

TRƯỜNG ĐẠI HỌC MỞ TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CÔNG NGHỆ PHẦN MỀM
XÂY DỰNG ỨNG DỤNG GIỚI THIỆU ÂM
THỰC CHO SMARTPHONE
CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

Sinh viên thực hiện: Nguyễn Quốc Việt

MSSV: 1651012221

Người hướng dẫn: Thạc sĩ Lưu Quang Phương

Tp.HCM, năm 2019

TRƯỜNG ĐẠI HỌC MỎ TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

PHIẾU GIAO ĐỒ ÁN CÔNG NGHỆ PHẦN MỀM

Tên đồ án: **Xây dựng ứng dụng giới thiệu ẩm thực cho smartphone**

Nhiệm vụ:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ngày giao đồ án: ngày ... tháng ... năm 2019

Ngày hoàn thành đồ án: ngày ... tháng năm 2019

Số lượng sinh viên thực hiện đồ án: 1

Họ và tên sinh viên: Nguyễn Quốc Việt MSSV: 1651012221

Trưởng Khoa/ Bộ môn

Tp.HCM, ngày ... tháng ... năm 2019

Người hướng dẫn

ThS. Lưu Quang Phương

NHẬN XÉT VÀ ĐÁNH GIÁ ĐIỂM CỦA NGƯỜI HƯỚNG DẪN

- Ý thức thực hiện:.....

.....

.....

.....

.....

.....

- Nội dung thực hiện:.....

.....

.....

.....

.....

.....

- Hình thức trình bày:

.....

.....

.....

.....

- Tổng hợp kết quả:

.....

.....

☐ Tổ chức báo cáo trước hội đồng

☐ Tổ chức chấm thuyết minh

Tp.HCM, ngày ... tháng ... năm 2019

Người hướng dẫn

ThS. Lưu Quang Phương

LỜI CẢM ƠN

Em xin chân thành cảm thầy Lưu Quang Phương đã tận tình giúp đỡ, hướng dẫn và cung cấp tài liệu liên quan đến đề án công nghệ phần mềm.

Em xin chân thành cảm ơn các thầy cô trong khoa Công Nghệ Thông Tin nói riêng và các thầy cô khác nói chung đã tạo điều kiện thuận lợi trong học tập.

Em cảm ơn các thầy cô trong khoa đã tận tình giảng dạy , trang bị cho em những kiến thức quý báu trong những năm học vừa qua.

Cảm ơn các anh chị và bạn bè đã góp ý xây dựng đề án này.

Sinh viên thực hiện

Nguyễn Quốc Việt

MỤC LỤC

LỜI CẢM ƠN	1
MỤC LỤC.....	2
DANH MỤC HÌNH	5
DANH MỤC BẢNG	6
LỜI NÓI ĐẦU	7
CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI	8
1.1 Đặt vấn đề	8
1.2 Mục đích nghiên cứu	8
1.3 Ý nghĩa đề tài	8
1.4 Nhiệm vụ đề tài	9
1.4.1 Về lý thuyết	9
1.4.2 Về chương trình	10
1.5 Phương pháp nghiên cứu	10
1.6 Phạm vi nghiên cứu	10
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	11
1.7 Ngôn ngữ lập trình java	11
1.7.1 Lịch sử Java	11
1.7.2 Khái niệm:	12
1.7.3 Đặc điểm ngôn ngữ:.....	12
1.7.4 Phương châm	13
1.7.5 Các phiên bản đã phát hành	13
1.7.6 Sự phổ biến.....	14
1.8 Android:	14

1.8.1	Khái niệm Android:	14
1.8.2	Mô Tả	15
1.8.3	Lịch sử phát triển:	16
1.8.4	Kiến trúc Android	18
1.8.5	Cấu trúc stack android	21
1.8.6	Bảo mật và tính riêng tư	23
1.8.7	Phát triển	24
1.8.8	Giấy phép phát hành	28
1.8.9	Bảng quyền và bằng phát minh	28
1.8.10	Sự đón nhận người dùng	29
1.8.11	Các thiết bị ngoài điện thoại và máy tính bảng	31
1.9	Android studio:	32
1.9.1	Khái niệm:	32
1.9.2	Chức năng:	32
1.10	Google Firebase:	32
1.10.1	Lịch sử	32
1.10.2	Tổng quan về Firebase:	33
1.10.3	Dịch vụ	35
CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG		37
1.11	Đặc tả yêu cầu hệ thống	37
1.12	Phân tích hệ thống	37
1.12.1	Mô hình cơ sở dữ liệu:	37
1.12.2	Chú thích cơ sở dữ liệu:	40
CHƯƠNG 4: ỨNG DỤNG GIỚI THIỆU ÂM THỰC CHO SMARTPHONE		41
4.1	Sơ đồ phân rã chức năng phần mềm	41
4.2	Phân tích hệ thống chức năng chính phần mềm	42

4.2.1	Đăng ký	42
4.2.2	Thêm, Sửa, Xóa Món Ăn cho Quản trị viên	43
4.3	Ứng dụng giới thiệu ẩm thực cho smartphone.....	44
CHƯƠNG 5: KẾT LUẬN VÀ KIẾN NGHỊ		51
5.1	Kết luận.....	51
5.2	Kiến nghị.....	51
TÀI LIỆU THAM KHẢO		52

DANH MỤC HÌNH

Hình 3.1 Cơ sở dữ liệu tổng quát	38
Hình 3.2 Node lưu lại	38
Hình 3.3 Node thành phần	38
Hình 3.4 Node vùng miền.....	38
Hình 3.5 Node người dùng	39
Hình 3.6 Node gợi ý	39
Hình 3.7 Node đánh giá.....	39
Hình 3.8 Node đặc sản.....	39
Hình 4. 1 Sơ đồ phân rã chức năng phần mềm.....	41
Hình 4. 2 Sơ đồ phân rã chức năng tìm kiếm	41
Hình 4. 3 Sơ đồ phân rã chức năng người dùng	41
Hình 4. 4 Sơ đồ phân rã chức năng người quản lý	42
Hình 4. 5 Sơ đồ phân rã chức năng người sử dụng.....	42
Hình 4. 6 Giao diện splash screen.....	44
Hình 4. 7 Giao diện chọn khu vực	44
Hình 4. 8 Danh sách các khu vực.....	44
Hình 4. 9 Giao diện ăn gì.....	44
Hình 4. 10 Giao diện uống gì.....	45
Hình 4. 11 Đánh giá bài viết.....	46
Hình 4. 12 Chi tiết bài viết.....	46
Hình 4. 13 Lưu bài viết.....	46
Hình 4. 14 Tìm bài viết.....	47
Hình 4. 15 Đăng nhập bằng tài khoản.....	48
Hình 4. 16 Đăng ký tài khoản	48
Hình 4. 17 Trang cá nhân người dùng.....	49
Hình 4. 18 Trang cá nhân người quản lý.....	49
Hình 4. 19 Menu chức năng người quản lý	49
Hình 4. 20 Chức năng thêm bài viết mới	50
Hình 4. 21 Chức năng cập nhật bài viết	50

DANH MỤC BẢNG

Bảng 3.1 Chú thích các key trong cơ sở dữ liệu	40
Bảng 5.1 Bảng kết quả đạt được và chưa đạt được.....	51

LỜI NÓI ĐẦU

Hiện nay dịch vụ du lịch ngày càng đẩy mạnh, khách du lịch nước ngoài và trong nước ngày một nhiều hơn, việc đến một địa điểm nào đó ngoài vui chơi giải trí còn phải quan tâm đến thức ăn nước uống đặc sản của nơi đó nhưng việc tìm hiểu các món ăn nước uống đặc sản một nơi thì mất khá nhiều thời gian và nhiều thông tin không tìm thấy hoặc tìm thấy sai lệch với thực tế. Vì vậy cần phần mềm giới thiệu ẩm thực giúp khách du lịch có thể dễ dàng tìm kiếm món ăn, nước uống đặc trưng của một khu vực một cách nhanh chóng và chi tiết.

Phần mềm đáp ứng được các nhu cầu cơ bản như sau:

- Tìm kiếm món ăn, nước uống theo từng khu vực, giá tiền, thành phần.
- Xem thông tin món ăn nước uống một cách chi tiết cụ thể.
- Lưu giữ lại những món ăn, nước uống mình thích.

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1 Đặt vấn đề

Công việc kinh doanh buôn bán đã xuất hiện từ rất lâu, trải qua mỗi giai đoạn lịch sử nó mang một hình thức đặc thù riêng. Trước kia, khi các công cụ hỗ trợ công việc mua bán chưa phát triển mạnh, thì người kinh doanh mua bán chỉ diễn ra dưới hình thức mua bán trực tiếp. Từ khi khoa học công nghệ phát triển, nó đã tạo ra phát triển các loại hình thức mua bán mới, điển hình là mua bán trực tuyến. Hình thức mua bán trực tuyến hỗ trợ đắc lực cho người kinh doanh tiếp xúc với được nhiều khách hàng, còn khách hàng thì công việc mua bán được diễn ra nhanh chóng, thuận lợi, tiết kiệm được thời gian ...

- Hiện nay, Thương Mại Điện Tử (TMĐT) đang được nhiều quốc gia quan tâm, coi là một trong những động lực phát triển chủ yếu của nền kinh tế. TMĐT đem lại những lợi ích tiềm tàng, giúp doanh nghiệp thu được thông tin phong phú về thị trường và đối tác, giảm chi phí tiếp thị và giao dịch, tạo dựng và củng cố quan hệ bạn hàng.

- Việc mua hàng qua mạng chỉ với thủ tục đăng ký mua sắm đơn giản nhưng đem lại nhiều lợi ích: tiết kiệm và chủ động về thời gian, tránh khỏi những phiền phức khó chịu. Vậy nên việc mua bán hàng qua mạng đang rất được mọi người quan tâm.

Trên cơ sở các kiến thức được học trong nhà trường và quá trình tìm hiểu các website trong thực tế, em đã quyết định chọn đề tài “Xây dựng ứng dụng giới thiệu ẩm thực trên nền Android”.

1.2 Mục đích nghiên cứu

- Xây dựng phần mềm giới thiệu ẩm thực
- Ứng dụng chạy trên hệ điều hành Android
- Ứng dụng trực quan, đơn giản, dễ thao tác với người sử dụng
- Đáp ứng được những nhu cầu cơ bản của người sử dụng trong việc tìm kiếm món ăn nước uống và xem thông tin một cách chi tiết

1.3 Ý nghĩa đề tài

Trước hết, đề tài “Xây dựng ứng dụng giới thiệu ẩm thực trên nền Android” là một ứng dụng được xây dựng như một đề tài thể hiện việc áp dụng những kiến thức quý báu đã

được các thầy cô của trường Đại học Mở Tp.HCM nhiệt tình truyền thụ lại cho chúng em. Và đặc biệt là sự theo dõi và quan tâm giúp đỡ của thầy Lưu Quang Phương trong suốt thời gian em thực hiện đề tài này. Thứ hai là những kinh nghiệm quý báu mà em có được trong quá trình thực hiện đề tài sẽ là hành trang tuyệt vời giúp ích rất nhiều cho công việc sau này của mình. Thứ ba, trong thời đại công nghệ thông tin phát triển như vũ bão ngày nay, công nghệ không ngừng được cải tiến, những công nghệ mới ra đời đòi hỏi người lập trình viên phải cố gắng nghiên cứu và tìm hiểu để áp dụng vào thực tiễn. Điều này cũng được thể hiện qua sự cố gắng em khi quyết định thực hiện đề tài này với các công nghệ được coi là mới nhất hay và được dự báo sẽ phát triển mạnh mẽ trong tương lai. Do vậy, em rất hy vọng sẽ tạo nên một sản phẩm hoàn chỉnh nhất có thể để đáp lại sự nhiệt tình giúp đỡ từ thầy cô và cả tâm huyết của em khi quyết định thực hiện đề tài này.

1.4 Nhiệm vụ đề tài

1.4.1 Về lý thuyết

Đề tài “Xây dựng ứng dụng giới thiệu ẩm thực trên nền Android” là một ứng dụng chạy trên điện thoại sử dụng hệ điều hành Android. Ứng dụng gồm:

- Phần ứng dụng chạy trên điện thoại được phát triển trên nền Android SDK.
- Phần mềm quản lý cho phép người quản lý có thể thêm xóa cập nhật món ăn, xử lý các thông tin từ ứng dụng Android gửi về.

Dưới đây là những chức năng chính mà em dự định phát triển ở phiên bản đầu tiên của ứng dụng này: Đề tài xây dựng ứng dụng mua bán fastfood trên nền android.

