

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
LỚP CỬ NHÂN TÀI NĂNG**

NGUYỄN VĂN LÊ TÂY 0912416

NGUYỄN XUÂN THÀNH 0912426

**XÂY DỰNG HỆ THỐNG THỰC TẠI ẢO
TĂNG CƯỜNG TRONG TÌM KIẾM DỊCH VỤ
TRÊN ĐIỆN THOẠI ANDROID**

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

GIÁO VIÊN HƯỚNG DẪN

TS. Đinh Bá Tiến

NIÊN KHÓA 2009 – 2013

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

Giáo viên hướng dẫn
[Ký tên và ghi rõ họ tên]

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.

TpHCM, ngày tháng năm

[Ký tên và ghi rõ họ tên]

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn khoa Công nghệ Thông tin, trường Đại học Khoa học Tự nhiên, Đại học Quốc gia Tp. Hồ Chí Minh đã tạo điều kiện thuận lợi cho chúng em học tập và thực hiện khóa luận tốt nghiệp này.

Chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Đinh Bá Tiến đã tận tình hướng dẫn, chỉ bảo chúng em trong quá trình thực hiện đề tài.

Chúng em xin chân thành cảm ơn quý Thầy/Cô trong Khoa Công nghệ Thông tin đã tận tình giảng dạy, trang bị cho chúng em những kiến thức quý báu trong những năm học vừa qua.

Bên cạnh sự chỉ dạy của Thầy/Cô đó là sự ủng hộ và động viên rất lớn từ phía ba mẹ và gia đình cũng như sự quan tâm, giúp đỡ của bạn bè trong suốt quá trình thực hiện khóa luận.

Mặc dù đã cố gắng hoàn thành khóa luận trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm, góp ý và tận tình chỉ bảo của quý Thầy Cô và các bạn.

Tp. Hồ Chí Minh, tháng 07 năm 2013

Nhóm sinh viên thực hiện

Nguyễn Văn Lê Tây & Nguyễn Xuân Thành

Khoa Công Nghệ Thông Tin

Lớp Cử Nhân Tài Năng

ĐỀ CƯƠNG CHI TIẾT

| | | |
|----------------------------------------------------------------------------------------------------------------------|-------------------|---------|
| Tên Đề Tài: XÂY DỰNG HỆ THỐNG THỰC TẠI ẢO TĂNG CƯỜNG TRONG TÌM KIẾM DỊCH VỤ TRÊN ĐIỆN THOẠI ANDROID | | |
| Giáo viên hướng dẫn: TS. Đinh Bá Tiến | | |
| Thời gian thực hiện: 15/08/2012 – 30/06/2013 | | |
| Sinh viên thực hiện: | Nguyễn Văn Lê Tây | 0912416 |
| | Nguyễn Xuân Thành | 0912426 |
| Loại đề tài: Tìm hiểu công nghệ thực tại ảo tăng cường và xây dựng ứng dụng minh họa trên điện thoại Android. | | |

Nội Dung Đề Tài:

- Khảo sát nền tảng smartphone phổ biến.
- Nghiên cứu công nghệ thực tại ảo tăng cường, định hướng phát triển.
- Phân tích và xử lý dữ liệu từ cảm biến gia tốc và cảm biến trọng trường, áp dụng vào xác định phương hướng của thiết bị.
- Xác định, phân tích các vấn đề và đưa ra các giải pháp trong lập trình Android.
- Xây dựng ứng dụng tìm kiếm dịch vụ minh họa áp dụng công nghệ thực tại ảo tăng cường.

Kế Hoạch Thực Hiện:

- 15/08/2012 – 01/09/2012: Thảo luận và chọn ý tưởng.
- 02/09/2012 – 14/09/2012: Khảo sát và chọn nền tảng smartphone để phát triển ứng dụng.
- 15/09/2012 – 01/10/2012: Tìm hiểu cách truy xuất và xử lý dữ liệu từ camera của smartphone.
- 02/10/2012 – 14/10/2012: Sửa lỗi camera khi chuyển khung nhìn. Hoàn thiện intent và chuyển đổi activity trong Android.
- 15/10/2012 – 01/11/2012: Nghiên cứu công nghệ thực tại ảo tăng cường.
- 02/11/2012 – 14/11/2012: Nghiên cứu sử dụng Google map view.
- 15/11/2012 – 01/12/2012: Xác định thông tin vị trí của thiết bị bằng GPS và Network kết hợp Cell-Id. Hiển thị các dịch vụ dạng danh sách hoặc biểu tượng theo hướng nhìn của người dùng.
- 02/12/2012 – 14/12/2012: Tạo và truy vấn cơ sở dữ liệu SQLite. Xử lý dữ liệu từ cảm biến, từ đó xác định phương hướng người dùng.
- 15/12/2013 – 14/02/2013: Thi học kỳ và nghỉ Tết Nguyên Đán.
- 15/02/2013 – 30/03/2013: Tiếp tục cài đặt Google map view, Augmented Reality view. Xây dựng biểu tượng tùy biến trên bản đồ.
- 01/03/2013 – 14/03/2013: Tìm đường đi trên bản đồ.
- 15/03/2013 – 30/03/2013: Tìm hiểu và xây dựng Webservice API.
- 01/04/2013 – 14/04/2013: Client truy vấn dữ liệu từ server thông qua Restful request.
- 15/04/2013 – 30/04/2013: Xây dựng chức năng tìm kiếm có hỗ trợ tiếng Việt tại client và server.
- 01/05/2013 – 15/05/2013: Lưu trữ dữ liệu tải về từ server.

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| <ul style="list-style-type: none"> • 15/05/2013 – 30/05/2013: Tìm hiểu push notification để gửi tin nhắn miễn phí trên Android. • 01/06/2013 – 14/06/2013: Hoàn thiện chức năng và giao diện ứng dụng. • 15/06/2013 – 30/06/2013: Viết báo cáo. | |
| Xác nhận của GVHD | Ngày.....tháng.....năm..... SV Thực hiện |



MỤC LỤC

| | |
|---------------------------------------------------------|------|
| ĐỀ CƯƠNG CHI TIẾT | i |
| MỤC LỤC | iv |
| MỤC LỤC HÌNH | x |
| MỤC LỤC BẢNG | xii |
| TÓM TẮT KHÓA LUẬN | xiii |
| MỞ ĐẦU | xiv |
| TỔNG QUAN | xvi |
| Chương 1: KHẢO SÁT HIỆN TRẠNG VÀ XÂY DỰNG Ý TƯỞNG | 1 |
| 1.1 Khảo sát hiện trạng | 1 |
| 1.1.1 Đặc điểm của ứng dụng trên smartphone | 1 |
| 1.1.2 Các ứng dụng tìm kiếm dịch vụ hiện có | 2 |
| 1.1.2.1 Gnavi Việt Nam | 2 |
| 1.1.2.2 Đi Đâu - Tìm địa điểm | 4 |
| 1.1.2.3 iFind Places | 6 |
| 1.1.3 Khái niệm Augmented Reality | 8 |
| 1.1.4 Hạn chế và những vấn đề còn tồn tại | 8 |
| 1.2 Phát biểu bài toán | 9 |
| 1.2.1 Ý tưởng | 9 |
| 1.2.2 Các tính năng chính của ứng dụng | 9 |
| 1.2.2.1 Tìm kiếm dịch vụ theo hướng | 9 |

| | | |
|--------------------------------------------------------------|-------------------------------------------------------------------------------------------|----|
| 1.2.2.2 | Tìm kiếm theo từ khóa | 9 |
| 1.2.2.3 | Theo dõi hướng và khoảng cách đến dịch vụ cụ thể..... | 9 |
| 1.2.2.4 | Hiển thị trực quan bản đồ google map | 10 |
| 1.2.2.5 | Kết bạn, Chat bằng tài khoản Gmail | 10 |
| Chương 2: CÁC VẤN ĐỀ CHÍNH TRONG QUÁ TRÌNH XÂY DỰNG ỨNG DỤNG | | 11 |
| 2.1 | Xác định phương hướng trên smartphone..... | 11 |
| 2.1.1 | Cảm biến gia tốc và cảm biến từ trường | 11 |
| 2.1.1.1 | Giới thiệu về cảm biến | 11 |
| 2.1.1.2 | Các góc quay Pitch – Roll – Yaw | 13 |
| 2.1.1.3 | Ma trận quay | 14 |
| 2.1.1.4 | Xác định giá trị của các góc quay Pitch – Roll – Yaw dựa vào ma trận quay | 15 |
| 2.1.1.5 | Hiện tượng Gimbal lock..... | 15 |
| 2.1.2 | Xác định phương hướng bằng cách sử dụng cảm biến gia tốc kết hợp cảm biến từ trường | 16 |
| 2.1.3 | Bộ lọc Kalman | 19 |
| 2.1.3.1 | Lý do sử dụng bộ lọc | 19 |
| 2.1.3.2 | Giới thiệu bộ lọc Kalman | 20 |
| 2.1.3.3 | Áp dụng bộ lọc Kalman vào lọc nhiễu tín hiệu cảm biến | 23 |
| 2.1.3.4 | Đánh giá kết quả | 24 |
| 2.2 | Phân tích và thiết kế ứng dụng minh họa..... | 27 |

| | | |
|-----------|---------------------------------------------|----|
| 2.2.1 | Mục tiêu | 27 |
| 2.2.2 | Yêu cầu ứng dụng | 27 |
| 2.2.3 | Mô hình Use case | 28 |
| 2.2.3.1 | Sơ đồ Use case | 28 |
| 2.2.3.2 | Đặc tả Use case | 29 |
| 2.2.3.2.1 | Lọc dịch vụ | 29 |
| 2.2.3.2.2 | Tìm kiếm dịch vụ | 30 |
| 2.2.3.2.3 | Thêm dữ liệu | 31 |
| 2.2.3.2.4 | Tải dữ liệu | 32 |
| 2.2.3.2.5 | Xem bản đồ | 33 |
| 2.2.3.2.6 | Chat | 34 |
| 2.2.3.2.7 | Sắp xếp dịch vụ | 35 |
| 2.2.3.2.8 | Cài đặt hệ thống | 36 |
| 2.2.4 | Thiết kế kiến trúc | 38 |
| 2.2.4.1 | Client-Server | 38 |
| 2.2.4.2 | Một số kiến trúc khác | 39 |
| 2.2.5 | Thiết kế cơ sở dữ liệu | 40 |
| 2.2.5.1 | Ngôn ngữ, nền tảng | 41 |
| 2.2.5.2 | Cấu trúc cơ sở dữ liệu | 41 |
| 2.2.6 | Giao diện | 42 |
| 2.3 | Cài đặt ứng dụng | 44 |
| 2.3.1 | Lấy hình ảnh từ camera của điện thoại | 44 |

| | | |
|---------|-----------------------------------------------------------|----|
| 2.3.2 | Tùy biến ListView trong Android | 47 |
| 2.3.3 | Xác định danh sách dịch vụ thuộc khung nhìn camera | 51 |
| 2.3.4 | Hiển thị Google Maps API v1 và tương tác biểu tượng | 56 |
| 2.3.4.1 | Hiển thị Google Maps API v1 | 56 |
| 2.3.4.2 | Hiển thị biểu tượng trên Google Maps..... | 57 |
| 2.3.5 | Xây dựng webserver API | 59 |
| 2.3.5.1 | Vai trò của server trong ứng dụng | 59 |
| 2.3.5.2 | Ngôn ngữ, nền tảng | 60 |
| 2.3.5.3 | Cài đặt Web Services | 60 |
| 2.3.6 | Truy vấn Restful Service trên Android..... | 62 |
| 2.3.6.1 | Thực thi một request | 62 |
| 2.3.6.2 | Truyền dữ liệu trong request..... | 63 |
| 2.3.7 | Sử dụng Google Talk service để push notification | 64 |
| 2.3.7.1 | Kết nối đến server Google Talk..... | 64 |
| 2.3.7.2 | Lưu trữ thông tin người dùng | 65 |
| 2.3.8 | Một số vấn đề khác | 65 |
| 2.3.8.1 | Hiển thị chỉ dẫn đường đi | 65 |
| 2.3.8.2 | Thực hiện cuộc gọi điện thoại..... | 66 |
| 2.3.8.3 | Tự động chuyển đổi màn hình | 66 |
| 2.3.8.4 | Nạp thêm dữ liệu vào ListView khi cuộn danh sách | 67 |
| 2.3.8.5 | Tìm kiếm dịch vụ với từ khóa tiếng Việt | 68 |

| | | |
|-----------------------------------|----------------------------------------------------------|----|
| Chương 3: | ỨNG DỤNG MINH HỌA..... | 71 |
| 3.1 | Tìm kiếm dịch vụ theo hướng | 71 |
| 3.2 | Lọc danh sách theo khoảng cách và theo loại dịch vụ..... | 72 |
| 3.3 | Tìm kiếm dịch vụ theo từ khóa..... | 73 |
| 3.4 | Theo dõi hướng và khoảng cách đến dịch vụ cụ thể | 75 |
| 3.5 | Xem thông tin chi tiết của dịch vụ..... | 75 |
| 3.6 | Hiển thị trực quan bản đồ Google Maps | 76 |
| 3.7 | Tìm đường đi | 78 |
| 3.8 | Thay đổi vị trí người dùng | 79 |
| 3.9 | Xem vị trí bạn bè trên bản đồ | 80 |
| 3.10 | Kết bạn, Chat bằng tài khoản Gmail..... | 81 |
| 3.11 | Thêm dịch vụ mới..... | 85 |
| 3.12 | Tải dữ liệu..... | 86 |
| 3.13 | Cài đặt hệ thống | 86 |
| 3.14 | Thêm dịch vụ mới thông qua Web Page | 88 |
| Chương 4: | KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN..... | 89 |
| 4.1 | Kết luận | 89 |
| 4.1.1 | Một số kết quả đạt được..... | 89 |
| 4.1.2 | Hạn chế..... | 90 |
| 4.2 | Hướng phát triển | 91 |
| DANH MỤC TÀI LIỆU THAM KHẢO | | 92 |

| | |
|-------------------------------------------|-----|
| PHỤ LỤC..... | 93 |
| Phụ lục A: Giới thiệu về Android..... | 93 |
| 1. Lịch sử..... | 93 |
| 2. Phát triển..... | 94 |
| 3. Kiến trúc và thiết kế..... | 95 |
| 3.1. Tầng Applications | 96 |
| 3.2. Tầng Application Framework | 97 |
| 3.3. Library | 97 |
| 3.4. Android Runtime..... | 98 |
| 3.5. Linux Kernel | 99 |
| 4. Android SDK..... | 99 |
| 5. Môi trường lập trình cho Android | 100 |

MỤC LỤC HÌNH

| | |
|-------------------------------------------------------------------------------------------------------|----|
| Hình 1-1 Màn hình ứng dụng <i>Gnavi</i> | 3 |
| Hình 1-2 Màn hình ứng dụng <i>Đi Đâu</i> | 5 |
| Hình 1-3 Màn hình ứng dụng <i>iFind Places</i> | 7 |
| Hình 2-1 Ba giá trị góc quay Pitch – Roll - Yaw | 13 |
| Hình 2-2 Mô hình hóa hoạt động của mạch lọc Kalman | 21 |
| Hình 2-3 Mô hình xử lý của bộ lọc Kalman | 22 |
| Hình 2-4 Mô hình xử lý của bộ lọc Kalman cho tín hiệu cảm biến..... | 23 |
| Hình 2-5 Xử lý tín hiệu Pitch bằng bộ lọc Kalman | 24 |
| Hình 2-6 Xử lý tín hiệu Roll bằng bộ lọc Kalman..... | 25 |
| Hình 2-7 Xử lý tín hiệu Yaw bằng bộ lọc Kalman | 25 |
| Hình 2-8 Sơ đồ Use case | 28 |
| Hình 2-9 Kiến trúc client – server | 38 |
| Hình 2-10 Áp dụng mẫu Strategy trong sắp xếp danh sách dịch vụ..... | 40 |
| Hình 2-11 Sơ đồ thực thể của cơ sở dữ liệu dịch vụ | 41 |
| Hình 2-12 Sơ đồ thực thể của cơ sở dữ liệu Chat | 42 |
| Hình 2-13 Biểu tượng của ứng dụng <i>Search Master</i> | 42 |
| Hình 2-14 Giao diện cho các mục trong ListView | 48 |
| Hình 2-15 Hình ảnh mô phỏng vị trí của thiết bị và vị trí của dịch vụ trong hệ tọa độ Descartes. | 52 |
| Hình 2-16 Giá trị α vượt quá miền giá trị $(-\pi; \pi]$ | 54 |
| Hình 2-17 Tạo Web Services trên Visual Studio 2010 | 60 |
| Hình 2-18 Mẫu Web API theo mô hình MVC | 61 |
| Hình 3-1 Màn hình chính của ứng dụng..... | 71 |
| Hình 3-2 Lọc danh sách theo khoảng cách và theo loại dịch vụ | 72 |
| Hình 3-3 Màn hình tìm kiếm dịch vụ theo từ khóa..... | 73 |

| | |
|-----------------------------------------------------------------------|----|
| Hình 3-4 Màn hình tìm kiếm dịch vụ theo từ khóa tại server | 74 |
| Hình 3-5 Màn hình theo dõi hướng đến dịch vụ cụ thể | 75 |
| Hình 3-6 Màn hình xem thông tin chi tiết dịch vụ..... | 76 |
| Hình 3-7 Màn hình bản đồ..... | 77 |
| Hình 3-8 Tìm đường đi..... | 78 |
| Hình 3-9 Thông tin chỉ dẫn đường đi..... | 79 |
| Hình 3-10 Thay đổi vị trí của người dùng..... | 80 |
| Hình 3-11 Vị trí bạn bè trên bản đồ | 81 |
| Hình 3-12 Màn hình đăng nhập chức năng Chat bằng tài khoản Gmail..... | 82 |
| Hình 3-13 Màn hình Chat với bạn bè..... | 83 |
| Hình 3-14 Màn hình kết bạn và trả lời kết bạn | 84 |
| Hình 3-15 Menu của màn hình chat..... | 84 |
| Hình 3-16 Màn hình thêm dịch vụ mới | 85 |
| Hình 3-17 Màn hình chờ tải dữ liệu..... | 86 |
| Hình 3-18 Màn hình cài đặt hệ thống..... | 88 |
| Hình 3-19 Thêm dịch vụ mới thông qua Web Page..... | 88 |
| Hình 4-1 Thị phần hệ điều hành Smartphone trong quý I năm 2013 | 95 |
| Hình 4-2 Sơ đồ kiến trúc hệ điều hành Android..... | 96 |

MỤC LỤC BẢNG

| | |
|-----------------------------------------------------------------------|----|
| Bảng 2-1 Các giá trị thu được từ cảm biến trên thiết bị Android | 16 |
| Bảng 2-2 Danh sách các Use case..... | 29 |
| Bảng 2-3 Danh sách các màn hình của ứng dụng | 43 |
| Bảng 4-1 Thị phần các phiên bản của hệ điều hành Android..... | 94 |

TÓM TẮT KHÓA LUẬN

Trong thời gian gần đây, công nghệ thực tại ảo tăng cường (Augmented Reality) bắt đầu thu hút sự chú ý của giới công nghệ cùng với sự phát triển mạnh mẽ của thị trường smartphone. Với công nghệ thực tại ảo tăng cường, người dùng sẽ có trải nghiệm mới trong việc tiếp nhận thông tin. Người dùng có thể đưa điện thoại về một hướng nào đó, ngay lập tức thiết bị phân tích dữ liệu, đối chiếu phương hướng, hình ảnh để cung cấp thông tin về địa điểm, dịch vụ... Tuy nhiên, các ứng dụng được phát triển thời gian qua mới chỉ dừng ở mức thử nghiệm, giới thiệu công nghệ, cho thấy công nghệ thực tại ảo tăng cường có thể sẽ làm được những gì. Do vậy, chúng em thực hiện khóa luận là bước mở đầu cho việc nghiên cứu và áp dụng công nghệ thực tại ảo tăng cường.

Bên cạnh đó, hệ điều hành Android ngày càng chiếm nhiều thị phần trong thị trường smartphone. Chính vì vậy, chúng em chọn nền tảng Android để xây dựng ứng dụng minh họa cho công nghệ thực tại ảo tăng cường.

Chúng em phân tích dữ liệu từ gia tốc kế và cảm biến gia tốc của điện thoại Android. Từ đó, cho phép xác định vị trí, phương hướng của người dùng so với vị trí của các dịch vụ và hiển thị thông tin của các dịch vụ trong phạm vi quan sát của người dùng.

Sau khi hoàn thành khóa luận, chúng em đã hoàn thành xây dựng ứng dụng minh họa công nghệ thực tại ảo tăng cường, cũng như tìm hiểu về các kỹ thuật lập trình trên Android.

MỞ ĐẦU

❖ Bối cảnh và nhu cầu thực tế

Chúng ta đang sống ở đầu thế kỷ 21, thời kỳ khoa học kỹ thuật đạt được nhiều thành tựu và được áp dụng mạnh mẽ vào các lĩnh vực của cuộc sống. Các thiết bị điện tử ngày càng có vai trò quan trọng hơn trong cuộc sống của con người, trở thành vật không thể thiếu để giúp con người có cuộc sống tiện nghi hơn. Điện thoại di động là minh chứng rõ ràng nhất cho điều đó. Ban đầu, điện thoại di động ra đời với mục đích chủ yếu là vì nhu cầu liên lạc cơ bản như gọi điện, nhắn tin. Nhưng cùng với sự phát triển nhanh chóng của khoa học kỹ thuật, rất nhiều các chức năng khác cũng được tích hợp vào trong điện thoại di động như: chụp ảnh, quay phim, lướt web, chơi trò chơi, nghe radio, xem tivi, quản lý chi tiêu, soạn thảo văn bản, ... Chính vì quá đa dụng như vậy cho nên cái tên điện thoại di động không còn phù hợp với một số dòng điện thoại hiện đại gần đây nữa, mà chúng được gọi là SmartPhone. Cùng những tiện ích to lớn mang lại và cấu hình ngày càng được cải thiện, smartphone đang dần chiếm lĩnh thị trường di động thế giới.

Bên cạnh sự phát triển mạnh mẽ của smartphone, các hệ điều hành trên smartphone cũng phát triển tương ứng, nổi bật nhất trong số đó là 2 hệ điều hành: Android và iOS. Tính tại thị trường Bắc Mỹ, thì hệ điều hành Android chiếm 75% smartphone, iOS là 14,9% (số liệu năm 2012), xét trên quy mô toàn cầu thì hệ điều hành Android được cài đặt trên 70,1% số điện thoại smartphone được bán ra toàn cầu, trong khi đó iOS chiếm chỉ 21%.

Song song với sự tăng trưởng mạnh mẽ của smartphone, các ứng dụng trên di động cũng phát triển nhanh chóng để đáp ứng nhu cầu của đông đảo người sử dụng. Tính đến tháng 4/2013, App Store và Play Store (2 nhà phân phối phần mềm di động lớn nhất thế giới), mỗi bên đã có khoảng 800.000 ứng dụng để cung cấp cho

hơn 1,1 tỉ người sử dụng smartphone. Chỉ tính Play Store, trong năm 2012, đã có khoảng 48 tỉ lượt download và cài đặt ứng dụng từ Play Store. Những con số đáng kể ở trên cho ta thấy tiềm năng cũng như khả năng ứng dụng mạnh mẽ của smartphone qua các ứng dụng di động.

Xu hướng hiện nay là xây dựng các ứng dụng hữu ích, tiện dụng và tương tác tốt với người sử dụng. Nhóm muốn phát triển ứng dụng tìm kiếm dịch vụ sử dụng camera và một số cảm biến của smartphone nhằm mang lại sự trải nghiệm mới mẻ hơn với người sử dụng.

❖ **Mục tiêu khóa luận**

Áp dụng công nghệ AR (Augmented Reality) vào ứng dụng trên smartphone. Cụ thể, tìm hiểu một số cảm biến, thiết bị được tích hợp trong smartphone, nghiên cứu áp dụng các tính năng của cảm biến đó trong ứng dụng thực tế nhằm tăng khả năng tương tác với người dùng. Ứng dụng được chọn để phát triển trong khóa luận này là ứng dụng tìm kiếm dịch vụ.

TỔNG QUAN

❖ Nội dung khóa luận

Nội dung của khóa luận được trình bày trong 4 chương của tài liệu này. Cụ thể các phần có thể khái quát như sau:

Chương 1: Trình bày kết quả khảo sát, đánh giá thực tế và nêu lên ý tưởng của nhóm.

Chương 2: Nêu lên những vấn đề gặp phải trong quá trình hiện thực hóa ý tưởng: cơ sở lý thuyết cần thiết, phân tích thiết kế ứng dụng và những kỹ thuật được sử dụng để xây dựng ứng dụng.

Chương 3: Kết quả của quá trình nghiên cứu và cài đặt ứng dụng. Nêu lên các tính năng chi tiết và giao diện chính của ứng dụng.

Chương 4: Kết quả và định hướng, triển vọng phát triển trong tương lai của đề tài nghiên cứu.

Phần cuối: Phần phụ lục.

Chương 1: KHẢO SÁT HIỆN TRẠNG VÀ XÂY DỰNG Ý TƯỞNG

1.1 Khảo sát hiện trạng

1.1.1 Đặc điểm của ứng dụng trên smartphone

Do đặc thù về phần cứng, các ứng dụng di động nói chung và ứng dụng smartphone nói riêng có nhiều đặc điểm khác biệt so với những ứng dụng trên PC và Web. Cụ thể:

- Đơn giản: Xu hướng của người dùng di động là không thích những phần mềm phức tạp, khó thao tác, bởi vì trên smartphone không có bàn phím rời và con chuột với chuột trái, chuột phải như ở PC và Laptop, điều này làm cho việc thao tác trên thiết bị di động không thích hợp cho những phần mềm lớn, có nhiều luồng xử lý rắc rối như ứng dụng trên PC. Người dùng thích sử dụng những tiện ích, nhỏ, hiệu quả, có thể đáp ứng tốt công việc, hay yêu cầu nào đó, tránh tích hợp quá nhiều, gây rắc rối, nhầm lẫn và nặng nề cho máy. Giao diện cũng yêu cầu tương tự, đơn giản và dễ sử dụng, bởi vì diện tích màn hình của thiết bị di động hạn chế, không đủ không gian để đưa nhiều nội dung vào. Một giao diện nhỏ gọn, dễ hiểu, dễ sử dụng sẽ là sự lựa chọn tốt nhất cho người sử dụng.
- Nhỏ: Vì là thiết bị di động nên không thể trang bị được các bộ vi xử lý, thiết bị lưu trữ mạnh mẽ như máy vi tính, tuy gần đây smartphone có những cải tiến đáng kể, chip vi xử lý cũng được cải thiện nhiều nhưng vẫn còn khoảng cách rất lớn so với PC và Laptop.
- Di động: Vì là phần mềm di động, không thể không để ý đến việc xử lý các sự kiện liên quan như: có điện thoại gọi đến, thông báo hết pin, có tin nhắn, ...

Các nhà phát triển phần mềm di động phải xử lý các trường hợp này để tạo nên phần mềm chất lượng, ổn định.

➤ Tương tác: Mạng lưới người sử dụng smartphone hiện nay rất đông đảo và mạnh mẽ, smartphone thường được kết nối với mạng xã hội để đáp ứng nhu cầu chia sẻ thông tin, tương tác giữa con người và con người trên điện thoại. Những ứng dụng đơn giản nhưng có đông đảo người sử dụng một phần là vì khả năng tương tác rất cao của nó, Instagram là một ví dụ điển hình.

1.1.2 Các ứng dụng tìm kiếm dịch vụ hiện có

1.1.2.1 Gnavi Việt Nam

Gnavi là hệ thống tìm kiếm và định vị nhà hàng quán ăn tại Việt Nam. *Gnavi* cung cấp cho người dùng những thông tin liên quan tới nhà hàng, món ăn mà người dùng muốn tìm kiếm.

