh dữ liệu dưới đây theo định dạng chuẩn mới nhất RSS 2.0. Các nguồn kênh tin này là miễn phí cho việc sử dụng dưới mục đích cá nhân và phi lợi nhuận. Bạn chỉ việc copy và dán các địa chỉ URL này vào code của bạn để xử lý lấy dữ liệu tin tức là được.

Mục	Kênh RSS
Trang chủ	<a href="http://24h.com.vn/upload/rss/trangchu24h.rss">24h.com.vn/upload/rss/trangchu24h.rss</a>
Tin tức trong ngày	<a href="http://24h.com.vn/upload/rss/tintuctrongngay.rss">24h.com.vn/upload/rss/tintuctrongngay.rss</a>
Bóng đá	<a href="http://24h.com.vn/upload/rss/bongda.rss">24h.com.vn/upload/rss/bongda.rss</a>
World Cup 2018	<a href="http://24h.com.vn/upload/rss/worldcup2018.rss">24h.com.vn/upload/rss/worldcup2018.rss</a>
An ninh - Hình sự	<a href="http://24h.com.vn/upload/rss/anninhhinhsu.rss">24h.com.vn/upload/rss/anninhhinhsu.rss</a>
Thời trang	<a href="http://24h.com.vn/upload/rss/thoitrang.rss">24h.com.vn/upload/rss/thoitrang.rss</a>
Thời trang Hi-tech	<a href="http://24h.com.vn/upload/rss/thoitranghitech.rss">24h.com.vn/upload/rss/thoitranghitech.rss</a>
Tài chính - Bất động sản	<a href="http://24h.com.vn/upload/rss/taichinhbatdongsan.rss">24h.com.vn/upload/rss/taichinhbatdongsan.rss</a>
Ẩm thực	<a href="http://24h.com.vn/upload/rss/amthuc.rss">24h.com.vn/upload/rss/amthuc.rss</a>
Làm đẹp	<a href="http://24h.com.vn/upload/rss/lamdep.rss">24h.com.vn/upload/rss/lamdep.rss</a>
Phim	<a href="http://24h.com.vn/upload/rss/phim.rss">24h.com.vn/upload/rss/phim.rss</a>
Giáo dục - du học	<a href="http://24h.com.vn/upload/rss/giaoducduhoc.rss">24h.com.vn/upload/rss/giaoducduhoc.rss</a>
Bạn trẻ - Cuộc sống	<a href="http://24h.com.vn/upload/rss/bantrecuocsong.rss">24h.com.vn/upload/rss/bantrecuocsong.rss</a>
Ca nhạc - MTV	<a href="http://24h.com.vn/upload/rss/canhacmtv.rss">24h.com.vn/upload/rss/canhacmtv.rss</a>
Thể thao	<a href="http://24h.com.vn/upload/rss/thethao.rss">24h.com.vn/upload/rss/thethao.rss</a>
Phi thường - kỳ quặc	<a href="http://24h.com.vn/upload/rss/phithuongkyquac.rss">24h.com.vn/upload/rss/phithuongkyquac.rss</a>
Công nghệ thông tin	<a href="http://24h.com.vn/upload/rss/congnghethongtin.rss">24h.com.vn/upload/rss/congnghethongtin.rss</a>
Ô tô	<a href="http://24h.com.vn/upload/rss/oto.rss">24h.com.vn/upload/rss/oto.rss</a>
Thị trường - Tiêu dùng	<a href="http://24h.com.vn/upload/rss/thitruongtieudung.rss">24h.com.vn/upload/rss/thitruongtieudung.rss</a>
Du lịch	<a href="http://24h.com.vn/upload/rss/dulich.rss">24h.com.vn/upload/rss/dulich.rss</a>
Sức khỏe đời sống	<a href="http://24h.com.vn/upload/rss/suckhoedoisong.rss">24h.com.vn/upload/rss/suckhoedoisong.rss</a>
Cười 24h	<a href="http://24h.com.vn/upload/rss/cuoi24h.rss">24h.com.vn/upload/rss/cuoi24h.rss</a>
Thế giới	<a href="http://24h.com.vn/upload/rss/tintucquocte.rss">24h.com.vn/upload/rss/tintucquocte.rss</a>
Đời sống Showbiz	<a href="http://24h.com.vn/upload/rss/doisongshowbiz.rss">24h.com.vn/upload/rss/doisongshowbiz.rss</a>
Giải trí	<a href="http://24h.com.vn/upload/rss/giaitri.rss">24h.com.vn/upload/rss/giaitri.rss</a>

Hình 3.1.1: Các danh mục tin RSS mà trang web 24h.com.vn cung cấp

- Đây là ví dụ về định dạng RSS 2.0.

```

<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
  <title>Tin tức trong ngày - 24H RSS</title>
  <description>Tin tức trong ngày 24h hôm nay - Cập nhật tin mới nhất
trong ngày, tin nhanh 24h về
  đời sống, pháp luật, xã hội, thời sự trong nước.
</description>
  <image>
    <url>https://static.24h.com.vn/images/logo24h-128x85.gif</url>
    <title>Tin tức 24h | tin nhanh bong da | the thao | thoi trang,
giai tri vn | báo
    online</title>
    <link>https://www.24h.com.vn</link>
  </image>
  <pubDate>Sun, 14 Oct 2018 18:14:01 +0700</pubDate>
  <language>vi-vn</language>
  <generator>Tin tức 24h</generator>
  <copyright>(c) 24h.com.vn</copyright>
  <ttl>5</ttl>
  <item>
    <title>Một người tử vong trong vụ cháy xưởng gỗ ở Hà
Nội</title>
    <description>
      Cơ quan chức năng địa phương cho hay, một trong số những nạn
nhân của vụ cháy tại xưởng sofa trong khu đô thị Trung Văn (Hà Nội), đã
tử vong.
    </description>
    <pubDate>Sun, 14 Oct 2018 16:41:53 +0700</pubDate>
    <link>
      https://www.24h.com.vn/tin-tuc-trong-ngay/mot-nguoi-tu-vong-
trong-vu-chay-xuong-ghe-o-ha-noi-c46a997234.html
    </link>
    <guid>
      https://www.24h.com.vn/tin-tuc-trong-ngay/mot-nguoi-tu-vong-
trong-vu-chay-xuong-ghe-o-ha-noi-c46a997234.html
    </guid>
  </item>
</channel>
</rss>

