

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

-----o0o-----



ISO 9001 : 2008

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-----o0o-----

XÂY DỰNG ỨNG DỤNG ANDROID TRUY XUẤT CƠ SỞ DỮ LIỆU

ĐỒ ÁN TỐT NGHIỆP HỆ ĐẠI HỌC CHÍNH QUY

Ngành: Công nghệ Thông tin

HẢI PHÒNG 2016

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

-----o0o-----

XÂY DỰNG ỨNG DỤNG ANDROID TRUY XUẤT CƠ SỞ DỮ LIỆU

ĐỒ ÁN TỐT NGHIỆP HỆ ĐẠI HỌC CHÍNH QUY

Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Nguyễn Văn Tuyên

Giáo viên hướng dẫn: Phùng Anh Tuấn

Mã sinh viên: 1112101006

HẢI PHÒNG 2016

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

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
-----o0o-----

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Nguyễn Văn Tuyên

Mã số: 1112101006

Lớp:CT1501

Ngành: Công nghệ Thông tin

Tên đề tài: Xây dựng ứng dụng Android truy xuất cơ sở dữ liệu

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp.

a. Nội dung:

- Tìm hiểu hệ điều hành Android.
- Tìm hiểu môi trường lập trình Android Studio.
- Tạo lập cơ sở dữ liệu trên internet.
- Ứng dụng Android Studio để xây dựng ứng dụng truy xuất cơ sở dữ liệu internet.

b. Các yêu cầu cần giải quyết:

- Nắm được một số khái niệm cơ bản của hệ điều hành Android.
- Tải và cài đặt môi trường lập trình ứng dụng cho thiết bị di động Android Studio.
- Tạo lập cơ sở dữ liệu trên hosting.
- Tìm hiểu kỹ thuật lập trình web-service truy xuất cơ sở dữ liệu internet theo yêu cầu từ ứng dụng Android.
- Tìm hiểu một số kỹ thuật lập trình thao tác với cơ sở dữ liệu internet (xem, xóa, sửa,...) từ ứng dụng Android.
- Sử dụng công cụ lập trình Android Studio, các kỹ thuật lập trình xây dựng chương trình thực nghiệm truy xuất cơ sở dữ liệu internet từ thiết bị di động Android.
- Đóng gói chương trình ứng dụng cho phép tải về từ internet và cài đặt trực tiếp trên thiết bị di động Android.

2. Các số liệu cần thiết để thiết kế, tính toán

3. Địa điểm thực tập

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: Phùng Anh Tuấn

Học hàm, học vị: Thạc sỹ

Cơ quan công tác: Trường Đại học Dân lập Hải Phòng

Nội dung hướng dẫn:

Tìm hiểu hệ điều hành Android.

- Tìm hiểu môi trường lập trình Android Studio.

- Tạo lập cơ sở dữ liệu trên internet.

- Lập trình web-service

- ứng dụng Android Studio để xây dựng chương trình ứng dụng truy xuất cơ sở dữ liệu internet từ thiết bị di động Android.

- Đóng gói ứng dụng cho phép tải về từ internet và cài đặt trực tiếp trên thiết bị di động Android.

Đề tài tốt nghiệp được giao ngày 03 tháng 10 năm 2016

Yêu cầu phải hoàn thành trước ngày 24 tháng 12 năm 2016

Đã nhận nhiệm vụ: Đ.T.T.N

Sinh viên

Đã nhận nhiệm vụ: Đ.T.T.N

Cán bộ hướng dẫn Đ.T.T.N

Hải Phòng, ngàytháng.....năm 2016

HIỆU TRƯỞNG

GS.TS.NGŨT Trần Hữu Nghị

PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

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

.....
.....
.....
.....
.....

2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp):

.....
.....
.....
.....
.....

1. Cho điểm của cán bộ hướng dẫn:

(Điểm ghi bằng số và chữ)

.....
.....

Ngày.....tháng.....năm 2016

Cán bộ hướng dẫn chính

(Ký, ghi rõ họ tên)

Mục Lục

LỜI CẢM ƠN	12
Chương 1: GIỚI THIỆU HỆ ĐIỀU HÀNH ANDROID	13
1.1 Giới thiệu hệ điều hành Android	13
1.2. Lịch sử phát triển	14
1.3. Giao diện Android.....	15
1.4. Ứng dụng.....	17
1.5. Quản lý bộ nhớ.....	17
1.6. Nhân Linux	18
1.7. Lịch nâng cấp.....	20
1.8. Cộng đồng mã nguồn mở.....	21
1.9. Bảo mật và tính riêng tư.....	22
1.10. Giấy phép phát hành	23
1.11. Đón nhận	24
Chương 2: KỸ THUẬT LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI ANDROID STUDIO	26
2.1. Sơ lược về Android Studio	26
2.1.1. Cài đặt android studio	26
2.1.2. Cấu trúc dự án android studio.....	30
2.1.3. Tạo giao diện chương trình trong android studio	38
2.2. Kỹ thuật lập trình cơ sở dữ liệu internet trên thiết bị di động	44
2.2.1. Dịch vụ web và cơ sở dữ liệu trên internet.....	44
2.2.2. Một số kỹ thuật lập trình cơ sở dữ liệu trên internet	55
Chương 3: CHƯƠNG TRÌNH THỰC NGHIỆM.....	70
3.1. Bài toán	70
3.2. Thiết kế cơ sở dữ liệu.....	70

3.3. Mô hình chương trình	71
3.4. Giao diện chương trình	72
3.5. Kết quả đạt được, hạn chế và hướng phát triển ứng dụng	73
KẾT LUẬN.....	75
TÀI LIỆU THAM KHẢO	76

LỜI CẢM ƠN

Để đồ 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 đồ 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 đồ án: "Xây dựng ứng dụng android truy xuất cơ sở dữ liệu". 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 đồ 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, đồ 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.

Xin chân thành cảm ơn!

Sinh viên

Nguyễn Văn Tuyền

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.

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),[20] 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, bán 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 rôbốt 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 5.0 Lollipop. 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.

Năm 2014, Google công bố Android Wear, hệ điều hành dành cho các thiết bị đeo được.

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 mẫu 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.

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 nhờ 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 tháng 9 năm 2012, có hơn 675.000 ứng dụng dành cho Android, và số lượng ứng dụng tải về từ Cửa hàng Play ước tính đạt 25 tỷ.

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 từ 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, nhắm 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 6.0 Marshmallow.

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".

1.10. Giấy phép phát hành

Mã nguồn của Android được cấp phép theo các giấy phép phần mềm mã nguồn mở tự do. Google đưa phần lớn mã nguồn (bao gồm cả các lớp mạng và điện thoại) theo Giấy phép Apache phiên bản 2.0, và phần còn lại, các thay đổi đối với nhân Linux, theo Giấy phép Công cộng GNU phiên bản 2. Liên minh Thiết bị cầm tay mở đã thực hiện các thay đổi trên nhân Linux, với mã nguồn lúc nào cũng công khai. Phần còn lại của Android được Google phát triển một

mình, và mã nguồn chỉ được công bố khi phát hành một phiên bản mới. Thông thường Google cộng tác với một nhà sản xuất phần cứng để cung cấp một thiết bị 'chủ lực' (thuộc dòng Google Nexus) với phiên bản mới nhất của Android, sau đó phát hành mã nguồn sau khi thiết bị này được bán ra.

Vào đầu năm 2011, Google quyết định tạm ngưng phát hành mã nguồn Android phiên bản 3.0 Honeycomb dành riêng cho máy tính bảng. Lý do, theo Andy Rubin trong một bài blog Android chính thức, là vì Honeycomb đã được làm gấp gáp để phục vụ cho Motorola Xoom, và họ không muốn các bên thứ ba tạo ra một "trải nghiệm người dùng cực kỳ tồi tệ" bằng cách cố gắng đưa vào điện thoại thông minh một phiên bản dành riêng cho máy tính bảng. Mã nguồn một lần nữa được xuất bản công khai vào tháng 11 năm 2011 với sự ra mắt của Android 4.0.

Mặc dù phần mềm là mã nguồn mở, các nhà sản xuất thiết bị không thể sử dụng thương hiệu Android của Google trừ khi Google chứng nhận rằng thiết bị của họ phù hợp với Tài liệu Định nghĩa Tương thích (Compatibility Definition Document - CDD). Các thiết bị cũng phải thỏa mãn định nghĩa này thì mới được cấp phép để cài các ứng dụng mã nguồn đóng của Google, gồm cả Google Play. Vì Android không hoàn toàn được phát hành theo giấy phép tương thích GPL, ví dụ như mã nguồn của Google là theo giấy phép Apache license, và cũng vì Google Play cho phép các phần mềm có bản quyền, Richard Stallman và Quỹ phần mềm tự do luôn chỉ trích Android và khuyên người dùng sử dụng hệ điều hành khác như Replicant.

1.11. Đón nhận

Android được đón nhận bằng một thái độ thờ ơ khi ra mắt vào năm 2007. Mặc dù những nhà phân tích rất ấn tượng với việc các công ty công nghệ có tiếng tầm hợp tác cùng Google để tạo ra Liên minh thiết bị di động mở, người ta vẫn không rõ liệu các nhà sản xuất có sẵn sàng thay thế hệ điều hành mà họ đang dùng bằng Android hay không. Ý tưởng về một nền tảng phát triển mã nguồn mở dựa trên Linux đã thu hút sự quan tâm, nhưng cũng đẩy lên những lo ngại rằng Android sẽ phải đối mặt với sự cạnh tranh mạnh mẽ từ những tay chơi có hạng trong thị trường điện thoại thông minh, như Nokia và Microsoft, và các hệ điều hành di động đối thủ cũng sử dụng Linux đang trong quá trình phát triển. Những công ty hàng đầu không giấu sự hoài nghi: Nokia được trích nói rằng

"chúng tôi không xem đó là một sự đe dọa," và một thành viên của nhóm Windows Mobile của Microsoft nói rằng "tôi không hiểu rồi họ sẽ có tác động ra sao".

Kể từ đó Android đã phát triển để trở thành hệ điều hành dành cho điện thoại thông minh phổ biến nhất trên thế giới và là "một trong những trải nghiệm di động nhanh nhất hiện nay." Các nhà bình luận thì nhấn mạnh vào bản chất mã nguồn mở của hệ điều hành chính là một trong những yếu tố quyết định sức mạnh, cho phép các công ty như (Kindle Fire), Barnes & Noble (Nook), Ouya, Baidu, và những hãng khác đổi hướng phần mềm và phát hành những phần cứng chạy trên phiên bản Android đã thay đổi của riêng họ. Kết quả, nó được trang web công nghệ Ars Technica mô tả là "đương nhiên là hệ điều hành mặc định khi phát hành phần cứng mới" cho những công ty không có nền tảng di động riêng của họ. Chính sự mở và uyển chuyển này cũng hiện diện ở cấp độ người dùng cuối: Android cho phép người dùng điện thoại điều chỉnh thoải mái thiết bị của họ và ứng dụng thì có sẵn trên các cửa hàng ứng dụng và trang web không phải của Google. Những đặc điểm này được xem là đóng góp vào những thế mạnh chính của điện thoại Android so với các điện thoại khác.

Android cũng bị phê phán vì thiếu sự hỗ trợ hậu mãi từ nhà sản xuất và nhà mạng, nếu so sánh với iOS của Apple. Với những thiết bị không mang nhãn hiệu Nexus, nhà mạng luôn kiểm tra các tiêu chuẩn của họ rồi thực hiện thay đổi cho riêng từng thiết bị (bắt nguồn từ sự điều chỉnh của nhà sản xuất và sự đa dạng của thiết bị Android) được xem là tác nhân chính trì hoãn việc cập nhật. Những nhà bình luận cũng nói rằng ngành công nghiệp thiết bị di động vì lý do lợi nhuận đã cố tình không cập nhật thiết bị của họ, vì thiếu cập nhật trên thiết bị hiện tại sẽ thúc đẩy việc mua thiết bị mới.

Chương 2: KỸ THUẬT LẬP TRÌNH CƠ SỞ DỮ LIỆU VỚI 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 Java hiện nay. Do đó Android Studio sẽ là môi trường phát triển ứng dụng tốt nhất cho Android.

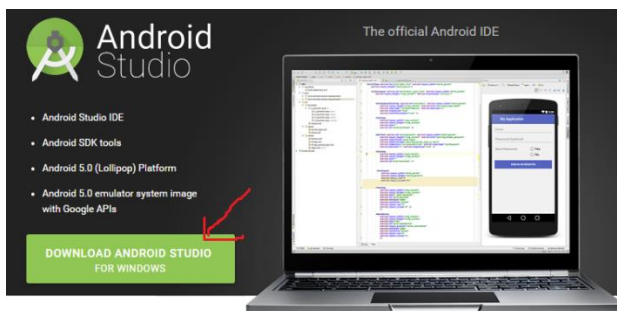
2.1.1. Cài đặt android studio

a. Yêu cầu phần cứng máy tính

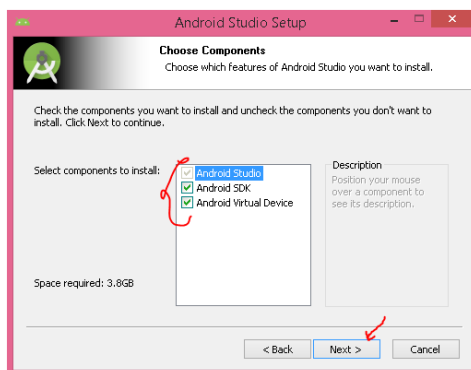
- Microsoft® Windows® 8/7 (32 or 64-bit)
- 4 GB RAM. (Tốt nhất là 8GB)
- 400 MB hard disk space + ít nhất 1GB cho Android SDK, emulator system images và caches
- Độ phân giải tối thiểu 1280 x 800
- Java Development Kit ()

b. Phần mềm android studio

- Vào đường dẫn: "<http://developer.android.com/sdk/index.html>"
- Để download bản mới nhất và tiến hành cài đặt click như hình:



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



- Tiếp tục chọn next và agree cho đến khi hoàn tất.

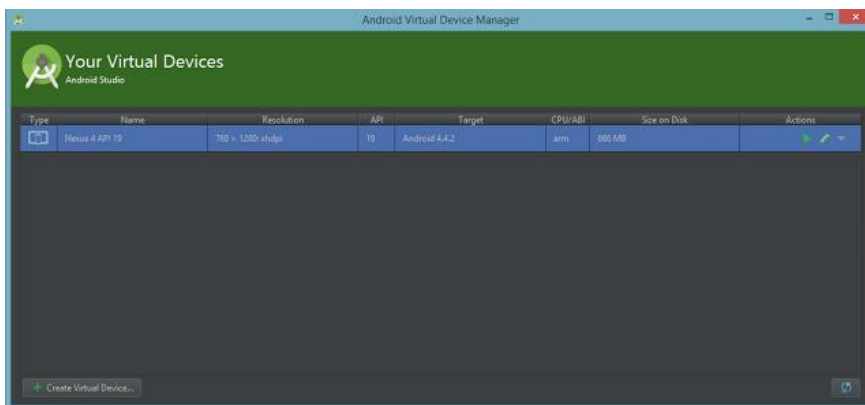
- Đây là màn hình khởi động.



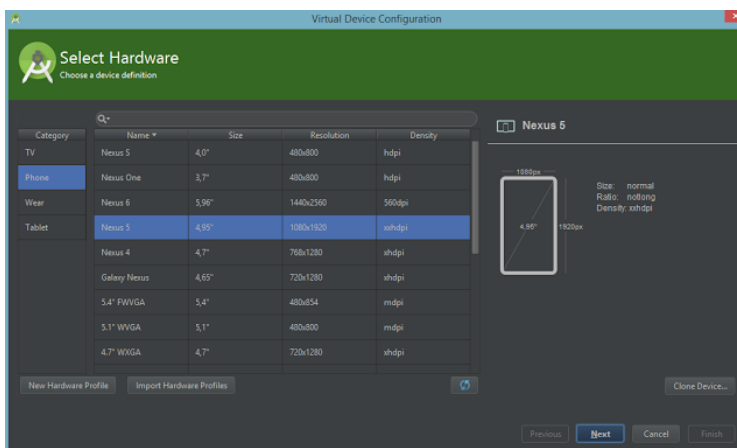
- Các bước cài đặt thiết bị ảo trong android studio

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. Trong Android Studio có cung cấp cho chúng ta một máy ảo Android mặc định là Android Virtual Device viết tắt là AVD.

