

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

---



**ISO 9001:2015**

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**Sinh viên : Nguyễn Ngọc Hải**

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

**HẢI PHÒNG - 2018**

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

---

**TÌM HIỂU DỊCH VỤ WEB CHO THIẾT BỊ DI ĐỘNG VÀ**  
**ỨNG DỤNG**

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

**Sinh viên : Nguyễn Ngọc Hải**

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

**HẢI PHÒNG - 2018**

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

---

**NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP**

**Sinh viên: Nguyễn Ngọc Hải**

**Mã SV: 1412101135**

**Lớp: CT1701**

**Ngành: Công nghệ thông tin**

**Tên đề tài: Tìm hiểu dịch vụ Web cho thiết bị di động và ứng dụng**

## LỜI MỞ ĐẦU

Mạng điện thoại di động xuất hiện tại Việt Nam từ đầu những năm 1990 và theo thời gian số lượng các thuê bao cũng như các nhà cung cấp dịch vụ di động tại Việt Nam ngày càng tăng. Do nhu cầu trao đổi thông tin ngày càng tăng và nhu cầu sử dụng sản phẩm công nghệ cao nhiều tính năng, cấu hình cao, chất lượng tốt, kiểu dáng mẫu mã đẹp, phong phú nên nhà cung cấp phải luôn luôn cải thiện, nâng cao những sản phẩm của mình. Do đó việc xây dựng các ứng dụng cho điện thoại di động đang là một ngành công nghiệp mới đầy tiềm năng và hứa hẹn nhiều sự phát triển vượt bậc của ngành khoa học kỹ thuật.

Cùng với sự phát triển của thị trường điện thoại di động là sự phát triển mạnh mẽ của xu hướng lập trình phần mềm ứng dụng cho các thiết bị di động. Phần mềm, ứng dụng cho điện thoại di động hiện nay rất đa dạng và phong phú trên các hệ điều hành di động cũng phát triển mạnh mẽ và đang thay đổi từ ngày. Các hệ điều hành J2ME, Android, IOS, Hybrid, Web based Mobile Application đã có rất phát triển trên thị trường truyền thông di động.

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

Ngày nay, với sự phát triển nhanh chóng của xã hội, nhu cầu mua bán các loại sản phẩm thông qua mạng Internet ngày càng phổ biến, vì vậy em đã chọn đề tài “**Tìm hiểu dịch vụ Web cho thiết bị di động và ứng dụng**” với mục đích nghiên cứu, tìm hiểu về dịch vụ web trong hệ điều hành Android và xây dựng ứng dụng mua bán các loại thiết bị điện tử để có thể đáp ứng được nhu cầu thiết yếu đó, giúp cho mọi người có thể tiết kiệm chi phí, thời gian đi lại và còn có thể cập nhật một cách nhanh chóng về thông tin sản phẩm thông qua ứng dụng một cách dễ dàng và tiện lợi.

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến quý thầy cô Trường Đại Học Dân Lập Hải Phòng, những người đã dìu dắt em tận tình, đã truyền đạt cho em những kiến thức và bài học quý báu trong suốt thời gian em theo học tại trường.

Em xin trân trọng gửi lời cảm ơn đến tất cả các thầy cô trong khoa Công Nghệ Thông Tin, đặc biệt là thầy giáo ThS. Phùng Anh Tuấn, thầy đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình làm tốt nghiệp. Với sự chỉ bảo của thầy, em đã có những định hướng tốt trong việc triển khai và thực hiện các yêu cầu trong quá trình làm đồ án tốt nghiệp.

Em xin cảm ơn những người thân và gia đình đã quan tâm, động viên và luôn tạo cho em những điều kiện tốt nhất trong suốt quá trình học tập và làm tốt nghiệp.

Ngoài ra, em cũng xin gửi lời cảm ơn tới tất cả bạn bè, đặc biệt là các bạn trong lớp CT1701 đã luôn gắn bó, cùng học tập và giúp đỡ em trong những năm qua và trong suốt quá trình thực hiện đồ án này.

Em xin chân thành cảm ơn!

*Hải Phòng, ....ngày .... tháng..... năm 2018*

Sinh viên

**Nguyễn Ngọc Hải**

# MỤC LỤC

CHƯƠNG 1: HỆ ĐIỀU HÀNH ANDROID.....	9
1.1. Giới thiệu chung về Hệ điều hành Android.....	9
1.2. Kiến trúc hệ điều hành Android .....	11
1.2.1. Tầng hạt nhân Linux.....	11
1.2.2. Tầng thư viện và thực thi Android.....	12
1.2.3. Tầng khung ứng dụng.....	13
1.2.4. Tầng ứng dụng.....	14
1.3. Mô tả hệ điều hành Android.....	15
1.3.1. Giao diện.....	15
1.3.2. Ứng dụng.....	17
1.4. Sự đón nhận của người dùng đối với hệ điều hành Android.....	18
CHƯƠNG 2: MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG ANDROID.....	20
2.1. Môi trường phát triển.....	20
2.1.1. Phần mềm Android Studio.....	20
2.1.2. Bộ công cụ phát triển Java SE Development Kit (JDK).....	20
2.1.3. Máy ảo Android Virtual Device (AVD).....	21
2.2. Các thành phần trong một dự án Android .....	21
2.2.1. Tập cấu hình dự án AndroidManifest.xml.....	21
a. Thành phần ứng dụng.....	21
b. Quyền truy xuất và sử dụng tài nguyên.....	22
c. Phiên bản SDK.....	22
2.2.2. Tập R.java.....	22
2.3. Vòng đời ứng dụng Android .....	22
2.3.1. Chu kỳ sống của các thành phần.....	22
2.3.2. Activity Stack.....	23
2.3.3. Các trạng thái của chu kỳ sống.....	23
2.3.4. Thời gian sống của Activity.....	24
2.3.5. Các phương thức của chu kỳ sống.....	24
2.4. Các thành phần giao diện trong Android.....	25
2.4.1. View và ViewGroup.....	25
a. TextView.....	26
b. EditText.....	26
c. Button.....	27
d. ImageView.....	28

e. ListView.....	28
2.4.2 <i>Layout</i> .....	29
a. RelativeLayout.....	29
b. TableLayout.....	30
c. FrameLayout.....	30
d. LinearLayout .....	30
CHƯƠNG 3: CƠ SỞ DỮ LIỆU WEB VÀ DỊCH VỤ WEB.....	32
3.1. Mạng Internet .....	32
3.1.1. <i>Khái niệm</i> .....	32
3.1.2. <i>Lợi ích của Internet trong cuộc sống</i> .....	32
3.1.3. <i>Các chương trình duyệt Web thông dụng hiện nay</i> .....	32
3.2. Web Hosting .....	33
3.2.1. <i>Khái niệm</i> .....	33
3.2.2. <i>Các loại hosting</i> .....	33
3.2.3. <i>Các thông số cần biết trong web hosting</i> .....	34
3.2.4. <i>Tại sao cần phải mua web hosting ?</i> .....	34
3.2.5. <i>Cách đăng kí một web hosting miễn phí</i> .....	34
3.3. Cơ sở dữ liệu web .....	38
3.3.1. <i>Khái niệm</i> .....	38
3.3.2. <i>Tổ chức dữ liệu</i> .....	39
3.3.3. <i>Phần mềm Cơ sở dữ liệu Web</i> .....	39
3.3.4. <i>Đối tượng áp dụng</i> .....	39
3.3.5. <i>Hệ quản trị CSDL MySQL</i> .....	39
3.4. Dịch vụ web .....	40
3.4.1. <i>Giới thiệu</i> .....	40
3.4.2. <i>Các công nghệ xây dựng dịch vụ Web</i> .....	40
3.4.3. <i>Đặc điểm của dịch vụ Web</i> .....	41
3.4.4. <i>Kiến trúc của dịch vụ Web</i> .....	41
3.4.5. <i>Các thành phần chính của dịch vụ web</i> .....	44
3.4.6. <i>Xây dựng một dịch vụ Web</i> .....	44
3.4.7. <i>Các dạng tương tác giữa Web Service với ứng dụng trên TBDD</i> .....	45
a. XML - eXtensible Markup Language .....	45
b. Định dạng JSON - JavaScript Object Notation .....	46
3.4.8. <i>Cấu trúc chung Web Service viết bằng PHP cho ứng dụng di động</i> .....	49
CHƯƠNG 4: CHƯƠNG TRÌNH THỬ NGHIỆM.....	50
4.1. Phát biểu bài toán .....	50
4.2. Thiết kế chức năng hệ thống.....	50

4.2.1. Chức năng xem sản phẩm theo loại sản phẩm .....	50
4.2.2. Chức năng quản lí giỏ hàng .....	51
4.2.3. Chức năng thanh toán sản phẩm .....	51
4.3. Thiết kế dữ liệu .....	52
4.3.1. Bảng thông tin sản phẩm .....	52
4.3.2. Bảng thông tin sản phẩm mới nhất .....	52
4.3.3. Bảng chi tiết đơn hàng .....	52
4.3.4. Bảng thông tin loại sản phẩm .....	53
4.3.5. Bảng thông tin đơn hàng .....	53
4.4. Một số giao diện chương trình .....	53
4.4.1 Giao diện màn hình chính và thanh menu của ứng dụng .....	53
4.4.2. Giao diện danh sách các sản phẩm của Điện thoại và Laptop .....	54
4.4.3. Giao diện thông tin chi tiết của các sản phẩm .....	54
4.4.4. Giao diện các sản phẩm trong giỏ hàng .....	55
4.4.5. Giao diện thanh toán sản phẩm và thông tin liên hệ .....	55
KẾT LUẬN .....	56
TÀI LIỆU THAM KHẢO .....	57



# CHƯƠNG 1: HỆ ĐIỀU HÀNH ANDROID

## 1.1. Giới thiệu chung về Hệ điều hành Android

Android là một hệ điều hành cho thiết bị di động dựa trên nền tảng linux phiên bản 2.6 dành cho các dòng điện thoại SmartPhone. Đầu tiên được ra đời bởi công ty liên hợp Android, sau đó được Google mua lại và phát triển từ năm 2005 và trở thành một hệ điều hành di động mã nguồn mở, miễn phí, mạnh mẽ và được ưa chuộng cao trên thế giới [1].



# ANDROID

*Hình 1.1. Biểu tượng của Hệ điều hành Android*

Hệ điều hành android một hệ điều hành rất mạnh mẽ, có bảo mật cao, hỗ trợ được nhiều công nghệ tiên tiến như 3G, GPS, EDGE, Wifi.. tương thích với nhiều phần cứng, hỗ trợ nhiều loại bộ nhập dữ liệu như keyboard, touch và trackball. Android là hệ điều hành di động nên có khả năng kết nối cao với các mạng không dây. Hỗ trợ công nghệ OpenGL nên có khả năng chơi các phương tiện media, hoạt hình cũng như trình diễn các khả năng đồ họa khác cực tốt, là tiền đề để phát triển các ứng dụng có giao diện phức tạp chẳng hạn như là các trò chơi [1].



*Hình 1.2. Các tính năng hỗ trợ trong Hệ điều hành Android*



Android liên tục được phát triển, mỗi bản cập nhật từ google là mỗi lần android được tối ưu hóa để hoạt động tốt hơn, nhanh và ổn định hơn, hỗ trợ thêm công nghệ mới. Chẳng hạn như theo một đánh giá thì android phiên bản 2.2 hoạt động nhanh hơn bản 2.1 tới 450%. Hiện nay, phiên bản mới nhất 2.3 phát hành ngày 6/12/2010 và đang tiếp tục được cập nhật.

Năm 2008, hệ điều hành android đã chính thức mở toàn bộ mã nguồn, điều đó cho phép các hãng điện thoại có thể đem mã nguồn về tùy chỉnh, thiết kế lại sao cho phù hợp với mỗi mẫu mã điện thoại của họ và điều quan trọng nữa là hệ điều hành mở này hoàn toàn miễn phí, không phải trả tiền nên giúp họ tiết kiệm khá lớn chi phí phát triển hệ điều hành. Những điều đó là cực kỳ tốt không chỉ đối với các hãng sản xuất điện thoại nhỏ mà ngay cả với những hãng lớn như Samsung, HTC....

Với Google, vì android hoàn toàn miễn phí, Google không thu tiền từ những hãng sản xuất điện thoại, tuy không trực tiếp hưởng lợi từ android nhưng bù lại, những dịch vụ của hãng như Google Search, Google Maps,... nhờ có android mà có thể dễ dàng xâm nhập nhanh vào thị trường di động vì mỗi chiếc điện thoại được sản xuất ra đều được tích hợp hàng loạt dịch vụ của Google. Từ đó hãng có thể kiếm bội, chủ yếu là từ các nguồn quảng cáo trên các dịch vụ đó [1].

Với các nhà phát triển ứng dụng (developers), việc hệ điều hành android được sử dụng phổ biến đồng nghĩa với việc họ có thể thoải mái phát triển ứng dụng trên nền android với sự tin tưởng là ứng dụng đó sẽ có thể chạy được ngay trên nhiều dòng điện thoại của các hãng khác nhau. Họ ít phải quan tâm là đang phát triển cho điện thoại nào, phiên bản bao nhiêu vì nền tảng android là chung cho nhiều dòng máy, máy ảo Java đã chịu trách nhiệm thực thi những ứng dụng phù hợp với mỗi dòng điện thoại mà nó đang chạy. Tất cả các chương trình ứng dụng được viết bằng ngôn ngữ Java kết hợp với XML nên có khả năng khả chuyên cao.

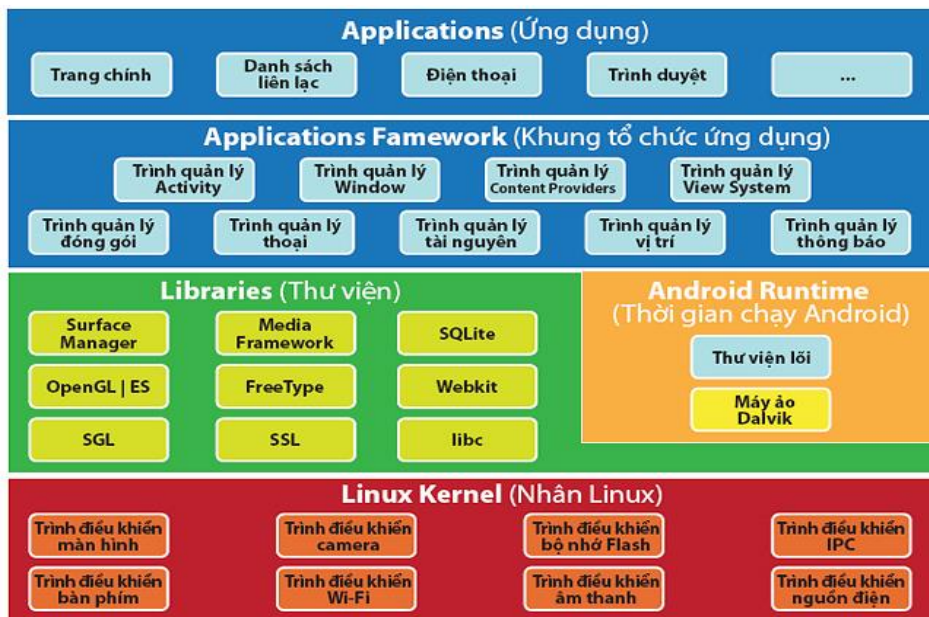
- Một số hãng sản xuất điện thoại có sử dụng hệ điều hành android tiêu biểu :

-  HTC với các dòng Desire HD, Evo 4G, DROID ERIS, Desire Z, Hero, Desire, Tattoo, Wildfire, Droid Incredible, Legend, Magic, Google Nexus One, Dream, Aria, Paradise .
-  LG với các dòng GT540 Optimus, Optimus Chic E720, Optimus One P500, GW620, Optimus Z, Optimus Q, KH5200 Andro-1, GW880, C710Aloha .