```

## 3.2. Kỹ thuật đọc lấy nội dung của trang web về điện thoại

### 3.2.1. Quy trình thực hiện

- Đầu tiên sẽ tiến hành thêm thư viện dependence vào trong file **build.gradle**:

```

implementation 'org.jsoup:jsoup:1.11.3'
implementation 'com.squareup.picasso:picasso:2.71828'

```

```

apply plugin: 'com.android.application'

android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "com.it.anhbh.demojsoup"
        minSdkVersion 22
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    //noinspection GradleCompatible
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'org.jsoup:jsoup:1.11.3'
    implementation 'com.squareup.picasso:picasso:2.71828'
}

```

Hình 3.2.1: Thư viện cần thêm vào

- **Jsoup**: là một thư viện mạnh mẽ được sử dụng khá nhiều trong Android dùng để bóc tách các thành phần của XML thành các đối tượng trong Java Android
- **Picasso**: là một thư viện Android mã nguồn mở rất phổ biến, dùng để load hoặc hiển thị hình ảnh trong ứng dụng Android.
- Tiếp theo ta cấp quyền cho phép ứng dụng lấy dữ liệu từ Internet trong **AndroidManifest**.

`<uses-permission android:name="android.permission.INTERNET" />`

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.it.anhbh.demojsoup">
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Demo"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme"
        tools:ignore="GoogleAppIndexingWarning">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Hình 3.2.2: Dòng cấp quyền truy cập Internet

- Tiếp theo sẽ tiến hành tạo giao diện đơn giản như sau:
  - Ở **activity\_main** tạo 1 listview
  - Tạo 1 **custom\_list\_item** cho mỗi item hiển thị

## activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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">

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

## custom\_list\_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingHorizontal="5dp">

    <ImageView
        android:id="@+id/iv_thumbnail"
        android:layout_width="95p"
        android:layout_height="80dp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="6dp">

        <TextView
            android:id="@+id/tv_title"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:textSize="12sp" />

        <TextView
            android:id="@+id/tv_pubDate"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="#CCC"
            android:textSize="10sp"
            android:textStyle="italic" />

    </LinearLayout>
</LinearLayout>
```



- Tiếp theo ta sẽ tạo một class **News** để tạo ra một đối tượng lưu các dữ liệu sau khi bóc tách từng thành phần của XML bằng **Jsoup**

### News.java

```
public class News {
    private String title;
    private String link;
    private String thumbnail;
    private String pubDate;

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getLink() {
        return link;
    }

    public void setLink(String link) {
        this.link = link;
    }

    public String getThumbnail() {
        return thumbnail;
    }

    public void setThumbnail(String thumbnail) {
        this.thumbnail = thumbnail;
    }

    public String getPubDate() {
        return pubDate;
    }

    public void setPubDate(String pubDate) {
        this.pubDate = pubDate;
    }
}
```

Các thuộc tính của class **News** gồm:

- **title**: tiêu đề tin tức
- **link**: đường dẫn trang chi tiết tin tức
- **thumbnail**: ảnh tin tức
- **pubdate**: Ngày đăng tin

- Bình thường nếu sử dụng luôn **ArrayAdapter** có sẵn, thì listview chỉ hiển thị dữ liệu dạng 1 đoạn text ở trên 1 row, còn bây giờ trên một row ta có cả text và image view thì không dùng nó được mà chúng ta phải custom lại để dữ liệu có thể hiển thị được, cho nên ta sẽ tạo class **CustomAdapter**:

### CustomAdapter.java

```
public class CustomAdapter extends ArrayAdapter<News> {
    Activity context;
    int resource;
    List<News> objects;
    public CustomAdapter(@NonNull Activity context, int resource,
@NonNull List<News> objects) {
        super(context, resource, objects);
        this.context = context;
        this.resource = resource;
        this.objects = objects;
    }
    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView,
@NonNull ViewGroup parent) {
        LayoutInflater inflater =
this.context.getLayoutInflater();
        convertView = inflater.inflate(this.resource, null);
        ImageView ivThumbnail =
convertView.findViewById(R.id.iv_thumbnail);
        TextView tvTitle =
convertView.findViewById(R.id.tv_title);
        TextView tvPubDate =
convertView.findViewById(R.id.tv_pubDate);
        News news = this.objects.get(position);
        tvTitle.setText(Html.fromHtml(news.getTitle()));
        tvPubDate.setText(Utility.getPeriod(news.getPubDate()));
        Picasso.get().load(news.getThumbnail()).into(ivThumbnail);
        return convertView;
    }
}
```

- Ở đây class **CustomAdaper** kế thừa lại thẳng **ArrayAdapter** có kiểu Generic là **News**, nghĩa là nó chỉ nhận vào kiểu dữ liệu là **News** thôi. Khi bạn kế thừa thì bắt buộc bạn phải ghi đè **@Override** constructor là:

```
public CustomAdapter(@NonNull Activity context, int resource,
@NonNull List<News> objects) {
    super(context, resource, objects);

    this.context = context;
    this.resource = resource;
    this.objects = objects;
}
```

- Tiếp đến là phương thức **getView()**, đây là phương thức để khởi tạo view để add nó vào ListView, ở đây nó sẽ tạo ra các row trên listview và set dữ liệu là các News lên các row đó, thực ra chẳng cần hiểu nó là cái gì bởi vì đây là cú pháp cứng chúng ta chỉ cần học thuộc rồi dùng, dùng nhiều lần dẫn đến quen và hiểu.

