

LỜI CẢM ƠN

Trong đợt thực hiện đồ án tốt nghiệp vừa qua, em đã nhận được sự hướng dẫn, giúp đỡ và động viên tận tình từ nhiều phía. Tất cả những điều đó đã trở thành một động lực rất lớn giúp tôi có thể hoàn thành tốt đồ án. Với tất cả sự cảm kích và trân trọng, tôi xin được gửi lời cảm ơn đến tất cả mọi người.

Trước tiên cho em được gửi lời cảm ơn đến lãnh đạo Khoa Công nghệ thông tin – Đại học Bách Khoa Đà Nẵng đã tạo điều kiện cho tôi được thực hiện đồ án tốt nghiệp cũng như cung cấp tất cả các cơ sở vật chất và trang thiết bị có thể cho em trong thời gian vừa qua. Xin cảm ơn thầy Phan Thanh Tao đã tận tình hướng dẫn giúp đỡ trong suốt thời gian tôi tham gia làm đồ án.

Em xin chân thành cảm ơn các thầy cô giáo của trường Đại học Bách Khoa Đà Nẵng, xin cảm ơn quý thầy cô đã tận tình dạy bảo, giúp đỡ tôi trong suốt thời gian tôi học tại trường cũng như trong thời gian tôi làm đồ án tốt nghiệp vừa qua.

Xin trân trọng cảm ơn!

LỜI CAM ĐOAN

Tôi xin cam đoan :

- 1 Những nội dung trong báo cáo này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của thầy Th.s Phan Thanh Tao.*
- 2 Mọi tham khảo dùng trong báo cáo này đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.*
- 3 Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, tôi xin chịu hoàn toàn trách nhiệm.*

Sinh viên

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

MỤC LỤC

CHƯƠNG 1 : GIỚI THIỆU CHUNG.....	1
1.1. Lý do chọn đề tài	1
1.2. Mục tiêu và nhiệm vụ.....	1
1.2.1. Mục tiêu	1
1.2.2. Nhiệm vụ	1
1.3. Đối tượng và phạm vi nghiên cứu.....	2
1.3.1 Phương pháp nghiên cứu lý thuyết.....	2
1.3.2. Phương pháp nghiên cứu thực nghiệm	2
1.4. Kết quả dự kiến	2
1.5. Bố cục luận văn.....	2
CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT	3
2.1. Giới thiệu hệ điều hành Android.	3
2.1.1. Tổng quan cấu trúc của Android.....	4
2.1.2. Sơ lược các phiên bản Android.....	5
2.1.3. Các thành phần cơ bản của ứng dụng Android	6
2.2. Lập trình trên hệ điều hành Android.....	11
2.2.1. Ngôn ngữ JAVA	11
2.2.2. Ngôn ngữ XML.....	14
2.3. Máy ảo Dalvik và máy ảo SDK	20
2.3.1. Máy ảo Dalvik.....	20
2.3.2. Android SDK	20
2.3.3. Android Emulator	22
CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	24
3.1. Phân tích yêu cầu	24
3.1.1. Mô tả bài toán.....	24
3.1.2. Yêu cầu chức năng	24
3.1.3. Yêu cầu phi chức năng.....	24
3.1.4. Yêu cầu giao diện	24
3.2. Phân tích và thiết kế hệ thống.....	24
3.2.1. Biểu đồ ca sử dụng	24
3.2.2. Biểu đồ hoạt động.....	33
3.2.3. Cơ sở dữ liệu SQLite.....	38
CHƯƠNG 4 : TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ	41
4.1. Các công cụ , môi trường phát triển	41

4.1.1. Yêu cầu.....	41
4.1.2. Cài đặt	41
4.1.3. Công cụ viết chương trình.....	41
4.1.4. Công cụ viết báo cáo	41
4.2. Demo chương trình	42
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	46
TÀI LIỆU THAM KHẢO	1

DANH SÁCH HÌNH VẼ

Hình 1: Giới thiệu hệ điều hành Android	3
Hình 2: Các thành phần của hệ điều hành Android	5
Hình 3: Logo và tên các phiên bản Android	6
Hình 4: Mô hình hoạt động của Activity	10
Hình 5: Cách biên dịch chương trình truyền thống	12
Hình 6: Cơ chế biên dịch chương trình của Java	13
Hình 7: Activity Stack.	17
Hình 8: Chu kỳ sống của một Activity.	17
Hình 9: Các sự kiện trong một chu kỳ sống.	19
Hình 10: Môi trường phát triển ứng dụng Android trong Eclipse	22
Hình 11: Android Emulator	23
Hình 12 : Sơ đồ user case tổng quát	27
Hình 13 : Sơ đồ usecase tổng quát.....	27
Hình 14 : Sơ đồ use-case chọn ngôn ngữ	28
Hình 15 : Sơ đồ user case thiết lập	28
Hình 16 : Sơ đồ user case tra từ.....	29
Hình 17 : Sơ đồ user case xem chi tiết từ	30
Hình 18 : Sơ đồ user case lưu từ.....	30
Hình 19 : Sơ đồ user case phát âm.....	31
Hình 20 : Sơ đồ user case thêm từ mới.....	31
Hình 21 : Sơ đồ user cập nhập lại từ.....	32
Hình 22: Sơ đồ user case xóa từ	33
Hình 23 : Sơ đồ hoạt động đăng nhập.....	33
Hình 24 : Sơ đồ hoạt động tra từ.....	34
Hình 25 : Sơ đồ hoạt động thêm từ mới	34
Hình 26 : Sơ đồ hoạt động phát âm	35
Hình 27 : Sơ đồ hoạt động lưu từ.....	35
Hình 28 : Sơ đồ hoạt động xóa từ đã lưu	36
Hình 29: Sơ đồ hoạt động xem chi tiết từ.....	36
Hình 30 : Sơ đồ hoạt động xóa từ	37
Hình 31 : Sơ đồ hoạt động cập nhập lại nghĩa của từ	37

Hình 32 : Mối quan hệ giữa các bảng	40
Hình 33: Màn hình khởi động.....	42
Hình 34 : Giao diện tra từ	42
Hình 35 : Giao diện xem chi tiết nghĩa của từ	43
Hình 36 : Giao diện đăng nhập	43
Hình 37 : Giao diện chọn ngôn ngữ cần quản lý	44
Hình 38 : Giao diện thêm từ	44
Hình 39 : Giao diện danh sách các từ	45
Hình 40 : Giao diện sửa , xóa từ	45

DANH SÁCH BẢNG BIỂU

Bảng 1: Mô tả các Use-Case trong sơ đồ tổng quát.....	28
Bảng 2 : Mô tả Use-Case chọn ngôn ngữ	28
Bảng 3: Mô tả user case cài đặt	28
Bảng 4 : Mô tả user case tra từ	29
Bảng 5 : Mô tả user case xem chi tiết nghĩa của từ	30
Bảng 6 : Mô tả user case lưu từ	31
Bảng 7 : Mô tả user case phát âm	31
Bảng 8 : Mô tả user case thêm từ mới	32
Bảng 9 : Mô tả user case cập nhập lại nghĩa của từ.....	32
Bảng 10 : Mô tả user case xóa từ.....	33
Bảng 11 : Bảng dữ liệu Admin	38
Bảng 12 : Bảng dữ liệu Word saved	38
Bảng 13 : Bảng dữ liệu anh việt	38
Bảng 14 : Bảng dữ liệu việt anh.....	39
Bảng 15 : Bảng dữ liệu việt nhật	39
Bảng 16 : Bảng dữ liệu việt pháp.....	39
Bảng 17 : Bảng dữ liệu bách khoa toàn thư.....	40

CHƯƠNG 1 : GIỚI THIỆU CHUNG

1.1. Lý do chọn đề tài

Hiện nay, công nghệ thông tin đang phát triển hết sức mạnh mẽ. Điều này đồng nghĩa với việc nhu cầu về công nghệ của con người cũng ngày càng cao. Từ việc phải làm việc và giải trí trên máy tính, giờ đây các thiết bị di động cũng đang dần đáp ứng được các nhu cầu của con người và càng được sử dụng nhiều hơn do tính linh hoạt, tiện lợi và di động cao.

Có rất nhiều nền tảng di động phổ biến như IOS, Android, WindowPhone... nhưng Android của Google đang là hệ điều hành có hướng phát triển mạnh nhất. Do Android là một hệ điều hành mở nên nhận được rất nhiều sự hỗ trợ từ cộng đồng mạng trên khắp thế giới.

Xã hội ngày càng phát triển, đòi hỏi về trình độ học vấn, hiểu biết càng cao thì việc chúng ta biết được nhiều ngôn ngữ, thứ tiếng sẽ là một lợi thế rất lớn. Tra cứu từ điển trên sách, hay máy tính đôi khi còn bất tiện, không di động thì tra cứu Từ điển trên chính chiếc điện thoại hay thiết bị di động chạy hệ điều hành Android sẽ vô cùng tiện lợi, có thể tra cứu từ điển mọi lúc mọi nơi. Từ những lý do trên, em đã suy nghĩ và quyết định lựa chọn đề tài : **“XÂY DỰNG ỨNG DỤNG TỪ ĐIỂN ĐA NGỮ TRÊN HỆ ĐIỀU HÀNH ANDROID”**

1.2. Mục tiêu và nhiệm vụ

1.2.1. Mục tiêu

- Nắm được quy trình phát triển phần mềm
- Biết vận dụng các kiến thức lý thuyết vào thực tế để làm ra sản phẩm cụ thể
- Biết cách tổ chức , triển khai , lên kế hoạch để làm ra sản phẩm cụ thể
- Biết cách đánh giá chất lượng sản phẩm của mình sau khi hoàn thành

1.2.2. Nhiệm vụ

- Tìm hiểu tài liệu hướng dẫn cách thức lập trình trên Hệ Điều Hành Android ,
SQLite

- Lập bảng kế hoạch chi tiết các công việc cần làm
- Tham dự đầy đủ , nộp phiếu thông đồ án đúng tiến độ
- Viết báo cáo và bảo vệ sản phẩm theo đúng kế hoạch đã đặt ra

1.3. Đối tượng và phạm vi nghiên cứu

Đề tài này sẽ kết hợp với hai phương pháp nghiên cứu đó là :

1.3.1 Phương pháp nghiên cứu lý thuyết

- Nghiên cứu tài liệu , ngôn ngữ và công nghệ có liên quan
- Tổng hợp các tài liệu

1.3.2. Phương pháp nghiên cứu thực nghiệm

- Phân tích yêu cầu thực tế của bài toán và xây dựng các bước phân tích hệ thống để hỗ trợ việc lập trình , xây dựng ứng dụng
- Đánh giá kết quả đạt được

1.4. Kết quả dự kiến

Hoàn thành công việc thiết kế và xây dựng chương trình từ điển đa ngôn ngữ đúng thời gian và đầy đủ các chức năng và giao diện thân thiện với người dùng.

1.5. Bố cục luận văn

Trong luận văn này em sẽ trình bày cụ thể những chi tiết liên quan đến ứng dụng đã thực hiện, luận văn được tổ chức thành những chương chính như sau:

Chương 1 : Giới thiệu chung

Chương 2: Cơ sở lý thuyết

Chương 3: Phân tích và thiết kế hệ thống

Chương 4 : Xây dựng ứng dụng

Kết luận và hướng phát triển

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

Trong chương này em sẽ trình bày về những cơ sở lý thuyết cơ bản về lập trình ứng dụng trên nền tảng Android đã áp dụng trong quá trình xây dựng đề tài.



Hình 1: Giới thiệu hệ điều hành Android

2.1. Giới thiệu hệ điều hành Android.

Android là hệ điều hành trên các thiết bị số di động như smartphone và tablet (và hiện nay là cả trên một số đầu phát HD, HD Player) phát triển bởi Google và dựa trên nền tảng Linux. Trước đây, Android được phát triển bởi công ty liên hợp Android. Vào năm 2005, Google đã mua lại Android nằm trong chiến lược tiến vào thị trường di động. Google muốn Android trở thành nền tảng “mở” và miễn phí. Do vậy hầu hết mã nguồn của Android được phát hành theo giấy phép mã nguồn mở Apache nên bất cứ ai muốn sử dụng Android có thể download toàn bộ mã nguồn. Hơn nữa các nhà sản xuất phần cứng có thể thêm các thành phần mở rộng độc quyền của hãng để tạo ra các sản phẩm khác nhau. Chính mô hình phát triển đơn giản này khiến Android trở nên rất hấp dẫn. Android có một cộng đồng những nhà phát triển rất lớn viết các ứng dụng cho hệ điều hành của mình.