- ✚ MOTOROLA với các dòng MILESTONE 2, BACKFLIP, Droid XTreme, MT710 ZHILING, MILESTONE, XT720 MOTOROI, A1680, XT800 ZHISHANG, DEFY, CHARM, XT806...
- ✚ SAMSUNG với các dòng máy I9000 Galaxy S, Galaxy Tab, Epic 4G, i5510, I5500 Galaxy 5, I7500 Galaxy, I5800 Galaxy 3, M110S Galaxy S, I6500U Galaxy, Galaxy Q, I5700 Galaxy Spica, I8520 Galaxy Beam, I909 Galaxy S.
- ✚ SONY : XPERIA X10, XPERIA X10 mini, XPERIA X8.
- ✚ ACER với các dòng máy beTouch T500, Liquid E, Stream, Liquid, beTouch E110, beTouch E130, beTouch E400, beTouch E120, Liquid Metal.
- ✚ Ngoài ra còn nhiều hãng điện thoại vừa và nhỏ khác nữa cũng sử dụng hệ điều hành android trong sản phẩm của mình...[2].

## 1.2. Kiến trúc hệ điều hành Android

Hệ điều hành android có 4 tầng từ dưới lên trên là tầng hạt nhân Linux (Phiên bản 2.6), Tầng Libraries & Android runtime, Tầng Application Framework và trên cùng là tầng Application [2].



Hình 1.3. Kiến trúc của Hệ điều hành Android

### 1.2.1. Tầng hạt nhân Linux<sup>1</sup>

Hệ điều hành android được phát triển dựa trên hạt nhân linux, cụ thể là hạt nhân linux phiên bản 2.6, điều đó được thể hiện ở lớp dưới cùng này. Tất cả mọi hoạt động

<sup>1</sup> Linux Kernel layer

của điện thoại muốn thi hành được thì đều được thực hiện ở mức cấp thấp ở lớp này bao gồm quản lý bộ nhớ (memory management), giao tiếp với phần cứng (driver model), thực hiện bảo mật (security), quản lý tiến trình (process) [1][2].

Tuy được phát triển dựa vào nhân linux nhưng thực ra nhân linux đã được nâng cấp và sửa đổi rất nhiều để phù hợp với tính chất của những thiết bị cầm tay như hạn chế về bộ vi xử lý, dung lượng bộ nhớ, kích thước màn hình, nhu cầu kết nối mạng không dây...

- Tầng này có các thành phần chủ yếu :

- + Display Driver : Điều khiển việc hiển thị lên màn hình cũng như thu nhận những điều khiển của người dùng lên màn hình (di chuyển, cảm ứng...)
- + Camera Driver : Điều khiển hoạt động của camera, nhận luồng dữ liệu từ camera trả về.
- + Bluetooth Driver : Điều khiển thiết bị phát và thu sóng Bluetooth.
- + USB driver : Quản lý hoạt động của các cổng giao tiếp USB.
- + Keypad driver : Điều khiển bàn phím.
- + Wifi Driver : Chịu trách nhiệm về việc thu phát sóng wifi.
- + Audio Driver : điều khiển các bộ thu phát âm thanh, giải mã các tín hiệu dạng audio thành tín hiệu số và ngược lại .
- + Binder IPC Driver : Chịu trách nhiệm về việc kết nối và liên lạc với mạng vô tuyến như CDMA, GSM, 3G, 4G, E để đảm bảo những chức năng truyền thông được thực hiện.
- + M-System Driver : Quản lý việc đọc ghi... lên các thiết bị nhớ như thẻ SD, flash .
- + Power Madagement : Giám sát việc tiêu thụ điện năng.

### ***1.2.2. Tầng thư viện và thực thi Android<sup>2</sup>***

Phần này có 2 thành phần là phần Library và Android Runtime :

#### **a. Phần thư viện**

- Phần này có nhiều thư viện được viết bằng C/C++ để các phần mềm có thể sử dụng, các thư viện đó được tập hợp thành một số nhóm như :

- + Thư viện hệ thống (System C library) : thư viện dựa trên chuẩn C, được sử dụng chỉ bởi hệ điều hành.

---

<sup>2</sup> Library và android runtime

- ✚ Thư viện Media (Media Libraries) : Có nhiều codec để hỗ trợ việc phát và ghi các loại định dạng âm thanh, hình ảnh, video thông dụng.
- ✚ Thư viện web (LibWebCore) : Đây là thành phần để xem nội dung trên web, được sử dụng để xây dựng phần mềm duyệt web (Android Browse) cũng như để các ứng dụng khác có thể nhúng vào. Nó cực kỳ mạnh, hỗ trợ được nhiều công nghệ mạnh mẽ như HTML5, JavaScript, CSS, DOM, AJAX..
- ✚ Thư viện SQLite : Hệ cơ sở dữ liệu để các ứng dụng có thể sử dụng.
- ✚ ....

## **b. Phần thực thi Android**

Phần này chứa các thư viện mà một chương trình viết bằng ngôn ngữ Java có thể hoạt động. Phần này có 2 bộ phận tương tự như mô hình chạy Java trên máy tính thường. Thứ nhất là các thư viện lõi (Core Library) , chứa các lớp như JAVA IO, Collections, Tập Access. Thứ hai là một máy ảo java (Dalvik Virtual Machine) [1][2].

Mặc dù cũng được viết từ ngôn ngữ Java nhưng một ứng dụng Java của hệ điều hành android không được chạy bằng JRE của Sun (nay là Oracle) (JVM) mà là chạy bằng máy ảo Dalvik do Google phát triển.

### **1.2.3. Tầng khung ứng dụng<sup>3</sup>**

Tầng này xây dựng bộ công cụ - các phần tử ở mức cao để các lập trình viên có thể nhanh chóng xây dựng ứng dụng. Nó được viết bằng Java, có khả năng sử dụng chung để tiết kiệm tài nguyên [2].

- Đây là một nền tảng mở, điều đó có 2 điều lợi:

- ✚ Với các hãng sản xuất điện thoại : Có thể tùy biến để phù hợp với cấu hình điện thoại mà họ sản xuất cũng như để có nhiều mẫu mã, style hợp thị hiếu người dùng. Vì thế nên tuy cùng chung nền tảng android mà điện thoại của Google có thể khác hẳn với Motorola, HTC, T-Mobile, Samsung...
- ✚ Với lập trình viên : Cho phép lập trình viên có thể sử dụng các API ở tầng trên mà không cần phải hiểu rõ cấu trúc bên dưới, tạo điều kiện cho lập trình viên tự do sáng tạo bởi vì chỉ cần quan tâm đến nội dung mà ứng dụng họ làm việc. Một tập hợp API rất hữu ích được xây dựng sẵn như hệ thống định vị, các dịch vụ chạy nền, liên lạc giữa các ứng dụng, các thành phần giao diện cấp cao ...

Giới thiệu một số thành phần của phần này :

---

<sup>3</sup> *Application Framework*

- ✚ Activity Manager : Quản lý các chu kỳ sống của một ứng dụng cũng như cung cấp công cụ điều khiển các Activity.
- ✚ Telephony Manager : Cung cấp công cụ để thực hiện việc liên lạc như gọi điện thoại.
- ✚ XMPP Service : Cung cấp công cụ để liên lạc trong thời gian thực.
- ✚ Location Manager : Cho phép xác định vị trí của điện thoại dựa vào hệ thống định vị toàn cầu GPS và Google Maps.
- ✚ Window Manager : Quản lý việc xây dựng và hiển thị các giao diện người dùng cũng như tổ chức quản lý các giao diện giữa các ứng dụng.
- ✚ Notification Manager : Quản lý việc hiển thị các thông báo (như báo có tin nhắn, có e-mail mới..).
- ✚ Resource Manager : Quản lý tài nguyên tĩnh của các ứng dụng bao gồm các tệp hình ảnh, âm thanh, layout, string. (Những thành phần không được viết bởi ngôn ngữ lập trình)
- ✚ .....

#### 1.2.4. Tầng ứng dụng<sup>4</sup>

Đây là lớp ứng dụng giao tiếp với người dùng, bao gồm các ứng dụng như :

Các ứng dụng cơ bản, được cài đặt đi liền với hệ điều hành là gọi điện (phone), quản lý danh bạ (Contacts), duyệt web (Browser), nhắn tin (SMS), đọc e-mail (Email-Client), bản đồ (Map), quay phim chụp ảnh (camera)..[2].

Các ứng dụng được cài thêm như các phần mềm chứng khoán (Stock), các trò chơi (Game), từ điển... Các chương trình có các đặc điểm là :

- ✚ Viết bằng Java, phần mở rộng là apk.
- ✚ Khi mỗi ứng dụng được chạy, nó có một phiên bản Virtual Machine được dựng lên để phục vụ cho nó. Nó có thể là một Active Program : Chương trình có giao diện với người sử dụng hoặc là một background : chương trình chạy nền hay là dịch vụ.
- ✚ Android là hệ điều hành đa nhiệm, điều đó có nghĩa là trong cùng một thời điểm, có thể có nhiều chương trình cùng chạy một lúc, tuy nhiên, với mỗi ứng dụng thì có duy nhất một thực thể (instance) được phép chạy mà thôi.

---

<sup>4</sup> Application

Điều đó có tác dụng hạn chế sự lạm dụng tài nguyên, giúp hệ thống hoạt động tốt hơn.

- ✚ Các ứng dụng được gán số ID của người sử dụng nhằm phân định quyền hạn khi sử dụng tài nguyên, cấu hình phần cứng và hệ thống.
- ✚ Android là một hệ điều hành có tính mở, khác với nhiều hệ điều hành di động khác, android cho phép một ứng dụng của bên thứ ba được phép chạy nền. Các ứng dụng đó chỉ có một hạn chế nhỏ đó là nó không được phép sử dụng quá 5~10% công suất CPU, điều đó nhằm để tránh độc quyền trong việc sử dụng CPU.
- ✚ Ứng dụng không có điếm vào cố định, không có phương thức main để bắt đầu.

### 1.3. Mô tả hệ điều hành Android

#### 1.3.1. Giao diện

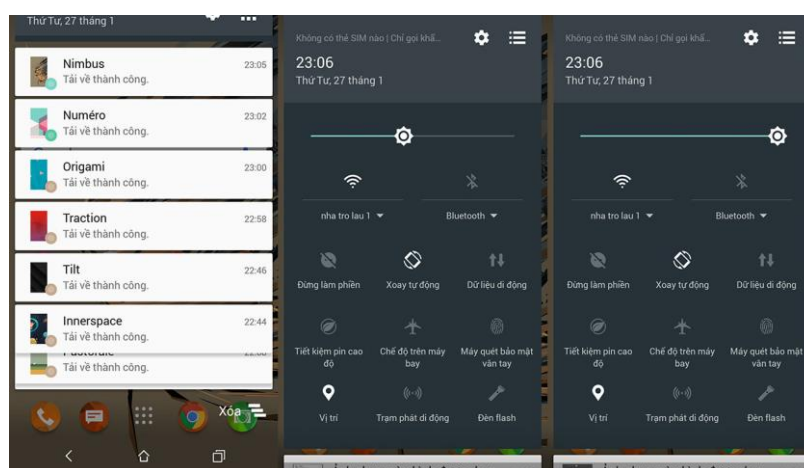
Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo giãn và thu lại để xử lý các đối tượng trên màn hình. Sự phản ứng với tác động của người dùng diễn ra gần như ngay lập tức, nhằm tạo ra giao diện cảm ứng mượt mà, thường dùng tính năng rung của thiết bị để tạo phản hồi rung cho người dùng [1]. 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.



Hình 1.4. Giao diện Android 6.0.1 Marshmallow trên điện thoại LG Nexus 5X

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[1].

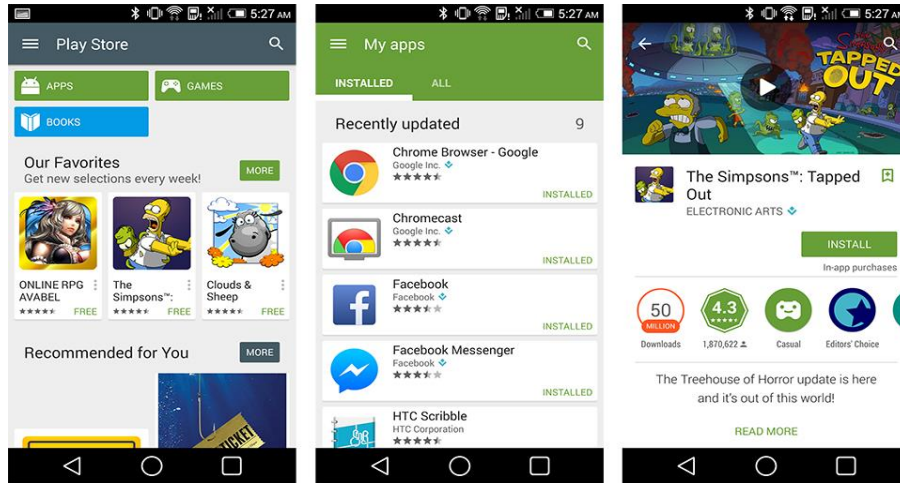
Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Thanh trạng thái này có thể "kéo" xuống để xem màn hình thông báo gồm thông tin quan trọng hoặc cập nhật của các ứng dụng, như email hay tin nhắn SMS mới nhận, mà không làm gián đoạn hoặc khiến người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



Hình 1.5. Thanh trạng thái trên Android 6.0



### 1.3.2. Ứng dụng



Hình 1.6. Kho ứng dụng CHPlay của 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 [1].

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.