```

@NonNull
@Override
public View getView(int position, @Nullable View convertView,
@NonNull ViewGroup parent) {
    LayoutInflater inflater =
    this.context.getSystemService(LayoutInflater.class);
    convertView = inflater.inflate(this.getResourceId(R.layout.item_news), null);
    TextView tvTitle =
    convertView.findViewById(R.id.tv_title);
    ImageView ivThumbnail =
    convertView.findViewById(R.id.iv_thumbnail);
    TextView tvPubDate =
    convertView.findViewById(R.id.tv_pubDate);
    News news = this.objects.get(position);
    tvTitle.setText(Html.fromHtml(news.getTitle()));
    tvPubDate.setText(Utility.getPeriod(news.getPubDate()));
    Picasso.get().load(news.getThumbnail()).into(ivThumbnail);
    return convertView;
}

```

- Vì khi gọi class **CustomAdaper** ở **MainActivity** ta sẽ truyền cho nó 1 list News nên dòng code trên sẽ có tác dụng là lấy ra từng đối tượng News một tương ứng với từng view, khi lấy được news nó sẽ có được dữ liệu của 1 News như: title, link, ... và dữ liệu này sẽ set lên trên row của listview.
- Và cuối cùng là return lại **convertView** chính là các row mà nãy giờ chúng ta đã tạo và set dữ liệu lên.
- Phần quan trọng ở MainActivity mình sẽ dùng AsyncTask để lấy dữ liệu trên dịch vụ RSS và parse nó bằng Jsoup
- Bây giờ ở MainActivity.java sẽ code như sau :
  - Code **AsyncTask** để lấy dữ liệu trên Internet
  - Parse dữ liệu dạng **Document** sang **ArrayList**
  - Set adapter cho **ListView**
  - Khởi tạo lớp kế thừa từ **AsyncTask** và chuyển đường dẫn cung cấp dữ liệu qua dịch vụ RSS để hiển thị lên giao diện.

## MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    ListView lvNews;
    CustomAdapter adapter;
    ArrayList<News> arrNews;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        lvNews = findViewById(R.id.lv_news);
        MyTask myTask = new MyTask();

myTask.execute("https://www.24h.com.vn/upload/rss/tintuctrongngay.rs
s");
    }
    class MyTask extends AsyncTask<String, Void, ArrayList<News>> {
        @Override
        protected ArrayList<News> doInBackground(String... strings)
        {
            arrNews = new ArrayList<>();
            try {
                Document document = Jsoup.connect(strings[0]).get();
                Elements elements = document.select("item");
                News news = null;

                for (Element element : elements) {
                    news = new News();
                    news.setTitle(element.select("title").text());
                    news.setThumbnail(Jsoup.parse(element.select("description").text()).
                    select("img").attr("src"));
                    news.setLink(element.select("link").text());
                    news.setPubDate(element.select("pubDate").text());

                    arrNews.add(news);
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
            return arrNews;
        }

        @Override
        protected void onPostExecute(ArrayList<News> news) {
            super.onPostExecute(news);

            adapter = new CustomAdapter(MainActivity.this,
            R.layout.custom_list_item, arrNews);
            lvNews.setAdapter(adapter);
        }
    }
}
```

- Như đã nói **Jsoup** là một thư viện của Java để cung cấp một bộ các API cho việc trích xuất định dạng XML . Có một vài phương thức cho việc đọc và phân tích một XML, trong trường hợp của chúng ta, chúng ta muốn lấy một vài thông tin từ một dữ liệu dạng XML trên server ở xa và chúng ta phải cung cấp một URL. Nếu chúng ta muốn phân tích trang như là DOM, chúng ta có:

```
Document document=Jsoup.connect(strings[0]).get();
```

- document** là một biến thuộc class **Document** để lưu trữ những tài liệu đã được lấy về. Bây giờ chúng ta đã có **Document** và chúng ta có thể tự do trích xuất thông tin. Chúng ta có thể lấy title và những thông tin khác đang được sử dụng trong các “Tag XML”.

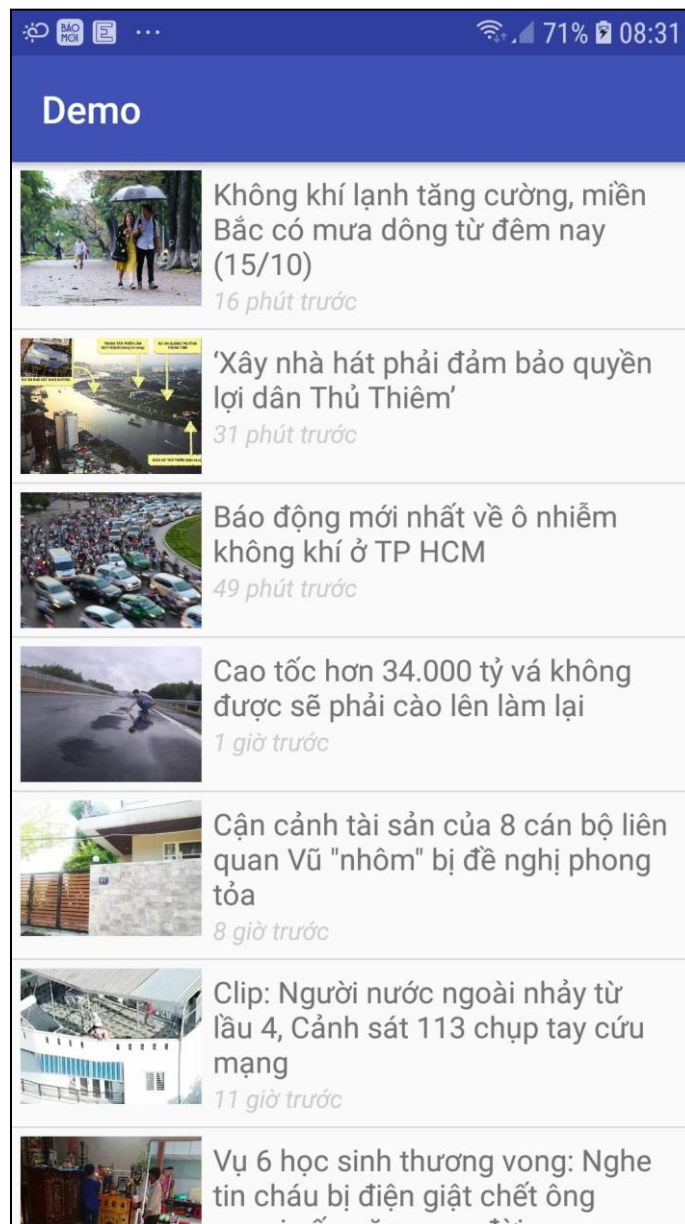
Cho ví dụ, nếu chúng ta muốn lấy tất cả các tag có tên là “item”:

```
Elements elements=document.select("item");
```