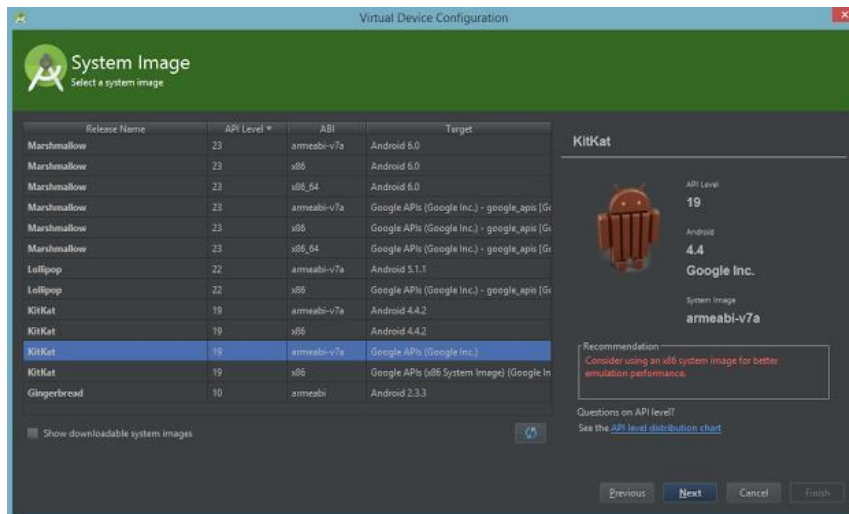
Để cài đặt máy ảo mở Android Studio lên và click vào nút **AVD Manager**. Cửa sổ AVD Manager sẽ xuất hiện:



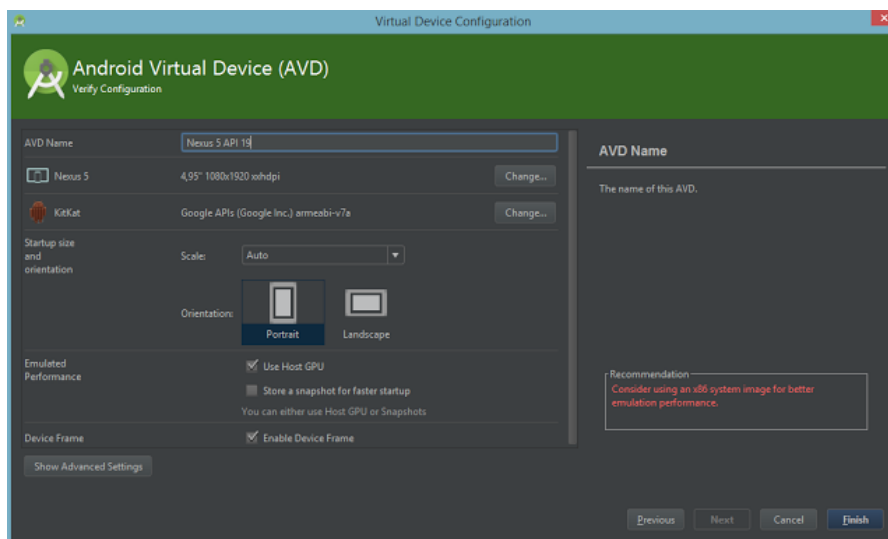
Ở đây ta đang có sẵn một cái máy ảo, ta sẽ tạo thêm một cái máy ảo nữa bằng cách ấn vào nút **Create Virtual Device** ở góc dưới cùng bên trái. Sau đó sẽ hiển thị ra một cửa sổ nữa:



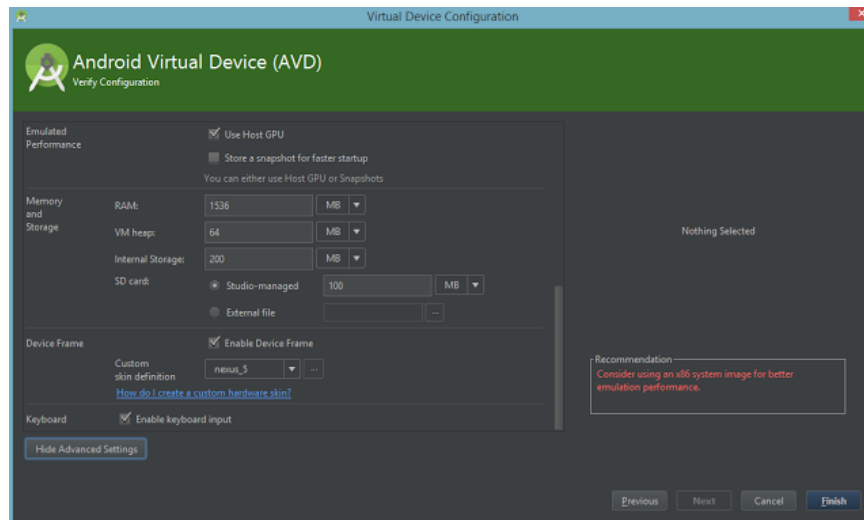
Ở đây có sẵn các mẫu điện thoại, ta chọn mẫu mà mình muốn, ví dụ chọn Nexus 5. Chọn xong bấm "Next" để tiếp tục:



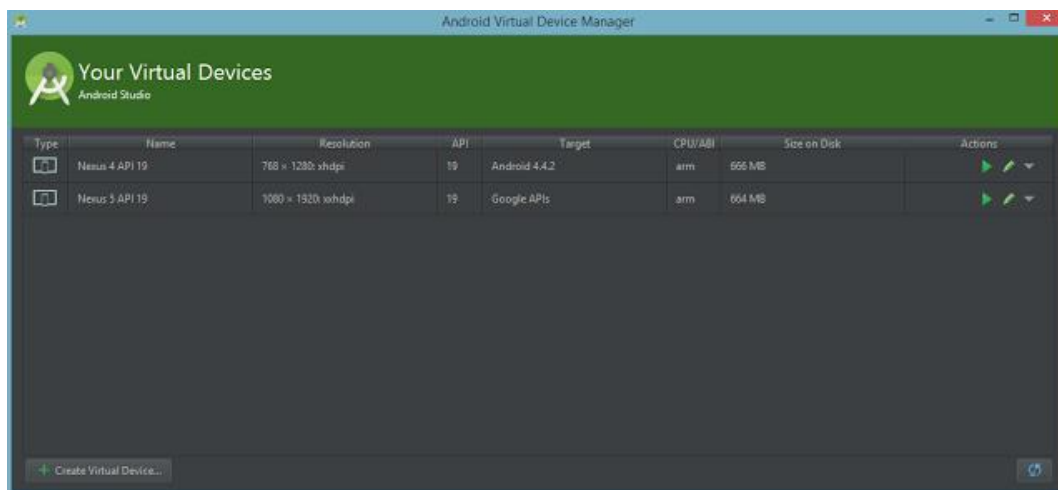
Ở đây các ta chọn phiên bản hệ điều hành muốn cài sau đó ấn Next để tiếp tục:



Ở cửa sổ tiếp theo này ta có thể tùy chỉnh cấu hình máy ảo, ví dụ như độ phân giải, CPU, máy nằm ngang hay dọc,... Nếu muốn tùy chỉnh nhiều hơn nữa ta bấm vào nút **Show Advanced Setting** ở phía dưới cùng bên trái:

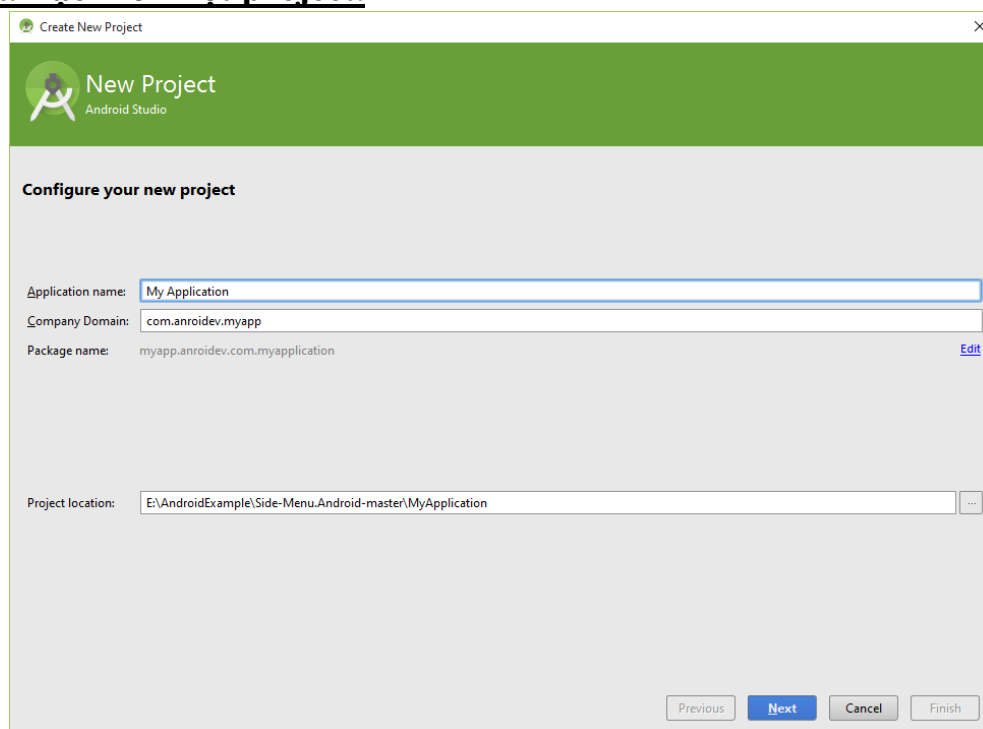


Ta kéo xuống phía dưới sẽ thấy thêm phần tùy chỉnh RAM, bộ nhớ trong, thẻ nhớ,... Sau khi tùy chỉnh theo ý muốn ấn "Finish" để tiến hành tạo máy ảo và chờ một lúc để Android Studio lưu thông tin máy ảo. Sau khi lưu xong ta sẽ thấy trong cửa sổ **AVD Manager** có thêm một cái máy ảo nữa:



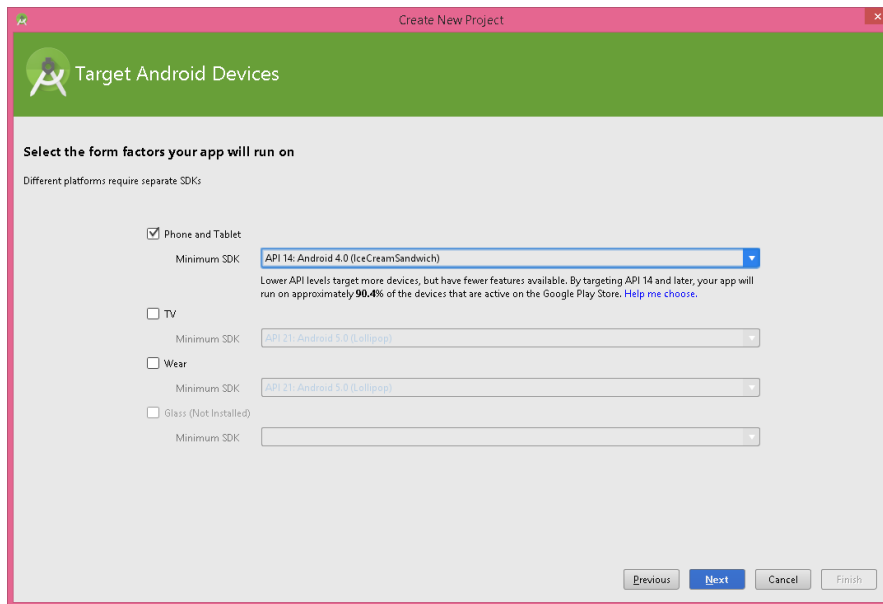
2.1.2. Cấu trúc dự án android studio

a. Tạo mới một project.



- **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.

b. Cài đặt một project.

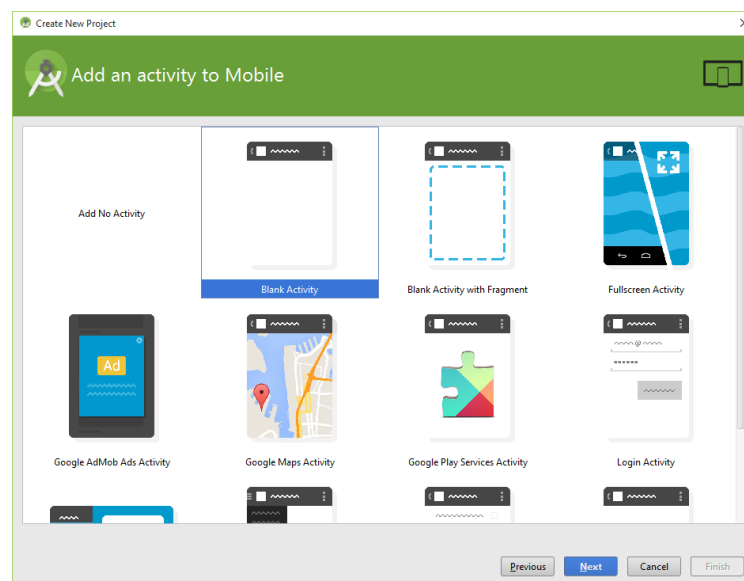


Ở hộp thoại trên cho phép ta lựa chọn là ứng dụng sẽ được viết cho những thiết bị nào (Phone and Tablet, TV, Wear).

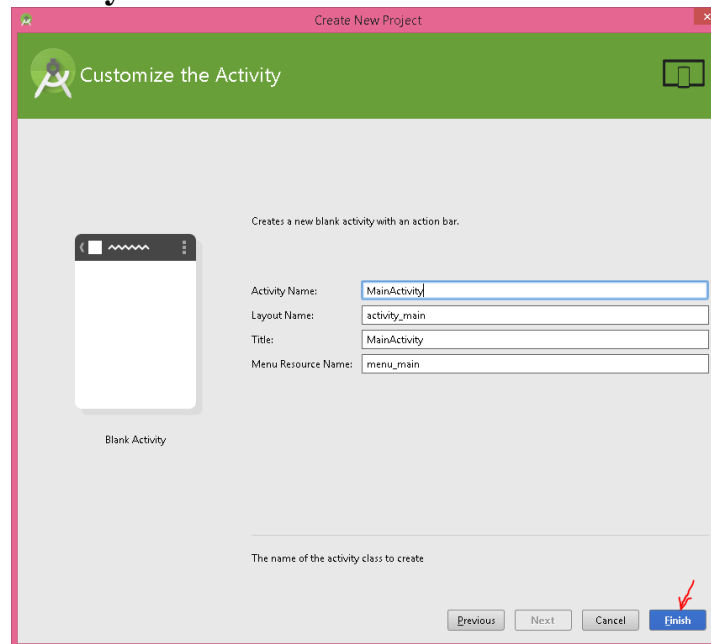
Ở mục Minimum SDK, quy định phiên bản android tối thiểu để chạy ứng dụng.

Hiện nay bản **API14 Android 4.0 (IceCreamSandwich)** vẫn đứng đầu về số lượng thiết bị sử dụng chiếm tới hơn 90%) nên thường lựa chọn.

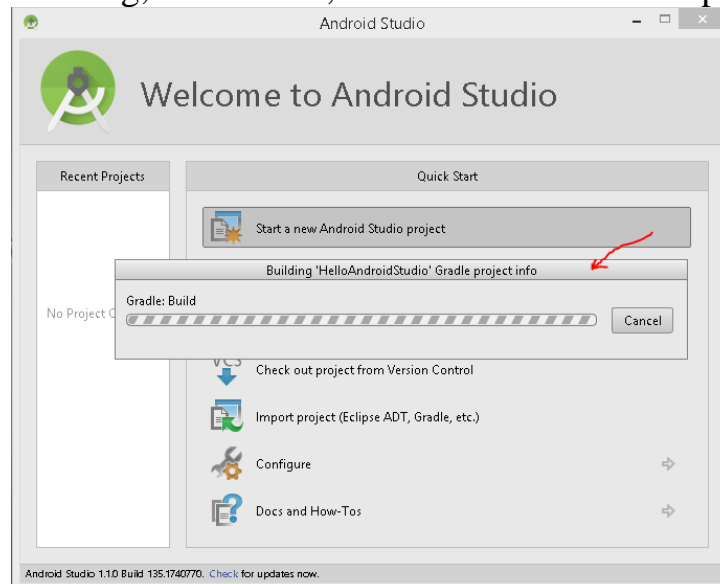
Màn hình này hiển thị cho phép chọn loại Activity mặc định.