Để vượt qua những hạn chế khi tiếp cận các dịch vụ của Google do sự Kiểm duyệt Internet tại Cộng hòa Nhân dân Trung Hoa, các thiết bị Android bán tại Trung Quốc lục địa thường được điều chỉnh chỉ được sử dụng dịch vụ đã được duyệt.

## 1.4. Sự đón nhận của người dùng đối với hệ điều hành Android

Android được đón nhận bằng một thái độ thờ ơ khi nó 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 dấy 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." [1].

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.

### ***\*Thị phần và tỷ lệ sử dụng***

Công ty nghiên cứu thị trường Canalsys đã ước tính trong quý 2 năm 2009 rằng Android có 2,8% thị phần điện thoại thông minh được bán ra toàn cầu. Đến quý 4 năm

2010 con số này tăng lên 33% thị phần, trở thành nền tảng điện thoại thông minh bán chạy hàng đầu. Đến quý 3 năm 2011 Gartner ước tính rằng hơn một nửa (52,5%) thị trường điện thoại thông minh thuộc về Android. Đến quý 3 năm 2012 Android đã có 75% thị phần điện thoại thông minh toàn cầu theo nghiên cứu của hãng IDC [1].

Vào tháng 7 năm 2011, Google nói rằng có 550.000 thiết bị Android mới được kích hoạt mỗi ngày, đỉnh điểm là 400.000 máy một ngày vào tháng 5, và có hơn 100 triệu thiết bị đã được kích hoạt với mức tăng 4,4% mỗi tuần. Vào tháng 9 năm 2012, 500 triệu thiết bị đã được kích hoạt với 1,3 triệu lượt kích hoạt mỗi ngày.

Thị phần của Android có khác nhau theo khu vực. Vào tháng 7 năm 2012, thị phần Android tại Mỹ là 52%, nhưng lên tới 90% tại Trung Quốc.

**\*Tỷ lệ sử dụng các phiên bản Android**

Tỷ lệ sử dụng các phiên bản khác nhau tính đến tháng 4 năm 2014. Phần lớn các thiết bị Android cho tới nay vẫn chạy hệ điều hành phiên bản 4.1.x Jelly Bean được phát hành ngày 9 tháng 7 năm 2012 nhờ tính ổn định và hỗ trợ tốt các máy có cấu hình thấp [1].

Phiên bản ↕	Tên mã ↕	Ngày phát hành ↕	Cấp API ↕	Phân bố (20 tháng 7 năm 2014) ▾
5.0	Lollipop	tháng 7 năm 2014	20	Dành cho người phát hành
4.1.x	Jelly Bean	9 tháng 7 năm 2012	16	25,2%
4.2.x	Jelly Bean	13 tháng 11 năm 2012	17	18,8%
4.4	KitKat	tháng 10 năm 2013	19	17,9%
2.3.3-2.3.7	Gingerbread	9 tháng 2 năm 2011	10	13%
4.0.x	Ice Cream Sandwich	16 tháng 12 năm 2011	15	11,4%
4.3	Jelly Bean	25 tháng 7 năm 2013	18	10,5%
2.2	Froyo	20 tháng 5 năm 2010	8	0,7%
2.3-2.3.2	Gingerbread	6 tháng 12 năm 2010	9	0,5%
3.2	Honeycomb	15 tháng 7 năm 2011	13	0%
3.1	Honeycomb	10 tháng 5 năm 2011	12	0%
2.0-2.1	Eclair	26 tháng 10 năm 2009	7	0%
1.6	Donut	15 tháng 9 năm 2009	4	0%

Hình 1.7. Tỷ lệ sử dụng các phiên bản Android

## CHƯƠNG 2: MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG ANDROID

### 2.1. Môi trường phát triển

#### 2.1.1. Phần mềm Android Studio

Android Studio là môi trường phát triển tích hợp (IDE) chính thức dành cho phát triển các ứng dụng chạy trên nền tảng Android[1].

Nó được ra mắt vào ngày 16 tháng 5 năm 2013 tại hội nghị Google I/O. Android Studio được phát hành miễn phí theo giấy phép Apache Licence 2.0.



Hình 2.1. Giao diện phần mềm Android Studio

Android Studio hoạt động dựa trên “IntelliJ IDEA Community Edition” (giống như kiểu WYSIWYG – What You See is What You Get) cho phép lập trình viên tạo ứng dụng, dễ dàng thực hiện các thay đổi và xem trước trong thời gian thực, đồng thời cũng có khả năng tăng tốc sản phẩm, thiết kế giao diện đẹp hơn trước. Đặc biệt là tiếng Việt cũng được hỗ trợ trong Android Studio[3].

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

Android Studio hỗ trợ các hệ điều hành Windows, Mac OS X và Linux, và là IDE chính thức của Google để phát triển ứng dụng Android gốc để thay thế cho Android Development Tools (ADT) dựa trên Eclipse[3].

#### 2.1.2. Bộ công cụ phát triển Java SE Development Kit (JDK)

JDK là một bộ công cụ phát triển Java, nó dành cho những người lập trình Java để phát triển ứng dụng. Về cơ bản nó bao gồm:

- ✚ JRE (Java Runtime Environment) là một môi trường chạy ứng dụng Java.

- ✚ Javac: Một chương trình để dịch mã mà bạn viết thành mã bytecode, khi ứng dụng Java chạy nó dịch mã bytecode thành mã máy tính và thực thi, điều đó có nghĩa là bytecode chỉ là một mã trung gian.
- ✚ Archive (jar): Là một chương trình nén các tệp thành một tệp duy nhất có đuôi jar. Thường dùng để đóng gói các tệp class.
- ✚ Javadoc: Là một công cụ tạo ra tài liệu hướng dẫn sử dụng API.
- ✚ Và các công cụ khác cần thiết cho phát triển Java[1].

### **2.1.3. Máy ảo Android Virtual Device (AVD)**

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.

AVD là một máy ảo Android được hỗ trợ chính thức từ Google. Vì là bản “chính chủ” nên máy ảo này sẽ có tính ổn định cao. Chẳng hạn như nó sẽ tiêu tốn bộ nhớ của máy tính ít hơn các máy ảo khác, nó còn hỗ trợ giả lập tất cả các loại thiết bị, từ điện thoại, máy tính bảng, thiết bị đeo được, và kể cả Android TV. Nhược điểm của máy ảo này là khá ít, khi mà mới đây AVD có hỗ trợ ứng dụng Google Play, giúp ta có thể cài đặt các ứng dụng có trên store về để sử dụng[4].

## **2.2. Các thành phần trong một dự án Android**

### **2.2.1. Tệp cấu hình dự án AndroidManifest.xml**

Khi một dự án Android được tạo ra đều có một tệp AndroidManifest.xml, tệp này được dùng để định nghĩa các màn hình (screen) sử dụng, các quyền (permission) cũng như các chủ đề (theme) cho ứng dụng. Đồng thời nó cũng chứa thông tin về phiên bản SDK cũng như main activity sẽ chạy đầu tiên, tệp tin này được tự động sinh ra khi tạo một dự án Android. Trong tệp manifest bao giờ cũng có 3 thành phần chính đó là: thành phần ứng dụng (application), các quyền (permission) và các phiên bản (version)[1][5].

#### **a. Thành phần ứng dụng**

Thẻ<application>, bên trong thẻ này chứa các thuộc tính được định nghĩa cho ứng dụng Android như:

- android:icon = “drawable resource” → Ở đây đặt đường dẫn đến tệp icon của ứng dụng khi cài đặt. VD: android:icon = “@drawable/icon”.

- android:name = “string” → thuộc tính này để đặt tên cho ứng dụng Android. Tên này sẽ được hiển thị lên màn hình sau khi cài đặt ứng dụng.

- android:theme = “drawable theme” → thuộc tính này để đặt theme cho ứng dụng. Các theme là các cách để hiển thị giao diện ứng dụng. Ngoài ra còn nhiều thuộc tính khác...

### **b. Quyền truy xuất và sử dụng tài nguyên**

Bao gồm các thuộc tính chỉ định quyền truy xuất và sử dụng tài nguyên của ứng dụng. Khi cần sử dụng một loại tài nguyên nào đó thì trong tệp manifest của ứng dụng cần phải khai báo các quyền truy xuất như sau:

```
<uses-permission  
android:name="android.permission.READ_PHONE_STATE"/>  
<uses-permission  
android:name="android.permission.ACCOUNT_MANAGER"/>  
<uses-permission android:name="android.permission.VIBRATE" />  
<uses-permission android:name="android.permission.CALL_PHONE"/>
```

### **c. Phiên bản SDK**

Thẻ xác định phiên bản SDK được khai báo như sau:

```
<uses-sdk android:minSdkVersion="7" />
```

Ở đây chỉ ra phiên bản SDK thấp nhất mà ứng dụng hiện đang sử dụng.

#### **2.2.2. Tệp R.java**

Tệp R.java là một tệp tự động sinh ra ngay khi tạo ứng dụng, tệp này chứa các thuộc tính được khai báo trong tệp XML như : id, layout, menu, class... của ứng dụng và các tài nguyên hình ảnh[1][5].

Mã nguồn của tệp R.java được tự động sinh khi có bất kì một sự kiện nào xảy ra làm thay đổi các thuộc tính trong ứng dụng. Chẳng hạn như, bạn kéo và thả một tệp hình ảnh từ bên ngoài vào dự án thì ngay lập tức thuộc tính đường dẫn đến tệp đó cũng sẽ được hình thành trong tệp R.java hoặc xoá một tệp hình ảnh thì đường dẫn tương ứng đến hình ảnh đó cũng tự động bị xoá.

### **2.3. Vòng đời ứng dụng Android**

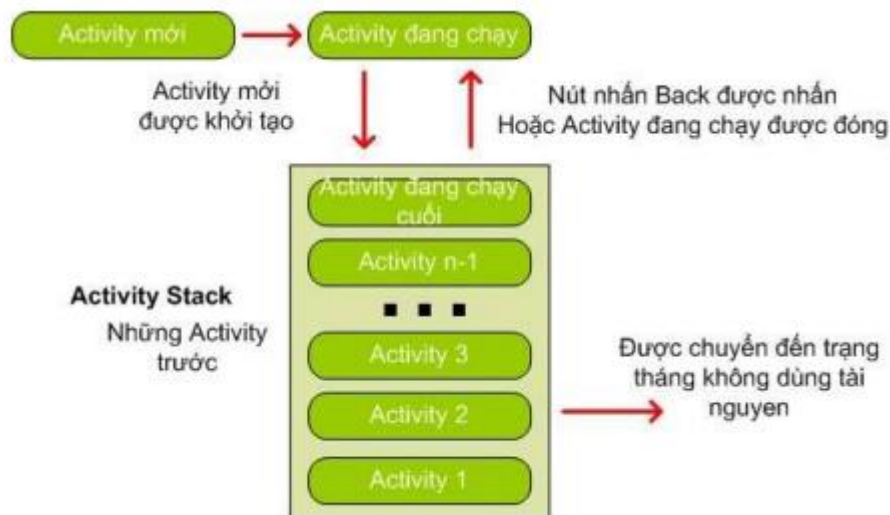
#### **2.3.1. Chu kỳ sống của các thành phần**

Các thành phần ứng dụng có một chu kỳ sống, tức là mỗi thành phần từ khi bắt đầu được tạo ra cho đến khi kết thúc, sẽ có những trạng thái như đang hoạt động (đang hiển thị hoặc đang ẩn) hoặc không hoạt động.

### 2.3.2. Activity Stack

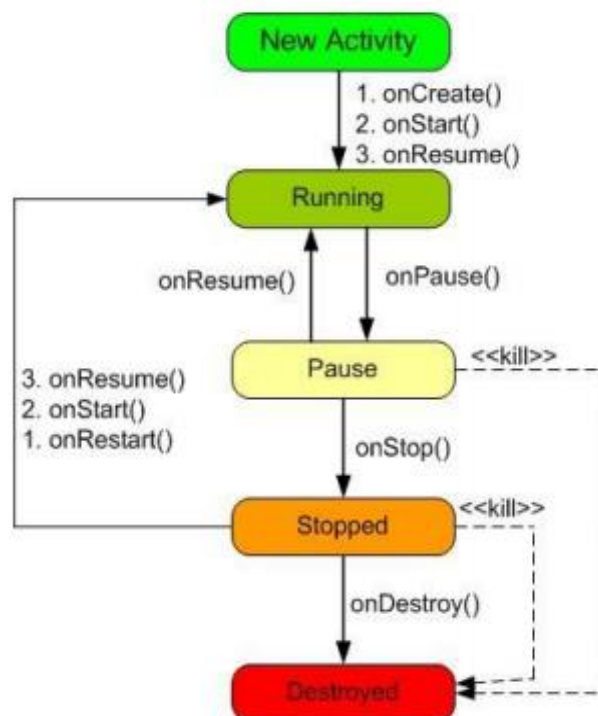
Bên trong hệ thống các activity được quản lý như một activity stack. Khi một Activity mới được tạo, nó được đặt ở đỉnh của stack và trở thành activity đang chạy, các activity trước sẽ tạm dừng và ở bên dưới activity mới và sẽ không thấy trong suốt quá trình activity mới tồn tại.

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



Hình 2.2. Activity Stack

### 2.3.3. Các trạng thái của chu kỳ sống



Hình 2.3. Chu kỳ sống của Activity

Một Activity chủ yếu có 4 chu kỳ chính sau:

- ✚ **active (running):** Activity đang hiển thị trên màn hình, những thao tác của người dùng sẽ được tập trung vào Activity này.
- ✚ **paused:** Activity được tạm dừng, tuy vẫn hiển thị (visible) nhưng không thể tương tác (lost focus). Có nghĩa là một activity mới ở trên nó, nhưng không bao phủ đầy màn hình. Một activity tạm dừng là còn sống, nhưng có thể bị kết thúc bởi hệ thống trong trường hợp thiếu vùng nhớ.
- ✚ **stop:** Activity bị thay thế hoàn toàn bởi Activity mới sẽ tiến đến trạng thái stop. Nó vẫn còn trạng thái và thông tin thành viên trong nó. Người dùng không thấy nó và thường bị loại bỏ khi hệ thống cần vùng nhớ cho các tác vụ khác.
- ✚ **killed:** Khi hệ thống bị thiếu bộ nhớ, nó sẽ giải phóng các tiến trình theo nguyên tắc ưu tiên. Các Activity ở trạng thái stop hoặc paused cũng có thể bị giải phóng và khi nó được hiển thị lại thì các Activity này phải khởi động lại hoàn toàn và phục hồi lại trạng thái trước đó.

#### 2.3.4. Thời gian sống của Activity

- ✚ **Entire lifetime:** Từ phương thức onCreate() cho tới onDestroy().
- ✚ **Visible lifetime:** Từ phương thức onStart() cho tới onStop().
- ✚ **Foreground lifetime:** Từ phương thức onResume() cho tới onPause().