Khách hàng tải ứng dụng về điện thoại có sử dụng hệ điều hành android 4.0. Khách hàng chạy ứng dụng có thể xem danh sách món ăn mà nhà hàng đã cập nhật trong cơ sở dữ liệu. Khách hàng có thể chọn món và lưu món ăn. Sau khi khách hàng lưu món ăn, thông tin sẽ được chuyển về lưu vào cơ sở dữ liệu thông qua Firebase.

Quản lý danh mục món ăn như cập nhật, xóa, thống kê.

Mục tiêu của đề tài này là:

- Tìm hiểu các hoạt động, nhu cầu ăn uống của khách du lịch trong thực tế.
- Tìm hiểu cách thức xây dựng một ứng dụng trên android.
- Xây dựng thành công ứng dụng giới thiệu món ăn trên android.

- Giới thiệu món ngon, đặc sản thu hút khách du lịch cho địa phương

1.4.2 Về chương trình

- Ứng dụng giới thiệu ẩm thực đã phần nào xây dựng và đáp ứng được một số chức năng chính:

- + Quản lý việc tài khoản,
- + Cho phép tìm kiếm thông tin trong hệ thống,...
- + Giao diện thân thiện, dễ sử dụng.

1.5 Phương pháp nghiên cứu

- Tìm kiếm các món ăn nước uống đặc sản của từng khu vực một cách chi tiết

1.6 Phạm vi nghiên cứu

- Tên ứng dụng: Giới thiệu ẩm thực
- Phạm vi sử dụng: Áp dụng cho smartphone hệ điều hành Android
- Đối tượng sử dụng: Mọi người sử dụng smartphone Android.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

1.7 Ngôn ngữ lập trình java

1.7.1 Lịch sử Java

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun Microsystems năm 1991. Ban đầu ngôn ngữ này được gọi là **Oak** (có nghĩa là cây sồi; do bên ngoài cơ quan của ông Gosling có trồng nhiều loại cây này), họ dự định ngôn ngữ đó thay cho C++, nhưng các tính năng giống Objective C. Không nên lẫn lộn Java với JavaScript, hai ngôn ngữ đó chỉ giống tên và loại cú pháp như C. Công ty Sun Microsystems đang giữ bản quyền và phát triển Java thường xuyên. Tháng 04/2011, công ty Sun Microsystems tiếp tục cho ra bản JDK 1.6.24.

Java được tạo ra với tiêu chí "Viết (code) một lần, thực thi khắp nơi" ("Write Once, Run Anywhere" (WORA)). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó. Môi trường thực thi của Sun Microsystems hiện hỗ trợ Sun Solaris, Linux, Mac OS, FreeBSD & Windows. Ngoài ra, một số công ty, tổ chức cũng như cá nhân khác cũng phát triển môi trường thực thi Java cho những hệ điều hành khác như BEA, IBM, HP.... Trong đó đáng nói đến nhất là IBM Java Platform hỗ trợ Windows, Linux, AIX & z/OS.

Những chi tiết về ngôn ngữ, máy ảo và API của Java được giữ bởi Cộng đồng Java (do Sun quản lý). Java được tạo ra vào năm 1991 do một số kỹ sư ở Sun, bao gồm ông James Gosling, một phần của *Dự án Xanh* (Green Project). Java được phát hành vào năm 1994, rồi nó trở nên nổi tiếng khi Netscape tuyên bố tại hội thảo SunWorld năm 1995 là trình duyệt Navigator của họ sẽ hỗ trợ Java. Về sau Java được hỗ trợ trên hầu hết các trình duyệt như Internet Explorer (Microsoft), Firefox (Mozilla), Safari (Apple)... Java được sử dụng chủ yếu trên môi trường NetBeans và Oracle. Sau khi [Oracle](#) mua lại công ty Sun Microsystems năm 2009-2010, Oracle đã mô tả họ là "*người quản lý công nghệ Java với cam kết không ngừng để bồi dưỡng một cộng đồng tham gia và minh bạch*".

1.7.2 Khái niệm:

Java (phiên âm Tiếng Việt: "*Gia-va*") là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp (class). 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.

Trước đây, Java chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C và C++, nhưng sau này nhờ công nghệ "biên dịch tại chỗ" - Just in time compilation, khoảng cách này đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn. Java chạy nhanh hơn những ngôn ngữ thông dịch như Python, Perl, PHP gấp nhiều lần. Java chạy tương đương so với C#, một ngôn ngữ khá tương đồng về mặt cú pháp và quá trình dịch/chạy

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.

Trong Java, hiện tượng rò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lý bởi Java Virtual Machine (JVM) bằng cách tự động "dọn dẹp rác". Người lập trình không phải quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Tuy nhiên khi sử dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của JVM) mà người lập trình không đóng (close) các streams thì rò rỉ dữ liệu vẫn có thể xảy ra.

1.7.3 Đặc điểm ngôn ngữ:

- Hướng đối tượng:
 - Mọi thực thể trong chương trình đều là một đối tượng (1 class xác định)
 - Các biến, hàm đều nằm trong một class nào đó
- Đơn giản:
 - Loại bỏ con trỏ
 - Loại bỏ lệnh goto
 - Không cho phép đa kế thừa (chuyển sang sử dụng interface)
- Độc lập phần cứng và hệ điều hành:
 - 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.

- Do đó một chương trình viết bằng Java có thể chạy trên nhiều thiết bị, nhiều hệ điều hành khác nhau.
- Mạnh mẽ:
 - Quá trình cấp phát, giải phóng bộ nhớ được thực hiện tự động.
 - Yêu cầu chặt chẽ khi khai báo dữ liệu, ép kiểu dữ liệu.
 - Tự động phát hiện lỗi lúc biên dịch.
 - Không sử dụng con trỏ hoặc các phép toán con trỏ.
- Phân tán
 - Java hỗ trợ lập trình cho các hệ thống phân tán như client-server, RMI... bằng Java web, UDP, TCP...
- Đa luồng
 - Java hỗ trợ lập trình đa luồng (multithreading); việc đồng bộ dữ liệu trong lập trình đa luồng cũng khá đơn giản.

1.7.4 Phương châm

Có 6 mục tiêu chính trong việc xây dựng ngôn ngữ Java:

1. Đơn giản, hướng đối tượng và quen thuộc.
2. Mạnh mẽ và an toàn.
3. Kiến trúc trung lập và di động.
4. Thực thi với hiệu suất cao.
5. Dịch ra bytecode, phân luồng và năng động.
6. Dễ sử dụng cho người dùng Java

1.7.5 Các phiên bản đã phát hành

Các phiên bản Java đã phát hành:

- JDK 1.0 (23 tháng 01, 1996)
- JDK 1.1 (19 tháng 2 năm 1997)

- JDK 1.1.5 (*Pumpkin*) 03 tháng 12 năm 1997
- JDK 1.1.6 (*Abigail*) 24 tháng 4 năm 1998
- JDK 1.1.7 (*Brutus*) 28 tháng 9 năm 1998
- JDK 1.1.8 (*Chelsea*) 08 tháng 4 năm 1999
- J2SE 1.2 (*Playground*) 08 tháng 12 năm 1998
 - J2SE 1.2.1 (*không có*) 30 tháng 3 năm 1999
 - J2SE 1.2.2 (*Cricket*) 08 tháng 7 năm 1999
- J2SE 1.3 (*Kestrel*) 08 tháng 5 năm 2000
 - J2SE 1.3.1 (*Ladybird*) 17 tháng 5 năm 2001
- J2SE 1.4.0 (*Merlin*) 06 tháng 02, 2002
 - J2SE 1.4.1 (*Hopper*) 16 tháng 9 năm 2002
 - J2SE 1.4.2 (*Mantis*) 26 tháng 6 năm 2003
- J2SE 5 (1.5.0) (*Tiger*) 30 tháng 9 năm 2004
- Java SE 6 (còn gọi là *Mustang*), được công bố 11 tháng 12 năm 2006, thông tin chính tại <http://java.sun.com/javase/6/>. Các bản cập nhật 2 và 3 được đưa ra vào năm 2007, bản cập nhật 4 đưa ra tháng 1 năm 2008.
- JDK 6.18, 2010
- Java SE 7 (còn gọi là *Dolphin*), được bắt đầu từ tháng 8 năm 2006 và công bố ngày 28 tháng 7 năm 2011.
- JDK 8, 18 tháng 3 năm 2014
- JDK 9, 21 tháng 9 năm 2017
- JDK 10, 20 tháng 3 năm 2018

1.7.6 Sự phổ biến

Tính đến ngày 9 tháng 12 năm 2017, Java là ngôn ngữ lập trình phổ biến nhất thế giới với tỉ lệ 13,268%, nhưng đã giảm 4,59% so với cùng kì năm trước. Java và C vẫn luôn chiếm hai vị trí đầu tiên trong bảng các ngôn ngữ lập trình phổ biến nhất suốt 20 năm qua, nhưng Java là ngôn ngữ duy nhất giữ được tỉ lệ trên 10%, trong khi C đã tụt xuống 6,478% vào ngày 2/8/2017

1.8 Android:

1.8.1 Khái niệm Android:

Android là một hệ điều hành có mã nguồn mở 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, sau này được chính Google mua lại vào năm 2005 và hệ điều hành Android đã

ra mắt vào năm 2007. Chiếc điện thoại đầu tiên chạy Android là HTC Dream được bán vào ngày 22 tháng 10 năm 2008.

1.8.2 Mô Tả

1.8.2.1 Giao diện

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

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

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

1.8.2.2 Ứng dụng

Android có lượng ứng dụng của bên thứ ba ngày càng nhiều, được chọn lọc và đặt trên một cửa hàng ứng dụng như Google Play hay Amazon Appstore để người dùng lấy về, hoặc bằng cách tải xuống rồi cài đặt tập tin APK từ trang web khác. Các ứng dụng trên Play Store cho phép người dùng duyệt, tải về và cập nhật các ứng dụng do Google và các

nhà phát triển thứ ba phát hành. Play Store được cài đặt sẵn trên các thiết bị thỏa mãn điều kiện tương thích của Google. Ứng dụng sẽ tự động lọc ra một danh sách các ứng dụng tương thích với thiết bị của người dùng, và nhà phát triển có thể giới hạn ứng dụng của họ chỉ dành cho những nhà mạng cố định hoặc những quốc gia cố định vì lý do kinh doanh. Nếu người dùng mua một ứng dụng mà họ cảm thấy không thích, họ được hoàn trả tiền sau 15 phút kể từ lúc tải về và một vài nhà mạng còn có khả năng mua giúp các ứng dụng trên Google Play, sau đó tính tiền vào trong hóa đơn sử dụng hàng tháng của người dùng. Đến tháng 9 năm 2012, có hơn 675.000 ứng dụng dành cho Android, và số lượng ứng dụng tải về từ Play Store ước tính đạt 25 tỷ.

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng Bộ phát triển phần mềm Android (SDK). SDK bao gồm một bộ đầy đủ các công cụ dùng để phát triển, gồm có công cụ gỡ lỗi, thư viện phần mềm, bộ giả lập điện thoại dựa trên QEMU, tài liệu hướng dẫn, mã nguồn mẫu, và hướng dẫn từng bước. Môi trường phát triển tích hợp (IDE) được hỗ trợ chính thức là Eclipse sử dụng phần bổ sung Android Development Tools (ADT). Các công cụ phát triển khác cũng có sẵn, gồm có Bộ phát triển gốc dành cho các ứng dụng hoặc phần mở rộng viết bằng C hoặc C++, Google App Inventor, một môi trường đồ họa cho những nhà lập trình mới bắt đầu, và nhiều nền tảng ứng dụng web di động đa nền tảng phong phú.