Hiện tại có khoảng hơn 250,000 ứng dụng cho Android OS và vào khoảng hơn 375,000 ứng dụng đã được đệ trình, điều này khiến Android trở thành hệ điều hành di động có môi trường phát triển lớn thứ 2. Các nhà phát triển viết ứng dụng cho Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào ngày 5 tháng 11 năm 2007 gắn với sự thành lập của liên minh thiết bị cầm tay mã nguồn mở, bao gồm 78 công ty phần cứng, phần mềm và viễn thông nhằm mục đích tạo nên một chuẩn mở cho điện thoại di động trong tương lai. Google công bố hầu hết các mã nguồn của

Android theo bản cấp phép Apache. Hệ điều hành Android bao gồm 12 triệu dòng mã; 3 triệu dòng XML, 2.8 triệu dòng mã C, 2.1 triệu mã Java và 1.75 triệu dòng mã C++.

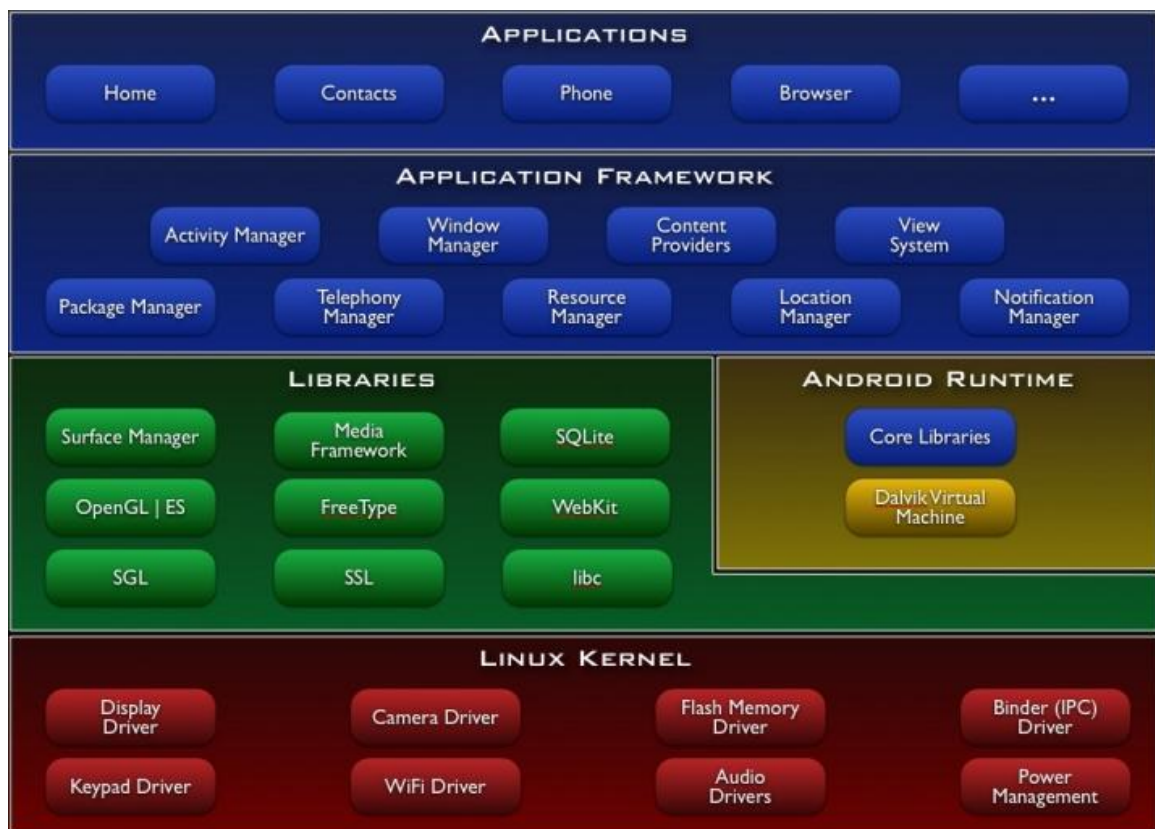
2.1.1. Tổng quan cấu trúc của Android

Nói một cách đơn giản thì Android bao gồm ba thành phần:

- Một hệ điều hành mở, miễn phí cho các thiết bị di động.
- Một nền tảng mã nguồn mở cho phát triển các ứng dụng.
- Thiết bị chạy hệ điều hành Android và các ứng dụng Android.

Chi tiết hơn thì Android gồm các thành phần quan trọng sau:

- Một thiết kế phần cứng cho thiết bị di động hỗ trợ bộ phần mềm.
- Một nhân hệ điều hành Linux cung cấp giao diện mức thấp với phần cứng, quản lý bộ nhớ và kiểm soát tiến trình. Tất cả đều tối ưu hoá cho thiết bị di động.
- Thư viện mã nguồn mở cho phát triển ứng dụng bao gồm SQLite, Webkit, OpenGL và quản lý đa phương tiện.
- Một hệ thống thực thi và lưu trữ cho ứng dụng Android bao gồm máy ảo Dalvik và các thư viện cơ bản cung cấp chức năng đặc thù cho Android. Hệ thống thực thi được thiết kế nhỏ gọn và hiệu quả khi sử dụng trên các thiết bị di động.
- Một kiến trúc ứng dụng cung cấp các dịch vụ hệ thống cho tầng ứng dụng bao gồm quản lý cửa sổ, quản lý vị trí, cung cấp nội dung và các cảm biến.
- Một kiến trúc giao diện người dùng để lưu trữ và chạy các ứng dụng.
- Một số ứng dụng được cài đặt sẵn được coi như một phần của bộ ứng dụng.
- Một bộ phát triển ứng dụng bao gồm các công cụ và tài liệu.



Hình 2: Các thành phần của hệ điều hành Android

2.1.2. Sơ lược các phiên bản Android

Android là một nền tảng mới và phát triển nhanh. Các phiên bản thông thường được phát hành sau khoảng 6 tháng. Sau đây là danh sách các phiên bản chính và sự thay đổi lớn trong mỗi phiên bản Android.

- ❖ OHA giới thiệu Android (5-11-2007).
- ❖ Phát hành SDK (12-11-2007).
- ❖ Apple pie - Android 1.0 (23/9/2008): Phát hành phiên bản Android 1.0 trên điện thoại Android đầu tiên – HTC Dream (G1).
- ❖ Banana bread - Android 1.1 (9-2-2009): Phát hành phiên bản cập nhật 1.1 riêng cho T-Mobile G1.
- ❖ Cupcake - Android 1.5 (30/4/2009).
- ❖ Donut - Android 1.6 (15/9/2009).
- ❖ Éclair - Android 2.0 (26/10/2009).

- ❖ Froyo - Android 2.2 (20/5/2010).
- ❖ Gingerbread - Android 2.3 (6/12/2010).
- ❖ Honeycomb - Android 3.0 (22/2/2011).
- ❖ Ice Cream Sandwich: được tuyên bố phát hành trong quý 4 năm 2011. Đây là sự kết hợp của hai phiên bản Gingerbread và Honeycomb.

Tên gọi của các phiên bản Android đều được đặt theo các món tráng miệng:



Hình 3: Logo và tên các phiên bản Android

2.1.3. Các thành phần cơ bản của ứng dụng Android

Ứng dụng Android bao gồm nhiều thành phần liên kết với nhau được giới hạn trong kiến trúc ứng dụng. Kiến trúc ứng dụng mô tả chi tiết mỗi thành phần và sự tương tác giữa các thành phần này cũng như siêu dữ liệu về yêu cầu phần cứng.

Các thành phần sau đây tạo nên một ứng dụng Android:

- ❖ **Activity:** là lớp thể hiện ứng dụng. Mỗi màn hình trong ứng dụng là một mở rộng của lớp Activity. Các Activity sử dụng các View tạo thành giao diện đồ

họa tương tác người dùng để thể hiện thông tin và phản hồi hành động của người dùng.

- ❖ **Service:** là thành phần chạy ẩn, cập nhật nguồn dữ liệu và thể hiện các Activity, kích hoạt các Notification. Các Service thường thực hiện các xử lý thông thường như tiếp tục Activity sau khi các Activity này không hoạt động hoặc hiển thị.
- ❖ **Content Provider:** là thành phần lưu trữ dữ liệu chia sẻ. Content Provider quản lý và chia sẻ cơ sở dữ liệu của ứng dụng. Chúng được coi như là phương tiện chia sẻ dữ liệu giữa các ứng dụng.
- ❖ **Intent:** là cấu trúc truyền thông điệp giữa các ứng dụng. Chúng ta có thể sử dụng Intent để truyền một thông tin tới toàn hệ thống hay tới Activity hay Service định trước để thực thi một hoạt động. Hệ thống sau đó sẽ xác định đối tượng nào sẽ thực hiện các hoạt động phù hợp.
- ❖ **Broadcast Receiver:** là nơi nhận các Intent. Chúng ta có thể tự tạo một Broadcast Receiver để ứng dụng có thể nhận được các Intent phù hợp với bộ lọc đặt trước. Broadcast Receiver tự động bắt đầu khi ứng dụng phản hồi với Intent nhận được và biến ứng dụng trở thành hướng sự kiện một cách hoàn hảo.
- ❖ **Widget:** thành phần trực quan của ứng dụng có thể thêm vào màn hình chính. Là một dạng Broadcast Receiver, Widget cho phép người dùng tạo ra các ứng dụng động, tương tác được nhúng ngay trên màn hình chính.
- ❖ **Notification:** là thành phần thông báo cho người dùng. Hộp thoại thông báo cho người dùng mà không gián đoạn các Activity hiện thời. Đây là kỹ thuật thông báo cho người dùng nằm trong một Service hay một Broadcast Receiver. Ví dụ như khi nhận được một tin nhắn hay cuộc gọi đến có thể thông báo cho người dùng bằng nhấp nháy đèn hay bằng âm thanh...

Một ứng dụng không nhất thiết phải gồm tất cả các thành phần trên nhưng để có một giao diện với người dùng thì ít nhất ứng dụng đó phải có một Activity. Mã đã được biên dịch của các thành phần của ứng dụng, tài nguyên liên quan như thư viện,

hình ảnh và các dữ liệu cần thiết khác được đóng gói trong một file apk duy nhất chính là dạng file thực thi của ứng dụng Android.

Như đã trình bày ở trên, Activity là thành phần quan trọng nhất và đóng vai trò chính trong việc xây dựng ứng dụng Android. Hệ điều hành Android quản lý ứng dụng dưới dạng stack : khi một Activity mới được khởi tạo, nó được xếp lên đầu stack và trở thành running activity, các Activity trước đó sẽ bị tạm dừng và chỉ hoạt động trở lại khi Activity mới được giải phóng.

Activity có 4 trạng thái:

- ❖ **Active (running):** Activity đang hiển thị trên màn hình (foreground).
- ❖ **Pause:** Activity vẫn hiển thị (visible) nhưng không thể tương tác. VD: một activity mới xuất hiện hiển thị giao diện đè lên trên activity cũ, nhưng giao diện này vẫn nhỏ hơn giao diện activity cũ. Do đó ta vẫn thấy được một phần giao diện của activity cũ nhưng không tương tác được với nó.
- ❖ **Stop:** Activity bị thay thế hoàn toàn bởi Activity mới sẽ tiến đến trạng thái stop.
- ❖ **Killed:** Khi hệ thống thiếu bộ nhớ, nó sẽ giải phóng các tiến trình theo nguyên tắc ưu tiên ở trên. 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 trạng thái trước đó.