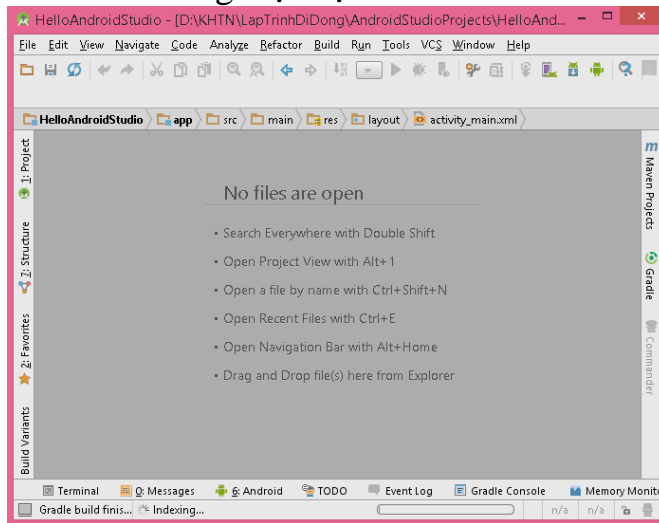
Chọn **Blank Activity** rồi bấm Next:



- **Activity Name:** Tên class Activity (java) để ta viết mã lệnh
 - **Layout Name:** Tên file XML làm giao diện cho Activity Name.
 - **Title:** Tiêu đề hiển thị khi kích hoạt Activity trên thiết bị.
 - **Menu Resource Name:** Tên file xml để tạo menu cho phần mềm.
- Sau khi cấu hình xong, bấm Finish, Màn hình Build Gradle project hiển thị:



Khi build xong mặc định có màn hình dưới đây:

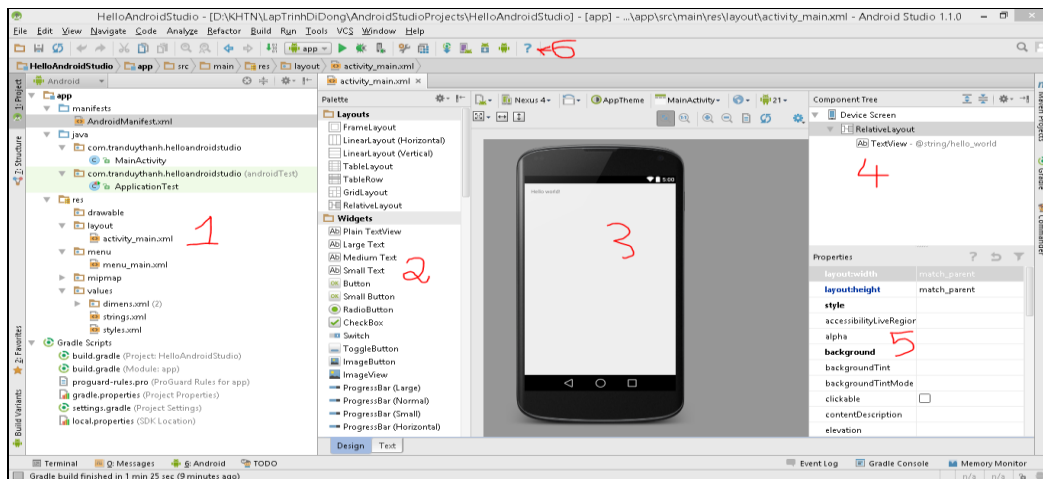


c. Màn hình làm việc của dự án android studio

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

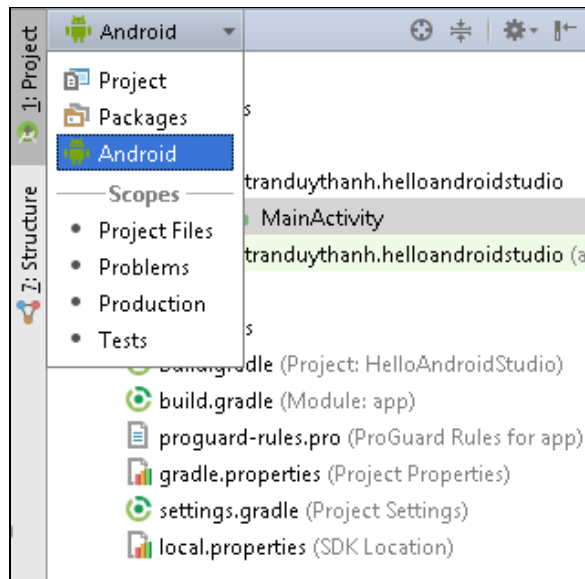
Mở Project mặc định **activity_main.xml** sẽ được chọn ta có màn hình như sau:



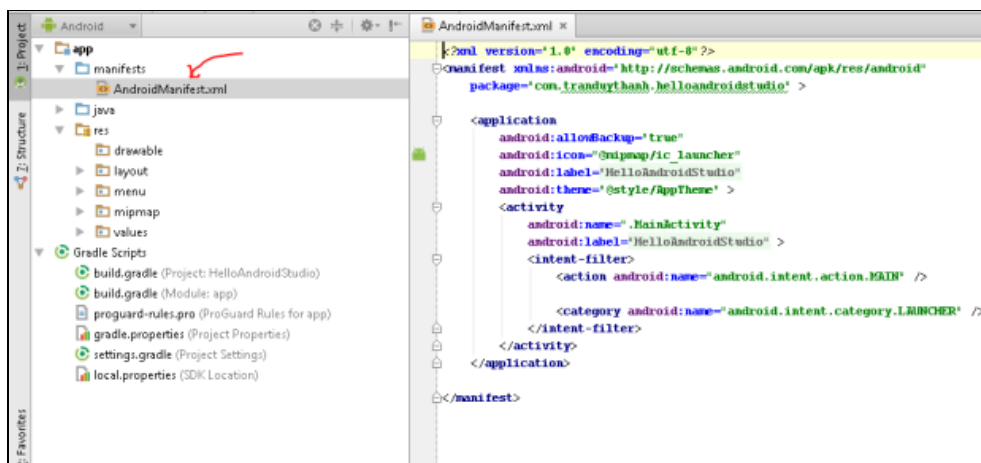
Ở trên tạm thời chia làm 6 vùng làm việc mà lập trình viên thường tương tác.

- 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**).



Màn hình ở chế độ Android:



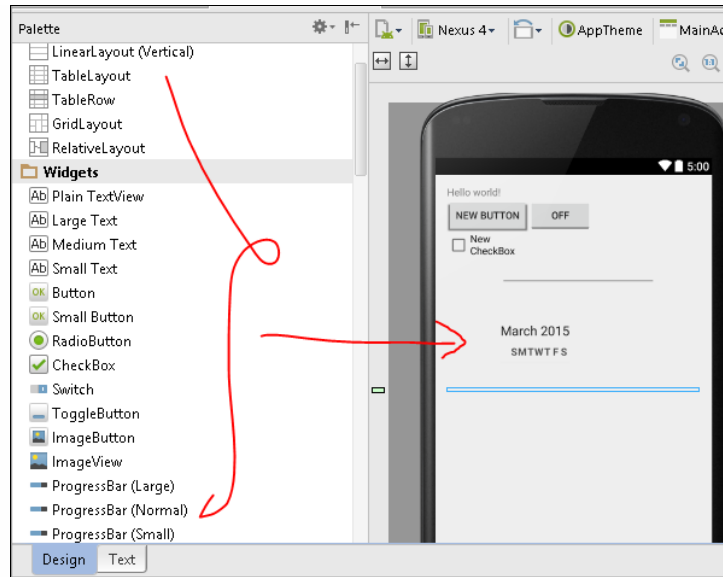
Ta có thể thấy **AndroidManifest.xml** nằm ở đây. File này vô cùng quan trọng trong việc cấu hình ứng dụng.

Các thư mục:

- **drawable**: chứa các file hình ảnh và xml trong ứng dụng.
- **layout**: chứa các giao diện màn hình được thiết kế dưới dạng xml.
- **values**: chứa các file lưu giá trị màu sắc, kích thước, chuỗi,....

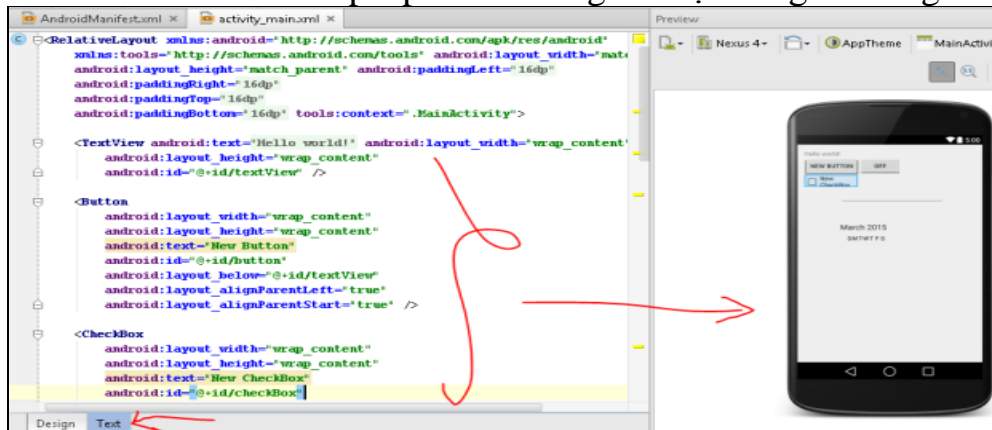
- Vùng 2.

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 3 (Giao Diện Thiết Bị) để thiết kế.



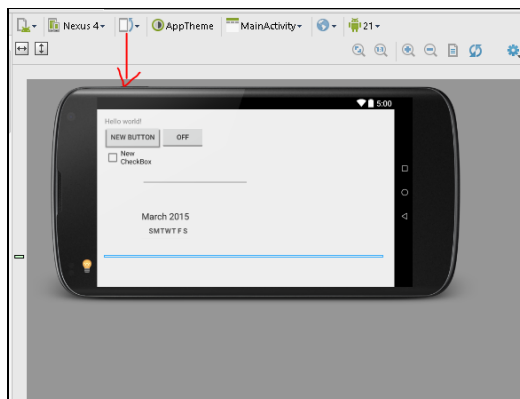
Ở vùng số 2 này nó có 2 tab: **Design** và **Text** ở góc trái dưới cùng.

- Tab **Design** là tab mà ta đang nhìn và thao tác với nó (cho phép thiết kế giao diện bằng cách kéo thả).
- Tab **Text** là tab cho phép ta thiết kế giao diện bằng viết Tag XML:



- Vùng 3.

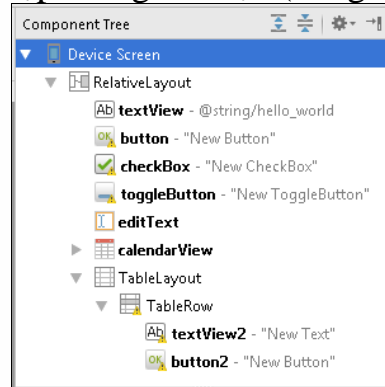
Là vùng giao diện thiết bị, cho phép các Control kéo thả vào đây và đồng thời cho ta hiệu chỉnh control.



Vùng 3 ta có thể chọn cách hiển thị theo nằm ngang nằm đứng, phóng to thu nhỏ, căn chỉnh control, lựa chọn loại thiết bị hiển thị....

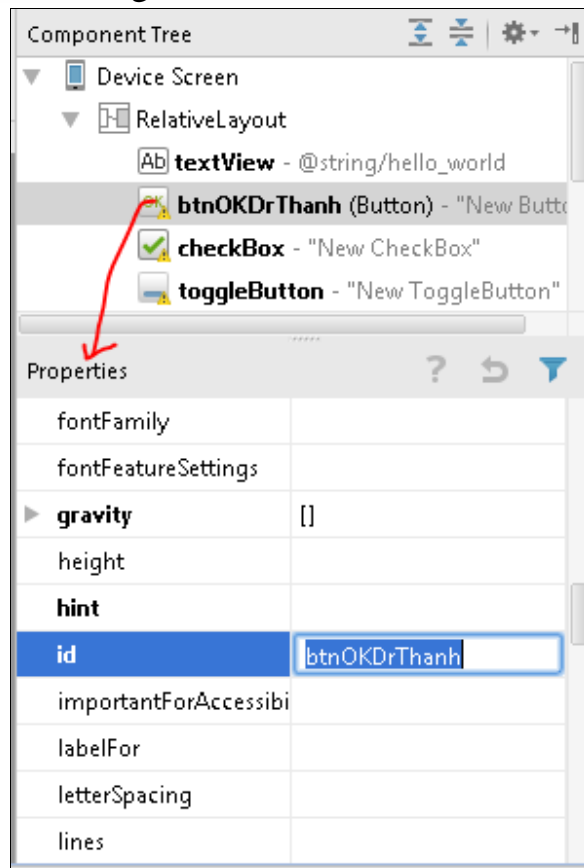
- Vùng 4.

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 lặp trên giao diện (vùng 3).



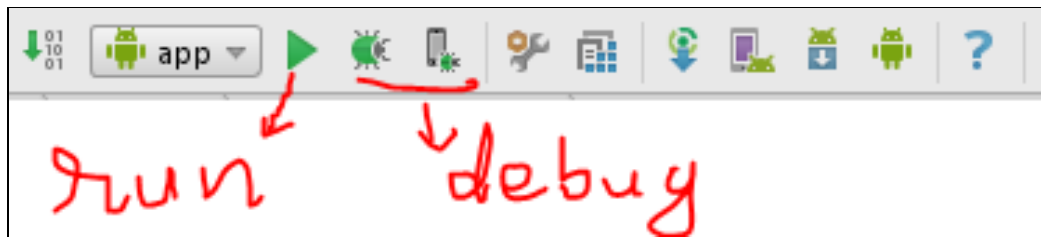
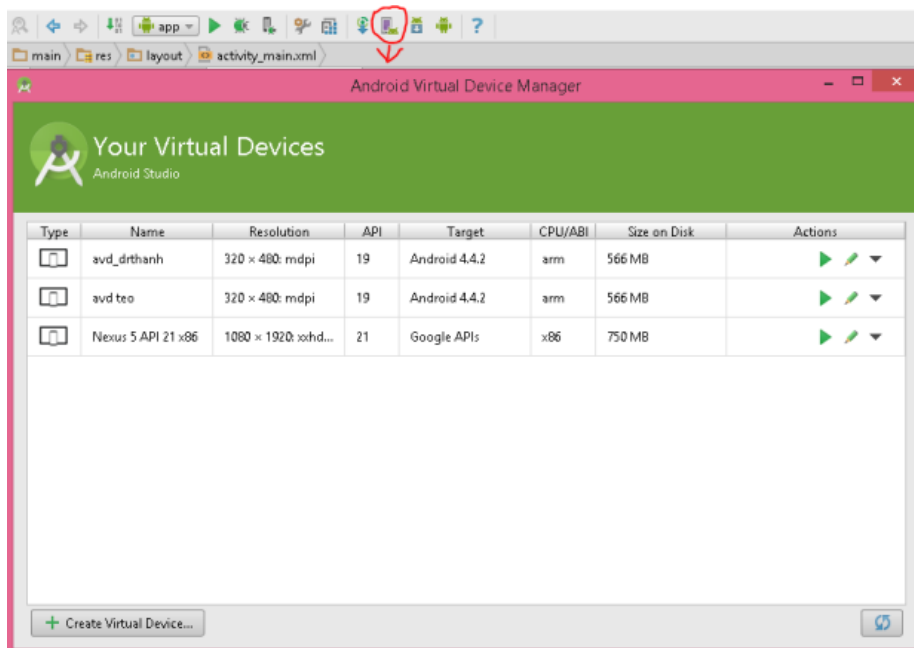
- Vùng 5.

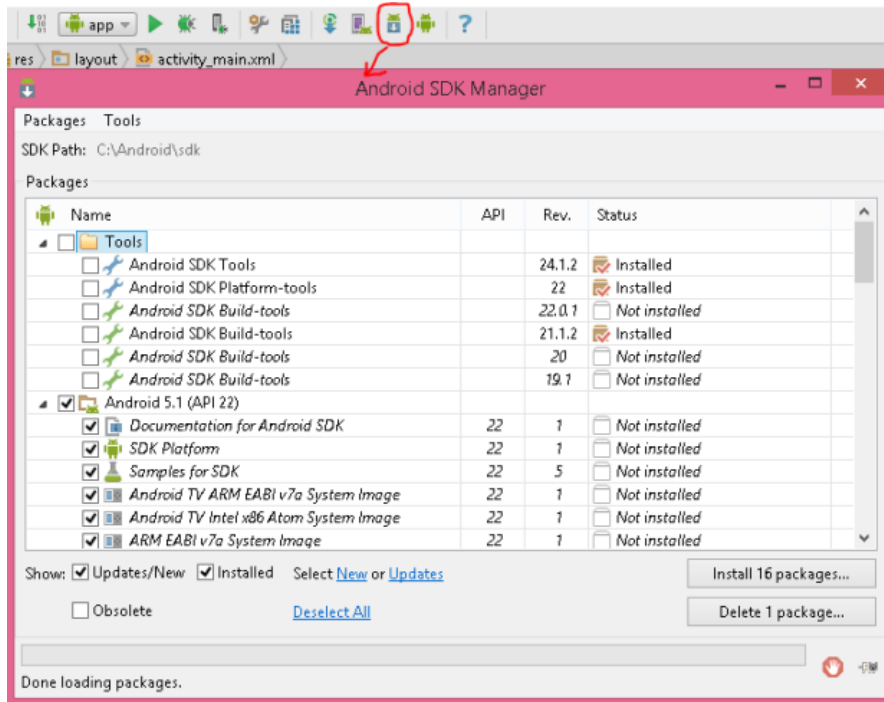
Vùng này rất quan trọng, đây là vùng cho phép thiết lập trạng thái hay thuộc tính cho các Control trên giao diện.