Khi xây dựng Activity cho ứng dụng cần phải viết lại phương thức onCreate() để thực hiện quá trình khởi tạo. Các phương thức khác có cần viết lại hay không tùy vào yêu cầu lập trình.

#### 2.3.5. Các phương thức của chu kỳ sống

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



- ✚ **onRestart():** được gọi khi ứng dụng chuyển sang onStop(), nhưng muốn khởi động lại bằng onStart().
- ✚ **onDestroy():** được gọi khi chúng ta nhấn back từ activity, hoặc call method finish() của activity.

## 2.4. Các thành phần giao diện trong Android

### 2.4.1. View và ViewGroup

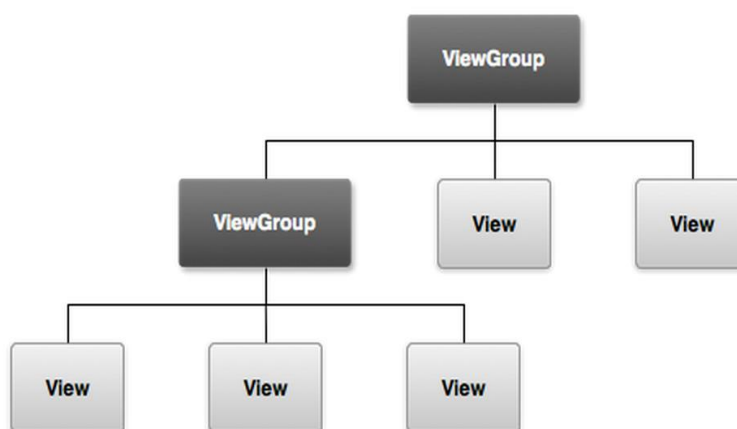
Những gì chúng ta nhìn thấy trên màn hình thiết bị android được gọi là View (trong window thường được gọi là control). View được vẽ trên thiết bị android với một hình chữ nhật[5].

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

Tất cả mọi widget đều có chung các thuộc tính cơ bản như là cách trình bày vị trí, background, kích thước, lề,... Tất cả những thuộc tính chung này được thể hiện hết ở trong đối tượng View.

Trong Android Platform, các màn hình luôn được bố trí theo một kiểu cấu trúc phân cấp như hình dưới. Một màn hình là một tập hợp các Layout và các widget được bố trí có thứ tự. Để thể hiện một màn hình thì trong hàm onCreate của mỗi activity cần phải được gọi một hàm setContentView(R.layout.main); hàm này sẽ load giao diện từ tệp XML lên để phân tích thành mã bytecode.

ViewGroup thực ra chính là View hay nói đúng hơn thì ViewGroup chính là các widget Layout được dùng để bố trí các đối tượng khác trong một màn hình.



Hình 2.4. Cấu trúc một giao diện ứng dụng android

## *Một số view cơ bản và thường xuyên sử dụng trong Android :*

### **a. TextView**

TextView là view sử dụng để hiển thị text lên màn hình và không cho phép chỉnh sửa khi dự án đã được khởi chạy. TextView được định nghĩa bởi thẻ <TextView> trong xml. Tất cả các view được khởi tạo bằng xml bắt buộc phải có hai thuộc tính đó là layout\_width và layout\_height[5].

```
<TextView
    android:id="@+id/textview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:textSize="20sp"
    android:textColor="#000000"
    android:text="Hello Android!"/>
```

Một số thuộc tính của TextView:

android:id : Xác định id cho TextView

android:text: Xác định text hiển thị lên TextView

android:textColor: Xác định màu của text.

android:textSize: Xác định kích thước của text.

android:textStyle: Xác định style của text gồm các giá trị italic (nghiêng), bold (in đậm), normal (kiểu thường).

### **b. EditText**

EditText là view dùng để lấy giá trị từ người dùng nhập vào. EditText được định nghĩa bởi thẻ <EditText> trong xml. Một số thuộc tính thuộc tính thường dùng của lớp EditText :

```
<EditText
    android:id="@+id/editText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:textSize="20sp"
    android:textColor="#000000"
    android:text="Hello Android!"/>
```

Một số thuộc tính của EditText:

android:text: Xác định text hiển thị lên EditText

android:textColor: Xác định màu của text.

android:textSize: Xác định kích thước của text.

android:textStyle: Xác định style của text gồm các giá trị italic (ngiêng), bold (in đậm), normal (kiểu thường).

Để lấy text của EditText trong java chúng ta làm như sau:

Bước 1: Lấy về EditText thông qua id trong file xml.

```
EditText edt;
```

```
edt = (EditText) findViewById(R.id.edittext);
```

Bước 2: Sử dụng phương thức getText() của EditText để lấy text.

```
String value = edt.getText().toString();
```

### c. Button

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.

Ví dụ về Button khai báo trong file xml.

```
<Button  
    android:id="@+id/btn_Click"  
    android:text="Click Me"  
    android:textStyle="italic"  
    android:textColor="@color/colorAccent"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

Một số thuộc tính của Button

android:id: Xác định id cho Button

android:text: Xác định văn bản sẽ hiển thị lên Button

android:textColor: Xác định màu văn bản của Button.

android:background: Xác định màu nền của Button

Để thực hiện những khối lệnh khi chúng ta click vào button, chúng ta làm như sau:

Bước 1: Lấy về đối tượng Button thông qua id được khai báo button ở file xml

```
Button btnClick = (Button)findViewById(R.id.btn_click);
```

Bước 2: Set bộ lắng nghe sự kiện cho Button thông qua phương thức setOnClickListener và handle các công việc cần xử lý trong phương thức onClick :

```
btnClick.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {}  
});
```

#### d. ImageView

ImageView là một view sử dụng rất nhiều trong ứng dụng android, ImageView sử dụng để hiển thị hình ảnh.

Ví dụ khai báo ImageView trong file xml :

```
<ImageView
    android:scaleType="centerCrop"
    android:src="@drawable/keep_calm"
    android:layout_gravity="center_horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

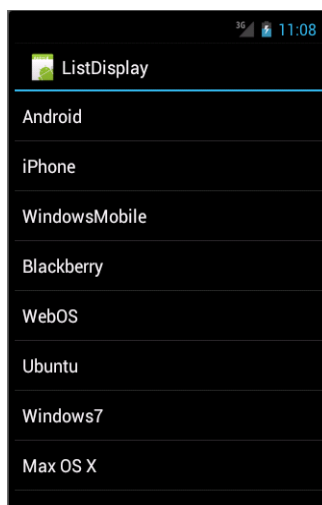
Một số thuộc tính phổ biến của ImageView

android:id: Xác định id.

android:src: Xác định source hình ảnh hoặc drawable.

android:scaleType: kiểu hiển thị hình ảnh của ImageView, có các giá trị fitCenter, fitStart, fitEnd, center, centerCrop, centerInside, matrix.

#### e. ListView



Hình 2.5. Minh họa cho một ListView

Được sử dụng để thể hiện một danh sách các thông tin theo từng dòng. Mỗi dòng thông thường được load lên từ một tệp XML đã được cố định trên đó số lượng thông tin và loại thông tin cần được thể hiện. Để thể hiện được một list thông tin lên một màn hình thì cần phải có 4 thành phần:

- Data Source: Data Source có thể là một ArrayList, HashMap hoặc bất kỳ một cấu trúc dữ liệu kiểu danh sách nào.
- Adapter: Adapter là một class trung gian giúp ánh xạ dữ liệu trong Data Source vào đúng vị trí hiển thị trong ListView. Chẳng hạn, trong Data Source có một trường name và trong ListView cũng có một TextView để thể hiện trường name

này. Tuy nhiên, ListView sẽ không thể hiển thị dữ liệu trong Data Source lên được nếu như Adapter không gán dữ liệu vào cho đối tượng hiển thị.

- Bản vẽ ( Layout) xml : Nơi thiết kế các cách hiển thị của ListView khi hiển thị lên màn hình.

- ListView: ListView là đối tượng để hiển thị các thông tin trong Data Source ra một cách trực quan và người dùng có thể thao tác trực tiếp trên đó.

## 2.4.2 Layout

Layout là một lớp điều khiển việc sắp xếp các thành phần con của nó xuất hiện trên màn hình. Bất cứ thành phần nào đó là một View (hoặc thừa kế từ View) đều có thể là con của một Layout. Tất cả các lớp Layout đều mở rộng từ lớp ViewGroup (mà kế thừa từ View), do đó bạn cũng có thể tạo một lớp Layout tùy biến của mình, bằng cách tạo một lớp mở rộng từ ViewGroup. Một số các loại Layout tiêu chuẩn trong Android :

### a. RelativeLayout

Layout này cho phép bố trí các widget theo một trục đối xứng ngang hoặc dọc. Để đặt được đúng vị trí thì các widget cần được xác định một mối ràng buộc nào đó với các widget khác. Các ràng buộc này là các ràng buộc trái, phải, trên, dưới so với một widget hoặc so với layout cha. Dựa vào những mối ràng buộc đó mà RelativeLayout cũng không phụ thuộc vào kích thước của màn hình thiết bị.[5]

Ngoài ra, nó còn có ưu điểm là giúp tiết kiệm layout sử dụng nhằm mục đích giảm lượng tài nguyên sử dụng khi load đồng thời đẩy nhanh quá trình xử lý.



Hình 2.6. Bố trí widget trong RelativeLayout

## b. TableLayout

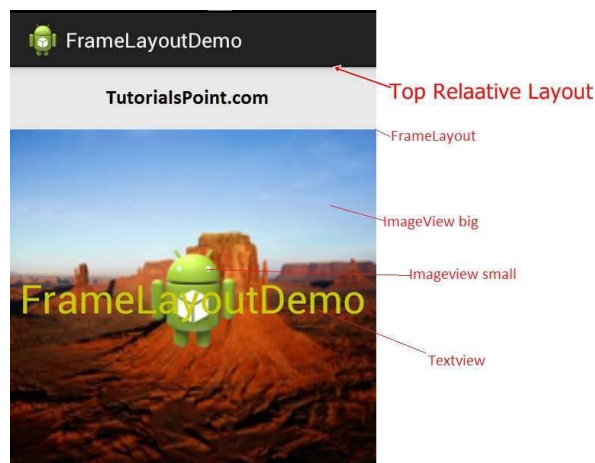
Layout này được sử dụng khi cần thiết kể một bảng chứa dữ liệu hoặc cần bố trí các widget theo các hàng và cột. Chẳng hạn như, giao diện của một chiếc máy tính đơn giản hoặc một danh sách dữ liệu.



Hình 2.7. Bố trí widget trong TableLayout

## c. FrameLayout

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

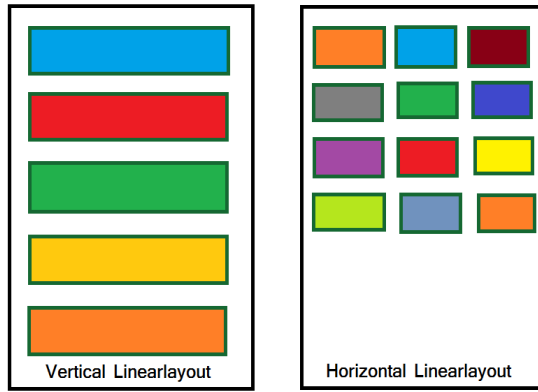


Hình 2.8. Bố trí widget trong FrameLayout

## d. LinearLayout

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

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



Hình 2.9. Bố trí các widget trong LinearLayout

## CHƯƠNG 3: CƠ SỞ DỮ LIỆU WEB VÀ DỊCH VỤ WEB

### 3.1. Mạng Internet

#### 3.1.1. Khái niệm

Internet là một mạng máy tính toàn cầu có thể được truy nhập công cộng gồm các mạng máy tính được liên kết với nhau. Hệ thống này truyền thông tin theo kiểu nối chuyển gói dữ liệu (packet switching) dựa trên một giao thức liên mạng đã được chuẩn hóa (giao thức IP). Hệ thống này bao gồm hàng ngàn mạng máy tính nhỏ hơn của các doanh nghiệp, của các viện nghiên cứu và các trường đại học, của người dùng cá nhân và các chính phủ trên toàn cầu.

#### 3.1.2. Lợi ích của Internet trong cuộc sống

Mạng Internet mang lại rất nhiều tiện ích hữu dụng cho người sử dụng, một trong các tiện ích phổ thông của Internet là hệ thống thư điện tử (email), trò chuyện trực tuyến (chat), công cụ tìm kiếm (search engine), các dịch vụ thương mại và chuyển ngân và các dịch vụ về y tế giáo dục như là chữa bệnh từ xa hoặc tổ chức các lớp học ảo. Chúng cung cấp một khối lượng thông tin và dịch vụ khổng lồ trên Internet.

Nguồn thông tin khổng lồ kèm theo các dịch vụ tương ứng chính là hệ thống các trang Web liên kết với nhau và các tài liệu khác trong WWW (World Wide Web). Trái với một số cách sử dụng thường ngày, Internet và WWW không đồng nghĩa. Internet là một tập hợp các mạng máy tính kết nối với nhau bằng dây đồng, cáp quang, v.v.; còn WWW, hay Web, là một tập hợp các tài liệu liên kết với nhau bằng các siêu liên kết (hyperlink) và các địa chỉ URL và nó có thể được truy nhập bằng cách sử dụng Internet. Tuy nhiên việc này không có gì khó hiểu bởi vì Web là môi trường giao tiếp chính của người sử dụng trên Internet. Đặc biệt trong thập kỷ đầu của thế kỷ 21 nhờ sự phát triển của các trình duyệt web và hệ quản trị nội dung nguồn mở đã khiến cho website trở nên phổ biến hơn, thế hệ web 2.0 cũng góp phần đẩy cuộc cách mạng web lên cao trào, biến web trở thành một dạng phần mềm trực tuyến hay phần mềm như một dịch vụ.

#### 3.1.3. Các chương trình duyệt Web thông dụng hiện nay

- ✚ Internet Explorer, Microsoft Edge có sẵn trong Microsoft Windows, của Microsoft.
- ✚ Mozilla Firefox của Tập đoàn Mozilla.
- ✚ Google Chrome của Google.
- ✚ Netscape Navigator của Netscape.



- ✚ Opera của Opera Software.
- ✚ Safari trong Mac OS X, macOS, iOS của Apple Computer.
- ✚ Maxthon của MySoft Technology.
- ✚ Avant Browser của Avant Force (Ý).

Cách thức thông thường để truy cập Internet là không dây, vệ tinh và qua điện thoại cầm tay, iPad, iPhone, Samsung Galaxy và một số dòng điện thoại cảm ứng khác.