Kích thước: 619KB

Đường dẫn: <https://play.google.com/store/apps/details?id=hiworld.gnavi>

Giao diện chính:



Hình 1-1 Màn hình ứng dụng *Gnavi*

Chức năng tìm kiếm: tìm kiếm nhà hàng lân cận dựa theo thông tin vị trí GPS. Ngoài ra còn có rất nhiều các chức năng tìm kiếm khác như tìm kiếm theo loại, vị trí địa lý, theo tuyến xe.

Hiển thị thông tin chi tiết của nhà hàng gồm: tên, địa chỉ, điện thoại, giờ phục vụ, thực đơn, hình ảnh, bản đồ... Ngoài ra, việc thông báo những thông tin dịch vụ đặc biệt (phiếu giảm giá, đợt khuyến mãi...) hay những thực đơn đặc biệt sẽ được truyền tải đến khách hàng một cách nhanh chóng, đầy đủ nhất.

Xếp hạng truy cập: số lượt truy cập vào shop sẽ được xếp hạng và hiển thị trong từng tuần, theo đó người dùng sẽ biết được nhà hàng nào có số lượng truy cập nhiều nhất trong tuần. Nhà hàng nổi tiếng là nhà hàng được hiển thị trong danh sách bảng xếp hạng của trang.

Nhận xét: thể hiện đầy đủ thông tin của nhà hàng. Xem được vị trí nhà hàng trên bản đồ. Có thông tin khuyến mãi và xếp hạng nhà hàng theo lượng truy cập. Ứng dụng chỉ tìm kiếm được loại dịch vụ nhà hàng. Khi không có kết nối mạng thì ứng dụng không tìm kiếm được. Ứng dụng bị lỗi khi xoay màn hình trong lúc đang tìm kiếm. Dù đang ở màn hình nào, khi xoay màn hình ứng dụng đều khởi tạo lại tài nguyên, chạy lại vào màn hình đầu tiên. Xem được vị trí nhà hàng trên bản đồ nhưng không hiển thị vị trí của người dùng.

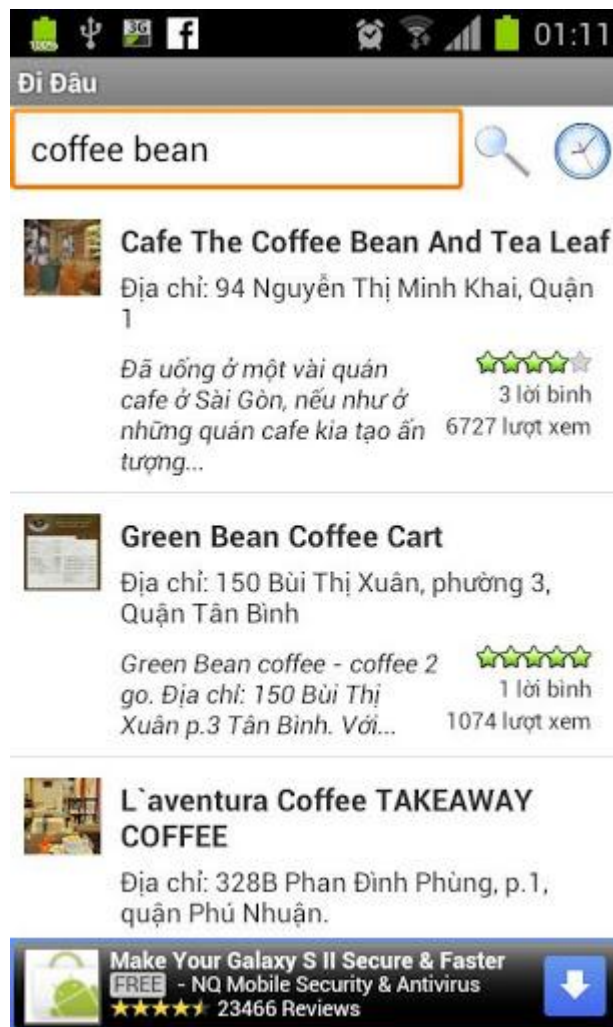
1.1.2.2 Đi Đâu - Tìm địa điểm

Đi Đâu là ứng dụng miễn phí giúp bạn tìm thông tin về các địa điểm, nhà hàng, quán ăn phổ biến và xem đánh giá, bình luận về các địa điểm này.

Kích thước: 595KB

Đường dẫn: <https://play.google.com/store/apps/details?id=com.vm.didau.free>

Giao diện chính:



Hình 1-2 Màn hình ứng dụng *Đi Đâu*

Đi Đâu khai thác thông tin từ trang web *Thổ Địa* (thodia.vn). Mọi thông tin đều do trang *Thổ Địa* giữ bản quyền. *Đi Đâu* không có liên hệ nào với nhà phát triển của trang *Thổ Địa*.

Chức năng chính:

- Tìm kiếm địa điểm từ *Thổ Địa*.
- Hiển thị thông tin chi tiết về mỗi địa điểm như địa chỉ, điện thoại, website, lời bình luận của các thành viên *Thổ Địa*.

- Hỗ trợ cache các địa điểm đã xem để có thể xem lại offline khi cần.

Nhận xét: dữ liệu phong phú từ trang *Thổ Địa*. Tải dữ liệu chi tiết về chậm. Chỉ cache các địa điểm đã chọn xem thông tin chi tiết. Không tìm kiếm được trên các địa điểm đã cache. Không có màn hình bản đồ để xem vị trí dịch vụ. Ứng dụng đơn giản chỉ tìm dịch vụ theo từ khóa và xem thông tin chi tiết. Màn hình ứng dụng bị chèn quảng cáo.

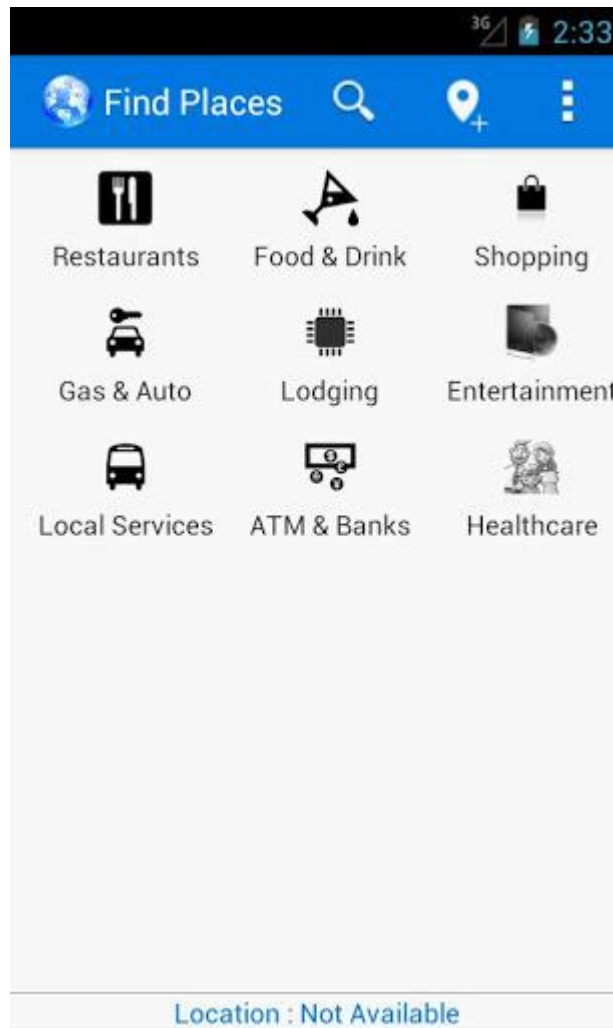
1.1.2.3 iFind Places

iFind Places giúp người dùng nhanh chóng tìm kiếm nhà hàng, thực phẩm và đồ uống, trung tâm thương mại, trạm xăng, nhà nghỉ, giải trí, dịch vụ địa phương, chăm sóc sức khỏe, ATM và ngân hàng.

Kích thước: 530KB

Đường dẫn: <https://play.google.com/store/apps/details?id=com.dikkar.ifind>

Giao diện chính:



Hình 1-3 Màn hình ứng dụng *iFind Places*

Tính năng:

- Hiển thị danh sách dịch vụ theo tùy chọn.
- Xem đánh giá cho mỗi địa điểm.
- Tìm đường đi.
- Thực hiện cuộc gọi đến địa điểm.
- Chia sẻ địa điểm qua Twitter, Facebook, Text, Email...

Nhận xét: dữ liệu phong phú từ google. Có phiên bản Free và Pro. Màn hình ứng dụng bị chèn quảng cáo. Thời gian xác định dịch vụ trên bản đồ chậm.

1.1.3 Khái niệm Augmented Reality

Khái niệm Augmented Reality (AR) có thể được dịch là “thực tại ảo tăng cường” hay “công nghệ tương tác thời gian thực”, là công nghệ sử dụng các tín hiệu từ hình ảnh thực tế thông qua camera, định vị GPS, micro,... rồi cung cấp cho người dùng thông tin cụ thể từ những tín hiệu đó, tạo cho người dùng cảm giác như đang tương tác thật sự với những sự vật hiện tượng trước mắt thông qua thiết bị điện tử [1].

Một số ứng dụng có sử dụng AR khá nổi tiếng như *Star Walk*: ứng dụng thiên văn, cho phép ta biết tên, hình dạng các chòm sao; *Blippar*: cung cấp các thông tin bổ sung về sản phẩm, nhãn hiệu, công ty mỗi khi người dùng trở thiết bị vào các mẫu quảng cáo trên báo, tạp chí, poster hay bao bì sản phẩm.

Các ứng dụng được áp dụng công nghệ AR có thể mang đến cho người sử dụng nhiều trải nghiệm mới mẻ, khiến cho người dùng không còn cảm thấy phần mềm hay ứng dụng không chỉ là những con số, hình ảnh khô khan, mà thay vào đó là sự tương tác với những sự vật hiện tượng có thật qua thiết bị trên tay. Từ đó có thể nhận định rằng tiềm năng phát triển của AR là rất lớn.

1.1.4 Hạn chế và những vấn đề còn tồn tại

Các ứng dụng tìm kiếm hiện có chủ yếu cung cấp thông tin dạng ký tự, chưa có phương pháp trực quan giúp người dùng xác định phương hướng, vị trí của dịch vụ. Một số có cung cấp hiển thị vị trí dịch vụ trên bản đồ nhưng không chi tiết, không thể hiện vị trí người dùng, không tìm kiếm hiển thị danh sách dịch vụ trên bản đồ. Một số ứng dụng không thể thực hiện tìm kiếm khi không có kết nối internet. Các ứng dụng bị chèn quảng cáo.

1.2 Phát biểu bài toán

1.2.1 Ý tưởng

Tận dụng thế mạnh về các thiết bị và cảm biến trên smartphone và áp dụng công nghệ thực tại ảo tăng cường để xây dựng ứng dụng tìm kiếm dịch vụ mới, khả năng tương tác cao, thân thiện và hấp dẫn.

1.2.2 Các tính năng chính của ứng dụng

1.2.2.1 Tìm kiếm dịch vụ theo hướng

Cho phép người sử dụng biết được các dịch vụ hiện có bằng cách đưa màn hình camera về hướng đó, danh sách các dịch vụ sẽ được hiển thị trên màn hình camera và sẽ được tự động cập nhật khi hướng xoay của camera bị thay đổi.

Kết quả tìm kiếm có thể được hiển thị dưới dạng danh sách hay dạng biểu tượng. Ở chế độ biểu tượng, vị trí của biểu tượng sẽ tương ứng với khoảng cách từ người dùng đến dịch vụ đó, chỉ có những dịch vụ nào nằm trong phạm vi quan sát thì mới được hiển thị lên màn hình camera.

1.2.2.2 Tìm kiếm theo từ khóa

Người dùng có thể sử dụng chức năng tìm kiếm của ứng dụng để tìm kiếm một dịch vụ nào đó theo tên, nếu không tìm thấy dữ liệu ở trong máy vì cơ sở dữ liệu trong máy chưa được cập nhật, người dùng có thể chọn tìm kiếm trực tiếp trên server nếu như máy được nối mạng.

1.2.2.3 Theo dõi hướng và khoảng cách đến dịch vụ cụ thể

Ứng dụng cung cấp đầy đủ thông tin dịch vụ cho người dùng. Người dùng có thể xem khoảng cách từ vị trí thiết bị đến vị trí dịch vụ, thực hiện cuộc gọi điện thoại đến dịch vụ cụ thể. Đặc biệt, giúp người dùng xác định phương hướng của dịch vụ được chọn một cách trực quan.

1.2.2.4 Hiển thị trực quan bản đồ google map

Phần mềm cho phép hiển thị các dịch vụ dạng biểu tượng trên bản đồ, lọc các dịch vụ theo loại, khoảng cách. Phần mềm cung cấp khả năng tìm đường đi: hiển thị hướng dẫn đường đi, vẽ đoạn đường trên bản đồ.

1.2.2.5 Kết bạn, Chat bằng tài khoản Gmail

Người dùng sử dụng tài khoản Gmail để đăng nhập chức năng Chat. Ứng dụng cho phép gửi lời kết bạn, xác nhận kết bạn, xem danh sách bạn online và gửi tin nhắn trong chức năng Chat.

Chương 2: CÁC VẤN ĐỀ CHÍNH TRONG QUÁ TRÌNH XÂY DỰNG ỨNG DỤNG

2.1 Xác định phương hướng trên smartphone

2.1.1 Cảm biến gia tốc và cảm biến từ trường

2.1.1.1 Giới thiệu về cảm biến

Cảm biến là một thiết bị đo đạc một đại lượng vật lý và chuyển đổi nó thành tín hiệu để người hoặc thiết bị khác có thể đọc, quan sát được. Ví dụ: nhiệt kế thủy ngân chuyển đổi sự thay đổi nhiệt độ thành sự thay đổi độ cao của thủy ngân trong ống thủy tinh để có thể đọc được [2].

Hầu hết các cảm biến đều được canh chỉnh độ chính xác dựa vào các chuẩn đã biết. Ví dụ: nhiệt độ của nước đá là 0°C và nhiệt độ của nước đang sôi là 100°C thì ta sẽ đánh dấu được bảng chia nhiệt độ cho nhiệt kế.

Bản thân cảm biến cũng ảnh hưởng đến kết quả đo đạc. Ví dụ: nếu đặt nhiệt kế thủy ngân vào ly nước thì bản thân nhiệt kế cũng ảnh hưởng đến nhiệt độ đo được của nước trong ly. Do đó, cảm biến phải được thiết kế sao cho mức ảnh hưởng của nó đến giá trị mà cảm biến đo được là nhỏ nhất.

Smartphone ngày càng đem lại nhiều tiện ích. Ngoài các chức năng cơ bản nghe, gọi, nhắn tin... các smartphone còn có khả năng định vị, thực hiện các hoạt động đa phương tiện. Các cảm biến được tích hợp trên smartphone càng giúp chúng hoạt động hiệu quả hơn, mang đến những trải nghiệm mới cho người dùng. Các cảm biến thường được trang bị cho smartphone là: Accelerometer (gia tốc kế), Magnetometer (cảm biến từ trường), Gyroscope (cảm biến con quay hồi chuyển), Acoustic Sensor (cảm biến âm thanh), Fingerprint Sensor (cảm biến dấu tay), Ambient Light Sensor (cảm biến ánh sáng), Proximity Sensor (cảm biến xa gần).

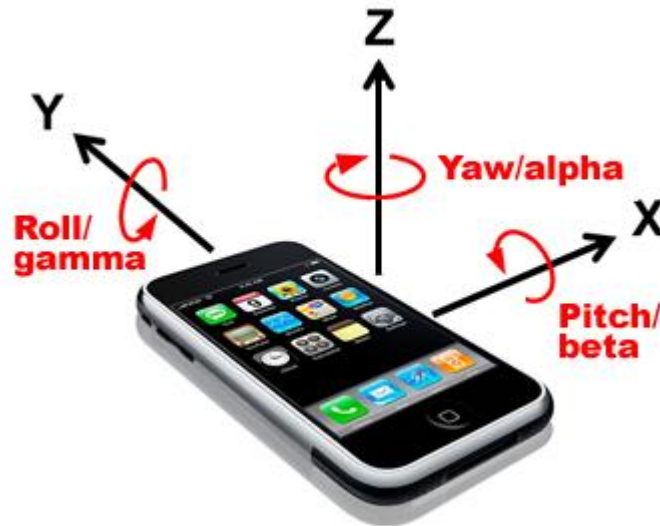
Accelerometer (gia tốc kế) là một thiết bị điện cơ học được dùng để đo gia tốc. Accelerometer thường được ứng dụng trong xác định "gia tốc tĩnh", tức là gia tốc trọng trường. Các gia tốc kế phổ biến trên smartphone đa phần là loại gia tốc kế 3 trục để cho phép ghi nhận gia tốc theo cả 3 chiều trong không gian. Như đã biết, gia tốc trọng trường tại một địa điểm là không đổi. Khi cảm biến quay một góc nào đó, hệ trục tọa độ gắn với cảm biến cũng quay theo nên hình chiếu của gia tốc trọng trường trên các trục sẽ thay đổi. Từ các giá trị đó, sau một khoảng thời gian ta tính được góc quay của cảm biến, gia tốc của chuyển động. Với gia tốc đo được từ gia tốc kế tích hợp trên smartphone, ta có thể nhận biết được nhiều hành động của người dùng với smartphone như di chuyển, xoay, lắc và trạng thái của smartphone (nằm, đứng, nghiêng).

Magnetometer (cảm biến từ trường) là cảm biến được dùng để xác định cường độ từ trường và hướng của từ trường. Vào năm 1833, cảm biến từ trường đầu tiên được phát minh bởi Carl Friedrich Gauss đo từ trường dựa vào hiệu ứng Hall. Có 2 loại cảm biến từ trường cơ bản: scalar magnetometer chỉ đo đặc độ mạnh của từ trường; vector magnetometer có thể đo nhiều thành phần của từ trường như hướng, cường độ. Cảm biến từ trường trang bị cho smartphone khả năng xác định hướng cực Bắc của Trái Đất, định hướng không gian của thiết bị.

Gyroscope (cảm biến con quay hồi chuyển) là một thiết bị được dùng để đo hoặc duy trì hướng dựa vào momen quay. Trong smartphone, cảm biến con quay hồi chuyển được dùng để xác định phương hướng, đo đặc góc quay, tốc độ quay của từng trục trên smartphone. Cảm biến con quay hồi chuyển là loại cảm biến khá mới trên smartphone, iPhone 4 là smartphone đầu tiên được tích hợp cảm biến con quay hồi chuyển để mang lại khả năng điều khiển thiết bị tốt hơn trong các trò chơi.

2.1.1.2 Các góc quay Pitch – Roll – Yaw

Pitch – Roll – Yaw là các giá trị góc quay của thiết bị trên 3 trục tọa độ so với trạng thái ban đầu.



Hình 2-1 Ba giá trị góc quay Pitch – Roll - Yaw

Pitch: góc quay của thiết bị quanh trục dọc theo chiều ngang thân máy (dọc theo trục x). Khi thiết bị nằm ngang thì giá trị *Pitch* bằng 0. Nếu ta xoay thiết bị theo chiều mũi tên quanh trục x thì *Pitch* sẽ tăng dần và ngược lại. Xoay đến khi thiết bị được lật ngược thì $Pitch = +/-180^\circ$

Roll: góc quay của thiết bị quanh trục dọc theo chiều dài thân máy (dọc theo trục y). Khi thiết bị nằm ngang thì giá trị *Roll* bằng 0. Nếu ta xoay thiết bị theo chiều mũi tên quanh trục y thì *Roll* sẽ tăng dần và ngược lại. Xoay đến khi thiết bị được lật ngược thì $Roll = +/-180^\circ$

Yaw: góc quay của thiết bị quanh trục thẳng đứng (dọc theo trục z). Khác với *Raw* và *Pitch*, giá trị *Yaw* khác với *Raw* và *Pitch*, giá trị *Yaw* cho biết hướng của thiết bị. Trong đó, $Yaw = 0/360^\circ$ tương ứng với hướng Bắc, $Yaw = 90^\circ$ tương ứng

với hướng Đông, $Yaw = 180^\circ$ tương ứng với hướng Nam, $Yaw = 270^\circ$ tương ứng với hướng Tây.

2.1.1.3 Ma trận quay

Một cách thuận tiện để mô tả các phép quay là sử dụng một ma trận quay (rotation matrix). Việc chuyển đổi vector theo phép quay chỉ đơn giản là nhân vector với ma trận quay [3]. Ma trận quay theo mỗi trục như sau:

$$R_z = \begin{bmatrix} C_y & S_y & 0 \\ -S_y & C_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_p & S_p \\ 0 & -S_p & C_p \end{bmatrix}$$

$$R_y = \begin{bmatrix} C_r & 0 & -S_r \\ 0 & 1 & 0 \\ S_r & 0 & C_r \end{bmatrix}$$

Với $C = \cos$, $S = \sin$, $y = Yaw$, $p = Pitch$ và $r = Roll$

Ma trận quay R thu được bằng cách nhân ba ma trận quay trên.

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$$

$$R = R_z * R_x * R_y$$

$$R = \begin{bmatrix} C_r C_y + S_r S_p S_y & C_p S_y & -S_r C_y + C_r S_p S_y \\ -C_r S_y + S_r S_p C_y & C_p C_y & S_r S_y + C_r S_p C_y \\ S_r C_p & -S_p & C_r C_p \end{bmatrix}$$

2.1.1.4 Xác định giá trị của các góc quay Pitch – Roll – Yaw dựa vào ma trận quay

Dựa vào ma trận quay thu được, ta xác định được giá trị góc quay Pitch – Roll – Yaw theo công thức:

$$Yaw = atan2(r_2, r_5)$$

$$Pitch = -asin2(r_8)$$

$$Roll = atan2(r_7, r_9)$$

2.1.1.5 Hiện tượng Gimbal lock

Gimbal lock là hiện tượng hai trục quay của một đối tượng chỉ cùng một hướng [3, 4].

Khi thiết bị được dựng thẳng đứng, giá trị $Pitch = 90^\circ$, $Cos(Pitch) = 0$. Chúng tôi không thể sử dụng các phương trình trên vì tất cả các yếu tố ma trận liên quan đến tính toán giá trị $Roll$ và Yaw đều bằng 0. Lúc này hiện tượng Gimbal lock xảy ra, trục tọa độ của góc quay $Roll$ và Yaw đều cùng là trục thẳng đứng.

Để giải quyết vấn đề này, chúng ta chọn trục thẳng đứng là trục của góc quay Yaw và cho $Roll = 0$. Ta có được $Roll = 0$, $Cos(Roll) = 1$, $Sin(Roll) = 0$, $Pitch = +/- 90^\circ$, $Cos(Pitch) = 0$. Kết quả thu được ma trận quay như sau:

$$R = \begin{bmatrix} C_y & 0 & S_p S_y \\ -S_y & 0 & S_p C_y \\ 0 & -S_p & 0 \end{bmatrix}$$

Tính toán Yaw từ giá trị r_1 và r_4 của ma trận quay như sau:

$$Yaw = atan2(-r_4, r_1)$$

2.1.2 Xác định phương hướng bằng cách sử dụng cảm biến gia tốc kết hợp cảm biến từ trường

Việc xác định phương hướng chính là tìm giá trị góc quay Yaw. Trên điện thoại Android, giá trị góc quay có miền giá trị $(-180^\circ, 180^\circ]$. Trong đó, $\text{Yaw} = 0^\circ$ tương ứng với hướng Bắc, $\text{Yaw} = 90^\circ$ tương ứng với hướng Đông, $\text{Yaw} = 180^\circ$ tương ứng với hướng Nam, $\text{Yaw} = -90^\circ$ tương ứng với hướng Tây.

Để tính toán hướng quay của thiết bị cần sử dụng kết hợp cảm biến từ trường (để xác định hướng cực Bắc) và gia tốc kế (dùng để xác định giá trị *Pitch* và *Roll*) hoặc sử dụng cảm biến con quay hồi chuyển. Kết hợp gia tốc kế và cảm biến từ trường cho kết quả không bị ảnh hưởng theo thời gian nhưng cho kết quả kém chính xác khi di chuyển nhanh. Cảm biến con quay hồi chuyển nhạy hơn với di chuyển nhanh nhưng tích lũy lỗi theo thời gian. Vì ứng dụng không cần quá nhạy cảm với di chuyển nhanh nên chúng ta chọn sử dụng kết hợp gia tốc kế và cảm biến từ trường để phương hướng trên Android.

| SENSOR-TYPE | VALUE COMPOSITION | COMMENTARY |
|---------------------|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| TYPE_ACCELEROMETER | value[0] : Lateral value[1] : Longitudinal value[2] : Vertical | Acceleration along three axes in m/s^2 . The Sensor Manager includes a set of gravity constants of the form <code>SensorManager.GRAVITY_*</code> |
| TYPE_GYROSCOPE | value[0] : Azimuth value[1] : Pitch value[2] : Roll | Device orientation in degrees along three axes. |
| TYPE_MAGNETIC_FIELD | value[0] : Lateral value[1] : Longitudinal value[2] : Vertical | Ambient magnetic field measured in microteslas (μT). |

Bảng 2-1 Các giá trị thu được từ cảm biến trên thiết bị Android

(Trích tài liệu [5] – bảng 14-1)

Hướng quay của thiết bị được tính toán trực tiếp từ kết quả nhận từ cảm biến accelerometer và magnetic field [6]. Hướng tiếp cận này cần sử dụng cả cảm biến accelerometer và magnetic field nên chúng ta cần khởi tạo và đăng ký theo dõi 2 cảm biến:

```
Sensor accelerometer =  
    sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);  
Sensor magField =  
    sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);  
  
sensorManager.registerListener(sensorEventListener, accelerometer,  
    SensorManager.SENSOR_DELAY_FASTEST);  
sensorManager.registerListener(sensorEventListener, magField,  
    SensorManager.SENSOR_DELAY_FASTEST);
```

Khi từ trường hoặc gia tốc của thiết bị thay đổi, phương thức `onSensorChanged` trong implementation *SensorEventListener* sẽ được gọi. Các giá trị từ trường hoặc gia tốc được chứa trong mảng *values* của tham số *SensorEvent*.

```
private final SensorEventListener sensorEventListener = new  
    SensorEventListener() {  
    public void onSensorChanged(SensorEvent event) {  
        boolean flag = false;  
  
        // Kiểm tra sự thay đổi của sensor ACCELEROMETER  
        // và MAGNETIC_FIELD  
        switch (event.sensor.getType())  
        {  
        case Sensor.TYPE_ACCELEROMETER:  
            for (int i = 0; i < 3; i++)  
                aValues[i] = event.values[i];  
            flag = true;  
            break;  
        case Sensor.TYPE_MAGNETIC_FIELD:  
            for (int i = 0; i < 3; i++)  
                mValues[i] = event.values[i];  
            flag = true;  
            break;  
        }  
    }  
}
```

Lớp *SensorManager* cung cấp phương thức *getRotationMatrix* để chuyển giá trị thu được từ 2 cảm biến accelerometer và magnetic field thành ma trận quay.

```
float[] R = new float[9];
SensorManager.getRotationMatrix(R, null, aValues, mValues);
```

Từ ma trận quay R có thể tính được các giá trị quay Pitch – Roll – Yaw. Tuy nhiên, người dùng có thể để màn hình thiết bị ở chế độ *rotation* khác nhau: *landscape* hoặc *portrait*. Vì vậy cần phải chuyển đổi ma trận quay về hệ quy chiếu phù hợp. Sử dụng phương thức *remapCoordinateSystem* để chuyển đổi ma trận sang hệ tham chiếu khác.

```
float[] outR = new float[9];
int rotation = ((WindowManager) getSystemService(Context.WINDOW_SERVICE)).
    getDefaultDisplay().getRotation();

int x_axis = SensorManager.AXIS_X;
int y_axis = SensorManager.AXIS_Y;
switch (rotation)
{
    case (Surface.ROTATION_90):
        x_axis = SensorManager.AXIS_Y;
        y_axis = SensorManager.AXIS_MINUS_X;
        break;
    case (Surface.ROTATION_180):
        y_axis = SensorManager.AXIS_MINUS_Y;
        break;
    case (Surface.ROTATION_270):
        x_axis = SensorManager.AXIS_MINUS_Y;
        y_axis = SensorManager.AXIS_X;
        break;
}

SensorManager.remapCoordinateSystem(R, x_axis, y_axis, outR);
```

Cuối cùng, từ ma trận quay, chúng ta áp dụng giải pháp xử lý hiện tượng Gimbal lock đã đề cập ở trên để tính giá trị góc quay Pitch – Roll – Yaw (đơn vị radian)

```
double Yaw, Roll, Pitch;

// Gimbal lock khi Pitch gần đạt +/-90 độ
// Sin(Pitch) gần đạt giá trị +/-1
if (outR[7] > 0.9)
```

```

{
    Yaw = Math.atan2(-outR[3], outR[0]);
    Pitch = Math.PI / 2;
    Roll = 0;
}
else if (outR[7] < -0.9)
{
    Yaw = Math.atan2(-outR[3], outR[0]);
    Pitch = -Math.PI / 2;
    Roll = 0;
}
else
{
    Yaw = Math.atan2(outR[1], outR[4]);
    Pitch = -Math.asin(outR[7]);
    Roll = Math.atan2(outR[6], outR[8]);
}

```

2.1.3 Bộ lọc Kalman

2.1.3.1 Lý do sử dụng bộ lọc

Trên thực tế, giá trị thu được từ cảm biến của thiết bị so với giá trị thật sự có sự khác biệt. Có nhiều yếu tố gây ra độ sai lệch của cảm biến:

- Công nghệ sản xuất, chất lượng sản xuất phần cứng ảnh hưởng đến độ nhạy của cảm biến.
- Thiết bị đặt gần vùng có từ trường mạnh gây nhiễu cho cảm biến.
- Thiết bị lỗi phần cứng trong quá trình sử dụng như chấn động mạnh...