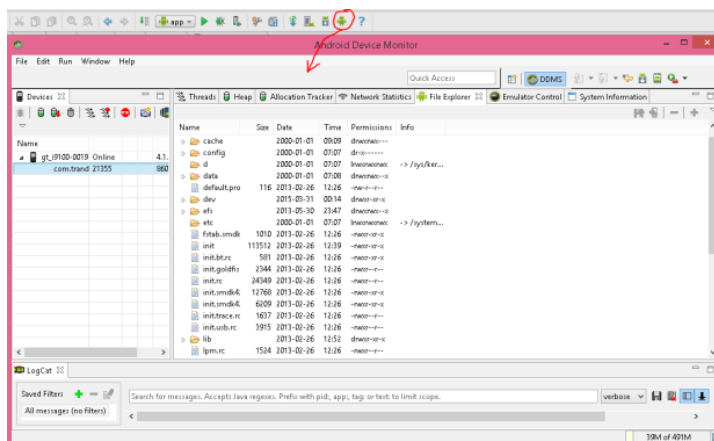
- Vùng 6

Là vùng các chức năng quan trọng thường dùng trong Android Studio.

Chạy ứng dụng và Debug ứng dụng:**Quản lý máy ảo (AVD Manager)****Quản lý Android SDK Manager (thường dùng để cập nhật).**



Quản lý Android Device Manager



2.1.3. Tạo giao diện chương trình trong android studio

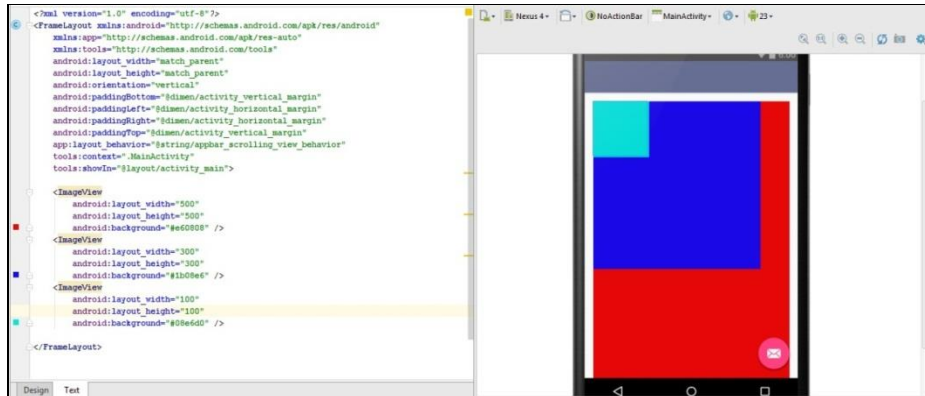
a. Giới thiệu android Layout

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.



- LinearLayout.

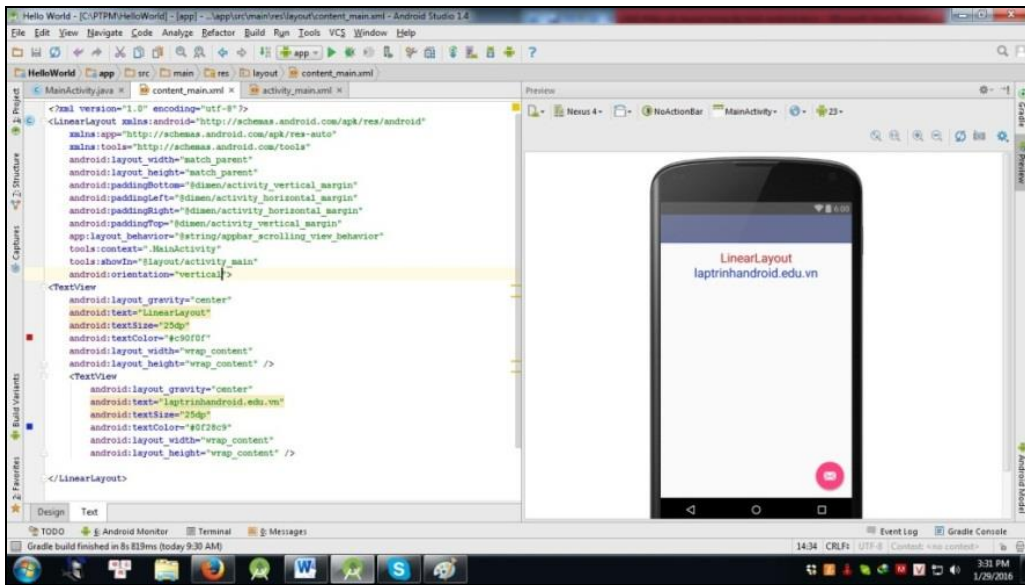
Layout này cho phép ta vẽ giao diện theo 2 hướng, từ trái qua phải hoặc từ trên xuống dưới. Để xét chiều cho các control trong layout ta sử dụng thuộc tính **orientation**.

- Android:orientation="horizontal" : Xếp các control từ trái sang phải (theo cột).
- Android:orientation="vertical" : Xếp các control từ trên xuống dưới (theo hàng).

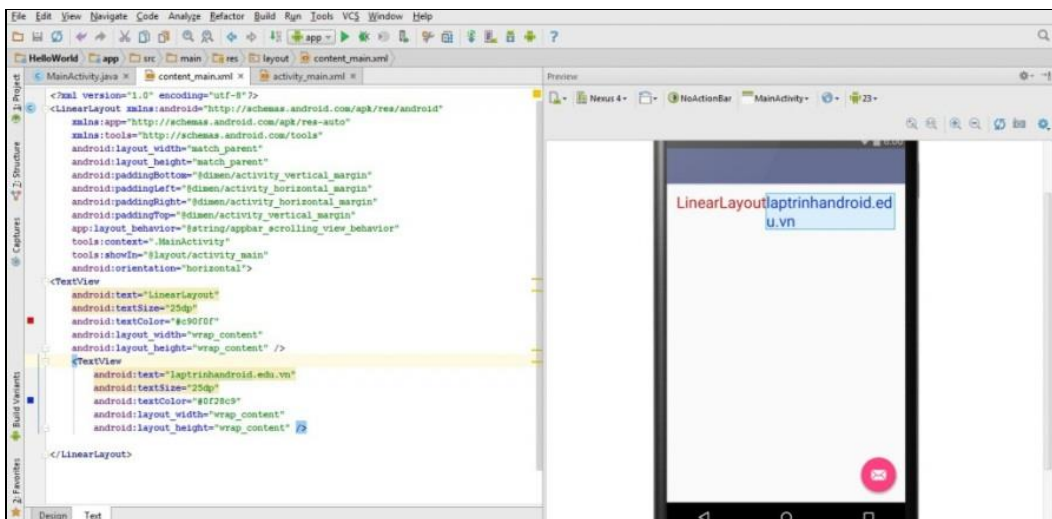
Với những giao diện có độ phức tạp vừa phải thì dùng LinearLayout là rất hiệu quả, rất thuận tiện trong thiết kế và đi bảo trì ứng dụng sau này.

Sau đây là đoạn XML demo cách sử dụng layout này:

Theo hàng



Theo cột

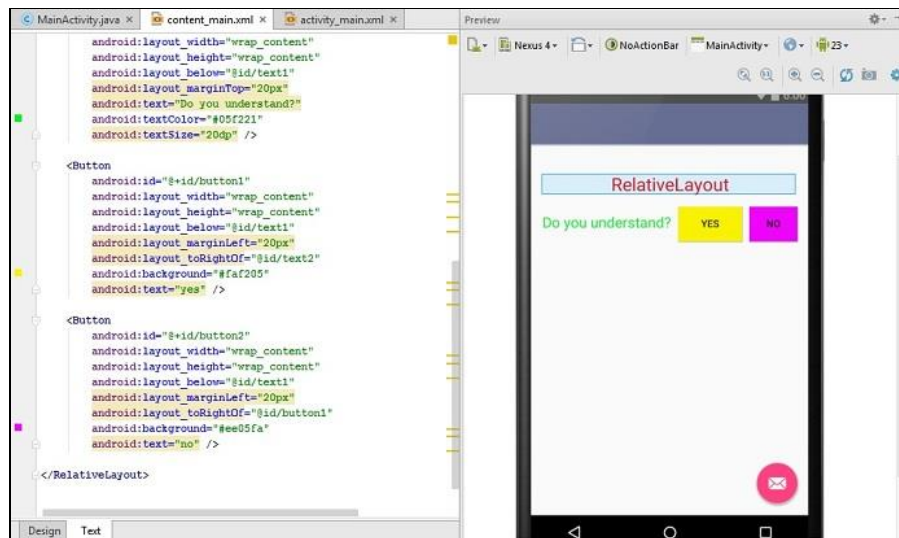


- RelativeLayout.

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 dễ 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:



b. 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 <TextView> trong xml.
- **EditText:** là view dùng để lấy giá trị từ người dùng nhập vào. EditText được định nghĩa bởi thẻ <EditText> 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.

c. 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. 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). Trong các giao tiếp event listener có những phương thức sau đây:

- **onClick():** Thuộc `View.OnClickListener`. Nó được gọi khi người dùng hoặc chạm vào item (khi ở chế độ cảm ứng), hoặc lựa chọn vào item với các phím điều hướng và nhấn nút "enter" phù hợp.
- **onLongClick():** Thuộc `View.OnLongClickListener`. Nó được gọi khi người dùng chạm và giữ item (khi ở chế độ cảm ứng), hoặc lựa chọn vào item với các phím điều hướng sau đó nhấn và giữ phím "enter".
- **onFocusChange():** Thuộc `View.OnFocusChangeListener`. Nó được gọi khi người dùng điều hướng ra khỏi item, bằng cách sử dụng phím điều hướng.
- **onKey():** Thuộc `View.OnKeyListener`. Nó được gọi khi người dùng lựa chọn và nhấn lên item.
- **onTouch():** Thuộc `View.OnTouchListener`. Nó được gọi khi người dùng thực hiện một hành động xác định đủ điều kiện như là một sự kiện cảm ứng, bao gồm việc nhấn, thoát ra, hoặc bất kỳ cử chỉ chuyển động vẽ trên màn hình (bên trong phạm vi của item).
- **onCreateContextMenu():**
Thuộc `View.OnCreateContextMenuListener`. Nó được gọi khi một menu ngữ cảnh (Context Menu) đang được xây dựng (là kết quả của một "long click"). Xem thêm thông tin về context menus trong hướng dẫn phát triển Menus.

Ví dụ dưới đây cho thấy làm thế nào để đăng ký một bộ bắt sự kiện khi nhấp chuột vào một Button.

```
// Create an anonymous implementation of OnClickListener
private OnClickListener mCorkyListener = new OnClickListener() {
    public void onClick(View v) {
        // do something when the button is clicked
    }
};

protected void onCreate(Bundle savedInstanceState) {
    ...
    // Capture our button from layout
    Button button = (Button)findViewById(R.id.corky);
    // Register the onClick listener with the implementation above
    button.setOnClickListener(mCorkyListener);
    ...
}
```

Ta cũng có thể tìm thấy cách thuận tiện hơn để bổ sung `OnClickListener` như một phần `Activity`. Ví dụ:

```
public class ExampleActivity extends Activity implements OnClickListener {
    protected void onCreate(Bundle savedInstanceState) {
        ...
        Button button = (Button)findViewById(R.id.corky);
        button.setOnClickListener(this);
    }

    // Implement the OnClickListener callback
    public void onClick(View v) {
        // do something when the button is clicked
    }
    ...
}
```

Chú ý rằng lời gọi `onClick()` trong ví dụ trên không trả về giá trị, nhưng các phương thức của bộ nghe sự kiện khác phải trả lại một biến kiểu boolean. Lý do phụ thuộc vào sự kiện này. Đây là một vài lý do:

- **onLongClick()** - Trả về một giá trị kiểu boolean để cho biết ta đã dùng sự kiện này và nó không cần thực hiện "long click") thêm nữa. Trả về giá trị *TRUE* để chỉ ra rằng ta đã xử lý sự kiện này và nó nên dừng lại ở đây; trả về *FALSE* nếu ta không xử lý nó và / hoặc sự kiện nên chuyển tới bất kỳ bộ nghe sự kiện **on-click** nào khác.
 - **onKey()** - Trả về một giá trị kiểu boolean để cho biết ta đã dùng sự kiện này và nó không cần được thực hiện thêm. Trả về giá trị *TRUE* để chỉ ra rằng ta đã xử lý sự kiện này và nó nên dừng lại ở đây; trả về *FALSE* nếu ta không xử lý nó và / hoặc sự kiện nên chuyển tới bất kỳ bộ nghe sự kiện **on-key** nào khác.
 - **onTouch()** - Trả về một giá trị kiểu boolean để cho biết: liệu bộ nghe của ta đã dùng sự kiện này hay chưa. Điều quan trọng là sự kiện này có thể có nhiều hành động nối tiếp nhau. Vì vậy, nếu trả về *FALSE*, ta biết rằng ta đã không sử dụng và cũng không quan tâm đến hành động tiếp theo từ sự kiện này. Như vậy, ta không được gọi tới bất kỳ thao tác nào khác bên trong sự kiện này.
- **Event Handlers** – Là phương thức xử lý khi phát sinh sự kiện. (Xử lý sự kiện)

Nếu ta đang xây dựng một thành phần tùy chỉnh từ `View`, ta sẽ phải định nghĩa một số phương thức sử dụng như của xử lý sự kiện mặc định. Trong tài

liệu về Custom Components, ta sẽ tìm hiểu một số callbacks thường được sử dụng để xử lý sự kiện, bao gồm:

- `onKeyDown(int, KeyEvent)` – Được gọi khi một sự kiện nhấn phím mới xảy ra.
- `onKeyUp(int, KeyEvent)` – Được gọi khi một sự kiện thả phím xảy ra.
- `onTrackballEvent(MotionEvent)` – Được gọi khi một sự kiện chuyển động trackball xảy ra.
- `onTouchEvent(MotionEvent)` – Được gọi khi một sự kiện chuyển động màn hình cảm ứng xảy ra.
- `onFocusChanged(boolean, int, Rect)` – Được gọi khi view được chọn (focus) hoặc bỏ chọn.

Có một số phương thức khác mà ta nên biết, chúng không phải là một phần của lớp View, nhưng có thể trực tiếp tác động đến cách bạn có thể xử lý các sự kiện. Vì vậy, khi quản lý sự kiện phức tạp hơn bên trong một layout, ta nên xem xét các phương pháp sau:

- `Activity.dispatchTouchEvent(MotionEvent)` – Điều này cho phép Activity bắt tất cả các sự kiện chạm màn hình trước khi chúng được gửi đến cửa sổ.
- `ViewGroup.onInterceptTouchEvent(MotionEvent)` – Điều này cho phép ViewGroup xem các sự kiện như chúng được gửi đến các View con.
- `ViewParent.requestDisallowInterceptTouchEvent(boolean)` – Gọi điều này trên View cha để xác định rằng nó không nên bắt các sự kiện chạm màn hình với `onInterceptTouchEvent(MotionEvent)`

2.2. Kỹ thuật lập trình cơ sở dữ liệu internet trên thiết bị di động

2.2.1. Dịch vụ web và cơ sở dữ liệu trên internet

Trong thời đại của xã hội công nghệ thông tin và nền kinh tế tri thức, mọi hoạt động của các cá nhân, tổ chức, doanh nghiệp đều mong muốn đạt hiệu quả cao, giành được thắng lợi trong sự cạnh tranh gay gắt thì đòi hỏi phải có những phương pháp để có thể cung cấp, trao đổi những thông tin, tri thức cần thiết một cách nhanh chính xác, thuận tiện và dễ dàng. Chính vì vậy việc tạo lập xây dựng lưu trữ thông tin, cơ sở dữ liệu chung trên môi trường Internet là một việc vô cùng cần thiết để các cá nhân, tổ chức, doanh nghiệp dễ dàng làm việc, thao tác với cơ sở dữ liệu mọi lúc, mọi nơi, mọi thiết bị khi có Internet thay vì thao tác với cơ sở dữ liệu cục bộ trên máy tính tại một vị trí.

