

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**



ISO 9001:2015

ĐỒ ÁN TỐT NGHIỆP

NGÀNH : CÔNG NGHỆ THÔNG TIN

Sinh viên : Hoàng Minh Thắng

Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn

HẢI PHÒNG - 2020

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG**

**XÂY DỰNG ỨNG DỤNG ANDROID QUẢN LÝ TIN
NHẮN CÁ NHÂN ONLINE**

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

**Sinh viên : Hoàng Minh Thắng
Giảng viên hướng dẫn: ThS. Phùng Anh Tuấn**

HẢI PHÒNG - 2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên: Hoàng Minh Thắng

Mã SV: 1412401042

Lớp : CT1901C

Ngành : Công Nghệ Thông Tin

**Tên đề tài: Xây dựng ứng dụng Android quản lý tin nhắn cá nhân
online**

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

.....

.....

.....

.....

.....

.....

.....

2. Các tài liệu, số liệu cần thiết

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. Địa điểm thực tập tốt nghiệp

.....

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên :

Học hàm, học vị :

Cơ quan công tác : Trường Đại học Quản lý và Công nghệ Hải Phòng

Nội dung hướng dẫn:

Đề tài tốt nghiệp được giao ngày 30 tháng 03 năm 2020

Yêu cầu phải hoàn thành xong trước ngày 30 tháng 06 năm 2020

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Hải Phòng, ngày tháng năm 2020

HIỆU TRƯỞNG

LỜI CẢM ƠN

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

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

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

Ngoài ra em cũng xin gửi lời cảm ơn đến tất cả bạn bè, đặc biệt là các bạn trong lớp CT1901C đã luôn gắn bó, cùng học tập và giúp đỡ em trong những năm qua và trong suốt quá trình thực hiện đồ án này.

Em xin chân thành cảm ơn!

Hải Phòng, ngày tháng năm 2020

Sinh viên

Hoàng Minh Thắng

LỜI MỞ ĐẦU

Mạng điện thoại di động xuất hiện tại Việt Nam từ đầu những năm 1990 và theo thời gian số lượng các thuê bao cũng như các nhà cung cấp dịch vụ di động tại Việt Nam ngày càng tăng. Do nhu cầu trao đổi thông tin ngày càng tăng và nhu cầu sử dụng sản phẩm công nghệ cao nhiều tính năng, cấu hình cao, chất lượng tốt, kiểu dáng, mẫu mã đẹp phong phú nên các nhà cung cấp phải luôn luôn cải thiện, nâng cao những sản phẩm của mình. Trên thị trường ngày càng có nhiều sản phẩm điện thoại di động thông minh với cấu hình vượt trội cho khả năng đa nhiệm cực kỳ cao với mức giá rất cạnh tranh và tính di động cực kỳ cao khiến chúng trở thành thứ không thể thiếu đối với đại đa số mọi người trong cuộc sống hiện đại.

Đi liền với sự phát triển của các thiết bị di động thông minh kéo theo hàng loạt nhu cầu về các phần mềm có thể sử dụng được tối đa khả năng mà chúng có thể đạt được. Để đáp ứng được điều này, hàng loạt công ty, doanh nghiệp lớn trên toàn cầu đã nhảy vào cuộc đua hệ điều hành cho thiết bị di động và nổi bật nhất phải kể đến đó là Android của Google và iOS của Apple. Từ đó các phần mềm và ứng dụng trên điện thoại di động dần phát triển mạnh mẽ, phong phú và đa dạng hơn rất nhiều.

Hệ điều hành Android ra đời với sự kế thừa những ưu việt của các hệ điều hành ra đời trước và sự kết hợp của nhiều công nghệ tiên tiến nhất hiện nay. Android đã nhanh chóng là đối thủ cạnh tranh mạnh mẽ với các hệ điều hành trước đó và đang là hệ điều hành chiếm nhiều thị phần nhất trên thị trường hiện nay. Có thể nói rằng Android là hệ điều hành di động của tương lai và được nhiều người dùng ưa chuộng nhất.

Trên thiết bị di động của chúng ta đôi khi có những tin nhắn quan trọng, nếu ta để lại trên máy có thể sẽ vô tình bị xóa mất hoặc máy bị virus, khiến chúng ta mất đi những thông tin cần thiết.

Để có thể bảo vệ được những tin nhắn quan trọng này, ta cần phải đưa thông tin sang một nơi khác tách khỏi điện thoại. Một trong những phương án khả thi nhất, đó là chúng ta sẽ lưu trữ những thông tin tin nhắn theo dạng online, tức là chuyển nó lên trên Host, ở đây là chúng ta sẽ lưu trữ trực tiếp.

Như vậy, bài toán chúng ta cần giải quyết ở đây là chống mất những nội dung tin nhắn quan trọng, do việc để trên điện thoại có thể bị xóa nhầm hoặc do virus, vì thế khi có tin nhắn đến, ta sẽ lấy nó tách khỏi điện thoại và đưa lên trên Host. Đó đó đề tài của em mới mang tên là “Xây dựng ứng dụng quản lý tin nhắn online”

MỤC LỤC

| | |
|--|----|
| MỤC LỤC | 8 |
| CHƯƠNG 1: TỔNG QUAN VỀ LẬP TRÌNH DI ĐỘNG | 10 |
| 1.1. Hệ điều hành android | 10 |
| 1.1.1. Hệ điều hành android là gì? | 10 |
| 1.1.2. Giao diện hệ điều hành Android | 11 |
| 1.1.3. Ưu điểm và nhược điểm..... | 11 |
| 1.1.4. Lịch sử phát triển của hệ điều hành Android..... | 12 |
| 1.1.5. Thiết bị nào đang chạy Android?..... | 14 |
| 2.1. Môi trường phát triển ứng dụng android..... | 14 |
| 2.1.1. Phần mềm Android Studio | 14 |
| 2.1.2. Bộ công cụ phát triển Java Development Kit | 15 |
| 2.1.3. Máy ảo Android | 15 |
| 2.2. Hướng dẫn cài đặt Android Studio | 16 |
| 2.3. Các thành phần của một dự án Android trong Android Studio | 19 |
| 2.3.1. Tập cấu hình dự án AndroidManifest.xml..... | 19 |
| 2.3.2. Tập R.java | 20 |
| 2.4. Các thành phần trong giao diện Android | 21 |
| 2.4.1. View | 21 |
| 2.4.2. ViewGroup..... | 21 |
| 2.4.3. Button | 23 |
| 2.4.4. ImageButton..... | 23 |
| 2.4.5. ImageView | 23 |
| 2.4.6. ListView | 24 |
| 2.4.7. TextView | 24 |
| 2.4.8. EditText..... | 25 |
| 2.4.9. CheckBox..... | 25 |
| 2.4.10. Activity & Intent | 25 |
| 2.5. Dịch vụ web | 27 |
| 2.5.1. Khái niệm..... | 27 |
| 2.5.2. Đặc điểm | 28 |

| | |
|--|-----------|
| 2.5.3. Những chuẩn chính của Web Services | 28 |
| 2.5.4. Các dạng tương tác giữa Web Service với ứng dụng trên thiết bị di động XML - eXtensible Markup Language | 30 |
| CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG | 32 |
| 2.1. Phát biểu bài toán | 32 |
| 2.2. Phân tích thiết kế dữ liệu..... | 32 |
| 2.2.1. Phân tích dữ liệu..... | 32 |
| 2.2.2. bảng thiết kế dữ liệu..... | 32 |
| 2.2.3. Bảng cơ sở dữ liệu vật lý | 32 |
| 2.3. Phân tích thiết kế chức năng | 33 |
| 2.3.1. Sơ đồ phân rã chức năng..... | 33 |
| 1. Xử lý tin nhắn | 33 |
| 1.1. Chức năng bắt tin nhắn đến..... | 33 |
| 1.2. Tách thông tin tin nhắn | 34 |
| 1.3. Đưa thông tin lên Host..... | 35 |
| 2. Xem tin nhắn..... | 37 |
| 2.1. Xem tin nhắn theo Số Điện Thoại | 37 |
| 2.2. Xem theo ngày | 41 |
| 3. Xem đầy đủ nội dung tin nhắn..... | 45 |
| 4. Xóa tin nhắn..... | 46 |
| CHƯƠNG 3: CHƯƠNG TRÌNH THỰC NGHIỆM..... | 50 |
| 3.1. Kết quả đã đạt được..... | 50 |
| 3.2. Một Số giao diện chính | 50 |
| 3.2.1. Chức năng bắt tin nhắn và đưa lên Host | 50 |
| 3.2.2. Chức năng xem tin nhắn | 51 |
| 3.2.3. Chức năng xóa tin nhắn..... | 53 |
| 3.2.4. Chức năng xem đầy đủ tin nhắn..... | 54 |
| KẾT LUẬN | 56 |
| TÀI LIỆU THAM KHẢO | 57 |

CHƯƠNG 1: TỔNG QUAN VỀ LẬP TRÌNH DI ĐỘNG

1.1. Hệ điều hành android

1.1.1. Hệ điều hành android là gì?

Android là một hệ điều hành mã nguồn mở dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng [1].



Hình 1.1. Biểu tượng hệ điều hành Android

Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google, sau này được chính Google mua lại vào năm 2005 và hệ điều hành Android đã ra mắt vào năm 2007. Chiếc điện thoại đầu tiên chạy Android là HTC Dream được bán vào ngày 22 tháng 10 năm 2008 [1].

Chính mã nguồn mở cùng với giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới. Android chiếm 75% thị phần điện thoại thông minh trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Tháng 10 năm 2012, đã có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play (cửa hàng ứng dụng chính của Android) ước tính khoảng 25 tỷ lượt. Hiện nay con số này đã giảm xuống do sự ảnh hưởng lớn của iOS từ Apple và một phần nhỏ của Windows Phone, tuy nhiên Android vẫn dẫn đầu thị phần.

1.1.2. Giao diện hệ điều hành Android

Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo dẫn và thu lại để xử lý các đối tượng trên màn hình.



Hình 1.2. Giao diện Android 5.0

Các thiết bị Android sau khi khởi động sẽ hiển thị màn hình chính, điểm khởi đầu với các thông tin chính trên thiết bị, tương tự như khái niệm desktop trên máy tính để bàn. Màn hình chính Android thường gồm nhiều biểu tượng (icon) và tiện ích (widget). Giao diện màn hình chính của Android có thể tùy chỉnh ở mức cao, cho phép người dùng tự do sắp đặt hình dáng cũng như hành vi của thiết bị theo sở thích.

Những ứng dụng do các hãng thứ ba có trên Google Play và các kho ứng dụng khác còn cho phép người dùng thay đổi "chủ đề" của màn hình chính, thậm chí bắt chước hình dáng của hệ điều hành khác như Windows Phone hay iOS. Phần lớn những nhà sản xuất, và một số nhà mạng, thực hiện thay đổi hình dáng và hành vi của các thiết bị Android của họ để phân biệt với các hãng cạnh tranh.

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

1.1.3. Ưu điểm và nhược điểm

Ưu điểm:

- Là hệ điều hành mã nguồn mở nên khả năng tùy biến cao, có thể tùy ý chỉnh sửa mà không có sự can thiệp hay cấm cản từ Google.

- Đa dạng sản phẩm, rất nhiều hãng điện thoại, thiết bị công nghệ đã ưu ái chọn Android cho thiết bị của họ, giá cả thì hợp lý từ bình dân đến cao cấp.
- Kho ứng dụng Google Play Store đồ sộ.
- Thân thiện và dễ sử dụng.
- Khả năng đa nhiệm, chạy cùng lúc nhiều ứng dụng cao.

Nhược điểm:

- Dễ nhiễm phần mềm độc hại và virus. Do tính chất mã nguồn mở, nhiều phần mềm không được kiểm soát có chất lượng không tốt hoặc lỗi bảo mật vẫn được sử dụng.
- Kho ứng dụng quá nhiều dẫn đến khó kiểm soát chất lượng, thiếu các ứng dụng thật sự tốt.
- Sự phân mảnh lớn. Trong khi một số thiết bị Android xuất sắc đã trình làng như Galaxy S5, Galaxy Note 4, Xperia Z3..., vẫn còn rất nhiều sản phẩm giá rẻ bình thường khác.
- Cập nhật không tự động với tất cả thiết bị. Khi một phiên bản hệ điều hành mới ra mắt, không phải tất cả sản phẩm đều được cập nhật, thậm chí nếu muốn trải nghiệm bạn thường xuyên phải mua mới thiết bị.

1.1.4. Lịch sử phát triển của hệ điều hành Android

a. Sự ra đời của Android

Vào tháng 10/2003, trước khi thuật ngữ “điện thoại thông minh” được hầu hết công chúng sử dụng và vài năm trước khi Apple công bố iPhone đầu tiên và hệ điều hành iOS, công ty Android Inc được thành lập ở Palo Alto, California. Bốn người sáng lập là Rich Miner, Nick Sears, Chris White và Andy Rubin.

