**TRƯỜNG ĐẠI BÁCH KHOA HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**DOÃN MINH ĐẠT**

**PHÁT TRIỂN ỨNG DỤNG TÌM KIẾM KHÁCH SẠN THÔNG MINH TRÊN NỀN TẢNG ANDROID**

**KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT**

**Hà Nội, 2015**

**TRƯỜNG ĐẠI BÁCH KHOA HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**DOÃN MINH ĐẠT - 20115505**

**PHÁT TRIỂN ỨNG DỤNG TÌM KIẾM KHÁCH SẠN THÔNG MINH TRÊN NỀN TẢNG ANDROID**

**KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT**

**GIÁO VIÊN HƯỚNG DẪN**

**TS.TRẦN VĨNH ĐỨC**

**NIÊN KHÓA 2011 – 2015**

# LỜI CÁM ƠN

Em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại Học Bách Khoa Hà Nội đã tạo điều kiện tốt cho em thực hiện đề tài này.

Em xin chân thành cảm ơn Thầy Trần Vĩnh Đức, là người đã tận tình hướng dẫn, chỉ bảo em trong suốt thời gian thực hiện đề tài. Em cũng xin cảm ơn một số bạn cùng lớp đã có những trao đổi, những chỉ dẫn giúp em giải quyết các vấn đề và hoàn thiện đề tài.

Em cũng xin gửi lời cảm ơn sâu sắc đến quý Thầy Cô trong Khoa đã tận tình giảng dạy, trang bị cho em những kiến thức quí báu trong những năm học vừa qua.

Em xin gửi lòng biết ơn sâu sắc đến gia đình và bạn bè đã ủng hộ, giúp đỡ và động viên em trong những lúc khó khăn cũng như trong suốt thời gian học tập và nghiên cứu.

Mặc dù em đã cố gắng hoàn thành sản phẩm trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, kính mong sự cảm thông và tận tình chỉ bảo của quý Thầy Cô và các bạn.

Sinh Viên thực hiện

Doãn Minh Đạt

# KHÁI QUÁT VỀ ĐỀ TÀI

**Tên Đề Tài**: Phát triển ứng dụng tìm kiếm khách sạn trên nền tảng Android

**Giáo viên hướng dẫn**: TS. Trần Vĩnh Đức

**Thời gian thực hiện**: từ ngày 15/11/2015 đến ngày 30/12/2015

**Loại đề tài:** Xây dựng ứng dụng dựa theo nền tảng công nghệ

**Môi trường làm việc** : Sản phẩm được thực hiện bởi cá nhân với sự trợ giúp của giáo viên hướng dẫn.

**Mục tiêu đề tài:**

Đề tài này thuộc hướng tìm hiểu công nghệ từ đó xây dựng ứng dụng. Mục tiêu của đề tài là tìm hiểu nền tảng Android và xây dựng được ứng dụng chạy ổn định trên môi trường Android cho thiết bị di động.

**Nội dung luận văn**

Luận văn bao gồm 5 chương:

**Chương 1:** Giới thiệu tổng quan.

**Chương 2:** Cơ sở lý thuyết.

**Chương 3:** Phân tích thiết kế.

**Chương 4:** Xây dựng ứng dụng.

**Chương 5:** Kết luận và hướng phát triển.

**Chương 1 : TỔNG QUAN**

**1.1.Giới thiệu tổng quan và lý do chọn đề tài**

Hơn 10 năm trước đây, mức tăng trrưởng du lịch Việt Nam vẫn còn xếp vào hàng cuối cùng trong khu vực, nhưng nay đã vươn lên hàng trung bình trung các nước ASEAN. Năm 1994 Việt Nam mới chỉ đón 1 triệu khách quốc tế, đến năm 2006 số lượng khách quốc tế đến Việt Nam đã tăng lên 3,6 triệu lượt người và năm 2007 con số này đạt gần 4,4 triệu lượt người, tăng gần 20 % so với năm 2006. Đối tượng khách quốc tế đi du lịch Việt Nam ngày càng mở rộng và đa dạng, bao gồm hầu hết các thị trường nguồn lớn trên thế giới từ Đông Bắc Á, ASEAN, châu âu, Bắc Mỹ đến các địa bàn châu lục khác. Để đáp ứng lượng khách tăng mạnh, ngành du lịch Việt Nam đã huy động các nguồn lực cả trong và ngoài nước để phát triển hạ tầng, cơ sở vật chất kỹ thuật và nhân lực du lịch. Cả nước hiện có 8.556 cơ sở lưu trú với tổng số 170.551 buồng, trong đó có 4.283 khách sạn được xếp hạng đạt tiêu chuẩn đến 5 sao với tổng số 109.198 buồng. Việt Nam hiện có trên 600 doanh nghiệp lữ hành quốc tế, trên 10.000 doanh nghiệp lữ hành nội địa. Phương tiện vận chuyển khách du lịch phát triển đa dạng và từng bước được hiện đại hoá. Nhiều cơ sở dịch vụ, vui chơi giải trí, sân golf… đã và đang được xây dựng để đáp ứng nhu cầu của du khách. Cùng với đó, sự phát triển của công nghệ di động đã bùng nổ trong vài năm qua. Các thiết bị cầm tay đã trở nên mạnh mẽ và rất phổ biến. Trong vài năm trở lại đây, hệ điều hành Android ra đời và trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới, 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. Những ưu điểm dễ thấy nhất của Android chính là khả năng tùy biến nhanh chóng, dễ làm quen, ứng dụng hỗ trợ phong phú, tương thích với đa cấu hình phần cứng của các nhà sản xuất. Dù đã hay chưa từng sử dụng smartphone Android thì bạn cũng không cần quá lo lắng về việc thay đổi. Hầu như tất cả những smartphone sử dụng hệ điều hành Android đều rất dễ sử dụng, đơn giản trong tùy biến theo sở thích cá nhân, cũng như có rất nhiều ứng dụng để thỏa mãn nhu cầu của từng người dùng. Đồng thời Android có mã nguồn mở đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Bởi vậy trước nhu cầu và lợi thế đó, em quyết định nghiên cứu và phát triển ứng dụng : **Tìm kiếm khách sạn thông minh trên nền tảng Android** để giúp khách du lịch trong và ngoài nước có thể tìm được những khách sạn với chất lượng dịch vụ tốt nhất,trải nghiệm những khoảnh khắc du lịch tuyệt vời tại Việt Nam.

**1.2.Mục tiêu đề tài**

1.s2.1.Lý thuyết

* Nghiên cứu phần mềm thiết kế đồ họa Photoshop CS5.
* Nghiên cứu phần mềm thiết kế bố cục ứng dụng Balsamiq Mockups.
* Nghiên cứu ngôn ngữ lập trình Java.
* Nghiên cứu lập trình ứng dụng trên nền tảng Android
* Nghiên cứu ngôn ngữ lập trình web PHP và Slim Framework
* Nghiên cứu hệ quản trị cơ sở dữ liệu MySQL và SQLite

1.2.2.Ứng dụng thực tiễn

* Xây dựng ứng dụng bao gồm client – server có khả năng giao tiếp với nhau.
* Ứng dụng hỗ trợ các tính năng giúp người dùng có thể tìm kiếm khách sạn ở gần,đặt phòng một cách dễ dàng.
* Ngoài ra người dùng có lưu thông tin cá nhân,xem lại lịch sử giao dịch.
* Liên kết mạng xã hội Facebook.

**1.3.Các bước nghiên cứu**

* Phân tích yêu cầu
* Lựa chọn công nghệ
* Nghiên cứu cơ sở lý thuyết của công nghệ đã chọn
* Áp dụng lý thuyết vào xây dựng ứng dụng thực tiễn
* Kiểm tra, tham khảo các ứng dụng khác để tối ưu hóa ứng dụng

**1.4.Bố cục đề tài**

* Tổng quan
* Cơ sở lý thuyết
* Phân tích thiết kế
* Xây dựng ứng dụng
* Kết luận và hướng phát triển

**Chương 2 : CƠ SỞ LÝ THUYẾT**

**2.1.Hệ điều hành Android**

2.1.1.Tổng quan



*Hình 2.1.1 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 cầm 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. 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.