- Ưu điểm của việc tạo lập cơ sở dữ liệu trên Internet:

- Cấu trúc phân tán dữ liệu thích hợp cho bản chất phân tán của nhiều người dùng.
- Dữ liệu được chia sẻ trên mạng nhưng vẫn cho phép quản trị dữ liệu địa phương (dữ liệu đặt tại mỗi trạm).
- Dữ liệu có tính tin cậy cao
- Dữ liệu có tính sẵn sàng cao.
- Hiệu năng của hệ thống được nâng cao hơn.
- Cho phép mở rộng các cá nhân, tổ chức, doanh nghiệp một cách linh hoạt.

- Nhược điểm:

- Việc thiết kế tạo lập cơ sở dữ liệu phức tạp hơn.
- Chi phí cao hơn.
- Đảm bảo an ninh khó khăn hơn.
- Đảm bảo tính nhất quán dữ liệu khó hơn.

a. Hosting là gì?

Hosting là dịch vụ lưu trữ dữ và chia sẻ liệu trực tuyến, là không gian trên máy chủ có cài đặt các dịch vụ Internet như world wide web (www), truyền file (FTP), Mail... , ta có thể chứa nội dung trang web hay dữ liệu trên không gian đó.

b. Các loại hosting

- **Shared hosting:** là một dịch vụ lưu trữ rất nhiều các trang web trên một máy chủ kết nối Internet. Mỗi trang web có phân vùng riêng của mình. Dịch vụ này là một lựa chọn kinh tế cho nhiều người chia sẻ tổng chi phí bảo trì thuê máy chủ.
- **Free web hosting:** là một dịch vụ lưu trữ miễn phí, thường được quảng cáo hỗ trợ. Free Hosting thường sẽ cung cấp một tên miền phụ (yoursite.example.com) hoặc một thư mục (www.example.com/~Yourname) hoặc ta có thể sử dụng tên miền của chính mình và cần một vài điều kiện kèm theo. Ngược lại, dịch vụ thu phí thường sẽ cung cấp một tên miền cấp thứ hai cùng với các máy chủ (www.tenmien.com). Nhiều máy chủ miễn phí không cho phép sử dụng tên miền riêng. Hosting miễn phí bị một số nước chặn không cho sử dụng như Trung Quốc.
- **Reseller hosting:** là một hình thức lưu trữ của máy chủ web mà chủ sở hữu tài khoản có khả năng sử dụng tài khoản của mình để phân bổ lại ổ cứng lưu trữ và băng thông để lưu trữ các trang web thay mặt cho bên thứ ba. Các đại lý mua một phần không gian trên máy chủ sau đó họ bán cho khách hàng thu lợi nhuận.

- **Email hosting:** là một dịch vụ thư điện tử đặc biệt khác với các dịch vụ email miễn phí hỗ trợ email hay webmail miễn phí. Doanh nghiệp thường chạy các dịch vụ lưu trữ thư điện tử riêng (Email hosting) theo tên miền của họ để tăng uy tín và chứng thực các thông điệp mà họ gửi đi. Email hosting cho phép tùy chỉnh cấu hình và số lượng lớn các tài khoản.
- **File hosting:** là dịch vụ lưu trữ tập tin trực tuyến, được thiết kế đặc biệt để lưu trữ các nội dung tĩnh, điển hình là các tập tin lớn mà không phải là các trang web. Thông thường họ cho phép truy cập qua giao thức FTP được tối ưu hóa phục vụ cho nhiều người sử dụng.
- **Windows hosting:** là một dịch vụ lưu trữ, cho rất nhiều các trang web trên một máy chủ chạy hệ điều hành Windows Sever kết nối Internet. Mỗi trang web có phân vùng riêng của mình thường sử dụng các phần mềm chia hosting như Hosting Controller, Plesk...
- **Linux hosting:** là một dịch vụ lưu trữ rất nhiều các trang web trên một máy chủ chạy hệ điều hành Linux kết nối Internet. Mỗi trang web có phân vùng riêng của mình thường sử dụng các phần mềm chia hosting như Cpanel, Direct Admin...
- **VPS Hosting:** một máy chủ riêng ảo (tiếng anh: *Virtual Private Server*), hay VPS là một phương pháp phân vùng một máy chủ vật lý thành máy tính nhiều máy chủ ảo, mỗi máy chủ đã có khả năng của riêng của mình chạy trên máy tính dành riêng. Mỗi máy chủ ảo riêng của nó có thể chạy full-fledged hệ điều hành, và mỗi máy chủ độc lập có thể được khởi động lại.

c. Các thông số cần biết trong hosting

Hệ điều hành (OS) của máy chủ : hiện tại có hai loại OS thông dụng là Linux và Windows.

- Hosting Linux: là Hosting chuyên hỗ trợ ngôn ngữ lập trình PHP, Joomla, các mã nguồn mở...
- Hosting Windows: Hosting Windows chuyên hỗ trợ về ngôn ngữ lập trình ASP, ASP.Net, HTML vì các Ngôn ngữ này, chạy chuyên trên Hosting Windows, do vậy khi load Web sẽ hỗ trợ tốt hơn, Hosting Windows có hỗ trợ ngôn ngữ PHP, nhưng chủ yếu là hỗ trợ chính là ASP ...
- Dung lượng: Bộ nhớ lưu trữ cho phép bạn tải file lên host.
- Băng thông: Bandwidth (băng thông) là thông số chỉ dung lượng thông tin tối đa mà website được lưu chuyển qua lại mỗi tháng.
- PHP: Phiên bản php hỗ trợ.
- Max file: Số lượng file tối đa có thể upload lên host.
- RAM: Bộ nhớ đệm.

- Addon domain: Số lượng domain ta có thể trở tới hosting.
- Subdomain: Số lượng tên miền phụ có thể tạo ra cho mỗi tên miền.
- Park domain: Số lượng tên miền có thể parking.
- Email accounts: Số lượng email đi kèm với hosting.
- FTP accounts: Số lượng FTP account bạn có thể tạo và dùng nó upload dữ liệu lên hosting.

d. Đăng ký mua và sử dụng hosting.

Tùy vào mục đích sử dụng của mỗi cá nhân, tổ chức và doanh nghiệp có thể chọn các hình thức đăng ký mua, thuê hosting khác nhau tại rất nhiều các nhà cung cấp hosting tại Việt Nam như:

- Mắt Bão – ODS
- PA Việt Nam
- DIGISTAR
- Nhân Hòa
- Viettel
- Tenten
- FPT
- VDC
- Vdata
- Vhost

Hiện nay cũng có một số hosting miễn phí phục vụ cho quá trình học tập nghiên cứu như:

- HOSTINGER.VN
- 000WEBHOST.COM
- **ATSPACE.COM**
- **AWARDSPACE.COM**
- HOSTBUDDY.COM

e. Các bước đăng ký sử dụng hosting

VD: Đăng ký tại "000webhost.com".

- Đến địa chỉ: "<https://members.000webhost.com/signup>"
- Điền đầy đủ các thông tin yêu cầu như hình:

Order Free Web Hosting

I want to host my own domain (domain must be registered already)
 I will choose your free subdomain (recommended)

or, I will choose your free subdomain (recommended)

www. .net16.net

Name → tên miền thứ cấp
 → họ và tên

Email → email

Password → mật khẩu

Repeat password → lặp lại mật khẩu

Like our Facebook Page (Thanks!)
 → Nhấp chuột để đồng ý điều khoản sử dụng dịch vụ

I agree to Terms Of Service → Nhấp chuột vào để xác nhận chống spam

Tôi không phải là người máy → reCAPTCHA Bảo mật - Điều khoản

→ Nhấp chuột vào để tạo tài khoản

- Sau khi điền đầy đủ thông tin, nhấp chuột vào “*Create my Account*”, nếu thành công ta sẽ nhận được trang thông báo như bên dưới:

List of your accounts

Account thietkewebsitemienphi.comlu.com with username a7984047 has been created! (account will be setup on server in about 1 minute).

» Domain	» Status	» Action
thietkewebsitemienph...	Waiting for email confirmation..	--

[Create new account](#)

- Đăng nhập email kiểm tra hộp thư đến để xác nhận đăng ký thành công.

Confirm your registration Inbox x

000webhost.com <no-reply@000webhost.com> 11:13 PM (6 minutes ago) ☆
 to me ▾

Dear [REDACTED] → nhấp chuột vào liên kết để xác nhận

Welcome to the best free hosting provider [000webhost.com](https://members.000webhost.com) !

To confirm your registration please click this link: <https://members.000webhost.com/email-confirm/15015068/7UsRCXhPW283f6ELrGy61qKcattl8eewSKcx4QoY>
 If clicking the link does not work, please copy and paste the URL into your browser instead.

Sincerely,
 000webhost.com team

Sau khi nhấp chuột vào liên kết, ta sẽ được đưa đến trang tài khoản trên 000webhost.com. Ta có thể thay đổi mật khẩu nếu muốn.



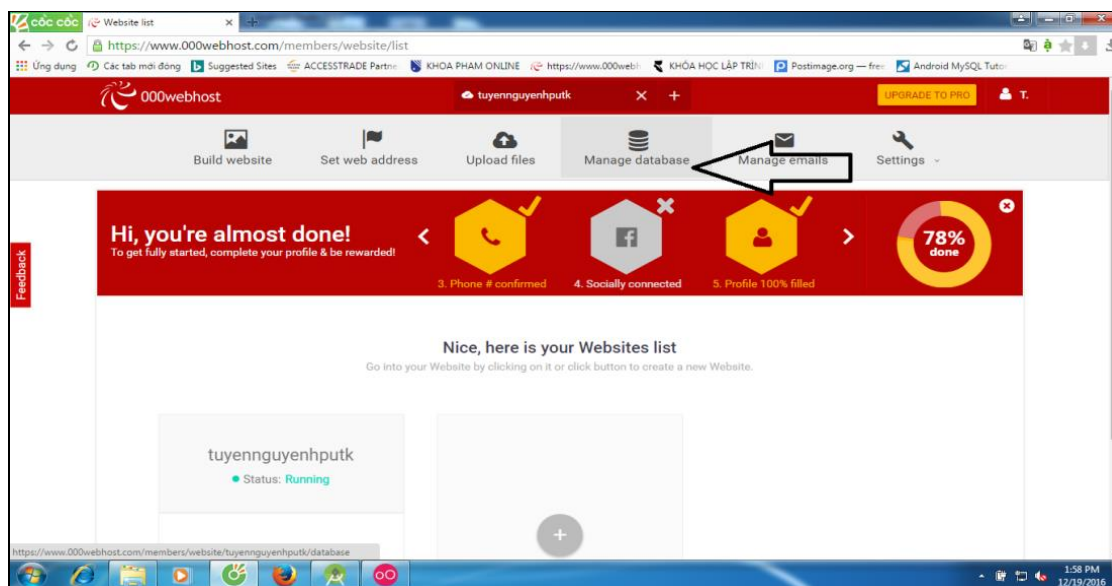
f. Tạo cơ sở dữ liệu trên hosting

- Tạo mới cơ sở dữ liệu

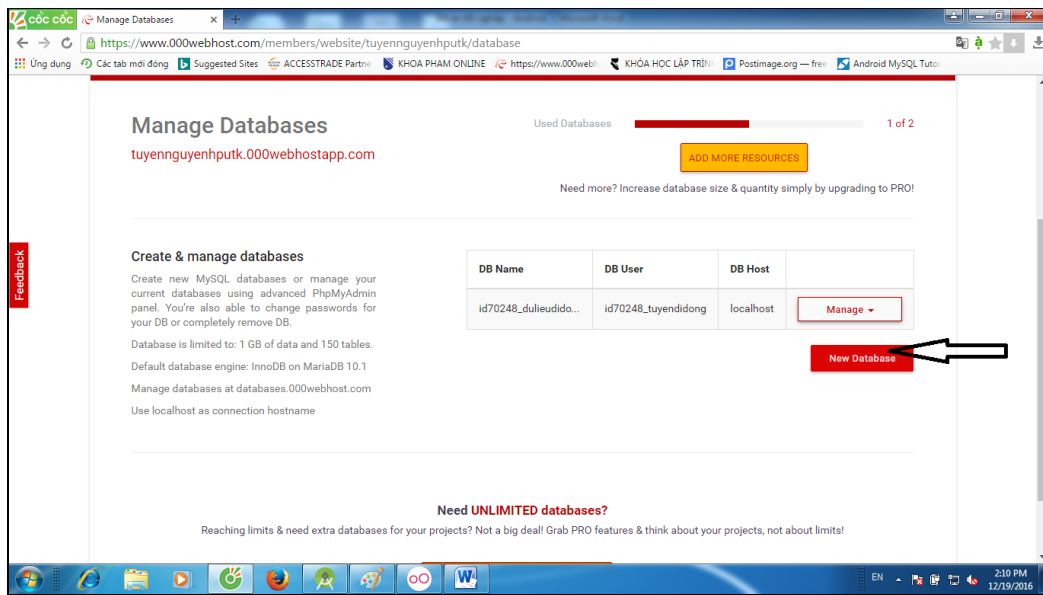
VD: Tạo cơ sở dữ liệu tại "000webhost.com".

Đăng nhập vào 000webhost.com bằng tài khoản đã đăng ký:

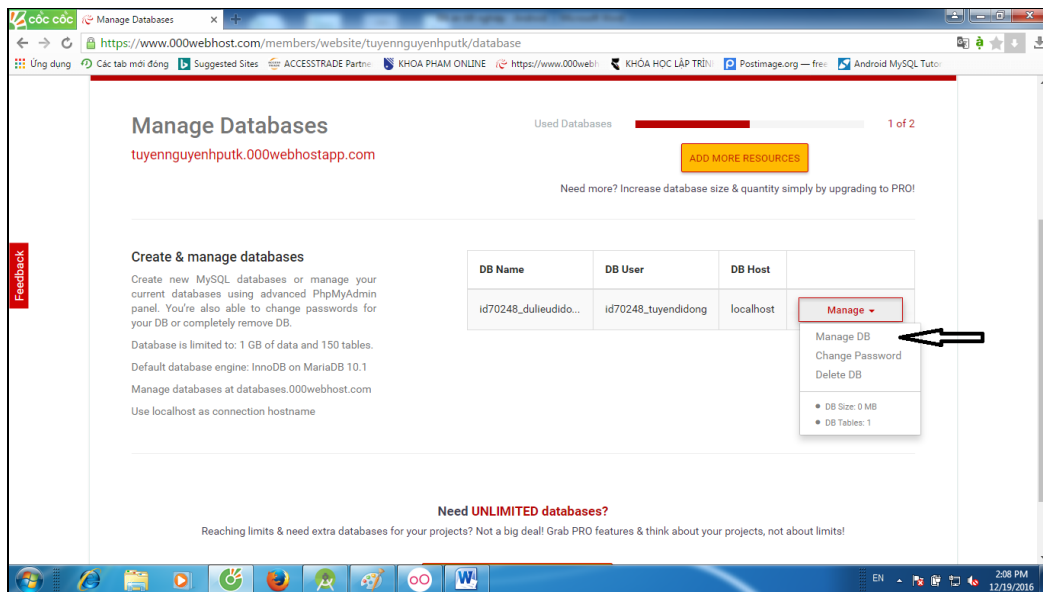
Click vào **Manage database**



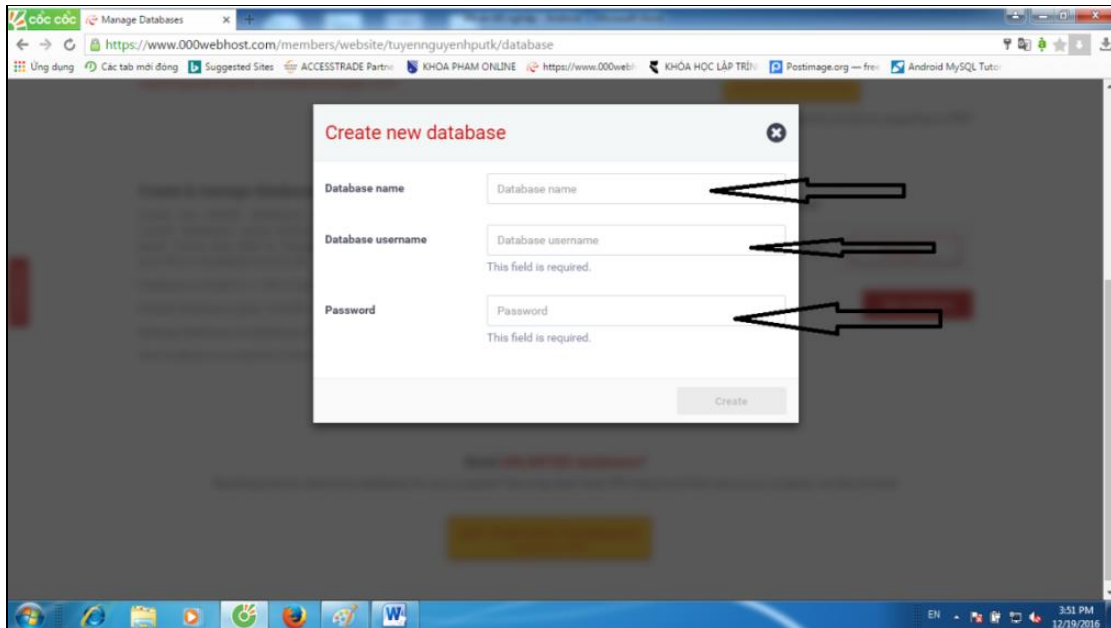
Tiếp theo chọn **New database** để tạo mới