Năm 2005, chương lớn tiếp theo trong lịch sử của Android được thực hiện khi Google mua lại công ty gốc. Ông Andy Rubin và các thành viên sáng lập khác vẫn tiếp tục phát triển hệ điều hành dưới quyền chủ sở hữu mới của họ. Quyết định này được đưa ra để sử dụng Linux làm nền tảng cho hệ điều hành Android và điều đó cũng có nghĩa là Android sẽ được cung cấp miễn phí cho các nhà sản xuất điện thoại di động của bên thứ ba. Google và nhóm Android cảm thấy công ty có thể kiếm tiền với các dịch vụ khác sử dụng hệ điều hành, bao gồm cả ứng dụng.

Hệ điều hành Android được chính thức ra mắt từ năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mã nguồn mở. Chiếc điện thoại Android đầu tiên được bán vào năm 2008 [2].

b. Biểu tượng Android



Hình 1.3. Biểu tượng Android bên trong trụ sở chính

Hình ảnh quen thuộc hiện nay cho hệ điều hành Android giống như sự kết hợp của một con robot và một lõi màu xanh lá cây do Irina Blok tạo ra. Theo Blok, thiết kế cuối được lấy cảm hứng từ việc nhìn vào biểu tượng nhà vệ sinh quen thuộc đại diện cho “đàn ông” và “phụ nữ” [2].

c. Những phiên bản của Android

- Android 1.5 Cupcake
- Android 1.6 Donut
- Android 2.0-2.1 Éclair
- Android 2.2 Froyo
- Android 2.3 Gingerbread
- Android 3.0 Honeycomb
- Android 4.0 Ice Cream Sandwich
- Android 4.1 Jelly Bean
- Android 4.4 KitKat
- Android 5.0 Lollipop
- Android 6.0 Marshmallow
- Android 7.0 Nougat
- Android 8.0 Oreo
- Android 9.0 Pie
- Android 10 Q (thử nghiệm)

1.1.5. Thiết bị nào đang chạy Android?

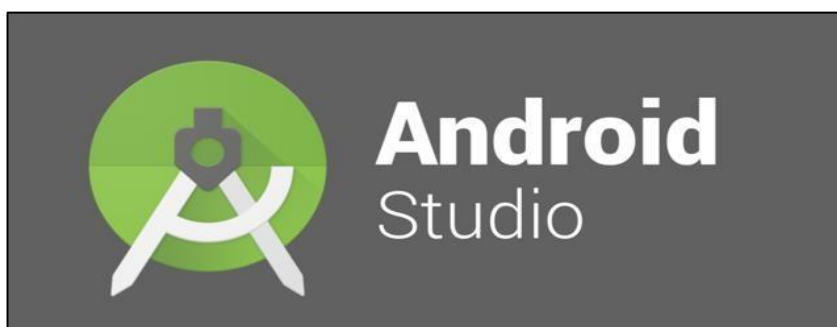
- Hiện tại Samsung vẫn đang dẫn đầu thị trường Android với nhiều thiết bị điện thoại và máy tính bảng từ bình dân đến cao cấp như: Galaxy V, Galaxy Core 2, Galaxy A3, Galaxy A5, Galaxy S5, Galaxy Note 4, Galaxy Alpha...
- Điện thoại Sony: Xperia Z3, Xperia Z3 Compact, Xperia Z2, Xpreia Z1, Xperia T2 Ultra, Xperia C3, Xperia E1...
- Điện thoại HTC: HTC Desire Eye, HTC One E8, HTC One M8, HTC One Max, HTC Desire 820S, HTC Desire 816, HTC Desire 510...
- Điện thoại Oppo: Oppo Find 7a, Oppo R5, Oppo N1 Mini, Oppo R1...
- Máy tính bảng chạy Android: Sony Xperia Z3 Tablet Compact, Samsung Galaxy Tab S 10.5 (SM-T805), Samsung Galaxy Tab S 8.4 (SM-T705), Google HTC Nexus 9 Volantis, Lenovo Yoga Tablet 2 Pro, Asus MeMo Pad 8, Dell Venue 8, Acer Iconia A1- 841, Acer Iconia B1-730
- Ngoài ra: Bản chất mở và cho phép thay đổi của Android giúp nó xuất hiện trên các thiết bị điện tử khác, như laptop và netbook, smartbook, Smart tivi và máy ảnh. Hơn thế nữa, hệ điều hành Android còn được ứng dụng trong kính mắt thông minh (Project Glass), đồng hồ đeo tay, tai nghe, máy nghe nhạc bỏ túi, điện thoại để bàn, và máy trò chơi điện tử chạy Android.

2.1. Môi trường phát triển ứng dụng android

2.1.1. Phần mềm Android Studio

Có nhiều công cụ để phát triển Android nhưng đến nay công cụ chính thức và mạnh mẽ nhất là Android Studio. Đây là IDE (Môi trường phát triển tích hợp) chính thức cho nền tảng Android, được phát triển bởi Google và được sử dụng để tạo phần lớn các ứng dụng mà chúng ta có thể sử dụng hàng ngày.

Android Studio lần đầu tiên được công bố tại hội nghị Google I/O vào năm 2013 và được phát hành cho công chúng vào năm 2014 sau nhiều phiên bản beta khác nhau. Trước khi được phát hành, các nhà phát triển Android thường sử dụng các công cụ như Eclipse IDE, một IDE Java chung cũng hỗ trợ nhiều ngôn ngữ lập trình khác [3].



Hình 2.1. Biểu tượng Android Studio

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

Hãy coi đây là đuôi cho code Java cho phép nó chạy trơn tru trên các thiết bị Android và tận dụng lợi thế của phần cứng gốc. Chúng ta cần sử dụng ngôn ngữ lập trình Java để viết các chương trình, Android SDK có nhiệm vụ kết nối các phần này lại với nhau. Cùng lúc đó Android Studio kích hoạt để chạy code, thông qua trình giả lập hoặc qua một phần cứng kết nối với thiết bị. Sau đó, chúng ta cũng có thể “gỡ rối” chương trình khi nó chạy và nhận phản hồi giải thích sự cố, v.v... để chúng ta có thể nhanh chóng giải quyết vấn đề.

Google đã nỗ lực rất nhiều để làm cho Android Studio trở nên mạnh mẽ và hữu ích nhất có thể. Nó cung cấp những gợi ý trực tiếp trong khi viết code và thường đề xuất những thay đổi cần thiết để sửa lỗi hoặc làm code hiệu quả hơn. Ví dụ, nếu không sử dụng biến, biến đó sẽ được tô đậm bằng màu xám. Và khi bắt đầu gõ một dòng code, Android Studio sẽ cung cấp danh sách gợi ý tự hoàn thành để giúp bạn hoàn thiện dòng code đó. Chức năng này rất hữu ích khi chúng ta không nhớ được chính xác cú pháp hoặc để tiết kiệm thời gian.

2.1.2. Bộ công cụ phát triển Java Development Kit

Java Development Kit (JDK - Bộ công cụ cho người phát triển ứng dụng bằng ngôn ngữ lập trình Java) là một tập hợp những công cụ phần mềm được phát triển bởi Sun Microsystems dành cho các nhà phát triển phần mềm, dùng để viết những applet Java hay những ứng dụng Java - bộ công cụ này được phát hành miễn phí gồm có trình biên dịch, trình thông dịch, trình giúp sửa lỗi debugger, trình chạy applet và tài liệu nghiên cứu.

2.1.3. Máy ảo Android

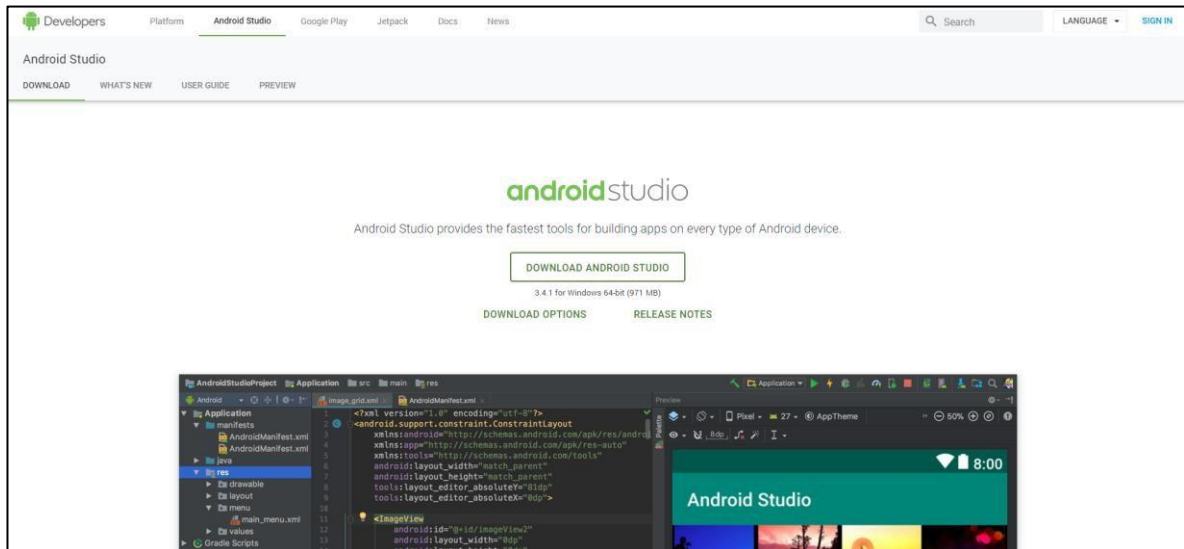
Máy ảo Android là một phần không thể thiếu khi chúng ta lập trình ứng dụng cho hệ điều hành Android, nó giúp chúng ta chạy thử ứng dụng ngay trên máy tính. Trong Android Studio có cung cấp cho chúng ta một máy ảo Android mặc định là Android Virtual Device, viết tắt là AVD.

AVD là một máy ảo Android được hỗ trợ chính thức từ Google. Vì là bản “chính chủ” nên máy ảo này sẽ có tính ổn định cao. Chẳng hạn như nó sẽ tiêu

tồn bộ nhớ của máy tính ít hơn các máy ảo khác, nó còn hỗ trợ giả lập tất cả các loại thiết bị, từ điện thoại, máy tính bảng, đồng hồ thông minh hay Android TV.

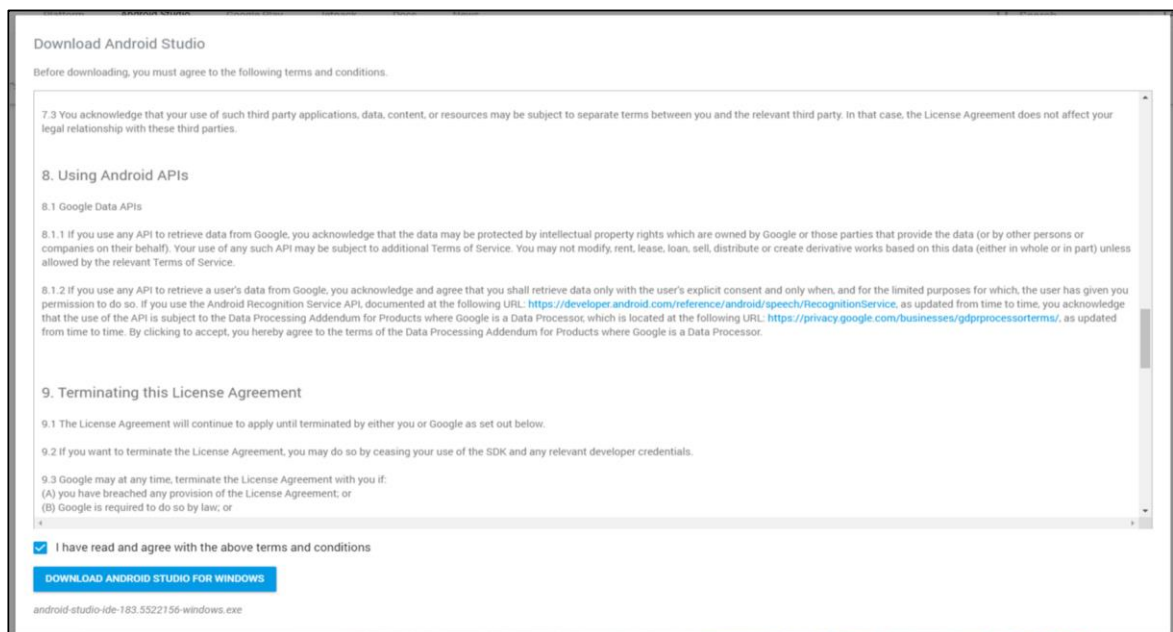
2.2. Hướng dẫn cài đặt Android Studio

Bước 1: Truy cập trang web <https://developer.android.com/studio> và nhấn nút **Download Android Studio**



Hình 2.2. Giao diện trang web tải bộ cài đặt Android Studio

Bước 2: Tick vào ô “I have read and agree with the above terms and conditions” và nhấn nút **Download Android Studio for Windows** nếu có thông báo xuất hiện