- Ta không thể sử dụng các API của **Jsoup** trong main thread vì nếu làm vậy chúng ta sẽ gặp phải một lỗi rất nghiêm trọng trong Android là **ANR** , do đó trong ví dụ này chúng ta nên sử dụng **AsyncTask**.

```
class MyTask extends AsyncTask<String, Void, ArrayList<News>> {
    @Override
    protected ArrayList<News> doInBackground(String... strings) {
        arrNews = new ArrayList<>();
        try {
            Document document = Jsoup.connect(strings[0]).get();
            Elements elements = document.select("item");
            News news = null;
            for (Element element : elements) {
                news = new News();
                news.setTitle(element.select("title").text());
                news.setThumbnail(Jsoup.parse(element.select("description").text())
                    .select("img").attr("src"));
                news.setLink(element.select("link").text());
                news.setPubDate(element.select("pubDate").text());
                arrNews.add(news);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return arrNews;
    }
    @Override
    protected void onPostExecute(ArrayList<News> news) {
        super.onPostExecute(news);
        adapter = new CustomAdapter(MainActivity.this,
            R.layout.custom_list_item, arrNews);
        lvNews.setAdapter(adapter);
    }
}
```

- **MyTask** là một class mở rộng từ **AsyncTask** cho việc phân tích định dạng XML sử dụng Jsoup.
- Rất dễ dàng và chúng ta có ngay kết quả như sau:



Hình 3.2.3: Dữ liệu hiển thị trên listview được lấy từ dịch vụ RSS

- Để xem được trang chi tiết tin tức mà người dùng ấn vào ta sẽ phải làm các công việc như sau:
  - Tạo một màn hình hiển thị trang chi tiết tin tức
  - Bắt sự kiện khi người dùng ấn vào một dòng bất kỳ trên **ListView**
  - Sử dụng **Intent** để truyền dữ liệu từ màn hình này sang màn hình khác
  - Hiện thị dữ liệu trang web bằng **WebView**

- Để tạo ra một màn hình khác ta click chuột phải vào tên project chọn: News → Activity → Empty Activity → Đặt tên là DetailActivity → Finish. Sau đó nó sẽ tạo ra 2 file **DetailActivity.class** và **activity\_detail.xml**
- Bước tiếp theo ta bắt sự kiện trong cho ListView  
Hàm **onCreate** trong **MainActivity.java**

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);
    lvNews = findViewById(R.id.lv_news);
    MyTask myTask = new MyTask();
    myTask.execute("https://www.24h.com.vn/upload/rss/tintuctrongngay.rss");
    lvNews.setOnItemClickListener(new
    AdapterView.OnItemClickListener() {

        @Override
        public void onItemClick(AdapterView<?> parent, View view, int
        position, long id) {
            Intent intent = new Intent(MainActivity.this,
            DetailActivity.class);
            intent.putExtra("link", arrNews.get(position).getLink());
            startActivity(intent);
        }
    });
}

```

- Trong hàm này ta bắt sự kiện khi người dùng nhấn vào một dòng bất kỳ ở trên **ListView** bằng cách gọi hàm **setOnItemClickListener** của đối tượng **ListView**

```

lvNews.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
    position, long id) {
        Intent intent = new Intent(MainActivity.this,
        DetailActivity.class);
        intent.putExtra("link", arrNews.get(position).getLink());
        startActivity(intent);
    }
});

```

- Trong sự kiện này nó sẽ tạo ra một hàm là **onItemClick**, hàm này sẽ được chạy khi người dùng ấn một dòng bất kỳ trên **ListView** và tham số có tên **position** chính là vị trí của tin tức trên **ListView** khi người dùng ấn vào.
- Từ cái vị trí đó ta sẽ sử dụng hàm **get**(vị trí tin tức trên **ListView**) trong đối tượng **ArrayList** để lấy được giá trị là link trang web tin tức thông qua hàm **getLink()**

- Khi đã có được link trang web tin tức thì ta chỉ việc tạo ra **Intent** để truyền dữ liệu là link trang web từ màn hình **MainActivity** sang màn hình hiện thị chi tiết tin tức **DetailActivity**

### activity\_detail.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout
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=".DetailActivity">

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

</RelativeLayout>
```

### DetailActivity.java

```
public class DetailActivity extends AppCompatActivity {
    WebView webView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);

        webView = findViewById(R.id.web_view);
        Intent intent = getIntent();
        String link = intent.getStringExtra("link");

        webView.getSettings().setJavaScriptEnabled(true);
        webView.loadUrl(link);
        webView.setWebViewClient(new WebViewClient());
    }
}
```

- Bên MainActivity đã gửi cho DetailActivity một intent và truyền dữ liệu vào intent bằng cách **putExtra** thì bên DetailActivity để nhận được dữ liệu thì phải **getIntent()** và tiến hành **getStringExtra** để lấy dữ liệu.

```
Intent intent = getIntent();
String link = intent.getStringExtra("link");
```



- Khi đã nhận được giá trị từ MainActivity thì việc bây giờ là hiển thị trang web lên **WebView**

```
webView.getSettings().setJavaScriptEnabled(true);
webView.loadUrl(link);
webView.setWebViewClient(new WebViewClient());
```

- Ở đây ta phải bật JavaScript Cho **WebView** để có thể hiển thị hết tất cả những gì có từ trang web của họ
- Để hiển thị trang web lên **WebView** chỉ việc gọi hàm **loadUrl** truyền vào link mà ta nhận được là xong.
- Cuối cùng là ta gọi hàm **setWebViewClient(new WebViewClient())** để khi load trang web tin tức thì nó vẫn nằm gọn trong ứng dụng của ta chứ không nó sẽ bật trình duyệt để hiển thị lên đó.