Được xây dựng trên một nền tảng mở, và một bộ thư viện đa năng, mạnh mẽ với nguyên lý mở, Android đã nhanh chóng được cộng đồng lập trình viên di động hưởng ứng mạnh mẽ. Nền tảng Android tích hợp nhiều tính năng nổi bật:

* Android là một hệ điều hành nhân Linux, đảm bảo sự tương tác với các phần cứng, quản lý bộ nhớ, điều khiển các tiến trình tối ưu cho các thiết bị di động.
* Bộ ứng dụng khung cho phép sử dụng lại và thay thế các thành phần riêng lẻ.
* Máy ảo Dalvik được tối ưu cho các thiết bị di động, chạy các ứng dụng lập trình trên ngôn ngữ Java.
* Các thư viện cho phát triển ứng dụng mã nguồn mở bao gồm SQLite, WebKit, OpenGL và trình quản lý đa phương tiện.
* Hỗ trợ các chuẩn đa phương tiện phổ biến, thoại trên nền GSM, Bluetooth EDGE, 3G và Wifi
* Hỗ trợ Camera, GPS, la bàn, máy đo gia tốc…
* Bộ phát triển ứng dụng SDK đầy đủ gồm thiết bị giả lập, công cụ sửa lỗi, tích hợp với Eclipse SDK.

Android cung cấp một tập hợp đầy đủ các phần mềm cho thiết bị di động bao gồm: hệ điều hành, các khung ứng dụng và các ứng dụng cơ bản.

2.1.2.Đặc điểm

* Tính mở :

Android được xây dựng từ dưới đi lên cho phép người phát triển tạo các ứng dụng di động hấp dẫn với đầy đủ các điểm mạnh của các thiết bị cầm tay hiện có. Android hoàn toàn mở, một ứng dụng có thể gọi tới bất kể một chức năng lõi của điện thoại như tạo cuộc gọi, gửi tin nhắn hay sử dụng máy ảnh, cho phép người phát triển tạo phong phú hơn, liên kết hơn các tính năng cho người dùng. Android được xây dựng trên nhân Linux mở. Thêm nữa, nó sử dụng một máy ảo mà đã được tối ưu hóa bộ nhớ và phần cứng với môi trường di động. Android là một mã nguồn mở, nó có thể được mở rộng để kết hợp tự do giữa các công nghệ nổi trội. Nền tảng này sẽ tiếp tục phát triển bởi cộng đồng phát triển để tạo ra các ứng dụng di động hoàn hảo.

* Tính ngang hàng của các ứng dụng :

Với Android, không có sự khác nhau giữa các ứng dụng điện thoại cơ bản với ứng dụng của bên thứ ba. Chúng được xây dựng để truy cập như nhau tới một loạt các ứng dụng và dịch vụ của điện thoại. Với các thiết bị được xây dựng trên nền tảng Android, người dùng có thể đáp ứng đầy đủ các nhu cầu mà họ thích. Chúng ta có thể đổi màn hình nền, kiểu gọi điện thoại, hay bất kể ứng dụng nào. Chúng ta thậm chí có thể hướng dẫn điện thoại chỉ xem những ảnh mình thích.

* Dễ xây dựng ứng dụng :

Android cung cấp bộ thư viện giao diện lập trình ứng dụng đồ sộ và các công cụ để viết các ứng dụng phức tạp. Ví dụ, Android có thể cho phép người phát triển biết được vị trí của thiết bị và cho phép các thiết bị giao tiếp với nhau để có thể tạo nên mạng xã hội chia sẻ ngang hàng rộng khắp. Thêm nữa, Android còn bao gồm một bộ công cụ đầy đủ giúp cho việc phát triển trở nên dễ dàng.

2.1.3. So sánh với các hệ điều đành khác

Bảng 2.1.3 So Sánh Các Hệ Điều Hành Điện Thoại

|  |  |  |
| --- | --- | --- |
|  | Ưu điểm | Nhược điểm |
| Android | Google Android là nền tảng mở, cho phép người dùng có thể tùy biến nền tảng theo ý thích, hơn nữa lại có một Liên minh thiết bị cầm tay mở hậu thuẫn, Google Android đang là đối thủ xứng tầm của iPhone của Apple. Google đang tích cực mở rộng cộng đồng phát triển các ứng dụng cho Android. Bộ công cụ phát triển phần mềm (SDK) đầy đủ, hỗ trợ đa nền (Linux, Windows hay Mac OS) do chạy trên máy ảo Java. Thư viện ngày càng hoàn thiện, dễ dàng cho người lập trình. | Hệ điều hành phân mảnh, không thống nhất trên các thiết bị, giới hạn về độ 'mở' và nhiều lỗ hổng bảo mật là những yếu điểm của Android OS. |
| Windows Phone | Có thư viện API khá giống với API trên Win32, các công cụ hỗ trợ lập trình đầy đủ với Visual Studio, điều này làm cho những người phát triển trên Win32 không mất công tìm hiểu lại các API và các công cụ lập trình. | Sự có mặt của iPhone và Android là hai trở ngại lớn với Windows Mobile. Hai nền tảng này đang hoàn thiện và được người dùng rất ưa chuộng. |
| iOS | **Màn hình cảm ứng đa điểm:** iPhone sử dụng hoàn toàn bằng cảm ứng và không sử dụng các nút. Với iPhone ta có thể điều khiển trên màn hình kể cả việc trượt của các ngón tay. Ta có thể phóng to ảnh bằng cách trượt hai ngón tay ra xa và thu nhỏ bằng cách ngược lại.  **Bộ cảm nhận gia tốc***:* Những phản ứng nhanh chóng của bộ cảm nhận gia tốc thay đổi độ phân giải màn hình từ dọc sang ngang tự động khi ta đặt điện thoại nằm ngang. Điều này làm sinh động thêm cho các trò chơi. Âm thanh, hình ảnh hoàn hảo. | Việc lập trình trên cho iPhone phải thực hiện trên hệ điều hành Mac, do đó không phải ai cũng có thể lập trình cho iPhone. Hơn thế, nếu muốn đưa chương trình ra máy thật người lập trình phải trả một khoản phí lập trình, điều này làm giảm tính cạnh tranh so với các đối thủ khác. |

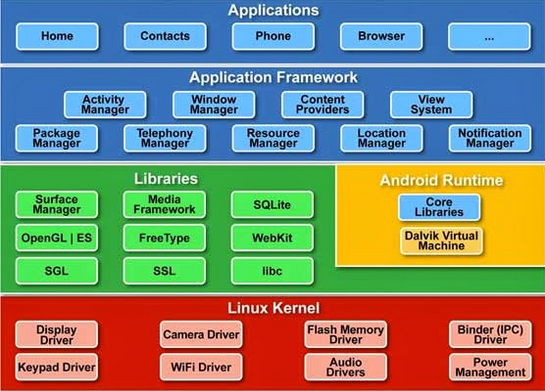
2.1.4.Kiến trúc và các thành phần

2.1.4.1.Kiến trúc tổng quát

Android bao gồm bốn thành phần sau:

* Hệ điều hành.
* Thư viện và các giao diện lập trình ứng dụng.
* Khung ứng dụng.
* Ứng dụng.

2.1.4.2.Hệ điều hành



*Hình 2.1.4.2 Kiến trúc tổng quát*

Android sử dụng nhân Linux 2.6 làm nhân cho các dịch vụ hệ thống như bảo mật, quản lý bộ nhớ, quản lý tiến trình (xử lý tiến trình, đa luồng), ngăn xếp mạng và trình điều khiển thiết bị (giao tiếp USB, giao tiếp hồng ngoại, không đây, v.v…). Nhân Linux này cũng có vai trò như một lớp trừu tượng giữa phần cứng và phần mềm.