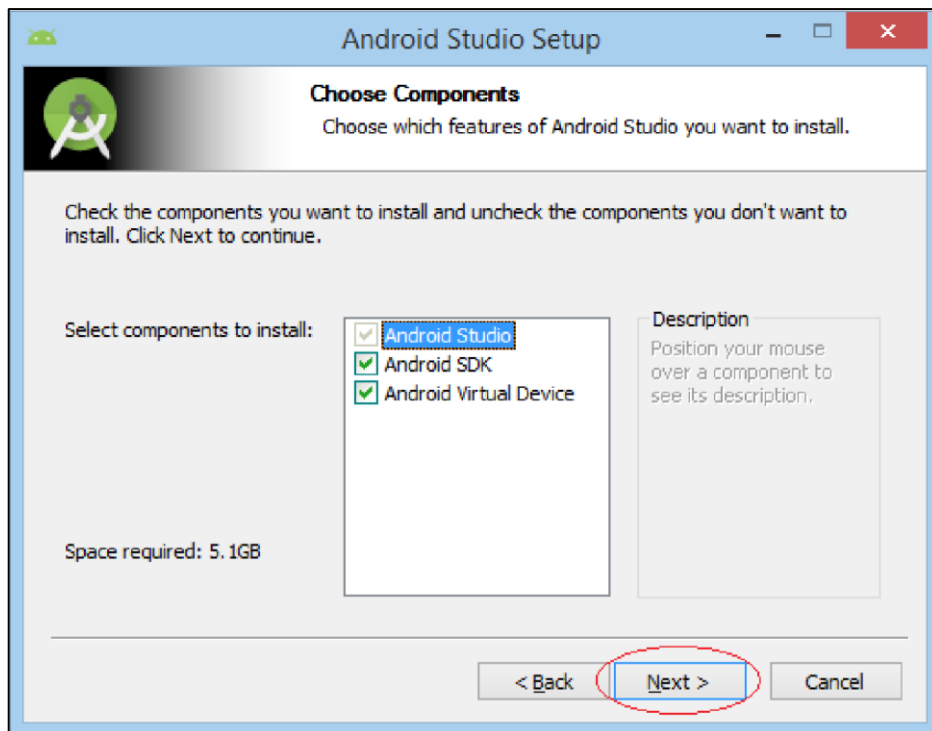


Hình 2.3. Xác nhận điều khoản sử dụng để có thể tải về

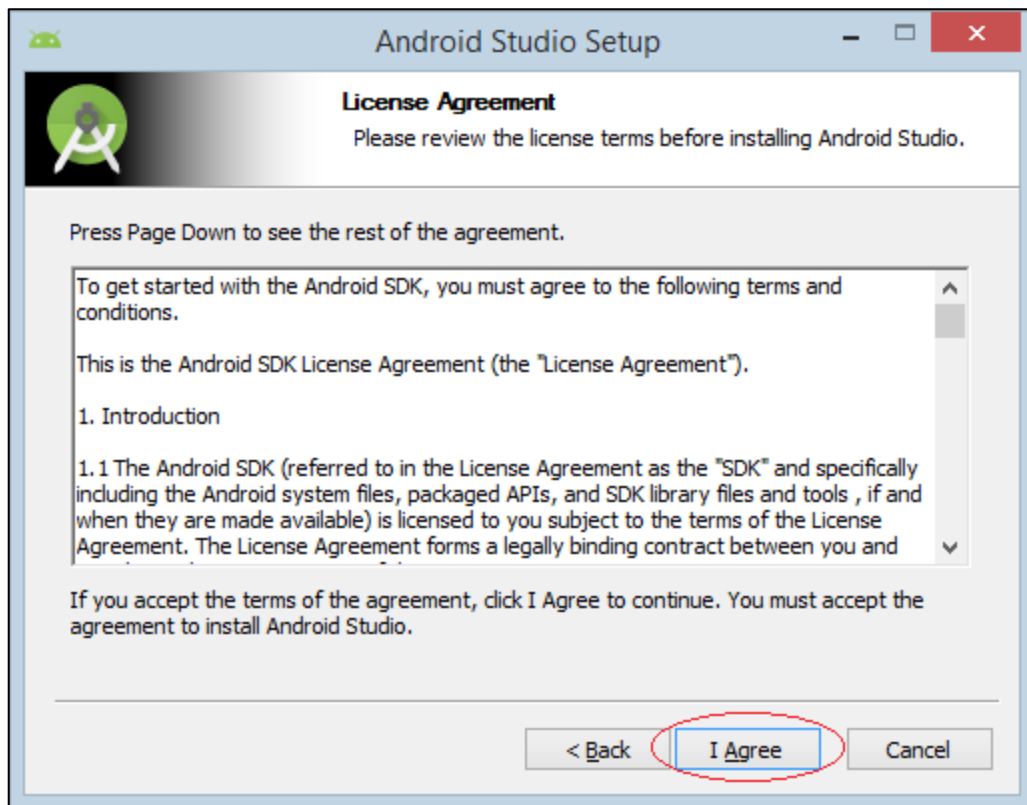
Bước 3: Cài đặt Android Studio bằng bộ cài vừa tải về



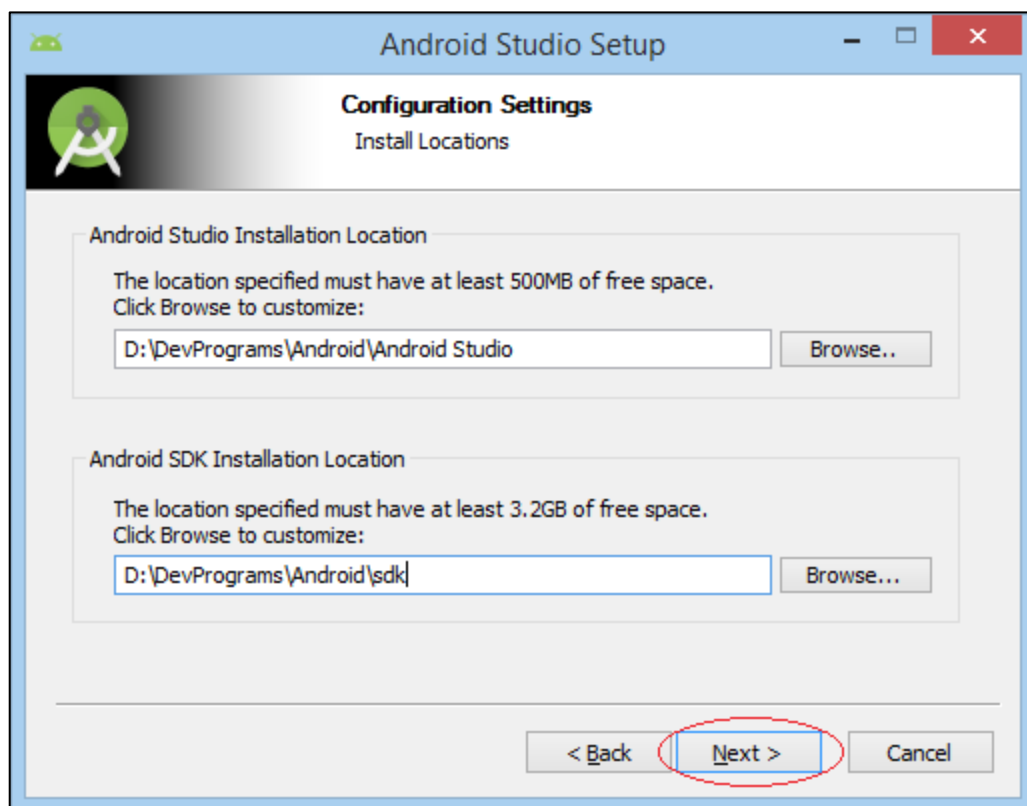
Hình 2.4. Giao diện cài đặt Android Studio



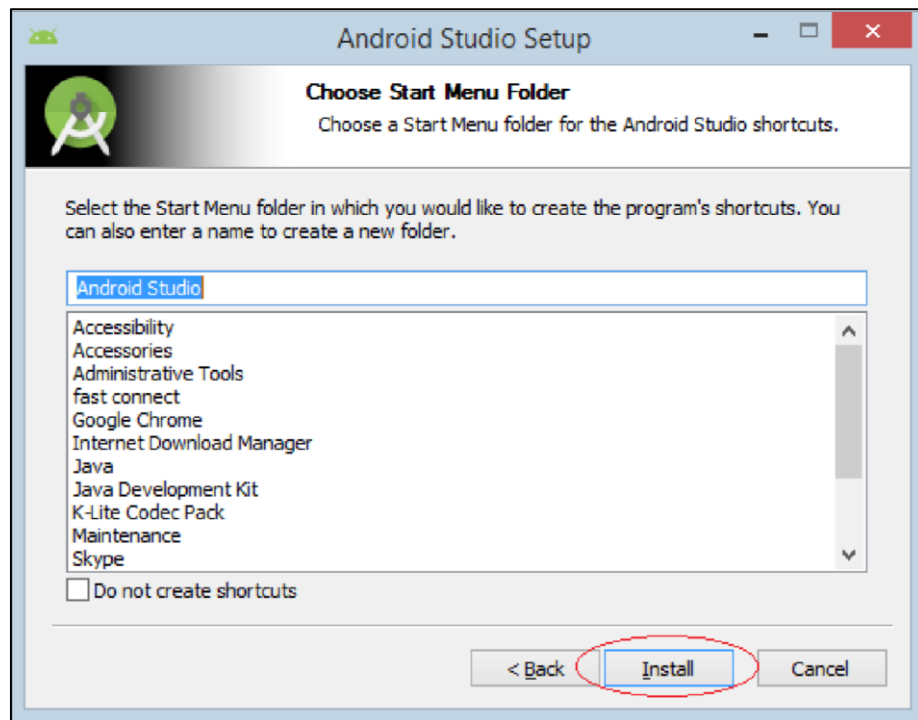
Hình 2.5. Tích vào tất cả các tùy chọn cài đặt



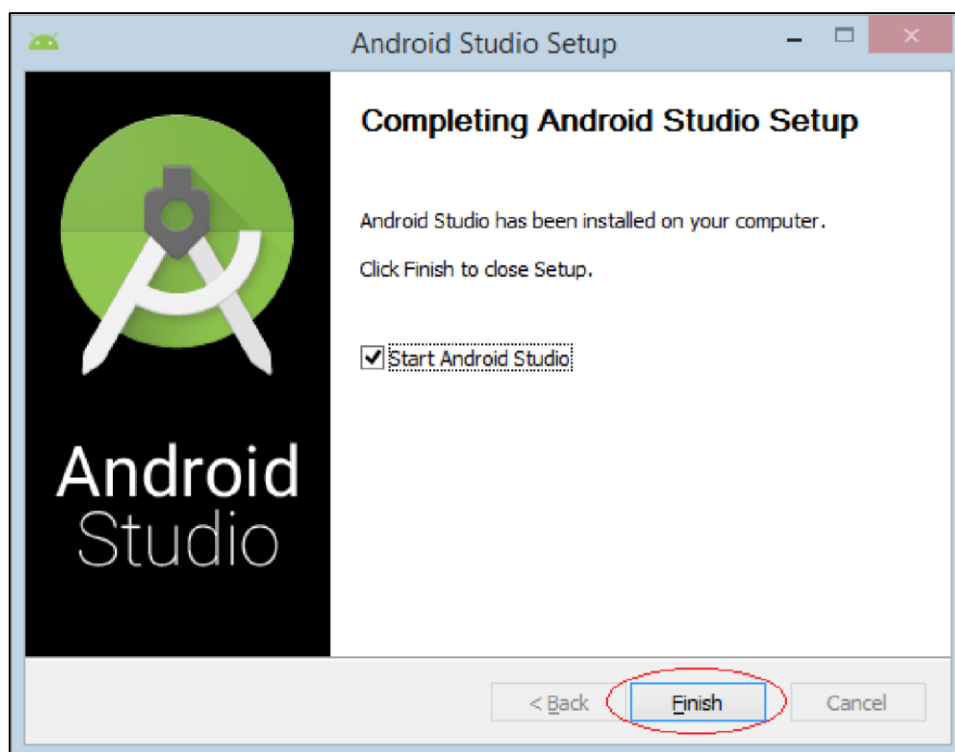
Hình 2.6. Đồng ý với điều khoản sử dụng



Hình 2.7. Chọn đường dẫn để bắt đầu cài đặt



Hình 2.8. Nhấn Install để bắt đầu cài đặt



Hình 2.9. Quá trình cài đặt Android Studio đã hoàn tất

2.3. Các thành phần của một dự án Android trong Android Studio

2.3.1. Tập cấu hình dự án AndroidManifest.xml

Trong bất kì một project Android nào khi tạo ra đều có một file AndroidManifest.xml, file này được dùng để định nghĩa các quyền (permission)

cũng như các giao diện (theme) cho ứng dụng. Đồng thời nó cũng chứa thông tin về main activity sẽ chạy đầu tiên.