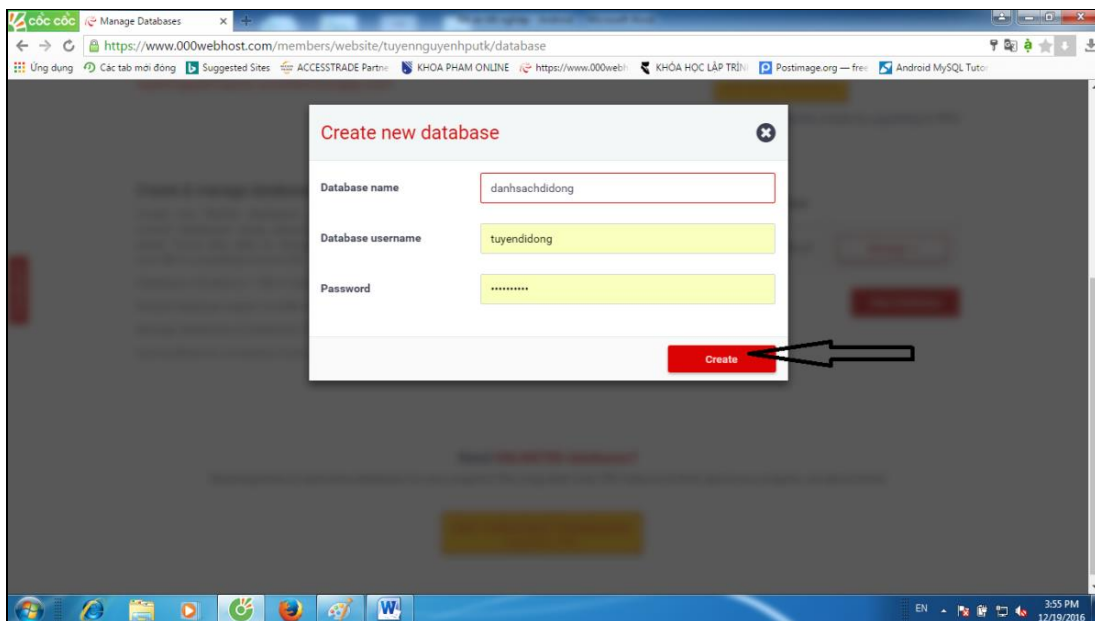
hoặc chọn **Manage DB** để thao tác với database đã có.



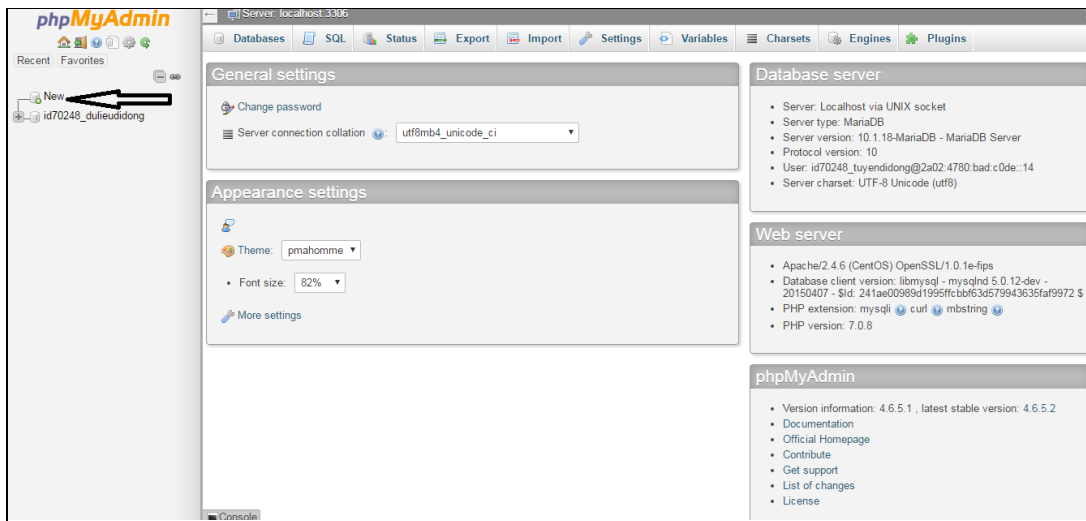
Nhập thông tin cho database cần tạo vào các ô như trong ảnh:



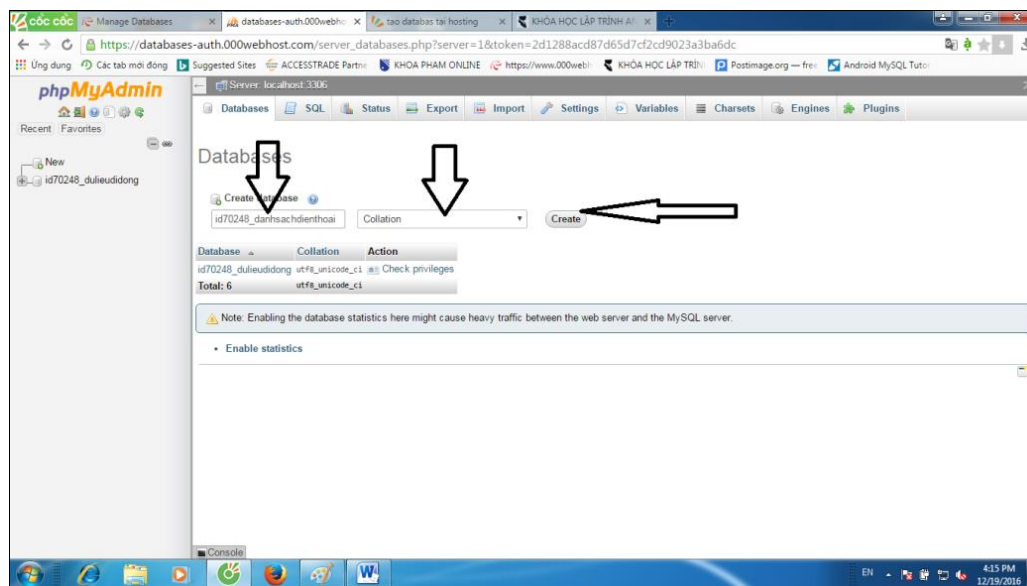
Ấn Create để hoàn tất tạo database.



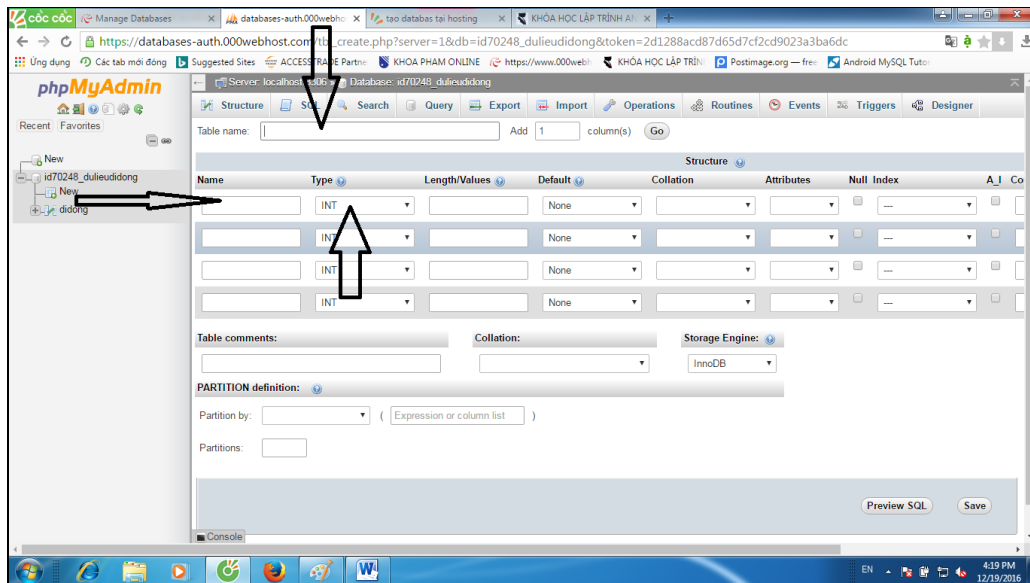
Sau khi tạo thành công ta có giao diện mới như sau:



Click vào *New* và điền thông tin còn trống sau đó click vào *Create* để tiến hành tạo database mới như hình:

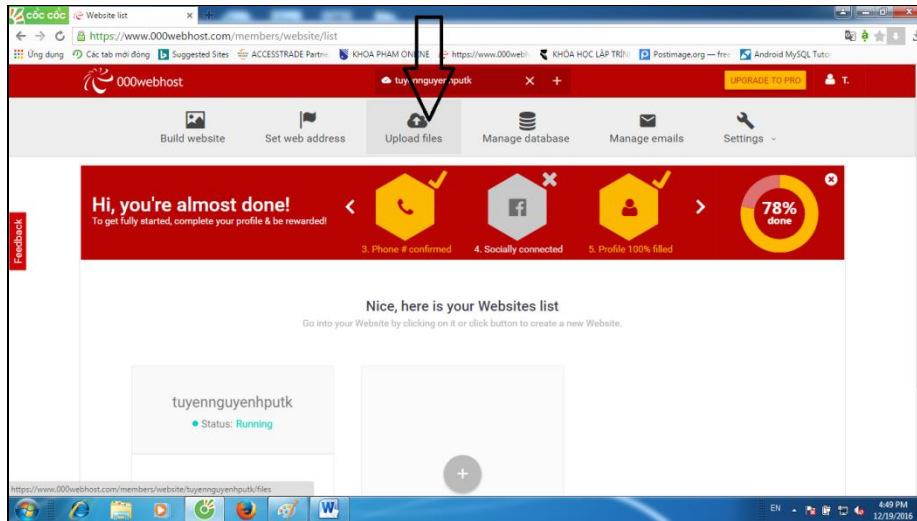


Nhập thông số cho bảng dữ liệu muốn tạo như: tên bảng tên hàng tên cột số hàng số cột, kiểu dữ liệu của hàng cột trong bảng... sau đó ấn **Save** để hoàn tất quá trình tạo bảng mới database.

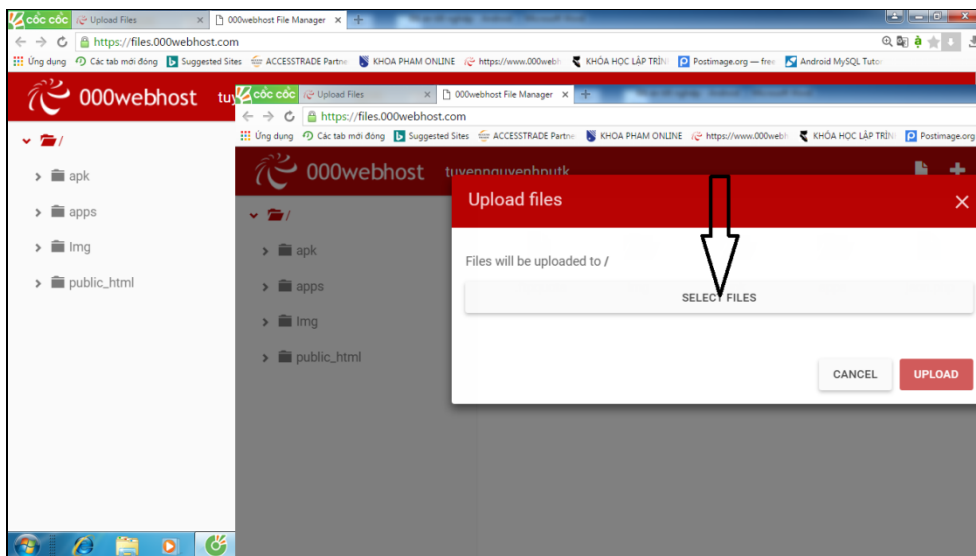


- Upload cơ sở dữ liệu lên hosting

Click vào **Upload files**



Click **SELECT FILES** để chọn cơ sở dữ liệu cần upload lên:



2.2.2. Một số kỹ thuật lập trình cơ sở dữ liệu trên internet

a. Xây dựng Webservice

Trên thực tế ứng dụng trên Android muốn trao đổi thông tin được với cơ sở dữ liệu trên Internet thì phải thông qua Webservice. Vì vậy việc xây dựng Webservice là rất quan trọng cho việc viết chương trình ứng dụng Android với cơ sở dữ liệu Internet. Muốn xây dựng được Webserver ta cần nắm được kỹ năng lập trình PHP và các kỹ thuật kết nối liên quan đến CSDL.

Ta tiến hành tạo Webservice theo các bước sau:

- Bước 1 Xây dựng chức năng kết nối CSDL với Android cho Webservice.

```

3  $servername = "localhost";
4  //34$host="127.0.0.1";
5  $username="id70248_tuyendidong";
6  $password="123456";
7  $link= new mysqli($servername,$username,$password);
8  if($link->connect_error)
9  {
10     die("Could not connect to Server".mysqli_errno($link));
11 }
12 //else echo "Connect to Server successfully!".<br/>";
13 $databasename="id70248_dulieudidong";
14 $test=mysqli_select_db($link,$databasename);
15 if(!$test)
16 {
17     die("Could not connect to Database!".mysqli_errno($link));
18 }

```

- Bước 2: Xây dựng chức năng truy vấn CSDL và xem CSDL cho Webservice rồi chuyển sang dạng mã JSON.

```

$action=$_POST["action"];
if($action=="view")
{
    mysqli_set_charset("utf-8");
    $query='SELECT * FROM didong';
    $listDidong=mysqli_query($link,$query);
    if(!$listDidong)
    {
        die("Truy vấn dữ liệu thất bại".mysqli_error($link));
    }
    else
    {
        $arrayDidong=array();
        while($row=mysqli_fetch_array($listDidong))
        {
            $id=$row["id"];
            $tendienthoai=$row["ten"];
            $namsanxuat=$row["year"];
            $gia=$row["gia"];
            $pic=$row["hinh_anh"];

            array_push($arrayDidong,new Didong($id, $tendienthoai, $namsanxuat, $gia,$pic));
        }
        echo json_encode($arrayDidong);
    }
}

```

- Bước 3: Xây dựng chức năng thêm dữ liệu vào CSDL cho Websevice.

```

else if($action=="insert")
{
    $id=$_POST["id"];
    $tendienthoai=$_POST["tendienthoai"];
    $namsanxuat=$_POST["namsanxuat"];
    $gia=$_POST["gia"];
    $hinh_anh=$_POST["hinh_anh"];
    $query="insert into didong (id,ten,year,gia,hinh_anh) values (null, '$tendienthoai', $namsanxuat,$gia,
    $kt=mysqli_query($link,$query);
    if(!$kt)
    {
        echo "Thêm dữ liệu thất bại!";
        die("Thêm dữ liệu thất bại!".mysqli_error($link));
    }
    else echo "Thêm dữ liệu thành công!";
    mysqli_close($link);
}

```

- Bước 4: Xây dựng chức năng sửa dữ liệu trên CSDL cho Websevice.

```

else if($action=="modify")
{
    $id=$_POST["id"];
    $tendienthoai=$_POST["tendienthoai"];
    $namsanxuat=$_POST["namsanxuat"];
    $gia=$_POST["gia"];
    $anh=$_POST["hinh_anh"];
    $query="update didong set ten='$tendienthoai',year=$namsanxuat, gia=$gia, hinh_anh='$anh' where id=$id";
    $kt=mysqli_query($link,$query);
    if(!$kt)
    {
        die("Sửa dữ liệu thất bại!".mysqli_error($link));
    }
    else echo "Sửa dữ liệu thành công!";
    mysqli_close($link);
}

```

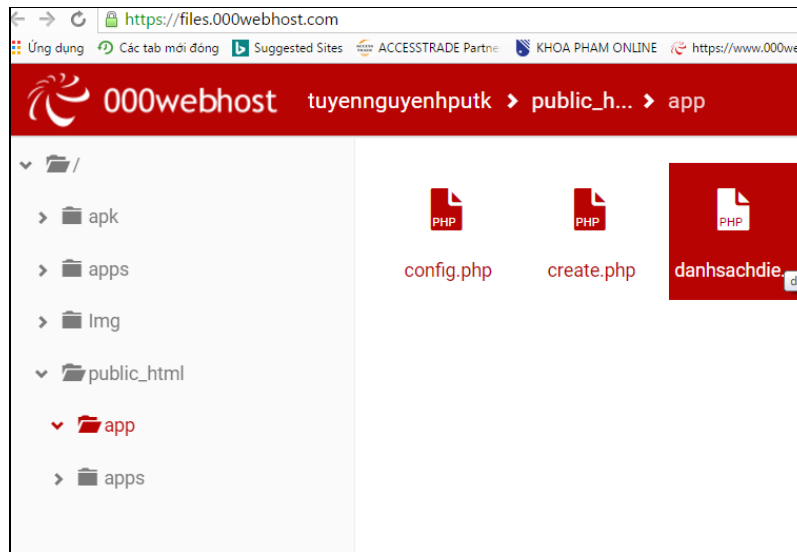
- Bước 5: Xây dựng chức năng xóa dữ liệu trên CSDL cho Websevice.

```

else if($action=="delete")
{
    $idxoa=$_POST["id"];
    $query="delete from didong where id=$idxoa";
    $kt=mysqli_query($link,$query);
    if(!$kt)
    {
        echo "Xóa dữ liệu thất bại!";
        die("Xóa dữ liệu thất bại".mysql_error($link));
    }
    else echo "Xóa dữ liệu thành công!";
    mysqli_close($link);
}

```


- Bước 6: Upload Webservice lên hosting sau đó lấy đường dẫn để chạy



"

<https://tuyennghuyenhputk.000webhostapp.com/app/danh sach dienthoai.php>"

Như vậy ta đã tạo dựng xong một Webservice với các chức năng cơ bản như: kết nối, truy vấn lấy dữ liệu về xem, thêm dữ liệu, sửa dữ liệu và xóa dữ liệu trên CSDL internet.

b. Kỹ thuật kiểm tra cấp quyền kết nối internet thiết bị.