## 3.2. Web Hosting

### 3.2.1. Khái niệm

Web Hosting là không gian trên máy chủ mà ở đó có cài đặt các dịch vụ Internet như world wide web (www), truyền file (FTP), Mail... , Web hosting thường được dùng để chứa nội dung trang web hay dữ liệu...

Web Hosting đồng thời cũng là nơi diễn ra tất cả các hoạt động giao dịch, trao đổi thông tin giữa người sử dụng Internet và Website.

Mỗi website đều có một host, khi truy cập vào website đó, thực ra là chúng ta đang truy cập vào host chứa website.

Ví dụ: Bình thường khi chúng ta có 1 file trong máy tính, ở dạng Localhost, và bây giờ chúng ta muốn cho người khác xem thì chúng ta cần tải file đó lên mạng, nơi để lưu trữ file đó gọi là hosting.



Hình 3.1. Web Hosting là gì?

### 3.2.2. Các loại hosting

- ✚ Shared hosting: Chia sẻ host
- ✚ Collocated hosting: Thuê chỗ đặt máy chủ
- ✚ Dedicated Server: Máy chủ dùng riêng
- ✚ Virtual Private Server: VPS là máy chủ riêng ảo

### 3.2.3. Các thông số cần biết trong web 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 bạn có thể trỏ 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.

### 3.2.4. Tại sao cần phải mua web hosting ?

Nếu không có hosting thì website sẽ chỉ hoạt động trên máy tính và ở dạng Localhost, dữ liệu sẽ không được chia sẻ trên mạng. Vậy nên rất cần thiết để có một gói hosting.

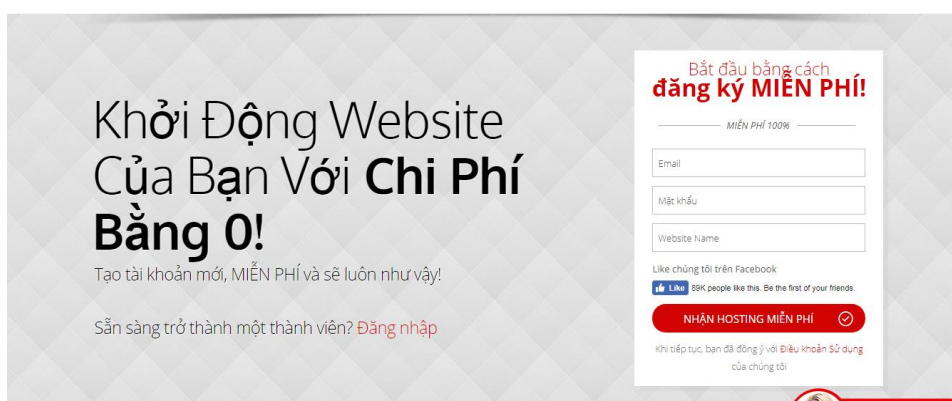
### 3.2.5. Cách đăng kí một web hosting miễn phí

Bước 1: Truy cập website : <https://vn.000webhost.com> .Sau đó chọn **Đăng kí miễn phí**.



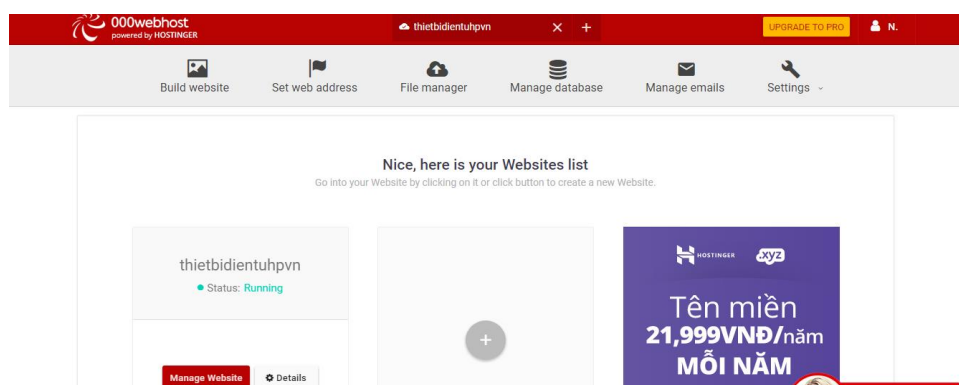
Hình 3.2. Giao diện trang chủ website

Bước 2 : Tiếp theo, bạn điền vào form đăng ký. Nhập địa chỉ email, password và tên website. Chọn nút **Nhận Hosting miễn phí**.



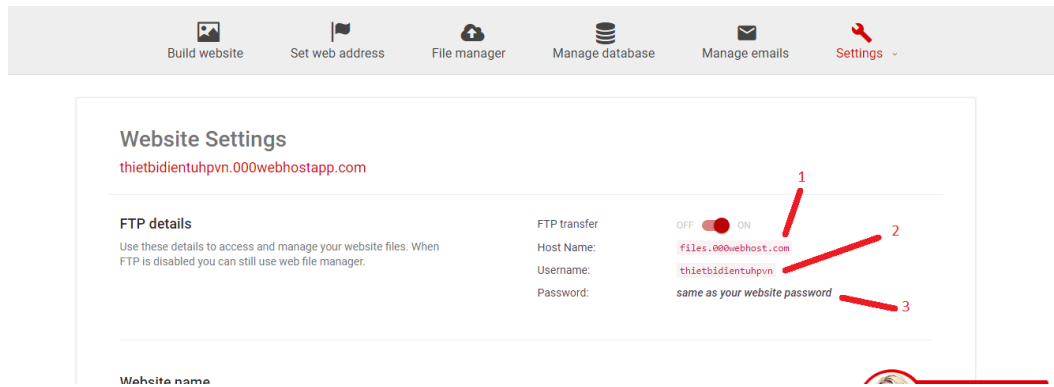
Hình 3.3. Giao diện đăng kí tài khoản

Sau đó, bạn vào hòm mail và thực hiện xác nhận email. Khi xác nhận mail xong website sẽ hiện ra giao diện như sau:



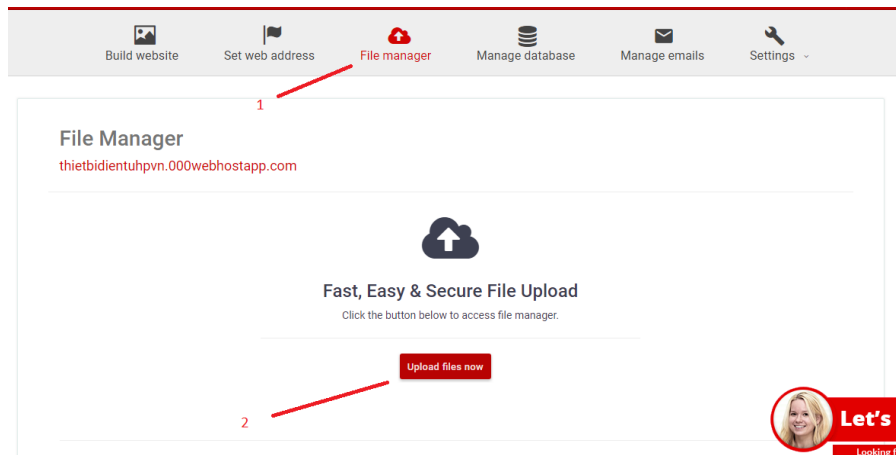
Hình 3.4. Giao diện website khi đăng kí thành công

Bước 3 : Về cách lấy file source code bạn sử dụng FTP Client như **FileZilla** để truy cập server của 000webhost hoặc thao tác trực tiếp trên web service. Thông tin kết nối bạn vào **Settings -> General**.



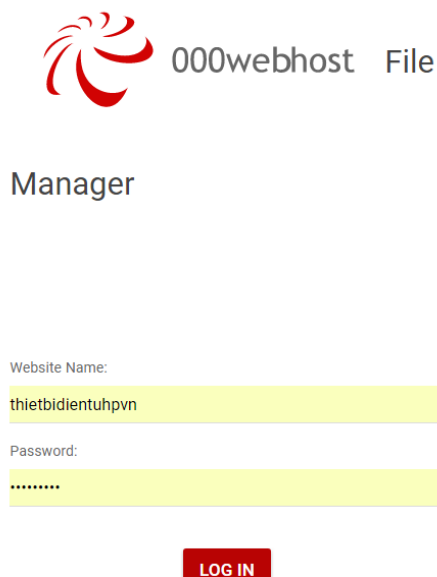
Hình 3.5. Các thông tin để kết nối với Filezilla

Bước 4: Cách up và quản lí các file đuôi .php để truy xuất và kết nối cơ sở dữ liệu lên Web service ta làm như sau. Đầu tiên ta chọn **File manager** -> **Upload file now** như hình:



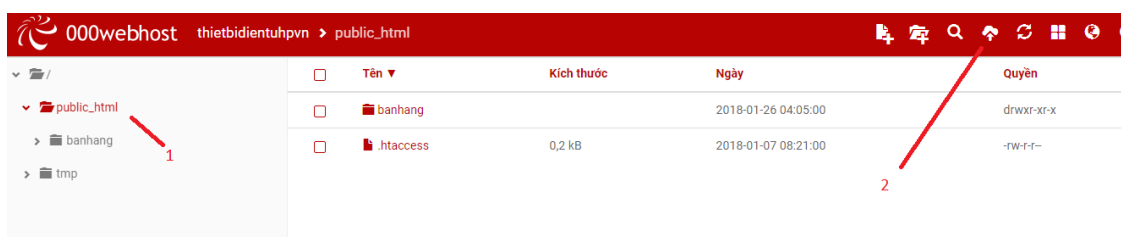
Hình 3.6. Cách up file lên web service

Bước 5: Tại màn hình đăng nhập, ta sẽ sử dụng Username và password như ở trong mục **Setting**(hình 3.5).



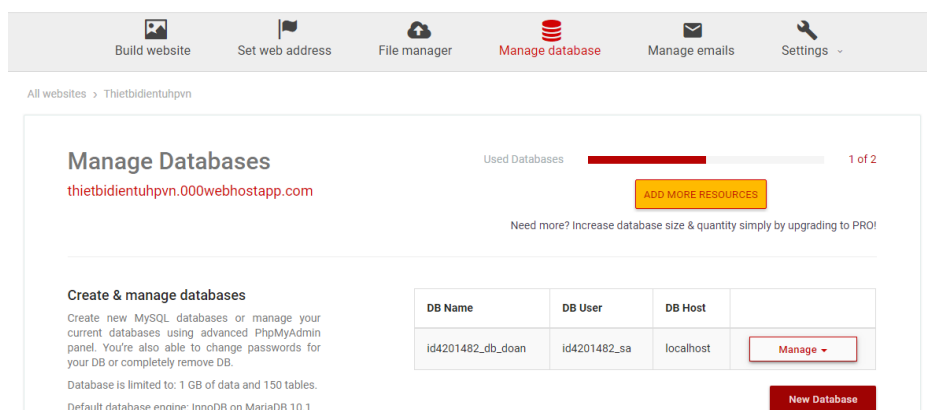
Hình 3.7. Màn hình đăng nhập của File manager

Trong thư mục **public\_html**, ta có thể upload những file,tài liệu,hình ảnh... lên web service bằng cách chọn **Upload files**.



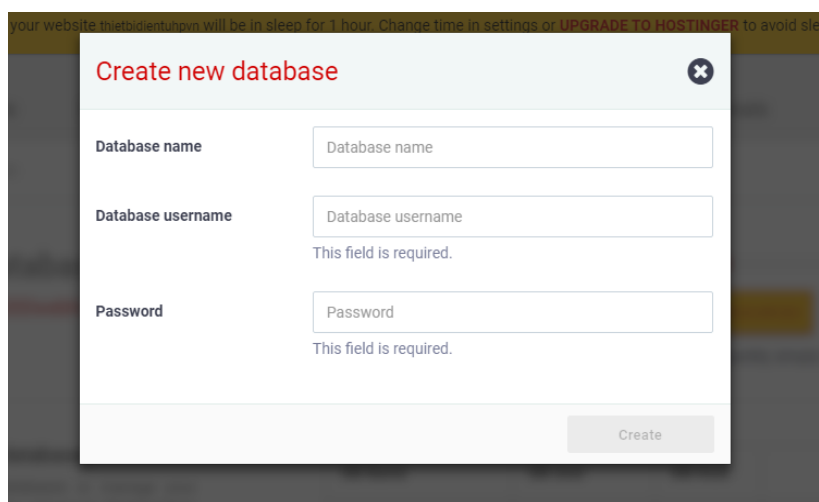
Hình 3.8. Cách up files lên web service

Bước 6 : Về lấy cơ sở dữ liệu bạn vào **Manage Database**. Sau đó ấn **New Database**.



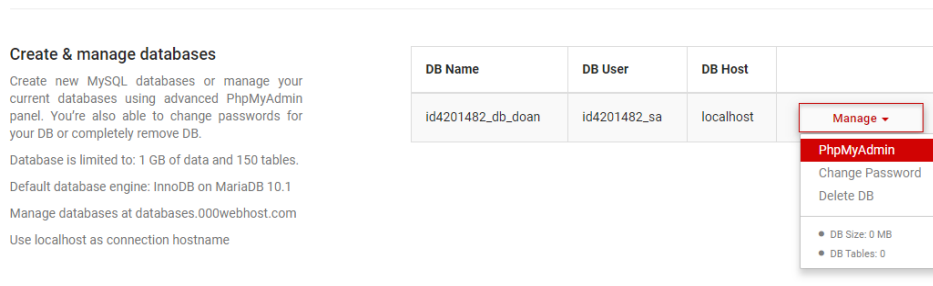
Hình 3.9. Cách tạo một Database

Tại cửa sổ **Create new database** điền name của database,username để đăng nhập vào database và password, sau đó chọn **Create** để tạo.



Hình 3.10. Cách tạo một Database

Sau khi tạo xong sẽ có 1 database như hình gồm có đầy đủ những thông số như DB name, DB user và DB Host. Để truy cập vào PhpMyAdmin để quản trị cơ sở dữ liệu ta chọn **Manage -> PhpMyAdmin**.