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.

2.1.4.3.Các thành phần của nhân Linux



*Hình 2.1.4.3 Các thành phần của nhân Linux Kernel*

* 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 về.
* Bluetooth Driver : Điều khiển thiết bị phát và thu sóng Bluetooth.
* USB Driver : Điều khiển bàn phím.
* Wifi Driver : Chịu trách nhiệm về việc thu phát sóng wifi.
* Audio Driver : Điều khiển các bộ thu phí phát âm thanh, giải mã các tính hiệu dạng audio thành tín hiệu số và ngược lại.
* Power Management : Giám sát việc tiêu thụ điện năng.
* M-system Driver : Quản lý việc đọc ghi… lên các thiết bị nhớ như thẻ SD, flash.

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.

2.1.4.4.Library – Thư viện



*Hình 2.1.4.4 Libraries*

Android cung cấp một số các APIs cho phát triển ứng dụng. Danh sách các API cơ bản sau được cung cấp bởi tất cả các thiết bị trên nền Android:

* android.util: Gói tiện ích cơ bản bao gồm nhiều lớp mức thấp như là các lớp quản lý (List, Stack…) lớp xử lý chuỗi, lớp xử lý XML.
* android.graphics: Cung cấp các lớp đồ họa mức thấp thực hiện các chức năng đồ họa, màu, vẽ cơ bản.
* android.database: Cung cấp các lớp mức thấp bắt buộc cho việc điều khiển cursor khi làm việc với các cơ sở dữ liệu.
* android.content: Các giao tiếp lập trình nội dung được dùng để quản lý truy cập dữ liệu và xuất bản bằng cách cung cấp các dịch vụ thao tác với tài nguyên, Content Provider, và các gói.
* android.view: View là lớp giao diện người dùng cơ bản nhất. Tất cả giao diện người dùng được tạo ra đều phải sử dụng một tập các View để cung cấp cho các thành phần tương tác người dùng.
* android.widget: Xây dựng dựa trên gói View. Những lớp widget những thành phần giao diện được tạo sẵn được sử dụng để tạo nên giao diện người dùng. Các widget bao gồm danh sách, nút bấm, hộp nhập, các kiểu trình bày (layout).
* com.google.android.maps: Bộ API mức cao cung cấp truy cập đến điều khiển bản đồ sẵn trong Android từ ứng dụng được xây dựng. Bao gồm cả lớp MapView cũng như Overlay và MapController để tương tác với bản đồ bên trong ứng dụng.
* android.provider: Để tạo thuận lợi cho người phát triển truy cập đến các Content Provider tiêu chuẩn (như là dữ liệu danh bạ), gói cung cấp (Provider) bao gồm các lớp cho phép truy cập đến cơ sở dữ liệu chuẩn trong tất cả các bản phân phối Android.
* android.telephony: Các API điện đàm cung cấp khả năng tương tác trực tiếp với tầng điện thoại trong các thiết bị, cho phép tạo, nhận, theo dõi các cuộc gọi, tình trạng các cuộc gọi và tin nhắn SMS.
* android.webkit: Gói WebKit cung cấp các API để làm việc với các nội dung Web-based bao gồm một lơp WebView để tạo ra giao diện web, nhúng trong ứng dụng và một trình quản lý cookie.

Cùng với các API của Android, còn có một tập các thư viện C/C++ như:

* **OpenGL**: Thư viện dùng để tạo ra các đồ họa 3D dựa vào chuẩn OpenGLES 1.0 API.
* **FreeType**: Hỗ trợ xử lý bitmap và font vector.
* **GGL**: Thư viện cơ bản, dùng để cung cấp các engine đồ họa 2D.
* **Libc**: Thư viện C chuẩn, được tối ưu cho các thiết bị Linux-based.
* **SQLite Engine**: Cơ sở dữ liệu quan hệ gọn nhẹ, dùng để lưu trữ dữ liệu của ứng dụng.
* **SSL**: Hỗ trợ sử dụng giao thức mã hóa Secure Sockets Layer trong bảo mật truyền thông Internet.

Ngoài các thư viện chuẩn của Android, để đáp ứng tiêu chí phù hợp với nhiều thiết bị khác nhau, Android còn có thể có các API phụ thuộc thiết bị như android.location, android.media, android.opengl, android.hardware, android.bluetooth, android.net.wifi, và android.telephony.

2.1.4.5.Android Runtime



*Hình 2.1.4.5 Android Runtime*

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 lỗi (Core Library), chứa các lớp như JAVA IO, Collections, File Access. Thứ hai là một máy ảo java(Dalvik Virtual Machine).

Mặc dù cũng được viết từ ngôn ngữ Java nhưng một ứng dụng Java của hệ điều hành Android không được chạy bằng JRE(Java Runtime Enviroment) của Sun (nay là Oracle)(JVM) mà là chạy bằng máy ảo Dalvik (máy ảo dalvik được đặt tên sau khi một nhà phát triển phần mềm của Google tạo ra nó sau khi đi thăm một ngôi làng ở nơi tổ tiên oong sinh sống) do Google phát triển, máy ảo davik thực thi cái tập tin davik(dex), khác với java là chuyển sang dạng bytecode. Định dạng này được tối ưu hóa cho bộ nhớ tối thiểu.

2.1.4.6.Application Framework



*Hình 2.1.4.6 Application Framework*

Kiến trúc của Android khuyến khích khái niệm thành phần sử dụng lại, cho phép công bố và chia sẻ các Activity, Service, dữ liệu, với các ứng dụng khác với quyền truy cập được quản lý bởi khai báo.

Cơ chế đó cho phép người lập trình tạo ra một trình quản lý danh bạ hoặc trình quay số điện thoại mà có các thành phần sngười khác có thể tạo mới giao diện và mở rộng chức năng thay vì tạo lại chúng.

Những dịch vụ sau là những dịch vụ kiến trúc cơ bản nhất của tất cả các ứng dụng, cung cấp một framework cho mọi mọi phần mềm được xây dựng:

* **Actitvity Manager**: Điều khiển vòng đời của các Activity bao gồm cả quản lý các tầng Activity.
* **Views**: Được sử dụng để tạo lập các giao diện người dùng cho các Activity.
* **Notification Mamager**: Cung cấp một cơ chế cố định và quy củ cho việc gửi các thông báo đến người dùng.
* **Content Provider**: Cho phép ứng dụng chia sẻ dữ liệu giữa các ứng dụng
* **Resource Manager** : Hỗ trợ các thành phần không thuộc mã nguồn như là chuỗi ký tự đồ họa được đặt bên ngoài.

2.1.4.7.Các thành phần của một ứng dụng Android

Một ứng dụng trên Android được cấu thành từ bốn thành phần cơ bản sau:

* Activities.
* Services.
* Broadcast Receivers.
* Content Provider.

Các thành phần này không nhất thiết phải có mặt đầy đủ trong ứng dụng. Chúng ta có thể xem các thành phần nào được sử dụng trong ứng dụng bằng việc xem khai báo trong file AndroidManifest.xml.

2.1.4.7.1.Activity

* Khái niệm

Một hoạt động là một giao diện người dùng trực quan mà người dùng có thể thực hiện trên đó mỗi khi được kích hoạt. Một ứng dụng có thể có nhiều hoạt động và chúng có thể gọi qua lại lẫn nhau. Mỗi Activity là một dẫn xuất của lớp android.app.Activity.

Mỗi hoạt động có một cửa sổ để vẽ lên. Thông thường cửa sổ này phủ đầy màn hình, ngoài ra nó cũng có thể có thêm các cửa sổ con khác như là hộp thoại…Nội dung của cửa sổ của hoạt động được cung cấp bởi một hệ thống cấp bậc các View (là đối tượng của lớp Views).