Trong các ứng dụng liên quan tới server, truy cập dữ liệu từ internet luôn đòi hỏi thiết bị luôn phải kết nối internet. Trong ứng dụng thực tế, ta cần phải thiết lập việc kiểm tra kết nối một cách liên tục. Và khi xử lý một vấn đề gì có liên quan tới kết nối, ta phải kiểm tra xem thiết bị có kết nối internet hay không.

Để kiểm tra kết nối internet ta làm như sau:

- Bước 1: Cấp quyền truy cập internet trong AndroidManifest.xml

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

- Bước 2 Xây dựng hàm kiểm tra kết nối

```
public class KiemTraKetNoiInternet {
    private Context _context;

    public KiemTraKetNoiInternet(Context context) {
        this._context = context;
    }

    public boolean checkMobileInternetConn() {
        //Tạo đối tượng ConnectivityManager để trả về thông tin mạng
        ConnectivityManager connectivity = (ConnectivityManager) _context
            .getSystemService(Context.CONNECTIVITY_SERVICE);

        if (connectivity != null) {
            //Nhận thông tin mạng
            NetworkInfo info =
                connectivity.getNetworkInfo(ConnectivityManager.TYPE_MOBILE);
            if (info != null) {
```

```

        //Tìm kiếm thiết bị đã kết nối được internet chưa
        if (info.isConnected()) {
            return true;
        }
    }
    return false;
}
}

```

- Bước 3: Xây dựng code trong MainActivity.java

```

public class MainActivity extends Activity{
    Boolean isConnectionExist = true;
    KiemTraKetNoiInternet cd;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        cd = new KiemTraKetNoiInternet(getApplicationContext());
        ktInternet();
    }
    // hàm thiết lập Dialog
    public void showAlertDialog(Context context, String title, String message,
    Boolean status) {
        AlertDialog.Builder alertDialog=new
        AlertDialog.Builder(MainActivity.this);
        alertDialog.setTitle(title);
        alertDialog.setMessage(message);
        alertDialog.setPositiveButton("Yes", new DialogInterface.
        OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                // hàm bật chức wifi cho thiết bị
                WifiManager wifiManager =
                (WifiManager)MainActivity.this.getSystemService(Context.WIFI_SERVICE);
                wifiManager.setWifiEnabled(true);
                MainActivity.this.recreate();
            }
        });
        alertDialog.setNegativeButton("Cancel", new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                dialog.cancel();
            }
        });
        alertDialog.show();
    }
    // hàm kiểm tra trạng thái kết nối internet
    public void ktInternet(){
        //Lấy trạng thái kết nối
        isConnectionExist = cd.checkMobileInternetConn();
        //Kiểm tra trạng thái và hiển thị Dialog
        if (isConnectionExist) {
            showAlertDialog(MainActivity.this, "THÔNG BÁO",
            "Không có kết nối internet!Bạn có muốn cài internet
            không?", false);
        } else {
            showAlertDialog(MainActivity.this, "Kết nối Internet thành
            công",
            "", true);
        }
    }
}

```

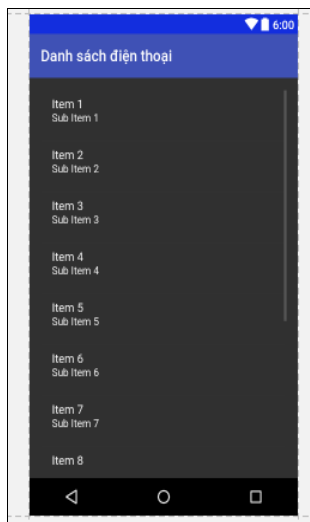
c. Kỹ thuật lấy dữ liệu từ CSDL trên internet về thiết bị di động.

Trong các ứng dụng cần lưu trữ và hiển thị theo danh sách các thông tin đa phần ta dùng ListView.

Để lấy được dữ liệu trên internet về thiết bị di động thì ta cần kết nối và lấy dữ liệu từ server thông qua Webservice có dữ liệu mã hóa dạng JSON.

Để lấy và hiển thị dữ liệu ở cơ sở dữ liệu trên internet về thiết bị di động ta làm như sau:

- Bước 1: Tạo giao diện file activity_main.xml



- Bước 2: Xây dựng code trong MainActivity.java

```
public class MainActivity extends Activity{
    ListView lvdidong;
    ArrayList<DiDong> mangdienthoai;
    ListAdapter adapter;
    mangdienthoai = new ArrayList<DiDong>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lvdidong = (ListView) findViewById(R.id.lvdidong);
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                new
                // gọi tới websevice để thực hiện chức năng kết nối, truy vấn và xem
                XemDuLieu().execute("https://tuyennguyennhputk.000webhostapp.com/app/index.p
                hp");
            }
        });
        // xây dựng hàm hiển thị dữ liệu
        private class XemDuLieu extends AsyncTask<String,Integer,String>
        {
            @Override
            protected String doInBackground(String... params) {
                String jSon = null;
                jSon = makePostRequestXemDL(params[0]);
                return jSon;
            }
            @Override
            protected void onPostExecute(String s) {
```

```

        try
        {
            final JSONArray mangJson= new JSONArray(s);
            for(int i = 0;i<mangJson.length();i++)
            {
                JSONObject jsonObject = mangJson.getJSONObject(i);
                mangdienthoai.add(
                    new DiDong(jsonObject.getInt("id"),
                                jsonObject.getString("tendienthoai"),
                                jsonObject.getInt("namsanxuat"),
                                jsonObject.getInt("gia"),
                                jsonObject.getString("pic")));
            }
            adapter= new
ListAdapter(MainActivity.this,R.layout.listview_row,mangdienthoai);
            lvdidong.setAdapter(adapter);
        }
        catch (JSONException e)
        {
            Toast.makeText(MainActivity.this,
e+"" ,Toast.LENGTH_LONG).show();
        }
    }
}
private String makePostRequestXemDL(String theURL) {
    HttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost(theURL);
    List nameValuePair = new ArrayList(1);
    nameValuePair.add(new BasicNameValuePair("action", "view"));
    try {
        httpPost.setEntity(new UrlEncodedFormEntity(nameValuePair));
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    String kq = "";
    try {
        HttpResponse response = httpClient.execute(httpPost);
        HttpEntity entity = response.getEntity();
        kq = EntityUtils.toString(entity);
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return kq;
}
}

```

- Bước 3: Xây dựng code trong ListAdapter.java

```

public class ListAdapter extends ArrayAdapter<DiDong> {
    public ListAdapter(Context context, int resource, List<DiDong> items)
    {
        super(context, resource, items);
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

        View view = convertView;
        if (view == null) {
            LayoutInflater inflater = LayoutInflater.from(getContext());
            view = inflater.inflate(R.layout.listview_row, null);
        }
        DiDong p = getItem(position);
        if (p != null) {
            // Anh xa + Gan gia tri
            TextView tvten = (TextView) view.findViewById(R.id.tvten);
            tvten.setText("Tên Điện Thoại: "+p.tendienthoai);
            TextView tvnamsx = (TextView) view.findViewById(R.id.tvnamsx);

```

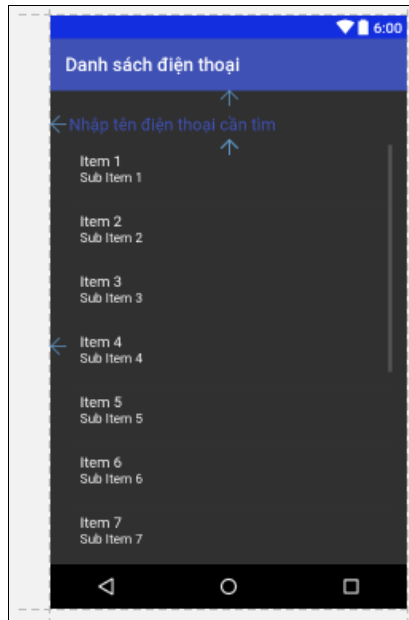
```
        tvnamsx.setText("Năm Sản Xuất: "+p.namsanxuat);
        TextView tvgia = (TextView) view.findViewById(R.id.tvgia);
        tvgia.setText("Giá Tiền: "+p.gia+ "VND");
        ImageView imgHinhAnh = (ImageView)
view.findViewById(R.id.imghinhanh);
        Picasso.with(getContext()).load(p.anh).into(imgHinhAnh);
    }
    return view;
}
}
```

d. Kỹ thuật tìm kiếm dữ liệu đã được hiển thị về trên thiết bị di động

Kỹ thuật này nhằm giúp người sử dụng dễ dàng tìm đến dữ liệu cần xem hay chỉnh sửa trong cơ sở dữ liệu một cách nhanh chóng.

Để thực hiện kỹ thuật này ta làm các bước sau:

- Bước 1: Tạo giao diện chính



- Bước 2: Xây dựng code trong MainActivity.java

```
public class MainActivity extends Activity{
    //khai báo biến
    ListView lvdidong;
    EditText inputSearch;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // ánh xạ
        lvdidong = (ListView) findViewById(R.id.lvdidong);
        inputSearch = (EditText) findViewById(R.id.inputSearch);
        mangdienthoai = new ArrayList<DiDong>();
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                new
                XemDuLieu().execute("https://tuyennguyenhputk.000webhostapp.com/app/ind
                ex.php");
            }
        });
        inputSearch.addTextChangedListener(new TextWatcher()
        {
            @Override
            public void onTextChanged(CharSequence charSequence, int i, int i1,
            int i2)
            {
                if(charSequence.toString().equals(""))
                {
                    new
                    XemDuLieu().execute("https://tuyennguyenhputk.000webhostapp.com/app/ind
                    ex.php");
                }
            }
        }
        else
```

```
        {
            timTheoTenDienThoai(charSequence.toString());
        }
    }

    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1,
    int i2)
    {
    }

    @Override
    public void afterTextChanged(Editable arg0)
    {
    }
    }

    );
}
// hàm tìm kiếm theo tên
private void timTheoTenDienThoai(String tenDienThoai)
{
    for(int i=0; i<mangdienthoai.size(); i++)
    {
        if(!mangdienthoai.get(i).tendienthoai.contains(tenDienThoai))
        {
            mangdienthoai.remove(i);
        }
    }
    adapter.notifyDataSetChanged();
}
}
```

e. Kỹ thuật thêm dữ liệu từ thiết bị di động lên CSDL trên internet

Để thực hiện thêm được dữ liệu từ ứng dụng trên thiết bị di động vào CSDL internet thì ta cần kết nối và đẩy được dữ liệu từ thiết bị di động thông qua Webservice có dữ liệu mã hóa dạng JSON để thêm vào các dữ liệu tương ứng đã được quy ước giữa CSDL và Webservice.

- Bước 1: Tạo giao diện cho màn hình thêm như sau:



- Bước 2: Xây code trong AddDiDong.java

```

public class AddDiDong extends AppCompatActivity {
    TextView tvaddtten, tvaddnam, tvaddgia, tvaddvnd, tvaddhinhanh;
    EditText edtaddtten, edtaddnam, edtaddgia, edtaddanh;
    Button btnadd, btnhuyadd;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_di_dong);
        edtaddtten=(EditText) findViewById(R.id.edtaddtten);
        edtaddtten.setText("");
        edtaddnam=(EditText) findViewById(R.id.edtaddnam);
        edtaddnam.setText("");
        edtaddgia=(EditText) findViewById(R.id.edtaddgia);
        edtaddgia.setText("");
        edtaddanh=(EditText) findViewById(R.id.edtaddanh);
        edtaddanh.setText("");
        btnhuyadd=(Button) findViewById(R.id.btnhuyadd);
        btnhuyadd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
        btnadd = (Button) findViewById(R.id.btnadd);
        btnadd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        new
                        //gọi webservice để thực hiện chức năng thêm của websevice
                        ThemDuLieu().execute("https://tuyennguyenhputk.000webhostapp.com/app/index.ph
                        p");
                        Intent myIntent = new
                        Intent(AddDiDong.this, MainActivity.class);
                        AddDiDong.this.startActivity(myIntent);
                    }
                });
            }
        });
    }
    // hàm thêm dữ liệu
    private class ThemDuLieu extends AsyncTask<String,Integer,String>
    {
        @Override
        protected String doInBackground(String... params) {
            String them;
            them= makePostRequest(params[0]);
            return them;
        }
        @Override
        protected void onPostExecute(String s) {
            Toast.makeText(getApplicationContext(),
            s,Toast.LENGTH_LONG).show();
        }
    }
    // hàm truyền tham số từ ứng dụng lên websevice
    private String makePostRequest(String url) {
        HttpClient httpClient = new DefaultHttpClient();
        // URL của trang web nhận request
        HttpPost httpPost = new HttpPost(url);
        // Các tham số truyền
        List nameValuePair = new ArrayList(6);
        nameValuePair.add(new BasicNameValuePair("action", "insert"));
    }
}

```



```

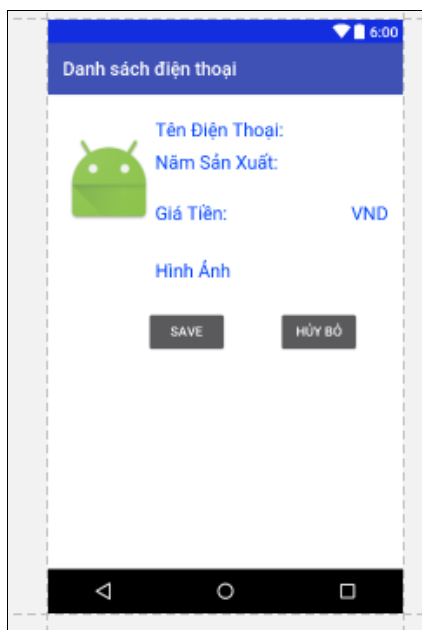
        nameValuePair.add(new BasicNameValuePair("id", null));
        nameValuePair.add(new BasicNameValuePair("tendienthoai",
edtaddten.getText()+""));
        nameValuePair.add(new BasicNameValuePair("namsanxuat",
edtaddnam.getText()+""));
        nameValuePair.add(new BasicNameValuePair("gia",
edtaddgia.getText()+""));
        nameValuePair.add(new BasicNameValuePair("hinh_anh",
edtaddanh.getText()+""));
        try {
            httpPost.setEntity(new UrlEncodedFormEntity(nameValuePair));
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        String kq = "";
        try {
            HttpResponse response = httpClient.execute(httpPost);
            HttpEntity entity = response.getEntity();
            kq = EntityUtils.toString(entity);
        }
        catch (ClientProtocolException e) {
            e.printStackTrace();
        }
        catch (IOException e) {
            e.printStackTrace();
        }
        return kq;
    }
}

```

f. Kỹ thuật sửa dữ liệu từ thiết bị di động

Để thực hiện sửa được dữ liệu bằng ứng dụng trên thiết bị di động thay thế vào CSDL internet thì ta cần kết nối và đẩy được dữ liệu đã sửa từ thiết bị di động thông qua Webservice có dữ liệu mã hóa dạng JSON để thay thế vào các dữ liệu tương ứng đã được quy ước giữa CSDL và Webservice.