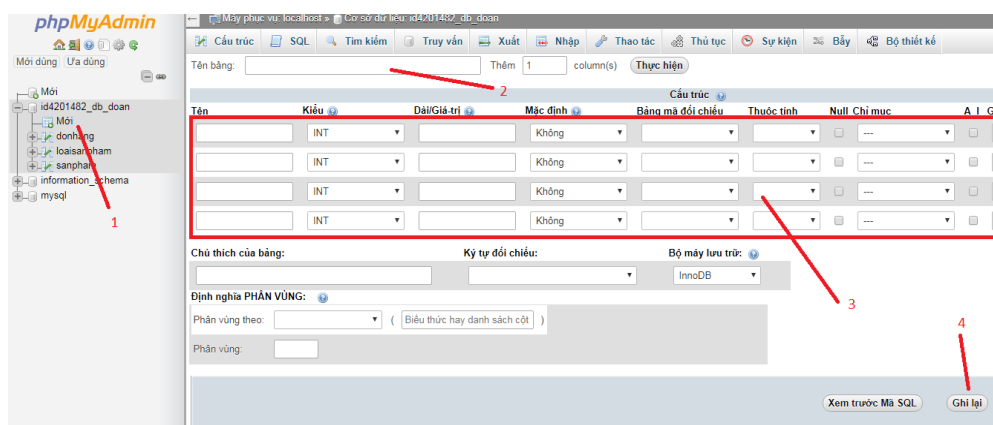


Hình 3.11. Cách truy cập vào phpMyAdmin

Bước 7 : Khi đã truy cập vào PhpMyAdmin, chúng ta sẽ có 1 database vừa khởi tạo ở bước trên với tên là **id420182\_db\_doan**. Tại đây ta có 2 cách để thêm các bảng vào trong database :

Cách 1 là khi ta đã có 1 database ở dạng localhost rồi thì ta chỉ cần xuất những bảng đó ra và nhập vào trong database mới.

Cách 2 đó là tạo các bảng mới bằng cách chọn nút **mới**, sau đó nhập tên bảng, các thuộc tính trong bảng, các kiểu của thuộc tính ...



Hình 3.12. Cách thêm bảng trong database

Như vậy bạn sẽ có file sql chứa cơ sở dữ liệu của website.

### 3.3. Cơ sở dữ liệu web

#### 3.3.1. Khái niệm

Một cơ sở dữ liệu Web là một ứng dụng cơ sở dữ liệu được thiết kế để quản lý và được truy cập thông qua Internet. Các nhà khai thác trang web có thể quản lý bộ sưu tập dữ liệu này và trình bày các kết quả phân tích dựa trên dữ liệu trong ứng dụng cơ sở dữ liệu Web. Cơ sở dữ liệu lần đầu tiên xuất hiện vào những năm 1990, và đã là một tài sản cho các doanh nghiệp, cho phép thu thập dữ liệu dường như vô hạn từ dữ liệu vô hạn của khách hàng[6].

### **3.3.2. Tổ chức dữ liệu**

Các cơ sở dữ liệu Web cho phép thu thập dữ liệu được tổ chức và biên mục một cách triệt để trong hàng trăm thông số. Cơ sở dữ liệu Web không yêu cầu phải có kỹ năng máy tính thành thạo, và nhiều chương trình phần mềm cơ sở dữ liệu cung cấp kiểu "click-and-create", nó dễ dàng mà không cần mã hóa phức tạp. Ta chỉ cần điền vào các trường và lưu từng bản ghi. Tổ chức dữ liệu theo cách bạn chọn, chẳng hạn như theo thời gian, theo thứ tự bảng chữ cái hoặc theo một bộ thông số cụ thể[6].

### **3.3.3. Phần mềm Cơ sở dữ liệu Web**

Các chương trình phần mềm cơ sở dữ liệu Web được tìm thấy trong các chương trình xuất bản dành cho máy tính để bàn, chẳng hạn như Microsoft Office Access và OpenOffice Base. Các chương trình khác bao gồm cơ sở dữ liệu Webex WebOffice và cơ sở dữ liệu Web FormLogix. Các ứng dụng phần mềm tiên tiến nhất có thể thiết lập mẫu thu thập dữ liệu, thăm dò ý kiến, các mẫu phản hồi và phân tích dữ liệu hiện tại trong thời gian thực[6].

### **3.3.4. Đối tượng áp dụng**

Các doanh nghiệp cả lớn và nhỏ có thể sử dụng các cơ sở dữ liệu Web để tạo ra các cuộc thăm dò trang web, các mẫu phản hồi, khách hàng hoặc khách hàng và các danh mục hàng tồn kho. Cơ sở dữ liệu Web hoàn toàn tùy biến được với nhu cầu của một cá nhân hoặc doanh nghiệp.

### **3.3.5. Hệ quản trị CSDL MySQL**

MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất lớn. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

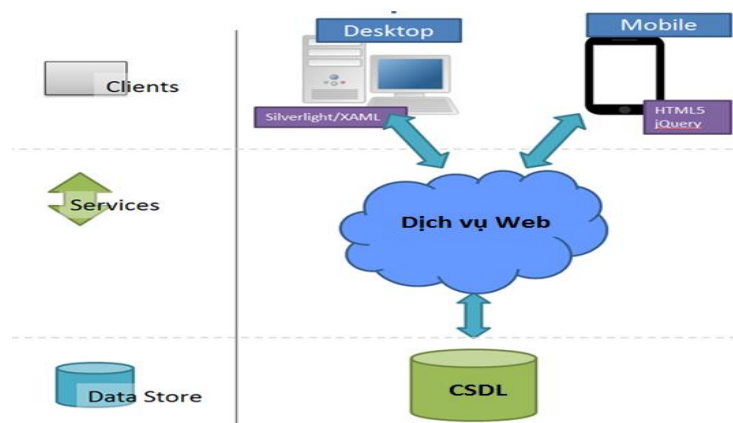
MySQL được sử dụng cho việc hỗ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OSX, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS.

## 3.4. Dịch vụ web<sup>5</sup>

### 3.4.1. Giới thiệu

#### Định nghĩa

Là sự kết hợp các ứng dụng trên máy tính cá nhân, thiết bị di động với ứng dụng trên các thiết bị khác, các cơ sở dữ liệu và các mạng máy tính để tạo thành một cơ cấu tính toán hoàn hảo mà người sử dụng có thể làm việc, yêu cầu, phân tích, khai thác, cập nhật,... thông tin với nó thông qua mạng Internet hoặc Wifi[1].



Hình 3.13. Dịch vụ Web

Theo định nghĩa của W3C (World Wide Web Consortium):

- ✚ Dịch vụ Web là một hệ thống phần mềm được thiết kế để hỗ trợ khả năng tương tác giữa các ứng dụng trên các máy tính khác nhau thông qua mạng Internet, giao diện chung và sự gắn kết của nó được mô tả bằng XML.
- ✚ Là tài nguyên phần mềm có thể xác định bằng địa chỉ URL.
- ✚ Thực hiện các chức năng và đưa ra các thông tin người dùng yêu cầu.
- ✚ Ứng dụng cơ bản của Dịch vụ Web là tích hợp các hệ thống.
- ✚ Các ứng dụng được tích hợp với cơ sở dữ liệu và các ứng dụng khác, người sử dụng sẽ giao tiếp với CSDL để tiến hành phân tích và lấy dữ liệu.

### 3.4.2. Các công nghệ xây dựng dịch vụ Web

Phân loại theo Công nghệ sử dụng để xây dựng dịch vụ Web cho ứng dụng di động dùng .NET Framework (phiên bản 4.5.3), hiện nay có các công nghệ chính sau:

- ✚ Web Service
- ✚ WCF Service
- ✚ WCF REST Service

<sup>5</sup> Web Service



## ✚ Web API Service

### 3.4.3. Đặc điểm của dịch vụ Web

#### a. Ưu điểm

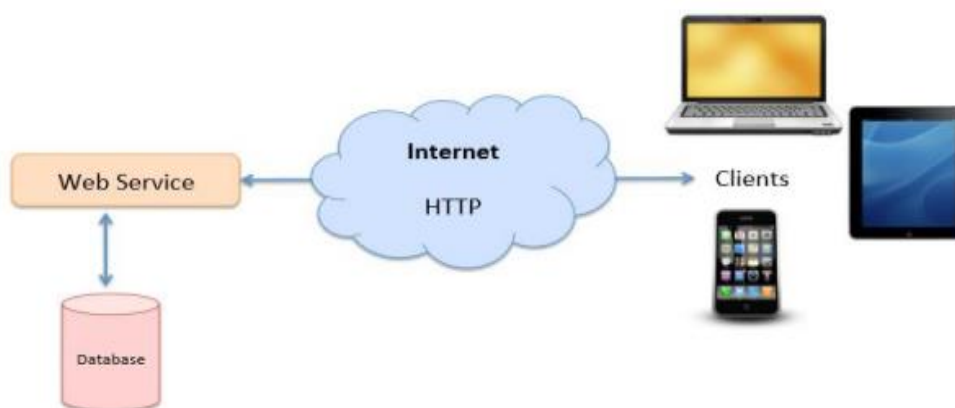
- Cung cấp khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau[5].
- Sử dụng các giao thức và chuẩn mở.
- Nâng cao khả năng tái sử dụng.
- Thúc đẩy đầu tư các hệ thống phần mềm đã tồn tại.
- Tạo mối quan hệ tương tác lẫn nhau và mềm dẻo giữa các thành phần trong hệ thống, dễ dàng cho việc phát triển các ứng dụng phân tán.
- Thúc đẩy hệ thống tích hợp, giảm sự phức tạp của hệ thống, hạ giá thành hoạt động, phát triển hệ thống nhanh và tương tác hiệu quả với hệ thống của các doanh nghiệp khác.

#### b. Khuyết điểm

- Vào những khoảng thời gian chết của Web service sẽ dẫn đến những thiệt hại lớn[5]:
  - ✚ Giao diện không thay đổi
  - ✚ Có thể lỗi nếu một máy khách không được nâng cấp
  - ✚ Thiếu các giao thức cho việc vận hành
- Có quá nhiều chuẩn cho Web Service khiến người dùng khó nắm bắt.
- Phải quan tâm nhiều hơn đến vấn đề an toàn và bảo mật.

### 3.4.4. Kiến trúc của dịch vụ Web

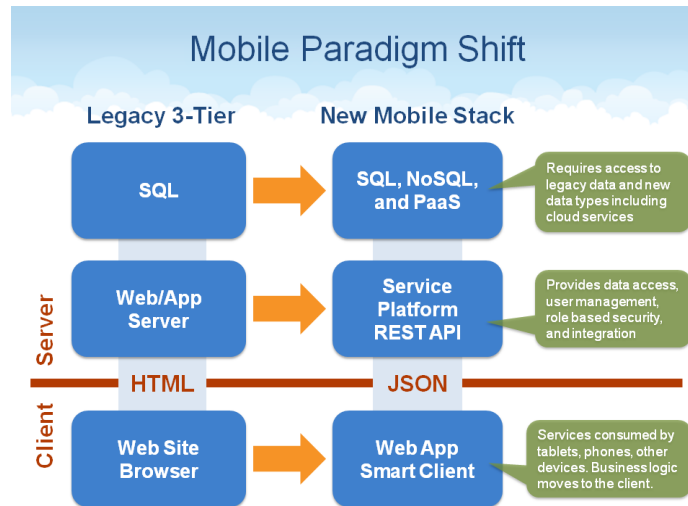
#### a. Mô hình hoạt động



Hình 3.14. Mô hình hoạt động của Web Service

## b. Dịch vụ Web gồm có 4 chuẩn chính:

- 1) SOAP (Simple Object Access Protocol)
- 2) WSDL (Web Service Description Language)
- 3) UDDI (Universal Description, Discovery, and Integration)
- 4) REST (Representational State Transfer)



Hình 3.15. Xây dựng và triển khai Web Service cho ứng dụng di động

### SOAP?

- ✚ SOAP là một giao thức giao tiếp có cấu trúc như XML và mã hóa thành định dạng chung cho các ứng dụng trao đổi với nhau.
- ✚ Ý tưởng bắt đầu từ Microsoft và phần mềm Userland.
- ✚ Một đặc tả việc sử dụng các tài liệu XML theo dạng các thông điệp.
- ✚ Đặc tả về SOAP định nghĩa một mô hình trao đổi dữ liệu dựa trên 3 khái niệm cơ bản: Các thông điệp là các tài liệu XML, chúng được truyền đi từ bên gửi đến bên nhận, bên nhận có thể chuyển tiếp dữ liệu đến nơi khác[7].

### Đặc trưng của SOAP

- ✚ SOAP được thiết kế đơn giản và dễ mở rộng.
- ✚ Tất cả các message SOAP đều được mã hóa sử dụng XML.
- ✚ SOAP sử dụng giao thức truyền dữ liệu riêng.
- ✚ Không có garbage collection phân tán, và cũng không có cơ chế tham chiếu. Vì thế SOAP client không giữ bất kỳ một tham chiếu đầy đủ nào về các đối tượng ở xa.
- ✚ SOAP không bị ràng buộc bởi bất kỳ ngôn ngữ lập trình nào hoặc công nghệ nào.

## WSDL?

- ✚ WSDL là một ngôn ngữ dựa trên nền XML dùng để định vị và mô tả WS.
- ✚ WSDL là viết tắt của Web services Description language.
- ✚ WSDL dựa trên nền tảng XML.
- ✚ WSDL được dùng để mô tả WS.
- ✚ WSDL được dùng để định vị WS.
- ✚ WSDL là một tiêu chuẩn W3C.

WSDL định nghĩa cách mô tả web service theo cú pháp tổng quát XML, bao gồm các thông tin[7] :

- ✚ Tên service
- ✚ Giao thức và kiểu mã hóa sẽ được sử dụng khi gọi các hàm của web service.
- ✚ Loại thông tin: những thao tác, những tham số, những kiểu dữ liệu gồm có giao diện của web service, công với tên cho giao diện này.
- ✚ Phần giao diện mô tả giao diện và giao thức kết nối, phần thi hành mô tả thông tin để truy xuất service.

## UDDI ?

- ✚ UDDI là một thư mục dịch vụ, nơi mà chúng ta có thể đăng ký và tìm kiếm các dịch vụ web.
- ✚ UDDI là viết tắt của Universal Description, Discovery and integration.
- ✚ UDDI là một thư mục dùng để lưu trữ thông tin về các dịch vụ web.
- ✚ UDDI là một thư mục các giao diện dịch vụ web được mô tả bởi WSDL.
- ✚ UDDI giao tiếp thông qua SOAP.
- ✚ UDDI được xây dựng để góp phần vào nền tảng Microsoft .NET.

Để có thể sử dụng các dịch vụ, trước tiên client phải tìm dịch vụ, ghi nhận thông tin về cách sử dụng dịch vụ và biết được đối tượng cung cấp dịch vụ. UDDI định nghĩa một số thành phần cho biết trước các thông tin này để cho phép các client truy tìm và nhận lại những thông tin yêu cầu sử dụng web services[7].

Cấu trúc UDDI gồm các thành phần:

- ✚ White pages: chứa thông tin liên hệ và các định dạng chính yếu của web services (tên giao dịch, địa chỉ...) Những thông tin này cho phép các đối tượng khác xác định được service.