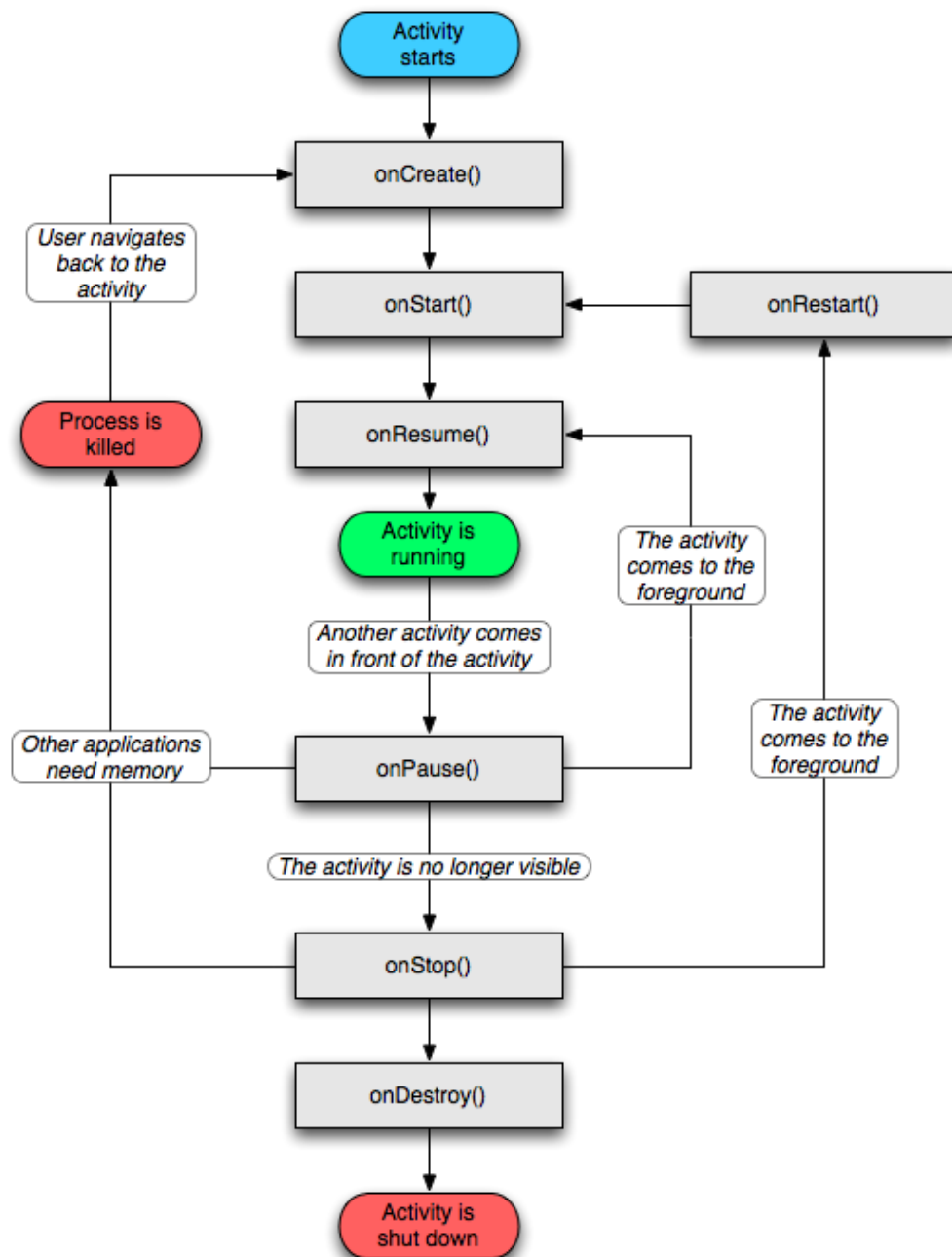
Tuỳ từng sự kiện mà các phương thức được gọi để đảm bảo trạng thái các Activity được hoạt động đúng. Chi tiết từng phương thức và sự kiện thay đổi trạng thái tương ứng:

- ❖ **onCreate():** Phương thức này khởi tạo Activity. Nó có thể bỏ qua trạng thái cũ để trở lại hoạt động. Phương thức tiếp theo luôn là onStart().
- ❖ **onRestart():** Được gọi sau khi một Activity dừng và muốn bắt đầu lại.
- ❖ **onStart():** Được gọi khi ứng dụng chuyển sang visible nhưng chưa tương tác được với người dùng.
- ❖ **onResume():** Activity có tương tác với người sử dụng và có thể nhận thông tin đầu vào. Kiểu tiến trình của ứng dụng được chuyển sang dạng hiển thị (foreground).

❖ **onPause():** Nếu ứng dụng không còn tương tác hay thiết bị không hoạt động, phương thức này được gọi và kiểu tiến trình ứng dụng sẽ được chuyển sang dạng visible. Sau khi thực hiện phương thức này, hệ thống cho phép giải phóng ứng dụng bất cứ lúc nào. Các thao tác sử dụng CPU và dữ liệu đều được lưu lại. Activity sẽ có thể được hoạt động trở lại hoặc dừng lại.

❖ **onStop():** Khi Activity không còn tương tác nữa thì kiểu tiến trình được chuyển sang chạy nền (background) và ứng dụng có thể bị giải phóng bất cứ lúc nào bởi hệ thống để lấy lại bộ nhớ. Activity sẽ bị giải phóng hoặc khởi động lại.

❖ **onDestroy():** Phương thức cuối cùng này được hệ thống gọi trước khi giải phóng ứng dụng hay ứng dụng loại bỏ Activity. Tiến trình của ứng dụng có thể chuyển sang kiểu rỗng nếu hệ thống muốn giữ lại ứng dụng để tiếp tục lại ứng dụng cho lần sau.



Hình 4: Mô hình hoạt động của Activity

2.2. Lập trình trên hệ điều hành Android

2.2.1. Ngôn ngữ JAVA

2.2.1.1. Tổng quan về ngôn ngữ JAVA

Java là một ngôn ngữ lập trình dạng lập trình hướng đối tượng (OOP). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy. Bằng cách này, Java thường chạy chậm hơn những ngôn ngữ lập trình thông dịch khác như C++, Python, Perl, PHP, C#...

Vào năm 1991, một nhóm các kỹ sư của Sun Microsystems có ý định thiết kế một ngôn ngữ lập trình để điều khiển các thiết bị điện tử như tivi, máy giặt, lò nướng,... Mặc dù C và C++ có khả năng làm việc này nhưng trình biên dịch lại phụ thuộc vào từng loại CPU.

Trình biên dịch thường phải tốn nhiều thời gian để xây dựng nên rất đắt, vì vậy để mỗi loại CPU có một trình biên dịch riêng là rất tốn kém. Do đó nhu cầu thực tế đòi hỏi một ngôn ngữ chạy nhanh, gọn, hiệu quả và độc lập thiết bị tức là có thể chạy trên nhiều loại CPU khác nhau, dưới các môi trường khác nhau. “Oak” đã ra đời và vào năm 1995 được đổi tên thành Java.

2.2.1.2. Một số tính chất ngôn ngữ JAVA

Cú pháp Java được vay mượn nhiều từ C & C++ nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn. Dùng bộ thư viện chuẩn KFC, nhiều đoạn code Java chỉ mất vài dòng trong khi C phải mất cả trang giấy.

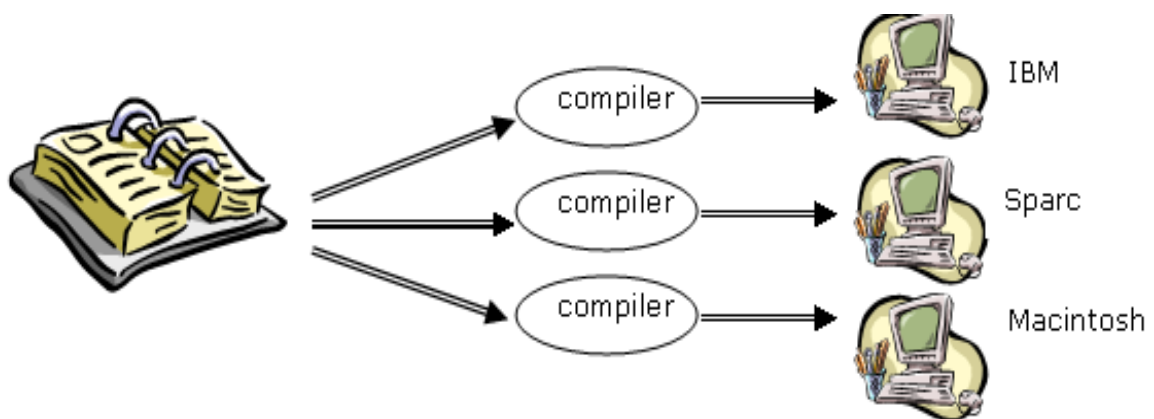
Đơn giản: Những người thiết kế mong muốn phát triển một ngôn ngữ dễ học và quen thuộc với đa số người lập trình. Do vậy Java loại bỏ các đặc trưng phức tạp của C và C++ như:

- ❖ Loại bỏ thao tác con trỏ, thao tác định nghĩa chồng toán tử.
- ❖ Không cho phép đa kế thừa mà sử dụng các giao diện.

- ❖ Không sử dụng lệnh “goto” cũng như file header (.h).
- ❖ Loại bỏ cấu trúc “struct” và “union”.

Hướng đối tượng: Java là ngôn ngữ lập trình thuần hướng đối tượng, mọi chương trình viết trên Java đều phải được xây dựng trên các đối tượng. Nếu trong C/C++ ta có thể tạo ra các hàm (chương trình con không gắn với đối tượng nào) thì trong Java ta chỉ có thể tạo ra các phương thức (chương trình con gắn liền với một lớp cụ thể). Trong Java không cho phép các đối tượng có tính năng đa kế thừa mà được thay thế bằng các giao diện (interface).

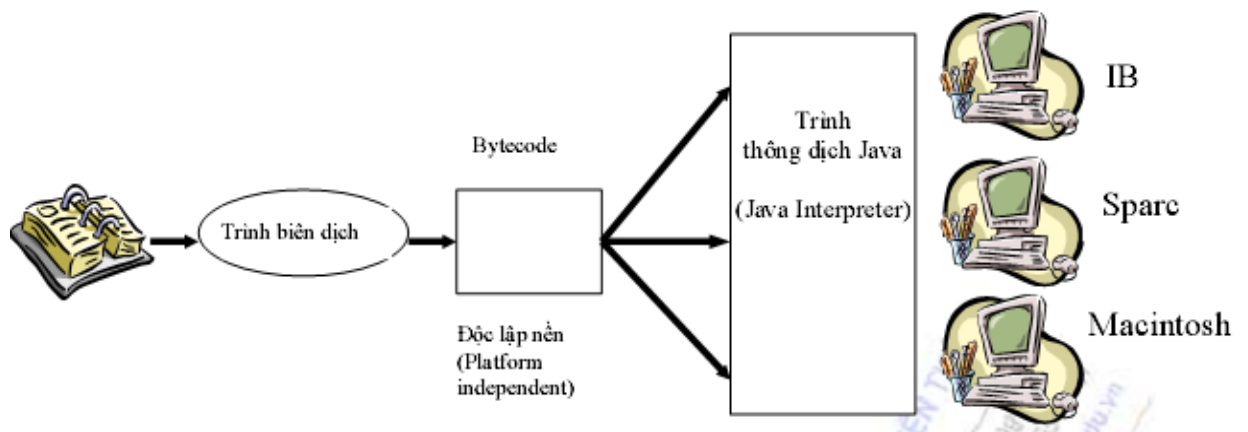
Độc lập phần cứng và hệ điều hành: Đối với các ngôn ngữ lập trình truyền thống như C/C++, phương pháp biên dịch được thực hiện như sau.



Hình 5: Cách biên dịch chương trình truyền thống

Với mỗi nền phần cứng khác nhau, có một trình biên dịch khác nhau để biên dịch mã nguồn chương trình cho phù hợp với nền phần cứng ấy. Do vậy, khi chạy trên một nền phần cứng khác bắt buộc phải biên dịch lại mã nguồn.

Đối với các chương trình viết bằng Java, trình biên dịch Javac sẽ biên dịch mã nguồn thành dạng bytecode. Sau đó, khi chạy chương trình trên các nền phần cứng khác nhau, máy ảo Java dùng trình thông dịch Java để chuyển mã bytecode thành dạng chạy được trên các nền phần cứng tương ứng. Do vậy, khi thay đổi nền phần cứng, không phải biên dịch lại mã nguồn Java.



Hình 6: Cơ chế biên dịch chương trình của Java

Manh mẽ:

- ❖ Java là ngôn ngữ yêu cầu chặt chẽ về kiểu dữ liệu.
- ❖ Kiểu dữ liệu phải khai báo tường minh.
- ❖ Java không sử dụng con trỏ và các phép toán con trỏ.
- ❖ Java kiểm tra tất cả các truy nhập đến mảng, chuỗi khi thực thi để đảm bảo rằng các truy nhập đó không ra ngoài giới hạn kích thước.
- ❖ Trong các môi trường lập trình truyền thống, lập trình viên phải tự mình cấp phát bộ nhớ, trước khi chương trình kết thúc thì phải tự giải phóng bộ nhớ đã cấp. Vấn đề có thể nảy sinh khi lập trình viên quên giải phóng bộ nhớ đã xin cấp trước đó. Trong chương trình Java, lập trình viên không phải bận tâm đến việc cấp phát bộ nhớ. Quá trình cấp phát, giải phóng được thực hiện tự động, nhờ dịch vụ thu nhặt những đối tượng không còn sử dụng nữa (garbage collection).
- ❖ Cơ chế bắt lỗi của Java giúp đơn giản hóa quá trình xử lý lỗi và hồi phục sau lỗi.

Bảo mật: Java cung cấp một môi trường quản lý thực thi chương trình với nhiều mức để kiểm soát tính an toàn:

- ❖ Ở mức thứ nhất, dữ liệu và các phương thức được đóng gói bên trong lớp. Chúng chỉ được truy xuất thông qua các giao diện mà lớp cung cấp.

- ❖ Ở mức thứ hai, trình biên dịch kiểm soát để đảm bảo mã là an toàn, và tuân theo các nguyên tắc của Java.
- ❖ Mức thứ ba được đảm bảo bởi trình thông dịch; chúng kiểm soát xem bytecode có đảm bảo các quy tắc an toàn trước khi thực thi không.
- ❖ Mức thứ tư kiểm soát việc nạp các lớp vào bộ nhớ để giám sát việc vi phạm giới hạn truy xuất trước khi nạp vào hệ thống.