File này được tự động sinh ra khi tạo một Android project. Trong file manifest bao giờ cũng có 3 thành phần chính đó là: application, permission và version.

a. Thành phần ứng dụng (Application)

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

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

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

android:theme = “drawable theme” => thuộc tính này để đặt giao diện chủ đề cho ứng dụng.

Ngoài ra còn nhiều thuộc tính khác...

b. Quyền truy cập và sử dụng tài nguyên (Permission)

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

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

2.3.2. Tệp R.java

File R.java là một file tự động sinh ra ngay khi tạo một dự án, file này được sử dụng để quản lý tất cả các đối tượng được khai báo trong tệp tin màn hình XML của ứng dụng và các tài nguyên hình ảnh.

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

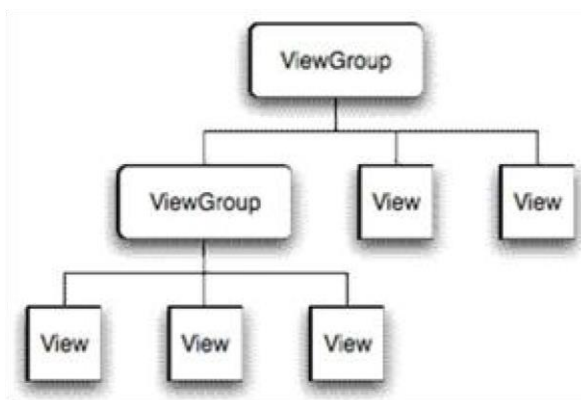
2.4. Các thành phần trong giao diện Android

2.4.1. View

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

Tất cả mọi widget đều có chung các thuộc tính cơ bản như là cách trình bày vị trí, nền, kích thước, lề,... Tất cả những thuộc tính chung này được thể hiện hết ở trong đối tượng View. Trong Android Platform, các màn hình (screen) luôn được bố trí theo một kiểu cấu trúc phân cấp như hình dưới.

Một màn hình là một tập hợp các Layout và các widget được bố trí có thứ tự. Để thể hiện một màn hình thì trong hàm onCreate của mỗi Activity cần phải được gọi một hàm là setContentView(R.layout.main); hàm này sẽ load giao diện từ file XML lên để phân tích thành mã bytecode.



Hình 2.4. View & ViewGroup

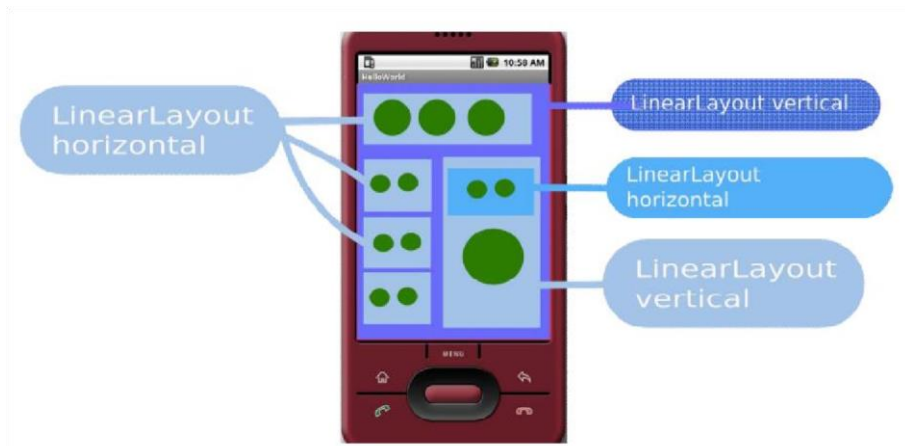
2.4.2. ViewGroup

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

a. *LinearLayout*

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

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



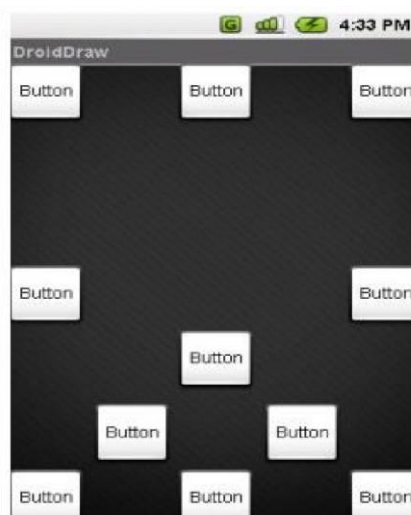
Hình 2.5. *LinearLayout*

b. AbsoluteLayout

Layout này được sử dụng để bố trí các widget vào một vị trí bất kỳ trong layout dựa vào 2 thuộc tính tọa độ x, y. Tuy nhiên, kiểu layout này rất ít khi được dùng bởi vì tọa độ của các đối tượng luôn cố định và sẽ không tự điều chỉnh được tỷ lệ khoảng cách giữa các đối tượng. Khi chuyển ứng dụng sang một màn hình có kích thước với màn hình thiết kế ban đầu thì vị trí của các đối tượng sẽ không còn được chính xác như ban đầu.

c. RelativeLayout

Layout này cho phép bố trí các widget theo một trục đối xứng ngang hoặc dọc. Để đặt được đúng vị trí thì các widget cần được xác định một mối ràng buộc nào đó với các widget khác. Các ràng buộc này là các ràng buộc trái, phải, trên, dưới so với một widget hoặc so với layout parent. Dựa vào những mối ràng buộc đó mà RelativeLayout cũng không phụ thuộc vào kích thước của screen thiết bị. Ngoài ra, nó còn có ưu điểm là giúp tiết kiệm layout sử dụng nhằm mục đích giảm lượng tài nguyên sử dụng khi load đồng thời đẩy nhanh quá trình xử lý.



Hình 2.6. RelativeLayout

2.4.3. Button

Sở dĩ widget button được giới thiệu đầu tiên trong số các widget khác là vì đây là đối tượng có thể nói là được dùng nhiều nhất trong hầu hết các ứng dụng Android.

Để thiết kế giao diện với một button ta có 2 cách như sau:

Thiết kế giao diện nút bấm bằng XML

```
<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/cmdButton1"
android:text="Touch me!"/>
```

Thay đổi thuộc tính và xử lý sự kiện bằng code

Để thay đổi thuộc tính hoặc xử lý một sự kiện của Button trong code ta làm như sau:

```
Button cmdButton = (Button) findViewById(R.id.cmdButton1);
cmdButton.setText("Touch Me!"); cmdButton.setOnClickListener(...);
```

2.4.4. ImageButton

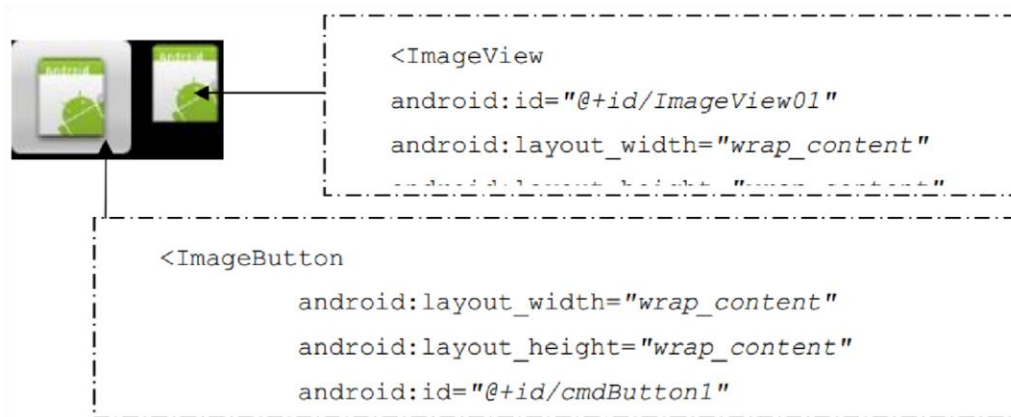
Cũng tương tự như Button, ImageButton chỉ có thêm một thuộc tính `android:src = "@drawable/icon"` để thêm hình ảnh vào và không có thẻ text.

```
<ImageButton
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/cmdButton1"
android:src="@drawable/icon"
android:onClick="touchMe"/>
```

2.4.5. ImageView

Được dùng để thể hiện một hình ảnh. Nó cũng giống như ImageButton, chỉ khác là không có hình dáng của một cái button.

```
ImageView iv = new ImageView(this);
iv.setImageResource(R.drawable.icon);
```



Hình 2.7. Sự khác nhau giữa *ImageView* và *ImageButton*

2.4.6. *ListView*

Được sử dụng để thể hiện một danh sách các thông tin theo từng cell. Mỗi cell thông thường được load lên từ một file XML đã được cố định trên đó số lượng thông tin và loại thông tin cần được thể hiện.

Để thể hiện được một list thông tin lên một screen thì cần phải có 3 yếu tố chính:

- **Data Source:** Data Source có thể là một ArrayList, HashMap hoặc bất kỳ một cấu trúc dữ liệu kiểu danh sách nào.
- **Adapter:** Adapter là một class trung gian giúp ánh xạ dữ liệu trong Data Source vào đúng vị trí hiển thị trong ListView. Chẳng hạn, trong Data Source có một trường name và trong ListView cũng có một TextView để thể hiện trường name này. Tuy nhiên, ListView sẽ không thể hiển thị dữ liệu trong Data Source lên được nếu như Adapter không gán dữ liệu vào cho đối tượng hiển thị.
- **ListView:** ListView là đối tượng để hiển thị các thông tin trong Data Source ra một cách trực quan và người dùng có thể thao tác trực tiếp trên đó.

2.4.7. *TextView*

TextView ngoài tác dụng là để hiển thị văn bản thì nó còn cho phép định dạng nội dung bằng thẻ html.

```
TextView textView = (TextView) findViewById(R.id.textView);
CharSequence styledText = Html.fromHtml("<i>This</i> is some
<b>styled</b> <s>text</s>"); textView.setText(styledText);
```

Nội dung TextView cũng có thể được định dạng bằng thẻ html ngay trong XML.

2.4.8. *EditText*

Trong Android đối tượng EditText được sử dụng như một TextField hoặc một TextBox.

```
<EditText
android:id="@+id/EditText01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textStyle="bold"
android:textSize="20dip"
android:textColor="#000000"
android:text="Hello Android!"
android:singleLine="true"
android:inputType="textCapWords"/>
```

Các thuộc tính cần chú ý sử dụng EditText đó là:

- **android:inputType** = “...” sử dụng để xác định phương thức nhập cho EditText. Chẳng hạn như khi bạn muốn một ô để nhập password hay một ô để nhập Email thì thuộc tính này sẽ làm điều đó.
- **android:singleLine** = “true” EditText của bạn sẽ trở thành một TextField, ngược lại sẽ là TextBox.

2.4.9. *CheckBox*

Nhận 2 giá trị true hoặc false. Đối tượng CheckBox cho phép chọn nhiều item cùng một lúc.

```
<CheckBox android:id="@+id/CheckBox01"
android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Check me"
android:checked="true"/>
```

2.4.10. *Activity & Intent*

a. *Activity*

Activity là một thành chính của một ứng dụng Android, được dùng để hiển thị một màn hình và nắm bắt các hoạt động xảy ra trên màn hình đó. Khi làm việc với Activity cần nắm bắt được một số kiến thức cơ bản như sau:

- Chu kỳ sống của một Activity
- Tạo menu và dialog
- Khởi động một Activity Để khởi động một Activity ta sử dụng Intent sẽ tìm hiểu kỹ hơn ở phần b.

Tuy nhiên, dùng nhiều nhất vẫn là cách chuyển giữa các Intent theo 2 loại:

o Khai báo không tường minh

Cung cấp chính xác thông tin của activity cần gọi bằng cách truyền vào tên class của Activity đó. VD: Từ Activity A muốn chuyển qua Activity B ta khai báo một Intent trong Activity A:

```
Intent Intent = new Intent(this, B.class);
startActivity(Intent);
```

- o Khai báo tường minh

Cung cấp các thao tác cần làm gì với loại dữ liệu nào, hệ thống sẽ tìm đến activity tương ứng để khởi động. VD: Để xem thông tin một contact nào đó trong Activity của ứng dụng Contact trong Android ta chỉ đến dữ liệu contact và chỉ đến Activity View contact như sau:

```
Intent i = new Intent();
i.setAction(Intent.ACTION_VIEW);
i.setData(Uri.withAppendedPath( android.provider.Contacts.
People.CONTENT_URI, "1)); startActivity(i);
```

- Tính liên lạc giữa 2 activity

Khi chuyển sang một Activity khác ta có thể gửi kèm dữ liệu trong Intent đó như sau:

```
Intent.putExtra("key1", "value1");
Intent.putExtra("key2", 23);
```

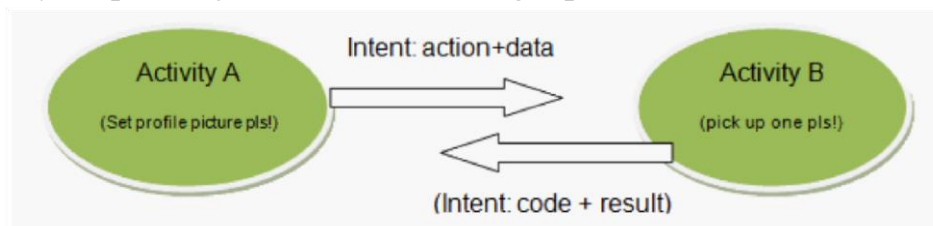
Bên phía Activity được khởi động hay được chuyển đến n, có thể lấy dữ liệu được gửi như sau:

```
getIntent().getStringExtra().getString("key1");
getIntent().getStringExtra().getInt("key2");
```

b. Intent

- Khái niệm Intent

- o Là một cấu trúc dữ liệu mô tả cách thức, đối tượng thực hiện của một Activity
- o Là cầu nối giữa các Activity: ứng dụng Android thường bao gồm nhiều Activity, mỗi Activity hoạt động độc lập với nhau và thực hiện những công việc khác nhau. Intent chính là người đưa thư, giúp các Activity có thể triệu gọi cũng như truyền các dữ liệu cần thiết tới một Activity khác. Điều này cũng giống như việc di chuyển qua lại giữa các Forms trong lập trình Windows Form.