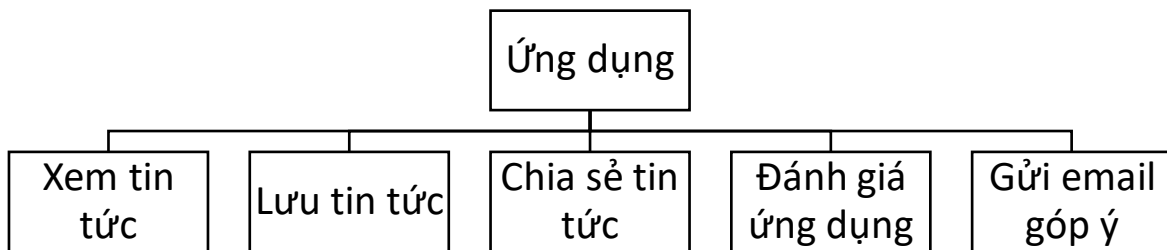
Hình 3.2.4: Trang hiển thị chi tiết tin tức

# CHƯƠNG 4: CHƯƠNG TRÌNH THỰC NGHIỆM

## 4.1. Phát biểu bài toán

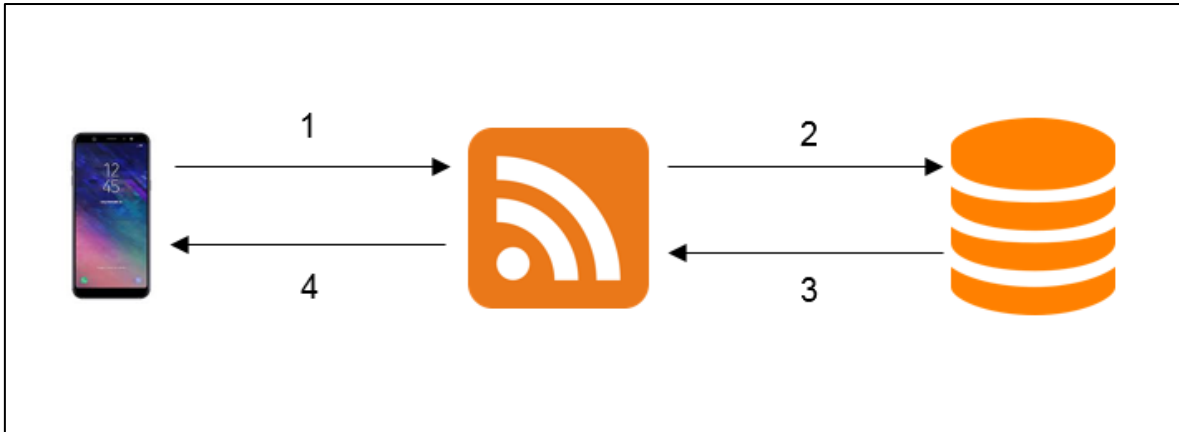
Ngày nay, vai trò của các ứng dụng đọc tin tức trên thiết bị di động ngày càng trở nên quan trọng. Khác với báo trí truyền thông có giới hạn cập nhật tin tức, các tờ báo trực tuyến đã cung cấp được sự tiện lợi trong việc cập nhật và phát hành thông tin. Về phía người dùng, họ xem thông tin mọi lúc mọi nơi với thiết bị di động của mình. Về phía người cung cấp thông tin, họ có thể dễ dàng cập nhật tin tức mới nhất, thời sự nhất thông qua dịch vụ cung cấp thông tin cực kỳ đơn giản là RSS (Really Simple Syndication). Do đó việc sử dụng các ứng dụng đọc tin tức luôn là điều cần thiết nhất hiện nay, nhằm đáp ứng nhu cầu cập nhật thông tin của mỗi người.

## 4.2. Mô hình chức năng



- **Xem tin tức:** ứng dụng cho phép người dùng chọn một tin bất kỳ để xem chi tiết nội dung tin tức đó.
- **Lưu tin tức:** lưu một tin bất kỳ mà người dùng muốn để sau này đọc, những tin tức mà người dùng đã xem vào trong bộ nhớ.
- **Chia sẻ tin tức:** ứng dụng tích hợp nút chia sẻ để người dùng thuận tiện trong việc chia sẻ tin tức lên các mạng xã hội, gmail, tin nhắn, vv...
- **Đánh giá ứng dụng:** đánh giá ứng dụng sau khi người dùng cài đặt và sử dụng ứng dụng.
- **Gửi email góp ý:** người dùng nếu muốn đóng góp ý kiến về ứng dụng thì có thể gửi email đến nhà phát triển

### 4.3. Mô hình hoạt động



#### Các bước hoạt động:

- **Bước 1:** Ứng dụng gọi tới dịch vụ RSS thông qua đường dẫn
- **Bước 2:** Dịch vụ RSS gửi yêu cầu từ client về cho cơ sở dữ liệu xử lý
- **Bước 3:** Cơ sở dữ liệu truy vấn dữ liệu trả về kết quả sau khi xử lý xong về cho dịch vụ RSS
- **Bước 4:** Dịch vụ RSS trả về nội dung tin tức dạng XML cho ứng dụng để bóc tách dữ liệu hiển thị lên màn hình.