Để vượt qua những hạn chế khi tiếp cận các dịch vụ của Google do sự Kiểm duyệt Internet tại Cộng hòa Nhân dân Trung Hoa, các thiết bị Android bán tại Trung Quốc lục địa thường được điều chỉnh chỉ được sử dụng dịch vụ đã được duyệt.^[51]

1.8.3 Lịch sử phát triển:

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

Google mua lại Android, Inc. vào ngày 17 tháng 8 năm 2005, biến nó thành một bộ phận trực thuộc Google. Những nhân viên của chủ chốt của Android, Inc., gồm Rubin, Miner và White, vẫn tiếp tục ở lại công ty làm việc sau thương vụ này.^[25] Vào thời điểm đó không có nhiều thông tin về công ty, nhưng nhiều người đồn đoán rằng Google dự tính tham gia thị trường điện thoại di động sau bước đi này. Tại Google, nhóm do Rubin đứng đầu đã phát triển một nền tảng thiết bị di động phát triển trên nền nhân Linux. Google quảng bá nền tảng này cho các nhà sản xuất điện thoại và các nhà mạng với lời hứa sẽ cung cấp một hệ thống uyển chuyển và có khả năng nâng cấp. Google đã liên hệ với hàng loạt

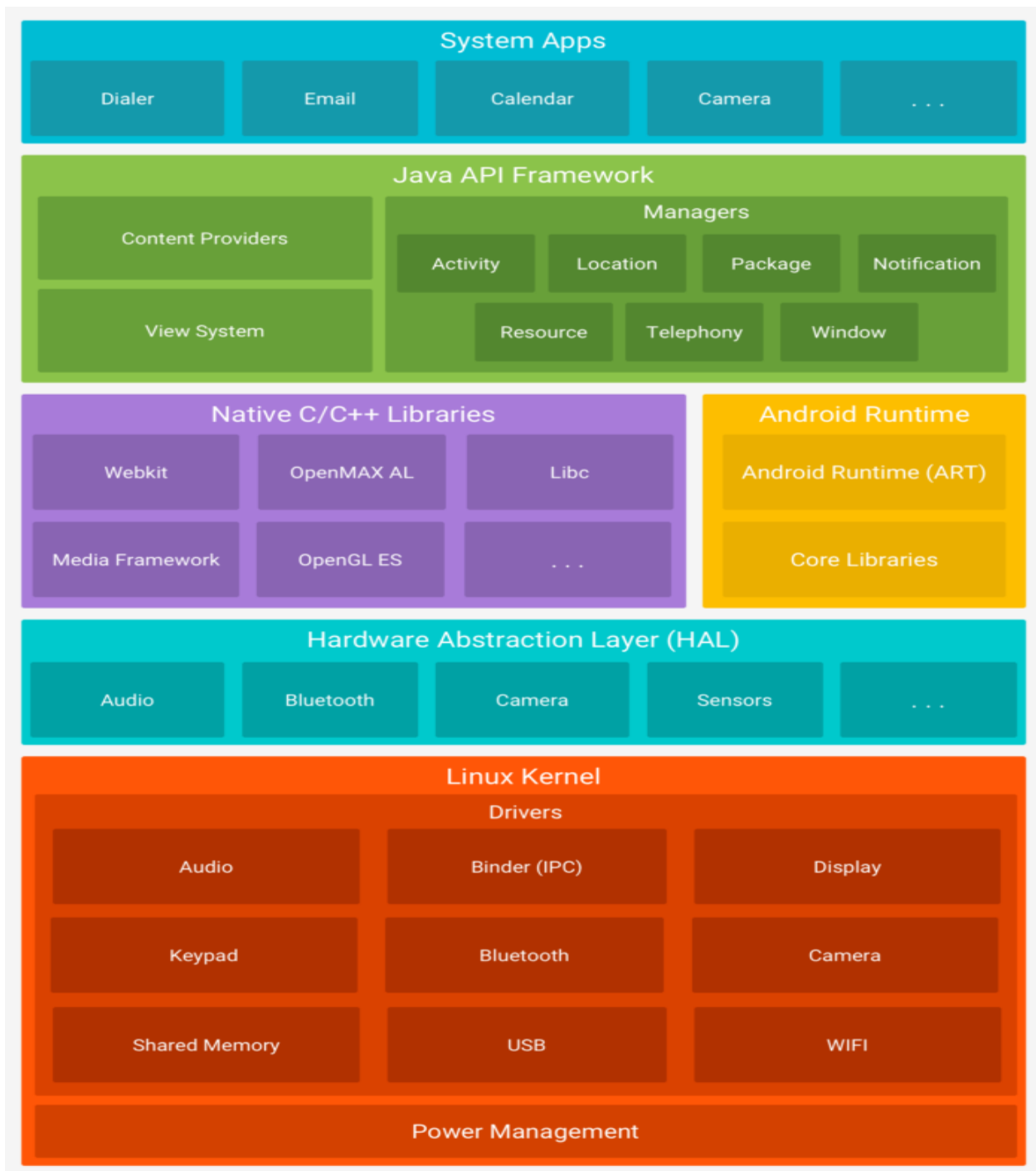
hãng phần cứng cũng như đối tác phần mềm, bán tin cho các nhà mạng rằng họ sẵn sàng hợp tác với các cấp độ khác nhau.

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

Ngày 5 tháng 11 năm 2007, Liên minh thiết bị cầm tay mở (Open Handset Alliance), một hiệp hội bao gồm nhiều công ty trong đó có Texas Instruments, Tập đoàn Broadcom, Google, HTC, Intel, LG, Tập đoàn Marvell Technology, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel và T-Mobile được thành lập với mục đích phát triển các tiêu chuẩn mở cho thiết bị di động. Cùng ngày, Android cũng được ra mắt với vai trò là sản phẩm đầu tiên của Liên minh, một nền tảng thiết bị di động được xây dựng trên nhân Linux phiên bản 2.6. Chiếc điện thoại chạy Android đầu tiên được bán ra là HTC Dream, phát hành ngày 22 tháng 10 năm 2008.¹ Biểu trưng của hệ điều hành Android mới là một con rôbốt màu xanh lá cây do hãng thiết kế Irina Blok tại California vẽ.

Từ năm 2008, Android đã trải qua nhiều lần cập nhật để dần dần cải tiến hệ điều hành, bổ sung các tính năng mới và sửa các lỗi trong những lần phát hành trước. Mỗi bản nâng cấp được đặt tên lần lượt theo thứ tự bảng chữ cái, theo tên của một món ăn tráng miệng; ví dụ như phiên bản 1.5 *Cupcake* (bánh bông lan nhỏ có kem) tiếp nối bằng phiên bản 1.6. Phiên bản mới nhất hiện nay là 9.0 Pie ra mắt vào tháng 8 năm 2018. Vào năm 2010, Google ra mắt loạt thiết bị Nexus—một dòng sản phẩm bao gồm điện thoại thông minh và máy tính bảng chạy hệ điều hành Android, do các đối tác phần cứng sản xuất. HTC đã hợp tác với Google trong chiếc điện thoại thông minh Nexus đầu tiên,¹ Nexus One. Kể từ đó nhiều thiết bị mới hơn đã gia nhập vào dòng sản phẩm này, như điện thoại Nexus 4 và máy tính bảng Nexus 10, lần lượt do LG và Samsung sản xuất. Google xem điện thoại và máy tính bảng Nexus là những thiết bị Android chủ lực của mình, với những tính năng phần cứng và phần mềm mới nhất của Android.

1.8.4 Kiến trúc Android



Chúng ta đã biết Android là một hệ điều hành nguồn mở dựa trên nhân Linux, được xây dựng theo ngăn xếp (stack) cho nhiều mảng rộng của thiết bị và các yếu tố hình thức nhất định. Nhìn vào hình ảnh minh họa ở trên chúng ta sẽ dễ nhìn nhận và hiểu hơn về kiểu

thiết kế này. Mỗi ngăn xếp (System Apps, Java API Framework ..) sẽ chứa một mảng các thành phần bên trong.

Chúng ta sẽ đi tìm hiểu về một vài stack trong sơ đồ trên (từ mức thấp đến mức cao):

- The Linux Kernel

Đây chính là nền tảng của Android, hệ điều hành được phát triển dựa trên nhân của hệ điều hành Linux. Việc sử dụng nhân Linux cho phép Android có nhiều đặc tính bảo mật và cho phép các nhà sản xuất thiết bị phát triển các bộ điều khiển (driver) cho thiết bị tốt nhất.

- Hardware Abstraction Layer (HAL)

Đây là ngăn có chức năng cung cấp các chuẩn giao tiếp phần mềm với phần cứng của thiết bị, giúp các ứng dụng trên Android có thể sử dụng được các phần cứng như camera, bluetooth... HAL chứa nhiều thư viện, nhiều module khác nhau, các thư viện này có chức năng điều khiển một phần cứng cụ thể. Ví dụ: module camera sẽ chịu trách nhiệm điều khiển các tác vụ liên quan đến camera gắn trên điện thoại/ máy tính bảng, khi ứng dụng chúng ta viết cần camera thì module camera sẽ được gọi để giúp ứng dụng của chúng ta thực hiện chức năng đó.

- Android Runtime

Với những thiết bị chạy hệ điều hành Android phiên bản 5.0 trở lên, thì mỗi ứng dụng sẽ chạy riêng một tiến trình với một instance (cụ thể) của Android Run Time. ART được viết để chạy được trên nhiều máy ảo ở cấp độ thấp bằng việc thực thi các file DEX. ART sẽ giúp các ứng dụng trên Android chạy nhanh hơn (chúng ta sẽ thấy rõ điều này khi sử dụng Android 5.0 trở lên và Android phiên bản nhỏ hơn, cùng cấu hình phần cứng). Đặc biệt khi bạn viết các ứng dụng yêu cầu nhiều hỗ trợ từ phần cứng như xử lý ảnh cần bộ nhớ, camera, tốc độ CPU thì bạn sẽ thấy sự khác biệt đáng kể cỡ nào.

Ngoài ra ART còn nhiều hỗ trợ rất tốt khác như: tối ưu hóa việc gom rác, giúp hệ thống giải phóng nhanh các phần nhớ không cần thiết. Hỗ trợ debug phần mềm tốt hơn các phiên bản cũ, báo chi tiết các lỗi crash ứng dụng ...

- Native C/C++ Libraries

Nhiều thành phần hệ thống nhân Android và các dịch vụ như ART và HAL như nêu ở trên được xây dựng từ native code, những dịch vụ này được viết bằng ngôn ngữ C/C++. Android cung cấp các Java Framework API để giúp ứng dụng viết bằng Java có thể sử dụng được nhưng thư viện/ dịch vụ Native này. Ví dụ: bạn có thể truy cập vào OpenGL ES nhờ vào Java OpenGL API để hỗ trợ việc vẽ và đa hình các chức năng đồ họa 2D/3D trong ứng dụng của bạn.

Nếu viết một ứng dụng bằng ngôn ngữ C/C++ bạn có thể sử dụng Android NDK thay vì Android SDK để truy cập một số thư viện native này một cách trực tiếp. Việc này sẽ giúp ứng dụng tận dụng được tối đa sức mạnh phần cứng, giúp ứng dụng chạy nhanh hơn như các ứng dụng về xử lý camera, xử lý video, truyền nhận video...

- Java API Framework

Đây là ngăn chứa các tính năng của Android mà bạn có thể sử dụng bằng các API viết bằng ngôn ngữ Java. Những API này được xây dựng thành các khối (block) cần thiết cho việc xây dựng ứng dụng của bạn, giúp bạn tạo ứng dụng một cách đơn giản và đơn giản trong việc sử dụng lại code. Cụ thể như:

1. Bộ quản lý tài nguyên: cung cấp cách để truy cập vào vị trí của strings, graphics, các file layout bạn viết bằng XML...
2. Quản lý notification: giúp các ứng dụng có thể hiển thị các màn hình thông báo trên thanh status bar của thiết bị.
3. Bộ quản lý Activity: quản lý các vòng đời của Activity trong các ứng dụng và cung cấp những chuyển đổi màn hình trong ngăn xếp (stack – khi tạo 1 activity nó sẽ được chứa vào ngăn xếp, mỗi ứng dụng khi chạy sẽ có một ngăn xếp để chứa các Activity của nó).
4. Content provider: cung cấp các cách chia sẻ dữ liệu, lưu trữ và xử lý dữ liệu...
5. ...