Hình 2.8. Hoạt động của Intent

- Dữ liệu của Intent

- o Intent về cơ bản là một cấu trúc dữ liệu, được mô tả trong lớp android.content.Intent
- o Các thuộc tính của một đối tượng Intent:

| Thuộc tính chính | Thuộc tính phụ |
|--|--|
| action -tên (string) của action mà Intent sẽ yêu cầu thực hiện -có thể là action được Android định nghĩa sẵn (built-in standard action) hoặc do người lập trình tự định nghĩa | category -thông tin về nhóm của action |
| data -dữ liệu mà Activity được gọi sẽ xử lý -định dạng Uri (thông qua hàm Uri.parse(data)) | type -định dạng kiểu dữ liệu (chuẩn MIME) -thường được tự động xác định |
| | component -chỉ định cụ thể lớp sẽ thực thi Activity -khi được xác định, các thuộc tính khác trở thành không bắt buộc (optional) |
| | extras -chứa tất cả các cặp (key,value) do ứng dụng thêm vào để truyền qua Intent (cấu trúc Bundle) |
| http://developer.android.com/reference/android/content/Intent.html | |

Hình 2.9. Các thuộc tính của một đối tượng Intent

- o Các Action được định nghĩa sẵn:

| Built-in Standard Actions | |
|---|---|
| ACTION_MAIN | ACTION_ANSWER |
| ACTION_VIEW | ACTION_INSERT |
| ACTION_ATTACH_DATA | ACTION_DELETE |
| ACTION_EDIT | ACTION_RUN |
| ACTION_PICK | ACTION_SYNC |
| ACTION_CHOOSER | ACTION_PICK_ACTIVITY |
| ACTION_GET_CONTENT | ACTION_SEARCH |
| ACTION_DIAL | ACTION_WEB_SEARCH |
| ACTION_CALL | ACTION_FACTORY_TEST |
| ACTION_SEND | ACTION_SENDTO |
| Built-in Standard Broadcast Actions | |
| ACTION_TIME_TICK | ACTION_PACKAGE_RESTARTED |
| ACTION_TIME_CHANGED | ACTION_PACKAGE_DATA_CLEARED |
| ACTION_TIMEZONE_CHANGED | ACTION_UID_REMOVED |
| ACTION_BOOT_COMPLETED | ACTION_BATTERY_CHANGED |
| ACTION_PACKAGE_ADDED | ACTION_POWER_CONNECTED |
| ACTION_PACKAGE_CHANGED | ACTION_POWER_DISCONNECTED |
| ACTION_PACKAGE_REMOVED | ACTION_SHUTDOWN |

Hình 2.10. Các Action được định nghĩa sẵn

2.5. Dịch vụ web

2.5.1. Khái niệm

Dịch vụ web (Web Services) là những thành phần ứng dụng dùng để chuyển đổi một ứng dụng thông thường sang một ứng dụng web. Đồng thời nó cũng xuất bản các chức năng của mình để mọi người dùng internet trên thế giới đều có thể sử dụng thông qua nền tảng web. Web Services truyền thông tin bằng cách sử dụng các giao thức mở, tài nguyên phần mềm có thể xác định bằng địa chỉ URL, thực hiện các chức năng và đưa ra các thông tin người dùng yêu cầu, các ứng dụng độc lập và tự mô tả chính nó. Nó bao gồm các modul độc lập cho hoạt động của khách hàng và doanh nghiệp và bản thân nó được thực thi trên server. Bất cứ một ứng dụng nào cũng đều có thể có một thành phần WS. WS có thể được tạo ra bằng bất kỳ một ngôn ngữ lập trình nào

2.5.2. Đặc điểm

□ Cho phép client và server tương tác ngay cả trong môi trường khác nhau. (Ví dụ server chạy linux, client chạy windows).

- Phần lớn được xây dựng dựa trên mã nguồn mở và phát triển các chuẩn đã được công nhận.

- Nó có thể triển khai bởi 1 phần mềm ứng dụng phía server (Ví dụ : PHP, Oracle Application server, Microsoft .NET)

Mặc dù có nhiều ưu điểm: Có nhiều mô-đun và có thể công bố lên internet, khả năng hoạt động rộng lớn với phần mềm khác nhau chạy nền tảng khác nhau, sử dụng các giao thức chuẩn mở, nâng cao khả năng tái sử dụng, cho phép các tiến trình / chức năng nghiệp vụ đóng gói trong giao diện dịch vụ web, dễ dàng cho việc phát triển các ứng dụng phân tán. Nhưng những thiệt hại lớn sẽ xảy ra vào khoảng thời gian chết của dịch vụ web, có quá nhiều chuẩn cho dịch vụ web khiến người dùng khó nắm bắt, phải quan tâm nhiều hơn tới vấn đề an toàn và bảo mật.

2.5.3. Những chuẩn chính của Web Services

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

SOAP?

SOAP là một giao thức giao tiếp có cấu trúc như XML và mã hóa thành định dạng chung cho các ứng dụng trao đổi với nhau. Ý tưởng bắt đầu từ Microsoft và phần mềm Userland. Một đặc tả việc sử dụng các tài liệu XML theo dạng các thông điệp.

Đặc tả về SOAP định nghĩa một mô hình trao đổi dữ liệu dựa trên 3 khái niệm cơ bản: Các thông điệp là các tài liệu XML, chúng được truyền đi từ bên gửi đến bên nhận, bên nhận có thể chuyển tiếp dữ liệu đến nơi khác.

Đặc trưng của SOAP

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

WSDL?

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

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

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

UDDI?

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

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

Cấu trúc UDDI gồm các thành phần:

- White pages: chứa thông tin liên hệ và các định dạng chính yếu của web services (tên giao dịch, địa chỉ...) Những thông tin này cho phép các đối tượng khác xác định được service.
- Yellow pages: chứa thông tin mô tả web service theo những chủng loại khác nhau. Những thông tin này cho phép các đối tượng thấy web service theo từng chủng loại của nó.
- Green pages: chứa thông tin kỹ thuật mô tả các hành vi và các chức năng của web service để tìm kiếm.

REST?

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

Đặc trưng của REST

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

2.5.4. Các dạng tương tác giữa Web Service với ứng dụng trên thiết bị di động XML - eXtensible Markup Language

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

Định dạng JSON - JavaScript Object Notation [4]

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

Vì sao nên sử dụng JSON?

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

- Có cách phân tích xử lý đơn giản, chỉ qua 1 bước (chuyển chuỗi nhận được từ máy chủ sang dữ liệu có thể sử dụng được).

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1. Phát biểu bài toán

Trên thiết bị di động của chúng ta đôi khi có những tin nhắn quan trọng, nếu ta để lại trên máy có thể sẽ vô tình bị xóa mất hoặc máy bị virus, khiến chúng ta mất đi những thông tin cần thiết.

Để có thể bảo vệ được những tin nhắn quan trọng này, ta cần phải đưa thông tin sang một nơi khác tách khỏi điện thoại. Một trong những phương án khả thi nhất, đó là chúng ta sẽ lưu trữ những thông tin tin nhắn theo dạng online, tức là chuyển nó lên trên Host, ở đây là chúng ta sẽ lưu trữ trực tiếp.

Như vậy, bài toán chúng ta cần giải quyết ở đây là chống mất những nội dung tin nhắn quan trọng, do việc để trên điện thoại có thể bị xóa nhầm hoặc do virus, vì thế khi có tin nhắn đến, ta sẽ lấy nó tách khỏi điện thoại và đưa lên trên Host. Đó đó đề tài của em mới mang tên là “Xây dựng ứng dụng quản lý tin nhắn online”

2.2. Phân tích thiết kế dữ liệu

2.2.1. Phân tích dữ liệu

Tin nhắn có rất nhiều thông tin, nhưng trên khuôn khổ ứng dụng mà ta xây dựng, một bản tin nhắn cần lưu trữ ở trên Host sẽ chỉ cần có những thông tin cơ bản sau: Số điện thoại, ngày nhận, giờ nhận và nội dung của tin nhắn. Đó đó, ta sẽ xây dựng một bảng thiết kế các dữ liệu đã nêu trên

2.2.2. bảng thiết kế dữ liệu

Đây là các dữ liệu tin nhắn nhận được mà ta phải đưa lên host bao gồm: SĐT, ngày nhận, giờ nhận và nội dung

| STT | Tên trường | Kiểu dữ liệu | Độ rộng | Mô tả | Ghi chú |
|-----|------------|--------------|----------|---------------|------------|
| 1 | id | int | 11 | Định danh | Khóa chính |
| 2 | sdt | char | 12 | Số điện thoại | |
| 3 | ngaynhan | date | 10 | Ngày nhận | |
| 4 | gionhan | time | 8 | Giờ nhận | |
| 5 | noidung | text | longtext | Nội dung | |

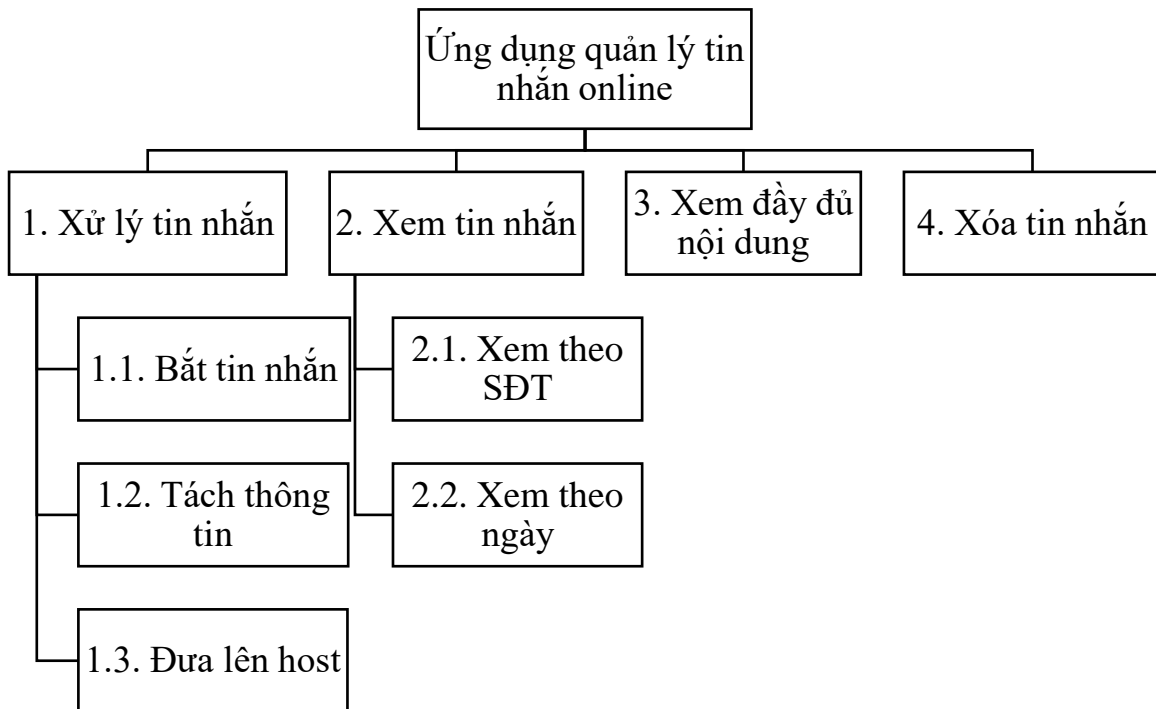
2.2.3. Bảng cơ sở dữ liệu vật lý

Dữ liệu vật lý hoàn chỉnh sau khi ta tạo xong ở trên hệ quản trị cơ sở dữ liệu

| # | Tên | Kiểu | Bảng mã đối chiếu | Thuộc tính | Null | Mặc định | Ghi chú | Thêm | Hành động |
|---|----------|----------|-------------------|------------|-------|----------|---------|----------------|-------------------|
| 1 | id | int(11) | | | Không | Không | | AUTO_INCREMENT | Thay đổi Xóa Thêm |
| 2 | sdt | char(12) | utf8_unicode_ci | | Không | Không | | | Thay đổi Xóa Thêm |
| 3 | ngaynhan | date | | | Không | Không | | | Thay đổi Xóa Thêm |
| 4 | gionhan | char(8) | utf8_unicode_ci | | Không | Không | | | Thay đổi Xóa Thêm |
| 5 | noidung | longtext | utf8_unicode_ci | | Không | | | | Thay đổi Xóa Thêm |