* Vòng đời hoạt động (circle activities)

Các hoạt động trong hệ thống được quản lý bởi một cấu trúc dữ liệu ngăn xếp. Khi có một hoạt động được khởi tạo, nó được đẩy vào trong ngăn xếp, chuyển sang trạng thái thực thi và hoạt trộng trước đó sẽ chuyển sang trạng thái chờ. Hoạt động này chỉ trở lại trang thái kích hoạt khi mà hoạt động vừa khởi tạo kết thúc việc thực thi.

Một Activity có 3 trạng thái chính:

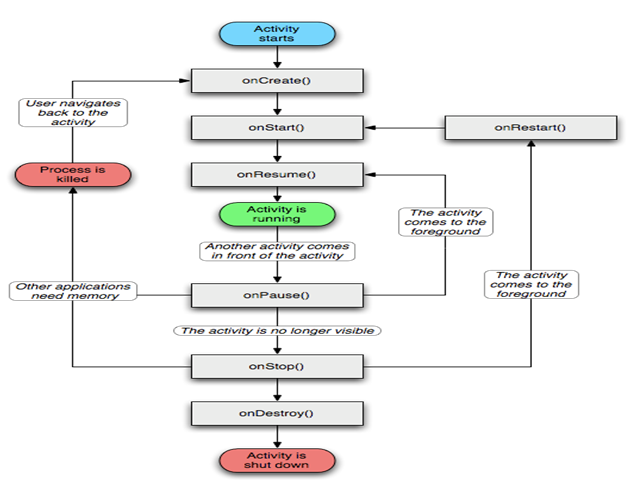
* Active hoặc running khi nó ở trên nhất màn hình và nhận tương tác người dùng.
* Paused khi Activity không còn là trọng tâm trên màn hình nhưng vẫn hiện thị trước người dùng.
* Stopped khi một Activity hoàn toàn bị che khuất, nó sẽ rơi vào trạng thái Stopped. Tuy nhiên, nó vẫn còn lưu trữ toàn bộ thông tin trạng thái. Và nó thường bị hệ thống đóng lại khi có tình trạng thiếu bộ nhớ.

Khi chuyển giữa các trạng thái, ứng dụng sẽ gọi các hàm callback ứng với các bước chuyển:

* void onCreate(Bundle savedInstanceState)
* void onStart()
* void onRestart()
* void onResume()
* void onPause()
* void onStop()
* void onDestroy()

Biểu đồ sau mô tả trạng thái trong vòng đời của một hoạt động. Hình chữ nhật thể hiện các phương thức Callback mà chúng ta có thể khai báo để gọi thực thi một số thao tác khi hoạt động chuyển sang trạng thái khác (phương thức Callback là phương thức được gọi lại bởi một phương thức khác khi có một sự kiện xảy ra). Các trạng thái chính của một hoạt động được thể hiện bởi các hình viên thuốc.

Vòng đời của một hoạt động có thể được thể hiện trong những quá trình sau:



*Hình 1.4.7.1 Vòng đời của một Activity*

Toàn bộ thời gian sống của một hoạt động bắt đầu từ lời gọi đầu tiên tới phương thức onCreate(Bundle) tới lời gọi phương thức onDestroy(). Trong quá trình này, một hoạt động sẽ khởi tạo lại tất cả các tài nguyên cần sử dụng trong phương thức onCreate() và giải phóng chúng khi phương thức onDestroy() được thực thi.

Thời gian sống có thể nhìn thấy của một hoạt động bắt đầu từ lời gọi tới phương thức onStart(), cho tới khi phương thức onStop() của nó được thực thi. Toàn bộ các tài nguyên đang được sử dụng bởi hoạt động vẫn tiếp tục được lưu giữ, người dùng có thể thấy giao diện nhưng không tương tác được với hoạt động do trong qua trình này hoạt động không ở trạng thái chạy tiền cảnh.

Thời gian sống tiền cảnh của một hoạt động là quá trình bắt dầu từ khi có lời gọi tới phương thức onResume() và kết thúc bằng lời gọi tới phương thức onPause(). Trong thời gian này, hoạt động chạy ở tiền cảnh và có thể tương tác với người dùng.

2.1.4.7.2.Service

Một dịch vụ (Service) là các đoạn mã được thực thi ngầm bởi hệ thống mà người sử dụng không thấy được. Mỗi service đều được mở rộng từ lớp cơ sở là service trong gói android.app. Có thể kết nối tới hoặc kích hoạt một Service thông qua interface mà Service đưa ra.Ví dụ như một chương trình chơi nhạc, sẽ có vài hoạt động cho phép người dùng duyệt danh sách các bài hát và lựa chọn bài nào để phát. Tuy nhiên, chức năng chơi nhạc không được thiết kế như một hoạt động bởi chúng ta sẽ muốn chuyển qua cửa sổ khác, như khi soạn tin nhắn thì bài nhạc vẫn tiếp tục được chơi. Trong trường hợp này, ứng dụng chơi nhạc sẽ khởi tạo một dịch vụ bằng cách sử dụng phương thức:

Context.startService().

Một ứng dụng có thể dễ dàng thực hiện liên kết tới một dịch vụ đang chạy (thậm chí khởi động nếu nó chưa thực thi) bằng phương thức Context.bindService(). Khi đó dịch vụ này sẽ cung cấp cho ứng dụng cơ chế để giao tiếp với chúng thông qua giao diện gọi là IBinder (đối với dịch vụ chơi nhạc có thể cho phép dừng hoặc chuyển qua bài nhạc kế tiếp).

* Vòng đời của một dịch vụ được hiểu là quá trình hoạt động từ khi nó được tạo ra cho tới khi bị loại khỏi hệ thống. Có hai cách thức để một dịch vụ có thể được chạy trong hệ thống.
* Khi hệ thống có lời gọi tới phương thức Context.startService(). Trong trường hợp này, dịch vụ sẽ được thực hiện liên tục cho tới khi hệ thống gọi phương thức Context.stopService().

Khi các ứng dụng gọi phương thức Context.bindService() để tạo kết nối với dịch vụ (dịch vụ sẽ được khởi tạo nếu tại thời điểm đó nó đang không hoạt động). Ứng dụng sẽ nhận được một đối tượng IBinder do dịch vụ trả lại để có thể gọi các phương thức Callback phù hợp để truy cập tới các trạng thái của dịch vụ. Nếu do lời gọi Context.bindService() mà dịch vụ được khởi tạo thì nó sẽ được thực thi cho tới khi nào kết nối trên (tức là đối tượng IBinder) vẫn còn tồn tại.



*Hình 4.7.2 Sơ đồ chuyển trạng thái của Service*

2.1.4.7.3. Bộ nhận quảng bá (Broadcast Receivers)

Bộ nhận quảng bá là một thành phần không làm gì cả nhưng nó nhận và phản hồi lại các thông báo quảng bá. Nhiều quảng bá có nguồn gốc từ mã hệ thống, ví dụ thông báo thay đổi múi giờ, pin yếu, ảnh đã chụp hay thay đổi ngôn ngữ. Các ứng dụng có thể khởi động quảng bá, ví dụ để các ứng dụng khác biết rằng dữ liệu đã được tải về xong trên thiết bị và sẵn sàng sử dụng.

Một ứng dụng có thể có bất kỳ số lượng bộ nhận quảng bá nào để nhận những thông báo quan trọng với nó. Tất cả các bộ nhận quảng bá được kế thừa từ lớp BroadcastReceiver.

Bộ nhận quảng bá không có giao diện. Tuy nhiên, chúng có thể khởi động một hoạt động để đáp lại thông tin mà nó nhận được, hay chúng có thể sử dụng NotificationManager để thông báo người dùng biết. Các thông báo có thể được sự chú ý của người dùng theo các cách các nhau như là sáng màn hình, rung thiết bị, bật âm thanh nào đấy… Thông thường, chúng đặt thông báo trên thanh trạng thái, nơi người dùng có thể nhận được thông báo.