- ✚ Yellow pages: chứa thông tin mô tả web service theo những chủng loại khác nhau. Những thông tin này cho phép các đối tượng thấy web service theo từng chủng loại của nó.
- ✚ Green pages: chứa thông tin kỹ thuật mô tả các hành vi và các chức năng của web service để tìm kiếm.

## **REST?**

REST (Representational State Transfer) là một kiến trúc phần mềm cho các hệ thống phân tán siêu truyền thông như WWW, được chọn sử dụng rộng rãi thay cho Web service dựa trên SOAP và WSDL.

### **Đặc trưng của REST**

- ✚ Là dạng client – server.
- ✚ Phân tách giao diện của client ra khỏi dữ liệu.
- ✚ Cho phép mỗi thành phần phát triển độc lập.
- ✚ Hỗ trợ đa nền tảng.
- ✚ Mỗi yêu cầu từ client phải có đủ thông tin cần thiết để server có thể hiểu được mà không cần phải lưu trữ thêm thông tin nào trước đó.
- ✚ Tất cả tài nguyên được truy cập thông qua một interface thống nhất (HTTP GET, PUT, POST, DELETE, ...).

#### **3.4.5. Các thành phần chính của dịch vụ web**

- ✚ XML – eXtensible Markup Language/ JSON (JavaScript Object Notation)
- ✚ WSDL – Web Service Description Language Universal Description, Discovery, and Integration (UDDI)
- ✚ SOAP – Simple Object Access Protocol
- ✚ RSS – Really Simple Syndication
- ✚ RDF – Resource Description Framework
- ✚ RESTful - Representational State Transfer
- ✚ WADL - Web Application Description Language[5].

#### **3.4.6. Xây dựng một dịch vụ Web**

- Có 4 giai đoạn chính để xây dựng một dịch vụ Web[5] :

- ✚ **Giai đoạn xây dựng** : phát triển và chạy thử ứng dụng dịch vụ Web, xây dựng các chức năng và định nghĩa dịch vụ.

- ✚ **Giai đoạn triển khai** công bố định nghĩa dịch vụ, xây dựng WSDL và triển khai mã thực thi của dịch vụ Web.
  - ✚ **Giai đoạn tiến hành** tìm kiếm và gọi thực thi dịch vụ Web bởi những người sử dụng dịch vụ.
  - ✚ **Giai đoạn quản lý** quản lý và quản trị dịch vụ, duy trì sự ổn định của dịch vụ, cập nhật thông tin mới, sửa lỗi khi nó xảy ra.
- Quy trình xây dựng một dịch vụ Web bao gồm các bước sau:

- 1) Định nghĩa và xây dựng các chức năng, các dịch vụ mà dịch vụ sẽ cung cấp.
- 2) Tạo WSDL cho dịch vụ.
- 3) Xây dựng SOAP hoặc REST server
- 4) Đăng ký WSDL với UDDI registry để cho phép các client có thể tìm thấy và truy xuất.
- 5) Client nhận tập tin WSDL và từ đó xây dựng SOAP hoặc REST client để có thể kết nối với SOAP hoặc REST server
- 6) Xây dựng ứng dụng phía client và sau đó gọi thực hiện dịch vụ thông qua việc kết nối tới SOAP hoặc REST server.

### **3.4.7. Các dạng tương tác giữa Web Service với ứng dụng trên TBDD**

#### **a. XML - eXtensible Markup Language**

- ✚ Là ngôn ngữ đánh dấu với mục đích chung do W3C đề nghị.
- ✚ Là một dạng chuẩn cho phép lưu các thông tin hướng cấu trúc, được tổ chức dưới dạng thẻ (tag) tương ứng.
- ✚ Các thẻ (tag) của XML thường không được định nghĩa trước mà chúng được tạo ra theo quy ước của người, (hoặc Chương trình) tạo ra XML theo những quy ước của chính người tạo.
- ✚ Giúp đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với Internet.
- ✚ Sử dụng các khai báo kiểu dữ liệu DTD (Document Type Definition) hay lược đồ Schema để mô tả dữ liệu.

## b. Định dạng JSON - JavaScript Object Notation

- ✚ Định nghĩa dữ liệu theo ngôn ngữ JavaScript, tiêu chuẩn ECMA-262 năm 1999.
- ✚ Là một định dạng văn bản đơn giản với các trường dữ liệu được lồng vào nhau.
- ✚ Dùng để trao đổi dữ liệu giữa các thành phần của một hệ thống tương thích với hầu hết các ngôn ngữ C, C++, C#, Java, JavaScript, Perl, Python...

### Vì sao nên sử dụng JSON?

- ✚ Có thể đọc hiểu và dễ dàng tiếp cận (human-readability).
- ✚ Dữ liệu truyền tải ngắn gọn so với những định dạng dữ liệu khác như: XML, HTML, ... → Tiết kiệm dung lượng hơn XML, HTML,...
- ✚ Dễ dàng chuyển đổi (parse) dữ liệu từ dạng chuỗi (nhận từ server) sang dữ liệu có thể sử dụng được (thành Object, Number, Array).
- ✚ Dễ truy cập nội dung.
- ✚ Với những ứng dụng AJAX lấy và xử lý dữ liệu từ 1 web service nào đó khác domain. Nếu nội dung trả về có dạng JSON thì javascript từ trang web của chúng ta có thể trực tiếp truy cập (dùng lệnh eval).
- ✚ Có cách phân tích xử lý đơn giản, chỉ qua 1 bước (chuyển chuỗi nhận được từ máy chủ sang dữ liệu có thể sử dụng được).

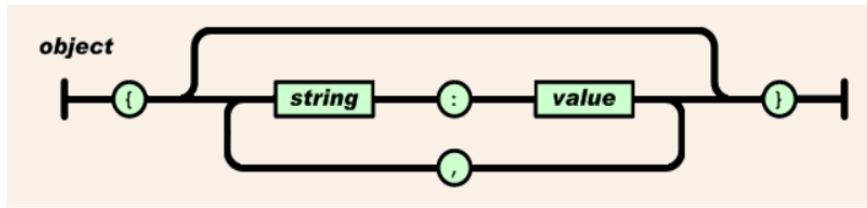
### Các kiểu dữ liệu JSON

JSON có 5 kiểu dữ liệu chính[8] :

- 1) Kiểu số (Number)
- 2) Kiểu chuỗi (String)
- 3) Kiểu mảng (Array)
- 4) Kiểu đối tượng (Object)
- 5) Giá trị (Value)

### Kiểu đối tượng (Object)

- ✚ 1 đối tượng là 1 hỗn độn của các cặp tên và giá trị.
- ✚ 1 đối tượng bắt đầu bởi dấu ngoặc đơn trái { và kết thúc với dấu ngoặc đơn phải }.
- ✚ Từng tên được theo sau bởi dấu 2 chấm (:) và các cặp tên/giá trị được tách ra bởi dấu phẩy (,).
- ✚ 1 đối tượng có thể chứa nhiều cặp tên/giá trị (Name/Values).

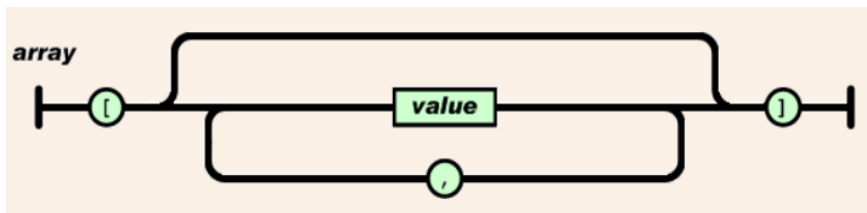


Ví dụ:

```
{ "ten": "Hai" , "ho": "Nguyen" }
```

### **Kiểu mảng (Array)**

- + 1 mảng là 1 tập hợp các giá trị đã được sắp xếp.
- + 1 mảng bắt đầu bởi dấu mở ngoặc vuông [ và kết thúc với dấu ngoặc vuông phải ].
- + Các giá trị được cách nhau bởi dấu phẩy (,).
- + 1 mảng có thể chứa nhiều đối tượng.



Ví dụ:

```
{
  "NhanVien":
  [
    {"tenNV": "An" , "hoNV": "Nguyễn" },
    {"tenNV": "Bình" , "hoNV": "Trần" },
    {"tenNV": "Châu" , "hoNV": "Lê" }
  ]
}
```

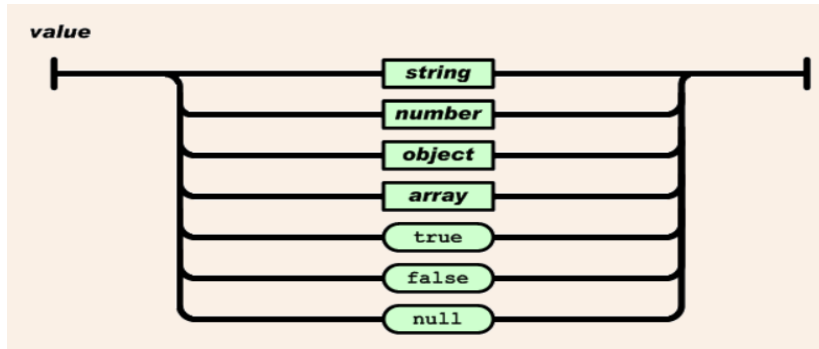
### **Kiểu giá trị (Value)**

- Giá trị JSON có thể là:

- + 1 số (nguyên (integer) hay số thực (floating point))
- + 1 chuỗi (nằm trong dấu nháy đôi "")
- + 1 số luận lý (true hoặc false)
- + 1 mảng (nằm trong dấu ngoặc vuông [])
- + 1 đối tượng (nằm trong dấu ngoặc nhọn {})

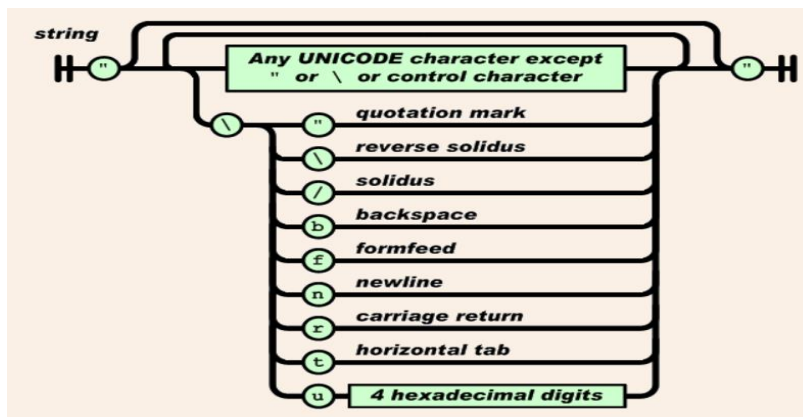
✚ Kiểu null

- Những cấu trúc này có thể đã được lồng vào nhau.



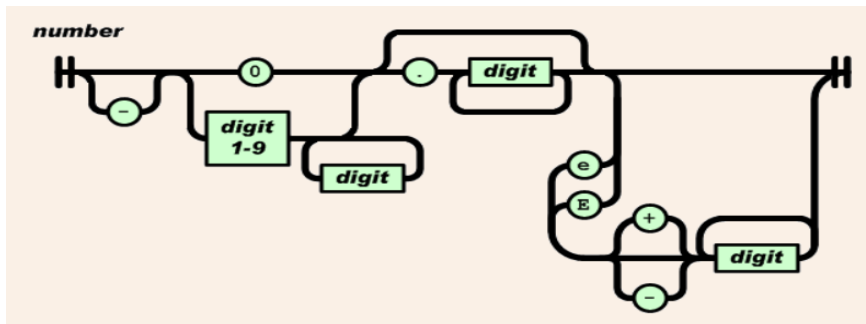
**Kiểu chuỗi (String)**

- ✚ 1 chuỗi (string) là 1 tập hợp của các số hay mẫu tự Unicode, được bao bọc trong các dấu trích dẫn kép (").
- ✚ Dùng dấu chéo (/) để thoát khỏi 1 chuỗi.
- ✚ 1 ký tự đã được hiển thị như là 1 chuỗi ký tự đơn độc.



**Kiểu số (Number)**

Giống với kiểu số trong C và Java, trừ định dạng bát phân và hex là không thể dùng.





### 3.4.8. Cấu trúc chung Web Service viết bằng PHP cho ứng dụng di động

Thông thường gồm 4 thành phần :

#### a. Kết nối CSDL Web

```
$conn = mysqli_connect("localhost","id4201482_sa","svcntt","id4201482_db_doan");
mysqli_query($conn, "SET NAMES 'utf8'");
```

#### b. Truy vấn CSDL Web

```
$query = "SELECT * FROM loaisanpham";
$data = mysqli_query($conn,$query);
```

#### c. Mã hóa dữ liệu sang dạng JSON

```
echo json_encode($mangsp);
```

#### d. Trả dữ liệu KQ về cho ứng dụng di động

```
<?php
include "connect.php";
$query = "SELECT * FROM loaisanpham";
$data = mysqli_query($conn,$query);
$mangsp = array();
while ($row = mysqli_fetch_assoc($data)) {
    array_push($mangsp,new loaisp(
        $row['maloaisanpham'],
        $row['tenloaisanpham'],
        $row['hinhanhloaisanpham']));
}
echo json_encode($mangsp);
class loaisp{

    function loaisp($maloaisanpham,$tenloaisanpham,$hinhanhloaisanpham){
        $this->maloaisanpham = $maloaisanpham;
        $this->tenloaisanpham = $tenloaisanpham;
        $this->hinhanhloaisanpham = $hinhanhloaisanpham;
    }
}
?>
```

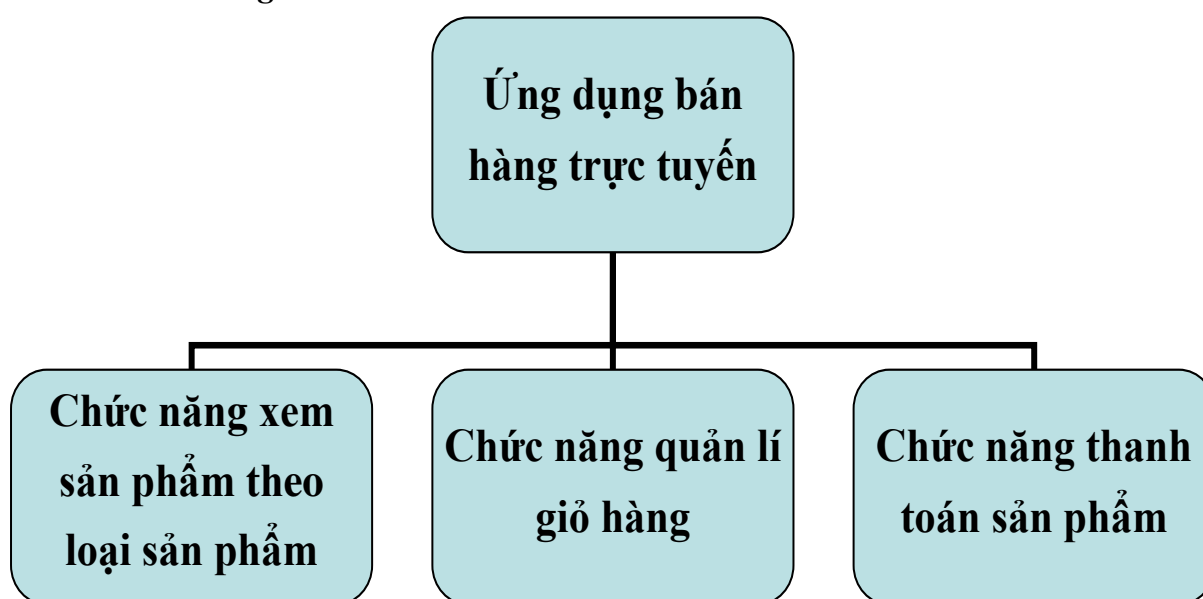
## CHƯƠNG 4: CHƯƠNG TRÌNH THỬ NGHIỆM

### 4.1. Phát biểu bài toán

Cửa hàng Thiết bị Ngon - Bỏ - Rẻ chuyên cung cấp, kinh doanh các loại sản phẩm công nghệ như điện thoại, laptop, PC, Tivi... tại Hà Nội. Cửa hàng đã có trang web để phục vụ cho khách hàng có thể nắm bắt được những thông tin của các sản phẩm cũng như phục vụ cho việc mua hàng online của những khách hàng ở xa được thuận tiện hơn. Nhưng nay cửa hàng muốn tạo thêm ứng dụng mua bán dành cho những người sử dụng hệ điều hành Android. Với ứng dụng này, người dùng chỉ việc tải ứng dụng về máy và sử dụng, việc này sẽ giúp cho những thao tác của khách hàng khi xem những thông tin sản phẩm, đặt mua, cập nhật các chương trình khuyến mãi của cửa hàng trở nên dễ dàng và tiện lợi hơn rất nhiều.