2.3. Phân tích thiết kế chức năng

2.3.1. Sơ đồ phân rã chức năng



Ta sẽ đi vào từng chức năng

1. Xử lý tin nhắn

1.1. Chức năng bắt tin nhắn đến

Khi có tin nhắn đến, để có thể bắt được tin nhắn, ta phải xây dựng một Class(lớp người dùng), ở đây ta đặt tên là Catchmessage trong android studio. Lớp này sẽ được kế thừa từ lớp Broadcast receiver(bắt tin quảng bá)

+ *Hàm khởi tạo:*

```
BroadcastReceiver receiver = null;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

+ *Đăng ký với hệ điều hành để nhận tin nhắn gửi đến*

```
IntentFilter filter = new IntentFilter("android.provider.Telephony.SMS_RECEIVED");
```

Để hệ điều hành android có thể gửi tin nhắn đến cho ứng dụng này thì trong android manifest ta phải đăng ký bắt sự kiện tin nhắn đến như sau:

```
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<intent-filter>
  <action android:name="android.intent.action.MAIN" />

  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Vì có phần đăng ký này trong android manifest, nên khi tin nhắn đến thì hệ điều hành mới có thể gửi tin nhắn đến ứng dụng để xử lý

1.2. Tách thông tin tin nhắn

Sau khi đã bắt được tin nhắn(hay lấy tin nhắn), ta sẽ tiếp tục xây dựng thêm một hàm nữa là hàm “processReceive” hay còn gọi là hàm xử lý tin nhắn. Hàm này sẽ tiến hành bóc tách các thông tin tin nhắn cần thiết bao gồm: SĐT, Ngày nhận, Giờ nhận, Nội dung

```
private void processReceive(Context context, Intent intent) {
    Toast.makeText(getApplicationContext(), "Alo ... có tin nhắn mới!",
    Toast.LENGTH_SHORT).show();
```

+ Khai báo biến *pdu* trong *Bundle* để lấy gói tin nhắn bao gồm các thông tin trên

```
String sms_extra = "pdu";
Bundle bundle = intent.getExtras();
```

Biến *Bundle* trả về các tin nhắn nhận được cùng một lúc chứa trong mảng lớn array và trong mảng lớn array sẽ chứa các mảng con object, mỗi một object sẽ là một tin nhắn khác nhau nhận được

```
Object[] obj_arr = (Object[]) bundle.get(sms_extra);
```

+ Dùng vòng lặp *for* để đọc từng tin nhắn

```
for(int i=0; i<obj_arr.length; i++){
```

+ Lệnh *createFromPdu* chuyển đổi để kiểm tra tin nhắn

```
SmsMessage smsMsg = SmsMessage.createFromPdu((byte[]) obj_arr[i]);
```

Sau khi đọc tin nhắn, ta sẽ tiến hành tách thông tin tin nhắn(hay còn gọi là phân tích tin nhắn) bao gồm các thông tin như đã nêu trên

+ *Lấy nội dung tin nhắn, ở đây ta khai báo là biến body*

```
String body = smsMsg.getMessageBody();
```

+ *Lấy SĐT nhắn tin, ở đây ta khai báo là biến phone_gui*

```
String phone_gui = smsMsg.getDisplayOriginatingAddress();
```

+ *Lấy thông tin ngày nhận, ở đây ta khai báo là biến date_nhan và định dạng là năm-tháng-ngày*

```
Date date = new Date(System.currentTimeMillis());
```

```
SimpleDateFormat date_format = new SimpleDateFormat("yyyy-MM-dd");  
String date_nhan = date_format.format(date);
```

+ *Lấy thông tin giờ nhận, ở đây ta khai báo là biến time_nhan và định dạng là giờ-phút-giây*

```
SimpleDateFormat time_format = new SimpleDateFormat("HH:mm:ss");  
String time_nhan = time_format.format(date);
```

1.3. Đưa thông tin lên Host

Sau khi đã bóc tách hay phân tích các thông tin tin nhắn xong, ta sẽ tiến hành đưa tất cả các thông tin trên lên Host. Và để có thể đưa được các thông tin trên lên Host, ta phải thông qua một bước trung gian là Webservice được viết bằng ngôn ngữ PHP

+ *Đưa nội dung tin nhắn lên Host*

```
xuLyThemTinNhan(phone_gui, date_nhan, time_nhan, body);
```

+ *Tiến hành xây dựng hàm xử lý thêm tin nhắn, đầu tiên là đẩy dữ liệu từ ứng dụng lên Webservice thêm tin nhắn, cấu trúc như sau*

```
public void xuLyThemTinNhan(final String phone_gui, final String date_nhan, final String  
time_nhan, final String body){
```

```
StringRequest stringRequest = new StringRequest(Request.Method.POST,  
Server.url_themtinnhan, new Response.Listener<String>() {
```

+ *Kiểm tra phản hồi lại xem thêm tin nhắn lên Host thành công hay thất bại*

```

public void onResponse(String response) {
    // Toast.makeText(getApplicationContext(),response,Toast.LENGTH_LONG).show();
    if(response.equals("1")){
        Toast.makeText(getApplicationContext(),"Thêm tin nhắn thành công!",
Toast.LENGTH_LONG).show();
    } else {
        Toast.makeText(getApplicationContext(),"Thêm tin nhắn thất bại!",
Toast.LENGTH_LONG).show();
    }
}
+ Nếu có lỗi xảy ra(có thể do đường truyền mạng hay lỗi dữ liệu)

```

```

new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(getApplicationContext(), "Lỗi thêm tin nhắn: "+error.toString(),
Toast.LENGTH_LONG).show();
    }
}

```

+ *Chuyển thành định dạng Json(đã mã hóa thông tin)*

```

@Override
protected Map<String, String> getParams() throws AuthFailureError {
    JSONArray jsonArray = new JSONArray();
    JSONObject jsonObject= new JSONObject();
    try {
        jsonObject.put("phone_gui", phone_gui);
        jsonObject.put("date_nhan", date_nhan);
        jsonObject.put("time_nhan", time_nhan);
        jsonObject.put("body", body);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    jsonArray.put(jsonObject);

    HashMap<String, String> hashMap = new HashMap<>();
    hashMap.put("tinnhan_json", jsonArray.toString());
    return hashMap;
}
};
RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
requestQueue.add(stringRequest);

```

+ *Tiến hành viết Webservice Thêm tin nhắn lên Host*

```

<?php
include "connection.php";

$DEBUGER = 0;

if($DEBUGER == 0){
    $tinnhan_json = $_POST['tinnhan_json'];
    $tinnhan_info = json_decode($tinnhan_json, true);

    foreach ($tinnhan_info as $value) {
        $sdt = $value['phone_gui'];
        $ngaynhan = $value['date_nhan'];
        $gionhan = $value['time_nhan'];
        $noidung = $value['body'];
    }
    //echo $sdt.' '.$ngaynhan.' '.$gionhan.' '.$noidung;
}

if($DEBUGER == 1){
    $sdt = '0352354982';
    $ngaynhan = '25-5-2020';
    $gionhan = '17.00';
    $noidung = 'xin chào';
}
}

// Viet cau truy van them tin nhan moi
$query = "INSERT INTO tinnhan(sdt,ngaynhan,gionhan,noidung)
VALUES ('$sdt', '$ngaynhan', '$gionhan', '$noidung')";

// Thuc thi cau truy van them tin nhan moi
$tinnhan_index = mysqli_query($link, $query);
if($tinnhan_index){
    echo "1";
} else {
    echo "0";
}
?>

```

Các thông tin tin nhắn gửi lên Host sẽ ở dạng file Json do đã mã hóa, nhiệm vụ của Webservice là phải giải mã file tập tin này và đưa về dạng dữ liệu thông tin có thể đọc được

2. Xem tin nhắn

Chức năng này cho phép người dùng có thể tìm kiếm lại và xem các tin nhắn đã gửi từ điện thoại lên cơ sở dữ liệu của Host theo SĐT đã nhận được hoặc là xem theo ngày

2.1. Xem tin nhắn theo Số Điện Thoại

Ta cũng sẽ tạo một class đặt tên là XemTheoPhone trong android studio và cũng sẽ viết một Webservice nữa để lấy tin nhắn theo SĐT từ cơ sở dữ liệu trên Host và hiển thị trong ứng dụng bằng công cụ ListView khi ta tìm kiếm

+ Xây dựng lớp adapter đổ về dạng ListView cho ứng dụng khi ta tìm kiếm

```
public class AdapterTinNhan extends BaseAdapter {
```

```
Context context;  
ArrayList<TinNhan> arrTinNhan;  
int id_tinnhan;
```

```
public AdapterTinNhan(Context context, ArrayList<TinNhan> arrTinNhan) {  
    this.context = context;  
    this.arrTinNhan = arrTinNhan;  
}
```

```
@Override  
public int getCount() {  
    return arrTinNhan.size();  
}
```

```
@Override  
public Object getItem(int position) {  
    return arrTinNhan.get(position);  
}
```

```
@Override  
public long getItemId(int position) {  
    return arrTinNhan.get(position).id;  
}
```

+ Tiến hành xây dựng hàm xử lý lấy tin nhắn từ cơ sở dữ liệu trên Host đổ về ứng dụng thông qua Webservice

```
private void timTheoPhone(){  
    StringRequest stringRequest = new StringRequest(Request.Method.POST,  
Server.url_laytinnhantheophone, new Response.Listener<String>() {
```

+ Kiểm tra phản hồi lại xem có lấy được danh sách tin nhắn hay không

```
public void onResponse(String response) {  
    //Toast.makeText(getApplicationContext(), response, Toast.LENGTH_LONG).show();  
    if (response.equals("0")) {  
        Toast.makeText(getApplicationContext(), "Không lấy được danh sách Tin Nhắn theo SĐT!", Toast.LENGTH_SHORT).show();  
    } else { // Lấy được danh sách tin nhan  
        try {  
            if (arrTinNhan.size() > 0) {  
                arrTinNhan.clear();  
            }  
            JSONArray jsonArray = new JSONArray(response);  
            for (int i = 0; i < jsonArray.length(); i++) {  
                JSONObject jsonObject = (JSONObject) jsonArray.get(i);  
                TinNhan tinNhan = new TinNhan();  
                tinNhan.id = jsonObject.getInt("id");  
                tinNhan.phone_gui = jsonObject.getString("sdt");
```

```

tinNhan.date_nhan = jsonObject.getString("ngaynhan");
tinNhan.time_nhan = jsonObject.getString("gionhan");
tinNhan.body = jsonObject.getString("noidung");
arrTinNhan.add(tinNhan);
AdapterTinNhan adapterTinNhan = new AdapterTinNhan(getApplicationContext(),
arrTinNhan);
lv_sodienthoai.setAdapter(adapterTinNhan);
adapterTinNhan.notifyDataSetChanged();
}
} catch (JSONException e) {
e.printStackTrace();
}

```

Nếu phản hồi sai thì không lấy được danh sách tin nhắn, còn ngược lại dữ liệu đổ về ứng dụng sẽ ở dạng Json bị mã hóa và trong đó có đầy đủ các thông tin tin nhắn được hiển thị về dạng ListView khi ta tìm kiếm theo SĐT

+ Nếu có lỗi xảy ra, phản hồi lại trên màn hình lỗi lấy tin nhắn theo SĐT

```

new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
Toast.makeText(getApplicationContext(), "Lỗi lấy tin nhắn theo SĐT : "+error,
Toast.LENGTH_LONG).show();
}
}

```

+ Mã hóa dữ liệu tin nhắn ở dạng Json khi lấy tin nhắn từ Host về ứng dụng

```

protected Map<String, String> getParams() throws AuthFailureError {
JSONArray jsonArray = new JSONArray();
JSONObject jsonPhone = new JSONObject();
try {
jsonPhone.put("phone", edt_sodienthoai.getText().toString());
} catch (JSONException e) {
e.printStackTrace();
}
jsonArray.put(jsonPhone);
HashMap<String, String> hashMap = new HashMap<String, String>();
hashMap.put("phone_json", jsonArray.toString());
return hashMap;
}
};
RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
requestQueue.add(stringRequest);
}

```

+ Tiến hành viết Webservice lấy tin nhắn theo SĐT từ Host về ứng dụng

```

<?php
include "connection.php";
$DEBUGER = 0;
if($DEBUGER == 0){

    $phone_json = $_POST['phone_json'];
    $phone_info = json_decode($phone_json, true);

    foreach ($phone_info as $value) {
        $phone = $value['phone'];
    }

}

if($DEBUGER == 1){
    $phone = '+15555215557';
}

// Tao truy van

$query = "SELECT id,sdt,ngaynhan,gionhan,noidung FROM tinnhan WHERE sdt='$phone'";