2.1.4.7.4.Content Provider

Các ứng dụng có thể lưu trữ dữ liệu của mình trong các tập tin hoặc sử dụng cơ sở dữ liệu SQLite sẵn có v.v… Content Provider có chức năng cung cấp một tập hợp các phương thức cho phép một ứng dụng có thể lưu trữ và lấy dữ liệu được quản lý bởi content provider đó.

Content Provider là một đặc trưng riêng của Android, nhờ đó mà các ứng dụng có thể chia sẻ dữ liệu với nhau một cách dễ dàng.

2.1.4.7.5.Các Intent

Content Provider được kích hoạt khi chúng được gọi từ một ContentResolver. Ba thành phần khác (hoạt động, dịch vụ và bộ nhận quảng bá) được kích hoạt bởi thông điệp không đồng bộ từ các Intent. Một Intent là một đối tượng có kiểu Intent chứa nội dung của thông điệp. Với các hoạt động và dịch vụ, nó gọi tên hành động được yêu cầu và xác định URI của dữ liệu tác động tới ở giữa. Ví dụ, nó có thể truyền tải một yêu cầu cho một hoạt động hiển thị một ảnh cho người dùng hay cho phép người dùng sửa văn bản. Với bộ nhận quảng bá, đối tượng Intent gọi tên của hành động được thông báo. Ví dụ, bộ nhận quảng bá có thể thông báo các phần nó quan tâm là nút chụp ảnh đã được bấm.

Có vài phương thức cho việc kích hoạt mỗi thành phần:

Một hoạt động được khởi chạy thông qua một đối tượng Intent Context.startActivity() hay Activity.startActivityForResult(). Hoạt động đáp lại có thể theo dõi Intent được tạo ra đó bằng phương thức getIntent() và cập nhật thông qua phương thức setIntent(Intent). Android gọi phương thức onNewIntent() để bỏ qua các Intent đến trước nó.

Một hoạt động thường bắt đầu hoạt động khác. Nếu nó muốn trả lại kết quả hoạt động nó đã khởi chạy, nó sẽ gọi phương thức:

startActivityForResult() thay cho phương thức startActivity(). Ví dụ, nếu nó khởi chạy một hoạt động mà cho phép người dùng lấy một ảnh, nó có thể muốn lấy kết quả của ảnh được chọn. Kết quả được trả về trong một đối tượng Intent thông qua phương thức onActivityResult().

Một dịch vụ được bắt đầu thông qua một đối tượng Intent là Context.startService(). Android gọi phương thức onStart() của dịch vụ và thông qua đối tượng Intent của nó.

Tương tự, một Intent có thể thông qua Context.bindService() để thiết lập một kết nối liên tục giữa các thành phần và dịch vụ đích. Dịch vụ nhận đối tượng Intent qua lời gọi onBind() (nếu dịch vụ chưa được chạy, bindService() có thể chọn bắt đầu nó). Cho ví dụ, một hoạt động có thể thiết lập kết nối với dịch vụ chơi nhạc đề cập ở phần trước để nó có thể cung cấp cho người dùng giao diện sử dụng để điều khiển chơi lại. Hoạt động sẽ gọi bindService để thiết lập kết nối và sau đó gọi phương thức đã định nghĩa bởi dịch vụ để áp dụng chơi lại ca khúc.

Một ứng dụng có thể khởi tạo một quảng bá thông qua đối tượng Intent bằng phương thức như:

Context.setBroadcast(),Context.setOrderedBroadcast()và Context.sendStickyBroadcast(). Android chuyển những Intent tới tất cả các bộ nhận quảng bá nào quan tâm bằng việc gọi phương thức onReceive() của nó.

2.1.4.7.6.Manifest

Trước khi có thể khởi chạy một ứng dụng thành phần, nó phải xem ứng dụng bao gồm những thành phần nào. Thêm nữa, các ứng dụng khai báo các thành phần của nó trong một tập tin khai báo để đóng gói lại vào trong gói Android (tập tin .apk chứa các mã nguồn, tập tin và tài nguyên).

Tập tin này có cấu trúc của tập tin XML và luôn có tên là AndroidManifest.xml trong mọi ứng dụng. Tập tin này thực hiện một số chức năng như thêm và khai báo các thành phần của ứng dụng, tên các thư viện ứng dụng cần liên kết tới (ngoài thư viện chuẩn của Android) và xác định các quyền cho ứng dụng.

Tuy nhiên, nhiệm vụ chính của tập tin khai báo là khai báo các thành phần của ứng dụng. Một hoạt động có thể được khai báo như sau:

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

<manifest . . . >

<application . . . >

<activity android:name="com.example.project.FreneticActivity"

android:icon="@drawable/small\_pic.png"

android:label="@string/freneticLabel"

. . . >

</activity>

. . .

</application>

</manifest>

Thuộc tính “name” của phần tử <activity> là tên các lớp con lớp Activity đã được cài đặt, thuộc tính “icon” và “label” trỏ đến tập tin tài nguyên chứa biểu tượng và nhãn được hiển thị cho người dùng.

Các thành phần khác được khai báo theo cách tương tự: <service> dịch vụ, <receiver> bộ nhận quảng bá và <provider> content provider. Các hoạt động, dịch vụ và content provider có thể cùng được khai báo trong tập tin khai báo hoặc có thể được tạo tự động trong mã (như đối tượng BroadcastReceiver) và được đăng ký với hệ thống bằng cách gọi Context.registerReceiver().

2.1.4.7.7.Bọ lọc Intent

Một đối tượng Intent có thể có tên rõ ràng trong thành phần đích. Nếu có, Android sẽ tìm thành phần đó (dựa trên khai báo trong tập tin khai báo) và kích hoạt nó. Nhưng nếu đích có tên không rõ ràng, Android phải xác định thành phần nào thích hợp nhất để đáp lại intent. Nó thực hiện so sánh đối tượng Intent với bộ lọc intent trong các đích có khả năng. Một thành phần của bộ lọc intent cho Android biết loại intent thành phần nào cần xử lý. Giống các thông tin cơ bản, chúng được khai trong tập tin khai báo. Đây là một ví dụ để thêm hai bộ lọc intent cho một hoạt động.

1. <intent-filter . . . >
2. <action android:name=
3. "android.intent.action.MAIN" />
4. <category android:name=
5. "android.intent.category.LAUNCHER" />
6. </intent-filter>
7. <intent-filter . . . >
8. <action android:name=
9. "com.example.project.BOUNCE" />
10. <data android:mimeType="image/jpeg" />
11. <category android:name=
12. "android.intent.category.DEFAULT" />
13. </intent-filter>

Bộ lọc đầu tiên – là sự kết hợp của hành động “android.intent.action.Main” và danh mục:

“android.intent.category.LAUNCHER” – là một ví dụ phổ biến. Nó đánh dấu lại một Intent và mô tả lại cho bộ khởi chạy ứng dụng, đặt vào trong danh mục các ứng dụng trên điện thoại. Nói theo cách khác, hoạt động làm lối vào cho ứng dụng sẽ được khởi chạy, người dùng sẽ thấy khi chọn khởi chạy ứng dụng.

Bộ lọc thứ hai được khai báo để hoạt động có thể thực thi trên một kiểu dữ liệu đặc biệt.

Một thành phần có thể có bao nhiêu số bộ lọc intent tùy thích, mỗi một cái lại được khai báo khác nhau cho các khả năng. Nếu nó không có một bộ lọc nào, nó có thể được được kích hoạt bởi các intent khác gọi đến.

Để tạo và đăng ký một bộ nhận quảng bá trong mã nguồn, bộ lọc intent tạo thẳng một đối tượng giống đối tượng IntentFilter. Tất cả các bộ lọc khác đều được cài đặt trong tập tin khai báo.

**2.2.Ngôn ngữ lập trình PHP**

2.2.1.Tổng quan

PHP (PHP: Hypertext Preprocessor) là một công nghệ đang nhận được sự quan tâm rất lớn từ thế giới lập trình. Do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ thực hiện và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

Khi sử dụng PHP, người dùng sẽ có được tốc độ nhanh hơn nhiểu so với các ngôn ngữ kịch bản khác, bởi PHP là phần mềm mã nguồn mở, được hỗ trợ bởi nhiều lập trình viên giỏi, có số lượng người dùng đông đảo. Ngoài ra, PHP chạy được trên nhiều hệ điều hành khác nhau.