Có thể phân loại các nguyên nhân gây sai lệch giá trị đo của cảm biến thành 2 dạng là: sai lệch một cách hệ thống và sai lệch ngẫu nhiên. Sai lệch một cách hệ thống có thể được giải quyết bằng cách chia lại giá trị đo. Sai lệch ngẫu nhiên có thể được giải quyết bằng quá trình xử lý tín hiệu nhờ các bộ lọc. Ở đây, chúng ta sử dụng bộ lọc Kalman.

2.1.3.2 Giới thiệu bộ lọc Kalman

Bộ lọc Kalman, được Rudolf Emil Kálmán công bố năm 1960, là thuật toán sử dụng chuỗi các giá trị đo lường, bị ảnh hưởng bởi nhiễu hoặc sai số, để ước đoán biến số nhằm tăng độ chính xác so với việc sử dụng duy nhất một giá trị đo lường. Bộ lọc Kalman thực hiện phương pháp truy hồi đối với chuỗi các giá trị đầu vào bị nhiễu, nhằm tối ưu hóa giá trị ước đoán trạng thái của hệ thống. Bộ lọc Kalman được ứng dụng rộng rãi trong kỹ thuật, phổ biến trong các ứng dụng định hướng, định vị và điều khiển các phương tiện di chuyển.

Bộ lọc Kalman đề cập đến bài toán tổng quát đi ước lượng trạng thái $x \in \mathcal{R}^n$ của một quá trình được mô hình hóa một cách rời rạc theo thời gian bằng một phương trình ngẫu nhiên tuyến tính [7]. Phương trình ngẫu nhiên tuyến tính áp dụng cho bộ lọc Kalman được phát biểu như sau:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

Với giá trị đo đạc $z \in \mathcal{R}^n$

$$z_k = Hx_k + v_k$$

Trong đó w và v là hai biến ngẫu nhiên đại diện cho tín hiệu nhiễu hệ thống và tín hiệu nhiễu đo đạc. Chúng ta giả sử các biến này độc lập, có phổ trắng và phân bố Gauss, có phương sai tương ứng là Q và R :

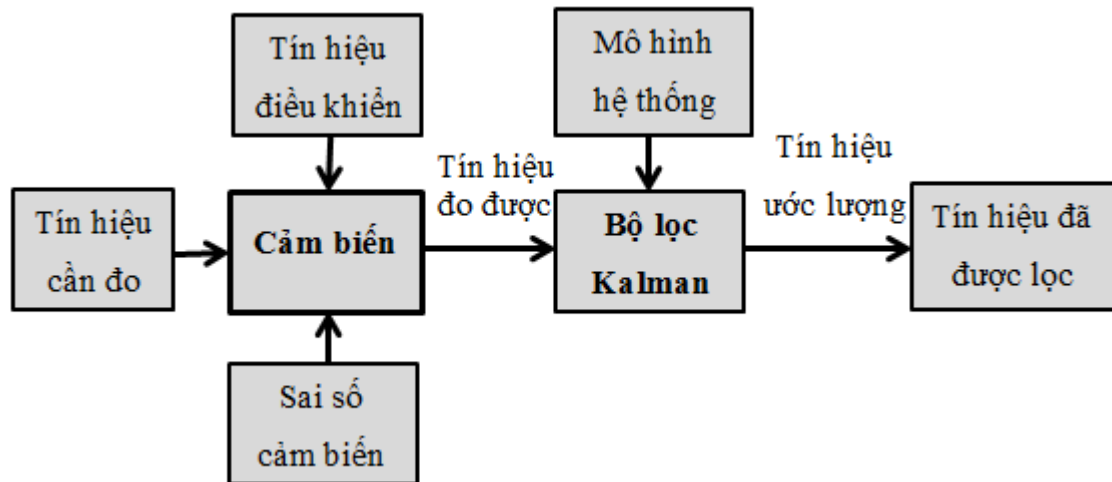
$$P(w) \sim N(0, Q)$$

$$P(v) \sim N(0, R)$$

Nếu x có kích thước n , u có kích thước l , z có kích thước m thì khi đó:

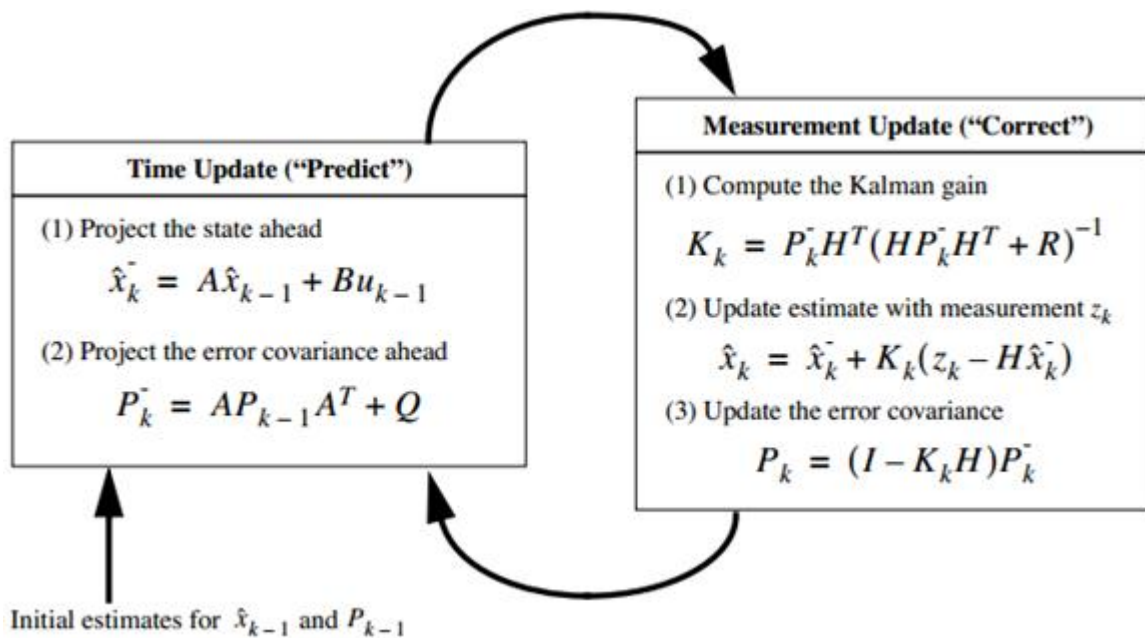
- A : ma trận kích thước $n \times n$. Thể hiện mối quan hệ giữa x và trạng thái trước nó.
- B : ma trận kích thước $n \times l$, là ma trận phụ thuộc vào điều khiển tối ưu u .

- H : ma trận kích thước $m \times n$.
- u : tín hiệu điều khiển.
- w : tín hiệu nhiễu của hệ thống.
- v : tín hiệu nhiễu đo đạc.



Hình 2-2 Mô hình hóa hoạt động của mạch lọc Kalman

Tín hiệu đo được chính là giá trị thu được từ cảm biến. Việc chúng ta cần thực hiện là xác định mô hình hệ thống. Kalman đã đưa ra mô hình bộ hệ thống (mô hình bộ lọc Kalman) như sau:



Hình 2-3 Mô hình xử lý của bộ lọc Kalman

(Trích tài liệu [7] – hình 1-2)

Trong đó:

- P : ma trận sai số của hệ thống.
- \hat{x}_k : giá trị ước lượng
- $\hat{\tilde{x}}_k^-$: giá trị dự đoán được cập nhật từ giá trị ước lượng \hat{x}_{k-1}

Bộ lọc Kalman gồm 2 phần:

- **Time Update**: ước lượng dự đoán. Dựa vào mô hình hệ thống đã thiết lập và giá trị x tiên nghiệm tính được tính toán giá trị ước đoán x. Và điều chỉnh giá trị sai số P.
- **Measurement Update**: làm chính xác dự đoán. Tính toán giá trị thật x dựa vào giá trị ước đoán, giá trị đo được và sai số ước lượng.

2.1.3.3 Áp dụng bộ lọc Kalman vào lọc nhiễu tín hiệu cảm biến

Để sử dụng bộ lọc Kalman, ta cần mô hình hóa hệ thống theo hai phương trình ngẫu nhiên tuyến tính của bộ lọc Kalman.

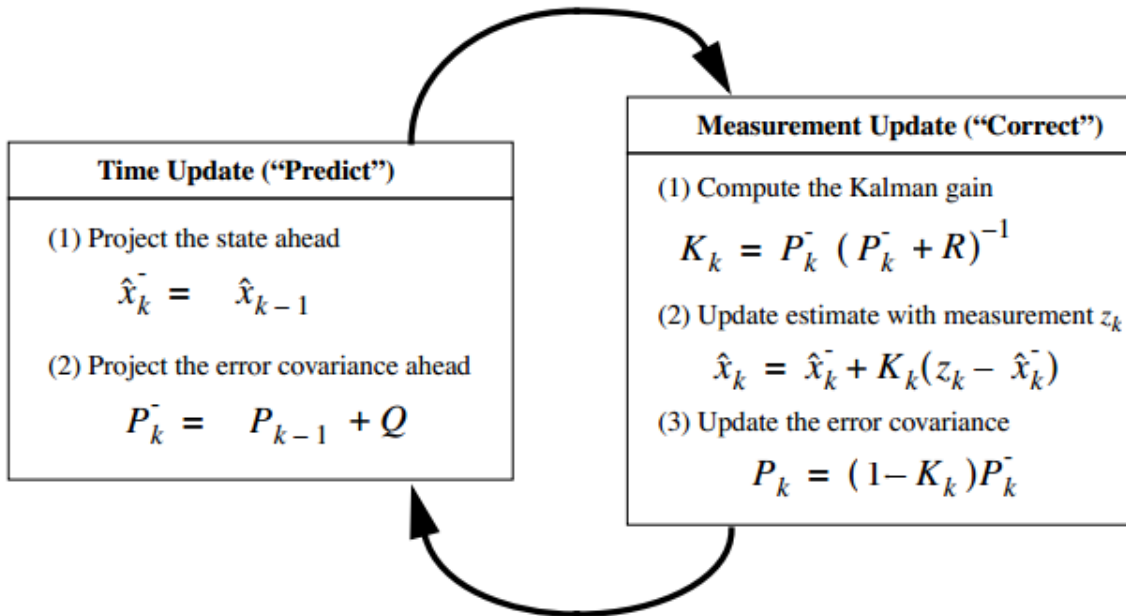
Tín hiệu cảm biến là tín hiệu số tương ứng biến có kích thước là 1. Tín hiệu cảm biến không có tín hiệu điều khiển u nên $u = 0$. Trạng thái không thay đổi qua các bước nên $A = 1$. Đo đặc nhiễu được tính trực tiếp ở mỗi trạng thái nên $H = 1$. Mô hình của bài toán:

$$x_k = x_{k-1} + w_{k-1}$$

Với giá trị đo đặc $z \in \mathcal{R}^n$

$$z_k = x_k + v_k$$

Do đó, ta xây dựng mô hình hệ thống:

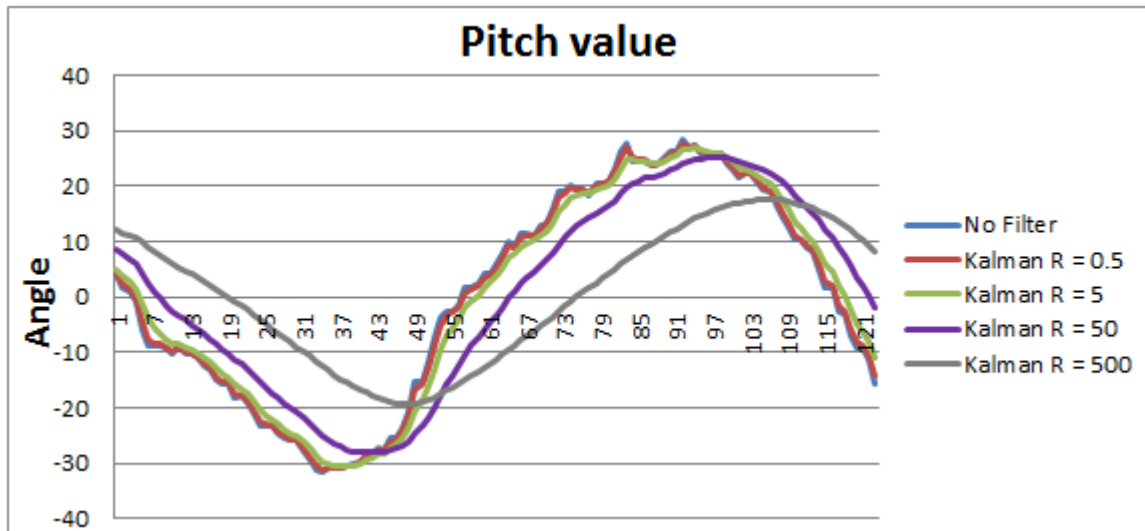


Hình 2-4 Mô hình xử lý của bộ lọc Kalman cho tín hiệu cảm biến

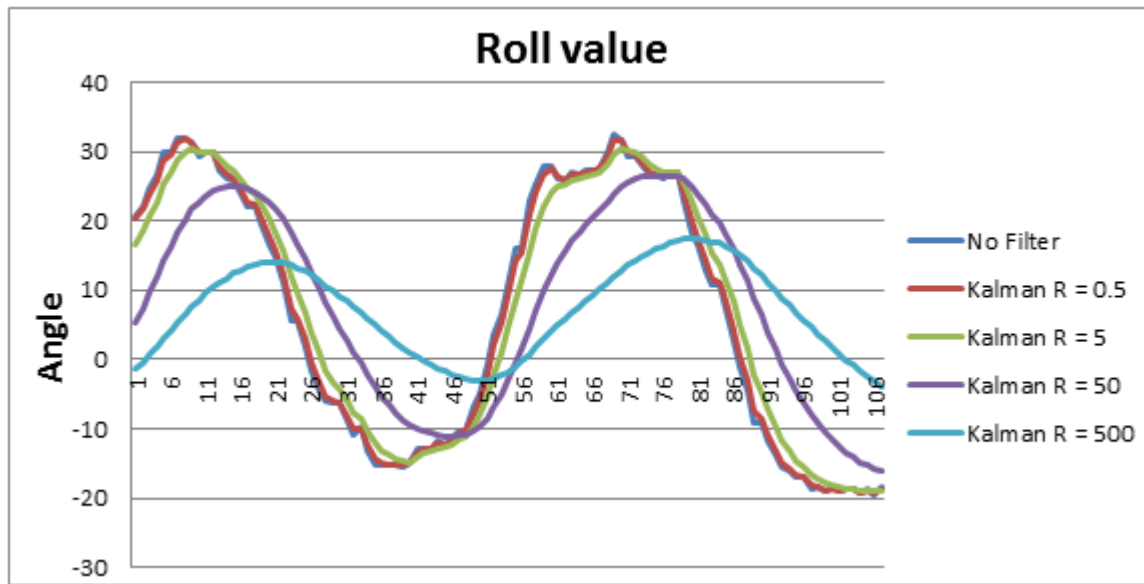
2.1.3.4 Đánh giá kết quả

Chúng ta chỉ lấy giá trị thu được từ cảm biến để lọc nên có thể cho lỗi hệ thống Q nhỏ. Để hệ thống đáp ứng được những thay đổi nhạy hơn ta chọn $Q = 1$. Giá trị P phải khác 0 và sau nhiều bước thực hiện, P sẽ đến điểm hội tụ nên ta có thể chọn $P = 2$. Giá trị R có ảnh hưởng đến độ mịn của dữ liệu nên chúng ta cần thực hiện nhiều thử nghiệm ứng với các giá trị R khác nhau để chọn R phù hợp. Giá trị góc quay Pitch – Roll – Yaw có giá trị từ -180° đến 180° .

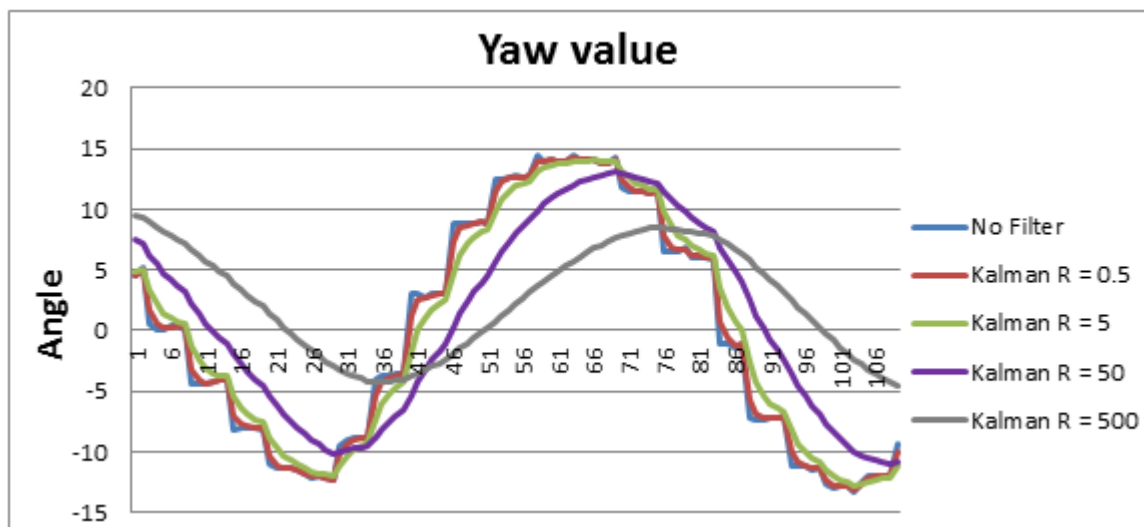
Chúng ta thực hiện thử nghiệm với R lần lượt bằng 0,5; 5; 50 và 500. Thu được kết quả như sau:



Hình 2-5 Xử lý tín hiệu Pitch bằng bộ lọc Kalman



Hình 2-6 Xử lý tín hiệu Roll bằng bộ lọc Kalman



Hình 2-7 Xử lý tín hiệu Yaw bằng bộ lọc Kalman

Khi áp dụng bộ lọc Kalman với $R = 0,5$, giá trị góc quay thu về sau khi lọc vẫn còn rất nhiều nhiễu. Biểu đồ thể hiện giá trị góc quay được lọc với $R = 0,5$ gần giống giá trị góc quay không được lọc.

Khi áp dụng bộ lọc Kalman với $R = 50$ và $R = 500$, giá trị góc quay thu được quá mịn, chậm phản hồi với những thay đổi nhanh.

Áp dụng bộ lọc Kalman với $R = 5$ cho kết quả mịn, ít nhiễu, có phản hồi nhanh với sự thay đổi. Vì vậy, chúng ta sẽ chọn $R = 5$ cho bộ lọc Kalman để áp dụng lọc nhiễu giá trị góc quay.

Ngoài ra, sau 1000 lần cập nhật với $R = 5$ và $Q = 1$, giá trị P hội tụ về giá trị 1,79129.

2.2 Phân tích và thiết kế ứng dụng minh họa

2.2.1 Mục tiêu

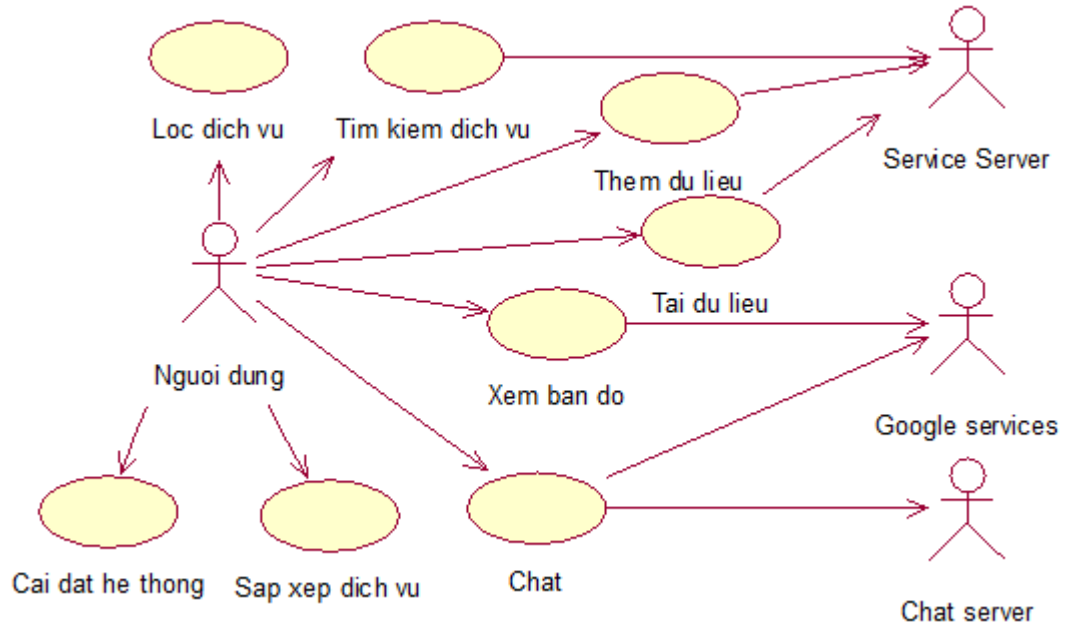
Tận dụng thế mạnh về các thiết bị ngoại vi và cảm biến trên smartphone và áp dụng công nghệ thực tại ảo tăng cường để xây dựng ứng dụng tìm kiếm dịch vụ mới, khả năng tương tác cao, thân thiện và hấp dẫn. Ngoài ra, trong quá trình xây dựng ứng dụng, nhóm cũng trao đổi kỹ thuật lập trình thiết bị di động, tìm hiểu và sử dụng nhiều thư viện API do Android cung cấp.

2.2.2 Yêu cầu ứng dụng

- Hiển thị và cung cấp thông tin dịch vụ theo hướng nhìn của người dùng.
- Theo dõi phương hướng đến một dịch vụ cụ thể.
- Lọc và sắp xếp danh sách dịch vụ hiển thị.
- Tìm kiếm dịch vụ với từ khóa tiếng Việt có dấu hay không dấu.
- Tải dữ liệu mới từ server.
- Thêm mới dịch vụ từ phía người dùng.
- Hiển thị các dịch vụ trên bản đồ.
- Tìm đường đi.
- Tự động chuyển từ màn hình hiển thị danh sách dịch vụ sang màn hình bản đồ khi người dùng đặt thiết bị song song với mặt đất.
- Đăng nhập, kết bạn, gửi tin nhắn cho bạn bè. Hiển thị danh sách bạn bè.

2.2.3 Mô hình Use case

2.2.3.1 Sơ đồ Use case



Hình 2-8 Sơ đồ Use case

Danh sách các Use case:

| STT | Tên Use-case | Ý nghĩa/Ghi chú |
|-----|------------------|----------------------------------------------------------------------------------------------------------|
| 1 | Lọc dịch vụ | Chức năng lọc danh sách dịch vụ theo hướng nhìn của người dùng, theo loại dịch vụ hoặc theo khoảng cách. |
| 2 | Tìm kiếm dịch vụ | Cho phép người dùng nhập từ khóa tìm kiếm tiếng Việt, thực hiện tìm kiếm tại client hoặc server. |
| 3 | Thêm dữ liệu | Người dùng thêm dịch vụ mới vào cơ sở dữ liệu. |
| 4 | Tải dữ liệu | Cập nhật dữ liệu từ server về thiết bị của người dùng. |
| 5 | Xem bản đồ | Chức năng hiển thị bản đồ và các biểu tượng cho |

| | | |
|---|------------------|--------------------------------------------------------------------------------------------------|
| | | dịch vụ. Người dùng có thể thay đổi vị trí của thiết bị và tìm đường đi. |
| 6 | Chat | Cho phép đăng nhập, kết bạn, nhắn tin với bạn bè bằng tài khoản Gmail. |
| 7 | Sắp xếp dịch vụ | Cho phép người dùng sắp xếp danh sách dịch vụ theo khoảng cách, theo tên hoặc theo loại dịch vụ. |
| 8 | Cài đặt hệ thống | Thiết lập các thông số của ứng dụng |

Bảng 2-2 Danh sách các Use case

2.2.3.2 Đặc tả Use case

2.2.3.2.1 Loại dịch vụ

- Tóm tắt: lọc danh sách dịch vụ theo hướng nhìn của người dùng, theo loại dịch vụ hoặc theo khoảng cách.
- Dòng sự kiện:
 - Dòng sự kiện chính:
 - Bước 1: Người dùng mở ứng dụng.
 - Bước 2: Màn hình chính hiện lên, nạp các dịch vụ vào danh sách hiển thị lên màn hình.
 - Bước 3: Người dùng thực hiện một trong 3 luồng sau đây:
 - Người dùng chọn mục “Distance” tại menu của ứng dụng: thực hiện luồng phụ lọc danh sách dịch vụ theo khoảng cách. Người dùng sẽ chọn một trong các khoảng cách 100m, 100m - 200m, 200m - 500m, 500m - 1000m, 1000m - 2000m, 2000m - 5000m.
 - Người dùng chọn mục “Setting” tại menu của ứng dụng, nếu chọn “Automatically filter services by orientation” thì luồng phụ lọc dữ liệu theo hướng nhìn của người dùng được gọi. Nếu chọn “Display

services icon” thì các dịch vụ được hiển thị dạng biểu tượng trên màn hình.

- Luồng phụ lọc dữ liệu theo loại dịch vụ được gọi khi người dùng đánh dấu vào các loại dịch vụ được hiển thị trên màn hình.
- Dòng sự kiện phụ: Không có.
- Các yêu cầu đặc biệt: Không có.
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case: ứng dụng ở màn hình chính.
- Trạng thái hệ thống sau khi thực hiện Use-case: danh sách dịch vụ đã được lọc.
- Điểm mở rộng: không có.