Phân tán: Java được thiết kế để hỗ trợ các ứng dụng chạy trên mạng bằng các lớp mạng (java.net). Hơn nữa, Java hỗ trợ nhiều nền chạy khác nhau nên chúng được sử dụng rộng rãi như là công cụ phát triển trên Internet - nơi sử dụng nhiều nền khác nhau.

Đa luồng: Chương trình Java cung cấp giải pháp đa luồng (Multithreading) để thực thi các công việc đồng thời. Chúng cũng cung cấp giải pháp đồng bộ giữa các luồng. Đặc tính hỗ trợ đa luồng này cho phép xây dựng các ứng dụng trên mạng chạy hiệu quả.

Linh động: Java được thiết kế như một ngôn ngữ động để đáp ứng cho những môi trường mở. Các chương trình Java chứa rất nhiều thông tin thực thi nhằm kiểm soát và truy nhập đối tượng lúc chạy. Điều này cho phép khả năng liên kết mã động.

2.2.2. Ngôn ngữ XML

XML, hoặc Extensible Markup Language (ngôn ngữ đánh dấu mở rộng), là một ngôn ngữ đánh dấu mà bạn có thể sử dụng để tạo ra thẻ riêng của mình. Nó được tạo nên bởi Liên minh mạng toàn cầu nhằm khắc phục những hạn chế của HTML - ngôn ngữ đánh dấu siêu văn bản, là cơ sở của mọi trang Web. Giống như HTML, XML cũng được dựa trên SGML – Standard Generalized Markup Language. Mặc dù SGML được sử dụng trong ngành công nghiệp xuất bản trong nhiều thập kỷ, nhưng sự phức tạp của nó đều khiến những ai từng sử dụng nó mà không có cách nào khác phải thấy mệt mỏi.

Các đặc điểm của XML:

- ❖ Dễ dàng viết được các chương trình xử lý dữ liệu

- ❖ Tài liệu XML dễ đọc và có tính hợp lý cao
- ❖ XML dễ dàng được sử dụng trên Internet
- ❖ XML hỗ trợ nhiều ứng dụng
- ❖ Không đặt nặng tính hình thức trong nội dung thẻ

Các loại văn bản XML:

- ❖ **Văn bản không hợp lệ:** không theo nguyên tắc cú pháp được quy định bởi đặc tính kỹ thuật XML.
- ❖ **Văn bản hợp lệ:** tuân theo nguyên tắc cú pháp XML và quy định trong DTD hoặc lược đồ.
- ❖ **Văn bản chuẩn:** tuân theo quy tắc cú pháp XML nhưng không có DTD hoặc lược đồ.

Một tài liệu XML được chia thành hai phần chính:

- Phần khai báo: khai báo cho tài liệu XML

- ❖ khai báo phiên bản, bảng mã ký tự sử dụng trong tài liệu
- ❖ định nghĩa kiểu cho tài liệu

- Phần thân: chứa nội dung dữ liệu

- ❖ gồm một hay nhiều phần tử,
- ❖ mỗi phần tử được chứa trong một cặp thẻ
- ❖ phần tử đầu tiên là phần tử gốc (root element).

2.2.3. Cơ sở dữ liệu SQLite

SQLite là một dạng CSDL tương tự như Mysql, PostgreSQL... Đặc điểm của SQLite là gọn, nhẹ, đơn giản. Chương trình gồm 1 file duy nhất vốn 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 lựa chọn phù hợp cho những người bắt đầu học database. Hiện nay thì SQLite đã được ứng dụng vào smartphone như iPhone và Android để lưu trữ dữ liệu.

SQLite là hệ thống quản lý các quan hệ cơ sở dữ liệu (RDBMS), nó cũng được coi là:

- Mã nguồn mở.
- Tiêu chuẩn.
- Gọn nhẹ.
- Đơn lớp.

2.2.4. Chu kỳ sống của ứng dụng Android

Một tiến trình Linux gói gọn một ứng dụng Android đã được tạo ra cho ứng dụng khi codes cần được xây dựng, khởi chạy và sẽ còn chạy cho đến khi:

- Nó không phụ thuộc.
- Hệ thống cần lấy lại bộ nhớ mà nó chiếm giữ cho các ứng dụng khác.

Một sự khác thường và đặc tính cơ bản của Android là thời gian sống của tiến trình ứng dụng không được điều khiển trực tiếp bởi chính nó. Thay vào đó, nó được xác định bởi hệ thống qua một kết hợp của:

- Những phần của ứng dụng mà hệ thống biết đang chạy.
- Những phần đó quan trọng như thế nào đối với người dùng.
- Bao nhiêu vùng nhớ chiếm lĩnh trong hệ thống.

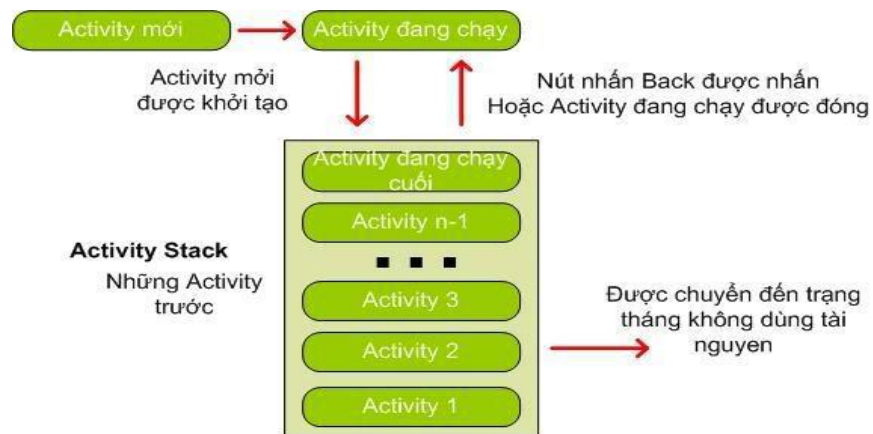
2.2.4.1. Chu kỳ sống thành phần

Các thành phần ứng dụng có một chu kỳ sống, tức là mỗi thành phần từ lúc bắt đầu khởi tạo và đến thời điểm kết thúc, đôi lúc chúng có thể là active (visible hoặc invisible) hoặc inactive.

2.2.4.2. Activity Stack

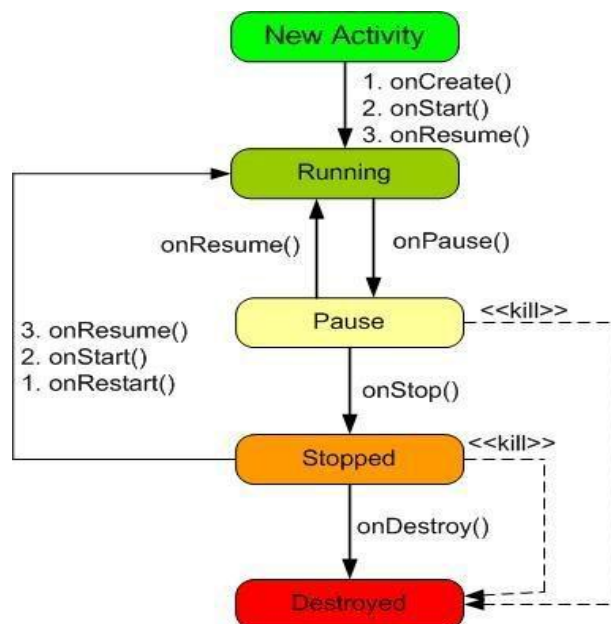
Bên trong hệ thống các activity được quản lý như một activity stack. Khi một Activity mới được start, nó được đặt ở đỉnh của stack và trở thành activity đang chạy activity trước sẽ ở bên dưới activity mới và sẽ không thấy trong suốt quá trình activity mới tồn tại.

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



Hình 7: Activity Stack.

2.2.4.3. Các trạng thái của chu kỳ sống



Hình 8: Chu kỳ sống của một Activity.

Một Activity chủ yếu có 4 chu kỳ chính sau:

- Active hoặc running: Khi Active là được chạy trên màn hình. Activity này tập trung vào những thao tác của người dùng trên ứng dụng.
- Paused: Activity là được tạm dừng (paused) khi mất focus nhưng người dùng vẫn trông thấy. 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.
- Stopped: Nếu nó hoàn toàn bao phủ bởi Activity khác. Nó vẫn còn trạng thái và thông tin thành viên trong nó. Người dùng không thấy nó và thường bị loại bỏ trong trường hợp hệ thống cần vùng nhớ cho 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.

2.2.4.4. Chu kỳ sống của ứng dụng

Trong một ứng dụng Android có chứa nhiều thành phần và mỗi thành phần đều có một chu trình sống riêng. Và ứng dụng chỉ được gọi là kết thúc khi tất cả các thành phần trong ứng dụng kết thúc. Activity là một thành phần cho phép người dùng giao tiếp với ứng dụng. Tuy nhiên, khi tất cả các Activity kết thúc và người dùng không còn giao tiếp được với ứng dụng nữa nhưng không có nghĩa là ứng dụng đã kết thúc. Bởi vì ngoài Activity là thành phần có khả năng tương tác người dùng thì còn có các thành phần không có khả năng tương tác với người dùng như là Service, Broadcast receiver. Có nghĩa là những thành phần không tương tác người dùng có thể chạy background dưới sự giám sát của hệ điều hành cho đến khi người dùng tự tắt chúng.

2.2.4.5. Các sự kiện trong chu kỳ sống của ứng dụng

Nếu một Activity được tạm dừng hoặc dừng hẳn, hệ thống có thể bỏ thông tin khác của nó từ vùng nhớ bởi việc gọi hàm finish() của nó, hoặc đơn giản giết tiến trình của nó. Khi nó được hiển thị lần nữa với người dùng, nó phải được hoàn toàn

restart và phục hồi lại trạng thái trước. Khi một Activity chuyển qua chuyển lại giữa các trạng thái, nó phải báo việc chuyển của nó bằng việc gọi hàm transition.

<code>void onCreate(Bundle savedInstanceState)</code>	<code>Void onPause()</code>
<code>void onStart()</code>	<code>Void onStop()</code>
<code>void onRestart()</code>	<code>Void onDestroy()</code>
<code>void onResume()</code>	

Hình 9: Các sự kiện trong một chu kỳ sống.

Tất cả các phương thức là những móc nối mà chúng ta có thể override để làm tương thích công việc trong ứng dụng khi thay đổi trạng thái. Tất cả các Activity bắt buộc phải có onCreate() để khởi tạo ứng dụng. Nhiều Activity sẽ cũng hiện thực onPause() để xác nhận việc thay đổi dữ liệu và mặt khác chuẩn bị dừng hoạt động với người dùng.

2.2.4.6. Thời gian sống của ứng dụng

Bảy phương thức chuyển tiếp định nghĩa trong chu kỳ sống của một Activity. Thời gian sống của một Activity diễn ra giữa lần đầu tiên gọi onCreate() đến trạng thái cuối cùng gọi onDestroy(). Một Activity khởi tạo toàn bộ trạng thái toàn cục trong onCreate(), và giải phóng các tài nguyên đang tồn tại trong onDestroy().

2.2.4.7. Thời gian hiển thị của Activity

Visible lifetime của một activity diễn ra giữa lần gọi một onStart() cho đến khi gọi onStop(). Trong suốt khoảng thời gian này người dùng có thể thấy activity trên màn hình, có nghĩa là nó không bị foreground hoặc đang tương tác với người dùng. Giữa 2 phương thức người dùng có thể duy trì tài nguyên để hiển thị activity đến người dùng.

2.2.4.8. Các hàm thực thi

- onCreate(...): hàm này được gọi khi lớp Activity được khởi tạo, dùng để thiết lập giao diện ứng dụng và thực thi những thao tác cơ bản.

- onStart(): hàm này được gọi khi lớp ứng dụng xuất hiện trên màn hình.

- onResume(): hàm được gọi ngay sau onStart hoặc khi người dùng focus ứng dụng, hàm này sẽ đưa ứng dụng lên top màn hình.

- onPause(): hàm được gọi khi hệ thống đang focus đến 1 activity trước đó.

- onStop(): hàm được gọi khi một activity khác được khởi động và focus.

- onRestart(): được gọi khi ứng dụng chuyển sang onStop(), nhưng muốn khởi động lại bằng onStart().

2.3. Máy ảo Dalvik và máy ảo SDK

2.3.1. Máy ảo Dalvik

Các ứng dụng Android và các thành phần cơ bản hầu hết được viết bằng Java. Thay vì sử dụng máy ảo Java, Android sử dụng máy ảo riêng. Máy ảo của Android được thiết kế và tối ưu cho các hệ thống nhỏ, không tương thích với máy ảo Java của Java ME. Những hệ thống nhỏ này có dung lượng RAM nhỏ, CPU chậm và hầu hết không có chỗ để trao đổi dữ liệu bù đắp cho dung lượng bộ nhớ nhỏ.