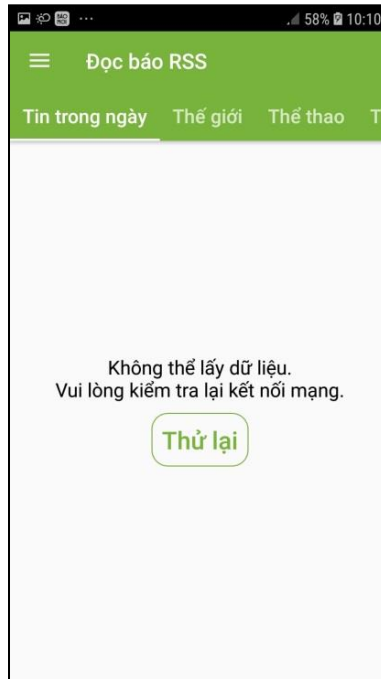
### 4.4. Giao diện ứng dụng

- Giao diện màn hình chính hiển thị danh sách tin tức của từng thể loại



Hình 4.4.1: Giao diện màn hình chính

- Là chương trình lấy dữ liệu trên internet lên phân kiểm tra kết nối internet của thiết bị vô cùng quan trọng.

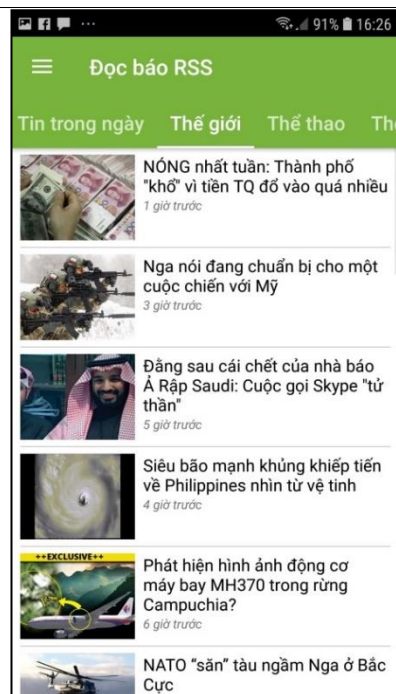


Hình 4.4.2: Màn hình hiển thị khi thiết bị không kết nối internet

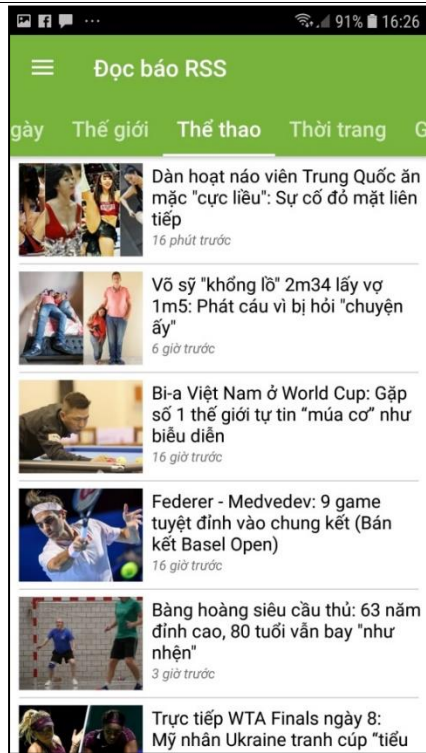
- Người dùng có thể xem các thể loại tin tức khác bằng cách chọn vào tên thể loại hoặc vuốt sang ngang.



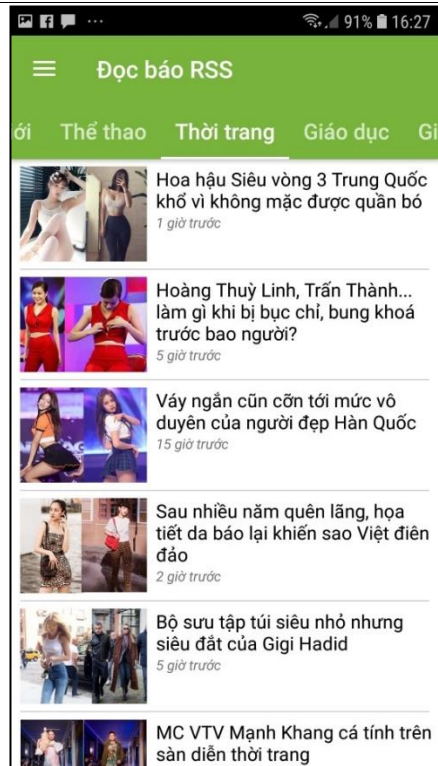
Hình 4.4.3: Giao diện màn hình thể loại tin trong ngày



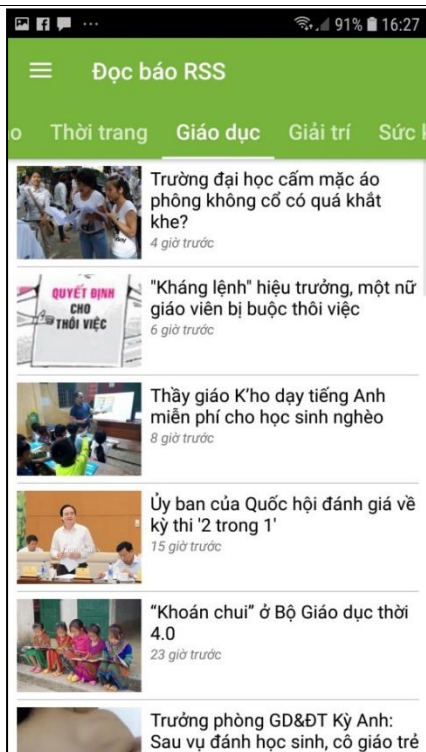
Hình 4.4.4: Giao diện màn hình thể loại tin thế giới