Rút ngằn thời gian phát triển: PHP cho phép tách phần HTML code và phần script, do đó có thể độc lập giữa công việc phát triển mã và thiết kế. Điều này vừa giúp lập trình viên dễ dàng hơn vừa có thể làm cho chương trình mềm dẻo hơn trong việc thay đổi giao diện.

PHP không chỉ là phần mềm mã nguồn mở, mà còn thực sự miễn phí kể cả sử dụng cho mục đích thương mại. Do là phần mềm mã nguồn mở, các lỗi của PHP được công khai và nhanh chóng được sửa chữa bởi nhiều chuyên gia.

Tốc độ: nhờ vào sức mạnh của Zend Engine, khi so sánh PHP với ASP có thể thấy PHP vượt trội hơn ở một số test, vượt trội hơn tốc độ biên dịch.

Tính khả chuyển: PHP được thiết kế để chạy trên nhiều nền tảng khác nhau, có thể làm việc với nhiều phần mềm máy chủ, cơ sở dữ liệu. PHP rất ổn định và tương hợp, PHP đã vận hành khá ổn định trên các hệ điều hành kể cả Unix, Windows…Đồng thời nó cũng nối với một số máy chủ như IIS hay Apache.

Bảng 2.2 Bảng So Sánh Các Công Nghệ Lập Trình Web

|  |  |  |
| --- | --- | --- |
| PHP | ASP.NET | JSP |
| Tốc độ xử lý nhanh, hiệu quả cao | Tốc độ xử lý nhanh, hiệu quả cao | Hơi chậm |
| Chi phí giá thành thấp (ngôn ngữ free không tốn chi phí mua bản quyền) | Chi phí giá thành cao (do một phần phải mua bản quền) | Chi phí giá thành cao |
| Thời gian code và triển nhanh, đơn giản | Thời gian code và triển khai hơi phức tạp, chậm hơn PHP | Thời gian code và triển khai hơi phức tạp, chậm hơn PHP |
| Số lượng nhà cung cấp hosting nhiều, dễ lựa chọn | Số lượng nhà cung cấp hosting không nhiều, vì vậy khó cho việc lựa chọn | Số lượng nhà cung cấp hosting ít, khó tìm |
| Số nhà cung cấp website nhiều, khả năng chọn lựa một trang web phù hợp tốt hơn | Số nhà cung cấp website nhiều, khả năng chọn lựa một trang web phù hợp tốt hơn | Số nhà cung cấp website ít, khả năng chọn lựa để xây dựng một trang web phù hợp rất khó |
| Khả năng mở rộng và phát triển dễ dàng và nhanh chóng | Mở rộng và phát triển website dẽ dàng | Khả năng mở rộng và phát triển website khó khăn |
| Các công cụ và công nghệ hỗ trợ phong phú, đa dạng. Phát triển web trên nền web 2.0. Sử dụng công nghệ Ajaxlàm cho quá trình duyệt web nhanh chóng và thân thiện hơn đối với người dùng. | Các công cụ và công nghệ hỗ trợ phong phú, đa dạng. Phát triển web trên nền web 2.0. Sử dụng công nghệ Ajax làm cho quá trình duyệt web nhanh chóng và thân thiện hơn đối với người dùng. | Công cụ và công nghệ hỗ trợ ít, khó tìm. |

2.2.2.Slim Framework

Slim là một framework PHP, chủ yếu sử dụng cho các ứng dụng web và API theo cấu trúc REST (Representational state transfer).Ưu điểm của Slim là kích thước nhỏ gọn, dễ dàng cài đặt và triển khai.

**2.3.Hệ quản trị cơ sở dữ liệu**

Hệ quản trị cơ sở dữ liệu là phần mềm hay hệ thống được thiết kế để quản trị một cơ sở dữ liệu (CSDL). Cụ thể, các chương trình thuộc loại này hỗ trợ khả năng lưu trữ, sửa chữa, xóa và tìm kiếm thông tin trong một cơ sở dữ liệu. Có rất nhiều loại hệ quản trị CSDL khác nhau: từ phần mềm nhỏ chạy trên máy tính cá nhân cho đến những hệ quản trị phức tạp chạy trên một hoặc nhiều siêu máy tính.

Tuy nhiên, đa số hệ quản trị CSDL trên thị trường đều có một đặc điểm chung là sử dụng ngôn ngữ truy vấn theo cấu trúc. Các hệ quản trị CSDL phổ biến được nhiều người biết đến là [MySQL](http://vi.wikipedia.org/wiki/MySQL), [Oracle](http://vi.wikipedia.org/wiki/Oracle), [PostgreSQL](http://vi.wikipedia.org/wiki/PostgreSQL), [SQL Server](http://vi.wikipedia.org/w/index.php?title=SQL_Server&action=edit&redlink=1) …Phần lớn các hệ quản trị CSDL kể trên hoạt động tốt trên nhiều hệ điều hành khác nhau như Linux, Unix và MacOS. Ngoại trừ SQL Server của Microsoft chỉ chạy trên hệ điều hành Windows.

MySQL đã trở thành CSDL mã nguồn mở phổ biến nhất trên thế giới vì cơ chế xử lý nhanh và ổn định của nó, sự đáng tin cậy cao và dễ sử dụng. Nó được sử dụng mọi nơi – ngay cả châu Nam Cực - bởi các nhà phát triển Web riêng lẻ cũng như rất nhiều các tổ chức lớn nhất trên thế giới để tiết kiệm thời gian và tiền bạc cho những Website có dung lượng lớn, phần mềm đóng gói – bao gồm cả những nhà đi đầu trong lĩnh vực công nghệ như Yahoo!, Google, Nokia, YouTube…

**Ưu điểm:**

* Mã nguồn mở, miễn phí.
* Tốc độ truy cập nhanh, ổn định, mức độ bảo mật cao, dễ sử dụng.
* Hoạt động trên mọi hệ điều hành.
* Dễ quản lý.
* Cộng đồng hỗ trợ rộng lớn.

**Nhược điểm:**

* Bảo mật chưa bằng SQL Server hay Oracle.
* Chưa hỗ trợ XML.

SQLite là phần mềm quản lý cơ sở dữ liệu (DBMS) tương tự như Mysql, PostgreSQL... Đặc điểm của SQLite là gọn, nhẹ, đơn giản. Chương trình gồm 1 file duy nhất vỏn vẹn chưa đến 500kB, không cần cài đặt, không cần cấu hình hay khởi động mà có thể sử dụng ngay. Dữ liệu database cũng được lưu ở một file duy nhất. Không có khái niệm user, password hay quyền hạn trong SQLite database.  
 SQLite không thích hợp với những hệ thống lớn nhưng ở quy mô vừa tầm thì SQLite phát huy uy lực và không hề yếu kém về mặt chức năng hay tốc độ. Với các đặc điểm trên SQLite được sử dụng nhiều trong việc phát triển, thử nghiệm v..v.. và là sự lưa chọn phù hợp chọn phù hợp cho việc phát triển ứng dụng.

**Tổng quan:**

* Cơ sở dữ liệu dạng quan hệ (RDBMS), hỗ trợ chuẩn SQL-92
* Sử dụng dưới dạng thư viện nhúng, không chạy ở theo kiểu server độc lập
* Hỗ trợ các ngôn ngữ phổ biến: C, C++, C#, Basic, Perl, Ruby, Python, PHP, Java .
* Không cần chỉ định kiểu dữ liệu (SQLite is typeless)
* Hỗ trợ mã UTF8
* Hỗ trợ command line
* Hỗ trợ transaction
* Hỗ trợ view
* Hỗ trợ C extensions

**2.4.WebService**

2.4.1.Tổng quan

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. Dịch vụ Web 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. Một dịch vụ Web được tạo nên bằng cách lấy các chức năng và đóng gói chúng sao cho các ứng dụng khác dễ dàng nhìn thấy và có thể truy cập đến những dịch vụ mà nó thực hiện, đồng thời có thể yêu cầu thông tin từ dịch vụ Web khác. Nó bao gồm các mô đun độc lập cho hoạt động của khách hàng và doanh nghiệp và bản thân nó được thực thi trên server.