Đây là điểm khác biệt giữa Android với các hệ điều hành cho di động khác như Symbian, iOS của Apple, webOS của Palm. Ngôn ngữ lập trình được sử dụng chính là C, C++ và objective C, trong khi với webOS sử dụng chủ yếu ngôn ngữ web như Javascript hay HTML.

Trình biên dịch byte code là máy ảo Dalvik. Thay vì sử dụng dạng byte code thông thường, Dalvik có định dạng riêng để phù hợp với yêu cầu đối với thiết bị cho Android. Mã của Android nhỏ hơn so với mã của Java và file dex được tạo ra cũng nhỏ.

2.3.2. Android SDK

Android SDK là bộ công cụ phát triển phần mềm cho phép lập trình viên tạo ra các ứng dụng cho các thiết bị sử dụng nền tảng Android. Android SDK bao gồm nền tảng Android, các công cụ, chương trình mẫu và các tài liệu hỗ trợ. Nó cũng là thành phần bổ sung trong bộ phát triển Java và có thể tích hợp vào Eclipse.

Android chứa các gói hỗ trợ các chức năng tính toán cơ bản như quản lý xâu, quản lý vào ra, tính toán từ các gói cơ bản trong Java:

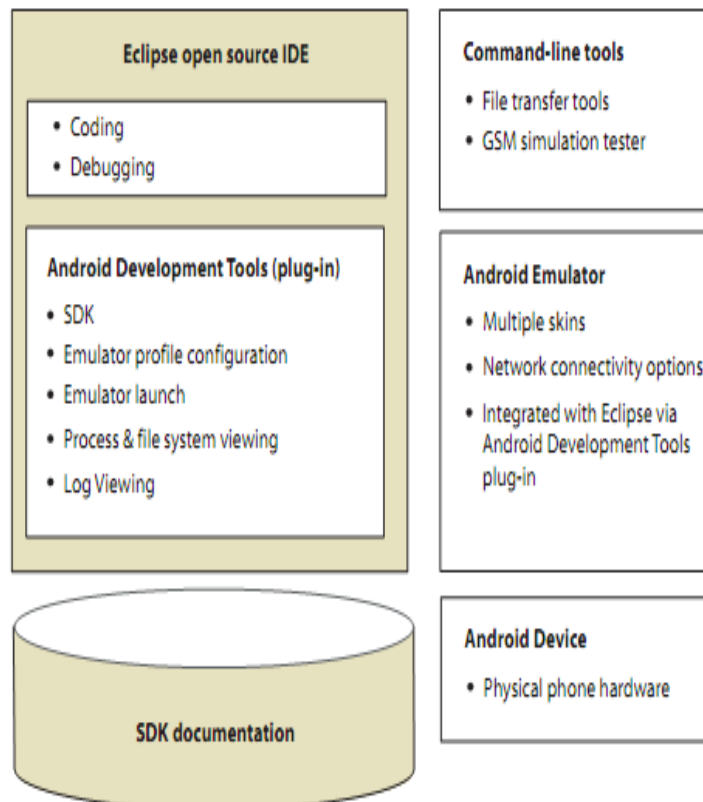
- ❖ Java.lang – lớp ngôn ngữ cơ bản của java.
- ❖ Java.io – quản lý vào ra.
- ❖ Java.net – quản lý kết nối mạng.

- ❖ Java.text – tiện ích xử lý văn bản.
- ❖ Java.math – hỗ trợ tính toán.
- ❖ Java.securcity – quản lí các thành phần bảo mật.
- ❖ Java.xml – các lớp về XML.
- ❖ org.apache – các lớp liên quan HTTP.

Ngoài ra Android có các lớp riêng như:

- ❖ android.app – truy cập mô hình ứng dụng Android.
- ❖ android.bluetooth – chức năng bluetooth của Android.
- ❖ android.content - quản lí dữ liệu trong Android.
- ❖ android.net – chứa các lớp Uri dùng để truy cập dữ liệu.
- ❖ android.graphics – các thành phần đồ họa.
- ❖ android.location – các dịch vụ liên quan đến vị trí (ví dụ như GPS).
- ❖ android.opengl – các lớp OpenGL.
- ❖ android.view – chứa các thành phần giao diện.
- ❖ android.webkit – chức năng của trình duyệt web.
- ❖ android.widget – các thành phần giao diện khác.

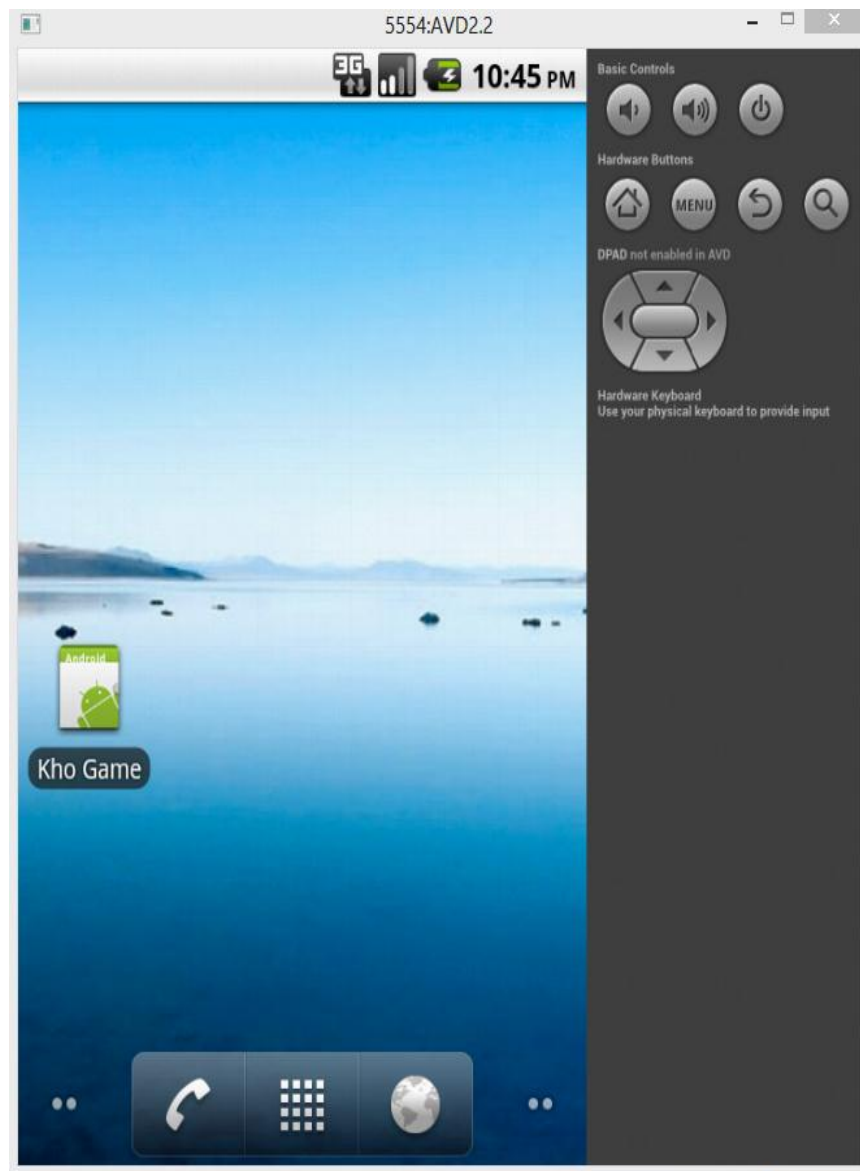
Ở đây chúng ta xem xét lập trình Android với Eclipse vì Eclipse cung cấp phong phú các thành phần hỗ trợ biên soạn java và môi trường debug mà còn có thể quản lí và điều khiển các yếu tố khác của ứng dụng Android trực tiếp từ Eclipse IDE.



Hình 10: Môi trường phát triển ứng dụng Android trong Eclipse

2.3.3. Android Emulator

Android SDK cung cấp một bộ giả lập thiết bị di động ảo. Máy ảo này cho phép chúng ta phát triển và kiểm tra các ứng dụng mà không cần một thiết bị Android thật. Máy ảo Android có đầy đủ các thành phần từ phần cứng đến phần mềm giống như một thiết bị thật ngoại trừ nó không thể thực hiện được một cuộc gọi. Máy ảo Android cho phép sử dụng chuột để định hướng hay bàn phím để tạo các sự kiện. Người dùng có thể cài đặt để kiểm thử phần mềm trên máy ảo với từng phiên bản khác nhau của Android. Khi ứng dụng được cài đặt nó có thể sử dụng các ứng dụng đã được cung cấp trong Android như nghe nhạc, xem video hay vào mạng. Máy ảo cũng có khả năng debug, truy cập vào đầu ra của nhân, giả lập các ngắt như khi nhân cuộc gọi hay tin nhắn thậm chí là các ảnh hưởng độ trễ của mạng và ngắt trên kênh truyền dữ liệu.



Hình 11: Android Emulator

CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Phân tích yêu cầu

3.1.1. Mô tả bài toán

Xây dựng ứng dụng từ điển trên hệ điều hành Android , ứng dụng có thể tra được nhiều loại ngôn ngữ , xem từ và chi tiết nghĩa của từ . Bên cạnh đó có thể thao tác thêm xóa hoặc sửa các từ .

3.1.2. Yêu cầu chức năng

Cho phép người dùng lựa chọn ngôn ngữ để tra , xem từ và nghĩa của từ đó .

Phát âm được ngôn ngữ cần tra , lưu lại các từ đã tra theo mong muốn .

Quản trị ngôn ngữ : thêm , xóa , cập nhập hoặc xóa các từ .

3.1.3. Yêu cầu phi chức năng

Ứng dụng chạy tốt khi mà cơ sở dữ liệu về từ điển có thể tăng trong tương lai

Ứng dụng có thể thực hiện chính xác kết quả và nhanh chóng với dữ liệu lớn

3.1.4. Yêu cầu giao diện

Chương trình yêu cầu giao diện phải:

- Thân thiện, đơn giản, dễ sử dụng.
- Hiện thị thông tin đầy đủ và chính xác.

3.2. Phân tích và thiết kế hệ thống

3.2.1. Biểu đồ ca sử dụng

Xác định tác nhân

- ❖ Người sử dụng
- ❖ Admin

Các trường hợp sử dụng

- ❖ Lựa chọn ngôn ngữ
- ❖ Xem danh sách các từ
- ❖ Tra từ
- ❖ Xem chi tiết nghĩa của từ
- ❖ Lưu tạm lại các từ đã tra
- ❖ Xóa các từ đã lưu tạm
- ❖ Thêm từ mới vào hệ thống
- ❖ Xóa từ

- ❖ Cập nhập lại từ
- ❖ Phát âm
- ❖ Cài đặt

Đặt tả các trường hợp sử dụng

- ❖ Trường hợp sử dụng : Lựa chọn ngôn ngữ

Tác nhân : người sử dụng , Admin .

Mô tả : Người dùng sẽ chọn ngôn ngữ cần tra trong danh sách các ngôn ngữ mà ứng dụng hỗ trợ.

- ❖ Trường hợp sử dụng : Tra từ

Tác nhân : người sử dụng , Admin

Mô tả : người dùng sẽ nhập từ cần tra vào thông qua bàn phím ảo của thiết bị

- ❖ Trường hợp sử dụng : Xem danh sách các từ đã lưu

Tác nhân : người sử dụng , Admin

Mô tả : người dùng có thể xem lại tất cả các từ đã lưu lại tương ứng với ngôn ngữ đang chọn.

- ❖ Trường hợp sử dụng : Xem chi tiết nghĩa của từ

Tác nhân : người sử dụng , Admin

Mô tả : Sau khi tìm được từ cần tra, người dùng sẽ nhấn vào từ đó để xem chi tiết đầy đủ nghĩa của từ đó.

- ❖ Trường hợp sử dụng : Xem danh sách các từ

Tác nhân : người sử dụng , Admin

Mô tả : Người dùng có thể xem được tất cả các từ có trong cơ sở dữ liệu theo từng trang, hoặc có thể chuyển đến số trang mong muốn để xem.

- ❖ Trường hợp sử dụng : Lưu từ

Tác nhân : người sử dụng , Admin

Mô tả : Với những từ đã tra được, người dùng có thể chọn lưu lại các từ đó để sau này dùng đến sẽ không cần phải tra lại, hoặc khi không nhớ từ đó là gì.

- ❖ Trường hợp sử dụng : Xóa các từ đã lưu

Tác nhân : người sử dụng , Admin

Mô tả : với các từ đã lưu tạm , xóa các từ đã lưu tạm theo mong muốn

❖ Trường hợp sử dụng : Thêm từ mới vào hệ thống

Tác nhân : Amin

Mô tả : Khi cần thêm một từ mới hay một cụm từ nào đó , người dùng sẽ chọn chức năng này

❖ Trường hợp sử dụng : xóa từ

Tác nhân : Amin

Mô tả : Xóa các từ không mong muốn khỏi hệ thống

❖ Trường hợp sử dụng : Cập nhập lại nghĩa của từ

Tác nhân : Admin

Mô tả : Nếu thấy một từ nào đó dịch chưa đúng, hoặc muốn cập nhật thêm vào từ đó, người dùng sẽ chọn chức năng này.