Hình 4.4.5: Giao diện màn hình thể loại tin thể thao



Hình 4.4.6: Giao diện màn hình thể loại tin thời trang



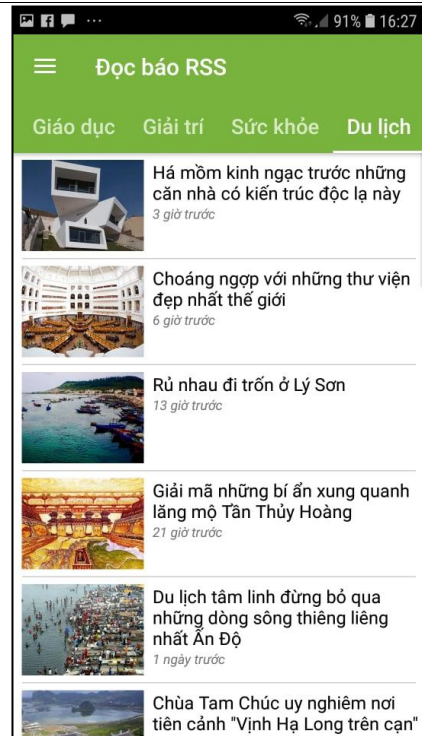
Hình 4.4.7: Giao diện màn hình thể loại tin giáo dục



Hình 4.4.8: Giao diện màn hình thể loại tin giải trí



Hình 4.4.9: Giao diện màn hình thể loại tin sức khỏe

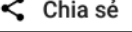


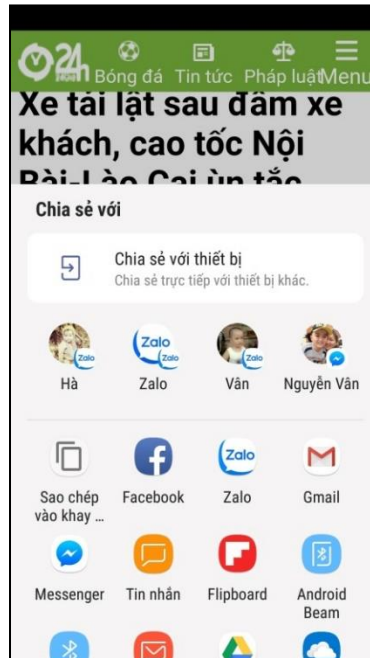
Hình 4.4.10: Giao diện màn hình thể loại tin du lịch

- Khi người dùng chọn bất kỳ tin tức nào muốn đọc trên màn hình thì ứng dụng sẽ chuyển đến màn hình hiển thị chi tiết tin tức





Hình 4.4.11: Giao diện màn hình chi tiết tin tức

- Để chia sẻ tin tức với các ứng dụng khác người dùng chọn vào nút  ở phía dưới cùng của giao diện màn hình chi tiết tin tức



Hình 4.4.12: Giao diện cửa sổ chia sẻ tin tức với các ứng dụng khác


- Người dùng muốn lưu tin tức yêu thích để xem lại sau này thì chọn  ở phía dưới cùng của giao diện màn hình chi tiết tin tức. Để xóa tin đã lưu người dùng chọn nút lại nút  lần nữa.

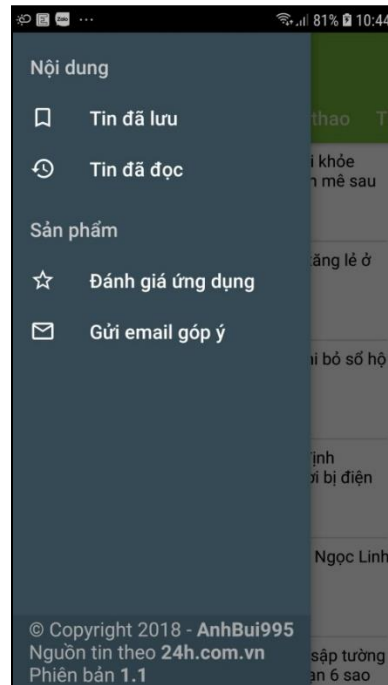


Hình 4.4.13: Thông báo hiển thị lưu tin thành công



Hình 4.4.14: Thông báo hiển thị xóa tin đã lưu thành công

- Khi người dùng chọn vào biểu tượng  trên màn hình thì sẽ xuất hiện menu hiển thị các tính năng hữu ích có trong ứng dụng



Hình 4.4.15: Menu hiển thị các tính năng của ứng dụng

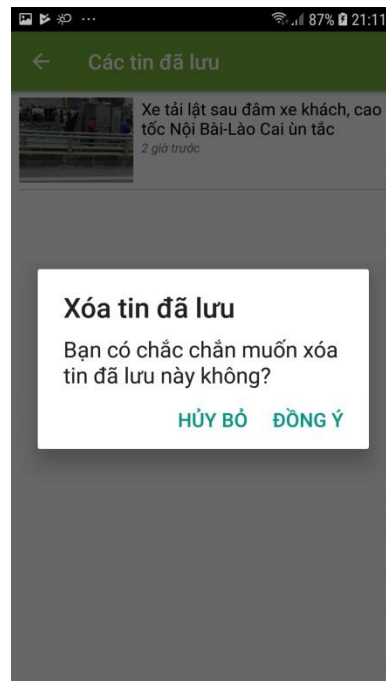
- Khi muốn xem các tin tức mà người dùng đã lưu thì chọn mục **Tin đã lưu** ở trong menu



Hình 4.4.16: Giao diện màn hình các tin đã lưu



- Nếu người dùng muốn xóa tin đã lưu trên danh sách tin tức thì chỉ cần nhấn giữ một dòng tin bất kỳ muốn xóa khoảng 1,5 giây thì lập tức xuất hiện thông báo như hình dưới đây:



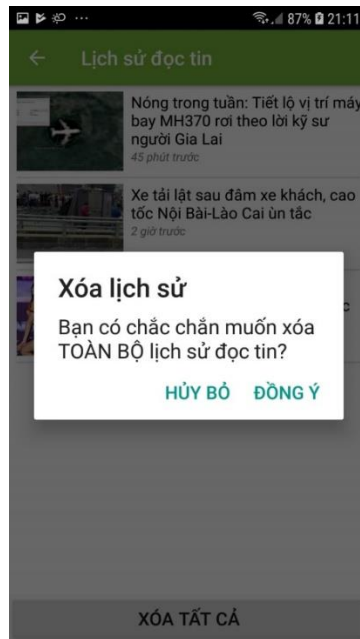
Hình 4.4.17: Giao diện cửa sổ thông báo xác nhận xóa tin đã chọn