Trước hết, có thể nói rằng ứng dụng cơ bản của Dịch vụ Web là tích hợp các hệ thống và là một trong những hoạt động chính khi phát triển hệ thống. Trong hệ thống này, các ứng dụng cần được tích hợp với cơ sở dữ liệu (CSDL) và các ứng dụng khác, người sử dụng sẽ giao tiếp với CSDL để tiến hành phân tích và lấy dữ liệu. Trong thời gian gần đây, việc phát triển mạnh mẽ của thương mại điện tử và B2B cũng đòi hỏi các hệ thống phải có khả năng tích hợp với CSDL của các đối tác kinh doanh (nghĩa là tương tác với hệ thống bên ngoài – bên cạnh tương tác với các thành phần bên trong của hệ thống trong doanh nghiệp).

2.4.2.Đặc điểm

2.4.2.1.Đặc điểm

* Dịch vụ Web cho phép client và server tương tác được với nhau ngay cả trong những môi trường khác nhau. Ví dụ, đặt Web server cho ứng dụng trên một máy chủ chạy hệ điều hành Linux trong khi người dùng sử dụng máy tính chạy hệ điều hành Windows, ứng dụng vẫn có thể chạy và xử lý bình thường mà không cần thêm yêu cầu đặc biệt để tương thích giữa hai hệ điều hành này.
* Phần lớn kĩ thuật của Dịch vụ Web được xây dựng dựa trên mã nguồn mở và được phát triển từ các chuẩn đã được công nhận, ví dụ như XML.
* Là sự kết hợp của việc phát triển theo hướng từng thành phần với những lĩnh vực cụ thể và cơ sở hạ tầng Web, đưa ra những lợi ích cho cả doanh nghiệp, khách hàng, những nhà cung cấp khác và cả những cá nhân thông qua mạng Internet.
* Một ứng dụng khi được triển khai sẽ hoạt động theo mô hình client-server. Nó có thể được triển khai bởi một phần mềm ứng dụng phía server ví dụ như PHP, Oracle Application server hay Microsoft.Net…
* Ngày nay dịch vụ Web đang rất phát triển, những lĩnh vực trong cuộc sống có thể áp dụng và tích hợp dịch vụ Web là khá rộng lớn như dịch vụ chọn lọc và phân loại tin tức (hệ thống thư viện có kết nối đến web portal để tìm kiếm các thông tin cần thiết); ứng dụng cho các dịch vụ du lịch (cung cấp giá vé, thông tin về địa điểm…), các đại lý bán hàng qua mạng, thông tin thương mại như giá cả, tỷ giá hối đoái, đấu giá qua mạng…hay dịch vụ giao dịch trực tuyến (cho cả B2B và B2C) như đặt vé máy bay, thông tin thuê xe…
* Các ứng dụng có tích hợp dịch vụ Web đã không còn là xa lạ, đặc biệt trong điều kiện thương mại điện tử đang bùng nổ và phát triển không ngừng cùng với sự lớn mạnh của Internet. Bất kì một lĩnh vực nào trong cuộc sống cũng có thể tích hợp với dịch vụ Web, đây là cách thức kinh doanh và làm việc có hiệu quả bởi thời đại ngày nay là thời đại của truyền thông và trao đổi thông tin qua mạng. Do vậy, việc phát triển và tích hợp các ứng dụng với dịch vụ Web đang được quan tâm phát triển là điều hoàn toàn dễ hiểu.

2.4.2.2.Ưu và nhược điểm

**Ưu điểm:**

* Dịch vụ Web cung cấp khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau.
* Sử dụng các giao thức và chuẩn mở. Giao thức và định dạng dữ liệu dựa trên văn bản (text), giúp các lập trình viên dễ dàng hiểu được.Nâng cao khả năng tái sử dụng.
* Thúc đẩy đầu tư các hệ thống phần mềm đã tồn tại bằng cách cho phép các tiến trình/chức năng nghiệp vụ đóng gói trong giao diện dịch vụ Web.
* 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.

**Nhược điểm:**

* Những thiệt hại lớn sẽ xảys ra vào khoảng thời gian chết của Dịch vụ Web, 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 dịch vụ Web khiến người dùng khó nắm bắt.
* Phải quan tâm nhiều hơn đến vấn đề an toàn và bảo mật.

2.4.3.JSON trong WebService

2.4.3.1.JSON là gì?

* JSON là cú pháp để lưu trữ và trao đổi thông tin văn bản. Cũng giống như XML.
* JSON là nhỏ hơn so với XML, và nhanh hơn và dễ dàng hơn để phân tích (parse).
* JSON sử dụng cú pháp JavaScript để mô tả đối tượng dữ liệu, nhưng JSON là ngôn ngữ và nền tảng độc lập. Trình phân tích cú pháp(parsers) JSON và các thư viện JSON tồn tại cho nhiều ngôn ngữ lập trình khác nhau.
* JSON là viết tắt của JavaScript Object Notation.
* JSON là định dạng trao đổi dữ liệu văn bản dung lượng nhẹ.
* JSON là ngôn ngữ độc lập.
* JSON được “tự mô tả” và dễ hiểu.

2.4.3.2.Cú pháp JSON

Các quy luật cú pháp JSON : Cú pháp JSON là một tập hợp con của cú pháp đối tượng JavaScript.

* Dữ liệu là các cặp tên/giá trị (name/values)
* Dữ liệu được phân cách bằng dấu phẩy
* Đối tượng nằm trong cặp ngoặc nhọn ‘{}’
* Dấu ngoặc vuông giữ mảng ‘[]’

**Chương 3 : PHÂN TÍCH THIẾT KẾ**

**3.1.Chức năng chính**

3.1.1.Chức năng của server

* Quản lý khách hàng
* Quản lý khách sạn
* Quản lý đơn đặt phòng
* Quản lý phản hồi

3.1.2.Chức năng của client

* Người dùng có thể đăng ký tài khoản , đăng nhập hệ thống
  + Đăng ký : Xử lý input của người dùng , input sẽ được gửi lên server để xử lý tiếp. Nhận response từ server và thông báo cho người dùng.
  + Đăng nhập : Xử lý input của người dùng ,check database local , nếu không tồn tại thì gửi yêu cầu đăng nhập của người dùng lên server và nhận response từ server.
* Người dùng có thể tìm kiếm khách sạn xung quanh khu vực hiện tại
* Người dùng có thể xem lịch sử đặt phòng tại khách sạn
* Người dùng có thể nhận được coupon ưu đãi thông qua việc đặt phòng thường xuyên
* Người dùng có thể gửi phản hồi , thắc mắc thông qua form
* Người dùng có thể mời bạn bạn bè cài app bằng cách chia sẻ thông qua tin nhắn,mạng xã hội,…