2.2.3.2.2 Tìm kiếm dịch vụ

- Tóm tắt: chức năng cho phép tìm kiếm dịch vụ trong cơ sở dữ liệu tại client hoặc server.
- Dòng sự kiện:
 - Dòng sự kiện chính:
 - Bước 1: Người dùng mở ứng dụng.
 - Bước 2: Người dùng nhấn phím *menu*.
 - Bước 3: Người dùng chọn mục “Search” trên màn hình.
 - Bước 4: Hệ thống chuyển sang màn hình tìm kiếm dịch vụ.
 - Bước 5: Người dùng nhập từ khóa tìm kiếm (từ khóa tiếng Việt có dấu hoặc không dấu).
 - Bước 6: Người dùng nhấn nút “Search” để tìm kiếm dịch vụ.
 - Bước 7: Hệ thống hiển thị danh sách dịch vụ có tên dịch vụ chứa từ khóa tìm kiếm.
 - Dòng sự kiện phụ:
 - Tại bước 6, nếu người dùng nhấn nút “Search” mà không tìm thấy kết quả thì:

- Hệ thống hiển thị thông báo gợi ý người dùng tìm kiếm tại server.
- Người dùng nhấn nút “Yes” để tiếp tục tìm kiếm tại server, nhấn nút “No” để hủy tìm kiếm.
- Tại bước 6, người dùng nhấn biểu tượng cài đặt, nút “Search on server” hiện ra cho phép tìm kiếm trực tiếp tại server.
- Các yêu cầu đặc biệt: nếu tìm tại server thì thiết bị phải có kết nối mạng.
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case: ứng dụng ở màn hình chính.
- Trạng thái hệ thống sau khi thực hiện Use-case: ứng dụng ở màn hình tìm kiếm.
- Điểm mở rộng: không có.

2.2.3.2.3 Thêm dữ liệu

- Tóm tắt: cho phép thêm dữ liệu mới từ phía người dùng.
- Dòng sự kiện:
 - Dòng sự kiện chính:
 - Bước 1: Người dùng mở ứng dụng.
 - Bước 2: Người dùng nhấn phím *menu*.
 - Bước 3: Người dùng chọn mục “Add service” trên màn hình.
 - Bước 4: Hệ thống chuyển sang màn hình thêm dịch vụ mới.
 - Bước 5: Người dùng nhập tên, địa chỉ, điện thoại và mô tả của dịch vụ.
 - Bước 6: Người dùng chọn loại dịch vụ.
 - Bước 7: Người dùng nhấn nút “Set Location” để điều chỉnh vị trí của dịch vụ.
 - Bước 8: Người dùng nhấn nút “OK” để hoàn tất thêm dịch vụ, hoặc “Cancel” để hủy luồng thực hiện.
 - Dòng sự kiện phụ:
 - Tại bước 6, nếu người dùng không chọn loại dịch vụ thì mặc định loại dịch vụ là “Hotel”.

- Tại bước 7, nếu người dùng không nhấn nút “Set Location” thì mặc định vị trí dịch vụ chính là vị trí của người dùng.
- Tại bước 8, nếu người dùng nhấn “OK” thì xuất hiện thông báo xác nhận truyền dữ liệu lên server.
- Các yêu cầu đặc biệt: thiết bị có kết nối mạng.
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case: ứng dụng ở màn hình chính.
- Trạng thái hệ thống sau khi thực hiện Use-case: dịch vụ được thêm vào cơ sở dữ liệu.
- Điểm mở rộng: không có.

2.2.3.2.4 Tải dữ liệu

- Tóm tắt: cho phép cập nhật dữ liệu từ server về cơ sở dữ liệu tại client. Người dùng có thể tải dịch vụ hoặc thông tin khuyến mãi.
- Dòng sự kiện:
 - Dòng sự kiện chính:
 - Bước 1: Người dùng mở ứng dụng
 - Bước 2: Người dùng nhấn phím *menu*.
 - Bước 3: Người dùng thực hiện một trong 2 luồng phụ sau đây:
 - Người dùng chọn mục “Download” tại menu của ứng dụng: thực hiện luồng phụ tải danh sách dịch vụ từ server. Các dịch vụ được tải về thuộc tất cả các loại dịch vụ và có khoảng cách từ dịch vụ đến vị trí người dùng nhỏ hơn khoảng cách lọc danh sách dịch vụ hiển thị tại màn hình chính.
 - Người dùng chọn mục “Discount Information” tại menu của ứng dụng: thực hiện luồng phụ tải các thông tin khuyến mãi. Các dịch vụ có khuyến mãi sẽ hiển thị lên màn hình.
 - Bước 4: Hệ thống hiển thị danh sách dịch vụ được tải về.

- Dòng sự kiện phụ: Không có.
- Các yêu cầu đặc biệt: Thiết bị có kết nối mạng.
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case: Ứng dụng ở màn hình chính.
- Trạng thái hệ thống sau khi thực hiện Use-case: Cơ sở dữ liệu được cập nhật.
- Điểm mở rộng: Không có.

2.2.3.2.5 Xem bản đồ

- Tóm tắt: Chức năng hiển thị bản đồ và các biểu tượng cho dịch vụ. Người dùng có thể thay đổi vị trí của thiết bị và tìm đường đi.
- Dòng sự kiện:
 - Dòng sự kiện chính:
 - Bước 1: Người dùng mở ứng dụng.
 - Bước 2: Người dùng nhấn phím *menu*.
 - Bước 3: Chọn mục “Google Map” trên màn hình.
 - Bước 4: Hệ thống chuyển sang màn hình bản đồ.
 - Bước 5: Người dùng thực hiện một trong các luồng phụ sau đây:
 - Người dùng chọn mục “Services” tại menu của ứng dụng: thực hiện luồng phụ lọc danh sách dịch vụ theo loại dịch vụ.
 - Người dùng chọn mục “Distance” tại menu của ứng dụng: thực hiện luồng phụ lọc danh sách dịch vụ theo khoảng cách.
 - Người dùng chọn mục “Driving Directions” tại menu của ứng dụng: thực hiện luồng phụ tìm đường đi. Người dùng kéo thả vị trí bắt đầu và vị trí kết thúc rồi nhấn “OK” để tìm đường đi.
 - Người dùng chọn mục “Your Location” tại menu của ứng dụng: thực hiện luồng phụ chuyển vị trí trung tâm của bản đồ về vị trí của người dùng.

- Người dùng chọn mục “Clear” tại menu của ứng dụng: thực hiện luồng phụ xóa các biểu tượng trên bản đồ.
- Người dùng chọn mục “Change Your Location” tại menu của ứng dụng: thực hiện luồng phụ thay đổi vị trí người dùng. Người dùng kéo thả biểu tượng bàn bay chỉ xuống để thay đổi vị trí người dùng.
- Người dùng chọn mục “Online Friend Lists” tại menu của ứng dụng: thực hiện luồng phụ hiển thị vị trí bạn bè trên bản đồ.
- Dòng sự kiện phụ: Không có.
- Các yêu cầu đặc biệt: Thiết bị có kết nối mạng khi thực hiện luồng tìm đường đi.
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case: Ứng dụng ở màn hình chính.
- Trạng thái hệ thống sau khi thực hiện Use-case: Ứng dụng ở màn hình bản đồ.
- Điểm mở rộng: Không có.

2.2.3.2.6 Chat

- Tóm tắt: Cho phép đăng nhập, kết bạn, nhắn tin với bạn bè bằng tài khoản Gmail.
- Dòng sự kiện:
 - Dòng sự kiện chính:
 - Bước 1: Người dùng mở ứng dụng.
 - Bước 2: Người dùng nhấn phím *menu*.
 - Bước 3: Người dùng chọn mục “Chat” trên màn hình.
 - Bước 4: Hệ thống chuyển sang màn hình Chat.
 - Bước 5: Người dùng nhập tài khoản gmail để đăng nhập vào tính năng Chat.
 - Bước 6: Người dùng chọn một trong các luồng phụ sau:

- Người dùng chọn nút “Add Friend” tại menu của ứng dụng: thực hiện luồng phụ gửi lời mời kết bạn. Người dùng nhập tài khoản Gmail của bạn bè và nhấn nút “Send Request” để hoàn tất gửi lời mời kết bạn.
- Người dùng chọn nút “Your Requests” tại menu của ứng dụng: thực hiện luồng phụ xác nhận yêu cầu kết bạn.
- Người dùng chọn nút “Recipient” tại menu của ứng dụng: thực hiện luồng phụ xem danh sách bạn bè, chọn người nhận tin nhắn.
- Người dùng nhấn nút “Send”: thực hiện luồng phụ gửi tin nhắn.
- Dòng sự kiện phụ:
 - Tại bước 5, nếu thiết bị không có kết nối mạng thì ứng dụng hiện thông báo yêu cầu kiểm tra kết nối mạng.
 - Tại bước 5, nếu người dùng nhập sai tài khoản hoặc mật khẩu thì thông báo đăng nhập thất bại.
- Các yêu cầu đặc biệt: Thiết bị có kết nối mạng.
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case: Ứng dụng ở màn hình chính.
- Trạng thái hệ thống sau khi thực hiện Use-case: Ứng dụng ở màn hình Chat.
- Điểm mở rộng: Không có.

2.2.3.2.7 Sắp xếp dịch vụ

- Tóm tắt: Cho phép người dùng sắp xếp danh sách dịch vụ theo khoảng cách, theo tên hoặc theo loại dịch vụ.
- Dòng sự kiện: Người dùng mở ứng dụng.
 - Dòng sự kiện chính:
 - Bước 1: Người dùng mở ứng dụng.
 - Bước 2: Người dùng nhấn phím *menu*.

- Bước 3: Người dùng chọn mục “Sort by” tại menu của màn hình chính.
- Bước 4: Hệ thống hiển thị danh sách các kiểu sắp xếp trên màn hình.
- Bước 5: Người dùng thực hiện một trong các luồng sau đây:
 - Người dùng chọn mục “Distance” tại menu của ứng dụng: hệ thống thực hiện sắp xếp danh sách dịch vụ theo thứ tự tăng dần của khoảng cách từ vị trí dịch vụ đến vị trí người dùng.
 - Người dùng chọn mục “Name” tại menu của ứng dụng: hệ thống thực hiện sắp xếp danh sách dịch vụ theo tên dịch vụ tăng dần.
 - Người dùng chọn mục “Category” tại menu của ứng dụng: hệ thống thực hiện sắp xếp danh sách dịch vụ theo loại dịch vụ tăng dần.
- Bước 6: Hệ thống hiển thị danh sách đã được sắp xếp.
 - Dòng sự kiện phụ: Không có.
- Các yêu cầu đặc biệt: Không có.
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case: Ứng dụng ở màn hình chính.
- Trạng thái hệ thống sau khi thực hiện Use-case: Hiển thị danh sách dịch vụ đã sắp xếp.
- Điểm mở rộng: không có.

2.2.3.2.8 Cài đặt hệ thống

- Tóm tắt: Thiết lập các thông số của hệ thống.
- Dòng sự kiện:
 - Dòng sự kiện chính:
 - Bước 1: Người dùng mở ứng dụng.
 - Bước 2: Người dùng nhấn phím *menu*.
 - Bước 3: Người dùng chọn mục “Settings” trên màn hình.
 - Bước 4: Hệ thống chuyển sang màn hình cài đặt hệ thống.

- Bước 5: Người dùng thực hiện một trong các luồng phụ sau đây:
 - Người dùng chọn mục “Automatically change screen” tại menu của ứng dụng: thực hiện luồng phụ tự động chuyển đổi giữa màn hình chính và màn hình bản đồ. Khi được kích hoạt, giao diện chính sẽ chuyển sang màn hình bản đồ nếu góc nghiêng giữa thiết bị với mặt đất $< 10^\circ$. Màn hình bản đồ sẽ chuyển sang giao diện chính nếu góc nghiêng giữa thiết bị với mặt đất $> 70^\circ$.
 - Người dùng chọn mục “Automatically filter services by orientation” tại menu của ứng dụng: thực hiện luồng phụ lọc danh sách dịch vụ theo hướng nhìn của người dùng.
 - Người dùng chọn mục “Display service icon” tại menu của ứng dụng: thực hiện luồng phụ hiển thị các dịch vụ trên màn hình dạng biểu tượng.
 - Người dùng chọn mục “Automatically update your location” tại menu của ứng dụng: thực hiện luồng phụ tự động cập nhật vị trí của người dùng.
- Bước 6: Người dùng nhấn nút “OK” để lưu lại các thông số của hệ thống, hoặc nhấn nút “Cancel” để hủy thay đổi các thông số của hệ thống.
 - Dòng sự kiện phụ: Không có.
- Các yêu cầu đặc biệt: Chức năng “Display service icon” chỉ có thể kích hoạt khi chức năng “Automatically filter services by orientation” đã được kích hoạt trước đó.
- Trạng thái hệ thống khi bắt đầu thực hiện Use-case: Ứng dụng ở màn hình chính.
- Trạng thái hệ thống sau khi thực hiện Use-case: Các thông số của hệ thống được lưu trữ.

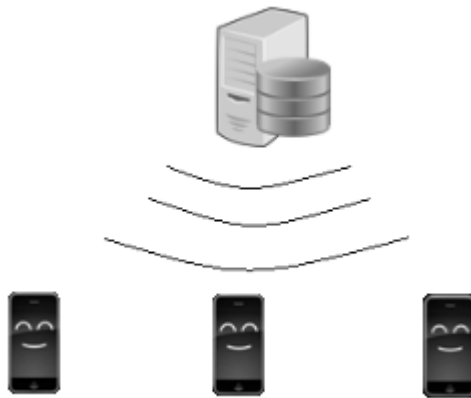
- Điểm mở rộng: Không có.

2.2.4 Thiết kế kiến trúc

Sau khi xem xét yêu cầu, quy mô của ứng dụng, khả năng của nhóm và một số điều kiện cơ sở hạ tầng hiện tại, nhóm phát triển dự định áp dụng một số kiến trúc sau để xây dựng ứng dụng.

2.2.4.1 Client-Server

Kiến trúc client-server là kiến trúc chính của ứng dụng.



Hình 2-9 Kiến trúc client – server

Server được xây dựng là nơi chứa tất cả các dữ liệu về dịch vụ, về thông tin tài khoản (nếu có) của khách hàng. Theo ý tưởng thiết kế, dữ liệu được lưu trữ nhất quán tại server, các thao tác cập nhật dữ liệu lên server từ client, hay trang web của ứng dụng sẽ phải trải qua một quá trình “kiểm soát” bởi đội ngũ chuyên biệt, dữ liệu được cho là đúng đắn, thỏa các tiêu chí đề ra sẽ được đưa lên server, chính thức có thể được sử dụng bởi client.

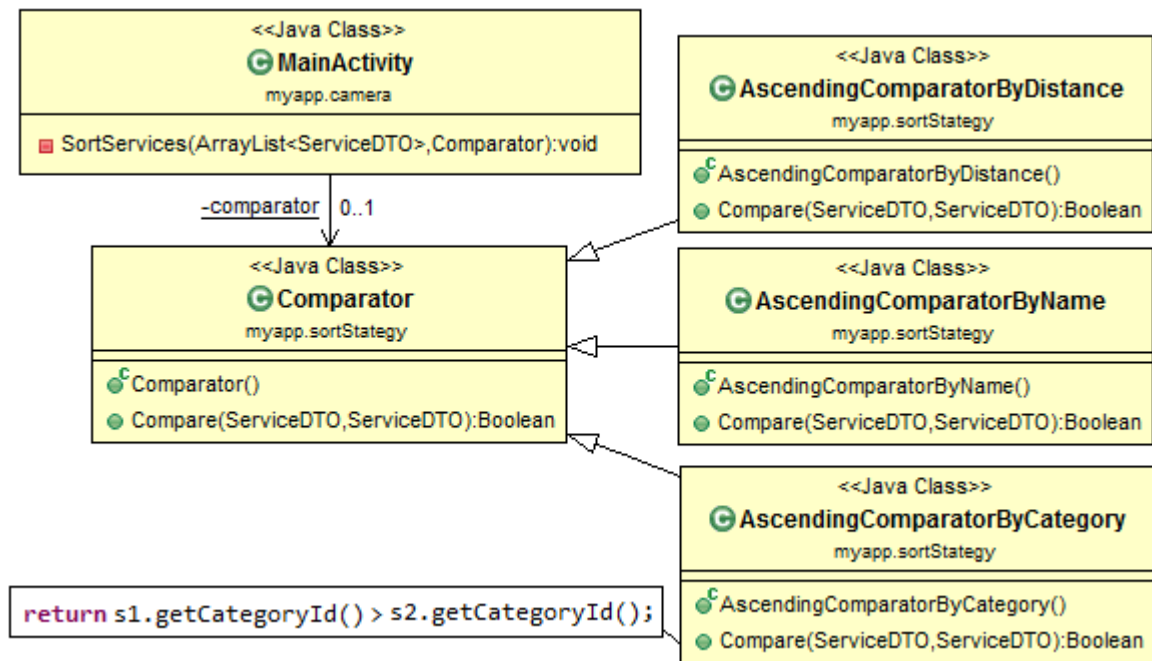
Server được xây dựng theo Web API, cung cấp các phương thức truy vấn theo kiểu Restful, tạo các điều kiện cho client có thể lấy dữ liệu một cách thuận lợi nhất.

Về phía client, các dữ liệu cần thiết cho việc xử lý logic đều có thể được lấy thông qua truy vấn lên server. Tuy nhiên, để phục vụ tốt hơn cho đối tượng người dùng không có mạng thường xuyên hoặc tốc độ truy cập mạng chậm, client có thể lưu trữ dữ liệu, để các lần xử lý sau, trực tiếp lấy dữ liệu từ nó mà không cần truy xuất lên server.

2.2.4.2 Một số kiến trúc khác

Bên cạnh kiến trúc client-server, để hiện thực hóa ứng dụng, nhóm đã áp dụng thêm một số kiến trúc khác: Web API được xây dựng theo mô hình MVC, áp dụng kiến trúc 3-layers. Chi tiết đã được nêu rõ hơn trong phần hiện thực hóa ứng dụng ở chương 3.

Chương trình có áp dụng mẫu thiết kế Strategy trong việc lựa chọn phương thức sắp xếp danh sách dịch vụ [8]. Lớp dịch vụ không phải là kiểu dữ liệu cơ bản mà được xây dựng gồm nhiều thuộc tính khác nhau. Nên không thể sử dụng các phép toán so sánh “<”, “>” để sắp xếp danh sách dịch vụ theo khoảng cách, theo tên hoặc theo loại dịch vụ. Nhằm sắp xếp dữ liệu với nhiều cách thức so sánh khác nhau, chúng em đã áp dụng mẫu Strategy.



Hình 2-10 Áp dụng mẫu Strategy trong sắp xếp danh sách dịch vụ

Lớp cha *Comparator* cung cấp phương thức ảo *Compare()* với tham số đầu vào là 2 đối tượng kiểu *ServiceDTO* cần so sánh, kết quả trả về là giá trị luận lý của phép so sánh 2 đối tượng. Chúng em đã xây dựng 3 lớp con *AscendingComparatorByDistance*, *AscendingComparatorByName* và *AscendingComparatorByCategory* kế thừa lớp cha *Comparator* tương ứng với kiểu so sánh tăng dần theo khoảng cách, theo tên và theo loại dịch vụ. Các lớp con kế thừa và thực hiện cài đặt lại phương thức *Compare()* từ lớp cha.

2.2.5 Thiết kế cơ sở dữ liệu

Hiện tại quy mô của ứng dụng còn nhỏ, chỉ ở mức độ nghiên cứu học hỏi, cho nên cơ sở dữ liệu được nhóm phát triển thống nhất là thiết kế đơn giản, gọn nhẹ, tuy nhiên vẫn phải đáp ứng đầy đủ yêu cầu lưu trữ thông tin của ứng dụng.

2.2.5.1 Ngôn ngữ, nền tảng

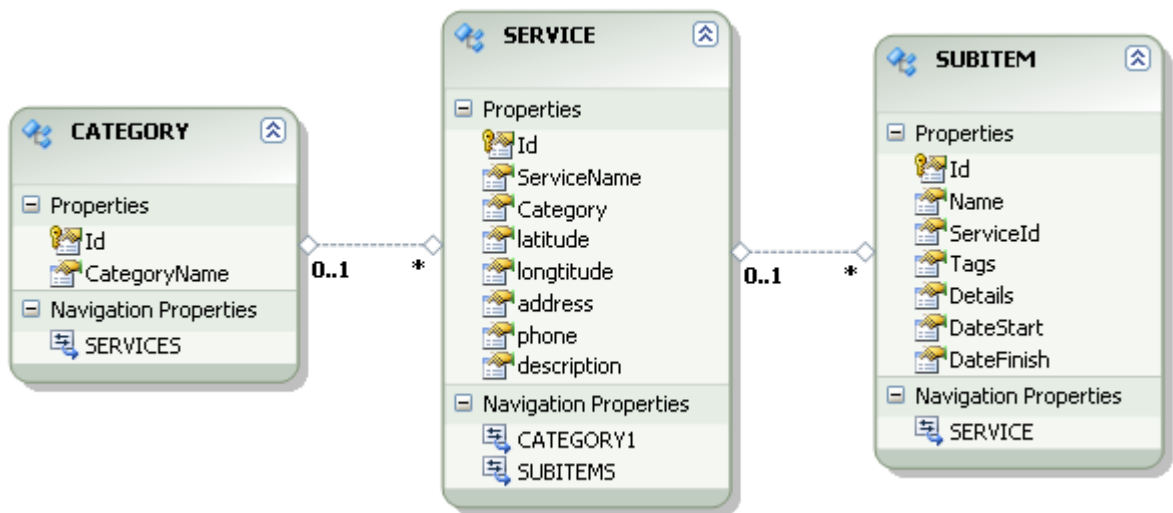
Server được xây dựng trên nền tảng ASP .NET 4.0, cho nên để cho việc tương tác được thuận tiện, cơ sở dữ liệu được chọn là SQL.

2.2.5.2 Cấu trúc cơ sở dữ liệu

Server được xây dựng trên nền tảng ASP .NET 4.0, cho nên để cho việc tương tác được thuận tiện, cơ sở dữ liệu được chọn là SQL.

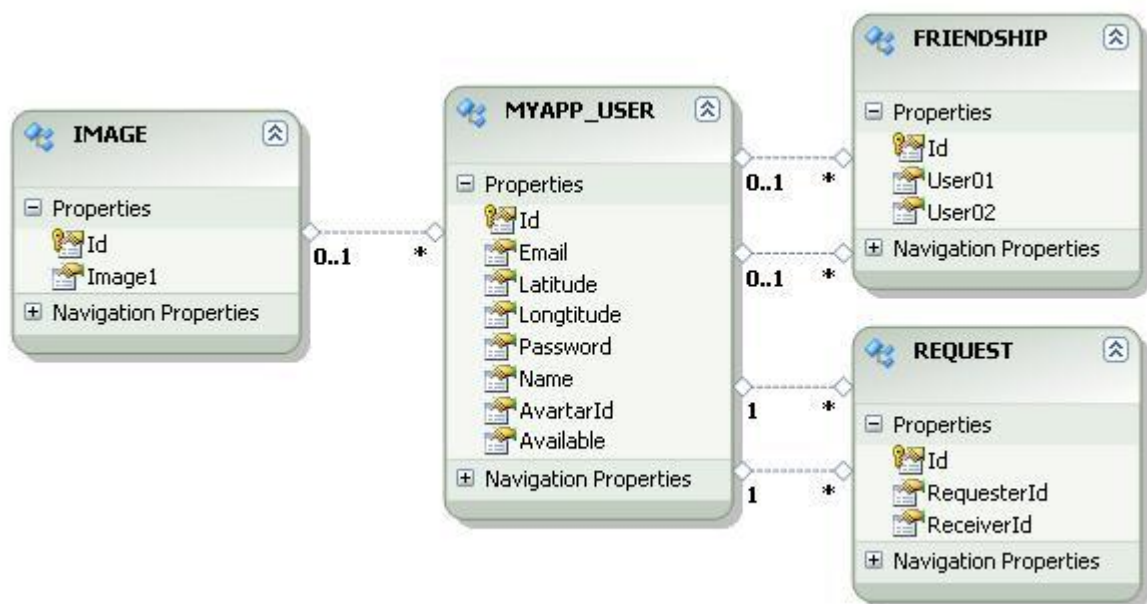
Để phục vụ cho ứng dụng, nhóm nghiên cứu thiết kế 2 cơ sở dữ liệu độc lập phục vụ cho từng nhu cầu cụ thể:

Cơ sở dữ liệu dịch vụ: lưu trữ các thông tin về dịch vụ, khuyến mãi, các thông tin liên quan khác, ... phục vụ cho chức năng chính của ứng dụng là tìm kiếm dịch vụ.



Hình 2-11 Sơ đồ thực thể của cơ sở dữ liệu dịch vụ

- Cơ sở dữ liệu cho chức năng chat: Lưu trữ các thông tin của người dùng như tên truy nhập, bạn bè, yêu cầu kết bạn,... nếu họ có nhu cầu sử dụng tính năng chat.



Hình 2-12 Sơ đồ thực thể của cơ sở dữ liệu Chat

2.2.6 Giao diện

Tên của ứng dụng: *Search Master*.

Biểu tượng của ứng dụng:



Hình 2-13 Biểu tượng của ứng dụng *Search Master*

Danh sách các màn hình

| STT | Tên màn hình | Ý nghĩa/Ghi chú |
|-----|-------------------------------------|----------------------------------------------------------------------------------------|
| 1 | Màn hình chính | Màn hình hiển thị danh sách dịch vụ được lọc và sắp xếp theo tùy chỉnh của người dùng. |
| 2 | Màn hình theo dõi hướng đến dịch vụ | Màn hình theo dõi hướng từ vị trí người dùng đến vị trí dịch vụ cụ thể. |

| | | |
|---|---------------------------|---------------------------------------------------------------------------------------------------------------------|
| 3 | Màn hình tìm kiếm | Màn hình cho người dùng nhập từ khóa tìm kiếm và thực hiện tìm kiếm tại server hoặc client. |
| 4 | Màn hình bản đồ | Màn hình hiển thị các dịch vụ dạng biểu tượng trên bản đồ. Cho phép tìm đường đi và thay đổi vị trí của người dùng. |
| 5 | Màn hình Chat | Màn hình cho phép người dùng đăng nhập bằng tài khoản Gmail, thực hiện kết bạn và gửi tin nhắn cho bạn bè. |
| 6 | Màn hình thêm dịch vụ mới | Màn hình cho người dùng nhập vào thông tin của dịch vụ mới và truyền lên server. |
| 7 | Màn hình cài đặt hệ thống | Màn hình cho phép người dùng thay đổi các thông số của hệ thống. |

Bảng 2-3 Danh sách các màn hình của ứng dụng

Chi tiết màn hình và cách thức sử dụng sẽ được giới thiệu ở chương 3.

2.3 Cài đặt ứng dụng

2.3.1 Lấy hình ảnh từ camera của điện thoại

Màn hình hiển thị chính của ứng dụng là màn hình camera, hiển thị khung cảnh thực tế ngoài đời thực, các kết quả trả về cũng được hiển thị trên đó. Cho nên, chúng ta cần xử lý để thu được hình ảnh từ camera của smartphone.

Hầu hết các smartphone hiện nay đều có tích hợp camera trong hệ thống phần cứng, và cũng đã được hỗ trợ mạnh mẽ từ hệ điều hành di động. Android cũng không ngoại lệ. Trong thư viện các hàm API của Android SDK đã cung cấp cho ta lớp *Camera* và *SurfaceView* để có thể truy xuất, điều khiển thiết bị camera.