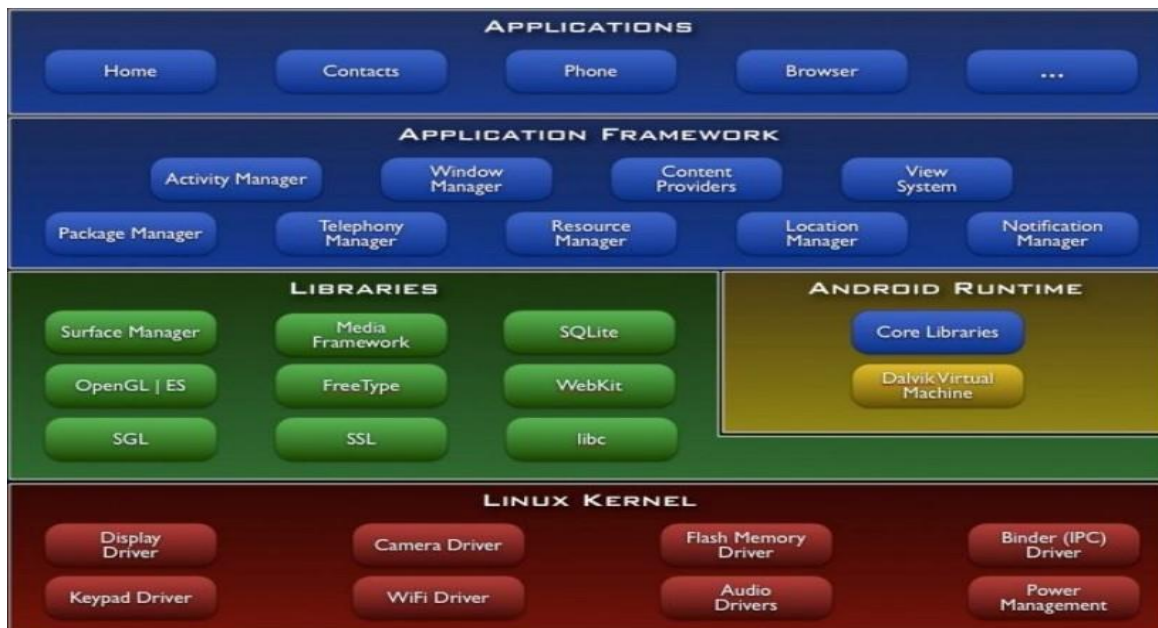
- System Apps

Đây là tầng chứa các ứng dụng cho người dùng cuối sử dụng, như các ứng dụng gọi điện, các ứng dụng của bạn khi xây dựng xong và cài đặt lên thiết bị nó sẽ nằm ở tầng này để người dùng sử dụng.

Ứng dụng cần chức năng nào, cần yêu cầu phần cứng nào sẽ gọi các tầng dưới hơn để yêu cầu và thực hiện.

1.8.5 Cấu trúc stack android

Mô hình sau thể hiện một cách tổng quát các thành phần của hệ điều hành Android. Mỗi một phần sẽ được đặc tả một cách chi tiết dưới đây



Hình 2. Cấu trúc stack hệ thống Android

1.8.5.1 Application framework

Bằng cách cung cấp một nền tảng phát triển mở, Android cung cấp cho các nhà phát triển khả năng xây dựng các ứng dụng cực kỳ phong phú và sáng tạo. Nhà phát triển được tự do tận dụng các thiết bị phần cứng, thông tin địa điểm truy cập, các dịch vụ chạy nền, thiết lập hệ thống báo động, thêm các thông báo để các thanh trạng thái, và nhiều, nhiều hơn nữa.

Nhà phát triển có thể truy cập vào các API cùng một khuôn khổ được sử dụng bởi các ứng dụng lõi. Các kiến trúc ứng dụng được thiết kế để đơn giản hóa việc sử dụng lại các thành phần; bất kỳ ứng dụng có thể xuất bản khả năng của mình và ứng dụng nào khác

sau đó có thể sử dụng những khả năng (có thể hạn chế bảo mật được thực thi bởi khuôn khổ). Cơ chế này cho phép các thành phần tương tự sẽ được thay thế bởi người sử dụng.

Cơ bản tất cả các ứng dụng là một bộ các dịch vụ và các hệ thống, bao gồm:

- Một tập hợp rất nhiều các View có khả năng kế thừa lẫn nhau dùng để thiết kế phân giao diện ứng dụng như: gridview, tableview, linearlayout,...
- Một “Content Provider” cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác (chẳng hạn như Contacts) hoặc là chia sẻ dữ liệu giữa các ứng dụng đó.
- Một “Resource Manager” cung cấp truy xuất tới các tài nguyên không phải là mã nguồn, chẳng hạn như: localized strings, graphics, and layout files.
- Một “Notifcation Manager” cho phép tất cả các ứng dụng hiển thị các custom alerts trong status bar. Activity Manager được dùng để quản lý chu trình sống của ứng dụng và điều hướng các activity.

1.8.5.2 Library

Android bao gồm một tập hợp các thư viện C/C++ được sử dụng bởi nhiều thành phần khác nhau trong hệ thống Android. Điều này được thể hiện thông qua nền tảng ứng dụng Android. Một số các thư viện cơ bản được liệt kê dưới đây:

- System C library: a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices.
- Media Libraries - based on PacketVideo's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.
- Surface Manager – Quản lý việc truy xuất vào hệ thống hiển thị.
- LibWebCore - a modern web browser engine which powers both the Android browser and an embeddable web view.
- SGL - the underlying 2D graphics engine.
- 3D libraries - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer.
- FreeType - bitmap and vector font rendering.

- SQLite - a powerful and lightweight relational database engine available to all applications.

1.8.5.3 Android Emulator

Android SDK và Plugin Eclipse được gọi là một Android Developer Tool (ADT). Các Android coder sẽ cần phải sử dụng công cụ IDE (Integrated Development Environment) này để phát triển, debugging và testing cho ứng dụng. Tuy nhiên, các coder cũng có thể không cần phải sử dụng IDE mà thay vào đó là sử dụng command line để biên dịch và tất nhiên là vẫn có Emulator như thường.

Android Emulator được trang bị đầy đủ hầu hết các tính năng của một thiết bị thật. Tuy nhiên, một số đã bị giới hạn như là kết nối qua cổng USB, camera và video, nghe phone, nguồn điện giả lập và bluetooth.

Android Emulator thực hiện các công việc thông qua một bộ xử lý mã nguồn mở, công nghệ này được gọi là QEMU (<http://bellard.org/qemu/>) được phát triển bởi Fabrice Bellard.

1.8.6 Bảo mật và tính riêng tư

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

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

Một nghiên cứu của công ty bảo mật Trend Micro đã liệt kê tình trạng lạm dụng dịch vụ trả tiền là hình thức phần mềm ác ý phổ biến nhất trên Android, trong đó tin nhắn SMS sẽ bị gửi đi từ điện thoại bị nhiễm đến một số điện thoại trả tiền mà người dùng không hề hay biết. Loại phần mềm ác ý khác hiển thị những quảng cáo không mong muốn và gây khó chịu trên thiết bị, hoặc gửi thông tin cá nhân đến bên thứ ba khi chưa được phép. De

dọa bảo mật trên Android được cho là tăng rất nhanh theo cấp số mũ; tuy nhiên, các kỹ sư Google phản bác rằng hiểm họa từ phần mềm ác ý và virus đã bị thổi phồng bởi các công ty bảo mật nhằm mục đích thương mại, và buộc tội ngành công nghiệp bảo mật đang lợi dụng sự sợ hãi để bán phần mềm diệt virus cho người dùng. Google vẫn giữ quan điểm rằng phần mềm ác ý thật sự nguy hiểm là cực kỳ hiếm, và một cuộc điều tra do F-Secure thực hiện cho thấy chỉ có 0,5% số phần mềm ác ý Android là len vào được cửa hàng Google Play.

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

Điện thoại thông minh Android có khả năng báo cáo vị trí của điểm truy cập Wi-Fi, phát hiện ra việc di chuyển của người dùng điện thoại, để xây dựng những cơ sở dữ liệu có chứa vị trí của hàng trăm triệu điểm truy cập. Những cơ sở dữ liệu này tạo nên một bản đồ điện tử để tìm vị trí điện thoại thông minh, cho phép chúng chạy các ứng dụng như Foursquare, Google Latitude, Facebook Places, và gửi những đoạn quảng cáo dựa trên vị trí. Phần mềm theo dõi của bên thứ ba như TaintDroid, một dự án nghiên cứu trong trường đại học, đôi khi có thể biết được khi nào thông tin cá nhân bị gửi đi từ ứng dụng đến các máy chủ đặt ở xa.

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

1.8.7 Phát triển

Android được Google tự phát triển riêng cho đến khi những thay đổi và cập nhật đã hoàn thiện, khi đó mã nguồn mới được công khai. Mã nguồn này, nếu không sửa đổi, chỉ chạy trên một số thiết bị, thường là thiết bị thuộc dòng Nexus. Có nhiều thiết bị có chứa những thành phần được giữ bản quyền do nhà sản xuất đặt vào thiết bị Android của họ.

1.8.7.1 Linux

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

trúc ARM. Người ta cũng hỗ trợ x86 thông qua dự án Android x86, và Google TV cũng sử dụng một phiên bản x86 đặc biệt của Android.

Nhân Linux dùng cho Android đã được Google thực hiện nhiều thay đổi về kiến trúc so với nhân Linux gốc. Android không có sẵn X Window System cũng không hỗ trợ các thư viện GNU chuẩn, nên việc chuyển các ứng dụng hoặc thư viện Linux có sẵn sang Android rất khó khăn. Các ứng dụng C đơn giản và SDL cũng được hỗ trợ bằng cách chèn những đoạn shim Java và sử dụng tương tự JNI, như khi người ta chuyển Jagged Alliance 2 sang Android.

Một số tính năng cũng được Google đóng góp ngược vào nhân Linux, đáng chú ý là tính năng quản lý nguồn điện có tên wakelock, nhưng bị những người lập trình chính cho nhân từ chối vì họ cảm thấy Google không có ý định sẽ tiếp tục bảo trì đoạn mã do họ viết. Google thông báo vào tháng 4 năm 2010 rằng họ sẽ thuê hai nhân viên để làm việc với cộng đồng nhân Linux, nhưng Greg Kroah-Hartman, người bảo trì nhân Linux hiện tại của nhánh ổn định, đã nói vào tháng 12 năm 2010 rằng ông ta lo ngại rằng Google không còn muốn đưa những thay đổi của mình vào Linux dòng chính nữa. Một số lập trình viên Android của Google tỏ ý rằng "nhóm Android thấy chán với quy trình đó," vì nhóm họ không có nhiều người và có nhiều việc khẩn cấp cần làm với Android hơn.

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

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

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

1.8.7.2 Quản lý bộ nhớ

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

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

1.8.7.3 Lịch cập nhật

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

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

Việc thiếu các hỗ trợ hậu mãi của nhà sản xuất và nhà mạng đã bị những nhóm người dùng và các trang tin công nghệ chỉ trích rất nhiều. Một số người viết còn nói rằng giới công nghiệp do cái lợi về tài chính đã cố tình *không* cập nhật thiết bị, vì nếu thiết bị hiện tại

không cập nhật sẽ thúc đẩy việc mua thiết bị mới, một thái độ được coi là "xúc phạm" *The Guardian* đã than phiền rằng phương cách phân phối bản cập nhật trở nên phức tạp chính vì những nhà sản xuất và nhà mạng đã cố tình làm nó như thế. Vào năm 2011, Google đã hợp tác cùng một số hãng công nghiệp và ra mắt "Liên minh Cập nhật Android", với lời hứa sẽ cập nhật thường xuyên cho các thiết bị trong vòng 18 tháng sau khi ra mắt. Tính đến năm 2012, người ta không còn nghe nhắc đến liên minh này nữa.

1.8.7.4 Cộng đồng mã nguồn mở

Android có một cộng đồng các lập trình viên và những người đam mê rất năng động. Họ sử dụng mã nguồn Android để phát triển và phân phối những phiên bản chỉnh sửa của hệ điều hành. Các bản Android do cộng đồng phát triển thường đem những tính năng và cập nhật mới vào nhanh hơn các kênh chính thức của nhà sản xuất/nhà mạng, tuy không được kiểm thử kỹ lưỡng cũng như không có đảm bảo chất lượng; cung cấp sự hỗ trợ liên tục cho các thiết bị cũ không còn nhận được bản cập nhật chính thức; hoặc mang Android vào những thiết bị ban đầu chạy một hệ điều hành khác, như HP Touchpad. Các bản Android của cộng đồng thường được root sẵn và có những điều chỉnh không phù hợp với những người dùng không rành rẽ, như khả năng ép xung hoặc tăng/giảm áp bộ xử lý của thiết bị. CyanogenMod là firmware của cộng đồng được sử dụng phổ biến nhất và hoạt động như một tổ chức của số đông khác.

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

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

1.8.8 Giấy phép phát hành

Mã nguồn của Android được cấp phép theo các giấy phép phần mềm mã nguồn mở tự do. Google đưa phần lớn mã nguồn (bao gồm cả các lớp mạng và điện thoại) theo Giấy phép Apache phiên bản 2.0, và phần còn lại, các thay đổi đối với nhân Linux, theo Giấy phép Công cộng GNU phiên bản 2. Liên minh Thiết bị cầm tay mở đã thực hiện các thay đổi trên nhân Linux, với mã nguồn lúc nào cũng công khai. Phần còn lại của Android được Google phát triển một mình, và mã nguồn chỉ được công bố khi phát hành một phiên bản mới. Thông thường Google cộng tác với một nhà sản xuất phần cứng để cung cấp một thiết bị 'chủ lực' (thuộc dòng Google Nexus) với phiên bản mới nhất của Android, sau đó phát hành mã nguồn sau khi thiết bị này được bán ra.