❖ Trường hợp sử dụng : phát âm

Tác nhân : người sử dụng , Admin

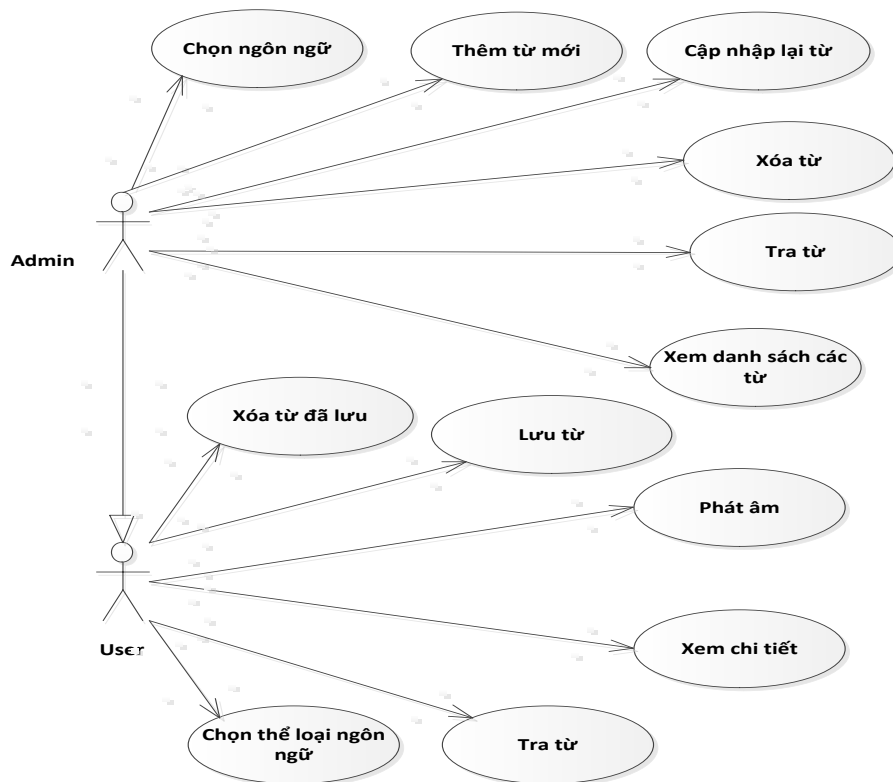
Mô tả : Tại phần xem chi tiết nghĩa của từ cần tra, người dùng có thể nghe cách phát âm của từ đó bằng cách nhấn vào biểu tượng phát âm .

❖ Trường hợp sử dụng : cài đặt

Tác nhân : người sử dụng , Admin

Mô tả : Thiết lập số kết quả hiển thị trên 1 trang

➤ **Biểu đồ Use-case tổng quát**



Hình 12 : Sơ đồ user case tổng quát

<<Usecase>>	Mục tiêu	Tác nhân chính
Chọn thể loại ngôn ngữ	Chọn ngôn ngữ cần tra	Người dùng/Admin
Tra từ	Hiển thị thông tin từ cần tra	Người dùng/Admin
Xem chi tiết	Hiển thị chi tiết từ cần tra	Người dùng/Admin
Phát âm	Phát âm từ tra được	Người dùng/Admin
Lưu từ	Lưu được các từ đã tra được	Người dùng/Admin
Xóa từ đã lưu	Xóa các từ đã lưu	Người dùng/Admin
Xem danh sách các từ	Xem danh sách các từ tra được	Người dùng/Admin

Thiết lập	Hiển thị số từ trên một trang	Người dùng/Admin
Thêm từ	Thêm từ mới vào Cơ Sở Dữ Liệu	Admin
Cập nhập lại từ	Cập nhập lại nghĩa của từ	Admin
Xóa từ	Xóa từ ra khỏi Cơ Sở Dữ Liệu	Admin

Bảng 1: Mô tả các Use-Case trong sơ đồ tổng quát

➤ *Use-case Chọn thể loại ngôn ngữ*

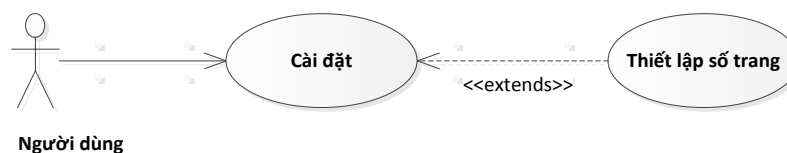


Hình 14 : Sơ đồ use-case chọn ngôn ngữ

<<Usecase>>	Mục tiêu	Tác nhân chính
Chọn thể loại ngôn ngữ	Chọn thể loại ngôn ngữ cần tra	Người dùng/Admin
Hiển thị danh sách ngôn ngữ	Hiển thị danh sách các ngôn ngữ cần tra	Người dùng/Admin

Bảng 2 : Mô tả Use-Case chọn ngôn ngữ

➤ *Use-case thiết lập*

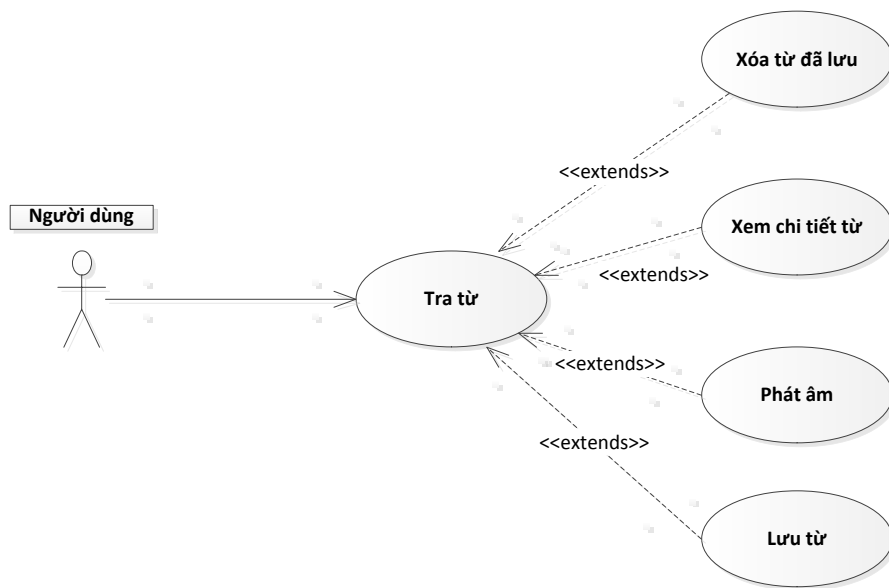


Hình 15 : Sơ đồ user case thiết lập

<<Usecase>>	Mục tiêu	Tác nhân chính
Cài đặt	Cài đặt số từ hiển thị	Người dùng/Admin
Thiết lập số trang	Hiển thị số từ trên một trang	Người dùng/Admin

Bảng 3: Mô tả user case cài đặt

➤ *Use-case tra từ*

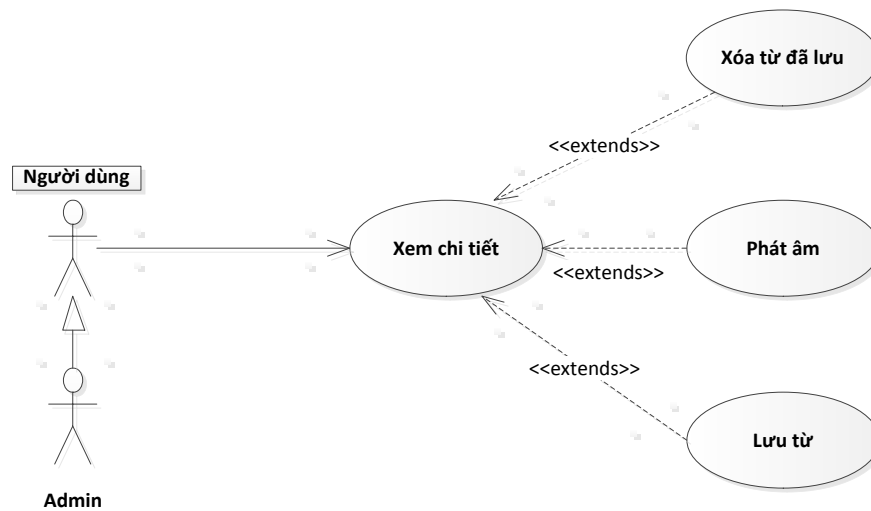


Hình 16 : Sơ đồ user case tra từ

<<Usecase>>	Mục tiêu	Tác nhân chính
Tra từ	Hiển thị thông tin các từ cần tra	Người dùng/Admin
Xem chi tiết các từ	Hiển thị xem chi tiết từ tra được	Người dùng/Admin
Lưu từ	Hiển thị lưu từ vừa tra được	Người dùng/Admin
Phát âm	Hiển thị phát âm từ vừa tra được	Người dùng/Admin
Xóa từ đã lưu	Hiện thi xóa các từ đã lưu	Người dùng/Admin

Bảng 4 : Mô tả user case tra từ

➤ *Use-case Xem chi tiết nghĩa của từ*

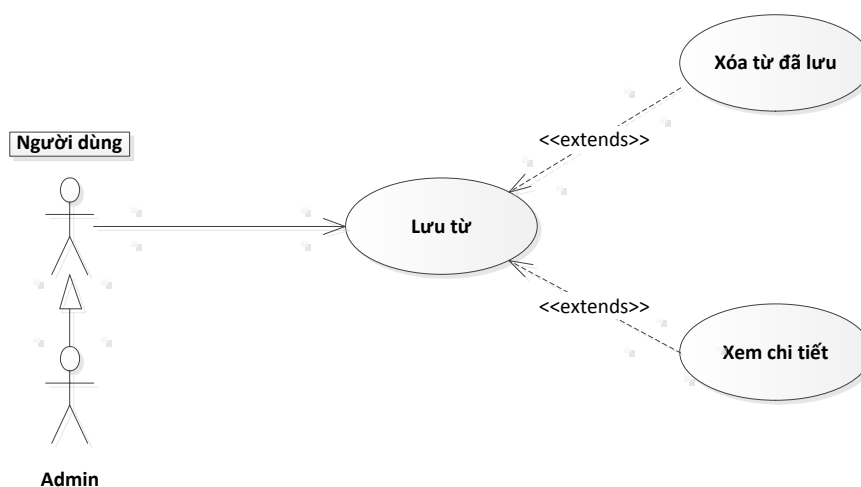


Hình 17 : Sơ đồ user case xem chi tiết từ

<<Usecase>>	Mục tiêu	Tác nhân chính
Xem chi tiết	Xem chi tiết các từ cần tra	Người dùng/Admin
Lưu từ	Hiển thị lưu từ vừa tra được	Người dùng/Admin
Phát âm	Hiển thị phát âm	Người dùng/Admin
Xóa từ đã lưu	Hiện thi xóa từ đã lưu	Người dùng/Admin

Bảng 5 : Mô tả user case xem chi tiết nghĩa của từ

➤ *Use-case Lưu từ*

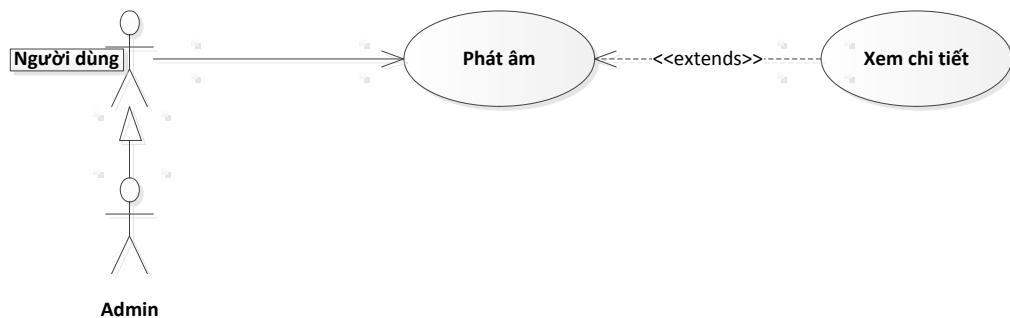


Hình 18 : Sơ đồ user case lưu từ

<<Usecase>>	Mục tiêu	Tác nhân chính
Lưu từ	Lưu lại các từ tra được	Người dùng/Admin
Xem chi tiết các từ	Hiển thị xem chi tiết từ đã lưu	Người dùng/Admin
Xóa từ đã lưu	Hiện thi xóa từ đã lưu	Người dùng/Admin