Để có thể truy cập được camera của điện thoại Android trên ứng dụng, ta cần phải khai báo về quyền truy cập của ứng dụng với camera trong tập tin Android Manifest như sau:

```
<uses-permission android:name="android.permission.CAMERA" />
```

Lớp *Camera* được dùng để thiết lập các thông số chụp ảnh, cho phép bắt đầu/ngừng hoạt động của camera, chụp ảnh và đặc biệt là có thể cho phép ta truy cập hình ảnh ghi nhận được từ camera của điện thoại.

Trong ứng dụng được tìm kiếm dịch vụ này, để truy xuất, quản lý hoạt động của camera, ta xây dựng lớp *CameraPreview* kế thừa từ *SurfaceView* và sử dụng *SurfaceHolder.Callback*. Trong đó, lớp *SurfaceView* dùng để cung cấp một bề mặt để nhận dữ liệu hình ảnh thu được từ camera và hiển thị ra màn hình, sự phân cấp của các view (khung nhìn) sẽ tự động xử lý khi ta đặt các view khác lên phía trên nó. Điều này cho phép ta thể hiện các đối tượng như hình vẽ, các điều khiển lên trên màn hình camera. Bên cạnh đó, ta sử dụng *SurfaceHolder.Callback* để quản lý sự thay đổi của bề mặt (surface) thông qua các phương thức sau :


```

public void surfaceChanged(SurfaceHolder holder, int format,
int w, int h);
public void surfaceCreated(SurfaceHolder holder);
public void surfaceDestroyed(SurfaceHolder holder);

```

Trong đó, hàm callback *surfaceChanged* được gọi khi có bất cứ sự thay đổi nào trên bề mặt xảy ra, ta sử dụng hàm này để quản lý và xử lý các thay đổi đó. Hàm *surfaceDestroy* được gọi khi bề mặt bị hủy.

Để bắt đầu nhận hình ảnh từ camera của điện thoại, ta gọi phương thức *Camera.open()*, sau đó thiết lập để hình ảnh được thể hiện lên bề mặt vừa mới tạo bằng hàm *setPreviewDisplay()*. Các thao tác này được thực hiện trong hàm callback *surfaceCreated()*, khi ta tạo bề mặt.

```

public void surfaceCreated(SurfaceHolder holder) {
    camera = Camera.open();
    camera.setPreviewDisplay(holder);
}

```

Bằng cách xử lý trong hàm *surfaceChange()* ta có thể điều chỉnh các thông số cho tùy chỉnh của camera, các thông số thiết lập hiện tại của camera có thể được lấy bằng hàm *camera.getParameters()*, sau đó, tùy theo yêu cầu ứng dụng mà ta xử lý các thông số này sao cho phù hợp và truyền lại cho camera thông qua các hàm cung cấp sẵn và thường có dạng *set...()*. Dưới đây là đoạn minh họa cho việc tùy chỉnh thông số camera, trong trường hợp này, chỉ đơn giản là gán lại các giá trị chiều cao và chiều dọc cho bề mặt hiển thị của camera.

```

public void surfaceChanged(SurfaceHolder holder, int format,
int w, int h) {
    Camera.Parameters parameters = camera.getParameters();

    parameters.setPreviewSize(w, h);

    camera.setParameters(parameters);
    camera.startPreview();
}

```

Một điều quan trọng cần phải lưu ý, camera là 1 loại tài nguyên hạn chế của điện thoại, được hệ điều hành cấp phát và quản lý rất nghiêm ngặt. Nếu ta quản lý không tốt, sẽ gây ra rất nhiều phiền toái cho chương trình. Ví dụ, khi ta không ngừng và thu hồi camera trước khi *intent* (chuyển màn hình ứng dụng sang ứng dụng khác), thì ứng dụng sẽ bị lỗi và ngừng hoạt động. Để tránh những lỗi như thế này, trong ứng dụng của mình, ta phải quản lý chặt chẽ, có hệ thống các loại tài nguyên hạn chế như camera. Đoạn mã sau minh họa việc ngừng và giải phóng camera khi bề mặt bị hủy:

```
public void surfaceDestroyed(SurfaceHolder holder) {  
    camera.stopPreview();  
    camera.setPreviewCallback(null);  
    camera.release();  
    camera = null;  
}
```

Lớp *Camera* còn cung cấp các interface callback khác để hỗ trợ cho người lập trình. Cụ thể:

```
Camera.AutoFocusCallback  
Camera.AutoFocusMoveCallback  
Camera.ErrorCallback  
Camera.FaceDetectionListener  
Camera.OnZoomChangeListener  
Camera.PictureCallback  
Camera.PreviewCallback  
Camera.ShutterCallback
```

AutoFocusCallback được gọi khi quá trình tự động lấy nét được hoàn tất. Lấy nét là quá trình điều chỉnh các thông số của camera sao cho hình ảnh của vật muốn chụp hay quay phim được rõ nét nhất. Những thiết bị không hỗ trợ chế độ tự động lấy nét sẽ được trả về một callback giả (tức là không có những giá trị hay thông tin hữu ích gì). Để sử dụng được interface này, ta phải có khai báo

`Android.hardware.camera.autofocus` trong phần `<uses-feature>` trong file Manifest.

AutoFocusMoveCallback được gọi khi quá trình tự động lấy nét bắt đầu hay kết thúc.

ErrorCallback được gọi khi có lỗi xảy ra với camera, phương thức public được cung cấp là *onError()*, bao gồm 2 tham số *error* và *camera* sẽ cho ta biết được mã của lỗi xảy ra và đối tượng bị lỗi.

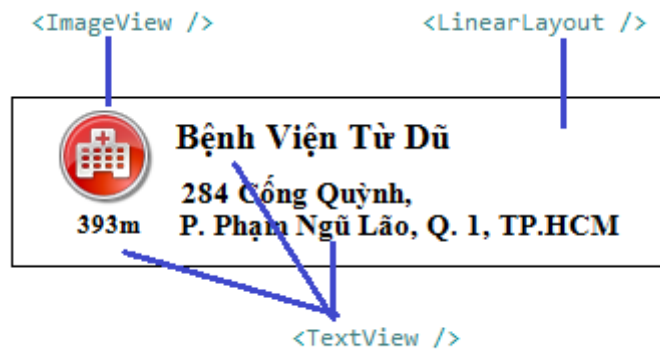
FaceDetectionListener là interface được cung cấp từ API 14, nhằm phục vụ cho việc nhận dạng khuôn mặt trên điện thoại Android. Phương thức được cung cấp là *onFaceDetection()*

OnZoomChangeListener là interface dùng để theo dõi việc thu, phóng (zoom) của camera. Phương thức cung cấp: *onZoomChange()*, được gọi khi giá trị phóng bị thay đổi.

PictureCallback được sử dụng để thu nhận dữ liệu của ảnh được chụp từ camera. Phương thức cung cấp: *onPictureTaken()*, được gọi khi đã có dữ liệu của ảnh được chụp từ camera.

2.3.2 Tùy biến ListView trong Android

ListView là một danh sách động, hiển thị các mục theo chiều dọc [5]. Để hiển thị danh sách dịch vụ trên Android, chúng ta có thể sử dụng *ListView*. Tuy nhiên, *ListView* căn bản chỉ hiển thị danh sách chuỗi ký tự. Vì vậy, ta phải tùy biến *ListView* để thể hiện đầy đủ thông tin cần thiết của dịch vụ trong mỗi mục.



Hình 2-14 Giao diện cho các mục trong **ListView**

Chúng ta cần tạo giao diện cho các mục trong **ListView** gồm các thành phần:

- `<ImageView />` (`id/imgIcon`): Ảnh đại diện cho loại dịch vụ.
- `<TextView />` (`id/tv_Distance`): Khoảng cách từ dịch vụ đến vị trí thiết bị.
- `<TextView />` (`id/tv_Name`): Tên dịch vụ
- `<TextView />` (`id/tv_Address`): Địa chỉ của dịch vụ

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10dp">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="65dip"
        android:layout_height="fill_parent">
        <ImageView
            android:id="@+id/imgIcon"
            android:layout_width="48dp"
            android:layout_height="48dp"
            android:gravity="center_vertical" />
        <TextView
            android:id="@+id/tv_Distance"
            android:layout_width="65dp"
            android:layout_height="20dp" />
    </LinearLayout>
</LinearLayout>
```

```

<LinearLayout
    android:orientation="vertical"
    android:layout_weight="1"
    android:layout_height="fill_parent">
    <TextView
        android:id="@+id/tv_Name"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <TextView
        android:id="@+id/tv_Address"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:singleLine="false"
        android:lines="2" />
    </LinearLayout>
</LinearLayout>

```

Trong Layout chính, ta khai báo thêm *ListView*

```

<ListView
    android:id="@+id/listView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"/>

```

Khi sử dụng *ListView* với bố cục mặc định thì chúng ta dùng *ArrayAdapter* để kết nối DataSource với *ListView*. Nhưng khi muốn tùy biến lại bố cục cho *ListView* thì chúng ta cần tạo một lớp mới kế thừa từ lớp *ArrayAdapter* và viết chồng phương thức *getView()*. Sau đây là lớp *ItemAdapter* kế thừa từ lớp *ArrayAdapter*

```

public class ItemAdapter extends ArrayAdapter<ServiceDTO> {
    Context context;
    int layoutResourceId;
    ArrayList<ServiceDTO> data = null;
    double curLat;
    double curLon;

    public ItemAdapter(Context context, int textViewResourceId,
        ArrayList<ServiceDTO> objects) {
        super(context, textViewResourceId, objects);
        this.context = context;
        this.layoutResourceId = textViewResourceId;
        this.data = objects;
    }
    /**
     * position: là vị trí của dịch vụ trong list
     * convertView: dùng để lấy về các control của mỗi item
     * parent: chính là datasource được truyền vào từ MainActivity

```

```

*/
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View row = convertView;
    ItemHolder holder = null; // holder giữ các widget của ListView
    if(row == null){
        LayoutInflater inflater = ((Activity)context).getLayoutInflater();
        row = inflater.inflate(layoutResourceId, parent, false);
        holder = new ItemHolder();
        holder.imgIcon = (ImageView)row.findViewById(R.id.imgIcon);
        holder.txtname = (TextView)row.findViewById(R.id.tv_Name);
        holder.txtAddress = (TextView)row.findViewById(R.id.tv_Address);
        holder.txtDistance = (TextView)row.findViewById(R.id.tv_Distance);
        row.setTag(holder);
    }
    else
        holder = (ItemHolder)row.getTag();
    //Cập nhật giá trị cho các widget của ListView
    ServiceDTO service = data.get(position);
    holder.txtname.setText(service.getName());
    holder.txtAddress.setText(service.getAddress());
    holder.txtAddress.setText(subItems.get(i).details);
    //Lớp Location cung cấp hàm distanceTo() để tính khoảng cách 2 điểm
    Location start = new Location("LocationStart");
    Location dest = new Location("LocationDest");
    start.setLatitude(curLat);
    start.setLongitude(curLon);
    dest.setLatitude(service.getLatitude());
    dest.setLongitude(service.getLongitude());
    holder.txtDistance.setText(String.valueOf(start.distanceTo(dest))+"m");
    holder.imgIcon.setImageResource(MyDictionary.dictionary.get(service.
getCategoryId(), 0)); //MyDictionary là lớp tính, key là categoryID của
//dịch vụ, value là ResourceID của hình ảnh.

    return row;
}
//Lớp ItemHolder chứa thông tin sẽ hiển thị lên ListView
static class ItemHolder{
    ImageView imgIcon;
    TextView txtname;
    TextView txtAddress;
    TextView txtDistance;
}
}

```

Chúng ta đã thiết kế giao diện cho *ListView*, xây dựng *ItemAdapter* để kết nối *DataSource*. Cuối cùng, chúng ta cần tìm *listView* trong giao diện chính, cập nhật *Adapter* cho *listView*.

```

ArrayList<ServiceDTO> service_data = new ArrayList<ServiceDTO>();
ItemAdapter itemAdapter;

```

```

itemAdapter = new ItemAdapter(this, R.layout.listView, service_data);
ListView lvItemCamera = (ListView) findViewById(R.id.listViewItemCamera);
lvItemCamera.setAdapter(itemAdapter);
lvItemCamera.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        // Xử lý sự kiện chạm vào một mục của ListView
    }
});

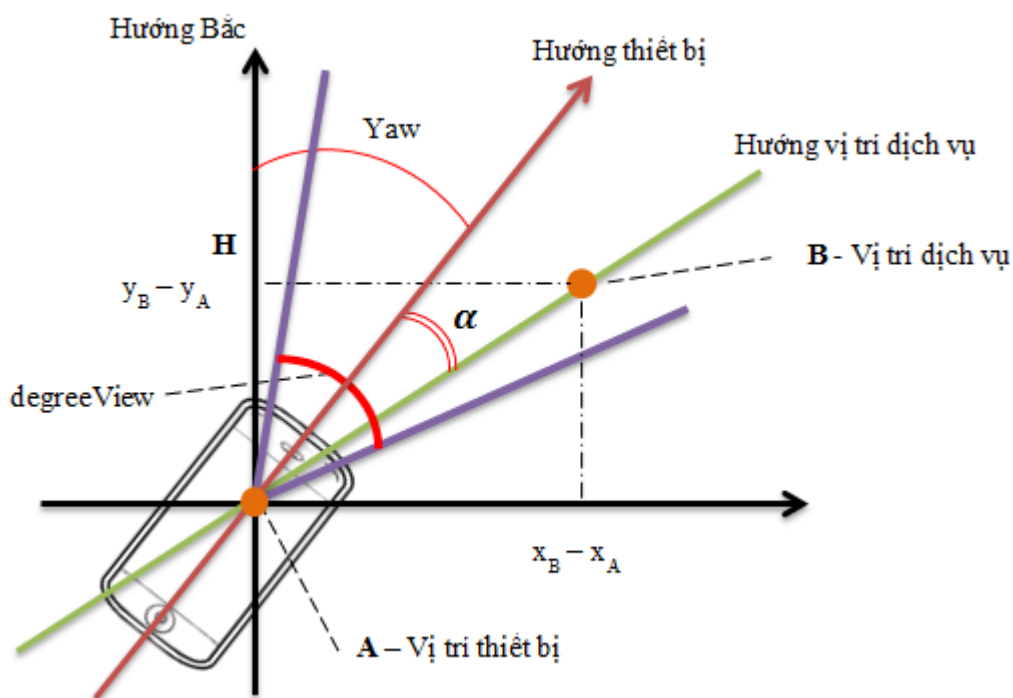
```

2.3.3 Xác định danh sách dịch vụ thuộc khung nhìn camera

Để xác định danh sách dịch vụ có thuộc khung nhìn camera hay không, ta cần tính toán vị trí của mỗi dịch vụ có thuộc phạm vi quan sát của người dùng hay không. Muốn thực hiện việc này, chúng ta sẽ sử dụng đến các yếu tố: giá trị góc quay *Yaw*, vị trí thiết bị, vị trí của dịch vụ và góc nhìn (*degreeView*).

Yaw là góc quay của thiết bị quanh trục thẳng đứng. *Yaw* có giá trị từ trong khoảng $(-180; 180)$. $Yaw = 0^\circ$ tương ứng với hướng Bắc, $Yaw = 90^\circ$ tương ứng với hướng Đông, $Yaw = +/-180^\circ$ tương ứng với hướng Nam, $Yaw = -90^\circ$ tương ứng với hướng Tây.

Để kết luận vị trí dịch vụ có thuộc phạm vi quan sát hay không ta dựng hình mô phỏng vị trí thiết bị và vị trí người dùng trong hệ tọa độ Descartes:



Hình 2-15 Hình ảnh mô phỏng vị trí của thiết bị và vị trí của dịch vụ trong hệ tọa độ Descartes.

Trong đó:

- A: vị trí của thiết bị.
- B: vị trí của dịch vụ.
- α : giá trị góc tạo bởi hướng của thiết bị và hướng vị trí của dịch vụ.
- Yaw: giá trị góc tạo bởi hướng của thiết bị và hướng Bắc.
- degreeView: góc nhìn, phạm vi quan sát của thiết bị.

Vị trí của dịch vụ nằm trong phạm vi quan sát khi $\alpha < \text{degreeView}/2$, với degreeView là yếu tố đầu vào được định trước. Nhìn hình trên **Error! Reference source not found.**, ta dễ dàng nhận thấy:

$$\widehat{HAB} = \alpha + \text{Yaw}$$

Yaw là yếu tố đầu vào được định trước. Tam giác HAB vuông tại H nên ta có thể tính được \widehat{HAB} dựa vào 2 cạnh góc vuông:

$$\tan \widehat{HAB} = \frac{HB}{HA}$$

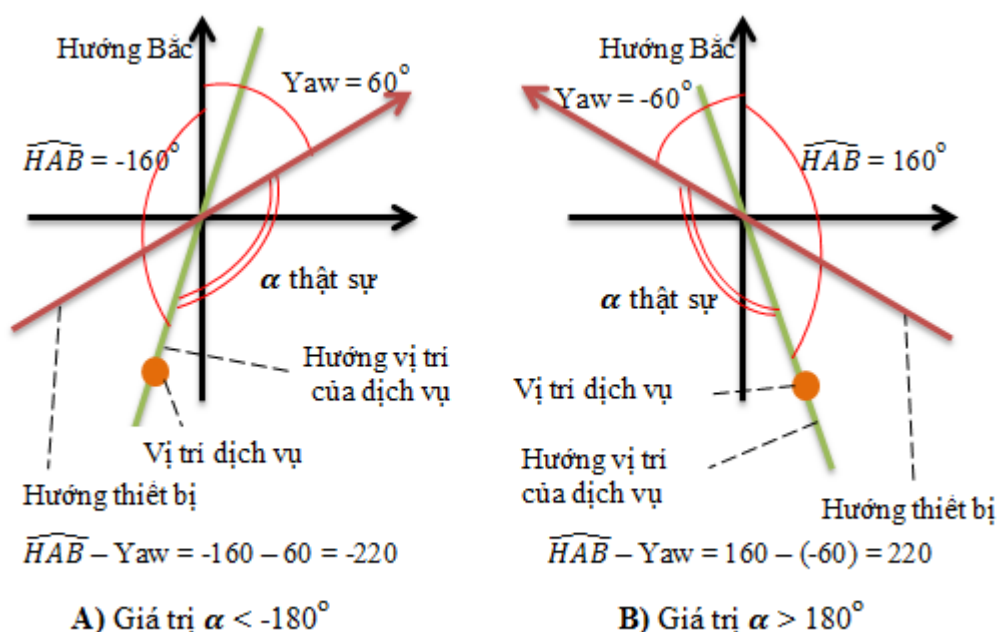
Áp dụng hàm $arctan$ để tính góc \widehat{HAB} . Tuy nhiên, hàm $arctan$ chỉ có miền giá trị trong khoảng $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. Nên ta cần sử dụng hàm $arctan2$ cho kết quả góc có miền giá trị trong khoảng $(-\pi, \pi]$

$$\widehat{HAB} = arctan2\left(\frac{x_B - x_A}{y_B - y_A}\right)$$

Trong đó hàm $arctan2$ được tính như sau

$$atan2(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right), & x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi, & y \geq 0, x < 0 \\ \arctan\left(\frac{y}{x}\right) - \pi, & y < 0, x < 0 \\ +\frac{\pi}{2}, & y > 0, x = 0 \\ -\frac{\pi}{2}, & y < 0, x = 0 \\ undefined, & y = 0, x = 0 \end{cases}$$

Chúng ta đã tính toán được \widehat{HAB} và có giá trị góc Yaw . Ta dễ tính được $\alpha = \widehat{HAB} - Yaw$. Giá trị \widehat{HAB} và Yaw có miền giá trị trong khoảng $(-\pi, \pi]$ nên $\widehat{HAB} - Yaw$ có miền giá trị trong khoảng $(-2\pi, 2\pi]$. Ta phải chuyển góc α về miền giá trị $(-\pi, \pi]$:



Hình 2-16 Giá trị α vượt quá miền giá trị $(-\pi; \pi]$

Theo hình A) ta có, $|\widehat{HAB} - Yaw| + \alpha_{\text{thật sự}} = 360^\circ$. Vị trí dịch vụ ở phía bên phải so với hướng của thiết bị nên $\alpha_{\text{thật sự}}$ có giá trị dương.

$$\alpha = 360^\circ - |\widehat{HAB} - Yaw|$$

Theo hình B) ta có, $|\widehat{HAB} - Yaw| + \alpha_{\text{thật sự}} = 360^\circ$. Vị trí dịch vụ ở phía bên trái so với hướng của thiết bị nên $\alpha_{\text{thật sự}}$ có giá trị âm.

$$\alpha = -(360^\circ - |\widehat{HAB} - Yaw|)$$

Giá trị α tính được có miền giá trị trong khoảng $(-\pi, \pi]$. α mang giá trị **dương** tương ứng vị trí của dịch vụ ở phía bên phải so với hướng của thiết bị. α mang giá trị **âm** tương ứng vị trí của dịch vụ ở phía bên trái so với hướng của thiết bị.

Áp dụng vào chương trình trên Android, ta viết đoạn mã như sau:

/**

* Hàm trả về góc giữa hướng của 1 dịch vụ và hướng của thiết bị

```

* @param longitude: Kinh độ của thiết bị
* @param latitude: Vĩ độ của thiết bị
* @param service: biến kiểu ServiceDTO chứa thông tin tọa độ của dịch vụ
* @param degreeNorth: là giá trị Yaw - góc quay quanh trục z
* @return trả về giá trị góc trong khoảng (-pi, pi]
*/
public double getDegreeOrientationDeviceWithService(double longitude, double
latitude, ServiceDTO service, float degreeNorth){
    double x = service.getLongitude() - longitude;
    double y = service.getLatitude() - latitude;
    double temp = Math.atan2(x, y)/Math.PI*180;
    double degree;
    if (temp - degreeNorth > 180)
        degree = (temp - degreeNorth) - 360;
    else if (temp - degreeNorth < -180)
        degree = 360 - (-(temp - degreeNorth));
    else
        degree = temp - degreeNorth;
    return degree;
}

```

Giá trị góc α nhận được có giá trị âm/ dương sẽ giúp ích khi vẽ biểu tượng cho dịch vụ trên màn hình. Nếu α mang giá trị dương thì biểu tượng sẽ vẽ bên phải so với trung tâm màn hình. Nếu α mang giá trị âm thì biểu tượng sẽ vẽ bên trái so với trung tâm màn hình.

Cuối cùng, ta kiểm tra giá trị α có nhỏ hơn $degreeView/2$ để đánh giá dịch vụ có nằm trong phạm vi quan sát hay không:

```

public boolean isServiceInView (double longitude, double latitude, ServiceDTO
service, float degreeNorth, float degreeView){
    double lon = longitude;
    double lat = latitude;
    //Vị trí thiết bị trùng với vị trí của dịch vụ
    if (service.getLongitude() == lon && service.getLatitude() == lat)
        return true;
    else
    {
        double alpha_degree = getDegreeOrientationDeviceWithService(
            longitude, latitude, service, degreeNorth);
        //Nếu giá trị tuyệt đối của góc alpha < degreeView/2
        //thì dịch vụ nằm trong phạm vi quan sát
        if (Math.abs(alpha_degree) < degreeView/2)
            return true;
    }
    return false;
}

```

2.3.4 Hiện thị Google Maps API v1 và tương tác biểu tượng

2.3.4.1 Hiện thị Google Maps API v1

Hệ điều hành Android hỗ trợ rất tốt việc sử dụng Google Maps. Trong bước tạo *project*, ta chọn *target SDK* là *Google APIs (API 8)*.

Để sử dụng Google Maps API trong ứng dụng, ta phải đăng ký key miễn phí từ trang <https://developers.google.com/maps/documentation/android/start>. Tại trang web này có hướng dẫn chi tiết cách lấy và sử dụng key của Google Maps.

Sau khi có được key, chúng ta bắt đầu tạo Activity hiện thị Google Maps [9]. Chúng ta tạo lớp mới kế thừa từ lớp *MapActivity*. Thư viện Android maps không phải là gói tiêu chuẩn trong Android package, mà là một API tùy chọn. Phải khai báo thư viện Android maps trong manifest trước khi sử dụng.

```
<uses-library android:name="com.google.android.maps" />
```

Trong quá trình thực thi, Google Maps cần tải dữ liệu để hiển thị các thông tin cần thiết nên ta cần cấp phát quyền truy cập internet cho ứng dụng.

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

Chúng ta chèn thêm *MapView* vào *layout* của chương trình

```
<com.google.android.maps.MapView
    xmlns:android="http://schemas.android.com/apk/res/Android"
    android:id="@+id/mapview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:apiKey="Your_API_key"
    android:clickable="true" />
```

Thư viện Android maps cung cấp một số lớp hỗ trợ quản lý bản đồ:

- *MapView*: lớp quản lý hiển thị bản đồ
- *Overlay*: lớp để thêm các chú thích cho bản đồ. *Overlay* cho phép ta dùng *Canvas* để vẽ nhiều tầng trước *Map View*

- MyLocationOverlay: là Overlay cho phép hiển thị biểu tượng la bàn trên màn hình.
- MapContRoller dùng để điều khiển bản đồ, thay đổi vị trí trung tâm của bản đồ và thay đổi thông số zoom.

Để sử dụng bản đồ, chúng ta cần khởi tạo MapView trong phương thức *onCreate()* và viết lại phương thức *isRouteDisplayed()*.

```
public class MyMapActivity extends MapActivity {
    private MapView mapView;
    private MapContRoller mapContRoller;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.map_layout);
        mapView = (MapView)findViewById(R.id.map_view);
        // Thay đổi vị trí trung tâm của bản đồ
        mapContRoller = mapView.getContRoller();
        mapContRoller.setCenter(new GeoPoint(0, 0));
        // Thêm điều khiển zoom để phóng to hoặc thu nhỏ bản đồ
        mapView.setBuiltInZoomControls(true);
        // Thay đổi các thông tin bản đồ vệ tinh, quan sát đường đi...
        mapView.setSatellite(true);
        mapView.setStreetView(true);
        mapView.setTraffic(true);
    }
    @Override
    protected boolean isRouteDisplayed() {
        // Phương thức trả về false nếu chương trình không hiển thị
        // hướng dẫn đường đi
        return false;
    }
}
```

2.3.4.2 Hiển thị biểu tượng trên Google Maps

Lớp MapView cung cấp phương thức *add(Overlay)* cho phép vẽ hình ảnh, chuỗi ký tự, đường thẳng... trên bản đồ. Chúng ta cần kế thừa lớp trừu tượng *ItemizedOverlay<OverlayItem>* để quản lý danh sách các Overlay. Trong lớp *MyItemizedOverlay* có danh sách *mOverlays* để quản lý các *OvelayItem*; *Drawable*

marker giữ hình ảnh sẽ được vẽ; viết lại hàm onTap() để xử lý sự kiện người dùng chạm vào biểu tượng trên bản đồ.

```
public class MyItemizedOverlay extends ItemizedOverlay<OverlayItem> {
    private ArrayList<OverlayItem> mOverlays = new ArrayList<OverlayItem>();
    private Context mContext;
    private Drawable marker=null;
    public MyItemizedOverlay(Drawable defaultMarker) {
        super(boundCenterBottom(defaultMarker));
        marker = defaultMarker;
    }
    public MyItemizedOverlay(Drawable defaultMarker, Context context) {
        super(boundCenterBottom(defaultMarker));
        mContext = context;
    }
    @Override
    public int size() {
        return mOverlays.size();
    }
    public void addOverlay(OverlayItem overlay) {
        mOverlays.add(overlay);
        populate();
    }
    public void RemoveOverlay(int index){
        mOverlays.remove(index);
        populate();
    }
    @Override
    public void draw(Canvas canvas, MapView mapView, boolean shadow) {
        super.draw(canvas, mapView, shadow);
    }
    @Override
    protected boolean onTap(int index) {
        OverlayItem item = mOverlays.get(index);
        // Xử lý thông tin của item được chạm vào
        return true;
    }
}
```

Khi cần hiển thị 1 biểu tượng lên màn hình ta chỉ cần thêm *MyItemizedOverlay* vào danh sách *Overlays* của *MapView*.