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

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

1.8.9 Bằng quyền và bằng phát minh

Cả Android và nhà sản xuất điện thoại Android đều bị dính líu đến nhiều vụ kiện tụng về bằng phát minh. Ngày 12 tháng năm 2010, Oracle kiện Google với cáo buộc vi phạm bản quyền và bằng phát minh liên quan đến ngôn ngữ lập trình Java. Oracle ban đầu muốn được đền bù thiệt hại 6,1 tỷ đô la Mỹ, nhưng bị tòa án liên bang Mỹ khước từ mức giá này và yêu cầu Oracle xem xét lại. Để đáp lại, Google đã đưa ra nhiều lý lẽ để bảo vệ, tuyên bố ngược lại là Android không hề vi phạm bằng phát minh hay bản quyền của Oracle, và rằng bằng phát minh của Oracle là vô hiệu, cùng một số lời bào chữa khác. Google nói rằng Android dựa trên Apache Harmony, một hiện thực phòng sạch của thư viện lớp Java (tức là xem hoạt động của thư viện, rồi lập trình lại bắt chước hoạt động đó nhưng không tham khảo hoặc lấy lại mã nguồn của thư viện gốc), rồi sau đó độc lập phát triển ra máy ảo đặt tên là Dalvik. Vào tháng 5 năm 2012 bồi thẩm đoàn của vụ án tuyên rằng Google không vi phạm bằng phát minh của Oracle, và sau đó thẩm phán tuyên rằng cấu trúc của Java API do Google sử dụng không đủ để được giữ bản quyền.

Ngoài vụ kiện trực tiếp chống lại Google, có nhiều cuộc chiến tranh thể mạng khác nhau gián tiếp chống lại Android bằng cách nhắm vào các nhà sản xuất thiết bị Android, nhằm làm nản lòng những nhà sản xuất muốn sử dụng nền tảng này do sự tăng chi phí để đưa thiết bị Android ra thị trường. Cả Apple và Microsoft đều đã kiện một số nhà sản xuất vì vi phạm bằng sáng chế, với cuộc chiến pháp lý chống Samsung dằng dai của Apple là vụ nổi bật nhất. Vào tháng 10 năm 2011 Microsoft nói rằng họ đã ký một số thỏa thuận cấp phép với 10 nhà sản xuất thiết bị Android, những hãng sản xuất 55% lợi nhuận toàn cầu của Android. Những công ty này có cả Samsung lẫn HTC. Vụ dàn xếp bằng phát minh của Samsung với Microsoft có một thỏa thuận rằng Samsung sẽ cung cấp thêm nguồn lực để phát triển và tiếp thị điện thoại chạy hệ điều hành Windows Phone của Microsoft.

Google đã bày tỏ công khai sự bất mãn của họ đối với hệ thống bằng phát minh tại Mỹ, buộc tội Apple, Oracle và Microsoft cố tình dìm Android thông qua các vụ kiện, thay vì phải sáng tạo và cạnh tranh bằng các sản phẩm và dịch vụ tốt hơn. Vào năm 2011-12, Google mua lại Motorola Mobility với giá 12,5 tỷ đô la Mỹ, một hành động được xem là phương cách để bảo vệ Android, vì Motorola Mobility nắm giữ hơn 17.000 bằng phát minh. Tháng 12 năm 2011 Google mua lại hơn một nghìn bằng sáng chế từ IBM.

1.8.10 Sự đón nhận người dùng

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

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

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

1.8.10.1 Thị phần và tỉ lệ sử dụng

Công ty nghiên cứu thị trường Canalsys đã ước tính trong quý 2 năm 2009 rằng Android có 2,8% thị phần điện thoại thông minh được bán ra toàn cầu. Đến quý 4 năm 2010 con số này tăng lên 33% thị phần, trở thành nền tảng điện thoại thông minh bán chạy hàng đầu. Đến quý 3 năm 2011 Gartner ước tính rằng hơn một nửa (52,5%) thị trường điện thoại thông minh thuộc về Android. Đến quý 3 năm 2012 Android đã có 75% thị phần điện thoại thông minh toàn cầu theo nghiên cứu của hãng IDC.

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

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

1.8.10.2 Máy tính bảng

Mặc dù thành công với điện thoại thông minh, việc sử dụng máy tính bảng Android vẫn còn chậm.

Một trong những nguyên nhân chính là tình huống con gà và quả trứng trong đó người tiêu dùng ngại mua máy tính bảng Android cho thiếu các ứng dụng máy tính bảng chất lượng cao, còn các lập trình viên thì ngại mất thời gian và tiền bạc để phát triển ứng dụng máy tính bảng cho đến khi nào thị phần của chúng đủ lớn. Nội dung và "hệ sinh thái" ứng dụng đã chứng tỏ rằng nó quan trọng hơn nhiều so với việc " nạp vào chạy" (sức mạnh xử lý phần cứng) khi nói đến máy tính bảng. Do thiếu các ứng dụng dành riêng cho máy tính bảng vào năm 2011, các máy tính bảng Android đời đầu đã phải sử dụng các ứng dụng sẵn có trên điện thoại thông minh dù hiển thị rất kém trên màn hình cỡ lớn, trong khi sự thống trị của Apple iPad được củng cố bởi một số lượng lớn ứng dụng iOS dành riêng cho máy tính bảng.

Mặc dù sự hỗ trợ từ ứng dụng chỉ mới ở mức sơ khai, một lượng đáng kể máy tính bảng Android (cùng với các loại máy tính bảng sử dụng các hệ điều hành khác, như HP TouchPad và BlackBerry Playbook) vẫn được tung ra thị trường trong nỗ lực cạnh tranh với sự thành công của iPad. *InfoWorld* đã nói rằng một số nhà sản xuất Android thoát đầu xem các máy tính bảng của họ như là một "thương vụ Frankenphone", một cơ hội đầu tư thấp ngắn hạn bằng cách đặt một hệ điều hành Android tối ưu cho điện thoại thông minh

(trước khi Android 3.0 "Honeycomb" dành cho máy tính bảng ra đời) trên một thiết bị mà không để ý tới giao diện người dùng. Cách làm này, như với Dell Streak, không những thất bại trong việc lôi kéo người dùng mà còn làm ảnh hưởng đến danh tiếng ban đầu của máy tính Android. Hơn nữa, một số máy tính bảng Android như Motorola Xoom được định giá bằng hoặc cao hơn iPad, làm tổn hại sức bán. Một ngoại lệ đó là Amazon Kindle Fire, được phát triển theo cách tiếp cận "chờ mà xem" dựa trên giá rẻ và khả năng truy cập vào hệ sinh thái ứng dụng và nội dung của Amazon.com.

Hiện tượng này bắt đầu thay đổi vào năm 2012 với sự ra mắt của Nexus 7 giá rẻ và một cú hích của Google dành cho các lập trình viên nhằm thúc đẩy họ viết các ứng dụng cho máy tính bảng tốt hơn. Máy tính bảng Android được kỳ vọng sẽ vượt qua iPad trong vòng một vài năm.

1.8.10.3 Tình trạng ăn cắp bản quyền sử dụng

Đã có những lo ngại về việc các ứng dụng trả tiền của Android quá dễ bị ăn cắp. Trong một cuộc phỏng vấn vào tháng 5 năm 2012 với Eurogamer, nhà phát triển Football Manager nói rằng tỷ lệ người chơi ăn cắp so với người chơi trả tiền là 9:1 với trò chơi Football Manager Handheld. Tuy nhiên, không phải tất cả các lập trình viên đều cho rằng tình trạng ăn cắp là một vấn đề; ví dụ như vào tháng 7 năm 2012 các lập trình viên của trò chơi Wind-up Knight nói rằng mức độ ăn cắp trò chơi của họ chỉ khoảng 12%, và phần lớn sản phẩm ăn cắp đến từ Trung Quốc, nơi người ta không thể mua ứng dụng từ Google Play.

Vào năm 2010, Google phát hành một công cụ để xác nhận việc mua bán để sử dụng trong các ứng dụng, nhưng các lập trình viên than phiền rằng như vậy là chưa đủ và quá dễ để bẻ khóa. Google trả lời rằng công cụ, đặc biệt là bản phát hành đầu tiên, chỉ có ý định làm nền tảng mẫu cho lập trình viên điều chỉnh và xây dựng theo yêu cầu, chứ không phải một giải pháp bảo mật hoàn chỉnh. Vào năm 2012 Google phát hành một tính năng trong Android 4.1 để mã hóa các ứng dụng trả tiền chỉ hoạt động trên thiết bị đã mua ứng dụng đó, nhưng tính năng này đã bị hoãn do vấn đề về kỹ thuật.

1.8.11 Các thiết bị ngoài điện thoại và máy tính bảng

Bản chất mở và cho phép thay đổi của Android giúp nó xuất hiện trên các thiết bị điện tử khác, như laptop và netbook, smartbook TV thông minh (Google TV) và máy ảnh (Nikon Coolpix S800c và Galaxy Camera). Hơn thế nữa, hệ điều hành Android còn được ứng dụng trong kính mắt thông minh (Project Glass), đồng hồ đeo tay, tai nghe, đầu CD và DVD cho xe hơi, gương soi, máy nghe nhạc bỏ túi và điện thoại để bàn và VoIP. Ouya, một máy trò chơi điện tử chạy Android, đã trở thành một trong những chiến dịch khởi động thành công nhất, gây quỹ được 8,5 triệu đô la Mỹ để phát triển, tiếp sau đó là các máy trò chơi điện tử dựa trên Android như Project Shield của NVIDIA.

Vào năm 2011, Google đã trình diễn "Android@Home", một công nghệ tự động hóa gia đình, sử dụng Android để điều khiển nhiều thiết bị gia dụng như công tắc điện, ổ cắm và thiết bị điều khiển nhiệt độ trong nhà. Chiếc đèn mẫu được quảng cáo là có thể được điều khiển từ điện thoại hoặc máy tính bảng Android, nhưng trường nhóm Android Andy

Rubin vẫn cẩn trọng cho rằng "tắt mở bóng đèn không phải là việc gì mới," ám chỉ nhiều dịch vụ tự động hóa gia đình đã gặp thất bại trước đây. Ông nói rằng Google có suy nghĩ tham vọng hơn và dự định của công ty là sử dụng vị trí của mình như một nhà cung cấp dịch vụ đám mây để mang sản phẩm Google đến gia đình của khách hàng.¹

1.9 Android studio:

1.9.1 Khái niệm:

Android Studio là một phần mềm bao gồm các bộ công cụ khác nhau dùng để phát triển ứng dụng chạy trên thiết bị sử dụng hệ điều hành Android như các loại điện thoại smartphone, các tablet... Android Studio được đóng gói với một bộ code editor, debugger, các công cụ performance tool và một hệ thống build/deploy (trong đó có trình giả lập simulator để giả lập môi trường của thiết bị điện thoại hoặc tablet trên máy tính) cho phép các lập trình viên có thể nhanh chóng phát triển các ứng dụng từ đơn giản tới phức tạp.

Việc xây dựng một ứng dụng mobile (hoặc tablet) bao gồm rất nhiều các công đoạn khác nhau. Đầu tiên chúng ta sẽ viết code ứng dụng sử dụng máy tính cá nhân hoặc laptop. Sau đó chúng ta cần build ứng dụng để tạo file cài đặt. Sau khi build xong thì chúng ta cần copy file cài đặt này vào thiết bị mobile (hoặc table) để tiến hành cài đặt ứng dụng và chạy kiểm thử (testing). Bạn thử tưởng tượng nếu với mỗi lần viết một dòng code bạn lại phải build ứng dụng, cài đặt trên điện thoại hoặc tablet và sau đó chạy thử thì sẽ vô cùng tốn thời gian và công sức. Android Studio được phát triển để giải quyết các vấn đề này. Với Android Studio tất cả các công đoạn trên được thực hiện trên cùng một máy tính và các quy trình được tinh gọn tới mức tối giản nhất.