Bảng 6 : Mô tả user case lưu từ

➤ *Use-case Phát âm*



Hình 19 : Sơ đồ user case phát âm

<<Usecase>>	Mục tiêu	Tác nhân chính
Phát âm	Phát âm các từ đã tra được	Người dùng/Admin
Xem chi tiết	Hiển thị xem chi tiết các từ phát âm	Người dùng/Admin

Bảng 7 : Mô tả user case phát âm

➤ *Use-case Thêm từ mới*

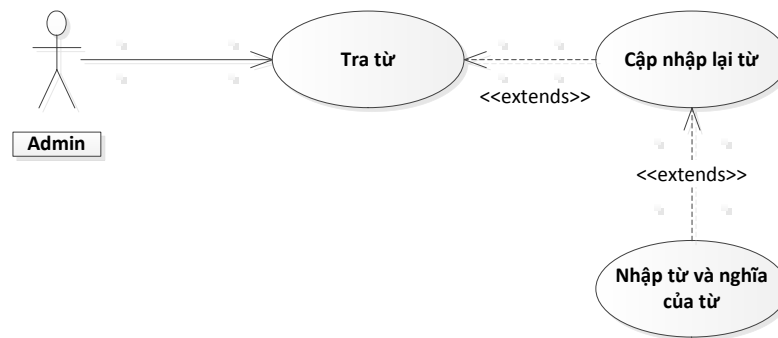


Hình 20 : Sơ đồ user case thêm từ mới

<<Usecase>>	Mục tiêu	Tác nhân chính
Thêm từ mới	Thêm từ mới vào cơ sở dữ liệu	Admin
Nhập từ và nghĩa của từ	Hiển thị nhập từ và nghĩa của từ	Admin

Bảng 8 : Mô tả user case thêm từ mới

➤ *Use-case Cập nhập lại nghĩa của từ*



Hình 21 : Sơ đồ user cập nhập lại từ

<<Usecase>>	Mục tiêu	Tác nhân chính
Tra từ	Tìm kiếm từ cần cập nhập	Admin
Cập nhập lại từ	Cập nhập lại từ	Admin
Nhập từ và nghĩa của từ	Hiển thị nhập từ và nghĩa của từ	Admin

Bảng 9 : Mô tả user case cập nhập lại nghĩa của từ

➤ *Use-case xóa từ*



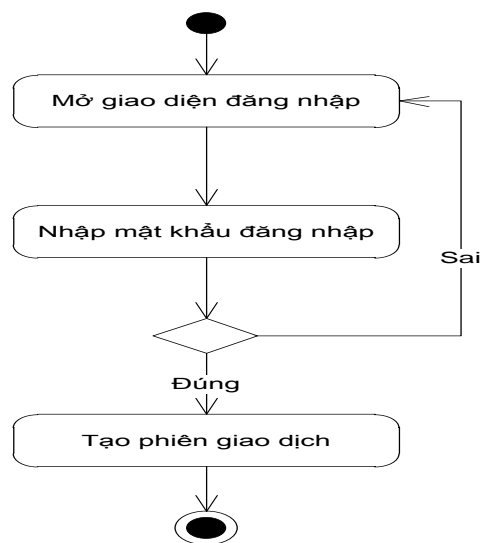
Hình 22: Sơ đồ user case xóa từ

<<Usecase>>	Mục tiêu	Tác nhân chính
Xóa từ	Xóa từ không mong muốn khỏi hệ thống	Admin

Bảng 10 : Mô tả user case xóa từ

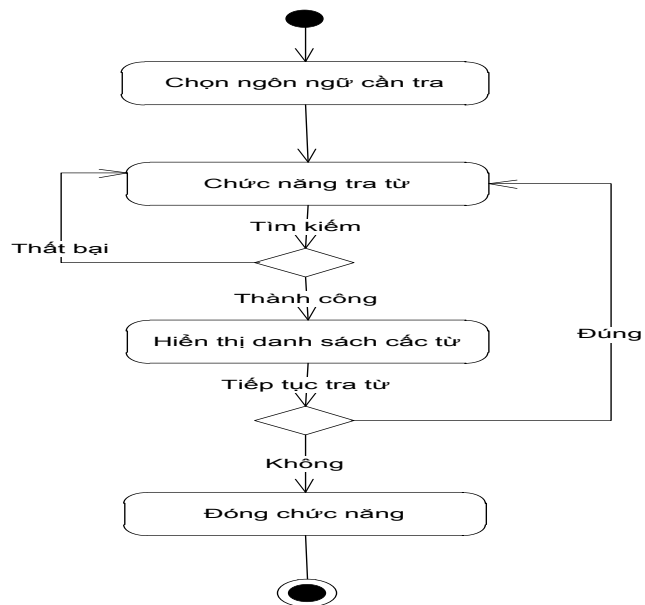
3.2.2. Biểu đồ hoạt động

➤ Sơ đồ hoạt động đăng nhập



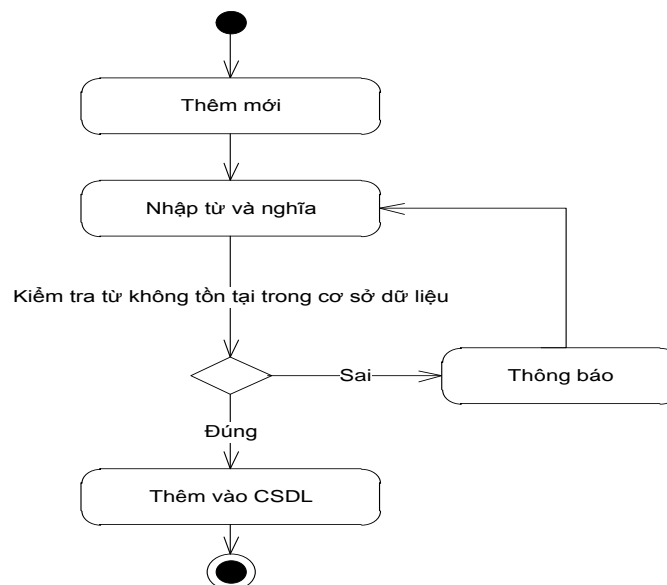
Hình 23 : Sơ đồ hoạt động đăng nhập

➤ Tra từ



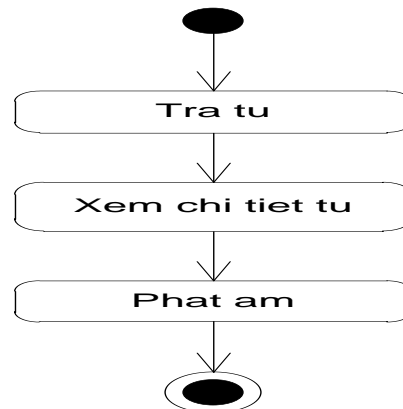
Hình 24 : Sơ đồ hoạt động tra từ

➤ Thêm từ mới



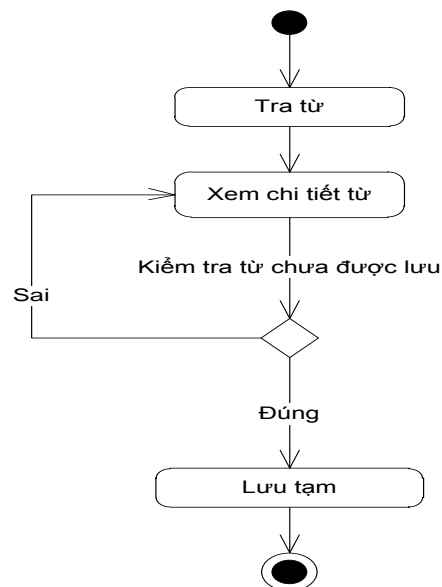
Hình 25 : Sơ đồ hoạt động thêm từ mới

➤ Phát âm



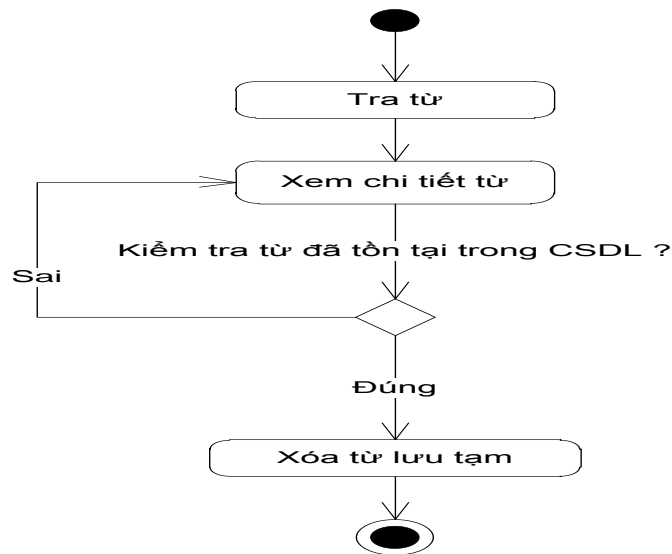
Hình 26 : Sơ đồ hoạt động phát âm

➤ Lưu từ



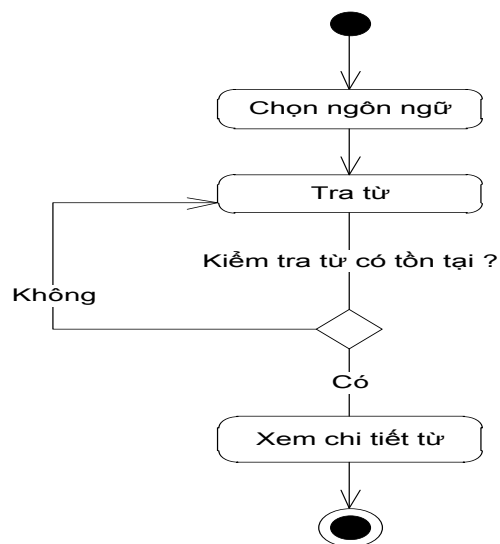
Hình 27 : Sơ đồ hoạt động lưu từ

➤ Xóa từ đã lưu



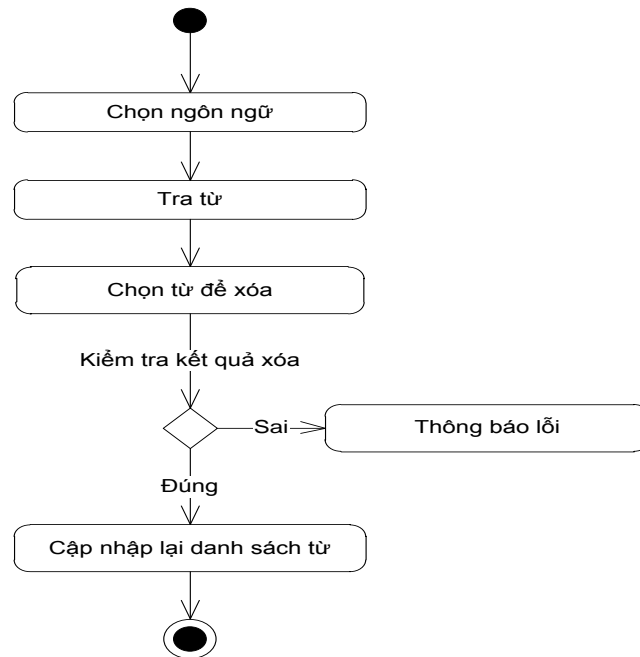
Hình 28 : Sơ đồ hoạt động xóa từ đã lưu

➤ Xem chi tiết từ



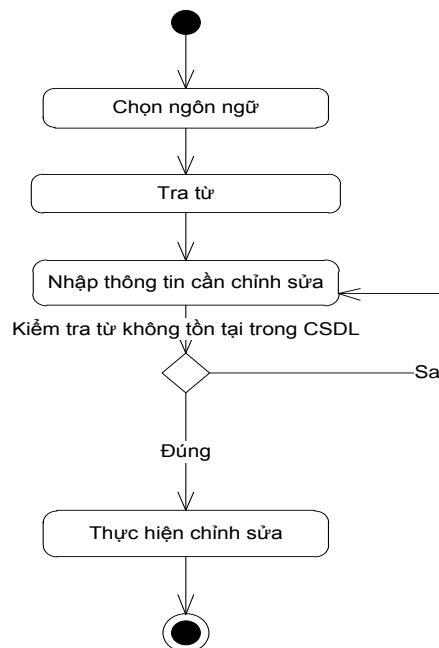
Hình 29: Sơ đồ hoạt động xem chi tiết từ

➤ Xóa từ



Hình 30 : Sơ đồ hoạt động xóa từ

➤ Cập nhập lại nghĩa của từ



Hình 31 : Sơ đồ hoạt động cập nhập lại nghĩa của từ

3.2.3. Cơ sở dữ liệu SQLite

3.2.3.1. Thiết kế dữ liệu cho hệ thống

➤ Bảng Admin

Tính chất	Tên trường	Kiểu dữ liệu	Kích thước	Chú thích
	UserName	Text		Tài khoản Admin
	Password	Text		Mật khẩu