```
List<Overlay> mapOverlays = mapView.getOverlays();
Drawable drawable = this.getResources().getDrawable(R.drawable.icon_ok);
MyItemizedOverlay itemizedOverlay = new MyItemizedOverlay(drawable, this);

OverlayItem overlayitem = new OverlayItem(new GeoPoint(0, 0), "Hello",
    "Sample Overlay item");
```

```
itemizedOverlay.addOverlay(overlayitem);
mapOverlays.add(itemizedOverlay);
```

Drawable cơ bản chỉ hiển thị hình ảnh. Chúng ta có thể tạo lớp kế thừa lớp Drawable để hiển thị hình ảnh và chuỗi ký tự cho mỗi Drawable.

```
public class IconTextDrawable extends Android.graphics.drawable.Drawable {
    private String text;
    private final Paint paint;
    Bitmap bmp;
    public IconTextDrawable(Context context, int resource_id , String _text)
    {
        this.text = _text;
        // Tạo và cập nhật các thông số cho Paint để vẽ chuỗi ký tự
        this.paint = new Paint();
        paint.setColor(Color RED);
        paint.setTextSize(12f);
        paint.setAntiAlias(true);
        paint.setStyle(Paint.Style.FILL);
        paint.setTextAlign(Paint.Align.CENTER);
        // Tạo Bitmap từ resource id của hình ảnh
        bmp = BitmapFactory.decodeResource(context.getResources(),
            resource_id);
    }
    @Override
    public void draw(Canvas canvas) {
        int yoff = -1*bmp.getHeight()/2;
        int xoff = -1*bmp.getWidth()/2;
        // Vẽ hình ảnh
        canvas.drawBitmap(bmp, xoff, yoff, paint);
        // Vẽ chuỗi ký tự
        canvas.drawText(text, 0, bmp.getHeight()/2+10, paint);
    }
}
```

2.3.5 Xây dựng webserver API

2.3.5.1 Vai trò của server trong ứng dụng

Server là thành phần đóng vai trò không thể thiếu trong ứng dụng của nhóm phát triển. Trong tìm kiếm dịch vụ, server là nơi cung cấp và cập nhật dữ liệu cho client, là nơi client truy xuất khi người dùng có yêu cầu tìm trực tiếp trên cơ sở dữ liệu, về phần Chat, server là nơi lưu giữ thông tin người dùng, quan hệ bạn bè, ...

Là thành phần quan trọng trong kiến trúc client-server giúp dữ liệu được lưu trữ phân tán, thống nhất. Tạo điều kiện để quản lý tốt dữ liệu.

2.3.5.2 Ngôn ngữ, nền tảng

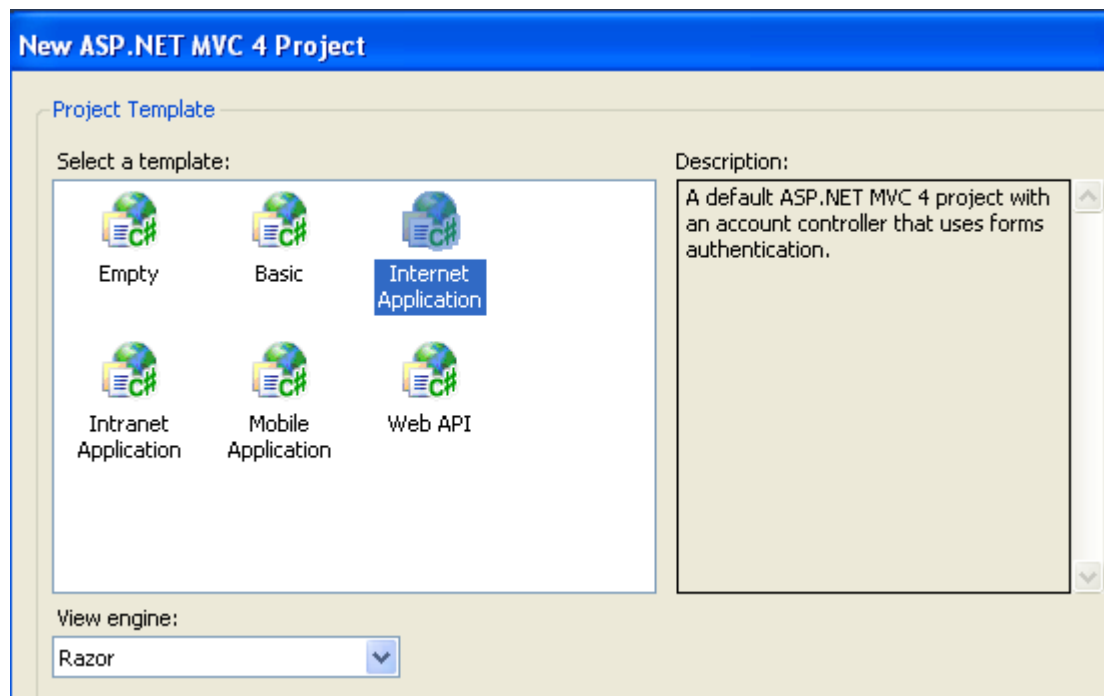
Web Services:

- + Nền tảng: ASP .NET 4.0
- + Kiến trúc : MVC Web, Web API
- + Công cụ phát triển: Visual Studio 2010

Cơ sở dữ liệu: SQL Server 2003

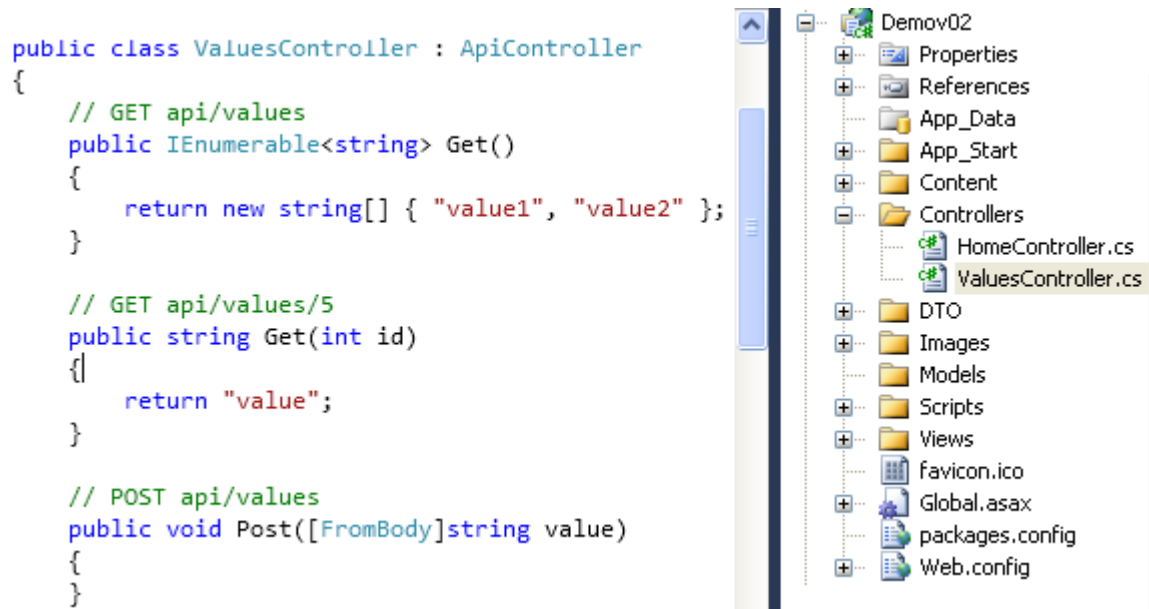
2.3.5.3 Cài đặt Web Services

Trong môi trường Visual Studio 2010 ta tạo một ASP.NET MVC 4 Web Application, sau đó chọn template là Web API, và chọn View Engine là Razor.



Hình 2-17 Tạo Web Services trên Visual Studio 2010

Ta đã tạo thành công Web Services project. Project cung cấp sẵn các thư mục và mẫu của code để xây dựng một Web API theo mô hình MVC như sau:



Hình 2-18 Mẫu Web API theo mô hình MVC

Ở đây ta sẽ làm việc trên 2 thư mục chính là **ContRollers**, **Models** và **Views**.

- **ContRollers**: Nơi chứa các lớp có nhiệm vụ điều hướng các yêu cầu người dùng, gọi các phương thức xử lý tương ứng và trả về kết quả nếu có. Cụ thể, trong webservice của nhóm xây dựng, ContRollers có nhiệm vụ nhận các request từ client, xử lý, trả về dữ liệu trong một số trường hợp như: tìm kiếm, lấy thông tin khuyến mãi,...
- **Models**: Thành phần chứa các nghiệp vụ logic, xử lý liên quan đến cơ sở dữ liệu. Ví dụ: Lấy danh sách dịch vụ, lấy các dịch vụ trong khoảng cách nhất định, lấy danh sách khuyến mãi của dịch vụ nếu có, thêm mới, xóa, sửa thông tin dịch vụ, ...
- **Views**: Là giao diện hiển thị thông tin, tương tác với người dùng. Hiện tại webservice của nhóm chưa hỗ trợ giao diện cho người sử dụng.

Sau khi đã cài đặt Web Services, ta tiến hành **publish**-tương tự như build trong phát triển ứng dụng trên PC, nhằm tạo ra các file thực thi cho Web Services. Để **publish** Web Services ta vào **Build -> Publish <tên project>**, lựa chọn hình thức publish tương ứng. Vì điều kiện tốc độ mạng yếu và sự hỗ trợ khá nghèo nàn của host miễn phí, nhóm chọn hình thức publish là “**File System**”, tức sẽ build thành các file thực thi rồi ta sẽ trực tiếp tải các file đó lên free host. Một số cách publish khác được Visual Studio hỗ trợ như:

- Web Deploy: hỗ trợ publish, tải file được build trực tiếp lên url trang web được host hỗ trợ.
- FTP: sau khi publish, các file thực thi sẽ được tải lên host thông qua FTP.

2.3.6 Truy vấn Restful Service trên Android

Ứng dụng của nhóm phát triển có áp dụng kiến trúc client-server, cho nên việc tương tác, truy xuất thông tin với server đóng vai trò hết sức quan trọng. Dù ứng dụng được xây dựng trên API 8, một API khá thấp nhưng cũng đã được hỗ trợ đầy đủ để đáp ứng nhu cầu liên lạc, tương tác với server của ứng dụng.

2.3.6.1 Thực thi một request

Interface *HttpClient* cung cấp các khuôn mẫu hàm cho các đối HTTP client. HTTP client thực hiện đóng gói thông tin của đối tượng cần thiết để thực thi các request lên server. Khái quát một số phương thức quan trọng mà interface *HttpClient* cung cấp như sau:

- *execute()*: Thực thi một request, có thể truyền thêm các tham số đầu vào cần thiết khác tùy theo mục đích và loại request.
- *getConnectionManager()*: Trả về đối tượng quản lý việc truy xuất của client hiện tại.

- *getParams()*: Lấy các thông số của HTTP client hiện tại.

Tùy theo loại request mà ta sử dụng lớp *HttpGet*, *HttpPost*, *HttpPut*, *HttpDelete* để tạo các request tương ứng. Sau đó sử dụng phương thức *execute()* của *HttpClient* tương ứng để thực thi request vừa được xây dựng.

2.3.6.2 Truyền dữ liệu trong request

Việc truyền dữ liệu trong request nói riêng và trao đổi thông tin giữa client và server nói chung được thực hiện thông qua JSON.

JSON (JavaScript Object Notation) là cú pháp để lưu trữ thông tin văn bản, được sử dụng rất rộng rãi trong việc truyền và nhận thông tin từ server hiện nay.

Trong ứng dụng mà nhóm thực hiện, để đơn giản việc sử dụng JSON, trong mỗi đối tượng dùng để lưu trữ dữ liệu (DTO) đều được xây dựng thêm phương thức *toJson()* để dễ dàng trả về đối tượng JSON tương ứng với chính nó khi cần thiết.

```
public class ObjectDTO {
    public JSONObject toJson() throws JSONException{
        JSONObject json = new JSONObject();
        //Các thao tác để truyền dữ liệu cho json
        return json;
    }
}
```

Sau đó, ta đưa dữ liệu JSON vào trong request (kiểu POST) thông qua *StringEntity* như dưới đây:

```
HttpPost post = new HttpPost(serverURL);
StringEntity se = new StringEntity(json.toString());
se.setContentType(new
BasicHeader(HTTP.CONTENT_TYPE, "application/json; charset=utf-8"));
post.setEntity(se);
```

Tốc độ truy xuất dữ liệu trên server không thể so sánh với dữ liệu trực tiếp trong máy, và tốc độ mạng không ổn định, nên ta cần đưa quá trình request trong tiến trình, để tránh làm chương trình bị đứng, chạy không ổn định.

2.3.7 Sử dụng Google Talk service để push notification

2.3.7.1 Kết nối đến server Google Talk

Ứng dụng tạo liên kết socket đến server của Google Talk thông qua lớp *XMPPConnection* trong thư viện smack. Smack là một thư viện mã nguồn mở được xây dựng cho mục đích chính là gửi gửi và nhận các tin nhắn nhanh (instant message). Chúng ta có thể tạo liên kết với server thông qua một minh họa đơn giản sau:

```
ConnectionConfiguration connConfig = new
ConnectionConfiguration(host, port, service);
XMPPConnection connection = new XMPPConnection(connConfig);
connection.connect();
connection.login(username, password);
```

Sau khi đã tạo kết nối đến server ta có thể bắt đầu gửi dữ liệu thông qua minh họa sau:

```
Message msg = new Message(recipient, Message.Type.chat);
msg.setBody(text);
connection.sendPacket(msg);
```

Để hoàn tất việc push notification ta cần thực hiện bước nữa, đó là thiết lập việc nhận dữ liệu từ server. Để tạo kết nối nhận dữ liệu như sau, ta dùng phương thức *setPacketListener*:

```
connection.setPacketListener(listener)
```

2.3.7.2 Lưu trữ thông tin người dùng

Chúng ta đã giải quyết được vấn đề push notification cho tính năng chat, tuy nhiên vẫn còn vấn đề khác như: kết bạn, lưu người dùng nếu cần thiết, lấy danh sách bạn, ... Phương pháp mà nhóm lựa chọn để đáp ứng các nhu cầu trên đó là xây dựng thêm một Web Service API nữa.

Chi tiết hơn về cấu trúc cơ sở dữ liệu đã được trình bày trong phần 2.2.5.2 và việc cài đặt web service này đã được trình bày trong phần 2.3.5.3

2.3.8 Một số vấn đề khác

2.3.8.1 Hiện thị chỉ dẫn đường đi

Chức năng hiển thị chỉ dẫn đường đi trên bản đồ sử dụng dịch vụ Google Directions API do Google cung cấp. Để hiểu sâu hơn về cách sử dụng dịch vụ này, chúng ta có thể tham khảo thông tin tại địa chỉ <https://developers.google.com/maps/documentation/directions/>

Chúng ta thực hiện gửi gói tin request:

```
maps.googleapis.com/maps/api/directions/json?origin=lat01,lon01&destination=lat02,lon02&sensor=false
```

Trong đó:

- json: là loại gói tin trả về từ request (ngoài ra còn có thể trả về xml).
- lat01, lon01: là giá trị vĩ độ và kinh độ của điểm đầu.
- lat02, lon02: là giá trị vĩ độ và kinh độ của điểm cuối.

Sau khi gửi request đến Google Directions API, chúng ta nhận về gói tin có cú pháp json. Trong tập tin json này, chúng ta sẽ rút trích được thông tin các đoạn đường đi từ điểm đầu đến điểm cuối.

Nhóm đã xây dựng các lớp *Route*, *Segment* để lưu trữ thông tin chỉ dẫn đường đi và lớp *GoogleParser* để rút trích thông tin từ gói tin json. Lớp *Segment* lưu trữ tọa độ của đoạn đường, chuỗi ký tự chỉ dẫn đường đi và độ dài đường đi. Lớp *Route* gồm danh sách các *Segment*, danh sách tọa độ bắt đầu của các đoạn đường đi, tổng chiều dài đường đi.

Sau khi xử lý gói tin json, chúng ta có danh sách các tọa độ bắt đầu của các đoạn đường đi được lưu trữ trong lớp *Route*. Từ các điểm tọa độ đó, ta sẽ vẽ các đoạn thẳng nối hai tọa độ liên tiếp nhau. Nhóm sử dụng lớp *Path* trong gói *graphics* để vẽ đoạn thẳng trên bản đồ.

```
path.moveTo(p1.x, p1.y); // Di chuyển đến điểm P1
path.lineTo(p2.x, p2.y); // Vẽ đoạn thẳng từ P1 đến P2
```

2.3.8.2 Thực hiện cuộc gọi điện thoại

Để có thể truy cập được khả năng thực hiện cuộc gọi điện thoại trên Android, ta cần phải khai báo quyền truy cập của ứng dụng với tính năng gọi điện thoại trong tập tin Android Manifest như sau:

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

Trong mã nguồn của ứng dụng, ta tạo *Intent* với *Action* là *ACTION_CALL* và truyền số điện thoại vào *Intent* để thực hiện cuộc gọi.

```
Intent callIntent = new Intent(Intent.ACTION_CALL);
callIntent.setData(Uri.parse("tel:" + "0909123456"));
startActivity(callIntent);
```

2.3.8.3 Tự động chuyển đổi màn hình

Nhóm xây dựng tính năng tự động chuyển đổi luân phiên giữa màn hình hiển thị danh sách dịch vụ và màn hình bản đồ. Khi được kích hoạt, màn hình hiển thị danh sách dịch vụ sẽ chuyển sang màn hình bản đồ nếu góc nghiêng giữa thiết bị với mặt đất $< 10^\circ$. Màn hình bản đồ sẽ chuyển sang màn hình hiển thị danh sách dịch vụ nếu góc nghiêng giữa thiết bị với mặt đất $> 70^\circ$.

Chúng ta sử dụng giá trị góc quay *Pitch* để xác định góc nghiêng của thiết bị so với mặt đất. Khi thiết bị được đặt song song với mặt đất thì giá trị góc quay *Pitch* = 0°. Khi thiết bị được dựng đứng thì giá trị góc quay *Pitch* tăng dần đến 90° hoặc giảm dần về -90°.

Khi ở màn hình hiển thị danh sách dịch vụ, nếu giá trị góc quay *Pitch* < 10° thì *Intent* chuyển sang *GoogleMapActivity* được gọi.

```
if (Pitch < 10) {  
    Intent intent = new Intent(MainActivity.this, GoogleMapActivity.class);  
    startActivity(intent);  
}
```

Khi ở màn hình bản đồ, nếu giá trị góc quay *Pitch* > 70° thì *Intent* chuyển sang *MainActivity* được gọi.

```
if (Pitch > 70) {  
    Intent intent = new Intent(GoogleMapActivity.this, .class);  
    startActivity(intent);  
}
```

2.3.8.4 Nạp thêm dữ liệu vào *ListView* khi cuộn danh sách

Nhóm khóa luận đã thực hiện cải tiến trong hiển thị danh sách *ListView*, cho phép nạp 10 mục vào *ListView*. Sau mỗi lần cuộn danh sách xuống đáy, *ListView* sẽ nạp thêm 10 mục tiếp theo. Để thực hiện tính năng này, chúng ta xây dựng lớp *EndlessScrollListener* kế thừa từ lớp *OnScrollListener* và đặt *ListView* lắng nghe sự kiện *setOnScrollListener* theo lớp *EndlessScrollListener*.

```
ListView listView;  
EndlessScrollListener scrollListener;  
listView.setOnScrollListener(scrollListener);
```

Bên trong lớp *EndlessScrollListener*, ta cần quan tâm đến phương thức *onScroll* (sự kiện phát sinh khi cuộn danh sách trong *ListView*). Nếu người dùng cuộn danh sách xuống đáy và có dữ liệu để nạp thêm thì phương thức ảo *onLoadMoreData* được gọi.

```
@Override
```

```

public void onScroll(AbsListView view, int firstVisibleItem,
                    int visibleItemCount, int totalItemCount) {
    //totalItemCount: Số lượng item được nạp vào ListView
    //firstVisibleItem: Chỉ số của mục đầu tiên hiển thị trên màn hình
    //totalItemCount: Tổng số mục được hiển thị trên màn hình
    boolean loadMore = totalItemCount <= firstVisibleItem
                        + visibleItemCount;

    if (hasMoreDataToLoad() && loadMore) {
        onLoadMoreData();
    }
}
protected abstract boolean hasMoreDataToLoad();
protected abstract void onLoadMoreData();

```

Để sử dụng lớp *EndlessScrollListener*, chúng ta cần cài đặt các phương thức ảo *hasMoreDataToLoad* và *onLoadMoreData*. Phương thức *hasMoreDataToLoad* kiểm tra còn dữ liệu để nạp thêm vào danh sách hay không. Phương thức *onLoadMoreData* thực thi quá trình nạp dữ liệu vào *ListView*.

```

scrollListener = new EndlessScrollListener() {
    @Override
    protected void onLoadMoreData(int page) {
        // Add 10 item to list adaptor
    }
    @Override
    protected boolean hasMoreDataToLoad() {
        if (services_data.size() > services_data_loaded.size())
            return true;
        return false;
    }
};

```

2.3.8.5 Tìm kiếm dịch vụ với từ khóa tiếng Việt

Nhóm khóa luận dùng phương pháp chuyển đổi từ khóa tìm kiếm và tên các dịch vụ về chuỗi ký tự không dấu. Sau đó, nếu dịch vụ nào có tên chứa từ khóa tìm kiếm thì sẽ được nạp lên danh sách hiển thị dịch vụ.

Chúng ta xây dựng lớp *ConvertVietNamChar* cho phép chuyển đổi chuỗi ký tự tiếng Việt có dấu về chuỗi ký tự không dấu. Mảng chuỗi ký tự *VietNamChar* chứa danh sách các ký tự có dấu trong tiếng Việt.

```

String[] VietNamChar = new String[]{
    "aaeeoouuiiddy",

```



```

        "ăăăăăăăăăăăăăăăăăă",
        "ÀÀÀÀÀÀÀÀÀÀÀÀÀÀÀÀÀÀ",
        "èèèèèèèèèèèèèèèè",
        "ÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊÊ",
        "ôôôôôôôôôôôôôôôôôô",
        "ỒỒỒỒỒỒỒỒỒỒỒỒỒỒỒỒ",
        "ưưưưưưưưưưưưưưưư",
        "ƯƯƯƯƯƯƯƯƯƯƯƯƯƯƯƯ",
        "ììììì",
        "ÌÌÌÌÌ",
        "đ",
        "Đ",
        "ỳỳỳỳ",
        "ỠỠỠỠ"
    };
}

```

Chúng ta tiến hành thay thế từng ký tự có dấu tiếng Việt trong chuỗi ký tự tìm kiếm bằng ký tự không dấu. Các ký tự có nguồn gốc từ ký tự “a” kết hợp dấu tiếng Việt (tương ứng với các ký tự trong chuỗi *VietNamChar[1]*) sẽ được thay thế bằng ký tự “a” (tương ứng với ký tự trong chuỗi *VietNamChar[0].charAt(0)*).

```

for (int i = 1; i < VietNamChar.length; i++)
    for (int j = 0; j < VietNamChar[i].length(); j++)
        str = str.replace(VietNamChar[i].charAt(j),
                           VietNamChar[0].charAt(i-1));

```

Tuy nhiên, khi nhập 1 ký tự tiếng Việt có dấu vào *EditText* trong Android, ta sẽ thu được 2 ký tự trong *EditText*. Ví dụ: nhập “à” (mã Unicode là 224) vào *EditText*. Ta dùng phương thức *getText()* để lấy giá trị trong ô nhập của *EditText* và thu được chuỗi ký tự “a” (mã Unicode là 97 và 768). Nếu nhập “â” (mã Unicode là 7847) thì ta thu được chuỗi “â” (mã Unicode là 226 và 768). Vì vậy, chúng ta cần loại các ký tự dấu khỏi chuỗi ký tự.

```

for (int i = 0; i < str.length(); i++)
    if (Character.codePointAt(str, i) == 768 //Dấu huyền
        || Character.codePointAt(str, i) == 769 //Dấu sắc
        || Character.codePointAt(str, i) == 777 //Dấu hỏi
        || Character.codePointAt(str, i) == 771 //Dấu ngã
        || Character.codePointAt(str, i) == 803 //Dấu nặng)
        str = str.substring(0, i) + str.substring(i+1, str.length());

```

Sau khi chuyển chuỗi ký tự tiếng Việt về chuỗi ký tự không dấu, chúng ta lọc lấy các dịch vụ có tên chứa chuỗi ký tự tìm kiếm.