1.9.2 Chức năng:

Chức năng của Android Studio là cung cấp giao diện để tạo các ứng dụng và xử lý phần lớn các công cụ quản lý file phức tạp đằng sau hậu trường. Ngôn ngữ lập trình được sử dụng ở đây là Java và được cài đặt riêng trên thiết bị của bạn. Android Studio rất đơn giản, bạn chỉ cần viết, chỉnh sửa và lưu các dự án của mình và các file trong dự án đó. Đồng thời, Android Studio sẽ cấp quyền truy cập vào Android SDK.

1.10 Google Firebase:

1.10.1 Lịch sử

Firebase phát triển từ Envolv, một startup trước đây được thành lập bởi James Tamplin và Andrew Lee vào năm 2011. Envolv cung cấp cho các nhà phát triển một API cho phép tích hợp chức năng trò chuyện trực tuyến vào trang web của họ. Sau khi phát hành dịch vụ trò chuyện, Tamplin và Lee thấy rằng nó đang được sử dụng để truyền dữ liệu

ứng dụng không phải là tin nhắn trò chuyện. Các nhà phát triển đã sử dụng Envolv để đồng bộ hóa dữ liệu ứng dụng như trạng thái trò chơi trong thời gian thực trên người dùng của họ. Tamplin và Lee quyết định tách hệ thống trò chuyện và kiến trúc thời gian thực cung cấp cho nó. Họ thành lập Firebase như một công ty riêng biệt vào tháng 9 năm 2011 và nó được ra mắt công chúng vào tháng 4 năm 2012.

Sản phẩm đầu tiên của Firebase là Cơ sở dữ liệu thời gian thực Firebase, API đồng bộ hóa dữ liệu ứng dụng trên các thiết bị iOS, Android và Web và lưu trữ trên đám mây của Firebase. Sản phẩm hỗ trợ các nhà phát triển phần mềm xây dựng các ứng dụng hợp tác, thời gian thực.

Vào tháng 5 năm 2012, một tháng sau khi ra mắt phiên bản beta, Firebase đã huy động được 1,1 triệu đô la tài trợ hạt giống từ các nhà đầu tư mạo hiểm Flybridge Capital Partners, Greylock Partners, Founder Collective và New Enterprise Associates.

Vào tháng 6 năm 2013, công ty đã huy động thêm 5,6 triệu đô la trong tài trợ Series A từ các nhà đầu tư mạo hiểm Union Square Ventures và Flybridge Capital Partners.

Năm 2014, Firebase đã ra mắt hai sản phẩm. Firebase Hosting và Xác thực Firebase. Điều này định vị công ty như một phụ trợ di động như một dịch vụ.

Vào tháng 10 năm 2014, Firebase đã được Google mua lại.

Vào tháng 10 năm 2015, Google đã mua lại Divshot để hợp nhất nó với nhóm Firebase.

Vào tháng 5 năm 2016, tại Google I / O, hội nghị nhà phát triển thường niên của công ty, Firebase đã mở rộng dịch vụ của mình để trở thành một nền tảng hợp nhất cho các nhà phát triển di động. Firebase hiện tích hợp với nhiều dịch vụ khác của Google, bao gồm Google Cloud Platform, AdMob và Google Ads để cung cấp các sản phẩm và quy mô rộng hơn cho các nhà phát triển. Google Cloud Messaging, dịch vụ Google để gửi thông báo đẩy tới các thiết bị Android, được thay thế bởi một sản phẩm Firebase, Firebase Cloud Messaging, đã thêm chức năng gửi thông báo đẩy tới các thiết bị iOS và Web.

Vào tháng 1 năm 2017, Google đã mua lại Fabric và Crashlytics từ Twitter để thêm các dịch vụ đó vào Firebase.

Vào tháng 10 năm 2017, Firebase đã ra mắt Cloud Firestore, một cơ sở dữ liệu tài liệu thời gian thực với tư cách là sản phẩm kế thừa cho Cơ sở dữ liệu thời gian thực Firebase ban đầu.

1.10.2 Tổng quan về Firebase:

Firebase là một dịch vụ cơ sở dữ liệu thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.

2.4.2. Lợi ích của Google Firebase:

- *Triển khai ứng dụng cực nhanh*: Firebase cung cấp cho bạn khá nhiều các API, hỗ trợ đa nền tảng giúp bạn tiết kiệm thời gian quản lý cũng như đồng bộ dữ liệu cung cấp hosting, hỗ trợ xác thực người dùng thì việc triển khai ứng dụng sẽ giảm được rất nhiều thời gian phát triển.

- *Bảo mật*: Với việc sử dụng các kết nối thông qua giao thức bảo mật SSL hoạt động trên nền tảng cloud đồng thời cho phép phân quyền người dùng database bằng cú pháp javascript cũng nâng cao hơn nhiều độ bảo mật cho ứng dụng của bạn.

- *Sự ổn định*: Firebase hoạt động dựa trên nền tảng cloud cung cấp bởi Google do đó hãy yên tâm về việc một ngày đẹp trời nào đó server ngừng hoạt động hay như DDOS hoặc là tốc độ kết nối như rùa bò. Một điều đáng lưu ý nữa đó là do hoạt động trên nền tảng Cloud vì vậy việc nâng cấp hay bảo trì server cũng diễn ra rất đơn giản mà không cần phải dùng server.

1.10.3 Dịch vụ

1.10.3.1 Phân tích Firebase

Firebase Analytics là một giải pháp đo lường ứng dụng miễn phí, cung cấp cái nhìn sâu sắc về việc sử dụng ứng dụng và sự tham gia của người dùng.

1.10.3.2 Tin nhắn đám mây Firebase

Trước đây được gọi là Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM) là một giải pháp đa nền tảng cho các tin nhắn và thông báo cho Android, iOS và các ứng dụng web, tính đến năm 2016 có thể được sử dụng miễn phí.

1.10.3.3 Firebase Auth

Firebase Auth là một dịch vụ có thể xác thực người dùng chỉ sử dụng mã phía máy khách. Nó hỗ trợ các nhà cung cấp đăng nhập xã hội Facebook, GitHub, Twitter và Google (và Google Play Games). Ngoài ra, nó bao gồm một hệ thống quản lý người dùng, theo đó các nhà phát triển có thể cho phép xác thực người dùng bằng email và mật khẩu đăng nhập được lưu trữ với Firebase.

1.10.3.4 Cơ sở dữ liệu thời gian thực Firebase

Firebase cung cấp một cơ sở dữ liệu thời gian thực và phụ trợ như một dịch vụ. Dịch vụ này cung cấp cho các nhà phát triển ứng dụng một API cho phép dữ liệu ứng dụng được đồng bộ hóa giữa các máy khách và được lưu trữ trên đám mây của Firebase. Công ty cung cấp các thư viện client cho phép tích hợp với Android, iOS, JavaScript, Java, Objective-C, Swift và Node.js ứng dụng. Cơ sở dữ liệu cũng có thể truy cập thông qua API REST và các ràng buộc cho một số khung

JavaScript như AngularJS, React, Ember.js và Backbone.js. API REST sử dụng giao thức Sự kiện do máy chủ gửi, là API để tạo kết nối HTTP để nhận thông báo đẩy từ máy chủ. Các nhà phát triển sử dụng cơ sở dữ liệu thời gian thực có thể bảo mật dữ liệu của họ bằng cách sử dụng các quy tắc bảo mật được thi hành bởi phía máy chủ của công ty.¹

1.10.3.5 Firebase Cloud Firestore

Vào ngày 31 tháng 1 năm 2019, Cloud Firestore đã chính thức được đưa ra khỏi phiên bản beta, biến nó thành một sản phẩm chính thức của dòng sản phẩm Firebase. Nó là sự kế thừa cho hệ thống cơ sở dữ liệu gốc của Firebase, Cơ sở dữ liệu thời gian thực và cho phép các tài liệu và trường lồng nhau thay vì chế độ xem dạng cây được cung cấp trong Cơ sở dữ liệu thời gian thực.

1.10.3.6 Lưu trữ Firebase

Firebase Storage cung cấp tải lên và tải xuống tệp an toàn cho các ứng dụng Firebase, bất kể chất lượng mạng. Nhà phát triển có thể sử dụng nó để lưu trữ hình ảnh, âm thanh, video hoặc nội dung khác do người dùng tạo. Lưu trữ Firebase được hỗ trợ bởi Google Cloud Storage .

1.10.3.7 Firebase Hosting

Firebase Hosting là dịch vụ lưu trữ web tĩnh và động được ra mắt vào ngày 13 tháng 5 năm 2014. Nó hỗ trợ lưu trữ các tệp tĩnh như CSS , HTML , JavaScript và các tệp khác, cũng như hỗ trợ thông qua Chức năng đám mây. Dịch vụ này cung cấp các tệp qua mạng phân phối nội dung (CDN) thông qua HTTP Secure (HTTPS) và mã hóa Lớp cổng bảo mật (SSL). Các đối tác của Firebase với Fastly, CDN, để cung cấp cho Firebase Hosting CDN sao lưu. Công ty nói rằng Firebase Hosting phát triển từ các yêu cầu của khách hàng; các nhà phát triển đã sử dụng Firebase cho cơ sở dữ liệu thời gian thực của mình nhưng cần một nơi để lưu trữ nội dung của họ.

1.10.3.8 Bộ ML

ML Kit là một hệ thống học máy di động dành cho các nhà phát triển được ra mắt vào ngày 8 tháng 5 năm 2018 trong bản beta trong Google I / O 2018 . API của ML Kit có nhiều tính năng bao gồm nhận dạng văn bản, phát hiện khuôn mặt, quét mã vạch, dán nhãn hình ảnh và nhận dạng cột mốc. Nó hiện có sẵn cho các nhà phát triển iOS hoặc Android . Bạn cũng có thể nhập các mô hình TensorFlow Lite của riêng mình , nếu API đã cho không đủ. API có thể được sử dụng trên thiết bị hoặc trên đám mây.

1.10.3.9 Tai nạn

Báo cáo sự cố tạo báo cáo chi tiết về các lỗi trong ứng dụng. Các lỗi được nhóm thành các cụm có dấu vết ngăn xếp tương tự và được xử lý theo mức độ nghiêm trọng của tác động đối với người dùng ứng dụng. Ngoài các báo cáo tự động, nhà phát triển có thể ghi nhật ký các sự kiện tùy chỉnh để giúp nắm bắt các bước dẫn đến sự cố. Trước khi có được Crashlytics , Firebase đã sử dụng Báo cáo sự cố Firebase của riêng mình.

1.10.3.10 Hiệu suất

Hiệu suất Firebase cung cấp cái nhìn sâu sắc về hiệu suất của ứng dụng và độ trễ mà người dùng ứng dụng trải nghiệm.

1.10.3.11 Phòng thí nghiệm thử nghiệm Firebase cho Android và iOS

Firebase Test Lab cho Android và iOS cung cấp cơ sở hạ tầng dựa trên đám mây để thử nghiệm ứng dụng Android và iOS. Với một thao tác, nhà phát triển có thể bắt đầu thử nghiệm ứng dụng của mình trên nhiều loại thiết bị và cấu hình thiết bị. Kết quả kiểm tra, bao gồm cả nhật ký, video và ảnh chụp màn hình, được cung cấp trong dự án trong bảng

điều khiển Firebase. Ngay cả khi nhà phát triển chưa viết bất kỳ mã kiểm tra nào cho ứng dụng của họ, Test Lab có thể tự động thực hiện ứng dụng, tìm kiếm sự cố. Lab Lab cho iOS hiện đang trong giai đoạn thử nghiệm.

1.10.3.12 Quản trị viên

Admob là một sản phẩm của Google tích hợp với đối tượng Firebase.