Bảng 11 : Bảng dữ liệu Admin

➤ Bảng WordSaved

Tính chất	Tên trường	Kiểu dữ liệu	Kích thước	Chú thích
Foreign key	<i>word_id</i>	Integer		
	language_id	Integer		

Bảng 12 : Bảng dữ liệu Word saved

➤ Bảng anh_viet

Tính chất	Tên trường	Kiểu dữ liệu	Kích thước	Chú thích
Primary key Unique	id	Integer		Mã của từ (Khóa chính)
	word	Text		Từ cần tra
	content	Text		Nghĩa của từ

Bảng 13 : Bảng dữ liệu anh việt

➤ *Bảng viet_anh*

Tính chất	Tên trường	Kiểu dữ liệu	Kích thước	Chú thích
Primary key Unique	id	Integer		Mã của từ (Khóa chính)
	word	Text		Từ cần tra
	content	Text		Nghĩa của từ

Bảng 14 : Bảng dữ liệu viet anh

➤ *Bảng viet_nhat*

Tính chất	Tên trường	Kiểu dữ liệu	Kích thước	Chú thích
Primary key Unique	Id	Integer		Mã của từ (Khóa chính)
	word	Text		Từ cần tra
	content	Text		Nghĩa của từ

Bảng 15 : Bảng dữ liệu viet nhật

➤ *Bảng viet_phap*

Tính chất	Tên trường	Kiểu dữ liệu	Kích thước	Chú thích
Primary key Unique	id	Integer		Mã của từ (Khóa chính)
	word	Text		Từ cần tra
	content	Text		Nghĩa của từ

Bảng 16 : Bảng dữ liệu viet pháp

➤ *Bảng viwiki (Bách khoa toàn thư)*

Tính chất	Tên trường	Kiểu dữ liệu	Kích thước	Chú thích
Primary key Unique	id	Integer		Mã của từ (Khóa chính)
	word	Text		Từ cần tra
	content	Text		Nghĩa của từ

Bảng 17 : Bảng dữ liệu bách khoa toàn thư

3.2.3.2. *Mối quan hệ giữa các bảng*



Hình 32 : Mối quan hệ giữa các bảng

CHƯƠNG 4 : TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

4.1. Các công cụ , môi trường phát triển

4.1.1. Yêu cầu

Để phát triển ứng dụng cho điện thoại Adnroid ta cần:

- Elipse IDE
- JDK

4.1.2. Cài đặt

Cài đặt môi trường phát triển Android cần chuẩn bị các phần mềm:

Eclipse (có thể download tại đây: <http://www.eclipse.org/downloads/>)

SDK (Địa chỉ tải về : <http://developer.android.com/sdk/index.html>)

ADT Plugin for Eclipse

4.1.3. Công cụ viết chương trình

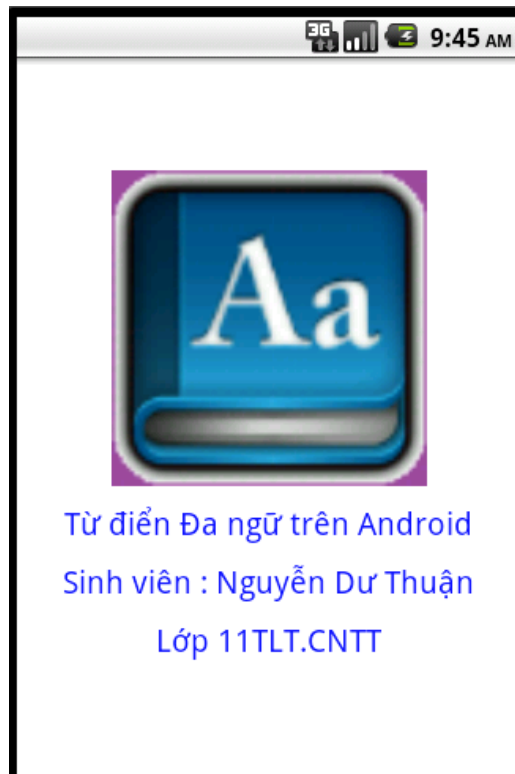
- Eclipse là công cụ để viết mã nguồn và biên dịch chương trình
- SQLite : dùng làm hệ cơ sở dữ liệu cho chương trình

4.1.4. Công cụ viết báo cáo

- Microsoft Word 2010: dùng để viết tài liệu
- UML 2.2 Template (Visio 2010) : dùng để vẽ các sơ đồ mô tả hệ thống

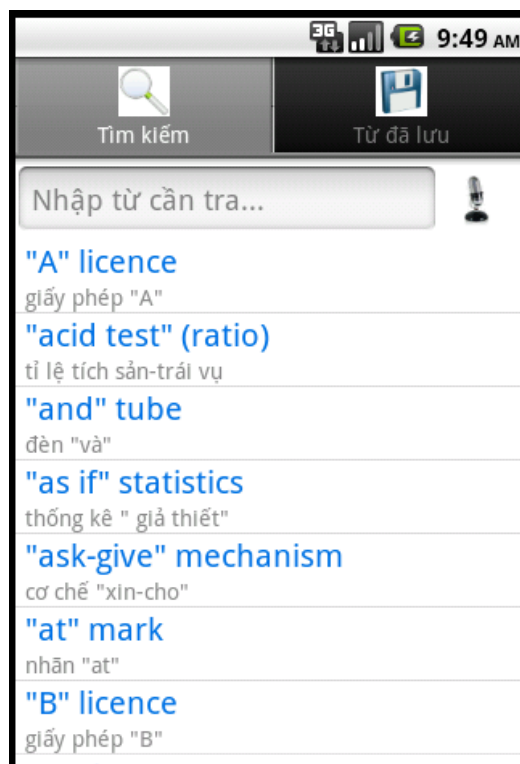
4.2. Demo chương trình

➤ Màn hình khởi động



Hình 33: Màn hình khởi động

➤ Giao diện tra từ



Hình 34 : Giao diện tra từ

➤ **Giao diện xem chi tiết nghĩa của từ**



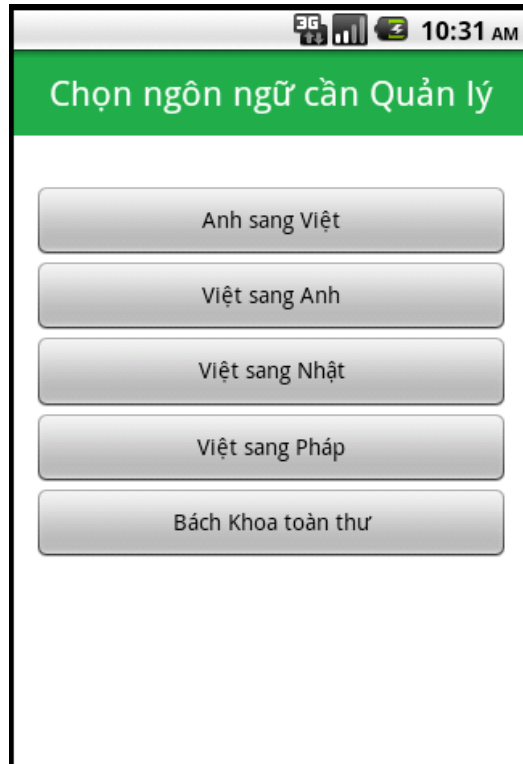
Hình 35 : Giao diện xem chi tiết nghĩa của từ

➤ **Giao diện đăng nhập**



Hình 36 : Giao diện đăng nhập

➤ **Giao diện chọn ngôn ngữ quản lý**



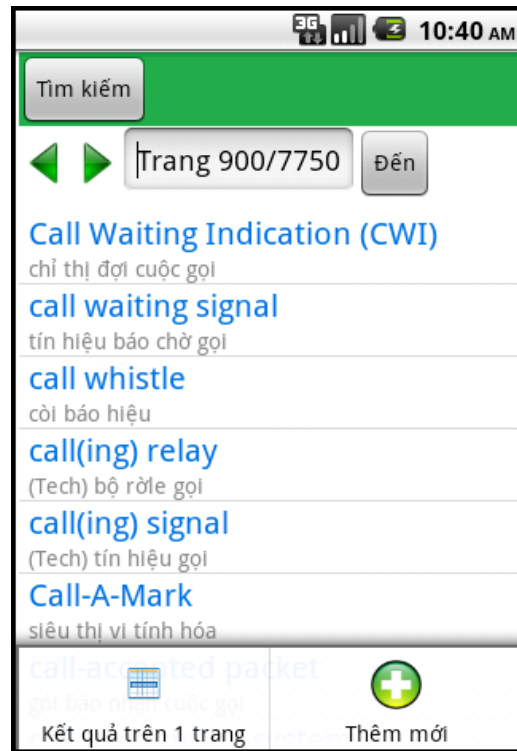
Hình 37 : Giao diện chọn ngôn ngữ cần quản lý

➤ **Giao diện thêm từ**



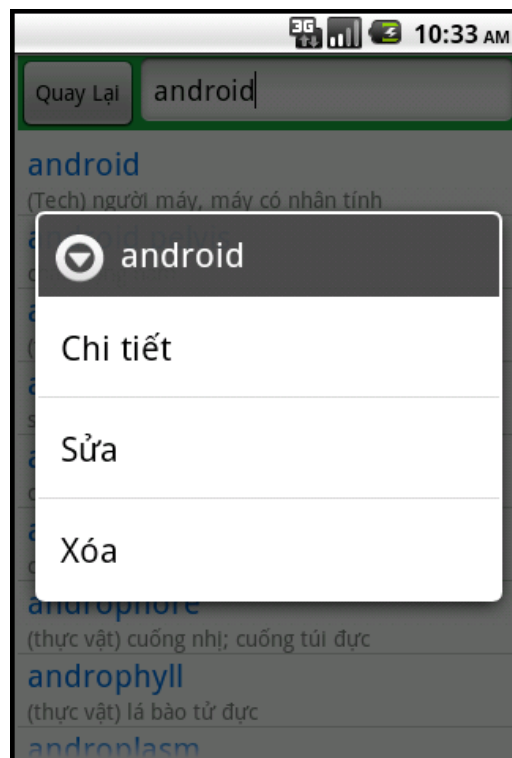
Hình 38 : Giao diện thêm từ

➤ **Giao diện danh sách các từ**



Hình 39 : Giao diện danh sách các từ

➤ **Giao diện sửa , xóa từ**



Hình 40 : Giao diện sửa , xóa từ

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

➤ Kết quả đạt được

❖ Về mặt lý thuyết

Qua quá trình làm đồ án này, em đã đạt được những kết quả về mặt lý thuyết như sau:

- Tìm hiểu được ngôn ngữ Java
- Biết áp dụng môn phân tích thiết kế hệ thống vào thực tế.
- Biết cách xây dựng ứng dụng từ điển đa ngữ trên nền tảng Android

❖ Về mặt thực tiễn

Từ lý thuyết, áp dụng kiến thức đó vào thực tế để phân tích bài toán cụ thể là “Xây dựng ứng dụng từ điển đa ngữ trên hệ điều hành Android”, em đã đạt được những kết quả như sau :

- Tra cứu từ điển đa ngôn ngữ: Anh-Việt, Việt-Anh, Việt-Nhật, Việt-Pháp, Bách khoa toàn thư.
- Xem danh sách các từ
- Tra từ
- Xem chi tiết nghĩa các từ
- Lưu tạm các từ đã tra và xem danh sách các từ tra được
- Quản lý từ : thêm , xóa , sửa các từ .
- Phát âm được từ đã tra (trong từ điển Anh – Việt)

➤ Hạn chế

Với những yêu cầu đã nêu ra khi thực hiện đề tài bên cạnh những kết quả đạt được vẫn còn một số điểm chưa đạt được :

- Chưa tích hợp được nhiều ngôn ngữ
- Chức năng còn thiếu như dịch đoạn văn bản

➤ Hướng phát triển của đề tài

- Tiếp tục hoàn thiện hơn ứng dụng cả về chức năng và giao diện
- Tích hợp thêm nhiều ngôn ngữ tra từ khác để giúp ứng dụng đầy đủ và phong phú hơn.
- Tìm hiểu và bổ sung thêm chức năng dịch đoạn văn, nhận dạng giọng nói, nhận dạng từ bằng hình ảnh.

TÀI LIỆU THAM KHẢO

- [1] **Wikipedia.** *Java (Ngôn ngữ lập trình)*
- [2] **Jeff Friesen.** *Learn Java for Android Development*, 2009.
- [3] Trang web: <http://www.lichsvietnam.vn/home.php>
- [4] Trang web: www.codeproject.com
- [5] **Google INC.** *Android Developer Docs.* (Online)
- [6] **Google INC.** *Google Maps API Web Services.* (Online)
- [7] **Wikipedia.** *Android(Hệ điều hành)* (Online)
- [8] **Wallace Jackson.** *Android Apps for Absolute Beginners*, 2011.
- [9] Trang web: <http://vietandroid.com>
- [10] Trang web: <http://www.dict.org.com>