```
ArrayList<ServiceDTO> services;  
String inputString;  
//Get text from EditText...  
for (int i = 0; i < services.size(); i++) {  
    String serviceName = ConvertVietNamChar.  
        Convert(services.get(i).getName());  
    if (inputString.toLowerCase().contains(TuTimKiem.toLowerCase()))  
        //Display services.get(i) to UI  
}
```

Chương 3: ỨNG DỤNG MINH HỌA

Chương trình được xây dựng gồm nhiều tính năng, tập trung vào hỗ trợ người dùng nhanh chóng tìm kiếm, xác định hướng, đường đi đến dịch vụ.



Hình 3-1 Màn hình chính của ứng dụng

3.1 Tìm kiếm dịch vụ theo hướng

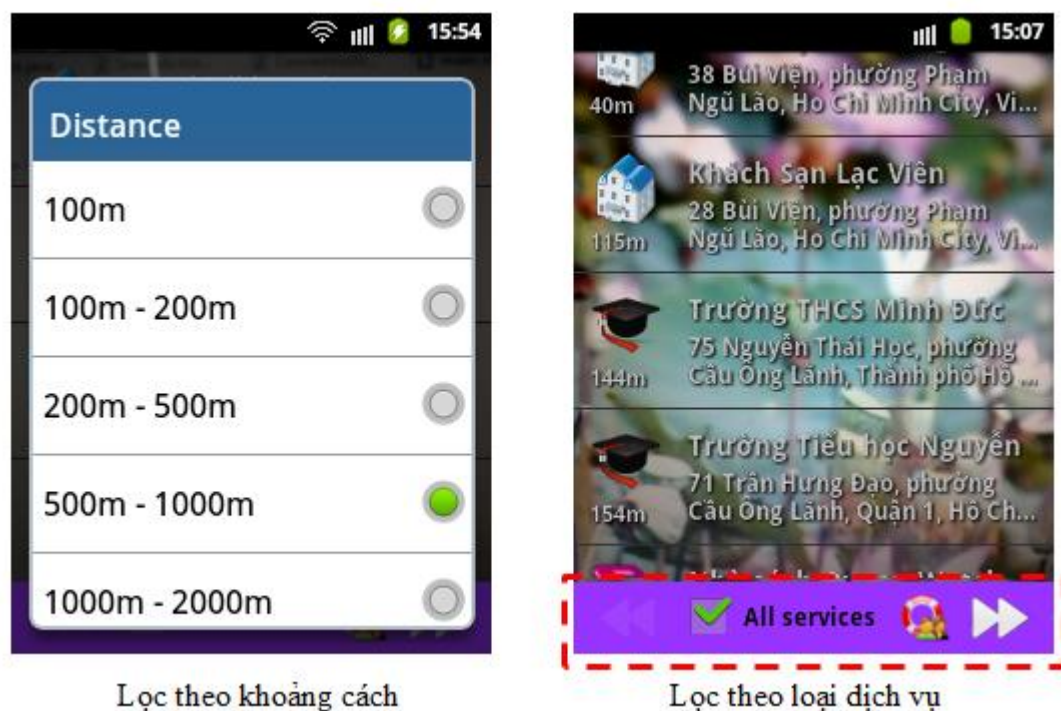
Các dịch vụ nằm trong phạm vi quan sát trực diện của người dùng sẽ được nạp vào danh sách hiển thị lên màn hình, giúp người dùng tìm dịch vụ theo phương hướng dễ dàng.

Chức năng này được kích hoạt khi người dùng chọn bật “Automatically filter services by orientation” trong màn hình *Settings*.

Tính năng này sử dụng cảm biến từ trường và cảm biến gia tốc để xác định hướng nhìn của người dùng. Từ đó, lọc ra các địa điểm có trong phạm vi quan sát của người dùng. Phạm vi quan sát được đặt giá trị cố định là $degreeView = 50^\circ$.

3.2 Lọc danh sách theo khoảng cách và theo loại dịch vụ

Người dùng có thể lọc lại danh sách dịch vụ phù hợp với nhu cầu. Ứng dụng được xây dựng để lọc danh sách theo khoảng cách hoặc theo loại dịch vụ.



Hình 3-2 Lọc danh sách theo khoảng cách và theo loại dịch vụ

Lọc danh sách theo khoảng cách gồm các mục (min - max): <100m, 100m - 200m, 200m - 500m, 500m - 1000m, 1000m - 2000m, 2000m - 5000m. Khoảng cách được tính từ vị trí của thiết bị đến vị trí của dịch vụ. Danh sách sau khi lọc chứa các dịch vụ có khoảng cách lớn hơn min và nhỏ hơn max.

Lọc danh sách theo loại dịch vụ: có 7 loại dịch vụ là Hotel, Restaurant, ATM, Hospital, Bookstore, Karaoke, School. Người dùng có thể chọn hiển thị tất cả các dịch vụ hoặc chọn một vài loại dịch vụ để hiển thị.

3.3 Tìm kiếm dịch vụ theo từ khóa

Chức năng cho phép người dùng xem danh sách dịch vụ bằng cách nhập từ khóa cần tìm.



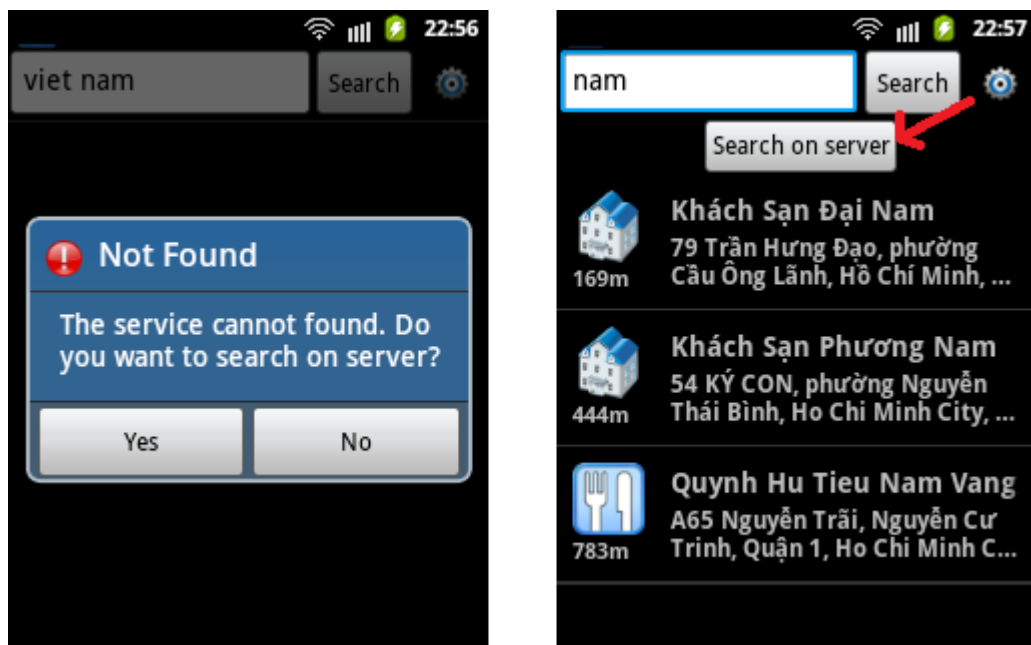
Hình 3-3 Màn hình tìm kiếm dịch vụ theo từ khóa

Giao diện của tính năng tìm kiếm dịch vụ theo từ khóa gồm các thành phần:

- Ô nhập từ khóa: người dùng nhập từ khóa có dấu hoặc không dấu vào ô tìm kiếm.
- Nút tìm kiếm: Sau khi nhập từ khóa, nhấn nút tìm kiếm để tìm dịch vụ từ cơ sở dữ liệu của thiết bị. Kết quả tìm kiếm sẽ được nạp vào danh sách dịch vụ.

- Nút cài đặt:   n/ hiện nút “Search on server”, cho phép tìm kiếm dịch vụ ngay tại server.
- Danh sách dịch vụ: danh sách chứa kết quả tìm kiếm. Danh sách ban đầu chỉ nạp 10 mục. Khi người dùng cuộn danh sách xuống cuối thì tự động nạp thêm 10 mục tiếp theo.
- Menu Map: Xem dịch vụ được chọn trong danh sách ở màn hình bản đồ.
- Menu Detail: Xem dịch vụ được chọn trong danh sách ở màn hình thông tin chi tiết.
- Menu Interact: Xem dịch vụ được chọn trong danh sách ở màn hình theo dõi hướng của dịch vụ.

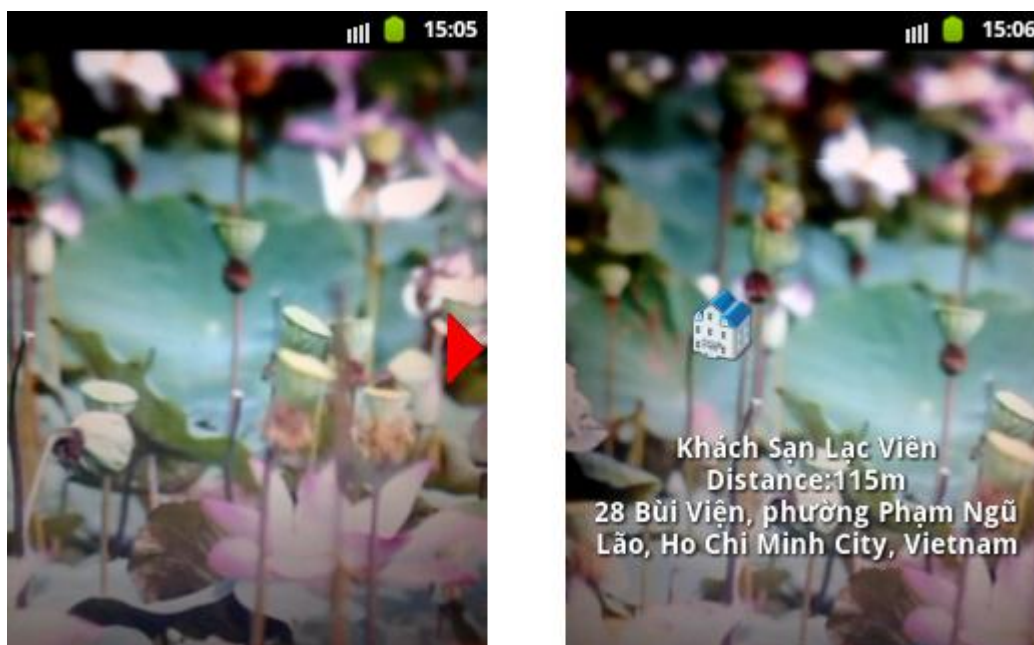
Trong trường hợp không tìm thấy bất cứ dịch vụ nào từ cơ sở dữ liệu của thiết bị, ứng dụng sẽ thông báo có tìm kiếm dữ liệu tại server hay không.



Hình 3-4 Màn hình tìm kiếm dịch vụ theo từ khóa tại server

3.4 Theo dõi hướng và khoảng cách đến dịch vụ cụ thể

Chức năng sử dụng đến cảm biến từ trường và cảm biến gia tốc, tính toán vị trí tương đối của vị trí dịch vụ cụ thể so với vị trí người dùng. Từ đó đưa ra hướng dẫn về hướng và thông tin của dịch vụ.



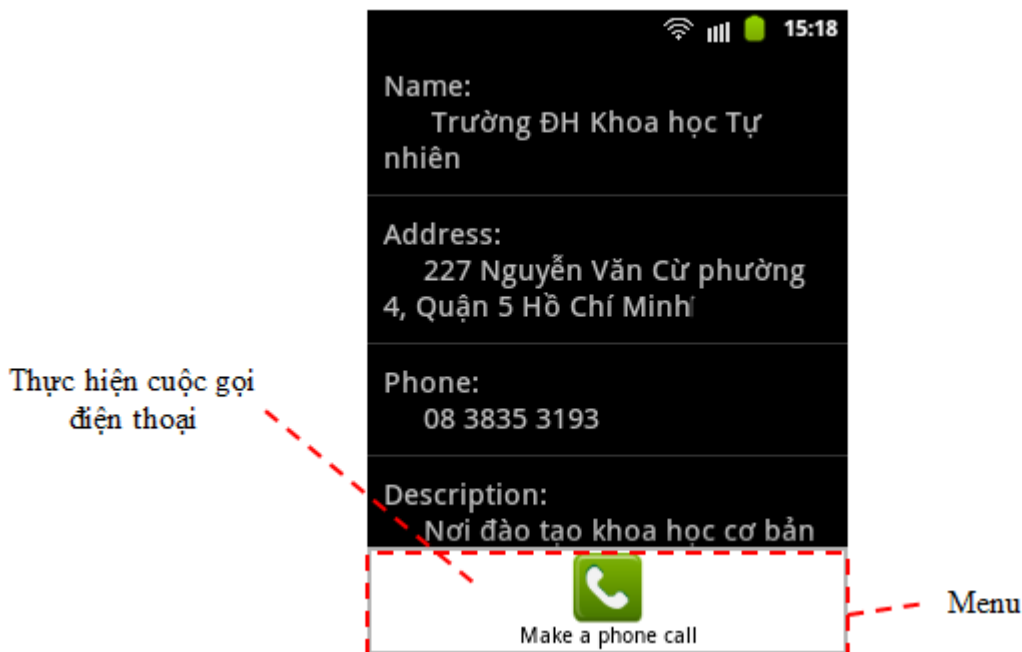
Hình 3-5 Màn hình theo dõi hướng đến dịch vụ cụ thể

Người dùng sẽ quay thiết bị để tìm vị trí của của dịch vụ cụ thể. Nếu vị trí dịch vụ không nằm trong phạm vi quan sát ($degreeView = 50^\circ$) thì màn hình hiện lên mũi tên nhấp nháy bên phải hoặc bên trái tương ứng với chỉ dẫn quay sang phải hoặc quay sang trái. Khi dịch vụ nằm trong phạm vi quan sát, màn hình sẽ hiển thị biểu tượng cho loại dịch vụ, tên dịch vụ, khoảng cách theo đường chim bay đến dịch vụ và địa chỉ của dịch vụ.

3.5 Xem thông tin chi tiết của dịch vụ

Người dùng xem được đầy đủ các thông tin của dịch vụ, bao gồm: tên dịch vụ, địa chỉ, điện thoại, mô tả. Đặc biệt hiển thị thông tin khuyến mãi nếu có.

Ngoài xem thông tin chi tiết, người dùng có thể thực hiện cuộc gọi tới nơi cung cấp dịch vụ cụ thể.



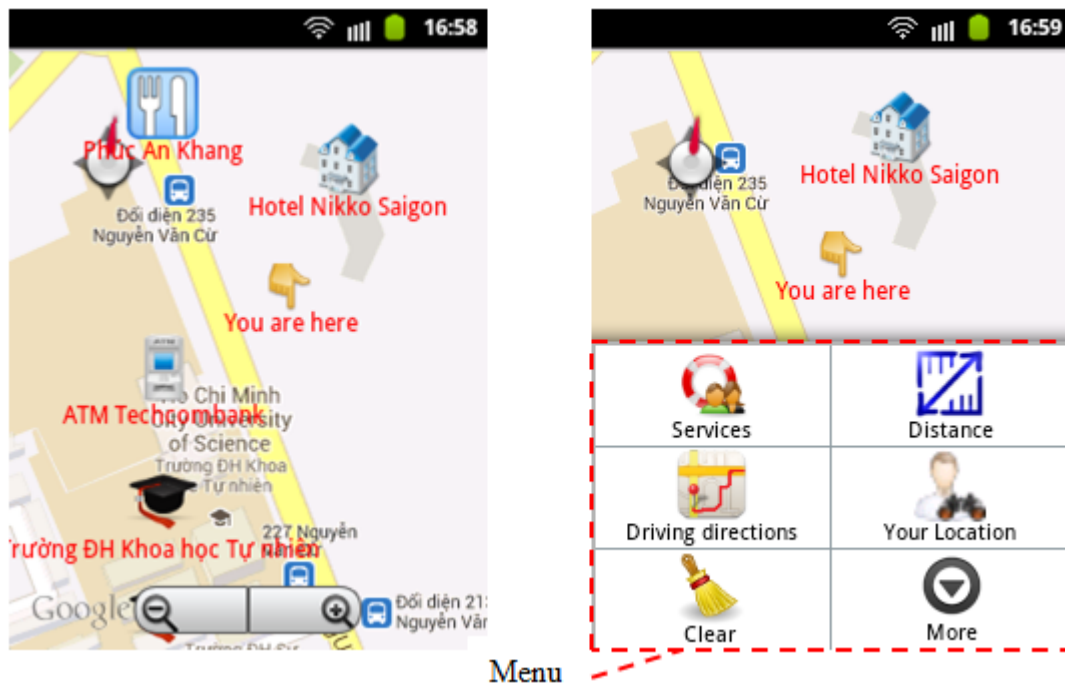
Hình 3-6 Màn hình xem thông tin chi tiết dịch vụ

3.6 **Hiển thị trực quan bản đồ Google Maps**

Ứng dụng cung cấp rất nhiều chức năng trên bản đồ Google Maps, tập trung vào hiển thị biểu tượng của dịch vụ trên bản đồ, thay đổi vị trí người dùng, tìm đường đi. Các tính năng cụ thể như sau:

- **Hiển thị biểu tượng:** các biểu tượng bao gồm biểu tượng la bàn để chỉ phương hướng (mũi tên đỏ của la bàn chỉ hướng Bắc); biểu tượng bàn tay chỉ xuống đại diện cho vị trí của người dùng; biểu tượng thể hiện loại dịch vụ (hình ảnh) và bên dưới là tên dịch vụ đi kèm.

- Services, Distance: lọc dịch vụ theo loại dịch vụ và khoảng cách. Tương tự chức năng lọc dịch vụ trên giao diện chính, tại màn hình bản đồ cho phép người dùng lọc danh sách dịch vụ sẽ hiển thị.
- Driving directions: chức năng tìm đường đi giúp người dùng nhận được chỉ dẫn đường đi từ điểm A đến điểm B.
- Your Location: chuyển vị trí trung tâm của bản đồ về vị trí của người dùng.
- Clear: Xóa các biểu tượng dịch vụ trên bản đồ.
- Change Your Location (trong mục *More*): Thay đổi vị trí của người dùng.
- Online Friend Lists (trong mục *More*): xem vị trí bạn bè trên bản đồ.



Hình 3-7 Màn hình bản đồ

3.7 Tìm đường đi

Chương trình sử dụng Google Directions API để gửi yêu cầu tính toán đường đi giữa 2 điểm. Gói tin json response trả về từ Google được xử lý để hiển thị thông tin, vẽ đường đi cho chức năng tìm đường đi.

Chức năng được kích hoạt khi người dùng chọn mục “Driving directions” trong Menu của màn hình bản đồ. Ngoài ra, khi người dùng chọn xem thông tin “Map” của một dịch vụ trong danh sách ở màn hình Search hoặc giao diện chính, tính năng tìm đường đi tự động được gọi để vẽ đường đi và hiển thị thông tin đường đi từ vị trí người dùng đến vị trí dịch vụ.

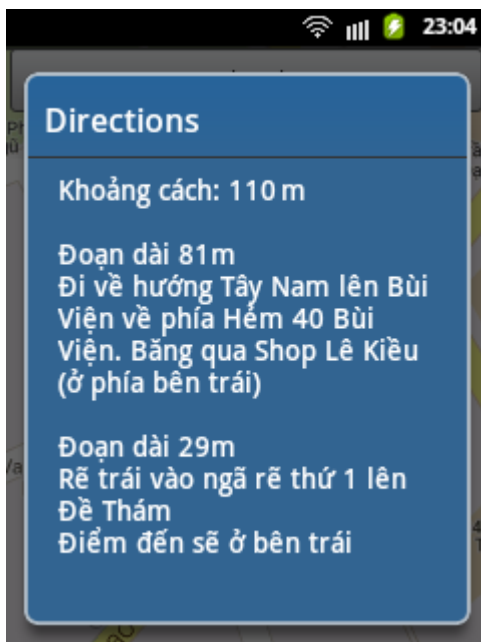


Hình 3-8 Tìm đường đi

Người dùng kéo thả hai biểu tượng hình giọt nước A và B để thay đổi vị trí bắt đầu và vị trí kết thúc. Sau khi nhấn nút “OK” và chờ trong giây lát, ứng dụng sẽ xử lý gói tin trả về từ Google để vẽ đường đi từ điểm A đến điểm B. Do việc xác định

đường đi cần gói tin json đến google nên nếu không có kết nối mạng, chức năng này sẽ không thể thực hiện và thông báo kiểm tra lại kết nối mạng sẽ hiện lên.

Nút “Get Directons” để hiển thị thông tin chỉ dẫn đường đi. Thông tin này gồm tổng khoảng cách từ điểm bắt đầu đến điểm đích và danh sách các đoạn đường chỉ dẫn.



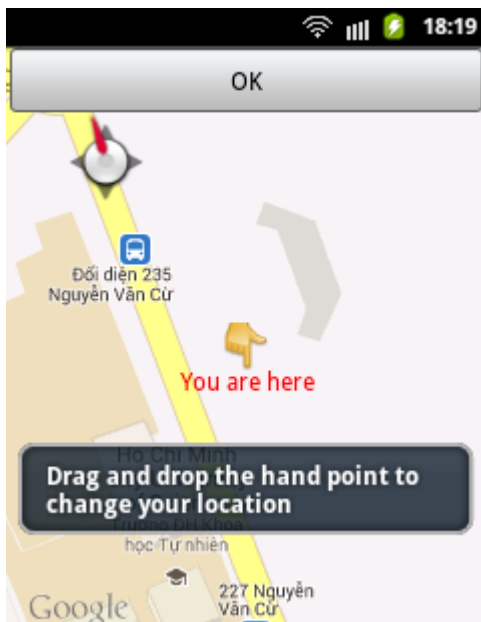
Hình 3-9 Thông tin chỉ dẫn đường đi

3.8 Thay đổi vị trí người dùng

Vị trí của người dùng được tự động xác định bằng GPS hoặc Network kết hợp Cell-ID. Nếu thiết bị có kết nối GPS thì vị trí của người dùng được xác định khá chính xác. Trong khi đó, xác định vị trí bằng Cell-ID khi kết nối 3G, đặc biệt là khi kết nối Wifi thì độ sai lệch lớn. Vì vậy, ta có nhu cầu điều chỉnh vị trí của người dùng khi cần thiết.

Chức năng được kích hoạt khi người dùng chọn mục “Change Your Location” trong Menu của màn hình bản đồ. Người dùng kéo và thả biểu tượng bàn tay chỉ

xuống để thay đổi vị trí của mình. Khi đã kéo thả biểu tượng đến vị trí thích hợp, nhấn nút “OK” để hoàn thành tính năng.



Hình 3-10 Thay đổi vị trí của người dùng

3.9 Xem vị trí bạn bè trên bản đồ

Chức năng hiển thị vị trí bạn bè trên bản đồ dạng biểu tượng hình mặt cười. Để thực hiện chức năng này người dùng phải đăng nhập tại màn hình Chat.

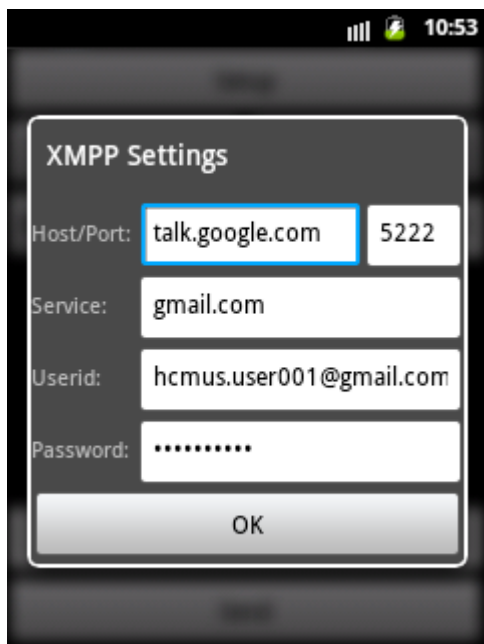


Hình 3-11 Vị trí bạn bè trên bản đồ

3.10 Kết bạn, Chat bằng tài khoản Gmail

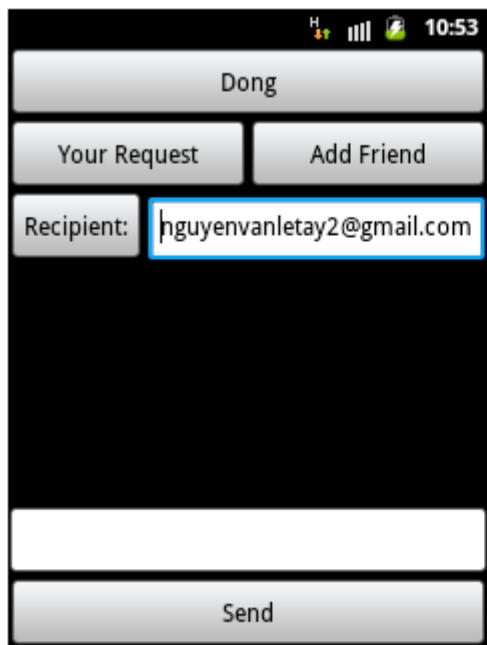
Đây là tổ hợp gồm nhiều tính năng con, cho phép người dùng đăng nhập bằng tài khoản Gmail, kết bạn, trả lời kết bạn, Chat với bạn bè.

- Đăng nhập bằng tài khoản Gmail: người dùng sử dụng tài khoản Gmail để đăng nhập tính năng Chat. Mật khẩu gửi lên server sẽ được *hash* trước khi gửi đi.

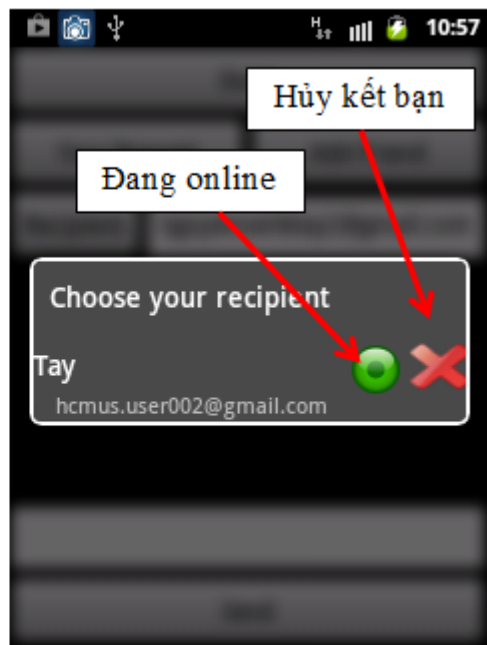


Hình 3-12 Màn hình đăng nhập chức năng Chat bằng tài khoản Gmail

- Chat với bạn bè: Người dùng gõ tin nhắn vào ô nhập tin nhắn và nhấn nút “Send” để gửi đi. Nhấn nút “Recipient” để chọn người nhận tin nhắn trong danh sách bạn bè, xem những người bạn đang online, hủy kết bạn.



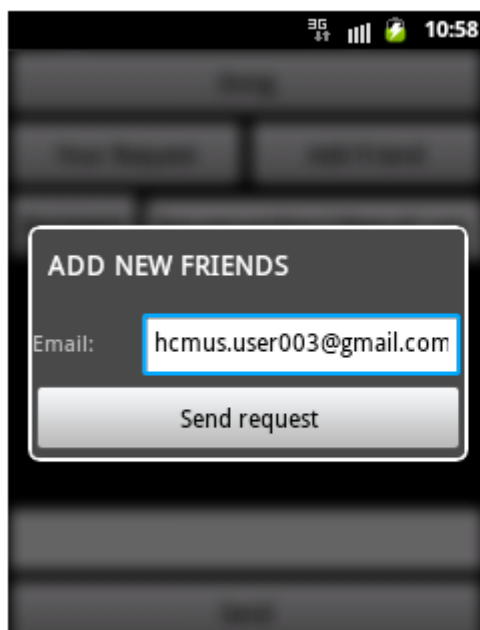
Chat với bạn bè



Chọn người nhận tin nhắn

Hình 3-13 Màn hình Chat với bạn bè

- Kết bạn và trả lời kết bạn: Người dùng nhấn nút “Add Friend” trên màn hình Chat với bạn bè để thêm bạn mới. Khi cửa sổ thêm bạn bè hiện lên, ta nhập địa chỉ Gmail của người bạn mới rồi nhấn “Send request”. Người dùng nhấn nút “Your Request” trên màn hình Chat với bạn bè để xem các yêu cầu kết bạn, chấp nhận kết bạn hoặc xóa lời mời kết bạn.



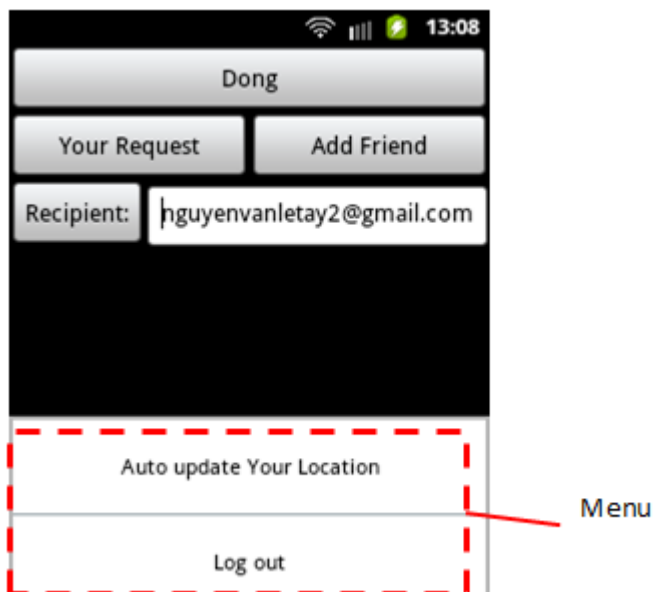
Gửi lời mời kết bạn



Xác nhận lời mời kết bạn

Hình 3-14 Màn hình kết bạn và trả lời kết bạn

Trong mục Menu của màn hình chat gồm 2 tính năng: bật/tắt cập nhật vị trí người dùng và đăng xuất tài khoản Gmail.



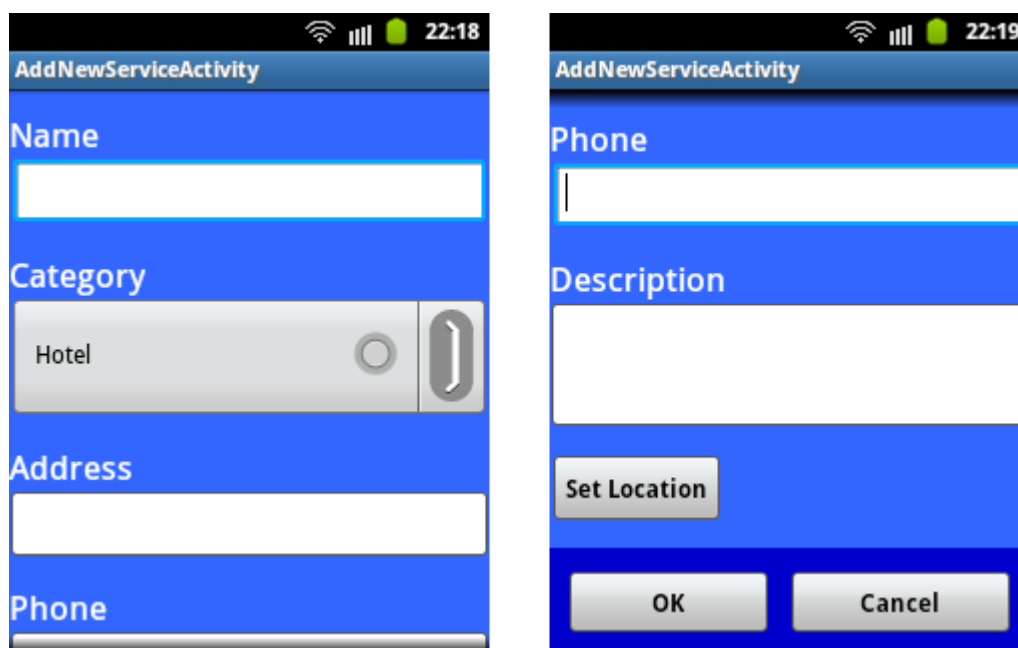
Hình 3-15 Menu của màn hình chat

Người dùng chọn mục “Auto update Your Location” để bật tính năng tự động cập nhật tọa độ của người dùng, cho phép bạn bè thấy vị trí của mình trên bản đồ. Chọn mục “Stop update Your Location” để tắt tính năng trên.

Chọn mục “Log out” để đăng xuất tài khoản Gmail.

3.11 Thêm dịch vụ mới

Khi có một dịch vụ mới chưa có trong cơ sở dữ liệu, người dùng có thể cung cấp thông tin và tạo dịch vụ mới. Dữ liệu về dịch vụ mới được truyền đến server xử lý.



The image displays two sequential screenshots of an Android application's 'AddNewServiceActivity' screen. The left screenshot, taken at 22:18, shows the initial form with fields for 'Name', 'Category' (set to 'Hotel'), 'Address', and 'Phone'. The right screenshot, taken at 22:19, shows the same form with the 'Phone' and 'Description' fields visible, and a 'Set Location' button below the 'Description' field. At the bottom of the right screenshot are 'OK' and 'Cancel' buttons.

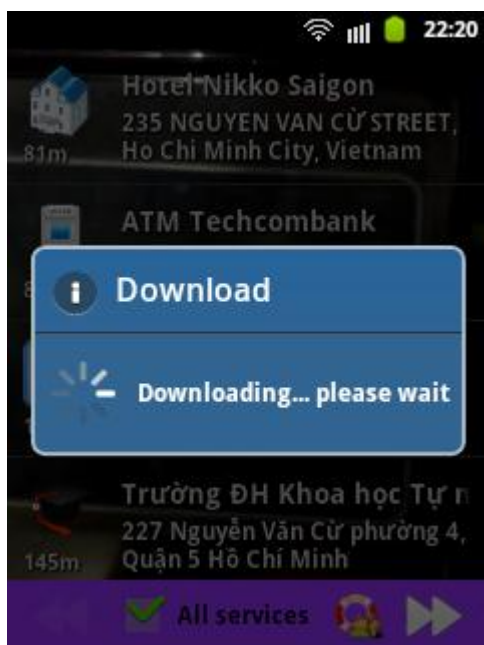
Hình 3-16 Màn hình thêm dịch vụ mới

Các thông tin người dùng cần cung cấp để thêm dịch vụ mới là: tên dịch vụ, loại dịch vụ, địa chỉ, điện thoại, mô tả. Nút “Set Location” cho phép người dùng kéo thả biểu tượng trên bản đồ để điều chỉnh vị trí của dịch vụ trên bản đồ. Nếu người dùng không nhấn nút “Set Location” thì mặc định vị trí của dịch vụ chính là vị trí của

người dùng. Cuối cùng người dùng nhấn nút “OK” để thực hiện thêm dịch vụ. Bảng thông báo xác nhận hiện lên yêu cầu người dùng chắc chắn về việc thêm dịch vụ.

3.12 Tải dữ liệu

Chức năng cho phép người dùng tải dữ liệu mới nhất về thiết bị. Ứng dụng sẽ yêu cầu lấy danh sách dịch vụ từ server. Các dịch vụ này có khoảng cách từ vị trí thiết bị đến vị trí dịch vụ nhỏ hơn khoảng cách lớn nhất để hiển thị danh sách dịch vụ trên giao diện chính. Dữ liệu tải về sẽ được lưu trong cơ sở dữ liệu SQLite của thiết bị để có thể tìm kiếm dịch vụ khi không có kết nối mạng. Người dùng phải chờ trong giây lát để việc tải dữ liệu hoàn tất.

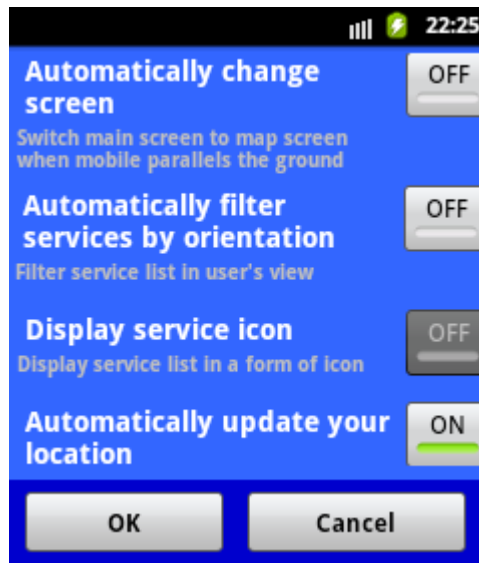


Hình 3-17 Màn hình chờ tải dữ liệu

3.13 Cài đặt hệ thống

Chức năng được kích hoạt khi người dùng chọn mục “Settings” trong Menu trên giao diện chính của ứng dụng. Người dùng tùy chỉnh các thông số hệ thống:

- Automatically change screen: Kích hoạt chế độ tự động chuyển đổi luân phiên giữa giao diện chính và màn hình bản đồ. Khi được kích hoạt, giao diện chính sẽ chuyển sang màn hình bản đồ nếu góc nghiêng giữa thiết bị với mặt đất $< 10^\circ$. Màn hình bản đồ sẽ chuyển sang giao diện chính nếu góc nghiêng giữa thiết bị với mặt đất $> 70^\circ$.
- Automatically filter services by orientation: tự động lọc danh sách dịch vụ theo hướng nhìn của người dùng. Danh sách hiển thị trên màn hình chỉ nạp các dịch vụ có vị trí nằm trong phạm vi quan sát của người dùng (*degreeView*).
- Display service icon: hiển thị các dịch vụ dạng biểu tượng trên màn hình nếu vị trí dịch vụ nằm trong phạm vi quan sát của người dùng (*degreeView*). Thông số này chỉ được chọn kích hoạt khi thông số “Automatically filter services by orientation” được kích hoạt.
- Automatically update your location: Tự động cập nhật vị trí của người dùng thông qua GPS hoặc Network kết hợp Cell-ID. Khi người dùng thực hiện chức năng “Thay đổi vị trí người dùng”, giá trị của thông số “Automatically update your location” tự động tắt về “OFF”.



Hình 3-18 Màn hình cài đặt hệ thống

3.14 Thêm dịch vụ mới thông qua Web Page

Nhóm khóa luận đã xây dựng một trang web cho phép nhập các thông tin của dịch vụ và thêm vào cơ sở dữ liệu tại máy chủ. Địa chỉ: taynguyen144.somee.com.

taynguyen144.somee.com

Map Satellite

Loại dịch vụ: Nhà hàng

Tên dịch vụ:

Địa chỉ:

Latitude: 10.763083764404735

Longitude: 106.68281539154054

Chi tiết:

Số điện thoại:

Hoàn tất

Hình 3-19 Thêm dịch vụ mới thông qua Web Page

Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

4.1.1 Một số kết quả đạt được

Trong quá trình thực hiện đề tài khóa luận, chúng em thu được một số kết quả như sau:

- Tìm hiểu thị phần các hệ điều hành, thị phần các phiên bản của hệ điều hành Android từ đó rút ra nền tảng để xây dựng ứng dụng.