```

```

//Truy van du lieu

$tinnhan_detail = mysqli_query($link, $query);

// Dem so ban ghi trong tinnhan_detail

$rownumber = mysqli_num_rows($tinnhan_detail);

//echo 'so ban ghi: '.$rownumber;

```

```

if($rownumber > 0){

    $arrayTinNhan=array();

    while($row=mysqli_fetch_array($tinnhan_detail))

    {

        $id = $row['id'];

        $sdt = $row['sdt'];

        $ngaynhan = $row['ngaynhan'];

        $gionhan = $row['gionhan'];

        $noidung = $row['noidung'];

        array_push($arrayTinNhan,new TinNhan($id, $sdt, $ngaynhan, $gionhan, $noidung));

    }

    echo json_encode($arrayTinNhan);

} else {

    echo "0";
}

```


2.2. Xem theo ngày

Ta cũng sẽ tạo một Class mang tên là XemTheoNgày trong android studio và cũng sẽ viết một Webservice nữa để lấy tin nhắn theo ngày từ cơ sở dữ liệu trên Host và đổ về ứng dụng theo dạng ListView khi ta tìm kiếm

+ *Xây dựng lớp adapter để lấy tin nhắn theo ngày đổ về dạng ListView cho ứng dụng khi ta tìm kiếm*

```
public class AdapterTinNhan extends BaseAdapter {

    Context context;
    ArrayList<TinNhan> arrTinNhan;
    int id_tinnhan;

    public AdapterTinNhan(Context context, ArrayList<TinNhan> arrTinNhan) {
        this.context = context;
        this.arrTinNhan = arrTinNhan;
    }

    @Override
    public int getCount() {
        return arrTinNhan.size();
    }

    @Override
    public Object getItem(int position) {
        return arrTinNhan.get(position);
    }

    @Override
    public long getItemId(int position) {
        return arrTinNhan.get(position).id;
    }
}
```

+ *Tiến hành xây dựng hàm xử lý lấy tin nhắn từ cơ sở dữ liệu trên Host đổ về ứng dụng thông qua Webservice*

```
private void timtheongay() {
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
    Server.url_laytinnhantheongay, new Response.Listener<String>() {
```

+ *Kiểm tra phản hồi lại xem có lấy được danh sách tin nhắn hay không*

```
public void onResponse(String response) {
    //Toast.makeText(getApplicationContext(), response, Toast.LENGTH_SHORT).show();
    if(response.equals("0")){
        Toast.makeText(getApplicationContext(), "Không lấy được danh sách Tin Nhắn!",
        Toast.LENGTH_SHORT).show();
    }
```

```

} else { // Lấy được danh sách tin nhắn
    try {
        if(arrTinNhan.size() > 0) {
            arrTinNhan.clear();
        }
        JSONArray jsonArray = new JSONArray(response);
        //Toast.makeText(getApplicationContext(), response, Toast.LENGTH_SHORT).show();
        for(int i=0;i<jsonArray.length();i++){
            JSONObject jsonObject = (JSONObject) jsonArray.get(i);
            TinNhan tinNhan = new TinNhan();
            tinNhan.id = jsonObject.getInt("id");
            tinNhan.phone_gui = jsonObject.getString("sdt");
            tinNhan.date_nhan = jsonObject.getString("ngaynhan");
            tinNhan.time_nhan = jsonObject.getString("gionhan");
            tinNhan.body = jsonObject.getString("noidung");

            arrTinNhan.add(tinNhan);
            AdapterTinNhan adapterTinNhan = new AdapterTinNhan(getApplicationContext(),
arrTinNhan);
            lv_xemtheongay.setAdapter(adapterTinNhan);
            adapterTinNhan.notifyDataSetChanged();
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

```

Nếu phản hồi sai thì không lấy được danh sách tin nhắn, còn ngược lại dữ liệu đổ về ứng dụng sẽ ở dạng Json bị mã hóa và trong đó có đầy đủ các thông tin tin nhắn được hiển thị về dạng ListView khi ta tìm kiếm theo ngày

+ Nếu có lỗi xảy ra, phản hồi lại trên màn hình lỗi lấy tin nhắn

```

new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(getApplicationContext(), "Lỗi lấy tin nhắn từ HOST: "+error,
Toast.LENGTH_SHORT).show();
    }
}

```

+ Mã hóa dữ liệu tin nhắn ở dạng Json khi lấy tin nhắn từ Host về ứng dụng

```

protected Map<String, String> getParams() throws AuthFailureError {
    JSONArray jsonArray = new JSONArray();
    JSONObject jsonDate = new JSONObject();

    try {
        jsonDate.put("date_start", edt_tungay.getText().toString());
        jsonDate.put("date_end", edt_denngay.getText().toString());
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

```

```

        jsonArray.put(jsonDate);
        HashMap<String, String> hashMap = new HashMap<String, String>();
        hashMap.put("date_json", jsonArray.toString());
        return hashMap;
    }
};
RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
requestQueue.add(stringRequest);

```

+ Chức năng datepicker để người dùng chọn nhanh ngày tháng năm theo lịch mà không cần nhập thủ công

```

private void chọnNgàyBatDau(){
    DatePickerDialog datePickerDialog = new DatePickerDialog(XemTheoNgay.this, new
    DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
            String thang=""; ngay="";
            if(month<10){
                thang=thang+"0"+(month+1);
            } else {
                thang=thang+(month+1);
            }
            if(dayOfMonth<10){
                ngay=ngay+"0"+dayOfMonth;
            } else {
                ngay = ngay+dayOfMonth;
            }
            edt_tungay.setText(year+"-"+thang+"-"+ngay);
        }
    },2020,00,01);
    datePickerDialog.show();
}

```

```

private void chọnNgàyKetThuc(){
    DatePickerDialog datePickerDialog = new DatePickerDialog(XemTheoNgay.this, new
    DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
            String thang=""; ngay="";
            if(month<10){
                thang=thang+"0"+(month+1);
            } else {
                thang=thang+(month+1);
            }
            if(dayOfMonth<10){
                ngay=ngay+"0"+dayOfMonth;
            } else {

```

```

        ngay = ngay+dayOfMonth;
    }
    edt_denngay.setText(year+"-"+thang+"-"+ngay);
}
},2020,00,01);
datePickerDialog.show();
}

```

+ Tiến hành viết Webservice lấy tin nhắn theo ngày từ Host về ứng dụng

```

<?php
include "connection.php";
$DEBUGER = 0;

if($DEBUGER == 0){
    $date_json = $_POST['date_json'];
    $date_info = json_decode($date_json, true);

    foreach ($date_info as $value) {
        $date_start = $value['date_start'];
        $date_end = $value['date_end'];
    }
}

if($DEBUGER == 1){
    $date_start = '2020-05-01';
    $date_end = '2020-05-31';
}

// Tao truy van
$query = "SELECT id,sdt,ngaynhan,gionhan,noidung FROM tinnhan WHERE ngaynhan>='$date_start' AND
ngaynhan<='$date_end'";

//Truy van du lieu
$tinnhan_detail = mysqli_query($link, $query);

```

```
// Dem so ban ghi trong tinnhan_detail
$rownumber = mysqli_num_rows($tinnhan_detail);
//echo 'so ban ghi: '.$rownumber;

if($rownumber > 0){
    $arrayTinNhan=array();
    while($row=mysqli_fetch_array($tinnhan_detail))
    {
        $id = $row['id'];
        $sdt = $row['sdt'];
        $ngaynhan = $row['ngaynhan'];
        $gionhan = $row['gionhan'];
        $noidung = $row['noidung'];
        array_push($arrayTinNhan,new TinNhan($id, $sdt, $ngaynhan, $gionhan, $noidung));
    }
    echo json_encode($arrayTinNhan);
} else {
    echo "0";
}
```

3. Xem đầy đủ nội dung tin nhắn

Đây là chức năng giúp cho người dùng có thể xem được đầy đủ nội dung của những tin nhắn quá dài mà trong phạm vi hiển thị của màn hình không thể hiển thị được đầy đủ. Khi đó nội dung tin nhắn sẽ được hiển thị đầy đủ ở một màn hình khác. Ta cũng sẽ tạo một class XemFullTinNhan, và để xem đầy đủ được tin nhắn, ta xây dựng cấu trúc như sau:

```
public class XemFullTinNhan extends AppCompatActivity {
```

```
    TextView txt_fullbody;
```

```
    Button btn_thoatxemfulltinnhan;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_xem_full_tin_nhan);
```

```
        addViews();
```

```
        addEvents();
```

```
        Intent data = getIntent();
```

```
        String body = data.getStringExtra("body");
```

```
        txt_fullbody.setText(body);
```

+ *Thiết lập chuyển màn hình để hiển thị được đầy đủ tin nhắn dài*

```
private void addViews() {  
    txt_fullbody = findViewById(R.id.txt_fullbody);
```

+ *Xây dựng cấu trúc trong Adapter*

```
private void xemFullTinNhan(String body){  
    Intent intent_xemfulltinnhan = new Intent(context, XemFullTinNhan.class);  
    intent_xemfulltinnhan.putExtra("body", body);  
  
    intent_xemfulltinnhan.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
    intent_xemfulltinnhan.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);  
    context.startActivity(intent_xemfulltinnhan);
```

4. Xóa tin nhắn

Với những tin nhắn không cần thiết hoặc không dùng đến nữa ta sẽ xóa đi và ta sẽ xóa trên cơ sở dữ liệu của Host. Để có thể xóa được ta cũng phải thông qua một Webservice.

Ta cũng sẽ tạo một Class mới là XoaTinNhan và ta xây dựng cấu trúc như sau:

```
public class XoaTinNhan extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_xoa_tin_nhan);  
  
        xacNhanXoaTinNhan();  
    }  
  
    private void xacNhanXoaTinNhan() {  
  
        AlertDialog.Builder alertDialog = new AlertDialog.Builder(XoaTinNhan.this);  
        alertDialog.setTitle("Thông Báo");  
        alertDialog.setIcon(R.drawable.ic_delete);  
        alertDialog.setMessage("Bạn muốn xóa tin nhắn này?");  
  
        alertDialog.setPositiveButton("YES", new DialogInterface.OnClickListener() {  
            @Override  
            public void onClick(DialogInterface dialog, int which) {  
                Intent data = getIntent();// Lấy id tin nhắn xóa truyền sang từ Adaptertinnhan  
                int id = data.getIntExtra("id",0);  
                xuLyXoaTinNhan(id);  
                finish();  
            }  
        });  
  
        alertDialog.setNegativeButton("NO", new DialogInterface.OnClickListener() {  
            @Override
```

```

        public void onClick(DialogInterface dialog, int which) {
            finish();
        }
    });
    alertDialog.show();
}

```

+ *Lấy thông tin nhắn từ Host về điện thoại để xóa*

```

private void xuLyXoaTinNhan(final int id){
    StringRequest stringRequest = new StringRequest(Request.Method.POST,
    Server.url_xoa_tin_nhan, new Response.Listener<String>() {

```

+ *Kiểm tra phản hồi từ Host xem tin nhắn đã được xóa hay chưa*

```

@Override
    public void onResponse(String response) {
        if (response.equals("1")) {
            Toast.makeText(getApplicationContext(), "Xóa tin nhắn thành công!",
            Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(getApplicationContext(), "Xóa tin nhắn thất bại!",
            Toast.LENGTH_LONG).show();
        }
    }
}

```

+ *Kiểm tra phản hồi lại nếu có lỗi xảy ra (có thể do mạng hay dữ liệu bị lỗi)*

```

new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(XoaTinNhan.this, "Lỗi xóa tin nhắn: " + error.toString(),
        Toast.LENGTH_LONG).show();
    }
}

```

+ *Giải mã Json về text khi Host trả về di động*

```

protected Map<String, String> getParams() throws AuthFailureError {
    JSONArray jsonArray = new JSONArray();
    JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put("id_xoa", id);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    jsonArray.put(jsonObject);

    HashMap<String, String> hashMap = new HashMap<>();
    hashMap.put("id_xoa_json", jsonArray.toString());
    return hashMap;
}

```

```

    }
};
RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
requestQueue.add(stringRequest);
}

```

+ Chuyển từ màn hình xem sang màn hình xóa

```

private void xoaTinNhan(int id){
    Intent intent_xoatinnhan = new Intent(context, XoaTinNhan.class);
    intent_xoatinnhan.putExtra("id", id);
    intent_xoatinnhan.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    intent_xoatinnhan.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    context.startActivity(intent_xoatinnhan);
}