1.10.3.13 Liên kết động Firebase

Liên kết Firebase động là các URL thông minh tự động thay đổi hành vi của chúng để cung cấp "trải nghiệm khả dụng tốt nhất" trên nhiều nền tảng, chẳng hạn như trình duyệt web trên máy tính để bàn, cũng như iOS và Android và liên kết sâu với ứng dụng di động. Liên kết động hoạt động trong tất cả các cài đặt ứng dụng: nếu người dùng mở Liên kết động trên iOS hoặc Android và ứng dụng chưa được cài đặt, người dùng sẽ được nhắc cài đặt ứng dụng trước. Sau khi cài đặt, ứng dụng sẽ bắt đầu chạy và có thể truy cập liên kết. [\[39\]](#)

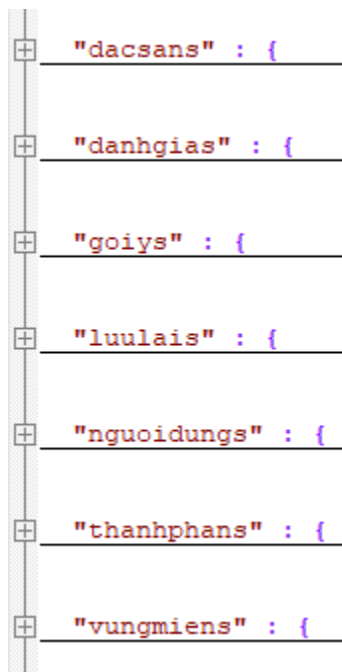
CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

1.11 Đặc tả yêu cầu hệ thống

Hiện nay dịch vụ du lịch ngày càng đầy mạnh, khách du lịch nước ngoài và trong nước ngày một nhiều hơn, việc đến một địa điểm nào đó ngoài vui chơi giải trí còn phải quan tâm đến thức ăn nước uống đặc sản của nơi đó nhưng việc tìm hiểu các món ăn nước uống đặc sản một nơi thì mất khá nhiều thời gian và nhiều thông tin không tìm thấy hoặc tìm thấy sai lệch với thực tế. Cần có một phần mềm đáp ứng được các nhu cầu cơ bản như: Tìm kiếm thức ăn nước uống theo từng khu vực, giá tiền, thành phần, xem chi tiết một món ăn hoặc nước uống nào đó, lưu giữ lại những món ăn nước uống mình thích...

1.12 Phân tích hệ thống

1.12.1 Mô hình cơ sở dữ liệu:



Hình 0.3 Cơ sở dữ liệu tổng quát

```

"luulais" : {
  "ID_User" : {
    "monans" : {
      "Ramdom_Key" : {
        "mabaiviet" : String,
        "makhuvuc" : String
      }
    },
    "nuocuongs" : {
      "Ramdom_key" : {
        "mabaiviet" : String,
        "makhuvuc" : String
      }
    }
  }
}

```

Hình 0.4 Node lưu lại

```

"thanhphans" : {
  "monans" : [ {
    "mathanhphan" : String,
    "tenthanhphan" : String
  } ],
  "nuocuongs" : [ {
    "mathanhphan" : String,
    "tenthanhphan" : String
  } ]
}

```

Hình 0.2 Node thành phần

```

"vungmiens" : {
  "KVMN" : {
    "ID_KhuVuc" : {
      "lat" : Double,
      "lng" : Double,
      "tentinh" : String
    }
  }
}

```

Hình 0.1 Node vùng miền

```

"nguoidungs" : {
  "ID_User" : {
    "avatar" : String,
    "email" : String,
    "gioitinh" : Boolean,
    "hoten" : String,
    "loaidangnhap" : String,
    "ngaysinh" : Long,
    "quanly" : Boolean,
    "quequan" : String
  }
}

```

Hình 0.5 Node người dùng

```

"goiys" : {
  "ID_ThanhPhan" : [ String ],
  "ID_ThanhPhan" : [ String ]
}

```

Hình 0.7 Node gợi ý

```

"danhgia" : {
  "ID_Bai_Viet" : {
    "ID_User" : {
      "binhluan" : String,
      "ngaydanhgia" : Long,
      "sosao" : Long,
      "tennguoidanhgia" : String
    }
  }
}

```

Hình 0.6 Node đánh giá

```

"dacsans" : {
  "KVMN" : {
    "ID_Khu_Vuc" : {
      "monans" : {
        "ID_User" : {
          "ID_Bai_Viet" : {
            "diachi" : String,
            "ghichu" : String,
            "giathamkhao" : Long,
            "gioithieu" : String,
            "hinh" : [ String ],
            "ten" : String,
            "thanhphan" : [ String ]
          }
        }
      }
    },
    "nuocuongs" : {
      "ID_User" : {
        "ID_Bai_Viet" : {
          "diachi" : String,
          "ghichu" : String,
          "giathamkhao" : Long,
          "gioithieu" : String,
          "hinh" : [ String ],
          "ten" : String,
          "thanhphan" : [ String ]
        }
      }
    }
  }
}

```

Hình 0.8 Node đặc sản

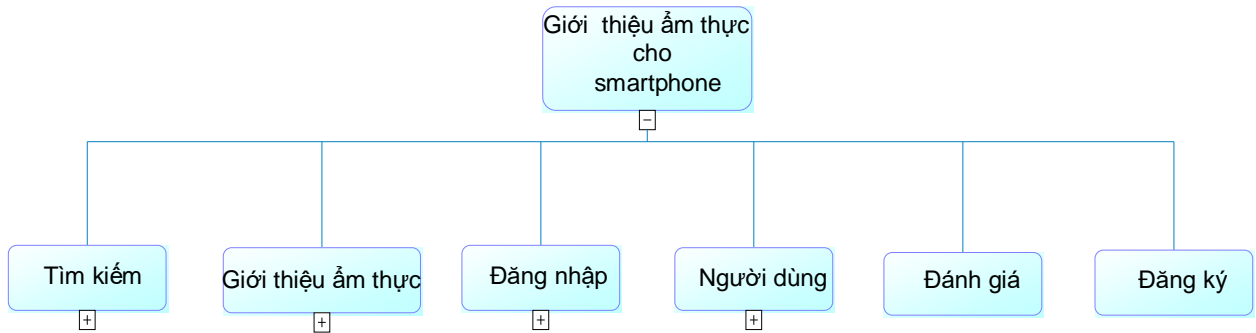
1.12.2 Chú thích cơ sở dữ liệu:

STT	Tên từ khóa	Ý nghĩa
1	ID_User	ID của user khi tạo tài khoản
2	ID_KhuVuc	ID của khu vực do người lập trình đặt ra
3	ID_Bai_Viet	ID bài viết được đặt theo cấu trúc <ID_User đăng bài>_ dacsans_<Loại(monans hoặc nuocuongs> _<ID_KhuVuc>_<Số thứ tự>
4	ID_ThanhPhan	ID thành phần do người lập trình đặt ra
5	Ramdom_Key	Mã tạo ngẫu nhiên của phần mềm android push dữ liệu lên

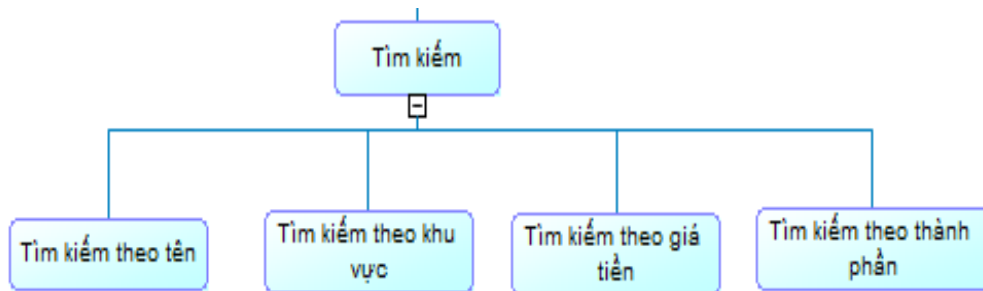
Bảng 0.1 Chú thích các key trong cơ sở dữ liệu

CHƯƠNG 4: ỨNG DỤNG GIỚI THIỆU ẨM THỰC CHO SMARTPHONE

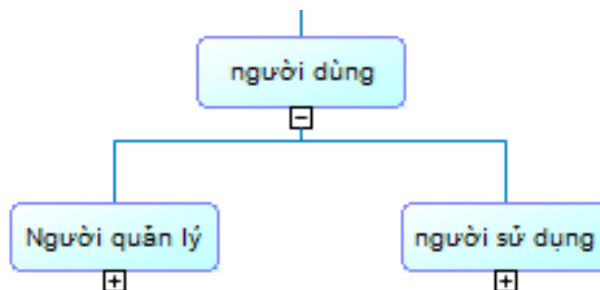
4.1 Sơ đồ phân rã chức năng phần mềm



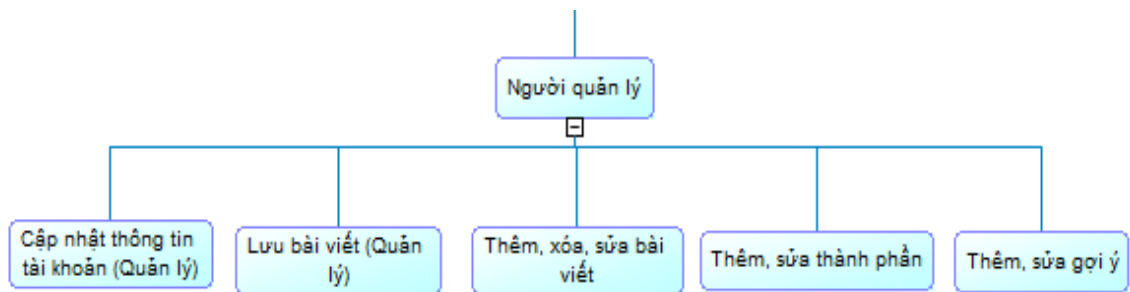
Hình 4. 1 Sơ đồ phân rã chức năng phần mềm



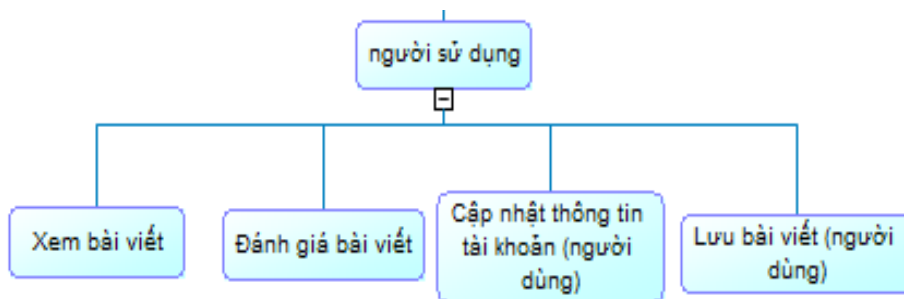
Hình 4. 2 Sơ đồ phân rã chức năng tìm kiếm



Hình 4. 3 Sơ đồ phân rã chức năng người dùng



Hình 4. 4 Sơ đồ phân rã chức năng người quản lý



Hình 4. 5 Sơ đồ phân rã chức năng người sử dụng

4.2 Phân tích hệ thống chức năng chính phần mềm

4.2.1 Đăng ký

- Tác vụ này dành cho khách hàng muốn đăng kí thông tin tài khoản
- Luồng dữ liệu vào:
- + Gmail khách hàng

- + Tên khách hàng
- + Mật khẩu
- + Quê quán
- + Giới tính
- + Ngày sinh

Xử lý hệ thống:

Các thông tin khách hàng: được nhập từ bàn phím. Khi khách hàng chọn mục đồng ý thì các thông tin khách hàng sẽ được inport vào database

Luồng dữ liệu ra:

- Thông báo kết quả
 - + Thông báo chưa nhập đầy đủ các thông tin đăng kí
 - + Thông báo tài khoản đã tồn tại
- Thông báo đã đăng nhập thành công. Và vào màn hình chính của chương trình
- Phát thảo giao diện

4.2.2 Thêm, Sửa, Xóa Món Ăn cho Quản trị viên

- Tác vụ này cho phép quản trị viên thêm, sửa, xóa vào món ăn cho từng khu vực

Dữ liệu vào

- + Loại thức ăn
- + Tên bài viết
- + Giá
- + Ghi chú
- + Thành phần
- + Hình ảnh
- + Giới thiệu
- + Khu vực

Xử lý hệ thống:

Các thông tin món ăn được nhập từ bàn phím. Khi quản trị viên chọn mục đăng thì các thông tin khách hàng sẽ được inport vào database

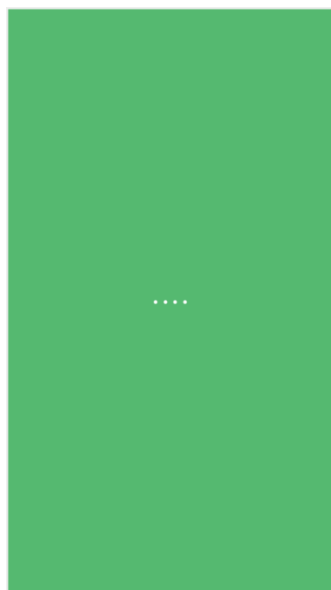
Luồng dữ liệu ra:

Thông báo kết quả

- + Thông báo chưa nhập đầy đủ các thông tin đăng kí
- + Hiện giao diện

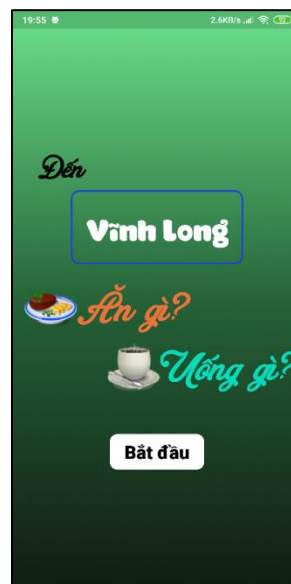
4.3 Ứng dụng giới thiệu ẩm thực cho smartphone

- Màn hình splash screen:



Hình 4. 6 Giao diện splash screen

- Màn hình chọn khu vực:



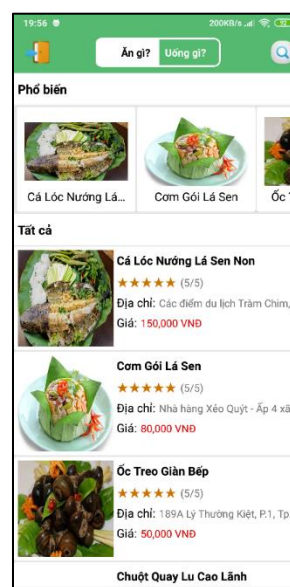
Hình 4. 7 Giao diện chọn khu vực

- Danh sách các khu vực:

Vĩnh Long	»
Đồng Tháp	»
Cà Mau	»
Hậu Giang	»
Bạc Liêu	»
Cần Thơ	»
Long An	»
An Giang	»
Tiền Giang	»
Kiên Giang	»
Bến Tre	»
Trà Vinh	»
Sóc Trăng	»

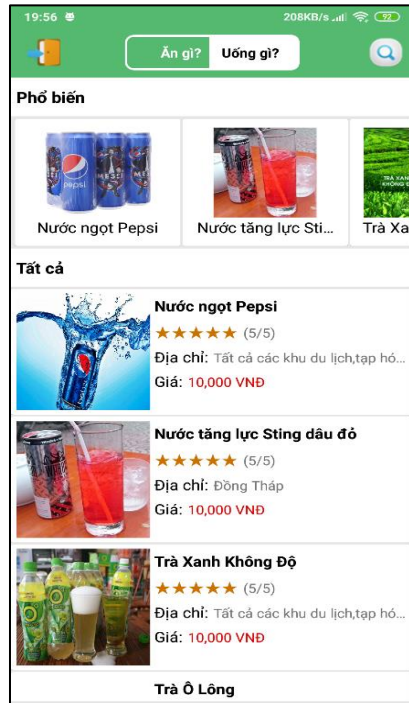
Hình 4. 8 Danh sách các khu vực

- Giao diện ăn gì:



Hình 4. 9 Giao diện ăn gì

- *Giao diện uống gì:*



Giao diện “uống gì” danh sách các bài viết về ước uống cho người dùng lựa chọn

Hình 4. 10 Giao diện uống gì

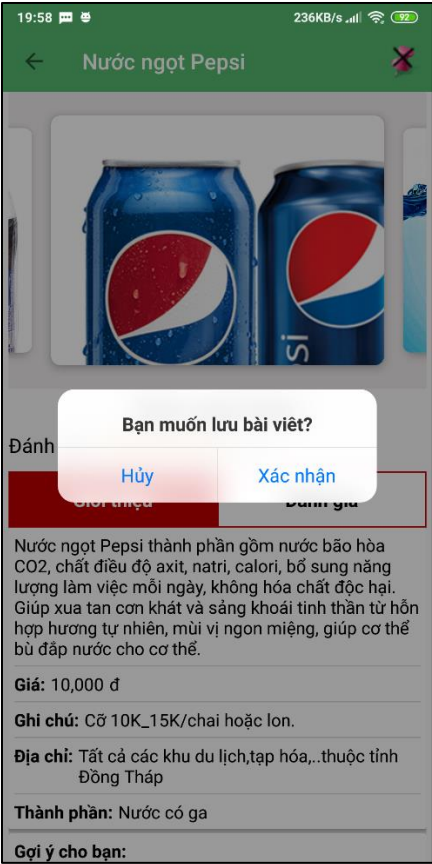
- *Giao diện chi tiết bài viết:*



Hình 4. 11 Đánh giá bài viết



Hình 4. 12 Chi tiết bài viết



Hình 4. 13 Lưu bài viết

- Tìm bài viết:

19:56 7.8KB/s 72

← Tìm kiếm bài viết

Khu vực

Vĩnh Long

Loại

☒ Thức ăn ☐ Nước uống

Tên món ăn

Nhập tên món ăn

Thành phần

Tất cả

Giá tiền tối đa

1,698,718 đ

LỌC

Cá Tai Tượng Chiên Xù

★★★★☆ (11/15)

Địa chỉ: Cù Lao An Bình, Vĩnh Long

Giá: 150,000 VNĐ

Lẩu Gà Nòi

★★★★★ (5/5)

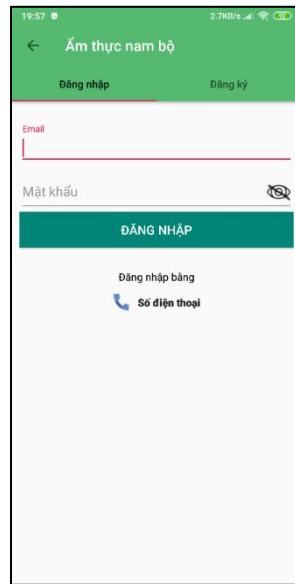
Địa chỉ: QL1A, Tân Ngãi, Vĩnh Long

Giá: 150,000 VNĐ

Hình 4. 14 Tìm bài viết

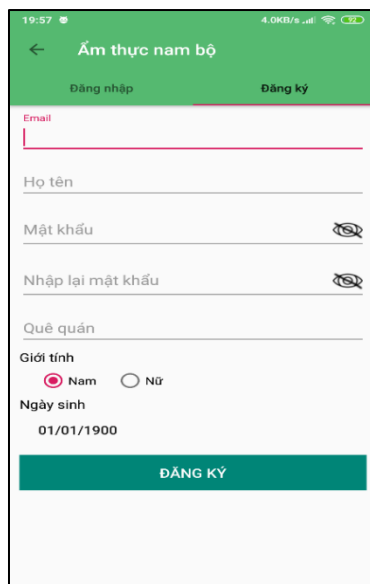
Người dùng có thể tìm bài viết theo tên, khu vực, loại, thành phần và giá tiền tối đa

- **Giao diện đăng nhập, đăng ký:**



Người dùng đăng nhập hệ thống bằng tài khoản đã đăng ký ở phần mềm để sử dụng một số chức năng như lưu bài viết, thêm bài viết....

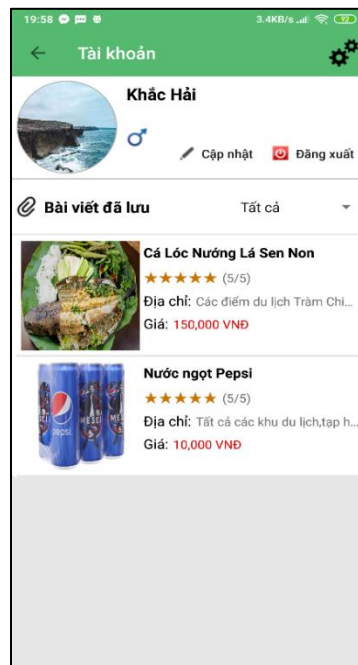
Hình 4. 15 Đăng nhập bằng tài khoản



Nếu người dùng chưa có tài khoản thì có thể đăng ký tài khoản để sử dụng các chức năng của phần mềm

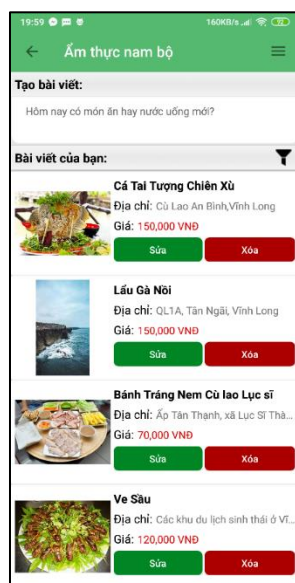
Hình 4. 16 Đăng ký tài khoản

- *Giao diện người dùng:*

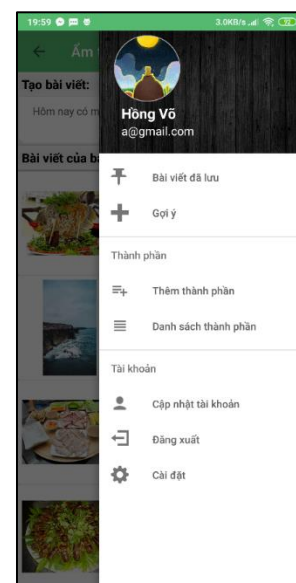


Hình 4. 17 Trang cá nhân người dùng

- *Giao diện người quản lý:*



Hình 4. 18 Trang cá nhân người quản lý



Hình 4. 19 Menu chức năng người quản lý

- ***Giao diện thêm bài viết mới:***

19:59 2.7KB/s 92%

← Bài viết mới ĐĂNG

Loại bài viết:
☒ Thức ăn ☐ Nước uống

Tên bài viết:
 Nhập tên bài viết

Địa chỉ:
 Nhập địa chỉ

Giá:
 Nhập giá

Ghi chú:
 Nhập ghi chú

Khu vực
 Chọn khu vực

Thành phần:
 Chọn thành phần

Hình ảnh:
 Chọn hình ảnh

Giới thiệu
 Nhập giới thiệu

Hình 4. 20 Chức năng thêm bài viết mới

- ***Giao diện cập nhật bài viết:***

20:02 3.9KB/s 93%

← Cập nhật bài viết LƯU

Tên bài viết:
 Cá Tai Tượng Chiên Xù

Địa chỉ:
 Cù Lao An Bình, Vĩnh Long

Giá:
 150000

Ghi chú:
 Dao Động từ 80000_150000VN

Thành phần:
 Cá

Hình ảnh:
 Chọn hình ảnh

Giới thiệu
 Thịt cá vẫn giòn giòn từ lúc chế biến cho tới khi nguội. Vị ngon ngọt tươi nguyên, thịt cá mềm mềm khi dùng với nước mắm chua chua cay cay, ăn kèm với các loại rau cải, cuốn bánh tráng sẽ khiến thực khách khó lòng quên được.

Hình 4. 21 Chức năng cập nhật bài viết

CHƯƠNG 5: KẾT LUẬN VÀ KIẾN NGHỊ

5.1 Kết luận

Phần mềm hoàn thành tất cả các yêu cầu cơ bản đặt ra và chạy được trên hầu hết các điện thoại sử dụng hệ điều hành android từ android 4.1 trở lên.

Kết quả đạt được và chưa đạt được:

Đã đạt	Chưa đạt
<ul style="list-style-type: none">• Đáp ứng được các chức năng, yêu cầu cơ bản như: Giới thiệu, tìm kiếm được các món ăn, nước uống của từng khu vực.• Thêm các chức năng cần thiết cho phần mềm như đánh giá về món ăn, nước uống.• Đăng nhập bằng số điện thoại• Quản trị bài viết dành cho người quản lý• Lưu bài viết vào tài khoản người dùng	<ul style="list-style-type: none">• Chưa đăng được hình có dung lượng cao.• Chưa chỉ đường chi tiết đến địa điểm cụ thể

Bảng 5.1 Bảng kết quả đạt được và chưa đạt được

5.2 Kiến nghị

Định hướng phát triển và cải tiến:

- Mở rộng môi trường làm việc của phần mềm trên nền web và hệ điều hành IOS.
- Thêm một số module như tìm nhà nghỉ, khách sạn, chỉ đường đến từng địa điểm một cách chi tiết và cụ thể

TÀI LIỆU THAM KHẢO

- [1] Document firebase (<https://firebase.google.com/docs>)
- [2] Document android (<https://developer.android.com/reference/org/w3c/dom/Document>)
- [3] Questions android (<https://stackoverflow.com/questions/tagged/android>)
- [4] Book android (<https://androidcoban.com/tai-lieu-android-co-ban-nang-cao.html>)
- [5] Document android (https://www.tutorialspoint.com/android/android_activities.htm)
- [6] Document android (<https://developer.android.com/reference>)
- [7] Document android (<https://www.devpro.edu.vn/tai-lieu-hoc-android>)
- [8] Android Cookbook (<https://aptech-danang.edu.vn/ace/tai-nguyen/sach-lap-trinh/chi-tiet/13999/android-cookbook-2nd-edition>)