- Tiếp thu nhiều kinh nghiệm lập trình điện thoại Android. Áp dụng các truy cập phần cứng cao cấp như camera, cảm biến...
- Hiểu nguyên lý, cách hoạt động của cảm biến trên điện thoại.
- Xử lý tín hiệu thu được từ gia tốc kế và cảm biến từ trường trên Android vào việc xác định phương hướng của thiết bị. Hiện thị danh sách các điểm dịch vụ trong phạm vi quan sát.
- Áp dụng thành công bộ lọc Kalman cho lọc nhiễu tín hiệu thu được từ cảm biến.
- Dựng được bản đồ Google Maps, thêm các biểu tượng tùy biến trên bản đồ. Xử lý json response từ Google để hiển thị thông tin, vẽ đường đi cho chức năng tìm đường đi.
- Xây dựng web API trên nền tảng ASP .NET 4.0, cảm nhận rõ hơn về xu hướng hiện tại – điện toán đám mây, giảm tải ở client, đưa dữ liệu lên trên server.
- Truy vấn Restful, sử dụng json trên android.
- Xây dựng khả năng Chat đăng nhập bằng Gmail.

4.1.2 Hạn chế

Bên cạnh những kết quả đạt được, nhóm vẫn còn nhiều hạn chế như:

- Cơ sở dữ liệu chưa phong phú. Ứng dụng chỉ cho phép tìm kiếm với 7 loại dịch vụ (Hotel, Restaurant, ATM, Hospital, Bookstore, Karaoke, School). Do hạn chế về nhân lực của nhóm khóa luận nên chỉ xây dựng được gần 1000 dòng dữ liệu cho bảng Service.
- Thông tin dịch vụ không có hình ảnh cụ thể của điểm dịch vụ đó. Chưa xây dựng tính năng xếp hạng, đánh giá dịch vụ từ người dùng.
- Chức năng Chat chỉ cho phép đăng ký/ đăng nhập với tài khoản Gmail.

4.2 Hướng phát triển

Với những bước tiến trong thế giới công nghệ, có thể nhận định, Augmented Reality không còn là giấc mơ xa vời mà đang từng bước được hiện thực hóa. Mặc dù những ứng dụng Augmented Reality hiện nay được tạo ra chủ yếu nhằm giới thiệu công nghệ, chưa có tính thực tiễn cao, Augmented Reality vẫn cho thấy tiềm năng phát triển to lớn của mình với khả năng tương tác cao giữa người dùng và thiết bị. Chính vì thế, chúng em đánh giá khóa luận là bước mở đầu cho việc nghiên cứu và áp dụng kỹ thuật Augmented Reality sâu hơn. Hiện tại, chúng em đã xây dựng được khả năng xác định vị trí của dịch vụ có tính tương tác cao với người dùng thông qua tính toán giá trị từ gia tốc kế và cảm biến từ trường. Theo hướng phát triển tương lai, chúng ta có thể áp dụng kỹ thuật Augmented Reality sâu hơn với việc thiết bị sẽ thu nhận hình ảnh, cung cấp thông tin của dịch vụ từ việc xử lý hình ảnh quan sát được.

Nhóm khóa luận vẫn chưa khai thác hết các Application Framework mà Android hỗ trợ. Hướng phát triển tiếp theo chúng ta áp dụng tính năng chạy dịch vụ nền, thêm notification cho chức năng Chat, tải dữ liệu của ứng dụng. Các khuyến mãi mới sẽ được thông báo về cho thiết bị, giúp người dùng cập nhật các tin khuyến mãi mới nhất. Khi chạy nền, tính năng Chat cho phép nhận tin nhắn từ bạn bè và xem vị trí của bạn bè đang online trên bản đồ.

Các lựa chọn chuyển màn hình của ứng dụng quá lạm dụng vào *Menu* của điện thoại nên chúng ta sẽ xây dựng các nút nhấn trực tiếp trên màn hình cảm biến. Ứng dụng trở nên thân thiện với người dùng.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] (2013). *Augmented reality*. Available:
http://en.wikipedia.org/wiki/Augmented_reality
- [2] T. V. Kính and L. A. Tú, "Nghiên cứu và xây dựng một số ứng dụng sử dụng cảm biến trên iPhone," Khoa Công Nghệ Thông Tin, Khoa học Tự nhiên thành phố Hồ Chí Minh, 2012.
- [3] S. Ayub, A. Bahraminisaab, and B. Honary, "A Sensor Fusion Method for Smart phone Orientation Estimation," 13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting 2012.
- [4] "Gimbal lock," 2013.
- [5] R. Meier, *Professional Android 2 Application Development*, 2 ed. Indianapolis, Indiana: Wiley Publishing, Inc, 2010.
- [6] G. Milette and A. Stroud, *Professional Android Sensor Programming*, 1 edition ed.: Wrox, June 5, 2012.
- [7] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," 1995.
- [8] E. Gamma, R. Helm, R. Johnson, and J. Vlisside, *Design patterns: elements of reusable object-oriented software*: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA 1995.
- [9] (April 9, 2013). *Google Maps Android v1 API*. Available:
<https://developers.google.com/maps/documentation/android/v1/>
- [10] (2013). *Android (operating system)*. Available:
[http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))

PHỤ LỤC

Phụ lục A: Giới thiệu về Android

Android là hệ điều hành trên điện thoại di động (và hiện nay là trên cả một số đầu phát HD, HD Player, TV) phát triển bởi Google và dựa trên nền tảng Linux. Trước đây, Android được phát triển bởi công ty liên hợp Android. Sau đó được Google mua lại vào năm 2005.

Các nhà phát triển viết ứng dụng cho Android dựa trên ngôn ngữ Java. Sự ra mắt của Android vào 05/11/2007 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. (Nguồn <http://developer.android.com>)

1. Lịch sử

Tháng 7 năm 2005, Google mua lại Android, Inc., một công ty nhỏ mới thành lập có trụ sở ở Palo Alto, California, Mỹ [10]. Những nhà đồng sáng lập của Android chuyển sang làm việc tại Google gồm có Andy Rubin (đồng sáng lập công ty Danger), Rich Miner (đồng sáng lập công ty Wildfire Communications), Nick Sears (từng là phó chủ tịch của T-Mobile), và Chris White (trưởng nhóm thiết kế và phát triển giao diện tại WebTV). Khi đó, có rất ít thông tin về các công việc của Android, ngoại trừ việc họ đang phát triển phần mềm cho điện thoại di động. Điều này tạo những tin đồn về việc Google có ý định bước vào thị trường điện thoại di động.

Tại Google, nhóm do Rubin đứng đầu đã phát triển một nền tảng thiết bị di động dựa trên hạt nhân Linux, được họ tiếp thị đến các nhà sản xuất thiết bị cầm tay và các nhà mạng trên những tiền đề về việc cung cấp một hệ thống mềm dẻo, có khả năng nâng cấp mở rộng cao. (Nguồn <http://developer.android.com>)

Mã nguồn của Android là mã nguồn mở và được cấp giấy phép Apache 2.0

2. Phát triển

Google đã liên tục xây dựng và nâng cấp hệ điều hành Android. Sau đây là danh sách các phiên bản hệ điều hành Android và thị phần của các phiên bản tính đến 03/06/2013 (Nguồn <http://developer.android.com/about/dashboards/index.html>)

| Version | Codename | API | Release date | Distribution |
|-------------|--------------------|-----|--------------|--------------|
| 1.5 | Cupcake | 3 | 30/04/2009 | 0% |
| 1.6 | Donut | 4 | 15/09/2009 | 0.1% |
| 2.1 | Eclair | 7 | 26/10/2009 | 1.5% |
| 2.2 | Froyo | 8 | 20/05/2010 | 3.2% |
| 2.3-2.3.2 | Gingerbread | 9 | 06/12/2010 | 0.1% |
| 2.3.3-2.3.7 | | 10 | 09/01/2011 | 36.4% |
| 3.2 | Honeycomb | 13 | 15/07/2011 | 0.1% |
| 4.0.3-4.0.4 | Ice Cream Sandwich | 15 | 16/12/2011 | 25.6% |
| 4.1.x | Jelly Bean | 16 | 09/07/2012 | 29.0% |
| 4.2.x | | 17 | 13/11/2012 | 4.0% |

Bảng 4-1 Thị phần các phiên bản của hệ điều hành Android

Từ khi ra đời, hệ điều hành Android ngày càng chiếm nhiều thị phần trong thị trường smartphone. Theo số liệu quý I năm 2013 của Gartner, thị phần hệ điều hành Android tiếp tục tăng trưởng và dẫn đầu với 74.4% (Nguồn <http://www.gartner.com/newsroom/id/2482816>)

**Worldwide Smartphone Sales to End Users
by Operating System in 1Q13 (Thousands of Units)**

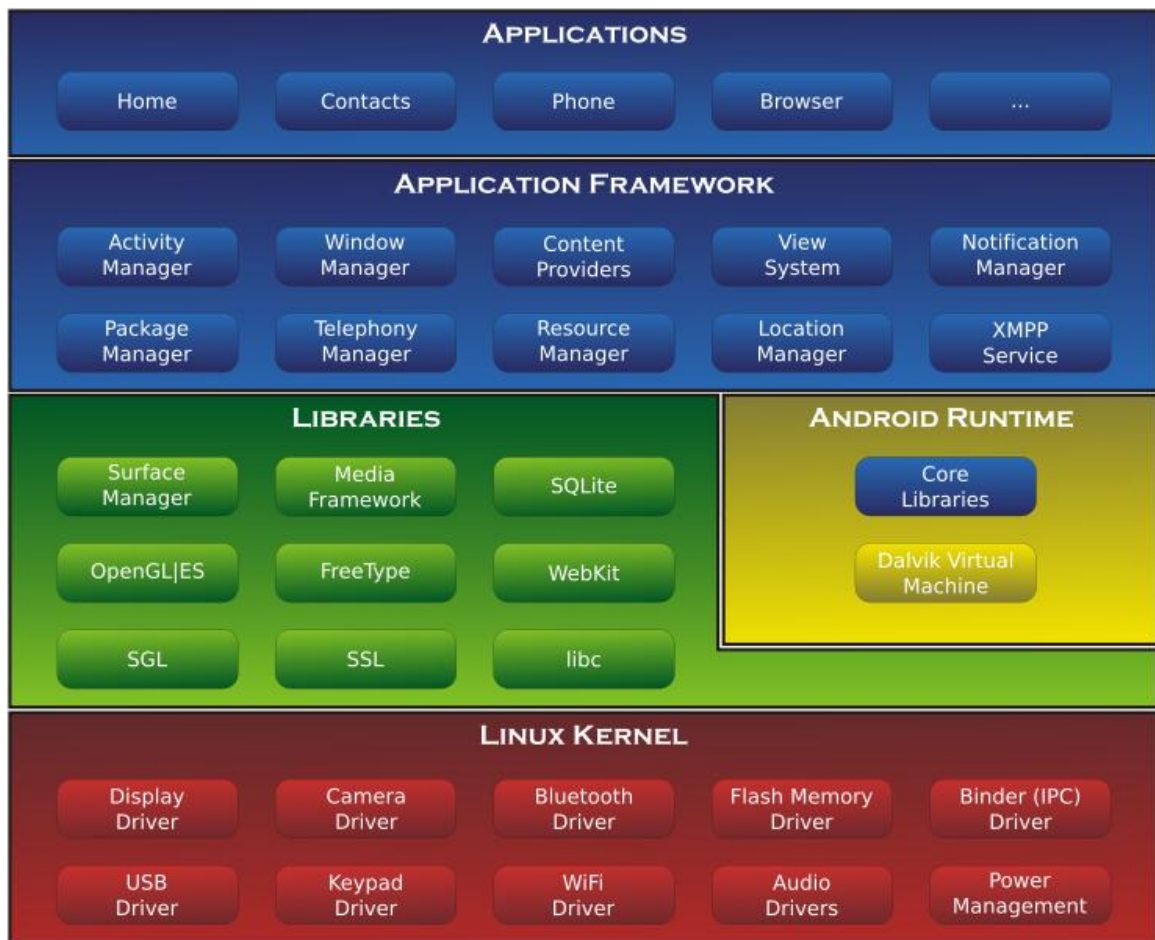
| Operating System | 1Q13 Units | 1Q13 Market Share (%) |
|-------------------------|-----------------------|----------------------------------|
| Android | 156,186.0 | 74.4 |
| iOS | 38,331.8 | 18.2 |
| BlackBerry | 6,218.6 | 3.0 |
| Microsoft | 5,989.2 | 2.9 |
| Bada | 1,370.8 | 0.7 |
| Symbian | 1,349.4 | 0.6 |
| Others | 600.3 | 0.3 |
| Total | 210,046.1 | 100.0 |

Source: Gartner (May 2013)

Hình 4-1 Thị phần hệ điều hành Smartphone trong quý I năm 2013

3. Kiến trúc và thiết kế

Các thành phần kiến trúc của hệ điều hành Android được thể hiện trong sơ đồ sau:



Hình 4-2 Sơ đồ kiến trúc hệ điều hành Android

Kiến trúc hệ điều hành Android được chia thành các tầng: Applications, Application Framework, Libraries, Android Runtime và Linux Kernel.

3.1. Tầng Applications

Hệ điều hành Android có tích hợp sẵn một số ứng dụng cần thiết cơ bản như: contactc, phone, message, camera, email... Tất cả ứng dụng ở tầng này được viết bằng ngôn ngữ Java.

3.2. Tầng Application Framework

Android cung cấp một nền tảng phát triển ứng dụng mở nên người phát triển ứng dụng có khả năng tạo ra các ứng dụng vô cùng sáng tạo và phong phú. Người phát triển ứng dụng được tự do sử dụng các tính năng cao cấp của thiết bị phần cứng như: chạy dịch vụ nền, thêm notification, truy cập thông tin cảm biến, thông tin định vị địa lý...

Người phát triển ứng dụng được phép sử dụng đầy đủ bộ API do hệ điều hành Android cung cấp. Kiến trúc ứng dụng của Android được thiết kế nhằm mục đích đơn giản hóa việc tái sử dụng các component. Qua đó, ứng dụng chia sẻ các tính năng với ứng dụng khác (VD: Ứng dụng gọi điện thoại cho phép các ứng dụng khác có thể sử dụng tính năng gọi điện thoại của nó).

Tầng này bao gồm một bộ các services và thành phần sau:

- Một tập phong phú các View có thể mở rộng được dùng để tạo giao diện như: gridview, textview, linearlayout, button...
- Content Provider: Cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác hoặc chia sẻ dữ liệu của chúng.
- Resource Manager: Cung cấp khả năng truy xuất các tài nguyên non-code như layout file, hình ảnh, menu...
- Notification Manager: cho phép hiển thị custom alert trên thanh status bar.
- Activity Manager: quản lý chu trình tồn tại của ứng dụng.

3.3. Library

Android chứa một tập hợp các thư viện C/C++. Sau đây là một số thư viện cơ bản:

- SQLite: Thư viện quản lý cơ sở dữ liệu.
- System C library: được xây dựng từ bộ thư viện hệ thống C chuẩn (libc), được điều chỉnh để tối ưu hóa cho các thiết bị chạy trên nền Linux.
- Media libraries: bộ thư viện hỗ trợ trình diễn và ghi các định dạng âm thanh và hình ảnh thông dụng.
- Surface manager: quản lý hệ thống hiển thị nội dung 2D và 3D.
- LibWebCore: Một web browser engine hiện đại được sử dụng trong trình duyệt của Android.
- SGL: Engine hỗ trợ đồ họa 2D.
- 3D libraries: Một thể hiện được xây dựng dựa trên các APIs của OpenGL ES 1.0. Những thư viện này sử dụng các tăng tốc 3D bằng phần cứng lẫn phần mềm để tối ưu hóa hiển thị 3D.
- FreeType: quản lý thể hiện bitmap và vector

3.4. Android Runtime

Android runtime bao gồm Dalvik Virtual machine và core java libraries .

Dalvik Virtual machine được phát triển từ JVM nhưng đã được tùy chỉnh để phù hợp với mobile và tránh các vấn đề tơi bản quyền. Không giống với JVM, Dalvik Virtual machine không chạy các file đuôi “.class” đặc trưng của JVM và thay vào đó nó chạy các file có đuôi “.dex”. Bản chất “.dex” được chuyển đổi từ “.class” và “.jar” tại thời điểm build chương trình. Do đó nó hiệu quả hơn nhiều byte code từ “.class”, và chúng chạy rất tốt trong các tiến trình nhỏ, sử dụng memory hiệu quả hơn.

Core java libraries được viết bằng java, nó cung cấp các function cơ bản và java, nhưng tất nhiên phải được chứng thực hoàn toàn ổn định trên mobile, như một số các function từ java SE.

3.5. Linux Kernel

Hệ điều hành Android dựa trên bộ nhân Linux 2.6 cho những dịch vụ hệ thống cốt lõi như: security, memory management, process management, network stack, driver model. Bộ nhân này chứa các driver làm nhiệm vụ như một lớp trung gian kết nối phần cứng thiết bị và phần ứng dụng.

4. Android SDK

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng bộ phát triển phần mềm Android hay còn gọi là Android SDK. Android SDK cung cấp đầy đủ công cụ cho lập trình viên để phát triển, gỡ rối, bộ giả lập điện thoại, tài liệu hướng dẫn...Bộ Android SDK bao gồm:

- Thư viện lập trình Android (Android API): đây là phần cốt lõi của bộ phát triển Android, từ các thư viện lập trình Android API, Google đã xây dựng nên các ứng dụng có sẵn.
- Công cụ phát triển: cung cấp sẵn cho các nhà phát triển các công cụ để lập trình, biên dịch, sửa lỗi mã nguồn trong ứng dụng.
- Tài liệu: đây là phần hướng dẫn sử dụng các thư viện, lớp / hàm có sẵn trong môi trường lập trình Android. Ngoài ra còn giải thích về cơ chế hoạt động của các ứng dụng trong Android.
- Ứng dụng mẫu: Bộ Android SDK còn cung cấp các đoạn mã nguồn của chương trình có sẵn.

- Trình giả lập Android: Để cung cấp sự thuận tiện cho người phát triển, bộ Android SDK đã cung cấp cho người dùng sẵn trình giả lập Android mô phỏng môi trường làm việc y như trên thiết bị thật để thuận tiện cho người phát triển có thể chạy hoặc sửa lỗi các ứng dụng trên thiết bị giả lập này mà không cần phải có thiết bị thật.

5. Môi trường lập trình cho Android

Môi trường phát triển tích hợp là một loại phần mềm máy tính có công dụng giúp đỡ các lập trình viên trong việc phát triển phần mềm.

Môi trường phát triển tích hợp (IDE) được hỗ trợ chính thức là Eclipse sử dụng phần bổ sung Android Development Tools (ADT). Tuy nhiên, người lập trình có thể sử dụng bất kỳ 1 IDE hay trình soạn thảo văn bản nào để viết code Java và XML rồi biên dịch nên ứng dụng hoàn chỉnh bằng cách sử dụng dòng lệnh (command lines).

Ứng dụng Android được đóng gói thành các file .apk và được lưu trong thư mục /data/app của hệ điều hành Android.

Một số công cụ hỗ trợ lập trình Android tiêu biểu:

- SQLite Manager: Là một addon của Firefox giúp quản lý cơ sở dữ liệu SQLite của Android.
- DroidDraw: Giúp thiết kế file XML giao diện ứng dụng.
- Balsamiq Mockups và AdobeFireworks: Giúp nhanh chóng phác thảo ý tưởng và giao diện sơ bộ của ứng dụng.
- StarUML: Vẽ các lược đồ UML hỗ trợ phân tích thiết kế.