```

+ Xây dựng Webservice để xóa tin nhắn

```

<?php
include "connection.php";
$DEBUSER = 0;
if($DEBUSER == 0){
    $idxoa_json = $_POST['idxoa_json'];
    $idxoa_info = json_decode($idxoa_json, true);
    foreach ($idxoa_info as $value) {
        $id_xoa = $value['id_xoa'];
    }
}
if($DEBUSER == 1){
    $id_xoa = 40;
}

// Tao truy van
$query = "DELETE FROM tinnhan WHERE id=$id_xoa";
//Truy van du lieu
$result = mysqli_query($link, $query);

if($result){
    echo "1";
} else {
    echo "0";
}

?>

```

Tin nhắn ta muốn xóa chính là dữ liệu tin nhắn từ Host trả về điện thoại ở dạng Json, Webservice sẽ giải mã nó và khi ta xóa tin nhắn đó đi thì cả ở trên điện thoại và Host cũng sẽ không còn

CHƯƠNG 3: CHƯƠNG TRÌNH THỰC NGHIỆM

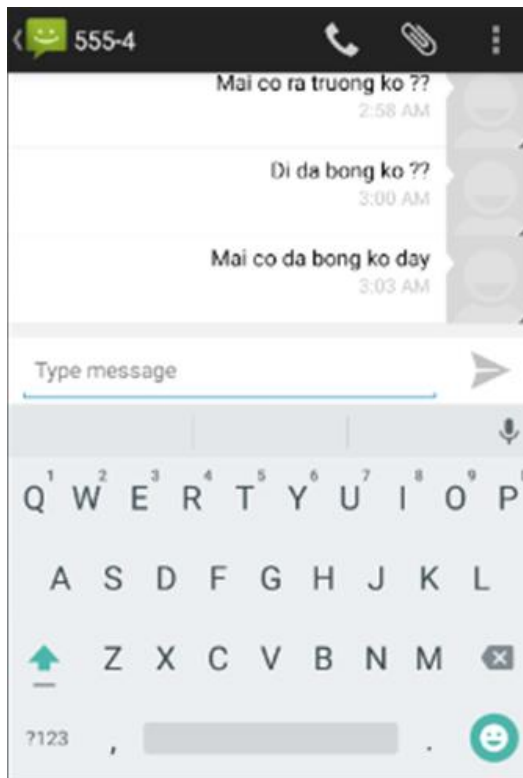
3.1. Kết quả đã đạt được

Sau 3 tháng miệt mài tìm hiểu, bỏ nhiều công sức và đặc biệt là được sự giúp đỡ nhiệt tình của thầy Phùng Anh Tuấn, em đã hoàn thành được một ứng dụng “Quản lý tin nhắn online” cơ bản. Do thời gian gấp rút và khối lượng công việc nhiều, nên ứng dụng của em còn nhiều thiếu sót và hạn chế, nhưng nhìn chung em đã làm được những chức năng cơ bản và cần thiết để thuận tiện cho người dùng có thể sử dụng, hy vọng trong tương lai em sẽ khắc phục và đưa ra thêm nhiều ý tưởng để hoàn thiện ứng dụng hơn. Sau đây, em xin được giải thích các chức năng cơ bản có trong ứng dụng, ta sẽ đi vào từng mục

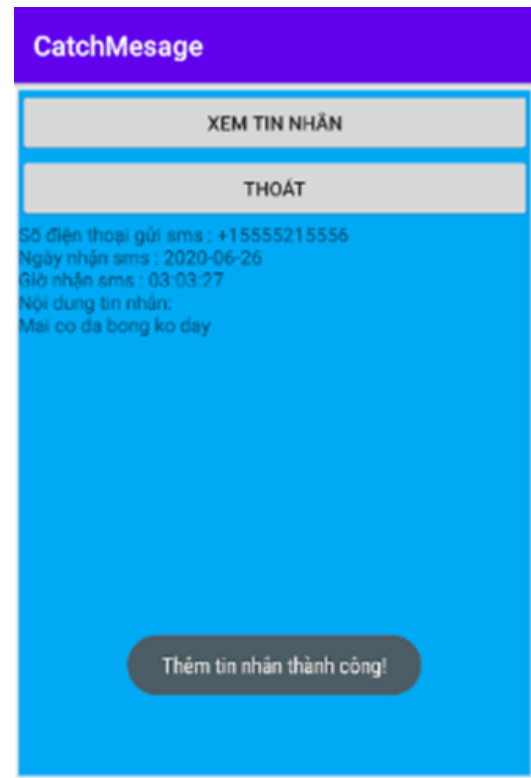
3.2. Một số giao diện chính

3.2.1. Giao diện bắt tin nhắn và đưa lên Host

Khi có người gửi tin nhắn đến, toàn bộ thông tin tin nhắn sẽ đều được hiển thị đầy đủ lên màn hình, bao gồm: SĐT, Ngày nhận, Giờ nhận, Nội dung tin nhắn

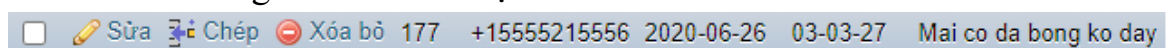


Hình 3.2. Giao diện gửi tin nhắn



Hình 3.3. Giao diện hiển thị tin nhắn nhận được

Thêm thành công lên cơ sở dữ liệu của Host



3.2.2. Giao diện quản lý tin nhắn

Đây là chức năng giúp cho người dùng có thể tìm kiếm, xem lại các tin nhắn của mình theo số điện thoại hoặc theo ngày từ trên Host.



Hình 3.4. Giao diện màn hình chính



Hình 3.5. Giao diện màn hình quản lý tin nhắn online

Trong chức năng quản lý tin nhắn online ta sẽ có 2 chức năng con đó là xem tin nhắn theo ngày và theo số điện thoại

a. Giao diện xem tin nhắn theo SĐT

Đây là chức năng giúp cho người dùng tìm kiếm được các tin nhắn mình cần từ trên Host theo SĐT. Người dùng chỉ cần nhập SĐT cần tìm vào và nhấn nút “Tìm theo SĐT”, lập tức màn hình sẽ hiển thị đầy đủ các tin nhắn người dùng cần theo một loạt các ListView ở bên dưới theo SĐT đã nhập



Hình 3.6. Màn hình xem tin nhắn theo SĐT

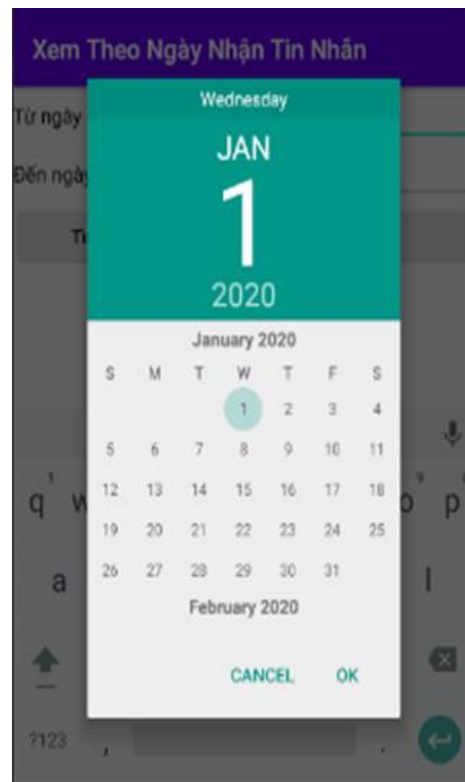
b. Giao diện xem tin nhắn theo ngày

Đây là chức năng giúp người dùng tìm kiếm được tất cả các tin nhắn cần thiết từ trên Host theo ngày tháng năm và tất cả các tin nhắn trong khoảng từ ngày bao nhiêu đến ngày bao nhiêu. Khi người dùng nhập ngày tháng năm vào và ấn nút “Tìm Theo Ngày”, lập tức màn hình sẽ hiển thị một loạt các tin nhắn theo điều kiện bao gồm đầy đủ các thông tin cần thiết

Bên cạnh đó là tiện ích tìm kiếm nhanh datepicker, đây là tiện ích tìm kiếm nhanh mà người dùng không cần phải nhập thủ công ngày tháng năm mà chỉ cần nhập vào ô trống, màn hình sẽ hiện thị lịch, người dùng chỉ cần tìm ngày tháng năm mình cần và nhấn ok là xong



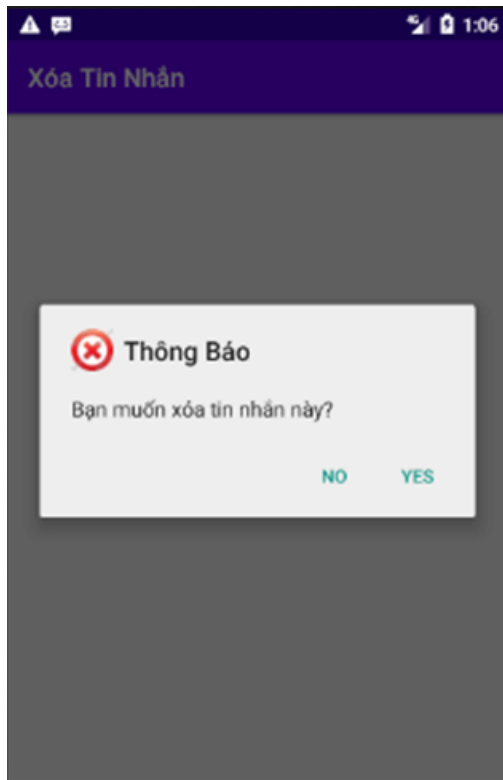
Hình 3.7. Màn hình xem tin nhắn theo ngày



Hình 3.8. giao diện lịch tùy chọn

3.2.3. Giao diện xóa tin nhắn

Đây là chức năng giúp cho người dùng có thể xóa các tin nhắn không cần thiết mà ta không muốn giữ lại nữa. Khi đó người dùng chỉ việc nhấn vào nút đỏ “Delete” bên cạnh các tin nhắn muốn xóa và ấn nút “yes” để xác nhận, lập tức tin nhắn sẽ được xóa cả ở trên điện thoại và trên Host



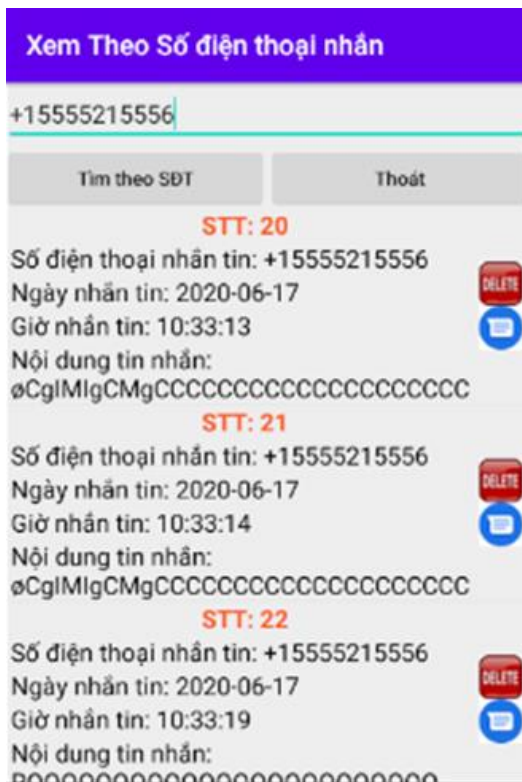
Hình 3.9. Giao diện khi ấn vào nút xóa



Hình 3.10. Giao diện khi xóa thành công

3.2.4. Giao diện xem đầy đủ tin nhắn

Với những tin nhắn quá dài mà màn hình không thể hiển thị được hết, khi đó ta cần phải có chức năng xem được đầy đủ tin nhắn. Muốn đọc được hết những tin nhắn quá dài, người dùng chỉ việc nhấn vào nút xanh bên dưới nút delete, lập tức màn hình sẽ chuyển sang giao diện khác, hiển thị đầy đủ các tin nhắn dài mà bị thiếu



Hình 3.11. Nhấn vào nút xanh bên cạnh tin nhắn dài



Hình 3.12. Giao diện màn hình hiển thị full tin nhắn dài

KẾT LUẬN

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

Đồ án đã thực hiện các nhiệm vụ sau:

- Tìm hiểu lịch sử và kiến trúc của Hệ điều hành Android.
- Nắm bắt được quy trình làm một phần mềm trên điện thoại di động.
- Tìm hiểu về Web, Web Service cho thiết bị di động.
- Tìm hiểu kỹ thuật lấy dữ liệu từ Web service cho thiết bị di động.

Ứng dụng chạy tốt trên mọi hệ điều hành android, tuy nhiên với chức năng bắt tin nhắn đến và đưa lên Host sẽ chỉ chạy được trên android 5 đổ xuống, còn từ android 6 trở lên chức năng này sẽ không sử dụng được, còn ngoài ra các chức năng khác đều hoạt động bình thường. Vì thời gian gấp rút nên hy vọng trong tương lai em sẽ tìm được phương pháp để sửa được vấn đề này và hoàn thiện ứng dụng hơn

TÀI LIỆU THAM KHẢO

- [1] <https://vi.wikipedia.org>
- [2] <https://viettimes.vn>
- [3] <https://quantrimang.com>
- [4] <https://www.json.org>