- Bước 1: Tạo giao diện cho màn hình sửa như sau:



- Bước 2: Xây dựng code trong MainActivity.java

```

lvdidong.setOnClickListener(new AdapterView.OnItemClickListener()
// xử lý sự kiện để chuyển sang màn hình sửa
{
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        Intent myIntent = new Intent(MainActivity.this, SuaActivity.class);
        myIntent.putExtra("ID", mangdienthoai.get(position).id);
        myIntent.putExtra("TenDienThoai",
mangdienthoai.get(position).getTendienthoai());
        myIntent.putExtra("Anh", mangdienthoai.get(position).getAnh());
        myIntent.putExtra("NamSX", mangdienthoai.get(position).getNamSX());
        myIntent.putExtra("Gia", mangdienthoai.get(position).getGia());
        MainActivity.this.startActivity(myIntent);
    }
});

```

- Bước 3: Xây dựng code trong SuaActivity.java

```

public class SuaActivity extends AppCompatActivity {
    Button btnSave, btnHuy;
    TextView tvidt, tvten_t, tvnamsx_t, tvgt_t;
    EditText edtGia_t, edttend_t, edtnamsx_t, edthinhanht;
    ImageView imghinhanht;
    String tenSP, anh;
    Integer id, gia, namSX;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sua);
        Intent intent = getIntent();
        id = intent.getIntExtra("ID", 0);
        tenSP = intent.getStringExtra("TenDienThoai");
        anh = intent.getStringExtra("Anh");
        gia = intent.getIntExtra("Gia", 0);
        namSX = intent.getIntExtra("NamSX", 0);
        edttend_t = (EditText) findViewById(R.id.edttend_t);
        edttend_t.setText(" "+tenSP);
        imghinhanht = (ImageView) findViewById(R.id.imghinhanht);
        Picasso.with(this).load(anh).into(imghinhanht);
        edtGia_t = (EditText) findViewById(R.id.edtGia_t);
        edtGia_t.setText(" "+gia);
        edtnamsx_t = (EditText) findViewById(R.id.edtnamsx_t);
        edtnamsx_t.setText(" "+namSX);
        edthinhanht = (EditText) findViewById(R.id.edthinhanht);
        edthinhanht.setText(" "+anh);
        btnSave = (Button) findViewById(btnsave);
        btnSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        new
                        SuaDuLieu().execute("https://tuyennguyenhputk.000webhostapp.com/app/index.p
                        hp");
                        Intent myIntent = new
                        Intent(SuaActivity.this, MainActivity.class);
                        SuaActivity.this.startActivity(myIntent);
                    }
                });
            }
        });
        btnHuy = (Button) findViewById(R.id.btnhuy);
        btnHuy.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

        public void onClick(View v) {
            finish();
        }
    });
}
// hàm sửa dữ liệu
private class SuaDuLieu extends AsyncTask<String,Integer,String>
{
    @Override
    protected String doInBackground(String... params) {
        String thongbao;
        thongbao= makePostRequest(params[0]);
        return thongbao;
    }
    @Override
    protected void onPostExecute(String s) {
        Toast.makeText(getApplicationContext(),
s,Toast.LENGTH_LONG).show();
    }
}
private String makePostRequest(String url) {
    HttpClient httpClient = new DefaultHttpClient();
    // URL của trang web nhận request
    HttpPost httpPost = new HttpPost(url);
    // Các tham số truyền
    List nameValuePair = new ArrayList(6);
    nameValuePair.add(new BasicNameValuePair("action", "modify"));
    nameValuePair.add(new BasicNameValuePair("id",id.toString()));
    nameValuePair.add(new BasicNameValuePair("tendienthoai",
edtTendnt_t.getText()+""));
    nameValuePair.add(new BasicNameValuePair("namsanxuat",
edtNamsx_t.getText()+""));
    nameValuePair.add(new BasicNameValuePair("gia",
edtGia_t.getText()+""));
    nameValuePair.add(new BasicNameValuePair("hinh_anh",
edtHinhhanht.getText()+""));
    try {
        httpPost.setEntity(new UrlEncodedFormEntity(nameValuePair));
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    String kq = "";
    try {
        HttpResponse response = httpClient.execute(httpPost);
        HttpEntity entity = response.getEntity();
        kq = EntityUtils.toString(entity);
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return kq;
}
}

```

g. Kỹ thuật xóa dữ liệu từ thiết bị di động

Để thao tác xóa được dữ liệu trên thiết bị di động thay vì xóa trên CSDL internet thì ta cần kết nối và đẩy được lệnh xóa từ thiết bị di động thông qua Webservice có dữ liệu mã hóa dạng JSON để xóa các dữ liệu tương ứng đã được quy ước giữa CSDL và Webservice.

- Xây dựng code trong MainActivity.java

+ Bắt và xử lý sự kiện LongClickListener cho ListView

```

lvdidong.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public boolean onItemClick(AdapterView<?> parent, View view, final
int position, long id) {
        idxoa=mangdienthoai.get(position).id;
        // thiết lập Dialog thông báo lên màn hình
        AlertDialog.Builder b=new AlertDialog.Builder(MainActivity.this);
        b.setTitle("Xác nhận xóa");
        b.setMessage("Bạn có muốn xóa không?");
        b.setPositiveButton("Có", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                new
Delete().execute("https://tuyennguyenhputk.000webhostapp.com/app/index.php");
                adapter.remove(adapter.getItem(position));
                adapter.notifyDataSetChanged();
            }
        });
        b.setNegativeButton("Không", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                dialog.cancel();
            }
        });
        b.create().show();
        return true;
    }
});

```

+ Xây dựng hàm xóa dữ liệu

```

private class Delete extends AsyncTask<String,Integer,String>{
    @Override
    protected String doInBackground(String... params) {
        String xoa;
        xoa= makePostRequestXoa(params[0]);
        return xoa;
    }
    @Override
    protected void onPostExecute(String s) {
        Toast.makeText(getApplicationContext(),s,Toast.LENGTH_LONG).show();
    }
}

private String makePostRequestXoa(String url) {
    HttpClient httpClient = new DefaultHttpClient();
    // URL của trang web nhận request
    HttpPost httpPost = new HttpPost(url);
    // Các tham số truyền
    List nameValuePair = new ArrayList(1);
    nameValuePair.add(new BasicNameValuePair("action", "delete"));
    nameValuePair.add(new BasicNameValuePair("id",idxoa+""));
    try {
        httpPost.setEntity(new UrlEncodedFormEntity(nameValuePair));
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
    String kq = "";
    try {
        HttpResponse response = httpClient.execute(httpPost);
        HttpEntity entity = response.getEntity();
        kq = EntityUtils.toString(entity);
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    }
}

```

```
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    return kq;  
}
```

Chương 3: CHƯƠNG TRÌNH THỰC NGHIỆM

3.1. Bài toán

Điện thoại Smart Phone chạy hệ điều hành Android ngày càng trở nên không thể thiếu được đối với chúng ta. Chúng vừa là phương tiện giúp chúng ta liên lạc, vừa là phương tiện giúp chúng ta học tập và tra cứu thông tin. Xuất phát từ nhu cầu thực tế, công ty điện thoại tư nhân X chuyên kinh doanh về điện thoại thông minh (smart phone). Giám đốc công ty thường đi công tác, ít có mặt tại công ty. Giám đốc có nhu cầu theo dõi, cập nhật thông tin điện thoại trực tuyến qua mạng internet mọi nơi mọi lúc thông qua smart phone chạy hệ điều hành Android. Các thao tác cần thực hiện trên cơ sở dữ liệu internet :

- Xem danh sách điện thoại: Lấy dữ liệu từ cơ sở dữ liệu trên internet về, hiển thị trên android smart phone.
- Thêm điện thoại: Bổ sung dữ liệu cho loại điện thoại mới vào cơ sở dữ liệu internet từ android smart phone.
- Thay đổi thông tin điện thoại: Sửa dữ liệu cho điện thoại đã có trong cơ sở dữ liệu internet từ android smart phone.
- Loại bỏ thông tin điện thoại: Xóa dữ liệu của một loại điện thoại đã có trong cơ sở dữ liệu internet từ android smart phone.

Chương trình này sẽ giúp giám đốc thực hiện nhu cầu trên một cách dễ dàng.

3.2. Thiết kế cơ sở dữ liệu

- Cơ sở dữ liệu tạo dựng là bảng "didong" gồm các cột như hình ảnh:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop Primary Unique Index Spatial More
2	ten	varchar(100)	utf8_general_ci		No	None			Change Drop Primary Unique Index Spatial More
3	year	int(11)			No	None			Change Drop Primary Unique Index Spatial More
4	gia	int(11)			No	None			Change Drop Primary Unique Index Spatial More
5	hinh_anh	varchar(100)	utf8_general_ci		No	None			Change Drop Primary Unique Index Spatial More

- Thực hiện Insert dữ liệu mô phỏng vào bảng ta có CSDL như ảnh:

id	ten	year	gia	hình_ảnh
11	Iphone 5s	2012	4100000	https://s6.postimg.org/it9i1rjpt/63586912377386995...
18	MEIZU PRO 6	2014	8490000	https://s6.postimg.org/lapumrt01/63604865341216121...
23	Samsung i8520 Beam	2010	2380000	https://s6.postimg.org/ijlqepk901/t_i_xu_ng_10.jpg
25	Oppo neo 7plus	2015	8980000	https://s6.postimg.org/eiq8sfp8x/63586899338591955...
26	Oppo neo 7	2015	8980000	https://s6.postimg.org/mnc52eftd/t_i_xu_ng_5.jpg
27	L300 g1	2009	650000	https://s6.postimg.org/r7p4pukr5/63600389549691323...
30	Iphone 6s	2015	19000000	https://s6.postimg.org/5dmhcbb81/63587081188960175...
31	Iphone 6s Plus	2015	21000000	https://s6.postimg.org/roj3m9t/images.jpg
33	Samsung galaxy s7	2016	16000000	https://s6.postimg.org/663e040i9/t_i_xu_ng_11.jpg
36	Oppo neo 7	2015	6700000	https://s6.postimg.org/eiq8sfp8x/63586899338591955...
37	Samsung A5 Gold	2014	8900000	https://s6.postimg.org/l14arsp75/63585164142386148...
38	Swim	2009	360000	https://s6.postimg.org/c47zhhnr1/63607043208774763...
39	Samsung G1	2008	1500000	https://s6.postimg.org/slvignjy5/63598401934222311...
40	Huawei Y6	2009	4900000	https://s6.postimg.org/its43t7c7l/63605640322965537...
41	Oppo A37 Gold	2013	5200000	https://s6.postimg.org/nm2nmgia9/63617147837586135...

- Thông tin CSDL trên hosting

Database server

- Server: Localhost via UNIX socket
- Server type: MariaDB
- Server version: 10.1.18-MariaDB - MariaDB Server
- Protocol version: 10
- User: id70248_tuyendidong@2a02:4780:bad:c0de::14
- Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.6 (CentOS) OpenSSL/1.0.1e-fips
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 241ae00989d1995ffcbbf63d579943635faf9972 \$
- PHP extension: mysqli [c](#) curl [m](#) mbstring [b](#)
- PHP version: 7.0.8

phpMyAdmin

- Version information: 4.6.5.1
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes

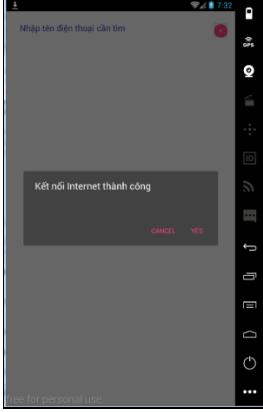
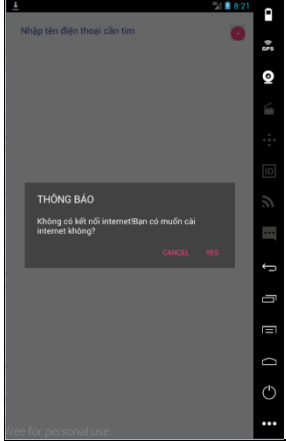
3.3. Mô hình chương trình



3.4. Giao diện chương trình

- Là chương trình xây dựng ứng dụng truy xuất CSDL trên internet lên phần kiểm tra kết nối internet của thiết bị vô cùng quan trọng.

Sau đây là giao diện ứng dụng khi thực kiểm tra kết nối Internet.

Có kết nối internet	Không có kết nối internet
	

- Giao diện ứng dụng khi đang xem dữ liệu.

Theo chiều dọc	Theo chiều ngang
	

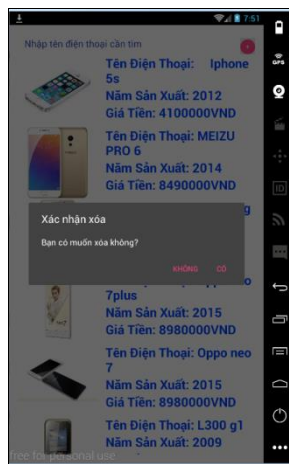
- Giao diện ứng dụng ở màn hình sửa thông tin dữ liệu



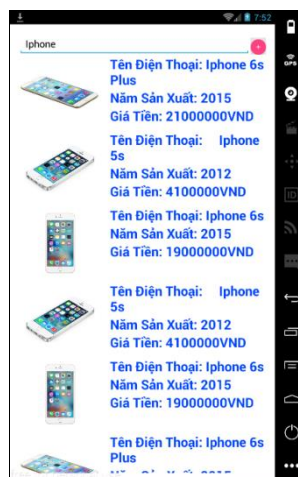
- Giao diện ứng dụng khi thực hiện chức năng thêm dữ liệu



- Giao diện ứng dụng khi thực hiện chức năng xóa dữ liệu



- Giao diện ứng dụng khi thực hiện chức năng tìm kiếm thông tin dữ liệu



3.5. Kết quả đạt được

Bước đầu đồ án đã đạt được kết quả như sau

- Tạo lập thành công cơ sở dữ liệu trên hosting.

- Tạo lập được trang Webservice trao đổi dữ liệu giữa ứng dụng Android và CSDL trên hosting
- Xây dựng thành công ứng dụng android đã làm có thể thao tác được trên CSDL khi có kết nối internet như: thêm mới, sửa, xóa.

Tuy nhiên vẫn còn tồn tại các hạn chế sau:

- Ứng dụng chưa có chức năng lưu trữ dữ liệu offline cho phép người dùng xem lại dữ liệu khi ứng dụng không truy xuất được CSDL trên internet
- Ứng dụng chưa có chức năng upload ảnh cục bộ lên cơ sở dữ liệu.

KẾT LUẬN

Trên đây em đã khảo sát trên mặt lý thuyết đối với xây dựng ứng dụng Android. Đồ án hướng tới mục tiêu xây dựng ứng dụng Android truy xuất cơ sở dữ liệu trên Internet. Trong khoảng thời gian nhất định dành cho việc thực hiện đề tài, nên một số vấn đề vẫn chưa được hoàn chỉnh. Tuy nhiên, đồ án đã đạt được một số kết quả:

- Về lý thuyết: Tìm hiểu, nghiên cứu được cách tạo cơ sở dữ liệu trên Internet, các kỹ thuật lập trình với cơ sở dữ liệu để xây dựng ứng dụng Android truy xuất cơ sở dữ liệu như: xây dựng Webservice, kiểm tra cấp quyền kết nối Internet, lấy dữ liệu từ Internet về Android, tìm kiếm dữ liệu, thêm, sửa, xóa dữ liệu từ Android lên CSDL Internet.
- Về thực nghiệm: Sử dụng các kỹ thuật lập trình với cơ sở dữ liệu Internet để xây dựng được ứng dụng truy xuất cơ sở dữ liệu với các thao tác với dữ liệu như: xem, sửa, thêm, xóa.

Do thời gian hạn chế, nên hiện tại đồ án mới chỉ dừng lại ở thao tác với dữ liệu khi kết nối Internet ổn định còn ứng dụng chưa có chức năng lưu trữ dữ liệu offline cho phép người dùng xem lại dữ liệu khi ứng dụng mất kết nối Internet truy xuất được CSDL trên Internet. Ứng dụng chưa có chức năng upload ảnh cục bộ lên CSDL trên Internet. Trong tương lai em sẽ tiếp tục phát triển ứng dụng hoàn chỉnh hơn.

TÀI LIỆU THAM KHẢO

Tài liệu tham khảo trực tuyến

- [1].[Online] <http://o7planning.org/>
- [2].[Online] <https://www.simplifiedcoding.net>
- [3].[Online] <http://khoapham.vn/KhoaPhamTraining/android/snippet/>