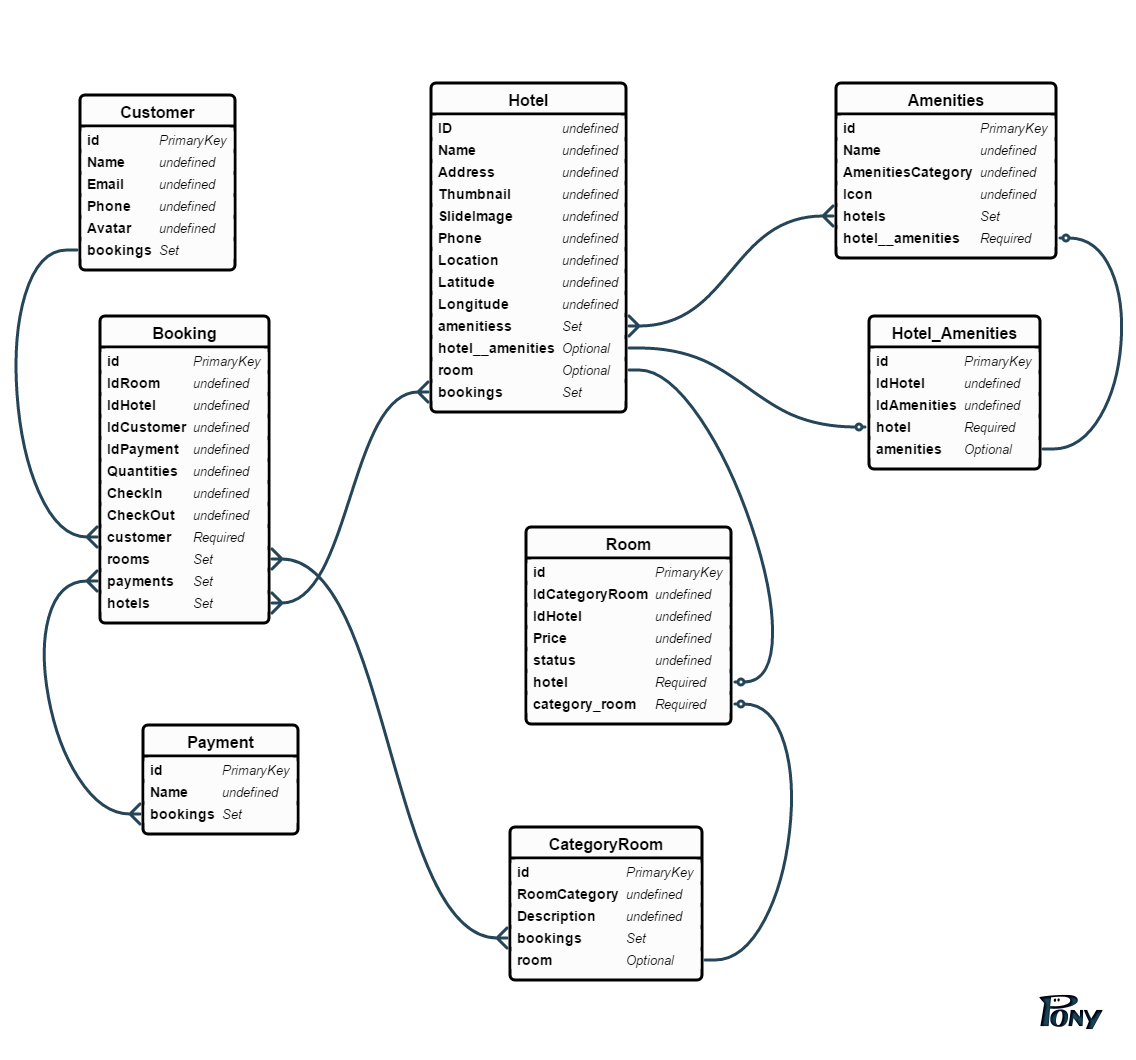
3.1.2.Mô hình hoạt động Client – Server

**3.2.Phân tích thiết kế**

3.2.1.Thiết kế cơ sở dữ liệu

3.2.1.1.Mô hình tổng quát

Mô hình cơ sở dữ liện tổng quát thể hiện mối quan hệ giữa các bảng với nhau, thể hiện những thực thể trong cơ sở dữ liệu.



Hình 3.2.1.1 Sơ đồ tổng quát mối liên hệ giữa các bảng trong cơ sở dữ liệu

* Diễn giải sơ đồ :

|  |  |  |
| --- | --- | --- |
| **Thực thể** | **Mô tả** | **Thuộc tính – Chú thích** |
| **Customer** | *Khách hàng* | * ID: id khách hàng * Name: tên đăng nhập * Password: mật khẩu * Email: địa chỉ email * Phone: số điện thoại * Avatar: ảnh đại diện |
| **Hotel** | *Khách sạn* | * ID : id khách sạn * Name : tên khách sạn * Address : địa chỉ khách sạn * Thumnail : ảnh đại diện khách sạn * Slide image : ảnh các phòng của khách sạn * Phone : số điện thoại liên hệ * Location : khu vực * Latitude : kinh độ của khách sạn * Longitude : vĩ độ của khách sạn |
| **Amenities** | *Tiện ích và dịch vụ của khách sạn* | * ID : id tiện ích * Name : tên tiện ích * AmenitiesCategory : loai tiện ích * Icon : ảnh mô tả tiện ích |
| **CategoryRoom** | *Các Loại phòng* | * ID : id loại phòng * RoomCategory : loại phòng * Description : mô tả |
| **Room** | *Lưu trữ phòng của các khách sạn* | * ID : id phòng * IDCategoryRoom : id loại phòng * IDHotel : id khách sạn * Price : giá phòng * Status : trạng thái phòng |
| **Hotel\_Amenities** | *Liên hệ khách sạn và tiện ích khách sạn* | * ID : id liên hệ * IDHotel : id khách sạn * IDAmenitites : id tiện ích |
| **Payment** | *Các hình thức thanh toán* | * ID : id hình thức thanh toán * Name : tên hình thức thanh toán |

3.2.1.2.Thiết kế chi tiết

* Bảng Customer :

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| ID | int | X | Đặc trưng cho từng khách hàng,phân biệt khách hàng với nhau |
| Name | varchar |  | Tên khách hàng |
| Password | varchar |  | Mật khẩu đăng nhập |
| Email | varchar |  | Email khách hàng |
| Phone | inn |  | Số điện thoại của khách hàng |
| Avatar | varchar |  | Ảnh đại diện của khách hàng |

* Bảng Hotel :

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| ID | int | X | Đặc trưng cho từng khách sạn,phân biệt khách sạn với nhau |
| Name | varchar |  | Tên khách sạn |
| Address | varchar |  | Địa chỉ khách sạn |
| Thumbnail | varchar |  | Ảnh đại diện khách sạn |
| Slide Image | varchar |  | Ảnh các phòng khách sạn |
| Phone | varchar |  | Số điện thoại khách sạn |
| Location | varchar |  | Địa điểm khách sạn |
| Latitude | varchar |  | Kinh độ của khách sạn |
| Longitude | varchar |  | Vĩ độ của khách sạn |

* Bảng Room:

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| ID | int | X | Đặc trưng cho từng phòng,phân biệt phòng với nhau |
| IdCategoryRoom | int |  | Id loại phòng |
| IdHotel | int |  | Id Khách sạn |
| Price | int |  | Giá phòng |
| Status | int |  | Trạng thái của phòng |

* Bảng Amenities:

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| ID | int | X | Đặc trưng cho từng tiện tích ,phân biệt tiện ích với nhau |
| Name | int |  | Tên tiện ích |
| IdAmenitiesCategory | int |  | Id Loại tiện ích |
| Icon | int |  | Ảnh mô tả tiện ích |

* Bảng CategoryRoom

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| ID | int | X | Đặc trưng cho từng loại phòng,phân biệt loại phòng với nhau |
| RoomCategory | varchar |  | Tên loại phòng |
| Description | varchar |  | Mô tả loại phòng |

* Bảng Payment

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| ID | int | X | Đặc trưng cho từng hình thức thanh toán,phân biệt các hình thức thanh toán với nhau |
| Name | varchar |  | Tên của hình thức thanh toán |

* Bang Hotel\_Amenities

|  |  |  |  |
| --- | --- | --- | --- |
| **Cột** | **Kiểu dữ liệu** | **Khóa chính** | **Ý nghĩa** |
| ID | int | X | Đặc trưng cho từng liên hệ giữa tiện ích với khách sạn có tiện ích |
| IdHotel | int |  | Id Khách sạn |
| IdAmenities | int |  | ID Tiện ích |