- Để xem danh sách các tin tức đã người dùng đã xem thì chọn mục **Tin đã đọc** ở trong menu



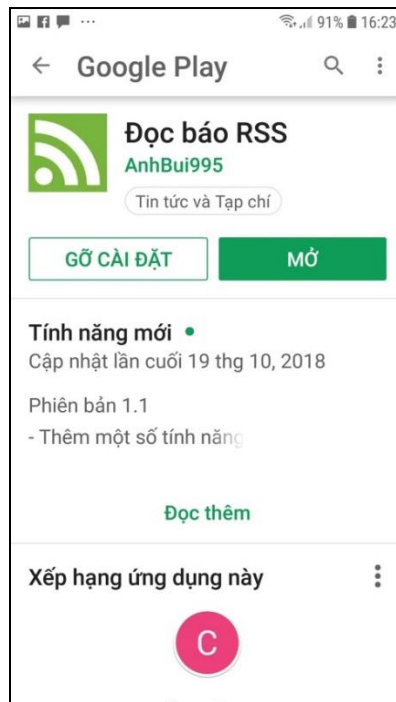
Hình 4.4.18: Giao diện màn hình các tin đã đọc

- Người dùng muốn xóa tất cả tin tức đã đọc thì ấn nút **XÓA TẤT CẢ** ở phía dưới cùng của giao diện màn hình. Lúc này sẽ xuất hiện thông báo để xác nhận xem người dùng có muốn xóa tất cả hay không.



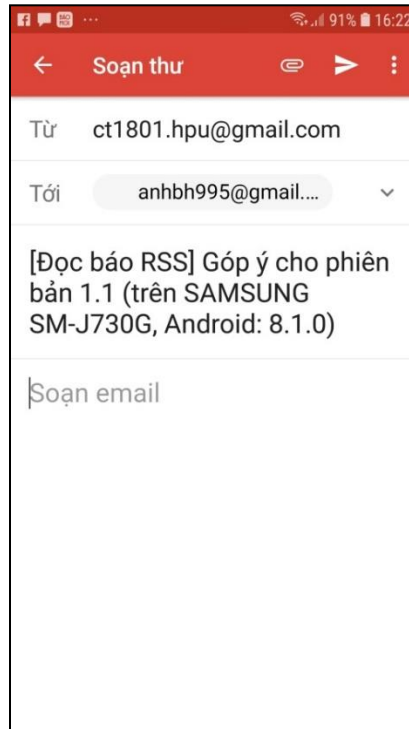
Hình 4.4.19: Giao diện cửa sổ thông báo xác nhận xóa tất cả tin đã đọc

- Sau khi người dùng đã sử dụng ứng dụng thì có thể chọn mục **Đánh giá ứng dụng** ở trong menu để đánh giá chất lượng của ứng dụng trên Google Play



Hình 4.4.20: Giao diện màn hình khi ứng dụng có ở trên Google Play

- Người dùng có thể chọn mục **Gửi email góp ý** ở trong menu để đóng góp ý kiến về ứng dụng trên thiết bị của họ.



Hình 4.4.21: Giao diện màn hình ứng dụng Gmail

## KẾT LUẬN

Sau một thời gian tìm hiểu đề tài “Xây dựng ứng dụng Android đọc báo mạng qua dịch vụ RSS” em đã thực hiện được nội dung đề tài của theo yêu cầu đặt ra. Với mục đích là tìm hiểu về hệ điều hành Android, hiểu biết về môi trường lập trình Android Studio và tìm hiểu các kỹ thuật để xây dựng một ứng dụng hoàn chỉnh trên thiết bị thật. Thông qua việc tìm hiểu và nắm bắt lý thuyết về hệ điều hành Android, bước đầu đã xây dựng thành công chương trình thực nghiệm với các chức năng cơ bản.

Trong quá trình tìm hiểu lý thuyết về hệ điều hành Android và xây dựng ứng dụng em đã thu được kết quả sau:

- Hiểu và sử dụng được các lớp đối tượng, phương thức hỗ trợ lập trình trên nền hệ điều hành Android.
- Hiểu được về dịch vụ RSS
- Hiểu được cách đọc nội dung tin tức từ trang web về điện thoại
- Xây dựng thành công ứng dụng với các chức năng như: xem tin tức, lưu tin tức, chia sẻ tin tức, đánh giá ứng dụng, gửi email góp ý.
- Đóng gói ứng dụng thành file .apk cho phép cài đặt trên điện thoại Android.
- Đưa thành công ứng dụng lên Google Play cho phép người dùng tìm kiếm và cài đặt ứng dụng trên thiết bị thật.

Do thời gian có hạn cùng với kiến thức của bản thân còn hạn chế nên em chưa tìm hiểu sâu về các kiến thức liên quan tới hệ điều hành Android nên chương trình vẫn còn rất nhiều hạn chế, em sẽ cố gắng phát triển thêm để chương trình thân thiện, dễ sử dụng và nhiều chức năng hơn.

## TÀI LIỆU THAM KHẢO

- [1]. [https://vi.wikipedia.org/wiki/Android\\_\(hệ\\_điều\\_hành\)](https://vi.wikipedia.org/wiki/Android_(hệ_điều_hành))
- [2]. <https://thangcoder.com/lap-trinh-android/hoc-lap-trinh-android-can-ban/activity-va-vong-doi-cua-no-trong-lap-trinh-android>
- [3]. <https://laptrinhstuduy.wordpress.com/2014/04/27/activity-trong-android/>
- [4]. <http://support.digisoft.com.vn/kien-thuc-web/rss-la-gi.aspx>
- [5]. <http://giasutinhoc.vn/lap-trinh-android-co-ban/xu-ly-su-kien-trong-android-bai-3/>