### 4.2. Thiết kế chức năng hệ thống

*Sơ đồ chức năng*



#### 4.2.1. Chức năng xem sản phẩm theo loại sản phẩm

##### a. Mô tả chức năng

Chức năng xem sản phẩm theo từng loại sản phẩm cho phép người dùng xem sản phẩm theo các loại cụ thể như : Điện thoại, Laptop, Tivi, Máy giặt, Điều hòa, Tủ lạnh.... Thuận tiện cho người dùng muốn tìm hiểu về số lượng các mặt hàng , các loại sản phẩm được kinh doanh, cũng như giá cả, thông số kỹ thuật của từng loại sản phẩm trên thị trường.

## **b. Mô tả hoạt động**

- ✚ Chương trình đưa ra một danh sách các loại sản phẩm được kinh doanh.
- ✚ Người dùng chọn loại sản phẩm cần xem.
- ✚ Chương trình đưa ra danh sách các sản phẩm theo loại sản phẩm mà người dùng đã chọn.
- ✚ Người dùng chọn sản phẩm cần xem.
- ✚ Chương trình đưa ra những thông tin chi tiết liên quan đến sản phẩm đó.

### **4.2.2. Chức năng quản lý giỏ hàng**

#### **a. Mô tả chức năng**

Chức năng quản lý giỏ hàng cho phép người dùng có thể tạo và quản lý danh sách các sản phẩm, chức năng cung cấp các thao tác như: thêm sản phẩm vào giỏ hàng và xóa sản phẩm đã thêm trong giỏ hàng giúp người dùng có thể dễ dàng quản lý danh sách các sản phẩm trong giỏ hàng của mình.

#### **b. Mô tả hoạt động**

- ✚ Chương trình đưa ra thông tin chi tiết sản phẩm
- ✚ Người dùng chọn “Thêm vào giỏ hàng” nếu muốn mua sản phẩm đó.
- ✚ Chương trình đưa ra danh sách những sản phẩm đã được thêm vào giỏ hàng.
- ✚ Người dùng có thể Thêm hoặc Xóa những sản phẩm trong giỏ hàng.

### **4.2.3. Chức năng thanh toán sản phẩm**

#### **a. Mô tả chức năng**

Chức năng thanh toán sản phẩm cho phép người dùng có thể mua những sản phẩm đã được chọn trong giỏ hàng của mình. Chức năng này giúp cho người dùng tiết kiệm được thời gian và chi phí đi lại khi muốn mua những sản phẩm tại cửa hàng.

#### **b. Mô tả hoạt động**

- ✚ Chương trình cho người dùng nhập thông tin khách hàng để lập đơn hàng.
- ✚ Khách hàng nhập đầy đủ thông tin và ấn xác nhận để hoàn tất thanh toán.
- ✚ Nhà cung cấp sẽ gọi điện xác nhận thông tin đơn hàng với khách hàng trước khi giao hàng.

### 4.3. Thiết kế dữ liệu

#### 4.3.1. Bảng thông tin sản phẩm

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Mô tả	Ghi chú
1	masanpham	int	11	Mã sản phẩm	Khóa chính
2	tensanpham	varchar	200	Tên sản phẩm	
3	giasanpham	int	15	Giá sản phẩm	
4	hinhsanpham	varchar	200	Hình ảnh sản phẩm	
5	motasanpham	varchar	10000	Mô tả sản phẩm	
6	maloaisanpham	int	5	Mã loại sản phẩm	Khóa phụ

#### 4.3.2. Bảng thông tin sản phẩm mới nhất

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Mô tả	Ghi chú
1	masanphammn	int	11	Mã sản phẩm	Khóa chính
2	tensanphammn	varchar	200	Tên sản phẩm	
3	giasanphammn	int	15	Giá sản phẩm	
4	hinhsanphammn	varchar	200	Hình ảnh sản phẩm	
5	motasanphammn	varchar	10000	Mô tả sản phẩm	
6	maloaisanpham	int	5	Mã loại sản phẩm	Khóa phụ

#### 4.3.3. Bảng chi tiết đơn hàng

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Mô tả	Ghi chú
1	machitietdonhang	int	11	Mã chi tiết đơn hàng	Khóa chính
2	tensanpham	varchar	500	Tên sản phẩm	
3	soluongsanpham	int	11	Số lượng sản phẩm	
4	giasanpham	int	11	Giá sản phẩm	
5	makhachhang	int	11	Mã khách hàng	
6	masanpham	int	11	Mã sản phẩm	Khóa phụ

#### 4.3.4. Bảng thông tin loại sản phẩm

STT	Tên trường	Kiểu dữ liệu	Độ rộng	Mô tả	Ghi chú
1	maloaisanpham	int	11	Mã loại sản phẩm	Khóa chính
2	tenloaisanpham	varchar	200	Tên loại sản phẩm	
3	hinhloaisanpham	varchar	200	Hình ảnh loại sản phẩm	

#### 4.3.5. Bảng thông tin đơn hàng

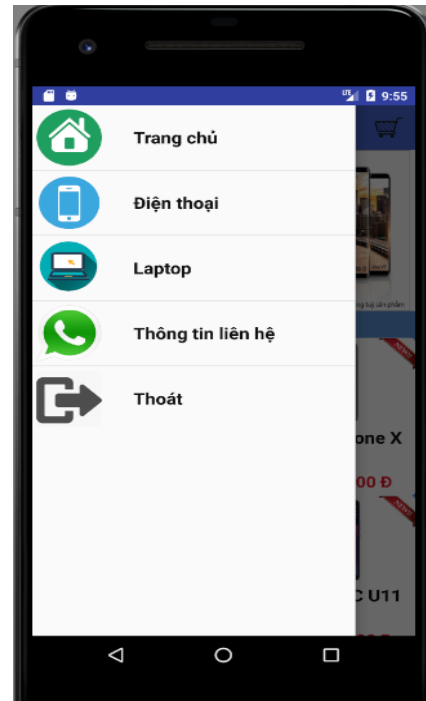
STT	Tên trường	Kiểu dữ liệu	Độ rộng	Mô tả	Ghi chú
1	makhachhang	int	11	Mã khách hàng	Khóa chính
2	tenkhachhang	varchar	200	Tên khách hàng	
3	sodienthoai	int	13	Số điện thoại	
4	email	varchar	200	Email khách hàng	
5	diachi	varchar	1000	Địa chỉ khách hàng	

### 4.4. Một số giao diện chương trình

#### 4.4.1 Giao diện màn hình chính và thanh menu của ứng dụng



Hình 4.1. Giao diện màn hình chính



Hình 4.2. Giao diện thanh menu

#### 4.4.2. Giao diện danh sách các sản phẩm của Điện thoại và Laptop



Hình 4.3. Giao diện các sản phẩm điện thoại



Hình 4.4. Giao diện các sản phẩm Laptop

#### 4.4.3. Giao diện thông tin chi tiết của các sản phẩm

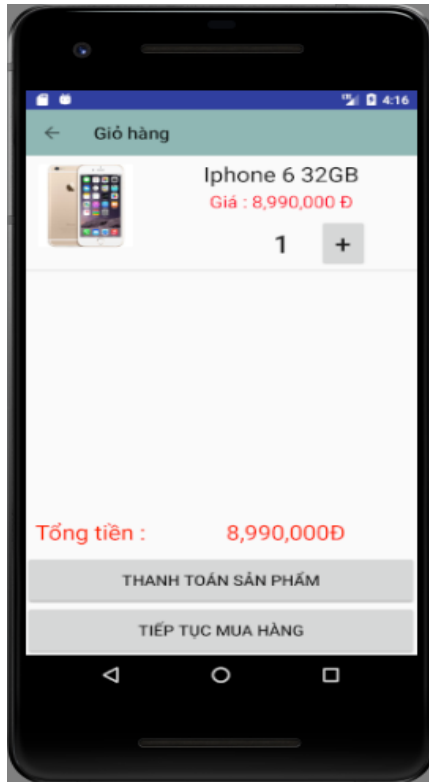


Hình 4.5. Giao diện chi tiết sản phẩm điện thoại

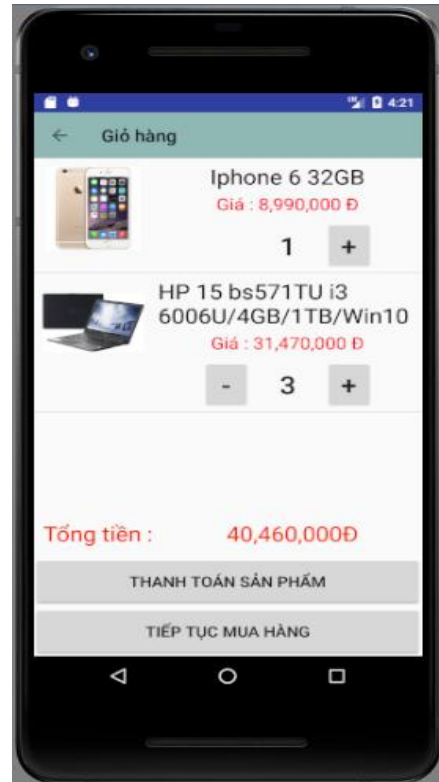


Hình 4.6. Giao diện chi tiết sản phẩm laptop

#### 4.4.4. Giao diện các sản phẩm trong giỏ hàng



Hình 4.7. Giao diện giỏ hàng có 1 sản phẩm

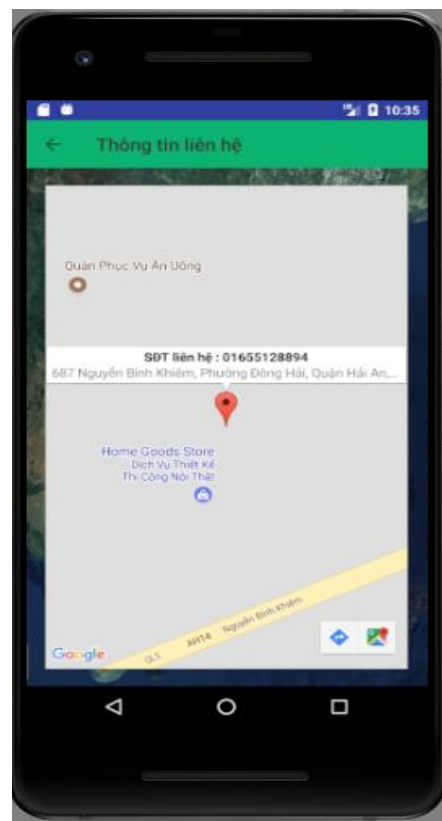


Hình 4.8. Giao diện giỏ hàng có 2 sản phẩm

#### 4.4.5. Giao diện thanh toán sản phẩm và thông tin liên hệ



Hình 4.9. Giao diện thanh toán sản phẩm



Hình 4.10. Giao diện thông tin liên hệ

## KẾT LUẬN

Trong đồ án này em đã nghiên cứu, tìm hiểu về dịch vụ web và cách phát triển ứng dụng trên Hệ điều hành Android.

Đồ án đã thực hiện các nhiệm vụ sau:

- Tìm hiểu lịch sử và kiến trúc của Hệ điều hành Android.
- Nắm bắt được quy trình làm một phần mềm trên mobile.
- Tìm hiểu về Web, Web Service cho thiết bị di động.
- Tìm hiểu kỹ thuật lấy dữ liệu từ Web service cho thiết bị di động.
- Các bước phát triển một ứng dụng trên Hệ Điều Hành Android.
- Phát triển ứng dụng mua bán thiết bị điện tử.

Trong thời gian nghiên cứu, xây dựng chương trình, em đã hết sức cố gắng làm việc với sự giúp đỡ tận tình của thầy giáo hướng dẫn. Chương trình đã đạt được kết quả nhất định. Tuy nhiên với thời gian ngắn, trình độ và kinh nghiệm còn hạn chế chương trình vẫn còn nhiều thiếu sót. Rất mong các thầy cô giáo và các bạn tận tình giúp đỡ để chương trình ngày càng được hoàn thiện hơn.

Hướng phát triển chương trình trong tương lai sẽ tiếp tục bổ sung thêm nhiều loại sản phẩm, cũng như các chức năng khác và phát triển thêm giao diện chương trình để giúp người dùng sử dụng tiện ích hơn.



## TÀI LIỆU THAM KHẢO

- [1] <https://vi.wikipedia.org>.
- [2] <https://monhoc.vn>.
- [3] <https://genk.vn>.
- [4] <https://yellowcodebooks.com>.
- [5] <http://expressmagazine.net>.
- [6] <https://www.techwalla.com>.
- [7] <https://viblo.asia>.
- [8] <https://www.json.org>.